CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-Section
SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: Mini App - User Registration & Authentication

Prepared By: Christian Kyle B. Tapales

Date of Submission: February 9, 2025

Version: 2

Table of Contents

1. Introduction

1.1. Purpose

To clearly define the functional and non-functional requirements for the **Mini App - User Registration & Authentication** system. It serves as a formal agreement between the stakeholders and the development team, outlining the system's objectives, features, and constraints. The intended audience includes system developers, testers, project managers, and business stakeholders who need a comprehensive understanding of the system's expected behavior.

1.2. Scope

The system, **Mini App - User Registration & Authentication**, is primarily designed to manage user identities and access control. The scope of this system include:

- **User Registration**: Allowing new users to create an account by providing necessary information.
- **User Authentication (Login/Logout)**: Verifying the identity of returning users and managing their session access.
- **Password Management**: Functionality for users to securely reset or change their passwords.
- **Data Validation**: Ensuring that all input during registration and authentication meets defined security and format requirements.

1.3. Definitions, Acronyms, and Abbreviations

Term/Acronym	Definition
FRS	An acronym for Functional Requirements Specification.
Mini App	The target application being built, which will integrate the User Registration & Authentication system.
User	Any individual who successfully registers and uses the Mini App.
Authentication	The process of verifying a user's identity (e.g., through a username and password).
Registration	The process by which a new user creates an account within the system.
System	The User Registration & Authentication component.

2. Overall Description

2.1. System Perspective

The User Registration & Authentication System will be an embedded component or subsystem within a larger Mini App framework. It is the initial and crucial entry point for all users, handling identity verification and session management. It functions as a security and access layer, decoupled from the Mini App's core business logic but providing validated user IDs and tokens for the rest of the application to operate. It is not a standalone application but integrates with the Mini App's frontend (user interface) and backend (database/API services).

2.2. User Classes and Characteristics

The primary and currently sole user class is the **Registered User**.

Characteristics: Individuals who require access to the Mini App. They must be able to perform basic computer operations, navigate a web/mobile interface, and provide personal information (e.g., email address, password) accurately during registration and login. They expect a seamless, secure, and intuitive experience.

Needs: To register quickly, log in securely, retrieve/reset a forgotten password, and maintain a protected account session.

2.3. Operating Environment

The system will operate in a multi-tiered architecture:

Client Side (Frontend): Accessible via modern web browsers (Chrome, Firefox, Safari, Edge) on desktop and mobile devices. A potential future native mobile application (iOS/Android) should also be considered in the design.

Server Side (Backend): Will run on a scalable, secure cloud platform (e.g., AWS, GCP, Azure). It will utilize a web server (e.g., Node.js, Apache, Nginx) and communicate with the database via secured API endpoints.

Database: A relational (SQL) or non-relational (NoSQL) database is required to store user credentials, profiles, and associated metadata.

2.4. Assumptions and Dependencies

A-1: Internet Connectivity: It is assumed that all users have reliable access to the internet to use the system.

A-2: Unique Identifiers: All users will have a unique email address or username that serves as their primary identifier.

D-1: Security Standards: The system is dependent on following industry-standard security and encryption protocols (e.g., hashing passwords with salts, using HTTPS).

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 new user to create an account within the Mini App system by providing required personal information and a secure password. It is the initial gateway to the application.

Functional Requirements:

- The System **SHALL** present a registration form requesting, at minimum: Email Address (as the unique identifier), Password, and Password Confirmation.

- The System **SHALL** validate the format of the Email Address to ensure it is a legitimate email structure.

- The System **SHALL** enforce a minimum password complexity rule (e.g., minimum 8 characters, at least one uppercase, one lowercase, one number, and one special character).

3.2. **Feature 2:** User Authentication (Login/Logout)

Description: This feature provides the mechanism for returning users to securely verify their identity to gain access to the Mini App, and a method to securely terminate their session.

Functional Requirements:

- The System **SHALL** present a login interface requesting the user's Email Address (or Username) and Password.

- The System **SHALL** verify the provided credentials against the stored user data using the secure hashing algorithm.

- The System **SHALL** grant the user a secure, session-based token (e.g., JWT) upon successful login.

4. Non-Functional Requirements

- **Performance**: Response time for login/registration (e.g., within 2 seconds).
- **Reliability**: High availability (e.g., 99.9% uptime).
- **Usability**: Clear error messages, intuitive flow, mobile responsiveness.

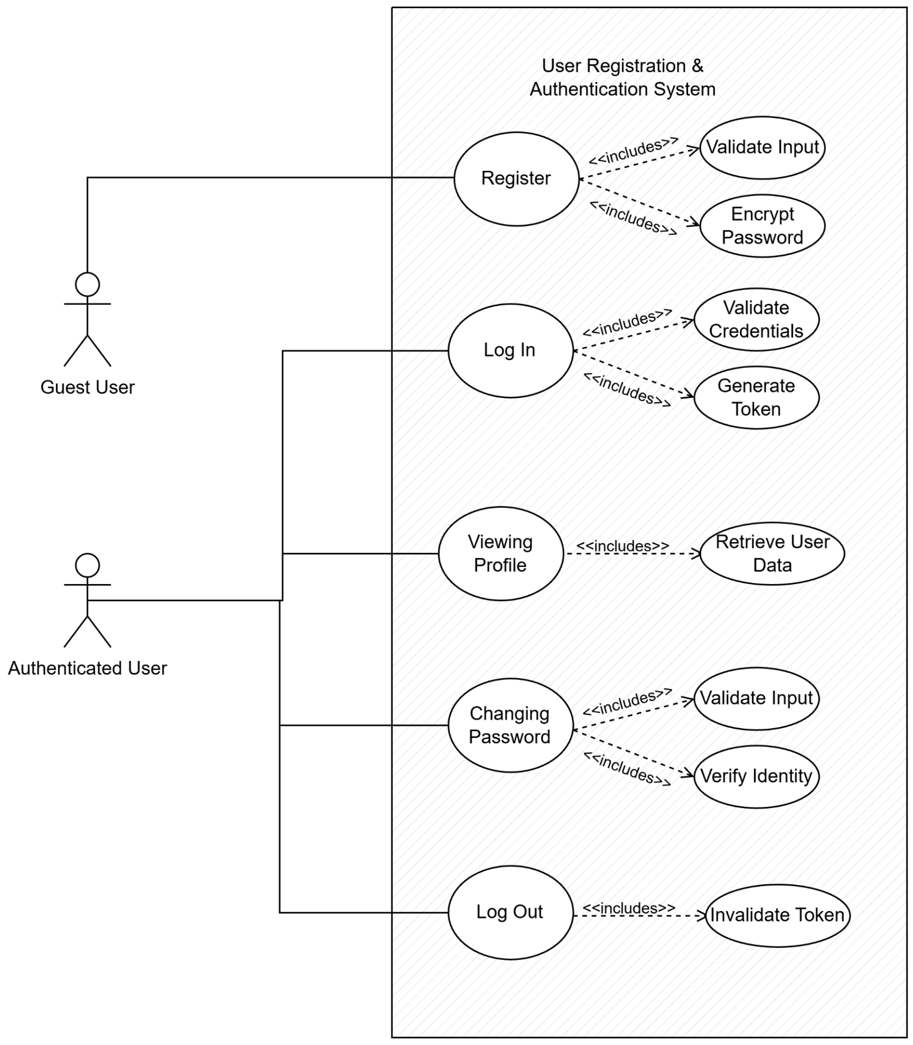
5. System Models (Diagrams)

Insert the necessary diagrams for the system:

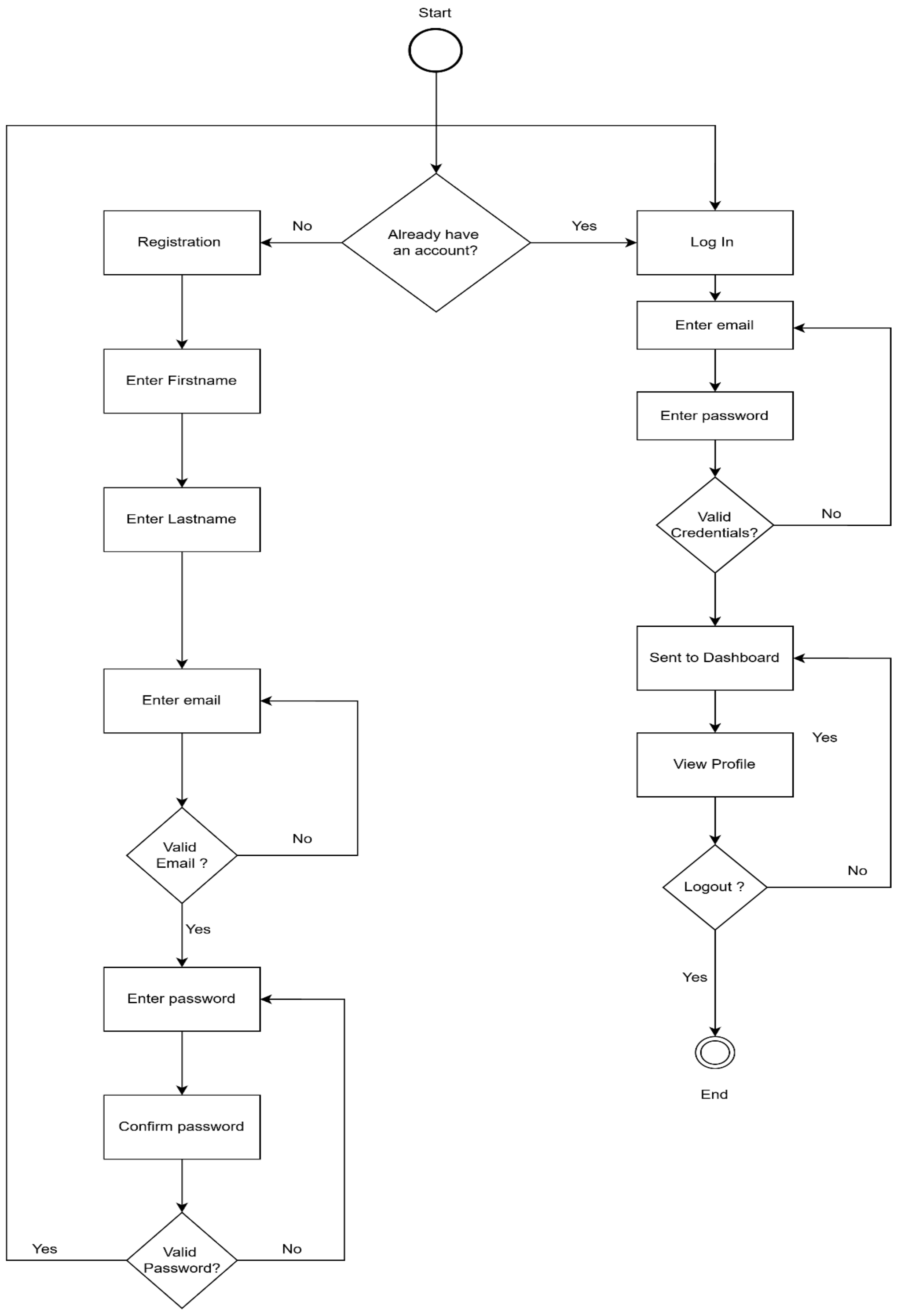
5.1. ERD

User	
PK	**<u>userID</u>**
	firstName
	lastName
	email
	password

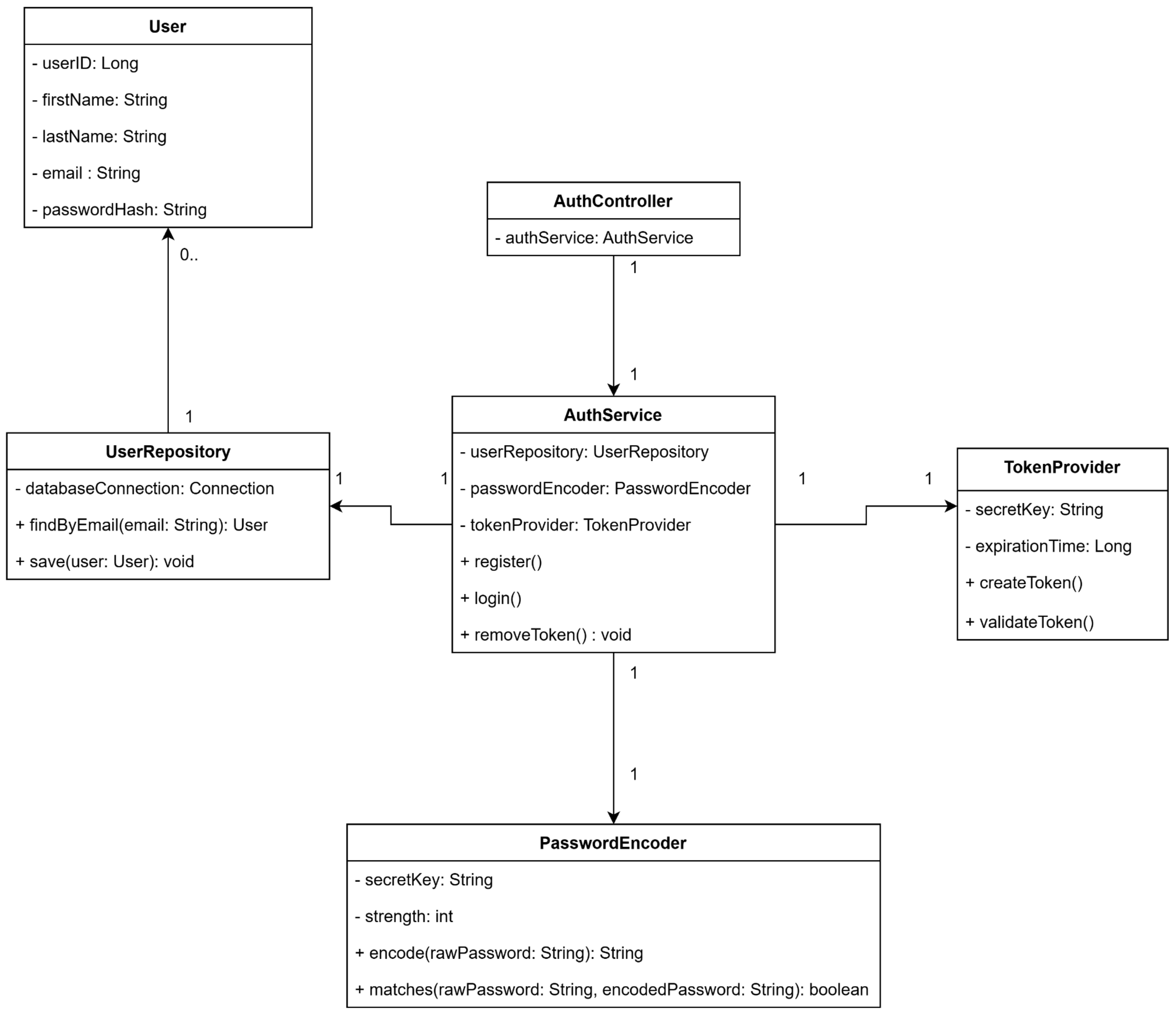
5.2. Use Case Diagram



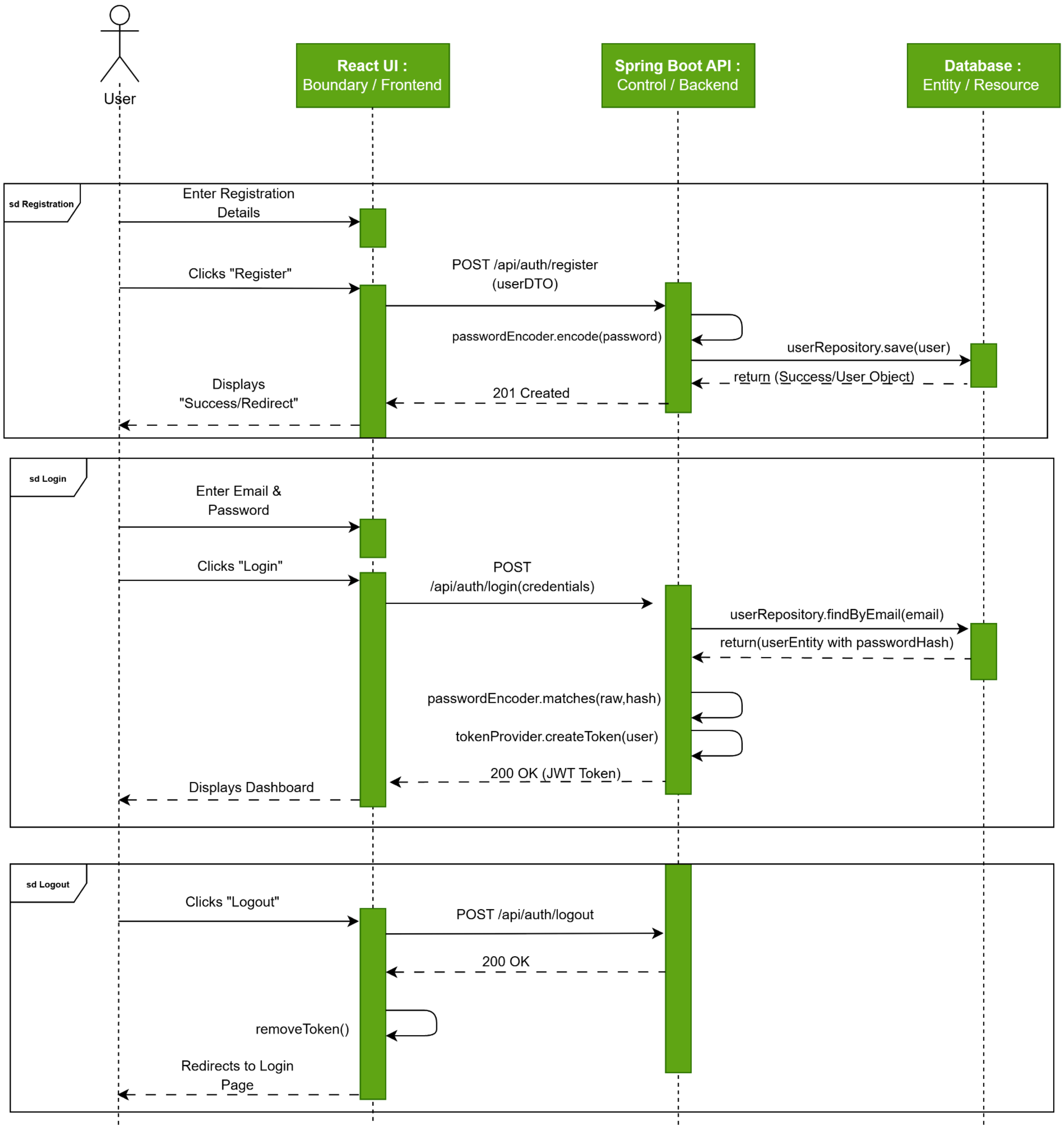
5.3. Activity Diagram



5.4. Class Diagram



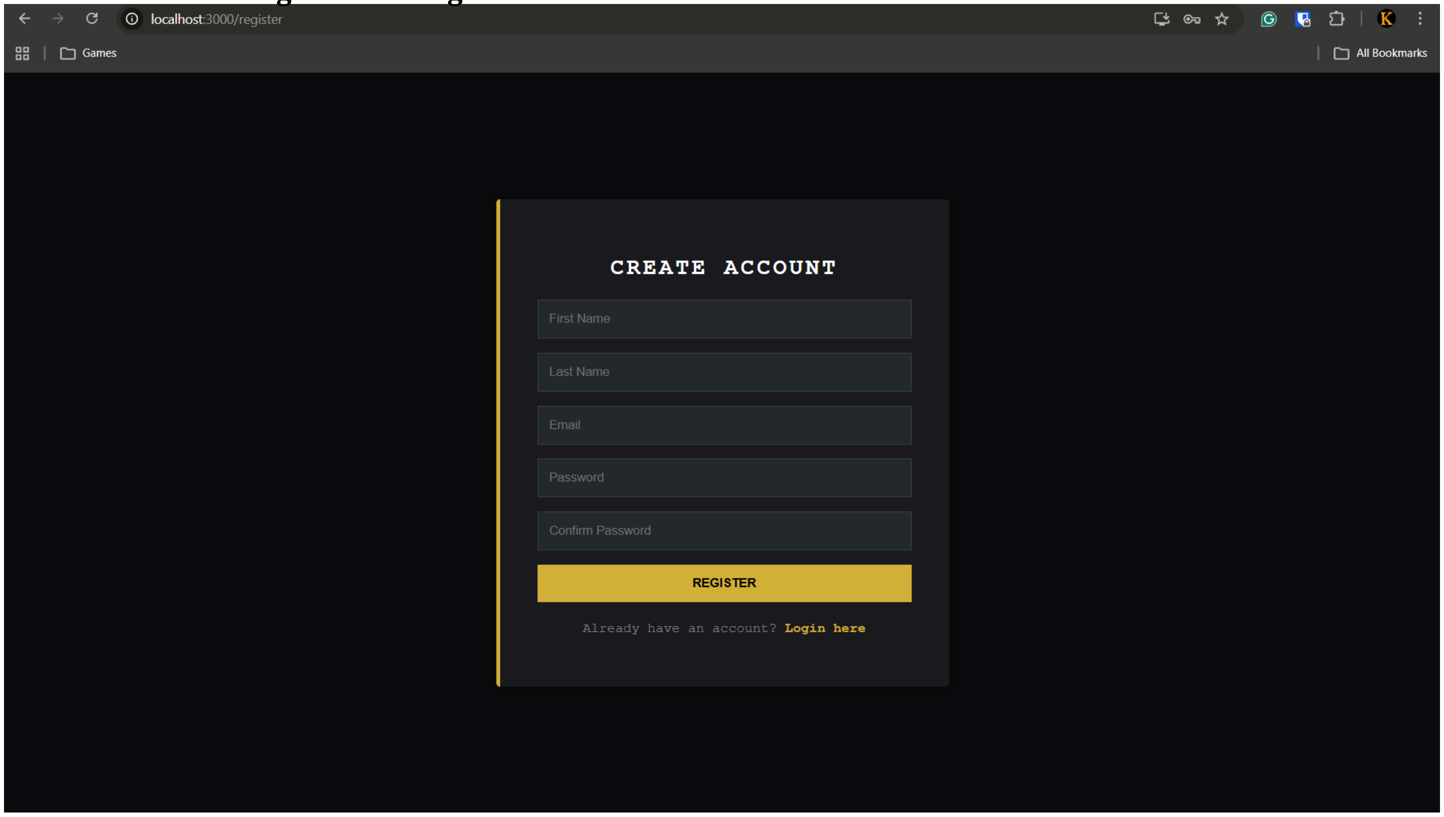
5.5. Sequence Diagram


User
React UI : Boundary / Frontend
