A special problem in partial fulfillment
of the requirements for the degree of

Bachelor of Science in Computer Science

Submitted by:

Krizia Faith V. Ilaga

May 2016

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

<table>
<thead>
<tr>
<th>Available to the general public</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available only after consultation with author/SP adviser</td>
<td>No</td>
</tr>
<tr>
<td>Available only to those bound by confidentiality agreement</td>
<td>No</td>
</tr>
</tbody>
</table>
ACCEPTANCE SHEET

The Special Problem entitled “Anterograde Amnesia Information System” prepared and submitted by Krizia Faith V. Ilaga in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Marvin John C. Ignacio, M.Sc. (cand.)
Adviser

EXAMINERS:

Approved          Disapproved

1. Gregorio B. Baes, Ph.D. (candidate)          
2. Avegail D. Carpio, M.Sc.            
4. Perlita E. Gasmen, M.Sc. (candidate)      

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

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

Leonardo R. Estacio Jr., Ph.D.
Dean
College of Arts and Sciences
Abstract

The fast-growing modern technology is evident in various health institutions. Through Health Informatics, information technology-based innovations for delivering healthcare continues to develop. Anterograde Amnesia Information System, or simply AAIS, is no different. AAIS is a web-based application system which is designed to provide technological assistance to patients with such condition by hosting compensatory techniques that will help them in their everyday lives. This system also aims to provide a remote monitoring capacity through the report generation feature.

Keywords: anterograde amnesia, up-pgh department of rehabilitation medicine, memory test, rehabilitation system
## Contents

**Acceptance Sheet** i

**Abstract** ii

**List of Figures** v

**List of Tables** vi

### I. Introduction 1

A. Background of the Study .................................................. 1

B. Statement of the Problem .................................................. 3

C. Objectives of the Study ................................................... 4

D. Significance of the Project ............................................. 7

E. Scope and Limitations .................................................... 8

F. Assumptions ................................................................. 8

### II. Review of Related Literature 9

### III. Theoretical Framework 13

A. UP-PGH Department of Rehabilitation Medicine, Section of Occupational Therapy .................................................. 13

B. Amnesia ................................................................. 14

C. Memory Rehabilitation Techniques ..................................... 15

D. Monitoring System ....................................................... 17

E. Rehabilitation System .................................................. 17

F. Information System ..................................................... 18

G. Database Management System ......................................... 18

H. Model-View-Controller Architecture .................................. 19

### IV. Design and Implementation 20

A. Use Case Diagram ....................................................... 20
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B. Context Diagram</td>
<td>24</td>
</tr>
<tr>
<td>C. Database Design</td>
<td>25</td>
</tr>
<tr>
<td>D. System Architecture</td>
<td>30</td>
</tr>
<tr>
<td>E. Technical Architecture</td>
<td>31</td>
</tr>
<tr>
<td>V. Results</td>
<td>32</td>
</tr>
<tr>
<td>VI. Discussions</td>
<td>51</td>
</tr>
<tr>
<td>VII. Conclusions</td>
<td>53</td>
</tr>
<tr>
<td>VIII. Recommendations</td>
<td>54</td>
</tr>
<tr>
<td>IX. Bibliography</td>
<td>55</td>
</tr>
<tr>
<td>X. Appendix</td>
<td>59</td>
</tr>
<tr>
<td>A. Source Code</td>
<td>59</td>
</tr>
<tr>
<td>XI. Acknowledgement</td>
<td>137</td>
</tr>
<tr>
<td>Number</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>1</td>
<td>Use Case Diagram: Overview</td>
</tr>
<tr>
<td>2</td>
<td>Use Case Diagram: Manage Users</td>
</tr>
<tr>
<td>3</td>
<td>Use Case Diagram: Account-related Functionalities</td>
</tr>
<tr>
<td>4</td>
<td>Use Case Diagram: Activity-related Functionalities</td>
</tr>
<tr>
<td>5</td>
<td>Context Diagram</td>
</tr>
<tr>
<td>6</td>
<td>Entity Relationship Diagram</td>
</tr>
<tr>
<td>7</td>
<td>Context Diagram</td>
</tr>
<tr>
<td>8</td>
<td>Home Page</td>
</tr>
<tr>
<td>9</td>
<td>Public Registration: Entering public ID</td>
</tr>
<tr>
<td>10</td>
<td>Public Registration: Registration form</td>
</tr>
<tr>
<td>11</td>
<td>Public Registration: Confirmation message</td>
</tr>
<tr>
<td>12</td>
<td>Public Registration: Confirmation email</td>
</tr>
<tr>
<td>13</td>
<td>Login Page</td>
</tr>
<tr>
<td>14</td>
<td>Forgot password modal</td>
</tr>
<tr>
<td>15</td>
<td>Forgot password email</td>
</tr>
<tr>
<td>16</td>
<td>System Admin Dashboard</td>
</tr>
<tr>
<td>17</td>
<td>Add User: Patient</td>
</tr>
<tr>
<td>18</td>
<td>Add User: Guardian</td>
</tr>
<tr>
<td>19</td>
<td>Add User: Select therapist</td>
</tr>
<tr>
<td>20</td>
<td>Add User: Verify details page</td>
</tr>
<tr>
<td>21</td>
<td>Add User: Generated username and password</td>
</tr>
<tr>
<td>22</td>
<td>Edit User: Re-assign therapists page</td>
</tr>
<tr>
<td>23</td>
<td>Delete User: Delete patient</td>
</tr>
<tr>
<td>24</td>
<td>Delete User: Replace guardian</td>
</tr>
<tr>
<td>25</td>
<td>Delete User: Assign new therapist</td>
</tr>
<tr>
<td>26</td>
<td>Edit Account</td>
</tr>
<tr>
<td>27</td>
<td>Dashboard</td>
</tr>
<tr>
<td>28</td>
<td>Journal Entries</td>
</tr>
</tbody>
</table>
**List of Tables**

1. user table .................................................. 26
2. patient table .................................................. 26
3. journal table .................................................. 27
4. reminder table .................................................. 27
5. photo table .................................................. 28
6. question table .................................................. 28
7. test table .................................................. 29
8. submission table .............................................. 29
I. Introduction

A. Background of the Study

Memory loss is a common condition around the world and amnesia is the most well-known syndrome associated with it. [1] Amnesia is defined as the loss of explicit memories, or the memories associated with the conscious recollection of experiences, events, and facts [2], that can be caused by damage to areas of the brain that are vital for memory processing. Though it is a common plot device in movies and television that amnesia makes you lose the sense of who you are, real-life occurrences generally don’t cause loss of self-identity. [3] Amnesia victims perform extremely poor on most explicit memory tasks while their implicit memory, or the procedural memories and preserved abilities [2], is typically not impaired.

There are two kinds of amnesia: retrograde and anterograde. Retrograde amnesia works backward; it is the inability to retrieve specific set of memories that occurred before a given time. Anterograde amnesia, on the other hand, is the inability to form and retain new memories because of the failure to transfer information from short-term into long-term memory. [4]

Amnesia has an annual incidence rate, the rate of newly-diagnosed patients with amnesia in a given year, of 6.8/100,000. [5] Since anterograde amnesia, unlike a temporary episode of memory loss, can be permanent [3], this rate will just continuously add up to the total rate of patients with amnesia.

It is considered normal for individuals with amnesia to misplace things and lose their train of thought. [6] According to a 2007 study by J. Lazar where patients with amnesia were observed and interviewed, amnesiac people have issues on organization and oftentimes have much trouble remembering what information was written on which piece of paper. This leads to the missing appointments and messages which further results to their need of greater dependence on others. Observed individuals were said to have a common scenario in their places: pieces
of paper were kept all over the place, either in books, pockets, or desks. [7]

There are no specific treatments and pharmaceutical drugs for curing amnesia, but the readily available techniques for enhancing memory and psychological support can help people with amnesia, including their families and friends, cope. [3] According to psychological researches, a number of helpful memory techniques are useful in exercising the explicit memory and improving the condition of patients with anterograde amnesia. Examples of tests for exercising the explicit memory are (1) recall memory test, a measure of explicit memory that involves recalling information that has previously been remembered, and (2) multiple-choice test, an example of a recognition memory test. [4] Such tests and games are available online like the tests being hosted on http://www.psychologistworld.com/ and http://www.humanbenchmark.com/.

It has also been proven that memory tests that are less technical and more personal are important because patients need tests on personal information and past events in his personal life. [3] Furthermore, tests with the self-reference effect, or the phenomenon that things are remembered easier if they are linked to thoughts about the self, are proven effective. [4] Memory evaluation through both technical and personal tests is very important since it can help determine the extent of memory loss and provide insights about what kind of help the person may need. Patients who suffer from anterograde amnesia need continuous good monitoring and rehabilitation through proper implementation of memory techniques to help them cope with the memories they are creating and losing every day.

The UP-PGH Department of Rehabilitation Medicine (DRM) offers services to patients with cognitive problems related to memory. Patients who seek rehabilitation services are usually diagnosed already with certain conditions and are usually just referred or forwarded to the department. DRM, however, also has its own rehab diagnosis to further understand the case and specific needs of the patient. DRM follows the centralized rules of UP-PGH on admitting patients. However, in the case of patients with cognitive problems related to memory such
as amnesia, it is rare that they get confined. They only usually do when they suffer another health problem that makes them medically unstable such as stroke. Therapy sessions for these patients fall under the Section of Occupational Therapy. They usually offer memory trainings and habitual/routine trainings. For memory trainings, they perform different kinds of memory tests for patients. Furthermore, they keep records of the patient’s statements as well as his guardian’s to be used for the succeeding consultations. The frequency of consultations that a patient needs can be determined depending on the severity of his case and it is still up to the doctors and the therapists to know how much he needs. However, Jose Luis Blanco OTRP of DRM mentioned that it is rare for them to have amnesiac patients who regularly seek therapy. They often just go to the hospital for a session or two, so they expect the patient, together with his family, to continue memory exercises at home.[8]

B. Statement of the Problem

Patients need separate diaries, journals, photographs, calendars, notes and memory tests to help him remember things. Given that anterograde amnesia is lifetime, organizing the traditional way of using paper-based compensatory techniques is difficult to deal with since aside from being normally disorganized, patients with amnesia show strong symptoms of confusion and disorientation. [3]

Since it is rare for patients to seek regular therapy for whatever reason, they usually resort to do memory exercises at home. The currently available online memory tests that they can take lack monitoring capacity since after you take the test, it’s up to you to keep track of your own score. Furthermore, these tests don’t include questions on the personal level of the patients which is a vital part of memory rehabilitation as mentioned in the previous section. Because of this, the current strategy to test the patients on a personal level is through printing tests over and over again, which adds weight to the load of papers that have to be organized, or have someone to repetitively ask him verbally. Because of the
absence of a central repository in the traditional way of gathering information for the tests, someone should manually record, compile, and update the inputs of all the family and friends of the patient who wish to contribute information repeatedly.

Another problem with the traditional way of coping is the possibility of losing track of the patient’s progress due to misplacement of files and disorganization since the scores that the patient has been getting from his tests are being manually recorded by the patient himself or somebody else and is being done repetitively. Furthermore, rehabilitation centers like that of UP-PGH DRM don’t get any form of updates with their patients unless they are contacted.

C. Objectives of the Study

• General

The objective of the study is to be able to create a web-based monitoring and rehabilitation system for patients with anterograde amnesia that will host the compensatory techniques that can help him improve his condition. The system is intended to keep the records of the patients such as journal entries, memory tests, reminders, and photographs to organize them for him. Since the system is intended to be deployed online, it will (1) have a remote monitoring capacity that will allow the patient’s legal guardian and therapist to generate his technical progress report in one click and (2) lessen the need for manual intervention in terms of contributing information. Lastly, unlike the memory tests available online, tests in the system will be on the personal level of the patient since the information will come from his family, friends, or anyone he encountered given that these information are validated by the legal guardian.
• Specific

1. Allow patients to
   (a) Edit user profile and account information
   (b) Receive an e-mail upon request for password recovery
   (c) Add, edit, delete, and view journal entries according to categories
       (consisting of date, title, event details)
   (d) Add, edit, delete, and view activity reminders according to categories
       (consisting of date and time, title, and reminder details)
   (e) Add, edit, delete, and view photos according to categories (accepted file formats are jpg, gif, and png)
   (f) Take multiple choice memory tests on
       i. General facts/knowledge
       ii. Personal life events
       iii. Photo recognition
   (g) Search for journal entries, activity reminders, and photos
   (h) Generate personal progress report through graphs

2. Allow the legal guardians to
   (a) Edit user profile and account information
   (b) Receive an e-mail upon request for password recovery
   (c) Add, edit, delete, and view personal journal entries for the patient
       according to categories
   (d) Accept or reject journal entries submitted by the occupational therapist and the public
   (e) Add, edit, delete, and view activity reminders for the patient according to categories
   (f) Accept or reject activity reminders submitted by the occupational therapist and the public
(g) Add, edit, delete, and view photos for the patient according to categories

(h) Accept or reject photos submitted by the occupational therapist and the public

(i) Contribute questions to memory tests

(j) Accept or reject questions for memory tests submitted by the occupational therapist and the public

(k) Create photo questions from previously uploaded photos

(l) Search for journal entries, activity reminders, and photos

(m) Generate graphical progress report of the patient

(n) Approve or reject requests for public accounts

3. Allow the occupational therapists to

   (a) Edit user profile and account information

   (b) Receive an e-mail upon request for password recovery

   (c) Generate graphical progress report of the patients

   (d) Submit journal entries

   (e) Submit, edit, and view activity reminders

   (f) Submit photos

   (g) Submit questions for memory tests

   (h) View patient list

4. Allow the public to

   (a) Request for an account and receive an e-mail once account is approved

   (b) Edit user profile and account information

   (c) Receive an e-mail upon request for password recovery

   (d) View submitted entries

   (e) Submit journal entries
(f) Submit activity reminders
(g) Submit photos
(h) Submit questions for memory tests

5. Allow the system administrators to
   (a) Receive an e-mail upon request for password recovery
   (b) View all user accounts
   (c) Add new user accounts
   (d) Delete user accounts
   (e) Re-assign therapists for patients

D. Significance of the Project

The goal of this study is to integrate various compensatory techniques in a single system. Since it is web-based, it will lessen the organization problems when using the traditional paper way by keeping track of the objects (journal entries, reminders, photos, memory tests) through compiling them in a single system, providing a user-friendly navigation and integrating a search function. Through the repository of submissions, memory tests in the system will be on the personal level of the patient and will also lessen the need for manually recording, compiling, and updating memory tests repeatedly just to compensate for the lack of personal tests in the currently available websites. The online system will make it easier for the patient’s legal guardian and therapist to remotely monitor and track his progress through the report-generating function. This answers the problem of losing track of the patient’s progress by misplacing files when scores are being manually recorded.
E. **Scope and Limitations**

The system has the following limitations:

1. The accounts of the patient and his guardian are made during the initial consultation at the hospital.

2. The questions from the memory tests will come only from the inputs by or approved by the legal guardian.

3. The progress report will solely base on the patient’s scores on memory tests.

4. The public users will have to request for an account and be approved by the legal guardian; only by then can they submit information for the patient.

5. Upon checking the patients progress report, the therapist may opt for a therapy session via an activity reminder.

F. **Assumptions**

The system has the following assumptions:

1. The patients are legitimate people with diagnosed case of anterograde amnesia.

2. The legal guardian is legitimately connected to the patient and is appointed by the patient’s family and friends.
II. Review of Related Literature

C. Morr et. al. [9] stated in his monitoring system that the three biggest factors impacting the delivery of healthcare are (1) aging population, (2) rising labor costs, and (3) complexity of diseases. In Canada, healthcare institutions are challenged to deliver quality care with limited resources. That is why monitoring systems, powered by information technology, for different health conditions are abundant nowadays.

In 2012, J. Pak and K. Park [10] shared the idea of the importance of remote monitoring for patients. They mentioned that medication adherence is one of the most important factors in treating diseases so they proposed a medication monitoring system based on remote device management methods to monitor the medicinal intake of patients especially those with chronic diseases. Their idea is to have a central server connected to the personal medication dispensers of patients at their own homes in order to keep track of their adherence better. This system was inspired by the idea of monitoring a patient’s progress remotely so it will be easier for people around him to get a hold of how the patient is doing.

Monitoring systems can also be used to monitor daily activities. T. Tsukiyama, in 2015, [11] tried to consider problems regarding the limited number of caregivers when it comes to monitoring the condition of elderly people. He proposed a sensor-based monitoring system that evaluates the health status of solitary elderly based on daily living activities, and provides forecasts of emergency situations to a local nursing center without explicit user interaction. Thus with the three studies mentioned above, it was evident that monitoring people remotely through technology is now a widely-used practice.

Because of the continuous development in technology, different rehabilitation techniques for coping up with certain conditions are also being automated widely through the development and use of rehabilitation systems. In 2014, Y.Z. Chong et. al. [12] designed and developed a cost-effective rehabilitation system in the form of a wearable upper extremities powered exoskeleton. The system’s purposes
are to provide rehabilitation to post-stroke individuals, as well as those with related movement disorders, and just plainly help anyone to increase lifting capabilities.

Aside from the physical rehabilitation systems, there are also a lot of cognitive rehabilitation systems which is one of the main ideas of this study: to provide an online occupational therapy rehabilitation. A system was proposed by L. Jiang et. al. [13] to provide rehabilitation and training to the individuals suffering from Attention Deficit Hyperactivity Disorder (ADHD) or other similar neurological disorders. The idea of the researchers is to produce a 3D game that uses Brain-Computer Interface or BCI technology, a direct communication pathway between a brain and an external device. It aims to fulfill the objective of enhancing attention and retain the essence of effective treatment on people with ADHD. The system was tested with chosen subjects and got a positive solution so it is expected to replace or complement existing counseling methods for treating ADHD.

Both monitoring systems and rehabilitation systems are proven efficient, at least by the mentioned studies, in the context of health-related problems and situations. Patients with anterograde amnesia, as mentioned in the previous chapter, are oftentimes confused and disoriented. Furthermore, it was also stated that memory techniques are used to help patients cope up with their inability to form new memories. Given these two facts, patients with anterograde amnesia are most likely candidates to benefit from a monitoring and rehabilitation system. In fact, there are a number of systems that were already made or proposed to help people, in general, with their memory difficulties. The following related literatures are in line with the support for memory recall for different types of people and their conditions.

In 2009, E. Svoboda and B. Richards [14] proposed a newly established theory-driven training method to specifically compensate for anterograde amnesia that capitalizes on emerging smartphone technologies. Basically, the study has a 55-year old woman subject, hidden in the name RR, who undergone two phases: skill acquisition and skill generalization. In the skill acquisition phase, she was trained
in eight 1-hr sessions to use her smartphone’s calendar for reminders instead of the usual paper-based way and was successful in acquiring the skill. In the second phase, RR further generalized use of her smartphone to everyday activities and successfully acquired other skill sets in using other smartphone features such as camera, address book, and voice recorder.

A.Zmily and Dirar S. [15] said that advances in mobile phone technology have paved the way for new applications including those which are used as medical devices. They, therefore, proposed using smartphones to enhance the quality of life for individuals with Alzheimer’s disease. They developed an integrated mobile application called ADcope that aims to maximize the patients’ remaining capabilities. The application contains three modules: (1) a memory wallet that contains pictures and sentences about familiar people, places and events, (2) a calendar with reminders of all daily activities, and (3) a module that utilizes Near Field Communication (NFC) tags that are placed on various things like drawers and doors. After the initial testing, it was concluded that the benefits of using the manual methods can also be achieved using a smartphone application. The integration of all of the modules into a single application simplified the exercises for the patients.

In 2014, a mobile memory game for patients with acquired brain damage was presented by M.J. Moron et. al. [16] Instead of formulating a new game, they decided to go with the classic, well-known memory card game because (1) the game should be familiar, straightforward and easy to learn, (2) level of difficulty is relatively easy to personalize, and (3) it has high usability and accessibility. The goal of the game is to find out pairs of matching cards and the initial prototype testing was done through letting two groups of users use the Android application. The results showed several usability and accessibility flaws but were identified and corrected.

A. Molina et. al. [17] conducted a study which aims to clinically validate a new modality of cognitive rehabilitation services based on telemedicine systems
for patients with moderate to severe traumatic brain injury. The idea was to give patients computerized cognitive rehabilitation program based on telemedicine systems. It was tested on eighty patients and neuropsychological test scores were recorded before and after treatment. As a result, patients showed significant improvement after the application of the program.

A paper by C. Dobbins et. al. [18] was presented in 2013 to investigate and present DigMem, a platform for creating human digital memories using pervasive devices and linked data. These human digital memories are archived through accessing the data from a number of services, thus resulting in a vast amount of diverse information. The study offers a novel solution into building memories that are composed of a variety of information and developing a prototype to demonstrate the design. Initial results were good but there’s a room for a lot of further works like using clustering and linking techniques to enable an endless stream of memories.

An interesting study was presented by M. Hamiz et. al. [19] in 2014 called "Repetitive Memorization Mobile Application Development for Elderly Memory Recall". The inspiration behind the idea is to help the elderly Muslims to avoid frustrations when they forget the Quranic verses that they recite regularly. This is a very important practice that they do to strengthen their spiritual lives. The researchers developed a mobile application that will help them memorize these verses using the repetitive memorization technique. This technique, also known as rote learning, deals with words, not ideas and relationships, meaning that it is the ability to remember things without giving any thought to the meaning. It is also a rule-based expert system that suggests what the user need to memorize based on the usage history. It was designed as interactive and flexible to deal with the elderly better.
III. Theoretical Framework

A. UP-PGH Department of Rehabilitation Medicine, Section of Occupational Therapy

The Department of Rehabilitation Medicine (DRM) of the Philippine General Hospital (PGH) started out in 1910 as part of the Department of Radiology. In the 1970s, the remarkable growth of the DRM was noticed and was then recognized as distinct from Radiology. The DRM was recognized as a clinical department of PGH on October 28, 1971 with Dr. Guillermo Damian as its first chairman and resident. [20]

The Section of Occupational Therapy is consistent with the mission-vision of the DRM. It aims to serve as the center of excellence in providing quality service to clients by helping people help themselves. It also molds globally competitive occupational therapy professionals and implements researches that will contribute to the growth of knowledge in the field of occupational therapy and rehabilitation medicine. [21]

With their mission of helping people help themselves, the Section of Occupational Therapy offers a variety of services for patients with memory problems in general. They offer general services such as ADL/iADL or (Instrumental) Activities for Daily Living skills training to help patients cope with everyday living given their condition. Other services include environmental modification, perceptual skills training, work and productive activities, work simplification techniques, and social skills training. They also offer cognitive skills training through memory training and habitual/routine training, the usual services offered to people with amnesia.[21] Cognitive skills are the core skills that the brain uses to think, learn, remember, reason etc. which are practically impaired for patients with amnesia. [22]
B. Amnesia

The Oxford Advanced Learner’s Dictionary simply defined amnesia as a medical condition in which somebody partly or completely loses their memory. [23] Amnesia can refer to a memory loss in the absence of, or out of proportion to, other cognitive problems that the affected individual might suffer. The typical amnesiac patient is unable to recall explicit memories; these are memories for facts and events that are capable of being consciously recalled. However, patients with amnesia, even those who are severely impaired individuals, retain implicit memories; this type of memory refers to a patient’s ability to demonstrate his responses in a material that has been registered in his memory even without his knowledge that he has this information. [24]

There are several root causes for amnesia. These causes are mostly traceable to brain injuries that are related to physical trauma, disease and infection, drug and alcohol abuse, or reduced blood flow to the brain also called as vascular sufficiency. Encephalitis, herpes, and other infections that damage tissues in the brain can cause amnesia. Doctors diagnosing amnesia look at several factors to diagnose amnesia and determine its cause. They may look into the patient’s illnesses, drug and medication history and his general health. Psychological exams and imaging tests are also important to evaluate the extent of amnesia that the patient has as well as to examine the possible brain damages. [25]

Generally, there are two kinds of amnesia. One is the loss of memories preceding the onset of brain damage; this kind is called the retrograde amnesia. On the other hand, the impaired ability to remember or recall memories beginning with the onset of a condition is called anterograde amnesia. Patients with anterograde amnesia are unable to learn and remember new or recent memories. The severity of memory defect varies with the nature of the disorder. [4] It can be so extreme like in HM’s case.

Henry Gustav Molaison, or popularly known as H.M., was an American patient whose life becomes the most popular case study there is for amnesia and other
memory disorders as well as for the cognitive neuropsychology in general. He was widely studied from 1957 until his death in 2008. In his case, H.M. consistently failed to recall having eaten a meal or even to get to know the people who provided his daily care. His case of anterograde amnesia was so severe that he could only hold onto new information for only a few seconds without distractions. [26] [27]

C. Memory Rehabilitation Techniques

There are several techniques and strategies that can help restore lost memory capacity, provide compensatory techniques or teach the use of external memory aids. These techniques have been developed to enhance new learning, or at least modify behavior of patients with amnesia. [28] Many rehabilitation techniques are actually used to treat individuals who specifically suffer from anterograde amnesia. Compensatory techniques, such as computers, beepers, written notes, diaries or through intensive training programs involving the active participation of the individual, as well as his/her family and friend circle, can be the focus of interventions for the patient. In moderately or severely injured individuals, it is proven that those interventions appealing to external aids, such as reminders in order to facilitate particular knowledge or skill acquisition, are effective. Interventions can also focus on reality orientation techniques which has the purpose to enhance orientation using stimulation and repetition of the basic orientation information. [29]

The primary objective of memory therapy is to lessen the patient’s dependency on others by training him to be as independent as possible in activities of daily life. Rehabilitation does, therefore, not only focuses on the acquisition of specific knowledge but to also develop the individual’s ability to remember to do something in the future. Memory rehabilitation has strongly been influenced by memory theory, and the interaction between both has stimulated the development of techniques that partly rely on implicit memory and therefore enable even patients with dense amnesia to acquire new information. Techniques such as the
spaced-retrieval technique, which prompts recall of information over increasingly longer retention intervals, and vanishing cue methods, which involves giving individuals cues which are gradually withdrawn throughout trials until responses do not depend on cues anymore, are examples of techniques influenced by the memory theory. [28]

Memory tests vary from one type to the other and patients with anterograde amnesia usually need types of tests that focus on the following:

1. Short Term Memory - this type of test is widely available on different websites. It usually requires participants to memorize certain sequences of words and/or numbers and have him repeat them afterwards. [30]

2. General Knowledge - a test where questions are something that are more general such as the name of the current president of the country. [3]

3. Personal Life Events - this test is composed of questions on the personal level of the patient where the focus depends on the type of memory. [30]

   (a) To exercise a patient’s episodic memory, questions usually focus on the significant life events of a patient with more importance on the details.

