Software Design Specification

Secure Hospital System (SHS)

Course 545 Project

Presented By
Group III

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this system is to develop and design a Secure Hospital System. A secure hospital system is used to allow patients, doctors, and hospital staff to manage and maintain medical information safely and securely quickly and easily. To properly maintain this private information, the system must properly store data and information so that it can only be accessed by each specific patient and the hospital staff (including doctors) that the information belongs to (Authentication and Authorization are highly important as we are dealing with sensitive PHIs).

Aim: "Develop a highly secure and reliable hospital system"

1.2. System Overview

The Secure Hospital System (SHS) will have six major roles: Administrator, Doctor, Hospital Staff, Lab Staff, and Patient. A patient will be able to create an account and from that account view their own medical information (i.e., past appointments, diagnoses, medical insurance, payments, and prescriptions), request lab tests and request appointments. Lab Staff will be able to manage lab tests (i.e., create, view, update, and delete) as well as view patient diagnoses. Hospital Staff will be able to manage patients (i.e., create new patients through emailing them a link, and view patients), update and view patient records, view lab reports/tests, request lab tests, view prescriptions, and reject/cancel appointments. Doctor's will be able to create prescriptions, recommend lab tests, update patient records, view records, view lab reports/tests, view prescriptions, request lab tests, create/update/remove diagnoses, and view patients. Administrators will be able to manage users and view logs tracking who does what within the system.

1.2.1. Patients

When accessing the system for the first time a patient will be greeted with a sign-up screen where the patient will be asked to create their account. After account creation, the patient can then sign in with that same account and will then be redirected to the patient dashboard. On the patient dashboard, the patient will be able to create an appointment by selecting the appointments tab in the navigation bar. The patient will also be able to view their past appointments on the patient dashboard as well as their medical information listed above.

1.2.2. Lab Staff

When accessing the system, Lab Staff should have an account created for them by administrators. Upon signing into the system, the Lab Staff will have access to the Lab Staff dashboard which will allow them to view all current and former lab tests in a table. From this dashboard the Lab Staff can then create new lab tests based on requests or edit existing ones from the table.

1.2.3. Hospital Staff

When accessing the system, Hospital Staff should have an account created for them by administrators. Upon signing into the system, the hospital staff will have access to the Hospital Staff dashboard. On this dashboard they will have the ability to view/search for patients and create new patients. Upon selecting a patient, the hospital staff can then view the prescriptions of the patient, update that patient's profile, view/request the patient's lab reports/tests, view/update the patient records, view the patient diagnosis and view/create transactions/bills for said patient. The hospital staff will also view the appointments tab in the header and from there they can reject or cancel appointments or approve the appointments requested by patients.

