Hygieia

Presented by: Yuji Shimojo, Cara Howie, and Thomas Vera

For: Dr. Clarence Huff CMSC 495

Created On: March 3, 2017

Table of Contents

Contents

Overview

This project started out with three different individuals seeking to build a computer project and resulted in a hospital management system, which with more time and development could be implemented into real world scenarios. We have developed this software to allow hospitals to input information only once and reduce the amount of work required to keep track of medical records. This type of management system overall will improve hospitals' efficiencyiy and make them more effective as doctors will have access to a patients entire health care records on one easy to read platform. Additionally, theses medical records can easily be converted to an encrypted zip file where patient files can be transferred between hospitals more effectively.

Software Design Components

The broad overview of the Hygieia system allows users to interface with the program through third party browsers like Internet Explorer, or Mozilla Firefox over port 80 (HTTP). Once the browser starts the TCP/IP handshake process the browser gets connected to an Apache Server. This server then connects to the Apache Tomcat server over port 8009 which then pulls up the JSP pages. These JSP pages then call to the Servlet which then call to the SQL database over port 3306 which stores all of the data for the system. Once the data is retrieved the process is completely reversed heading back to the Servlet over port 3306 back to the JSP which goes back to the Apache server over port 8009 and finally

reaches back to the user over port 80. See figure 1.1 for a visual representation of the state approach. The next items will break down into each page the design plan that goes behind it.

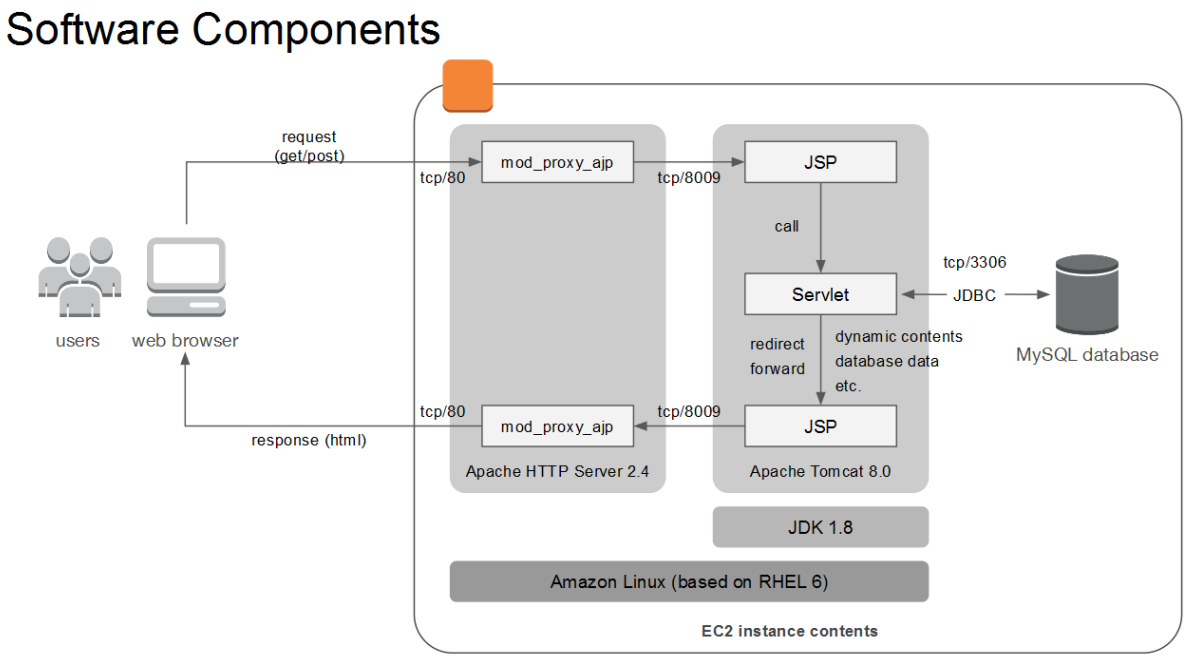

Figure 1.1

Hygieia Layout

Login Page Design:

The login page displays the logo for our system, and allows a user to log in with a valid username and password. This login page contains two labels that are *Login Name* and *Password* which are used to identify their respective text fields, so that the user can easily understand where to enter their information. Additionally, there are two buttons at the bottom called *login* and *reset*. When the *login* button is clicked the program checks whether either field is blank. If one is then the program throws an internal error that then redirects the user back to the login page. If the login name and password fields

are not empty then the system captures those two fields and sends the information to the backend database using the sql query “SELECT id FROM users WHERE login_name = ? AND password = ?”. The backend database will then try to find that information. If the authentication failed then the program sends the user back to the login page. If the user successfully logs in then the program will send the user to the *Home* page. In figure 1.2 there is a n image of the login page.

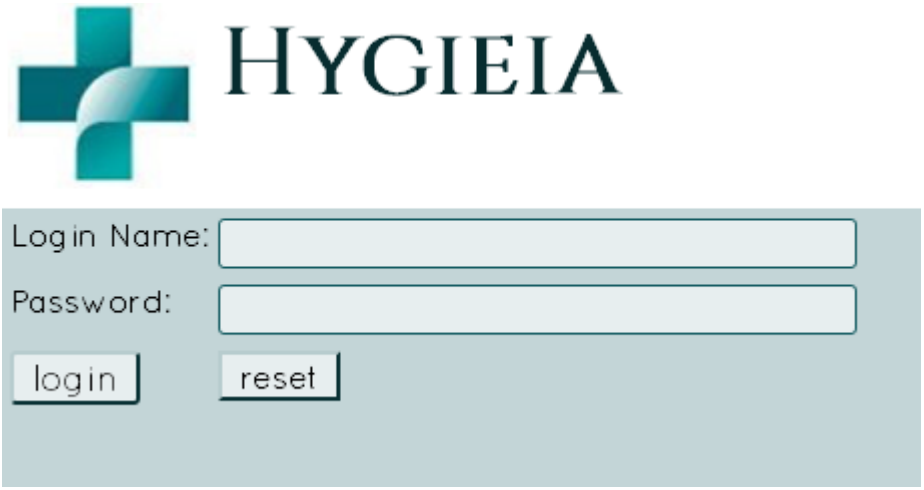

Figure 1.2

Home Page:

The home page is used to display information specific to the user currently logged in. It shows them their account’s role type as well as username, and address. See figure 1.3 below.

