



**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: Din, Jon Nicole F.

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.....5
  - 5.2. Use Case Diagram .....5
  - 5.3. Activity Diagram .....6
  - 5.4. Class Diagram.....7
  - 5.5. Sequence Diagram .....7
- 6. Appendices .....7

## 1. Introduction

### 1.1. Purpose

The purpose of this system is to provide a secure, reliable, and user-friendly authentication mechanism that allows users to register, log in, access protected resources, and log out safely. The system ensures that only authenticated users can access restricted pages, such as dashboards, while protecting user credentials through secure validation, encryption, and session management.

### 1.2. Scope

The system will:

- Enable users to create new accounts
- Allow registered users to log in using valid credentials
- Prevent unauthenticated users from accessing protected pages
- Provide a secure logout mechanism to end user sessions

### 1.3. Definitions, Acronyms, and Abbreviations

|              |  |
|--------------|--|
| User         | An individual who registers and logs in to the system    |
| Registration | The process of creating a new user account               |
| Login        | The process of verifying user credentials to gain access |
| Logout       | The action of terminating an active user session         |
| Dashboard    | A protected page accessible only to authenticated users  |

## 2. Overall Description

### 2.1. System Perspective

The system functions as an authentication module responsible for verifying user identities, managing session states, and controlling access to protected resources within the application.

### 2.2. User Classes and Characteristics

User – Can register and attempt to log in but cannot access protected pages.

Authenticated User – Can access protected pages such as profiles or dashboards and can log out securely.

### 2.3. Operating Environment

- Web browsers or mobile devices used by users
- Backend server responsible for authentication and session handling

- Database used to securely store user credentials and related data

#### 2.4. Assumptions and Dependencies

- Users will provide accurate and valid information during registration
- Passwords will be encrypted and securely stored
- The system relies on a functioning database to store and retrieve user data

### 3. System Features and Functional Requirements

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

#### 3.1. Feature 1:

Description:

Functional Requirements:

- Users must provide full name, birthdate, email, and password
- The system must validate all input fields and prevent duplicate registrations
- User passwords must be securely hashed before being stored

#### 3.2. Feature 2:

Description:

Functional Requirements:

- Users must enter a valid email/username and password
- The system must verify the entered credentials against stored data
- If credentials are invalid, the system must display an error message and allow retry

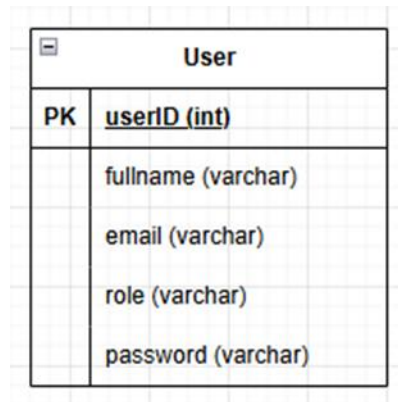
### 4. Non-Functional Requirements

- Performance: The system must authenticate users within 2–3 seconds
- Security: Passwords must be encrypted, and user sessions must be protected from unauthorized access.
- Usability: Registration and login interfaces must be simple and easy to use
- Reliability: The system must maintain at least 99% availability
- Maintainability: The system code should be modular to allow easy updates and maintenance

