



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

# IT342-Section SYSTEMS INTEGRATION AND ARCHITECTURE 1

---

## **FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)**

---

Project Title: Mini App – User Registration & Authentication

Prepared By: Lubanga, Van Kehrby M.

Date of Submission: 01/30/2026

Version: 1.0

# Table of Contents

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

## 1. Introduction

### 1.1. Purpose

To provide secure and reliable way for users to register, login, access resources, and logout.

### 1.2. Scope

The system will:

- Allow users to create accounts
- Allow users to log in using credentials
- Restrict access to protected pages for unauthenticated users.
- Allow users to log out securely.

### 1.3. Definitions, Acronyms, and Abbreviations

User	A person who registers and logs in to the system
Registration	The process of creating a new user account
Login	The process of authenticating a registered user
Logout	Ending a user session
Dashboard	Protected page visible only to authenticated users

## 2. Overall Description

### 2.1. System Perspective

It operates as a module responsible for validating users, maintaining session states, and controlling access to protected resources.

### 2.2. User Classes and Characteristics

User – Can register and attempt to log in. Cannot access to pages.

Authenticated User – Can access protected pages view profile/dashboard, and logout.

### 2.3. Operating Environment

- Web browser or mobile device for user access
- Backend server handling authentication
- Database to store user credentials securely

### 2.4. Assumptions and Dependencies

- Users will provide valid credentials
- Passwords will be encrypted and secured

- System depends on running database for storing user data

### 3. System Features and Functional Requirements

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

#### 3.1. Feature 1: User Registration

Description: User can create an account to access protected pages

Functional Requirements:

- Users provide full name, birthdate, email, and password
- System validates input and prevents duplicate accounts
- Passwords are securely hashed

#### 3.2. Feature 2: User Login

Description: Registered users can log in to access protected pages

Functional Requirements:

- Users enter email/username and password
- System verifies credentials
- Invalid credentials trigger an error and allow retry

### 4. Non-Functional Requirements

- Performance: System must authenticate users within 2–3 seconds.
- Security: Passwords must be encrypted; sessions protected from hijacking.
- Usability: Simple and clear forms for registration and login.
- Reliability: The system must remain available 99% of the time.
- Maintainability: Code should be modular for easy updates and fixes

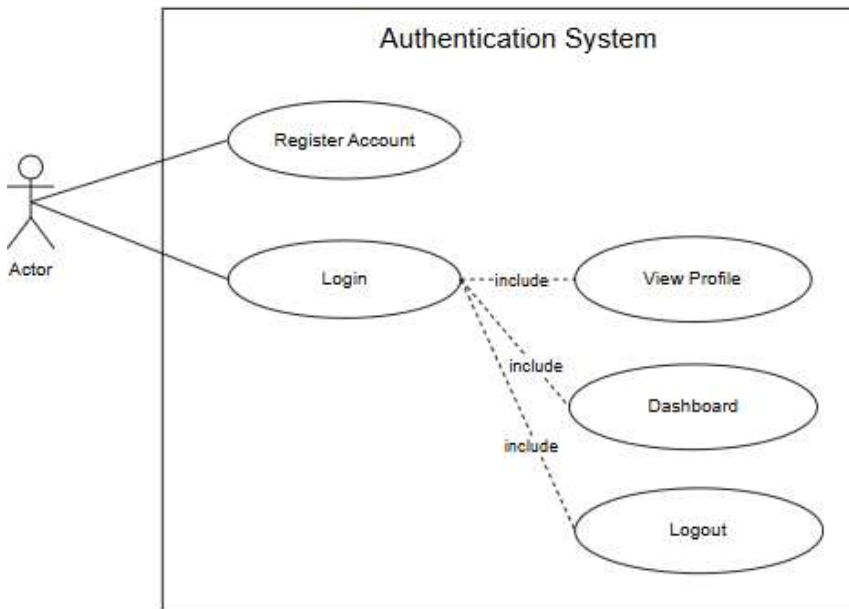
### 5. System Models (Diagrams)

