Palestine Technical University – Kadoorie

College of Engineering and Technology

Department of Computer Engineering

"Online Medical Appointment Management System"

Prepared by :

Batool Khaled Hussien (202210061) (sec 2)
Rayan Essa Awad (202210063) (sec 2)
Waad Abdalhafeeth Amer (202210307) (sec 2)
Waad Rabee Sammar (202210654) (sec 1)

Supervised by:
Dr. Nael Salman

Project submitted in Software Engineering Course Requirement for the Bachelor Degree Tulkarm, Palestine

2025

Software Requirements Specification (SRS)

Abstract

Going to medical clinics and waiting for appointments can be tiring and take a lot of time, especially with the high demand for healthcare. That is why we created an easy-to-use online system to manage medical appointments. This system makes booking simple and helps doctors and patients organize their schedules.

Patients and doctors can sign up on the platform. It shows available appointment times for each doctor, so patients can easily choose a suitable time. Doctors can also update their schedules when needed. Patients can cancel or change appointments through the platform, reducing waiting times and making services more efficient.

On our platform, patients can search for doctors by specialty, check ratings, and pick the best available appointment.

This system makes healthcare services easier for both doctors and patients. It helps reduce waiting times and improves the overall experience. We also follow international security and privacy standards to keep patient data safe, making our platform secure and easy to use.

قد يكون الذهاب إلى العيادات الطبية وانتظار المواعيد مرهقًا ويستغرق الكثير من الوقت، خاصةً مع زيادة الطلب على الخدمات الصحية. لهذا السبب، قمنا بإنشاء نظام إلكتروني سهل الاستخدام لإدارة المواعيد الطبية. هذا النظام يجعل الحجز بسيطًا ويساعد الأطباء والمرضى في تنظيم جداولهم.

يمكن للمرضى والأطباء التسجيل في المنصة. يُظهر النظام المواعيد المتاحة لكل طبيب، مما يسمح للمرضى باختيار الوقت المناسب بسهولة. كما يمكن للأطباء تحديث جداولهم عند الحاجة. يمكن للمرضى إلغاء أو تعديل المواعيد من خلال المنصة، مما يقلل من أوقات الانتظار ويحسّن كفاءة الخدمات.

على منصتنا، يمكن للمرضى البحث عن الأطباء حسب التخصص، الاطلاع على التقييمات، واختيار الموعد الأفضل المتاح.

هذا النظام يجعل الخدمات الصحية أسهل لكل من الأطباء والمرضى. كما يساعد على تقليل أوقات الانتظار وتحسين التجربة العامة. نلتزم أيضًا بمعايير الأمان والخصوصية العالمية لحماية بيانات المرضى، مما يجعل منصتنا آمنة وسهلة الاستخدام.

1. Introduction

1.1 Objective

Booking a doctor's appointment can be a cumbersome process, but it should be seamless, especially when dealing with sick patients. Studies show that over 67% of patients prefer to book appointments online, highlighting how technology is addressing a real-world problem.
Online booking offers several advantages. Increased flexibility allows patients to schedule appointments at their convenience without being restricted to clinic hours. Time and cost savings reduce the need for long phone calls or unnecessary trips to the clinic. Enhanced doctor-patient communication ensures better coordination, reminders, and follow-ups.
Lastly, online booking helps fill appointment gaps, optimizing schedules and reducing no-shows.

1.2 Document Conventions

This part of the SRS specifies the standards and conventions used throughout the document. For example, terms like "User" refer to any individual interacting with the system, including patients, doctors, and administrators. "Patient" refers to registered individuals managing appointments, while "Doctor" refers to healthcare providers offering consultations.

Additionally, this section defines formatting rules and notation used for diagrams, flowcharts, and structured elements to maintain consistency in documentation. Conventions such as bold text for headings, italic text for emphasis, and monospaced font for code snippets ensure clarity and uniformity across the document.

1.3 Intended Audience and Reading Suggestions

This document is structured to serve multiple stakeholders with distinct interests. Project managers should focus on the Project Scope (§1.4), Functional Requirements (§3), and System Models (§6) to understand implementation costs, timelines, and integration

requirements. Developers will find the Functional Requirements (§3), Software Interfaces (§4.3), and Class Diagrams (§6.4) most valuable for technical implementation. The Quality Assurance team should prioritize Non-Functional Requirements (§5) and Use Case Diagrams (§6.1) to develop comprehensive test cases addressing performance, security, and usability. Stakeholders (clinic owners) are directed to the Abstract, Product Features (§2.2), and Security Requirements (§5.3) to evaluate business value and operational alignment. End users (doctors and patients) will benefit most from reviewing User Documentation (§2.6) and User Interfaces (§4.1) sections to understand system registration, appointment booking, and daily usage procedures. This targeted approach ensures each audience efficiently accesses relevant information while maintaining document cohesion.

1.4 Project Scope

This system is designed to simplify and improve the process of booking and managing medical appointments. It allows patients to schedule, modify, or cancel appointments easily. Doctors can manage their schedules and track patient bookings efficiently.
The system securely stores patient and doctor information and provides a clear and organized interface for managing appointments. It can be used by hospitals, clinics, and independent medical practitioners to enhance scheduling efficiency and reduce administrative burdens.

2. Overall Description

2.1 Product Perspective

This system is an independent web application developed to optimize the scheduling of medical appointments. It integrates seamlessly into the larger healthcare framework, where clinics and hospitals seek to minimize manual tasks. Utilizing established technologies like web forms, user authentication, and calendar applications, it is specifically designed to enhance the interaction between doctors and patients.

2.2 Product Features

- Registration and Login for both doctors and patients.
- Search Feature to locate doctors based on specialty or location.
- Appointment Booking with real-time availability.
- Appointment Management options for rescheduling or cancellation.
- Doctor Dashboard for managing availability and viewing appointments.
- Patient Interface to monitor upcoming appointments and review doctor ratings.
- Notification System for appointment confirmations and reminders.
- Secure Data Management in accordance with privacy regulations.

2.3 User Classes and Characteristics

- Patients: without technical expertise looking for a fast and straightforward way to book appointments. They require an intuitive user interface and helpful guidance.
- Doctors: Healthcare professionals who oversee their availability. They need clear views of their schedules and the ability to easily update appointment times.
- Administrators: Individuals tasked with managing system users and monitoring platform performance. They may require more comprehensive dashboards and access control features.

2.4 Operating Environment

- Platform: Web-based; accessible via browsers (Chrome, Firefox, Edge).
- Backend: Hosted on a cloud server with secure database storage (e.g., MySQL, PostgreSQL).
- Frontend: Built using HTML/CSS, JavaScript frameworks (e.g., React or plain JS).
- Devices: Compatible with desktop, tablet, and mobile.
- Internet Connection: Required for access and updates.

