



**CEBU INSTITUTE OF TECHNOLOGY**  
**U N I V E R S I T Y**

# IT342-G4 SYSTEMS INTEGRATION AND ARCHITECTURE 1

---

## **FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)**

---

Project Title: MedLink

Prepared By: Josan B. Sumarago Date

of Submission: February 3, 2026

Version:

## **Table of Contents**

1. Introduction .....	3
1.1. Purpose .....	3

1.2.	Scope .....	3
1.3.	Definitions, Acronyms, and Abbreviations .....	3
2.	Overall Description .....	3
2.1.	System Perspective .....	3
2.2.	User Classes and Characteristics .....	4
2.3.	Operating Environment .....	4
2.4.	Assumptions and Dependencies .....	4
3.	System Features and Functional Requirements .....	4
3.1.	Feature 1: .....	4
3.2.	Feature 2: .....	5
4.	Non-Functional Requirements .....	5
5.	System Models (Diagrams) .....	6
5.1.	ERD .....	7
5.2.	Use Case Diagram .....	8
5.3.	Activity Diagram .....	9
5.4.	Class Diagram .....	10
5.5.	Sequence Diagram .....	11
6.	Appendices .....	11

## 1. Introduction

### 1.1. Purpose

The purpose of this document is to define the requirements for the **MedLink System**, a healthcare management portal structured around a **User-centered data model**. This document is intended for project stakeholders, developers, and testers to understand the system's functionality—ranging from **Guest User registration** to **Authenticated User session management**—as visualized in the accompanying system diagrams.

+4

### 1.2. Scope

Based on the system models, the initial scope of the MedLink System includes:

- **User Management & Registration:** Allowing Guest Users to create accounts with specific attributes including **Email, Password, and Role** (Patient/Admin/Doctor).

+2

- **Secure Authentication:** A complete workflow for **Login and Logout** that validates credentials and manages session state as detailed in the Activity and Sequence diagrams.

+2

- **Profile & Dashboard Access:** A protected interface where authenticated users can view their unique profile data retrieved from the backend.

+2

- **Relational Data Foundation:** A database architecture designed to support specialized user classes, specifically **Doctors** (license and specialization data) and **Patients** (demographics and medical vitals).

+1

### 1.3. Definitions, Acronyms, and Abbreviations

- **SRS:** Software Requirements Specification.
- **API:** Application Programming Interface (The communication layer between the React UI and Auth Service).

+1

- **JWT:** JSON Web Token (The secure token generated by the System to validate active sessions).

+2

- **UI:** User Interface (The frontend component where users interact with Registration and Login forms).

+2

- **DBMS:** Database Management System (Stores the relational entities defined in the ERD).

+1

- **DTO:** Data Transfer Object (Objects like `RegisterDto` and `LoginDto` used by the Auth Controller to process requests).

.

## 2. Overall Description

### 2.1. System Perspective

MedLink functions as a Client-Server application.

- Frontend: A responsive web interface (React.js) accessible via standard web browsers.
- Backend: A RESTful API (Spring Boot) that processes logic and connects to the database.
- Database: A relational database that stores user credentials and profile information.

### 2.2. User Classes and Characteristics

User Classes and Characteristics

1. Guest User: An unauthenticated visitor who can only access public pages (Landing Page, Login, Registration).
2. Authenticated User (Patient/Admin): A registered user who has successfully logged in. They have access to protected features like the Dashboard/Profile and Logout functions.

### 2.3. Operating Environment

- Client Side: Modern web browser (Chrome, Firefox, Edge, Safari) with JavaScript enabled.
- Server Side: Java Runtime Environment (JRE) 17 or higher.
- Database: MySQL or PostgreSQL server.
- Network: Stable internet connection for API communication.

### 2.4. Assumptions and Dependencies

- Assumption: Users have a valid email address for registration.
- Assumption: The browser supports local storage or cookies for session management.
- Dependency: The system relies on the availability of the hosting server and database connectivity.

### 3. System Features and Functional Requirements

Describe each major feature of the system and its functional requirements.

