Foundation of Advancement Of Science and Technology

National University of computer and emerging sciences

SOFTWARE REQUIREMENT SPECIFICATIONS

APPOINTDOC
DOCTOR APPOINTMENT SYSTEM
VERSION 1.0

INSTRUCTOR MS. JAVERIA FAROOQ (BCS-6E)
SOFTWARE ENGINEERING (CS-3009)

- MUHAMMAD HAMZA (K21-4579)
- MUHAMMAD SALAR (K21-4619)
- EMMANUEL (K21-4871)
- RITESH KUMAR (K21-3961)

Foundation of Advancement of Science and Technology
National University of Computer and Emerging Sciences
Department of Computer Science
Karachi, Pakistan
Sunday, March 31, 2024

Contents

SOFTWARE REQUIREMENT SPECIFICATIONS

1 Introduction

1.1 Motivation

The APPOINTDOC system aims to streamline the process of scheduling and managing doctor appointments. In today's fast-paced healthcare environment, efficient appointment management is crucial for both patients and medical practitioners. By providing an intuitive and reliable platform, APPOINTDOC seeks to enhance the patient experience and optimize doctors' schedules.

1.2 Stakeholders

The following stakeholders are involved in the APPOINTDOC system:

- **Patients**:
  - Users seeking medical appointments.
  - Primary interaction: Booking, rescheduling, and canceling appointments.
  - Requirement elicitation: Through user interviews and feedback.
- **Doctors and Medical Staff**:
  - Responsible for managing appointments.
  - Primary interaction: Viewing schedules, confirming appointments.
  - Requirement elicitation: Interviews with doctors and staff.
- **Administrators**:
  - Oversee system configuration, user management, and security.
  - Primary interaction: Managing user accounts, system settings.
  - Requirement elicitation: Interviews with hospital administrators.
- **Developers**:
  - Technical team responsible for system design, development, and maintenance.
  - Primary interaction: Implementing features, ensuring system functionality.
  - Requirement elicitation: Collaboration with other stakeholders.

1.3 Assumptions and Dependencies

To ensure successful implementation, we make the following assumptions and dependencies:

- **Reliable Internet Connectivity**: Users have access to stable internet connections for seamless interaction with the system.
- **Integration with Existing Systems**: APPOINTDOC integrates with hospital databases for patient records and other relevant data.
- **Compliance with Healthcare Regulations**: The system adheres to privacy laws and other healthcare-related regulations.

2 Functional Requirements

2.1 Appointment Booking

2.1.1 Feature: Book Appointment

- **Description**: Patients can schedule appointments with specific doctors on preferred dates and times.
- **User Interaction**: Patients provide their details, select a doctor, and choose an available time slot.
- **System Behavior**: The system confirms the appointment and updates the doctor's schedule.

2.1.2 Feature: View Appointments

- **Description**: Doctors and staff can view their upcoming appointments.
- **User Interaction**: Doctors log in and access their appointment list.
- **System Behavior**: Displays a list of scheduled appointments with patient details.

2.2 Appointment Management

2.2.1 Feature: Cancel Appointment

- **Description**: Patients can cancel booked appointments.
- **User Interaction**: Patients log in, select the appointment, and cancel.
- **System Behavior**: Updates the appointment status and notifies the doctor.

2.2.2 Feature: Reschedule Appointment

- **Description**: Patients can request to reschedule appointments.
- **User Interaction**: Patients provide reasons and preferred new time slots.
- **System Behavior**: Notifies the doctor and suggests alternative slots.

3 Non-Functional Requirements

3.1 Performance

- **Response Times**: The system should respond within seconds for most interactions.
- **Throughput**: Handle concurrent users during peak hours.

3.2 Reliability

- **Availability**: The system should be available 24/7, with minimal downtime for maintenance.
- **Fault Tolerance**: Regularly back up appointment data to prevent loss.

3.3 Security

- **Authentication**: Users must log in with valid credentials.
- **Authorization**: Role-based access control (patient, doctor, admin).
- **Data Privacy**: Protect patient information according to privacy laws.

3.4 Usability

- **Accessibility**: Ensure the system is usable by people with different needs.
- **User Interface Design**: Intuitive and consistent design for easy navigation.

3.5 Other

- **Scalability**: Design the system to accommodate future growth.
- **Data Privacy**: Protect patient information (compliance with privacy laws).
- **Auditability**: Maintain detailed logs for troubleshooting and accountability.
- **Error Handling**: Clear procedures for unexpected scenarios.

4 Constraints

4.1 Budget Constraints

- The project must operate within a specified budget.
- Optimize costs while delivering essential features.