Spring Boot API : Control / Backend
Database : Entity / Resource
sd Registration
Enter Registration Details
Clicks "Register"
POST /api/auth/register (userDTO)
passwordEncoder.encode(password)
userRepository.save(user)
return (Success/User Object)
201 Created
Displays "Success/Redirect"
sd Login
Enter Email & Password
Clicks "Login"
POST /api/auth/login(credentials)
userRepository.findByEmail(email)
return(userEntity with passwordHash)
passwordEncoder.matches(raw,hash)
tokenProvider.createToken(user)
200 OK (JWT Token)
Displays Dashboard
sd Logout
Clicks "Logout"
POST /api/auth/logout
200 OK
removeToken()
Redirects to Login Page

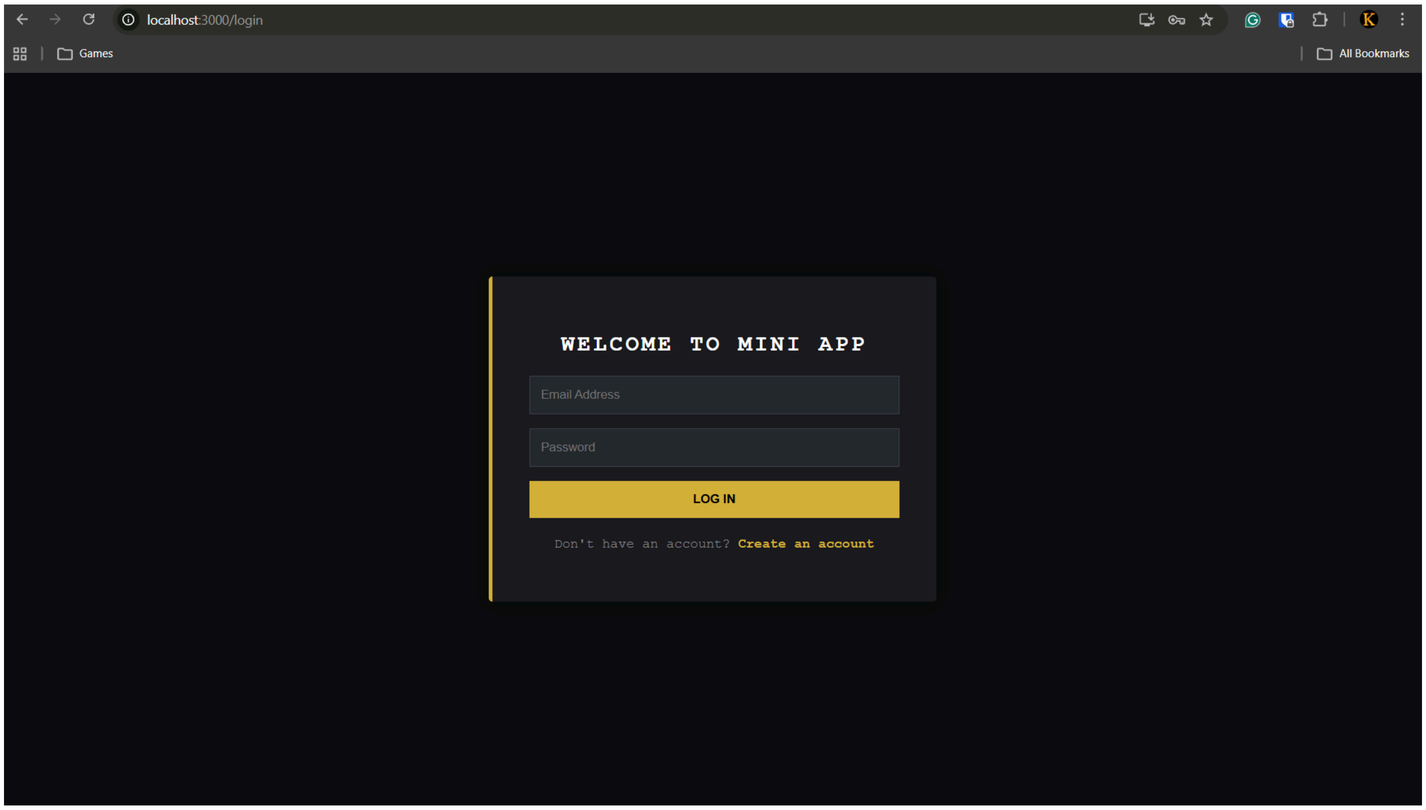
6. Appendices

Include any additional information, references, or support materials.

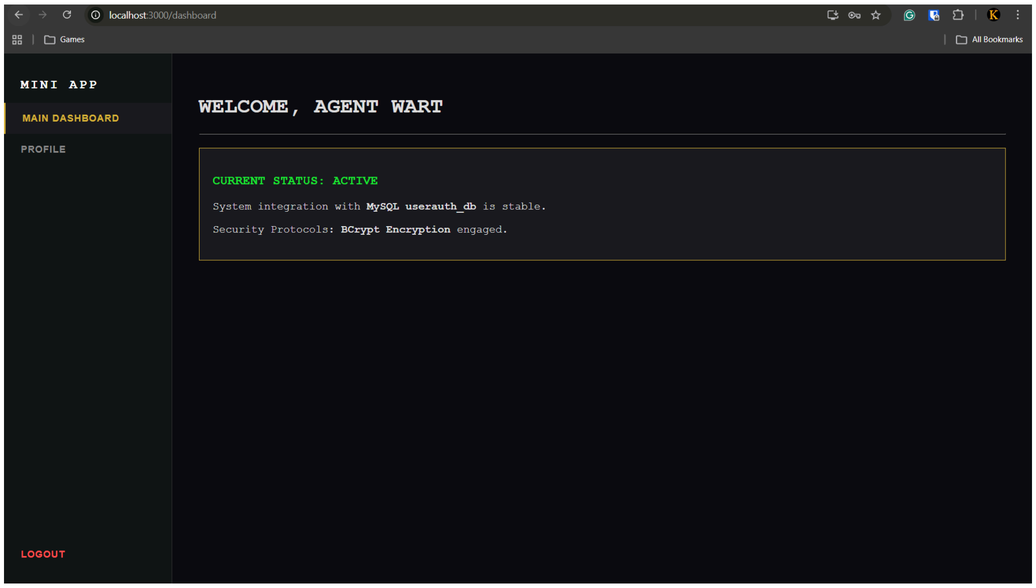
- **Registration Page**:



- **Login Page**:



- **Dashboard:**



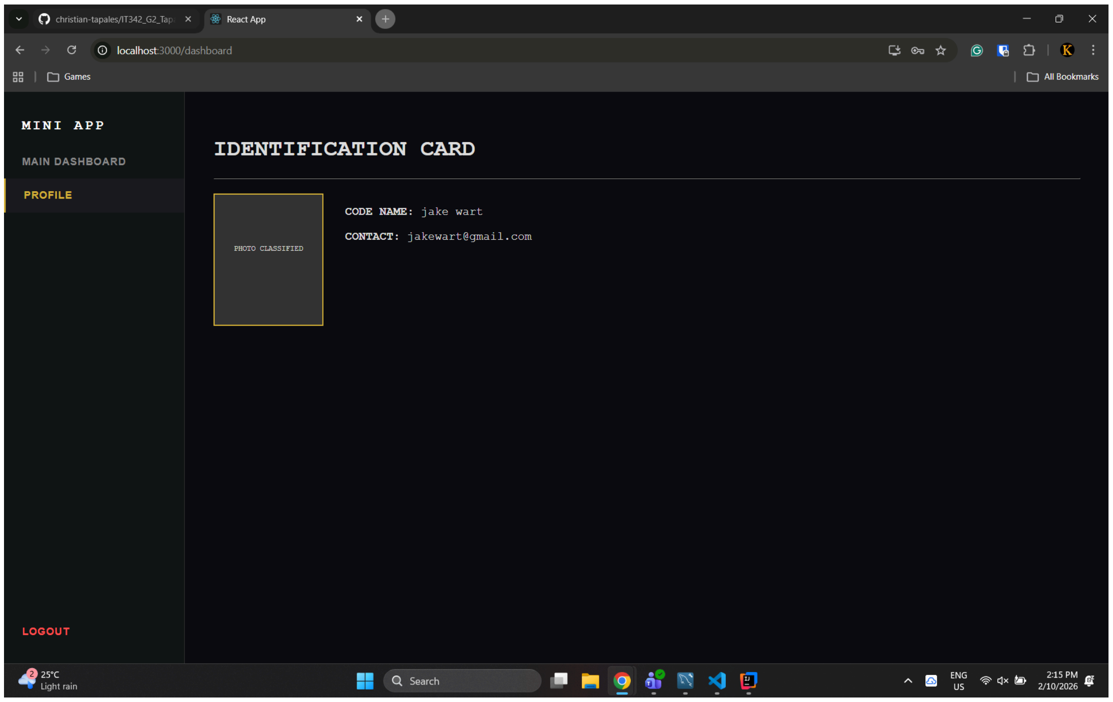
WELCOME, AGENT WART

CURRENT STATUS: ACTIVE

System integration with **MySQL userauth_db** is stable.

Security Protocols: **BCrypt Encryption** engaged.

- **Profile:**

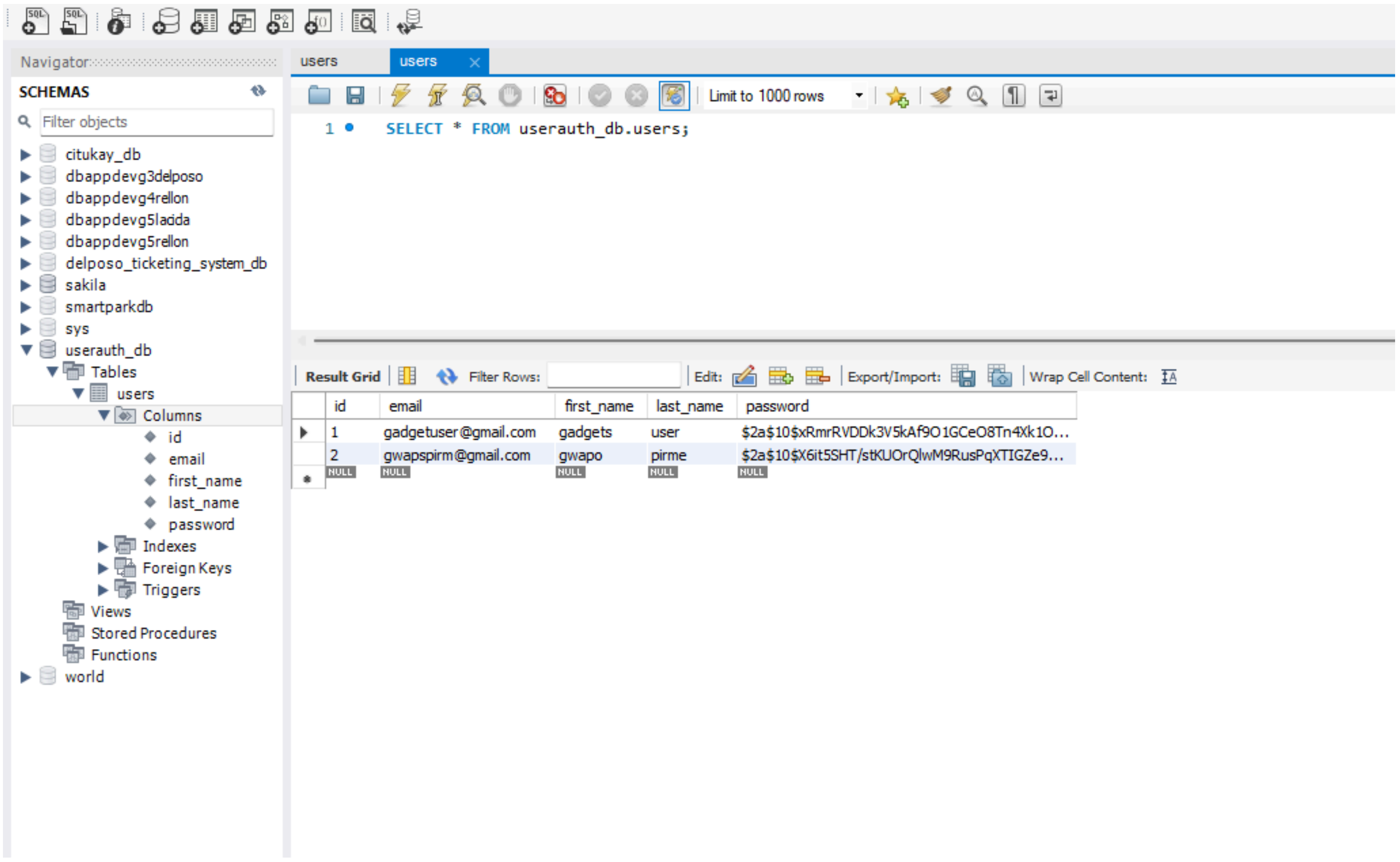


IDENTIFICATION CARD

CODE NAME: jake wart

CONTACT: jakewart@gmail.com

- **Database Evidence (To prove "No Plain Text"):**

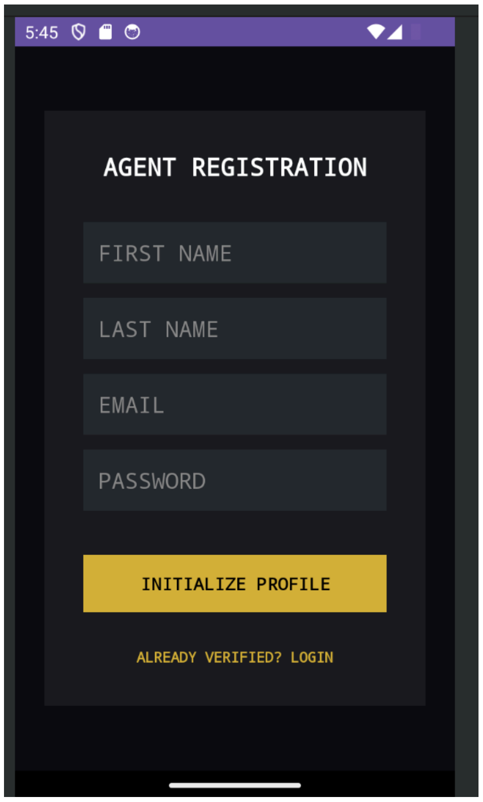


```
SELECT * FROM userauth_db.users;
```