2.5 Design and Implementation Constraints

- Security Standards: Must comply with data protection regulations (e.g., GDPR-like standards).
- Platform Limitation: Initial version is limited to web access only.
- User Authentication: Requires strong password policies and secure login/logout flows.
- Localization: Currently designed in English, other languages may require future updates.

2.6 User Documentation

- A User Guide is provided for patients and doctors explaining:
  → How to sign up/log in.
  → How to book or cancel appointments.
  → How to update availability (for doctors).
- Includes FAQ Section and Contact Support Info.
- Available as a downloadable PDF and in-app help section.

2.7 Assumptions and Dependencies

- Users have basic internet knowledge and access to a device with a browser.
- Clinics or doctors are responsible for keeping schedules up to date.
- System depends on email/SMS APIs for sending notifications.
- The application relies on stable hosting services and up-to-date browser support.

3. System Features

- **DESCRIPTION and PRIORITY**

  The **Online Medical Appointment Management System** enables patients and doctors to book and manage medical appointments through a user-friendly web interface. Instead of physically going to clinics and waiting, patients can browse available doctors, filter by specialty and location, and book appointments online at their convenience. The system improves time efficiency, enhances scheduling, and delivers a better overall healthcare experience. It is highly relevant in today's world where digital healthcare is increasingly preferred.

- **STIMULUS/RESPONSE SEQUENCES**
  → Patient searches for a doctor by name, specialty, or filter criteria
  → System displays matching doctors with availability
  → Patient selects a time and confirms booking
  → Patient can cancel or reschedule the appointment within system rules
  → Doctor is notified of bookings
  → Admin oversees and manages the system operations

3.1 Functional Requirements

3.1.1 User Registration and Login: FR-001

Description: The system shall allow patients and doctors to register by providing their name, email, password, and role. Users can then log in with their credentials.
Priority: High
Roles Involved:
→ **Patient/Doctor:** Register and log in to access services.
→ **Admin:** Can manage user accounts.

3.1.2 Doctor Browsing: FR-002

Description: The system shall allow patients to browse and search for doctors. Search can be filtered by specialty, rating, location, availability, and consultation type (online/in-person).
Priority: Medium
Roles Involved:

- → **Patient:** Searches and views doctor profiles.
- → **Doctor:** Profile is displayed with professional details.
- → **Admin:** Can add/edit doctor profiles and availability.

3.1.3 Appointment Booking: FR-003

Description: The system shall allow patients to book appointments by selecting a doctor, date, and available time slot.
Priority: High
Constraints:

- → Booking on holidays or weekends (if restricted) is not allowed.
- → Booking in the past is not permitted.
- → The system shall prevent double-booking of time slots.

Roles Involved:

- → **Patient:** Books appointment.
- → **Doctor:** Receives notification of new booking.
- → **Admin:** Can override restrictions and manage time slot availability.

3.1.4 Appointment Management: FR-004

Description: The system shall allow patients to cancel or reschedule their appointments within allowed policies.
Priority: Medium
Constraints:

- → Past appointments cannot be modified or canceled.
- → Same-day modifications may be restricted or require admin approval.

Roles Involved:

→ **Patient:** Can cancel or reschedule within limits.

→ **Doctor:** Can view changes to schedule.

→ **Admin:** Can manage cancellation rules and approve late changes.

3.1.5 Appointment Tracking: FR-005

Description: The system shall allow users to view their appointment history and upcoming bookings.

Priority: Medium

Roles Involved:

→ **Patient:** Views history and upcoming visits.

→ **Doctor:** Sees full schedule and appointments.

→ **Admin:** Can access full appointment records for monitoring.

3.1.6 Admin Interface: FR-006

Description: The system shall provide a dedicated interface for the administrator to manage users, doctor profiles, appointments, and system rules.

Priority: High

Admin Capabilities:

→ View, add, edit, or remove users (patients or doctors).

→ Manage doctor profiles and appointment availability.

→ Enforce business rules (e.g., holiday dates, max appointments/day).

→ Approve or reject sensitive actions like same-day cancellations.

→ Generate reports on booking statistics, user activity, and trends.

→ Send system-wide announcements or individual notifications.

3.1.7 Notification System: FR-007

Description:
The system shall notify users (patients, doctors, and admins) of important events such as new appointments, cancellations, changes, or system-wide announcements.
Priority: High
Roles Involved:
→ **Patient**: Receives confirmation and reminders.
→ **Doctor**: Gets notified of bookings, cancellations, or schedule changes.
→ **Admin**: Sends and manages system notifications.

3.1.8 Appointment Approval by Doctor: FR-008

Description:
The system may require doctors to **manually approve** appointment requests before they are confirmed. Once a patient books an appointment, the request is marked as **"Pending Approval"** until the doctor either accepts or rejects it.
Priority: Optional / Future Enhancement
Constraints:
→ Appointment remains unconfirmed until the doctor takes action.
→ If no response is received within a configurable time frame (e.g., 24 hours), the system may auto-decline or auto-approve based on system rules.
→ Patients are notified of approval or rejection.
Roles Involved:
Patient: Sees the status of pending appointments and gets notified of approval/rejection.
→ **Doctor**: Has the ability to approve or reject booking requests from patients.
→ **Admin**: Can enable or disable this feature for the whole system or specific doctors. May also override approvals in urgent cases.

3.2 Non-Functional Requirements:

3.2.1 Performance Requirements

R1: The system shall respond to any search or booking action within **3 seconds**.
R2: The system shall support up to **1000 concurrent users** without significant performance degradation.
R3: The system shall process **appointment creation, cancellation, or rescheduling** within 2 seconds under normal load.

3.2.2 Usability Requirements

R1: The system shall be **user-friendly** and suitable for users with no technical background.
R2: The user interface shall be **intuitive and responsive**, functioning well on both desktop and mobile devices.
R3: The system shall provide a **user manual or help section** explaining how to register, book, cancel, and manage appointments.

3.2.3 Security Requirements

R1: All sensitive user data (e.g., passwords, personal info) shall be encrypted using **SSL/TLS protocols**.
R2: The system shall implement **two-factor authentication** for doctors and admins.
R3: Access to user data shall be **restricted** and managed through role-based permissions.

3.2.4 Reliability Requirements

R1: The system shall have **99.9% uptime**, excluding scheduled maintenance.
R2: Daily **automated backups** shall be performed every 24 hours.
R3: In case of system failure, data shall be **restored within 2 hours**.

3.2.5 Maintainability Requirements

R1: The system shall allow easy updates for **doctor schedules, holidays, and policies**.
R2: The admin panel shall support **modification of rules** without technical expertise.
R3: Maintenance operations shall be performed in **low-traffic hours** and shall not impact core functionalities

3.2.6 Compatibility Requirements

R1: The system shall be compatible with the latest versions of **Chrome, Firefox, Safari, and Edge**.
R2: The platform shall be fully responsive and work on **iOS and Android** smartphones and tablets.
R3: The system shall operate smoothly on major operating systems, including **Windows, macOS, Android, and iOS**.

