

UNIVERSITY OF THE PHILIPPINES MANILA  
COLLEGE OF ARTS AND SCIENCES  
DEPARTMENT OF PHYSICAL SCIENCES AND MATHEMATICS

HEALTHY EATING ALERT SYSTEM: FOOD INTAKE  
MONITOR

A special problem in partial fulfillment  
of the requirements for the degree of  
**Bachelor of Science in Computer Science**

Submitted by:

Precious Joy U. Dizon

May 2015

Permission is given for the following people to have access to this SP:

Available to the general public	Yes
Available only after consultation with author/SP adviser	No
Available only to those bound by confidentiality agreement	No

## ACCEPTANCE SHEET

The Special Problem entitled “Healthy Eating Alert System: Food Intake Monitor” prepared and submitted by Precious Joy U. Dizon in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

---

**Avegail D. Carpio, M.Sc.**  
Adviser

### EXAMINERS:

	Approved	Disapproved
1. Gregorio B. Baes, Ph.D. ( <i>candidate</i> )	_____	_____
2. Avegail D. Carpio, M.Sc.	_____	_____
3. Perlita E. Gasmien, M.Sc. ( <i>candidate</i> )	_____	_____
4. Ma. Sheila A. Magboo, M.Sc.	_____	_____
5. Vincent Peter C. Magboo, M.D., M.Sc.	_____	_____
6. Bernie B. Terrado, M.Sc. ( <i>candidate</i> )	_____	_____

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

<hr/> <b>Ma. Sheila A. Magboo, M.Sc.</b> Unit Head Mathematical and Computing Sciences Unit Department of Physical Sciences and Mathematics	<hr/> <b>Marcelina B. Lirazan, Ph.D.</b> Chair Department of Physical Sciences and Mathematics
---	---

---

**Alex C. Gonzaga, Ph.D., Dr.Eng.**  
Dean  
College of Arts and Sciences

## **Abstract**

HEAS:FIM (Healthy Eating Alert System: Food Intake Monitor) is a system that helps monitor fitness and calorie intake towards a healthier living. There may be existing applications but what makes it different is that it automatically computes the amount of calories per serving and restricts certain ingredients based on health conditions, and allows creation of recipes and daily meal planning with corresponding calorific values. Achievements can be shared and there can be health tips, recommended recipes and meals that can be recommended by a nutrition expert.

*Keywords:* HEAS:FIM, Food Intake Monitor, Nutrition, Health Condition

# Contents

Acceptance Sheet	i
Abstract	ii
List of Figures	vii
List of Tables	ix
<b>I. Introduction</b>	<b>1</b>
A. Background of the Study . . . . .	1
B. Statement of the Problem . . . . .	2
C. Objectives of the Study . . . . .	2
D. Significance of the Project . . . . .	4
E. Scope and Limitations . . . . .	4
F. Assumptions . . . . .	5
<b>II. Review of Related Literature</b>	<b>6</b>
<b>III. Theoretical Framework</b>	<b>9</b>
A. Nutrition . . . . .	9
B. IBW (Ideal Body Weight) . . . . .	11
C. BMI (Body Mass Index) . . . . .	11
D. Nutrition Information[5] . . . . .	12
E. Health Conditions . . . . .	29
F. Web Applications . . . . .	34
G. Database Management System . . . . .	35
<b>IV. Design and Implementation</b>	<b>36</b>
A. Context Diagram . . . . .	36
B. Use Cases . . . . .	37
B.1 Top Level Use Case Diagram . . . . .	37

B..2	Profile Management . . . . .	38
B..3	Content Management . . . . .	39
B..4	Input Health Tips . . . . .	40
B..5	Browse Recipes, Food Nutritional Contents, Health Tips . .	41
B..6	Account Management . . . . .	42
B..7	Add User . . . . .	43
B..8	View Users . . . . .	44
B..9	Delete User . . . . .	45
B..10	Manage Password . . . . .	46
B..11	Plan Daily Meals . . . . .	47
C.	Database Structure . . . . .	48
C.1	Entity Relationship Diagram . . . . .	48
C.2	Data Dictionary . . . . .	49
<b>V.</b>	<b>Architecture</b>	<b>57</b>
A.	System Architecture . . . . .	57
B.	Technical Architecture . . . . .	57
<b>VI.</b>	<b>Results</b>	<b>59</b>
A.	General View . . . . .	59
A.1	HEAS:FIM Home Page . . . . .	59
A.2	Profile . . . . .	60
B.	Administrator View . . . . .	60
B.1	Administrator Home Page . . . . .	60
B.2	Add User . . . . .	61
B.3	View Users . . . . .	61
C.	Content Manager View . . . . .	62
C.1	Content Manager Home Page . . . . .	62
C.2	View, Edit, or Delete Food Exchange List Item/s . . . . .	62
C.3	View, Edit, or Delete Health Tip/s . . . . .	63

C..4	Add Content . . . . .	64
C..5	Add Food Exchange List Item . . . . .	64
C..6	Create Recipe . . . . .	65
C..7	Recommend Daily Meals . . . . .	65
C..8	Add Meal . . . . .	66
C..9	View or Delete Meal/s . . . . .	67
C..10	Add Health Tip . . . . .	68
C..11	View . . . . .	69
C..12	View Created Recipes . . . . .	69
D.	User View . . . . .	70
D..1	User Home Page . . . . .	70
D..2	Fitness Profile . . . . .	71
D..3	View Information . . . . .	71
D..4	Add Information . . . . .	72
D..5	Recipes . . . . .	72
D..6	Create Recipe . . . . .	73
D..7	View Created Recipes . . . . .	74
D..8	Plan Daily Meals . . . . .	74
D..9	Add Meal . . . . .	75
D..10	View or Delete Meal/s . . . . .	76
D..11	Share Achievement . . . . .	77
D..12	Share Achievement on Site . . . . .	77
D..13	Share Achievement on Facebook . . . . .	78
D..14	Browse . . . . .	78
D..15	Browse Shared Recipes . . . . .	79
D..16	Browse Recommended Recipes . . . . .	79
D..17	Browse Recommended Daily Meals . . . . .	79
D..18	Browse Food Nutritional Information . . . . .	80

**VII. Discussions 81**

<b>VIII. Conclusions</b>	<b>83</b>
<b>IX. Recommendations</b>	<b>84</b>
<b>X. Bibliography</b>	<b>85</b>
<b>XI. Appendix</b>	<b>88</b>
A. Forms . . . . .	88
B. Source Code . . . . .	168
B.1 Configs . . . . .	168
B.2 Controllers . . . . .	175
B.3 Models . . . . .	211
B.4 Libraries . . . . .	226
B.5 Layout . . . . .	262
<b>XII. Acknowledgement</b>	<b>271</b>

## List of Figures

1	Context Diagram of HEAS:FIM . . . . .	36
2	Top Level Use Case Diagram of HEAS:FIM . . . . .	37
3	Activity Diagram: HEAS:FIM Profile Management . . . . .	38
4	Use Case (HEAS:FIM Content Management) . . . . .	39
5	Activity Diagram: HEAS:FIM Input Health Tips . . . . .	40
6	Activity Diagram: HEAS:FIM Browse Recipes, Food Nutritional Contents, Health Tips . . . . .	41
7	Use Case (HEAS:FIM Account Management) . . . . .	42
8	Activity Diagram: HEAS:FIM Add User . . . . .	43
9	Activity Diagram: HEAS:FIM View Users . . . . .	44
10	Activity Diagram: HEAS:FIM Delete User . . . . .	45
11	Activity Diagram: HEAS:FIM Manage Password . . . . .	46
12	Flow Chart: HEAS:FIM Plan Daily Meals . . . . .	47
13	HEAS:FIM Entity Relationship Diagram . . . . .	48
14	HEAS:FIM System Architecture . . . . .	57
15	HEAS:FIM Home Page . . . . .	59
16	HEAS:FIM Profile . . . . .	60
17	HEAS:FIM Administrator Home Page . . . . .	60
18	Add User . . . . .	61
19	View Users . . . . .	61
20	HEAS:FIM Content Manager Home Page . . . . .	62
21	View, Edit, or Delete Food Exchange List Item/s . . . . .	62
22	View, Edit, or Delete Health Tip/s . . . . .	63
23	Add Content . . . . .	64
24	Add Food Exchange List Item . . . . .	64
25	Create Recipe . . . . .	65
26	Recommend Daily Meals . . . . .	65
27	Add Meal . . . . .	66



28	Add Recipe Meal . . . . .	66
29	View Recipe Meal . . . . .	67
30	View Meal/s . . . . .	67
31	View Added Meals . . . . .	68
32	Add Health Tip . . . . .	68
33	View . . . . .	69
34	View Created Recipes . . . . .	69
35	HEAS:FIM User Home Page . . . . .	70
36	Fitness Profile . . . . .	71
37	View Information . . . . .	71
38	Add Information . . . . .	72
39	Recipes . . . . .	72
40	Create Recipe . . . . .	73
41	View Created Recipes . . . . .	74
42	Plan Daily Meals . . . . .	74
43	Add Meal . . . . .	75
44	Add Recipe Meal . . . . .	75
45	View or Delete Added Meals . . . . .	76
46	Share Achievement . . . . .	77
47	Share Achievement on Site . . . . .	77
48	Share Achievement on Facebook . . . . .	78
49	Browse . . . . .	78
50	Browse Shared Recipes . . . . .	79
51	Browse Recommended Recipes . . . . .	79
52	Browse Recommended Daily Meals . . . . .	79
53	Browse Food Nutritional Information . . . . .	80

## List of Tables

1	The Calorie Need for Male . . . . .	10
2	The Calorie Need for Female . . . . .	10
3	The Calorie Need for Pregnancy and Lactating . . . . .	11
4	Body Mass Index . . . . .	12
5	Composition of Food Exchanges . . . . .	13
6	Vegetable Exchanges Equivalent . . . . .	13
7	Vegetable Exchanges . . . . .	15
8	Processed . . . . .	16
9	Fruit Exchange . . . . .	17
10	Milk Exchanges Equivalent . . . . .	18
11	Milk Exchanges . . . . .	18
12	Rice Exchanges . . . . .	19
13	Meat and Fish Exchange Equivalent . . . . .	20
14	A. Low Fat Meat and Fish Exchanges . . . . .	20
15	B. Medium Fat Meat and Fish Exchanges . . . . .	25
16	C. High Fat Meat and Fish Exchanges . . . . .	27
17	Fat Exchange List . . . . .	28
18	Daily Calorie Meal Plans . . . . .	30
19	Dietary Approach to Stop Hypertension . . . . .	31
20	users table . . . . .	49
21	fitness_profile table . . . . .	50
22	health_conditions table . . . . .	51
23	own_recipes table . . . . .	51
24	recommended_recipes table . . . . .	52
25	shared_recipes table . . . . .	52
26	recipe_images table . . . . .	53
27	recipe_images_recommended table . . . . .	53
28	plan_daily_meals table . . . . .	53

29	recommended_daily_meals table . . . . .	54
30	shared_achievements table . . . . .	54
31	health_tips table . . . . .	54
32	ingredients table . . . . .	55
33	ingredients_recommended table . . . . .	55
34	food_exchange_list table . . . . .	56
35	health_conditions_restrictions table . . . . .	56

# I. Introduction

## A. Background of the Study

Nutrition, nourishment, or aliment, is the supply of materials, specifically food, required by organisms and cells in order to survive[1]. In science and human medicine, nutrition is the consumption and utilization of food[1]. Food is a material that consists essentially of protein, carbohydrate, and fat used by the body of an organism in growth sustenance, repair, vital processes and in furnishing of energy[2]. It is a basic necessity of every human being and is the body's main source of nutrients. A nutrient is a source of nourishment, an ingredient in food, e.g. protein, carbohydrate, fat, vitamin, mineral, fiber and water[1]. The nutritional condition of a person is linked with his/her diet. Diet is the measure of the state of one's health[3]. Proper nutrition can be attained through proper diet. Food may be the driving force that the body needs but there are certain limitations in order to achieve a healthier diet.

One needs a sufficient allowance of food in order to meet the nutritional needs of the body[4]. A sensible diet ensures that all food groups are represented in daily meals[5]. Improper diet can be the cause of certain health problems and conditions. A person's diet may depend on factors like nationality, culture, influence, present health condition, nature of work, or simply on the available food.

The Filipino diet is as much a reflection of its history as it is a reflection of the country's geography and natural resources[6]. Filipinos have a vast variety of food to choose from. Aside from the local dishes, the Filipino cuisine was further compounded by the coming of several foreign influences[6]. The current Filipino diet is said to be very high in sodium and fat owing perhaps to predominant cooking techniques such as deep frying, the increasing preference for meat products such as pork, and the heavy seasoning that comes along with every dish[6]. This is a

far cry from the traditional practices of just boiling vegetables and grilling fish, foods that are now only commonly consumed in some provinces[6].

There are existing diet applications that monitor food intake. SparkPeople[7] is a web application that allows daily meal planning, takes note of health conditions and has a reward system. NutriCoach[8] analyzes nutritional and calorific information of food intake for one day meal. Fat Secret[9] offers the same functionalities as SparkPeople[7].

## **B. Statement of the Problem**

Existing diet applications such as SparkPeople[7] does not allow creation of recipes from scratch and restriction of ingredients for every health condition. NutriCoach[8] does not allow creation of recipes from scratch, saving of data for history, and also does not take note of health conditions. Fat Secret[9] does not take note of health conditions and there is no automatic counting of calorie need.

Although there are a lot of web and mobile applications that monitor food intake, there is no application that allows daily meal planning, creation of recipes from scratch with corresponding calorific values, restriction of ingredients considering health conditions, weight and calorie monitor, daily health tips, and reward system all at the same time.

## **C. Objectives of the Study**

The main function of the system is the ability to plan daily meals from created or browsed recipes with ingredient/intake restrictions taking into consideration Health Conditions. It allows computation and classification of BMI (Body Mass Index) and Nutritional Status. The calorie need is determined based on sex, gender, weight, height and health conditions.

This research will create a Food Intake Monitoring system with the following functionalities:

1. Allows the Registered User to
  - (a) Input and Edit Profile Information
  - (b) Input Health Conditions
  - (c) Input Daily Calorie Intake
  - (d) Browse for Recommended Recipes
  - (e) Browse Food Nutritional Information
  - (f) Compute Body Mass Index
  - (g) Monitor Weight via Graph
  - (h) Monitor the Calorie Balance
  - (i) Create Own Recipes with Corresponding Calorific Values based on Health Conditions
  - (j) Plan Daily Meals Considering the Health Conditions
  - (k) Share Achievements and Recipes to Other Registered Users, on Facebook or Twitter
2. Allow the Content Manager to
  - (a) Input and Share Recommended Recipes with the Corresponding Nutritional Contents Considering Health Conditions
  - (b) Recommend Daily Meals Considering the Health Conditions
  - (c) Input and Share Health Tips
  - (d) Input Calorific Equivalent of Food Items and Ingredients in the Food Exchange List
3. Allow the Administrator to
  - (a) Manage User Accounts and Their Roles

## **D. Significance of the Project**

Filipinos love to eat. How could one resist having a vast, insanely delicious variety of food to choose from? But most of the time, our taste buds crave for the unhealthy ones for they seem to be more tasty. When the food tastes so good, we tend to be indulgent and eat more. Sometimes, we are unaware of the nutritional contents of what we eat and we exceed beyond the recommended daily calorie intake. Among the six nutrients that are vital to optimal health, fat receives the most attention among weight watchers and those with health conditions[10]. According to the National Institute of Health, Filipinos are at high risk of having fatty liver syndrome, hypertension, atherosclerosis, heart attack and stroke[10]. The lack of monitoring of food intake, nutritional information, content per serving, and weight constitute to unhealthy diet that may lead to certain health problems and conditions.

This system will help monitor the calories of food and regulate food intake towards a healthier diet and ideal weight. The Body Mass Index can be computed to know if one is underweight, normal weight, overweight, or obese. There would be a recommendation to gain, maintain, or lose weight depending on the BMI Classification and Nutritional Status. With the current weight, the goal weight can be set up. To achieve the goal weight, the weight progress, calorie intake and daily meals can be monitored, considering the health conditions linked to diet, accompanied by recommended, custom, and shared recipes with the corresponding nutritional information.

## **E. Scope and Limitations**

1. This system will focus on the general Filipino diet.
2. There will be no recommendations for cultural or religious restrictions.
3. The target weight is based on the Body Mass Index.

4. The nutritional information will depend on the Food Exchange List from Food and Nutrition Research Institute.
5. The Health Conditions considered are Diabetes, Hypertension, and Kidney Stones.

## **F. Assumptions**

1. The weight condition is determined by the Body Mass Index.
2. The monitoring of the calorie intake is based on the daily meals.
3. The Content Manager is a Nutrition Expert.
4. The restrictions of the amount of ingredients for the health conditions shall be based on existing guidelines from the National Institutes of Health.



## II. Review of Related Literature

Self-monitoring is the centerpiece of behavioral weight loss intervention programs[11]. Diet is one component of self-monitoring in behavioral weight loss studies that includes recording dietary intake so that individuals are aware of their current behaviors[11]. There are many factors that may impact successful long-term change in eating habits[12]. For people who make a commitment to changing their eating habits, accurate logs of what they eat may help in monitoring progress toward set goals[12]. Monitoring food intake has been shown to help individuals become more aware of what and how much they eat and drink[13].

There are a number of different types of data that have a part to play in comprehensive nutritional intake information capture which may include: Food Type (where no food identifier is available), Food Portion Size, Food Identifier Information (Food identifiers including barcodes, are not always available, but when they are available, they provide a powerful and unambiguous information input.), Nutrition Fact Panels, Recipe Information, Aggregate Intake Information, Estimation Approaches[14]. The Nutrition Fact panel on food packaging provides calorie information for each serving of food or beverage and can assist consumers in monitoring their intake[13].

Currently, food logging is typically done by hand using paper diaries, spreadsheets, or a growing number of specialized applications[12]. Nutritionists have explored alternative methods such as daily interviews with trained experts[12]. While these methods improve accuracy, they are costly and still require substantial time investment[12]. Thus, the benefit of specialized applications to an individual is the reduction in the cost of healthcare.[15] There are existing applications for self-monitoring of diet.

SparkPeople[7] is an application that considers if a female is pregnant and lactat-

ing, monitors weight and weight goal if one wants to lose weight 1 kg or 2 kg per week or just maintain, takes note of one's activeness to be sedentary, lightly active, or active, takes note of health conditions, offers parenting tips if one has kids, allows daily meal planning with nutritional information, and offers a reward system.

PlateMate is a system that allows users to take photos of their meals and receive estimates of food intake and composition[12]. It crowdsources nutritional analysis from photographs using Amazon Mechanical Turk, automatically coordinating untrained workers to estimate a meal's calories, fat, carbohydrates, and protein[12]. It reduces user effort but is not always accurate because sometimes, the food on the photo taken is not properly recognized.

Similar to PlateMate is the FoodLog[16]. It is a multimedia food-recording system where users upload photographs of their meals then a food diary is automatically constructed using image-processing functions such as food-image detection and food-balance estimation[16]. The dietary composition of the meal is recorded and the information is recorded for later viewing in formats such as a figure or a symbolic graphical representation[16]. It provides a convenient way of recording a users diet, but in the current state, the image-processing algorithms for estimating dietary composition are not very accurate[16]. The main reason for this is the use of a single global model for analyzing the photographs from users who may have different dietary preferences and habits[16].

Lose It is an application that allows the user to set weight loss goals and establish a daily calorie budget based on the Mifflin-St. Jeor equation (Estimating Percent Calorie Restriction, [scientificpsychic.com](http://scientificpsychic.com))[17]. The user enters the food consumed and the program subtracts the caloric amount of the food from the total calories allowed per day[17]. The allowable caloric recommendation can be altered by entering exercise activities and durations, which are also tracked and can add

calories to the overall caloric budget[17]. Diet and exercise data are entered using a searchable database (loseit.com)[17]. Weight may be entered on a daily basis and feedback regarding macronutrient percentages and calories burned through exercise is given via a daily report[17].

Food Fight is an application that collects dietary and lifestyle data from the user[18]. There is a reward system for those who have achieved the goal based on actual results. There are more similar diet self-monitoring applications such as SapoFitness[19], Hyperfit[18], POND(Pattern-Oriented Nutrition Diary)[20], SuperTracker[13], Calorie Counter[21], My Diet Diary Calorie Counter[22], MyFitnessPal[23], NutriCoach[8], and Fat Secret[9] that serve as a reminder tool, calorie tracker, meal planner, or simply a food diary.

### III. Theoretical Framework

#### A. Nutrition

One of the important things in health information is nutrition[15]. Food intake and nutritional status affect health significantly[15]. Every food has more than one type of nutrients[15]. Since every food has different nutrients, it is advisable to consume various food in one meal that have the source of energy substance (carbohydrates), builder substance (protein), and regulator substance (vitamin and mineral)[15].

The nutritional needs of a person vary depending on factors such as weight, height, and age.

For the male calorie need, where  $C$  is calories,  $W$  is weight,  $H$  is height, and  $A$  is age[15]:

$$C = 66 + (13.7 * W) + 5 (H) - (6.8 * A)$$

For the female calorie need, where  $C$  is calories,  $W$  is weight,  $H$  is height, and  $A$  is age[15]:

$$C = 655 + (9.6 * W) + (1.7 * H) - (4.7 * A)$$

For the male and female protein need, where  $C$  is calories and  $NC$  is the need of calories[15]:

$$C = 0.25 * NC$$

For the male and female fat need, where  $C$  is calories and  $NC$  is the need of calories[15]:

$$C = 0.70 * NC$$

For the male and female carbohydrate need, where C is calories and NC is the need of calories[15]:

$$C = 0.65 * NC$$

Based on age group (year), weight (kilogram=kg), height (centimeter=cm), energy (kilocalories=kcal), protein (gram=gr), and activity, the calorie needs for male, female and daily menu for household dose and nutritional needs of families in a day can be seen in Table 1, Table 2, and Table 3[15]:

Table 1: The Calorie Need for Male

Age group (year)	Weight (kg)	Height (cm)	Energy (kcal)	Protein (gr)	Activity
3-9	12	90	1500	35	Normal
10-12	30	135	2000	45	Normal
13-15	45	150	2400	69	Normal
16-19	56	160	2500	66	Normal
20-59	62	165	2800	55	Light
20-59	62	165	3600	55	Medium
20-59	62	165	3600	55	Heavy
60+	62	165	2200	55	Normal

Table 2: The Calorie Need for Female

Age group (year)	Weight (kg)	Height (cm)	Energy (kcal)	Protein (gr)	Activity
3-9	12	90	1500	35	Normal
10-12	35	149	1900	54	Normal
13-15	46	153	21200	62	Normal
16-19	50	156	2000	51	Normal
20-59	54	156	2050	48	Light
20-59	54	156	2250	48	Medium
20-59	54	156	2600	48	Heavy
60+	54	54	1850	48	Normal

Table 3: The Calorie Need for Pregnancy and Lactating

Age group (year)	Weight (kg)	Height (cm)	Energy (kcal)	Protein (gr)	Activity
-	-	-	+285	+12	Pregnant
0-6 month	-	-	+70	+16	Lactating
7-12 month	-	-	+500	+12	Lactating
60+	-	-	+400	+11	Lactating

## B. IBW (Ideal Body Weight)

The Ideal Body Weight (IBW) in kg can be computed with the given formula below [24]:

Males:  $IBW = 50 \text{ kg} + 2.3 \text{ kg}$  for each inch over 5 feet.

Females:  $IBW = 45.5 \text{ kg} + 2.3 \text{ kg}$  for each inch over 5 feet.

## C. BMI (Body Mass Index)

Body Mass Index (BMI) is an inexpensive and easy-to-perform method used to find out if a person is underweight, normal weight, overweight, or obese[15]. It is obtained from a persons weight and height. It provides a reliable indicator of body fatness for most people and is used to screen for weight categories that may lead to health problems[15]. It can be considered an alternative for direct measures of body fat[15]. However, it is not a diagnostic tool[15]. For children and teens, BMI is age and sex-specific and is often referred to as BMI-for-age[15]. For adults, BMI is interpreted through categories that do not take into account sex or age[15].

The Body Mass Index (BMI) where BMI is Body Mass Index, W is weight and H is height[15]:

$$BMI = W / H^2$$

BMI Classification and Nutritional Status can be seen in Table 4[15]:

Table 4: Body Mass Index

BMI	Classification	Nutritional Status
<18.5	Under weight	Malnutrition
18.5 - 24.9	Healthy weight	Normal
25.0 - 34.9	Over weight	Over Nutrition
>35.0	Obese	Very Excess Nutrition

## D. Nutrition Information[5]

### Food Exchange List

A sensible diet ensures that all food groups are represented in daily meals. The exchange list provides a good base for planning meals by classifying foods into seven groups. These are: vegetables, fruits, milk, rice, meat, fat and sugar.

Each item in any of the groups may be substituted for any other item in the same group, as long as it is taken in the amount indicated. But foods listed in different groups are not of equal value and may not be interchanged.

From the Rice Exchange List for instance, you may have 2 slices of bread in exchange for half a cup of rice. But you may not substitute half a cup of mongo since this belongs to a different group altogether.

Legend: g = gram, EP = edible portion, AP = as purchased

Each exchange in the list contains:

Table 5: Composition of Food Exchanges

List	Food	Measure (g)	Carbohydrate (g)	Protein	Fat	Energy (Kcal)	(Kj)
I-A	Veg. A	1 cup raw					
		$\frac{1}{2}$ cup cooked					
I-A	Veg. A	2 cups raw	3	1		16	67
		1 cup cooked or					
I-B	Veg. B	$\frac{1}{2}$ cup raw	3	1		16	67
		$\frac{1}{2}$ cup cooked					
II	Fruit	varies	10			40	167
III	Milk						
	Whole	varies	12	8	10	170	711
	Low Fat	4 tbsp	12	8	5	125	523
	Skimmed	varies	12	8	tr	80	335
IV	Rice	varies	23	2		100	418
V	Meat						
	Low Fat	varies		8	1	41	172
	Medium Fat	varies		8	6	86	360
	High Fat	varies		8	10	122	510
VI	Fat	1tbsp			5	45	188

Table 6: Vegetable Exchanges Equivalent

	CHO	PRO	FAT	ENERGY
	(g)	(g)	(g)	(KCalories)
Group A	-	-	-	-
Group B	3	1	-	16



Group A vegetables contain negligible carbohydrates, protein, and energy if 1 exchange or less is used. When 2 exchanges are used, compute as one Group B Vegetable. Unless otherwise specified, the portion size for one vegetable exchange is:

Vegetable A: 1 exchange = 1 cup raw (25g) or  $\frac{1}{2}$  cup cooked (45g)

Vegetable B: 1 exchange =  $\frac{1}{2}$  cup raw (40g) or  $\frac{1}{2}$  cup cooked (45 g)

This list shows the kind of vegetables to use for one vegetable exchange:

1 exchange = 10 grams carbohydrate = 40 Kcalories

The list shows the kind and amounts of foods to use for one fruit exchange:

Table 7: Vegetable Exchanges

Group A	Group B
Acelgas	Fresh; Abitsuela, bunga (snap green beans pod)
Alagaw Leaves	Carrot
Alugbati Leaves	Coconut Shoot
Ampalaya Fruit	Cowpea, pods (bunga)
Balbalulang (Seaweed)	Lima, beans, pods (patani)
Bamboo Shoot (Labong)	Langka, hilaw
Banana Heart (Puso ng Saging)	Mungbean Sprout (toge)
Bataw Pods	Pigeon Pea Pods
Beets	Rimas
Cabbage	Singkamas Pods (bunga)
Camote Leaves	Singkamas Tuber (lamang gubat)
Cassava Leaves and Tops	Squash Fruit
Cauliflower	Stringbeans Pod (sitaw, bunga)
Celery	
Chayote Fruit	
Chayote Leaves	
Cucumber	
Eggplant	
Gabi Leaves	
Himbabao	
Kangkong	
Katuray	
Katuray Flowers	
Lettuce	
Malunggay Leaves	
Malunggay Pods	
Mushroom Fresh	
Mustard Leaves	
Okra	
Onion Bulb	
Pako	
Papaya Green	
Patola	
Pepper Fruit	
Pepper Leaves	
Petchay	
Pokpoklo (seaweed)	
Radish	
Saluyot	
Sigarilyas Pods	
Spinach	
Squash Flower (kalabasa, bulaklak)	
Squash Leaves (kalabasa, dahon)	
Sweet pea pods (sitsaro)	
Talinum	
Tomato	
Unsoy	
Upo	

Table 8: Processed

FOOD	Weight (g) Edible Portion	MEASURE
Asparagus Tips	90	1 cup
Baby Corn	15	2 (8 cm long x 5- $\frac{1}{2}$ cm circumference each)
Green Peas, canned (gisantes, canned)	15	1 tablespoon
Golden Sweet Corn, canned	20	2 tablespoons
Mushroom, canned	110	$\frac{1}{3}$ cup
Tomato Juice, canned	60	$\frac{1}{2}$ cup undiluted
Water Chestnut, canned	25	3 (2cm diameter each)

One exchange of each of the sub-groups of milk contains:

1 Exchange = 23 grams carbohydrate

(2 grams protein) = 100 Calories

This list shows the kinds and amounts of rice, rice equivalents, bread and bakery products to use for one rice exchange:

Table 9: Fruit Exchange

FOOD	Weight (g)		MEASURE
	A. P.	E.P.	
Anonas	65	35	$\frac{1}{2}$ of 5 x 8 cm
Apple	86	65	$\frac{1}{2}$ of 8 cm diameter or 1 (6 cm diameter)
Atis	70	45	1 (5 cm diameter)
Balimbing	153	135	1 $\frac{1}{2}$ of 9 x 5 cm
Banana			
Bungulan	60	40	$\frac{1}{2}$ of 15 x 4 cm
Lakatan	51	40	1 (9 x 3 cm)
Saba	70	40	1 (1 x 4 cm)
Cashew	78	70	1 (7 x 6 - $\frac{1}{2}$ cm)
Chico	54	45	1 (4 cm diameter)
Dalanghita	300	135	2 (6 cm diameter each)
Datiles	61	50	1 cup
Duhat	80	60	20 (2 cm diameter each)
Durian	150	130	1 segment of 6 - $\frac{2}{3}$ x 4 $\frac{1}{2}$ cm or 1 $\frac{1}{2}$ tbsp.
Grapes	69	55	10 (2 cm diameter each) or 4 (3 cm diameter each)
Guava	81	80	2 (4 cm diameter each)
Guyabano	86	60	1 slice (8 x 6 x 2 cm) or $\frac{1}{2}$ cup
Jackfruit, ripe	118	40	3 segments (6 cm diameter each)
Kamachile	110	55	7 pods
Lanzones	103	70	7 (4 x 2 cm each)
Lychees	77	50	5 (3 cm diameter each)
Mabolo	83	50	$\frac{2}{3}$ of 6 cm diameter
Makopa	169	135	3 (4 cm diameter each)
Mango, green	90	65	1 slice (11 x 6 cm)
Medium ripe	90	65	1 slice (11 x 6 cm)
Ripe	103	60	1 slice (12 x 7 cm) $\frac{1}{2}$ cup cubed
Indian	140	80	1 (6 cm diameter)
Paho	92	70	9 small
Mangosteen	212	55	3 (6 cm diameter each)
Marang	21	35	$\frac{1}{2}$ of 12 x 10 cm
Melon			
Kastila	317	200	1 slice (12 x 10 x 3 cm) or 1 - $\frac{1}{3}$ cup
Papaya, ripe	133	85	1 slice (10 x 6 x 2 cm) or $\frac{3}{4}$ cup
Pear	118	85	1 slice (10 x 6 x 2 cm) or $\frac{1}{2}$ cup
Rambutan	139	50	8 (3 cm diameter each)
Santol	127	75	1 (7 cm diameter each)
Singkamas tuber	124	110	$\frac{1}{2}$ of (9 cm diameter each) or 1 cup
Siniguelas	78	50	5 (3 cm diameter each)
Star Apple	123	65	$\frac{1}{2}$ of 6 cm diameter
Strawberry	168	165	1 $\frac{1}{4}$ cups
Suha	160	90	3 segments (8 x 4 x 3 cm each)
Tamarind, ripe	34	15	2 of 6 cm diameter
Tiesa	41	30	$\frac{1}{4}$ of 10 cm diameter
Watermelon	226	140	1 slice (12 x 6 x 3 cm) or 1 cup

Table 10: Milk Exchanges Equivalent

	CHO	PRO	FAT	ENERGY (KCalories)
Whole Milk	12	8	10	170
Low Fat Milk	12	8	5	125
Skimmed/Very Low Fat Milk	12	8	Trace	80

Table 11: Milk Exchanges

FOOD	Weight (g) A.P. / E.P.	MEASURE
*Whole Milk		
Milk, evaporated undiluted	125	$\frac{1}{2}$ cup
Milk, evaporated filled undiluted	125	$\frac{1}{2}$ cup
Milk, fresh carabao's	250	1 cup
Milk, fresh cow's	250	1 cup
Milk, powdered	30	$\frac{1}{4}$ cup or 4 level tbsp.
*Low Fat Milk		
Powdered	30	$\frac{1}{4}$ cup or 4 level tbsp
Lite Low Fat Milk	250	1 tetra-brick
*Skimmed (Non-Fat)/Very Low Fat Milk		
Buttermilk Liquid	185	$\frac{2}{3}$ cup
Powdered	25	$\frac{1}{4}$ cup or 4 level tbsp
Long Life Skimmed Milk	250	1 cup
Yoghurt	125	$\frac{1}{2}$ cup

Table 12: Rice Exchanges

FOOD	Weight (g)		MEASURE
	A.P. / E.P.		
<b>A. RICE AND RICE PRODUCTS</b>			
1. Rice, cooked		80	$\frac{1}{2}$ cup, packed
2. Rice, gruel (lugaw)			
Thin consistency		705	$4\frac{1}{2}$ cups
Medium consistency		435	3 cups
Thick consistency		250	$1\frac{1}{2}$ cups
<b>B. RICE EQUIVALENTS</b>			
1. Bread			
Pan Americano		40	
Pan de Bonete		40	
Pan de Leche		40	
Pan de Monay		40	
Pan de Sal Rolls		40	
(hotdog/hamburger)			
Whole wheat bread			
		45	
2. Corn and Corn Products			
Binatog		90	
Corn/Rice curls and the like		25	
Corn, boiled	120	65	
Baby corn		90	
3. Noodles, cooked Bihon			
Macaroni,			
Sotanghon,			
Spaghetti			
4. Oatmeal		185	1 cup
5. Rootcrops			
Sweet Potato	91	80	$\frac{1}{2}$ of 11 cm long x $4\frac{1}{2}$ cm diameter or $\frac{1}{2}$ cup
Cassava	115	85	1 (5cm long x 4 cm diameter) or 1 cup
Gabi	130	100	2 (6cm long x 4cm diameter each) or 1 cup
Potato	195	165	$2\frac{1}{2}$ (7 cm long x 4cm diameter each) or $1\frac{1}{3}$ cups
Ube	155	130	1 ( $8\frac{1}{2}$ cm long x $4\frac{1}{2}$ cm diameter)
6. Beans and Nuts			
Chestnuts, roasted	55	40	11 pieces large or 20 pieces small
(Kastanyas, binusa)			
7. Other			
Cornstarch		25	5 teaspoons
Flour, all purpose		25	3 tablespoons
Sago, cooked		120	$\frac{1}{2}$ , cooked

This list is subdivided into 3 sub-groups based on the amount of fat and calories:  
 Low Fat, Medium Fat and High Fat meat and fish.

Each exchange of meat and fish or substitute in the sub-group contains:

Table 13: Meat and Fish Exchange Equivalent

	CHO	PRO	FAT	ENERGY (KCalories)
Low Fat Meat and Fish	-	8	1	41
Medium Fat Meat and Fish	-	8	6	86
High Fat Meat and Fish	-	8	10	122

### List 5 Meat and Fish Exchange

This list shows the kinds and amounts of low fat meat and fish to use for one low fat meat and fish exchange:

Table 14: A. Low Fat Meat and Fish Exchanges

FOOD	Wt. (g)	MEASURE
E. P. cooked		
I. Lean Meat		
a. Beef		
Shank (bias, lean meat, laman)		
Round (pierna corta or pierna larga)	30	1 slice, matchbox size (5 x 3- $\frac{1}{2}$ x 1 - $\frac{1}{2}$ cm)
Tenderloin (solomilyo),		
Sirloin Steak (tagiliran, hulihan)		
Centerloin (tagiliran, unahan)		
b. Carabeef		
Shank (bias) round (hita)		

*Continued on next page*

Table 14 – *Continued from previous page*

FOOD	Wt. (g)	MEASURE
	E. P. cooked	
Meat (laman) bahagya,	30	1 slice matchbox size (5 x 3- $\frac{1}{2}$ x 1- $\frac{1}{2}$ cm)
Katamtaman at walang taba		
Shoulder (paypay), round (pierna corta at pierna larga)		
Rump (tapadera)		
c. Lean Pork		
Tenderloin, well trimmed	30	1 slice matchbox size (5 x 3- $\frac{1}{2}$ x 1- $\frac{1}{2}$ cm)
d. Chicken		
Leg (hita) or	30	1 small leg (13 - $\frac{1}{2}$ cm long x 3 cm diameter )
Meat (laman) or	30	1 slice matchbox slice (5x3- $\frac{1}{2}$ cm x 1- $\frac{1}{2}$ cm)
Breast meat (pitso)	30	$\frac{1}{4}$ breast (6cm long)
2. Variety Meats/ Internal Organs		
Blood (dugo)		
- pork, beef, chicken		

*Continued on next page*



Table 14 – *Continued from previous page*

FOOD	Wt. (g)	MEASURE
	E. P. cooked	
Gizzard (balun-balunan)		
- chicken		
Heart (puso)		
- pork, beef, carabeef		
Liver (atay)		
- pork, beef, carabeef, chicken		
Lungs (baga)		
- pork, beef, carabeef, chicken	35	$\frac{3}{4}$ cup
Omassum (librilyo)		
- beef, carabeef		
Small Intestine (bitukang maliit)		
- pork, beef, carabeef		
Spleen (lapay)		
- pork, beef, carabeef		
Tripe (goto)		
- beef		
Uterus (bahay guya)		
- pork, beef		
3. Fish		
Large Variety: (e.g. bakoko, bangus, dalag, labahita, lapu-lapu, etc.)	35	1 slice (7 x 3 x 2 cm)
Medium Variety:		
Hasa-hasa, dalagang bukid	35	1 (18 x 4- $\frac{1}{2}$ cm)
Galunggong	35	1 (14 x 3- $\frac{1}{2}$ cm)

*Continued on next page*

Table 14 – *Continued from previous page*

FOOD	Wt. (g)	MEASURE
	E. P. cooked	
Hito	35	$\frac{1}{2}$ of 22 x 5 cm
Small Variety: Sapsap	35	2 (10 x 5 cm each)
Tilapya	35	2 (12 x 5 cm each)
Tamban	35	2 (12 - $\frac{1}{2}$ x 3 cm each)
Dilis	35	$\frac{1}{4}$ cup
4. Other Seafoods		
Alamang, tagunton	30	1- $\frac{1}{4}$ tablespoons
Aligue:		
Alimango	15	1 tablespoon
Alimasag	50	3 tablespoons
Alimango/Alimasag, laman	20	$\frac{1}{4}$ cup or $\frac{1}{2}$ piece medium
Lobster	30	2 tablespoons
Talangka	30	75 pieces A. P.
Shrimps: Puti	25	5 (12cm each)
Sugpo	25	2 (13cm each)
Suwahe	25	5 (12cm each)
Octopus (pugita)	30	$\frac{1}{2}$ cup
Squid (pusit)	25	3 (7 x 3 cm each)
Shells: Halaan	75	$\frac{1}{3}$ cup shelled or 3 cups with shell
Kuhol	50	$\frac{1}{2}$ cup shelled or 3 cups with shell
Susong pilipit	65	$\frac{1}{3}$ cup shelled or

*Continued on next page*

Table 14 – *Continued from previous page*

FOOD	Wt. (g)	MEASURE
	E. P. cooked	
		2 cups with shell
Paros	60	1 cup shelled or 2- $\frac{2}{3}$ cups with shell
5. Beans		
Pigeon pea seeds, dried (kadyos, buto, tuyo)	55	$\frac{1}{3}$ cup
6. Cheese		
Cottage cheese	60	$\frac{1}{3}$ cup
7. Processed Foods		
Dried:		
Daing:		
Alakaak, Alumahan,	20	1 (15- $\frac{1}{2}$ x 8 cm)
Bisugo, Biyang puti		
Lapu-lapu	20	$\frac{1}{4}$ of 30 x 40 cm
Sapsap	20	3 (9 x 5 cm each)
Tamban	20	1 (16 x 3 cm)
Tinapa:		
Bangus	30	$\frac{1}{4}$ of 20 x 8 cm
Galunggong	30	1 (16 x 4 cm)
Tamban	25	1 (16 x 5 cm)
Tuyo:		
Alamang	15	$\frac{1}{3}$ cup
Ayungin, dilis sapsap	20	3 (11 - $\frac{1}{2}$ x 8cm each)
Tunsoy	15	1 (8 x 1 cm)

*Continued on next page*

Table 14 – *Continued from previous page*

FOOD	Wt. (g)	MEASURE
	E. P. cooked	
Pusit		
Canned:		
Salmon	40	$\frac{1}{3}$ cup flaked
Tuna in brine	30	$\frac{1}{3}$ cup flaked

This list shows the kinds and amounts of medium fat meat and fish to use for one medium fat meat and fish exchange:

Table 15: B. Medium Fat Meat and Fish Exchanges

FOOD	Wt. (g) E. P. cooked	MEASURE
I. Lean Meat		
a. Beef		
Flank (kabilugan)	30	1 slice matchbox size (5 x 3- $\frac{1}{2}$ $\frac{1}{2}$ x 1- $\frac{1}{2}$ cm)
Brisket (punta y pecho)		
Plate (tadyang)		
Chuck (paypay)		
b. Pork		
Leg (pata)	30	1 slice (4 cm dia. x 2 leg (pata) cm thick)
2. Variety	35	$\frac{1}{4}$ cup
Meats/Internal Organs		

*Continued on next page*

Table 15 – *Continued from previous page*

FOOD	Wt. (g) E. P. cooked	MEASURE
Brain (utak)		
- pork, beef, carabeef		
3. Fish		
Carpa	35	1 slice (15 x 7 x 2 cm)
4. Egg		
Chicken	60	1 piece
Quail's egg	70	9 pieces
Salted duck's egg	60	1 piece
5. Cheese		
Cheese, cheddar	35	1 slice (6 x 3 x 2 cm)
6. Chicken		
Wings (pakpak)	25	1 medium or 2 small
Head (ulo)	35	2 heads
7. Beans		
Soybeans (utaw)	40	$\frac{1}{2}$ cup
8. Processed foods		
a. Fish Products		
Sardines canned		
in oil/tomato		
sauce	45	1 (10 x 4- $\frac{1}{2}$ cm)
Tuna sardines	50	1- $\frac{1}{2}$ of 6 x 4 x 3 cm each
Tuna spread,	30	2 tablespoons

*Continued on next page*

Table 15 – *Continued from previous page*

FOOD	Wt. (g) E. P. cooked	MEASURE
canned		
b. Meat Products		
Corned beef	40	3 tablespoons
Ham sausage	55	3 of 9cm diameter x 0.3 thick each
c. Bean Products		
Soybean cheese, soft (tofu)	100	$\frac{1}{2}$ cup
Soybean cheese, soft (tokwa)	60	1 (6 x 6 x 2 cm)

The list shows the kinds and amounts of high fat meat and fish to use for one high fat meat and fish exchange:

Table 16: C. High Fat Meat and Fish Exchanges

FOOD	Wt. (g) E. P. cooked	MEASURE
1. Pork		
Ham (pigue)	35	1 slice (3 cm cube)
2. Variety meats/internal organ		
Tongue (dila) - pork, beef	35	$\frac{3}{4}$ cup
3. Egg		
Duck's egg	70	1 piece
Balut	65	1 piece
Penoy	60	1 piece

*Continued on next page*

Table 16 – *Continued from previous page*

FOOD	Wt. (g) E. P. cooked	MEASURE
4. Nuts		
Peanuts, roasted	25	$\frac{1}{3}$ cup
5. Cheese		
Cheese, filled	50	1 slice (6 x 3 2- $\frac{1}{2}$ cm)
Cheese, pimiento flavored	40	1 slice (6 x 3 x 2 cm)
6. Processed foods		
Meat Products		
Frankfurters	60	1- $\frac{1}{2}$ of 12 x 1- $\frac{1}{3}$ cm
Salami	50	3 slices (8 x 8 x 1 cm each)
Vienna sausage	70	4 (5 x 2 x 2 cm)

This list shows the kinds and amounts of fat to use for one fat exchange:

Table 17: Fat Exchange List

FOOD	Wt. (g) E. P. cooked	MEASURE
POLYUNSATURATED FATS/OIL (corn, margarine, soybean		

*Continued on next page*

Table 17 – *Continued from previous page*

FOOD	Wt. (g)	E. P. cooked	MEASURE
rapeseed-canola, rice			
sunflower, safflower and			
sesame	5		1 teaspoon
MONOUNSATURATED			
FATS			
pili nut	5		5 pieces
peanut oil, olive oil	5		1 teaspoon
shortening	5		1 teaspoon

## E. Health Conditions

A persons diet may be different from the usual depending on certain health conditions. There are a lot but the focus will be on the three major health conditions linked with diet, namely, diabetes, hypertension, and kidney stones.

Diabetes is a health condition where the body cannot produce or properly use insulin, a substance that helps the body use sugar (glucose) for energy[25]. The body is incapable of using glucose, letting it stay in the blood stream, resulting in high levels of sugar[25]. There are two main types of diabetes: Type 1 or insulin-dependent which is treated with daily insulin injections, regular exercise, and a balanced meal plan and Type 2 or non-insulin dependent which is treated with an individualized diet plan that usually restricts calories especially from fat, so that a healthy weight can be reached and maintained[26]. In Type 1, the daily meal plan is customized to an individuals needs, which is likely to include 2-3 snacks eaten at set times each day[26]. In Type 2, treatment may include a regular exercise plan and if diet and exercise cannot control blood glucose, oral medication or insulin injections may be needed[26].



Carbohydrate counting is a meal planning method for people with diabetes that allows diabetes to be managed through diet[27]. In monitoring the amount of carbohydrates, there is a better control over blood sugar levels[27]. Typically, 30-45 grams of carbohydrate for females and 45-60 grams of carbohydrate for males are allowed at a meal[27].The table below shows the Daily Calorie Meal Plans (Table 18) for the Type 2 Diabetes[28].

Table 18: Daily Calorie Meal Plans

Calorie	1,200	1,500	1,800	2,000	2,500
Meal Plans (Daily)					
Starch	5	7	8	9	11
Fruit	3	3	4	4	6
Milk	2	2	3	3	3
Vegetables	2	2	3	4	5
Meat and Meat Substitutes	4	4	6	6	8
Fat	3	4	4	5	6

Hypertension is having a high blood pressure[29]. According to the Philippine Society of Hypertension (PSH), more than 10 million Filipinos suffer from hypertension of which half are not aware of their condition, which prompted them to recommend a specialized diet plan for those suffering from abnormal blood pressure[30]. Based on studies conducted in four medical centers in the United States is the Dietary Approach to Stop Hypertension or DASH, in which findings showed that blood pressures were reduced with an eating plan that is low in saturated fat, cholesterol, and total fat that gives emphasis on fruits, vegetables, and fat-free or low-fat milk and milk products[30]. Table 19 shows the DASH Diet[31].

Table 19: Dietary Approach to Stop Hypertension

Food Group	Daily Servings (except as noted)	Serving Sizes	Examples and Notes	Significance of Each Food Group to the DASH Eating Plan
Grains and Grain Products	7-8	1 slice bread 1 oz dry cereal* $\frac{1}{2}$ cup cooked rice, pasta, or cereal	Whole wheat bread, English muffin, pita bread, bagel cereals, grits oatmeal crackers unsalted pretzels and popcorn	Major sources of energy and fiber
Vegetables	4-5	$\frac{1}{2}$ cup cooked vegetable 6oz vegetable juice broccoli, turnip, greens, collards, kale, spinach, artichokes, green beans, lima beans, sweet potatoes	Tomatoes, potatoes, carrots, green peas, squash,	Rich sources of potassium, magnesium and fiber
Fruits	4-5	6 oz fruit juice 1 medium fruit $\frac{1}{4}$ cup dried fruit $\frac{1}{2}$ cup fresh, frozen or canned fruit	Apricots, bananas, dates grapes, oranges orange juice, grapefruit grapefruit juice, mangoes,	Important sources of potassium, magnesium and fiber

*Continued on next page*

Table 19 – *Continued from previous page*

Food Group	Daily Servings (except as noted)	Serving Sizes	Examples and Notes	Significance of Each Food Group to the DASH Eating Plan
			melons, peaches, pineapples, prunes, raisins, strawberries, tangerines	
Low fat or fat free dairyfoods	2-3	8 oz milk 1 cup yogurt 1 $\frac{1}{2}$ oz cheese	Fat free (skim) or low fat (1 % milk, fat free	Major sources of calcium and protein or lowfat buttermilk, fat free or lowfat regular or frozen yogurt, lowfat and fat free cheese
Meats, poultry, and fish	2 or less	3oz cooked meats, poultry, or fish	Select only lean: trim away visible fats: broil, roast or boil, instead of frying; remove skin from poultry	Rich sources of protein and magnesium
Nuts, seeds, and dry beans	4-5 per week	$\frac{1}{3}$ cup or 1 $\frac{1}{2}$ oz nuts 2 Tbsp seeds	Almonds, filberts, mixed or $\frac{1}{2}$ oz walnuts,	Rich sources of energy, nuts, peanut, magnesium,

*Continued on next page*

Table 19 – *Continued from previous page*

Food Group	Daily Servings (except as noted)	Serving Sizes	Examples and Notes	Significance of Each Food Group to the DASH Eating Plan
		$\frac{1}{2}$ cup cooked dry beans	sunflower seeds, kidney beans, lentils, peas	potassium, protein and fiber
Fats and oils **	2-3	1 tsp soft margarine 1 Tbsp lowfat mayonnaise 2 Tbsp light salad dressing 1 tsp vegetable oil	Soft margarine, lowfat mayonnaise, light salad dressing, vegetable oil (such as olive, corn, canola or safflower	DASH has 27 percent of calories as fat, including that in or added to foods
Sweets	5 per week	1 Tbsp sugar 1 Tbsp jelly or jam $\frac{1}{2}$ oz jelly beans 8 oz lemonade	Maple syrup, sugar, jelly, jam, fruit-flavored gelatin, jelly beans, hard candy, fruit punch, sorbet, ices	Sweets should be low in fat

\*Equals  $\frac{1}{2}$ -1  $\frac{1}{4}$  cup, depending on cereal type. Check the products nutrition label.

\*Fat content changes serving counts for fats and oils:

For example, 1 Tbsp of regular salad dressing equals 1 serving:

1 Tbsp of a low fat dressing equals  $\frac{1}{2}$  serving:

1 Tbsp of a fat free dressing equals 0 servings.

Kidney stones form when substances in the urine such as calcium, oxalate, and

phosphorus become highly concentrated[32]. Studies have shown that the Dietary Approaches to Stop Hypertension (DASH), shown in Table 19, can reduce the risk of kidney stones, for it is high in fruits and vegetables, moderate in low-fat dairy products, and low in animal protein[32].

## **F. Web Applications**

A web application is an application utilizing web and [web] browser technologies to accomplish one or more tasks over a network, typically through a [web] browser[33]. Below is the breakdown of web applications[33]:

Application [Software]: Application software, also known as an application or an app, is computer software designed to help the user to perform singular or multiple related specific tasks.

Web technologies: Flash, Silverlight, JavaScript, HTML and CSS, Java, various other programming languages and other computer technology intended for the use across networks.

Network: The internet or intranet.

Browser: Firefox, Chrome, Safari, Opera, Internet Explorer and various others.

Different web application technologies aside, the most confusion as to what qualifies as a web app comes from so many websites out there with varying degrees of usefulness[33]. Not all websites are web apps. If it doesn't perform some sort of task, functioning in some way, it's probably not a web app[33]. A simple web application is compatible on all mobile devices, laptop computers, and desktop computers that have an internet connection.

## **G. Database Management System**

A database management system is a set of software programs that allows users to create, edit, and update data in database files, and store and retrieve data from those database files[34]. Using DBMS, data can be added, deleted, changed, sorted, or searched[34]. Examples include MySQL, PostgreSQL, Microsoft Access, SQL Server, FileMaker, Oracle, RDBMS, dBASE, Clipper, and FoxPro[35].

## IV. Design and Implementation

### A. Context Diagram

Healthy Eating Alert System: Food Intake Monitor (HEAS: FIM) has three user types: Registered User, Content Manager, and Administrator.

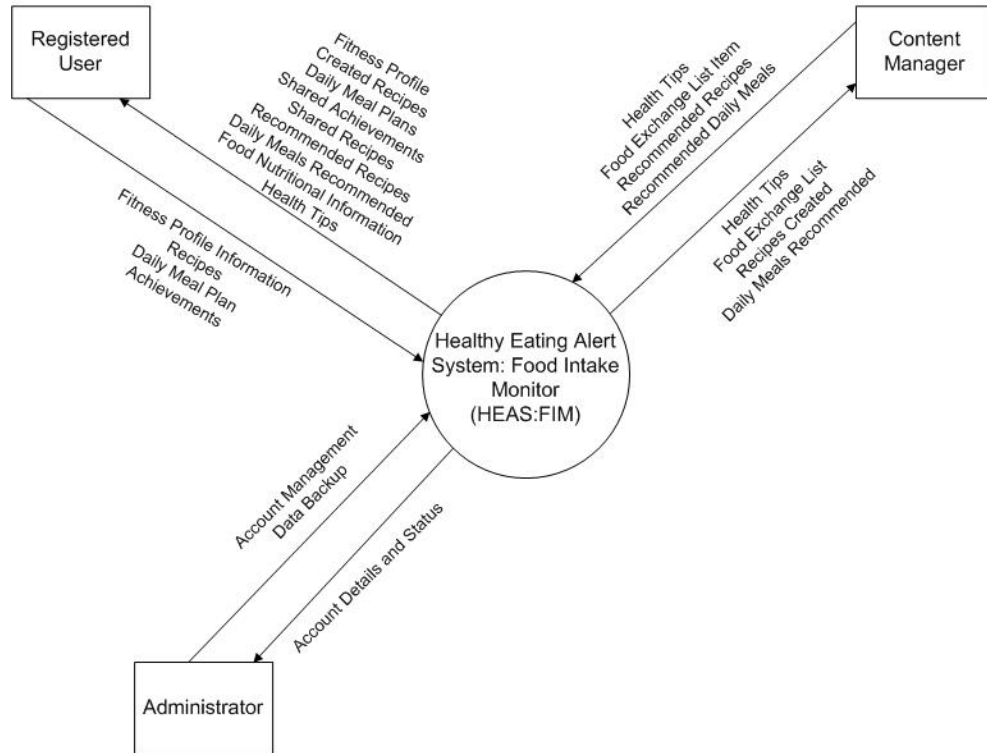


Figure 1: Context Diagram of HEAS:FIM

Figure 1 shows the context diagram of the system. HEAS:FIM system interaction with the three external actors is shown. The Registered User is responsible for his/her own profile. He/she will add fitness profile information to his/her own fitness profile. He/she could also create his/her own recipes with corresponding calorific values and consideration of health conditions. He/she can then make daily meal plans with the recipes he/she created. Achievement can also be shared on the system. The Content Manager is responsible for the health tips that would be seen by the registered users, entries in the food exchange list coming from reliable sources such as FNRI, and recommended recipes and daily meals with corresponding calorific values and consideration of health conditions that the reg-

istered users can browse. Lastly, the Administrator is responsible for account management, with functionalities such as creating another administrator or content manager, activation/deactivation of accounts and editing of account details, and data backup.

## B. Use Cases

### B.1 Top Level Use Case Diagram

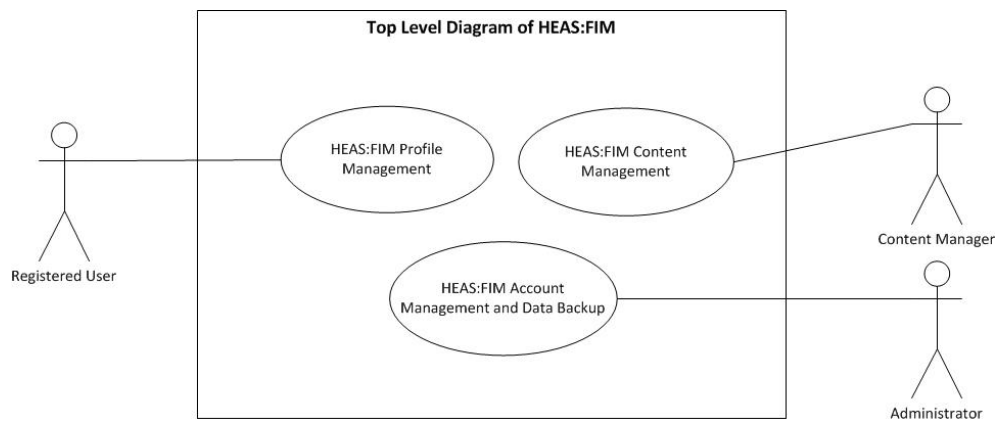


Figure 2: Top Level Use Case Diagram of HEAS:FIM

Figure 2 shows the three actors in the HEAS:FIM: the Registered User, the Content Manager, and the Administrator. The main role of the Registered User shall be to fill out and manage own profile. The content manager shall provide content for some recommended recipes, daily meals, and health tips. The Administrator shall manage accounts of all actors of the HEAS:FIM and shall backup system data.



## B..2 Profile Management

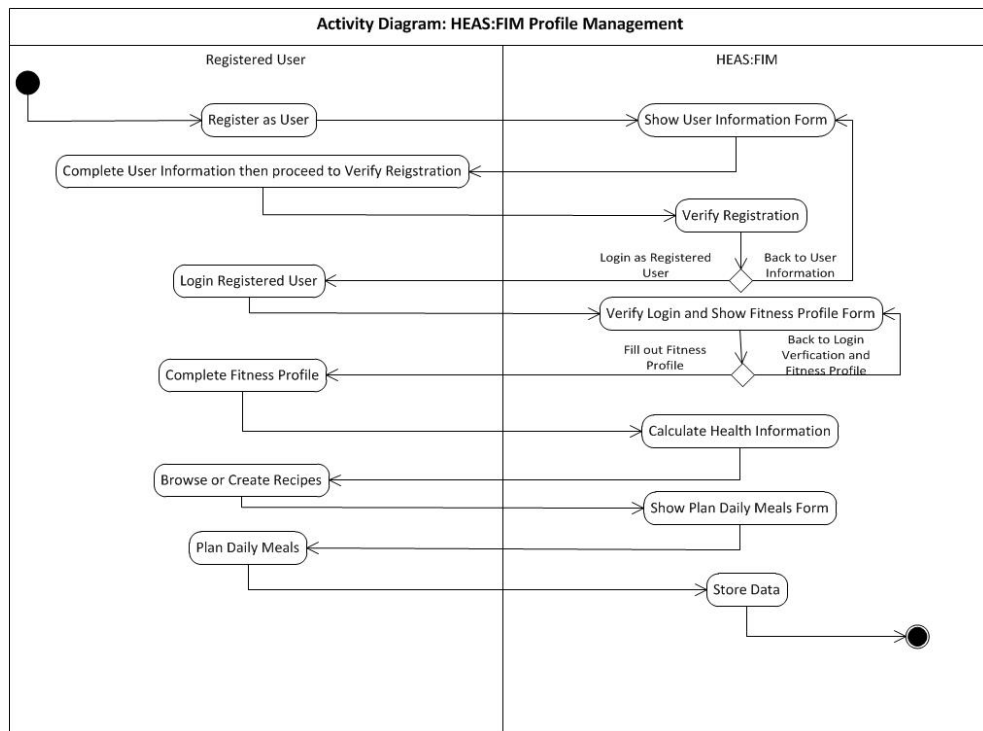


Figure 3: Activity Diagram: HEAS:FIM Profile Management

Figure 3 shows the Activity Diagram of the Registered User Profile Management. The Registered User, before becoming one, shall first register as a user. The User Information Form will be shown. The form shall be filled out completely, and then the system will verify the registration. If successful, the user may login. Otherwise, there would be an alert to fill out the User Information Form correctly and completely. Next, the Fitness Profile Form will be shown. If the form is filled out, Health Information such as BMI, Calorie Need, and Restrictions will be calculated. The Registered User can then Browse Recipes, Create Recipes, and finally, Plan Daily Meals.

### B..3 Content Management

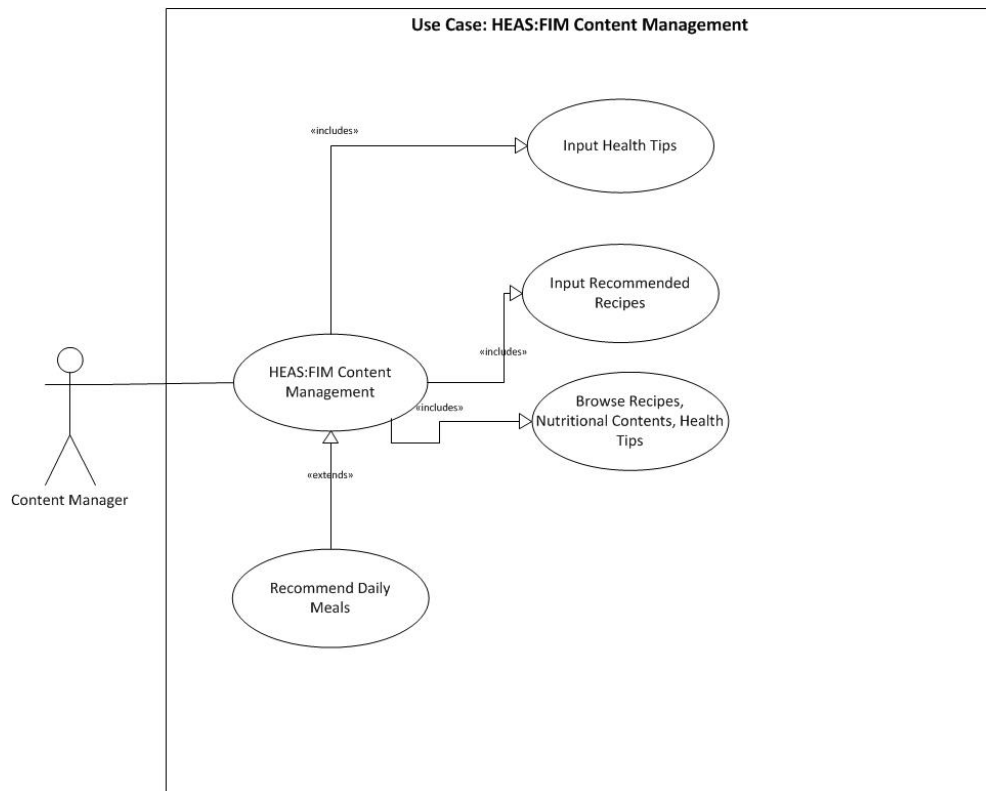


Figure 4: Use Case (HEAS:FIM Content Management)

Figure 4 shows the sub explosion of the HEAS:FIM Content Management. The Content Manager shall input health tips and recommended recipes. The data will then be stored in the database as shown in Figure 5. After which, he/she can browse recipes, nutritional contents, and health tips to choose which to share as shown in Figure 6. Then, the system will recommend daily meals.

## B..4 Input Health Tips

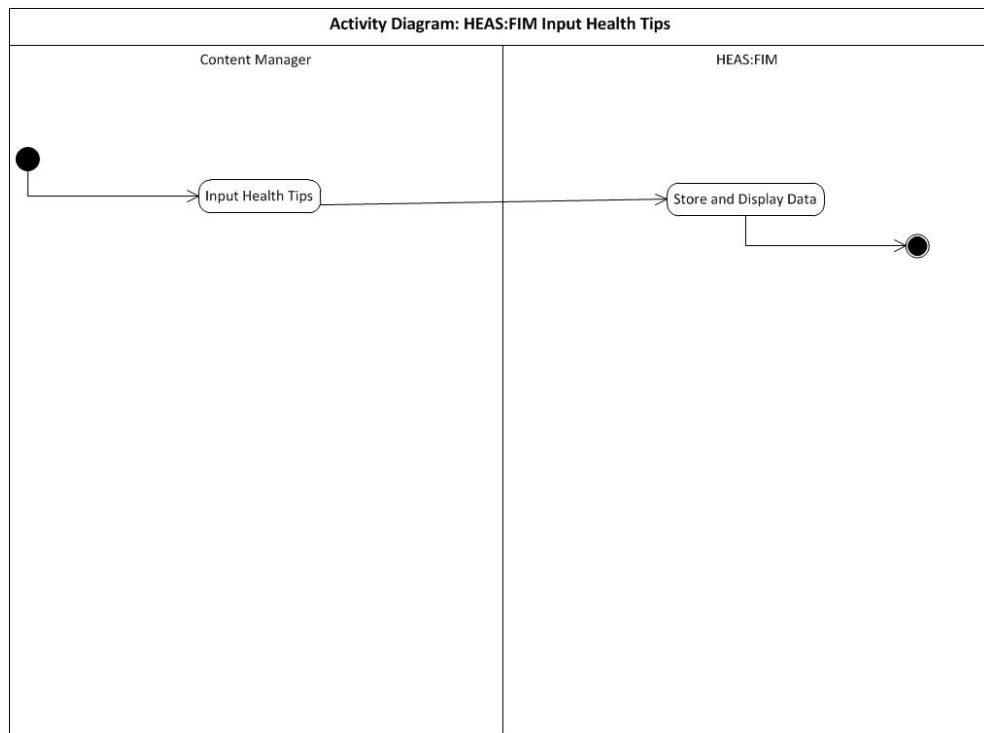


Figure 5: Activity Diagram: HEAS:FIM Input Health Tips

## B..5 Browse Recipes, Food Nutritional Contents, Health Tips

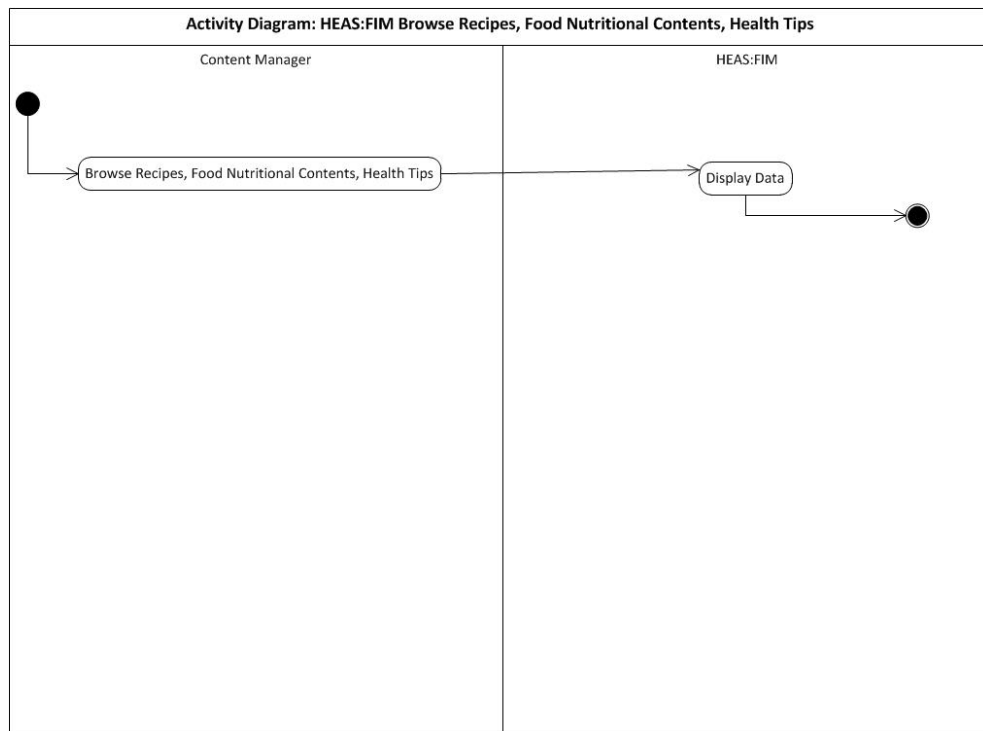


Figure 6: Activity Diagram: HEAS:FIM Browse Recipes, Food Nutritional Contents, Health Tips

## B..6 Account Management

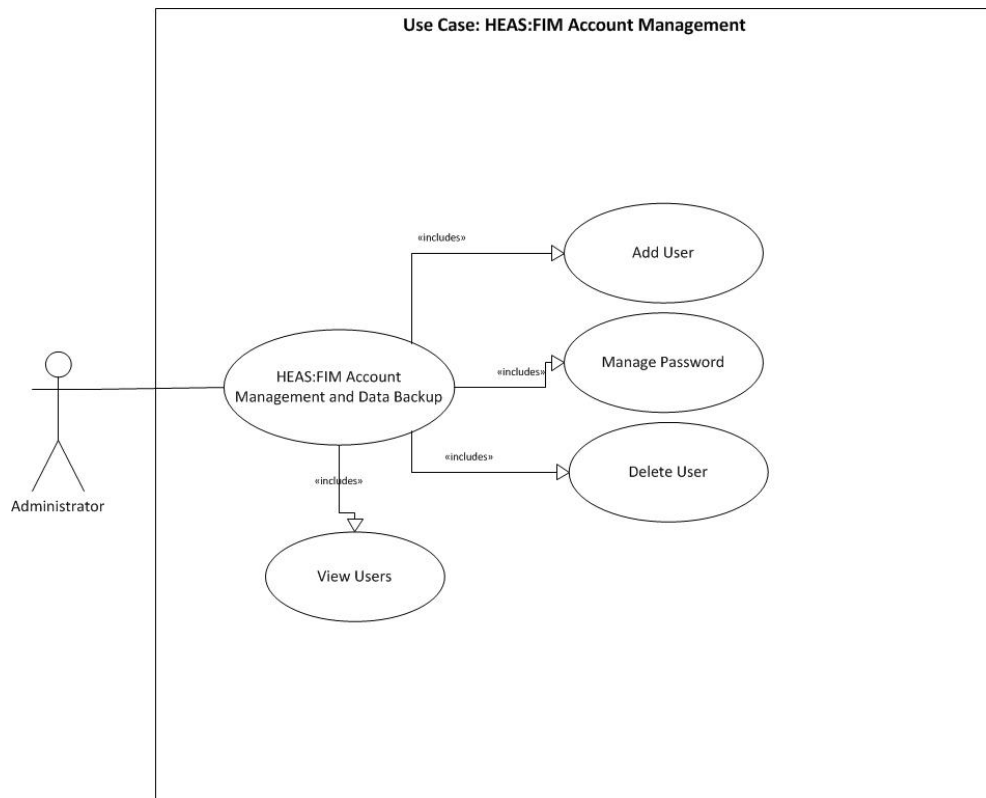


Figure 7: Use Case (HEAS:FIM Account Management)

Figure 7 shows that the Administrator shall manage accounts. He/she can manually add user (Figure 8), view users (Figure 9), delete user (Figure 10), and manage password (Figure 11).

## B..7 Add User

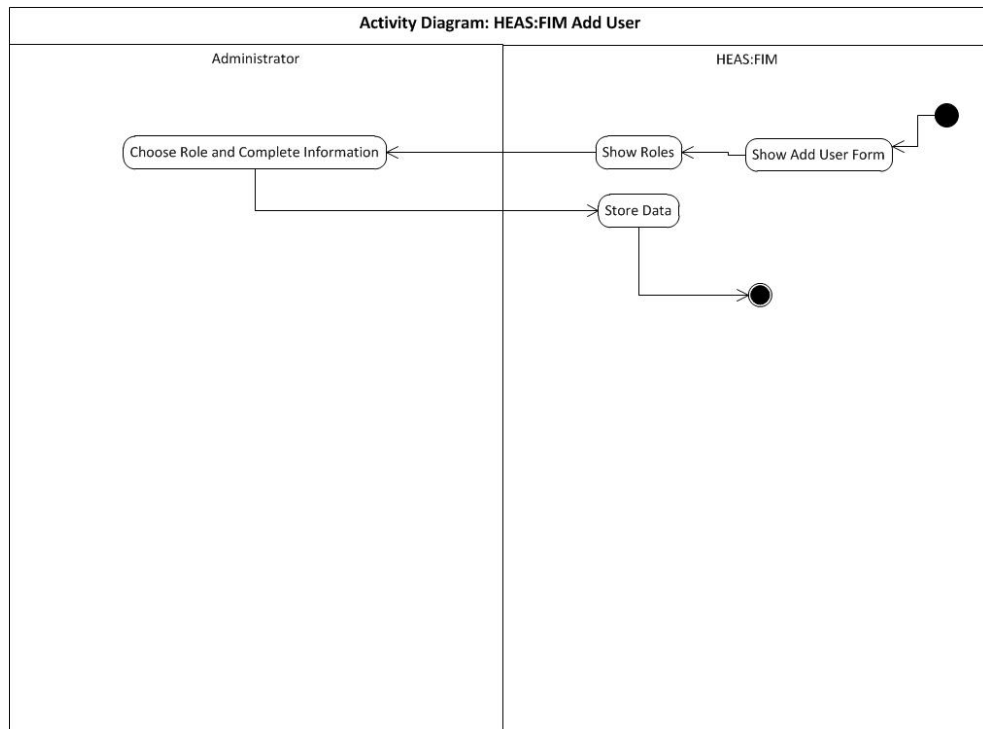


Figure 8: Activity Diagram: HEAS:FIM Add User

The Administrator can add users particularly Content Managers by filling out and completing the Add User Form. The data will be stored in the database after submitting the form.

