University of the Philippines Manila
College of Arts and Sciences
Department of Physical Sciences and Mathematics

An Information Management System for Las Piñas Doctors Hospital

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

Submitted by:

Maria Fiona Q Morella
April 2014

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

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

ACCEPTANCE SHEET

The Special Problem entitled "An Information Management System for Las Piñas Doctors Hospital" prepared and submitted by Maria Fiona Q. Morella in partial fulfilment of the requirements for the degree of Bachelor of Science has been examined and is recommended for acceptance.

_____

Gregorio B. Baes, Ph.D. (*candidate*)
Adviser

EXAMINERS:

	Approved	**Disapproved**
1. Avegail D. Carpio, M.Sc.	_____	_____
2. Richard Bryann L. Chua, M.Sc.	_____	_____
3. Aldrich Colin K. Co, M.Sc (*candidate*)	_____	_____
4. Vincent Peter C. Magboo, M.D., M.Sc.	_____	_____
5. Ma. Sheila A. Magboo, M.Sc.	_____	_____
6. Geoffrey A. Solano, M.Sc.	_____	_____
7. Bernie B. Terrado, 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.
Unit Head
Mathematical and Computing Sciences Unit
Department of Physical Sciences and Mathematics

_____

Marcelina B. Lirazan, Ph.D.
Chair
Department of Physical Sciences and Mathematics

_____

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

Abstract

This study presents the information system designed for Las Piñas Doctors Hospital. The system was designed to provide medical, medical support, and administrative functions not previously available to the hospital. The system allows physicians to view the medical records of their patients, and using this information use the system to order medical tests and procedures for their patients. The results of said procedures are added to the system by a medical technician, at which point they may be viewed by the physicians and patients with whom the result is associated.
Keywords: Hospital information system, patient portal, electronic medical record

Contents

List of Tables

List of Figures

I. Introduction

A. Background of the Study

As computers began to become daily fixtures of everyday life about 30 years ago, it was not long before the health care industry came to realize its potential for revolutionizing the delivery of medical services. Much as it had in the banking, retail and transportation industries, the crunching power and speed of the information technology machines to process large blocks of data offered a powerful weapon for research to tame and conquer diseases, as well as bright prospects for cost savings by improving efficiency in health care systems.

Though the health care industry was slower than most in taking advantage of the wealth of possibilities offered by computing power, enormous technological advances since then in data storage, graphical interfaces, wireless, and interaction between humans and computers through the Internet and the World Wide Web eventually helped make computers a mainstay of the industry.

Computers are now widely used in medical imaging, laboratory procedures, and the transmission, delivery and sharing of medical data as well as health care across great distances. It improved decision-making by clinicians and allowed for faster access to results of medical procedures as well as literature, and reduced the possibility of medical errors. To use an analogy, computers have made possible a level of efficiency, accuracy and speed of delivery that could be compared to that of a well-designed and smoothly functioning roadway or rail service.

However, the infrastructure does not run smoothly or efficiently at all stages of health care delivery because the quality of the service remains uneven, partly due

to their failure to take advantage of the efficiencies offered by technology. Certain segments, specifically where they concern the patient and his or her doctor, are stuck in technology that are decades behind. The use of health information technology has shown to greatly improve the quality of health care.

Las Piñas Doctors Hospital (LPDH) was founded in 1982 and originally located in BF Almanza, Las Piñas City. The hospital has since expanded, and now has a 100-bed capacity and is located at Aguilar Avenue, Pulanglupa 2, Las Piñas City. The location of Las Piñas Doctors Hospital is a unique advantage, as its location very near the boundary of Las Piñas and Parañaque City means it it has a customer base in both cities.

There are many ways with which the current system of the hospital can be improved. The global nonprofit Healthcare Information and Management Systems Society (HIMSS), which promotes the use of information technology in healthcare, has an EMR (electronic medical record) Adoption Model which classifies the extent of a hospital's electronic medical record capabilities according to eight stages, from Stage 0 to Stage 7. [2]

Stage 0 hospital systems have none of the three key ancillary department systems (laboratory, pharmacy, and radiology) installed. Systems of the highest stage, Stage 7, have mostly (if not completely) paperless environments with inter-operability among systems and can seamlessly exchange information even with non-associated entities.

Using the HIMSS EMR Adoption model as a metric, it would be fair to unofficially classify LPDH as having a Stage 1 system, as it meets the requirements: all three major ancillary clinical systems are installed; but the hospital is lacking a centralized clinical data repository from which physicians may access orders and test results.

B. Statement of the Problem

Hospital information systems (HIS) are the application of computers and related technology in managing and administrating a hospital and in providing patient care. From the perspective of the health care provider, the investment in information technology can lead to greater profitability through the increase in efficiency and productivity.

The increase in productivity comes from methods like computerised order entry, which decreases the amount of time spent on administrative tasks and freeing up more time for clinical tasks. This increased productivity leads to greater clinical capacity for the hospital and better quality of patient care.

As previously touched upon in the background of the study, LPDH has a medical support subsystems for laboratory, pharmacy, and radiology, but does not have a clinical data repository that centralises information from these systems.

C. Objectives of the Study

The objective of the study is to develop a hospital information system for Las Pinñas Doctors Hospital. The system would comprise of a variety of medical, medical support, administrative, and security functions.

1. The medical and medical support functions of the system are as follows:
    (a) an electronic medical record (EMR) detailing the past history of patients
        i. This medical record may be viewed and searched by the physicians attending to the patient.
2. The information security concerns of the system are as follows:

(a) confidentiality - access to data must be limited only to those with the right to the information

(b) integrity - data must be verifiable and backed up by duplicate systems.

(c) availability - information must be available to users as soon as possible.

The users of the system are categorised by the role they play; they are only allowed to access information judged necessary. The types of users are as follows:

1. Doctors. Doctors associated with the hospital are automatically assigned accounts. The level of access granted to a doctor is such that they may:
   (a) order tests to be performed on their patients
   (b) view the medical records of patients to whom they are assigned
   (c) enter diagnosis for the test results
2. Patients
   (a) Outpatients who have laboratory or radiological tests performed at the hospital may view these results online using an account that is automatically generated for them should they choose to opt in to the service.
3. Clerks may create and manage records for patients
4. System administrators. The system administrators are in charge of the management of user account information. Their functions in the system include:
   (a) Creating new accounts, managing existing ones, deleting accounts if necessary.
   (b) Updating the medical procedures provided by each department
   (c) Adding or managing new hospital departments as necessary
   (d) Adding new tables for medical test result formats
5. Medical technicians or radiology technicians
   (a) View the list of medical tests whose results they are responsible for adding

to the database

(b) Accomplish the entry of patient test results into the database.

6. Hospital administrators. Since hospital administrators are overseers, they are given access to the records of patients to be able to respond when necessary in cases of complaints, delays, or other such concerns.

D. Significance of the Study

The aim of any health information system is to contribute to and improve patient care.

As previously touched upon in the introduction, the implementation of a hospital information system reduces cost for the hospital by reducing paperwork and eliminating redundant processes. It also reduces the amount of time spent on administrative tasks, freeing up more time for clinical duties, thus allowing for improvement in the quality of care and increased clinical capacity for the hospital.

Among the many ways information systems improve a hospital's productivity includes reducing the time lags with respect to patient care, such as in the time it takes for a patient's laboratory results to be delivered. The faster a patient's laboratory results are known, the faster the patient's physicians can decide on the proper course of action.

E. Scope and Limitations

1. The system acquires its data from the medical tests performed at the hospital, and so the accuracy of the results depends on the accuracy of those machines

and the medical personnel who use them.

2. It will not work retroactively. Results will be available to users when the system is put into place, and they may view their results at a later date, but results from before the creation of the system will not be available .

3. The doctors who are associated with the hospital are automatically assigned accounts on the web portal. However, it is assumed that physicians from other hospitals who wish to view results on the portal will have to seek permission from the hospital, or rely on the patient to give them the results that they have obtained. This system is limited to one hospital, and will not be designed for sharing between institutions.

F. Assumptions

1. All physicians affiliated with the hospital are provided when an account when the system is put into place, but it is assumed that when a doctor wishes to sign up for the account after the initial implementation period then the physician must contact a system administrator to provide them with an account.

2. It is assumed that any user who wishes to use the system and get an account has a valid, working email address, since it is a necessary part of the sign-up process.

3. The data entered into the system is assumed to be accurate and confirmed by the necessary authorities. Once a test result is verified and uploaded into the system, it is assumed that the hospital is accountable for the accuracy of the data available on the system. The hospital will be held accountable if any

inaccuracy in the results lead to misdiagnosis of a patient. The legal issues pertaining to any inaccuracy are not within the scope of this study.

4. It is assumed that all the participating doctors and medical technicians abide by the hospital's regulations and the laws concerning correctness of information and confidentiality of patient records.

5. It is assumed that the physicians attend to patients on a first-come first-serve basis.

II. Review of Related Literature

Telemedicine is the use of telecommunication—such as telephones, computer networks, or the internet—to transmit medical information from one place to another. In the early 20th century, when the most cutting-edge technology available to humankind was the telephone, electrocardiograph data was transmitted over telephone wires[3], but most advances in telemedicine have happened over the past thirty years, spurred on by the advances in telecommunications. The replacement of analog methods of communication with digital ones, as well as the dropping prices of the technology used in it has led to a more widespread adoption of the practice worldwide. The versatility of modern-day communication systems means that all manner of medical data can be transmitted electronically—ranging from simple numerical records such as blood test results to the more complicated medical images.

According to the World Health Organization[1], telemedicine can be further classified into two types, according to the timing of the interaction[4]. The first is real-time or synchronous telemedicine, which allows for instantaneous interaction. One such example is videoconferencing—simultaneous two-way video and audio communication. The other type is asynchronous, or store-and-forward telemedicine. Data from one end is recorded and then sent to another user at a different time.

One of the most popular branches of telemedicine is teleradiology [4], or the transmission of radiological images for the purpose of research or diagnosis. In a history of teleradiology published in the journal Radiology, it was noted that commercial teleradiology systems first became available in the 1980s, but the high costs and poor performances of such systems meant that they were not widely adopted. It wasn't until the late 1990s that technology had progressed to a point where teleradiology

[1] http://www.who.int

was viable. [5] Communication systems such as the Internet reduced the costs of such systems, and at the same time, medical imaging had also begun to shift from film to digital capture. When adopted, teleradiology can massively change the strategies used by radiologists. The use of teleradiology eliminates the need for radiologists to travel to the hospital where their services are needed.

The market for teleradiology is large because of two reasons. Firstly, the demand for radiology services far outstrips the number of trained radiologists who exist to service that need. A news article in NBC News published in 2004 noted that a shortage of trained radiologists in the United States had led many American hospitals to solve this problem by outsourcing the work to countries like Australia, India, Israel, and Lebanon.[6] Secondly, it is often the case in developing countries that hospitals in rural areas do not have sufficient radiological personnel. Telemedicine allows radiologists based elsewhere to provide services to remote locations.[4]

A necessary practice in teleradiology is the digitisation of x-ray films. Specialised film digitisers are sometimes used, but this option is considered by some to be too costly to implement. As a result, cheaper alternatives such as flatbed scanners and digital cameras have been used in place of more expensive equipment. A study conducted in 2012 and published in Telemedicine Journal and e-Health found no significant statistical difference in diagnostic accuracy when using a film digitiser, flatbed scanner, and digital camera. [7][8]

The concept of telemedicine has, in more recent times, given way to the concept of eHealth. eHealth is an umbrella term that was coined in 1999 to describe the combined use of information and communication technology in health care. According to Evan Rosen, writing in *Telemedicine Today*,[10] while telemedicine by its etymology simply means 'practicing medicine at a distance', eHealth encompasses more than that—

'ehealth is more about informing patients, electronic billing and mining patient data.' In a review of the perception of eHealth published in the Journal of Medical Internet Research in 2005, it was found that the Internet was mentioned in over half of the published definitions.[11]

Another article in the same issue of that journal explored the scope of eHealth, suggesting that the issues currently dominating eHealth include professional and consumer informatics, and electronic health records. [12] They concluded that the definition of the term eHealth that best represented its place in medical research and healthcare was one given by Gunter Eysenbach in the *Journal of Medical Internet Research* [9]: 'e-health is an emerging field in the intersection of medical informatics, public health and business, referring to health services and information delivered or enhanced through the Internet and related technologies.'

A related concept is that of the hospital information system (HIS), or computers and other technology in the management of the hospital or in the provision of patient care. Traditionally, the healthcare industry has been slower to adopt to new technology: in 1986, hospitals in the United States spent around 2% of operating expenses on IT, compared to 4% in the insurance industry and 12% in banking. [1]

The Healthcare Information and Management Systems Society, a global nonprofit organization for promoting the use of information technology in healthcare, created its EMR Adoption Model to understand the level of electronic medical record capability of a hospital. [2] There are eight cumulative stages in the EMR adoption model, from Stage 0 to Stage 7.

Stage 0 systems have no medical support subsystems like laboratory, pharmacy, and radiology. Stage 1 systems include these ancillary systems. Stage 2 is characterised by the presence of a clinical data repository that centralizes information from

the ancillary systems. Stage 3 systems include a clinical decision support system and nursing documentation. Stage 4 systems allow for computerised physician order entry (CPOE). Stage 5 systems have radio frequency identification (RFID) or other auto identification technologies integrated with the CPOE and pharmacy. Stage 6 means a hospital has full physician documentation, clinical decision support, and a radiological picture archive and communication system (PACS) that totally displaces all film-based image. The final stage, Stage 7, means that the hospital no longer uses paper charts for patient care management, and that clinical information is standardised and easily shared even with non-associated entities (such as other hospitals, clinics, payers, and so on).

The preponderance of health care information on the Internet has allowed professionals and patients alike to turn to it for healthcare advice. Notably, one way that the internet has empowered patients is by allowing them to see their medical records. Although giving patients access to their records is an idea that pre-dates the world wide web, the Internet does make it much easier to implement. [14]

Studies have found benefits to allowing patients access to their records. A 2003 review of the effects of promoting medical records to patients in the Journal of the American Medical Informatics Association highlighted studies where it was shown that accessible records helped reassure patients, improved patient recall, and encouraged patients to take a more active part in their treatment. [15] Additionally, surveys indicate that the patients most likely to want access to their records are those with long-term conditions.[16]

Websites where patients can view their electronic health records are often called patient portals. In the United States, the care consortium Kaiser Permanente[2] began

[2]http://kp.org

providing online health care services in 1996. Use of the site grew over the years: in the year 2004 there were over ten million visits to the site, and by 2007 there were nearly thirty-three million total visits— three times as many as in 2004.

They began their personal health record, My Health Manager, in 2007. The services made available by My Health Manager included a personal health record that allowed members to view data such as lab results, immunizations, and past office visits. [17] By 2008, 27% of the total membership of Kaiser Permanente had registered accounts on the website, and over 60% of those users visited the site two or more times in a six-month period. Viewing test results was the most visited feature on the website.

A study published in the Journal of the American Medical Informatics Association in 2006 detailed the creation and usage statistics of an Internet portal called PatientSite[3], for the Beth Israel Deaconess Medical Center in Boston, Massachusetts. [18] The features on PatientSite allowed patients to contact their doctors for advice or to schedule an appointment, renew prescriptions, review laboratory, and pathology test results, and view and add comments to their medical records.

A report prepared for the California Health Care Foundation[4] in 2011 detailed the impact of patient portals as described by the literature.[19] The aforementioned PatientSite and Kaiser Permanente were among the sites reviewed in the report.

The report noted two measures when assessing patient portals. The first was the success they had at *implementing* the portal in the short-term, and the second was a long-term measure on the *impact* on quality and care and efficiency. The studies also documented the *characteristics* and *uptake* of portal users, i.e. who was using

[3]https://www.patientsite.org/
[4]http://chcf.org

the portals and how often, as well as the opinions and concerns of those users.

The PatientSite study indicated that those who enrolled in the patient portal were on average younger than those who did not enroll. Those who enrolled also had fewer medical problems and visited medical offices fewer times a year. This suggested a divide between healthy, economically advantaged site enrollees and less healthy, disadvantaged non-enrollees.

On the other hand, other studies documented the use of patient portals among individuals with chronic illnesses. A 2007 study of the University of Pittsburgh Medical Center's (UPMC) pilot portal My UPMC[20] obtained feedback from patients with diabetes to find which features would be useful for them. They reported that users saw positive value in the portal.

However, attempts to provide access to personal health records in the United Kingdom have not been as successful as in the USA. In 2007, the publicly funded National Health Service (NHS) began a project that would allow people to access and maintain a summary version of their health record via their HealthSpace[5] website. [21]. But on the 25th of May, 2012, the NHS confirmed that it would close the HealthSpace site by March 2013. This followed a speech earlier that week by Dr Charles Gutteridge, national clinical director for informatics at the UK's Department of Health, in which he said that using the site was "just too difficult". [22] The site closed on the 31st of March, 2013.

In 2008, just 0.13% of the people invited to open accounts had taken advantage of the opportunity, a markedly poorer turnout than the 5-10% that the original business model had hoped for. [23] The researchers found that patients with access to the basic HealthSpace account were disappoint to find their record empty, and were unwilling

[5]http://www.healthspace.nhs.uk

to enter the data themselves. One of the researchers observed the usage or non-usage of HealthSpace of a subset of 20 people with diabetes. Only three of the twenty invited agreed to try HealthSpace, and seventeen refused.

Some of the seventeen who did not use HealthSpace did not have computers or internet access at home, or did not see these technologies as useful in this particular area of their lives. Others were already documenting other ways of monitoring their condition. The three who agreed to try HealthSpace abandoned it soon after first accessing it and further declared that they were uninterested in using it again. This was in stark contrast with the results observed by the US-based Kaiser Permanente. This came as a disappointment to the researchers, because HealthSpace was inspired by the very successful Kaiser model.

III. Theoretical Framework

A. Electronic Health Record

An electronic health record (EHR) is a record that documents a patient's medical and health information on a continuing basis. Although the terms electronic health record and electronic medical record (EMR) are often used interchangeably, the difference, according to the US Department of Health's Office of the National Coordinator for Health Information Technology (ONC) is that the EMR is the legal patient record, while the EHR merely uses the EMR as a data source. The EHR is only what the patients, physicians, and healthcare providers are given access to. [24]

B. Hospital Information System

A hospital information system (HIS) refers to the technology used in hospital administration and management and the provision of patient care. These systems take care of administrative tasks such as patient billing as well as clinical tasks like the management of medical tests.

C. Patient Portal

A patient portal is a website through which patients can access their personal health records. A typical patient portal enables users to review their results, complete forms, and contact their healthcare providers. A patient portal enhances patient access and increases efficiency and productivity. [19]

Portals have been shown to make people pay more attention to their health, and benefits not only the patient but the health-care providers themselves, because their use will result in more efficiency and more cost-effective measures. A hospital does not have to hire numerous employees to take phone calls, allowing a hospital to optimise its resources by employing staffers to concentrate on other, more labor-intensive aspects of providing healthcare.

Despite the advantages of the adoption of patient portals, concerns about cost and liability are among the many issues cited as barriers to implementation. However, as consumers have started to manage the basic transactions of other aspects of their lives on the internet— such as online shopping and online banking— they have shown their willingness to adopt online health care services as well.[25]

D. Database Management System

A database system is an organised collection of data. Databases are managed using software called database management systems (DBMS), which facilitates the creation, management, storage and retrieval of the data in the database. The DBMS manages both the data itself and the data structures set up to support it.

A relational database is a database which makes use of the entity-relationship model for describing the data contained within it. Data is stored in tables, each uniquely identifiable by a primary key.

E. Data Warehouse

The data warehouse acts as a repository for the significant data used by various systems. It is the integration of data from several sources to create a central repository.

F. Clinical Data Repository

Also known as a clinical data warehouse, a clinical data repository consolidates data from many clinical sources to present a unified view of a single patient. This is useful for clinicians in that it allows them to retrieve data from a single source rather than go to specific clinical departments for different types of information.

G. Model-View-Controller

The Model/View/Controller (MVC) is a pattern used in software architecture that is used to build user interfaces. It was first introduced by Trygve Reenskaug of the University of Oslo for use in Smaltalk-76, and although it was originally developed for personal computing, it has seen wide adoption as the architecture of choice for applications on the world wide web.[26] The MVC consists of three kinds of objects: the Model is the application object, the View is its presentation on screen, and the Controller defines how the information model changes based on the user input.

- The **model** manages the retrieval of user data. The model layer concerns itself with the business logic of the application: retrieving data, processing and validating it.

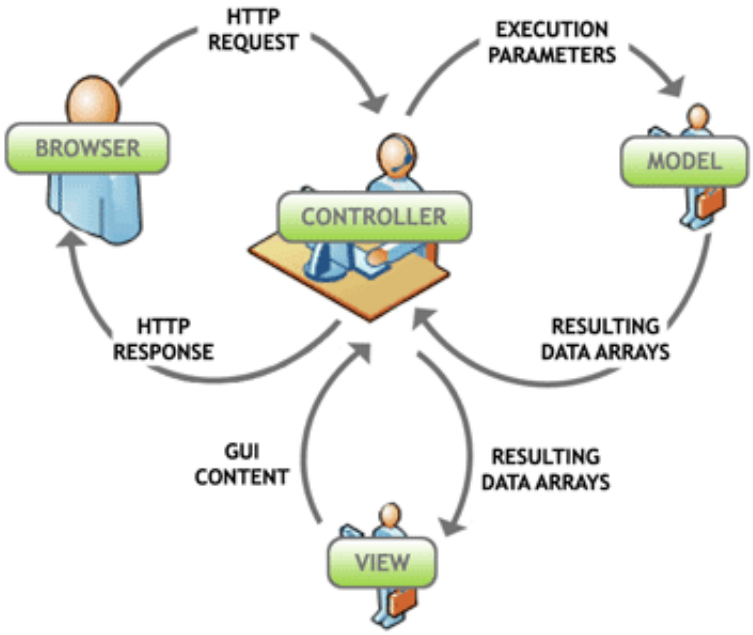

Figure 1: the Model View Controller pattern as it is frequently used in web development

- The **view** layer renders a presentation of the data retrieved by the model. By keeping the view separate from the model, the application is able to present the information in multiple different ways, in whatever format suits the application's users.

- The **controller** layer is the middleman between the application and its users. Once the data provided by the model is arranged in a view and presented to the user, the controller handles requests from the user, and is also responsible for passing this information back to the system.

The advantage of MVC is the clear separation of presentation and application logic. This facilitates support for different types of users depending on their needs - the model returns the same data, but the controller chooses a different view to render them.

H. Laravel

Laravel[6] is an open-source PHP 5.3 web application framework written by Taylor Otwell. It was created to take care of common tasks in development in web projects, such as authentication, routing, sessions, and caching. The goal of a framework such as Laravel is to take the tedium out of project development, making using it much easier than writing everything from scratch. It provides powerful tools needed in many applications, and has extensive documentation available. [27] The Laravel framework allows for the use of the MVC architecture.

[6]http://www.laravel.com

IV. Design and Implementation

A. Entity Relationship Diagram

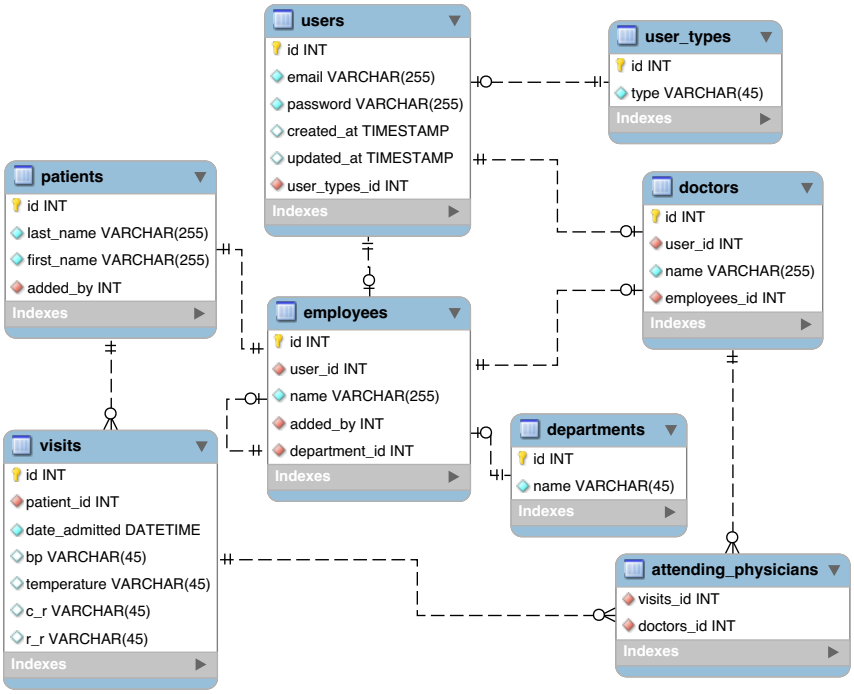

Figure 2: Entity Relationship Diagram

Figure 2 shows the main Entity Relationship Diagram (ERD) for the system. Each user is identified by an automatically-incrementing `id` as a primary key. The other information contained in the user table is the `email` and `password` used to log in to the system, as well as a boolean to indicate whether that user is active. The `user_types_id` field is used to identify which of the user groups (patients, physicians, medical technicians, and so on) the user belongs to.

The `employees` entities are created as records for employees of the hospital such as doctors, system administrators, clerks, hospital administrators, and medical technicians. Each table identifies the person by their `employee_id`, name, and contains an `added_by` field to identify who added the user to the database.

Similarly, patients have a record in the `patients` table. Since the `patients` table is relatively long, many of the fields have been omitted from the ERD in Figure 2. However, these fields are listed and explained in the Data Dictionary section below.

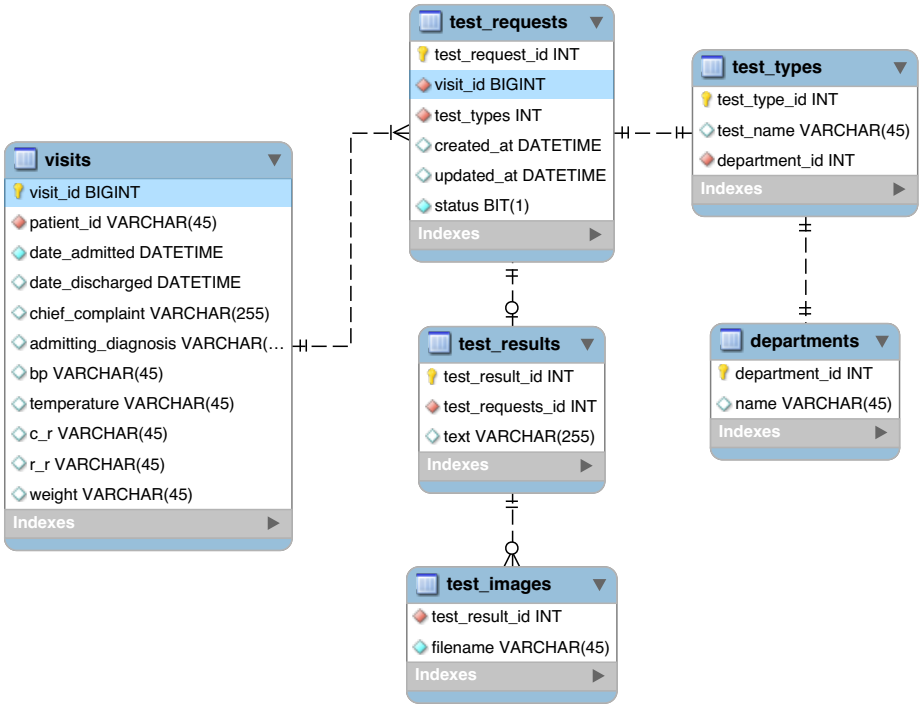

Figure 3: Entity Relationship Diagram for tests

Every time a patient visits the hospital for testing, a `visit` record is created, with a unique `visit_id`, and a foreign key identifying the patient. Included in the table are the date of the visit and the admitting diagnosis and primary complaints of the patient, if applicable, as well as other fields which are omitted in Figure 3 but are present in the Data Dictionary section below.

Associated with each `visit` is a table called `attending_physicians`, which indicates which doctors are attending to the patient for that visit.

Whenever a doctor wishes to order a test for one of his patients, an entry under

test_requests is created. The test_request table contains a visit_id foreign key linking it back to the visits table, test_request_id as a primary key, and a test_type_id that is a foreign key of the test_types table that indicates what type of test it is.

Whenever a new test_types entry is created, a new table is created, the name of which starts with test_. It is in this table that the test result information is contained. And since there may be many different types of tests, the structures of these tables may vary from each other. What all these tables have in common is that they have a test_result_id linking them back to the entry in the test_results table that in turn has a foreign key associating it with the entry in test_requests that corresponds to the test request to which it is a result.

In the Data Dictionary section below, examples of test result tables are presented. However, the system allows for many more tables to be created, and thus, once in place, the database may have more tables added to it.

B. Context Diagram

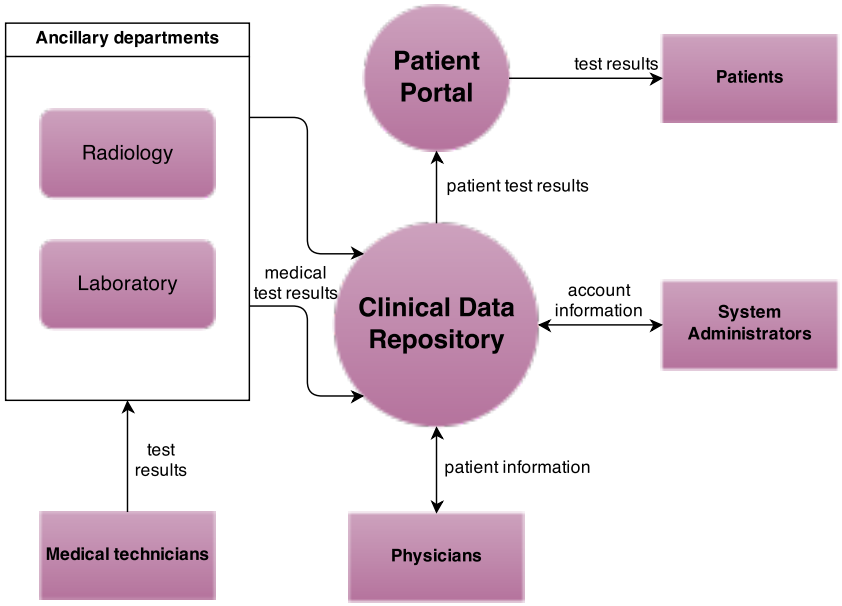

Figure 4: Context Diagram

The context diagram in Figure 4 shows the way the different types of users interact with the system.

Medical technicians who belong to the ancillary laboratory and radiology departments upload the results of medical test procedures to the central data warehouse of the clinical data repository, at which point it can be accessed directly by the physicians associated with the hospital, or through an online portal by the patients to whom the results belong.

C. Data Flow Diagram

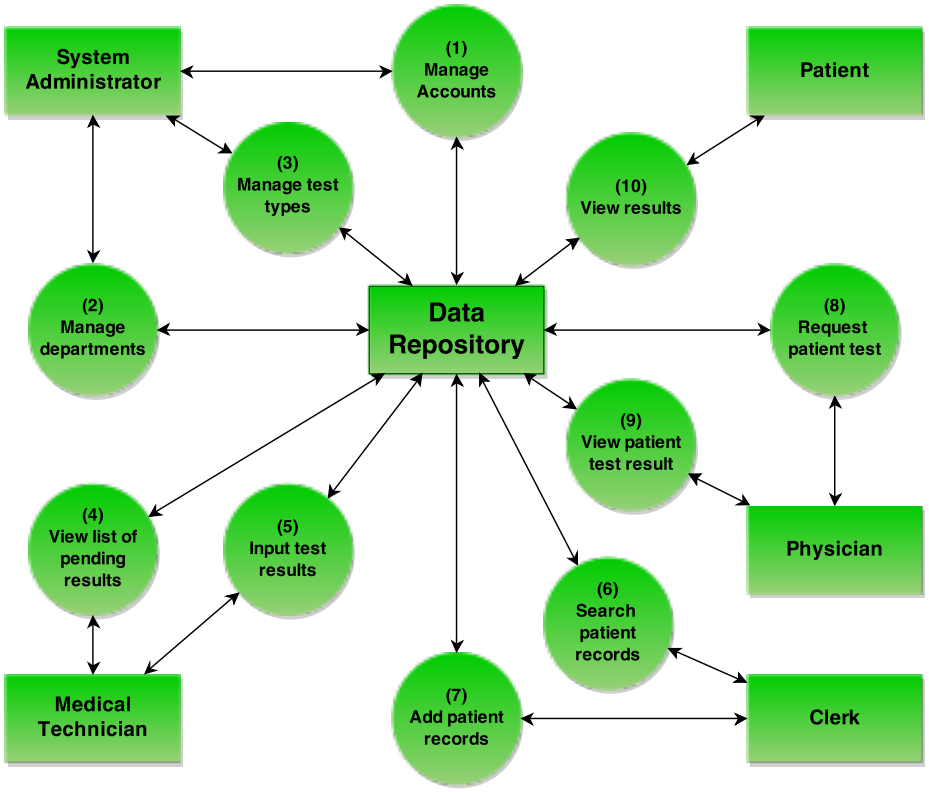

Figure 5: Data Flow Diagram

Figure 5 shows the top level data flow diagram.The proposal details the main processes of the system: (1) manage accounts, (2) manage departments, (3) manage test types, (4) view list of pending results, (5) input test results, (6) search patient records, (7) add patient records, (8) request patient test, (9) view patient test results, and (10) view results as a patient.

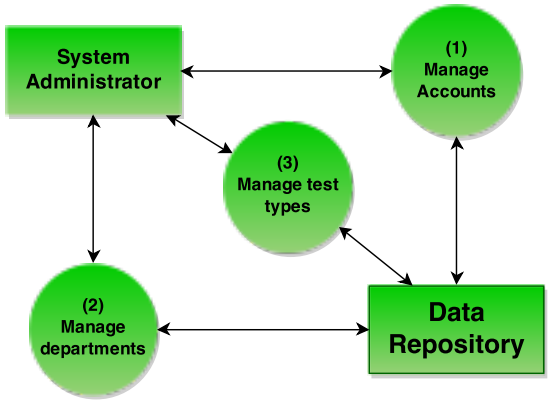

Figure 6: Data Flow Diagram for system administrator functionalities

The three functionalities of the system administrator, as shown in 6, are the management of user accounts, test types, and departments. In (1), the task of the system administrator is to create and delete accounts for other users of the system as necessary, likewise for (2), the management of departments in the hospital.

For the third functionality, manage test types, if a new type of medical test is to be offered by the hospital, a system administrator may use this functionality to create a new table in the database for said test.

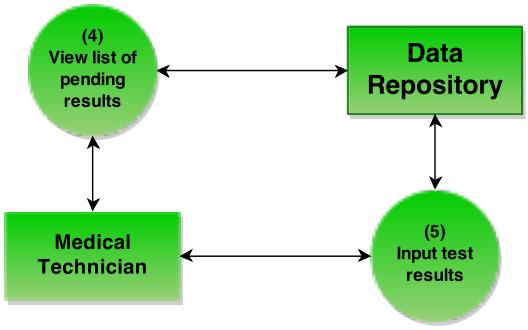

Figure 7: Data Flow Diagram for medical technician functionalities

Figure 7 shows that the medical technician has two main actions to perform in the system. The first is (4), viewing the list of pending results. When a medical technician has logged in to their account in the system they are presented with a list of medical test requests that have either not yet been performed or have not

yet been added to the database. When the results from said medical tests become available, the functionality of (5), input test results, comes into play. The medical technician uploads the results, the test request is marked as fulfilled, and the doctors and patients concerned may view the results.

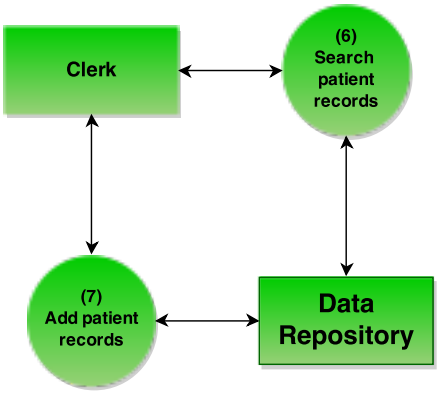

Figure 8: Data Flow Diagram for clerk functionalities

Figure 8 details the two functions performed by the clerk in the system. Both are accomplished whenever a patient visits the hospital. If the patient has indicated that they have already visited the hospital previously, the clerk may search the database of existing patients (5) for a record that matches the current patient; if a record for that patient is found they need not create a new entry for the patient, merely register a new patient visit.

If the patient does not have a record, the (7) add patient record functionality may be used to create a new record for the patient. Whenever a visit record is created – be it for old or new patients– the visit is marked as pending (not yet attended to by the physician) and assigned to a doctor (who acts as the attending physician for that patient). The record of this visit can then be viewed by the doctor.

As seen in Figure 9, a doctor can view list of patients pending – the patients processed by the clerk (as seen in Figure 8) but not yet attended to. From this list, the doctor may click on each, enter initial diagnosis and request tests for the patient.

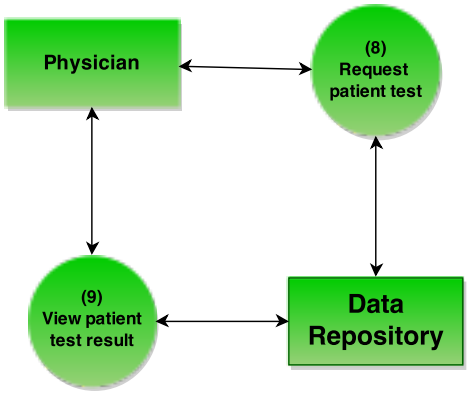

Figure 9: Data Flow Diagram for physician functionalities

When a doctor requests tests for the patient, test requests are created, marked pending and are attended to by the medical technicians in the associated departments.

When the medical test results of the patient arrive, the doctor may view them and based on the information given, order more tests if the results are inconclusive or give the patient a final diagnosis.

If an old patient wishes to follow up on a pending diagnosis they need only indicate to the secretary that they are there for a follow-up visit, and the details of that particular visit appear in the doctor's inbox.

Figure 10: Data Flow Diagram for patient functionalities

Finally, in Figure 10 we have the 'view results' functionality available to patients. Once results become available (as detailed in Figure 7), a patient may access them from the website.

D. Data Dictionary

The following tables show the definitions for the fields in the database.

Field Name	Type	Description
id	BIGINT	PK; ID assigned to each user
user_type	ENUM	Type of user: patient, doctor, data encoder, clerk, system administrator
email	VARCHAR	Email address. Must be unique, no two accounts can use the same email address
password	VARCHAR	Password

Table 1: users

Field Name	Type	Description
department_id	BIGINT	PK, auto generated
department_name	VARCHAR	name of hospital department

Table 2: departments

Field Name	Type	Description
employee_id	VARCHAR(45)	Employee ID
user_id	BIGINT	User ID associated with the account in the system
first_name	VARCHAR(45)	First name
last_name	VARCHAR(45)	Last name
department_id	BIGINT	Foreign key of departments, indicates the department the encoder belongs to

Table 3: employees

Field Name	Type	Description
patient_id	VARCHAR(45)	Patient ID assigned by the hospital
first_name	VARCHAR(45)	First name
middle_name	VARCHAR(45)	Middle Name
last_name	VARCHAR(45)	Last name
birthdate	DATE	date of birth
birthplace	VARCHAR(45)	place of birth
sex	ENUM('male','female')	gender
civil_status	ENUM('single',...)	Civil status
telephone_number	VARCHAR(45)	Landline telephone number
mobile_number	VARCHAR	User's mobile telephone number
is_dependent	BIT(1)	
is_stockholder	BIT(1)	
is_stockholder_dependent	BIT(1)	
address	VARCHAR(255)	Patient's address
religion	VARCHAR(45)	Patient's religion
nationality	VARCHAR(45)	Patient's nationality
occupation	VARCHAR(45)	Patient's occupation
company_name	VARCHAR(45)	Name of company
company_address	VARCHAR(255)	Company's address

Table 4: patients

father_name	VARCHAR(45)	Patient's father's name
mother_name	VARCHAR(45)	Patient's mother's name
parent_address	VARCHAR(45)	Patient's parents' address
spouse_name	VARCHAR(45)	Patient's spouse's name
spouse_occupation	VARCHAR(45)	Patient's spouse's occupation
spouse_company	VARCHAR(45)	Patient's spouse's company
emergency_contact_name	VARCHAR(45)	Name of emergency contact person
emergency_contact_address	VARCHAR(255)	Name of emergency contact person's address
emergency_contact_number	VARCHAR(45)	Telephone number of emergency contact person
account_responsibili_name	VARCHAR(45)	Person responsible for the account
account_responsibili_number	VARCHAR(45)	telephone number of person responsible for the account
account_responsibili_address	VARCHAR(255)	address of person responsible for the account
account_responsibili_relationship	VARCHAR(45)	relationship of patient and person responsible for the account
added_by	VARCHAR(45)	FK

Table 5: patients cont'd

Field Name	Type	Description
visit_id	BIGINT	Primary key identifying the visit; automatically generated
patient_id	VARCHAR(45)	Foreign key from patient_record identifying the patient
date_of_visit	DATETIME	Date and time patient was admitted to the hospital
chief_complaint	VARCHAR(255)	Chief complaint
admitting_diagnosis	VARCHAR(255)	Admitting diagnosis
final_diagnosis	VARCHAR(255)	Final diagnosis
bp	VARCHAR(45)	Blood pressure
temperature	VARCHAR(45)	Temperature
c_r	VARCHAR(45)	
r_r	VARCHAR(45)	
weight	VARCHAR(45)	Weight
added_by	VARCHAR(45)	FK
status	BIT(1)	

Table 6: visits

Field Name	Type	Description
visit_id	BIGINT	ID of the patient; foreign key from visit
doctor_id	BIGINT	ID of the attending physician; foreign key from employees

Table 7: attending_physicians

Field Name	Type	Description
id	BIGINT	PK, automatically generated
test_type_id	BIGINT	Foreign key of test_types table
visit_id	BIGINT	foreign key of visit table
status	BIT(1)	pending(0)/done(1)

Table 8: test_requests

Field Name	Type	Description
id	BIGINT	PK, auto generated
department_id	BIGINT	FK departments table
name	VARCHAR	name/description of test
fields	VARCHAR	json describing fields of test
table_name	VARCHAR	name of table in which results are contained

Table 9: test_types

Field Name	Type	Description
test_results_id	BIGINT	PK, automatically generated
test_request_id	BIGINT	FK of test_requests table
added_by	VARCHAR(45)	FK
date_added	DATETIME	

Table 10: test_results

Field Name	Type	Description
cholesterol	FLOAT	
triglycerides	FLOAT	
hdl	FLOAT	
ldl	FLOAT	

Table 11: test_lipid_profile

Field Name	Type	Description
sgpt	FLOAT	
sgot	FLOAT	
alkaline_phosphate	FLOAT	
bilirubin	FLOAT	
amylase	FLOAT	
lipase	FLOAT	
tpag	FLOAT	

Table 12: test_liver_profile

Field Name	Type	Description
hemoglobin	FLOAT	
hematocrit	FLOAT	
red_blood_cells	FLOAT	
white_blood_cells	FLOAT	
segmenters	FLOAT	
lymphocytes	FLOAT	
eosinophils	FLOAT	
monocytes	FLOAT	
mcv	FLOAT	
mch	FLOAT	
mchc	FLOAT	
rdw_cv	FLOAT	
platelet_count	FLOAT	
hba1c	FLOAT	
fbs	FLOAT	
hgt	FLOAT	
bun	FLOAT	
creatinine	FLOAT	
uric_acid	FLOAT	
ptt	FLOAT	
ogtt	FLOAT	
gct	FLOAT	

Table 13: test_hematology

Field Name	Type	Description
sodium	FLOAT	
potassium	FLOAT	
chloride	FLOAT	
calcium	FLOAT	
magnesium	FLOAT	
phosphorus	FLOAT	

Table 14: test_electrolytes

Field Name	Type	Description
color	FLOAT	
transparency	FLOAT	
reaction	FLOAT	
sp_gravity	FLOAT	
sugar	FLOAT	
protein	FLOAT	
wbc_hpf	FLOAT	
rbc_hpf	FLOAT	
epithelial_cells	FLOAT	
amorp_urates	FLOAT	
bacteria	FLOAT	

Table 15: test_urinalysis

Field Name	Type	Description
FT3	FLOAT	
FT4	FLOAT	
T3	FLOAT	
T4	FLOAT	
TSH	FLOAT	
PSA	FLOAT	

Table 16: test_thyroid

Field Name	Type	Description
HBSag	FLOAT	
HBeAG	FLOAT	
anti_HBs	FLOAT	
anti_HBc_IgG	FLOAT	
anti_HBC_IgM	FLOAT	
anti_HBe	FLOAT	
anti_HAV_IgG	FLOAT	
anti_HAV_IgM	FLOAT	
anti_HCV	FLOAT	
anti_HBe	FLOAT	
HBa1c	FLOAT	
FBS	FLOAT	

Table 17: test_thyroid

Field Name	Type	Description
troponin_I	FLOAT	
cpk_mb	FLOAT	
cpk_total	FLOAT	

Table 18: test_cardiac

V. Architecture

A. System Architecture

Composer[7] is used to manage the PHP dependencies of the project, including the Laravel framework, Cartalyst Sentry user management system. All the dependent libraries are declared in `composer.json`. Grunt[8] and Bower[9] (which are both dependent on NPM[10]) are used to manage front-end javascript packages. Packages are installed using Bower by declaring them in `bower.json` and are stored in the `public/components/` folder, and the necessary javascript and css files are organized and minified using the Grunt task runner (`Gruntfile.js`).

B. Technical Architecture

The patient portal is a web application following the client-server architecture. The following shows the ideal configuration. The web service is run on a host server that supports PHP and the Laravel framework. The host service must satisfy the following minimum system requirements:

- 2GHz processor or faster
- 16 GB of RAM or higher
- Broadband internet connection

[7] http://getcomposer.org
[8] http://gruntjs.com
[9] http://bower.io/
[10] https://www.npmjs.org/

- Microsoft Windows, Mac OS X, Linux, or any operating system that supports at least PHP 5.3.7
- The MCrypt PHP extension
- A MySQL database engine

The web service can be accessed by the client on a computer, smartphone, or tablet which has the following requirements:

- Google Chrome
- 256 MB of RAM or higher
- Adobe Reader, or any software capable of displaying PDF files
- A screen resolution of at least 1024x768 is advised, but not necessary.

VI. Results

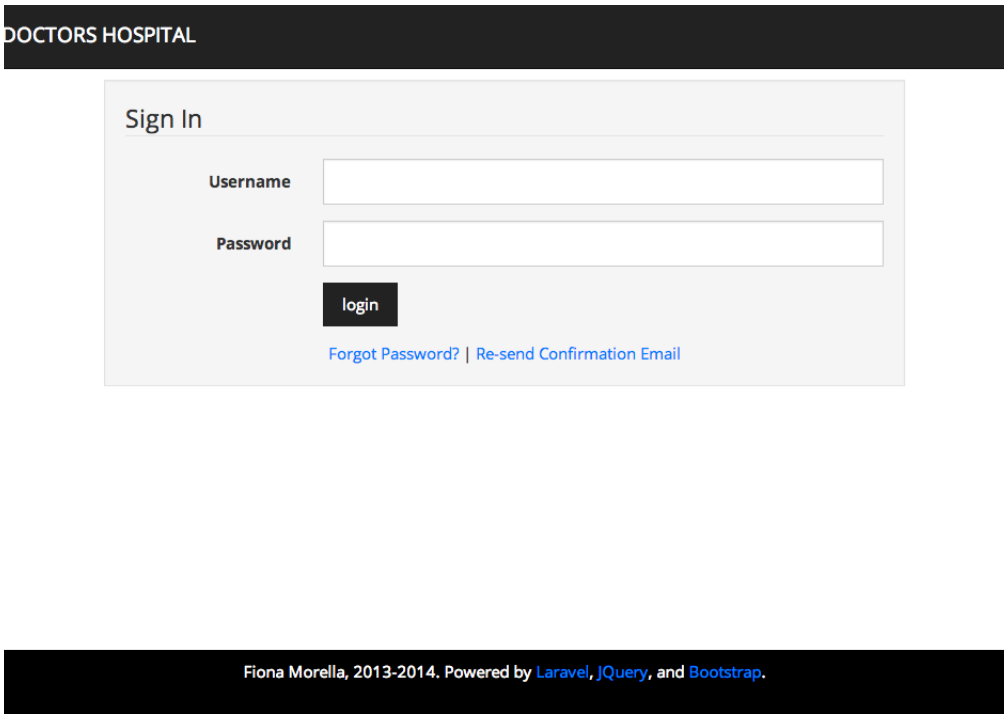

Figure 11: Login

Figure 11 shows the site's log in screen. A returning user who already has an account can enter their email and password to log in.

A user must have clicked on the activation email they were sent upon account creation to activate their account. Unactivated users are not allowed to log in. If for some reason the use did not see the activation form, they may choose to resend it by clicking on 'Re-send confirmation email' and filling their email into the form, as in Figure 12. Figure 13 shows the email that is sent.

If a user has forgotten their password, they may reset their password by clicking on the 'forgot password' link, as seen in Figure 14. After they have entered their email address in the form and submitted it, an email containing a link resetting their password will be sent to that address.

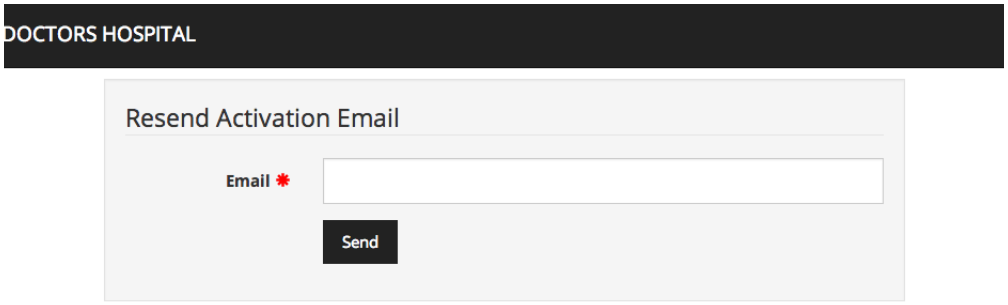

Figure 12: Resend Activation Email Form

Dear Bill Gates,

Thank you for signing up for the Las Piñas Doctors Hospital Information System. You may now activate your account at http://localhost:8888/laravel/portal/public/activate/35/VQLkXTIw6TdrvgMySollxFrMRKIFwIpr9ukRaTd68t.

Your temporary login details are as follows:

Username:

GatesBill

Password:

mc9Kp5BR

You may change your password once your account has been activated.

You have received this email because an account was recently registered with this email address on the LPDH system. If you did not authorize this action, please ignore this email.

Figure 13: Activation Email

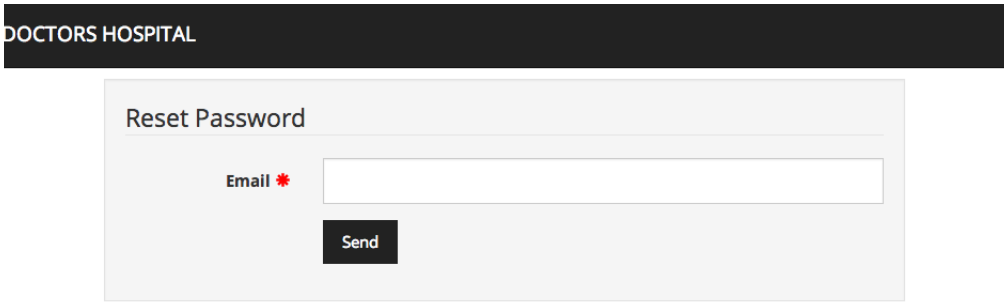

Figure 14: Forgot Password Form

Once they have clicked on that link, they will be sent to a page in which they must enter the new password of their choice (Figure 15). Once the form has been submitted, they may log in with the new password.