4. System Architecture

4.1 Overview

→ The medical appointment booking system is designed based on a standard Client/Server architecture. It is structured using multiple layers to promote maintainability, modularity, and scalability. The frontend communicates with the backend through secure HTTP(S) requests, and the backend interacts with the database to manage appointments, users, and doctors' data

4.2 Architectural Components

→ Frontend (Client-side): Built using HTML, CSS, JavaScript, and Bootstrap for responsiveness. It handles user interaction, navigation, and UI rendering for patients, doctors, and administrators.

→ Backend (Server-side): Developed using Node.js, it processes user requests, applies business logic (such as appointment validations), and communicates with the database. → Database: A MySQL database stores all system records including users, doctors, appointments, notifications, and availability schedules.

→ Hosting Environment: The platform is intended to be hosted on a cloud-based server for availability and scalability.

4.3 System Layers

→ Presentation Layer: Responsible for user interaction through the browser. Interfaces include Home Page, Login/Register, Appointment Booking, Doctor Dashboard, and Admin Panel.

→ Business Logic Layer: Manages core functions such as appointment scheduling, doctor availability, cancellation policies, and approval workflows (e.g., optional doctor approval).

→ Data Access Layer: Interfaces with the MySQL database to query, insert, update, and delete data related to appointments, users, and settings.

4.4 Communication Model

→ All communication between the frontend and backend occurs over secure HTTP/HTTPS protocols.

→ Integration with external APIs such as WhatsApp or email services for appointment notifications.

4.5 Deployment Environment

→ The system is deployed on a web server such as Apache or Nginx.

→ It supports all major web browsers: Chrome, Firefox, Safari, and Edge. →

→ The design is responsive, enabling access from desktop computers, laptops, tablets, and smartphones.

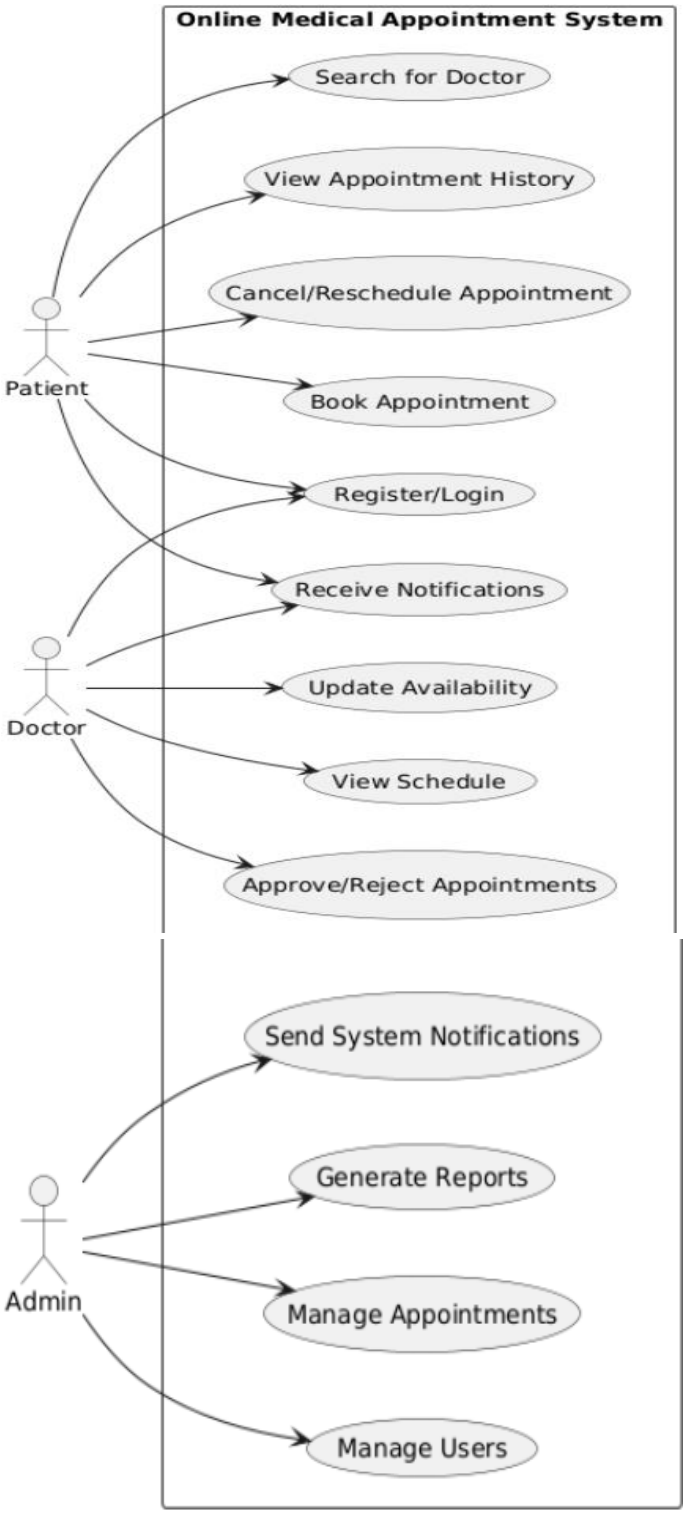
5. System Models

5.1 Use Case Diagram

This section describes the main interactions between users (actors) and the system in the form of use cases. The use case diagram visually represents the functionality that the system provides and how each type of user interacts with it.

The main goal of the system is to simplify the process of booking, canceling, and managing medical appointments online. Each user role has a specific set of interactions with the system.

→ **Patient**: A registered user who can search for doctors, book appointments, cancel or reschedule them, and receive notifications.
→ **Doctor**: A medical professional who can manage their appointment schedule, view upcoming appointments, and update availability.
→ **Admin**: A system administrator who can manage user accounts, oversee doctor schedules, and maintain the system's data integrity.


