Software Requirements Specifications

for

Health Center Management Application

Prepared By

Group 12
Software Engineering - IT314

April 28, 2023

List of Contents

Introduction

Problem Statement

A health center management system is a software application that helps to manage the day-to-day operations of a health center. It includes features such as appointment scheduling, patient records management, and billing. The system allows for easy communication between health center staff and patients, making it a useful tool for both parties. The system also allows for the management of medical staff schedules and can generate reports on patients' medical history and treatment plans. The system can be integrated with electronic health records (EHR) systems for easy access to patients' medical history. The use of a health center management system can greatly improve the efficiency and organization of a health center, resulting in better care for patients and a more streamlined operation for staff. The software engineering project would involve the design, development, testing, and deployment of a health center management system, following software development best practices and methodologies.

Purpose

The purpose of a health center management application is to help the healthcare center manage its daily operations more efficiently and effectively. The system automates and streamlines processes such as patient registration, appointment scheduling, medical billing, electronic health records (EHRs), and medicine ordering.

By using a health center management system, the healthcare center can improve patient care by reducing wait times, minimizing errors, and enhancing the accuracy and accessibility of patient data. The system also enables healthcare providers to access real-time patient information, which can improve diagnosis and treatment.

The application allows the patients to book appointments online for both offline and online consultation. The application can enable the patients to keep all their medical records at one place and get their lab test reports also using the same application.

In summary, the purpose of a health center management system is to improve the ease of performing tasks for the patients, keeping all their records safe at a single location for faster access, manages the schedule of the doctors to avoid any clashes in appointments and also allows users to order medicines online.

Scope

The scope of an application for health center management is vast and can benefit both patients and healthcare providers.

The application can provide patients with a user-friendly interface to schedule appointments online, eliminating the need for long wait times and reducing staff workload. Patients can also order medicines through the app, saving time and effort for both patients and healthcare providers.

The application can also allow patients to view their prescriptions and download lab test reports, enabling them to access their medical information quickly and easily. This feature can help patients stay informed about their health and communicate more effectively with their healthcare providers.

From a health center perspective, the application can streamline administrative tasks and reduce paperwork, allowing staff to focus on patient care. The app can also improve inventory management by tracking medicine orders and reducing wastage.

In summary, the application's scope for health center management is to improve patient experience and engagement, increase staff productivity, optimize inventory management, and improve overall healthcare operations.

Overall Description

Product perspective

From a product perspective, a health center management application should be designed to streamline the daily operations of a health center, while also improving the quality of patient care. The software should be easy to use and efficient, and should provide a comprehensive set of features that enable health center administrators and staff to manage patient information, appointments, billing and finances, inventory, and other important tasks.

Some key points that the software must include from a product perspective are:

- Patient Management: The application must allow the patient to book an appointment, upload their previous medical reports, consult a doctor, get the

prescription and pay for consultation. The users should also be able to give feedback about the doctors and staff who treated them through this application.

- Appointment Management: The application must provide tools for scheduling and managing appointments to the patients and also sends reminders or notification to patients.
- Billing and Finances: The user must be provided with all kinds of payment methods that can be used to pay bills through the application.
- Laboratory Management: The doctor after a consultation with a patient can recommend a laboratory test along with the lab. The user must be able to receive the test results on the application itself.
- The patient should be able to order medicines through this application. The medicines are delivered to the patient after the chemist checks the prescription and availability of medicine in the inventory.
- The application must provide real-time guidance and recommendations based on patient data. It can also provide a video calling feature for online consultation for those who are not able to visit the health center. The application must send reminders or notifications about appointments and medicines regularly to the users.

Overall, a health center management software is designed with the needs of both patients and health center staff in mind, providing a user-friendly and efficient platform that supports the delivery of high-quality patient care.

User characteristics

When designing a health center management software, it is important to consider the characteristics of the users who will be interacting with the system.

Here are some user characteristics that should be taken into account:

- Domain knowledge: As the domain of our project is healthcare domain, it is important for users to have some basic knowledge about the domain and the software should be created in a way that it is easy to understand even for users having little knowledge about the domain.
- Technical proficiency: The users may have different levels of technical proficiency, so the software should be easy to use and navigate, with clear instructions and help documentation.
- Role and responsibilities: Different users have different roles and responsibilities within the health center like patients, doctors, laboratory assistant, manager and

chemist, so the software should be customized to meet the needs of each role, providing the relevant information and functionalities.

By taking into account these user characteristics, the health center management software can be designed to meet the needs of its users and provide the best possible support for medical care and administrative tasks.

Assumptions

- The system is designed to be user-friendly and intuitive.
- The system is scalable and can accommodate changes in the health center's operations and requirements over time.
- The system must comply with all relevant laws, regulations, and industry standards governing the management of healthcare information and patient data.
- The system has a patient-centered approach to healthcare.
- The health center is committed to providing high-quality healthcare services to all patients.
- The health center is dedicated to protecting patient privacy and confidentiality
- Each user must have a valid user id and password.
- Server must be running for the system to function.
- Users must log in to the system to access any record.

Specific Requirements

Need for the project

1. A software application that helps in day-to-day management of the health center.
2. Provide easy communication between health center staff and patients.
3. Allow the patient to book appointments and order medicines.
4. Managing inventory of the medicine availability of the hospital.
5. Patient’s records must be maintained and monitored regularly.
6. Provide security for the user data and patient health records should be kept confidential.
7. Billing and financial management based on patient records.
8. The software should manage the schedules of the medical staff without any conflicts.
9. Information about the doctors like experience, achievements, etc should be transparent to the users.

Features of the project

1. User and staff registration, authentication and authorization will be provided.
2. Users will be able to book appointments for their health checkups and can also delete their scheduled appointments.
3. Users can view their uploaded reports based on their medical history and treatment plans.
4. Medical staff can check their schedules using the app.
5. Software will provide a video calling feature for those who are not able to visit the health center.
6. Application will send reminders or notifications about appointments and medicines regularly.
7. In case of unavailability of laboratory equipment, user reports should be sent to the laboratory recommended by the doctor at the allotted time selected by the user.
8. Reports from the laboratory should be available to the user through the app.
9. The user would be able view his prescriptions and also upload them to order medicines.
10. Medicines can be delivered directly to the patient’s address if the patient demands.

Functional Requirements

1. User and staff registration, authentication and authorization will be provided.
2. Users will be able to book appointments for their health checkups and can alter their appointments.
3. Users can get their reports based on a patient's medical history and treatment plans.
4. Users will be able to give feedback about the doctor and staff who treated them through this application.
5. All kinds of payment methods can be used to pay bills through the app.
6. The application will provide local language support.
7. Software will provide a video calling feature for those who are not able to visit the health center.
8. Patients can update details about medical insurance and previous medical treatments.
9. Application will send reminders or notifications about appointments and medicines regularly.
10. In case of unavailability of laboratory equipment, user reports should be sent to the laboratory recommended by the doctor at the allotted time selected by the user.

11. The user would be able view his prescriptions and also upload them to order medicines.
12. Medicines can be delivered directly to the patient's address if the patient demands.

Non-Functional Requirements with their Justifications

1. Information of the patient like medical history, reports and treatment details, personal information should be secured.
   Justification : There is a need to keep the patient's data safe and secure as many users want their data to be private. Leakage of patient's data might lead to dissatisfaction for the patient. There is also a chance of booking manipulations by attackers.

2. The application should be available around-the-clock to the users.
   Justification : Patient requires the system to be available whenever he tries to access it and the system should respond to requests sent at any time by the user. As patient health issues might be serious and the unavailability of the system may lead to failure of addressing those patients.

3. The application should manage the schedules of medical staff without any conflicts.
   Justification : As the appointment schedules are automatically created and updated by the system, it should ensure that there are no conflicts occurring in the schedules of medical staff. As medical staff might not check their schedules regularly to report clashes.

4. Notifications about change in appointment timings should be sent to the patient within 5 minutes.
   Justification : If there are any change in timings of an already booked appointment due to the unavailability of a doctor or due to some other reasons, then the patient should be notified quickly so that the patients have updated information regarding the appointment. This is critical when the appointment time is changed at the last moment.

5. Reliability, concurrent request handling and high performance, low latency.
   Justification : The system is able handle multiple requests from the users at the same time to ensure that there is no inconsistency in schedules or timings. High

latency in responding, low performance might lead to a poor experience for the patient.

6. The User Interface should be simple and easy to use.
   Justification : The patients who are using the app might prefer a clear and understandable user interface as many patients might not have a good technical experience in using complex applications.

7. Reports of the test should be available within two days.
   Justification : Patients want their lab test results as soon as possible. Delays in the reports might lead to patients getting tensed about their health conditions.

8. The system should be able to automatically provide backup of the data regularly.
   Justification : As there is a chance of system crash or database crash or any such hazards, it might lead to partial or complete loss of patients data. So, there should be some backup mechanism used by the system to ensure that the data will be available even though there is a system crash.

Design goals

1) **Reliability:**
   Reliability is a major concern as low reliability leads to higher risk of errors and failures which potentially harms the patients using the system.
2) **Maintainability**:
   The system should be easy to maintain and manage as it should provide services all around the clock to ensure that emergency services will be serviced.
3) **Efficiency:**
   There should not be any delays in providing services to patients as it might lead to poor user experience and patients may dislike the system.
4) **User-friendliness:**
   Most patients using the application may not be professional users and can find it difficult if the application is not user friendly.
5) **Flexibility:**
   The system should be flexible enough to adapt to new changes in future as there will always be some evolving requirements by the users.
6) **Fault tolerance:**

The system should be resistant to any potential failures and in case of failures it needs to be recovered using backups and other methods to ensure that patients data is not lost.

Software Model

Incremental Model

We have used the incremental model for our software development.