   “Tell me about your wedding day.”

   “What happened during the time you joined the international quiz bee?”

   (b) To exercise a patient’s long term remote memory, questions are usually about what happened in a normal event relatively long period of time ago.

   “Where did you go last summer vacation?”

   “What did you get from your parents on Christmas two years ago?”

   (c) To exercise a patient’s long term recent memory, questions are of the things that happened recently.

   “What did you eat for dinner last night?”

   “What did you watch after breakfast?”
4. Photo Recognition - a test where picture of important people, places, and events are shown to the patient for recognition. [31]

D. Monitoring System

A Monitoring System is a system that allows its users to capture data, process and disseminate information in a systematic way. It provides sufficient information to the decision makers to (1) take initiatives to improve the quality of the program and (2) measure the expected objectives and outputs. To put it up shortly, monitoring system ensures that activities and objectives are being carried out on the right track by checking on them, measuring progress, identifying problems as they come up and finding strengths that can be built up. [32]

Monitoring systems is a very broad area; they can be used for monitoring disasters, risks, business processes and more. In the field of health, monitoring systems for different conditions are abundant nowadays because of the continuously increasing number of people suffering from certain diseases while the labor force assigned to give the necessary care is decreasing. Monitoring systems are, therefore, important to help the caregivers aid the patients without much manual intervention. [9]

E. Rehabilitation System

Rehabilitation engineering is the use of engineering science and principles which aims to (1) provide technological solutions to assist individuals with disabilities, and (2) aid the recovery of physical and cognitive functions lost through technologies. [33] Rehabilitation system is the software part of rehabilitation engineering. In a research conducted for developing a system for stroke patients [34], their findings were that home-based rehabilitation systems should be

- An aid to therapy
- Not specific to one ‘model of therapy’
• Flexible to individual needs
• Simple and compact to uses
• Take into account other conditions/difficulties of the patient
• Give feedback on rehabilitation performance

F. Information System

Information system is a computer science subfield that can be defined as an integrated set of components for collecting, storing, and processing data. The principle of information systems focus more on delivering information, knowledge, and digital products. [35]

In addition to the embedded role of information technology in clinical and diagnostics equipment, IS are uniquely positioned and used to capture, store, process, and communicate timely information to decision makers. These processes done by information systems aim for a better coordination of healthcare at both the individual and population levels. Information systems have great potential to reduce healthcare costs and improve outcomes. [36]

G. Database Management System

A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. In general, a DBMS manipulates the data itself, the data format, field names, record structure and file structure. Rules to validate and manipulate this data are also being defined by the DBMS. Fourth-generation query languages, such as SQL, are used along with the DBMS package to interact with a database. [37]

The DBMS essentially serves as an interface between the database and end users or application programs. It allows the end users or application programs to create, read, update and delete data in a database. It also ensures that data is consistently organized and remains easily accessible. The DBMS manages three
important things: (1) the data, (2) the database engine that allows data to be accessed and modified, and (3) the database schema, which defines the logical structure of the database. [38]

H. Model-View-Controller Architecture

The Model-View-Controller (MVC) design pattern assigns objects in an application one of three roles: model, view, or controller. The pattern defines not only the roles that objects will be assigned to do in the application, it also defines the way objects communicate with each other. Each of the three types of objects is separated from the others by abstract boundaries and communicates with objects of the other types across those boundaries. [39] The model is where the applications data objects are stored. Models are usually the one communicating with the database. The view is basically the user interface; it is what is presented to the users and how users interact with the application. The view is usually made with HTML, CSS, JavaScript and often, templates. The controller is the decision maker and it serves as the connector between the model and the view. The controller updates the view when the model changes. It also adds event listeners to the view and updates the model when the user manipulates the view. [40]
IV. Design and Implementation

A. Use Case Diagram

The system will have five types of users: the patients, the legal guardians, the occupational therapists, the system administrators, and the general public. Figure 1 shows the overview of the roles that each of them can perform. As a convention for this system, journal entries, photos, reminders and memory tests will be referred to as objects. All users who have accounts can perform specific account-related functionalities. The patient and the guardian have the privileges of performing object-related functionalities. They, together with the therapist, are eligible of generating progress report; the patient and guardian can only generate the report of the patient himself while the therapist can view progress reports of all of his patients. The therapist and the public, as discussed in the previous chapters, can submit objects subject to the approval of the guardian. Lastly, the system administrator is the one in charge for managing user accounts.
Figure 2 shows the role of the system administrator for managing user accounts. All user accounts are made through the system administrator. During the initial visit to the hospital, accounts will be made both for the patient and his guardian. They will be given temporary usernames and passwords that they can change later on. System admins are also responsible for deleting user accounts, re-assigning therapists.
When the users log in, they can edit their accounts immediately to change their password into something easier to remember than the randomly generated string assigned to them. This happens by updating their profiles. Figure 3 shows all other account-related functionalities for the users. They are all capable of requesting and receiving an e-mail for forgotten passwords upon request. Each of them are capable of viewing accounts of a specific set of people: the patient and the guardian can only view each others account plus the therapist assigned to the patient, the therapist can view all of his patients accounts, the public can view only his account, and the system administrator can view all accounts. The public users account is requested through the patients public ID and the guardian has the power to accept or reject this request.
The patient can add, edit, delete, and view objects, which were either made by or for him, except for memory tests. He can view these objects according to filters or categories. He can also search for these objects through keywords. The patient can take memory tests of different types at most once a day for each. Just the same as the patient, the legal guardian can add, edit, delete and view objects for the patient and search for these objects. He can contribute questions for the memory tests of the patient and even create photo questions from the readily available photos in the system. The legal guardian also has the power to accept or reject submissions of objects by the public and the therapist for the patient.
B. Context Diagram

Figure 2 shows the context diagram of the system. This shows the limitations and restrictions for each type of user that can interact with the system. The five types of users which are the patient, the legal guardian, the occupational therapist, the public and the system administrator are restricted to the things that they can input to the system and is limited to the outputs that they are allowed to get from the system.

![Figure 5: Context Diagram](image)
C. Database Design

The system will be database-dependent since it is intended to keep records. Therefore, a good database design is necessary to better fulfill the system’s objectives. Figure 3 shows the Entity Relationship Diagram for the database design of the system followed by the tables for Data Dictionary.

Figure 6: Entity Relationship Diagram
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_id</td>
<td>int(11)</td>
<td>unique id of the user</td>
</tr>
<tr>
<td>username</td>
<td>varchar(20)</td>
<td>username of the user</td>
</tr>
<tr>
<td>password</td>
<td>varchar(50)</td>
<td>password of the user</td>
</tr>
<tr>
<td>last_name</td>
<td>varchar(50)</td>
<td>last name of the user</td>
</tr>
<tr>
<td>first_name</td>
<td>varchar(50)</td>
<td>first name of the user</td>
</tr>
<tr>
<td>sex</td>
<td>varchar(1)</td>
<td>sex of the user: M (male) or F (female)</td>
</tr>
<tr>
<td>birthday</td>
<td>date</td>
<td>birthday of the user</td>
</tr>
<tr>
<td>email</td>
<td>varchar(50)</td>
<td>email address of the user</td>
</tr>
<tr>
<td>user_type</td>
<td>tinyint(1)</td>
<td>type of user; 0-admin, 1-patient, 2-guardian, 3-therapist</td>
</tr>
<tr>
<td>profile_picture</td>
<td>varchar(255)</td>
<td>filename of the user’s profile picture</td>
</tr>
</tbody>
</table>

Table 1: user table

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_id</td>
<td>int(11)</td>
<td>user id of the patient</td>
</tr>
<tr>
<td>public_id</td>
<td>int(11)</td>
<td>patient id for submissions</td>
</tr>
<tr>
<td>guardian_id</td>
<td>int(11)</td>
<td>user id of the patient’s guardian</td>
</tr>
<tr>
<td>therapist_id</td>
<td>int(11)</td>
<td>user id of the patient’s guardian</td>
</tr>
<tr>
<td>request</td>
<td>tinyint(1)</td>
<td>new public id request; 0-no request, 1-request</td>
</tr>
</tbody>
</table>

Table 2: patient table
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>journal_id</td>
<td>int(11)</td>
<td>unique id of the journal entry</td>
</tr>
<tr>
<td>user_id</td>
<td>int(11)</td>
<td>user id of the patient</td>
</tr>
<tr>
<td>author_id</td>
<td>int(11)</td>
<td>user id of the author; 0 if from public</td>
</tr>
<tr>
<td>author</td>
<td>varchar(255)</td>
<td>name of the author</td>
</tr>
<tr>
<td>date</td>
<td>date</td>
<td>date of the event</td>
</tr>
<tr>
<td>title</td>
<td>char(100)</td>
<td>title of the event</td>
</tr>
<tr>
<td>details</td>
<td>longtext</td>
<td>narrative description of the event</td>
</tr>
<tr>
<td>status</td>
<td>tinyint(1)</td>
<td>status of the entry; 0-pending, 1-accepted</td>
</tr>
<tr>
<td>date_created</td>
<td>date</td>
<td>date the entry was created</td>
</tr>
</tbody>
</table>

Table 3: journal table

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reminder_id</td>
<td>int(11)</td>
<td>unique id of the reminder</td>
</tr>
<tr>
<td>user_id</td>
<td>int(11)</td>
<td>user id of the patient</td>
</tr>
<tr>
<td>author_id</td>
<td>int(11)</td>
<td>user id of the author; 0 if from public</td>
</tr>
<tr>
<td>author</td>
<td>varchar(255)</td>
<td>name of the author</td>
</tr>
<tr>
<td>date_time</td>
<td>datetime</td>
<td>date/time of when to remind</td>
</tr>
<tr>
<td>title</td>
<td>varchar(50)</td>
<td>title of the reminder</td>
</tr>
<tr>
<td>details</td>
<td>longtext</td>
<td>details of the reminder</td>
</tr>
<tr>
<td>status</td>
<td>tinyint(1)</td>
<td>status of the entry; 0-pending, 1-accepted</td>
</tr>
<tr>
<td>date_created</td>
<td>date</td>
<td>date the entry was created</td>
</tr>
</tbody>
</table>

Table 4: reminder table
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>photo_id</td>
<td>int(11)</td>
<td>unique id of the photo</td>
</tr>
<tr>
<td>user_id</td>
<td>int(11)</td>
<td>user id of the patient</td>
</tr>
<tr>
<td>author_id</td>
<td>int(11)</td>
<td>user id of the author; 0 if from public</td>
</tr>
<tr>
<td>author</td>
<td>varchar(255)</td>
<td>name of the author</td>
</tr>
<tr>
<td>filename</td>
<td>varchar(20)</td>
<td>filename of the photo</td>
</tr>
<tr>
<td>date</td>
<td>date</td>
<td>date the photo was taken</td>
</tr>
<tr>
<td>caption</td>
<td>longtext</td>
<td>description of the photo</td>
</tr>
<tr>
<td>status</td>
<td>tinyint(1)</td>
<td>status of the entry; 0-pending, 1-accepted</td>
</tr>
<tr>
<td>date_created</td>
<td>date</td>
<td>date the entry was created</td>
</tr>
</tbody>
</table>

Table 5: photo table

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>question_id</td>
<td>int(11)</td>
<td>unique id of the question</td>
</tr>
<tr>
<td>user_id</td>
<td>int(11)</td>
<td>user id of the patient</td>
</tr>
<tr>
<td>author_id</td>
<td>int(11)</td>
<td>user id of the author; 0 if from public</td>
</tr>
<tr>
<td>author</td>
<td>varchar(255)</td>
<td>name of the author</td>
</tr>
<tr>
<td>question_details</td>
<td>longtext</td>
<td>the question itself</td>
</tr>
<tr>
<td>filename</td>
<td>varchar(20)</td>
<td>filename of photo(if any); 0 if none</td>
</tr>
<tr>
<td>choiceA</td>
<td>varchar(255)</td>
<td>multiple choice answer A</td>
</tr>
<tr>
<td>choiceB</td>
<td>varchar(255)</td>
<td>multiple choice answer B</td>
</tr>
<tr>
<td>choiceC</td>
<td>varchar(255)</td>
<td>multiple choice answer C</td>
</tr>
<tr>
<td>choiceD</td>
<td>varchar(255)</td>
<td>multiple choice answer D</td>
</tr>
<tr>
<td>answer</td>
<td>varchar(50)</td>
<td>correct answer</td>
</tr>
<tr>
<td>type</td>
<td>tinyint(1)</td>
<td>1-personal, 2-general knowledge</td>
</tr>
<tr>
<td>status</td>
<td>tinyint(1)</td>
<td>status of the entry; 0-pending, 1-accepted</td>
</tr>
<tr>
<td>date_created</td>
<td>date</td>
<td>date the entry was created</td>
</tr>
</tbody>
</table>

Table 6: question table
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>test_id</td>
<td>int(11)</td>
<td>unique id of the test</td>
</tr>
<tr>
<td>user_id</td>
<td>int(11)</td>
<td>user id of the patient</td>
</tr>
<tr>
<td>test_type</td>
<td>tinyint(1)</td>
<td>type of the test; 1 - personal, 2 - photo recognition, 3 - general knowledge</td>
</tr>
<tr>
<td>date</td>
<td>date</td>
<td>date taken</td>
</tr>
<tr>
<td>items</td>
<td>tinyint(2)</td>
<td>number of items</td>
</tr>
<tr>
<td>score</td>
<td>tinyint(2)</td>
<td>score</td>
</tr>
</tbody>
</table>

Table 7: test table

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>submission_id</td>
<td>int(11)</td>
<td>unique id of the submission</td>
</tr>
<tr>
<td>user_id</td>
<td>int(11)</td>
<td>user id of the patient</td>
</tr>
<tr>
<td>submission_type</td>
<td>tinyint(1)</td>
<td>type of submission; 1-journal, 2-reminder, 3-photo, 4-question</td>
</tr>
<tr>
<td>submission_type_id</td>
<td>int(11)</td>
<td>id of the entry in its respected table</td>
</tr>
<tr>
<td>date</td>
<td>date</td>
<td>date taken</td>
</tr>
</tbody>
</table>

Table 8: submission table
D. System Architecture

The system will be implemented in PHP programming language using the Model-View-Controller architecture through the CodeIgniter framework. The front end will be implemented in HTML, CSS, JavaScript and jQuery using the Bootstrap framework. For the database, the system will be using MySQL. Figure 4 shows the visual representation of the system architecture.

![System Architecture Diagram](image)

Figure 7: Context Diagram
E. Technical Architecture

The minimum requirements for the server machine include:

- PHP version 5.6
- MySQL version 5
- Apache version 2.4
- 1 GB free disk space
- 2.4 MHz processor
- 500MB RAM

The minimum requirements for the client include:

- 1 GB RAM
- JavaScript-enabled browser
V. Results

1. General Public View

![Figure 8: Home Page](image)

Figure 8 shows the home page of the system. It gives a brief description about the features of the website.

![Figure 9: Public Registration: Entering public ID](image)

For public registration, they should get a hold of the patient’s public ID. Figure 9 shows the page where the public ID should be entered to proceed.
Figure 10: Public Registration: Registration form

Figure 11: Public Registration: Confirmation message
Figure 10 shows the form for account registration. When account information has no input errors, a confirmation message will appear as shown in Figure 11. Figure 12 shows the e-mail that will be received by the user once his account was approved by the legal guardian.

Figure 12: Public Registration: Confirmation email

Figure 13: Login Page
At the login tab in the home page, a user will be asked for his username and password as shown in Figure 13. Figure 14 shows the modal that will ask for your username for recovering a forgotten password. An email will be sent to the email address of the username entered like the e-mail message in Figure 15.

Figure 14: Forgot password modal

Hello. This your account information:

Username: kilaga
Password: helloworld

Please delete this e-mail after taking note of your account information for security purposes. You might also want to change your password.

Figure 15: Forgot password email
2. System Administrator View

![System Admin Dashboard](image1)

**Figure 16: System Admin Dashboard**

The system admin’s dashboard shows the users overview as seen in Figure 16.

![Add User: Patient](image2)

**Figure 17: Add User: Patient**
Figure 18: Add User: Guardian

Figure 19: Add User: Select therapist
Figures 17-21 show the process of adding new users. This is specifically for adding patient and guardian which always come in pair. Figure 17 shows the adding of patient information, this will be followed by the guardian registration form in Figure 18. In Figure 19, a therapist will be selected to be assigned to the patient. The system allows the admin to verify the details entered as shown in Figure 20. Figure 21 shows the display of the auto-generated usernames and passwords for the new accounts. Adding therapist and admin accounts follow the same process.
Figure 22 shows the list of all of the patients with the Assign option to re-assign new therapists for them.

Figure 23: Delete User: Delete patient
Figure 23 shows the page where patients can be deleted. This is the same as those for other types of users. However, deleting a certain type of user leads to another process. As seen in Figure 24, deleting a guardian requires the admin to create a new guardian account for the patient. Figure 25 happens when deleting a therapist account, the admin must re-assign all his patients to other therapists.
Figure 26: Edit Account

Figure 26 shows the edit account page which is similar to all the other users’ edit account page.

3. Patient View

Figure 27: Dashboard

Figure 27 shows the patient’s dashboard where navigation guide and activity reminders are displayed. This is the same for the guardians and therapists.
The dashboard-like design for viewing journal entries can be seen in Figure 28, this is similar to the other types of entries. The entries can be filtered according to the entered category in the box at the right side. Figure 29 shows the edit entry page.
Figure 30: Create entries

This page is where entries are being created for the patient. Figure 30 is also available in the therapists’ and guardians’ accounts.

Figure 31: Memory Test Home Page
With memory tests being one of the main features, Figures 31-33 show the process of taking one. Figure 31 shows the introduction page, Figure 32 shows the test proper and Figure 33 shows the results page where the patients can review their answers.
Figure 34 shows the progress report generation through graphs given the specified span of time.

Figure 35: Search Results

The search function is also available for the guardians. This displays all entries matching the keyword/s entered as shown in Figure 35.
4. Guardian View

A lot of functionalities and/or pages in the guardian view are similar to that of the patient’s. This section will cover only the functionalities unique to the guardians.

Figure 36: Create Photo Questions

Figure 36 shows the role of the guardian to create new photo questions from old photos which are previously uploaded in the system.

Figure 37: Submissions

Figure 37 shows the role of the guardian to accept or reject submissions.
Figure 37 shows the Submissions page where the guardian accepts and/or rejects entries for the patient.

![Figure 37: Submissions Page](image1)

Figure 38: Account Requests

One of the guardian’s function is to accept and/or reject public accounts. Figure 38 shows the requested accounts page while Figure 39 shows all the accepted public accounts linked to the patient.

![Figure 39: Manage Public Users](image2)

Figure 39: Manage Public Users
5. Therapist View

Some functionalities and/or pages in the therapist view were also already discussed previously. This section will cover only the functionalities unique to the therapists.

![Figure 40: Patients List](image)

Figure 40: Patients List

Figure 40 shows the patients list for the therapist.

![Figure 41: Therapist: Reminders page](image)

Figure 41: Therapist: Reminders page

As seen in Figure 41, therapists can view all their scheduled appointments with their patients in a page, both the pending and confirmed appointments.
In the therapist view, the page where progress reports are generated include an additional field, the patient. Figure 42 shows how a therapist can view the progress reports of all of his patients.

6. Public User View

The public users have only a very limited number of functionalities. Aside from the account-related functions (edit profile, forgot password, etc.), here are the remaining functionalities that a public user can perform.

Figure 42: Therapist: Generate Report

Figure 43: Public User Dashboard
Figure 43 shows the public user’s dashboard. Here, he can see all the entries that he submitted for the patient as well as the status of the submission (pending or confirmed, and cancelled for activity reminders).

Figure 44: Submit Entries

Figure 44 shows the submit entries page for the public, this is just the same as the one in the patient view.
VI. Discussions

The Anterograde Amnesia Information System (AAIS) is a web-based application system for patients with such condition that hosts the compensatory techniques that can help them improve their condition. The system has four main features: (1) it keeps all records of journal entries, activity reminders, photographs, and memory tests for the patient in an organized manner, (2) it allows people who know the patient contribute information, (3) it hosts memory tests for the online rehabilitation of the patient, and (4) it has a remote monitoring capacity that will allow the patients legal guardian and therapist to generate his technical progress report.

AAIS offers its services to five types of user including the general public. The first type of user is the system administrator. The system administrators job is basically to manage the other users in the system. It is through the system administrators account that patients and guardians accounts are created during the initial visit to the hospital. He is in charge for creating all user accounts as well as deleting them except for the public accounts. He also has an additional task of re-assigning therapists to patients in case of some changes.

The roles of the patient and the guardian are almost the same. They can both add, edit, delete, and search for entries. Both types of user can also generate the graphical progress report of the patient. Though they may seem to have the same functionalities, they also differ on some roles. While the patient can take memory tests, the guardian can add questions for the patients memory test. The guardian is also the one responsible for accepting and/or rejecting the entries submitted by the fourth and fifth user which are the therapists and the public, respectively. The therapists can also view his patients progress reports. He can also add, edit, delete, and view his own reminders (usually appointments with his patients). Both the therapists and the public can submit entries for the patient that will be subjected to the guardians approval. While the therapist can submit entries for any of his patients, anyone from the public can only submit entries once his account was
approved by the guardian.

The system uses various technologies in order to ensure that the objectives of
the study are properly met and achieved. The whole application has, mainly, a
PHP back end with CodeIgniter used as the framework for an efficient Model-
View-Controller implementation. It is important to take note that MVC provides
a smoother relationships between the functionalities in a system thus making
the code more flexible and easier to edit. CodeIgniter, however, is not a perfect
framework. Issues like the lack of bug fixing feature happened while developing
the system but this was balanced out by the fact that using a framework, in
general, with MVC architecture makes it a lot easier for the developer to make
changes. For the front end, Bootstrap, a framework that displays a simple and
user-friendly interface, is used. The simplicity of Bootstrap is actually relevant
so patients won't have to deal with the burden of complex interfaces. Bootstrap
integrates HTML, CSS, Javascript, and jQuery files to produce an aesthetically
appealing look. MySQL is used as the database for the system simply because it
has a popular good reputation and it is easy to use and play with. The Google
Chart API was used for generating charts in presenting the progress report of
the patients. As an additional feature, encrypted cookies using OpenSSL were
used throughout the system instead of sessions for better security. Though there
were some minor difficulties with using the mentioned technologies, none of them
became a hindrance in completing the objectives of the study.
VII. Conclusions

Through the different frameworks, libraries, and technologies used, the system was properly developed to meet the objectives of the study. The system has five types of user namely, (1) the patients and (2) their guardians, (3) the therapists, (4) the system administrators, and (5) the public. The patients, guardians, therapists, and the public, given the proper permission, can be able to create entries (journal entries, activity reminders, photos, and memory test questions) for the patients. Through this record-keeping feature of the system, problems with disorganization for the patients can be lessen as compared to the traditional paper-based way of keeping records.

The currently available online memory tests lack two things: (1) personal-level tests, and (2) monitoring capacity. Since memory test questions are created and/or submitted by the people who actually know the patient, the questions are on the patient’s personal level. This means that questions that will be asked are about his life, experiences, relevant people, places, etc. The system also keeps track of the scores that the patient gets from his memory test for monitoring purposes which are the progress reports. These reports allow the legal guardians and therapists to generate the technical progress report of a patient based on his performance on the memory tests given a specific period of time. The reports are in a graphical form for easier and faster understanding. The system is made for online use; that is why remote monitoring and contributing information is now possible hence lessening the need for manual intervention.

The proposed objectives of the system is intended to attain the goal of helping patients with anterograde amnesia cope in their everyday lives. With the functionalities successfully developed, the system can now be considered as ready for deployment.
VIII. Recommendations

The Anterograde Amnesia Information System was developed to aid patients with such condition in living their lives as normal as possible. The system was built to, as what the Section of Occupational Therapy of the Department of Rehabilitation Medicine in UP-PGH would say, “help people help themselves.” Though all the major objectives in this study were achieved, there are still a lot of improvements that can make the system better in order to help the people better.

Probably the most important recommendation is to make a mobile app clone of this web-based system so the users can more conveniently use it anywhere. However, future developers who plan to do this should take into consideration a lot of limitations that mobile phones have: (1) small screen display which can be hard especially for patients with memory problems and cognitive impairment, and (2) small memory which is a pretty huge problem given that the application keeps hold of possibly lifetime records.