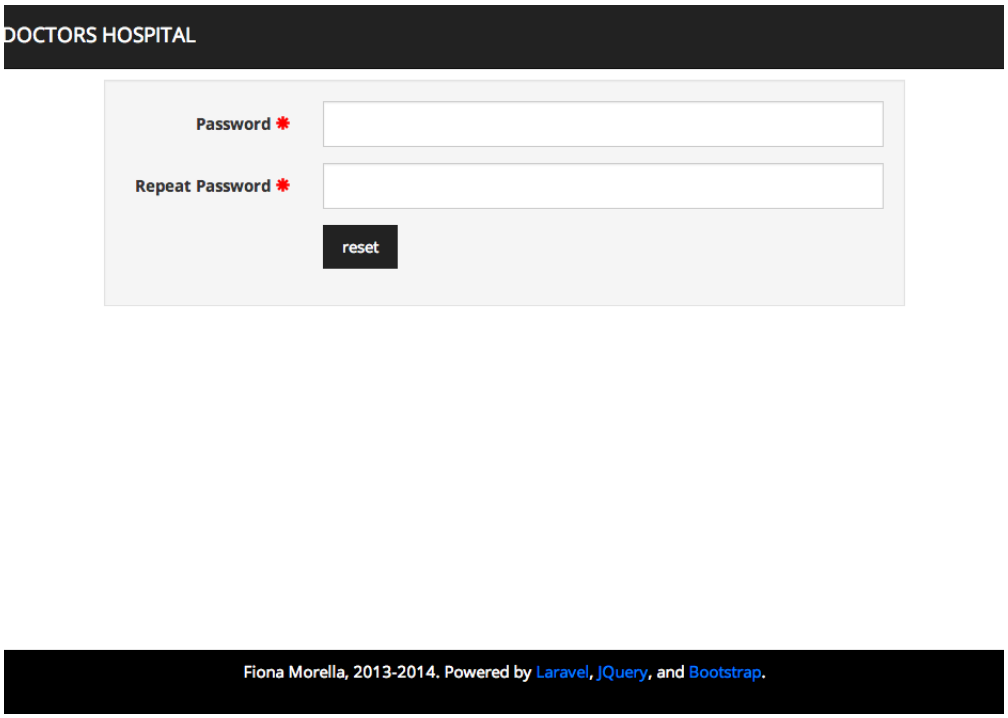

Figure 15: Reset Password Form

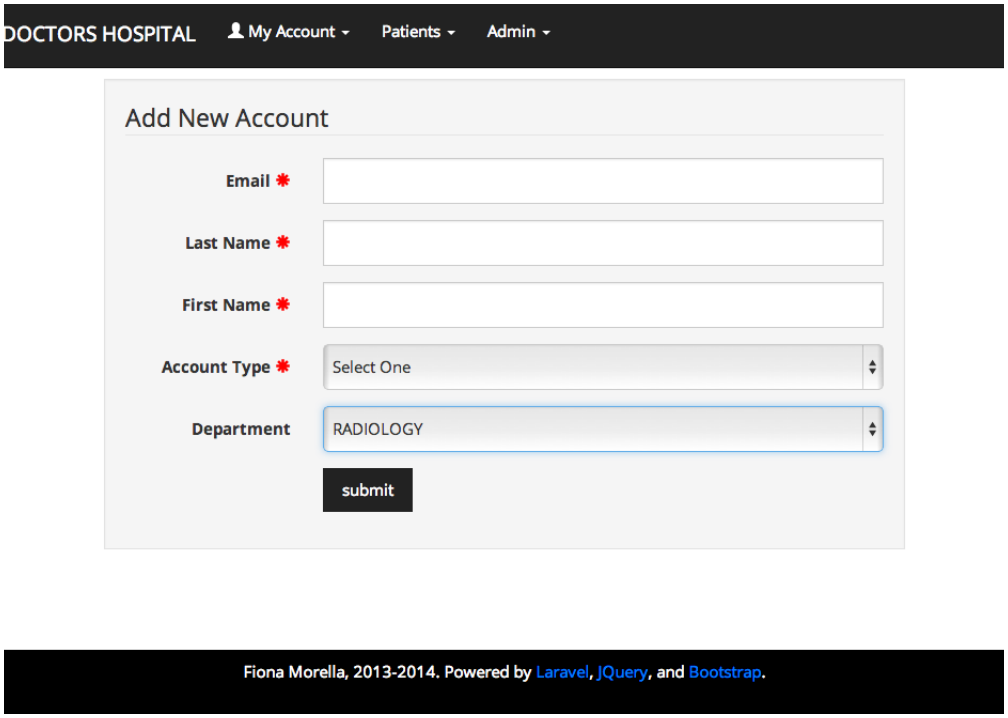

Figure 16: Add new account

Accounts are created by either the clerks or the system administrators. The system administrator is tasked with creating new accounts for employees of the hospital such as physicians, medical technicians, clerks, hospital administrators, and other system administrators. Figure 16 shows the form for account creation. The email address, last name, first name, and account type is required. The employee may also be assigned to a department.

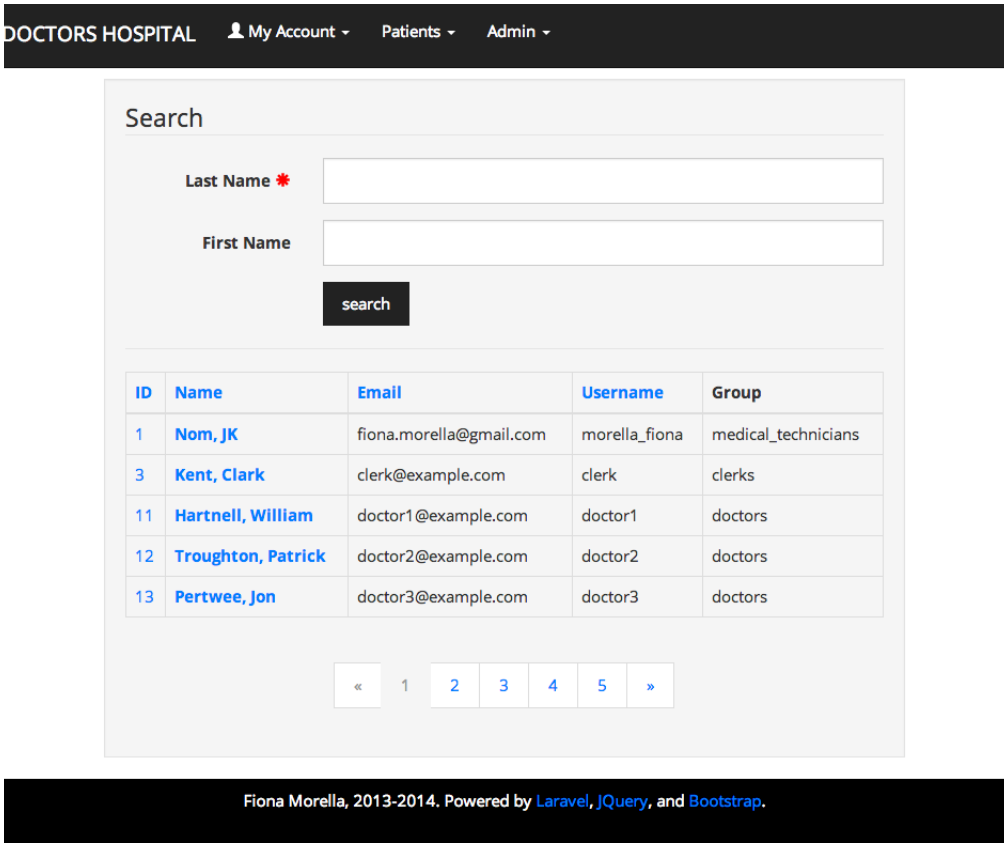

Figure 17: User index

Figure 17 shows the user index page, where it is possible to search and sort through users registered in the database. The name of each user is a hyperlink to another page that displays the basic account information for that particular user (Figure 18). From this page, the system administrator may also disable the account so that it may no longer be used to access the system, in case the user ceases to be affiliated with the hospital and as such will be granted no more access to the system.

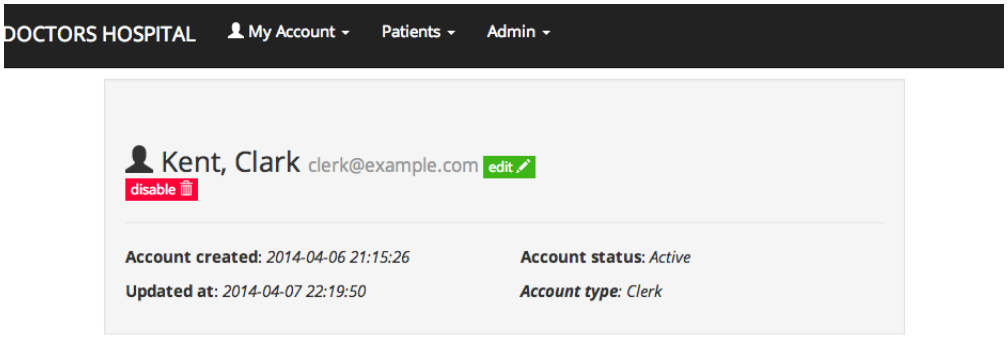

Figure 18: User information

In addition, the system administrator is also charged with adding new departments or to the database. Seen in Figure 19 is a page that displays all the departments and also allows the user to create a new one, edit the name of one, or disable it.

When a new department is created, unaffiliated medical technicians already in the database can be assigned to it, or new employees can be added and new accounts added to the database as necessary.

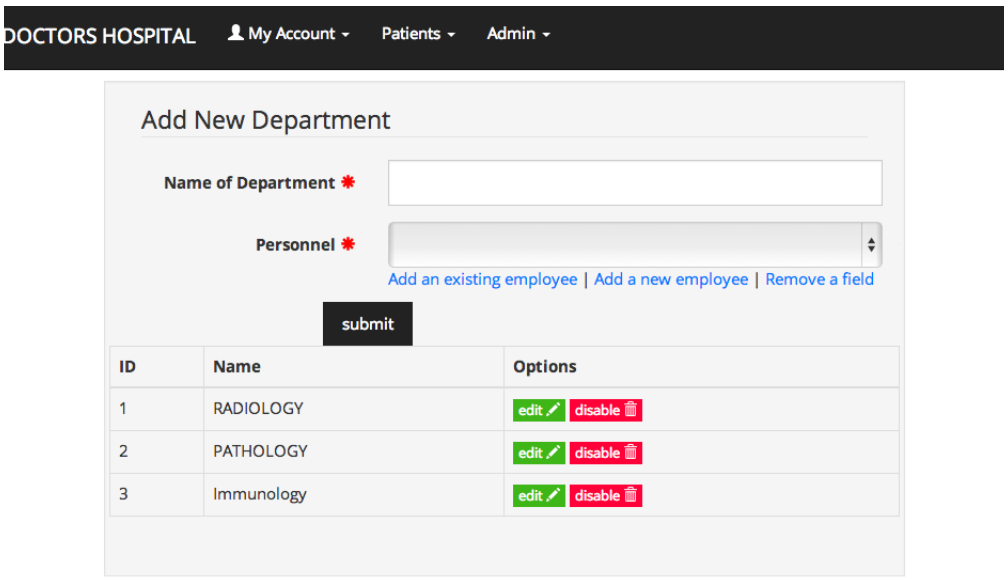

Figure 19: Manage departments

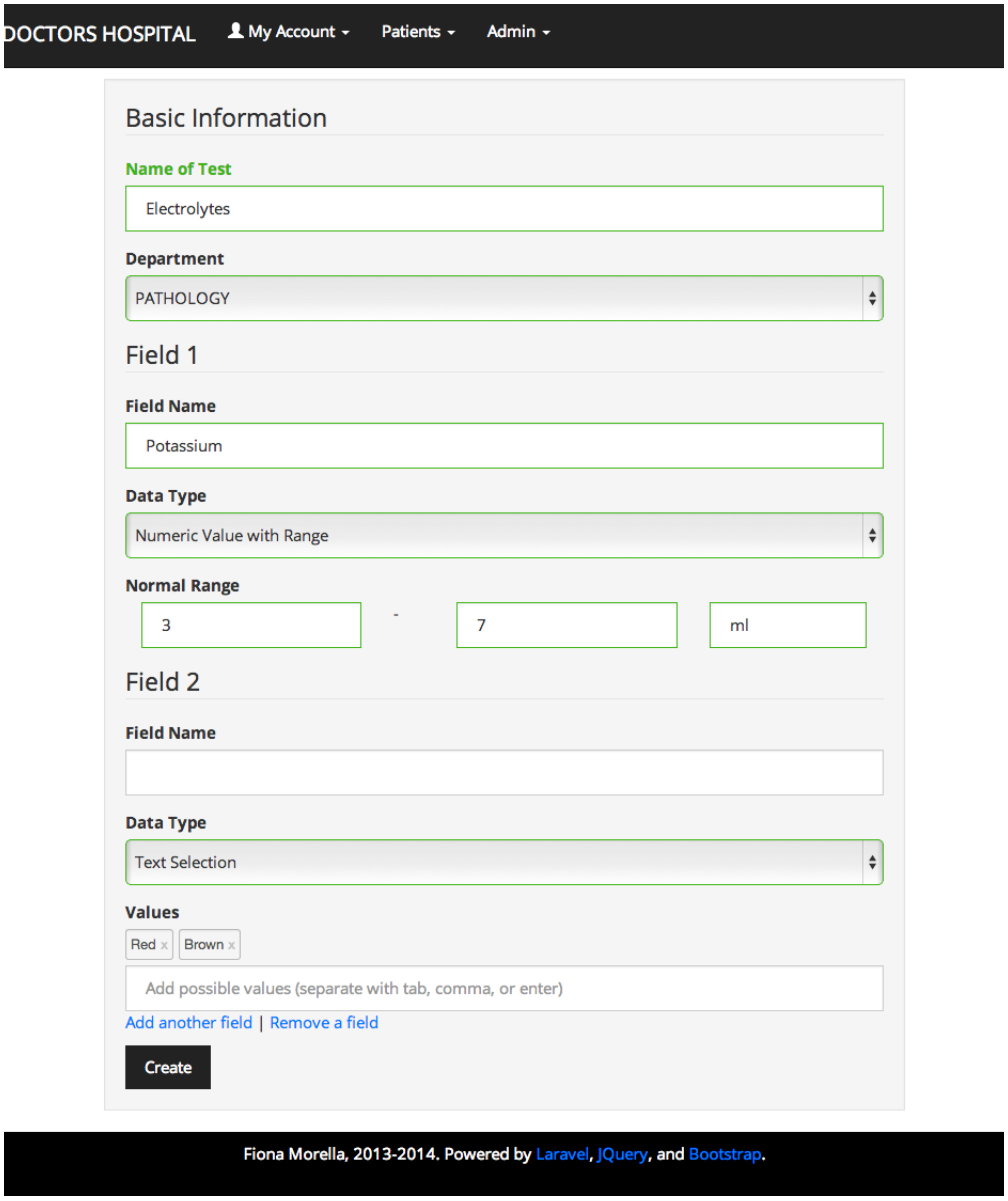

Figure 20: Add new test type

In Figure 20 is the page for the creation of new medical tests. The name of the test and the department it belongs to are the basic required fields. The test can also have a number of different fields each representing values to be tested for. The field can be of type image, a number, or enum.

If the data type of the field is a number, a minimum and maximum must also be filled in to represent the normal value. A unit of measurement is also required.

If the field is enum, preset values for the field may be added by typing into the field marked 'values' and separating each separate value by using comma, tab, or enter.

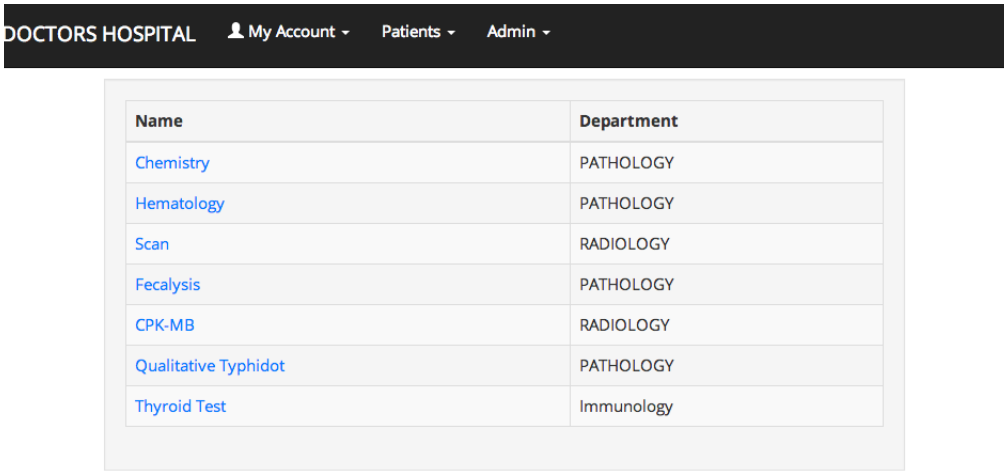

Figure 21: List of tests

The page containing an index of all the test types - listing the name of the test and the department of the test - can be seen in Figure 21. Clicking on the name of a test will show the page in Figure 22 where the fields of that particular test are listed.

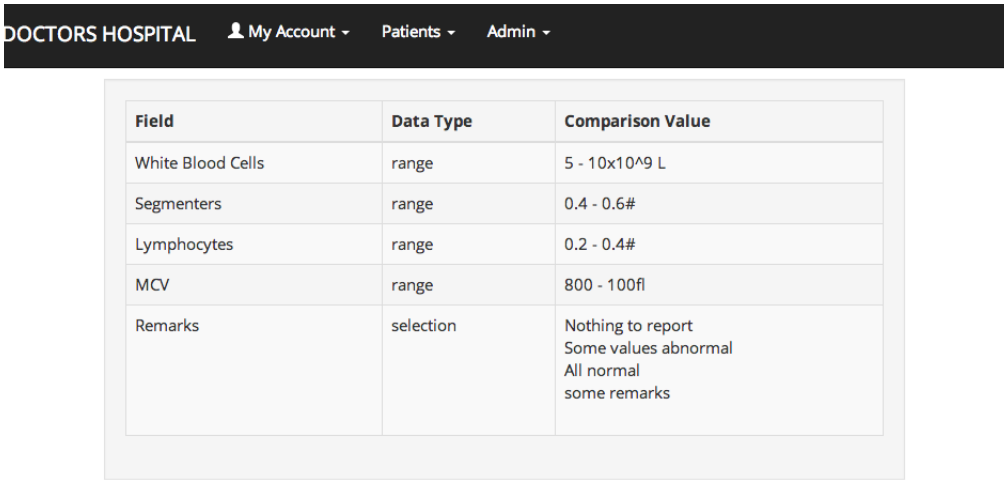

Figure 22: Test type

A functionality open to certain users (clerks, physicians, hospital administrators) is the ability to view an index and search page listing all the records of patients, as seen in Figure 23. From this page, clerks (and only clerks) may click on the add button in each entry to add a new visit entry for that particular patient.

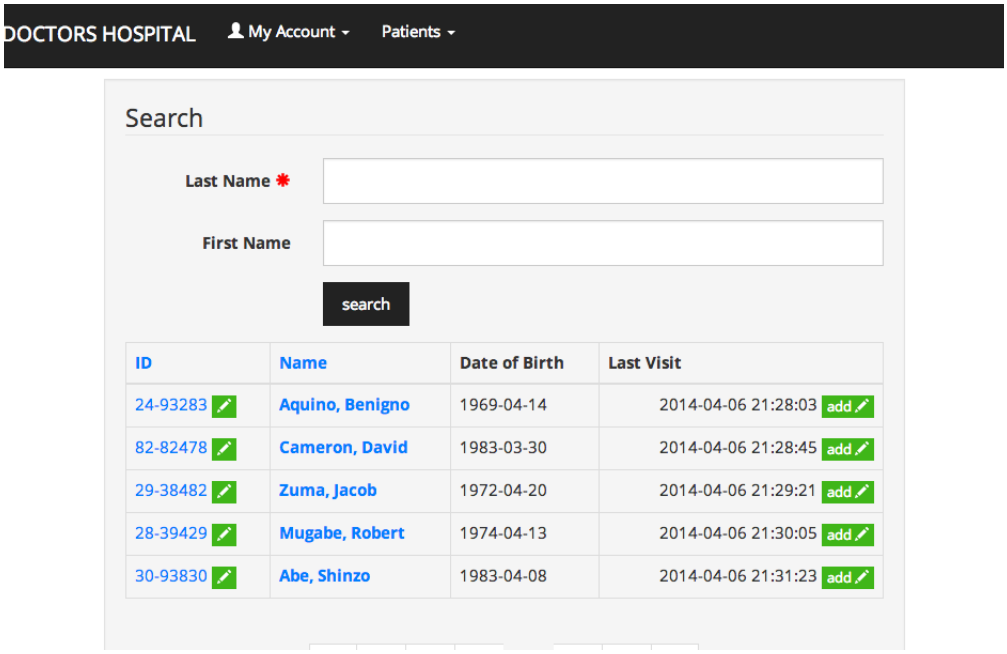

Figure 23: Patient index

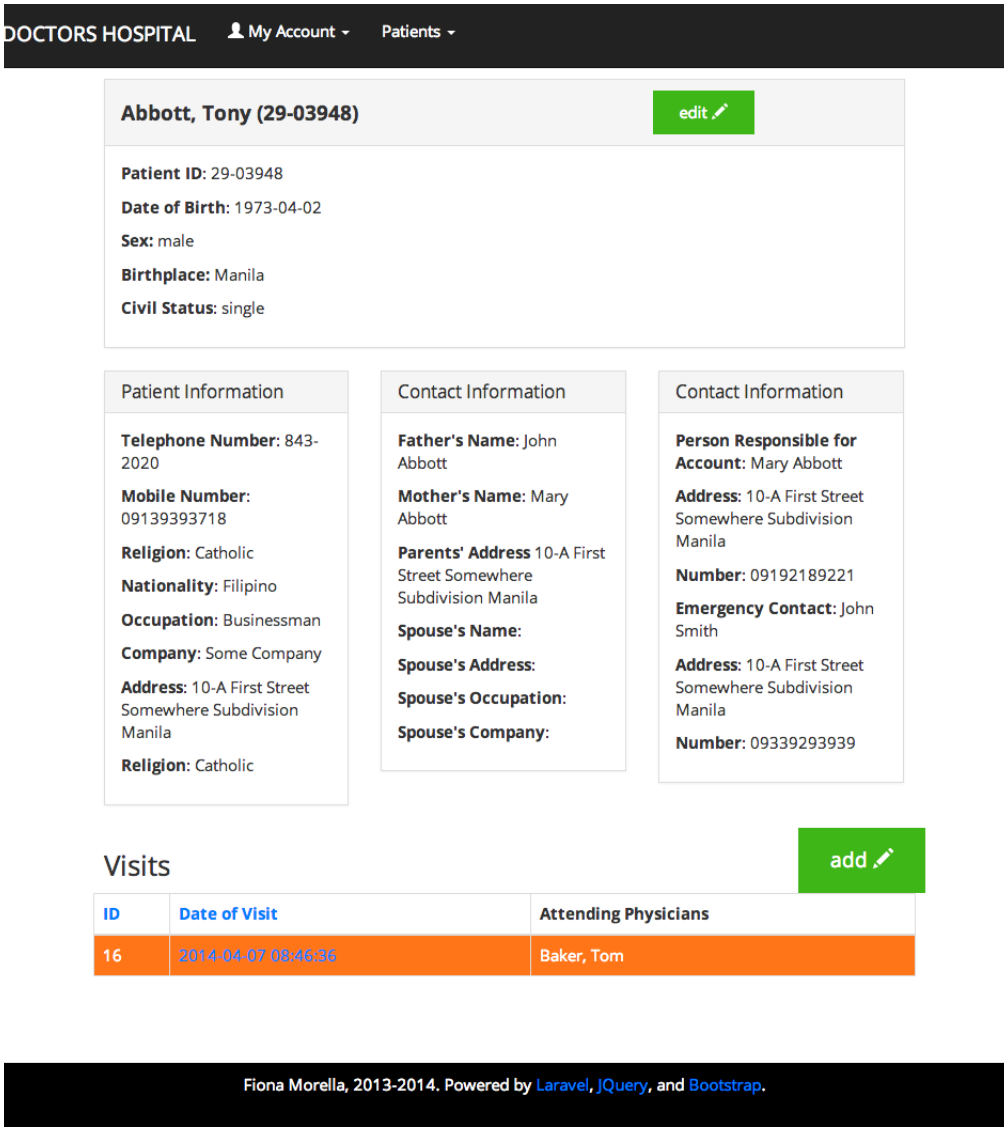

Figure 24: Patient record

Clicking on each patient's record takes the user to a page that displays information about the patient, such as their date of birth, contact number, address, and emergency contact details. Also visible is a list of every visit the patient had at the hospital.

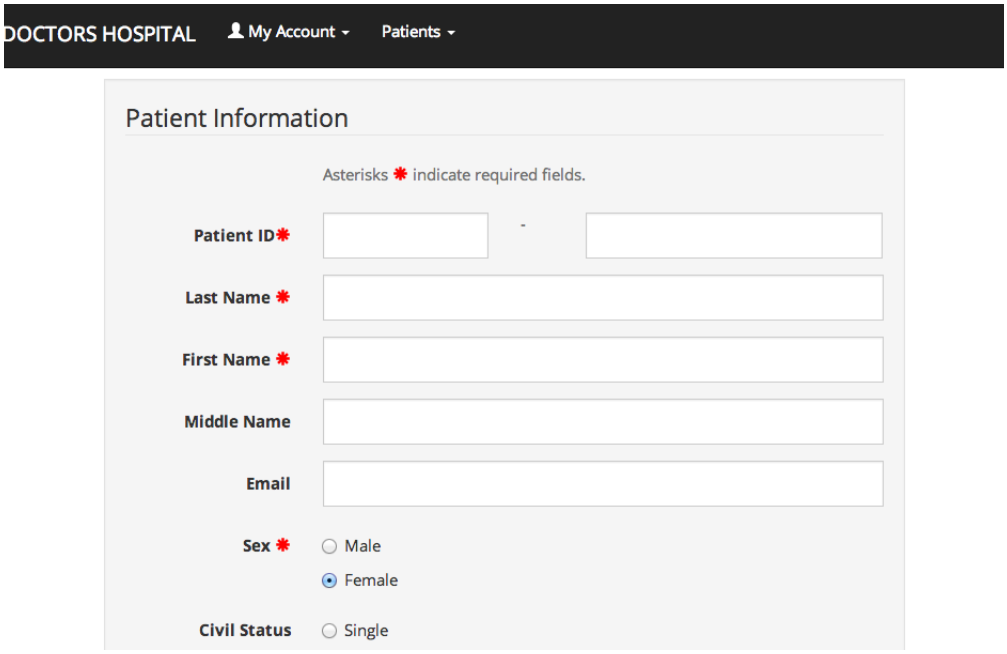

Figure 25: Add Patient

Figure 25 & 26 shows the page clerks must fill in to add new patients and visits to the system. In addition to required information such as the patient ID, name, and gender of the patient, the clerk must also fill in a valid email address in the email field to create an account for that user. Other information, such as the patient's contact information and other such details, are also given fields, but are not mandatory.

The other mandatory field in Figure 26 is the attending physician assigned to the patient for that visit.

If the patient is new and does not yet have a record, a new patient record must be created for them, using the form as seen in Figures 25 and 26.

However, if a patient already has a record, and the 'add visit' button (as seen in Figure 23) is clicked, the page for adding a visit only contains the bottom half of the form, the fields in Figure 26, since the patient information obtained from Figure 25 is already present in the system.

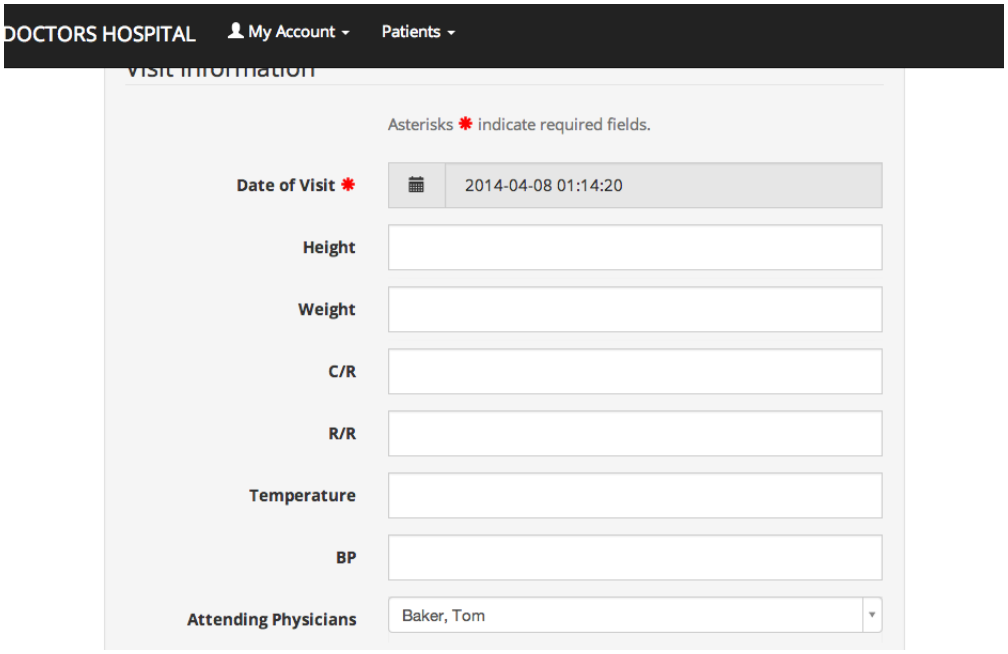

Figure 26: Add Visit

When the clerk creates a new patient record or adds a new visit to a patient record, the patient's name is added to the list of pending patients on the page in Figure 27. These patients disappear from the list when they are attended to by the physician to whom they are assigned, or they can be removed from the list by the clerk by clicking the 'delete' button next to the visit date - if, say, the patient has left without seeing the doctor or cancelled the appointment or something along those lines.

Patients who disappear from the list of pending patients in Figure 27 disappear because they have been attended to by the physician and their visit status has changed. Once the physician has made the initial diagnosis and ordered the tests to be performed for the patient, the patient's record is moved from the 'pending' list to the list of patients waiting for their results (Figure 28). When the system determines that all the results for that particular patient on that particular visit have come in, a button appears. Clicking this button allows the clerk to return the patient's to the doctor's inbox to await its final diagnosis.

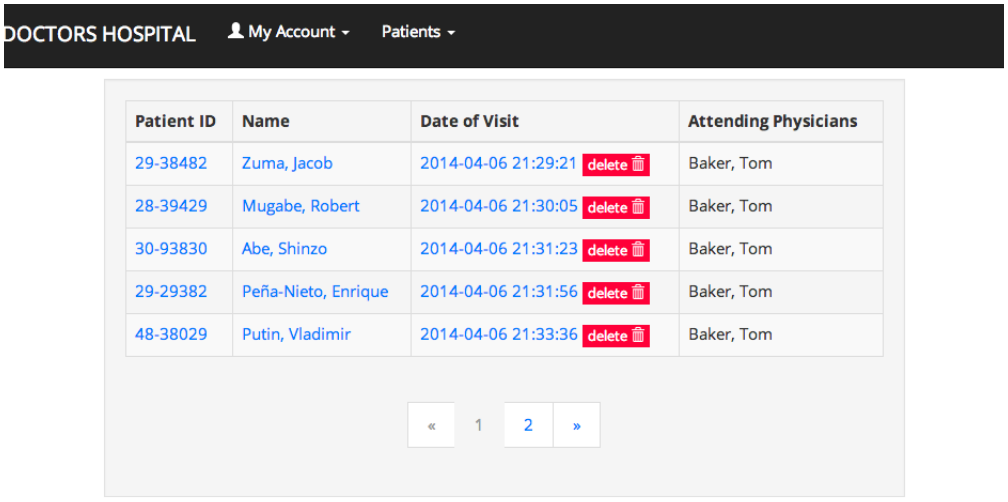

Figure 27: Patients pending

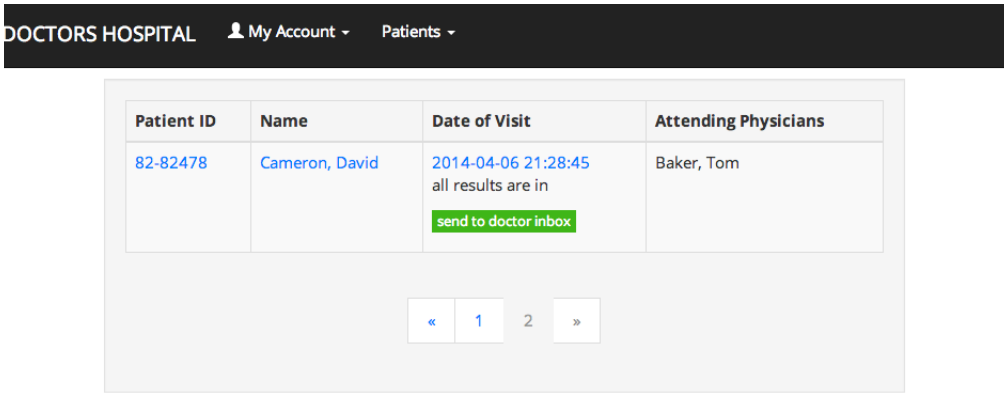

Figure 28: Patients waiting

On the doctor's side of things, Figure 29 is the Inbox page, containing the all the patients with visit records that have not been attended to. These entries appear in chronological order by date of visit, with the earliest patients in the front, but they can theoretically be accessed in any order. It is assumed that the doctor would attend

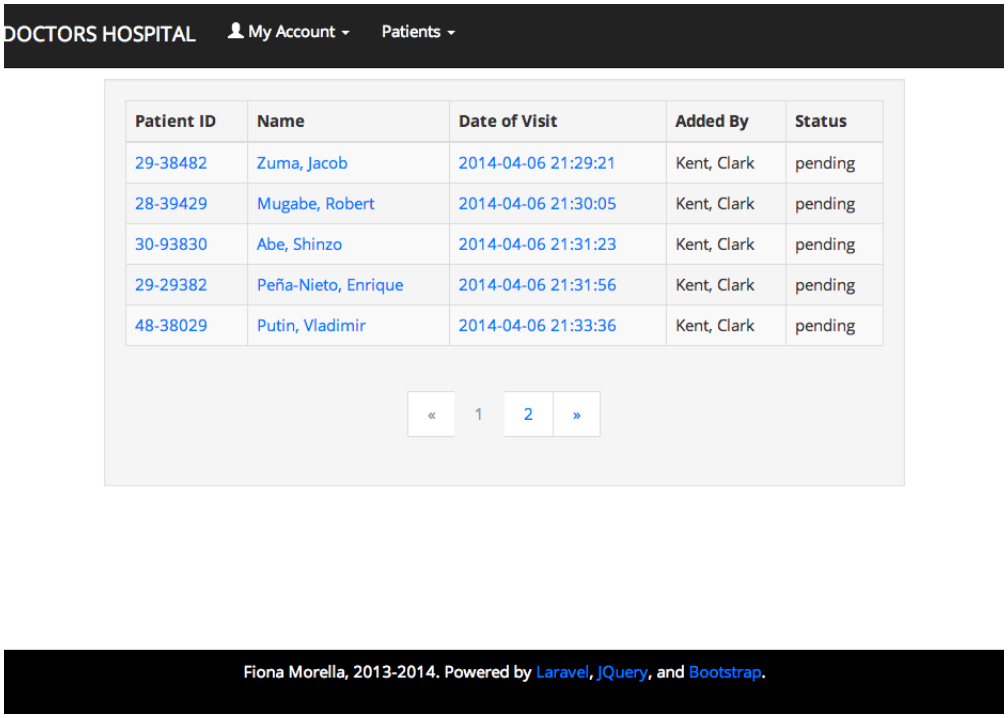

Figure 29: Patient inbox

to patients in chronological order.

Clicking on an entry leads to the page seen in Figure 30. It is from this page that the doctor can request medical procedures to be performed on a patient. Once the tests have been ordered, a table appears keeping track of which and how many tests have been ordered. Links to the results of said tests also appear when they have been uploaded to the system. When all the test results have arrived the ''final diagnosis' box appears, allowing the physician to add a final diagnosis and close that particular record.

In addition to an inbox for pending patients, doctors also have an inbox for patients waiting for diagnosis. When the button to send the record back to the doctor inbox is pressed in Figure 28, that visit appears in the inbox of patients waiting for diagnosis, where a doctor may click on each entry to return to the diagnose page (Figure 30) and input the final diagnosis.

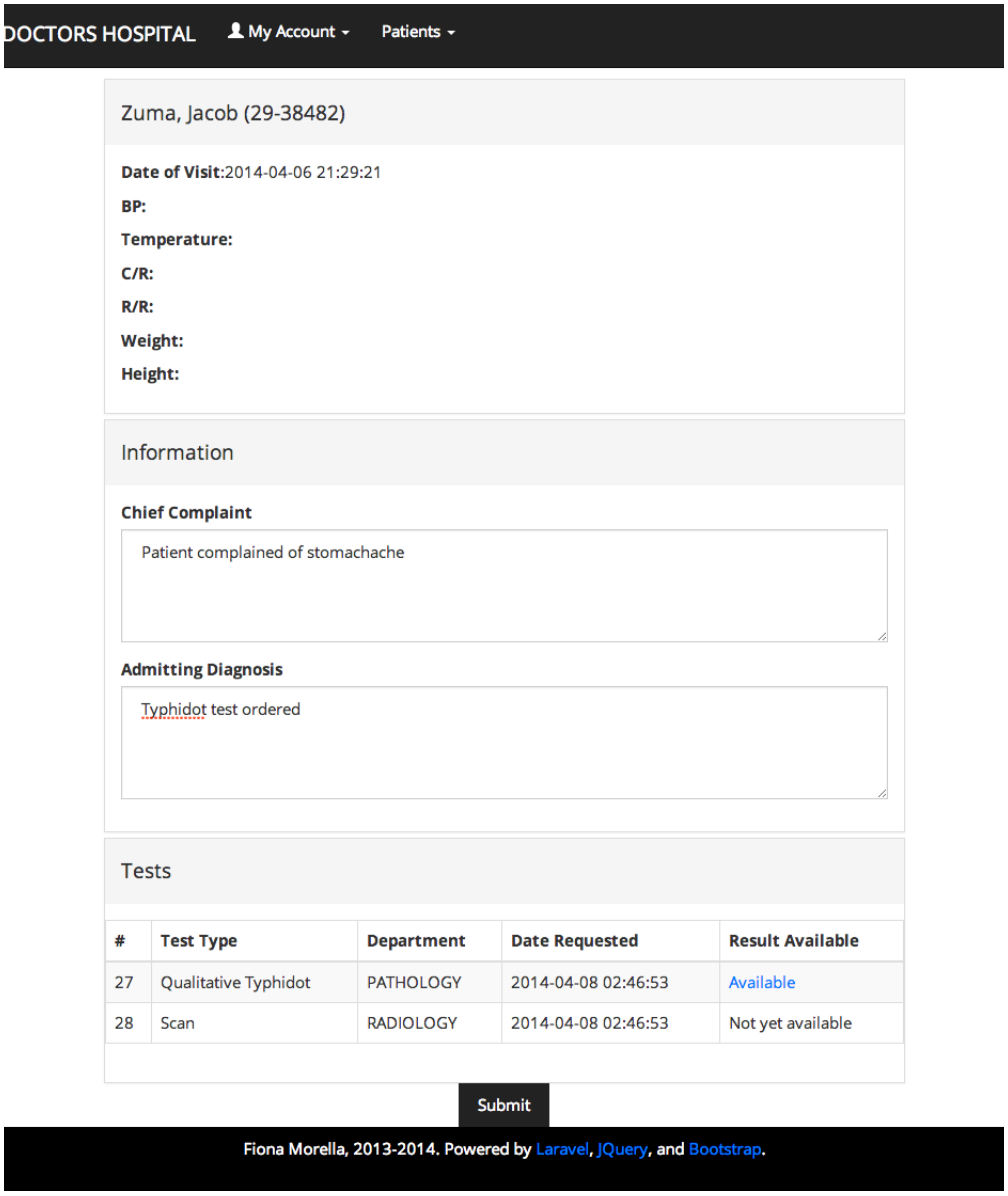

Figure 30: Diagnose

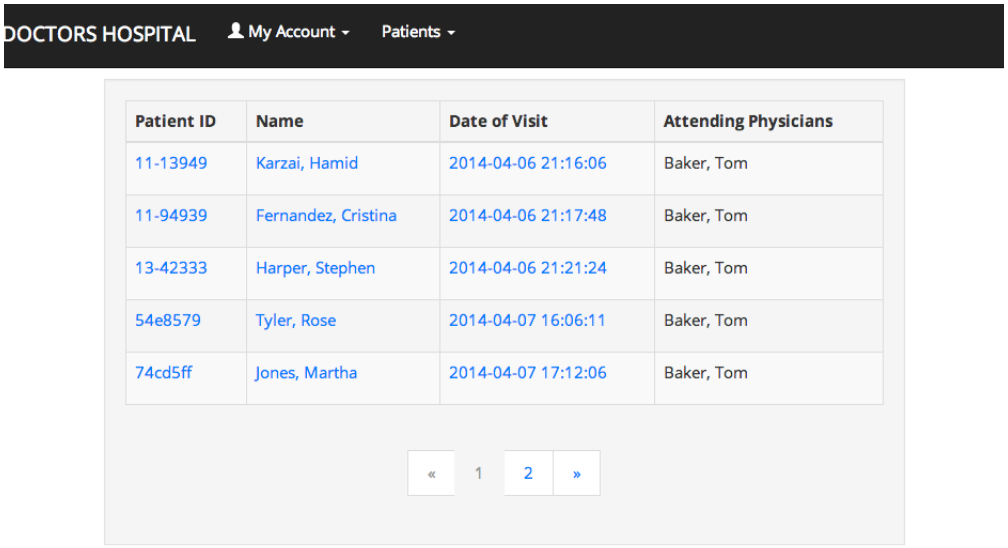

Figure 31: patient outbox

The patient outbox (Figure 31) is a page that exists for both doctors and clerks. The doctor's outbox includes all the completed patient visits – those with complete test results and final diagnoses - for which the doctor was an attending physician, while for clerks it contains all the completed patient visits that were added to the database by that clerk.

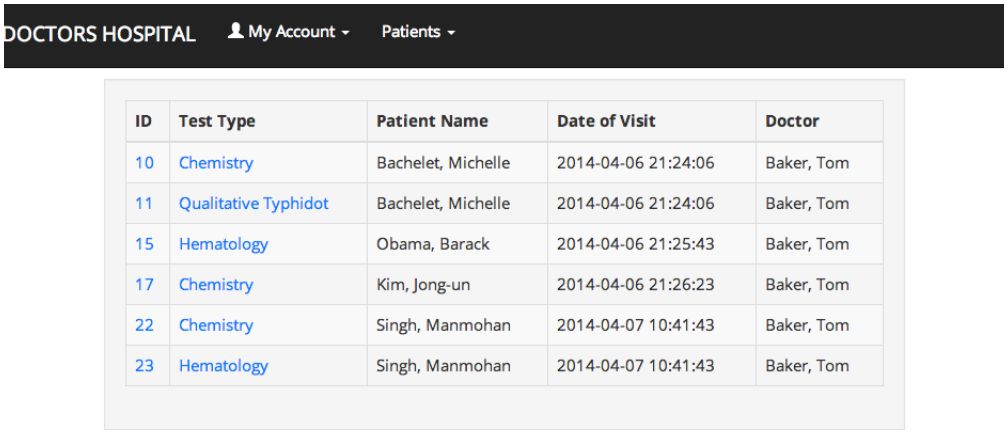

Figure 32: Test Request Inbox

The medical technician is also presented with an inbox (Figure 32, in which all

the pending test requests pertaining to that medical technician's department are displayed. When the data encoder has obtained the results corresponding to a certain test, they may then click on the entry in particular to begin the process of filling it in.

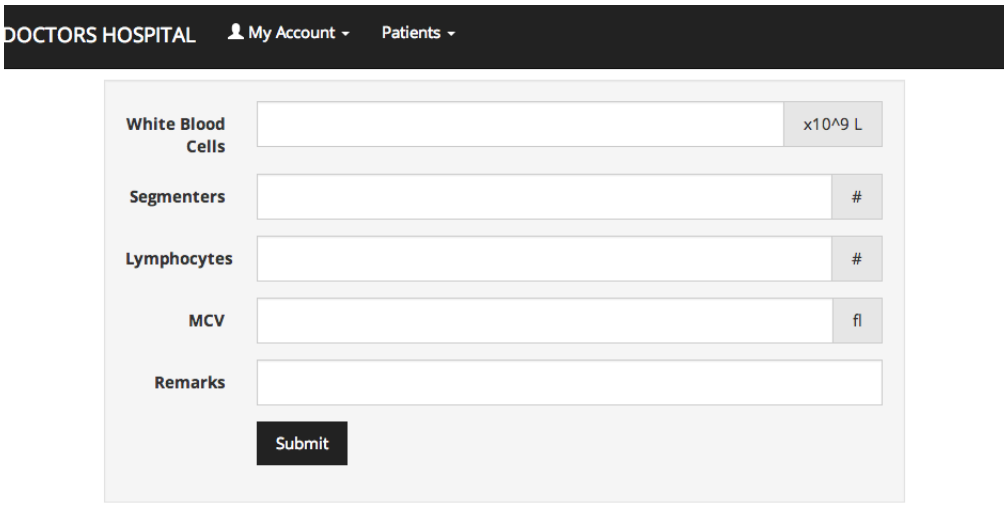

Figure 33: Add Test Result

This takes them to a page like Figure 33, in which they may enter all the data pertaining to the test. Once the form is accomplished, the page in Figure 35 or 36, containing all the relevant test results, is created. Naturally, input is checked for errors before it is passed to the database, and Figure **??** is an example of what warnings were to appear if fields were filled up improperly.

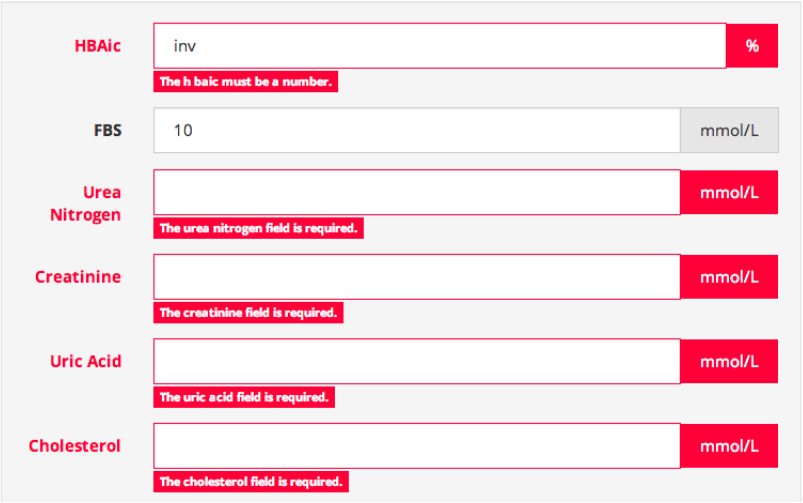

Figure 34: Error checking

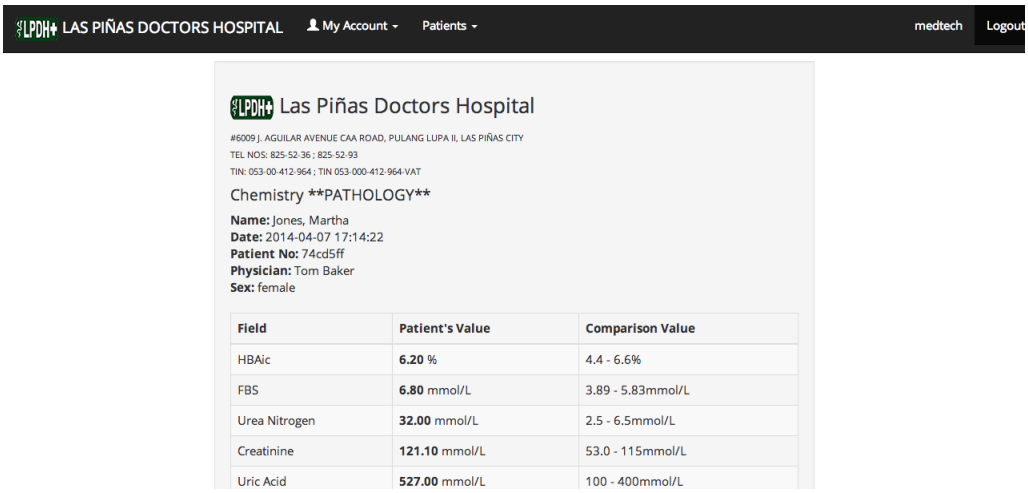

Figure 35: View Test Result

When a patient logs in to the system, they are presented with a page like Figure 37. There they can see the details of every visit of theirs to the hospital, including data like the attending physician, the diagnoses, and links to all the test results (Figure 33).

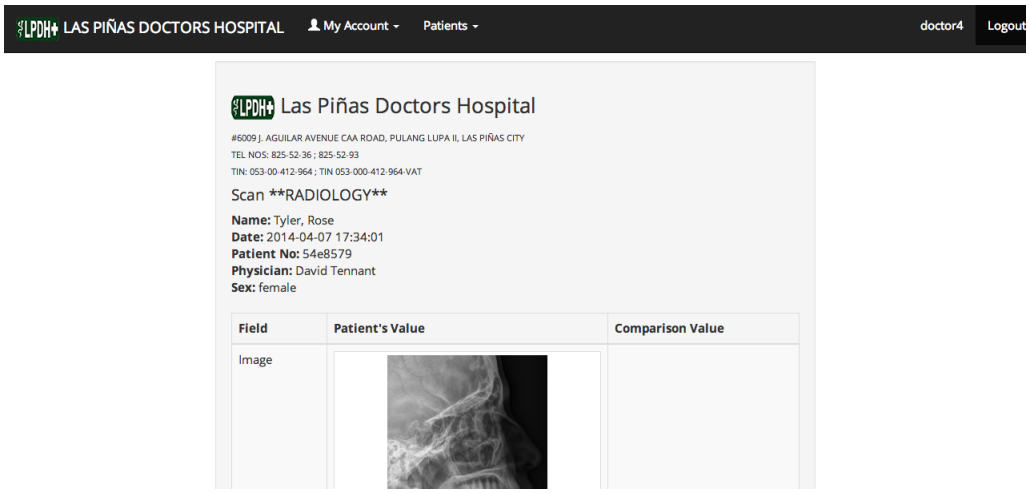

Figure 36: View Test Result: example of test with image field

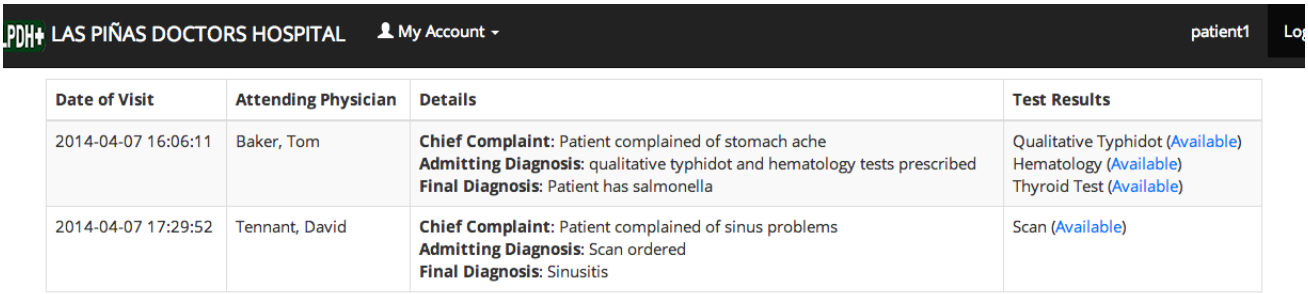

Figure 37: Patient's results

VII. Discussion

This hospital information system for Las Piñas Doctors Hospital is created for the benefit of the physicians, employees, and patients of said hospital. It improves the efficiency and productivity in output of the hospital and allows for greater capacity for patients and improved care. The system serves as a database for patient records, an improvement over the paper record system previously implemented.

There are six types of users in the system: the system administrators, hospital administrators, physicians, medical technicians, clerks, and patients. System administrates are responsible for the management of accounts, departments, and test tables. Hospital administrators have access to the records of patients and reports of system activity. The physicians use the system to view the records of patients to whom they are assigned, and using the system they may order medical tests and procedures for said patients. Medical technicians are responsible for adding the results of said medical procedures to the system, and lastly, patients may view the results of their medical tests and procedures once they become available. The objectives of the study detailed in the first chapter are fulfilled by these functions.

VIII. Conclusion

The goal of creating an information management system for Las Piñas Doctors Hospital has been met. The system improves upon the previous un-automated, paper-based system of the hospital to a more secure electronic database that is less vulnerable to human error and leads to speedier and more efficient clinical care.

IX. Recommendation

Although the targets for the system have been met, there are areas which may be improved upon. For instance, billing of accounts, a feature not currently implemented, could become part of the system in later improvements. Also modifications could be made to allow for sharing of information even to external or non-related entities such as other hospitals.

X. Bibliography

[1] Borzekowski, R. (2002). Health care finance and the early adoption of hospital information systems. Divisions of Research & Statistics and Monetary Affairs, Federal Reserve Board.

[2] Healthcare Information Management Systems Society. U.S. EMR Adption Model Trends.

[3] Craig, J. (2005). Introduction to the practice of telemedicine. Journal of Telemedicine and Telecare, 11(1).

[4] Telemedicine: opportunities and developments in member states. (2010). World Health Organization.

[5] Thrall, J. Teleradiology Part I. History and Clinical Applications. (2007). Radiology, 243(3), 613-617.

[6] Some U.S. hospitals outsourcing work: Shortage of radiologists spurs growing telemedicine trend. (2004). NBC News.

[7] Salazar, A. et al. Comparison Between Differently Priced Devices for Digital Capture of Films Using Computed Tomography as a Gold Standard: A Multireader Multicase Receiver Operating Characteristic Curve Study. Telemedicine Journal and e-Health, 17(4), 275-282.

[8] Salazar, et al. Comparison between different cost devices for digital capture of X-ray films: an image approach. Journal of Digital Imaging, 25(1), 91-100.

[9] Eysenbach, G. What is e-health? (2001). Journal of Medical Internet Research.,

3(2).

[10] Rosen, E. The death of telemedicine? (2000). Telemedicine Today, 8(1), 14-17.

[11] Oh, H et al. What is eHealth (3): A Systematic Review of Published Definitions. (2005). Journal of Medical Internet Research. 7(1).

[12] Pagliari, C. What is eHealth (4): A Scoping Exercise to Map the Field. (2005). Journal of Medical Internet Research. 7(1).

[13] Habib, J. EHRs, Meaningful Use, and a Model EMR. (2010). Drug Benefit Trends, 22(4).

[14] Eysenbach, G. Consumer health informatics. (2000). British Medical Journal, 320(7251).

[15] Ross, S & Lin, C. The Effects of Promoting Patient Access to Medical Records: A Review. (2003). Journal of the American Medical Informatics Association, 10(2), 129-138.

[16] Pagliari, C. Potential of electronic personal health records. (2007). British Medical Journal, 335(7815), 330-333.

[17] Silvestre, A et al. If You Build It, Will They Come? The Kaiser Permanente Model of Online Health Care. (2009). Health Affairs, 28(2) ,334-344.

[18] Weingart, et al. Who Uses the Patient Internet Portal? The PatientSite Experience. (2006). Journal of the American Medical Informatics Association, 13(1), 91-95.