1.2.4. Doctors

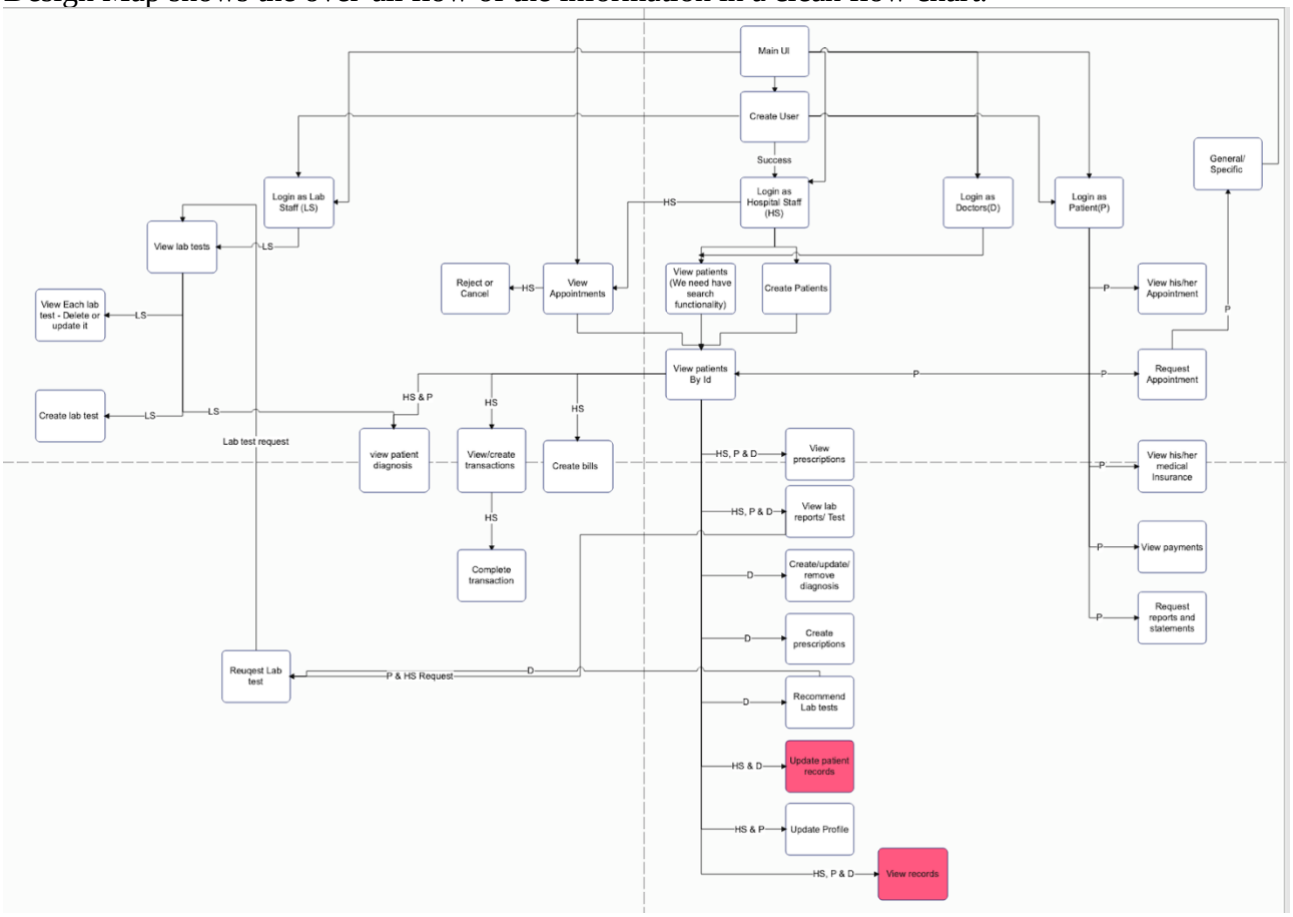
When accessing the system, Doctors should have an account created for them by administrators. Upon signing into the system, doctors will have access to the Doctor dashboard. From here doctors can view/search for patients. Upon selecting a patient, the doctor can then view/create a prescription for that patient, create/update/remove a diagnosis for a patient, view/update that patient's records, recommend a lab test to that patient, and view lab reports/tests for that patient. A doctor can also access the appointments tab from the header and from there view which upcoming appointments they have.

1.2.5. Administrator

Administrators must have their account either created from the start or by other administrators. Upon signing into the system, the administrator will have access to two tabs in the header, these tabs are "Users" and "Logs." In the users tab an administrator will be able to see all the users on the system and their user id. From here an administrator will be able to delete or create new users. In the logs tab an administrator will be able to view logs of every action that occurs within the system. This is to manage security and to track what changes are made and when those changes are made.

1.3. Data Flow Diagram

Design Map shows the over-all flow of the information in a clean flow chart.



2. Design Considerations

2.1. Assumptions

- Multiple insurance providers are not present and assume that the hospital is tied up with a single insurance provider (X) and if the patient wishes to take a policy, then they must take from this X only.
- Types of Charges available are:
    - Appointment cost
    - Lab test cost
- We are not considering hospital inventory (syringes, instruments and so on) and materials management of medical materials(pharmaceuticals).
- We won't be considering medical equipment monitoring such as defective or redundant medical equipment's.

2.2. Constraints

None that we are aware of.

2.3. System Environment

The Secure Hospital System in this project is developed using different technologies. ReactJS is used for the frontend development and Spring Boot for the backend. ReactJS has many advantages which is the key for choosing it. It has the concept of reusable components, well defined JavaScript library and ease of handling different dependencies. JavaScript Extension is used for allowing HTML syntax to render subcomponents. The Virtual DOM concept of ReactJS resolved the issue of slow performance due to frequent updates of the DOM. Instead of modifying the DOM directly, virtual DOM is first updated which is then turned into DOM by react. ReactJS code is easily testable by various tools. ReactJS is known for being an SEO friendly language as it supports easy movement of different search engines.

Spring Boot at the backend is a framework widely used to create stand-alone applications. It uses the POJO model which means the plain old java project which is not restricted and doesn't need any class path. Dependency is handled well using this framework. All the required dependencies are packaged to avoid any dependency issues. The main advantage of spring boot is its automatic configuration. According to the jars included, Sprint Boot will automatically align our application with them. Setup for the server can be eliminated by using Spring Boot as it includes the web server. Simple setup and ease of development makes it a better choice. There is no need for XML configuration which avoids the ambiguity and makes it easier for the developers.