In terms of functionalities, the memory test part has a huge room for improvement. Future developers should think more creatively and create wider variations of these tests. It is recommended to explore other types of test aside from multiple choice. It will also be good to have short term memory tests be included in the system with supporting details such as timers to make the tests a bit more fun and creative. The progress reports can also be improved in a way where the patient’s progress will be presented through more than just line graphs. Overall, there’s still a lot of room for improvement in the system that future developers can explore and try to meet.
IX. Bibliography


X. Appendix

A. Source Code

Listing 1: Public_Controller.php

```php
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Public_Controller extends CI_Controller
{
    private $publicCookie;
    private $userCookie;

    public function __construct()
    {
        date_default_timezone_set('Asia/Manila');

        if( $this->AAIS_model->hasCookie('thePublicCookie') ) {
            $this->publicCookie = $this->AAIS_model->getCookie('thePublicCookie');
        }
        if( $this->AAIS_model->hasCookie('theUserCookie') ) {
            $this->userCookie = $this->AAIS_model->getCookie('theUserCookie');
        }
    }

    public function index()
    {
        if($this->AAIS_model->hasCookie('thePublicCookie')){
            if($this->publicCookie->error == 2)
                echo "<script>alert('Patient ID does not exist.')</script>
            ;

            $this->publicCookie = $this->AAIS_model->deleteCookie('thePublicCookie');
        }
        if($this->AAIS_model->hasCookie('theUserCookie')){
            if($this->userCookie->error == 1)
                echo "<script>alert('Username-Password combination does not exist.')</script>
            ;

            $this->AAIS_model->deleteCookie('theUserCookie');
        }
    }

    //sysad cookies
    if($this->AAIS_model->hasCookie('addAccountCookie')){
        $this->AAIS_model->deleteCookie('addAccountCookie');
    }

    //patient/guardian cookies
    if($this->AAIS_model->hasCookie('theEntryCookie')){
        $this->AAIS_model->deleteCookie('theEntryCookie');
    }
    if($this->AAIS_model->hasCookie('theSubmissionCookie')){
        $this->AAIS_model->deleteCookie('theSubmissionCookie');
    }
    if($this->AAIS_model->hasCookie('theSearchCookie')){
        $this->AAIS_model->deleteCookie('theSearchCookie');
    }

    $data["content"] = "public/main.php";
    $this->load->view('main.php', $data);
    //this->load->view( 'main.php', $data);
    //this->load->view( 'main.php', $data);
}

public function loginPage()
{
    $this->load->view('public/login.php');
    $this->load->view('public/main.php');
    $this->load->view('public/dashboard.php');
}

public function login()
{
    $userType = $this->AAIS_model->login();
    if($userType==0){
        redirect(base_url('sysad/dashboard'));
    } else if($userType==1){
        redirect(base_url('patient/dashboard'));
    } else if($userType==2){
        redirect(base_url('guardian/dashboard'));
    } else if($userType==3){
        redirect(base_url('ot/dashboard'));
    } else if($userType==4){
        redirect(base_url('publicUser/dashboard'));
    }

    public function publicPanel()
    {
        if(isset($_POST['back']))
            redirect(base_url());
        $data["content"] = "public/public.php"
        $data["cookie"] = $this->publicCookie;
        $data["patients"] = $this->AAIS_model->getPatientList();
        $this->load->view('main.php', $data);
    }

    public function submitAccount()
    {
        if(!isset($_FILES["picture"]['name'])){
            $this->publicCookie->error = 1;
            $this->AAIS_model->setCookie('thePublicCookie', $this->publicCookie); redirect(base_url('public'));
        }
    }
}
```

59
Listing 2: SystemAdmin_Controller.php

```php
<?php
defined( 'BASEPATH' ) OR exit( 'No direct script access allowed' );
class SystemAdmin_Controller extends CI_Controller
{
    private $userCookie;
    private $addAccountCookie;
    private $searchResults;
    private $patient;

    public function __construct()
    {
        date_default_timezone_set( 'Asia/Manila' );
        if (isset( $_POST[ 'logout' ]))
        {
            $this->redirect( base_url() );
        }
        if ($this->form_validation->run() == FALSE)
        {
            $this->publicCookie->error = 1;
            $this->AAIS_model->setCookie( (array) $this->publicCookie , ' thePublicCookie' );
        }
        else
        {
            if ($this->AAIS_model->dateChecker( $_POST[ 'birthday' ]) == 1){
                $this->AAIS_model->error = 2;
                $this->AAIS_model->setCookie( (array) $this->publicCookie , ' thePublicCookie' );
                redirect( base_url( 'public' ) );
            }
            $date = date( "YmdHis" );
            $filename = $this->AAIS_model->moveUpUpload( 'picture' , $date_time);
            $this->AAIS_model->addPublic( $_POST[ 'firstName' ] , $_POST[ 'lastName' ] , $_POST[ 'sex' ] ,
            $_POST[ 'birthday' ] , $_POST[ 'email' ] , $filename , $_POST[ 'note' ]);
            $this->publicCookie->error = 3;
            $this->AAIS_model->setCookie( (array) $this->publicCookie , ' thePublicCookie' );
            redirect( base_url( 'public' ) );
        }

    public function newSubmission()
    {
        $this->publicCookie->step = 1;
        $this->AAIS_model->setCookie( (array) $this->publicCookie , ' thePublicCookie' );
        redirect( base_url( 'public' ) );
    }

    public function forgotPassword()
    {
        $this->AAIS_model->sendPassword($_POST[ 'username' ]);
        redirect( base_url( 'public' ) );
    }
}
```

if (isset( $_POST[ 'logout' ]))
{
    $this->redirect( $base_url( ) );
}
```
POST [ ' birthday ' ];

Name ' , ' required ' ) ;

POST [ ' therapist ' ];

model

POST [ ' email ' ];

model

POST) )

i d ) ;

POST

i d

POST [ ' firstName ' ];

POST [ ' email ' ];

model

public function addUserStepThree()
{
    $temp = $this->userCookie->step;
    $this->userCookie->setStep = 3;
    $this->AAIS_model->setCookie( ( array ) $this->userCookie , ' theUserCookie ' );
    if($temp==2){
        $this->form_validation->run () == FALSE || $this->AAIS_model->dateChecker( $POST['birthday '] )===false{
            $this->userCookie->error = 1;
            $this->AAIS_model->setCookie( ( array ) $this->userCookie , ' theUserCookie ' );
            redirect( base_url('sysad/addUserPanel') ) ;
        }else{
            $this->addAccountCookie->guardianFirst = $POST['firstName ' ];
            $this->addAccountCookie->guardianLast = $POST ['lastName ' ];
            $this->addAccountCookie->guardianSex = $POST [ ' sex ' ];
            $this->addAccountCookie->guardianBirthday = $POST['birthday '] ;
            $this->addAccountCookie->guardianEmail = $POST['email ' ];
            $this->AAIS_model->setCookie( ( array ) $this->addAccountCookie , ' addAccountCookie ' );
            redirect( base_url('sysad/addUserPanel') ) ;
        }
    }

    public function addUserStepFour()
{
    $temp = $this->userCookie->step;
    $this->userCookie->setStep = 4;
    $this->AAIS_model->setCookie( ( array ) $this->userCookie , ' theUserCookie ' );
    if($temp==3){
            $this->addAccountCookie->set_id = $POST['therapist ' ];
            $this->AAIS_model->setCookie( ( array ) $this->addAccountCookie , ' addAccountCookie ' ) ;
            redirect( base_url('sysad/addUserPanel') ) ;
        }else{
            $this->userCookie->setStep = 4;
            $this->AAIS_model->createAccount()
            redirect( base_url('sysad/addUserPanel') ) ;
        }
    }

    public function addUserStepFive()
{
    $this->userCookie->setStep = 5;
    $this->AAIS_model->setCookie( ( array ) $this->userCookie , ' theUserCookie ' );
    $this->AAIS_model->createAccount()
    redirect( base_url('sysad/addUserPanel') ) ;
}

    public function addUserPanel()
{
    $data['cookie'] = $this->userCookie;
    $data['addCookie'] = $this->userCookie;
    if($this->userCookie->step==3)
        $data['therapist'] = $this->AAIS_model->getTherapists();
    else if($this->userCookie->step==4)
        $data['therapist'] = $this->AAIS_model->getUser($this->userCookie->set_id);  
    else if($this->userCookie->step==5)
        $this->AAIS_model->deleteCookie( ' addAccountCookie ' ) ;
    $data['content'] = ' sysad/addUser/ main.php ' ;
    $this->load->view( 'home.php ' , $data ) ;
}

    public function addOT()
{
    $this->userCookie->active = "addOT" ;
    $this->AAIS_model->setCookie( ( array ) $this->userCookie , ' theUserCookie ' ) ;
    $data['cookie'] = $this->userCookie;
    $data['content'] = ' sysad/addOT/ main.php ' ;
    $this->load->view( 'home.php ' , $data ) ;
}

    public function addOTStepTwo()
{
    if(isset($POST)){
        $this->load->library( " form_validation " );
        $this->form_validation->set_rules( ' firstName ' , ' First Name ' , ' required ' );
        $this->form_validation->set_rules( ' lastName ' , ' Last Name ' , ' required ' );
        $this->form_validation->set_rules( ' sex ' , ' Sex ' , ' required ' );
        $this->form_validation->set_rules( ' email ' , ' Email ' , ' required|valid_email ' );

        if($this->form_validation->run () == FALSE || $this->
AAIS_model->dateChecker( $POST['birthday']=='1');
if ($this->userCookie->error == 1)
    $this->userCookie->error = 1;
$this->AAIS_model->
    setCookie( array
        $this->userCookie, 'theUserCookie');
    redirect( base_url('sysad/addSyad'));
    return
} else
    $data["details"] = $this->AAIS_model->insertOT( $POST['firstName'], $POST['lastName'], $POST['sex'], $POST['birthday'], $POST['email']);
    $this->userCookie;
    $data["content"] = 'sysad/editAccount';
    $data["content"] = 'sysad/editAccount';
    $this->load->view( 'home.php', $data);

public function addSysadStepTwo()
{
    if(isset($POST))
    {
        $this->load->library( 'form_validation');
        $this->form_validation->
            set_rules( 'firstName', 'First_Name', 'required');
        $this->form_validation->
            set_rules( 'lastName', 'Last_Name', 'required');
        $this->form_validation->
            set_rules( 'sex', 'Sex', 'required');
        $this->form_validation->
            set_rules( 'birthday', 'Birthday', 'required');
        $this->form_validation->
            set_rules( 'email', 'Email', 'required');
        $this->form_validation->
            set_rules( 'password', 'Password', 'required');
        if(!is_array( $FILES['dp'])['name'] == '')
        {
            $filename = $this->
                userCookie->
                profile_picture;
        } else
        {
            $date_time = date( 'YmdHis');
            $filename = $this->
                AAIS_model->
                moveDpUpload( 'dp', $date_time);
        }
    if($this->form_validation->run () == FALSE || $this->
        AAIS_model->dateChecker( $POST['birthday']=='1')
        $this->userCookie->error = 1;
    $this->AAIS_model->
        setCookie( array
            $this->userCookie, 'theUserCookie');
        redirect( base_url('sysad/addSyad'));
    } else
    {
        $data["details"] = $this->AAIS_model->
            insertSyad( $POST['firstName'], $POST['lastName'], $POST['sex'], $POST['birthday'], $POST['email']);
        $this->userCookie;
        $data["content"] = 'sysad/addSyad/generate.php';
        $this->load->view( 'home.php', $data);
    }

public function editAccount()
{
    $this->userCookie->active = "none";
    $this->AAIS_model->setCookie( array
        $this->userCookie, 'theUserCookie');
    $data["cookie"] = $this->userCookie;
    $data["content"] = 'sysad/editAccount';
    $this->load->view( 'home.php', $data);

public function editAccountStepTwo()
{
    if(isset($POST))
    {
        if($this->form_validation->run () == FALSE || $this->
            AAIS_model->
            dateChecker( $POST['birthday']=='1')
            $this->userCookie->error = 1;
        $this->AAIS_model->
            setCookie( array
                $this->userCookie, 'theUserCookie');
            redirect( base_url('sysad/editAccount'));
    } else
    {
        $this->userCookie->error = 1;
        $this->AAIS_model->
            setCookie( array
                $this->userCookie, 'theUserCookie');
            redirect( base_url('sysad/editAccount'));
    }

62
private function deleteUserPanel($type) {
    if ($type==1)
        $data['patients'] = $this->AAIS_model->getPatients();
    else if ($type==2)
        $data['searchResults'] = $this->searchResults;
    else if ($type==3)
        $data['guardians'] = $this->AAIS_model->getGuardians();
    else if ($type==4)
        $data['searchResults'] = $this->searchResults;
    else if ($type==5)
        $data['search'] = 0;
    $data['therapists'] = $this->AAIS_model->getTherapists();
    else if ($type==6)
        $data['search'] = 1;
    $data['therapists'] = $this->AAIS_model->getTherapists();
    else if ($type==7)
        $data['admins'] = $this->AAIS_model->getAdmins();
    else if ($type==8)
        $data['admins'] = $this->AAIS_model->getAdmins();
    $data['cookie'] = $this->userCookie;
    $data['content'] = 'sysad/delete/main.php';
    $this->load->view('home.php', $data);
}

public function deletePatient() {
    $this->userCookie->active = "deletePatient";
    $this->userCookie->step = 1;
    $this->AAIS_model->setCookie(array($this->userCookie, 'theUserCookie'));
    $this->deleteUserPanel(1);
}

public function deletePatientStepTwo() {
    $this->AAIS_model->deletePatient($POST['user_id']);
    redirect(base_url('sysad/deletePatient'), 1);
}

public function deletePatientAll() {
    $this->AAIS_model->deleteAllPatients();
    redirect(base_url('sysad/deletePatient'), 1);
}

public function deletePatientSearch() {
    $this->searchResults = $this->AAIS_model->searchUser($POST['search'], 1);
    $this->deleteUserPanel(2);
}

public function deleteGuardian() {
    $this->userCookie->active = "deleteGuardian";
    $this->userCookie->step = 2;
    $this->AAIS_model->setCookie(array($this->userCookie, 'theUserCookie'));
    $this->deleteUserPanel(3);
}

public function deleteGuardianStepTwo() {
    // $patient = $this->AAIS_model->deleteUser($POST['user_id']);
    $patient = $this->AAIS_model->getPatientGuardian($POST['user_id']);
    $array = array(
        'guardian_id' => $POST['user_id'],
        'user_id' => $patient->user_id,
        'first_name' => $patient->first_name,
        'last_name' => $patient->last_name
    );
    $this->AAIS_model->setCookie(array('patientCookie', $array));
    redirect(base_url('sysad/deleteGuardian/new'), 1);
}

public function deleteGuardianNew() {
    $data['patient'] = $this->user->patient;
    $data['cookie'] = $this->userCookie;
    $data['content'] = 'sysad/delete/newGuardian.php';
    $this->load->view('home.php', $data);
}

public function deleteGuardianNewStepTwo() {
    $this->load->library('form_validation');
    $this->form_validation->set_rules('first_name', 'First Name', 'required');
    $this->form_validation->set_rules('last_name', 'Last Name', 'required');
    $this->form_validation->set_rules('sex', 'Sex', 'required');
    $this->form_validation->set_rules('phone', 'Phone', 'required');
    $this->form_validation->set_rules('email', 'Email', 'valid_email');
    if ($this->form_validation->run() == FALSE) {
        $this->userCookie->error = 1;
    }
    $this->AAIS_model->setCookie(array($this->userCookie, 'theUserCookie'));
    redirect(base_url('sysad/deleteGuardian/new'), 1);
}

else {
    $this->AAIS_model->deleteUser($this->user->guardian_id);
    $data['details'] = $this->AAIS_model->insertGuardian($POST['first_name'], $POST['last_name'], $POST['sex'], $POST['phone'], $POST['email']);
    $this->userCookie->active = "deleteOT";
    $data['cookie'] = $this->userCookie;
    $data['content'] = 'sysad/delete/newGuardian.php';
    $this->load->view('home.php', $data);
}

public function deleteGuardianSearch() {
    $this->searchResults = $this->AAIS_model->searchUser($POST['search'], 2);
    $this->deleteUserPanel(4);
}

public function deleteOT() {
    $this->userCookie->active = "deleteOT";
    $this->userCookie->step = 3;
}
Listing 3: User_Controller.php

```php
defined('BASEPATH') or exit('No direct script access allowed');

class UserController extends CI_Controller {
    private $userCookie; // constructors
    private $entryCookie;
    private $submissionCookie;
    private $searchCookie;

    public function __construct()
    {
        parent::__construct();
        date_default_timezone_set('Asia/Manila');

        if(isset($_POST['logout']))
        {
            redirect(base_url());
        }
        if($this->session->has_cookie('theEntryCookie'))
        {
            $this->session->set_cookie('theEntryCookie');
        }
        if($this->session->has_cookie('theSubmissionCookie'))
        {
            $this->session->set_cookie('theSubmissionCookie');
        }
        if($this->session->has_cookie('theSearchCookie'))
        {
            $this->session->set_cookie('theSearchCookie');
        }

        $this->load->view('home.php', $data);
    }

    public function deleteSysad() {
        $this->userCookie->active = "deleteSysad";
        $this->userCookie->step = 4;
        $this->session->set_cookie('theUserCookie', 'theUserCookie');
        $this->session->set_cookie('theUserCookie');
        $this->load->view('home.php', $data);
    }

    public function assignTherapist() {
        $this->userCookie->active = "assignTherapist";
        $this->session->set_cookie('theUserCookie', 'theUserCookie');
        $this->load->view('home.php', $data);
    }

    public function editAccount() {
        $this->userCookie->active = "none";
        $this->session->set_cookie('theUserCookie', 'theUserCookie');
        $this->session->set_cookie('theUserCookie', 'theUserCookie');
        $this->load->view('home.php', $data);
    }

    // other functions...
}
```
$this->load->library('form_validation');

$this->form_validation->set_rules('firstName', 'First Name', 'required');
$this->form_validation->set_rules('lastName', 'Last Name', 'required');
$this->form_validation->set_rules('email', 'Email', 'trim|required|valid_email');
$this->form_validation->set_rules('username', 'Username', 'required');
$this->form_validation->set_rules('password', 'Password', 'required');

if (!$FILES['dp'] || $name == '')
    $filename = $this->userCookie->profile_picture;
else
    $date_time = date("YmHis");
    $filename = $this->AAIS_model->moveDpUpload('dp', $date_time);

if ($this->form_validation->run () == FALSE || $this->AAIS_model->dateChecker($POST['birthday']) == 1)
    $id = $this-> AAIS_model->setCookie(array)
    $this->userCookie, $this->userCookie->editAccount();
    redirect(base_url('user/editAccount'));
else
    $id = $this->userCookie->error = 1;
    $this->AAIS_model->setAccount($id, $POST['first_name'], $POST['last_name'], $POST['sex'], $POST['birthday'], $POST['email'], $POST['password'], $filename);
    redirect(base_url('user/editAccount'));

public function journals()
{
    $this->userCookie->active = "journals"
    $this->AAIS_model->setCookie(array)
    $this->userCookie, 'theUserCookie';
    if ($this->AAIS_model->hasCookie('theEntryCookie'))
        $this->AAIS_model->deleteCookie('theEntryCookie');
    if ($this->AAIS_model->hasCookie('theSearchCookie'))
        $this->AAIS_model->deleteCookie('theSearchCookie');
    $this->load->library('pagination');
    $this->config['base_url'] = site_url('UserController/journals');
    $config['total_rows'] = $this->AAIS_model->countJournals($this->userCookie->user_id, $this->userCookie->user_id);
    $config['per_page'] = 5;
    $config['uri_segment'] = 3;
    $this->choice = $config['total_rows'] / $config['per_page'];
    $config['num_links'] = floor($this->choice);

    if ($this->pagination->initialize($config))
        $this->load->view('home.php');
        $this->load->view('home.php');

public function viewJournal()
{
    $this->load->view('home.php');
    $this->load->view('home.php');

public function deleteJournal()
{
    $this->load->view('home.php');
    $this->load->view('home.php');

public function editJournal()
{
    $this->load->view('home.php');
    $this->load->view('home.php');

65
'error' => 0);
$this->AAIS_model->setCookie(
array, "theEntryCookie")
redirect( base_url('user/editJournal'));
}
$data["categories"] = $this->
AAIS_model->getCategories(
$this->userCookie->user_id);
$data["cookie"] = $this->
userCookie;
$data["content"] = "user/journals"
/ed.php'
$this->load->view('home.php', $data);
}
public function editJournalStepTwo()
{
if(isset($POST)){
$this->load->library('form_validation');
$this->form_validation->
set_rules('title', 'Title', 'required');
$this->form_validation->
set_rules('date', 'Date', 'required');
$this->form_validation->
set_rules('details', 'Details', 'required');
if($this->form_validation->
run() == FALSE){
$this->sentryCookie->
error = 1;
$this->AAIS_model->
setCookie( array ) $this->
entryCookie,
theEntryCookie);
redirect( base_url('user/editJournal')
});
else if($this->AAIS_model->
dateChecker($POST[ 'date']) == 1){
$this->sentryCookie->
error = 3;
$this->AAIS_model->
setCookie( array ) $this->
entryCookie,
theEntryCookie);
redirect( base_url('user/editJournal')
});
else{
$this->sentryCookie->
title = $POST[ 'title']
$this->sentryCookie->
date = $POST[ 'date']
$this->sentryCookie->
details = $POST[ 'details']
$this->sentryCookie->
category = $POST[ 'category']
$this->AAIS_model->
setCookie( array ) $this->
entryCookie,
theEntryCookie);
$this->AAIS_model->
editJournal[
$POST[ 'id' ],
$POST[ 'title' ],
$POST[ 'date' ],
$POST[ 'details' ],
$POST[ 'category' ]
]
redirect( base_url('user/editJournal')
});
}
else{
$this->sentryCookie->error = 1;
$this->AAIS_model->
setCookie( array )
$this->sentryCookie, 
theEntryCookie);
redirect( base_url('user/editJournal')
});
}
public function filter($type){
$this->userCookie->category = 
$POST[ 'category']
$this->AAIS_model->setCookie( array ) $this->userCookie, 
theUserCookie);
switch($type){
case 1: redirect( base_url('user/journals'));
break;
case 2: redirect( base_url('user/photos')
break;
case 3: redirect( base_url('user/photos')
break;
public function reminders(){
$this->userCookie->active = "reminders"
$this->AAIS_model->
setCookie( array ) $this->userCookie, 
theUserCookie);
if($this->AAIS_model->
hasCookie('theEntryCookie'))
$this->AAIS_model->
deleteCookie('theEntryCookie');
if($this->AAIS_model->
hasCookie('theSearchCookie'))
$this->AAIS_model->
deleteCookie('theSearchCookie');
$this->load->helper('url')
$this->load->library('pagination')
//pagination settings
 sailors['base_url'] = site_url('User_Controller/reminders');
$sailors['total_rows'] = $this->
AAIS_model->countReminders($this->
userCookie->user_id, $this->
userCookie->user_id); $this->
config[ 'per_page'] = "5";
$sailors[ 'uri_segment'] = "3";
$sailors[ 'choice'] = $this->
config['total_rows'] / $this->
config[ 'per_page'];
$sailors[ 'num_links'] = floor($this->
//config for bootstrap pagination
class integration
$sailors[ 'full_tag_open'] = "<ul class="pagination">"
$sailors[ 'full_tag_close'] = '</ul>'
$sailors[ 'first_tag_open'] = '<li>
$sailors[ 'first_tag_close'] = '</li>'
$sailors[ 'prev_tag_open'] = '<li class="
$sailors[ 'prev_tag_close'] = '</li>'
$sailors[ 'next_tag_open'] = '<li>
$sailors[ 'next_tag_close'] = '</li>'
$sailors[ 'last_tag_open'] = '<li>
$sailors[ 'last_tag_close'] = '</li>'
$sailors[ 'cur_tag_open'] = '<li class="
$sailors[ 'cur_tag_close'] = '</li>'
$sailors[ 'num_tag_open'] = '<li>
$sailors[ 'num_tag_close'] = '</li>'
$this->pagination->initialize($sailors)
$data[ 'page'] = ($this->uri->segment(3)) ? $this->uri->segment(3) : 0;
//call the model function to get the
department data
$data['reminders'] = $this->AAIS_model->
getReminders($sailors['per_page'], $data['page']
); $this->
userCookie->user_id, $this->
userCookie->user_id);
$data[ 'pagination'] = $this->
pagination->create_links;
public function deleteReminder() {
    if ($this->isAAIS_model->deleteReminder($POST['delete'])) {
        $this->entryCookie = $this->setCookie(
            array("category" => 0)
        )
    } else if ($this->isAAIS_model->hasCookie('theSearchCookie')) {
        redirect(base_url('user/remainders/index'));
    } else { 
        $this->isAAIS_model->setCookie(array("user/remainders/edit.php") == 0)
    }
}

public function editReminder() {
    if ($this->isAAIS_model->hasCookie('theEntryCookie')) {
        data["reminder"] = $this->entryCookie;
    } else { 
        $row = $this->isAAIS_model->getReminder($POST['edit'])
    $array = array("reminder_id" => row, "title" => row->title, "date_time" => row->date_time, "details" => row->details, "category" => row->category, "error" => 0)
    $this->isAAIS_model->setCookie(array("user/editReminder")
        redirect(base_url('user/editReminder'))
    }
}

public function editReminderStepTwo() {
    if (isset($POST)) {
        if ($this->isAAIS_model->dateChecker($POST['date_time'])) {
            $this->entryCookie = $this->setCookie(array("user/remainders/edit.php")
                redirect(base_url('user/editReminder'))
        }
        $this->isAAIS_model->setCookie(array("user/remainders/edit.php")
            redirect(base_url('user/editReminder'))
    }
}

public function photos() {
    $this->userCookie = "active" = "photos";
    $this->isAAIS_model->setCookie(array("user/remainders/edit.php")
        redirect(base_url('user/editReminder'))
    }

