



CEBU INSTITUTE OF TECHNOLOGY
U N I V E R S I T Y

IT342-G4

SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: Laboratory Activity: Mini App – User Registration
& Authentication

Prepared By: Ma. Melessa V. Cabasag

Date of Submission: February 3, 2026

Version: 1.1

Table of Contents

- 1. Introduction.....3
 - 1.1. Purpose.....3
 - 1.2. Scope.....3
 - 1.3. Definitions, Acronyms, and Abbreviations3
- 2. Overall Description.....3
 - 2.1. System Perspective3
 - 2.2. User Classes and Characteristics.....3
 - 2.3. Operating Environment.....4
 - 2.4. Assumptions and Dependencies.....4
- 3. System Features and Functional Requirements4
 - 3.1. Feature 1:4
 - 3.2. Feature 2:.....4
- 4. Non-Functional Requirements.....5
- 5. System Models (Diagrams)5
 - 5.1. ERD5
 - 5.2. Use Case Diagram6
 - 5.3. Activity Diagram7
 - 5.4. Class Diagram.....8
 - 5.5. Sequence Diagram.....8
- 6. Appendices.....9

1. Introduction

1.1. Purpose

The purpose of this system is to provide a secure and reliable authentication mechanism for a mobile and web-based application. The system will manage user registration, login, session management, and account validation. This document is intended for software developers, project stakeholders, and QA testers who will design, implement, and maintain the authentication system.

1.2. Scope

The authentication system will:

- Allow users to register an account with unique credentials.
- Authenticate users through login and maintain sessions securely.
- Provide password validation and error handling.

Boundaries: The system does **not** include advanced user management features like role-based access control, third-party authentication (OAuth), or user activity tracking. It focuses solely on secure authentication and session management.

1.3. Definitions, Acronyms, and Abbreviations

List and define

- **Auth System** – Authentication System for user login and registration.
- **User** – Any person accessing the application.
- **Session** – Temporary data storage to maintain user login state.
- **PWD** – Password.
- **UI** – User Interface.
- **API** – Application Programming Interface.

2. Overall Description

2.1. System Perspective

The authentication system is a core component of the larger application ecosystem, connecting both the mobile app (Kotlin) and web interface (HTML/PHP) and spring boot for backend interacting to database. It acts as a gatekeeper to ensure only verified users can access protected features. It interacts with the database for storing and validating user credentials and sessions.

2.2. User Classes and Characteristics

- **Regular Users:** Users who register to access app features. Familiar with basic smartphone/web usage.

- **Admin (Optional Future Use):** Oversees user accounts and system health. More technical knowledge.

2.3. Operating Environment

- **Hardware:** Android devices for mobile app, PC or mobile browser for web access.
- **Software:** Visual Studio or Android Studio (Kotlin), Web server with PHP, MySQL database.
- **Tools/Frameworks:** Firebase (optional for notifications), Session management classes, PHP/SQL and Spring boot for backend operations.

2.4. Assumptions and Dependencies

- Internet connectivity is required to communicate with the database.
- Users will provide valid email addresses or usernames.
- Database schema is pre-configured to store user credentials securely.
- External dependencies include the Android SDK, PHP runtime, MySQL server, and Firebase (if used).

3. System Features and Functional Requirements

3.1. Feature 1: User Registration

Description:

Allows new users to create an account by entering a unique username/email and password. Validates the input to ensure security.

Functional Requirements:

- Users must provide a unique username or email.
- Passwords must meet security criteria (e.g., minimum length, alphanumeric).
- System must check if the user already exists and return an appropriate message.

3.2. Feature 2: User Login

Description:

Allows registered users to log in and create a session. Validates credentials against the database.

Functional Requirements:

- Users must enter correct username/email and password.
- System verifies credentials and initiates a session upon success.
- Displays an error message for invalid login attempts.

4. Non-Functional Requirements

- **Performance:** Login and registration requests should complete within 2 seconds.
- **Security:** Passwords must be hashed before storing; sessions must be secured.
- **Usability:** Intuitive UI with clear messages for errors or validation.
- **Reliability:** 99% uptime of authentication services.
- **Scalability:** Able to handle multiple concurrent users without performance degradation.
- **Maintainability:** Code should be modular for easy updates or addition of new features.

