Healthcare Appointment Scheduling Application (DocLink)

***Submitted* by**

Group – 6

Bhagath Simha Reddy Vummannagari
Ganesh Gampa
Lokesh Mallarapu
Rajendra Prasad Banoth
Venkata Ganesh Kumar Kancham

Submitted on 12/7/2023

Information systems project

(BIS698 -11:00 am to 12:15 pm Monday)

CENTRAL MICHIGAN UNIVERSITY

Table of Contents:

---

1. **Background ……. 3**
2. **Business Problems ……... 3**
3. **Project Description …… 3**
4. **Functional Requirements …… 4**
5. **Out of Scope …… 4**
6. **Project Feasibility …….. 4**
7. **Use Case Diagrams ……... 5 - 8**
8. **Data Flow Diagrams (DFD) ……..... 8-12**
9. **Entity Relationship Diagram (ERD) ……. 12-14**
10. **System Screenshots …… 14 -19**
11. **Conclusion ……. 19**

1. Background:

The present-day healthcare industry has experienced in the recent past a technological revolution but there arises the question of providing patients with that much needed and deserved time as they require. Another major problem that can be attributed to the patients is the difficulty of arranging for an appointment with the doctor for normal sickness or emergencies that are not severe. It is also true that many patients cannot make an appointment with the doctor of their choice in a fairly short period- this again leads to a long waiting time. Therefore, this project proposal aims to develop a solution in the form of a Healthcare Appointment Scheduling Application that enables one to search through a database of doctors in his/her region and, if available, fix a preferential appointment with him/her.

2. Business Problems:

For now, the system where people can book doctor appointments across most regions is still complicated particularly for such minor health issues. The current system requires the patients to search for the doctors themselves, and in the process, they have to call several clinics or get to know about it from someone else. This results in:

- **Delays in care:** Patients delay getting the medical attention they need, leading to prolonged suffering or complications.
- **Inefficiency:** Doctors may have available slots that go unfilled because patients are unaware of their availability.
- **Poor resource utilization:** Clinics and hospitals often struggle to optimize doctor schedules, resulting in idle time for healthcare professionals.

These problems not only affect the patients but also, the service providers who are a part of the operation that has a portion of its work either eliminated or made somewhat less effective in the long run. These are the challenges that require a strategic business need to be done to enhance patients' satisfaction as well as the health care organization's resources.

3. Project Description

The proposed service namely Healthcare Appointment Scheduling Application will be the link between patients and healthcare providers. The system will allow:

- **Doctor registration:** DOCTORS, NURSES, OTHER HEALTHCARE PERSONNEL: These users will be required to open an account on the platform, to showcase their specialty, working schedule, and clinic location among other details.
- **Patient registration and search:** Employers will make profiles for patients and the patient will then look for doctors that befit his or her medical condition, proximity, and time of availability.
- **Real-time booking:** If there exists a doctor to fulfil the needs of a patient in a particular health facility, he or she will be able to schedule an appointment in real time. From the doctor's availability, the slots will be provided on the system.
- **Notifications and reminders:** It will notify the patients and the doctors about appointments, hence discouraging no-shows.

Functional Requirements:

The economic feasibility of the **DocLink Healthcare Appointment Scheduling Application** is highly favorable, as it relies on open-source tools like Python and MySQL, incurring no direct monetary costs. The primary investment is the time and effort of team members, estimated at 60-80 hours per person. Hosting and deployment are handled locally, avoiding additional expenses. While the focus is on learning and practical application, the project also addresses real-world healthcare challenges and provides a strong foundation for potential future development or commercialization. This makes it a valuable and cost-effective academic initiative.

Out of Scope:

- In-app medical consultations (telemedicine).
- Payment processing for medical services.
- Detailed medical record storage.

4. Project Feasibility

Economic Feasibility:

The implementation of the Healthcare Appointment Scheduling Application implies capital investment into the software development as well as the hosting space and publicity. There remains a good chance to make money through charged services for doctors' accounts and the clinics they represent ads or commissions on booked time slots. Similarly, as for the healthcare providers, the gains made in efficiency can be translated to savings thus recouping the development cost.

Technical Feasibility:

It is technically possible since it involves the use of already available technologies including cloud databases, geolocation services, and mobile application development tools. Incorporation of a Mobile Application Development Team: To accomplish this, a development team with specializations in mobile applications, cloud architecture, and user interfaces will be needed. Implementation of the system will also be in phases with the initial features including registration, search, and appointment booking.

Schedule Feasibility:

The key features of the project will be implemented gradually in 3-5 months with the subsequent implementation of additional options. This involves the time that is required to design the system, develop it, test it, and implement it. Moreover, doctors and patients, as potential users, will be asked to provide feedback on the functionality of the design at different stages of development.

Operational Feasibility:

There is high operational feasibility because the application will enhance the availability of healthcare services. For the healthcare-providing organization, this approach will lead to the improvement of scheduling and reduction of no-show rates whereas the patient results in the beneficial factor of enhanced availability of care. Friendly for patients and doctors alike it will

not be difficult to adopt new methods of operation or techniques and training will not be necessary.

Use Case Diagrams:

Register User (Patient/Doctor):

Use Case Name: Register user
Actor: Patient, Doctor
Description: The patient or doctor provides personal information to create an account on the system.
Trigger: The user wants to use the healthcare appointment system.
Normal Course: 1. The user selects Sign up. 2. User fills in personal details, including email, password, and role 3. The system validates the information. 4. The system verifies data and creates an account. 5. An account is created, and a confirmation message is displayed.
Postconditions: The user's account is successfully created and stored.

Log in User:

Use Case Name: Login user
Actor: Patient, Doctor
Description: The patient or doctor logs into the system to access their respective dashboard.
Trigger: The user wants to manage appointments or availability.
Normal Course: 1. User selects "Login." 2. User enters credentials (email, password). 3. The system verifies credentials. 4. The user is redirected to the dashboard
Postconditions: The user is logged in.

Reset Password:

Use Case Name: Reset Password
Actor: System, Patient, Doctor
Description: The user resets their account password.
Trigger: The user forgets their password.
Normal Course: 1. User selects "Forgot Password."

2. User enters their registered email. 3. The system verifies the email and allows the user to set a new password. 4. Password is updated in the system
Postconditions: The user's password is reset.

View Doctor Dashboard:

Use Case Name: View Doctor Dashboard
Actor: Doctor
Description: The doctor views their profile, availability, and appointment history.
Trigger: The doctor logs in and navigates to their dashboard.
Normal Course: 1. Doctor logs into the system. 2. The system displays the doctor's profile, manage availability options, and appointment history. 3. Doctor can add, cancel availability, or update profile.
Postconditions: The doctor’s information and actions are accessible and manageable.

View Patient Dashboard:

Use Case Name: View Patient Dashboard
Actor: Patient
Description: The patient views their profile, upcoming appointments, and history.
Trigger: The patient logs in and navigates to their dashboard.
Normal Course: 1. Patient logs into the system. 2. The system displays user information, upcoming appointments, and appointment history. 3. Patient can book or cancel appointments and update their profile.
Postconditions: The patient's information is displayed, and appointment actions are accessible.

Update User Profile:

Use Case Name: Update User Profile
Actor: Patient, Doctor
Description: The user updates their personal information.
Trigger: The user wants to edit their profile details.
Normal Course: 4. User selects "Edit Profile." 5. User updates the desired fields. 6. The system validates and saves the changes.
Postconditions: Profile is successfully updated.

Book an Appointment:

Use Case Name: Book an Appointment
Actor: Patient
Description: The patient books an appointment with a doctor based on availability
Trigger: The patient selects a doctor and a suitable time slot.
Normal Course: 1. The patient selects a doctor from the search results. 2. The patient selects an available time slot. 3. The system confirms the booking.
Postconditions: Appointment is created and saved in the system.

Cancel an Appointment:

Use Case Name: cancel an appointment
Actor: Patient, Doctor
Description: The patient or doctor cancels an existing appointment.
Trigger: The user wants to change or cancel an appointment.
Normal Course: 1. User selects cancel an appointment from the dashboard. 2. User selects an appointment from the dashboard 3. User confirms cancellation. 4. The system updates the appointment status to canceled.
Postconditions: Appointment is updated or canceled.

Manage Doctor Availability:

Use Case Name: Manage Doctor Availability
Actor: Doctor
Description: The doctor adds or cancels availability slots.
Trigger: The doctor needs to update their schedule.
Normal Course: 1. Doctor selects "Add Availability" or "Cancel Availability." 2. Doctor specifies the date and time. 3. The system updates the availability records..
Postconditions: Availability is successfully managed.

Generate Completed Appointments Report:

Use Case Name: Generate Completed Appointments Report
Actor: Doctor
Description: The doctor generates a PDF report of their completed appointments.
Trigger: The user wants to see past and upcoming appointments.
Normal Course: 1. The doctor logs into their dashboard. 2. The doctor clicks the **"Get Reports"** button.

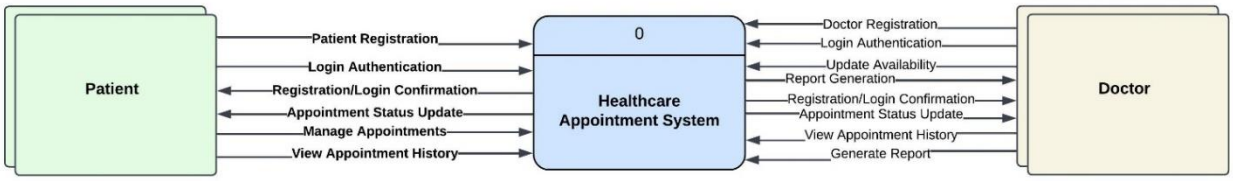
3. The system fetches all completed appointments from the database. 4. The system generates a PDF report that includes: • Patient Name • Gender • Date of Birth 5. Appointment Date and Time 6. Total number of completed appointments 7. The report is saved in the designated directory.
Postconditions: The doctor wants to document or review their completed appointments.

View Appointment History:

Use Case Name: View Appointment History
Actor: Patient, Doctor
Description: The user views a history of their appointments.
Trigger: The user wants to see past and upcoming appointments.
Normal Course: 8. User can see "Appointment History" on their dashboard. 9. The system retrieves and displays appointment details
Postconditions: Appointment history is displayed.

Data Flow Diagram (DFD):

Context Level diagram:



The context-level Data Flow Diagram (DFD) provides an overview of the **Healthcare Appointment System**, illustrating the interaction between the primary external entities **Patients**, **Doctors,** and the central system. The system acts as the intermediary, processing and managing the flow of information between these entities.

1. Patient Interactions

- **Input to the System**:
    - **Patient Registration**: Patients provide their details to register with the system.
    - **Login Authentication**: Patients submit credentials for authentication.
    - **Manage Appointments**: Patients request actions such as booking, modifying, or canceling appointments.
- **Output from the System**:
    - **Registration/Login Confirmation**: The system confirms successful registration or login.
    - **Appointment Status Update**: Notifications regarding the status of appointments (e.g., confirmed, canceled).
    - **View Appointment History**: Patients can retrieve records of their past appointments.

2. Doctor Interactions

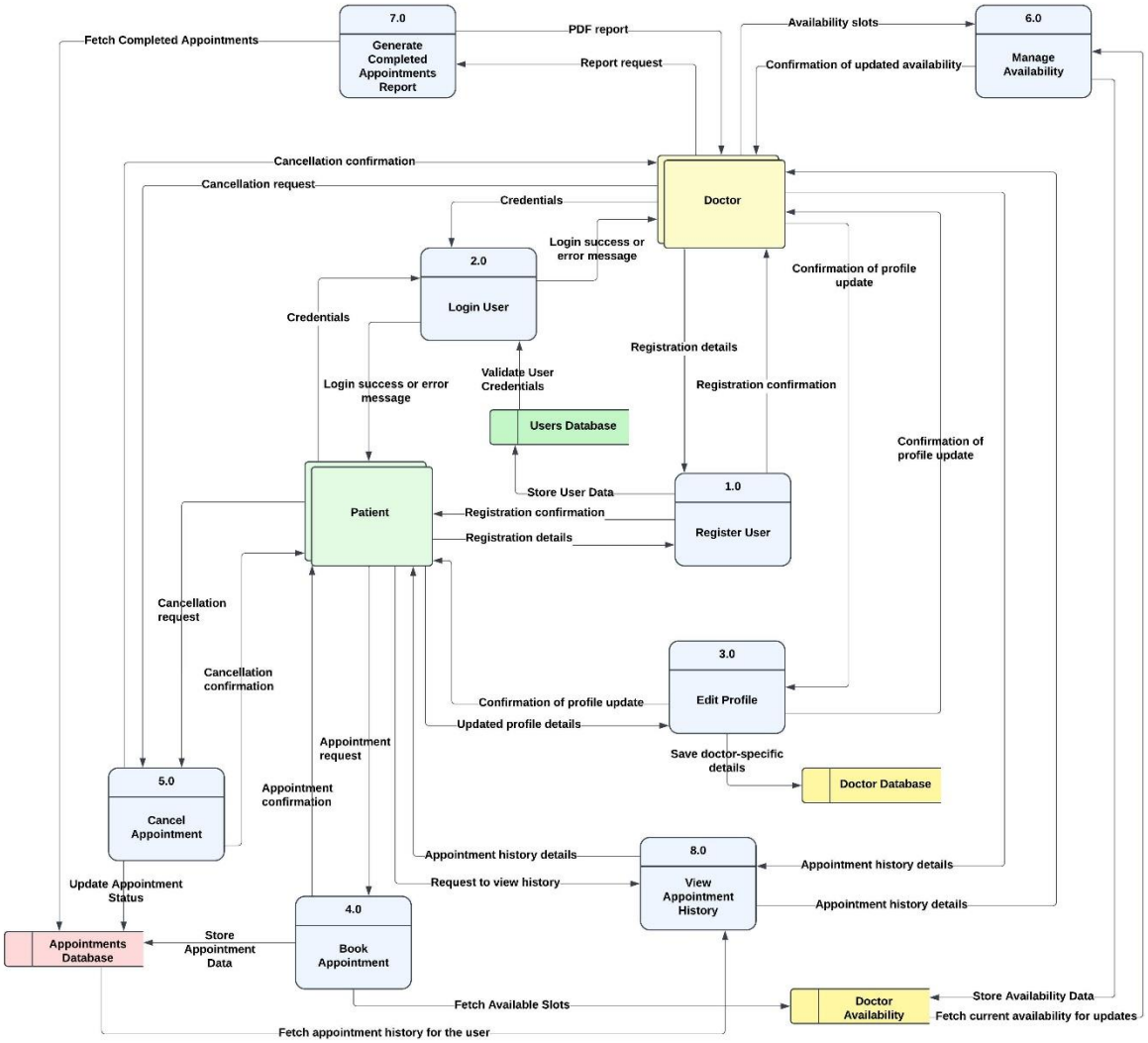
- **Input to the System**:
    - **Doctor Registration**: Doctors submit their details for registration.
    - **Login Authentication**: Doctors provide login credentials for access.
    - **Update Availability**: Doctors update their availability to accept appointments.
- **Output from the System**:
    - **Registration/Login Confirmation**: The system confirms successful registration or login.
    - **Appointment Status Update**: Notifications about appointment requests or changes.
    - **View Appointment History**: Doctors can access details of their completed appointments.

3. Healthcare Appointment System

The system is the central hub that:

- Facilitates **secure data exchange** between patients and doctors.
- Handles **registration, authentication, and appointment management** processes.
- Maintains and provides access to **appointment history** for both patients and doctors.

Level 0 Diagram:



The **Level-0 DFD** provides a high-level view of the **Healthcare Appointment System**, illustrating the major processes, data flows, and interactions between the **Patient**, **Doctor**, and the system. This diagram represents the overall functionality of the system and highlights the data exchanges required to perform key tasks.

External Entities:

- **Patient:** Registers, logs in, books appointments, edits profile, and views appointment history.
- **Doctor:** Registers, logs in, manages availability, views appointment history, cancels appointments, and generates reports.

Data Stores:

- **Users Database:** Stores user credentials and profile information.
- **Appointments Database:** Contains details of all booked, canceled, and completed appointments.
- **Doctor Database:** Maintains doctor-specific information such as specialization and availability.
- **Doctor Availability:** Tracks and updates the availability slots provided by doctors.

Key Processes and Data Flows:

1. **Process 1.0: Register User**
    - **Input**: Registration details from Patient or Doctor.
    - **Process**: Validates the input and stores the user data in the **Users Database**.
    - **Output**: Sends a registration confirmation to the respective user.
2. **Process 2.0: Login User**
    - **Input**: User credentials from Patient or Doctor.
    - **Process**: Validates credentials against the **user database**.
    - **Output**: Sends a success or error message based on login status.
3. **Process 3.0: Edit Profile**
    - **Input**: Updated profile details from Patient or Doctor.
    - **Process**: Saves changes to the respective database
    - **Output**: Sends confirmation of the profile update.
4. **Process 4.0: Book Appointment**
    - **Input**: Appointment request details from Patient.
    - **Process**: Stores appointment data in the **Appointments Database** and fetches available slots from **Doctor Availability**.
    - **Output**: Sends an appointment confirmation to the Patient.
5. **Process 5.0: Cancel Appointment**
    - **Input**: Cancellation request from Patient or Doctor.
    - **Process**: Updates the **Appointments Database** to reflect the cancellation.

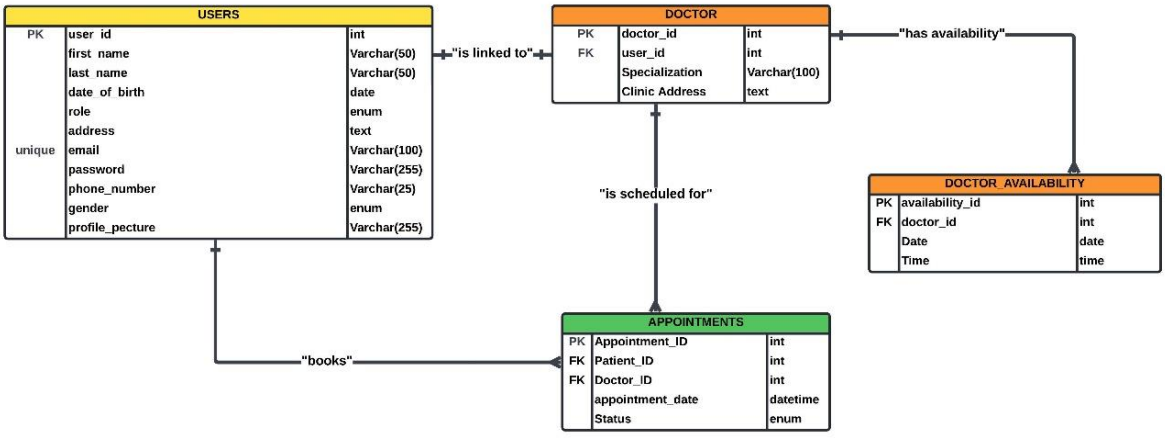
- **Output**: Sends cancellation confirmation to the user.

6. **Process 6.0: Manage Availability**
    - **Input**: Availability slots from doctor.
    - **Process**: Updates availability data in the **Doctor Availability** database.
    - **Output**: Sends confirmation of updated availability to the Doctor.
7. **Process 7.0: Generate Completed Appointments Report**
    - **Input**: Report request from doctor.
    - **Process**: Fetches completed appointment data from the **Appointments Database**.
    - **Output**: Generates a PDF report and sends it to the Doctor.
8. **Process 8.0: View Appointment History**
    - **Input**: History request from Patient or Doctor.
    - **Process**: Fetches appointment history from the **Appointments Database**.
    - **Output**: Displays the appointment history to the user.

    .

Entity Relationship Diagram (ERD):

The Entity-Relationship Diagram (ERD) provides a high-level blueprint of the database design for the Healthcare Appointment System. It represents the core entities, their attributes, and relationships, enabling efficient data management and seamless integration of the system's functionalities.



Key Entities and Their Roles:

1. Users: Represents all users of the system, including both patients and doctors.

Attributes:

- user_id (Primary Key): Unique identifier for each user.
- first_name, last_name, date_of_birth: Basic personal information.
- role: Defines whether the user is a Patient or Doctor.
- email, password: Unique credentials for login and authentication.
- address, phone_number, gender, profile_picture: Additional profile details.

Purpose: Central table for storing user information.

2. Doctor: Contains information specific to doctors.

Attributes:

- doctor_id (Primary Key): Unique identifier for each doctor.
- user_id (Foreign Key referencing Users): Links doctor-specific data to the Users table.
- specialization: The area of expertise of the doctor.
- clinic_address: The clinic's location.

Purpose: Provides additional details about doctors, enabling features like specialization search.

3. Appointments: Tracks all appointments between patients and doctors.

Attributes:

- appointment_id (Primary Key): Unique identifier for each appointment.
- patient_id (Foreign Key referencing Users): Identifies the patient.
- doctor_id (Foreign Key referencing Doctor): Identifies the doctor.
- appointment_date: Scheduled date and time for the appointment.
- status: Enum field indicating the appointment's status (e.g., booked, canceled, completed).

Purpose: Central table for managing all appointment-related data.

4. Doctor Availability: Stores the availability schedule for doctors.

Attributes:

- availability_id (Primary Key): Unique identifier for each availability record.
- doctor_id (Foreign Key referencing Doctor): Links availability data to a specific doctor.
- date, time: Specifies when the doctor is available.

Purpose: Ensures appointment booking aligns with the doctor's availability.

Relationships Between Entities:

Users and Doctor:

- **Type: One-to-One**
- **Description:** Each doctor is also a user in the system, but not all users are doctors. This relationship ensures that doctor-specific data is linked to the general user information.

Users and Appointments:

- **Type: One-to-Many**
- **Description:** A single patient (user) can book multiple appointments in the system. Each appointment, however, is linked to only one patient.

Doctor and Appointments:

- **Type: One-to-Many**
- **Description:** A doctor can have multiple appointments scheduled with different patients. Each appointment references a specific doctor.

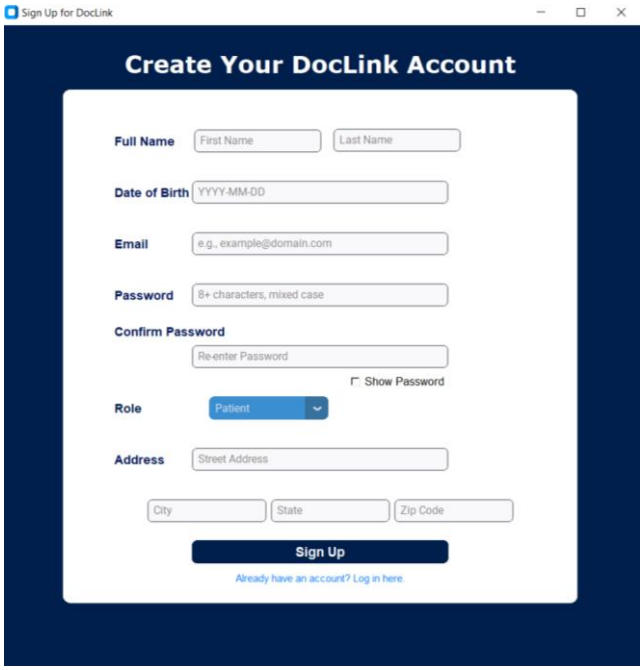
Doctor and Doctor Availability:

- **Type: One-to-Many**
- **Description:** Each doctor can have multiple availability slots, allowing patients to book appointments based on the doctor's available times.

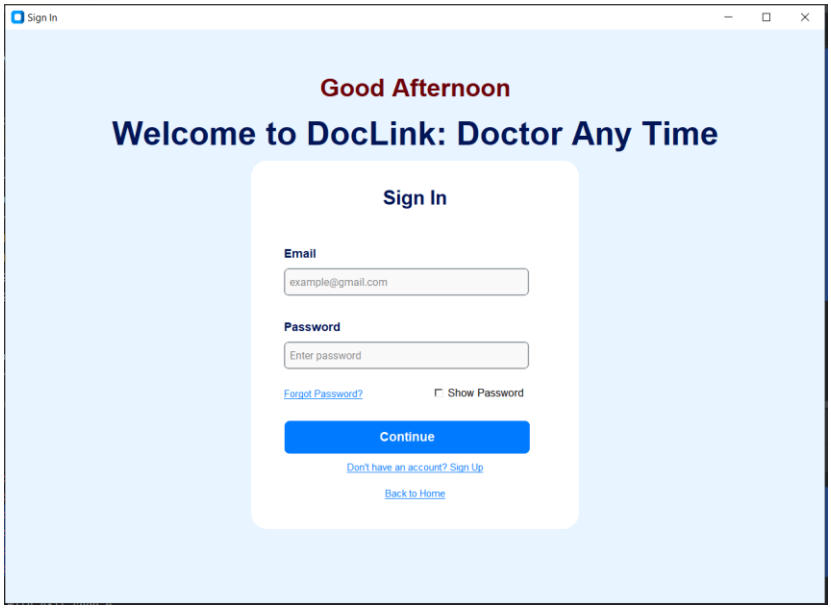
System screenshots:

Home page:

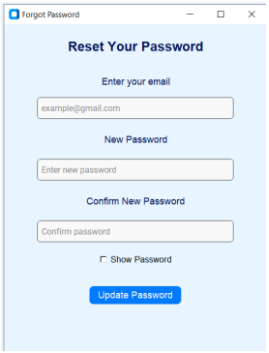
Sign-up page:



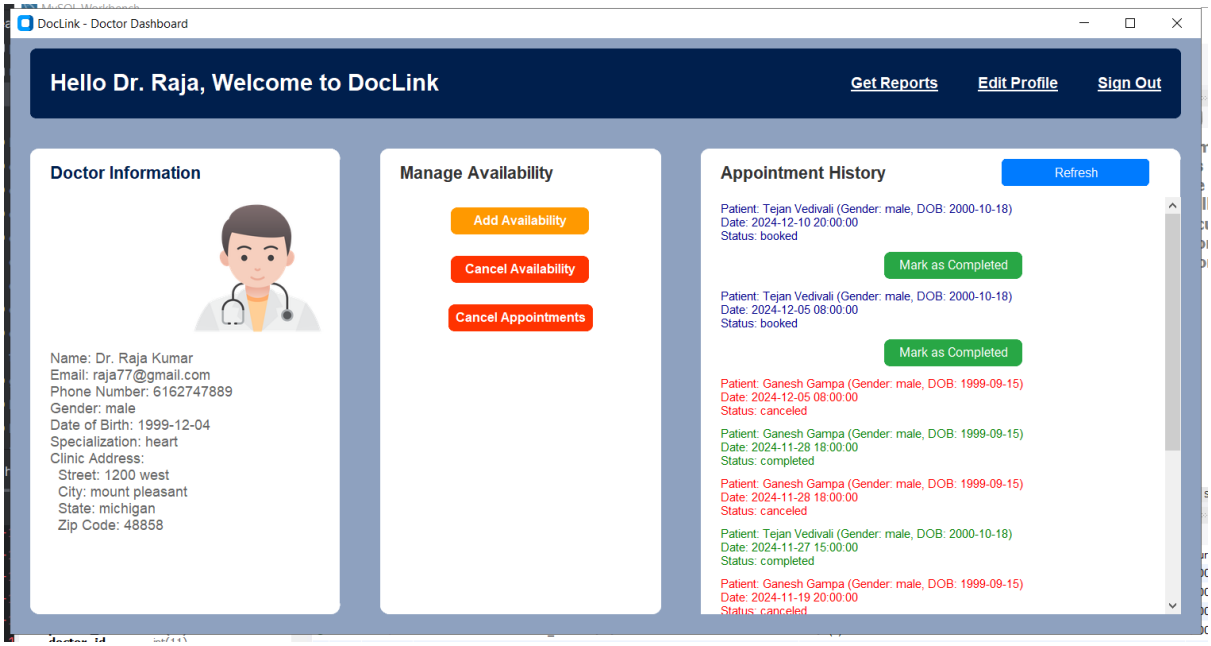
Login page:



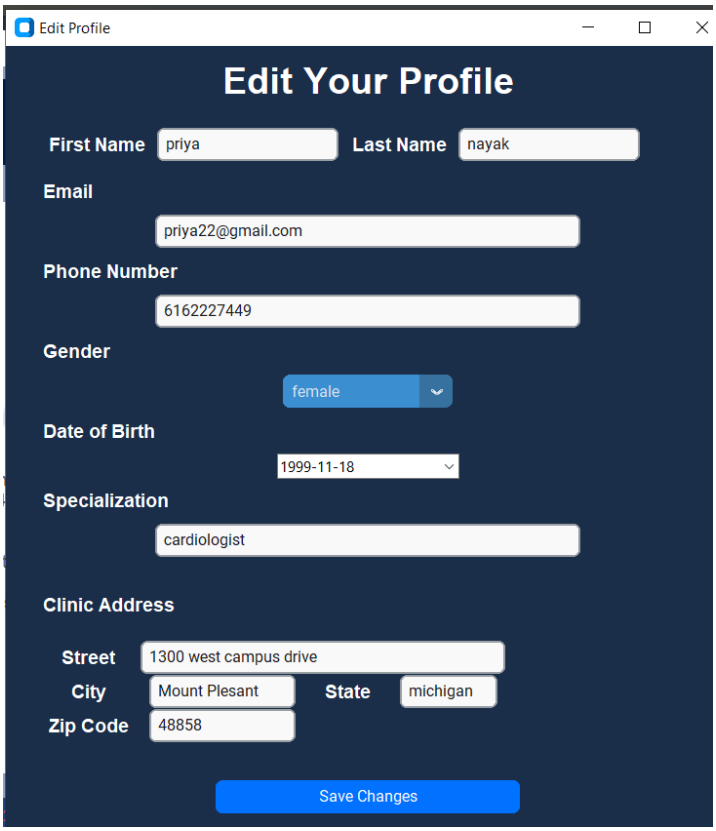
Forgot Page:



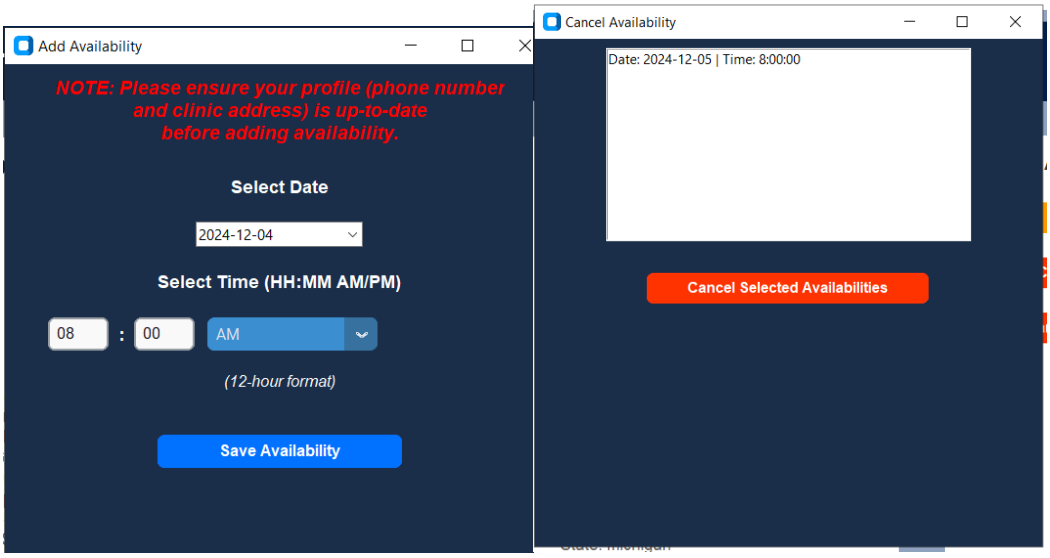
Doctor Dashboard:



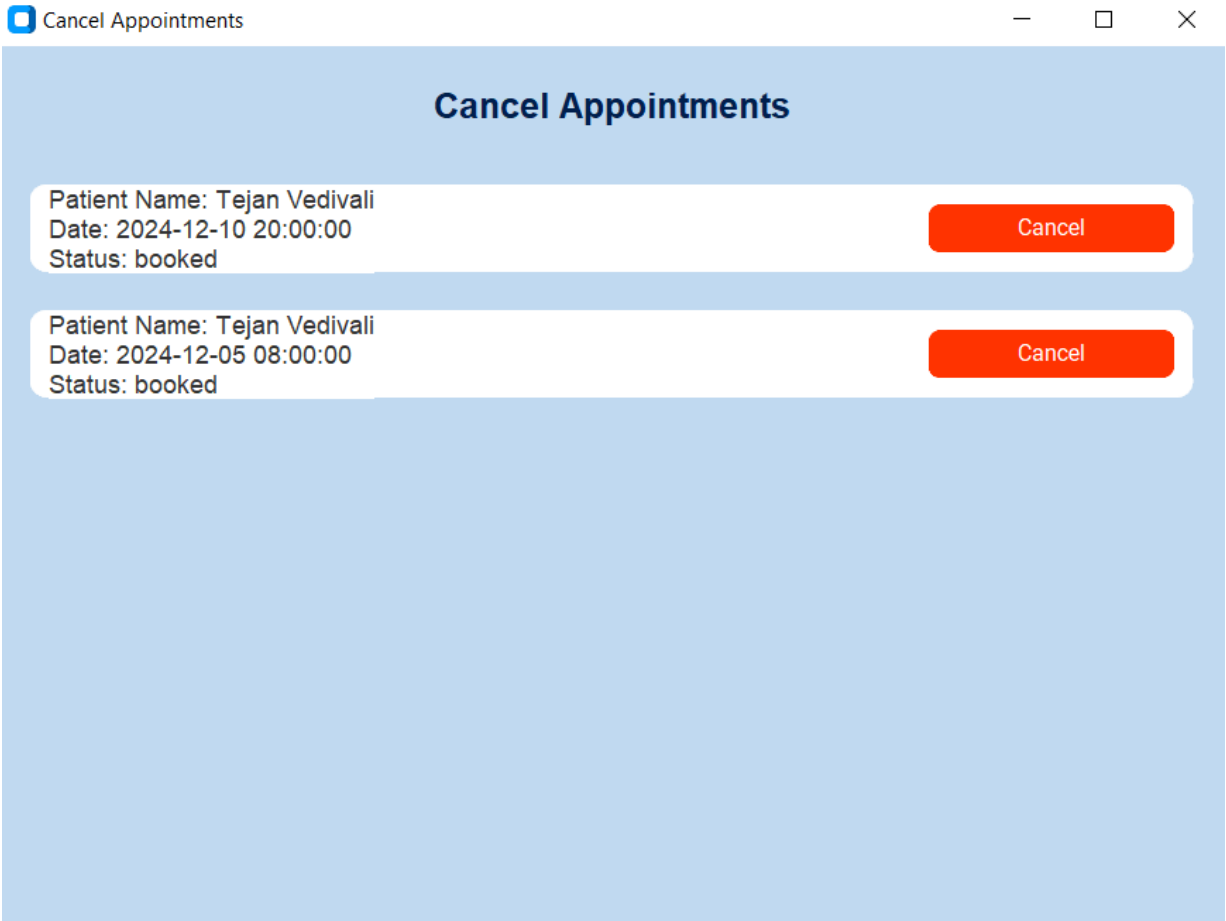
Edit profile:



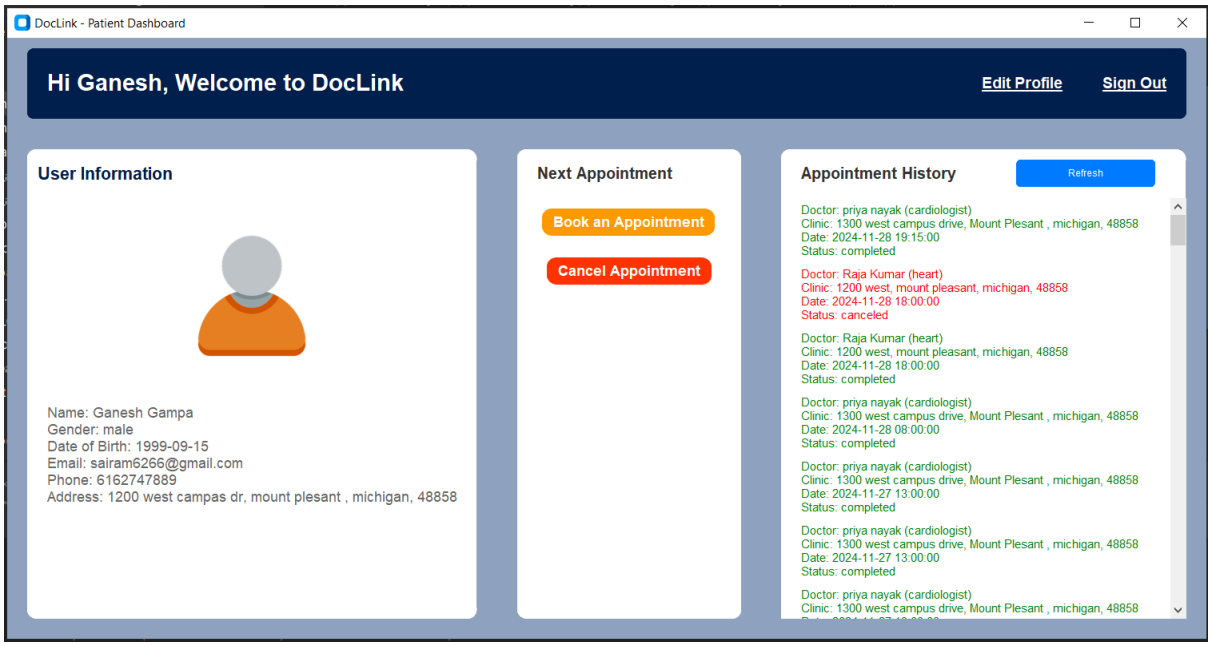
Add Availability and cancel Availability:



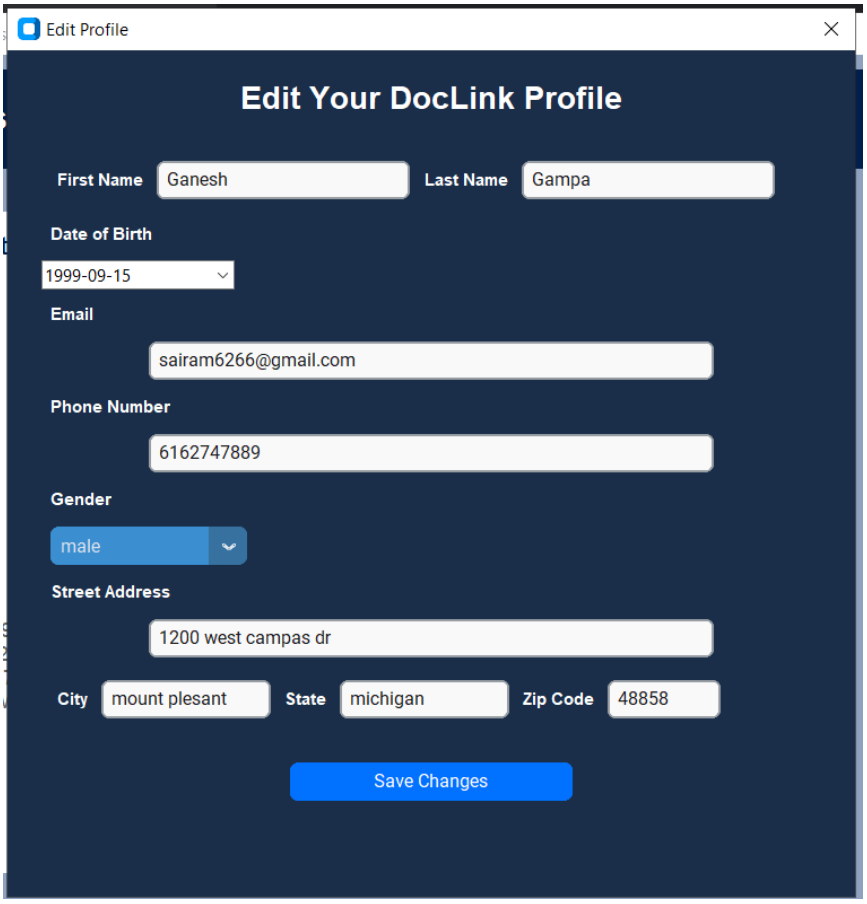
Cancel Appointments:



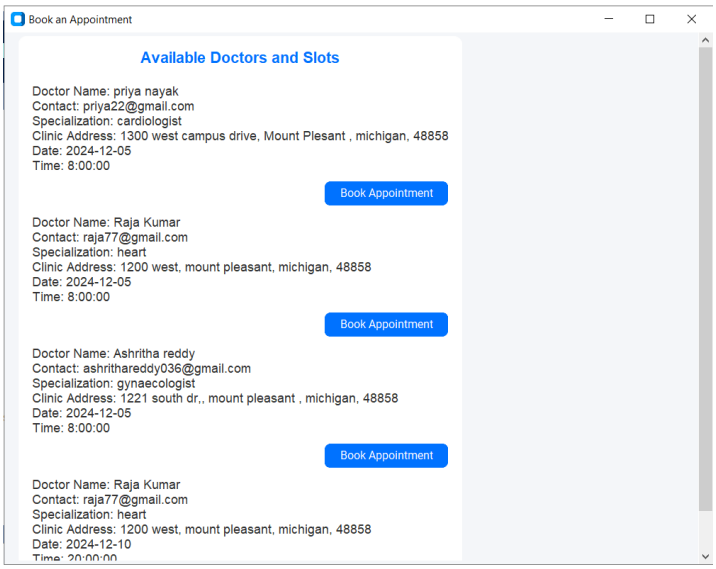
Patients Dashboard:



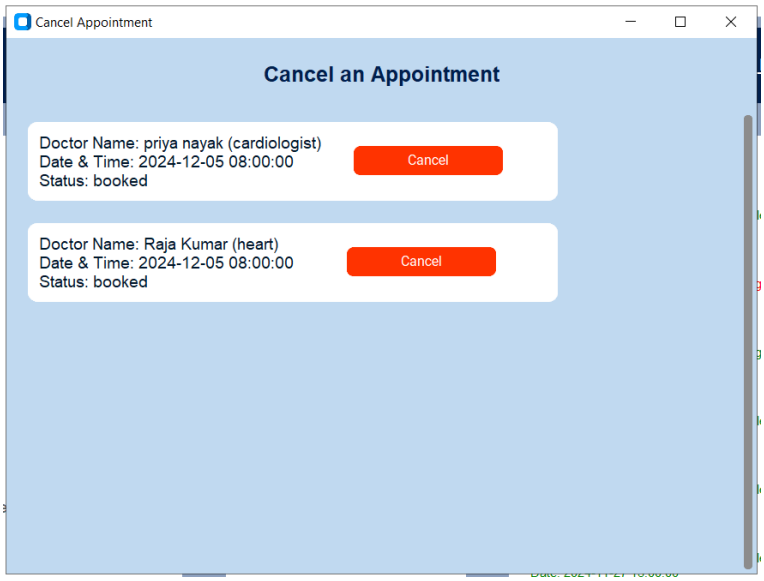
Edit Profile:



Book an Appointment:



Cancel An Appointment:



Conclusion

The DocLink Healthcare Appointment Scheduling Application addresses inefficiencies in healthcare appointment management by providing a user-friendly platform for patients and doctors. It simplifies appointment booking with real-time availability, personalized dashboards, and robust validations. The system enhances resource utilization, reduces no-show rates, and improves patient access to care. With its scalable design and efficient features, DocLink is a practical, innovative solution that streamlines healthcare scheduling and supports future growth opportunities.