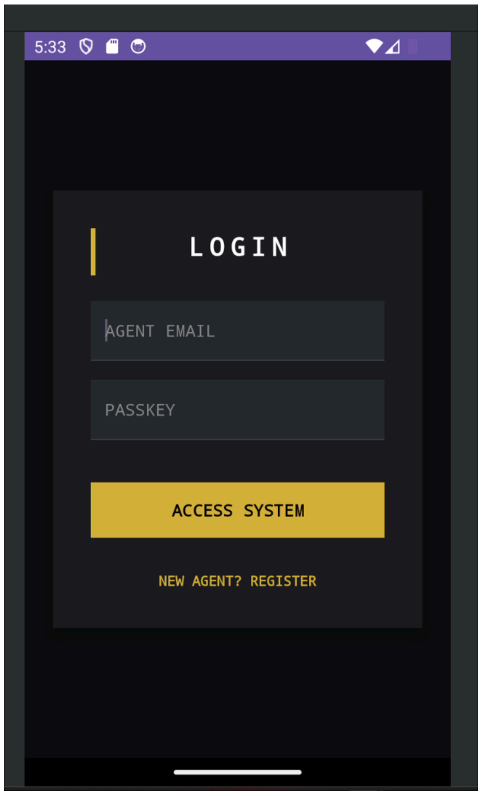
id	email	first_name	last_name	password
1	gadgetuser@gmail.com	gadgets	user	$2a$10$xRmrRVDDk3V5kAf9O1GCeO8Tn4Xk1O...
2	gwapspirm@gmail.com	gwapo	pirme	$2a$10$X6it5SHT/stKUOrQlwM9RusPqXTIGZe9...
NULL	NULL	NULL	NULL	NULL

- **Mobile Application Screenshots:**

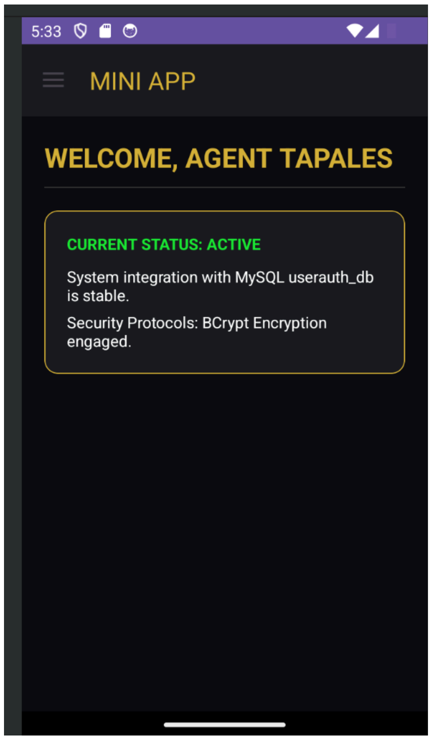
Registration



Login



Main Dashboard



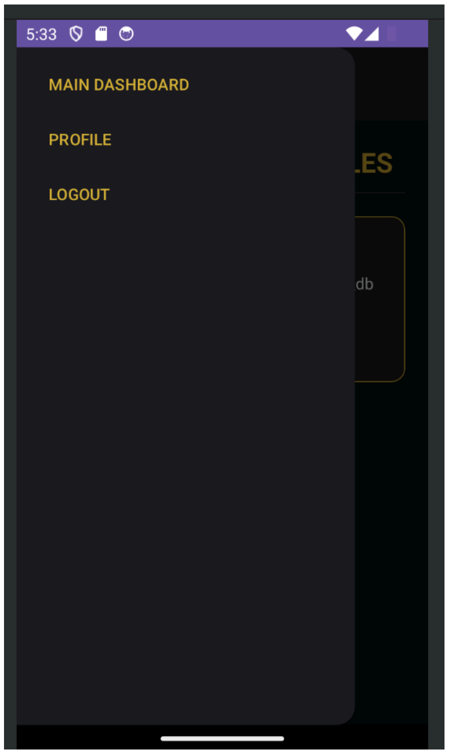
WELCOME, AGENT TAPALES

CURRENT STATUS: ACTIVE

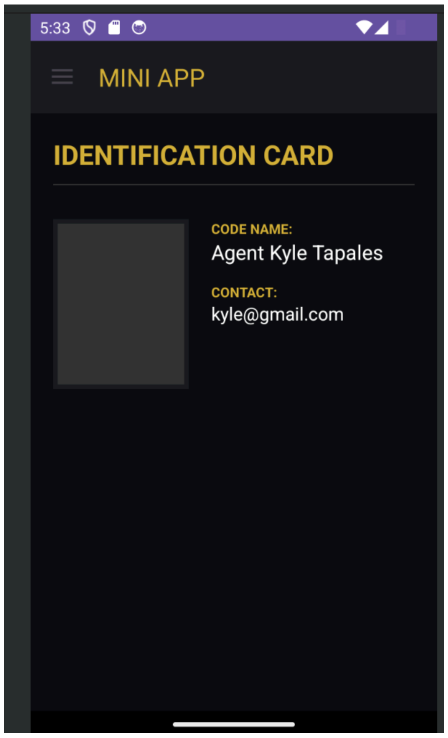
System integration with MySQL userauth_db is stable.

Security Protocols: BCrypt Encryption engaged.

Sidebar



Profile



IDENTIFICATION CARD

CODE NAME:
Agent Kyle Tapales

CONTACT:
kyle@gmail.com