

# System Design Document

Group 06

Medsphere

Team Members:  
Adonia Sequeira  
Ganesh Rajasekar

**Course:**  
Software Engineering CSCI 6231

04/11/2024

## Table of Contents

- 1 Introduction
  - 1.1 Purpose
  - 1.2 Scope
  - 1.3 Overview
  - 1.4 References
- 2 System Design
  - 2.1 Overview
  - 2.2 Review Modules
  - 2.3 Diagrams
- 3 System Architecture
  - 3.1 Overview
  - 3.2 Architectural Design
  - 3.3 Review: External Components
  - 3.4 Diagram
- 4 Data Design
  - 4.1 Review Data Tables
  - 4.2 Class Diagram
- 5 Human Interface Design
  - 5.1 Overview of User Interface
  - 5.2 Screen Images and Interactions
  - 5.3 Screen Objects & Actions
- 6 Issues to be resolved

# **1 Introduction**

## **1.1 Purpose**

The Medsphere healthcare appointment scheduling system aims to improve patient access and convenience while streamlining the overall appointment management process for doctors and clinics.

## **1.2 Scope**

The system's scope consists of the following essential functionalities:

- Search and filter healthcare providers using numerous parameters (location, specialty, insurance, ratings, etc.).
- See real-time physician availability and plan appointments (in-person or virtual).
- Reschedules or cancels existing appointments.
- Receive appointment reminders and notifications through several methods (email, SMS, push).
- View video recordings and transcripts of virtual visits.
- View patient profiles and medical histories.
- Conduct virtual visits and create video recordings and transcripts.

## **1.3 Overview**

The Medsphere healthcare appointment scheduling system is a robust, user-friendly platform for booking and managing healthcare appointments for consumers, doctors and clinic personnel.

## **1.4 References**

Larson, E. W., & Gray, C. F. (2021). Project scope management: A practical guide to requirements for project success (4th ed.). McGraw-Hill Education.

Valpadasu Hema et al 2020 IOP Conf. Ser.: Mater. Sci. Eng. 981 022060

Desai, M., Tardiff-Douglin, M., Miller, I. R., Blitzer, S. C., Gardner, D. L., Thompson, T. M., & Levine, D. M. (2023). Implementation of Agile in health care: Methodology for a multi-site home hospital accelerator. medRxiv, 2023-12.

Brown, T. M., & Jackson, K. L. (2019). Mobile Application System Design: Strategies for Success.

## **2 System Design**

### **2.1 Overview**

Our healthcare appointment scheduling app's system design incorporates a modular and scalable architecture, with the goal of streamlining the booking and management of appointments for patients and healthcare professionals. Our system intends to redefine the appointment booking experience in healthcare by emphasizing modular design, flexible integration, and user-centricity, while also improving communication, efficiency, and patient pleasure.

### **2.2 Modules**

1. Patient Portal:
  - Patients can arrange appointments online depending on provider availability.
  - Allow patients to claim open/vacant appointment slots.
  - Allow patients to check in for their appointments via the portal.
2. Provider Availability and Location:
  - Display healthcare professionals' schedules for multiple places and days.
  - Clearly indicate where each appointment will take place.
3. Appointment Reminders:
  - Send patients automatic reminders about their forthcoming appointments.
  - Help to reduce the number of missed appointments (no-shows).
  - Maximize revenue streams by reducing unused appointment slots.
4. Recurring Appointments:
  - Allow patients to set up recurrent appointments.
  - Support repeated appointments on a weekly or monthly basis.
  - Automate the scheduling process for repeat appointments.
  - Save time and effort for both patients and administrative personnel.
5. Financial Management:
  - Manage patient billing and invoicing processes.
  - Manage insurance claim submission and processing.
  - Facilitate appointment payment processing.
  - Ensure that insurance companies get accurate claim submissions for reimbursement.

The modules address important areas of appointment scheduling, such as patient self-service, provider scheduling, reminders, repeating appointments, and financial management. The combination of these modules is intended to streamline appointment scheduling, improve patient experience, and maximize administrative efficiency and revenue management.