*Insert the necessary diagrams for the system:*

#### 5.1. ERD

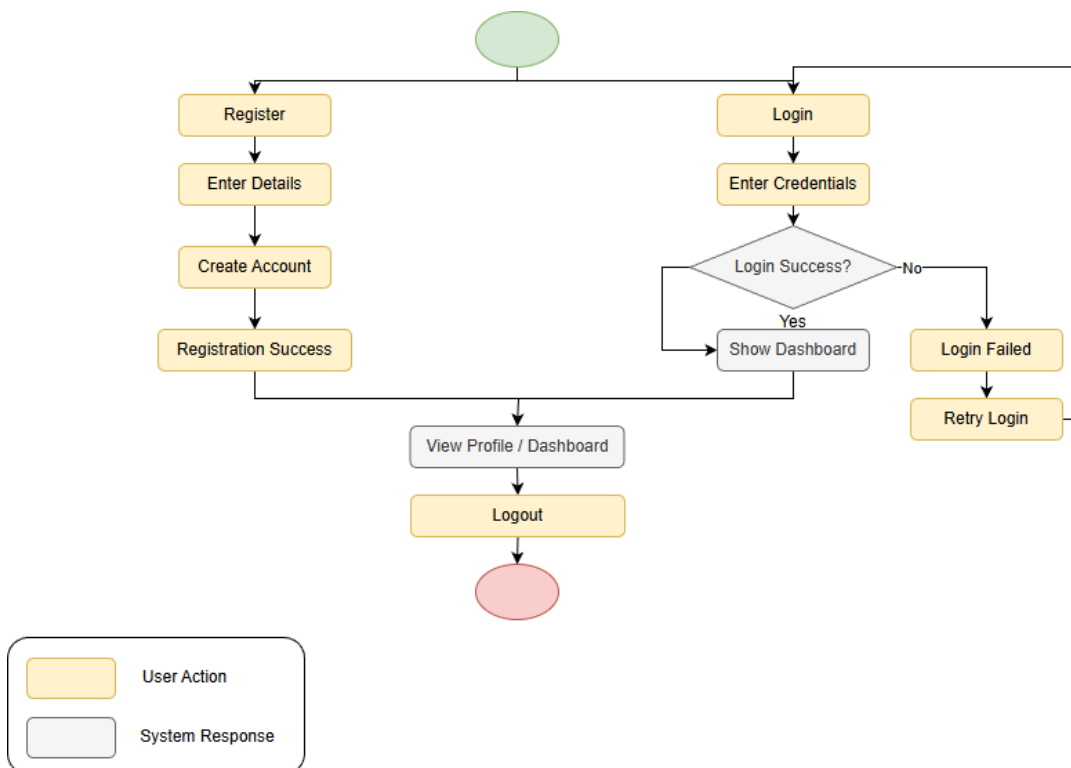
User	
PK	<u>uniqueID</u>
	fullName
	birthDate
	email
	password

#### 5.2. Use Case Diagram

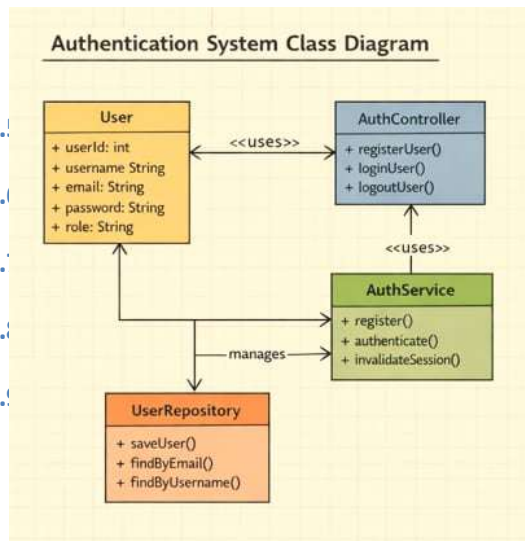


### 5.3.

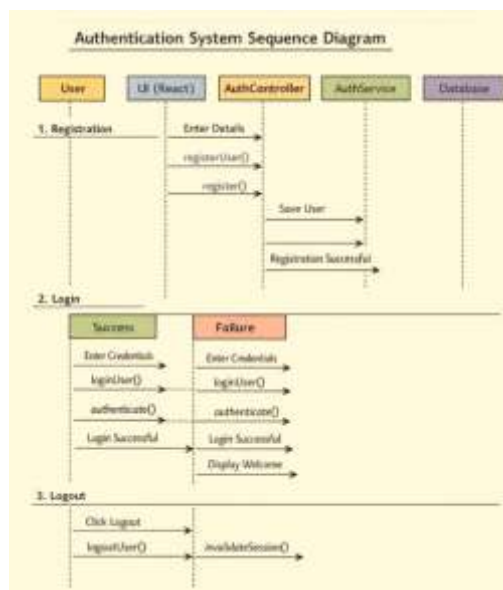
#### Activity Diagram



## 5.4. Class Diagram



## 5.5 Sequence Diagram



## 6. Appendices

Include any additional information, references, or support materials.