[19] Emont, S. Measuring the Impact of Patient Portals: What the Literature Tells

Us. (2011). California Healthcare Foundation, Oakland, CA.

[20] Zickmund, et al. Interest in the use of computerised patient portals: role of the provider-patient relationship. (2007). Journal of General Internal Medicine, 23(Suppl 1).

[21] Greenhaigh, et al. Patients' attitudes to the summary care record and HealthSpace: qualitative study. (2008). British Medical Journal, 336:1290.

[22] Guardian Government Computing. NHS axes HealthSpace: 'Just too difficult' to use. (2012). The Register.

[23] Adoption, non-adoption, and abandonment of a personal electronic health record: case study of HealthSpace. (2010). British Medical Journal. 41:c5814.

[24] Habib, J. EHRs, Meaningful Use, and a Model EMR. (2010). Drug Benefit Trends, 22(4).

[25] Many US Consumers Want Major Changes in Health Care Design, Delivery. (2008). Deloitte.

[26] Leff, A. and Rayfield, J.Web-Application Development Using the Model/View/-Controller Design Pattern. (2001). Enterprise Distributed Object Computing Conference.

[27] Laravel Documentation. Retrieved from http://laravel.com/docs

XI. Appendix

A. Source Code

Listing 1: ../portal/bower.json

```
{
  "name": "portal",
  "version": "0.0.0",
  "authors": [
    "Fiona <fiona.morella@gmail.com>"
  ],
  "license": "MIT",
  "private": true,
  "ignore": [
    "**/.*",
    "node_modules",
    "bower_components",
    "public/components",
    "test",
    "tests"
  ],
  "dependencies": {
    "chosen": "~1.0.0",
  "jquery": "1.9.1",
    "eonasdan-bootstrap-datetimepicker": "~2.1.11",
    "jquery-ui": "1.10.3"
  }
}
```

Listing 2: ../portal/composer.json

```
{
  "name": "laravel/laravel",
  "description": "The Laravel Framework.",
  "keywords": ["framework", "laravel"],
  "license": "MIT",

  "require": {
    "laravel/framework": "4.0.*",
    "cartalyst/sentry": "2.0.*",
      "codesleeve/stapler": "dev-master"

    },
  "autoload": {
    "classmap": [
      "app/commands",
      "app/controllers",
      "app/models",
      "app/database/migrations",
      "app/database/seeds",
```

```
20        "app/tests/TestCase.php"
21      ]
22    },
23
24        "require-dev": {
25            "phpdocumentor/phpdocumentor": "2.*"
26        },
27
28    "scripts": {
29      "post-install-cmd": [
30        "php artisan optimize"
31      ],
32      "post-update-cmd": [
33        "php artisan clear-compiled",
34        "php artisan optimize"
35      ],
36      "post-create-project-cmd": [
37        "php artisan key:generate"
38      ]
39    },
40    "config": {
41      "preferred-install": "dist"
42    },
43    "minimum-stability": "stable"
44  }
```

Listing 3: ../portal/Gruntfile.js

```
1   module.exports = function(grunt) {
2
3     // Project configuration.
4     grunt.initConfig({
5       pkg: grunt.file.readJSON('package.json'),
6     //compile less file and minify resulting css
7     less: {
8           development: {
9             options: {
10              compress: true,
11          cleancss: true,
12   //         yuicompress: true,
13   //         optimization: 2
14            },
15            files: {
16              "public/css/main.min.css": "public/less/main.less"
17            }
18          }
19        },
20      //concatenate all javascript files into main.js
21    concat: {
22      js: {
23        src:[
24          "public/components/jquery/jquery.js",
25          "public/components/moment/moment.js",
26          "public/components/bootstrap/dist/js/bootstrap.js",
27          "public/components/eonasdan-bootstrap-datetimepicker/src/js/bootstrap-datetimepicker.js",
```

```
28          "public/components/eonasdan-bootstrap-datetimepicker/src/js/locales/en.js",
29          "public/components/jquery-ui/ui/jquery-ui.js",
30          "public/components/select2/select2.js",
31          "public/components/jquery/jquery-migrate.js",
32          "public/components/fancybox/source/jquery.fancybox.js",
33          "public/components/tagmanager/tagmanager.js"
34
35          ],
36        dest: "public/js/main.js"
37      }
38    },
39    //minify main.js -> main.min.js
40    uglify:{
41      dest: {
42        files:{
43          "public/js/main.min.js" : "public/js/main.js"
44          }
45        }
46      }
47
48    });
49
50    grunt.loadNpmTasks('grunt-contrib-less');
51    grunt.loadNpmTasks('grunt-contrib-concat');
52    grunt.loadNpmTasks('grunt-contrib-uglify');
53
54
55    // Default task(s).
56    grunt.registerTask('default', ['less','concat','uglify']);
57
58  };
```

Listing 4: ../portal/package.json

```
1  {
2    "name": "portal",
3    "version": "0.0.0",
4    "main": "index.js",
5    "devDependencies": {
6      "grunt": "~0.4.2",
7      "grunt-contrib-jshint": "~0.6.3",
8      "grunt-contrib-nodeunit": "~0.2.0",
9      "grunt-contrib-uglify": "~0.2.7",
10     "grunt-contrib-less": "~0.8.3",
11     "grunt-contrib-cssmin": "~0.7.0",
12     "grunt-contrib-sass": "~0.6.0",
13     "grunt-contrib-compass": "~0.7.0",
14     "grunt-contrib-concat": "~0.3.0"
15   }
16 }
```

Listing 5: ../portal/server.php

```
1  <?php
```

```
2
3  $uri = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);
4
5  $uri = urldecode($uri);
6
7  $paths = require __DIR__.'/bootstrap/paths.php';
8
9  $requested = $paths['public'].$uri;
10
11 // This file allows us to emulate Apache's "mod_rewrite" functionality from the
12 // built-in PHP web server. This provides a convenient way to test a Laravel
13 // application without having installed a "real" web server software here.
14 if ($uri !== '/' and file_exists($requested))
15 {
16   return false;
17 }
18
19 require_once $paths['public'].'/index.php';
```

Listing 6: ../portal/app/helpers.php

```
1  <?php
2  /**
3   * Various extensions of the Laravel framework to help make my life easier
4   *
5   *
6   */
7  require 'helpers/ErrorProvider.php';
8  require 'helpers/ArrayPaginationBootstrapPresenter.php';
9  require 'helpers/macros.php';
10 require 'helpers/composers.php';
11 require 'helpers/ColumnPresenter.php';
12 require 'helpers/relativetime.php';
13 //require '../vendor/nanodocumet/Nanodicom/nanodicom.php';
```

Listing 7: ../portal/app/routes.php

```
1  <?php
2
3  /*
4  |--------------------------------------------------------------------------
5  | Application Routes
6  |--------------------------------------------------------------------------
7  |
8  | Here is where you can register all of the routes for an application.
9  | It's a breeze. Simply tell Laravel the URIs it should respond to
10 | and give it the Closure to execute when that URI is requested.
11 |
12 */
13
14 //Session routes
15
16 Route::get  ('profile', array('as'=>'profile','uses'=>'UserController@profile'));
17 Route::get    ('login',  array('as' => 'login', 'uses' => 'SessionController@create'));
```

```
18  Route::post   ('login', array('as' => 'login.post','uses' => 'SessionController@store'));
19  Route::get    ('logout', array('as' => 'logout', 'uses' => 'SessionController@destroy'));
20  Route::resource ('sessions','SessionController',array('only' => array('create', 'store', 'destroy')));
21
22  //User routes
23  Route::post   ('created',array('as'=>'created','uses'=>'UserController@created'));
24  Route::get    ('resend', array('as'=>'resend','uses'=>'UserController@requestActivation'));
25  Route::post   ('resend',array('as'=>'resend.post','uses'=>'UserController@sendActivation'));
26  Route::get    ('activate/{id}/{code}',array('as'=>'activate','uses'=>'UserController@activate'));
27  Route::get    ('forgot',array("before"=>"guest","as"=>"forgot","uses"=>"UserController@requestReset"));
28  Route::post   ('forgot',array('as'=>'forgot.post','uses'=>'UserController@sendReset'));
29  Route::get    ('reset/{id}/{code}',array('as'=>'reset','uses'=>'UserController@reset'));
30  Route::post   ('reset',array("as"=>"reset.post","uses"=>"UserController@resetPassword"));
31  Route::get    ('settings',array('before'=>'auth','as'=>'settings','uses'=>'UserController@settings'));
32  Route::post   ('users/changePassword',array('as'=>'users.changePassword','uses'=>'UserController@changePassword')
        );
33  Route::post   ('users/changeEmail',array('as'=>'users.changeEmail','uses'=>'UserController@changeEmail'));
34  Route::get    ('confirm/{id}/{code}',array('as'=>'confirm','uses'=>'UserController@confirmChangeEmail'));
35  Route::post ('users/disable/{id}',array('as'=>'users.disable','uses'=>'UserController@disable'));
36  Route::resource ('users','UserController',array('only' => array('index','show','update','destroy','edit','create'
        ,'store')));
37
38  //Patient routes
39  Route::get    ('outbox',array('as'=>'outbox','uses'=>'PatientController@outbox'));
40  Route::get    ('outbox_doctor',array('as'=>'outbox_doctor','uses'=>'PatientController@outbox_doctor'));
41  Route::get    ('waiting',array('as'=>'waiting','uses'=>'PatientController@waiting'));
42  Route::get    ('waiting_clerk', array('as'=>'waiting_clerk','uses'=>'PatientController@waiting_clerk'));
43
44  Route::get('waiting_for_diagnosis',array('as'=>'waiting_for_diagnosis','uses'=>'PatientController@waiting2'));
45
46
47  Route::get('pending',array('as'=>'pending','uses'=>'PatientController@pending'));
48  Route::get    ('inbox',array('as'=>'inbox','uses'=>'PatientController@inbox'));
49
50  Route::get    ('patients/search',array(
51            "before" => "auth_search_patient",
52            'uses' => 'PatientController@search',
53            "as" => "patients.search"
54          ));
55  Route::post   ('patients/redirect',array(
56            'as' => 'patients.redirect',
57            'uses' => 'PatientController@redirectById'
58          ));
59  Route::get    ('patients/active',array('as'=>'patients.active','uses'=> 'PatientController@active'));
60  Route::resource ('patients','PatientController');
61
62  //Visit routes
63
64  Route::get    ('mine',array('as'=>'mine','uses'=>'VisitController@mine'));
65  Route::any    ('diagnose/{visits}',array('as'=>'diagnose','uses'=>'VisitController@diagnose'));
66  Route::post   ('visits/status/{visits}',array('as'=>'visits.status','uses'=>'VisitController@status'));
67  Route::get    ('visits/create/{patients}',array('as'=>'visits.create','uses'=>'VisitController@create'));
68  Route::post   ('visits/create',array('as'=>'visits.create.post','uses'=>'VisitController@store'));
69  Route::resource ('visits','VisitController',array('only'=>array('show','edit','update','destroy')));
70
71  //Doctor routes
```

```
72  Route::resource ('doctors','DoctorController',array('only'=>array('index','show')));
73
74  Route::post('departments/disable/{id}',array('as'=>'departments.disable','uses'=>'DepartmentController@disable'))
        ;
75  Route::resource('departments','DepartmentController',array('except'=>array('edit','create')));
76  Route::get('types/values/{types}/{name}',array('as'=>'types.values','uses'=>'TestTypeController@values'));
77  Route::get('types/dropdown/{departments}',array('as'=>'types.dropdown','uses'=>'TestTypeController@dropdown'));
78  Route::resource('types','TestTypeController',array('except'=>array('edit','update')));
79
80  Route::get('requests/inbox',array('as'=>'requests.inbox','uses'=>'TestRequestController@inbox'));
81  Route::get('requests/outbox',array('as'=>'requests.outbox','uses'=>'TestRequestController@outbox'));
82
83  Route::resource('requests','TestRequestController',array('except'=>array('create','store','edit','update')));
84
85  Route::get ('results/create/{requests}',array('as'=>'results.create','uses'=>'TestResultController@create'));
86
87  Route::any('results/search',array('as'=>'results.search','uses'=>'TestResultController@search'));
88  Route::resource('results','TestResultController',array('except'=>array('create','index')));
89
90  Route::any('landing',function(){
91    return View::make('landing');
92  });
93
94
95
96  Route::any('report',function(){
97    if(Input::get()){
98      $results = null;
99      switch(Input::get('period'))
100     {
101
102       case "year":
103       $start = new DateTime( Input::get('year')."-01-01");
104       $end = new DateTime((Input::get('year')+1)."-01-01");
105       break;
106       case "month":
107       $start = new DateTime( Input::get('year')."-".Input::get('month')."-01");
108       $end = new DateTime( Input::get('year')."-".Input::get('month')."-01");
109       $end = $end->add(new DateInterval('P1M'));
110
111       break;
112       case "week":
113       $start = new DateTime( Input::get('year')."-".Input::get('month')."-".Input::get('day'));
114       $end = new DateTime( Input::get('year')."-".Input::get('month')."-".Input::get('day'));
115       $end = $end->add(new DateInterval('P1W'));
116
117       break;
118
119       case "day":
120       $start = new DateTime( Input::get('year')."-".Input::get('month')."-".Input::get('day'));
121       $end = new DateTime( Input::get('year')."-".Input::get('month')."-".Input::get('day'));
122       $end = $end->add(new DateInterval('P1D'));
123
124       break;
125     }
126     $results = TestResult::whereBetween('created_at',array($start,$end))->get();
```

```
127     $v = Visit::whereBetween('date_of_visit',array($start,$end));
128     $distinct = $v->select('patient_id')->distinct()->count();
129     $numvisits= $v->select('patient_id')->get()->count();
130     //var_dump($count);
131     $visits = $v->get();
132
133     return View::make('results.report')->with('results',$results)->with('visits',$visits)->with('distinct',
            $distinct)->with('numvisits',$numvisits);
134   }
135   return View::make('results.report');
136 });
```

Listing 8: ../portal/app/filters.php

```
1  <?php
2
3  /*
4  |--------------------------------------------------------------------------
5  | Application & Route Filters
6  |--------------------------------------------------------------------------
7  |
8  | Below you will find the "before" and "after" events for the application
9  | which may be used to do any work before or after a request into your
10 | application. Here you may also register your custom route filters.
11 |
12 */
13
14 App::before(function($request)
15 {
16   //
17 });
18
19
20 App::after(function($request, $response)
21 {
22   //
23 });
24
25 /*
26 |--------------------------------------------------------------------------
27 | Authentication Filters
28 |--------------------------------------------------------------------------
29 |
30 | The following filters are used to verify that the user of the current
31 | session is logged into this application. The "basic" filter easily
32 | integrates HTTP Basic authentication for quick, simple checking.
33 |
34 */
35
36 Route::filter('auth', function()
37 {
38   if (!Sentry::check()) return Redirect::route('login');
39 });
40
41 Route::filter('admin',function()
```

```
42  {
43
44    try{
45      $user = Sentry::getUser();
46      $group = Sentry::findGroupByName('admin');
47
48      if(!$user->inGroup($group))
49      {
50        App::abort(403);
51      }
52    }
53    catch (Cartalyst\Sentry\Users\UserNotFoundException $e)
54    {
55      App:abort(403);
56    }
57
58  });
59
60  /**
61   * reference filter as hasAccess:[permission]
62   *
63   */
64  Route::filter('hasAccess',function($route, $request, $value){
65
66
67
68      if (!Sentry::check()) return Redirect::route('login');
69
70
71    //Check if someone without access is attempting to view their own record.
72    //Normally, all these routes should be closed to those without access.
73    //However, if the user is trying to view their own record
74    //Then they should be allowed
75
76    if($param = $route->getParameter('users'))
77    {
78      $id = Sentry::getUser()->id;
79    }
80    elseif ($param = $route->getParameter('patients'))
81    {
82      if($patient = Sentry::getUser()->patient)
83      {
84        $id = $patient->id;
85      }
86    }
87    elseif ($param = $route->getParameter('visits'))
88    {
89      if($patient = Sentry::getUser()->patient)
90      {
91        $id = $patient->id;
92      }
93      $param = Visit::find($param)->patient_id;
94    }
95
96    try{
97      $user = Sentry::getUser();
```

```
98      if((!$user->hasAccess($value)) && $id != $param){
99        //forbidden
100       App::abort(403);
101     }
102   }
103   catch (Cartalyst\Sentry\Users\UserNotFoundException $e)
104      {
105      App::abort(403);
106      }
107
108
109
110
111 });
112
113 /**
114  * Reference filter as inGroup:[group]
115  *
116  *
117  */
118 Route::filter('inGroup',function($route,$request,$value){
119
120     if (!Sentry::check()) return Redirect::route('login');
121
122   try{
123     $user = Sentry::getUser();
124     $group = Sentry::findGroupByName($value);
125
126     if((!$user->inGroup($group)))
127     {
128       App::abort(403);
129     }
130   }
131   catch (Cartalyst\Sentry\Users\UserNotFoundException $e)
132   {
133     App:abort(403);
134   }
135
136 });
137
138 Route::filter('showResults',function($route,$request){
139   $testId = $route->getParameter('results');
140   $test = TestResult::find($testId);
141
142   $user = Sentry::getUser();
143   $permission = false;
144
145   //var_dump($test);
146
147
148   if($user->inGroup(Sentry::findGroupByName('patients')))
149   {
150     if($test->request->visit->patient->user->id==$user->id)
151     {
152       $permission = true;
153     }
```

```
154     }
155
156     if($user->inGroup(Sentry::findGroupByName('admin')))
157     {
158       $permission = true;
159     }
160     if($user->inGroup(Sentry::findGroupByName('medical_technicians')))
161     {
162       $permission = true;
163     }
164     if($user->inGroup(Sentry::findGroupByName('hospital_admin')))
165     {
166       $permission = true;
167     }
168     if($user->inGroup(Sentry::findGroupByName('doctors')))
169     {
170       $permission = true;
171     }
172
173     if(!$permission){
174       App::abort(403);
175     }
176   });
177
178
179   /*
180   |--------------------------------------------------------------------------
181   | Guest Filter
182   |--------------------------------------------------------------------------
183   |
184   | The "guest" filter is the counterpart of the authentication filters as
185   | it simply checks that the current user is not logged in. A redirect
186   | response will be issued if they are, which you may freely change.
187   |
188   */
189
190   Route::filter('guest', function()
191   {
192     if (Sentry::check()) return Redirect::to('login');
193   });
194
195   /*
196   |--------------------------------------------------------------------------
197   | CSRF Protection Filter
198   |--------------------------------------------------------------------------
199   |
200   | The CSRF filter is responsible for protecting your application against
201   | cross-site request forgery attacks. If this special token in a user
202   | session does not match the one given in this request, we'll bail.
203   |
204   */
205
206   Route::filter('csrf', function()
207   {
208     if (Session::token() != Input::get('_token'))
209     {
```

```
210       throw new Illuminate\Session\TokenMismatchException;
211     }
212   });
```

Listing 9: ../portal/app/config/app.php

```
1   <?php
2
3   return array(
4
5     /*
6     |--------------------------------------------------------------------------
7     | Application Debug Mode
8     |--------------------------------------------------------------------------
9     |
10    | When your application is in debug mode, detailed error messages with
11    | stack traces will be shown on every error that occurs within your
12    | application. If disabled, a simple generic error page is shown.
13    |
14    */
15
16    'debug' => true,
17
18    /*
19    |--------------------------------------------------------------------------
20    | Application URL
21    |--------------------------------------------------------------------------
22    |
23    | This URL is used by the console to properly generate URLs when using
24    | the Artisan command line tool. You should set this to the root of
25    | your application so that it is used when running Artisan tasks.
26    |
27    */
28
29    'url' => 'http://localhost',
30
31    /*
32    |--------------------------------------------------------------------------
33    | Application Timezone
34    |--------------------------------------------------------------------------
35    |
36    | Here you may specify the default timezone for your application, which
37    | will be used by the PHP date and date-time functions. We have gone
38    | ahead and set this to a sensible default for you out of the box.
39    |
40    */
41
42    'timezone' => 'Asia/Singapore',
43
44    /*
45    |--------------------------------------------------------------------------
46    | Application Locale Configuration
47    |--------------------------------------------------------------------------
48    |
49    | The application locale determines the default locale that will be used
```

```
50      | by the translation service provider. You are free to set this value
51      | to any of the locales which will be supported by the application.
52      |
53      */
54
55      'locale' => 'en',
56
57      /*
58      |--------------------------------------------------------------------------
59      | Encryption Key
60      |--------------------------------------------------------------------------
61      |
62      | This key is used by the Illuminate encrypter service and should be set
63      | to a random, 32 character string, otherwise these encrypted strings
64      | will not be safe. Please do this before deploying an application!
65      |
66      */
67
68      'key' => 'RKYOBxZfVml41z8310ZhaghHuCRRJGjL',
69
70      /*
71      |--------------------------------------------------------------------------
72      | Autoloaded Service Providers
73      |--------------------------------------------------------------------------
74      |
75      | The service providers listed here will be automatically loaded on the
76      | request to your application. Feel free to add your own services to
77      | this array to grant expanded functionality to your applications.
78      |
79      */
80
81      'providers' => array(
82
83        'Illuminate\Foundation\Providers\ArtisanServiceProvider',
84        'Illuminate\Auth\AuthServiceProvider',
85        'Illuminate\Cache\CacheServiceProvider',
86        'Illuminate\Foundation\Providers\CommandCreatorServiceProvider',
87        'Illuminate\Session\CommandsServiceProvider',
88        'Illuminate\Foundation\Providers\ComposerServiceProvider',
89        'Illuminate\Routing\ControllerServiceProvider',
90        'Illuminate\Cookie\CookieServiceProvider',
91        'Illuminate\Database\DatabaseServiceProvider',
92        'Illuminate\Encryption\EncryptionServiceProvider',
93        'Illuminate\Filesystem\FilesystemServiceProvider',
94        'Illuminate\Hashing\HashServiceProvider',
95        'Illuminate\Html\HtmlServiceProvider',
96        'Illuminate\Foundation\Providers\KeyGeneratorServiceProvider',
97        'Illuminate\Log\LogServiceProvider',
98        'Illuminate\Mail\MailServiceProvider',
99        'Illuminate\Foundation\Providers\MaintenanceServiceProvider',
100       'Illuminate\Database\MigrationServiceProvider',
101       'Illuminate\Foundation\Providers\OptimizeServiceProvider',
102       'Illuminate\Pagination\PaginationServiceProvider',
103       'Illuminate\Foundation\Providers\PublisherServiceProvider',
104       'Illuminate\Queue\QueueServiceProvider',
105       'Illuminate\Redis\RedisServiceProvider',
```

```
    'Illuminate\Auth\Reminders\ReminderServiceProvider',
    'Illuminate\Foundation\Providers\RouteListServiceProvider',
    'Illuminate\Database\SeedServiceProvider',
    'Illuminate\Foundation\Providers\ServerServiceProvider',
    'Illuminate\Session\SessionServiceProvider',
    'Illuminate\Foundation\Providers\TinkerServiceProvider',
    'Illuminate\Translation\TranslationServiceProvider',
    'Illuminate\Validation\ValidationServiceProvider',
    'Illuminate\View\ViewServiceProvider',
    'Illuminate\Workbench\WorkbenchServiceProvider',
    'Cartalyst\Sentry\SentryServiceProvider',
    'Codesleeve\Stapler\StaplerServiceProvider'

  ),

  /*
  |--------------------------------------------------------------------------
  | Service Provider Manifest
  |--------------------------------------------------------------------------
  |
  | The service provider manifest is used by Laravel to lazy load service
  | providers which are not needed for each request, as well to keep a
  | list of all of the services. Here, you may set its storage spot.
  |
  */

  'manifest' => storage_path().'/meta',

  /*
  |--------------------------------------------------------------------------
  | Class Aliases
  |--------------------------------------------------------------------------
  |
  | This array of class aliases will be registered when this application
  | is started. However, feel free to register as many as you wish as
  | the aliases are "lazy" loaded so they don't hinder performance.
  |
  */

  'aliases' => array(

    'App'             => 'Illuminate\Support\Facades\App',
    'Artisan'         => 'Illuminate\Support\Facades\Artisan',
    'Auth'            => 'Illuminate\Support\Facades\Auth',
    'Blade'           => 'Illuminate\Support\Facades\Blade',
    'Cache'           => 'Illuminate\Support\Facades\Cache',
    'ClassLoader'     => 'Illuminate\Support\ClassLoader',
    'Config'          => 'Illuminate\Support\Facades\Config',
    'Controller'      => 'Illuminate\Routing\Controllers\Controller',
    'Cookie'          => 'Illuminate\Support\Facades\Cookie',
    'Crypt'           => 'Illuminate\Support\Facades\Crypt',
    'DB'              => 'Illuminate\Support\Facades\DB',
    'Eloquent'        => 'Illuminate\Database\Eloquent\Model',
    'Event'           => 'Illuminate\Support\Facades\Event',
    'File'            => 'Illuminate\Support\Facades\File',
    'Form'            => 'Illuminate\Support\Facades\Form',
```

```
    'Hash'            => 'Illuminate\Support\Facades\Hash',
    'HTML'            => 'Illuminate\Support\Facades\HTML',
    'Input'           => 'Illuminate\Support\Facades\Input',
    'Lang'            => 'Illuminate\Support\Facades\Lang',
    'Log'             => 'Illuminate\Support\Facades\Log',
    'Mail'            => 'Illuminate\Support\Facades\Mail',
    'Paginator'       => 'Illuminate\Support\Facades\Paginator',
    'Password'        => 'Illuminate\Support\Facades\Password',
    'Queue'           => 'Illuminate\Support\Facades\Queue',
    'Redirect'        => 'Illuminate\Support\Facades\Redirect',
    'Redis'           => 'Illuminate\Support\Facades\Redis',
    'Request'         => 'Illuminate\Support\Facades\Request',
    'Response'        => 'Illuminate\Support\Facades\Response',
    'Route'           => 'Illuminate\Support\Facades\Route',
    'Schema'          => 'Illuminate\Support\Facades\Schema',
    'Seeder'          => 'Illuminate\Database\Seeder',
    'Session'         => 'Illuminate\Support\Facades\Session',
    'Str'             => 'Illuminate\Support\Str',
    'URL'             => 'Illuminate\Support\Facades\URL',
    'Validator'       => 'Illuminate\Support\Facades\Validator',
    'View'            => 'Illuminate\Support\Facades\View',
    'Sentry'      => 'Cartalyst\Sentry\Facades\Laravel\Sentry',

  ),

);
```

Listing 10: ../portal/app/config/auth.php

```
<?php

return array(

  /*
  |--------------------------------------------------------------------------
  | Default Authentication Driver
  |--------------------------------------------------------------------------
  |
  | This option controls the authentication driver that will be utilized.
  | This drivers manages the retrieval and authentication of the users
  | attempting to get access to protected areas of your application.
  |
  | Supported: "database", "eloquent"
  |
  */

  'driver' => 'eloquent',

  /*
  |--------------------------------------------------------------------------
  | Authentication Model
  |--------------------------------------------------------------------------
  |
  | When using the "Eloquent" authentication driver, we need to know which
  | Eloquent model should be used to retrieve your users. Of course, it
```

```
27      | is often just the "User" model but you may use whatever you like.
28      |
29      */
30
31      'model' => 'User',
32
33      /*
34      |--------------------------------------------------------------------------
35      | Authentication Table
36      |--------------------------------------------------------------------------
37      |
38      | When using the "Database" authentication driver, we need to know which
39      | table should be used to retrieve your users. We have chosen a basic
40      | default value but you may easily change it to any table you like.
41      |
42      */
43
44      'table' => 'users',
45
46
47      'username' => 'email',
48      'password' => 'password',
49
50
51      /*
52      |--------------------------------------------------------------------------
53      | Password Reminder Settings
54      |--------------------------------------------------------------------------
55      |
56      | Here you may set the settings for password reminders, including a view
57      | that should be used as your password reminder e-mail. You will also
58      | be able to set the name of the table that holds the reset tokens.
59      |
60      | The "expire" time is the number of minutes that the reminder should be
61      | considered valid. This security feature keeps tokens short-lived so
62      | they have less time to be guessed. You may change this as needed.
63      |
64      */
65
66      'reminder' => array(
67
68        'email' => 'emails.auth.reminder',
69
70        'table' => 'password_reminders',
71
72        'expire' => 60,
73
74      ),
75
76  );
```

Listing 11: ../portal/app/config/cache.php

```
1   <?php
2
```

```
3  return array(
4
5      /*
6      |--------------------------------------------------------------------------
7      | Default Cache Driver
8      |--------------------------------------------------------------------------
9      |
10     | This option controls the default cache "driver" that will be used when
11     | using the Caching library. Of course, you may use other drivers any
12     | time you wish. This is the default when another is not specified.
13     |
14     | Supported: "file", "database", "apc", "memcached", "redis", "array"
15     |
16     */
17
18     'driver' => 'file',
19
20     /*
21     |--------------------------------------------------------------------------
22     | File Cache Location
23     |--------------------------------------------------------------------------
24     |
25     | When using the "file" cache driver, we need a location where the cache
26     | files may be stored. A sensible default has been specified, but you
27     | are free to change it to any other place on disk that you desire.
28     |
29     */
30
31     'path' => storage_path().'/cache',
32
33     /*
34     |--------------------------------------------------------------------------
35     | Database Cache Connection
36     |--------------------------------------------------------------------------
37     |
38     | When using the "database" cache driver you may specify the connection
39     | that should be used to store the cached items. When this option is
40     | null the default database connection will be utilized for cache.
41     |
42     */
43
44     'connection' => null,
45
46     /*
47     |--------------------------------------------------------------------------
48     | Database Cache Table
49     |--------------------------------------------------------------------------
50     |
51     | When using the "database" cache driver we need to know the table that
52     | should be used to store the cached items. A default table name has
53     | been provided but you're free to change it however you deem fit.
54     |
55     */
56
57     'table' => 'cache',
58
```

```
59    /*
60    |--------------------------------------------------------------------------
61    | Memcached Servers
62    |--------------------------------------------------------------------------
63    |
64    | Now you may specify an array of your Memcached servers that should be
65    | used when utilizing the Memcached cache driver. All of the servers
66    | should contain a value for "host", "port", and "weight" options.
67    |
68    */
69
70    'memcached' => array(
71
72      array('host' => '127.0.0.1', 'port' => 11211, 'weight' => 100),
73
74    ),
75
76    /*
77    |--------------------------------------------------------------------------
78    | Cache Key Prefix
79    |--------------------------------------------------------------------------
80    |
81    | When utilizing a RAM based store such as APC or Memcached, there might
82    | be other applications utilizing the same cache. So, we'll specify a
83    | value to get prefixed to all our keys so we can avoid collisions.
84    |
85    */
86
87    'prefix' => 'laravel',
88
89  );
```

Listing 12: ../portal/app/config/compile.php

```
1   <?php
2
3   return array(
4
5     /*
6     |--------------------------------------------------------------------------
7     | Additional Compiled Classes
8     |--------------------------------------------------------------------------
9     |
10    | Here you may specify additional classes to include in the compiled file
11    | generated by the `artisan optimize` command. These should be classes
12    | that are included on basically every request into the application.
13    |
14    */
15
16
17
18  );
```

Listing 13: ../portal/app/config/database.php

```php
<?php

return array(

  /*
  |--------------------------------------------------------------------------
  | PDO Fetch Style
  |--------------------------------------------------------------------------
  |
  | By default, database results will be returned as instances of the PHP
  | stdClass object; however, you may desire to retrieve records in an
  | array format for simplicity. Here you can tweak the fetch style.
  |
  */

  'fetch' => PDO::FETCH_CLASS,

  /*
  |--------------------------------------------------------------------------
  | Default Database Connection Name
  |--------------------------------------------------------------------------
  |
  | Here you may specify which of the database connections below you wish
  | to use as your default connection for all database work. Of course
  | you may use many connections at once using the Database library.
  |
  */

  'default' => 'mysql',

  /*
  |--------------------------------------------------------------------------
  | Database Connections
  |--------------------------------------------------------------------------
  |
  | Here are each of the database connections setup for your application.
  | Of course, examples of configuring each database platform that is
  | supported by Laravel is shown below to make development simple.
  |
  |
  | All database work in Laravel is done through the PHP PDO facilities
  | so make sure you have the driver for your particular database of
  | choice installed on your machine before you begin development.
  |
  */

  'connections' => array(

    //'sqlite' => array(
    //  'driver'   => 'sqlite',
    //  'database' => __DIR__.'/../database/production.sqlite',
    //  'prefix'   => '',
    //),

    'mysql' => array(
      'driver'    => 'mysql',
```

```
57          'host'      => '127.0.0.1',
58          'database'  => 'portal',
59          'username'  => 'root',
60          'password'  => 'password',
61          'charset'   => 'utf8',
62          'collation' => 'utf8_unicode_ci',
63          'prefix'    => '',
64               'port' => '8889',
65      ),
66
67      //'pgsql' => array(
68      //  'driver'   => 'pgsql',
69      //  'host'     => 'localhost',
70      //  'database' => 'database',
71      //  'username' => 'root',
72      //  'password' => '',
73      //  'charset'  => 'utf8',
74      //  'prefix'   => '',
75      //  'schema'   => 'public',
76      //),
77
78      //'sqlsrv' => array(
79      //  'driver'   => 'sqlsrv',
80      //  'host'     => 'localhost',
81      //  'database' => 'database',
82      //  'username' => 'root',
83      //  'password' => '',
84      //  'prefix'   => '',
85      //),
86
87  ),
88
89  /*
90  |--------------------------------------------------------------------------
91  | Migration Repository Table
92  |--------------------------------------------------------------------------
93  |
94  | This table keeps track of all the migrations that have already run for
95  | your application. Using this information, we can determine which of
96  | the migrations on disk have not actually be run in the databases.
97  |
98  */
99
100 'migrations' => 'migrations',
101
102 /*
103 |--------------------------------------------------------------------------
104 | Redis Databases
105 |--------------------------------------------------------------------------
106 |
107 | Redis is an open source, fast, and advanced key-value store that also
108 | provides a richer set of commands than a typical key-value systems
109 | such as APC or Memcached. Laravel makes it easy to dig right in.
110 |
111 */
112
```

```
113     'redis' => array(
114
115       'cluster' => true,
116
117       'default' => array(
118         'host'     => '127.0.0.1',
119         'port'     => 6379,
120         'database' => 0,
121       ),
122
123     ),
124
125 );
```

Listing 14: ../portal/app/config/mail.php

```
1   <?php
2
3   return array(
4
5     /*
6     |--------------------------------------------------------------------------
7     | Mail Driver
8     |--------------------------------------------------------------------------
9     |
10    | Laravel supports both SMTP and PHP's "mail" function as drivers for the
11    | sending of e-mail. You may specify which one you're using throughout
12    | your application here. By default, Laravel is setup for SMTP mail.
13    |
14    | Supported: "smtp", "mail", "sendmail"
15    |
16    */
17
18    'driver' => 'smtp',
19
20    /*
21    |--------------------------------------------------------------------------
22    | SMTP Host Address
23    |--------------------------------------------------------------------------
24    |
25    | Here you may provide the host address of the SMTP server used by your
26    | applications. A default option is provided that is compatible with
27    | the Postmark mail service, which will provide reliable delivery.
28    |
29    */
30
31    'host' => 'mailtrap.io',
32
33    /*
34    |--------------------------------------------------------------------------
35    | SMTP Host Port
36    |--------------------------------------------------------------------------
37    |
38    | This is the SMTP port used by your application to delivery e-mails to
39    | users of your application. Like the host we have set this value to
```

```
40      | stay compatible with the Postmark e-mail application by default.
41      |
42      */
43
44      'port' => 2525,
45
46      /*
47      |--------------------------------------------------------------------------
48      | Global "From" Address
49      |--------------------------------------------------------------------------
50      |
51      | You may wish for all e-mails sent by your application to be sent from
52      | the same address. Here, you may specify a name and address that is
53      | used globally for all e-mails that are sent by your application.
54      |
55      */
56
57      'from' => array('address' => 'superawesomeplaceholder@gmail.com', 'name' => 'Test Dummy'),
58
59      /*
60      |--------------------------------------------------------------------------
61      | E-Mail Encryption Protocol
62      |--------------------------------------------------------------------------
63      |
64      | Here you may specify the encryption protocol that should be used when
65      | the application send e-mail messages. A sensible default using the
66      | transport layer security protocol should provide great security.
67      |
68      */
69
70      'encryption' => '',
71
72      /*
73      |--------------------------------------------------------------------------
74      | SMTP Server Username
75      |--------------------------------------------------------------------------
76      |
77      | If your SMTP server requires a username for authentication, you should
78      | set it here. This will get used to authenticate with your server on
79      | connection. You may also set the "password" value below this one.
80      |
81      */
82
83      'username' => 'portal-8d7087cb351d131f',
84
85      /*
86      |--------------------------------------------------------------------------
87      | SMTP Server Password
88      |--------------------------------------------------------------------------
89      |
90      | Here you may set the password required by your SMTP server to send out
91      | messages from your application. This will be given to the server on
92      | connection so that the application will be able to send messages.
93      |
94      */
95
```

```
    'password' => 'e244836ff938c8ec',

    /*
    |--------------------------------------------------------------------------
    | Sendmail System Path
    |--------------------------------------------------------------------------
    |
    | When using the "sendmail" driver to send e-mails, we will need to know
    | the path to where Sendmail lives on this server. A default path has
    | been provided here, which will work well on most of your systems.
    |
    */

    'sendmail' => '/usr/sbin/sendmail -bs',

    /*
    |--------------------------------------------------------------------------
    | Mail "Pretend"
    |--------------------------------------------------------------------------
    |
    | When this option is enabled, e-mail will not actually be sent over the
    | web and will instead be written to your application's logs files so
    | you may inspect the message. This is great for local development.
    |
    */

    'pretend' => false,

);
```

Listing 15: ../portal/app/config/queue.php

```
<?php

return array(

    /*
    |--------------------------------------------------------------------------
    | Default Queue Driver
    |--------------------------------------------------------------------------
    |
    | The Laravel queue API supports a variety of back-ends via an unified
    | API, giving you convenient access to each back-end using the same
    | syntax for each one. Here you may set the default queue driver.
    |
    | Supported: "sync", "beanstalkd", "sqs", "iron"
    |
    */

    'default' => 'sync',

    /*
    |--------------------------------------------------------------------------
    | Queue Connections
    |--------------------------------------------------------------------------
```

```
24    |
25    | Here you may configure the connection information for each server that
26    | is used by your application. A default configuration has been added
27    | for each back-end shipped with Laravel. You are free to add more.
28    |
29    */
30
31    'connections' => array(
32
33      'sync' => array(
34        'driver' => 'sync',
35      ),
36
37      'beanstalkd' => array(
38        'driver' => 'beanstalkd',
39        'host'   => 'localhost',
40        'queue'  => 'default',
41      ),
42
43      'sqs' => array(
44        'driver' => 'sqs',
45        'key'    => 'your-public-key',
46        'secret' => 'your-secret-key',
47        'queue'  => 'your-queue-url',
48        'region' => 'us-east-1',
49      ),
50
51      'iron' => array(
52        'driver'  => 'iron',
53        'project' => 'your-project-id',
54        'token'   => 'your-token',
55        'queue'   => 'your-queue-name',
56      ),
57
58    ),
59
60  );
```

Listing 16: ../portal/app/config/session.php

```
1   <?php
2
3   return array(
4
5     /*
6     |--------------------------------------------------------------------------
7     | Default Session Driver
8     |--------------------------------------------------------------------------
9     |
10    | This option controls the default session "driver" that will be used on
11    | requests. By default, we will use the lightweight native driver but
12    | you may specify any of the other wonderful drivers provided here.
13    |
14    | Supported: "native", "cookie", "database", "apc",
15    |            "memcached", "redis", "array"
```

```
16      |
17      */
18
19      'driver' => 'native',
20
21      /*
22      |--------------------------------------------------------------------------
23      | Session Lifetime
24      |--------------------------------------------------------------------------
25      |
26      | Here you may specify the number of minutes that you wish the session
27      | to be allowed to remain idle before it expires. If you want them
28      | to immediately expire when the browser closes, set it to zero.
29      |
30      */
31
32      'lifetime' => 120,
33
34      /*
35      |--------------------------------------------------------------------------
36      | Session File Location
37      |--------------------------------------------------------------------------
38      |
39      | When using the native session driver, we need a location where session
40      | files may be stored. A default has been set for you but a different
41      | location may be specified. This is only needed for file sessions.
42      |
43      */
44
45      'files' => storage_path().'/sessions',
46
47      /*
48      |--------------------------------------------------------------------------
49      | Session Database Connection
50      |--------------------------------------------------------------------------
51      |
52      | When using the "database" session driver, you may specify the database
53      | connection that should be used to manage your sessions. This should
54      | correspond to a connection in your "database" configuration file.
55      |
56      */
57
58      'connection' => null,
59
60      /*
61      |--------------------------------------------------------------------------
62      | Session Database Table
63      |--------------------------------------------------------------------------
64      |
65      | When using the "database" session driver, you may specify the table we
66      | should use to manage the sessions. Of course, a sensible default is
67      | provided for you; however, you are free to change this as needed.
68      |
69      */
70
71      'table' => 'sessions',
```

```
72
73      /*
74      |--------------------------------------------------------------------------
75      | Session Sweeping Lottery
76      |--------------------------------------------------------------------------
77      |
78      | Some session drivers must manually sweep their storage location to get
79      | rid of old sessions from storage. Here are the chances that it will
80      | happen on a given request. By default, the odds are 2 out of 100.
81      |
82      */
83
84      'lottery' => array(2, 100),
85
86      /*
87      |--------------------------------------------------------------------------
88      | Session Cookie Name
89      |--------------------------------------------------------------------------
90      |
91      | Here you may change the name of the cookie used to identify a session
92      | instance by ID. The name specified here will get used every time a
93      | new session cookie is created by the framework for every driver.
94      |
95      */
96
97      'cookie' => 'laravel_session',
98
99      /*
100     |--------------------------------------------------------------------------
101     | Session Cookie Path
102     |--------------------------------------------------------------------------
103     |
104     | The session cookie path determines the path for which the cookie will
105     | be regarded as available. Typically, this will be the root path of
106     | your application but you are free to change this when necessary.
107     |
108     */
109
110     'path' => '/',
111
112     /*
113     |--------------------------------------------------------------------------
114     | Session Cookie Domain
115     |--------------------------------------------------------------------------
116     |
117     | Here you may change the domain of the cookie used to identify a session
118     | in your application. This will determine which domains the cookie is
119     | available to in your application. A sensible default has been set.
120     |
121     */
122
123     'domain' => null,
124
125 );
```

Listing 17: ../portal/app/config/view.php

```
<?php

return array(

  /*
  |--------------------------------------------------------------------------
  | View Storage Paths
  |--------------------------------------------------------------------------
  |
  | Most templating systems load templates from disk. Here you may specify
  | an array of paths that should be checked for your views. Of course
  | the usual Laravel view path has already been registered for you.
  |
  */

  'paths' => array(__DIR__.'/../views'),

  /*
  |--------------------------------------------------------------------------
  | Pagination View
  |--------------------------------------------------------------------------
  |
  | This view will be used to render the pagination link output, and can
  | be easily customized here to show any view you like. A clean view
  | compatible with Twitter's Bootstrap is given to you by default.
  |
  */

  'pagination' => 'pagination',

);
```

Listing 18: ../portal/app/config/workbench.php

```
<?php

return array(

  /*
  |--------------------------------------------------------------------------
  | Workbench Author Name
  |--------------------------------------------------------------------------
  |
  | When you create new packages via the Artisan "workbench" command your
  | name is needed to generate the composer.json file for your package.
  | You may specify it now so it is used for all of your workbenches.
  |
  */

  'name' => '',

  /*
  |--------------------------------------------------------------------------
```

```
    | Workbench Author E-Mail Address
    |--------------------------------------------------------------------------
    |
    | Like the option above, your e-mail address is used when generating new
    | workbench packages. The e-mail is placed in your composer.json file
    | automatically after the package is created by the workbench tool.
    |
    */

    'email' => '',

);
```

Listing 19: ../portal/app/config/packages/cartalyst/sentry/config.php

```
<?php
/**
 * Part of the Sentry package.
 *
 * NOTICE OF LICENSE
 *
 * Licensed under the 3-clause BSD License.
 *
 * This source file is subject to the 3-clause BSD License that is
 * bundled with this package in the LICENSE file.  It is also available at
 * the following URL: http://www.opensource.org/licenses/BSD-3-Clause
 *
 * @package    Sentry
 * @version    2.0.0
 * @author     Cartalyst LLC
 * @license    BSD License (3-clause)
 * @copyright  (c) 2011 - 2013, Cartalyst LLC
 * @link       http://cartalyst.com
 */

return array(

    /*
    |--------------------------------------------------------------------------
    | Default Authentication Driver
    |--------------------------------------------------------------------------
    |
    | This option controls the authentication driver that will be utilized.
    | This drivers manages the retrieval and authentication of the users
    | attempting to get access to protected areas of your application.
    |
    | Supported: "eloquent" (more coming soon).
    |
    */

    'driver' => 'eloquent',

    /*
    |--------------------------------------------------------------------------
    | Default Hasher
```

```
41    |--------------------------------------------------------------------------
42    |
43    | This option allows you to specify the default hasher used by Sentry
44    |
45    | Supported: "native", "bcrypt", "sha256", "whirlpool"
46    |
47    */
48
49    'hasher' => 'native',
50
51    /*
52    |--------------------------------------------------------------------------
53    | Cookie
54    |--------------------------------------------------------------------------
55    |
56    | Configuration specific to the cookie component of Sentry.
57    |
58    */
59
60    'cookie' => array(
61
62      /*
63      |--------------------------------------------------------------------------
64      | Default Cookie Key
65      |--------------------------------------------------------------------------
66      |
67      | This option allows you to specify the default cookie key used by Sentry.
68      |
69      | Supported: string
70      |
71      */
72
73      'key' => 'cartalyst_sentry',
74
75    ),
76
77    /*
78    |--------------------------------------------------------------------------
79    | Groups
80    |--------------------------------------------------------------------------
81    |
82    | Configuration specific to the group management component of Sentry.
83    |
84    */
85
86    'groups' => array(
87
88      /*
89      |--------------------------------------------------------------------------
90      | Model
91      |--------------------------------------------------------------------------
92      |
93      | When using the "eloquent" driver, we need to know which
94      | Eloquent models should be used throughout Sentry.
95      |
96      */
```

```
97
98      'model' => 'Cartalyst\Sentry\Groups\Eloquent\Group',
99
100   ),
101
102   /*
103   |--------------------------------------------------------------------------
104   | Users
105   |--------------------------------------------------------------------------
106   |
107   | Configuration specific to the user management component of Sentry.
108   |
109   */
110
111   'users' => array(
112
113     /*
114     |--------------------------------------------------------------------------
115     | Model
116     |--------------------------------------------------------------------------
117     |
118     | When using the "eloquent" driver, we need to know which
119     | Eloquent models should be used throughout Sentry.
120     |
121     */
122
123     'model' => 'User',
124
125     /*
126     |--------------------------------------------------------------------------
127     | Login Attribute
128     |--------------------------------------------------------------------------
129     |
130     | If you're using the "eloquent" driver and extending the base Eloquent
131     | model, we allow you to globally override the login attribute without
132     | even subclassing the model, simply by specifying the attribute below.
133     |
134     */
135
136     'login_attribute' => 'username',
137
138   ),
139
140   /*
141   |--------------------------------------------------------------------------
142   | User Groups Pivot Table
143   |--------------------------------------------------------------------------
144   |
145   | When using the "eloquent" driver, you can specify the table name
146   | for the user groups pivot table.
147   |
148   | Default: users_groups
149   |
150   */
151
152   'user_groups_pivot_table' => 'users_groups',
```

```
153
154     /*
155     |--------------------------------------------------------------------------
156     | Throttling
157     |--------------------------------------------------------------------------
158     |
159     | Throttling is an optional security feature for authentication, which
160     | enables limiting of login attempts and the suspension & banning of users.
161     |
162     */
163
164     'throttling' => array(
165
166       /*
167       |--------------------------------------------------------------------------
168       | Throttling
169       |--------------------------------------------------------------------------
170       |
171       | Enable throttling or not. Throttling is where users are only allowed a
172       | certain number of login attempts before they are suspended. Suspension
173       | must be removed before a new login attempt is allowed.
174       |
175       */
176
177       'enabled' => true,
178
179       /*
180       |--------------------------------------------------------------------------
181       | Model
182       |--------------------------------------------------------------------------
183       |
184       | When using the "eloquent" driver, we need to know which
185       | Eloquent models should be used throughout Sentry.
186       |
187       */
188
189       'model' => 'Cartalyst\Sentry\Throttling\Eloquent\Throttle',
190
191       /*
192       |--------------------------------------------------------------------------
193       | Attempts Limit
194       |--------------------------------------------------------------------------
195       |
196       | When using the "eloquent" driver and extending the base Eloquent model,
197       | you have the option to globally set the login attempts.
198       |
199       | Supported: int
200       |
201       */
202
203       'attempt_limit' => 5,
204
205       /*
206       |--------------------------------------------------------------------------
207       | Suspension Time
208       |--------------------------------------------------------------------------
```

```
209       |
210       | When using the "eloquent" driver and extending the base Eloquent model,
211       | you have the option to globally set the suspension time, in minutes.
212       |
213       | Supported: int
214       |
215       */
216
217       'suspension_time' => 15,
218
219     ),
220
221 );
```

Listing 20: ../portal/app/config/packages/codesleeve/stapler/filesystem.php

```
1  <?php
2
3  return [
4
5    /*
6    |--------------------------------------------------------------------------
7    | Stapler File Url
8    |--------------------------------------------------------------------------
9    |
10   | The url (relative to your project document root) where files will be stored.
11   | It is composed of 'interpolations' that will be replaced their
12   | corresponding values during runtime.  It's unique in that it functions as both
13   | a configuration option and an interpolation.
14   |
15   */
16
17   'url' => '/system/:class/:attachment/:id_partition/:style/:filename',
18
19   /*
20   |--------------------------------------------------------------------------
21   | Stapler File Path
22   |--------------------------------------------------------------------------
23   |
24   | Similar to the url, the path option is the location where your files will
25   | be stored at on disk.  It should be noted that the path option should not
26   | conflict with the url option.  Stapler provides sensible defaults that take
27   | care of this for you.
28   |
29   */
30
31   'path' => ':laravel_root/public:url',
32
33   /*
34   |--------------------------------------------------------------------------
35   | Override File Permissions Flag
36   |--------------------------------------------------------------------------
37   |
38   | Override the default file permissions used by stapler when creating a new
39   | file in the file system.  Leaving this value as null will result in stapler
```

```
    | chmod'ing files to 0666.  Set it to a specific octal value and stapler will
    | chmod accordingly.  Set it to false to prevent chmod from occuring (useful
    | for non unix-based environments).
    |
    */

    'override_file_permissions' => null,

];
```

Listing 21: ../portal/app/config/packages/codesleeve/stapler/mimes.php

```
<?php

return array(

    'hqx'   => 'application/mac-binhex40',
    'cpt'   => 'application/mac-compactpro',
    'csv'   => array('text/x-comma-separated-values', 'text/comma-separated-values', 'application/octet-stream'),
    'bin'   => 'application/macbinary',
    'dms'   => 'application/octet-stream',
    'lha'   => 'application/octet-stream',
    'lzh'   => 'application/octet-stream',
    'exe'   => array('application/octet-stream', 'application/x-msdownload'),
    'class' => 'application/octet-stream',
    'psd'   => 'application/x-photoshop',
    'so'    => 'application/octet-stream',
    'sea'   => 'application/octet-stream',
    'dll'   => 'application/octet-stream',
    'oda'   => 'application/oda',
    'pdf'   => array('application/pdf', 'application/x-download'),
    'ai'    => 'application/postscript',
    'eps'   => 'application/postscript',
    'ps'    => 'application/postscript',
    'smi'   => 'application/smil',
    'smil'  => 'application/smil',
    'mif'   => 'application/vnd.mif',
    'xls'   => array('application/excel', 'application/vnd.ms-excel', 'application/msexcel'),
    'ppt'   => array('application/powerpoint', 'application/vnd.ms-powerpoint'),
    'wbxml' => 'application/wbxml',
    'wmlc'  => 'application/wmlc',
    'dcr'   => 'application/x-director',
    'dir'   => 'application/x-director',
    'dxr'   => 'application/x-director',
    'dvi'   => 'application/x-dvi',
    'gtar'  => 'application/x-gtar',
    'gz'    => 'application/x-gzip',
    'php'   => array('application/x-httpd-php', 'text/x-php'),
    'php4'  => 'application/x-httpd-php',
    'php3'  => 'application/x-httpd-php',
    'phtml' => 'application/x-httpd-php',
    'phps'  => 'application/x-httpd-php-source',
    'js'    => 'application/x-javascript',
    'swf'   => 'application/x-shockwave-flash',
    'sit'   => 'application/x-stuffit',
```

```
    'tar'   => 'application/x-tar',
    'tgz'   => array('application/x-tar', 'application/x-gzip-compressed'),
    'xhtml' => 'application/xhtml+xml',
    'xht'   => 'application/xhtml+xml',
    'zip'   => array('application/x-zip', 'application/zip', 'application/x-zip-compressed'),
    'mid'   => 'audio/midi',
    'midi'  => 'audio/midi',
    'mpga'  => 'audio/mpeg',
    'mp2'   => 'audio/mpeg',
    'mp3'   => array('audio/mpeg', 'audio/mpg', 'audio/mpeg3', 'audio/mp3'),
    'aif'   => 'audio/x-aiff',
    'aiff'  => 'audio/x-aiff',
    'aifc'  => 'audio/x-aiff',
    'ram'   => 'audio/x-pn-realaudio',
    'rm'    => 'audio/x-pn-realaudio',
    'rpm'   => 'audio/x-pn-realaudio-plugin',
    'ra'    => 'audio/x-realaudio',
    'rv'    => 'video/vnd.rn-realvideo',
    'wav'   => 'audio/x-wav',
    'bmp'   => 'image/bmp',
    'gif'   => 'image/gif',
    'jpeg'  => array('image/jpeg', 'image/pjpeg'),
    'jpg'   => array('image/jpeg', 'image/pjpeg'),
    'jpe'   => array('image/jpeg', 'image/pjpeg'),
    'png'   => 'image/png',
    'tiff'  => 'image/tiff',
    'tif'   => 'image/tiff',
    'css'   => 'text/css',
    'html'  => 'text/html',
    'htm'   => 'text/html',
    'shtml' => 'text/html',
    'txt'   => 'text/plain',
    'text'  => 'text/plain',
    'log'   => array('text/plain', 'text/x-log'),
    'rtx'   => 'text/richtext',
    'rtf'   => 'text/rtf',
    'xml'   => 'text/xml',
    'xsl'   => 'text/xml',
    'mpeg'  => 'video/mpeg',
    'mpg'   => 'video/mpeg',
    'mpe'   => 'video/mpeg',
    'qt'    => 'video/quicktime',
    'mov'   => 'video/quicktime',
    'avi'   => 'video/x-msvideo',
    'movie' => 'video/x-sgi-movie',
    'doc'   => 'application/msword',
    'docx'  => 'application/vnd.openxmlformats-officedocument.wordprocessingml.document',
    'xlsx'  => 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet',
    'word'  => array('application/msword', 'application/octet-stream'),
    'xl'    => 'application/excel',
    'eml'   => 'message/rfc822',
    'json'  => array('application/json', 'text/json'),

);
```