```php
// config for bootstrap pagination class integration
$config['full_tag_open'] = '<ul class="pagination">';
$config['full_tag_close'] = '</ul>';
$config['first_tag_open'] = '<li class="disabled">';
$config['first_tag_close'] = '</li>';
$config['prev_tag_open'] = '<li class="prev">';
$config['prev_tag_close'] = '</li>';
$config['prev_link'] = '&laquo;';
$config['prev_page'] = '&laquo;';
$config['next_tag_open'] = '<li class="next">';
$config['next_tag_close'] = '</li>';
$config['next_link'] = '&raquo;';
$config['next_page'] = '&raquo;';
$config['last_tag_open'] = '<li class="last"></li>';
$config['last_tag_close'] = '</li>';
$config['last_link'] = 'active class="last" aria-current="true">';
$config['cur_tag_open'] = '<li class="active">';
$config['cur_tag_close'] = '</li>';
$config['num_tag_open'] = '<li>';
$config['num_tag_close'] = '</li>';

$data['content'] = 'user/photos';
$this->load->view('home.php', $data);

public function editPhoto($id)
{
    $this->load->model('AAIS_model');
    $data['photos'] = $this->AAIS_model->getPhotos($config['per_page'], $data['page'], $this->userCookie->user_id, $this->userCookie->category);
    $data['pagination'] = $this->pagination->create_links();
    $this->load->view('home.php', $data);
}

public function deletePhoto($id)
{
    $this->load->model('AAIS_model');
    if (!$this->AAIS_model->hasCookie('theSearchCookie'))
    {
        redirect(base_url('user/photos'));
    }
    else
    {
        $row = $this->AAIS_model->getPhoto($POST['id']);
        array( 'path' => base_url('assets/photos/'), 'row' => $row);
        $this->load->view('home.php', $data);
    }
}
```

array) $this->
    submissionCookie,
    theSubmissionCookie);
$data["entry"] = $this->
    submissionCookie;
}
else{
    array =
    array(
        'pageType' => $pageType,
        'error' => 0
    );
    $this->AAIS_model->setCookie( $array, "theSubmissionCookie" );
    redirect( base_url("user/create/".
        $pageType) );
}
$data["categories"] = $this->
    AAIS_model->getCategories($this->
        userCookie->user_id);
$data["cookie"] = $this->userCookie;
$data["content"] = "user/create/main.
    php";
$this->load->view("home.php",$data);
}
public function submit($type){
    $this->load->library("form_validation");
    $name = $this->userCookie->first_name
    - $this->userCookie->last_name;
    if($type==1 || $type==2){
        $this->form_validation->set_rules( $date,
            'date', 'required' );
        $this->form_validation->set_rules( $tite,
            'title', 'required' );
        $this->form_validation->set_rules( $details,
            'details', 'required' );
        if(FILES['title'] || name' == "")
            $this->submissionCookie->error
            = 1;
        $this->AAIS_model->setCookie( $array, $this->
            userCookie->user_id,
            theSubmissionCookie" );
        redirect( base_url("user/
            create/3") );
    }
    if($type==3){
        $this->form_validation->set_rules( $question,
            'question', 'required' );
        $this->form_validation->set_rules( $choiceA,
            'choiceA', 'required' );
        $this->form_validation->set_rules( $choiceB,
            'choiceB', 'required' );
        $this->form_validation->set_rules( $choiceC,
            'choiceC', 'required' );
        $this->form_validation->set_rules( $choiceD,
            'choiceD', 'required' );
        $this->form_validation->set_rules( $answer,
            'answer', 'required' );
        if($this->form_validation->run() ==
            FALSE)
            $this->submissionCookie->error
            = 1;
        $this->AAIS_model->setCookie( $array, $this->
            submissionCookie,
            theSubmissionCookie" );
        redirect( base_url("user/create/".
            $type) );
    }
    else{
    $qtype = $this->
        dateChecker($this->
            POST['date'] )
        ? 1
        : 0;
    $this->submissionCookie->error
    = 3;
    $this->AAIS_model->setCookie( $array, $this->
        submissionCookie,
        theSubmissionCookie" );
    redirect( base_url("user/
        create/".
        $type) );
    }
    else if($type==4){
        if($this->AAIS_model->
            dateChecker($this->
                POST['date'] )
            ? 1
            : 0)
            $this->submissionCookie->error
            = 3;
        $this->AAIS_model->setCookie( $array, $this->
            submissionCookie,
            theSubmissionCookie" );
        redirect( base_url("user/
            create/".
            $type) );
    }
    $author_id = $this->userCookie->user_id;
    if($this->userCookie->user_type
        == 2)
        $author_id = $this->userCookie->user_id;
    switch($type){
    case 1:
        $this->AAIS_model->
            addJournal($this->
                userCookie->user_id,
                $POST['date'] ,
                $POST['title'] ,
                $POST['details'] ,
                $author_id, $name, 1,
                $POST['category'] );
        break;
    case 2:
        $this->AAIS_model->
            addReminder($this->
                userCookie->user_id,
                $POST['date'] ,
                $POST['title'] ,
                $POST['details'] ,
                $author_id, $name, 1,
                $POST['category'] );
        break;
    case 3:
        $date_time = date( "YmdHis" );
        $filename = $this->
            AAIS_model->
            movePhotoUpload( $this->
                userCookie->user_id,
                $filename, $name, 1,
                $POST['category'] );
        break;
    case 4:
        $filename = 0;
        $qtype = $POST['type'] ;
        if($FILES['filename']
            == "")
            $date_time = date( "YmdHis" );
            $filename = $this->
                AAIS_model->
                movePhotoUpload( $this->
                    userCookie->user_id,
                    $filename, $name, 1,
                    $POST['category'] );
        break;
    case 5:
        $filename = 0;
        $qtype = $POST['type'] ;
        if($FILES['filename']
            == "")
            $date_time = date( "YmdHis" );
            $filename = $this->
                AAIS_model->
                movePhotoUpload( $this->
                    userCookie->user_id,
                    $filename, $name, 1,
                    $POST['category'] );
        break;
public function questions() {
    ${this->userCookie->active = "questions";}
    ${this->AAIS_model->setCookie()} (array)
    ${this->userCookie, 'theUserCookie'}
    if($this->AAIS_model->hasCookie('theEntryCookie'))
        ${this->AAIS_model->deleteCookie('theEntryCookie');}
    if($this->AAIS_model->hasCookie('theSearchCookie'))
        ${this->AAIS_model->deleteCookie('theSearchCookie');}
    ${this->load->helper('url');}
    ${this->load->library('pagination');}
    //pagination settings
    $config['full_tag_open'] = '<ul class="pagination">';
    $config['full_tag_close'] = '</ul>';
    $config['first_tag_open'] = '<li class="active">';
    $config['prev_tag_open'] = '&raquo;';
    $config['full_tag_close'] = '</li>';
    $config['first_tag_close'] = '</li>';
    $config['prev_tag_close'] = '</li>';
    $config['next_tag_open'] = '<li> &raquo;';
    $config['next_tag_close'] = '</li>';
    $config['last_tag_open'] = '</li>';
    $config['last_tag_close'] = '</li>';
    $config['cur_tag_open'] = '<li class="active"><a href="#">';
    $config['cur_tag_close'] = '</li>';
    $this->pagination->initialize($config);
    $data['page'] = (isset($this->uri->segment(3)) ? $this->uri->segment(3) : 0);
    //call the model function to get the department data
    $data['questions'] = $this->AAIS_model->getQuestions($config['per_page']);
    $data['page'] = $this->userCookie->user_id; 1;
    $data['pagination'] = $this->pagination->create_links();
    //load the department view
    $data['cookie'] = $this->userCookie;
    $data['content'] = 'user/questions/main.php';
    $this->load->view('home.php', $data);
}

public function deleteQuestion() {
    $this->AAIS_model->deleteQuestion({

if($this->AAIS_model->hasCookie('theSearchCookie'))
    $data['question'] = $this->entryCookie('type', 'user/questions/edit.php');
else{
    $row = $this->AAIS_model->getQuestion($$POST['edit'])
    array(
        'question_id' => $row->question_id;
        'question_details' => $row->question_details,
        'type' => $row->type,
        'filename' => $row->filename,
        'choiceA' => $row->choiceA,
        'choiceB' => $row->choiceB,
        'choiceC' => $row->choiceC,
        'choiceD' => $row->choiceD,
        $this->load->view('home.php', $data);
}
}
public function editQuestionTwo() {
    if(isset($$POST))
        $this->load->library('form_validation');
    $this->form_validation->set_rules('choiceA', 'Question', 'required');
    $this->form_validation->set_rules('choiceB', 'required');
    $this->form_validation->set_rules('choiceC', 'required');
    $this->form_validation->set_rules('choiceD', 'required');
    $this->form_validation->set_rules('type', 'required');
    $this->form_validation->run() == FALSE)
        $this->entryCookie('error') == 1;
    $this->AAIS_model->setCookie('entryCookie')
        $this->entryCookie('any') $this->entryCookie('theEntryCookie');
    redirect_base_url('');
}
public function search($pageType)
{
    $this->userCookie->active = "search";
    $this->AAS_model->setCookie( (array)
        $this->userCookie , 'theUserCookie'
    );

    if (!isset($_POST['keyword']))
    {
        $this->AAS_model->setCookie( (array)
            $this->userCookie->pageType = 1 ;
            $this->searchCookie ,'
        theSearchCookie' ) ;
    }
    else
    {
        if (isset($_POST['keyword']))
        {
            $this->searchCookie->keyword = $pageType ,
                $this->searchCookie->pageType = 1 ;
            $this->AAS_model->setCookie( (array)
                $this->searchCookie ,'
            theSearchCookie' ) ;
        }
        else
        {
            if (!$this->AAS_model->hasCookie('theEntryCookie') )
            $this->AAS_model->deleteCookie('theEntryCookie');

            switch($pageType)
            { 
            case '1' : $data['entries'] = $this
                    ->AAS_model->searchJournals( (array)
                        $this->userCookie->user_id ,
                        1 , $this->searchCookie->keyword ;
                        $this->searchCookie->pageType = 1 ;
                    $this->AAS_model->setCookie( (array)
                        $this->searchCookie ,'
                    theSearchCookie' ) ;
                break;
            case '2' : $data['entries'] = $this
                    ->AAS_model->searchReminders ( (array)
                        $this->userCookie->user_id ,
                        1 , $this->searchCookie->keyword ;
                        $this->searchCookie->pageType = 2 ;
                    $this->AAS_model->setCookie( (array)
                        $this->searchCookie ,'
                    theSearchCookie' ) ;
                break;
            case '3' : $data['entries'] = $this
                    ->AAS_model->searchPhotos ( (array)
                        $this->userCookie->user_id ,
                        1 , $this->searchCookie->keyword ;
                        $this->searchCookie->pageType = 3 ;
                    $this->AAS_model->setCookie( (array)
                        $this->searchCookie ,'
                    theSearchCookie' ) ;
                break;
            case '4' : $data['entries'] = $this
                    ->AAS_model->searchQuestions ( (array)
                        $this->userCookie->user_id ,
                        1 , $this->searchCookie->keyword ;
                        $this->searchCookie->pageType = 4 ;
                    $this->AAS_model->setCookie( (array)
                        $this->searchCookie ,'
                    theSearchCookie' ) ;
                break;
            default : break;
            }
        } 
    } 

    $this->load->view('user/search/'
        $data);
Listing 4: Patient_Controller.php

```php
public function view($type){
    $this->userCookie->active = "view";
    $this->AAIS_model->setCookie( (array)
        $this->userCookie, 'theUserCookie' );
    $array = array();
    switch($type){
        case 1: #journal = $this->AAIS_model->getJournal(\$_POST['view']);
            array_push($array , $journal);
            $data['entries'] = $array;
            break;
        case 2: #reminder = $this->AAIS_model->getReminder(\$_POST['view']);
            array_push($array , $reminder);
            $data['entries'] = $array;
            break;
        default:
            break;
    }$data['cookie'] = $this->userCookie;
    $data['type'] = $type;
    $data['content'] = 'user/search/view.php';
    $this->load->view('home.php', $data);
}

public function report($pageTitle){
    $this->userCookie->active = "report";
    $this->AAIS_model->setCookie( (array)
        $this->userCookie, 'theUserCookie' );
    if($pageTitle=="2"){
        $data['personal'] = $this->AAIS_model->getScores($this->userCookie->user_id, 1, \$_POST['start'] , \$_POST['end']);
        $data['photo'] = $this->AAIS_model->getScore($this->userCookie->user_id, 2, \$_POST['start'] , \$_POST['end']);
        $data['general'] = $this->AAIS_model->getScores($this->userCookie->user_id, 2, \$_POST['start'] , \$_POST['end']);
    }
    $this->userCookie->step = $pageTitle;
    $this->AAIS_model->setCookie( (array)
        $this->userCookie, 'theUserCookie' );
    $data['cookie'] = $this->userCookie;
    $data['content'] = 'user/report/main.php';
    $this->load->view('home.php', $data);
}
```

```php
Listing 4: Patient_Controller.php
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Patient_Controller extends CI_Controller{
    private $userCookie;
    public function __construct(){
        parent::__construct();
        date_default_timezone_set("Asia/Manila");
        if(isset(\$_POST['logout'])){ redirect(base_url());}
        if($this->AAIS_model->hasCookie('theUserCookie')){
            $this->userCookie = $this->AAIS_model->getCookie('theUserCookie');
        }else redirect(base_url());
    }
    public function dashboard(){
        $this->userCookie->active = "dashboard";
        $this->AAIS_model->setCookie( (array)
            $this->userCookie, 'theUserCookie' );
        $data['reminders'] = $this->AAIS_model->getReminders(3, 0, $this->userCookie->user_id, 1, "All");
        $data['cookie'] = $this->userCookie;
        $data['content'] = 'patient/dashboard.php';
        $this->load->view('home.php', $data);
    }
    public function connections(){
        $this->userCookie->active = "connections";
        $this->AAIS_model->setCookie( (array)
            $this->userCookie, 'theUserCookie' );
        $data['guardian'] = $this->AAIS_model->getGuardian($this->userCookie->user_id);
        $data['therapist'] = $this->AAIS_model->getTherapist($this->userCookie->user_id);
        $data['cookie'] = $this->userCookie;
        $data['content'] = 'patient/connections.php';
        $this->load->view('home.php', $data);
    }
    public function personalTest(){
        $this->userCookie->active = "personalTest";
        $this->AAIS_model->setCookie( (array)
            $this->userCookie, 'theUserCookie' );
        $check1 = $this->AAIS_model->getLastTest($this->userCookie->user_id, 1);
        if($check1==0){
            $this->userCookie->error = 1;
            $this->AAIS_model->getAIPersonalQuestions($this->userCookie->user_id, 1);
            $data['cookie'] = $this->userCookie;
            $data['content'] = 'patient/test/personalTestProper.php';
            $this->load->view('home.php', $data);
        }
    }
    public function personalTestProper(){
        $data['questions'] = $this->AAIS_model->getAIQuestions($this->userCookie->user_id, 1);
        $data['cookie'] = $this->userCookie;
        $data['content'] = 'patient/test/personalTestProper.php';
        $this->load->view('home.php', $data);
    }
    public function submitPersonalTest(){
        $score = 0;
        $items = count(\$_POST['correct']);
        $answers = array();
        for($i=0; $i<count($items); $i++){
            array_push($answers, \$_POST['答案'.$i]);
            if(\$_POST['correct'][$i] == \$_POST['答案'.$i]){
                $score++;}
        }$this->AAIS_model->recordTest($this->userCookie->user_id, 1, $items, $score);
        $data['score'] = $score . "/" . $items;
        $data['questions'] = $this->AAIS_model->getTestQuestions(\$_POST['items']);
        $data['answers'] = $answers;
        $data['cookie'] = $this->userCookie;
        $data['content'] = 'patient/test/personalTestAnswers.php';
        $this->load->view('home.php', $data);
    }
    public function photoRecognition(){
        ...
    }
}
```
Listing 5: Guardian_Controller.php

defined( 'BASEPATH' ) OR exit( 'No direct script access allowed' );

class Guardian_Controller extends CI_Controller {

    private $userCookie;

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

        $date_default_timezone_set( 'Asia/Manila' );

        if( isset( $POST['logout'] ) )
        {
            redirect( base_url() );
        }

        if( $this->A AIS_model->hasCookie( 'theUserCookie' ) )
        {
            $this->userRequests();
            redirect( base_url() );
        }

        public function dashboard()
        {
            $this->userCookie->active = "dashboard"

            $this->A AIS_model->setCookie( array( $this->userCookie, 'theUserCookie' ) );

            $check1 = $this->A AIS_model->getLastTest( $this->userCookie->user_id, 3 );

            if( $check1 >0 )
            {
                $this->userCookie->error = 3;

                $this->A AIS_model->setCookie( array( $this->userCookie, 'theUserCookie' ) );

                $data[ 'cookie' ] = $this->userCookie;

                $data[ 'content' ] = 'patient/test/photoRecognitionProper.php';

                $this->load->view( 'home.php', $data );
            }

            public function generalKnowledge()
            {
                $this->userCookie->active = "generalKnowledge"

                $this->A AIS_model->setCookie( array( $this->userCookie, 'theUserCookie' ) );

                $check1 = $this->A AIS_model->getLastTest( $this->userCookie->user_id, 1 );

                if( $check1 >0 )
                {
                    $this->userCookie->error = 3;

                    $this->A AIS_model->setCookie( array( $this->userCookie, 'theUserCookie' ) );

                    $data[ 'cookie' ] = $this->userCookie;

                    $data[ 'content' ] = 'patient/test/photoRecognitionProper.php';

                    $this->load->view( 'home.php', $data );
                }

                public function submitGeneralKnowledge()
                {
                    $score = 0;

                    $items = count( $POST['correct'] );

                    $answers = array();

                    for( $i = 0; $i < $items; $i++ )
                    {
                        array_push( $answers, $POST['answer'][ $i ] );
                    }

                    if( $POST['correct'][ $i ] == $POST['answer'][ $i ] )
                    {
                        $score++;;
                    }

                    $this->A AIS_model->recordTest( $this->userCookie->user_id, 3, $items, $score );

                    $data[ 'score' ] = $score . '/' . $items;

                    $data[ 'questions' ] = $this->A AIS_model->getTestQuestions( $POST['items'] );

                    $data[ 'answers' ] = $answers;

                    $data[ 'cookie' ] = $this->userCookie;

                    $data[ 'content' ] = 'guardian/generalKnowledgeProper.php';

                    $this->load->view( 'home.php', $data );
                }

                public function photoRecognitionProper()
                {
                    $this->A AIS_model->setCookie( array( $this->userCookie, 'theUserCookie' ) );

                    $check1 = $this->userCookie->error = 2;

                    $this->A AIS_model->setCookie( array( $this->userCookie, 'theUserCookie' ) );

                    $data[ 'cookie' ] = $this->userCookie;

                    $data[ 'content' ] = 'patient/test/photoRecognitionProper.php';

                    $this->load->view( 'home.php', $data );
                }

                public function photoRecognition()
                {
                    $data[ 'questions' ] = $this->A AIS_model->getPhotoQuestions($this->userCookie->user_id, 1);

                    $data[ 'cookie' ] = $this->userCookie;

                    $data[ 'content' ] = 'patient/test/photoRecognitionProper.php';

                    $this->load->view( 'home.php', $data );
                }

                public function submitPhotoRecognition()
                { $score = 0;

                    $items = count( $POST['correct'] );

                    $answers = array();

                    for( $i = 0; $i < $items; $i++ )
                    {
                        array_push( $answers, $POST['answer'][ $i ] );
                    }

                    if( $POST['correct'][ $i ] == $POST['answer'][ $i ] )
                    {
                        $score++;;
                    }

                    $this->A AIS_model->recordTest( $this->userCookie->user_id, 3, $items, $score );

                    $data[ 'score' ] = $score . '/' . $items;

                    $data[ 'questions' ] = $this->A AIS_model->getTestQuestions( $POST['items'] );

                    $data[ 'answers' ] = $answers;

                    $data[ 'cookie' ] = $this->userCookie;

                    $data[ 'content' ] = 'patient/test/generalKnowledgeProper.php';

                    $this->load->view( 'home.php', $data );
                }

                public function userRequests()
                {
                    $this->userCookie->active = "userRequests"

                    $this->A AIS_model->setCookie( array( $this->userCookie, 'theUserCookie' ) );

                    $this->load->helper( 'url' );

                    $this->load->library( 'pagination' );

                    $config[ 'base_url' ] = site_url( 'Guardian_Controller/userRequests' );

                    $config[ 'total_rows' ] = $this->A AIS_model->countUserRequests( $this->userCookie->user_id, 0 );

                    $config[ 'per_page' ] = "5";

                    $config[ 'uri_segment' ] = 3;

                    $choice = $config[ 'total_rows' ] / $config[ 'per_page' ];

                    $config[ 'num_links' ] = floor( $choice );

                    $data[ 'cookie' ] = $this->userCookie;

                    $data[ 'content' ] = 'patient/test/photoRecognitionProper.php';

                    $this->load->view( 'home.php', $data );
                }

                public function generalKnowledgeProper()
                {
                    $data[ 'questions' ] = $this->A AIS_model->getAllGeneralQuestions($this->userCookie->user_id, 1);

                    $data[ 'cookie' ] = $this->userCookie;

                    $data[ 'content' ] = 'patient/test/generalKnowledgeProper.php';

                    $this->load->view( 'home.php', $data );
                }

                public function submitGeneralKnowledge()
                {
                    $score = 0;

                    $items = count( $POST['correct'] );

                    $answers = array();

                    for( $i = 0; $i < $items; $i++ )
                    {
                        array_push( $answers, $POST['answer'][ $i ] );
                    }

                    if( $POST['correct'][ $i ] == $POST['answer'][ $i ] )
                    {
                        $score++;;
                    }

                    $this->A AIS_model->recordTest( $this->userCookie->user_id, 3, $items, $score );

                    $data[ 'score' ] = $score . '/' . $items;

                    $data[ 'questions' ] = $this->A AIS_model->getTestQuestions( $POST['items'] );

                    $data[ 'answers' ] = $answers;

                    $data[ 'cookie' ] = $this->userCookie;

                    $data[ 'content' ] = 'guardian/generalKnowledgeProper.php';

                    $this->load->view( 'home.php', $data );
                }

            break;
        } else {
            redirect( base_url() );
        }

    private function $config( 'total_rows' ) = $this->A AIS_model->countUserRequests( $this->userCookie->user_id, 0 );

    private function $config( 'per_page' ) = "5";

    private function $config[ 'uri_segment' ] = 3;

    private function $config[ 'num_links' ] = floor( $choice );


// config for bootstrap pagination
$class integration

$post type = "pagination"

$open = "submit"

 postData 

// call the model function to get the page
$users = $this->AAS_model->

// load the department view
$data[cookie] = $this->userCookie;
$data[content] = 'guardian/

// load the view
$data[page] = $this->

public function submissions()

switch($POST['submission_type'])

case '1': $this->AAS_model->deleteReminder($POST['submission_type_id'])

public function accept()

switch($POST['submission_type'])

case '1': $this->AAS_model->acceptJournal($POST['submission_type_id'])

public function connections()

$this->userCookie->active = "connections"

$this->AAS_model->deleteSubmission($POST['submission_type_id'])

$this->userCookie->active = "submissions"

switch($POST['submission_type'])

case '1': $this->AAS_model->acceptPhoto($POST['submission_type_id'])

$data[page] = $this->AAS_model->

$this->userCookie = $this->userCookie;
$data = getUserRequests($this->

public function acceptUser()

$this->AAS_model->acceptUser($

public function rejectUser()

$this->AAS_model->deleteSubmission($POST['submission_type_id'])

$this->userCookie = $this->userCookie;
$data = getUserRequests($this->

public function reject()
public function processPhotoQuestion() {
    if ($this->form_validation->run() == FALSE) {
        $this->userCookie->error = 1;
        redirect(base_url('guardian/photosQuestions.php'));
    } else {
        $name = $this->userCookie->first_name . ' ' . $this->userCookie->last_name;
        $id = $this->userCookie->user_id;
        $data['question'] = $this->AAIS_model->getQuestion($id);
        $data['cookie'] = $this->userCookie;
        $data['content'] = 'guardian/photoQuestionPreview.php';
        $this->load->view('home.php', $data);
    }
}

public function categories() {
    $this->userCookie->active = "categories"
    $this->AAIS_model->setCookie(array)
    $this->userCookie, $this->userCookie;
    $data['category'] = $this->AAIS_model->getCategories($this->userCookie);
    $data['content'] = 'guardian/categories.php';
    $this->load->view('home.php', $data);
}

public function photoQuestions() {
    $this->userCookie->active = "photoQuestion"
    $this->AAIS_model->setCookie(array)
    $this->userCookie, $this->userCookie;
    $data['photos'] = $this->AAIS_model->getUserPhotos($this->userCookie->user_id);
    $data['cookie'] = $this->userCookie;
    $data['content'] = 'guardian/photosQuestions.php';
    $this->load->view('home.php', $data);
}

public function addCategory() {
    $this->AAIS_model->addCategory($this->userCookie->user_id, $POST['category']);
    redirect(base_url('guardian/categories.php'));
}

public function deleteCategory() {
    $this->AAIS_model->deleteCategory($POST['category_id']);
    redirect(base_url('guardian/categories.php'));
}

public function manageUsers() {
    $this->userCookie->active = "manageUsers"
    $this->AAIS_model->setCookie(array)
    $this->userCookie, $this->userCookie;
    $this->load->helper('url');
    $this->load->library('pagination');
    $config['per_page'] = 5;
    $config['uri_segment'] = 2;
    $choice = $config['total_rows'] / $config['per_page'];
    $config['num_links'] = floor($choice);
    $config['pagination'] = '<ul class="pagination">';
    $config['first_tag_open'] = '<li><a href='; $config['first_tag_close'] = '</a>';
    $config['prev_tag_open'] = '&laquo;'; $config['prev_tag_close'] = '</li>);
    $config['first_link'] = '&raquo;'; $config['next_link'] = '</li>);
    $config['last_tag_open'] = '<li><a href='; $config['last_tag_close'] = '</a>);';
    $config['last_tag_open'] = '<li><a href='; $config['last_tag_close'] = '</a>);';
    $this->pagination->initialize($config);
    $data['page'] = ($this->uri->segment(3)) ? $this->uri->segment(3) : 1;
    $this->load->view('home.php', $data);
}

public function deleteUser() {
    $this->AAIS_model->deleteUser($this->userCookie->user_id);
    $data['cookie'] = $this->userCookie;
    $data['content'] = 'guardian/manageUsers.php';
    $this->load->view('home.php', $data);
}
Listing 6: OT_Controller.php