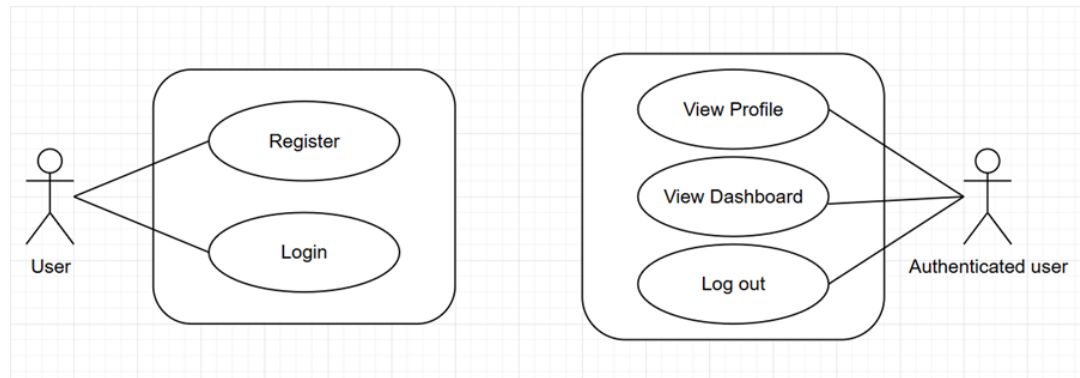
### 5. System Models (Diagrams)