Listing 22: ../portal/app/config/packages/codesleeve/stapler/s3.php

```
<?php

return [

  /*
  |--------------------------------------------------------------------------
  | AWS Access Key
  |--------------------------------------------------------------------------
  |
  | This is an alphanumeric text string that uniquely
  | identifies the user who owns the account. No two accounts can have the
  | same AWS Access Key.
  |
  */

  'key' => '',

  /*
  |--------------------------------------------------------------------------
  | AWS Secret Key
  |--------------------------------------------------------------------------
  |
  | This key plays the role of a  password . It's called secret because it is
  | assumed to be known by the owner only.  A Password with Access Key forms a
  | secure information set that confirms the user's identity.
  | You are advised to keep your Secret Key in a safe place.
  |
  */

  'secret' => '',

  /*
  |--------------------------------------------------------------------------
  | S3 Bucket
  |--------------------------------------------------------------------------
  |
  | The bucket where you wish to store your objects.  Every object in Amazon S3
  | is stored in a bucket.  If the specified bucket doesn't exist Stapler will
  | attempt to create it.  The bucket name will not be interpolated.
  | You can define the bucket as a closure if you want to determine it's name at
  | runtime. Stapler will call that closure with attachment as the only argument.
  |
  */

  'bucket' => '',

  /*
  |--------------------------------------------------------------------------
  | S3 ACL (s3_permissions)
  |--------------------------------------------------------------------------
  |
  | This is a string/array that should be one of the canned access policies
  | that S3 provides (private, public-read, public-read-write, authenticated-read,
  | bucket-owner-read, bucket-owner-full-control). The default for Stapler is
```

```
    | public-read.  An associative array (style => permission) may be passed to
    | specify permissions on a per style basis.
    |
    */

    'ACL' => 'public-read',

    /*
    |--------------------------------------------------------------------------
    | S3 Scheme (s3_protocol)
    |--------------------------------------------------------------------------
    |
    | The protocol for the URLs generated to your S3 assets. Can be either 'http'
    | or 'https'. Defaults to 'http' when your ACL is 'public-read' (the default)
    | and 'https' when your ACL is anything else.
    |
    */
    'scheme' => 'http',

    /*
    |--------------------------------------------------------------------------
    | S3 Region
    |--------------------------------------------------------------------------
    |
    | The region name of your bucket (e.g. 'us-east-1', 'us-west-1', 'us-west-2',
    | 'eu-west-1').  Determines the base url where your objects are stored at
    | (e.g a region of us-west-2 has a base url of s3-us-west-2.amazonaws.com).
    | Defaults to empty (US Standard *).
    |
    */

    'region' => '',

    /*
    |--------------------------------------------------------------------------
    | S3 Path
    |--------------------------------------------------------------------------
    |
    | This is the key under the bucket in which the file will be stored.
    | The URL will be constructed from the bucket and the path.
    | This is what you will want to interpolate. Keys should be unique,
    | like filenames, and despite the fact that S3 (strictly speaking) does not
    | support directories, you can still use a / to separate parts of your file name.
    |
    */

    'path' => ':attachment/:id/:style/:filename',

];
```

Listing 23: ../portal/app/config/packages/codesleeve/stapler/stapler.php

```
<?php

return [
```

```
4
5     /*
6     |--------------------------------------------------------------------------
7     | Stapler Storage Driver
8     |--------------------------------------------------------------------------
9     |
10    | The default mechanism for handling file storage.  Currently Stapler supports
11    | both file system and Amazon S3 as options.
12    |
13    */
14
15    'storage' => 'filesystem',
16
17    /*
18    |--------------------------------------------------------------------------
19    | Stapler Image Processing Library
20    |--------------------------------------------------------------------------
21    |
22    | The default library used for image processing.  Can be one GD, Imagick, or
23    | Gmagick.
24    |
25    */
26
27    'image_processing_library' => 'GD',
28
29
30    /*
31    |--------------------------------------------------------------------------
32    | Stapler Default Url
33    |--------------------------------------------------------------------------
34    |
35    | The url (relative to your project document root) containing a default image
36    | that will be used for attachments that don't currently have an uploaded image
37    | attached to them.
38    |
39    */
40
41    'default_url' => '/:attachment/:style/missing.png',
42
43    /*
44    |--------------------------------------------------------------------------
45    | Stapler Default Style
46    |--------------------------------------------------------------------------
47    |
48    | The default style returned from the Stapler file location helper methods.
49    | An unaltered version of uploaded file is always stored within the 'original'
50    | style, however the default_style can be set to point to any of the defined
51    | syles within the styles array.
52    |
53    */
54
55    'default_style' => 'original',
56
57    /*
58    |--------------------------------------------------------------------------
59    | Stapler Styles
```

```
    |--------------------------------------------------------------------------
    |
    | An array of image sizes defined for the file attachment.
    | Stapler will attempt to format the file upload into the defined style.
    |
    */

    'styles' => [],

    /*
    |--------------------------------------------------------------------------
    | Keep Old Files Flag
    |--------------------------------------------------------------------------
    |
    | Set this to true in order to prevent older file uploads from being deleted
    | from storage when a record is updated with a new upload.
    |
    */

    'keep_old_files' => false,

    /*
    |--------------------------------------------------------------------------
    | Preserve Files Flag
    |--------------------------------------------------------------------------
    |
    | Set this to true in order to prevent file uploads from being deleted
    | from the file system when an attachment is destroyed.  Essentially this
    | ensures the preservation of uploads event after their corresponding database
    | records have been removed.
    |
    */
    'preserve_files' => false

];
```

Listing 24: ../portal/app/start/artisan.php

```
<?php

/*
|--------------------------------------------------------------------------
| Register The Artisan Commands
|--------------------------------------------------------------------------
|
| Each available Artisan command must be registered with the console so
| that it is available to be called. We'll register every command so
| the console gets access to each of the command object instances.
|
*/
```

Listing 25: ../portal/app/start/global.php

```
<?php
```

```
2
3   /*
4   |--------------------------------------------------------------------------
5   | Register The Laravel Class Loader
6   |--------------------------------------------------------------------------
7   |
8   | In addition to using Composer, you may use the Laravel class loader to
9   | load your controllers and models. This is useful for keeping all of
10  | your classes in the "global" namespace without Composer updating.
11  |
12  */
13
14  ClassLoader::addDirectories(array(
15
16    app_path().'/commands',
17    app_path().'/controllers',
18    app_path().'/models',
19    app_path().'/database/seeds',
20
21  ));
22
23  /*
24  |--------------------------------------------------------------------------
25  | Application Error Logger
26  |--------------------------------------------------------------------------
27  |
28  | Here we will configure the error logger setup for the application which
29  | is built on top of the wonderful Monolog library. By default we will
30  | build a rotating log file setup which creates a new file each day.
31  |
32  */
33
34  $logFile = 'log-'.php_sapi_name().'.txt';
35
36  Log::useDailyFiles(storage_path().'/logs/'.$logFile);
37
38  /*
39  |--------------------------------------------------------------------------
40  | Application Error Handler
41  |--------------------------------------------------------------------------
42  |
43  | Here you may handle any errors that occur in your application, including
44  | logging them or displaying custom views for specific errors. You may
45  | even register several error handlers to handle different types of
46  | exceptions. If nothing is returned, the default error view is
47  | shown, which includes a detailed stack trace during debug.
48  |
49  */
50
51  App::error(function(Exception $exception, $code)
52  {
53    Log::error($exception);
54    switch ($code)
55        {
56            case 403:
57                return Response::view('errors.403', array(), 403);
```

```
58
59              case 404:
60                  return Response::view('errors.404', array(), 404);
61
62              // case 500:
63  //                  return Response::view('errors.500', array(), 500);
64
65              // default:
66      //                  return Response::view('errors.default', array(), $code);
67          }
68  });
69
70  /*
71  |--------------------------------------------------------------------------
72  | Maintenance Mode Handler
73  |--------------------------------------------------------------------------
74  |
75  | The "down" Artisan command gives you the ability to put an application
76  | into maintenance mode. Here, you will define what is displayed back
77  | to the user if maintenace mode is in effect for this application.
78  |
79  */
80
81  App::down(function()
82  {
83    return Response::view('errors.503',array(),503);
84  });
85
86  /*
87  |--------------------------------------------------------------------------
88  | Require The Filters File
89  |--------------------------------------------------------------------------
90  |
91  | Next we will load the filters file for the application. This gives us
92  | a nice separate location to store our route and application filter
93  | definitions instead of putting them all in the main routes file.
94  |
95  */
96
97  require app_path().'/filters.php';
98
99  /*
100  | Load the view composers and form macros, etc
101  |
102  |
103  */
104 require app_path().'/helpers.php';
```

Listing 26: ../portal/app/start/local.php

```
1  <?php
2
3   //
```

Listing 27: ../portal/app/controllers/BaseController.php

```
<?php

/**
 * Base controller
 *
 *
 */
class BaseController extends Controller {

  /**
   * Setup the layout used by the controller.
   *
   * @return void
   */
  protected function setupLayout()
  {
    if ( ! is_null($this->layout))
    {
      $this->layout = View::make($this->layout);
    }
  }

}
```

Listing 28: ../portal/app/controllers/DepartmentController.php

```
<?php

class DepartmentController extends \BaseController {


  public function __construct()
  {
    $this->beforeFilter('auth');
    $this->beforeFilter('csrf',['on'=>'post']);
    $this->beforeFilter('admin');
  }
  /**
   * Display a listing of the resource.
   *
   * @return Response
   */
  public function index()
  {
    $departments = Department::where('active',1)->paginate(5);
    $users = Sentry::findAllUsersInGroup(Sentry::findGroupByName('medical_technicians'));
    $users = $users->filter(function($user){return $user->has('employee');});
    $personnel = array();
    foreach($users as $user)
    {
      if($user->employee->department_id==NULL)
      {
        $personnel[$user->employee->id] = $user->employee->last_name . ", " . $user->employee->first_name;
```

```
28        }
29
30      }
31
32      //var_dump($personnel);
33      return View::make('departments.index')->with('departments',$departments)->with('personnel',$personnel);
34    }
35
36    /**
37     * Store a newly created resource in storage.
38     *
39     * @return Response
40     */
41    public function store()
42    {
43
44      $addedBy = Sentry::getUser()->employee->id;
45
46      $validator = Validator::make(Input::all(),array('department_name'=>'required'));
47      if($validator->passes()){
48        $department = new Department;
49        $department->department_name = Input::get('department_name');
50
51        $department->save();
52        if(Input::get('assigned')){
53        foreach(Input::get('assigned') as $assigned)
54        {
55          $employee = Employee::find($assigned);
56          $employee->department_id = $department->id;
57          $employee->save();
58
59        }
60        }
61        if(Input::get('employee_email')){
62          $emails = Input::get('employee_email');
63          $names = Input::get('employee_name');
64          $newemployees = array();
65          foreach($emails as $key=>$email)
66          {
67            if(count(explode(',',$names[$key]))!=2){
68              $errorProvider = new ErrorProvider;
69              $errorProvider->addMessage('general','Invalid name for new employee.');
70              return Redirect::back()->withErrors($errorProvider);
71            }
72
73            $employeename = explode(",",$names[$key]);
74            try{
75              $user = User::registerWithOptions($email,User::generatePassword(),'medical_technicians',$employeename
                      [0],$employeename[1]);
76              $employeeRecord = new Employee;
77              $employeeRecord = new Employee;
78              $employeeRecord->user_id = $user->id;
79              $employeeRecord->first_name = $employeename[1];
80              $employeeRecord->last_name = $employeename[0];
81              $employeeRecord->added_by = $addedBy;
82              $employeeRecord->department_id = $department->id;
```

```
83              $employeeRecord->save();
84
85            }
86            catch(Exception $e){
87              $errorProvider = new ErrorProvider;
88              $errorProvider->addMessage('general','Could not register user '.$employeename[0] );
89              return Redirect::back()->withErrors($errorProvider);
90            }
91
92
93          }
94        }
95
96
97        return Redirect::to('departments')->with('success','New department created successfully');
98      }
99      else
100     {
101       return Redirect::to('departments')->withInput()->withErrors($validator);
102     }
103   }
104
105   /**
106    * Display the specified resource.
107    *
108    * @param  int  $id
109    * @return Response
110    */
111   public function show($id)
112   {
113     //TODO a page showing all the tests of the department
114   }
115
116
117   /**
118    * Update the specified resource in storage.
119    *
120    * @param  int  $id
121    * @return Response
122    */
123   public function update($id)
124   {
125     $validator = Validator::make(Input::all(),array('department_name'=>'required'));
126     if($validator->passes())
127     {
128       $department = Department::find($id);
129       $department->department_name = Input::get('department_name');
130       $department->save();
131
132       return Redirect::route('departments.index')->with('success','Changes saved successfully.');
133     }
134     else
135     {
136       $errorProvider = new ErrorProvider;
137       $errorProvider->addMessage('general','Could not save changes.');
138       return Redirect::route('departments.index')->withErrors($errorProvider);
```

```
139       }
140     }
141
142     /**
143      * Remove the specified resource from storage.
144      *
145      * @param  int  $id
146      * @return Response
147      */
148     // public function destroy($id)
149   //  {
150   //    $department = Department::find($id);
151   //    $department->delete();
152   //    return Redirect::to('departments')->with('success','Department successfully deleted.');
153   //  }
154
155   public function disable($id)
156   {
157     $department = Department::find($id);
158     $department->active = 0;
159     $department->save();
160     return Redirect::to('departments')->with('success','Department successfully disabled.');
161   }
162
163   }
```

Listing 29: ../portal/app/controllers/PatientController.php

```
1   <?php
2
3   class PatientController extends \BaseController {
4
5     public function __construct()
6     {
7       $this->beforeFilter('auth');
8       $this->beforeFilter('csrf',['on'=>'post']);
9       $this->beforeFilter('hasAccess:create_patient',
10      array('only'=>array('create','store','edit','update')));
11      $this->beforeFilter('hasAccess:delete_patient',array('only'=>array('destroy')));
12      $this->beforeFilter('inGroup:clerks',array('only'=>array('outbox','pending')));
13      $this->beforeFilter('inGroup:doctors',array('only'=>array('inbox','outbox_doctor','waiting','waiting2')));
14
15    }
16
17    /**
18     *
19     * @return response
20     */
21    public function redirectById(){
22      $validator = Validator::make(Input::all(),array(
23        'id'=>'required|exists:patients,patient_id|numeric'
24      ));
25      if($validator->passes()){
26
27        return Redirect::to('patients/'.Input::get('id'));
```

```
28      }
29      else{
30        return Redirect::to('patients/search')->withErrors($validator);
31      }
32    }
33
34    /**
35    * Show the search form
36    *
37    * @return Response
38    **/
39    public function search()
40    {
41      if(Input::get())
42      {
43        //if there are search params
44
45          if(Input::has('last_name')){
46            if(Input::has('first_name')){
47              $results = Patient::where('last_name',
48              'LIKE','%'.Input::get('last_name').'%')
49                ->where('first_name','LIKE','%'.Input::get('first_name').'%');
50
51            }
52            else
53            {
54              $results = Patient::where('last_name',
55              'LIKE','%'.Input::get('last_name').'%');
56            }
57            $allowed = array('last_name','id');
58            $sort = in_array(Input::get('sort'),$allowed) ? Input::get('sort') : 'id';
59            $order = Input::get('order') == 'desc' ? 'desc' : 'asc';
60            $results = $results->orderBy($sort,$order);
61            $results = $results->paginate(5);
62
63          }else{
64            //get everything
65            $allowed = array('last_name','id');
66            $sort = in_array(Input::get('sort'),$allowed) ? Input::get('sort') : 'id';
67            $order = Input::get('order') == 'desc' ? 'desc' : 'asc';
68            $results = Patient::orderBy($sort,$order)->paginate(5);
69          }
70
71      }else{
72        //no search params then display everything
73        $results = Patient::paginate(5);
74      }
75      //TODO Repopulate input when a search is performed
76      return View::make('patients.search')->with('results',$results);
77
78    }
79
80
81
82    /**
83    * Show the form for creating a new resource.
```

```
84    *
85    * @return Response
86    */
87    public function create()
88    {
89
90      return View::make('patients.create')->with('doctors',User::getDoctors());
91    }
92
93
94    /**
95    * Store a newly created resource in storage.
96    *
97    * @return Response
98    */
99    public function store()
100   {
101     $added_by = Sentry::getUser()->employee->id;
102     //echo $added_by;
103     $validator = Validator::make(Input::all(),
104     [
105       "last_name"=>"required",
106       "first_name"=>"required",
107       "sex"=>"required",
108       "birthdate"=>"required",
109       "date_of_visit"=>"required",
110       "hospital_id_1"=>"required",
111       "hospital_id_2"=>"required"
112
113         ]);
114
115       // If the option to create a new user account has been selected,
116       // validate to make sure the email address given is unique.
117       if(Input::has('email')){
118         $validator->sometimes('email','required|unique:users|email',function($input){
119           return true;
120         });
121       }
122
123       if($validator->passes())
124       {
125
126         $patient = new Patient;
127
128         //important to note: this line MUST be executed before the user
129         //is registered, because otherwise Sentry::getUser() will return
130         //the newly created User rather than the User who instantiated
131         //the User creation process
132         $addedBy = Sentry::getUser()->employee->employee_id;
133
134         if(Input::has('email'))
135         {
136           $email = Input::get('email');
137
138           $user = User::registerWithOptions($email,User::generatePassword(),'patients',Input::get('last_name'),
                Input::get('first_name'));
```

```
139             $patient->user_id = $user->id;
140           }
141           $patient->hospital_id = Input::get('hospital_id_1')."-".Input::get('hospital_id_2');
142
143           $patient->email = Input::get('email');
144           $patient->last_name =  Input::get('last_name');
145           $patient->first_name =  Input::get('first_name');
146           $patient->middle_name = Input::get('middle_name');
147           $patient->sex = Input::get('sex');
148           $patient->civil_status = Input::get('civil_status');
149           $patient->birthdate = Input::get('birthdate');
150           $patient->birthplace = Input::get('birthplace');
151           $patient->telephone_number = Input::get('telephone_number');
152           $patient->mobile_number = Input::get('mobile_number');
153           $patient->address = Input::get('address');
154           $patient->religion = Input::get('religion');
155           $patient->nationality = Input::get('nationality');
156           $patient->occupation = Input::get('occupation');
157
158           if(Input::has('is_dependent'))
159           {
160             $patient->is_dependent = Input::get('is_dependent');
161           }
162
163           if(Input::has('is_stockholder'))
164           {
165             $patient->is_stockholder = Input::get('is_stockholder');
166           }
167
168           if(Input::has('is_stockholder_dependent'))
169           {
170             $patient->is_stockholder_dependent =              Input::get('is_stockholder_dependent');
171           }
172
173           $patient->company_name = Input::get('company_name');
174           $patient->company_address = Input::get('company_address');
175           $patient->mother_name = Input::get('mother_name');
176           $patient->father_name = Input::get('father_name');
177           $patient->parent_address = Input::get('parent_address');
178           $patient->spouse_name = Input::get('spouse_name');
179           $patient->spouse_occupation = Input::get('spouse_occupation');
180           $patient->spouse_company = Input::get('spouse_company');
181
182           $patient->emergency_contact_name =            Input::get('emergency_contact_name');
183           $patient->emergency_contact_address =             Input::get('emergency_contact_address');
184           $patient->emergency_contact_number =            Input::get('emergency_contact_number');
185           $patient->account_responsibility_name
186             = Input::get('account_responsibility_name');
187           $patient->account_responsibility_relationship
188             = Input::get('account_responsibility_relationship');
189           $patient->account_responsibility_address
190             = Input::get('account_responsibility_address');
191           $patient->account_responsibility_number
192             = Input::get('account_responsibility_number');
193
194           $patient->added_by = $added_by;
```

```
195
196           $patient->save();
197           $visit = new Visit;
198           $visit->date_of_visit = Input::get('date_of_visit');
199
200           $visit->height = Input::get('height');
201           $visit->weight = Input::get('weight');
202           $visit->c_r = Input::get('c_r');
203           $visit->r_r = Input::get('r_r');
204           $visit->temperature = Input::get('temperature');
205           $visit->bp = Input::get('bp');
206           $visit->added_by = Sentry::getUser()->employee->id;
207
208           $patient->visits()->save($visit);
209
210
211           if(Input::has('attending_physicians'))
212           {
213             $attending_physicians = Input::get('attending_physicians');
214             foreach($attending_physicians as $attending_physician)
215             {
216               $visit->attendingPhysicians()->attach($attending_physician);
217             }
218           }
219
220
221
222           return Redirect::to('patients/'.$patient->id)
223             ->with('success','Patient successfully added!');
224         }
225         else
226         {
227
228           return Redirect::to('patients/create')->withInput()->withErrors($validator);
229
230         }
231
232       }
233
234     /**
235     * Display the specified resource.
236     *
237     * @param  int  $id
238     * @return Response
239     */
240     public function show($id)
241     {
242       $patient = Patient::find($id);
243       $visits = Visit::where('patient_id',$id);//paginate(5);
244
245       if(Input::get())
246       {
247         $allowed = array('date_of_visit','id');
248         $sort = in_array(Input::get('sort'),$allowed) ? Input::get('sort') : 'id';
249         $order = Input::get('order') == 'asc' ? 'asc' : 'desc';
250         $visits = $visits->orderBy($sort,$order);
```

```
251       }
252
253       $visits = $visits->paginate(5);
254
255       return View::make('patients/show')->with('patient',$patient)
256         ->with('visits',$visits);
257     }
258
259     /**
260     * Show the form for editing the specified resource.
261     *
262     * @param  int  $id
263     * @return Response
264     */
265     public function edit($id)
266     {
267       $patient =  Patient::find($id);
268
269       return View::make('patients/edit')->with('patient',$patient);
270     }
271
272     /**
273     * Update the specified resource in storage.
274     *
275     * @param  int  $id
276     * @return Response
277     */
278     public function update($id)
279     {
280       $validator = Validator::make(Input::all(),array(
281         "last_name"=>"required",
282         "first_name"=>"required",
283         "sex"=>"required",
284         "birthdate"=>"required",
285       ));
286
287       if($validator->passes())
288       {
289         $patient = Patient::find($id);
290         $patient->email = Input::get('email');
291         $patient->last_name =  Input::get('last_name');
292         $patient->first_name =  Input::get('first_name');
293         $patient->middle_name = Input::get('middle_name');
294         $patient->sex = Input::get('sex');
295         $patient->civil_status = Input::get('civil_status');
296         $patient->birthdate = Input::get('birthdate');
297         $patient->birthplace = Input::get('birthplace');
298         $patient->telephone_number = Input::get('telephone_number');
299         $patient->mobile_number = Input::get('mobile_number');
300         $patient->address = Input::get('address');
301         $patient->religion = Input::get('religion');
302         $patient->nationality = Input::get('nationality');
303         $patient->occupation = Input::get('occupation');
304
305         if(Input::has('is_dependent'))
306         {
```

```
307           $patient->is_dependent = Input::get('is_dependent');
308         }
309
310         if(Input::has('is_stockholder'))
311         {
312           $patient->is_stockholder = Input::get('is_stockholder');
313         }
314
315         if(Input::has('is_stockholder_dependent'))
316         {
317           $patient->is_stockholder_dependent =              Input::get('is_stockholder_dependent');
318         }
319
320         $patient->company_name = Input::get('company_name');
321         $patient->company_address = Input::get('company_address');
322         $patient->mother_name = Input::get('mother_name');
323         $patient->father_name = Input::get('father_name');
324         $patient->parent_address = Input::get('parent_address');
325         $patient->spouse_name = Input::get('spouse_name');
326         $patient->spouse_occupation = Input::get('spouse_occupation');
327         $patient->spouse_company = Input::get('spouse_company');
328
329         $patient->emergency_contact_name =            Input::get('emergency_contact_name');
330         $patient->emergency_contact_address =             Input::get('emergency_contact_address');
331         $patient->emergency_contact_number =            Input::get('emergency_contact_number');
332         $patient->account_responsibility_name
333           = Input::get('account_responsibility_name');
334         $patient->account_responsibility_relationship
335           = Input::get('account_responsibility_relationship');
336         $patient->account_responsibility_address
337           = Input::get('account_responsibility_address');
338         $patient->account_responsibility_number
339           = Input::get('account_responsibility_number');
340
341         $patient->save();
342         return Redirect::to('patients/'.$id)
343           ->with('success','Successfully edited patient record!');
344
345       }
346       else
347       {
348
349         return Redirect::to('patients/'.$id.'/edit')
350           ->withErrors($validator)
351             ->withInput();
352       }
353     }
354
355     /**
356     * Remove the specified resource from storage.
357     *
358     * @param  int  $id
359     * @return Response
360     */
361     public function destroy($id)
362     {
```

```
363       $patient = Patient::find($id);
364       $patient->delete();
365       return Redirect::to('patients/search')->with('success','Patient record successfully deleted.');
366     }
367
368     public function pending()
369     {
370       $user = Sentry::getUser()->employee->id;
371       $visits = Visit::where('status','pending')->orderBy('date_of_visit')->paginate(5);
372       return View::make('patients.pending')->with('visits',$visits);
373     }
374
375     public function outbox()
376     {
377       $user = Sentry::getUser()->employee->id;
378       //echo $user;
379       $visits = Visit::where('added_by',$user)
380         ->where('status','closed')->orderBy('date_of_visit')->paginate(5);
381       //var_dump($visits);
382
383       return View::make('patients.outbox')->with('visits',$visits);
384     }
385
386     public function waiting_clerk()
387     {
388       $user = Sentry::getUser()->employee->id;
389       $visits = Visit::
390       where('status','waiting')->orderBy('date_of_visit')->paginate(5);
391
392       return View::make('patients.outbox')->with('visits',$visits);
393     }
394
395
396     public function outbox_doctor()
397     {
398       $doctor = Sentry::getUser()->employee;
399       $visits = $doctor->visits()->where('status','closed')->orderBy('date_of_visit')
400         ->paginate(5);
401       return View::make('patients.outbox')->with('visits',$visits);
402
403     }
404
405     public function inbox()
406     {
407
408       $doctor = Sentry::getUser()->employee;
409       $visits = $doctor->visits()->where('status','pending')->where('date_of_visit','>=',new DateTime('today'))->
              orderBy('date_of_visit')->paginate(5);
410
411       return View::make('patients.inbox')->with('visits',$visits);
412     }
413
414     public function waiting()
415     {
416       $doctor = Sentry::getUser()->employee;
417       $visits = $doctor->visits()->where('status','waiting')->orderBy('date_of_visit')->paginate(5);
```

```
418
419       return View::make('patients.waiting')->with('visits',$visits);
420     }
421
422     public function waiting2()
423     {
424       $doctor = Sentry::getUser()->employee;
425       $visits = $doctor->visits()->where('status','waiting for diagnosis')->orderBy('date_of_visit')->paginate(5);
426       return View::make('patients.waiting2')->with('visits',$visits);
427     }
428
429
430
431 }
```

Listing 30: ../portal/app/controllers/SessionController.php

```
1  <?php
2
3  class SessionController extends \BaseController {
4
5
6
7    /**
8     * Show the form for creating a new resource.
9     *
10    * @return Response
11    */
12   public function create()
13   {
14     return View::make('login');
15   }
16
17   /**
18    * Store a newly created resource in storage.
19    *
20    * @return Response
21    */
22   public function store()
23   {
24         $validator = Validator::make(Input::all(), [
25             "username" => "required",
26             "password" => "required"
27         ]);
28         if ($validator->passes())
29         {
30           $credentials = array(
31         'username'=>Input::get('username'),
32         'password'=>Input::get('password')
33       );
34
35       try{
36
37         $user = Sentry::authenticate($credentials, false);
38
```

```
39
40          if($user){
41
42            return Redirect::to('landing');
43
44            if($user->inGroup(Sentry::findGroupByName('patients')))
45            {
46              return Redirect::route('mine')
47                ->with('success','You are now logged in.');
48            }
49
50            if($user->inGroup(Sentry::findGroupByName('admin')))
51            {
52              return Redirect::route('landing')
53                ->with('success','You are now logged in.');
54            }
55            if($user->inGroup(Sentry::findGroupByName('medical_technicians')))
56            {
57              return Redirect::route('landing')
58                ->with('success','You are now logged in.');
59            }
60            if($user->inGroup(Sentry::findGroupByName('hospital_admin')))
61            {
62              return Redirect::route('landing')
63                ->with('success','You are now logged in.');
64            }
65            if($user->inGroup(Sentry::findGroupByName('doctors')))
66            {
67              return Redirect::route('landing')
68                ->with('success','You are now logged in.');
69            }
70            if($user->inGroup(Sentry::findGroupByName('clerks')))
71            {
72              return Redirect::route('landing')
73                ->with('success','You are now logged in.');
74            }
75
76
77          }
78          else{
79            $errorProvider = new ErrorProvider;
80            $errorProvider->addMessage('general','Could not log in.');
81            return Redirect::route('login')->withErrors($errorProvider);
82          }
83        }
84        catch(\Exception $e){
85
86          $errorProvider = new ErrorProvider;
87          $errorProvider->addMessage('general','Could not log in.');
88          return Redirect::route('login')->withErrors($errorProvider);
89        }
90      }
91      else{
92        return Redirect::route('login')->withErrors($validator);
93
94      }
```

```
95    }
96
97    /**
98     * Remove the specified resource from storage.
99     *
100    * @param  int  $id
101    * @return Response
102    */
103   public function destroy()
104   {
105     Sentry::logout();
106     return Redirect::route('login')->with('message','Logged out');
107   }
108
109 }
```

Listing 31: ../portal/app/controllers/TestRequestController.php

```
1   <?php
2
3   class TestRequestController extends \BaseController {
4     public function __construct()
5     {
6       $this->beforeFilter('auth');
7       $this->beforeFilter('inGroup:medical_technicians');
8       $this->beforeFilter('csrf',['on'=>'post']);
9
10    }
11
12    public function inbox()
13    {
14      $deptId= Sentry::getUser()->employee->department_id;
15      $requests = TestRequest::with('type')->where('status','pending')->get();
16      $requests = $requests->filter(function($request)use($deptId)
17      {
18        if($deptId == $request->type->department_id)
19        {
20          return $request;
21        }
22      });
23
24      //paginate collection after filter? how?
25
26      return View::make('requests.inbox')
27        ->with('requests',$requests);
28    }
29
30    public function outbox()
31    {
32      $employeeId = Sentry::getUser()->employee->id;
33
34      $requests = TestRequest::with('result')->where('status','completed')->get();
35      $requests = $requests->filter(function($request)use($employeeId)
36      {
37        if($employeeId == $request->result->added_by)
```

```
38        {
39          return $request;
40        }
41      });
42
43      //paginate collection after filter? how?
44
45      return View::make('requests.outbox')->with('requests',$requests);
46
47    }
48
49  }
```

Listing 32: ../portal/app/controllers/TestResultController.php

```
1   <?php
2
3   class TestResultController extends \BaseController {
4
5
6     public function __construct()
7     {
8       $this->beforeFilter('auth');
9       $this->beforeFilter('inGroup:medical_technicians',array('only'=>array('create','store')));
10      $this->beforeFilter('showResults',array('only'=>array('show')));
11    }
12    /**
13    * Show the form for creating a new resource.
14    *
15    * @return Response
16    */
17    public function create($id)
18    {
19
20      $request = TestRequest::find($id);
21      $added_by = Sentry::getUser()->employee->id;
22
23      return View::make('results.create')->with('request',$request)->with('added_by',$added_by);
24
25    }
26
27    /**
28    * Store a newly created resource in storage.
29    *
30    * @return Response
31    */
32    public function store()
33    {
34
35      $result = new TestResult;
36      $request = TestRequest::find(Input::get('test_request_id'));
37      $result->test_request_id = $request->id;
38      $result->added_by = Employee::find(Input::get('added_by'))->id;
39
40
```

```
41      $type = json_decode($request->type);
42
43      //$fields = json_decode($request->fields,true);
44      $fieldContents = array();
45
46      $typeFields = json_decode($request->type->fields,true);
47      //var_dump($typeFields);
48
49      $rules = array();
50
51      foreach($typeFields as $key=>$field)
52      {
53        switch($field["data_type"])
54        {
55          case "image":
56          $rules[$key] = 'image|required';
57          $image = Input::file($key);
58          $filename = $request->visit->patient->id .'-'. date('YmdGis');
59          $location = public_path(). "/uploads/";
60          $success =
61            $image->move($location,$filename.'.'.              $image->getClientOriginalExtension());
62          if($success)
63          {
64
65            $new_name =  $filename.'.'.$image->getClientOriginalExtension();
66            $handle = fopen($location.$new_name,'rb');
67            $img = new Imagick();
68            $img->readImageFile($handle);
69            $img->thumbnailImage(200, 0);
70            $img->writeImage($location.$filename."-thumb.jpg");
71            $fieldContents[$key] = $new_name;
72          }
73          else{
74
75            $errorProvider = new ErrorProvider;
76            $errorProvider->addMessage('general','Error uploading image.');
77            return Redirect::back()->withErrors($errorProvider)->withInput();
78          }
79          break;
80          case "range":
81          $rules[$key] = 'numeric|required';
82
83            $fieldContents[$key] = Input::get($key);
84
85
86          break;
87          case "selection":
88          $fieldContents[$key] = Input::get($key);
89
90          if(!in_array(Input::get($key),$field["preset_values"]))
91          {
92
93            $field["preset_values"][] = Input::get($key);
94            $typeFields[$key]["preset_values"][] = Input::get($key);
95            $newf = $request->type;
96
```

```
97            $newf->fields = json_encode($typeFields);
98            $newf->save();
99
100         }
101
102         break;
103
104       }
105     }
106     $validator = Validator::make(Input::all(),$rules);
107
108     if($validator->passes())
109     {
110       $fieldContents['test_request_id'] = $request->id;
111
112       DB::table($request->type->table_name)->insert($fieldContents);
113
114       $request->status = "completed";
115       $request->save();
116
117       $result->save();
118
119       return Redirect::route('results.show',array('results'=>$result->id));
120
121     }
122     else
123     {
124       return Redirect::back()->withErrors($validator)->withInput();
125     }
126
127
128
129
130   }
131
132   /**
133   * Show the form for editing the specified resource.
134   *
135   * @param  int  $id
136   * @return Response
137   */
138   public function edit($id)
139   {
140     //
141   }
142
143   /**
144   * Update the specified resource in storage.
145   *
146   * @param  int  $id
147   * @return Response
148   */
149   public function update($id)
150   {
151     //
152   }
```

```
153
154     public function show($id)
155     {
156       $result = TestResult::find($id);
157
158       $tableName = $result->request->type->table_name;
159       $resultData = DB::table($tableName)->where('test_request_id',$result->request->id)->first();
160
161
162       return View::make('results.show')->with('result',$result)->with('resultData',$resultData);
163     }
164
165     /**
166     * Remove the specified resource from storage.
167     *
168     * @param  int  $id
169     * @return Response
170     */
171     public function destroy($id)
172     {
173       //
174     }
175
176
177
178     //TODO
179     public function search()
180     {
181       $types = Type::all()->lists('test_name');
182       $physicians = User::getDoctors();
183
184       if(Input::get())
185       {
186         //$results = TestResult::
187         if(Input::has('physician'))
188         {
189
190         }
191       }
192
193       return View::make('results.search')->with('types',$types)->with('physicians',$physicians);
194     }
195
196 }
```

Listing 33: ../portal/app/controllers/TestTypeController.php

```
1  <?php
2
3
4  use Illuminate\Database\Schema\Blueprint;
5
6  class TestTypeController extends \BaseController {
7
8    public function __construct()
```

```
9     {
10      $this->beforeFilter('auth');
11      $this->beforeFilter('csrf',['on'=>'post']);
12      $this->beforeFilter('admin',array('except'=>array('dropdown','values')));
13    }
14
15    /**
16     * Display a listing of the resource.
17     *
18     * @return Response
19     */
20    public function index()
21    {
22      $types = Type::all();
23      return View::make('types.index')->with('types',$types);
24    }
25
26    /**
27     * Show the form for creating a new resource.
28     *
29     * @return Response
30     */
31    public function create()
32    {
33      $departments = Department::where('active',1)->get();
34      return View::make("types.create")->with('departments',$departments);
35    }
36
37    /**
38     * Store a newly created resource in storage.
39     *
40     * @return Response
41     */
42    public function store()
43    {
44
45      //var_dump(Input::all());
46      //convert table name
47      $test_data_name = 'test_'.strtolower(str_replace(' ','_',preg_replace("/[^\w\d ]/ui", '', Input::get('
            test_name'))));
48
49      if(!Schema::hasTable($test_data_name)){
50        $count = Input::get('count');
51        $results = array();
52        for($i=1;$i<=$count;$i++)
53        {
54          $field_name = Input::get('field_name'.$i);
55          $data_name = strtolower(str_replace(' ','_',preg_replace("/[^\w\d ]/ui", '', $field_name)));
56          $data_type = Input::get('data_type'.$i);
57          $results[$data_name]=
58            array(
59              'field_name'=>$field_name,
60              'data_type'=>$data_type
61            );
62            switch($data_type){
63              case "image":
```

```
64              //that's all folks
65              break;
66
67              case "range":
68              if(!is_numeric(Input::get('min'.$i))||!is_numeric(Input::get('max'.$count)))
69
70              {
71                $errorProvider = new ErrorProvider();
72                $errorProvider->addMessage('general','Range minimum and maximum must be numeric values');
73              //  return Redirect::back()->withErrors($errorProvider);
74
75              }
76              $results[$data_name]['unit']
77                = Input::get('unit'.$i);
78              $results[$data_name]['comparison_value'] = Input::get('min'.$i)." - ".Input::get('max'.$i). Input::
                    get('unit'.$i);
79              break;
80              case "selection":
81
82              $results[$data_name]['preset_values'] = explode(",",Input::get('hidden-selection'.$i));
83              break;
84            }
85        }
86
87        $fields = json_encode($results);
88        //var_dump($fields);
89        $type = new Type;
90        $type->test_name = Input::get('test_name');
91        $type->department_id = Input::get('department_id');
92        $type->fields = $fields;
93        $type->table_name = $test_data_name;
94
95        Schema::create($test_data_name,function(Blueprint $table) use($results){
96          $table->increments('id');
97
98          $table->integer('test_request_id')->unsigned();
99          $table->foreign('test_request_id')->references('id')
100           ->on('test_requests')->onDelete('cascade');
101
102
103         foreach($results as $data_name => $details)
104         {
105           switch($details["data_type"])
106           {
107             case "image":
108               $table->string($data_name);
109               break;
110             case "range":
111               $table->float($data_name);
112               break;
113             case "selection":
114               $table->string($data_name);
115               break;
116           }
117         }
118       });
```

```
119         $type->save();
120
121         return Redirect::to('types/'.$type->id);
122
123       }
124       else
125       {
126         $errorProvider = new ErrorProvider();
127         $errorProvider->addMessage('general','This test already exists.');
128         return Redirect::back()->withErrors($errorProvider);
129
130
131       }
132
133     }
134
135     /**
136      * Display the specified resource.
137      *
138      * @param  int  $id
139      * @return Response
140      */
141     public function show($id)
142     {
143       $type = Type::find($id);
144       return View::make("types.show")->with('type',$type);
145     }
146
147
148
149     /**
150      * Remove the specified resource from storage.
151      *
152      * @param  int  $id
153      * @return Response
154      */
155     public function destroy($id)
156     {
157       $type = Type::find($id);
158       $table_name = $type->table_name;
159       $type->delete();
160       Schema::drop($table_name);
161
162     }
163
164     public function dropdown($id)
165     {
166       $department = Department::find($id);
167       return View::make('types.dropdown')->with('types',$department->types);
168
169     }
170
171     public function values($id,$data_name)
172     {
173       $type = Type::find($id);
174       $fields = json_decode($type->fields,true);
```

```
175       return json_encode($fields[$data_name]['preset_values']);
176     }
177
178   }
```

Listing 34: ../portal/app/controllers/UserController.php

```
1   <?php
2   use Illuminate\Support\MessageBag;
3   class UserController
4     extends Controller
5   {
6     public function __construct(){
7       // $this->beforeFilter('auth',array('except'=>array(
8   //      'requestActivation',
9   //      'sendActivation',
10  //      'activate',
11  //      'requestReset',
12  //      'sendReset',
13  //      'reset',
14  //      'resetPassword',
15  //    )));
16  //
17      $this->beforeFilter('auth',array('only'=>array('settings')));
18      $this->beforeFilter('admin',array('only'=>array('index','edit','create','disable')));
19      $this->beforeFilter('csrf',['on'=>'post']);
20
21    }
22
23    /**
24    * Display a listing of the resource.
25    *
26    * @return Response
27    */
28    public function index()
29    {
30
31        if(Input::get()){
32
33
34          $allowed = array('username','last_name','id','email');
35          $sort = in_array(Input::get('sort'),$allowed) ? Input::get('sort') : 'users.id';
36          if($sort == 'id'){
37            $sort = 'users.id';
38          }
39          if($sort== 'last_name'){
40            $sort = 'employees.last_name';
41          }
42          $order = Input::get('order') =='desc' ? 'desc': 'asc';
43          $users = User::has('employee')->
44            leftJoin('employees','users.id','=','employees.user_id');
45
46
47          if(Input::has('last_name')){
48            if(Input::has('first_name')){
```

```
49              $users = $users->where('employees.last_name',
50              'LIKE','%'.Input::get('last_name').'%')
51                ->where('employees.first_name','LIKE','%'.Input::get('first_name').'%');
52
53            }
54            else
55            {
56              $users = $users->where('employees.last_name',
57              'LIKE','%'.Input::get('last_name').'%');
58            }
59          }
60
61          $users = $users->orderBy($sort,$order);
62
63        }
64        else
65        {
66
67          $users = User::has('employee');
68          $users = $users->leftJoin('employees','users.id','=','employees.user_id');
69
70
71        }
72
73        $users = $users->paginate(5);
74
75      return View::make('users/index')->with('users',$users);
76
77    }
78
79
80
81    /**
82    * Show the form for creating a new resource.
83    *
84    * @return Response
85    */
86    public function create()
87    {
88      $departments = Department::where('active',1)->lists('department_name');
89      return View::make('users/create')->with('departments',$departments);
90    }
91
92    /**
93    * Store a newly created resource in storage.
94    *
95    * @return Response
96    */
97    public function store()
98    {
99      $addedBy = Sentry::getUser()->employee->id;
100     $validator = Validator::make(Input::all(),array(
101       "email" => "required|email|unique:users",
102       "last_name" => "required",
103       "first_name" => "required",
104       "account_type" => "in:admin,clerks,medical_technicians,doctors,hospital_admin"
```

```
105       ));
106       if($validator->passes())
107       {
108
109       //  $addedBy = Sentry::getUser()->employee;//->employee->employee_id;
110
111         $email = Input::get('email');
112         $accountType = Input::get('account_type');
113         $lastName = Input::get('last_name');
114         $firstName = Input::get('first_name');
115         $password = User::generatePassword();
116         $user = User::registerWithOptions($email,$password,$accountType,$lastName,$firstName);
117
118
119         $employeeRecord = new Employee;
120         $employeeRecord->user_id = $user->id;
121         $employeeRecord->first_name = Input::get('first_name');
122         $employeeRecord->last_name = Input::get('last_name');
123         $employeeRecord->added_by = $addedBy;
124         $employeeRecord->department_id = Input::get('department');
125         $employeeRecord->save();
126
127         return View::make('users.created',
128           array(
129             'username'=>$user->username,
130             'password'=>$password,
131             'email'=>$user->email,
132             'id'=>$user->id,
133           ));
134
135
136       }
137       else
138       {
139         return Redirect::route('users.create')->withInput()->withErrors($validator);
140       }
141     }
142
143
144
145     /**
146      * Shows the form for editing the specified resource
147      *
148      * @return response
149      */
150     public function edit($id){
151       $user = Sentry::findUserById($id);
152        return View::make('users/edit',array('user'=>$user));
153     }
154
155     /**
156     * Display the specified resource.
157     *
158     * @param  int  $id
159     * @return Response
160     */
```

```
161     public function show($id)
162     {
163       $user = Sentry::findUserById($id);
164
165       return View::make('users/show',array('user'=>$user));
166     }
167
168
169
170     /**
171     * Show the form for editing
172     *
173     * @param  int  $id
174     * @return Response
175     */
176     public function settings()
177     {
178       return View::make('users/settings');
179
180     }
181
182     /**
183     * Update the specified resource in storage.
184     *
185     * @param  int  $id
186     * @return Response
187     */
188     public function update($id)
189     {
190       $validator = Validator::make(Input::all(),array(
191         'last_name' => 'required',
192         'first_name' => 'required',
193         'email' => 'required|email'
194       ));
195
196       if($validator->passes())
197       {
198
199         $user = Sentry::findUserById($id);
200         if($user->inGroup(Sentry::findGroupByName('patients')) )
201         {
202           $user->patient->last_name = Input::get('last_name');
203           $user->patient->first_name = Input::get('first_name');
204           $user->patient->email = Input::get('email');
205           $user->email = Input::get('email');
206           $user->push();
207         }
208         else
209         {
210           $user->employee->last_name = Input::get('last_name');
211           $user->employee->first_name = Input::get('first_name');
212           $user->email = Input::get('email');
213           $user->push();
214         }
215
216         return Redirect::to('users/'.$user->id)->with('success','Account successfully edited.');
```

```
217
218       }
219       else
220       {
221         return Redirect::to('users'.$user->id.'/edit')->withInput()->withErrors($validator);
222       }
223     }
224
225     /**
226     * Remove the specified resource from storage.
227     *
228     * @param  int  $id
229     * @return Response
230     */
231     public function destroy($id)
232     {
233       $user = Sentry::findUserById($id);
234       $user->delete();
235       return Redirect::route('users.index')->with('success','User account successfully deleted.');
236     }
237
238     /**
239      * Form for resending re-activation email
240      *
241      * @return response
242      */
243     public function requestActivation()
244     {
245       return View::make('users/resend');
246     }
247
248     /**
249      * Send activation email
250      *
251      * @return response
252      */
253     public function sendActivation()
254     {
255       $validator = Validator::make(Input::all(),array(
256         "email" =>"required|exists:users|email"
257       ));
258       if($validator->passes())
259       {
260         $user = User::where('email',Input::get('email'))->first();
261         try{
262           $activationCode = $user->getActivationCode();
263         }catch (Exception $e){
264           return Redirect::route('resend')->withInput()
265             ->with('success','That user has already been activated.');
266         }
267         $mailData = array(
268           "code"=>$user->getActivationCode(),
269           "id"=>$user->id,
270
271         );
272         $email = $user->email;
```

```
273
274
275         Mail::send('emails.activate',$mailData,function($message) use($email){
276           $message->to($email)->subject('Activate your account');
277         });
278         return Redirect::route('login')->withInput()
279           ->with('success','Mail sent. Please click on the validation
280             link in your email to continue.');
281
282       }
283       else
284       {
285         return Redirect::route('resend')->withInput()->withErrors($validator);
286       }
287     }
288
289     /**
290     * Given user id and activation code, attempt to activate
291     *
292     * @param int $id
293     * @param int $code
294     * @return Response
295     */
296     public function activate($id, $code)
297     {
298       try{
299         $user = Sentry::findUserById($id);
300         if($user->attemptActivation($code)
301         ){
302           //success
303           Sentry::login($user,false);
304           return Redirect::route('login')
305             ->with('success','Account has been successfully activated! You may now log in.');
306         }
307         else
308         {
309           $errorProvider = new ErrorProvider;
310           $errorProvider->addMessage('general','Unable to activate user.');
311           //failure
312           return Redirect::route('login')->withErrors($errorProvider);
313         }
314       }
315       catch (Cartalyst\Sentry\Users\UserNotFoundException $e)
316       {
317         $errorProvider = new ErrorProvider;
318         $errorProvider->addMessage('general','User not found.');
319         return Redirect::route('login')->withErrors($errorProvider);
320       }
321       catch (Cartalyst\Sentry\Users\UserAlreadyActivatedException $e)
322       {
323
324         return Redirect::route('login')->with('success','This user account
325           has already been activated. You may now sign in.');
326       }
327     }
328
```

```
329     /**
330      * Form for password reset request
331      *
332      * @return response
333      */
334     public function requestReset()
335     {
336       return View::make('users/forgot');
337     }
338
339     /**
340      * Send password reset email
341      *
342      * @return response
343      */
344     public function sendReset()
345     {
346       $validator = Validator::make(Input::all(),array(
347         "email" =>"required|exists:users"
348       ));
349
350       if($validator->passes())
351       {
352         $user = User::where('email',Input::get('email'))->first();
353
354         $mailData = array(
355           "id"=>$user->id,
356           "code"=>$user->getResetPasswordCode()
357
358         );
359         $email = $user->email;
360
361         Mail::send('emails.auth.reminder',$mailData,function($message) use($email){
362           $message->to($email)->subject('Reset Password');
363
364         });
365         return Redirect::route('login')->withInput()
366           ->with('success','Mail sent. Please click on the reset
367             link in your email to continue.');
368
369       }else{
370         return Redirect::route('users/forgot')->withInput()->withErrors($validator);
371       }
372     }
373
374     /**
375      * Form for resetting password
376      *
377      * @param string $id
378      * @param string $code
379      */
380     public function reset($id,$code)
381     {
382
383       return View::make('users.reset',array("id"=>$id,"code"=>$code));
384
```

```
385     }
386     public function changeEmail(){
387       $validator = Validator::make(Input::all(),array(
388         "password" => "required",
389         "new_email" => "required|email"
390       ));
391       if($validator->passes())
392       {
393         $user = Sentry::getUser();
394         if($user->checkPassword(Input::get('password')))
395         {
396           //send a confirmation to the new email
397           $emailCode = $user->getEmailCode();
398           $email = Input::get('new_email');
399
400           $user->temp_email = Input::get('new_email');
401           $user->save();
402           $mailData = array(
403             "id"=>$user->id,
404             "code"=>$emailCode,
405           );
406
407
408           Mail::send('emails.auth.confirm',$mailData,function($message) use ($email)
409           {
410             $message->to($email)->subject('Confirm email address change');
411           });
412           return Redirect::route('settings')
413             ->with('success','An email has been sent to '.$email.'.
414               Please click on the validation link in your email to confirm the change.');
415         }
416         else
417         {
418           $errorProvider = new ErrorProvider;
419           $errorProvider->addMessage('password','Password is invalid.');
420           return Redirect::route('settings')->withErrors($errorProvider)->withInput();
421         }
422       }
423       else
424       {
425         return Redirect::route('settings')->withErrors($validator)->withInput();
426       }
427     }
428     /**
429      * Upon opening the confirmation link in the email
430      *
431      * @return response
432      */
433     public function confirmChangeEmail($id,$code)
434     {
435       try{
436         $user = Sentry::findUserById($id);
437         if($user->attemptChangeEmail($code))
438         {
439           //celebrate!
440           return Redirect::route('settings')
```

```
441             ->with('success','Successfully changed email.');
442         }
443         else
444         {
445           $errorProvider = new ErrorProvider;
446           $errorProvider->addMessage('general','Unable to change email');
447           return Redirect::route('settings')
448             ->withErrors($errorProvider);
449         }
450       }catch(Exception $e){
451         $errorProvider = new ErrorProvider;
452         $errorProvider->addMessage('general','User not found.');
453         return Redirect::route('login')->withErrors($errorProvider);
454       }
455     }
456     /**
457      *  changing password
458      *
459      * @return response
460      */
461     public function changePassword()
462     {
463       $validator = Validator::make(Input::all(),array(
464         "old_password" => 'required',
465         "new_password" => "required|min:6",
466         "password_confirmation" => "same:new_password",
467       ));
468       if($validator->passes())
469       {
470         $user = Sentry::getUser();
471         if($user->checkPassword(Input::get('old_password'))){
472           $user->password = Input::get('new_password');
473
474           if($user->save()){
475
476             //great success
477             return Redirect::route('settings')
478               ->with('success','Password successfully changed.');
479           }
480           else
481           {
482             //failure! :(
483             $errorProvider = new ErrorProvider;
484             $errorProvider->addMessage('general','Unable to change password');
485             return Redirect::route('settings')->withErrors($errorProvider);
486           }
487         }
488         else
489         {
490           //old password is wrrrrrong
491           $errorProvider = new ErrorProvider;
492           $errorProvider->addMessage('old_password','Password is invalid.');
493           return Redirect::route('settings')->withErrors($errorProvider);
494         }
495       }
496       else
```

```
497       {
498         //something wrong with your input
499         return Redirect::route('settings')->withErrors($validator);
500       }
501     }
502
503     /**
504      * Reset password
505      *
506      * @return response
507      */
508     public function resetPassword()
509     {
510       $validator = Validator::make(Input::all(),array(
511         "password" => "required|min:6",
512         "password_confirmation" => "same:password",
513       ));
514       if($validator->passes())
515       {
516         $user = Sentry::findUserById(Input::get('id'));
517         $code = Input::get('code');
518         $password = Input::get('password');
519         if($user->checkResetPasswordCode($code))
520         {
521           if($user->attemptResetPassword($code,$password))
522           {
523             return Redirect::route('login')
524               ->with('success','Success! You may now log in with your new password.');
525           }
526           else
527           {
528             //password reset failed
529             $errorProvider = new ErrorProvider;
530             $errorProvider->addMessage('general','Unable to reset password.');
531             return Redirect::route('login')->withErrors($errorProvider);
532           }
533         }
534         else
535         {
536           //code is wrong
537           $errorProvider = new ErrorProvider;
538           $errorProvider->addMessage('general','Password reset code is invalid.');
539           return Redirect::route('login')->withErrors($errorProvider);
540         }
541       }
542       else
543       {
544         return Redirect::route('reset',
545         array("id"=>Input::get('id'),"code"=>Input::get('code')))->withErrors($validator);
546       }
547     }
548
549     public function disable($id)
550     {
551       try
552       {
```

```
553           // Find the user using the user id
554           $throttle = Sentry::findThrottlerByUserId($id);
555
556           // Ban the user
557           $throttle->ban();
558
559         return Redirect::back();
560
561       }
562       catch (Cartalyst\Sentry\Users\UserNotFoundException $e)
563       {
564
565       }
566     }
567
568     public function profile(){
569       return "";
570     }
571
572
573
574 }
```