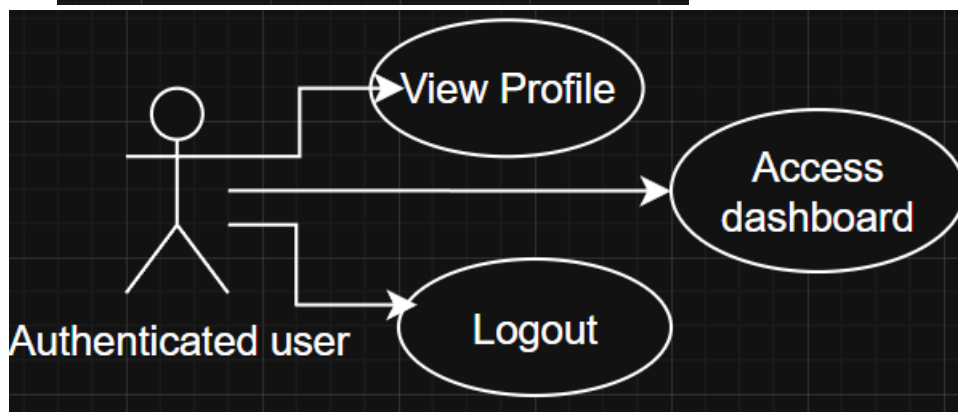
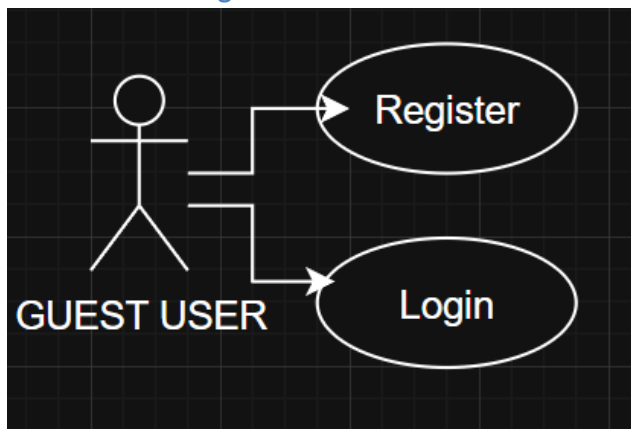
5. System Models (Diagrams)

Insert the necessary diagrams for the system:

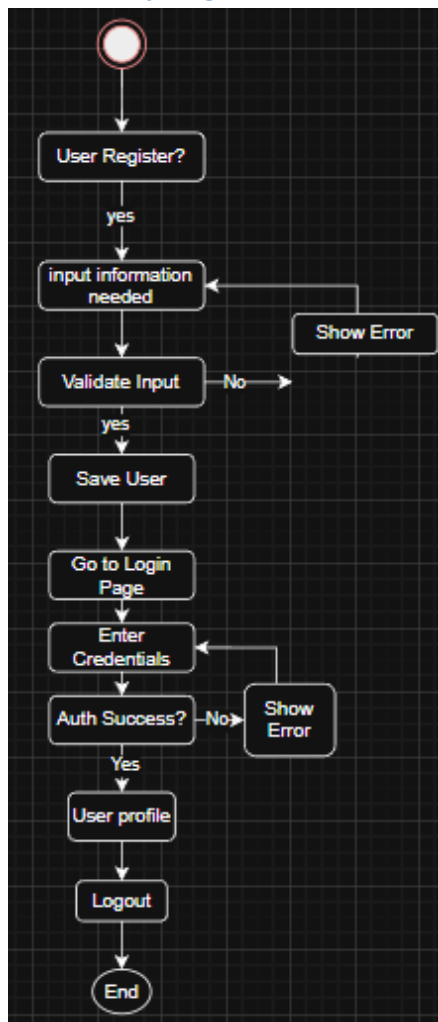
5.1. ERD

USER	
PK	ID
FK1	email
	username
	password_hash
	firstname
	lastname
	is_active
	role

