



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

IT342-Section SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: User Registration and Authentication

Prepared By: Aloria, Sophia Bianca

Date of Submission: 2/19/2026

Version: 2

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 this system is to manage user authentication for a web application. The system will allow users to register accounts, log in, access their profile, and log out securely.

1.2. Scope

This system will handle:

- User registration with firstname, lastname, email, username, and password
- Login authentication with email/username and password
- Profile access for authenticated users only
- Logout functionality
- Prevention of unauthorized access to protected pages

Boundaries:

- Does **not** include password recovery/reset
- Does **not** include multi-factor authentication
- Focused on a single web application environment

1.3. Definitions, Acronyms, and Abbreviations

Term & Definition

UI -Interface

ERD -Entity Relationship Diagram

Auth -Authentication

API -Application Programming Interface

JWT -JSON Web Token, for secure session management

React -Frontend library for building UI

Spring Boot -Backend Java framework

2. Overall Description

2.1. System Perspective

The system is a **web-based authentication module** integrated into a larger application. It communicates between:

- **Frontend (React UI)** for user interaction
- **Backend (Spring Boot API)** for authentication logic
- **Database** to store user credentials and session information

2.2. User Classes and Characteristics

User Type	Characteristics
Guest User	Can register and log in; cannot access protected pages
Authenticated User	Has logged in; can view profile and log out

2.3. Operating Environment

- **Frontend:** ReactJS
- **Backend:** Spring Boot
- **Database:** MySQL/PostgreSQL
- **Tools:** Postman (API testing), Git, Maven/Gradle

2.4. Assumptions and Dependencies

- Users have internet access and a modern web browser
- Database is available and properly configured
- Backend and frontend will be hosted on servers accessible to users
- JWT or session management will be used for authentication

3. System Features and Functional Requirements

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

3.1. Feature 1: User Registration

Description:

Allows a guest user to create a new account with firstname, lastname, email, username, and password.

Functional Requirements:

- User provides firstname, lastname, email, username, and password
- System validates input (email format, password strength)
- System stores hashed password in database
- Confirmation message is displayed

3.2. Feature 2: User Login

Description:

Allows a registered user to authenticate and access protected pages.

Functional Requirements:

- User submits email/username and password
- System validates credentials
- System creates a session or JWT token
- Authenticated user is redirected to profile

3.3 Feature 3: View Profile

Description:

Allows authenticated users to view their personal information and account details.

Functional Requirements:

- Protected route checks authentication
- User information is retrieved from the database
- Dashboard displays username, email, and other relevant data

3.4 Feature 3: Logout

Description:

Allows authenticated users to log out and invalidate their session.

Functional Requirements:

- System terminates the session/JWT
- User is redirected to login page
- Protected pages are inaccessible after logout

1. Non-Functional Requirements

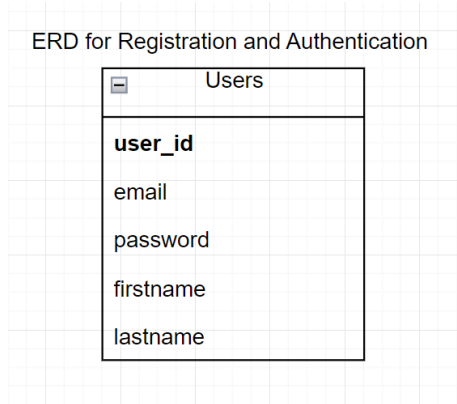
Specify system quality attributes such as performance, security, usability, reliability, etc.

- **Performance:** API responses under 500ms
- **Security:** Passwords hashed, JWT tokens used
- **Usability:** Clear error messages for login/registration failures
- **Reliability:** System available 99% of the time
- **Scalability:** Support for thousands of concurrent users

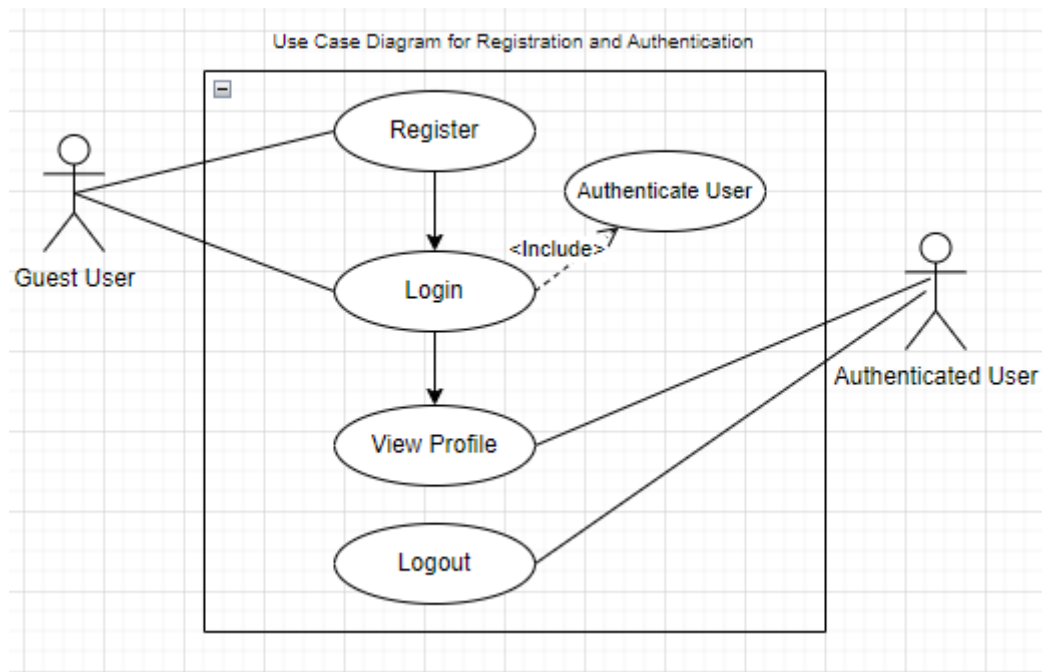
2. System Models (Diagrams)

Insert the necessary diagrams for the system:

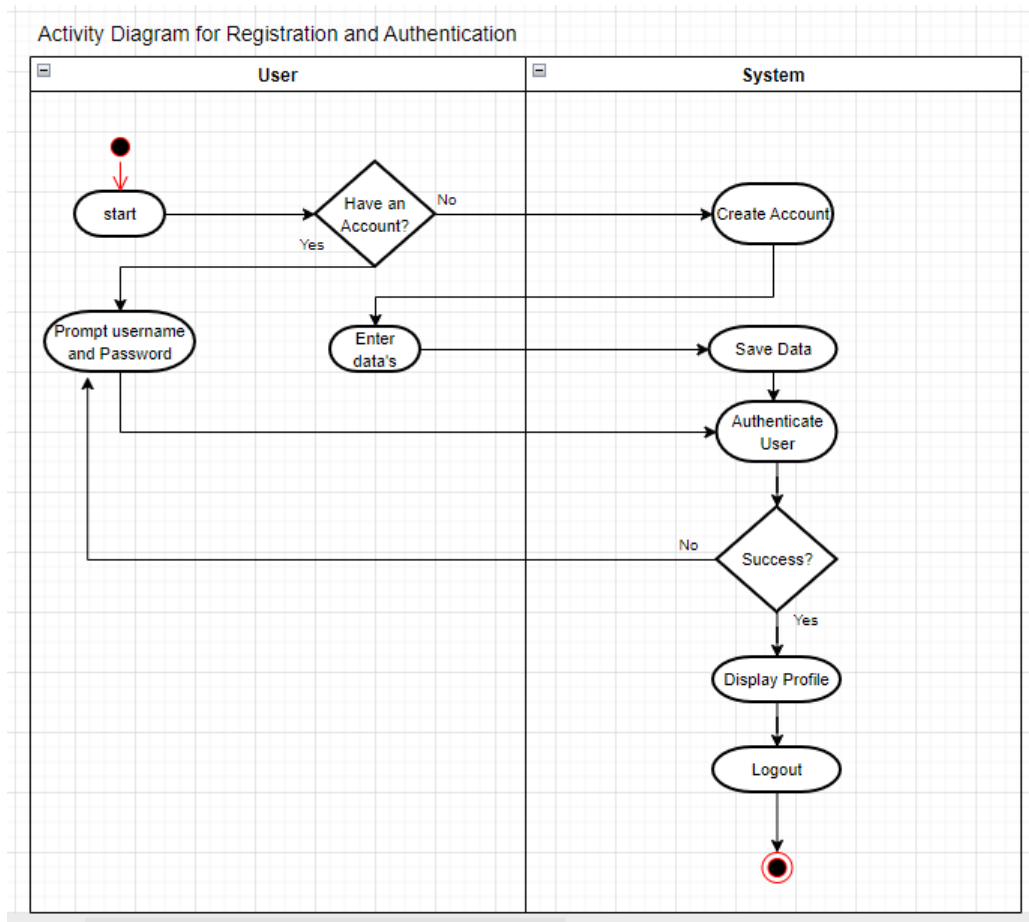
2.1. ERD



2.2. Use Case Diagram

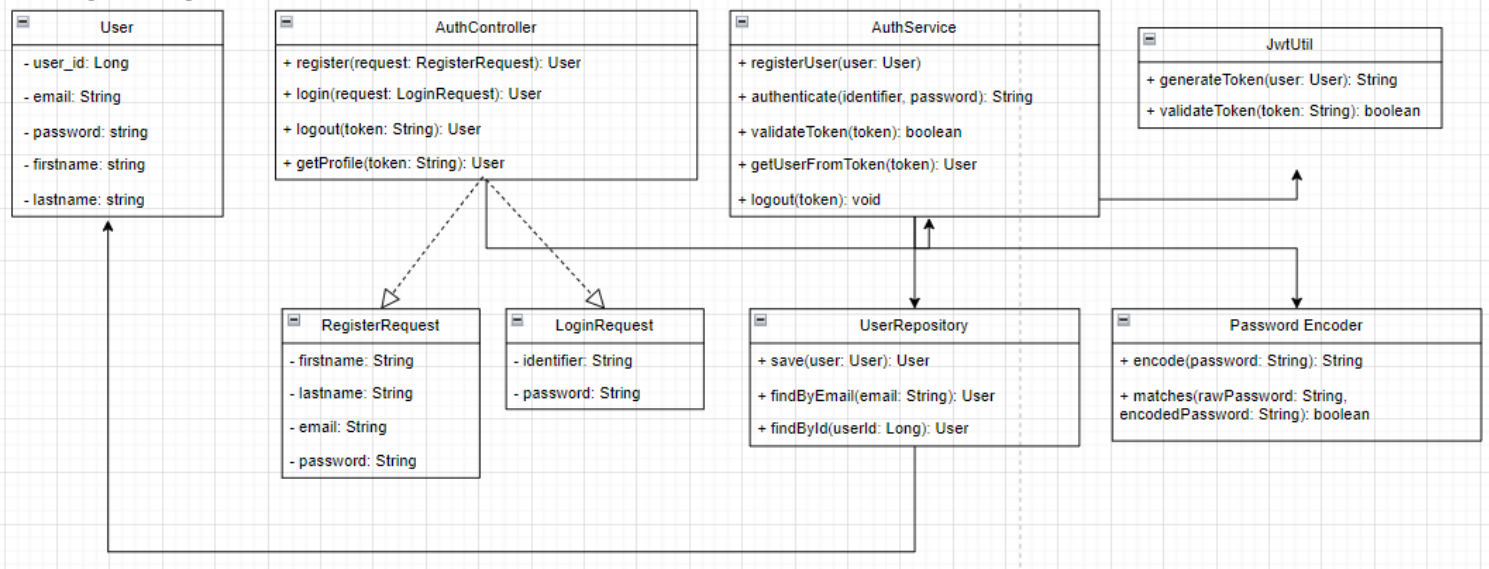


2.3. Activity Diagram

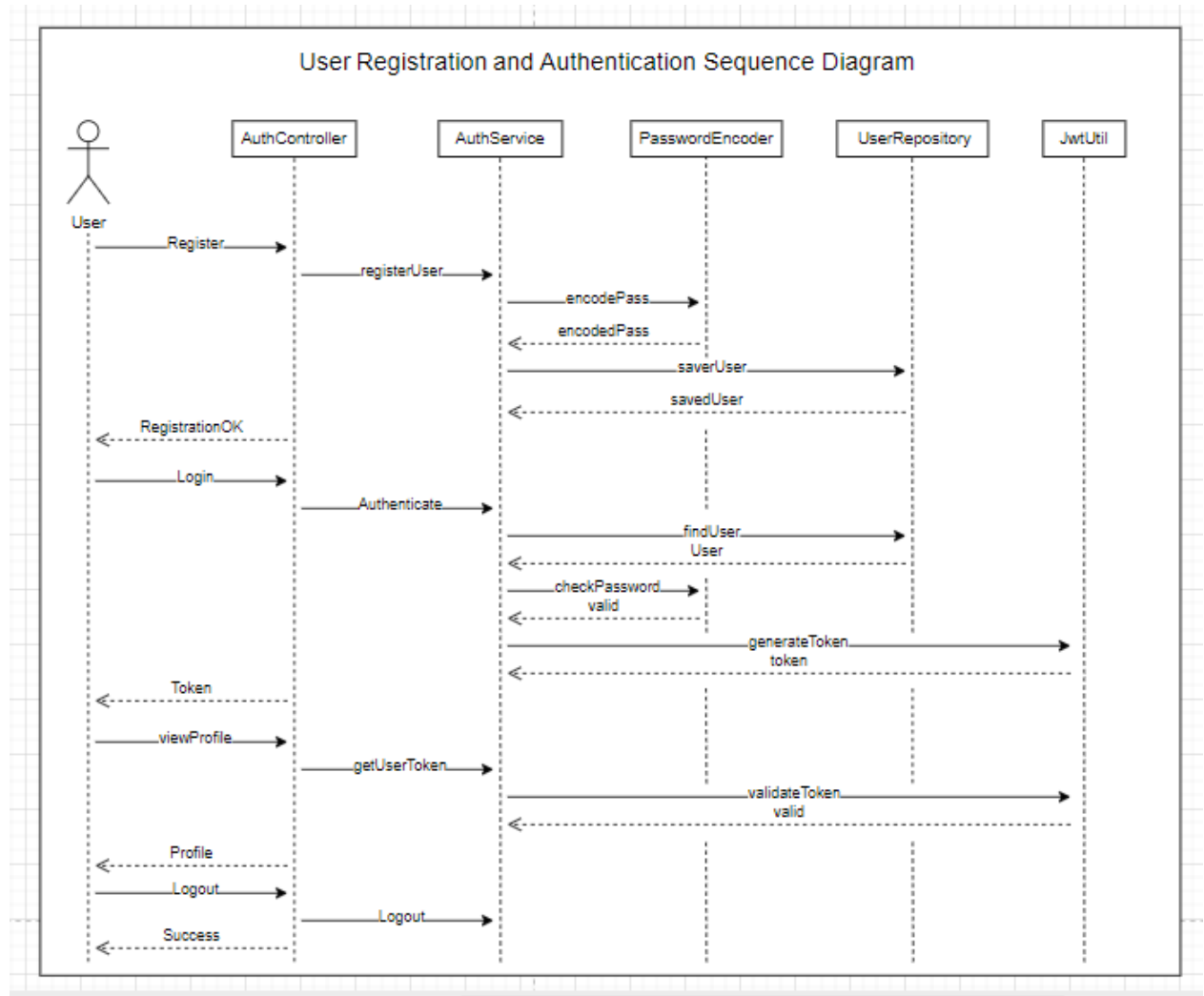


2.4. Class Diagram

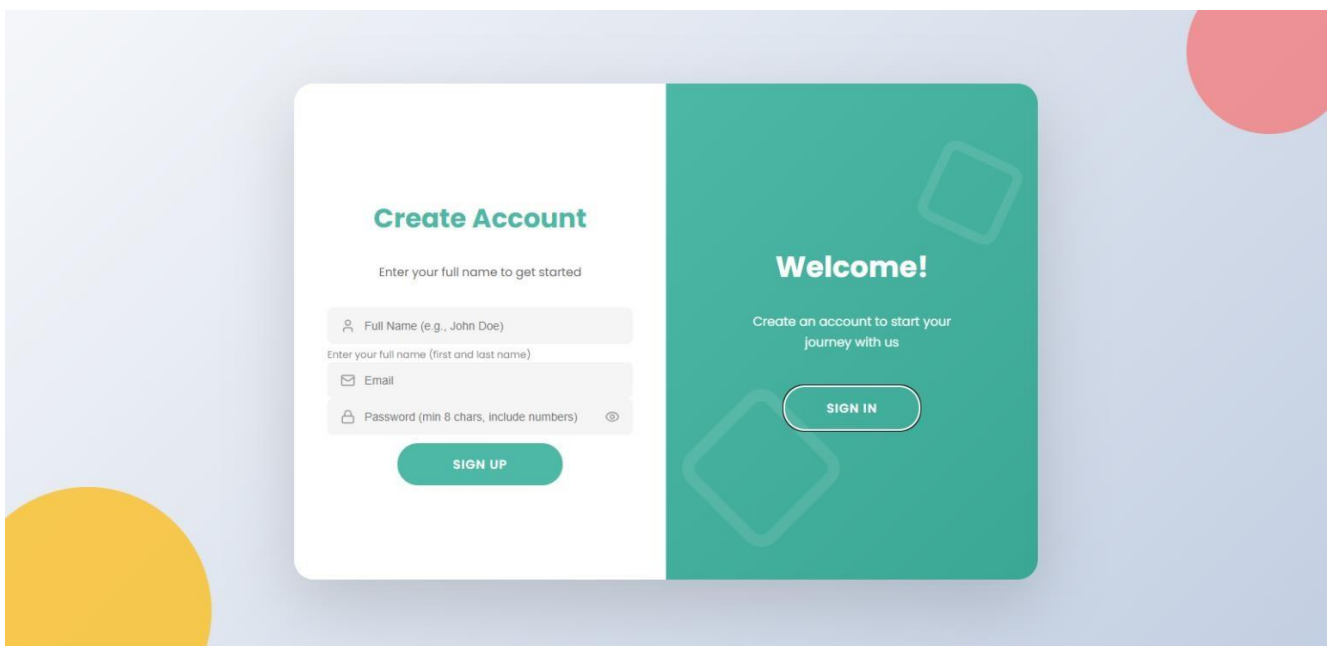
Class Diagram for Registration and Authentication

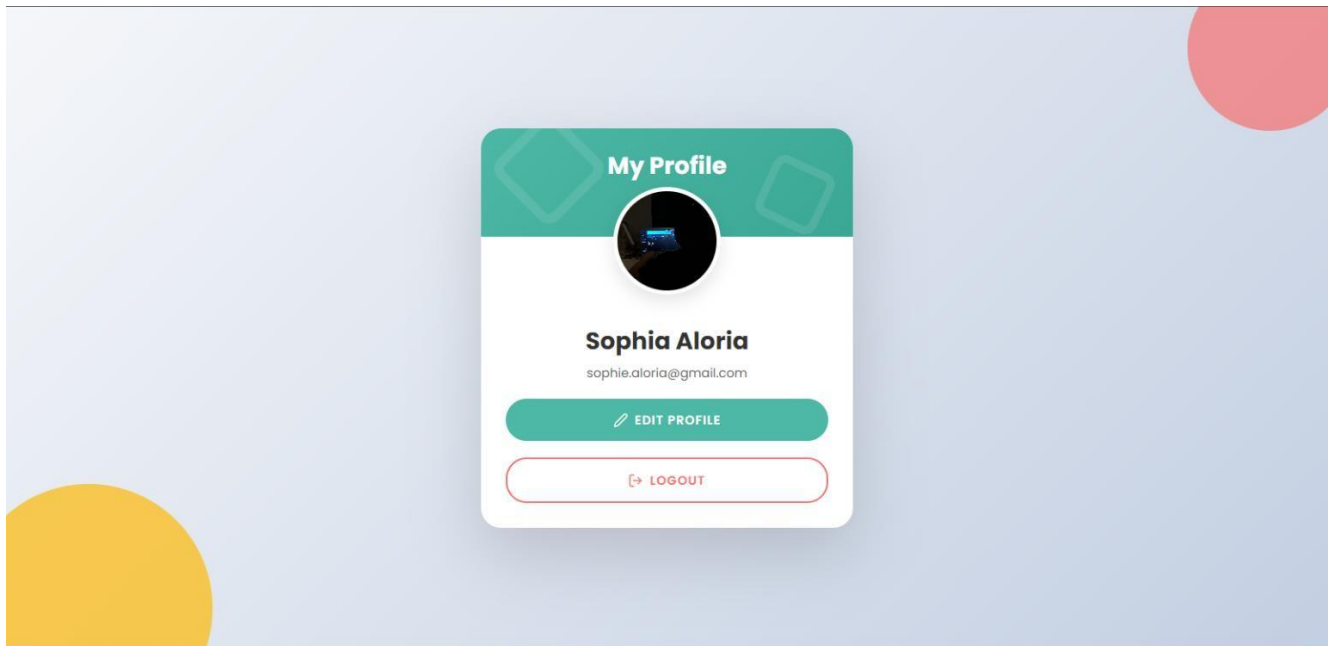
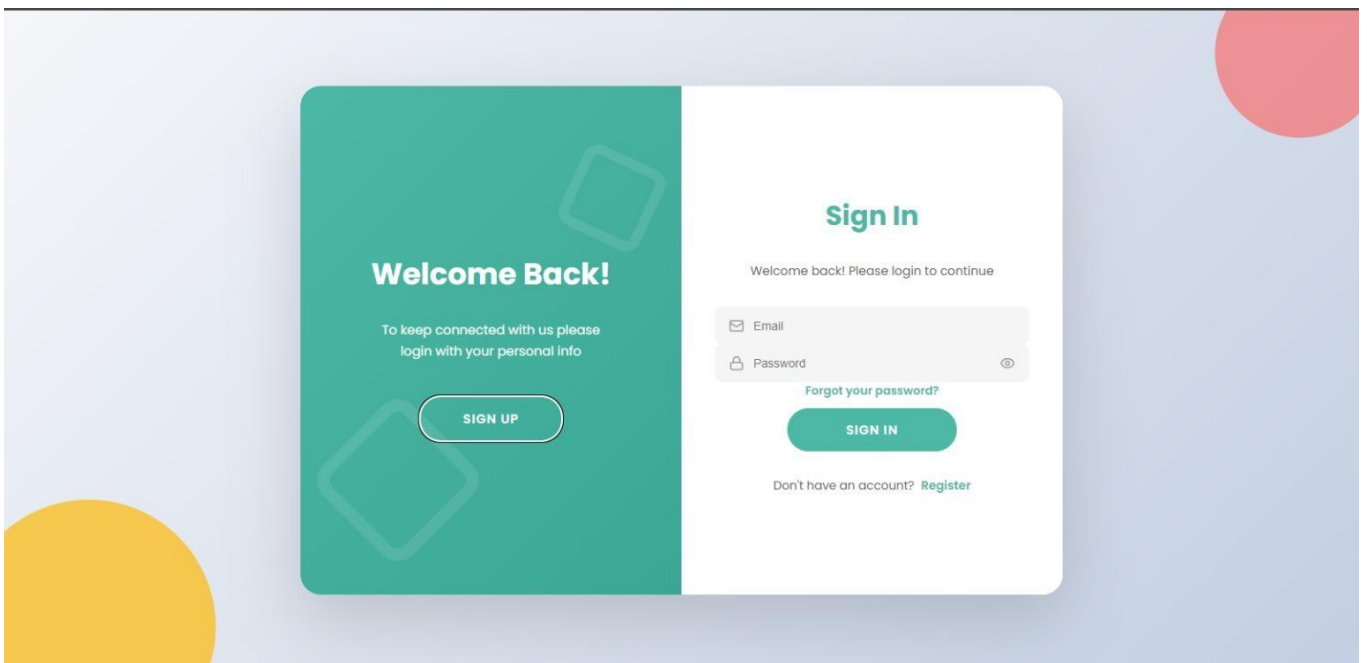


2.5. Sequence Diagram



Screenshots of the WEB UI





Screenshots of the Mobile App

9:24

Create Account

Enter your full name to get started

Full Name (e.g., John Doe)

Sophia Aloria

Enter your full name (first and last name)

Email

sophia.aloria@gmail.com

Password (min 8 chars, include numbers)

sophia256

SIGN UP

Already have an Account? Click Here!

9:26

Sign In

Welcome back! Please login to continue

Email

sophia.aloria@gmail.com

Password

sophia256


Forgot your password?

SIGN IN

Don't have an account? Register

9:25

My Profile



First Name

Sophia

Last Name (optional)

Aloria


Email

sophia.aloria@gmail.com

SAVE CANCEL

9:25

My Profile



Sophia Aloria

sophia.aloria@gmail.com

EDIT PROFILE

LOGOUT

3. Appendices

This section contains supporting materials and references used in the development of the User Authentication System.

A. References

- Spring Boot Official Documentation – <https://spring.io/projects/spring-boot>
- ReactJS Documentation – <https://react.dev>
- JSON Web Token (JWT) Introduction – <https://jwt.io/introduction>
- REST API Design Best Practices – <https://restfulapi.net>
- OWASP Authentication Guidelines – <https://owasp.org>

B. Supporting Diagrams

- Use Case Diagram – User Account Management
- Class Diagram – Authentication System Structure
- Sequence Diagram – User Authentication Flow
- ERD – User Database Structure

C. Tools and Technologies

- Frontend: ReactJS
- Backend: Spring Boot
- Database: MySQL / PostgreSQL
- Version Control: Git & GitHub
- API Testing Tool: Postman