Reason:

- In our software development requirements are superior and can be changed as per the requirements. The incremental model allows for flexibility in the development process, allowing for changes to be made as needed. This is particularly important in healthcare, where requirements may change over time due to advancements in medical technology or changes in healthcare policies.
- The incremental model delivers the software in increments, which means that each increment provides a working system that can be tested, evaluated and improved before the next increment is developed. This approach helps our group to identify issues and make necessary changes before the system is fully implemented, resulting in a more effective and efficient system. In short, we have enough time so that we can develop different modules of the software and can test it in parallel with development.
- The incremental model helps to mitigate risks or we can say risk mitigation by breaking down the development process into smaller and more manageable increments. This approach helps us to identify potential risks early on in the development process, allowing for prompt action to be taken to minimize or eliminate these risks. Here the risk of failure is reduced as we are developing the software in smaller increments.
- The incremental model encourages user involvement in the development process. This can be particularly important in healthcare, where the end-users are often the medical staff, who will be using the system to manage patient care. Involving users in the development process can help to ensure that the system meets their needs and is user-friendly, leading to better adoption and utilization of the system.
- The incremental model allows us to deliver a working system more quickly, which is particularly beneficial in healthcare where there may be urgent needs to implement new systems or processes to improve patient care.

- Also at the end it will lead to an improved software quality as each increment will be thoroughly tested and refined.

Use Case Descriptions

1) Name: Fee payment

- Actors: User, manager
- Goal: To make payment for booking an appointment/ lab test/ ordering medicine.
- References to requirements: Features of the project - 7
- Pre-conditions:
    - Users must be logged in.
- Description:
    1. The user will be displayed the total payment amount according to the charges of his appointments/orders.
    2. The user will be asked to choose a payment method from the various options available.
    3. On successful completion of the payment, the user will be sent the receipt of the payment and the manager will be notified about the payment.
- Exceptions:
    - 3a. On payment failure due to any reason, the user is notified about the failure and is asked to perform the payment again.
- Post-conditions:
    - The transaction is shown to the user in the payment history section.

2) Name: Book/update/cancel appointments

- Actors: User, manager, doctor
- Goal: To schedule/update/cancel appointments of users with their respective doctors.
- References to requirements: Features of the project - 2
- Pre-conditions:
    - The user must be logged in.
    - The doctor must be registered in the system.
    - For updating/cancelling, the user must have an appointment booked with the doctor previously.
- Description:
    1. The user selects the doctor to book/cancel/update an appointment with.
    2. The user will be asked to enter the time period when he wants to book an appointment.

3. The user is then displayed the available slots of that doctor within that period.
4. The user selects a slot and confirms the appointment.
5. For updating/cancelling appointments, the user needs to select the appointment slot.
6. For updating, steps 2-4 are performed.
7. The doctor and the manager will be notified about the appointment.

- Exceptions:
    - 4a. If no slot is available in the time period or if the patient is not available when the slot is free, then the user can re-select the time period.
- Post-conditions:
    - The appointment will be displayed to the user in the upcoming appointments section in case of booking/updating.
    - The appointment will be removed from the upcoming appointments section in case of cancellation.

3) Name: Update medical history and medical insurance details

- Actors: User
- Goal: To allow the user to update his past medical history and/or current medical insurance details
- References to requirements: Features of the project - 9
- Pre-conditions:
    - The user must be logged in.
- Description:
    1. The user is asked about his previous medical history and is required to enter details of his medical condition.
    2. The user is also asked to enter about the medication he is currently taking if any.
    3. The user is asked to enter the details of his medical insurance policy if he has any.

4) Name: Download reports

- Actors: User, doctor
- Goal: To enable the user and doctor to download lab reports
- References to requirements: Features of the project - 3
- Pre-conditions:
    - The user must have performed a lab test earlier.

    - The doctor must be assigned to the user if he wants to download the user's report.
- Description:
    1. The user can go to the notifications section and view the lab test reports.
    2. The user can download the report on his device.
    3. The doctor can open his user's profile and can download the lab test reports of the user.
- Post-conditions:
    - The downloaded reports must be available for the user in the downloaded reports section.

5) Name: Sending notifications/reminders

- Actors: User, doctor, laboratory assistant
- Goal: To remind the user/doctor/lab assistant about their upcoming actions through notifications.
- References to requirements: Features of the project - 10
- Pre-conditions:
    - The user/doctor/lab assistant must be having upcoming actions.
- Description:
    1. The users are notified about their appointments a day earlier.
    2. Doctors are reminded about their entire set of appointments on the next day.
    3. Lab assistants would be reminded about all the tests to be performed on the next day.
    4. The default notification period will be a day before but can be changed as well.
- Post-conditions:
    - The reminders will be shown to the user/doctor/lab assistant in the notifications section.

6) Name: Recommend laboratory test along with lab

- Actors: Doctor, user
- Goal: The doctor recommends the user to perform some test(s) if required and also recommends the lab where the user could get the test performed
- Pre-conditions:
    - A doctor can recommend a test to a user only if he is assigned to the user.
- Description:

1. On checking the user, the doctor can recommend a few tests to the user if necessary.
2. The doctor can also recommend the lab where the user could get the tests performed.

- Post-conditions:
    - The user can book lab tests if necessary.

7) Name: Order medicines

- Actors: User, chemist
- Goal: The user can order medicines from the chemist online.
- References to requirements: Features of the project - 14
- Pre-conditions:
    - The doctor may have prescribed medicines to the user.
    - The user must be logged in.
- Description:
    1. The user has to enter the name of the medicines he wants to order or upload a photo of the prescription.
    2. The user has to specify the quantity of each medicine required.
    3. The user is asked to select a previously entered delivery address or add a new delivery address.
    4. The user is then asked to make the payment.
    5. On successful payment, the user is sent the receipt and the chemist is informed about the order.
- Exceptions:
    - 5a. On payment failure due to any reason, the user is notified about the failure and is asked to perform the payment again.
- Post-conditions:
    - The medicines will be delivered on the entered delivery address.

8) Name: Update reports

- Actors: Manager, Laboratory assistant, User
- References to requirements: Features of the Project-3
- Preconditions:
    - Lab assistant and doctors must send the reports to the manager
- Description:
    1. The Manager will receive the reports from the lab assistant and doctors.
    2. The manager will Update the lab reports to the user account

- Exceptions
    - 2a: If the update is unsuccessful the manager will retry to update the documents
- Post-conditions:
    - The manager will be able to see the time of the updated documents made to the user.

9) Name: Online Consultation

- Actors: Doctor, User
- References to requirements: Features of project-8
- Preconditions:
    - User,Doctor must be logged in
    - User must selected the online consultation option
    - Users must have a device that supports video call feature and internet.
- Description:
    1. The user will receive a video call from the doctor.
- Exceptions:
    - 1a:In case the user has not been able to connect, the call doctor will call the user again.
- Post-conditions:
    - Doctors can prescribe medicines and recommend lab tests to the user
    - User will be able to see the prescribed medicines and lab tests

10) Name: Check patients record

- Actors: Doctor, manager
- References to requirements: Features of project-3
- Preconditions:
    - Must patient must have a record
- Description:
    1. Doctor and manager can check patient's medical records
- Post-conditions:
    - Doctors can update patients' records and send them to the manager.

11) Name: View Schedule

- Actors: Doctor, User, Laboratory assistant, Manager
- References to requirements: Features of project-5
- Preconditions:

    - Doctors, User, Laboratory assistant, Manager must be logged in.
- Description:
    1. Doctors, laboratory assistants can view schedules which are booked by the user
    2. User can view their booked appointments
    3. Manager can view schedules of Doctors and laboratory assistants
- Post-conditions:
    - Doctor, User, Laboratory assistant can update their schedules

Finalized tools, technologies and frameworks

Tools: Android Studio
Technology: Android
Framework: Android Native (in kotlin), Firebase.
Reason:
Android is developed and maintained by Google and there are abundant resources available to learn. Android has a large community of developers and users compared to other app development frameworks like Flutter. There are many open source libraries available for almost everything in Android which can be directly used while developing, hence reducing the development time and effort.

Firebase is a Backend as a service(Baas) which includes a real-time database, cloud storage for storing large documents, authentication and testing the application. It also provides Push Notification service.

Database for the project
Database chosen: Firebase Database
Reason :

- Firebase Database is a real-time No-SQL cloud-hosted database developed by Google.No separate server is needed to host the database, hence it is cost effective.
- The database can be accessed directly from the client devices without having a need for an application server to interact with the database.
- Even when the client device is offline, firebase database provides the latest fetched data and once the device is reconnected, the data will be synced automatically.

Effort estimation

Overall use-case size point estimation:

Technical Complexity Factor (TCF)

Technical Factor	Calculated factor
T1	10
T2	4
T3	2
T5	2
T6	2
T7	2
T8	6
T9	3
T10	2
TOTAL	**33**

TCF = 0.6+(0.01*33) = 0.93

Environmental Complexity Factor (ECF)

Environmental Factor	Calculated Factor
E1	6
E2	1
E5	1
E6	10
Total	**18**

ECF = 1.4 + (-0.03*18) = 0.86

Unadjusted Use Case Weight (UUCW)

Use Case Type	Result
Simple	60

Total	60

UUCW = 60

Unadjusted Actor Weight (UAW)

Actor Type	Result
Simple	5
Total	**5**

UAW = 5

Unadjusted Use Case Points (UUCP)
UUCP = UUCW + UAW = 60+5 = 65

Productivity Factor (PF)
PF = 20

Final Calculation:
UCP = TCP * ECF * UUCP
UCP = 0.93 * 0.86 * 65 = 51.987

Estimate = UCP * PF
Estimate = 51.987 * 20 = **1039.47 hours**

Domain analysis modelling

Boundary, Entity and control objects

Boundary objects:

- Authentication interface

  It is located at the boundary of the system where the user interacts with the system and uses it to login or signup.

- Payment gateway and interface

  It is used by the user to make payments for booking appointments/lab tests or ordering medicines.

- Android interfaces

These interfaces allow the user to interact with the system while performing the desired functionality.

Entity objects:

- Patient profile and medical history

  It is an entity object as it contains data about real world objects i.e personal and medical details of the patient and is present at the end of the path from the user through boundary and control objects.

- Medical report

  It contains the data of the result of the lab test performed on a patient by a lab assistant.

- Appointments

  It contains the information about the scheduled consultation of a user with a doctor with the exact date and time.

- Medicine order

  It contains the information of the medicines ordered by the user with the bill date and amount.

- Doctor's schedule

  It contains all the information about all the appointments of a doctor within a specific time period.

- Payment history

  It contains the data about all the payments made by a user in the past.

- Notifications

  It contains information about some reminders for the user. Notifications are sent for all important activities like reminding about upcoming appointments, medicine deliveries, lab reports and many more.

- Prescription

  It contains data generated by the doctor for a patient about the medicines that the patient must take and the lab reports that he/she must perform after consulting him/her.

Control objects:

- Patient Data Manager

It is a control object as it controls the data flow between boundary interfaces and their respective entity objects related to patients' medical and personal data.

- Appointments interface

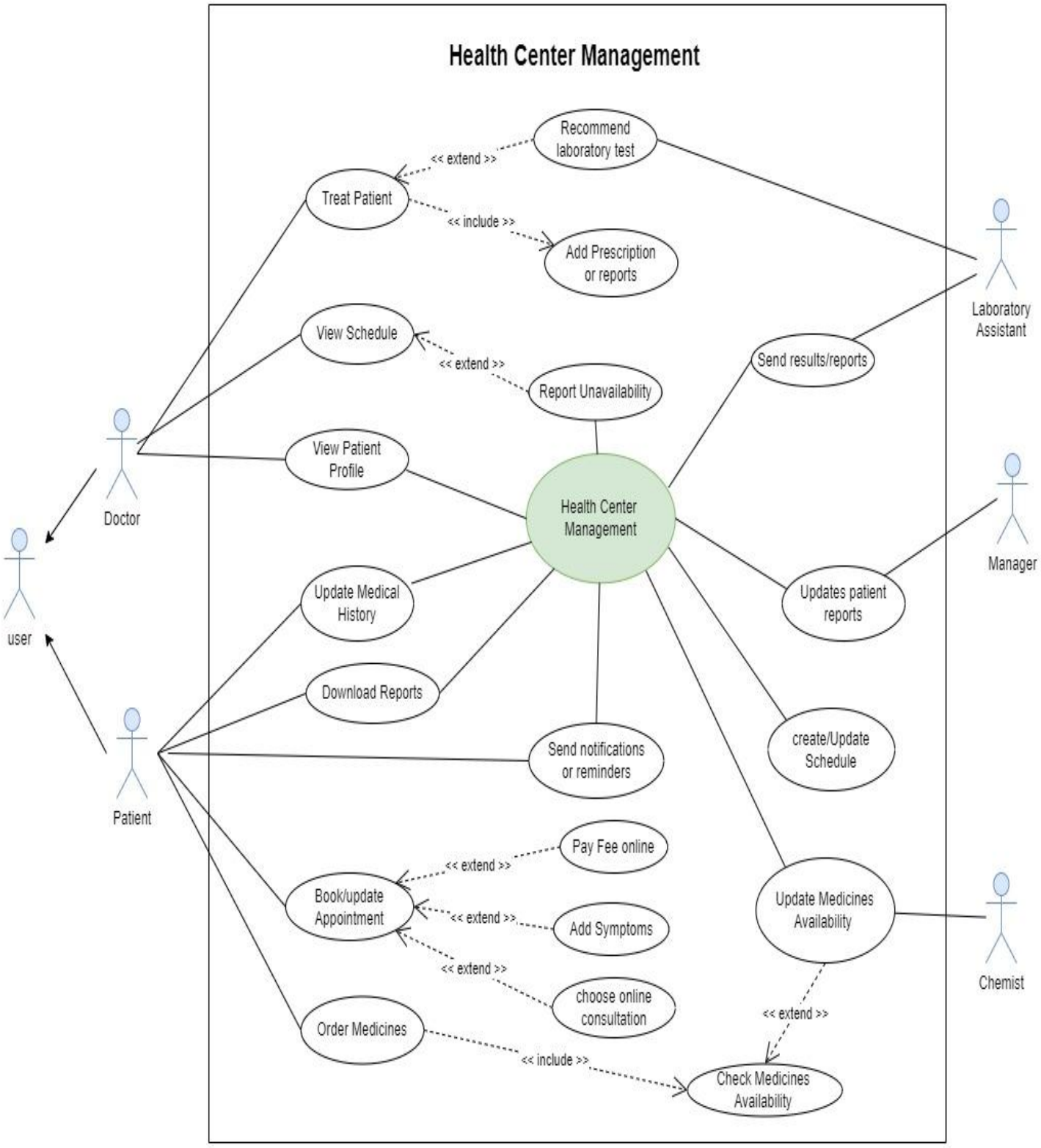
  It controls appointments booking by users, and assigns it to doctors and also manages any conflicts between schedules, performs rescheduling and so on.