Listing 35: ../portal/app/controllers/VisitController.php

```
1  <?php
2  /**
3   * Resource controller for handling visits
4   *
5   * @package default
6   */
7
8  class VisitController extends \BaseController {
9
10   public function __construct()
11   {
12     $this->beforeFilter('auth');
13     $this->beforeFilter('csrf',array('on'=>'post'));
14     $this->beforeFilter('hasAccess:create_patient',array('on'=>'create','store','edit','update','show','delete'))
           ;
15
16     $this->beforeFilter('inGroup:doctors',array('on'=>'diagnose'));
17     $this->beforeFilter('inGroup:patients',array('only'=>array('mine')));
18
19   }
20
21   /**
22    * Show the form for creating a new resource.
23    *
24    * @return Response
25    */
26   public function create($id)
27   {
28
29
```

```
30      $patient = Patient::find($id);
31
32      return View::make('visits.create')->with('patient',$patient)->with('doctors',User::getDoctors());
33
34    }
35
36    /**
37     * Store a newly created resource in storage.
38     *
39     * @return Response
40     */
41    public function store()
42    {
43        $validator = Validator::make(Input::all(),
44      [
45        "id"=>"required",
46        "date_of_visit"=>'required'
47      ]);
48
49      if($validator->passes())
50      {
51        $visit = new Visit;
52        $visit->date_of_visit = Input::get('date_of_visit');
53
54        $visit->height = Input::get('height');
55        $visit->weight = Input::get('weight');
56        $visit->c_r = Input::get('c_r');
57        $visit->r_r = Input::get('r_r');
58        $visit->temperature = Input::get('temperature');
59        $visit->bp = Input::get('bp');
60        $visit->added_by = Sentry::getUser()->employee->id;
61
62        $patient = Patient::find(Input::get('id'));
63        $visit = $patient->visits()->save($visit);
64
65        if(Input::has('attending_physicians'))
66        {
67          $attending_physicians = Input::get('attending_physicians');
68          foreach($attending_physicians as $attending_physician)
69          {
70            $visit->attendingPhysicians()->attach($attending_physician);
71          }
72        }
73
74        return Redirect::to('patients/'.Input::get('id'))
75          ->with('success','Patient visit successfully added');
76      }
77      else
78      {
79
80
81        return Redirect::route('visits.create')->with('patients', Input::get('id'))->withErrors($validator);
82
83      }
84    }
85
```

```
86    /**
87     * Display the specified resource.
88     *
89     * @param  int  $id
90     * @return Response
91     */
92    public function show($id)
93    {
94      $visit = Visit::find($id);
95      return View::make('visits/show')->with('visit',$visit);
96    }
97
98    /**
99     * Show the form for editing the specified resource.
100    *
101    * @param  int  $id
102    * @return Response
103    */
104   public function edit($id)
105   {
106     $group = Sentry::findGroupByName('doctors');
107
108     $doctors = Sentry::findAllUsersInGroup($group);
109     $lists = [];
110
111
112
113     $doctors->each(function($doctor){
114       $doctor->load('employee');
115     });
116     $doctors = $doctors->toArray();
117
118     foreach($doctors as $doctor){
119       $lists[$doctor["employee"]["id"]] = $doctor["employee"]["last_name"].", ".$doctor["employee"]["first_name"
              ];
120     }
121     $visit = Visit::find($id);
122     return View::make('visits/edit')->with('visit',$visit)->with('doctors',$lists);
123   }
124
125   /**
126    * Update the specified resource in storage.
127    *
128    * @param  int  $id
129    * @return Response
130    */
131   public function update($id)
132   {
133       $validator = Validator::make(Input::all(),
134     [
135       "date_of_visit"=>"required"
136
137     ]);
138
139       if($validator->passes()){
140         $visit = Visit::find($id);
```

```
141           $visit->date_admitted = Input::get('date_admitted');
142           $visit->date_discharged = Input::get('date_discharged');
143           if(Input::has('is_outpatient'))
144             $visit->is_outpatient = Input::get('is_outpatient');
145           $visit->height = Input::get('height');
146           $visit->weight = Input::get('weight');
147           $visit->c_r = Input::get('c_r');
148           $visit->r_r = Input::get('r_r');
149           $visit->temperature = Input::get('temperature');
150           $visit->bp = Input::get('bp');
151           $visit->added_by = Input::get('employee_id');
152           $visit->admitting_diagnosis = Input::get('admitting_diagnosis');
153           $visit->chief_complaint = Input::get('chief_complaint');
154
155
156           $visit->save();
157
158           return Redirect::to('visits/'.$id)
159             ->with('success','Successfully edited visit record!');
160         }
161         else{
162           return Redirect::to('visits/'.$id.'/edit')
163             ->withErrors($validator)
164               ->withInput();
165         }
166
167     }
168
169     /**
170      * Remove the specified resource from storage.
171      *
172      * @param  int  $id
173      * @return Response
174      */
175     public function destroy($id)
176     {
177       $visit = Visit::find($id);
178       $patientId = $visit->patient->id;
179       $visit->delete();
180       return Redirect::to('pending')->with('success','Visit deleted.');
181
182     }
183
184     public function diagnose($id)
185     {
186
187       if(Input::get())
188       {
189         //var_dump(Input::get('requests'));
190
191         if(Input::has('requests')){
192         $tests = Input::get('requests');
193         foreach ($tests as $test)
194         {
195           $request = new TestRequest;
196           $request->test_type = $test;
```

```
197           $request->visit_id = Input::get('visit_id');
198           $request->fields = json_encode(Input::get(Type::find($test)->table_name));
199           $request->save();
200
201
202
203         }
204         $visit = Visit::find($id);
205
206         $visit->admitting_diagnosis = Input::get('admitting_diagnosis');
207         $visit->chief_complaint = Input::get('chief_complaint');
208
209         $visit->status = 'waiting';
210         $visit->save();
211
212         return Redirect::route('inbox');
213
214          }
215       $visit = Visit::find($id);
216
217       $visit->admitting_diagnosis = Input::get('admitting_diagnosis');
218       $visit->chief_complaint = Input::get('chief_complaint');
219
220       $visit->status = 'waiting';
221       $visit->save();
222
223       if(Input::has('final_diagnosis'))
224       {
225         $visit->final_diagnosis = Input::get('final_diagnosis');
226         $visit->status = 'closed';
227         $visit->save();
228         return Redirect::route('inbox');
229       }
230       }
231         $visit = Visit::find($id);
232
233         $departments = Department::where('active',1)->get();
234          $tests = array();
235
236         return View::make('visits.diagnose')->with('visit',$visit)->with('departments',$departments);
237
238     }
239
240     public function mine()
241     {
242       $user = Sentry::getUser()->patient;
243       $visits = $user->visits()->orderBy('date_of_visit')->paginate(5);
244       return View::make('visits.mine')->with('visits',$visits);
245     }
246     public function status($id)
247     {
248       $visit = Visit::find($id);
249       $visit->status = 'waiting for diagnosis';
250       $visit->save();
251
252       return Redirect::route('visits.show',$visit->id);
```

```
253     }
254
255 }
```

Listing 36: ../portal/app/database/migrations/2013_11_23_212921_create_users_table.php

```
1   <?php
2
3   use Illuminate\Database\Schema\Blueprint;
4   use Illuminate\Database\Migrations\Migration;
5
6   class CreateUsersTable extends Migration {
7
8     /**
9     * Run the migrations.
10    *
11    * @return void
12    */
13    public function up()
14    {
15
16      Schema::table('users', function(Blueprint $table)
17      {
18
19        $table->dropColumn('email');
20        $table->dropColumn('first_name','last_name');
21
22
23      });
24
25      Schema::table('users', function(Blueprint $table)
26      {
27
28
29        $table->string('username')->after('id')->unique();
30        $table->string('email')->after('username')->nullable()->unique();
31        $table->string('email_code')->nullable();
32        $table->string('temp_email')->nullable();
33
34        //default created by sentry but unnecessary here
35
36
37      });
38    }
39
40    /**
41    * Reverse the migrations.
42    *
43    * @return void
44    */
45    public function down()
46    {
47      Schema::table('users', function($table)
48      {
49        $table->dropColumn('username','email_code','temp_email');
```

```
50      });
51    }
52
53  }
```

Listing 37: ../portal/app/database/migrations/2013_11_23_213040_create_patient_record_table.php

```
1   <?php
2
3   use Illuminate\Database\Schema\Blueprint;
4   use Illuminate\Database\Migrations\Migration;
5
6   class CreatePatientRecordTable extends Migration {
7
8     /**
9      * Run the migrations.
10     *
11     * @return void
12     */
13    public function up()
14    {
15      //
16      Schema::create('patients', function(Blueprint $table)
17      {
18
19        $table->engine = 'InnoDB';
20
21        //patient_id PK
22        $table->increments('id');
23        $table->string('hospital_id',45)->unique();
24
25        //user_id foreign key
26        $table->integer('user_id')->unsigned()->unique()->nullable();
27        $table->foreign('user_id')->references('id')->on('users')
28          ->onUpdate('cascade')->onDelete('restrict');
29
30        $table->string('first_name',45);
31        $table->string('middle_name',45)->nullable();
32        $table->string('last_name',45);
33
34        $table->date('birthdate');
35        $table->string('birthplace',45)->nullable();
36        $table->string('email')->nullable();
37
38        $table->enum('sex',array('male','female'));
39        $table->enum('civil_status',array('single','married','widowed'))
40          ->nullable();
41        $table->string('telephone_number',10)->nullable();
42        $table->string('mobile_number',15)->nullable();
43        $table->boolean('is_dependent')->default(0);
44        $table->boolean('is_stockholder')->default(0);
45        $table->boolean('is_stockholder_dependent')->default(0);
46
47        $table->string('address')->nullable();
48        $table->string('religion',45)->nullable();
```

```
49          $table->string('nationality',45)->nullable();
50          $table->string('occupation',45)->nullable();
51          $table->string('company_name',45)->nullable();
52          $table->string('company_address')->nullable();
53
54          $table->string('father_name',45)->nullable();
55          $table->string('mother_name',45)->nullable();
56          $table->string('parent_address')->nullable();
57
58          $table->string('spouse_name',45)->nullable();
59          $table->string('spouse_occupation',45)->nullable();
60          $table->string('spouse_company',45)->nullable();
61
62          $table->string('emergency_contact_name',45)->nullable();
63          $table->string('emergency_contact_address')->nullable();
64          $table->string('emergency_contact_number',45)->nullable();
65
66          $table->string('account_responsibility_name',45)->nullable();
67          $table->string('account_responsibility_address')->nullable();
68          $table->string('account_responsibility_number',45)->nullable();
69          $table->string('account_responsibility_relationship',45)
70              ->nullable();
71
72
73
74          $table->timestamps();
75
76      });
77    }
78
79
80    /**
81     * Reverse the migrations.
82     *
83     * @return void
84     */
85    public function down()
86    {
87
88      Schema::dropIfExists('patients');
89    }
90
91  }
```

Listing 38: ../portal/app/database/migrations/2013_11_24_135908_create_visit_table.php

```
1   <?php
2
3   use Illuminate\Database\Migrations\Migration;
4   use Illuminate\Database\Schema\Blueprint;
5
6
7   class CreateVisitTable extends Migration {
8
9     /**
```

```
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16       Schema::create('visits', function(Blueprint $table)
17       {
18
19
20         $table->engine = 'InnoDB';
21
22         $table->increments('id');
23
24
25
26
27         //user_id foreign key
28         $table->integer('patient_id')->unsigned()->nullable();;
29         $table->foreign('patient_id')->references('id')->on('patients')
30           ->onUpdate('CASCADE')->onDelete('CASCADE');
31
32         $table->datetime('date_of_visit');
33         $table->string('chief_complaint')->nullable();
34         $table->string('admitting_diagnosis')->nullable();
35         $table->string('final_diagnosis')->nullable();
36
37         $table->string('bp',10)->nullable();
38         $table->string('temperature',10)->nullable();
39         $table->string('c_r',10)->nullable();
40         $table->string('r_r',10)->nullable();
41         $table->string('weight',10)->nullable();
42         $table->string('height',10)->nullable();
43
44
45         $table->timestamps();
46
47       });
48     }
49
50     /**
51      * Reverse the migrations.
52      *
53      * @return void
54      */
55     public function down()
56     {
57
58       if(Schema::hasTable('visits')){
59         Schema::table('visits', function($table)
60         {
61             $table->dropForeign('visits_patient_id_foreign');
62         });
63       }
64       Schema::dropIfExists('visits');
65     }
```

```
66
67  }
```

Listing 39: ../portal/app/database/migrations/2013_11_27_152807_create_employee_record_table.ph

```
1   <?php
2
3   use Illuminate\Database\Migrations\Migration;
4   use Illuminate\Database\Schema\Blueprint;
5
6
7   class CreateEmployeeRecordTable extends Migration {
8
9     /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14    public function up()
15    {
16      Schema::create('employees', function(Blueprint $table)
17      {
18
19        $table->engine = 'InnoDB';
20
21        $table->increments('id');
22
23
24        //user_id foreign key
25        $table->integer('user_id')->unsigned()->unique();
26        $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
27
28        $table->string('first_name',45);
29        $table->string('last_name',45);
30
31        $table->timestamps();
32
33
34      });
35    }
36
37    /**
38     * Reverse the migrations.
39     *
40     * @return void
41     */
42    public function down()
43    {
44      if(Schema::hasTable('visits')){
45        Schema::table('visits', function($table)
46        {
47          $table->dropForeign('visits_added_by_foreign');
48        });
49      }
50
```

```
51      if(Schema::hasTable('patients')){
52        Schema::table('patients', function($table)
53        {
54            $table->dropForeign('patients_user_id_foreign');
55          $table->dropForeign('patients_added_by_foreign');
56        });
57      }
58
59      if(Schema::hasTable('employees')){
60        Schema::table('employees', function($table)
61        {
62            $table->dropForeign('employees_user_id_foreign');
63            $table->dropForeign('employees_added_by_foreign');
64
65        });
66      }
67
68      Schema::dropIfExists('employees');
69    }
70
71  }
```

Listing 40: ../portal/app/database/migrations/2013_11_27_152906_create_foreign_keys.php

```
1   <?php
2
3   use Illuminate\Database\Migrations\Migration;
4   use Illuminate\Database\Schema\Blueprint;
5
6
7   class CreateForeignKeys extends Migration {
8
9     /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14    public function up()
15    {
16      Schema::table('patients', function(Blueprint $table)
17      {
18        $table->integer('added_by')->unsigned();
19        $table->foreign('added_by')->references('id')->on('employees')->onDelete('no action');
20      });
21
22
23
24      Schema::table('employees', function(Blueprint $table)
25      {
26        $table->integer('added_by')->unsigned()->nullable();
27        $table->foreign('added_by')->references('id')->on('employees')->onDelete('no action');
28      });
29
30      Schema::table('visits', function(Blueprint $table)
31      {
```

```
32        $table->integer('added_by')->unsigned();
33        $table->foreign('added_by')->references('id')->on('employees')->onDelete('no action');
34      });
35    }
36
37    /**
38     * Reverse the migrations.
39     *
40     * @return void
41     */
42    public function down()
43    {
44      //
45    }
46
47  }
```

Listing 41: ../portal/app/database/migrations/2013_11_27_155515_create_attending_physicians_tabl

```
1   <?php
2
3   use Illuminate\Database\Migrations\Migration;
4   use Illuminate\Database\Schema\Blueprint;
5
6   class CreateAttendingPhysiciansTable extends Migration {
7
8     /**
9      * Run the migrations.
10     *
11     * @return void
12     */
13    public function up()
14    {
15      Schema::create('attending_physicians', function(Blueprint $table)
16      {
17         $table->increments("id");
18
19        $table->integer('doctor_id')->unsigned();
20        $table->foreign('doctor_id')->references('id')->on('employees')->onDelete('cascade');
21        $table->integer('visit_id')->unsigned();
22        $table->foreign('visit_id')->references('id')->on('visits')->onDelete('cascade');
23        $table->unique(array("doctor_id","visit_id"));
24      });
25    }
26
27    /**
28     * Reverse the migrations.
29     *
30     * @return void
31     */
32    public function down()
33    {
34      if(Schema::hasTable('attending_physicians')){
35        Schema::table('attending_physicians', function($table)
36        {
```

```
37            $table->dropForeign('attending_physicians_doctor_id_foreign');
38            $table->dropForeign('attending_physicians_visit_id_foreign');
39
40        });
41      }
42      Schema::dropIfExists('attending_physicians');
43    }
44
45  }
```

Listing 42: ../portal/app/database/migrations/2014_02_19_122937_create_departments_table.php

```
1   <?php
2
3   use Illuminate\Database\Migrations\Migration;
4   use Illuminate\Database\Schema\Blueprint;
5
6
7   class CreateDepartmentsTable extends Migration {
8
9     /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14    public function up()
15    {
16
17      Schema::create('departments',function(Blueprint $table)
18      {
19
20        $table->increments('id');
21        $table->string('department_name');
22        $table->timestamps();
23        $table->boolean('active')->default(0);
24      });
25
26
27
28    }
29
30    /**
31     * Reverse the migrations.
32     *
33     * @return void
34     */
35    public function down()
36    {
37      Schema::dropIfExists('departments');
38
39    }
40
41  }
```

Listing 43: ../portal/app/database/migrations/2014_02_19_123146_create_test_type_table.php

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;

class CreateTestTypeTable extends Migration {

  /**
   * Run the migrations.
   *
   * @return void
   */
  public function up()
  {


    Schema::create('test_types',function(Blueprint $table)
    {
      $table->increments('id');
      $table->string('test_name');
      $table->integer('department_id')->unsigned();
      $table->foreign('department_id')->references('id')->on('departments');
      $table->longtext('fields');
      $table->string('table_name');

    });


  }

  /**
   * Reverse the migrations.
   *
   * @return void
   */
  public function down()
  {

    Schema::dropIfExists('test_types');

  }

}
```

Listing 44: ../portal/app/database/migrations/2014_02_19_124035_add_departments_table_to_emplo

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;


class AddDepartmentsTableToEmployeeRecords extends Migration {
```

```
8
9     /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14    public function up()
15    {
16      Schema::table('employees',function(Blueprint $table){
17        $table->integer('department_id')->unsigned()->nullable();
18        $table->foreign('department_id')->references('id')->on('departments');
19
20      });
21    }
22
23    /**
24     * Reverse the migrations.
25     *
26     * @return void
27     */
28    public function down()
29    {
30      //
31      Schema::table('employees',function(Blueprint $table){
32          $table->dropForeign('employees_department_id_foreign');
33
34      });
35
36
37    }
38
39  }
```

Listing 45: ../portal/app/database/migrations/2014_02_20_165523_modify_visits_table.php

```
1   <?php
2
3   use Illuminate\Database\Migrations\Migration;
4   use Illuminate\Database\Schema\Blueprint;
5
6
7   class ModifyVisitsTable extends Migration {
8
9     /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14    public function up()
15    {
16      Schema::table('visits', function(Blueprint $table){
17        $table->enum('status',array('pending','waiting','closed','waiting for diagnosis'))->default('pending');
18      });
19
20    }
```

```
21
22    /**
23     * Reverse the migrations.
24     *
25     * @return void
26     */
27    public function down()
28    {
29      Schema::table('visits', function(Blueprint $table){
30        $table->dropColumn('status');
31      });
32    }
33
34  }
```

Listing 46: ../portal/app/database/migrations/2014_02_21_104454_create_test_requests_table.php

```
1   <?php
2
3   use Illuminate\Database\Migrations\Migration;
4   use Illuminate\Database\Schema\Blueprint;
5
6
7   class CreateTestRequestsTable extends Migration {
8
9     /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14    public function up()
15    {
16      Schema::create('test_requests',function(Blueprint $table)
17      {
18        $table->increments('id');
19
20        $table->integer('visit_id')->unsigned();
21        $table->foreign('visit_id')->references('id')->on('visits');
22
23        $table->integer('test_type')->unsigned();
24        $table->foreign('test_type')->references('id')->on('test_types');
25
26        $table->enum('status',array('pending','completed'))->default('pending');
27        $table->string('fields',1000)->nullable();
28        $table->timestamps();
29      });
30    }
31
32    /**
33     * Reverse the migrations.
34     *
35     * @return void
36     */
37    public function down()
38    {
```

```
39      Schema::dropIfExists('test_requests');
40    }
41
42  }
```

Listing 47: ../portal/app/database/migrations/2014_02_27_162719_create_test_results_table.php

```
1   <?php
2
3   use Illuminate\Database\Migrations\Migration;
4   use Illuminate\Database\Schema\Blueprint;
5
6   class CreateTestResultsTable extends Migration {
7
8     /**
9      * Run the migrations.
10     *
11     * @return void
12     */
13    public function up()
14    {
15      Schema::create('test_results', function(Blueprint $table)
16      {
17        $table->increments('id');
18        $table->timestamps();
19        $table->integer('test_request_id')->unsigned();
20        $table->unique('test_request_id');
21        $table->foreign('test_request_id')->references('id')->on('test_requests');
22
23        $table->integer('added_by')->unsigned();
24        $table->foreign('added_by')->references('id')->on('employees');
25      });
26    }
27
28    /**
29     * Reverse the migrations.
30     *
31     * @return void
32     */
33    public function down()
34    {
35      Schema::dropIfExists('test_results');
36    }
37
38  }
```

Listing 48: ../portal/app/helpers/ArrayPaginationBootstrapPresenter.php

```
1   <?
2   class ArrayPaginationBootstrapPresenter extends Illuminate\Pagination\BootstrapPresenter{
3     protected $queries;
4     protected $route;
5
6     public function __construct($currentPage,$total,$route,$queries){
```

```
7       $this->lastPage = $total;
8       $this->currentPage = $currentPage;
9       $this->queries = $queries;
10      $this->route = $route;
11    }
12    public function getUrl($page){
13      $parameters = array(
14        'page'=>$page,
15      );
16      if(count($this->queries)>0){
17        $parameters = array_merge($parameters,$this->queries);
18      }
19
20      return $this->route.'?'.http_build_query($parameters,null,'&');
21    }
22
23    public function getPrevious($text = '&laquo;'){
24      // If the current page is less than or equal to one, it means we can't go any
25      // further back in the pages, so we will render a disabled previous button
26      // when that is the case. Otherwise, we will give it an active "status".
27      if ($this->currentPage <= 1)
28      {
29        return '<li class="disabled"><span>'.$text.'</span></li>';
30      }
31      else
32      {
33        $url = $this->getUrl($this->currentPage-1);
34
35        return '<li><a href="'.$url.'">'.$text.'</a></li>';
36      }
37    }
38
39    public function getNext($text = '&raquo;')
40    {
41      // If the current page is greater than or equal to the last page, it means we
42      // can't go any further into the pages, as we're already on this last page
43      // that is available, so we will make it the "next" link style disabled.
44      if ($this->currentPage >= $this->lastPage)
45      {
46        return '<li class="disabled"><span>'.$text.'</span></li>';
47      }
48      else
49      {
50        $url = $this->getUrl($this->currentPage + 1);
51
52        return '<li><a href="'.$url.'">'.$text.'</a></li>';
53      }
54    }
55
56    public function getLink($page)
57    {
58      $url = $this->getUrl($page);
59
60      return '<li><a href="'.$url.'">'.$page.'</a></li>';
61    }
62  }
```

Listing 49: ../portal/app/helpers/ColumnPresenter.php

```
<?php

/**
 * Creates links for sortable column headers
 *
 * @package default
 */
class ColumnPresenter{
  protected $name;
  protected $title;
  protected $queries;
  protected $route;
  protected $order;
  protected $sort;

  public function __construct($name,$title,$route,$queries)
  {
    $this->name = $name;
    $this->title = $title;
    $this->queries = $queries;
    $this->route = $route;
  }

  public static function show($name,$title,$route,$queries){
    $presenter = new ColumnPresenter($name,$title,$route,$queries);
    return $presenter->render();
  }

  public function render()
  {
    if(isset($this->queries['sort']))
    {
      if($this->queries['sort']==$this->name)
      {
        $this->order = 'asc';
        $this->sort = $this->name;
        if(array_key_exists('order',$this->queries))
        {
          $this->order = $this->queries['order'] == 'desc' ? 'asc': 'desc';
        }
        else
        {

          $this->order = 'desc';
        }
      }

    }



    return '<a href="'.$this->getUrl() .'">'. $this->title . "</a>" . $this->getChevron();
  }
  public function getChevron()
```

```
55    {
56      if(array_key_exists('sort',$this->queries)
57      &&$this->queries['sort']==$this->name){
58        if($this->order == 'asc')
59        {
60          return " <span class=\"glyphicon glyphicon-chevron-up\"></span>";
61        }
62        else{
63          return " <span class=\"glyphicon glyphicon-chevron-down\"></span>";
64        }
65      }
66      else
67      {
68        return '';
69      }
70    }
71
72    public function getUrl()
73    {
74      $parameters = array(
75        'sort'=>$this->name,
76        'order'=>$this->order
77      );
78      if(count($this->queries)>0){
79        $parameters = array_merge($parameters,array_except($this->queries,array('sort','order','page')));
80      }
81
82      return $this->route.'?'.http_build_query($parameters,null,'&');
83
84    }
85
86  }
```

Listing 50: ../portal/app/helpers/composers.php

```
1  <?/**
2   * View composers
3   *
4   * @author Fiona Morella
5   */
6
7  //fix to make paginator work with sortable columns
8  View::composer('pagination', function($view) {
9    $queryString = array_except(Input::query(), $view->paginator->getEnvironment()->getPageName());
10   $view->paginator->appends($queryString);
11 });
```

Listing 51: ../portal/app/helpers/ErrorProvider.php

```
1  <?php
2  /**
3   * Provides Validator-like errors
4   *
5   * @package default
```

```
 */

use Illuminate\Support\Contracts\MessageProviderInterface;
use Illuminate\Support\MessageBag;

class ErrorProvider implements MessageProviderInterface{
  private $messages;
  function __construct(){
    $this->messages = new MessageBag;
  }
  public function addMessage($key,$message){
    $this->messages->add($key,$message);
  }
  public function getMessageBag(){
    return $this->messages;
  }

}
```

Listing 52: ../portal/app/helpers/macros.php

```
<?php
/**
 * Form macros
 *
 * @author Fiona Morella
 */

Form::macro('editBtn',function($size,$name,$id=""){
  if($id=="")
  return "<button type=\"submit\" class=\"btn btn-success ".$size. "\">".$name." <span class=\"glyphicon
       glyphicon-pencil\"></span></button>";
  else return "<button type=\"submit\" class=\"btn btn-success ".$size. "\" id=\"".$id."\">".$name." <span class
       =\"glyphicon glyphicon-pencil\"></span></button>";
});
Form::macro('deleteBtn',function($size,$name){
  return "<button type=\"submit\" class=\"btn btn-danger ".$size. "\">".$name." <span class=\"glyphicon glyphicon
       -trash\"></span></button>";

});
```

Listing 53: ../portal/app/lang/en/pagination.php

```
<?php

return array(

  /*
  |--------------------------------------------------------------------------
  | Pagination Language Lines
  |--------------------------------------------------------------------------
  |
  | The following language lines are used by the paginator library to build
  | the simple pagination links. You are free to change them to anything
```

```
    | you want to customize your views to better match your application.
    |
    */

    'previous' => '&laquo; Previous',

    'next'     => 'Next &raquo;',

);
```

Listing 54: ../portal/app/lang/en/reminders.php

```
<?php

return array(

    /*
    |--------------------------------------------------------------------------
    | Password Reminder Language Lines
    |--------------------------------------------------------------------------
    |
    | The following language lines are the default lines which match reasons
    | that are given by the password broker for a password update attempt
    | has failed, such as for an invalid token or invalid new password.
    |
    */

    "password" => "Passwords must be six characters and match the confirmation.",

    "user"     => "We can't find a user with that e-mail address.",

    "token"    => "This password reset token is invalid.",

);
```

Listing 55: ../portal/app/lang/en/validation.php

```
<?php

return array(

    /*
    |--------------------------------------------------------------------------
    | Validation Language Lines
    |--------------------------------------------------------------------------
    |
    | The following language lines contain the default error messages used by
    | the validator class. Some of these rules have multiple versions such
    | such as the size rules. Feel free to tweak each of these messages.
    |
    */

    "accepted"         => "The :attribute must be accepted.",
    "active_url"       => "The :attribute is not a valid URL.",
```

```
18    "after"            => "The :attribute must be a date after :date.",
19    "alpha"            => "The :attribute may only contain letters.",
20    "alpha_dash"       => "The :attribute may only contain letters, numbers, and dashes.",
21    "alpha_num"        => "The :attribute may only contain letters and numbers.",
22    "array"            => "The :attribute must be an array.",
23    "before"           => "The :attribute must be a date before :date.",
24    "between"          => array(
25      "numeric" => "The :attribute must be between :min - :max.",
26      "file"    => "The :attribute must be between :min - :max kilobytes.",
27      "string"  => "The :attribute must be between :min - :max characters.",
28      "array"   => "The :attribute must have between :min - :max items.",
29    ),
30    "confirmed"        => "The :attribute confirmation does not match.",
31    "date"             => "The :attribute is not a valid date.",
32    "date_format"      => "The :attribute does not match the format :format.",
33    "different"        => "The :attribute and :other must be different.",
34    "digits"           => "The :attribute must be :digits digits.",
35    "digits_between"   => "The :attribute must be between :min and :max digits.",
36    "email"            => "The :attribute format is invalid.",
37    "exists"           => "The selected :attribute is invalid.",
38    "image"            => "The :attribute must be an image.",
39    "in"               => "The selected :attribute is invalid.",
40    "integer"          => "The :attribute must be an integer.",
41    "ip"               => "The :attribute must be a valid IP address.",
42    "max"              => array(
43      "numeric" => "The :attribute may not be greater than :max.",
44      "file"    => "The :attribute may not be greater than :max kilobytes.",
45      "string"  => "The :attribute may not be greater than :max characters.",
46      "array"   => "The :attribute may not have more than :max items.",
47    ),
48    "mimes"            => "The :attribute must be a file of type: :values.",
49    "min"              => array(
50      "numeric" => "The :attribute must be at least :min.",
51      "file"    => "The :attribute must be at least :min kilobytes.",
52      "string"  => "The :attribute must be at least :min characters.",
53      "array"   => "The :attribute must have at least :min items.",
54    ),
55    "not_in"           => "The selected :attribute is invalid.",
56    "numeric"          => "The :attribute must be a number.",
57    "regex"            => "The :attribute format is invalid.",
58    "required"         => "The :attribute field is required.",
59    "required_if"      => "The :attribute field is required when :other is :value.",
60    "required_with"    => "The :attribute field is required when :values is present.",
61    "required_without" => "The :attribute field is required when :values is not present.",
62    "same"             => "The :attribute and :other must match.",
63    "size"             => array(
64      "numeric" => "The :attribute must be :size.",
65      "file"    => "The :attribute must be :size kilobytes.",
66      "string"  => "The :attribute must be :size characters.",
67      "array"   => "The :attribute must contain :size items.",
68    ),
69    "unique"           => "The :attribute has already been taken.",
70    "url"              => "The :attribute format is invalid.",
71
72    /*
73    |--------------------------------------------------------------------------
```

```
74    | Custom Validation Language Lines
75    |--------------------------------------------------------------------------
76    |
77    | Here you may specify custom validation messages for attributes using the
78    | convention "attribute.rule" to name the lines. This makes it quick to
79    | specify a specific custom language line for a given attribute rule.
80    |
81    */
82
83    'custom' => array(),
84
85    /*
86    |--------------------------------------------------------------------------
87    | Custom Validation Attributes
88    |--------------------------------------------------------------------------
89    |
90    | The following language lines are used to swap attribute place-holders
91    | with something more reader friendly such as E-Mail Address instead
92    | of "email". This simply helps us make messages a little cleaner.
93    |
94    */
95
96    'attributes' => array(),
97
98 );
```

Listing 56: ../portal/app/models/Department.php

```
1  <?php
2
3  class Department extends Eloquent{
4    protected $table = 'departments';
5    protected $primaryKey = 'id';
6
7    public function types()
8    {
9      return $this->hasMany('Type','department_id');
10   }
11
12
13 }
```

Listing 57: ../portal/app/models/Employee.php

```
1  <?php
2  /**
3   * Eloquent model for the 'employee_records table
4   *
5   */
6  class Employee extends Eloquent{
7    protected $table = 'employees';
8    protected $primaryKey = 'id';
9
10   public function user()
```

```
11    {
12      return $this->belongsTo('User','user_id','id');
13    }
14    public function visits()
15    {
16      return $this->belongsToMany('Visit','attending_physicians','doctor_id','visit_id');
17    }
18  }
```

Listing 58: ../portal/app/models/Patient.php

```
1   <?php
2   /**
3    * 'patients' eloquent model
4    *
5    * @package default
6    */
7   class Patient extends Eloquent{
8     protected $table = 'patients';
9     protected $primaryKey = 'id';
10
11      public function user()
12      {
13        return $this->belongsTo('User');
14      }
15
16    public function visits(){
17      return $this->hasMany('Visit',"patient_id");
18    }
19  }
```

Listing 59: ../portal/app/models/TestRequest.php

```
1   <?php
2   class TestRequest extends Eloquent{
3     protected $table = 'test_requests';
4
5     public function type()
6     {
7       return $this->belongsTo('Type','test_type','id');
8     }
9     public function visit()
10    {
11      return $this->belongsTo('Visit');
12    }
13
14    public function result()
15    {
16      return $this->hasOne('TestResult');
17    }
18  }
```

Listing 60: ../portal/app/models/TestResult.php

```php
<?php
class TestResult extends Eloquent{
  protected $table = 'test_results';

  public function request()
  {
    return $this->belongsTo('TestRequest','test_request_id');
  }

  public function added_by()
  {
    return $this->belongsTo('Employee','added_by');
  }

}
```

Listing 61: ../portal/app/models/Type.php

```php
<?php

class Type extends Eloquent{
  protected $table='test_types';
  public $timestamps = false;

  public function department(){
    return $this->belongsTo('Department');
  }
}
```

Listing 62: ../portal/app/models/User.php

```php
<?php

use Illuminate\Auth\UserInterface;
/**
* Users model extends Cartalyst User model
*
* @package default
* @author Fiona Morella
*/
class User extends \Cartalyst\Sentry\Users\Eloquent\User implements UserInterface{

  /**
  * The database table used by the model.
  *
  * @var string
  */
  protected $table = 'users';

  /**
  * The attributes excluded from the model's JSON form.
  *
  * @var array
  */
```

```
24    protected $hidden = array(
25      'password',
26      'reset_password_code',
27      'activation_code',
28      'persist_code',
29      'email_code'
30    );
31
32    public function getEmailCode()
33    {
34      $this->email_code = $emailCode = $this->getRandomString();
35
36      $this->save();
37
38      return $emailCode;
39    }
40
41    public function attemptChangeEmail($emailCode)
42    {
43      if($emailCode==$this->email_code)
44      {
45        $this->email_code = null;
46        $this->email = $this->temp_email;
47        return $this->save();
48      }
49      return false;
50    }
51
52    /**
53    * Get the unique identifier for the user.
54    *
55    * @return mixed
56    */
57    public function getAuthIdentifier()
58    {
59      return $this->getKey();
60    }
61
62    /**
63    * Get the password for the user.
64    *
65    * @return string
66    */
67    public function getAuthPassword()
68    {
69      return $this->password;
70    }
71
72    /**
73    * Get the e-mail address where password reminders are sent.
74    *
75    * @return string
76    */
77    public function getReminderEmail()
78    {
79      return $this->email;
```

```
80    }
81
82    /**
83    * Get the patient record associated
84    *
85    * @return mixed
86    */
87    public function patient()
88    {
89      return $this->hasOne('Patient');
90    }
91
92
93    //TODO relationship not working
94    /**
95    * Get the employee record associated
96    *
97    * @return mixed
98    *
99    */
100   public function employee()
101   {
102     return $this->hasOne('Employee','user_id','id');
103   }
104
105   public static function getDoctors(){
106     $group = Sentry::findGroupByName('doctors');
107
108     $doctors = Sentry::findAllUsersInGroup($group);
109     $lists = [];
110
111
112
113     $doctors->each(function($doctor){
114       $doctor->load('employee');
115     });
116
117
118     $doctors = $doctors->toArray();
119
120     foreach($doctors as $doctor){
121       $lists[$doctor["employee"]["id"]] = $doctor["employee"]["last_name"].", ".$doctor["employee"]["first_name"
              ];
122     }
123     asort($lists);
124     Log::debug(json_encode($lists));
125     return $lists;
126   }
127
128   public static function registerWithOptions($email,$password,$group,$lastName,$firstName)
129   {
130     $username = $lastName."_".$firstName;
131     $num = 1;
132     $usernameWithNumbers =  preg_replace('/[^A-Za-z0-9\. -]/', '', iconv("UTF-8", "ISO-8859-1//TRANSLIT",
            $username));
133     //check if username exists, if it does add a number to the end
```

```
134
135         while(true){
136
137           try{
138             $user = Sentry::register(array(
139               "email"    => $email,
140               "username"  => $usernameWithNumbers,
141               "password"  => $password,
142             ));
143             break;
144           }
145           catch(Exception $e){
146
147           }
148           $usernameWithNumbers = $username . ++$num;
149
150         }
151
152
153         $user->email = $email;
154
155         $user->addGroup(Sentry::getGroupProvider()->findByName($group));
156
157         $activationCode = $user->getActivationCode();
158         if($email!=null){
159           $mailData = array(
160             'email'=>$email,
161             'username'=>$user->username,
162             'id'=>$user->id,
163             'code'=>$activationCode,
164             'password'=>$password,
165             'first_name'=>$firstName,
166             'last_name'=>$lastName,
167           );
168
169           try{
170             Mail::send('emails.activate',$mailData,function($message) use($mailData){
171               $message->to($mailData["email"])->subject('Please activate your account');
172               });
173           }
174           catch(Exception $e){
175             //Session::put('general','The account was created, but sending an email failed.');
176             Log::error('Unable to send activation email');
177           }
178
179
180         }
181
182         $user->save();
183
184         return Sentry::findUserById($user->id);
185         //For whatever blasted reason simply using return $user doesn't work
186       }
187
188       /**
189       * Generates a random and easily readable password
```

```
190      *
191      * @param string $passLength
192      * @return string
193      * @author http://www.catchstudio.com/labs/password-generator/
194      */
195      public static function generatePassword($passLength = 8){
196
197              $str = "bcdfghjkmnprtvwxyBCDFGHJKMNPRTVWXY23456789";
198
199              $len = strlen($str);
200              $pw = '';
201
202              for ($i=0;$i<$len;$i++)
203                $pw .= substr($str, rand(0, $len-1), 1);
204
205              // the finished password
206              $pw = str_shuffle($pw);
207              // }
208              return substr($pw,0,8);
209            }
210          }
```

Listing 63: ../portal/app/models/Visit.php

```
1  <?php
2  /**
3   * Eloquent model for the 'visits' table
4   *
5   */
6  class Visit extends Eloquent{
7    protected $table = 'visits';
8    protected $primaryKey = 'id';
9
10   public function patient(){
11     return $this->belongsTo("Patient");
12   }
13
14   public function attendingPhysicians(){
15     return $this->belongsToMany("Employee","attending_physicians","visit_id","doctor_id");
16   }
17   public function requests(){
18     return $this->hasMany("TestRequest","visit_id");
19   }
20
21 }
```

Listing 64: ../portal/app/views/login.blade.php

```
1  @extends("layout/bootstrap")
2  @section("head")
3  @parent
4
5  @stop
6  @section('title')
```

```
@parent
  - Login
@stop
@section("content")
<div class="row">
  <div class="col-lg-offset-2 col-lg-8">
    <div class="well">
    {{ Form::open([
        "route"        => "login",
        "autocomplete" => "off",
    "url" => "login",
    "class" => "form-horizontal"
    ]) }}
  <fieldset>
    <legend>Sign In</legend>
    <div class="form-group {{$errors->has('email') ? 'has-error': ''}}">
      {{Form::label('username','Username',array('class'=>'col-sm-3 control-label'))}}
      <div class="col-sm-9">
        {{Form::text('username',null,array('class'=>'form-control'))}}
        @if($errors->has('username'))
        <span class="help-inline label label-danger">
          {{$errors->first('username')}}
        </span>
        @endif
      </div>
    </div>
    <div class="form-group {{$errors->has('password') ? 'has-error': ''}}">
      {{Form::label('password','Password',array('class'=>'col-sm-3 control-label'))}}
      <div class="col-sm-9">
        {{Form::password('password',array('type'=>'password','class'=>'form-control'))}}
        @if($errors->has('password'))
        <span class="help-inline label label-danger">
          {{$errors->first('password')}}
        </span>
        @endif
      </div>
  </fieldset>

    <div class="form-group">
      <div class="col-sm-offset-3 col-sm-9">
        {{ Form::submit("login",array("class"=>"btn btn-default")) }}
      </div>
    </div>
    {{ Form::close() }}

  <div class="text-center"><a href="{{ URL::route('forgot') }}">Forgot Password?</a> | <a href="{{URL::route('
      resend')}}">Re-send Confirmation Email</a></div>
</div></div></div>
@stop
      @section('footer')
        @parent
    @stop
```

Listing 65: ../portal/app/views/pagination.blade.php

```
{{-- Pagination using Laravel's Paginator class and Twitter Bootstrap styling--}}
<?php  $presenter = new Illuminate\Pagination\BootstrapPresenter($paginator); ?>
@if($paginator->getLastPage()>1)
  <ul class="pagination">
    <?php echo $presenter->render(); ?>
  </ul>
@endif
```

Listing 66: ../portal/app/views/landing.blade.php

```
@extends("layout/bootstrap")
@section("head")
@parent

@stop
@section('title')
@parent

@stop
@section("content")
<div class="row">
  <div class="col-lg-offset-2 col-lg-8">
    <div class="well">
  @if(Sentry::check())

      <?$loggedInUser = Sentry::getUser()?>
      <ul>
        @if($loggedInUser->inGroup(Sentry::findGroupByName('patients')))

        <li>
          {{link_to_route('mine','My Results')}}

        </li>
        @endif
        <li>
          {{link_to_route('settings','Settings')}}
        </li>
        @if($loggedInUser->inGroup(Sentry::findGroupByName('doctors')))
        <li>
          {{link_to_route('inbox','Patient Inbox')}}
        </li>
        <li>
          {{link_to_route('waiting','Patients Awaiting Results')}}
        </li>
        <li>
          {{link_to_route('waiting_for_diagnosis','Patients Awaiting Diagnosis')}}
        </li>
        <li>
          {{link_to_route('outbox_doctor','Patient Outbox')}}
        </li>
        @endif
        @if($loggedInUser->inGroup(Sentry::findGroupByName('medical_technicians')))                    <li>
          {{link_to_route('requests.inbox','Test Requests Inbox')}}
        </li>
        <li>
```

```
46            {{link_to_route('requests.outbox','Test Requests Outbox')}}
47          </li>
48          @endif
49          @if($loggedInUser->hasAccess('create_patient'))
50
51          <li>
52            {{link_to_route('outbox','Patient Outbox')}}
53          </li>
54          <li>
55            {{link_to_route('waiting_clerk','Patients Awaiting Results')}}
56          </li>
57          @endif
58      @if($loggedInUser->hasAccess('search_patient'))
59
60
61          <li>
62            <a href="{{ URL::route('patients.search')}}">Patient Search</a>
63          </li>
64          @if ($loggedInUser->hasAccess('create_patient'))
65          <li><a href="{{ URL::route('patients.create') }}">Add Patient</a></li>
66          @endif
67
68      @endif
69
70      @if($loggedInUser->inGroup(Sentry::findGroupByName('admin')))
71                <li>{{link_to_route('users.create','Add New User')}}</li>
72                <li>{{link_to_route('users.index','User Index')}}</li>
73            <li>{{link_to_route('departments.index','Departments')}}</li>
74            <li>{{link_to_route('types.create','Create New Test Type')}}</li>
75            <li>{{link_to_route('types.index','List of Test Types')}}</li>
76
77        @endif
78          </ul>
79
80  @endif
81  </div></div></div>
82  @stop
83        @section('footer')
84          @parent
85      @stop
```

Listing 67: ../portal/app/views/departments/index.blade.php

```
1   @extends("layout/bootstrap")
2   @section("head")
3   @parent
4
5   @stop
6   @section('title')
7   @parent
8     - Departments
9   @stop
10  @section("content")
11  <div class="row">
12    <div class="col-lg-offset-2 col-lg-8">
```

```
13      <div class="well">
14        {{Form::open(array('url'=>'departments','method'=>'post','class'=>'form-horizontal','autocomplete'=>'off'))
              }}
15          <div class="col-sm-12"><legend>Add New Department</legend></div>
16
17            <div class="form-group {{$errors->has('department_name') ? 'has-error':''}}">
18              <label for="department_name" class="col-sm-4 control-label"> Name of Department <span class="required
                      glyphicon glyphicon-asterisk"></span></label>
19              <div class="col-sm-8">
20              {{Form::text('department_name',null,array('class'=>'form-control'))}}
21
22              @if($errors->has('department_name'))
23              <span class="help-inline label label-danger">
24                {{$errors->first('department_name')}}
25              </span>
26              @endif
27            </div>
28            </div>
29            <div class="form-group {{$errors->has('assigned') ? 'has-error': ''}}">
30            <label for="assigned" class="col-sm-4 control-label"> Personnel <span class="required glyphicon
                   glyphicon-asterisk"></span></label>
31            <div id="assigned">
32            <div class="col-sm-8">
33              {{Form::select('assigned[]',$personnel,"",
34              array('class'=>'form-control'))}}
35
36            </div>
37            </div>
38            <div class="col-sm-8 col-sm-offset-4"><p><a href="#" id="addcurrent"><span>Add an existing employee</
                   span></a> | <a href="#" id="addnew"><span>Add a new employee</span></a> | <a href="#" id="remove">
                   Remove a field</a></div>
39            </p>
40            <div class="col-sm-8 col-sm-offset-3">
41              {{Form::submit('submit',array('class'=>'btn btn-default'))}}
42
43              </div>
44
45
46        {{Form::close()}}
47        <hr>
48        <table class="table table-bordered table-hover">
49          <thead>
50            <tr>
51              <th>
52                ID
53              </th>
54              <th>
55                Name
56              </th>
57              @if(Sentry::getUser()->inGroup(Sentry::findGroupByName('admin')))
58              <th>
59                Options
60              </th>
61              @endif
62            </tr>
63            <tbody>
```

```
64              @foreach($departments as $department)
65              <tr>
66                <td>{{$department->id}}</td>
67                <td>{{$department->department_name}}</td>
68                @if(Sentry::getUser()->inGroup(Sentry::findGroupByName('admin')))
69                <td>
70                  {{Form::open(array('method'=>'put',
71                    'route'=>array('departments.update',$department->id),
72                    'style'=>'display:none','id'=>'edit'.$department->id))}}
73                  {{Form::text('department_name',null,array('class'=>'input-sm'))}}
74                  {{Form::editBtn('btn-xs','rename')}}
75                  {{Form::button('cancel <span class=\"glyphicon glyphicon-remove\">',
76                    array('class'=>'btn btn-warning btn-xs btn-cancel'))}}
77                  {{Form::close()}}
78                  {{Form::open(
79                    array('id'=>'edit','style'=>'display:inline'))}}
80                  {{Form::editBtn('btn-xs btn-rename','edit','edit'.$department->id)}}
81                  {{Form::close()}}
82                {{Form::open(
83                    array('route'=>array('departments.disable',$department->id),
84                    'style'=>'display:inline'))}}
85                  {{Form::deleteBtn('btn-xs','disable')}}
86                  {{Form::close()}}
87                </td>
88                @endif
89              </tr>
90              @endforeach
91            </tbody>
92          </thead>
93        </table>
94        <div class="text-center">
95      {{  $departments->links()}}</div>
96  </div>
97  </div>
98  </div>
99  @stop
100       @section('footer')
101         @parent
102         @include('layout/confirm-delete')
103         {{HTML::script('js/departments.js')}}
104
105         {{HTML::script('js/edit.js')}}
106
107     @stop
```

Listing 68: ../portal/app/views/emails/activate.blade.php

```
1   <!DOCTYPE html>
2   <html lang="en-US">
3     <head>
4       <meta charset="utf-8">
5     </head>
6     <body>
7       @if(isset($first_name)&&isset($last_name))
8       <p>Dear {{$first_name}} {{$last_name}},</p>
```

```
9       @endif
10      <p>Thank you for signing up for the Las Pi&ntilde;as Doctors Hospital Information System. You may now
            activate your account at <a href="{{URL::to('activate',array('id'=>$id,'code'=>$code))}}">{{URL::to('
            activate',array('id'=>$id,'code'=>$code))}}</a>.</p>
11      @if (isset($password))
12      <p>Your temporary login details are as follows:</p>
13      <p>Username: <pre>{{$username}}</pre></p>
14      <p>Password: <pre>{{$password}}</pre>  </p>
15      <p>You may change your password once your account has been activated.</p>
16      @endif
17
18      <small>You have received this email because an account was recently registered with this email address on the
            LPDH system. If you did not authorize this action, please ignore this email.</small>
19    </body>
20  </html>
```

Listing 69: ../portal/app/views/emails/auth/confirm.blade.php

```
1   <!DOCTYPE html>
2   <html lang="en-US">
3     <head>
4       <meta charset="utf-8">
5     </head>
6     <body>
7       <h2>Confirm change of email</h2>
8
9       <div>
10        Please click here to confirm email change: {{ URL::to('confirm', array('id'=>$id,'code'=>$code)) }}.
11      </div>
12    </body>
13  </html>
```

Listing 70: ../portal/app/views/emails/auth/reminder.blade.php

```
1   <!DOCTYPE html>
2   <html lang="en-US">
3     <head>
4       <meta charset="utf-8">
5     </head>
6     <body>
7       <h2>Password Reset</h2>
8
9       <div>
10        To reset your password, complete this form: <a href="{{URL::to('reset',array('id'=>$id,'code'=>$code))
              }}">{{ URL::to('reset', array('id'=>$id,'code'=>$code)) }}</a>.
11      </div>
12    </body>
13  </html>
```

Listing 71: ../portal/app/views/errors/403.blade.php

```
1   @extends("layout/bootstrap")
2   @section("head")
```

```
3  @parent
4
5  @stop
6  @section('title')
7  @parent
8  - ERROR 403
9  @stop
10 @section("content")
11 <div class="jumbotron">
12
13   <h1>403 ERROR: FORBIDDEN</h1>
14   <p>You do not have permission to view that page.</p>
15   <p><a class="btn btn-primary btn-lg" role="button" href="{{URL::route('profile')}}">Back to home page &raquo;</
         a></p>
16 </div>
17 @stop
```

Listing 72: ../portal/app/views/errors/404.blade.php

```
1  @extends("layout/bootstrap")
2  @section("head")
3  @parent
4
5  @stop
6  @section('title')
7  @parent
8  - ERROR 404
9  @stop
10 @section("content")
11 <div class="jumbotron">
12
13   <h1>404 ERROR: NOT FOUND</h1>
14   <p>Page not found.</p>
15   <p><a class="btn btn-primary btn-lg" role="button" href="{{URL::route('profile')}}">Back to home page &raquo;</
         a></p>
16 </div>
17 @stop
```

Listing 73: ../portal/app/views/errors/500.blade.php

```
1  @extends("layout/bootstrap")
2  @section("head")
3  @parent
4
5  @stop
6  @section('title')
7  @parent
8  - ERROR 500
9  @stop
10 @section("content")
11 <div class="jumbotron">
12
13   <h1>500 ERROR: INTERNAL SERVER ERROR</h1>
```

```
14    <p>Sorry about that! Please try again later. If you continue to receive this error, please contact site
          administrators.</p>
15    <p><a class="btn btn-primary btn-lg" role="button" href="{{URL::route('profile')}}">Back to home page &raquo;</
          a></p>
16  </div>
17  @stop
```