4.2 Time Constraints

- Develop and deploy the system within the agreed timeframe.
- Use agile methodologies to manage time effectively.

4.3 Resource Constraints

- Personnel availability (developers, designers).
- Infrastructure capacity (servers, network).
- Software tools (licensing, open-source alternatives).

4.4 Risk Constraints

- Identify and manage project risks proactively.

4.5 Technology Constraints

- Compatibility with existing systems.
- Choose technologies with long-term support.

4.6 Compliance Constraints

- Legal and regulatory compliance (data privacy, healthcare laws).

4.7 Organizational Structure Constraints

- Clear communication channels and decision-making processes.

5 Architecture Design Overview

The APPOINTDOC system architecture includes several high-level components that interact seamlessly to create a robust and efficient doctor appointment system:

5.1 User Interface (UI)

- The **UI** serves as the primary interaction point for users (patients, doctors, and administrators).
- Key features include:
  - **Appointment Booking**: Patients can easily schedule appointments by selecting doctors, preferred dates, and available time slots.
  - **View Appointments**: Doctors and staff access their upcoming appointment lists, including patient details.
  - **Cancellation and Rescheduling**: Patients can cancel or request rescheduling of appointments.

5.2 Backend Services

- The **Backend Services** layer handles critical business logic and system functionality.
- Responsibilities:
  - **Authentication and Authorization**: Validates user credentials and controls access based on roles (patient, doctor, admin).
  - **Appointment Management**: Manages appointment scheduling, availability, and notifications.
  - **Data Processing**: Processes user requests, updates schedules, and communicates with the database.

5.3 Database

- The **Database** stores essential data related to appointments, patients, doctors, and administrative settings.
- Components:
  - **Appointment Records**: Store details of scheduled appointments (patient ID, doctor ID, date, time).
  - **User Profiles**: Maintain user information (name, contact details, role).
  - **Doctor Schedules**: Track doctors' availability.

5.4 Authentication and Authorization

- **Authentication** ensures secure access to the system:
  - Users log in with valid credentials (username/password or other authentication methods).
  - Sessions are managed securely.
- **Authorization** controls user permissions:
  - Role-based access (patient, doctor, admin).
  - Fine-grained permissions for specific features (e.g., only doctors can view patient medical history).

5.5 Notification Service

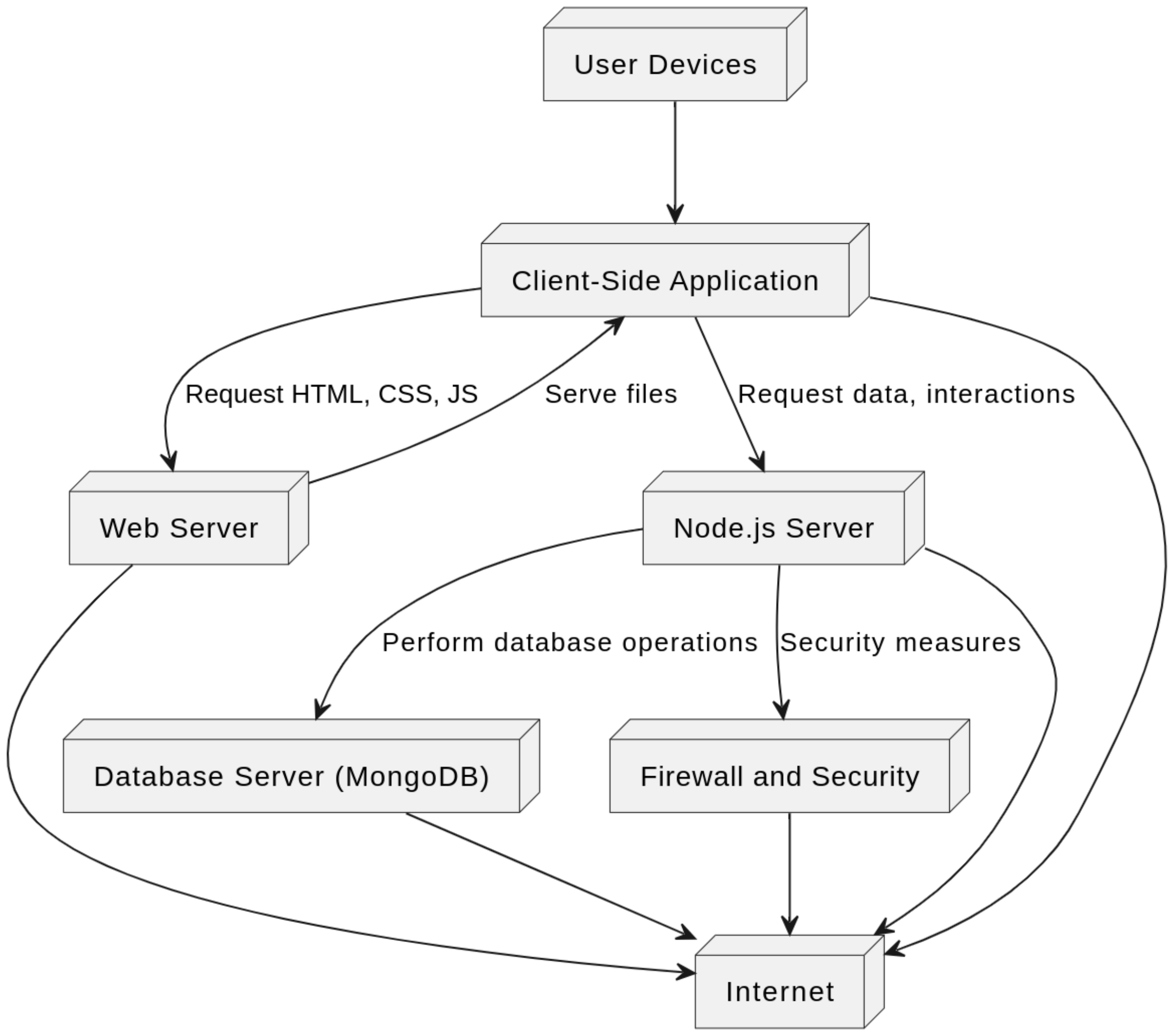
- The **Notification Service** keeps users informed:
  - **Appointment Reminders**: Sends reminders to patients and doctors before scheduled appointments.
  - **Critical Alerts**: Notifies administrators about system events (e.g., server downtime, security breaches).

5.6 Integration with Existing Systems

- APPOINTDOC integrates seamlessly with other hospital systems:
  - **Patient Records**: Integrates with existing databases to retrieve patient information.
  - **Billing and Insurance**: Shares relevant data for billing and insurance purposes.
  - **Electronic Health Records (EHR)**: Ensures consistency across systems.

5.7 Component Diagram

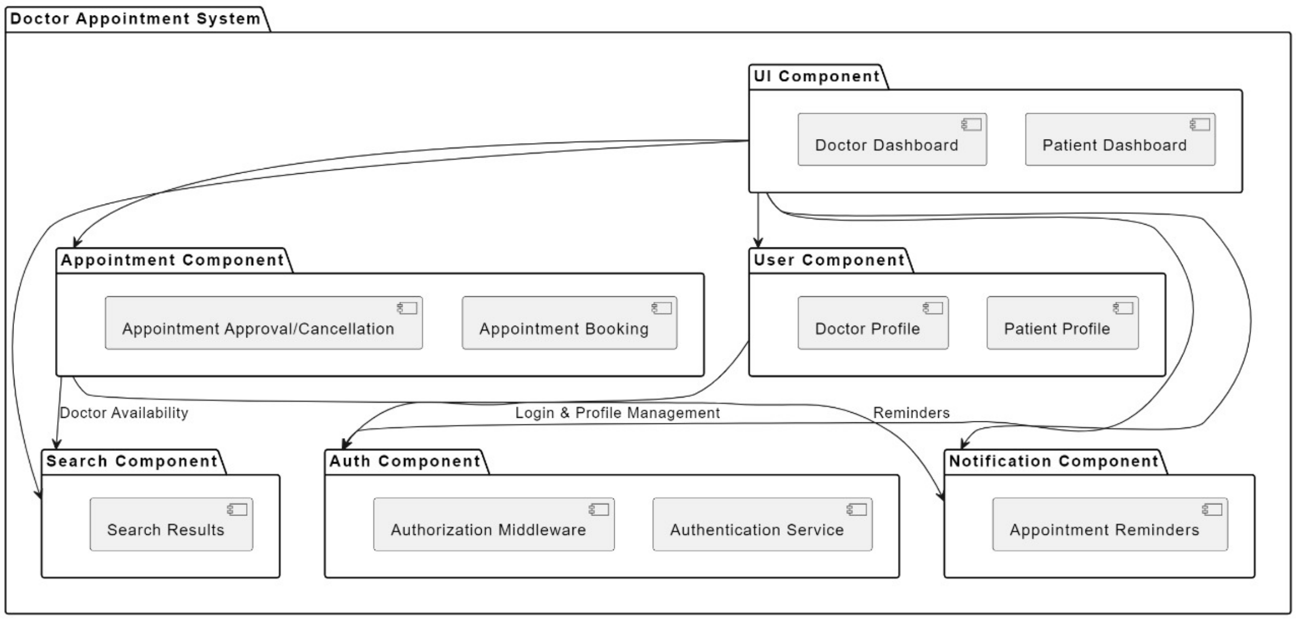
5.7.1 MERN Stack



- **User Devices**:
  - Represents devices (such as laptops, smartphones, tablets) used by end-users (patients, doctors, administrators).
  - Interacts with the client-side application.
- **Client-Side Application (React)**:
  - The front-end part of the MERN stack.
  - Responsible for rendering UI components, handling user interactions, and making requests to the server.
  - Communicates with the Node.js server via RESTful APIs.
- **Web Server (Node.js)**:
  - Serves the client-side application to user devices.
  - Handles incoming HTTP requests from clients.
  - Routes requests to appropriate endpoints (API routes).

- **Node.js Server (Express)**:
  - The back-end part of the MERN stack.
  - Manages business logic, data processing, and database interactions.
  - Implements RESTful APIs for CRUD operations (Create, Read, Update, Delete).
- **Database Server (MongoDB)**:
  - Stores data related to appointments, user profiles, and other system information.
  - MongoDB is a NoSQL database used for its flexibility and scalability.
  - Communicates with the Node.js server to retrieve or update data.
- **Firewall and Security**:
  - Ensures network security by controlling incoming and outgoing traffic.
  - Protects against unauthorized access and potential threats.
  - May include security measures like authentication, authorization, and encryption.
- **Internet**:
  - Represents the global network infrastructure.
  - Facilitates communication between user devices, web servers, and database servers.