Online Medical Appointment System
Search for Doctor
View Appointment History
Cancel/Reschedule Appointment
Book Appointment
Register/Login
Receive Notifications
Update Availability
View Schedule
Approve/Reject Appointments
Send System Notifications
Generate Reports
Manage Appointments
Manage Users
Patient
Doctor
Admin

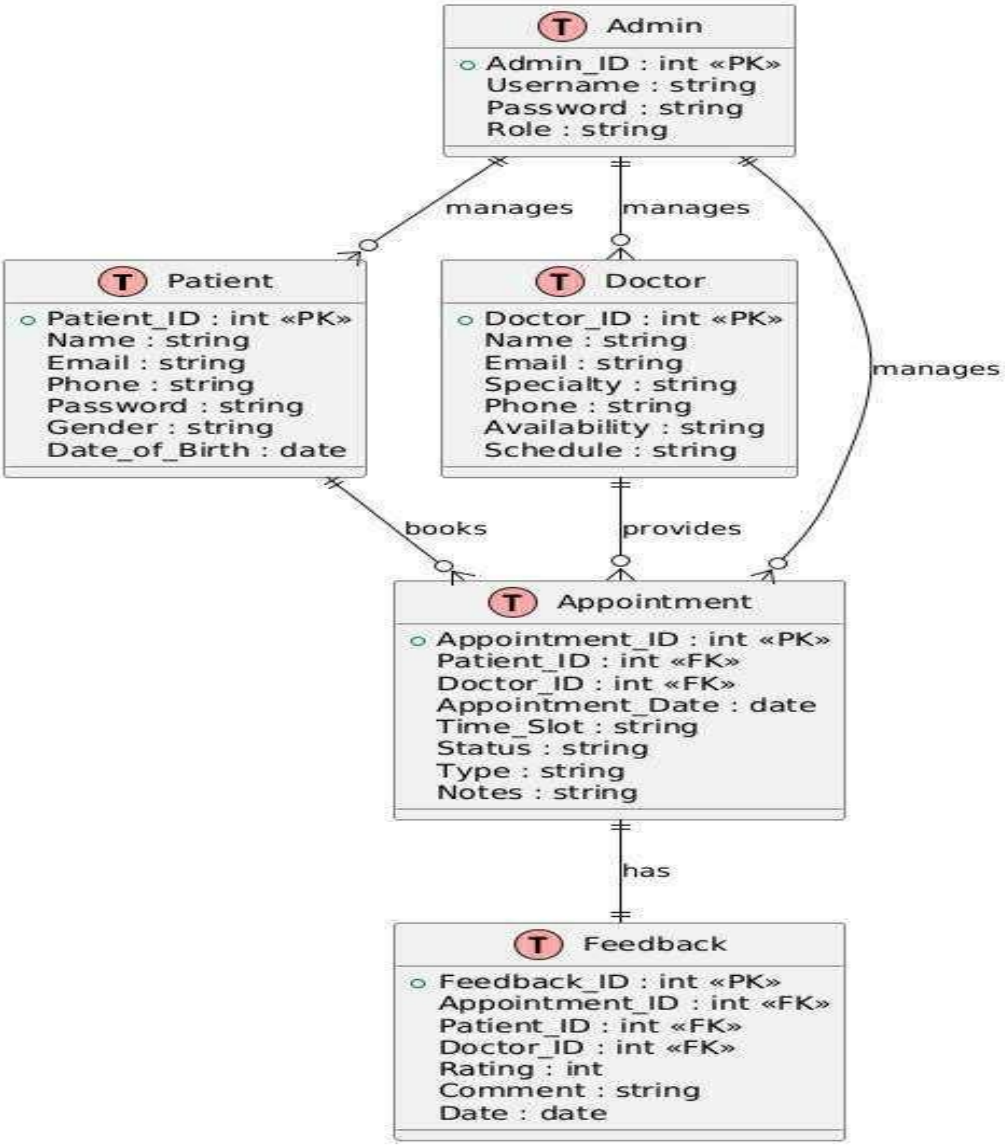
5.2 Class Diagram

A Class Diagram is a static structure diagram in the Unified Modeling Language (UML) that shows the system's classes, their attributes, and the relationships among the objects. It is used primarily in object-oriented software design to illustrate the system architecture and logic at a high level.

Components :

- **Classe:**
  → Patient: Patient_ID, Name, Email, Phone, Password, Gender, Date_of_Birth
  → Doctor: Doctor_ID, Name, Email, Specialty, Phone, Availability, Schedule
  → Appointment: Appointment_ID, Patient_ID, Doctor_ID, Appointment_Date, Time_Slot, Status
  → Admin: Admin_ID, Username, Password, Role
  → Feedback: Feedback_ID, Patient_ID, Doctor_ID, Rating, Comment, Date.

- **Relationships:**
  → A Patient can have multiple Appointments and can leave multiple Feedback entries.
  → A Doctor can have many Appointments and receive Feedback.
  → An Admin is responsible for managing the system, including appointments and users.



5.3 Sequence Diagram

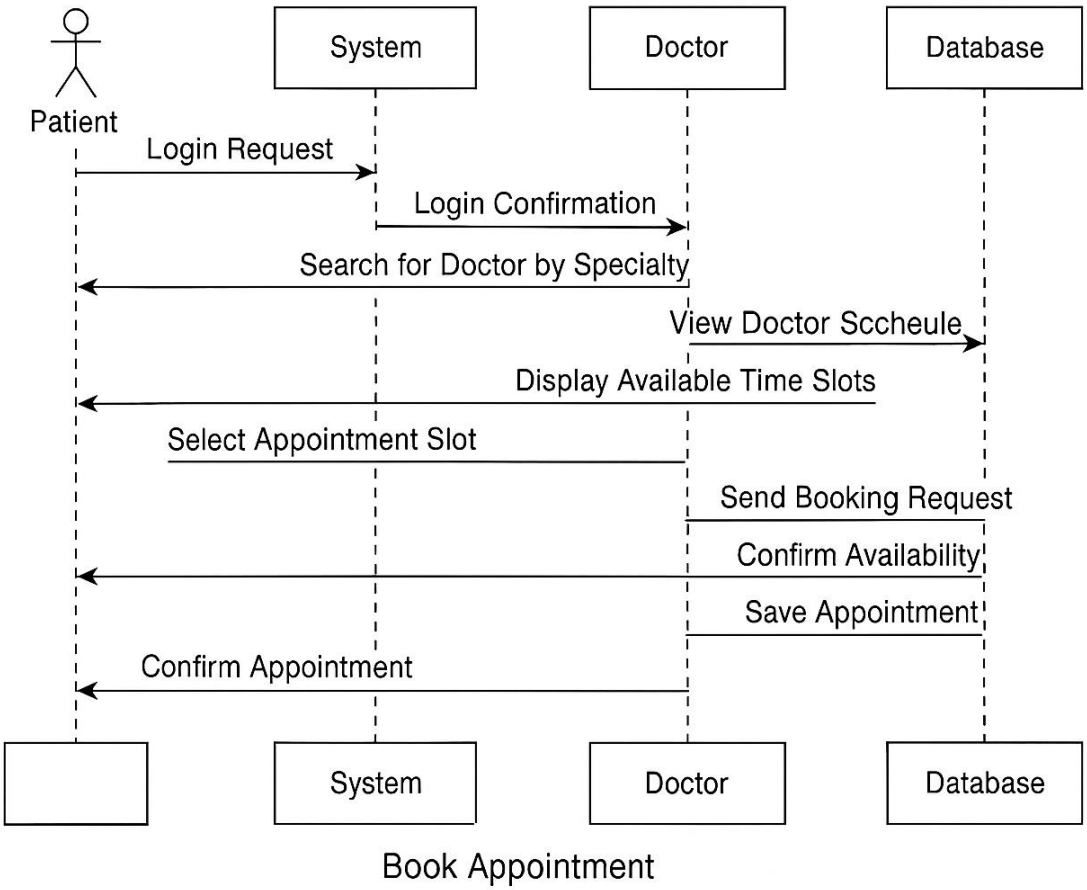
A Class Diagram is a static structure diagram in the Unified Modeling Language (UML) that shows the system's classes, their attributes, and the relationships among the objects. It is used primarily in object-oriented software design to illustrate the system architecture and logic at a high level.