```php
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class OT_Controller extends CI_Controller {

private $userCookie;
private $submissionCookie;
private $entryCookie;

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

date_default_timezone_set('Asia/Manila');

if(isset($_POST['logout'])){
    redirect(base_url());
}
if( $this->AAIS_model->hasCookie('theUserCookie')) {
    $this->userCookie = $this->AAIS_model->getCookie('theUserCookie');
}
else
    redirect(base_url());

if( $this->AAIS_model->hasCookie('theSubmissionCookie')) {
    $this->submissionCookie = $this->AAIS_model->getCookie('theSubmissionCookie');
}
if( $this->AAIS_model->hasCookie('theEntryCookie')) {
    $this->entryCookie = $this->AAIS_model->getCookie('theEntryCookie');
}

public function dashboard() {
    $data['content'] = 'ot/dashboard.php';
    $data['cookie'] = $this->userCookie;
    $data['content'] = 'ot/dashboard.php';
    $this->load->view('home.php', $data);
}

public function editAccount() {
    $data['content'] = 'ot/editAccount.php';
    $this->load->view('home.php', $data);
}

public function editAccountStepTwo() {
    if(isset($_POST)) {
        if($_FILES['dp']['name'] == '') {
            $filename = $this->userCookie->profile_picture;
        } else {
            $date_time = date('YmdHis');
            $filename = $this->AAIS_model->moveDpUpload('dp', $date_time);
        }
        if($this->form_validation->run() == FALSE || $this->AAIS_model->dateChecker($_POST['birthday']) == 0) {
            $this->userCookie->error = 1;
            $this->AAIS_model->setCookie(array)
                $this->userCookie = $this->AAIS_model->setCookie('theUserCookie');
                redirect(base_url('ot/editAccount'));
        } else {
            $id = $this->userCookie->user_id;
            if($this->userCookie->user_type == 2) {
                $id = $this->userCookie->guardian_id;
            }
            $this->AAIS_model->setCookie(array)
                $this->userCookie = $this->AAIS_model->setCookie('theUserCookie');
                redirect(base_url('ot/editAccount'));
        }
    } else {
        $this->userCookie->error = 1;
        $this->AAIS_model->setCookie(array)
            $this->userCookie = $this->AAIS_model->setCookie('theUserCookie');
            redirect(base_url('ot/editAccount'));
    }
}

public function create($pageType) {
    $this->userCookie->active = "create";
    $this->AAIS_model->setCookie(array)
        $this->userCookie = $this->AAIS_model->setCookie('theUserCookie');
    $data['content'] = 'ot/editAccount.php';
    $this->load->view('home.php', $data);
}
```

76
public function submit($type)
{
    $this->load->library('form_validation');
    $this->userCookie->first_name = $this->userCookie->last_name;
    $this->form_validation->set_rules('patient', 'patient', 'required');
    if($type==1 || $type==2){
        $this->form_validation->set_rules('date', 'date', 'required');
        if($type==3){
            $this->form_validation->set_rules('details', 'details', 'required');
        }
    }
    if($type==4){
        $this->form_validation->set_rules('question', 'question', 'required');
    }
    $this->form_validation->run();
    if($type==1 || $type==3){
        if ($this->AAIS_model->dateChecker($_POST['date']) == 1){
            $this->submissionCookie = $_POST['id'] = 1;
            $this->AAIS_model->setCookie($array = $this->submissionCookie, 'theSubmissionCookie');
            redirect(base_url('ot/create/' . $type));
        }
    } else {
        if ($type==1 || $type==3){
            if ($this->AAIS_model->dateChecker($_POST['date']) == 0){
                $this->submissionCookie = error = 3;
                $this->AAIS_model->setCookie($array = $this->submissionCookie, 'theSubmissionCookie');
                redirect(base_url('ot/create/' . $type));
            }
            if ($type==2){
                if ($this->AAIS_model->dateChecker($_POST['date']) == 0){
                    $this->submissionCookie = error = 3;
                    $this->AAIS_model->setCookie($array = $this->submissionCookie, 'theSubmissionCookie');
                    redirect(base_url('ot/create/' . $type));
                }
            }
        }
    }
}

public function patients()
{
    $this->userCookie->active = "patients"
    $this->AAIS_model->setCookie($array = $this->userCookie, 'theUserCookie', 'connections'= $this->AAIS_model->getOFTconnections($this->userCookie->user_id));
    $data['cookie'] = $this->userCookie;
}
function reminders() {
    $this->userCookie = new UserCookie('theUserCookie');

    $this->AAIS_model->setCookie(array('name' => $this->userCookie->name, 'title' => $this->userCookie->title);

    if($this->AAIS_model->checkCookie('theEntryCookie')) {
        $this->AAIS_model->deleteCookie('theEntryCookie');
        $data['reminder'] = $this->AAIS_model->getOTReminders($this->userCookie); // Get OT reminders
    }

    $data['content'] = 'ot/patients.php';
    $data['cookie'] = $this->userCookie;
    $data['reminders'] = $this->AAIS_model->getReminders($this->userCookie);
    $data['content'] = 'ot/reminders/main.php';
    $data['name'] = $this->userCookie;
    $this->load->view('home.php', $data);
}

function deleteReminder() {
    $this->AAIS_model->deleteReminders($this->userCookie);
    $this->AAIS_model->deleteCookie('theEntryCookie');
    $data['reminder'] = $this->AAIS_model->getReminders($this->userCookie);
    $data['content'] = 'ot/reminders/main.php';
    $this->load->view('home.php', $data);
}

function editReminder() {
    if($this->AAIS_model->checkCookie('theEntryCookie')) {
        $row = $this->AAIS_model->getReminder($this->userCookie);
        $array = array('patient_name' => $row->patient_name, 'reminder_id' => $row->reminder_id, 'title' => $row->title, 'date_time' => $row->date_time, 'details' => $row->details, 'error' => 0);
        $this->AAIS_model->setCookie(array('name' => 'theEntryCookie':
            redirect(base_url('ot/editReminder')));
    }
    $data['cookie'] = $this->userCookie;
    $data['content'] = 'ot/reminders/edit.php';
    $this->load->view('home.php', $data);
}

function report($pageType) {
    $this->userCookie = new UserCookie('theUserCookie');
    $data['personal'] = $this->AAIS_model->getScores($POST['patient'] => 1, $POST['start'] => 2, $POST['end']);
    $data['photo'] = $this->AAIS_model->getScores($POST['patient'] => 1, $POST['start'] => 2, $POST['end']);
    $data['general'] = $this->AAIS_model->getScores($POST['patient'] => 1, $POST['start'] => 2, $POST['end']);
    $data['name'] = $this->AAIS_model->getPatient($POST['patient']);
    if($pageType == 2) {
        $this->userCookie = new UserCookie('theUserCookie');
        $data['patient_list'] = $this->AAIS_model->getPatientOT($this->userCookie,
            $data['name'] => $this->userCookie;
            $data['content'] = 'ot/report/main.php';
    } else if($this->AAIS_model->dateChecker($POST['date_time'] == 0) {
        $this->userCookie = new UserCookie('theUserCookie');
        $data['reminders'] = $this->AAIS_model->setReminder($this->userCookie,
            $this->userCookie->reminders('ReminderStatus($POST['id'])';
            redirect(base_url('ot/editReminder')));
    } else {
        $this->userCookie = new UserCookie('theUserCookie');
        $data['content'] = 'ot/reminders/edit.php';
        $this->load->view('home.php', $data);
    }
}

function editReminderStepTwo() {
    if(isset($POST)) {
        $this->load->library('form_validation');
        $this->form_validation->set_rules('title', 'Title', 'required');
        $this->form_validation->set_rules('details', 'Details', 'required');
        if($this->form_validation->run() === FALSE) {
            $this->userCookie = new UserCookie('theUserCookie');
            $this->userCookie = new UserCookie('theUserCookie');
            $data['content'] = 'ot/report/main.php';
    } else {
        $this->userCookie = new UserCookie('theUserCookie');
        $data['content'] = 'ot/report/main.php';
    }
Listing 7: PublicUser_Controller.php

```
<?php
defined('_BASEPATH') or exit('No direct script access allowed');

class PublicUser_Controller extends C1_Controller {
    private $userCookie;
    private $submissionCookie;

    public function __construct()
    {
        $this->load->library('PublicUser_Controller/dashboard');
    }

    public function dashboard()
    {
        $this->userCookie->active = "dashboard";
        $this->A AIS_model->hasCookie('theUserCookie');
        $this->load->helper('url');
        $this->load->library('pagination');
        $this->A AIS_model->setCookie(array($this->userCookie, 'theUserCookie'));
        $this->load->library('pagination');
        $this->A AIS_model->setCookie(array($this->userCookie, 'theSubmissionCookie'));
    }

    public function setCookie($type)
    {
        $this->userCookie->active = "$type";
        $this->A AIS_model->setCookie(array($this->userCookie, 'theUserCookie'));
    }

    public function editAccountStepTwo()
    {
        if(isset($_POST['logout'])){ $this->dashboard();
            if( $this->A AIS_model->hasCookie('theUserCookie') )
            {$this->userCookie = $this->A AIS_model->getCookie('theUserCookie');
                else redirect(base_url());
            if( $this->A AIS_model->hasCookie('theSubmissionCookie') $this->userCookie = $this->A AIS_model->getCookie('theSubmissionCookie');
        }
        $this->load->helper('url');
        $this->load->library('pagination');
        $this->A AIS_model->countUserSubmissions($this->userCookie->publicUser_id, $this->userCookie->step); $this->dashboard().
        $config['total_rows'] = $this->A AIS_model->countUserSubmissions($this->userCookie->publicUser_id, $this->userCookie->step); $config['per_page'] = 5;
        $config['uri_segment'] = 3;
        $choice = $config['total_rows'] / $config['per_page'];
        $config['num_links'] = floor($choice);
        $this->load->library('form_validation');
        $this->form_validation->set_rules('firstname', 'First Name', 'required');
        $this->form_validation->set_rules('lastname', 'Last Name', 'required');
        $this->form_validation->set_rules('email', 'Email', 'valid_email');
        $this->form_validation->set_rules('username', 'Username', 'required');
        $this->form_validation->set_rules('password', 'Password', 'required');
        if(!isset($_FILES['dp']["name"]){
            $filename = $this->userCookie->profile_picture;
        } else{
            $date_time = date("YmdHis");
            $filename = $this->A AIS_model->moveDpUpload('dp', $date_time);
        }
        if(!$this->form_validation->run() == FALSE){ $this->A AIS_model->setCookie(' theUserCookie'); $POST["birthday"] !==[])
        $this->userCookie->error = $this->A AIS_model->setCookie(' theUserCookie');
        $this->userCookie->error = $this->A AIS_model->setCookie(' theUserCookie');
        redirect(base_url('publicUser/edAccount'));
    }
}
```

```
if($type==4){
    $this->form_validation->set_rules('question', 'question', 'required');
    $this->form_validation->set_rules('choiceA', 'choiceA', 'required');
    $this->form_validation->set_rules('choiceB', 'choiceB', 'required');
    $this->form_validation->set_rules('choiceC', 'choiceC', 'required');
    $this->form_validation->set_rules('answer', 'answer', 'required');
}

if($this->form_validation->run() == FALSE){
    if($this->session->userdata('pageType')){
        $this->session->set_flashdata('error', 3);
        $this->session->set_flashdata('subPageType', 'create');
        $this->session->set_flashdata('pageType', $type);
    }
    else if($type==2){
        if($this->session->userdata('pageType')){
            $this->session->set_flashdata('error', 3);
            $this->session->set_flashdata('subPageType', 'create');
            $this->session->set_flashdata('pageType', $type);
        }
        else if($type==3){
            if($this->session->userdata('pageType')){
                $this->session->set_flashdata('error', 3);
                $this->session->set_flashdata('subPageType', 'create');
                $this->session->set_flashdata('pageType', $type);
            }
        }
    }
}
else if($type==1 || $type==3){
    if($this->session->userdata('pageType')){
        $this->session->set_flashdata('error', 3);
        $this->session->set_flashdata('subPageType', 'create');
        $this->session->set_flashdata('pageType', $type);
    }
}
Listing 8: AAIS_Model.php

```php
defined( 'BASEPATH' ) OR exit( 'No direct script access allowed' );

class AAIS_model extends CI_Model
{

  public function encrypt( $data , $password , $type )
  {
    return openssl_encrypt($data , 'aes-256-cbc' , $password , OPENSSL_RAWDATA , $iv );
  }

  public function decrypt( $data , $password , $type )
  {
    return openssl_decrypt($data , 'aes-256-cbc' , $password , OPENSSL_RAWDATA , $iv );
  }

  public function getPassword( $first , $last , $password )
  {
    $encrypt_password = base64_encode(md5( $first ));
    $encrypt_iv = substr(base64_encode(md5($last)) , 0, 16);
    return base64_encode($this->encrypt( $password , $encrypt_password , $encrypt_iv ));
  }

  public function setPassword( $first , $last , $password )
  {
    $encrypt_password = base64_encode(md5( $first ));
    $encrypt_iv = substr(base64_encode(md5($last)) , 0, 16);
    return base64_encode($this->encrypt( $password , $encrypt_password , $encrypt_iv ));
  }

  public function login()
  {
    $this->db->where( 'username' , $POST['username'] );
    $query = $this->db->get( 'user' );
    if($query->num_rows() > 0){
      $row = $query->row();
      $pw = $this->db->getPassword($row->first_name , $row->last_name , $row->password);
      if($pw == $POST['password']){
        $user = $row->user_id;
        $publicUser = 0;
        $guardian = 0;
        $public = 0;
        $step = 1;
      }
    }
  }

  public function getCookie( $name )
  {
    $password = base64_encode(md5( $password ));
    $encrypted_cookie = substr(base64_encode(md5($first)) , 0, 16);
    return base64_encode($this->encrypt( $password , $encrypted_cookie , $encrypted_iv ));
  }

  public function setCookie( array , $name )
  {
    $json = json_encode( $encrypted_cookie );
    $this->db->setCookie( array , $name , $encrypted_cookie , (60*60*24) );
  }

  public function hasCookie( $name )
  {
    return ( get_cookie($name) != true );
  }

  public function deleteCookie( $name )
  {
    delete_cookie($name);
  }

  /* All user Functions */

  public function getAllUsers()
  {
    $query = $this->db->get( 'user' );
    if($query->num_rows() > 0){
      $row = $query->row();
      $pw = $this->db->getPassword($row->first_name , $row->last_name , $row->password);
      if($pw == $this->db->get( 'user' )->password){
        $user = $row->user_id;
        $publicUser = 0;
        $guardian = 0;
        $public = 0;
        $step = 1;
      }
    }
  }
}
```

81
begin code

```
if (isset($setType)) {
    $setAuthor = $this->db->setAuthor($id, $name);
}
```

end code
public function getPatients() {
    $this->db->where('user_type', 1);
    $query = $this->db->get('user');
    return $query->result();
}

public function getPatient($id) {
    $this->db->where('user_id', $id);
    $query = $this->db->get('user');
    return $query->row();
}

public function getPatientOT($id) {
    $patient_list = array();
    $this->db->where('therapist_id', $id);
    $query = $this->db->get('patient');
    foreach($query->result() as $row)
    {    
        $user = $row->user_id;
        $query2 = $this->db->get('user');
        $query2 = $query2->row();
        $public = $row2->public_id;
        $public2 = $query2->row();
        $array = array(
            'id' => $row2->id,
            'first_name' => $row2->first_name,
            'last_name' => $row2->last_name,
            'email' => $row2->email,
            'user_type' => $row2->user_type,
            'public_id' => $public,
            'public_user_id' => $publicUser,
            'public' => $row2->public,
            'step' => 1,
            'category' => 'all');
        $this->AAIS_model->setCookie($array, "theUserCookie");
    }
}

private function changeAuthors($id, $name) {
    $array = array("author" => $name);
    $this->db->where('user_id', $id);
    $this->db->order_by('id', "desc");
    $query = $this->db->get('user');
    $array = $query->result();
    foreach($array as $user)
    {    
        $user->user_id = $user->id;
        $query2 = $this->db->get('patient');
        $query2 = $query2->row();
        $array2 = $query2->result();
        $array3 = $query3->row();
        $array = array(
            'patient' => $row,
            'guardian' => $array2->row());
        array_push($connections, $array);
    }
    return $connections;
}

public function getGuardians($id) {
    $this->db->where('user_id', $id);
    $query = $this->db->get('user');
    $row = $query->row();
    $query2 = $this->db->get('patient');
    $query2 = $query2->row();
    $array = array(
        'patient' => $row,
        'guardian' => $array2->row());
    array_push($connections, $array);
    return $connections;
}

public function getTherapists() {
    $this->db->where('user_type', 3);
    $query = $this->db->get('user');
    return $query->result();
}
return $query->result();
}

public function getTherapist($id)
{
    $this->db->where( 'user_id', $id );
    $query = $this->db->get('patient');
    return $this->db->getUser($query->row());
}

public function getOTConnections($id)
{
    $connections = array();
    $this->db->where('therapist_id', $id);
    $query = $this->db->get('patient');
    foreach($query->result() as $row)
    {
        array_push($connections, $row);
    }
    return $connections;
}

public function getAdmins()
{
    $this->db->where('user_type', 0);
    $this->db->order_by('last_name', 'asc');
    $query = $this->db->get('user');
    return $query->result();
}

public function getUser($id)
{
    $this->db->where('user_id', $id);
    $query = $this->db->get('user');
    return $query->row();
}

public function searchUser($search, $userType)
{
    if (strtolower($search) !== false) {
        $string = explode(',', $search);
        $this->db->where('user_type', $userType); // $this->db->where('first_name', $string[0]);
        foreach($string as $a)
        {
            $this->db->or_where('first_name', $a);
        }
        $this->db->or_where('last_name', $a);
        $query = $this->db->get('user');
        return $query->result();
    } else {
        $this->db->where('user_type', $userType);
        $this->db->order_by('first_name', 'asc');
        $this->db->or_where('last_name', $search);
        $this->db->where('user_type', $userType);
        $query = $this->db->get('user');
        return $query->result();
    }
}

// Objects Functions
*
private function submit($user, $type, $id, $dateSub)
{
    $array = array();
    $user_id => $user,
    "submission_type" => $type,
    "submission_type_id" => $id,
    "date_submitted" => $dateSub;
    $this->db->insert('submission', $array);
    return $this->db->affected_rows();
}

public function countSubmissions($user)
{
    $this->db->where('user_id', $user);
    $query = $this->db->get('submission');
    return $query->num_rows();
}

public function getSubmissions($user, $limit, $start)
{
    $this->db->where('user_id', $user);
    $this->db->order_by('date_submitted', 'desc');
    $query = $this->db->get('submission', $limit, $start);
    $result = $query->result();
    $entries = array();
    foreach($result as $row)
    {
        switch($row->submission_type)
        {
            case '1':
                $this->db->where('journal_id');
                $query2 = $this->db->get('journal');
                $array = array()
                "object" => $query2->row();
                "submission_id" => $row->submission_id
                "submission_type" => $row->submission_type
                "journal_id";
                array_push($entries, $array);
                break;
            case '2':
                $this->db->where('reminder_id');
                $query2 = $this->db->get('reminder');
                $array = array()
                "object" => $query2->row();
                "submission_id" => $row->submission_id;
                "submission_type" => $row->submission_type
                "reminder_id";
                array_push($entries, $array);
                break;
            case '3':
                $this->db->where('photo_id');
                $query2 = $this->db->get('photo');
                $array = array()
                "object" => $query2->row();
                "submission_id" => $row->submission_id;
                "submission_type" => $row->submission_type
                "photo_id";
                array_push($entries, $array);
                break;
        }
    }
    return $entries;
}
public function getJournal($id)
{
    $this->db->where('journal_id', $id);
    $query = $this->db->get('journal');
    return $query->row();
}

public function countJournals($user, $category)
{
    $this->db->where('user_id', $user);
    $this->db->where('status', 1);
    if ($category != 'all')
        $this->db->where('category', $category);
    $query = $this->db->get('journal');
    return $query->num_rows();
}

public function deleteJournal($id)
{
    $this->db->where('journal_id', $id);
    $this->db->delete('journal');
}

public function editJournal($id, $title, $date, $details, $category)
{
    $array = array(
        'title' => $title,
        'date' => $date,
        'details' => $details,
        'category' => $category
    );
    $this->db->where('journal_id', $id);
    $this->db->update('journal', $array);
}

public function acceptJournal($id)
{
    $array = array(
        'status' => 1
    );
    $this->db->where('journal_id', $id);
    $this->db->update('journal', $array);
}

public function searchJournals($user, $status, $keyword)
{
    $this->db->where('user_id', $user);
    $this->db->where('status', $status);
    $this->db->where('author LIKE%', $keyword%
        OR title LIKE% $keyword%
        OR details LIKE% $keyword%
    );
    $this->db->order_by('date_created', 'desc');
    $query = $this->db->get('journal');
    return $query->result();
}

public function getJournals($limit, $start, $user, $status, $category)
{
    $this->db->where('user_id', $user);
    $this->db->where('status', $status);
    if ($category == 'all')
        $this->db->where('category', $category);
    $this->db->order_by('date_created', 'desc');
    $query = $this->db->get('journal', $limit, $start);
    return $query->result();
}

85
public function acceptReminder($id)
{
    $array = array(
        "status" => 1
    );
    $this->db->where( 'reminder_id' , $id ) ;
    $this->db->update( "reminder" , $array ) ;
}

public function addPhoto($user , $date , $filename , $caption = NULL)
{
    $array = array(
        "user_id" => $user ,
        "author_id" => $userid ,
        "author" => $author ,
        "filename" => $filename ,
        "date" => $date ,
        "caption" => $caption ,
        "status" => $status ,
        "date_created" => $dateSub
    );
    if ($status == 0)
    {
        $id = $this->db->insert_id();
        return $this->db->insert( "photo" , $array , $limit , $start , $user , $status , $category );
    }
    return $this->db->insert( "photo" , $array , $limit , $start , $user , $status , $category );
}

public function getUserPhotos($user)
{
    $array = array(
        "user_id" => $user ,
        "status" => $status ,
        "category" => $category
    );
    $this->db->where( "user_id" , $user );
    $this->db->where( "status" , $status );
    $this->db->where( "category" , $category );
    $this->db->order_by("date_created" , "desc" );
    $query = $this->db->get( "photo" );
    $query->result();
}

public function countPhotos($user , $status , $category)
{
    $array = array(
        "status" => $status ,
        "category" => $category
    );
    $this->db->where( "user_id" , $user );
    $this->db->where( "status" , $status );
    $this->db->where( "category" , $category );
    $query = $this->db->get( "photo" );
    $query->result();
}

public function deletePhoto($id)
{
    $this->db->where( 'photo_id' , $id );
    $query = $this->db->get( "photo" );
    $query->row();
    unlink($query->row['filename']);
    $this->db->delete( "photo" , $id );
    return $this->db->delete( "photo" , $id );
}

public function deleteReminder($id)
{
    $this->db->where( 'reminder_id' , $id );
    $theid = $this->db->delete( "reminder" );
    return $theid;
}

public function editReminder($id , $title , $date , $details , $category)
{
    $array = array(
        "title" => $title ,
        "date_time" => $date ,
        "details" => $details ,
        "category" => $category
    );
    $this->db->where( 'reminder_id' , $id ) ;
    $this->db->update( "reminder" , $array ) ;
}

public function acceptReminder($id)
{
    $array = array(
        "status" => 1
    );
    $this->db->where( 'reminder_id' , $id ) ;
    $this->db->update( "reminder" , $array ) ;
}

public function addPhoto($user , $date , $filename , $caption = NULL)
{
    $array = array(
        "user_id" => $user ,
        "author_id" => $userid ,
        "author" => $author ,
        "filename" => $filename ,
        "date" => $date ,
        "caption" => $caption ,
        "status" => $status ,
        "date_created" => $dateSub
    );
    if ($status == 0)
    {
        $id = $this->db->insert_id();
        return $this->db->insert( "photo" , $array , $limit , $start , $user , $status , $category );
    }
    return $this->db->insert( "photo" , $array , $limit , $start , $user , $status , $category );
}
$keyword%'));

foreach($items as $i){
	$query = $this->db->get('question', $i);
	num_rows($query);
}

array_push($questions, $query->row());

return $questions;

public function movePhotoUpload($input_name, $file_name)
{
	$ext = pathinfo($FILES[$input_name]['name'], PATHINFO_EXTENSION);
	$filename = "($file_name).{$ext}";
	destination_path = getcwd() . '/' . $FILES['input_name']['path'] . $input_name . $EXTENSION;
	namespace = move_uploaded_file($FILES['input_name']['tmp_name'], $destination_path);

return $filename;

}
public function moveUploadedFile($input_name, $file_name) {
    $ext = pathinfo($FILES[$input_name]['name'], PATHINFO_EXTENSION);
    $filename = $file_name['tmp_name'];
    $destination_path = getcwd() . '/assets/' . $filename;
    move_uploaded_file($FILES[$input_name]['tmp_name'], $destination_path);
    return $filename;
}

public function recordTest($user, $type, $items, $score) {
    $date = date("Y-m-d");
    $array = array(
        "user_id" => $user,  
        "test_type" => $type,  
        "date" => $date,  
        "items" => $items,  
        "score" => $score
    );
    $this->db->insert('test', $array);
}

public function getScores($user, $type, $start, $end) {
    $this->db->where('user_id', $user);
    $this->db->where('test_type', $type);
    $this->db->where('date', $date)
    $query = $this->db->get('test');
    return $query->result();
}

public function getLastTest($user, $type) {
    $date = date("Y-m-d");
    $this->db->where('user_id', $user);
    $this->db->where('test_type', $type);
    $this->db->where('date', $date);
    $query = $this->db->get('test');
    return $query->num_rows();
}

public function dateChecker($date){
    $then = new DateTime($date);
    $now = new DateTime();
    if($then < $now) { //past
        return 0;
    }
    return 1; //future
}

