Palestine Technical University-Kadoorie

Faculty of Engineering and Technology

Department of Computer Systems Engineering

Software Engineering Requirements Document

Online Managing Patients' Appointments System

Submitted By:

Student's Name	Student's ID
Amr Haytham Hoji	202110624
Amjad Khalil Al-Tanni	202110719
Haneen Mohammad Sabra	202111914
Areej Essam Amer	202110242

Table Of Contents

Abstract

Waiting for a long time In order to schedule an appointment in the current traditional system in the hospital, and this system has always caused problems, whether on the part of customers (patients), or on the part of the hospital, among these problems: it wastes a lot of time and effort for both parties, and it's Inconveniencing for the patients by coming to the hospital to get an appointment.

Our system will be like an online Medical Management service provider with easy-to-use customizable options, it allows the patient to see the available booking appointments and let him to choose the appropriate appointment for him and pay within a few clicks, and makes it Easier for hospitals to deal with patients, and creating reports.

The system is accessible from anywhere, it will basically decrease the manual work and effort for both parties, and provide human resources in hospitals by allowing reservations to be handled through the site instead of the employee, and improve the quality of maintaining records.

Abstract (human translate)

يسبب النظام التقليدي لحجز المواعيد في المستشفيات العديد من المشاكل للمرضى و الاطباء و من هذه المشاكل : الانتظار لساعات طويلة لحجز موعد مع الطبيب ، و هذا يسبب مضيعة للوقت و الجهد للطرفين ، و يسبب ازعاج للمرضى بالقدوم الى المشفى لحجز موعد.

سيوفر نظامنا خدمة الادارة الطبية عبر الانترنت بشكل سهل حيث سيتمكن المريض من رؤية المواعيد المتاحة و حجز موعد مناسب له و الدفع بطريقة سريعة و سهلة ، و يسهل على المستشفيات التعامل مع المرضى و انشاء التقارير لهم.

النظام سيكون متاح في كل مكان ، و بشكل اساسي سوف يقلل من العمل اليدوي و الجهد لكل من المريض و الطبيب ، عن طريق التعامل مع الحجوزات من خلال الموقع بدلا من الاوراق و هذا سيؤدي لتحسين جودة الحفاظ على السجلات وزيادة كفاءة المستشفيات.

Abstract (Google translate)

الانتظار لفترة طويلة من أجل تحديد موعد في النظام التقليدي الحالي في المستشفى، ودائماً ما يسبب هذا النظام مشاكل سواء من جانب العملاء (المرضى)، أو من جانب المستشفى، ومن هذه المشاكل: فهو يهدر الكثير من الوقت والجهد لكلا الطرفين، كما أنه يزعج المرضى بالحضور إلى المستشفى للحصول على موعد.

سيكون نظامنا بمثابة مزود خدمة الإدارة الطبية عبر الإنترنت مع خيارات قابلة للتخصيص سهلة الاستخدام، فهو يسمح للمريض بالاطلاع على مواعيد الحجز المتاحة والسماح له باختيار الموعد المناسب له والدفع خلال بضع نقرات، ويجعله يسهل على المستشفيات التعامل مع المرضى، وإنشاء التقارير.

يمكن الوصول إلى النظام من أي مكان، وسيعمل بشكل أساسي على تقليل العمل اليدوي والجهد لكلا الطرفين، وتوفير الموارد البشرية في المستشفيات من خلال السماح بمعالجة الحجوزات من خلال الموقع بدلاً من الموظف، وتحسين جودة الاحتفاظ بالسجلات

CHAPTER 1: INTRODUCTION

1.1 Project Overview & Background

- For decades, hospitals have been struggling with managing and organizing patient appointments, due to the increased influx of patients to hospitals and the difficulty of organizing and scheduling patient appointments.
- Appointments were handled by the patient coming to the hospital to take the appointment, and this system has disadvantages, **including:**

  - ✓ Inconveniencing the patient to come to the hospital to book an appointment, taking into account the distances he will travel to the hospital.
  - ✓ It is necessary to have an employee to book appointments for patients.
  - ✓ Wasting time and effort on the employee and patient.

Hence, there is a need for a site that manages patient appointments in hospitals and facilitates communication between hospitals and patients.

1.2 Problem Domain

The challenges faced by the current traditional ordering system are a major problem Obstacles to achieving efficiency and customer satisfaction:

Patients will have to stand in long lines to submit requests to book an appointment.

Hospital staff will manually record appointment requests.

After submitting an application, the patient must wait until he receives the application confirmation paper.

Manual work reduces efficiency and increases the possibility of errors occurring during the application process.

1.3 Project Objectives

Our project aims to develop an online appointment booking system for use in the appointment booking service for patients in hospitals, which will allow hospitals to quickly and easily manage the list of booking appointments available through the website. The customer can complete the operation he wants quickly, easily and without complications.

How We Propose to Address the Diagnosed Problems:

We aim to provide a solution to the problems facing hospitals in managing patient appointments by helping patients (customers) book an appointment that is appropriate for them and also appropriate for the hospital management.

This can be achieved by creating a database dedicated to hospitals to store the largest possible number of hospitals in Palestinian cities.

- Save time:

Through the service provided by the site, the customer can, within a few minutes, book an appointment, thus saving the customer time and effort, and thus increasing the customer's satisfaction with the service provided to him through the site.

- Security of data:

The data is well protected for personal use by registering for an account through Personal email and password for each user.

- Minimized manual data entry:

Automation ensures data accuracy during the booking request submission process. in general We propose an online appointment booking system, the main advantage of our system is that it is highly Simplifies the booking process for both the customer and the hospital administration.

The project will improve the management of medical appointments and increase their efficiency and accuracy. Because this system will reduce all manual work by replacing the traditional system in a computer system. It will eliminate manual work such as the employee entering patient data and booking an appointment for him.

1.4 Project Scope

The system will be a web-based platform that will be developed to allow customers (patients) to view everything, appointments booked and unbooked, doctors you can book with, and then place the order through the system. It is available for customers to book an appointment with a specific doctor, and they can also cancel this reservation, and they can also pay through the website.

The system will be able to allow the hospital administration to update appointments and make changes to them, and will also allow employees to create the reports they want to create, such as the monthly patient records report.

CHAPTER 2: REQUIREMENTS GATHERING APPROACH

At first we didn't know what to do about how to manage patient appointments in hospitals, so we discussed among ourselves and also visited some websites that support the idea of managing patient appointments in hospitals.

So we decided how to collect user requirements through web system.

We also consulted some experts with much more experience than us, who gave us guidance on how to collect user requirements.

CHAPTER 3: USER REQUIREMENTS DEFINITION

System requirements include the functions and features that the system includes to meet users' needs, including ease of use. Designing user-friendly, interactive interfaces helps achieve this by providing availability information to doctors and enabling patients to easily schedules and book suitable appointments, with just a few clicks. It also allows hospitalists to schedule their appointments and manage wait times to ensure maximum benefit from appointment scheduling and minimize gaps. It also ensures security, privacy and other essential aspects that contribute to improving user experience and system effectiveness.

3.1 Functional User Requirements:

Functional requirements are the specifications and characteristics that a system or application must meet to perform specific functions and meet users' needs.

Admin:

- Manage user accounts including creating, updating, and deleting accounts.
- Manage Physician files, and schedule configurations.
- Permission to generate reports on appointment statistics, user activity, and system performance.

User (patients):

The system shall enable the patients to

- Log in and create an account.
- Display a list of available locations for hospitals and clinics and the ability to choose the hospital that the user desires.
- View a list of available doctors and each doctor's schedule with available appointments and the ability to search for any doctor and his specialty
- View comprehensive information about available doctors, such as their specialty, working hours, and patient ratings.
- Possibility of booking an appointment for a visit.
- Confirm or cancel the appointment
- Providing an interface for communication between patients and doctors.
- Possibility of Pay medical fees via the application.
- Performance evaluation.

Manager (hospital Manager):

They have additional administrative responsibilities compared to regular users

- Supervising the scheduling and resource management process.
- Manage doctor accounts including creating, updating, and deleting accounts.
- Handling employee appointments.
- Create reports on appointment scheduling efficiency and patient satisfaction.
- View and monitor the performance of doctors, administrative staff, and other users regarding appointment management.

Doctor:

- Access and manage their appointment schedule, including viewing and updating booked appointments.
- Accept or reject appointment requests based on availability and personal schedule, and confirm accepted appointments.
- Provide notifications to patients about changes in appointments or cancellations.
- Ability to communicate with patients prior to appointments to provide necessary guidance.

System:

- Secure user authentication and authorization mechanisms to ensure data privacy and system security.
- Intuitive user interfaces for administrators, users, and managers to interact with the system easily.
- Confirm reservations and send alerts to users before the visit date.
- The system creates a daily list for each hospital of patients who have been confirmed to attend appointments.
- Providing reports and statistics on booking appointments and evaluating the overall performance of the system.
- Real-time synchronization of appointment data across different system components to prevent scheduling conflicts.

3.2 Non Functional User Requirements:

Non-functional requirements define how the system performs rather than the functions it must provide.

- **Security**: Ensuring patient data confidentiality and verifying identity for accessing sensitive medical information.
- **Performance**: Ensuring quick response times and handling high loads of requests.
- **Availability**: Providing continuous operation of the application without significant downtime.
- **Scalability:** The ability to increase the application's capacity and expand it to meet growing needs.
- **Maintainability**: Ease of system updates, bug fixes, and version management.

- **Usability**: Intuitive and user-friendly interface for a comfortable user experience.
- **Compliance**: Adherence to regulations and laws related to medical data protection and exchange.
- **Reliability**: Ensuring the stability of the application and minimizing the chances of failures.
- **Infrastructure independence**: The ability to run the application on a variety of devices and platforms without major issues.
- **Environmental performance**: Optimizing resource usage efficiency and reducing energy consumption.

These requirements contribute to improving the system's quality, user experience, and ensuring its effective continuity.

CHAPTER 4: SYSTEM ARCHITECTURE:

This chapter will provide an overview of the system's structure and architecture, showing the distribution of tasks across the different system modules, and how they relate to each other

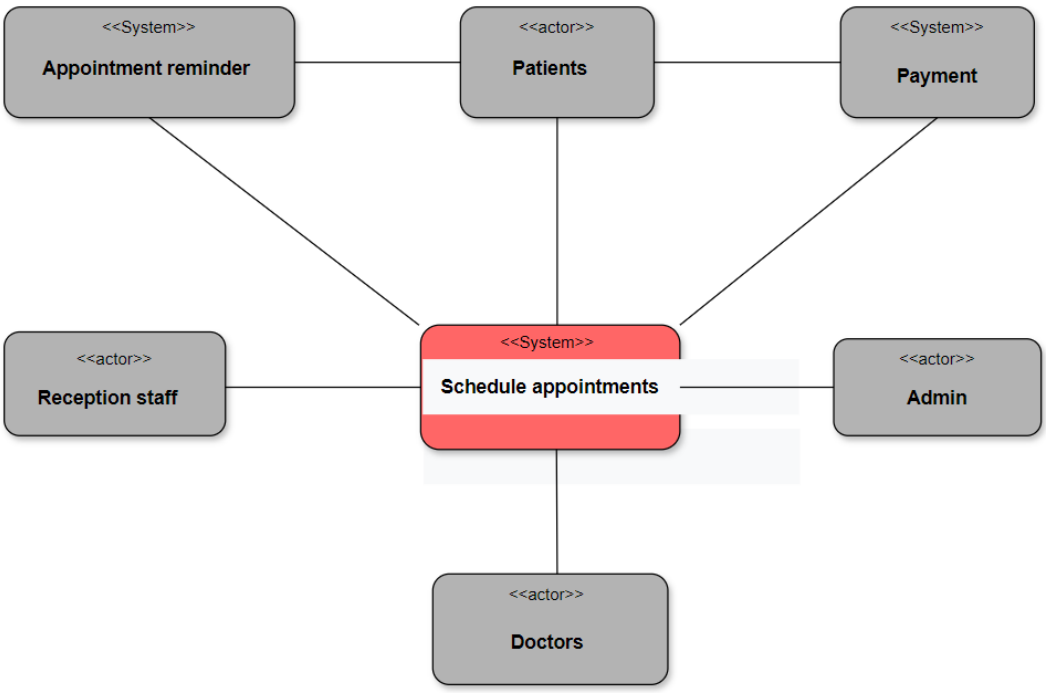


Figure 1: General Overview of our System's Architecture

4.1 What is Outside the System's Borders:

System boundaries are created to define what is inside the system and what is outside it.

Such as dealing with payment methods:

We try to keep our system simple, so as not to interfere with the payment system

Details, as it requires integrating our system with an external system for which we are responsible

Handling payments and payment methods (such as Visa card).

4.2 Context Diagram:

This is a brief structure that explains the environment in which the system exists and helps understand how the system communicates with what lies outside the system's boundaries and how they relate to each other.

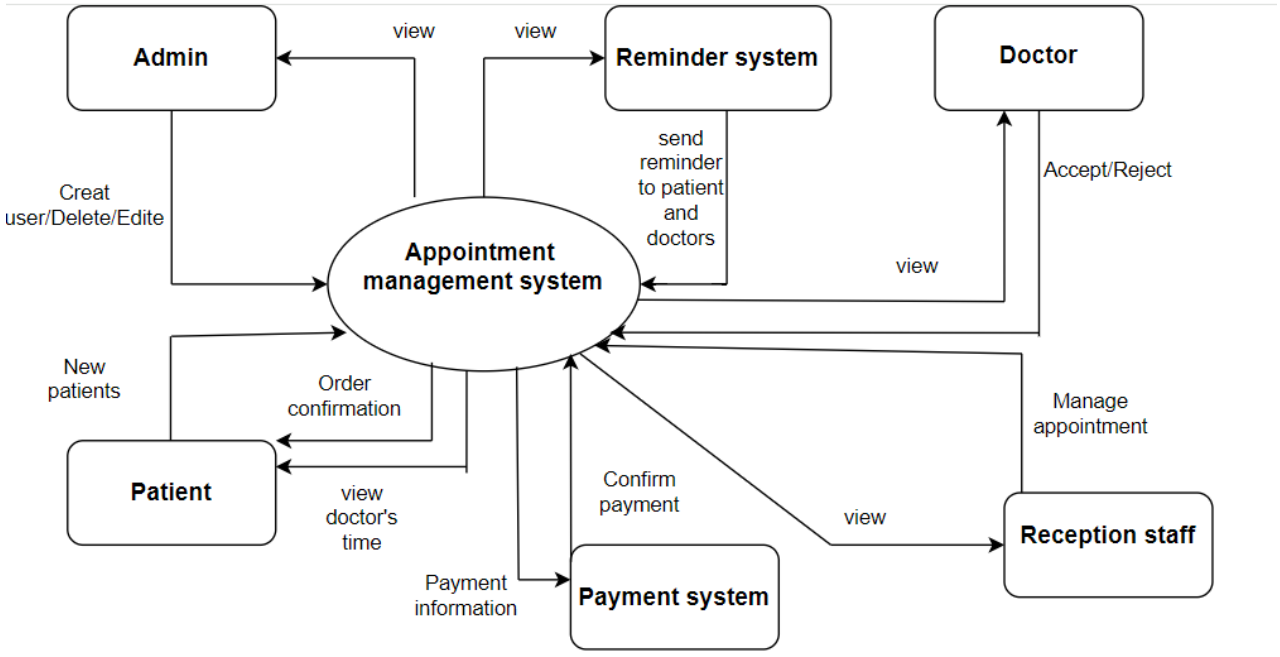


Figure 2: Context Diagram of our System's Architecture

CHAPTER 5: SYSTEM REQUIREMENTS SPECIFICATION:

Now I will try to describe some functional and nonfunctional requirements that were stated in chapter 3, but in more detail.

5.1 FUNCTIONAL SYSTEM REQUIREMENTS:

The system has a home screen, through which any user who has created an account can log into the system using their username and password.

Table 1: < Functional System Requirement > User Login

User Login	
Brief Description	The log in use case allows the user to login into the system.
Actor(s)	A user, could be: • An admin. • A patient • A doctor • Hospital's Manager Note: I think that there will be a different login form for each type of user mentioned above, because each type will be stored in its own database.
Flow of Events	At Homepage ➔ Login Form; 1. The system prompts the user for his/her username and password. 2. The user enters his/her username and password. 3. The system validates the entered information, ensures that the entered data actually exists in the user's database. 4. The user is Logged in. 5. The use case ends.

Pre-conditions	Having an **account** before.
Post-conditions	Secure connection to the database
Success	The user entered data that is stored in the user account

Table 2: < Functional System Requirement > **hospital's** Manager

Scheduling the appointments by **hospital's Manager**	
Brief Description	Allows hospital's manager to create and **Scheduling appointments** .
Actor(s)	The hospital's manager (each new added hospital will be assigned to a manager who will be responsible for this store)
Flow of Events	After the manager has logged in successfully, now they can go to the nav bar at the top ➔ supervise to schedule appointments and resource management process. • The manager can edit doctors accounts for example add new doctors accounts, updating, and deleting accounts. •The manager can schedule and edit the appointments whenever he/she is want. •Handling employee appointments. • The manager can create reports on appointment scheduling efficiency and patient satisfaction. •View and monitor the performance of doctors, administrative staff, and other users regarding appointment management.

Pre-conditions	Having hospital's manager, the logged successfully to their account.
Security	Secure connection to the database. Only the hospital's manager and doctors can access the appointment and make such changes (**insert, update, or delete**) Other people such as patients, can only browse the list of available doctors and each doctor's schedule with available appointments, but do not have the rights to insert, update any schedule.
Post-conditions	The hospital's manager is recommended to make a backup of their appointment's schedules after each change.
Success	The hospital's manager has created their appointments schedules and it has been booked by patients.

Table 3: < Functional System Requirement > **patient** Functions

user/ patient

Brief Description	Let the patient view and browse the appointments at any hospital they desire, then give them the opportunity to book an appointment with any doctor they want .
Actor(s)	The **patient** (user who wants to book an appointment)
Flow of Events	**patient** after login: 1. After the patient has created an account and logged in successfully, now a button "My location" will show the nearest hospitals to the patient's location and "choose location" will show all hospitals in the location that the patient has chosen. 2. Now the patient can browse a list of all **top doctors** and information about them, then the patient can **choose** and **message** any doctor he wants. 3. patient can book more than appointment from appointments page, The page will view doctors available and booked appointments and information about it like (doctor name, date , place , time). 4. After the patient clicks on "book an appointment "the site will send a notification to confirm the appointment. Note: the patient can confirm or cancel the appointment. 5. after confirming, the appointment will then be added to the appointments database, with the patient's ID and other related info that indicates that this appointment is booked by this specific patient. 6. Then the patient can pay with any payment method he wants from payment method page
Pre-conditions	Booking an appointment requires patient to login successfully.
Security	Secure connection to the database. Once the patient has booked an appointment, a notification will be sent directly to their email to confirm. If the appointment was not booked by the patient! He/she can cancel it, and the patient can change their password.

Post-conditions	After the appointment is complete, the patient will be asked to give a rating for the service and doctors.
Success	that the appointment is completed successfully, easily, and as the patient wanted

5.2 NON-FUNCTIONAL SYSTEM-REQUIREMENTS:

In systems engineering a non-functional requirement specifies how the system should behave and they specify constraints upon the system's behavior.
One could also think of nonfunctional requirements as quality attributes of a system.

1. Security:

- The system has login and sign-up pages.
- To register, the user needs to enter an email and password which must be at least 7 characters, and must include at least: 1 lowercase letter, 1 uppercase letter, numbers, and one of the following characters (#, $, *, !).

2. Ease of use:

- The system should have a simple and easy-to-learn graphic user interfaces.
- A new user should be able to use most of the system's functionalities in less than half an hour of training.
- Provides help frame such as error messages while entering invalid data.

3. Availability:

- Each hospital can specify their appointments days and hours, and then their appointments on our system will be available for patients to book there appointments easily.

4. Reliability:

- The Backup will be daily to prevent any data lost or when system is break down.

5. Supportability:

- Support will be provided by team members 24 hours a day so that a number will be added to contact him in the event of any defect in the system or inquiries about something specific.

CHAPTER 6: SYSTEM MODELS:

6.1 Use-case Diagrams:

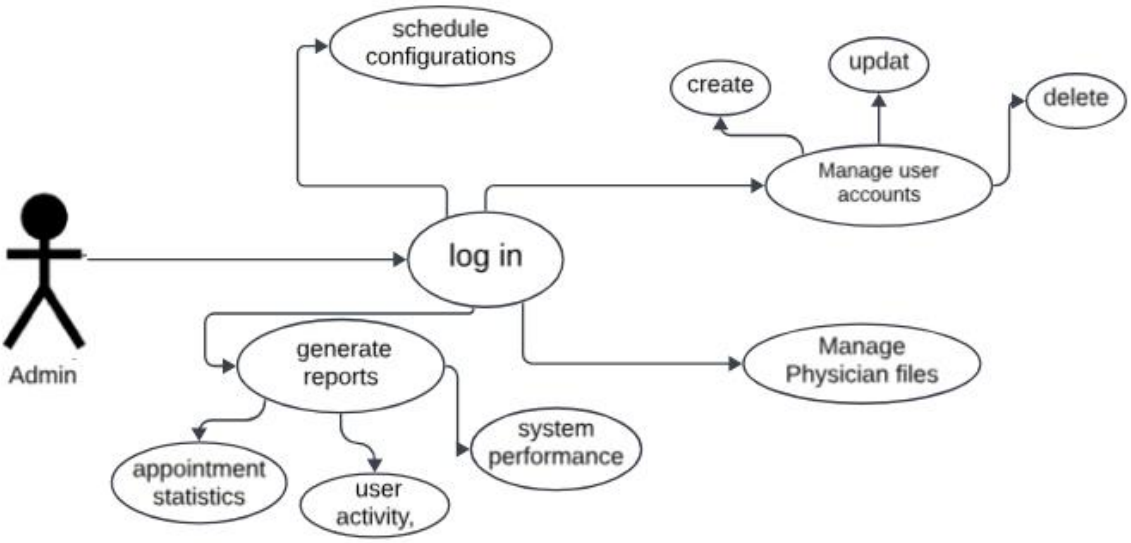


Figure 3: Admin's Use-case Diagram

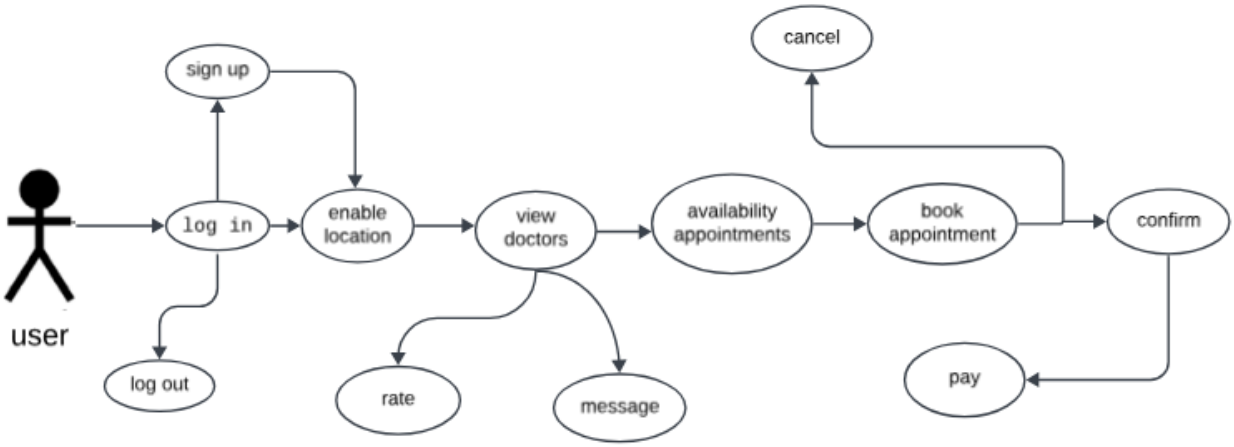


Figure 4: user's Use-case Diagram

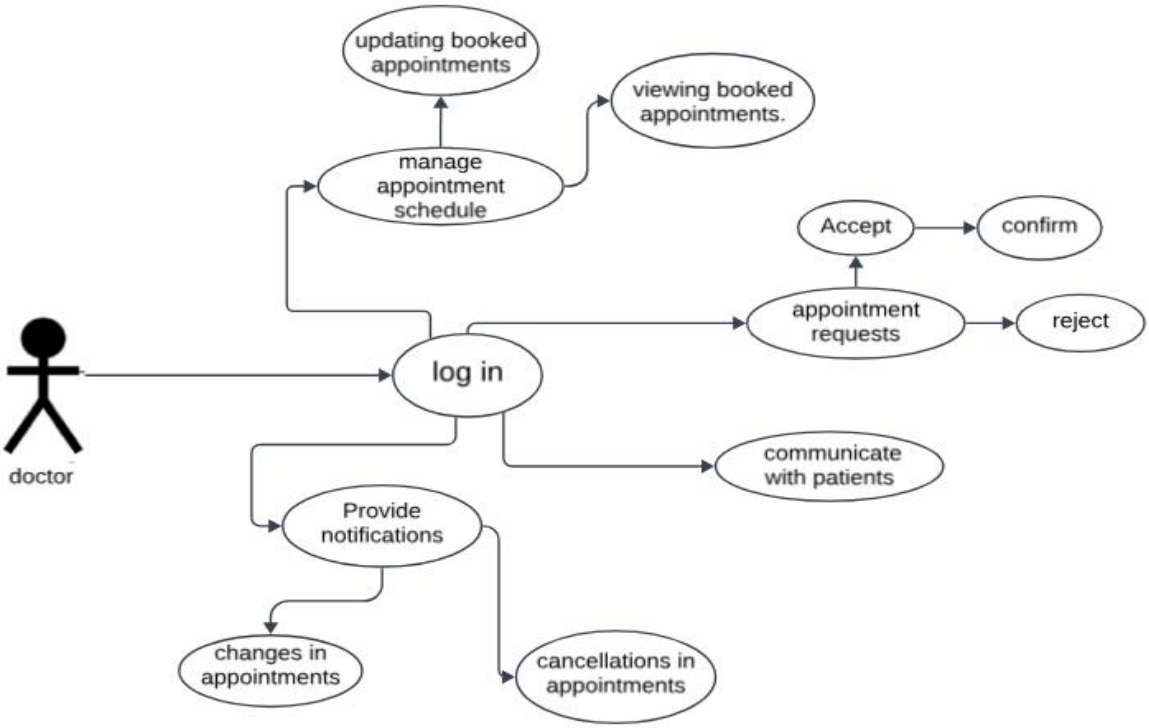


Figure 5: Doctor's Use-case Diagram

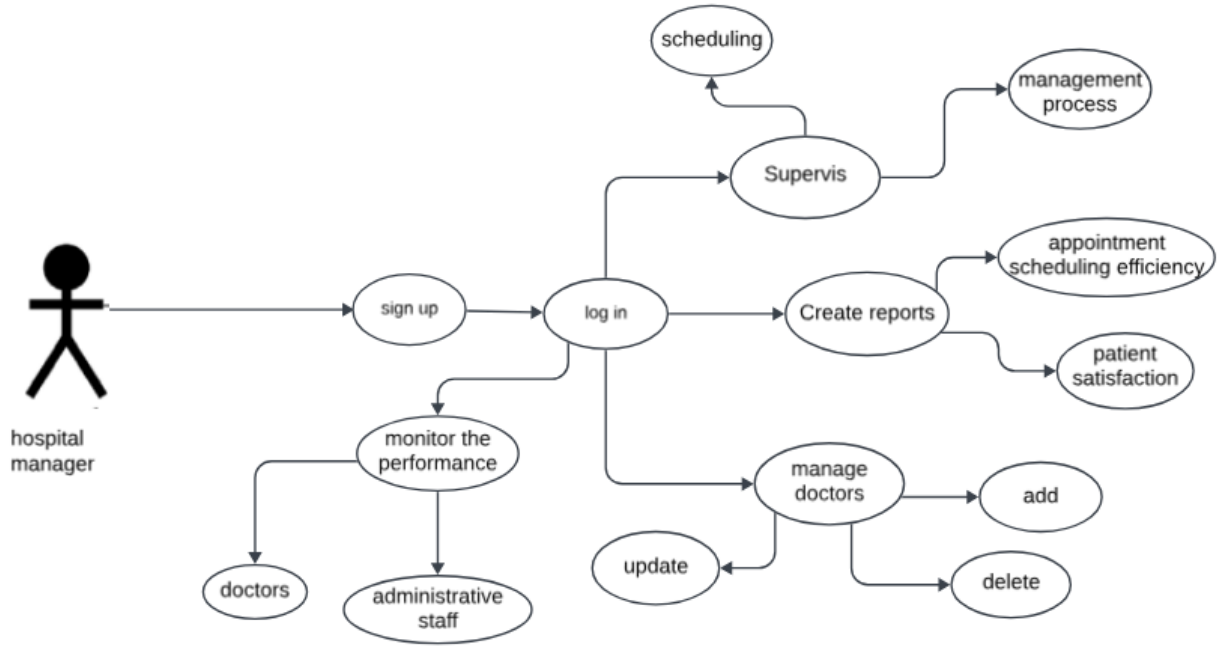


Figure 6: Manger's Use-case Diagram

6.2 Class Diagrams:

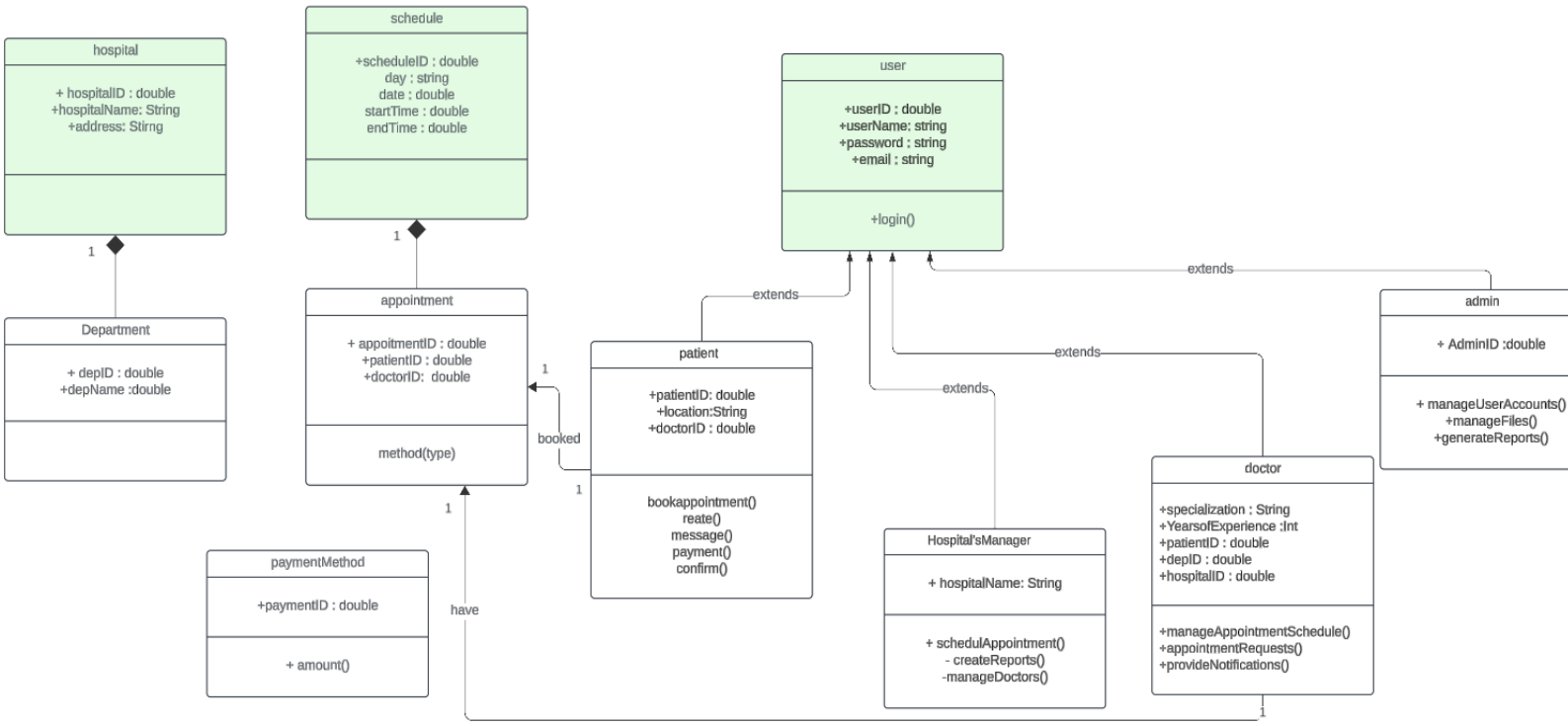


Figure 7: Class Diagrams

6.3 Activity Diagrams:

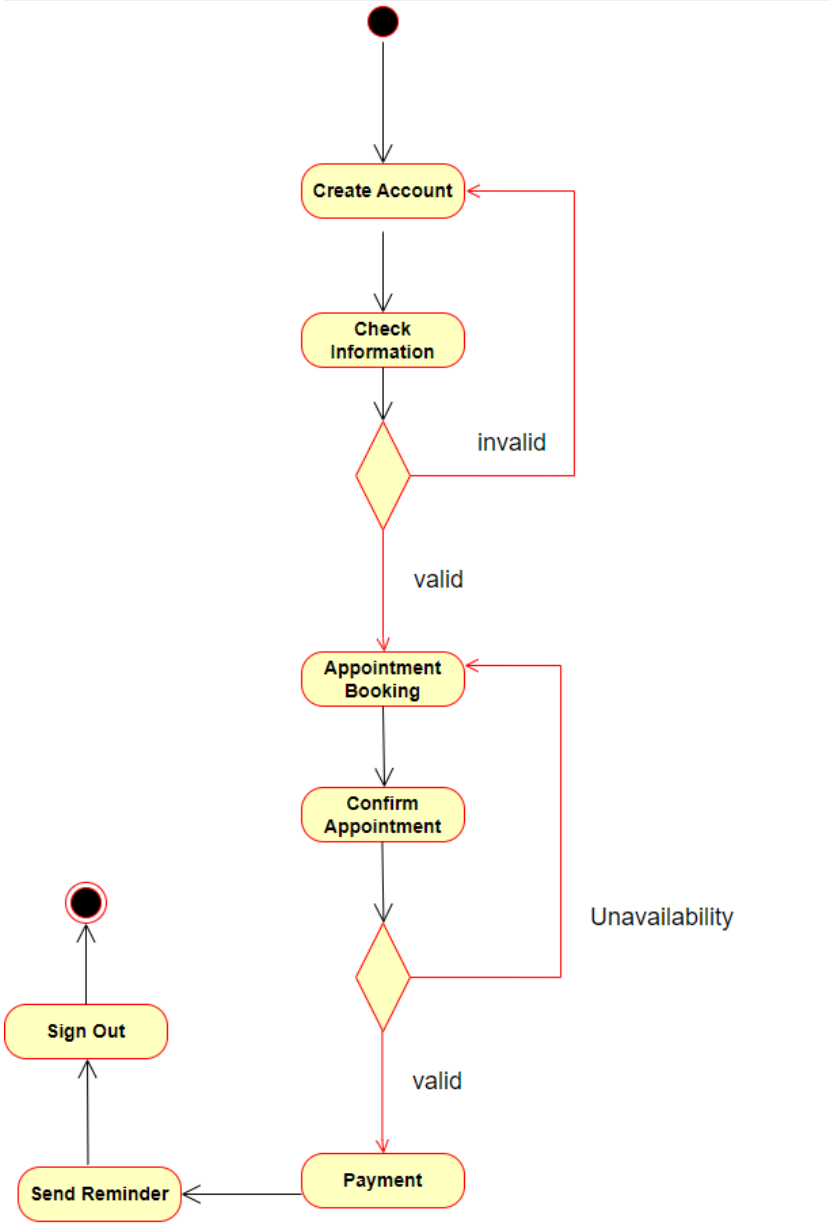


Figure 8: General Activity Diagram of a Patient to book an appointment for the first time

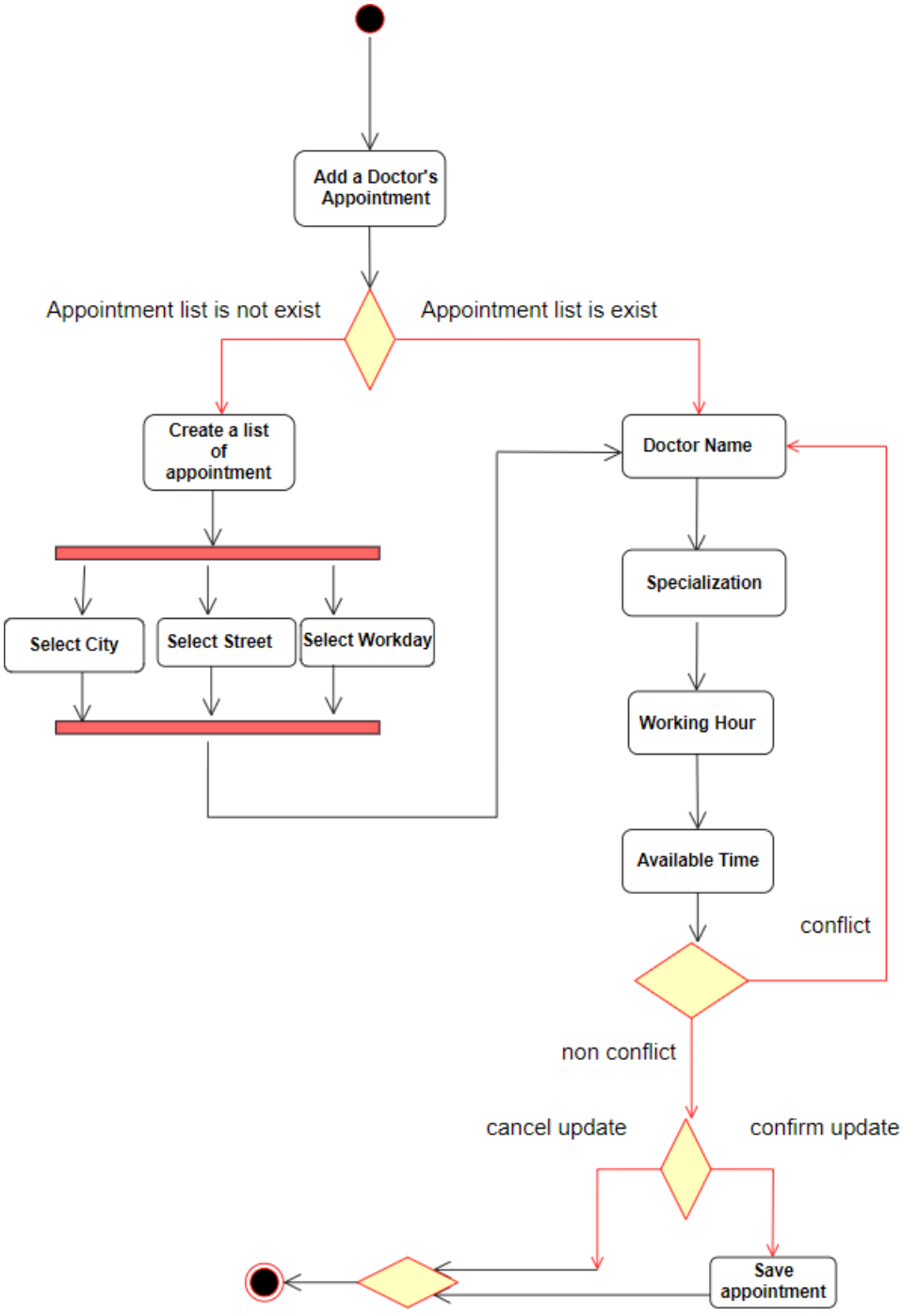


Figure 9: Activity Diagram shows the Flow of Events that a System's Admin should Follow to create a list of appointments.

6.4 Sequence Diagrams

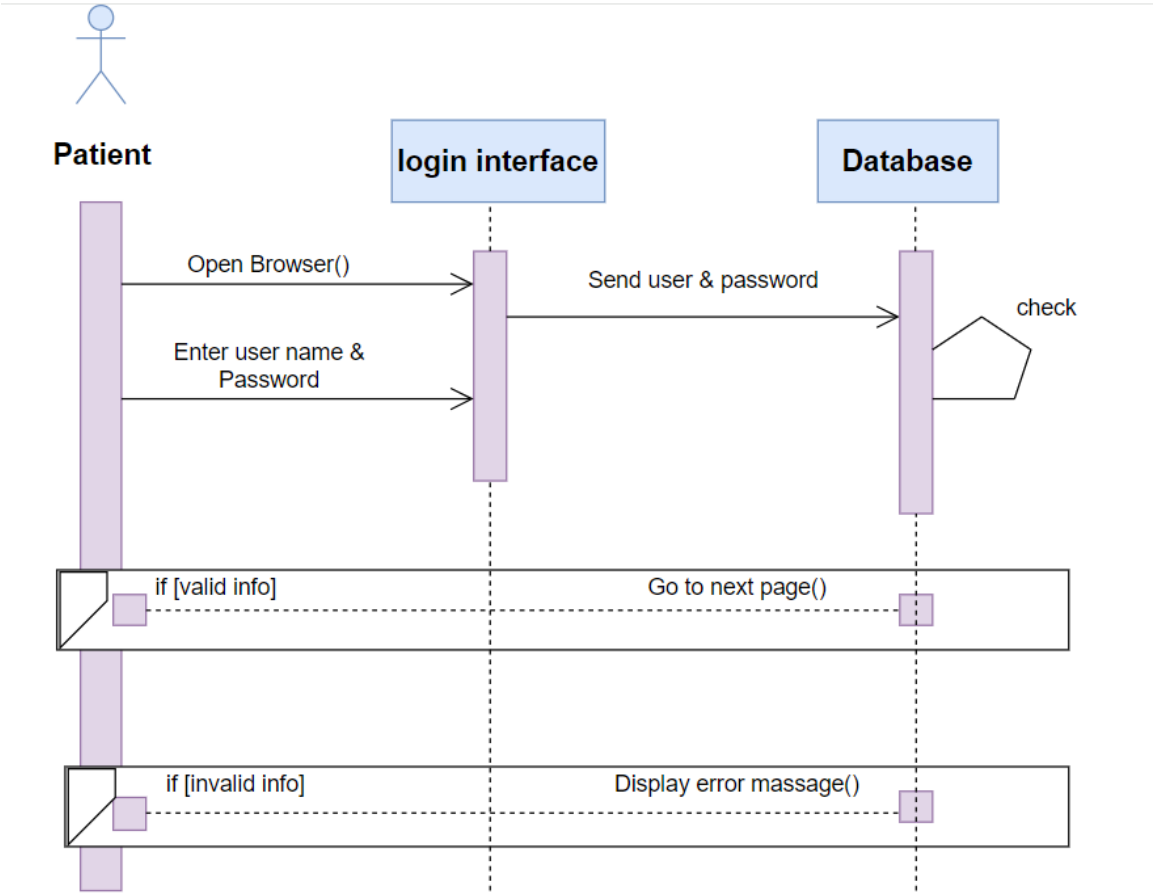


Figure10: Patient login Sequence Diagram e Diagram

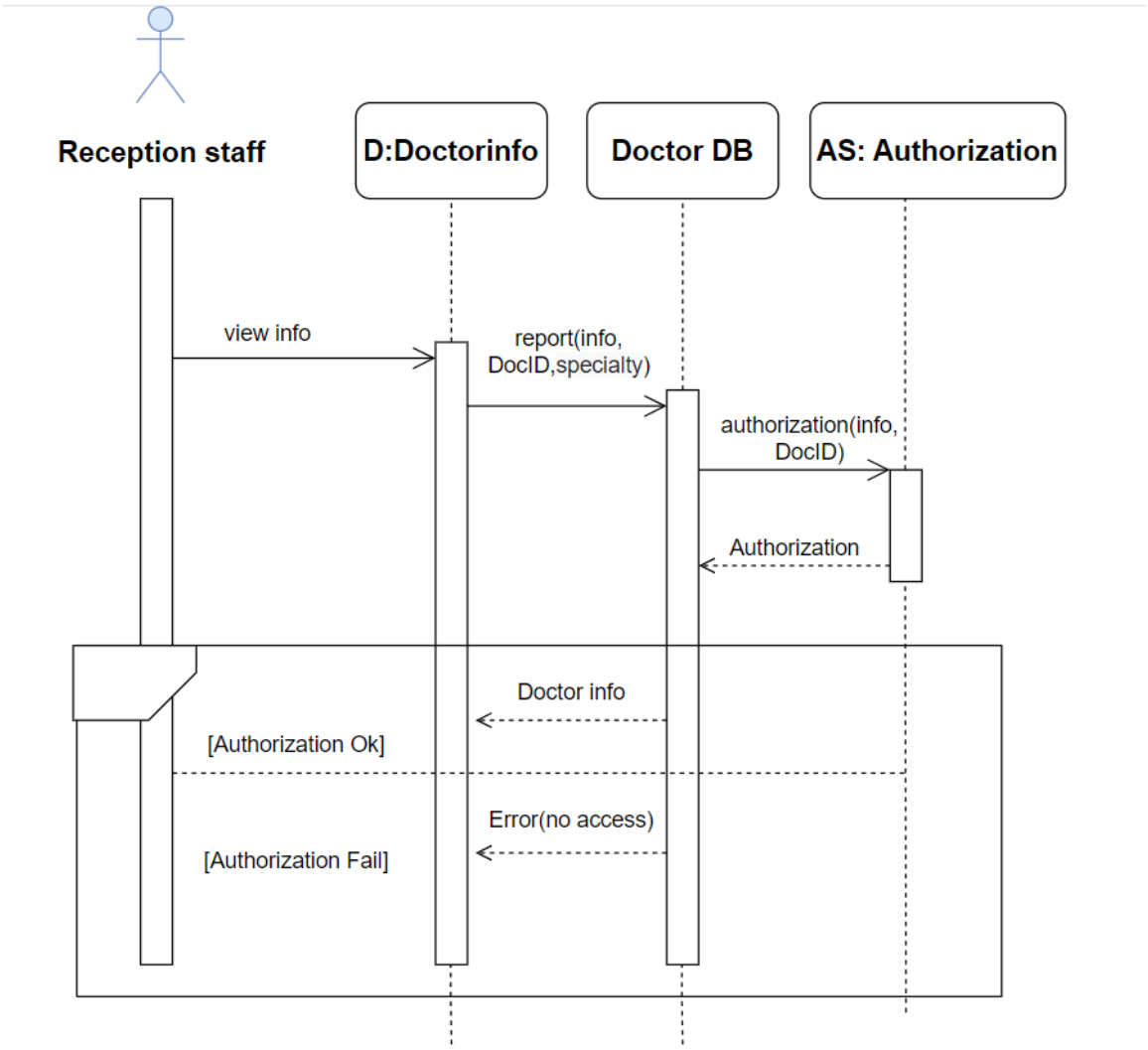


Figure11: Reception staff Views Info about doctors

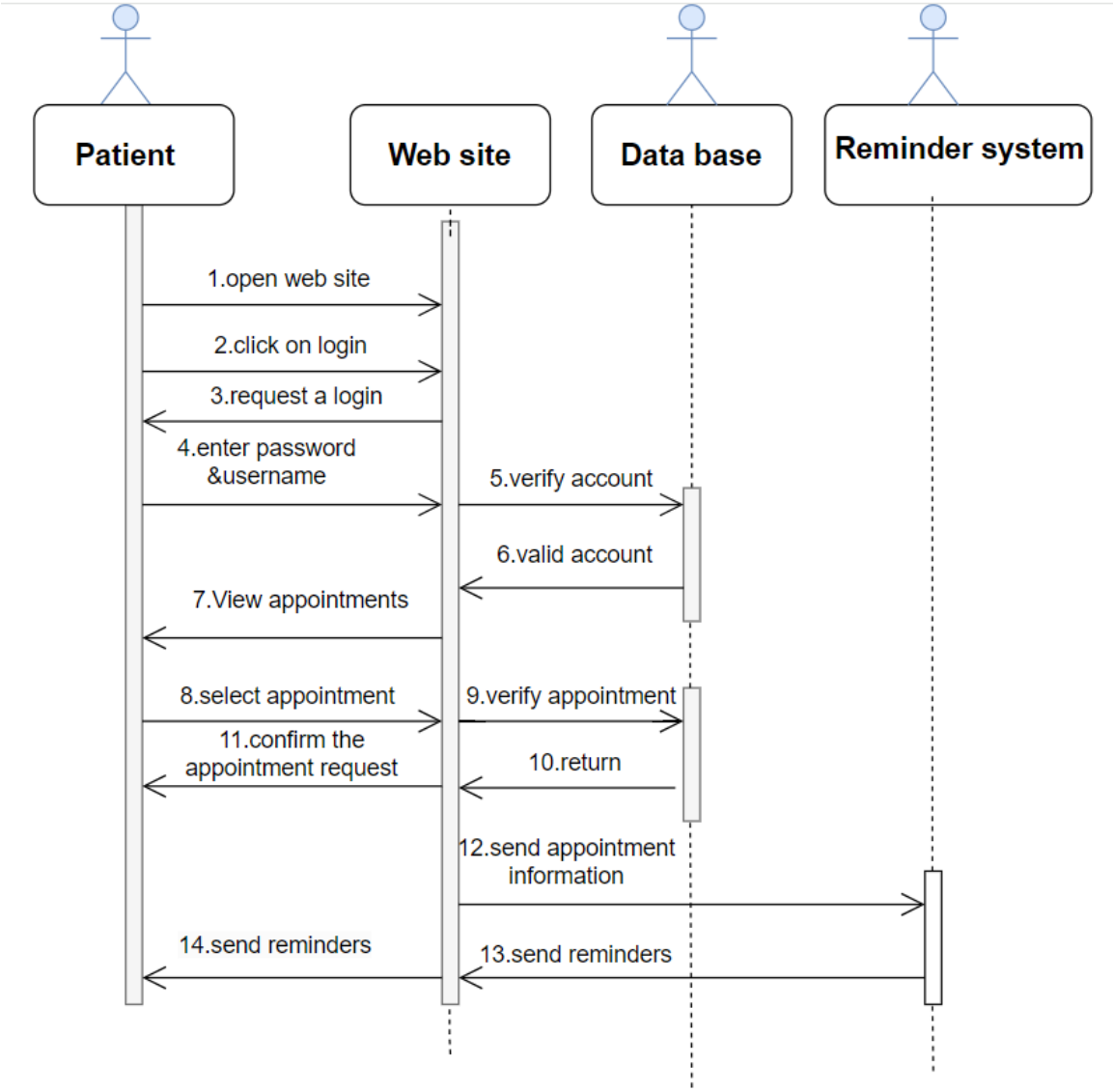


Figure12: Comprehensive Sequence Diagram Showing almost all Relationships between Customer and System's objects.

6.5 ER model:

The major features of the patient appointments database system as shown in below **entity–relationship model** (**ER model**

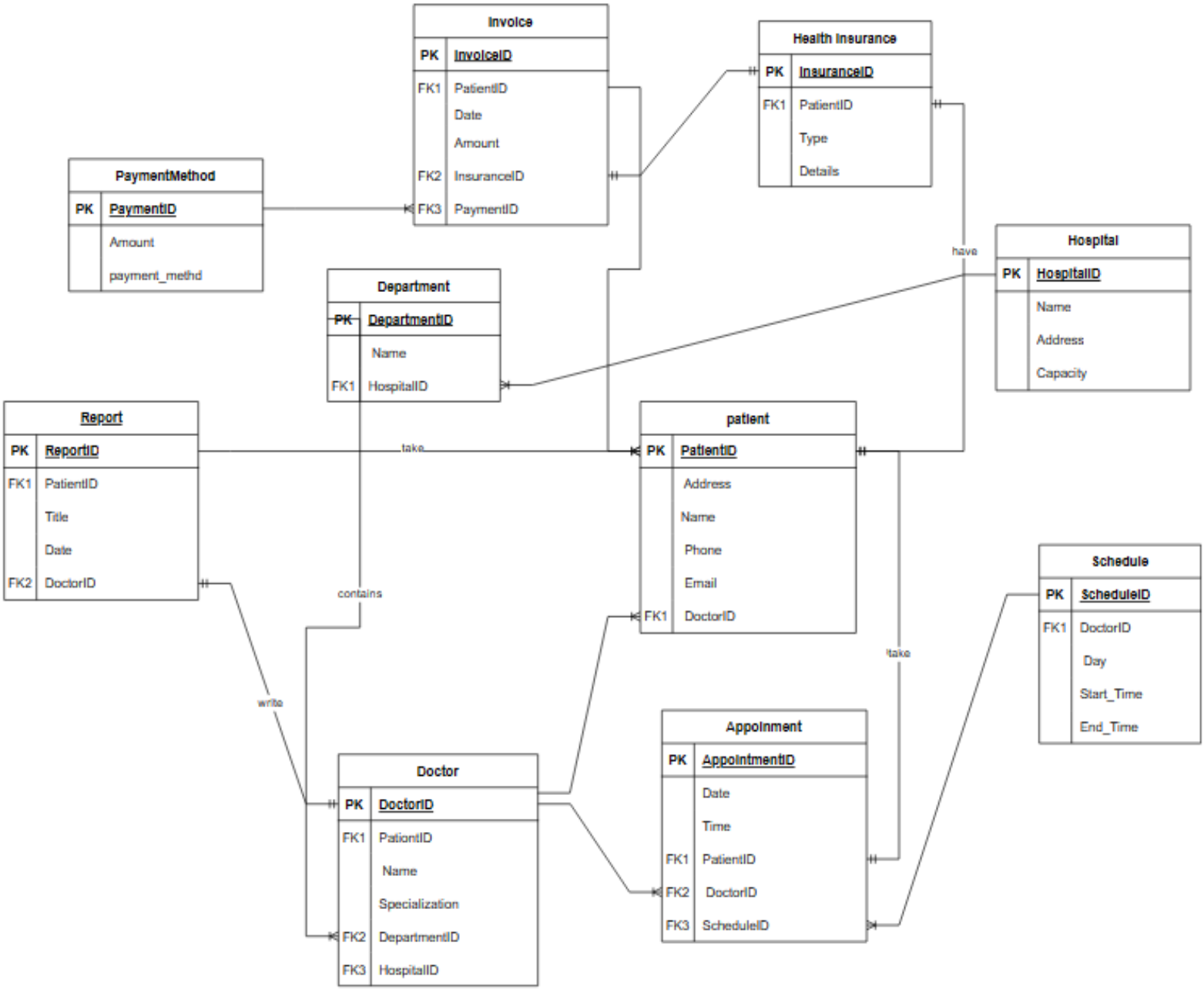


Figure13: ER (Entity-Relationship) Diagram

CAPTER 7: SYSTEM EVOLUTION:

SYSTEM LIMITATION:

1. **No-Show Patients:**
    - One of the main challenges is dealing with patients who do not show up for their appointments (no-shows).
    - No-shows can disrupt the schedule, waste resources, and affect other patients.
    - Implementing strategies to reduce no-show rates (such as reminders) is essential.