Listing 74: ../portal/app/views/errors/503.blade.php

```
1   @extends("layout/bootstrap")
2   @section("head")
3   @parent
4
5   @stop
6   @section('title')
7   @parent
8   - ERROR 503
9   @stop
10  @section("content")
11  <div class="jumbotron">
12
13    <h1>503 ERROR: INTERNAL SERVER ERROR</h1>
14    <p>The service is currently down for maintenance. Please try again later.</p>
15    <p><a class="btn btn-primary btn-lg" role="button" href="{{URL::route('profile')}}">Back to home page &raquo;</
          a></p>
16  </div>
17  @stop
```

Listing 75: ../portal/app/views/forms/patient–basic–edit.blade.php

```
1   <div class="form-group"><div class="col-sm-offset-3 col-sm-9"><div class="help-block">Asterisks <span class="
        required glyphicon glyphicon-asterisk"></span> indicate required fields.</div></div>
2   </div>
3   <div>
4     <div class="form-group {{$errors->has('hospital_id') ? 'has-error': ''}}">
5       <label for='hospital_id' class='col-sm-3 control-label'>Patient ID<span class="required glyphicon glyphicon-
            asterisk"></span></label>
6
7       <div class="col-sm-3 ">
8         {{Form::text('hospital_id_1',null,array('class'=>'form-control ','maxlength'=>2))}}</div><div class="col-sm
              -1">-</div><div class="col-sm-5">
9         {{Form::text('hospital_id_2',null,array('class'=>'form-control','maxlength'=>5))}}</div>
10        @if($errors->has('hospital_id_1'))
11        <span class="help-inline label label-danger">
12          {{$errors->first('hospital_id')}}
13        </span>
14        @endif
15      </div>
16    </div>
17    <div class="form-group {{$errors->has('last_name') ? 'has-error': ''}}">
18      <label for='last_name' class='col-sm-3 control-label'>Last Name <span class="required glyphicon glyphicon-
            asterisk"></span></label>
19
20      <div class="col-sm-9">
```

```
21        {{Form::text('last_name',null,array('class'=>'form-control'))}}
22        @if($errors->has('last_name'))
23        <span class="help-inline label label-danger">
24          {{$errors->first('last_name')}}
25        </span>
26        @endif
27      </div>
28    </div>
29    <div class="form-group {{$errors->has('first_name') ? 'has-error': ''}}">
30      <label for='first_name' class='col-sm-3 control-label'>First Name <span class="required glyphicon glyphicon-
            asterisk"></span></label>
31
32      {{--for some reason clicking on the label won't auto-focus the input--}}
33      <div class="col-sm-9">
34        {{Form::text('first_name',null,array('class'=>'form-control'))}}
35        @if($errors->has('first_name'))
36        <span class="help-inline label label-danger">
37          {{$errors->first('first_name')}}
38        </span>
39        @endif
40      </div>
41    </div>
42    <div class="form-group {{$errors->has('middle_name') ? 'has-error': ''}}">
43      {{Form::label('middle_name','Middle Name',array('class'=>'col-sm-3 control-label'))}}
44      <div class="col-sm-9">
45        {{Form::text('middle_name',null,array('class'=>'form-control'))}}
46        @if($errors->has('middle_name'))
47        <span class="help-inline label label-danger">
48          {{$errors->first('middle_name')}}
49        </span>
50        @endif
51      </div>
52    </div>
53    <div class="form-group {{$errors->has('email') ? 'has-error': ''}}">
54  <label for='email' class='col-sm-3 control-label'>Email <span></span></label>
55      <div class="col-sm-9">{{Form::text('email', null,array('class'=>'form-control'))}}</div>
56      @if($errors->has('email'))
57      <span class="help-inline label label-danger">
58        {{$errors->first('email')}}
59      </span>
60      @endif
61    </div>
62    <div class="form-group {{$errors->has('sex') ? 'has-error': ''}}">
63  <label for='sex' class='col-sm-3 control-label'>Sex <span class="required glyphicon glyphicon-asterisk"></span></
        label>
64      <div class="col-sm-9">
65        <div class="radio">
66          <label>
67            {{ Form::radio('sex','male','male',array('id'=>'male')) }}
68            Male
69          </label>
70        </div>
71        <div class="radio">
72          <label>
73            {{ Form::radio('sex','female','male',array('id'=>'male')) }}
74            Female
```

```
75          </label>
76        </div>
77        @if($errors->has('sex'))
78        <span class="help-inline label label-danger">
79          {{$errors->first('sex')}}
80        </span>
81        @endif
82      </div>
83    </div>
84    <div class="form-group {{$errors->has('civil_status') ? 'has-error': ''}}">
85      {{Form::label('civil_status','Civil Status',array('class'=>'col-sm-3 control-label'))}}
86      <div class="col-sm-9">
87        <div class="radio">
88          <label>
89            {{ Form::radio('civil_status','single','single') }}
90            Single
91          </label>
92        </div>
93        <div class="radio">
94          <label>
95            {{ Form::radio('civil_status','married','married') }}
96            Married
97          </label>
98        </div>
99        <div class="radio">
100         <label>
101           {{ Form::radio('civil_status','widowed','widowed') }}
102           Widowed
103         </label>
104       </div>
105       @if($errors->has('civil_status'))
106       <span class="help-inline label label-danger">
107         {{$errors->first('civil_status')}}
108       </span>
109       @endif
110     </div>
111   </div>
112   <div class="form-group {{$errors->has('birthdate') ? 'has-error': ''}}">
113   <label for='birthdate' class='col-sm-3 control-label'>
114     Date of Birth <span class="required glyphicon glyphicon-asterisk"></span>
115   </label>
116   <div class="col-sm-9">
117     <div class="input-group date" id="birthdate">
118       <span class="input-group-addon"><span class="glyphicon glyphicon-calendar"></span></span>
119       {{Form::text('birthdate',null,array('data-format'=>'YYYY-MM-DD', 'class'=>'form-control'))}}
120     </div>
121     @if($errors->has('birthdate'))
122     <span class="help-inline label label-danger">
123       {{$errors->first('birthdate')}}
124     </span>
125     @endif
126   </div>
127   </div>
128
129   <div class="form-group {{$errors->has('birthplace') ? 'has-error': ''}}">
130     {{Form::label('birthplace','Place of Birth',array('class'=>'col-sm-3 control-label'))}}
```

```
131       <div class="col-sm-9">
132         {{Form::text('birthplace',null,array('class'=>'form-control'))}}
133
134         @if($errors->has('birthplace'))
135         <span class="help-inline label label-danger">
136           {{$errors->first('birthplace')}}
137         </span>
138         @endif
139       </div>
140     </div>
141     <div class="form-group {{$errors->has('telephone_number') ? 'has-error': ''}}">
142       {{Form::label('telephone_number','Telephone Number',array('class'=>'col-sm-3 control-label'))}}
143       <div class="col-sm-9">
144         {{Form::text('telephone_number',null,array('class'=>'form-control'))}}
145         @if($errors->has('telephone_number'))
146         <span class="help-inline label label-danger">
147           {{$errors->first('telephone_number')}}
148         </span>
149         @endif
150       </div>
151     </div>
152
153     <div class="form-group {{$errors->has('mobile_number') ? 'has-error': ''}}">
154       {{Form::label('mobile_number','Mobile Number',array('class'=>'col-sm-3 control-label'))}}
155       <div class="col-sm-9">
156         {{Form::text('mobile_number',null,array('class'=>'form-control'))}}
157         @if($errors->has('mobile_number'))
158         <span class="help-inline label label-danger">
159           {{$errors->first('mobile_number')}}
160         </span>
161         @endif
162       </div>
163     </div>
164     <div class="form-group {{$errors->has('address') ? 'has-error': ''}}">
165       {{Form::label('address','Address',array('class'=>'col-sm-3 control-label'))}}
166       <div class="col-sm-9">{{Form::text('address',null,array('class'=>'form-control'))}}
167         @if($errors->has('address'))
168         <span class="help-inline label label-danger">
169           {{$errors->first('address')}}
170         </span>
171         @endif
172
173     </div></div>
174     <div class="form-group {{$errors->has('religion') ? 'has-error': ''}}">
175       {{Form::label('religion','Religion',array('class'=>'col-sm-3 control-label'))}}
176       <div class="col-sm-9">{{Form::text('religion',null,array('class'=>'form-control','id'=>'religion'))}}
177         @if($errors->has('religion'))
178         <span class="help-inline label label-danger">
179           {{$errors->first('religion')}}
180         </span>
181         @endif
182     </div></div>
183     <div class="form-group {{$errors->has('nationality') ? 'has-error': ''}}">
184       {{Form::label('nationality','Nationality',array('class'=>'col-sm-3 control-label'))}}
185       <div class="col-sm-9">{{Form::text('nationality',null,array('class'=>'form-control'))}}
186         @if($errors->has('nationality'))
```

```
187         <span class="help-inline label label-danger">
188           {{$errors->first('nationality')}}
189         </span>
190         @endif
191
192     </div></div>
193     <div class="form-group {{$errors->has('occupation') ? 'has-error': ''}}">
194       {{Form::label('occupation','Occupation',array('class'=>'col-sm-3 control-label'))}}
195       <div class="col-sm-9">{{Form::text('occupation',null,array('class'=>'form-control'))}}
196         @if($errors->has('occupation'))
197         <span class="help-inline label label-danger">
198           {{$errors->first('occupation')}}
199         </span>
200         @endif
201
202     </div></div>
203     <div class="form-group"><div class="col-sm-offset-3 col-sm-9">
204       <label>{{Form::checkbox('is_dependent',1,false)}} Dependent</label>
205     </div></div>
206     <div class="form-group"><div class="col-sm-offset-3 col-sm-9">
207       <label>{{Form::checkbox('is_stockholder',1,false)}} Stockholder</label>
208     </div></div>
209     <div class="form-group"><div class="col-sm-offset-3 col-sm-9">
210       <label>{{Form::checkbox('is_stockholder_dependent',1,false)}} Dependent of Stockholder</label>
211     </div></div>
```

Listing 76: ../portal/app/views/forms/patient–contact.blade.php

```
1   <div class="form-group">{{Form::label('company_name','Company',array('class'=>'col-sm-4 control-label'))}}
2   <div class="col-sm-8">{{Form::text('company_name',null,array('class'=>'form-control'))}}</div></div>
3   <div class="form-group">{{Form::label('company_address','Company Address',array('class'=>'col-sm-4 control-
        label'))}}
4   <div class="col-sm-8">{{Form::text('company_address',null,array('class'=>'form-control'))}}</div></div>
5   <div class="form-group">{{Form::label('father_name','Father\'s Name',array('class'=>'col-sm-4 control-label'))
        }}
6   <div class="col-sm-8">{{Form::text('father_name',null,array('class'=>'form-control'))}}</div></div>
7   <div class="form-group">{{Form::label('mother_name','Mother\'s Name',array('class'=>'col-sm-4 control-label'))
        }}
8   <div class="col-sm-8">{{Form::text('mother_name',null,array('class'=>'form-control'))}}</div></div>
9   <div class="form-group">{{Form::label('parent_address','Parents\' Address',array('class'=>'col-sm-4 control-
        label'))}}
10  <div class="col-sm-8">{{Form::text('parent_address',null,array('class'=>'form-control'))}}</div></div>
11  <div id="spouseInfo">
12    <div class="form-group">{{Form::label('spouse_name','Spouse\'s Name',array('class'=>'col-sm-4 control-label')
          )}}
13    <div class="col-sm-8">{{Form::text('spouse_name',null,array('class'=>'form-control'))}}</div></div>
14    <div class="form-group">{{Form::label('spouse_occupation','Spouse\'s Occupation',array('class'=>'col-sm-4
          control-label'))}}
15    <div class="col-sm-8">{{Form::text('spouse_occupation',null,array('class'=>'form-control'))}}</div></div>
16    <div class="form-group">{{Form::label('spouse_company','Spouse\'s Company',array('class'=>'col-sm-4 control-
          label'))}}
17    <div class="col-sm-8">{{Form::text('spouse_company',null,array('class'=>'form-control'))}}</div></div>
18
19  <div class="form-group">{{Form::label('emergency_contact_name','Emergency Contact\'s Name',array('class'=>'col-
        sm-4 control-label'))}}
```

```
20    <div class="col-sm-8">{{Form::text('emergency_contact_name',null,array('class'=>'form-control'))}}</div></div>
21    <div class="form-group">{{Form::label('emergency_contact_number','Emergency Contact\'s Number',array('class'=>'
          col-sm-4 control-label'))}}
22    <div class="col-sm-8">{{Form::text('emergency_contact_number',null,array('class'=>'form-control'))}}</div></div
          >
23    <div class="form-group">{{Form::label('emergency_contact_address','Emergency Contact\'s Address',array('class
          '=>'col-sm-4 control-label'))}}
24    <div class="col-sm-8">{{Form::text('emergency_contact_address',null,array('class'=>'form-control'))}}</div></
          div>
25
26    <div class="form-group">{{Form::label('account_responsibility_name','Person Responsible for the Account',array
          ('class'=>'col-sm-4 control-label'))}}
27    <div class="col-sm-8">{{Form::text('account_responsibility_name',null,array('class'=>'form-control'))}}</div></
          div>
28    <div class="form-group">{{Form::label('account_responsibility_relationship','Relationship to Patient',array('
          class'=>'col-sm-4 control-label'))}}
29    <div class="col-sm-8">{{Form::text('account_responsibility_relationship',null,array('class'=>'form-control'))
          }}</div></div>
30    <div class="form-group">{{Form::label('account_responsibility_number','Number',array('class'=>'col-sm-4 control
          -label'))}}
31    <div class="col-sm-8">{{Form::text('account_responsibility_number',null,array('class'=>'form-control'))}}</div
          ></div>
32    <div class="form-group">{{Form::label('account_responsibility_address','Address',array('class'=>'col-sm-4
          control-label'))}}
33    <div class="col-sm-8">{{Form::text('account_responsibility_address',null,array('class'=>'form-control'))}}</div
          ></div>
```

Listing 77: ../portal/app/views/forms/visit.blade.php

```
1
2   <?
3   if(!$newRecord){
4     if($edit){
5       $input = null;
6       $repop2 = Input::old('attending_physicians') ?: $visit->attendingPhysicians->lists('doctor_id');
7     }else{
8       $input = date("Y-m-d H:i:s");
9       $repop2 = null;
10    }
11  }else{
12    $input = date("Y-m-d H:i:s");
13    $repop2 = null;
14  }
15
16  ?>
17  <div class="form-group"><div class="col-sm-offset-4 col-sm-8"><div class="help-block">Asterisks <span class="
        required glyphicon glyphicon-asterisk"></span> indicate required fields.</div></div>
18  </div>
19  <div class="form-group {{$errors->has('date_of_visit') ? 'has-error': ''}}">
20  <label for='date_of_visit' class='col-sm-4 control-label'>
21    Date of Visit <span class="required glyphicon glyphicon-asterisk"></span>
22  </label>
23  <div class="col-sm-8">
24    <div class="input-group date" id="date_of_visit">
25      <span class="input-group-addon"><span class="glyphicon glyphicon-calendar"></span></span>
```

```
26      {{Form::text('date_of_visit',$input,array('readonly','data-format'=>'YYYY-MM-DD HH:mm:00', 'class'=>'form-
            control'))}}
27    </div>
28    @if($errors->has('date_of_visit'))
29
30    <span class="help-inline label label-danger">
31      {{$errors->first('date_of_visit')}}
32    </span>
33    @endif
34  </div>
35  </div>
36
37
38  <div class="form-group {{$errors->has('height') ? 'has-error': ''}}" >
39  {{Form::label('height','Height',array('class'=>'col-sm-4 control-label'))}}
40  <div class="col-sm-8">
41    {{Form::text('height',null,array('class'=>'form-control'))}}
42    @if($errors->has('height'))
43    <span class="help-inline label label-danger">
44      {{$errors->first('height')}}
45    </span>
46    @endif
47  </div>
48  </div>
49  <div class="form-group {{$errors->has('weight') ? 'has-error': ''}}">
50  {{Form::label('weight','Weight',array('class'=>'col-sm-4 control-label'))}}
51  <div class="col-sm-8">
52    {{Form::text('weight',null,array('class'=>'form-control'))}}
53    @if($errors->has('weight'))
54    <span class="help-inline label label-danger">
55      {{$errors->first('weight')}}
56    </span>
57    @endif
58  </div>
59  </div>
60  <div class="form-group {{$errors->has('c_r') ? 'has-error': ''}}">
61  {{Form::label('c_r','C/R',array('class'=>'col-sm-4 control-label'))}}
62  <div class="col-sm-8">
63    {{Form::text('c_r',null,array('class'=>'form-control'))}}
64    @if($errors->has('c_r'))
65    <span class="help-inline label label-danger">
66      {{$errors->first('c_r')}}
67    </span>
68    @endif
69  </div>
70  </div>
71  <div class="form-group {{$errors->has('r_r') ? 'has-error': ''}}">
72  {{Form::label('r_r','R/R',array('class'=>'col-sm-4 control-label'))}}
73  <div class="col-sm-8">
74    {{Form::text('r_r',null,array('class'=>'form-control'))}}
75    @if($errors->has('r_r'))
76    <span class="help-inline label label-danger">
77      {{$errors->first('r_r')}}
78    </span>
79    @endif
80
```

```
</div>
</div>
<div class="form-group {{$errors->has('temperature') ? 'has-error': ''}}">
{{Form::label('temperature','Temperature',array('class'=>'col-sm-4 control-label'))}}
<div class="col-sm-8">
  {{Form::text('temperature',null,array('class'=>'form-control'))}}
  @if($errors->has('temperature'))
  <span class="help-inline label label-danger">
    {{$errors->first('temperature')}}
  </span>
  @endif

</div>
</div>
<div class="form-group {{$errors->has('bp') ? 'has-error': ''}}">
{{Form::label('bp','BP',array('class'=>'col-sm-4 control-label'))}}
<div class="col-sm-8">
  {{Form::text('bp',null,array('class'=>'form-control'))}}
  @if($errors->has('bp'))
  <span class="help-inline label label-danger">
    {{$errors->first('bp')}}
  </span>
  @endif

</div>
</div>
@if($newRecord)
<div class="form-group {{$errors->has('attending_physicians') ? 'has-error': ''}}">
{{Form::label('attending_physicians','Attending Physicians',array('class'=>'col-sm-4 control-label'))}}
<div class="col-sm-8">
  {{Form::select('attending_physicians[]',$doctors,$repop2,
  array(
    'id' => 'attending_physicians',
  'class'=>'form-control populate'))}}
  @if($errors->has('attending_physicians'))
  <span class="help-inline label label-danger">
    {{$errors->first('attending_physicians')}}
  </span>
  @endif
</div>
</div>
  @endif
```

Listing 78: ../portal/app/views/layout/bootstrap.blade.php

```
<!DOCTYPE html>
<html lang=  e n  >
<head>
  @section('head')

  {{HTML::style('css/main.min.css')}}

  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  @show
    <title>
```

```
11        @section('title')
12        @show
13      </title>
14  </head>
15
16  <body>
17    <div id="wrap">
18      @include('layout/navbar')
19      <div class="container">
20        @if(Session::has('success'))
21          <div class="alert alert-success alert-dismissable">
22                    <button type="button" class="close" data-dismiss="alert">&times;</button>
23            {{Session::get('success')}}
24          </div>
25        @endif
26        @if(Session::has('general'))
27        <div class="alert alert-danger alert-dismissable">
28                  <button type="button" class="close" data-dismiss="alert">&times;</button>
29          {{Session::get('general')}}
30        </div>
31        @endif
32        @if($errors->has('general'))
33        @foreach($errors->get('general') as $error)
34        <div class="alert alert-danger alert-dismissable">
35                  <button type="button" class="close" data-dismiss="alert">&times;</button>
36          {{$error}}
37        </div>
38        @endforeach
39        @endif
40        @yield('content')
41      </div>
42    </div>
43      <div id="push"></div><!-- so that footer will always stick to the bottom-->
44
45    <div id="footer">
46      @include('layout/footer')
47    </div>
48  </body>
49  </html>
```

Listing 79: ../portal/app/views/layout/confirm–delete.blade.php

```
1   <div id="dialog-confirm" title="Confirm Deletion">Are you sure you want to delete this record?</div>
2   {{HTML::script('js/confirm-delete.js')}}
```

Listing 80: ../portal/app/views/layout/footer.blade.php

```
1   @section("footer")
2   <footer>
3     <div class="container">
4       <div class="text-center">
5        <p>Fiona Morella, 2013-2014. Powered by <a href="http://laravel.com/">Laravel</a>, <a href="http://jquery.
              com">JQuery</a>, and <a href="http://getbootstrap.com">Bootstrap</a>.
6        </p>
```

```
    </div>
</div>
</footer>
<!-- here comes the javascript-->
{{HTML::script('js/main.min.js')}}
@show
```

Listing 81: ../portal/app/views/layout/navbar.blade.php

```
@section('navbar')
<?php $loggedInUser = Sentry::getUser(); ?>
<nav class="navbar navbar-default navbar-fixed-top" role="navigation">
  <!-- Brand and toggle get grouped for better mobile display -->
  <div class="navbar-header">
    <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#bs-example-navbar-collapse
        -1">
      <span class="sr-only">Toggle navigation</span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>

    </button>
  </div>

  <!-- Collect the nav links, forms, and other content for toggling -->
  <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
    <ul class="nav navbar-nav">
          <a class="navbar-brand" href="#"><img src='{{asset("img/logo.png")}}'/> LAS PI&Ntilde;AS DOCTORS
              HOSPITAL</a>

      @if(Sentry::check())
        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-toggle="dropdown">
            <span class="glyphicon glyphicon-user"></span> My Account
            <b class="caret"></b>
          </a>
          <ul class="dropdown-menu">
            @if($loggedInUser->inGroup(Sentry::findGroupByName('patients')))

            <li>
              {{link_to_route('mine','My Results')}}

            </li>
            @endif
            <li>
              {{link_to_route('settings','Settings')}}
            </li>

          </ul>
        </li>
        @if($loggedInUser->hasAccess('search_patient'))
        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-toggle="dropdown">
            Patients
            <b class="caret"></b>
```

```
45            </a>
46            <ul class="dropdown-menu">
47              <li>
48                <a href="{{ URL::route('patients.search')}}">Patient Search</a>
49              </li>
50              @if ($loggedInUser->hasAccess('create_patient'))
51              <li><a href="{{ URL::route('patients.create') }}">Add Patient</a></li>
52              @endif
53              @if($loggedInUser->inGroup(Sentry::findGroupByName('doctors')))
54              <li>
55                {{link_to_route('inbox','Patient Inbox')}}
56              </li>
57
58              <li>
59                {{link_to_route('waiting','Patients Awaiting Results')}}
60              </li>
61              <li>
62                {{link_to_route('waiting_for_diagnosis','Patients Awaiting Diagnosis')}}
63              </li>
64              <li>
65                {{link_to_route('outbox_doctor','Patient Outbox')}}
66              </li>
67              @endif
68              @if($loggedInUser->inGroup(Sentry::findGroupByName('medical_technicians')))                    <li>
69                {{link_to_route('requests.inbox','Test Requests Inbox')}}
70              </li>
71              <li>
72                {{link_to_route('requests.outbox','Test Requests Outbox')}}
73              </li>
74              @endif
75              @if($loggedInUser->hasAccess('create_patient'))
76
77              <li>
78                {{link_to_route('outbox','Patient Outbox')}}
79              </li>
80              <li>
81                {{link_to_route('waiting_clerk','Patients Awaiting Results')}}
82              </li>
83              <li>
84                {{link_to_route('pending','Patients Pending')}}
85              </li>
86              @endif
87            </ul>
88          </li>
89          @endif
90
91          @if($loggedInUser->inGroup(Sentry::findGroupByName('admin')))
92          <li class="dropdown">
93                  <a href="#" class="dropdown-toggle" data-toggle="dropdown">Admin <b class="caret"></b></a>
94                  <ul class="dropdown-menu">
95                    <li>{{link_to_route('users.create','Add New User')}}</li>
96                    <li>{{link_to_route('users.index','User Index')}}</li>
97                <li>{{link_to_route('departments.index','Departments')}}</li>
98                <li>{{link_to_route('types.create','Create New Test Type')}}</li>
99                <li>{{link_to_route('types.index','List of Test Types')}}</li>
100                 </ul>
```

```
101                 </li>
102             @endif
103       </ul>
104       <ul class="nav navbar-nav navbar-right">
105           @if(!Sentry::check())
106
107           <li class="active"><a href="{{URL::route('login')}}">Login</a></li>
108           @else
109           <p class="navbar-text">{{$loggedInUser->username}}</p>
110
111           <li class="active"><a href="{{URL::route('logout')}}">Logout</a></li>
112           @endif
113
114         </li>
115       </ul>
116       @endif
117     </div>
118   </nav>
119   @show
```

Listing 82: ../portal/app/views/password/reset.blade.php

```
1   @extends("layout/bootstrap")
2   @section("content")
3   {{HTML::style("css/forms.css");}}
4   <div class="form">
5     <h2>Reset Password</h2>
6       {{ Form::open(array('route'=> array('reset',$token))) }}
7
8
9       {{ Form::hidden("token", $token) }}
10
11      <p>
12          {{ Form::label("email", "Email") }}
13      </p>
14      <p>
15          {{ Form::text("email", Input::get("email"), [
16              "placeholder" => "me@example.com"
17          ]) }}
18      </p>
19          <p>
20          {{ Form::label("password", "Password") }}
21      </p>
22      <p>
23          {{ Form::password("password", [
24              "placeholder" => "                              "
25          ]) }}
26      </p>
27          <p>
28          {{ Form::label("password_confirmation", "Retype Password") }}
29      </p>
30      <p>
31          {{ Form::password("password_confirmation", [
32              "placeholder" => "                              "
33          ]) }}
```

```
34      </p>
35      <p class="reset">
36          {{ Form::submit("reset password") }}
37      </p>
38      {{ Form::close() }}
39  </div>
40  @stop
41  @section("footer")
42      @parent
43      <script src="//polyfill.io"></script>
44  @stop
```

Listing 83: ../portal/app/views/patients/create.blade.php

```
1   @extends("layout/bootstrap")
2   @section("head")
3   @parent
4
5   @stop
6   @section('title')
7   @parent
8   - Add Patient
9   @stop
10  @section("content")
11
12  <div class="row">
13
14
15  <div class="col-lg-offset-2 col-lg-8">
16    <div class="well">
17      {{ Form::open(array("url"=>"patients","class"=>"form-horizontal")) }}
18        {{Form::hidden('employee_id',Sentry::getUser()->employee->employee_id)}}
19
20        <?php $newRecord = true; ?>
21
22        <fieldset>
23          <legend>Patient Information</legend>
24          @include("forms/patient-basic-edit")
25        </fieldset>
26        <fieldset>
27          <legend>Contact Information</legend>
28          @include("forms/patient-contact")
29        </fieldset>
30        <fieldset>
31          <legend>Visit Information</legend>
32          @include("forms/visit")
33        </fieldset>
34        <div class="form-group">
35          <div class="col-sm-offset-4 col-sm-8">
36            {{ Form::submit("Submit",array("id"=>"submit","class"=>"btn btn-default")) }}
37          </div>
38        </div>
39
40        {{Form::close()}}
41      </div>
```

```
42    </div>
43  </div>
44
45  @stop
46    @section('footer')
47      @parent
48    {{HTML::script('js/forms.js')}}
49
50      @stop
```

Listing 84: ../portal/app/views/patients/edit.blade.php

```
1   @extends("layout/bootstrap")
2     @section("head")
3       @parent
4     @stop
5     @section('title')
6       @parent
7         - Edit Patient Record
8       @stop
9     @section("content")
10  <div class="row">
11
12    <div class="col-lg-offset-2 col-lg-8">
13      <div class="well">
14        {{Form::model($patient,array(
15          'class'=>'form-horizontal',
16          'role'=>'form',
17          'method'=>'PUT',
18          'route'=>array('patients.update',$patient->id)))}}
19          <?php $newRecord = false; ?>
20          <fieldset>
21            <legend>Patient Information</legend>
22            @include("forms/patient-basic-edit")
23          </fieldset>
24          <fieldset>
25            <legend>Contact Information</legend>
26            @include("forms/patient-contact")
27          </fieldset>
28            {{-- @include("forms/patient-contact") --}}
29          <div class="form-group">
30            <div class="col-sm-offset-4 col-sm-8">
31              {{ Form::submit("submit",array("id"=>"submit","class"=>"btn btn-default")) }}
32            </div>
33          </div>
34
35          {{Form::close()}}
36        </div>
37      </div>
38    </div>
39    @stop
40      @section("footer")
41        @parent
42    {{HTML::script('js/forms.js')}}
43    @stop
```

Listing 85: ../portal/app/views/patients/inbox.blade.php

```
@extends("layout/bootstrap")
  @section("head")
    @parent
@stop
@section('title')
  @parent
- Inbox
@stop
  @section("content")
<div class="row">
  <div class="col-lg-offset-2 col-lg-8">
    <div class="well">
      <table class="table table-bordered table-striped">
        <thead>
          <th>Patient ID</th>
          <th>Name </th>
          <th>Date of Visit</th>
          <th>Added By</th>
          <th>Status</th>
        </thead>
        <tbody>
          @foreach($visits as $visit)
          <tr>
            <td>{{link_to_route('patients.show',$visit->patient->hospital_id,array('patients'=>$visit->patient->
                 id))}}</td>
            <td>{{link_to_route('diagnose',$visit->patient->last_name.", ". $visit->patient->first_name,array('
                 visits'=>$visit->id))}}</td>
            <td>
              {{link_to_route('diagnose',$visit->date_of_visit,array('visits'=>$visit->id))}}
            </td>
            <td>
              {{Employee::find($visit->added_by)->last_name}},
              {{Employee::find($visit->added_by)->first_name}}
            </td>
            <td>
              {{$visit->status}}
            </td>
          </tr>

          @endforeach
        </tbody>
      </table>
      <div class="text-center">
    {{  $visits->links()}}</div>

    </div>
  </div>
</div>
@stop
  @section('footer')
    @parent
      @stop
```

Listing 86: ../portal/app/views/patients/outbox.blade.php

```
@extends("layout/bootstrap")
  @section("head")
    @parent
@stop
@section('title')
  @parent
- Outbox
@stop
  @section("content")
<div class="row">
  <div class="col-lg-offset-2 col-lg-8">
    <div class="well">
      <table class="table table-bordered table-striped">
        <thead>
          <th>Patient ID</th>
          <th>Name </th>
          <th>Date of Visit</th>
          <th>Attending Physicians</th>
        </thead>
        <tbody>
          @foreach($visits as $visit)
          <tr>
            <td>{{link_to_route('patients.show',$visit->patient->hospital_id,array('patients'=>$visit->patient->
                 id))}}</td>
            <td>{{link_to_route('patients.show',$visit->patient->last_name.", ". $visit->patient->first_name,
                 array('patients'=>$visit->patient->id))}}</td>
            <td>
              {{link_to_route('visits.show',$visit->date_of_visit,array('visits'=>$visit->id))}}
              @if(Sentry::getUser()->inGroup(Sentry::findGroupByName('clerks')))
              <p>

                 @if($visit->status=='waiting')
                 @if($visit->requests->filter(function($request){if($request->status!='completed') return
                      $request;})->count()==0)
                all results are in {{Form::open(array('url'=>'visits/status/'.$visit->id,'method'=>'post'))}}{{
                     Form::submit('send to doctor inbox',array('style'=>'display:inline','class'=>'btn btn-
                     success btn-xs'))}}{{Form::close()}}
                 @else
                 waiting for results

                 @endif


              @endif</p>
              @endif
            </td>
            <td>
              @foreach ($visit->attendingPhysicians as $physician)
              {{$physician->last_name}}, {{$physician->first_name}} <br />
              @endforeach
            </td>
          </tr>

          @endforeach
```

```
50          </tbody>
51        </table>
52        <div class="text-center">
53      {{  $visits->links()}}</div>
54
55      </div>
56    </div>
57  </div>
58  @stop
59    @section('footer')
60      @parent
61        @stop
```

Listing 87: ../portal/app/views/patients/pending.blade.php

```
1   @extends("layout/bootstrap")
2     @section("head")
3       @parent
4   @stop
5     @section('title')
6       @parent
7         - Outbox
8   @stop
9     @section("content")
10  <div class="row">
11    <div class="col-lg-offset-2 col-lg-8">
12      <div class="well">
13        <table class="table table-bordered table-striped">
14          <thead>
15            <th>Patient ID</th>
16            <th>Name </th>
17            <th>Date of Visit</th>
18            <th>Attending Physicians</th>
19          </thead>
20          <tbody>
21            @foreach($visits as $visit)
22            <tr>
23              <td>{{link_to_route('patients.show',$visit->patient->hospital_id,array('patients'=>$visit->patient->
                      id))}}</td>
24              <td>{{link_to_route('patients.show',$visit->patient->last_name.", ". $visit->patient->first_name,
                      array('patients'=>$visit->patient->id))}}</td>
25              <td>
26                {{link_to_route('visits.show',$visit->date_of_visit,array('visits'=>$visit->id))}}
27                @if(Sentry::getUser()->inGroup(Sentry::findGroupByName('clerks')))
28                {{Form::open(
29                  array('route'=>array('visits.destroy',$visit->id),
30                  'style'=>'display:inline'))}}
31                  {{Form::hidden('_method','DELETE')}}
32                  {{Form::deleteBtn('btn-xs','delete')}}
33                  {{Form::close()}}
34                  @endif
35                </td>
36                <td>
37                  @foreach ($visit->attendingPhysicians as $physician)
38                  {{$physician->last_name}}, {{$physician->first_name}} <br />
```

```
                @endforeach
              </td>
            </tr>

            @endforeach
          </tbody>
        </table>
        <div class="text-center">
          {{  $visits->links()}}</div>

        </div>
      </div>
    </div>
    @stop
      @section('footer')
        @parent
    @stop
```

Listing 88: ../portal/app/views/patients/search.blade.php

```
@extends("layout/bootstrap")
  @section("head")
    @parent
    @stop
  @section('title')
    @parent
            - Search
@stop
  @section("content")
<div class="row">
  <div class="col-lg-offset-2 col-lg-8">
    <div class="well">

      {{ Form::open([
        "route" => array("patients.search"),
        "method" =>"get",
        "class"=>"form-horizontal",
        "autocomplete"=>"off",
        "id"=>"search",
          ]) }}

        <fieldset>
          <legend>Search</legend>

          <div class="form-group {{$errors->has('last_name') ? 'has-error': ''}}">
            <label for='last_name' class='col-sm-3 control-label'>Last Name <span class="required glyphicon
                 glyphicon-asterisk"></span></label>
            <div class="col-sm-9">
              {{Form::text('last_name',null,array('class'=>'form-control'))}}
              @if($errors->has('last_name'))
              <span class="help-inline label label-danger">
                {{$errors->first('last_name')}}
              </span>
              @endif
            </div>
```

```
35            </div>
36            <div class="form-group {{$errors->has('first_name') ? 'has-error': ''}}">
37              {{Form::label('first_name','First Name',array('class'=>'col-sm-3 control-label'))}}
38              <div class="col-sm-9">
39                {{Form::text('first_name',null,array('class'=>'form-control'))}}
40                @if($errors->has('first_name'))
41                <span class="help-inline label label-danger">
42                  {{$errors->first('first_name')}}
43                </span>
44                @endif
45              </div>
46            </div>
47          </fieldset>
48              <div class="form-group">
49                <div class="col-sm-offset-3 col-sm-9">
50                  {{ Form::submit("search",array("id"=>"submit","class"=>"btn btn-default")) }}
51                </div>
52              </div>
53              {{Form::close()}}
54              @if(isset($results))
55
56              <table class="table table-bordered table-hover">
57                <thead>
58                  <tr>
59                    <th>{{ColumnPresenter::show('hospital_id','ID',URL::route('patients.search'),Input::query())
                           }}</th>
60                    <th>{{ColumnPresenter::show('last_name','Name',URL::route('patients.search'),Input::query())
                           }}</th>
61
62                    <th><b>Date of Birth</b></th>
63                    <th><b>Last Visit</b></th>
64                  </tr>
65                </thead>
66                <tbody>
67                  @foreach ($results as $result)
68                  <tr>
69                    <td>
70                      <a href="{{ URL::route('patients.show',["patients"=>$result->id]) }}">
71                        {{$result->hospital_id}}
72                      </a>
73                      @if(Sentry::getUser()->hasAccess('create_patient'))
74                      {{Form::open(array('route'=>array('patients.edit',$result->id),'method'=>'get','style'=>'
                            display:inline'))}}{{Form::editBtn('btn-xs','')}}
75                      {{Form::close()}}
76                      @endif
77                    </td>
78                    <td> <b><a href="{{ URL::route('patients.show',["patients"=>$result->id]) }}">{{ $result->
                           last_name }}, {{$result->first_name}} {{$result->middle_name}}</a></b>
79
80                    </td>
81                    <td>{{$result->birthdate}}</td>
82                    <td>
83                      <div class="text-right">
84                      @if($result->visits()->first()!=null)
85                        {{$result->visits()->first()->date_of_visit}}
86                      @endif
```

```
                    @if(Sentry::getUser()->hasAccess('create_patient'))
                      {{Form::open(array(
                      'route'=>array('visits.create',$result->id),
                      'method'=>'GET','style'=>'display:inline'))}}
                      {{Form::editBtn('btn-xs','add')}}
                      {{Form::close()}}
                    @endif
                  </div>
                  </td>
                </tr>
                @endforeach
              </tbody>
              </table>
              <div class="text-center">
            {{  $results->links()}}</div>
              @endif
    </div>
  </div>
</div>

    @stop
    @section('footer')
    @parent

    @stop
```

Listing 89: ../portal/app/views/patients/show.blade.php

```
@extends("layout/bootstrap")
  @section("head")
    @parent
@stop
  @section('title')
    @parent
- Patient Record
@stop
  @section("content")

      <div class="row">
        <div class="col-md-offset-2 col-md-8">
          <div class="panel panel-default">
            <div class="panel-heading">
              <div class="row">
                <div class="col-md-8">
                  <h4><b>{{$patient->last_name}},
                  {{$patient->first_name}}
                  {{$patient->middle_name}} ({{$patient->hospital_id}})</b></h4>
                </div>
                <div class="col-md-2">
                  @if(Sentry::getUser()->hasAccess('create_patient'))
                  {{Form::open(array(
                    'route'=>array('patients.edit',$patient->id),
                    'method'=>'GET'))}}
                    {{Form::editBtn('btn-block','edit')}}
                    {{Form::close()}}
```

```
                    @endif
                </div>
                <div class="col-md-2">
                    @if(Sentry::getUser()->hasAccess('delete_patient'))
                    {{Form::open(array('route'=>array('patients.destroy',$patient->id)))}}
                    {{Form::hidden('_method', 'DELETE') }}
                    {{Form::deleteBtn('btn-block','delete')}}
                    {{Form::close()}}
                    @endif
                  </div>
              </div>
            </div>
            <div class="panel-body">
                <p><b>Patient ID</b>: {{$patient->hospital_id}}</p>
                <p><b>Date of Birth</b>: {{$patient->birthdate}}</p>
                <p><b>Sex:</b> {{$patient->sex}}</p>
                <p><b>Birthplace:</b> {{$patient->birthplace}}</p>
                <p><b>Civil Status</b>: {{$patient->civil_status}}</p>









              </div>


            </div>
          </div>
        </div>

      <div class="row">
        <div class="col-lg-offset-2 col-lg-8">
          <div class="row">
          <div class="col-lg-4">
            <div class="panel panel-default">
              <div class="panel-heading">
                <h3 class="panel-title">Patient Information</h3>
              </div>
              <div class="panel-body">

                <p><b>Telephone Number</b>: {{$patient->telephone_number}}</p>
                <p><b>Mobile Number</b>: {{$patient->mobile_number}}</p>
                <p><b>Religion</b>: {{$patient->religion}}</p>
                <p><b>Nationality</b>: {{$patient->nationality}}</p>
                <p><b>Occupation</b>: {{$patient->occupation}}</p>
                <p><b>Company</b>: {{$patient->company_name}}</p>
                <p><b>Address</b>: {{$patient->address}}</p>
                <p><b>Religion</b>: {{$patient->religion}}</p>
              </div>
            </div>
          </div>
          <div class="col-lg-4">
            <div class="panel panel-default">
              <div class="panel-heading">
                <h3 class="panel-title">Contact Information</h3>
              </div>
              <div class="panel-body">
```

```
                <p><b>Father's Name</b>: {{$patient->father_name}}</p>
                <p><b>Mother's Name</b>: {{$patient->mother_name}}</p>
                <p><b>Parents' Address</b> {{$patient->parent_address}}</p>
                <p><b>Spouse's Name</b>: {{$patient->spouse_name}}</p>
                <p><b>Spouse's Address</b>: {{$patient->spouse_address}}</p>
                <p><b>Spouse's Occupation</b>: {{$patient->spouse_occupation}}</p>
                <p><b>Spouse's Company</b>: {{$patient->spouse_company}}</p>
              </div>
            </div>
          </div>
          <div class="col-lg-4">
            <div class="panel panel-default">
              <div class="panel-heading">
                <h3 class="panel-title">Contact Information</h3>
              </div>
              <div class="panel-body">
                <p><b>Person Responsible for Account</b>: {{$patient->account_responsibility_name}}</p>
                <p><b>Address</b>: {{$patient->account_responsibility_address}}</p>
                <p><b>Number</b>: {{$patient->account_responsibility_number}}</p>


                <p><b>Emergency Contact</b>: {{$patient->emergency_contact_name}}</p>
                <p><b>Address</b>: {{$patient->emergency_contact_address}}</p>
                <p><b>Number</b>: {{$patient->emergency_contact_number}}</p>
              </div>
            </div>
          </div>
          </div>
        </div>


        <div class="row">
          <div class="col-lg-offset-2 col-lg-8">
            <div class="row">
              <div class="col-lg-10"><h3>Visits</h3></div>
              <div class="col-lg-2">
                @if(Sentry::getUser()->hasAccess('create_patient'))
                  {{Form::open(array(
                  'route'=>array('visits.create',$patient->id),
                  'method'=>'GET'))}}
{{Form::editBtn('btn-lg','add')}}               {{Form::close()}}
              @endif
            </div>
          </div>
        </div>
        <div class="row">
          <div class="col-lg-offset-2 col-lg-8">

            @if($visits->count()>0)
          <table id="visit-table" class="table table-bordered table-hover">
            <thead>
              <th><b>{{ColumnPresenter::show('id','ID',URL::route('patients.show',$patient->id),Input::query())}}</
                   b></th>
              <th><b>{{ColumnPresenter::show('date_of_visit','Date of Visit',URL::route('patients.show',$patient->
                   id),Input::query())}}</b></th>
```

```
138             @if(Sentry::getUser()->inGroup(Sentry::findGroupByName('hospital_admin'))||Sentry::getUser()->inGroup
                    (Sentry::findGroupByName('medical_technicians'))||Sentry::getUser()->inGroup(Sentry::
                    findGroupByName('doctors')))
139             <th><b>Chief Complaint</b></th>
140             <th><b>Admitting Diagnosis</b></th>
141             @endif
142             <th><b>Attending Physicians</b></th>
143           </thead>
144           <tbody>
145
146             @foreach($visits as $visit)
147             <tr class='
148             <?
149               switch($visit->status){
150               case "pending":
151               echo "warning";
152               break;
153               case "waiting":
154
155               break;
156               case "closed":
157               echo "success";
158               break;}
159             ?>
160
161           '>
162                    <td>{{$visit->id}}</td>
163               <td>{{link_to_route('visits.show',$visit->date_of_visit,array('visits'=>$visit->id))}}
164
165               </td>
166               @if(Sentry::getUser()->inGroup(Sentry::findGroupByName('hospital_admin'))||Sentry::getUser()->
                      inGroup(Sentry::findGroupByName('medical_technicians'))||Sentry::getUser()->inGroup(Sentry::
                      findGroupByName('doctors')))
167               <td>{{$visit->chief_complaint}}</td>
168               <td>{{$visit->admitting_diagnosis}}</td>
169               @endif
170               <td>
171
172                 @foreach ($visit->attendingPhysicians as $physician)
173                 {{$physician->last_name}}, {{$physician->first_name}} <br />
174                 @endforeach
175               </td>
176             </tr>
177             @endforeach
178           </tbody>
179         </table>
180         @endif
181         <div class="text-center">
182
183             {{$visits->links()}}
184           </div>
185         </div>
186       </div>
187   @stop
188     @section('footer')
189   @parent
```

```
190       @include('layout/confirm-delete')
191       @stop
```

Listing 90: ../portal/app/views/patients/waiting.blade.php

```
1   @extends("layout/bootstrap")
2     @section("head")
3       @parent
4   @stop
5   @section('title')
6     @parent
7   - Patients Awaiting Results
8   @stop
9     @section("content")
10  <div class="row">
11    <div class="col-lg-offset-2 col-lg-8">
12      <div class="well">
13        <table class="table table-bordered table-striped">
14          <thead>
15            <th>Patient ID</th>
16            <th>Name </th>
17            <th>Date of Visit</th>
18            <th>Added By</th>
19            <th>Status</th>
20          </thead>
21          <tbody>
22            @foreach($visits as $visit)
23            <tr>
24              <td>{{link_to_route('patients.show',$visit->patient->hospital_id,array('patients'=>$visit->patient->
                    id))}}</td>
25              <td>{{link_to_route('diagnose',$visit->patient->last_name.", ". $visit->patient->first_name,array('
                    visits'=>$visit->id))}}</td>
26              <td>
27                {{link_to_route('diagnose',$visit->date_of_visit,array('visits'=>$visit->id))}}
28              </td>
29              <td>
30                {{Employee::find($visit->added_by)->last_name}},
31                {{Employee::find($visit->added_by)->first_name}}
32              </td>
33              <td>
34                {{$visit->status}}
35              </td>
36            </tr>
37
38            @endforeach
39          </tbody>
40        </table>
41        <div class="text-center">
42      {{  $visits->links()}}</div>
43
44      </div>
45    </div>
46  </div>
47  @stop
48    @section('footer')
```

```
49      @parent
50        @stop
```

Listing 91: ../portal/app/views/patients/waiting2.blade.php

```
1   @extends("layout/bootstrap")
2     @section("head")
3       @parent
4   @stop
5   @section('title')
6     @parent
7   - Patients Awaiting Results
8   @stop
9     @section("content")
10  <div class="row">
11    <div class="col-lg-offset-2 col-lg-8">
12      <div class="well">
13        <table class="table table-bordered table-striped">
14          <thead>
15            <th>Patient ID</th>
16            <th>Name </th>
17            <th>Date of Visit</th>
18            <th>Added By</th>
19            <th>Status</th>
20          </thead>
21          <tbody>
22            @foreach($visits as $visit)
23            <tr>
24              <td>{{link_to_route('patients.show',$visit->patient->hospital_id,array('patients'=>$visit->patient->
                    id))}}</td>
25              <td>{{link_to_route('diagnose',$visit->patient->last_name.", ". $visit->patient->first_name,array('
                    visits'=>$visit->id))}}</td>
26              <td>
27                {{link_to_route('diagnose',$visit->date_of_visit,array('visits'=>$visit->id))}}
28              </td>
29              <td>
30                {{Employee::find($visit->added_by)->last_name}},
31                {{Employee::find($visit->added_by)->first_name}}
32              </td>
33              <td>
34                {{$visit->status}}
35              </td>
36            </tr>
37
38            @endforeach
39          </tbody>
40        </table>
41        <div class="text-center">
42      {{  $visits->links()}}</div>
43
44      </div>
45    </div>
46  </div>
47  @stop
48    @section('footer')
```

```
49      @parent
50        @stop
```

Listing 92: ../portal/app/views/requests/inbox.blade.php

```
1   @extends("layout/bootstrap")
2     @section("head")
3       @parent
4   @stop
5   @section('title')
6     @parent
7   - Inbox
8   @stop
9     @section("content")
10  <div class="row">
11    <div class="col-lg-offset-2 col-lg-8">
12      <div class="well">
13        <table class="table table-bordered table-striped">
14          <thead>
15            <th>ID</th>
16            <th>Test Type</th>
17            <th>Patient Name </th>
18            <th>Date of Visit</th>
19            <th>Doctor</th>
20          </thead>
21          <tbody>
22            @foreach($requests as $request)
23            <tr>
24              <td>{{link_to_route('results.create',$request->id,array('requests'=>$request->id))}}
25              <td>{{link_to_route('results.create',$request->type->test_name,array('requests'=>$request->id))}}
26              <?$typeFields = json_decode($request->type->fields) ?>
27                @if($request->fields)
28                <?$fields = json_decode($request->fields,true)?>
29                <?
30                $count = count($fields);
31                //print_r($fields);
32                $i = 0;
33                if($fields!=null){
34
35                  echo"(";
36                foreach($fields as $field)
37                {
38                  echo $typeFields->$field->field_name;
39
40                  if($i< $count-1)
41                  {
42                    echo ', ';
43                  }
44                  $i++;
45                }
46                echo(")");
47                }
48                ?>
49                @endif
50
```

```
51              </td>
52              <td>{{$request->visit->patient->last_name}}, {{$request->visit->patient->first_name}}</td>
53              <td>{{$request->visit->date_of_visit}}</td>           <td>
54                @foreach ($request->visit->attendingPhysicians as $physician)
55                {{$physician->last_name}}, {{$physician->first_name}} <br />
56                @endforeach
57              </td>
58            </tr>
59            @endforeach
60          </tbody>
61        </table>
62
63      </div>
64    </div>
65  </div>
66  @stop
67    @section('footer')
68      @parent
69        @stop
```

Listing 93: ../portal/app/views/requests/outbox.blade.php

```
1   @extends("layout/bootstrap")
2     @section("head")
3       @parent
4   @stop
5   @section('title')
6     @parent
7   - Outbox
8   @stop
9     @section("content")
10  <div class="row">
11    <div class="col-lg-offset-2 col-lg-8">
12      <div class="well">
13        <table class="table table-bordered table-striped">
14          <thead>
15            <th>ID</th>
16            <th>Test Type</th>
17            <th>Patient Name </th>
18            <th>Date of Visit</th>
19            <th>Doctor</th>
20          </thead>
21          <tbody>
22            @foreach($requests as $request)
23            <tr>
24              <td>{{link_to_route('results.show',$request->id,array('results'=>$request->result->id))}}
25              <td>{{link_to_route('results.show',$request->type->test_name,array('results'=>$request->result->id))
                    }}
26              <?$typeFields = json_decode($request->type->fields) ?>
27                @if($request->fields)
28                <?$fields = json_decode($request->fields,true)?>
29                <?
30                $count = count($fields);
31                //print_r($fields);
32                $i = 0;
```

```
33                if($fields!=null){
34
35                  echo"(";
36                foreach($fields as $field)
37                {
38                  echo $typeFields->$field->field_name;
39
40                  if($i< $count-1)
41                  {
42                    echo ', ';
43                  }
44                  $i++;
45                }
46                echo(")");
47                }
48                ?>
49                @endif
50
51              </td>
52              <td>{{$request->visit->patient->last_name}}, {{$request->visit->patient->first_name}}</td>
53              <td>{{$request->visit->date_of_visit}}</td>           <td>
54                @foreach ($request->visit->attendingPhysicians as $physician)
55                {{$physician->last_name}}, {{$physician->first_name}} <br />
56                @endforeach
57              </td>
58            </tr>
59            @endforeach
60          </tbody>
61        </table>
62
63      </div>
64    </div>
65  </div>
66  @stop
67    @section('footer')
68      @parent
69        @stop
```

Listing 94: ../portal/app/views/results/create.blade.php

```
1   @extends("layout/bootstrap")
2     @section("head")
3       @parent
4
5         @stop
6           @section('title')
7             @parent
8               - Add Test Result
9   @stop
10    @section("content")
11  <div class="row">
12    <div class="col-lg-offset-2 col-lg-8">
13      <div class="well">
14        {{Form::open(array('files'=>'true',"enctype"=>"multipart/form-data",'class'=>'form-horizontal','url'=>'
              results'))}}
```

```
15        {{Form::hidden('added_by',$added_by)}}
16        {{Form::hidden('test_request_id',$request->id)}}
17        <?
18        $fields = json_decode($request->fields,true);
19        $typeFields = json_decode($request->type->fields);
20
21
22
23        foreach(json_decode($request->type->fields,true) as $field => $data)
24        {
25          ?>
26
27          <div class="form-group {{$errors->has($field) ? 'has-error' : ''}}">
28            <label class="col-sm-2 control-label" for '{{$field}}'>
29                {{$data['field_name']}}
30            </label>
31              @if($data['data_type']=='image')
32              <div class="col-sm-10">
33
34                {{Form::file($field,null,array('id'=>$field,'class'=>'form-control'))}}
35                @if($errors->has($field))
36                  <span class="help-inline label label-danger">
37                  {{$errors->first($field)}}
38                  </span>
39                @endif
40              </div>
41              @elseif($data['data_type']=='selection')
42                <div class="col-sm-10">
43
44
45                    <input name="{{$field}}" type="text" id="{{$field}}" class="form-control" value="{{Input::old(
                          $field)}}"></input>
46                @if($errors->has($field))
47                  <span class="help-inline label label-danger">
48                  {{$errors->first($field)}}
49                  </span>
50                @endif
51
52                </div>
53
54
55              @elseif($data['data_type']=='range')
56              <div class="col-sm-10">
57                <div class="input-group">
58                  <input name="{{$field}}" type="text" id="{{$field}}" class="form-control" value="{{Input::old(
                        $field)}}"></input>
59
60                  @if(isset($data['unit']))
61                              <span class="input-group-addon">{{$data['unit']}}</span>
62                  @endif
63
64                </div>
65                @if($errors->has($field))
66                  <span class="help-inline label label-danger">
67                  {{$errors->first($field)}}
68                  </span>
```

```
                @endif
              </div>
              @endif


          </div>


          <?
        }

        ?>
        <div class="form-group">
            <div class="col-sm-offset-2 col-sm-10">
              <button type="submit" class="btn btn-default">Submit</button>
            </div>
          </div>
          {{Form::close()}}
      </div>
    </div>
</div>
      @stop
        @section('footer')


      @parent
        <script>

        $(document).ready(function(){
          @foreach(json_decode($request->type->fields,true) as $field => $data)

        $(function(){

          $("#{{$field}}").autocomplete({source:'../../types/values/{{$request->type->id}}/{{$field}}',delay:500});
        });
        @endforeach
      });
        </script>
      @stop
```

