EASY HEALTH APP

SOFTWARE REQUIREMENT SPECIFICATIONS DOCUMENT

Hemanth Reddy Ramala (Team Lead)
Jaswanth Reddy Gangula
Karthik Yogendra Bangera
Meghana Bhojaraj Joshi
Rama Teja Repaka
Rishit Reddy Muthyala

FALL 2017

INSTRUCTOR: EDMUND YU, Ph.D.

Table of Contents

Table of Figures

Revision History

Name	Date	Reason for Changes	Version
Team 10	10/16/2017	Initial Version	Version 1.0
Team 10	10/22/2017	Modification of Class, Database and Package diagrams.	Version 1.1
Team 10	1/12/2017	Addition of sequence and state diagrams	Version 1.2

1 Introduction

1.1 Purpose

The purpose of this software requirements specification document is to support the development of the EasyHealth Android application. The requirements will cover the full scope of the system and shall also support features beyond the first release of application.

1.2 Document Conventions

Requirements are described in this document with “shall” or “should” to depict the priority and importance for the requirement to be implemented in the application. Shall is used for requirements that must be implemented and should is used to indicate a goal. All requirements are written using a natural language.

1.3 Intended Audience and Reading Suggestions

This document is to be read by team member involved in development of ‘EasyHealth’ android app for complete understanding of the system requirements and specifications for successful completion of project. It’s to be read before the development of application begins and during the development and testing phase to validate that the requirements are met before submitting the application.

1.4 Product Scope

This document is intended for members of the Team 10. The development of this app would consolidate all the features required by any type of patient. This app would be the first consultant for every user who is a patient/patient’s relative. The six modules of this app that would have major implementation would be: login module, prescription tracker, hospital locator, blood donor and patient awareness. The app is a complete package of services that is one touch away on the user’s phone. It would revolutionize the healthcare marketplace and generate awareness among the users all over the world.

1.5 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2 Overall Description

2.1 Product Perspective

This application will be a new product developed for Android mobile devices. The "Easy Health"app would consolidate all the features required by any type of patient. This app would be the first consultant for every user who is a patient/patient's relative. The app would be divided into six modules that would be briefly stated in the below section.

2.2 Product Functions

The six modules of this app that would require have major functionality would be: login module, prescription tracker, hospital locator, blood donor and patient awareness.

- The app would have different types of users: patients/patient's relatives, doctors, nurses, and blood donors.
- The login page would require the user to check against the option of user type.
- For a doctor or a nurse, the user would need to specify details like the location of their clinic or hospital and their specialization and also experience. Whereas the patient/relative would need to provide details such as name, patient's name if the user is a relative, address, age.
- For the patient/relative profile; the user would have the options to fix appointments with the doctors of their preference, locate the most suitable hospital/clinic/blood bank in the vicinity and track medications prescribed to them. They would also be able to keep a track of their prescriptions.
- There would be features to find suitable blood donors. The donors profile would also require an additional recent health check-up status to validate their donor profile.
- Lastly, there would also be a feature based on patient awareness; this feature would consist of media and content related to latest and common diseases that a patient needs to be aware of. This feature would also consist of the precautionary measures that the user needs to be aware of.
- For the doctor and the nurse profiles, the user would be able to keep track of the all the patients they have attended/treated or have given consultation advice. Also, the doctors can keep a track of the medications they have prescribed to their patients.

3 System Architecture

This section details the main components and technicalities that guide the development of the EasyHealth application. These components are highly reused within the application and affect the whole system, from the storage of data to the views presented to clients.

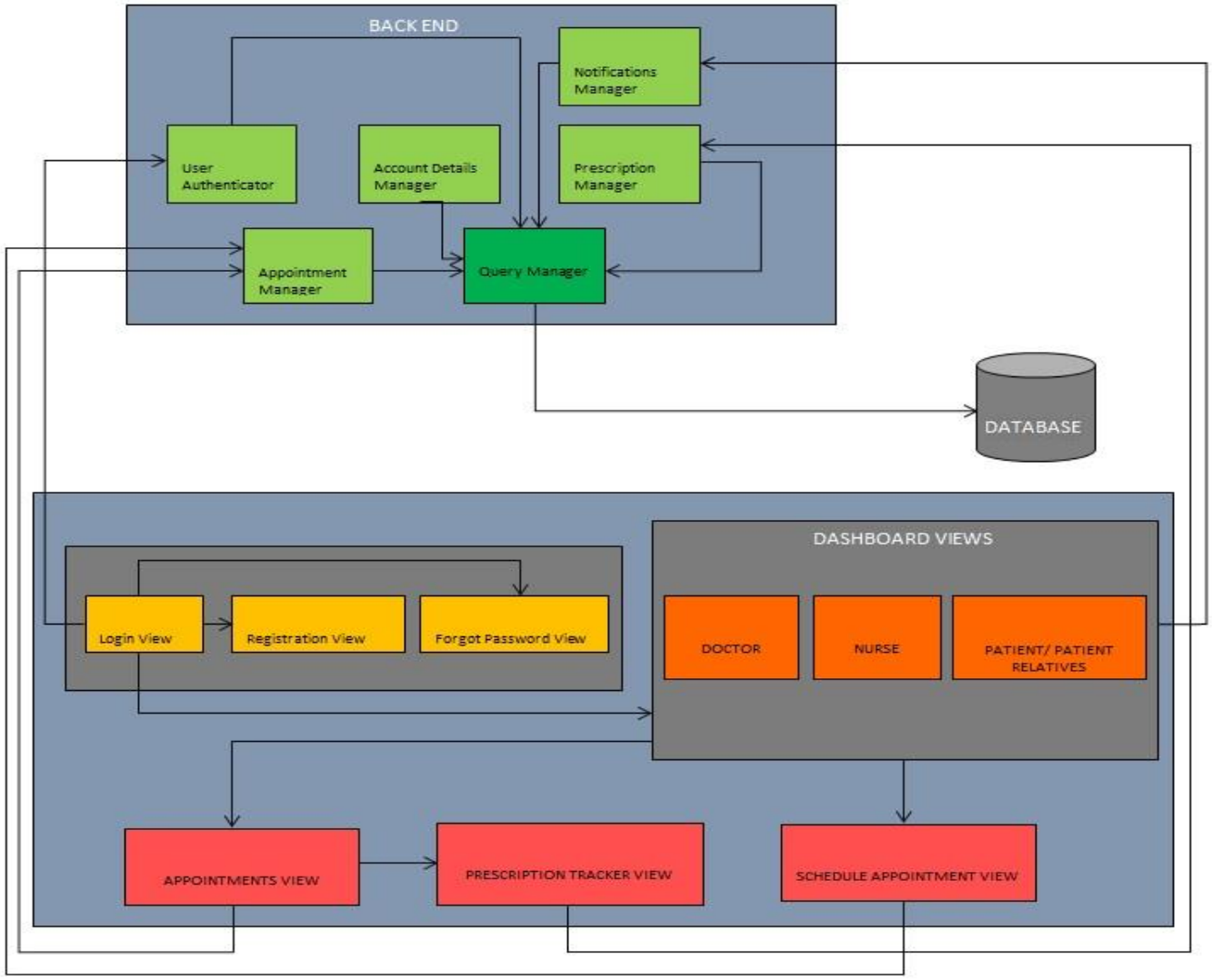

Figure 1: System Architecture

The system architecture of this application is based on the MVVM (Model-view viewmodel), which is a variant of the MVC model. It consists of 3 subsystems- backend, user interface and database.

Backend:

The following are the modules in the backend that are depicted in the system architecture diagram:

- The User Authenticator module authenticates the user details from the database using the query manager.
- The Appointment Manager handles appointment related functionalities such as fetching and adding appointments to the database.
- The Account Details Manager fetches the user details for the dashboard view once the user details are validated.
- The Notifications manager would fetch and add notifications to the database. The fetched notifications are visible to the user in the dashboard view.
- The Prescription Manager allows users to fetch/update and add prescriptions to the database.
- The Query Manager module handles all the database operations that have to be performed by all the above modules.

Database:

- For this application, the team would utilize the Firebase as its database.
- The Firebase Realtime Database enables developers build rich, collaborative applications by allowing secure access to the database directly from client-side code.
- Data is persisted locally, and even while offline, realtime events continue to fire, giving the end user a responsive experience.
- Thus, it enables developers to build a great realtime experience that can serve millions of users without compromising on responsiveness.

User Interface:

The following are the modules in the user interface that are depicted in the system architecture diagram:

- The Login module contains the entry view of our application, i.e. the Login view. It also has the registration view if the user is not signed up in the database. It also contains the forgot password view; in case the user has forgotten his/her password.
- The Dashboard module views for 3 user types- Doctor, Nurse and Patient. The views are inflated based on the successful login of the user type.
- The Appointment view allows users to view the pending and attended appointments.
- The Prescription Tracker view allows the users to view prescriptions for the appointments that have taken place.

