

Clinical Management System

Web Application

By:

*Names: Ziad Elshazly Mohamed Magdy Merna Wesa Katreen William*

*IDs: 221001682*  *221001891 221002000 221001947*

Under Supervision Of:

*Names: Dr. Mohamed Hassan Mahmoud Elgazzar Eng. Ahmed Jamal Metwaly*

# Project Overview:

This project involves creating a web-based **Clinical Management System (CMS)** designed to streamline clinic operations, manage patient information, and provide patients with the convenience of online appointment reservations. The system will incorporate user-specific interfaces and role-based access to ensure secure and efficient functionality for various user types, including patients, receptionists, doctors, and administrators.

# Problem Statement:

**Description of the Problem**: Clinics often struggle with managing appointments efficiently, keeping patient records organized, and maintaining secure access to sensitive information. Manual systems or generic solutions lead to mismanagement of appointments, delays in accessing patient information, and potential data security issues.

**Proposed Solution**: The Clinical Management System will provide an online reservation feature for patients, allowing them to book appointments and view their own medical records securely. The system will also assign specific roles to clinic staff, ensuring that only authorized users can access or modify particular information. This tailored solution will reduce administrative burden, improve data security, and enhance patient satisfaction.

# Proposed Solution:

The Clinical Management System will provide an online reservation feature for patients, allowing them to book appointments and view their own medical records securely. The system will also assign specific roles to clinic staff, ensuring that only authorized users can access or modify particular information. This tailored solution will reduce administrative burden, improve data security, and enhance patient satisfaction.

# User Personas:

#### Patient:

* **Needs**: Ability to book and view their own appointments, access to view their personal records.
* **Goals**: Reduce waiting times and improve convenience by managing appointments online.
* **Permissions**: Can only interact with their own data, including viewing their own records and scheduling appointments.

#### Receptionist:

* **Needs**: Manage and schedule patient appointments, check-in patients upon arrival.
* **Goals**: Ensure smooth clinic operation by keeping an organized schedule.
* **Permissions**: Access to scheduled appointments for all patients but restricted from viewing medical records.

#### Doctor:

* **Needs**: Access patient records, update notes, and view scheduled appointments.
* **Goals**: Streamline patient information access to prepare for appointments efficiently.
* **Permissions**: Can view and update medical records for assigned patients but does not have full administrative privileges.

#### Admin:

* **Needs**: Manage system-wide settings, including user role assignments and security protocols.
* **Goals**: Maintain system security and oversee smooth operations by managing user access.
* **Permissions**: Full access to all features and modules, including role management and system settings.

# Requirements:

#### Functional Requirements:

* **Role-Based Authentication**: All users must log in with specific credentials tied to their role.
* **Online Appointment Booking** (for patients): Patients can book, view, and cancel their appointments via the system.
* **Appointment Management** (for Receptionist/Admin): Receptionists and admins can manage all patient appointments, including scheduling and cancellations.
* **Patient Record Management** (for doctors): Doctors can view, and update patient records related to their appointments.
* **Role Management** (for Admin): Admin users can assign roles and manage permissions within the system.

#### Non-Functional Requirements:

* **Security**: Ensure secure data handling with encryption and role-based access control (e.g., receptionist)
* **Scalability**: Design the system to accommodate future growth in terms of features and users.

In addition to Support up to 1000 concurrent users without performance degradation, ensuring response times stay below 5 second for 95% of requests.

* **Reliability**: Minimize downtime and provide a stable user experience (Maintain an uptime of 99.9% monthly, allowing for a maximum of 43 minutes of downtime).
* **Usability**: Ensure an intuitive interface that can be navigated easily by users of varying technical backgrounds.

