CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-G5
SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION

(FRS)

Project Title: BayanLink: Barangay Services Portal

Prepared By: Ceniza, Jian Marc

Date of Submission: 02/13/2026

Version: 3

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this system is to provide a centralized digital platform for residents and officials of a Barangay to manage administrative actions and health-related services. It aims to replace manual, paper-based workflows or verbal announcement/spread of information with automated processes, specifically for official document issuing and health consultation scheduling.

1.2. Scope

The system will serve as an online portal where residents can manage their local government requests and interactions. It will facilitate user registration, secure login, and profile management. It will implement a transactional system for requesting official barangay documents and a separate module for booking medical appointments at the barangay health center. The system is limited to the administrative functions of a single barangay. It will not handle financial payments as to leave any payments face-to-face, and does not provide real-time medical diagnosis on the system, but rather focuses on the scheduling of health services with officials and medical professionals.

1.3. Definitions, Acronyms, and Abbreviations

List and define important terms used in this document.

- **BayanLink:** The official name of the integrated Barangay and Health Service Portal.
- **SIA:** System Integration and Architecture (the course for which this project is developed).
- **JPA:** Java Persistence API; used to manage relational data in Java applications.
- **Lombok:** A Java library that automatically plugs into the editor and builds tools to reduce boilerplate code.
- **Maven:** A build automation tool used primarily for Java projects to manage dependencies.
- **MySQL:** The relational database management system used to store user and transaction records.
- **RHU:** Rural Health Unit; refers to the local health center within the community.
- **Spring Boot:** The open-source Java-based framework used to create the stand-alone, production-grade application.

2. Overall Description

2.1. System Perspective

The system would be a basis of the community's large database of residents that can be inaccurate and inadequately updated with manual documentations. This system is an application that acts as a digital bridge between the residents and the local government unit (LGU). In a larger context, it serves as the primary digital interface for the **Barangay Management System**. The digital database will then be available to a larger scale, like a town the barangay is involved with, for updates on the residents' status. While it currently functions as a standalone system for a specific barangay, its architecture is designed to potentially integrate with higher-level municipal health systems or national ID databases in the future.

2.2. User Classes and Characteristics

Target Users:

- **Residents:** General members of the community. They are primary users who register to request official documents and book health center appointments. They generally possess basic digital literacy.
- **Barangay Staff/Officials:** Administrative personnel responsible for verifying resident information, approving/denying document requests, and posting community announcements.
- **Health Workers:** Nurses, midwives, or RHU staff who manage the clinic schedule, update service availability (e.g. vaccines or doctor check-ups), and confirm patient records.
- **System Admin:** Technical personnel responsible for maintaining the database, managing user roles, and ensuring system security.

2.3. Operating Environment

Hardware:

- **Server-side:** A local machine or cloud server with at least 4GB RAM and 20GB of storage to host the Spring Boot JAR and MySQL database.
- **Client-side:** Any device with a modern web browser (Smartphone, Tablet, or PC).

Software:

- **Runtime Environment:** Java Runtime Environment (JRE) 17.
- **Database:** MySQL Server 8.0+.
- **Browser Support:** Google Chrome, Mozilla Firefox, or Microsoft Edge.

Development Tools: Spring Initializr, Android Studio, IntelliJ IDEA, MySQL Workbench, and Visual Studio Code.

2.4. Assumptions and Dependencies

List any assumptions and external dependencies that may affect the system.

Assumptions:

- Users have access to a stable internet connection to access the portal.
- Residents possess a valid government-issued ID for account verification during registration.
- The Barangay Hall has at least one dedicated computer to manage incoming requests.

Dependencies:

- **External Libraries:** The system depends on the Spring Boot Framework and Maven repositories for dependency management.
    - Spring Security
    - Data JPA
    - Spring Web
    - Spring Web Services
    - Spring Dev Tools
    - MySQL Connector
    - Lombok
- **Database Availability:** The application requires a constantly running MySQL service to perform transactional functions. The database can also be modified to an online database, such as supabase.
- **Spring Security:** The "Login/Logout" and "Profile" features are dependent on the correct configuration of the Spring Security module for authentication.

3. System Features and Functional Requirements

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

3.1. Feature 1: User Authentication and Access Control

Description: This module serves as the security gateway for the system, handling identity verification and session management using Spring Security. It ensures that only authorized individuals can access the portal's internal features.
Functional Requirements:

- **Multi-Role Registration:** The system allows new residents to register for an account, while administrative and health accounts are managed by the System Admin.
- **Secure Unified Login:** The system shall authenticate all user classes (Resident, Staff, Health Worker) via a single login interface using email/username and BCrypt-encrypted passwords.
- **Role-Based Access Control:** The system shall automatically redirect users to their specific dashboard based on their role (e.g., Staff to Admin Panel, Residents to Home Dashboard).
- **Secure Session Management:** The system shall maintain a secure user session upon login and provide a **Logout** function to invalidate the session and clear all authentication tokens and recent user data.
- **Account Recovery:** The system shall provide a "Forgot Password" facility to allow users to securely reset their credentials via their registered email.

3.2. Feature 2: User Profile Management

Description: A dedicated module for all user classes to manage their personal and professional information. It acts as the "Digital ID" that stores data required for all other transactional modules.

Functional Requirements:

- **Personal Information Display:** Every logged-in user shall have access to a personal Profile Page displaying their name, contact details, and account creation date.
- **Information Update:** The system shall allow users to update their contact numbers, home addresses, and profile pictures.
- **Role-Specific Data View:** The profile shall display unique fields based on user type: Residents see "Verification Status," while Staff and Health Workers see "Position/Department" and "Assigned Office".
- **Credential Security:** The system shall require the user to input their current password as a verification step before allowing changes to their profile data.
- **Activity Logs:** The profile page shall include a "Recent Activity" section where users can view their own last interactions with the system.

3.3. Feature 3: Integrated E-Service Dashboard

Description: The home page/dashboard that displays the Community Announcements and provides quick access to transactional modules based on

the user's role.
Functional Requirements:

- **Dynamic Announcement Feed:** The system shall display real-time barangay notices and health alerts on the main dashboard for all users.
- **Module Navigation:** The system shall display shortcut buttons to the Document and Health modules for Residents, and the Management for Staff.
- **For Residents:** It features prominent "Quick Action" buttons: [Request a Document] and [Book an Appointment].
- **For Staff/Officials:** It features monitoring widgets: [Pending Document Requests Count] and [Today's Patient Schedule].

3.4. Feature 4: Digital Document Transaction Module

Description: Digitizes the application and issuance of barangay certifications.
Functional Requirements:

- **Resident View (Request Page):** A user-friendly form where residents select the type of document needed (e.g., Barangay Clearance, Certificate of Indigency). It includes a file-upload section for required identification and a "Request History" table to track the progress of their applications in real-time.
- **Staff/Official View (Management Page):** A centralized administrative dashboard that lists all incoming document requests sorted by urgency or date. Staff can view uploaded requirements, update the status (e.g., "In Progress" or "Ready for Pickup"), and generate a digital log of issued certifications.

3.5. Feature 5: Community Scheduling Module

Description: Manages the health center's logistics and patient appointments. As well as the scheduling for the barangay Captain’s availability for counseling.
Functional Requirements:

- **Resident View (Scheduling Page):** An interactive calendar interface where residents can browse available time slots for various health services or barangay captain’s counseling. Upon selection, the system generates an appointment confirmation with a reference number for the resident to present during their visit.
- **Staff/Official View (Monitoring Page):** A master schedule view for health workers and nurses to see the list of confirmed patients for the

day. It allows them to manage clinic capacity by blocking off holidays or full slots and marking patients as "Attended" or "Cancelled" to keep accurate health center records.

4. Non-Functional Requirements

- Security and Privacy:
  - **Data Encryption**: All user passwords must be hashed using the **BCrypt** password-hashing function before being stored in the MySQL database.
  - **Authorization (RBAC)**: The system shall enforce Role-Based Access Control, ensuring that only users with "Admin" or "Health Worker" roles can access transactional records that are not their own.
  - **Session Integrity**: The system shall implement secure session management via Spring Security to prevent unauthorized access through session hijacking.
  - **Privacy Compliance**: All personal and medical data must be handled in accordance with local data privacy standards, ensuring that residents' sensitive information is not exposed to other users.
- Performance and Scalability
  - **Response Latency**: The system shall load the Main Dashboard and the Universal Profile page within 5 seconds under standard local network conditions.
  - **Concurrent Capacity**: The backend (Spring Boot) shall be configured to handle at least 100 concurrent users without a measurable degradation in transaction processing speed.
  - **Database Optimization**: All search queries for document requests and health appointments must be indexed in MySQL to ensure quick retrieval as the record count grows over time.
- Usability
  - **Responsive Interface**: The UI shall be optimized for both desktop and mobile web browsers to accommodate residents who primarily access the portal via smartphones.
  - **User Guidance**: The system shall provide clear error messages and "Success" notifications (e.g., "Appointment Booked Successfully") to guide users through the transactional processes.
  - **Simplified Navigation**: The transition between the Auth, Profile, and Transactional modules shall be intuitive, requiring no more than two clicks to reach any major feature from the dashboard.
- Reliability and Availability