/* System Admin Functions */

public function setAddAccountCookie() {
    $array = array(
        "patientFirst" => "",  
        "patientLast" => "",  
        "patientSex" => "",  
        "patientEmail" => "",  
        "patientPassword" => "",  
        "guardianFirst" => "",  
        "guardianLast" => "",  
        "guardianSex" => "",  
        "guardianBirthday" => "",  
        "guardianEmail" => "",  
        "guardianUsername" => "",  
        "guardianPassword" => "",  
        "ot_id" => "4"
    );
    $this->setCookie($array, "addAccountCookie");
}

public function createAccount() {
    $cookie = $this->getCookie('addAccountCookie');
    $this->generateUsernameAndPassword($cookie->patientFirst, $cookie->patientLast);
    $cookie->guardianUsername = $this->guardianUsername;
    $cookie->guardianPassword = $this->guardianPassword;
    $this->setCookie($array, "addAccountCookie");
    $this->generateUsernameAndPassword($cookie->patientFirst, $cookie->patientLast);
    $cookie->patientUsername = $patient['user_name'];
    $cookie->patientPassword = $password;
    $this->setCookie($array, "addAccountCookie");
}
setPassword($first, $last, $user['password']);

    $lastName => $first, 
    "name" => $first, 
    "sex" => $sex, 
    "birthday" => $birthday, 
    "email" => $email, 
    "user_type" => 4,
    "filename" => $filename
);

public function insertOT($first, $last, $user)
{

array = array( 
    "user_id" => $cookie->getUser_id(),
    "public_id" => $public_id,
    "note" => $note,
    "status" => 0
);

public function deletePatient($id)
{

    $query = $this->db->where( 'user_id', $id )
    ->get( 'patient' );

public function insertSysad($first, $last, $sex, $birthday, $email, $patientID)
{

$sysad = $this->generateUsernameAndPassword($first, $last);

$array = array( 
    "username" => $sysad['username'], 
    "password" => $sysad['password']
);

public function insertGuardian($first, $last, $user)
{

$array = array( 
    "username" => $guardian['username'], 
    "password" => $guardian['password']
);

private function generateRandomString($length)
{

$characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';

$randomString = ''; 

for ($i = 0; $i < $length; $i++)
{
    $randomString .= $characters[ rand(0, strlen($characters) - 1) ];
}

return $randomString;

public function insertOT($first, $last, $sex, $birthday, $email)
{

$ot = $this->generateUsernameAndPassword($first, $last);

array = array( 
    "username" => $ot['username'], 
    "password" => $ot['password']
);

private function generateRandomString($length)
{

$characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';

$randomString = ''; 

for ($i = 0; $i < $length; $i++)
{
    $randomString .= $characters[ rand(0, strlen($characters) - 1) ];
}

return $randomString;

private function generatePublicID()
{

public = $this->generateRandomString($length);

private function generateRandomString($length)
{

$characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';

$randomString = ''; 

for ($i = 0; $i < $length; $i++)
{
    $randomString .= $characters[ rand(0, strlen($characters) - 1) ];
}

return $randomString;

private function generateRandomString($length)
{

$characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';

$randomString = ''; 

for ($i = 0; $i < $length; $i++)
{
    $randomString .= $characters[ rand(0, strlen($characters) - 1) ];
}

return $randomString;
$this->db->where( 'user_id', $id );
$query3 = $this->db->get('user');
$guard = $query3->row();
if($guard->profile_picture != "blank.png"){
    $path = "assets/users/" . $guard->profile_picture;
    unlink($path);
}
$this->db->where( 'user_id', $id );
$this->db->delete('user');

public function deleteAllPatients() {
    $this->db->where( 'user_type', 1 );
    $this->db->delete('user');
    $this->db->delete('user');
    $this->db->delete('user');
}

public function checkOTPatients($id) {
    $query = $this->db->query('SELECT * FROM patients WHERE user_id = ' . $id);
    $row = $query->row();
    if($row->profile_picture != "blank.png"){
        $path = "assets/users/" . $row->profile_picture;
        unlink($path);
    }
    $this->db->where( 'user_id', $id );
    $this->db->delete('user');
}

public function countUserRequests($user, $limit, $start) {
    $array = $this->db->where( 'user_id', $user )->limit($limit, $start)->get('user');
    return $array->num_rows();
}

public function getUserRequests($user, $limit, $start) {
    $array = $this->db->where( 'user_id', $user )->limit($limit, $start)->get('user');
    return $array;
}

public function changeReminderStatus($id) {
    $this->db->where('reminder_id', $id);
    $query = $this->db->get('reminder');
    $row = $query->row();
    if($row->status == 0) {
        $data = array('status' => 0);
        $this->db->where('reminder_id', $id);
        $this->db->update('reminder', $data);
        $dateSub = date("Y-m-d");
        $this->submit($row->user_id, 2, $id, $dateSub);
    }
}

public function getUser($user) {
    $array = $this->db->where('user_id', $user)->get('user');
    return $array;
}

public function acceptUser($user) {
    $data = array('status' => 1);
    $this->db->where('user_id', $user)->update('user', $data);
}

public function updateTherapist($patient, $therapist) {
    for($i=0; $i<count($patients); $i++) {
        $data = array('therapist_id' => $therapist[$i]);
        $this->db->where('user_id', $patients[$i])->update('patient', $data);
    }
}

public function getTherapist($patient) {
    $data = array('therapist_id' => $this->db->get('therapist'));
    $this->db->update('patient', $data);
}

public function deleteOldReminders($id, $field) {
    $this->db->where( $field, $id );
    $query = $this->db->get('reminder');
    foreach($query->result() as $row) {
        if($row->status == 0) {
            $this->db->where('submission_type_id', $row->id)->delete('reminder');
        }
    }
}

function getPassword($row) {
    $password = $row->password;
    return $password;
}

function updatePassword($row) {
    $password = $row->password;
    $this->db->update('user', $password);
    return $password;
}

function deleteReminder($row) {
    $this->db->delete('reminder');
    return $row;
}

function num_users() {
    $query = $this->db->get('user');
    return $query->count();
}

function deleteOldReminders($id, $field) {
    $this->db->where( $field, $id );
    $query = $this->db->get('reminder');
    foreach($query->result() as $row) {
        if($row->status == 0) {
            $this->db->where('submission_type_id', $row->id)->delete('reminder');
        }
    }
}

function changeReminderStatus($id) {
    $this->db->where('reminder_id', $id);
    $query = $this->db->get('reminder');
    $row = $query->row();
    if($row->status == 0) {
        $data = array('status' => 0);
        $this->db->where('reminder_id', $id);
        $this->db->update('reminder', $data);
        $dateSub = date("Y-m-d");
        $this->submit($row->user_id, 2, $id, $dateSub);
    }
}

function getUserRequests($user, $limit, $start) {
    $array = $this->db->where( 'user_id', $user )->limit($limit, $start)->get('user');
    return $array;
}

function getUser($user) {
    $array = $this->db->where('user_id', $user)->get('user');
    return $array;
}

function acceptUser($user) {
    $data = array('status' => 1);
    $this->db->where('user_id', $user)->update('user', $data);
}

function updateTherapist($patient, $therapist) {
    for($i=0; $i<count($patients); $i++) {
        $data = array('therapist_id' => $therapist[$i]);
        $this->db->where('user_id', $patients[$i])->update('patient', $data);
    }
}

function getTherapist($patient) {
    $data = array('therapist_id' => $this->db->get('therapist'));
    $this->db->update('patient', $data);
}

function deleteOldReminders($id, $field) {
    $this->db->where( $field, $id );
    $query = $this->db->get('reminder');
    foreach($query->result() as $row) {
        if($row->status == 0) {
            $this->db->where('submission_type_id', $row->id)->delete('reminder');
        }
    }
}

function changeReminderStatus($id) {
    $this->db->where('reminder_id', $id);
    $query = $this->db->get('reminder');
    $row = $query->row();
    if($row->status == 0) {
        $data = array('status' => 0);
        $this->db->where('reminder_id', $id);
        $this->db->update('reminder', $data);
        $dateSub = date("Y-m-d");
        $this->submit($row->user_id, 2, $id, $dateSub);
    }
}

function getPassword($row) {
    $password = $row->password;
    return $password;
}

function updatePassword($row) {
    $password = $row->password;
    $this->db->update('user', $password);
    return $password;
}

function deleteReminder($row) {
    $this->db->delete('reminder');
    return $row;
}

function num_users() {
    $query = $this->db->get('user');
    return $query->count();
}

function deleteOldReminders($id, $field) {
    $this->db->where( $field, $id );
    $query = $this->db->get('reminder');
    foreach($query->result() as $row) {
        if($row->status == 0) {
            $this->db->where('submission_type_id', $row->id)->delete('reminder');
        }
    }
}
Listing 9: main.php

```php
<?php

public function deleteCategory($id)
{
    $this->db->where('category_id', $id);
    $query = $this->db->get('category');
    $row = $query->row();
    if ($row)
    {
        $array = array('category_id', $id);
        $this->db->delete('category', $array);
    } else {
        $this->session->set_flashdata('error', 'Category does not exist!');
    }
}

public function countUserSubmissions($user, $type)
{
    if ($type == 'journal')
    {
        $this->db->where('author_id', $user);
        $query = $this->db->get('journal');
        $result = $query->result();
        return $query->num_rows();
    } elseif ($type == 'photo')
    {
        $this->db->where('author_id', $user);
        $query = $this->db->get('photo');
        $result = $query->result();
        return $query->num_rows();
    } elseif ($type == 'question')
    {
        $this->db->where('author_id', $user);
        $query = $this->db->get('question');
        $result = $query->result();
        return $query->num_rows();
    } else {
        $this->db->where('user_id', $user);
        $query = $this->db->get('category');
        $result = $query->result();
        return $result;
    }
}

public function getCategories($user)
{
    $this->db->where('user_id', $user);
    $query = $this->db->get('category');
    return $query->result();
}

public function addCategory($user, $name)
{
    $array = array(
        'user_id' => $user,
        'name' => $name
    );
    $this->db->insert('category', $array);
}

public function getuserSubmissions($user, $type, $limit, $start)
{
    $this->db->where('author_id', $user);
    switch($type)
    {
    case 1:
        $query = $this->db->get('journal');
        break;
    case 2:
        $query = $this->db->get('reminder');
        break;
    case 3:
        $query = $this->db->get('photo');
        break;
    case 4:
        $query = $this->db->get('question');
        break;
    }
    return $query->result();
}

public function getUserSubmissions($user, $type)
{
    $this->db->where('user_id', $user);
    $this->db->order_by('date_created', 'desc');
    switch($type)
    {
    case 1:
        $query = $this->db->get('journal');
        break;
    case 2:
        $query = $this->db->get('reminder');
        break;
    case 3:
        $query = $this->db->get('photo');
        break;
    case 4:
        $query = $this->db->get('question');
        break;
    }
    return $query->result();
}

public function getUserSubmissions($user, $author_id, $limit, $start)
{
    $this->db->where('author_id', $author_id);
    $this->db->order_by('date_created', 'desc');
    switch($type)
    {
    case 1:
        $query = $this->db->get('journal');
        break;
    case 2:
        $query = $this->db->get('reminder');
        break;
    case 3:
        $query = $this->db->get('photo');
        break;
    case 4:
        $query = $this->db->get('question');
        break;
    }
    return $query->result();
}

```

<!DOCTYPE html>
<html>
<head>
    <title>Anteroade Amnesia Information System</title>
    <link href="/assets/css/bootstrap.min.css" rel="stylesheet">
    <link href="/assets/css/app.css" rel="stylesheet">
</head>
<body>
    <!-- Your code goes here -->
</body>
</html>
```
Listing 10: home.php

```php
<!--DOCUMENT html -->
<html>
<head>
<title>Anterograde Amnesia Information System</title>
</head>
<body>
<body>
</body>
</html>
```
```php
break;

case 1:
    ?
    <li class="glyphicon glyphicon-th-large aria-hidden="true">"ophotos"
        &nbsp; &nbsp; &nbsp; Photo</li>
    </ul>
</div>

<!doctype html>
<html>
<head>
    <title>Activecare Patient Portal</title>
    <meta charset="utf-8">
</head>
<body>
    <nav class="navbar navbar-default navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <a class="navbar-brand" href="\base_url\user\photos">Photos</a>
            </div>
            <ul class="nav navbar-nav">
                <li class="active"><a href="\base_url\user\generalKnowledge">General Knowledge</a></li>
                <li><a href="\base_url\user\journal">Journal</a></li>
                <li><a href="\base_url\user\reminders">Reminders</a></li>
                <li><a href="\base_url\user\questions">Questions</a></li>
                <li><a href="\base_url\user\photoRecognition">Photo Recognition</a></li>
                <li><a href="\base_url\user\report">Report</a></li>
            </ul>
            <button class="navbar-toggle" type="button">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
        </div>
    </nav>

    <div class="container">
        <div class="row">
            <div class="col-md-9">
                <h1>Welcome to Activecare Portal</h1>
                <p>This is your personal portal to manage your health information and access various functionalities provided by Activecare.</p>
            </div>
            <div class="col-md-3">
                <aside>
                    <h2>Quick Links</h2>
                    <ul>
                        <li><a href="#">General Knowledge</a></li>
                        <li><a href="#">Journal</a></li>
                        <li><a href="#">Reminders</a></li>
                        <li><a href="#">Questions</a></li>
                        <li><a href="#">Photo Recognition</a></li>
                        <li><a href="#">Report</a></li>
                    </ul>
                </aside>
            </div>
        </div>
    </div>

    <footer class="navbar navbar-inverse">
        <div class="container">
            <p>&copy; 2023 Activecare. All rights reserved. Privacy Policy & Terms of Service</p>
        </div>
    </footer>
</body>
</html>
```
case 3:

```php
if ( $cookie == true ) {
    $cookie = json_decode( $cookie, true );
    echo '<div class="container-fluid">';
    echo '<div class="row"><div class="blackPanel">';
    echo '<div class="row">';
    echo '</div></div>';  // Page Heading
    echo '</div>';  // Page Row
    echo '</div> <!-- Container-fluid -->
    <!-- Page Content -->
    <div id="page-wrapper" class="highlighted">';
    echo '<div id="page-footer">';
    echo '</div>';  // Page Footer
}
```

Listing 11: public/main.php

```php
if ( ! $this->IS_AAISS ) {  // The public controller
    return;
}
```

Listing 12: public.php
Listing 13: login.php

```php
<?php

patientID = $POST["patientID"]; 
if (isset($patientID))

foreach ($patients as $patient) {
    if ($patient->public_id == $patientID) 
        $name = $patient->first_name . ", " . $patient->last_name;

$array = array(
    "isLoggedln" => 1,
    "user_id" => $patient->user_id,
    "name" => $patient->first_name . ", " . $patient->last_name,
    "step" => 1,
    "pageType" => 1,
    "error" => 0
);

$this->AAIS_model->setCookie($array, 
    thePublicCookie);

redirect(base_url(‘public’));

}

if ($here == 0) {

$array = array( 
    "error" => 2);

$this->AAIS_model->setCookie($array, 
    thePublicCookie);

$cookie = json_decode(json_encode($array));

redirect(base_url());

}

if (!isLoggedln) 
    include_once(‘requestAccount.php’);

else {

    if ($publicCookie = $this->AAIS_model->
        deleteCookie(‘thePublicCookie’))
        redirect(base_url());
    }

Listing 14: request.php

```php

Listing 15: requestAccount.php

```php

```
Listing 16: sysad/dashboard.php

```php
<h1 class="page-header">Dashboard <small>Users Overview</small></h1>

<h4 class="line"><span>System Administrators</span></h4>

```
Listing 17: sysad/editAccount.php

$1 = 0;
foreach ($connections as $c) {
    $1++;
    <div class="col-md-3">
    <div class="panel panel-default">
        <table style="font-size: 14px;">
            <col width="100">
            <col width="10">
            <col width="300">
        </table>
    </div>
</div>
</div>

<form method="POST" action="/sysad/editAccount/stepTwo/" enctype="multipart/form-data">
    <div class="row">
        <div class="col-md-3">
            <div class="panel panel-default">
                <table>
                    <div>
                        <div>
                            <php
                            IF ($1%2 == 1)
                        <div class="col-md-3" style="border-right: 1 solid #b00101;">
                            <br>
                        </div>
                    </div>
                    else
                <div class="col-md-3">
                    If ($1%2 == 1) {
                    <div class="col-md-3" style="border-right: 1 solid #b00101;">
                        <br>
                    </div>
                </div>
        </table>
    </div>
</div>
    </php>
    else
    
    <div class="col-md-3">
        <div class="form-group">
            <input type="text" name="firstName" id="firstName">
            <input type="text" name="lastName" id="lastName">
            <input type="password" name="password" id="password">
            <input type="password" name="confirmPassword" id="confirmPassword">
            <input type="submit" value="Submit">
        </div>
    </div>
</form>

have successfully edited your account.
</div>

<?php
}
>

<form method="POST" action="/sysad/editAccount/stepTwo/" enctype="multipart/form-data">
    <div class="row">
        <div class="col-md-3">
            <div class="panel panel-default">
                <table>
                    <tr>
                        <td rowspan='2'><img src="/assets/users/" . $c['profile_picture'] />
                    </td>
                    <td><php echo $c['patient'] ?></td>
                    <td><php echo $c['patient'] ?></td>
                    <td><php echo $c['patient'] ?></td>
                    <td><php echo $c['patient'] ?></td>
                </tr>
                </div>
            
        </div>
    </div>
</form>

Listing 17: sysad/editAccount.php

<div class="pad">
    <h1 class="page-header">Edit Account <small>System Administrator</small></h1>
    </div>

    <?php
    If ($cookie->error == 1) {
        $cookie->error = 0;
        $this->AAAS_module->setCookie( array $cookie, 'theUserCookie' );
    }
    <div class="alert alert-danger" role="alert">
        <span class="glyphicon glyphicon-exclamation-sign" aria-hidden="true"></span>
        <span class="sr-only">Error: </span>
    Please check your input. All fields are required</div>
    <div>
    </div>

    If ($cookie->error == 2) {
        $cookie->error = 0;
        $this->AAAS_module->setCookie( array $cookie, 'theUserCookie' );
    }
    <div class="alert alert-success" role="alert">
        <span class="glyphicon glyphicon-ok-circle" aria-hidden="true"></span>
        <span class="sr-only">Success!</span>
    You have successfully edited your account.
</div>

<?php
}
Listing 18: addUser/main.php

```php
Listing 18: addUser/main.php
```

Listing 19: addUser/addPatient.php

```php
Listing 19: addUser/addPatient.php
```

Listing 20: addUser/addGuardian.php

```php
Listing 20: addUser/addGuardian.php
```
Listing 21: addUser/selectOT.php

```php
<div class="progress">
  <div class="progress-bar progress-bar-striped progress-bar-active" role="progressbar" style="width: 60%;"></div>
</div>

Step 3: Assign a Therapist
</div>
</div>

<form method="POST" action="/sysad/addUser/stepFour/'">
  <div class="form-group">
    <label for="firstName">First Name</label>
    <input type="text" name="firstName" id="firstName" class="form-control" value=""><span class="glyphicon glyphicon-arrow-left" title="aria-hidden=true"></span></div>
  </div>

  <div class="form-group">
    <label for="lastName">Last Name</label>
    <input type="text" name="lastName" id="lastName" class="form-control" value=""><span class="glyphicon glyphicon-arrow-left" title="aria-hidden=true"></span></div>
  </div>

  <div class="form-group">
    <label for="email">Email Address</label>
    <input type="email" name="email" id="email" class="form-control" value=""><span class="glyphicon glyphicon-arrow-left" title="aria-hidden=true"></span></div>
  </div>

  <div class="form-group">
    <label for="gender">Gender</label>
    <select name="gender" id="gender">
      <option value="M">Male</option>
      <option value="F">Female</option>
    </select>
  </div>

  <div class="form-group">
    <label for="birthday">Birthday</label>
    <input type="date" name="birthday" id="birthday" class="form-control" value=""><span class="glyphicon glyphicon-arrow-left" title="aria-hidden=true"></span></div>
  </div>

  <div class="form-group">
    <label for="email">Choose a Cookie</label>
    <input type="email" name="email" id="email" class="form-control" value=""><span class="glyphicon glyphicon-arrow-left" title="aria-hidden=true"></span></div>
  </div>

  <div class="form-group">
    <label for="theapist">Theapist</label>
    <select name="theapist" id="theapist">
      <option value="">Select a Therapist</option>
      <?php
      foreach ($therapists as $therapist){
        if ($therapist->user_id == $addCookie->ot_id){
          echo "<option value="." therapist->user_id."." value="." selected="selected">". therapist->first_name."." therapist->last_name."." option="";}
        }else{
          echo "<option value="." therapist->user_id."." value="">". therapist->first_name."." therapist->last_name."." option="";}
      }else{ ?>
  </div>

  <div class="form-group">
    <button type="submit" class="btn btn-primary">Submit</button>
  </div>
</form>


```

```html

```
Listing 24: addOT/main.php

```php
<h1>Page header</h1>
```

Listing 23: addUser/generate.php

```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
You can change your username and password when you login. Another account can be added.

Listing 25: addOT/generate.php

```php
<form method="POST" action="/sysad/addSysad/stepTwo/">
  <div class="col-md-6">
    <div class="form-group">
      <label for="username">Username</label>
      <input type="text" name="username" id="username" class="form-control">
    </div>
  </div>
  
  <div class="col-md-6">
    <div class="form-group">
      <label for="password">Password</label>
      <input type="password" name="password" id="password" class="form-control">
    </div>
  </div>

  <p>Step 1: Input System Admin Information</p>
</form>

Listing 26: addSysad/main.php

```
Listing 27: addSysad/generate.php

```php
<?php
    $this->redirect('sysad/assignNewTherapist.php');
</td>
</tr>
</table>
</div>
</div>
</div>
</div>
</body>
</html>

Listing 28: edit/main.php

```php
    switch($cookie->step) {
        case 1: include_once('editPublicID.php'); break;
        case 2: include_once('assignTherapist.php'); break;
        default: break;
    }
</div>
```

Listing 29: edit/assignTherapist.php

```php
if ($cookie->error == 2) {
    $this->redirect('sysad/addSysad.php');
}
```

Listing 30: delete/main.php

```php
    switch($cookie->step) {
        case 1: include_once('deletePatient.php'); break;
        case 2: include_once('deleteGuardian.php'); break;
        case 3: include_once('deleteOT.php'); break;
        case 4: include_once('deleteSysad.php'); break;
        default: break;
    }
```
### Listing 32: delete/deleteGuardian.php

```php
<form method="POST" action="/sysad/deletePatient/stepTwo/">
  <input type="hidden" name="user_id" value="@echo $row->user_id@" />
  <button class="btn btn-default" type="submit" aria-hidden="true"><span class="glyphicon glyphicon-trash"></span></button>
</form>

<?php
if (count($patients) == 0) {
  echo '<td colspan="3">No search result found.</td></tr>';
}

<?php
foreach ($patients as $row) {
  // Code to display patient information
  $i++; if ($i == 103) break;
}

<?php
foreach ($guardians as $row) {
  // Code to display guardian information
  $i = 0;
  if ($i > 0) {
    echo '<td class="left">Id</td><td class="left">Name</td><td class="left">Surname</td><br/>';
  } else {
    echo '<td colspan="3">No search result found.</td></tr>';
  }

  foreach ($patients as $row) {
    // Code to determine if patient is linked to the guardian
    if ($row->linked) {
      echo '<td class="warning">This patient is linked to this guardian.</td>';
    } else {
      echo '<td class="danger">This patient is not linked to this guardian.</td>';
    }
  }

  // Code to display delete button
  if ($row->linked) {
    echo '<button class="btn btn-danger" type="submit" aria-hidden="true"><span class="glyphicon glyphicon-trash"></span></button>
  }

  // Code to display search form
  // Code to display patient table
</form>
```

```html
Listing 32: delete/deleteGuardian.php

<form method="POST" action="/sysad/deletePatient/stepTwo/">
  <input type="hidden" name="user_id" value="@echo $row->user_id@" />
  <button class="btn btn-default" type="submit" aria-hidden="true"><span class="glyphicon glyphicon-trash"></span></button>
</form>

<?php
if (count($patients) == 0) {
  echo '<td colspan="3">No search result found.</td></tr>';
}

<?php
foreach ($patients as $row) {
  // Code to display patient information
  $i++; if ($i == 103) break;
}

<?php
foreach ($guardians as $row) {
  // Code to display guardian information
  $i = 0;
  if ($i > 0) {
    echo '<td class="left">Id</td><td class="left">Name</td><td class="left">Surname</td><br/>';
  } else {
    echo '<td colspan="3">No search result found.</td></tr>';
  }

  foreach ($patients as $row) {
    // Code to determine if patient is linked to the guardian
    if ($row->linked) {
      echo '<td class="warning">This patient is linked to this guardian.</td>';
    } else {
      echo '<td class="danger">This patient is not linked to this guardian.</td>';
    }
  }

  // Code to display delete button
  if ($row->linked) {
    echo '<button class="btn btn-danger" type="submit" aria-hidden="true"><span class="glyphicon glyphicon-trash"></span></button>
  }