## B..8 View Users

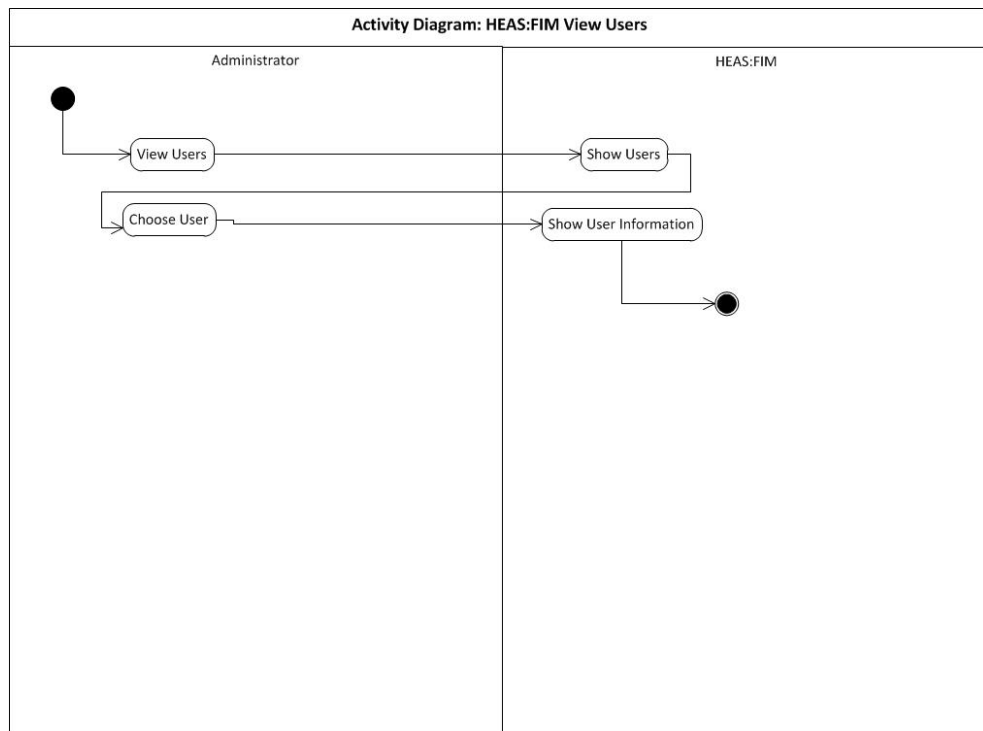


Figure 9: Activity Diagram: HEAS:FIM View Users

The Administrator can view users by selecting Show Users. A particular user can be viewed by selecting Choose User. The User Information will be shown.

## B..9 Delete User

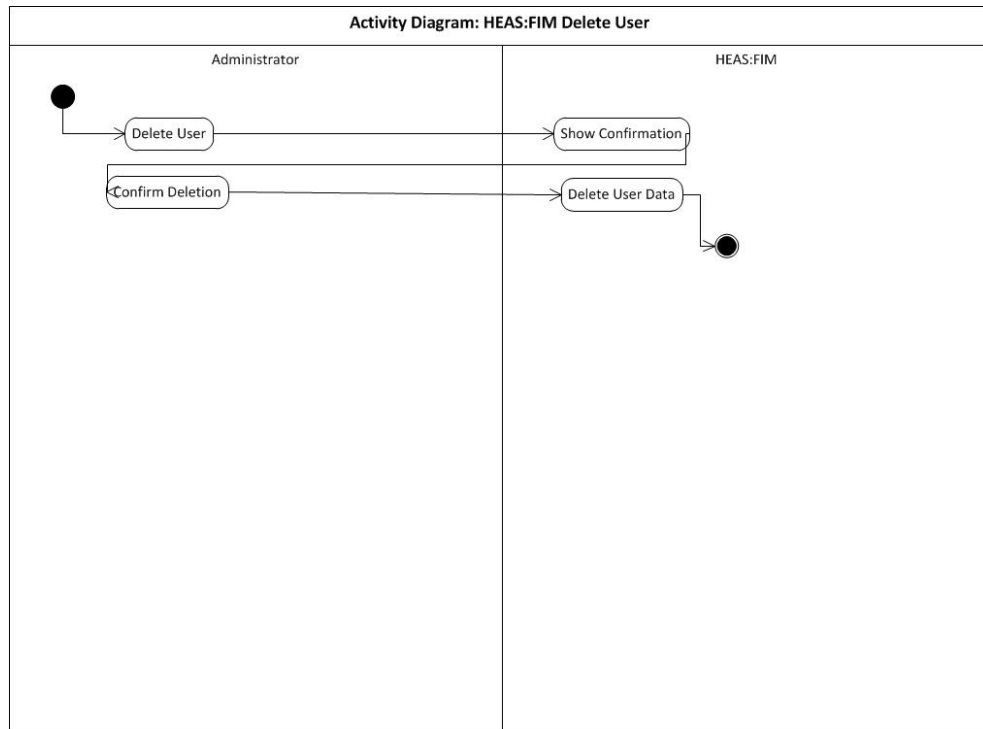


Figure 10: Activity Diagram: HEAS:FIM Delete User

The Administrator, when necessary, can delete a particular user by first selecting the user (Figure 9). After clicking the Delete Button, a confirmation message whether to continue or not the deletion will be shown. When chosen to continue, User Data will be deleted.



## B..10 Manage Password

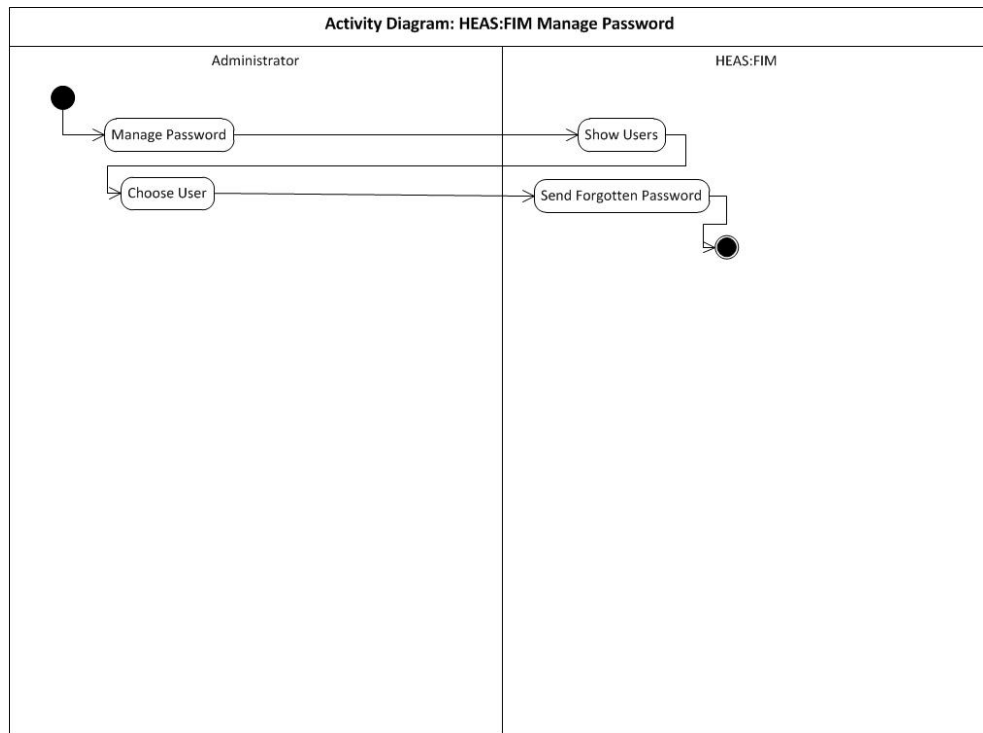


Figure 11: Activity Diagram: HEAS:FIM Manage Password

The Administrator can manage password by first choosing the user then by sending a forgotten password link to the e-mail address linked to the account.

## B..11 Plan Daily Meals

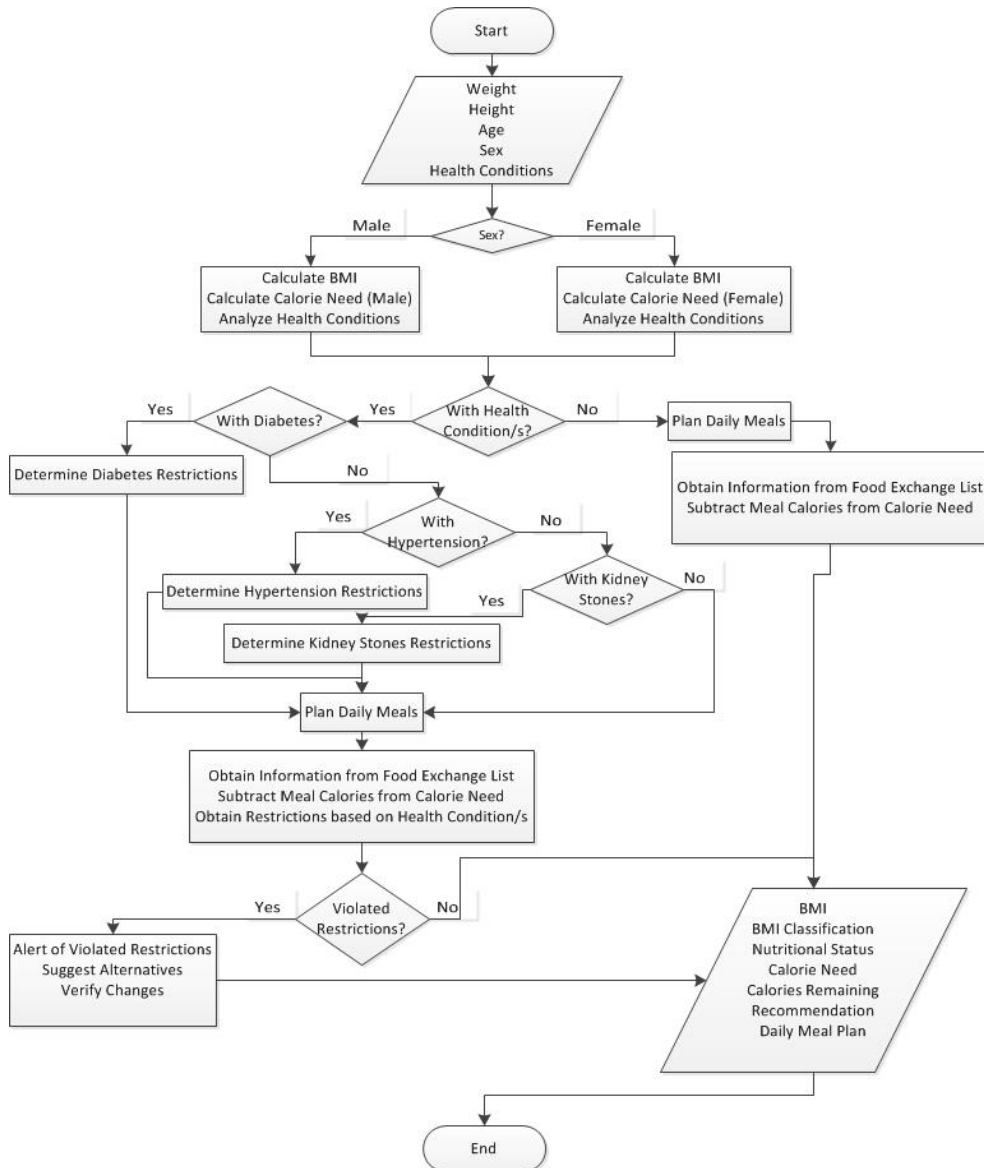


Figure 12: Flow Chart: HEAS:FIM Plan Daily Meals

## C. Database Structure

### C.1 Entity Relationship Diagram

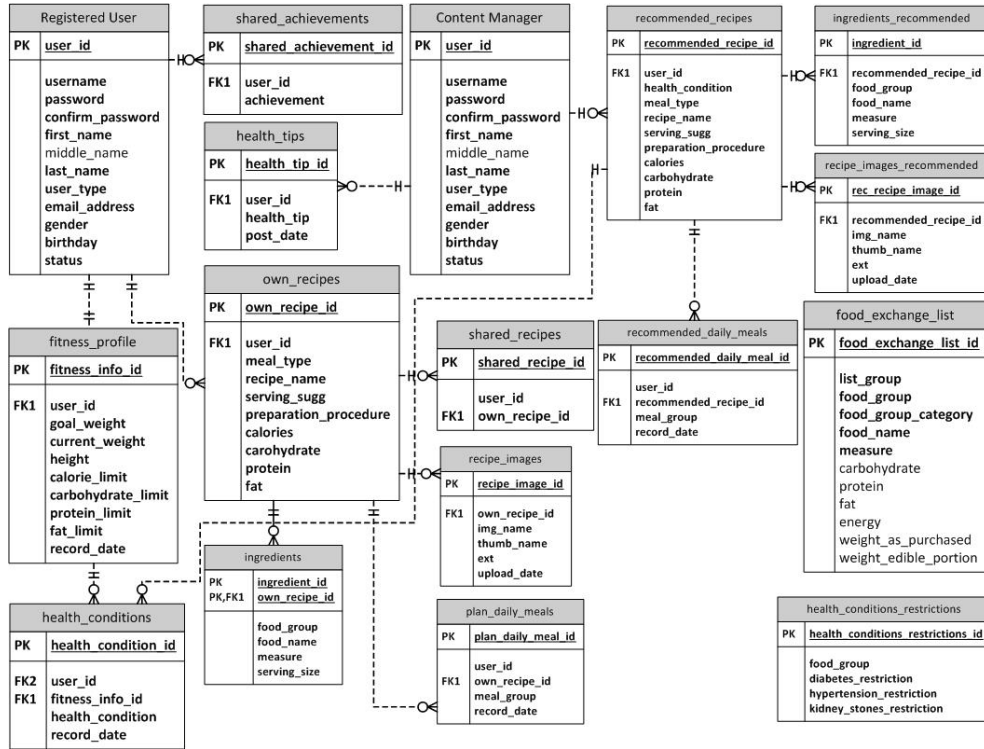


Figure 13: HEAS:FIM Entity Relationship Diagram

Figure 13 shows the Entity Relationship Diagram of the HEAS:FIM. Each registered user has his/her own user information. He/she can have one and only one fitness profile (fitness\_profile table). In a fitness profile, there can be more than one health condition linked to the profile owner (health\_conditions table). He/she can create (own\_recipes table) or share more than one recipe (shared\_recipes table) and share achievements (shared\_achievements table) such as achieved goal weight. Every created and shared recipe is linked to health conditions, if exists. The Content Manager can create and share health tips (health\_tips table) and recommend recipes customized with particular health conditions. The food exchange list (food\_exchange\_list table) contains the composition and exchanges equivalent of food exchanges. The health\_conditions\_restrictions table contain certain restrictions for Diabetes, Hypertension, and Kidney Stones.

## C..2 Data Dictionary

Table 20: users table

Attribute	Data Type	Description
<b>user_id</b>	int (20)	Primary Key, User ID
username	varchar (20)	User Username
password	varchar (50)	User Password
confirm_password	varchar (50)	User Confirm Password
first_name	varchar (20)	User First Name
middle_name	varchar (20)	User Middle Name (optional)
last_name	varchar (20)	User Last Name
user_type	enum ('Administrator', 'Content Manager', 'User')	User User Type
email_address	varchar (50)	User E-mail Address
gender	enum ('Male', 'Female')	User Gender
birthday	date	User Date of Birth
status	enum ('Activated', 'Deactivated')	User Account Status

Table 21: fitness\_profile table

Attribute	Data Type	Description
<b>fitness_info_id</b>	int (20)	Primary Key, Fitness Info ID
user_id	int (20)	Foreign Key, References users (user_id), User ID
goal_weight	float	Stores the Goal Weight entered by the Registered User
current_weight	float	Stores the Current Weight entered by the Registered User
height	float	Stores the Height entered by the Registered User
calorie_limit	float	Daily Calorie Limit of the Registered User in Kcal
carbohydrate_limit	float	Daily Carbohydrate Limit of the Registered User in g
protein_limit	float	Daily Protein Limit of the Registered User in g
fat_limit	float	Daily Fat Limit of the Registered User in g
record_date	date	Date when the record was entered

Table 22: health\_conditions table

Attribute	Data Type	Description
<b>health_condition_id</b>	int (20)	Primary Key, Health Condition ID
user_id	int (20)	Foreign Key, References users (user_id), User ID
fitness_info_id	int (20)	Foreign Key, References fitness_profile (fitness_info_id), Fitness Info ID
health_condition	set ('Diabetes', 'Hypertension', 'Kidney Stones', 'None')	Health Condition
record_date	date	Date when the record was entered

Table 23: own\_recipes table

Attribute	Data Type	Description
<b>own_recipe_id</b>	int (20)	Primary Key, Own Recipe ID
user_id	int (20)	Foreign Key, References users (user_id), User ID
health_condition	set ('Diabetes', 'Hypertension', 'Kidney Stones', 'None')	Health Condition
meal_type	enum ('Breakfast', 'Lunch', 'Dinner', 'Snack')	Meal Type
recipe_name	varchar (50)	Recipe Name
serving_sugg	float	Recipe Serving Size
preparation_procedure	longtext	Recipe Preparation Procedure
calories	float	Recipe Calories in Kcal
carbohydrate	float	Recipe Carbohydrate in g
protein	float	Recipe Protein in g
fat	float	Recipe Fat in g

Table 24: recommended\_recipes table

Attribute	Data Type	Description
<b>recommended_recipe_id</b>	int (20)	Primary Key, Recommended Recipe ID
user_id	int (20)	Foreign Key, References users (user_id), User ID
health_condition	set ('Diabetes','Hypertension', 'Kidney Stones','None')	Health Condition
meal_type	enum ('Breakfast', 'Lunch', 'Dinner', 'Snack')	Meal Type
recipe_name	varchar (50)	Recipe Name
serving_sugg	float	
preparation_procedure	longtext	Recipe Preparation Procedure
calories	float	Recipe Calories in Kcal
carbohydrate	float	Recipe Carbohydrate in g
protein	float	Recipe Protein in g
fat	float	Recipe Fat in g

Table 25: shared\_recipes table

Attribute	Data Type	Description
<b>shared_recipe_id</b>	int (20)	Primary Key, Shared Recipe ID
user_id	int (20)	Foreign Key, References users (user_id), User ID
own_recipe_id	int (20)	Foreign Key, References own_recipes (own_recipe_id), Own Recipe ID

Table 26: recipe\_images table

Attribute	Data Type	Description
<b>recipe_image_id</b>	int (20)	Primary Key, Recipe Image ID
own_recipe_id	int (20)	Foreign Key, References own_recipes (own_recipe_id), Own Recipe ID
img_name	varchar (40)	Image Name
thumb_name	varchar (40)	Image Thumbnail Name
ext	varchar (8)	Image File Extension
upload_date	varchar (20)	Image Upload Date

Table 27: recipe\_images\_recommended table

Attribute	Data Type	Description
<b>recipe_image_id</b>	int (20)	Primary Key, Recipe Image ID
recommended_recipe_id	int (20)	Foreign Key, References recommended_recipes (recommended_recipe_id), Recommended Recipe ID
img_name	varchar (40)	Image Name
thumb_name	varchar (40)	Image Thumbnail Name
ext	varchar (8)	Image File Extension
upload_date	varchar (20)	Image Upload Date

Table 28: plan\_daily\_meals table

Attribute	Data Type	Description
<b>plan_daily_meal_id</b>	int (20)	Primary Key, Plan Daily Meal ID
user_id	int (20)	Foreign Key, References users (user_id), User ID
own_recipe_id	int (20)	Foreign Key, References own_recipes (own_recipe_id), Own Recipe ID
meal_group	enum ('Breakfast', 'Lunch', 'Dinner', 'Snack')	Meal Group
record_date	date	Date when the record was entered



Table 29: recommended\_daily\_meals table

Attribute	Data Type	Description
<b>recommended_daily_meal_id</b>	int (20)	Primary Key, Recommended Daily Meal ID
user_id	int (20)	Foreign Key, References users (user_id), User ID
recommended_recipe_id	int (20)	Foreign Key, References recommended_ recipes (recommended_ recipe_id), Recommended Recipe ID
meal_group	enum ('Breakfast', 'Lunch', 'Dinner', 'Snack')	Meal Group
record_date	date	Date when the record was entered

Table 30: shared\_achievements table

Attribute	Data Type	Description
<b>shared_achievement_id</b>	int (20)	Primary Key, Shared Achievement ID
user_id	int (20)	Foreign Key, References users (user_id), User ID
achievement	longtext	Achievement

Table 31: health\_tips table

Attribute	Data Type	Description
<b>health_tip_id</b>	int (20)	Primary Key, Health Tip ID
user_id	int (20)	Foreign Key, References users (user_id), User ID
health_tip post_date	longtext date	Health Tip Date when the record was posted

Table 32: ingredients table

Attribute	Data Type	Description
<b>ingredient_id</b>	int (20)	Primary Key, Ingredient ID
own_recipe_id	int (20)	Foreign Key, References own_recipes (own_recipe_id), Own Recipe ID
food_group	varchar (50)	Food Group
food_name	varchar (50)	Food Name
measure	varchar (50)	Measure with Unit
serving_size	float	Serving Size

Table 33: ingredients\_recommended table

Attribute	Data Type	Description
<b>ingredient_id</b>	int (20)	Primary Key, Ingredient ID
recommended_recipe_id	int (20)	Foreign Key, References recommended_ recipes (recommended_ recipe_id), Recommended Recipe ID
food_group	varchar (50)	Food Group
food_name	varchar (50)	Food Name
measure	varchar (50)	Measure with Unit
serving_size	float	Serving Size

Table 34: food\_exchange\_list table

Attribute	Data Type	Description
<b>food_exchange_list_id</b>	int (20)	Primary Key, Food Exchange List ID
list_group	varchar (50) ('I-A', 'I-B', 'I-P', 'II', 'III', 'IV', 'V', 'VI', 'Other')	List Group
food_group	varchar (50) ('Vegetable', 'Fruit', 'Milk', 'Rice', 'Meat', 'Fat', 'Other')	Food Group
food_group_category	varchar (50) ('Vegetable', 'Fruit', 'Milk', 'Rice', 'Meat', 'Fat', 'Other')	Food Group + Category
food_name	varchar (50)	Food Name
measure	varchar (50)	Measure with Unit
carbohydrate	float	Carbohydrate in g
protein	float	Protein in g
fat	float	Fat in g
energy	float	Calories, in Kcal
weight_as_purchased	float	Weight in g
weight_edible_portion	float	Weight in g

Table 35: health\_conditions\_restrictions table

Attribute	Data Type	Description
<b>health_conditions_restrictions_id</b>	int (20)	Primary Key, Health Conditions Restrictions ID
food_group	varchar (50)	Food Group
diabetes_restriction	float	Restriction Quantity by Food Group
hypertension_restriction	float	Restriction Quantity by Food Group
kidney_stones_restriction	float	Restriction Quantity by Food Group

## V. Architecture

### A. System Architecture

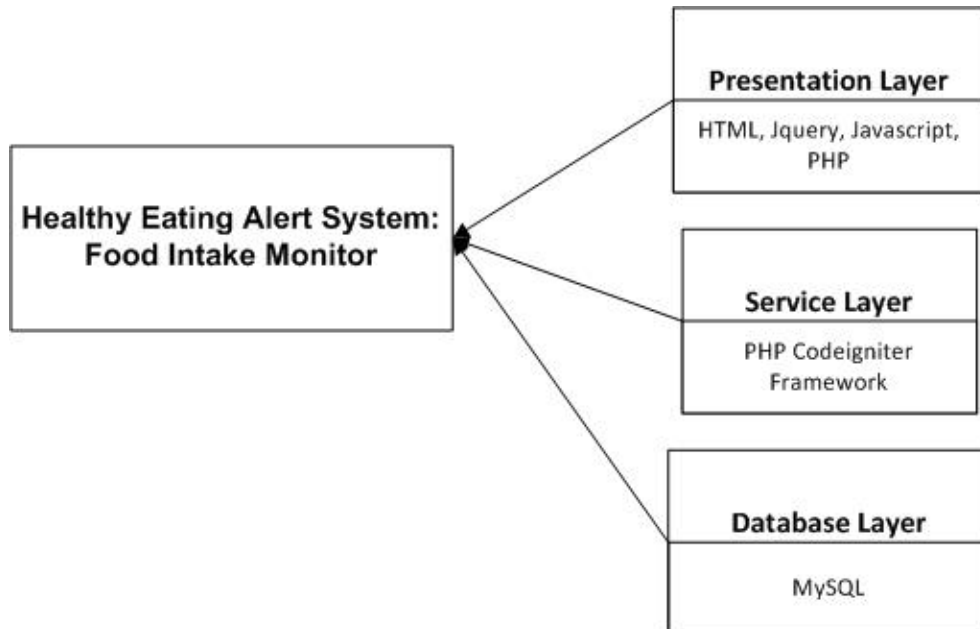


Figure 14: HEAS:FIM System Architecture

Figure 14 shows the three layers of HEAS:FIM. The Presentation Layer contains the web user interface encoded in HTML, JQuery, Javascript, and PHP. The Service Layer is linked to the Presentation Layer through the Codeigniter Framework which is based on the MVC(Model-View-Controller) development pattern and uses the PHP language. The Database Layer is linked to the Service Layer in which MySQL is used by HEAS:FIM as its main database.

### B. Technical Architecture

The minimum requirements for Healthy Eating Alert System: Food Intake Monitor:

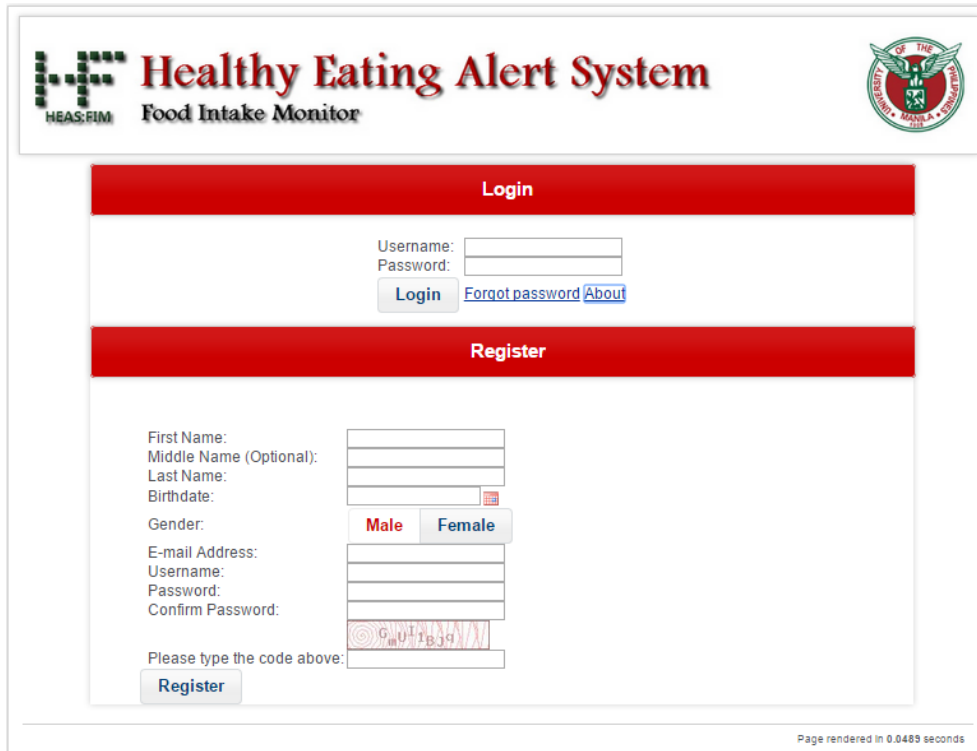
- Web Server: Apache
- Database Server: MySQL
- MySQL Administration Tool: phpMyAdmin
- PHP Framework: CodeIgniter

- Web Browser: best on Google Chrome
- Stable Internet Connection
- 1.4 GHz Solo Core
- 1 GB RAM

## VI. Results

### A. General View

#### A..1 HEAS:FIM Home Page




The screenshot displays the HEAS:FIM Home Page. At the top left, there is a logo for HEAS:FIM and the text "Healthy Eating Alert System Food Intake Monitor". At the top right, there is a circular logo of the University of the Philippines Manila. The page is divided into two main sections: "Login" and "Register".

**Login Section:**

- Username:
- Password:
- Buttons: [Login](#), [Forgot password](#), [About](#)

**Register Section:**

- First Name:
- Middle Name (Optional):
- Last Name:
- Birthdate:
- Gender:  Male  Female
- E-mail Address:
- Username:
- Password:
- Confirm Password:
- Image: 
- Please type the code above:
- Button: [Register](#)

Page rendered in 0.0489 seconds

Figure 15: HEAS:FIM Home Page

Figure 15 shows the home page of Healthy Eating Alert System: Food Intake Monitor. The administrator, content manager, and user can login to their accounts through this page. If a user does not have an account yet, he/she may register also through this page. A content manager account is created by the administrator. In case of forgotten password, there is a Forgot password link.

## A..2 Profile

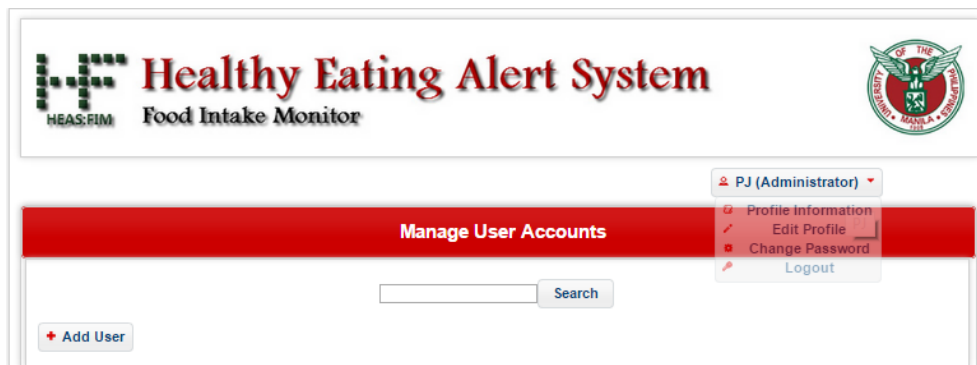


Figure 16: HEAS:FIM Profile

Figure 16 shows the common button for the administrator, content manager, and user. It displays the name and user type of the one logged in. Profile information can be viewed and edited. Password can be changed. It contains the logout link.

## B. Administrator View

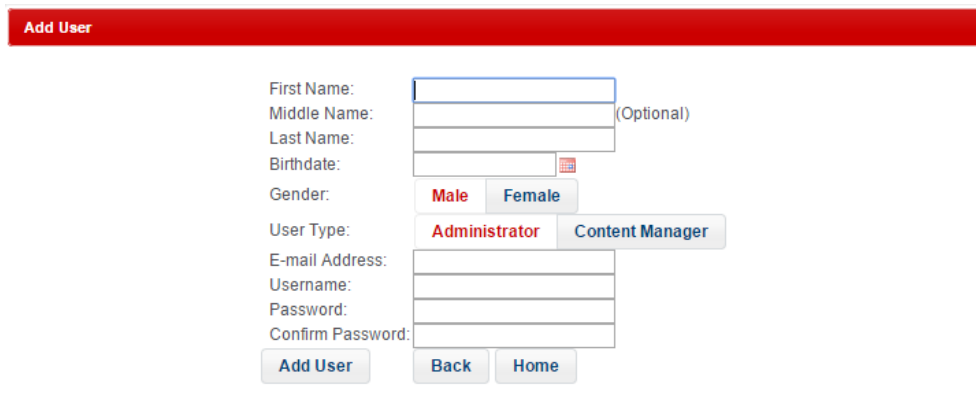
### B..1 Administrator Home Page



Figure 17: HEAS:FIM Administrator Home Page

Figure 17 shows the administrator home page. Administrative functions can be accessed through the buttons. He/She can manage accounts.

## B..2 Add User



First Name:

Middle Name:  (Optional)

Last Name:

Birthdate:

Gender:  Male  Female

User Type:  Administrator  Content Manager

E-mail Address:

Username:

Password:

Confirm Password:

Figure 18: Add User

By clicking the Add User button on the administrator home page, the administrator can create an account for administrator or content manager. (Figure 18).

## B.3 View Users



HEAS:FIM **Healthy Eating Alert System**  
Food Intake Monitor

PJ (Administrator)

Manage User Accounts

Search

+ Add User

Username	First Name	Middle Name	Last Name	User Type	E-mail Address	Gender	Birthday	Age	Actions	Status
rtd23	Rolando	Taganayon	Dizon	User	rtd23@gmail.com	Male	1950-May-23	64		Activated
eud06	Edith	Udarbe	Dizon	User	eudizon06@gmail.com	Female	1961-Jan-06	54		Activated
pmud18	Pat	Udarbe	Dizon	Content Manager	patdizon2009@yahoo.com	Female	1994-Jul-18	20		Activated
pjud31	PJ	Udarbe	Dizon	Administrator	preciousjoydizon@yahoo.com	Female	1992-Jul-31	22		Activated

Figure 19: View Users

By clicking the Search button on the administrator home page, the administrator can view, edit, activate/deactivate, or delete existing accounts. (Figure 19).



## C. Content Manager View

### C..1 Content Manager Home Page



Figure 20: HEAS:FIM Content Manager Home Page

Figure 20 shows the content manager home page. Content Manager functions can be accessed through the buttons. He/She can add content (Food Exchange Item, Recommended Recipe, Recommended Daily Meal Plan, Health Tip), view the content/s he/she added, and delete content/s (Food Exchange Item, Recommended Recipe, Recommended Daily Meal Plan, Health Tip).

### C..2 View, Edit, or Delete Food Exchange List Item/s






List Group	Food Group	Food Group Category	Food Name	Measure	Carbohydrate	Protein	Fat	Energy	Weight as Purchased	Weight Edible Portion	Actions
I-A	Vegetable	Vegetable, Vegetable A	Acelgas	1 cup raw or 1/2 cup cooked	0	0	0	0	25	45	 
I-A	Vegetable	Vegetable, Vegetable A	Alagav Leaves	1 cup raw or 1/2 cup cooked	0	0	0	0	25	45	 
I-A	Vegetable	Vegetable, Vegetable A	Alugbati Leaves	1 cup raw or 1/2 cup cooked	0	0	0	0	25	45	 

Figure 21: View, Edit, or Delete Food Exchange List Item/s

By clicking the Search button for Food Exchange List on the content manager home page, the content manager can view, edit, or delete Food Exchange List item/s. (Figure 21).

### C..3 View, Edit, or Delete Health Tip/s

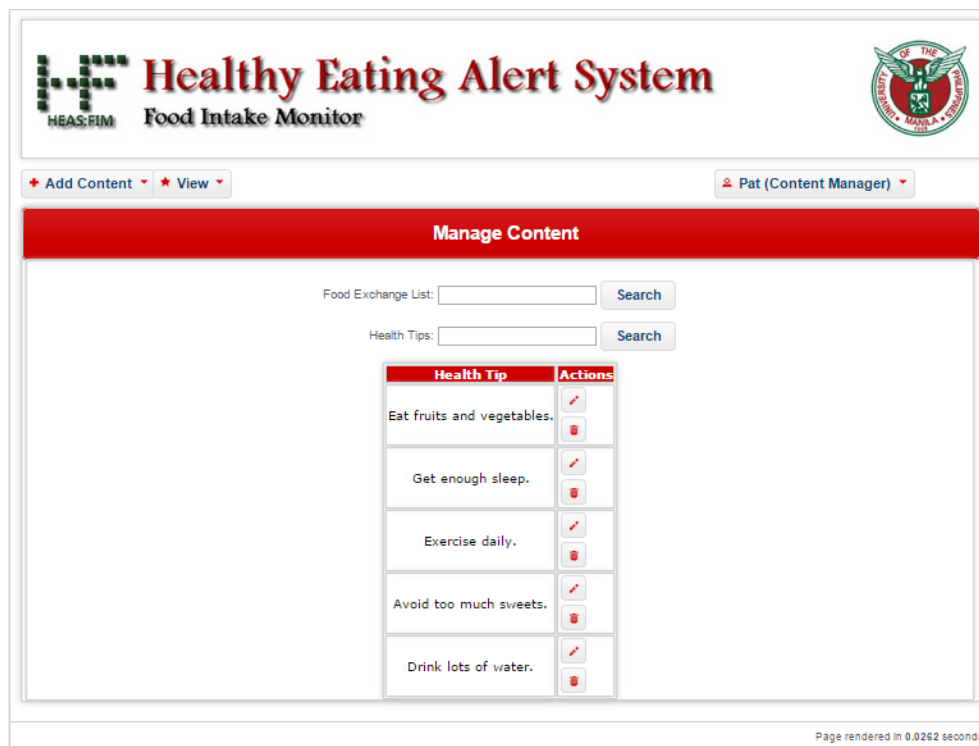


Figure 22: View, Edit, or Delete Health Tip/s

By clicking the Search button for Health Tips on the content manager home page, the content manager can view, edit, or delete Health Tip/s. (Figure 22).

## C..4 Add Content



Figure 23: Add Content

By clicking the Add Content button, links to add Food Exchange Item, Recommended Recipe, Recommended Daily Meal Plan, and Health Tip can be accessed. (Figure 23).

## C..5 Add Food Exchange List Item

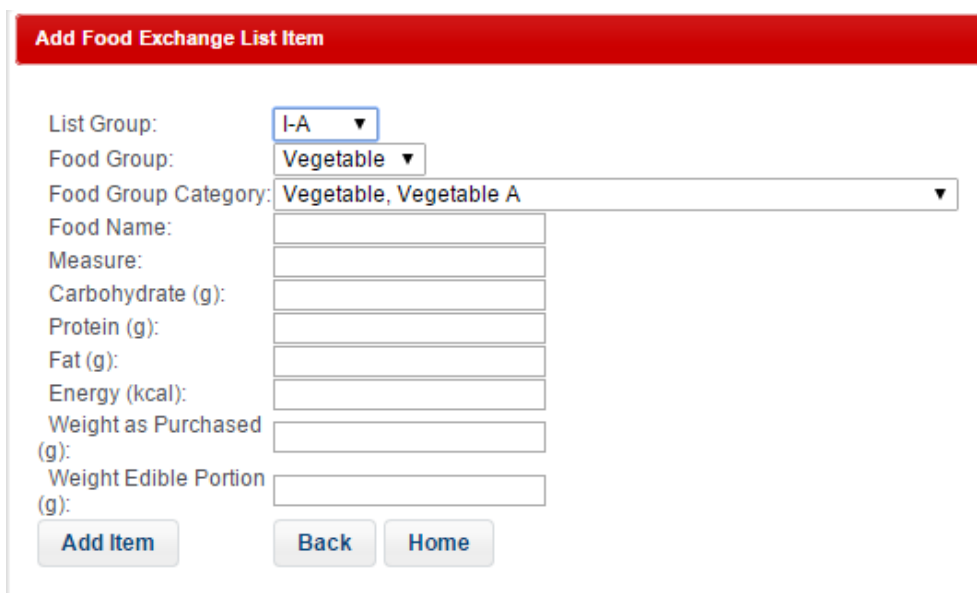


Figure 24: Add Food Exchange List Item

By selecting the Food Exchange Item under Add Content, the content manager can add item/s to the Food Exchange List. (Figure 24).

## C..6 Create Recipe

The screenshot shows a web form titled "Create Recipe" with a red header. The form includes the following fields and controls:

- Health Condition:** A label followed by a "Guidelines for Health Conditions" link and four radio button options: Diabetes, Hypertension, Kidney Stones, and None.
- Meal Type:** A dropdown menu currently set to "Breakfast".
- Recipe Image:** A "Choose File" button and the text "No file chosen".
- Recipe Name:** A text input field.
- Serving Size:** A text input field with "+" and "-" buttons below it.
- Ingredients:** A label followed by four stacked text input fields for "Ingredient", "Food Group", "Measure (g)", and "Quantity".
- Procedure:** A text area with a small diagonal icon in the bottom right corner.
- Nutritional Information:** Four stacked text input fields for "Total Calories (Kcal)", "Total Carbohydrates (g)", "Total Protein (g)", and "Total Fat (g)".
- Buttons:** A "Create Recipe" button on the left and "Back" and "Home" buttons on the right.

Figure 25: Create Recipe

By selecting the Recipe under Add Content, the content manager can create recipe/s that would be recommended to the users. (Figure 25).

## C..7 Recommend Daily Meals

The screenshot shows a web form titled "Recommend Daily Meals" with a red header. The form includes the following elements:

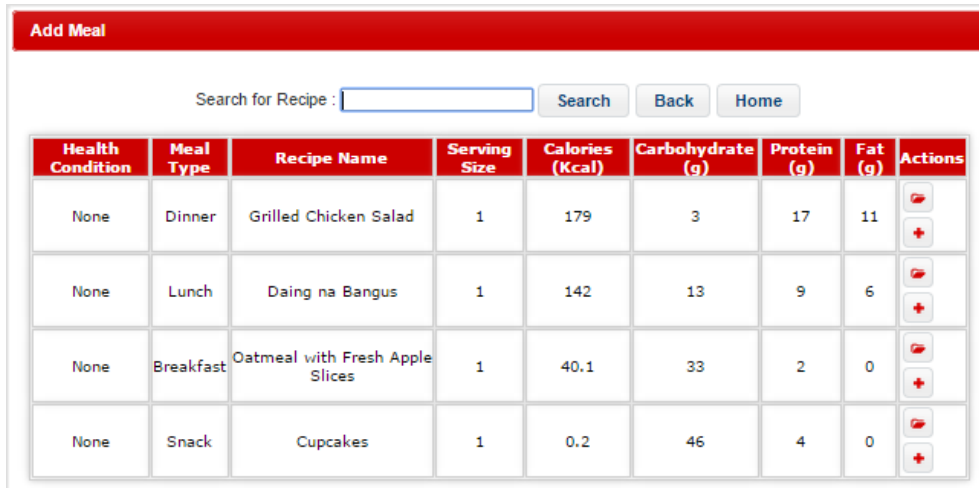
- Buttons:** "Back" and "Home" buttons at the top.
- Search Section:** A red bar containing "Add Meals to Recommended Daily Meals" and "Search Meal" buttons.
- Form Fields:** A "Search Recommended Daily Meals" label, a "Date:" label, a text input field, a calendar icon, and a "Go" button.

Figure 26: Recommend Daily Meals

By selecting the Daily Meal Plan under Add Content, the content manager can

add meals that would be recommended to the users by clicking Search Meal or view or delete added meals by choosing a date and clicking Go. (Figure 26).

### C..8 Add Meal



The screenshot shows the 'Add Meal' interface. At the top, there is a red header bar with the text 'Add Meal'. Below the header, there is a search bar with the placeholder text 'Search for Recipe :', followed by 'Search', 'Back', and 'Home' buttons. Below the search bar is a table with the following data:









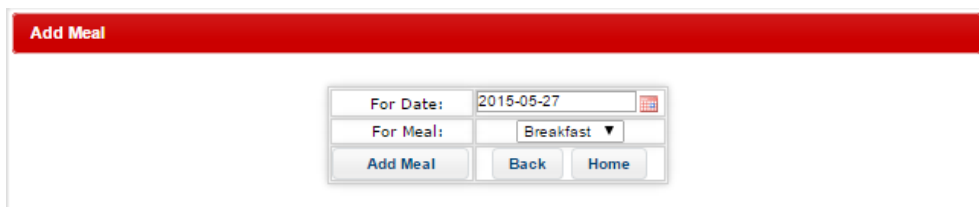

Health Condition	Meal Type	Recipe Name	Serving Size	Calories (Kcal)	Carbohydrate (g)	Protein (g)	Fat (g)	Actions
None	Dinner	Grilled Chicken Salad	1	179	3	17	11	 
None	Lunch	Daing na Bangus	1	142	13	9	6	 
None	Breakfast	Oatmeal with Fresh Apple Slices	1	40.1	33	2	0	 
None	Snack	Cupcakes	1	0.2	46	4	0	 

Figure 27: Add Meal



The screenshot shows the 'Add Meal' interface. At the top, there is a red header bar with the text 'Add Meal'. Below the header, there is a form with the following fields and buttons:

For Date:  


For Meal:  

Figure 28: Add Recipe Meal



Figure 29: View Recipe Meal

(Figure 26). The content manager can add the recipe meal to a chosen date (Figure 28) or view the recipe first (Figure 29).

### C..9 View or Delete Meal/s

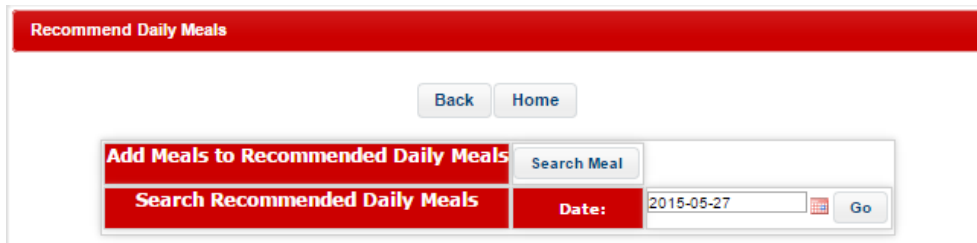


Figure 30: View Meal/s

Recommend Daily Meals

Back Home

Add Meals to Recommended Daily Meals Search Meal

Search Recommended Daily Meals Date: 2015-05-27 Go

Recipes

Health Condition	Meal Group	Recipe Name	Calories (Kcal)	Carbohydrates (g)	Protein (g)	Fat (g)	Actions
None	Breakfast	Oatmeal with Fresh Apple Slices	40.1	33	2	0	<input type="button" value="View"/> <input type="button" value="Delete"/>
None	Lunch	Daing na Bangus	142	13	9	6	<input type="button" value="View"/> <input type="button" value="Delete"/>
None	Dinner	Grilled Chicken Salad	179	3	17	11	<input type="button" value="View"/> <input type="button" value="Delete"/>
None	Snack	Cupcakes	0.2	46	4	0	<input type="button" value="View"/> <input type="button" value="Delete"/>

Recipe Nutritional Information

Total Calories	Total Carbohydrates	Total Protein	Total Fat
361.3	95	32	17

Figure 31: View Added Meals

(Figure 30). The content manager can view added meals and details (Figure 31) by choosing a date and view or delete each recipe. (Figure 31).

### C..10 Add Health Tip

Add Health Tip

Add Health Tip Back Home

Figure 32: Add Health Tip

By selecting the Health Tip under Add Content, the content manager can add health tip that would be shared to users. (Figure 32).

### C..11 View



Figure 33: View

By clicking the View button, link to Recipes Created can be accessed. (Figure 33).

### C..12 View Created Recipes

Health Condition	Meal Type	Recipe Name	Serving Size	Total Calories (Kcal)	Total Carbohydrate (g)	Total Protein (g)	Total Fat (g)	Actions
None	Dinner	Grilled Chicken Salad	1	179	3	17	11	
None	Lunch	Daing na Bangus	1	142	13	9	6	
None	Breakfast	Oatmeal with Fresh Apple Slices	1	40.1	33	2	0	
None	Snack	Cupcakes	1	0.2	46	4	0	

Below the table, there are 'Back' and 'Home' buttons.

Figure 34: View Created Recipes

By selecting Recipes Created under View, recipes created can be viewed, edited, or deleted. (Figure 34).



## D. User View

### D..1 User Home Page

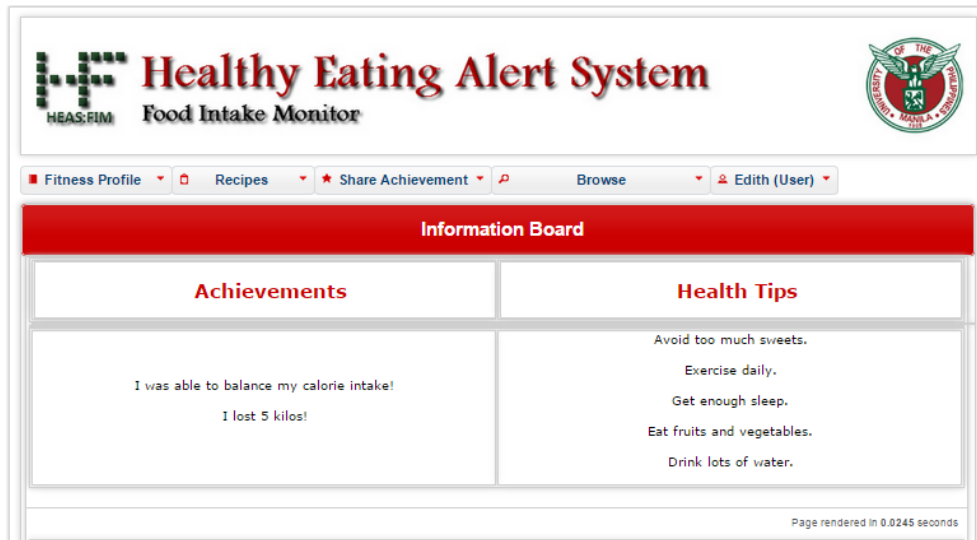


Figure 35: HEAS:FIM User Home Page

Figure 35 shows the user home page. User functions can be accessed through the buttons. He/She can view shared achievements and health tips, view, add or delete fitness information from his/her Fitness Profile, create recipes, view or delete created recipes, plan daily meals, share achievement, and browse shared and recommended recipes, recommended daily meal plan, and food nutritional information.

## D..2 Fitness Profile



Figure 36: Fitness Profile

By clicking the Fitness Profile button, links to view, add or delete Fitness Profile information can be accessed. (Figure 36).

## D..3 View Information

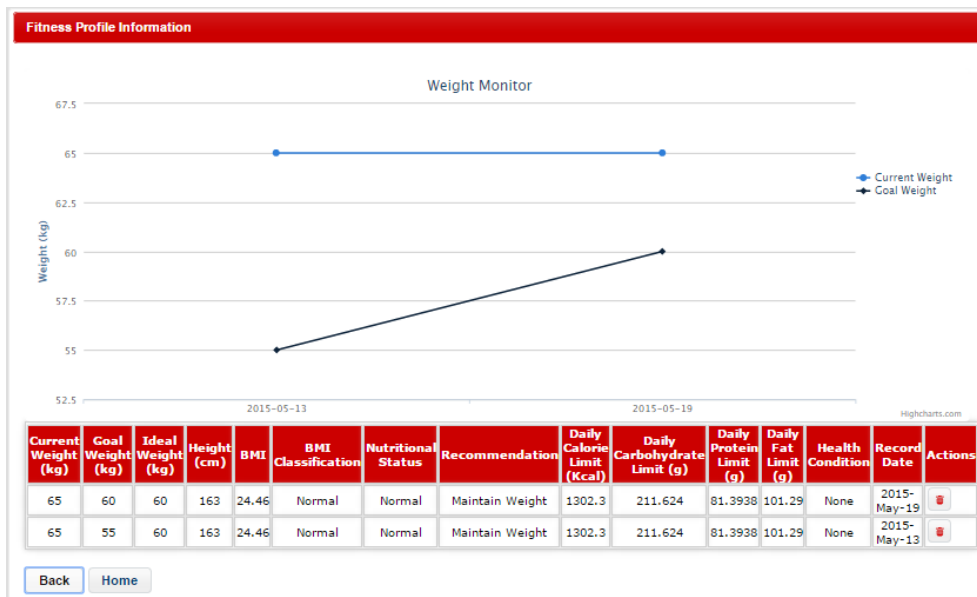


Figure 37: View Information

By selecting View Information under Fitness Profile, Fitness Profile information can be viewed or deleted. (Figure 37).

#### D..4 Add Information

**Add Fitness Profile Information**

Current Weight (kg):

Goal Weight (kg):

Height (cm):

Health Condition:  Diabetes  
 Hypertension  
 Kidney Stones  
 None

Date:

[Add Information](#) [Back](#) [Home](#)

Figure 38: Add Information

By selecting Add Information under Fitness Profile, Fitness Profile information can be added. (Figure 38).

#### D..5 Recipes

**Healthy Eating Alert System**  
Food Intake Monitor

OF THE  
UNIVERSITY OF THE  
PHILIPPINES  
MANILA

■ Fitness Profile ▾ Recipes ▾ Share Achievement ▾ Browse ▾ Edith (User) ▾

Create Recipe  
 Created Recipes  
 Plan Daily Meals

**Information Board**

Achievements	Health Tips
I was able to balance my calorie intake!	Drink lots of water.
I lost 5 kilos!	

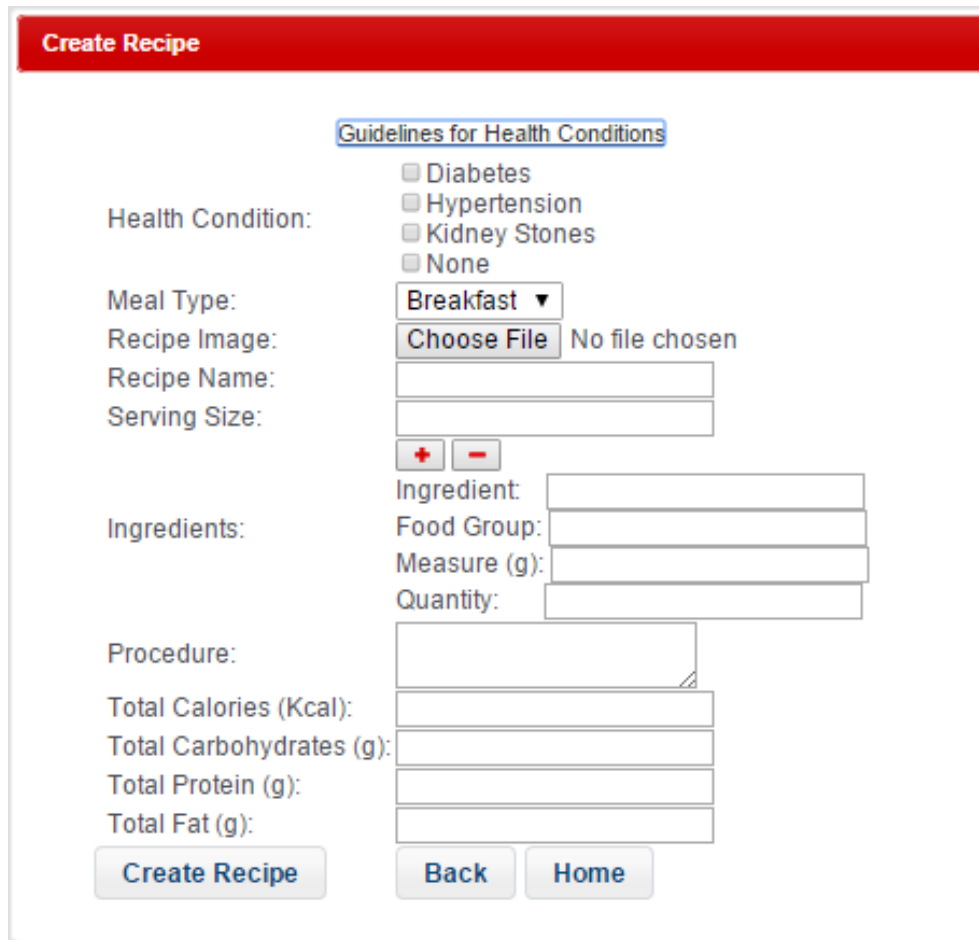
Page rendered in 0.0250 seconds

Figure 39: Recipes

By clicking the Recipes button, links to create, view, or delete created recipes,

and plan, view, or delete daily meals can be accessed. (Figure 39).

## D..6 Create Recipe



The screenshot shows a web form titled "Create Recipe" with a red header. The form is organized into several sections:

- Guidelines for Health Conditions:** A sub-section with four radio button options: Diabetes, Hypertension, Kidney Stones, and None.
- Health Condition:** A label for the radio button options.
- Meal Type:** A dropdown menu currently set to "Breakfast".
- Recipe Image:** A "Choose File" button and the text "No file chosen".
- Recipe Name:** A text input field.
- Serving Size:** A text input field.
- Ingredients:** A section containing a "+" and "-" button, and four input fields labeled "Ingredient:", "Food Group:", "Measure (g):", and "Quantity:".
- Procedure:** A text area for the recipe instructions.
- Nutritional Information:** Four input fields for "Total Calories (Kcal)", "Total Carbohydrates (g)", "Total Protein (g)", and "Total Fat (g)".

At the bottom of the form are three buttons: "Create Recipe", "Back", and "Home".

Figure 40: Create Recipe

By selecting the Create Recipe under Recipes, the user can create recipe/s considering health conditions if any. (Figure 40).

## D..7 View Created Recipes

Health Condition	Meal Type	Recipe Name	Serving Size	Total Calories (Kcal)	Total Carbohydrate (g)	Total Protein (g)	Total Fat (g)	Actions
None	Snack	Cupcakes	1	0.2	46	4	0	[Share] [Edit] [Delete] [Add]
None	Dinner	Grilled Chicken Salad	1	179	3	17	11	[Share] [Edit] [Delete] [Add]
None	Lunch	Daing na Bangus	1	142	13	9	6	[Share] [Edit] [Delete] [Add]
None	Breakfast	Oatmeal with Fresh Apple Slices	1	40.1	33	2	0	[Share] [Edit] [Delete] [Add]

[Back](#) [Home](#)

Figure 41: View Created Recipes

By selecting Created Recipes under Recipes, recipes created can be viewed, edited, deleted, or shared. (Figure 41).

## D..8 Plan Daily Meals

[Back](#) [Home](#)

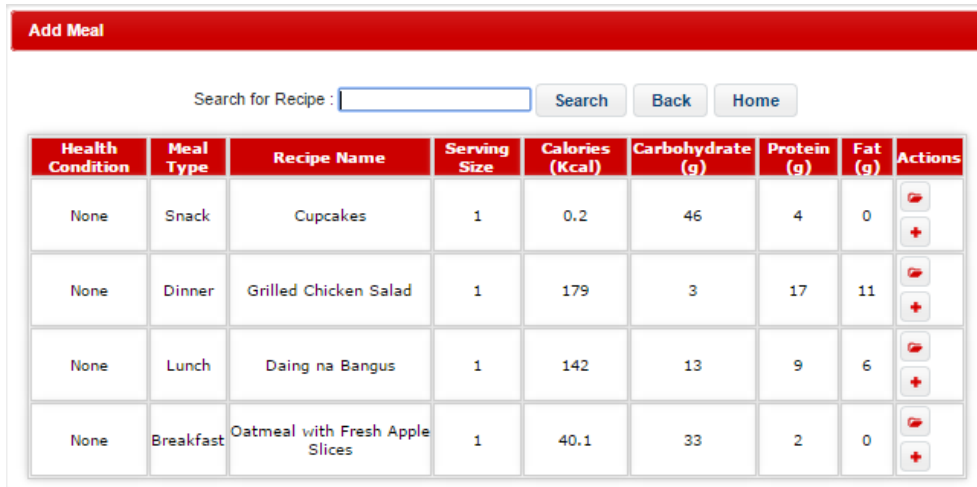
[Add Meals to Daily Meal Plan](#)

[Search Daily Meal Plan](#) **Date:**

Figure 42: Plan Daily Meals

By selecting Plan Daily Meals under Recipes, the user can add meals by clicking Search Meal or view or delete added meals by choosing a date and clicking Go. (Figure 42).

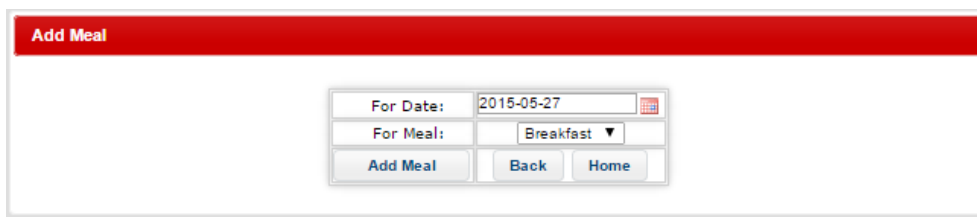
## D..9 Add Meal



The screenshot shows the 'Add Meal' interface. At the top, there is a red header bar with the text 'Add Meal'. Below the header, there is a search bar with the placeholder text 'Search for Recipe :', followed by 'Search', 'Back', and 'Home' buttons. Below the search bar is a table with the following columns: Health Condition, Meal Type, Recipe Name, Serving Size, Calories (Kcal), Carbohydrate (g), Protein (g), Fat (g), and Actions. The table contains four rows of data:

Health Condition	Meal Type	Recipe Name	Serving Size	Calories (Kcal)	Carbohydrate (g)	Protein (g)	Fat (g)	Actions
None	Snack	Cupcakes	1	0.2	46	4	0	
None	Dinner	Grilled Chicken Salad	1	179	3	17	11	
None	Lunch	Daing na Bangus	1	142	13	9	6	
None	Breakfast	Oatmeal with Fresh Apple Slices	1	40.1	33	2	0	

Figure 43: Add Meal



The screenshot shows the 'Add Meal' interface. At the top, there is a red header bar with the text 'Add Meal'. Below the header, there is a form with the following fields: 'For Date:' with a date input field containing '2015-05-27' and a calendar icon, 'For Meal:' with a dropdown menu showing 'Breakfast', and 'Add Meal', 'Back', and 'Home' buttons.

Figure 44: Add Recipe Meal

(Figure 43). The user can add the recipe meal to a chosen date (Figure 44) or view the recipe first.

## D..10 View or Delete Meal/s

Plan Daily Meals

Back
Home

Add Meals to Daily Meal Plan

Search Meal

Search Daily Meal Plan

Date: 2015-05-27

Go

Recipes

Health Condition	Meal Group	Recipe Name	Calories (Kcal)	Carbohydrates (g)	Protein (g)	Fat (g)	Actions
None	Breakfast	Oatmeal with Fresh Apple Slices	40.1	33	2	0	
None	Lunch	Daing na Bangus	142	13	9	6	
None	Dinner	Grilled Chicken Salad	179	3	17	11	
None	Snack	Cupcakes	0.2	46	4	0	

Recipe Nutritional Information

Total Calories	Total Carbohydrates	Total Protein	Total Fat
361.3	95	32	17

Fitness Information

Current Health Condition

None

	Limit	Remaining	Consumed	Recommendation
Calorie	1302.3	941	361.3	You may want to have more meals.
Carbohydrate	211.624	116.624	95	You may want to have more meals.
Protein	81.3938	49.3938	32	You may want to have more meals.
Fat	101.29	84.29	17	You may want to have more meals.

Figure 45: View or Delete Added Meals

The user can view added meals and details by choosing a date and view or delete each recipe. (Figure 45).

## D..11 Share Achievement



Figure 46: Share Achievement

By clicking the Share Achievement button, links to share achievement on site or on Facebook can be accessed. (Figure 46).

## D..12 Share Achievement on Site

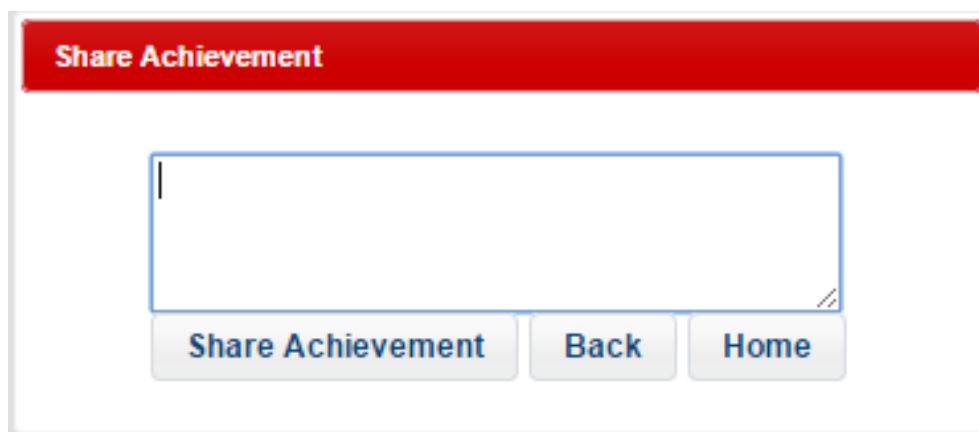


Figure 47: Share Achievement on Site

By selecting on this Site under Share Achievement, the user can share his/her achievement/s on the site. (Figure 47).



### D..13 Share Achievement on Facebook

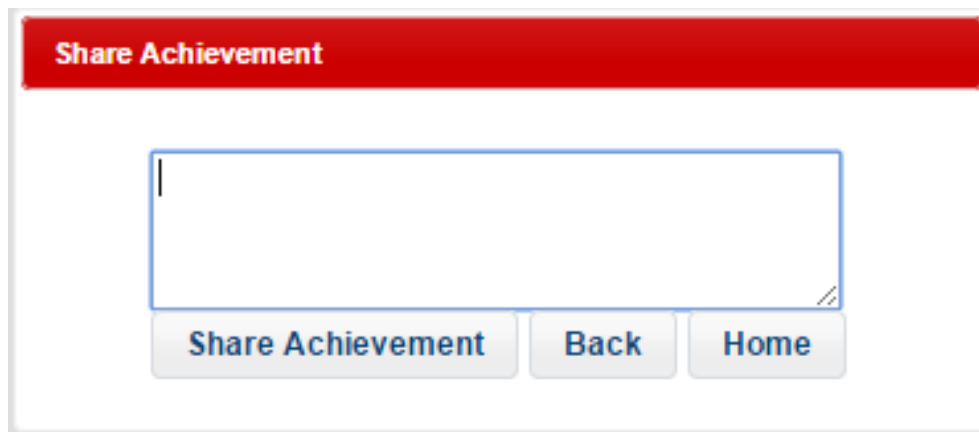


Figure 48: Share Achievement on Facebook

By selecting on Facebook under Share Achievement, the user can share his/her achievement/s on Facebook. (Figure 48).

### D..14 Browse



Figure 49: Browse

By clicking the Browse button, the user can browse for shared and recommended recipes, recommended daily meals, and food nutritional information. (Figure 49).

## D..15 Browse Shared Recipes

Health Condition	Meal Type	Recipe Name	Serving Size	Total Calories (Kcal)	Total Carbohydrate (g)	Total Protein (g)	Total Fat (g)	Actions
None	Snack	Cupcakes	1	0.2	46	4	0	
None	Dinner	Grilled Chicken Salad	1	179	3	17	11	
None	Lunch	Daing na Bangus	1	142	13	9	6	
None	Breakfast	Oatmeal with Fresh Apple Slices	1	40.1	33	2	0	

Figure 50: Browse Shared Recipes

By selecting Shared Recipes under Browse, the user can view recipes shared by other users. (Figure 50).

## D..16 Browse Recommended Recipes

Health Condition	Meal Type	Recipe Name	Serving Size	Total Calories (Kcal)	Total Carbohydrate (g)	Total Protein (g)	Total Fat (g)	Actions
None	Dinner	Grilled Chicken Salad	1	179	3	17	11	
None	Lunch	Daing na Bangus	1	142	13	9	6	
None	Breakfast	Oatmeal with Fresh Apple Slices	1	40.1	33	2	0	
None	Snack	Cupcakes	1	0.2	46	4	0	

Figure 51: Browse Recommended Recipes

By selecting Recommended Recipes under Browse, the user can view recommended recipes. (Figure 51).

## D..17 Browse Recommended Daily Meals

Back Home

Search Recommended Daily Meals Date:  Go

Figure 52: Browse Recommended Daily Meals

By selecting Recommended Daily Meals under Browse, the user can view recommended daily meals by date. (Figure 52).

#### D..18 Browse Food Nutritional Information

List Group	Food Group	Food Group Category	Food Name	Measure	Carbohydrate	Protein	Fat	Energy	Weight as Purchased	Weight Edible Portion
I-A	Vegetable	Vegetable, Vegetable A	Acelgas	1 cup raw or 1/2 cup cooked	0	0	0	0	25	45
I-A	Vegetable	Vegetable, Vegetable A	Alagaw Leaves	1 cup raw or 1/2 cup cooked	0	0	0	0	25	45
I-A	Vegetable	Vegetable, Vegetable A	Alugbati Leaves	1 cup raw or 1/2 cup cooked	0	0	0	0	25	45
I-A	Vegetable	Vegetable, Vegetable A	Ampalaya Fruit	1 cup raw or 1/2 cup cooked	0	0	0	0	25	45
I-A	Vegetable	Vegetable, Vegetable A	Ampalaya Leaves	1 cup raw or 1/2 cup cooked	0	0	0	0	25	45
I-A	Vegetable	Vegetable, Vegetable A	Balbalulang (Seaweed)	1 cup raw or 1/2 cup cooked	0	0	0	0	25	45

Figure 53: Browse Food Nutritional Information

By selecting Food Nutritional Information under Browse, the user can view the nutritional information of food. (Figure 53).

## VII. Discussions

Healthy Eating Alert System:Food Intake Monitor or HEAS:FIM is a system that helps monitor fitness and calorie intake towards a healthier living that focuses mainly on the Filipino diet. In this special problem, the application that has been developed is a combination of some of the separate functionalities of existing similar applications and an advancement as it allows creation of recipes from scratch and considers some of the health conditions. With the Codeigniter Framework having an MVC (Model-View-Controller) development pattern, there was not much difficulty experienced in developing the system aside from some Javascript coding, restriction of the amount of ingredients when needed due to health conditions, and integration with social media such as Facebook. Creation of forms with the Codeigniter validation and designing of forms were done using HTML, PHP, JQuery, Javascript, and CSS through Notepad++. The view files, which are in the presentation layer of the system architecture, contain the web interface made using the mentioned programming languages. The controller files, which contain the functionalities, serve as the link between the service layer and the database layer in which MySQL queries are made through the model files using PHP. Codeigniter's database class containing database functions made it easier to perform MySQL queries. Given this flow, it was not too difficult combining the three parts. HEAS:FIM offers daily meal planning, recipe creation from scratch with corresponding calorific values, restriction of the amount of ingredients considering health conditions, current weight versus goal weight and calorie monitor, recommendation of recipes, daily meals, and health tips, and sharing of achievements that will help monitor the calories of food to regulate food intake towards a healthier diet and ideal weight.

Although the use of Codeigniter made the creation of HEAS:FIM somewhat secure and efficient, there were still some difficulties. One of which was on the creation of the Create Recipe form. The built-in Codeigniter form validation cannot seem

to validate the added ingredients along with restrictions of which are contained in Javascript dynamic textboxes. After taking some time searching for a possible solution, the solution was to perform the validation both on the client-side, using Javascript, and the server-side. The data from the client-side should already be in the right form before passing to the server-side. Still, these difficulties did not hinder in achieving the goals of the system.

## VIII. Conclusions

Healthy Eating Alert System: Food Intake Monitor allows users to plan daily meals, create recipes from scratch with corresponding calorific values and restriction of the amount of ingredients considering health conditions, monitor calorie intake and goal weight if achieved, share achievements, and view health tips, recommended recipes and recommended meals. The system could help them monitor fitness and calorie intake towards a healthier living. Access to these functionalities could be gained by registering an account and logging in using the registered username and password provided that the account is activated.

A content manager is in charge of managing some of the information namely creation of recipes from scratch with corresponding calorific values and restriction of the amount of ingredients considering health conditions that would be recommended to users along with daily meal plans, health tips, and the entries in the Food Exchange List.

An administrator is responsible for managing user accounts. He/she can add, edit, delete, activate/deactivate an account.

## **IX. Recommendations**

Healthy Eating Alert System:Food Intake Monitor is a Health and Fitness application whose goal is to help monitor fitness and calorie intake for a healthier living. In this version, three health conditions: Diabetes, Hypertension, Kidney Stones were considered. The contents in the Food Exchange List were based from a diet guide from a hospital. The measurement used for the ingredients is grams, for the weight, kilograms, and for the height, centimeters. To have a more powerful HEAS:FIM, other health conditions can be added, more entries to the Food Exchange List from reliable sources to have a bigger database of ingredients, and automatic conversion of the measurement units to other desired units so there would be no need to manually convert, making the system much more efficient. Integration of the other social media can be done. Development of a mobile version and connection to physical hardware or health gears such as the existing android wear run accessories can also be done to make the system more handy. Other advanced functionalities can be added to the system.

## X. Bibliography

- [1] “What is nutrition? why is nutrition important?.” <http://www.medicalnewstoday.com/articles/160774.php>, July 2013.
- [2] “Food.” <http://global.britannica.com/EBchecked/topic/212568/food>.
- [3] “How your nutritional status affects your health.” <http://www.dummies.com/how-to/content/how-your-nutritional-status-affects-your-health.html>, May 2011.
- [4] N. M. S. T. M. City, “Guide to a computed diet,”
- [5] N. M. S. T. M. City, “Food exchange list,”
- [6] “Filipino diet.” <http://famouswonders.com/filipino-diet/>, 2011.
- [7] “Sparkpeople.” <http://www.sparkpeople.com>.
- [8] “Nutricoach.” <http://www.centrum.com.ph/node/13>.
- [9] “Fat secret.” <http://www.fatsecret.com>.
- [10] “How much fat is in your diet?.” <http://www.gnc.com.ph/home/article/how-much-fat-is-in-your-diet>, 2013.
- [11] M. S. L. E. Burke, J. Wang, “Self-monitoring in weight loss: A systematic review of the literature,” *American Dietetic Association*, vol. 111, no. 1, 2011.
- [12] H. Z. K. Z. G. J. Noronha, E.Hysen, “Platemate: Crowdsourcing nutrition analysis from food photographs,” *Harvard School of Engineering and Applied Sciences*, 2011.
- [13] C. for Nutrition Policy and P. U. D. of Agriculture, “Diet self-monitoring and body weight: A review of the evidence,” *Nutrition Insight*, vol. 48, 2012.



- [14] R. Steele, “An overview of the state of the art of automated capture of dietary intake information,” *The University of Sydney Discipline of Health Informatics*, 2013.
- [15] D. S. Adriyendi, “Prediction nutrition status using body mass index on mobile device,” *International Journal of Bio-Science and Bio-Technology*, vol. 5, no. 1, 2013.
- [16] H. L. G. C. d. S. K. Aizawa, Y. Maruyama, “Food balance estimation by using personal dietary tendencies in a multimedia food log,” *Journal of Latex Class Files*, vol. 6, no. 1, 2013.
- [17] B. Cunningham, “Smart phones and dietary tracking: A feasibility study,” *Arizona State University*, 2012.
- [18] T. Chang, “Food fight a social diet network mobile application,” *University of California at Berkeley Electrical Engineering and Computer Sciences*, 2012.
- [19] J. J. P. C. R. B. M. Silva, I. M. Lopes and P. Ray, “Sapofitness: A mobile health application for dietary evaluation in e-health networking applications and services (healthcom),” *2011 13th IEEE International Conference On*, pp. 375–380, 2011.
- [20] J. F. A. H. Andrew, G. Borriello, “Simplifying mobile phone food diaries,” *University of Washington Computer Science & Engineering DUB Group*, 2013.
- [21] I. Everyday Health, “Calorie counter.” <https://itunes.apple.com/ph/app/calorie-counter-dining-out/id375176386?mt=8>, 2014.
- [22] MedHelp, “My diet diary calorie counter.” <https://itunes.apple.com/ph/app/my-diet-diary-calorie-counter/id414169919?mt=8>, 2012–2015.

