CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-Section SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: AccessHub

Prepared By: Reden Misael Relacion

Date of Submission: 02/02/2026

Version: version 1.0

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to define the functional and non-functional requirements for the "Mini App – User Registration & Authentication" system. This document serves as a blueprint for the implementation phase, detailing the system flow for user registration, secure login, and session management.

1.2. Scope

The system is a web-based application designed to demonstrate secure identity management. The boundaries of the system include:

- **User Registration:** Allowing new users to create an account by providing personal details.
- **Authentication:** Verifying user identity via email and password (Login).
- **Authorization:** Granting access to the protected Dashboard only to logged-in users.
- **Session Management:** Securely logging out and clearing user sessions.

1.3. Definitions, Acronyms, and Abbreviations

- **FRS:** Functional Requirements Specification.
- **JWT (JSON Web Token):** A standard method for securely transmitting information between parties as a JSON object, used here for maintaining user sessions.
- **BCrypt:** A password-hashing algorithm used to encrypt passwords before storage.
- **API:** Application Programming Interface.

2. Overall Description

2.1. System Perspective

This system operates as a standalone web application utilizing a Client-Server architecture.

- **Frontend:** React.js application serving as the User Interface.
- **Backend:** Spring Boot API handling business logic and security.

- **Database:** Relational database (MySQL) for storing user credentials.

2.2. User Classes and Characteristics

- **Guest User:** An unauthenticated user. They have limited access permissions and can only view the Login and Registration pages.

- **Authenticated User:** A user who has successfully logged in. They have full access permissions to view the Dashboard and use the Logout function.

2.3. Operating Environment

- **Client Side:** Modern Web Browser (Chrome, Edge, Firefox).

- **Server Side:** Java Development Kit (JDK) 17+, Spring Boot Framework.

- **Database:** MySQL.

- **Tools:** Draw.io (Diagramming), Postman (API Testing).

2.4. Assumptions and Dependencies

- It is assumed that the database server is running and accessible by the API.

- The system depends on the `Spring Security` library for handling authentication protocols.

- Passwords must be hashed before storage; plain text storage is prohibited.

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 a Guest User to create a new account by providing their personal details. The system validates the input and stores the password securely.

Functional Requirements:

- **REQ-1.1:** The system shall accept First Name, Last Name, Email, and Password as input.
- **REQ-1.2:** The system shall verify that the email address is not already registered in the database.
- **REQ-1.3:** The system must encrypt the user's password using BCrypt before saving it.
- **REQ-1.4:** Upon successful registration, the system shall redirect the user to the Login page.

3.2. Feature 2:

Authentication & Session Management Description: This feature handles the secure login process, dashboard access, and logout capabilities.

Functional Requirements:

- **REQ-2.1:** The system shall validate the entered email and password against the database records.
- **REQ-2.2:** If credentials are valid, the system shall generate a session token (JWT) and grant access.
- **REQ-2.3:** The system shall restrict access to the "Dashboard" page; if a user is not logged in, they are redirected to Login.
- **REQ-2.4:** The system shall provide a "Logout" button that clears the stored session/token and redirects the user to the Login page.

4. Non-Functional Requirements

- **Security:** User passwords must never be stored in plain text. API endpoints for the dashboard must be protected by an authentication filter.

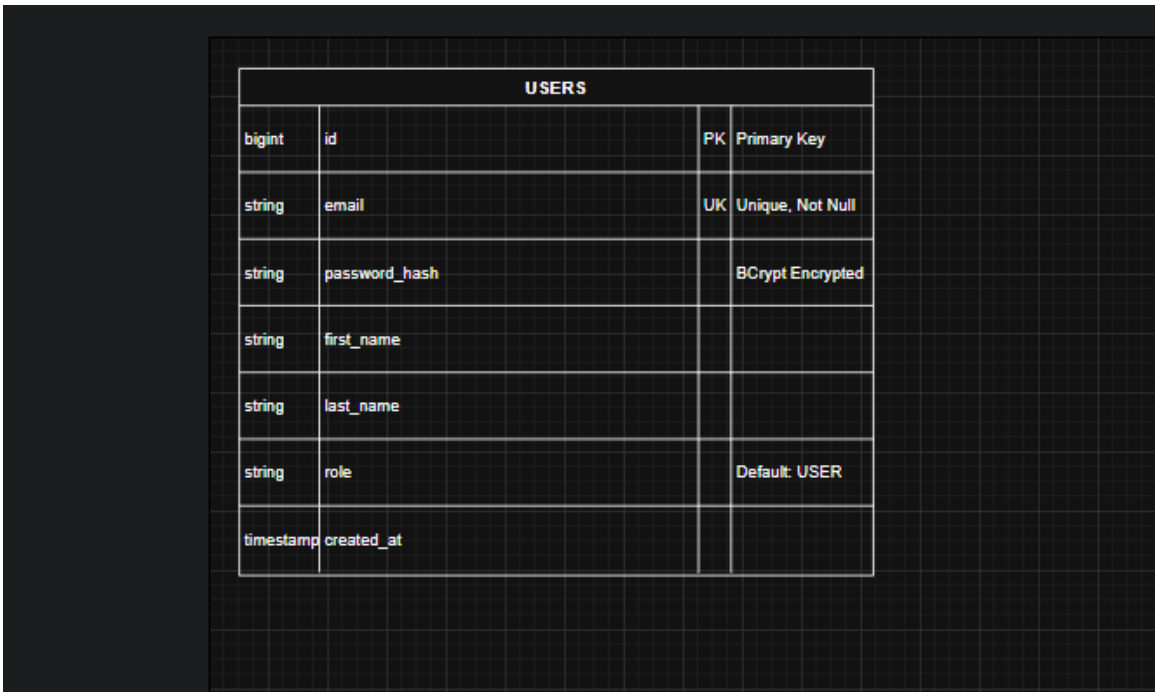
- **Performance:** Login and Registration transactions should complete within 2 seconds.

- **Usability:** The UI must be intuitive, clearly distinguishing between the "Register" and "Login" forms. Error messages (e.g., "Invalid Credentials") must be displayed clearly.

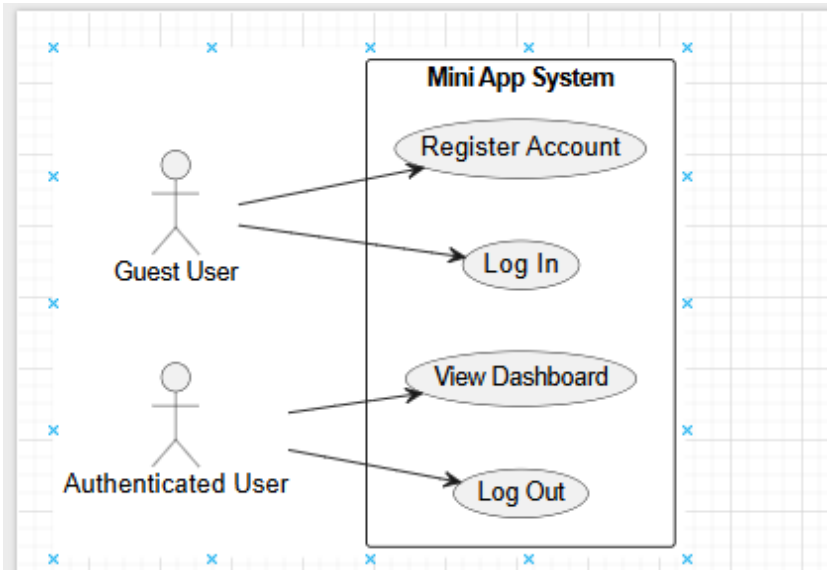
- **Reliability:** The system should handle database connection errors gracefully without crashing.

5. System Models (Diagrams)

5.1. ERD

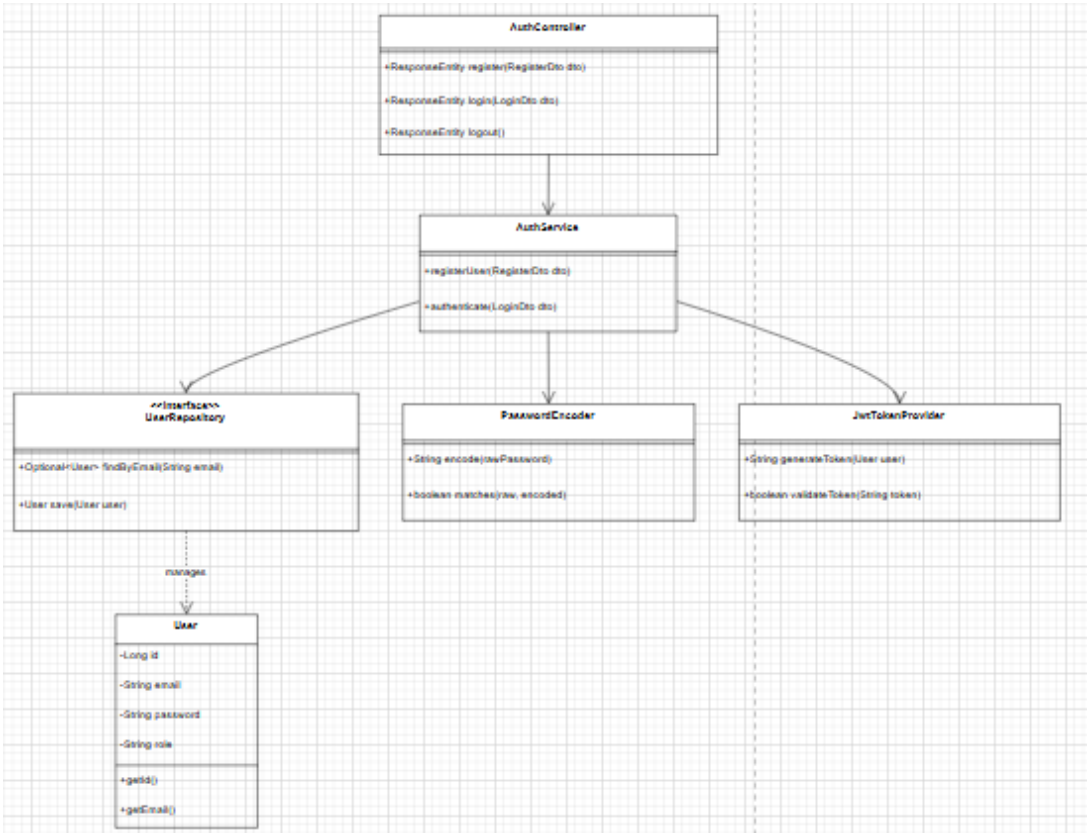


5.2. Use Case Diagram

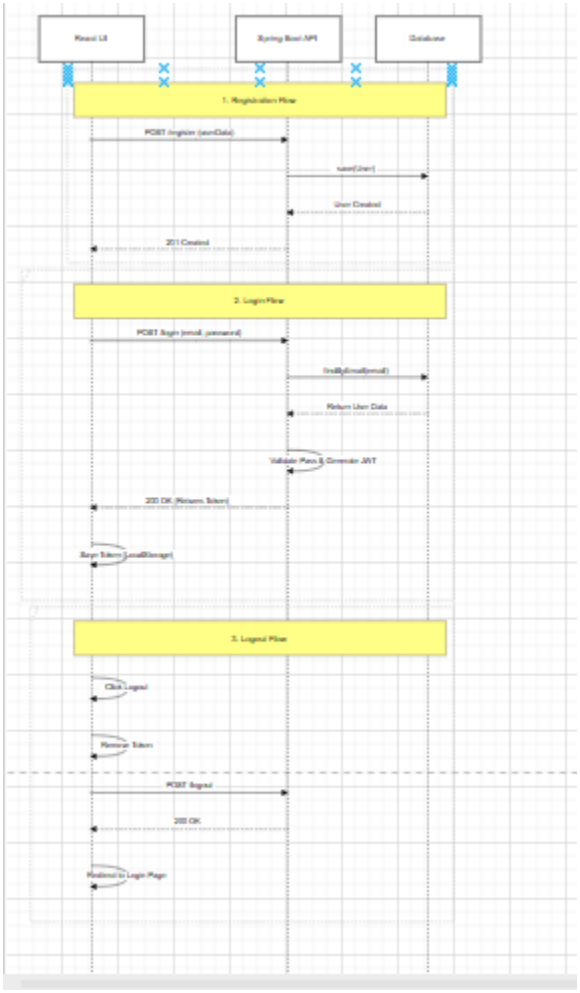


5.3. Activity Diagram

5.4. Class Diagram



5.5. Sequence Diagram



6. Appendices

- **Appendix A:** Laboratory Instruction Sheet (Mini App – User Registration & Authentication).

- **Appendix B:** Spring Security Documentation Reference.