MobileDoc

Your Personalized Healthcare Companion

Mahdee Mushfique Kamal 1805032
Abu Humayed Azim Fahmid 1805036
Iftekhar E Mahbub Zeeon 1805038
Khondker Salman Sayeed 1805050
H.A.Z. Sameen Shahgir 1805053
Shamit Fatin 1805055

Contents

1 Introduction

MobileDoc is a telemedicine platform to address the lack of healthcare infrastructure in many rural areas and the increasing number of MBBS doctors. Through its platform, MobileDoc enables licensed doctors to provide remote consultations to patients via messaging and video calls. With a focus on providing quality healthcare services, MobileDoc offers a range of medical services including treatment for common illnesses, primary healthcare and family physician consultancy, counseling from licensed psychiatrists, and guidance from licensed nutritionists. By connecting patients with licensed healthcare professionals, MobileDoc aims to bridge the gap between underserved populations and quality healthcare services. MobileDoc, as a telemedicine solution, aims to contribute to the provision of quality healthcare services, particularly in underserved areas.

2 Scope of the Project

2.1 Motivation

Our app aims to bridge the gap between patients and doctors by providing a personalized healthcare companion that connects patients with licensed healthcare professionals via messaging and video calls.

Common illnesses such as allergies, cold and flu, conjunctivitis, and stomach aches require instructions and prescriptions from a doctor, but visiting a hospital or clinic is not always necessary. This is especially true in areas where there is a lack of primary healthcare infrastructure, such as family physician services.

In fact, according to recent statistics, there is a 44.2% vacancy in Upazila level hospitals, and there is a correlation between the distance from cities and the number of vacant health personnel positions. As a result, many individuals living in rural areas have limited access to healthcare professionals.

However, everyone deserves access to quality healthcare, regardless of their location. This is why our app connects patients with licensed healthcare professionals who can provide medical advice, diagnoses, and prescriptions, all from the comfort of their own homes.

Furthermore, our app also caters to the increasing number of MBBS doctors who are completing further study and want to live in urban areas. By providing them with a platform to connect with patients virtually, they can still provide medical assistance to those living in rural areas or to those who are unable to visit a clinic in person.

Accessibility to mobile internet is growing rapidly across the country. This is why we believe that our app can revolutionize the way patients receive healthcare. By using our app, patients can easily and quickly connect with healthcare professionals, receive diagnoses and prescriptions, and manage their healthcare needs conveniently and effectively.

2.2 Objectives of MobileDoc

MobileDoc is a comprehensive telemedicine solution designed to address the healthcare needs of individuals in a variety of settings.

Treatment of common/minor illness: MobileDoc offers treatment for a wide range of common and minor illnesses. This includes conditions such as allergies, cold and flu, conjunctivitis, stomach aches, and other similar ailments. Users can receive medical advice, prescriptions, and treatment recommendations from licensed healthcare professionals on the platform.

Consultancy as primary healthcare or family physician: MobileDoc provides consultancy services as primary healthcare or family physician. Users can consult with licensed healthcare professionals to receive advice and support on a wide range of medical concerns, including chronic illnesses, preventive care, and health maintenance.

Healthcare to the rural population: MobileDoc's telemedicine solution enables individuals living in rural areas to receive quality healthcare services. With the lack of healthcare infrastructure in many rural areas, individuals may not have access to traditional medical facilities. However, with MobileDoc, they can connect with licensed healthcare professionals via video calls, and receive the medical advice and prescriptions they need, all from the comfort of their own homes.

Counseling from licensed psychiatrists: MobileDoc provides counseling services from licensed psychiatrists. Individuals struggling with mental health concerns, such as anxiety or depression, can connect with a licensed psychiatrist on the platform to receive the support and guidance they need to manage their mental health concerns effectively.

Information and guidance from licensed nutritionists: MobileDoc offers information and guidance from licensed nutritionists. Users can receive guidance and support for a wide range of concerns, including weight management, chronic illness, and dietary restrictions. The licensed nutritionists on the platform can provide personalized nutrition plans and recommendations to help users improve their overall health and well-being.

Advantages for junior doctors: MobileDoc also enables junior doctors who have not yet specialized to get patients with common illnesses. This provides an opportunity for junior doctors to gain valuable experience in treating a wide range of medical conditions, and to develop their skills and expertise in the field of medicine.

Work from home: MobileDoc provides doctors with the flexibility to work from home. With traditional healthcare infrastructure, doctors are often required to work in hospitals or clinics, which can be challenging for those with families or other commitments. However, with MobileDoc, doctors can connect with patients from the comfort of their own homes, providing a more flexible and convenient work environment.

2.3 Types of Users

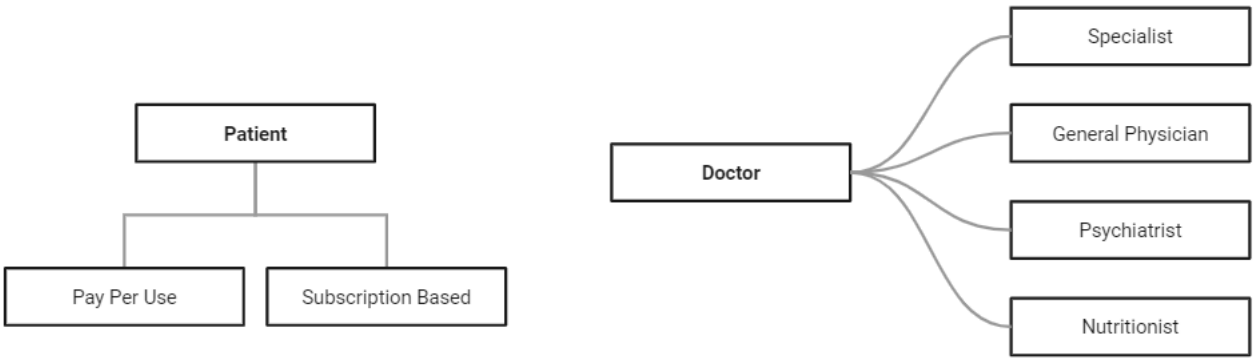

Figure 1: Types of Users

2.4 Modules

MobileDoc comprises of the following modules to provide patients and healthcare professionals with a seamless experience.

- Module for collecting user submitted medical history
- Module for submitting symptoms
- Module for scheduling
- Module for video call session
- Module for prescription
- Module for peer review

Each of these modules caters to a unique function to provide patients and healthcare professionals with a convenient and efficient platform for comprehensive medical consultations.

3 BPMN Diagram

3.1 The Full Diagram

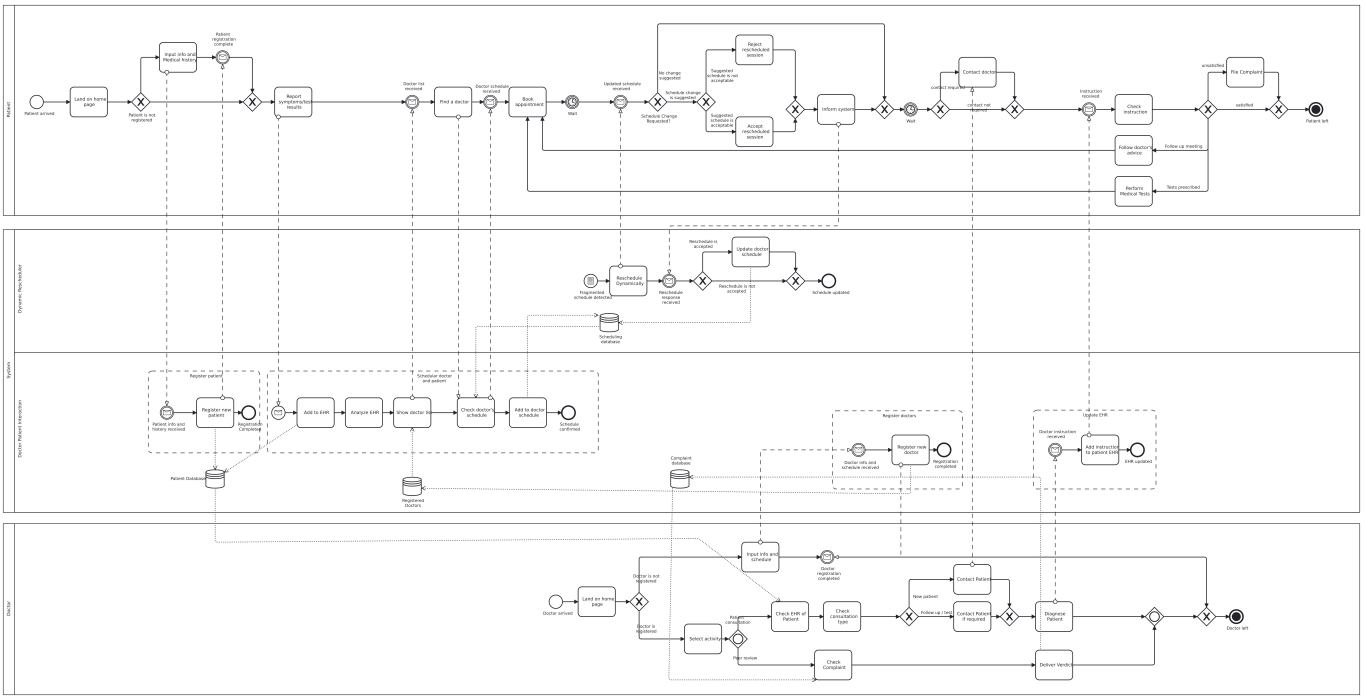

Figure 2: The Full BPMN Diagram

3.2 Patient

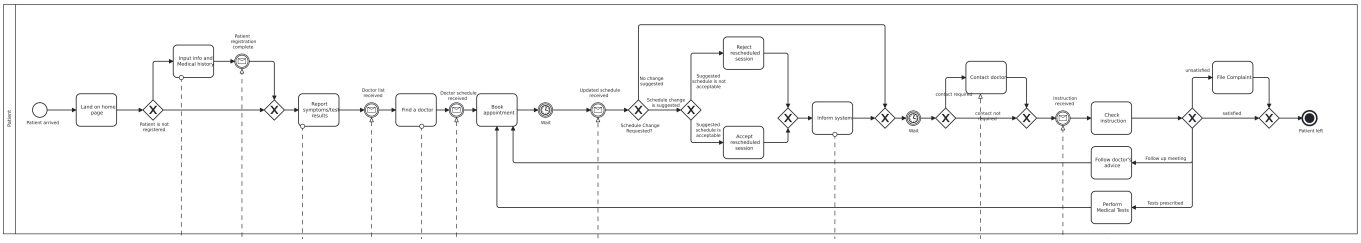

Figure 3: Patient BPMN Diagram

3.3 System

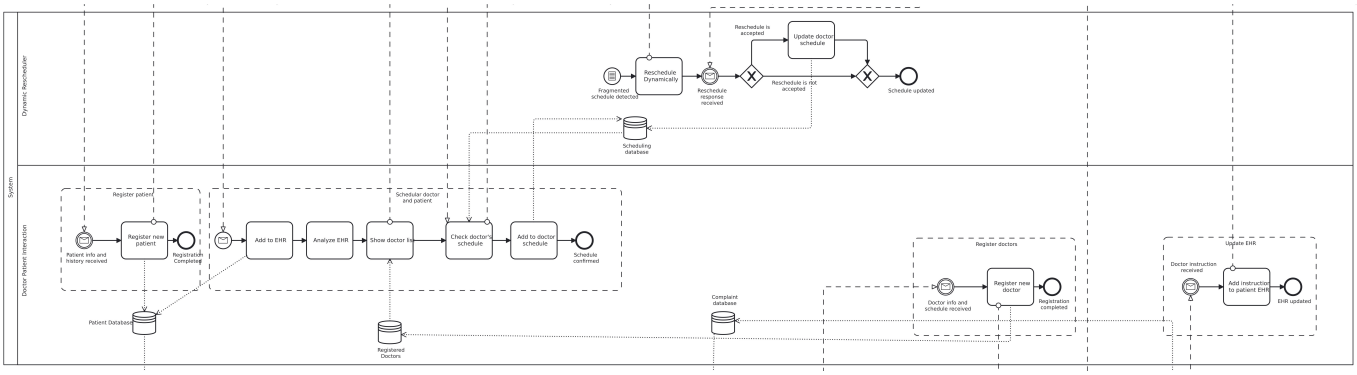

Figure 4: System BPMN Diagram

3.4 Doctor

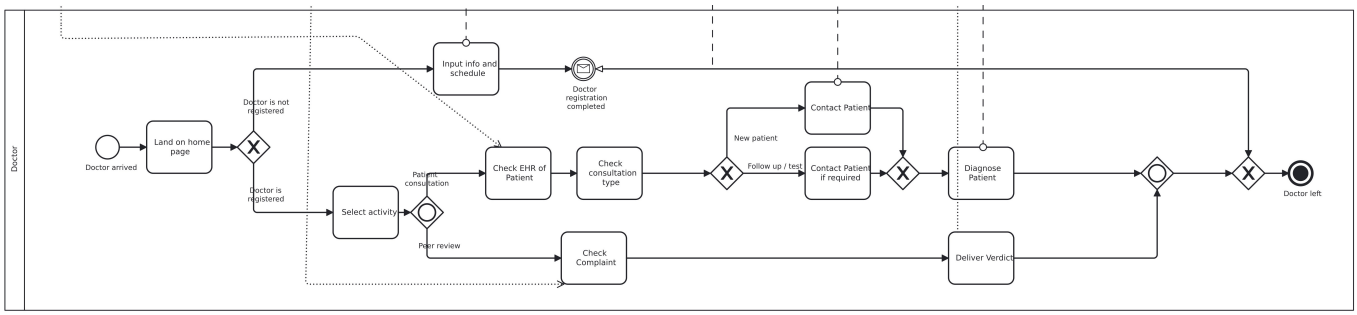

Figure 5: Doctor BPMN Diagram

3.5 Feedback from Evaluator

- Patient's medical history is important for the doctor to view in order to make a correct diagnosis and advice properly. Therefore, there should be a way to integrate a patient's previous test reports and prescriptions into the MobileDoc system. We have integrated that in our database and showed it in our ER diagram
- The doctor should be able to view these inputs in the patient's Electronic Health Record. We have incorporated this feature as well.

4 Mock UI

4.1 Patient Registration

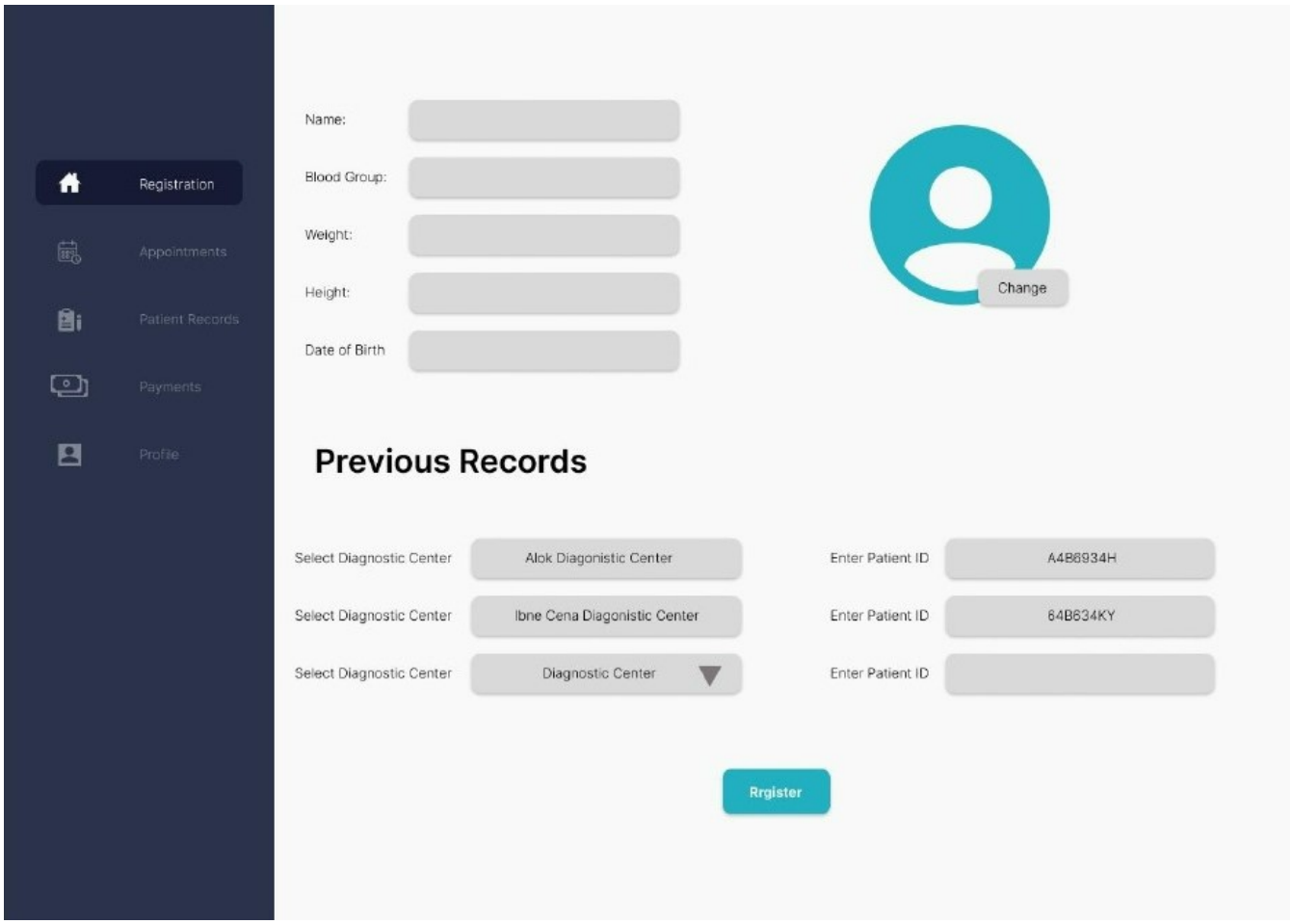

Figure 6: Patient Registration Mock UI

4.2 Electronic Health Record

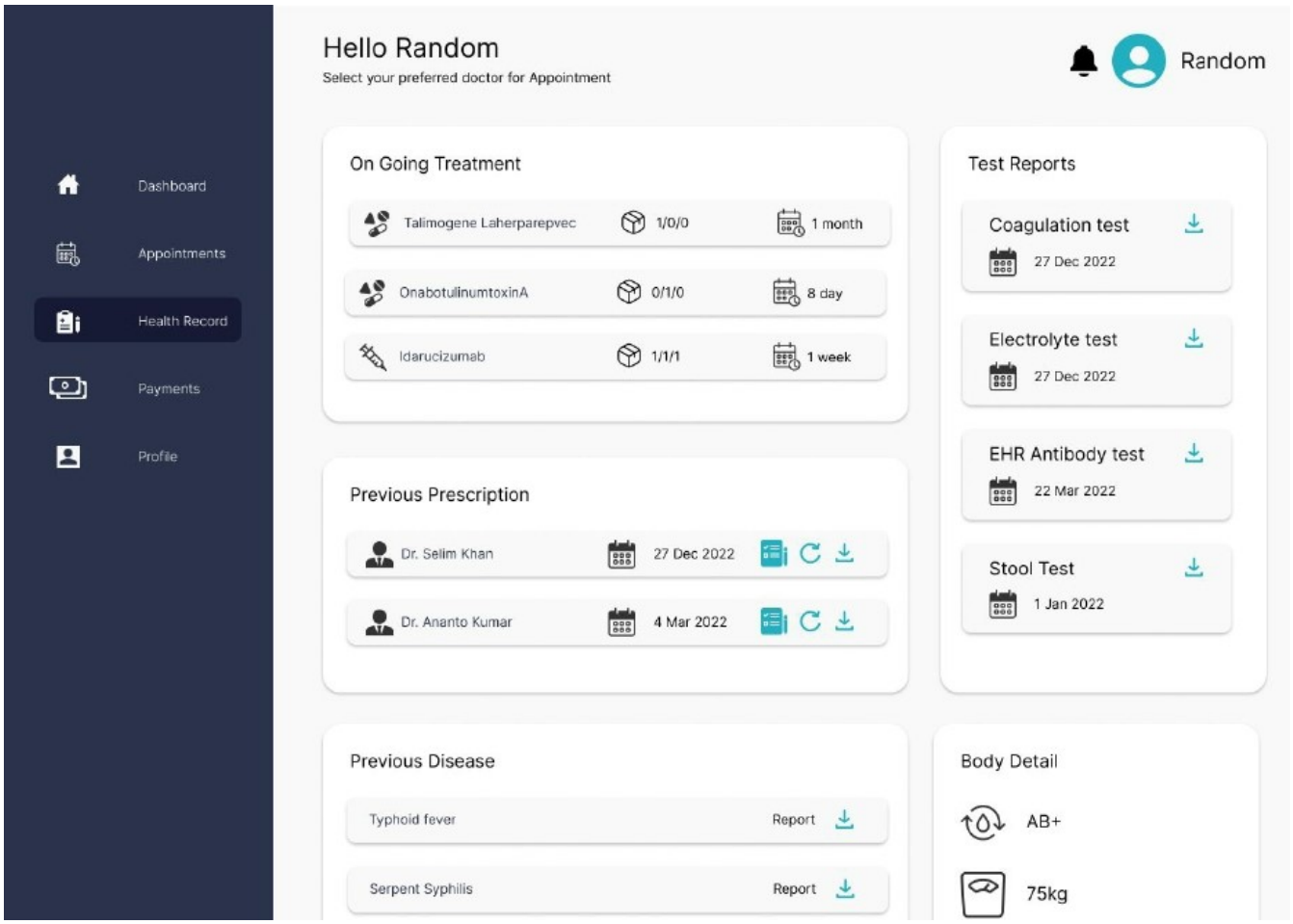

Figure 7: Electronic Health Record Mock UI

4.3 Patient Dashboard UI

4.3.1 Patient Dashboard without Appointment

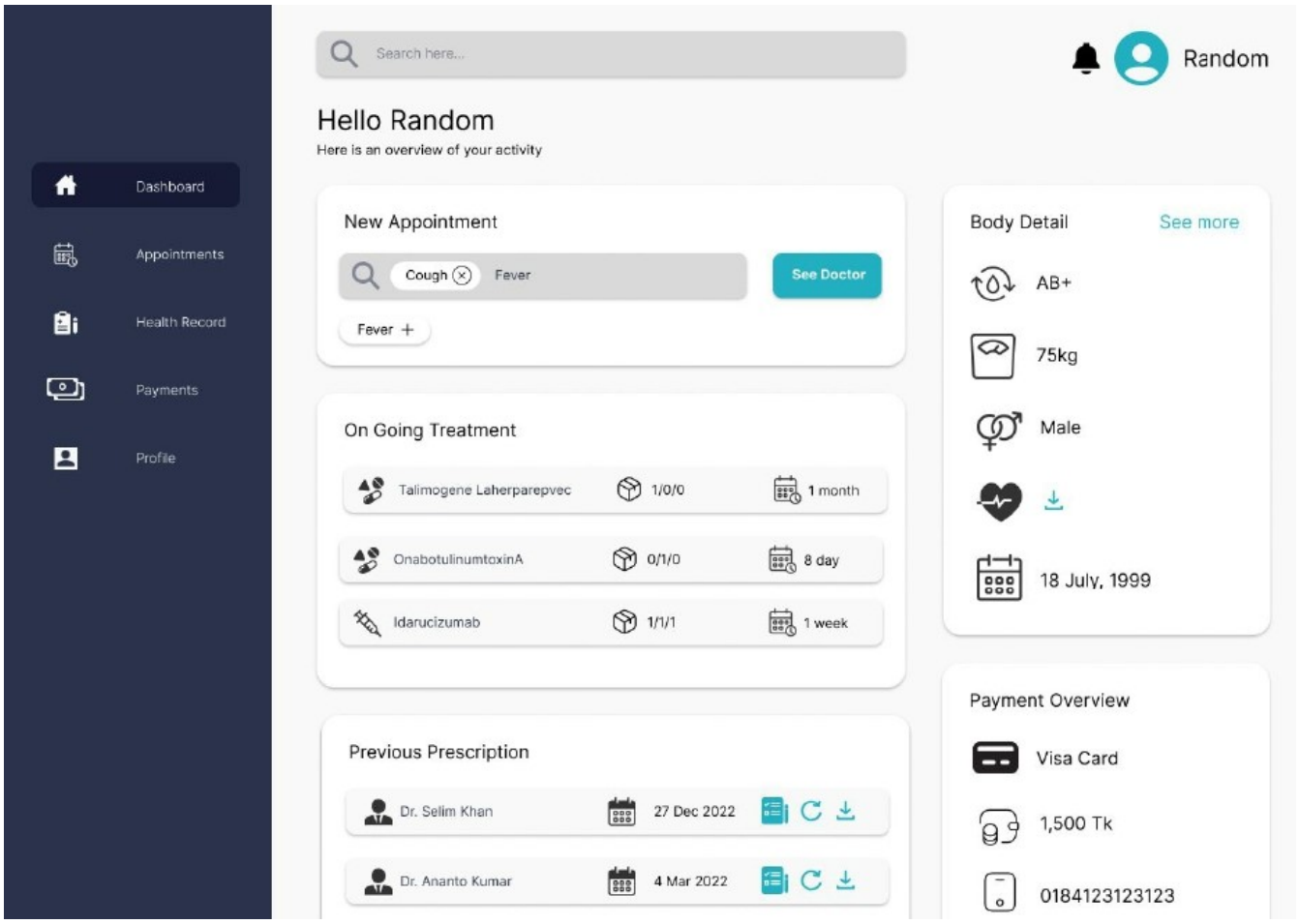

Figure 8: Patient Dashboard without Appointment Mock UI

4.3.2 Appointment

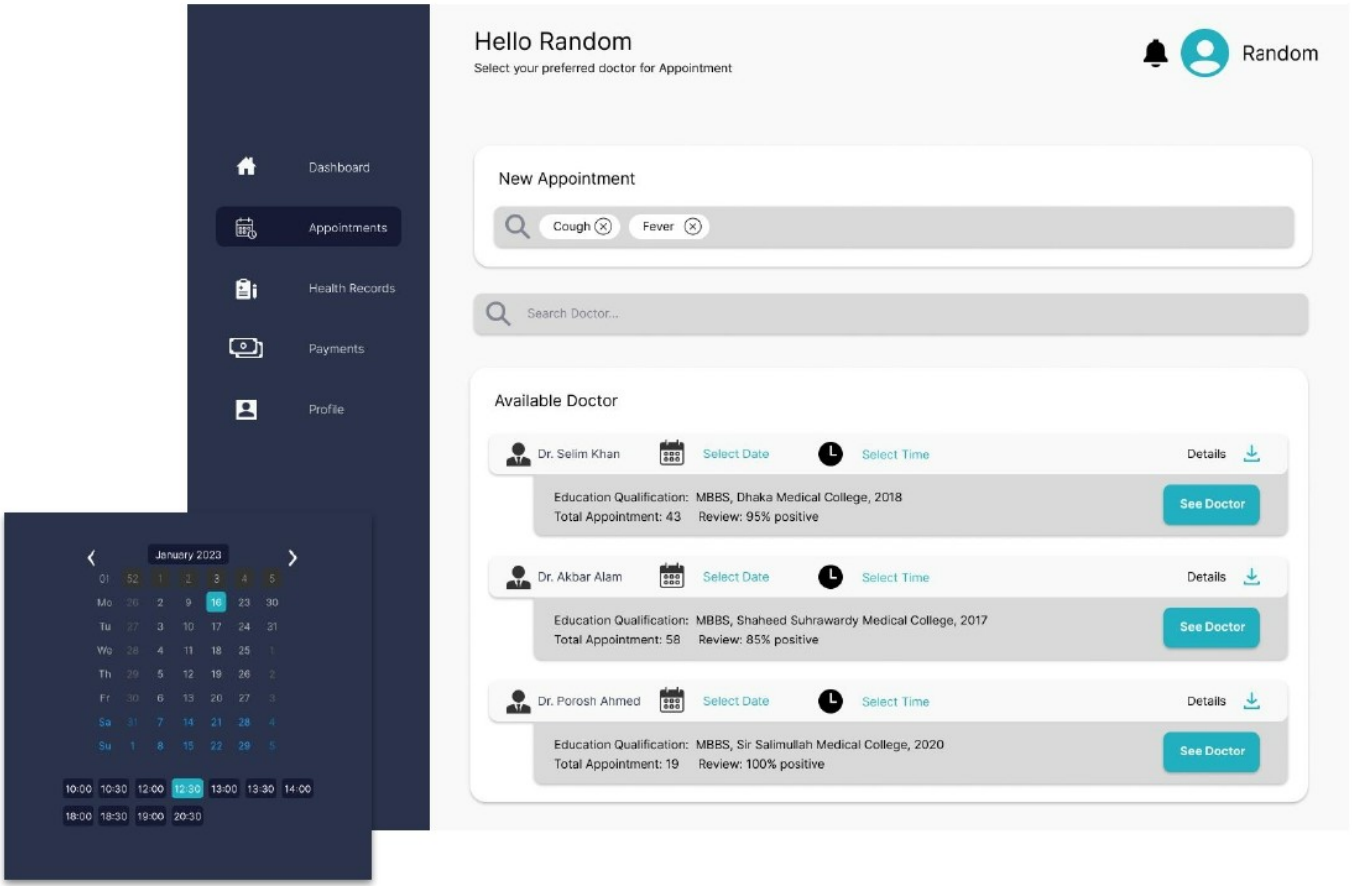

Figure 9: Appointment Mock UI

4.3.3 Patient Dashboard after Appointment

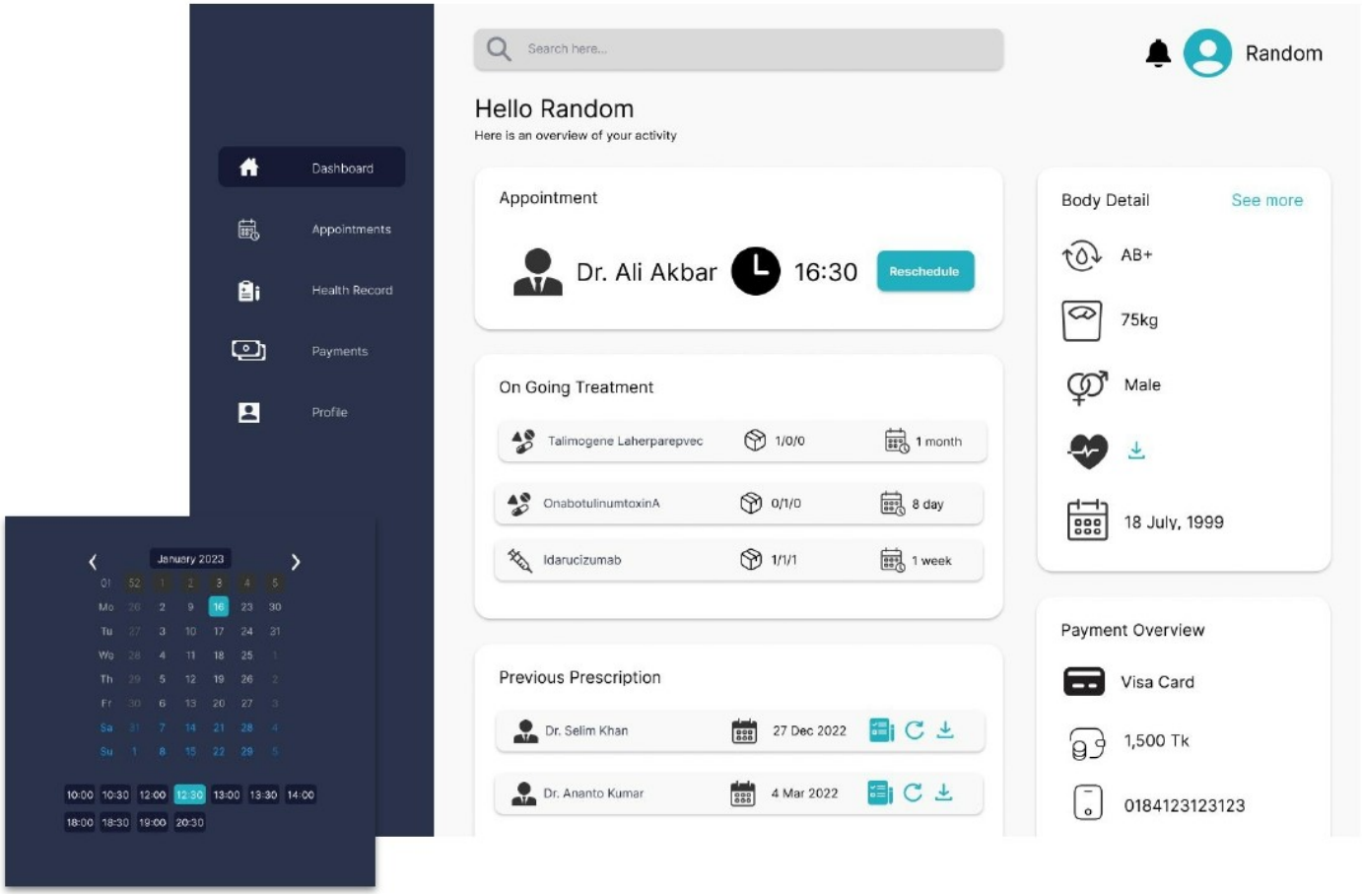

Figure 10: Patient Dashboard after Appointment Mock UI

4.4 Doctor Dashboard UI

4.4.1 Doctor Dashboard without upcoming Appointment

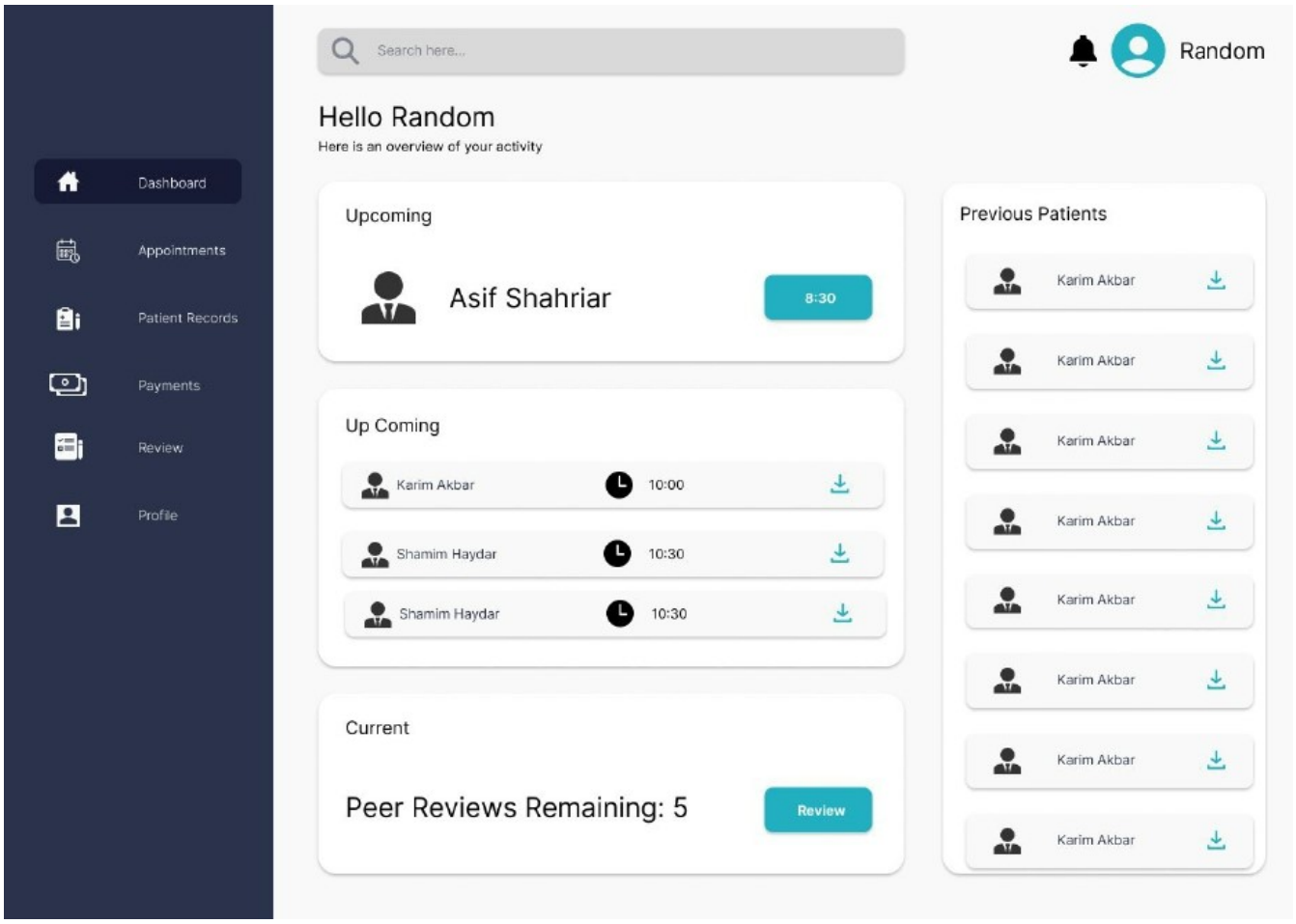

Figure 11: Doctor Dashboard without upcoming Appointment Mock UI

4.4.2 Doctor Dashboard without ongoing Appointment

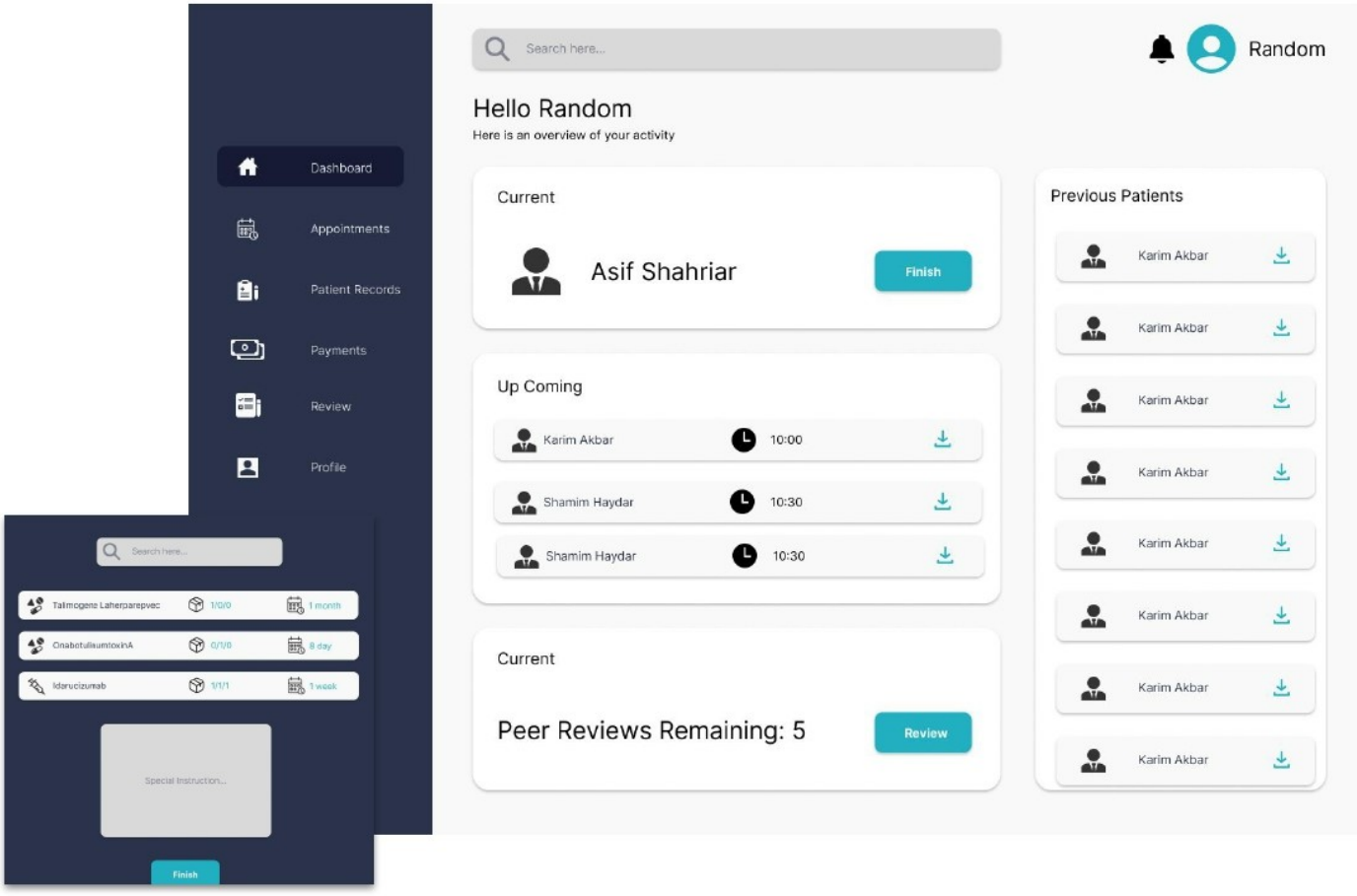

Figure 12: Doctor Dashboard without ongoing Appointment Mock UI

4.4.3 Schedule Calendar

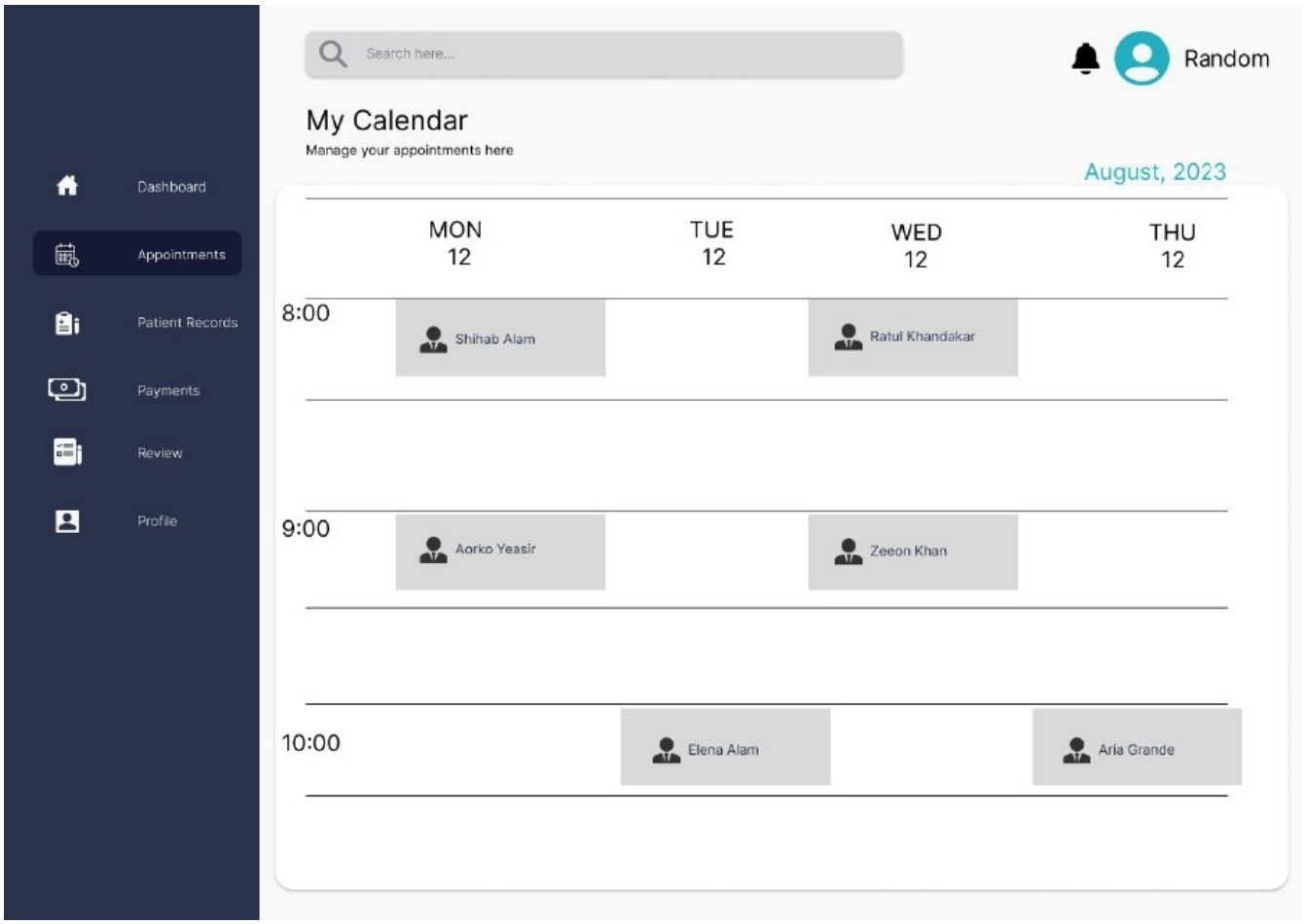

Figure 13: Schedule Calendar Mock UI

4.4.4 Peer Review

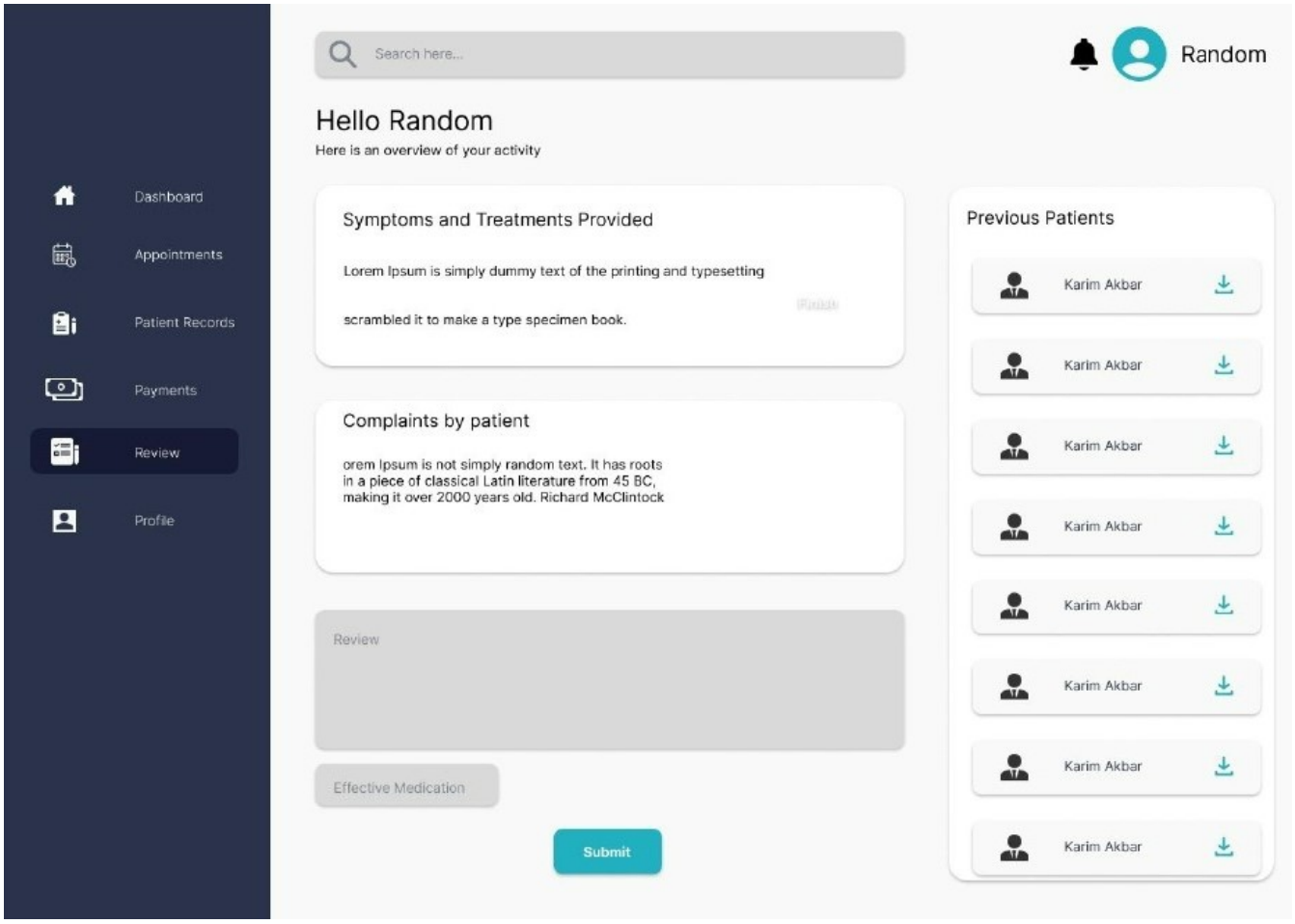

Figure 14: Peer Review Mock UI

4.5 Feedback from Evaluator

- Timeslots for appointments should be shorter to save time if appointments are short. Schedules should be in 2-3 levels. We have decided to follow google calendar view for this.
- Patient health record should not only contain diagnostic info. It should also contain previous doctor's advice for the patients. We have created our ER diagram accordingly to contain doctor's previous advice for each patients.
- Electronic Health Record should contain graphs portraying patient health status, like weight change trend, or kidney Creatinine level trend etc. We have designed our class diagram accordingly to generate graphs from the medical data of patients.

5 ER Diagram

5.1 Full Diagram

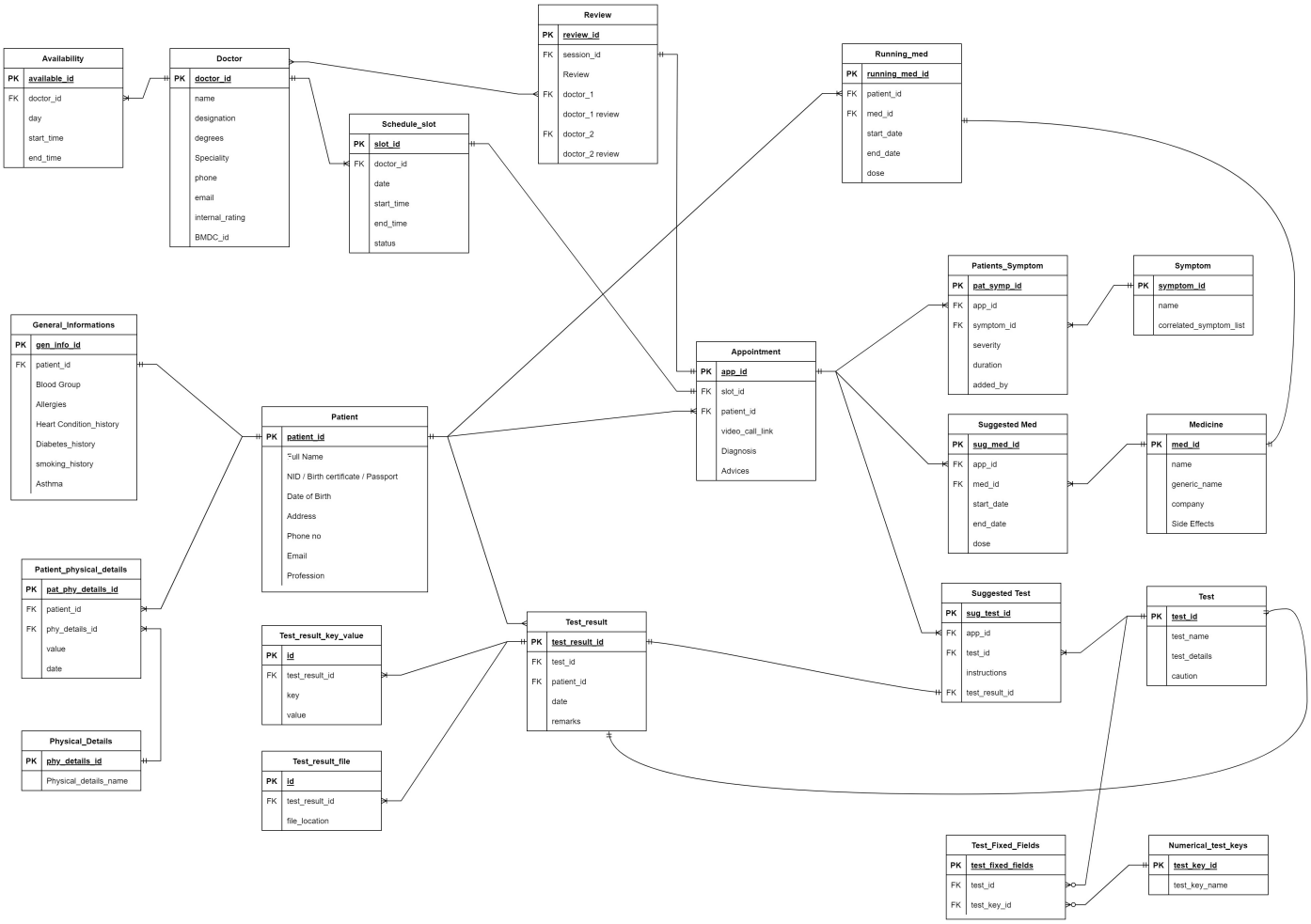

Figure 15: Full ER Diagram

5.2 Patient

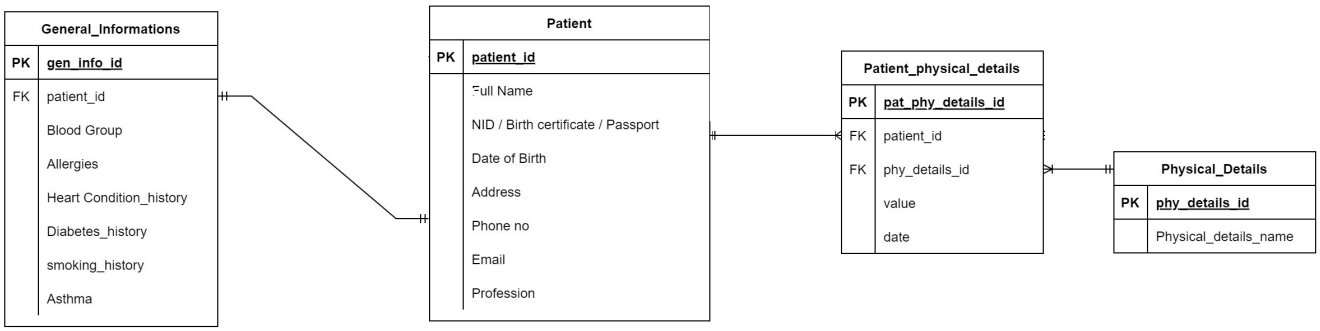

Figure 16: Patient ER Diagram

5.3 Electronic Health Record

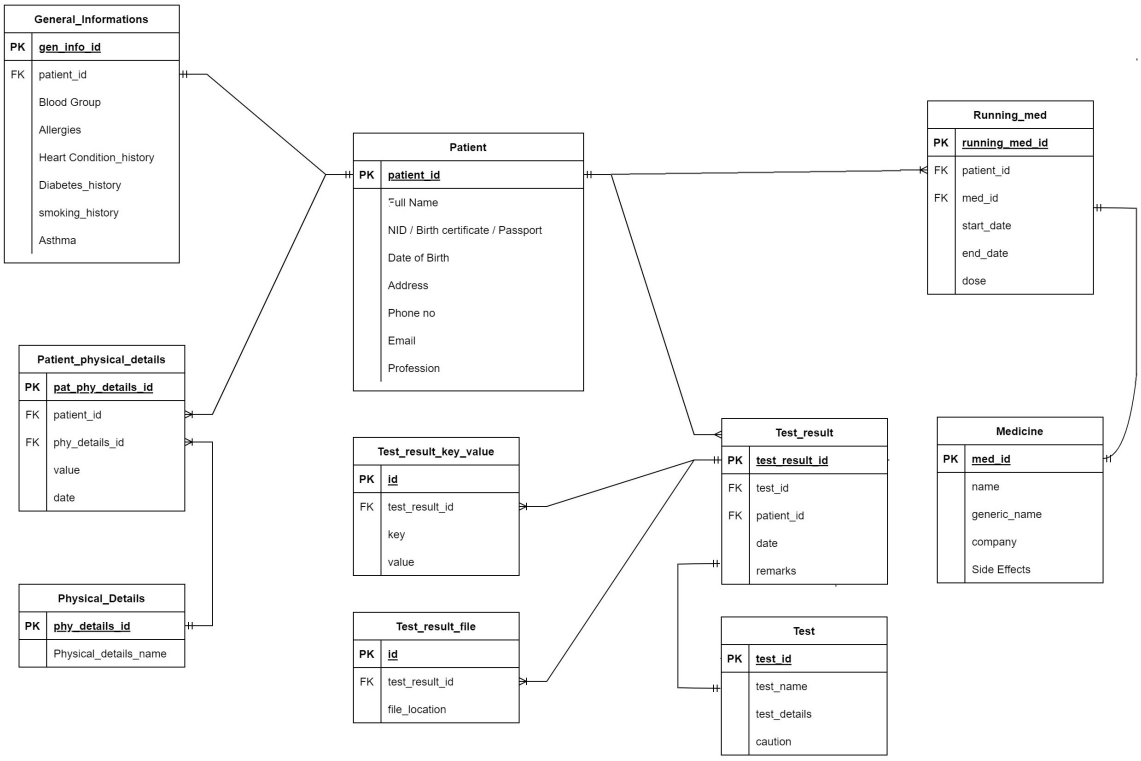

Figure 17: Electronic Health Record ER Diagram

5.4 Doctor

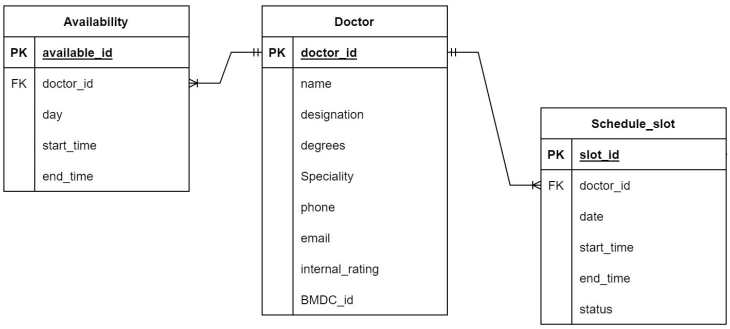

Figure 18: Doctor ER Diagram

5.5 Appointment, Prescription and Review

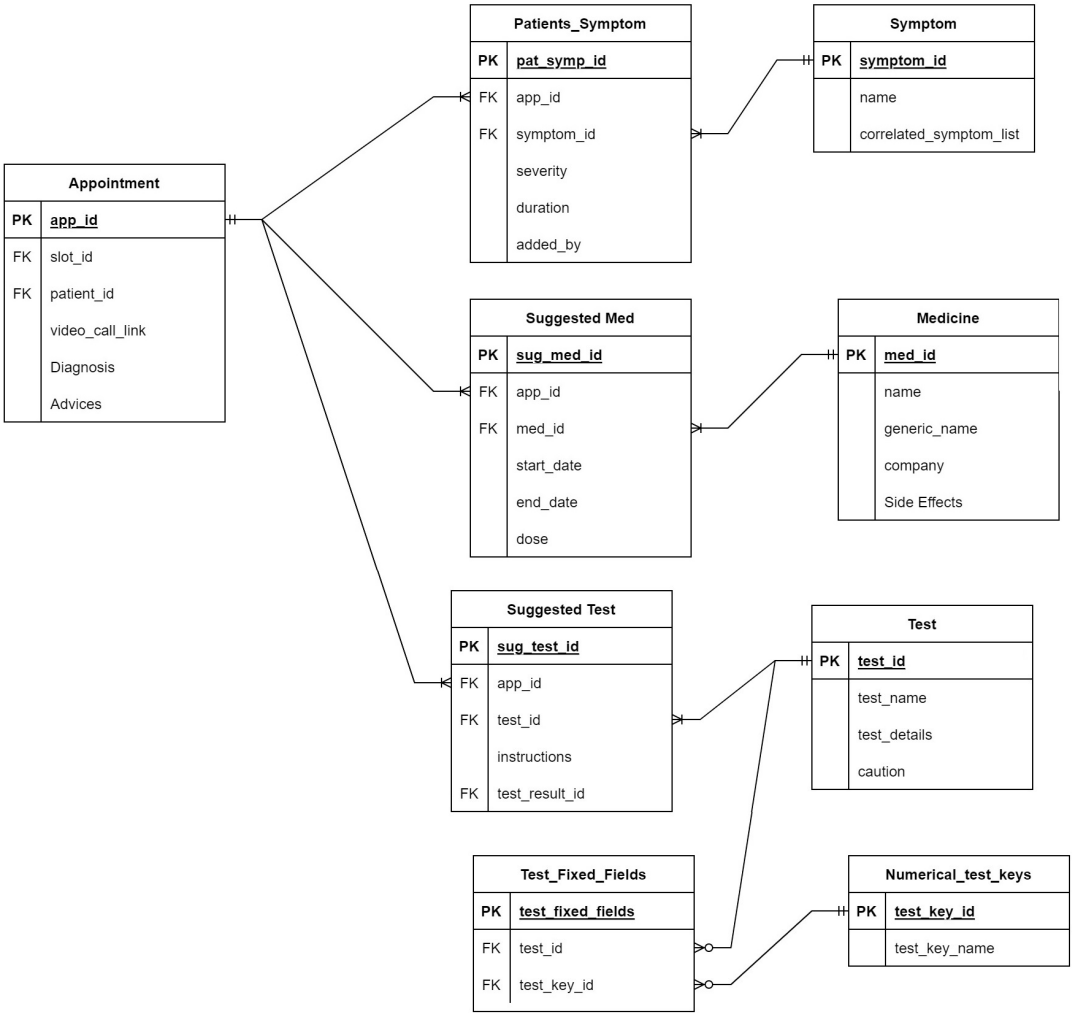

Figure 19: Appointment, Prescription and Review ER Diagram

5.6 Feedback from Evaluator

- We were asked to design the ER diagram with proper notations. We have updated this ER diagram with proper Crow's foot notation.

6 Class Diagram

6.1 The Design Pattern

We have used MVC (Model–View–Controller) design pattern. This is used for developing user interfaces that divide the related program logic into three interconnected elements.

1. Model
2. View
3. Controller

6.2 Patient Registration

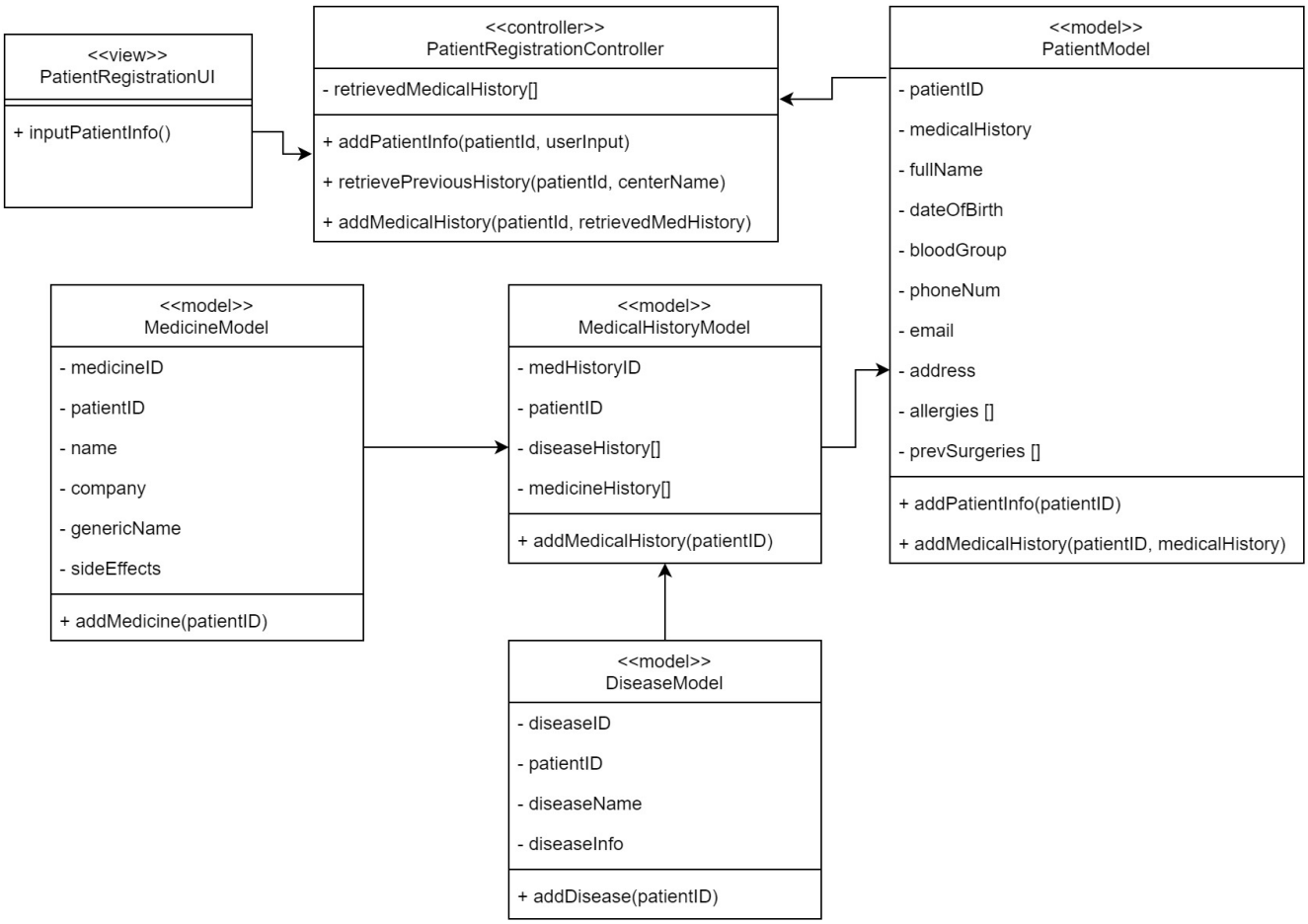

Figure 20: Patient Registration Class Diagram

6.3 Doctor Registration

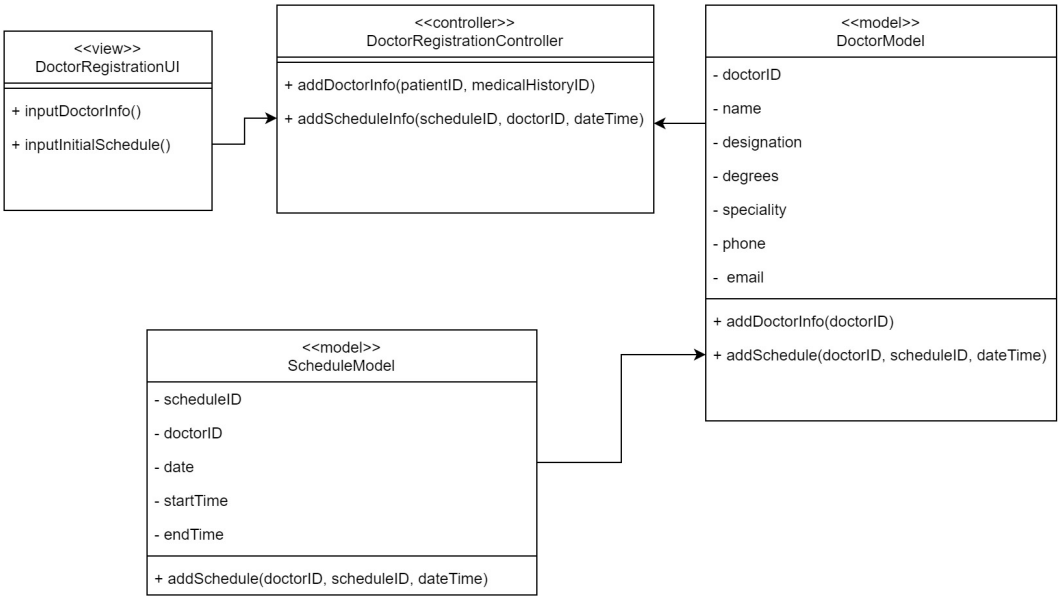

Figure 21: Doctor Registration Class Diagram

6.4 Patient Appointment Booking

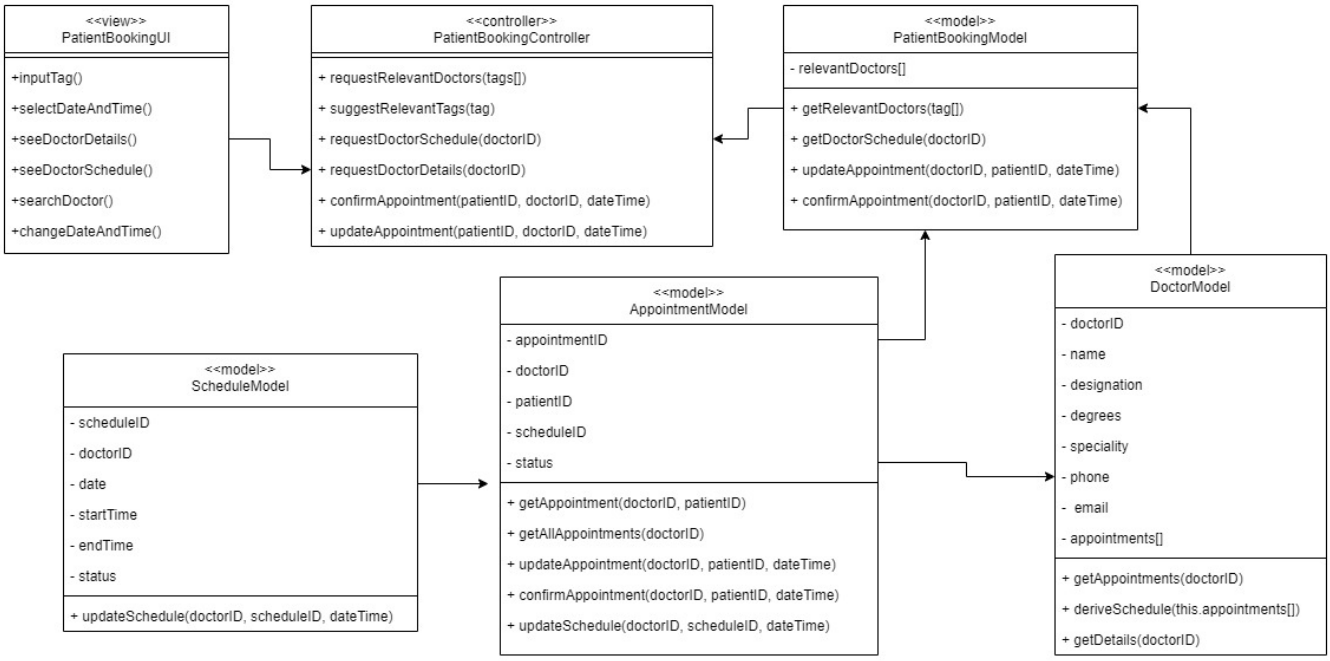

Figure 22: Patient Appointment Booking Class Diagram

6.5 Patient Dashboard

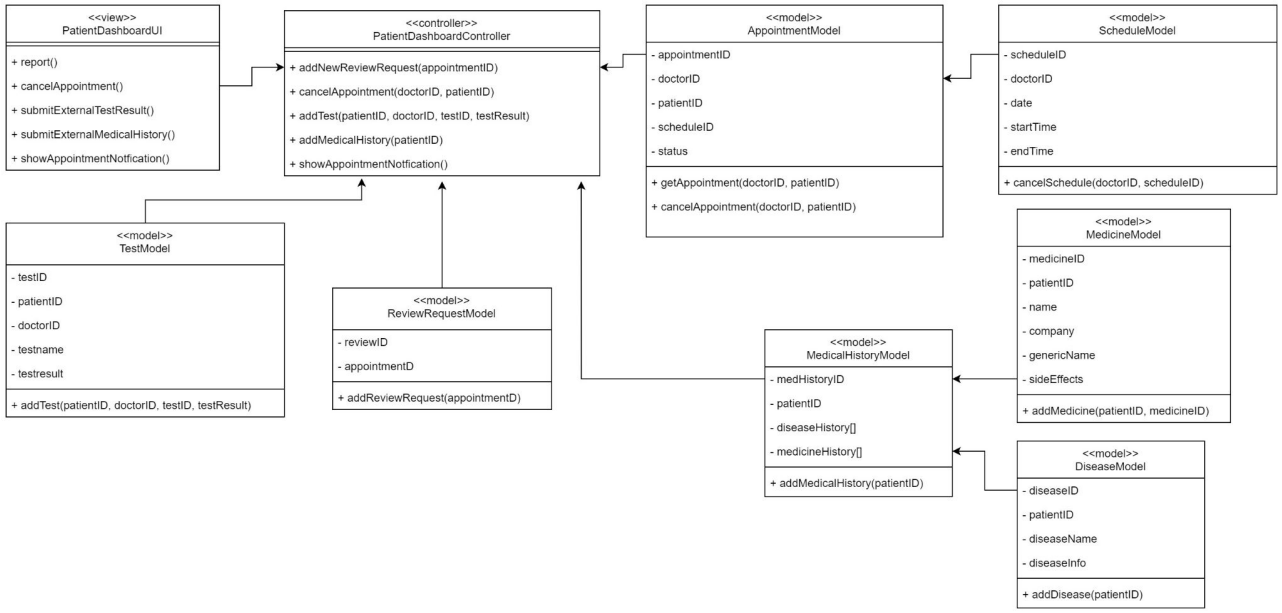

Figure 23: Patient Dashboard Class Diagram

6.6 Doctor Dashboard

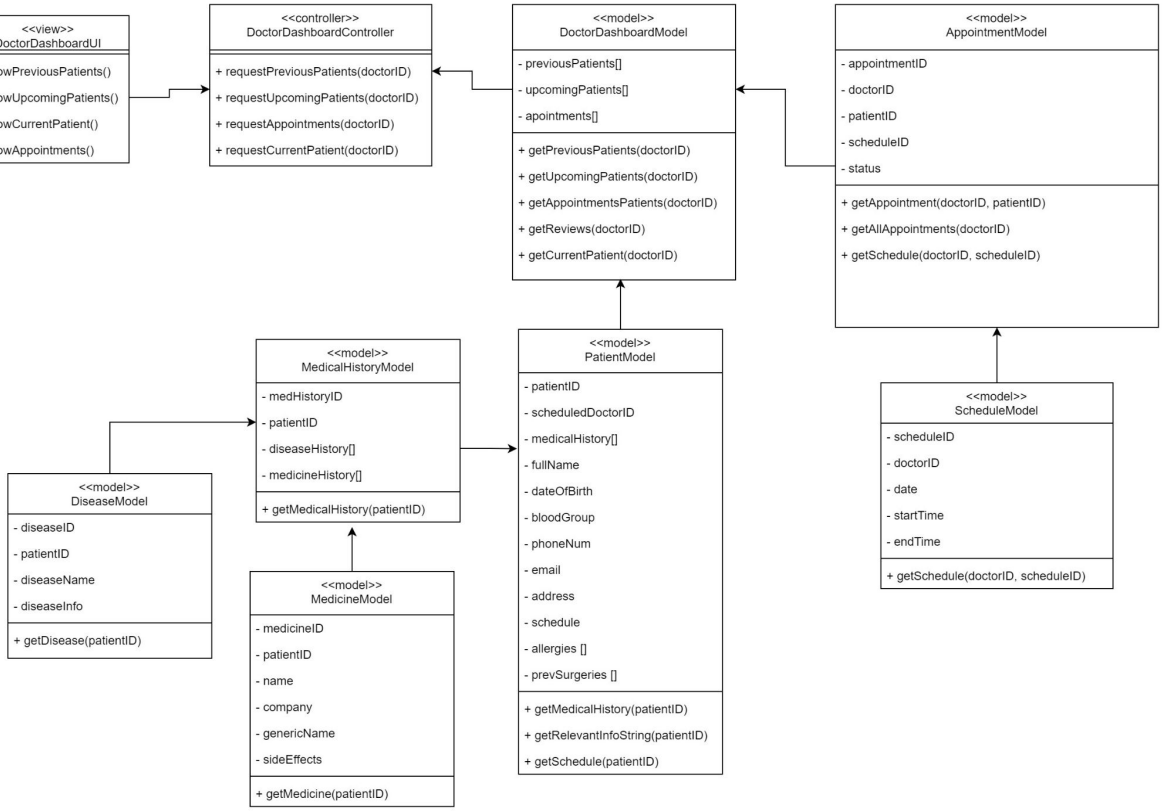

Figure 24: Doctor Dashboard Class Diagram

6.7 Peer Review

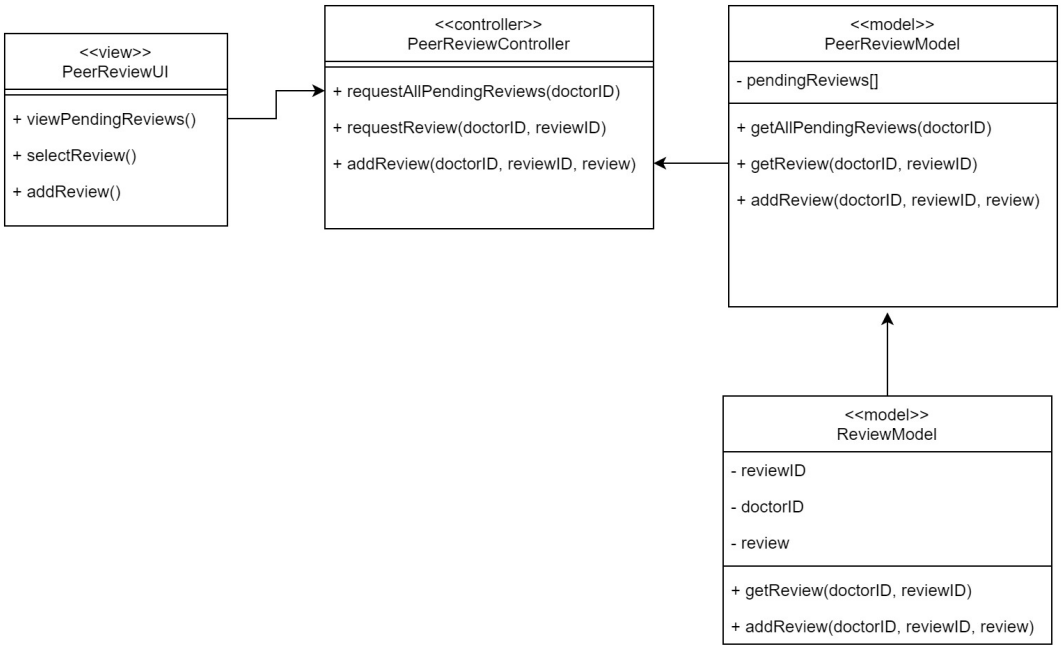

Figure 25: Peer Review Class Diagram

6.8 Rescheduler

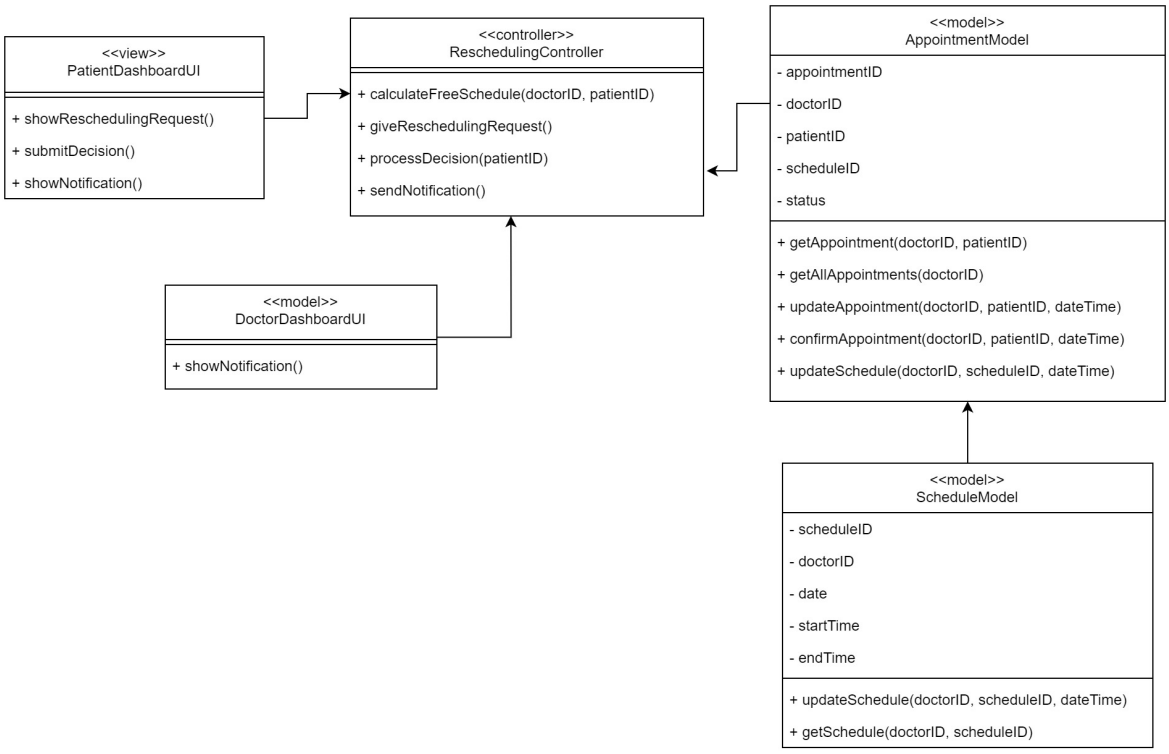

Figure 26: Rescheduler Class Diagram

6.9 Video Session

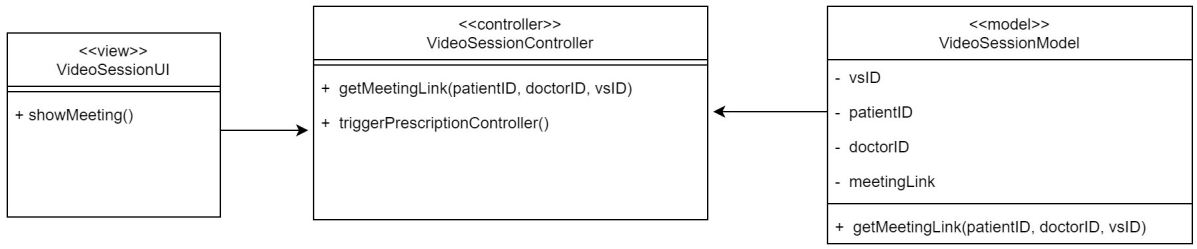

Figure 27: Video Session Class Diagram

6.10 Prescription

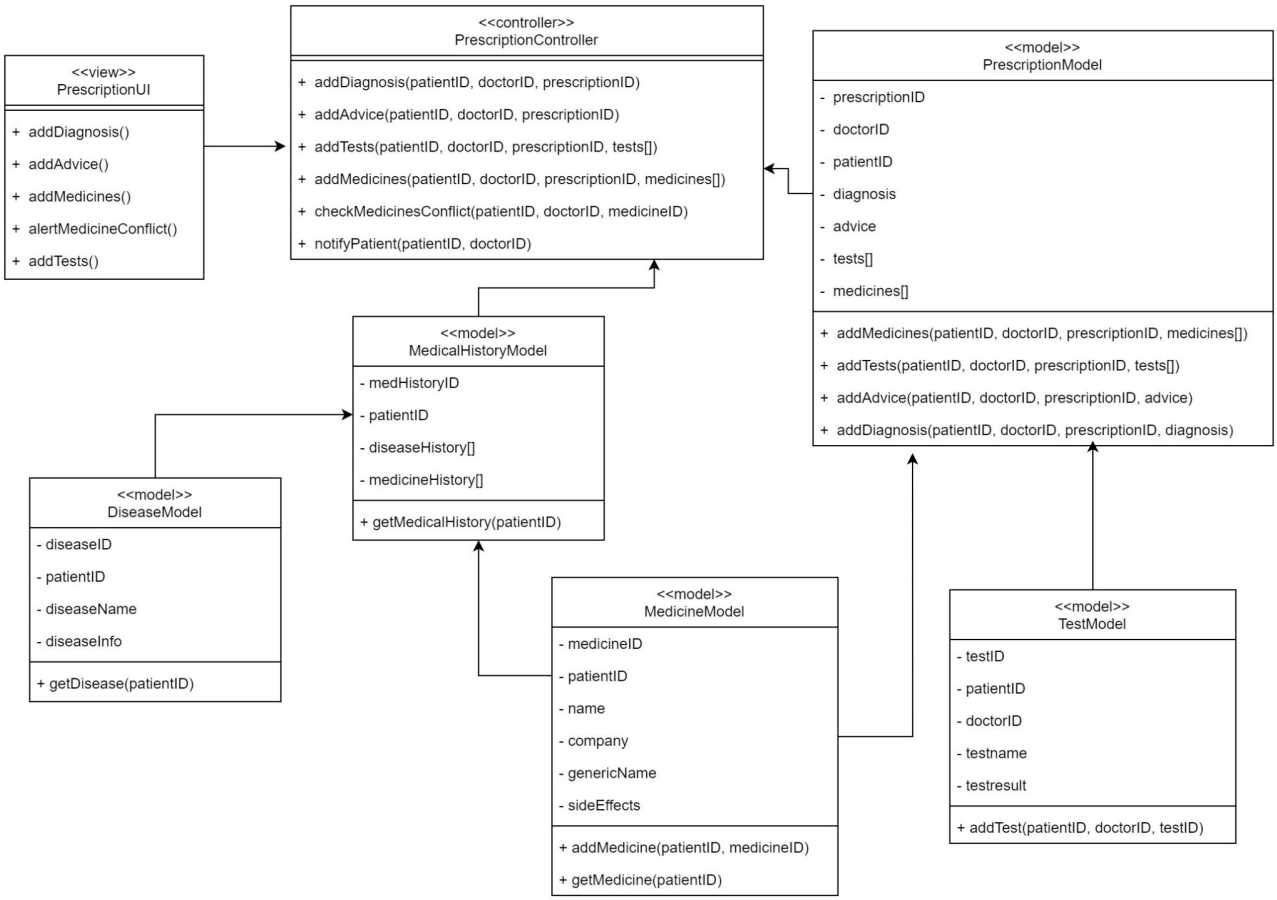

Figure 28: Prescription Class Diagram

6.11 Electronic Health Record

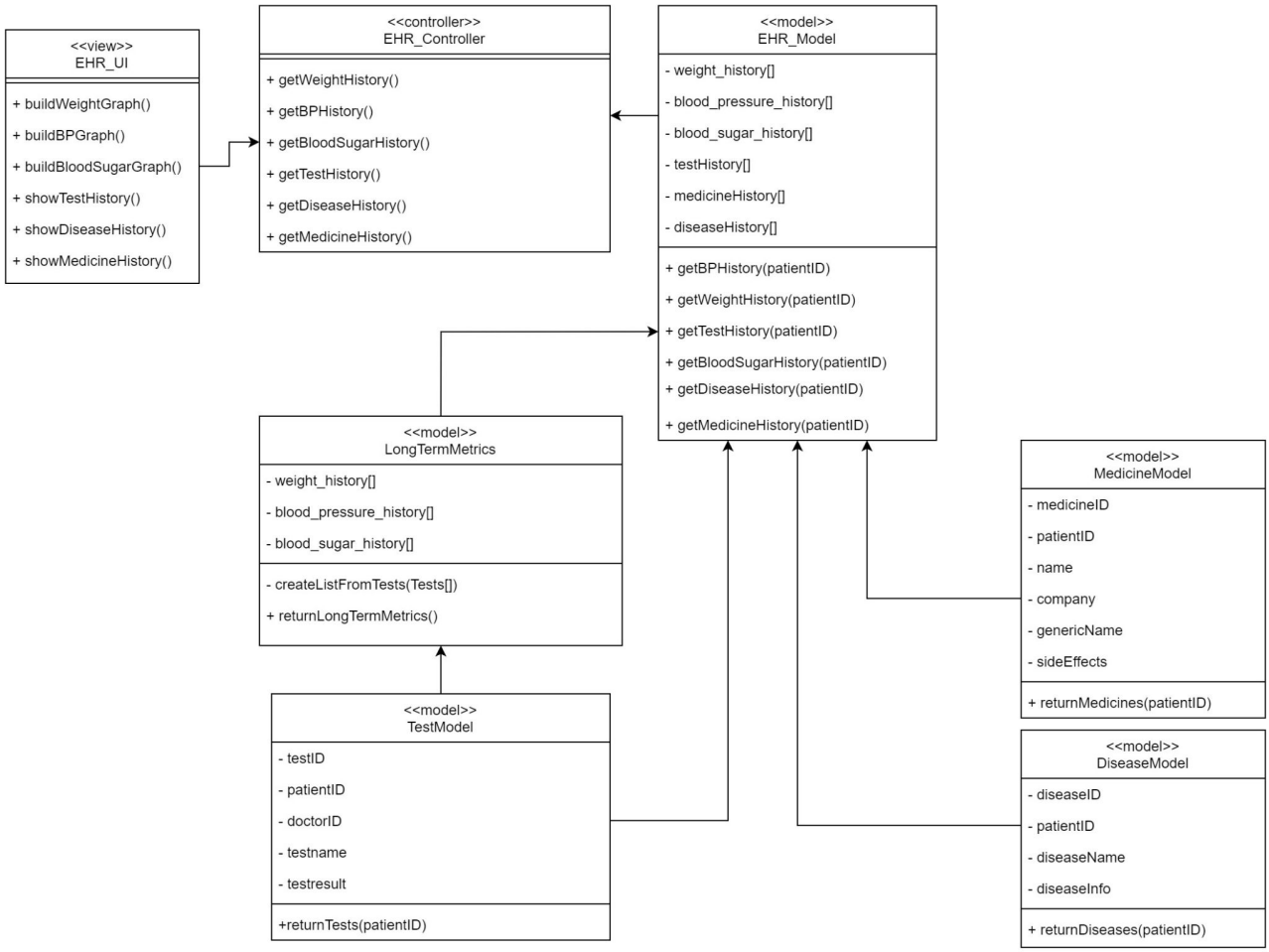

Figure 29: Electronic Health Record Class Diagram

6.12 Feedback from Evaluator

- The ScheduleModel in Patient Appointment Booking needed a status variable to keep track if the schedule was free or booked. We have added the variable accordingly to the class diagram above.

7 State Diagram

7.1 Booking

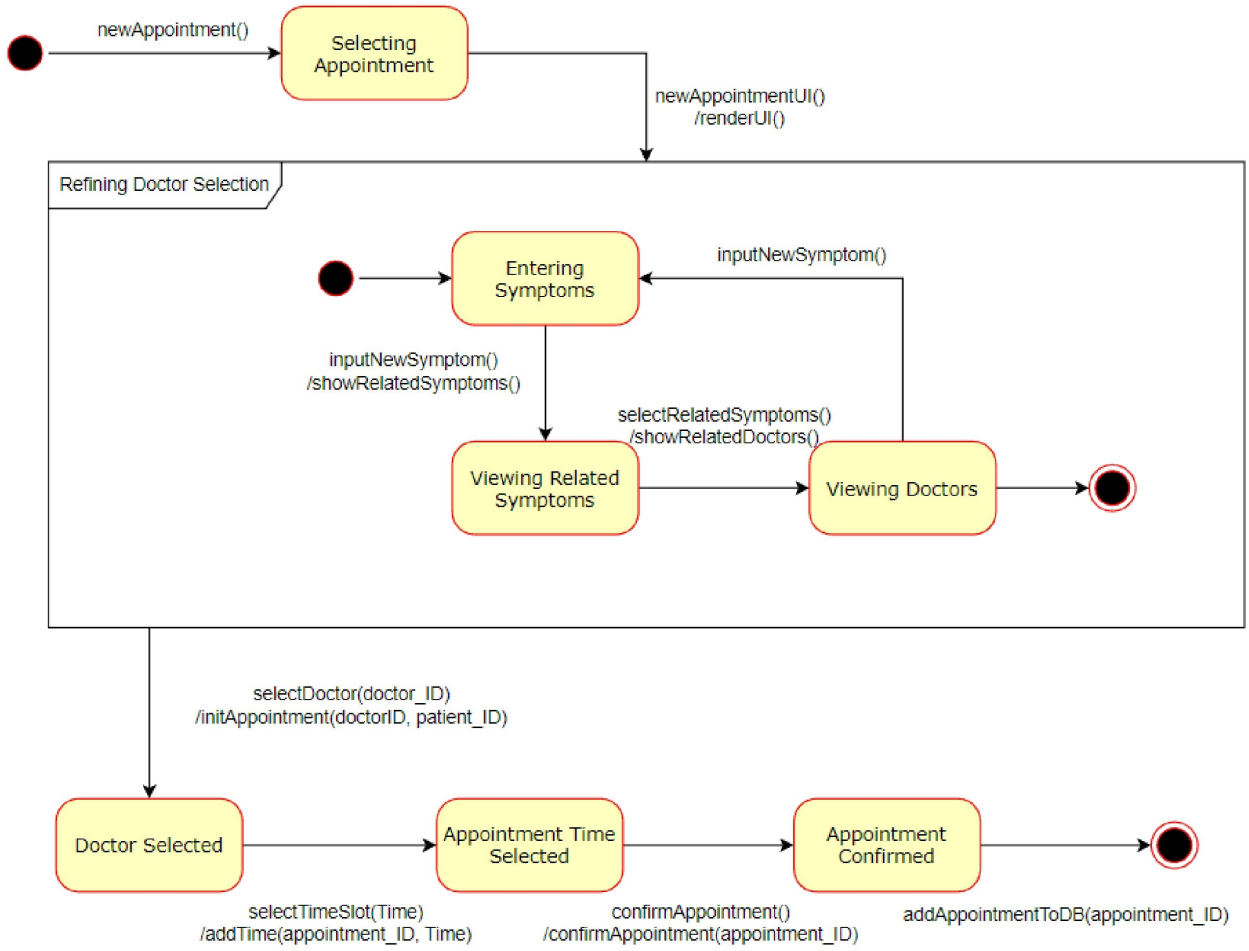

Figure 30: Booking State Diagram

7.2 Appointment Patient Perspective

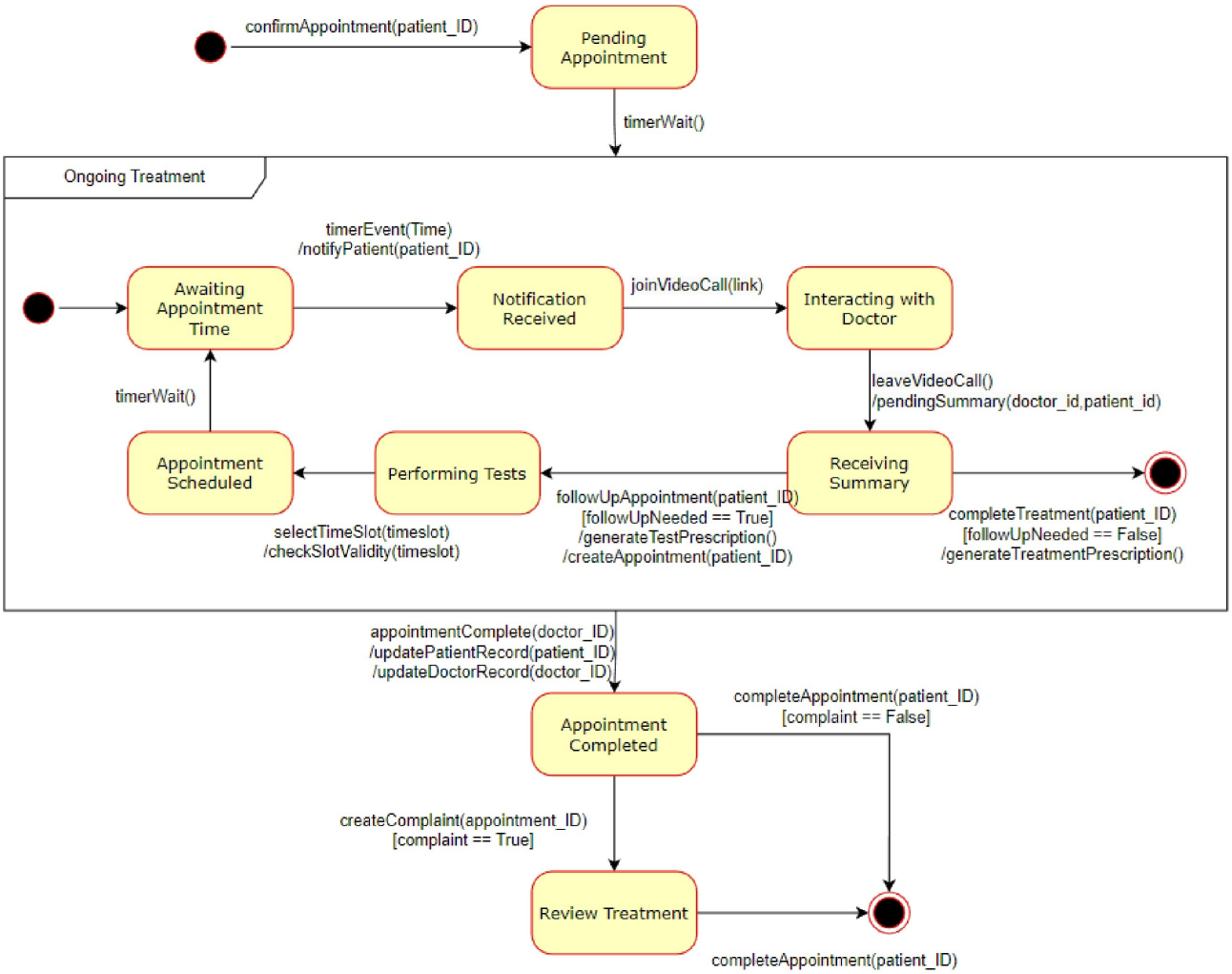

Figure 31: Appointment Patient Perspective State Diagram

7.3 Appointment Doctor Perspective

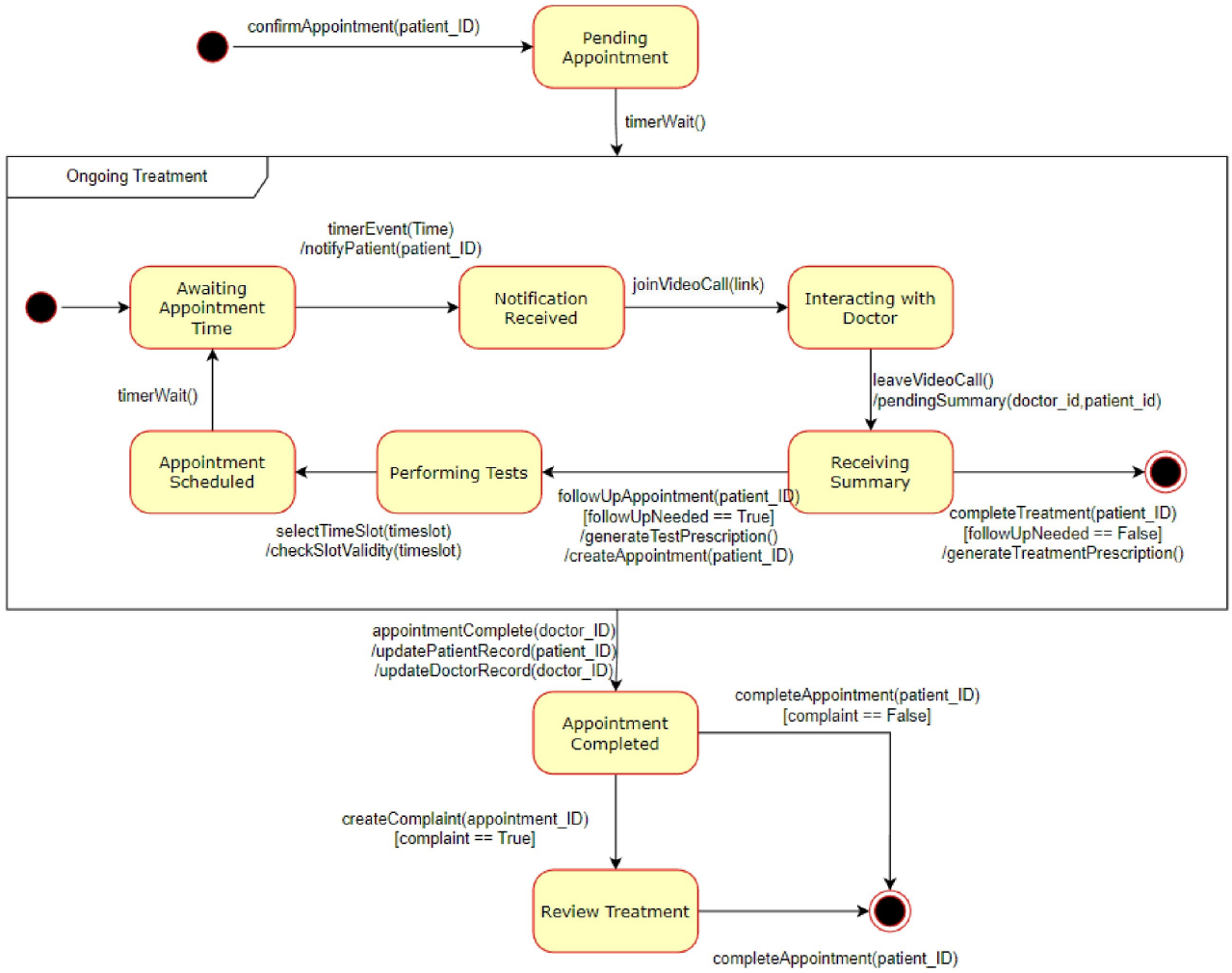

Figure 32: Appointment Doctor Perspective State Diagram

7.4 Rescheduling

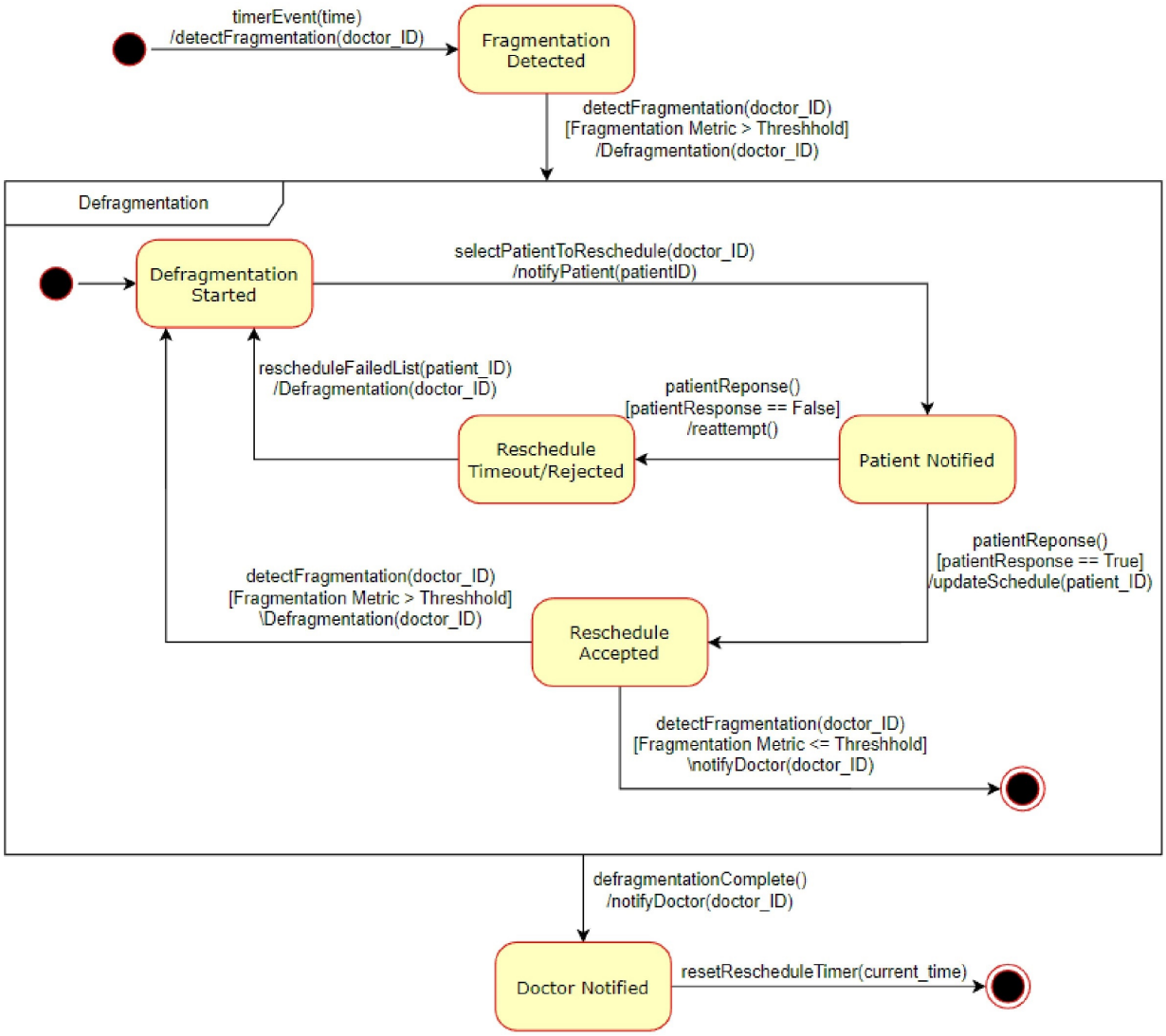

Figure 33: Rescheduling State Diagram

Evaluator's Feedback: Doctors should have an option to trigger rescheduling manually, either by notifying a possible delay on his part or by cancelling entire spans of appointments on any given date.

Patients could be prompted during the initial appointment time selection phase to select and rank multiple time slots instead of one selection. This information can be utilized during rescheduling, if necessary. These features have been acknowledged in the implementation phase.

7.5 Peer Review

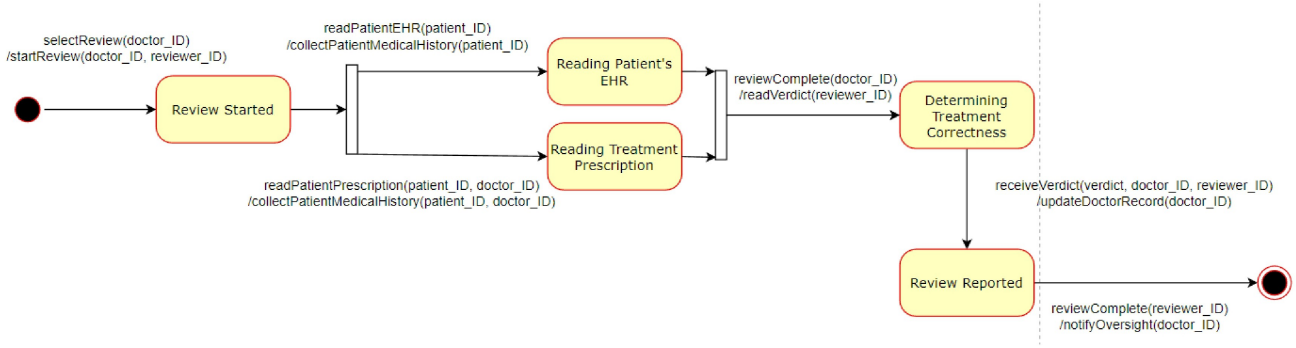

Figure 34: Peer Review State Diagram

8 Sequence Diagram

8.1 Booking

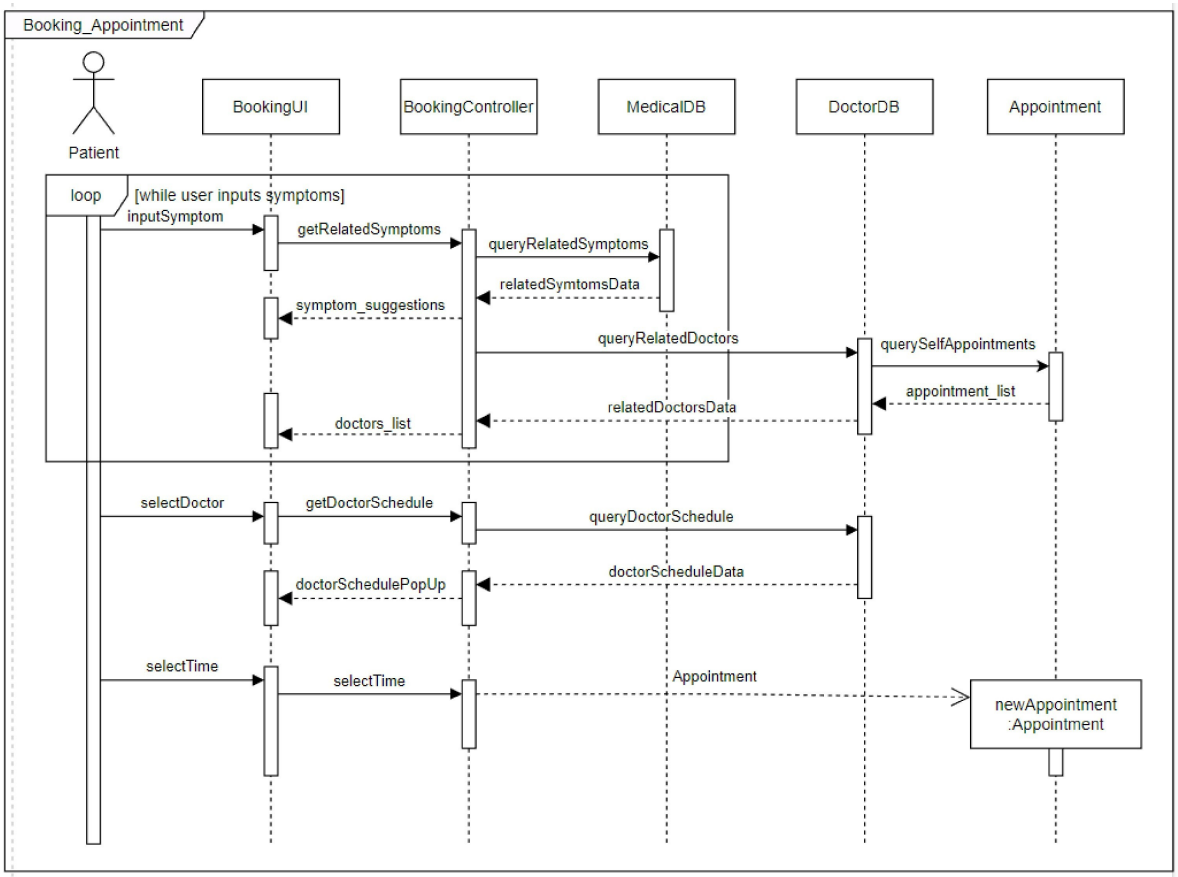

Figure 35: Booking Sequence Diagram

8.2 Appointment Patient Perspective

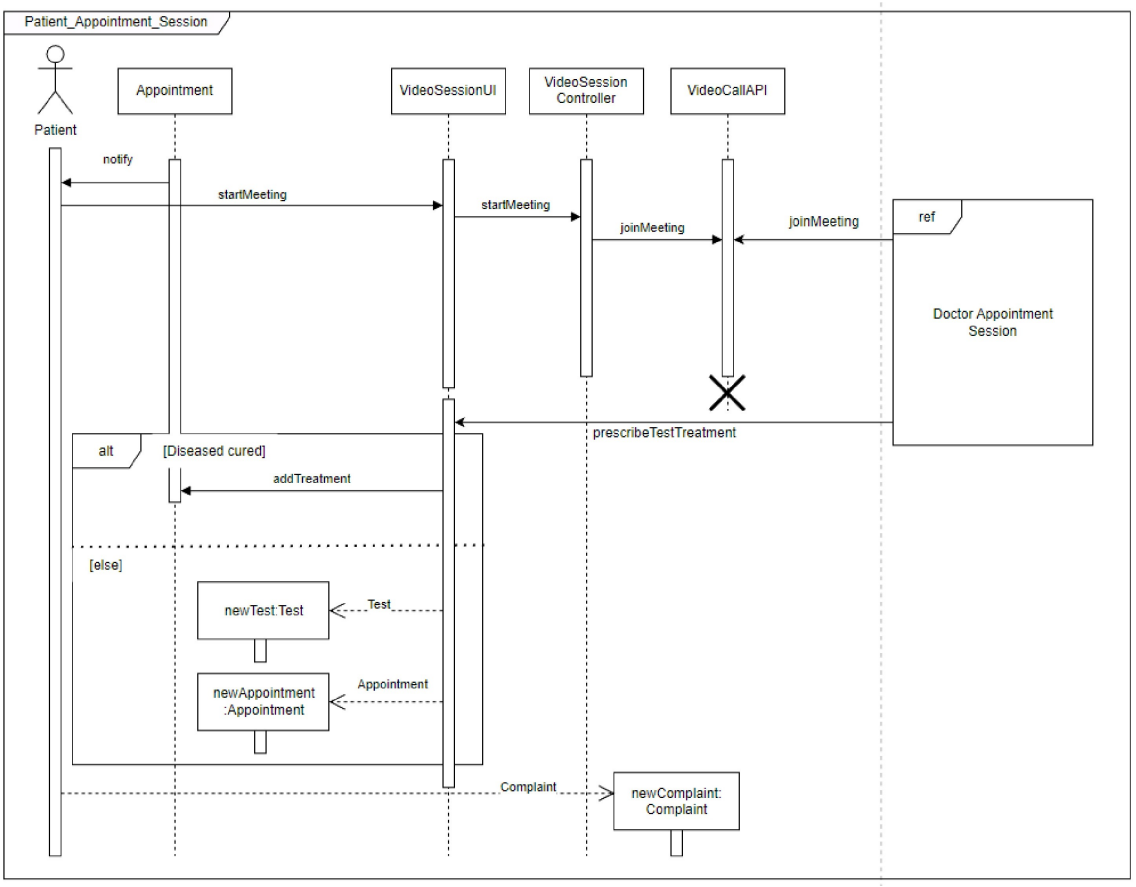

Figure 36: Appointment Patient Perspective Sequence Diagram

8.3 Appointment Doctor Perspective

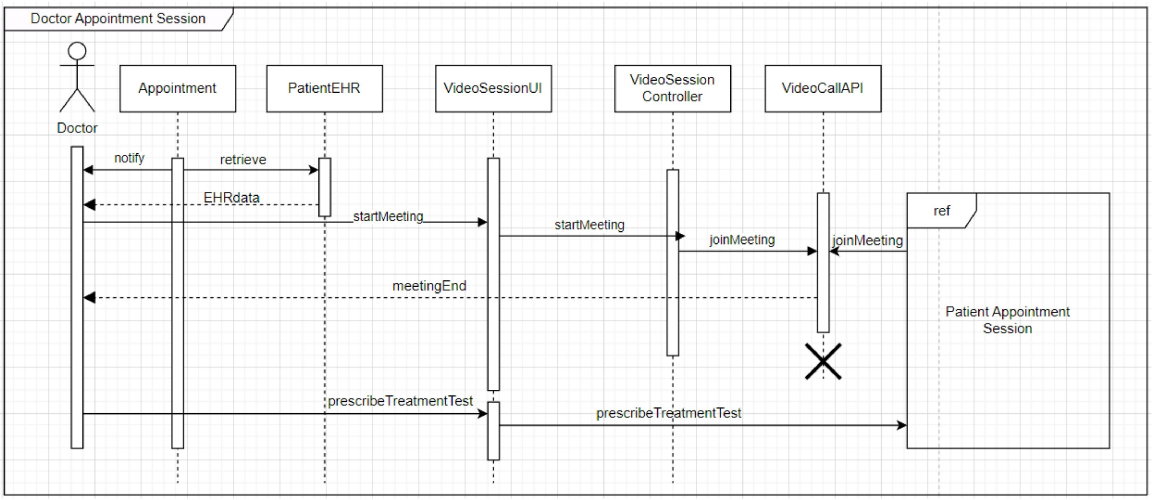

Figure 37: Appointment Doctor Perspective Sequence Diagram

Evaluator's Feedback: When presenting the patient's EHR, the information should be dynamically reordered based on how relevant they are for the patient's current symptoms.

When the doctor prescribes medicine, the system should clearly notify the doctor if the prescribed medicine conflicts with any of the patient's existing medication.

The diagnosis of each appointment should be a separate entity, not an attribute of Appointment. That way pre-existing conditions or past diseases of a newly registering users can be added to the database directly. Regarding the past medical history of a newly registered user: some medical tests are fairly indicative of the underlying disease but if the it is not, the user should be prompted to enter the disease manually.

8.4 Peer Review

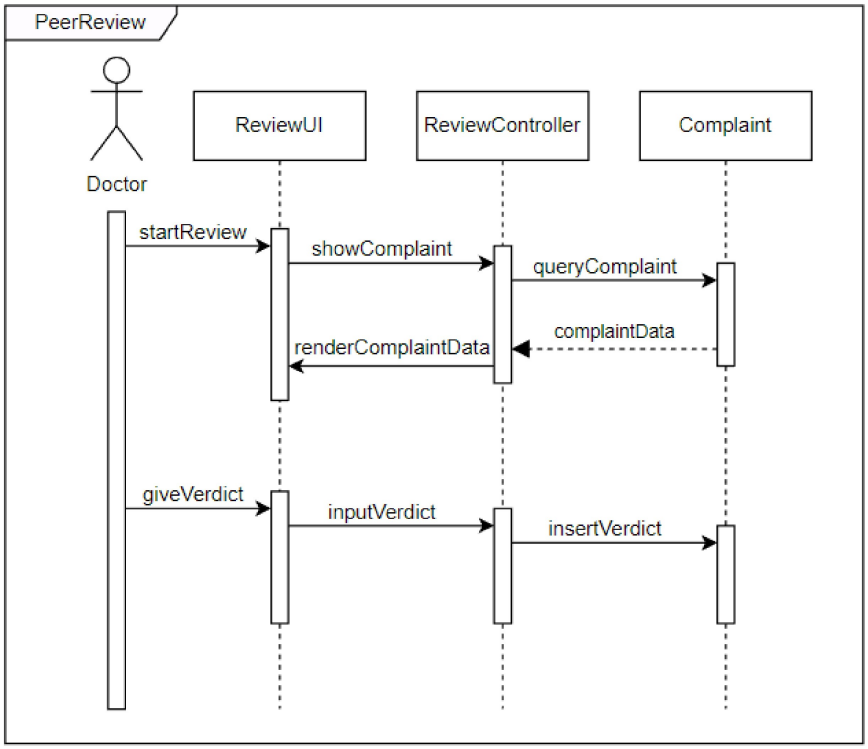

Figure 38: Peer Review Sequence Diagram

8.5 Electronic Health Record

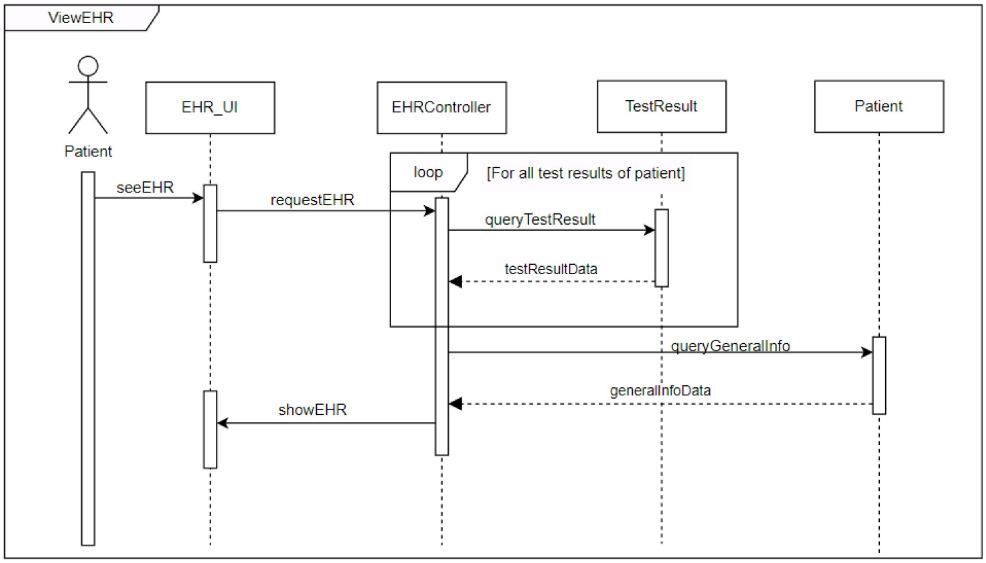

Figure 39: Electronic Health Record Sequence Diagram

8.6 Manual Test Report

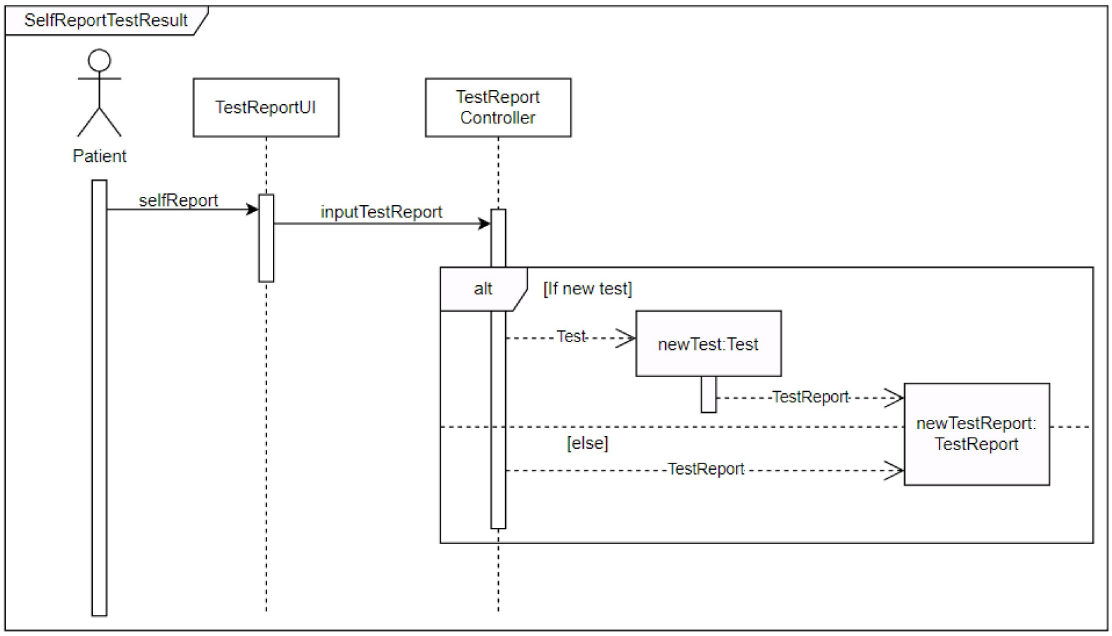

Figure 40: Manual Test Report Sequence Diagram

Evaluator's Feedback: Select criteria such as weight, blood sugar level, heart rate etc. should be separate from other, more serious tests. They could be displayed as long running graphs in the patient's EHR.

9 Collaboration Diagram

9.1 Appointment Booking

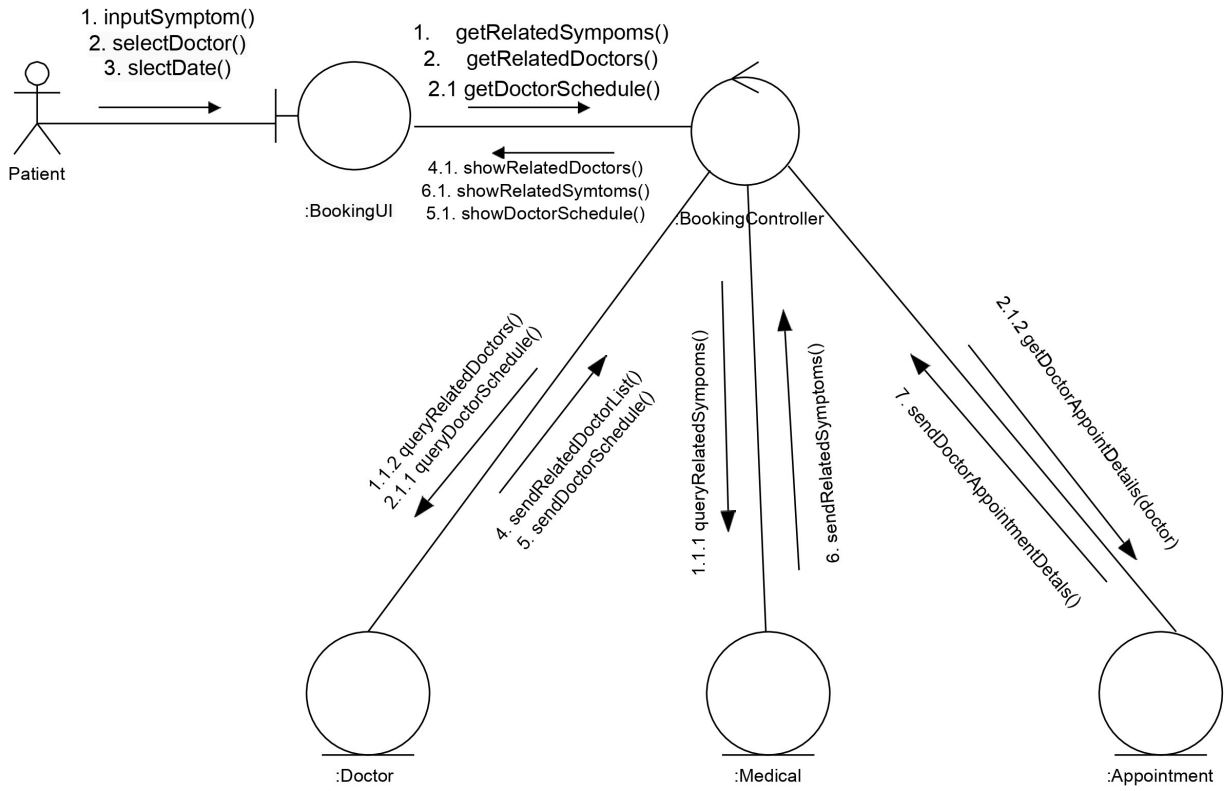

Figure 41: Appointment Booking Collaboration Diagram

9.2 Review

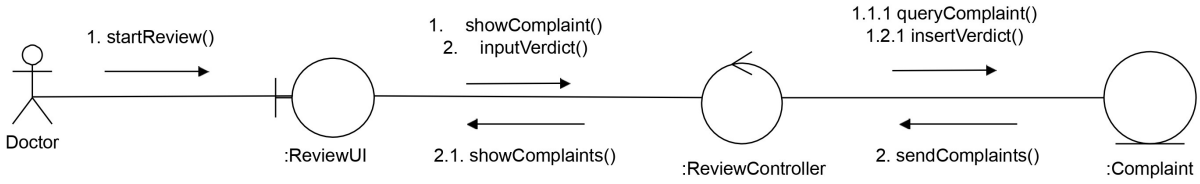

Figure 42: Review Collaboration Diagram

9.3 Doctor Appointment Session

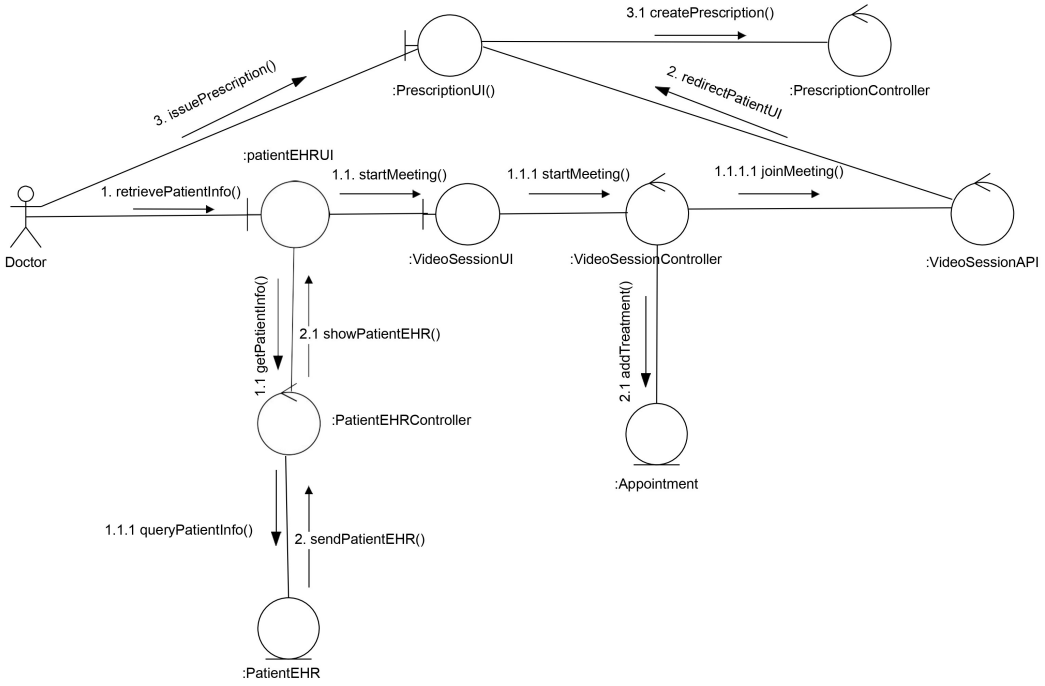

Figure 43: Doctor Appointment Session Collaboration Diagram

9.4 Patient Appointment Session

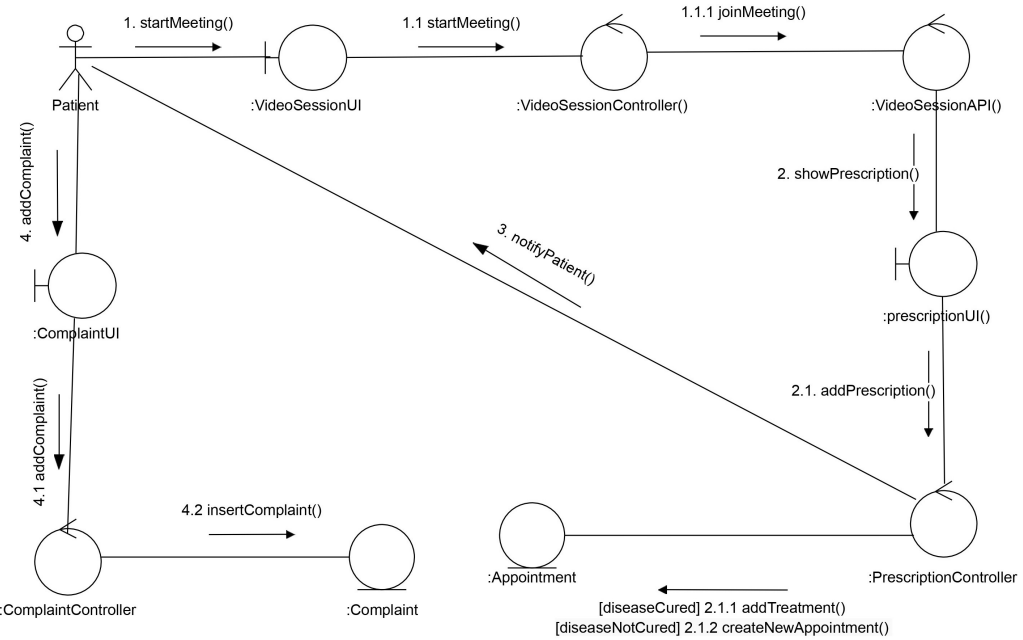

Figure 44: Patient Appointment Session Collaboration Diagram

9.5 Electronic Health Record

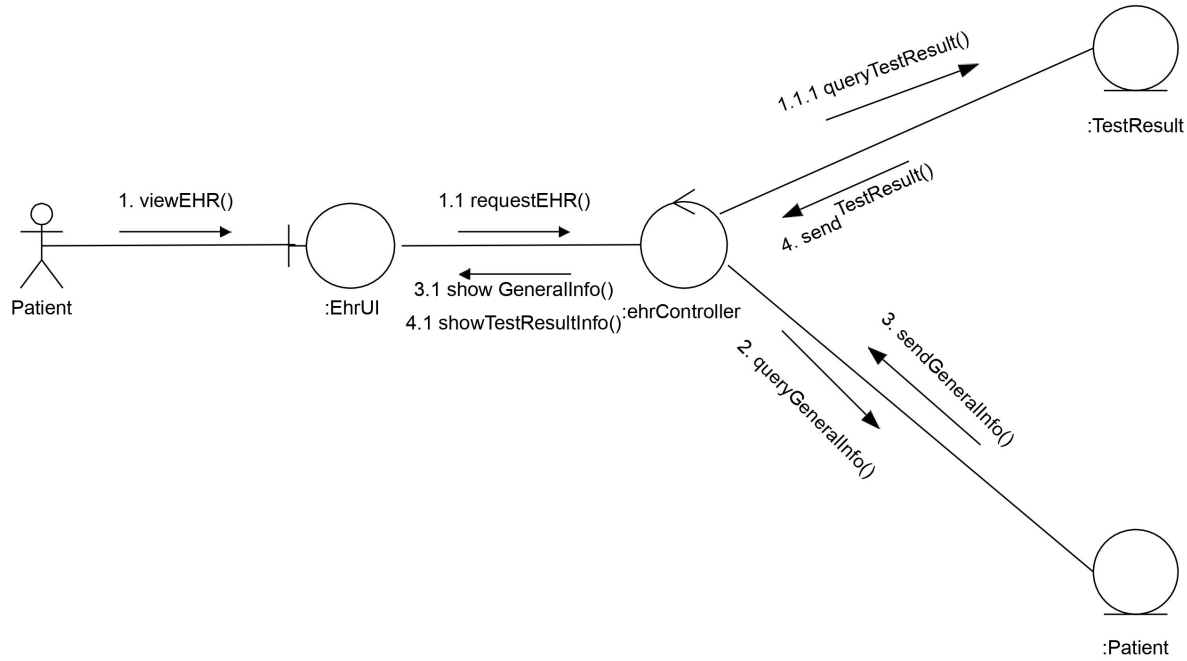

Figure 45: Electronic Health Record Collaboration Diagram

9.6 Test Report

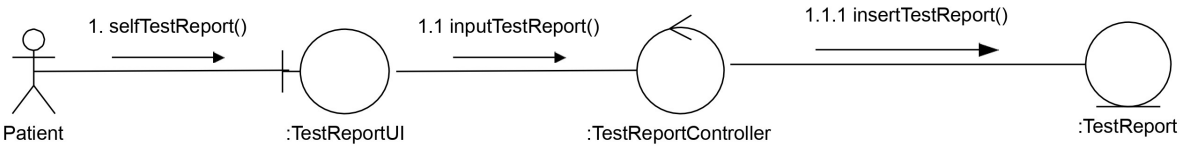

Figure 46: Test Report Collaboration Diagram

10 Partial Software Implementation

10.1 Backend and CI/CD

10.1.1 Database Configuration

Database: MongoDB atlas

Version: 5.0.15

Region: GCP / Mumbai (asia-south1)

Cluster Tier: M0 Sandbox (General)

Type: Replica Set - 3 nodes

10.1.2 Database

The central database program we used is MongoDB which is a NoSQL database program. MongoDB uses JSON-like documents with optional schema. Since we require a new user's medical history, our database needs to support direct information extraction from different diagnostics centers which may use a wide variety of internal database designs. This naturally means that all communication must occur via API calls which use JSON. MongoDB makes this communication seamless since it already uses a JSON-like document structure. Moreover, were we to use a traditional SQL database program, objects like 'Test', 'Test Report', 'Disease', 'Medicine' etc. would have many different attributes, which would have required one or more separate attribute tables with one-to-many relationship with the main table. Operations like displaying the patients EHR would then become extremely slow owing to the large number of join operations it would require for each 'Test Report', 'Disease' etc. of the Electronic Health Record.

10.1.3 Collections in the Database

1. doctor
2. medicine
3. patient
4. session
5. symptoms
6. test_result

10.1.4 Backend

Backend Framework: We have chosen to use FastAPI as the primary web framework for building the platform's APIs. FastAPI is a modern, high-performance web framework for building APIs with Python 3.7+ that is based on standard Python type hints. Its speed and efficiency make it an ideal choice for building our telemedicine platform that requires fast and reliable responses.

In addition to FastAPI, we have also utilized Pydantic models as an ORM and data validation tool. Pydantic is a data validation and settings management library that uses Python's type annotations to provide fast and simple data validation and serialization capabilities.

Backend API Endpoints:

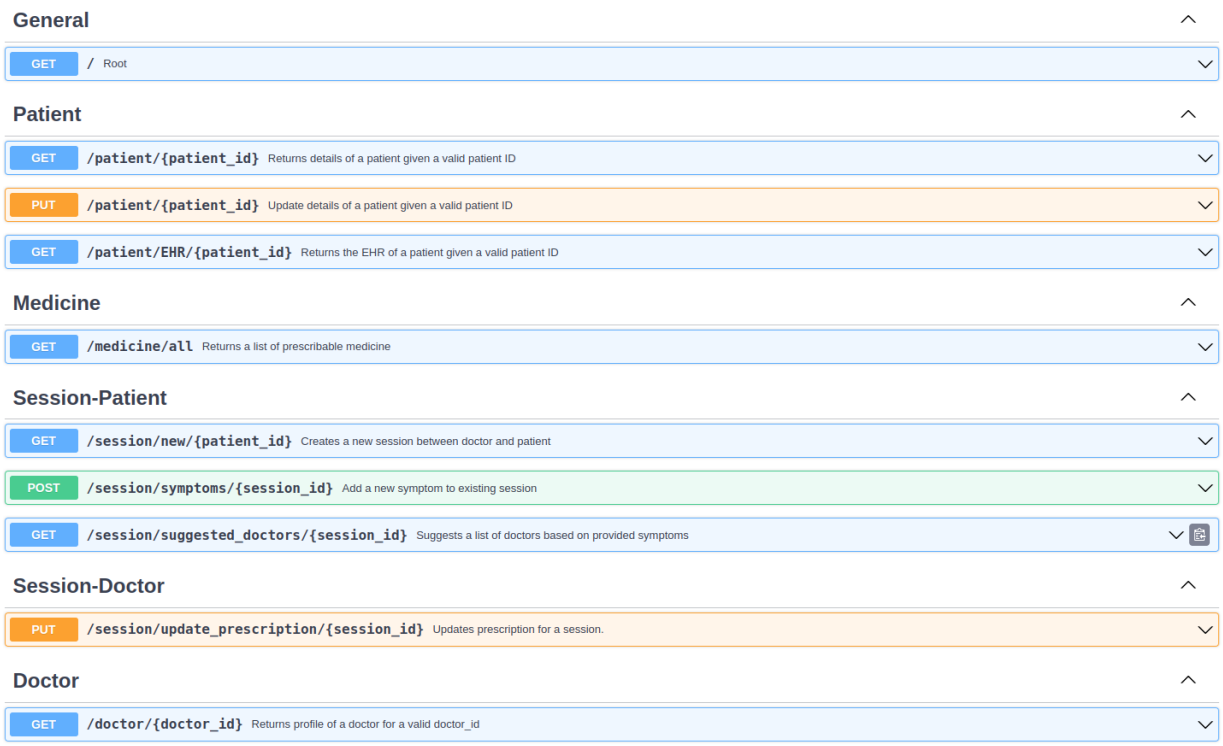

Figure 47: Backend API Endpoints

10.1.5 Deployment

MobileDoc's backend was deployed using the reliable and scalable infrastructure of Google Cloud Platform. The development process was fully automated using cloud build triggers (cloudbuild.yaml), providing a streamlined and efficient workflow. This automated process allowed for any push on the main branch of the repository to trigger the App engine build (app.yaml), resulting in a rapid and seamless deployment process.

Furthermore, MobileDoc has been configured to take advantage of App engine's advanced deployment features such as rollback and traffic splitting between versions. This allows for

greater flexibility and control over the deployment process, ensuring that updates and changes are carefully managed and rolled out to users in a controlled and safe manner. The use of App engine's deployment features provides an additional layer of security and stability to the platform, ensuring a seamless user experience and maximum uptime for critical healthcare services. The implementation of these advanced deployment features demonstrates MobileDoc's commitment to delivering a reliable and high-quality telemedicine solution to patients and healthcare providers alike.

10.1.6 Unit Test

Four unit tests were added to check the status code and completion status of the business logic for several endpoints, ensuring the quality and reliability of the platform. These unit tests were fully automated using a Github action workflow, which triggers the tests in the background for any push or pull request on the main or dev branch, providing immediate feedback.