- [23] MyFitnessPal.com, “Calorie counter & diet tracker by myfitnesspal.” <https://itunes.apple.com/ph/app/calorie-counter-diet-tracker/id341232718?mt=8>, 2009–2015.
- [24] “Ideal body weight.” <http://www.manuelsweb.com/IBW.html>.
- [25] “Healthy asian recipes,” *Racial and Ethnic Approaches to Community Health*.
- [26] R. Rodibaugh, “The exchange list system for diabetic meal planning,” *Family and Consumer Sciences University of Arkansas: Division of Agriculture*.
- [27] “Carbohydrate counting with filipino foods,” *San Jose State University*, 2012.
- [28] “The type 2 diabetes meal planner.” <http://www.mydrs.com/data/files/Type2DiabetesMealPlanner.pdf>.
- [29] “Eat heart smart filipino style,” *American Heart Association*.
- [30] “Promoting good nutrition and healthy diet module 3.” <http://www.wpro.who.int/philippines/publications/module3.pdf>.
- [31] “The dash diet,” *National Institutes of Health*.
- [32] “Diet for kidney stone prevention,” *National Kidney and Urologic Diseases Information Clearinghouse*, 2013.
- [33] “What is a web app? heres our definition.” <http://web.appstorm.net/general/opinion/what-is-a-web-app-heres-our-definition/>, January 2011.
- [34] “An introduction to database management systems.” [http://www.personal.psu.edu/glh10/ist110/topic/topic07/topic07\\_05.html](http://www.personal.psu.edu/glh10/ist110/topic/topic07/topic07_05.html).
- [35] “Dbms.” <http://www.techterms.com/definition/dbms>.

# XI. Appendix

## A. Forms

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#addFPButton, #addFPBackButton, #addFPHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#record_date" ).datepicker({
clickInput: true,
changeMonth: true, //allow user to select month
changeYear: true, //allow user to select from year range
maxDate: '0',
dateFormat: 'yy-mm-dd',
yearRange: "-100:+0",
onClose: function() {
$( "#record_date" ).valid();
},
gotoCurrent: true}).datepicker( 'setDate', "0" );

$( "#addFPDialog" ).dialog({
autoOpen: true,
width: 400,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"

});
});

function confirm_add_fitness_profile() {
//var current_weight = document.getElementById("current_weight").value;
//var goal_weight = document.getElementById("goal_weight").value;
//var height = document.getElementById("height").value;
//var health_condition = document.getElementById("health_condition").value;
//var record_date = document.getElementById("record_date").value;
//var msg = "Are you sure you want to submit the following information?" + "\n" + "Current
Weight (kg): " + current_weight + "\n" + "Goal Weight (kg): " + goal_weight + "\n" + "
Height (cm): " + height + "\n" + "Health Condition: " + health_condition + "\n" + "
Record Date: " + record_date;
var msg = "Are you sure you want to submit the following information?";
return confirm(msg);
}
</script>
</head>
<body>
<div id="addFPDialog" title="Add Fitness Profile Information">
<br>
<font color="red" size="2"><b><?php echo validation_errors(); ?></b></font>
<?php echo form_open('home.user/add_fitness_profile_save '); ?>
<center>
<table>
<tr></tr></table>
<tr>
<td><input type="text" value="<?php echo set_value('current_weight '); ?>">
</td>
<td><input type="text" value="<?php echo set_value('goal_weight '); ?>">
</td>
<td><input type="text" value="<?php echo set_value('height '); ?>">
</td>
</tr>
</table>
</div>

```

```

<td><input id="height" name="height" type="text" value="<?php echo set_value('height'); ?>" >
</tr>
<tr>
<td><label for="health_condition">&nbsp; Health Condition:</label>
<td><input type="checkbox" id="health_condition" name="health_condition[]" value="Diabetes" <?
php if (!empty($_POST["health_condition"]) && in_array("Diabetes", $_POST["
health_condition"])) { echo "checked"; ?>>Diabetes<br>
<input type="checkbox" id="health_condition" name="health_condition[]" value="Hypertension" <?
php if (!empty($_POST["health_condition"]) && in_array("Hypertension", $_POST["
health_condition"])) { echo "checked"; ?>>Hypertension<br>
<input type="checkbox" id="health_condition" name="health_condition[]" value="Kidney Stones"
<?php if (!empty($_POST["health_condition"]) && in_array("Kidney Stones", $_POST["
health_condition"])) { echo "checked"; ?>>Kidney Stones<br>
<input type="checkbox" id="health_condition" name="health_condition[]" value="None" <?php if
(!empty($_POST["health_condition"]) && in_array("None", $_POST["health_condition"])) {
echo "checked"; ?>>None<br>
<!--<td><select id="health_condition" name="health_condition">
<option value="Diabetes" <?php if(isset($_POST["health_condition"]) && $_POST["
health_condition"] == "Diabetes") echo "selected";?>>Diabetes</option>
<option value="Hypertension" <?php if(isset($_POST["health_condition"]) && $_POST["
health_condition"] == "Hypertension") echo "selected";?>>Hypertension</option>
<option value="Kidney Stones" <?php if(isset($_POST["health_condition"]) && $_POST["
health_condition"] == "Kidney Stones") echo "selected";?>>Kidney Stones</option>
<option value="None" <?php if(isset($_POST["health_condition"]) && $_POST["health_condition"]
== "None") echo "selected";?>>None</option>
</select>-->
</tr>
<!--<tr>
<td><label for="health_conditions">&nbsp; Health Conditions (if any, e.g. Diabetes Type II,
Hypertension, Kidney Stones):</label>
<td><input id="health_conditions" name="health_conditions" type="text" value="<?php echo
set_value('health_conditions'); ?>" >
</tr>-->
<tr>
<td><label for="record_date">&nbsp; Date:</label>
<td><input id="record_date" name="record_date" type="text" class="date-pick dp-applied" value
="<?php echo set_value('record_date'); ?>" readonly="readonly">
<a href="#record_date" class="dp-choose-date" title="Choose date">Choose date</a>
</tr>
<tr>
<td><button id="addFPButton" type="submit" name="submit" value="submit" onclick="return
confirm_add_fitness_profile()">Add Information</button>
<td><button id="addFPBackButton" type="button" name="addFPBackButton" value="Back" onclick="
window.location.href='<?php echo site_url(); ?>home_user'">Back</button>
<button id="addFPHomeButton" type="button" name="addFPHomeButton" value="Home" onclick="window
.location.href='<?php echo site_url(); ?>home_user'">Home</button>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#addFELItemButton, #addFELItemBackButton, #addFELItemHomeButton" ).button();

$('amf').css('display', 'none');

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#addFELItemDialog" ).dialog({
autoOpen: true,
width: 750,
modal: true,
closeOnEscape: false,
beforeClose: function( event, ui ) { return false; },
dialogClass: "noclose"

});
});

```

```

$(document).ready(function() {
$("#list_group").change(function() {
/*dropdown post */
$.ajax({
url:"<?php echo
base_url();?>index.php/home_cmanager/buildDropFEL",
data: {list_group:
$(this).val()},
type: "POST",
success: function(data) {
$("#food_group").html(data);
}
});
toggleTextBox(this);
});

$("#list_group").change(function() {
/*dropdown post */
$.ajax({
url:"<?php echo
base_url();?>index.php/home_cmanager/buildDropFEL2",
data: {list_group:
$(this).val()},
type: "POST",
success: function(data) {
$("#food_group_category").html(data);
}
});
toggleTextBox(this);
});
});

function toggleTextBox(element) {
//alert(element.value);
if(element.value == "Other") {
if(document.getElementById(element.id+"other") == null)
$(document.createElement("input")).attr({
type: "text",
id: element.id+"other",
name: element.name+"other"
}).insertAfter(element);
} else {
$("#"+element.id+"other").remove();
}
}
</script>
</head>
<body>
<div id="addFELItemDialog" title="Add Food Exchange List Item">
<br>

<font color="red" size="2"><b><?php echo validation_errors(); ?></b></font>
<?php echo form_open('home_cmanager/add_felitem_save '); ?>
<center>
<table>
<tr></tr><tr></tr>
<tr>
<td><label for="list_group">&nbsp;   List Group:</label>
<td><?php echo form_dropdown('list_group ', $lg, '', 'class="required" id="list_group" '); ?>
</tr>
<tr>
<td><label for="food_group">&nbsp;   Food Group:</label>
<td><select id ="food_group" name="food_group" class="food_group" onchange='toggleTextBox(this
)'>
<option value="" >Please select Food Group</option>
</select>
</tr>
<tr>
<td><label for="food_group_category">&nbsp;   Food Group Category:</label>
<td><select id ="food_group_category" name="food_group_category" class="food_group_category"
onchange='toggleTextBox(this)''>
<option value="">Please select Food Group Category</option>
</select>
</tr>
<tr>
<td><label for="food_name">&nbsp;   Food Name:</label>
<td><input id="food_name" name="food_name" type="text" value="<?php echo set_value('food_name
'); ?> " >
</tr>
<tr>
<td><label for="measure">&nbsp;   Measure:</label>
<td><input id="measure" name="measure" type="text" value="<?php echo set_value('measure');
?>" >
</tr>
<tr>
<td><label for="carbohydrate">&nbsp;   Carbohydrate (g):</label>
<td><input id="carbohydrate" name="carbohydrate" type="text" value="<?php echo set_value('
carbohydrate '); ?>" >
</tr>
<tr>
<td><label for="protein">&nbsp;   Protein (g):</label>
<td><input id="protein" name="protein" type="text" value="<?php echo set_value('protein ');
?>" >
</tr>
<tr>
<td><label for="fat">&nbsp;   Fat (g):</label>
<td><input id="fat" name="fat" type="text" value="<?php echo set_value('fat '); ?>" >
</tr>

```

```

<tr>
<td><label for="energy">&nbsp; Energy (Kcal):</label>
<td><input id="energy" name="energy" type="text" value="<?php echo set_value('energy'); ?>" >
</tr>
<tr>
<td><label for="weight_as_purchased">&nbsp; Weight as Purchased (g):</label>
<td><input id="weight_as_purchased" name="weight_as_purchased" type="text" value="<?php echo
set_value('weight_as_purchased'); ?>" >
</tr>
<tr>
<td><label for="weight_edible_portion">&nbsp; Weight Edible Portion (g):</label>
<td><input id="weight_edible_portion" name="weight_edible_portion" type="text" value="<?php
echo set_value('weight_edible_portion'); ?>" >
</tr>
<tr>
<td><button id="addFELItemButton" type="submit" name="submit" value="submit">Add Item</button>
<td><button id="addFELItemBackButton" type="button" name="addFELItemBackButton" value="Back"
onclick="window.location.href='<?php echo site_url(); ?>home_cmanager'">Back</button>
<button id="addFELItemHomeButton" type="button" name="addFELItemHomeButton" value="Home"
onclick="window.location.href='<?php echo site_url(); ?>home_cmanager'">Home</button>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#addHealthTipButton, #addHealthTipBackButton, #addHealthTipHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#addHealthTipDialog" ).dialog({
autoOpen: true,
width: 530,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"

});
});
</script>
</head>
<body>
<div id="addHealthTipDialog" title="Add Health Tip">
<br>
<font color="red" size="2"><b><?php echo validation_errors(); ?></b></font>
<?php echo form_open('home_cmanager/add_healthtip_save'); ?>
<center>
<table>
<tr></tr><tr></tr>
<tr>
<td><textarea rows="4" cols="50" id="health_tip" name="health_tip" type="longtext" value="<?
php echo set_value('health_tip'); ?>" ></td>
</tr>
<tr>
<td><button id="addHealthTipButton" type="submit" name="submit" value="submit">Add Health Tip
</button>
<button id="addHealthTipBackButton" type="button" name="addHealthTipBackButton" value="Back"
onclick="window.location.href='<?php echo site_url(); ?>home_cmanager'">Back</button>
<button id="addHealthTipHomeButton" type="button" name="addHealthTipHomeButton" value="Home"
onclick="window.location.href='<?php echo site_url(); ?>home_cmanager'">Home</button>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url();?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url();?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#addMealBackButton, #addMealButton, #addMealHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#record_date" ).datepicker({
clickInput:true,
changeMonth: true, //allow user to select month
changeYear: true, //allow user to select from year range
minDate: '0',
dateFormat: 'yy-mm-dd',
yearRange: "-100:+0",
onClose: function() {
$( '#record_date' ).valid();
},
gotoCurrent: true}).datepicker('setDate','0');

$( "#addMealDialog" ).dialog({
autoOpen: true,
width: 850,
modal: true,
closeOnEscape: false,
beforeclose: function( event, ui ) { return false; },
dialogClass: "noclose"
});
</script>
</head>
<body>
<div id="addMealDialog" title="Add Meal">
<br>
<font color="red" size="2"><b><?php echo validation_errors();?></b></font>
<?php echo form_open("home.cmanager/add_meal/$recommended_recipe_id");?>
<center>
<table>
<tr style="display:none;">
<td><label for="recommended_recipe_id">&nbsp;&nbsp;&nbsp;Recipe ID:</label>
<td><input id="recommended_recipe_id" name="recommended_recipe_id" type="text" value="<?php echo $recommended_recipe_id;?>" readonly="readonly">
</tr>
<tr>
<td><label for="record_date">&nbsp;&nbsp;&nbsp;For Date:</label>
<td><input id="record_date" name="record_date" type="text" class="date-pick dp-applied" value="<?php echo set_value('record_date');?>" readonly="readonly">
<a href="#record_date" class="dp-choose-date" title="Choose date">Choose date</a>
</tr>
<tr>
<td><label for="meal-group">&nbsp;&nbsp;&nbsp;For Meal:</label>
<td><select id="meal-group" name="meal-group">
<option value="Breakfast" <?php if(isset($_POST["meal_group"]) && $_POST["meal_group"] == "Breakfast") echo "selected";?>>Breakfast</option>
<option value="Lunch" <?php if(isset($_POST["meal_group"]) && $_POST["meal_group"] == "Lunch") echo "selected";?>>Lunch</option>
<option value="Dinner" <?php if(isset($_POST["meal_group"]) && $_POST["meal_group"] == "Dinner") echo "selected";?>>Dinner</option>
<option value="Snack" <?php if(isset($_POST["meal_group"]) && $_POST["meal_group"] == "Snack") echo "selected";?>>Snack</option>
</select>
</tr>
<tr>
<td><button id="addMealButton" type="submit" name="submit" value="submit">Add Meal</button>
<td><button id="addMealBackButton" type="button" name="addMealBackButton" value="Back" onclick="window.location.href='<?php echo site_url();?>home.cmanager/search_meal'">Back</button>
<td><button id="addMealHomeButton" type="button" name="addMealHomeButton" value="Home" onclick="window.location.href='<?php echo site_url();?>home.cmanager'">Home</button>
</tr>
</table>
</center>
<?php echo form_close();?>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url();?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url();?>validation/jquery-1.11.0.min.js"></script>
</script>

$(function() {

$( "#addMealBackButton, #addMealButton, #addMealHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#record_date" ).datepicker({
clickInput:true,
changeMonth: true, //allow user to select month
changeYear: true, //allow user to select from year range
minDate: '0',
dateFormat: 'yy-mm-dd',
yearRange: "-100:+0",
onClose: function() {
$('#record_date').valid();
},
gotoCurrent: true}).datepicker('setDate','0');

$( "#addMealDialog" ).dialog({
autoOpen: true,
width: 850,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"
});
</script>
</head>
<body>
<div id="addMealDialog" title="Add Meal">
<br>
<font color="red" size="2"><b><?php echo validation_errors();?></b></font>
<?php echo form_open("home_user/add-meal/$own_recipe_id");?>
<center>
<table>
<tr style="display:none;">
<td><label for="own_recipe_id">&nbsp; Recipe ID:</label>
<td><input id="own_recipe_id" name="own_recipe_id" type="text" value="<?php echo $own_recipe_id; ?>" readonly="readonly">
</tr>
<tr>
<td><label for="record_date">&nbsp; For Date:</label>
<td><input id="record_date" name="record_date" type="text" class="date-pick dp-applied" value="<?php echo set_value('record_date'); ?>" readonly="readonly">
<a href="#" record_date" class="dp-choose-date" title="Choose date">Choose date</a>
</tr>
<tr>
<td><label for="meal_group">&nbsp; For Meal:</label>
<td><select id="meal_group" name="meal_group">
<option value="Breakfast" <?php if (isset($_POST["meal_group"]) && $_POST["meal_group"] == "Breakfast") echo "selected";?>>Breakfast</option>
<option value="Lunch" <?php if (isset($_POST["meal_group"]) && $_POST["meal_group"] == "Lunch") echo "selected";?>>Lunch</option>
<option value="Dinner" <?php if (isset($_POST["meal_group"]) && $_POST["meal_group"] == "Dinner") echo "selected";?>>Dinner</option>
<option value="Snack" <?php if (isset($_POST["meal_group"]) && $_POST["meal_group"] == "Snack") echo "selected";?>>Snack</option>
</select>
</tr>
<tr>
<td><button id="addMealButton" type="submit" name="submit" value="submit">Add Meal</button>
<td><button id="addMealBackButton" type="button" name="addMealBackButton" value="Back" onclick="window.location.href='<?php echo site_url();?>home_user/search_meal'">Back</button>
<td><button id="addMealHomeButton" type="button" name="addMealHomeButton" value="Home" onclick="window.location.href='<?php echo site_url();?>home_user'">Home</button>
</tr>
</table>
</center>
<?php echo form_close();?>
</div>
</body>
</html>

```



```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
var quantity;
$(function() {
var data2 = [];
var index = 0;

$( "#addRecipeButton , #addRecipeBackButton , #addRecipeHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#record_date" ).datepicker({
clickInput:true,
changeMonth: true, //allow user to select month
changeYear: true, //allow user to select from year range
dateFormat: 'yy-mm-dd',
yearRange: "-100:+0",
onClose: function() {
$( '#record_date' ).valid();
}
});
$( "#addRecipeDialog" ).dialog({
autoOpen: true,
width: 600,
modal: true,
closeOnEscape: false,
beforeclose: function( event, ui ) { return false; },
dialogClass: "noclose"
});

var id;
$( '#ingredients' ).autocomplete({
minLength: 0,
source: function( request, response ) {
id = this.element.context.attributes[ name="id" ].value;
//console.log(id);
$.ajax( {
url: "<?php echo site_url('home_cmanager/suggestions' ); ?>",
data: request,
dataType: "json",
type: "POST",
success: function( data ) {
//console.log( JSON.stringify( data ) );
response( $.map( data, function( v, i ) {
if ( v.measure != null )
data2[ data2.length ] = v.measure;
else if ( v.food_name != null )
data2[ data2.length ] = v.food_name;
else if ( v.energy != null )
data2[ data2.length ] = v.energy;
else if ( v.carbohydrate != null )
data2[ data2.length ] = v.carbohydrate;
else if ( v.protein != null )
data2[ data2.length ] = v.protein;
else if ( v.fat != null )
data2[ data2.length ] = v.fat;
else if ( v.food_group != null )
data2[ data2.length ] = v.food_group;

return {
label: v.food_name
}
}
)
));
}
}
);
},
select: function( event, ui ) {
var num = id.substring( id.indexOf( "s" ) + 1, id.length );
var measure = null;
var food_group = null;
var calorie , carbohydrate , protein , fat;
for( var i = 0; i < data2.length; i +=7 ) {
if ( data2[ i ] == ui.item.value ) {
measure = data2[ i + 1 ];
calorie = data2[ i + 2 ];
carbohydrate = data2[ i + 3 ];

```

```

protein = data2[i + 4];
fat = data2[i + 5];
food_group = data2[i+6];
}
}
//console.log(calorie);
$('#ingredient_measure'+num).val (measure);
$('#food_group'+num).val (food_group);
/*$('#calories ').val(+$('#calories ').val() ++calorie);
$('#carbohydrate ').val(+$('#carbohydrate ').val() ++carbohydrate);
$('#protein ').val(+$('#protein ').val() ++protein);
$('#fat ').val(+$('#fat ').val() ++fat);*/
}
/*focus: function( event, ui ) {
$( ".ingredients" ).val( ui.item.label);
},
select: function( event, ui ) {
}*/
});
});

function update(id) {
var request = new XMLHttpRequest(), url = "<?php echo site_url('home_cmanager/get_fa '); ?>",
params = "ingredient="+$("#ingredients"+id).val();

request.onreadystatechange = function() {
if(request.readyState == 4 && request.status == 200) {
var temp = request.responseText.split(","); i;

$('#calories ').val(+$('#calories ').val() - (+temp[0] * quantity[id]));
$('#carbohydrate ').val(+$('#carbohydrate ').val() - (+temp[1] * quantity[id]));
$('#protein ').val(+$('#protein ').val() - (+temp[2] * quantity[id]));
$('#fat ').val(+$('#fat ').val() - (+temp[3] * quantity[id]));

for(i = 0; i < temp.length; i++)
temp[i] = +temp[i] * $("#ingredient_quantity"+id).val();

$('#calories ').val(+$('#calories ').val() + temp[0]);
$('#carbohydrate ').val(+$('#carbohydrate ').val() + temp[1]);
$('#protein ').val(+$('#protein ').val() + temp[2]);
$('#fat ').val(+$('#fat ').val() + temp[3]);

quantity[id] = $("#ingredient_quantity"+id).val();
}
}
request.open("POST", url, true);
request.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
request.send(params);
}

$(document).ready(function(){
var counter = 1;
var data2 = [];
quantity = new Array();

quantity[0] = $("#ingredient_quantity0").val() == "" ? 0 : $("#ingredient_quantity0").val();

$("#addIngredientButton").click(function () {

/*if(counter >10){
alert("Only 10 textboxes allow");
return false;
}
*/

quantity[counter] = 0;

var newTextBoxDiv = $(document.createElement('div')).attr("id", 'TextBoxDiv' + counter);
newTextBoxDiv.after().html('<br><label>Ingredient: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</label>' +
'<input type="text" class="ingredients" name="ingredients['+counter+']" id="ingredients' +
counter + '" value="" >');

var newTextBoxDiv1= $(document.createElement('div')).attr("id", 'TextBoxDiv1' + counter);
newTextBoxDiv1.after().html('<label>Measure (g): </label>' +
'<input type="text" class="measure" name="ingredient_measure['+counter+']" id="
ingredient_measure' + counter + '" value="" >');

var newTextBoxDiv2= $(document.createElement('div')).attr("id", 'TextBoxDiv2' + counter);
newTextBoxDiv2.after().html('<label>Quantity: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</label>' +
'<input type="text" name="ingredient_quantity['+counter+']" id="ingredient_quantity' + counter
+ '" value="" onchange="update('+counter+')" >');

var newTextBoxDiv3= $(document.createElement('div')).attr("id", 'TextBoxDiv3' + counter);
newTextBoxDiv3.after().html('<label>Food Group: </label>' +
'<input type="text" class="food_group" name="food_group['+counter+']" id="food_group' +
counter + '" value="" >');

newTextBoxDiv.appendTo("#TextBoxesGroup");
newTextBoxDiv3.appendTo("#TextBoxesGroup");
newTextBoxDiv1.appendTo("#TextBoxesGroup");
newTextBoxDiv2.appendTo("#TextBoxesGroup");

document.getElementById("ingredient_measure" + counter).readOnly=true;
document.getElementById("food_group" + counter).readOnly=true;

var id;
$('#ingredients').autocomplete({
minLength: 0,
source: function(request, response) {
id = this.element.context.attributes[name="id"].value;

```

```

//console.log(id);
$.ajax({
url: "<?php echo site_url('home_cmanager/suggestions '); ?>",
data: request,
dataType: "json",
type: "POST",
success: function(data) {
response($.map(data, function(v,i){
if(v.measure != null)
data2[data2.length] = v.measure;
else if(v.food_name != null)
data2[data2.length] = v.food_name;
else if(v.energy != null)
data2[data2.length] = v.energy;
else if(v.carbohydrate != null)
data2[data2.length] = v.carbohydrate;
else if(v.protein != null)
data2[data2.length] = v.protein;
else if(v.fat != null)
data2[data2.length] = v.fat;
else if(v.food_group != null)
data2[data2.length] = v.food_group;

return {
label: v.food_name
};
}));
});
},
select: function(event, ui) {
var num = id.substring(id.indexOf("s") + 1, id.length);
var measure = null;
var food_group = null;
var calorie, carbohydrate, protein, fat;
for(var i = 0; i < data2.length; i+=7){
if(data2[i] == ui.item.value) {
measure = data2[i + 1];
calorie = data2[i + 2];
carbohydrate = data2[i + 3];
protein = data2[i + 4];
fat = data2[i + 5];
food_group = data2[i+6];
}
}
//console.log(calorie);
$('#ingredient_measure'+num).val (measure);
$('#food_group'+num).val (food_group);
/*$('#calories ').val($('#calories ').val() ++calorie);
$('#carbohydrate ').val($('#carbohydrate ').val() ++carbohydrate);
$('#protein ').val($('#protein ').val() ++protein);
$('#fat ').val($('#fat ').val() ++fat);
*/
});
counter++;
});

$("#removeIngredientButton").click(function () {
if (counter==1){
alert("No more textbox to remove");
return false;
}
counter--;

var request = new XMLHttpRequest(), url = "<?php echo site_url('home_cmanager/get_fa '); ?>",
params = "ingredient="+($('#ingredients'+counter).val() == "" ? 0 : $('#ingredients'+
counter).val());

request.onreadystatechange = function() {
if(request.readyState == 4 && request.status == 200) {
var temp = request.responseText.split(",");
$("#TextBoxDiv" + counter).remove();
$("#TextBoxDiv3" + counter).remove();
$("#TextBoxDiv1" + counter).remove();
$("#TextBoxDiv2" + counter).remove();

if(quantity[counter] != "" && quantity[counter] != null) {
$('#calories ').val($('#calories ').val() - (+temp[0] * quantity[counter]));
$('#carbohydrate ').val($('#carbohydrate ').val() - (+temp[1] * quantity[counter]));
$('#protein ').val($('#protein ').val() - (+temp[2] * quantity[counter]));
$('#fat ').val($('#fat ').val() - (+temp[3] * quantity[counter]));
}
}
}
request.open("POST", url, true);
request.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
request.send(params);
});
});

function healthConditionsGuidelinesModal() {
$("#healthConditionsGuidelinesModalDialog").dialog({
autoOpen: false,
modal: true,
title: 'Guidelines for Health Conditions',
width: 'auto',
height: 'auto',
buttons: {

```

```

"Ok": function () {
$(this).dialog("close");
},
}).dialog('open');
return false;
}

function validate() {
var errors = "<p style='color: red; font-weight: bold'>";
if($("#recipe_name").val().trim() == "")
errors += "The Recipe Name field is required.<br>";
if($("#serving_sugg").val().trim() == "")
errors += "The Serving Size field is required.<br>";
if($("#preparation_procedure").val().trim() == "")
errors += "The Procedure field is required.<br>";
if($("#ingredients0").val().trim() == "")
errors += "The Ingredients field is required.<br>";
var file = $("#input[name='userfile']").val();
if(file != "") {
var filetypes = ["gif", "jpg", "jpeg", "png"];
var ext = file.substr(file.lastIndexOf(".") + 1, file.length).toLowerCase();
if(filetypes.indexOf(ext) < 0)
errors += "The image file type is invalid.<br>";
}

var ingredients = new Array();

$("#ingr_cont input[type='text']").each(function() {
if($(this).attr("id").indexOf("ingredients") > -1 && $(this).val().trim() != "") {
var id = $(this).attr("id").substr($(this).attr("id").indexOf("s") + 1, $(this).attr("id").length);
var value = {ingredient: $(this).val().trim(), foodgroup: $("#food_group"+id).val().trim()};
if($("#ingredient_quantity"+id).val().trim() == "")
errors += "The Quantity field is required.<br>";
else
value.quantity = $("#ingredient_quantity"+id).val().trim();

if(value.quantity != null)
ingredients.push(value);
}
});

var health_condition = $("#health_condition").val(), i;

for (i = 0; i < ingredients.length; i++) {
var food_group = ingredients[i].foodgroup, count = 0, j;

for(j = 0; j < ingredients.length; j++)
if(ingredients[i].foodgroup == food_group)
count += parseFloat(ingredients[j].quantity);

console.log(count+";" + food_group);

if(health_condition == "Diabetes") {
if((food_group == "Vegetable" || food_group == "Fruit") && count > 4) {
errors += "You are only allowed a maximum of 4 servings of "+food_group+" daily.<br>";
} else if(food_group == "Milk" && count > 3) {
errors += "You are only allowed a maximum of 3 servings of Milk daily.<br>";
} else if(food_group == "Rice" && count > 9) {
errors += "You are only allowed a maximum of 9 servings of Rice daily.<br>";
} else if(food_group == "Meat" && count > 6) {
errors += "You are only allowed a maximum of 6 servings of Meat daily.<br>";
} else if(food_group == "Meat" && count > 5) {
errors += "You are only allowed a maximum of 5 servings of Fat daily.<br>";
}
} else if(health_condition == 'Hypertension' || health_condition == 'Kidney Stones') {
if((food_group == "Vegetable" || food_group == "Fruit") && count > 5) {
errors += "You are only allowed a maximum of 5 servings of "+food_group+" daily.<br>";
} else if(food_group == "Milk" && count > 5) {
errors += "You are only allowed a maximum of 5 servings of Milk daily.<br>";
} else if(food_group == "Rice" && count > 8) {
errors += "You are only allowed a maximum of 8 servings of Rice daily.<br>";
} else if(food_group == "Meat" && count > 2) {
errors += "You are only allowed a maximum of 2 servings of Meat daily.<br>";
} else if(food_group == "Meat" && count > 3) {
errors += "You are only allowed a maximum of 3 servings of Fat daily.<br>";
}
}
}

errors += "</p>";
if(errors != "<p style='color: red; font-weight: bold'></p>") {
$("#addRecipeDialog").prepend(errors);
return false;
}
}
</script>
</head>
<body>
<div id="healthConditionsGuidelinesModalDialog" style="display: none;">
<center>
<table>
<tr>
<td>
<p>Below are the maximum daily servings for each food group for each health condition:
</tr>
<tr>
<th><h4>Food Group</th>

```

```

<th><h4>Diabetes</th>
<th><h4>Hypertension</th>
<th><h4>Kidney Stones</th>
</tr>
<tr>
<td>Vegetable
<td>4
<td>5
<td>5
</tr>
<tr>
<td>Fruit
<td>4
<td>5
<td>5
</tr>
<tr>
<td>Milk
<td>3
<td>5
<td>5
</tr>
<tr>
<td>Rice
<td>9
<td>8
<td>8
</tr>
<tr>
<td>Meat
<td>6
<td>2
<td>2
</tr>
<tr>
<td>Fat
<td>5
<td>3
<td>3
</tr>
</table>
</center>
</div>

<div id="addRecipeDialog" title="Create Recipe ">
<br>
<font color="red" size="2"><b><?php echo validation_errors(); ?><?php if (isset($error_display
)) { echo $error_display; } ?></b></font>
<?php echo form_open_multipart('home/cmanager/add_recipe_save', array('onsubmit'=>'return
validate()')); ?>
<center>
<table>
<tr>
<tr>
<a href="#" onclick="javascript:healthConditionsGuidelinesModal();">Guidelines for Health
Conditions</a>
</tr>
<tr></tr><tr></tr>
<tr>
<td><label for="health_condition">&nbsp;&nbsp;&nbsp; Health Condition:</label>
<td><input type="checkbox" id="health_condition" name="health_condition[]" value="Diabetes" <?
php if ((!empty($_POST["health_condition"])) && in_array("Diabetes", $_POST["
health_condition"])) { echo "checked";} ?>>Diabetes<br>
<input type="checkbox" id="health_condition" name="health_condition[]" value="Hypertension" <?
php if ((!empty($_POST["health_condition"])) && in_array("Hypertension", $_POST["
health_condition"])) { echo "checked";} ?>>Hypertension<br>
<input type="checkbox" id="health_condition" name="health_condition[]" value="Kidney Stones"
<?php if ((!empty($_POST["health_condition"])) && in_array("Kidney Stones", $_POST["
health_condition"])) { echo "checked";} ?>>Kidney Stones<br>
<input type="checkbox" id="health_condition" name="health_condition[]" value="None" <?php if
((!empty($_POST["health_condition"])) && in_array("None", $_POST["health_condition"]))) {
echo "checked";} ?>>None<br>
<!--<td><select id="health_condition" name="health_condition">
<option value="Diabetes" <?php if(isset($_POST["health_condition"])) && $_POST["
health_condition"] == "Diabetes") echo "selected";?>>Diabetes</option>
<option value="Hypertension" <?php if(isset($_POST["health_condition"])) && $_POST["
health_condition"] == "Hypertension") echo "selected";?>>Hypertension</option>
<option value="Kidney Stones" <?php if(isset($_POST["health_condition"])) && $_POST["
health_condition"] == "Kidney Stones") echo "selected";?>>Kidney Stones</option>
<option value="None" <?php if(isset($_POST["health_condition"])) && $_POST["health_condition"]
== "None") echo "selected";?>>None</option>
</select>-->
</tr>
<tr>
<tr>
<td><label for="meal_type">&nbsp;&nbsp;&nbsp; Meal Type:</label>
<td><select id="meal_type" name="meal_type">
<option value="Breakfast" <?php if(isset($_POST["meal_type"])) && $_POST["meal_type"] == "
Breakfast") echo "selected";?>>Breakfast</option>
<option value="Lunch" <?php if(isset($_POST["meal_type"])) && $_POST["meal_type"] == "Lunch")
echo "selected";?>>Lunch</option>
<option value="Dinner" <?php if(isset($_POST["meal_type"])) && $_POST["meal_type"] == "Dinner
") echo "selected";?>>Dinner</option>
<option value="Snack" <?php if(isset($_POST["meal_type"])) && $_POST["meal_type"] == "Snack")
echo "selected";?>>Snack</option>
</select>
</tr>
<tr>
<tr>
<td><label for="userfile">&nbsp;&nbsp;&nbsp; Recipe Image:</label>
<td><input name="userfile" type="file" size="20" >
</tr>

```

```

<tr>
<td><label for="recipe_name">&nbsp; Recipe Name:</label>
<td><input id="recipe_name" name="recipe_name" type="text" value="<?php echo set_value('
recipe_name'); ?>" >
</tr>
<tr>
<td><label for="serving_sugg">&nbsp; Serving Size:</label>
<td><input id="serving_sugg" name="serving_sugg" type="text" value="<?php echo set_value('
serving_sugg'); ?>" size="2" >
</tr>
<tr><td><label for="ingredients">&nbsp; Ingredients:</label>
<td>
<button id="addIngredientButton" name="addIngredientButton" value="" type="button" value="Add
"><span class="ui-icon ui-icon-plusthick"></span></button>
<button id="removeIngredientButton" name="removeIngredientButton" value="Remove" type="button
"><span class="ui-icon ui-icon-minusthick"></span></button><br>
Ingredient: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input id="ingredients0" name="ingredients[0]" class="
ingredients" type="text" value="<?php echo set_value('ingredients'); ?>" ><br>
Food Group: <input id="food_group0" name="food_group[0]" class="food_group" type="text" value
="<?php echo set_value('food_group'); ?>" readonly ><br>
Measure (g): <input id="ingredient_measure0" name="ingredient_measure[0]" class="measure" type
="text" value="<?php echo set_value('ingredient_measure'); ?>" readonly ><br>
Quantity: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input id="ingredient_quantity0" name="
ingredient_quantity[0]" type="text" value="<?php echo set_value('ingredient_quantity');
?>" onchange="update(0)" >
<br>
<div id='TextBoxesGroup'></div>
</tr>
<tr>
<td><label for="preparation_procedure">&nbsp; Procedure:</label>
<td><textarea id="preparation_procedure" name="preparation_procedure" type="text"><?php echo
set_value('preparation_procedure'); ?></textarea>
</tr>
<!--<tr>
<td><label for="total_quantity">&nbsp; Total Quantity:</label>
<td><input id="total_quantity" name="total_quantity" type="text" value="<?php echo set_value
('total_quantity'); ?>">
</tr>-->
<tr>
<td><label for="calories">&nbsp; Total Calories (Kcal):</label>
<td><input id="calories" name="calories" class="calories" type="text" value="<?php echo
set_value('calories'); ?>" readonly>
</tr>
<tr>
<td><label for="carbohydrate">&nbsp; Total Carbohydrates (g):</label>
<td><input id="carbohydrate" name="carbohydrate" type="text" value="<?php echo set_value('
carbohydrate'); ?>" readonly>
</tr>
<tr>
<td><label for="protein">&nbsp; Total Protein (g):</label>
<td><input id="protein" name="protein" type="text" value="<?php echo set_value('protein');
?>" readonly>
</tr>
<tr>
<td><label for="fat">&nbsp; Total Fat (g):</label>
<td><input id="fat" name="fat" type="text" value="<?php echo set_value('fat'); ?>" readonly>
</tr>
<tr>
<td><button id="addRecipeButton" type="submit" name="submit" value="submit">Create Recipe</
button>
<td><button id="addRecipeBackButton" type="button" name="addRecipeBackButton" value="Back"
onclick="window.location.href='<?php echo site_url(); ?>home.cmanager'">Back</button>
<button id="addRecipeHomeButton" type="button" name="addRecipeHomeButton" value="Home" onclick
="window.location.href='<?php echo site_url(); ?>home.cmanager'">Home</button>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url(); ?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url(); ?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url(); ?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url(); ?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
var quantity;
$(function() {
var data2 = [];
var index = 0;

$("#addRecipeButton, #addRecipeBackButton, #addRecipeHomeButton").button();

// Hover states on the static widgets

```

```

$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#record_date" ).datepicker({
clickInput:true,
changeMonth: true, //allow user to select month
changeYear: true, //allow user to select from year range
dateFormat: 'yy-mm-dd',
yearRange: "-100:+0",
onClose: function() {
$( "#record_date" ).valid();
}
});
$( "#addRecipeDialog" ).dialog({
autoOpen: true,
width: 600,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"
});

var id;
$( '.ingredients' ).autocomplete({
minLength: 0,
source: function(request, response) {
id = this.element.context.attributes[name="id"].value;
//console.log(id);
$.ajax({
url: "<?php echo site_url('home_user/suggestions'); ?>",
data: request,
dataType: "json",
type: "POST",
success: function(data) {
//console.log(JSON.stringify(data));
response($.map(data, function(v,i){
if(v.measure != null)
data2[data2.length] = v.measure;
else if(v.food_name != null)
data2[data2.length] = v.food_name;
else if(v.energy != null)
data2[data2.length] = v.energy;
else if(v.carbohydrate != null)
data2[data2.length] = v.carbohydrate;
else if(v.protein != null)
data2[data2.length] = v.protein;
else if(v.fat != null)
data2[data2.length] = v.fat;
else if(v.food_group != null)
data2[data2.length] = v.food_group;

return {
label: v.food_name
};
})));
}
});
},
select: function(event, ui) {
var num = id.substring(id.indexOf("s") + 1, id.length);
var measure = null;
var food_group = null;
var calorie, carbohydrate, protein, fat;
for (var i = 0; i < data2.length; i+=7){
if(data2[i] == ui.item.value) {
measure = data2[i + 1];
calorie = data2[i + 2];
carbohydrate = data2[i + 3];
protein = data2[i + 4];
fat = data2[i + 5];
food_group = data2[i+6];
}
}
//console.log(calorie);
$( '#ingredient_measure'+num ).val (measure);
$( '#food_group'+num ).val (food_group);
/*$( '#calories' ).val(+( $('#calories' ).val() + +calorie);
$( '#carbohydrate' ).val(+( $('#carbohydrate' ).val() + +carbohydrate);
$( '#protein' ).val(+( $('#protein' ).val() + +protein);
$( '#fat' ).val(+( $('#fat' ).val() + +fat);*/
}
/*focus: function( event, ui ) {
$( ".ingredients" ).val( ui.item.label);
},
select: function( event, ui ) {
}*/
});
});

function update(id) {
var request = new XMLHttpRequest(), url = "<?php echo site_url('home_user/get_fa'); ?>",
params = "ingredient="+$( "#ingredients"+id ).val();

```

```

request.onreadystatechange = function() {
if(request.readyState == 4 && request.status == 200) {
var temp = request.responseText.split(","); i;

$('#calories').val($('#calories').val() - (+temp[0] * quantity[id]));
$('#carbohydrate').val($('#carbohydrate').val() - (+temp[1] * quantity[id]));
$('#protein').val($('#protein').val() - (+temp[2] * quantity[id]));
$('#fat').val($('#fat').val() - (+temp[3] * quantity[id]));

for(i = 0; i < temp.length; i++)
temp[i] = +temp[i] * $('#ingredient_quantity'+id).val();

$('#calories').val($('#calories').val() + temp[0]);
$('#carbohydrate').val($('#carbohydrate').val() + temp[1]);
$('#protein').val($('#protein').val() + temp[2]);
$('#fat').val($('#fat').val() + temp[3]);

quantity[id] = $('#ingredient_quantity'+id).val();
}
}
request.open("POST", url, true);
request.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
request.send(params);
}

$(document).ready(function(){
var counter = 1;
var data2 = [];
quantity = new Array();

quantity[0] = $('#ingredient_quantity0').val() == "" ? 0 : $('#ingredient_quantity0').val();

$('#addIngredientButton').click(function () {

/*if(counter>10){
alert("Only 10 textboxes allow");
return false;
}
*/

quantity[counter] = 0;

var newTextBoxDiv = $(document.createElement('div')).attr("id", 'TextBoxDiv' + counter);
newTextBoxDiv.after().html('<br><label>Ingredient: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</label>' +
'<input type="text" class="ingredients" name="ingredients['+counter+']" id="ingredients' +
counter + '" value="" >');

var newTextBoxDiv1= $(document.createElement('div')).attr("id", 'TextBoxDiv1' + counter);
newTextBoxDiv1.after().html('<label>Measure (g): </label>' +
'<input type="text" class="measure" name="ingredient_measure['+counter+']" id="
ingredient_measure' + counter + '" value="" >');

var newTextBoxDiv2= $(document.createElement('div')).attr("id", 'TextBoxDiv2' + counter);
newTextBoxDiv2.after().html('<label>Quantity: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</label>' +
'<input type="text" name="ingredient_quantity['+counter+']" id="ingredient_quantity' + counter
+ '" value="" onchange="update('+counter+')" >');

var newTextBoxDiv3= $(document.createElement('div')).attr("id", 'TextBoxDiv3' + counter);
newTextBoxDiv3.after().html('<label>Food Group: </label>' +
'<input type="text" class="food_group" name="food_group['+counter+']" id="food_group' +
counter + '" value="" >');

newTextBoxDiv.appendTo("#TextboxesGroup");
newTextBoxDiv3.appendTo("#TextboxesGroup");
newTextBoxDiv1.appendTo("#TextboxesGroup");
newTextBoxDiv2.appendTo("#TextboxesGroup");

document.getElementById("ingredient_measure" + counter).readOnly=true;
document.getElementById("food_group" + counter).readOnly=true;

var id;
$('#ingredients').autocomplete({
minLength: 0,
source: function(request, response) {
id = this.element.context.attributes[name="id"].value;
//console.log(id);
$.ajax({
url: "<?php echo site_url('home-user/suggestions'); ?>",
data: request,
dataType: "json",
type: "POST",
success: function(data) {
response($.map(data, function(v,i){
if(v.measure != null)
data2[data2.length] = v.measure;
else if(v.food_name != null)
data2[data2.length] = v.food_name;
else if(v.energy != null)
data2[data2.length] = v.energy;
else if(v.carbohydrate != null)
data2[data2.length] = v.carbohydrate;
else if(v.protein != null)
data2[data2.length] = v.protein;
else if(v.fat != null)
data2[data2.length] = v.fat;
else if(v.food_group !=null)
data2[data2.length] = v.food_group;

return {
label: v.food_name
}
}

}
}
});
}
});

```



```

});
});
});
});
},
select: function(event, ui) {
var num = id.substring(id.indexOf("s") + 1, id.length);
var measure = null;
var food_group = null;
var calorie, carbohydrate, protein, fat;
for (var i = 0; i < data2.length; i+=7){
if (data2[i] == ui.item.value) {
measure = data2[i + 1];
calorie = data2[i + 2];
carbohydrate = data2[i + 3];
protein = data2[i + 4];
fat = data2[i + 5];
food_group = data2[i+6];
}
}
//console.log(calorie);
$('#ingredient_measure'+num).val (measure);
$('#food_group'+num).val (food_group);
/*$('#calories ').val($('#calories ').val() ++calorie);
$('#carbohydrate ').val($('#carbohydrate ').val() ++carbohydrate);
$('#protein ').val($('#protein ').val() ++protein);
$('#fat ').val($('#fat ').val() ++fat);
*/
});
counter++;
});

$("#removeIngredientButton").click(function () {
if (counter==1){
alert("No more textbox to remove");
return false;
}
counter--;

var request = new XMLHttpRequest(), url = "<?php echo site_url('home_user/get_fa '); ?>",
params = "ingredient="+($("#ingredients"+counter).val() == "" ? 0 : ($("#ingredients"+
counter).val()));

request.onreadystatechange = function() {
if (request.readyState == 4 && request.status == 200) {
var temp = request.responseText.split(",");
$("#TextBoxDiv" + counter).remove();
$("#TextBoxDiv3" + counter).remove();
$("#TextBoxDiv1" + counter).remove();
$("#TextBoxDiv2" + counter).remove();

if (quantity[counter] != "" && quantity[counter] != null) {
$('#calories ').val($('#calories ').val() - (+temp[0] * quantity[counter]));
$('#carbohydrate ').val($('#carbohydrate ').val() - (+temp[1] * quantity[counter]));
$('#protein ').val($('#protein ').val() - (+temp[2] * quantity[counter]));
$('#fat ').val($('#fat ').val() - (+temp[3] * quantity[counter]));
}
}
}
request.open("POST", url, true);
request.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
request.send(params);
});
});

function healthConditionsGuidelinesModal() {
$("#healthConditionsGuidelinesModalDialog").dialog({
autoOpen: false,
modal: true,
title: 'Guidelines for Health Conditions',
width: 'auto',
height: 'auto',
buttons: {
"Ok": function () {
$(this).dialog("close");
}
},
}).dialog('open');
return false;
}

function validate() {
var errors = "<p style='color: red; font-weight: bold'>";
if ($("#recipe_name").val().trim() == "")
errors += "The Recipe Name field is required.<br>";
if ($("#serving_sugg").val().trim() == "")
errors += "The Serving Size field is required.<br>";
if ($("#preparation_procedure").val().trim() == "")
errors += "The Procedure field is required.<br>";
if ($("#ingredients0").val().trim() == "")
errors += "The Ingredients field is required.<br>";
var file = $("#input[name='userfile']").val();
if (file != "") {
var filetypes = ["gif", "jpg", "jpeg", "png"];
var ext = file.substr(file.lastIndexOf(".") + 1, file.length).toLowerCase();
if (filetypes.indexOf(ext) < 0)
errors += "The image file type is invalid.<br>";
}
}

```

```

var ingredients = new Array();

$("#ingr_cont input[type='text']").each(function() {
if($(this).attr("id").indexOf("ingredients") > -1 && $(this).val().trim() != "") {
var id = $(this).attr("id").substr($(this).attr("id").indexOf("s") + 1, $(this).attr("id").length);
var value = {ingredient: $(this).val().trim(), foodgroup: $("#food_group"+id).val().trim()};
if($("#ingredient_quantity"+id).val().trim() == "")
errors += "The Quantity field is required.<br>";
else
value.quantity = $("#ingredient_quantity"+id).val().trim();

if(value.quantity != null)
ingredients.push(value);
}
});

var health_condition = $("#health_condition").val(), i;

for (i = 0; i < ingredients.length; i++) {

var food_group = ingredients[i].foodgroup, count = 0, j;

for(j = 0; j < ingredients.length; j++)
if(ingredients[j].foodgroup == food_group)
count += parseFloat(ingredients[j].quantity);

console.log(count+";"+food_group);

if(health_condition == "Diabetes") {
if((food_group == "Vegetable" || food_group == "Fruit") && count > 4) {
errors += "You are only allowed a maximum of 4 servings of "+food_group+" daily.<br>";
} else if(food_group == "Milk" && count > 3) {
errors += "You are only allowed a maximum of 3 servings of Milk daily.<br>";
} else if(food_group == "Rice" && count > 9) {
errors += "You are only allowed a maximum of 9 servings of Rice daily.<br>";
} else if(food_group == "Meat" && count > 6) {
errors += "You are only allowed a maximum of 6 servings of Meat daily.<br>";
} else if(food_group == "Meat" && count > 5) {
errors += "You are only allowed a maximum of 5 servings of Fat daily.<br>";
}
} else if(health_condition == 'Hypertension' || health_condition == 'Kidney Stones') {
if((food_group == "Vegetable" || food_group == "Fruit") && count > 5) {
errors += "You are only allowed a maximum of 5 servings of "+food_group+" daily.<br>";
} else if(food_group == "Milk" && count > 5) {
errors += "You are only allowed a maximum of 5 servings of Milk daily.<br>";
} else if(food_group == "Rice" && count > 8) {
errors += "You are only allowed a maximum of 8 servings of Rice daily.<br>";
} else if(food_group == "Meat" && count > 2) {
errors += "You are only allowed a maximum of 2 servings of Meat daily.<br>";
} else if(food_group == "Meat" && count > 3) {
errors += "You are only allowed a maximum of 3 servings of Fat daily.<br>";
}
}
}

errors += "</p>";
if(errors != "<p style='color: red; font-weight: bold'></p>") {
$("#addRecipeDialog").prepend(errors);
return false;
}
}
</script>
</head>
<body>
<div id="healthConditionsGuidelinesModalDialog" style="display: none;">
<center>
<table>
<tr>
<td>
<p>Below are the maximum daily servings for each food group for each health condition:
</p>
<tr>
<th><h4>Food Group</th>
<th><h4>Diabetes</th>
<th><h4>Hypertension</th>
<th><h4>Kidney Stones</th>
</tr>
<tr>
<td>Vegetable
<td>4
<td>5
<td>5
</tr>
<tr>
<td>Fruit
<td>4
<td>5
<td>5
</tr>
<tr>
<td>Milk
<td>3
<td>5
<td>5
</tr>
<tr>
<td>Rice
<td>9

```



```

<br>
<div id='TextBoxesGroup'></div>
</tr>
<tr>
<td><label for="preparation_procedure">&nbsp;   Procedure:</label>
<td><textarea id="preparation_procedure" name="preparation_procedure" type="text"><?php echo
set_value('preparation_procedure'); ?></textarea>
</tr>
<!--<tr>
<td><label for="total_quantity">&nbsp;   Total Quantity:</label>
<td><input id="total_quantity" name="total_quantity" type="text" value="<?php echo set_value
('total_quantity'); ?>">
</tr-->
<tr>
<td><label for="calories">&nbsp;   Total Calories (Kcal):</label>
<td><input id="calories" name="calories" class="calories" type="text" value="<?php echo
set_value('calories'); ?>" readonly>
</tr>
<tr>
<td><label for="carbohydrate">&nbsp;   Total Carbohydrates (g):</label>
<td><input id="carbohydrate" name="carbohydrate" type="text" value="<?php echo set_value('
carbohydrate'); ?>" readonly>
</tr>
<tr>
<td><label for="protein">&nbsp;   Total Protein (g):</label>
<td><input id="protein" name="protein" type="text" value="<?php echo set_value('protein');
?>" readonly>
</tr>
<tr>
<td><label for="fat">&nbsp;   Total Fat (g):</label>
<td><input id="fat" name="fat" type="text" value="<?php echo set_value('fat'); ?>" readonly>
</tr>
<tr>
<td><button id="addRecipeButton" type="submit" name="submit" value="submit">Create Recipe</
button>
<td><button id="addRecipeBackButton" type="button" name="addRecipeBackButton" value="Back"
onclick="window.location.href='<?php echo site_url(); ?>home_user'">Back</button>
<td><button id="addRecipeHomeButton" type="button" name="addRecipeHomeButton" value="Back" onclick
="window.location.href='<?php echo site_url(); ?>home_user'">Home</button>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#addUserButton, #addUserBackButton, #addUserHomeButton" ).button();
$( "#userTypeRadioButtonSet, #genderRadioButtonSet" ).buttonset();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#birthdate" ).datepicker({
clickInput:true,
changeMonth: true, //allow user to select month
changeYear: true, //allow user to select from year range
dateFormat: 'yy-mm-dd',
yearRange: "-100:+0",
onClose: function() {
$( '#birthdate' ).valid();
if ( $( '#birthdate' ).val() != '' ) {
$( "#birthdate_verify" ).css({ "background-image": "url('<?php echo base_url();?>layout/images/
correct.png')" });
} else {
$( "#birthdate_verify" ).css({ "background-image": "url('<?php echo base_url();?>layout/images/
birthdate_required_error.png'" });
}
}
});
$( "#addUserDialog" ).dialog({
autoOpen: true,

```

```

width: 950,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"

});
});

$(document).ready(function() {
$("#username").keyup(function() {
if ($("#username").val().length >= 4) {
$.ajax({
type: "POST",
url: "<?php echo base_url();>home_admin/check_user",
data: "name="+$("#username").val(),
success: function (msg) {
if (msg=="true") {
$("#usr_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/correct.png')" });
}
else {
$("#usr_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/username_exists_error.png')" });
}
}
});
}
else if ($("#username").val().length < 4) {
$("#usr_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/username_length_error.png')" });
}
else {
$("#usr_verify").css({ "background-image": "none" });
}
});

$("#password").keyup(function(){
if ($("#confirm_password").val().length >= 8) {
if ($("#confirm_password").val() != $("#password").val()) {
$("#confirmpassword_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/password_match_error.png')" });
}
$("#password_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/password_match_error.png')" });
}
pwd=false;
register_show();
}

else {
$("#confirmpassword_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/correct.png')" });
$("#password_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/correct.png')" });
}
}
else if ($("#password").val().length < 8) {
$("#password_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/password_length_error.png')" });
}
else {
$("#password_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/correct.png')" });
}
});

$("#confirm_password").keyup(function() {
if ($("#password").val().length >=8) {
if ($("#confirm_password").val() != $("#password").val()) {
$("#confirmpassword_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/password_match_error.png')" });
}
$("#password_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/password_match_error.png')" });
}
pwd=false;
register_show();
}
else {
$("#confirmpassword_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/correct.png')" });
$("#password_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/correct.png')" });
}
}
});

$("#email_address").keyup(function() {
var email = $("#email_address").val();
if (email != 0) {
if (isValidEmailAddress(email)) {
$.ajax({
type: "POST",
url: "<?php echo base_url();>home_admin/check_email",
data: "email="+$("#email_address").val(),
success: function (msg) {
if (msg=="true") {
$("#email_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/correct.png')" });
}
email_con=true;
register_show();
}
}
});
}
}
});

```

```

else {
$("#email-verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
email_exists_error.png')" });
}
});
}
else {
$("#email-verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
email_invalid_error.png')" });
}
}
else {
$("#email-verify").css({ "background-image": "none" });
}
});

$("#first_name").keyup(function() {
if ($('#first_name').val() != '') {
$("#first_name_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
correct.png')" });
} else {
$("#first_name_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
first_name_required_error.png')" });
}
});

$("#last_name").keyup(function() {
if ($('#last_name').val() != '') {
$("#last_name_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
correct.png')" });
} else {
$("#last_name_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
last_name_required_error.png')" });
}
});
});

function isValidEmailAddress(emailAddress) {
var pattern = new RegExp(/^(("[\w-\s]+")|([\w-]+(?:\.[\w-]+)*)|("[\w-\s]+")([\w-]+(?:\.[\w-]+)
*))@((?![\w-]+\.)*\w[\w-]{0,66})\.([a-z]{2,6}(?!\.[a-z]{2})?)$|(@
\[?(25[0-5]|2[0-4][0-9]|1[0-9]{2}\.|[0-9]{1,2}\.))
|(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})\.
){2}(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})\]?$/i);
return pattern.test(emailAddress);
}

</script>
</head>
<body>
<div id="addUserDialog" title="Add User">
<br>
<font color="red" size="2"><b><?php echo validation_errors(); ?></b></font>
<?php echo form_open('home_admin/save', array('username' => 'myform')); ?>
<center>
<table>
<tr></tr><tr></tr>
<tr>
<td><label for="first_name">&nbsp; First Name:</label>
<td><input id="first_name" name="first_name" type="text" value="<?php echo set_value('
first_name'); ?>">
<span id="first_name_verify" class="verify"></span>
</tr>
<tr>
<td><label for="middle_name">&nbsp; Middle Name:</label>
<td><input id="middle_name" name="middle_name" type="text" value="<?php echo set_value('
middle_name'); ?>">(Optional)
</tr>
<tr>
<td><label for="last_name">&nbsp; Last Name:</label>
<td><input id="last_name" name="last_name" type="text" value="<?php echo set_value('last_name
'); ?>">
<span id="last_name_verify" class="verify"></span>
</tr>
<tr>
<td><label for="birthdate">&nbsp; Birthdate:</label>
<td><input id="birthdate" name="birthdate" type="text" class="date-pick dp-applied" value="<?
php echo set_value('birthdate'); ?>" readonly="readonly">
<a href="#birthdate" class="dp-choose-date" title="Choose date">Choose date</a>
<span id="birthdate_verify" class="verify"></span>
</tr>
<tr>
<td><label for="gender">&nbsp; Gender:</label>
<td>
<div id="genderRadioButtonSet">
<input type="radio" id="genderMale" name="genderRadio" value="Male" <?php echo set_radio('
genderRadio', 'Male', TRUE); ?> <label for="genderMale">Male</label>
<input type="radio" id="genderFemale" name="genderRadio" value="Female" <?php echo set_radio('
genderRadio', 'Female'); ?><label for="genderFemale">Female</label>
</div>
</tr>
<tr>
<td><label for="user_type">&nbsp; User Type:</label>
<td>
<div id="userTypeRadioButtonSet">
<input type="radio" id="userTypeAdministrator" name="userTypeRadio" value="Administrator" <?
php echo set_radio('userTypeRadio', 'Administrator', TRUE); ?><label for="
userTypeAdministrator">Administrator</label>

```

```



```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url(); ?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url(); ?>layout/jquery-ui-1.10.3.custom/css/
        blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url(); ?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
        script>
<script src="<?php echo base_url(); ?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
        .js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
        .min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#editUserButton, #editUserBackButton, #editUserHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#editUserDialog" ).dialog({
autoOpen: true,
width: 650,
modal: true,
closeOnEscape: false,
beforeclose: function( event, ui ) { return false; },
dialogClass: "noclose"

});
});

$(document).ready(function() {

$("#password").keyup(function() {
if ($("#confirm_password").val().length >= 8) {
if ($("#confirm_password").val() != $("#password").val()) {
$("#confirmpassword_verify").css({ "background-image": "url('<?php echo base_url(); ?>layout/
images/password_match_error.png')" });
$("#password_verify").css({ "background-image": "url('<?php echo base_url(); ?>layout/images/
password_match_error.png')" });
pwd=false;
register.show();
}
}

```

```

else {
$("#confirmpassword_verify").css({ "background-image": "url('<?php echo base_url();?>layout/
images/correct.png')" });
$("#password_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
correct.png')" });
}
else if ($("#password").val().length < 8) {
$("#password_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
password_length_error.png')" });
}
else {
$("#password_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
correct.png')" });
}
});

$("#confirm_password").keyup(function() {
if ($("#password").val().length >=8) {
if ($("#confirm_password").val() != ($("#password").val()) {
$("#confirmpassword_verify").css({ "background-image": "url('<?php echo base_url();?>layout/
images/password_match_error.png')" });
$("#password_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
password_match_error.png')" });
pwd=false;
register_show();
}
else {
$("#confirmpassword_verify").css({ "background-image": "url('<?php echo base_url();?>layout/
images/correct.png')" });
$("#password_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
correct.png')" });
}
}
});

</script>
</head>
<body>
<div id="editUserDialog" title="Change Password">
<font color="red" size="2"><b><?php echo validation_errors();?></b></font>

<?php echo form_open("home_admin/edit_password_user/$username");?>
<center>
<table>
<tr></tr><tr></tr>
<tr>
<td><input id="username" name="username" type="text" style="visibility:hidden" value="<?php
echo $username; ?>" readonly="readonly" >
</td>
<td><label for="old_password">&nbsp;   Old Password:</label>
<td><input id="old_password" name="old_password" type="password" >
</td>
<td><label for="password">&nbsp;   Password:</label>
<td><input id="password" name="password" type="password"><span id="password_verify" class="
verify"></span>
</td>
<td><label for="confirm_password">&nbsp;   Confirm Password:</label>
<td><input id="confirm_password" name="confirm_password" type="password"><span id="
confirmpassword_verify" class="verify"></span>
</td>
<td><button id="editUserButton" type="submit" name="submit" value="submit">Change Password</
button>
<td><button id="editUserBackButton" type="button" name="editUserBackButton" value="Back"
onclick="window.location.href='<?php echo site_url(); ?>home_admin'">Back</button>
<td><button id="editUserHomeButton" type="button" name="editUserHomeButton" value="Back" onclick="
window.location.href='<?php echo site_url(); ?>home_admin'">Home</button>
</td>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</body>
</html>

<hr/>

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url();?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url();?>validation/jquery-1.11.0.min.js"></script>
</script>

```





```

<td><input id="confirm_password" name="confirm_password" type="password"><span id="
confirmpassword_verify" class="verify"></span>
</tr>
<tr>
<td><button id="editUserButton" type="submit" name="submit" value="submit">Change Password</
button>
<td><button id="editUserBackButton" type="button" name="editUserBackButton" value="Back"
onclick="window.location.href='<?php echo site_url(); ?>home_cmanager'">Back</button>
<button id="editUserHomeButton" type="button" name="editUserHomeButton" value="Back" onclick="
window.location.href='<?php echo site_url(); ?>home_cmanager'">Home</button>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#editUserButton, #editUserBackButton, #editUserHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#editUserDialog" ).dialog({
autoOpen: true,
width: 650,
modal: true,
closeOnEscape: false,
beforeclose: function( event, ui ) { return false; },
dialogClass: "noclose"
});
});

$(document).ready(function() {

$( "#password" ).keyup(function() {
if ( $( "#confirm_password" ).val().length >= 8 ) {
if ( $( "#confirm_password" ).val() != $( "#password" ).val() ) {
$( "#confirmpassword_verify" ).css( { "background-image": "url('<?php echo base_url();?>layout/
images/password_match_error.png')" } );
$( "#password_verify" ).css( { "background-image": "url('<?php echo base_url();?>layout/images/
password_match_error.png'" } );
pwd=false;
register.show();
}
else {
$( "#confirmpassword_verify" ).css( { "background-image": "url('<?php echo base_url();?>layout/
images/correct.png'" } );
$( "#password_verify" ).css( { "background-image": "url('<?php echo base_url();?>layout/images/
correct.png'" } );
}
}
else if ( $( "#password" ).val().length < 8 ) {
$( "#password_verify" ).css( { "background-image": "url('<?php echo base_url();?>layout/images/
password_length_error.png'" } );
}
else {
$( "#password_verify" ).css( { "background-image": "url('<?php echo base_url();?>layout/images/
correct.png'" } );
}
});

$( "#confirm_password" ).keyup(function() {
if ( $( "#password" ).val().length >= 8 ) {
if ( $( "#confirm_password" ).val() != $( "#password" ).val() ) {
$( "#confirmpassword_verify" ).css( { "background-image": "url('<?php echo base_url();?>layout/
images/password_match_error.png'" } );
$( "#password_verify" ).css( { "background-image": "url('<?php echo base_url();?>layout/images/
password_match_error.png'" } );
pwd=false;
register.show();
}

```

```

}
else {
$("#confirmpassword_verify").css({ "background-image": "url('php echo base_url();?&gt;layout/
images/correct.png')" });
$("#password_verify").css({ "background-image": "url('<?php echo base_url();?&gt;layout/images/
correct.png')" });
}
}
});
});
});

&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div id="editUserDialog" title="Change Password"&gt;
&lt;font color="red" size="2"&gt;&lt;b&gt;&lt;?php echo validation_errors(); ?&gt;&lt;/b&gt;&lt;/font&gt;

&lt;?php echo form_open("home_user/edit_password_user/$username");?&gt;
&lt;center&gt;
&lt;table&gt;
&lt;tr&gt;&lt;/tr&gt;&lt;tr&gt;&lt;/tr&gt;
&lt;tr&gt;
&lt;td&gt;&lt;input id="username" name="username" type="text" style="visibility:hidden" value="&lt;?php
echo $username; ?&gt;" readonly="readonly" &gt;
&lt;/tr&gt;
&lt;tr&gt;
&lt;td&gt;&lt;label for="old_password"&gt;&amp;nbsp;&amp;nbsp;&amp;nbsp; Old Password:&lt;/label&gt;
&lt;td&gt;&lt;input id="old_password" name="old_password" type="password" &gt;
&lt;/tr&gt;
&lt;tr&gt;
&lt;td&gt;&lt;label for="password"&gt;&amp;nbsp;&amp;nbsp;&amp;nbsp; Password:&lt;/label&gt;
&lt;td&gt;&lt;input id="password" name="password" type="password"&gt;&lt;span id="password_verify" class="
verify"&gt;&lt;/span&gt;
&lt;/tr&gt;
&lt;tr&gt;
&lt;td&gt;&lt;label for="confirm_password"&gt;&amp;nbsp;&amp;nbsp;&amp;nbsp; Confirm Password:&lt;/label&gt;
&lt;td&gt;&lt;input id="confirm_password" name="confirm_password" type="password"&gt;&lt;span id="
confirmpassword_verify" class="verify"&gt;&lt;/span&gt;
&lt;/tr&gt;
&lt;tr&gt;
&lt;td&gt;&lt;button id="editUserButton" type="submit" name="submit" value="submit"&gt;Change Password&lt;/
button&gt;
&lt;td&gt;&lt;button id="editUserBackButton" type="button" name="editUserBackButton" value="Back"
onclick="window.location.href='&lt;?php echo site_url(); ?&gt;home_user'"&gt;Back&lt;/button&gt;
&lt;button id="editUserHomeButton" type="button" name="editUserHomeButton" value="Back" onclick="
window.location.href='&lt;?php echo site_url(); ?&gt;home_user'"&gt;Home&lt;/button&gt;
&lt;/tr&gt;
&lt;/table&gt;
&lt;/center&gt;
&lt;?php echo form_close(); ?&gt;
&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$("#editUserButton, #editUserBackButton, #editUserHomeButton").button();
$("#userTypeRadioButtonSet, #genderRadioButtonSet").buttonset();

// Hover states on the static widgets
$("#icons li").hover(
function() {
$(this).addClass("ui-state-hover");
},
function() {
$(this).removeClass("ui-state-hover");
}
);

$("#birthdate").datepicker({
clickInput:true,
changeMonth:true, //allow user to select month
changeYear:true, //allow user to select from year range
dateFormat:'yy-mm-dd',
yearRange:"-100:+0",
onClose: function() {
$("#birthdate").valid();
if ($("#birthdate").val() != '') {
$("#birthdate_verify").css({ "background-image": "url('php echo base_url();?&gt;layout/images/
correct.png')" });
} else {
</pre

```

```

$("#birthdate_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
    birthdate_required_error.png')" });
}
});
$("#editUserDialog").dialog({
    autoOpen: true,
    width: 650,
    modal: true,
    closeOnEscape: false,
    beforeclose: function (event, ui) { return false; },
    dialogClass: "noclose"
});
});
});

$(document).ready(function() {
$("#username").keyup(function() {
if ($("#username").val().length >= 4) {
$.ajax({
type: "POST",
url: "<?php echo base_url();?>home_admin/check_user",
data: "name="+$("#username").val(),
success: function (msg) {
if (msg=="true") {
$("#usr_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/correct
    .png')" });
}
else {
$("#usr_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
    username_exists_error.png')" });
}
});
}
else if ($("#username_reg").val().length < 4) {
$("#usr_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
    username_length_error.png')" });
}
else {
$("#usr_verify").css({ "background-image": "none" });
}
});

$("#email_address").keyup(function() {
var email = ($("#email_address").val());
if (email != 0) {
if (isValidEmailAddress(email)) {
$.ajax({
type: "POST",
url: "<?php echo base_url();?>home_admin/check_email",
data: "email="+$("#email_address").val(),
success: function (msg) {
if (msg=="true") {
$("#email_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
    correct.png')" });
email_con=true;
register_show();
}
else {
$("#email_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
    email_exists_error.png')" });
}
});
}
else {
$("#email_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
    email_invalid_error.png')" });
}
}
else {
$("#email_verify").css({ "background-image": "none" });
}
});

$("#first_name").keyup(function() {
if ($("#first_name").val() != '') {
$("#firstname_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
    correct.png')" });
}
else {
$("#firstname_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
    first_name_required_error.png')" });
}
});

$("#last_name").keyup(function() {
if ($("#last_name").val() != '') {
$("#lastname_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
    correct.png')" });
}
else {
$("#lastname_verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
    last_name_required_error.png')" });
}
});
});

function isValidEmailAddress(emailAddress) {

```





```

    }
  });
} else {
$("#email_verify").css({ "background-image": "url('?php echo base_url();?>layout/images/
email_invalid_error.png')" });
}
} else {
$("#email_verify").css({ "background-image": "none" });
}
});

$("#first_name").keyup(function() {
if ($("#first_name").val() != '') {
$("#firstname_verify").css({ "background-image": "url('?php echo base_url();?>layout/images/
correct.png'" });
} else {
$("#firstname_verify").css({ "background-image": "url('?php echo base_url();?>layout/images/
first_name_required_error.png'" });
}
});

$("#last_name").keyup(function() {
if ($("#last_name").val() != '') {
$("#lastname_verify").css({ "background-image": "url('?php echo base_url();?>layout/images/
correct.png'" });
} else {
$("#lastname_verify").css({ "background-image": "url('?php echo base_url();?>layout/images/
last_name_required_error.png'" });
}
});
});

function isValidEmailAddress(emailAddress) {
var pattern = new RegExp(/^(("[\w-\s]+")|([\w-]+(?:\.[\w-]+)*)|("[\w-\s]+")([\w-]+(?:\.[\w-]+)
*))@((?:(?([\w-]+\.)?[\w-]{0,66})\.([a-z]{2,6})?(?(\.[a-z]{2})?)$)|(@
\[?((25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})\.
(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})\.
{2}(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})\]?$/i);
return pattern.test(emailAddress);
}

</script>
</head>
<body>
<div id="editUserDialog" title="Edit Profile Information">
<font color="red" size="2"><b><?php echo validation_errors(); ?></b></font>

<?php echo form_open("home.cmanager/edit_profile_user/$username");?>
<center>
<table>
<tr></tr><tr></tr>
<tr>
<td style="visibility:hidden">
<div id="userTypeRadioButtonSet">
<input type="radio" id="userTypeAdministrator" name="userTypeRadio" value="Administrator" <?
php if($user_type == "Administrator") echo "checked"; ?><label for="
userTypeAdministrator">Administrator</label>
<input type="radio" id="userTypeContentManager" name="userTypeRadio" value="Content Manager"
<?php if($user_type== "Content Manager") echo "checked"; ?><label for="
userTypeContentManager">Content Manager</label>
<input type="radio" id="userTypeUser" name="userTypeRadio" value="User" <?php if($user_type ==
"User") echo "checked"; ?><label for="userTypeUser">User</label>
</div>
</tr>
<tr>
<td><label for="username">&nbsp;Username:</label>
<td><input id="username" name="username" type="text" value="<?php echo $username; ?>"
readonly="readonly" >
<span id="usr_verify" class="verify"></span>
</tr>
<tr>
<td><label for="first_name">&nbsp;First Name:</label>
<td><input id="first_name" name="first_name" type="text" value="<?php echo $first_name; ?>">
<span id="firstname_verify" class="verify"></span>
</tr>
<tr>
<td><label for="middle_name">&nbsp;Middle Name:</label>
<td><input id="middle_name" name="middle_name" type="text" value="<?php echo $middle_name;
?>">(Optional)
</tr>
<tr>
<td><label for="last_name">&nbsp;Last Name:</label>
<td><input id="last_name" name="last_name" type="text" value="<?php echo $last_name; ?>">
<span id="lastname_verify" class="verify"></span>
</tr>
<tr>
<td><label for="birthdate">&nbsp;Birthdate:</label>
<td><input id="birthdate" name="birthdate" type="text" class="date-pick dp-applied" value="<?
php echo $birthday; ?>" readonly="readonly">
<a href="#birthdate" class="dp-choose-date" title="Choose date">Choose date</a>
<span id="birthdate_verify" class="verify"></span>
</tr>
<tr>
<td><label for="gender">&nbsp;Gender:</label>
<td>
<div id="genderRadioButtonSet">

```

```

<input type="radio" id="genderMale" name="genderRadio" value="Male" <?php if($gender == "Male
    ") echo "checked"; ?><label for="genderMale">Male</label>
<input type="radio" id="genderFemale" name="genderRadio" value="Female" <?php if($gender== "
    Female") echo "checked"; ?><label for="genderFemale">Female</label>
</div>
</tr>
<tr>
<td><label for="email_address">&nbsp; E-mail Address:</label>
<td><input id="email_address" name="email_address" type="text" value="<?php echo
    $email_address; ?>">
<span id="email-verify" class="verify"></span>
</tr>
<tr>
<td><button id="editUserButton" type="submit" name="submit" value="submit">Update Profile </
    button>
<td><button id="editUserBackButton" type="button" name="editUserBackButton" value="Back"
    onclick="window.location.href='<?php echo site_url(); ?>home_cmanager'">Back</button>
<td><button id="editUserHomeButton" type="button" name="editUserHomeButton" value="Back"
    onclick="window.location.href='<?php echo site_url(); ?>home_cmanager'">Home</button>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
    blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
    script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
    .js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
    .min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#editUserButton, #editUserBackButton, #editUserHomeButton" ).button();
$( "#userTypeRadioButtonSet, #genderRadioButtonSet" ).buttonset();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#birthdate" ).datepicker({
clickInput:true,
changeMonth: true, //allow user to select month
changeYear: true, //allow user to select from year range
dateFormat: 'yy-mm-dd',
yearRange: "-100:+0",
onClose: function() {
$( "#birthdate" ).valid();
if ( $( "#birthdate" ).val() != '' ) {
$( "#birthdate_verify" ).css({ "background-image": "url('<?php echo base_url();?>layout/images/
    correct.png')" });
} else {
$( "#birthdate_verify" ).css({ "background-image": "url('<?php echo base_url();?>layout/images/
    birthdate_required_error.png'" });
}
}
});
$( "#editUserDialog" ).dialog({
autoOpen: true,
width: 650,
modal: true,
closeOnEscape: false,
beforeclose: function( event, ui ) { return false; },
dialogClass: "noclose"
});

$(document).ready(function() {
$( "#username" ).keyup(function() {
if ( $( "#username" ).val().length >= 4 ) {
$.ajax({
type: "POST",
url: "<?php echo base_url();?>home_user/check_user",
data: "name="+$( "#username" ).val(),
success: function( msg ) {
if ( msg=="true" ) {
$( "#usr_verify" ).css({ "background-image": "url('<?php echo base_url();?>layout/images/correct
    .png'" });
}
}
}
}
}
}