- The Schedule Appointment view used for the patient to create an appointment with the doctor or nurse of their preferred choice.

4 User Requirements Definition

- **User Requirement 1 (UR1):** **Login Module**

Requirement: Users need to able to sign up and login into the App in order to make use of all the attributes and functionalities presented by the App.

Explanation: The App shall be able to let a user register firstly by providing their username, password, the type of user (Doctor, Patient, or Nurse). If it's a new user then they have to sign up or if it's an existing one then can log in directly. Furthermore, the signup process is absolutely free; as the user doesn't have to pay for the registration.

- **User Requirement 2 (UR2):** **Prescription Tracker**

Requirement: The users which included patients, doctors, or patient's relatives are provided with a prescription tracker feature which allows them to view and check the prescriptions history with attributes such as date, time, appointment details, place, doctor's name, symptoms of the disease and other medication details.

Explanation: One of the important features of the App includes the Prescription Tracker view through which the user (patient or the patient relative) would be able view the entire history of the prescriptions, the respective doctor who prescribed and the symptoms / disease for which it is prescribed. Also, the doctors will be able to view and check what they've prescribed to the respective user accordingly. One advantage here is; the users can view their prescriptions in detail from the type they've signed up i.e., the data is stored and recorded so that at any point of time it can retrieved later.

- **User Requirement 3 (UR3):** **Appointment Scheduler**

Requirement: The users will be able to schedule an appointment in their desired hospitals with the preferred doctor.

Explanation: Using the database we query all the hospitals in the user's location and list them in the App for the user. This lets users to select the hospital and once the hospital is

selected; all the doctors in that hospital are listed. From then on they can choose and schedule an appointment.

- **User Requirement 4 (UR4):** **Blood Donor**

Requirement: Blood Donors are the frequently needed every now and then. Users can search and find details about blood donors present in the city. A view in relation with the donors is present where all the details such as name, age, place of residence, contact number and their blood groups are listed.

Explanation: The user can click on one of the listed donors and can further contact the blood donors as per their requirements. The blood donor feature is one of the prominent features offered by the EasyHealth App.

- **User Requirement 5 (UR5):** **Patient Awareness**

Requirement: This feature would consist of a wide range of videos and documents related to the latest and common diseases; along with precautionary measures that a patient needs to be aware of.

Explanation: The user would have the privilege to view videos and read documents on trending healthcare news such as virus outbreaks. This would generate awareness among the users and would also enlighten them with the precautionary measures.

5 System Requirements Definition

- **System Requirement 1.1 – 1.6:**

1.1 The user must provide the login details.

1.2 The verification of the user login is done through the backend login interface.

1.3 It will connect to the database and checks if the user exists in the database.

1.4 If the user doesn't exist it should throw an error.

1.5 Registration page should be provided to the user for sign up.

1.6 Once all the details have been entered they should be saved to the database through the backend service.

- **System Requirement 2.1 – 2.4:**

2.1 For Patient/ Patient Relative the prescriptions should be tracked for the appointments.

2.2 The user interface should provide a detail section to view the prescriptions.

2.3 There should be a backend service to fetch the prescription details for specific appointment.

2.4 Database should hold the prescription details from which the backend service will query.

- **System Requirement 3.1 – 3.5:**

3.1 For Patient/ Patient Relative they should be able to schedule an appointment.

3.2 The user interface should provide a view to schedule an appointment.

3.3 There should backend service to create an appointment in the database.

3.4 The backend should also update the respective Doctor's/Nurse's appointment holders in the database.

3.5 Database should hold the appointment details for doctor, nurse and patient/patient-relative.

- **System Requirement 4.1 – 4.4:**

5.1 For Patient/ Patient Relative they should be able to view all the Blood donors.

5.2 The user interface should provide a view to list all the available blood donors and their details.

5.3 There should backend service to fetch all the blood donor list and details from the database.

5.4 Database should hold all the blood donor details so that they can be fetched later for users.

- **System Requirement 5.1 – 5.3:**

6.1 User Interface should provide a view to list all the awareness programs available from the database and view them upon selection.

6.2 There should be a backend service to retrieve all the awareness program details.

6.3 Database should hold the awareness program details for the backend service to query.

6 System Models

6.1 Use-Case Diagrams

6.1.1 Use-Case (Patient)

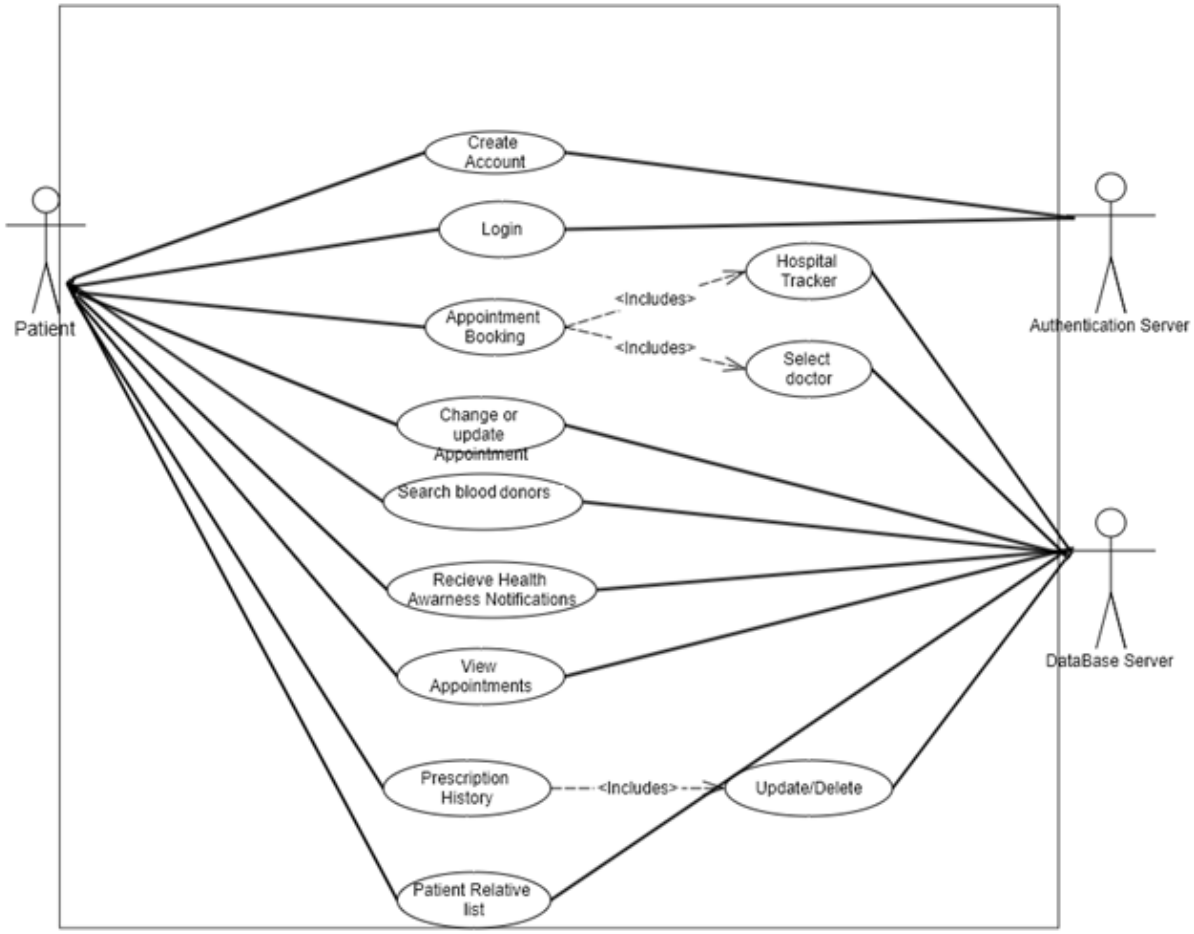

Figure 2: Use-Case (Patient)

For the patient, apart from the login features that are present for all the users; the additional features would be related to appointments, prescription history and access to the patient awareness feature. Firstly, the app would suggest the patient a list of hospitals/blood donors in the vicinity. After selecting the hospital, the app would enable the patient or their relative to book/update an appointment according to their convenience with the doctor of their preference. The patient can also view upcoming appointments as well as prescriptions that were listed after previous appointments. The patient would have

access to the bonus feature of patient awareness; to view videos and documents related to healthcare news.

6.1.2 Use-Case (Blood Donor)

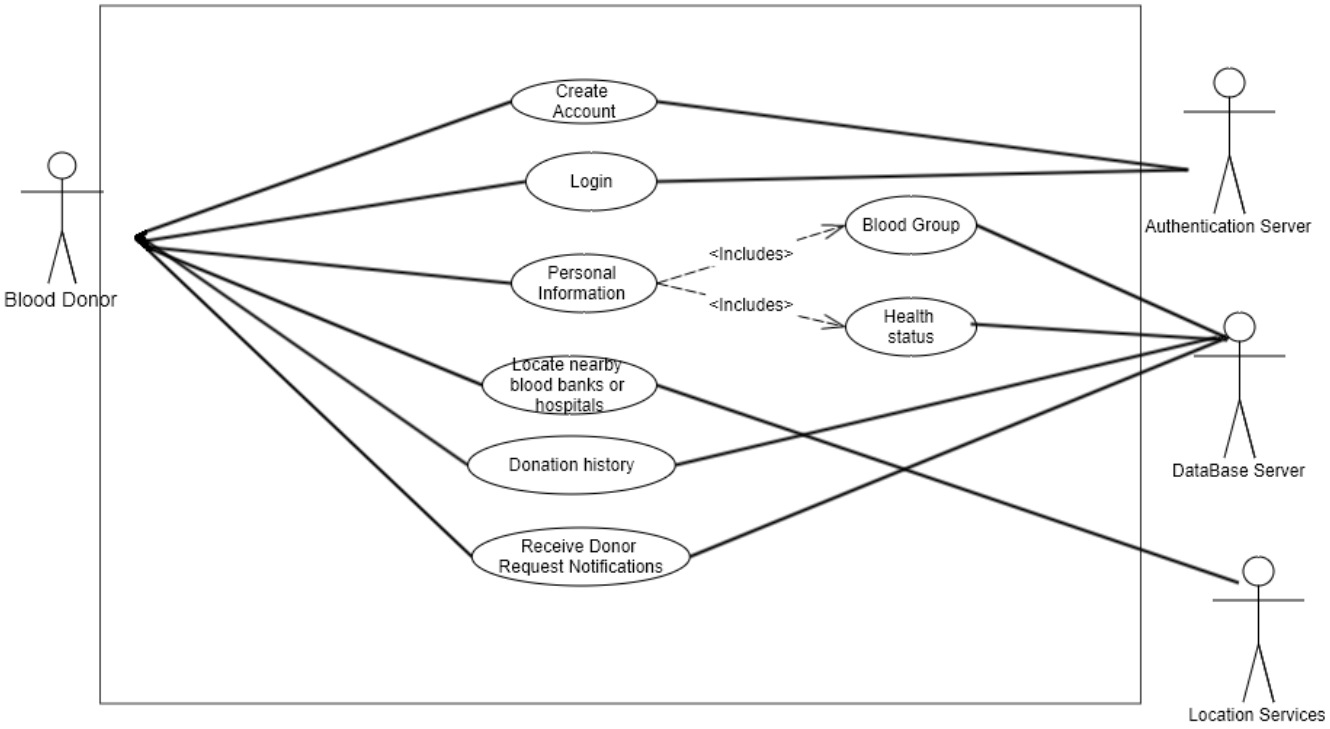

Figure 3: Use-Case (Blood Donor)

The blood donor can provide details such as blood group would he be donating and also a details of a recent health check-up which would validate that the donor could proceed with the donation in the future. The donor would also be able to locate nearby blood banks/hospitals. The donor would be able to view donation history and also receive notifications related donor requests.

6.1.3 Use-Case (Doctor)

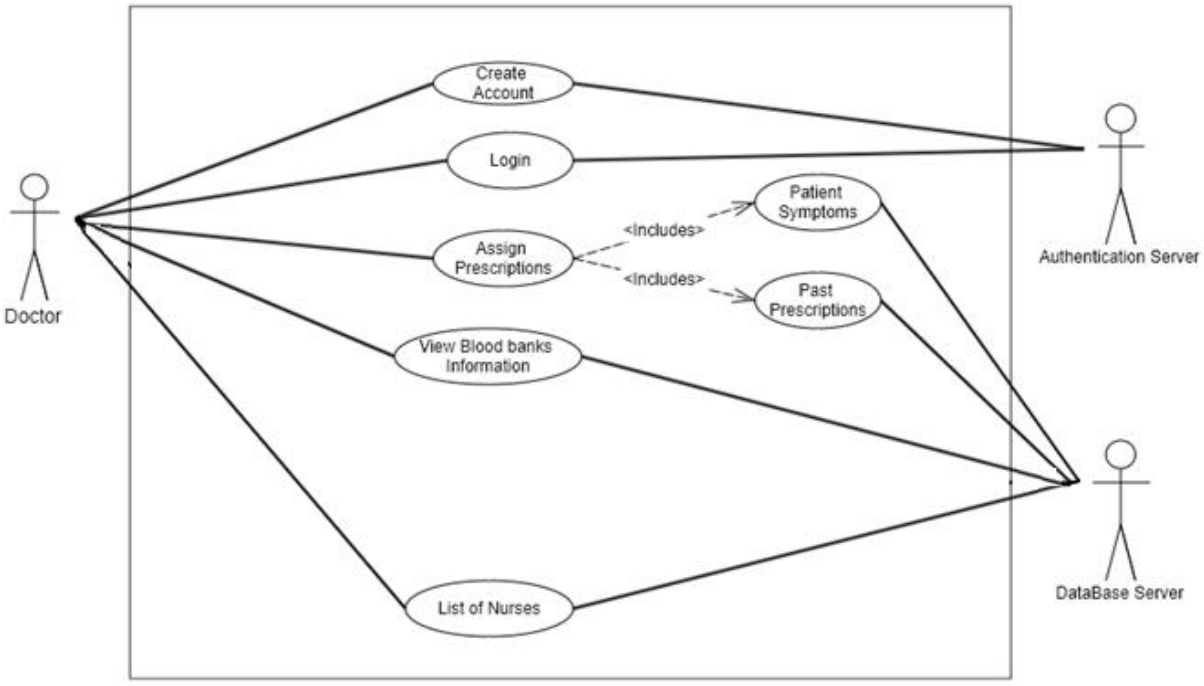

Figure 4: Use-Case (Doctor)

The doctor has the ability to assign prescriptions to the patients assigned to him/her based on the patient's symptoms and a check on the patient's past prescriptions. The doctor can also view nearby blood bank and blood donor details.

6.1.4 Use-Case (Nurse)

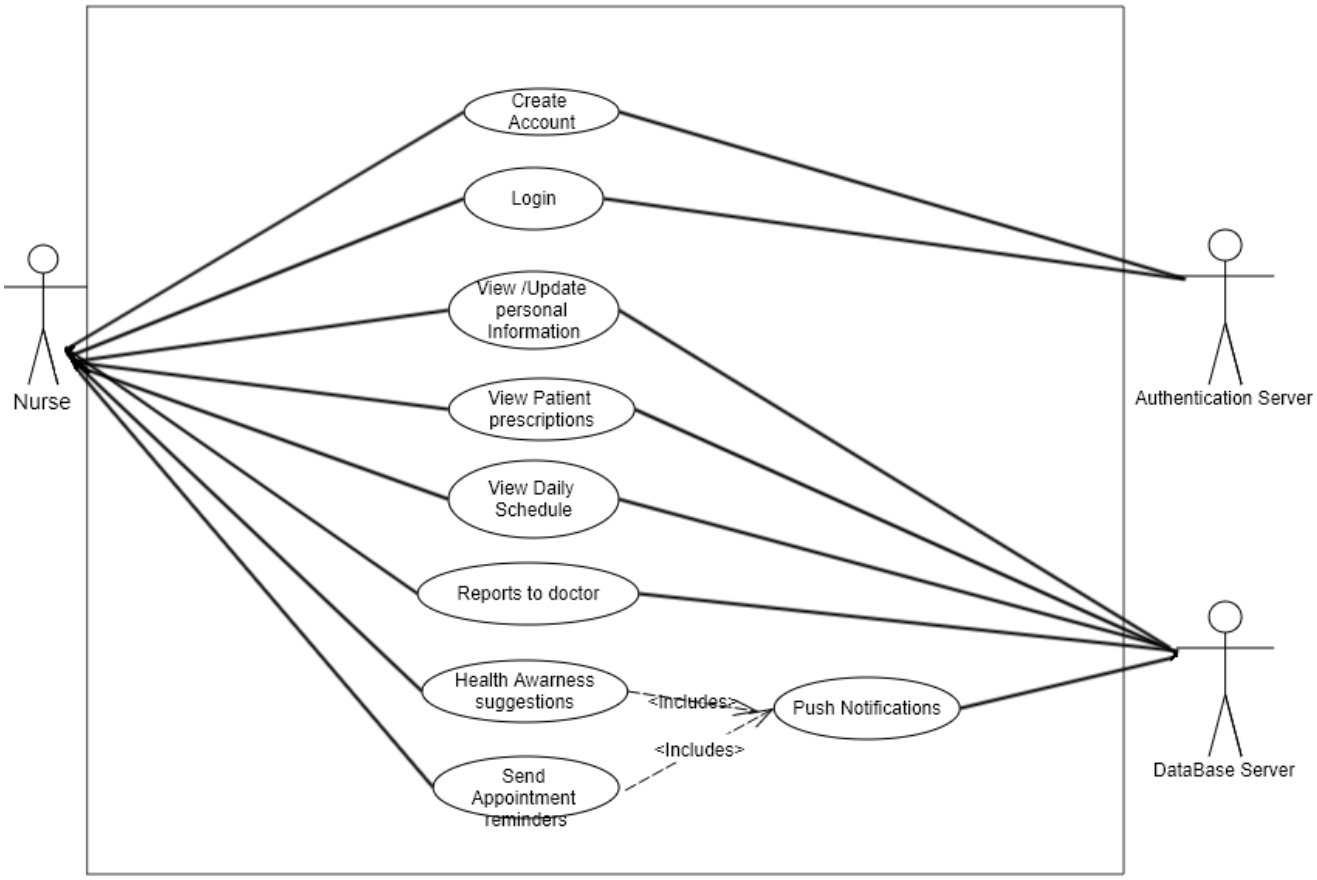

Figure 5: Use-Case (Nurse)

The nurse has features like to view prescriptions to the patients along with their daily schedule. They can view and update personal information. They would also have notifications related to appointment reminders.