Listing 95: ../portal/app/views/results/report.blade.php

```
@extends("layout/bootstrap")
  @section("head")
    @parent

@stop
  @section('title')
    @parent

      @stop
        @section("content")
<div class="row">
  <div class="col-lg-offset-2 col-lg-8">
    <div class="well">
```

```
14        {{Form::open(array('method'=>'get'))}}
15        <div class="form-group">View report for the
16        <select name='period' id="period">
17          <option value='' disabled selected>select span</option>
18          <option value="year">year</option>
19          <option value="month">month</option>
20          <option value="week">week</option>
21          <option value="day">day</option>
22
23        </select>
24        <span id="yearspan"></span> <span id="monthspan"></span> <span id="dayspan"></span></div>
25        <div class="form-group">
26        {{Form::submit('Go',array('class'=>'btn btn-default'))}}
27        {{Form::close()}}
28
29        </div>
30        @if(Input::get())
31        <h2>Report for the {{Input::get('period')}} starting {{Input::get('year')? Input::get('year'): ""}}{{Input
              ::get('month')? '/'.Input::get('month'): ""}}{{Input::get('day')? '/'.Input::get('day'): ""}}</h2>
32        <?if(isset($results)&&$results!=null)
33        {
34          echo "<h3>Tests Performed</h3>";
35          $counter = array();
36          foreach($results as $result)
37          { if(!isset($counter[$result->request->test_type]))
38            {
39              $counter[$result->request->test_type] = 0;
40            }
41            $counter[$result->request->test_type]++;
42          }
43
44          foreach($counter as $key=>$count)
45          {
46
47            echo "<b>".Type::find($key)->test_name."</b>: ".$count."<br />";
48          }
49        }
50
51
52          echo "<h3>Patient Stats</h3>";
53          echo "<b> Number of Visits:</b> ". $numvisits."<br />";
54          echo "<b> Number of Unique Visitors</b>: ".$distinct."<br/>";
55
56
57        ?>
58        @endif
59      </div>
60    </div>
61  </div>
62  @stop
63  @section('footer')
64
65    @parent
66    {{HTML::script('js/report.js')}}
67      @stop
```

Listing 96: ../portal/app/views/results/search.blade.php

```
@extends("layout/bootstrap")
  @section("head")
    @parent
    @stop
  @section('title')
    @parent
            - Search
@stop
  @section("content")
<div class="row">
  <div class="col-lg-offset-2 col-lg-8">
    <div class="well">

      <br />
      <br />
      <br />
            {{ Form::open([
              "method" =>"post",
              "class"=>"form-horizontal",
              "autocomplete"=>"off"
                ]) }}
                <label for="test_type" class="col-sm-3 control-label">Test Type <span class="glyphicon glyphicon-
                    asterisk required"></span></label>
                  {{Form::select('test_type',$types,null,array('class'=>'form-control'))}}

                  <label for="physician">
                    Physician</label>
                    {{Form::select('physician',$physicians,'',array('class'=>'form-control'))}}
                      {{Form::submit('submit',array('class'=>'btn btn-default'))}}




          {{Form::close()}}

    </div>
  </div>
</div>
```

Listing 97: ../portal/app/views/results/show.blade.php

```
@extends("layout/bootstrap")
  @section("head")
    @parent

@stop
@section('title')
@parent

@stop
  @section("content")

```

```
<div class="row">
  <div class="col-lg-offset-2 col-lg-8">
    <div class="well">

        <h3><img src="{{asset('img/logo.png')}}" /> Las Pi&ntilde;as Doctors Hospital</h3>
        <p> <small><small>#6009 J. AGUILAR AVENUE CAA ROAD, PULANG LUPA II, LAS PI&Ntilde;AS CITY<BR />
            TEL NOS: 825-52-36 ; 825-52-93<BR />
            TIN: 053-00-412-964 ; TIN 053-000-412-964-VAT</small></SMALL></p>

        <h4>{{$result->request->type->test_name}}  **{{$result->request->type->department->department_name}}**</
             h4>

        <b>Name:</b> {{$result->request->visit->patient->last_name}}, {{$result->request->visit->patient->
             first_name}}

        <br/>



        <b>Date:</b> {{$result->created_at}}
        <br />
        <b>Patient No:</b> {{$result->request->visit->patient->hospital_id}}
        <br />
        <b>Physician:</b>
        @foreach($result->request->visit->attendingPhysicians as $physician)
        {{$physician->first_name." ".$physician->last_name}}
        @endforeach
        <br />
        <b>Sex:</B> {{$result->request->visit->patient->sex}}<br/> <br />

        <table class="table table-bordered table-striped"><thead>
          <th>Field</th>
          <th>Patient's Value</th>
          <th>Comparison Value</th>
        </thead>
          <tbody>
        <?
        $typeFields = json_decode($result->request->type->fields);

        foreach(json_decode($result->request->type->fields,true) as $key=>$field)
        {
          ?>
          <tr>
            <td>
          <?
          $field_name = $typeFields->$key->field_name;

          echo $field_name;

          ?>
        </td>
        <td>
          <?
            switch($typeFields->$key->data_type)
            {
              case "image":
                echo
```

```
66                    "<a href='".asset('uploads/'.$resultData->$key."' class='thumbnail'>"
67                      .HTML::image('uploads/'.substr($resultData->$key,0,-4).'-thumb.jpg')."</a>";
68               break;
69               case "range":
70               case "int":
71               case "float":
72                 echo "<b>";
73                 echo $resultData->$key;
74                 echo "</b> ";
75                 if(isset($typeFields->$key->unit))
76                   echo $typeFields->$key->unit;
77               break;
78               case "varchar":
79               case "selection":
80                 echo "<b>";
81                 echo $resultData->$key;
82                 echo "</b>";
83
84               break;
85             }
86           ?>
87         </td>
88
89         <td>
90           @if(isset($typeFields->$key->comparison_value))
91           <?
92           echo $typeFields->$key->comparison_value;
93           ?>
94           @endif
95         </td>
96         </tr>
97           <?
98         }     ?>
99
100       </tbody>
101       </table>
102
103       <b>Medical Technologist</b>: {{Employee::where('id',$result->added_by)->first()->last_name}}, {{Employee::
              where('id',$result->added_by)->first()->first_name}}
104     </div>
105   </div>
106 </div>
107     @stop
108 @section('footer')
109
110 @parent
111   <script type="text/javascript">
112 $(document).ready(function() {
113
114
115   /* Using custom settings */
116
117   $("a.thumbnail").fancybox({
118     'hideOnContentClick': true
119   });
120
```

```
121
122
123 });
124
125 </script>
126 @stop
```

Listing 98: ../portal/app/views/types/create.blade.php

```
1  @extends("layout/bootstrap")
2  @section("head")
3  @parent
4
5  @stop
6  @section('title')
7  @parent
8    - Test Types
9  @stop
10 @section("content")
11 <div class="row">
12   <div class="col-lg-offset-2 col-lg-8">
13     <div class="well">
14         {{Form::open(
15         array(
16           'autocomplete'=>'off',
17           'url'=>'types',
18           'method'=>'post',
19           'id'=>'form'
20         ))}}
21         <fieldset>
22           <input type="hidden" value="0" id="count" name="count"></input>
23           <div class="form-group">
24             <legend>Basic Information</legend>
25             <label class="control-label" for="test_name">Name of Test</label>
26             <input required class="required form-control" name="test_name" type="text">
27
28           </div>
29           <div class="form-group">
30                 <label for="department_id">Department</label>
31                 <select class="form-control" name='department_id'>
32                 @foreach($departments as $department)
33                 <option value='{{$department->id}}'>
34                   {{$department->department_name}}
35                 </option>
36                 @endforeach
37                 </select>
38               </div>
39             <!-- <div class="form-group">
40               <div class="checkbox">
41               <label><input type="checkbox" name="no_comparison"  id="no-comparison">No comparison values</label>
42             </div>
43             </div> -->
44         </fieldset>
45         <div id="fields">
46
```

```
47
48          </div>
49          <p><a href="#" id="add"><span>Add another field</span></a> | <a href="#" id="remove">Remove a field</a>
50          </p>
51
52
53          {{Form::submit('Create',array('class'=>'btn btn-default'))}}
54        {{Form::close()}}
55      </div>
56    </div>
57  </div>
58  @stop
59  @section('footer')
60    @parent
61      {{HTML::script('components/jquery-validation/jquery.validate.js')}}
62      {{HTML::script('js/types.js')}}
63  @stop
```

Listing 99: ../portal/app/views/types/dropdown.blade.php

```
1  @foreach($types as $type)
2    <option value="{{$type->id}}">{{$type->test_name}}</option>
3  @endforeach
```

Listing 100: ../portal/app/views/types/index.blade.php

```
1  @extends("layout/bootstrap")
2  @section("head")
3  @parent
4
5  @stop
6  @section('title')
7  @parent
8    - Test Types
9  @stop
10 @section("content")
11 <div class="row">
12   <div class="col-lg-offset-2 col-lg-8">
13     <div class="well">
14       <table class="table table-bordered table-striped">
15         <thead>
16           <th>Name</th>
17           <th>Department</th>
18         </thead>
19         <tbody>
20         @foreach($types as $type)
21         <tr>
22
23           <td>{{link_to_route('types.show',$type->test_name,array('types'=>$type->id))}}</td>
24           <td>{{$type->department->department_name}}</td>
25         </tr>
26
27         @endforeach
28       </tbody>
```

```
29        </table>
30      </div>
31    </div>
32  </div>
33  @stop
34  @section('footer')
35    @parent
36  @stop
```

Listing 101: ../portal/app/views/types/show.blade.php

```
1   @extends("layout/bootstrap")
2   @section("head")
3   @parent
4
5   @stop
6   @section('title')
7   @parent
8     - Test Types
9   @stop
10  @section("content")
11  <div class="row">
12    <div class="col-lg-offset-2 col-lg-8">
13      <div class="well">
14        <table class="table table-bordered table-striped"><thead>
15          <th>Field</th>
16          <th>Data Type</th>
17          <th>Comparison Value</th>
18        </thead>
19        <tbody>
20          <?  $typeFields =json_decode($type->fields);?>
21          @foreach(json_decode($type->fields,true) as $key=>$field)
22          <tr><td>
23            {{$typeFields->$key->field_name}}
24          </td>
25        <td>{{$typeFields->$key->data_type}}</td>
26      <td>
27        <?
28          if(isset($typeFields->$key->comparison_value))
29          {
30            echo $typeFields->$key->comparison_value;
31          }
32          else if(isset($typeFields->$key->preset_values)){
33            foreach($typeFields->$key->preset_values as $value){
34              echo $value."<br />";
35            }
36          }
37          ?>
38      </td></tr>
39        @endforeach
40        </tbody>
41        </table>
42      </div>
43
44    </div>
```

```
45  </div>
46  @stop
47  @section('footer')
48    @parent
49      {{HTML::script('components/jquery-validation/jquery.validate.js')}}
50      {{HTML::script('js/types.js')}}
51  @stop
```

Listing 102: ../portal/app/views/users/create.blade.php

```
1   @extends("layout/bootstrap")
2   @section("head")
3   @parent
4   @stop
5   @section('title')
6   @parent
7   - Add New Account
8   @stop
9   @section("content")
10    <div class="row">
11      <div class="col-lg-offset-2 col-lg-8">
12        <div class="well">
13        {{ Form::open([
14          "id"=>"create",
15          "url"=>"users",
16          "method" =>"post",
17          "class"=>"form-horizontal",
18          "autocomplete"=>"off"
19            ]) }}
20          <fieldset>
21            <legend>Add New Account</legend>
22            <div class="form-group  {{$errors->has('email') ? 'has-error': ''}}">
23              <label for="email" class="col-sm-3 control-label">Email <span class="glyphicon glyphicon-asterisk
                    required"></span></label>
24              <div class="col-sm-9">
25                {{Form::text('email',null,array('class'=>'form-control'))}}
26                @if($errors->has('email'))
27                <span class="help-inline label label-danger">
28                  {{$errors->first('email')}}
29                </span>
30                @endif
31              </div>
32            </div>
33            <div class="form-group  {{$errors->has('last_name') ? 'has-error': ''}}">
34  <label for="last_name" class="col-sm-3 control-label">Last Name <span class="glyphicon glyphicon-asterisk
        required"></span></label>           <div class="col-sm-9">
35                {{Form::text('last_name',null,array('class'=>'form-control'))}}
36                @if($errors->has('last_name'))
37                <span class="help-inline label label-danger">
38                  {{$errors->first('last_name')}}
39                </span>
40                @endif
41              </div>
42            </div>
43            <div class="form-group  {{$errors->has('first_name') ? 'has-error': ''}}">
```

```
44              <label for="first_name" class="col-sm-3 control-label">First Name <span class="glyphicon glyphicon-
                     asterisk required"></span></label>
45              <div class="col-sm-9">
46                  {{Form::text('first_name',null,array('class'=>'form-control'))}}
47                  @if($errors->has('first_name'))
48                  <span class="help-inline label label-danger">
49                    {{$errors->first('first_name')}}
50                  </span>
51                  @endif
52              </div>
53            </div>
54            <div class="form-group  {{$errors->has('account_type') ? 'has-error': ''}}">
55              <label for="account_type" class="col-sm-3 control-label">Account Type <span class="glyphicon
                     glyphicon-asterisk required"></span></label>
56              <div class="col-sm-9">
57                {{Form::select('account_type',array(''=>'Select One','admin'=>'System Administrator','clerks'=>'
                       Clerk','medical_technicians'=>'Medical Technician','doctors'=>'Doctor'),'',array('class'=>'
                       form-control'))}}
58                @if($errors->has('account_type'))
59                <span class="help-inline label label-danger">
60                  {{$errors->first('account_type')}}
61                </span>
62                @endif
63              </div>
64            </div>
65
66            <div class="form-group {{$errors->has('department')}}">
67              <label for="department" class="col-sm-3 control-label">Department</label>
68              <div class="col-sm-9">
69              {{Form::select('department',$departments,null,array('class'=>'form-control'))}}
70              @if($errors->has('department'))
71
72              @endif
73              </div>
74            </div>
75            <div class="form-group">
76                  <div class="col-sm-offset-3 col-sm-9">
77                    {{ Form::submit("submit",array("id"=>"submit","class"=>"btn btn-default")) }}
78                  </div>
79            </div>
80          </fieldset>
81        {{ Form::close() }}
82
83        <div id="info">
84
85        </div>
86        </div>
87      </div>
88    </div>
89  @stop
90    @section('footer')
91      @parent
92  {{HTML::script('js/forms.js')}}
93
94      @stop
```

Listing 103: ../portal/app/views/users/created.blade.php

```
1  @extends("layout/bootstrap")
2  @section("head")
3  @parent
4  @stop
5  @section('title')
6  @parent
7  - New Account Created
8  @stop
9  @section("content")
10   <div class="row">
11     <div class="col-lg-offset-2 col-lg-8">
12       <div class="well">
13         <p><b>Username</b>: {{$username}}</p>
14         <p><b>Email</b>: {{$email}}</p>
15         <p><b>Password</b>: {{$password}}</p>
16         <p>This window will only appear once. Click {{link_to_route('users.create','here')}} to return to the
               account creation page, or click {{link_to_route('users.show','here',array('users'=>$id))}} to go to
               the user profile.</p>
17       </div>
18     </div>
19   </div>
20       @stop
21 @section('footer')
22   @parent
23   {{HTML::script('js/forms.js')}}
24   @stop
```

Listing 104: ../portal/app/views/users/edit.blade.php

```
1  @extends("layout/bootstrap")
2    @section("head")
3      @parent
4  @stop
5    @section('title')
6      @parent
7  - Edit
8  @stop
9    @section("content")
10 <div class="row">
11   <div class="col-md-offset-2 col-md-8">
12     <div class="well">
13         {{ Form::open(array(
14                'action' => array('UserController@update', $user->id),
15                'method' => 'put',
16                'class' => 'form-horizontal',
17                'role' => 'form'
18                )) }}
19
20         @if($user->inGroup(Sentry::getGroupProvider()->findByName('patients')))
21         <?
22           $lastName = $user->patient->last_name;
23           $firstName = $user->patient->first_name;
24         ?>
```

```
25          @elseif($user->inGroup(Sentry::getGroupProvider()->findByName('doctors')))
26          <?
27            $lastName = $user->doctor->last_name;
28            $firstName = $user->doctor->first_name;
29          ?>
30          @else
31          <?
32            $lastName = $user->employee->last_name;
33            $firstName = $user->employee->first_name;
34          ?>
35
36          @endif
37
38             <fieldset>
39               <legend>Edit</legend>
40              <div class="form-group {{$errors->has('email') ? 'has-error': ''}}">
41                <label class='col-sm-3 control-label' for='email'>Email <i class="required glyphicon glyphicon-
                       asterisk"></i></label>
42
43                <div class="col-sm-9">
44                  {{Form::text('email',$user->email,array('class'=>'form-control'))}}
45                </div>
46              </div>
47
48              <div class="form-group {{$errors->has('last_name') ? 'has-error': ''}}">
49                <label for='last_name' class='col-sm-3 control-label'>Last Name <span class="required glyphicon
                       glyphicon-asterisk"></span></label>
50
51                <div class="col-sm-9">
52                  {{Form::text('last_name',$lastName,array('class'=>'form-control'))}}
53                  @if($errors->has('last_name'))
54                  <span class="help-inline label label-danger">
55                    {{$errors->first('last_name')}}
56                  </span>
57                  @endif
58                </div>
59              </div>
60
61
62              <div class="form-group {{$errors->has('first_name') ? 'has-error': ''}}">
63                <label for='first_name' class='col-sm-3 control-label'>First Name <i class="required glyphicon
                       glyphicon-asterisk"></i></label>
64
65                {{--for some reason clicking on the label won't auto-focus the input--}}
66                <div class="col-sm-9">
67                  {{Form::text('first_name',$firstName,array('class'=>'form-control'))}}
68                  @if($errors->has('first_name'))
69                  <span class="help-inline label label-danger">
70                    {{$errors->first('first_name')}}
71                  </span>
72                  @endif
73                </div>
74              </div>
75
76
77            <div class="form-group">
```

```
78              <div class="col-sm-offset-3 col-sm-9">
79                {{ Form::submit("submit",array("id"=>"submit","class"=>"btn btn-default")) }}
80              </div>
81            </div>
82             </fieldset>
83             {{Form::close()}}
84      </div>
85    </div>
86  </div>
87
88    @stop
89        @section('footer')
90          @parent
91          @include('layout/confirm-delete')
92      @stop
```

Listing 105: ../portal/app/views/users/forgot.blade.php

```
1   @extends("layout/bootstrap")
2   @section("content")
3   {{HTML::style("css/forms.css");}}
4
5     <div class="row">
6       <div class="col-lg-offset-2 col-lg-8">
7         <div class="well">
8       {{ Form::open([
9           "route"        => "forgot",
10          "autocomplete" => "off",
11      "class" => "form-horizontal"
12      ]) }}
13      <fieldset>
14        <legend>Reset Password</legend>
15        <div class="form-group">
16          <label for='email' class='col-sm-3 control-label'>Email <span class="required glyphicon glyphicon-
                asterisk"></span></label>
17          <div class="col-sm-9">
18            {{Form::text('email',null,array('class'=>'form-control'))}}
19          </div>
20        </div>
21      </fieldset>
22
23      <div class="form-group">
24        <div class="col-sm-offset-3 col-sm-9">
25          {{ Form::submit("Send",array("class"=>"btn btn-default")) }}
26        </div>
27      </div>
28      {{ Form::close() }}
29  </div></div></div>
30  @stop
31  @section("footer")
32      @parent
33  @stop
```

Listing 106: ../portal/app/views/users/index.blade.php

```
@extends("layout/bootstrap")
@section("head")
@parent
@stop
@section('title')
@parent
- Accounts
@stop
@section("content")
<div class="row">
  <div class="col-lg-offset-2 col-lg-8">
    <div class="well">

      {{ Form::open([
        "url" => "users",
        "method" =>"get",
        "class"=>"form-horizontal",
        "autocomplete"=>"off",
        "id"=>"search",
          ]) }}

        <fieldset>
          <legend>Search</legend>

          <div class="form-group {{$errors->has('last_name') ? 'has-error': ''}}">
            <label for='last_name' class='col-sm-3 control-label'>Last Name <span class="required glyphicon
                 glyphicon-asterisk"></span></label>
            <div class="col-sm-9">
              {{Form::text('last_name',null,array('class'=>'form-control'))}}
              @if($errors->has('last_name'))
              <span class="help-inline label label-danger">
                {{$errors->first('last_name')}}
              </span>
              @endif
            </div>
          </div>
          <div class="form-group {{$errors->has('first_name') ? 'has-error': ''}}">
            {{Form::label('first_name','First Name',array('class'=>'col-sm-3 control-label'))}}
            <div class="col-sm-9">
              {{Form::text('first_name',null,array('class'=>'form-control'))}}
              @if($errors->has('first_name'))
              <span class="help-inline label label-danger">
                {{$errors->first('first_name')}}
              </span>
              @endif
            </div>
          </div>
        </fieldset>
            <div class="form-group">
              <div class="col-sm-offset-3 col-sm-9">
                {{ Form::submit("search",array("id"=>"submit","class"=>"btn btn-default")) }}
              </div>
            </div>
            {{Form::close()}}
            <hr />

```

```
<table class="table table-bordered table-hover">
  <thead>
    <th>
      {{ColumnPresenter::show('id','ID',URL::route('users.index'),Input::query())}}
    </th>

    <th>{{ColumnPresenter::show('last_name','Name',URL::route('users.index'),Input::query())}}
</th>
    <th>
      {{ColumnPresenter::show('email','Email',URL::route('users.index'),Input::query())}}
    </th>
    <th>
      {{ColumnPresenter::show('username','Username',URL::route('users.index'),Input::query())}}
    </th>
    <th>
        Group

    </th></thead>
    <tbody>
      @foreach($users as $user)
      <tr>
        <td>
          <a href="{{URL::route('users.show',array('users'=>$user->user_id))}}">
          {{$user->user_id}}
        </a>
        </td>
        <td>
        <b> <a href="{{URL::route('users.show',array('users'=>$user->user_id))}}">
            {{$user->last_name}},
            {{$user->first_name}}</b>
          </a>
        </td>
        <td>
          {{$user->email}}
        </td>
        <td>
          {{$user->username}}
        </td>
        <td>
          @foreach(Sentry::findUserById($user->user_id)->getGroups() as $group)

          {{$group->name}}
          @endforeach
        </td>
      </tr>
      @endforeach
    </tbody>
</table>
  <div class="text-center">
            {{  $users->links()}}
          </div>


    </div>
  </div>
</div>
```

```
112    @stop
```

Listing 107: ../portal/app/views/users/profile.blade.php

```
1   @extends("layout/bootstrap")
2   @section("content")
3
4   @if(isset($visits))
5   <div class="row">
6     <div class="col-lg-4">
7       <div class="well">
8           <h2>Hello, {{Sentry::getUser()->email}} </h2>
9
10      </div>
11    </div>
12
13  </div>
14  @endif
15  @stop
```

Listing 108: ../portal/app/views/users/resend.blade.php

```
1   @extends("layout/bootstrap")
2     @section("head")
3       @parent
4
5   @stop
6     @section('title')
7       @parent
8         - Resend Activation Email
9   @stop
10    @section("content")
11
12      <div class="row">
13        <div class="col-lg-offset-2 col-lg-8">
14          <div class="well">
15        {{ Form::open([
16            "route"        => "resend.post",
17            "autocomplete" => "off",
18        "class" => "form-horizontal"
19        ]) }}
20        <fieldset>
21          <legend>Resend Activation Email</legend>
22          <div class="form-group">
23            <label for='email' class='col-sm-3 control-label'>Email <span class="required glyphicon glyphicon-
                  asterisk"></span></label>
24            <div class="col-sm-9">
25              {{Form::text('email',null,array('class'=>'form-control'))}}
26            </div>
27          </div>
28        </fieldset>
29
30        <div class="form-group">
31          <div class="col-sm-offset-3 col-sm-9">
```

```
32            {{ Form::submit("Send",array("class"=>"btn btn-default")) }}
33          </div>
34        </div>
35        {{ Form::close() }}
36    </div></div></div>
37        @stop
```

Listing 109: ../portal/app/views/users/reset.blade.php

```
1   @extends("layout/bootstrap")
2   @section("content")
3   {{HTML::style("css/forms.css");}}
4
5     <div class="row">
6       <div class="col-lg-offset-2 col-lg-8">
7         <div class="well">
8             {{ Form::open(array(
9             'route'=>'reset.post',
10            'autocomplete'=>'off',
11            'class'=>'form-horizontal')) }}
12          {{Form::hidden('code',$code)}}
13          {{Form::hidden('id',$id)}}
14          <fieldset>
15            <div class="form-group {{$errors->has('password') ? 'has-error': ''}}">
16              <label for='password' class='col-sm-3 control-label'>Password <span class="required glyphicon
                      glyphicon-asterisk"></span></label>
17              <div class="col-sm-9">
18                {{Form::password('password',array('type'=>'password','class'=>'form-control'))}}
19                @if($errors->has('date_admitted'))
20                <span class="help-inline label label-danger">
21                  {{$errors->first('password')}}
22                </span>
23                @endif
24              </div>
25            </div>
26              <div class="form-group {{$errors->has('password_confirmation') ? 'has-error': ''}}">
27                <label for='password_confirmation' class='col-sm-3 control-label'>Repeat Password <span class="
                        required glyphicon glyphicon-asterisk"></span></label>
28                <div class="col-sm-9">
29                  {{Form::password('password_confirmation',array('type'=>'password','class'=>'form-control'))}}
30                  @if($errors->has('password_confirmation'))
31                  <span class="help-inline label label-danger">
32                    {{$errors->first('password_confirmation')}}
33                  </span>
34                  @endif
35                </div>
36              </div>
37          </fieldset>
38          <div class="form-group">
39            <div class="col-sm-offset-3 col-sm-9">
40              {{ Form::submit("reset",array("class"=>"btn btn-default")) }}
41            </div>
42          </div>
43          {{Form::close()}}
44        </div>
```

```
45      </div>
46    </div>
47    @stop
```

Listing 110: ../portal/app/views/users/settings.blade.php

```
1   @extends("layout/bootstrap")
2     @section("head")
3       @parent
4   @stop
5     @section('title')
6       @parent
7   - Settings
8   @stop
9     @section("content")
10  <div class="row">
11    <div class="col-lg-offset-2 col-lg-8">
12      <div class="well">
13        {{Form::open(array('class'=>'form-horizontal','action'=>array('UserController@changeEmail')))}}
14        <fieldset>
15          <legend>Email</legend>
16          <span class="help-block">After completing this form, a message will be sent to the new email you
                 specified. Please click on the link contained in that email to complete the process of changing your
                  email address.</span>
17          <div class="form-group">
18            <label class='col-sm-3 control-label'>Email</label>
19
20            <div class="col-sm-9">
21              <p class="form-control-static">{{Sentry::getUser()->email}}</p>
22            </div>
23          </div>
24          <div class="form-group {{$errors->has('new_email') ? 'has-error': ''}}">
25            <label class='col-sm-3 control-label' for='new_email'>New Email <span class="required glyphicon
                   glyphicon-asterisk"></span></label>
26
27            <div class="col-sm-9">
28              {{Form::text('new_email',null,array('class'=>'form-control'))}}
29            </div>
30          </div>
31          <div class="form-group {{$errors->has('password') ? 'has-error': ''}}">
32            <label for='password' class='col-sm-3 control-label'>Password <span class="required glyphicon glyphicon
                   -asterisk"></span></label>
33            <div class="col-sm-9">
34              {{Form::password('password',array(
35              'type'=>'password',
36              'class'=>'form-control'))}}
37              @if($errors->has('password'))
38              <span class="help-inline label label-danger">
39                {{$errors->first('password')}}
40              </span>
41              @endif
42            </div>
43          </div>
44          <div class="form-group">
45            <div class="col-sm-offset-3 col-sm-9">
```

```
46              {{ Form::submit("submit",array("id"=>"submit","class"=>"btn btn-default")) }}
47            </div>
48          </div>
49        </fieldset>
50        {{Form::close()}}
51
52
53        {{Form::open(array(
54          'route'=>'users.changePassword',
55          'method'=>'POST',
56          'role'=>'form',
57          'class'=>'form-horizontal'))}}
58          <fieldset>
59            <legend>Change Password</legend>
60
61          <div class="form-group {{$errors->has('old_password') ? 'has-error': ''}}">
62            <label for='old_password' class='col-sm-3 control-label'>Password <span class="required glyphicon
                   glyphicon-asterisk"></span></label>
63            <div class="col-sm-9">
64              {{Form::password('old_password',array('type'=>'password','class'=>'form-control'))}}
65              @if($errors->has('old_password'))
66              <span class="help-inline label label-danger">
67                {{$errors->first('old_password')}}
68              </span>
69              @endif
70            </div>
71          </div>
72          <div class="form-group {{$errors->has('new_password') ? 'has-error': ''}}">
73            <label for='new_password' class='col-sm-3 control-label'>New Password <span class="required glyphicon
                   glyphicon-asterisk"></span></label>
74            <div class="col-sm-9">
75              {{Form::password('new_password',array(
76              'class'=>'form-control'))}}
77              @if($errors->has('new_password'))
78              <span class="help-inline label label-danger">
79                {{$errors->first('new_password')}}
80              </span>
81              @endif
82            </div>
83          </div>
84          <div class="form-group {{$errors->has('password_confirmation') ? 'has-error': ''}}">
85              <label for='password_confirmation' class='col-sm-3 control-label'>Repeat Password <span class="
                     required glyphicon glyphicon-asterisk"></span></label>
86              <div class="col-sm-9">
87                {{Form::password('password_confirmation',array('type'=>'password','class'=>'form-control'))}}
88                @if($errors->has('password_confirmation'))
89                <span class="help-inline label label-danger">
90                  {{$errors->first('password_confirmation')}}
91                </span>
92                @endif
93              </div>
94            </div>
95          </fieldset>
96            <div class="form-group">
97              <div class="col-sm-offset-3 col-sm-9">
98                {{ Form::submit("submit",array("id"=>"submit","class"=>"btn btn-default")) }}
```

```
99              </div>
100           </div>
101       {{Form::close()}}
102
103     </div>
104   </div>
105 </div>
106     @stop
107
108 @section('footer')
109   @parent
110     @stop
```

Listing 111: ../portal/app/views/users/show.blade.php

```
1  @extends("layout/bootstrap")
2  @section("head")
3  @parent
4
5  @stop
6  @section('title')
7  @parent
8  - User
9  @stop
10 @section("content")
11 <div class="row">
12   <div class="col-md-offset-2 col-md-8">
13     <div class="well">
14       <div class="page-header">
15
16         <h3><span class="glyphicon glyphicon-user"></span>
17           {{$user->employee->last_name}}, {{$user->employee->first_name}}
18
19
20         <small>
21           {{$user->email}}
22           {{Form::open(array('route'=>array('users.edit',$user->id),'method'=>'GET','style'=>'display:inline'))}}
23           {{Form::editBtn('btn-xs','edit')}}
24           {{Form::close()}}
25           @if(Sentry::getUser()->inGroup(Sentry::findGroupByName('admin')))
26                       {{Form::open(array('route'=>array('users.disable',$user->id)))}}{{Form::deleteBtn('btn-xs
                              ','disable')}}{{Form::close()}}
27                       @endif
28         </small>
29         </h3>
30
31       </div>
32       <div class="row">
33         <div class="col-sm-6">
34           <p><b>Account created</b>: <em>{{$user->created_at}}</em></p>
35           <p><b>Updated at</b>: <em>{{$user->updated_at}}</em></p>
36         </div>
37         <div class="col-sm-6">
38           <p><b>Account status</b>: <em>
39
```

```
40              <?
41                $throttle = Sentry::findThrottlerByUserId($user->id);
42
43                  if($banned = $throttle->isBanned())
44                  {
45                      echo "Deactivated";
46                  }
47                else{
48                  if($user->activated){
49                    echo "Active";
50                  }
51                  else
52                  {
53                    echo "Not yet activated";
54                  }
55                }
56
57              ?>
58
59            </p>
60            <p><b>Account type</b>: @foreach($user->getGroups() as $group)
61              <?
62              switch($group["name"]){
63                case 'doctors': echo "Doctor"; break;
64                case "admin": echo "Admin"; break;
65                case "clerks": echo "Clerk"; break;
66                case "medical_technicians": echo "Medical Technician"; break;
67                case "hospital_admin": echo "Hospital Admin"; break;
68                case "patients": echo "Patient";
69                ?>
70                {{Form::open(array(
71                  'style'=>'display:inline',
72                  'method'=>'get',
73                  'route'=> array('patients.show',$user->patient->id)))}}
74                  {{Form::submit('view patient record',array('class'=>'btn btn-default'))}}
75                {{Form::close()}}
76                <?
77                break;
78
79              }
80              ?>
81              @endforeach
82            </p>
83          </div>
84        </div>
85      </div>
86    </div>
87  </div>
88
89    @stop
90        @section('footer')
91          @parent
92      @stop
```

Listing 112: ../portal/app/views/visits/create.blade.php

```
@extends("layout/bootstrap")
  @section("head")
    @parent

      @stop
        @section('title')
          @parent
            - Add Patient
@stop
  @section("content")

<div class="row">
  <div class="col-lg-offset-2 col-lg-8">
    <div class="well">
      {{ Form::open(array("route"=>array("visits.create.post"),"class"=>"form-horizontal","autocomplete"=>"off"))
            }}
      <fieldset>
      <legend>{{$patient->last_name ." ", $patient->first_name}}</legend>
      <?$edit = false;
      $newRecord = true;?>
      {{Form::hidden('employee_id',Sentry::getUser()->employee->employee_id)}}
      {{Form::hidden('id',$patient->id)}}
      @include("forms/visit")
    </fieldset>
      <div class="form-group">
        <div class="col-sm-offset-4 col-sm-8">
          {{ Form::submit("Submit",array("class"=>"btn btn-default")) }}
        </div>
      </div>

      {{Form::close() }}
    </div>
  </div>
</div>

  @stop
    @section('footer')
      @parent
  {{HTML::script('js/forms.js')}}
  @stop
```

Listing 113: ../portal/app/views/visits/diagnose.blade.php

```
@extends("layout/bootstrap")
  @section("head")
    @parent
@stop
  @section('title')
    @parent
- Patient Details
@stop
  @section("content")

<div class="row">
  {{Form::model($visit,array('role'=>'form'))}}
```

```
13    {{Form::hidden('visit_id',$visit->id)}}
14    <div class="col-md-offset-2 col-md-8">
15      <div class="panel-group">
16        <div class="panel panel-default">
17          <div class="panel-heading">
18            <h4>{{$visit->patient->last_name}}, {{$visit->patient->first_name}}  ({{{$visit->patient->hospital_id}}})
                    </h4>
19          </div>
20          <div class="panel-body">
21            <p><b>Date of Visit</b>:{{$visit->date_of_visit}}
22
23            </p>
24            <p><b>BP:</b> {{$visit->bp}}</p>
25            <p><b>Temperature:</b> {{$visit->temperature}}</p>
26            <p><b>C/R:</b> {{$visit->c_r}}</p>
27            <p><b>R/R:</b> {{$visit->r_r}}</p>
28            <p><b>Weight: </b> {{$visit->weight}}</p>
29            <p><b>Height: </b> {{$visit->height}}</p>
30
31
32
33
34          </div>
35        </div>
36
37
38        <div class="panel panel-default">
39          <div class="panel-heading"><h4>Information</h4></div>
40
41          <div class="panel-body">
42            <div class="form-group">
43              <label for="chief_complaint">Chief Complaint</label>
44              {{Form::textarea('chief_complaint',$visit->chief_complaint,array('rows'=>'4','class'=>'form-control')
                     )}}
45            </div>
46            <div class="form-group">
47              <label for="admitting_diagnosis">Admitting Diagnosis</label>
48              {{Form::textarea('admitting_diagnosis',$visit->admitting_diagnosis,array('rows'=>'4','class'=>'form-
                     control'))}}
49            </div>
50
51          </div>
52        </div>
53
54        <div class="panel panel-default">
55          <div class="panel-heading"><h4>Tests</h4></div>
56          <div class="panel-body">
57          @if($visit->status=='pending')
58            <div id="requests">
59              <div class="row">
60                <div class="form-group">
61                  <div class="col-lg-5">
62                    <select class="form-control departments" name="departments">
63                      <option value="0">Select One</option>
64
65                      @foreach($departments as $department)
```

```
66                    <option value='{{$department->id}}'>{{$department->department_name}}</option>
67                    @endforeach
68                  </select>
69                </div>
70                <div class="col-lg-5">
71                  {{Form::select('requests[]',array(),null,array('class'=>'form-control requests-dropdown','
                        disabled'))}}
72                </div>
73                <div class="col-lg-2">
74                  <a href="#" class="removeme">Remove</a>
75                </div>
76              </div>
77            </div>
78          </div>
79          <p><a href="#" id="add"><span>Add another</span></a></p>
80          @endif
81          <div class="row">
82            {{--a table of yr requests--}}
83            <table class="table table-hover table-bordered table-striped">
84              <thead>
85                <th>#</th>
86                <th>Test Type</th>
87                <th>Department</th>
88                <th>Date Requested</th>
89                <th>Result Available</th>
90              </thead>
91              <tbody>
92                <?$completed = true;?>
93
94                <?if(empty($visit->requests))
95                {
96                  $completed = false;
97                }
98                ?>
99
100               @foreach($visit->requests as $request)
101               <tr>
102                 <td>
103                   {{$request->id}}
104                 </td>
105                 <td>
106                   {{$request->type->test_name}}
107                 </td>
108                 <td>
109                   {{$request->type->department->department_name}}
110                 </td>
111                 <td>
112                   {{$request->created_at}}
113                 </td>
114                 <td>
115                   @if($request->status=='pending')
116                   Not yet available
117                   <? $completed = false;?>
118                   @elseif($request->status=='completed')
119                   {{link_to_route('results.show','Available',array('results'=>$request->result->id))}}
120                   @endif
```

```
                    </td>

                  </tr>
                  @endforeach
                </tbody>
              </table>
            </div>


          </div>

        </div>
        @if($visit->requests->filter(function($request){if($request->status!='completed') return $request;})->count()==0&&$visit->status!='pending')
        <div class="panel panel-default">
          <div class="panel-heading"><h4>Final Diagnosis</h4></div>

          <div class="panel-body">
            <div class="form-group">
              <label for="final_diagnosis">Final Diagnosis</label>
              {{Form::textarea('final_diagnosis',$visit->final_diagnosis,array('rows'=>'4','class'=>'form-control'))}}
            </div>
          </div>
        </div>
        @endif
        <div class="row">
          <div class="text-center">
            {{Form::submit('Submit',array('class'=>'btn btn-default'))}}
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
</div>


@stop
@section("footer")
  @parent
    {{HTML::script('js/requests.js')}}

    {{HTML::script('js/checkboxes.js')}}
@stop
```

Listing 114: ../portal/app/views/visits/edit.blade.php

```
@extends("layout/bootstrap")
  @section("head")
    @parent
  @stop
  @section('title')
    @parent
    - Visit
  @stop
```

```
9    @section("content")
10     <div class="row">
11       <div class="col-md-offset-2 col-md-8">
12         <div class="well">
13           {{ Form::model($visit,array('method'=>'PUT','class'=>'form-horizontal','route'=>array('visits.update',
                 $visit->id))) }}
14           <fieldset>
15             <legend>Edit Visit</legend>
16             <? $edit = true; ?>
17             <? $newRecord = false; ?>
18             {{Form::hidden('employee_id',Sentry::getUser()->employee->employee_id)}}
19             @include("forms/visit")
20           </fieldset>
21           <div class="form-group">
22             <div class="col-sm-offset-4 col-sm-8">
23               {{ Form::submit("Submit",array("class"=>"btn btn-default")) }}
24             </div>
25           </div>
26
27           {{Form::close() }}
28         </div>
29       </div>
30     </div>
31   @stop
32     @section('footer')
33       @parent
34   {{HTML::script('js/forms.js')}}
35   @stop
```

Listing 115: ../portal/app/views/visits/mine.blade.php

```
1  @extends("layout/bootstrap")
2  @section("head")
3  @parent
4
5  @stop
6  @section('title')
7  @parent
8  @stop
9  @section("content")
10   <table class="table table-hover table-bordered table-striped">
11     <thead>
12       <th>
13         Date of Visit
14       </th>
15       <th>
16         Attending Physician
17       </th>
18       <th>Details</th>
19       <th>
20         Test Results
21       </th>
22     </thead>
23     <tbody>
24       @foreach($visits as $visit)
```

```
25        <tr>
26          <td>{{$visit->date_of_visit}}</td>
27          <td>
28            @foreach ($visit->attendingPhysicians as $physician)
29            {{$physician->last_name}}, {{$physician->first_name}} <br />
30            @endforeach
31          </td>
32          <td>
33            <b>Chief Complaint</b>: {{$visit->chief_complaint}}
34            <br/>
35            <b>Admitting Diagnosis</b>: {{$visit->admitting_diagnosis}}
36            <br/>
37            <b>Final Diagnosis</b>: {{$visit->final_diagnosis}}
38          </td>
39          <td>
40            @foreach($visit->requests as $request)
41              {{$request->type->test_name}}
42
43              @if($request->status=='pending')
44            (Not yet available)
45          @elseif($request->status=='completed')
46            ({{link_to_route('results.show','Available',array('results'=>$request->result->id))}})
47            @endif
48            <br />
49            @endforeach
50          </td>
51        </tr>
52        @endforeach
53      </tbody>
54    </table>
55    <div class="text-center">
56
57    {{$visits->links()}}
58    </div>
59  @stop
60    @section('footer')
61      @parent
62  @stop
```

Listing 116: ../portal/app/views/visits/show.blade.php

```
1   @extends("layout/bootstrap")
2   @section("head")
3   @parent
4
5   @stop
6   @section('title')
7   @parent
8   - Visit
9   @stop
10  @section("content")
11  <div class="row">
12    <div class="col-md-offset-2 col-md-8">
13      <div class="panel panel-default">
14        <div class="panel-heading">
```

```
15          <div class="row">
16            <div class="col-md-8">
17              <h4><b>{{$visit->patient->last_name}}, {{$visit->patient->first_name}}   (#{{$visit->patient->id}})</
                    b></h4>
18            </div>
19            <div class="col-md-2">
20              @if(Sentry::getUser()->hasAccess('create_patient'))
21               {{Form::open(array('route'=>array('visits.edit',$visit->id),'method'=>'GET'))}}{{Form::editBtn('btn-
                     block','edit')}}{{Form::close()}}
22              @endif
23            </div>
24
25          </div>
26        </div>
27        <div class="panel-body">
28          <p><b>Date of Visit</b>:{{$visit->date_of_visit}}
29
30        </p>
31        @if(Sentry::getUser()->inGroup(Sentry::findGroupByName('clerks')))
32        <p><b>Visit Status</b>:
33         @if($visit->status=='waiting')
34         @if($visit->requests->filter(function($request){if($request->status!='completed') return $request;})->
               count()==0)
35        all results are in {{Form::open(array('url'=>'visits/status/'.$visit->id,'method'=>'post'))}}{{Form::submit
              ('change',array('style'=>'display:inline','class'=>'btn btn-default btn-xs'))}}{{Form::close()}}
36         @else
37         waiting for results
38
39         @endif
40         @else
41         {{$visit->status}}
42         @endif
43
44      </p>
45        @endif
46
47            <p><b>Attending Physicians:</b><br />
48            @foreach ($visit->attendingPhysicians as $physician)
49              {{$physician->last_name}}, {{$physician->first_name}} <br />
50            @endforeach
51            </p>
52
53            <p><b>BP:</b> {{$visit->bp}}</p>
54            <p><b>Temperature:</b> {{$visit->temperature}}</p>
55            <p><b>C/R:</b> {{$visit->c_r}}</p>
56            <p><b>R/R:</b> {{$visit->r_r}}</p>
57            <p><b>Weight: </b> {{$visit->weight}}</p>
58            <p><b>Height: </b> {{$visit->height}}</p>
59      @if(Sentry::getUser()->inGroup(Sentry::findGroupByName('hospital_admin'))||Sentry::getUser()->inGroup(Sentry
            ::findGroupByName('medical_technicians'))||Sentry::getUser()->inGroup(Sentry::findGroupByName('doctors')
            ))
60      <p><b>Chief Complaint:</b> {{$visit->chief_complaint}}</p>
61      <p><b>Admitting Diagnosis:</b> {{$visit->admitting_diagnosis}}</p>
62      <p><b>Final Diagnosis:</b> {{$visit->final_diagnosis}}</p>
63
64
```

```
65      @endif
66
67        </div>
68
69
70        </div>
71      </div>
72      @if(Sentry::getUser()->inGroup(Sentry::findGroupByName('hospital_admin'))||Sentry::getUser()->inGroup(Sentry
            ::findGroupByName('medical_technicians'))||Sentry::getUser()->inGroup(Sentry::findGroupByName('doctors')
            ))
73    <table class="table table-hover table-bordered table-striped">
74      <thead>
75        <th>#</th>
76        <th>Test Type</th>
77        <th>Department</th>
78        <th>Date Requested</th>
79        <th>Result Available</th>
80      </thead>
81      <tbody>
82        @foreach($visit->requests as $request)
83        <tr>
84          <td>
85            {{$request->id}}
86          </td>
87          <td>
88            {{$request->type->test_name}}
89          </td>
90          <td>
91            {{$request->type->department->department_name}}
92          </td>
93          <td>
94            {{$request->created_at}}
95          </td>
96          <td>
97            @if($request->status=='pending')
98            Not yet available
99          @elseif($request->status=='completed')
100           {{link_to_route('results.show','Available',array('results'=>$request->result->id))}}
101           @endif
102         </td>
103
104       </tr>
105       @endforeach
106     </tbody>
107   </table>
108   @endif
109   </div>
110
111
112 @stop
113   @section('footer')
114     @parent
115   @include('layout/confirm-delete')
116 @stop
```

Listing 117: ../portal/bootstrap/autoload.php

```
<?php

define('LARAVEL_START', microtime(true));

/*
|--------------------------------------------------------------------------
| Register The Composer Auto Loader
|--------------------------------------------------------------------------
|
| Composer provides a convenient, automatically generated class loader
| for our application. We just need to utilize it! We'll require it
| into the script here so that we do not have to worry about the
| loading of any our classes "manually". Feels great to relax.
|
*/

require __DIR__.'/../vendor/autoload.php';

/*
|--------------------------------------------------------------------------
| Include The Compiled Class File
|--------------------------------------------------------------------------
|
| To dramatically increase your application's performance, you may use a
| compiled class file which contains all of the classes commonly used
| by a request. The Artisan "optimize" is used to create this file.
|
*/

if (file_exists($compiled = __DIR__.'/compiled.php'))
{
  require $compiled;
}

/*
|--------------------------------------------------------------------------
| Setup Patchwork UTF-8 Handling
|--------------------------------------------------------------------------
|
| The Patchwork library provides solid handling of UTF-8 strings as well
| as provides replacements for all mb_* and iconv type functions that
| are not available by default in PHP. We'll setup this stuff here.
|
*/

Patchwork\Utf8\Bootup::initMbstring();

/*
|--------------------------------------------------------------------------
| Register The Laravel Auto Loader
|--------------------------------------------------------------------------
|
| We register an auto-loader "behind" the Composer loader that can load
| model classes on the fly, even if the autoload files have not been
```

```
| regenerated for the application. We'll add it to the stack here.
|
*/

Illuminate\Support\ClassLoader::register();

/*
|--------------------------------------------------------------------------
| Register The Workbench Loaders
|--------------------------------------------------------------------------
|
| The Laravel workbench provides a convenient place to develop packages
| when working locally. However we will need to load in the Composer
| auto-load files for the packages so that these can be used here.
|
*/

if (is_dir($workbench = __DIR__.'/../workbench'))
{
  Illuminate\Workbench\Starter::start($workbench);
}
```

Listing 118: ../portal/bootstrap/paths.php

```
<?php

return array(

  /*
  |--------------------------------------------------------------------------
  | Application Path
  |--------------------------------------------------------------------------
  |
  | Here we just defined the path to the application directory. Most likely
  | you will never need to change this value as the default setup should
  | work perfectly fine for the vast majority of all our applications.
  |
  */

  'app' => __DIR__.'/../app',

  /*
  |--------------------------------------------------------------------------
  | Public Path
  |--------------------------------------------------------------------------
  |
  | The public path contains the assets for your web application, such as
  | your JavaScript and CSS files, and also contains the primary entry
  | point for web requests into these applications from the outside.
  |
  */

  'public' => __DIR__.'/../public',

  /*
```

```
32    |--------------------------------------------------------------------------
33    | Base Path
34    |--------------------------------------------------------------------------
35    |
36    | The base path is the root of the Laravel installation. Most likely you
37    | will not need to change this value. But, if for some wild reason it
38    | is necessary you will do so here, just proceed with some caution.
39    |
40    */
41
42    'base' => __DIR__.'/..',
43
44    /*
45    |--------------------------------------------------------------------------
46    | Storage Path
47    |--------------------------------------------------------------------------
48    |
49    | The storage path is used by Laravel to store cached Blade views, logs
50    | and other pieces of information. You may modify the path here when
51    | you want to change the location of this directory for your apps.
52    |
53    */
54
55    'storage' => __DIR__.'/../app/storage',
56
57 );
```

Listing 119: ../portal/bootstrap/start.php

```
1  <?php
2
3  /*
4  |--------------------------------------------------------------------------
5  | Create The Application
6  |--------------------------------------------------------------------------
7  |
8  | The first thing we will do is create a new Laravel application instance
9  | which serves as the "glue" for all the components of Laravel, and is
10 | the IoC container for the system binding all of the various parts.
11 |
12 */
13
14 $app = new Illuminate\Foundation\Application;
15
16 $app->redirectIfTrailingSlash();
17
18 /*
19 |--------------------------------------------------------------------------
20 | Detect The Application Environment
21 |--------------------------------------------------------------------------
22 |
23 | Laravel takes a dead simple approach to your application environments
24 | so you can just specify a machine name or HTTP host that matches a
25 | given environment, then we will automatically detect it for you.
26 |
```

```
27  */
28
29  $env = $app->detectEnvironment(array(
30
31    'local' => array('your-machine-name'),
32
33  ));
34
35  /*
36  |--------------------------------------------------------------------------
37  | Bind Paths
38  |--------------------------------------------------------------------------
39  |
40  | Here we are binding the paths configured in paths.php to the app. You
41  | should not be changing these here. If you need to change these you
42  | may do so within the paths.php file and they will be bound here.
43  |
44  */
45
46  $app->bindInstallPaths(require __DIR__.'/paths.php');
47
48  /*
49  |--------------------------------------------------------------------------
50  | Load The Application
51  |--------------------------------------------------------------------------
52  |
53  | Here we will load the Illuminate application. We'll keep this is in a
54  | separate location so we can isolate the creation of an application
55  | from the actual running of the application with a given request.
56  |
57  */
58
59  $framework = $app['path.base'].'/vendor/laravel/framework/src';
60
61  require $framework.'/Illuminate/Foundation/start.php';
62
63  /*
64  |--------------------------------------------------------------------------
65  | Return The Application
66  |--------------------------------------------------------------------------
67  |
68  | This script returns the application instance. The instance is given to
69  | the calling script so we can separate the building of the instances
70  | from the actual running of the application and sending responses.
71  |
72  */
73
74  return $app;
```

Listing 120: ../portal/public/index.php

```
1  <?php
2  /**
3   * Laravel - A PHP Framework For Web Artisans
4   *
```

```
 * @package  Laravel
 * @author   Taylor Otwell <taylorotwell@gmail.com>
 */

/*
|--------------------------------------------------------------------------
| Register The Auto Loader
|--------------------------------------------------------------------------
|
| Composer provides a convenient, automatically generated class loader
| for our application. We just need to utilize it! We'll require it
| into the script here so that we do not have to worry about the
| loading of any our classes "manually". Feels great to relax.
|
*/

require __DIR__.'/../bootstrap/autoload.php';

/*
|--------------------------------------------------------------------------
| Turn On The Lights
|--------------------------------------------------------------------------
|
| We need to illuminate PHP development, so let's turn on the lights.
| This bootstraps the framework and gets it ready for use, then it
| will load up this application so that we can run it and send
| the responses back to the browser and delight these users.
|
*/

$app = require_once __DIR__.'/../bootstrap/start.php';

/*
|--------------------------------------------------------------------------
| Run The Application
|--------------------------------------------------------------------------
|
| Once we have the application, we can simply call the run method,
| which will execute the request and send the response back to
| the client's browser allowing them to enjoy the creative
| and wonderful applications we have created for them.
|
*/

$app->run();

/*
|--------------------------------------------------------------------------
| Shutdown The Application
|--------------------------------------------------------------------------
|
| Once the app has finished running, we will fire off the shutdown events
| so that any final work may be done by the application before we shut
| down the process. This is the last thing to happen to the request.
|
*/
```

```
61
62  $app->shutdown();
```

Listing 121: ../portal/public/robots.txt

```
1  User-agent: *
2  Disallow:
```

Listing 122: ../portal/public/css/custom.css

```
1  .required{
2    color:red;
3  }
```

