CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-G1
SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: User Registration and Authentication

Prepared By: Abadinas, Treasure Louise S.

Date of Submission: 02/07/2026

Version: 2

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to provide a detailed description of the functional and non-functional requirements for the User Registration and Authentication system. This document is intended for project stakeholders, developers, and quality assurance testers to ensure a shared understanding of the system's core identity management features.

1.2. Scope

The system provides a secure gateway for users to access protected application resources. Its boundaries include:

- User Registration: Allowing new guests to create accounts by providing names, emails, and passwords.
- Authentication: Verifying credentials to grant system access.
- Exclusions: This system does not include password recovery (forgot password) or third-party OAuth (e.g., Google/Facebook login) unless specified in future versions.

1.3. Definitions, Acronyms, and Abbreviations

- User – A person who registers and accesses the system
- Authentication – Process of verifying a user’s identity
- Registration – Creating a new user account
- Credentials – User login details such as username and password
- FRS – Functional Requirements Specification
- SRS – Software Requirements Specification

2. Overall Description

2.1. System Perspective

This system acts as the primary security layer and entry point for a larger application ecosystem. It interacts directly with a database to store and retrieve user credentials and integrates with the application's landing page to manage user traffic based on authentication status.

2.2. User Classes and Characteristics

1. **Guest User:** Individuals seeking access. They are expected to have basic web navigation skills and a valid email address.
2. **Authenticated User:** Individuals who have successfully registered. They require the ability to view their specific profile and securely terminate their session.

2.3. Operating Environment

1. **Client Side:** Modern web browsers (Chrome, Firefox, Safari, Edge).
2. **Server Side:** Web server capable of executing validation logic.
3. **Database:** A relational database management system (RDBMS) to host the User table.

2.4. Assumptions and Dependencies

1. **Assumption:** Users have access to a stable internet connection.
2. **Dependency:** The system depends on a functioning database to persist user data during the registration "Saves data" step.

3. System Features and Functional Requirements

3.1. Feature 1: User Registration

Description: Allows unauthenticated guests to create a new identity within the system.
Functional Requirements:

- The system shall provide a registration form requesting name, email, and password.

- The system shall validate that the email is not already in use.
- The system shall save the new user record to the database upon successful validation.

3.2. Feature 2: User Authentication (Login)

Description: Verifies the identity of returning users to grant them access to the "Home" display.
Functional Requirements:

- The system shall allow users to input their email (username) and password.

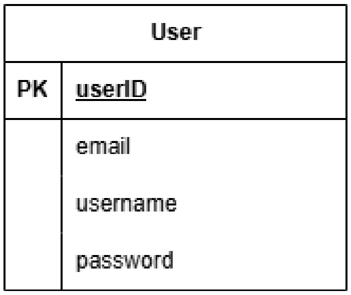
- The system shall verify the credentials against the stored database records.
- Upon successful authentication, the system shall redirect the user to the application's Home page.
- The system shall provide a Logout function to end the active session.

4. Non-Functional Requirements

- **Security:** All passwords must be hashed before being saved to the database.
- **Performance:** Credential validation and login redirection should occur within less than 2 seconds.
- **Usability:** The "Create account form" must be mobile-responsive and include clear error messages for invalid inputs.
- **Availability:** The authentication service should be available 99.9% of the time to ensure users can always access the system.

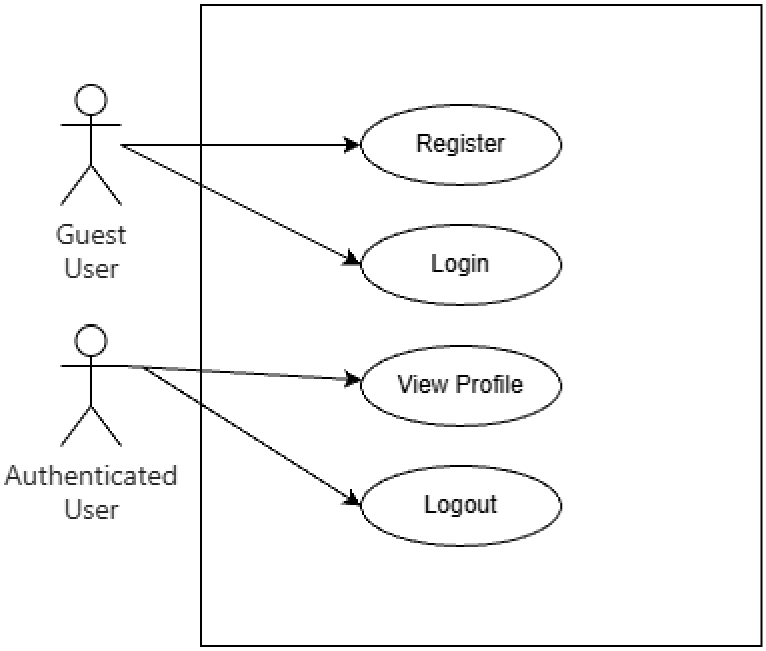
5. System Models (Diagrams)

5.1. ERD

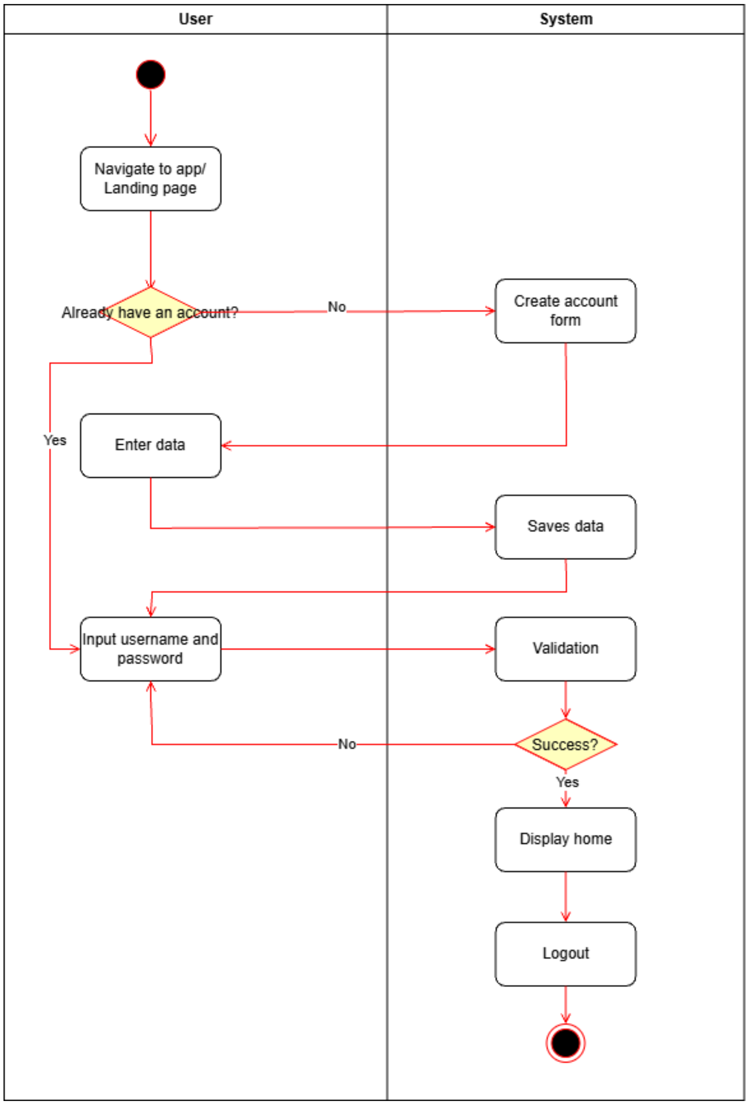


User	
PK	userID
	email
	username
	password

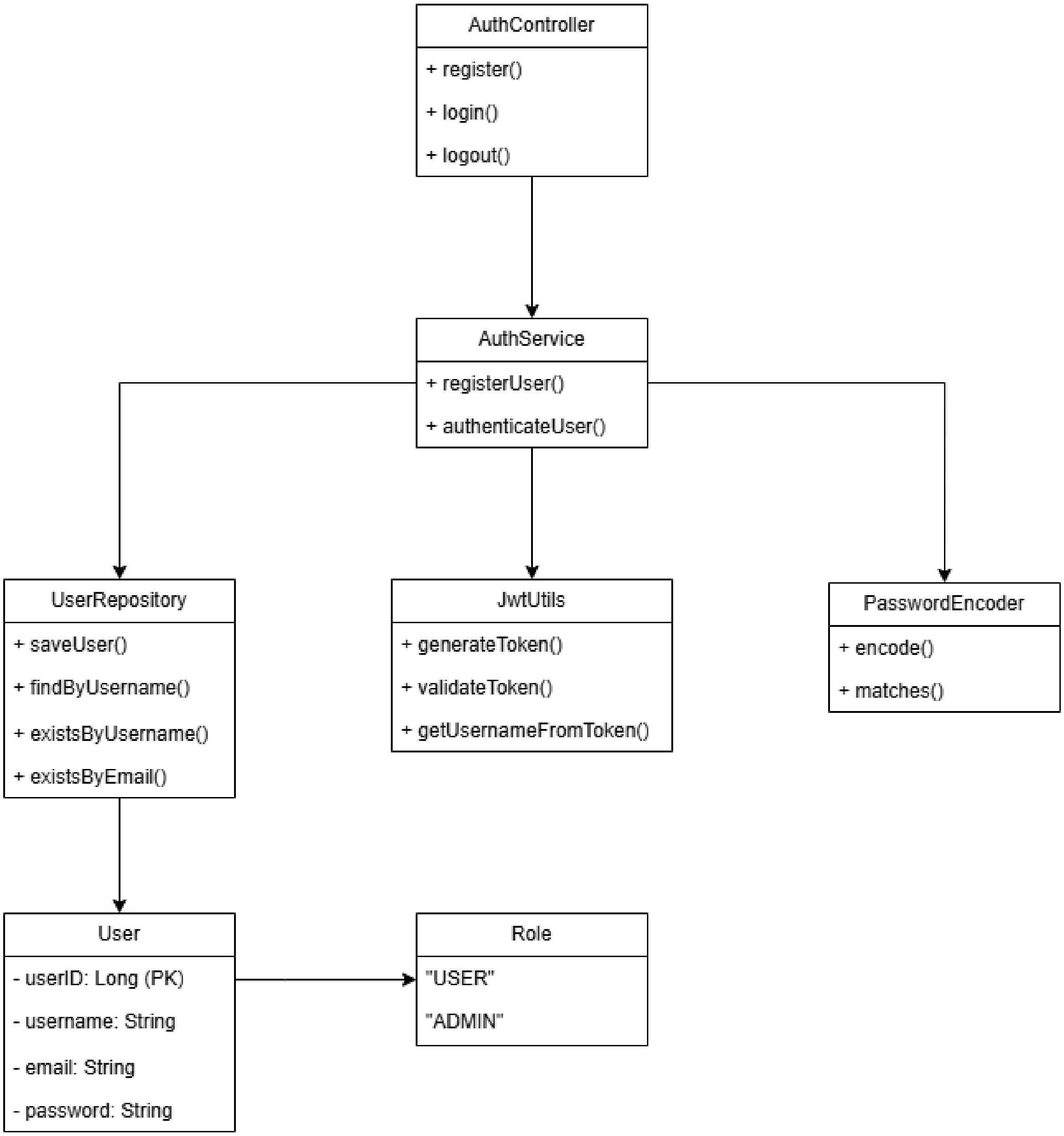
5.2. Use Case Diagram



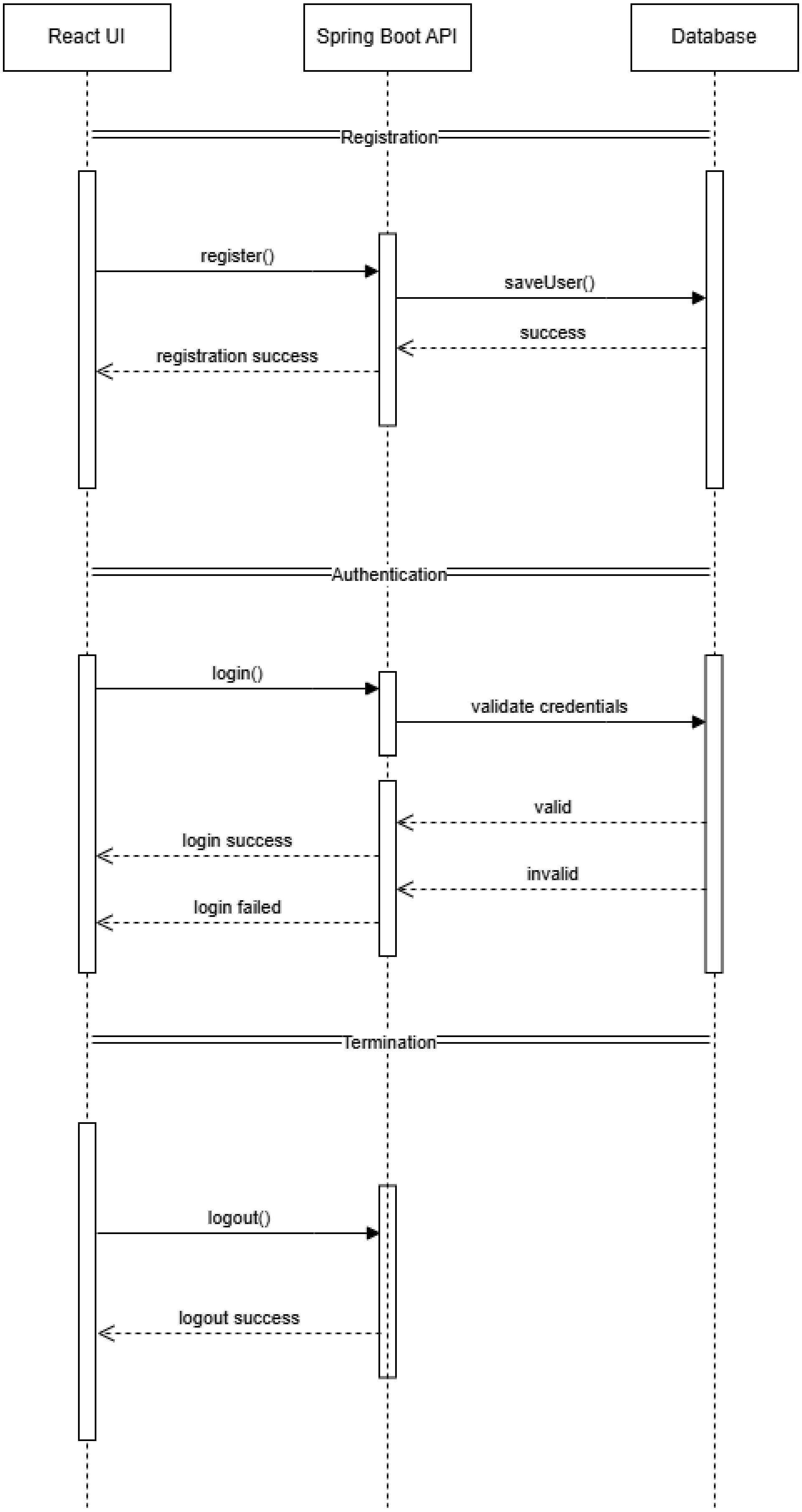
5.3. Activity Diagram



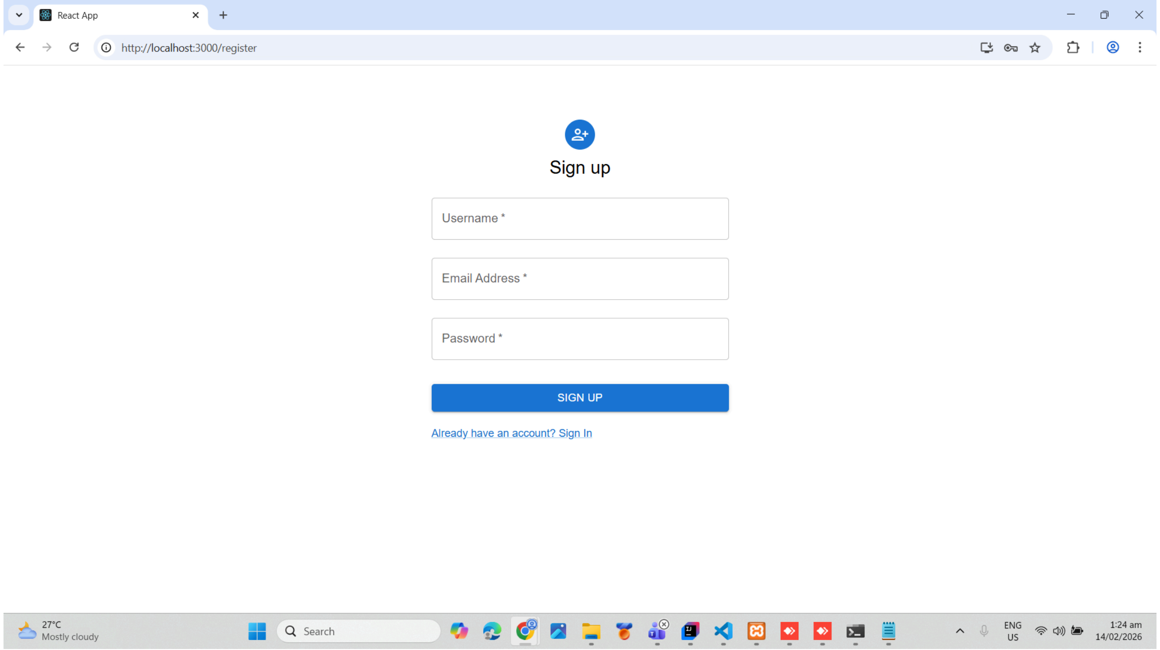
5.4. Class Diagram

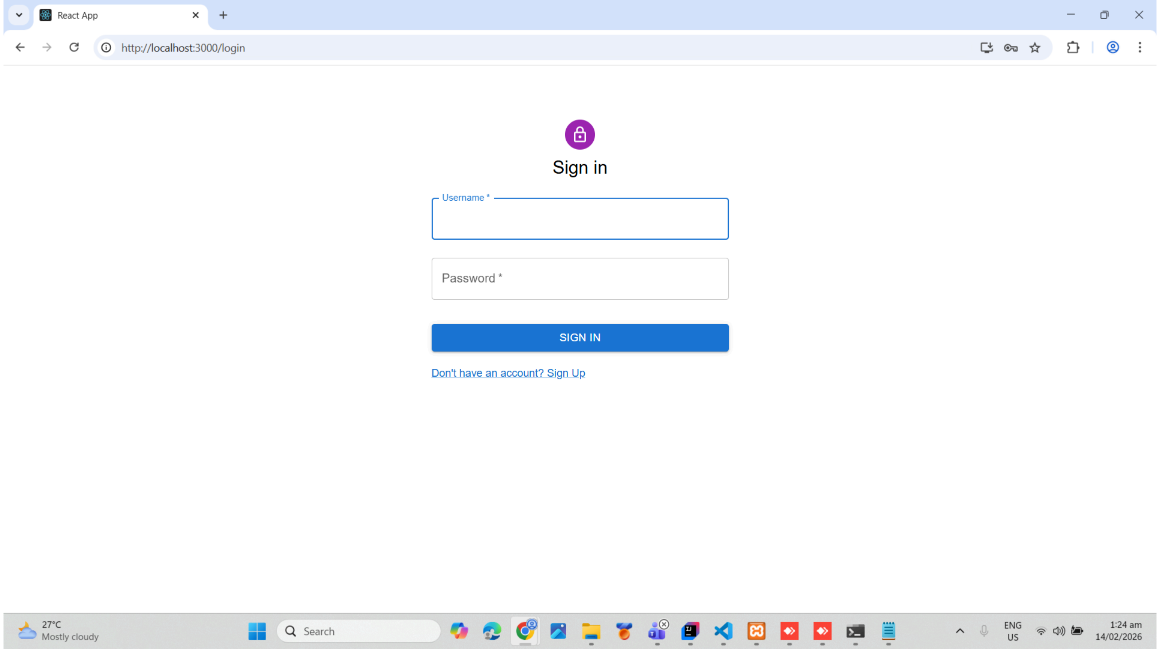


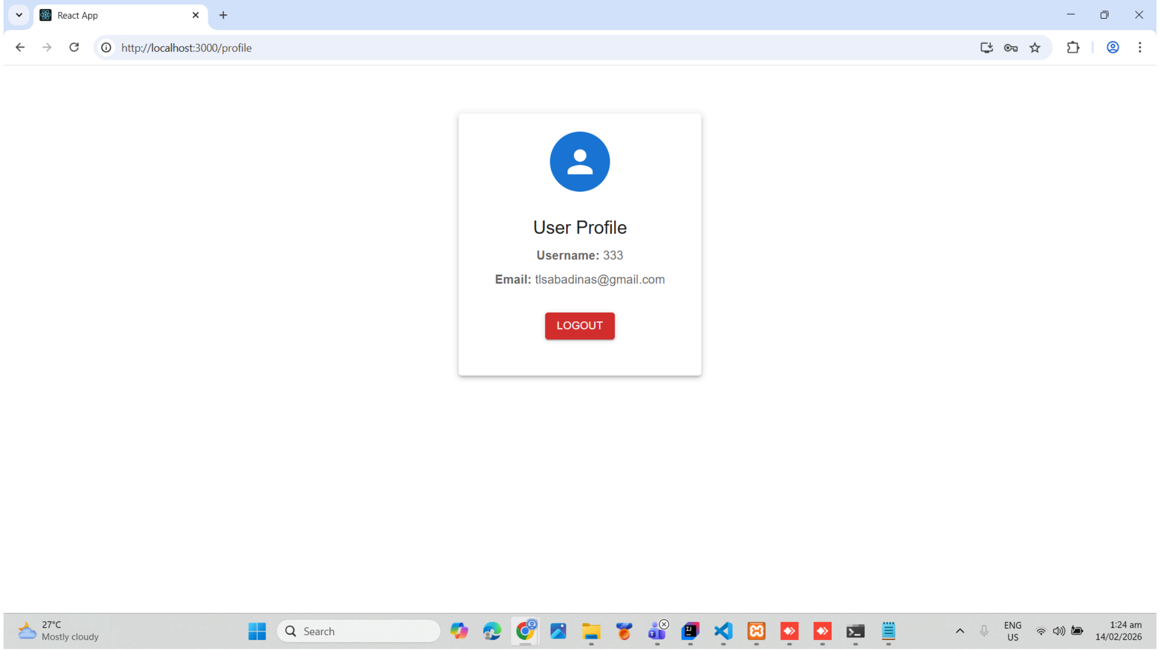
5.5. Sequence Diagram



5.6. Screenshots of the Web UI


React App
http://localhost:3000/register
Sign up
Username *
Email Address *
Password *
SIGN UP
Already have an account? Sign In
27°C
Mostly cloudy
Search
ENG
US
1:24 am
14/02/2026


React App
http://localhost:3000/login
Sign in
Username *
Password *
SIGN IN
Don't have an account? Sign Up
27°C
Mostly cloudy
Search
ENG
US
1:24 am
14/02/2026



User Profile

Username: 333

Email: tlsabadinas@gmail.com

LOGOUT

6. Appendices

- Technological References: Developed using ReactJS for the UI, Spring Boot for the API, and MySQL for data storage.
- Tools: All diagrams were created using draw.io / diagrams.net.
- External Services: Assumes access to internet connectivity and correctly configured backend/database services.