6.2 Activity Diagrams

6.2.1 Activity (Patient)

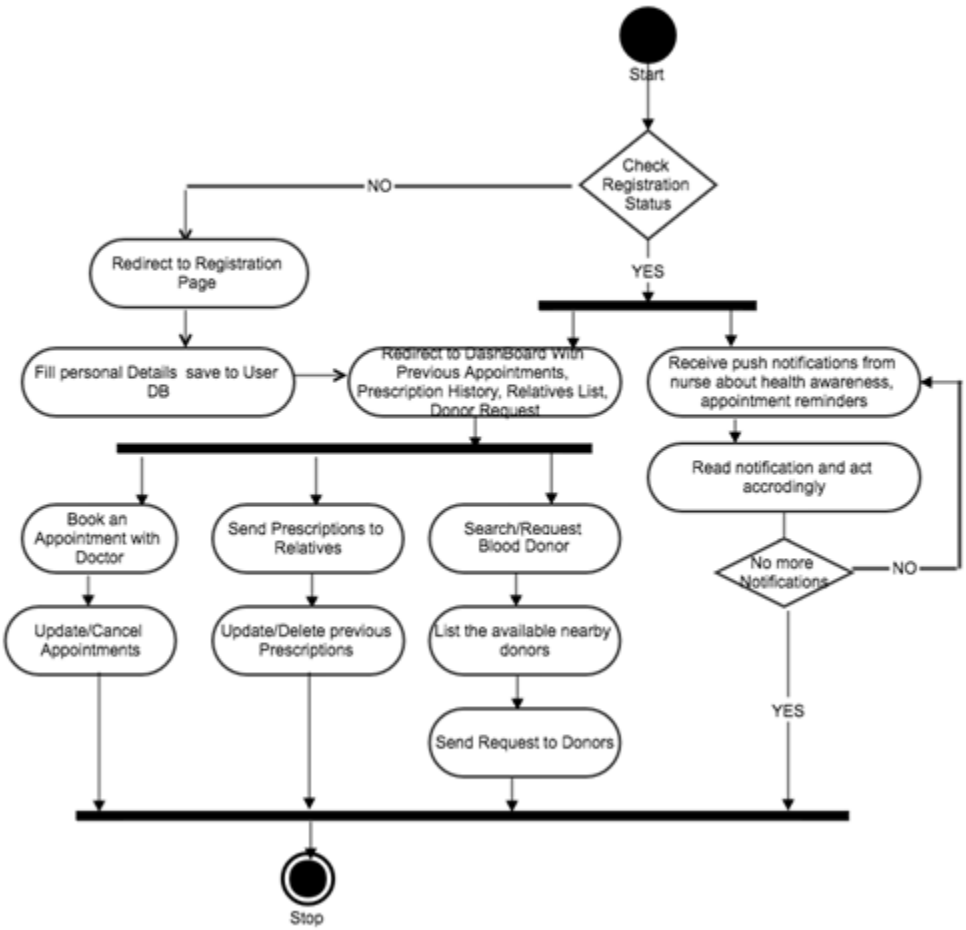

Figure 6: Activity Diagram (Patient)

6.2.2 Activity (Blood Donor)

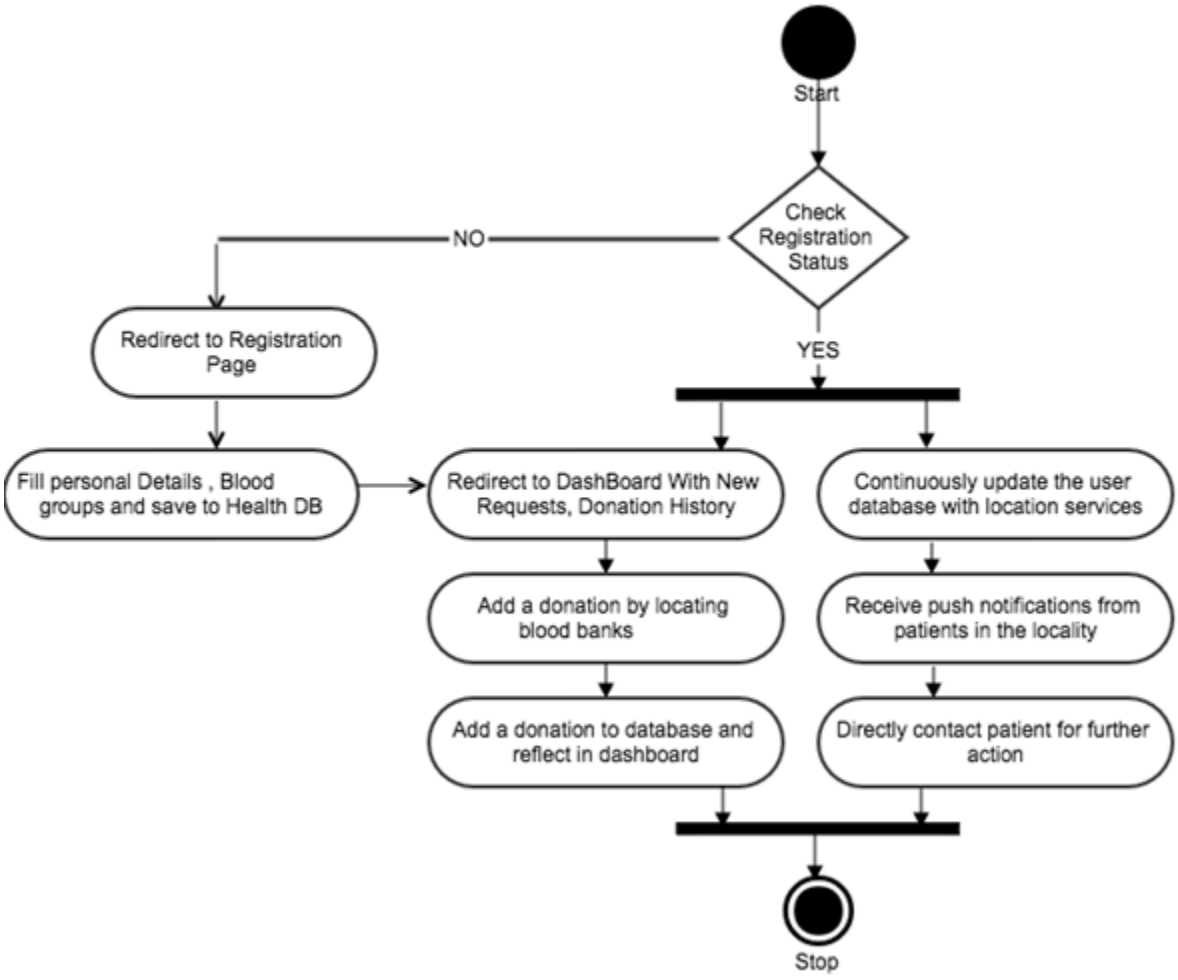

Figure 7: Activity Diagram (Blood Donor)

