



CEBU INSTITUTE OF TECHNOLOGY
U N I V E R S I T Y

IT342-G4

SYSTEMS INTEGRATION AND

ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS

SPECIFICATION (FRS)

Project Title: User Authentication and Profile Management System

Prepared By: Ybañez, Liezel, A.

Date of Submission: February 02, 2026

Version: 1.0

Table of Contents

- 1. Introduction3
 - 1.1. Purpose.....3
 - 1.2. Scope.....3
 - 1.3. Definitions, Acronyms, and Abbreviations3
- 2. Overall Description3
 - 2.1. System Perspective.....3
 - 2.2. User Classes and Characteristics.....3
 - 2.3. Operating Environment.....3
 - 2.4. Assumptions and Dependencies3
- 3. System Features and Functional Requirements4
 - 3.1. Feature 1:.....4
 - 3.2. Feature 2:.....4
- 4. Non-Functional Requirements.....4
- 5. System Models (Diagrams)4
 - 5.1. ERD.....5
 - 5.2. Use Case Diagram5
 - 5.3. Activity Diagram6
 - 5.4. Class Diagram.....7
 - 5.5. Sequence Diagram8
- 6. Appendices8

1. Introduction

1.1. Purpose

This document specifies the functional requirements for a User Authentication System that handles registration, login, profile viewing, and logout functionalities. The intended audience includes developers, testers, and stakeholders involved in the system implementation.

1.2. Scope

The system will provide secure user authentication with the following capabilities:

- User registration with validation
- Secure login with session management
- Profile/dashboard viewing for authenticated users
- Logout functionality
- Protection of routes from unauthorized access

1.3. Definitions, Acronyms, and Abbreviations

- **ERD**: Entity Relationship Diagram
- **JWT**: JSON Web Token
- **API**: Application Programming Interface
- **UI**: User Interface
- **DTO**: Data Transfer Object

2. Overall Description

2.1. System Perspective

The authentication system integrates with a larger web application, serving as the gateway for user access to protected resources. It follows a client-server architecture with React frontend and Spring Boot backend.

2.2. User Classes and Characteristics

- **Guest User**: Unauthenticated user who can only register or login
- **Authenticated User**: Logged-in user who can view profile and logout

2.3. Operating Environment

- **Frontend**: React.js with modern browser support
- **Backend**: Spring Boot 3.x with Java 17+
- **Database**: MySQL/PostgreSQL
- **Tools**: Maven/Gradle, Node.js

2.4. Assumptions and Dependencies

- Users have valid email addresses
- System uses HTTPS for secure communication
- JWT tokens for session management

- Password encryption using BCrypt

3. System Features and Functional Requirements

Describe each major feature of the system and its functional requirements.

3.1. Feature 1: User Registration

Description: Allows new users to create an account

Functional Requirements:

- User can fill registration form with username, email, password
- System validates email format and password strength
- System checks for duplicate email/username
- Successful registration creates user record and redirects to login

3.2. Feature 2: User Authentication

Description: Handles login and logout operations

Functional Requirements:

- User can login with email/username and password
- System validates credentials against database
- Successful login generates JWT token and session
- User can logout, destroying session/token
- Protected routes check authentication status

4. Non-Functional Requirements

- Performance: Login response < 2 seconds
- Security: Passwords stored as hashes, JWT token expiration
- Usability: Intuitive UI with clear error messages
- Reliability: 99.9% uptime for authentication services
- Scalability: Support for 10,000 concurrent users

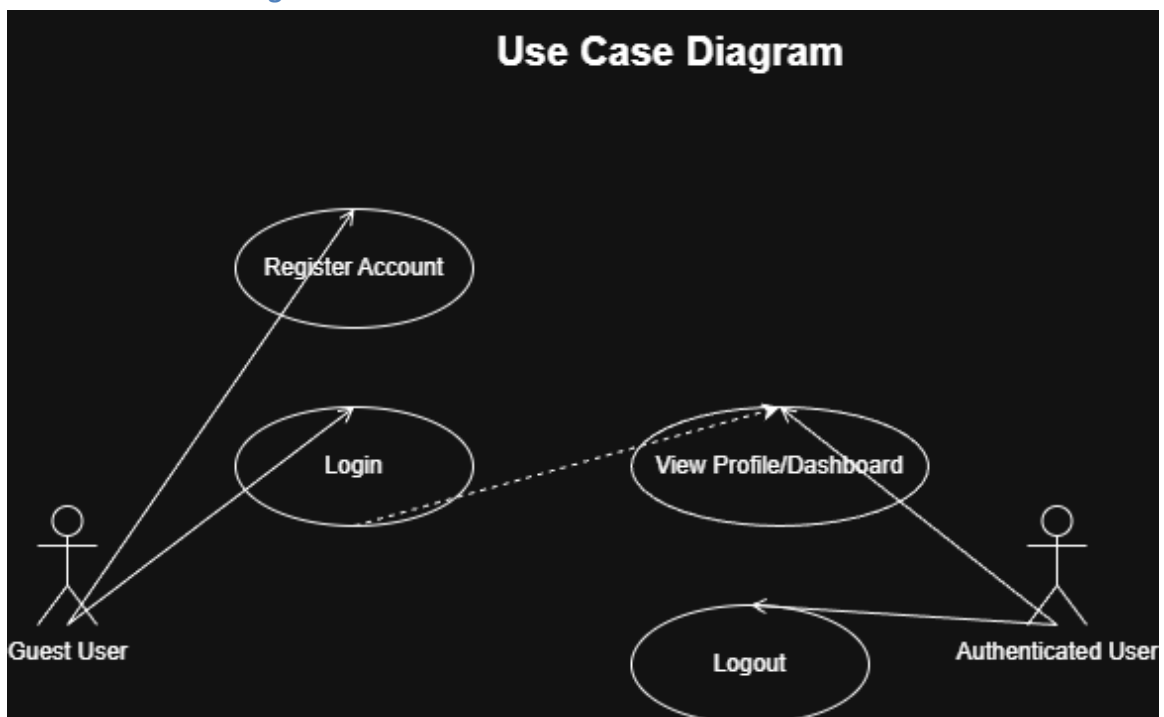
5. System Models (Diagrams)

Insert the necessary diagrams for the system:

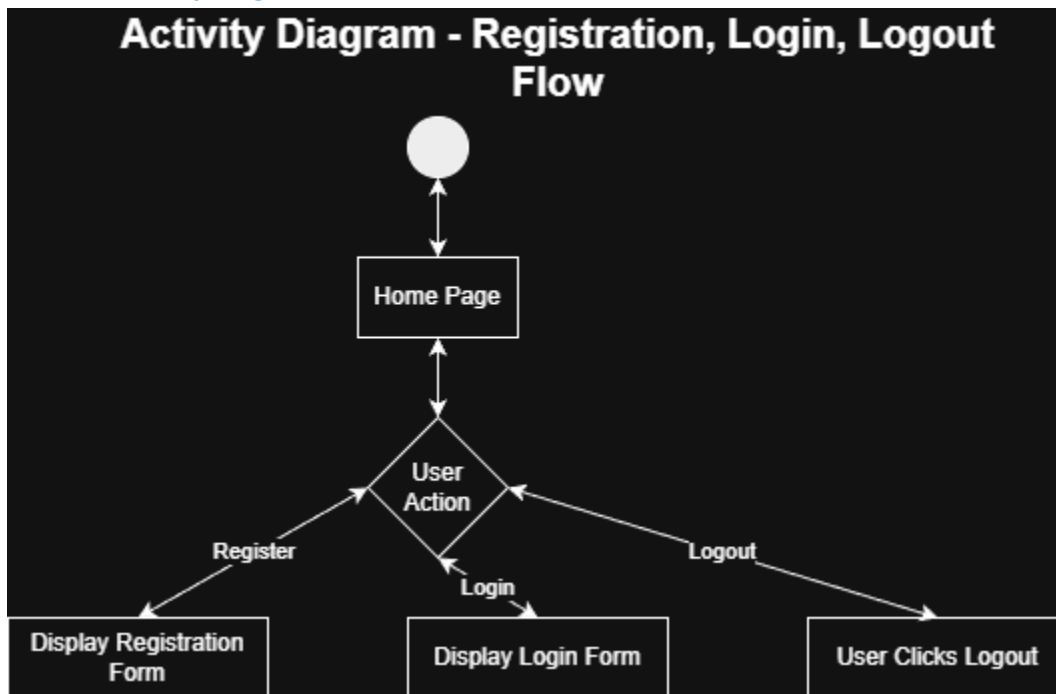
5.1. ERD

Users
id (PK, int)
username (varchar)
email (varchar)
password_hash (varchar)
created_at (datetime)
updated_at (datetime)
is_active (boolean)

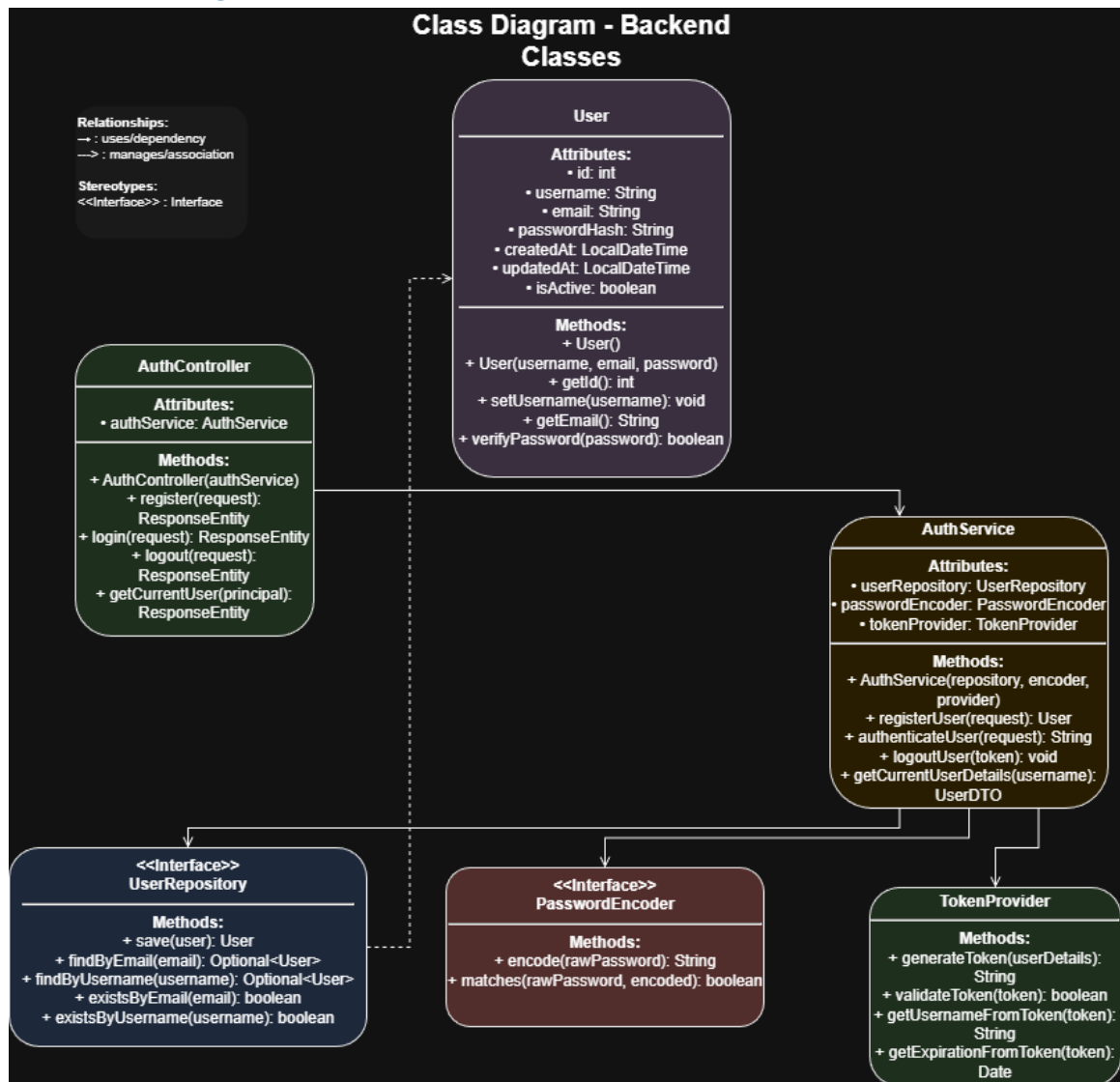
5.2. Use Case Diagram



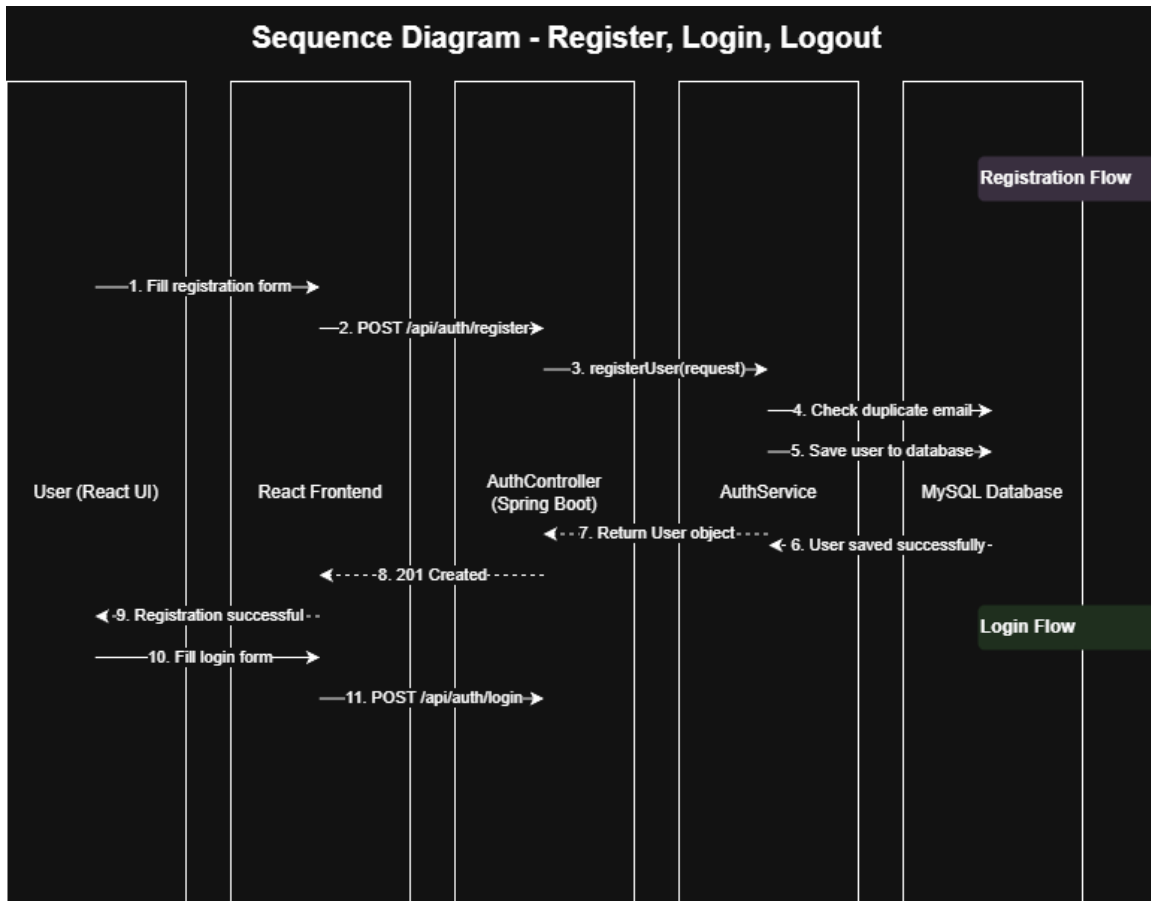
5.3. Activity Diagram



5.4. Class Diagram



5.5. Sequence Diagram



6. Appendices

- References: Spring Security Documentation, JWT RFC 7519
- Tools Used: Draw.io for diagram creation