CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-Section SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: User Registration and Authentication System

Prepared By: Frances Anne B. Riconalla

Date of Submission: February 2, 2026

Version: 1

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 document is to describe the functional requirements of the User Registration and Authentication System. It serves as a reference for developers, instructors, and stakeholders to understand the system's functionality, behavior, and design prior to implementation.

1.2. Scope

The system provides basic user authentication functionality, including:

- User registration
- User login
- Access to protected pages (profile/dashboard)
- User logout

The system does not include advanced features such as password recovery, email verification, or role-based administration. It focuses solely on authentication and access control.

1.3. Definitions, Acronyms, and Abbreviations

- Authentication – The process of verifying a user's identity.
- Guest User – A user who is not logged in.
- Authenticated User – A user who has successfully logged in.
- Token – A temporary credential used to maintain authenticated sessions.
- API – Application Programming Interface.

2. Overall Description

2.1. System Perspective

The User Registration and Authentication System operates as a backend service that supports a web-based frontend. It follows a client-server architecture where the frontend communicates with the backend through RESTful APIs, and the backend interacts with a relational database.

2.2. User Classes and Characteristics

- Guest User
    - Can register and log in
    - Cannot access protected pages
- Authenticated User
    - Has valid login credentials
    - Can access profile/dashboard
    - Can log out of the system

2.3. Operating Environment

- Frontend: Web browser (React-based UI)

- Backend: Spring Boot (Java)
- Database: MySQL
- Tools: IntelliJ IDEA, VS Code, Git, Postman

2.4. Assumptions and Dependencies

- Users have internet access.
- The database is available and reachable by the backend.
- Authentication tokens are securely generated and invalidated.
- The system assumes correct configuration of the backend environment.

3. System Features and Functional Requirements

3.1. Feature 1: User Registration

Description: Allows a guest user to create a new account in the system
Functional Requirements:

- The system shall allow users to submit registration details.
- The system shall validate registration data.
- The system shall store user information securely in the database.
- The system shall confirm successful registration.

3.2. Feature 2: User Login and Authentication

Description: Allows registered users to log in and access protected resources.
Functional Requirements:

- The system shall authenticate users using valid credentials.
- The system shall generate an authentication token upon successful login.
- The system shall allow authenticated users to access protected pages.
- The system shall deny access to protected pages for unauthenticated users.

3.3. Feature 3: User Logout

Description: Allows authenticated users to terminate their session.
Functional Requirements:

- The system shall allow authenticated users to log out.
- The system shall invalidate the user's authentication token.
- The system shall prevent further access to protected pages after logout

4. Non-Functional Requirements

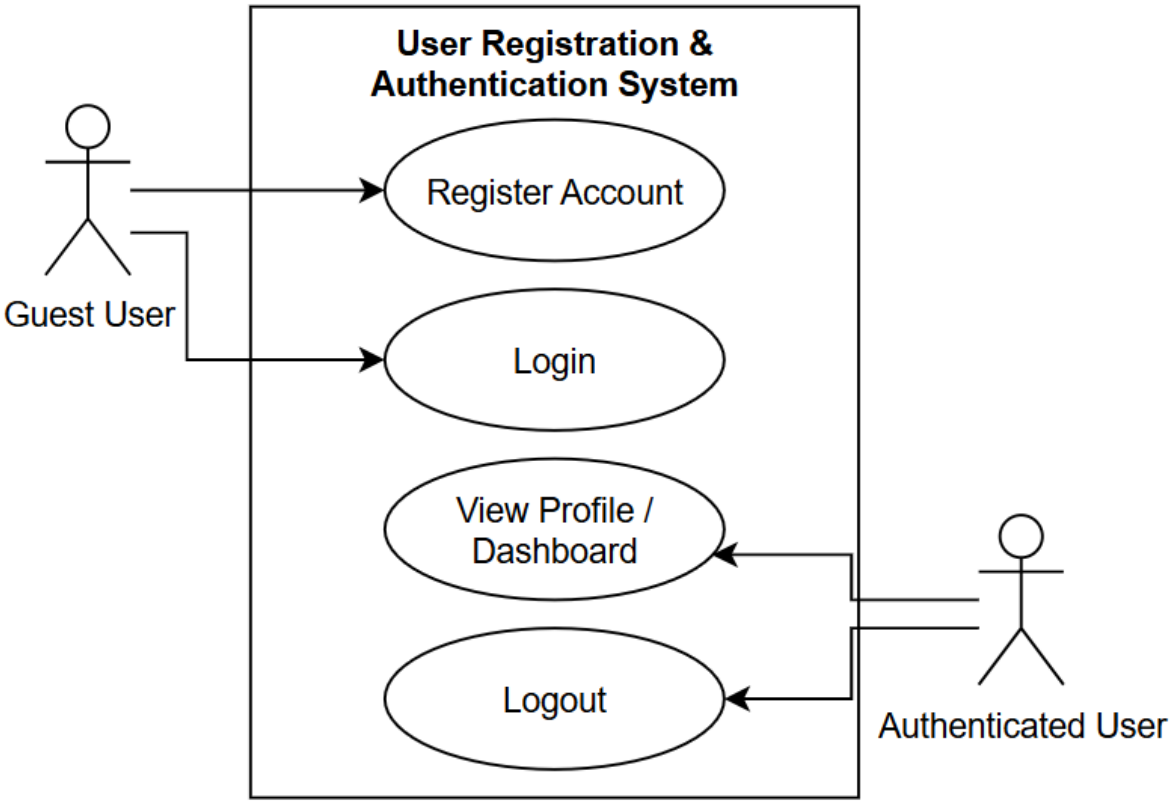
- **Security:** Passwords shall be stored in hashed form.
- **Performance:** Authentication requests should be processed within acceptable response time.
- **Usability:** The system should provide clear feedback for authentication actions.
- **Reliability:** The system should handle concurrent user requests without failure.
- **Maintainability:** The system design should support future enhancements.

5. System Models (Diagrams)

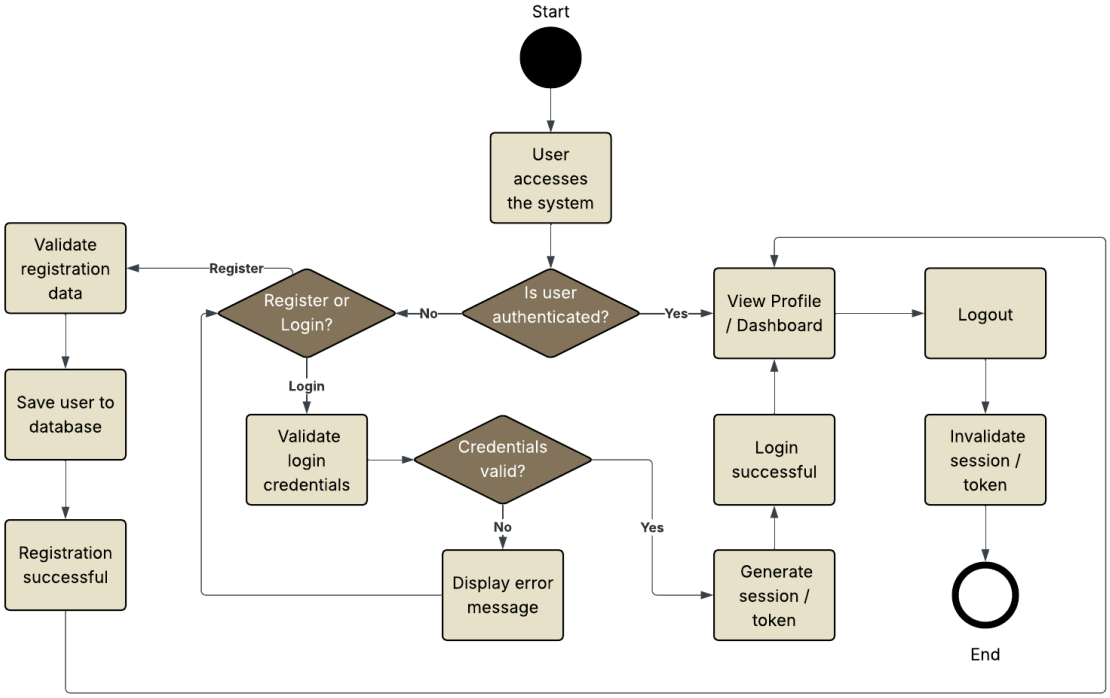
5.1. ERD

USERS		
PK	id	BIGINT
	username	VARCHAR(50) [UNIQUE]
	email	VARCHAR(100) [UNIQUE]
	password_hash	VARCHAR(255)
	role	VARCHAR(20)
	is_active	BOOLEAN
	created_at	TIMESTAMP
	updated_at	TIMESTAMP