6.2.3 Activity (Doctor)

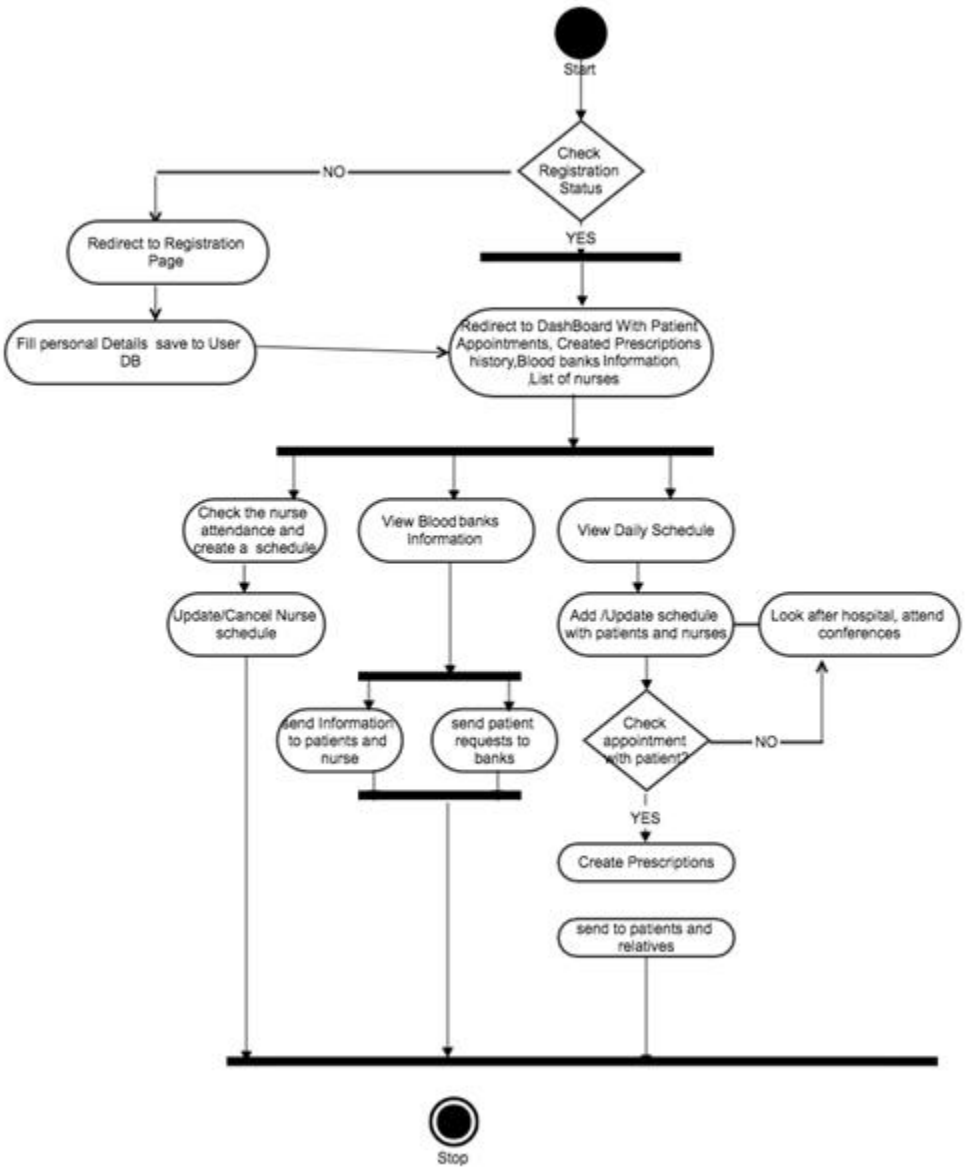

Figure 8: Activity Diagram (Doctor)

6.2.4 Activity (Nurse)

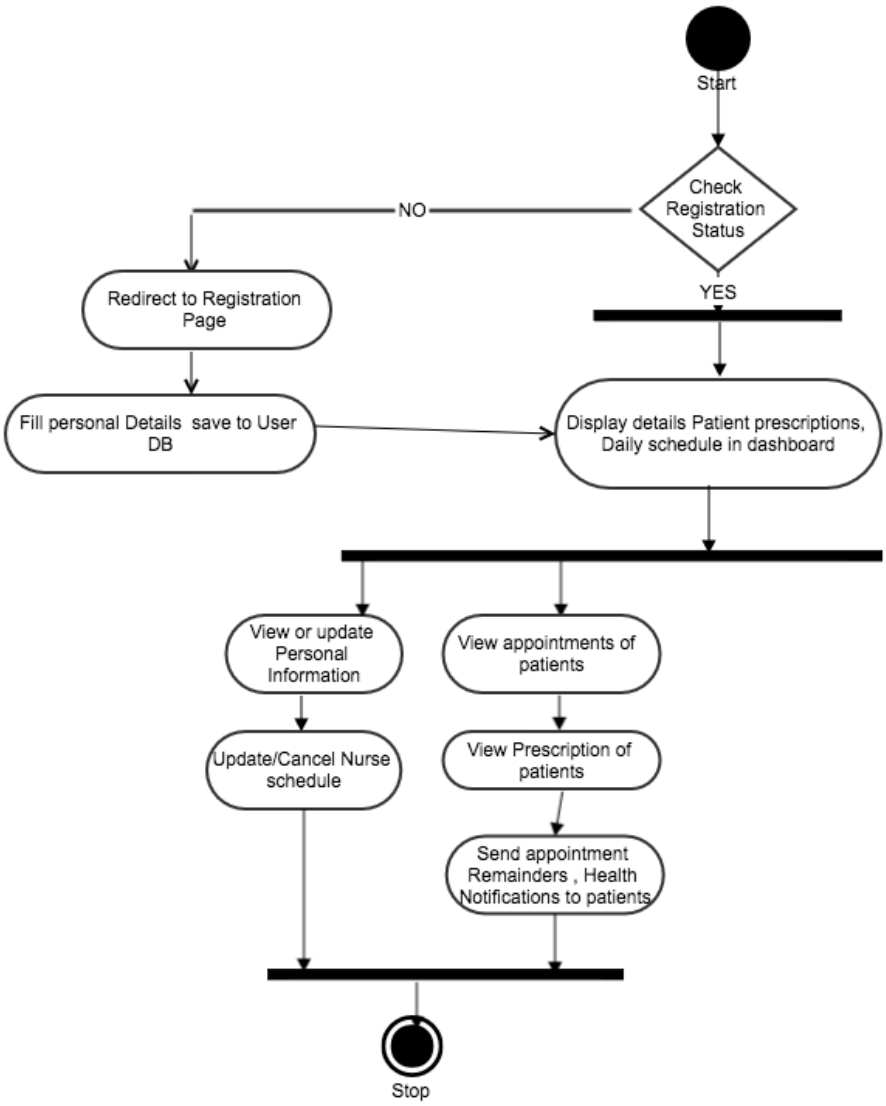

Figure 9: Activity Diagram (Nurse)

6.3 Package Diagram

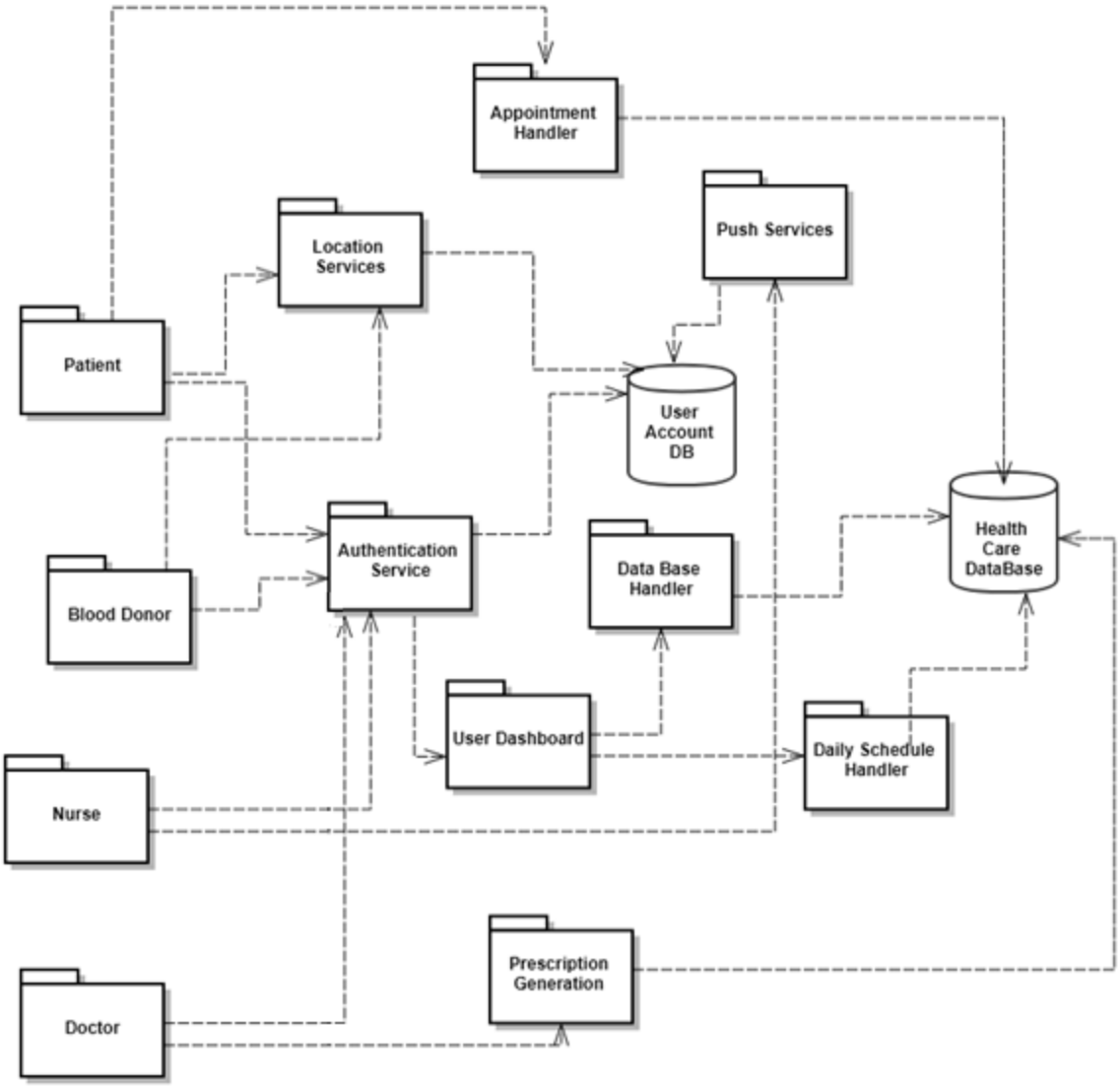

Figure 10: Package Diagram

6.4 Class Diagram

In the below diagram, all the class names ending with view are the controller classes for views/activities of our application. Rest of the classes constitute the core back-end business logic of our application.