## **2.3 Diagrams**

### **a. Use Case Review**

#### **1. Patient Search and Selection**

Use Case: A patient wants to find a healthcare provider based on their specific needs and preferences.

Actors: Patient, System

Pre-Conditions: Patient is registered and authenticated in the system.

Flow of Events:

- Patient accesses the Medsphere system through the web or mobile app.
- Patient navigates to the provider search functionality.
- Patient inputs search criteria (e.g., location, specialty, insurance, ratings).
- System retrieves & displays a list of matching providers based on the search criteria.
- Patient reviews the provider profiles and availability.
- Patient selects a preferred provider and proceeds to the appointment booking process.

Post-Conditions: Patient has identified a suitable provider and is ready to book an appointment.

#### **2. Appointment Booking**

Use Case: A patient wants to book an appointment with a selected healthcare provider.

Actors: Patient, System

Pre-Conditions: Patient has identified a preferred provider through the search and selection process.

Flow of Events:

- Patient navigates to the appointment booking functionality.
- System displays the provider's available appointment slots (in-person and virtual).
- Patient selects the preferred appointment type (in-person or virtual) and time slot.
- Patient provides any necessary information (e.g., reason for visit, medical history).
- System validates the availability and creates the appointment record.
- System generates and sends appointment confirmation to the patient.

Post-Conditions:

Appointment is successfully booked, and the patient receives a confirmation.

### 3. Appointment Rescheduling

Use Case: A patient wants to reschedule an existing appointment.

Actors: Patient, System

Pre-Conditions: Patient has an existing appointment in the system.

Flow of Events:

- Patient accesses the appointment management functionality.
- System displays the patient's upcoming appointments.
- Patient selects the appointment they want to reschedule.
- System presents the available rescheduling options (in-person or virtual).
- Patient chooses a new appointment slot and confirms the rescheduling request.
- System updates the appointment record and sends a rescheduling confirmation to the patient.

Post-Conditions: Appointment is successfully rescheduled, and the patient receives a confirmation.

#### b. Data Flow Level 0-3 for System

##### DFD Level 0:

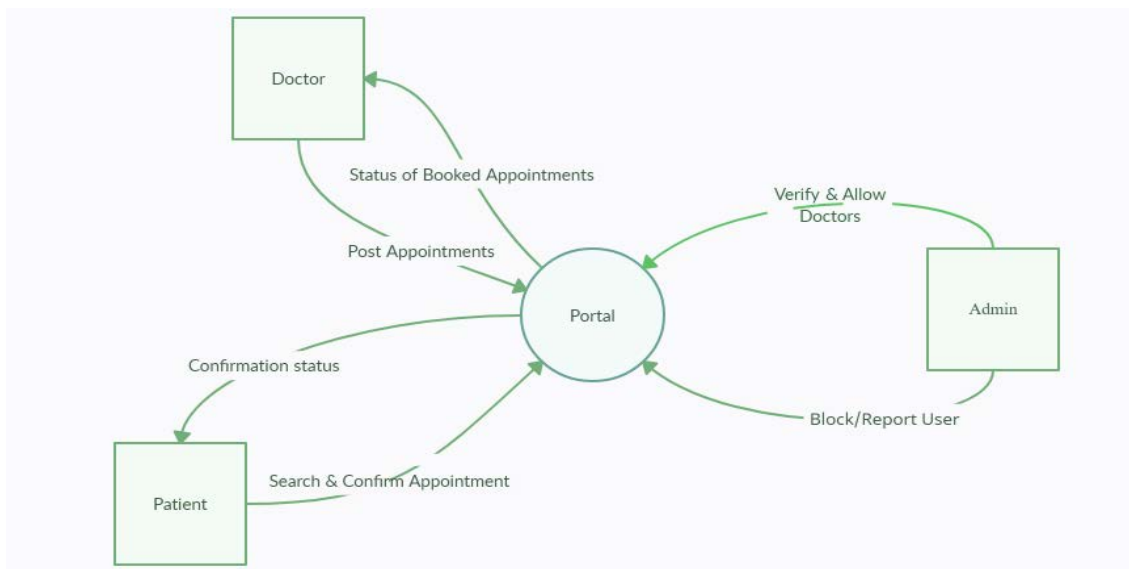


Figure 1: DFD Level 0 of Appointment Scheduling App.

## DFD Level 1:

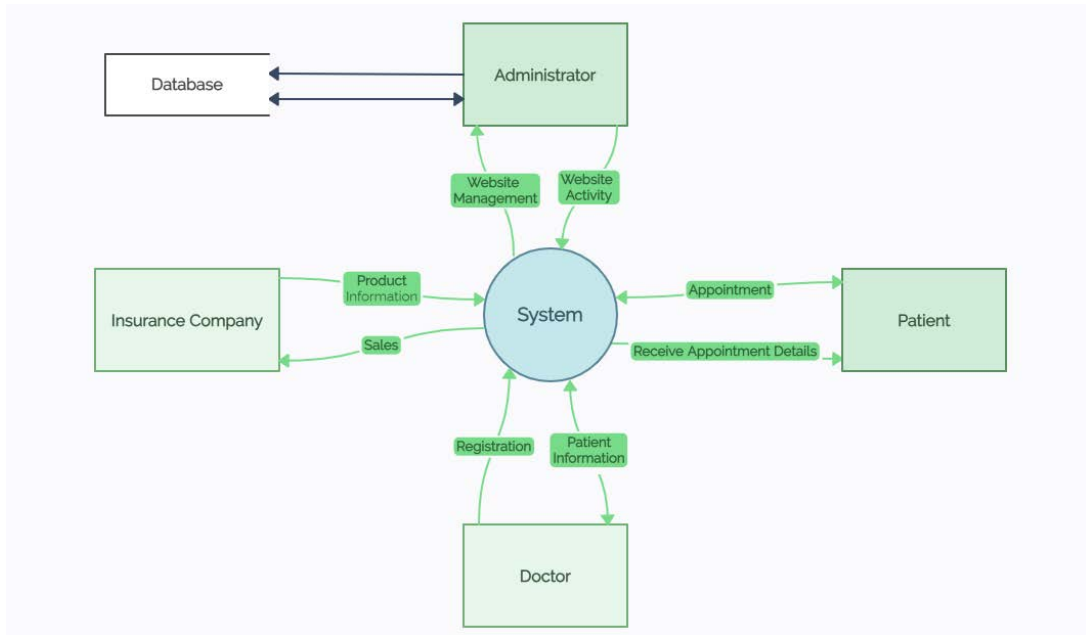


Figure 2: DFD Level 1 of Appointment Scheduling App.