Database SQL is used for storing the data of the Secure Hospital System in the form of tables. The information related to the registered users, hospital staff, doctors, patients, and the lab staff are stored in separate tables in the database. The information regarding the lab results of a patient, his/her insurance details and the appointments are stored in the form of tables and can be accessed in his/her profile on the website by drawing information from the database tables.

2.4. Design Methodologies

Design methodology defines the procedure for proceeding with the design process. The traditional design methodology includes the Waterfall process. It was widely used but had a disadvantage of not adapting to the changes that were being made during the development phase. Agile methodology is such a technique which adapts to the changing requirements and reduces the risk factor. For this project, Agile methodology is followed, and the four main values of the agile methodology are given utmost importance in the project which include:

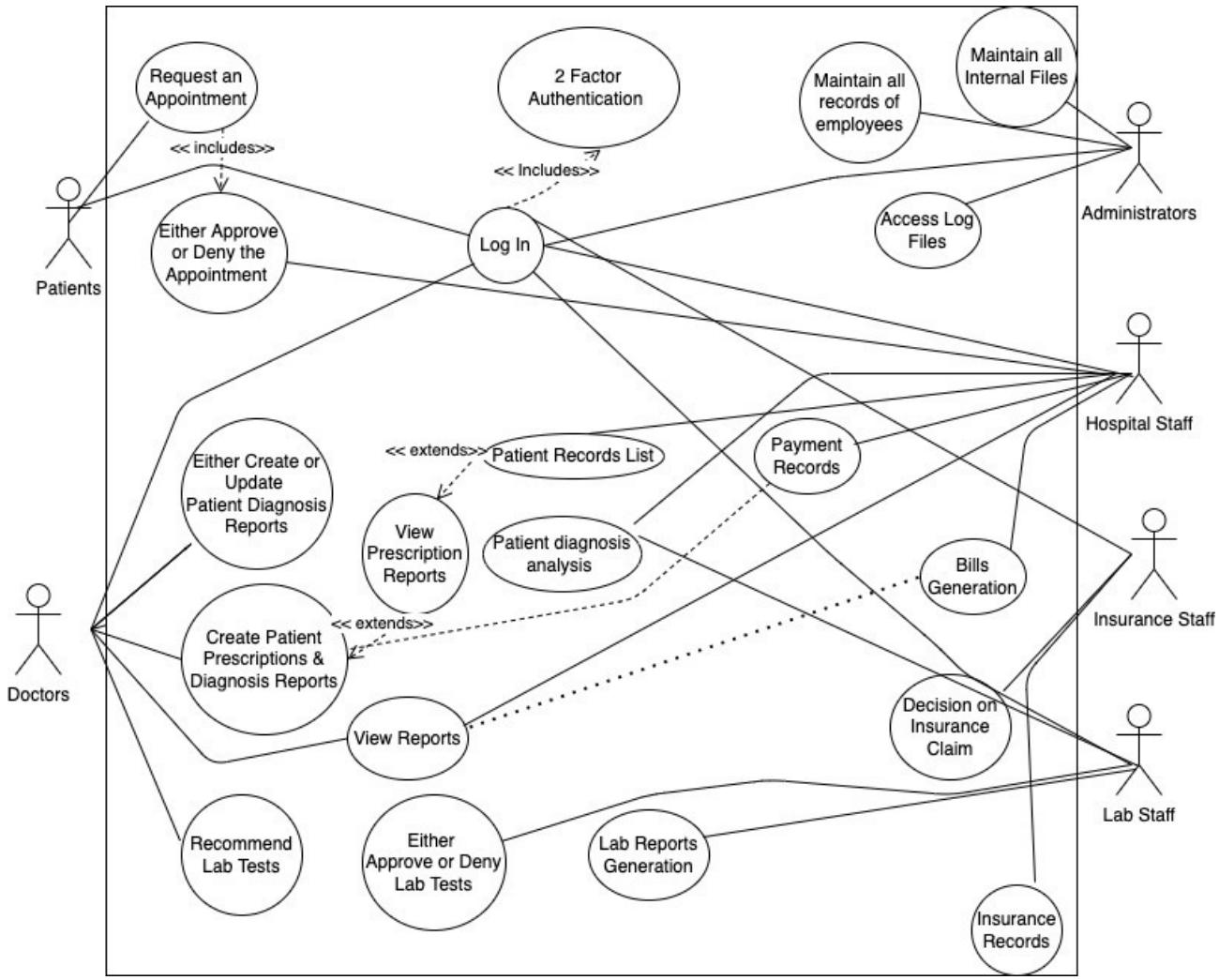
1. Interactions between the team are given priority over tools and processes.
2. Developing software over lengthy documentation.
3. Customers are treated as the top priority.
4. Welcoming the change at any point during the development.