Listing 123: ../portal/public/css/footer.css

```
1  html,body {
2    height: 100%;
3    /* The html and body elements cannot have any padding or margin. */
4  }
5
6  /* Wrapper for page content to push down footer */
7  #wrap {
8    min-height: 100%;
9    height: auto !important;
10   height: 100%;
11   /* Negative indent footer by it's height */
12   margin: 0 auto -60px;
13   padding-top: 70px;
14 }
15
16 /* Set the fixed height of the footer here */
17 #push,
18 #footer {
19   height: 60px;
20 }
21 #footer {
22   background-color: #000;
23   color: #fff;
24   padding-top:10px;
25 }
26
27 /* Lastly, apply responsive CSS fixes as necessary */
28 @media (max-width: 767px) {
29   #footer {
30     margin-left: -20px;
31     margin-right: -20px;
32     padding-left: 20px;
33     padding-right: 20px;
34   }
35 }
```

Listing 124: ../portal/public/css/select2–bootstrap.css

```
/**
 * Select2 Bootstrap CSS
 * Compatible with Select2 3.3.2, 3.4.1, 3.4.2 and Twitter Bootstrap 3.0.0
 * MIT License
 */
/**
 * Reset Bootstrap 3 .form-control styles which - if applied to the
 * original <select>-element the Select2-plugin may be run against -
 * are copied to the .select2-container.
 *
 * 1. Overwrite .select2-container's original display:inline-block
 *    with Bootstrap 3's default for .form-control, display:block;
 *    courtesy of @juristr (@see https://github.com/fk/select2-bootstrap-css/pull/1)
 */
.select2-container.form-control {
  background: transparent;
  border: none;
  display: block;
  /* 1 */
  margin: 0;
  padding: 0;
}

/**
 * Adjust Select2 inputs to fit Bootstrap 3 default .form-control appearance.
 */
.select2-container .select2-choices .select2-search-field input,
.select2-container .select2-choice,
.select2-container .select2-choices {
  background: none;
  padding: 0;
  border-color: #cccccc;
  border-radius: 4px;
  color: #555555;
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
  background-color: white;
  filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);
  -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075);
  box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075);
}

.select2-search input {
  border-color: #cccccc;
/*  border-radius: 4px;*/
  color: #555555;
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
  background-color: white;
  filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);
  -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075);
  box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075);
}

.select2-container .select2-choices .select2-search-field input {
  -webkit-box-shadow: none;
```

```
  box-shadow: none;
}

/**
 * Adjust Select2 input heights to match the Bootstrap default.
 */
.select2-container .select2-choice {
  height: 34px;
  line-height: 1.42857;
}

/**
 * Address Multi Select2's height which - depending on how many elements have been selected -
 * may grown higher than their initial size.
 */
.select2-container.select2-container-multi.form-control {
  height: auto;
}

/**
 * Address Bootstrap 3 control sizing classes
 * @see http://getbootstrap.com/css/#forms-control-sizes
 */
.select2-container.input-sm .select2-choice,
.input-group-sm .select2-container .select2-choice {
  height: 30px;
  line-height: 1.5;
/*  border-radius: 3px;*/
}

.select2-container.input-lg .select2-choice,
.input-group-lg .select2-container .select2-choice {
  height: 45px;
  line-height: 1.33;
  border-radius: 6px;
}

.select2-container-multi .select2-choices .select2-search-field input {
  height: 32px;
}

.select2-container-multi.input-sm .select2-choices .select2-search-field input,
.input-group-sm .select2-container-multi .select2-choices .select2-search-field input {
  height: 28px;
}

.select2-container-multi.input-lg .select2-choices .select2-search-field input,
.input-group-lg .select2-container-multi .select2-choices .select2-search-field input {
  height: 43px;
}

/**
 * Adjust height and line-height for .select2-search-field amd multi-select Select2 widgets.
 *
 * 1. Class repetition to address missing .select2-chosen in Select2 < 3.3.2.
 */
```

```
.select2-container-multi .select2-choices .select2-search-field input {
  margin: 0;
}

.select2-chosen,
.select2-choice > span:first-child,
.select2-container .select2-choices .select2-search-field input {
  padding: 6px 12px;
}

.input-sm .select2-chosen,
.input-group-sm .select2-chosen,
.input-sm .select2-choice > span:first-child,
.input-group-sm .select2-choice > span:first-child,
.input-sm .select2-choices .select2-search-field input,
.input-group-sm .select2-choices .select2-search-field input {
  padding: 5px 10px;
}

.input-lg .select2-chosen,
.input-group-lg .select2-chosen,
.input-lg .select2-choice > span:first-child,
.input-group-lg .select2-choice > span:first-child,
.input-lg .select2-choices .select2-search-field input,
.input-group-lg .select2-choices .select2-search-field input {
  padding: 10px 16px;
}

.select2-container-multi .select2-choices .select2-search-choice {
  margin-top: 5px;
  margin-bottom: 3px;
}

.select2-container-multi.input-sm .select2-choices .select2-search-choice,
.input-group-sm .select2-container-multi .select2-choices .select2-search-choice {
  margin-top: 3px;
  margin-bottom: 2px;
}

.select2-container-multi.input-lg .select2-choices .select2-search-choice,
.input-group-lg .select2-container-multi .select2-choices .select2-search-choice {
  line-height: 24px;
}

/**
 * Adjust the single Select2's dropdown arrow button appearance.
 *
 * 1. For Select2 v.3.3.2.
 */
.select2-container .select2-choice .select2-arrow,
.select2-container .select2-choice div {
  border-left: 1px solid #cccccc;
  background: none;
  filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);
}

```

```
.select2-dropdown-open .select2-choice .select2-arrow,
.select2-dropdown-open .select2-choice div {
  border-left-color: transparent;
  background: none;
  filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);
}

/**
 * Adjust the dropdown arrow button icon position for the single-select Select2 elements
 * to make it line up vertically now that we increased the height of .select2-container.
 *
 * 1. Class repetition to address missing .select2-chosen in Select2 v.3.3.2.
 */
.select2-container .select2-choice .select2-arrow b,
.select2-container .select2-choice div b {
  background-position: 0 3px;
}

.select2-dropdown-open .select2-choice .select2-arrow b,
.select2-dropdown-open .select2-choice div b {
  background-position: -18px 3px;
}

.select2-container.input-sm .select2-choice .select2-arrow b,
.input-group-sm .select2-container .select2-choice .select2-arrow b,
.select2-container.input-sm .select2-choice div b,
.input-group-sm .select2-container .select2-choice div b {
  background-position: 0 1px;
}

.select2-dropdown-open.input-sm .select2-choice .select2-arrow b,
.input-group-sm .select2-dropdown-open .select2-choice .select2-arrow b,
.select2-dropdown-open.input-sm .select2-choice div b,
.input-group-sm .select2-dropdown-open .select2-choice div b {
  background-position: -18px 1px;
}

.select2-container.input-lg .select2-choice .select2-arrow b,
.input-group-lg .select2-container .select2-choice .select2-arrow b,
.select2-container.input-lg .select2-choice div b,
.input-group-lg .select2-container .select2-choice div b {
  background-position: 0 9px;
}

.select2-dropdown-open.input-lg .select2-choice .select2-arrow b,
.input-group-lg .select2-dropdown-open .select2-choice .select2-arrow b,
.select2-dropdown-open.input-lg .select2-choice div b,
.input-group-lg .select2-dropdown-open .select2-choice div b {
  background-position: -18px 9px;
}

/**
 * Address Bootstrap's validation states and change Select2's border colors and focus states.
 * Apply .has-warning, .has-danger or .has-succes to #select2-drop to match Bootstraps' colors.
 */
.has-warning .select2-choice,
```

```
.has-warning .select2-choices {
  border-color: #c09853;
}
.has-warning .select2-container-active .select2-choice,
.has-warning .select2-container-multi.select2-container-active .select2-choices {
  border-color: #a47e3c;
  -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075), 0 0 6px #dbc59e;
  box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075), 0 0 6px #dbc59e;
}
.has-warning.select2-drop-active {
  border-color: #a47e3c;
}
.has-warning.select2-drop-active.select2-drop.select2-drop-above {
  border-top-color: #a47e3c;
}

.has-error .select2-choice,
.has-error .select2-choices {
  border-color: #b94a48;
}
.has-error .select2-container-active .select2-choice,
.has-error .select2-container-multi.select2-container-active .select2-choices {
  border-color: #953b39;
  -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075), 0 0 6px #d59392;
  box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075), 0 0 6px #d59392;
}
.has-error.select2-drop-active {
  border-color: #953b39;
}
.has-error.select2-drop-active.select2-drop.select2-drop-above {
  border-top-color: #953b39;
}

.has-success .select2-choice,
.has-success .select2-choices {
  border-color: #468847;
}
.has-success .select2-container-active .select2-choice,
.has-success .select2-container-multi.select2-container-active .select2-choices {
  border-color: #356635;
  -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075), 0 0 6px #7aba7b;
  box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075), 0 0 6px #7aba7b;
}
.has-success.select2-drop-active {
  border-color: #356635;
}
.has-success.select2-drop-active.select2-drop.select2-drop-above {
  border-top-color: #356635;
}

/**
 * Make Select2's active-styles - applied to .select2-container when the widget receives focus -
 * fit Bootstrap 3's .form-element:focus appearance.
 */
.select2-container-active .select2-choice,
.select2-container-multi.select2-container-active .select2-choices {
```

```
    border-color: #66afe9;
    outline: none;
    -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075), 0 0 8px rgba(82, 168, 236, 0.6);
    box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075), 0 0 8px rgba(82, 168, 236, 0.6);
    -webkit-transition: border-color ease-in-out 0.15s, box-shadow ease-in-out 0.15s;
    transition: border-color ease-in-out 0.15s, box-shadow ease-in-out 0.15s;
  }

  .select2-drop-active {
    border-color: #66afe9;
  }

  .select2-drop-auto-width,
  .select2-drop.select2-drop-above.select2-drop-active {
    border-top-color: #66afe9;
  }

  /**
   * Select2 widgets in Bootstrap Input Groups
   *
   * When Select2 widgets are combined with other elements using Bootstrap 3's
   * "Input Group" component, we don't want specific edges of the Select2 container
   * to have a border-radius.
   *
   * In Bootstrap 2, input groups required a markup where these style adjustments
   * could be bound to a CSS-class identifying if the additional elements are appended,
   * prepended or both.
   *
   * Bootstrap 3 doesn't rely on these classes anymore, so we have to use our own.
   * Use .select2-bootstrap-prepend and .select2-bootstrap-append on a Bootstrap 3 .input-group
   * to let the contained Select2 widget know which edges should not be rounded as they are
   * directly followed by another element.
   *
   * @see http://getbootstrap.com/components/#input-groups
   */
  .input-group.select2-bootstrap-prepend [class^="select2-choice"] {
    border-bottom-left-radius: 0 !important;
    border-top-left-radius: 0 !important;
  }

  .input-group.select2-bootstrap-append [class^="select2-choice"] {
    border-bottom-right-radius: 0 !important;
    border-top-right-radius: 0 !important;
  }

  .select2-dropdown-open [class^="select2-choice"] {
    border-bottom-right-radius: 0 !important;
    border-bottom-left-radius: 0 !important;
  }

  .select2-dropdown-open.select2-drop-above [class^="select2-choice"] {
    border-top-right-radius: 0 !important;
    border-top-left-radius: 0 !important;
  }

  /**
```

```
 * Adjust Select2's choices hover and selected styles to match Bootstrap 3's default dropdown styles.
 */
.select2-results .select2-highlighted {
  color: white;
  background-color: #428bca;
}

/**
 * Adjust alignment of Bootstrap 3 buttons in Bootstrap 3 Input Groups to address
 * Multi Select2's height which - depending on how many elements have been selected -
 * may grown higher than their initial size.
 */
.select2-bootstrap-append .select2-container-multiple,
.select2-bootstrap-append .input-group-btn,
.select2-bootstrap-append .input-group-btn .btn,
.select2-bootstrap-prepend .select2-container-multiple,
.select2-bootstrap-prepend .input-group-btn,
.select2-bootstrap-prepend .input-group-btn .btn {
  vertical-align: top;
}

/**
 * Make Multi Select2's choices match Bootstrap 3's default button styles.
 */
.select2-container-multi .select2-choices .select2-search-choice {
  color: #555555;
  background: white;
  border-color: #cccccc;
  filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);
  -webkit-box-shadow: none;
  box-shadow: none;
}

.select2-container-multi .select2-choices .select2-search-choice-focus {
  background: #ebebeb;
  border-color: #adadad;
  color: #333333;
  -webkit-box-shadow: none;
  box-shadow: none;
}

/**
 * Address Multi Select2's choice close-button vertical alignment.
 */
.select2-search-choice-close {
  margin-top: -7px;
  top: 50%;
}

/**
 * Adjust the single Select2's clear button position (used to reset the select box
 * back to the placeholder value and visible once a selection is made
 * activated by Select2's "allowClear" option).
 */
.select2-container .select2-choice abbr {
  top: 50%;
```

```
}

/**
 * Adjust "no results" and "selection limit" messages to make use
 * of Bootstrap 3's default "Alert" style.
 *
 * @see http://getbootstrap.com/components/#alerts-default
 */
.select2-results .select2-no-results,
.select2-results .select2-searching,
.select2-results .select2-selection-limit {
  background-color: #fcf8e3;
  color: #c09853;
}

/**
 * Address disabled Select2 styles.
 *
 * 1. For Select2 v.3.3.2.
 * 2. Revert border-left:0 inherited from Select2's CSS to prevent the arrow
 *    from jumping when switching from disabled to enabled state and vice versa.
 */
.select2-container.select2-container-disabled .select2-choice,
.select2-container.select2-container-disabled .select2-choices {
  cursor: not-allowed;
  background-color: #eeeeee;
  border-color: #cccccc;
}
.select2-container.select2-container-disabled .select2-choice .select2-arrow,
.select2-container.select2-container-disabled .select2-choice div,
.select2-container.select2-container-disabled .select2-choices .select2-arrow,
.select2-container.select2-container-disabled .select2-choices div {
  background-color: transparent;
  border-left: 1px solid transparent;
  /* 2 */
}

/**
 * Address Select2's loading indicator position - which should not stick
 * to the right edge of Select2's search input.
 *
 * 1. in .select2-search input
 * 2. in Multi Select2's .select2-search-field input
 * 3. in the status-message of infinite-scroll with remote data (@see http://ivaynberg.github.io/select2/#
      infinite)
 *
 * These styles alter Select2's default background-position of 100%
 * and supply the new background-position syntax to browsers which support it:
 *
 * 1. Android, Safari < 6/Mobile, IE<9: change to a relative background-position of 99%
 * 2. Chrome 25+, Firefox 13+, IE 9+, Opera 10.5+: use the new CSS3-background-position syntax
 *
 * @see http://www.w3.org/TR/css3-background/#background-position
 *
 * @todo Since both Select2 and Bootstrap 3 only support IE8 and above,
 * we could use the :after-pseudo-element to display the loading indicator.
```

```
 * Alternatively, we could supply an altered loading indicator image which already
 * contains an offset to the right.
 */
.select2-search input.select2-active,
.select2-container-multi .select2-choices .select2-search-field input.select2-active,
.select2-more-results.select2-active {
  background-position: 99%;
  /* 4 */
  background-position: right 4px center;
  /* 5 */
}
```

Listing 125: ../portal/public/js/confirm–delete.js

```
    $(document).ready(function() {

        var currentForm;
           $(function() {
               $("#dialog-confirm").dialog({
                   resizable: false,
                   modal: true,
                   autoOpen: false,
                   buttons: {
                       'Delete': function() {
                           $(this).dialog('close');
                           currentForm.submit();
                       },
                       'Cancel': function() {
                           $(this).dialog('close');
                       }
                   }
               });
               $(".btn-danger").click(function() {
                 currentForm = $(this).closest('form');
                 $("#dialog-confirm").dialog('open');
                 return false;
               });
           });
    });
```

Listing 126: ../portal/public/js/departments.js

```
$(document).ready(function(){
  var count = 1;


  $(function(){
    var $line = $("#assigned div:first").clone();


    $('#addnew').click(function(e){
      e.preventDefault();
      count++;
```

```
12        $('#assigned').append("<div class='newemp'><div class='col-sm-4 col-sm-offset-4'><input class='form-control
              ' name='employee_name[]' placeholder='Last Name, First Name'></input></div><div class='col-sm-4'><
              input name='employee_email[]' class='form-control' placeholder='email'></input></div></div>");
13      });
14      $('#addcurrent').click(function(e){
15        e.preventDefault();
16        count++;
17        $('#assigned').append($line.clone().addClass('col-sm-offset-4'));
18
19      });
20
21      $('#remove').click(function(e){
22        e.preventDefault();
23        if(count>1){
24        count--;
25        $("#assigned div.newemp:last").remove();
26
27        $("#assigned div.col-sm-8:last").remove();
28
29        }
30      });
31
32
33
34    });
35
36  });
```

Listing 127: ../portal/public/js/edit.js

```
1   $(document).ready(function() {
2     $(function(){
3       $(".btn-rename").click(function(e){
4         e.preventDefault();
5         $("#" + this.id).toggle();
6         $(this).hide();
7       });
8     });
9     $(function(){
10      $(".btn-cancel").click(function(e){
11        $(this).parent().hide();
12        $(this).parent().next().children().show();
13      });
14    })
15    });
```

Listing 128: ../portal/public/js/forms.js

```
1   $(document).ready(function() {
2     $(function(){ $("#attending_physicians").select2()  });
3     $(function(){ $("#assigned").select2()  });
4
5     // $(function(){   $('#date_of_visit').datetimepicker({
6   //    pickTime:false,
```

```
//  }); });

    $(function(){ $('#birthdate').datetimepicker({
                    pickTime: false,
          startDate:'1/1/1900'
                }); });

  });
```

Listing 129: ../portal/public/js/report.js

```
$(document).ready(function() {
  $(function(){
    $("#period").change(function(e){
      //var d                              = new Date();
      if($(this).val()=='year')
      {
        $('#yearspan').html('<select name  = "year" id="year"><option value="2014">2014</option></select>');
        $('#monthspan').text('');
        $('#dayspan').text('');
      }
      else if($(this).val()=='month')
      {
        $('#yearspan').html('<select name  = "year" id="year"><option value="2014">2014</option></select>');
        $('#monthspan').html('<select name = "month" id="month"> <option value="" disabled selected>select month
             </option><option value="01">January</option>    <option value="02">February</option>    <option
             value="03">March</option>    <option value="04">April</option>    <option value="05">May</option>
                <option value="06">June</option>    <option value="07">July</option>    <option value="08">August
             </option>    <option value="09">September</option>    <option value="10">October</option>    <option
              value="11">November</option>    <option value="12">December</option></select>');
        $('#dayspan').text('');
      }
      else if($(this).val()=='week'||$(this).val()=='day')
      {

        $('#yearspan').html('<select name  = "year" id="year"><option value="2014">2014</option></select>');
        $('#monthspan').html('<select name = "month" id="month"> <option value="" disabled selected>select month
             </option><option value="01">January</option>    <option value="02">February</option>    <option
             value="03">March</option>    <option value="04">April</option>    <option value="05">May</option>
                <option value="06">June</option>    <option value="07">July</option>    <option value="08">August
             </option>    <option value="09">September</option>    <option value="10">October</option>    <option
              value="11">November</option>    <option value="12">December</option></select>');
        $('#dayspan').html('<select name   = "day" id="day"> <option value="" disabled selected>select day</
             option><option value="01">01</option>    <option value="02">02</option>    <option value="03">03</
             option>    <option value="04">04</option>    <option value="05">05</option>    <option value
             ="06">06</option>    <option value="07">07</option>    <option value="08">08</option>    <option
             value="09">09</option>    <option value="10">10</option>    <option value="11">11</option>    <
             option value="12">12</option>    <option value="13">13</option>    <option value="14">14</option>
                <option value="15">15</option>    <option value="16">16</option>    <option value="17">17</option
             >    <option value="18">18</option>    <option value="19">19</option>    <option value="20">20</
             option>    <option value="21">21</option>    <option value="22">22</option>    <option value
             ="23">23</option>    <option value="24">24</option>    <option value="25">25</option>    <option
             value="26">26</option>    <option value="27">27</option>    <option value="28">28</option>    <
             option value="29">29</option>    <option value="30">30</option>    <option value="31">31</option></
             select>');
```

```
23        }
24
25      });
26
27      $('#month').live('change',function(e){
28        if($("#period").val()!='week'&&$("#period").val()!='day'){
29        if($(this).val()=='01'||$(this).val()=='03'||$(this).val()=='05'||$(this).val()=='07'||$(this).val()
              =='08'||$(this).val()=='10'||$(this).val()=='12'){
30          $('#dayspan').html('<select name   = "day" id="day"> <option value="" disabled selected>select day</
                option><option value="01">01</option>    <option value="02">02</option>    <option value="03">03</
                option>    <option value="04">04</option>    <option value="05">05</option>    <option value
                ="06">06</option>    <option value="07">07</option>    <option value="08">08</option>    <option
                value="09">09</option>    <option value="10">10</option>    <option value="11">11</option>    <
                option value="12">12</option>    <option value="13">13</option>    <option value="14">14</option>
                   <option value="15">15</option>    <option value="16">16</option>    <option value="17">17</option
                >    <option value="18">18</option>    <option value="19">19</option>    <option value="20">20</
                option>    <option value="21">21</option>    <option value="22">22</option>    <option value
                ="23">23</option>    <option value="24">24</option>    <option value="25">25</option>    <option
                value="26">26</option>    <option value="27">27</option>    <option value="28">28</option>    <
                option value="29">29</option>    <option value="30">30</option>    <option value="31">31</option></
                select>');
31        }
32        else if($(this).val()=='02')
33        {
34          if('#year'%4==0)
35          {
36            $('#dayspan').html('<select name   = "day" id="day"> <option value="" disabled selected>select day</
                  option><option value="01">01</option>    <option value="02">02</option>    <option value="03">03</
                  option>    <option value="04">04</option>    <option value="05">05</option>    <option value
                  ="06">06</option>    <option value="07">07</option>    <option value="08">08</option>    <option
                  value="09">09</option>    <option value="10">10</option>    <option value="11">11</option>    <
                  option value="12">12</option>    <option value="13">13</option>    <option value="14">14</option>
                     <option value="15">15</option>    <option value="16">16</option>    <option value="17">17</
                  option>    <option value="18">18</option>    <option value="19">19</option>    <option value
                  ="20">20</option>    <option value="21">21</option>    <option value="22">22</option>    <option
                  value="23">23</option>    <option value="24">24</option>    <option value="25">25</option>    <
                  option value="26">26</option>    <option value="27">27</option>    <option value="28">28</option>
                     <option value="29">29</option> </select>  ');
37          }
38          else{
39            $('#dayspan').html('<select name   = "day" id="day"> <option value="" disabled selected>select day</
                  option><option value="01">01</option>    <option value="02">02</option>    <option value="03">03</
                  option>    <option value="04">04</option>    <option value="05">05</option>    <option value
                  ="06">06</option>    <option value="07">07</option>    <option value="08">08</option>    <option
                  value="09">09</option>    <option value="10">10</option>    <option value="11">11</option>    <
                  option value="12">12</option>    <option value="13">13</option>    <option value="14">14</option>
                     <option value="15">15</option>    <option value="16">16</option>    <option value="17">17</
                  option>    <option value="18">18</option>    <option value="19">19</option>    <option value
                  ="20">20</option>    <option value="21">21</option>    <option value="22">22</option>    <option
                  value="23">23</option>    <option value="24">24</option>    <option value="25">25</option>    <
                  option value="26">26</option>    <option value="27">27</option>    <option value="28">28</option>
                    ');
40          }
41        }
42        else
43        {
```

```
44
45          $('#dayspan').html('<select name   = "day" id="day"> <option value="" disabled selected>select day</
                option><option value="01">01</option>    <option value="02">02</option>    <option value="03">03</
                option>    <option value="04">04</option>    <option value="05">05</option>    <option value
                ="06">06</option>    <option value="07">07</option>    <option value="08">08</option>    <option
                value="09">09</option>    <option value="10">10</option>    <option value="11">11</option>    <
                option value="12">12</option>    <option value="13">13</option>    <option value="14">14</option>
                   <option value="15">15</option>    <option value="16">16</option>    <option value="17">17</option
                >    <option value="18">18</option>    <option value="19">19</option>    <option value="20">20</
                option>    <option value="21">21</option>    <option value="22">22</option>    <option value
                ="23">23</option>    <option value="24">24</option>    <option value="25">25</option>    <option
                value="26">26</option>    <option value="27">27</option>    <option value="28">28</option>    <
                option value="29">29</option> </select>  ');
46        }
47      }
48      });
49    });
50
51
52  });
```

Listing 130: ../portal/public/js/requests.js

```
1   $(document).ready(function(){
2     var count = 1;
3
4
5     $(function(){
6       var $line = $("#requests div:first").clone();
7
8
9       $('#add').click(function(e){
10        e.preventDefault();
11        count++;
12        $("#requests").append($line.clone());
13      });
14
15      $('.removeme').live('click',function(e){
16        e.preventDefault();
17        count--;
18        $('#count').val(count);
19        $(this).parents('.form-group').remove();
20      });
21
22      $('.departments').live('change',function(e){
23        var req_dropdown = $(this).parent().parent().find('.requests-dropdown');
24
25
26        var dept_id = $(this).val();
27        if(dept_id!=0){
28          req_dropdown.removeAttr('disabled');
29          req_dropdown.empty();
30          $(req_dropdown).load('../types/dropdown/'+dept_id);
31        }
32        else
```

```
33        {
34          req_dropdown.attr('disabled','disabled');
35          req_dropdown.empty();
36        }
37
38
39      });
40      // $('.departments').change(function(e){
41  //      var req_dropdown = $(this).parent().parent().find('.requests-dropdown');
42  //
43  //
44  //        req_dropdown.removeAttr('disabled');
45  //        var dept_id = $(this).val();
46  //        req_dropdown.empty();
47  //        $(req_dropdown).load('../types/dropdown/'+dept_id);
48  //
49  //
50  //    });
51    });
52
53  });
```

Listing 131: ../portal/public/js/types.js

```
1   function snake(e){
2     return e.toLowerCase().replace(/[^a-z0-9\s]/gi, '').replace(/[_\s]/g, '_');
3   }
4
5   $(document).ready(function() {
6
7     var count = 0;
8
9
10
11
12    $(function(){
13
14      $('#add').click(function(e){
15        e.preventDefault();
16        count++;
17        $('#fields').append('<fieldset><legend>Field '+count+'</legend><div class="form-group"><label>Field Name</
              label><input required type="text" name="field_name'+count+'" class="form-control" id="field_'+count
              +'"></div><div class="form-group"><label>Data Type</label><select name="data_type'+count+'" required
              class="form-control data-type"><option value="" disabled selected>Select One</option><option value="
              image">Image</option><option value="range">Numeric Value with Range</option><option value="selection">
              Text Selection</option></select></div><div class="details"></div></fieldset>');
18        $('#count').val(count);
19
20        $('.data-type').change(function(e){
21          if($(this).val()=='image')
22          {
23            $(this).parents('fieldset').find('.details').html('');
24
25          }
26          else if($(this).val()=='range')
```

```
27          {
28            // $(this).parents('fieldset').find('.comparison-value').attr('readonly',false);
29            $(this).parents('fieldset').find('.details').html('<label>Normal Range</label><div class="form-group"><
                  div class="col-sm-4"><input type="number" value="" class="form-control number" placeholder="min"
                  name="min'+count+'"></input></div><div class="col-sm-1">-</div><div class="col-sm-4"><input type="
                  number" placeholder="max" value="" class="form-control number" name="max'+count+'"></input></div><
                  div class="col-sm-3"><input type="text" placeholder="unit" value="" class="form-control" name="
                  unit'+count+'"></input></div></div>');
30
31            $('.number').rules("add",{number:true});
32
33
34          }
35          else if($(this).val()=='selection')
36          {
37            //$(this).parents('fieldset').find('.details').html('');
38            $(this).parents('fieldset').find('.details').html('<label>Values</label><div id="values"><input type="
                  text" value="" class="selection form-control" name="selection'+count+'" id="selectionbox'+count+'"
                   placeholder="Add possible values (separate with tab, comma, or enter)" /></div>');
39            $("#selectionbox"+count).tagsManager();
40          }
41        });
42
43
44
45
46
47
48
49      });
50
51      $('#remove').click(function(e){
52        e.preventDefault();
53        count--;
54        $('#count').val(count);
55        $('#fields fieldset:last-child').remove();
56      });
57
58
59
60
61
62
63      $("form").validate({
64        highlight: function(element) {
65          $(element).closest('.form-group').removeClass('has-success').addClass('has-error');
66
67        },
68        unhighlight: function(element){
69          $(element).closest('.form-group').removeClass('has-error').addClass('has-success');
70          $(element).parent().remove('span.help-inline');
71        },
72        errorPlacement: function(error, element) {
73          $(element).parent().append("<span class='help-inline has-error'>"+error.text()+"</span>");
74         },
75      });
```

```
76
77      $('.number').rules("add",{number:true});
78
79      $('.data-type').live('change',function(e){
80        if($(this).val()=='int'||$(this).val()=='float'){
81          $(this).parents('fieldset').append('<div class="form-group"><label>Unit</label><input required type="text
                " name="unit'+count+'" class="form-control unit" id="field_'+count+'"></div>');
82        }
83        else{
84          var $last = $(this).parents('fieldset').last();
85
86          if($last.has('.unit')){
87            $last.find('.unit').parent().remove();
88          }
89        }
90
91      });
92
93    });
94
95
96  });
```

Listing 132: ../portal/public/less/bootswatch.less

```
1   // Cosmo 3.0.3
2   // Bootswatch
3   // ------------------------------------------------------
4
5   @import url("//fonts.googleapis.com/css?family=Open+Sans:400italic,700italic,400,700");
6
7   // Navbar ================================================================
8
9   // Buttons ===============================================================
10
11  .btn {
12    border: none;
13  }
14
15  // Typography ============================================================
16
17  .text-primary,
18  .text-primary:hover {
19    color: @brand-primary;
20  }
21
22  .text-success,
23  .text-success:hover {
24    color: @brand-success;
25  }
26
27  .text-danger,
28  .text-danger:hover {
29    color: @brand-danger;
30  }
```

```
31
32  .text-warning,
33  .text-warning:hover {
34    color: @brand-warning;
35  }
36
37  .text-info,
38  .text-info:hover {
39    color: @brand-info;
40  }
41
42  // Tables ==================================================================
43
44  .table {
45
46    tr.success,
47    tr.warning,
48    tr.danger {
49      color: #fff;
50    }
51  }
52
53  // Forms ===================================================================
54
55
56  .has-warning {
57    .help-block,
58    .control-label {
59      color: @brand-warning;
60    }
61
62    .form-control,
63    .form-control:focus {
64      border: 1px solid @brand-warning;
65    }
66  }
67
68  .has-error {
69    .help-block,
70    .control-label {
71      color: @brand-danger;
72    }
73
74    .form-control,
75    .form-control:focus {
76      border: 1px solid @brand-danger;
77    }
78  }
79
80  .has-success {
81    .help-block,
82    .control-label {
83      color: @brand-success;
84    }
85
86    .form-control,
```

```
87    .form-control:focus {
88      border: 1px solid @brand-success;
89    }
90  }
91
92  // Navs ================================================================
93
94  .nav-pills {
95
96    & > li > a {
97      border-radius: 0;
98    }
99  }
100
101 .dropdown-menu {
102
103   & > li > a:hover,
104   & > li > a:focus {
105     background-image: none;
106   }
107 }
108
109 .pagination {
110
111   .active > a,
112   .active > a:hover {
113     border-color: #ddd;
114   }
115 }
116
117 // Indicators ==========================================================
118
119 .alert {
120   border: none;
121
122   .alert-link {
123     text-decoration: underline;
124     color: #fff;
125   }
126 }
127
128 .label {
129   border-radius: 0;
130 }
131
132 .close {
133   opacity: 1;
134 }
135
136 // Progress bars =======================================================
137
138 .progress {
139   height: 8px;
140   .box-shadow(none);
141 }
142
```

```
143 // Containers ======================================================================
```

Listing 133: ../portal/public/less/main.less

```
1  //All the CSS files in main.css
2
3  @import "../components/bootstrap/less/bootstrap.less";
4  @import "../components/select2/select2.css";
5  @import "../css/select2-bootstrap.css";
6  @import "bootswatch.less";
7  @import "variables.less";
8  @import "../components/eonasdan-bootstrap-datetimepicker/src/less/bootstrap-datetimepicker.less";
9  @import "../components/fancybox/source/jquery.fancybox.css";
10 @import "../components/jquery-ui/themes/smoothness/jquery-ui.css";
11 @import "../components/tagmanager/tagmanager.css";
12 @import "../css/footer.css";
13 @import "../css/custom.css";
```

Listing 134: ../portal/public/less/variables.less

```
1  // Cosmo 3.0.3
2  // Variables
3  // --------------------------------------------------
4
5
6  // Global values
7  // --------------------------------------------------
8
9  // Grays
10 // -------------------------
11
12 @gray-darker:            lighten(#000, 13.5%); // #222
13 @gray-dark:              lighten(#000, 20%);   // #333
14 @gray:                   lighten(#000, 33.5%); // #555
15 @gray-light:             lighten(#000, 60%);   // #999
16 @gray-lighter:           lighten(#000, 90%);
17
18 // Brand colors
19 // -------------------------
20
21 @brand-primary:         #007FFF;
22 @brand-success:         #3FB618;
23 @brand-warning:         #FF7518;
24 @brand-danger:          #FF0039;
25 @brand-info:            #9954BB;
26
27 // Scaffolding
28 // -------------------------
29
30 @body-bg:               #fff;
31 @text-color:            @gray-dark;
32
33 // Links
34 // -------------------------
```

```
35
36  @link-color:            @brand-primary;
37  @link-hover-color:      darken(@link-color, 15%);
38
39  // Typography
40  // -------------------------
41
42  @font-family-sans-serif:  "Open Sans", Calibri, Candara, Arial, sans-serif;
43  @font-family-serif:       Georgia, "Times New Roman", Times, serif;
44  @font-family-monospace:   Menlo, Monaco, Consolas, "Courier New", monospace;
45  @font-family-base:        @font-family-sans-serif;
46
47  @font-size-base:          15px;
48  @font-size-large:         ceil(@font-size-base * 1.25); // ~18px
49  @font-size-small:         ceil(@font-size-base * 0.85); // ~12px
50
51  @font-size-h1:            floor(@font-size-base * 2.6); // ~36px
52  @font-size-h2:            floor(@font-size-base * 2.15); // ~30px
53  @font-size-h3:            ceil(@font-size-base * 1.7); // ~24px
54  @font-size-h4:            ceil(@font-size-base * 1.25); // ~18px
55  @font-size-h5:            @font-size-base;
56  @font-size-h6:            ceil(@font-size-base * 0.85); // ~12px
57
58  @line-height-base:        1.428571429; // 20/14
59  @line-height-computed:    floor(@font-size-base * @line-height-base); // ~20px
60
61  @headings-font-family:    @font-family-base;
62  @headings-font-weight:    300;
63  @headings-line-height:    1.1;
64  @headings-color:          inherit;
65
66
67  // Iconography
68  // -------------------------
69
70  @icon-font-path:          "../fonts/";
71  @icon-font-name:          "glyphicons-halflings-regular";
72
73
74  // Components
75  // -------------------------
76  // Based on 14px font-size and 1.428 line-height (~20px to start)
77
78  @padding-base-vertical:          10px;
79  @padding-base-horizontal:        18px;
80
81  @padding-large-vertical:         18px;
82  @padding-large-horizontal:       30px;
83
84  @padding-small-vertical:         5px;
85  @padding-small-horizontal:       10px;
86
87  @padding-xs-vertical:            1px;
88  @padding-xs-horizontal:          5px;
89
90  @line-height-large:              1.33;
```

```
@line-height-small:              1.5;

@border-radius-base:             0px;
@border-radius-large:            0px;
@border-radius-small:            0px;

@component-active-color:         #fff;
@component-active-bg:            @brand-primary;

@caret-width-base:               4px;
@caret-width-large:              5px;

// Tables
// ------------------------

@table-cell-padding:                 8px;
@table-condensed-cell-padding:       5px;

@table-bg:                           transparent; // overall background-color
@table-bg-accent:                    #f9f9f9; // for striping
@table-bg-hover:                     #f5f5f5;
@table-bg-active:                    @table-bg-hover;

@table-border-color:                 #ddd; // table and cell border


// Buttons
// ------------------------

@btn-font-weight:                normal;

@btn-default-color:              #fff;
@btn-default-bg:                 @gray-darker;
@btn-default-border:             @btn-default-bg;

@btn-primary-color:              @btn-default-color;
@btn-primary-bg:                 @brand-primary;
@btn-primary-border:             @btn-primary-bg;

@btn-success-color:              @btn-default-color;
@btn-success-bg:                 @brand-success;
@btn-success-border:             @btn-success-bg;

@btn-warning-color:              @btn-default-color;
@btn-warning-bg:                 @brand-warning;
@btn-warning-border:             @btn-warning-bg;

@btn-danger-color:               @btn-default-color;
@btn-danger-bg:                  @brand-danger;
@btn-danger-border:              @btn-danger-bg;

@btn-info-color:                 @btn-default-color;
@btn-info-bg:                    @brand-info;
@btn-info-border:                @btn-info-bg;

@btn-link-disabled-color:        @gray-light;
```

```
147
148
149 // Forms
150 // -------------------------
151
152 @input-bg:                       #fff;
153 @input-bg-disabled:              @gray-lighter;
154
155 @input-color:                    @text-color;
156 @input-border:                   #ccc;
157 @input-border-radius:            @border-radius-base;
158 @input-border-focus:             #66afe9;
159
160 @input-color-placeholder:        @gray-light;
161
162 @input-height-base:              (@line-height-computed + (@padding-base-vertical * 2) + 2);
163 @input-height-large:             (ceil(@font-size-large * @line-height-large) + (@padding-large-vertical * 2) +
        2);
164 @input-height-small:             (floor(@font-size-small * @line-height-small) + (@padding-small-vertical * 2) +
        2);
165
166 @legend-color:                   @text-color;
167 @legend-border-color:            #e5e5e5;
168
169 @input-group-addon-bg:           @gray-lighter;
170 @input-group-addon-border-color: @input-border;
171
172
173 // Dropdowns
174 // -------------------------
175
176 @dropdown-bg:                    #fff;
177 @dropdown-border:                rgba(0,0,0,.15);
178 @dropdown-fallback-border:       #ccc;
179 @dropdown-divider-bg:            #e5e5e5;
180
181 @dropdown-link-color:            @gray-dark;
182 @dropdown-link-hover-color:      #fff;
183 @dropdown-link-hover-bg:         @dropdown-link-active-bg;
184
185 @dropdown-link-active-color:     #fff;
186 @dropdown-link-active-bg:        @component-active-bg;
187
188 @dropdown-link-disabled-color:   @text-muted;
189
190 @dropdown-header-color:          @text-muted;
191
192
193 // COMPONENT VARIABLES
194 // --------------------------------------------------
195
196
197 // Z-index master list
198 // -------------------------
199 // Used for a bird's eye view of components dependent on the z-axis
200 // Try to avoid customizing these :)
```

```
201
202 @zindex-navbar:            1000;
203 @zindex-dropdown:          1000;
204 @zindex-popover:           1010;
205 @zindex-tooltip:           1030;
206 @zindex-navbar-fixed:      1030;
207 @zindex-modal-background:  1040;
208 @zindex-modal:             1050;
209
210 // Media queries breakpoints
211 // --------------------------------------------------
212
213 // Extra small screen / phone
214 // Note: Deprecated @screen-xs and @screen-phone as of v3.0.1
215 @screen-xs:                  480px;
216 @screen-xs-min:              @screen-xs;
217 @screen-phone:               @screen-xs-min;
218
219 // Small screen / tablet
220 // Note: Deprecated @screen-sm and @screen-tablet as of v3.0.1
221 @screen-sm:                  768px;
222 @screen-sm-min:              @screen-sm;
223 @screen-tablet:              @screen-sm-min;
224
225 // Medium screen / desktop
226 // Note: Deprecated @screen-md and @screen-desktop as of v3.0.1
227 @screen-md:                  992px;
228 @screen-md-min:              @screen-md;
229 @screen-desktop:             @screen-md-min;
230
231 // Large screen / wide desktop
232 // Note: Deprecated @screen-lg and @screen-lg-desktop as of v3.0.1
233 @screen-lg:                  1200px;
234 @screen-lg-min:              @screen-lg;
235 @screen-lg-desktop:          @screen-lg-min;
236
237 // So media queries don't overlap when required, provide a maximum
238 @screen-xs-max:              (@screen-sm-min - 1);
239 @screen-sm-max:              (@screen-md-min - 1);
240 @screen-md-max:              (@screen-lg-min - 1);
241
242
243 // Grid system
244 // --------------------------------------------------
245
246 // Number of columns in the grid system
247 @grid-columns:              12;
248 // Padding, to be divided by two and applied to the left and right of all columns
249 @grid-gutter-width:         30px;
250
251 // Navbar collapse
252
253 // Point at which the navbar becomes uncollapsed
254 @grid-float-breakpoint:     @screen-sm-min;
255 // Point at which the navbar begins collapsing
256 @grid-float-breakpoint-max: (@grid-float-breakpoint - 1);
```

```
257
258
259  // Navbar
260  // -------------------------
261
262  // Basics of a navbar
263  @navbar-height:                    50px;
264  @navbar-margin-bottom:             @line-height-computed;
265  @navbar-border-radius:             @border-radius-base;
266  @navbar-padding-horizontal:        floor(@grid-gutter-width / 2);
267  @navbar-padding-vertical:          ((@navbar-height - @line-height-computed) / 2);
268
269  @navbar-default-color:             #fff;
270  @navbar-default-bg:                @gray-darker;
271  @navbar-default-border:            darken(@navbar-default-bg, 6.5%);
272
273  // Navbar links
274  @navbar-default-link-color:                #fff;
275  @navbar-default-link-hover-color:          #fff;
276  @navbar-default-link-hover-bg:             darken(@navbar-default-bg, 10%);
277  @navbar-default-link-active-color:         @navbar-default-link-hover-color;
278  @navbar-default-link-active-bg:            @navbar-default-link-hover-bg;
279  @navbar-default-link-disabled-color:       #ccc;
280  @navbar-default-link-disabled-bg:          transparent;
281
282  // Navbar brand label
283  @navbar-default-brand-color:               @navbar-default-link-color;
284  @navbar-default-brand-hover-color:         #fff;
285  @navbar-default-brand-hover-bg:            none;
286
287  // Navbar toggle
288  @navbar-default-toggle-hover-bg:           @navbar-default-link-hover-bg;
289  @navbar-default-toggle-icon-bar-bg:        #fff;
290  @navbar-default-toggle-border-color:       transparent;
291
292
293  // Inverted navbar
294  //
295  // Reset inverted navbar basics
296  @navbar-inverse-color:                      #fff;
297  @navbar-inverse-bg:                         @brand-primary;
298  @navbar-inverse-border:                     darken(@navbar-inverse-bg, 10%);
299
300  // Inverted navbar links
301  @navbar-inverse-link-color:                 #fff;
302  @navbar-inverse-link-hover-color:           #fff;
303  @navbar-inverse-link-hover-bg:              darken(@navbar-inverse-bg, 10%);
304  @navbar-inverse-link-active-color:          @navbar-inverse-link-hover-color;
305  @navbar-inverse-link-active-bg:             @navbar-inverse-link-hover-bg;
306  @navbar-inverse-link-disabled-color:        #fff;
307  @navbar-inverse-link-disabled-bg:           transparent;
308
309  // Inverted navbar brand label
310  @navbar-inverse-brand-color:                @navbar-inverse-link-color;
311  @navbar-inverse-brand-hover-color:          #fff;
312  @navbar-inverse-brand-hover-bg:             none;
```

```
313
314 // Inverted navbar toggle
315 @navbar-inverse-toggle-hover-bg:            @navbar-inverse-link-hover-bg;
316 @navbar-inverse-toggle-icon-bar-bg:         #fff;
317 @navbar-inverse-toggle-border-color:        transparent;
318
319
320 // Navs
321 // --------------------------
322
323 @nav-link-padding:                          10px 15px;
324 @nav-link-hover-bg:                         @gray-lighter;
325
326 @nav-disabled-link-color:                   @gray-light;
327 @nav-disabled-link-hover-color:             @gray-light;
328
329 @nav-open-link-hover-color:                 #fff;
330
331 // Tabs
332 @nav-tabs-border-color:                     #ddd;
333
334 @nav-tabs-link-hover-border-color:          @gray-lighter;
335
336 @nav-tabs-active-link-hover-bg:             @body-bg;
337 @nav-tabs-active-link-hover-color:          @gray;
338 @nav-tabs-active-link-hover-border-color:   #ddd;
339
340 @nav-tabs-justified-link-border-color:            #ddd;
341 @nav-tabs-justified-active-link-border-color:     @body-bg;
342
343 // Pills
344 @nav-pills-border-radius:                   @border-radius-base;
345 @nav-pills-active-link-hover-bg:            @component-active-bg;
346 @nav-pills-active-link-hover-color:         @component-active-color;
347
348
349 // Pagination
350 // --------------------------
351
352 @pagination-bg:                         #fff;
353 @pagination-border:                     #ddd;
354
355 @pagination-hover-bg:                   @gray-lighter;
356
357 @pagination-active-bg:                  #f5f5f5;
358 @pagination-active-color:               @gray-light;
359
360 @pagination-disabled-color:             @gray-light;
361
362
363 // Pager
364 // --------------------------
365
366 @pager-border-radius:                   @border-radius-base;
367 @pager-disabled-color:                  @gray-light;
368
```

```
369
370 // Jumbotron
371 // -------------------------
372
373 @jumbotron-padding:              30px;
374 @jumbotron-color:                inherit;
375 @jumbotron-bg:                   @gray-lighter;
376 @jumbotron-heading-color:        inherit;
377 @jumbotron-font-size:            ceil(@font-size-base * 1.5);
378
379
380 // Form states and alerts
381 // -------------------------
382
383 @state-success-text:             #fff;
384 @state-success-bg:               @brand-success;
385 @state-success-border:           darken(spin(@state-success-bg, -10), 5%);
386
387 @state-info-text:                #fff;
388 @state-info-bg:                  @brand-info;
389 @state-info-border:              darken(spin(@state-info-bg, -10), 7%);
390
391 @state-warning-text:             #fff;
392 @state-warning-bg:               @brand-warning;
393 @state-warning-border:           darken(spin(@state-warning-bg, -10), 3%);
394
395 @state-danger-text:              #fff;
396 @state-danger-bg:                @brand-danger;
397 @state-danger-border:            darken(spin(@state-danger-bg, -10), 3%);
398
399
400 // Tooltips
401 // -------------------------
402 @tooltip-max-width:           200px;
403 @tooltip-color:               #fff;
404 @tooltip-bg:                  rgba(0,0,0,.9);
405
406 @tooltip-arrow-width:         5px;
407 @tooltip-arrow-color:         @tooltip-bg;
408
409
410 // Popovers
411 // -------------------------
412 @popover-bg:                          #fff;
413 @popover-max-width:                   276px;
414 @popover-border-color:                rgba(0,0,0,.2);
415 @popover-fallback-border-color:       #ccc;
416
417 @popover-title-bg:                    darken(@popover-bg, 3%);
418
419 @popover-arrow-width:                 10px;
420 @popover-arrow-color:                 #fff;
421
422 @popover-arrow-outer-width:           (@popover-arrow-width + 1);
423 @popover-arrow-outer-color:           rgba(0,0,0,.25);
424 @popover-arrow-outer-fallback-color:  #999;
```

```
425
426
427  // Labels
428  // -------------------------
429
430  @label-default-bg:            @btn-default-bg;
431  @label-primary-bg:            @brand-primary;
432  @label-success-bg:            @brand-success;
433  @label-info-bg:               @brand-info;
434  @label-warning-bg:            @brand-warning;
435  @label-danger-bg:             @brand-danger;
436
437  @label-color:                 #fff;
438  @label-link-hover-color:      #fff;
439
440
441  // Modals
442  // -------------------------
443  @modal-inner-padding:         20px;
444
445  @modal-title-padding:         15px;
446  @modal-title-line-height:     @line-height-base;
447
448  @modal-content-bg:                             #fff;
449  @modal-content-border-color:                   rgba(0,0,0,.2);
450  @modal-content-fallback-border-color:          #999;
451
452  @modal-backdrop-bg:           #000;
453  @modal-header-border-color:   #e5e5e5;
454  @modal-footer-border-color:   @modal-header-border-color;
455
456
457  // Alerts
458  // -------------------------
459  @alert-padding:               15px;
460  @alert-border-radius:         @border-radius-base;
461  @alert-link-font-weight:      bold;
462
463  @alert-success-bg:            @state-success-bg;
464  @alert-success-text:          @state-success-text;
465  @alert-success-border:        @state-success-border;
466
467  @alert-info-bg:               @state-info-bg;
468  @alert-info-text:             @state-info-text;
469  @alert-info-border:           @state-info-border;
470
471  @alert-warning-bg:            @state-warning-bg;
472  @alert-warning-text:          @state-warning-text;
473  @alert-warning-border:        @state-warning-border;
474
475  @alert-danger-bg:             @state-danger-bg;
476  @alert-danger-text:           @state-danger-text;
477  @alert-danger-border:         @state-danger-border;
478
479
480  // Progress bars
```

```
// -------------------------
@progress-bg:                 #ccc;
@progress-bar-color:          #fff;

@progress-bar-bg:             @brand-primary;
@progress-bar-success-bg:     @brand-success;
@progress-bar-warning-bg:     @brand-warning;
@progress-bar-danger-bg:      @brand-danger;
@progress-bar-info-bg:        @brand-info;


// List group
// -------------------------
@list-group-bg:               #fff;
@list-group-border:           #ddd;
@list-group-border-radius:    @border-radius-base;

@list-group-hover-bg:         #f5f5f5;
@list-group-active-color:     @component-active-color;
@list-group-active-bg:        @component-active-bg;
@list-group-active-border:    @list-group-active-bg;

@list-group-link-color:          #555;
@list-group-link-heading-color:  #333;


// Panels
// -------------------------
@panel-bg:                    #fff;
@panel-inner-border:          #ddd;
@panel-border-radius:         @border-radius-base;
@panel-footer-bg:             #f5f5f5;

@panel-default-text:          @gray-dark;
@panel-default-border:        #ddd;
@panel-default-heading-bg:    #f5f5f5;

@panel-primary-text:          #fff;
@panel-primary-border:        @brand-primary;
@panel-primary-heading-bg:    @brand-primary;

@panel-success-text:          @state-success-text;
@panel-success-border:        @state-success-border;
@panel-success-heading-bg:    @state-success-bg;

@panel-warning-text:          @state-warning-text;
@panel-warning-border:        @state-warning-border;
@panel-warning-heading-bg:    @state-warning-bg;

@panel-danger-text:           @state-danger-text;
@panel-danger-border:         @state-danger-border;
@panel-danger-heading-bg:     @state-danger-bg;

@panel-info-text:             @state-info-text;
@panel-info-border:           @state-info-border;
@panel-info-heading-bg:       @state-info-bg;
```

```
537
538
539 // Thumbnails
540 // -------------------------
541 @thumbnail-padding:           4px;
542 @thumbnail-bg:                @body-bg;
543 @thumbnail-border:            #ddd;
544 @thumbnail-border-radius:     @border-radius-base;
545
546 @thumbnail-caption-color:     @text-color;
547 @thumbnail-caption-padding:   9px;
548
549
550 // Wells
551 // -------------------------
552 @well-bg:                     #f5f5f5;
553
554
555 // Badges
556 // -------------------------
557 @badge-color:                 #fff;
558 @badge-link-hover-color:      #fff;
559 @badge-bg:                    @gray-light;
560
561 @badge-active-color:          @link-color;
562 @badge-active-bg:             #fff;
563
564 @badge-font-weight:           bold;
565 @badge-line-height:           1;
566 @badge-border-radius:         10px;
567
568
569 // Breadcrumbs
570 // -------------------------
571 @breadcrumb-bg:               #f5f5f5;
572 @breadcrumb-color:            #ccc;
573 @breadcrumb-active-color:     @gray-light;
574 @breadcrumb-separator:        "/";
575
576
577 // Carousel
578 // ------------------------
579
580 @carousel-text-shadow:                        0 1px 2px rgba(0,0,0,.6);
581
582 @carousel-control-color:                      #fff;
583 @carousel-control-width:                      15%;
584 @carousel-control-opacity:                    .5;
585 @carousel-control-font-size:                  20px;
586
587 @carousel-indicator-active-bg:                #fff;
588 @carousel-indicator-border-color:             #fff;
589
590 @carousel-caption-color:                      #fff;
591
592
```

```
// Close
// -------------------------
@close-font-weight:           bold;
@close-color:                 #000;
@close-text-shadow:           0 1px 0 #fff;


// Code
// -------------------------
@code-color:                  #c7254e;
@code-bg:                     #f9f2f4;

@pre-bg:                      #f5f5f5;
@pre-color:                   @gray-dark;
@pre-border-color:            #ccc;
@pre-scrollable-max-height:   340px;

// Type
// -------------------------
@text-muted:                  @gray-light;
@abbr-border-color:           @gray-light;
@headings-small-color:        @gray-light;
@blockquote-small-color:      @gray-light;
@blockquote-border-color:     @gray-lighter;
@page-header-border-color:    @gray-lighter;

// Miscellaneous
// -------------------------

// Hr border color
@hr-border:                   @gray-lighter;

// Horizontal forms & lists
@component-offset-horizontal: 180px;


// Container sizes
// --------------------------------------------------

// Small screen / tablet
@container-tablet:             ((720px + @grid-gutter-width));
@container-sm:                 @container-tablet;

// Medium screen / desktop
@container-desktop:            ((940px + @grid-gutter-width));
@container-md:                 @container-desktop;

// Large screen / wide desktop
@container-large-desktop:      ((1140px + @grid-gutter-width));
@container-lg:                 @container-large-desktop;
```

XII. Acknowledgements

Thanks to:

Sir Greg Baes.

Family. Cecilia Quiambao, Cecil Morella, Severina Quiambao, Grace Bayaua, Tiks Quiambao, Rebecca Quiambao.

Block 12. Too many to mention.