CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-G1
SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: User and Authentication

Prepared By: Kean Maverick W. Saligue

Date of Submission: February 6, 2026

Version: 2

Table of Contents

1. Introduction 3
   1.1. Purpose 3
   1.2. Scope 3
   1.3. Definitions, Acronyms, and Abbreviations 3
2. Overall Description 4
   2.1. System Perspective 4
   2.2. User Classes and Characteristics 4
   2.3. Operating Environment 4
   2.4. Assumptions and Dependencies 5
3. System Features and Functional Requirements 5
   3.1. Feature 1: 5
   3.2. Feature 2: 6
4. Non-Functional Requirements 6
5. System Models (Diagrams) 7
   5.1. ERD 7
   5.2. Use Case Diagram 7
   5.3. Activity Diagram 8
   5.4. Class Diagram 5
   5.5. Sequence Diagram 5
6. Appendices 11

1. Introduction

1.1. Purpose

The purpose of this document is to define the Functional Requirements Specification (FRS) for the *User and Authentication System*. This document is prepared to describe the system's functional and non-functional requirements, architectural design, and operational constraints. It serves as a reference for system development, evaluation, and future enhancement.

1.2. Scope

The User and Authentication System is designed to provide secure user registration, login, profile management, and logout functionalities. The system enforces access control by restricting protected resources to authenticated users only. This document covers the analysis, design, and documentation of the system and will be used as the basis for implementation in subsequent development phases.

1.3. Definitions, Acronyms, and Abbreviations

Term / Acronym	Definition
FRS	Functional Requirements Specification. A document that describes the system's functional and non-functional requirements.
UI	User Interface. The part of the system that allows users to interact with the application.
API	Application Programming Interface. A set of rules that enables communication between system components.
ERD	Entity Relationship Diagram. A diagram that represents the structure of the system database.
UML	Unified Modeling Language. A standard modeling language used to create system diagrams.
JWT	JSON Web Token. A security token used for user authentication.
DB	Database. A structured collection of data used to store user information.
ReactJS	A JavaScript framework used to develop the system's frontend interface.
Spring Boot	A Java-based framework used to develop the system's backend

Term / Acronym	Definition
	services.
Client-Server Architecture	A system structure where the client communicates with the server to request services.
Relational Database	A database system that stores data in tables with defined relationships.
Authentication Token	A digital credential generated after login to verify user identity.
Encryption	The process of securing sensitive data such as passwords.

2. Overall Description

2.1. System Perspective

The User and Authentication System is a web-based application developed using a client-server architecture. The system will use React-based front-end, a Spring Boot-based back-end, and a relational database for data persistence. Communication between system components is handled through RESTful APIs.

2.2. User Classes and Characteristics

Identify the different types of users and their characteristics.

User Class	Description
Guest User	Users who have not logged in and may register
Authenticated User	Users who have successfully logged in and can access system features

2.3. Operating Environment

The system is intended to operate in the following environment:

- Operating System: Windows, macOS, or Linux
- Frontend Framework: ReactJS
- Backend Framework: Spring Boot (Java)
- Database Management System: MySQL or PostgreSQL

- Web Browser: Chrome, Edge, Firefox
- Development Tools: Visual Studio Code and IntelliJ IDEA

2.4. Assumptions and Dependencies

The development and operation of the system are based on the following assumptions:

- The user has stable internet connectivity
- The application server and database server are operational
- Required authentication and encryption libraries are available
- The deployment environment supports Java and Node.js
- System maintenance is performed regularly

3. System Features and Functional Requirements

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

3.1. Feature 1: User Registration

Description: This feature allows users to create an account and securely authenticate using valid credentials.
Functional Requirements:

- The system shall validate all input data before processing
- The system shall allow users to register with valid personal information
- The system shall encrypt passwords prior to database storage

3.2. Feature 2: User Login

Description: This feature enables registered users to authenticate themselves and access protected system resources.
Functional Requirements:

- The system shall allow users to log in using registered email and password
- The system shall validate login credentials
- The system shall authenticate users against stored records
- The system shall generate a secure authentication token upon successful login
- The system shall deny access for invalid credentials
- The system shall redirect authenticated users to their dashboard

3.3. Feature 3: User Profile Management

Description: This feature allows authenticated users to view and update their personal information through the system interface.
Functional Requirements:

- The system shall allow authenticated users to view their profile information
- The system shall allow users to edit personal details such as name, username , email, phone number, and password
- The system shall validate updated information before saving
- The system shall encrypt updated passwords before storage
- The system shall update user records in the database
- The system shall confirm successful profile updates to the user

3.4. Feature 4: Logout

Description: This feature enables authenticated users to securely terminate their session and prevent unauthorized access.
Functional Requirements:

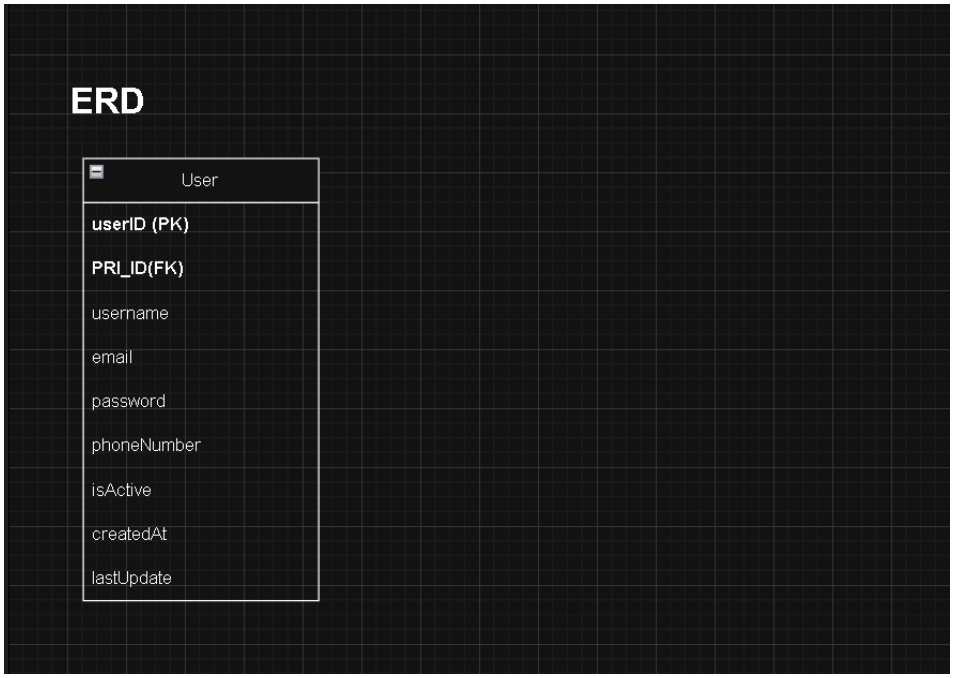
- The system shall allow users to log out at any time
- The system shall invalidate active authentication tokens upon logout
- The system shall clear user session data
- The system shall redirect users to the login page after logout
- The system shall prevent access to protected resources after logout

4. Non-Functional Requirements

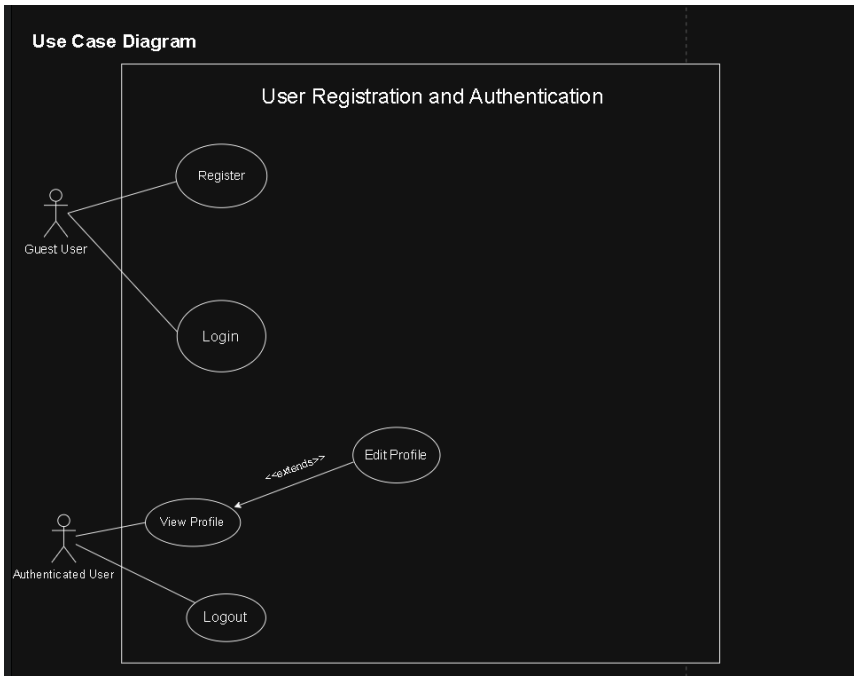
Category	Requirement
Performance	The system shall process user requests within two seconds
Security	User credentials shall be encrypted and protected
Usability	The interface shall be clear and easy to navigate
Reliability	The system shall operate consistently under normal conditions
Availability	The system shall maintain at least 99% uptime
Scalability	The system shall support future user growth
Maintainability	The system shall follow modular and documented design

5. System Models (Diagrams)

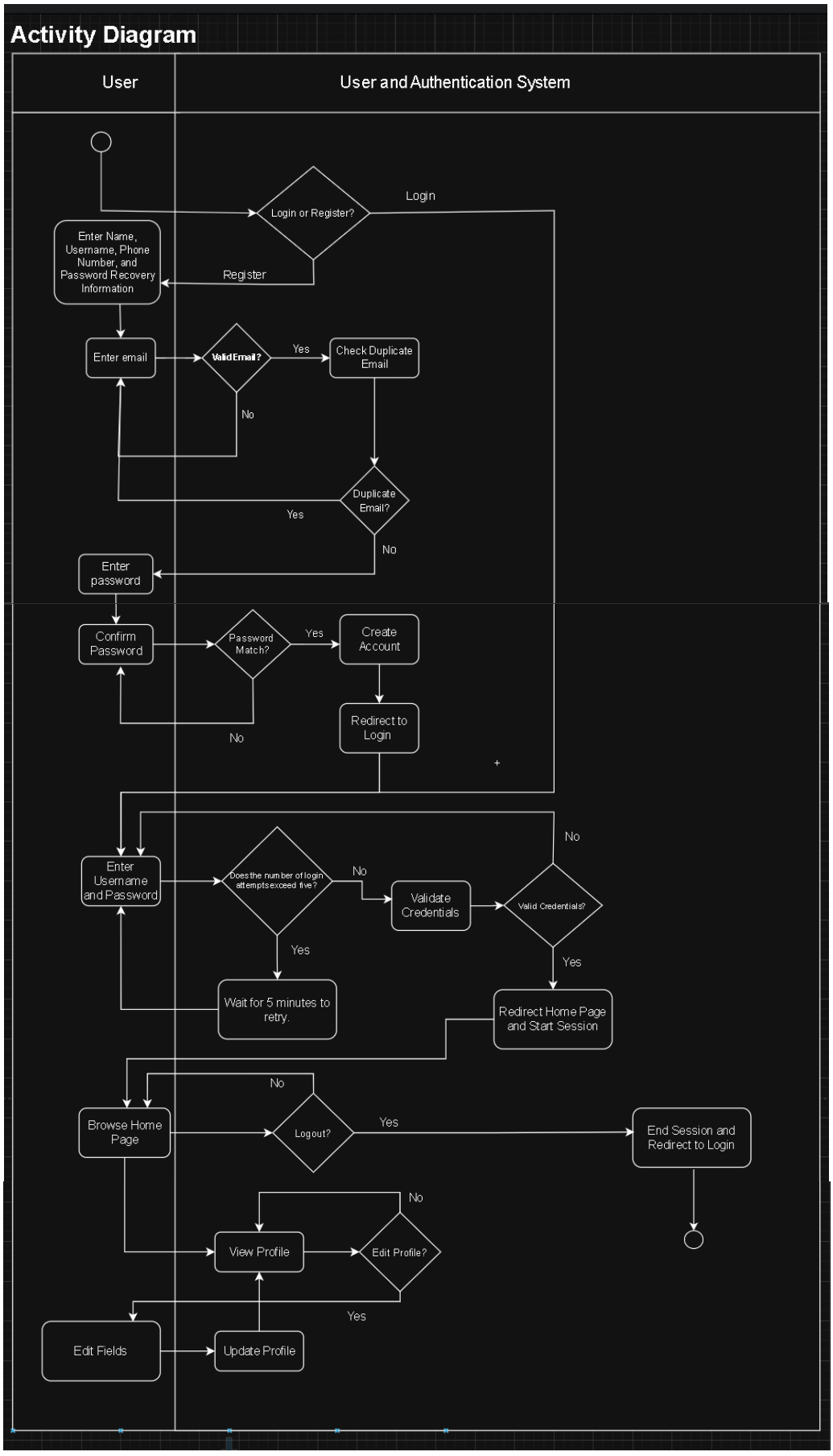
5.1. ERD



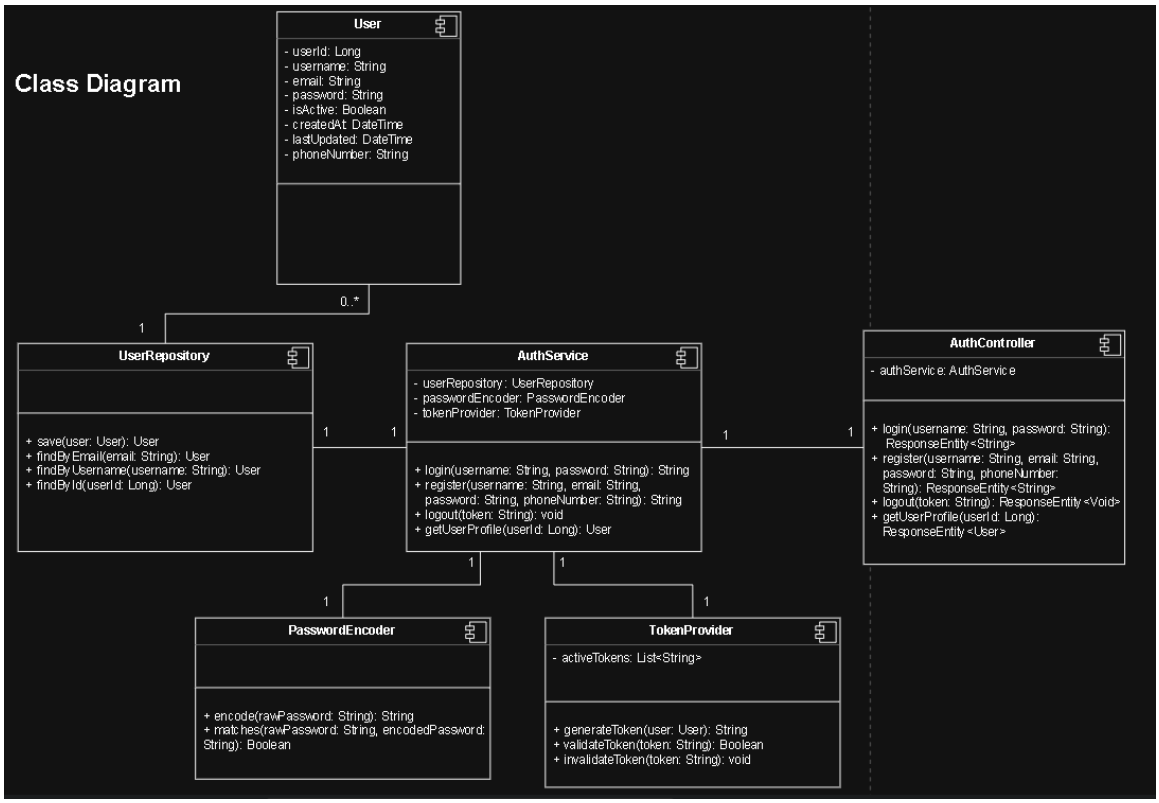
5.2. Use Case Diagram



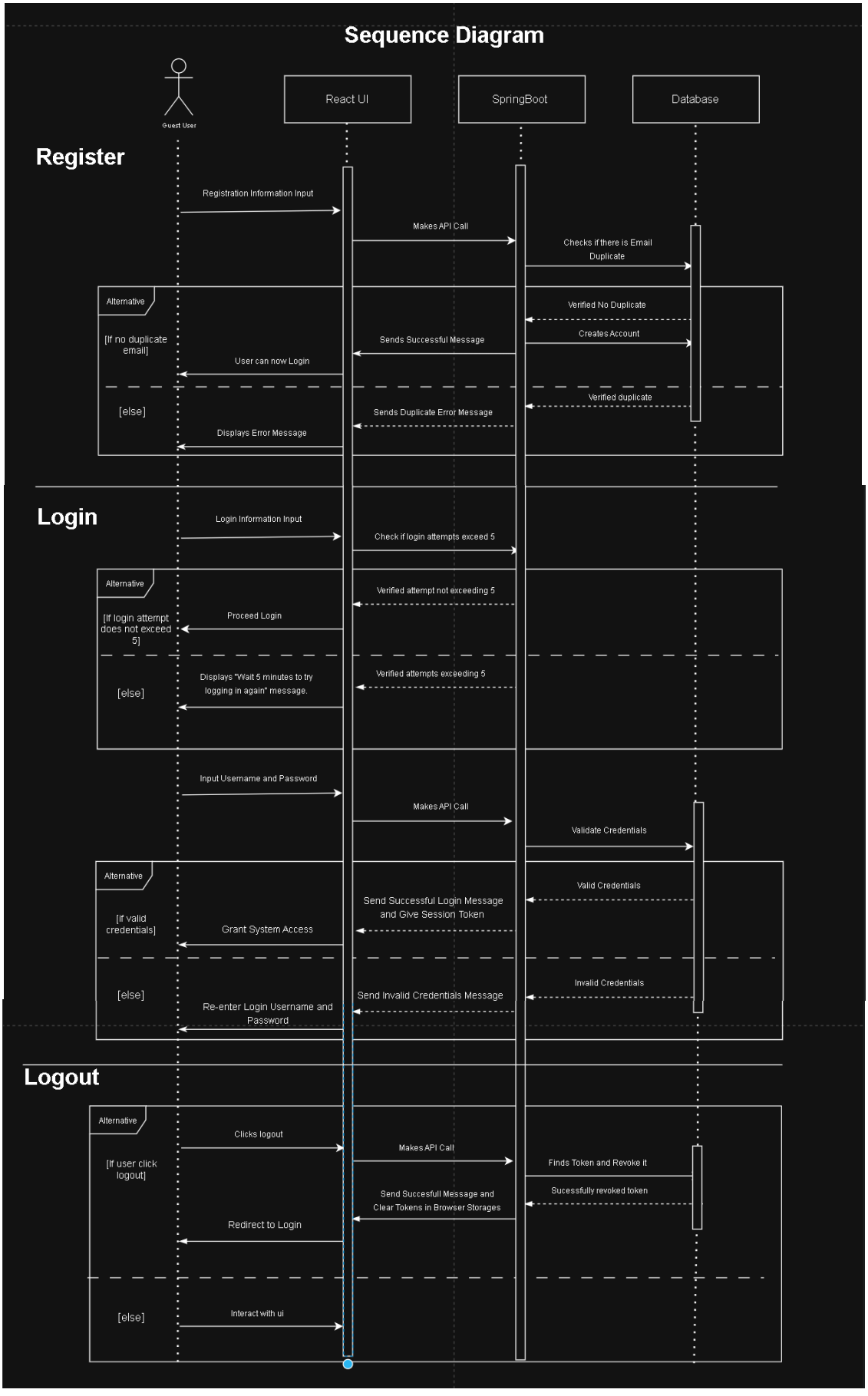
5.3. Activity Diagram



5.4. Class Diagram



5.5. Sequence Diagram



6. Appendices

References

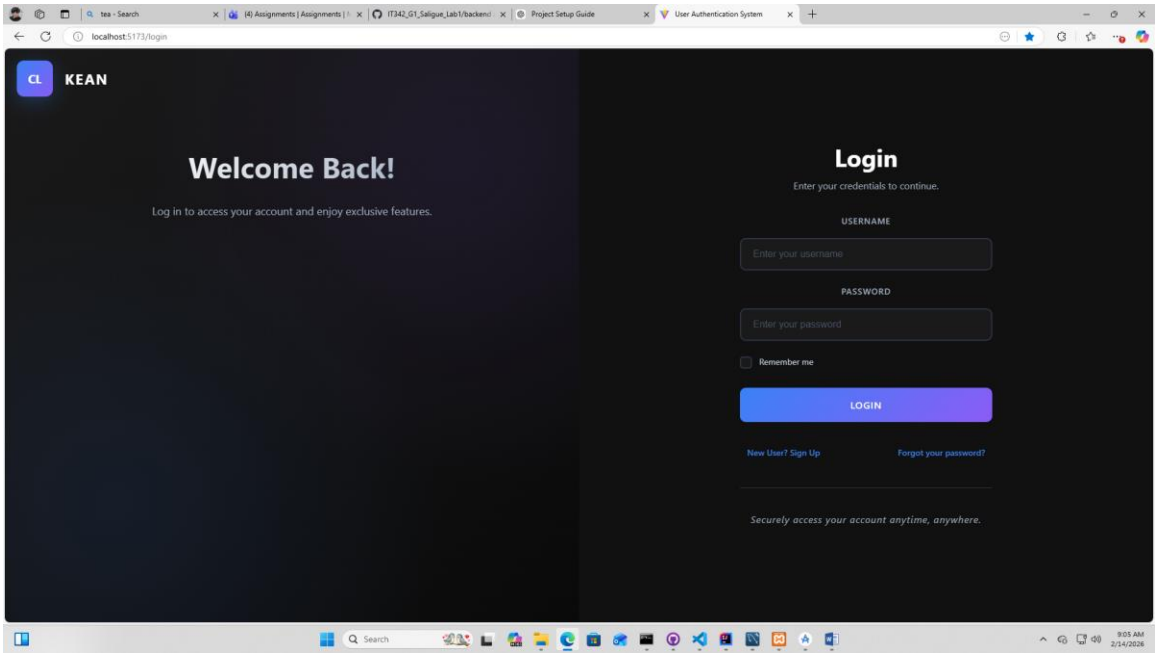
- Spring Boot Technical Documentation
- ReactJS Development Guide
- UML Modeling Standards
- JWT Security Specifications

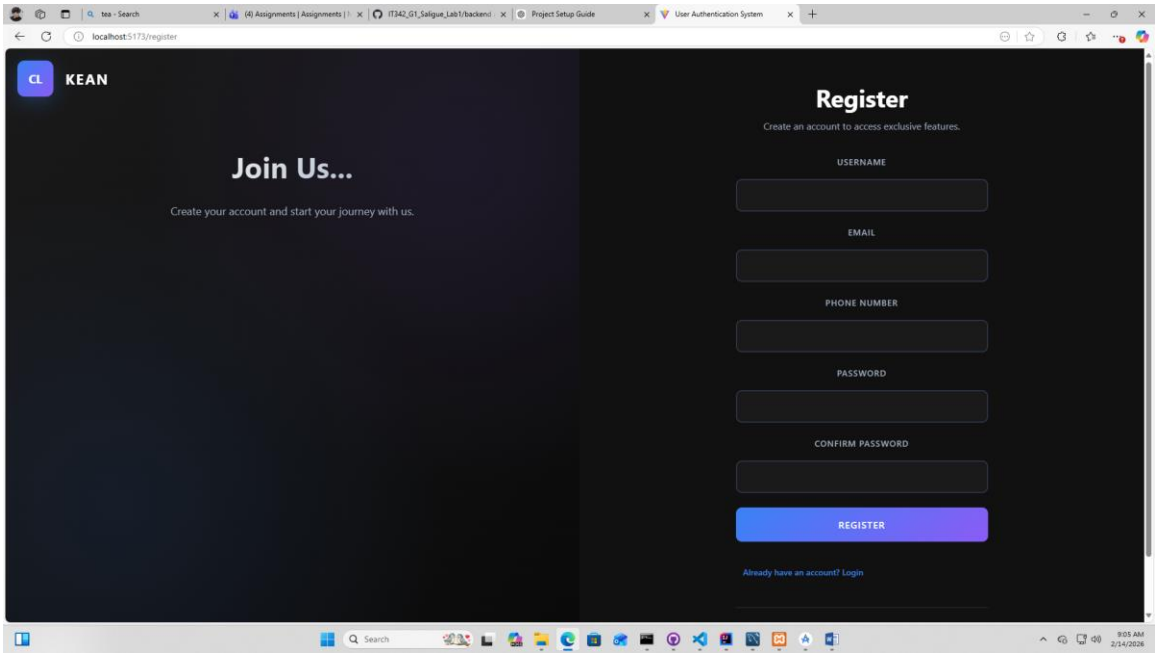
Supporting Materials

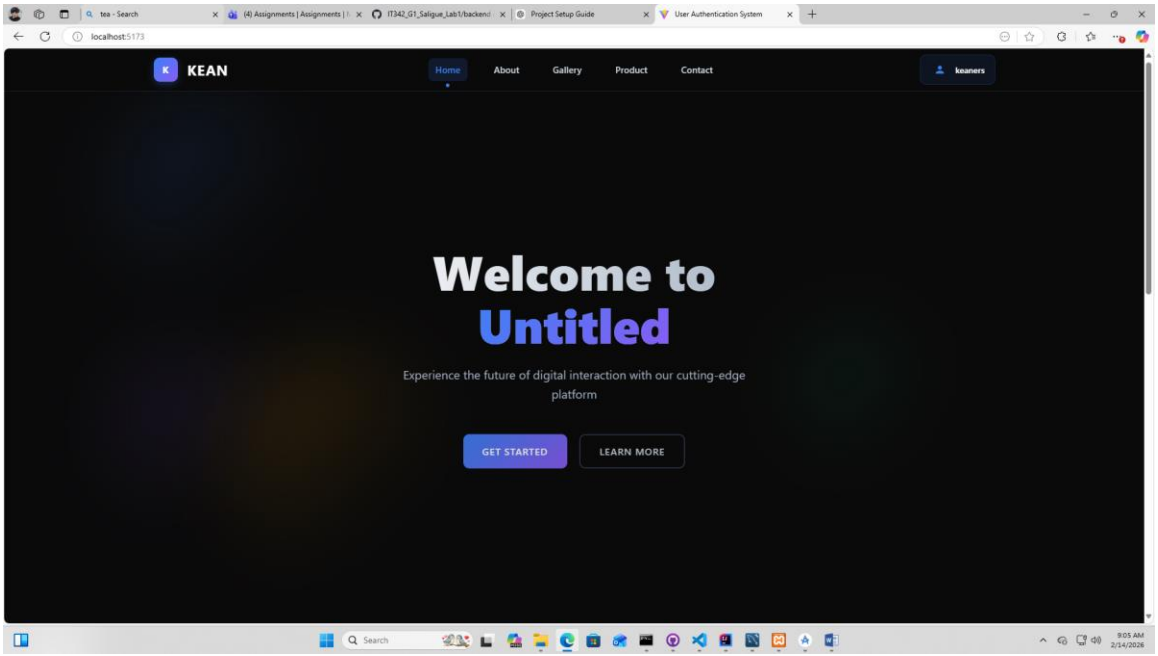
- System diagrams created using draw.io
- Development notes and planning documents
- Implementation guidelines for future phases

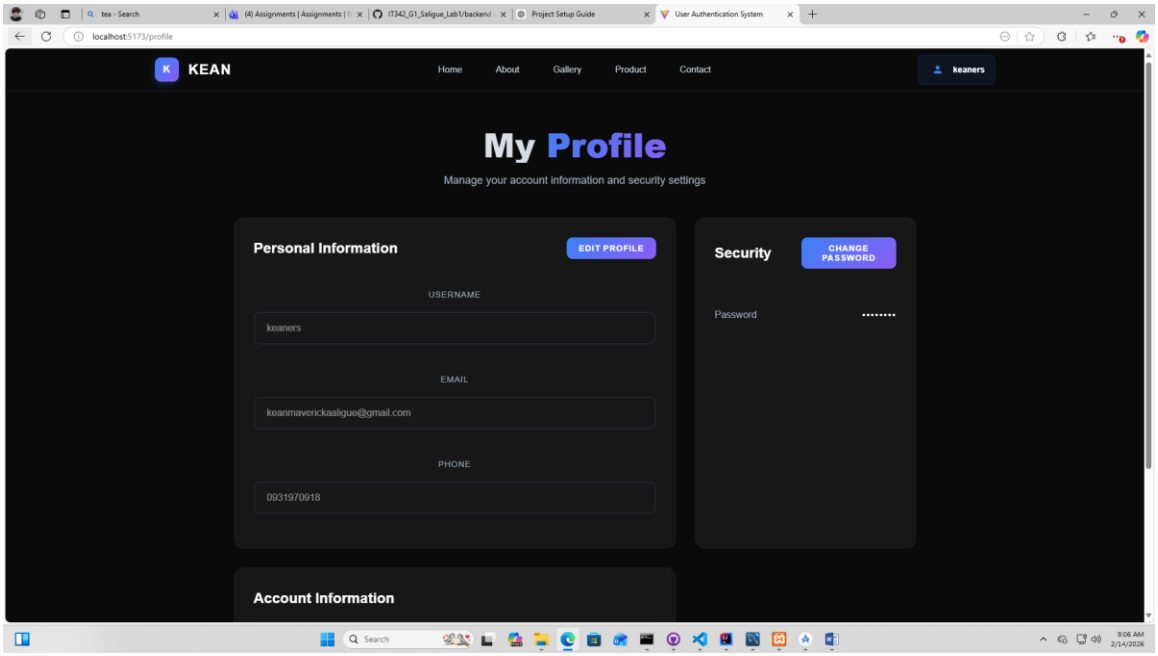
7. Screenshots

Web




KEAN
Join Us...
Create your account and start your journey with us.
Register
Create an account to access exclusive features.
USERNAME
EMAIL
PHONE NUMBER
PASSWORD
CONFIRM PASSWORD
REGISTER
Already have an account? Login


KEAN
Home
About
Gallery
Product
Contact
keaners
Welcome to
Untitled
Experience the future of digital interaction with our cutting-edge platform
GET STARTED
LEARN MORE


KEAN
Home
About
Gallery
Product
Contact
keaners
My Profile
Manage your account information and security settings
Personal Information
EDIT PROFILE
USERNAME
keaners
EMAIL
koanmaverickaaligue@gmail.com
PHONE
0931970918
Security
CHANGE PASSWORD
Password
Account Information