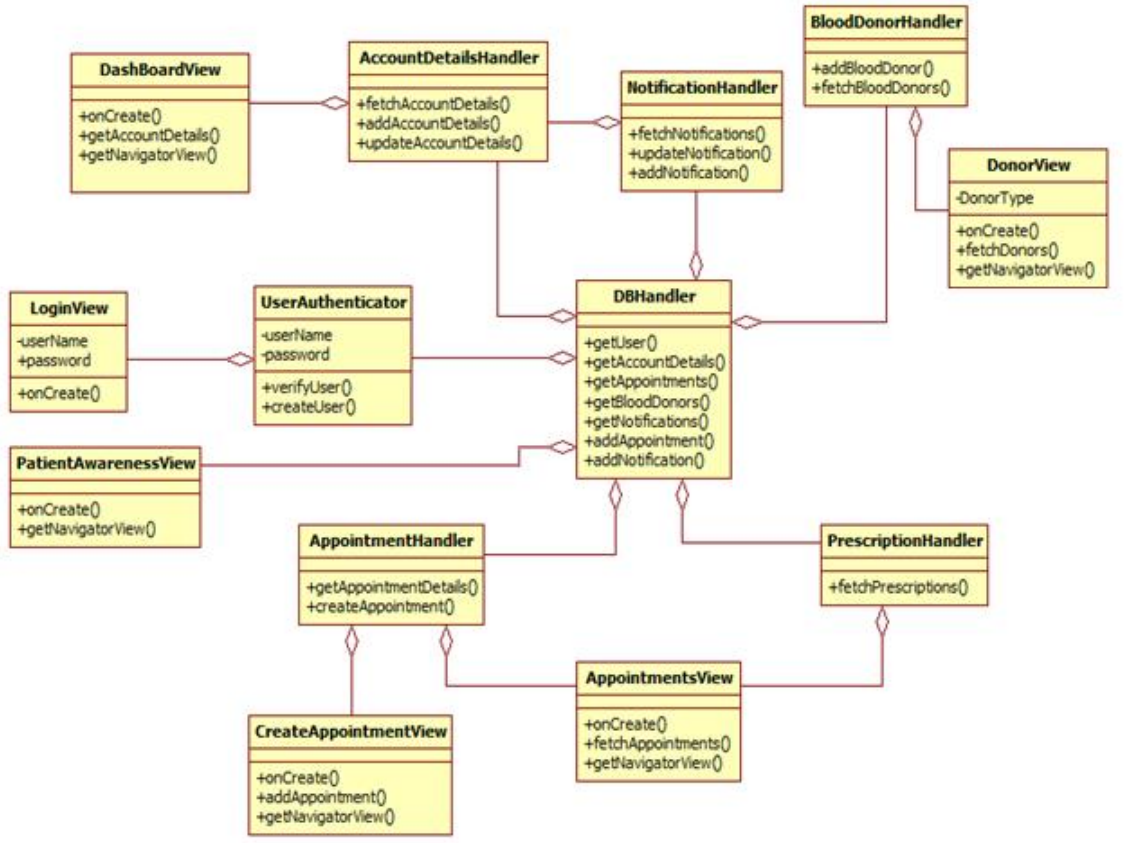

Figure 11: Class Diagram

6.5 Sequence Diagrams

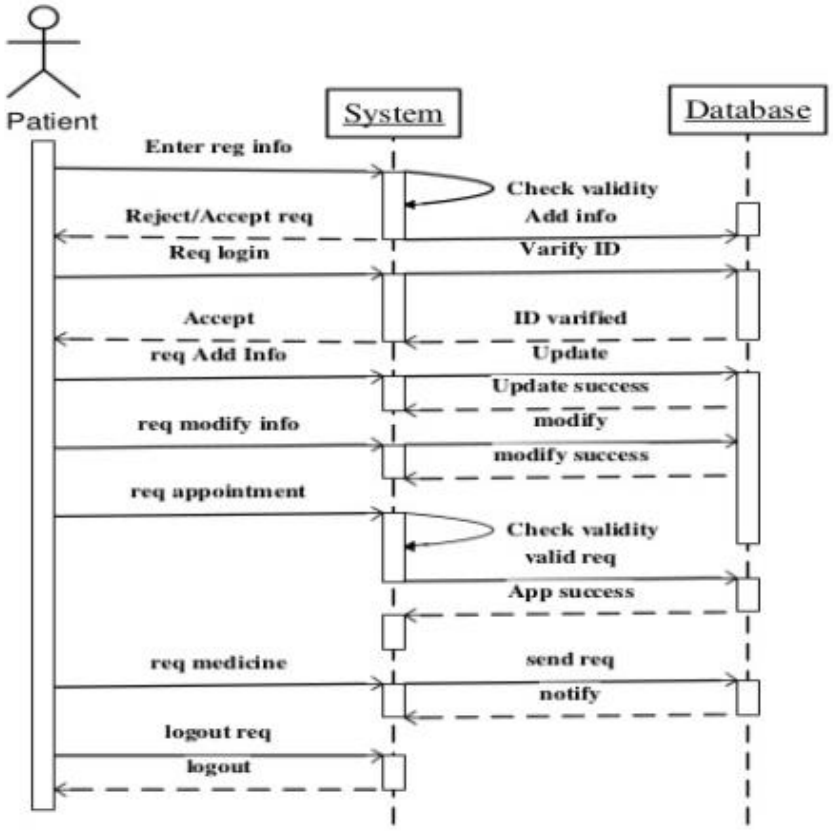

Figure 12: Patient Sequence Diagram

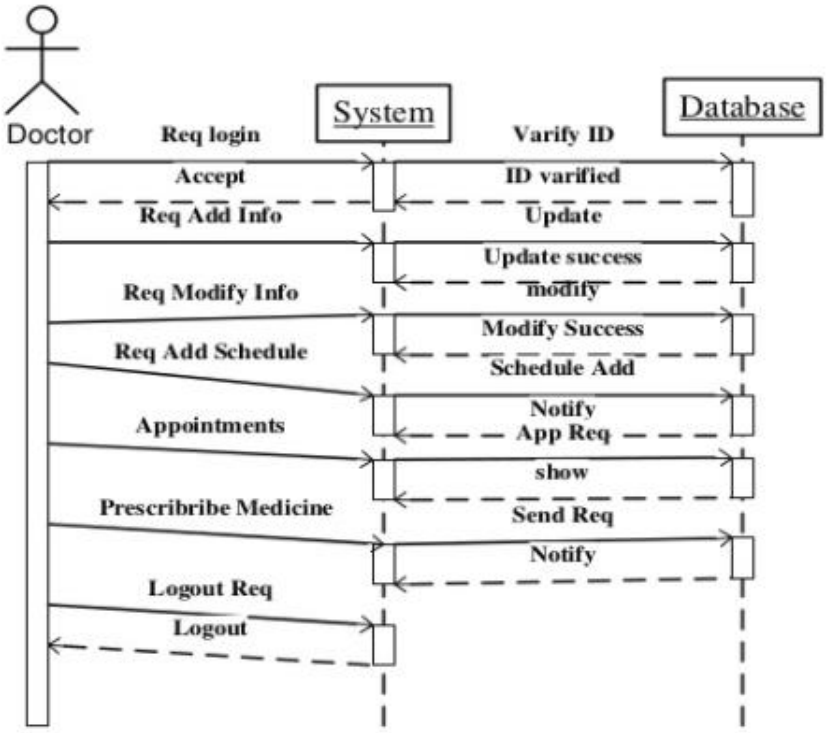

Figure 13: Doctor Sequence Diagram

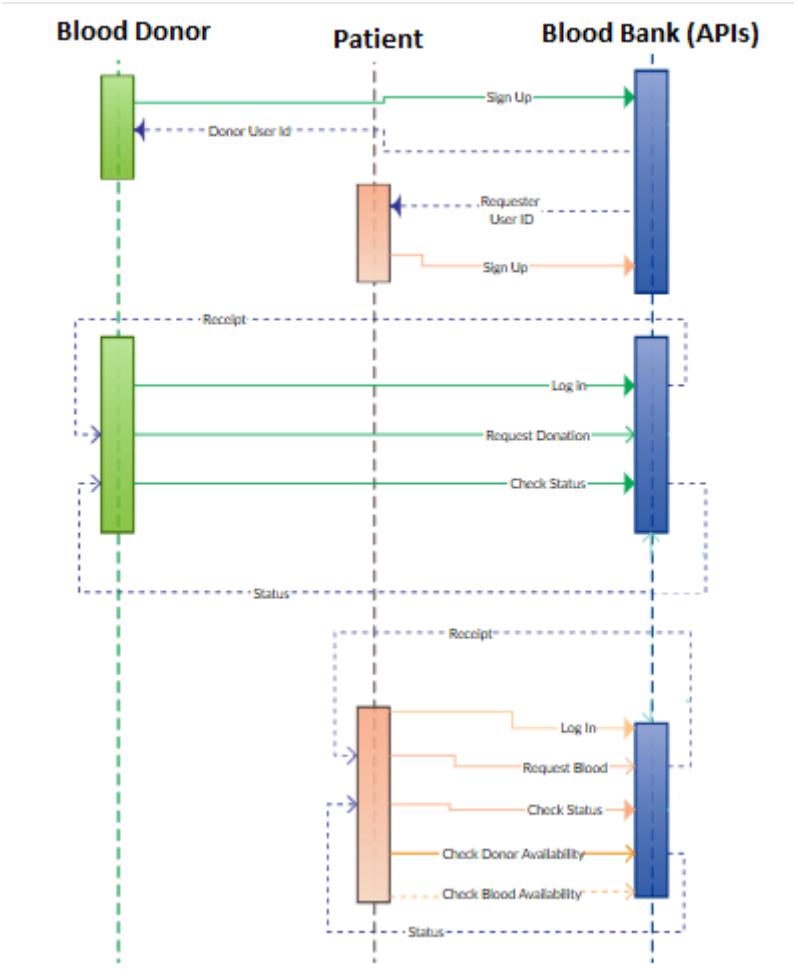

Figure 14: Blood Donor Sequence Diagram

6.6 State Diagrams

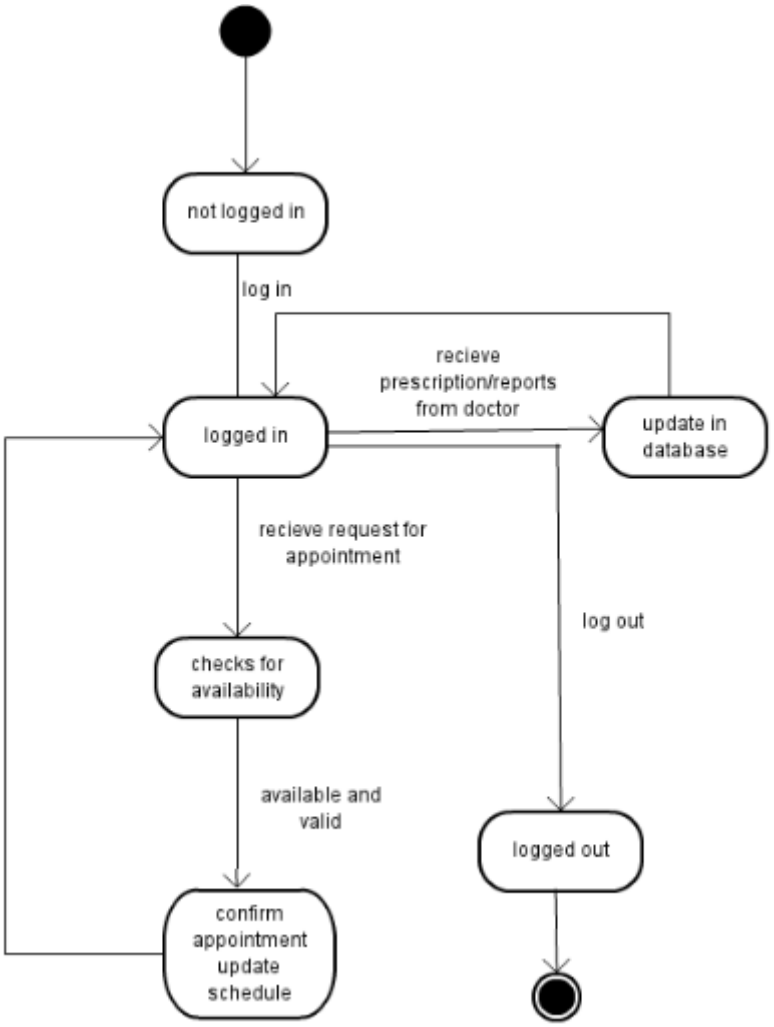

Figure 15: Patient State Diagram

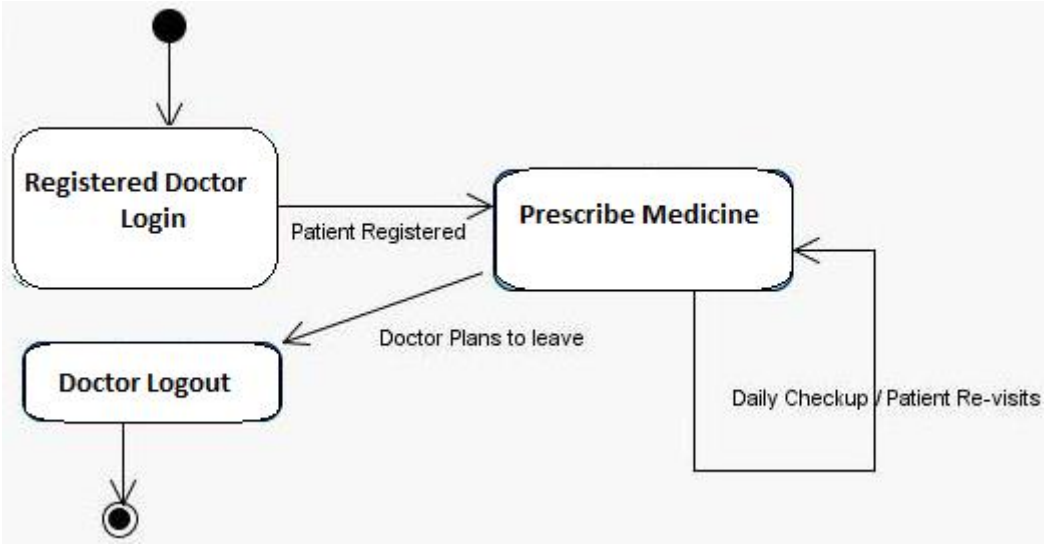

Figure 16: Doctor State Diagram

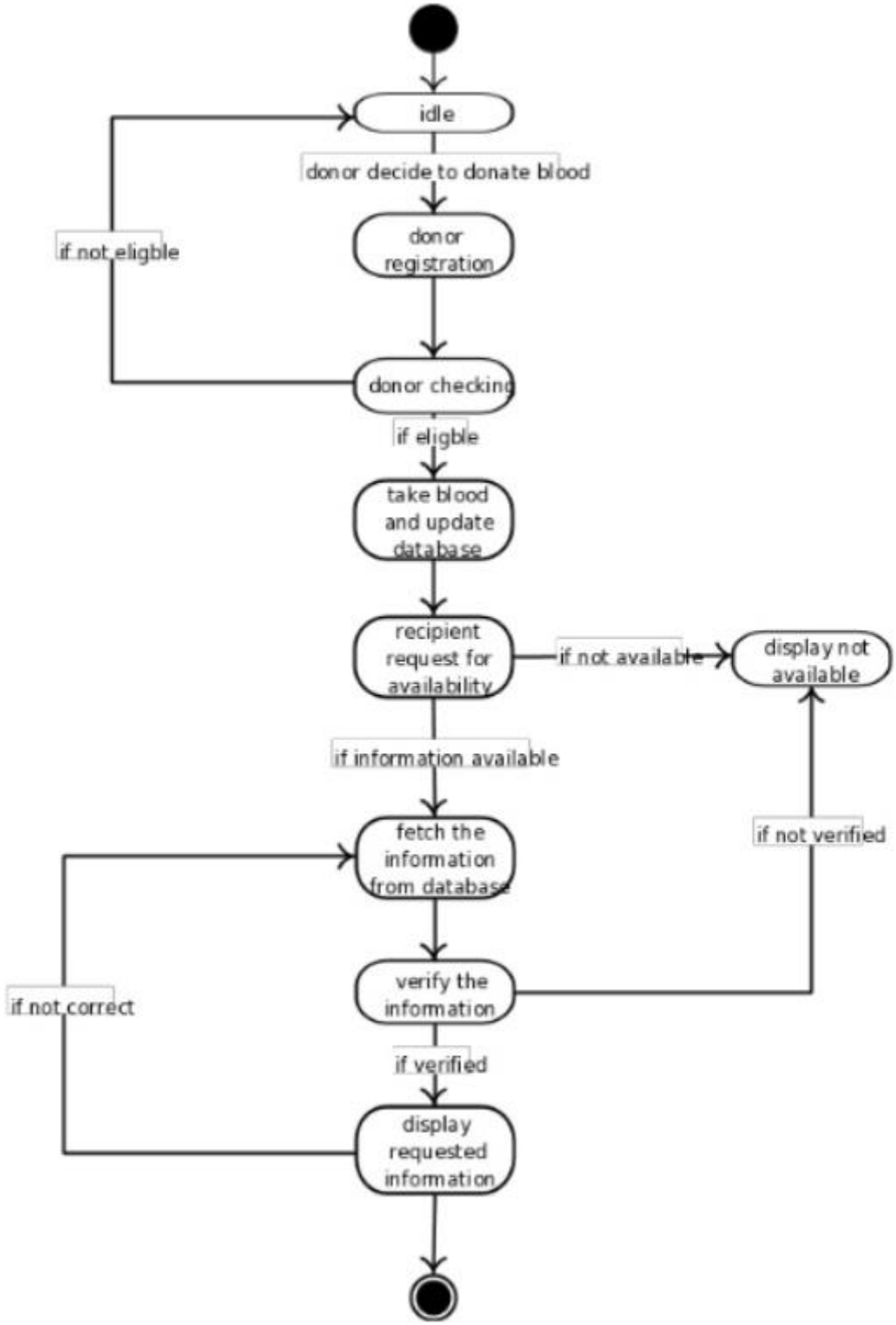

Figure 17: Blood Donor State Diagram

7 Non-Functional Requirements

- **Efficiency Requirements (ER):** These requirements describe the throughput, efficiency and response time of the system.

  ER-1: The user interaction and application response time should be less than two seconds.

  ER-2: The notifications for the user in the application should be processed in one second or less.