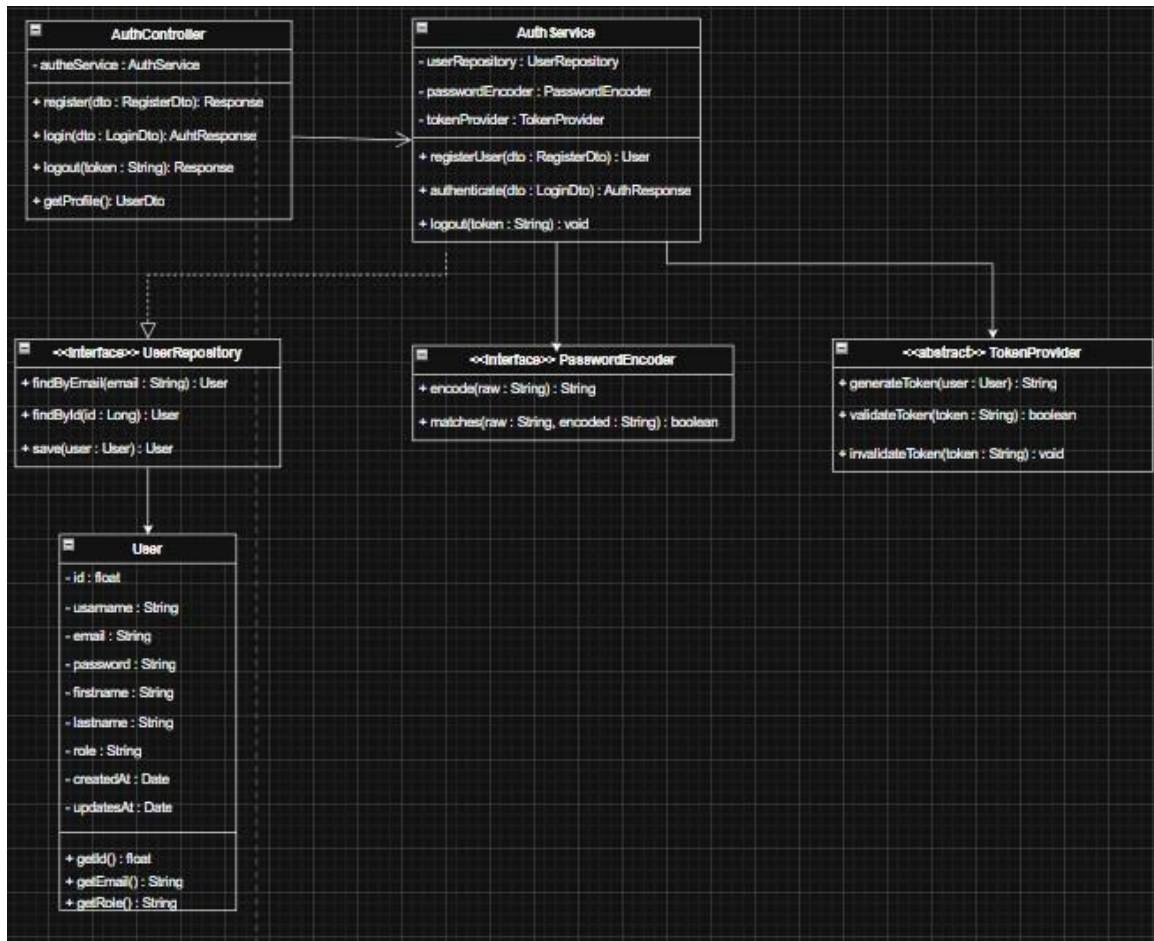
5.2. Use Case Diagram



5.3. Activity Diagram

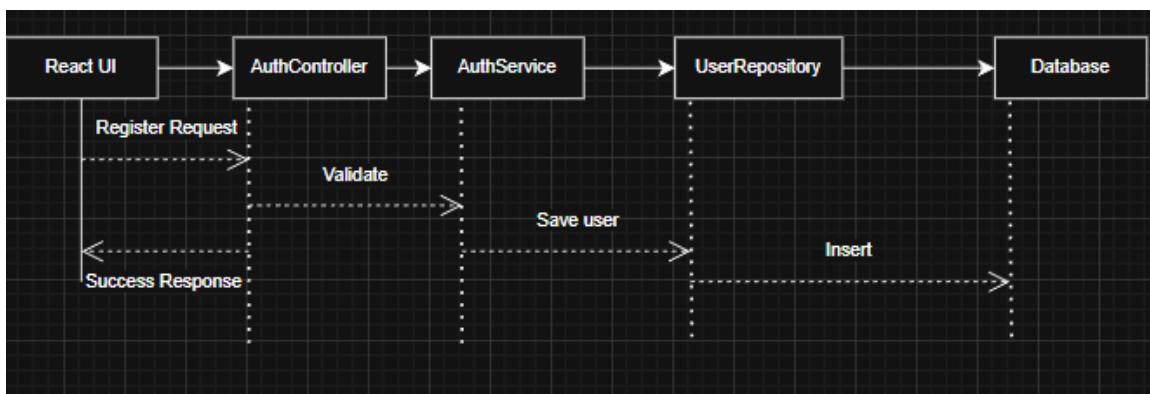


5.4. Class Diagram

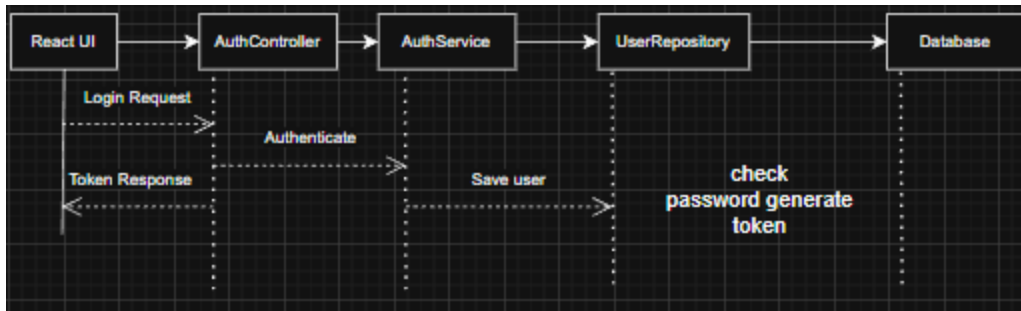


5.5. Sequence Diagram

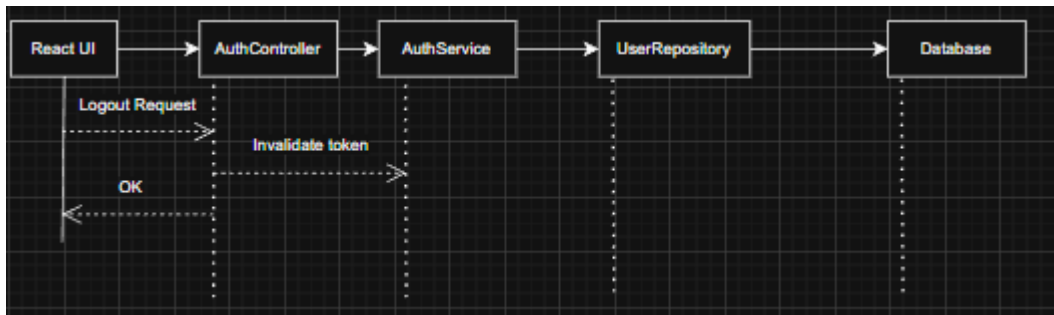
Register Sequence Diagram



Login Sequence Diagram



Login Sequence Diagram



6. Appendices

Include any additional information, references, or support materials.

7. Documentation

Register Page:

The screenshot shows a web browser at localhost:3000/register. The page has a dark red background. On the left, there's a 'MiniMap' logo with the text 'Create an account to get started.' and buttons for 'Login' and 'Register'. On the right, there's a registration form with the following fields:

- Firstname: Melessa
- Lastname: Cabasag
- Email: melessa@gmail.com
- Username: (Optional) Molly123
- Password: (masked with dots)
- Confirm Password: (masked with dots)

Below the form, there are two requirements: 'Minimum 8 characters' and 'Contains an uppercase letter'. At the bottom, a small footer reads 'A simple minimap-app version by Ma. Melessa Cabasag'.