A sequence diagram is a type of UML diagram that depicts the interactions between users and system components in chronological order. It illustrates the flow of messages between objects to accomplish a specific use case.

Use Case: Book Appointment

Participants:

- Patient (Actor)
- System (Object)
- Doctor (Object)
- Database (Object)



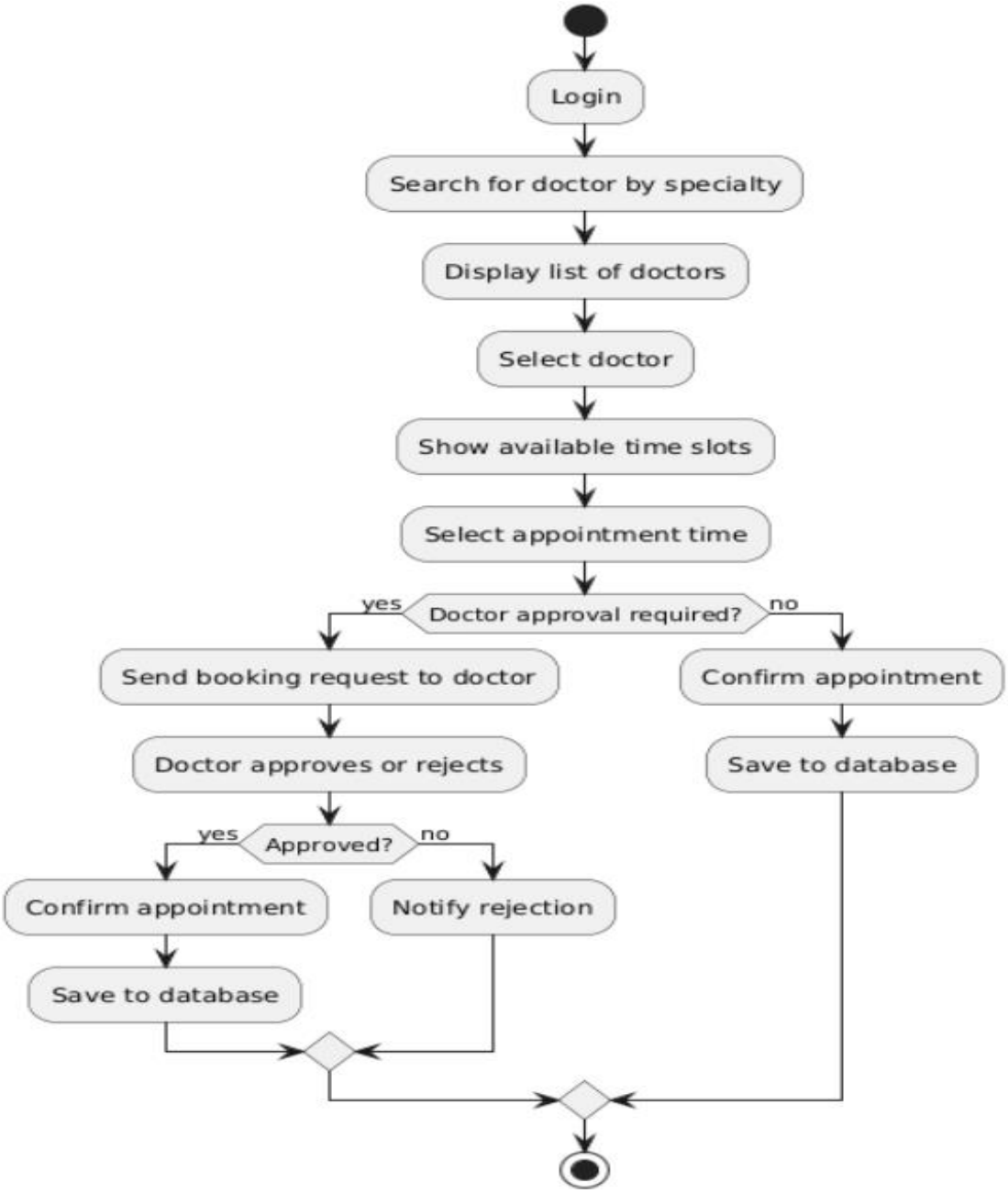
Book Appointment

5.4 Activity Diagram

An Activity Diagram is a type of UML diagram used to represent the flow of control between various activities or operations within the system. It shows the dynamic aspects of the system by modeling the workflow from start to finish.

It is especially useful for visualizing the appointment booking process in the medical system.

Use Case: Book an Appointment


Login
Search for doctor by specialty
Display list of doctors
Select doctor
Show available time slots
Select appointment time
Doctor approval required?
yes
no
Send booking request to doctor
Confirm appointment
Doctor approves or rejects
Save to database
Approved?
yes
no
Confirm appointment
Notify rejection
Save to database

5.5 Entity relationship Diagram (ERD)

An Entity Relationship Diagram (ERD) is a visual representation of the data structure used within a system. It shows how entities (tables) such as Patient, Doctor, Appointment, Admin, and Feedback are related to one another through connections and identifiers.