- **Availability Requirements (AR):** These requirements describe the degree to which system is up and running.

  AR-1: If the system is non-operational; it shall present the users with a notification to inform them regarding the next availability of the system.

  AR-2: The application shall be available to the users throughout the clock.

- **Usability Requirements (USR):** These requirements describe the ease with which the user will be able to operate and learn the system.

  USR-1: To increase the application usage by people; the user interface should be as intuitive as possible.

  USR-2: The application shall be easy to use for users above the age of 15years.

- **Safety Requirements (SAR):** These requirements describe the degree to which the application prevents harm to users of the system.

  SAR-1: The application shall block the user who misuses the hospital staff/donor contact details.

  SAR-2: The application shall not allow drugs to be prescribed to patients without validation.

- **Performance Requirements (PR):** These requirements describe how well the system performs under various circumstances.

  PR-1: The application shall not crash more than once while its being used by the user.

- Security Requirements (SER): These requirements describe the extent to which the system safeguards the sensitive information from intrusions.

  SER-1: The user sensitive data such as password shall be protected via encryption before storing them to the database.

8 Appendix A: Database Diagram

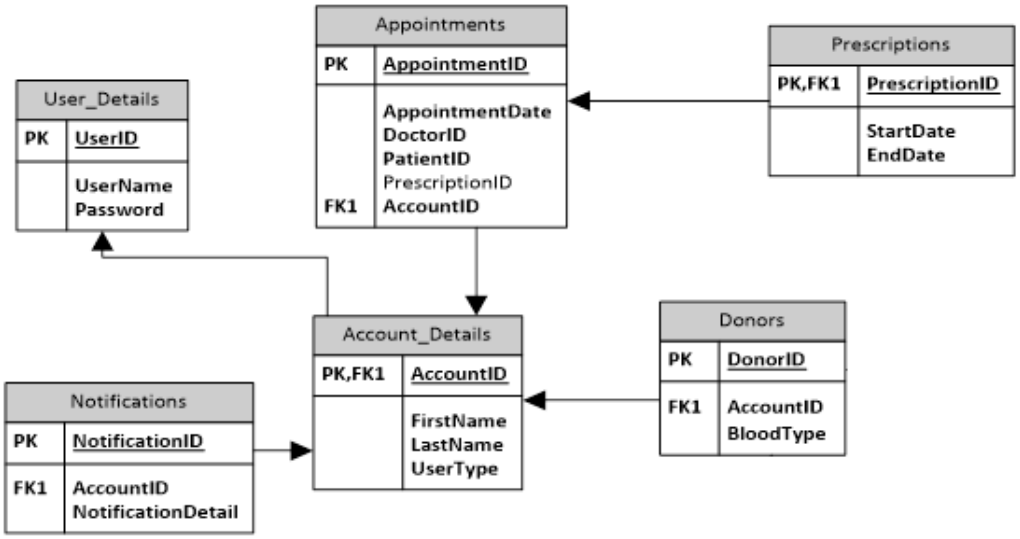

Figure 18: Database Diagram

9 Appendix B: Prototype Designs

Below are a few screenshots of the prototype design that the team intends to replicate during development.

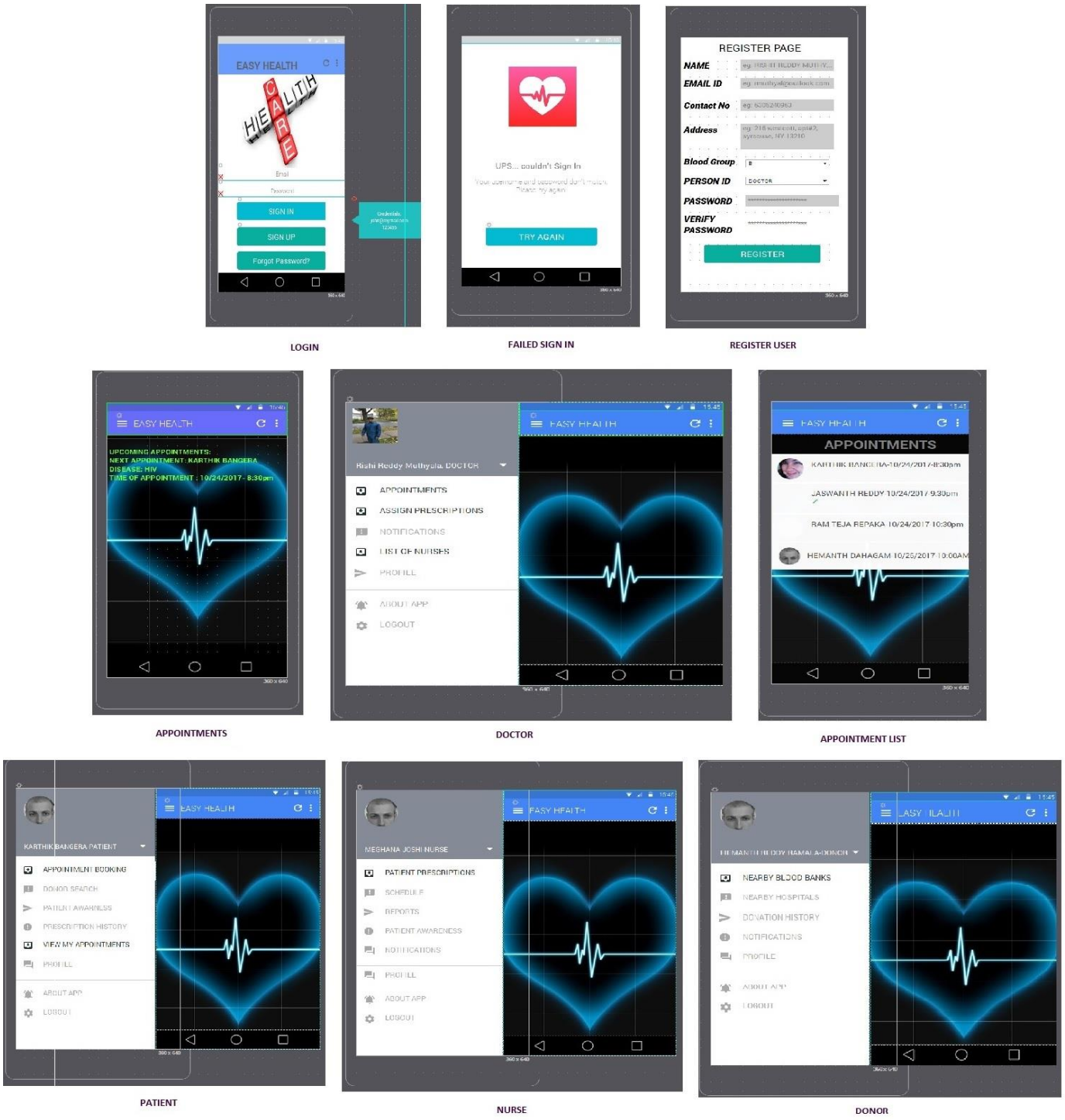

Figure 19: Sample Prototype Designs