5.7.2 Doctor Appointment System

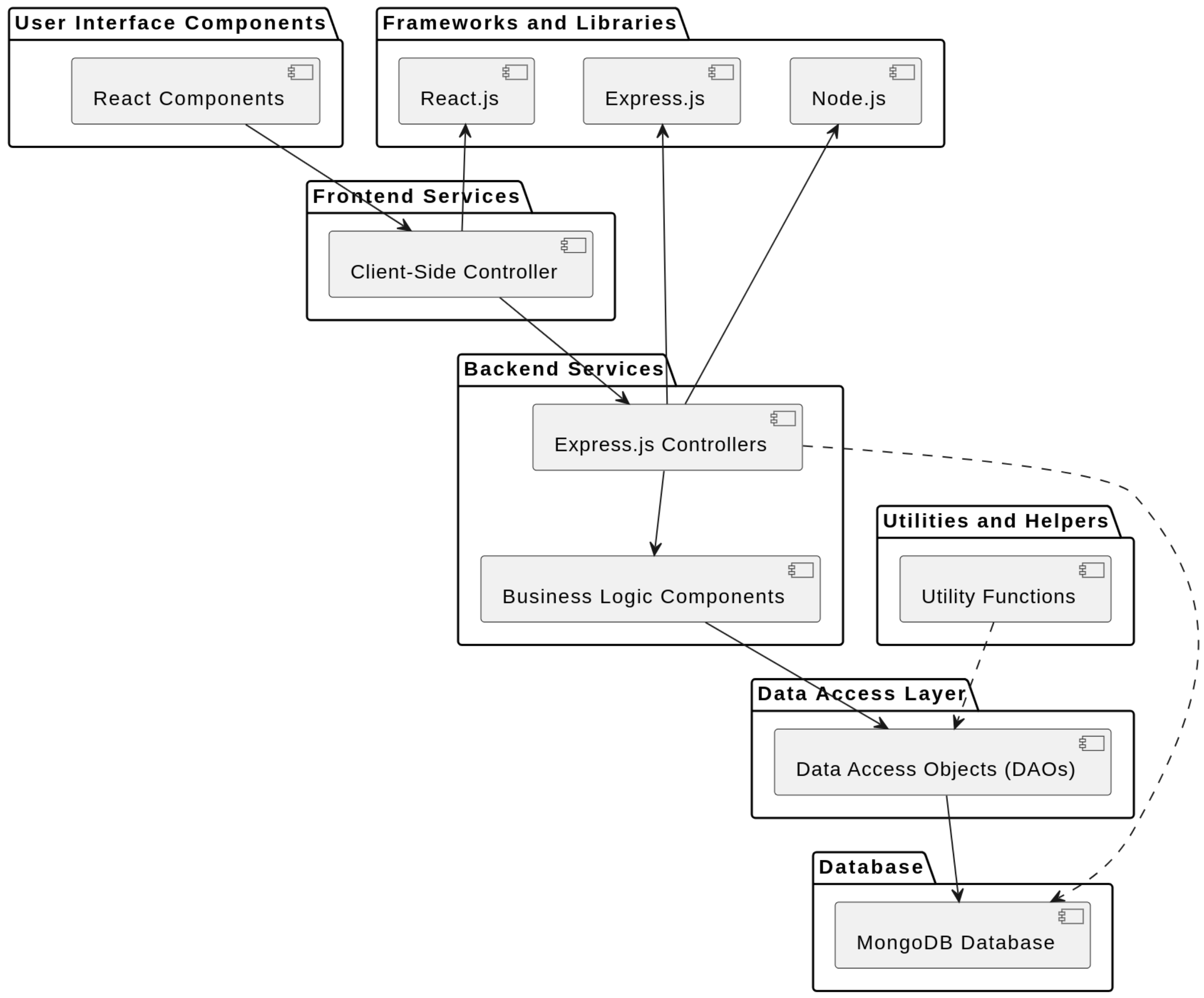


- **UI (User Interface)**
  - Represents the visual part of the system that users interact with.
  - Responsible for displaying appointment-related information to patients and doctors.
  - Includes features like appointment booking forms, appointment lists, and notifications.
- **User**
  - Represents the end-users of the system (patients, doctors, and administrators).
  - Interacts with the UI to perform actions such as booking appointments, viewing schedules, and managing appointments.
- **Appointment Component**
  - Manages all aspects related to appointments:
    * Booking: Allows patients to schedule appointments.
    * Cancellation: Allows patients to cancel appointments.
    * Rescheduling: Handles requests from patients to change appointment times.
    * Communicates with the database to update appointment records.
- **Search Component**
  - Responsible for searching and retrieving relevant information:
    * Patient search: Allows doctors and administrators to find patient records.
    * Doctor availability search: Helps patients find available time slots for specific doctors.
- **Auth (Authentication) Component**
  - Ensures secure access to the system:
    * Validates user credentials during login.
    * Manages user sessions.

    - Role-based access control: Differentiates between patients, doctors, and administrators.

- **Notification Component**
  - Sends notifications to users:
    - Appointment reminders: Alerts patients and doctors before scheduled appointments.
    - Critical alerts: Notifies administrators about system events (e.g., server issues).
- **Integration with Existing Systems**
  - Connects with other hospital systems:
    - Patient records: Retrieves patient information from existing databases.
    - Billing and insurance systems: Shares relevant data for billing purposes.
    - Electronic Health Records (EHR): Ensures consistency across systems.

5.8 Deployment Diagram



- **User Devices**:
  - Represents devices (such as laptops, smartphones, tablets) used by end-users (patients, doctors, administrators).
  - Interacts with the client-side application.
- **Client-Side Application (React)**:
  - The front-end part of the MERN stack.
  - Responsible for rendering UI components, handling user interactions, and making requests to the server.
  - Communicates with the Node.js server via RESTful APIs.
- **Web Server (Node.js)**:
  - Serves the client-side application to user devices.
  - Handles incoming HTTP requests from clients.
  - Routes requests to appropriate endpoints (API routes).
- **Node.js Server (Express)**:
  - The back-end part of the MERN stack.

  - Manages business logic, data processing, and database interactions.
  - Implements RESTful APIs for CRUD operations (Create, Read, Update, Delete).

- **Database Server (MongoDB)**:
  - Stores data related to appointments, user profiles, and other system information.
  - MongoDB is a NoSQL database used for its flexibility and scalability.
  - Communicates with the Node.js server to retrieve or update data.

- **Firewall and Security**:
  - Ensures network security by controlling incoming and outgoing traffic.
  - Protects against unauthorized access and potential threats.
  - May include security measures like authentication, authorization, and encryption.

- **Internet**:
  - Represents the global network infrastructure.
  - Facilitates communication between user devices, web servers, and database servers.

6 Revision History

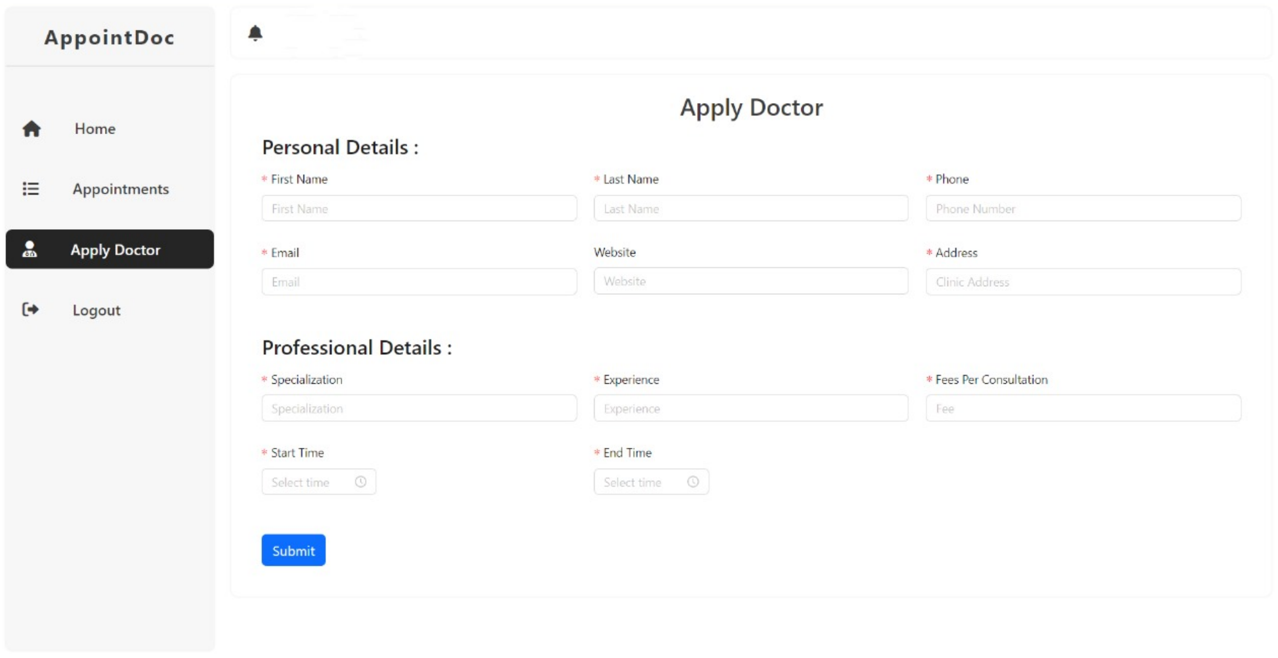
- Project Proposal Submission (February 21, 2024):
  - Submitted the initial project proposal.
  - Outlined the project's objectives, scope, and stakeholders.

- SRS and Prototype Submission (March 31, 2024):
  - Submitted the Software Requirements Specification (SRS) document.
  - Included detailed functional and non-functional requirements.
  - Provided a prototype and UI design for the system.

PROTOTYPE

7 User Interface

7.1 Doctor Application



Apply Doctor

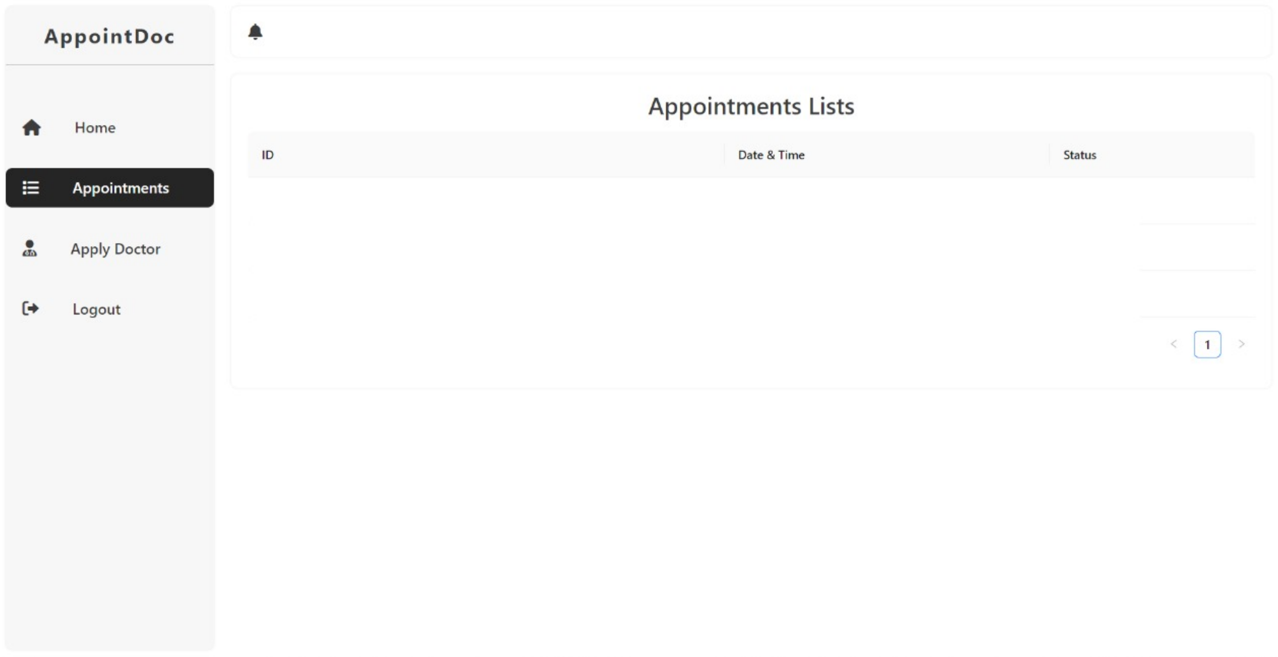
Personal Details :

* First Name	* Last Name	* Phone
First Name	Last Name	Phone Number
* Email	Website	* Address
Email	Website	Clinic Address

Professional Details :

* Specialization	* Experience	* Fees Per Consultation
Specialization	Experience	Fee
* Start Time	* End Time	
Select time	Select time	

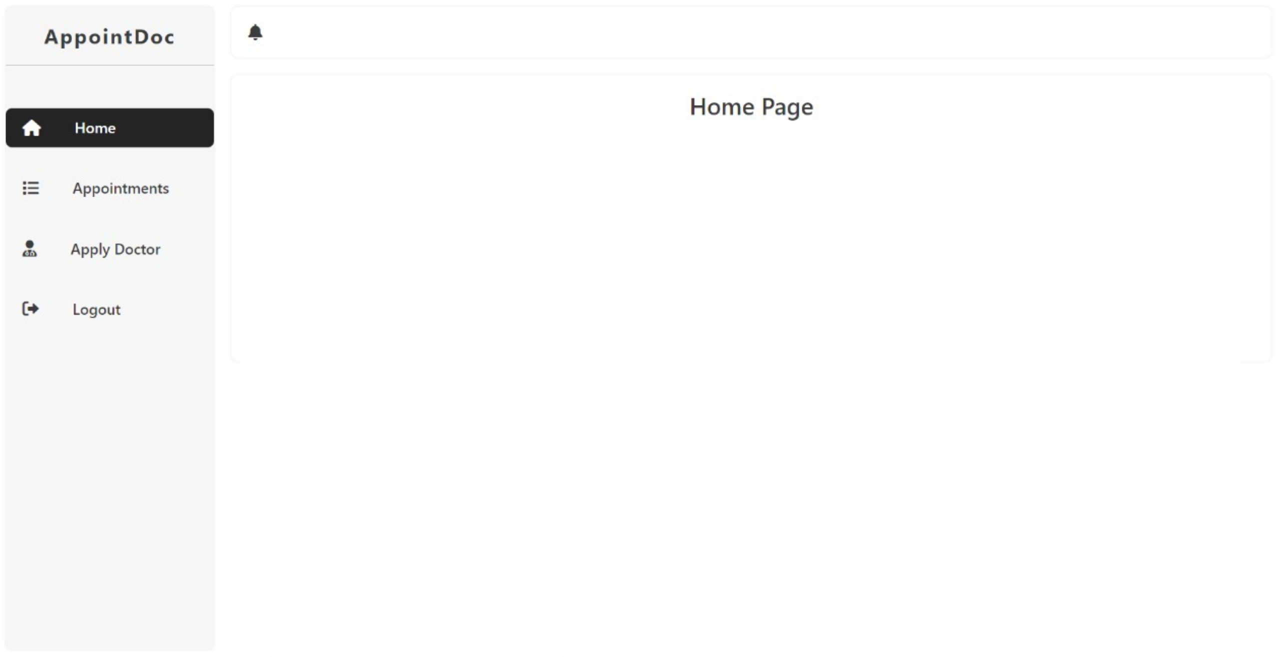
7.2 Appointments Listing



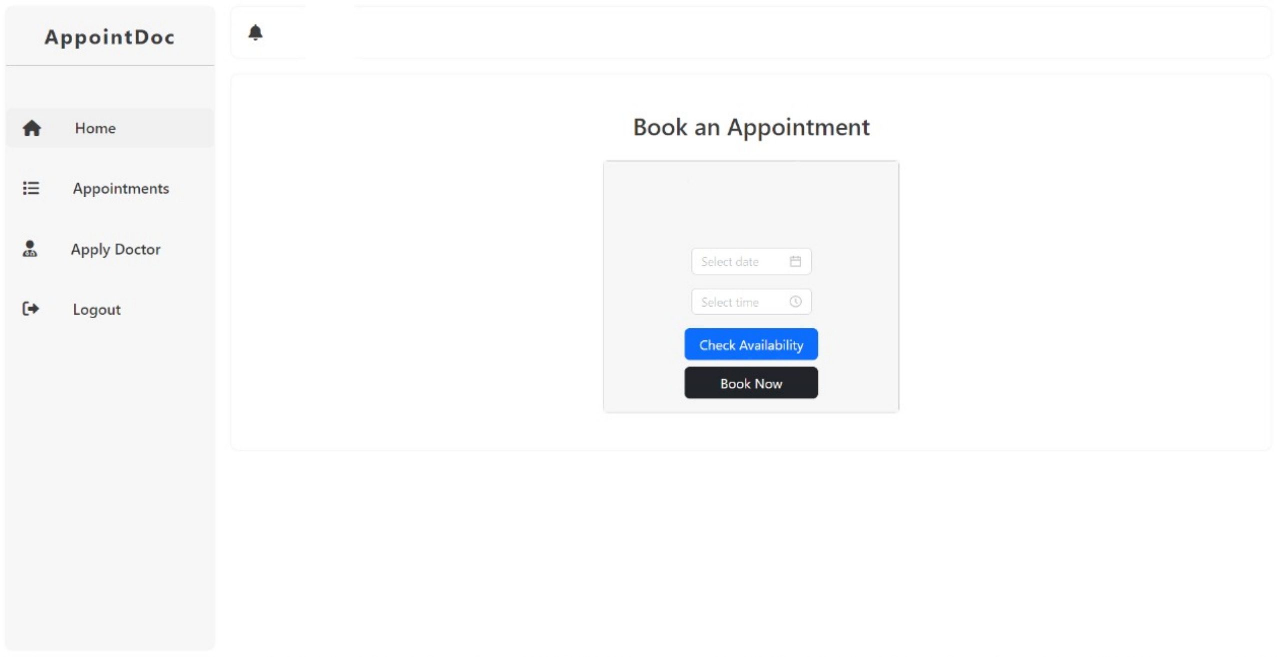
Appointments Lists

ID	Date & Time	Status

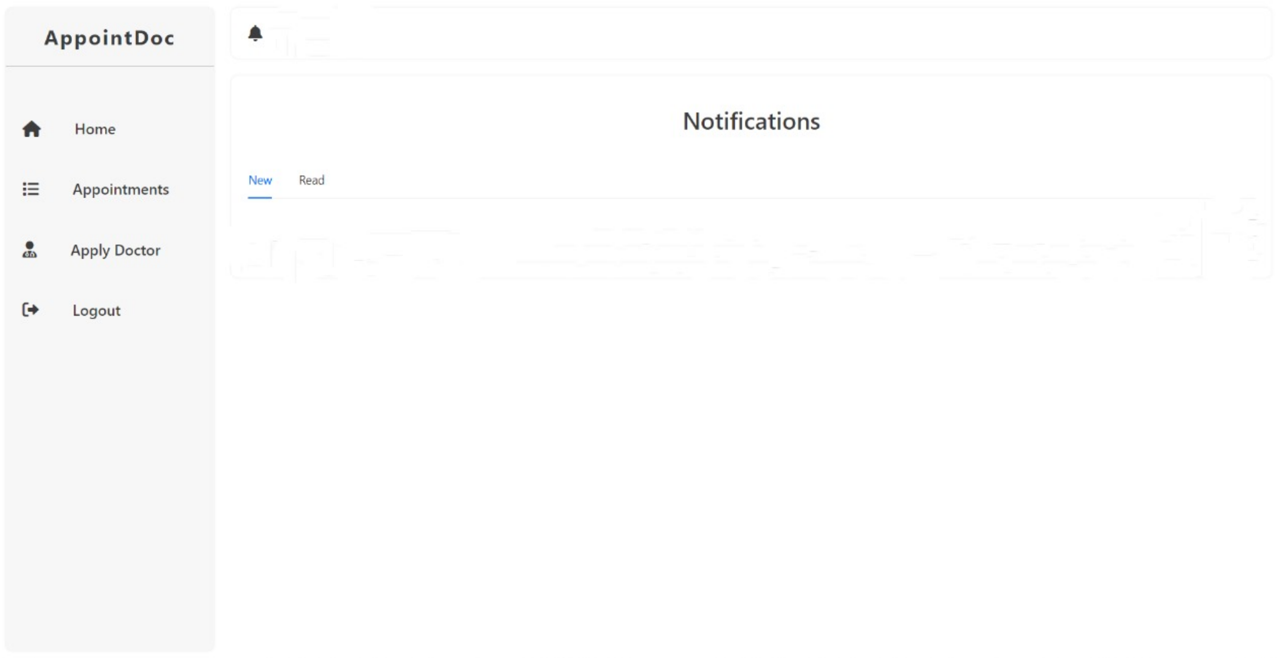
7.3 Home Page



7.4 Appointments Booking



7.5 Notifications Panel


AppointDoc
Home
Appointments
Apply Doctor
Logout
Notifications
New
Read