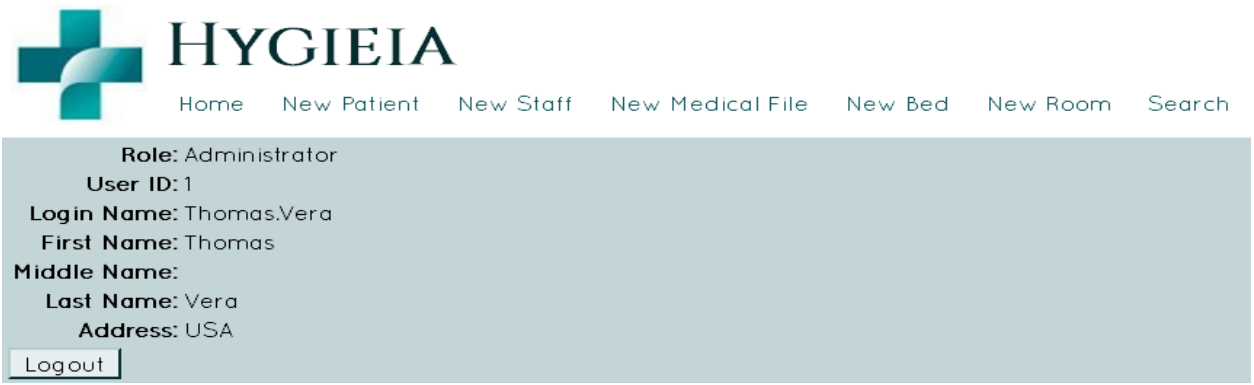

Figure 1.3

New Patient Page:

The next page is the *New Patient* page, which is used for hospital employees to enter new patients into their system and record basic information that can be used during their current and future visits. The form has twelve fields that allow users to enter various information. The first section is for the patient's name and is broken down into three separate fields. There are three labeled text input fields *First Name, Middle Name*, and *Last Name*. T fields allow all alphabetical characters to be inputted; however, we do not allow any numbers or ";" or " ' " as these can lead to SQL injections on the back end. Additionally the name fields check that name length is less than or equal to 30 characters. The first and last names are required fields, but the middle name is not. If the input is invalid for any of these reasons the program notifies the user when they attempt to submit the form. The next field is *SSN,* is for the patient's social security number. This allows a nine digit number with no dashes, and the program auto generates an example for the user to follow. If the field is left blank or does not contain nine numeric characters, an error will be thrown alerting the user that the format is incorrect. The next field is labeled *Admitted Date* which is used by doctors and billing staff. This field must be formatted as MM/DD/YYYY, otherwise it will throw and error. Additionally, if it is left black it will also throw an error and notify the user that the field has the incorrect format. The next field is labeled Insurance, which allows the medical

provider to input the patient's insurance information. Then the next section is used for assigning a doctor to the primary care. These fields are labeled together as "*Doctor Name* and have placeholder text to differentiate the *First* and *Last* name text input fields. Then there is a radio button group, which contains two radio buttons labeled *Inpatient* and *Outpatient*. This field is used mainly for billing, and automatically selects *Inpatient*, so an employee must manually change the selection to *Outpatient* if necessary. The next field is *Address*, where a patient's mailing address will go. The system checks whether the text field contains over 100 characters and displays and error to the user and will not submit the form until it is corrected. Then there are two *Password* fields that allow the user to create a complex password that has one uppercase, one lowercase one number, one special character, and is at least 12 characters long. The password will be used by the patient's to gain access to the system to review their medical files only. If the two password fields' input does not match, an error is displayed to the user and they cannot successfully submit the form until it is corrected. Finally, there are two buttons located on the bottom left corner, which are labeled *Submit* and *Logout*. The *Submit* button will check for all the things stated above and then output any error messages. If no errors are found then is will submit the information on the form to the backend database. The *Logout* button will close out the webpage and bring the user back to the login screen. In figure 1.4 a rough draft of what the page will look like is displayed. Figure 1.5 shows an example of error messages being displayed.

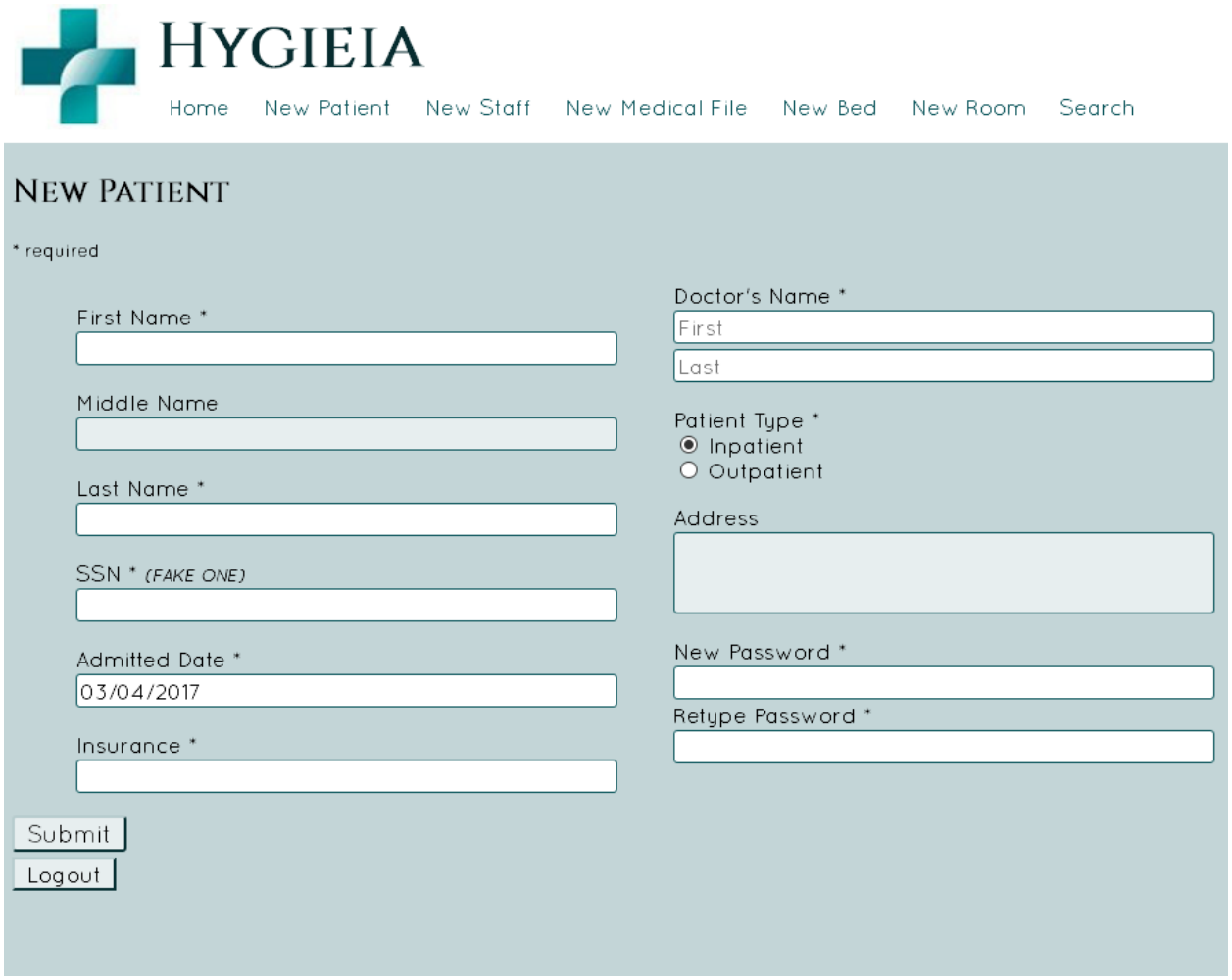
HYGIEIA
Home
New Patient
New Staff
New Medical File
New Bed
New Room
Search
NEW PATIENT
* required
First Name *
Middle Name
Last Name *
SSN * (FAKE ONE)
Admitted Date *
03/04/2017
Insurance *
Doctor's Name *
First
Last
Patient Type *
Inpatient
Outpatient
Address
New Password *
Retype Password *
Submit
Logout

Figure 1.4

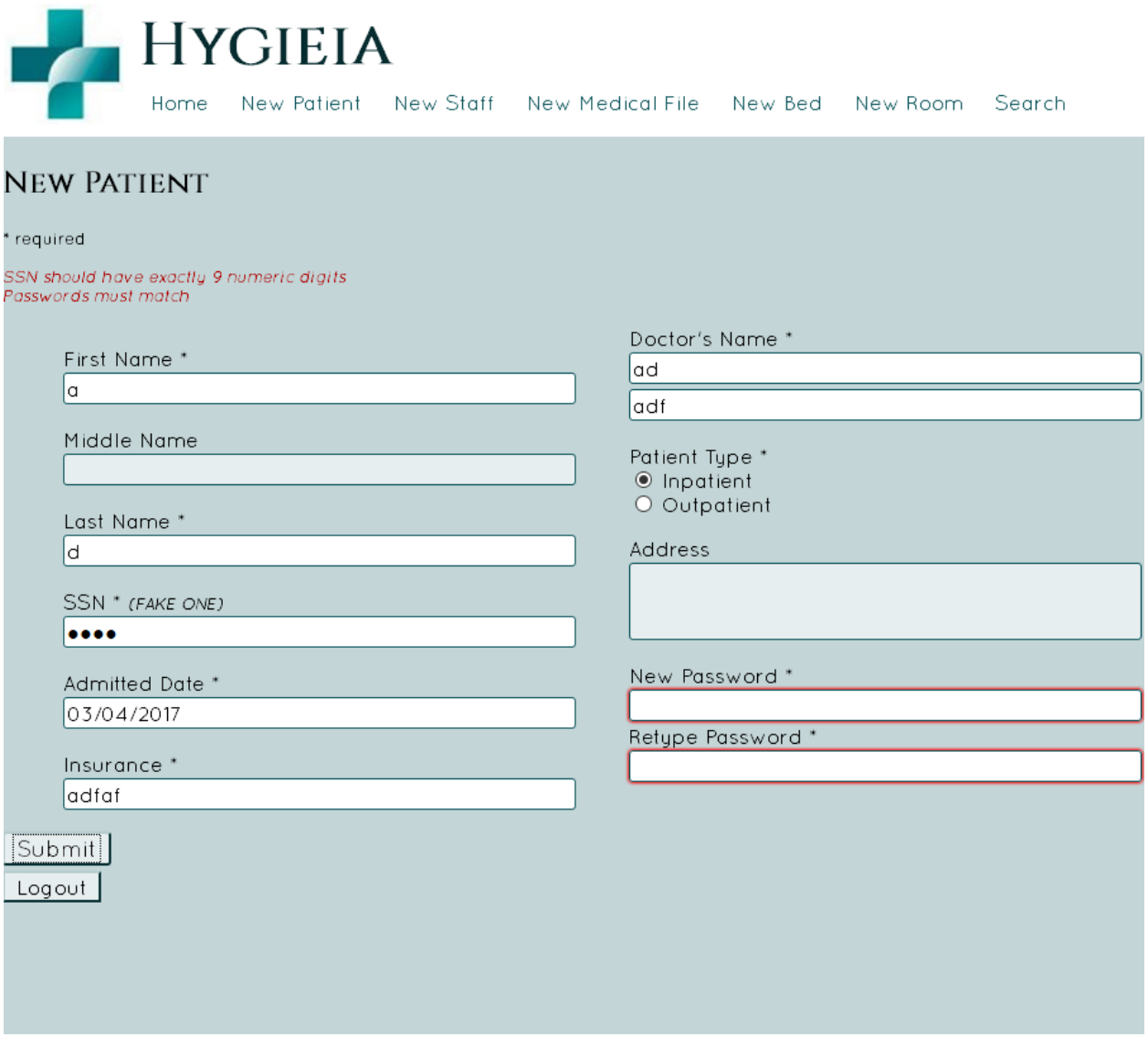

Figure 1.5

New Medical Files Page:

The medical files page is used both by patients and the medical staff. The patients only have read access to their files, but the medical staff is allowed to modify information in this form. The first field is *Patient* ID, which is used to identify which patient is currently being treated. This ensures that all medical records are attached to a unique patient and are not shared. The next field is labeled *Date of Visit* which is used to document the initial day the patient came into the hospital for this particular

symptom. The format of this field must be MM/DD/YYYY and any other format will give an error. The next field is a checkbox labeled *Ambulance Service Used*. This field is primarily used for billing, and other statistical information. The next field is labeled *Bed Name*, which must match the name of one of the beds that the hospital has. Once a bed is selected, it then gets disabled and cannot be selected by another medical team until this patient is no longer using it. This ensures that the same bed is not registered to two different patients. The next two fields work in conjunction to determine the current status of equipment in the hospital and patient's billing amount. They are labeled *Start Bed Date* and *End Bed Date*. Both of these fields will need to be formatted the same way as all other dates (MM/DD/YYYY). If they are formatted any other way then an error will be displayed telling the user that the format is incorrect. The next field is labeled *Disease Name*, which is used for the diagnosis. This field accepts alpha/numeric characters and can hold up to 200 characters. If this field contains more than 200 characters and/or contains special characters it will send an error and alert the user that the format is invalid. The next field is labeled *Treatment* that is a text field where the medical team can enter the treatments they have attempted thus far. This text field can handle up to 1000 alpha/numeric characters and will only send out an error if one of these conditions is broken. The next field is labeled *Medicine Name,* which is a text field that can handle up to 200 alpha/numeric characters. If there are special characters inputted here the system will send out an error message to the user informing them that there are illegal characters in this field. The next field is a check box that is labeled *Administered*. This should be checked if once a patient is given their medication. There is also a field for *Medicine Name*, which is optional unless *Administered* is checked because if a medicine is given, then there should be a record of what type it was. The next field is labeled *Medical Notes,* which can be used for any information not recorded elsewhere on the form such as: symptoms, descriptions of pain, or discolorations. This field can hold up to 1000 alpha/numeric characters, and if this limit is reached an error message will appear. The final field that requires user keyboard input is *Billing Amount,* which is

used to enter the total bill for a patient's care. The required format is a string of numbers with two decimals after the decimal point. The field will not allow negative numbers or non-numeric characters (including currency symbols). The maximum length of the bill is set to 11 characters, which includes the tenths, and hundreds places. If any input besides numeric digits and the decimal point is inputted then the system will display an error message. Finally, there are two buttons on the bottom of the screen labeled *Submit*, and *Logout*. The *Submit* button validates that the input follows all rules stated above and produces any necessary error messages. If no errors are found then the data is passed to the database. The *Logout* button performs the same action as above where it send the user back to the login screen without saving any data. See figure 1.6 below

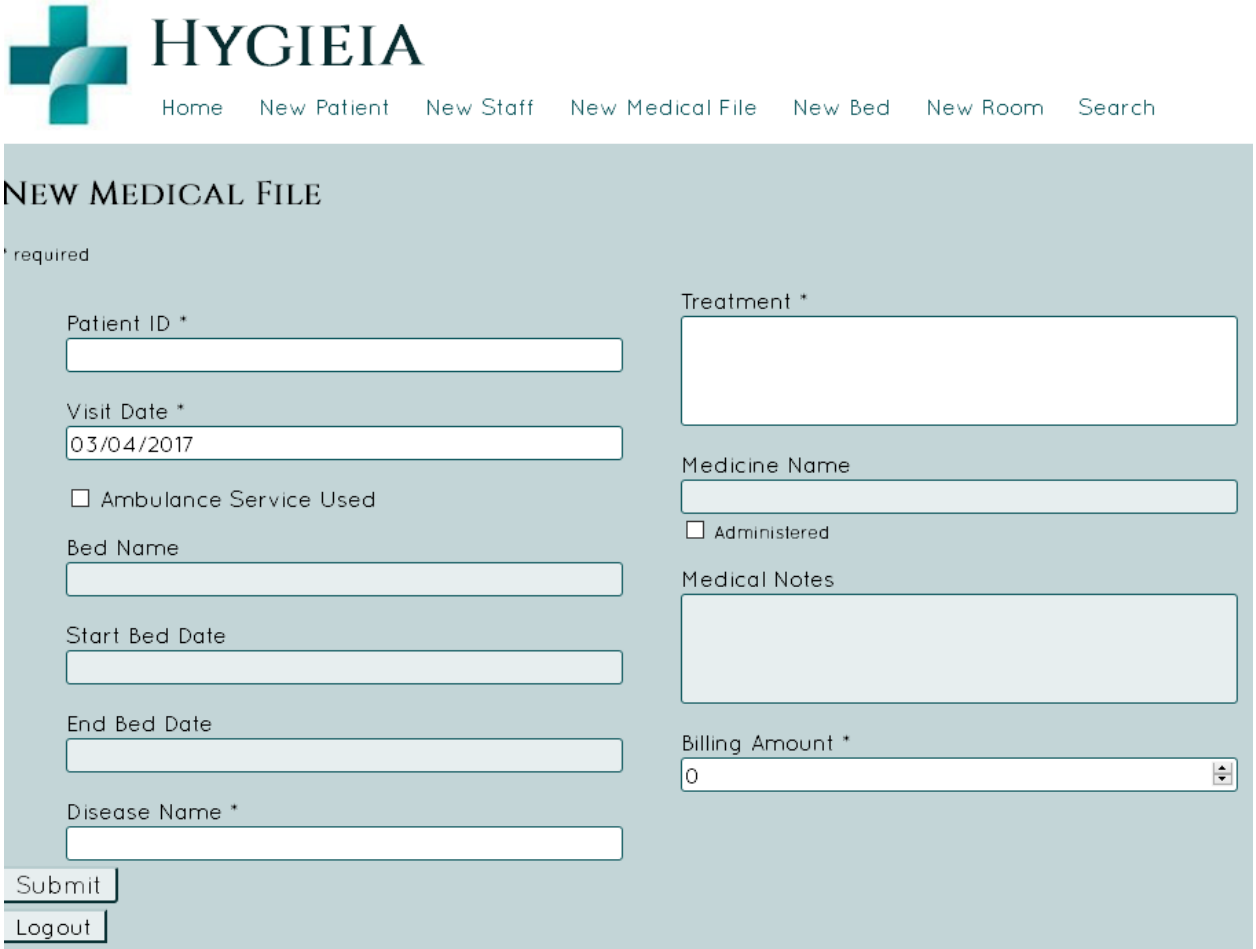

Figure 1.6

New Staff Page:

The *New Staff* page is used to add new employees at the hospital. The first fields are labeled *First Name*, *Middle Name*, and *Last Name.* These fields accept up to 30 alphabetical characters per field, and must have something in the fields except for the middle name. If one of these conditions is not met then the program will throw an error and display it to the user. The next field is a group of radio buttons labeled *Staff Type* which has two radio buttons called *Staff* and *Doctor*. These fields are used to determine where this information is going to be saved in the database. The next field is labeled *SSN,* which acts the same way as New Patient's SSN with the same potential errors. The next field is labeled *Address* which acts the same way as the Patient *Address* in the new patient page, and has the same errors as well. Next is a field labeled *Qualifications,* which allows for up to 500 alpha/numeric characters. This field is used to describe education level achievements and certifications. If there are any special characters or the character limit is reached an error will display letting the user know that the format is incorrect. The next section is labeled *Certifications Expirations,* which is used to keep track of when employees' certifications expire. The next field is labeled *Cell Number* which is used for emergency calls or other professional business. This number must be inputted using the format DDDDDDDDDD with no space, dashes, or parenthesis. If the data inputted is not all numbers and does not equal the ten digit requirement then an error is generated displaying how to properly input the information in this field. The next field is labeled *Email Address* and is used as another avenue to contact staff members about important information. This field allows all characters except " ' " and " = ". This field also requires a "@" symbol and a ".com/.edu/.net/.gov" be at the end of the field. It is also limited to 50 characters. If any condition is not met then there error message displays to the user. The next field is labeled *Payroll* and has the same limitations as *Billing Amount* from the *New Medical Files* page. The next field is labeled

Personnel Notes which is used to any other information that the employer should know that is not elsewhere on the form, such as disabilities. This field accepts upt to 200 alpha/numeric characters. The next section is for the *Password and* has the exact same rules as the *New Patient Password* fields do. Then there are *Clock In/ Clock Out* fields along with a *Status* radio group button that is used to identify what times the employee comes into work and whether it is under normal or emergency conditions. Finally, this page has the same buttons as the medical files page at the very bottom, with the same functionality. See figure 1.7 below.

Figure 1.7

New Bed:

The *New Bed* page allows administrators to add new beds to existing rooms. The first field is *Room ID* which looks to be an Id number associated with a room that is already in the system. The next

field is *Bed Name* which uses the same rules as the *First Name* fields above in the previous sections. See figure 1.8 below

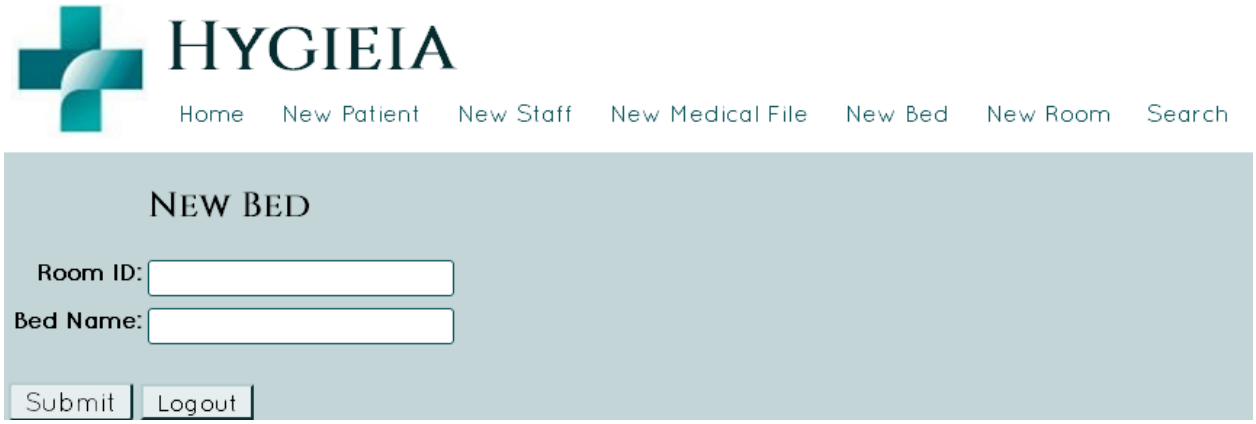

Figure 1.8

New Room:

The *New Room* page allows administrators to add additional rooms to different floors. The first field is *Room Name* which follows the same rules as *Bed Name.* The next field is *Floor* which follows the same rules as *First Name* in the fields above. See figure 1.9 below

