



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

# IT342-G6

## SYSTEMS INTEGRATION AND ARCHITECTURE 1

---

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

---

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

Prepared By: Paul Andrie E. Bibit

Date of Submission: 1/31/2026

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.....3
  - 3.1. Feature 1:.....3
  - 3.2. Feature 2:.....3
- 4. Non-Functional Requirements.....3
- 5. System Models (Diagrams)..... 4
  - 5.1. ERD.....4
  - 5.2. Use Case Diagram..... 4
  - 5.3. Activity Diagram.....4
  - 5.4. Class Diagram.....4
  - 5.5. Sequence Diagram..... 4
- 6. Appendices..... 4

## 1. Introduction

### 1.1. Purpose

Design the system flow for User Registration, Login, and Logout.

## 1.2. Scope

This activity focuses only on documentation and diagrams.

## 1.3. Definitions, Acronyms, and Abbreviations

**FRS** – Functional Requirements Specification

**ERD** – Entity Relationship Diagram

## 2. Overall Description

### 2.1. System Perspective

N/A

### 2.2. User Classes and Characteristics

**Guest User** - A user who has not logged in; can register and log in.

**Authenticated User** - A logged-in user who can access protected pages and log out.

### 2.3. Operating Environment

Draw.io / diagrams.net for the diagrams.

### 2.4. Assumptions and Dependencies

N/A

## 3. System Features and Functional Requirements

### 3.1. Feature 1: User Registration

**Description:** Allows new users to create an account by providing a unique username, email, and password.

**Functional Requirements:**

- 3.1. The system shall validate that the username and email are not already taken in the database.
- 3.2. The system shall hash the user's password using BCrypt before storing it in the MySQL database.
- 3.3. The system shall return a success message upon successful data persistence.

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

**Description:** Enables registered users to access the protected dashboard and securely end their session.

**Functional Requirements:**

- The system shall verify credentials by comparing the input password with the hashed password stored in the database.
- The system shall create a virtual session by storing user data in the browser's local storage upon successful login.
- The system shall provide a logout function that clears the local storage and redirects the user to the login page.

### 3.3. Feature 3: Protected Dashboard

**Description:** A restricted area accessible only to authenticated users.

**Functional Requirements:**

- The system shall display the authenticated user's specific information (Username, Email) fetched from the database.
- The system shall prevent unauthenticated guest users from accessing the dashboard view.

## 4. Non-Functional Requirements

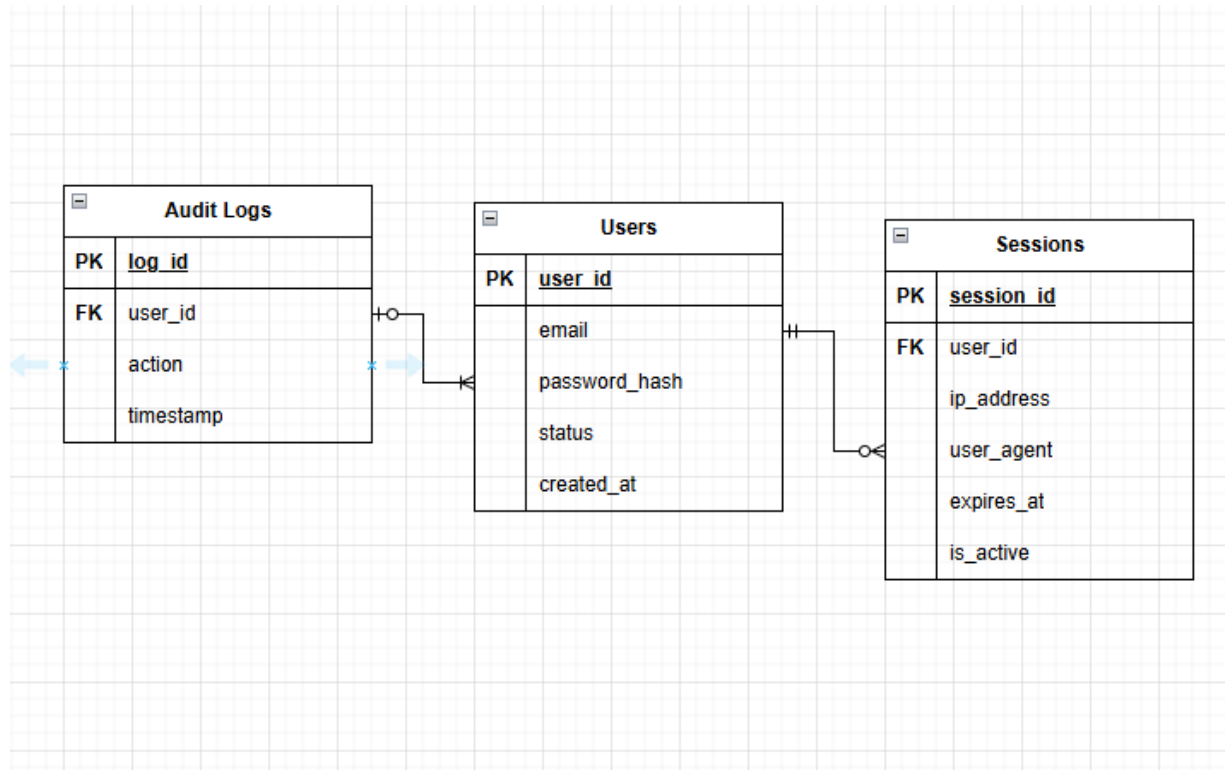
**Security:** All user passwords must be encrypted using industry-standard hashing (BCrypt).

**Usability:** The web interface must provide clear error messages for invalid login attempts or duplicate registrations.

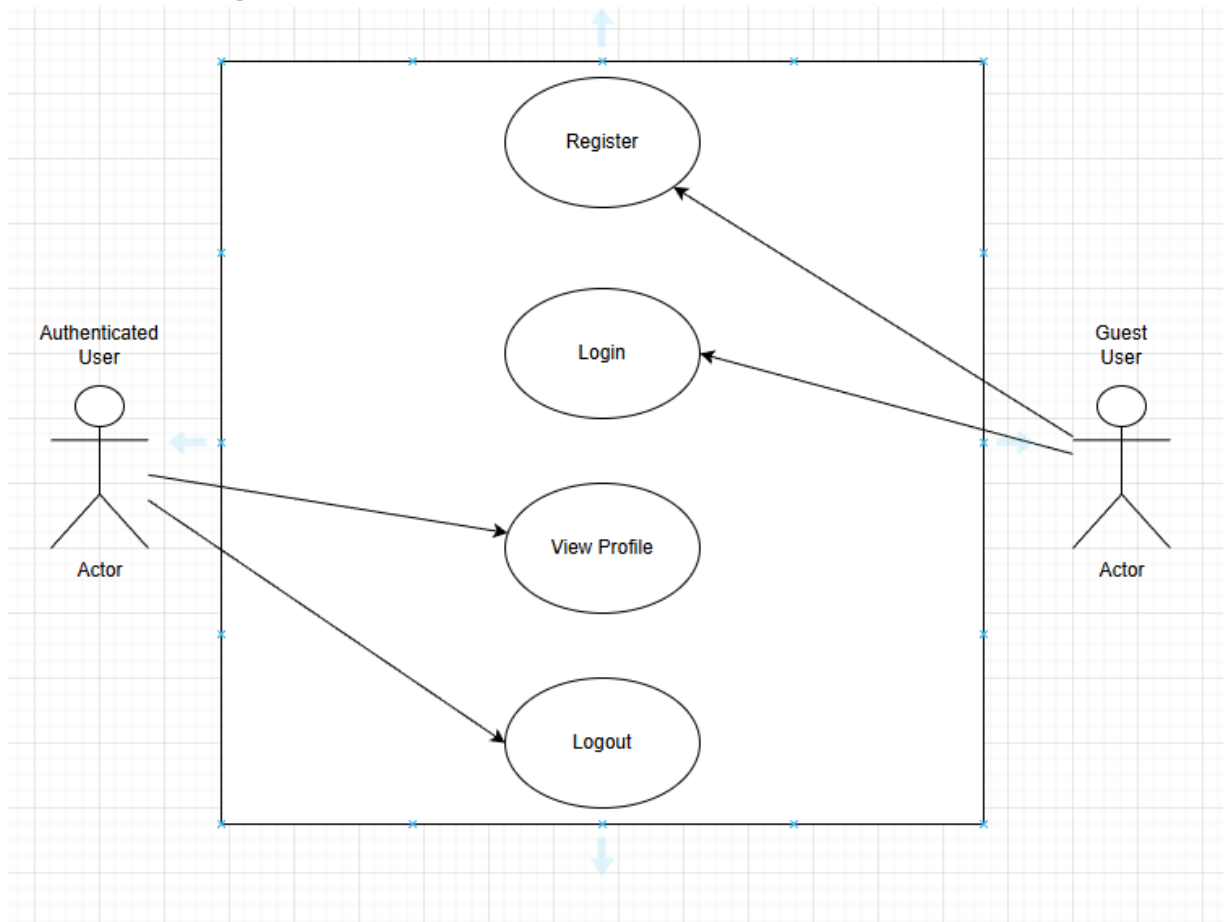
**Availability:** The backend (Spring Boot) and database (MySQL) must be operational for the frontend (React) to process requests.

## 5. System Models (Diagrams)

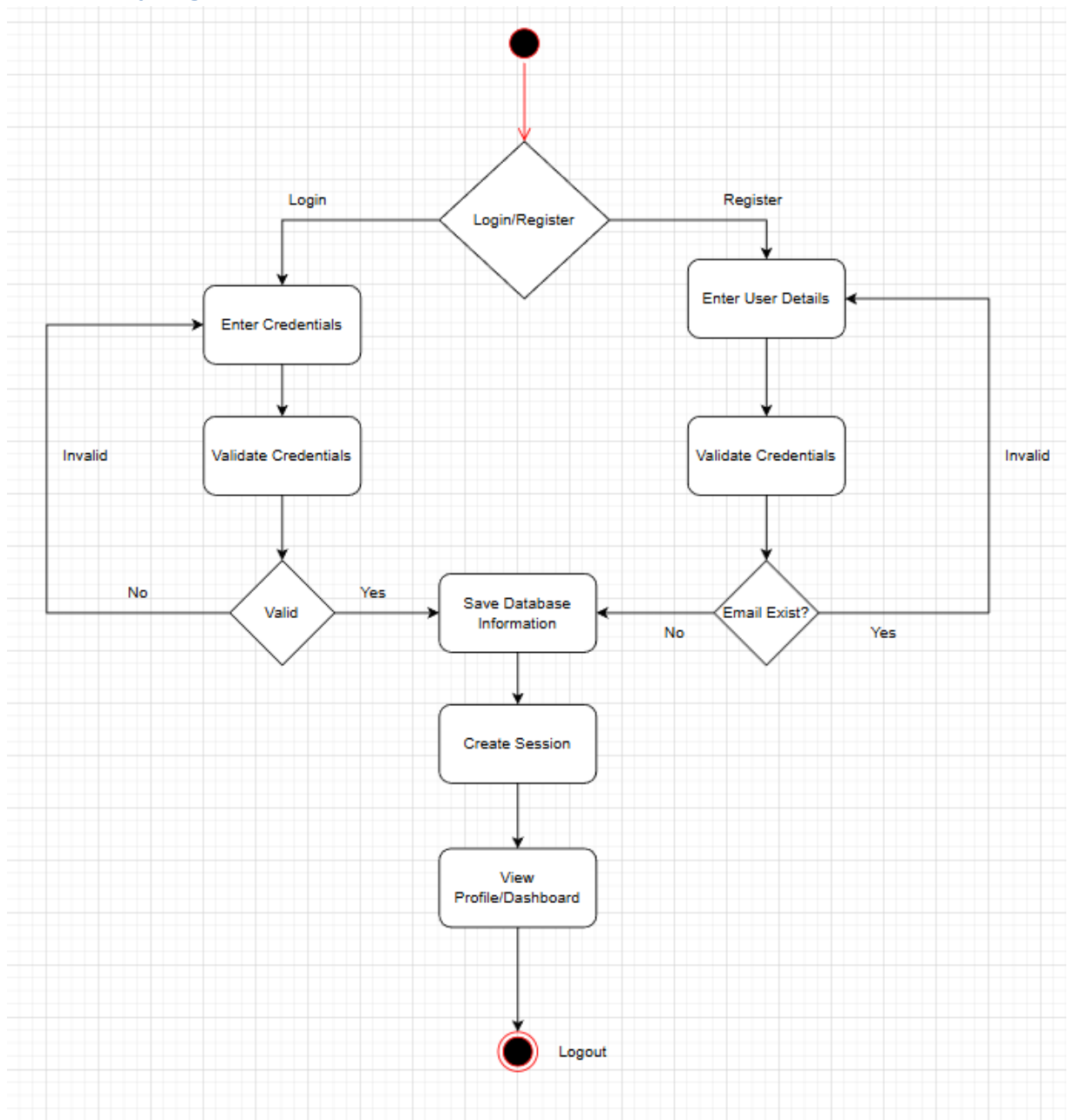
### 5.1. ERD



## 5.2. Use Case Diagram



### 5.3. Activity Diagram



### 5.4. Class Diagram N/A

### 5.5. Sequence Diagram N/A

## 6. Appendices