  // Code to display search form
  // Code to display patient table
</form>
```
```
Listing 33: delete/newGuardian.php

```php
<?php

<div class="pad">
<h1 class="page-header">Delete User <small>Guardian</small></h1>
<h3>Create New Guardian</h3>
</div>

<form method="POST" action="<?php echo base_url('sysadm/deleteGuardian/newStepTwo') ?>">
  <div class="row">
    <div class="col-md-3">
      <div class="panel panel-default">
        <div class="panel-body">
          <table class="add">
            <col width="150">
            <col width="500">
            <tr>
              <td>First Name</td>
              <td><input type="text" name="firstName" id="firstName" class="form-control"></td>
            </tr>
            <tr>
              <td>Last Name</td>
              <td><input type="text" name="lastName" id="lastName" class="form-control"></td>
            </tr>
            <tr>
              <td>Sex</td>
              <td>
                <select name="sex" id="sex" class="form-control">
                  <option value="M">Male</option>
                  <option value="F">Female</option>
                </select>
              </td>
            </tr>
            <tr>
              <td>Birthday</td>
              <td><input type="date" name="birthday" id="birthday" class="form-control"></td>
            </tr>
            <tr>
              <td>Email Address</td>
              <td><input type="text" name="email" id="email" class="form-control"></td>
            </tr>
          </table>
        </div>
      </div>
    </div>
  </div>
  <div class="col-md-9">
    <button class="btn btn-primary" type="submit">Submit</button>
  </div>
</form>
```

Listing 34: delete/generateNew-Guardian.php

```php
<?php

<div class="pad">
<h1 class="page-header">Delete User <small>Guardian</small></h1>
<h3>New Account</h3>
</div>

<div class="row">
  <div class="col-md-8">
    <div class="panel panel-default">
      <div class="panel-body">
        <h3 class="title">USER NAME &nbsp; &nbsp; &nbsp; PASSWORD</h3>
        <form method="POST" action="<?php echo base_url('sysadm/deleteOT/search'); ?>">
          <input type="text" name="username" id="username" class="form-control">
          <input type="password" name="password" id="password" class="form-control">
          <input type="submit" value="Submit" class="btn btn-primary" />
        </form>
      </div>
    </div>
  </div>
</div>
```

Listing 35: delete/deleteOT.php

```php
<?php

<div class="pad">
<h1 class="page-header">Delete User <small>Therapist</small></h1>
</div>

<div class="row">
  <div class="col-md-6">
    <div class="panel panel-default">
      <div class="panel-body">
        <a href="<?php echo base_url('sysadm/deleteGuardian') ?>">Delete another guardian</a>
      </div>
    </div>
  </div>
</div>
```
Listing 36: delete/newOT.php

```php
<?php
$the_therapists = $search == 1 ? $therapists : array_filter($therapists, function ($row) { return $row['id'] != $current; });
foreach ($the_therapists as $row) {
    $first = $row['first_name'] . ' ' . $row['last_name'];
    $controller = 'user_id';
    $id = $row['id'];
    $user_id = $row['id'];
    $user_name = $row['username'];
    echo $first, ' ', $user_name, ' ', $id, $controller, ' ', $user_id, ' $id != $current );
    $button = $row['id'];
    echo $first, ' ', $user_name, ' ', $id, $controller, ' ', $user_id, ' $id != $current );
}</td>
</table>
</div>
</div>
</div>
</div>

Listing 37: delete/generateNewOT.php

```php
<?php
foreach ($patients as $row) {
    $first = $row['first_name'] . ' ' . $row['last_name'];
    $controller = 'user_id';
    $id = $row['id'];
    $user_id = $row['id'];
    $user_name = $row['username'];
    $user_id = $row['id'];
    $user_name = $row['username'];
    $first = $row['first_name'] . ' ' . $row['last_name'];
    $controller = 'user_id';
    $id = $row['id'];
    $user_id = $row['id'];
    $user_name = $row['username'];
    $user_id = $row['id'];
    $user_name = $row['username'];
    $first = $row['first_name'] . ' ' . $row['last_name'];
    $controller = 'user_id';
    $id = $row['id'];
    $user_id = $row['id'];
    $user_name = $row['username'];
    $user_id = $row['id'];
    $user_name = $row['username'];
}</td>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
```
Listing 38: delete/deleteSysad.php

```
<?php
$count = count($row);
if($count > 0) {
    echo "You have successfully requested a new public ID for the patient.";
}
else {
    echo "No search result found.";
}
?>
```

Listing 39: user/editAccount.php

```
<?php
$cookie = $this->theUserCookie();
if($cookie->error == 2) {
    $cookie->error = 0;
    $this->AIS_model->setCookie( array $cookie, 'theUserCookie' );
}
else if($cookie->error == 3) {
    $cookie->error = 0;
    $this->AIS_model->setCookie( array $cookie, 'theUserCookie' );
}
else {  // this->error == 1
    $this->AIS_model->setCookie( array $cookie, 'theUserCookie' );
}
?>
```
Listing 40: create/main.php

```php
<div class="nav nav-tabs">
  <li role="presentation" class="<?php if($entry->pageType == 1) echo 'active'; ?>"><a href="/table"><span class="glyphicon glyphicon-pencil" aria-hidden="true"></span> Journal</a></li/>
  <li role="presentation" class="<?php if($entry->pageType == 2) echo 'active'; ?>"><a href="/user/create/2/">Photo</a></li>
  <li role="presentation" class="<?php if($entry->pageType == 3) echo 'active'; ?>"><a href="/user/create/3/">Reminder</a></li/>
  <li role="presentation" class="<?php if($entry->pageType == 4) echo 'active'; ?>"><a href="/user/create/4/">Memory</a></li/>
</div/>

<?php
if($cookie->user_type == 2){

} else{


case 1: echo "Journal entries are accounts of recent events that are relevant to the patient. When submitting an entry, try to make the little as informative and possible. Also, try to make the details as detailed as possible so that the doctor will avoid confusing the patient.";
break;

case 2: echo "Activity reminders are very helpful for patients who remember their appointments and/or activities to be done in the coming days/weeks/months. When submitting a reminder, make all the relevant details including the time, who, what, when, where, and who as much as possible.

break;

case 3: echo "Photos are very important for patients who want visual representations and/or evidence of the existence of things they have to do.

break;

case 4: echo "Memory Text Questions can be of two types: (1) Personal, which are questions about the personal level of the patient, and (2) General Knowledge, which are questions about the general public such as the name of the current president of the Philippines.

break;

}?>
</div/>
</div>
</div>
</div/>
</div>
</div>
</body>
</html>
```

107
Listing 41: create/journal.php

```php
<?php
$entry->error = 0;
$this->>AAIS_model->setCookie( array $entry, 'theSubmissionCookie' );
}
else if($entry->error==3)
{
  ?php
  error( $id ) ;
  foreach( $categories as $c )
  {
  echo $c
  }
  ?>
  <form method="POST" action="/create/journal.php">
  <label for="type">Type</label>
  <select name="category" id="category" class="form-control">
    <option value="General Knowledge">General Knowledge</option>
    <option value="Personal Questions">Personal Questions</option>
    <option value="Choice A">Choice A</option>
    <option value="Choice B">Choice B</option>
    <option value="Choice C">Choice C</option>
    <option value="Choice D">Choice D</option>
  </select>
  <label for="details">Details</label>
  <textarea name="details" id="details" class="form-control" rows="8"></textarea>
  <br>
  <label for="title">Title</label>
  <input type="text" name="title" id="title" class="form-control">
  </br>
  <label for="date">Date and Time</label>
  <input type="date" name="date" id="date" class="form-control">
  </br>
  <label for="answer">Answer</label>
  <textarea name="answer" id="answer" class="form-control" rows="10"></textarea>
  </br>
  <label for="question">Question</label>
  <input type="text" name="question" id="question" class="form-control">
  </br>
  <label for="answer">Correct Answer</label>
  <textarea name="answer" id="answer" class="form-control" rows="10"></textarea>
  </br>
  <button type="submit" class="btn-primary">Submit</button>
  </form>
<?php
}
Listing 42: create/photo.php

```asp
<?php
<?php
Listing 43: create/question.php

```html
<form method="POST" action="/create/question.php">
  <label for="question">Question</label>
  <input type="text" name="question" id="question" class="form-control">
  </br>
  <label for="answer">Correct Answer</label>
  <textarea name="answer" id="answer" class="form-control" rows="10"></textarea>
  </br>
  <button type="submit" class="btn-primary">Submit</button>
  </form>
<?php
Listing 44: create/remind.php

```html
<form method="POST" action="/create/remind.php">
  <label for="date">Date and Time</label>
  <input type="date" name="date" id="date" class="form-control">
  </br>
  <button type="submit" class="btn-primary">Submit</button>
  </form>
<?php
```
Listing 45: create/success.php

```php
<div class="col-md-2"></div>
</center>
</div>
</div>
</div>
</center>
</div>
</div>
</div>
</div>
</div>
</div>
<h3>You have successfully submitted an entry</h3>
<a href="<?php base_url('user/create/').'.'.$type?>">Submit another entry.</a>
</div>
</div>
</center>
</div>
</div>
</div>
</div>
<h4>Listing 46: journals/main.php

```php
<form action="<?php echo base_url('user/journals');?>" method="POST">  
<input type="hidden" name="id" value="" />
<input type="hidden" name="category" value="" />
<input type="hidden" name="journal_id" value="" />
<input type="hidden" name="journal_title" value="" />
<input type="hidden" name="journal_type" value="" />
<input type="hidden" name="journal_status" value="" />
<input type="hidden" name="journal_date" value="" />
<input type="hidden" name="journal_created" value="" />
<input type="hidden" name="journal_updated" value="" />
<input type="hidden" name="journal_deleted" value="" />
</form>
</div>
</div>
</div>
</div>
</div>
</div>
<h4>Listing 47: journals/edit.php

```php
<form method="POST" action="<?php base_url('user/editJournal');?>" enctype="multipart/form-data">  
<h1>Journal Entries</h1>
<h4 class="page-header">View, edit, delete journal entries</h4>
<div class="col-md-12">
<h4>Listing 47: journals/edit.php

```php
<form method="POST" action="<?php base_url('user/journals');?>" enctype="multipart/form-data">  
<h1>Journal Entries</h1>
<h4 class="page-header">View, edit, delete journal entries</h4>
<div class="col-md-12">
<h4>Listing 47: journals/edit.php

```php
<form method="POST" action="<?php base_url('user/journals');?>" enctype="multipart-form-data">  
<h1>Journal Entries</h1>
<h4 class="page-header">View, edit, delete journal entries</h4>
<div class="col-md-12">
<h4>Listing 47: journals/edit.php

```php
<form method="POST" action="<?php base_url('user/journals');?>" enctype="multipart-form-data">  
<h1>Journal Entries</h1>
<h4 class="page-header">View, edit, delete journal entries</h4>
<div class="col-md-12">
<h4>Listing 47: journals/edit.php

```php
<form method="POST" action="<?php base_url('user/journals');?>" enctype="multipart-form-data">  
<h1>Journal Entries</h1>
<h4 class="page-header">View, edit, delete journal entries</h4>
<div class="col-md-12">
<h4>Listing 47: journals/edit.php

```php
<form method="POST" action="<?php base_url('user/journals');?>" enctype="multipart-form-data">  
<h1>Journal Entries</h1>
<h4 class="page-header">View, edit, delete journal entries</h4>
<div class="col-md-12">
<h4>Listing 47: journals/edit.php

```php
<form method="POST" action="<?php base_url('user/journals');?>" enctype="multipart-form-data">  
<h1>Journal Entries</h1>
<h4 class="page-header">View, edit, delete journal entries</h4>
<div class="col-md-12">
<h4>Listing 47: journals/edit.php

```php
<form method="POST" action="<?php base_url('user/journals');?>" enctype="multipart-form-data">  
<h1>Journal Entries</h1>
<h4 class="page-header">View, edit, delete journal entries</h4>
<div class="col-md-12">
<h4>Listing 47: journals/edit.php

```php
<form method="POST" action="<?php base_url('user/journals');?>" enctype="multipart-form-data">  
<h1>Journal Entries</h1>
<php>
if ($journal->error==1){
    $journal->error = 0;
    $this->AAS_model->setCookie( (array) $journal , 'theEntryCookie' );
    ?><div class="alert alert-danger role="alert">
    <span class="glyphicon exclamation-sign" aria-hidden="true"></span>&lt;/span&gt;&lt;span class="sr-only">Error:&lt;/span&gt; Please check your input. All fields are required
    </div>&lt;/php&gt;
} else if ($journal->error==2){
    $journal->error = 0;
    $this->AAS_model->setCookie( (array) $journal , 'theEntryCookie' );
    ?><div class="alert alert-success role="alert">
    <span class="glyphicon ok-circle" aria-hidden="true"></span>&lt;/span&gt;&lt;span class="sr-only">Success!&lt;/span&gt; You have sucessfully edited this entry.
    </div>&lt;/php&gt;
} else if ($journal->error==3){
    $journal->error = 0;
    $this->AAS_model->setCookie( (array) $journal , 'theEntryCookie' );
    ?><div class="alert alert-danger role="alert">
    <span class="glyphicon exclamation-sign" aria-hidden="true"></span>&lt;/span&gt;&lt;span class="sr-only">Invalid date!&lt;/span&gt;
    </div>&lt;/php&gt;
} else {<br>
    $journal->error = 0;
    $this->AAS_model->setCookie( (array) $journal , 'theEntryCookie' );
    ?><div class="alert alert-danger role="alert">
    <span class="glyphicon exclamation-sign" aria-hidden="true"></span>&lt;/span&gt;&lt;span class="sr-only">Error:&lt;/span&gt; Invalid date!&lt;/span&gt;
    </div>&lt;/php&gt;
<?
</div>
<h3 class="title">&lt;center&gt;J O U R N A L &nbsp; I N F O R M A T I O N&lt;/center&gt;&lt;/h3&gt;
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>&lt;h1 class="small"&gt;Photos &lt;small&gt;View, edit, delete journal photos&lt;/small&gt;&lt;/h1&gt;
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>&lt;form method="POST" action="&lt;a href="user/search/1"&gt;Search&lt;/a&gt;"
</div>&lt;div class="form-group"&gt;
&lt;label for="category"&gt;&lt;span class="glyphicon glyphicon-tag" aria-hidden="true"></span>&lt;/span&gt;&lt;span class="sr-only">Select a category&lt;/span&gt;
    &lt;option value="all"&gt;All&lt;/option&gt;
    &lt;php foreach($categories as $c){&lt;/php&gt;
        &lt;option value="&lt;php echo $c-&gt;name?&gt;"&gt;&lt;php echo $c-&gt;name?&gt;&lt;/option&gt;
    &lt;/php)&gt;
    &lt;/select&gt;&lt;/div&gt;
&lt;div type="submit" class="btn btn-primary"&gt;Filter &lt;/button&gt;
&lt;/form&gt;
&lt;br&gt;&lt;br&gt;
&lt;div class="panel panel-info2"&gt;
    &lt;div class="panel-body"&gt;
    Photos are very important to visually represent the existence of memories that can help patients remember more of the things that they have to. You can edit photos to add or edit information about the photo by clicking on the pen button below the one you wish to edit. You can also delete photos by clicking on the trash button below the one you wish to delete.
    &lt;br&gt;Please be reminded that this is a collection of entries that were created/submitted either by the patient, his guardian, his therapist, and the general public that were approved by the guardian
    &lt;/div&gt;
    &lt;/div&gt;
    &lt;/div&gt;
</div>&lt;div class="col-md-8"&gt;
    &lt;h4&gt;Filter: &lt;php echo $cookie-&gt;category?&gt;&lt;/h4&gt;
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>&lt;option value="&lt;php echo &lt;php echo $name2?&gt;"&gt;&lt;php echo $name2?&gt;&lt;/option&gt;
</div>&lt;/php&gt;
</div>
</div>
</div>
</div>
</div>
</div>
</div>&lt;option value="&lt;php echo &lt;php echo $name?&gt;"&gt;&lt;php echo $name?&gt;&lt;/option&gt;
</div>&lt;/php&gt;
</div>
</div>
</div>
</div>
</div>
</div>&lt;option value="&lt;php echo &lt;php echo $name?&gt;"&gt;&lt;php echo $name?&gt;&lt;/option&gt;
</div>&lt;/php&gt;
</div>
</div>
</div>
</div>
</div>
</div>
</div>&lt;option value="&lt;php echo &lt;php echo $name?&gt;"&gt;&lt;php echo $name?&gt;&lt;/option&gt;
</div>&lt;/php&gt;
</div>
</div>
</div>
</div>
</div>
</div>
</div>&lt;option value="&lt;php echo &lt;php echo $name?&gt;"&gt;&lt;php echo $name?&gt;&lt;/option&gt;
</div>&lt;/php&gt;
</div>
</div>
</div>
</div>
</div>
</div>
</div>&lt;option value="&lt;php echo &lt;php echo $name?&gt;"&gt;&lt;php echo $name?&gt;&lt;/option&gt;
</div>&lt;/php&gt;
</div>
</div>
</div>
</div>
</div>
</div>
</div>&lt;option value="&lt;php echo &lt;php echo $name?&gt;"&gt;&lt;php echo $name?&gt;&lt;/option&gt;
</div>&lt;/php&gt;
</div>
</div>
</div>
</div>
</div>
</div>
</div>&lt;option value="&lt;php echo &lt;php echo $name?&gt;"&gt;&lt;php echo $name?&gt;&lt;/option&gt;
</div>&lt;/php&gt;
</div>
</div>
</div>
</div>
</div>
</div>
</div>&lt;option value="&lt;php echo &lt;php echo $name?&gt;"&gt;&lt;php echo $name?&gt;&lt;/option&gt;
</div>&lt;/php&gt;
</div>
</div>
</div>
</div>
</div>
</div>
</div>&lt;option value="&lt;php echo &lt;php echo $name?&gt;"&gt;&lt;php echo $name?&gt;&lt;/option&gt;
</div>&lt;/php&gt;
</div>
</div>
</div>
</div>
</div>
</div>
</div>
Listing 49: photos/edit.php

```php
<?php
if (isset($_POST['photo']) && isset($_POST['delete'])) {
    $photoId = $_POST['photo'];
    $deletePhoto = $_FILES['photo']['name'];
    if (isset($deletePhoto)) {
        // Code for deleting the photo
    } else {
        // Code for deleting the photo
    }
}

// Code for handling the form submission
```

111
Listing 50: questions/main.php

```php
<h1 class="page-header">Memory Test Questions</h1>
<form method="POST" action="/questions/main.php">
    <h4>No questions found</h4>
</form>
```

Listing 51: questions/edit.php

```php
<h1 class="page-header">Memory Test Questions</h1>
<form method="POST" action="/questions/editQuestion.php">
    <h4>No questions found</h4>
</form>
```
Listing 52: reminders/main.php

```php
echo $question->choiceD ? '': '</td>
</tr>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

```
remember appointments and/or activities to be done in coming days/weeks/months. Upcoming events are always displayed in the home page to remind you better. You can edit activity reminders to add or edit information about the schedule by clicking on the pen button below the one you wish to edit. You can also delete activity reminder by clicking on the trash button below the one you wish to delete.

Please be reminded that this is a collection of entries that were created/submitted either by the patient, his guardian, his therapist, and the general public that were approved by the guardian

Listing 53: reminders/edit.php

```php
<?php

$reminders = $this->AAS_model->getReminder(array 'user', 'theEntryCookie');
if($this->AAS_model->checkCookie('theSearchCookie')){
    echo '<a href="#search" class="form-inline glyphicon glyphicon-search"></a>

    <div class="panel panel-default">
        <div class="panel-body">
            <form method="POST" action="?php
                <input type="hidden" name="action" value="" />
                <input type="hidden" name="id" value="" />
                <button class="btn btn-default" type="submit" onclick="return confirm('Are you sure you want to delete this reminder?');">Submit</button>
            </form>
        </div>
    </div>
}
```

```php
<?php

// Filter: <php echo $cookie; ?>
</h4>
<div class="panel panel-default">
    <div class="panel-body">
        <php
            foreach($i = 0; $i < count($reminders); $i++) {
                echo '<h3>' . $reminders[$i]->title . '</h3>' .
                    '<p>' . $reminders[$i]->author . '</p>' .
                    '<p>' . $reminders[$i]->date_created . '</p>' .
                    '<p>' . $reminders[$i]->date_time . '</p>' .
                    '<p>' . $reminders[$i]->details . '</p>' .
            }
        </php>
        <h5><span class="glyphicon glyphicon-tag" aria-hidden="true"></span></h5>
            <php if($reminders[$i]->category == "all") echo $reminders[$i]->category . "" />
            <php if($this->AAS_model->setCookie(array 'user', 'theEntryCookie'))
                <p class="alert alert-info">
                    You have successfully edited this entry.
                </p>
            </php>
        </form>
        <form method="POST" action="?php
            <input type="hidden" name="action" value="" />
            <input type="hidden" name="id" value="" />
            <button class="btn btn-default" type="submit" onclick="return confirm('Are you sure you want to delete this reminder?');">Submit</button>
        </form>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<114
Listing 54: report/main.php

```php
<h1 class="page-header">Reports <small>Patient Progress Report</small></h1>
<div class="row">
  <div class="col-md-2"></div>
  <div class="col-md-8">
    <form method="POST" action="/user/generateReport" id="form-range"><br>
      <input type="date" name="start" id="start" class="form-control" value=""><br>
      <input type="date" name="end" id="end" class="form-control" value=""><br>
      <button type="submit" class="btn btn-primary">Generate Report</button>
    </form>
  </div>
</div>
<script type="text/javascript" src="/static/js/chart.js"></script>
</div>
```

Listing 55: report/graph.php

```php
// var chart = new google.visualization.LineChart(document.getElementById('curve_chart1'));
var chart_div = document.getElementById('curve_chart1');
var chart_div.innerHTML = '<img src="" + chart.getImageURI() + '">';
console.log(chart_div.innerHTML);
</script>
```

```php
var form = new google.visualization.LineChart(document.getElementById('curve_chart1'));
```

```php
title: 'Performance on Photo Recognition Tests',
legend: { position: 'bottom' }
```

```php
title: 'Performance on Photo Recognition Tests',
```

```php
class="form-control" value=""></div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
```
Listing 57: search/journal.php

```php
 grep foreach ($entries as $entry) {
   if ($entry->type == 2) { // echo "<span class="glyphicon glyphicon-pencil" aria-hidden="true">"; //echo "<a href="$entry->url" (user/ search/1)"?>
   foreach ($photos as $photo) {
     if ($entry->id == $photo->id) {
       echo "<img src="$photo->url" alt="" /"" title="">
     }
   }
   echo "Continue reading...

Listing 58: search/photo.php

```php
 foreach ($entries as $photo) {
   if ($photo->type == 2) { // echo "<a href="$photo->url" (user/ search/1)"?>
     foreach ($photos as $photo) {
       if ($photo->id == $entry->id) {
         echo "<img src="$photo->url" alt="" /"" title="">
       }
     }
   }
   echo "Continue reading...
```
Listing 59: search/question.php

```php
<?php
foreach ($entries as $question) {
    if ($question[filename] == 0) {
        $url = "assets/photos/" . $question[filename];
        $echo createElement('center') . $question[title] . " - \\
    questionております。
    echo "created." . $question[created];
}
    ?><br/>
</div>
if (count($entries) == 0) echo "center">"ch4No_questions_found." . $entries.
<?php
Listing 61: search/view.php

```PHP
```null
```
Listing 62: patient/dashboard.php

```
<h1 class="page-header">Dashboard <small>Overview</small></h1>
<div class="col-md-8">
<p>Welcome, &lt;php echo $cookie-&gt;first_name . " " . $cookie-&gt;last_name; ?&gt;&lt;/p&gt;

AAS is created for the benefit of helping people like you to cope with your condition. Aside from being the direct beneficiary of this system, you as a patient also plays an important role in this system to further make your experience here more efficient. Your nearest appointments and schedules are listed on the right side to remind you better. You can click the view all button to see the others. To better guide you in navigating the system, instructions were placed around the system. Try to explore and navigate now!