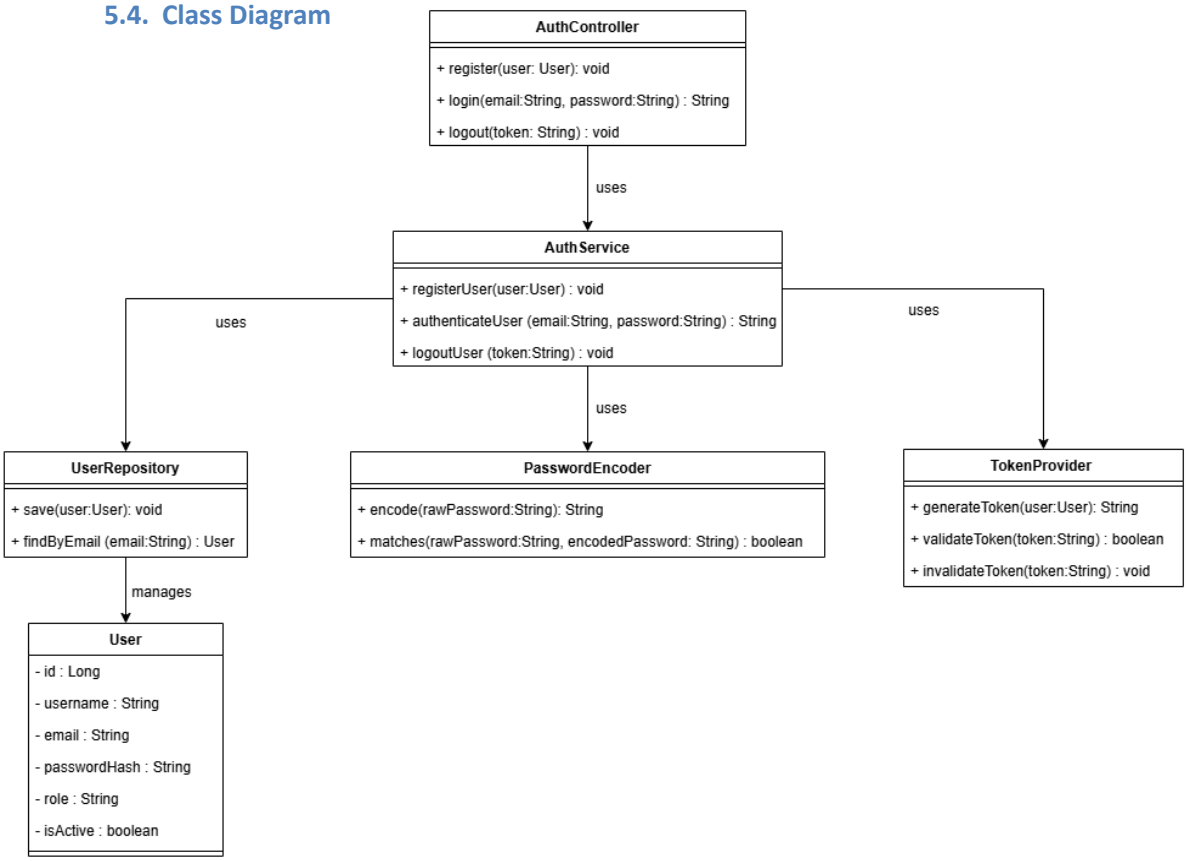
5.2. Use Case Diagram



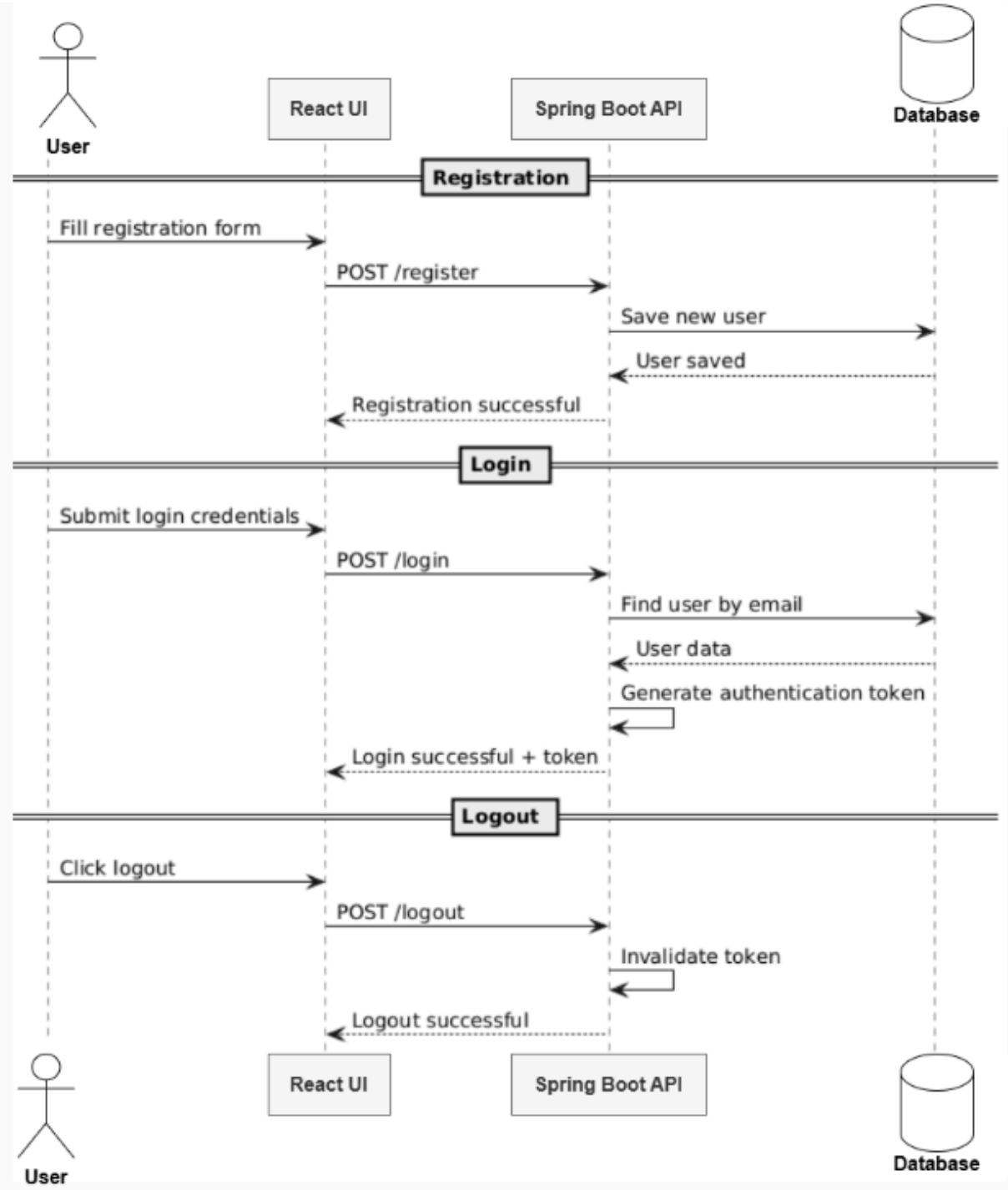
5.3. Activity Diagram



5.4. Class Diagram



5.5. Sequence Diagram

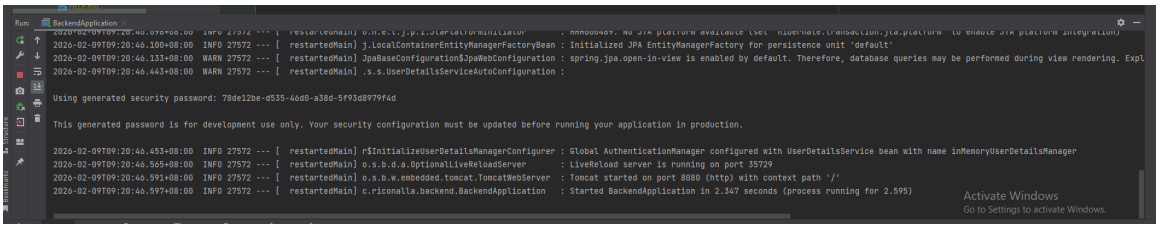


6. Appendices

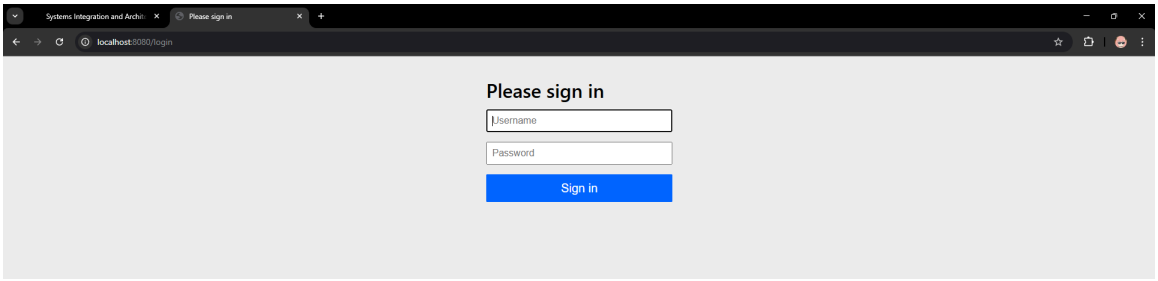
No additional appendices are required for this version of the system.

SCREENSHOTS:

Screenshot 1 — Backend Running



Screenshot 2 — Login Page



Screenshot 3 — Unauthorized Access