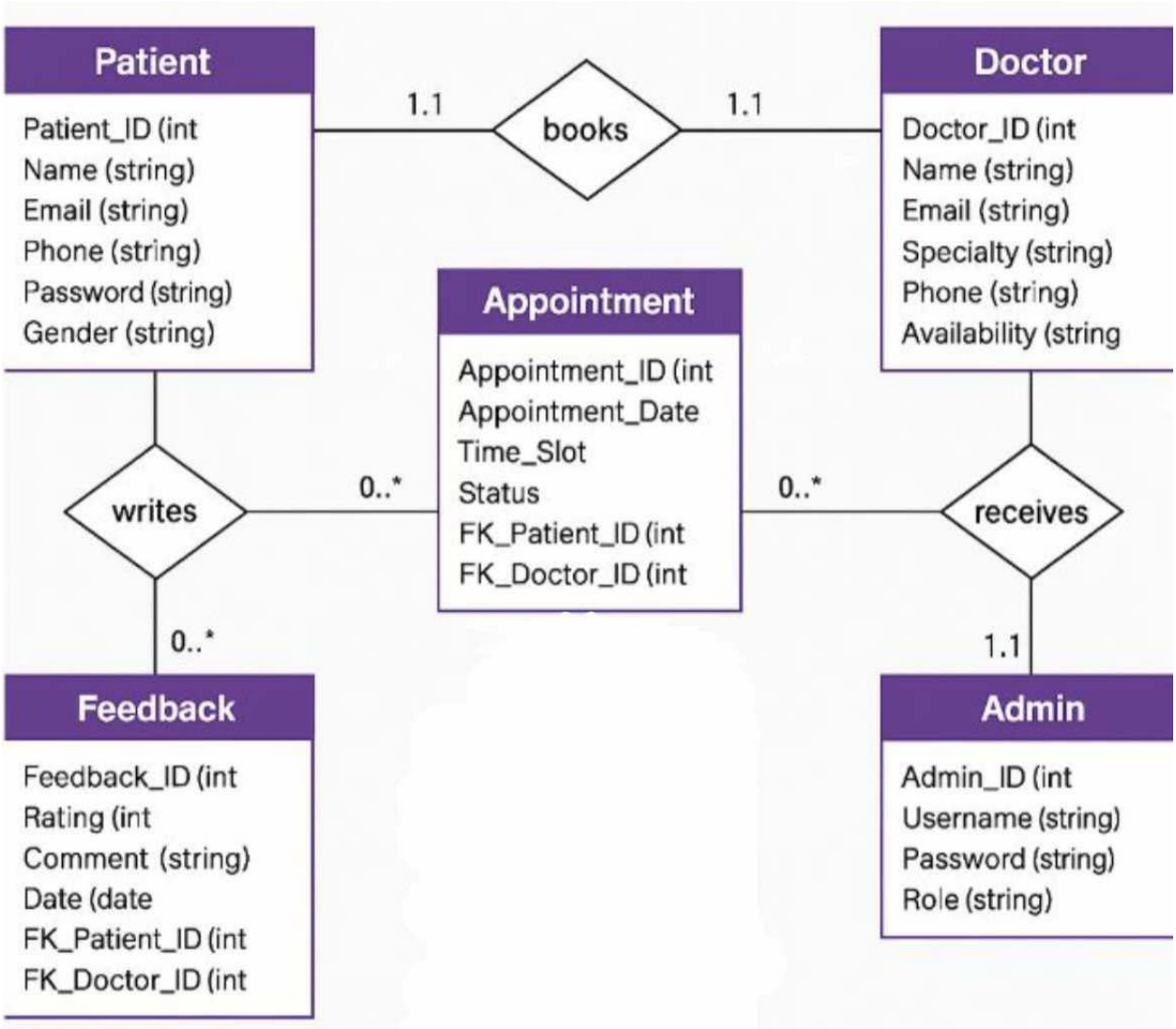
Main Components of the ERD:

Entities: Represented as boxes, each containing a set of attributes. For example:

- Patient: Patient_ID, Name, Email, Phone, etc.
- Doctor: Doctor_ID, Name, Specialty, Availability, etc.
- Appointment: Appointment_ID, Patient_ID, Doctor_ID, Date, Status, etc.
- Attributes: The data fields that describe each entity. For example, "Name" and "Email" are attributes of the Patient entity.
- Relationships: The lines connecting entities indicate how data in one table relates to another. For instance:

  → A Patient can book multiple Appointments.

  → A Doctor can be assigned to many Appointments.
  → A Patient can give Feedback for a Doctor after an Appointment.
  → Admins manage or validate Appointments.


Patient
Patient_ID (int
Name (string)
Email (string)
Phone (string)
Password (string)
Gender (string)
1.1
books
1.1
Doctor
Doctor_ID (int
Name (string)
Email (string)
Specialty (string)
Phone (string)
Availability (string
Appointment
Appointment_ID (int
Appointment_Date
Time_Slot
Status
FK_Patient_ID (int
FK_Doctor_ID (int
writes
0..*
0..*
receives
0..*
1.1
Feedback
Feedback_ID (int
Rating (int
Comment (string)
Date (date
FK_Patient_ID (int
FK_Doctor_ID (int
Admin
Admin_ID (int
Username (string)
Password (string)
Role (string)

6. Design Model (Interface)

6.1 User Interface Overview

The "Online Medical Appointment Management System" is designed to provide an intuitive and user-friendly experience for all types of users, including patients, doctors, and administrators. The interface is responsive, accessible, and aligns with the system's goal of streamlining the appointment booking process and clinic management.

Users interact with the system through key pages such as the Home Page, Doctor Search, Appointment Booking, Patient Dashboard, and Admin Panel. The layout follows a clean structure with a fixed navigation bar, organized content sections, and clearly defined buttons to guide users through each task.

In addition to the main functional pages, the platform includes a secure and accessible Login / Sign Up page where users can register or log in smoothly. The form includes realtime validation, informative prompts, and error messages to ensure a hassle-free experience for new and returning users.

The system layout is optimized for all devices, offering a consistent and smooth interface whether accessed from a desktop, tablet, or mobile phone. The design uses neutral colors, readable fonts, and appropriate spacing to enhance usability and reduce cognitive load.

6.2 Screenshots of User Interface

Below are screenshots representing the key interfaces of the Online Medical Appointment Booking System. Each screenshot showcases a core feature or page of the platform as implemented:

Log In

Email:

password:

Create Account

Log In

Register

Name:

Email:

Password:

Phone Number:

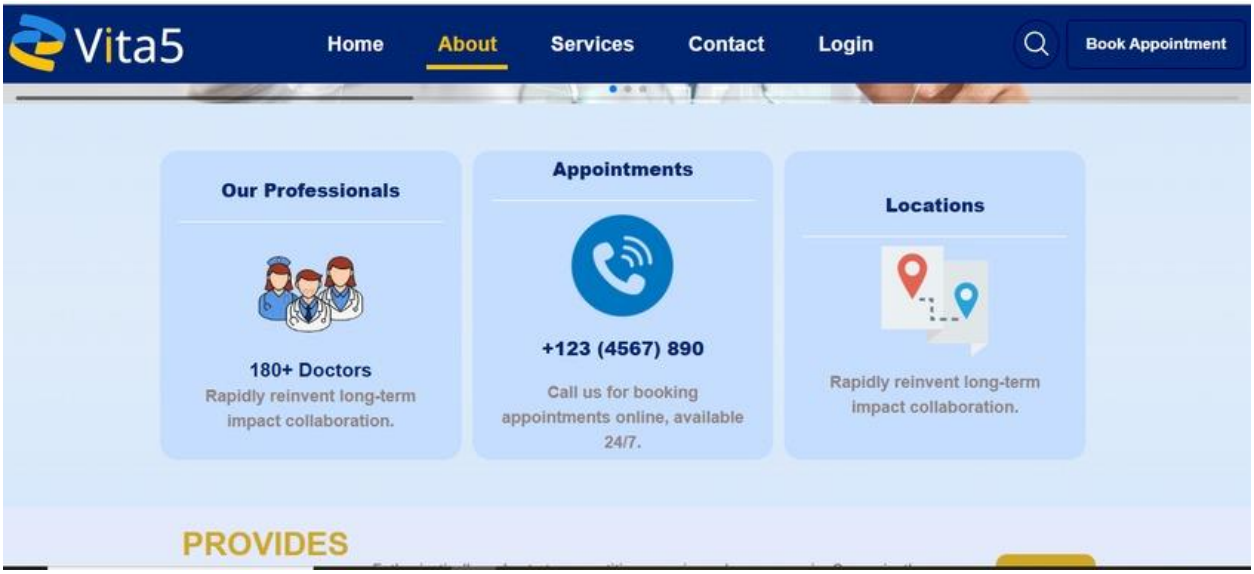
Are you a doctor or a patient?