2. **Resource Constraints**:
    - Hospitals have limited resources, including doctors, nurses, and examination rooms.
    - Balancing patient demand with available resources can be challenging

SYSTEM ENHANCEMENT:

In future, we are looking forward to developing a mobile application for our Managing Patients' Appointments System using an android platform.

The mobile application contains these features:

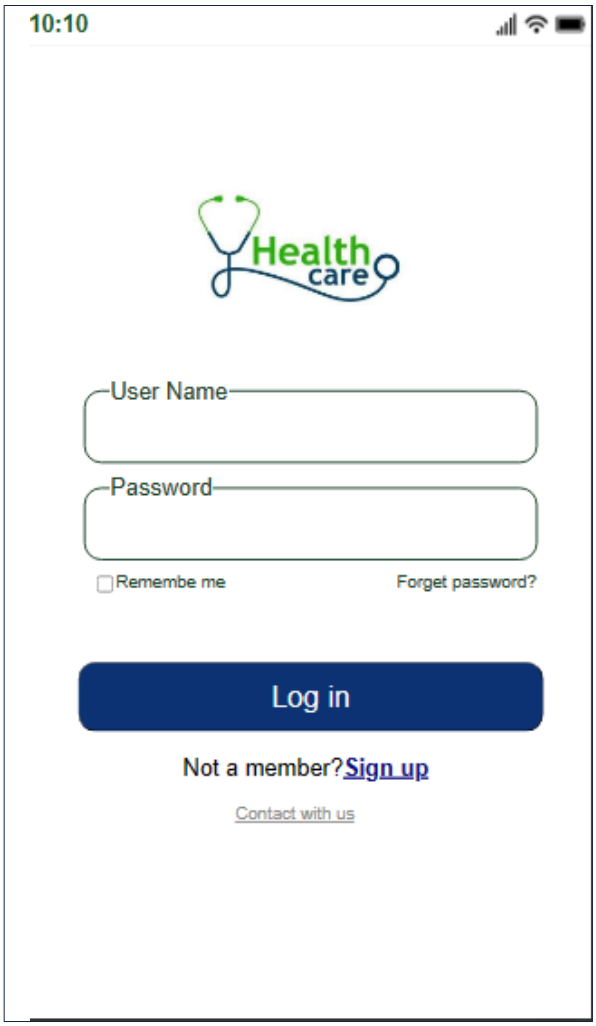
Integration with Electronic Health Records (EHR):

- Seamlessly integrate appointment scheduling with EHR systems.
- Ensure real-time synchronization of patient data between scheduling and clinical systems

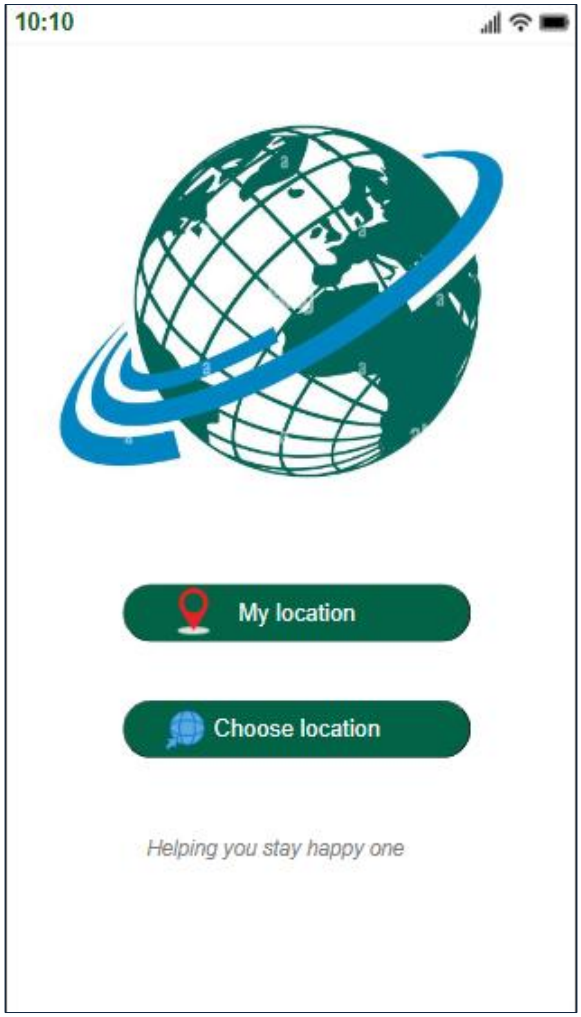
Multi-Language Support:

- Supports multiple languages to accommodate diverse patient populations

CHAPTER 8: USER INTERFACES



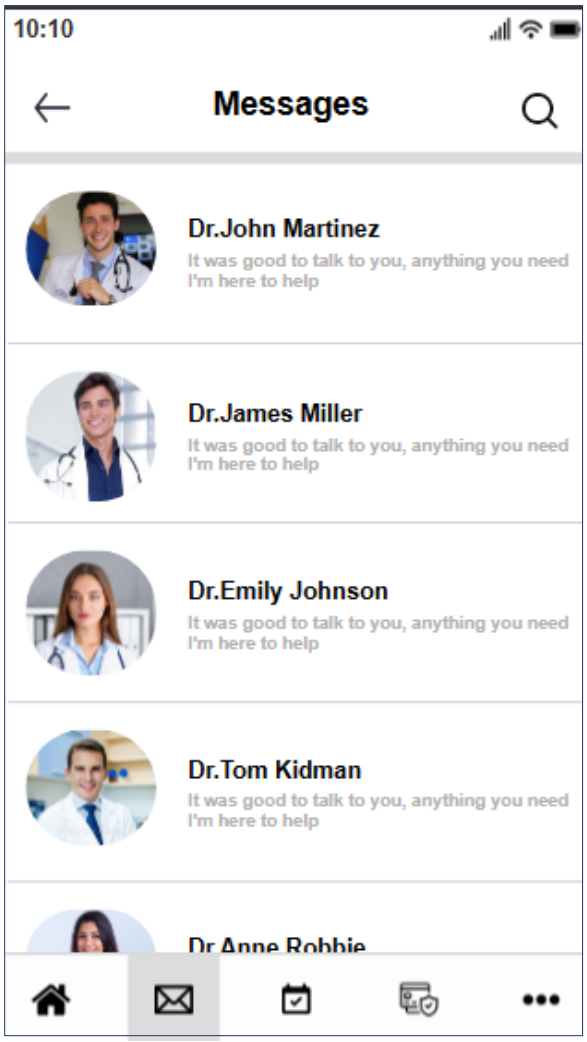
Log in or sign up interface



enable you location or choose area

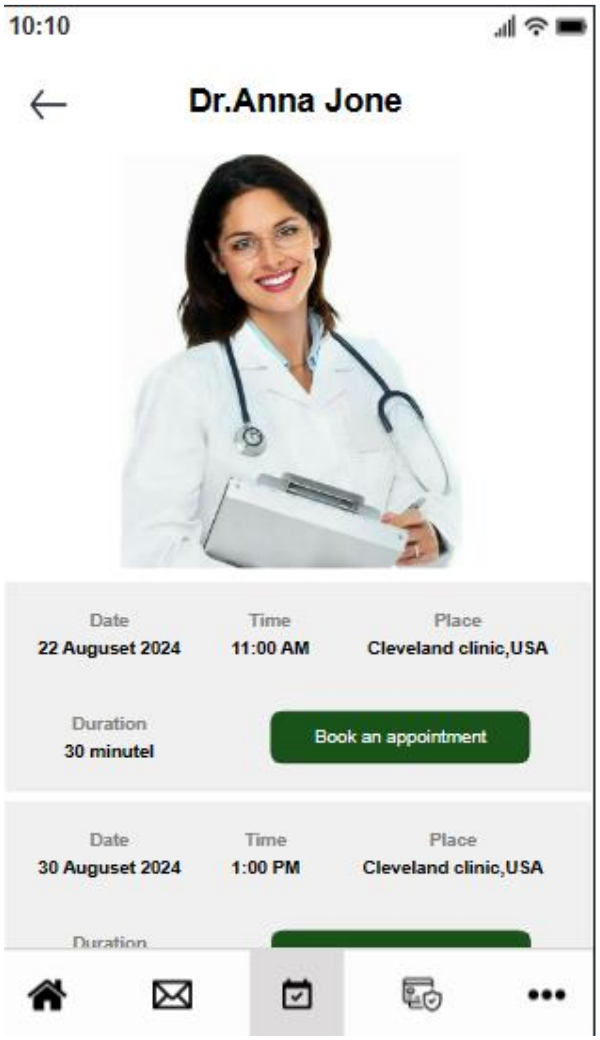
`

Home page

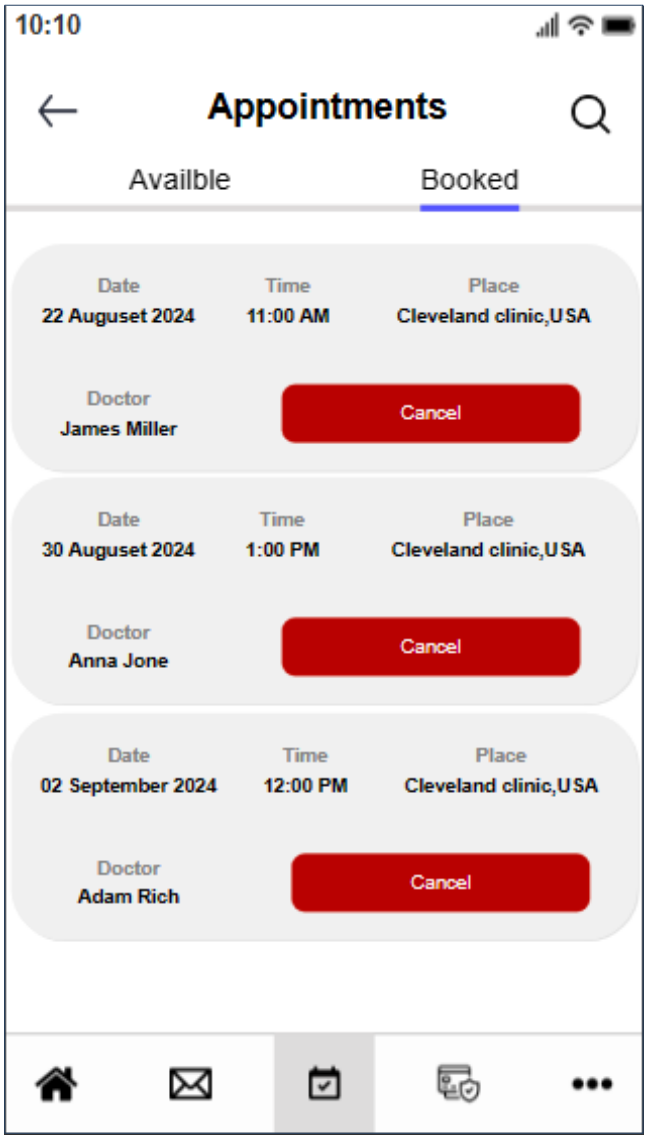


message page

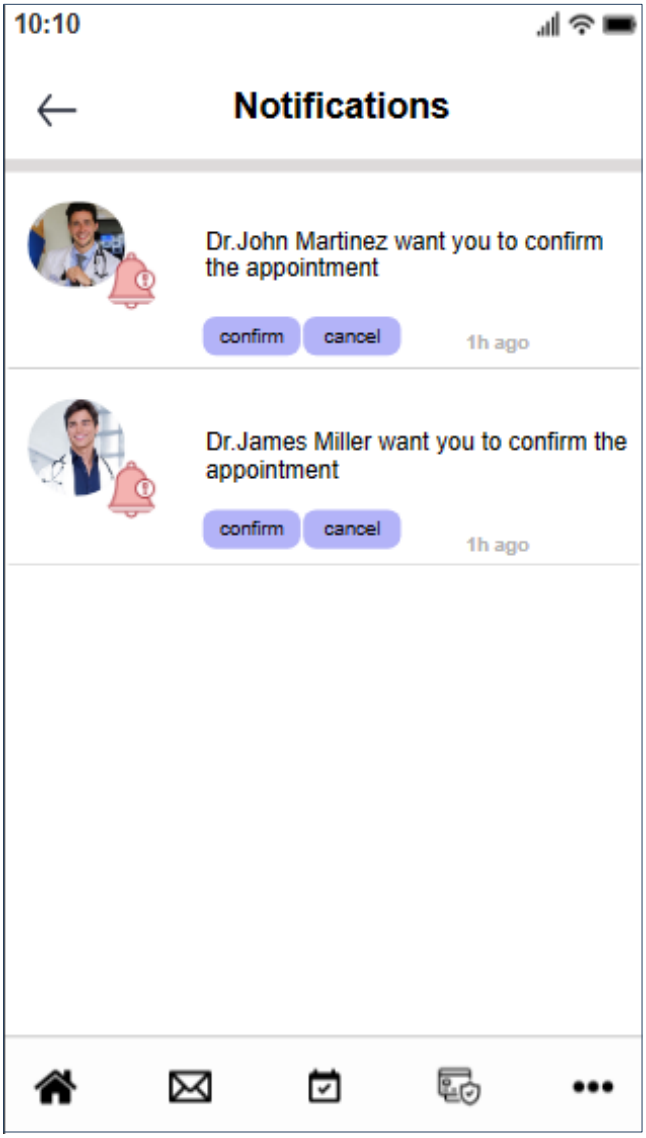
Available doctors



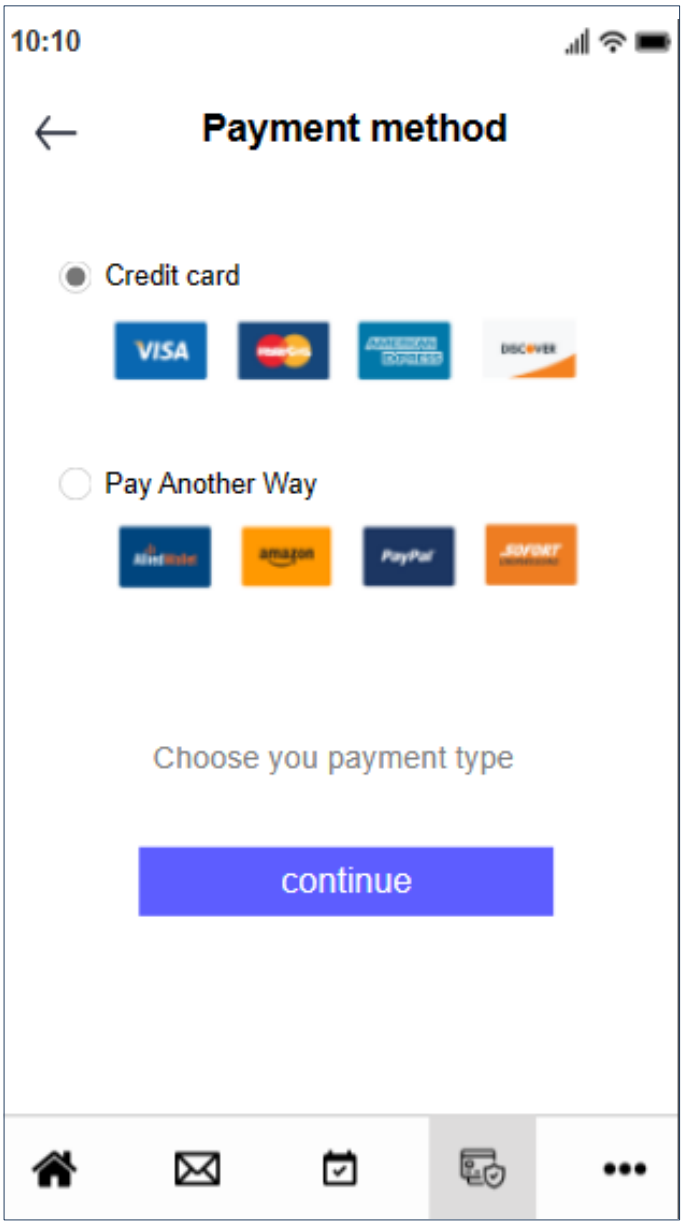
available doctor's appointment



User Booked appointment



notification



Payments method

CHAPTER 9: Conclusion

The work on this project was very useful. At the beginning, we had some difficulty in coming up with the idea of the project, and then in gathering the user and system requirements.

I think our project is not as simple as it looks. It requires much work to be tackled. We all consider ourselves as juniors or beginners in the web development field. We are not very experienced with web technologies. This will be the first big project we will face. So we are planning to work cooperatively and benefit of each other.

References:

1) **Lucid Chart** & **Draw.io** Tools for Drawing UML Diagrams.

2) What is Sequence Diagram?
https://www.visual-paradigm.com/guide/uml- unified-modeling-language/what is- sequence-diagram/

3) **The Book** :
https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/Software-Engineering-9th-Edition-by-Ian-Sommerville.pdf

4) **EHR Integration Guide: Benefits, Challenges, and Best Practices - Softemii Blog (softermii.com)**

5) **https://github.com/amrhoji11/Our-Project**