CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-Section SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: User Registration & Authentication

Prepared By: Ronel Atillo

Date of Submission: 2/8/2026

Version: 3

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 3
   2.3. Operating Environment 3
   2.4. Assumptions and Dependencies 3
3. System Features and Functional Requirements 3
   3.1. Feature 1: 3
   3.2. Feature 2: 3
4. Non-Functional Requirements 3
5. System Models (Diagrams) 4
   5.1. ERD 4
   5.2. Use Case Diagram 4
   5.3. Activity Diagram 4
   5.4. Class Diagram 4
   5.5. Sequence Diagram 4
6. Appendices 4

1. Introduction

1.1. Purpose

The purpose of the system is to show the requirements for user registration and authentication systems. It is also to show and define to developers and other stakeholders who need a clear process, overview of the system, features, scope, and technical requirements.

1.2. Scope

The system allows users to login, register, view their own profile or dashboard, and logout. Just like in my design, the dashboard or profile are being protected, so guest users can't proceed to such pages when they are not authenticated. So in simple words, the scope is just only authentication and user data management.

1.3. Definitions, Acronyms, and Abbreviations

- API - Application Programming Interface
- UI - User Interface
- Auth - Authentication
- JWT - JSON Web Token
- ERD - Entity Relationship Diagram
- DTO - Data Transfer Object

2. Overall Description

2.1. System Perspective

The system is a web app that has React UI, Spring Boot Backend, and a relational database. The UI communicates with the backend through APIs, the backend handles validations, encrypting passwords, storing users in databases, database operations, and token creation.

2.2. User Classes and Characteristics

Authenticated User - a logged in user that can access his own dashboard and profile, and can logout.

Guest user - a not logged in user, it can only register and log in.

2.3. Operating Environment

Specify the hardware, software, and tools required to operate the system.

Hardware:

- Laptop or desktop computer
- Has a 8gb of ram
- 64gb hard disk

Software:

- Database: MySQL
- Web browser: Chrome
- Frontend runtime: node.js
- Backend runtime: java jdk 21

Backend:

- IDE: VS Code and Intellij
- Database tool: MySQL workbench
- Version control: Git and Github
- Build tool: Gradle

2.4. Assumptions and Dependencies

- Emails should be unique
- Passwords are stored in the database as hash passwords, not plain text.
- Token-based authentication is used (JWT)
- Backend and database services must be running and reachable.

3. System Features and Functional Requirements

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

3.1. Feature 1: User Registration

Description: It allows users to create an account with their inputted information.
Functional Requirements:
- It has a feature where users can input their email, password, address, and phone number.
- The system will validate the inputted information if there are any invalid inputs such as email without '@' symbol.
- The system also prevents duplicate email addresses during registration.

3.2. Feature 2: User login

Description: It allows registered users to log in or authenticate, so that they can access protected pages such as dashboard and profile.
Functional Requirements:
- The system will verify their inputted credentials
- It will then return a token (JWT) as login is successful.
- The system also shows error messages if such credentials are invalid.

3.3. Feature 3: Dashboard / Profile

Description: It allows registered users to log in or authenticate, so that they can access protected pages such as dashboard and profile.
Functional Requirements:
- If the user is not authenticated, then the system will restrict the access to the pages.
- The system will display user information after logging in.

3.4. Feature 4: Logout

Description: This feature will allow users to end their session.
Functional Requirements:
- The system can invalidate the created or current token.
- The system will redirect the user to the login page.

4. Non-Functional Requirements

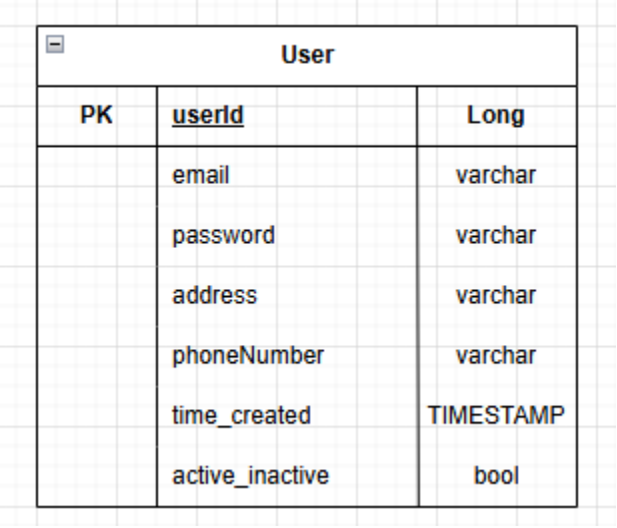
Specify system quality attributes such as performance, security, usability, reliability, etc.

- Performance: Login/Registration responses should complete within 2-3 seconds
- Security: Passwords are hash before storing in the Databases. Protected Pages require login or authentication.
- Usability: UI is simple and clear with error messages.
- Scalability: Must support multiple active users.

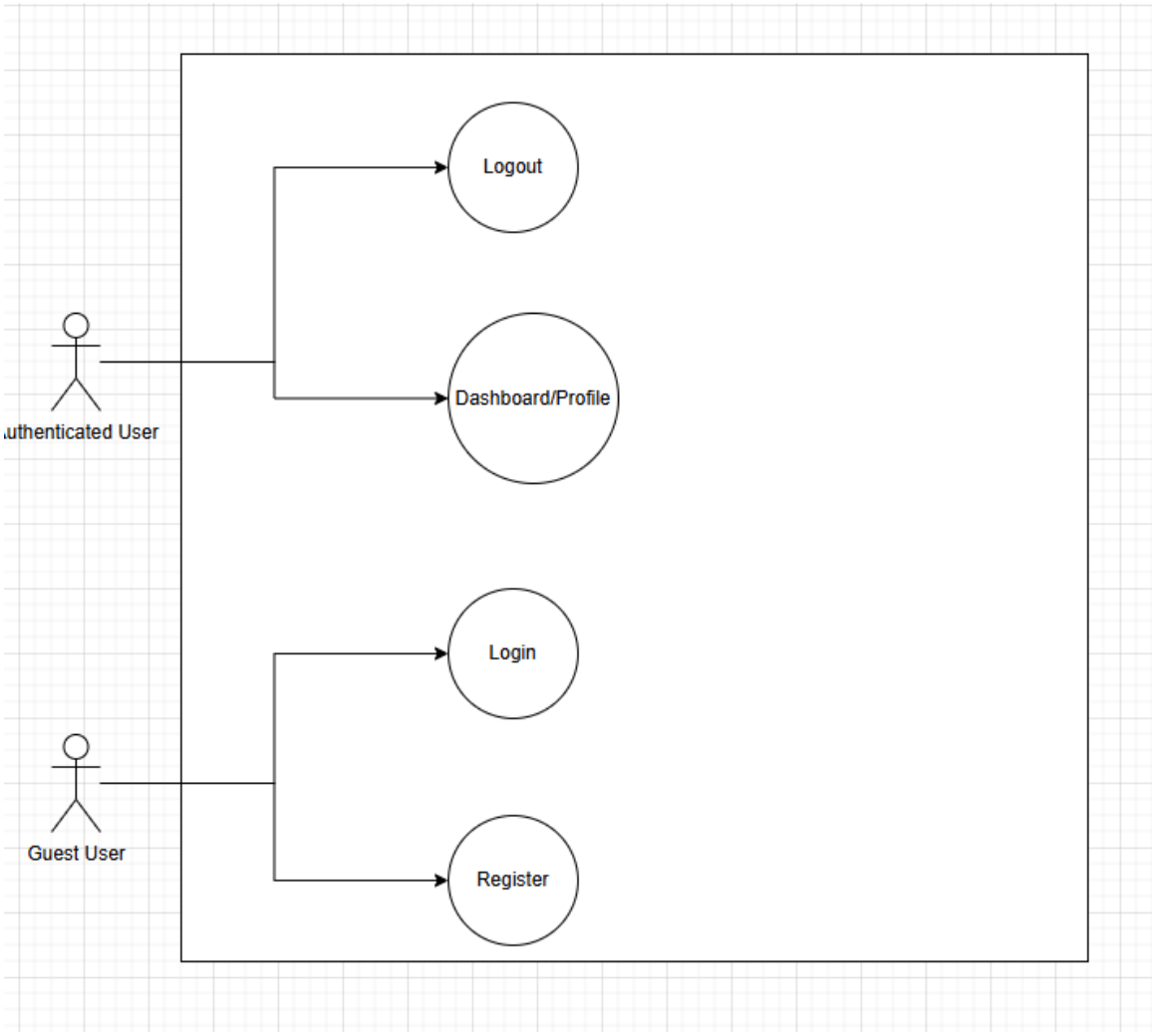
5. System Models (Diagrams)

Insert the necessary diagrams for the system:

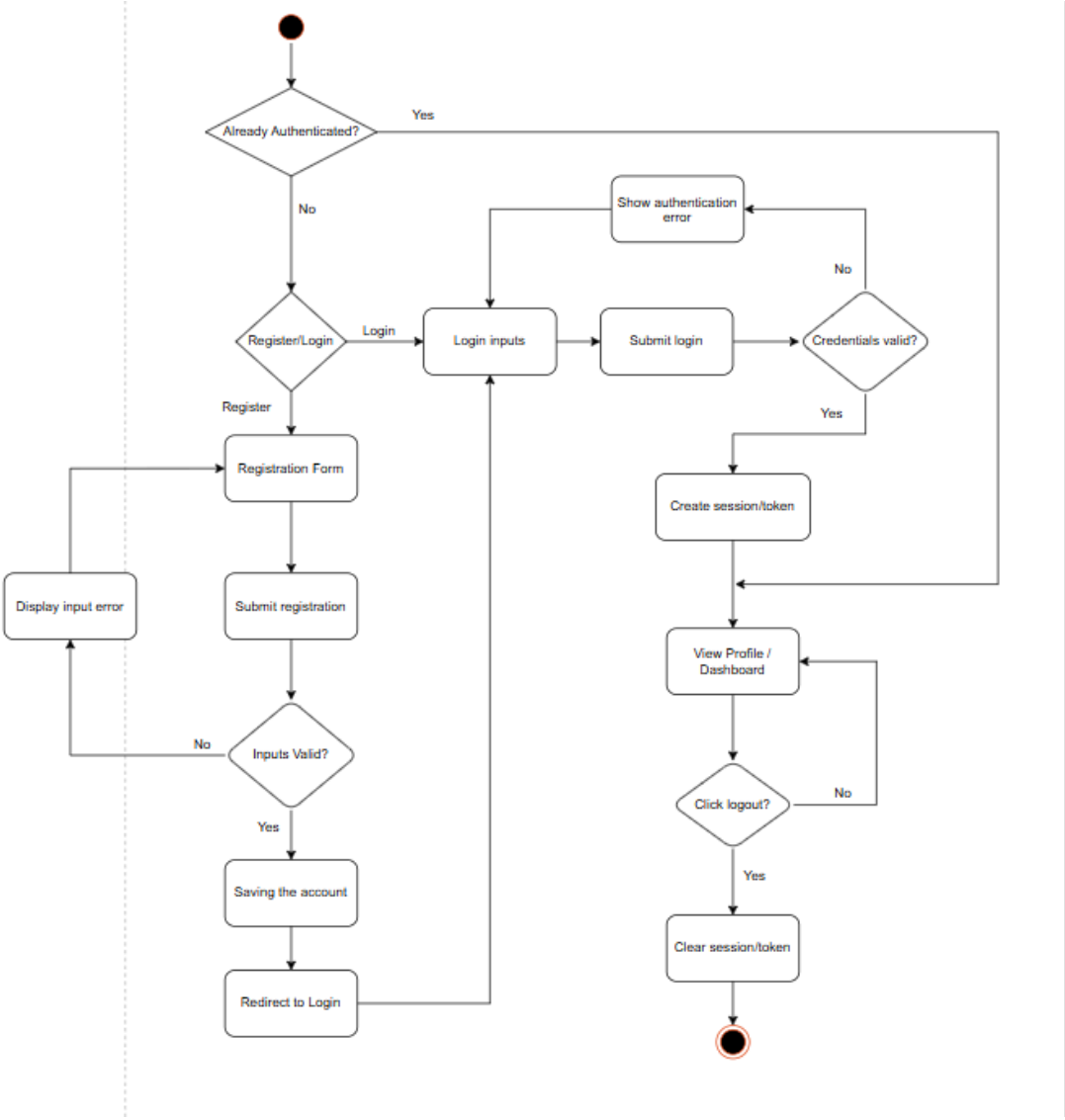
5.1. ERD



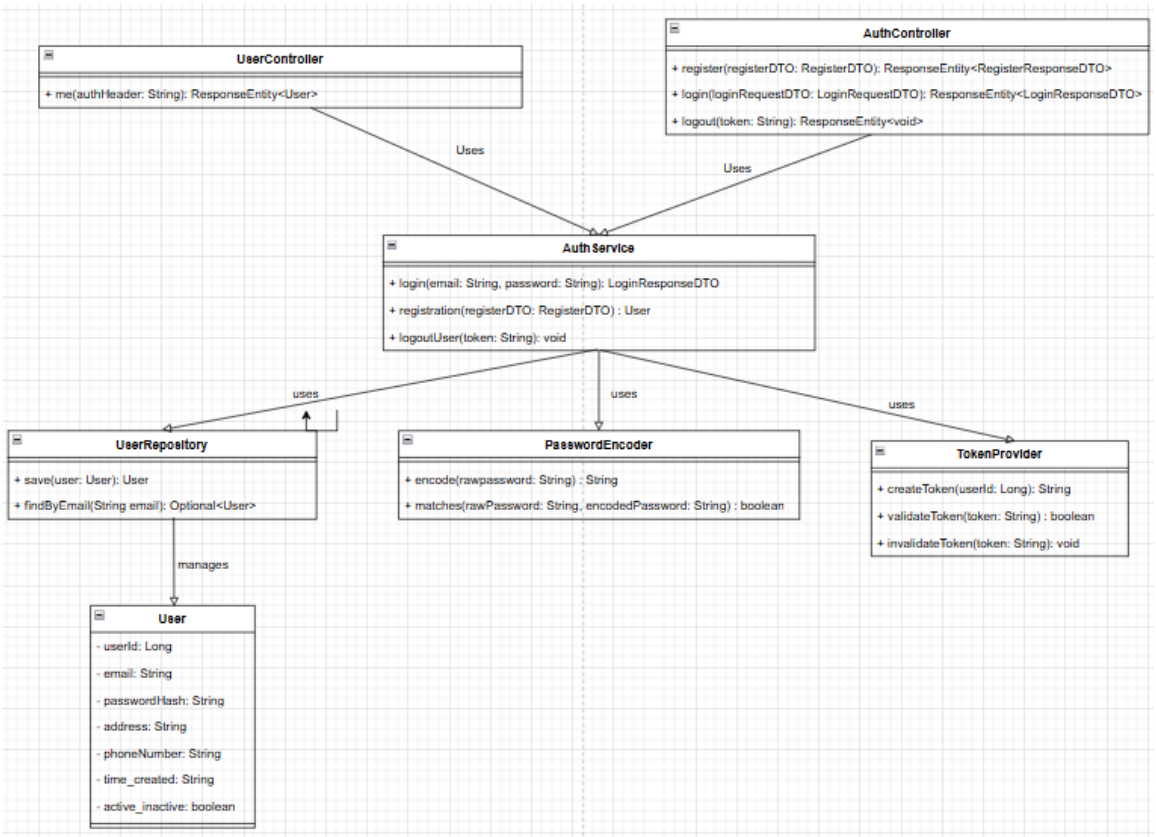
5.2. Use Case Diagram



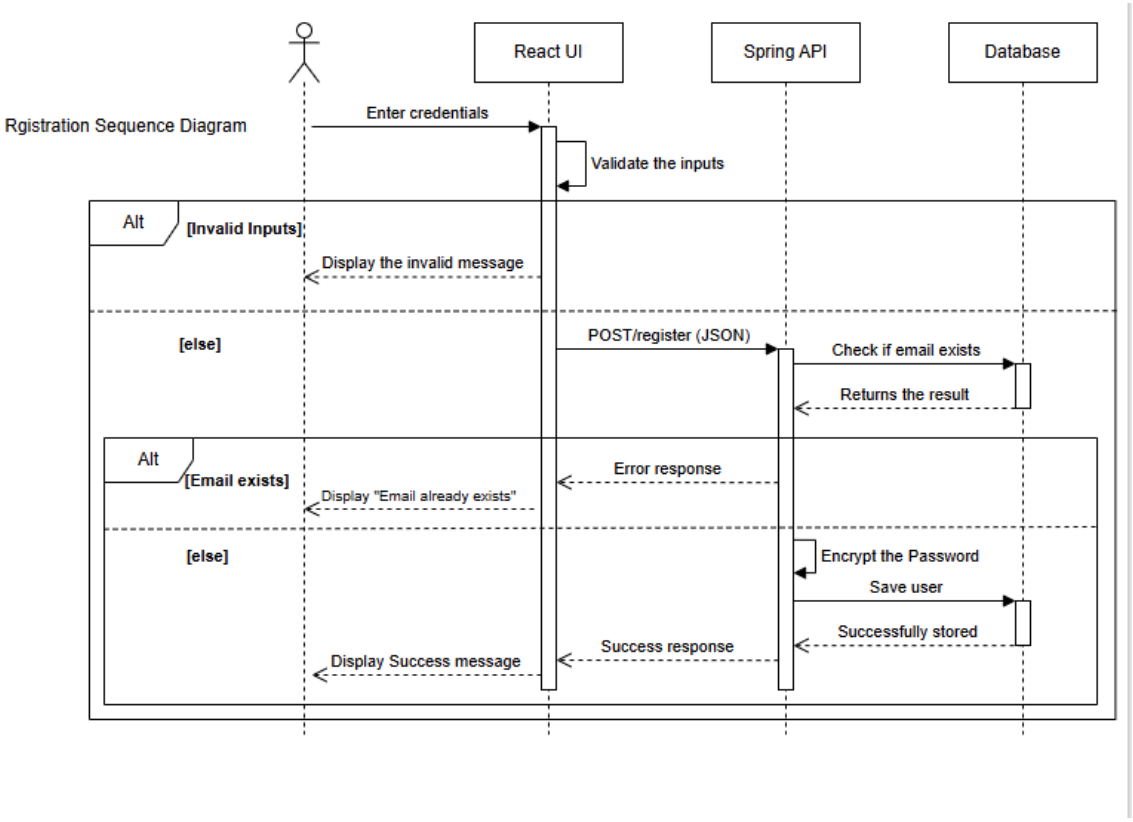
5.3. Activity Diagram

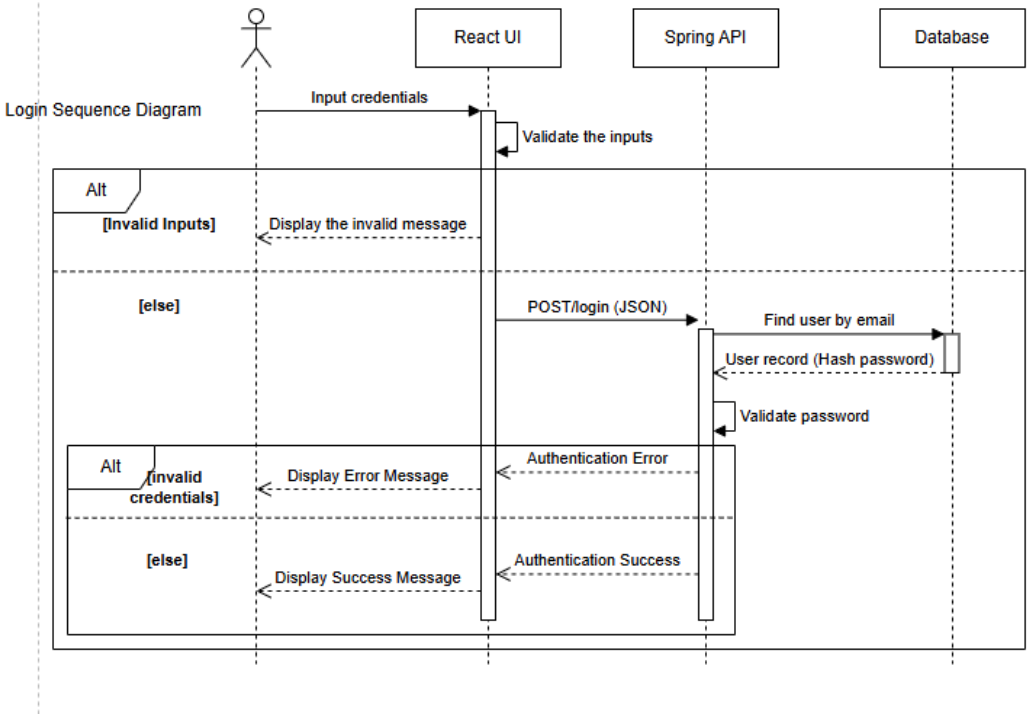


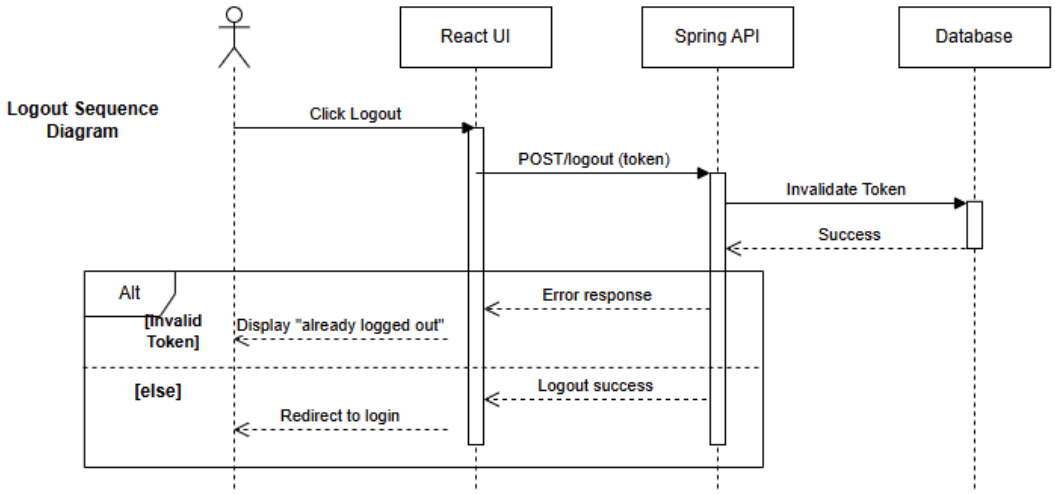
5.4. Class Diagram



5.5. Sequence Diagram





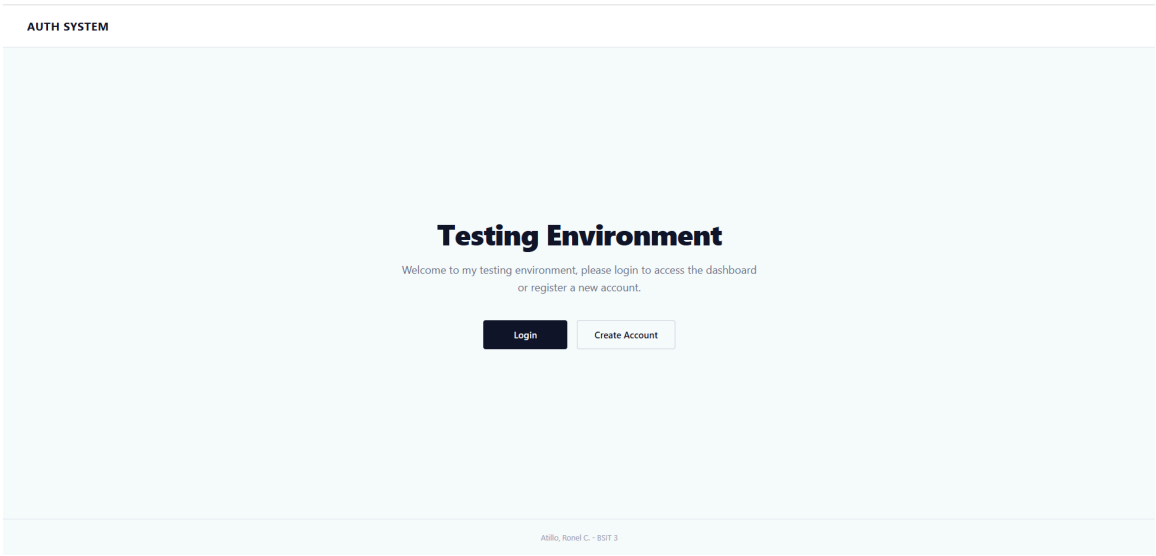


6. Appendices

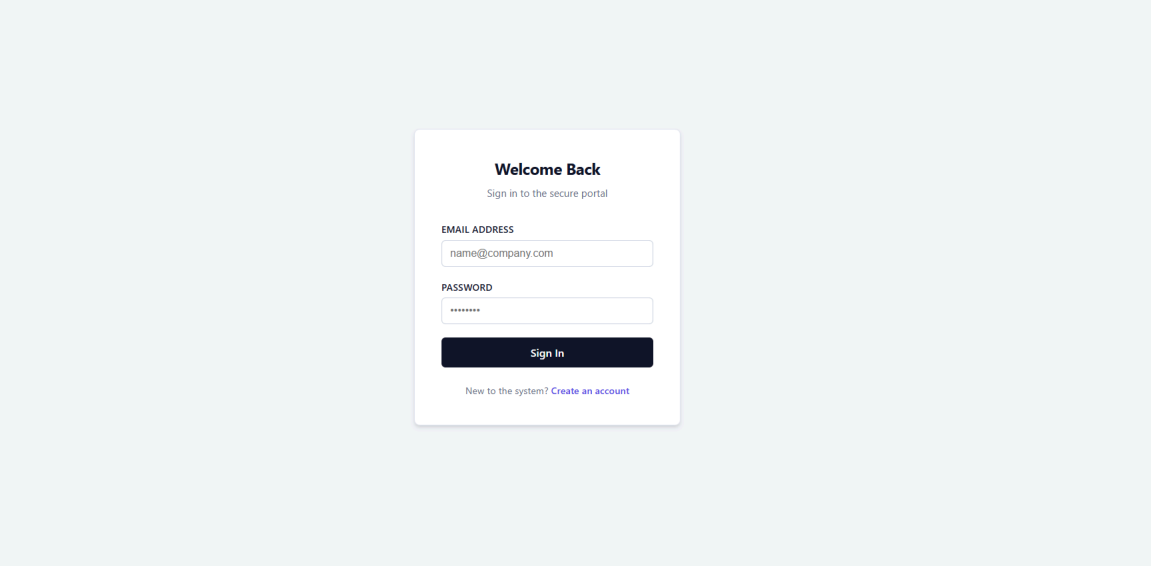
Include any additional information, references, or support materials.

- draw.io
- Google.drive
- AI: ChatGPT
- https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/
- https://www.perforce.com/blog/alm/what-are-non-functional-requirements-examples

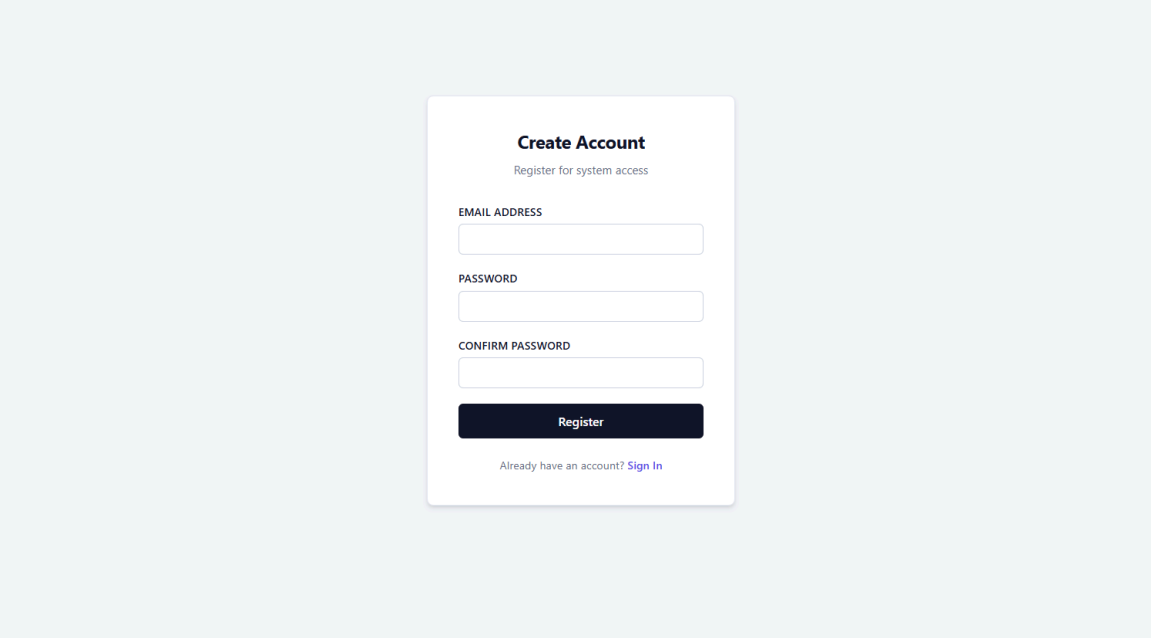
Landing Page:



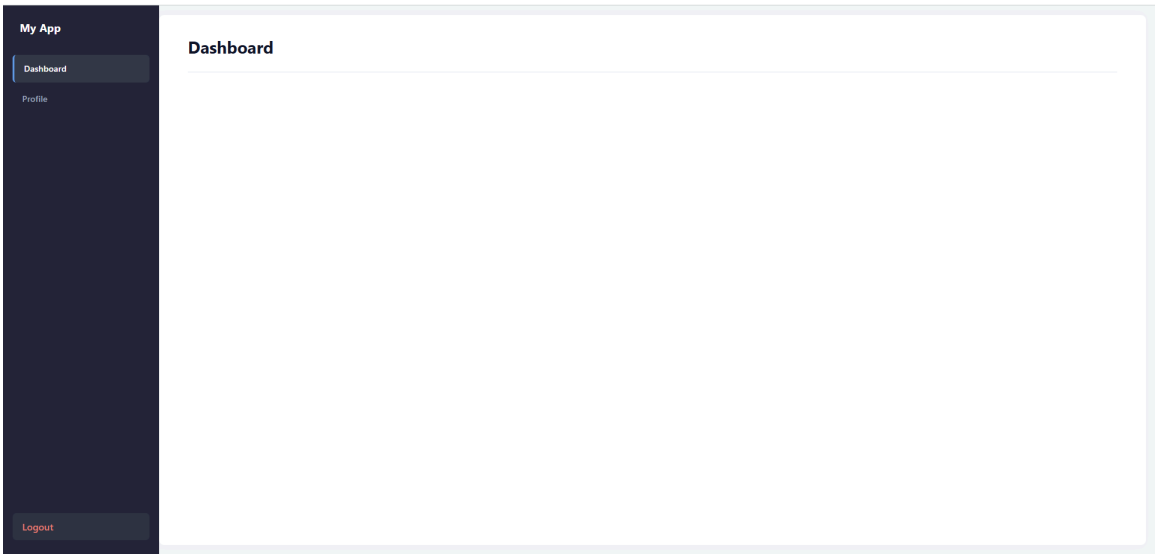
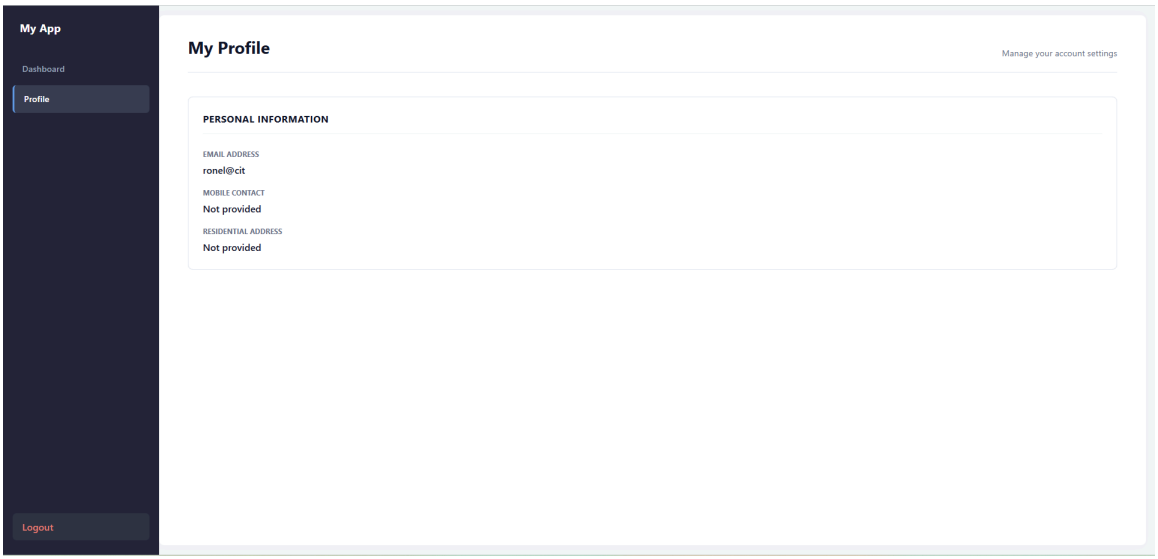
Login Page:



Registration Page:



Dashboard:



Profile: