CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-G1
SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: User Registration & Authentication

Prepared By: John Hector P. Villarta

Date of Submission: February 6, 2026

Version: 2

Table of Contents

1. 3
   1.1. 3
   1.2. 3
   1.3. 3
2. 3
   2.1. 3
   2.2. **Error! Bookmark not defined.**
   2.3. **Error! Bookmark not defined.**
   2.4. 3
3. 4
   3.1. 4
   3.2. 4
4. 4
5. 5
   5.1. 5
   5.2. 5
   5.3. 6
   5.4. 7
   5.5. 7
6. 9

1. Introduction

1.1. Purpose

The purpose of this document is to describe the functional and non-functional requirements of the User Registration & Authentication System. This document is intended for system analysts, developers, instructors, and stakeholders involved in the design, development, and evaluation of the system. The diagrams and requirements defined in this document will serve as the basis for the system implementation in the next development phase.

1.2. Scope

The system provides basic account management and authentication features for a web-based application. It allows users to register an account, log in using their credentials, view their profile or dashboard after successful authentication, and log out of the system. The system also ensures that protected pages and features cannot be accessed by users who are not authenticated. The scope of this project is limited to registration, authentication, session handling, and access control.

1.3. Definitions, Acronyms, and Abbreviations

- ERD – Entity Relationship Diagram
- UI – User Interface
- API – Application Programming Interface
- JWT – JSON Web Token
- Authentication – Process of verifying the identity of a user
- Authorization – Process of determining access rights to system resources

2. Overall Description

2.1. System Perspective

The User Registration & Authentication System is a supporting subsystem of a larger web-based application. It acts as a security and identity management layer that communicates with the front-end user interface and the backend services. The system is responsible for validating user credentials, managing sessions or tokens, and controlling access to protected resources.

2.2. User Classes and Characteristics

- Guest User A user who has not logged in. A guest user can access public pages and is allowed to register and log in.
- Authenticated User A registered user who has successfully logged in. An authenticated user can access protected pages such as the profile or dashboard and can log out of the system.

2.3. Operating Environment

- Client-side application developed using React

- Backend service developed using Spring Boot
- Database management system (e.g., MySQL or PostgreSQL)
- Web browser (Google Chrome, Mozilla Firefox, Microsoft Edge)
- Development and modeling tools such as draw.io / diagrams.net

2.4. Assumptions and Dependencies

- Users have access to a stable internet connection.
- The system depends on a relational database to store user information.
- The system depends on the backend authentication service and API availability.
- Password hashing and token generation libraries are available and properly configured.

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 a new user to create an account in the system.
Functional Requirements:

- The system shall allow a guest user to register using a username, email, and password.
- The system shall validate user input before saving.
- The system shall hash passwords before storing them.
- The system shall prevent duplicate email registrations.
- The system shall store the user data in the database.

3.2. Feature 2: User Authentication and Session Management

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

- The system shall allow a registered user to log in using an email and password.
- The system shall validate the provided credentials against the stored user records.
- The system shall deny access when invalid credentials are provided.
- The system shall create an authenticated session or issue an access token after a successful login
- The system shall allow authenticated users to access protected pages such as the profile or dashboard.
- The system shall prevent guest users from accessing protected pages.
- The system shall allow authenticated users to log out of the system.
- The system shall invalidate the active session or token upon logout.

4. Non-Functional Requirements

- Security: The system shall securely store passwords using a hashing algorithm and shall not store plain text passwords.

- Usability: The system shall provide clear error messages for invalid login or registration attempts.

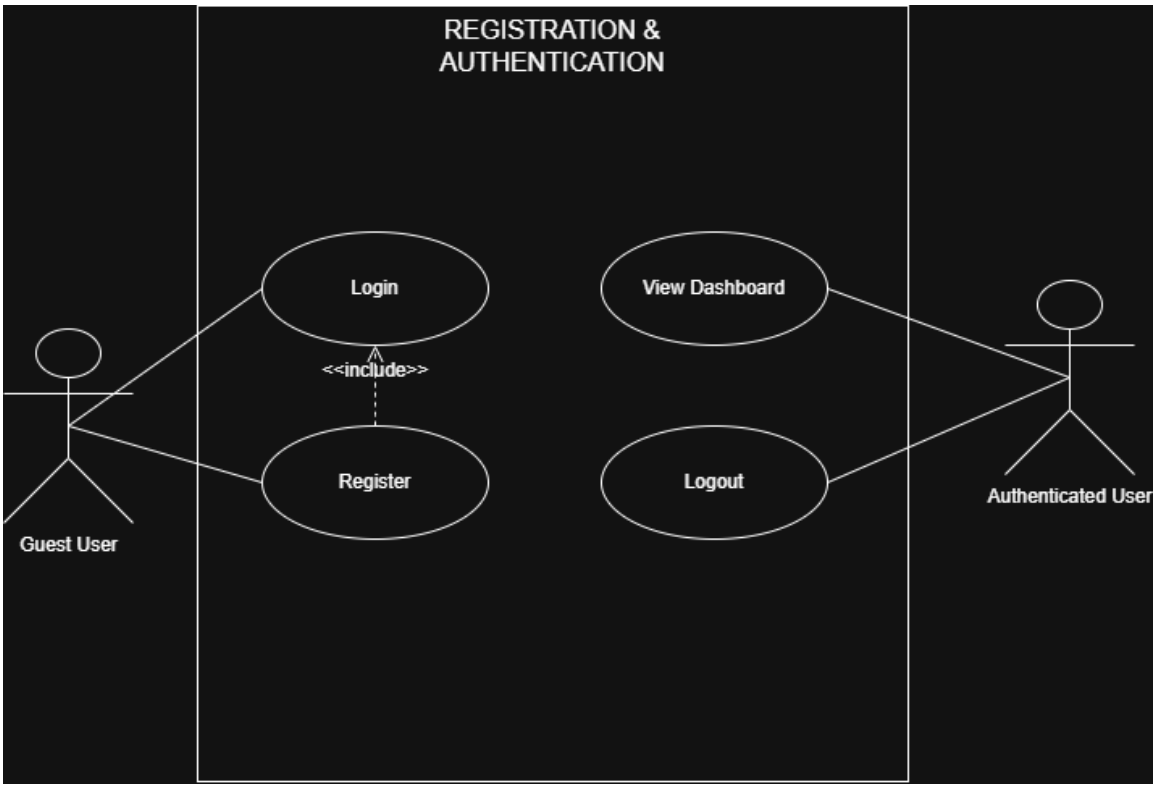
5. System Models (Diagrams)

Insert the necessary diagrams for the system:

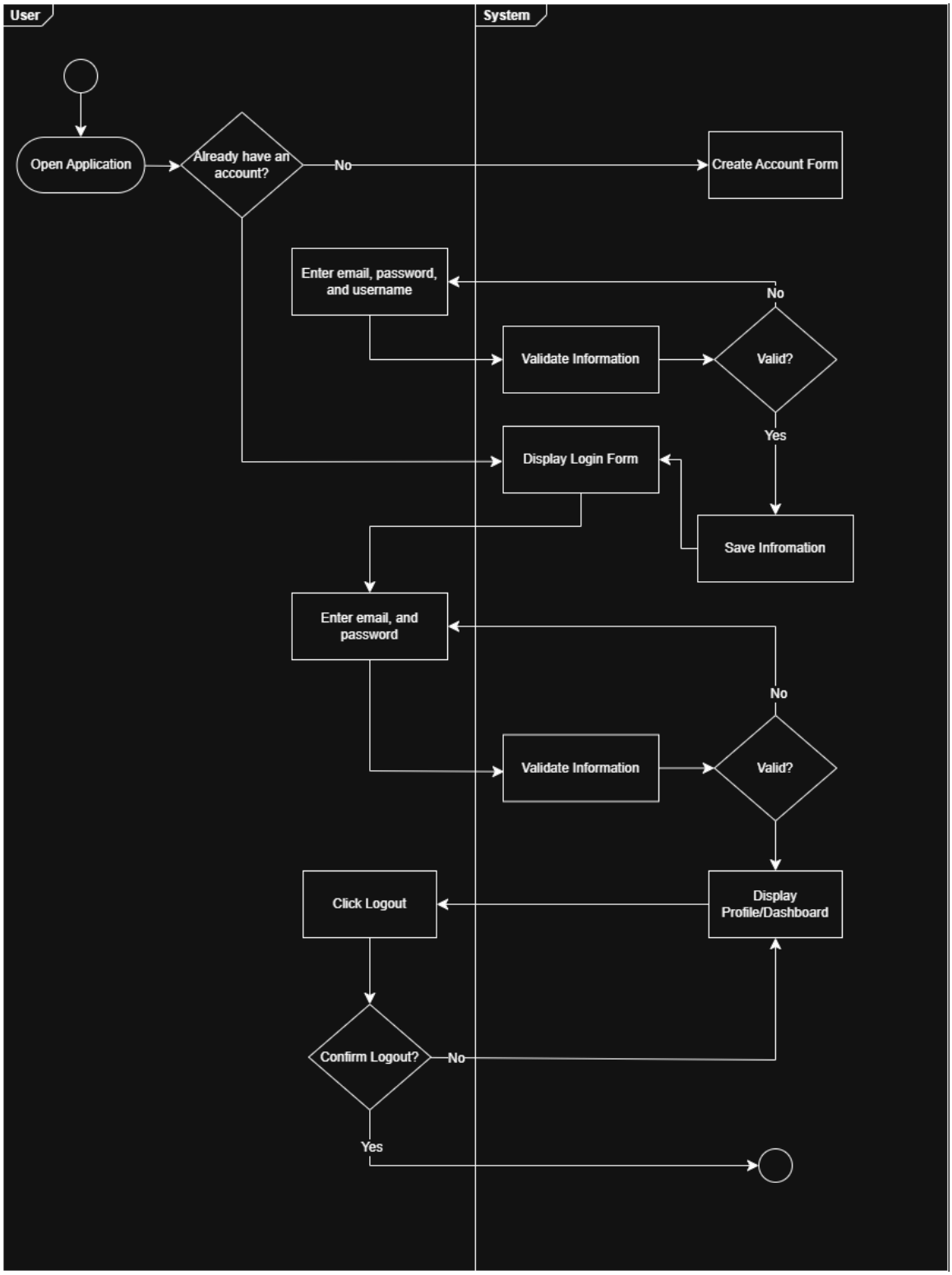
5.1. ERD

User		
PK	userId	int
	email	varchar(unique)
	username	varchar
	passwordHash	varchar
	firstname	varchar
	lastname	varchar
	isActive	boolean
	createdAt	timestamp
	updatedAt	timestamp

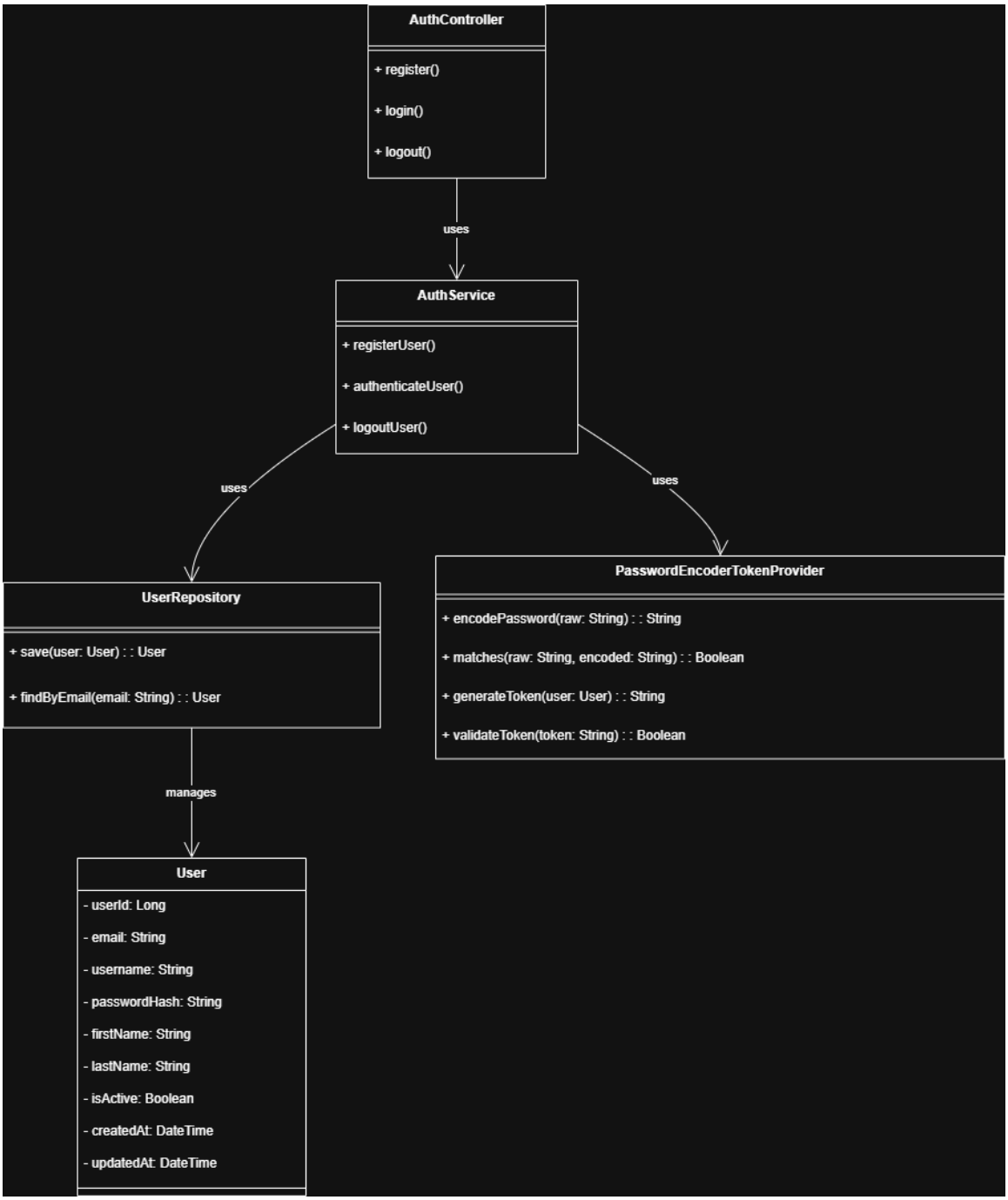
5.2. Use Case Diagram



5.3. Activity Diagram

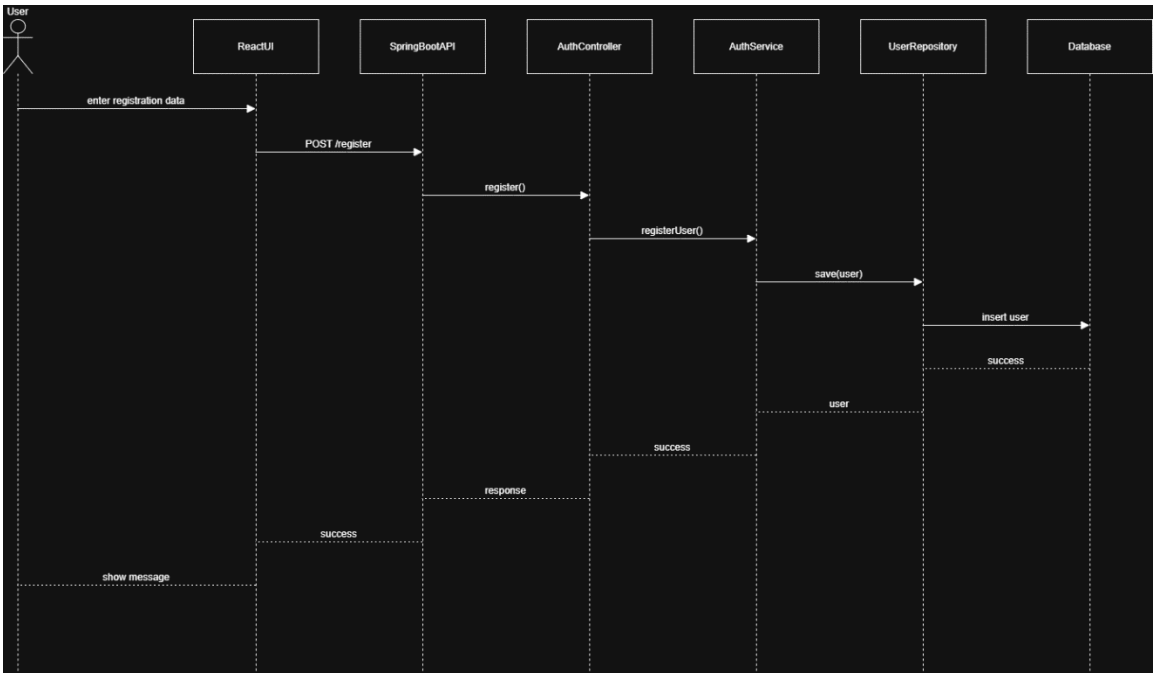


5.4. Class Diagram

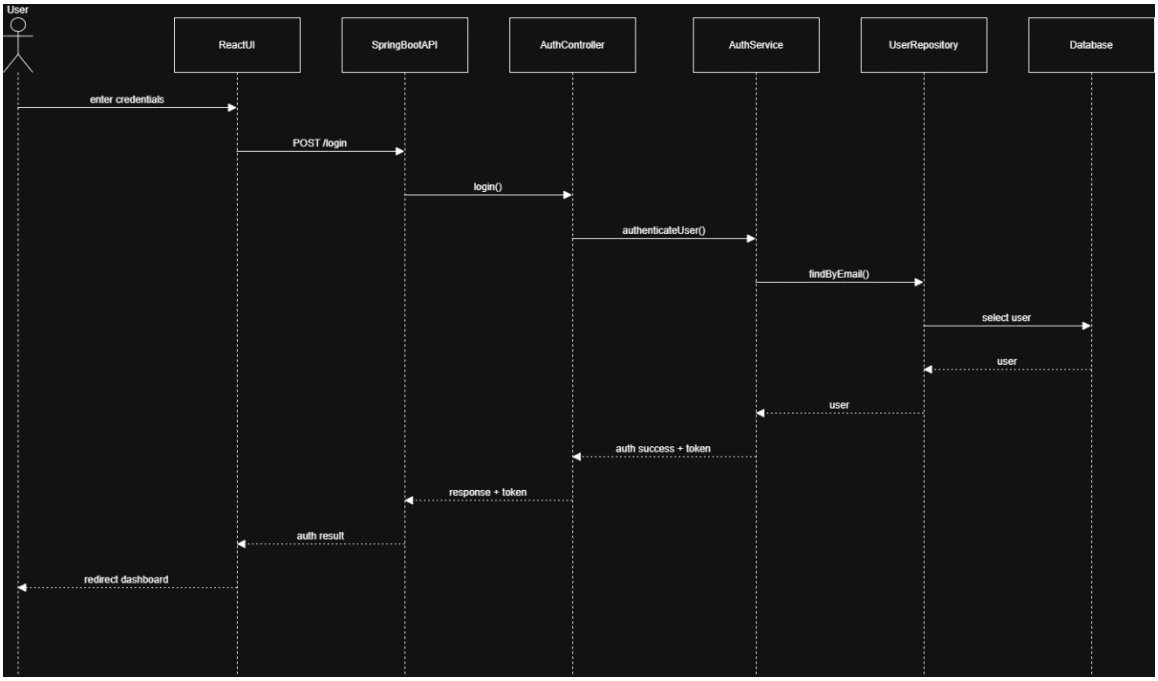


5.5. Sequence Diagram

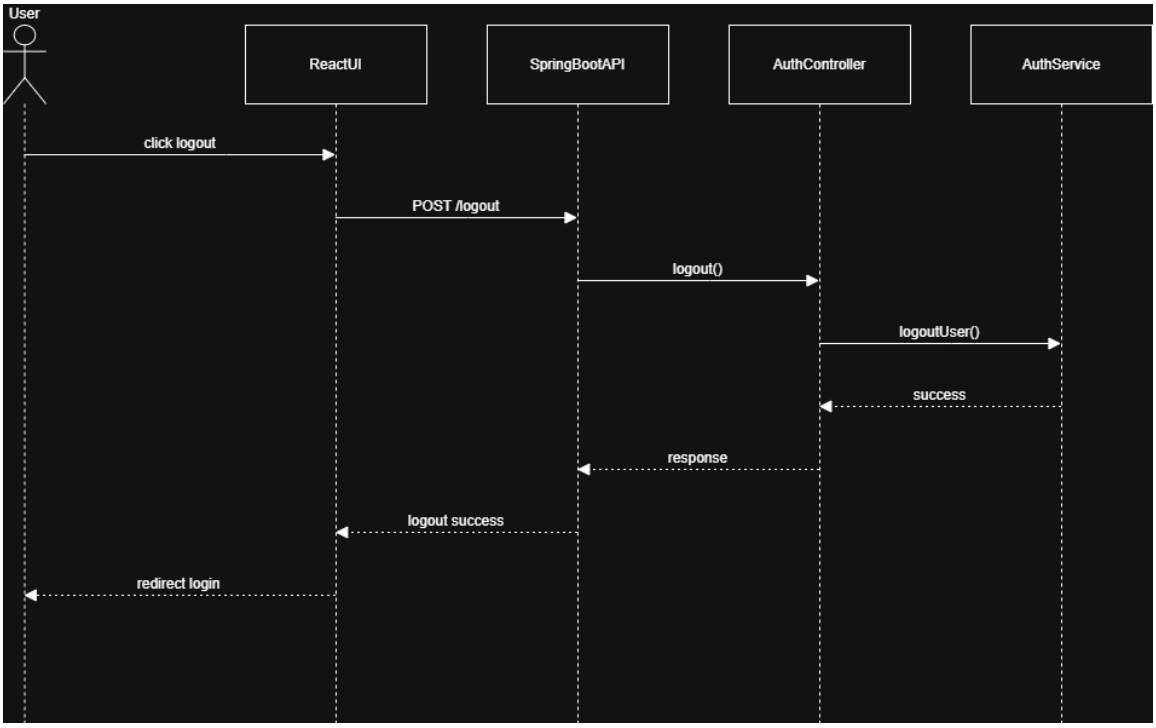
- Registration



- Login



- Logout



6. Appendices

This document is prepared for academic purposes under IT342 – Systems Integration and Architecture

1. The diagrams and specifications will be used as reference materials for the implementation phase during the face-to-face laboratory session.