# System Development Approach

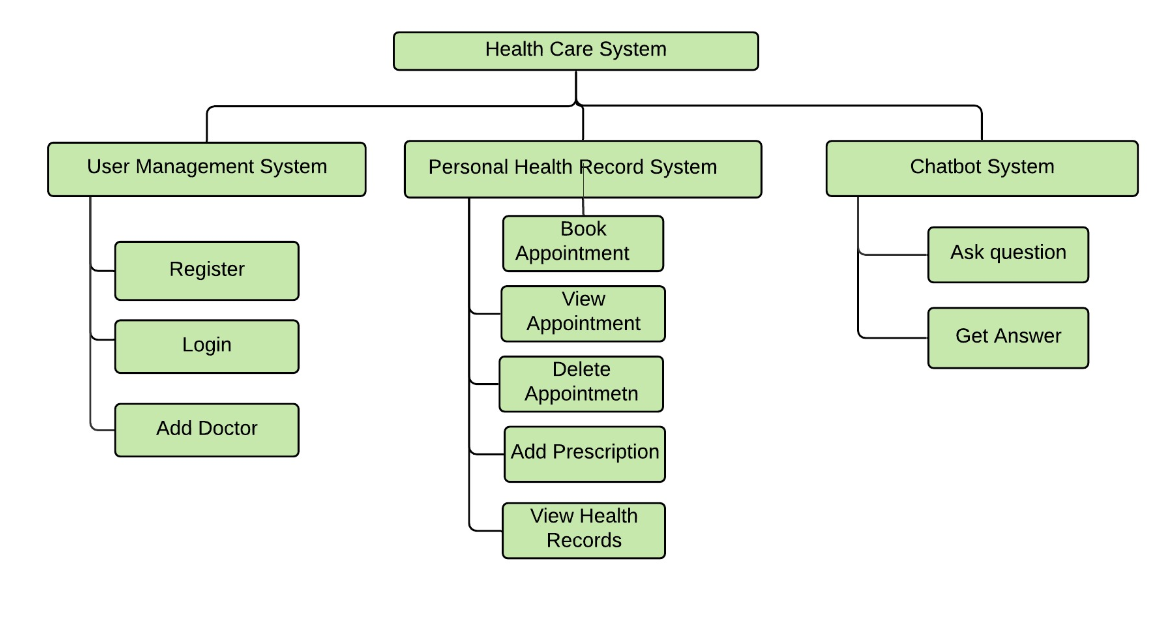
It can therefore be argued that a project succeeds when it is well managed. Development team or the manager of the development must well choose a software development methodology which is best suited to the project. Every methodology has its own method, advantage and disadvantage and exist for a purpose.

Scrum is one of the examples of agile software development methodology. Agile is the way that most software is developed in the 21st century. Repetition of development and testing at some organized intervals within the complete life cycle of the particular project is the key consideration of the agile software development method. The above method of the development is a parallel process with testing different from the waterfall method where development and testing are sequential. (Guru99, 2020)

Thus, there are several things to take into account while choosing a software development process, the most crucial of which is how to handle changes in end-user requirements. The primary advantage of Agile approaches is that modifications and communication techniques are followed in a clear-cut and efficient way. Frequent meetings with the supervisor are feasible now that the Agile system is in place.

For our project, the Agile Scrum framework would be the most suitable methodology. During the project's operating phase, requirements that were not found during the beginning phase can be resolved. Scrum is particularly adaptable to requirements changes. It can be used for projects involving small teams, like ours.

## **Functional Decomposition Diagram (FDD)**

****   
 Figure1: FDD of the System

## **System Requirement Specifications (SRS)**

This system requirement describes the software system that is being developed for the Health Care Chatbot System. Drawing on the department's and customers' transactions, it delineates the main functional, non-functional, and usability requirements.

**Functional Conditions**

These specifications define the functions the system must perform, such as how it must process outputs, react to inputs, and act in a specific manner under certain conditions.

Example:

A user should be able to log in to a system.

A system ought to enable users to add and remove packages and objects.

A user should be able to create bills in the system.

**Non-Functional Requirements**

These prerequisites are the non-functional services that the system makes use of to support enhanced system functionality. These specifications outline the limitations and characteristics of the system such as dependability, speed of occupation, response time, portability, storage, etc. Three categories can be distinguished from it:

**Usability**

These requirements guarantee that users will find the produced system easy to use and that it will fulfill all necessary objectives while operating flawlessly.

Legend

UMS: User Management System

PHMS: Personal Health Management System

CS: Chatbot System

F: Functional Requirement

NF: Non-functional Requirement

UR: Usability Requirement

### **SRS Table for User Management System (SubSystem 1)**

|  |  |  |
| --- | --- | --- |
| Requirement Code | Requirement Description | MoSCoW |
| UMS-F-01 | The system should allow normal user to sign up and login to the system. | Must have |
| UMS-F-02 | The system should allow user to reset the password. | Must have |
| UMS-UR-03 | System should suggest to add numeric  and special characters in password. | Must have |
| UMS-UR-04 | System should give the option of making the credentials visible while login. | Should have |
| UMS-NF-05 | System should hide the login credentials of the user. | Must Have |
| UMS-F-06 | The system should allow user to chat with CHATBOT system without logged in. | Must have |
| UMS-F-07 | System should allow doctor to login to the system. | Must Have |
| UMS-F-08 | System should allow admin to Add Doctor. | Must Have |
| UMS-F-09 | System should allow admin to delete doctor. | Must have |

A picture containing diagram

Description automatically generated  
  
  
   
 Figure 2: Sequence Diagram for User management System

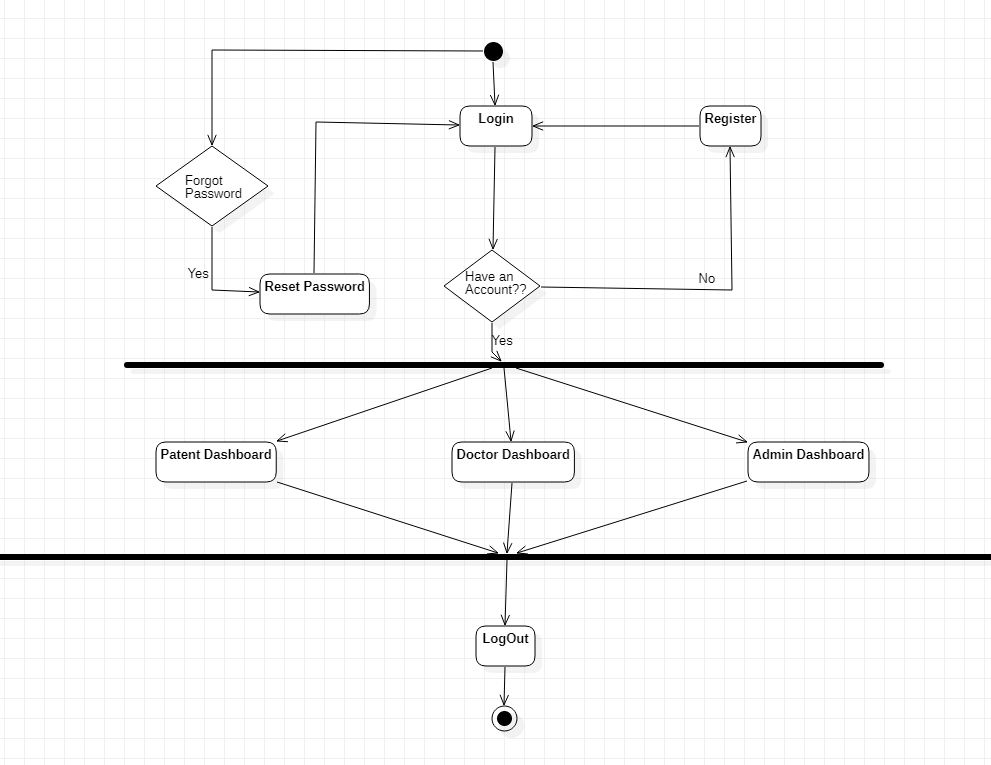


Figure 12: Activity Diagram of UMS

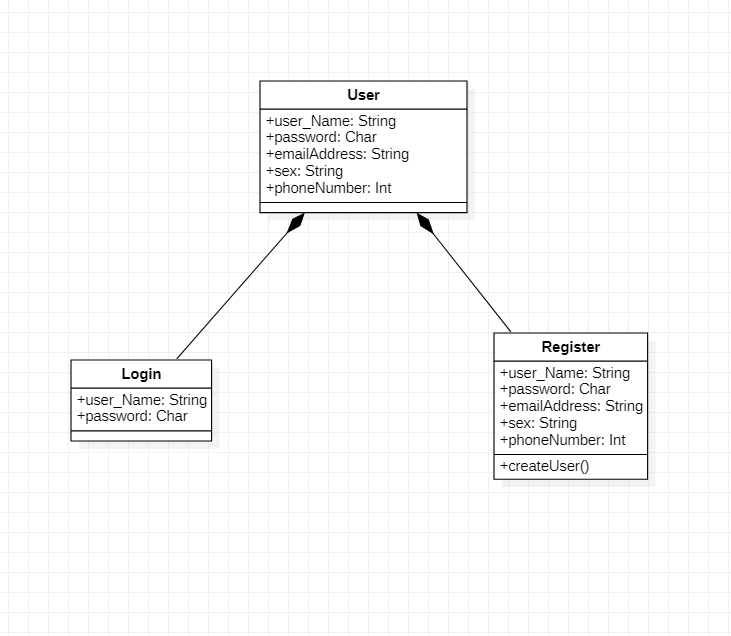
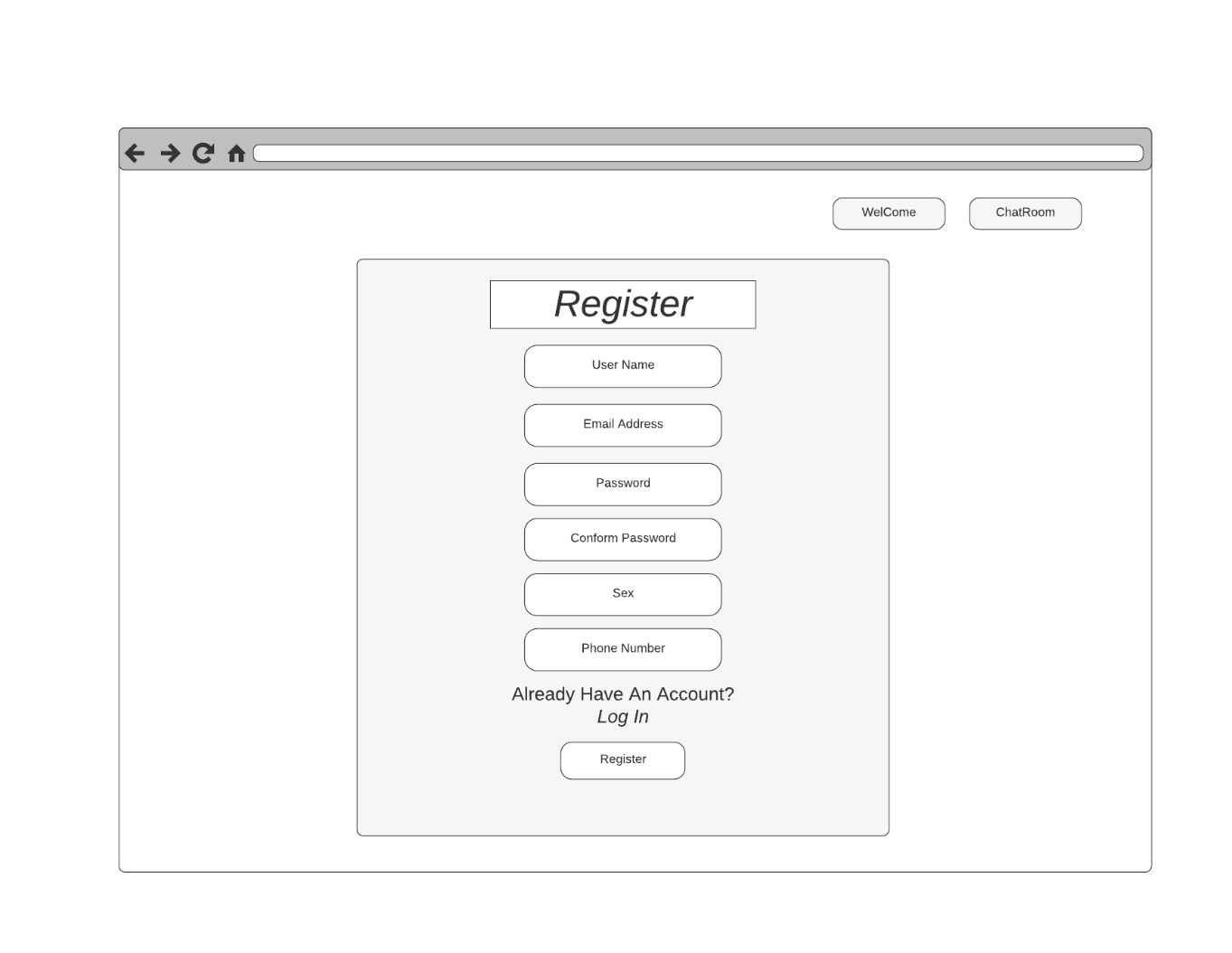
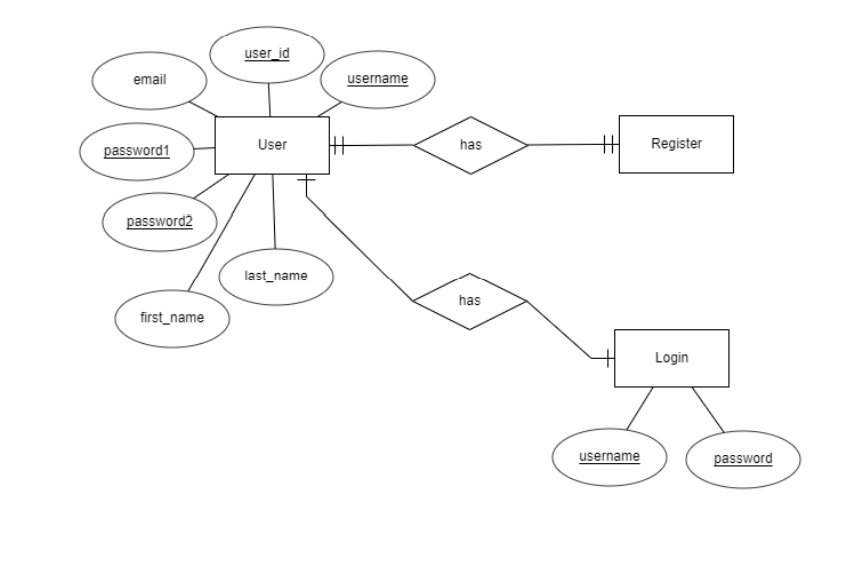


Figure 13: Class Diagram for UMS

  
 Figure 14: Wireframe of User Registration Form

Graphical user interface

Description automatically generated Figure 15: Wireframe of User Login System

  
 Figure 17: Detailed ER Diagram of UMS

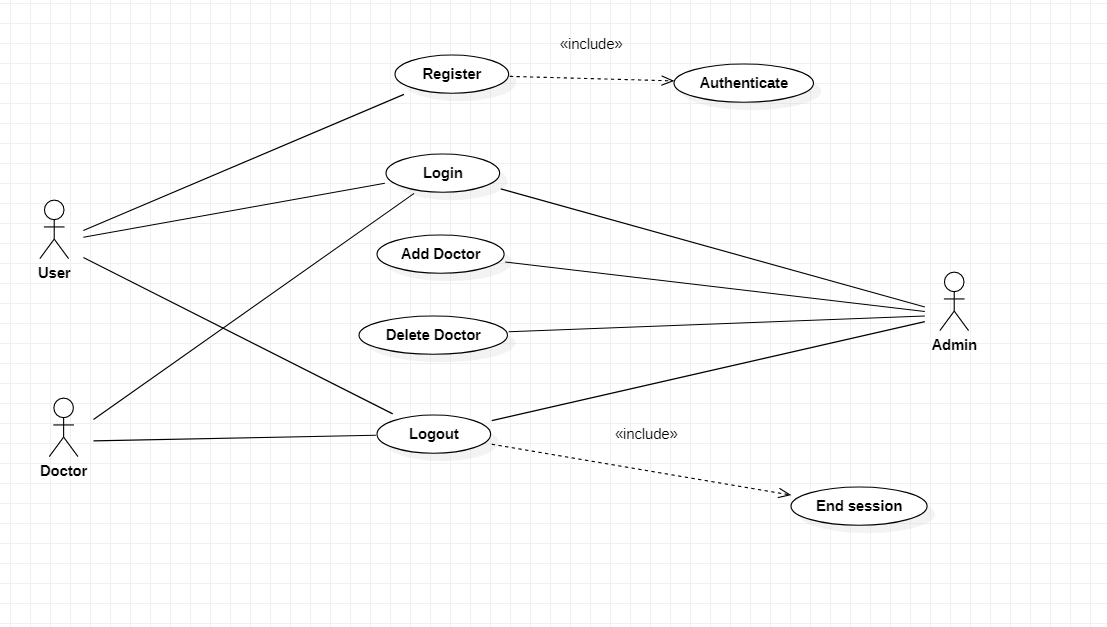


Figure 18: Use case Diagram of UMS

### 5.2.3 SRS Table of Personal Health Management System (Subsystem 2)

|  |  |  |
| --- | --- | --- |
| Requirement Code | Requirement Description | MoSCoW |
| PHMS-F-01 | Allow user (patient) to book appointment. | Must have |
| PHMS-F-02 | Allow user to delete appointment. | Must have |
| PHMS-F-03 | Allow Doctor to view appointment of users. | Must have |
| PHMS-F-04 | Allow Doctor to add prescription in the booked appointment. | Must have |
| PHMS-F-05 | Allow user to see doctor’s prescription report. | Must have |
| PHMS-F-06 | Allow user to know about follow up date. | Should have |
| PHMS-NF-07 | Allow user to download the prescription report. | Could have |
| PHMS-UR-08 | Allow user to Know about doctors Speciality. | Could have |
| PHMS-UR-09 | Allow user to delete health records from the system. | Could have |