- Payment methods

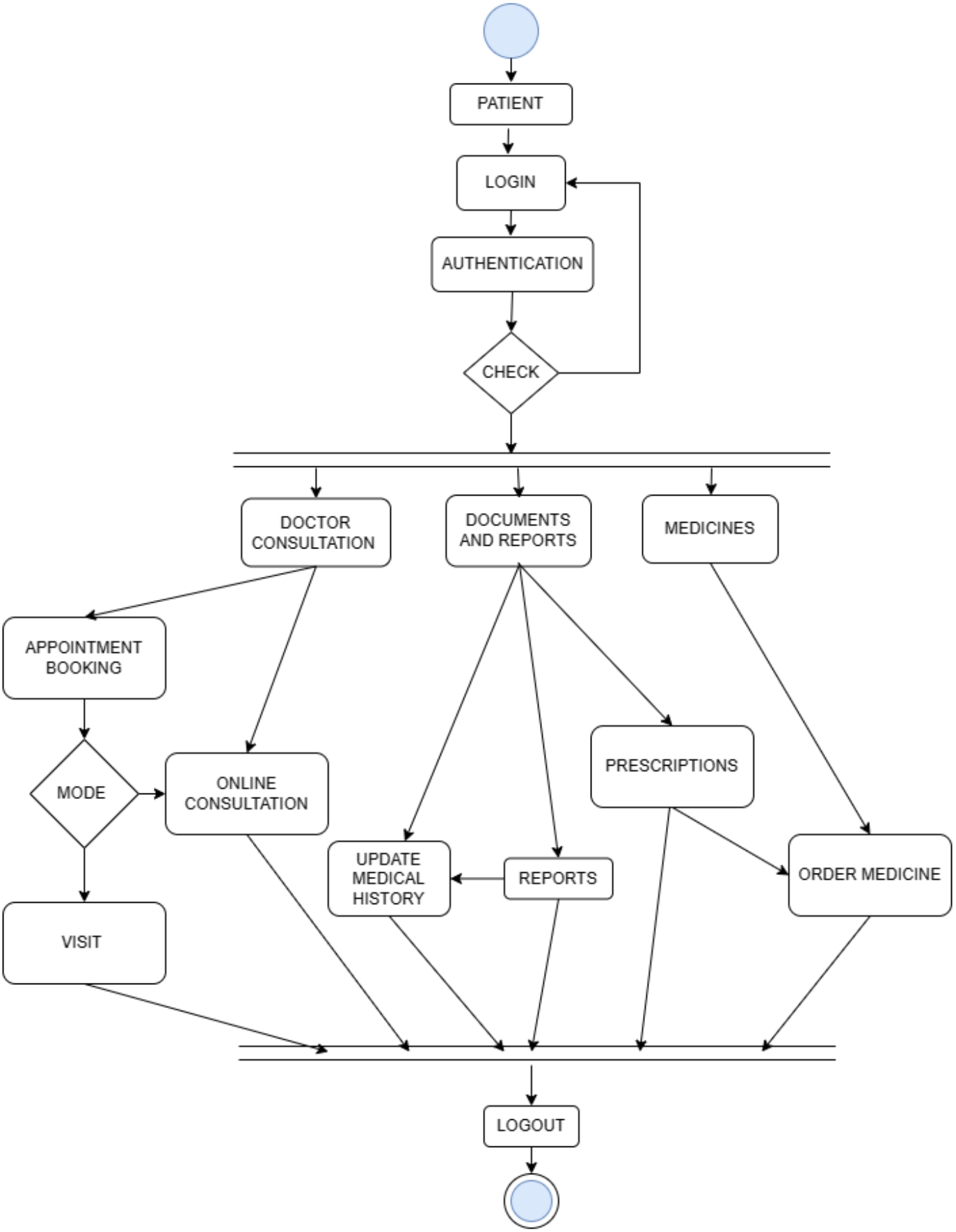
  It is a control object as it directs a user from either starting to book an appointment/lab test to successfully booking it or searching for medicines from the chemist and placing an order.