*Insert the necessary diagrams for the system:*

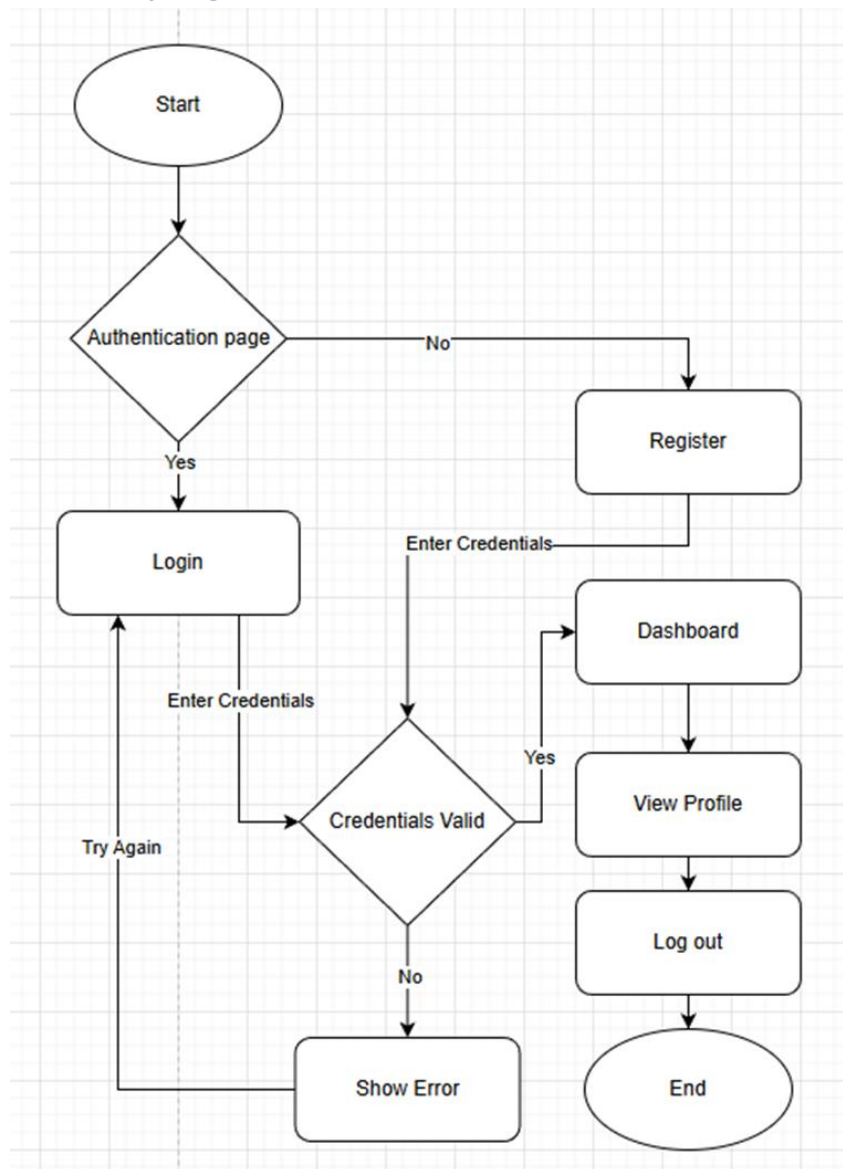
### 5.1. ERD



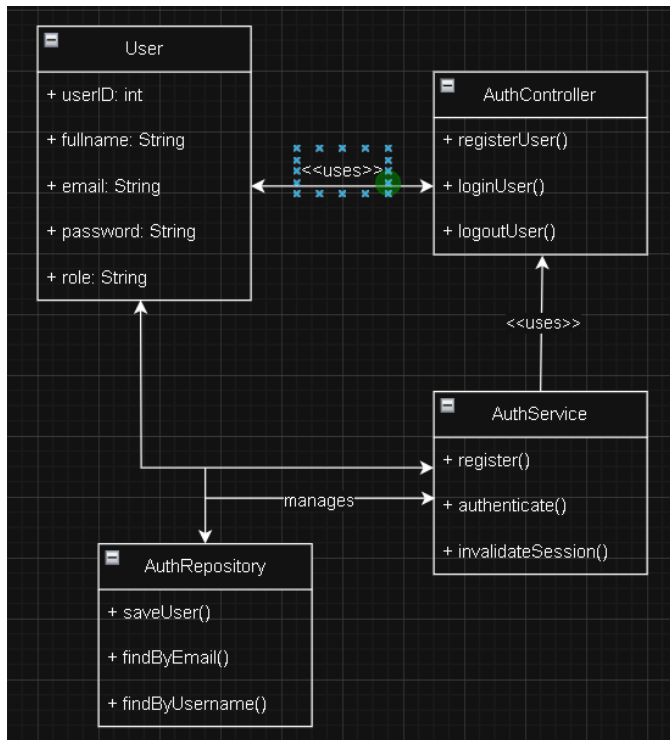
### 5.2. Use Case Diagram



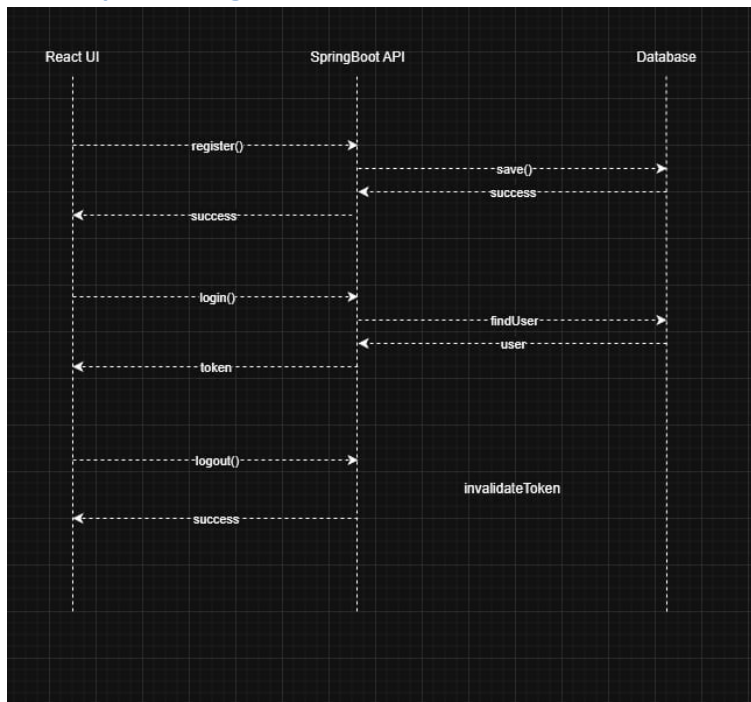
### 5.3. Activity Diagram



## 5.4. Class Diagram



## 5.5. Sequence Diagram



## 6. Appendices

Include any additional information, references, or support materials.