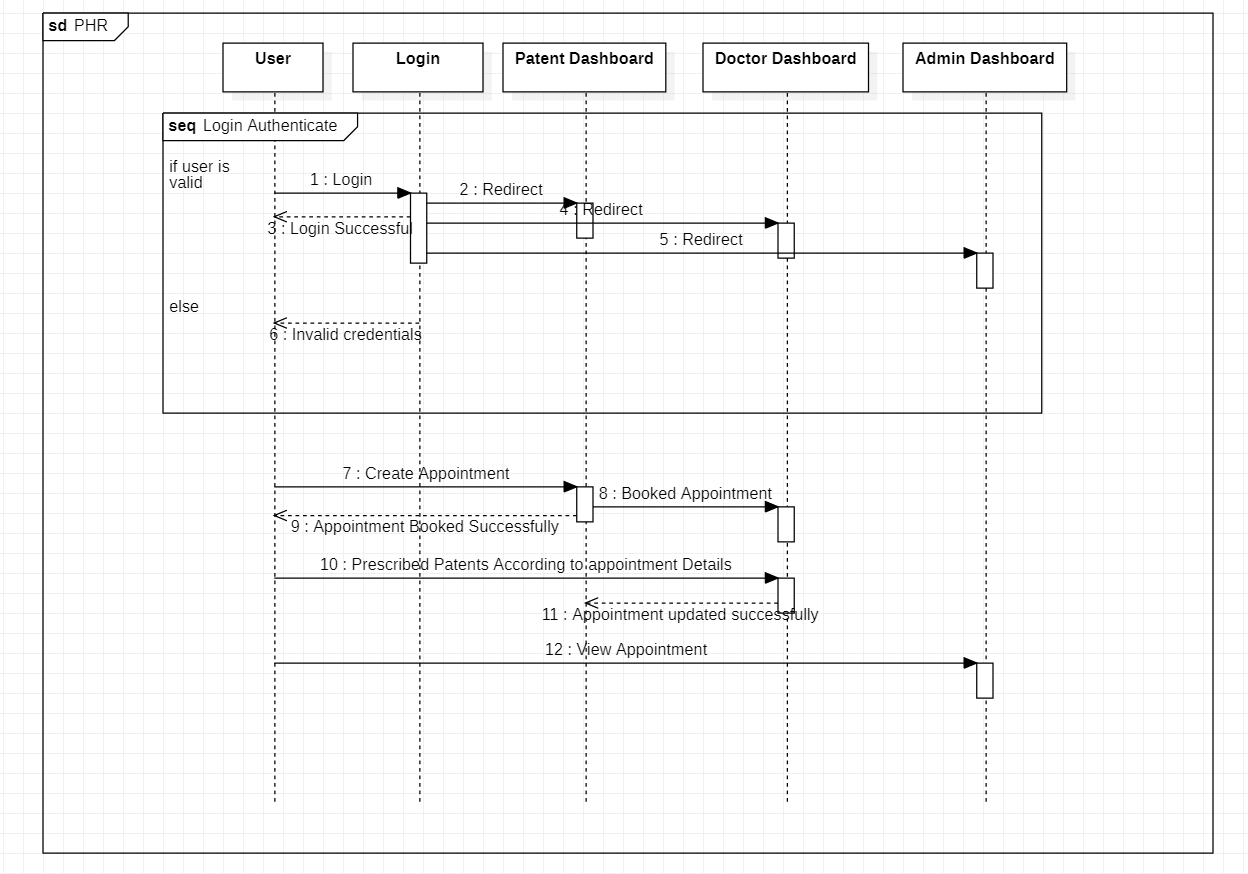


Figure 19: Sequence Diagram for PHMS

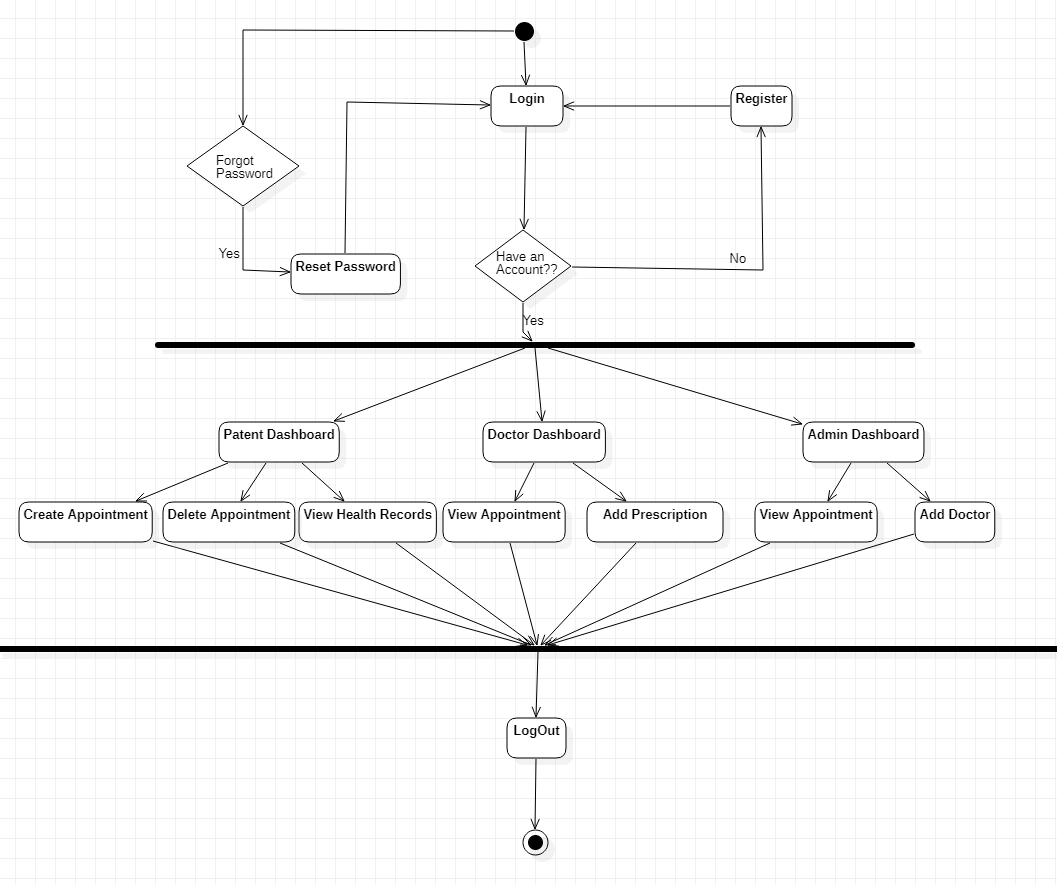


Figure 20: Activity Diagram of PHMS

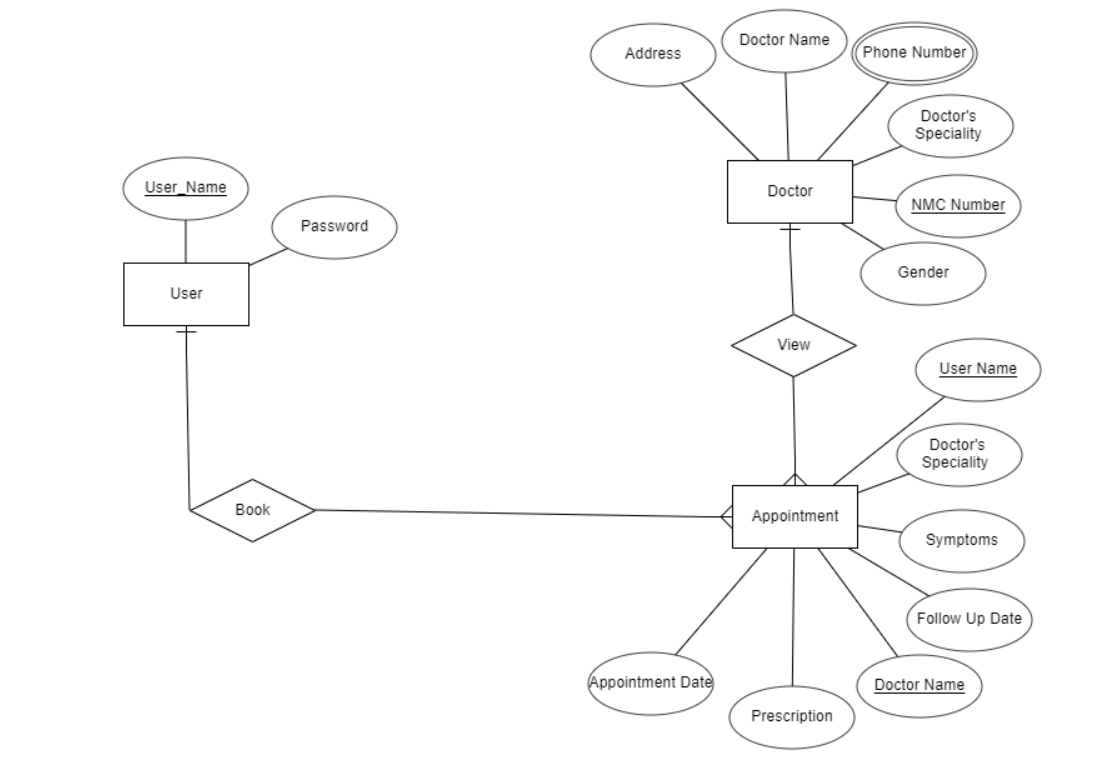


Figure 21: Detailed ER Diagram of PHMS

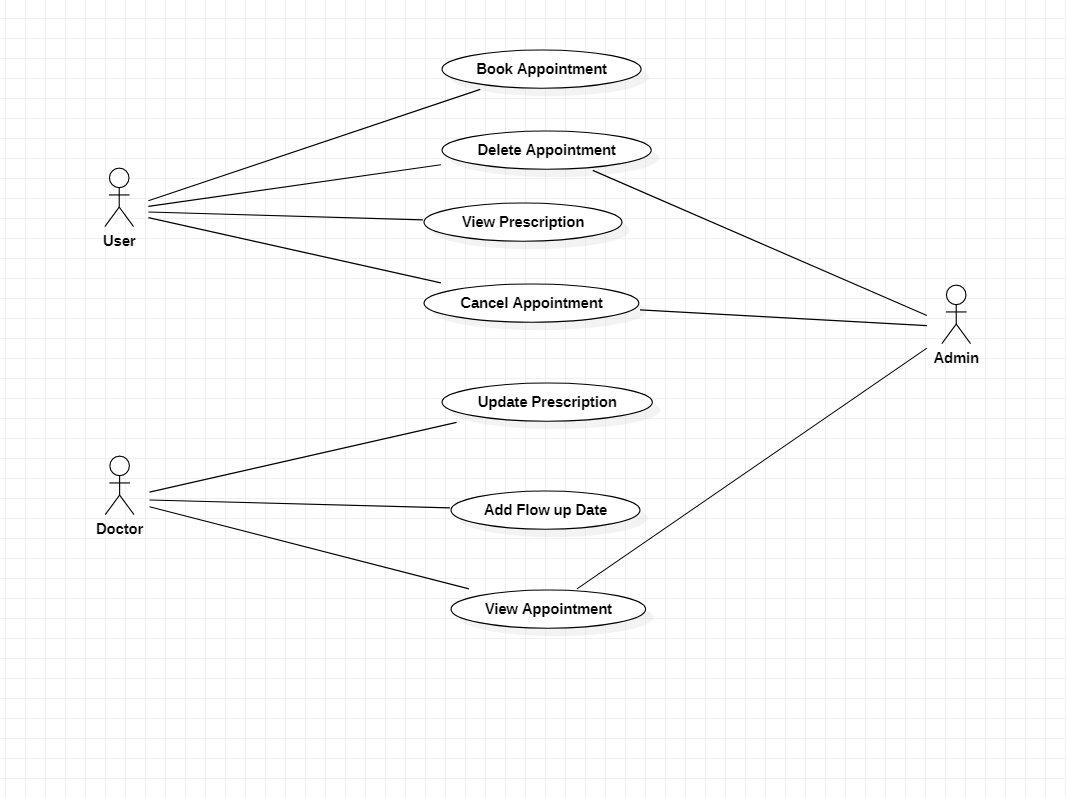


Figure 22: Use Case Diagram of PHMS

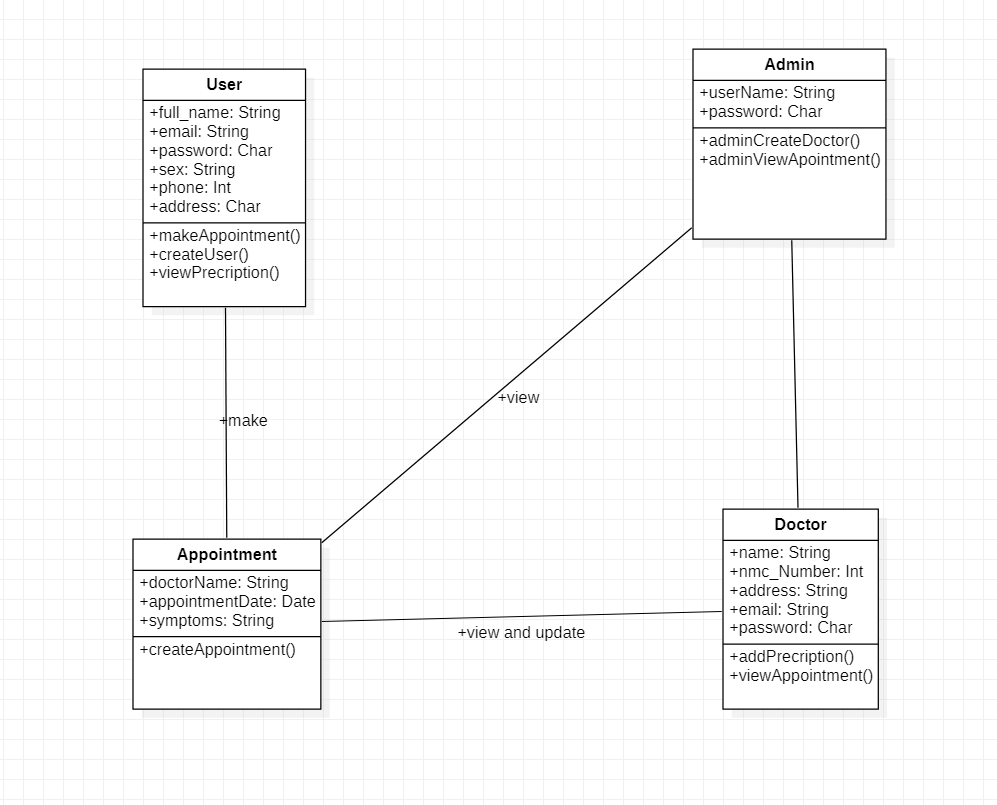


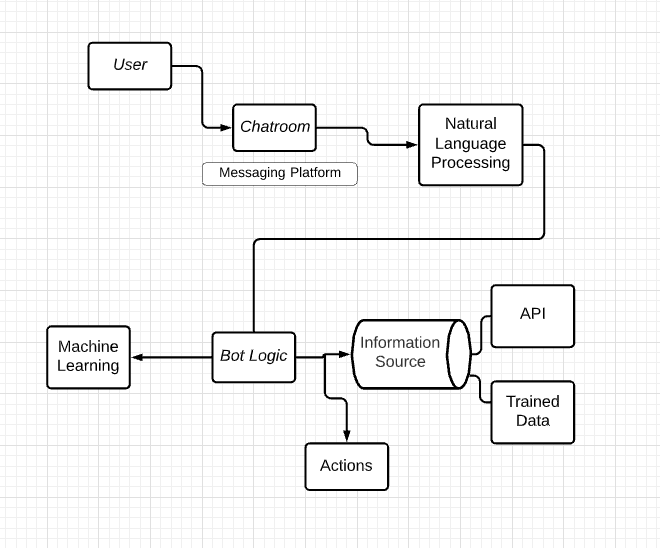
Figure 23: Class Diagram of PHMS

Table

Description automatically generated  
 Figure 24: Wireframe of PHMS

### 5.2.3 SRS Table of Chatbot System (Subsystem 3)

|  |  |  |
| --- | --- | --- |
| Requirement Code | Requirement Description | MoSCoW |
| CS-F-01 | Should allow user to interact with chatbot interface without registration. | Must have |
