CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-Section SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: Mini-App User Registration & Authentication

Prepared By: Sharbelle Temperatura

Date of Submission: February 3, 2026

Version:

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to define the functional and non-functional requirements of the Mini-App User Registration & Authentication System. This document is intended for instructors, developers, and students who will design, implement, and evaluate the system.

1.2. Scope

The purpose of this document is to define the functional and non-functional requirements of the Mini-App User Registration & Authentication System. This document is intended for instructors, developers, and students who will design, implement, and evaluate the system.

1.3. Definitions, Acronyms, and Abbreviations

Term	Definition
FRS	Functional Requirements Specification
UI	User Interface
API	Application Programming Interface
ERD	Entity Relationship Diagram
UML	Unified Modeling Language
Authentication	Process of verifying user identity

2. Overall Description

2.1. System Perspective

The system is a standalone mini web application that follows a client-server architecture. The frontend communicates with a backend service that handles authentication logic and database operations.

2.2. User Classes and Characteristics

User Class	Description
Guest User	Unregistered or logged-out user who can register or log in

Authenticated User	Logged-in user who can view their profile and log out

2.3. Operating Environment

- **Frontend:** Web browser (Chrome, Edge, Firefox)
- **Backend:** Spring Boot application
- **Database:** Relational Database (MySQL or similar)
- **Tools:** draw.io / diagrams.net, IDE (IntelliJ / VS Code)

2.4. Assumptions and Dependencies

- Users have access to the internet
- The database server is available
- The backend API is operational
- Authentication uses encrypted passwords

3. System Features and Functional Requirements

3.1. Feature 1: User Registration

Description: Allows a guest user to create a new account by providing required information.

Functional Requirements:

- The system shall allow a guest user to register using a username, email, full name, and password
- The system shall validate user input before saving
- The system shall store the password in encrypted form
- The system shall prevent duplicate usernames or emails

3.2. Feature 2: User Login

Description: Allows a registered user to log in using valid credentials.
Functional Requirements:
- The system shall authenticate users using username and password
- The system shall deny access for invalid credentials
- The system shall create a session or token upon successful login

3.3. Feature 3: View Profile

Description: Allows authenticated users to view their profile information.
Functional Requirements:
- The system shall allow only authenticated users to access the profile page
- The system shall display the user's stored information
- The system shall redirect unauthenticated users to the login page

3.4. Feature 4: User Logout

Description: Allows authenticated users to log out of the system.
Functional Requirements:
- The system shall invalidate the user session or token
- The system shall redirect the user to the login page after logout

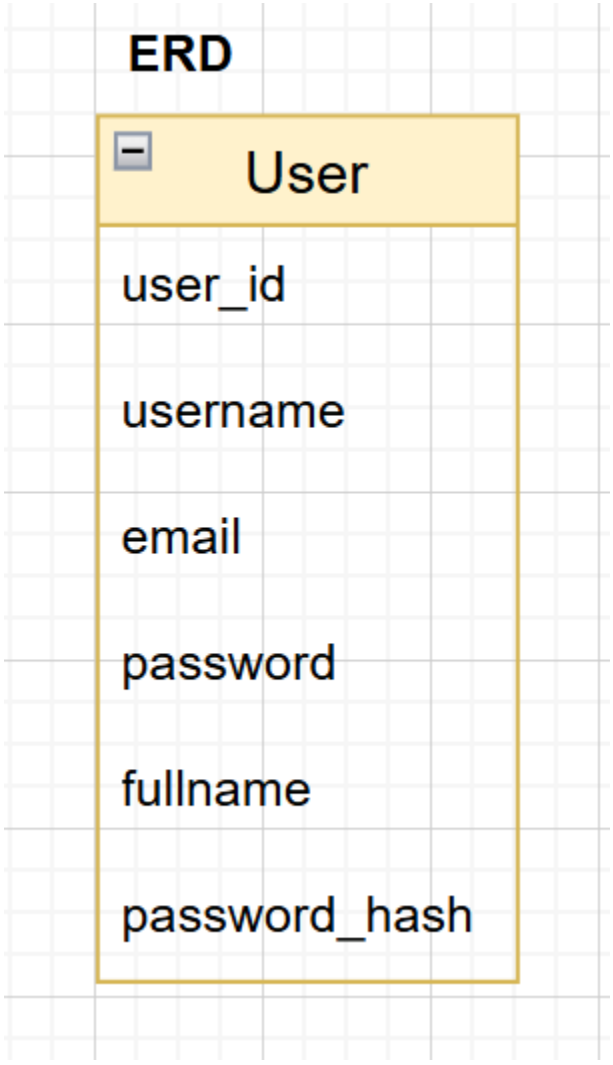
4. Non-Functional Requirements

- **Security:** Passwords must be encrypted
- **Usability:** The interface should be simple and easy to use
- **Performance:** Login and registration should complete within acceptable time
- **Reliability:** The system should prevent unauthorized access

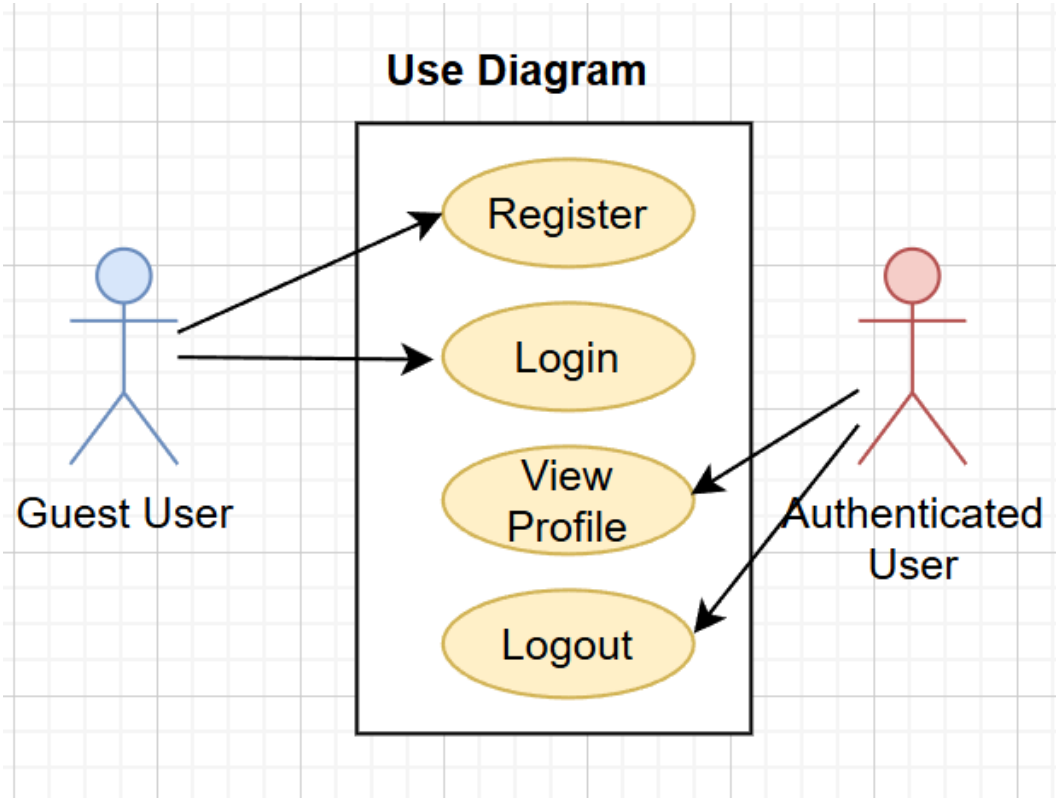
5. System Models (Diagrams)

Insert the necessary diagrams for the system:

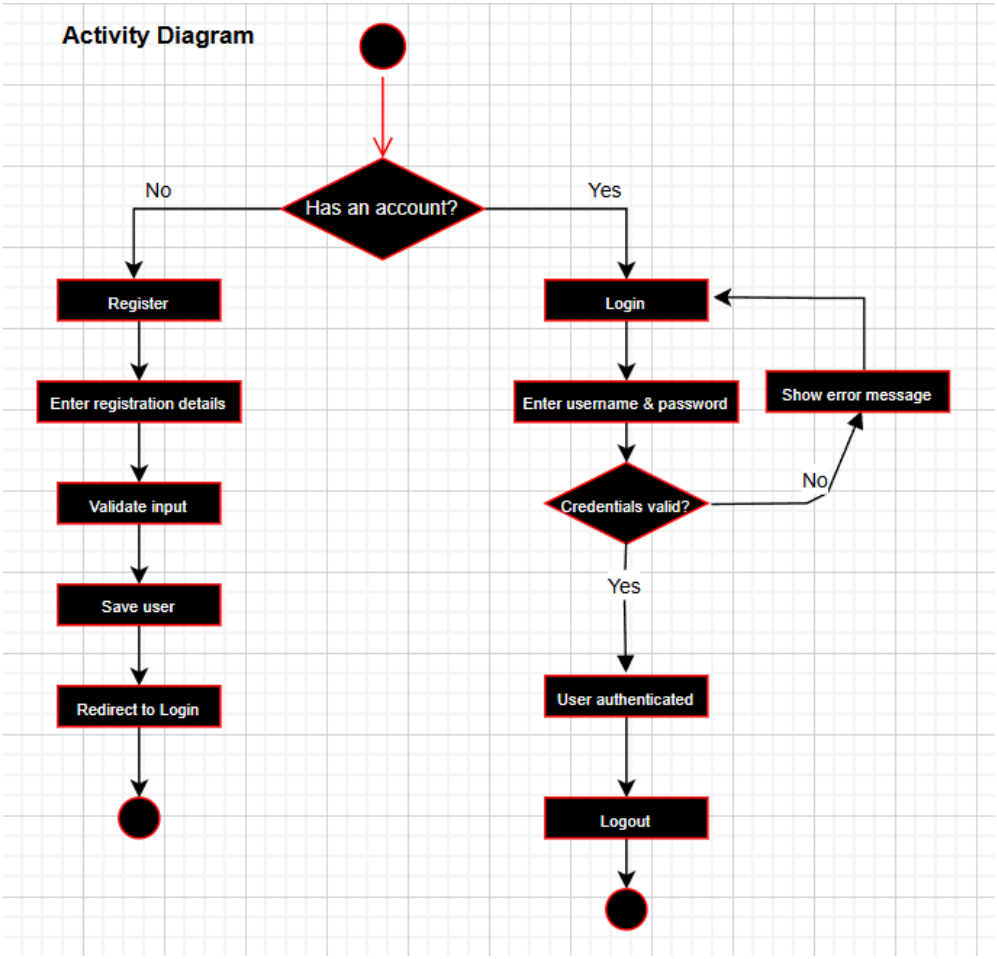
5.1. ERD



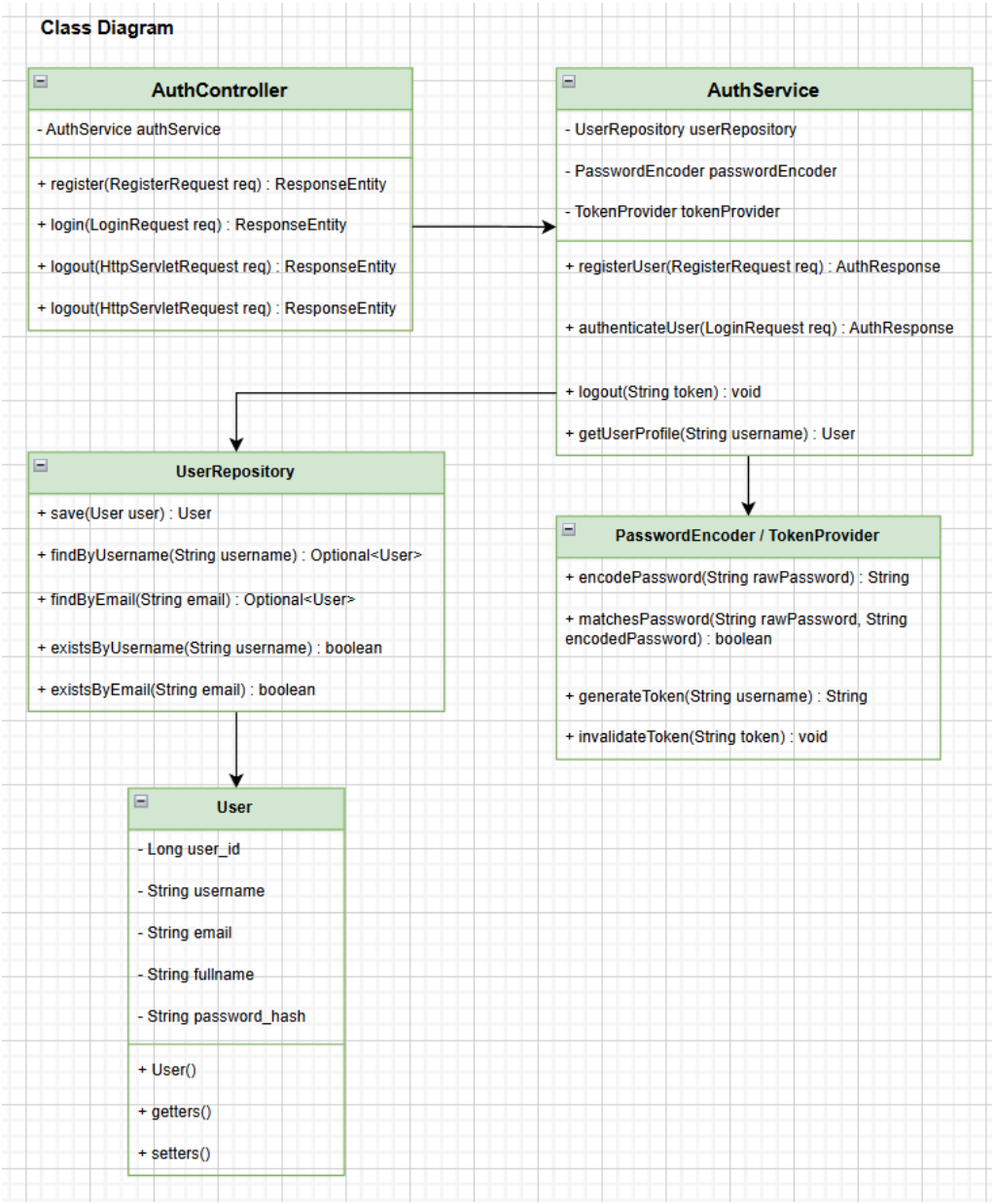
5.2. Use Case Diagram



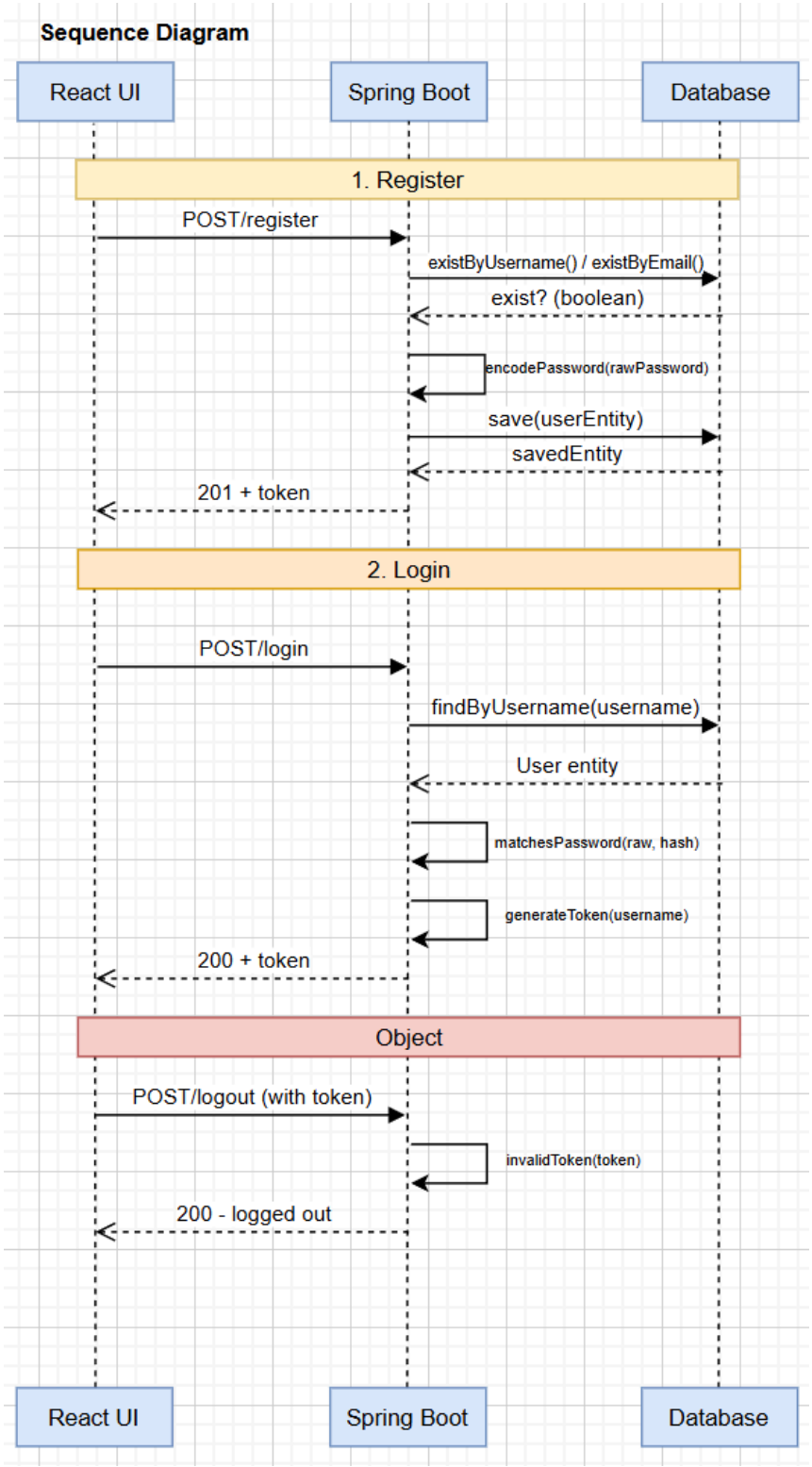
5.3. Activity Diagram



5.4. Class Diagram



5.5. Sequence Diagram



6. Appendices

This appendix contains the UML diagrams used to model and document the Mini-App User Registration and Authentication system. These diagrams provide a visual representation of the system's structure and behavior, including the Entity Relationship Diagram (ERD), Use Case Diagram, Activity Diagram, Class Diagram, and Sequence Diagram.