# System Architecture & Design:

#### 

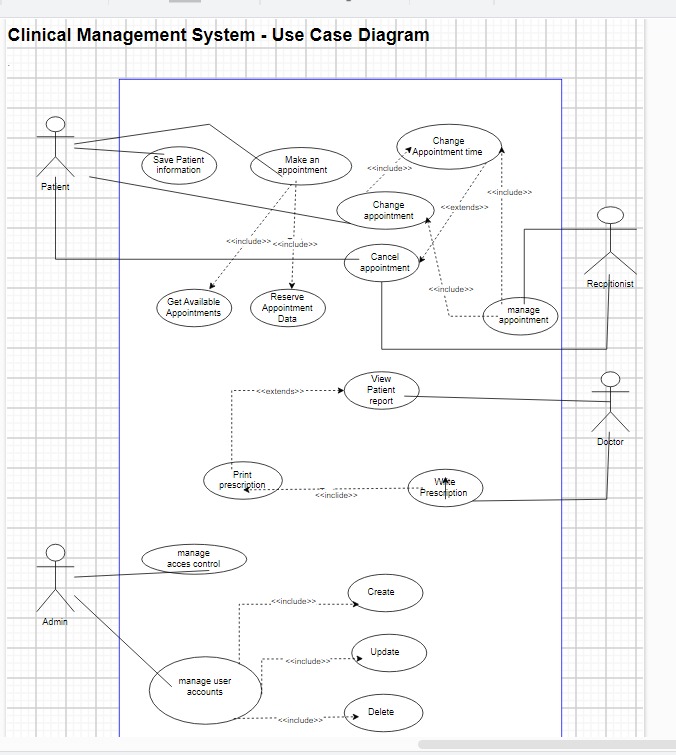
#### Architecture Overview:

The Clinical Management System will use a three-layer architecture:

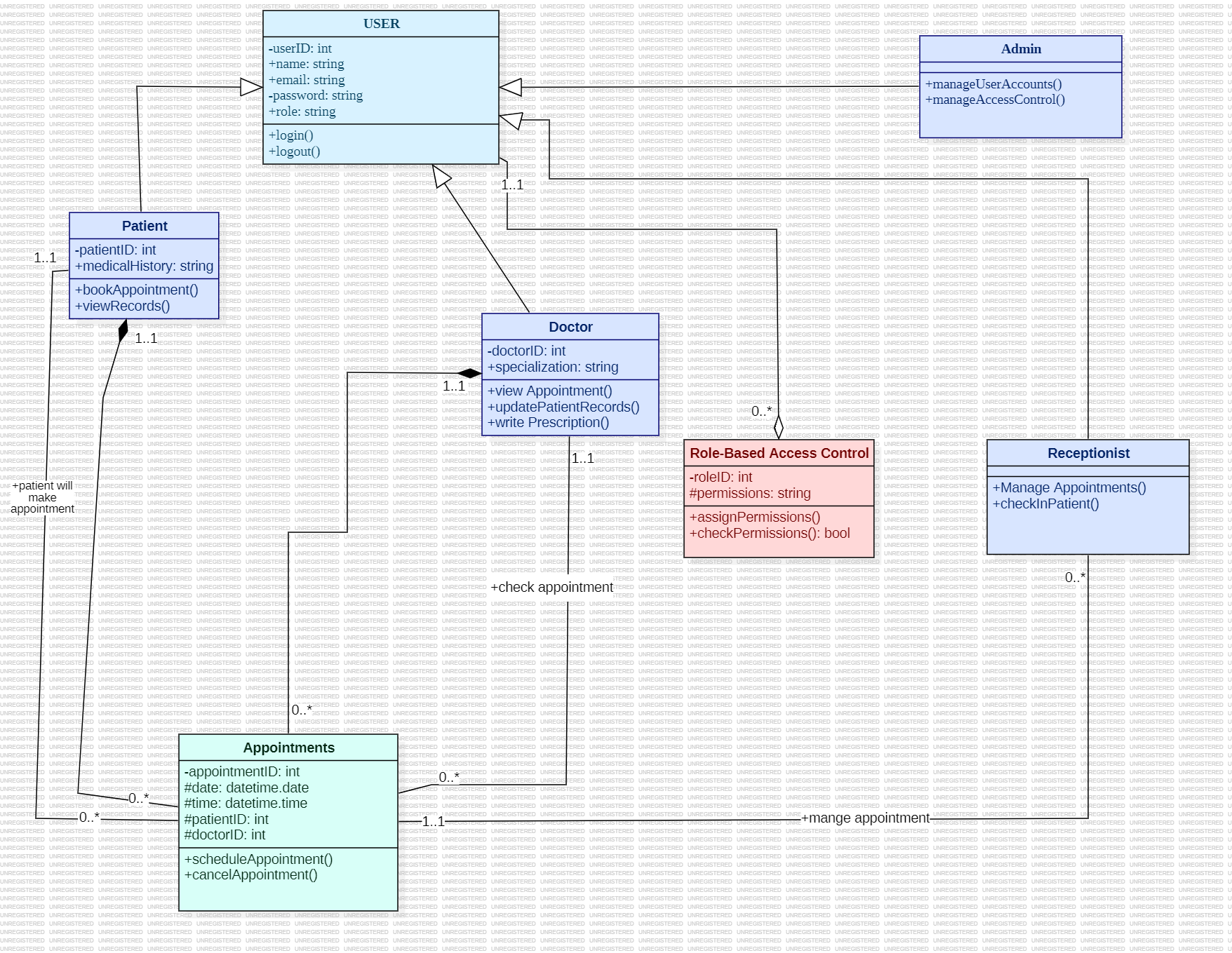
1. **Presentation Layer**: Web-based interface for interacting with the system. This layer will differ depending on the user role, with specific views for patients, doctors, and admin users.
2. **Application Layer**: Manages business logic, handling requests like booking an appointment, viewing records, and role-based permissions.
3. **Database Layer**: Stores user credentials, appointment details, patient records, and other essential data.