| CS-F-02 | Should greet user. | Must have |
| CS-F-03 | Allow user to Ask question regarding health. | Must have |
| CS-UR-04 | User could use voice chat to ask questions. | Could have |

  
 Figure 25: Chatbot Architecture

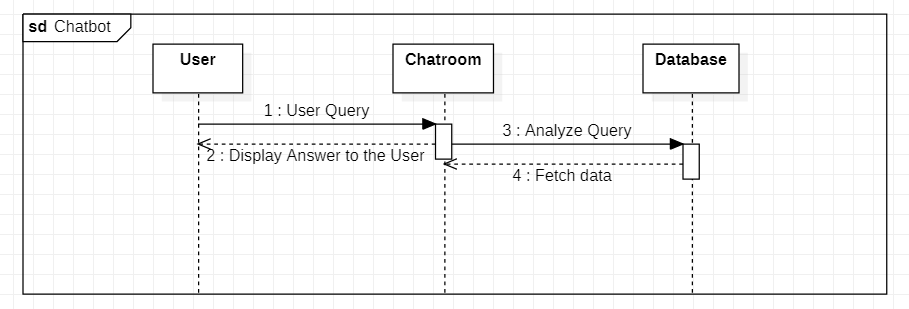
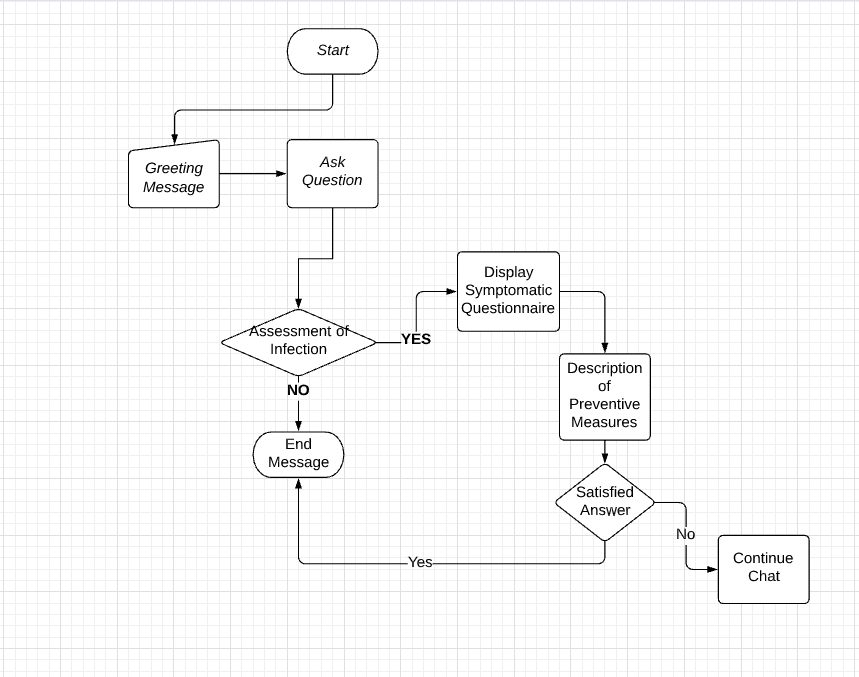


Figure 26: Sequence Diagram for Chatbot System

  
 Figure 27: Flow chart Diagram of Chatbot System

Graphical user interface, text, application

Description automatically generated  
 Figure 28: Wireframe Diagram of Chatbot Interface

Diagram

Description automatically generated  
 Figure 29: Wireframe of Homepage

Graphical user interface, text

Description automatically generated  
 Figure 30:- About Us Wireframe