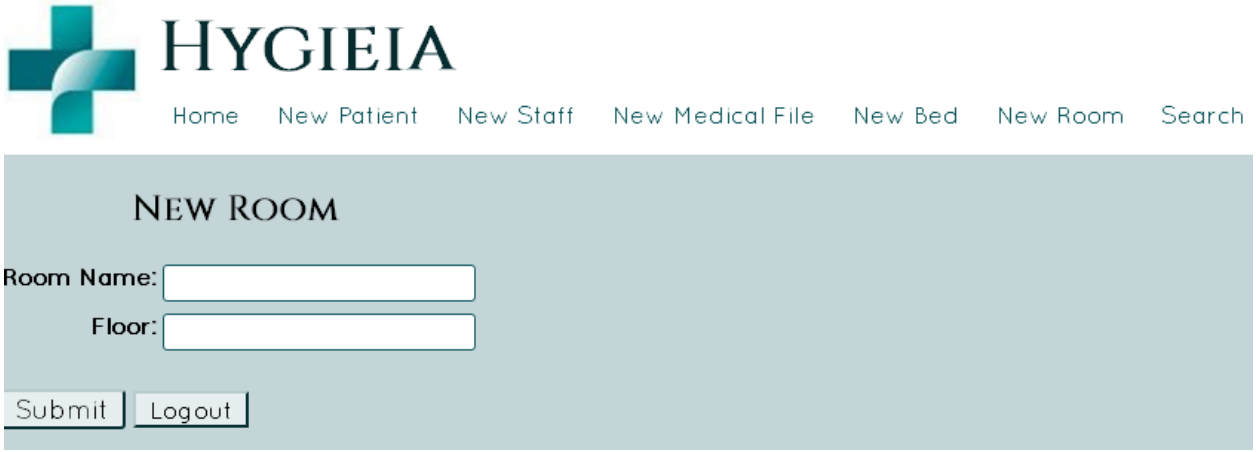

Figure 1.9

Search Page:

The search page is used to search the system for existing documentation on certain people/files. The first field is labeled *Search for,* which has a drop down box with the following options in it, *User,* and *Medical File* to indicate which records you are searching. The next field is labeled *First Name* which follows the same rules at the *Name* fields above. The next field is labeled *Last Name* which follows the same rules as *First Name*. Finally at the button is two buttons *Search* and *Reset.* The *Search* button checks to for the fields stated above and if no errors are found it submits the request. The program then looks into the database and pulls the proper files linked with the requested information. See figure 1.10 below.

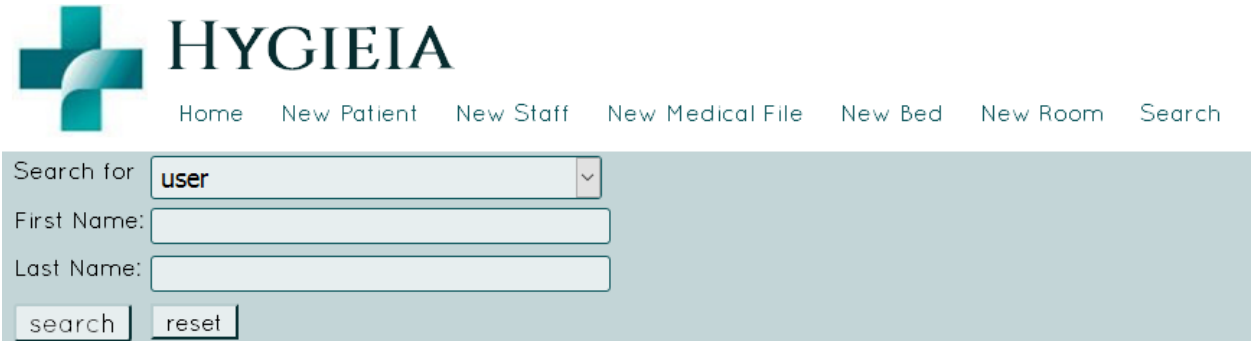

Figure 1.10

Overall Page Functions

Each page (except Login) has navigation tabs at the top labeled *Home, New Patient, New Staff, New Medical File, New Bed, New Room,* and *Search*. Depending on the account type of the user logged in, not all of these may be displayed. Clicking on one of these tabs redirects the system to the desired webpage.

Database Design

The database is linked in a multitude of ways that makes it flexible yet efficient. It boost a time and data stamp protocol that changes every time an element is modified The first table is called “Users” which holds username, passwords, roles, first/middle/last names, SSN, and address. The table is then linked to the “Staff” table which contains the qualifications, certification expirations, cell phone number, email address, payroll, personnel details, and a Boolean of is doctor. Thus the “Doctor” table is connected to this table as well as the “Shift” table which has clock in time, clock out time and status. The “Patient” table is connected to the “Doctor” table and has the type of patient (in or out) and the type of insurance. The “Bed_Usage” table is then connect to the “Patients” table and describes the state date, end date and status. This table is then connected to the “Beds” table which has the name of the bed listed which is then connected to the “Rooms” table. This table has the name of the room and the floor number. Finally the “Patient_Records” table has the initial visit day, disease name, treatment, medicine given (yes or no), medicine name, medical notes, ambulance services (yes or no) and the billing amount. It is connected to the “Beds” table and the “Patients” table. For a more graphical view see figure 1.11.


USERS
ID :INT (U)
CREATED_AT :DATETIME
UPDATED_AT :DATETIME
LOGIN_NAME :CHAR(30) (U)
PASSWORD :CHAR(50)
ROLE :TINYINT
FIRST_NAME :CHAR(30)
MIDDLE_NAME :CHAR(30)
LAST_NAME :CHAR(30)
SOCIAL_SECURITY_NUMBER :CHAR(15) (U)
ADDRESS :CHAR(100)
STAFF
ID :INT (U)
USER_ID :INT
CREATED_AT :DATETIME
UPDATED_AT :DATETIME
QUALIFICATION :CHAR(50)
CERTIFICATION_EXPIRATIONS :DATE
CELL_PHONE_NUMBER :CHAR(20)
EMAIL_ADDRESS :CHAR(50)
PAYROLL :CHAR(30)
PERSONAL_DETAILS :VARCHAR(500)
IS_DOCTOR :BOOLEAN
DOCTORS
ID :INT (U)
STAFF_ID :INT
CREATED_AT :DATETIME
UPDATED_AT :DATETIME
PATIENTS
ID :INT (U)
USER_ID :INT
DOCTOR_ID :INT
CREATED_AT :DATETIME
UPDATED_AT :DATETIME
TYPE :TINYINT
INSURANCE :CHAR(100)
PATIENT_RECORDS
ID :INT (U)
PATIENT_ID :INT
BED_ID :INT
CREATED_AT :DATETIME
UPDATED_AT :DATETIME
VISIT_DATE :DATE
DISEASE_NAME :VARCHAR(500)
TREATMENT :VARCHAR(1000)
MEDICINE_GIVEN :BOOLEAN
MEDICINE_NAME :VARCHAR(500)
MEDICAL_NOTES :VARCHAR(1000)
AMBULANCE_SERVICE_USED :BOOLEAN
BILLING_AMOUNT :CHAR(30)
BED_USAGE
ID :INT (U)
BED_ID :INT
PATIENT_ID :INT
CREATED_AT :DATETIME
UPDATED_AT :DATETIME
START_DATE :DATE
END_DATE :DATE
STATUS :TINYINT
SHIFTS
ID :INT (U)
STAFF_ID :INT
CREATED_AT :DATETIME
UPDATED_AT :DATETIME
CLOCK_IN_TIME :TIME
CLOCK_OUT_TIME :TIME
STATUS :TINYINT
BEDS
ID :INT (U)
ROOM_ID :INT
CREATED_AT :DATETIME
UPDATED_AT :DATETIME
NAME :CHAR(30)
ROOMS
ID :INT (U)
CREATED_AT :DATETIME
UPDATED_AT :DATETIME
NAME :CHAR(30)
FLOOR :TINYINT

Figure 1.11

Contributions/Development History

The creation of this management system was only possible because of the contributions made by Yuji Shimojo, Cara Howie, and Thomas Vera. Yuji Shimojo was instrumental into this projects success. He built and designed the backend database, created the live test environment, consolidated all the source code, and founded a collaborative space for all of use to communicate ideas in. Cara Howie was paramount in designing the front end interaction forms, as well as developed multiple style guides which enabled the system to be user friendly and self-explanatory. Thomas Vera provided all the documentation guides as well as phase documentation to explain and express how the teams process was doing. In addition to this the entire team spent time looking over each other's code to ensure that the code optimized while still remaining secure. We were able to accomplish this by centralizing all of our source code onto GitHub with the following URL link https://github.com/yujishimojo/CMSC495_HospitalManagementSystem .

While this project was being developed we were able to pull our contributions together to tackle difficult problems. One of the first road blocks we faced was getting the backend end database set up with the proper fields. This took multiple iterations as we continued to optimize the back end without losing any important information. The next challenge was setting up the front end web pages/forms and deciding what fields needed to be a part of the forms. Once this was decided the next challenge was integrating the front end and the back end to work together. This took multiple attempts, but we were finally able to get the system to work. Some of the final challenges came at the end when we were polished the final product, and re implemented the system.

What is this project supposed to do?

This project is designed to allow a hospital to have a single enterprise system that ensures data needs only to be entered one time into the system. This will allow multiple departments to see relevant information, but only certain roles can modify certain fields. Additionally this will improve the efficiency of the hospital and reduce the number of errors.

System Specification

- 2.5GHz processor
- 1 GB of Ram
- JDK version 8.0
- Internet Connectivity
- Tomcat version 8.0
- JSP/HTML/CSS/JavaScript as frontend
- Servlet as backend
- MySQL version 5.7

Case Scenarios

1. Once going to the web interface for the system, the user will be prompted for a username a password.
2. Based on the role of the user, certain forms will be displayed. For patients only their medical records will be displayed, while for medical staff have access to *New Patient, New Medical File,* and *Search* web pages, and administrators have access to the entire system.
3. For administrators they have the ability to input new patients, new staff members, new medical files and the ability to search for users and medical files.
4. For medical staff members they have the ability to input new patients, new medical files and to search up users and medical files.
5. For patients they only have access to look up their own medical files.
6. Once any of these forms have been modified or reviewed the user clicks the *Submit* button which then sends all the inputted data to the database.
7. The database then timestamps the entry and saves the data in the appropriate fields.
8. Once all modification and reviews are done the user then clicks a log out button which securely logs the user off.

Milestones

Jan 22 - Design what the system is supposed to do and how different pieces fit together - Team

Jan 25 - Create database tables and fields - Yuji

Jan 25 – Create user guide and test plan - Thomas

Jan 25 - Create the login/logout web page - Cara

Jan 26 – Review login/logout web page – Yuji, Thomas

Jan 28 - Create the new patient registration page – Cara / Yuji

Jan 30 – Review new patient registration page – Thomas / Yuji

Feb 2 – Create employee records page – Cara

Feb 5 – Review employee records page – Yuji/ Thomas

Feb 13 – Create patient medical file page – Cara

Feb 17 – Review patient medical file page – Thomas/Yuji

Feb 23 – Create hospital search page - Cara

Feb 25 – Review hospital search page – Yuji / Thomas

Feb 27- Connect all pieces with backend data base - Yuji

Feb 28 – Verify all data is transferred to database - Yuji

Mar 1 – test valid and invalid data inputs. – Thomas

Mar 3 – Final polish - Team

Mar 5 – Turn in project - Thomas

Team Member Roles & Responsibilities

Yuji Shimojo – Database lead, backend development lead, Program reviewer

As database lead Yuji will design and implement the creation of tables and fields that hold all our patients data. Yuji will also verify that the data being sent to the database is correct, and ensure that the database is properly secured from traditional manipulation techniques. As the backend development lead Yuji will implement servlets with database connections. Additionally, Yuji will help review and make suggestions for other aspects of the project.

Cara Howie – Website design lead, Program reviewer

As the website design lead Cara will design, implement the front end interface that all the users will interact with. Cara will also ensure that the code has been designed to work efficiently, and that the code has been designed to prevent common manipulation techniques. Additionally, Cara will help review and make suggestions for other aspects of the projects.

Thomas Vera – Project lead, Technical writer

As project lead Thomas will ensure to keep up to date on group progress and any issues that will arises. Thomas will also provide any help in any area like designing test data and make suggestions for improvement. Additionally, Thomas will write all documentation for user guides, and manuals that will help troubleshoot any issues.

User Guide

Introduction:

Our team of developers has dedicated their time and efforts in hopes of providing enterprise solutions at cost effective prices to local hospitals. This new system will allow all patient records to be to

be stored in a centralized database that will improve the overall efficiency of your hospital and allow for future growth. This allows for additional modules to be added as they come out and/or as your hospital needs see fit.

Hygieia:

Hygieia comes with the base components to allow hospitals to added new patients, create new medical files for patients, and employee records. All these components are stored in the centralized database that has multiple layers of security.

New Patient	New Medical Files	New Staff	New Bed	New Room	Search
Patient Name	Patient ID	Staff/Doctor Name	Room ID	Room Name	Search Type
Patient Social Security Number	Date of Visit	Staff Social Security Number	Bed Name	Floor	Name
Admitted Date	Ambulance Service	Staff Type			
Doctor's Name	Bed Name	Staff Address			
Patient Type	Start Bed Date	Qualifications			
Patient Address	End Bed Date	Certifications			
Patient Insurance	Disease Name	Expirations			
Patient password	Treatment	Cell Number			
	Medicine Name	Email			
	Administered	Payroll			
	Medical Notes	Personnel Notes			
	Billing Amount	Staff password			
		Clock In/Out time			
		Status			

Overall System Features:

- **Modular design:** Each module is designed to satisfy a singular functionality of a hospital. This allows the system to update only certain modular and not affect existing ones, creating a more stable environment.
- **Relational database:** MySQL is used to store and collect all of the systems data. This provides the benefits of high performance as well as high availability, which in a fast pace environment is

critical to success. It also provides one of the lowest total costs of ownership meaning that the setup and long term maintenances of running the server cost effective in the long run. Additionally, it proved the ability to growth with the hospital as new capabilities are integrated into the hospital.

- **Web Interface:** The systems modules are designed to work on any modern web browser, allowing any computer or mobile device to be capable of accessing to the system with in the hospital.
- **Security:** The system comes with a secure log in feature and allows the administrator to assign roles to different users. These different roles ensure that only certain people can modify the data that is relative to them.

Modules Description

New Patient:

Overview: Once the information below is sent to the backend database employees can search data up by any of the fields listed below. If any discrepancies are found then an authorized user must enter the system and modify the appropriate field.

1. **Patient Name:**

There are three different fields that are labeled *First Name, Middle Name,* and *Last Name*. Using them, a general staff member will be able to add in the patient's full name which then gets associated with the database on the back end. The *First* and *Last* name are required before the system allows a user to *Submit* the form.

2. **Patient Social Security Number:**

   This required field allows general staff members to add the patient's social security number. As the administrator types in the social security number, black dots are displayed onscreen instead of the password so that it cannot be read over the user's shoulder.

3. **Admitted Date:**

   This required field is automatically populated by the date that form is filled out based on the date and time of the server running the frontend, but may be changed manually by the user. The format must be *MM/DD/YYYY*. Example (01/22/2019).

4. **Patient Insurance:**

   This required field allows general staff members to add the patient's health insurance, which will help calculate the final bill. Additionally, it will help determine how much your hospital must charge to remain profitable.

5. **Doctor's Name:**

   This required field has a *First* and *Last* name section that needs to be filled out by a proper employee. The doctor's name will need to match one of the existing doctor records in the back end database. This gives a quick way to see who initially diagnosed the symptoms.

6. **Patient Type:**

This required field is a radio button that allows the general staff to select whether this person is an *Inpatient* or an *Outpatient*. This field will also impact the billing and what information will need to be filled out for the medical file forms.

7. **Patient Address:**

   This field will allow general staff members to add the patient's physical address which can be used for mail, such as reminder cards and billing.

8. **Password:**

   This required field is where a password is inputted to allow the new patient to have access to only their own records.

New Medical Files:

Overview: Once the information below is sent to the backend database medical staff can search data up by any of the fields listed below. If any discrepancies are found then an authorized user must enter the system and modify the appropriate field. This module is to be used by only medical staff for the patient's health information.

1. **Patient ID:**

   This required field is used to associate patients with their medical records. It ensures that medical file search results for a specific patient are comprehensive and contain no other patients' records.

2. **Date of Visit:**

This field is automatically populated by the date that form is filled out based on the date and time of the server running the frontend, but may be changed manually by the user. The format must be *MM/DD/YYYY*. Example (01/22/2019).

3. **Ambulance Service:**

   This is a check box that indicates whether the patient was transported to the hospital using an ambulance. This information needed since the cost goes toward the patient's final bill once the patient is fully recovered. It is important to note that this field only needs to be checked once per visit if multiple forms are filled out for the same visit so that the patient is only charged once.

4. **Bed Name:**

   This field is used to identify which bed the patient is currently using and to indirectly indicate to the medical staff the patient's location.

5. **Start Bed Date:**

   This field (along with end bed date) is used to track time spent in the hospital for billing and determine bed availability. The format must be *MM/DD/YYYY*. Example (01/22/2019).

6. **End Bed Date:**

   This field is required for outpatients and inpatients that stay for over 24 hours. This field (along with start bed date) is used to track time spent in the hospital for billing and determine bed availability. The format must be *MM/DD/YYYY*. Example (01/22/2019).

7. **Disease Name:**

   This field allows for medical staff members to enter the patient's current diagnosis.

8. **Treatment:**

   This field allows the medical staff members to enter the recommended treatment. This allows doctors to view patient treatment history.

9. **Medicine Name:**

This field is writable and is required to send the information to the back end database only if the *Administered* check box is checked.

10. **Administered:**

    This field is a check box that is defaulted as unchecked, but should be checked once the medicine is given to the patient.

11. **Medical Notes:**

    This field allows the medical team to write a brief summary of information not covered elsewhere in the file, such as stats on the patient, and descriptions of symptoms.

12. **Billing Amount:**

    This field is used to enter the total bill of the service provided on this medical form. This field must be filled out using two decimal points and can be incremented up or down a cent by using the up and down arrows entered using the keyboard. The form does not accept non-numeric characters (including currency symbols) or negative numbers (for example: 53.68).

New Staff

Overview: Once the information below is sent to the backend database administration staff can search data up by any of the fields listed below. If any discrepancies are found then an authorized user must enter the system and modify the appropriate field. This module is to be used by only administration staff for hospital employee information.

1. **Staff/Doctor Name:**

   There are three different fields that are labeled *First Name, Middle Name,* and *Last Name*. Using them, an administrative staff member is able to add in the employee's full name,

which then gets associated with the database on the back end. In the case of doctors, their name gets added to a database that is used throughout the other modules.

2. **Staff Social Security Number:**

   This required field allows general admin members to add the employees' social security number. As the administrator types in the social security number, black dots are displayed onscreen instead of the password so that it cannot be read over the user's shoulder.

3. **Staff Type:**

   This required field is two radio buttons labeled *Staff* and *Doctor*. The *Staff* button is automatically selected. If the *Doctor* button is selected it will tell the program to add the doctor's name to the database mentioned above.

4. **Staff Address:**

   This field will allow admin staff members to add the employees' physical address, which can be used for sending paychecks, notifications, or in cases of emergencies.

5. **Qualifications:**

   This required field is used for describing employees' qualifications, such as certifications, higher educations, years of experiences, and medical awards won. This information can be useful to human resources when they need to replace staff members, or for annual reviews of qualifications.

6. **Certification Expirations:**

   These fields include a text box that is used to type in the expiration date of the certification stated above. The expiration date needs to be inputted in the following format *MM/DD/YYYY*. Example (01/22/2019).

7. **Cell Number:**

This field is used to save the cell number of the employees as a ten-digit number with no special characters. This information can be used for recalls and emergency situations.

8. **Email Address:**

   This required field is a textbox where admin staff members can add a staff member's email address. The field must end with *@xxxxx.xxx*. Example (doctored@ha.net). This will be used to in contact with staff members in not time sensitive, for example letting them know their certification is about to expire.

9. **Payroll:**

   This required field is a textbox where admin staff members can add the base pay of the staff member. This field is then used for pay checks, and review of profits vs lose in the quarterly hospital budget meetings.

10. **Personnel Notes:**

    This field allow admin staff members to write down any need notes like disabilities, garnish wages, disciplinary actions, or bonuses.

11. **Password:**

    This required field is where a password is inputted to allow the new staff member to have access to the system. Once they have access to the system they will be able to modify certain fields base on their roles.

12. **Clock In/Out Time:**

    This field is used to record when employees clock in and clock out of work. They must use the 24-hour time format of HH:MM:SS. Ex. (11:21:00)

13. **Status:**

This field is two radios buttons that say *Normal* and *Emergency* and should be used to tell why an employee came in to work. If they were scheduled then they should use the *Normal* button, while if they were called in they should use the *Emergency* button.

New Bed

1. **Room ID:**

   This field is used only by administrators and associates a new bed with the room that they are in. This feature can be used when new beds are put in to replace old ones.

2. **Bed Name:**

   This field is used to differentiate between multiple beds that may potentially be in the same room.

New Room

1. **Room Name:**

   This field is used only by administrators and identifies the new room is being created.

2. **Floor:**

   This field is used only by administrators and is used to tell the system what floor the new room should be associated with.

Search

1. **Search Type:**

   This field is a drop down box that is labeled *User* and *Medical Files*. Depending on which type you chose, the system will search different sets of records and display appropriate results.

2. **Name:**

This field has a *First Name* and a *Last Name* field in it which both need to be filled out in order to pull up the correct records.

Navigating Hygieia

Overview: This section discuss how any user will navigate through the system to get to the need information. Using the information here will help train and get any employee caught up to speed on this new system.

Login/Logout Page:

When a user first comes to the web interface they are presented a prompt for *Login Name:* and *Password*. The user will need to type in their user name and then type in their password. The user name is not case sensitive, but the password is case sensitive and will not allow the user to log in if it is incorrect. Once these two fields are filled out the user will need to click the *login* button which verifies the given information with the database records. If they match, then the system lets the user in. If they do not match the system redisplays the login screen with both fields cleared out. Additionally, at the login screen there is a *reset* button that can be clicked to clear both fields.

Home Page:

The system then brings the user to the home page which shows the user their role, ID number, Login name, First, Middle and Last name, and their Address. On this page there is a *Logout* button on the bottom left side of the screen which allows the user to securely exit the system. This method should always be used to logout as it is the most secure way.

New Patient:

Once in the system, there is a navigation bar at the top of the page containing links to the H*ome, New Patient, New Staff, New Medical Files, New Bed, New Room* and *Search* pages, which can allow the user to access different pieces of information based on their account type. To reach the *New Patient* page the user only need to click the *New Patient* tab up top. Once there the page displays all the information listed above in the **Module Description** sections. Once all the information is entered, the user will need to click the *Submit* button at the bottom left of the screen to save the file to the database; however, before this data is sent back the system will perform a series of checks. The checks will make sure the information is inputted correctly. If any field is inputted incorrectly the system will display a message to the user and clear the field that is invalid. Once the field has been corrected the user will need to click the *Submit* button once again. Additionally, there is a *Logout* button on the bottom left side of the screen as well.

New Medical Files:

Once in the system, there is a navigation bar at the top of the page containing links to the H*ome, New Patient, New Staff, New Medical Files, New Bed, New Room* and *Search* pages, which can allow the user to access different pieces of information based on their account type. To reach the *New Medical Files* page the user only need to click the *New Medical Files* tab up top. Once clicked the page will display all the information listed above in the **Module Description** section. Once all the information is entered the user will need to click the *Submit* button at the bottom left of the screen to save the file to the database; however, before this data is sent back the system will perform a series of checks. The checks will make sure the information is inputted correctly. If any field is inputted incorrectly the system will display a message to the user and clear the field that invalid. Once the field has been correct the user will need to click the *Submit*

button once again. Additionally, there is a *Logout* button on the bottom left side of the screen as well.

New Staff:

Once in the system, there is a navigation bar at the top of the page containing links to the H*ome, New Patient, New Staff, New Medical Files, New Bed, New Room* and *Search* pages, which can allow the user to access different pieces of information based on their account type. To reach the *New Staff* page the user only need to click the *New Staff* tab up top. Once clicked the page will display all the information listed above in the **Module Description** section. Once all the information is entered the user will need to click the *Submit* button at the bottom left of the screen to save the file to the database; however, before this data is sent back the system will perform a series of checks. The checks will make sure the information is inputted correctly. If any field is inputted incorrectly the system will display a message to the user and clear the field that invalid. Once the field has been correct the user will need to click the *Submit* button once again. Additionally, there is a *Logout* button on the bottom left side of the screen as well.

New Bed:

Once in the system, there is a navigation bar at the top of the page containing links to the H*ome, New Patient, New Staff, New Medical Files, New Bed, New Room* and *Search* pages, which can allow the user to access different pieces of information based on their account type. To reach the *New Bed* page the user only need to click the *New Bed* tab up top. Once clicked the page will display all the information listed above in the **Module Description** section. Once the two pieces of information are entered the user will click the *Submit* button and the information will be sent to the back end database after the system checks to ensure the room ID actually exist. Additionally, there is a *Logout* button on the bottom left side of the screen as well.

New Room:

Once in the system, there is a navigation bar at the top of the page containing links to the H*ome, New Patient, New Staff, New Medical Files, New Bed, New Room* and *Search* pages, which can allow the user to access different pieces of information based on their account type. To reach the *New Room* page the user only need to click the *New Room* tab up top. Once clicked the page will display all the information listed above in the **Module Description** section. Once the two pieces of information are entered the user will click the *Submit* button and the information will be sent to the back end database. Additionally, there is a *Logout* button on the bottom left side of the screen as well.

Search:

Once in the system, there is a navigation bar at the top of the page containing links to the H*ome, New Patient, New Staff, New Medical Files, New Bed, New Room* and *Search* pages, which can allow the user to access different pieces of information based on their account type. To reach the *Search* page the user only need to click the *Search* tab up top. Once at this page the user is able to select a drop down of different fields based on the **Module Description** section. Next to this drop down box will be a textbox that the user will type in the name they are looking for. This will not be case sensitive to allow for maximum search results and ensure no medical documents are missed for a patient. Then the user will need to click the *Search* button at the bottom of the left hand screen that will bring up the list of documents a user is looking for. The user will select the file they wish to see. Additionally, there is a *Reset* button on the bottom left side of the screen as well.

Minimum System Requirements for Hygieia

Hardware:

- 2.5 GHz processor
- 1 GB of RAM
- Internet Connectivity

Software:

- JDK version 8.0
- Tomcat version 8.0
- MySQL version 5.7
- Servlet as backend
- JSP/HTML/CSS/JavaScript as a frontend

Test Plan

The first part of the test plan will be to use the administrator username and password to see if we can log into the system fully from the URL http://ec2-52-91-115-171.compute-1.amazonaws.com/Login.jsp . The username will be *Dr.Huff and the password is 0r.H#y0up@s2.* The next part of the test plan it to verify that all web pages pull up properly within the system. Then the next part of the test plan is to try different inputs into each of the forms. The attached document shows a list of inputs for each form that will be tested. These inputs will help to ensure that our Hygieia can accept and handle proper and improper data. The first set of data is proper data and the goal of this test set is to see if data is transfer back to database properly. The next set of data is set to test improper values. Some of the values are letters and symbols where numbers should be and vice versa. The other data is common SQL injections attacks, which we chose to use since we are running a MySQL for the backend database. When this improper data is submitted to the system error messages should display to the user, telling them what field is wrong and clear out that field.

Additionally, we will attempt to get to each web page without logging in, and we expect that each attempt will redirect us back to the log in screen. Then test plan will then consist of trying to search for patients, employees, and medical files using the search page, which we expect to pull only proper documents related to the search field. Then our next set of test should try using a medical staff username: Dr.Clarence and password: 0r.C1@re&n(e and verify only the web pages *Home, New Patient, New Medical File* and *Search* only pull up. We will then try out input validation process again to make sure this users role does not have any flukes in their privileges. Finally we will repeat the process but with using a patient role with a username: Clarence.Huff and password: C1@re&(eHuf2. The user's role

should only display the *Home,* and *Medical Files* web pages. This role should not have any access to input into the system, but just to read records inside the system.

Results

After complete our test above we have verified that the system is running how we intended it to run. The system is using the appropriate roles correctly and is only displaying the web pages or forms that we have set with the roles. Additionally, the system is rejecting basic SQL injection attacks and validating inputs to make the system run smoothly.

Lessons Learned:

We were able to learn a lot from this lesson. One of the things we learned was that a system like this takes time to develop well. In fact if we had more time to work on it I believe we would have got a system that could be implemented in hospital today. We also would need more time in the sense that most of us work at another job and so we could not give our full attention to this project. If we had done this I believe our project could have been that much stronger. We did however play on each other's strengths which is why our project turned out so well and why we as a team did not have many disagreements. We all brought up new ideas and had others cross check our ideas with each other to ensure we are producing the best product we can. Another way to improve this project would be to add more modules and include a feature that allowed us to add, remove, and modify items such as rooms, beds, records, and personnel. Overall I believe that we did an amazing job with our project and I would work with my team again any day of the week.