## DFD Level 2:

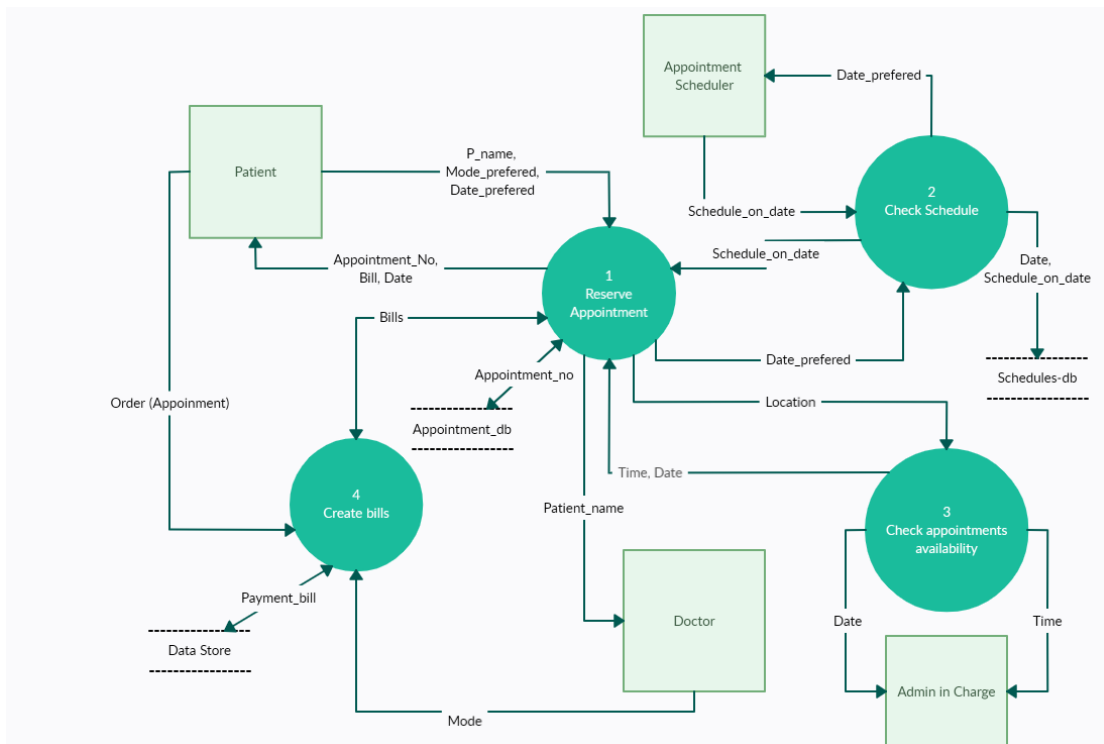


Figure 3: DFD Level 2 of Appointment Scheduling App.

### c. UML Use case

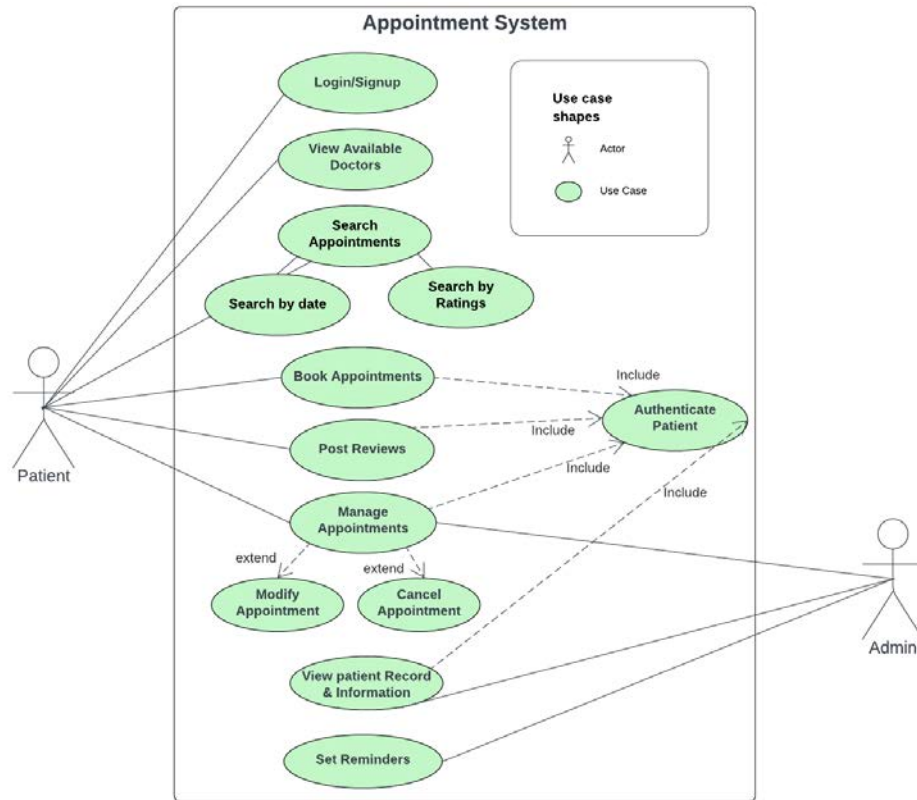


Figure 4: UML of Appointment Scheduling App.