#### **Architectural Diagrams:**

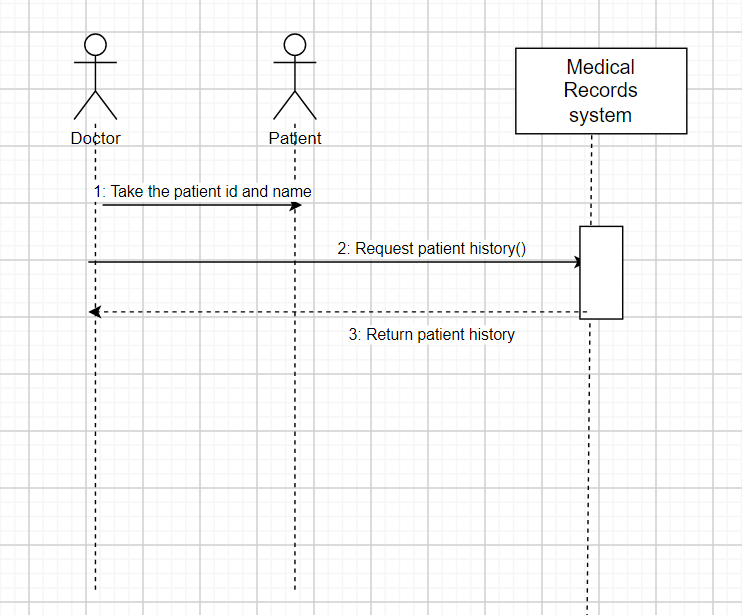
Use-Case Diagram:

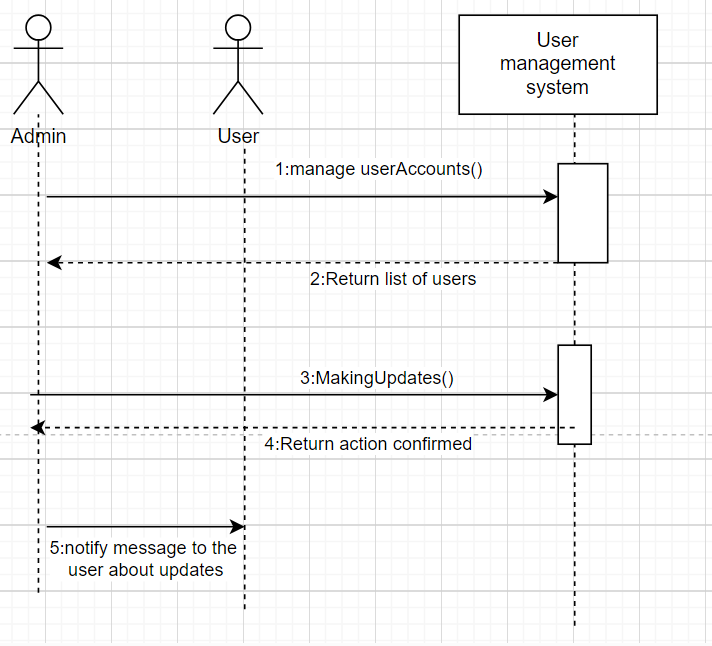


UML Diagram:

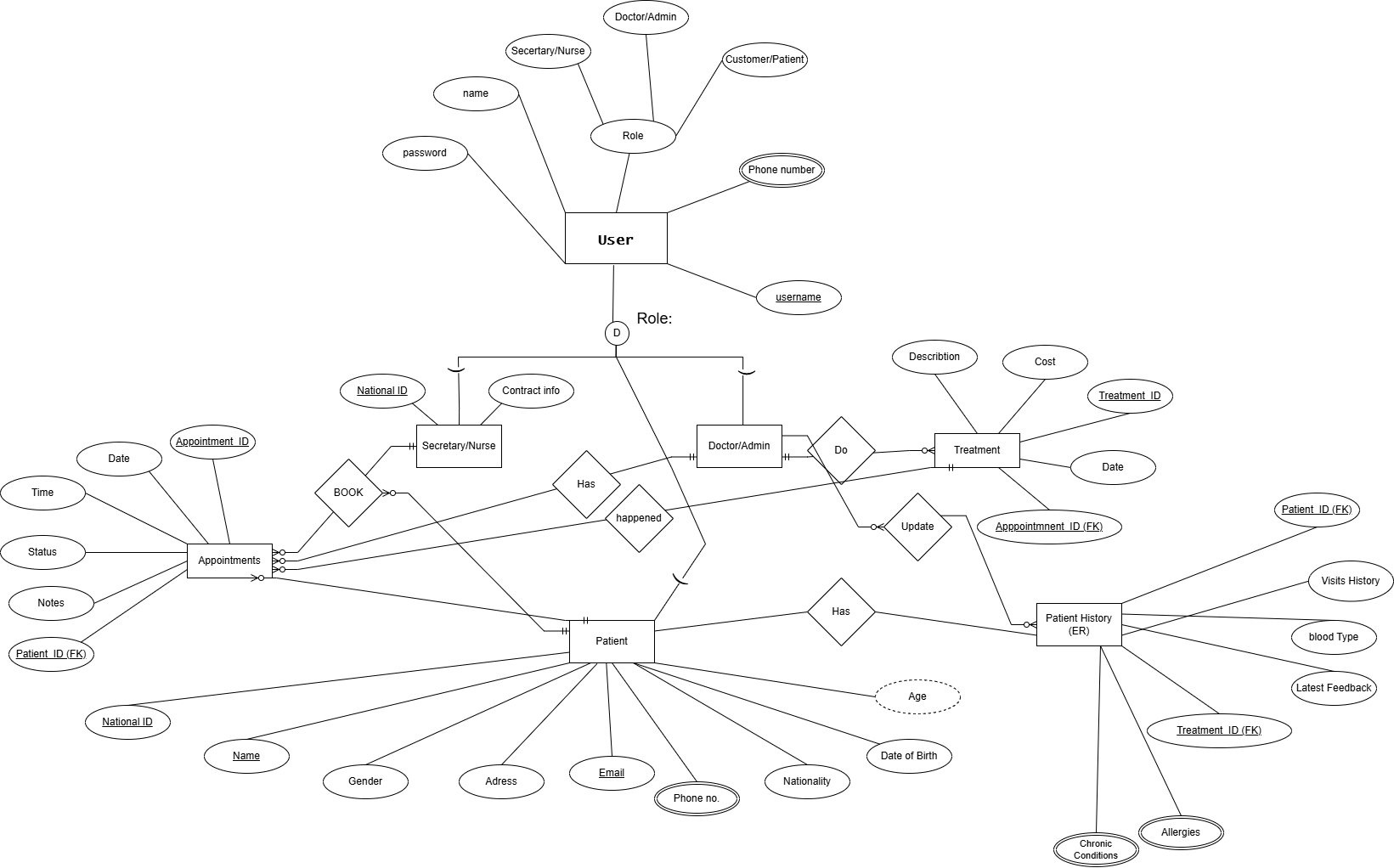


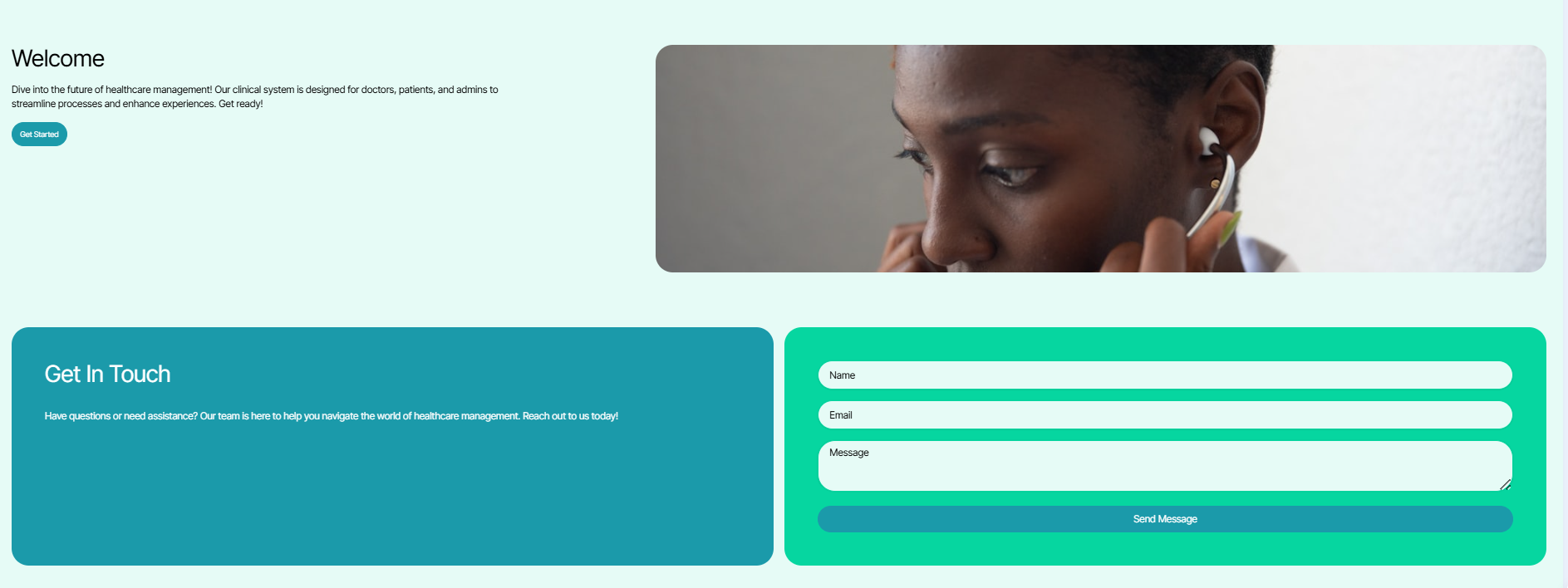
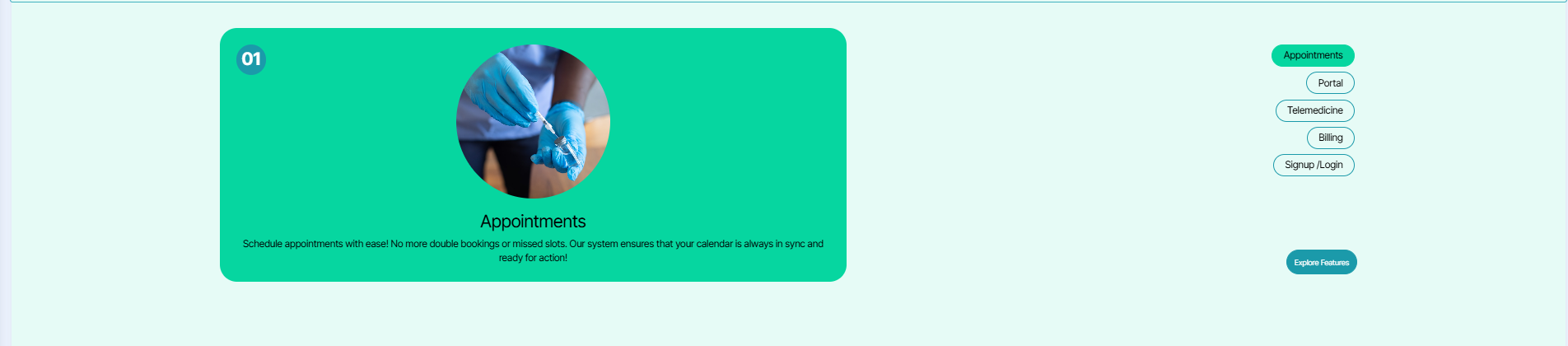
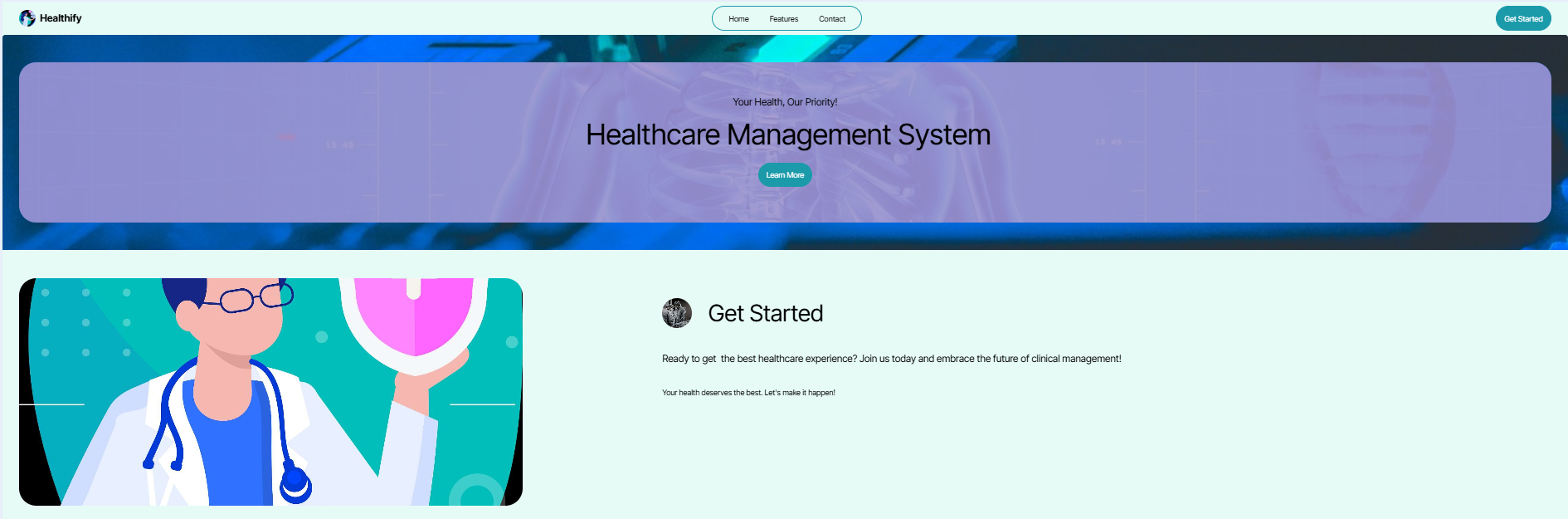
Sequence Diagram:

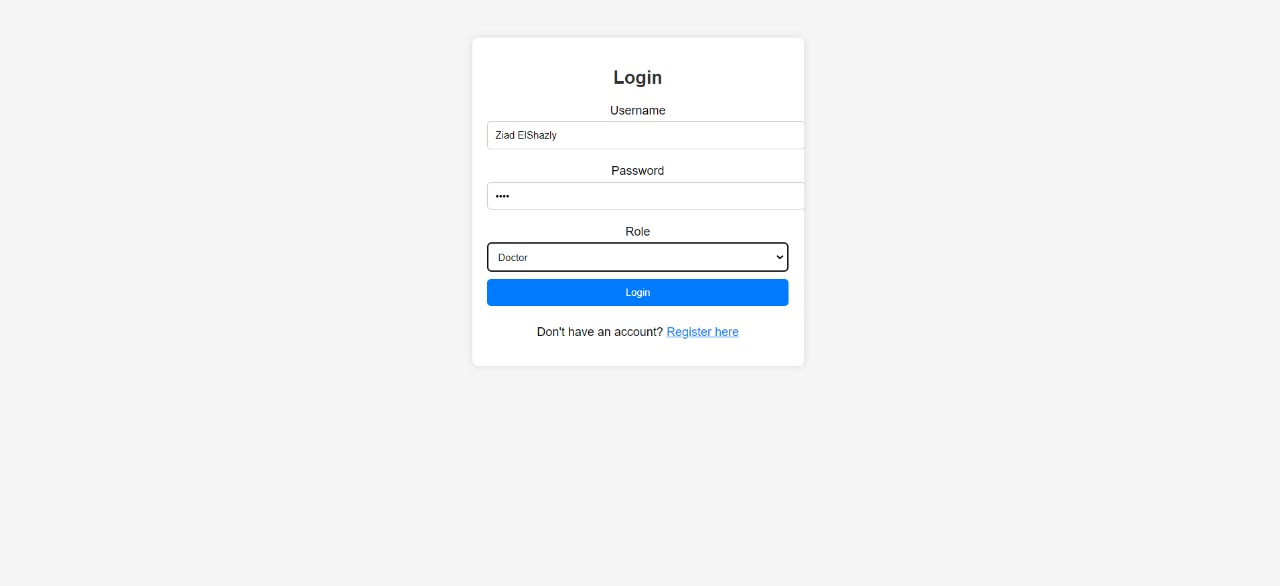


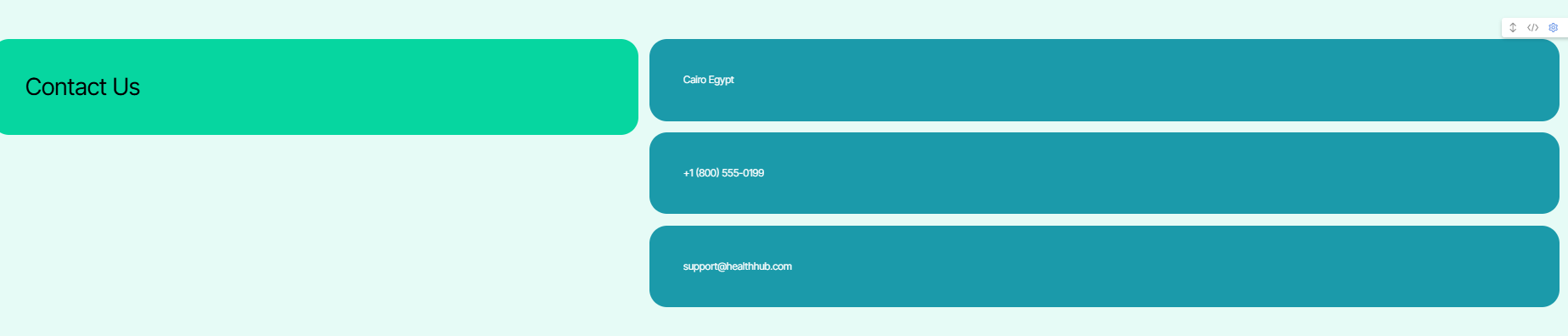


**Database Design (ERD):**



**Wireframes (Frontend Design Drawings)** 





**Conclusion**

The **Clinical Management System (CMS)** is designed to address key operational challenges faced by clinics, providing an efficient solution for managing patient appointments, records, and secure user access. By implementing role-based access and a user-friendly online reservation feature, this system aims to enhance patient convenience, streamline clinic workflows, and improve data security. This proposal outlines the initial Project structure, user personas, requirements, and system design, forming a solid foundation for development under the Agile method. Moving forward, our team will focus on delivering a Minimum Viable Product (MVP) in Phase One, with plans for iterative enhancements in Phase Two based on feedback and testing outcomes.