UML Diagrams

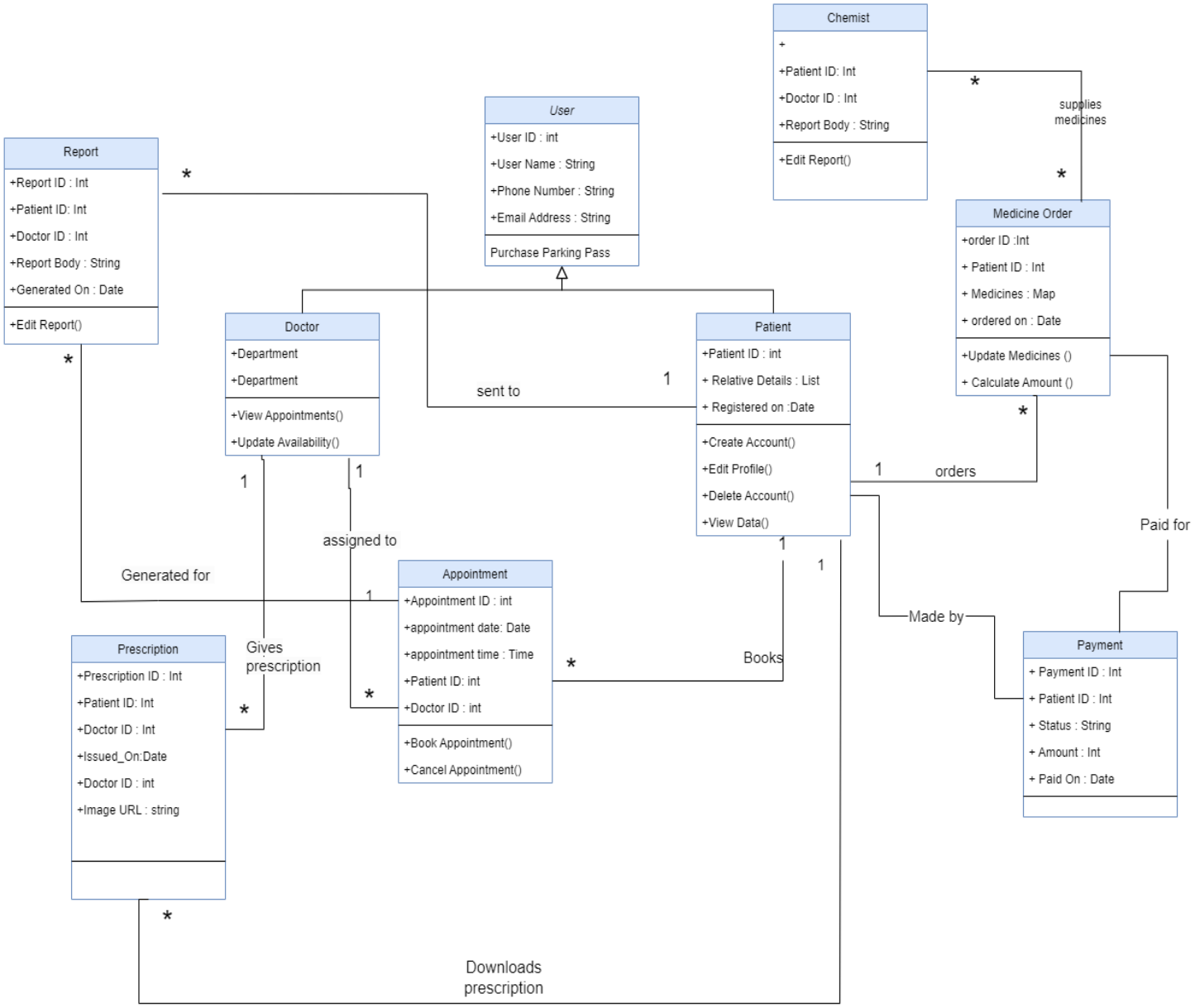
Use Case Diagram



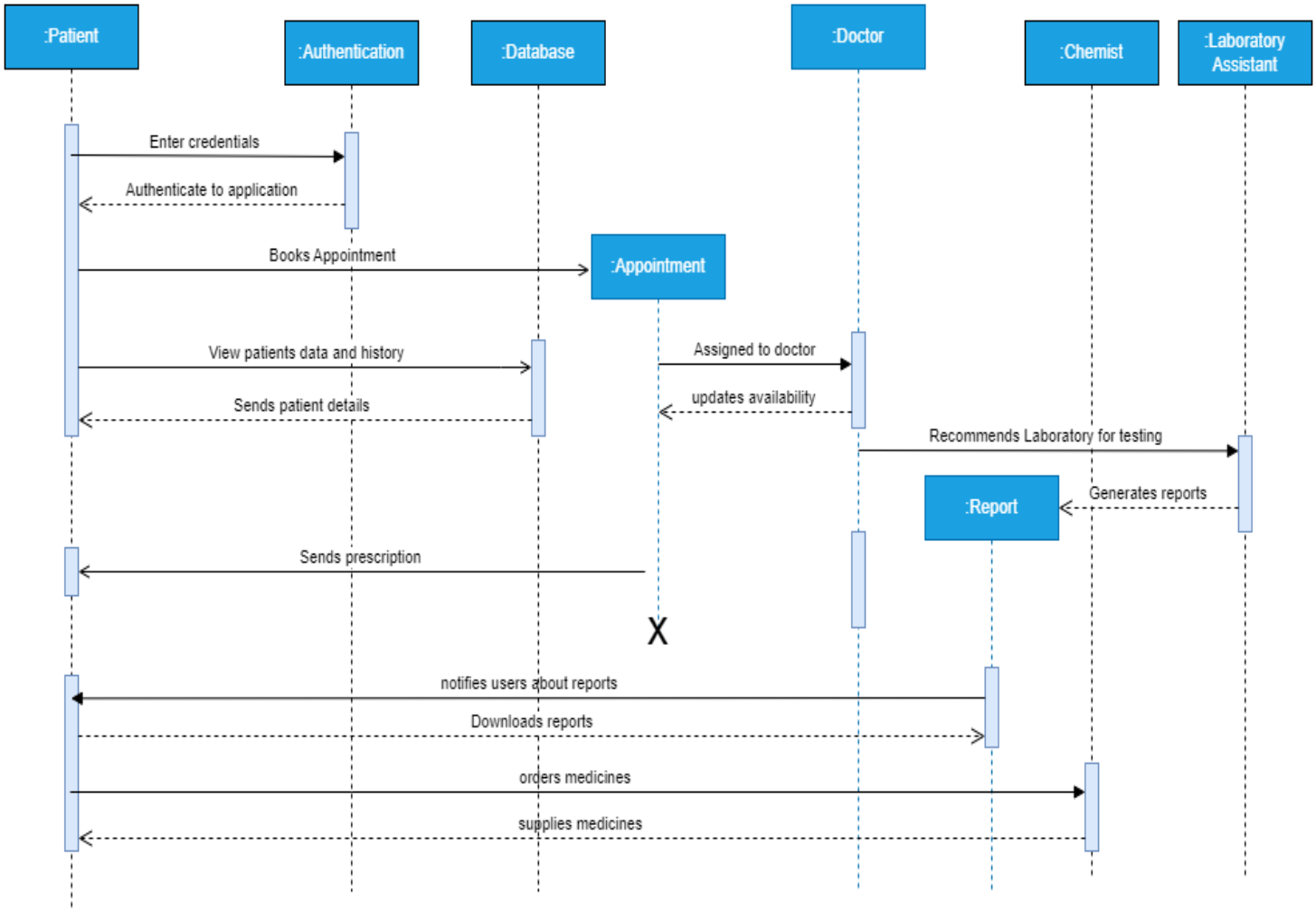
Activity Diagram


PATIENT
LOGIN
AUTHENTICATION
CHECK
DOCTOR CONSULTATION
DOCUMENTS AND REPORTS
MEDICINES
APPOINTMENT BOOKING
MODE
ONLINE CONSULTATION
VISIT
PRESCRIPTIONS
UPDATE MEDICAL HISTORY
REPORTS
ORDER MEDICINE
LOGOUT

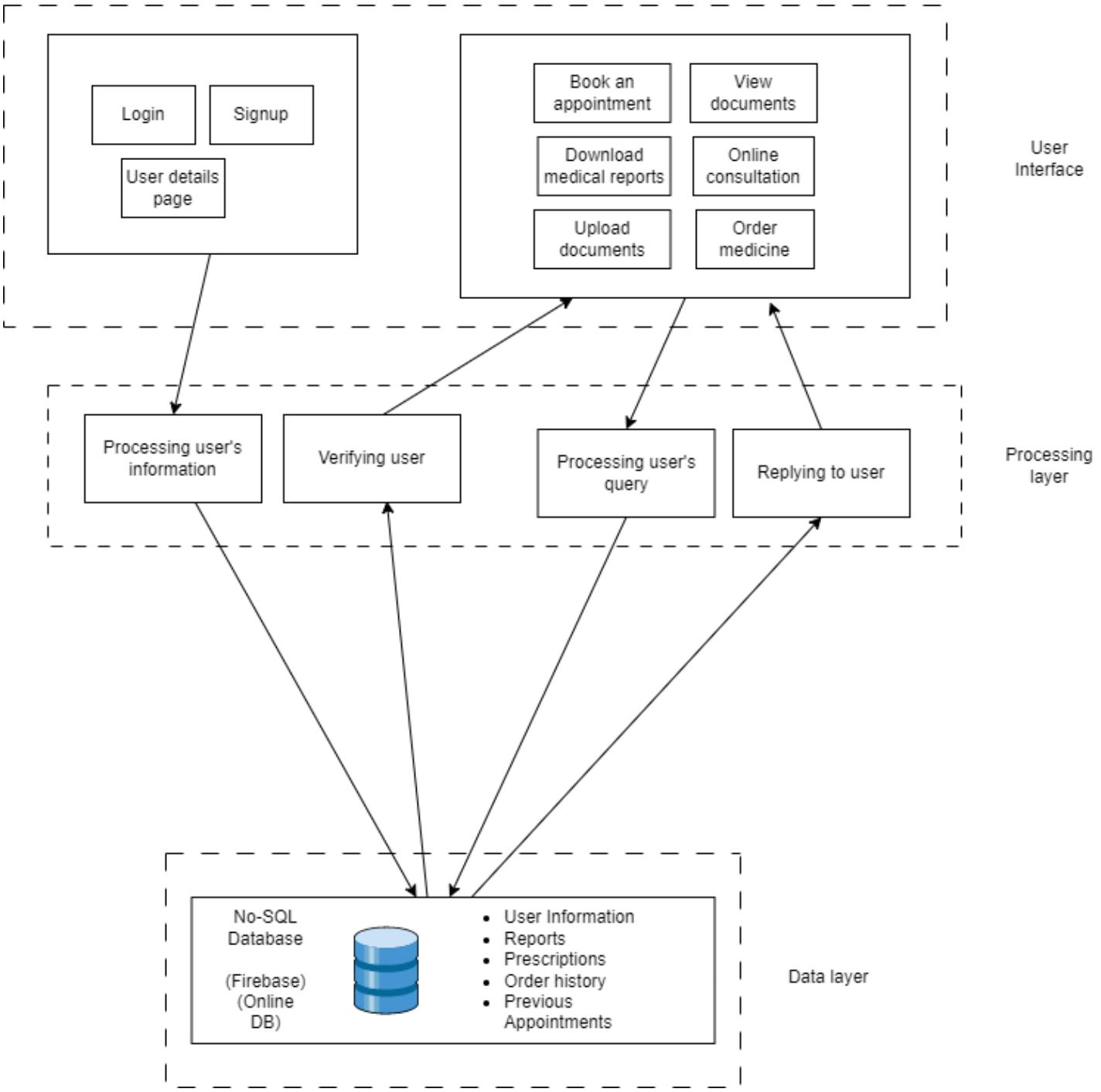
Class Diagram



Sequence Diagram



High level system design



Architecture : Layered Architecture

We have used layered architecture consisting of 3 layers.

- The upper layer is the user interface using which the users interact with frontend of the system. This is the only layer that is exposed to users.

- The middle layer is the processing layer which is the backend of the application which processes user queries and sends appropriate responses.
- The lower layer is the data layer which stores the data of the system. It is a No-SQL database provided by firebase used for storing and querying data.

State Diagram