Login page:

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/login'. The page has a dark red background. On the left, there is a 'Mini-App' section with the text 'Sign in to your account.' and two buttons: 'Login' and 'Register'. On the right, there is a larger login form with a 'Login' button at the top and a 'Register' button at the bottom. The form contains two input fields: 'Username or Email:' with the value 'Melly123' and 'Password:' with a masked password '*****'. Below the password field is a 'Login' button. At the bottom of the form, there is a link that says 'Don't have an account? Register'. The browser's address bar also shows 'import favorites', 'Gmail: Email from...', and 'fsdgi'sdf'.

localhost:3000/login

import favorites | Gmail: Email from... | fsdgi'sdf

Mini-App
Sign in to your account.

Login Register

Username or Email:
Melly123

Password:

Login

Don't have an account? Register

A simple mini-app version by Ma Melessa Cabasag

Dashboard Page:

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/dashboard'. The page has a dark red background. On the left, there is a sidebar with the text 'Mini-App BY MA. MELESSA CABASAG'. Below this, there are two buttons: 'Dashboard' and 'Profile'. At the bottom of the sidebar, there is a 'Logout' button. The main content area on the right has a large heading 'Hi Melessa!' with a hand emoji, followed by the text 'Welcome to your Mini-App Dashboard'. The browser's address bar also shows 'import favorites', 'Gmail: Email from...', and 'fsdgi'sdf'.

localhost:3000/dashboard

import favorites | Gmail: Email from... | fsdgi'sdf

Mini-App
BY MA. MELESSA CABASAG

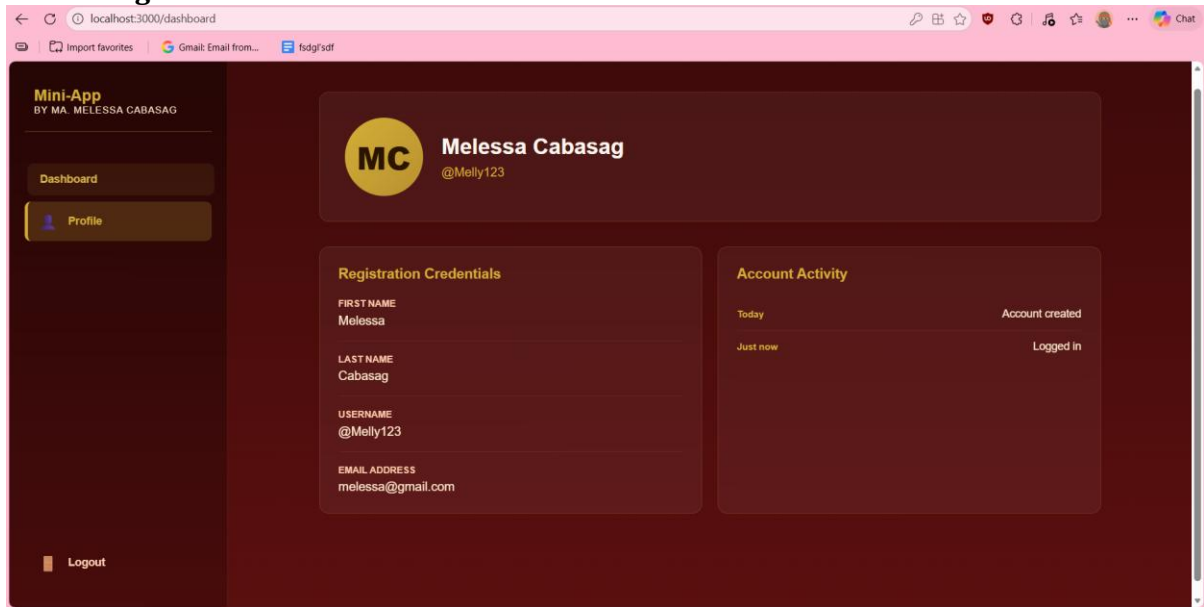
Dashboard Profile

Logout

Hi Melessa! 🖐️

Welcome to your Mini-App Dashboard

Profile Page:



Logout Page:

