CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-G4
SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: User Registration & Authentication

Prepared By: Pranz Nicole C. Rabe

Date of Submission: 02/03/2026

Version: 1.0

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to describe the design and system flow of a User Registration and Authentication system. It outlines the system's structure, features, and processes using diagrams and written explanations. This document is intended for students, instructors, and developers who want to understand how user authentication works.

1.2. Scope

The system allows users to create an account, log in using their credentials, access a protected profile or dashboard, and log out securely. The scope of this project focuses only on system design and documentation, including database structure, system processes, and interactions.

1.3. Definitions, Acronyms, and Abbreviations

ERD - Entity Relationship Diagram
UI – User Interface
API - Application Programming Interface
Authentication - Process of verifying user identity
JWT - JSON Web Token (used for login sessions)

2. Overall Description

2.1. System Perspective

The User Registration and Authentication system is a web-based system composed of a frontend interface and a backend server connected to a database. The frontend (React UI) allows users to interact with the system, while the backend (Spring Boot API) handles business logic, authentication, and data storage in the database.

2.2. User Classes and Characteristics

Guest User - A user who has not logged in. Can register an account and access the login page.

Authenticated User - A logged-in user who can access protected pages such as the profile or dashboard and can log out.

2.3. Operating Environment

The system is designed to operate in the following environment:

- Frontend: React.js (web browser-based interface)
- Backend: Spring Boot (Java-based REST API)
- Database: Relational database such as MySQL or PostgreSQL

.

2.4. Assumptions and Dependencies

- Users have access to the internet and a supported web browser
- The backend server is running and connected to the database
- Email addresses are unique per user
- Authentication tokens or sessions are securely managed

3. System Features and Functional Requirements

3.1. Feature 1: User Registration

Description: Allows a guest user to create a new account by providing necessary information such as username, email, and password.
Functional Requirements:
- The system shall allow users to enter registration details
- The system shall validate user input
- The system shall store user information securely in the database

3.2. Feature 2: User Login and Logout

Description: Allows registered users to log in using their credentials, access protected pages, and log out securely.
Functional Requirements:
- The system shall authenticate users using email and password
- The system shall grant access to the dashboard upon successful login
- The system shall prevent access to protected pages when logged out

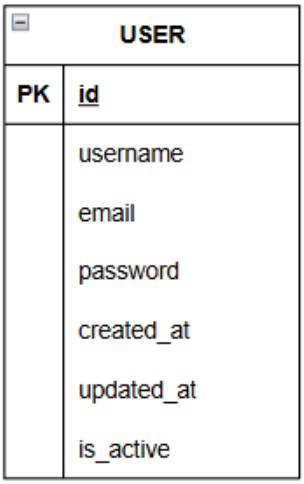
4. Non-Functional Requirements

- Security: Passwords must be encrypted, and unauthorized access must be prevented
- Performance: The system should respond quickly to login and registration requests
- Usability: The interface should be simple and easy to navigate
- Reliability: The system should handle errors gracefully and remain available

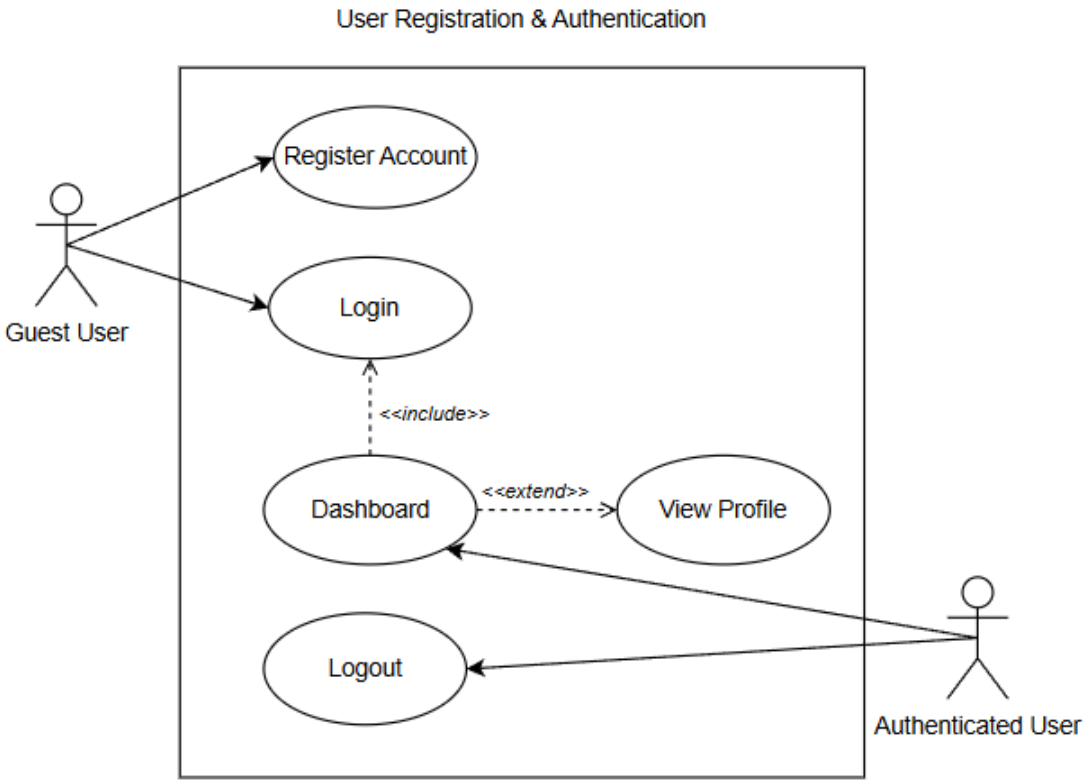
5. System Models (Diagrams)

Insert the necessary diagrams for the system:

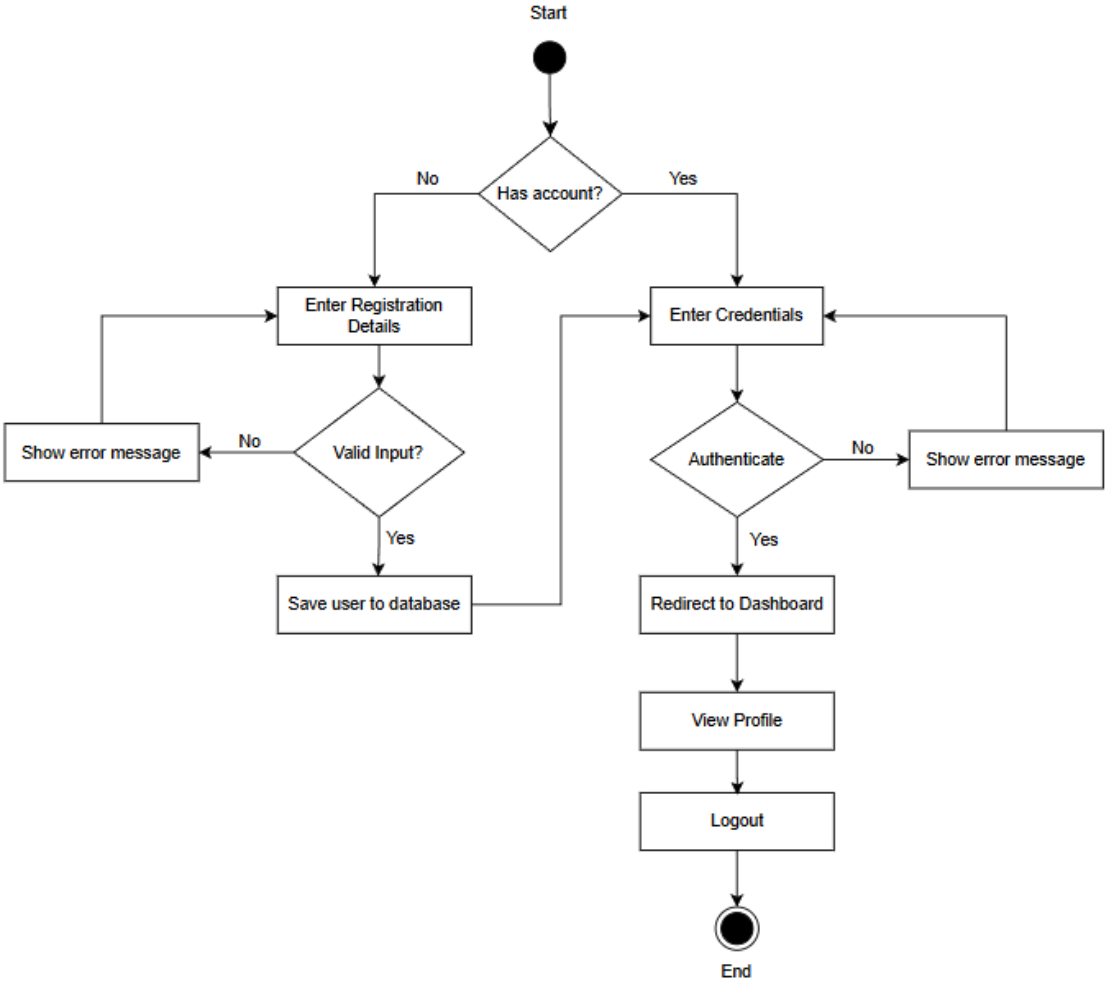
5.1. ERD



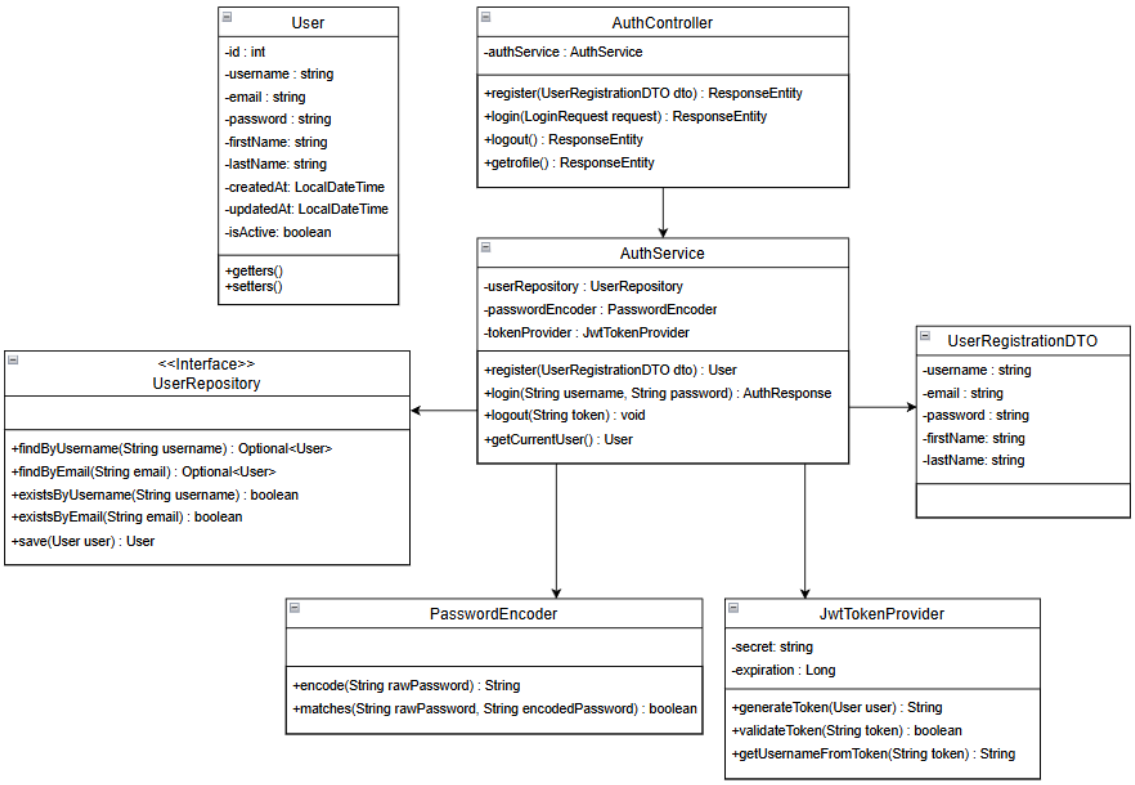
5.2. Use Case Diagram



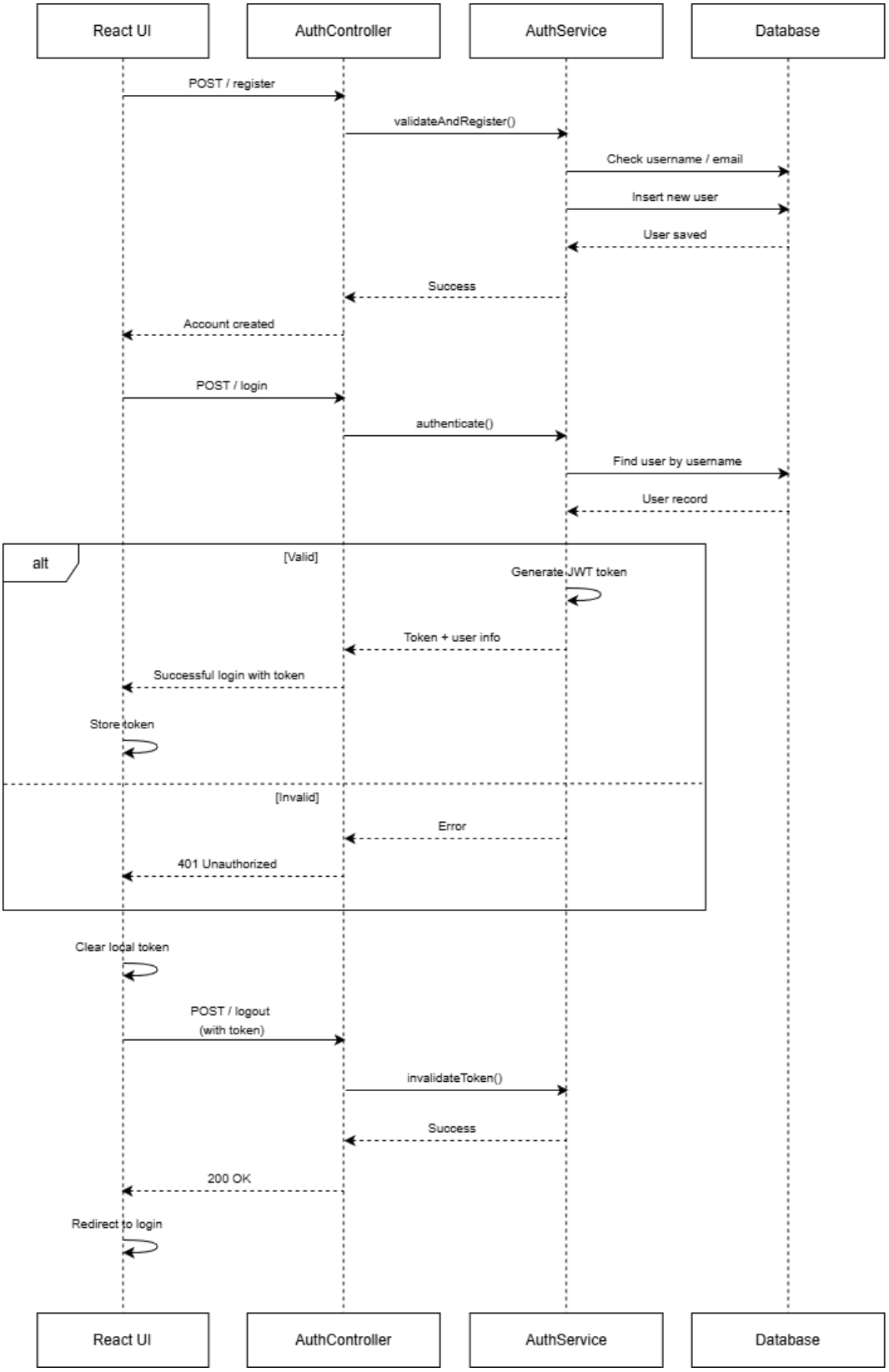
5.3. Activity Diagram



5.4. Class Diagram



5.5. Sequence Diagram



6. Appendices

References:

[1] Draw.io, “Sequence Diagrams – A Complete Guide,” https://www.drawio.com/blog/sequence-diagrams.

[2] GeeksforGeeks, “UML Activity Diagrams,” https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-activity-diagrams/.