



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

IT342-Section SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: Laboratory Activity: Mini App – User Registration
& Authentication

Prepared By: Bianca Margarete G. Vallo

Date of Submission: February 3, 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 define the functional and non-functional requirements of a Mini Application that supports user registration, login, profile access, and logout. This document is intended for instructors, system designers, and developers who will analyze the system flow and architecture prior to implementation.

1.2. Scope

The system provides basic user authentication features. It allows users to create an account, log in securely, view their profile, and log out. The system restricts access to protected pages when the user is not authenticated.

1.3. Definitions, Acronyms, and Abbreviations

Term	Definition
ERD	Entity Relationship Diagram
UI	User Interface
API	Application Programming Interface
JWT	JSON Web Token
FRS	Functional Requirements Specification

2. Overall Description

2.1. System Perspective

The system is a web-based application composed of:

- A React frontend for user interaction
- A Spring Boot backend API for authentication logic
- A relational database for storing user data

The system follows a client-server architecture.

2.2. User Classes and Characteristics

User Type	Description
Guest User	Can register and log in but cannot access protected pages
Authenticated User	Can view profile/dashboard and log out

2.3. Operating Environment

- **Frontend:** Web browser (Chrome, Edge, Firefox)
- **Backend:** Spring Boot REST API
- **Database:** MySQL / PostgreSQL
- **Diagram Tool:** draw.io

2.4. Assumptions and Dependencies

- Users have internet access
- Passwords are securely hashed
- Authentication uses token-based security
- The backend API is available during system operation

3. System Features and Functional Requirements

3.1. Feature 1: User Registration

Description: Allows a guest user to create a new account by providing valid credentials.

Functional Requirements:

- The system shall allow users to register using a unique email/username
- The system shall validate user input
- The system shall store encrypted passwords
- The system shall prevent duplicate accounts

3.2. Feature 2: User Login

Description: Allows registered users to authenticate and access protected pages.

Functional Requirements:

- The system shall authenticate users using email/username and password
- The system shall generate an authentication token upon successful login
- The system shall deny access if credentials are invalid

3.3. Feature 3: View Profile

Description: Allows authenticated users to view their profile information.

Functional Requirements:

- The system shall restrict access to authenticated users only
- The system shall display user-specific information

3.4. Feature 4: User Logout

Description: Allows authenticated users to end their session.

Functional Requirements:

- The system shall invalidate the authentication token

- The system shall redirect users to the login page

4. Non-Functional Requirements

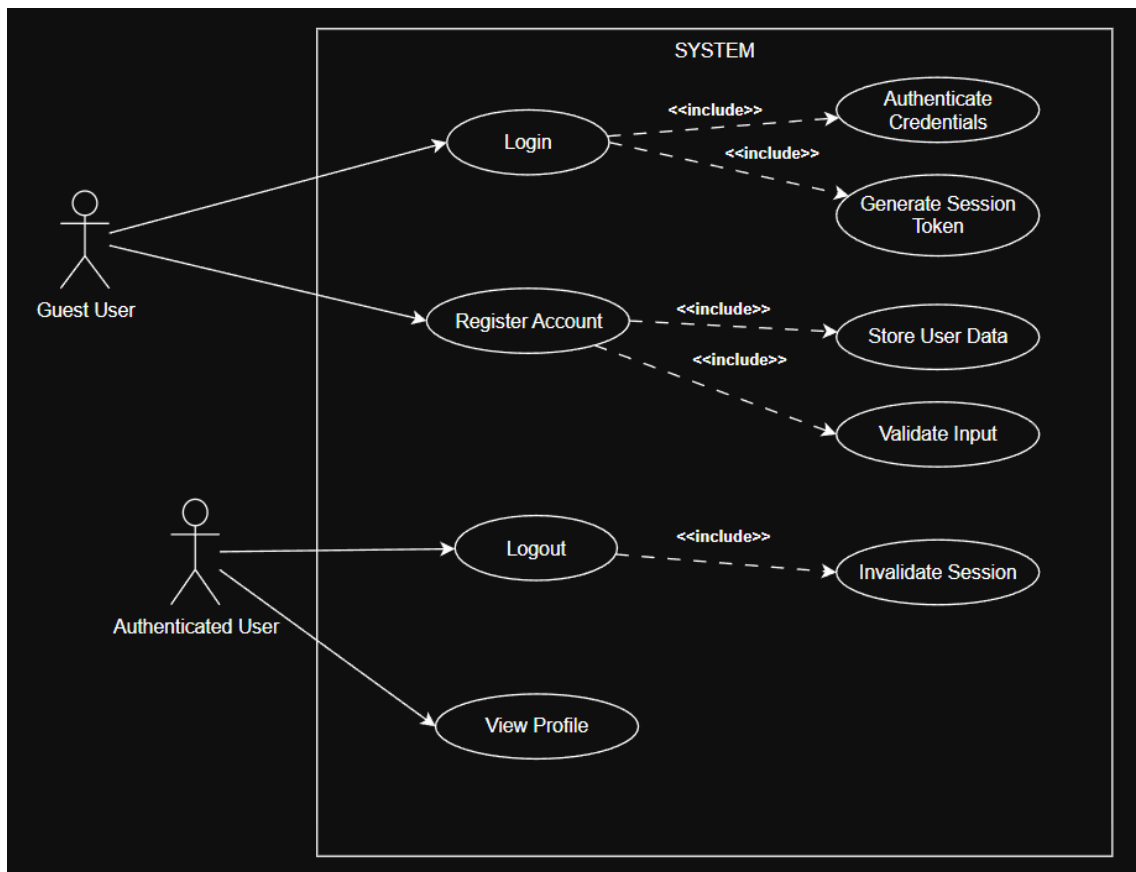
- Passwords must be encrypted and protected pages must require authentication
- The interface must be simple and intuitive for end users.
- Authentication responses should occur within acceptable response time
- System must consistently handle login and logout actions

5. System Models (Diagrams)

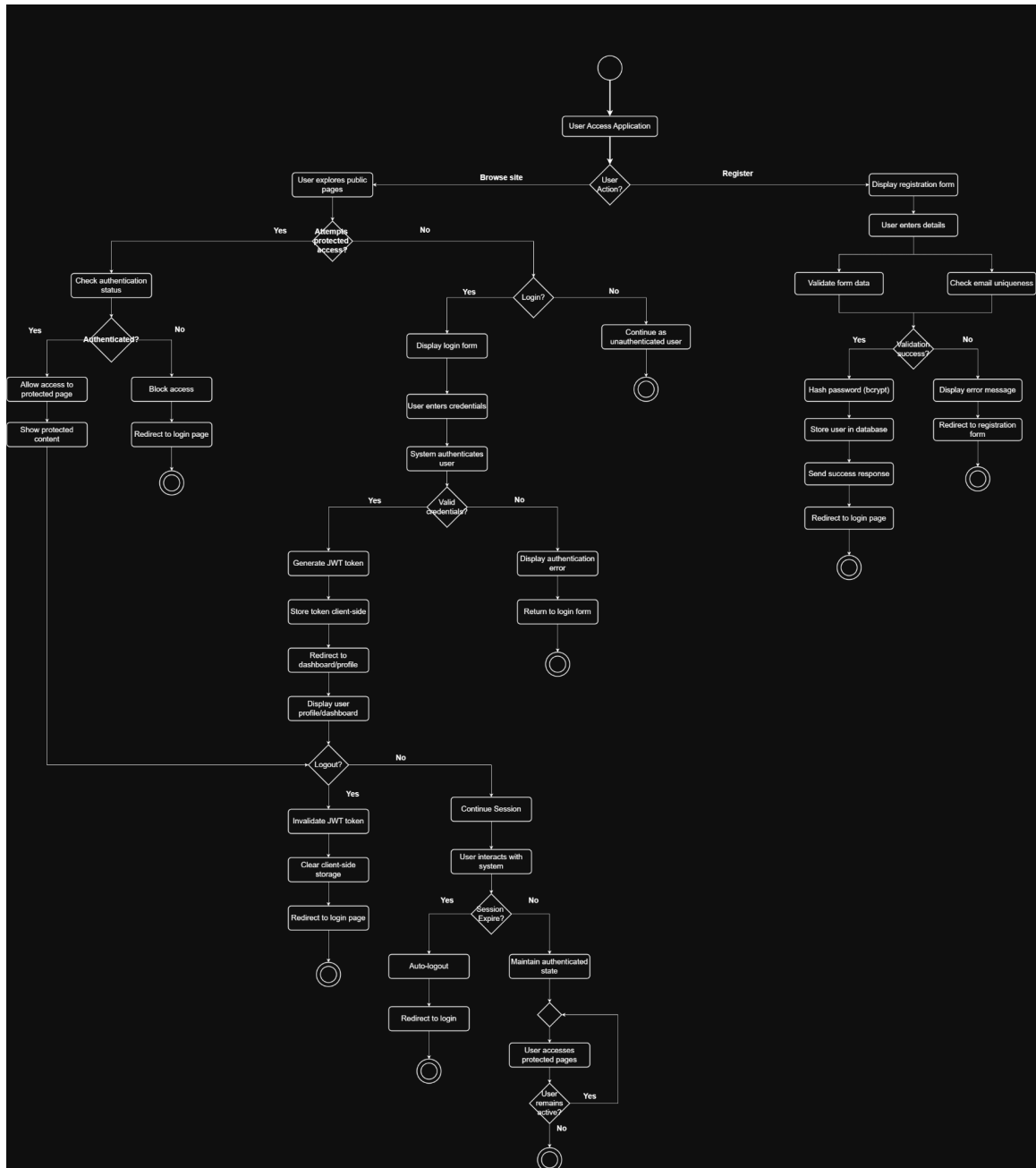
5.1. ERD

Users	
PK	<u>UserID</u>
	Username
	Email
	FirstName
	LastName
	PasswordHash
	PhoneNumber
	IsEmailVerified
	IsActive
	CreatedAt
	UpdatedAt
	LastLoginAt

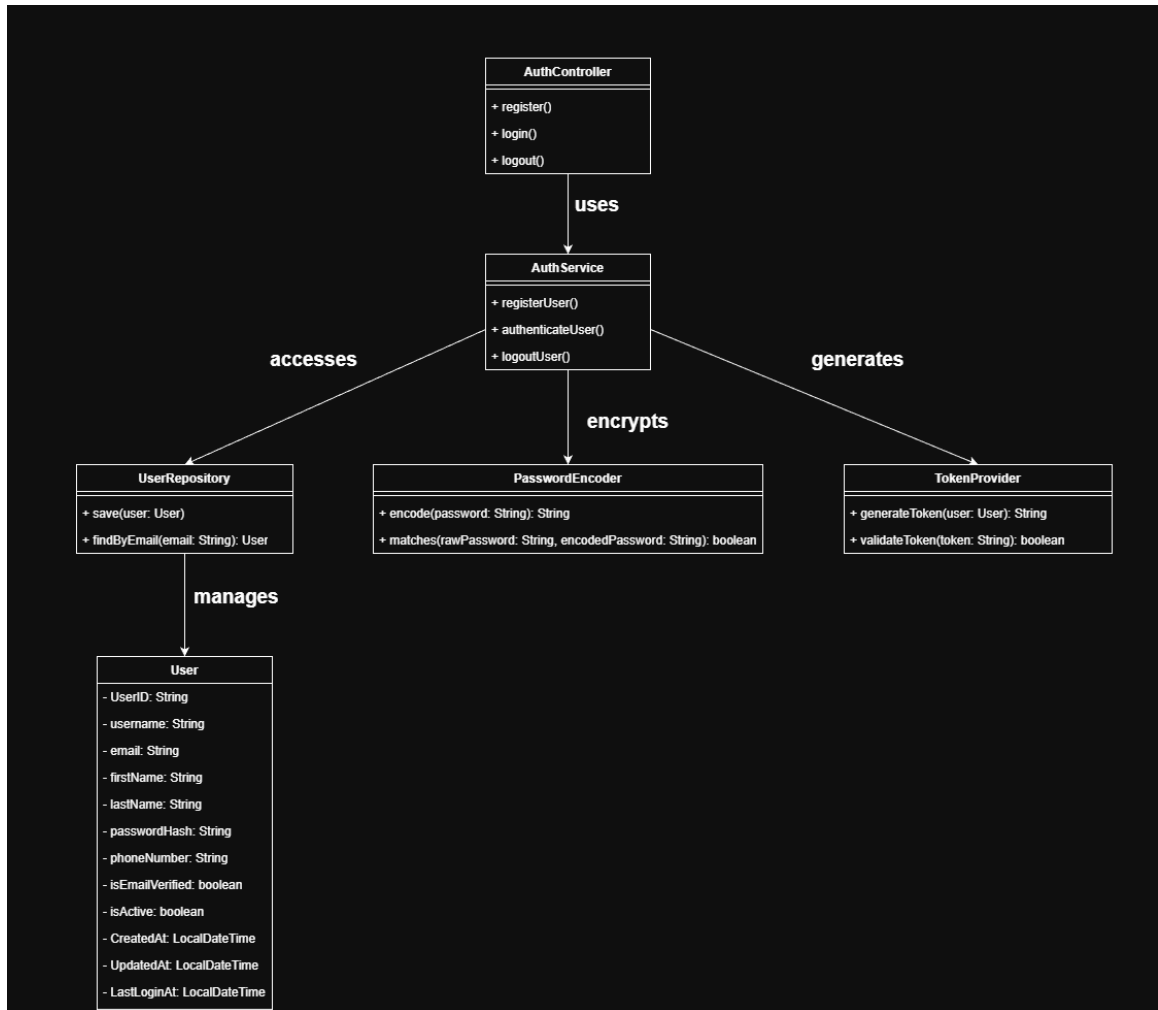
5.2. Use Case Diagram



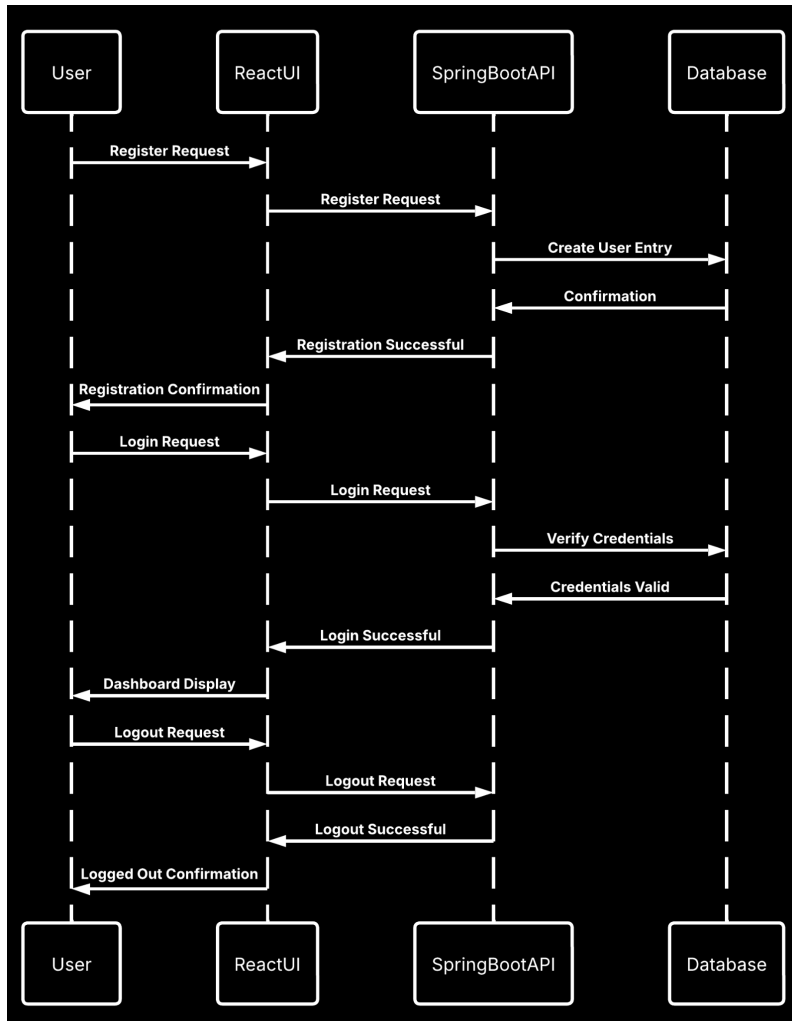
5.3. Activity Diagram



5.4. Class Diagram



5.5. Sequence Diagram



6. Appendices

Appendix A: Diagram Development Tools

All system diagrams included in this document were created using draw.io

Appendix B: Future Enhancements

The following features may be considered for future versions of the system:

- Password recovery and reset functionality
- Role-based access control
- Multi-factor authentication
- User activity and audit logging