#### 3.1. Feature 1:

Description: Allows a Guest User to create a new account by providing necessary personal details.

Functional Requirements:

- FR-01: The system shall provide a registration form requiring Name, Email, and Password.
- FR-02: The system shall validate that the email is in a correct format and does not already exist in the database.
- FR-03: The system shall encrypt the user's password using a hashing algorithm (e.g., BCrypt) before saving it to the database.
- FR-04: Upon successful registration, the system shall redirect the user to the Login page.

#### 3.2. Feature 2: User Authentication (Login)

Description: Allows registered users to access their accounts securely using their credentials.

Functional Requirements:

- FR-05: The system shall validate the user's email and password against stored records.
- FR-06: The system shall generate a secure session token (JWT) upon successful login.
- FR-07: The system shall display an error message for invalid credentials (e.g., "Invalid email or password").
- FR-08: The system shall redirect the user to the Dashboard upon successful login.

#### 3.3. Feature 3: Profile Management & Logout

Description: Allows authenticated users to view their dashboard and securely end their session.

Functional Requirements:

- FR-09: The system shall display the User Dashboard only to users holding a valid active session token.
- FR-10: The Dashboard shall display the user's name and role retrieved from the backend.

- FR-11: The system shall provide a "Logout" button that invalidates the user's session token or clears client-side storage.
- FR-12: Upon logout, the user shall be immediately redirected to the Login page/Landing page.

## 4. Non-Functional Requirements

### Security:

- Passwords must never be stored in plain text; they must be hashed.
- API endpoints for the Profile must be protected against unauthorized access (401 Unauthorized).

### Performance:

- The system shall respond to login/registration requests within 2 seconds under normal load.

### Reliability:

- The system should be available 99% of the time during business hours.

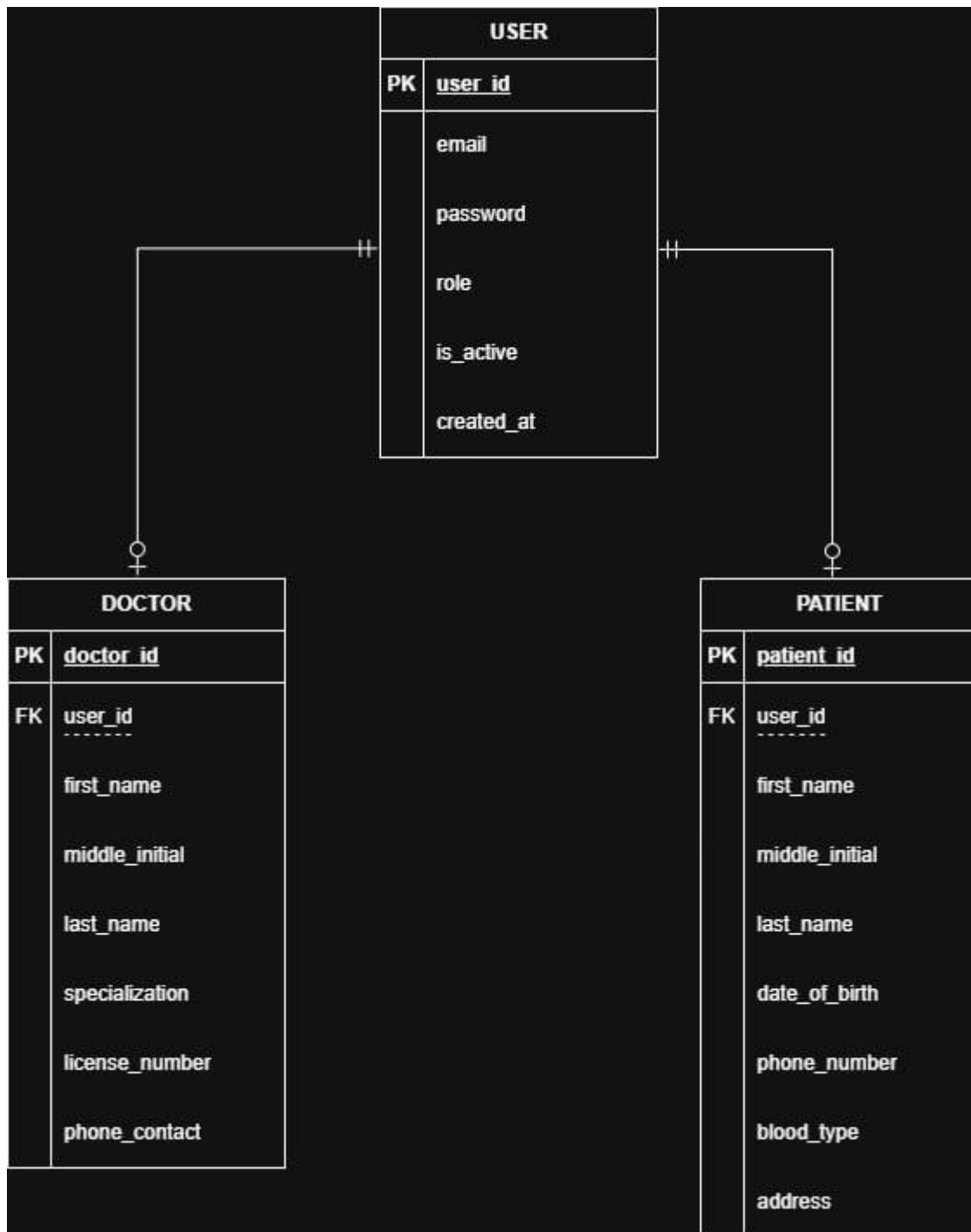
### Usability:

- The user interface shall be responsive and usable on both desktop and mobile screens.
- Error messages shall be clear and user-friendly (non-technical).

## 5. System Models (Diagrams)

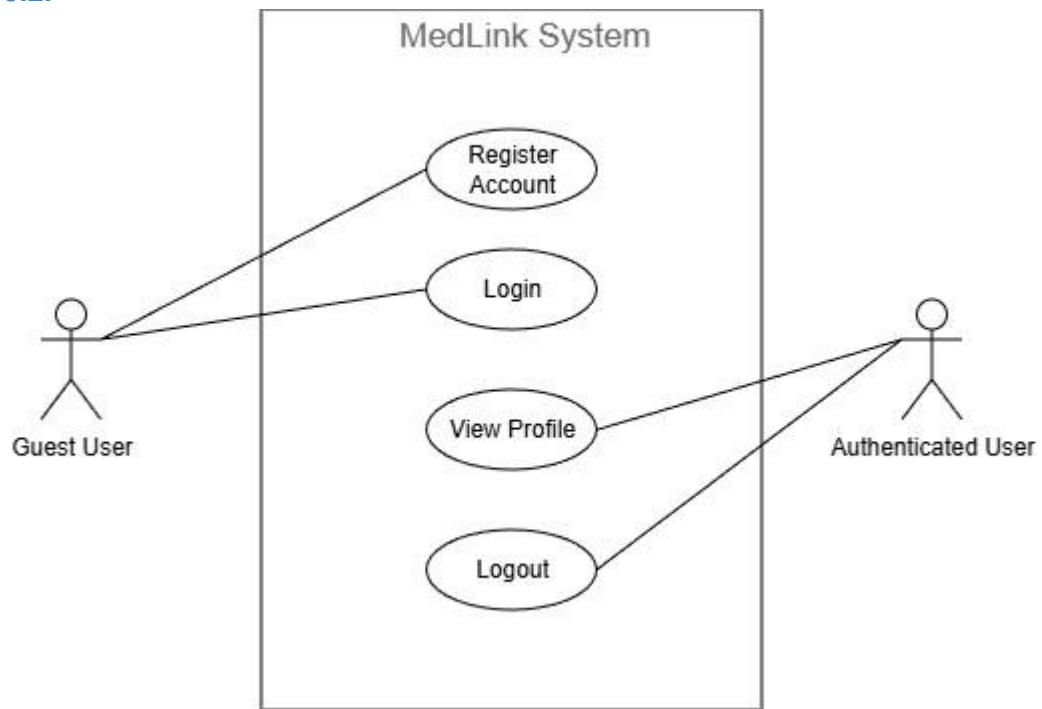
*Insert the necessary diagrams for the system:*

## 5.1. ERD



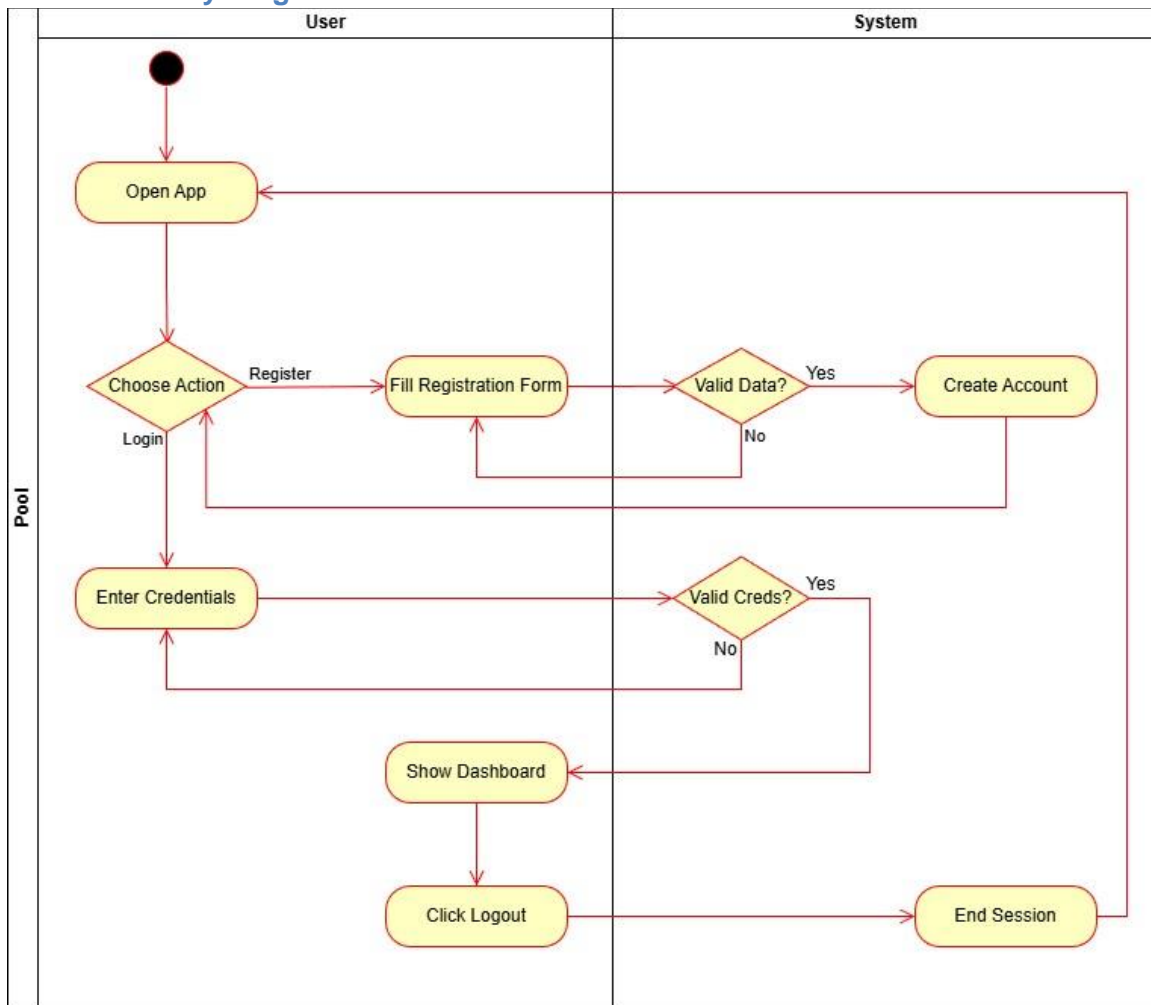
Use Case Diagram

## 5.2.



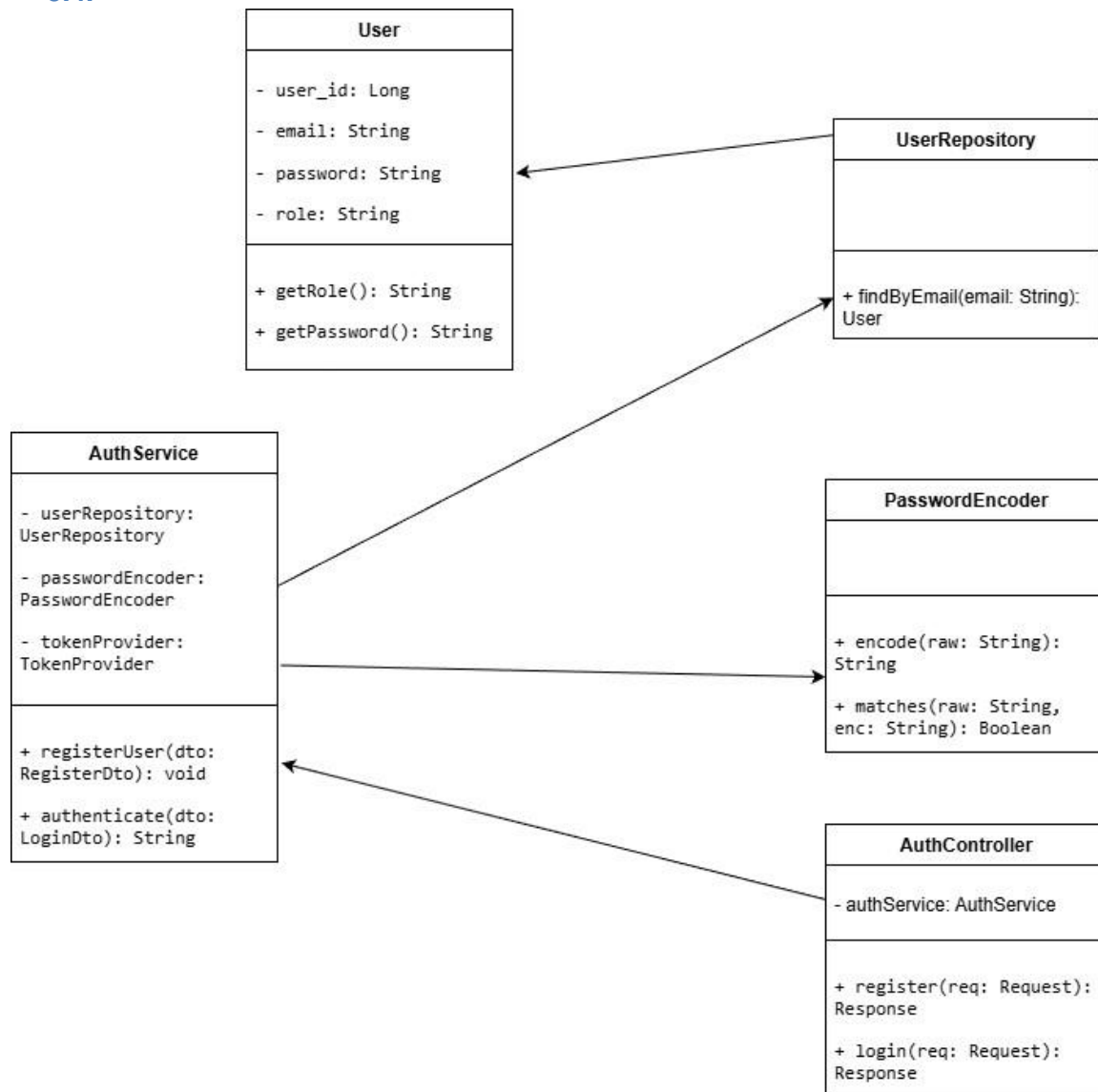