An Agile project management tool, Jira, is used in the project for keeping track of the work being done. The team conducts five different meetings in the process of a sprint.

1. A Sprint planning meeting is conducted to discuss the work that needs to be dedicated to that sprint and analyze the time frame required for each user story. This is the meeting with the longest duration in the entire sprint and is very crucial.
2. Daily Standup meetings are of short duration and are straight to the point. This meeting includes three questions - What have been done today? What will I accomplish tomorrow? and are there any impediments in the process?
3. A Sprint Review meeting is conducted at the end of the sprint where the feedback is collected from the team. Few of the points from the feedback are considered and the additional work, if any, is added to the next sprint.
4. A Sprint Retrospective meeting is conducted to discuss what went wrong and what went right this sprint. Improvements to be made in the next sprint will also be discussed during this meeting.
5. Product backlog refinement meetings are conducted in between the sprints. Prioritizing the tasks is an important aspect of this meeting.

3. Architecture

3.1. Overview



3.2. Subsystems, components, and modules

We are using React JS For frontend and Spring Boot for backend, MYSQL for database and Block chain to store secure information like insurance and transactions records.

3.3. Integration of components and subsystems

Spring Boot can be combined with MySQL and the Blockchain. Because insurance and transaction data are private and must be protected, we are integrating Block Chain with Spring Boot. Except for insurance and transactions, Spring Boot with MYSQL is used for all other patient-related data.

3.4. Strategies

Using web3j library together with Spring Boot and Docker image of Ethereum Geth client allows implementing blockchain technology.

4. Database Schema

4.1. Database

As we have several relationships among the tables, we would like to choose MySQL as the DBMS for our project. One of the main reasons we chose this is because the hospital system mainly contains structured data and MySQL supports this out of the box. Moreover, MySQL also has direct spring-data-JPA support which will allow us to directly use ORM concepts into our database through the Spring Project.

4.2. Tables and fields

We have the following Entities:

1. User
2. Appointments
3. Diagnosis
4. Available Lab Tests
5. Lab Results
6. Insurance policies
7. Claims
8. Transactions

The respective fields of each table can be found in the ER diagrams below

4.3. Relationships

Relationships:

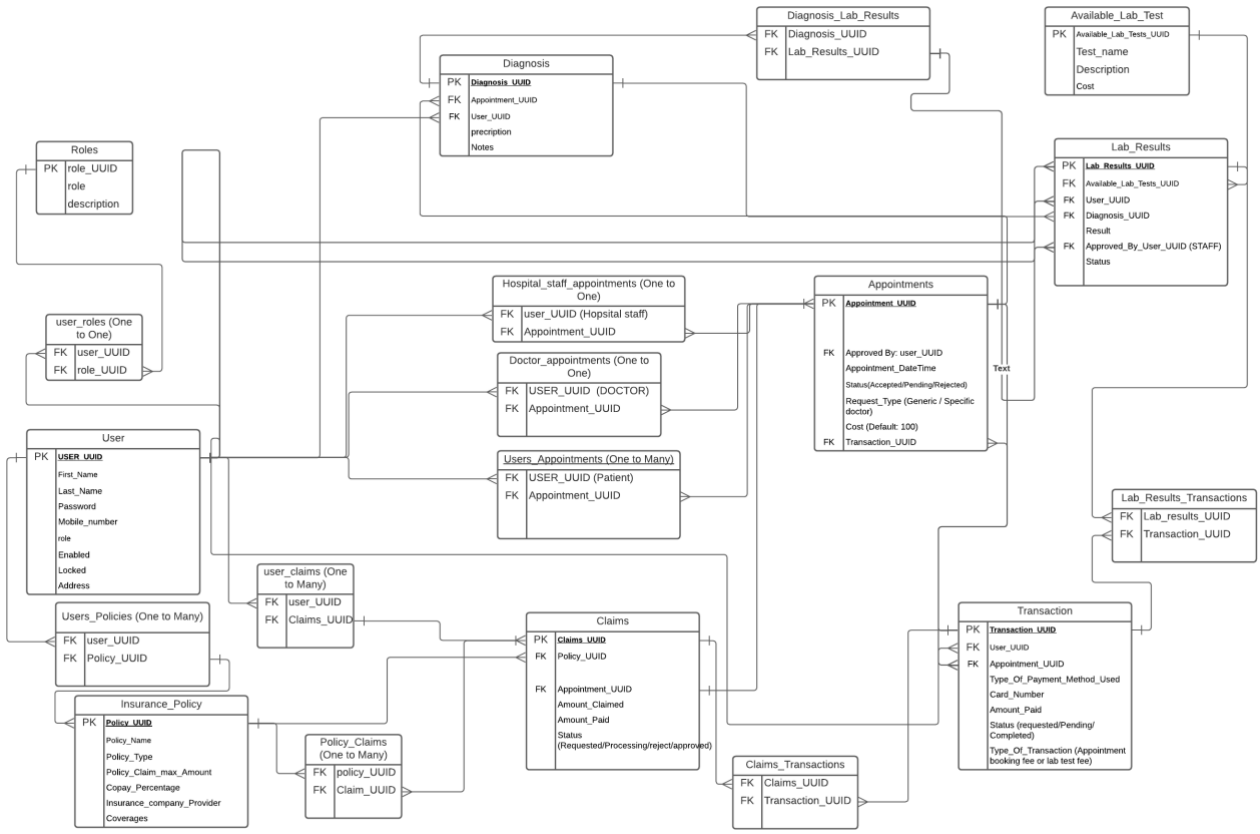
- Many to Many:
  - **User_roles**: Designed in such a way that this table holds **Many to Many relationship** between the roles and users table if required i.e. one user can have many roles and one role can be assigned to multiple users. [users_roles table holds this relation]
- One to Many:
  - **Appointment to transactions:** A single appointment can have multiple payments.
  - **Insurance_Policy to Claims:** A single policy can have mulitple claims.
  - **Claims to Transactions**: A single claims can have multiple transactions. If a transaction fails, it will be as a record in Blockchain with status failed in the transactions table.
  - **Lab_Test to Transactions**: A single lab test record have multiple transactions. If a transaction fails, it be as a record in Blockchain with status failed in the transactions table with.
  - **Users to Insurance_Policy**: One user can take multiple policies.

