CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-Section SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: Mini App

Prepared By: Noel, John Emilio

Date of Submission: 2/9/2026

Version: 2

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this system is to provide a secure user authentication process that allows users to register, log in, log out, and view their profile. This document is intended for students, instructors, developers, and anyone who wants to understand how the system works.

1.2. Scope

The system will handle user registration, login authentication, logout, and basic profile viewing. It will not handle advanced features such as payment processing, admin management, or analytics. The system focuses only on user authentication and session handling.

1.3. Definitions, Acronyms, and Abbreviations

User – A person who uses the system
Guest User – A user who is not logged in
Authenticated User – A user who has successfully logged in
UI – User Interface
API – Application Programming Interface
Token – A security key used to identify a logged-in user

2. Overall Description

2.1. System Perspective

The system is a web-based application where the frontend (React UI) communicates with a backend server (Spring Boot API). The backend connects to a database to store and retrieve user information. Users interact with the system through a web browser.

2.2. User Classes and Characteristics

Guest User
- Can register and log in
- Has no access to protected pages

Authenticated User
- Can view their profile
- Can log out
- Has basic knowledge of using web applications

2.3. Operating Environment

Hardware: PC, laptop, or mobile device
Software: Web browser (Chrome, Edge, Firefox)
Frontend: React
Backend: Spring Boot
Database: MySQL or similar relational database

2.4. Assumptions and Dependencies

Users have access to the internet
The backend server is running and reachable
The database is available and properly configured
The system depends on external libraries for security and password encryption

3. System Features and Functional Requirements

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

3.1. Feature 1:

Description: User Registration
Functional Requirements:
- The system shall allow users to enter registration details
- The system shall validate user input
- The system shall store user data securely in the database

3.2. Feature 2:

Description: User Login
Functional Requirements:
- The system shall accept user login credentials
- The system shall verify credentials against stored data
- The system shall allow access only to authenticated users

3.3. Feature 3:

Description: User Logout
Functional Requirements:
- The system shall allow users to log out
- The system shall invalidate the user session or token
- The system shall redirect users to the login page

4. Non-Functional Requirements

Performance:

The system shall respond to user actions such as login, registration, and logout within a reasonable time under normal usage.

Security:

User passwords shall be securely encrypted, and only authenticated users shall be allowed to access protected features.

Usability:

The system shall have a simple and user-friendly interface that is easy to understand for first-time users.

Reliability:
The system shall operate consistently without unexpected crashes and handle errors properly.

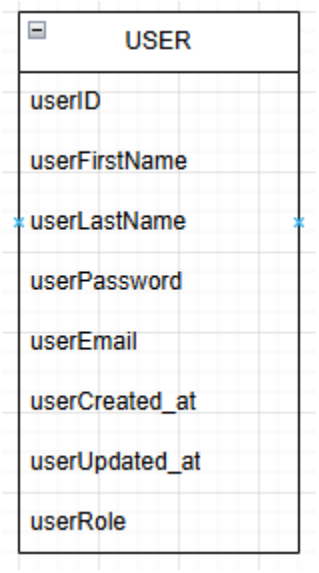
Availability:
The system shall be accessible to users whenever the server and internet connection are available.

Scalability:
The system shall support an increasing number of users without major performance issues.

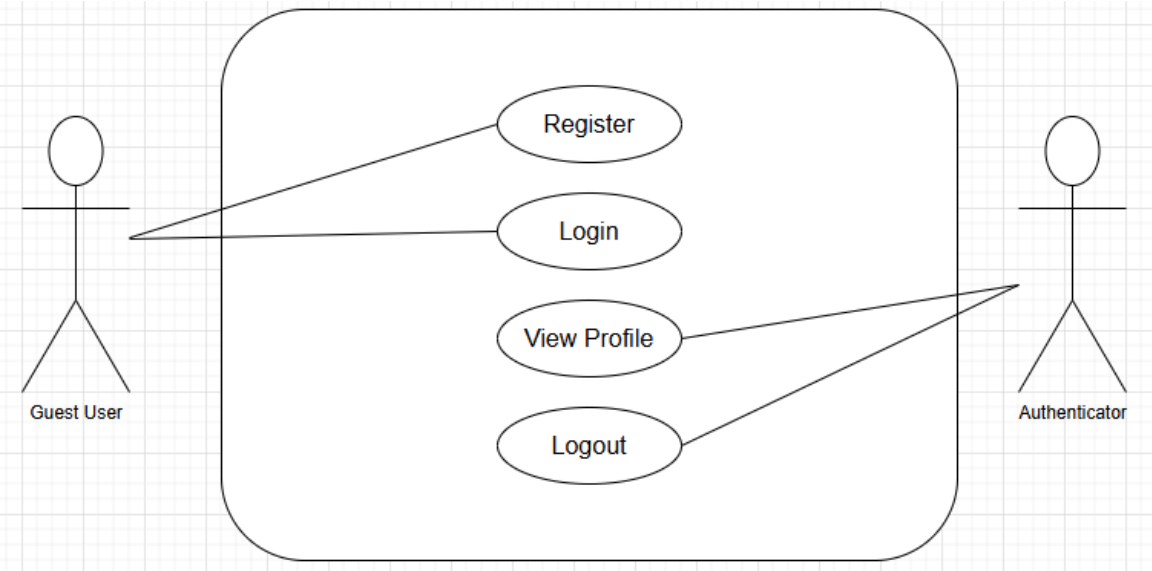
Maintainability:
The system shall be easy to update, debug, and maintain by developers.

5. System Models (Diagrams)

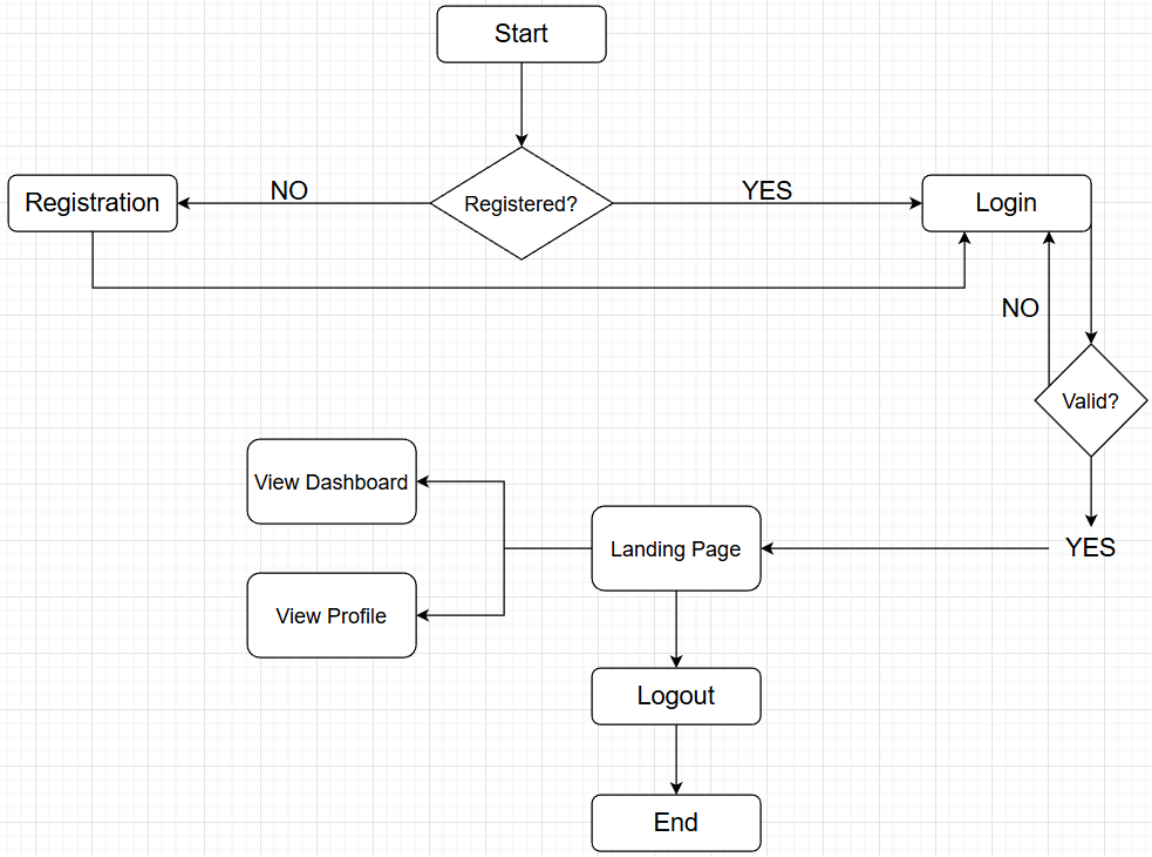
5.1. ERD



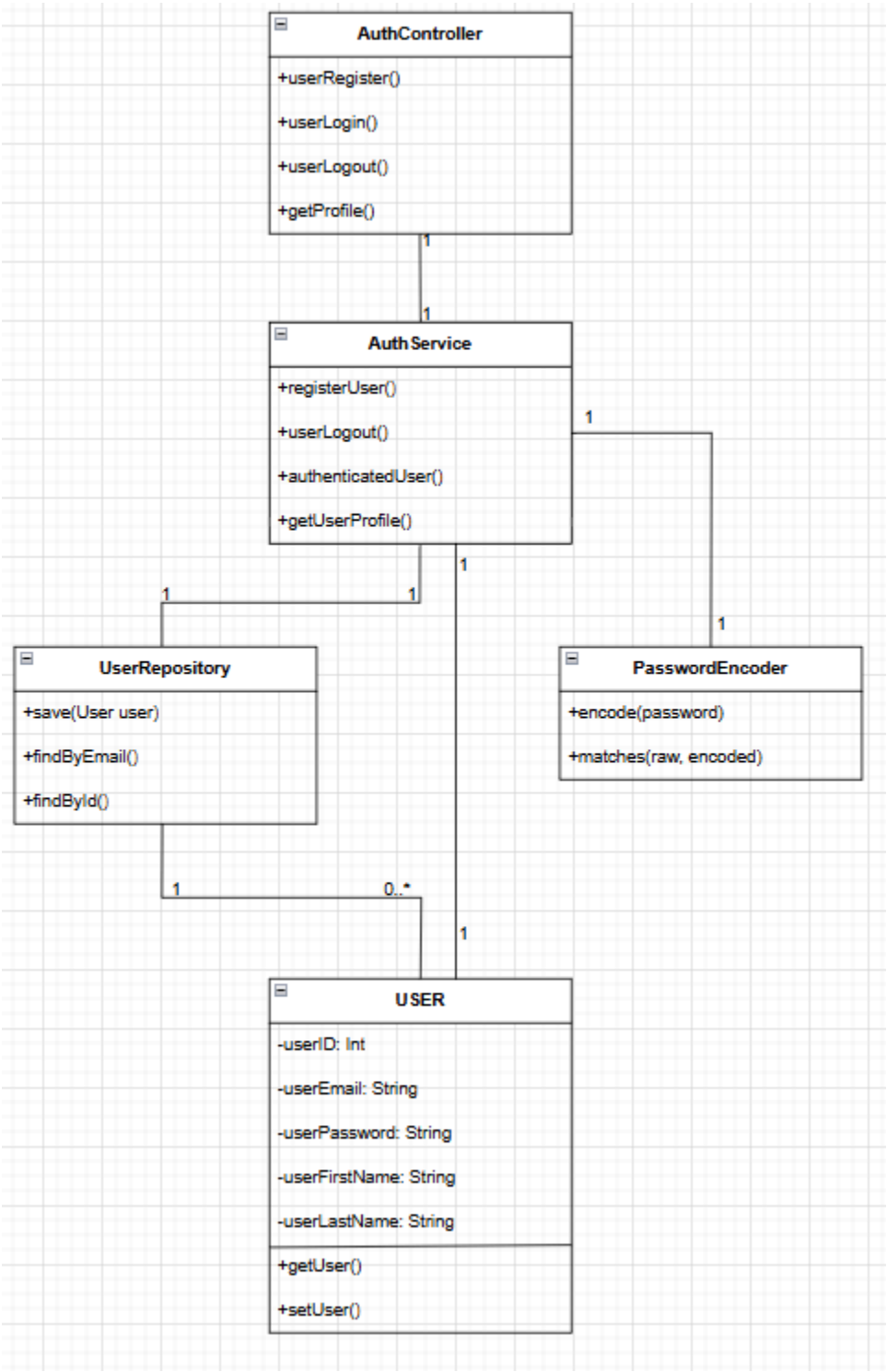
5.2. Use Case Diagram



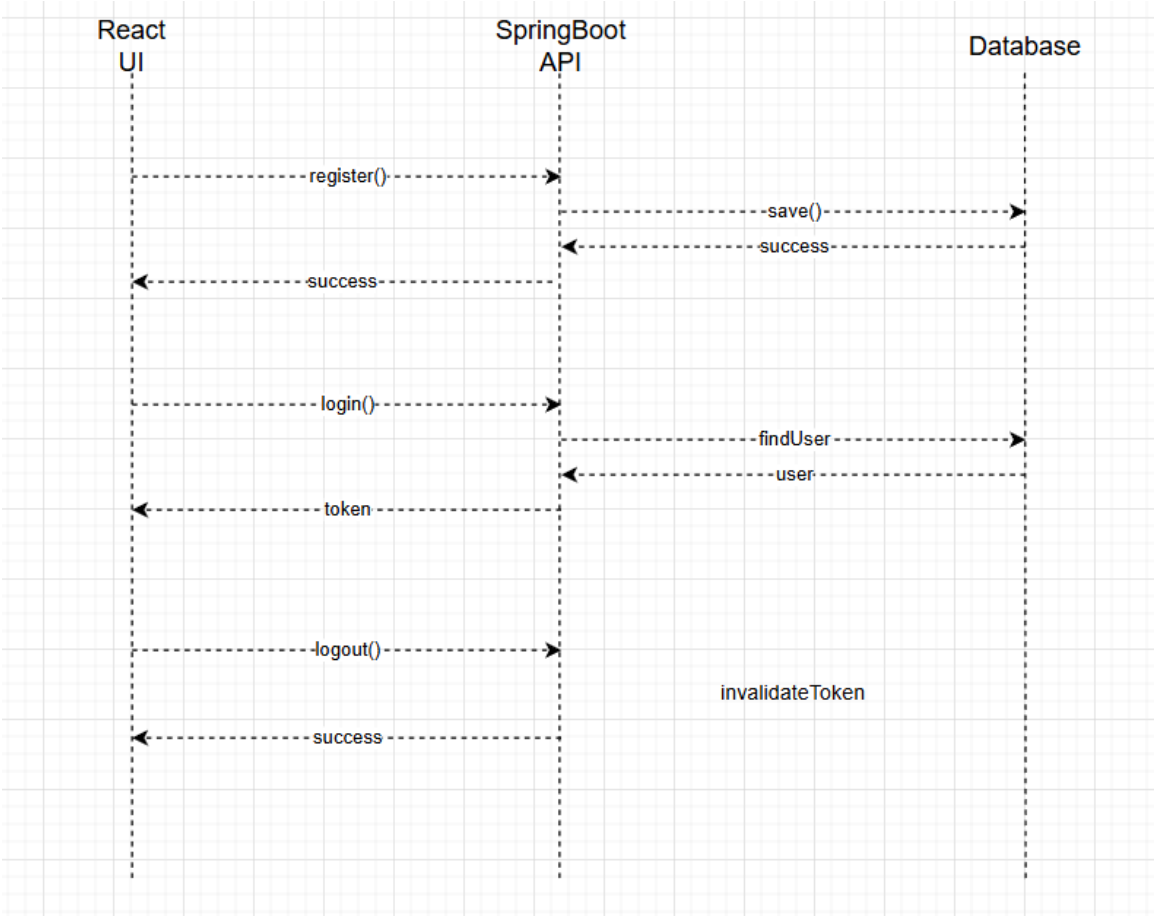
5.3. Activity Diagram



5.4. Class Diagram



5.5. Sequence Diagram



6. Appendices

Appendix A: Diagrams

This appendix includes the system diagrams used to explain the design and behavior of the system. These diagrams consist of the Use Case Diagram, Activity Diagram, Class Diagram, ERD, and Sequence Diagram.

Appendix B: Tools and Technologies

The system was designed using modern web development tools and technologies such as React for the user interface, Spring Boot for the backend API, and a relational database for data storage.

Appendix C: Assumptions

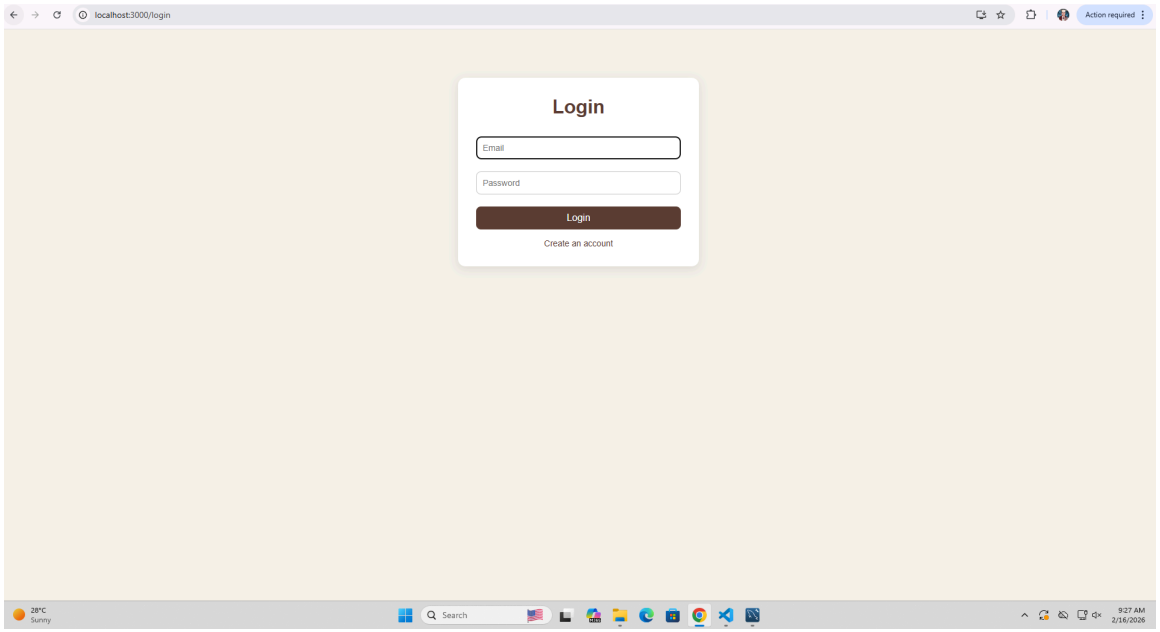
The system assumes that users have access to a stable internet connection and a compatible web browser. It also assumes that the backend server and database are properly configured and running.

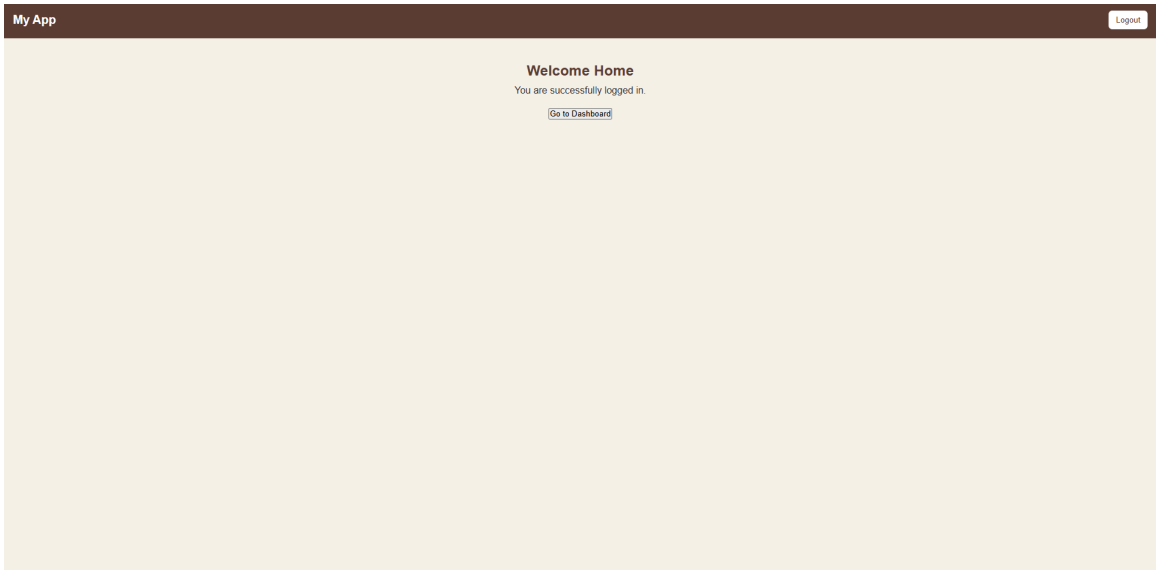
Appendix D: References

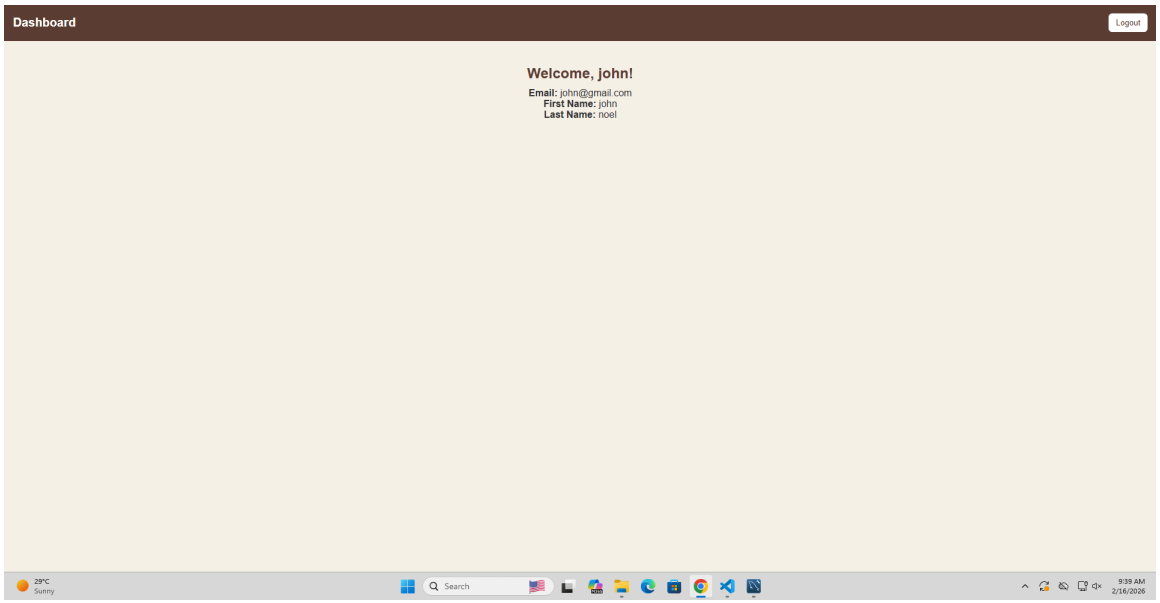
- Course lecture notes
- UML and software engineering reference materials
- Official documentation for React and Spring Boot

7. Screenshots

WEB




My App
Logout
Welcome Home
You are successfully logged in.
Go to Dashboard


Dashboard
Logout
Welcome, john!
Email: john@gmail.com
First Name: john
Last Name: noel
29°C
Sunny
Search
9:39 AM
2/16/2026