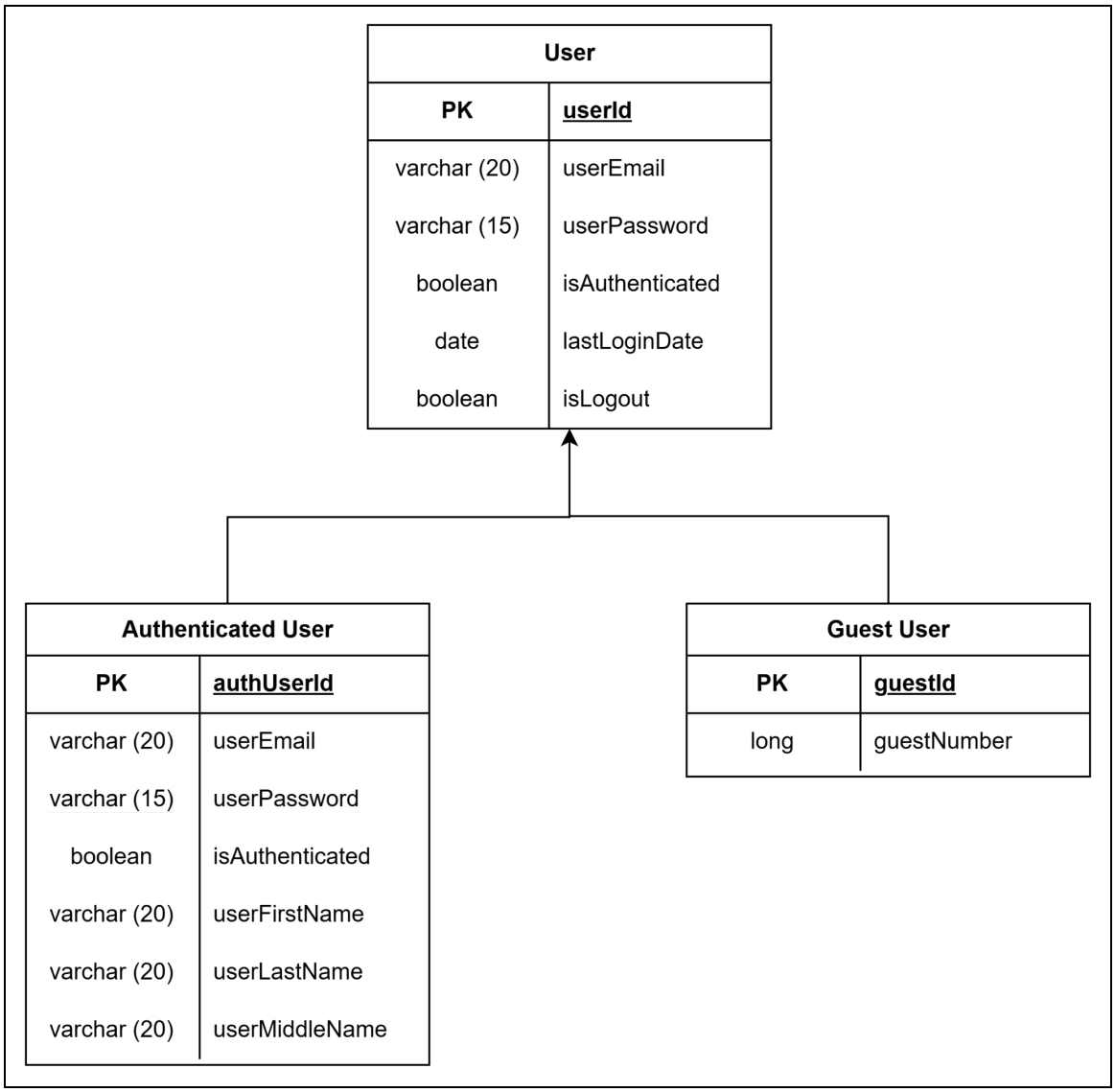
- **System Uptime**: The application shall target **99% availability** during the barangay's official operating hours (e.g., 8:00 AM to 5:00 PM).
- **Data Persistence**: All transactions must follow the **ACID** (Atomicity, Consistency, Isolation, Durability) properties provided by Spring Data JPA to ensure that document requests are never lost during a system crash.
- **Fault Tolerance**: The system shall implement basic error-handling pages (e.g., custom 404 or 500 error pages) to prevent the application from exposing raw stack traces to the user.

■ Maintainability

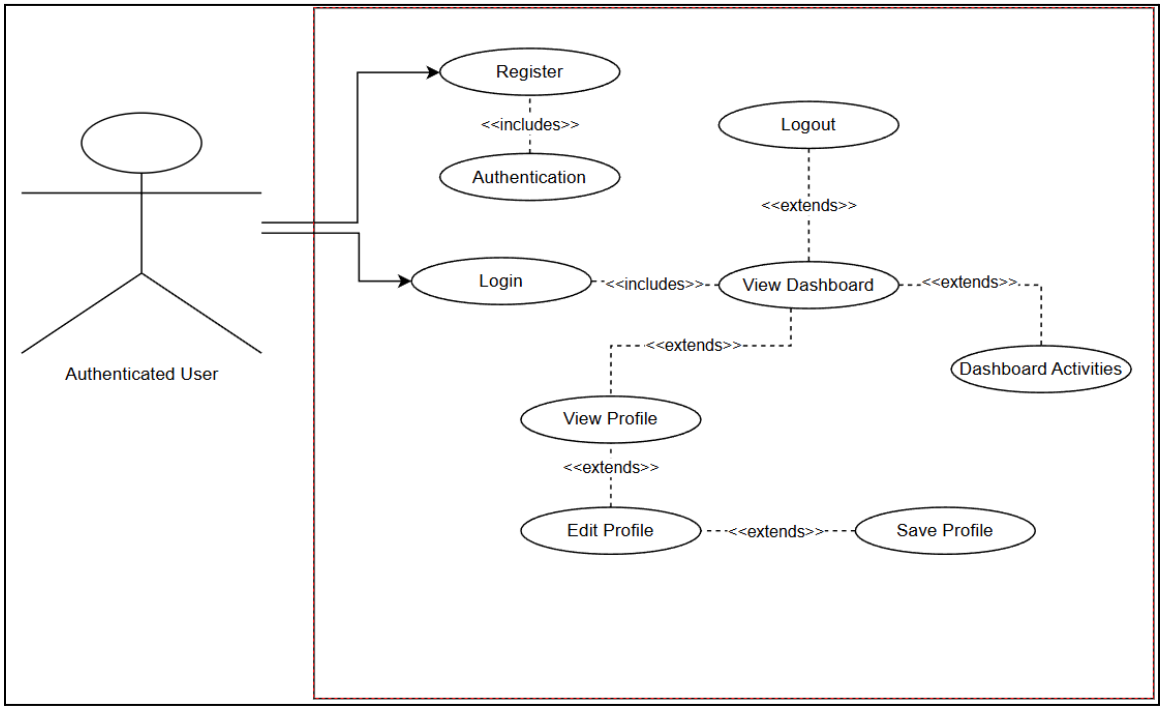
- **Modular Codebase**: By separating the **Auth Module** and **User/Profile Module**, the system shall allow for independent updates to security protocols without affecting the business logic of the transactions.
- **Automated Documentation**: The project shall utilize **Swagger/OpenAPI** or clear JavaDoc comments to allow future student developers to understand the API endpoints for the Document and Health modules.

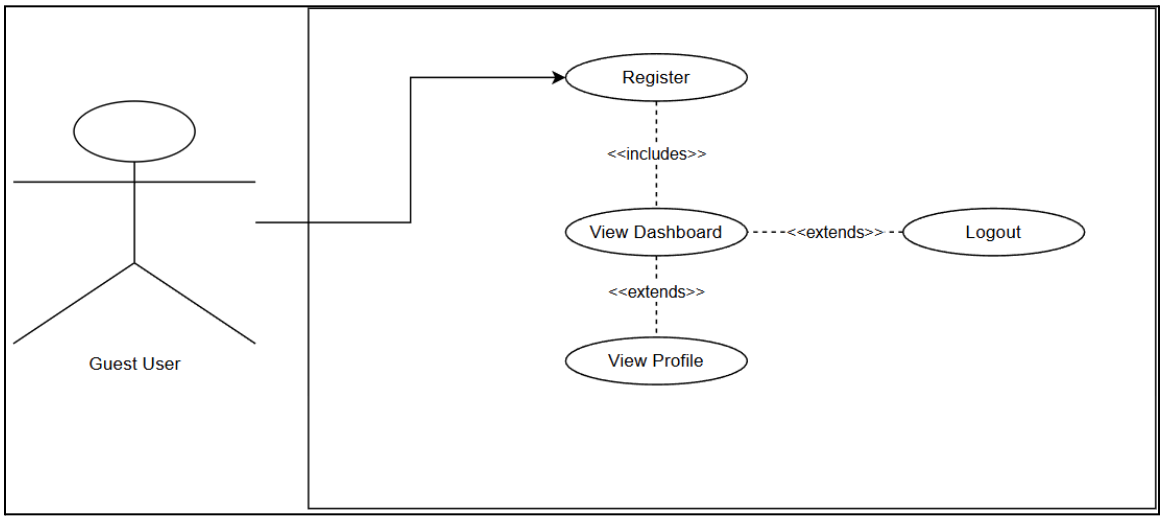
5. System Models (Diagrams)

5.1. ERD

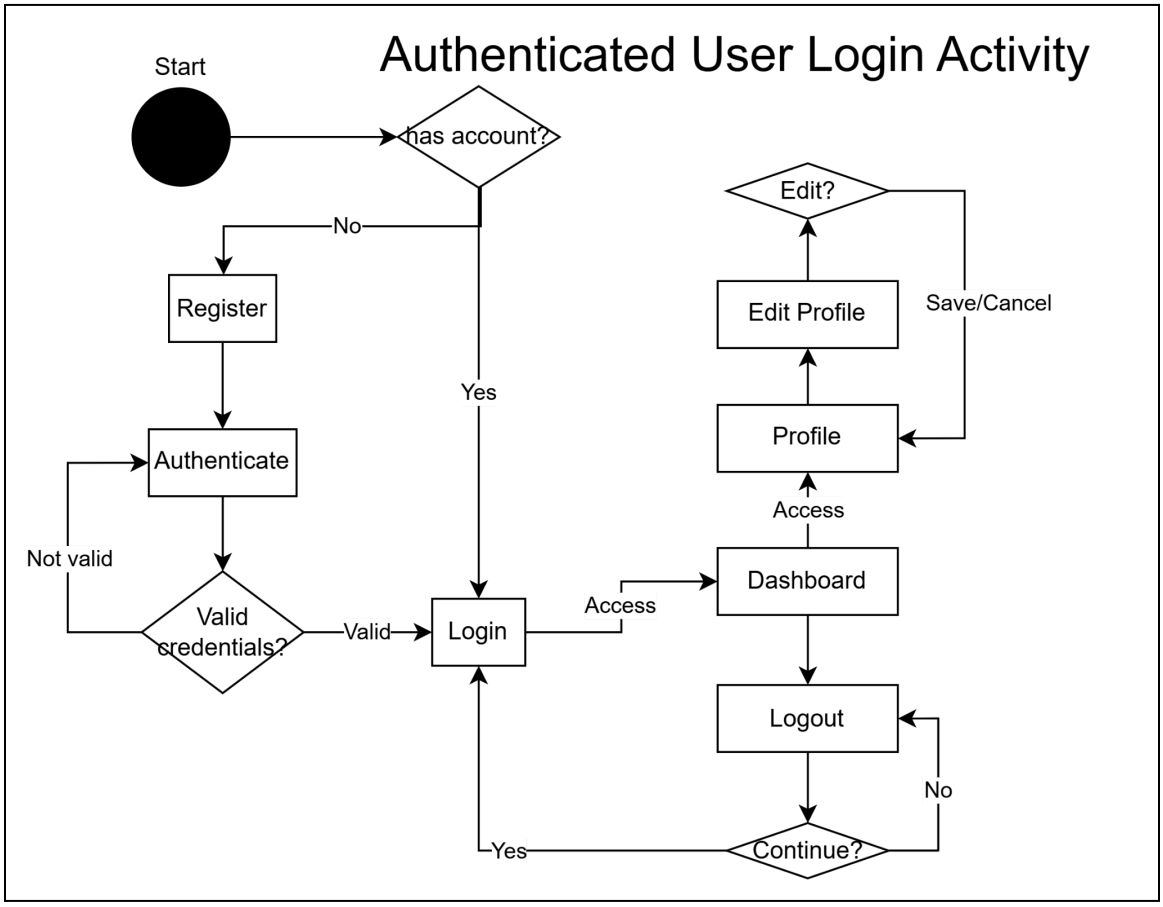


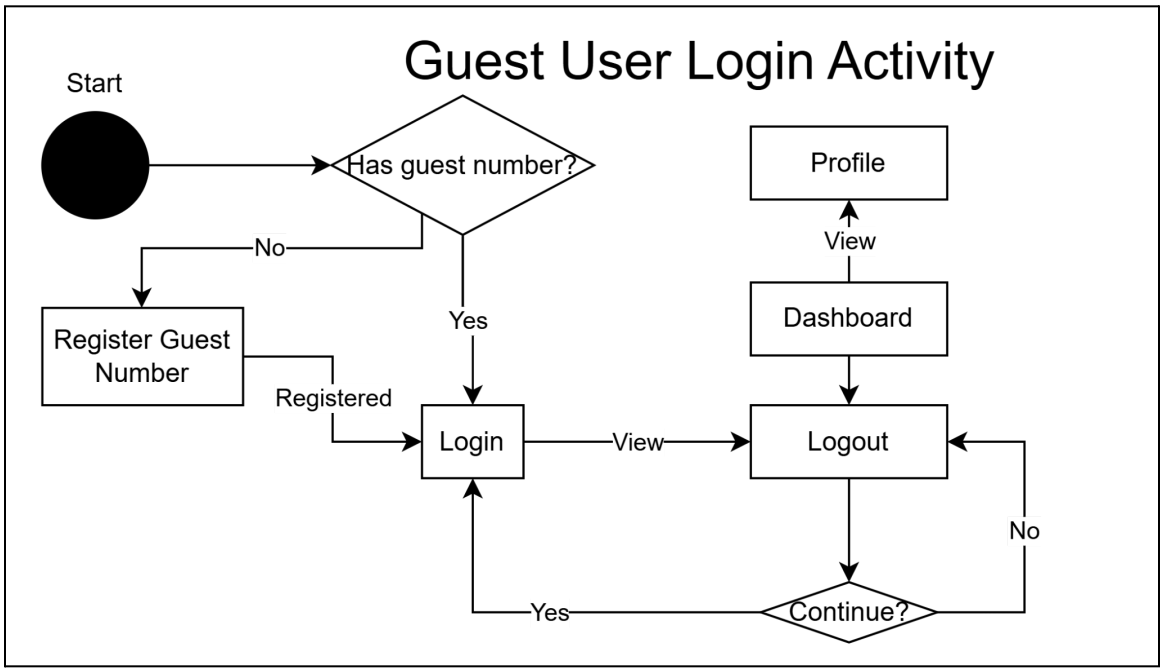
5.2. Use Case Diagram



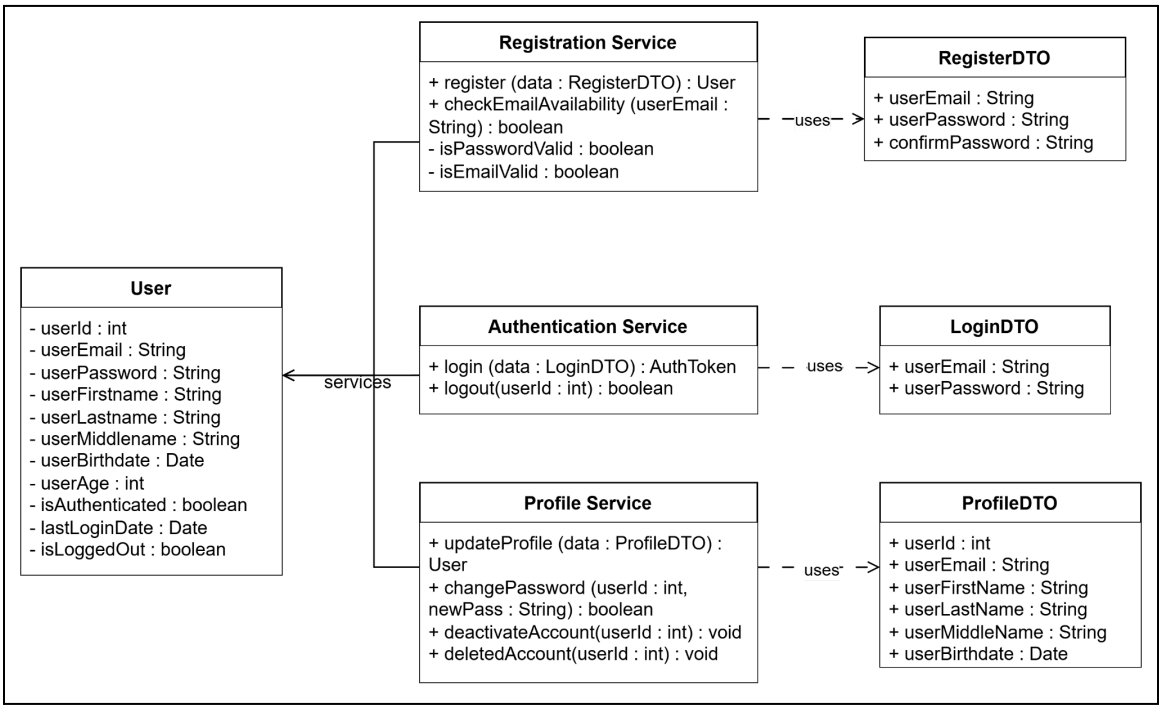


5.3. Activity Diagram

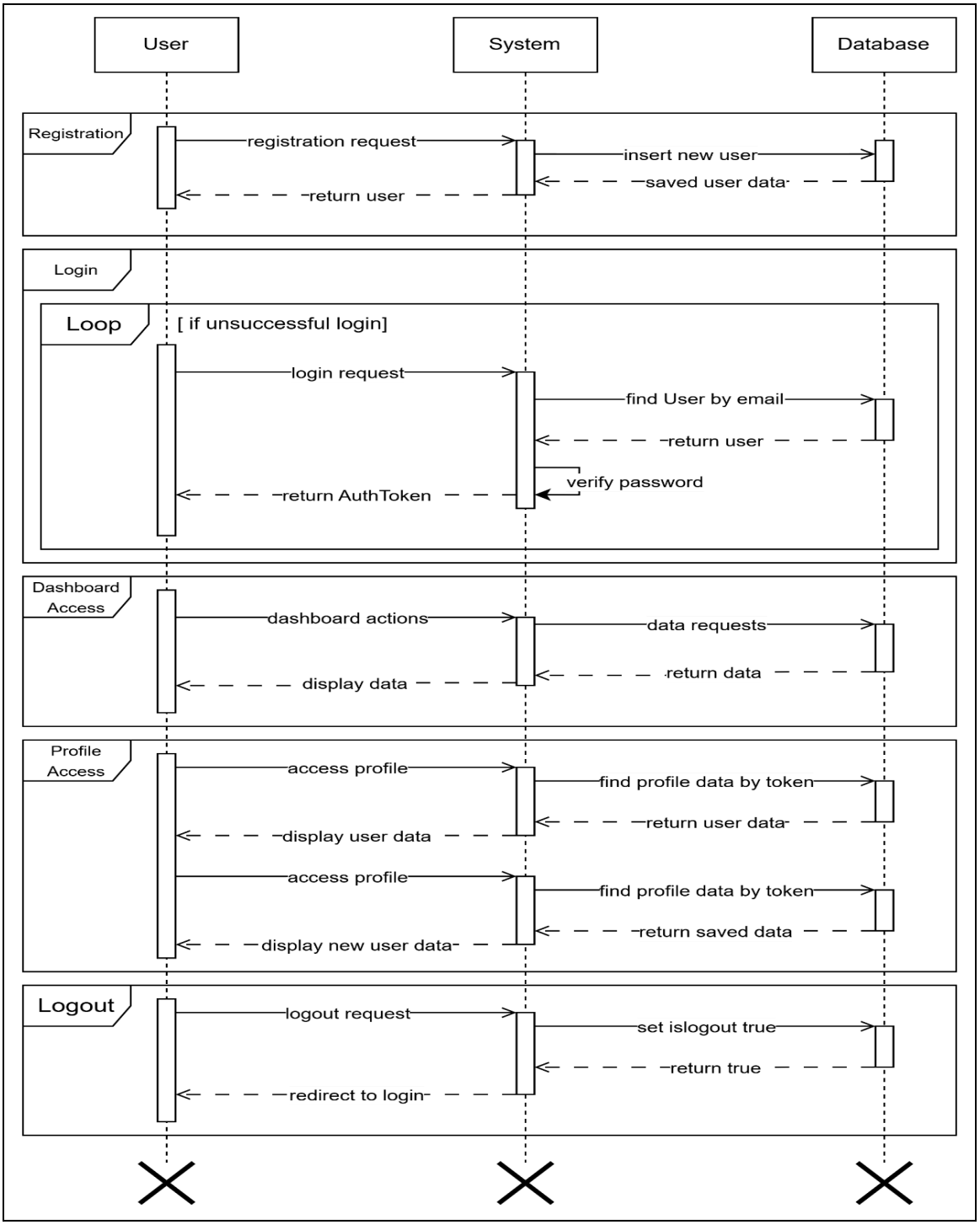




5.4. Class Diagram

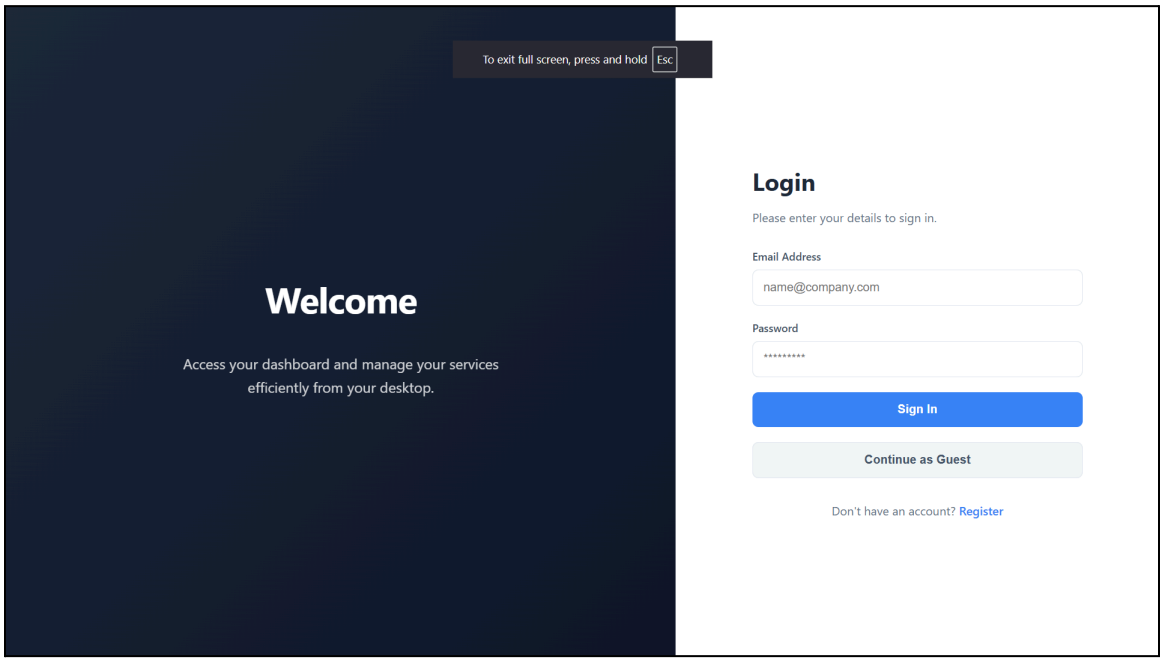


5.5. Sequence Diagram

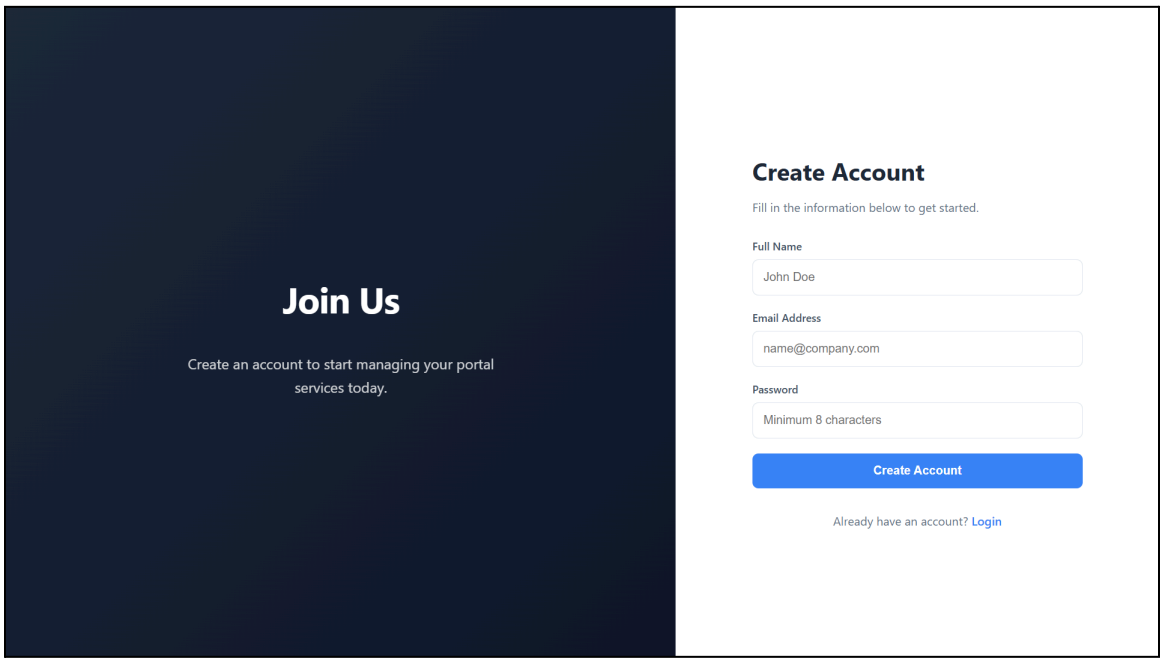


6. Appendices

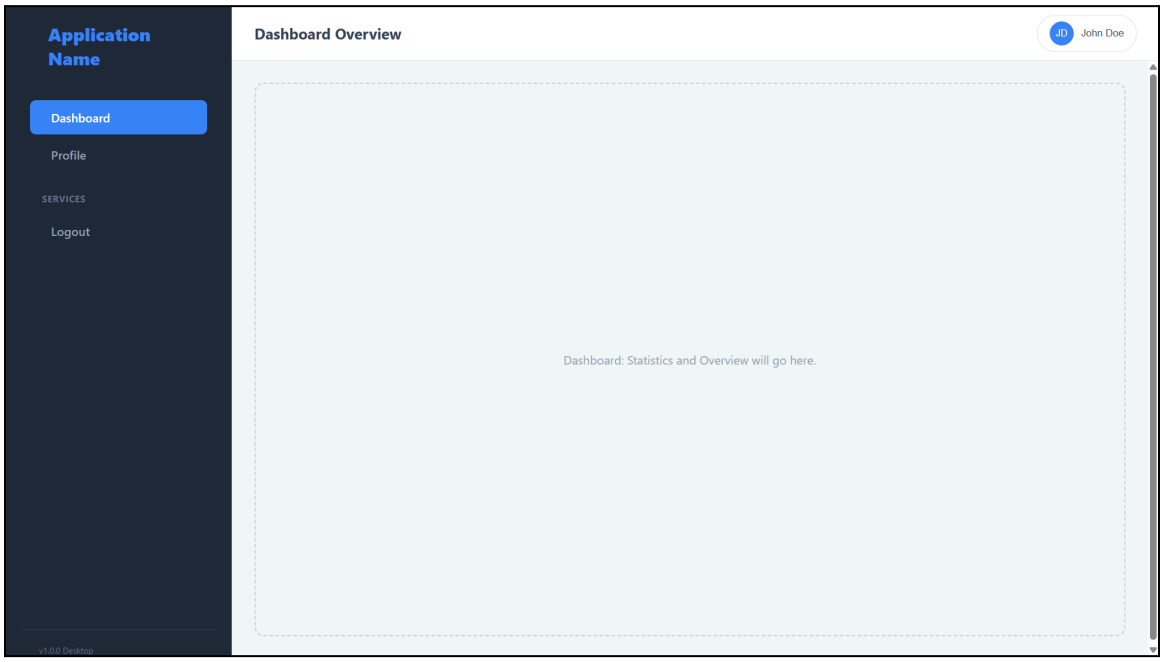
Login Page



Register Page



Dashboard



Profile Page