- **Diagnosis to Lab_Results**: A single diagnosis can require multiple lab tests [Diagnosis_Lab_Results].
- **User to Appointments**: A single user can make multiple appointments.

- One to One:
  - **User to Appointments**: we get two one to one relationships [user (doctor) to Appointment and user(hospital staff) to appointments]

More relationship comes into the picture when we implement all the functionalities.

4.4. ER diagram



5. User Interface

5.1. Header

This is the common header that will be part of all pages with minor changes based on the current page that is being displayed. It will always have the Secure *Hospital System* label at the top left side corner of the page.

5.2. Login Page

After the link is put in the browser, it will redirect to the login page which will have one simple window with text boxes for *Username* and *Password* to log in. There will also be a link for the sign-up page in the header which will be explained below.

5.3. Signup Page

Through the login page, there will be an option in the header to go to the sign-up page. This page will have various text boxes to take the user's details such as Name, Date of Birth, Email, etc. By default, the users will be given the patient role and it will be the job of an admin to give the correct role afterwards.

5.4. Employee UI

5.4.1. Administrators

Administrators have the most power in the application. They have the option to create, view, modify and delete any of the other employee records as well as have complete access to all the internal files.

They will have the option to authorize any kind of transactions such as insurance or doctor appointments, etc. Admins will be the only set of users that will have access to system log files which can’t be accessed by anyone else.

5.4.2. Hospital Staff

Hospital Staff will have the option to approve doctor appointments, view patient records and diagnosis and update them. They can also view prescriptions, lab test reports, create transaction requests, receipts, and bills.

5.4.3. Doctors (Non-Employee)

Doctors can view and update patient records, create prescriptions, recommend lab tests and check the lab test reports.

5.4.4. Lab Staff

Lab staff can create, update, and delete lab test reports. They can also view patient diagnosis and approve/deny the lab tests requests.

5.4.5. Insurance Staff

Insurance Staff can view, review, and validate the claims received from patients. They can also create new insurance policies and records.
They can approve/deny the claim requests and authorize fund transfer after given approval.

5.4.6. Patient

Patients can request for doctor appointments and can only view their records, prescriptions, diagnosis, lab test reports, insurance claims and payment receipts.

They can only update their personal information if needed.

All the options will be given in one column on the left side, clicking on them will open the UI on the main window to do the required operations.

6. Project Timelines

Few important points to be noted:

- We are handling various individual and independent functionalities of the roles parallelly.
- We are in the process of implementing common generic functionalities which are common to multiple roles to reduce the work load.

Monthly view of all the deliverables till the tentative finish date of the project:

Weekly view of all the deliverables till the tentative finish date of the project: