IT342-G4
SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: Mini App – User Registration & Authentication

Prepared By: Junjie L. Geraldez

Date of Submission: February 11, 2026

Version: 1

Table of Contents

1. Introduction

1.1. Purpose
The purpose of this system is to allow users to register, log in, access protected pages, and log out securely. This document is intended for students and developers as a guide to understanding the system design and process before implementation.

1.2. Scope
The system covers user registration, login, logout, and access control to protected pages such as the user dashboard or profile. It includes basic user data storage, authentication validation, and session or token management, but excludes advanced features like user role management, password recovery, and administrative functions.

1.3. Definitions, Acronyms, and Abbreviations
- User – A person who interacts with the system by registering, logging in, or accessing protected pages.
- Authentication – The process of verifying a user's identity using login credentials.
- Authorization – The process of determining whether a user is allowed to access protected resources.
- Guest User – A user who is not logged in and has limited access to the system.
- Authenticated User – A user who has successfully logged in and can access protected pages.
- UI (User Interface) – The frontend part of the system where users interact (e.g., React UI).
- API (Application Programming Interface) – The backend service (Spring Boot) that handles requests from the frontend.
- Token – A secure string used to maintain user session and authentication state.
- Database (DB) – The storage system that saves user account information.

2. Overall Description

2.1. System Perspective
The system is a web-based authentication module that works with a React frontend, Spring Boot backend, and a database. It manages user login, registration, and access to protected pages within a larger application.

2.2. User Classes and Characteristics

- Guest User – Not logged in; can register or log in; cannot access protected pages.
- Authenticated User – Logged in; can view dashboard/profile; can log out; has access to protected pages.

2.3. Assumptions and Dependencies

- Users have a device with internet access.
- The frontend (React) and backend (Spring Boot) are running and connected.
- Database is available and operational.
- Users provide valid registration and login information.
- Token/session management is supported by the system.

3. System Features and Functional Requirements

- **User Registration** o Users can create an account with a username, email, and password.
  - o System validates input and stores encrypted passwords in the database.
- **User Login** o Users can log in using email/username and password.
  - o System verifies credentials and creates a session or token.
- **View Profile / Dashboard** o Logged-in users can access their profile or dashboard.
  - o Protected pages are blocked for non-authenticated users.
- **User Logout**
  - o Users can log out, ending their session or invalidating the token. o After logout, access to protected pages is denied.

3.1. Feature 1: User Registration

Description: Allows a new user to create an account by providing a username, email, and password.
Functional Requirements:

- System must validate that all required fields are filled.
- Email and username must be unique.
- Password must be encrypted before storing in the database.
- Upon successful registration, the user can log in.

3.2. Feature 2:

Description: Allows registered users to access the system by entering their email/username and password.
Functional Requirements:
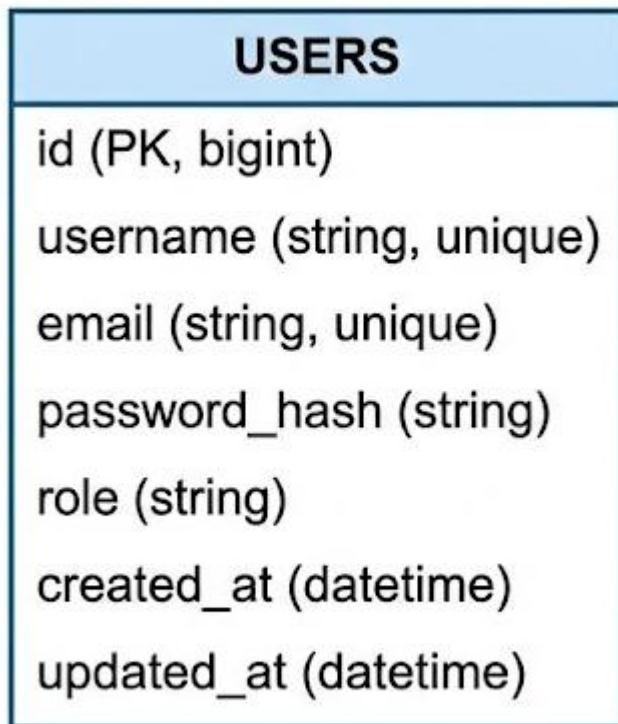
- System must verify user credentials.
- Successful login creates a session or token.
- Invalid credentials show an error message.
- Logged-in users can access protected pages
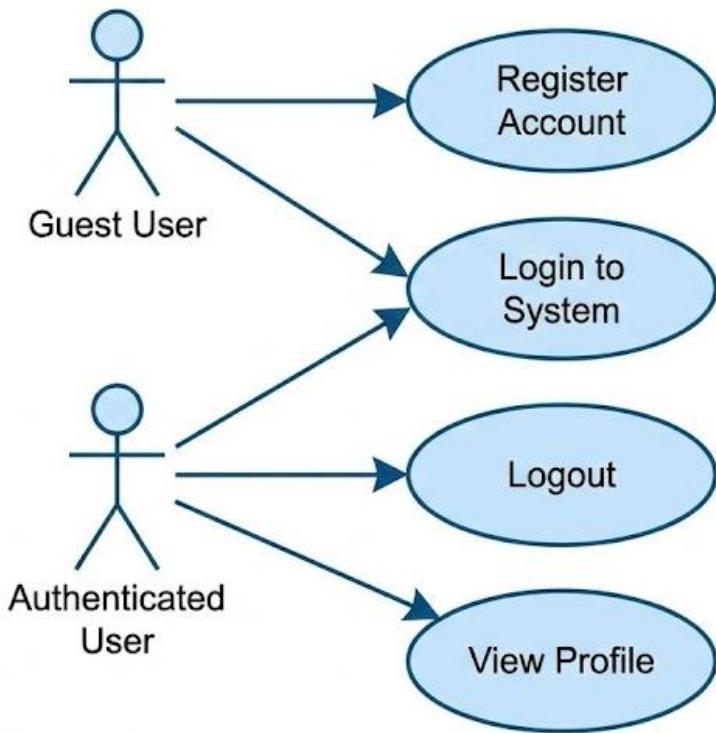
4. System Models (Diagrams)

*Insert the necessary diagrams for the system:*

4.1. ERD

| USERS |
| --- |
| id (PK, bigint) |
| username (string, unique) |
| email (string, unique) |
| password_hash (string) |
| role (string) |
| created_at (datetime) |
| updated_at (datetime) |
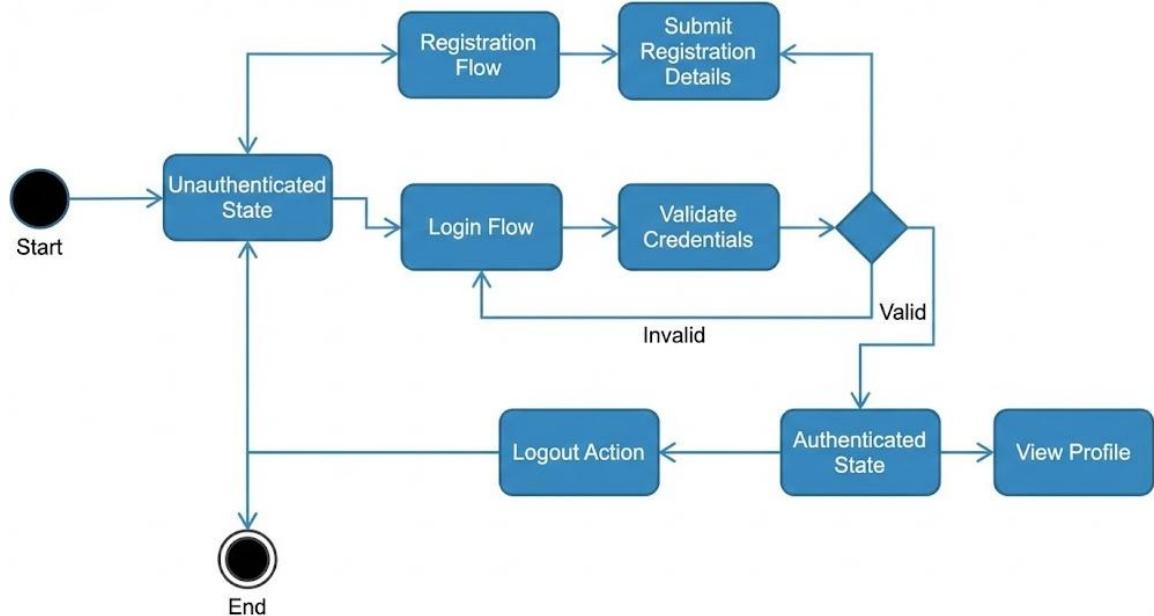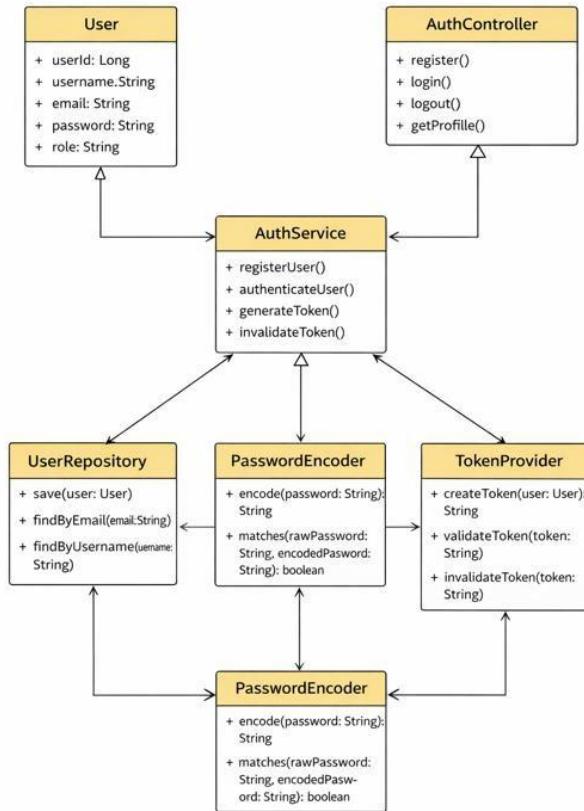
4.2. Use Case Diagram
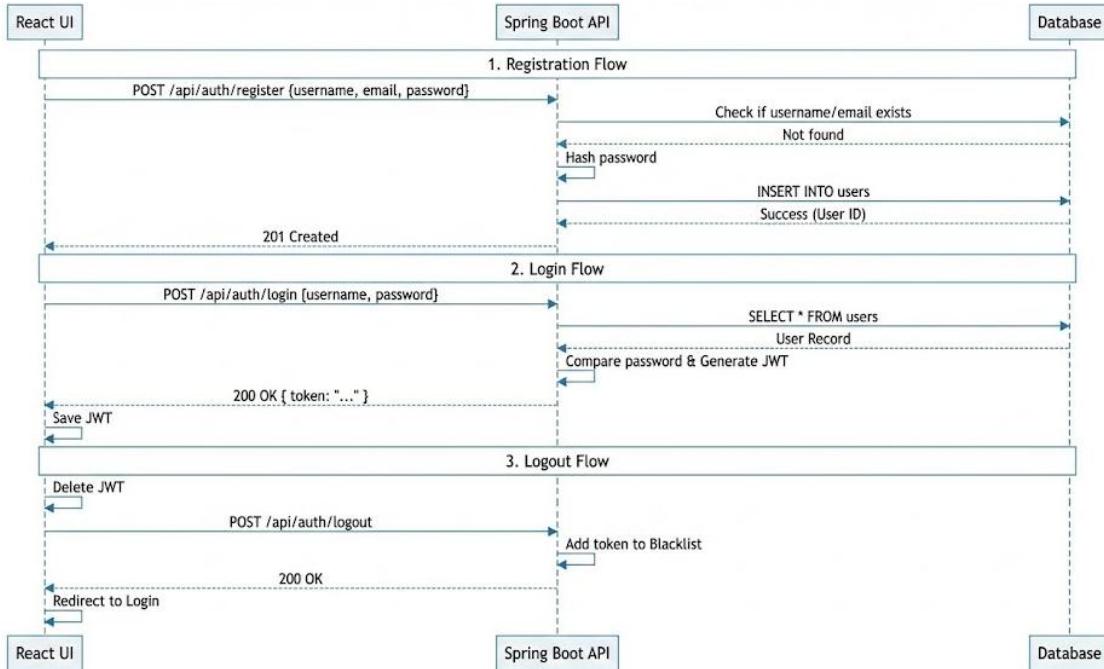
4.3. Activity Diagram
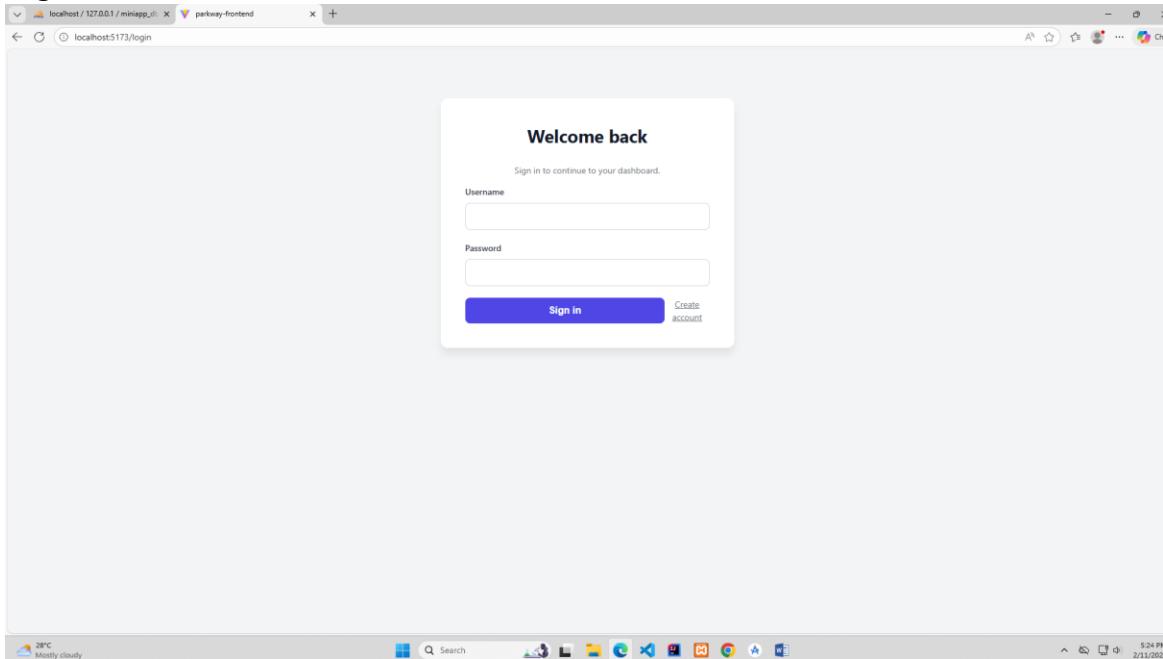
4.4. Class Diagram
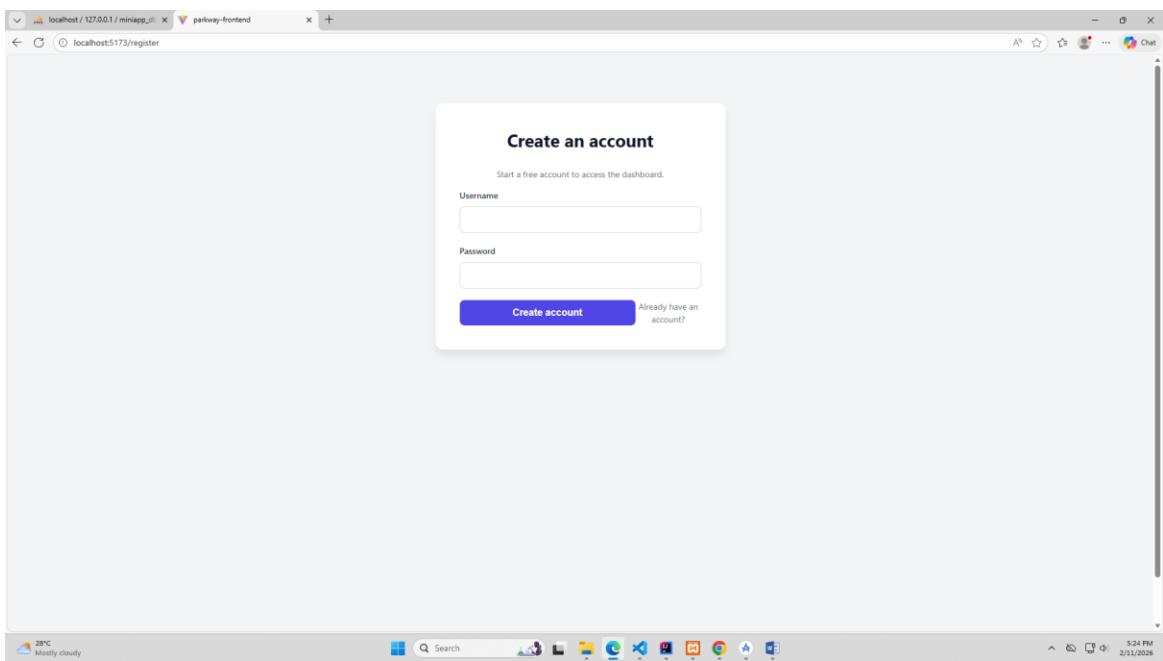
4.5. Sequence Diagram

Login

Registration

Dashboard

Profile

LOGOUT

DATABASE MYSQL