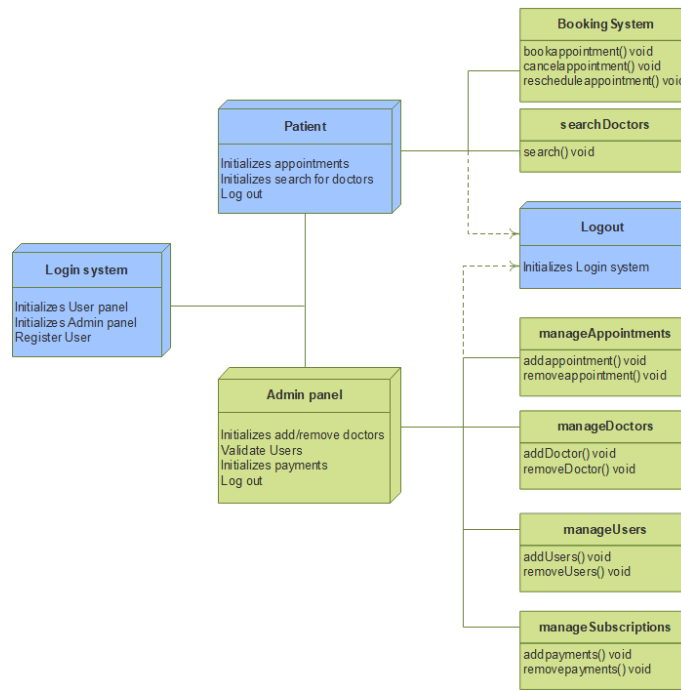


Figure 5: Deployment Diagram of Appointment Scheduling App.



## 3 System Architecture

### 3.1 Overview

The system architecture of our healthcare appointment scheduling app is intended to provide a scalable, dependable and secure platform for managing appointments in healthcare settings.

### 3.2 Architectural Design: Environment Review

#### a. Application Environment

**Frontend:** Web and mobile interfaces built with HTML, CSS, JavaScript, React.js, and React Native to provide cross-platform compatibility.

**Backend:** Node.js and Express.js are used for server-side functionality, API request handling, and database interactions.

**Database:** MongoDB is a database used to store patient records, appointment details, user profiles, and system configurations.

**External Services:** Integration of third-party APIs for payment processing, calendar synchronization, and communication (SMS, email).

**Cloud infrastructure** is deployed on cloud platforms such as AWS, Azure, or Google Cloud Platform to improve scalability, reliability, and accessibility.

#### b. Security and Compliance

**Data Encryption:** Ensure that all sensitive data, including patient information and appointment details, are encrypted both in transit and at rest to prevent unauthorized access.

**Compliance Standards:** Adhere to relevant healthcare regulations such as HIPAA (Health Insurance Portability and Accountability Act) to protect patient privacy and ensure data security.

**Audit Trails:** Maintain detailed audit logs to track access to patient records and appointment scheduling activities for compliance purposes.

**Security Testing:** Regularly conduct security assessments and penetration testing to identify and address potential vulnerabilities in the application.

#### c. Scalability & Performance

**Elastic Architecture:** Design the application with a scalable architecture that can easily accommodate increasing user loads and appointment scheduling demands.

**Load Balancing:** Implement load balancers to distribute incoming traffic evenly across multiple servers or instances to ensure optimal performance.

**Caching:** Utilize caching mechanisms to store frequently accessed data and reduce database load, improving overall system performance.

**Monitoring and Optimization:** Implement monitoring tools to continuously monitor application performance and identify bottlenecks, allowing for timely optimization and scaling.

**d. Availability & Disaster Recovery**

**High Availability Architecture:** Design the application with redundancy at every level to minimize downtime and ensure continuous availability of appointment scheduling services.

**Failover Mechanisms:** Implement automatic failover mechanisms to quickly redirect traffic to standby servers or instances in case of hardware or software failures.

**Data Backup and Recovery:** Regularly back up critical data and implement robust disaster recovery plans to ensure quick recovery in the event of data loss or system failure.

**Geographic Redundancy:** Utilize geographically distributed data centers or cloud regions to provide redundancy and mitigate the impact of regional outages or disasters.

**e. Development & Deployment Process**

**Continuous Integration/Continuous Deployment (CI/CD):** Implement CI/CD pipelines to automate the build, testing, and deployment processes, ensuring rapid and reliable delivery of updates and new features.

**DevOps Practices:** Foster collaboration between development and operations teams to streamline the development and deployment process, improving efficiency and reducing time to market.

**Version Control:** Utilize version control systems to track changes to code and configuration files, enabling easier collaboration and rollback to previous versions if needed.

**Testing and Quality Assurance:** Implement rigorous testing practices, including unit testing, integration testing, and automated testing, to ensure the reliability and quality of the application before deployment.

### **3.3 External Components**

#### **a. Electronic Health Record (EHR) Systems**

Integration with EHR systems enables healthcare providers to access patient records, medical histories, treatment plans, and other pertinent information during the appointment booking process. This integration supports continuity of treatment and allows healthcare providers to make informed decisions based on detailed patient data.

#### **b. Telehealth Platforms**

Integration with telehealth systems allows for virtual appointments and remote consultations between patients and healthcare providers. This enables consumers to arrange appointments, make video conversations, and receive medical advice from the comfort of their own homes, increasing accessibility and convenience for both patients and healthcare providers.

#### **c. Calendar Applications**

Users may synchronize their appointments across several platforms thanks to integration with popular calendar apps like Google Calendar, Outlook, and Apple Calendar. This provides consistency and helps customers stay organized by displaying appointments in their preferred calendar interface and providing reminders.

#### **d. Payment Gateways**

Integration with payment gateways allows for secure and convenient payment processing of appointment fees, co-payments, and other healthcare-related charges. This enables patients to pay online using credit/debit cards, digital wallets, or other payment methods, expediting the billing process and increasing financial transparency.

#### **e. SMS and Email Services**

Integration with SMS and email services enables the program to send automatic notifications, reminders, and alerts to patients and healthcare providers. This includes appointment reminders, confirmation messages, rescheduling notifications, and other appointment-related communications.

#### **f. Geolocation Services**

Integration with geolocation services enables the program to offer location-based features such as identifying nearby healthcare professionals, clinics, and medical facilities. This allows patients to easily find and arrange appointments with healthcare professionals in their area, increasing accessibility and minimizing travel time.

#### **g. Health Insurance Verification Systems**

Integration with health insurance verification systems allows healthcare practitioners to check patients' insurance coverage and eligibility in real time. This guarantees that patients have proper insurance information on file and allows healthcare professionals to accurately bill insurance companies for services provided during appointments.

### 3.4 Diagram

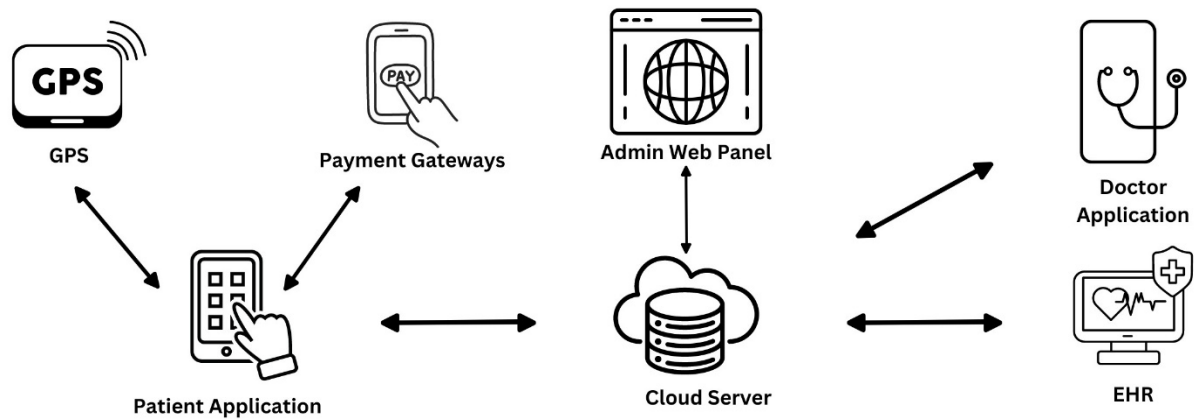


Figure 6: System Architecture Diagram of Appointment Scheduling App.

## 4 Data Design

### 4.1 Review Data Tables

Table Name	Description
Person	Stores personal information about individuals, such as name, date of birth, address, contact details, and login credentials.
Department	Contains data about different departments within the organization.
Patient	Stores information specific to patients, including PatientID, Weight, Height, InsuranceDetails, and other patient-related attributes. Has a relationship with the Person table, indicating that a patient is a type of person.
Doctor	Holds information about doctors, such as DoctorID, Speciality, and DeptID (referring to the department they belong to). Has a relationship with the Person table, indicating that a doctor is a type of person.
Patient History	Maintains the medical history of patients, including PatientID, Appointments, Allergies, Diagnosis, and DoctorID (referring to the doctor who treated the patient).
Appointment	Stores information about scheduled appointments, such as AppointmentID, PatientID (referring to the patient), DoctorID (referring to the doctor), Date, Time, Duration, and Status.
Schedule	Contains information about the schedule, including ScheduleID, DoctorID (referring to the doctor's schedule), Day, StartTime, EndTime, Room, and other schedule-related details.

Table 1: Data Tables of Appointment Scheduling App.

## 4.2 Class Diagram or ERD

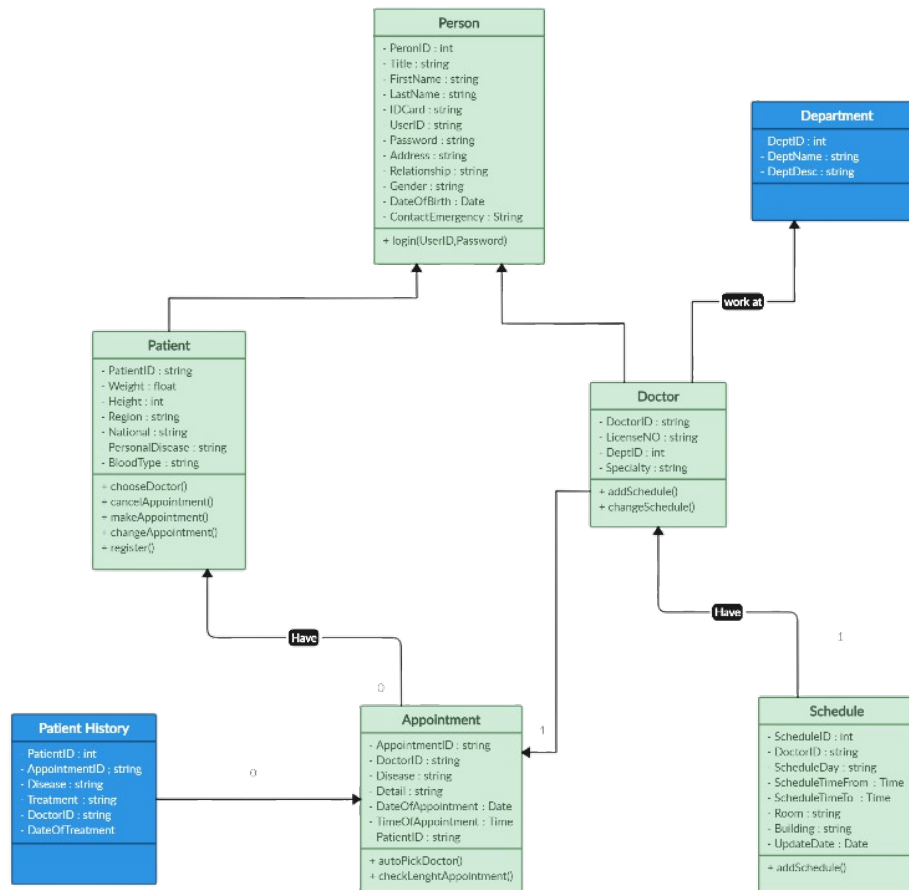


Figure 7: Class Diagram / ERD of Appointment Scheduling App.

## 5 Human Interface Design

### 5.1 Overview

Healthcare systems are typically complicated, with sophisticated workflows involving patients, healthcare professionals, and administrative personnel. An effective UI/UX design eliminates these complications and provides users with a seamless and intuitive experience. For patients, this means quicker access to information, more efficient appointment scheduling, and a clearer picture of their medical journey.