### 5.3. Activity Diagram



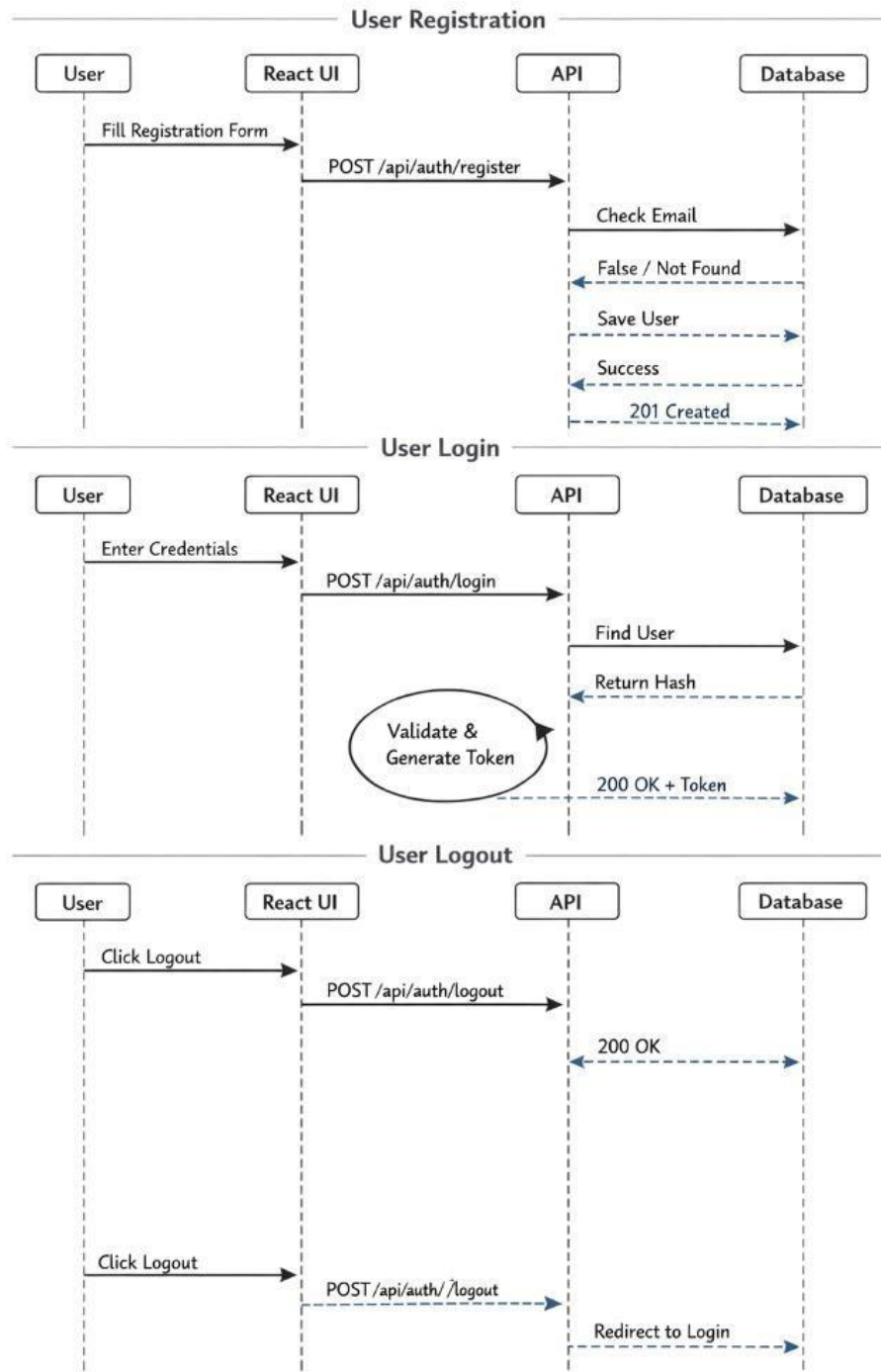
### Class Diagram

## 5.4.



Sequence Diagram

## 5.5.



## 6. Appendices

Include any additional information, references, or support materials.

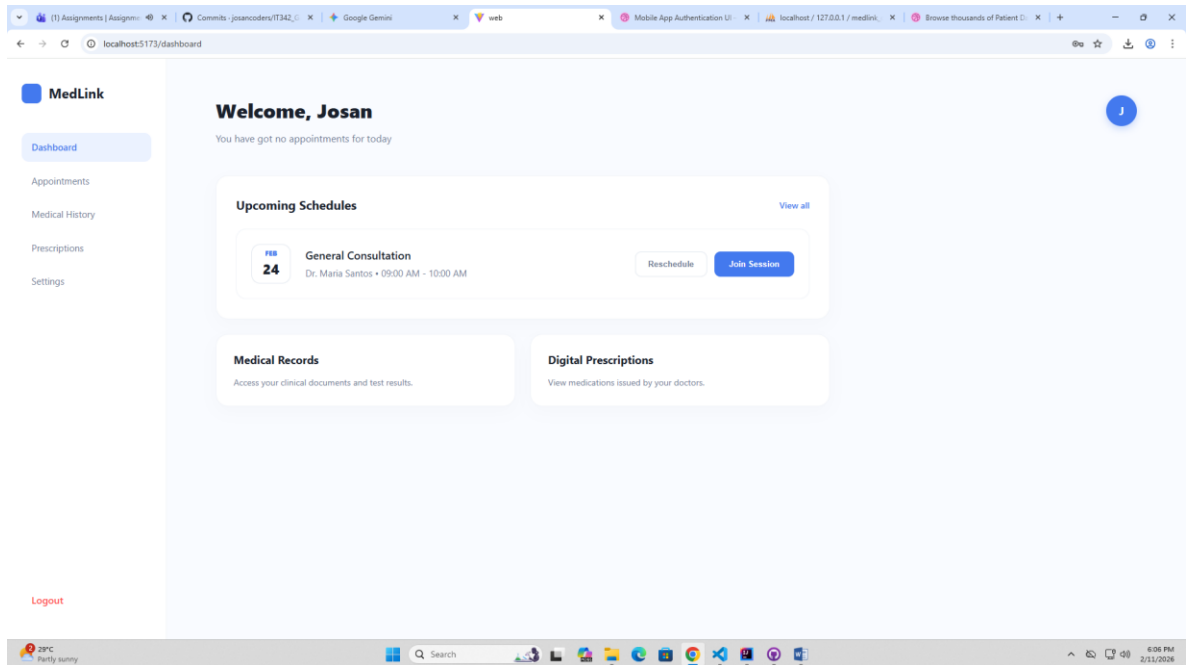
## 5.6. Login

The screenshot shows a web browser window with the URL `localhost:5173`. The page displays the MedLink logo at the top, followed by the heading "Sign In". Below this, there are two input fields: "Email Address" with the placeholder text "Enter your email" and "Password" with the placeholder text "Enter your password". A blue "Sign in" button is positioned below the password field. At the bottom of the form, there is a link that says "Don't have an account? [Register here](#)". The browser's taskbar at the bottom shows the system clock as 6:06 PM on 2/11/2026.

## Registration

The screenshot shows a web browser window with the URL `localhost:5173/register`. The page displays the MedLink logo at the top, followed by the heading "Create Account". Below this, there are two input fields: "Email Address" with the placeholder text "name@example.com" and "Password" with the placeholder text "Create a password". A blue "Register" button is positioned below the password field. At the bottom of the form, there is a link that says "Already have an account? [Login here](#)". The browser's taskbar at the bottom shows the system clock as 6:06 PM on 2/11/2026.

## 5.7. Dashboard



## Profile Page