</div>
</div>
```

Listing 63: patient/connections.php

```
<h1 class="page-header">Connections <small></small></h1>
<div class="col-md-8">
<p>Guardian and Therapist <small></small></p>
</div>
</div>
```

Listing 64: generalKnowledge.php

```
<div class="page-header">Memory Test <small>General Knowledge</small></div>

Unlike the other two kinds of memory tests, this kind tests your awareness of general knowledge and/or facts. The questions aren't personal, they think of this as something that many people can answer. Example of the questions is that you may encounter are "Who is the current president of our country?" or "What is our national flower?" or "What do you call a female deer?" Despite of this test not being on your personal level, you can gain the benefit of helping you system. Try to see the others. To better guide you in answering the questions, just answer what you think and feel is right. All 10 items are of multiple choice type of test. Choose the best answer. Your scores in the tests will be the basis of the progress report that will be generated in the reports section. Click "Take Test" to start your memory test now.

```html
&lt;br&gt;&lt;br&gt;&lt;b&gt;Note:&lt;/b&gt; You are allowed to take only one test per category per day. If the "Take Test" button is disabled, that means that you already took the test earlier today. You can always come back tomorrow to take another one.

&lt;br&gt;
```html

Listing 65: generalKnowledge-Proper.php

```
<div class="page-header">Memory Test <small>General Knowledge</small></div>

```html
```
As you know, memory tests are very helpful when they touch you on the personal level. Personal test is a test of how much you know about the events and experiences that you had in your life so far. Questions have a wide range of variation. It can go from naming a person being described in the question to a date when an event happened. Remember that these questions are submitted and/or created by the people who really know you. You don’t have to feel pressured.
Listing 68: personalTestProper.php

```php
<hl class="page-header">Memory Test <small>
Personal Test</small></hl>
<div class="pad">
  <div class="row">
    <!-- Content goes here -->
  </div>
</div>
```

Listing 69: personalTestAnswers.php

```php
<hl class="page-header">Memory Test <small>
Personal Test</small></hl>
<div class="pad">
  <div class="row">
    <!-- Content goes here -->
  </div>
</div>
```
photos are important to visually represent your memories for you. Photo recognition is a fun memory test where you have to answer questions based on the given photo. This test is also on your personal level meaning that the questions are based on the events and experiences that you had in your life. Questions can come from naming the person in the photo to remembering the date when the photo was taken. Remember that all of these questions are created and/or submitted by the people who really know you. You don’t have the feel pressured in answering the questions just answer what you think and feel to the right. All 10 items are of multiple choice type of test. Choose the best answer. Your answers in the tests will be the basis of the progress report that will be generated in the reports section. Click on "Take Test" to start your memory test now.

<h3>Take Test</h3>
<button id='save_answer'>Submit</button>

Listing 71: photoRecognitionProper.php

Listing 72: photoRecognitionAnswers.php
AAMS is created for the benefit of helping patients with anterograde amnesia cope with their condition. You, as a guardian, plays an important role in this system to further make the experience of the patient here more efficient. The nearest appointments and schedules involving the patient are listed on the right side to remind you better. You can click the view all button to see the others. To better guide you in navigating the system, instructions were placed around the system. Try to explore and navigate now!

Listing 73: guardian/dashboard.php

Listing 74: guardian/connections.php
Listing 75: guardian/categories.php

```php
<h1 class="page-header">Categories <small>Manage entry categories</small></h1>
```

```php
div
class="pad"
div
class="row"
td
class="col-md-3"
div
class="panel panel-default panel-body"
```

```php
td
td
td
text
```

```php
td
td
td
text
```

```php
td
```

```php
td
td
td
text
```

Listing 76: guardian/manageUsers.php

```php
<h1 class="page-header">Manage Users <small>View, delete users</small></h1>
```

```php
div
class="line"
```

```php
foreach ($categories as $cat) {

```

```php
return confirm ('Are you sure you want to remove this category?');
```

```php
td
```

```php
td
text
```

```php
td
text
```

```php
td
text
```

Listing 77: guardian/userRequests.php

```php
<h1 class="page-header">Account Requests <small>Accept/Reject users</small></h1>
```

```php
div
class="row"
div
class="col-md-4"
div
class="panel panel-default panel-body"
```

```php
foreach ($users as $user) {
```

```php
return confirm ('Are you sure you want to remove this user?');
```

```php
td
text
```

```php
text
```

```php
return confirm ('Are you sure you want to accept this user?');
```

```php
td
text
```

```php
td
text
```

Listing 78: guardian/submissions.php
These entries are submitted either by the patient or anyone from the public who knows the patient's public ID. Please be careful on accepting and rejecting entries. Make sure to accept only those legitimate and relevant entries. Otherwise, reject it to avoid bombarding the patient with too many entries.
Listing 80: guardian/photoQuestion-Preview.php

```php
<?php
    // ... (code snippet)
?>
```

Listing 81: ot/dashboard.php

```php
<?php
    // ... (code snippet)
?>
```
Listing 82: ot/editAccount.php

```php
<?php
echo $r['reminder']->date_time;
?>

<?php
$cookie = $r['reminder']->title;

<?php
if($r['patient']->first_name === ''){
    $details = substr($r['reminder']->details, 0, 255);
    echo $details . '...';
}

<?php
++$i;
if($i==3)
break;
}
if(count($reminders)==0){
    echo '<h5>No reminder reminders. /center></h5>;
}
else{
    ?></div>
    </div>
    </div>
    </div>
</div>
</div>
</div>
</div>

Listing 83: ot/patients.php

```
Listing 85: ot/create/journal.php

```php
<form method="POST" action="<?php echo base_url('ot/submit/1'); ?>">
  <div class="form-group">
    <label for="patient">Patient</label>
    <input type="text" name="patient" id="patient" class="form-control">
  </div>
  <div class="form-group">
    <label for="title">Title</label>
    <input type="text" name="title" id="title" class="form-control">
  </div>
  <div class="form-group">
    <label for="details">Details</label>
    <textarea name="details" id="details" class="form-control" rows="4"></textarea>
  </div>
  <div class="form-group">
    <button type="submit" class="btn btn-primary">Submit</button>
  </div>
</form>
```

Listing 86: ot/create/photo.php

```php
<form method="POST" action="<?php echo base_url('ot/submit/3'); ?>" enctype="multipart/form-data">
  <div class="form-group">
    <label for="patient">Patient</label>
    <input type="text" name="patient" id="patient" class="form-control">
  </div>
  <div class="form-group">
    <label for="title">Title</label>
    <input type="text" name="title" id="title" class="form-control">
  </div>
  <div class="form-group">
    <label for="details">Details</label>
    <textarea name="details" id="details" class="form-control" rows="4"></textarea>
  </div>
  <div class="form-group">
    <input type="file" name="picture" id="picture" class="form-control" accept="image/*" required="true">
  </div>
  <div class="form-group">
    <button type="submit" class="btn btn-primary">Submit</button>
  </div>
</form>
```

Listing 87: ot/create/question.php

```php
<form method="POST" action="<?php echo base_url('ot/submit/4'); ?>" enctype="multipart/form-data">
  <div class="form-group">
    <label for="patient">Patient</label>
    <input type="text" name="patient" id="patient" class="form-control">
  </div>
  <div class="form-group">
    <label for="title">Title</label>
    <input type="text" name="title" id="title" class="form-control">
  </div>
  <div class="form-group">
    <label for="details">Details</label>
    <textarea name="details" id="details" class="form-control" rows="4"></textarea>
  </div>
  <div class="form-group">
    <button type="submit" class="btn btn-primary">Submit</button>
  </div>
</form>
```

Listing 88: ot/create/reminder.php

```php
<form method="POST" action="<?php echo base_url('ot/submit/2'); ?>">
  <div class="form-group">
    <label for="patient">Patient</label>
    <input type="text" name="patient" id="patient" class="form-control">
  </div>
  <div class="form-group">
    <label for="title">Title</label>
    <input type="text" name="title" id="title" class="form-control">
  </div>
  <div class="form-group">
    <label for="details">Details</label>
    <textarea name="details" id="details" class="form-control" rows="4"></textarea>
  </div>
  <div class="form-group">
    <button type="submit" class="btn btn-primary">Submit</button>
  </div>
</form>
```

Listing 89: ot/report/main.php

```php
<?php include_once('header.php'); ?><br>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
```
Listing 90: ot/report/graph.php

```php
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>

<script type="text/javascript">
var data = new google.visualization.DataTable();
data.addColumn('date', 'Date');
data.addColumn('number', 'Test_Scores');
</script>

foreach($personal as $p){
  echo "data.addRows([new Date("\".{$p}\"\".date\"\".\"\".{$p}\"\".score\"\".\"\"),'\"\")];";
}

var options = {
  title: 'Performance on General Knowledge Tests',
curveType: 'function',
legend: { position: 'bottom' }
};

// var chart = new google.visualization.LineChart(document.getElementById('curve_chart1'));
var chart_div = document.getElementById('curve_chart1');
chart_div.style = 'width: 800px';
chart_div.innerHTML = '-';
var data = new google.visualization.DataTable();data.addColumn('date', 'Date');
data.addColumn('number', 'Test_Scores');</script>
</div>
</div>
</div>
</div>
</form>

Listing 91: ot/reminders/main.php

```php
var options = {
  title: 'Performance on Photo Recognition Tests',
curveType: 'function',
legend: { position: 'bottom' }
};
</script>

var chart = new google.visualization.LineChart(document.getElementById('curve_chart2'));
var chart_div = document.getElementById('curve_chart2');
chart_div.innerHTML = '-';
var data = new google.visualization.DataTable();
data.addColumn('date', 'Date');
data.addColumn('number', 'Test_Scores');
</script>
</div>
</div>
</div>
</div>
</form>

<h1 class="page-header">Activity Reminders</h1>
Listing 92: ot/reminders/edit.php

```php
<div class="pad">Activity Reminders <small>Patient Appointments</small></div>

<?php
if ($reminder->error == 1) {
    $reminder->error = 0;
    $this->AIS_model->setCookie (array
        'reminder', 'theEntryCookie');
}

<div class="alert alert-danger role="alert">
    <span class="glyphicons glyphicons-exclamation-warning sign" aria-hidden="true"></span>
    Error: Please check your input. All fields are required.
</div>

else if ($reminder->error == 2) {
    $reminder->error = 0;
    $this->AIS_model->setCookie (array
        'reminder', 'theEntryCookie');
}

    <div class="alert alert-success role="alert">
        <span class="glyphicons glyphicons-ok-circle sign" aria-hidden="true"></span>
        You have successfully edited this entry.
    </div>

else if ($reminder->error == 3) {
    $reminder->error = 0;
}
```

Listing 93: publicUser/dashboard.php

```php
<div class="pad">Dashboard <small>Submissions Overview</small></div>

```
Listing 94: publicUser/editAccount.php

```php
<?php
if ($cookie->error==1){
    $this->AAAIS_model->setCookie ( array $cookie , 'theUserCookie' );
    ?><div class="alert alert-danger role="alert">
        &lt;span class="glyphicon glyphicon-exclamation-circle" aria-hidden="true"></span>&lt;/div>
        Error: Please check your input. All fields are required
    </div>
} else if ($cookie->error==2){
    $cookie->error = 0;
    $this->AAAIS_model->setCookie ( array $cookie , 'theUserCookie' );
    ?><div class="alert alert-success role="alert">
        &lt;span class="glyphicon glyphicon-ok-circle" aria-hidden="true"></span>&lt;/div>
        You have successfully edited your account.
    </div>
} else {
    form method="POST" action="&lt;?php echo $base_url('publicUser/editAccount/stepTwo') ?&gt;" enctype="multipart/form-data">
        &lt;div class="col-md-3"&gt;
            &lt;h3 title="A C C O U N T k n b p : I N F O R M A T I O N"&gt;
                &lt;/h3&gt;
                &lt;div class="col-md-3"&gt;
                    &lt;div class="col-md-6"&gt;
                        &lt;table class="table"&gt;
                            &lt;tr&gt;
                                &lt;th&gt;Username&lt;/th&gt;
                                &lt;td&gt;&lt;input type="text" name="username" id="&lt;?php echo $cookie->username ?&gt;" type="text"&gt;&lt;/td&gt;
                            &lt;/tr&gt;
                            &lt;tr&gt;
                                &lt;th&gt;Password&lt;/th&gt;
                                &lt;td&gt;&lt;input type="password" name="password" id="&lt;?php echo $cookie->password ?&gt;" type="password"&gt;&lt;/td&gt;
                            &lt;/tr&gt;
                            &lt;tr&gt;
                                &lt;th&gt;First Name&lt;/th&gt;
                                &lt;td&gt;&lt;input type="text" name="firstName" id="&lt;?php echo $cookie->firstName ?&gt;" type="text"&gt;&lt;/td&gt;
                            &lt;/tr&gt;
                            &lt;tr&gt;
                                &lt;th&gt;Last Name&lt;/th&gt;
                                &lt;td&gt;&lt;input type="text" name="lastName" id="&lt;?php echo $cookie->lastName ?&gt;" type="text"&gt;&lt;/td&gt;
                            &lt;/tr&gt;
                            &lt;tr&gt;
                                &lt;th&gt;Email Address&lt;/th&gt;
                                &lt;td&gt;&lt;input type="email" name="email" id="&lt;?php echo $cookie->email ?&gt;" type="email"&gt;&lt;/td&gt;
                            &lt;/tr&gt;
                            &lt;tr&gt;
                                &lt;th&gt;Sex&lt;/th&gt;
                                &lt;td&gt;&lt;select name="sex" id="&lt;?php echo $cookie->sex ?&gt;" type="text"&gt;&lt;/td&gt;
                            &lt;/tr&gt;
                            &lt;tr&gt;
                                &lt;th&gt;Birthday&lt;/th&gt;
                                &lt;td&gt;&lt;input type="date" name="birthday" id="&lt;?php echo $cookie->birthday ?&gt;" type="date"&gt;&lt;/td&gt;
                            &lt;/tr&gt;
                            &lt;tr&gt;
                                &lt;th&gt;Telephone&lt;/th&gt;
                                &lt;td&gt;&lt;input type="text" name="telephone" id="&lt;?php echo $cookie->telephone ?&gt;" type="text"&gt;&lt;/td&gt;
                            &lt;/tr&gt;
                            &lt;tr&gt;
                                &lt;th&gt;Sex&lt;/th&gt;
                                &lt;td&gt;&lt;select name="sex" id="&lt;?php echo $cookie->sex ?&gt;" type="text"&gt;&lt;/td&gt;
                            &lt;/tr&gt;
                            &lt;tr&gt;
                                &lt;th&gt;Sea&lt;/th&gt;
                                &lt;td&gt;&lt;select name="sea" id="&lt;?php echo $cookie->sea ?&gt;" type="text"&gt;&lt;/td&gt;
                            &lt;/tr&gt;
                            &lt;tr&gt;
                                &lt;th&gt;Email Address&lt;/th&gt;
                                &lt;td&gt;&lt;input type="email" name="email" id="&lt;?php echo $cookie->email ?&gt;" type="email"&gt;&lt;/td&gt;
                            &lt;/tr&gt;
                            &lt;tr&gt;
                                &lt;th&gt;Sex&lt;/th&gt;
                                &lt;td&gt;&lt;select name="sex" id="&lt;?php echo $cookie->sex ?&gt;" type="text"&gt;&lt;/td&gt;
                            &lt;/tr&gt;
                            &lt;tr&gt;
                                &lt;th&gt;Telephone&lt;/th&gt;
                                &lt;td&gt;&lt;input type="text" name="telephone" id="&lt;?php echo $cookie->telephone ?&gt;" type="text"&gt;&lt;/td&gt;
                            &lt;/tr&gt;
                        &lt;/table&gt;
                    &lt;/div&gt;
                    &lt;div class="col-md-8"&gt;
                        &lt;button class="btn btn-primary" type="submit"&gt;Submit&lt;/button&gt;
                    &lt;/div&gt;
                &lt;/div&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/form&gt;
    &lt;hr&gt;
    &lt;div class="text-center"&gt;&lt;h3&gt;&lt;i&gt;&lt;/i&gt;&lt;/h3&gt;&lt;/div&gt;
}}&gt;
```
Listing 98: publicUser/reminder.php

```php
foreach ($entries as $reminder) {
  if ($reminder->status == 1) {
    echo "<center>Delayed
    " . $reminder->title . "</center>"
  } else { /* other cases */
    echo "<center>Approved</center>"
  }
}
```

Listing 99: publicUser/create/main.php

```php
switch($entry->type) {
  case 1:
    // create Journal Entries
    break;
  case 2:
    // create Activity Reminders
    break;
  case 3:
    // create Photos
    break;
  case 4:
    // create Memory Test Questions
    break;
}
```

Listing 100: publicUser/question.php

```php
foreach ($entries as $question) {
  if ($question->status == 1) {
    echo "<center>Delayed
    " . $question->title . "</center>"
  } else { /* other cases */
    echo "<center>Approved</center>"
  }
}
```
Listing 101: publicUser/create/photo.php

```php
<?php
if ($entry->error==1){
    echo '<div class="alert alert-danger role="alert">
        <span class="glyphicon glyphicon-exclamation-sign" aria-hidden="true"></span>
        Please check your input. All fields are required
    </div>
    <br/>
    <br/>
}</php>

Listing 102: publicUser/create/question.php

```
Listing 103: publicUser/create/reminder.php

```php
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

$route['default_controller'] = 'Public_Controller'; /* MODIFIED */
$route['404_override'] = '404_not_found';
$route['translate_uri_dashes'] = FALSE;

/* General Routes */
$route['login'] = 'Public_Controller/login';
$route['logout'] = 'Public_Controller/logout';

/* Public Routes */
$route['public'] = 'Public_Controller/publicPanel';
$route['public/submitAccount'] = 'Public_Controller/submitAccount';
$route['forgotPassword'] = 'Public_Controller/forgotPassword';
$route['loginPage'] = 'Public_Controller/loginPage';
$route['request'] = 'Public_Controller/request';

/* System Admin Routes */
$route['sysad/dashboard'] = 'SystemAdmin_Controller/dashboard';
$route['sysad/editAccount'] = 'SystemAdmin_Controller/editAccount';
$route['sysad/editAccount/stepTwo'] = 'SystemAdmin_Controller/editAccountStepTwo';
$route['sysad/addUser'] = 'SystemAdmin_Controller/addUser';
$route['sysad/addUserPanel'] = 'SystemAdmin_Controller/addUserPanel';
$route['sysad/addUser/stepTwo'] = 'SystemAdmin_Controller/addUserStepTwo';
$route['sysad/addUser/stepThree'] = 'SystemAdmin_Controller/addUserStepThree';
$route['sysad/addUser/stepFive'] = 'SystemAdmin_Controller/addUserStepFive';

/* SysAd Routes */
$route['sysad/sysad'] = 'SystemAdmin_Controller/sysad';
$route['sysad/sysad/stepTwo'] = 'SystemAdmin_Controller/sysadStepTwo';
$route['sysad/sysad/stepThree'] = 'SystemAdmin_Controller/sysadStepThree';

$route['sysad/sysad/new'] = 'SystemAdmin_Controller/sysad/new';
$route['sysad/sysad/new/stepTwo'] = 'SystemAdmin_Controller/sysadNewStepTwo';
$route['sysad/sysad/new/search'] = 'SystemAdmin_Controller/sysadNewSearch';

/* User Routes */
$route['user/editAccount'] = 'User_Controller/editAccount';
```

Listing 104: routes.php

```php
<form method="POST" action="<?php echo base_url('publicUser/submit/2'); ?>">
<div align="center">
<label for="answer">Correct Answer</label>
<br>
<select name="category" id="category" class="form-control">
<option value="A">A</option>
<option value="B">B</option>
<option value="D">D</option>
<option value="F">F</option>
</select>
<br>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
```
Controller / dashboard
Controller / submissions
Controller / photoQuestions
Controller /
Controller / rejectUser
Controller / userRequests
Controller / editReminderStepTwo
Controller / accept
Controller / editAccountStepTwo
Guardian
Guardian / userRequests
Guardian / guardUserRequest
Guardian / createUser
Guardian / rejectUser
Guardian / submissions
Guardian / controller / accept
Guardian / controller / reject

$route['user/editAccount/stepTwo'] = 'User_Controller/editAccountStepTwo';
$route['user/submit/2'] = 'User_Controller/submit/2';
$route['user/search/3'] = 'User_Controller/search/3';
$route['user/report'] = 'User_Controller/report/1';
$route['user/generateReport'] = 'User_Controller/generateReport';

Guardian
Guardian / controller / reject

$route['guardian/dashboard'] = 'Guardian_Controller/dashboard';
$route['guardian/userRequests'] = 'Guardian_Controller/userRequests';
$route['guardian/acceptUser'] = 'Guardian_Controller/acceptUser';
$route['guardian/rejectUser'] = 'Guardian_Controller/rejectUser';
$route['guardian/submissions'] = 'Guardian_Controller/submissions';
$route['guardian/accept'] = 'Guardian_Controller/accept';
$route['guardian/reject'] = 'Guardian_Controller/reject';

Patient
Patient / controller / dashboard
Patient / patient / dashboard
Patient / patient / connections
Patient / controller / connections
Patient / patient / personalTest
Patient / controller / personalTest
Patient / controller / photoRecognition
Patient / controller / photoRecognitionProper
Patient / controller / submitPhotoRecognition
Patient / controller / submitPhotoRecognitionProper
Patient / patient / generalKnowledge
Patient / controller / generalKnowledge
Patient / controller / generalKnowledgeProper
Patient / controller / submitGeneralKnowledge
Patient / controller / submitGeneralKnowledgeProper

$route['patient/dashboard'] = 'Patient_Controller/dashboard';
$route['patient/connections'] = 'Patient_Controller/connections';
$route['patient/personalTest'] = 'Patient_Controller/personalTest';
$route['patient/photoRecognition'] = 'Patient_Controller/photoRecognition';
$route['patient/photoRecognitionProper'] = 'Patient_Controller/photoRecognitionProper';
$route['patient/generalKnowledge'] = 'Patient_Controller/generalKnowledge';
$route['patient/generalKnowledgeProper'] = 'Patient_Controller/generalKnowledgeProper';
$route['patient/submitGeneralKnowledge'] = 'Patient_Controller/submitGeneralKnowledge';
$route['patient/submitGeneralKnowledgeProper'] = 'Patient_Controller/submitGeneralKnowledgeProper';

Public User
Public User / dashboard

$route['publicUser/dashboard'] = 'PublicUserController/dashboard';
```
Listing 105: app.css

body {
  margin-top: 50px;
  font-size: 12pt;
}

footer {
  margin-left: 0;
  padding: 1em;
  position: fixed;
  bottom: 0;
  width: 100% !important;
  background-color: #333;
  color: white;
}
footer * {
  margin: 0;
}

.panel-info2{
  background-color: #eee;
  opacity: 0.8;
}

.panel-body{
  padding: 20px;
}

.title{
  color: #337ab7;
}

.pad{
  margin-left: 20px;
  margin-right: 20px;
}

.nav-tabs > li > a {
  color: #337ab7;
  background-color: #eee;
}

.clear{
  background-color: Transparent;
  background-repeat: no-repeat;
  border: none;
  cursor: pointer;
  overflow: hidden;
  outline: none;
  color: #000000;
}

.clear2{
  background-color: Transparent;
  background-repeat: no-repeat;
  border: none;
  cursor: pointer;
  overflow: hidden;
  outline: none;
  color: #000000;
}

.table.add td + td {
  border-left: 2px solid black;
  padding-left: 10px;
  padding-bottom: 10px;
}

hr{
  width: auto;
  height: 8px;
  margin-left: auto;
  margin-right: auto;
  background-color: #5bc0de;
  border: 0 none;
  margin-top: 5px;
  margin-bottom: 5px;
}

.pic{
  max-width: 60%;
  max-height: auto;
}

.dp{
  height: 120px;
  width: 120px;
}

.choosePhoto{
  height: auto;
  width: 200px;
  display: inline-block;
  margin-right: 5px;
  margin-bottom: 10px;
}

div.choosePhoto > img{
  display: block;
  width: auto;
  height: auto;
  max-width: 100%;
  max-height: 100%;
}

.bottom{
  margin-bottom: 0px;
}

.blue{
  width: 100%;
  background-color: #27ab77;
  border-radius: 10px;
  padding-left: 15px;
  padding-right: 15px;
  color: white;
  border: solid 1px #27ab77;
}

.gray{
  color: #999;
}

.smallPic{
  max-height: 100%;
  max-width: 100%;
}

h4.line6{
  width:100%; border-bottom: 1px solid #000; line-height:0.1em; margin:10px 0 50px;
}

h4.line span ( background:#fff; padding:0 10px ; )

.blah > .row > .col-md-3nth-child(4n+1) {
  clear: both;
}

.panel:hover:hover{
  border: solid 1px #27ab77;
}

a.active{
  text-decoration: underline;
  font-weight: bold;
}
```
XI. Acknowledgement

“AYOKO NA.”

How many times have I uttered those words in the last four years? Moreover, how many times have I uttered those words in the last 7 months (proposal to defense period)? I lost count of the times when I was on the verge of giving up and just throw it all away. There were times when I just wanted to walk away from everything and forget the world but well, I never did. I am too blessed with a lot of supportive people around me. So for the next paragraphs, allow me to thank the people who made this possible and also, let me use my native tongue in expressing myself para mas feel!

To my family. Thank you so so so much sa lahat lahat ng ginawa niyo for me. Thank you sa pag-intindi sakin lalo na sa mga times na ang cranky ko na and di niyo na ko makausap sa sobrang busy. Thank you kasi never niyo kong iniwan. And to my kuya, thank you sa lahat ng suporta mo especially sa funds. Nung bumigay na yung netbook ko, you didnt even have a second thought na ibili ako ng bagong laptop for school. Hayaan mo kuya, babawi ako sayo. Cheers to our dreams! I love you guys.

To my adviser. You have no idea how thankful I am na ikaw yung adviser ko (both a SP and sa life). I cant find the words that will be enough to express how much I appreciate everything that you did for me. No words, grabe. There were times when I myself stopped believing that I can do it but you never got tired of pushing me to continue. You’re the other half of this SP, I couldn’t have finished this without your guidance kahit lagi mo kong inaasar and binubully hahaha. Maraming salamat.

To my professors. Tough love wow!! Hahaha. Now I understand why you had to be very strict to us sometimes to the point that I am cursing you deep inside already. I know you were just trying to teach us lessons, both academically and in life, to make us better. So to every one of my professor from freshman year, from the bottom of my heart, maraming salamat po.
To block 12. GUYS!!!!!!!!!!! We did it omg huluhlu. You know what? I am SO proud of our block. I saw how much effort we all gave in to this. I remember the times when it was 4am and almost everyone was still up doing acads. We never ceased to support each other, andyan yung mga random messages from each other to motivate and inspire. I love you guys so much. Iba kayo, iba.

To all my friends. Thank you for always being there for me especially to TBL. Thank you for always being my one call away friends na puputahan and ilalabas ako pag sobrang kailangan kong mag-relax and mag-escape muna from reality kahit for a few hours lang. You guys are the best! Bond soon!!!

To Miguel and kuya Luis. Yup, I told you I would save a space for you guys! You were my life savers. Thank you Miguel for referring kuya Luis to me and thank you kuya Luis for allowing me to interview you FOUR days before my defense. I hope I can make it up to you guys soon!

To my pretty, adorable, and still fresh self. Girl sobrang proud ako sayo!!! WE DID IT, WE FREAKING DID IT.

And finally, to You. None of these would be possible without You. Thank you so much Lord, the best ka talaga.

P.S. Shoutout din sa Kopiko 78, alam mo na.