When designing a healthcare app, it is critical to deliberately use white space to highlight vital screen features. This generates visual contrast, allowing consumers to absorb information intuitively. Maintain a balance by separating distinct themes or functions with white space without going overboard, resulting in a smooth user

experience. Avoid overloading the interface with extraneous items just to cover blank spaces, as this might overwhelm consumers and detract from usability.

## 5.2 Screen Images



Figure 8: Login screen of Appointment Scheduling App.

### Screen Objectives and Actions:

- App logo "Medsphere"
- Two buttons: "Login" and "Signup"

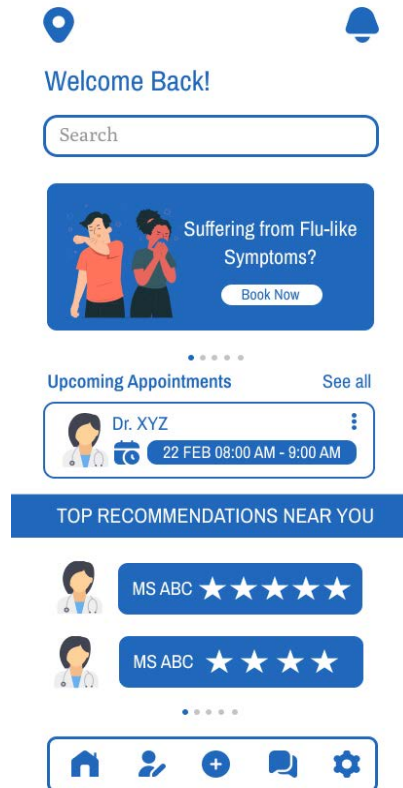


Figure 9: Dashboard of Appointment Scheduling App.

### Screen Objectives and Actions:

- Welcome screen with a search bar
- Advertisement banner for "Suffering from Flu-like Symptoms?" with a "Book Now" button
- Upcoming appointment notification with doctor's name and appointment details
- Section for "TOP RECOMMENDATIONS NEAR YOU" with doctor's ratings
- Navigation icons at the bottom for Home, Profile, New Appointments, Chat Support and Settings

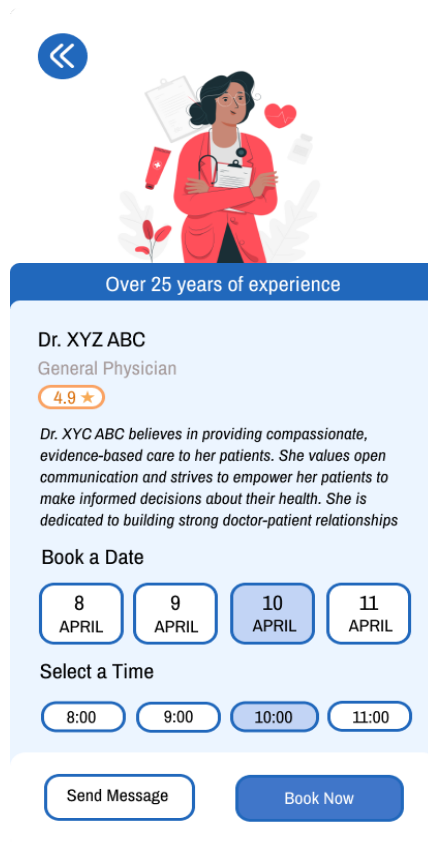


Figure 10: Schedule an appointment of Appointment Scheduling App.

### Screen Objectives and Actions:

- Doctor profile screen with a back button
- Doctor's photo and name
- Doctor's specialty "General Physician" and rating information
- Doctor's bio describing their approach and values
- Calendar with selectable dates for booking an appointment
- Time slot options for booking
- "Send Message" and "Book Now" button

## 6. Issues To Be Resolved

**Appointment Scheduling Constraints:** The system may need to handle various scheduling constraints, such as preventing double-booking of appointments, managing availability of doctors across multiple locations, and ensuring that appointments do not overlap with other scheduled events or appointments.