```



```

else {
$("#usr_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/
username_exists_error.png')" });
}
});
}
else if($("#username_reg").val().length < 4) {
$("#usr_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/
username_length_error.png')" });
}
else {
$("#usr_verify").css({ "background-image": "none" });
}
});

$("#email_address").keyup(function() {
var email = $("#email_address").val();
if (email != 0) {
if (isValidEmailAddress(email)) {
$.ajax({
type: "POST",
url: "<?php echo base_url();>home_user/check_email",
data: "email="+$("#email_address").val(),
success: function(msg) {
if (msg=="true") {
$("#email_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/
correct.png')" });
email_con=true;
register.show();
}
else {
$("#email_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/
email_exists_error.png')" });
}
});
}
else {
$("#email_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/
email_invalid_error.png')" });
}
}
else {
$("#email_verify").css({ "background-image": "none" });
}
});

$("#first_name").keyup(function() {
if ($("#first_name").val() != '') {
$("#first_name_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/
correct.png')" });
} else {
$("#first_name_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/
first_name_required_error.png')" });
}
});

$("#last_name").keyup(function() {
if ($("#last_name").val() != '') {
$("#last_name_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/
correct.png')" });
} else {
$("#last_name_verify").css({ "background-image": "url('<?php echo base_url();>layout/images/
last_name_required_error.png')" });
}
});

function isValidEmailAddress(emailAddress) {
var pattern = new RegExp(/^(([\w-\s]+)|([\w-]+(?:\.[\w-]+)*)|("[\w-\s]+"|([\w-]+(?:\.[\w-]+)
*))|((?![\w-]+\.[\w-]{0,66})\.([a-z]{2,6}|(?!\.[a-z]{2})?)$)|(@
\[?((25[0-5]\.|2[0-4][0-9]\.|1[0-9]{2}\.|[0-9]{1,2}\.))
(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})\.)
\{2}(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})\}\?$/i);
return pattern.test(emailAddress);
}

</script>
</head>
<body>
<div id="editUserDialog" title="Edit Profile Information">
<font color="red" size="2"><b><?php echo validation_errors(); ?></b></font>

<?php echo form_open("home_user/edit_profile_user/$username");?>
<center>
<table>
<tr></tr><tr></tr>
<tr>
<td style="visibility: hidden">
<div id="userTypeRadioButtonSet">
<input type="radio" id="userTypeAdministrator" name="userTypeRadio" value="Administrator" <?
php if($user_type == "Administrator") echo "checked"; ?><label for="
userTypeAdministrator">Administrator</label>
<input type="radio" id="userTypeContentManager" name="userTypeRadio" value="Content Manager"
<?php if($user_type == "Content Manager") echo "checked"; ?><label for="
userTypeContentManager">Content Manager</label>
<input type="radio" id="userTypeUser" name="userTypeRadio" value="User" <?php if($user_type ==
"User") echo "checked"; ?><label for="userTypeUser">User</label>

```

```

</div>
</tr>
<tr>
<td><label for="username">&nbsp; Username:</label>
<td><input id="username" name="username" type="text" value="<?php echo $username; ?>"
        readonly="readonly" >
<span id="usr_verify" class="verify"></span>
</tr>
<tr>
<td><label for="first_name">&nbsp; First Name:</label>
<td><input id="first_name" name="first_name" type="text" value="<?php echo $first_name; ?>">
<span id="firstname_verify" class="verify"></span>
</tr>
<tr>
<td><label for="middle_name">&nbsp; Middle Name:</label>
<td><input id="middle_name" name="middle_name" type="text" value="<?php echo $middle_name;
?>">(Optional)
</tr>
<tr>
<td><label for="last_name">&nbsp; Last Name:</label>
<td><input id="last_name" name="last_name" type="text" value="<?php echo $last_name; ?>">
<span id="lastname_verify" class="verify"></span>
</tr>
<tr>
<td><label for="birthdate">&nbsp; Birthdate:</label>
<td><input id="birthdate" name="birthdate" type="text" class="date-pick dp-applied" value="<?
php echo $birthday; ?>" readonly="readonly">
<a href="#birthdate" class="dp-choose-date" title="Choose date">Choose date</a>
<span id="birthdate_verify" class="verify"></span>
</tr>
<tr>
<td><label for="gender">&nbsp; Gender:</label>
<td>
<div id="genderRadioButtonSet">
<input type="radio" id="genderMale" name="genderRadio" value="Male" <?php if($gender == "Male
") echo "checked"; ?><label for="genderMale">Male</label>
<input type="radio" id="genderFemale" name="genderRadio" value="Female" <?php if($gender== "
Female") echo "checked"; ?><label for="genderFemale">Female</label>
</div>
</tr>
<tr>
<td><label for="email_address">&nbsp; E-mail Address:</label>
<td><input id="email_address" name="email_address" type="text" value="<?php echo
$email_address; ?>">
<span id="email_verify" class="verify"></span>
</tr>
<tr>
<td><button id="editUserButton" type="submit" name="submit" value="submit">Update Profile </
button>
<td><button id="editUserBackButton" type="button" name="editUserBackButton" value="Back"
onclick="window.location.href='<?php echo site_url(); ?>home_user'">Back</button>
<button id="editUserHomeButton" type="button" name="editUserHomeButton" value="Home" onclick="&
window.location.href='<?php echo site_url(); ?>home_user'">Home</button>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url();>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url();>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#editUserButton, #editUserBackButton, #editUserHomeButton" ).button();
$( "#userTypeRadioButtonSet, #genderRadioButtonSet, #statusRadioButtonSet" ).buttonset();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#birthdate" ).datepicker({
clickInput:true,
changeMonth: true, //allow user to select month
changeYear: true, //allow user to select from year range
dateFormat: 'yy-mm-dd',
yearRange: "-100:+0",

```

```

onClose: function() {
$('#birthdate').valid();
if ($('#birthdate').val() != '') {
$('#birthdate_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
yes.png')" });
} else {
$('#birthdate_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
no.png')" });
}
});
$("#editUserDialog").dialog({
autoOpen: true,
width: 650,
modal: true,
closeOnEscape: false,
beforeclose: function(event, ui) { return false; },
dialogClass: "noclose"
});

$(document).ready(function() {
$('#username').keyup(function() {
if ($('#username').val().length >= 4) {
$.ajax({
type: "POST",
url: "<?php echo base_url();>home_admin/check_user",
data: "name="+$('#username').val(),
success: function(msg) {
if (msg=="true") {
$('#usr_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/yes.png
')" });
}
else {
$('#usr_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/no.png
')" });
}
}
});
}
else {
$('#usr_verify').css({ "background-image": "none" });
}
});

$('#email_address').keyup(function() {
var email = $('#email_address').val();
if (email != 0) {
if (isValidEmailAddress(email)) {
$.ajax({
type: "POST",
url: "<?php echo base_url();>home_admin/check_email",
data: "email="+$('#email_address').val(),
success: function(msg) {
if (msg=="true") {
$('#email_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/yes.
png')" });
email_con=true;
register.show();
}
else {
$('#email_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/no.
png')" });
}
}
});
}
else {
$('#email_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/no.
png')" });
}
}
});

$('#first_name').keyup(function() {
if ($('#first_name').val() != '') {
$('#firstname_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
yes.png')" });
} else {
$('#firstname_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
no.png')" });
}
});

$('#last_name').keyup(function() {
if ($('#last_name').val() != '') {
$('#lastname_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
yes.png')" });
} else {
$('#lastname_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/no
.png')" });
}
});
}
});

```



```

<td><button id="editUserBackButton" type="button" name="editUserBackButton" value="Back"
    onclick="window.location.href='<?php echo site_url(); ?>home_admin'">Back</button>
<button id="editUserHomeButton" type="button" name="editUserHomeButton" value="Back" onclick="
    window.location.href='<?php echo site_url(); ?>home_admin'">Home</button>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
    blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
    script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
    .js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
    .min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
$(document).ready(function() {
$("#ad-inner-nav").hide();

$(".ad-top-nav").mouseenter(function() {
$("#ad-inner-nav").fadeIn();
});

$(".ad-top-nav").mouseleave(function() {
$("#ad-inner-nav").fadeOut();
});

$("#viewProfileBoxDialog").dialog({
autoOpen: false,
width: 650,
modal: true,
closeOnEscape: false
});

// Link to open the dialog
$("#viewProfileBoxDialogButton").click(function( event ) {
$("#viewProfileBoxDialog").dialog( "open" );
event.preventDefault();
});
$("#success_msg").fadeOut(5000);

});

$(function() {
$("#logoutButton, #userSearchButton").button();

// Hover states on the static widgets
$("#icons li").hover(
function() {
$(this).addClass( "ui-state-hover" );
},
function() {
$(this).removeClass( "ui-state-hover" );
}
);

$("#search").autocomplete( {
source: function(request, response) {
$.ajax( {
url: "<?php echo site_url('home-admin/suggestions'); ?>",
data: {
term: $("#search").val()
},
dataType: "json",
type: "POST",
success: function(data) {
response(data);
}
});
},
minLength: 1
});
});
</script>
</head>
<body>
<div id="container">

<div id="header"></div>

<div id="body">

<div id="logoutBox">
<ul id="icons" class="ui-widget ui-helper-clearfix ad-top-nav">

```

```

<li id="settingsButton" class="ui-state-default ui-corner-all" title="<?php echo $first_name;
?>">
<span class="ui-icon ui-icon-person"></span><span class="settings-person"><?php echo
$first_name; ?> (<?php echo $user_type; ?>)</span>
<span class="ui-icon ui-icon-triangle-l-s settings-triangle"></span>
</li>
<li id="ad-inner-nav" class="ui-state-default ui-corner-all ad-inner-nav">
<div id="viewProfileBoxDialogButton" class="ad-inner-nav-option" title="Profile Information"><
span class="ui-icon ui-icon-script"></span>Profile Information</div>
<a href="<?php echo site_url(); ?>home_admin/edit_profile_user/<?php echo $username; ?>"><div
class="ad-inner-nav-option" title="Edit Profile"><span class="ui-icon ui-icon-pencil"></
span>Edit Profile</div>
<a href="<?php echo site_url(); ?>home_admin/edit_password_user/<?php echo $username; ?>"><div
class="ad-inner-nav-option" title="Change Password"><span class="ui-icon ui-icon-gear"></
span>Change Password</div>
<a href="<?php echo site_url(); ?>home_admin/logout"><div class="ad-inner-nav-option" title="
Logout"><span class="ui-icon ui-icon-key"></span>Logout</div></a>
</li>
</ul>
</div>
<div id="viewProfileBoxDialog" title="Profile Information">
<table>
<tr>
<td>Name:
<td><?php echo $first_name; ?> <?php echo $last_name; ?>
</td>
</tr>
<tr>
<td>Username:
<td><?php echo $username; ?>
</td>
</tr>
<tr>
<td>User Type:
<td><?php echo $user_type; ?>
</td>
</tr>
<tr>
<td>E-mail Address:
<td><?php echo $email_address; ?>
</td>
</tr>
<tr>
<td>Gender:
<td><?php echo $gender; ?>
</td>
</tr>
<tr>
<td>Birthday:
<td><?php echo $birthday; ?>
</td>
</tr>
</table>
</div>

<div id="manageUserAccountsBox">
<div id="success_msg">
<center><font color="red" size="2"><b><?php echo $this->session->flashdata('success_msg');
?></b></font></center>
</div>
<div id="manageUserAccountsHeading" class="ui-widget-header ui-corner-all">
<center><h2>&nbsp; Manage User Accounts</h2></center>
</div>

<div id="viewUsersBox">
<br>
<?=form_open('home_admin/search');?>
<?php $search = array('name'=>'search', 'id'=>'search', 'value'=>''); ?>
<center><?=form_input($search);?><button id="userSearchButton" type="submit" value="">Search</
button></center>
<?=form_close();?>

<div id="addUserBox">
<a href="<?php echo site_url(); ?>home_admin/add-user">
<p>
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li id="addUserDialogButton" class="ui-state-default ui-corner-all" title="Add User"><span
class="ui-icon ui-icon-plusthick"></span>Add User &nbsp;</li>
</ul>
</p>
</a>
</div>

<?php if (is_array($query) && count($query)) { ?>
<center>
<table>
<tr>
<th style="display:none;">ID</th>
<th>Username</th>
<th>First Name</th>
<th>Middle Name</th>
<th>Last Name</th>
<th>User Type</th>
<th>E-mail Address</th>
<th>Gender</th>
<th>Birthday</th>
<th>Age</th>
<th>Actions</th>
<th>Status</th>
</tr>
<?php foreach($query as $item): ?>
<?php $username= $item->username; ?>
<tr>
<td style="display:none;"><?= $item->user_id ?></td>
<td><?= $item->username ?></td>

```

```

<td><?=$item->first_name ?></td>
<td><?=$item->middle_name ?></td>
<td><?=$item->last_name ?></td>
<td><?=$item->user_type ?></td>
<td><?=$item->email_address ?></td>
<td><?=$item->gender ?></td>
<td><?=$date("Y-M-d", strtotime($item->birthday)); ?></td>
<td><?=$date_diff(date_create($item->birthday), date_create('today'))->y; ?></td>
<td>
<a id="editUserDialogButton" class="editUserDialogButton" href='../home_admin/edit_user/<?php
    echo $username ?>'>
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="Edit">
<span class="ui-icon ui-icon-pencil"></span>
</li>
</ul>
</a>
<a href='../home_admin/delete_user/<?php echo $username; ?>' onclick="return confirm('Are you
    sure you want to delete?')">
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="Delete">
<span class="ui-icon ui-icon-trash"></span>
</li>
</ul>
</a>
</td><?=$item->status ?></td>
</tr>
<?php endforeach;?>
</table>
</center>
<?php } ?>
</div>
</div>
</div>
</div>
<p class="footer">Page rendered in <strong>{elapsed_time}</strong> seconds</p>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url();>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url();>validation/jquery-1.11.0.min.js"></script>
<script>
$(document).ready(function() {
$("#cm-inner-nav").hide();

$(".cm-top-nav").mouseenter(function() {
$("#cm-inner-nav").fadeIn();
});

$(".cm-top-nav").mouseleave(function() {
$("#cm-inner-nav").fadeOut();
});

$("#content-inner-nav").hide();

$(".content-top-nav").mouseenter(function() {
$("#content-inner-nav").fadeIn();
});

$(".content-top-nav").mouseleave(function() {
$("#content-inner-nav").fadeOut();
});

$("#content-content-inner-nav").hide();

$(".content-content-top-nav").mouseenter(function() {
$("#content-content-inner-nav").fadeIn();
});

$(".content-content-top-nav").mouseleave(function() {
$("#content-content-inner-nav").fadeOut();
});

$("#viewProfileBoxDialog").dialog({
    autoOpen: false,
    width: 650,
    modal: true,
    closeOnEscape: false
});

// Link to open the dialog
$("#viewProfileBoxDialogButton").click(function( event ) {
$("#viewProfileBoxDialog").dialog( "open" );
});

```

```

event.preventDefault();
});

$("#success_msg").fadeOut(5000);

});

$(function() {
$("#logoutButton, #userSearchButton").button();

// Hover states on the static widgets
$("#icons li").hover(
function() {
$(this).addClass("ui-state-hover");
},
function() {
$(this).removeClass("ui-state-hover");
}
);
});
</script>
</head>
<body>
<div id="container">

<div id="header"></div>

<div id="body">

<div id="logoutBox">
<ul id="icons" class="ui-widget ui-helper-clearfix cm-top-nav">
<li id="settingsButton" class="ui-state-default ui-corner-all" title="<?php echo $first_name; ?>">
<span class="ui-icon ui-icon-person"></span><span class="settings-person"><?php echo $first_name; ?> (<?php echo $user_type; ?>)</span>
<span class="ui-icon ui-icon-triangle-1-s settings-triangle"></span>
</li>
<li id="cm-inner-nav" class="ui-state-default ui-corner-all cm-inner-nav">
<div id="viewProfileBoxDialogButton" class="cm-inner-nav-option" title="Profile Information"><span class="ui-icon ui-icon-script"></span>Profile Information</div>
<a href="<?php echo site_url(); ?>home_cmanager/edit_profile_user/<?php echo $username; ?>"><div class="cm-inner-nav-option" title="Edit Profile"><span class="ui-icon ui-icon-pencil"></span>Edit Profile</div>
<a href="<?php echo site_url(); ?>home_cmanager/edit_password_user/<?php echo $username; ?>"><div class="cm-inner-nav-option" title="Change Password"><span class="ui-icon ui-icon-gear"></span>Change Password</div>
<a href="<?php echo site_url(); ?>home_cmanager/logout"><div class="cm-inner-nav-option" title="Logout"><span class="ui-icon ui-icon-key"></span>Logout</div></a>
</li>
</ul>
</div>

<div id="viewProfileBoxDialog" title="Profile Information">
<table>
<tr>
<td>Name:
<td><?php echo $first_name; ?> <?php echo $last_name; ?>
</tr>
<tr>
<td>Username:
<td><?php echo $username; ?>
</tr>
<tr>
<td>User Type:
<td><?php echo $user_type; ?>
</tr>
<tr>
<td>E-mail Address:
<td><?php echo $email_address; ?>
</tr>
<tr>
<td>Gender:
<td><?php echo $gender; ?>
</tr>
<tr>
<td>Birthday:
<td><?php echo $birthday; ?>
</tr>
</table>
</div>

<div id="manageContentBox">
<div id="success_msg">
<center><font color="red" size="2"><b><?php echo $this->session->flashdata('success_msg'); ?></b></font></center>
</div>
<div id="manageContentHeading" class="ui-widget-header ui-corner-all">
<center><h2>&nbsp; Manage Content</h2></center>
</div>

<div id="contentMenuBox">
<ul id="icons" class="ui-widget ui-helper-clearfix content-top-nav">
<li id="addContentButton" class="ui-state-default ui-corner-all" title="Add Content">
<span class="ui-icon ui-icon-plusthick"></span><span class="content-plusthick">Add Content</span>
<span class="ui-icon ui-icon-triangle-1-s content-triangle"></span>
</li>
<li id="content-inner-nav" class="ui-state-default ui-corner-all content-inner-nav">

```





```

</table>
</center>
<?php } ?>

<?php if (is_array($query_ht)) { ?>
<center>
<table>
<tr>
<th style="display:none;">ID</th>
<th>Health Tip</th>
<th>Actions</th>
</tr>
<?php foreach($query_ht as $item): ?>
<?php $health_tip_id = $item->health_tip_id; ?>
<tr>
<td style="display:none;"><?= $item->health_tip_id ?></td>
<td><?= $item->health_tip ?></td>
<td>
<a id="editHTDialogButton" class="editHTDialogButton" href='../home.cmanager/edit.ht'<?php
echo $health_tip_id; ?>'>
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="Edit">
<span class="ui-icon ui-icon-pencil"></span>
</li>
</ul>
</a>
<a href='../home.cmanager/delete_health_tip'<?php echo $health_tip_id; ?>" onclick="return
confirm('Are you sure you want to delete?')">
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="Delete">
<span class="ui-icon ui-icon-trash"></span>
</li>
</ul>
</a>
</td>
</tr>
<?php endforeach;?>
</table>
</center>
<?php } ?>
</div>
</div>
</div>
<p class="footer">Page rendered in <strong>{elapsed_time}</strong> seconds</p>
</body>
</html>

```

```

<?php
if($user_type == "Administrator") {
    redirect('home_admin', 'refresh'); //redirect to Administrator Page
}
else if($user_type == "Content Manager") {
    redirect('home.cmanager', 'refresh'); //redirect to Content Manager Page
}
else if($user_type == "User") {
    redirect('home_user', 'refresh'); //redirect to Registered User Page
}
?>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url();?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url();?>validation/jquery-1.11.0.min.js"></script>
<script>
$(document).ready(function() {
$("#us-inner-nav").hide();

$(".us-top-nav").mouseenter(function() {
$("#us-inner-nav").fadeIn();
});

$(".us-top-nav").mouseleave(function() {
$("#us-inner-nav").fadeOut();
});

$("#user-inner-nav").hide();

$(".user-top-nav").mouseenter(function() {
$("#user-inner-nav").fadeIn();
});

```

```

$( ".user-top-nav" ).mouseleave( function () {
    $("#user-inner-nav" ).fadeOut ();
});

$("#user-user-inner-nav" ).hide ();

$( ".user-user-top-nav" ).mouseenter( function () {
    $("#user-user-inner-nav" ).fadeIn ();
});

$( ".user-user-top-nav" ).mouseleave( function () {
    $("#user-user-inner-nav" ).fadeOut ();
});

$("#user-user-user-inner-nav" ).hide ();

$( ".user-user-user-top-nav" ).mouseenter( function () {
    $("#user-user-user-inner-nav" ).fadeIn ();
});

$( ".user-user-user-top-nav" ).mouseleave( function () {
    $("#user-user-user-inner-nav" ).fadeOut ();
});

$("#user-user-user-user-inner-nav" ).hide ();

$( ".user-user-user-user-top-nav" ).mouseenter( function () {
    $("#user-user-user-user-inner-nav" ).fadeIn ();
});

$( ".user-user-user-user-top-nav" ).mouseleave( function () {
    $("#user-user-user-user-inner-nav" ).fadeOut ();
});

$( "#viewProfileBoxDialog" ).dialog({
    autoOpen: false ,
    width: 650,
    modal: true ,
    closeOnEscape: false
});

// Link to open the dialog
$( "#viewProfileBoxDialogButton" ).click( function( event ) {
    $( "#viewProfileBoxDialog" ).dialog( "open" );
    event.preventDefault ();
});

$("#success-msg" ).fadeOut(5000);
});

$( function () {
    $( "#logoutButton, #userSearchButton" ).button ();

    // Hover states on the static widgets
    $( "#icons li" ).hover(
        function () {
            $( this ).addClass( "ui-state-hover" );
        },
        function () {
            $( this ).removeClass( "ui-state-hover" );
        }
    );

    startRefresh ();
    startRefresh1 ();
});

/*      var auto_refresh = setInterval(
function ()
{
    $('#userHealthTipsBox' ).fadeOut( 'slow' ).load( '<?php echo site_url (); ?>/home_user/user_ht_feed' ).fadeIn( "slow" );
}, 10000);*/

function startRefresh () {
    setTimeout( startRefresh, 1000 );
    $.get( '<?php echo site_url (); ?>home_user/user_ht_feed', function( data ) {
        $('#userHealthTipsBox' ).html( data );
    });
}

function startRefresh1 () {
    setTimeout( startRefresh1, 1000 );
    $.get( '<?php echo site_url (); ?>home_user/user_achievements_feed', function( data ) {
        $('#userAchievementsBox' ).html( data );
    });
}
</script>
</head>
<body>
<div id="container">
<div id="header"></div>
<div id="body">
<div id="logoutBox">
<ul id="icons" class="ui-widget ui-helper-clearfix us-top-nav">
<li id="settingsButton" class="ui-state-default ui-corner-all" title="<?php echo $first_name; ?>">

```





```

<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#resetPasswordButton, #resetPasswordBackButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);
});
</script>
</head>
<body>
<div id="container">
<div id="header"></div>
<div id="body">
<center><font color="red" size="2"><b><?php echo validation_errors(); ?></b></font></center>
<div id="resetPasswordBox">
<div id="resetPasswordHeading" class="ui-widget-header ui-corner-all">
<center><h2>&nbsp;&nbsp;  Reset Password</h2></center>
</div>
<?php echo form_open('login/doforget'); ?>
<center>
<table>
<tr></tr><tr></tr>
<tr>
<td><label for="email_address">&nbsp;&nbsp;  E-mail Address:</label>
<td><input id="email_address" name="email_address" type="text" value="<?php echo set_value('
email_address'); ?>">
<span id="email_verify" class="verify"></span>
</tr>
<tr>
<td><button id="resetPasswordButton" type="submit" name="submit" value="submit">Reset Password
</button>
<td><button id="resetPasswordBackButton" type="button" name="resetPasswordBackButton" value="
Back" onclick="window.location.href='<?php echo site_url(); ?>login'">Back</button>
</tr>
<tr></tr><tr></tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</div>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="shortcut icon" type="image/x-icon" href="<?php echo base_url();?>layout/images/
heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#loginButton, #registerButton" ).button();
$( "#userTypeRadioButtonSet, #genderRadioButtonSet" ).buttonset();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);
});

$( "#birthdate" ).datepicker({
clickInput:true,
changeMonth: true,//this option for allowing user to select month
changeYear: true, //this option for allowing user to select from year range
maxDate: '0',
dateFormat: 'yy-mm-dd',
yearRange: "-100:+0",

```

```

onClose: function() {
$('#birthdate').valid();
if ($('#birthdate').val() != '') {
$('#birthdate-verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
correct.png')" });
} else {
$('#birthdate-verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
birthdate_required_error.png')" });
}
});
});

$(document).ready(function() {
$('#username_reg').keyup(function() {
if ($('#username_reg').val().length >= 4) {
$.ajax({
type: "POST",
url: "<?php echo base_url();>index.php/login/check_user",
data: "name="+$('#username_reg').val(),
success: function(msg) {
if (msg=="true") {
$('#usr_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
correct.png')" });
}
else {
$('#usr_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
username_exists_error.png')" });
}
}
});
}
else if ($('#username_reg').val().length < 4) {
$('#usr_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
username_length_error.png')" });
}
else {
$('#usr_verify').css({ "background-image": "none" });
}
});

$('#password_reg').keyup(function() {
if ($('#confirm_password_reg').val().length >= 8) {
if ($('#confirm_password_reg').val() != $('#password_reg').val()) {
$('#confirmpassword_verify').css({ "background-image": "url('<?php echo base_url();>layout/
images/password_match_error.png')" });
}
$('#password_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
password_match_error.png')" });
}
pwd=false;
register_show();
}

else {
$('#confirmpassword_verify').css({ "background-image": "url('<?php echo base_url();>layout/
images/correct.png')" });
$('#password_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
correct.png')" });
}
}
else if ($('#password_reg').val().length < 8) {
$('#password_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
password_length_error.png')" });
}
else {
$('#password_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
correct.png')" });
}
});

$('#confirm_password_reg').keyup(function() {
if ($('#password_reg').val().length >= 8) {
if ($('#confirm_password_reg').val() != $('#password_reg').val()) {
$('#confirmpassword_verify').css({ "background-image": "url('<?php echo base_url();>layout/
images/password_match_error.png')" });
}
$('#password_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
password_match_error.png')" });
}
pwd=false;
register_show();
}
else {
$('#confirmpassword_verify').css({ "background-image": "url('<?php echo base_url();>layout/
images/correct.png')" });
$('#password_verify').css({ "background-image": "url('<?php echo base_url();>layout/images/
correct.png')" });
}
}
});

$('#email_address').keyup(function() {
var email = $('#email_address').val();
if (email != 0) {
if (isValidEmailAddress(email)) {
$.ajax({
type: "POST",
url: "<?php echo base_url();>index.php/login/check_email",
data: "email="+$('#email_address').val(),
success: function(msg) {
if (msg=="true") {

```

```

$("#email-verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
correct.png')" });
email_con=true;
register.show();
}
else {
$("#email-verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
email-exists_error.png')" });
}
});
}
else {
$("#email-verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
email-invalid_error.png')" });
}
}
else {
$("#email-verify").css({ "background-image": "none" });
}
});

$("#first_name").keyup(function() {
if ($("#first_name").val() != '') {
$("#firstname-verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
correct.png')" });
} else {
$("#firstname-verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
first_name_required_error.png')" });
}
});

$("#last_name").keyup(function() {
if ($("#last_name").val() != '') {
$("#lastname-verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
correct.png')" });
} else {
$("#lastname-verify").css({ "background-image": "url('<?php echo base_url();?>layout/images/
last_name_required_error.png')" });
}
});

function isValidEmailAddress(emailAddress) {
var pattern = new RegExp(/^(("[\w-\s]+")|([\w-]+(?:\.[\w-]+)*)|("[\w-\s]+")([\w-]+(?:\.[\w-]+)
*))@((?:(?([\w-]+\.)?[\w-]{0,6})\.([a-z]{2,6}|(?:\.[a-z]{2})?)$)|(@
\[(?((25[0-5]|\.[0-4][0-9]|1[0-9]{2}\.|[0-9]{1,2})\.)
(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})\.
{2})(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})\])?$/i);
return pattern.test(emailAddress);
}

function aboutModal() {
$("#aboutModal").dialog({
autoOpen: false,
modal: true,
title: 'About',
width: 'auto',
height: 'auto',
buttons: {
"Ok": function () {
$(this).dialog("close");
},
},
}).dialog('open');
return false;
}
</script>
</head>
<body>

<div id="container">
<div id="header"></div>

<div id="body">
<center><font color="red" size="2"><b><?php echo validation_errors(); ?></b></font></center>

<div id="loginMainBox">

<div id="aboutModal" style="display:none;" >
<center>
HEAS:FIM (Healthy Eating Alert System: Food Intake Monitor)<br>
is a system that helps monitor fitness and calorie intake<br>
towards a healthier living. It automatically computes the<br>
amount of calories per serving and restricts certain<br>
ingredients based on health conditions.
</center>
</div>

<div id="loginHeading" class="ui-widget-header ui-corner-all">
<center><h2>&nbsp; Login</h2></center>
</div>

<div id="loginBox">
<br>
<?php echo form_open('verifylogin'); ?>
<center>
<table>
<tr></tr>
<tr>

```



```

<td><label for="username">Username:</label>
<td><input id="username" name="username" type="text">
</tr>
<tr>
<td><label for="password">Password:</label>
<td><input id="password" name="password" type="password">
</tr>
<tr></tr>
<tr>
<td><button id="loginButton" type="submit">Login</button>
<td><a href="<?php site_url() ?>login/forget_password">Forgot password</a>
<a href="#" onclick="javascript:aboutModal();" >About</a>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>

<div id="registerHeading" class="ui-widget-header ui-corner-all">
<center><h2>&nbsp;   Register</h2></center>
</div>

<div id="signupBox">
<br>
<?php echo form_open('login/save', array('username' => 'myform')); ?>
<center>
<table>
<tr></tr>
<tr></tr>
<tr>
<td><label for="first_name">&nbsp;   First Name:</label>
<td><input id="first_name" name="first_name" type="text" value="<?php echo set_value('
first_name'); ?>">
<td><span id="firstname_verify" class="verify"></span>
</tr>
<tr>
<td><label for="middle_name">&nbsp;   Middle Name (Optional):</label>
<td><input id="middle_name" name="middle_name" type="text" value="<?php echo set_value('
middle_name'); ?>">
</tr>
<tr>
<td><label for="last_name">&nbsp;   Last Name:</label>
<td><input id="last_name" name="last_name" type="text" value="<?php echo set_value('last_name
'); ?>">
<td><span id="lastname_verify" class="verify"></span>
</tr>
<tr>
<td><label for="birthdate">&nbsp;   Birthdate:</label>
<td><input id="birthdate" name="birthdate" type="text" class="date-pick dp-applied" value="<?
php echo set_value('birthdate'); ?>" readonly="readonly">
<a href="#birthdate" class="dp-choose-date" title="Choose date">Choose date</a>
<td><span id="birthdate_verify" class="verify"></span>
</tr>
<tr>
<td><label for="gender">&nbsp;   Gender:</label>
<td>
<div id="genderRadioButtonSet">
<input type="radio" id="genderMale" name="genderRadio" value="Male" <?php echo set_radio('
genderRadio', 'Male', TRUE); ?><label for="genderMale">Male</label>
<input type="radio" id="genderFemale" name="genderRadio" value="Female" <?php echo set_radio('
genderRadio', 'Female'); ?><label for="genderFemale">Female</label>
</div>
</tr>

<div id="userTypeRadioButtonSet" style="visibility:hidden">
<input type="radio" id="userTypeUser" name="userTypeRadio" value="User" checked><label for="
userTypeUser">User</label>
</div>
<tr>
<td><label for="email_address">&nbsp;   E-mail Address:</label>
<td><input id="email_address" name="email_address" type="text" value="<?php echo set_value('
email_address'); ?>">
<td><span id="email_verify" class="verify"></span>
</tr>
<tr>
<td><label for="username_reg">&nbsp;   Username:</label>
<td><input id="username_reg" name="username_reg" type="text" value="<?php echo set_value('
username_reg'); ?>">
<td><span id="usr_verify" class="verify"></span>
</tr>
<tr>
<td><label for="password_reg">&nbsp;   Password:</label>
<td><input id="password_reg" name="password_reg" type="password" value="<?php echo set_value
('password_reg'); ?>">
<td><span id="password_verify" class="verify"></span>
</tr>
<tr>
<td><label for="confirm_password_reg">&nbsp;   Confirm Password:</label>
<td><input id="confirm_password_reg" name="confirm_password_reg" type="password" value="<?php
echo set_value('confirm_password_reg'); ?>">
<td><span id="confirmpassword_verify" class="verify"></span>
</tr>
<tr>
<td>
<td><?php echo $cap_img ?>
</tr>
<tr>
<td>&nbsp;   Please type the code above:
<td><input type="text" id="captcha" name="captcha" value="" />

```

```

</tr>
<tr>
<td><button id="registerButton" type="submit" name="submit" value="submit">Register</button>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</div>
<p class="footer">Page rendered in <strong>{elapsed_time}</strong> seconds</p>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url();>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url();>validation/jquery-1.11.0.min.js"></script>
<script>

$(function() {

$( "#planDailyMealsBackButton, #planDailyMealsHomeButton, #searchMealButton, #goButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#record_date" ).datepicker({
clickInput:true,
changeMonth: true, //allow user to select month
changeYear: true, //allow user to select from year range
dateFormat: 'yy-mm-dd',
yearRange: "-100:+0",
onClose: function() {
$( '#record_date' ).valid();
}
});

$( "#planDailyMealsDialog" ).dialog({
autoOpen: true,
width: 850,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"
});
$(document).ready(function() {
$( "#success_msg" ).fadeOut(5000);
});
</script>
</head>
<body>
<div id="planDailyMealsDialog" title="Plan Daily Meals">
<br>
<div id="success_msg">
<center><font color="red" size="2"><b><?php echo $this->session->flashdata('success_msg');>
?></b></font></center>
</div>
<font color="red" size="2"><b><?php echo validation_errors(); ?></b></font>
<center>
<table>
<tr>
<th><h3>Add Meals to Daily Meal Plan</th>
<?php echo form_open('home_user/search_meal'); ?>
<td><button id="searchMealButton" type="submit" name="submit" value="submit">Search Meal</button></td>
<button id="planDailyMealsBackButton" type="button" name="planDailyMealsBackButton" value="Back" onclick="window.location.href='<?php echo site_url(); ?>home_user'">Back</button>
<button id="planDailyMealsHomeButton" type="button" name="planDailyMealsHomeButton" value="Home" onclick="window.location.href='<?php echo site_url(); ?>home_user'">Home</button>
<?php echo form_close(); ?>
</tr>
<tr>
<td><?php echo form_open('home_user/select_by_date'); ?>

```

```

<td><input id="record_date" name="record_date" type="text" class="date-pick dp-applied" value
    = "<?php echo set_value('record_date'); ?>" readonly="readonly">
<a href="#record_date" class="dp-choose-date" title="Choose date">Choose date</a>
<button id="goButton" type="submit" name="submit" value="submit">Go</button>
</tr></tr></tr></tr>
</tr>
<?php if (isset($result_display)) { ?>
<?php if ($result_display == 'No record found !') {
echo $result_display;
} else { ?>
<tr></tr></tr></tr></tr></tr></tr></tr>
<tr>
<th><h3>Recipes</th>
</tr>
<tr>
<!--<th>ID</th>-->
<!--<th>Record Date</th>-->
<th>Health Condition</th>
<th>Meal Group</th>
<th>Recipe Name</th>
<th>Calories (Kcal)</th>
<th>Carbohydrates (g)</th>
<th>Protein (g)</th>
<th>Fat (g)</th>
<th>Actions</th>
</tr>
<?php $sum_calories = 0; ?>
<?php $sum_carbohydrates = 0; ?>
<?php $sum_protein = 0; ?>
<?php $sum_fat = 0; ?>
<?php foreach ($result_display as $value) { ?>
<?php
if (isset($value->calories)) {
$calories = $value->calories;
$sum_calories += $calories;
}
?>
<?php
if (isset($value->carbohydrate)) {
$carbohydrates = $value->carbohydrate;
$sum_carbohydrates += $carbohydrates;
}
?>
<?php
if (isset($value->protein)) {
$protein = $value->protein;
$sum_protein += $protein;
}
?>
<?php
if (isset($value->fat)) {
$fat = $value->fat;
$sum_fat += $fat;
}
?>

<?php $calorie_limit = isset($value->calorie_limit) ? $value->calorie_limit : "" ; ?>
<?php $carbohydrate_limit = isset($value->carbohydrate_limit) ? $value->carbohydrate_limit :
    "" ; ?>
<?php $protein_limit = isset($value->protein_limit) ? $value->protein_limit : "" ; ?>
<?php $fat_limit = isset($value->fat_limit) ? $value->fat_limit : "" ; ?>
<?php $health_condition = isset($value->health_condition) ? $value->health_condition : "" ; ?>
<tr>
<!--<td><?php $plan_daily_meal_id = $value->plan_daily_meal_id; echo $plan_daily_meal_id; ?></td>
-->
<!--<td><?php echo $value->record_date; ?></td>-->
<!--<td><?php $own_recipe_id = $value->own_recipe_id; echo $own_recipe_id ?></td>-->
<td><?php if (isset($value->meal_group)): ?>
<td><?php if (isset($value->health_condition)) {
echo $value->health_condition;
}
?></td>
<td><?php if (isset($value->meal_group)) {
echo $value->meal_group;
}
?></td>
<td><?php if (isset($value->recipe_name)) {
echo $value->recipe_name;
}
?></td>
<td><?php if (isset($value->calories)) {
echo $value->calories;
}
?></td>
<td><?php if (isset($value->carbohydrate)) {
echo $value->carbohydrate;
}
?></td>
<td><?php if (isset($value->protein)) {
echo $value->protein;
}
?></td>
<td><?php if (isset($value->fat)) {
echo $value->fat;
}
?></td>
<td>
<a href="../../../home_user/view_created_plan_recipe/<?php echo $own_recipe_id; ?>" >
<ul id="icons" class="ui-widget ui-helper-clearfix">

```

```

<li class="ui-state-default ui-corner-all" title="View Recipe">
<span class="ui-icon ui-icon-folder-open"></span>
</li>
</ul>
</a>
<a href="../home_user/delete_meal/<?php echo $plan_daily_meal_id; ?>" onclick="return confirm
('Are you sure you want to delete?')">
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="Delete">
<span class="ui-icon ui-icon-trash"></span>
</li>
</ul>
</a>
</td>
<?php endif; ?>
<?php } ?>
</tr>
<tr></tr><tr></tr></tr></tr></tr></tr></tr></tr></tr></tr></tr></tr>
<tr>
<th><h3>Recipe Nutritional Information</th>
</tr>
<tr>
<th>Total Calories</th>
<th>Total Carbohydrates</th>
<th>Total Protein</th>
<th>Total Fat</th>
</tr>
<tr>
<td><?php echo $sum_calories; ?>
<td><?php echo $sum_carbohydrates; ?>
<td><?php echo $sum_protein; ?>
<td><?php echo $sum_fat; ?>
</tr>
<tr></tr><tr></tr>
<tr></tr><tr></tr>
<tr></tr><tr></tr>
<tr>
<th><h3>Fitness Information</th>
</tr>
<tr></tr><tr></tr>
<tr></tr><tr></tr>
<tr>
<th>Current Health Condition</th>
</tr>
<tr>
<td><?php echo $health_condition; ?>
</tr>
<tr></tr><tr></tr>
<tr></tr><tr></tr>
<tr>
<th></th>
<th>Limit</th>
<th>Remaining</th>
<th>Consumed</th>
<th>Recommendation</th>
</tr>
<tr>
<td class = "redbox">Calorie
<td><?php echo $calorie_limit; ?>
<td><?php echo $calorie_limit - $sum_calories; ?>
<td><?php echo $sum_calories; ?>
<td>
<?php
if (($calorie_limit - $sum_calories) < 0) {
echo "<font color='red' size='2'><b>Calorie Limit Exceeded!</b></font> You may want to remove
or replace a meal with another meal of lower calorie content.";
}
else if (($calorie_limit - $sum_calories) == 0 || ($calorie_limit - $sum_calories) < 10) {
echo "You are about to exceed your Calorie Limit. You may want to remove or replace a meal
with another meal of lower calorie content.";
}
else if ($sum_calories < $calorie_limit) {
echo "You may want to have more meals.";
}
else {
echo "-";
}
?>
</tr>
<tr>
<td class = "redbox">Carbohydrate
<td><?php echo $carbohydrate_limit; ?>
<td><?php echo $carbohydrate_limit - $sum_carbohydrates; ?>
<td><?php echo $sum_carbohydrates; ?>
<td>
<?php
if ($carbohydrate_limit - $sum_carbohydrates) < 0) {
echo "<font color='red' size='2'><b>Carbohydrate Limit Exceeded!</b></font> You may want to
remove or replace a meal with another meal of lower carbohydrate content.";
}
else if (($carbohydrate_limit - $sum_carbohydrates) == 0 || ($carbohydrate_limit -
$sum_carbohydrates) < 10) {
echo "You are about to exceed your Carbohydrate Limit. You may want to remove or replace a
meal with another meal of lower carbohydrate content.";
}
else if ($sum_carbohydrates < $carbohydrate_limit) {
echo "You may want to have more meals.";
}
else {

```

```

echo "-";
}
?>
</tr>
<tr>
<td class = "redbox">Protein
<td><?php echo $protein_limit; ?>
<td><?php echo $protein_limit - $sum_protein; ?>
<td><?php echo $sum_protein; ?>
<td>
<?php
if (($protein_limit - $sum_protein) < 0) {
echo "<font color='red' size='2'><b>Protein Limit Exceeded!</b></font> You may want to remove
or replace a meal with another meal of lower protein content.";
}
else if (($protein_limit - $sum_protein) == 0 || ($protein_limit - $sum_protein) < 10) {
echo "You are about to exceed your Protein Limit. You may want to remove or replace a meal
with another meal of lower protein content.";
}
else if ($sum_protein < $protein_limit) {
echo "You may want to have more meals.";
}
else {
echo "-";
}
}
?>
</tr>
<tr>
<td class = "redbox">Fat
<td><?php echo $fat_limit; ?>
<td><?php echo $fat_limit - $sum_fat; ?>
<td><?php echo $sum_fat; ?>
<td>
<?php
if (($fat_limit - $sum_fat) < 0) {
echo "<font color='red' size='2'><b>Fat Limit Exceeded!</b></font> You may want to remove or
replace a meal with another meal of lower fat content.";
}
else if (($fat_limit - $sum_fat) == 0 || ($fat_limit - $sum_fat) < 10) {
echo "You are about to exceed your Fat Limit. You may want to remove or replace a meal with
another meal of lower fat content.";
}
else if ($sum_fat < $fat_limit) {
echo "You may want to have more meals.";
}
else {
echo "-";
}
}
?>
</tr>
<?php } ?>
<?php } ?>
<?php echo form_close(); ?>
</table>
</center>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url();?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url();?>validation/jquery-1.11.0.min.js"></script>
</script>

$(function() {

$( "#recommendDailyMealsBackButton, #recommendDailyMealsHomeButton, #searchMealButton, #goButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#record_date" ).datepicker({
clickInput:true,
changeMonth: true, //allow user to select month
changeYear: true, //allow user to select from year range
dateFormat: 'yy-mm-dd',
yearRange: "-100:+0",

```

```

onClose: function() {
$('#record_date').valid();
};
}");
$("#planDailyMealsDialog").dialog({
autoOpen: true,
width: 850,
modal: true,
closeOnEscape: false,
beforeclose: function(event, ui) { return false; },
dialogClass: "noclose"
});
});

$(document).ready(function() {
$("#success_msg").fadeOut(5000);
});
</script>
</head>
<body>
<div id="planDailyMealsDialog" title="Recommend Daily Meals">
<br>
<div id="success_msg">
<center><font color="red" size="2"><b><?php echo $this->session->flashdata('success_msg');
?></b></font></center>
</div>
<font color="red" size="2"><b><?php echo validation_errors(); ?></b></font>
<center>
<table>
<tr>
<th><h3>Add Meals to Recommended Daily Meals</th>
<?php echo form_open('home_cmanager/search_meal'); ?>
<td><button id="searchMealButton" type="submit" name="submit" value="submit">Search Meal</
button></td>
<button id="recommendDailyMealsBackButton" type="button" name="recommendDailyMealsBackButton"
value="Back" onclick="window.location.href='<?php echo site_url(); ?>home_cmanager'">Back
</button>
<button id="recommendDailyMealsHomeButton" type="button" name="recommendDailyMealsHomeButton"
value="Home" onclick="window.location.href='<?php echo site_url(); ?>home_cmanager'">Home
</button>
<?php echo form_close(); ?>
</tr>
<?php echo form_open('home_cmanager/select_by_date'); ?>
<tr>
<th><h3>Search Recommended Daily Meals</th>
<h5><th><label for="record_date">&nbsp;   Date:</label></th>
<td><input id="record_date" name="record_date" type="text" class="date-pick dp-applied" value
="<?php echo set_value('record_date'); ?>" readonly="readonly">
<a href="#record_date" class="dp-choose-date" title="Choose date">Choose date</a>
<button id="goButton" type="submit" name="submit" value="submit">Go</button>
<tr></tr><tr></tr>
</tr>
<?php if (isset($result_display)) { ?>
<?php if ($result_display == 'No record found !') {
echo $result_display;
} else { ?>
<tr>
<th><h3>Recipes</th>
</tr>
<tr>
<!--<th>ID</th-->
<!--<th>Record Date</th-->
<th>Health Condition</th>
<th>Meal Group</th>
<th>Recipe Name</th>
<th>Calories (Kcal)</th>
<th>Carbohydrates (g)</th>
<th>Protein (g)</th>
<th>Fat (g)</th>
<th>Actions</th>
</tr>
<?php $sum_calories = 0; ?>
<?php $sum_carbohydrates = 0; ?>
<?php $sum_protein = 0; ?>
<?php $sum_fat = 0; ?>
<?php foreach ($result_display as $value) { ?>
<?php
if(isset($value->calories)) {
$calories = $value->calories;
$sum_calories += $calories;
}
?>
<?php
if(isset($value->carbohydrate)) {
$carbohydrates = $value->carbohydrate;
$sum_carbohydrates += $carbohydrates;
}
?>
<?php
if(isset($value->protein)) {
$protein = $value->protein;
$sum_protein += $protein;
}
?>
<?php
if(isset($value->fat)) {
$fat = $value->fat;
$sum_fat += $fat;
}
}
}

```

```

?>
<tr>
<!--<td><?php $recommended_daily_meal_id = $value->recommended_daily_meal_id; echo
    $recommended_daily_meal_id; ?></td>-->
<!--<td><?php $recommended_recipe_id = $value->recommended_recipe_id; echo
    $recommended_recipe_id; ?></td>-->
<!--<td><?php echo $value->record_date; ?></td>-->
<td><?php if(isset($value->health_condition)) {
echo $value->health_condition;
}
?></td>
<td><?php if(isset($value->meal_group)) {
echo $value->meal_group;
}
?></td>
<td><?php if(isset($value->recipe_name)) {
echo $value->recipe_name;
}
?></td>
<td><?php if(isset($value->calories)) {
echo $value->calories;
}
?></td>
<td><?php if(isset($value->carbohydrate)) {
echo $value->carbohydrate;
}
?></td>
<td><?php if(isset($value->protein)) {
echo $value->protein;
}
?></td>
<td><?php if(isset($value->fat)) {
echo $value->fat;
}
?></td>
<td>
<a href="..../home_cmanager/view_created_recommended_recipe/<?php echo $recommended_recipe_id;
?>" >
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="View Recipe">
<span class="ui-icon ui-icon-folder-open"></span>
</li>
</ul>
</a>
<a href="..../home_cmanager/delete_meal/<?php echo $recommended_daily_meal_id; ?>" onclick="
return confirm('Are you sure you want to delete?')">
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="Delete">
<span class="ui-icon ui-icon-trash"></span>
</li>
</ul>
</a>
</td>
<?php } ?>
</tr>
<tr></tr><tr></tr><tr></tr><tr></tr><tr></tr><tr></tr></tr>
<tr>
<th><h3>Recipe Nutritional Information</th>
</tr>
<tr>
<th>Total Calories</th>
<th>Total Carbohydrates</th>
<th>Total Protein</th>
<th>Total Fat</th>
</tr>
<tr>
<td><?php echo $sum_calories; ?>
<td><?php echo $sum_carbohydrates; ?>
<td><?php echo $sum_protein; ?>
<td><?php echo $sum_fat; ?>
</tr>
<?php } ?>
<?php } ?>
<?php echo form_close(); ?>
</table>
</center>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url();>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url();>validation/jquery-1.11.0.min.js"></script>
</script>

```

```

$(function() {

$( "#addMealBackButton, #addMealHomeButton, #addMealButton, #userSearchMealButton, #
addMealButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#record_date" ).datepicker({
clickInput:true,
changeMonth: true, //allow user to select month
changeYear: true, //allow user to select from year range
dateFormat: 'yy-mm-dd',
yearRange: "-100:+0",
onClose: function() {
$( "#record_date" ).valid();
}
});

$( "#addMealDialog" ).dialog({
autoOpen: true,
width: 850,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"
});
$(document).ready(function() {
$( "#success_msg" ).fadeOut(5000);
});
</script>
</head>
<body>
<div id="addMealDialog" title="Add Meal">
<br>
<div id="success_msg">
<center><font color="red" size="2"><b><?php echo $this->session->flashdata('success_msg');
?></b></font></center>
</div>
<!--<?php echo $record_date; ?>-->
<?=form_open('home_cmanager/search_meal');?>
<?php $search_meal = array('name'=>'search_meal', 'id'=>'search_meal', 'value'=>''); ?>
<center>Search for Recipe : <?=form_input($search_meal);?>
<button id="userSearchMealButton" type="submit" value="" title="Search Meal">Search</button>
<button id="addMealBackButton" type="button" name="addMealBackButton" value="Back" onclick="
window.location.href='<?php echo site_url(); ?>home_cmanager/recommend_daily_meals'">Back
</button>
<button id="addMealHomeButton" type="button" name="addMealHomeButton" value="Home" onclick="
window.location.href='<?php echo site_url(); ?>home_cmanager'">Home</button>
</center>
<?=form_close();?>

<?php if (is_array($query_meal)) { ?>
<center>
<?php if (!empty($query_meal)): ?>
<table>
<tr>
<th style="display:none;">ID</th>
<th>Health Condition</th>
<th>Meal Type</th>
<th>Recipe Name</th>
<th>Serving Size</th>
<th>Calories (Kcal)</th>
<th>Carbohydrate (g)</th>
<th>Protein (g)</th>
<th>Fat (g)</th>
<th>Actions</th>
</tr>
<?php foreach($query_meal as $item): ?>
<?php $recommended_recipe_id = $item->recommended_recipe_id ?>
<tr>
<td style="display:none;"><?= $item->recommended_recipe_id ?></td>
<td><?= $item->health_condition ?></td>
<td><?= $item->meal_type ?></td>
<td><?= $item->recipe_name ?></td>
<td><?= $item->servicing_sugg ?></td>
<td><?= $item->calories ?></td>
<td><?= $item->carbohydrate ?></td>
<td><?= $item->protein ?></td>
<td><?= $item->fat ?></td>
<td>
<a href="..../home_cmanager/view_created_recipe_meal/<?php echo $recommended_recipe_id; ?>" >
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="View Recipe">
<span class="ui-icon ui-icon-folder-open"></span>
</li>
</ul>
</a>
<a href="..../home_cmanager/add_meal/<?php echo $recommended_recipe_id; ?>">
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="Add Meal">

```



```

<span class="ui-icon ui-icon-plusthick"></span>
</li>
</ul>
</a>
</tr>
<?php endforeach;?>
</table>
<?php else: ?>
<?php echo "No data to display."; ?>
<?php endif; ?>
</center>
<?php } ?>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
</script>

$(function() {

$( "#addMealBackButton, #addMealHomeButton, #addMealButton, #userSearchMealButton, #addMealButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#record_date" ).datepicker({
clickInput:true,
changeMonth: true, //allow user to select month
changeYear: true, //allow user to select from year range
dateFormat: 'yy-mm-dd',
yearRange: "-100:+0",
onClose: function() {
$( "#record_date" ).valid();
}
});

$( "#addMealDialog" ).dialog({
autoOpen: true,
width: 850,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"
});

$(document).ready(function() {
$( "#success_msg" ).fadeOut(5000);
});
</script>
</head>
<body>
<div id="addMealDialog" title="Add Meal">
<br>
<div id="success_msg">
<center><font color="red" size="2"><b><?php echo $this->session->flashdata('success_msg'); ?></b></font></center>
</div>
<!--<?php echo $record_date; ?>-->
<?form_open('home-user/search_meal');?>
<?php $search_meal = array('name'=>'search_meal', 'id'=>'search_meal', 'value'=>''); ?>
<center>Search for Recipe : <?form_input($search_meal);?>
<button id="userSearchMealButton" type="submit" value="" title="Search Meal">Search</button>
<button id="addMealBackButton" type="button" name="addMealBackButton" value="Back" onclick="window.location.href='<?php echo site_url(); ?>home-user/plan_daily_meals'">Back</button>
<button id="addMealHomeButton" type="button" name="addMealHomeButton" value="Back" onclick="window.location.href='<?php echo site_url(); ?>home-user'">Home</button>
</center>
<?form_close();?>

<?php if (is_array($query_meal)) { ?>
<center>
<?php if (!empty($query_meal)): ?>
<table>

```

```

<tr>
<th style="display:none;">ID</th>
<th>Health Condition</th>
<th>Meal Type</th>
<th>Recipe Name</th>
<th>Serving Size</th>
<th>Calories (Kcal)</th>
<th>Carbohydrate (g)</th>
<th>Protein (g)</th>
<th>Fat (g)</th>
<th>Actions</th>
</tr>
<?php foreach($query_meal as $item): ?>
<?php $own_recipe_id = $item->own_recipe_id ?>
<tr>
<td style="display:none;"><?= $item->own_recipe_id ?></td>
<td><?= $item->health_condition ?></td>
<td><?= $item->meal_type ?></td>
<td><?= $item->recipe_name ?></td>
<td><?= $item->serving_sugg ?></td>
<td><?= $item->calories ?></td>
<td><?= $item->carbohydrate ?></td>
<td><?= $item->protein ?></td>
<td><?= $item->fat ?></td>
<td>
<a href="../../home_user/view_created_recipe_meal/<?php echo $own_recipe_id; ?>" >
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="View Recipe">
<span class="ui-icon ui-icon-folder-open"></span>
</li>
</ul>
</a>
<a href="../../home_user/add_meal/<?php echo $own_recipe_id; ?>">
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="Add Meal">
<span class="ui-icon ui-icon-plusthick"></span>
</li>
</ul>
</a>
</td>
</tr>
<?php endforeach;?>
</table>
<?php else: ?>
<?php echo "No data to display."; ?>
<?php endif; ?>
</center>
<?php } ?>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#userSearchFELButton, #userSearchFELBackButton, #userSearchFELHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#userSearchFELDialog" ).dialog({
autoOpen: true,
width: 925,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"
});
});
</script>
</head>
<body>
<div id="userSearchFELDialog" title="Food Nutritional Information">
<br>
<?form_open('home_user/search_fel');?>

```

```

<?php $search_fel = array('name'=>'search_fel','id'=>'search_fel','value'=>''); ?>
<center>Food Nutritional Information: <?form_input($search_fel);?>
<button id="userSearchFELButton" type="submit" value="" title="Search Food Nutritional
Information">Search</button>
<button id="userSearchFELBackButton" type="button" name="userSearchFELBackButton" value="Back"
onclick="window.location.href='<?php echo site_url(); ?>home_user'">Back</button>
<button id="userSearchFELHomeButton" type="button" name="userSearchFELHomeButton" value="Home"
onclick="window.location.href='<?php echo site_url(); ?>home_user'">Home</button>
</center>
<?form_close();?>

<?php if (is_array($query_fel)) { ?>
<center>
<?php if (!empty($query_fel)): ?>
<table>
<tr>
<th>List Group</th>
<th>Food Group</th>
<th>Food Group Category</th>
<th>Food Name</th>
<th>Measure</th>
<th>Carbohydrate (g)</th>
<th>Protein (g)</th>
<th>Fat (g)</th>
<th>Energy (Kcal)</th>
<th>Weight as Purchased (g)</th>
<th>Weight Edible Portion (g)</th>
</tr>
<?php foreach($query_fel as $item): ?>
<tr>
<td><?=$item->list_group ?></td>
<td><?=$item->food_group ?></td>
<td><?=$item->food_group_category ?></td>
<td><?=$item->food_name ?></td>
<td><?=$item->measure ?></td>
<td><?=$item->carbohydrate ?></td>
<td><?=$item->protein ?></td>
<td><?=$item->fat ?></td>
<td><?=$item->energy ?></td>
<td><?=$item->weight_as_purchased ?></td>
<td><?=$item->weight_edible_portion ?></td>
</tr>
<?php endforeach;?>
</table>
<?php else: ?>
<?php echo "No data to display."; ?>
<?php endif; ?>
</center>
<?php } ?>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url();?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url();?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#shareAchievementButton, #shareAchievementBackButton, #shareAchievementHomeButton" ).button(
() );

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#shareAchievementDialog" ).dialog({
autoOpen: true,
width: 485,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"
});
});
</script>
</head>
<body>

```

```

<div id="shareAchievementDialog" title="Share Achievement">
<br>
<font color="red" size="2"><b><?php echo validation_errors(); ?></b></font>
<?php echo form_open('home-user/post_facebook'); ?>
<center>
<table>
<tr></tr><tr></tr>
<tr>
<td><textarea rows="4" cols="40" id="achievement" name="achievement" type="longtext" value
="<?php echo set_value('achievement'); ?>"></textarea>
</tr>
<tr>
<td><button id="shareAchievementButton" type="submit" name="submit" value="submit">Share
Achievement</button>
<button id="shareAchievementBackButton" type="button" name="shareAchievementBackButton" value
="Back" onclick="window.location.href='<?php echo site_url(); ?>home-user'">Back</button>
<button id="shareAchievementHomeButton" type="button" name="shareAchievementHomeButton" value
="Back" onclick="window.location.href='<?php echo site_url(); ?>home-user'">Home</button>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url();?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url();?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#shareAchievementButton, #shareAchievementBackButton, #shareAchievementHomeButton" ).button
() );

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#shareAchievementDialog" ).dialog({
autoOpen: true,
width: 485,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"

});
});
</script>
</head>
<body>
<div id="shareAchievementDialog" title="Share Achievement">
<br>
<font color="red" size="2"><b><?php echo validation_errors(); ?></b></font>
<?php echo form_open('home-user/share_achievement_save'); ?>
<center>
<table>
<tr></tr><tr></tr>
<tr>
<td><textarea rows="4" cols="40" id="achievement" name="achievement" type="longtext" value
="<?php echo set_value('achievement'); ?>" ></textarea>
</tr>
<tr>
<td><button id="shareAchievementButton" type="submit" name="submit" value="submit">Share
Achievement</button>
<button id="shareAchievementBackButton" type="button" name="shareAchievementBackButton" value
="Back" onclick="window.location.href='<?php echo site_url(); ?>home-user'">Back</button>
<button id="shareAchievementHomeButton" type="button" name="shareAchievementHomeButton" value
="Home" onclick="window.location.href='<?php echo site_url(); ?>home-user'">Home</button>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {
$( "#shareAchievementButton , #shareAchievementBackButton , #shareAchievementHomeButton" ).button
() );

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#shareAchievementDialog" ).dialog({
autoOpen: true ,
width: 485 ,
modal: true ,
closeOnEscape: false ,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"

});
});
</script>
</head>
<body>
<div id="shareAchievementDialog" title="Share Achievement">
<br>
<font color="red" size="2"><b><?php echo validation_errors(); ?></b></font>
<?php echo form_open('home_user/post_twitter'); ?>
<center>
<table>
<tr></tr><tr></tr>
<tr>
<td><textarea rows="4" cols="40" id="achievement" name="achievement" type="longtext" value
="<?php echo set_value('achievement'); ?>" ></textarea>
</tr>
<tr>
<td><button id="shareAchievementButton" type="submit" name="submit" value="submit">Share
Achievement</button>
<td><button id="shareAchievementBackButton" type="button" name="shareAchievementBackButton"
value="Back" onclick="window.location.href='<?php echo site_url(); ?>home_user'">Back</
button>
<button id="shareAchievementHomeButton" type="button" name="shareAchievementHomeButton" value
="Home" onclick="window.location.href='<?php echo site_url(); ?>home_user'">Home</button>
</tr>
</table>
</center>
<?php echo form_close(); ?>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script>
$(function() {

$( "#planDailyMealsBackButton , #planDailyMealsHomeButton , #searchMealButton , #goButton" ).
button() );

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {

```



```

<tr>
<!--<td><?php $plan_daily_meal_id = $value->plan_daily_meal_id; echo $plan_daily_meal_id; ?></
td>-->
<!--<td><?php echo $value->record_date; ?></td>-->
<!--<td><?php $recommended_recipe_id = $value->recommended_recipe_id; echo
    $recommended_recipe_id ?></td>-->
<td><?php if(isset($value->health_condition)) {
echo $value->health_condition;
}
?></td>
<td><?php if(isset($value->meal_group)) {
echo $value->meal_group;
}
?></td>
<td><?php if(isset($value->recipe_name)) {
echo $value->recipe_name;
}
?></td>
<td><?php if(isset($value->calories)) {
echo $value->calories;
}
?></td>
<td><?php if(isset($value->carbohydrate)) {
echo $value->carbohydrate;
}
?></td>
<td><?php if(isset($value->protein)) {
echo $value->protein;
}
?></td>
<td><?php if(isset($value->fat)) {
echo $value->fat;
}
?></td>
<td>
<a href="../../home_user/view_created_recommended_recipe/<?php echo $recommended_recipe_id; ?>" >
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="View Recipe">
<span class="ui-icon ui-icon-folder-open"></span>
</li>
</ul>
</a>
</td>
<?php } ?>
</tr>
<tr>
<th><h3>Information</h3>
</tr>
<tr>
<h5><th><label for="total_calories">&nbsp; Total Calories:</label></th>
<td><?php echo $sum_calories; ?>
</tr>
<tr>
<h5><th><label for="total_carbohydrates">&nbsp; Total Carbohydrates:</label></th>
<td><?php echo $sum_carbohydrates; ?>
</tr>
<tr>
<h5><th><label for="total_protein">&nbsp; Total Protein:</label></th>
<td><?php echo $sum_protein; ?>
</tr>
<tr>
<h5><th><label for="total_fat">&nbsp; Total Fat:</label></th>
<td><?php echo $sum_fat; ?>
</tr>
<?php } ?>
<?php } ?>
<?php echo form_close(); ?>
</table>
</center>
</div>
</body>
</html>

```

---

```

<?php if (is_array($achievement_feed)) { ?>
    <center>
        <?php foreach($achievement_feed as $item): ?>
            <?= $item->achievement ?><br><br>
        <?php endforeach;?>
    </center>
</?php } ?>

```

---

```

<?php if (is_array($ht_feed)) { ?>
    <center>
        <?php foreach($ht_feed as $item): ?>
            <?= $item->health_tip ?><br><br>
        <?php endforeach;?>
        <!--<tr>
            <?php if(isset($pagination)) {
                echo $pagination;
            } ?>
        </tr>-->
    </center>
</?php } ?>

```

---

```

<html lang="en">

```

```

<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url(); ?>chart/highcharts.js"></script>
<script src="<?php echo base_url(); ?>chart/exporting.js"></script>
<script>
$(function() {
$( "#viewCreatedRecipesBackButton, #viewCreatedRecipesHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#viewCreatedRecipesDialog" ).dialog({
autoOpen: true,
width: 950,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"
});
});
</script>
</head>
<body>
<div id="viewCreatedRecipesDialog" title="Created Recipes">
<br>
<?php if ( is_array( $query ) ) { ?>
<center>
<table>
<tr>
<th>ID</th>
<th>Health Condition</th>
<th>Meal Type</th>
<th>Recipe Name</th>
<th>Serving Size</th>
<th>Total Calories (Kcal)</th>
<th>Total Carbohydrate (g)</th>
<th>Total Protein (g)</th>
<th>Total Fat (g)</th>
<th>Actions</th>
</tr>
<?php foreach( $query as $item ): ?>
<?php $own_recipe_id = $item->own_recipe_id ?>
<tr>
<td><?=$ item->own_recipe_id ?></td>
</tr>
<?php endforeach;?>
</table>
</center>
<?php } ?>
<br>
<button id="viewCreatedRecipesBackButton" type="button" name="viewCreatedRecipesBackButton" value="Back" onclick="window.location.href='<?php echo site_url(); ?>home.user'">Back</button>
<button id="viewCreatedRecipesHomeButton" type="button" name="viewCreatedRecipesHomeButton" value="Home" onclick="window.location.href='<?php echo site_url(); ?>home.user'">Home</button>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url(); ?>chart/highcharts.js"></script>

```



```

<script src="<?php echo base_url(); ?>chart/exporting.js">/script>
<script>
$(function() {
$( "#viewCreatedRecipesBackButton, #viewCreatedRecipesHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#viewCreatedRecipesDialog" ).dialog({
autoOpen: true,
width: 950,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"

});
});
$(document).ready(function() {
$("#success_msg").fadeOut(5000);
});
</script>
</head>
<body>
<div id="viewCreatedRecipesDialog" title="Created Recipes">
<br>
<div id="success_msg">
<center><font color="red" size="2"><b><?php echo $this->session->flashdata('success_msg');
?></b></font></center>
</div>
<?php if (is_array($query)) { ?>
<center>
<?php if (!empty($query)): ?>
<table>
<tr>
<th style="display:none;">ID</th>
<th>Health Condition</th>
<th>Meal Type</th>
<th>Recipe Name</th>
<th>Serving Size</th>
<th>Total Calories (Kcal)</th>
<th>Total Carbohydrate (g)</th>
<th>Total Protein (g)</th>
<th>Total Fat (g)</th>
<th>Actions</th>
</tr>
<?php foreach($query as $item): ?>
<?php $recommended_recipe_id= $item->recommended_recipe_id ?>
<?php $recipe_name= $item->recipe_name ?>
<tr>
<td style="display:none;"><?=$item->recommended_recipe_id ?></td>
<td><?=$item->health.condition ?></td>
<td><?=$item->meal.type ?></td>
<td><?=$item->recipe.name ?></td>
<td><?=$item->serving_sugg ?></td>
<td><?=$item->calories ?></td>
<td><?=$item->carbohydrate ?></td>
<td><?=$item->protein ?></td>
<td><?=$item->fat ?></td>
<td>
<a href=" ../home_cmanager/view_created_recipe/<?php echo $recommended_recipe_id; ?>" >
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="View Recipe">
<span class="ui-icon ui-icon-folder-open"></span>
</li>
</ul>
</a>
<a href=" ../home_cmanager/edit_created_recipe/<?php echo $recommended_recipe_id; ?>" >
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="Edit Recipe">
<span class="ui-icon ui-icon-pencil"></span>
</li>
</ul>
</a>
<a href=" ../home_cmanager/delete_created_recipe/<?php echo $recommended_recipe_id; ?>" onclick
="return confirm('Are you sure you want to delete?')">
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="Delete">
<span class="ui-icon ui-icon-trash"></span>
</li>
</ul>
</a>
</td>
</tr>
<?php endforeach;?>
</table>
<?php else: ?>
<?php echo "No data to display."; ?>
<?php endif; ?>
</center>
<?php } ?>
<br>

```

```

<button id="viewCreatedRecipesBackButton" type="button" name="viewCreatedRecipesBackButton"
value="Back" onclick="window.location.href='<?php echo site_url(); ?>home_cmanager'">Back
</button>
<button id="viewCreatedRecipesHomeButton" type="button" name="viewCreatedRecipesHomeButton"
value="Home" onclick="window.location.href='<?php echo site_url(); ?>home_cmanager'">Home
</button>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url(); ?>layout/images/heasfimicon.ico"
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url(); ?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url(); ?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url(); ?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url(); ?>chart/highcharts.js"></script>
<script src="<?php echo base_url(); ?>chart/exporting.js"></script>
<script>
$(function() {
$( "#viewCreatedRecipesBackButton , #viewCreatedRecipesHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#viewCreatedRecipesDialog" ).dialog({
autoOpen: true ,
width: 950,
modal: true ,
closeOnEscape: false ,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"

});
});

$(document).ready(function() {
$("#success_msg").fadeOut(5000);
});
</script>
</head>
<body>
<div id="viewCreatedRecipesDialog" title="Created Recipes">
<br>
<div id="success_msg">
<center><font color="red" size="2"><b><?php echo $this->session->flashdata('success_msg');
?></b></font></center>
</div>
<?php if (is_array($query)) { ?>
<center>
<?php if (!empty($query)): ?>
<table>
<tr>
<th style="display:none;">ID</th>
<th>Health Condition</th>
<th>Meal Type</th>
<th>Recipe Name</th>
<th>Serving Size</th>
<th>Total Calories (Kcal)</th>
<th>Total Carbohydrate (g)</th>
<th>Total Protein (g)</th>
<th>Total Fat (g)</th>
<th>Actions</th>
</tr>
<?php foreach($query as $item): ?>
<?php $own_recipe_id= $item->own_recipe_id ?>
<?php $recipe_name= $item->recipe_name ?>
<tr>
<td style="display:none;"><?=$item->own_recipe_id ?></td>
<td><?=$item->health_condition ?></td>
<td><?=$item->meal_type ?></td>
<td><?=$item->recipe_name ?></td>
<td><?=$item->serving_sugg ?></td>
<td><?=$item->calories ?></td>
<td><?=$item->carbohydrate ?></td>
<td><?=$item->protein ?></td>
<td><?=$item->fat ?></td>
<td>
<a href=" ../home_user/view_created_recipe/<?php echo $own_recipe_id; ?>" >
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="View Recipe">

```

```

<span class="ui-icon ui-icon-folder-open"></span>
</li>
</ul>
</a>
<a href="../../home-user/edit-created-recipe/<?php echo $own_recipe_id; ?>" >
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="Edit Recipe">
<span class="ui-icon ui-icon-pencil"></span>
</li>
</ul>
</a>
<a href="../../home-user/delete-created-recipe/<?php echo $own_recipe_id; ?>" onclick="return
confirm('Are you sure you want to delete?')">
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="Delete">
<span class="ui-icon ui-icon-trash"></span>
</li>
</ul>
</a>
<a href="../../home-user/share-created-recipe/<?php echo $own_recipe_id; ?>">
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="Share on Site">
<span class="ui-icon ui-icon-link"></span>
</li>
</ul>
</a>
</td>
</tr>
<?php endforeach;?>
</table>
<?php else: ?>
<?php echo "No data to display."; ?>
<?php endif; ?>
</center>
<?php } ?>
<br>
<button id="viewCreatedRecipesBackButton" type="button" name="viewCreatedRecipesBackButton"
value="Back" onclick="window.location.href='<?php echo site_url(); ?>home-user'">Back</
button>
<button id="viewCreatedRecipesHomeButton" type="button" name="viewCreatedRecipesHomeButton"
value="Back" onclick="window.location.href='<?php echo site_url(); ?>home-user'">Home</
button>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url(); ?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url(); ?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url(); ?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url(); ?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url(); ?>chart/highcharts.js"></script>
<script src="<?php echo base_url(); ?>chart/exporting.js"></script>
<script>
$(function() {
$( "#viewCreatedRecipesBackButton, #viewCreatedRecipesHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#viewCreatedRecipesDialog" ).dialog({
autoOpen: true,
width: 950,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"

});
});
</script>
</head>
<body>
<div id="viewCreatedRecipesDialog" title='View Recipe'>
<br>
<?php if ( is_array($query) ) { ?>
<center>
<table>
<tr>

```

```

<?php $a = $query[0]->img_name;
$b = $query[0]->ext;
?>

</tr>
<tr>
<td class = "redbox"><b>Recipe Name:</b>
<td><?php $a = $query[0]->recipe_name; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Serving Size:</b>
<td><?php $a = $query[0]->serving_sugg; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Meal Type:</b>
<td><?php $a = $query[0]->meal_type; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Health Condition:</b>
<td><?php $a = $query[0]->health_condition; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Calories (Kcal):</b>
<td><?php $a = $query[0]->calories; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Carbohydrates (g):</b>
<td><?php $a = $query[0]->carbohydrate; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Protein (g):</b>
<td><?php $a = $query[0]->protein; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Fat (g):</b>
<td><?php $a = $query[0]->fat; echo $a; ?>
</tr>
<tr>
<th>Ingredients</th>
<th>Measure (g)</th>
</tr>
<?php foreach($query as $item): ?>
<?php $recommended_recipe_id= $item->recommended_recipe_id ?>
<tr>
<td><?= $item->food_name ?></td>
<td><?php $b= $item->measure;
$c= $item->serving_size;
echo $b*$c;
?>
</td>
</tr>
<?php endforeach;?>
<tr>
<td class = "redbox"><b>Procedure:</b>
<td><?php $a = $query[0]->preparation_procedure; echo $a; ?>
</tr>
</table>
</center>
<?php } ?>
<br>
<button id="viewCreatedRecipesBackButton" type="button" name="viewCreatedRecipesBackButton"
value="Back" onclick="window.location.href='<?php echo site_url(); ?>home.user/
recommended_select_by_date'">Back</button>
<button id="viewCreatedRecipesHomeButton" type="button" name="viewCreatedRecipesHomeButton"
value="Home" onclick="window.location.href='<?php echo site_url(); ?>home.user'">Home</
button>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url(); ?>chart/highcharts.js"></script>
<script src="<?php echo base_url(); ?>chart/exporting.js"></script>
<script>
$(function() {
$( "#viewFPBackButton, #viewFPHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {

```

```

$( this ).removeClass( "ui-state-hover" );
}
);

$( "#viewFPDialog" ).dialog({
autoOpen: true ,
width: 1100,
modal: true ,
closeOnEscape: false ,
beforeclose: function (event , ui) { return false ; },
dialogClass: "noclose"

});

});

$(document).ready(function() {
var options = {
chart: {
renderTo: 'weightChartBox',
type: 'line',
marginRight: 130,
marginBottom: 25
},
title: {
text: 'Weight Monitor',
x: -20 //center
},
subtitle: {
text: '',
x: -20
},
xAxis: {
categories: []
},
yAxis: {
title: {
text: 'Weight (kg)'
},
plotLines: [{
value: 0,
width: 1,
color: '#808080'
}],
tooltip: {
/* formatter: function() {
return '<b>'+ this.series.name + '</b><br/>'+
this.x + ': '+ this.y;
}*/
valueSuffix: 'kg'
},
legend: {
layout: 'vertical',
align: 'right',
verticalAlign: 'top',
x: -10,
y: 100,
borderWidth: 0
},
series: []
}

$.getJSON("weightmonitor_getdata", function(json) {
options.xAxis.categories = json[0]['data'];
options.series[0] = json[1];
options.series[1] = json[2];
//options.series[2] = json[3];
chart = new Highcharts.Chart(options);
});

$("#success_msg").fadeOut(5000);
});

</script>
</head>
<body>
<div id="viewFPDialog" title="Fitness Profile Information">
<br>
<div id="success_msg">
<center><font color="red" size="2"><b><?php echo $this->session->flashdata('success_msg');
?></b></font></center>
</div>
<?php if (is_array($query)) { ?>
<center>
<?php if(!empty($query)): ?>
<table>
<tr>
<div id="weightChartBox" style="min-width: 400px; height: 400px; margin: 0 auto"></div>
</tr>
<tr>
<th style="display:none;">ID</th>
<th>Current Weight (kg)</th>
<th>Goal Weight (kg)</th>
<th>Ideal Weight (kg)</th>
<th>Height (cm)</th>

```

```

<th>BMI</th>
<th>BMI Classification</th>
<th>Nutritional Status</th>
<th>Recommendation</th>
<th>Daily Calorie Limit (Kcal)</th>
<th>Daily Carbohydrate Limit (g)</th>
<th>Daily Protein Limit (g)</th>
<th>Daily Fat Limit (g)</th>
<th>Health Condition</th>
<th>Record Date</th>
<th>Actions</th>
</tr>
<?php foreach($query as $item): ?>
<?php $fitness_info_id = $item->fitness_info_id ?>
<tr>
<td style="display:none;"><?=$item->fitness_info_id ?></td>
<td><?=$item->current_weight ?></td>
<td><?=$item->goal_weight ?></td>
<td><?php
if ($gender = "Male") {
if (($item->height) > 152.4) {
$IBW = (((($item->height) - 152.4)*(0.3937))*(2.3)) + 50;
echo round($IBW);
}
else {
$IBW = 50;
echo round($IBW);
}
}
else {
if (($item->height) > 152.4) {
$IBW = (((($item->height) - 152.4)*(0.3937))*(2.3)) + 45.5;
echo round($IBW);
}
else {
$IBW = 45.5;
echo round($IBW);
}
}
?>
</td>
<td><?=$item->height ?></td>
<td><?=$BMI=round((($item->current_weight)/((($item->height/100)*($item->height/100))), 2) ?></td>
<td>
<?php
if ($BMI<=18.5) {
echo "Underweight";
} elseif (($BMI > 18.5) && ($BMI <= 24.9)) {
echo "Normal";
} elseif ($BMI > 24.9 && $BMI <= 29.9) {
echo "Overweight";
} elseif ($BMI > 29.9 && $BMI <= 39.9) {
echo "Obese";
} else {
echo "Insanely Obese";
}
?>
</td>
<td>
<?php
if ($BMI<=18.5) {
echo "Malnourished";
} elseif (($BMI > 18.5) && ($BMI <= 24.9)) {
echo "Normal";
} elseif ($BMI > 24.9 && $BMI <= 29.9) {
echo "Overnourished";
} elseif ($BMI > 29.9 && $BMI <= 39.9) {
echo "Very Excessively Nourished";
} else {
echo "Overflowingly Nourished";
}
?>
</td>
<td>
<?php
if ($BMI<=18.5) {
echo "Gain Weight";
} elseif (($BMI > 18.5) && ($BMI <= 24.9)) {
echo "Maintain Weight";
} elseif ($BMI > 24.9 && $BMI <= 29.9) {
echo "Lose Weight";
} elseif ($BMI > 29.9 && $BMI <= 39.9) {
echo "Lose Weight";
} else {
echo "Lose Weight";
}
?>
</td>
<td><?=$item->calorie_limit ?></td>
<td><?=$item->carbohydrate_limit ?></td>
<td><?=$item->protein_limit ?></td>
<td><?=$item->fat_limit ?></td>
<td><?=$item->health_condition ?></td>
<td><?=$date("Y-M-d", strtotime($item->record_date)); ?></td>
<td>
<a href="../../home_user/delete_fitness_profile/<?php echo $fitness_info_id; ?>" onclick="return confirm('Are you sure you want to delete?')">
<ul id="icons" class="ui-widget ui-helper-clearfix">

```

```

<li class="ui-state-default ui-corner-all" title="Delete">
<span class="ui-icon ui-icon-trash"></span>
</li>
</ul>
</a>
</td>
</tr>
<?php endforeach;?>
</table>
<?php else: ?>
<?php echo "No data to display."; ?>
<?php endif; ?>
</center>
<?php } ?>
<br>

<button id="viewFPBackButton" type="button" name="viewFPBackButton" value="Back" onclick="
window.location.href='<?php echo site_url(); ?>home-user'">Back</button>
<button id="viewFPHomeButton" type="button" name="viewFPHomeButton" value="Home" onclick="
window.location.href='<?php echo site_url(); ?>home-user'">Home</button>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url(); ?>chart/highcharts.js"></script>
<script src="<?php echo base_url(); ?>chart/exporting.js"></script>
<script>
$(function() {
$( "#viewCreatedRecipesBackButton, #viewCreatedRecipesHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#viewCreatedRecipesDialog" ).dialog({
autoOpen: true,
width: 950,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"

});
});
</script>
</head>
<body>
<div id="viewCreatedRecipesDialog" title='View Recipe'>
<br>

<?php if (is_array($query)) { ?>
<center>
<table>
<tr>
<?php $a = $query[0]->img.name;
$b = $query[0]->ext;
?>

</tr>
<tr>
<td class = "redbox"><b>Recipe Name:</b>
<td><?php $a = $query[0]->recipe.name; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Serving Size:</b>
<td><?php $a = $query[0]->serving.sugg; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Meal Type:</b>
<td><?php $a = $query[0]->meal.type; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Health Condition:</b>
<td><?php $a = $query[0]->health.condition; echo $a; ?>
</tr>

```

```

<tr>
<td class = "redbox"><b>Total Calories (Kcal):</b>
<td><?php $a = $query[0]->calories; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Carbohydrates (g):</b>
<td><?php $a = $query[0]->carbohydrate; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Protein (g):</b>
<td><?php $a = $query[0]->protein; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Fat (g):</b>
<td><?php $a = $query[0]->fat; echo $a; ?>
</tr>
<tr>
<th>Ingredients</th>
<th>Measure (g)</th>
</tr>
<?php foreach($query as $item): ?>
<?php $recommended_recipe_id= $item->recommended_recipe_id ?>
<tr>
<td><?= $item->food_name ?></td>
<td><?php $b= $item->measure;
$c= $item->servicing_size;
echo $b*$c;
?>
</td>
</tr>
<?php endforeach;?>
<tr>
<td class = "redbox"><b>Procedure:</b>
<td><?php $a = $query[0]->preparation_procedure; echo wordwrap($a, 15, "<br>\n", TRUE); ?>
</tr>
</table>
</center>
<?php } ?>
<br>
<button id="viewCreatedRecipesBackButton" type="button" name="viewCreatedRecipesBackButton"
value="Back" onclick="window.location.href='<?php echo site_url(); ?>home_cmanager/
search_meal'">Back</button>
<button id="viewCreatedRecipesHomeButton" type="button" name="viewCreatedRecipesHomeButton"
value="Home" onclick="window.location.href='<?php echo site_url(); ?>home_cmanager'">Home
</button>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url(); ?>chart/highcharts.js"></script>
<script src="<?php echo base_url(); ?>chart/exporting.js"></script>
<script>
$(function() {
$( "#viewCreatedRecipesBackButton, #viewCreatedRecipesHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#viewCreatedRecipesDialog" ).dialog({
autoOpen: true,
width: 950,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"
});
});
</script>
</head>
<body>
<div id="viewCreatedRecipesDialog" title='View Recipe'>
<br>
<?php if (is_array($query)) { ?>

```



```

<center>
<table>
<tr>
<?php $a = $query[0]->img_name;
$b = $query[0]->ext;
?>

</tr>
<tr>
<td class = "redbox"><b>Recipe Name:</b>
<td><?php $a = $query[0]->recipe_name; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Serving Size:</b>
<td><?php $a = $query[0]->serving_sugg; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Meal Type:</b>
<td><?php $a = $query[0]->meal_type; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Health Condition:</b>
<td><?php $a = $query[0]->health_condition; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Calories (Kcal):</b>
<td><?php $a = $query[0]->calories; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Carbohydrates (g):</b>
<td><?php $a = $query[0]->carbohydrate; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Protein (g):</b>
<td><?php $a = $query[0]->protein; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Fat (g):</b>
<td><?php $a = $query[0]->fat; echo $a; ?>
</tr>
<tr>
<th>Ingredients</th>
<th>Measure (g)</th>
</tr>
<?php foreach($query as $item): ?>
<?php $own_recipe_id= $item->own_recipe_id ?>
<tr>
<td><?= $item->food_name ?></td>
<td><?php $b= $item->measure;
$c= $item->serving_size;
echo $b*$c;
?>
</td>
</tr>
<?php endforeach;?>
<tr>
<td class = "redbox"><b>Procedure:</b>
<td><?php $a = $query[0]->preparation_procedure; echo wordwrap($a, 15, "<br>\n", TRUE); ?>
</tr>
</table>
</center>
<?php } ?>
<button id="viewCreatedRecipesBackButton" type="button" name="viewCreatedRecipesBackButton"
value="Back" onclick="window.location.href='<?php echo site_url(); ?>home.user/search_meal
'">Back</button>
<button id="viewCreatedRecipesHomeButton" type="button" name="viewCreatedRecipesHomeButton"
value="Home" onclick="window.location.href='<?php echo site_url(); ?>home.user'">Home</
button>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url(); ?>chart/highcharts.js"></script>
<script src="<?php echo base_url(); ?>chart/exporting.js"></script>
</script>
$(function() {
$( "#viewCreatedRecipesBackButton, #viewCreatedRecipesHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {

```

```

$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#viewCreatedRecipesDialog" ).dialog({
autoOpen: true,
width: 950,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"

});
});
</script>
</head>
<body>
<div id="viewCreatedRecipesDialog" title='View Recipe'>
<br>

<?php if ( is_array( $query ) ) { ?>
<center>
<table>
<tr>
<?php $a = $query[0]->img_name;
$b = $query[0]->ext;
?>

</tr>
<tr>
<td class = "redbox"><b>Recipe Name:</b>
<td><?php $a = $query[0]->recipe_name; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Serving Size:</b>
<td><?php $a = $query[0]->servicing_sugg; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Meal Type:</b>
<td><?php $a = $query[0]->meal_type; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Health Condition:</b>
<td><?php $a = $query[0]->health_condition; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Calories (Kcal):</b>
<td><?php $a = $query[0]->calories; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Carbohydrates (g):</b>
<td><?php $a = $query[0]->carbohydrate; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Protein (g):</b>
<td><?php $a = $query[0]->protein; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Fat (g):</b>
<td><?php $a = $query[0]->fat; echo $a; ?>
</tr>
<tr>
<th>Ingredients</th>
<th>Measure (g)</th>
</tr>
<?php foreach( $query as $item ): ?>
<?php $recommended_recipe_id= $item->recommended_recipe_id ?>
<tr>
<td><?= $item->food_name ?></td>
<td><?php $b= $item->measure;
$c= $item->servicing_size;
echo $b*$c;
?>
</td>
</tr>
<?php endforeach;?>
<tr>
<td class = "redbox"><b>Procedure:</b>
<td><?php $a = $query[0]->preparation_procedure; echo wordwrap($a, 15, "<br>\n", TRUE); ?>
</tr>
</table>
</center>
<?php } ?>
<br>
<button id="viewCreatedRecipesBackButton" type="button" name="viewCreatedRecipesBackButton"
value="Back" onclick="window.location.href='<?php echo site_url(); ?>home_cmanager/
show_created_recipes'">Back</button>
<button id="viewCreatedRecipesHomeButton" type="button" name="viewCreatedRecipesHomeButton"
value="Home" onclick="window.location.href='<?php echo site_url(); ?>home_cmanager'">Home
</button>
</div>
</body>
</html>
<html lang="en">

```

```

<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url(); ?>chart/highcharts.js"></script>
<script src="<?php echo base_url(); ?>chart/exporting.js"></script>
<script>
$(function() {
$( "#viewCreatedRecipesBackButton, #viewCreatedRecipesHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#viewCreatedRecipesDialog" ).dialog({
autoOpen: true,
width: 950,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"

});
});
</script>
</head>
<body>
<div id="viewCreatedRecipesDialog" title='View Recipe'>
<br>

<?php if ( is_array( $query ) ) { ?>
<center>
<table>
<tr>
<?php $a = $query[0]->img_name;
$b = $query[0]->ext;
?>

</tr>
<tr>
<td class = "redbox"><b>Recipe Name:</b>
<td><?php $a = $query[0]->recipe_name; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Serving Size:</b>
<td><?php $a = $query[0]->servicing_sugg; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Meal Type:</b>
<td><?php $a = $query[0]->meal_type; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Health Condition:</b>
<td><?php $a = $query[0]->health_condition; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Calories (Kcal):</b>
<td><?php $a = $query[0]->calories; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Carbohydrates (g):</b>
<td><?php $a = $query[0]->carbohydrate; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Protein (g):</b>
<td><?php $a = $query[0]->protein; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Fat (g):</b>
<td><?php $a = $query[0]->fat; echo $a; ?>
</tr>
<tr>
<th>Ingredients</th>
<th>Measure (g)</th>
</tr>
<?php foreach( $query as $item): ?>
<?php $own_recipe_id= $item->own_recipe_id ?>
<tr>
<td><?= $item->food_name ?></td>
<td><?php $b= $item->measure;
$c= $item->servicing_size;
echo $b*$c;

```

```

?>
</td>
</tr>
<?php endforeach;?>
<tr>
<td class = "redbox"><b>Procedure:</b></td>
<td><?php $a = $query[0]->preparation-procedure; echo wordwrap($a, 15, "<br>\n", TRUE); ?></td>
</div>
</tr>
</table>
</center>
<?php } ?>
<br>
<button id="viewCreatedRecipesBackButton" type="button" name="viewCreatedRecipesBackButton"
value="Back" onclick="window.location.href='<?php echo site_url(); ?>home_user/
show_created_recipes'">Back</button>
<button id="viewCreatedRecipesHomeButton" type="button" name="viewCreatedRecipesHomeButton"
value="Home" onclick="window.location.href='<?php echo site_url(); ?>home_user'">Home</
button>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url();?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url();?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url();?>chart/highcharts.js"></script>
<script src="<?php echo base_url();?>chart/exporting.js"></script>
<script>
$(function() {
$( "#viewCreatedRecipesBackButton, #viewCreatedRecipesHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#viewCreatedRecipesDialog" ).dialog({
autoOpen: true,
width: 950,
modal: true,
closeOnEscape: false,
beforeclose: function( event, ui ) { return false; },
dialogClass: "noclose"

});
});
</script>
</head>
<body>
<div id="viewCreatedRecipesDialog" title='View Recipe'>
<br>

<?php if ( is_array( $query ) ) { ?>
<center>
<table>
<tr>
<?php $a = $query[0]->img_name;
$b = $query[0]->ext;
?>

</tr>
<tr>
<td class = "redbox"><b>Recipe Name:</b>
<td><?php $a = $query[0]->recipe_name; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Serving Size:</b>
<td><?php $a = $query[0]->servings_sugg; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Meal Type:</b>
<td><?php $a = $query[0]->meal_type; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Health Condition:</b>
<td><?php $a = $query[0]->health_condition; echo $a; ?>
</tr>
<tr>

```

```

<td class = "redbox"><b>Total Calories (Kcal):</b>
<td><?php $a = $query[0]->calories; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Carbohydrates (g):</b>
<td><?php $a = $query[0]->carbohydrate; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Protein (g):</b>
<td><?php $a = $query[0]->protein; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Fat (g):</b>
<td><?php $a = $query[0]->fat; echo $a; ?>
</tr>
<tr>
<th>Ingredients </th>
<th>Measure (g)</th>
</tr>
<?php foreach($query as $item): ?>
<?php $own_recipe_id= $item->own_recipe_id ?>
<tr>
<td><?= $item->food_name ?></td>
<td><?php $b= $item->measure;
$c= $item->servicing_size;
echo $b*$c;
?>
</td>
</tr>
<?php endforeach;?>
<tr>
<td class = "redbox"><b>Procedure:</b>
<td><?php $a = $query[0]->preparation_procedure; echo wordwrap($a, 15, "<br>\n", TRUE); ?>
</tr>
</table>
</center>
<?php } ?>
<br>
<button id="viewCreatedRecipesBackButton" type="button" name="viewCreatedRecipesBackButton"
value="Back" onclick="window.location.href='<?php echo site_url(); ?>home.user/
select_by_date'">Back</button>
<button id="viewCreatedRecipesHomeButton" type="button" name="viewCreatedRecipesHomeButton"
value="Home" onclick="window.location.href='<?php echo site_url(); ?>home.user'">Home</
button>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url(); ?>chart/highcharts.js"></script>
<script src="<?php echo base_url(); ?>chart/exporting.js"></script>
<script>
$(function() {
$( "#viewCreatedRecipesBackButton , #viewCreatedRecipesHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#viewCreatedRecipesDialog" ).dialog({
autoOpen: true ,
width: 950,
modal: true ,
closeOnEscape: false ,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"

});
});
</script>
</head>
<body>
<div id="viewCreatedRecipesDialog" title='View Recipe'>
<br>

<?php if ( is_array($query) ) { ?>
<center>

```

```

<table>
<tr>
<?php $a = $query[0]->img_name;
$b = $query[0]->ext;
?>

</tr>
<tr>
<td class = "redbox"><b>Recipe Name:</b>
<td><?php $a = $query[0]->recipe_name; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Serving Size:</b>
<td><?php $a = $query[0]->servng_sugg; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Meal Type:</b>
<td><?php $a = $query[0]->meal_type; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Health Condition:</b>
<td><?php $a = $query[0]->health_condition; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Calories (Kcal):</b>
<td><?php $a = $query[0]->calories; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Carbohydrates (g):</b>
<td><?php $a = $query[0]->carbohydrate; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Protein (g):</b>
<td><?php $a = $query[0]->protein; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Fat (g):</b>
<td><?php $a = $query[0]->fat; echo $a; ?>
</tr>
<tr>
<th>Ingredients</th>
<th>Measure (g)</th>
</tr>
<?php foreach($query as $item): ?>
<?php $recommended_recipe_id= $item->recommended_recipe_id ?>
<tr>
<td><?= $item->food_name ?></td>
<td><?php $b= $item->measure;
$c= $item->servng_size;
echo $b*$c;
?>
</td>
</tr>
<?php endforeach;?>
<tr>
<td class = "redbox"><b>Procedure:</b>
<td><?php $a = $query[0]->preparation_procedure; echo wordwrap($a, 15, "<br>\n", TRUE); ?>
</tr>
</table>
</center>
<?php } ?>
<br>
<button id="viewCreatedRecipesBackButton" type="button" name="viewCreatedRecipesBackButton"
value="Back" onclick="window.location.href='<?php echo site_url(); ?>home_cmanager/
select_by_date'">Back</button>
<button id="viewCreatedRecipesHomeButton" type="button" name="viewCreatedRecipesHomeButton"
value="Home" onclick="window.location.href='<?php echo site_url(); ?>home_cmanager'">Home
</button>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url(); ?>chart/highcharts.js"></script>
<script src="<?php echo base_url(); ?>chart/exporting.js"></script>
<script>
$(function() {
$( "#viewCreatedRecipesBackButton , #viewCreatedRecipesHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );

```

```

    },
    function() {
    $( this ).removeClass( "ui-state-hover" );
    }
    );

    $( "#viewCreatedRecipesDialog" ).dialog({
    autoOpen: true,
    width: 950,
    modal: true,
    closeOnEscape: false,
    beforeclose: function (event, ui) { return false; },
    dialogClass: "noclose"

    });
    });
</script>
</head>
<body>
<div id="viewCreatedRecipesDialog" title='View Recipe'>
<br>

<?php if ( is_array( $query ) ) { ?>
<center>
<table>
<tr>
<?php $a = $query[0]->img_name;
$b = $query[0]->ext;
?>

</tr>
<tr>
<td class = "redbox"><b>Recipe Name:</b>
<td><?php $a = $query[0]->recipe_name; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Serving Size:</b>
<td><?php $a = $query[0]->servicing_sugg; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Meal Type:</b>
<td><?php $a = $query[0]->meal_type; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Health Condition:</b>
<td><?php $a = $query[0]->health_condition; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Calories (Kcal):</b>
<td><?php $a = $query[0]->calories; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Carbohydrates (g):</b>
<td><?php $a = $query[0]->carbohydrate; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Protein (g):</b>
<td><?php $a = $query[0]->protein; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Fat (g):</b>
<td><?php $a = $query[0]->fat; echo $a; ?>
</tr>
<tr>
<th>Ingredients</th>
<th>Measure (g)</th>
</tr>
<?php foreach( $query as $item): ?>
<?php $recommended_recipe_id= $item->recommended_recipe_id ?>
<tr>
<td><?= $item->food_name ?></td>
<td><?php $b= $item->measure;
$c= $item->servicing_size;
echo $b*$c;
?>
</td>
</tr>
<?php endforeach;?>
<tr>
<td class = "redbox"><b>Procedure:</b>
<td><?php $a = $query[0]->preparation_procedure; echo wordwrap($a, 15, "<br>\n", TRUE); ?>
</tr>
</table>
</center>
<?php } ?>
<br>
<button id="viewCreatedRecipesBackButton" type="button" name="viewCreatedRecipesBackButton"
value="Back" onclick="window.location.href='<?php echo site_url(); ?>home.user/
show_recommended_recipes'">Back</button>
<button id="viewCreatedRecipesHomeButton" type="button" name="viewCreatedRecipesHomeButton"
value="Home" onclick="window.location.href='<?php echo site_url(); ?>home.user'">Home</
button>
</div>
</body>
</html>

```

```

<html lang="en">
<head>

```

```

<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url();?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url();?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url();?>chart/highcharts.js"></script>
<script src="<?php echo base_url();?>chart/exporting.js"></script>
<script>
$(function() {
$("#userSearchRRBackButton, #userSearchRRButton, #userSearchRRHomeButton").button();

// Hover states on the static widgets
$("#icons li").hover(
function() {
$(this).addClass("ui-state-hover");
},
function() {
$(this).removeClass("ui-state-hover");
}
);

$("#viewCreatedRecipesDialog").dialog({
autoOpen: true,
width: 950,
modal: true,
closeOnEscape: false,
beforeclose: function(event, ui) { return false; },
dialogClass: "noclose"
});
});
</script>
</head>
<body>
<div id="viewCreatedRecipesDialog" title="Recommended Recipes">
<br>
<?form_open('home_user/search_recrec');?>
<?php $search_recrec = array('name'=>'search_recrec', 'id'=>'search_recrec', 'value'=>'');?>
<center>Recommended Recipes: <?form_input($search_recrec);?>
<button id="userSearchRRButton" type="submit" value="" title="Search Recommended Recipes">
Search</button>
<button id="userSearchRRBackButton" type="button" name="userSearchRRBackButton" value="Back"
onclick="window.location.href='<?php echo site_url();?>home_user'">Back</button>
<button id="userSearchRRHomeButton" type="button" name="userSearchRRHomeButton" value="Home"
onclick="window.location.href='<?php echo site_url();?>home_user'">Home</button>
</center>
<?form_close();?>

<?php if (is_array($query)) { ?>
<center>
<?php if (!empty($query)): ?>
<table>
<tr>
<th style="display:none;">ID</th>
<th>Health Condition</th>
<th>Meal Type</th>
<th>Recipe Name</th>
<th>Serving Size</th>
<th>Total Calories (Kcal)</th>
<th>Total Carbohydrate (g)</th>
<th>Total Protein (g)</th>
<th>Total Fat (g)</th>
<th>Actions</th>
</tr>
<?php foreach($query as $item): ?>
<?php $recommended_recipe_id= $item->recommended_recipe_id ?>
<?php $recipe_name= $item->recipe_name ?>
<tr>
<td style="display:none;"><?=$item->recommended_recipe_id ?></td>
<td><?=$item->health_condition ?></td>
<td><?=$item->meal_type ?></td>
<td><?=$item->recipe_name ?></td>
<td><?=$item->serving_sugg ?></td>
<td><?=$item->calories ?></td>
<td><?=$item->carbohydrate ?></td>
<td><?=$item->protein ?></td>
<td><?=$item->fat ?></td>
<td>
<a href="..../home_user/view-recommended-recipe/<?php echo $recommended_recipe_id; ?>" >
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="View Recipe">
<span class="ui-icon ui-icon-folder-open"></span>
</li>
</ul>
</a>
</td>
</tr>
<?php endforeach;?>
</table>
<?php else: ?>

```



```

<?php echo "No data to display."; ?>
<?php endif; ?>
</center>
<?php } ?>
<br>
</div>
</body>
</html>

```

---

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="<?php echo base_url();?>validation/jquery-validation-1.11.1/dist/jquery.validate.min.js"></script>
<script src="<?php echo base_url();?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url();?>chart/highcharts.js"></script>
<script src="<?php echo base_url();?>chart/exporting.js"></script>
</script>
$(function() {
$( "#viewCreatedRecipesBackButton, #viewCreatedRecipesHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#viewCreatedRecipesDialog" ).dialog({
autoOpen: true,
width: 950,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"
});
});
</script>
</head>
<body>
<div id="viewCreatedRecipesDialog" title='View Recipe'>
<br>

<?php if ( is_array( $query ) ) { ?>
<center>
<table>
<tr>
<?php $a = $query[0]->img_name;
$b = $query[0]->ext;
?>

</tr>
<tr>
<td class = "redbox"><b>Recipe Name:</b>
<td><?php $a = $query[0]->recipe_name; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Serving Size:</b>
<td><?php $a = $query[0]->servicing_sugg; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Meal Type:</b>
<td><?php $a = $query[0]->meal_type; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Health Condition:</b>
<td><?php $a = $query[0]->health_condition; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Calories (Kcal):</b>
<td><?php $a = $query[0]->calories; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Carbohydrates (g):</b>
<td><?php $a = $query[0]->carbohydrate; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Protein (g):</b>
<td><?php $a = $query[0]->protein; echo $a; ?>
</tr>
<tr>
<td class = "redbox"><b>Total Fat (g):</b>
<td><?php $a = $query[0]->fat; echo $a; ?>
</tr>

```

```

<tr>
<th>Ingredients</th>
<th>Measure (g)</th>
</tr>
<?php foreach($query as $item): ?>
<?php $shared_recipe_id= $item->shared_recipe_id ?>
<tr>
<td><?=$ $item->food_name ?></td>
<td><?php $b= $item->measure;
$c= $item->servicing_size;
echo $b*$c;
?>
</td>
</tr>
<?php endforeach;?>
<tr>
<td class = "redbox"><b>Procedure:</b>
<td><?php $a = $query[0]->preparation_procedure; echo wordwrap($a, 15, "<br>\n", TRUE); ?>
</tr>
</table>
</center>
<?php } ?>
<br>
<button id="viewCreatedRecipesBackButton" type="button" name="viewCreatedRecipesBackButton"
value="Back" onclick="window.location.href='<?php echo site_url(); ?>home.user/
show.shared.recipes'">Back</button>
<button id="viewCreatedRecipesHomeButton" type="button" name="viewCreatedRecipesHomeButton"
value="Home" onclick="window.location.href='<?php echo site_url(); ?>home.user'">Home</
button>
</div>
</body>
</html>

```

```

<html lang="en">
<head>
<meta charset="utf-8">
<title>Healthy Eating Alert System</title>
<link rel="icon" type="image/ico" href="<?php echo base_url();?>layout/images/heasfimicon.ico
">
<link rel="stylesheet" type="text/css" href="<?php echo base_url();?>layout/style.css">
<link rel="stylesheet" href="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/css/
blitzer/jquery-ui-1.10.3.custom.css" />
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></
script>
<script src="<?php echo base_url();?>layout/jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom
.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-validation-1.11.1/dist/jquery.validate
.min.js"></script>
<script src="<?php echo base_url(); ?>validation/jquery-1.11.0.min.js"></script>
<script src="<?php echo base_url(); ?>chart/highcharts.js"></script>
<script src="<?php echo base_url(); ?>chart/exporting.js"></script>
<script>
$(function() {
$( "#userSearchRRBackButton, #userSearchRRButton, #userSearchRRHomeButton" ).button();

// Hover states on the static widgets
$( "#icons li" ).hover(
function() {
$( this ).addClass( "ui-state-hover" );
},
function() {
$( this ).removeClass( "ui-state-hover" );
}
);

$( "#viewCreatedRecipesDialog" ).dialog({
autoOpen: true,
width: 950,
modal: true,
closeOnEscape: false,
beforeclose: function (event, ui) { return false; },
dialogClass: "noclose"

});
});
</script>
</head>
<body>
<div id="viewCreatedRecipesDialog" title="Shared Recipes">
<br>
<?form_open('home.user/search_sharedrec');?>
<?php $search_sharedrec = array('name'=>'search_sharedrec', 'id'=>'search_sharedrec', 'value
'=>'' ); ?>
<center>Shared Recipes: <?form_input($search_sharedrec);?>
<button id="userSearchRRButton" type="submit" value="" title="Search Shared Recipes">Search</
button>
<button id="userSearchRRBackButton" type="button" name="userSearchRRBackButton" value="Back"
onclick="window.location.href='<?php echo site_url(); ?>home.user'">Back</button>
<button id="userSearchRRHomeButton" type="button" name="userSearchRRHomeButton" value="Home"
onclick="window.location.href='<?php echo site_url(); ?>home.user'">Home</button>
</center>
<?form_close();?>

<?php if (is_array($query)) { ?>
<center>
<?php if (!empty($query)): ?>
<table>
<tr>

```

```

<th style="display:none;">ID</th>
<th>Health Condition</th>
<th>Meal Type</th>
<th>Recipe Name</th>
<th>Serving Size</th>
<th>Total Calories (Kcal)</th>
<th>Total Carbohydrate (g)</th>
<th>Total Protein (g)</th>
<th>Total Fat (g)</th>
<th>Actions</th>
</tr>
<?php foreach($query as $item): ?>
<?php $shared_recipe_id= $item->shared_recipe_id ?>
<?php $recipe_name= $item->recipe_name ?>
<tr>
<td style="display:none;"><?= $item->shared_recipe_id ?></td>
<td><?= $item->health_condition ?></td>
<td><?= $item->meal_type ?></td>
<td><?= $item->recipe_name ?></td>
<td><?= $item->serving_sugg ?></td>
<td><?= $item->calories ?></td>
<td><?= $item->carbohydrate ?></td>
<td><?= $item->protein ?></td>
<td><?= $item->fat ?></td>
<td>
<a href="../../home_user/view_shared_recipe/<?php echo $shared_recipe_id; ?>" >
<ul id="icons" class="ui-widget ui-helper-clearfix">
<li class="ui-state-default ui-corner-all" title="View Recipe">
<span class="ui-icon ui-icon-folder-open"></span>
</li>
</ul>
</a>
</td>
</tr>
<?php endforeach;?>
</table>
<?php else: ?>
<?php echo "No data to display."; ?>
<?php endif; ?>
</center>
<?php } ?>
<br>
</div>
</body>
</html>

```

---

## B. Source Code

### B.1 Configs

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/*
|-----|
| AUTO-LOADER |
|-----|
| This file specifies which systems should be loaded by default.
|
| In order to keep the framework as light-weight as possible only the
| absolute minimal resources are loaded by default. For example,
| the database is not connected to automatically since no assumption
| is made regarding whether you intend to use it. This file lets
| you globally define which systems you would like loaded with every
| request.
|
|-----|
| Instructions |
|-----|
| These are the things you can load automatically:
|
| 1. Packages
| 2. Libraries
| 3. Helper files
| 4. Custom config files
| 5. Language files
| 6. Models
|
*/

/*
|-----|
| Auto-load Packages |
|-----|
| Prototype:
|
| $autoload['packages'] = array(APPPATH.'third_party', '/usr/local/shared');
|
*/

$autoload['packages'] = array();

/*
|-----|
| Auto-load Libraries |
|-----|

```

```

|-----|
| These are the classes located in the system/libraries folder |
| or in your application/libraries folder. |
|-----|
| Prototype: |
|           $autoload['libraries'] = array('database', 'session', 'xmlrpc'); |
|*/
$autoload['libraries'] = array('database','session','facebook');

/*
|-----|
| Auto-load Helper Files |
|-----|
| Prototype: |
|           $autoload['helper'] = array('url', 'file'); |
|*/
$autoload['helper'] = array('html','form','url','file');

/*
|-----|
| Auto-load Config files |
|-----|
| Prototype: |
|           $autoload['config'] = array('config1', 'config2'); |
| NOTE: This item is intended for use ONLY if you have created custom |
| config files. Otherwise, leave it blank. |
|*/
$autoload['config'] = array();

/*
|-----|
| Auto-load Language files |
|-----|
| Prototype: |
|           $autoload['language'] = array('lang1', 'lang2'); |
| NOTE: Do not include the "_lang" part of your file. For example |
| "codeigniter_lang.php" would be referenced as array('codeigniter'); |
|*/
$autoload['language'] = array();

/*
|-----|
| Auto-load Models |
|-----|
| Prototype: |
|           $autoload['model'] = array('model1', 'model2'); |
|*/
$autoload['model'] = array();

/* End of file autoload.php */
/* Location: ./application/config/autoload.php */
|-----|

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

/*
|-----|
| Base Site URL |
|-----|
| URL to your CodeIgniter root. Typically this will be your base URL, |
| WITH a trailing slash: |
|           http://example.com/ |
| If this is not set then CodeIgniter will guess the protocol, domain and |
| path to your installation. |
|*/
$config['base_url'] = 'https://agila.upm.edu.ph/~pjdizon/FIM/';

/*
|-----|
| Index File |
|-----|
| Typically this will be your index.php file, unless you've renamed it to |
| something else. If you are using mod_rewrite to remove the page set this |
| variable so that it is blank. |

```

```

|
| */
| $config['index_page'] = '';
|
| /*
| -----
| | URI PROTOCOL
| -----
| |
| | This item determines which server global should be used to retrieve the
| | URI string. The default setting of 'AUTO' works for most servers.
| | If your links do not seem to work, try one of the other delicious flavors:
| |
| | 'AUTO'                Default - auto detects
| | 'PATH.INFO'           Uses the PATH.INFO
| | 'QUERY.STRING'        Uses the QUERY.STRING
| | 'REQUEST.URI'         Uses the REQUEST.URI
| | 'ORIG.PATH.INFO'      Uses the ORIG.PATH.INFO
| |
| */
| $config['uri_protocol'] = 'AUTO';
|
| /*
| -----
| | URL suffix
| -----
| |
| | This option allows you to add a suffix to all URLs generated by CodeIgniter.
| | For more information please see the user guide:
| |
| | http://codeigniter.com/user\_guide/general/urls.html
| |
| */
|
| $config['url_suffix'] = '';
|
| /*
| -----
| | Default Language
| -----
| |
| | This determines which set of language files should be used. Make sure
| | there is an available translation if you intend to use something other
| | than english.
| |
| */
| $config['language'] = 'english';
|
| /*
| -----
| | Default Character Set
| -----
| |
| | This determines which character set is used by default in various methods
| | that require a character set to be provided.
| |
| */
| $config['charset'] = 'UTF-8';
|
| /*
| -----
| | Enable/Disable System Hooks
| -----
| |
| | If you would like to use the 'hooks' feature you must enable it by
| | setting this variable to TRUE (boolean). See the user guide for details.
| |
| */
| $config['enable_hooks'] = FALSE;
|
| /*
| -----
| | Class Extension Prefix
| -----
| |
| | This item allows you to set the filename/classname prefix when extending
| | native libraries. For more information please see the user guide:
| |
| | http://codeigniter.com/user\_guide/general/core\_classes.html
| | http://codeigniter.com/user\_guide/general/creating\_libraries.html
| |
| */
| $config['subclass_prefix'] = 'MY_';
|
| /*
| -----
| | Allowed URL Characters
| -----
| |
| | This lets you specify with a regular expression which characters are permitted
| | within your URLs. When someone tries to submit a URL with disallowed
| | characters they will get a warning message.
| |
| | As a security measure you are STRONGLY encouraged to restrict URLs to
| | as few characters as possible. By default only these are allowed: a-z 0-9%.:_-
| |
| | Leave blank to allow all characters -- but only if you are insane.
|

```

```

| DO NOT CHANGE THIS UNLESS YOU FULLY UNDERSTAND THE REPERCUSSIONS!!
|
| */
$config['permitted_uri_chars'] = 'a-z 0-9~%.:_-\'';
|
| /*
| -----
| | Enable Query Strings
| -----
|
| By default CodeIgniter uses search-engine friendly segment based URLs:
| example.com/who/what/where/
|
| By default CodeIgniter enables access to the $_GET array. If for some
| reason you would like to disable it, set 'allow_get_array' to FALSE.
|
| You can optionally enable standard query string based URLs:
| example.com?who=me&what=something&where=here
|
| Options are: TRUE or FALSE (boolean)
|
| The other items let you set the query string 'words' that will
| invoke your controllers and its functions:
| example.com/index.php?c=controller&m=function
|
| Please note that some of the helpers won't work as expected when
| this feature is enabled, since CodeIgniter is designed primarily to
| use segment based URLs.
|
| */
$config['allow_get_array'] = TRUE;
$config['enable_query_strings'] = FALSE;
$config['controller_trigger'] = 'c';
$config['function_trigger'] = 'm';
$config['directory_trigger'] = 'd'; // experimental not currently in use
|
| /*
| -----
| | Error Logging Threshold
| -----
|
| If you have enabled error logging, you can set an error threshold to
| determine what gets logged. Threshold options are:
| You can enable error logging by setting a threshold over zero. The
| threshold determines what gets logged. Threshold options are:
|
|     0 = Disables logging, Error logging TURNED OFF
|     1 = Error Messages (including PHP errors)
|     2 = Debug Messages
|     3 = Informational Messages
|     4 = All Messages
|
| For a live site you'll usually only enable Errors (1) to be logged otherwise
| your log files will fill up very fast.
|
| */
$config['log_threshold'] = 0;
|
| /*
| -----
| | Error Logging Directory Path
| -----
|
| Leave this BLANK unless you would like to set something other than the default
| application/logs/ folder. Use a full server path with trailing slash.
|
| */
$config['log_path'] = '';
|
| /*
| -----
| | Date Format for Logs
| -----
|
| Each item that is logged has an associated date. You can use PHP date
| codes to set your own date formatting
|
| */
$config['log_date_format'] = 'Y-m-d H:i:s';
|
| /*
| -----
| | Cache Directory Path
| -----
|
| Leave this BLANK unless you would like to set something other than the default
| system/cache/ folder. Use a full server path with trailing slash.
|
| */
$config['cache_path'] = '';
|
| /*
| -----
| | Encryption Key
| -----
|
| If you use the Encryption class or the Session class you
| MUST set an encryption key. See the user guide for info.

```

```

|
| */
| $config['encryption_key'] = 'REALLYLONGNUMBER';
|
| /*
| -----
| | Session Variables
| -----
|
| 'sess_cookie_name'           = the name you want for the cookie
| 'sess_expiration'           = the number of SECONDS you want the session to last.
|   by default sessions last 7200 seconds (two hours). Set to zero for no expiration.
| 'sess_expire_on_close'      = Whether to cause the session to expire automatically
|   when the browser window is closed
| 'sess_encrypt_cookie'       = Whether to encrypt the cookie
| 'sess_use_database'         = Whether to save the session data to a database
| 'sess_table_name'           = The name of the session database table
| 'sess_match_ip'             = Whether to match the user's IP address when reading
|   the session data
| 'sess_match_useragent'      = Whether to match the User Agent when reading the session
|   data
| 'sess_time_to_update'       = how many seconds between CI refreshing Session Information
|
| */
| $config['sess_cookie_name']   = 'ci_session';
| $config['sess_expiration']    = 7200;
| $config['sess_expire_on_close'] = FALSE;
| $config['sess_encrypt_cookie'] = FALSE;
| $config['sess_use_database']   = FALSE;
| $config['sess_table_name']    = 'ci_sessions';
| $config['sess_match_ip']      = FALSE;
| $config['sess_match_useragent'] = TRUE;
| $config['sess_time_to_update'] = 300;
|
| /*
| -----
| | Cookie Related Variables
| -----
|
| 'cookie_prefix' = Set a prefix if you need to avoid collisions
| 'cookie_domain' = Set to .your-domain.com for site-wide cookies
| 'cookie_path'   = Typically will be a forward slash
| 'cookie_secure' = Cookies will only be set if a secure HTTPS connection exists.
|
| */
| $config['cookie_prefix']   = "";
| $config['cookie_domain']   = "";
| $config['cookie_path']     = "/";
| $config['cookie_secure']   = FALSE;
|
| /*
| -----
| | Global XSS Filtering
| -----
|
| Determines whether the XSS filter is always active when GET, POST or
| COOKIE data is encountered
|
| */
| $config['global_xss_filtering'] = FALSE;
|
| /*
| -----
| | Cross Site Request Forgery
| -----
|
| Enables a CSRF cookie token to be set. When set to TRUE, token will be
| checked on a submitted form. If you are accepting user data, it is strongly
| recommended CSRF protection be enabled.
|
| 'csrf_token_name' = The token name
| 'csrf_cookie_name' = The cookie name
| 'csrf_expire' = The number in seconds the token should expire.
|
| */
| $config['csrf_protection'] = FALSE;
| $config['csrf_token_name'] = 'csrf_test_name';
| $config['csrf_cookie_name'] = 'csrf_cookie_name';
| $config['csrf_expire'] = 7200;
|
| /*
| -----
| | Output Compression
| -----
|
| Enables Gzip output compression for faster page loads. When enabled,
| the output class will test whether your server supports Gzip.
| Even if it does, however, not all browsers support compression
| so enable only if you are reasonably sure your visitors can handle it.
|
| VERY IMPORTANT:  If you are getting a blank page when compression is enabled it
| means you are prematurely outputting something to your browser. It could
| even be a line of whitespace at the end of one of your scripts. For
| compression to work, nothing can be sent before the output buffer is called
| by the output class. Do not 'echo' any values with compression enabled.
|
| */
| $config['compress_output'] = FALSE;
|
| /*
| -----

```

```

| Master Time Reference
|-----|
| Options are 'local' or 'gmt'. This pref tells the system whether to use
| your server's local time as the master 'now' reference, or convert it to
| GMT. See the 'date helper' page of the user guide for information
| regarding date handling.
|
|*/
$config['time_reference'] = 'local';

/*
| Rewrite PHP Short Tags
|-----|
| If your PHP installation does not have short tag support enabled CI
| can rewrite the tags on-the-fly, enabling you to utilize that syntax
| in your view files. Options are TRUE or FALSE (boolean)
|
|*/
$config['rewrite_short_tags'] = FALSE;

/*
| Reverse Proxy IPs
|-----|
| If your server is behind a reverse proxy, you must whitelist the proxy IP
| addresses from which CodeIgniter should trust the HTTP_X_FORWARDED_FOR
| header in order to properly identify the visitor's IP address.
| Comma-delimited, e.g. '10.0.1.200,10.0.1.201'
|
|*/
$config['proxy_ips'] = '';

/* End of file config.php */
/* Location: ./application/config/config.php */
|-----|
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/*
|-----|
| DATABASE CONNECTIVITY SETTINGS
|-----|
| This file will contain the settings needed to access your database.
|
| For complete instructions please consult the 'Database Connection'
| page of the User Guide.
|
|-----|
| EXPLANATION OF VARIABLES
|-----|
|
| ['hostname'] The hostname of your database server.
| ['username'] The username used to connect to the database
| ['password'] The password used to connect to the database
| ['database'] The name of the database you want to connect to
| ['dbdriver'] The database type. ie: mysql. Currently supported:
|               mysql, mysqli, postgres, odbc, mssql, sqlite, oci8
| ['dbprefix'] You can add an optional prefix, which will be added
|               to the table name when using the Active Record class
| ['pconnect'] TRUE/FALSE - Whether to use a persistent connection
| ['db_debug'] TRUE/FALSE - Whether database errors should be displayed.
| ['cache_on'] TRUE/FALSE - Enables/disables query caching
| ['cachedir'] The path to the folder where cache files should be stored
| ['char_set'] The character set used in communicating with the database
| ['dbcollat'] The character collation used in communicating with the database
|               NOTE: For MySQL and MySQLi databases, this setting is only
|               used
|               as a backup if your server is running PHP < 5.2.3 or MySQL <
|               5.0.7
|               (and in table creation queries made with DB Forge).
|               There is an incompatibility in PHP with
|               mysql_real_escape_string() which
|               can make your site vulnerable to SQL injection if you are
|               using a
|               multi-byte character set and are running versions lower than
|               these.
|               Sites using Latin-1 or UTF-8 database character set and
|               collation are unaffected.
| ['swap_pre'] A default table prefix that should be swapped with the dbprefix
| ['autoinit'] Whether or not to automatically initialize the database.
| ['stricton'] TRUE/FALSE - forces 'Strict Mode' connections
|               - good for ensuring strict SQL while
|               developing
|
| The $active_group variable lets you choose which connection group to
| make active. By default there is only one group (the 'default' group).
|
| The $active_record variables lets you determine whether or not to load
| the active record class
|
|*/
$active_group = 'default';
$active_record = TRUE;

```



```

$db['default']['hostname'] = 'localhost'; //hostname is localhost
$db['default']['username'] = 'FoodIntakeMon'; //username is root
$db['default']['password'] = 'pnUvqYprynEdNyZn'; //no password
$db['default']['database'] = 'FoodIntakeMon'; //database is fim
$db['default']['dbdriver'] = 'mysql';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;

```

```

/* End of file database.php */
/* Location: ./application/config/database.php */

```

```

<?php
/*$config = array(
    'appId' => '411694662300964',
    'secret' => '4449887dcad9b56c0e8dbc66bf63b331'
);*/
/*$config['appId'] = '411694662300964';
$config['secret'] = '4449887dcad9b56c0e8dbc66bf63b331';*/
$config['appId'] = '593833454087083';
$config['secret'] = 'a9a92c6d827ae7ed0503f689141d2b05';
?>

```

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/*

```

---

#### URI ROUTING

---

This file lets you re-map URI requests to specific controller functions.

Typically there is a one-to-one relationship between a URL string and its corresponding controller class/method. The segments in a URL normally follow this pattern:

```
example.com/class/method/id/
```

In some instances, however, you may want to remap this relationship so that a different class/function is called than the one corresponding to the URL.

Please see the user guide for complete details:

```
http://codeigniter.com/user_guide/general/routing.html
```

---

#### RESERVED ROUTES

---

There are two reserved routes:

```
$route['default_controller'] = 'welcome';
```

This route indicates which controller class should be loaded if the URI contains no data. In the above example, the "welcome" class would be loaded.

```
$route['404_override'] = 'errors/page_missing';
```

This route will tell the Router what URI segments to use if those provided in the URL cannot be matched to a valid route.

```
*/
```

```

$route['default_controller'] = "login"; //default page login.php controller linked to
login_index.php view
$route['MY_Pagination/{:any}'] = 'login/index';
$route['404_override'] = '';

//$default_controller = "login";
//$controller_exceptions = array('verifylogin','logout','home','home_user','home_admin','home_cmanager');

//$route['default_controller'] = $default_controller;
//$route["^(?!\\b".implode('\\b|\\b', $controller_exceptions). "\\b).*"] = $default_controller
.'/$1';
//$route['404_override'] = '';
//$route['show_fitness_profile'] = 'home_user/show_fitness_profile';
//$route['weightmonitor_getdata'] = 'home_user/weightmonitor_getdata';

/* End of file routes.php */
/* Location: ./application/config/routes.php */

```

```

<?php
$config['tweet_consumer_key'] = "Y9wHRf3wQOAd5KLaW5LRwqiu2";
$config['tweet_consumer_secret'] = "vchyOy5I2lyJUaqBmkqkFUzXSsuf4ayrMeioxkghag958LkDBI";

```

?>

---

## B.2 Controllers

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
session_start(); //we need to call PHP's session object to access it through CI
```

```
class Home extends CI_Controller {

    function __construct() {
        parent::__construct();
    }

    function index() {
        if($this->session->userdata('logged_in')) {
            $session_data = $this->session->userdata('logged_in');
            $data['username'] = $session_data['username'];
            $data['first_name'] = $session_data['first_name'];
            $data['user_type'] = $session_data['user_type'];
            $this->load->view('home_index', $data);
        }
        else {
            //If no session, redirect to login page
            redirect('login', 'refresh');
        }
    }
}

?>
```

---

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
session_start(); //we need to call PHP's session object to access it through CI
```

```
class Home_Admin extends CI_Controller {

    function __construct() {
        parent::__construct();
        $this->load->model('home_admin_model');
    }

    //INDEX
    function index() {
        $this->load->model('home_admin_model');
        $data['query'] = "";
        // $this->load->vars($data);
        if($this->session->userdata('logged_in')) {
            $session_data = $this->session->userdata('logged_in');
            $data['user_id'] = $session_data['user_id'];
            $data['username'] = $session_data['username'];
            $data['first_name'] = $session_data['first_name'];
            $data['last_name'] = $session_data['last_name'];
            $data['user_type'] = $session_data['user_type'];
            $data['email_address'] = $session_data['email_address'];
            $data['gender'] = $session_data['gender'];
            $data['birthday'] = $session_data['birthday'];
            $this->load->view('home_admin_index', $data);
        }
        else {
            //If no session, redirect to login page
            redirect('login', 'refresh');
        }
    }

    //LOGOUT
    function logout() {
        $this->session->unset_userdata('logged_in');
        session_destroy();
        redirect('login', 'refresh');
    }

    //USER SEARCH CONTROLLERS
    function suggestions() {
        $this->load->model('home_admin_model');
        $term = $this->input->post('term', TRUE);

        if (strlen($term) < 1) break;

        $rows = $this->home_admin_model->GetAutocomplete(array('keyword' => $term));

        $json_array = array();
        foreach ($rows as $row)
            array_push($json_array, $row->user_id, $row->username, $row->first_name, $row
                ->middle_name, $row->last_name, $row->user_type, $row->email_address, $row
                ->gender, $row->birthday);
        echo json_encode($json_array);
    }

    function search() {

        $this->load->model('home_admin_model');

        $data['query'] = $this->home_admin_model->get_search();

        // $data['query'] = $this->home_admin_model->get_search();
        if($this->session->userdata('logged_in')) {
```

```

        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('home_admin_index', $data);
    }
}

//ADD USER CONTROLLERS
function add_user() {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
    $this->load->view('add_user_index');
}

function save() {
    $this->load->library('form_validation');
    // field name, error message, validation rules
    $this->form_validation->set_rules('username', 'Username', 'trim|required|
    alpha_numeric|min_length[4]|xss_clean|callback_check_user_ci');
    $this->form_validation->set_rules('password', 'Password', 'trim|
    required|alpha_numeric|min_length[8]');
    $this->form_validation->set_rules('confirm_password', 'Confirm
    Password', 'trim|required|alpha_numeric|matches[password]');
    $this->form_validation->set_rules('first_name', 'First Name', 'trim|
    required|alpha_numeric|xss_clean');
    $this->form_validation->set_rules('middle_name', 'Middle Name', '
    alpha_numeric|xss_clean');
    $this->form_validation->set_rules('last_name', 'Last Name', 'trim|
    required|alpha_numeric|xss_clean');
    $this->form_validation->set_rules('birthdate', 'Birthdate', '
    required|xss_clean');
    $this->form_validation->set_rules('genderRadio', 'Gender', 'required
    ');
    $this->form_validation->set_rules('userTypeRadio', 'User Type', '
    required');
    $this->form_validation->set_rules('email_address', 'Email Address',
    'required|valid_email|callback_check_email_ci');

    if($this->form_validation->run() == FALSE) {
        // $this->index();
        $this->load->view('add_user_index');
    }
    else {
        $this->load->model('home_admin_model');
        if($this->input->post('submit')) {
            $this->home_admin_model->process();
            // $message = "Registration Successful!";
            // if ((isset($message) && ($message != ''))) {
            // echo '<script>
            // location.replace("../home_admin");
            // </script>';
            // }
            $this->session->set_flashdata('success_msg', 'Registration
            Successful. ');
            redirect('home_admin', 'refresh');
        }
    }
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

public function check_user_ci() {
    $usr=$this->input->post('username');
    $result=$this->home_admin_model->check_user_exist($usr);
    if($result) {
        $this->form_validation->set_message(__FUNCTION__, 'Username already
        exists. ');
        return false;
    }
    else {
        return true;
    }
}

public function check_user() {
    $usr=$this->input->post('name');
    $result=$this->home_admin_model->check_user_exist($usr);
    if($result) {
        echo "false";
    }
    else {
        echo "true";
    }
}

```

```

    }
}

public function check_email_ci() {
    $email=$this->input->post('email_address');
    $result=$this->home_admin_model->check_email_exist($email);
    if($result) {
        $this->form_validation->set_message(__FUNCTION__, 'E-mail Address
            already registered. ');
        return false;
    }
    else {
        return true;
    }
}

public function check_email() {
    $email=$this->input->post('email');
    $result=$this->home_admin_model->check_email_exist($email);
    if($result) {
        echo "false";
    }
    else {
        echo "true";
    }
}

//DELETE USER CONTROLLER
function delete_user($username) {
    $this->home_admin_model->del_user($username);
    // $message = "User was successfully deleted!";
    //if ((isset($message)) && ($message != '')) {
        //echo '<script>
            //location.replace("../home_admin");
        //</script>';
    //}
    $this->session->set_flashdata('success_msg', 'User was successfully
        deleted. ');
        redirect('home_admin', 'refresh');
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//EDIT USER CONTROLLER
function edit_user($username) {
    $item = $this->home_admin_model->get_user($username);
    $this->load->library('form_validation');
    // field name, error message, validation rules
    $this->form_validation->set_rules('username', 'Username', 'trim|required|
        min_length[4]|xss_clean');
    $this->form_validation->set_rules('first_name', 'First Name', 'trim|required|
        xss_clean');
    $this->form_validation->set_rules('middle_name', 'Middle Name', 'xss_clean');
    $this->form_validation->set_rules('last_name', 'Last Name', 'trim|required|
        xss_clean');
    $this->form_validation->set_rules('birthdate', 'Birthdate', 'required|
        xss_clean');
    $this->form_validation->set_rules('genderRadio', 'Gender', 'required');
    $this->form_validation->set_rules('userTypeRadio', 'User Type', 'required');
    $this->form_validation->set_rules('email_address', 'Email Address', 'required|
        valid_email');

    if (isset($_POST) && !empty($_POST)) {
        $data = array(
            'username' => $this->input->post('username'),
            'first_name' => $this->input->post('first_name'),
            'middle_name' => $this->input->post('middle_name'),
            'last_name' => $this->input->post('last_name'),
            'gender' => $this->input->post('genderRadio'),
            'user_type' => $this->input->post('userTypeRadio'),
            'email_address' => $this->input->post('email_address'),
            'birthday' => $this->input->post('birthdate'),
            'status' => $this->input->post('statusRadio')
        );
    }

    if($this->form_validation->run() == FALSE) {
        // $this->index();
    }
    else {
        $this->load->model('home_admin_model');
        if($this->input->post('submit')) {
            $this->home_admin_model->update_profile($username,
                $data);
            // $message = "Update User Successful!";
            //if ((isset($message)) && ($message != '')) {
                //echo '<script>
                    //location.replace("../home_admin");
                //</script>';
            //}
            $this->session->set_flashdata('success_msg', '
                Update User Successful. ');
            redirect('home_admin', 'refresh');
        }
    }
}

```

```

    }
}
$data['item'] = $item;
$data['username'] = $this->form_validation->set_value('username', $item['username']);
$data['first_name'] = $this->form_validation->set_value('first_name', $item['first_name']);
$data['middle_name'] = $this->form_validation->set_value('middle_name', $item['middle_name']);
$data['last_name'] = $this->form_validation->set_value('last_name', $item['last_name']);
$data['birthday'] = $this->form_validation->set_value('birthdate', $item['birthday']);
$data['gender'] = $this->form_validation->set_value('genderRadio', $item['gender']);
$data['user_type'] = $this->form_validation->set_value('userTypeRadio', $item['user_type']);
$data['email_address'] = $this->form_validation->set_value('email_address', $item['email_address']);

if($this->session->userdata('logged_in')) {
    $session_data = $this->session->userdata('logged_in');
    $data['username'] = $session_data['username'];
}
else {
    //If no session, redirect to login page
    redirect('login', 'refresh');
}

$this->load->view('edit_user_index', $data);
}

//EDIT PROFILE CONTROLLER
function edit_profile_user() {
    $this->load->library('form_validation');
    // field name, error message, validation rules
    $this->form_validation->set_rules('username', 'Username', 'trim|required|min-length[4]|xss-clean');
    $this->form_validation->set_rules('first_name', 'First Name', 'trim|required|xss-clean');
    $this->form_validation->set_rules('middle_name', 'Middle Name', 'xss-clean');
    $this->form_validation->set_rules('last_name', 'Last Name', 'trim|required|xss-clean');
    $this->form_validation->set_rules('birthdate', 'Birthdate', 'required|xss-clean');
    $this->form_validation->set_rules('genderRadio', 'Gender', 'required');
    $this->form_validation->set_rules('userTypeRadio', 'User Type', 'required');
    $this->form_validation->set_rules('email_address', 'Email Address', 'required|valid_email');

    if (isset($_POST) && !empty($_POST)) {
        $data = array(
            'username' => $this->input->post('username'),
            'first_name' => $this->input->post('first_name'),
            'middle_name' => $this->input->post('middle_name'),
            'last_name' => $this->input->post('last_name'),
            'gender' => $this->input->post('genderRadio'),
            'user_type' => $this->input->post('userTypeRadio'),
            'email_address' => $this->input->post('email_address'),
            'birthday' => $this->input->post('birthdate')
        );

        $username = $this->input->post('username');

        if($this->form_validation->run() == FALSE) {
            // $this->index();
        }
        else {
            $this->session->set_userdata('logged_in', $data);
            $this->load->model('home_admin_model');
            if($this->input->post('submit')) {
                $this->home_admin_model->update_profile($username, $data);
                // $message = "Profile Update Successful!";
                // if ((isset($message) && ($message != ''))) {
                // echo '<script>
                // location.replace("../home_admin");
                // </script >';
                // }

                $this->session->set_flashdata('success_msg', 'Profile Update Successful. ');
                redirect('home_admin', 'refresh');
            }
        }
    }
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['middle_name'] = $session_data['middle_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
    }
    else {

```

```

        //If no session , redirect to login page
        redirect('login', 'refresh');
    }

    $this->load->view('edit_profile_admin_index', $data);
}

//EDIT PASSWORD CONTROLLER
function edit_password_user() {
    $this->load->library('form_validation');
    // field name, error message, validation rules
    $this->form_validation->set_rules('old_password', 'Old Password', 'trim|required|min_length[8]|callback_check_old_password');
    $this->form_validation->set_rules('password', 'Password', 'trim|required|min_length[8]');
    $this->form_validation->set_rules('confirm_password', 'Confirm Password', 'trim|required|matches[password]');

    if (isset($_POST) && !empty($_POST)) {
        $data = array(
            'password' => MD5($this->input->post('password')),
            'confirm_password' => MD5($this->input->post('confirm_password'))
        );
    }

    $username = $this->input->post('username');

    if($this->form_validation->run() == FALSE) {
        // $this->index();
    }
    else {
        $this->load->model('home_admin_model');
        if($this->input->post('submit')) {
            $this->home_admin_model->update_password($username, $data);
            // $message = "Password Change Successful!";
            // if ((isset($message) && ($message != ''))) {
            //     echo '<script>
            //         //location.replace("../home.admin");
            //     </script>';
            // }
            $this->session->set_flashdata('success_msg', 'Password Change Successful. ');
            redirect('home.admin', 'refresh');
        }
    }

    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session , redirect to login page
        redirect('login', 'refresh');
    }

    $this->load->view('edit_password_admin_index', $data);
}

public function check_old_password() {
    $opwd=MD5($this->input->post('old_password'));
    $result=$this->home_admin_model->check_old_password_exist($opwd);
    if($result) {
        return true;
    }
    else {
        $this->form_validation->set_message(__FUNCTION__, 'Incorrect Old Password. ');
        return false;
    }
}
}
?>

```

---

```

<?php if (! defined('BASEPATH')) exit('No direct script access allowed');
session_start(); //we need to call PHP's session object to access it through CI

```

```

class Home.Cmanager extends CI_Controller {

    function __construct() {
        parent::__construct();
        $this->load->helper(array('form', 'url'));
        $this->load->model('home_cmanager_model');
    }

    function index() {
        $this->load->model('home_cmanager_model');
        $data['query_fel'] = "";
        $data['query_ht'] = "";
        if($this->session->userdata('logged_in')) {
            $session_data = $this->session->userdata('logged_in');
            $data['user_id'] = $session_data['user_id'];
            $data['username'] = $session_data['username'];
            $data['first_name'] = $session_data['first_name'];
            $data['last_name'] = $session_data['last_name'];
        }
    }
}

```

```

        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('home_cmanager_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//LOGOUT
function logout() {
    $this->session->unset_userdata('logged_in');
    session_destroy();
    redirect('login', 'refresh');
}

public function check_user() {
    $usr=$this->input->post('name');
    $result=$this->home_cmanager_model->check_user_exist($usr);
    if($result) {
        echo "false";
    }
    else {
        echo "true";
    }
}

public function check_email() {
    $eml=$this->input->post('email');
    $result=$this->home_cmanager_model->check_email_exist($eml);
    if($result) {
        echo "false";
    }
    else {
        echo "true";
    }
}

//EDIT PROFILE CONTROLLER
function edit_profile_user() {
    $this->load->library('form_validation');
    // field name, error message, validation rules
    $this->form_validation->set_rules('username', 'Username', 'trim|required|
        min_length[4]|xss_clean');
    $this->form_validation->set_rules('first_name', 'First Name', 'trim|required|
        xss_clean');
    $this->form_validation->set_rules('middle_name', 'Middle Name', 'xss_clean');
    $this->form_validation->set_rules('last_name', 'Last Name', 'trim|required|
        xss_clean');
    $this->form_validation->set_rules('birthdate', 'Birthdate', 'required|
        xss_clean');
    $this->form_validation->set_rules('genderRadio', 'Gender', 'required');
    $this->form_validation->set_rules('userTypeRadio', 'User Type', 'required');
    $this->form_validation->set_rules('email_address', 'Email Address', 'required|
        valid_email');

    if (isset($_POST) && !empty($_POST)) {
        $data = array(
            'username' => $this->input->post('username'),
            'first_name' => $this->input->post('first_name'),
            'middle_name' => $this->input->post('middle_name'),
            'last_name' => $this->input->post('last_name'),
            'gender' => $this->input->post('genderRadio'),
            'user_type' => $this->input->post('userTypeRadio'),
            'email_address' => $this->input->post('email_address'),
            'birthdate' => $this->input->post('birthdate')
        );

        $username = $this->input->post('username');

        if($this->form_validation->run() == FALSE) {
            // $this->index();
        }
        else {
            $this->session->set_userdata('logged_in', $data);
            $this->load->model('home_cmanager_model');
            if($this->input->post('submit')) {
                $this->home_cmanager_model->update_profile($username,
                    $data);
                // $message = "Profile Update Successful!";
                // if ((isset($message)) && ($message != '')) {
                //     //echo '<script>
                //         //location.replace("../
                //             home_cmanager");
                //     //</script >';
                // }
                $this->session->set_flashdata('success_msg', '
                    Profile Update Successful. ');
                redirect('home_cmanager', 'refresh');
            }
        }
    }
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
    }
}

```

```

        $data['middle_name'] = $session_data['middle_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }

    $this->load->view('edit_profile_cmanager_index', $data);
}

//EDIT PASSWORD CONTROLLER
function edit_password_user() {
    $this->load->library('form_validation');
    // field name, error message, validation rules
    $this->form_validation->set_rules('old_password', 'Old Password', 'trim|required|min_length[8]|callback_check_old_password');
    $this->form_validation->set_rules('password', 'Password', 'trim|required|min_length[8]');
    $this->form_validation->set_rules('confirm_password', 'Confirm Password', 'trim|required|matches[password]');

    if (isset($_POST) && !empty($_POST)) {
        $data = array(
            'password' => MD5($this->input->post('password')),
            'confirm_password' => MD5($this->input->post('confirm_password'))
        );

        $username = $this->input->post('username');

        if($this->form_validation->run() == FALSE) {
            //$this->index();
        }
        else {
            $this->load->model('home_cmanager_model');
            if($this->input->post('submit')) {
                $this->home_cmanager_model->update_password($username, $data);
                //$message = "Password Change Successful!";
                //if ((isset($message) && ($message != ''))) {
                //echo '<script>
                //location.replace("../home_cmanager");
                //</script >';
                //}
                $this->session->set_flashdata('success_msg', 'Password Change Successful. ');
                redirect('home_cmanager', 'refresh');
            }
        }
    }

    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }

    $this->load->view('edit_password_cmanager_index', $data);
}

public function check_old_password() {
    $opwd=MD5($this->input->post('old_password'));
    $result=$this->home_cmanager_model->check_old_password_exist($opwd);
    if($result) {
        return true;
    }
    else {
        $this->form_validation->set_message(__FUNCTION__, 'Incorrect Old Password. ');
        return false;
    }
}

//ADD FOOD EXCHANGE LIST CONTROLLERS
function check_felitem_exists() {
    $food_name = $this->input->post('food_name');
    $result=$this->home_cmanager_model->check_felitem_exist($food_name);
    if($result) {
        $this->form_validation->set_message(__FUNCTION__, 'Food Item already exists. ');
        return false;
    }
    else {
        return true;
    }
}

function add_felitem() {
    if($this->session->userdata('logged_in')) {
        $this->load->model('home_cmanager_model');
    }
}

```



```

        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['lg'] = $this->home_cmanager_model->get_list_group();
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
    $this->load->view('add_food_exchange_list_index', $data);
}

public function buildDropFEL(){
    $this->load->model('home_cmanager_model');
    //set selected list_group from POST
    $li_group = $this->input->post('list_group',TRUE);
    //run the query for the food group we specified earlier
    $fgData['fgDrop'] = $this->home_cmanager_model->get_food_group($li_group);
}

$output = null;

foreach ($fgData['fgDrop']->result() as $row){
    //here we build a dropdown item line for each query result
    $output .= "<option value=\"" . $row->list_group . "\">" . $row->food_group . "</option>";
}

echo $output;
}

public function buildDropFEL2(){
    $this->load->model('home_cmanager_model');
    //set selected list_group from POST
    $li_group = $this->input->post('list_group',TRUE);
    //run the query for the food group we specified earlier
    $fgcData['fgcDrop'] = $this->home_cmanager_model->get_food_group_category($li_group);
}

$output = null;

foreach ($fgcData['fgcDrop']->result() as $row){
    //here we build a dropdown item line for each query result
    $output .= "<option value=\"" . $row->list_group . "\">" . $row->food_group_category . "</option>";
}

echo $output;
}

function add_felitem_save() {
    $this->load->library('form_validation');
    // field name, error message, validation rules
    $this->form_validation->set_rules('list_group', 'List Group');
    $this->form_validation->set_rules('food_group', 'Food Group');
    $this->form_validation->set_rules('food_group_category', 'Food Group Category');
    $this->form_validation->set_rules('food_name', 'Food Name', 'required|callback_check_felitem_exists');
    $this->form_validation->set_rules('measure', 'Measure', 'required');
    $this->form_validation->set_rules('carbohydrate', 'Carbohydrate', 'required|numeric');
    $this->form_validation->set_rules('protein', 'Protein', 'required|numeric');
    $this->form_validation->set_rules('fat', 'Fat', 'required|numeric');
    $this->form_validation->set_rules('energy', 'Energy', 'required|numeric');
    $this->form_validation->set_rules('weight_as_purchased', 'Weight as Purchased', 'required|numeric');
    $this->form_validation->set_rules('weight_edible_portion', 'Weight Edible Portion', 'required|numeric');

    //echo '<pre>';
    //print_r($_POST);
    //echo '</pre>';
    //die();

    if($this->form_validation->run() == FALSE) {
        // $this->index();
        $this->load->view('add_food_exchange_list_index');
    }
    else {
        $this->load->model('home_cmanager_model');
        if($this->input->post('submit')) {
            $this->home_cmanager_model->process_felitem();
            // $message = "Adding of Food Exchange List Item Successful!";
            // if ((isset($message) && ($message != ''))) {
            // echo '<script>
            // location.replace("../home_cmanager");
            // </script>';
            // }
            $this->session->set_flashdata('success_msg', 'Adding of Food Exchange List Item Successful. ');
            redirect('home_cmanager', 'refresh');
        }
    }
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

```

```

}

//DELETE FEL ITEM
function delete_fel_item($food_exchange_list_id) {
    $this->home_cmanager_model->del_fel_item($food_exchange_list_id);
    //$message = "Food Exchange List Item was successfully deleted!";
    //if ((isset($message)) && ($message != '')) {
        //echo '<script>
            //location.replace("../home.cmanager");
        //</script>';
    //}
    $this->session->set_flashdata('success-msg', 'Food Exchange List Item
        was successfully deleted. ');
        redirect('home_cmanager', 'refresh');
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//ADD HEALTH TIP CONTROLLERS
function add_healthtip() {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
    $this->load->view('add-health-tip-index');
}

function add_healthtip_save() {
    $this->load->library('form-validation');
    // field name, error message, validation rules
    $this->form_validation->set_rules('health_tip', 'Health Tip', 'required');

    if($this->form_validation->run() == FALSE) {
        //$this->index();
        $this->load->view('add-health-tip-index');
    }
    else {
        $this->load->model('home_cmanager_model');
        if($this->input->post('submit')) {
            $this->home_cmanager_model->process_healthtip();
            //$message = "Adding of Health Tip Successful!";
            //if ((isset($message)) && ($message != '')) {
                //echo '<script>
                    //location.replace("../home.cmanager");
                //</script>';
            //}
            $this->session->set_flashdata('success-msg', 'Adding of Health
                Tip Successful. ');
                redirect('home_cmanager', 'refresh');
        }
    }
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//DELETE HEALTH TIP
function delete_health_tip($health_tip_id) {
    $this->home_cmanager_model->del_health_tip($health_tip_id);
    //$message = "Health Tip was successfully deleted!";
    //if ((isset($message)) && ($message != '')) {
        //echo '<script>
            //location.replace("../home.cmanager");
        //</script>';
    //}
    $this->session->set_flashdata('success-msg', 'Health Tip was
        successfully deleted. ');
        redirect('home_cmanager', 'refresh');
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

function search_fel() {
    $this->load->model('home_cmanager_model');

    $data['query_fel'] = $this->home_cmanager_model->get_search_fel();

    $data['query_ht'] = "";
}

```

```

if($this->session->userdata('logged_in')) {
    $session_data = $this->session->userdata('logged_in');
    $data['username'] = $session_data['username'];
    $data['first_name'] = $session_data['first_name'];
    $data['last_name'] = $session_data['last_name'];
    $data['user_type'] = $session_data['user_type'];
    $data['email_address'] = $session_data['email_address'];
    $data['gender'] = $session_data['gender'];
    $data['birthday'] = $session_data['birthday'];
    $this->load->view('home_cmanager_index', $data);
}
else {
    //If no session, redirect to login page
    redirect('login', 'refresh');
}
}

function search_ht() {
    $this->load->model('home_cmanager_model');
    $data['query_ht'] = $this->home_cmanager_model->get_search_ht();
    $data['query_fel'] = "";
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
    }
    $this->load->view('home_cmanager_index', $data);
}
else {
    //If no session, redirect to login page
    redirect('login', 'refresh');
}
}

//VIEW CREATED RECIPES
function show_created_recipes() {
    $data['query'] = $this->home_cmanager_model->process_view_created_recipes();
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
    }
    $this->load->view('view_created_recipes_cmanager_index', $data);
}
else {
    //If no session, redirect to login page
    redirect('login', 'refresh');
}
}

//EDIT CREATED RECIPES
function edit_created_recipe($recommended_recipe_id) {
    $item = $this->home_cmanager_model->get_created_recipe($recommended_recipe_id);
    $this->load->library('form_validation');
    // field name, error message, validation rules
    // $this->form_validation->set_rules('own_recipe_id', 'Username', 'trim|
    // required|min_length[4]|xss_clean');
    // $this->form_validation->set_rules('user_id', 'First Name', 'trim|required|
    // xss_clean');
    $this->form_validation->set_rules('health_condition', 'Health Condition', '
    required');
    $this->form_validation->set_rules('meal_type', 'Meal Type', 'required');
    $this->form_validation->set_rules('recipe_name', 'Recipe Name', 'required');
    $this->form_validation->set_rules('serving_sugg', 'Serving Size', 'required|
    numeric');
    $this->form_validation->set_rules('ingredients', 'Ingredients', '
    callback_health_conditions_restrictions');
    $this->form_validation->set_rules('preparation_procedure', 'Procedure', '
    required');
    $this->form_validation->set_rules('calories', 'Calories', 'required|numeric');
    $this->form_validation->set_rules('carbohydrate', 'Carbohydrates', 'required|
    numeric');
    $this->form_validation->set_rules('protein', 'Protein', 'required|numeric');
    $this->form_validation->set_rules('fat', 'Fat', 'required|numeric');

    if (isset($_POST) && !empty($_POST)) {
        $data = array(
            'own_recipe_id' => $this->input->post('own_recipe_id'),
            'user_id' => $this->input->post('user_id'),
            'health_condition' => $this->input->post('health_condition'),
            'health_condition' => implode(",",$this->input->post('
            health_condition')),
            'meal_type' => $this->input->post('meal_type'),
            'recipe_name' => $this->input->post('recipe_name'),
            'serving_sugg' => $this->input->post('serving_sugg'),
            'preparation_procedure' => $this->input->post('
            preparation_procedure'),
            'calories' => $this->input->post('calories'),
            'carbohydrate' => $this->input->post('carbohydrate'),
            'protein' => $this->input->post('protein'),

```

```

        'fat' => $this->input->post('fat')
    );

    $data2 = array();

    $data2['food_name'] = $this->input->post('ingredients');
    $data2['food_group'] = $this->input->post('food_group');
    $data2['measure'] = $this->input->post('ingredient_measure');
    $data2['serving_size'] = $this->input->post('ingredient_quantity');
    $data2['ingredient_id'] = $this->input->post('ingredient_id');

    if($this->form-validation->run() == FALSE) {
        // $this->index();
    }
    else {
        $this->load->model('home_cmanager_model');
        if($this->input->post('submit')) {
            $this->home_cmanager_model->update_recipe(
                $recommended_recipe_id, $data, $data2);
            // $message = "Edit Recipe Successful!";
            // if ((isset($message) && ($message != ''))) {
                // echo '<script>
                // location.replace("../home_cmanager")
                // </script >';
            // }

            $this->session->set_flashdata('success_msg', '
                Edit Recipe Successful. ');
            redirect('home_cmanager', 'refresh');
        }
    }
}
$data['item'] = $item;
// $data['own_recipe_id'] = $this->form-validation->set_value('own_recipe_id',
// $item['own_recipe_id']);
// $data['user_id'] = $this->form-validation->set_value('user_id', $item['
// user_id']);
// $data['health_condition'] = $this->form-validation->set_value('
// health_condition', $item['health_condition']);
// $data['meal_type'] = $this->form-validation->set_value('meal_type', $item['
// meal_type']);
// $data['recipe_name'] = $this->form-validation->set_value('recipe_name',
// $item['recipe_name']);
// $data['serving_sugg'] = $this->form-validation->set_value('serving_sugg',
// $item['serving_sugg']);

// $data['food_name'] = $this->form-validation->set_value('ingredients', $item
// ['food_name']);
// $data['food_group'] = $this->form-validation->set_value('food_group', $item
// ['food_group']);
// $data['measure'] = $this->form-validation->set_value('ingredient_measure',
// $item['measure']);
// $data['serving_size'] = $this->form-validation->set_value('
// ingredient_quantity', $item['serving_size']);

// $data['preparation_procedure'] = $this->form-validation->set_value('
// preparation_procedure', $item['preparation_procedure']);
// $data['calories'] = $this->form-validation->set_value('calories', $item['
// calories']);
// $data['carbohydrate'] = $this->form-validation->set_value('carbohydrate',
// $item['carbohydrate']);
// $data['protein'] = $this->form-validation->set_value('protein', $item['
// protein']);
// $data['fat'] = $this->form-validation->set_value('fat', $item['fat']);

// print_r($item);
// die();

if($this->session->userdata('logged_in')) {
    $session_data = $this->session->userdata('logged_in');
    $data['username'] = $session_data['username'];
}
else {
    // If no session, redirect to login page
    redirect('login', 'refresh');
}
$this->load->view('edit_recipe_cmanager_index', $data);
}

// VIEW SELECTED RECIPE
function view_created_recipe($recommended_recipe_id) {
    $this->load->model('home_cmanager_model');
    $data['query'] = $this->home_cmanager_model->get_recipe_contents(
        $recommended_recipe_id);
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('view_recipe_cmanager_index', $data);
    }
}

```

```

else {
    //If no session, redirect to login page
    redirect('login', 'refresh');
}
}

//DELETE CREATED RECIPE
function delete_created_recipe($recommended_recipe_id) {
    $this->home_cmanager_model->del_created_recipe($recommended_recipe_id);
    //$message = "Recipe was successfully deleted!";
    //if ((isset($message)) && ($message != '')) {
        //echo '<script>
        //location.replace("../show_created_recipes");
        //</script>';
    //}
    $this->session->set_flashdata('success_msg', 'Recipe was successfully
    deleted. ');
    redirect('home_cmanager/show_created_recipes
    ', 'refresh');
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//DELETE INGREDIENTS
function delete_ingredient() {
    $this->home_cmanager_model->del_ingredient($this->input->post('ingredient_id')
    );
}

//CREATE RECIPE CONTROLLER
function add_recipe () {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('add_recipe_cmanager_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

function add_recipe_save () {
    $this->load->library('form_validation');
    // field name, error message, validation rules
    $this->form_validation->set_rules('health_condition', 'Health Condition', '
    required');
    $this->form_validation->set_rules('meal_type', 'Meal Type', 'required');
    $this->form_validation->set_rules('recipe_name', 'Recipe Name', 'required');
    $this->form_validation->set_rules('serving_sugg', 'Serving Size', 'required|
    numeric');
    $this->form_validation->set_rules('ingredients', 'Ingredients', '
    callback_health_conditions_restrictions');
    $this->form_validation->set_rules('preparation_procedure', 'Procedure', '
    required');
    $this->form_validation->set_rules('calories', 'Calories', 'required|numeric');
    $this->form_validation->set_rules('carbohydrate', 'Carbohydrates', 'required|
    numeric');
    $this->form_validation->set_rules('protein', 'Protein', 'required|numeric');
    $this->form_validation->set_rules('fat', 'Fat', 'required|numeric');

    if($this->form_validation->run() == FALSE) {
        //echo '<pre>'.print_r($this->input->post(),TRUE).'\</pre>';
        //die();
        //$this->index();
        //if($this->input->post())
    //}
    // Merge the submitted data with the current data
    // $data = $this->input->post();
    //echo '<pre>'.print_r($data).'\</pre>';
    //die();
    //}
    // $data = $this->input->post();
    // $ingredients = $this->input->post('ingredients');
    //echo '<pre>'.print_r($ingredients).'\</pre>';
    //die();
    //echo '<pre>'.print_r($data).'\</pre>';
    //die();

    $this->load->view('add_recipe_cmanager_index'/*, $data*/);
}
else {
    $this->load->model('home_cmanager_model');
    if($this->input->post('submit')) {
        /*Create config for image library*/
        $this->load->library('image_lib');
    }
}
}

```

```

$config['image_library'] = 'gd2';

/* Create config for upload library*/
$config['upload_path'] = './uploads/';
$config['allowed_types'] = 'gif|jpg|png';
$config['max_size'] = '0';
$config['max_width'] = '0';
$config['max_height'] = '0';
$this->load->library('upload', $config);

if( ! $this->upload->do_upload('userfile')) {
    if (!empty($_FILES['userfile']['name'])) {
        // $error = array('error' => $this->
        upload->display_errors());
        $error = $this->upload->display_errors
        ();
        $data['error_display'] = $error;
        $this->load->view('
        add_recipe_cmanager_index', $data)
        ;
    } else {

        $filename = 'default' ;
        $fileExtension = 'png';

        $data3['raw_name'] = $filename;
        $data3['file_ext'] = $fileExtension;

        $file=array(
            'img_name'=>$data3['raw_name
            '],
            'thumb_name'=>$data3['raw_name
            '].'_thumb',
            'ext'=>$data3['file_ext'],
            'upload_date'=>time()
        );

        $this->home_cmanager_model->process_addrecipe(
        $file);

    }
} else {
    $data3=$this->upload->data();
    $path_info = pathinfo($_FILES["userfile"]["
    name"]);
    $filename = $path_info['filename'];
    $fileExtension = $path_info['extension'];

    // $data3=$this->upload->data();
    // $this->thumb($data3);

    $data3['raw_name'] = $filename;
    $data3['file_ext'] = $fileExtension;

    $file=array(
        'img_name'=>$data3['raw_name
        '],
        'thumb_name'=>$data3['raw_name
        '].'_thumb',
        'ext'=>$data3['file_ext'],
        'upload_date'=>time()
    );

    $this->home_cmanager_model->process_addrecipe(
    $file);

    $data3 = array('upload_data' => $this->upload
    ->data());
}

// $message = "Creating of Recipe Successful!";
// if ((isset($message)) && ($message != '')) {
// echo '<script>
// location.replace("../
// home_cmanager/
// show_created_recipes");
// </script >';
// }
$this->session->set_flashdata('success_msg', '
Creating of Recipe Successful. ');
redirect('home_cmanager/show_created_recipes
', 'refresh');
}
}
if($this->session->userdata('logged_in')) {
    $session_data = $this->session->userdata('logged_in');
    $data['username'] = $session_data['username'];
}
else {
    // If no session, redirect to login page
    redirect('login', 'refresh');
}
}

/*function thumb($data3) {
    $config['image_library'] = 'gd2';
    $config['source_image'] =$data3['full_path'];
    $config['create_thumb'] = TRUE;
    $config['maintain_ratio'] = TRUE;
}

```

```

$config['width'] = 275;
$config['height'] = 250;
$this->load->library('image_lib', $config);
$this->image_lib->resize();
}*/

function health_conditions_restrictions($ingredients) {
    $health_condition = $_POST['health_condition'];
    for($i = 0; $i < count($ingredients); $i++) {
        if(!isset($ingredients[$i])) {
            $this->form_validation->set_message(__FUNCTION__, 'The
            Ingredient field is required.');
```

return false;

```
        }

        if(!isset($_POST['ingredient_quantity'][$i])) {
            $this->form_validation->set_message(__FUNCTION__, 'The
            Quantity field is required.');
```

return false;

```
        }

        $food_group = $_POST['food_group'][$i];
        $counts = 0;

        for($j = 0; $j < count($ingredients); $j++) {
            if($_POST['food_group'][$i] == $food_group)
                $counts += $_POST['ingredient_quantity'][$j];
        }

        if($health_condition == 'Diabetes') {
            if(($food_group == "Vegetable" || $food_group == "Fruit") &&
                $counts > 4) {
                $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 4 servings of '.
                $food_group.' daily.');
```

return false;

```
            } else if($food_group == "Milk" && $counts > 3) {
                $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 3 servings of Milk
                daily.');
```

return false;

```
            } else if($food_group == "Rice" && $counts > 9) {
                $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 9 servings of Rice
                daily.');
```

return false;

```
            } else if($food_group == "Meat" && $counts > 6) {
                $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 6 servings of Meat
                daily.');
```

return false;

```
            } else if($food_group == "Meat" && $counts > 5) {
                $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 5 servings of Fat
                daily.');
```

return false;

```
            }
        } else if($health_condition == 'Hypertension' || $health_condition ==
        'Kidney Stones') {
            if(($food_group == "Vegetable" || $food_group == "Fruit") &&
                $counts > 5) {
                $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 5 servings of '.
                $food_group.' daily.');
```

return false;

```
            } else if($food_group == "Milk" && $counts > 5) {
                $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 5 servings of Milk
                daily.');
```

return false;

```
            } else if($food_group == "Rice" && $counts > 8) {
                $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 8 servings of Rice
                daily.');
```

return false;

```
            } else if($food_group == "Meat" && $counts > 2) {
                $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 2 servings of Meat
                daily.');
```

return false;

```
            } else if($food_group == "Meat" && $counts > 3) {
                $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 3 servings of Fat
                daily.');
```

return false;

```
            }
        }

        }

    }

    return true;
}

}

//AUTOCOMPLETE ADD INGREDIENTS
function suggestions() {
    $this->load->model('home_cmanager_model');
    $term = $this->input->post('term', TRUE);

    if (strlen($term) < 1) break;
}

```

```

$rows = $this->home_cmanager_model->GetAutocomplete(array('keyword' => $term))
;

$json_array = array();
foreach ($rows as $row){
//array_push($json_array, $row->food_name, $row->measure);
$json_array []['food_name'] = $row->food_name;
$json_array []['measure'] = $row->weight_edible_portion;
$json_array []['energy'] = $row->energy;
$json_array []['carbohydrate'] = $row->carbohydrate;
$json_array []['protein'] = $row->protein;
$json_array []['fat'] = $row->fat;
$json_array []['food_group'] = $row->food_group;
}
echo json_encode($json_array);
}

function get_fa() {
$this->load->model('home_cmanager_model');
$ingredient = $this->input->post('ingredient');
if (strlen($ingredient) < 1) break;

$rows = $this->home_cmanager_model->GetAutocomplete(array('keyword' =>
$ingredient));

$response = "";
foreach ($rows as $row)
$response = $row->energy." ,". $row->carbohydrate." ,". $row->protein." ,". $row->
fat;

echo $response;
}

//PLAN DAILY MEALS
function recommend_daily_meals () {
if($this->session->userdata('logged_in')) {
    $session_data = $this->session->userdata('logged_in');
    $data['username'] = $session_data['username'];
    $data['first_name'] = $session_data['first_name'];
    $data['last_name'] = $session_data['last_name'];
    $data['user_type'] = $session_data['user_type'];
    $data['email_address'] = $session_data['email_address'];
    $data['gender'] = $session_data['gender'];
    $data['birthday'] = $session_data['birthday'];
    $this->load->view('recommend_daily_meals_index', $data);
}
else {
    //If no session, redirect to login page
    redirect('login', 'refresh');
}
}

//VIEW SELECTED RECIPE UNDER RECOMMEND MEALS
function view_created_recommended_recipe($recommended_recipe_id) {
$this->load->model('home_cmanager_model');
$data['query'] = $this->home_cmanager_model->get_recipe_contents(
    $recommended_recipe_id);
if($this->session->userdata('logged_in')) {
    $session_data = $this->session->userdata('logged_in');
    $data['username'] = $session_data['username'];
    $data['first_name'] = $session_data['first_name'];
    $data['last_name'] = $session_data['last_name'];
    $data['user_type'] = $session_data['user_type'];
    $data['email_address'] = $session_data['email_address'];
    $data['gender'] = $session_data['gender'];
    $data['birthday'] = $session_data['birthday'];
    $this->load->view('view_recipe_recommended_index', $data);
}
else {
    //If no session, redirect to login page
    redirect('login', 'refresh');
}
}

//VIEW DAILY MEALS BY DATE
public function select_by_date() {
$record_date = $this->input->post('record_date');
if ($record_date != "") {
    $result = $this->home_cmanager_model->show_data_by_date($record_date);
    if ($result != false) {
        $data['result_display'] = $result;
    } else {
        $data['result_display'] = "No record found !";
    }
} else {
    $data['date_error_message'] = "Date field is required";
}
//$data['show_table'] = $this->view_table();
$this->load->view('recommend_daily_meals_index', $data);
//var_dump($result);
}

//SEARCH MEAL
function search_meal() {
// $this->load->library('form_validation');
// field name, error message, validation rules
// $this->form_validation->set_rules('record_date', 'Date', 'required');
}

```



```

//if($this->form_validation->run() == FALSE) {
    // $this->index();
    // $this->load->view('plan_daily_meals_index');
//}
//else {
    //if($this->input->post('submit')) {
    //    $data['record_date'] = $this->input->post('record_date');
    //}
    $this->load->model('home_cmanager_model');
    $data['query_meal'] = $this->home_cmanager_model->get_search_meal();
    $this->load->view('search_meal_cmanager_index', $data);
//}
if($this->session->userdata('logged_in')) {
    $session_data = $this->session->userdata('logged_in');
    $data['username'] = $session_data['username'];
    $data['first_name'] = $session_data['first_name'];
    $data['last_name'] = $session_data['last_name'];
    $data['user_type'] = $session_data['user_type'];
    $data['email_address'] = $session_data['email_address'];
    $data['gender'] = $session_data['gender'];
    $data['birthday'] = $session_data['birthday'];
}
else {
    //If no session, redirect to login page
    redirect('login', 'refresh');
}
}

//ADD MEAL
function add_meal($recommended_recipe_id) {
    $data['recommended_recipe_id'] = $recommended_recipe_id;

    $this->load->model('home_cmanager_model');
    if($this->input->post('submit')) {
        $this->home_cmanager_model->process_addmeal(
            $recommended_recipe_id);
        // $message = "Adding of Meal Successful!";
        //if ((isset($message)) && ($message != '')) {
        //echo '<script>
        //location.replace("../search_meal");
        //</script>';
        //}
        $this->session->set_flashdata('success_msg', '
            Adding of Meal Successful. ');
        redirect('home_cmanager/search_meal', 'refresh');
    }
}

if($this->session->userdata('logged_in')) {
    $session_data = $this->session->userdata('logged_in');
    $data['username'] = $session_data['username'];
    $data['first_name'] = $session_data['first_name'];
    $data['middle_name'] = $session_data['middle_name'];
    $data['last_name'] = $session_data['last_name'];
    $data['user_type'] = $session_data['user_type'];
    $data['email_address'] = $session_data['email_address'];
    $data['gender'] = $session_data['gender'];
    $data['birthday'] = $session_data['birthday'];
}
else {
    //If no session, redirect to login page
    redirect('login', 'refresh');
}

$this->load->view('add_meal_cmanager_index', $data);
}

//DELETE MEAL
function delete_meal($recommended_daily_meal_id) {
    $this->home_cmanager_model->del_meal($recommended_daily_meal_id);
    // $message = "Meal was successfully deleted!";
    //if ((isset($message)) && ($message != '')) {
    //echo '<script>
    //location.replace("../select_by_date");
    //</script>';
    //}
    $this->session->set_flashdata('success_msg', 'Meal was successfully
        deleted. ');
        redirect('home_cmanager/select_by_date', 'refresh');
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//VIEW SELECTED RECIPE FOR MEAL
function view_created_recipe_meal($recommended_recipe_id) {
    $this->load->model('home_cmanager_model');
    $data['query'] = $this->home_cmanager_model->get_recipe_contents(
        $recommended_recipe_id);
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
}

```

```

        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('view_recipe_add_meal_cmanager_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//EDIT FEL CONTROLLER
function edit_fel($food_exchange_list_id) {
    $item = $this->home_cmanager_model->get_fel_edit($food_exchange_list_id);
    $this->load->library('form_validation');
    // field name, error message, validation rules
    $this->form_validation->set_rules('list_group', 'List Group');
    $this->form_validation->set_rules('food_group', 'Food Group');
    $this->form_validation->set_rules('food_group_category', 'Food Group Category');
    $this->form_validation->set_rules('food_name', 'Food Name', 'required');
    $this->form_validation->set_rules('measure', 'Measure', 'required');
    $this->form_validation->set_rules('carbohydrate', 'Carbohydrate', 'required|numeric');
    $this->form_validation->set_rules('protein', 'Protein', 'required|numeric');
    $this->form_validation->set_rules('fat', 'Fat', 'required|numeric');
    $this->form_validation->set_rules('energy', 'Energy', 'required|numeric');
    $this->form_validation->set_rules('weight_as_purchased', 'Weight as Purchased', 'required|numeric');
    $this->form_validation->set_rules('weight_edible_portion', 'Weight Edible Portion', 'required|numeric');

    if (isset($_POST) && !empty($_POST)) {
        $data = array(
            // 'list_group' => $this->input->post('list_group'),
            // 'food_group' => $this->input->post('food_group'),
            // 'food_group_category' => $this->input->post('food_group_category'),
            'food_name' => $this->input->post('food_name'),
            'measure' => $this->input->post('measure'),
            'carbohydrate' => $this->input->post('carbohydrate'),
            'protein' => $this->input->post('protein'),
            'fat' => $this->input->post('fat'),
            'energy' => $this->input->post('energy'),
            'weight_as_purchased' => $this->input->post('weight_as_purchased'),
            'weight_edible_portion' => $this->input->post('weight_edible_portion')
        );

        if (isset($_POST['list_groupother'], $_POST['food_groupother'], $_POST['food_group_categoryother'])) {
            $data['list_group'] = $this->input->post('list_groupother');
            $data['food_group'] = $this->input->post('food_groupother');
            $data['food_group_category'] = $this->input->post('food_group_categoryother');
        } else {
            $data['list_group'] = $this->input->post('list_group');
            $data['food_group'] = $this->input->post('food_group');
            $data['food_group_category'] = $this->input->post('food_group_category');
        }

        if ($this->form_validation->run() == FALSE) {
            // $this->index();
        } else {
            $this->load->model('home_cmanager_model');
            if ($this->input->post('submit')) {
                $this->home_cmanager_model->update_fel($food_exchange_list_id, $data);
                // $message = "Update Food Exchange List Item Successful!";
                // if ((isset($message) && ($message != ''))) {
                //     echo '<script>
                //         //location.replace("../home_cmanager")
                //     </script>';
                // }
                $this->session->set_flashdata('success_msg', 'Update Food Exchange List Item Successful');
                redirect('home_cmanager', 'refresh');
            }
        }
    }
    $data['item'] = $item;
    $data['list_group'] = $this->form_validation->set_value('list_group', $item['list_group']);
    $data['food_group'] = $this->form_validation->set_value('food_group', $item['food_group']);
    $data['food_group_category'] = $this->form_validation->set_value('food_group_category', $item['food_group_category']);
    $data['food_name'] = $this->form_validation->set_value('food_name', $item['food_name']);
    $data['measure'] = $this->form_validation->set_value('measure', $item['measure']);
}

```

```

        $data['carbohydrate'] = $this->form_validation->set_value('carbohydrate',
            $item['carbohydrate']);
        $data['protein'] = $this->form_validation->set_value('protein', $item['
            protein']);
        $data['fat'] = $this->form_validation->set_value('fat', $item['fat']);
        $data['energy'] = $this->form_validation->set_value('energy', $item['energy
            ']);
        $data['weight_as_purchased'] = $this->form_validation->set_value('
            weight_as_purchased', $item['weight_as_purchased']);
        $data['weight_edible_portion'] = $this->form_validation->set_value('
            weight_edible_portion', $item['weight_edible_portion']);
        $data['lg'] = $this->home_cmanager_model->get_list_group();

        if($this->session->userdata('logged_in')) {
            $session_data = $this->session->userdata('logged_in');
            $data['username'] = $session_data['username'];
        }
        else {
            //If no session, redirect to login page
            redirect('login', 'refresh');
        }

        $this->load->view('edit_fel_index', $data);
    }

    //EDIT HT CONTROLLER
    function edit_ht($health_tip_id) {
        $item = $this->home_cmanager_model->get_ht_edit($health_tip_id);
        $this->load->library('form_validation');
        // field name, error message, validation rules
        $this->form_validation->set_rules('health_tip', 'Health Tip', 'required');

        if (isset($_POST) && !empty($_POST)) {
            $data = array(
                'health_tip' => $this->input->post('health_tip')
            );
        }

        if($this->form_validation->run() == FALSE) {
            //$this->index();
        }
        else {
            $this->load->model('home_cmanager_model');
            if($this->input->post('submit')) {
                $this->home_cmanager_model->update_ht($health_tip_id,
                    $data);
                //$message = "Edit Health Tip Successful!";
                //if ((isset($message) && ($message != ''))) {
                    //echo '<script>
                        //location.replace("../home_cmanager")
                    ;
                    //</script >';
                //}
                $this->session->set_flashdata('success_msg', '
                    Edit Health Tip Successful. ');
                redirect('home_cmanager', 'refresh');
            }
        }
        $data['item'] = $item;
        $data['health_tip'] = $this->form_validation->set_value('health_tip', $item['
            health_tip']);

        if($this->session->userdata('logged_in')) {
            $session_data = $this->session->userdata('logged_in');
            $data['username'] = $session_data['username'];
        }
        else {
            //If no session, redirect to login page
            redirect('login', 'refresh');
        }

        $this->load->view('edit_ht_index', $data);
    }
}
?>

```

---

```

<?php if (! defined('BASEPATH')) exit('No direct script access allowed');
session_start(); //we need to call PHP's session object to access it through CI

```

```

class Home_User extends CI_Controller {

    function __construct() {
        parent::__construct();
        $this->load->helper(array('form', 'url'));
        $this->load->model('home_user_model');
    }

    function index() {
        if($this->session->userdata('logged_in')) {
            $session_data = $this->session->userdata('logged_in');
            $data['user_id'] = $session_data['user_id'];
            $data['username'] = $session_data['username'];
            $data['first_name'] = $session_data['first_name'];
            $data['last_name'] = $session_data['last_name'];
            $data['user_type'] = $session_data['user_type'];
        }
    }
}

```

```

        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('home_user_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

function logout() {
    $this->session->unset_userdata('logged_in');
    session_destroy();
    redirect('login', 'refresh');
}

public function check_user() {
    $usr=$this->input->post('name');
    $result=$this->home_user_model->check_user_exist($usr);
    if($result) {
        echo "false";
    }
    else {
        echo "true";
    }
}

public function check_email() {
    $seml=$this->input->post('email');
    $result=$this->home_user_model->check_email_exist($seml);
    if($result) {
        echo "false";
    }
    else {
        echo "true";
    }
}

//EDIT PASSWORD CONTROLLER
function edit_password_user() {
    $this->load->library('form_validation');
    // field name, error message, validation rules
    $this->form_validation->set_rules('old_password', 'Old Password', 'trim|
    required|min_length[8]|callback_check_old_password');
    $this->form_validation->set_rules('password', 'Password', 'trim|required|
    min_length[8]');
    $this->form_validation->set_rules('confirm_password', 'Confirm Password', '
    trim|required|matches[password]');

    if (isset($_POST) && !empty($_POST)) {
        $data = array(
            'password' => MD5($this->input->post('password')),
            'confirm_password' => MD5($this->input->post('confirm_password
            '));
    );

    $username = $this->input->post('username');

    if($this->form_validation->run() == FALSE) {
        // $this->index();
    }
    else {
        $this->load->model('home_user_model');
        if($this->input->post('submit')) {
            $this->home_user_model->update_password($username,
                $data);
            // $message = "Password Change Successful!";
            // if ((isset($message) && ($message != ''))) {
            // echo '<script>
            // location.replace("../
            // home-user");
            // }
            // }
            $this->session->set_flashdata('success-msg', 'Password Change
            Successful. ');
            redirect('home-user', 'refresh');
        }
    }
}

if($this->session->userdata('logged_in')) {
    $session_data = $this->session->userdata('logged_in');
    $data['username'] = $session_data['username'];
}
else {
    //If no session, redirect to login page
    redirect('login', 'refresh');
}

$this->load->view('edit_password_user_index', $data);
}

public function check_old_password() {
    $opwd=MD5($this->input->post('old_password'));
    $result=$this->home_user_model->check_old_password_exist($opwd);
    if($result) {
        return true;
    }
}

```

```

        else {
            $this->form_validation->set_message(__FUNCTION__, 'Incorrect Old
                Password. ');
            return false;
        }
    }

    //EDIT PROFILE CONTROLLER
    function edit_profile_user() {
        $this->load->library('form_validation');
        // field name, error message, validation rules
        $this->form_validation->set_rules('username', 'Username', 'trim|required|
            min_length[4]|xss_clean');
        $this->form_validation->set_rules('first_name', 'First Name', 'trim|required|
            xss_clean');
        $this->form_validation->set_rules('middle_name', 'Middle Name', 'xss_clean');
        $this->form_validation->set_rules('last_name', 'Last Name', 'trim|required|
            xss_clean');
        $this->form_validation->set_rules('birthdate', 'Birthdate', 'required|
            xss_clean');
        $this->form_validation->set_rules('genderRadio', 'Gender', 'required');
        $this->form_validation->set_rules('userTypeRadio', 'User Type', 'required');
        $this->form_validation->set_rules('email_address', 'Email Address', 'required|
            valid_email');

        if (isset($_POST) && !empty($_POST)) {
            $data = array(
                'user_id' => $this->input->post('user_id'),
                'username' => $this->input->post('username'),
                'first_name' => $this->input->post('first_name'),
                'middle_name' => $this->input->post('middle_name'),
                'last_name' => $this->input->post('last_name'),
                'gender' => $this->input->post('genderRadio'),
                'user_type' => $this->input->post('userTypeRadio'),
                'email_address' => $this->input->post('email_address'),
                'birthday' => $this->input->post('birthdate')
            );

            $username = $this->input->post('username');
            $user_id = $this->input->post('userid');

            if($this->form_validation->run() == FALSE) {
                // $this->index();
            }
            else {
                $this->session->set_userdata('logged_in', $data);
                $this->load->model('home_user_model');
                if($this->input->post('submit')) {
                    $this->home_user_model->update_profile($username,
                        $data);
                    // $message = "Profile Update Successful!";
                    // if ((isset($message) && ($message != ''))) {
                    //     echo '<script>
                    //         //location.replace("../
                    //             home_user");
                    //     </script >';
                    // }
                    $this->session->set_flashdata('success_msg', 'Profile Update
                        Successful. ');
                    redirect('home_user', 'refresh');
                }
            }
        }
        if($this->session->userdata('logged_in')) {
            $session_data = $this->session->userdata('logged_in');
            $data['username'] = $session_data['username'];
            $data['first_name'] = $session_data['first_name'];
            $data['middle_name'] = $session_data['middle_name'];
            $data['last_name'] = $session_data['last_name'];
            $data['user_type'] = $session_data['user_type'];
            $data['email_address'] = $session_data['email_address'];
            $data['gender'] = $session_data['gender'];
            $data['birthday'] = $session_data['birthday'];
        }
        else {
            // If no session, redirect to login page
            redirect('login', 'refresh');
        }

        $this->load->view('edit_profile_user_index', $data);
    }

    //SEARCH FOOD EXCHANGE LIST
    function search_fel() {
        $this->load->model('home_user_model');
        $data['query_fel'] = $this->home_user_model->get_search_fel();
        if($this->session->userdata('logged_in')) {
            $session_data = $this->session->userdata('logged_in');
            $data['username'] = $session_data['username'];
            $data['first_name'] = $session_data['first_name'];
            $data['last_name'] = $session_data['last_name'];
            $data['user_type'] = $session_data['user_type'];
            $data['email_address'] = $session_data['email_address'];
            $data['gender'] = $session_data['gender'];
            $data['birthday'] = $session_data['birthday'];
        }
        $this->load->view('searchfel_user_index', $data);
    }
    else {
        // If no session, redirect to login page
    }
}

```

```

        redirect('login', 'refresh');
    }
}

//HEALTH TIPS NEWS FEED
function user_ht_feed() {
    $this->load->model('home_user_model');
    /*$count = $this->home_user_model->count_ht();

    $this->load->library('pagination');
    $config['total-rows'] = $count;

    $config['per-page'] = 2;
    $this->pagination->initialize($config);
    $data['pagination'] = $this->pagination->create_links();

    $data['ht_feed'] = $this->home_user_model->get_search_ht_ret(array(
    'limit' => $config['per-page'],
    'offset' => $this->pagination->offset));*/

    $data['ht_feed'] = $this->home_user_model->get_search_ht();
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('user_ht_feed_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//ACHIEVEMENTS NEWS FEED
//HEALTH TIPS NEWS FEED
function user_achievements_feed() {
    $this->load->model('home_user_model');
    $data['achievement_feed'] = $this->home_user_model->get_search_achievements();
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('user_achievement_feed_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//FITNESS PROFILE
function show_fitness_profile() {
    $data['query'] = $this->home_user_model->process_viewfp();
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('view_fitness_profile_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

function weightmonitor_getdata() {
    $weightchart_data = $this->home_user_model->get_weightchart_data();

    $category = array();
    $category['name'] = 'Category';

    $series1 = array();
    $series1['name'] = 'Current Weight';

    $series2 = array();
    $series2['name'] = 'Goal Weight';

    foreach ($weightchart_data as $row) {
        $category['data'][] = $row->record_date;
        $series1['data'][] = $row->current_weight;
        $series2['data'][] = $row->goal_weight;
    }

    $result = array();

```

```

        array_push($result , $category);
        array_push($result , $series1);
        array_push($result , $series2);
    }
    echo json_encode($result , JSON_NUMERIC_CHECK);
}

public function check_fitness_profile() {
    $rec_date=$this->input->post('record_date');
    $result=$this->home_user_model->check_fp_exist($rec_date);
    if($result) {
        $this->form_validation->set_message(__FUNCTION__, 'You already entered
            data for the selected date. ');
        return false;
    }
    else {
        return true;
    }
}

function add_fitness_profile () {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('add_fitness_profile_index', $data);
    }
    else {
        //If no session , redirect to login page
        redirect('login', 'refresh');
    }
}

function add_fitness_profile_save () {
    $this->load->library('form_validation');
    // field name, error message, validation rules
    $this->form_validation->set_rules('goal.weight', 'Goal Weight', 'required|
        numeric');
    $this->form_validation->set_rules('current.weight', 'Current Weight', '
        required|numeric');
    $this->form_validation->set_rules('height', 'Height', 'required|numeric');
    $this->form_validation->set_rules('record_date', 'Record Date', 'required|
        callback_check_fitness_profile');
    $this->form_validation->set_rules('health.condition', 'Health Condition', '
        required');

    if($this->form_validation->run() == FALSE) {
        //$this->index();
        $this->load->view('add_fitness_profile_index');
    }
    else {
        $this->load->model('home_user_model');
        if($this->input->post('submit')) {
            $this->home_user_model->process_addfp();
            //$message = "Adding of Information Successful!";
            //if ((isset($message) && ($message != ''))) {
            //echo '<script>
                //location.replace("../home-user/show_fitness_profile
                    ");
            //</script>';
            //}

            $this->session->set_flashdata('success-msg', 'Information
                successfully added. ');
            redirect('home-user/show_fitness_profile', 'refresh');
        }
    }
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session , redirect to login page
        redirect('login', 'refresh');
    }
}

//DELETE FITNESS PROFILE CONTROLLER
function delete_fitness_profile($fitness_info_id) {
    $this->home_user_model->del_fitness_profile($fitness_info_id);
    //$message = "Fitness Profile was successfully deleted!";
    //if ((isset($message) && ($message != ''))) {
    //echo '<script>
        //location.replace("../show_fitness_profile");
    //</script>';
    //}
    $this->session->set_flashdata('success-msg', 'Fitness Profile was
        successfully deleted. ');
        redirect('home-user/show_fitness_profile', '
            refresh');
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
    }
}

```

```

        $data['username'] = $session_data['username'];
    }
    else {
        //If no session , redirect to login page
        redirect('login', 'refresh');
    }
}

//SHARE ACHIEVEMENT CONTROLLERS
function share_achievement() {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session , redirect to login page
        redirect('login', 'refresh');
    }
    $this->load->view('share_achievement_index');
}

function share_achievement_save() {
    $this->load->library('form_validation');
    // field name, error message, validation rules
    $this->form_validation->set_rules('achievement', 'Achievement', 'required');

    if($this->form_validation->run() == FALSE) {
        // $this->index();
        $this->load->view('share_achievement_index');
    }
    else {
        $this->load->model('home_user_model');
        if($this->input->post('submit')) {
            $this->home_user_model->process_achievement();
            // $message = "Sharing of Achievement Successful!";
            // if ((isset($message) && ($message != ''))) {
            // echo '<script>
            // location.replace("../home_user");
            // </script>';
            // }
            $this->session->set_flashdata('success_msg', 'Sharing of
            Achievement Successful. ');
            redirect('home_user', 'refresh');
        }
    }
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session , redirect to login page
        redirect('login', 'refresh');
    }
}

//VIEW CREATED RECIPES
function show_created_recipes () {
    $data['query'] = $this->home_user_model->process_view_created_recipes();
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('view_created_recipes_index', $data);
    }
    else {
        //If no session , redirect to login page
        redirect('login', 'refresh');
    }
}

//EDIT CREATED RECIPES
function edit_created_recipe($own_recipe_id) {
    $item = $this->home_user_model->get_created_recipe($own_recipe_id);
    $this->load->library('form_validation');
    // field name, error message, validation rules
    // $this->form_validation->set_rules('own_recipe_id', 'Username', 'trim|
    // required|min_length[4]|xss_clean');
    // $this->form_validation->set_rules('user_id', 'First Name', 'trim|required|
    // xss_clean');
    $this->form_validation->set_rules('health_condition', 'Health Condition', '
    required');
    $this->form_validation->set_rules('meal_type', 'Meal Type', 'required');
    $this->form_validation->set_rules('recipe_name', 'Recipe Name', 'required');
    $this->form_validation->set_rules('serving_sugg', 'Serving Size', 'required|
    numeric');
    $this->form_validation->set_rules('ingredients', 'Ingredients', '
    callback_health_conditions_restrictions');
    $this->form_validation->set_rules('preparation_procedure', 'Procedure', '
    required');
    $this->form_validation->set_rules('calories', 'Calories', 'required|numeric');
    $this->form_validation->set_rules('carbohydrate', 'Carbohydrates', 'required|
    numeric');
    $this->form_validation->set_rules('protein', 'Protein', 'required|numeric');
    $this->form_validation->set_rules('fat', 'Fat', 'required|numeric');
}

```



```

if (isset($_POST) && !empty($_POST)) {
    $data = array(
        // 'own_recipe_id' => $this->input->post('own_recipe_id'),
        // 'user_id' => $this->input->post('user_id'),
        // 'health_condition' => $this->input->post('health_condition')
        'health_condition' => implode(",",$this->input->post('
            health_condition')),
        'meal_type' => $this->input->post('meal_type'),
        'recipe_name' => $this->input->post('recipe_name'),
        'serving_sugg' => $this->input->post('serving_sugg'),
        'preparation_procedure' => $this->input->post('
            preparation_procedure'),
        'calories' => $this->input->post('calories'),
        'carbohydrate' => $this->input->post('carbohydrate'),
        'protein' => $this->input->post('protein'),
        'fat' => $this->input->post('fat')
    );

    $data2 = array();

    $data2['food_name'] = $this->input->post('ingredients');
    $data2['food_group'] = $this->input->post('food_group');
    $data2['measure'] = $this->input->post('ingredient_measure');
    $data2['serving_size'] = $this->input->post('ingredient_quantity');
    $data2['ingredient_id'] = $this->input->post('ingredient_id');

    if($this->form-validation->run() == FALSE) {
        // $this->index();
    }
    else {
        $this->load->model('home_user_model');
        if($this->input->post('submit')) {
            $this->home_user_model->update_recipe($own_recipe_id ,
                $data,$data2);
            // $message = "Edit Recipe Successful!";
            // if ((isset($message) && ($message != '')) {
            //     // echo '<script>
            //         // location.replace("../home-user");
            //     // </script>';
            // }
            $this->session->set_flashdata('success_msg', 'Edit Recipe
                Successful. ');
            redirect('home-user', 'refresh');
        }
    }
}
$data['item'] = $item;
// $data['own_recipe_id'] = $this->form-validation->set_value('own_recipe_id',
//     $item['own_recipe_id']);
// $data['user_id'] = $this->form-validation->set_value('user_id', $item['
//     user_id']);
// $data['health_condition'] = $this->form-validation->set_value('
//     health_condition', $item['health_condition']);
// $data['meal_type'] = $this->form-validation->set_value('meal_type', $item['
//     meal_type']);
// $data['recipe_name'] = $this->form-validation->set_value('recipe_name',
//     $item['recipe_name']);
// $data['serving_sugg'] = $this->form-validation->set_value('serving_sugg',
//     $item['serving_sugg']);

// $data['food_name'] = $this->form-validation->set_value('ingredients', $item
//     ['food_name']);
// $data['food_group'] = $this->form-validation->set_value('food_group', $item
//     ['food_group']);
// $data['measure'] = $this->form-validation->set_value('ingredient_measure',
//     $item['measure']);
// $data['serving_size'] = $this->form-validation->set_value('
//     ingredient_quantity', $item['serving_size']);

// $data['preparation_procedure'] = $this->form-validation->set_value('
//     preparation_procedure', $item['preparation_procedure']);
// $data['calories'] = $this->form-validation->set_value('calories', $item['
//     calories']);
// $data['carbohydrate'] = $this->form-validation->set_value('carbohydrate',
//     $item['carbohydrate']);
// $data['protein'] = $this->form-validation->set_value('protein', $item['
//     protein']);
// $data['fat'] = $this->form-validation->set_value('fat', $item['fat']);

// print_r($item);
// die();

if($this->session->userdata('logged_in')) {
    $session_data = $this->session->userdata('logged_in');
    $data['username'] = $session_data['username'];
}
else {
    // If no session, redirect to login page
    redirect('login', 'refresh');
}
$this->load->view('edit_recipe_index', $data);
}

```

```

//SHOW RECOMMENDED DAILY MEALS
function show_recommended_daily_meals () {
    if ($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('show_recommended_daily_meals_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//SELECT BY DATE RECOMMENDED MEALS
public function recommended_select_by_date() {
    $record_date = $this->input->post('record_date');
    if ($record_date != "") {
        $result = $this->home_user_model->recommended_show_data_by_date(
            $record_date);
        if ($result != false) {
            $data['result_display'] = $result;
        } else {
            $data['result_display'] = "No record found!";
        }
    } else {
        $data['date_error_message'] = "Date field is required";
    }
    // $data['show_table'] = $this->view_table();
    $this->load->view('show_recommended_daily_meals_index', $data);
    //var_dump($result);
}

//VIEW SELECTED RECIPE UNDER RECOMMENDED MEALS
function view_created_recommended_recipe($recommended_recipe_id) {
    $this->load->model('home_user_model');
    $data['query'] = $this->home_user_model->get_recommended_recipe_contents(
        $recommended_recipe_id);
    if ($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('view_daily_recommended_recipe_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//VIEW RECOMMENDED RECIPES
function show_recommended_recipes () {
    $data['query'] = $this->home_user_model->process_view_recommended_recipes();
    if ($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('view_recommended_recipes_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//SEARCH RECOMMENDED RECIPES
function search_recrec () {
    $this->load->model('home_user_model');
    $data['query'] = $this->home_user_model->get_search_recrec();
    if ($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('view_recommended_recipes_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

```

```

}

//VIEW SHARED RECIPES
function show_shared_recipes () {
    $data['query'] = $this->home_user_model->process_view_shared_recipes();
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('view_shared_recipes_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//VIEW SELECTED SHARED RECIPE
function view_shared_recipe($shared_recipe_id) {
    $this->load->model('home_user_model');
    $data['query'] = $this->home_user_model->get_shared_recipe_contents(
        $shared_recipe_id);
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('view_shared_recipe_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//SEARCH SHARED RECIPES
function search_sharedrec () {
    $this->load->model('home_user_model');
    $data['query'] = $this->home_user_model->get_search_sharedrec();
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('view_shared_recipes_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//VIEW SELECTED RECIPE
function view_created_recipe($own_recipe_id) {
    $this->load->model('home_user_model');
    $data['query'] = $this->home_user_model->get_recipe_contents($own_recipe_id);
    //var_dump($data['query']);
    //die();
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('view_recipe_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//VIEW SELECTED RECIPE
function view_recommended_recipe($recommended_recipe_id) {
    $this->load->model('home_user_model');
    $data['query'] = $this->home_user_model->get_recommended_recipe_contents(
        $recommended_recipe_id);
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
    }
}

```

```

        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('view_recommended_recipe_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//DELETE CREATED RECIPE
function delete_created_recipe($own_recipe_id) {
    $this->home_user_model->del_created_recipe($own_recipe_id);
    //message = "Recipe was successfully deleted!";
    //if ((isset($message) && ($message != '')) {
        //echo '<script>
        //location.replace("../show_created_recipes");
        //</script>';
    //}
    $this->session->set_flashdata('success_msg', 'Recipe was successfully
    deleted. ');
    redirect('home_user/show_created_recipes', 'refresh');
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//DELETE INGREDIENTS
function delete_ingredient() {
    $this->home_user_model->del_ingredient($this->input->post('ingredient_id'));
}

//SHARE RECIPE
function share_created_recipe($own_recipe_id) {

    // $own_recipe_id = $this->input->post('own_recipe_id');
    // $data['own_recipe_id'] = $own_recipe_id;
    // echo $own_recipe_id;

    $this->load->model('home_user_model');
    //if($this->input->post('submit')) {
        $this->home_user_model->process_sharerecipe(
            $own_recipe_id);
        //message = "Sharing of Recipe Successful!";
    //if ((isset($message) && ($message != '')) {
        //echo '<script>
        //location.replace("../show_created_recipes");
        //</script>';
    //}
    //}
    $this->session->set_flashdata('success_msg', 'Sharing of Recipe
    Successful. ');
    redirect('home_user/show_created_recipes', 'refresh');

    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['middle_name'] = $session_data['middle_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//CREATE RECIPE CONTROLLER
function add_recipe () {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('add_recipe_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

function add_recipe_save () {
    $this->load->library('form_validation');
}

```

```

// field name, error message, validation rules
$this->form_validation->set_rules('health_condition', 'Health Condition', '
required');
$this->form_validation->set_rules('meal_type', 'Meal Type', 'required');
$this->form_validation->set_rules('recipe_name', 'Recipe Name', 'required');
$this->form_validation->set_rules('serving_sugg', 'Serving Size', 'required|
numeric');
$this->form_validation->set_rules('ingredients', 'Ingredients', '
callback_health_conditions_restrictions');
$this->form_validation->set_rules('preparation_procedure', 'Procedure', '
required');
$this->form_validation->set_rules('calories', 'Calories', 'required|numeric');
$this->form_validation->set_rules('carbohydrate', 'Carbohydrates', 'required|
numeric');
$this->form_validation->set_rules('protein', 'Protein', 'required|numeric');
$this->form_validation->set_rules('fat', 'Fat', 'required|numeric');

if($this->form_validation->run() == FALSE) {
    //echo '<pre>'.print_r($this->input->post(),TRUE).'\</pre>';
    //die();
    //$this->index();
    //if($this->input->post())
}

// Merge the submitted data with the current data
// $data = $this->input->post();
//echo '<pre>'.print_r($data).'\</pre>';
//die();
}

// $data = $this->input->post();
// $ingredients = $this->input->post('ingredients');
//echo '<pre>'.print_r($ingredients).'\</pre>';
//die();
//echo '<pre>'.print_r($data).'\</pre>';
//die();

$this->load->view('add_recipe_index'/*, $data*/);
} else {
    $this->load->model('home_user_model');
    if($this->input->post('submit')) {

        /*Create config for image library*/
        $this->load->library('image_lib');
        $config['image_library'] = 'gd2';

        /* Create config for upload library*/
        $config['upload_path'] = './uploads/';
        $config['allowed_types'] = 'gif|jpg|png';
        $config['max_size'] = '0';
        $config['max_width'] = '0';
        $config['max_height'] = '0';
        $this->load->library('upload', $config);

        if( ! $this->upload->do_upload('userfile')) {
            if (!empty($_FILES['userfile']['name'])) {
                // $error = array('error' => $this->
                upload->display_errors());
                $error = $this->upload->display_errors
                ();
                $data['error_display'] = $error;
                $this->load->view('add_recipe_index',
                $data);
            } else {

                $filename = 'default';
                $fileExtension = 'png';

                $data3['raw_name'] = $filename;
                $data3['file_ext'] = $fileExtension;

                $file=array(
                    'img_name'=>$data3['raw_name
                    '],
                    'thumb_name'=>$data3['raw_name
                    '].'_thumb',
                    'ext'=>$data3['file_ext'],
                    'upload_date'=>time()
                );

                $this->home_user_model->process_addrecipe(
                $file);
            }
        } else {
            $data3=$this->upload->data();
            $path_info = pathinfo($_FILES["userfile"]["
            name"]);
            $filename = $path_info['filename'];
            $fileExtension = $path_info['extension'];

            // $data3=$this->upload->data();
            // $this->thumb($data3);

            $data3['raw_name'] = $filename;
            $data3['file_ext'] = $fileExtension;

            $file=array(
                'img_name'=>$data3['raw_name
                '],

```

```

        'thumb_name'=>$data3['raw_name
        '].'thumb',
        'ext'=>$data3['file_ext'],
        'upload_date'=>time()
    );

    $this->home_user_model->process_addrecipe(
        $file);

    $data3 = array('upload_data' => $this->upload
        ->data());
    }

    //message = "Creating of Recipe Successful!";
    //if ((isset($message)) && ($message != '')) {
        //echo '<script>
            //location.replace("../
                home_user/
                show_created_recipes");
            //</script >';
        //}
        $this->session->set_flashdata('success_msg', '
            Creating of Recipe Successful. ');
        redirect('home_user/show_created_recipes', '
            refresh');
    }
}

if($this->session->userdata('logged_in')) {
    $session_data = $this->session->userdata('logged_in');
    $data['username'] = $session_data['username'];
}
else {
    //If no session, redirect to login page
    redirect('login', 'refresh');
}
}

/*function thumb($data3) {
    $config['image_library'] = 'gd2';
    $config['source_image'] = $data3['full_path'];
    $config['create_thumb'] = TRUE;
    $config['maintain_ratio'] = TRUE;
    $config['width'] = 275;
    $config['height'] = 250;
    $this->load->library('image-lib', $config);
    $this->image-lib->resize();
}*/

function health_conditions_restrictions($ingredients) {
    $health_condition = $_POST['health-condition'];
    for($i = 0; $i < count($ingredients); $i++) {
        if(!isset($ingredients[$i])) {
            $this->form_validation->set_message(__FUNCTION__, 'The
                Ingredient field is required. ');
            return false;
        }

        if(!isset($_POST['ingredient-quantity'][$i])) {
            $this->form_validation->set_message(__FUNCTION__, 'The
                Quantity field is required. ');
            return false;
        }
    }

    $food_group = $_POST['food-group'][$i];
    $counts = 0;

    for($j = 0; $j < count($ingredients); $j++) {
        if($_POST['food-group'][$i] == $food_group)
            $counts += $_POST['ingredient-quantity'][$j];
    }

    if($health_condition == 'Diabetes') {
        if(($food_group == "Vegetable" || $food_group == "Fruit") &&
            $counts > 4) {
            $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 4 servings of '.
                $food_group.'. daily. ');
            return false;
        }
        else if($food_group == "Milk" && $counts > 3) {
            $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 3 servings of Milk
                daily. ');
            return false;
        }
        else if($food_group == "Rice" && $counts > 9) {
            $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 9 servings of Rice
                daily. ');
            return false;
        }
        else if($food_group == "Meat" && $counts > 6) {
            $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 6 servings of Meat
                daily. ');
            return false;
        }
        else if($food_group == "Meat" && $counts > 5) {
            $this->form_validation->set_message(__FUNCTION__, 'You
                are only allowed a maximum of 5 servings of Fat
                daily. ');
            return false;
        }
    }
}

```

```

    } else if($health_condition == 'Hypertension' || $health_condition ==
    'Kidney Stones') {
        if(($food_group == "Vegetable" || $food_group == "Fruit") &&
            $counts > 5) {
            $this->form_validation->set_message(__FUNCTION__, 'You
            are only allowed a maximum of 5 servings of ' .
            $food_group . ' daily. ');
            return false;
        } else if($food_group == "Milk" && $counts > 5) {
            $this->form_validation->set_message(__FUNCTION__, 'You
            are only allowed a maximum of 5 servings of Milk
            daily. ');
            return false;
        } else if($food_group == "Rice" && $counts > 8) {
            $this->form_validation->set_message(__FUNCTION__, 'You
            are only allowed a maximum of 8 servings of Rice
            daily. ');
            return false;
        } else if($food_group == "Meat" && $counts > 2) {
            $this->form_validation->set_message(__FUNCTION__, 'You
            are only allowed a maximum of 2 servings of Meat
            daily. ');
            return false;
        } else if($food_group == "Meat" && $counts > 3) {
            $this->form_validation->set_message(__FUNCTION__, 'You
            are only allowed a maximum of 3 servings of Fat
            daily. ');
            return false;
        }
    }
}

return true;
}
}

//AUTOCOMPLETE ADD INGREDIENTS
//USER SEARCH CONTROLLERS
function suggestions() {
    $this->load->model('home_user_model');
    $term = $this->input->post('term', TRUE);

    if (strlen($term) < 1) break;

    $rows = $this->home_user_model->GetAutocomplete(array('keyword' => $term));

    $json_array = array();
    foreach ($rows as $row){
        //array_push($json_array, $row->food_name, $row->measure);
        $json_array[][ 'food_name' ] = $row->food_name;
        $json_array[][ 'measure' ] = $row->weight_edible_portion;
        $json_array[][ 'energy' ] = $row->energy;
        $json_array[][ 'carbohydrate' ] = $row->carbohydrate;
        $json_array[][ 'protein' ] = $row->protein;
        $json_array[][ 'fat' ] = $row->fat;
        $json_array[][ 'food_group' ] = $row->food_group;
    }
    echo json_encode($json_array);
}

function get_fa() {
    $this->load->model('home_user_model');
    $ingredient = $this->input->post('ingredient');
    if (strlen($ingredient) < 1) break;

    $rows = $this->home_user_model->GetAutocomplete(array('keyword' => $ingredient
    ));

    $response = "";
    foreach ($rows as $row)
        $response = $row->energy . " , " . $row->carbohydrate . " , " . $row->protein . " , " . $row->
        fat;

    echo $response;
}

//PLAN DAILY MEALS
function plan_daily_meals () {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('plan_daily_meals_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//VIEW SELECTED RECIPE UNDER PLAN MEALS
function view_created_plan_recipe($own_recipe_id) {
    $this->load->model('home_user_model');
    $data['query'] = $this->home_user_model->get_recipe_contents($own_recipe_id);
    if($this->session->userdata('logged_in')) {

```

```

        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('view_recipe_plan_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//VIEW DAILY MEALS BY DATE
public function select_by_date() {
    $record_date = $this->input->post('record_date');
    if ($record_date != "") {
        $result = $this->home_user_model->show_data_by_date($record_date);
        if ($result != false) {
            $data['result_display'] = $result;
        } else {
            $data['result_display'] = "No record found!";
        }
    } else {
        $data['date_error_message'] = "Date field is required";
    }
    // $data['show_table'] = $this->view_table();
    $this->load->view('plan_daily_meals_index', $data);
    //var_dump($result);
}

//SEARCH MEAL
function search_meal() {
    // $this->load->library('form_validation');
    // field name, error message, validation rules
    // $this->form_validation->set_rules('record_date', 'Date', 'required');

    // if ($this->form_validation->run() == FALSE) {
        // $this->index();
        // $this->load->view('plan_daily_meals_index');
    // }
    // else {
        // if ($this->input->post('submit')) {
            // $data['record_date'] = $this->input->post('record_date');
            // }
            $this->load->model('home_user_model');
            $data['query_meal'] = $this->home_user_model->get_search_meal();
            $this->load->view('search_meal_index', $data);
        // }
        if ($this->session->userdata('logged_in')) {
            $session_data = $this->session->userdata('logged_in');
            $data['username'] = $session_data['username'];
            $data['first_name'] = $session_data['first_name'];
            $data['last_name'] = $session_data['last_name'];
            $data['user_type'] = $session_data['user_type'];
            $data['email_address'] = $session_data['email_address'];
            $data['gender'] = $session_data['gender'];
            $data['birthday'] = $session_data['birthday'];
        }
        else {
            //If no session, redirect to login page
            redirect('login', 'refresh');
        }
    }
}

//ADD MEAL
function add_meal($own_recipe_id) {
    // $own_recipe_id = $this->input->post('own_recipe_id');
    $data['own_recipe_id'] = $own_recipe_id;
    // echo $own_recipe_id;

    $this->load->model('home_user_model');
    if ($this->input->post('submit')) {
        $this->home_user_model->process_addmeal($own_recipe_id);
        // $message = "Adding of Meal Successful!";
        // if ((isset($message) && ($message != ''))) {
            // echo '<script>
                // location.replace("../search_meal");
            // </script>';
            // }
            $this->session->set_flashdata('success_msg', '
                Adding of Meal Successful. ');
            redirect('home_user/search_meal', 'refresh');
        }
    }

    if ($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['middle_name'] = $session_data['middle_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
    }
}

```



```

        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }

    $this->load->view('add_meal_index', $data);
}

//DELETE MEAL
function delete_meal($plan_daily_meal_id) {
    $this->home_user_model->del_meal($plan_daily_meal_id);
    // $message = "Meal was successfully deleted!";
    // if ((isset($message) && ($message != ''))) {
    //     //echo '<script>
    //         //location.replace("../select_by_date");
    //     //</script>';
    //     $this->session->set_flashdata('success_msg', 'Meal was
    //         successfully deleted. ');
    //     redirect('home_user/select_by_date', 'refresh')
    //     ;
    // }
    // }
    if ($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//VIEW SELECTED RECIPE FOR MEAL
function view_created_recipe_meal($own_recipe_id) {
    $this->load->model('home_user_model');
    $data['query'] = $this->home_user_model->get_recipe_contents($own_recipe_id);
    if ($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
        $data['first_name'] = $session_data['first_name'];
        $data['last_name'] = $session_data['last_name'];
        $data['user_type'] = $session_data['user_type'];
        $data['email_address'] = $session_data['email_address'];
        $data['gender'] = $session_data['gender'];
        $data['birthday'] = $session_data['birthday'];
        $this->load->view('view_recipe_add_meal_index', $data);
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

//FACEBOOK
function share_achievement_fb() {
    if ($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $data['username'] = $session_data['username'];
    }
    else {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
    $this->load->view('share_achievement_fb_index');
}

function post_facebook() {
    parse_str($_SERVER['QUERY_STRING'], $_REQUEST);
    $CI = &get_instance();
    $CI->config->load("facebook", TRUE);
    $config = $CI->config->item('facebook');
    $this->load->library('Facebook', $config);

    // Try to get the user's id on Facebook
    $userId = $this->facebook->getUser();

    // If user is not yet authenticated, the id will be zero
    if ($userId == 0) {
        // Generate a login url
        $data['url'] = $this->facebook->getLoginUrl(array('scope'=>'email', 'publish_actions'
            '/*', 'redirect_uri' => 'https://agila.upm.edu.ph/~pjdizon/FIM/home_user/
            share_achievement_fb'*/));
        $url = $data['url'];
        echo "<a href='$url'>Login</a>";

        // $this->load->view('share_achievement_fb_index', $data);
    }
    else {
        // Get user's data and print it
        // $user = $this->facebook->api('/me');
        // print_r($user);
        // $data['user_profile'] = $this->facebook->api('/me');
        $this->facebook->api('/me/feed/', 'POST', array(
            'message' => $this->input->post('achievement'),
            'redirect_uri' => 'https://agila.upm.edu.ph/~pjdizon/
            FIM/home_user'));
    }
}

```

```

        $message = "Posting of Achievement to Facebook
        Successful!";

        if ((isset($message) && ($message != ''))) {
            echo '<script>
                alert("'.str_replace(array("\r","\n"),
                '', $message).'");
                location.replace("../home_user");
            </script >';
        }
    }
}

/*function post_twitter() {
    // It really is best to auto-load this library!
    $this->load->library('tweet');

    // Enabling debug will show you any errors in the calls you're making,
    // e.g:
    $this->tweet->enable_debug(TRUE);

    // If you already have a token saved for your user
    // (in a db for example) - See line #37
    //
    // You can set these tokens before calling logged_in to try using the
    // existing tokens.
    // $tokens = array('oauth_token' => 'foo', 'oauth_token_secret' => '
    // bar');
    $tokens = array('oauth_token' => '787163882-5
    XZQdNXKxKfxs7CMpnK8KOQfGG96p9obqSINTcab', 'oauth_token_secret' =>
    'IUqdGhuXJMbp2LzhAsxFPxBOwPf7B5jek85ZugKHJ9qA');
    $this->tweet->set_tokens($tokens);

    if ( !$this->tweet->logged_in() ) {
        // This is where the url will go to after auth.
        // ( Callback url )
        $this->tweet->set_callback(site_url('home_user/auth'));
        // Send the user off for login!
        $this->tweet->login();
    }
    else {
        // You can get the tokens for the active logged in user:
        // $tokens = $this->tweet->get_tokens();

        //
        // These can be saved in a db alongside a user record
        // if you already have your own auth system.
    }
    /*$message = "Posting of Achievement to Twitter Successful!";

        if ((isset($message) && ($message != ''))) {
            echo '<script>
                alert("'.str_replace(array("\r","\n"),
                '', $message).'");
                location.replace("../home_user");
            </script >';
        }*/
        // $tokens = $this->tweet->get_tokens();

    // $user = $this->tweet->call('get', 'account/
    // verify_credentials');
    //
    // Will throw an error with a stacktrace.

    /*$user = $this->tweet->call('get', 'account/verify_credentials');
    var_dump($user);

    $friendship = $this->tweet->call('get', 'friendships/show', array(
    'source_screen_name' => $user->screen_name, 'target_screen_name'
    => 'pjud31'));
    var_dump($friendship);

    if ( $friendship->relationship->target->following == FALSE )
    {
        $this->tweet->call('post', 'friendships/create', array('
        screen_name' => $user->screen_name, 'follow' => TRUE));
    }

    $this->tweet->call('post', 'statuses/update', array('status' => $this
    ->input->post('achievement')));

    $options = array(
        'count' => 10,
        'page' => 2,
        'include_entities' => 1
    );

    $timeline = $this->tweet->call('get', 'statuses/home_timeline');
    var_dump($timeline);
}*/
}

/*function auth() {
    $tokens = $this->tweet->get_tokens();
}
}

```

```

// $user = $this->tweet->call('get', 'account/
    verify_credentials');
//
// Will throw an error with a stacktrace.
$user = $this->tweet->call('get', 'account/verify_credentials');
var_dump($user);

$friendship = $this->tweet->call('get', 'friendships/show', array(
    ('source_screen_name' => $user->screen_name, 'target_screen_name'
=> 'pjud31'));
var_dump($friendship);

if ( $friendship->relationship->target->following == FALSE )
{
    $this->tweet->call('post', 'friendships/create', array('
        screen_name' => $user->screen_name, 'follow' => TRUE));
}

$this->tweet->call('post', 'statuses/update', array('status' => $this
->input->post('achievement')));

$options = array(
    'count' => 10,
    'page' => 2,
    'include_entities' => 1
);

$timeline = $this->tweet->call('get', 'statuses/home.timeline');
var_dump($timeline);
}*/

}
?>

```

---

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Login extends CI_Controller {

    function __construct() {
        parent::__construct();
        $this->load->model('login_model');
    }

    function index() {
        $this->load->helper(array('form'));
        $this->load->helper('captcha');
        $vals = array(
            'img_path' => './captcha/',
            'img_url' => 'https://agila.upm.edu.ph/~pjdizon/FIM/captcha/'
        );
        $cap = create_captcha($vals);
        $data = array(
            'captcha_time' => $cap['time'],
            'ip_address' => $this->input->ip_address(),
            'word' => $cap['word']
        );
        $this->session->set_userdata($data);
        $data['cap_img'] = $cap['image'];
        $this->load->view('login_index', $data);
    }

    function save() {
        $this->load->library('form_validation');
        // field name, error message, validation rules
        $this->form_validation->set_rules('username_reg', 'Username', 'trim|
            required|alpha_numeric|min_length[4]|xss_clean|
            callback_check_user_ci');
        $this->form_validation->set_rules('password_reg', 'Password', 'trim|
            required|alpha_numeric|min_length[8]');
        $this->form_validation->set_rules('confirm_password_reg', 'Confirm
            Password', 'trim|required|alpha_numeric|matches[password_reg]');
        $this->form_validation->set_rules('first_name', 'First Name', 'trim|
            required|alpha_numeric|xss_clean');
        $this->form_validation->set_rules('middle_name', 'Middle Name', '
            alpha_numeric|xss_clean');
        $this->form_validation->set_rules('last_name', 'Last Name', 'trim|
            required|alpha_numeric|xss_clean');
        $this->form_validation->set_rules('birthdate', 'Birthdate', '
            required|xss_clean');
        $this->form_validation->set_rules('genderRadio', 'Gender', 'required
            ');
        $this->form_validation->set_rules('userTypeRadio', 'User Type', '
            required');
        $this->form_validation->set_rules('email_address', 'Email Address',
            'required|valid_email|callback_check_email_ci');
        $this->form_validation->set_rules('captcha', 'Security Code', 'trim|
            required|callback_check_captcha');

        if($this->form_validation->run() == FALSE) {
            $this->index();
        }
        else {
            $this->load->model('login_model');
            if($this->input->post('submit')) {

```

```

        $this->login_model->process();
    }

    $message = "Registration Successful!";
    if ((isset($message) && ($message != '')) {
        echo '<script>
            alert("'.str_replace(array("\r","\n"), '',
                $message).'");
            location.replace("../login");
        </script>';
    }
}

}

public function check_captcha() {
    $expiration = time() - 7200; // Two hour limit
    $cap=$this->input->post('captcha');
    if($this->session->userdata('word')== $cap
    AND $this->session->userdata('ip_address')== $this->input->
    ip_address()
    AND $this->session->userdata('captcha-time')> $expiration)
    {
        return true;
    }
    else {
        $this->form_validation->set_message('check_captcha', 'Security
        number does not match. ');
        return false;
    }
}

public function check_user_ci() {
    $usr=$this->input->post('username_reg');
    $result=$this->login_model->check_user_exist($usr);
    if($result) {
        $this->form_validation->set_message(__FUNCTION__, 'Username
        already exists. ');
        return false;
    }
    else {
        return true;
    }
}

public function check_user() {
    $usr=$this->input->post('name');
    $result=$this->login_model->check_user_exist($usr);
    if($result) {
        echo "false";
    }
    else {
        echo "true";
    }
}

public function check_email_ci() {
    $eml=$this->input->post('email_address');
    $result=$this->login_model->check_email_exist($eml);
    if($result) {
        $this->form_validation->set_message(__FUNCTION__, 'E-mail Address
        already registered. ');
        return false;
    }
    else {
        return true;
    }
}

public function check_email() {
    $eml=$this->input->post('email');
    $result=$this->login_model->check_email_exist($eml);
    if($result) {
        echo "false";
    }
    else {
        echo "true";
    }
}

//FORGOT PASSWORD CONTROLLERS
public function forget_password() {
    $this->load->view('login.forget_index');
}

public function doforget() {
    $this->load->library('form_validation');

    $this->form_validation->set_rules('email_address', 'Email Address', 'required|
    valid_email|callback_check_email_ci_exists');

    if($this->form_validation->run() == FALSE) {
        //$this->index();
        $this->load->view('login.forget_index');
    }
    else {
        $this->load->helper('url');
        $email_address= $_POST['email_address'];
        $q = $this->db->query("select * from users where email_address='".$
        $email_address."'");
    }
}

```

```

        if ($q->num_rows > 0) {
            $r = $q->result();
            $user=$r[0];
            $this->resetpassword($user);
        }
        $message = "Email Sent. Please check your email for details.";
        if ((isset($message)) && ($message != '')) {
            echo '<script>
                alert("'.str_replace(array("\r","\n"), '',
                    $message).'");
                location.replace("../login");
            </script >';
        }
    }

    private function resetpassword($user) {
        date_default_timezone_set('GMT');
        $this->load->helper('string');
        $password= random_string('alnum', 16);
        $this->db->where('username', $user->username);
        $this->db->update('users', array('password'=>MD5($password), 'confirm_password'
            =>MD5($password)));
        $ci = get_instance();
        $ci->load->library('email');
        $config['protocol'] = "smtp";
        $config['smtp_host'] = "ssl://smtp.gmail.com";
        $config['smtp_port'] = "465";
        $config['smtp_user'] = "heasfim.noreply@gmail.com";
        $config['smtp_pass'] = "sasablayakothissem2015";
        $config['charset'] = "utf-8";
        $config['mailtype'] = "html";
        $config['newline'] = "\r\n";

        $ci->email->initialize($config);

        $ci->email->from('heasfim.noreply@gmail.com', 'HEAS:FIM');
        // $list = array('xxx@gmail.com');
        $ci->email->to($user->email_address);
        // $this->email->reply_to('my_email@gmail.com', 'My Email');
        $ci->email->subject('Password Reset');
        $ci->email->message('You have requested a new password. Here is your new password:'.
            $password);
        $ci->email->send();
    }

    public function check_email_ci_exists() {
        $email=$this->input->post('email_address');
        $result=$this->login_model->check_email_exist($email);
        if($result) {
            return true;
        }
        else {
            $this->form_validation->set_message(__FUNCTION__, 'E-mail Address does
                not exist. ');
            return false;
        }
    }
}

?>

```

---

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
session_start(); //we need to call PHP's session object to access it through CI

```

```

class Logout extends CI_Controller {

    function __construct() {
        parent::__construct();
    }

    //LOGOUT
    function logout() {
        $this->session->unset_userdata('logged_in');
        session_destroy();
        redirect('login', 'refresh');
    }

}

?>

```

---

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

```

```

class VerifyLogin extends CI_Controller {

    function __construct() {
        parent::__construct();
        $this->load->model('login_model', '', TRUE);
    }

    function index() {
        //This method will have the credentials validation
        $this->load->library('form_validation');
    }
}

```

```

$this->form_validation->set_rules('username', 'Username', 'trim|required|xss_clean');
$this->form_validation->set_rules('password', 'Password', 'trim|required|xss_clean|callback_check_database');

if($this->form_validation->run() == FALSE) {
    //Field validation failed. User redirected to login page
    // $this->load->view('login_index');
    $this->load->helper(array('form'));
    $this->load->helper('captcha');
    $vals = array(
        'img_path' => './captcha/',
        'img_url' => 'https://agila.upm.edu.ph/~pjdizon/FIM/captcha/'
    );
    $cap = create_captcha($vals);
    $data = array(
        'captcha_time' => $cap['time'],
        'ip_address' => $this->input->ip_address(),
        'word' => $cap['word']
    );
    $this->session->set_userdata($data);
    $data['cap_img'] = $cap['image'];
    $this->load->view('login_index', $data);
}
else {
    //Go to private area
    redirect('home', 'refresh');
}
}

function check_database($password) {
    //Field validation succeeded. Validate against database
    $username = $this->input->post('username');
    $first_name = $this->input->post('first_name');
    $user_type = $this->input->post('user_type');

    //query the database
    $result = $this->login_model->login($username, $password);

    if($result) {
        $sess_array = array();
        foreach($result as $row) {
            $sess_array = array(
                'user_id' => $row->user_id,
                'username' => $row->username,
                'first_name' => $row->first_name,
                'middle_name' => $row->middle_name,
                'last_name' => $row->last_name,
                'user_type' => $row->user_type,
                'email_address' => $row->email_address,
                'gender' => $row->gender,
                'birthday' => $row->birthday,
                'status' => $row->status
            );
        }
        if($row->status != "Deactivated") {
            $this->session->set_userdata('logged_in', $sess_array);
            return TRUE;
        }
        else {
            $this->form_validation->set_message('check_database', 'Your account might be deactivated at the moment.');
```

---

## B.3 Models

```
<?php
```

```

class Home_Admin_Model extends CI_Model {
    //USER SEARCH MODEL/S
    function GetAutocomplete($options = array()) {
        $this->db->select('user_id, username, first_name, middle_name, last_name,
            user_type, email_address, gender, birthday');
        $this->db->like('username', $options['keyword'], 'after');
        $query = $this->db->get('users');
        return $query->result();
    }

    function get_search() {
        $match = $this->input->post('search');
        $this->db->like('user_id', $match);
        $this->db->or_like('username', $match);
        $this->db->or_like('first_name', $match);
        $this->db->or_like('middle_name', $match);
        $this->db->or_like('last_name', $match);
    }
}

```

```

        $this->db->or_like('user_type', $match);
        $this->db->or_like('email_address', $match);
        $this->db->or_like('gender', $match);
        $this->db->or_like('birthday', $match);
        $this->db->order_by('user_id', 'desc');

        $query = $this->db->get('users');
        return $query->result();
    }

    //ADD USER MODEL/S
    function process(){
        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $confirm_password = $this->input->post('confirm_password');
        $first_name = $this->input->post('first_name');
        $middle_name = $this->input->post('middle_name');
        $last_name = $this->input->post('last_name');
        $gender = $this->input->post('genderRadio');
        $user_type = $this->input->post('userTypeRadio');
        $email_address = $this->input->post('email_address');
        $birthday = $this->input->post('birthdate');

        $data = array(
            'username' => $username,
            'password' => MD5($password),
            'confirm_password' => MD5($confirm_password),
            'first_name' => $first_name,
            'middle_name' => $middle_name,
            'last_name' => $last_name,
            'gender' => $gender,
            'user_type' => $user_type,
            'email_address' => $email_address,
            'birthday' => $birthday
        );
        $this->db->insert('users', $data);
    }

    public function check_user_exist($usr) {
        $this->db->where("username", $usr);
        $query=$this->db->get("users");
        if($query->num_rows()>0) {
            return true;
        }
        else {
            return false;
        }
    }

    public function check_email_exist($eml) {
        $this->db->where("email_address", $eml);
        $query=$this->db->get("users");
        if($query->num_rows()>0) {
            return true;
        }
        else {
            return false;
        }
    }

    //DELETE USER MODEL
    public function del_user($username) {
        $this->db->where('username', $username);
        $this->db->delete('users');
    }

    //EDIT USER MODEL
    public function update_user($username, $data) {
        $this->db->where('username', $username);
        $this->db->update('users', $data);
    }

    function get_user($username) {
        $this->db->select('username, first_name, middle_name, last_name, user_type,
            email_address, gender, birthday, status');
        $this->db->where('username', $username);
        $query = $this->db->get('users');

        return $query->row_array();
        //return $query->result();
    }

    //EDIT PROFILE MODEL
    public function update_profile($username, $data) {
        $this->db->where('username', $username);
        $this->db->update('users', $data);
    }

    //EDIT PASSWORD MODEL
    public function update_password($username, $data) {
        $this->db->where('username', $username);
        $this->db->update('users', $data);
    }

    public function check_old_password_exist($opwd) {
        $this->db->where("password", $opwd);
        $query=$this->db->get("users");
        if($query->num_rows()>0) {
            return true;
        }
    }

```

```

        else {
            return false;
        }
    }
}
?>

```

---

```

<?php
class Home_Cmanager_Model extends CI_Model {

    public function check_user_exist($usr) {
        $this->db->where("username",$usr);
        $query=$this->db->get("users");
        if($query->num.rows()>0) {
            return true;
        }
        else {
            return false;
        }
    }

    public function check_email_exist($eml) {
        $this->db->where("email_address",$eml);
        $query=$this->db->get("users");
        if($query->num.rows()>0) {
            return true;
        }
        else {
            return false;
        }
    }

    //EDIT PROFILE MODEL
    public function update_profile($username, $data) {
        $this->db->where('username', $username);
        $this->db->update('users', $data);
    }

    //EDIT PASSWORD MODEL
    public function update_password($username, $data) {
        $this->db->where('username', $username);
        $this->db->update('users', $data);
    }

    public function check_old_password_exist($opwd) {
        $this->db->where("password", $opwd);
        $query=$this->db->get("users");
        if($query->num.rows()>0) {
            return true;
        }
        else {
            return false;
        }
    }

    //ADD HEALTH TIP MODEL
    function process_healthtip() {
        $health_tip = $this->input->post('health_tip');
        if($this->session->userdata('logged_in')) {
            $session_data = $this->session->userdata('logged_in');
            $user_id= $session_data['user_id'];
            $data['user_id'] = $user_id;
            $data['health_tip'] = $health_tip;
            $data['post_date'] = date('y-m-d');
            $this->db->insert('health_tips',$data);
        }
        else
            echo('Session Timeout.');
```

```

    }

    //DELETE HEALTH TIP MODEL
    public function del_health_tip($health_tip_id) {
        // $this->db->where('fitness_info_id', $fitness_info_id);
        $this->db->delete('health_tips', array('health_tip_id' => $health_tip_id));
    }

    //ADD FOOD EXCHANGE LIST ITEM
    public function check_felitem_exist($food_name) {
        $this->db->where("food_name",$food_name);
        $query=$this->db->get("food_exchange_list");
        if($query->num.rows()>0) {
            return true;
        }
        else {
            return false;
        }
    }

    function process_felitem(){
        if(isset($_POST['list_groupother'], $_POST['food_groupother'], $_POST['
            food_group_categoryother'])) {
            $list_group = $this->input->post('list_groupother');
            $food_group = $this->input->post('food_groupother');
            $food_group_category = $this->input->post('food_group_categoryother');
        } else {

```



```

        $list_group = $this->input->post('list_group');
        $food_group = $this->input->post('food_group');
        $food_group_category = $this->input->post('food_group_category');
    }

    $food_name = $this->input->post('food_name');
    $measure = $this->input->post('measure');
    $carbohydrate = $this->input->post('carbohydrate');
    $protein = $this->input->post('protein');
    $fat = $this->input->post('fat');
    $energy = $this->input->post('energy');
    $weight_as_purchased = $this->input->post('weight_as_purchased');
    $weight_edible_portion = $this->input->post('weight_edible_portion');

    $data = array(
        'list_group' => $list_group,
        'food_group' => $food_group,
        'food_group_category' => $food_group_category,
        'food_name' => $food_name,
        'measure' => $measure,
        'carbohydrate' => $carbohydrate,
        'protein' => $protein,
        'fat' => $fat,
        'energy' => $energy,
        'weight_as_purchased' => $weight_as_purchased,
        'weight_edible_portion' => $weight_edible_portion
    );
    $this->db->insert('food_exchange_list', $data);
}

//DELETE FEL ITEM
public function del_fel_item($food_exchange_list_id) {
    // $this->db->where('fitness_info_id', $fitness_info_id);
    $this->db->delete('food_exchange_list', array('food_exchange_list_id' =>
        $food_exchange_list_id));
}

function get_search_fel() {
    $match = $this->input->post('search_fel');
    $this->db->like('list_group', $match);
    $this->db->or_like('food_group', $match);
    $this->db->or_like('food_group_category', $match);
    $this->db->or_like('food_name', $match);
    $this->db->or_like('measure', $match);
    $this->db->or_like('carbohydrate', $match);
    $this->db->or_like('protein', $match);
    $this->db->or_like('fat', $match);
    $this->db->or_like('energy', $match);
    $this->db->or_like('weight_as_purchased', $match);
    $this->db->or_like('weight_edible_portion', $match);

    $query = $this->db->get('food_exchange_list');
    return $query->result();
}

function get_search_ht() {
    $match = $this->input->post('search_ht');
    $this->db->like('health_tip', $match);
    $this->db->order_by('health_tip_id', 'desc');

    $query = $this->db->get('health_tips');
    return $query->result();
}

//VIEW CREATED RECIPES MODEL
function process_view_created_recipes() {
    if ($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id = $session_data['user_id'];
        $data['user_id'] = $user_id;

        /* $this->db->select('o.*, i.*');
        $this->db->from('own_recipes o, ingredients i');
        $this->db->where('o.user_id', $user_id);
        $this->db->where('o.own_recipe_id = i.own_recipe_id');
        $query = $this->db->get();
        return $query->result();*/

        $this->db->select('*');
        $this->db->from('recommended_recipes');
        $this->db->where("user_id", $user_id);
        $this->db->order_by('recommended_recipe_id', 'desc');
        $query = $this->db->get();
        return $query->result();
    }
}

//EDIT CREATED RECIPE
function get_created_recipe($recommended_recipe_id) {
    $this->db->select('o.*, i.*, r.*');
    $this->db->from('recommended_recipes o, ingredients_recommended i,
        recipe_images_recommended r');
    $this->db->where('o.recommended_recipe_id', $recommended_recipe_id);
    $this->db->where('o.recommended_recipe_id = i.recommended_recipe_id');
    $this->db->where('o.recommended_recipe_id = r.recommended_recipe_id');
    $query = $this->db->get();

    //return $query->row_array();
    return $query->result();
}

```

```

}

//UPDATE RECIPE MODEL
public function update_recipe($recommended_recipe_id, $data,$data2) {
    $this->db->where('recommended_recipe_id', $recommended_recipe_id);
    $this->db->update('recommended_recipes', $data);
    // $own_recipe_id = $this->db->insert_id();

    // $this->db->update('ingredients', $data2);
    for($i = 0; $i < count($data2['food_name']); $i++) {
        $data3 = array(
            'food_name'=>$data2['food_name'][$i],
            'food_group'=>$data2['food_group'][$i],
            'measure'=>$data2['measure'][$i],
            'serving_size'=>$data2['serving_size'][$i],
        );
        if($data2['ingredient_id'][$i] == null || $data2['ingredient_id'][$i]
            == "") {
            $data3['recommended_recipe_id'] = $recommended_recipe_id;
            $this->db->insert('ingredients_recommended', $data3);
        } else {
            $this->db->where('ingredient_id', $data2['ingredient_id'][$i]);
            $this->db->update('ingredients_recommended', $data3);
        }
    }
}

//VIEW RECIPE
function get_recipe_contents($recommended_recipe_id) {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id= $session_data['user_id'];
        $data['user_id'] = $user_id;

        /* $this->db->select('o.*, i.*');
        $this->db->from('own_recipes o, ingredients i');
        $this->db->where('o.user_id', $user_id);
        $this->db->where('o.own_recipe_id = i.own_recipe_id');
        $query = $this->db->get();
        return $query->result();*/

        /* $this->db->select('*');
        $this->db->from('ingredients');
        $this->db->where("own_recipe_id", $own_recipe_id);
        $query = $this->db->get();
        return $query->result();*/

        $this->db->select('r.*, i.*, c.*');
        $this->db->from('recommended_recipes r, ingredients_recommended i,
            recipe_images_recommended c');
        $this->db->where('i.recommended_recipe_id', $recommended_recipe_id);
        $this->db->where('i.recommended_recipe_id = r.recommended_recipe_id');
        $this->db->where('c.recommended_recipe_id = r.recommended_recipe_id');
        $query = $this->db->get();
        return $query->result();
    }
}

//DELETE CREATED RECIPE
public function del_created_recipe($recommended_recipe_id) {
    // $this->db->where('fitness_info_id', $fitness_info_id);
    $this->db->delete('recommended_recipes', array('recommended_recipe_id' =>
        $recommended_recipe_id));
    $this->db->delete('ingredients_recommended', array('recommended_recipe_id' =>
        $recommended_recipe_id));
    $this->db->delete('recipe_images_recommended', array('recommended_recipe_id'
        => $recommended_recipe_id));
}

//DELETE INGREDIENTS
public function del_ingredient($ingredient_id) {
    $this->db->delete('ingredients_recommended', array('ingredient_id' =>
        $ingredient_id));
}

//ADD RECIPE MODEL
function process_addrecipe($data3){
    // $health_condition = $this->input->post('health_condition');
    // $health_condition = implode(" ", $this->input->post('health_condition'));
    $meal_type = $this->input->post('meal_type');
    $recipe_name = $this->input->post('recipe_name');
    $serving_sugg = $this->input->post('serving_sugg');
    $preparation_procedure = $this->input->post('preparation_procedure');
    $calories = $this->input->post('calories');
    $carbohydrate = $this->input->post('carbohydrate');
    $protein = $this->input->post('protein');
    $fat = $this->input->post('fat');

    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id= $session_data['user_id'];

        $data['user_id'] = $user_id;
        $data['health_condition'] = $health_condition;
        $data['meal_type'] = $meal_type;
        $data['recipe_name'] = $recipe_name;
        $data['serving_sugg'] = $serving_sugg;
        $data['preparation_procedure'] = $preparation_procedure;
    }
}

```

```

        $data['calories'] = $calories;
        $data['carbohydrate'] = $carbohydrate;
        $data['protein'] = $protein;
        $data['fat'] = $fat;

        $this->db->insert('recommended_recipes', $data);

        $recommended_recipe_id = $this->db->insert_id();

        if (isset($_POST['ingredients'], $_POST['food_group'], $_POST['ingredient_measure'], $_POST['ingredient_quantity'])) {
            for ($i=0; $i<count($_POST['ingredients']); $i++) {
                $ingredients = $_POST['ingredients'][$i];
                $food_group = $_POST['food_group'][$i];
                $ingredient_measure = $_POST['ingredient_measure'][$i];
                $ingredient_quantity = $_POST['ingredient_quantity'][$i];
                $data2['recommended_recipe_id'] =
                    $recommended_recipe_id;
                $data2['food_name'] = $ingredients;
                $data2['food_group'] = $food_group;
                $data2['measure'] = $ingredient_measure;
                $data2['serving_size'] = $ingredient_quantity;
                /*
                $counts = array_count_values($food_group);
                echo $counts['Vegetable'];
                */
                $this->db->insert('ingredients_recommended', $data2);
            }
        }

        $data3['recommended_recipe_id'] = $recommended_recipe_id;

        $this->db->insert('recipe_images_recommended', $data3);

    }
    else
        echo "Session Timeout";
}

//ADD INGREDIENT AUTOCOMPLETE
function GetAutocomplete($options = array()) {
    $this->db->select('food_group, food_name, weight_edible_portion, energy, carbohydrate, protein, fat');
    $this->db->like('food_name', $options['keyword'], 'after');
    $query = $this->db->get('food_exchange_list');
    return $query->result();
}

//SEARCH FOR MEAL MODEL
function get_search_meal() {
    $match = $this->input->post('search_meal');
    $this->db->like('recommended_recipe_id', $match);
    $this->db->or_like('user_id', $match);
    $this->db->or_like('health_condition', $match);
    $this->db->or_like('meal_type', $match);
    $this->db->or_like('recipe_name', $match);
    $this->db->or_like('serving_sugg', $match);
    $this->db->or_like('preparation_procedure', $match);
    $this->db->or_like('calories', $match);
    $this->db->or_like('carbohydrate', $match);
    $this->db->or_like('protein', $match);
    $this->db->or_like('fat', $match);
    $this->db->order_by('recommended_recipe_id', 'desc');
    $query = $this->db->get('recommended_recipes');
    return $query->result();
}

//ADD MEAL
function process_addmeal($recommended_recipe_id) {
    if ($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id = $session_data['user_id'];
        $meal_group = $this->input->post('meal_group');
        $record_date = $this->input->post('record_date');

        $data['user_id'] = $user_id;
        $data['recommended_recipe_id'] = $recommended_recipe_id;
        $data['meal_group'] = $meal_group;
        $data['record_date'] = $record_date;

        $this->db->insert('recommended_daily_meals', $data);
    }
    else
        echo "Session Timeout";
}

//DELETE MEAL
public function del_meal($recommended_daily_meal_id) {
    // $this->db->where('fitness_info_id', $fitness_info_id);
    $this->db->delete('recommended_daily_meals', array('recommended_daily_meal_id' => $recommended_daily_meal_id));
}

//VIEW DAILY MEALS BY DATE MODEL
public function show_data_by_date($record_date) {
    if ($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
    }
}

```

```

$user_id= $session_data['user_id'];
//$data['user_id'] = $user_id;
//$condition = "record_date = " . "''" . $record_date . "''";

$this->db->select('d.*, r.*');
$this->db->from('recommended_daily_meals d, recommended_recipes r');
$this->db->where("d.user_id", $user_id);
$this->db->where('d.record_date', $record_date);
$this->db->where('d.recommended_recipe_id = r.recommended_recipe_id');
$this->db->order_by('meal_group', 'asc');

$query = $this->db->get();

if ($query->num_rows() > 0) {
    return $query->result();
} else {
    return false;
}
}

//EDIT FEL MODEL
public function update_fel($food_exchange_list_id, $data) {
    $this->db->where('food_exchange_list_id', $food_exchange_list_id);
    $this->db->update('food_exchange_list', $data);
}

function get_fel_edit($food_exchange_list_id) {
    $this->db->select('*');
    $this->db->where('food_exchange_list_id', $food_exchange_list_id);
    $query = $this->db->get('food_exchange_list');

    return $query->row_array();
    //return $query->result();
}

//EDIT HT MODEL
public function update_ht($health_tip_id, $data) {
    $this->db->where('health_tip_id', $health_tip_id);
    $this->db->update('health_tips', $data);
}

function get_ht_edit($health_tip_id) {
    $this->db->select('*');
    $this->db->where('health_tip_id', $health_tip_id);
    $query = $this->db->get('health_tips');

    return $query->row_array();
    //return $query->result();
}

//GET FEL LIST GROUP
public function get_list_group(){
    $this->db->select('list_group');
    //$this->db->distinct();
    $query = $this->db->get('food_exchange_list');
    foreach($query->result_array() as $row){
        $data[$row['list_group']]=$row['list_group'];
    }
    //$data['Other'] = 'Other';
return $data;
}

//GET FEL FOOD GROUP BY LIST GROUP
public function get_food_group($li_group=string){
    $this->db->select('list_group, food_group');
    $this->db->where('list_group', $li_group);
    $this->db->distinct();
    $query = $this->db->get('food_exchange_list');
    return $query;
}

//GET FEL FOOD GROUP CATEGORY BY LIST GROUP
public function get_food_group_category($li_group=string){
    $this->db->select('list_group, food_group_category');
    $this->db->where('list_group', $li_group);
    $this->db->distinct();
    $query = $this->db->get('food_exchange_list');
    return $query;
}
}
?>

```

---

```
<?php
```

```

class Home_User_Model extends CI_Model {
function __construct()
{
    // Call the Model constructor
    parent::__construct();
}

public function check_user_exist($usr) {
    $this->db->where("username", $usr);
    $query=$this->db->get("users");
    if ($query->num_rows()>0) {
        return true;
    }
}
}

```

```

    }
    else {
        return false;
    }
}

public function check_email_exist($eml) {
    $this->db->where("email_address", $eml);
    $query=$this->db->get("users");
    if($query->num_rows()>0) {
        return true;
    }
    else {
        return false;
    }
}

//EDIT PROFILE MODEL
public function update_profile($username, $data) {
    $this->db->where('username', $username);
    $this->db->update('users', $data);
}

//EDIT PASSWORD MODEL
public function update_password($username, $data) {
    $this->db->where('username', $username);
    $this->db->update('users', $data);
}

public function check_old_password_exist($opwd) {
    $this->db->where("password", $opwd);
    $query=$this->db->get("users");
    if($query->num_rows()>0) {
        return true;
    }
    else {
        return false;
    }
}

//SEARCH FOOD EXCHANGE LIST MODEL
function get_search_fel() {
    $match = $this->input->post('search_fel');
    $this->db->like('list_group', $match);
    $this->db->or_like('food_group', $match);
    $this->db->or_like('food_group_category', $match);
    $this->db->or_like('food_name', $match);
    $this->db->or_like('measure', $match);
    $this->db->or_like('carbohydrate', $match);
    $this->db->or_like('protein', $match);
    $this->db->or_like('fat', $match);
    $this->db->or_like('energy', $match);
    $this->db->or_like('weight_as_purchased', $match);
    $this->db->or_like('weight_edible_portion', $match);

    $query = $this->db->get('food_exchange_list');
    return $query->result();
}

//HEALTHTIPS FEED MODEL
/*public function count_ht() {
    return count($this->get_search_ht());
}

public function get_search_ht_ret() {
    $params = func_get_args();
    $limit = $params[0]['limit'] ? $params[0]['limit'] : 0;
    $offset = $params[0]['offset'] ? $params[0]['offset'] : 0;

    $query = $this->get_search_ht();
    $ret_array = array();

    for ($i = 0; $i < $limit; $i++) {
        if ($offset < count($query)) {
            $ret_array[] = $query[$offset];
            $offset++;
        }
    }

    return $ret_array;
}*/

function get_search_ht() {
    $this->db->order_by("post_date", "desc");
    //$this->db->limit(5);
    $query = $this->db->get('health_tips');
    return $query->result();
}

//ACHIEVEMENTS FEED MODEL
function get_search_achievements() {
    $this->db->order_by("shared_achievement_id", "desc");
    $this->db->limit(5);
    $query = $this->db->get('shared_achievements');
    return $query->result();
}

//ADD FP INFO MODEL
function process_addfp() {

```

```

$goal_weight = $this->input->post('goal_weight');
$current_weight = $this->input->post('current_weight');
$height = $this->input->post('height');

// $health_condition = $this->input->post('health_condition');
$health_condition = implode(",",$this->input->post('health_condition'));

        //print_r($health_condition);
        //die();

$record_date = $this->input->post('record_date');

if($this->session->userdata('logged_in')) {
    $session_data = $this->session->userdata('logged_in');
    $user_id= $session_data['user_id'];

    $this->db->select('birthday');
    $this->db->from('users');
    $this->db->where(" user_id", $user_id);
    $query_birthday = $this->db->get();
    $query_birthday->result();
    foreach ($query_birthday->result() as $row) {
        //echo $row->birthday;
        $birthday = $row->birthday;
        //echo $birthday;
    }

    $this->db->select('gender');
    $this->db->from('users');
    $this->db->where(" user_id", $user_id);
    $query_gender = $this->db->get();
    $query_gender->result();
    foreach ($query_gender->result() as $row) {
        //echo $row->birthday;
        $gender = $row->gender;
        //echo $gender;
    }

    if($gender == "Male"){
        $age = $record_date - $birthday;
        $calorie_limit = 66 + (13.7 * $current_weight) + (5 * $height)
            - (6.8 * $age);
        $carbohydrate_limit = 0.65 * $calorie_limit;
        $protein_limit = 0.25 * $calorie_limit;
        $fat_limit = 0.70 * $calorie_limit;
    } else {
        $age = $record_date - $birthday;
        $calorie_limit = 655 + (9.6 * $current_weight) + (1.7 *
            $height) - (4.7 * $age);
        $carbohydrate_limit = (0.65 * $calorie_limit) / 4;
        $protein_limit = (0.25 * $calorie_limit) / 4;
        $fat_limit = (0.70 * $calorie_limit) / 9;
    }

    $data['user_id'] = $user_id;
    $data['goal_weight'] = $goal_weight;
    $data['current_weight'] = $current_weight;
    $data['height'] = $height;
    $data['calorie_limit'] = $calorie_limit;
    $data['carbohydrate_limit'] = $carbohydrate_limit;
    $data['protein_limit'] = $protein_limit;
    $data['fat_limit'] = $fat_limit;
    $data['record_date'] = $record_date;
    $this->db->insert('fitness_profile', $data);

    $fitness_info_id = $this->db->insert_id();
    $data2['health_condition'] = $health_condition;
    $data2['user_id'] = $user_id;
    $data2['fitness_info_id'] = $fitness_info_id;
    $data2['record_date'] = $record_date;
    /* if (isset($_POST['health_condition'])) {
        for ( $i=0;$i<count($_POST['health_condition']);$i++) {
            $health_condition = $_POST['health_condition'][$i];
            $data2['user_id'] = $user_id;
            $data2['fitness_info_id'] = $fitness_info_id;
            $data2['health_condition'] = $health_condition;
            $data2['record_date'] = $record_date;*/

            $this->db->insert('health_conditions', $data2);
        }
    }

    else
        echo "Session Timeout";
}

public function check_fp_exist($rec_date) {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id= $session_data['user_id'];
        //$data['user_id'] = $user_id;
    }

    $this->db->where(" record_date", $rec_date);
    $this->db->where(" user_id", $user_id);
    $query=$this->db->get(" fitness_profile");
    if($query->num_rows()>0) {
        return true;
    }
}

```

```

    }
    else {
        return false;
    }
}

//VIEW FP INFO MODEL
function process_viewfp() {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id= $session_data['user_id'];
        $data['user_id'] = $user_id;

        /*$this->db->order_by(" record_date", "desc");
        //$this->db->limit(1);
        $this->db->where(" user_id", $user_id);
        $query = $this->db->get('fitness_profile ');
        return $query->result();

        $this->db->select('health_conditions ');
        $this->db->from('health_conditions ');
        $this->db->where(" user_id", $user_id);
        $query = $this->db->get();
        return $query->result();*/

        /*
        SELECT * FROM fitness_profile f INNER JOIN health_conditions h ON f.
            user_id = h.user_id WHERE f.fitness_info_id = h.fitness_info_id
        */
        $this->db->select('f.*, h.*');
        $this->db->from('fitness_profile f, health_conditions h');
        $this->db->where("f.user_id", $user_id);
        $this->db->where('f.user_id = h.user_id');
        $this->db->where('f.fitness_info_id = h.fitness_info_id ');
        $this->db->order_by('f.record_date ', 'desc ');

        $query = $this->db->get();
        return $query->result();
    }
}

//DELETE FITNESS PROFILE MODEL
public function del_fitness_profile($fitness_info_id) {
    //$this->db->where('fitness_info_id', $fitness_info_id);
    $this->db->delete('fitness_profile ', array('fitness_info_id ' =>
        $fitness_info_id));
    $this->db->delete('health_conditions ', array('fitness_info_id ' =>
        $fitness_info_id));
}

//WEIGHT GRAPH CHART MODEL
function get_weightchart_data() {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id= $session_data['user_id'];
        $data['user_id'] = $user_id;

        $this->db->select('record_date , current_weight , goal_weight ');
        $this->db->from('fitness_profile ');
        $this->db->where(" user_id", $user_id);
        $this->db->order_by(" record_date", 'asc ');
        $query = $this->db->get();
        return $query->result();
    }
}

//SHARE ACHIEVEMENT MODEL
//ADD HEALTH TIP MODEL
function process_achievement() {
    $achievement = $this->input->post('achievement');
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id= $session_data['user_id'];
        $data['user_id'] = $user_id;
        $data['achievement'] = $achievement;
        $this->db->insert('shared_achievements ', $data);
    }
    else
        echo('Session Timeout. ');
}

//VIEW CREATED RECIPES MODEL
function process_view_created_recipes() {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id= $session_data['user_id'];
        $data['user_id'] = $user_id;

        /*$this->db->select('o.*, i.*');
        $this->db->from('own_recipes o, ingredients i');
        $this->db->where('o.user_id', $user_id);
        $this->db->where('o.own_recipe_id = i.own_recipe_id ');
        $query = $this->db->get();
        return $query->result();*/

        $this->db->select('* ');
        $this->db->from('own_recipes ');
        $this->db->where(" user_id", $user_id);
    }
}

```

```

        $this->db->order_by('own_recipe_id', 'desc');
        $query = $this->db->get();
        return $query->result();
    }
}
//EDIT CREATED RECIPE
function get_created_recipe($own_recipe_id) {
    $this->db->select('o.*, i.*, r.*');
    $this->db->from('own_recipes o, ingredients i, recipe_images r');
    $this->db->where('o.own_recipe_id', $own_recipe_id);
    $this->db->where('o.own_recipe_id = i.own_recipe_id');
    $this->db->where('o.own_recipe_id = r.own_recipe_id');
    $query = $this->db->get();

    //return $query->row_array();
    return $query->result();
}

//UPDATE RECIPE MODEL
public function update_recipe($own_recipe_id, $data, $data2) {
    $this->db->where('own_recipe_id', $own_recipe_id);
    $this->db->update('own_recipes', $data);
    // $own_recipe_id = $this->db->insert_id();

    // $this->db->update('ingredients', $data2);
    for($i = 0; $i < count($data2['food_name']); $i++) {
        $data3 = array(
            'food_name'=>$data2['food_name'][$i],
            'food_group'=>$data2['food_group'][$i],
            'measure'=>$data2['measure'][$i],
            'serving_size'=>$data2['serving_size'][$i],
        );
        if($data2['ingredient_id'][$i] == null || $data2['ingredient_id'][$i]
            == "") {
            $data3['own_recipe_id'] = $own_recipe_id;
            $this->db->insert('ingredients', $data3);
        } else {
            $this->db->where('ingredient_id', $data2['ingredient_id'][$i]);
            $this->db->update('ingredients', $data3);
        }
    }
}

//VIEW RECOMMENDED RECIPES
function process_view_recommended_recipes() {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id = $session_data['user_id'];
        $data['user_id'] = $user_id;

        /*$this->db->select('health_condition');
        $this->db->from('health_conditions');
        $this->db->where("user_id", $user_id);
        $this->db->order_by("record_date", "desc");
        $this->db->limit(1);
        $query_health_condition = $this->db->get();
        $query_health_condition->result();
        foreach ($query_health_condition->result() as $row) {
            $health_condition = $row->health_condition;
        }*/

        $this->db->select('*');
        $this->db->from('recommended_recipes');
        // $this->db->where('health_condition', $health_condition);
        $this->db->order_by('recommended_recipe_id', 'desc');
        // $this->db->where("user_id", $user_id);
        $query = $this->db->get();
        return $query->result();
    }
}

//SEARCH RECOMMENDED RECIPES
function get_search_recrec() {
    $match = $this->input->post('search_recrec');
    $this->db->like('recommended_recipe_id', $match);
    $this->db->or_like('health_condition', $match);
    $this->db->or_like('meal_type', $match);
    $this->db->or_like('recipe_name', $match);
    $this->db->or_like('serving_sugg', $match);
    $this->db->or_like('calories', $match);
    $this->db->or_like('carbohydrate', $match);
    $this->db->or_like('protein', $match);
    $this->db->or_like('fat', $match);
    $this->db->order_by('recommended_recipe_id', 'desc');

    $query = $this->db->get('recommended_recipes');
    return $query->result();
}

//VIEW SHARED RECIPES
function process_view_shared_recipes() {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id = $session_data['user_id'];
        $data['user_id'] = $user_id;

        $this->db->select('s.*, o.*');
    }
}

```



```

        $this->db->from('shared_recipes s, own_recipes o');
        $this->db->where('s.own_recipe_id = o.own_recipe_id');
        $this->db->order_by('shared_recipe_id', 'desc');
        $query = $this->db->get();
        return $query->result();
    }
}

//SEARCH SHARED RECIPES
function get_search_sharedrec() {
    $match = $this->input->post('search_sharedrec');

    $this->db->select('*');
    $this->db->from('shared_recipes');
    $this->db->join('own_recipes', 'shared_recipes.own_recipe_id = own_recipes.own_recipe_id');

    $this->db->like('shared_recipe_id', $match);
    $this->db->or_like('health_condition', $match);
    $this->db->or_like('meal_type', $match);
    $this->db->or_like('recipe_name', $match);
    $this->db->or_like('serving_sugg', $match);
    $this->db->or_like('calories', $match);
    $this->db->or_like('carbohydrate', $match);
    $this->db->or_like('protein', $match);
    $this->db->or_like('fat', $match);
    $this->db->order_by('shared_recipe_id', 'desc');

    $query = $this->db->get();
    return $query->result();
}

//VIEW DAILY MEALS BY DATE MODEL
public function recommended_show_data_by_date($record_date) {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id = $session_data['user_id'];
        $data['user_id'] = $user_id;
        // $condition = "record_date = " . "" . $record_date . "" ;

        $this->db->select('health_condition');
        $this->db->from('health_conditions');
        $this->db->where("user_id", $user_id);
        $this->db->order_by("record_date", "desc");
        $this->db->limit(1);
        $query_health_condition = $this->db->get();
        $query_health_condition->result();
        foreach ($query_health_condition->result() as $row) {
            $health_condition = $row->health_condition;
        }

        $this->db->select('d.*, r.*');
        $this->db->from('recommended_daily_meals d, recommended_recipes r');
        $this->db->where('d.record_date', $record_date);
        $this->db->where('d.recommended_recipe_id = r.recommended_recipe_id');
        $this->db->where('r.health_condition', $health_condition);
        $this->db->order_by('meal_group', 'asc');

        $query = $this->db->get();

        if ($query->num.rows() > 0) {
            return $query->result();
        } else {
            return false;
        }
    }
}

//VIEW RECIPE
function get_recipe_contents($own_recipe_id) {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id = $session_data['user_id'];
        $data['user_id'] = $user_id;

        /*$this->db->select('o.*, i.*');
        $this->db->from('own_recipes o, ingredients i');
        $this->db->where('o.user_id', $user_id);
        $this->db->where('o.own_recipe_id = i.own_recipe_id');
        $query = $this->db->get();
        return $query->result();*/

        /*$this->db->select('*');
        $this->db->from('ingredients');
        $this->db->where("own_recipe_id", $own_recipe_id);
        $query = $this->db->get();
        return $query->result();*/

        $this->db->select('o.*, i.*, r.*');
        $this->db->from('own_recipes o, ingredients i, recipe_images r');
        $this->db->where('i.own_recipe_id', $own_recipe_id);
        $this->db->where('i.own_recipe_id = o.own_recipe_id');
        $this->db->where('r.own_recipe_id = o.own_recipe_id');
        $query = $this->db->get();
        return $query->result();
    }
}
}

```

```

//VIEW RECOMMENDED RECIPE
function get_recommended_recipe_contents($recommended_recipe_id) {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id = $session_data['user_id'];
        $data['user_id'] = $user_id;

        /*$this->db->select('o.*, i.*');
        $this->db->from('own_recipes o, ingredients i');
        $this->db->where('o.user_id', $user_id);
        $this->db->where('o.own_recipe_id = i.own_recipe_id');
        $query = $this->db->get();
        return $query->result();*/

        /*$this->db->select('*');
        $this->db->from('ingredients');
        $this->db->where("own_recipe_id", $own_recipe_id);
        $query = $this->db->get();
        return $query->result();*/

        $this->db->select('r.*, i.*, c.*');
        $this->db->from('recommended_recipes r, ingredients_recommended i,
            recipe_images_recommended c');
        $this->db->where('i.recommended_recipe_id', $recommended_recipe_id);
        $this->db->where('i.recommended_recipe_id = r.recommended_recipe_id');
        $this->db->where('c.recommended_recipe_id = r.recommended_recipe_id');
        $query = $this->db->get();
        return $query->result();
    }
}

//VIEW SHARED RECIPE
function get_shared_recipe_contents($shared_recipe_id) {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id = $session_data['user_id'];
        $data['user_id'] = $user_id;

        /*$this->db->select('o.*, i.*');
        $this->db->from('own_recipes o, ingredients i');
        $this->db->where('o.user_id', $user_id);
        $this->db->where('o.own_recipe_id = i.own_recipe_id');
        $query = $this->db->get();
        return $query->result();*/

        /*$this->db->select('*');
        $this->db->from('ingredients');
        $this->db->where("own_recipe_id", $own_recipe_id);
        $query = $this->db->get();
        return $query->result();*/

        $this->db->select('s.*, o.*, i.*, r.*');
        $this->db->from('shared_recipes s, own_recipes o, ingredients i,
            recipe_images r');
        $this->db->where('s.shared_recipe_id', $shared_recipe_id);
        $this->db->where('s.own_recipe_id = o.own_recipe_id');
        $this->db->where('o.own_recipe_id = i.own_recipe_id');
        $this->db->where('r.own_recipe_id = o.own_recipe_id');
        $query = $this->db->get();
        return $query->result();
    }
}

//DELETE CREATED RECIPE
public function del_created_recipe($own_recipe_id) {
    /*$this->db->where('fitness_info_id', $fitness_info_id);
    $this->db->delete('own_recipes', array('own_recipe_id' => $own_recipe_id));
    $this->db->delete('ingredients', array('own_recipe_id' => $own_recipe_id));
    $this->db->delete('recipe_images', array('own_recipe_id' => $own_recipe_id));
    */
}

//DELETE INGREDIENTS
public function del_ingredient($ingredient_id) {
    $this->db->delete('ingredients', array('ingredient_id' => $ingredient_id));
}

//SHARE CREATED RECIPE
function process_share_recipe($own_recipe_id) {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id = $session_data['user_id'];

        $data['user_id'] = $user_id;
        $data['own_recipe_id'] = $own_recipe_id;

        $this->db->insert('shared_recipes', $data);
    }
    else
        echo "Session Timeout";
}

//ADD RECIPE MODEL
function process_add_recipe($data3) {
    // $health_condition = $this->input->post('health_condition');
    $health_condition = implode(", ", $this->input->post('health_condition'));
    $meal_type = $this->input->post('meal_type');
    $recipe_name = $this->input->post('recipe_name');
    $serving_sugg = $this->input->post('serving_sugg');
    $preparation_procedure = $this->input->post('preparation_procedure');
}

```

```

$calories = $this->input->post('calories');
$carbohydrate = $this->input->post('carbohydrate');
$protein = $this->input->post('protein');
$fat = $this->input->post('fat');

if($this->session->userdata('logged-in')) {
    $session_data = $this->session->userdata('logged-in');
    $user_id = $session_data['user_id'];

    $data['user_id'] = $user_id;
    $data['health_condition'] = $health_condition;
    $data['meal_type'] = $meal_type;
    $data['recipe_name'] = $recipe_name;
    $data['serving_sugg'] = $serving_sugg;
    $data['preparation_procedure'] = $preparation_procedure;
    $data['calories'] = $calories;
    $data['carbohydrate'] = $carbohydrate;
    $data['protein'] = $protein;
    $data['fat'] = $fat;

    $this->db->insert('own_recipes', $data);

    $own_recipe_id = $this->db->insert_id();

    if (isset($_POST['ingredients'], $_POST['food_group'], $_POST['ingredient_measure'], $_POST['ingredient_quantity'])) {
        for ($i=0; $i<count($_POST['ingredients']); $i++) {
            $ingredients = $_POST['ingredients'][$i];
            $food_group = $_POST['food_group'][$i];
            $ingredient_measure = $_POST['ingredient_measure'][$i];
            $ingredient_quantity = $_POST['ingredient_quantity'][$i];
            $data2['own_recipe_id'] = $own_recipe_id;
            $data2['food_name'] = $ingredients;
            $data2['food_group'] = $food_group;
            $data2['measure'] = $ingredient_measure;
            $data2['serving_size'] = $ingredient_quantity;
            /*
            $counts = array_count_values($food_group);
            echo $counts['Vegetable'];
            */
            $this->db->insert('ingredients', $data2);
        }
    }

    /* $file=array(
        'img_name'=>$data['raw_name'],
        'thumb_name'=>$data['raw_name'].'_thumb',
        'ext'=>$data['file_ext'],
        'upload_date'=>time()
    );*/

    /* $data3['raw_name'];
    $data3['raw_name'].'_thumb';
    $data3['file_ext'];
    $data3['upload_date'] = time();*/
    $data3['own_recipe_id'] = $own_recipe_id;

    $this->db->insert('recipe_images', $data3);

}

else
    echo "Session Timeout";
}

//ADD INGREDIENT AUTOCOMplete
//USER SEARCH MODEL/S
function GetAutocomplete($options = array()) {
    $this->db->select('food_group, food_name, weight_edible_portion, energy, carbohydrate, protein, fat');
    $this->db->like('food_name', $options['keyword'], 'after');
    $query = $this->db->get('food_exchange_list');
    return $query->result();
}

//SEARCH FOR MEAL MODEL
function get_search_meal() {
    if($this->session->userdata('logged-in')) {
        $session_data = $this->session->userdata('logged-in');
        $user_id = $session_data['user_id'];
    }
    else
        echo "Session Timeout";

    /* $this->db->select('health_condition');
    $this->db->from('health_conditions');
    $this->db->where("user_id", $user_id);
    $this->db->order_by("record.date", "desc");
    $this->db->limit(1);
    $query_health_condition = $this->db->get();
    $query_health_condition->result();
    foreach ($query_health_condition->result() as $row) {
        $health_condition = $row->health_condition;
    }*/

    $match = $this->input->post('search_meal');
    /* $query = "(own_recipe_id LIKE '%" . $this->db->escape_like_str($match) . "%' OR
    user_id LIKE '%" . $this->db->escape_like_str($match) . "%' OR meal_type LIKE
    '%" . $this->db->escape_like_str($match) . "%'";

```

```

."%" OR recipe_name LIKE '%" . $this->db->escape_like_str($match)."%" OR
serving_sugg LIKE '%" . $this->db->escape_like_str($match)."%" OR
preparation_procedure LIKE '%" . $this->db->escape_like_str($match)."%" OR
."%" OR calories LIKE '%" . $this->db->escape_like_str($match)."%" OR
carbohydrate LIKE '%" . $this->db->escape_like_str($match)."%" OR protein
LIKE '%" . $this->db->escape_like_str($match)."%" OR fat LIKE '%" . $this->db->escape_like_str($match)."%" AND
health_condition = "*/
$this->db->like('own_recipe_id', $match);
$this->db->or_like('user_id', $match);
$this->db->or_like('meal_type', $match);
$this->db->or_like('recipe_name', $match);
$this->db->or_like('serving_sugg', $match);
$this->db->or_like('preparation_procedure', $match);
$this->db->or_like('calories', $match);
$this->db->or_like('carbohydrate', $match);
$this->db->or_like('protein', $match);
$this->db->or_like('fat', $match);
$this->db->or_like('health_condition', $match);
// $this->db->where($query, $health_condition);
$this->db->order_by('own_recipe_id', 'desc');
$this->db->where('user_id', $user_id);
$query = $this->db->get('own_recipes');
return $query->result();
}

//ADD MEAL
function process_addmeal($own_recipe_id) {
if($this->session->userdata('logged_in')) {
    $session_data = $this->session->userdata('logged_in');
    $user_id = $session_data['user_id'];
    $meal_group = $this->input->post('meal_group');
    $record_date = $this->input->post('record_date');

    $data['user_id'] = $user_id;
    $data['own_recipe_id'] = $own_recipe_id;
    $data['meal_group'] = $meal_group;
    $data['record_date'] = $record_date;

    $this->db->insert('plan_daily_meals', $data);
}
else
    echo "Session Timeout";
}

//DELETE MEAL
public function del_meal($plan_daily_meal_id) {
    // $this->db->where('fitness_info_id', $fitness_info_id);
    $this->db->delete('plan_daily_meals', array('plan_daily_meal_id' =>
        $plan_daily_meal_id));
}

//VIEW DAILY MEALS BY DATE MODEL
public function show_data_by_date($record_date) {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id = $session_data['user_id'];
        // $data['user_id'] = $user_id;
        // $condition = "record_date = " . " " . $record_date . " " ;

        $this->db->select('p.*, o.*');
        $this->db->from('plan_daily_meals p, own_recipes o');
        $this->db->where("p.user_id", $user_id);
        $this->db->where('p.record_date', $record_date);
        $this->db->where('p.own_recipe_id = o.own_recipe_id');
        $this->db->order_by('meal_group', 'asc');

        $query1 = $this->db->get();

        $this->db->select('f.*, h.*');
        $this->db->from('fitness_profile f, health_conditions h');
        $this->db->where('f.user_id', $user_id);
        $this->db->where('f.fitness_info_id = h.fitness_info_id');
        // $this->db->where('f.record_date = h.record_date');
        $this->db->order_by("f.record_date", "desc");
        $this->db->limit(1);
        $query2 = $this->db->get();

        $result1 = $query1->result();
        $result2 = $query2->result();

        if ($query1->num_rows() > 0 && $query2->num_rows() > 0) {
            return array_merge($result1, $result2);
        }
        else {
            return false;
        }
    }

    /* if ($query->num_rows() > 0) {
        return $query->result();
    }
    else {
        return false;
    }
    */
}

}

public function show_calorie_limit() {
    if($this->session->userdata('logged_in')) {
        $session_data = $this->session->userdata('logged_in');
        $user_id = $session_data['user_id'];

```

```

        $data['user_id'] = $user_id;

        $this->db->select('*');
        $this->db->from('fitness_profile');
        $this->db->where('user_id', $user_id);
        $this->db->order_by("record_date", "desc");
        $this->db->limit(1);
        $query = $this->db->get();
        if ($query->num_rows() > 0) {
            return $query->result();
        } else {
            return false;
        }
    }
}

?>

```

---

```

<?php

Class Login_Model extends CI_Model {
    function login($username, $password) {
        $query = $this->db->get_where('users', array('username'=>$username, '
            password'=>MD5($password)));

        if($query->num_rows() == 1) {
            return $query->result();
        }

        else {
            return false;
        }
    }

    function process(){
        $username = $this->input->post('username_reg');
        $password = $this->input->post('password_reg');
        $confirm_password = $this->input->post('
            confirm_password_reg');
        $first_name = $this->input->post('first_name');
        $middle_name = $this->input->post('middle_name');
        $last_name = $this->input->post('last_name');
        $gender = $this->input->post('genderRadio');
        $user_type = $this->input->post('userTypeRadio');
        $email_address = $this->input->post('email_address');
        $birthday = $this->input->post('birthdate');

        $data = array(
            'username' => $username,
            'password' => MD5($password),
            'confirm_password' => MD5($confirm_password),
            'first_name' => $first_name,
            'middle_name' => $middle_name,
            'last_name' => $last_name,
            'gender' => $gender,
            'user_type' => $user_type,
            'email_address' => $email_address,
            'birthday' => $birthday
        );
        $this->db->insert('users', $data);

    }

    public function check_user_exist($usr) {
        $this->db->where("username", $usr);
        $query=$this->db->get("users");
        if($query->num_rows()>0) {
            return true;
        }
        else {
            return false;
        }
    }

    public function check_email_exist($eml) {
        $this->db->where("email_address", $eml);
        $query=$this->db->get("users");
        if($query->num_rows()>0) {
            return true;
        }
        else {
            return false;
        }
    }
}

?>

```

---

## B..4 Libraries

```

<?php
/**
 * Copyright 2011 Facebook, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may
 * not use this file except in compliance with the License. You may obtain
 * a copy of the License at

```

```

*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/

if (!function_exists('curl_init')) {
    throw new Exception('Facebook needs the CURL PHP extension.');
```

```

}
if (!function_exists('json_decode')) {
    throw new Exception('Facebook needs the JSON PHP extension.');
```

```

}

/**
 * Thrown when an API call returns an exception.
 *
 * @author Naitik Shah <naitik@facebook.com>
 */
class FacebookApiException extends Exception
{
    /**
     * The result from the API server that represents the exception information.
     *
     * @var mixed
     */
    protected $result;

    /**
     * Make a new API Exception with the given result.
     *
     * @param array $result The result from the API server
     */
    public function __construct($result) {
        $this->result = $result;

        $code = 0;
        if (isset($result['error_code']) && is_int($result['error_code'])) {
            $code = $result['error_code'];
        }

        if (isset($result['error_description'])) {
            // OAuth 2.0 Draft 10 style
            $msg = $result['error_description'];
        } else if (isset($result['error']) && is_array($result['error'])) {
            // OAuth 2.0 Draft 00 style
            $msg = $result['error']['message'];
        } else if (isset($result['error_msg'])) {
            // Rest server style
            $msg = $result['error_msg'];
        } else {
            $msg = 'Unknown Error. Check getResult()';
        }

        parent::__construct($msg, $code);
    }

    /**
     * Return the associated result object returned by the API server.
     *
     * @return array The result from the API server
     */
    public function getResult() {
        return $this->result;
    }

    /**
     * Returns the associated type for the error. This will default to
     * 'Exception' when a type is not available.
     *
     * @return string
     */
    public function getType() {
        if (isset($this->result['error'])) {
            $error = $this->result['error'];
            if (is_string($error)) {
                // OAuth 2.0 Draft 10 style
                return $error;
            } else if (is_array($error)) {
                // OAuth 2.0 Draft 00 style
                if (isset($error['type'])) {
                    return $error['type'];
                }
            }
        }

        return 'Exception';
    }

    /**
     * To make debugging easier.
     *
     * @return string The string representation of the error
     */
    public function __toString() {
        $str = $this->getType() . ': ';
    }
}

```

```

        if ($this->code != 0) {
            $sstr .= $this->code . ': ';
        }
        return $sstr . $this->message;
    }
}

/**
 * Provides access to the Facebook Platform. This class provides
 * a majority of the functionality needed, but the class is abstract
 * because it is designed to be sub-classed. The subclass must
 * implement the four abstract methods listed at the bottom of
 * the file.
 *
 * @author Naitik Shah <naitik@facebook.com>
 */
abstract class BaseFacebook
{
    /**
     * Version.
     */
    const VERSION = '3.2.3';

    /**
     * Signed Request Algorithm.
     */
    const SIGNED_REQUEST_ALGORITHM = 'HMAC-SHA256';

    /**
     * Default options for curl.
     *
     * @var array
     */
    public static $CURL_OPTS = array(
        CURLOPT_CONNECTTIMEOUT => 10,
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_TIMEOUT => 60,
        CURLOPT_USERAGENT => 'facebook-php-3.2',
    );

    /**
     * List of query parameters that get automatically dropped when rebuilding
     * the current URL.
     *
     * @var array
     */
    protected static $DROP_QUERY_PARAMS = array(
        'code',
        'state',
        'signed_request',
    );

    /**
     * Maps aliases to Facebook domains.
     *
     * @var array
     */
    public static $DOMAIN_MAP = array(
        'api' => 'https://api.facebook.com/',
        'api_video' => 'https://api-video.facebook.com/',
        'api_read' => 'https://api-read.facebook.com/',
        'graph' => 'https://graph.facebook.com/',
        'graph_video' => 'https://graph-video.facebook.com/',
        'www' => 'https://www.facebook.com/',
    );

    /**
     * The Application ID.
     *
     * @var string
     */
    protected $appId;

    /**
     * The Application App Secret.
     *
     * @var string
     */
    protected $appSecret;

    /**
     * The ID of the Facebook user, or 0 if the user is logged out.
     *
     * @var integer
     */
    protected $user;

    /**
     * The data from the signed_request token.
     *
     * @var string
     */
    protected $signedRequest;

    /**
     * A CSRF state variable to assist in the defense against CSRF attacks.
     *
     * @var string
     */

```

```

protected $state;

/**
 * The OAuth access token received in exchange for a valid authorization
 * code. null means the access token has yet to be determined.
 *
 * @var string
 */
protected $accessToken = null;

/**
 * Indicates if the CURL based @ syntax for file uploads is enabled.
 *
 * @var boolean
 */
protected $fileUploadSupport = false;

/**
 * Indicates if we trust HTTP_X_FORWARDED_* headers.
 *
 * @var boolean
 */
protected $trustForwarded = false;

/**
 * Indicates if signed_request is allowed in query parameters.
 *
 * @var boolean
 */
protected $allowSignedRequest = true;

/**
 * Initialize a Facebook Application.
 *
 * The configuration:
 * - appId: the application ID
 * - secret: the application secret
 * - fileUpload: (optional) boolean indicating if file uploads are enabled
 * - allowSignedRequest: (optional) boolean indicating if signed_request is
 *                       allowed in query parameters or POST body. Should be
 *                       false for non-canvas apps. Defaults to true.
 *
 * @param array $config The application configuration
 */
public function __construct($config) {
    $this->setAppId($config['appId']);
    $this->setAppSecret($config['secret']);
    if (isset($config['fileUpload'])) {
        $this->setFileUploadSupport($config['fileUpload']);
    }
    if (isset($config['trustForwarded']) && $config['trustForwarded']) {
        $this->trustForwarded = true;
    }
    if (isset($config['allowSignedRequest'])
        && !$config['allowSignedRequest']) {
        $this->allowSignedRequest = false;
    }
    $state = $this->getPersistentData('state');
    if (!empty($state)) {
        $this->state = $state;
    }
}

/**
 * Set the Application ID.
 *
 * @param string $appId The Application ID
 *
 * @return BaseFacebook
 */
public function setAppId($appId) {
    $this->appId = $appId;
    return $this;
}

/**
 * Get the Application ID.
 *
 * @return string the Application ID
 */
public function getAppId() {
    return $this->appId;
}

/**
 * Set the App Secret.
 *
 * @param string $apiSecret The App Secret
 *
 * @return BaseFacebook
 * @deprecated Use setAppSecret instead.
 * @see setAppSecret()
 */
public function setApiSecret($apiSecret) {
    $this->setAppSecret($apiSecret);
    return $this;
}
}

/**

```



```

* Set the App Secret.
*
* @param string $appSecret The App Secret
*
* @return BaseFacebook
*/
public function setAppSecret($appSecret) {
    $this->appSecret = $appSecret;
    return $this;
}

/**
* Get the App Secret.
*
* @return string the App Secret
*
* @deprecated Use getAppSecret instead.
* @see getAppSecret()
*/
public function getApiSecret() {
    return $this->getAppSecret();
}

/**
* Get the App Secret.
*
* @return string the App Secret
*/
public function getAppSecret() {
    return $this->appSecret;
}

/**
* Set the file upload support status.
*
* @param boolean $fileUploadSupport The file upload support status.
*
* @return BaseFacebook
*/
public function setFileUploadSupport($fileUploadSupport) {
    $this->fileUploadSupport = $fileUploadSupport;
    return $this;
}

/**
* Get the file upload support status.
*
* @return boolean true if and only if the server supports file upload.
*/
public function getFileUploadSupport() {
    return $this->fileUploadSupport;
}

/**
* Get the file upload support status.
*
* @return boolean true if and only if the server supports file upload.
*
* @deprecated Use getFileUploadSupport instead.
* @see getFileUploadSupport()
*/
public function useFileUploadSupport() {
    return $this->getFileUploadSupport();
}

/**
* Sets the access token for api calls. Use this if you get
* your access token by other means and just want the SDK
* to use it.
*
* @param string $access_token an access token.
*
* @return BaseFacebook
*/
public function setAccessToken($access_token) {
    $this->accessToken = $access_token;
    return $this;
}

/**
* Extend an access token, while removing the short-lived token that might
* have been generated via client-side flow. Thanks to http://bit.ly/b0Pt0H
* for the workaround.
*/
public function setExtendedAccessToken() {
    try {
        // need to circumvent json_decode by calling _oauthRequest
        // directly, since response isn't JSON format.
        $access_token_response = $this->_oauthRequest(
            $this->getUrl('graph', '/oauth/access-token'),
            $params = array(
                'client_id' => $this->getAppId(),
                'client_secret' => $this->getAppSecret(),
                'grant_type' => 'fb_exchange_token',
                'fb_exchange_token' => $this->getAccessToken(),
            )
        );
    } catch (FacebookApiException $e) {

```

```

    // most likely that user very recently revoked authorization.
    // In any event, we don't have an access token, so say so.
    return false;
}

if (empty($access_token_response)) {
    return false;
}

$response_params = array();
parse_str($access_token_response, $response_params);

if (!isset($response_params['access_token'])) {
    return false;
}

$this->destroySession();

$this->setPersistentData(
    'access_token', $response_params['access_token']
);
}

/**
 * Determines the access token that should be used for API calls.
 * The first time this is called, $this->accessToken is set equal
 * to either a valid user access token, or it's set to the application
 * access token if a valid user access token wasn't available. Subsequent
 * calls return whatever the first call returned.
 *
 * @return string The access token
 */
public function getAccessToken() {
    if ($this->accessToken !== null) {
        // we've done this already and cached it. Just return.
        return $this->accessToken;
    }

    // first establish access token to be the application
    // access token, in case we navigate to the /oauth/access_token
    // endpoint, where SOME access token is required.
    $this->setAccessToken($this->getApplicationAccessToken());
    $user_access_token = $this->getUserAccessToken();
    if ($user_access_token) {
        $this->setAccessToken($user_access_token);
    }

    return $this->accessToken;
}

/**
 * Determines and returns the user access token, first using
 * the signed request if present, and then falling back on
 * the authorization code if present. The intent is to
 * return a valid user access token, or false if one is determined
 * to not be available.
 *
 * @return string A valid user access token, or false if one
 * could not be determined.
 */
protected function getUserAccessToken() {
    // first, consider a signed request if it's supplied.
    // if there is a signed request, then it alone determines
    // the access token.
    $signed_request = $this->getSignedRequest();
    if ($signed_request) {
        // apps.facebook.com hands the access_token in the signed_request
        if (array_key_exists('oauth_token', $signed_request)) {
            $access_token = $signed_request['oauth_token'];
            $this->setPersistentData('access_token', $access_token);
            return $access_token;
        }

        // the JS SDK puts a code in with the redirect_uri of ''
        if (array_key_exists('code', $signed_request)) {
            $code = $signed_request['code'];
            if ($code && $code == $this->getPersistentData('code')) {
                // short-circuit if the code we have is the same as the one presented
                return $this->getPersistentData('access_token');
            }

            $access_token = $this->getAccessTokenFromCode($code, '');
            if ($access_token) {
                $this->setPersistentData('code', $code);
                $this->setPersistentData('access_token', $access_token);
                return $access_token;
            }
        }

        // signed request states there's no access token, so anything
        // stored should be cleared.
        $this->clearAllPersistentData();
        return false; // respect the signed request's data, even
        // if there's an authorization code or something else
    }

    $code = $this->getCode();
    if ($code && $code != $this->getPersistentData('code')) {
        $access_token = $this->getAccessTokenFromCode($code);
    }
}

```

```

    if ($access_token) {
        $this->setPersistentData('code', $code);
        $this->setPersistentData('access_token', $access_token);
        return $access_token;
    }

    // code was bogus, so everything based on it should be invalidated.
    $this->clearAllPersistentData();
    return false;
}

// as a fallback, just return whatever is in the persistent
// store, knowing nothing explicit (signed request, authorization
// code, etc.) was present to shadow it (or we saw a code in $_REQUEST,
// but it's the same as what's in the persistent store)
return $this->getPersistentData('access_token');
}

/**
 * Retrieve the signed request, either from a request parameter or,
 * if not present, from a cookie.
 *
 * @return string the signed request, if available, or null otherwise.
 */
public function getSignedRequest() {
    if (!$this->signedRequest) {
        if ($this->allowSignedRequest && !empty($_REQUEST['signed_request'])) {
            $this->signedRequest = $this->parseSignedRequest(
                $_REQUEST['signed_request']
            );
        } else if (!empty($_COOKIE[$this->getSignedRequestCookieName()])) {
            $this->signedRequest = $this->parseSignedRequest(
                $_COOKIE[$this->getSignedRequestCookieName()]);
        }
    }
    return $this->signedRequest;
}

/**
 * Get the UID of the connected user, or 0
 * if the Facebook user is not connected.
 *
 * @return string the UID if available.
 */
public function getUser() {
    if ($this->user !== null) {
        // we've already determined this and cached the value.
        return $this->user;
    }

    return $this->user = $this->getUserFromAvailableData();
}

/**
 * Determines the connected user by first examining any signed
 * requests, then considering an authorization code, and then
 * falling back to any persistent store storing the user.
 *
 * @return integer The id of the connected Facebook user,
 * or 0 if no such user exists.
 */
protected function getUserFromAvailableData() {
    // if a signed request is supplied, then it solely determines
    // who the user is.
    $signed_request = $this->getSignedRequest();
    if ($signed_request) {
        if (array_key_exists('user_id', $signed_request)) {
            $user = $signed_request['user_id'];

            if ($user != $this->getPersistentData('user_id')){
                $this->clearAllPersistentData();
            }

            $this->setPersistentData('user_id', $signed_request['user_id']);
            return $user;
        }

        // if the signed request didn't present a user id, then invalidate
        // all entries in any persistent store.
        $this->clearAllPersistentData();
        return 0;
    }

    $user = $this->getPersistentData('user_id', $default = 0);
    $persisted_access_token = $this->getPersistentData('access_token');

    // use access_token to fetch user id if we have a user access_token, or if
    // the cached access token has changed.
    $access_token = $this->getAccessToken();
    if ($access_token &&
        $access_token != $this->getApplicationAccessToken() &&
        !($user && $persisted_access_token == $access_token)) {
        $user = $this->getUserFromAccessToken();
        if ($user) {
            $this->setPersistentData('user_id', $user);
        } else {
            $this->clearAllPersistentData();
        }
    }
}

```

```

    return $user;
}

/**
 * Get a Login URL for use with redirects. By default, full page redirect is
 * assumed. If you are using the generated URL with a window.open() call in
 * JavaScript, you can pass in display=popup as part of the $params.
 *
 * The parameters:
 * - redirect_uri: the url to go to after a successful login
 * - scope: comma separated list of requested extended perms
 *
 * @param array $params Provide custom parameters
 * @return string The URL for the login flow
 */
public function getLoginUrl($params=array()) {
    $this->establishCSRFTokenState();
    $currentUrl = $this->getCurrentUrl();

    // if 'scope' is passed as an array, convert to comma separated list
    $scopeParams = isset($params['scope']) ? $params['scope'] : null;
    if ($scopeParams && is_array($scopeParams)) {
        $params['scope'] = implode(',', $scopeParams);
    }

    return $this->getUrl(
        'www',
        'dialog/oauth',
        array_merge(
            array(
                'client_id' => $this->getAppId(),
                'redirect_uri' => $currentUrl, // possibly overwritten
                'state' => $this->state,
                'sdk' => 'php-sdk-' . self::VERSION
            ),
            $params
        )
    );
}

/**
 * Get a Logout URL suitable for use with redirects.
 *
 * The parameters:
 * - next: the url to go to after a successful logout
 *
 * @param array $params Provide custom parameters
 * @return string The URL for the logout flow
 */
public function getLogoutUrl($params=array()) {
    return $this->getUrl(
        'www',
        'logout.php',
        array_merge(array(
            'next' => $this->getCurrentUrl(),
            'access_token' => $this->getUserAccessToken(),
        ), $params)
    );
}

/**
 * Get a login status URL to fetch the status from Facebook.
 *
 * @param array $params Provide custom parameters
 * @return string The URL for the logout flow
 */
public function getLoginStatusUrl($params=array()) {
    return $this->getLoginUrl(
        array_merge(array(
            'response_type' => 'code',
            'display' => 'none',
        ), $params)
    );
}

/**
 * Make an API call.
 *
 * @return mixed The decoded response
 */
public function api(/* polymorphic */) {
    $args = func_get_args();
    if (is_array($args[0])) {
        return $this->restserver($args[0]);
    } else {
        return call_user_func_array(array($this, '_graph'), $args);
    }
}

/**
 * Constructs and returns the name of the cookie that
 * potentially houses the signed request for the app user.
 * The cookie is not set by the BaseFacebook class, but
 * it may be set by the JavaScript SDK.
 *
 * @return string the name of the cookie that would house
 * the signed request value.
 */
protected function getSignedRequestCookieName() {

```

```

    return 'fbsr_'. $this->getAppId();
}

/**
 * Constructs and returns the name of the cookie that potentially contain
 * metadata. The cookie is not set by the BaseFacebook class, but it may be
 * set by the JavaScript SDK.
 *
 * @return string the name of the cookie that would house metadata.
 */
protected function getMetadataCookieName() {
    return 'fbm_'. $this->getAppId();
}

/**
 * Get the authorization code from the query parameters, if it exists,
 * and otherwise return false to signal no authorization code was
 * discoverable.
 *
 * @return mixed The authorization code, or false if the authorization
 * code could not be determined.
 */
protected function getCode() {
    if (!isset($_REQUEST['code']) || !isset($_REQUEST['state'])) {
        return false;
    }
    if ($this->state === $_REQUEST['state']) {
        // CSRF state has done its job, so clear it
        $this->state = null;
        $this->clearPersistentData('state');
        return $_REQUEST['code'];
    }
    self::errorLog('CSRF state token does not match one provided.');
```

```

    return false;
}

/**
 * Retrieves the UID with the understanding that
 * $this->accessToken has already been set and is
 * seemingly legitimate. It relies on Facebook's Graph API
 * to retrieve user information and then extract
 * the user ID.
 *
 * @return integer Returns the UID of the Facebook user, or 0
 * if the Facebook user could not be determined.
 */
protected function getUserFromAccessToken() {
    try {
        $user_info = $this->api('/me');
        return $user_info['id'];
    } catch (FacebookApiException $e) {
        return 0;
    }
}

/**
 * Returns the access token that should be used for logged out
 * users when no authorization code is available.
 *
 * @return string The application access token, useful for gathering
 * public information about users and applications.
 */
public function getApplicationAccessToken() {
    return $this->appId.'|'. $this->appSecret;
}

/**
 * Lays down a CSRF state token for this process.
 *
 * @return void
 */
protected function establishCSRFTokenState() {
    if ($this->state === null) {
        $this->state = md5(uniqid(mt_rand(), true));
        $this->setPersistentData('state', $this->state);
    }
}

/**
 * Retrieves an access token for the given authorization code
 * (previously generated from www.facebook.com on behalf of
 * a specific user). The authorization code is sent to graph.facebook.com
 * and a legitimate access token is generated provided the access token
 * and the user for which it was generated all match, and the user is
 * either logged in to Facebook or has granted an offline access permission.
 *
 * @param string $code An authorization code.
 * @param string $redirect_uri Optional redirect URI. Default null
 *
 * @return mixed An access token exchanged for the authorization code, or
 * false if an access token could not be generated.
 */
protected function getAccessTokenFromCode($code, $redirect_uri = null) {
    if (empty($code)) {
        return false;
    }

    if ($redirect_uri === null) {
```

```

    $redirect_uri = $this->getCurrentUrl();
}

try {
    // need to circumvent json_decode by calling _oauthRequest
    // directly, since response isn't JSON format.
    $access_token_response =
        $this->_oauthRequest(
            $this->getUrl('graph', '/oauth/access_token'),
            $params = array('client_id' => $this->getAppId(),
                           'client_secret' => $this->getAppSecret(),
                           'redirect_uri' => $redirect_uri,
                           'code' => $code));
} catch (FacebookApiException $e) {
    // most likely that user very recently revoked authorization.
    // In any event, we don't have an access token, so say so.
    return false;
}

if (empty($access_token_response)) {
    return false;
}

$response_params = array();
parse_str($access_token_response, $response_params);
if (!isset($response_params['access_token'])) {
    return false;
}

return $response_params['access_token'];
}

/**
 * Invoke the old restserver.php endpoint.
 *
 * @param array $params Method call object
 *
 * @return mixed The decoded response object
 * @throws FacebookApiException
 */
protected function _restserver($params) {
    // generic application level parameters
    $params['api_key'] = $this->getAppId();
    $params['format'] = 'json-strings';

    $result = json_decode($this->_oauthRequest(
        $this->getApiUrl($params['method']),
        $params
    ), true);

    // results are returned, errors are thrown
    if (is_array($result) && isset($result['error_code'])) {
        $this->throwApiException($result);
        // @codeCoverageIgnoreStart
    }
    // @codeCoverageIgnoreEnd

    $method = strtolower($params['method']);
    if ($method === 'auth.expiresession' ||
        $method === 'auth.revokeauthorization') {
        $this->destroySession();
    }

    return $result;
}

/**
 * Return true if this is video post.
 *
 * @param string $path The path
 * @param string $method The http method (default 'GET')
 *
 * @return boolean true if this is video post
 */
protected function isVideoPost($path, $method = 'GET') {
    if ($method === 'POST' && preg_match("/^(\/)(.+)(\/)(videos)$/", $path)) {
        return true;
    }
    return false;
}

/**
 * Invoke the Graph API.
 *
 * @param string $path The path (required)
 * @param string $method The http method (default 'GET')
 * @param array $params The query/post data
 *
 * @return mixed The decoded response object
 * @throws FacebookApiException
 */
protected function _graph($path, $method = 'GET', $params = array()) {
    if (is_array($method) && empty($params)) {
        $params = $method;
        $method = 'GET';
    }
    $params['method'] = $method; // method override as we always do a POST

    if ($this->isVideoPost($path, $method)) {

```

```

        $domainKey = 'graph_video';
    } else {
        $domainKey = 'graph';
    }

    $result = json_decode($this->_oauthRequest(
        $this->getUrl($domainKey, $path),
        $params
    ), true);

    // results are returned, errors are thrown
    if (is_array($result) && isset($result['error'])) {
        $this->throwAPIException($result);
        // @codeCoverageIgnoreStart
    }
    // @codeCoverageIgnoreEnd

    return $result;
}

/**
 * Make a OAuth Request.
 *
 * @param string $url The path (required)
 * @param array $params The query/post data
 *
 * @return string The decoded response object
 * @throws FacebookApiException
 */
protected function _oauthRequest($url, $params) {
    if (!isset($params['access_token'])) {
        $params['access_token'] = $this->getAccessToken();
    }

    if (isset($params['access_token']) && !isset($params['appsecret_proof'])) {
        $params['appsecret_proof'] = $this->getAppSecretProof($params['access_token']);
    }

    // json_encode all params values that are not strings
    foreach ($params as $key => $value) {
        if (!is_string($value) && !($value instanceof CURLFile)) {
            $params[$key] = json_encode($value);
        }
    }

    return $this->makeRequest($url, $params);
}

/**
 * Generate a proof of App Secret
 * This is required for all API calls originating from a server
 * It is a sha256 hash of the access_token made using the app secret
 *
 * @param string $access_token The access_token to be hashed (required)
 *
 * @return string The sha256 hash of the access_token
 */
protected function getAppSecretProof($access_token) {
    return hash_hmac('sha256', $access_token, $this->getAppSecret());
}

/**
 * Makes an HTTP request. This method can be overridden by subclasses if
 * developers want to do fancier things or use something other than curl to
 * make the request.
 *
 * @param string $url The URL to make the request to
 * @param array $params The parameters to use for the POST body
 * @param CurlHandler $ch Initialized curl handle
 *
 * @return string The response text
 */
protected function makeRequest($url, $params, $ch=null) {
    if (!$ch) {
        $ch = curl_init();
    }

    $sopts = self::$CURL_OPTS;
    if ($this->getFileUploadSupport()) {
        $sopts[CURLOPT_POSTFIELDS] = $params;
    } else {
        $sopts[CURLOPT_POSTFIELDS] = http_build_query($params, null, '&');
    }
    $sopts[CURLOPT_URL] = $url;

    // disable the 'Expect: 100-continue' behaviour. This causes CURL to wait
    // for 2 seconds if the server does not support this header.
    if (isset($sopts[CURLOPT_HTTPHEADER])) {
        $existing_headers = $sopts[CURLOPT_HTTPHEADER];
        $existing_headers[] = 'Expect:';
        $sopts[CURLOPT_HTTPHEADER] = $existing_headers;
    } else {
        $sopts[CURLOPT_HTTPHEADER] = array('Expect:');
    }

    curl_setopt_array($ch, $sopts);
    $result = curl_exec($ch);

    $errno = curl_errno($ch);

```

```

// CURLOPT_SSL_CACERT || CURLOPT_SSL_CACERT_BADFILE
if ($?errno == 60 || $errno == 77) {
    self::errorLog('Invalid or no certificate authority found, '.
        'using bundled information');
    curl_setopt($ch, CURLOPT_CAINFO,
        dirname(__FILE__) . DIRECTORY_SEPARATOR . 'fb_ca_chain_bundle.crt');
    $result = curl_exec($ch);
}

// With dual stacked DNS responses, it's possible for a server to
// have IPv6 enabled but not have IPv6 connectivity. If this is
// the case, curl will try IPv4 first and if that fails, then it will
// fall back to IPv6 and the error EHOSTUNREACH is returned by the
// operating system.
if ($result == false && empty($opts[CURLOPT_IPRESOLVE])) {
    $matches = array();
    $regex = '/Failed to connect to ([^:]*): Network is unreachable/';
    if (preg_match($regex, curl_error($ch), $matches)) {
        if (strlen(@inet_pton($matches[1])) == 16) {
            self::errorLog('Invalid IPv6 configuration on server, '.
                'Please disable or get native IPv6 on your server.');
            self::$CURL_OPTS[CURLOPT_IPRESOLVE] = CURL_IPRESOLVE_V4;
            curl_setopt($ch, CURLOPT_IPRESOLVE, CURL_IPRESOLVE_V4);
            $result = curl_exec($ch);
        }
    }
}

if ($result == false) {
    $e = new FacebookApiException(array(
        'error_code' => curl_errno($ch),
        'error' => array(
            'message' => curl_error($ch),
            'type' => 'CurlException',
        ),
    ));
    curl_close($ch);
    throw $e;
}
curl_close($ch);
return $result;
}

/**
 * Parses a signed_request and validates the signature.
 *
 * @param string $signed_request A signed token
 *
 * @return array The payload inside it or null if the sig is wrong
 */
protected function parseSignedRequest($signed_request) {
    if (!$signed_request || strpos($signed_request, '.') == false) {
        self::errorLog('Signed request was invalid!');
        return null;
    }

    list($encoded_sig, $payload) = explode('.', $signed_request, 2);

    // decode the data
    $sig = self::base64UrlDecode($encoded_sig);
    $data = json_decode(self::base64UrlDecode($payload), true);

    if (!isset($data['algorithm'])
        || strtoupper($data['algorithm']) != self::SIGNED_REQUEST_ALGORITHM)
    {
        self::errorLog(
            'Unknown algorithm. Expected ' . self::SIGNED_REQUEST_ALGORITHM);
        return null;
    }

    // check sig
    $expected_sig = hash_hmac('sha256', $payload,
        $this->getAppSecret(), $raw = true);

    if (strlen($expected_sig) != strlen($sig)) {
        self::errorLog('Bad Signed JSON signature!');
        return null;
    }

    $result = 0;
    for ($i = 0; $i < strlen($expected_sig); $i++) {
        $result |= ord($expected_sig[$i]) ^ ord($sig[$i]);
    }

    if ($result == 0) {
        return $data;
    } else {
        self::errorLog('Bad Signed JSON signature!');
        return null;
    }
}

/**
 * Makes a signed_request blob using the given data.
 *
 * @param array $data The data array.
 *
 * @return string The signed request.

```



```

*/
protected function makeSignedRequest($data) {
    if (!is_array($data)) {
        throw new InvalidArgumentException(
            'makeSignedRequest expects an array. Got: ' . print_r($data, true));
    }
    $data['algorithm'] = self::SIGNED.REQUEST.ALGORITHM;
    $data['issued_at'] = time();
    $json = json_encode($data);
    $b64 = self::base64UrlEncode($json);
    $raw_sig = hash_hmac('sha256', $b64, $this->getAppSecret(), $raw = true);
    $sig = self::base64UrlEncode($raw_sig);
    return $sig.'.'. $b64;
}

/**
 * Build the URL for api given parameters.
 *
 * @param string $method The method name.
 *
 * @return string The URL for the given parameters
 */
protected function getApiUrl($method) {
    static $READ_ONLY_CALLS =
        array('admin.getallocation' => 1,
            'admin.getappproperties' => 1,
            'admin.getbannedusers' => 1,
            'admin.getlivestreamvialink' => 1,
            'admin.getmetrics' => 1,
            'admin.getrestrictioninfo' => 1,
            'application.getpublicinfo' => 1,
            'auth.getapppublickey' => 1,
            'auth.getsession' => 1,
            'auth.getsignedpublicsessiondata' => 1,
            'comments.get' => 1,
            'connect.getunconnectedfriendscount' => 1,
            'dashboard.getactivity' => 1,
            'dashboard.getcount' => 1,
            'dashboard.getglobalnews' => 1,
            'dashboard.getnews' => 1,
            'dashboard.multigetcount' => 1,
            'dashboard.multigetnews' => 1,
            'data.getcookies' => 1,
            'events.get' => 1,
            'events.getmembers' => 1,
            'fbml.getcustomtags' => 1,
            'feed.getappfriendstories' => 1,
            'feed.getregisteredtemplatebundlebyid' => 1,
            'feed.getregisteredtemplatebundles' => 1,
            'fql.multiquery' => 1,
            'fql.query' => 1,
            'friends.arefriends' => 1,
            'friends.get' => 1,
            'friends.getappusers' => 1,
            'friends.getlists' => 1,
            'friends.getmutualfriends' => 1,
            'gifts.get' => 1,
            'groups.get' => 1,
            'groups.getmembers' => 1,
            'intl.gettranslations' => 1,
            'links.get' => 1,
            'notes.get' => 1,
            'notifications.get' => 1,
            'pages.getinfo' => 1,
            'pages.isadmin' => 1,
            'pages.isappadded' => 1,
            'pages.isfan' => 1,
            'permissions.checkavailableapiaccess' => 1,
            'permissions.checkgrantedapiaccess' => 1,
            'photos.get' => 1,
            'photos.getalbums' => 1,
            'photos.gettags' => 1,
            'profile.getinfo' => 1,
            'profile.getinfooptions' => 1,
            'stream.get' => 1,
            'stream.getcomments' => 1,
            'stream.getfilters' => 1,
            'users.getinfo' => 1,
            'users.getloggedinuser' => 1,
            'users.getstandardinfo' => 1,
            'users.hasapppermission' => 1,
            'users.isappuser' => 1,
            'users.isverified' => 1,
            'video.getuploadlimits' => 1);

    $name = 'api';
    if (isset($READ_ONLY_CALLS[strtolower($method)])) {
        $name = 'api-read';
    } else if (strtolower($method) == 'video.upload') {
        $name = 'api-video';
    }
    return self::getUrl($name, 'restserver.php');
}

/**
 * Build the URL for given domain alias, path and parameters.
 *
 * @param string $name The name of the domain
 * @param string $path Optional path (without a leading slash)
 * @param array $params Optional query parameters

```

```

*
* @return string The URL for the given parameters
*/
protected function getUrl($name, $path='', $params=array()) {
    $url = self::$DOMAIN_MAP[$name];
    if ($path) {
        if ($path[0] === '/') {
            $path = substr($path, 1);
        }
        $url .= $path;
    }
    if ($params) {
        $url .= '?' . http_build_query($params, null, '&');
    }
    return $url;
}

/**
 * Returns the HTTP Host
 *
 * @return string The HTTP Host
 */
protected function getHttpHost() {
    if ($this->trustForwarded && isset($_SERVER['HTTP_X_FORWARDED_HOST'])) {
        $forwardProxies = explode(',', $_SERVER['HTTP_X_FORWARDED_HOST']);
        if (!empty($forwardProxies)) {
            return $forwardProxies[0];
        }
    }
    return $_SERVER['HTTP_HOST'];
}

/**
 * Returns the HTTP Protocol
 *
 * @return string The HTTP Protocol
 */
protected function getHttpProtocol() {
    if ($this->trustForwarded && isset($_SERVER['HTTP_X_FORWARDED_PROTO'])) {
        if ($_SERVER['HTTP_X_FORWARDED_PROTO'] === 'https') {
            return 'https';
        }
        return 'http';
    }
    /*apache + variants specific way of checking for https*/
    if (isset($_SERVER['HTTPS']) &&
        ($_SERVER['HTTPS'] === 'on' || $_SERVER['HTTPS'] == 1)) {
        return 'https';
    }
    /*nginx way of checking for https*/
    if (isset($_SERVER['SERVER_PORT']) &&
        ($_SERVER['SERVER_PORT'] === '443')) {
        return 'https';
    }
    return 'http';
}

/**
 * Returns the base domain used for the cookie.
 *
 * @return string The base domain
 */
protected function getBaseDomain() {
    // The base domain is stored in the metadata cookie if not we fallback
    // to the current hostname
    $metadata = $this->getMetadataCookie();
    if (array_key_exists('base_domain', $metadata) &&
        !empty($metadata['base_domain'])) {
        return trim($metadata['base_domain'], '.');
    }
    return $this->getHttpHost();
}

/**
 * Returns the Current URL, stripping it of known FB parameters that should
 * not persist.
 *
 * @return string The current URL
 */
protected function getCurrentUrl() {
    $protocol = $this->getHttpProtocol() . '://';
    $host = $this->getHttpHost();
    $currentUrl = $protocol.$host.$_SERVER['REQUEST_URI'];
    $parts = parse_url($currentUrl);

    $query = '';
    if (!empty($parts['query'])) {
        // drop known fb params
        $params = explode('&', $parts['query']);
        $retained_params = array();
        foreach ($params as $param) {
            if ($this->shouldRetainParam($param)) {
                $retained_params[] = $param;
            }
        }
    }

    if (!empty($retained_params)) {
        $query = '?' . implode($retained_params, '&');
    }
}

```

```

    }
}

// use port if non default
$port =
isset($parts['port']) &&
((($protocol == 'http://' && $parts['port'] != 80) ||
($protocol == 'https://' && $parts['port'] != 443))
? ':' . $parts['port'] : '');

// rebuild
return $protocol . $parts['host'] . $port . $parts['path'] . $query;
}

/**
 * Returns true if and only if the key or key/value pair should
 * be retained as part of the query string. This amounts to
 * a brute-force search of the very small list of Facebook-specific
 * params that should be stripped out.
 *
 * @param string $param A key or key/value pair within a URL's query (e.g.
 * 'foo=a', 'foo=', or 'foo').
 *
 * @return boolean
 */
protected function shouldRetainParam($param) {
    foreach (self::$DROP_QUERY_PARAMS as $drop_query_param) {
        if ($param == $drop_query_param ||
            strpos($param, $drop_query_param.'=') == 0) {
            return false;
        }
    }

    return true;
}

/**
 * Analyzes the supplied result to see if it was thrown
 * because the access token is no longer valid. If that is
 * the case, then we destroy the session.
 *
 * @param array $result A record storing the error message returned
 * by a failed API call.
 */
protected function throwAPIException($result) {
    $e = new FacebookApiException($result);
    switch ($e->getType()) {
        // OAuth 2.0 Draft 00 style
        case 'OAuthException':
            // OAuth 2.0 Draft 10 style
            case 'invalid_token':
            // REST server errors are just Exceptions
            case 'Exception':
                $message = $e->getMessage();
                if ((strpos($message, 'Error validating access token') != false) ||
                    (strpos($message, 'Invalid OAuth access token') != false) ||
                    (strpos($message, 'An active access token must be used') != false))
                {
                    $this->destroySession();
                }
                break;
    }

    throw $e;
}

/**
 * Prints to the error log if you aren't in command line mode.
 *
 * @param string $msg Log message
 */
protected static function errorLog($msg) {
    // disable error log if we are running in a CLI environment
    // @codeCoverageIgnoreStart
    if (php_sapi_name() != 'cli') {
        error_log($msg);
    }
    // uncomment this if you want to see the errors on the page
    // print 'error_log: '.$msg."\n";
    // @codeCoverageIgnoreEnd
}

/**
 * Base64 encoding that doesn't need to be urlencode()ed.
 * Exactly the same as base64_encode except it uses
 * - instead of +
 * _ instead of /
 * No padded =
 *
 * @param string $input base64UrlEncoded input
 *
 * @return string The decoded string
 */
protected static function base64UrlDecode($input) {
    return base64_decode(strtr($input, '-_', '+/'));
}

/**

```

```

* Base64 encoding that doesn't need to be urlencode()ed.
* Exactly the same as base64_encode except it uses
* - instead of +
* - instead of /
*
* @param string $input The input to encode
* @return string The base64Url encoded input, as a string.
*/
protected static function base64UrlEncode($input) {
    $str = strtr(base64_encode($input), '+/', '-_');
    $str = str_replace('=', '', $str);
    return $str;
}

/**
 * Destroy the current session
 */
public function destroySession() {
    $this->accessToken = null;
    $this->signedRequest = null;
    $this->user = null;
    $this->clearAllPersistentData();

    // Javascript sets a cookie that will be used in getSignedRequest that we
    // need to clear if we can
    $cookie_name = $this->getSignedRequestCookieName();
    if (array_key_exists($cookie_name, $_COOKIE)) {
        unset($_COOKIE[$cookie_name]);
        if (!headers_sent()) {
            $base_domain = $this->getBaseDomain();
            setcookie($cookie_name, '', 1, '/', '.'.$base_domain);
        } else {
            // @codeCoverageIgnoreStart
            self::errorLog(
                'There exists a cookie that we wanted to clear that we couldn\'t '.
                'clear because headers was already sent. Make sure to do the first '.
                'API call before outputting anything.'
            );
            // @codeCoverageIgnoreEnd
        }
    }
}

/**
 * Parses the metadata cookie that our Javascript API set
 *
 * @return array an array mapping key to value
 */
protected function getMetadataCookie() {
    $cookie_name = $this->getMetadataCookieName();
    if (!array_key_exists($cookie_name, $_COOKIE)) {
        return array();
    }

    // The cookie value can be wrapped in " characters so remove them
    $cookie_value = trim($_COOKIE[$cookie_name], '"');

    if (empty($cookie_value)) {
        return array();
    }

    $parts = explode('&', $cookie_value);
    $metadata = array();
    foreach ($parts as $part) {
        $pair = explode('=', $part, 2);
        if (!empty($pair[0])) {
            $metadata[urldecode($pair[0])] =
                (count($pair) > 1) ? urldecode($pair[1]) : '';
        }
    }

    return $metadata;
}

/**
 * Finds whether the given domain is allowed or not
 *
 * @param string $big The value to be checked against $small
 * @param string $small The input string
 *
 * @return boolean Returns TRUE if $big matches $small
 */
protected static function isAllowedDomain($big, $small) {
    if ($big === $small) {
        return true;
    }
    return self::endsWith($big, '.'.$small);
}

/**
 * Checks if $big string ends with $small string
 *
 * @param string $big The value to be checked against $small
 * @param string $small The input string
 *
 * @return boolean TRUE if $big ends with $small
 */
protected static function endsWith($big, $small) {
    $len = strlen($small);

```

```

    if ($len == 0) {
        return true;
    }
    return substr($big, -$len) == $small;
}

/**
 * Each of the following four methods should be overridden in
 * a concrete subclass, as they are in the provided Facebook class.
 * The Facebook class uses PHP sessions to provide a primitive
 * persistent store, but another subclass—one that you implement—
 * might use a database, memcache, or an in-memory cache.
 *
 * @see Facebook
 */

/**
 * Stores the given ($key, $value) pair, so that future calls to
 * getPersistentData($key) return $value. This call may be in another request.
 *
 * @param string $key
 * @param array $value
 *
 * @return void
 */
abstract protected function setPersistentData($key, $value);

/**
 * Get the data for $key, persisted by BaseFacebook::setPersistentData()
 *
 * @param string $key The key of the data to retrieve
 * @param boolean $default The default value to return if $key is not found
 *
 * @return mixed
 */
abstract protected function getPersistentData($key, $default = false);

/**
 * Clear the data with $key from the persistent storage
 *
 * @param string $key
 *
 * @return void
 */
abstract protected function clearPersistentData($key);

/**
 * Clear all data from the persistent storage
 *
 * @return void
 */
abstract protected function clearAllPersistentData();
}

```

---

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/**
 * Copyright 2011 Facebook, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may
 * not use this file except in compliance with the License. You may obtain
 * a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

require_once "base_facebook.php";

/**
 * Extends the BaseFacebook class with the intent of using
 * PHP sessions to store user ids and access tokens.
 */
class Facebook extends BaseFacebook
{
    /**
     * Cookie prefix
     */
    const FBSS_COOKIE_NAME = 'fbss';

    /**
     * We can set this to a high number because the main session
     * expiration will trump this.
     */
    const FBSS_COOKIE_EXPIRE = 31556926; // 1 year

    /**
     * Stores the shared session ID if one is set.
     *
     * @var string
     */
    protected $sharedSessionID;

    /**

```

```

* Identical to the parent constructor, except that
* we start a PHP session to store the user ID and
* access token if during the course of execution
* we discover them.
*
* @param array $config the application configuration. Additionally
* accepts "sharedSession" as a boolean to turn on a secondary
* cookie for environments with a shared session (that is, your app
* shares the domain with other apps).
*
* @see BaseFacebook::__construct
*/
public function __construct($config) {
    if ((function_exists('session_status')
        && session_status() !== PHP_SESSION_ACTIVE) || !session_id()) {
        session_start();
    }
    parent::__construct($config);
    if (!empty($config['sharedSession'])) {
        $this->initSharedSession();

        // re-load the persisted state, since parent
        // attempted to read out of non-shared cookie
        $state = $this->getPersistentData('state');
        if (!empty($state)) {
            $this->state = $state;
        } else {
            $this->state = null;
        }
    }
}

/**
 * Supported keys for persistent data
 *
 * @var array
 */
protected static $kSupportedKeys =
    array('state', 'code', 'access_token', 'user_id');

/**
 * Initiates Shared Session
 */
protected function initSharedSession() {
    $cookie_name = $this->getSharedSessionCookieName();
    if (isset($_COOKIE[$cookie_name])) {
        $data = $this->parseSignedRequest($_COOKIE[$cookie_name]);
        if ($data && !empty($data['domain']) &&
            self::isAllowedDomain($this->getHttpHost(), $data['domain'])) {
            // good case
            $this->sharedSessionID = $data['id'];
            return;
        }
        // ignoring potentially unreachable data
    }
    // evil/corrupt/missing case
    $base_domain = $this->getBaseDomain();
    $this->sharedSessionID = md5(uniqid(mt_rand(), true));
    $cookie_value = $this->makeSignedRequest(
        array(
            'domain' => $base_domain,
            'id' => $this->sharedSessionID,
        )
    );
    $_COOKIE[$cookie_name] = $cookie_value;
    if (!headers_sent()) {
        $expire = time() + self::FBSS_COOKIE_EXPIRE;
        setcookie($cookie_name, $cookie_value, $expire, '/', '.'.$base_domain);
    } else {
        // @codeCoverageIgnoreStart
        self::errorLog(
            'Shared session ID cookie could not be set! You must ensure you '
            . 'create the Facebook instance before headers have been sent. This '
            . 'will cause authentication issues after the first request.'
        );
        // @codeCoverageIgnoreEnd
    }
}

/**
 * Provides the implementations of the inherited abstract
 * methods. The implementation uses PHP sessions to maintain
 * a store for authorization codes, user ids, CSRF states, and
 * access tokens.
 */

/**
 * {@inheritdoc}
 *
 * @see BaseFacebook::setPersistentData()
 */
protected function setPersistentData($key, $value) {
    if (!in_array($key, self::$kSupportedKeys)) {
        self::errorLog('Unsupported key passed to setPersistentData.');
```

```

    $_SESSION[$session_var_name] = $value;
}

/**
 * {@inheritdoc}
 *
 * @see BaseFacebook::getPersistentData()
 */
protected function getPersistentData($key, $default = false) {
    if (!in_array($key, self::$kSupportedKeys)) {
        self::errorLog('Unsupported key passed to getPersistentData.');
```

return \$default;

```
    }

    $session_var_name = $this->constructSessionVariableName($key);
    return isset($_SESSION[$session_var_name]) ?
        $_SESSION[$session_var_name] : $default;
}

/**
 * {@inheritdoc}
 *
 * @see BaseFacebook::clearPersistentData()
 */
protected function clearPersistentData($key) {
    if (!in_array($key, self::$kSupportedKeys)) {
        self::errorLog('Unsupported key passed to clearPersistentData.');
```

return;

```
    }

    $session_var_name = $this->constructSessionVariableName($key);
    if (isset($_SESSION[$session_var_name])) {
        unset($_SESSION[$session_var_name]);
    }
}

/**
 * {@inheritdoc}
 *
 * @see BaseFacebook::clearAllPersistentData()
 */
protected function clearAllPersistentData() {
    foreach (self::$kSupportedKeys as $key) {
        $this->clearPersistentData($key);
    }
    if ($this->sharedSessionID) {
        $this->deleteSharedSessionCookie();
    }
}

/**
 * Deletes Shared session cookie
 */
protected function deleteSharedSessionCookie() {
    $cookie_name = $this->getSharedSessionCookieName();
    unset($_COOKIE[$cookie_name]);
    $base_domain = $this->getBaseDomain();
    setcookie($cookie_name, '', 1, '/', '.'.$base_domain);
}

/**
 * Returns the Shared session cookie name
 *
 * @return string The Shared session cookie name
 */
protected function getSharedSessionCookieName() {
    return self::FBSS_COOKIE_NAME . '_' . $this->getAppId();
}

/**
 * Constructs and returns the name of the session key.
 *
 * @see setPersistentData()
 * @param string $key The key for which the session variable name to construct.
 *
 * @return string The name of the session key.
 */
protected function constructSessionVariableName($key) {
    $parts = array('fb', $this->getAppId(), $key);
    if ($this->sharedSessionID) {
        array_unshift($parts, $this->sharedSessionID);
    }
    return implode('-', $parts);
}
}

```

---

```

<?php
if (! defined('BASEPATH')) exit('No direct script access allowed');
/**
 * @name MY_Pagination.php
 * @version 1.0
 * @author Joost van Veen www.accentinteractive.nl
 * @created: Sun Jul 27 16:27:26 GMT 2008 16:27:26
 *
 * A simple Pagination extension to make working with pagination a bit easier.
 * I created this lib because I had URIs in my app in which the paging element
 * was not always in the same segment, which makes it a pain if you work with
 * the default pagination class.

```

```

*
* This simple lib accomplishes the following:
* - It determines waht the 'base-url' is, so you don't have to set it yourself
* - It removes the need for you setting the infamous 'uri.segment' setting
*
* Basically ,it sets paging at the end of the uri , without having to pass a uri
* segment. The library relies on a unique pagination selector , which it uses to
* determine if and where the pagnition offset is located in the URI.
*
* E.g. /example/pagination/Page/3
*
* The lib searches for the pagination_selector ('Page', in the above example)
* and retracts the proper offset value (in this case 3)
*
* The pagination links are automatically created , just as in CI's default
* pagination lib.
*
* Requirements
* Codeigniter 2+
* PHP 5
* A *unique* pagination selector (default is 'Page') - unique meaning a string
* you are sure will never appear in the uri , except for pagination.
*
* If there we use pagination , it must ALWAYS follow the following syntax and be
* located at the END of the URI:
* PAGINATION.SELECTOR/offset
*
* The PAGINATION.SELECTOR is a special string which we know will ONLY be in the
* URI when paging is set. Let's say the PAGINATION.SELECTOR is 'Page' (since most
* coders never use any capitals in the URI, most of the times any string with
* a single capital character in it will suffice).
*
* Example use (in controller):
* // Initialize pagination
* $config['total_rows'] = $this->db->count_all_results('my_table');
* $config['per_page'] = 10; // You'd best set this in a config file , but hey
* $this->pagination->initialize($config);
* $this->data['pagination'] = $this->pagination->create_links();
*
* // Retrieve paginated results , using the dynamically determined offset
* $this->db->limit($config['per_page'], $this->pagination->offset);
* $query = $this->db->get('my_table');
*
*/
class MY_Pagination extends CI_Pagination
{
    var $offset = 0;
    var $pagination_selector = 'Page';
    var $index_page;
    function MY_Pagination ()
    {
        parent::__construct();
        log_message('debug', "MY_Pagination Class Initialized");

        $this->index_page = config_item('index_page') != '' ? config_item('index_page') . '/'
            : '';
        $this->_set_pagination_offset();
    }
    /**
     * Set dynamic pagination variables in $CI->data['pagvars']
     */
    /*
    function _set_pagination_offset ()
    {

        // Instantiate the CI super object so we have access to the uri class
        $CI = & get_instance();

        // Store pagination offset if it is set
        if (strpos($CI->uri->uri_string(), $this->pagination_selector)) {

            // Get the segment offset for the pagination selector
            $segments = $CI->uri->segment_array();

            // Loop through segments to retrieve pagination offset
            foreach ($segments as $key => $value) {

                // Find the pagination_selector and work from there
                if ($value == $this->pagination_selector) {

                    // Store pagination offset
                    $this->offset = $CI->uri->segment($key + 1);

                    // Store pagination segment
                    $this->uri_segment = $key + 1;

                    // Set base url for paging. This only works if the
                    // pagination_selector and paging offset are AT THE END of
                    // the URI!
                    $uri = $CI->uri->uri_string();
                    $pos = strpos($uri, $this->pagination_selector);
                    $this->base_url = config_item('base_url') . $this->index_page . substr(
                        $uri, 0, $pos + strlen($this->pagination_selector));

                }

            }

        }
    }
}
else { // Pagination selector was not found in URI string. So offset is 0

```



```

        $this->offset = 0;
        $this->uri_segment = 0;
        $this->base_url = config_item('base_url') . $this->index_page . $CI->uri->
            uri_string() . '/' . $this->pagination_selector;
    }
}
}
}

```

---

```

<?php
/**
 * tmhOAuth
 *
 * An OAuth library written in PHP.
 * The library supports file uploading using multipart/form as well as general
 * REST requests. OAuth authentication is sent using an Authorization Header.
 *
 * @author themattharris
 * @version 0.8.4
 *
 * 06 Aug 2014
 */
defined('__DIR__') or define('__DIR__', dirname(__FILE__));

class tmhOAuth {
    const VERSION = '0.8.4';
    var $response = array();

    /**
     * Creates a new tmhOAuth object
     *
     * @param string $config, the configuration to use for this request
     * @return void
     */
    public function __construct($config=array()) {
        $this->buffer = null;
        $this->reconfigure($config);
        $this->reset_request_settings();
        $this->set_user_agent();
    }

    public function reconfigure($config=array()) {
        // default configuration options
        $this->config = array_merge(
            array(
                // leave 'user-agent' blank for default, otherwise set this to
                // something that clearly identifies your app
                'user-agent' => '',
                'host' => 'api.twitter.com',
                'method' => 'GET',

                'consumer_key' => '',
                'consumer_secret' => '',
                'token' => '',
                'secret' => '',

                // OAuth2 bearer token. This should already be URL encoded
                'bearer' => '',

                // oauth signing variables that are not dynamic
                'oauth_version' => '1.0',
                'oauth_signature_method' => 'HMAC-SHA1',

                // you probably don't want to change any of these curl values
                'curl_http_version' => CURL_HTTP_VERSION_1.1,
                'curl_connecttimeout' => 30,
                'curl_timeout' => 10,

                // for security this should always be set to 2.
                'curl_ssl_verifyhost' => 2,
                // for security this should always be set to true.
                'curl_ssl_verifypeer' => true,
                // for security this should always be set to true.
                'use_ssl' => true,

                // you can get the latest cacert.pem from here http://curl.haxx.se/ca/cacert.pem
                // if you're getting HTTP 0 responses, check cacert.pem exists and is readable
                // without it curl won't be able to create an SSL connection
                'curl_cainfo' => __DIR__ . DIRECTORY_SEPARATOR . 'cacert.pem',
                'curl_capath' => __DIR__,

                // in some cases (very very odd ones) the SSL version must be set manually.
                // unless you know why your are changing this, you should leave it as false
                // to allow PHP to determine the value for this setting itself.
                'curl_sslversion' => false,

                'curl_followlocation' => false, // whether to follow redirects or not

                // support for proxy servers
                'curl_proxy' => false, // really you don't want to use this if you are
                // using streaming
                'curl_proxyuserpwd' => false, // format username:password for proxy, if
                // required
                'curl_encoding' => '', // leave blank for all supported formats, else
                // use gzip, deflate, identity etc

                // streaming API configuration
            )
        );
    }
}

```

```

        'is_streaming' => false,
        'streaming_eol' => "\r\n",
        'streaming_metrics_interval' => 10,

        // header or querystring. You should always use header!
        // this is just to help me debug other developers implementations
        'as_header' => true,
        'force_nonce' => false, // used for checking signatures. leave as false
        for auto
        'force_timestamp' => false, // used for checking signatures. leave as false
        for auto
    ),
    $config
);
}

private function reset_request_settings($options=array()) {
    $this->request_settings = array(
        'params' => array(),
        'headers' => array(),
        'with_user' => true,
        'multipart' => false,
    );

    if (!empty($options))
        $this->request_settings = array_merge($this->request_settings, $options);
}

/**
 * Sets the useragent for PHP to use
 * If '$this->config['user-agent']' already has a value it is used instead of one
 * being generated.
 *
 * @return void value is stored to the config array class variable
 */
private function set_user_agent() {
    if (!empty($this->config['user-agent']))
        return;

    $ssl = ($this->config['curl_ssl_verifyhost'] && $this->config['curl_ssl_verifypeer'] &&
        $this->config['use_ssl']) ? '+' : '-';
    $ua = 'tmhOAuth ' . self::VERSION . $ssl . 'SSL - //github.com/themathharris/tmhOAuth';
    $this->config['user-agent'] = $ua;
}

/**
 * Generates a random OAuth nonce.
 * If 'force_nonce' is false a nonce will be generated, otherwise the value of '$this->
    config['force_nonce']' will be used.
 *
 * @param string $length how many characters the nonce should be before MD5 hashing. default
    12
 * @param string $include_time whether to include time at the beginning of the nonce.
    default true
 * @return $nonce as a string
 */
private function nonce($length=12, $include_time=true) {
    if ($this->config['force_nonce'] == false) {
        $prefix = $include_time ? microtime() : '';
        return md5(substr($prefix . uniqid(), 0, $length));
    } else {
        return $this->config['force_nonce'];
    }
}

/**
 * Generates a timestamp.
 * If 'force_timestamp' is false a timestamp will be generated, otherwise the value of '
    $this->config['force_timestamp']' will be used.
 *
 * @return $time as a string
 */
private function timestamp() {
    if ($this->config['force_timestamp'] == false) {
        $time = time();
    } else {
        $time = $this->config['force_timestamp'];
    }
    return (string) $time;
}

/**
 * Encodes the string or array passed in a way compatible with OAuth.
 * If an array is passed each array value will will be encoded.
 *
 * @param mixed $data the scalar or array to encode
 * @return $data encoded in a way compatible with OAuth
 */
private function safe_encode($data) {
    if (is_array($data)) {
        return array_map(array($this, 'safe_encode'), $data);
    } else if (is_scalar($data)) {
        return str_replace(
            array('+', '%7E'),
            array('~', '%'),
            rawurlencode($data)
        );
    } else {
        return '';
    }
}

```

```

}
}

/**
 * Decodes the string or array from it's URL encoded form
 * If an array is passed each array value will be decoded.
 *
 * @param mixed $data the scalar or array to decode
 * @return string $data decoded from the URL encoded form
 */
private function safe_decode($data) {
    if (is_array($data)) {
        return array_map(array($this, 'safe_decode'), $data);
    } else if (is_scalar($data)) {
        return rawurldecode($data);
    } else {
        return '';
    }
}

/**
 * Prepares OAuth1 signing parameters.
 *
 * @return void all required OAuth parameters, safely encoded, are stored to the class
 * variable '$this->request_settings['oauth1-params']'
 */
private function prepare_oauth1_params() {
    $defaults = array(
        'oauth_nonce' => $this->nonce(),
        'oauth_timestamp' => $this->timestamp(),
        'oauth_version' => $this->config['oauth_version'],
        'oauth_consumer_key' => $this->config['consumer_key'],
        'oauth_signature_method' => $this->config['oauth_signature_method'],
    );

    // include the user token if it exists
    if ( $oauth_token = $this->token() )
        $defaults['oauth_token'] = $oauth_token;

    $this->request_settings['oauth1-params'] = array();

    // safely encode
    foreach ($defaults as $k => $v) {
        $this->request_settings['oauth1-params'][$this->safe_encode($k)] = $this->safe_encode($v);
    }
}

private function token() {
    if ( $this->request_settings['with_user'] ) {
        if (isset($this->config['token']) && !empty($this->config['token'])) return $this->
            config['token'];
        elseif (isset($this->config['user_token'])) return $this->config['user_token'];
    }
    return '';
}

private function secret() {
    if ( $this->request_settings['with_user'] ) {
        if (isset($this->config['secret']) && !empty($this->config['secret'])) return $this->
            config['secret'];
        elseif (isset($this->config['user_secret'])) return $this->config['user_secret'];
    }
    return '';
}

/**
 * Extracts and decodes OAuth parameters from the passed string
 *
 * @param string $body the response body from an OAuth flow method
 * @return array the response body safely decoded to an array of key => values
 */
public function extract_params($body) {
    $kvs = explode('&', $body);
    $decoded = array();
    foreach ($kvs as $kv) {
        $kv = explode('=', $kv, 2);
        $kv[0] = $this->safe_decode($kv[0]);
        $kv[1] = $this->safe_decode($kv[1]);
        $decoded[$kv[0]] = $kv[1];
    }
    return $decoded;
}

/**
 * Prepares the HTTP method for use in the base string by converting it to
 * uppercase.
 *
 * @return void value is stored to the class variable '$this->request_settings['method']'
 */
private function prepare_method() {
    $this->request_settings['method'] = strtoupper($this->request_settings['method']);
}

/**
 * Prepares the URL for use in the base string by ripping it apart and
 * reconstructing it.
 *
 * Ref: 3.4.1.2

```

```

*
* @return void value is stored to the class array variable '$this->request_settings['url']'
*/
private function prepare_url() {
    $parts = parse_url($this->request_settings['url']);

    $port = isset($parts['port']) ? $parts['port'] : false;
    $scheme = $parts['scheme'];
    $host = $parts['host'];
    $path = isset($parts['path']) ? $parts['path'] : false;

    $port or $port = ($scheme == 'https') ? '443' : '80';

    if (($scheme == 'https' && $port != '443') || ($scheme == 'http' && $port != '80')) {
        $host = "Host:$port";
    }

    // the scheme and host MUST be lowercase
    $this->request_settings['url'] = strtolower("$scheme://$host");
    // but not the path
    $this->request_settings['url'] .= $path;
}

/**
 * If the request uses multipart, and the parameter isn't a file path, prepend a space
 * otherwise return the original value. we chose a space here as twitter whitespace trims
 * the beginning of the tweet. we don't use \0 here because it's the character for string
 * termination.
 *
 * @param the parameter value
 * @return string the original or modified string, depending on the request and the input
 * parameter
 */
private function multipart_escape($value) {
    if (!$this->request_settings['multipart'] || strpos($value, '@') !== 0)
        return $value;

    // see if the parameter is a file.
    // we split on the semi-colon as it's the delimiter used on media uploads
    // for fields with semi-colons this will return the original string
    list($file) = explode(';', substr($value, 1), 2);
    if (file_exists($file))
        return $value;

    return " $value";
}

/**
 * Prepares all parameters for the base string and request.
 * Multipart parameters are ignored as they are not defined in the specification,
 * all other types of parameter are encoded for compatibility with OAuth.
 *
 * @param array $params the parameters for the request
 * @return void prepared values are stored in the class array variable '$this->
 * request_settings'
 */
private function prepare_params() {
    $doing_oauth1 = false;
    $this->request_settings['prepared_params'] = array();
    $prepared = &$this->request_settings['prepared_params'];
    $prepared_pairs = array();
    $prepared_pairs_with_oauth = array();

    if (isset($this->request_settings['oauth1_params'])) {
        $oauth1 = &$this->request_settings['oauth1_params'];
        $doing_oauth1 = true;
        $params = array_merge($oauth1, $this->request_settings['params']);

        // Remove oauth_signature if present
        // Ref: Spec: 9.1.1 ("The oauth_signature parameter MUST be excluded.")
        unset($params['oauth_signature']);

        // empty the oauth1 array. we reset these values later in this method
        $oauth1 = array();
    } else {
        $params = $this->request_settings['params'];
    }

    // Parameters are sorted by name, using lexicographical byte value ordering.
    // Ref: Spec: 9.1.1 (1)
    uksort($params, 'strcmp');

    // set this now so we're not doing it on every parameter
    $supports_curl_file = class_exists('CurlFile', false);

    // encode params unless we're doing multipart
    foreach ($params as $k => $v) {
        $k = $this->request_settings['multipart'] ? $k : $this->safe_encode($k);

        if (is_array($v))
            $v = implode(',', $v);

        // we don't need to do the multipart escaping if we support curlfile
        if ($supports_curl_file && ($v instanceof CurlFile)) {
            // leave $v alone
        } elseif ($this->request_settings['multipart']) {
            $v = $this->multipart_escape($v);
        }
    }
}

```

```

    } else {
        $v = $this->safe_encode($v);
    }

    // split parameters for the basestring and authorization header, and recreate the oauth1
    array
    if ($doing_oauth1) {
        // if we're doing multipart, only store the oauth_* params, ignore the users request
        params
        if ((strpos($k, 'oauth') === 0) || !$this->request_settings['multipart'])
            $prepared_pairs_with_oauth[] = "{k}={v}";

        if (strpos($k, 'oauth') === 0) {
            $oauth1[$k] = $v;
            continue;
        }
    }
    $prepared[$k] = $v;

    if (!$this->request_settings['multipart'])
        $prepared_pairs[] = "{k}={v}";
}

if ($doing_oauth1) {
    $this->request_settings['basestring_params'] = implode('&', $prepared_pairs_with_oauth);
}

// setup params for GET/POST/PUT method handling
if (!empty($prepared)) {
    $content = implode('&', $prepared_pairs);

    switch ($this->request_settings['method']) {
        case 'PUT':
            // fall through to POST as PUT should be treated the same
        case 'POST':
            $this->request_settings['postfields'] = $this->request_settings['multipart'] ?
                $prepared : $content;
            break;
        default:
            $this->request_settings['querystring'] = $content;
            break;
    }
}
}

/**
 * Prepares the OAuth signing key
 *
 * @return void prepared signing key is stored in the class variable 'signing_key'
 */
private function prepare_signing_key() {
    $left = $this->safe_encode($this->config['consumer_secret']);
    $right = $this->safe_encode($this->secret());
    $this->request_settings['signing_key'] = $left . '&' . $right;
}

/**
 * Prepare the base string.
 * Ref: Spec: 9.1.3 ("Concatenate Request Elements")
 *
 * @return void prepared base string is stored in the class variable 'base_string'
 */
private function prepare_base_string() {
    $url = $this->request_settings['url'];

    # if the host header is set we need to rewrite the basestring to use
    # that, instead of the request host. otherwise the signature won't match
    # on the server side
    if (!empty($this->request_settings['headers']['Host'])) {
        $url = str_ireplace(
            $this->config['host'],
            $this->request_settings['headers']['Host'],
            $url
        );
    }

    $base = array(
        $this->request_settings['method'],
        $url,
        $this->request_settings['basestring_params']
    );
    $this->request_settings['basestring'] = implode('&', $this->safe_encode($base));
}

/**
 * Signs the OAuth 1 request
 *
 * @return void oauth_signature is added to the parameters in the class array variable '
 * $this->request_settings'
 */
private function prepare_oauth_signature() {
    $this->request_settings['oauth1_params']['oauth_signature'] = $this->safe_encode(
        base64_encode(
            hash_hmac(
                'sha1', $this->request_settings['basestring'], $this->request_settings['signing_key'], true
            )
        )
    );
}
}

```

```

/**
 * Prepares the Authorization header
 *
 * @return void prepared authorization header is stored in the class variable headers['
 *         Authorization']
 */
private function prepare_auth_header() {
    if (!$this->config['as_header'])
        return;

    // oauth1
    if (isset($this->request_settings['oauth1_params'])) {
        // sort again as oauth_signature was added post param preparation
        uksort($this->request_settings['oauth1_params'], 'strcmp');
        $encoded_quoted_pairs = array();
        foreach ($this->request_settings['oauth1_params'] as $k => $v) {
            $encoded_quoted_pairs[] = "{$k}=\"{$v}\"";
        }
        $header = 'OAuth ' . implode(' ', $encoded_quoted_pairs);
    } elseif (!empty($this->config['bearer'])) {
        $header = 'Bearer ' . $this->config['bearer'];
    }

    if (isset($header))
        $this->request_settings['headers']['Authorization'] = $header;
}

/**
 * Create the bearer token for OAuth2 requests from the consumer_key and consumer_secret.
 *
 * @return string the bearer token
 */
public function bearer_token_credentials() {
    $credentials = implode(':', array(
        $this->safe_encode($this->config['consumer_key']),
        $this->safe_encode($this->config['consumer_secret'])
    ));
    return base64_encode($credentials);
}

/**
 * Make an HTTP request using this library. This method doesn't return anything.
 * Instead the response should be inspected directly.
 *
 * @param string $method the HTTP method being used. e.g. POST, GET, HEAD etc
 * @param string $url the request URL without query string parameters
 * @param array $params the request parameters as an array of key=value pairs. Default empty
 *         array
 * @param string $useauth whether to use authentication when making the request. Default
 *         true
 * @param string $multipart whether this request contains multipart data. Default false
 * @param array $headers any custom headers to send with the request. Default empty array
 * @return int the http response code for the request. 0 is returned if a connection could
 *         not be made
 */
public function request($method, $url, $params=array(), $useauth=true, $multipart=false,
    $headers=array()) {
    $options = array(
        'method' => $method,
        'url' => $url,
        'params' => $params,
        'with_user' => true,
        'multipart' => $multipart,
        'headers' => $headers
    );
    $options = array_merge($this->default_options(), $options);

    if ($useauth) {
        return $this->user_request($options);
    } else {
        return $this->unauthenticated_request($options);
    }
}

public function aponly_request($options=array()) {
    $options = array_merge($this->default_options(), $options, array(
        'with_user' => false,
    ));
    $this->reset_request_settings($options);
    if ($options['without_bearer']) {
        return $this->oauth1_request();
    } else {
        $this->prepare_method();
        $this->prepare_url();
        $this->prepare_params();
        $this->prepare_auth_header();
        return $this->curlit();
    }
}

public function user_request($options=array()) {
    $options = array_merge($this->default_options(), $options, array(
        'with_user' => true,
    ));
    $this->reset_request_settings($options);
    return $this->oauth1_request();
}

public function unauthenticated_request($options=array()) {

```

```

    $options = array_merge($this->default_options(), $options, array(
        'with_user' => false,
    ));
    $this->reset_request_settings($options);
    $this->prepare_method();
    $this->prepare_url();
    $this->prepare_params();
    return $this->curlit();
}

/**
 * Signs the request and adds the OAuth signature. This runs all the request
 * parameter preparation methods.
 *
 * @param string $method the HTTP method being used. e.g. POST, GET, HEAD etc
 * @param string $url the request URL without query string parameters
 * @param array $params the request parameters as an array of key=value pairs
 * @param boolean $with_user whether to include the user credentials when making the request
 *
 * @return void
 */
private function oauth1_request() {
    $this->prepare_oauth1_params();
    $this->prepare_method();
    $this->prepare_url();
    $this->prepare_params();
    $this->prepare_base_string();
    $this->prepare_signing_key();
    $this->prepare_oauth_signature();
    $this->prepare_auth_header();
    return $this->curlit();
}

private function default_options() {
    return array(
        'method' => 'GET',
        'params' => array(),
        'with_user' => true,
        'multipart' => false,
        'headers' => array(),
        'without_bearer' => false,
    );
}

/**
 * Make a long poll HTTP request using this library. This method is
 * different to the other request methods as it isn't supposed to disconnect
 *
 * Using this method expects a callback which will receive the streaming
 * responses.
 *
 * @param string $method the HTTP method being used. e.g. POST, GET, HEAD etc
 * @param string $url the request URL without query string parameters
 * @param array $params the request parameters as an array of key=value pairs
 * @param string $callback the callback function to stream the buffer to.
 * @return void
 */
public function streaming_request($method, $url, $params=array(), $callback='') {
    if ( ! empty($callback) ) {
        if ( ! is_callable($callback) ) {
            return false;
        }
        $this->config['streaming_callback'] = $callback;
    }
    $this->metrics['start'] = time();
    $this->metrics['interval_start'] = $this->metrics['start'];
    $this->metrics['messages'] = 0;
    $this->metrics['last_messages'] = 0;
    $this->metrics['bytes'] = 0;
    $this->metrics['last_bytes'] = 0;
    $this->config['is_streaming'] = true;
    $this->request($method, $url, $params);
}

/**
 * Handles the updating of the current Streaming API metrics.
 *
 * @return array the metrics for the streaming api connection
 */
private function update_metrics() {
    $now = time();
    if (($this->metrics['interval_start'] + $this->config['streaming_metrics_interval']) >
        $now)
        return null;

    $this->metrics['mps'] = round( ($this->metrics['messages'] - $this->metrics['last_messages']
        ) / $this->config['streaming_metrics_interval'], 2);
    $this->metrics['bps'] = round( ($this->metrics['bytes'] - $this->metrics['last_bytes'] ) /
        $this->config['streaming_metrics_interval'], 2);

    $this->metrics['last_bytes'] = $this->metrics['bytes'];
    $this->metrics['last_messages'] = $this->metrics['messages'];
    $this->metrics['interval_start'] = $now;
    return $this->metrics;
}

/**
 * Utility function to create the request URL in the requested format.
 * If a fully-qualified URI is provided, it will be returned.

```

```

* Any multi-slashes (except for the protocol) will be replaced with a single slash.
*
*
* @param string $request the API method without extension
* @param string $extension the format of the response. Default json. Set to an empty string
*   to exclude the format
* @return string the concatenation of the host, API version, API method and format, or
*   $request if it begins with http
*/
public function url($request, $extension='json') {
    // remove multi-slashes
    $request = preg_replace('$([:]+)/+$', '$1/', $request);

    if (stripos($request, 'http') === 0 || stripos($request, '//') === 0) {
        return $request;
    }

    $extension = strlen($extension) > 0 ? ".$extension" : '';
    $proto = $this->config['use-ssl'] ? 'https://' : 'http://';

    // trim trailing slash
    $request = ltrim($request, '/');

    $pos = strlen($request) - strlen($extension);
    if (substr($request, $pos) === $extension)
        $request = substr_replace($request, '', $pos);

    return implode('/', array(
        $proto,
        $this->config['host'],
        $request . $extension
    ));
}

/**
 * Public access to the private safe decode/encode methods
 *
 * @param string $text the text to transform
 * @param string $mode the transformation mode. either encode or decode
 * @return string $text transformed by the given $mode
 */
public function transformText($text, $mode='encode') {
    return $this->{"safe_$mode"}($text);
}

/**
 * Utility function to parse the returned curl headers and store them in the
 * class array variable.
 *
 * @param object $ch curl handle
 * @param string $header the response headers
 * @return string the length of the header
 */
private function curlHeader($ch, $header) {
    $this->response['raw'] .= $header;

    list($key, $value) = array_pad(explode(':', $header, 2), 2, null);

    $key = trim($key);
    $value = trim($value);

    if ( ! isset($this->response['headers'][$key])) {
        $this->response['headers'][$key] = $value;
    } else {
        if ( ! is_array($this->response['headers'][$key])) {
            $this->response['headers'][$key] = array($this->response['headers'][$key]);
        }
        $this->response['headers'][$key][] = $value;
    }

    return strlen($header);
}

/**
 * Utility function to parse the returned curl buffer and store them until
 * an EOL is found. The buffer for curl is an undefined size so we need
 * to collect the content until an EOL is found.
 *
 * This function calls the previously defined streaming callback method.
 *
 * @param object $ch curl handle
 * @param string $data the current curl buffer
 * @return int the length of the data string processed in this function
 */
private function curlWrite($ch, $data) {
    $l = strlen($data);
    if (strpos($data, $this->config['streaming_eol']) === false) {
        $this->buffer .= $data;
        return $l;
    }

    $buffered = explode($this->config['streaming_eol'], $data);
    $content = $this->buffer . $buffered[0];

    $this->metrics['messages']++;
    $this->metrics['bytes'] += strlen($content);

    if ( ! is_callable($this->config['streaming_callback']))
        return 0;
}

```



```

$metrics = $this->update_metrics();
$stop = call_user_func(
    $this->config['streaming_callback'],
    $content,
    strlen($content),
    $metrics
);
$this->buffer = $buffered[1];
if ($stop)
    return 0;

return $l;
}

/**
 * Makes a curl request. Takes no parameters as all should have been prepared
 * by the request method
 *
 * the response data is stored in the class variable 'response'
 *
 * @return int the http response code for the request. 0 is returned if a connection could
 *         not be made
 */
private function curlit() {
    $this->response = array(
        'raw' => ''
    );

    // configure curl
    $c = curl_init();

    if ($this->request_settings['method'] == 'GET' && isset($this->request_settings['
        querystring'])) {
        $this->request_settings['url'] = $this->request_settings['url'] . '?' . $this->
            request_settings['querystring'];
    } elseif ($this->request_settings['method'] == 'POST' || $this->request_settings['method']
        == 'PUT') {
        $postfields = array();
        if (isset($this->request_settings['postfields']))
            $postfields = $this->request_settings['postfields'];

        curl_setopt($c, CURLOPT_POSTFIELDS, $postfields);
    }

    curl_setopt($c, CURLOPT_CUSTOMREQUEST, $this->request_settings['method']);

    curl_setopt_array($c, array(
        CURLOPT_HTTP_VERSION => $this->config['curl_http_version'],
        CURLOPT_USERAGENT => $this->config['user_agent'],
        CURLOPT_CONNECTTIMEOUT => $this->config['curl_connecttimeout'],
        CURLOPT_TIMEOUT => $this->config['curl_timeout'],
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_SSL_VERIFYPEER => $this->config['curl_ssl_verifypeer'],
        CURLOPT_SSL_VERIFYHOST => $this->config['curl_ssl_verifyhost'],

        CURLOPT_FOLLOWLOCATION => $this->config['curl_followlocation'],
        CURLOPT_PROXY => $this->config['curl_proxy'],
        CURLOPT_ENCODING => $this->config['curl_encoding'],
        CURLOPT_URL => $this->request_settings['url'],
        // process the headers
        CURLOPT_HEADERFUNCTION => array($this, 'curlHeader'),
        CURLOPT_HEADER => false,
        CURLINFO_HEADER_OUT => true,
    ));

    if ($this->config['curl_cainfo'] !== false)
        curl_setopt($c, CURLOPT_CAINFO, $this->config['curl_cainfo']);

    if ($this->config['curl_capath'] !== false)
        curl_setopt($c, CURLOPT_CAPATH, $this->config['curl_capath']);

    if ($this->config['curl_proxyuserpwd'] !== false)
        curl_setopt($c, CURLOPT_PROXYUSERPWD, $this->config['curl_proxyuserpwd']);

    if ($this->config['curl_sslversion'] !== false)
        curl_setopt($c, CURLOPT_SSLVERSION, $this->config['curl_sslversion']);

    if ($this->config['is_streaming']) {
        // process the body
        $this->response['content-length'] = 0;
        curl_setopt($c, CURLOPT_TIMEOUT, 0);
        curl_setopt($c, CURLOPT_WRITEFUNCTION, array($this, 'curlWrite'));
    }

    if (! empty($this->request_settings['headers'])) {
        foreach ($this->request_settings['headers'] as $k => $v) {
            $headers[] = trim($k . ': ' . $v);
        }
        curl_setopt($c, CURLOPT_HTTPHEADER, $headers);
    }

    if (isset($this->config['block']) && (true == $this->config['block']))
        return 0;

    // do it!
    $response = curl_exec($c);
    $code = curl_getinfo($c, CURLINFO_HTTP_CODE);
    $info = curl_getinfo($c);

```

```

$error = curl_error($c);
$errorno = curl_errno($c);
curl_close($c);

// store the response
$this->response['code'] = $code;
$this->response['response'] = $response;
$this->response['info'] = $info;
$this->response['error'] = $error;
$this->response['errno'] = $errno;

if (!isset($this->response['raw'])) {
    $this->response['raw'] = '';
}
$this->response['raw'] .= $response;

return $code;
}
}

```

---

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

```

```

/**
 * CodeIgniter Twitter
 *
 * A CodeIgniter library to interact with Api Twitter.
 * Original author: elliothaughin
 *
 * @package      CodeIgniter
 * @category     Libraries
 * @author       Marcos Sanz
 * @link         https://github.com/marsanla/codeigniter-twitter
 * @license      http://www.opensource.org/licenses/mit-license.html
 */

```

```

class tweet {

    private $_oauth = NULL;

    function __construct() {
        $this->_oauth = new tweetOauth();
    }

    function __call($method, $args) {
        if (method_exists($this, $method)) {
            return call_user_func_array(array($this, $method), $args);
        }

        return call_user_func_array(array($this->_oauth, $method), $args);
    }

    function logged_in() {
        return $this->_oauth->loggedIn();
    }

    function get_header() {
        return $this->_oauth->getHeader();
    }

    function set_callback($url) {
        $this->_oauth->setCallback($url);
    }

    function login($sign_in_twitter = false) {
        return $this->_oauth->login($sign_in_twitter);
    }

    function logout() {
        return $this->_oauth->logout();
    }

    function get_tokens() {
        $tokens = array(
            'oauth_token' => $this->_oauth->getAccessToken(),
            'oauth_token_secret' => $this->_oauth->getAccessSecret()
        );

        return $tokens;
    }

    function set_tokens($tokens) {
        return $this->_oauth->setAccessTokens($tokens);
    }

}

```

```

class tweetException extends Exception {

    function __construct($string) {
        parent::__construct($string);
    }

    public function __toString() {
        return "exception " . __CLASS__ . " with message " . $this->getMessage()
            . " in " . $this->getFile() . ":" . $this->getLine() . "\nStack trace
            :\n" . $this->getTraceAsString();
    }
}

```

```

    }
}

class tweetConnection {
    // Allow multi-threading.

    private $_mch = NULL;
    private $_properties = array();

    public $header = NULL;

    function __construct() {
        $this->_mch = curl_multi_init();

        $this->_properties = array(
            'code' => CURLINFO_HTTP_CODE,
            'time' => CURLINFO_TOTAL_TIME,
            'length' => CURLINFO_CONTENT_LENGTH_DOWNLOAD,
            'type' => CURLINFO_CONTENT_TYPE
        );
    }

    private function _initConnection($url) {
        $this->_ch = curl_init($url);
        curl_setopt($this->_ch, CURLOPT_RETURNTRANSFER, TRUE);
        curl_setopt($this->_ch, CURLOPT_VERBOSE, TRUE);
        curl_setopt($this->_ch, CURLOPT_HEADER, TRUE);
    }

    public function get($url, $params) {
        if (count($params['request']) > 0) {
            $url .= '?';

            foreach ($params['request'] as $k => $v) {
                $k = str_replace(array('+', '%7E'), array(' ', '~'),
                    rawurlencode($k));
                $v = str_replace(array('+', '%7E'), array(' ', '~'),
                    rawurlencode($v));
                $url .= "{$k}={$v}&";
            }

            $url = substr($url, 0, -1);
        }

        $this->_initConnection($url);
        $response = $this->_addCurl($url, $params);

        return $response;
    }

    public function post($url, $params) {
        $post = '';

        foreach ($params['request'] as $k => $v) {
            $k = str_replace(array('+', '%7E'), array(' ', '~'), rawurlencode($k));
            $v = str_replace(array('+', '%7E'), array(' ', '~'), rawurlencode($v));
            $post .= "{$k}={$v}&";
        }

        $post = substr($post, 0, -1);

        $this->_initConnection($url, $params);
        curl_setopt($this->_ch, CURLOPT_POST, 1);
        curl_setopt($this->_ch, CURLOPT_POSTFIELDS, $post);

        $response = $this->_addCurl($url, $params);

        return $response;
    }

    private function _addOauthHeaders(&$ch, $url, $oauthHeaders) {
        $_h = array(
            'Expect:',
            'Connection: Keep-Alive',
            'Cache-Control: no-cache'
        );

        if (isset($_SERVER['HTTP_USER_AGENT'])) {
            array_push($_h, 'User-Agent: ' . $_SERVER['HTTP_USER_AGENT']);
        }

        $urlParts = parse_url($url);
        $oauth = 'Authorization: OAuth realm="' . $urlParts['path'] . '"';

        foreach ($oauthHeaders as $name => $value) {
            $oauth .= "{$name}=\"{$value}\"";
        }

        // Additional headers
        $_h[] = substr($oauth, 0, -1);

        curl_setopt($ch, CURLOPT_HTTPHEADER, $_h);
    }

    private function _addCurl($url, $params = array()) {

```

```

    if (!empty($params['oauth'])) {
        $this -> _addOAuthHeaders($this -> _ch, $url, $params['oauth']);
    }

    $ch = $this -> _ch;

    $key = (string)$ch;
    $this -> _requests[$key] = $ch;

    $response = curl_multi_add_handle($this -> _mch, $ch);

    if ($response === CURLM_OK || $response === CURLM_CALL_MULTI_PERFORM) {
        do {
            $mch = curl_multi_exec($this -> _mch, $active);
        } while ( $mch === CURLM_CALL_MULTI_PERFORM );

        return $this -> _getResponse($key);
    } else {
        return $response;
    }
}

private function _getResponse($key = NULL) {
    if ($key == NULL)
        return FALSE;

    if (isset($this -> _responses[$key])) {
        return $this -> _responses[$key];
    }

    $running = NULL;

    do {
        $response = curl_multi_exec($this -> _mch, $running_curl);

        if ($running !== NULL && $running_curl != $running) {
            $this -> _setResponse($key);

            if (isset($this -> _responses[$key])) {
                $response = new tweetResponseOAuth((object)$this ->
                    _responses[$key]);

                if (isset($response -> _resp -> header -> status)) {
                    $response -> _resp -> header -> status =
                        $response -> _resp -> code;
                    $this -> header = $response -> _resp ->
                        header;
                }

                if ($response -> _resp -> code != 200) {
                    throw new tweetException($response -> _resp
                        -> code . ' | Request Failed: ' .
                        $response -> _resp -> data -> errors[0]
                        -> message);
                }

                return $response;
            }
        }

        $running = $running_curl;
    } while ( $running_curl > 0);
}

private function _setResponse($key) {
    while ($done = curl_multi_info_read($this -> _mch)) {
        $key = (string)$done['handle'];

        $header_size = curl_getinfo($done['handle'], CURLINFO_HEADER_SIZE);
        $header = substr(curl_multi_getcontent($done['handle']), 0,
            $header_size);
        $body = substr(curl_multi_getcontent($done['handle']), $header_size);

        $this -> _responses[$key]['header'] = $header;
        $this -> _responses[$key]['data'] = $body;

        foreach ($this->_properties as $curl_key => $value) {
            $this -> _responses[$key][$curl_key] = curl_getinfo($done['
                handle'], $value);
        }

        curl_multi_remove_handle($this -> _mch, $done['handle']);
    }
}

}

class tweetResponseOAuth {
    private $__construct;

    public function __construct($resp) {
        $this -> _resp = $resp;

        if (strpos($this -> _resp -> type, 'json') !== FALSE) {
            $this -> _resp -> data = json_decode($this -> _resp -> data);
        }
    }
}

```

```

        $this -> __resp -> header = (object)$this -> _formatHeader($this ->
            __resp -> header);
    }
}

public function __get($name) {
    if ($this -> __resp -> code < 200 || $this -> __resp -> code > 299)
        return FALSE;

    if (is_string($this -> __resp -> data)) {
        parse_str($this -> __resp -> data, $result);
    } else {
        $result = $this -> __resp -> data;
    }

    foreach ($result as $k => $v) {
        $this -> $k = $v;
    }

    if ($name === '_result') {
        return $result;
    }

    return $result[$name];
}

private function _formatHeader($header) {
    $result = explode("\n", $header);

    $result_header = array();
    foreach ($result as $k => $v) {
        if ($k == 0 || trim($v) == '')
            continue;

        list($key, $value) = array_pad(explode(':', $v), 2, null);
        $result_header[strtolower(trim($key))] = trim($value);
    }

    return $result_header;
}
}

class tweetOAuth extends tweetConnection {

    private $_obj;
    private $_tokens = array();
    private $_authorizationUrl = 'http://api.twitter.com/oauth/authorize'; // Login each
        time
    private $_authenticationUrl = 'http://api.twitter.com/oauth/authenticate'; // Keep
        logged in
    private $_requestTokenUrl = 'http://api.twitter.com/oauth/request_token';
    private $_accessTokenUrl = 'http://api.twitter.com/oauth/access_token';
    private $_signatureMethod = 'HMAC-SHA1';
    private $_version = '1.0';
    private $_apiVersion = '1.1';
    private $_apiUrl = 'http://api.twitter.com';
    private $_searchUrl = 'http://search.twitter.com';
    private $_responseType = 'json';
    private $_callback = NULL;
    private $_errors = array();
    private $_enable_debug = FALSE;

    function __construct() {
        parent::__construct();

        $this -> _obj = &get_instance();
        $this -> _obj -> load -> config('tweet');
        $this -> _obj -> load -> library('session');
        $this -> _obj -> load -> library('unit_test');
        $this -> _obj -> load -> helper('url');

        $this -> _tokens = array(
            'consumer_key' => $this -> _obj -> config -> item('tweet_consumer_key'),
            'consumer_secret' => $this -> _obj -> config -> item('tweet_consumer_secret'),
            'access_key' => $this -> _getAccessKey(),
            'access_secret' => $this -> _getAccessSecret()
        );

        $this -> _checkLogin();
    }

    function __destruct() {
        if (!$this -> _enable_debug)
            return;

        if (!empty($this -> _errors)) {
            foreach ($this -> _errors as $key => $e) {
                echo '<pre>' . $e . '</pre>';
            }
        }
    }

    public function enable_debug($debug) {
        $debug = (bool)$debug;
        $this -> _enable_debug = $debug;
    }
}

```

```

public function getHeader() {
    return $this -> header;
}

public function call($method, $path, $args = NULL) {
    $response = $this -> _httpRequest(strtoupper($method), $this -> _apiUrl . '/'
        . $this -> _apiVersion . '/' . $path . '/' . $this -> _responseType, $args
    );

    // var_dump($response);
    // die();

    return ($response === NULL) ? FALSE : $response -> _result;
}

public function search($args = NULL) {
    $response = $this -> _httpRequest('GET', $this -> _searchUrl . '/search.' .
        $this -> _responseType, $args);

    return ($response === NULL) ? FALSE : $response -> _result;
}

public function loggedIn() {
    $access_key = $this -> _getAccessKey();
    $access_secret = $this -> _getAccessSecret();

    $loggedIn = FALSE;

    if ($this -> _getAccessKey() !== NULL && $this -> _getAccessSecret() !== NULL)
    {
        $loggedIn = TRUE;
    }

    $this -> _obj -> unit -> run($loggedIn, TRUE, 'Logged In');
    return $loggedIn;
}

private function _checkLogin() {
    if (isset($_GET['oauth_token'])) {
        $this -> _setAccessKey($_GET['oauth_token']);
        $token = $this -> _getAccessToken();

        $token = $token -> _result;

        $token = ( is_bool($token) ) ? $token : (object)$token;

        if (!empty($token -> oauth_token) && !empty($token ->
            oauth_token_secret)) {
            $this -> _setAccessKey($token -> oauth_token);
            $this -> _setAccessSecret($token -> oauth_token_secret);
        }

        redirect(current_url());
        return NULL;
    }
}

public function login($sign_in_twitter = false) {
    if (($this -> _getAccessKey() === NULL || $this -> _getAccessSecret() === NULL
    )) {
        header('Location: ' . $this -> _getAuthorizationUrl($sign_in_twitter));
        ;
        return;
    }

    return $this -> _checkLogin();
}

public function logout() {
    $this -> _obj -> session -> unset_userdata('twitter_oauth_tokens');
}

public function getTokens() {
    return $this -> _tokens;
}

private function _getConsumerKey() {
    return $this -> _tokens['consumer_key'];
}

private function _getConsumerSecret() {
    return $this -> _tokens['consumer_secret'];
}

public function getAccessKey() {
    return $this -> _getAccessKey();
}

private function _getAccessKey() {
    $tokens = $this -> _obj -> session -> userdata('twitter_oauth_tokens');
    return ($tokens === FALSE || !isset($tokens['access_key']) || empty($tokens[
        'access_key'])) ? NULL : $tokens['access_key'];
}

private function _setAccessKey($access_key) {
    $tokens = $this -> _obj -> session -> userdata('twitter_oauth_tokens');

    if ($tokens === FALSE || !is_array($tokens)) {

```

```

        $tokens = array('access_key' => $access_key);
    } else {
        $tokens['access_key'] = $access_key;
    }

    $this->_obj->session->set_userdata('twitter_oauth_tokens', $tokens);
}

public function getAccessSecret() {
    return $this->_getAccessSecret();
}

private function _getAccessSecret() {
    $tokens = $this->_obj->session->userdata('twitter_oauth_tokens');
    return ($tokens === FALSE || !isset($tokens['access_secret']) || empty($tokens
        ['access_secret'])) ? NULL : $tokens['access_secret'];
}

private function _setAccessSecret($access_secret) {
    $tokens = $this->_obj->session->userdata('twitter_oauth_tokens');

    if ($tokens === FALSE || !is_array($tokens)) {
        $tokens = array('access_secret' => $access_secret);
    } else {
        $tokens['access_secret'] = $access_secret;
    }

    $this->_obj->session->set_userdata('twitter_oauth_tokens', $tokens);
}

private function _setAccessTokens($tokens) {
    $this->_setAccessKey($tokens['oauth_token']);
    $this->_setAccessSecret($tokens['oauth_token_secret']);
}

public function setAccessTokens($tokens) {
    return $this->_setAccessTokens($tokens);
}

private function _getAuthorizationUrl($sign_in_twitter = false) {
    $token = $this->_getRequestToken();
    if ($sign_in_twitter) {
        return $this->_authenticationUrl . '?oauth_token=' . $token->
            oauth_token;
    }
    return $this->_authorizationUrl . '?oauth_token=' . $token->oauth_token;
}

private function _getRequestToken() {
    return $this->_HttpRequest('GET', $this->_requestTokenUrl);
}

private function _getAccessToken() {
    return $this->_HttpRequest('GET', $this->_accessTokenUrl);
}

protected function _HttpRequest($method = null, $url = null, $params = null) {
    if (empty($method) || empty($url))
        return FALSE;
    if (empty($params['oauth_signature']))
        $params = $this->_prepareParameters($method, $url, $params);

    if (isset($_REQUEST['oauth_verifier'])) {
        $params["oauth"]["oauth_verifier"] = $_REQUEST['oauth_verifier'];
    }

    $this->_connection = new tweetConnection();

    try {
        switch ($method) {
            case 'GET' :
                $response = $this->_connection->get($url, $params)
                    ;
                $this->header = $this->_connection->header;
                return $response;
                break;

            case 'POST' :
                $response = $this->_connection->post($url, $params
                    );
                $this->header = $this->_connection->header;
                return $response;
                break;

            case 'PUT' :
                return NULL;
                break;

            case 'DELETE' :
                return NULL;
                break;
        }
    } catch (tweetException $e) {
        $this->_errors[] = $e;
    }
}

private function _getCallback() {
    return $this->_callback;
}

```

```

}

public function setCallback($url) {
    $this->_callback = $url;
}

private function _prepareParameters($method = NULL, $url = NULL, $params = NULL) {
    if (empty($method) || empty($url))
        return FALSE;

    $callback = $this->_getCallback();

    if (!empty($callback)) {
        $oauth['oauth_callback'] = $callback;
    }

    $this->setCallback(NULL);

    $oauth['oauth_consumer_key'] = $this->_getConsumerKey();
    $oauth['oauth_token'] = $this->_getAccessKey();
    $oauth['oauth_nonce'] = $this->_generateNonce();
    $oauth['oauth_timestamp'] = time();
    $oauth['oauth_signature_method'] = $this->_signatureMethod;
    $oauth['oauth_version'] = $this->_version;

    array_walk($oauth, array($this, '_encode_rfc3986'));

    if (is_array($params)) {
        array_walk($params, array($this, '_encode_rfc3986'));
    }

    $encodedParams = array_merge($oauth, (array)$params);

    ksort($encodedParams);

    $oauth['oauth_signature'] = $this->_encode_rfc3986($this->
        _generateSignature($method, $url, $encodedParams));
    return array('request' => $params, 'oauth' => $oauth);
}

private function _generateNonce() {
    return md5(uniqid(rand(), TRUE));
}

private function _encode_rfc3986($string) {
    return str_ireplace(array('+', '%7E'), array(' ', '~'), rawurlencode($string));
}

private function _generateSignature($method = null, $url = null, $params = null) {
    if (empty($method) || empty($url))
        return FALSE;

    // concatenating
    $concatenatedParams = '';

    foreach ($params as $k => $v) {
        $k = $this->_encode_rfc3986($k);
        $v = $this->_encode_rfc3986($v);
        $concatenatedParams .= "{$k}={$v}&";
    }

    $concatenatedParams = $this->_encode_rfc3986(substr($concatenatedParams, 0,
        -1));

    // normalize url
    $normalizedUrl = $this->_encode_rfc3986($this->_normalizeUrl($url));
    $method = $this->_encode_rfc3986($method);
    // don't need this but why not?

    $signatureBaseString = "{$method}&{$normalizedUrl}&{$concatenatedParams}";
    return $this->_signString($signatureBaseString);
}

private function _normalizeUrl($url = NULL) {
    $urlParts = parse_url($url);

    if (!isset($urlParts['port']))
        $urlParts['port'] = 80;

    $scheme = strtolower($urlParts['scheme']);
    $host = strtolower($urlParts['host']);
    $port = intval($urlParts['port']);

    $retval = "{$scheme}://{$host}";

    if ($port > 0 && ($scheme === 'http' && $port !== 80) || ($scheme === 'https'
        && $port !== 443)) {
        $retval .= ":{port}";
    }

    $retval .= $urlParts['path'];

    if (!empty($urlParts['query'])) {
        $retval .= "?{$urlParts['query']}";
    }

    return $retval;
}

```



```

private function _signString($string) {
    $retval = FALSE;
    switch ( $this->_signatureMethod ) {
        case 'HMAC-SHA1' :
            $key = $this -> _encode_rfc3986($this -> _getConsumerSecret()
                . '&' . $this -> _encode_rfc3986($this ->
                    _getAccessSecret()));
            $retval = base64_encode(hash_hmac('sha1', $string, $key, true)
                );
            break;
        }
    return $retval;
}

}

/* End of file tweet.php */
/* Location: ./application/libraries/tweet.php */

```

---

## B.5 Layout

```

::selection{ background-color: #E13300; color: white; }
::moz-selection{ background-color: #E13300; color: white; }
::webkit-selection{ background-color: #E13300; color: white; }

body {
    background-color: #ffffff;
    margin-top: 10px;
    margin-bottom: 10px;
    margin-left: 100px;
    margin-right: 100px;
    font: 13px/20px normal Helvetica, Arial, sans-serif;
    color: #4F5155;
}

a {
    color: #003399;
    background-color: transparent;
    font-weight: normal;
}

h1 {
    color: #fff;
    background-color: transparent;
    border-bottom: 1px solid #D0D0D0;
    font-size: 19px;
    font-weight: normal;
    margin: 0 0 14px 0;
    padding: 14px 15px 10px 15px;
}

code {
    font-family: Consolas, Monaco, Courier New, Courier, monospace;
    font-size: 12px;
    background-color: #f9f9f9;
    border: 1px solid #D0D0D0;
    color: #002166;
    display: block;
    margin: 14px 0 14px 0;
    padding: 12px 10px 12px 10px;
}

#body{
    margin: 0 15px 0 15px;
}

p.footer{
    text-align: right;
    font-size: 11px;
    border-top: 1px solid #D0D0D0;
    line-height: 32px;
    padding: 0 10px 0 10px;
    margin: 20px 0 0 0;
}

#loginMainBox{
    width: 860px; /*500px*/
    margin-top: 10px;
    margin-left: 70px; /*250px*/
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    -moz-border-radius: 10px;
    border-top-left-radius: 10px 5px;
    border-top-right-radius: 10px;
}

#resetPasswordBox{
    width: 500px;
    margin-top: 10px;
    margin-left: 250px;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    -moz-border-radius: 10px;
    border-top-left-radius: 10px 5px;
    border-top-right-radius: 10px;
}

```

```

#loginBox{
    width: 300px;
    margin-left: 280px; /*90px*/
}

#logoutBox{
    width: 350px;
    margin-left: 750px;
    margin-bottom: 10px;
}

#signupBox{
    width: 400px;
    margin-left: 50px;
}

#aboutBox{
    width: 950px;
    margin-left: 0px;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
}

#container{
    width: 1023px;
    margin-top: 10px;
    margin-left: 60px;
    margin-right: 40px;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
}

#header{
    background-image: url('../layout/images/HeasFimBanner.jpg');
    height: 142px;
    width: 1000px;
    margin: 10px;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    font-family: Verdana;
}

#manageUserAccountsBox{
    width: 999px;
    margin-left: -5px;
    margin-top: 50px;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    -moz-border-radius: 10px;
    border-top-left-radius: 10px 5px;
    border-top-right-radius: 10px;
}

#manageContentBox{
    width: 999px;
    margin-left: -5px;
    margin-top: 50px;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    -moz-border-radius: 10px;
    border-top-left-radius: 10px 5px;
    border-top-right-radius: 10px;
}

#userContentBox{
    width: 999px;
    margin-left: -5px;
    margin-top: 50px;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    -moz-border-radius: 10px;
    border-top-left-radius: 10px 5px;
    border-top-right-radius: 10px;
}

#manageUserAccountsHeading{
    background-color: #CC0000;
    color: #ffffff;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
}

#manageContentHeading{
    background-color: #CC0000;
    color: #ffffff;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
}

#userContentHeading{
    background-color: #CC0000;
    color: #ffffff;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
}

```

```

#loginHeading{
    width: 860px; /*500px*/
    margin-left: 0px;
    background-color: #CC0000;
    color: #ffffff;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
}

#registerHeading{
    width: 860px; /*500px*/
    margin-left: 0px;
    background-color: #CC0000;
    color: #ffffff;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
}

#userRecipesViewHeading{
    background-color: #ffffff;
    color: #CC0000;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    /* position: absolute;
    top: 275px;
    width: 318px;*/
}

#userRecipesBox{
}

#userAchievementsViewHeading{
    background-color: #ffffff;
    color: #CC0000;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    /* position: absolute;
    top: 275px;
    width: 318px;*/
}

#userAchievementsBox{
}

#userHealthTipsViewHeading{
    background-color: #ffffff;
    color: #CC0000;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    /* position: absolute;
    top: 275px;
    width: 318px;*/
}

#userHealthTipsBox{
}

#addUserBox{
    margin-left: 10px;
}

#contentMenuBox{
    margin-left: 10px;
}

#userMenuBox{
    margin-left: 10px;
}

#viewUsersBox{
    width: 985px;
    margin-left: 5px;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
}

#viewUsersBox table{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    font-family: Verdana;
    font-size: 10pt;
    color: #000000;
}

#viewUsersBox td{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    text-align: center;
}

#viewUsersBox th{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    background-color: #CC0000;
}

```

```

        color: white;
    }

#userFeedHeading{
    width: 985px;
    margin-left: 5px;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
}

#userFeedHeading table{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    font-family: Verdana;
    font-size: 10pt;
    color: #000000;
    width: 980px;
}

#userFeedHeading td{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    text-align: center;
}

#userFeedHeading tr{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    text-align: center;
}

#userFeedHeading th{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    background-color: #CC0000;
    color: white;
}

#userViewBox{
    width: 985px;
    margin-left: 5px;
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
}

#userViewBox table{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    font-family: Verdana;
    font-size: 10pt;
    color: #000000;
    width: 980px;
}

#userViewBox td{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    text-align: center;
}

#userViewBox tr{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    text-align: center;
}

#userViewBox th{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    background-color: #CC0000;
    color: white;
}

#userSearchFELDialog table{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    font-family: Verdana;
    font-size: 10pt;
    color: #000000;
}

#userSearchFELDialog td{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    text-align: center;
}

#userSearchFELDialog th{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    background-color: #CC0000;
    color: white;
}

#viewFPDialog table{
    border: 1px solid #D0D0D0;
    -webkit-box-shadow: 0 0 8px #D0D0D0;
    font-family: Verdana;
    font-size: 10pt;
}

```

```

        color: #000000;
    }
    #viewFPDialog td{
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        text-align: center;
    }
    #viewFPDialog th{
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        background-color: #CC0000;
        color: white;
    }
    /*CREATE RECIPE GUIDELINES*/
    #healthConditionsGuidelinesModalDialog table {
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        font-family: Verdana;
        font-size: 10pt;
        color: #000000;
    }
    #healthConditionsGuidelinesModalDialog td{
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        text-align: center;
    }
    #healthConditionsGuidelinesModalDialog th{
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        background-color: #CC0000;
        color: white;
    }
    /*VIEW CREATED RECIPES*/
    #viewCreatedRecipesDialog table {
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        font-family: Verdana;
        font-size: 10pt;
        color: #000000;
    }
    #viewCreatedRecipesDialog td{
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        text-align: center;
    }
    #viewCreatedRecipesDialog th{
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        background-color: #CC0000;
        color: white;
    }
    .redbox {
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        background-color: #CC0000;
        color: white;
    }
    /*ADD MEAL*/
    /*VIEW CREATED RECIPES*/
    #addMealDialog table {
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        font-family: Verdana;
        font-size: 10pt;
        color: #000000;
    }
    #addMealDialog td{
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        text-align: center;
    }
    #addMealDialog th{
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        background-color: #CC0000;
        color: white;
    }
    #addMealButton{
        width: 120px;
    }
    /*PLAN DAILY MEALS*/
    #planDailyMealsDialog table {
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        font-family: Verdana;
        font-size: 10pt;
    }

```

```

        color: #000000;
    }
    #planDailyMealsDialog td{
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        text-align: center;
    }
    #planDailyMealsDialog th{
        border: 1px solid #D0D0D0;
        -webkit-box-shadow: 0 0 8px #D0D0D0;
        background-color: #CC0000;
        color: white;
    }
    /*BREAKFAST RECIPES*/
    #breakfast_recipes {
        border: 1px solid #D0D0D0;
    }
    #icons {
        margin: 0;
        padding: 0;
    }
    #icons li {
        margin: 2px;
        position: relative;
        padding: 4px 0;
        cursor: pointer;
        float: left;
        list-style: none;
    }
    #icons span.ui-icon {
        float: left;
        margin: 0 4px;
    }
    a.dp-choose-date {
        float: left;
        width: 16px;
        height: 16px;
        padding: 0;
        margin: 5px 3px 0;
        display: block;
        text-indent: -2000px;
        overflow: hidden;
        background: url(../layout/jquery-ui-1.10.3.custom/development-bundle/demos/
            datepicker/images/calendar.gif) no-repeat;
    }
    a.dp-choose-date.dp-disabled {
        background-position: 0 -20px;
        cursor: default;
    }
    /* makes the input field shorter once the date picker code
    * has run (to allow space for the calendar icon
    */
    input.dp-applied {
        width: 140px;
        float: left;
    }
    .verify {
        margin-top: -7px;
        margin-left: 9px;
        position: absolute;
        width: 402px;
        height: 16px;
    }
    .noclose .ui-dialog-titlebar-close {
        display: none;
    }
    .us-top-nav {
        width: auto;
        position: absolute;
        top: 175px;
        left: 893px;
    }
    .settings-person {
        float: left;
    }
    .settings-triangle {
        float: left;
    }
    .us-inner-nav {
        width: 172px;
        text-align: center;
        position: absolute;
        top: 30px;
        left: -145px;
    }
    .us-inner-nav-option {
        -webkit-transition: .2s;
        -moz-transition: .2s;
    }

```

```

        -o-transition: .2s;
        transition: .2s;
        font-weight: bold;
    }

.us-inner-nav-option:hover {
    color:#cc0000;
}
/*****/
.cm-top-nav {
    width: auto;
    position: absolute;
    top: 175px;
    left: 893px;
}

.settings-person {
    float: left;
}

.settings-triangle {
    float: left;
}

.cm-inner-nav {
    width: 172px;
    text-align: center;
    position: absolute;
    top: 30px;
    left: -210px;
}

.cm-inner-nav-option {
    -webkit-transition: .2s;
    -moz-transition: .2s;
    -o-transition: .2s;
    transition: .2s;
    font-weight: bold;
}

.cm-inner-nav-option:hover {
    color:#cc0000;
}
/*****/
.ad-top-nav {
    width: auto;
    position: absolute;
    top: 175px;
    left: 893px;
}

.settings-person {
    float: left;
}

.settings-triangle {
    float: left;
}

.ad-inner-nav {
    width: 172px;
    text-align: center;
    position: absolute;
    top: 30px;
    left: -181px;
}

.ad-inner-nav-option {
    -webkit-transition: .2s;
    -moz-transition: .2s;
    -o-transition: .2s;
    transition: .2s;
    font-weight: bold;
}

.ad-inner-nav-option:hover {
    color:#cc0000;
}

/*CONTENT MANAGER NAVIGATION*/
.content-top-nav {
    width: 200px;
    position: absolute;
    top: 175px;
    left: 170px;
}

.content-triangle {
    float: left;
}

.content-plusthick {
    float: left;
}

.content-inner-nav {
    width: 135px;
    text-align: center;
    position: absolute;
}

```

```

        top: -4px;
        left: 0px;
    }

    .content-inner-nav-option {
        -webkit-transition: .2s;
        -moz-transition: .2s;
        -o-transition: .2s;
        transition: .2s;
        font-weight: bold;
    }

    .content-inner-nav-option:hover {
        color:#cc0000;
    }

        .content-content-top-nav {
            width: 200px;
            position: absolute;
            top: 175px;
            left: 307px;
        }

    .content-content-triangle {
        float: left;
    }

    .content-content-star {
        float: left;
    }

    .content-content-inner-nav {
        width: 150px;
        text-align: center;
        position: absolute;
        top: -4px;
        left: 0px;
    }

    .content-content-inner-nav-option {
        -webkit-transition: .2s;
        -moz-transition: .2s;
        -o-transition: .2s;
        transition: .2s;
        font-weight: bold;
    }

    .content-content-inner-nav-option:hover {
        color:#cc0000;
    }

/*USER NAVIGATION*/
    .user-top-nav {
        width: 200px;
        position: absolute;
        top: 175px;
        left: 170px;
    }

    .user-triangle {
        float: left;
    }

    .user-note {
        float: left;
    }

    .user-inner-nav {
        width: 157px;
        text-align: center;
        position: absolute;
        top: -4px;
        left: 0px;
    }

    .user-inner-nav-option {
        -webkit-transition: .2s;
        -moz-transition: .2s;
        -o-transition: .2s;
        transition: .2s;
        font-weight: bold;
    }

    .user-inner-nav-option:hover {
        color:#cc0000;
    }

    .user-user-top-nav {
        width: 200px;
        position: absolute;
        top: 175px;
        left: 329px;
    }

    .user-user-triangle {
        float: left;
    }

    .user-user-clipboard {

```



```

        float: left;
    }

    .user-user-inner-nav {
        width: 147px;
        text-align: center;
        position: absolute;
        top: -4px;
        left: 0px;
    }

    .user-user-inner-nav-option {
        -webkit-transition: .2s;
        -moz-transition: .2s;
        -o-transition: .2s;
        transition: .2s;
        font-weight: bold;
    }

    .user-user-inner-nav-option:hover {
        color:#cc0000;
    }

    .user-user-user-top-nav {
        width: 200px;
        position: absolute;
        top: 175px;
        left: 478px;
    }

    .user-user-user-triangle {
        float: left;
    }

    .user-user-user-star {
        float: left;
    }

    .user-user-user-inner-nav {
        width: 183px;
        text-align: center;
        position: absolute;
        top: -4px;
        left: 0px;
    }

    .user-user-user-inner-nav-option {
        -webkit-transition: .2s;
        -moz-transition: .2s;
        -o-transition: .2s;
        transition: .2s;
        font-weight: bold;
    }

    .user-user-user-inner-nav-option:hover {
        color:#cc0000;
    }

    .user-user-user-user-top-nav {
        width: 240px;
        position: absolute;
        top: 175px;
        left: 663px;
    }

    .user-user-user-user-triangle {
        float: left;
    }

    .user-user-user-user-search {
        float: left;
    }

    .user-user-user-user-inner-nav {
        width: 228px;
        text-align: center;
        position: absolute;
        top: -4px;
        left: 0px;
    }

    .user-user-user-user-inner-nav-option {
        -webkit-transition: .2s;
        -moz-transition: .2s;
        -o-transition: .2s;
        transition: .2s;
        font-weight: bold;
    }

    .user-user-user-user-inner-nav-option:hover {
        color:#cc0000;
    }

    /*NEWSFEED CSS*/

```

---

## **XII. Acknowledgement**

Foremost, I would like to express my deepest gratitude to my thesis adviser, Ma'am Avegail D. Carpio for the guidance, patience, and motivation. My thesis would not be feasible without her. Thank you, ma'am Ave.

To UP Manila thank you so much for all the experiences that molded me into someone who can possibly stand even the strongest storm.

To my fellow Iron Man fan, you are the best! Thank you so much for everything.

To all my friends, thank you so much for all these years of friendship.

To my parents and my sister, thank you for the unending support and encouragement, for being with me through it all. I am so lucky to have you. I could not ask for more.

As a Scholastican would say after every prayer, "That in All Things, God May Be Glorified!"