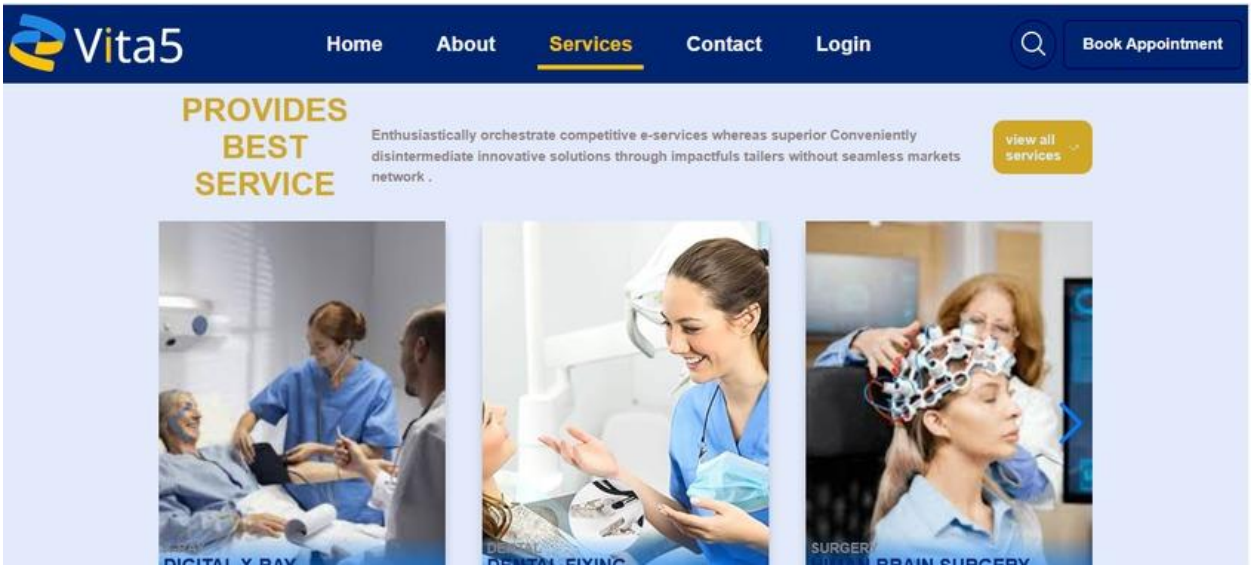
Patient

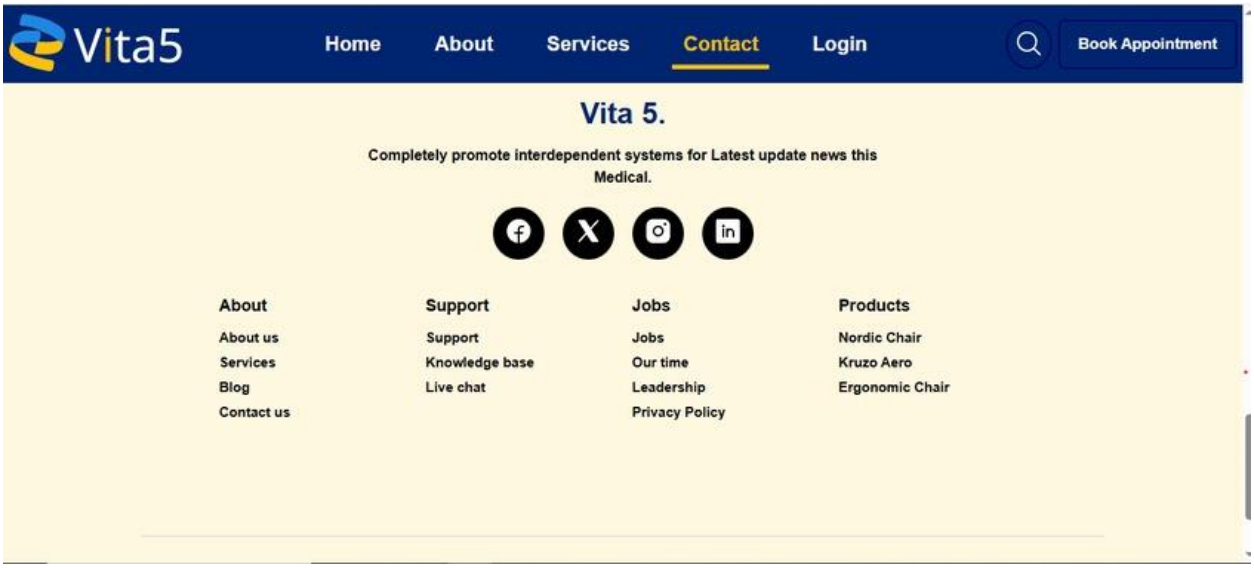
Already have an account?

Register

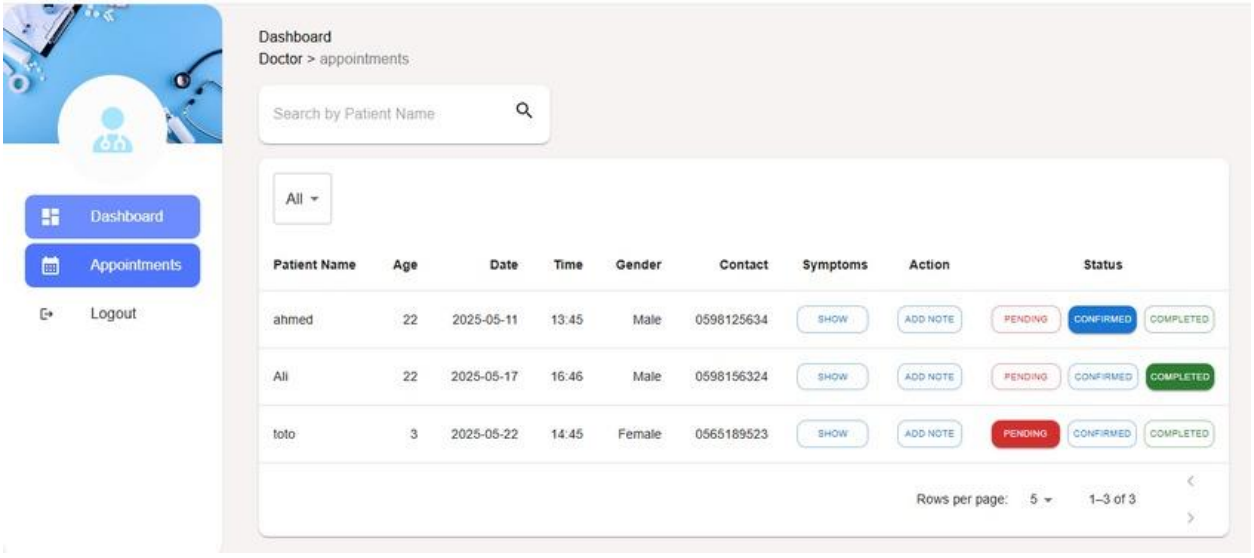

Vita5
Home
About
Services
Contact
Login
Book Appointment
Welcome to Our Healthcare System
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Learn More


Vita5
Home
About
Services
Contact
Login
Book Appointment
Our Professionals
180+ Doctors
Rapidly reinvent long-term impact collaboration.
Appointments
+123 (4567) 890
Call us for booking appointments online, available 24/7.
Locations
Rapidly reinvent long-term impact collaboration.
PROVIDES

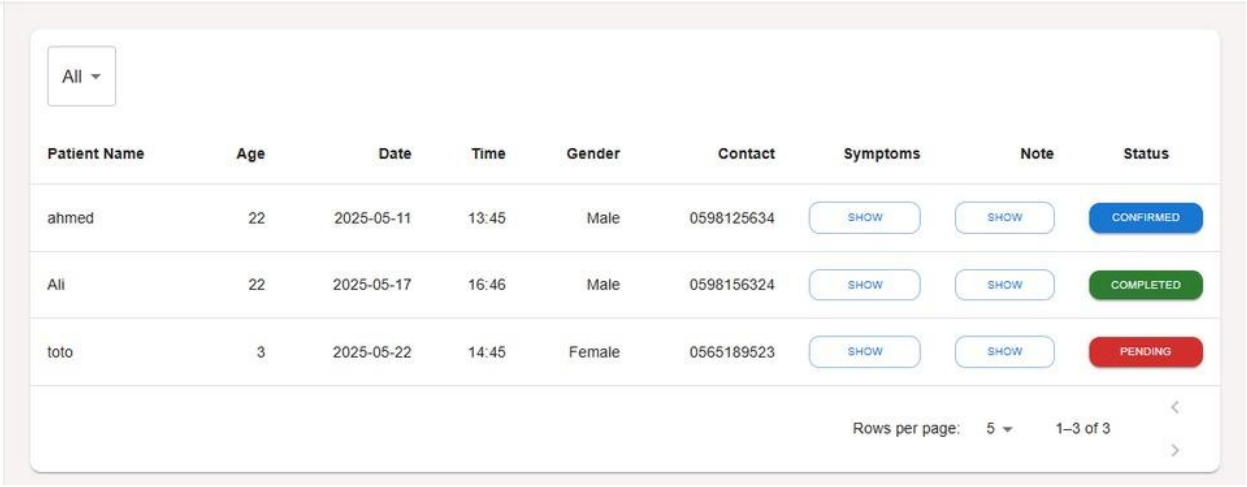

Vita5
Home
About
Services
Contact
Login
Book Appointment
PROVIDES BEST SERVICE
Enthusiastically orchestrate competitive e-services whereas superior Conveniently disintermediate innovative solutions through impactfuls tailers without seamless markets network .
view all services


Vita5
Home
About
Services
Contact
Login
Book Appointment
Vita 5.
Completely promote interdependent systems for Latest update news this Medical.
About
About us
Services
Blog
Contact us
Support
Support
Knowledge base
Live chat
Jobs
Jobs
Our time
Leadership
Privacy Policy
Products
Nordic Chair
Kruzo Aero
Ergonomic Chair


Dashboard
Doctor >
Total Patients
3
15% From Last Week
Patients Today
0
15% From Last Week
Appointments Today
0
15% From Last Week
Dashboard
Appointments
Logout
Pending
Confirmed
5
4
3
2
1
0
2025-05-22
2025-05-11
2025-05-17
Appointments


Dashboard
Doctor > appointments
Search by Patient Name
All
Dashboard
Appointments
Logout
Patient Name
Age
Date
Time
Gender
Contact
Symptoms
Action
Status
ahmed
22
2025-05-11
13:45
Male
0598125634
SHOW
ADD NOTE
PENDING
CONFIRMED
COMPLETED
Ali
22
2025-05-17
16:46
Male
0598156324
SHOW
ADD NOTE
PENDING
CONFIRMED
COMPLETED
toto
3
2025-05-22
14:45
Female
0565189523
SHOW
ADD NOTE
PENDING
CONFIRMED
COMPLETED
Rows per page: 5
1–3 of 3


Dashboard
My Appointments
Book Appointment
Logout
Health Records
Heart Rate: 140 BPM
Body Temperature: 37.5 °C
Glucose Level: 70 - 90 mg/dL
Blood Pressure: 100/70 mmHg
BMI: 20.1 kg/m²
Health Report
140
105
70
35
0
Heart Rate
Temperature
Upcoming Appointment
Dr. waed - 2025-05-22, 14:45
May 2025
S M T W T F S
27 28 29 30 1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31


All
Patient Name Age Date Time Gender Contact Symptoms Note Status
ahmed 22 2025-05-11 13:45 Male 0598125634 SHOW SHOW CONFIRMED
Ali 22 2025-05-17 16:46 Male 0598156324 SHOW SHOW COMPLETED
toto 3 2025-05-22 14:45 Female 0565189523 SHOW SHOW PENDING
Rows per page: 5
1–3 of 3

Conclusion

The **Online Medical Appointment Booking System** is designed to make it easier for patients to book doctor appointments. Instead of calling or visiting the clinic, patients can check available times, choose a doctor, and book directly online. This saves time and makes the process simple and convenient.

The system works well on all devices, including smartphones, tablets, and computers. It gives a smooth experience for both patients and healthcare providers. Clinics can manage appointments more easily, with fewer mistakes and less manual work.

Key features include real-time doctor availability, automatic reminders, optional doctor approval for bookings, and secure login. All personal and medical data is protected using strong security measures to keep the system safe and trustworthy.

Overall, the system improves the healthcare experience by making it faster, easier, and more organized. It helps both patients and doctors, and it's ready for future features like telemedicine and access to medical records.

References

- 830-1984 - IEEE Guide for Software Requirements Specifications.
- Software Requirements Specification document with example - Krazytech.com
- https://www.geeksforgeeks.org/software-engineering/
- Krazytech. (n.d.). Sample SRS Document for Software Projects. Retrieved from https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database
- Zumper.com
- Apartments.com
- Apartable.com
- homes.com9