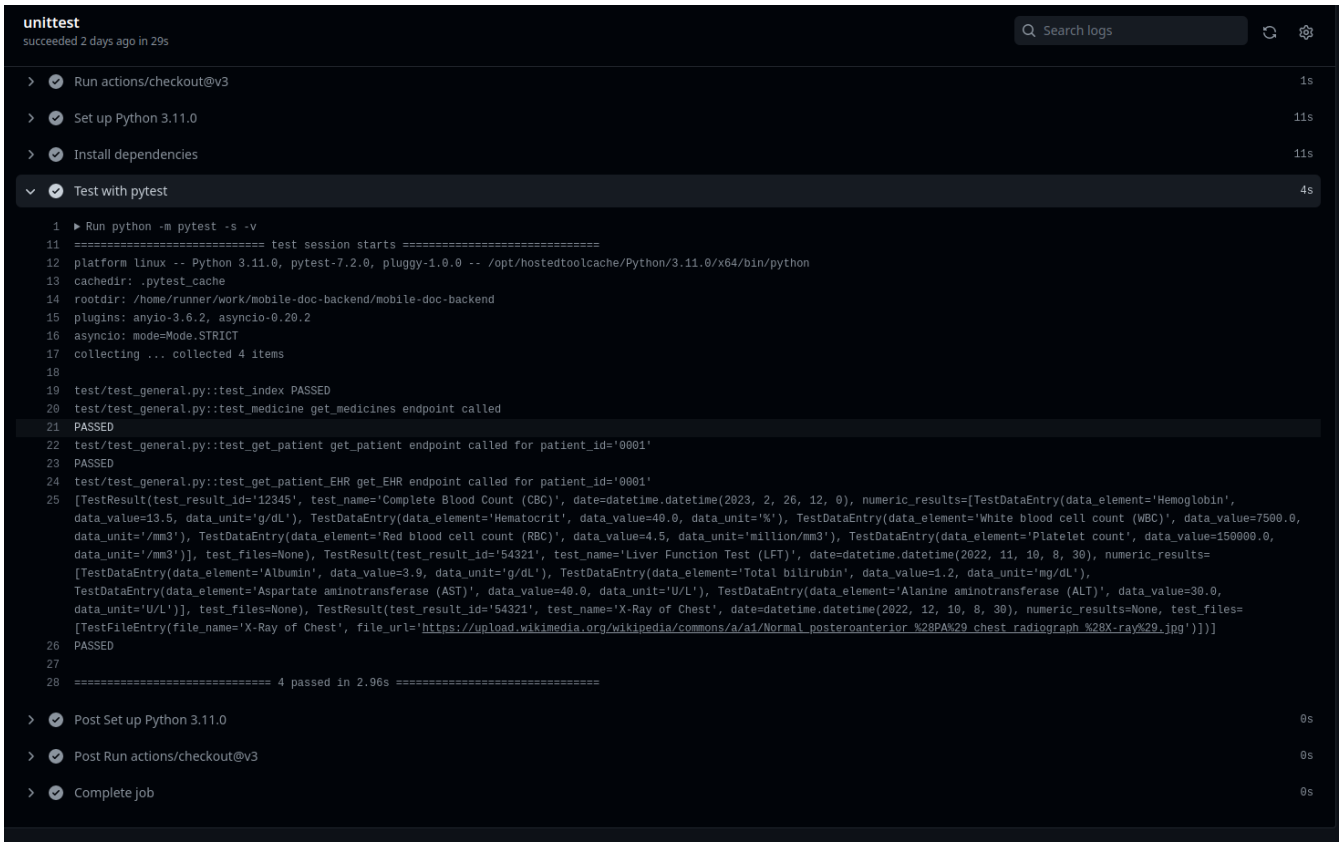

Figure 48: Unit Test

10.2 User Interface

10.2.1 Patient's Symptom Search

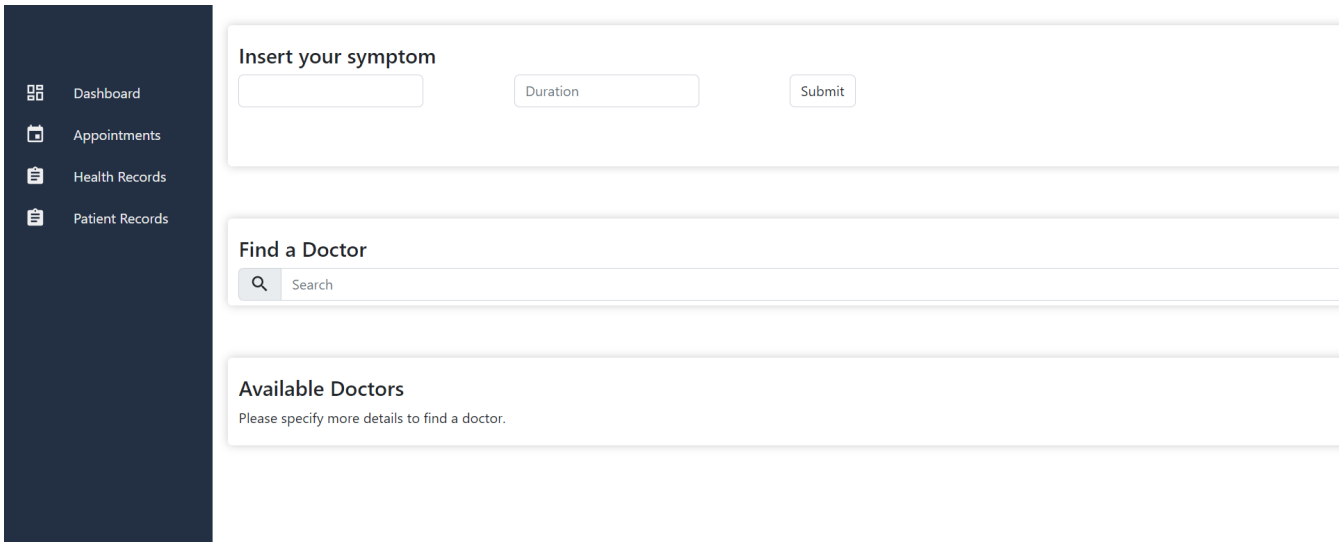

Figure 49: Patient's Symptom Search

10.2.2 Suggested Symptoms

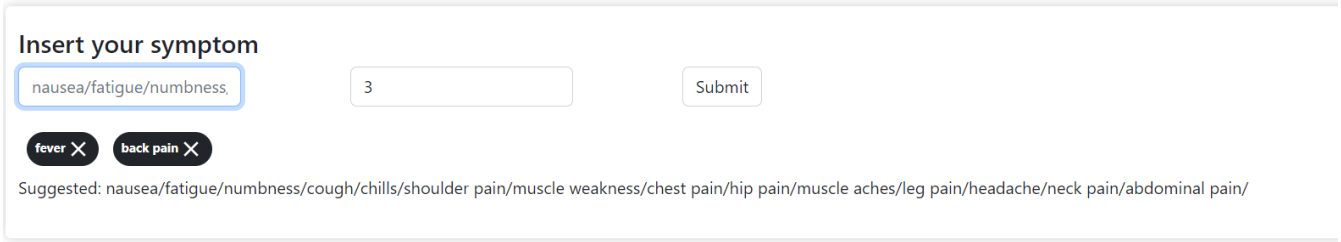

Figure 50: Suggested Symptoms

10.2.3 Suggested Doctors

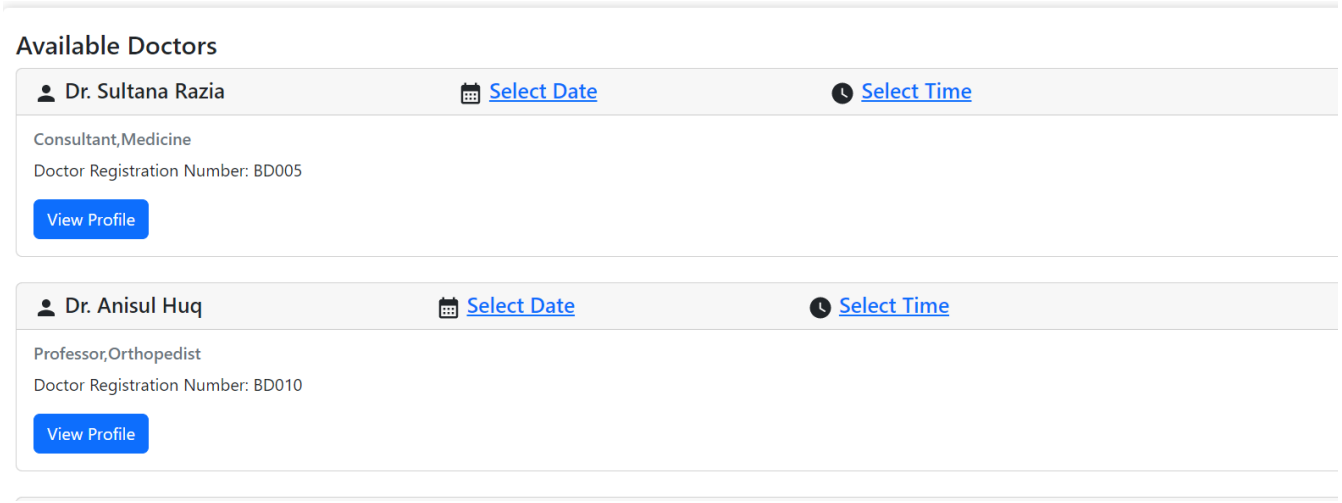

Figure 51: Suggested Doctors

10.2.4 Patient History

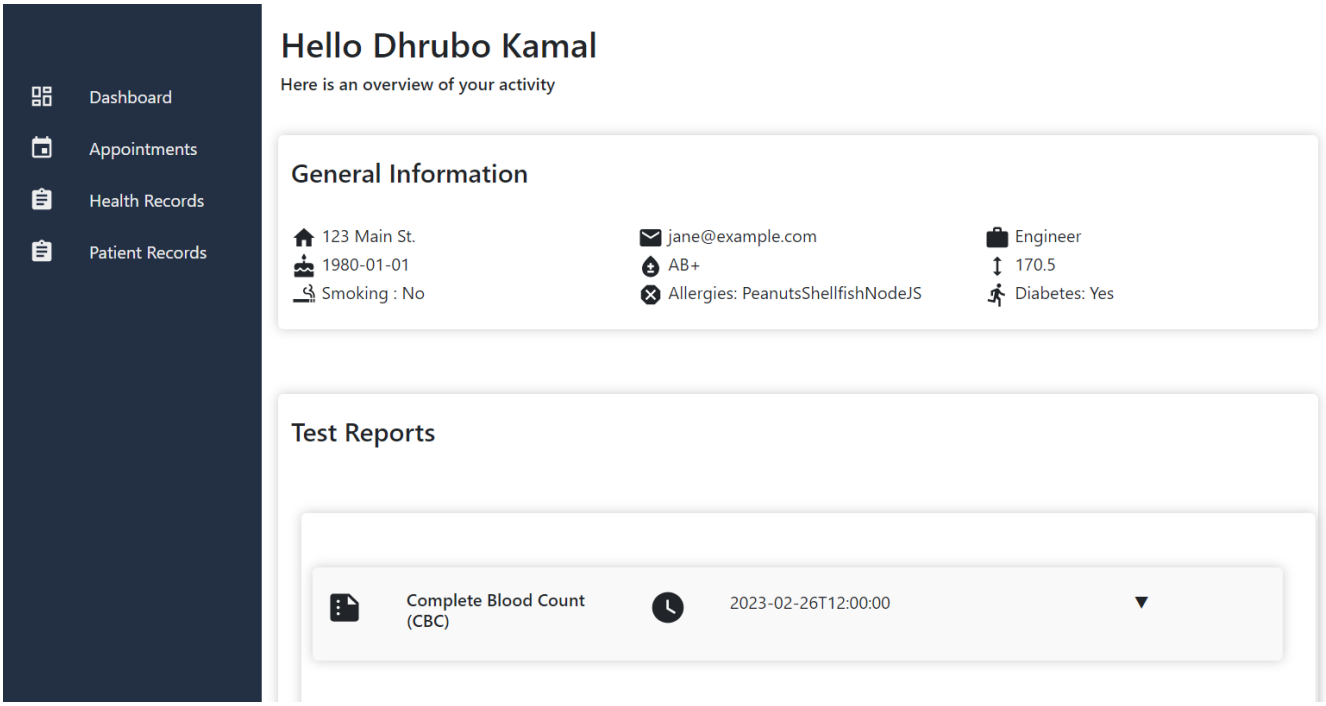

Figure 52: Patient History

10.2.5 Patient Test Result

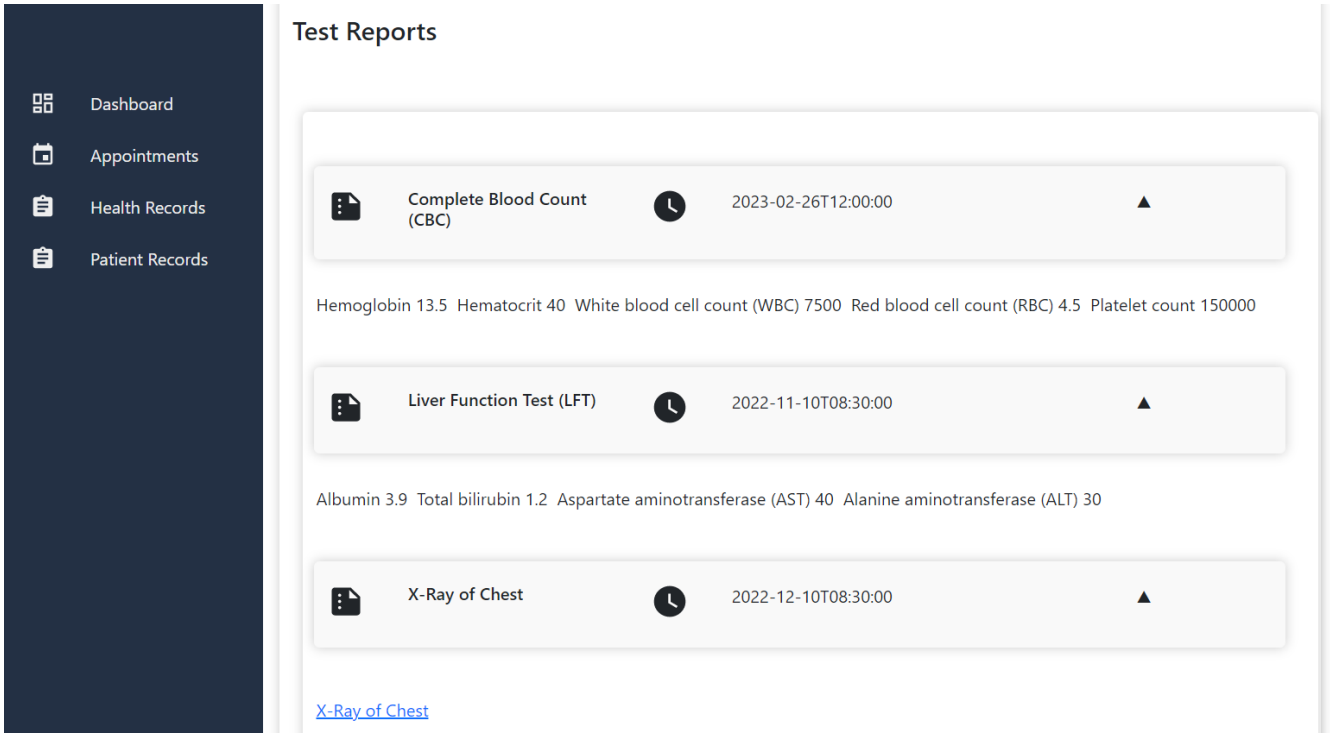

Figure 53: Patient Test Result

10.2.6 Doctor's Dashboard

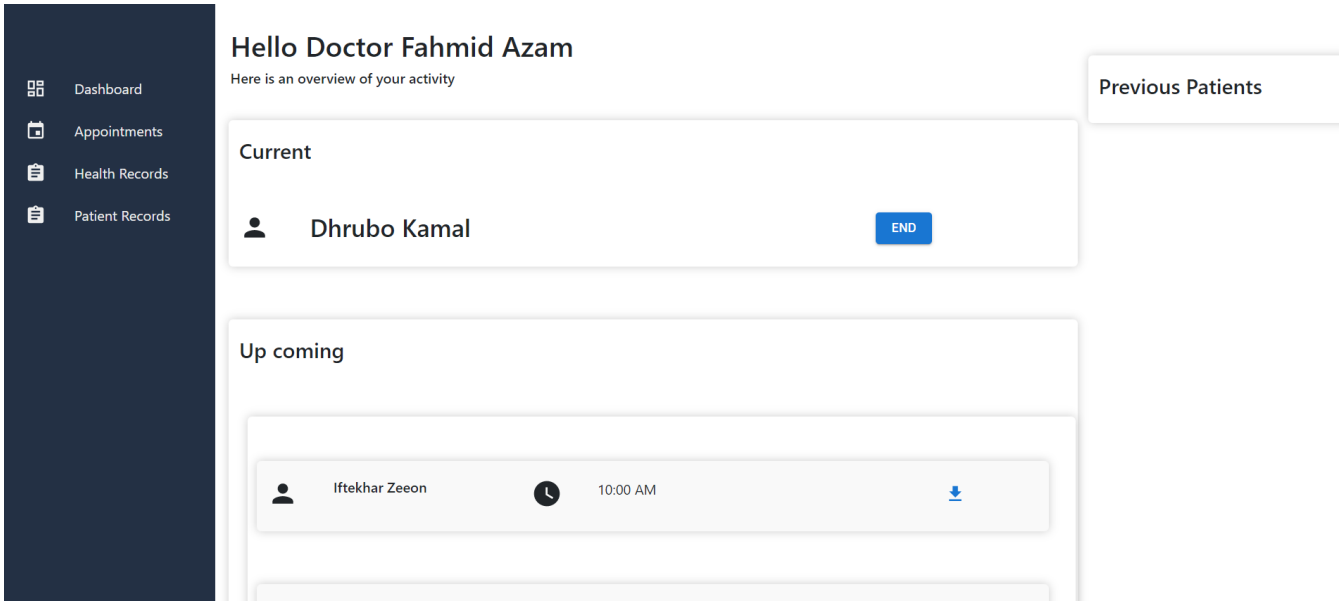

Figure 54: Doctor's Dashboard

10.2.7 Doctor's Prescription

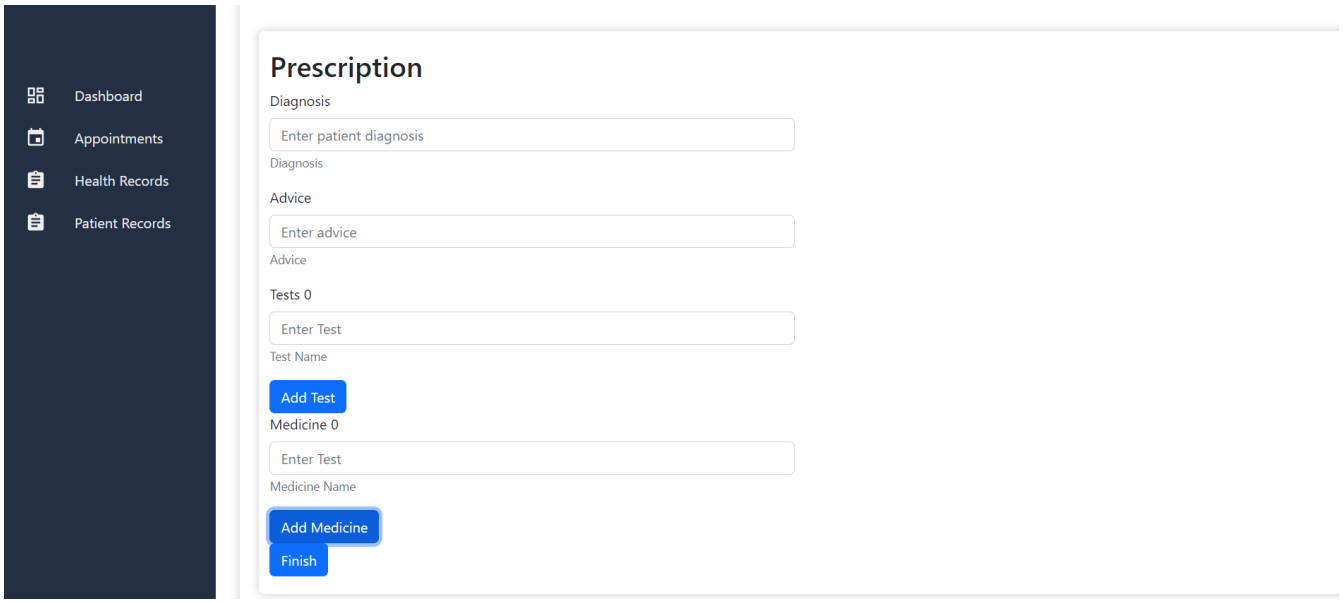

Figure 55: Doctor's Prescription