



CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-G6

SYSTEMS INTEGRATION AND

ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

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

Prepared By: Kent Michael E.Bebita

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

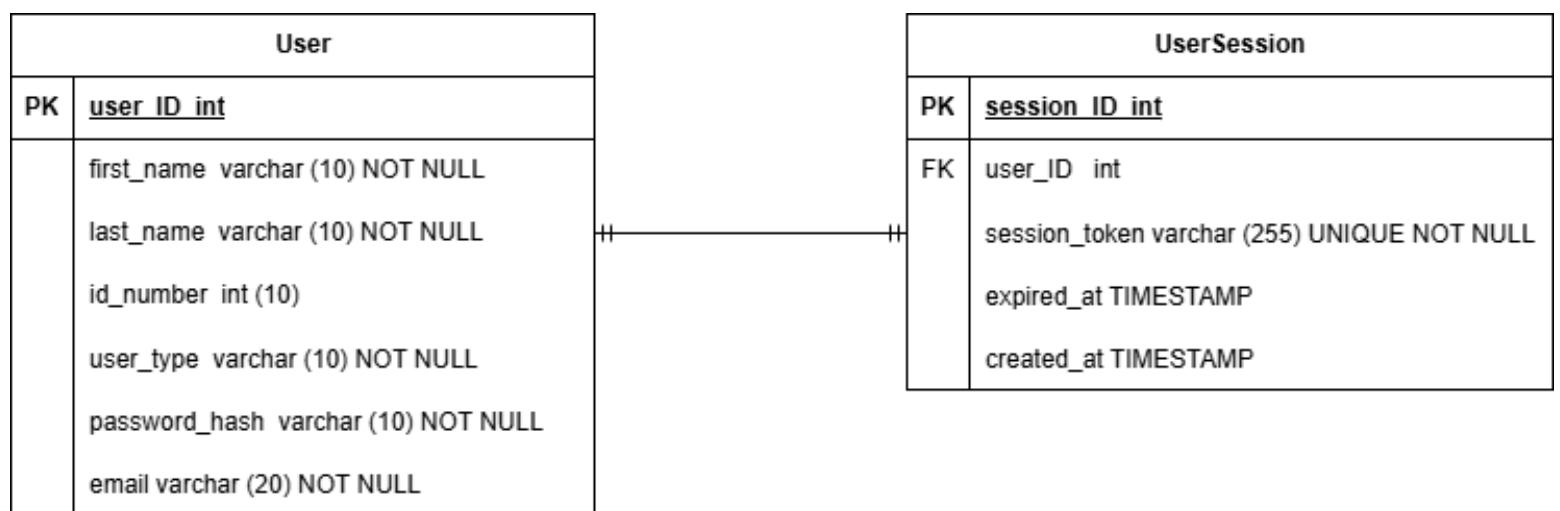
Security: All user passwords must be encrypted using industry-standard hashing (BCrypt).

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

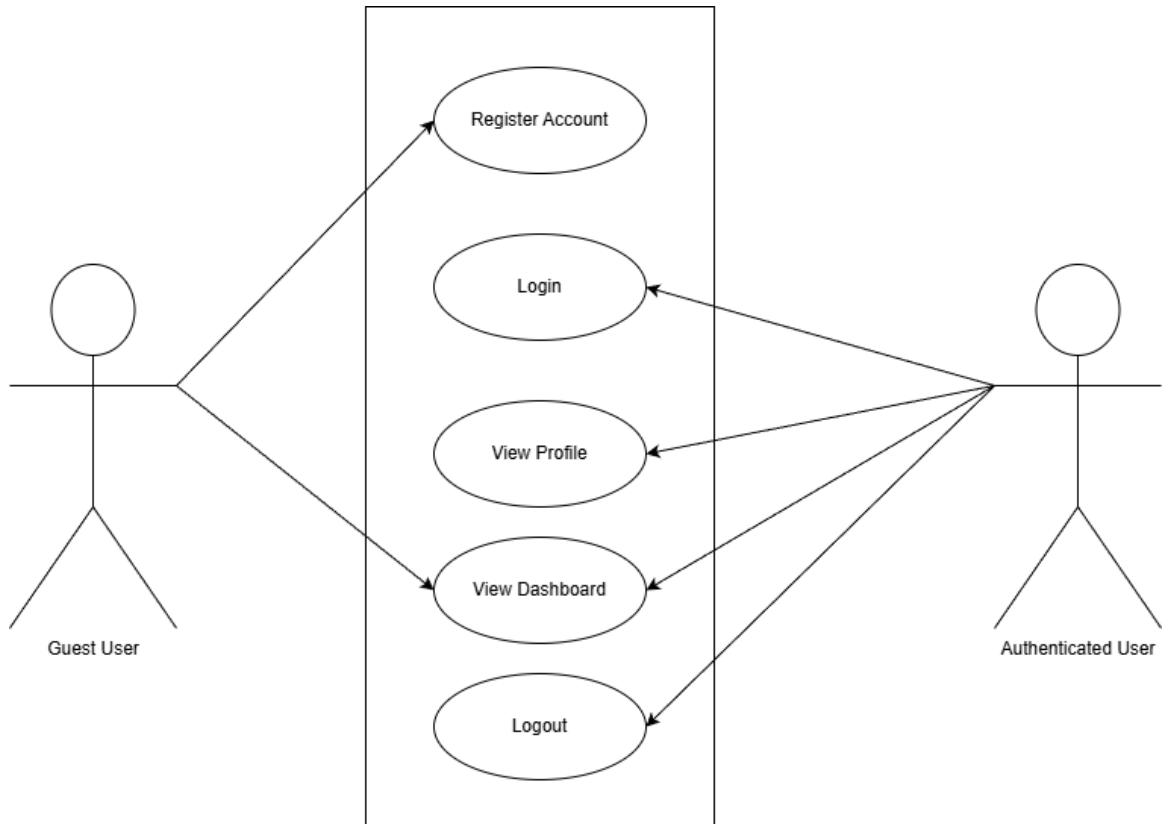
Availability: The backend (Spring Boot) and database (MySQL) must be operational for the frontend (React) to process requests.

5. System Models (Diagrams)

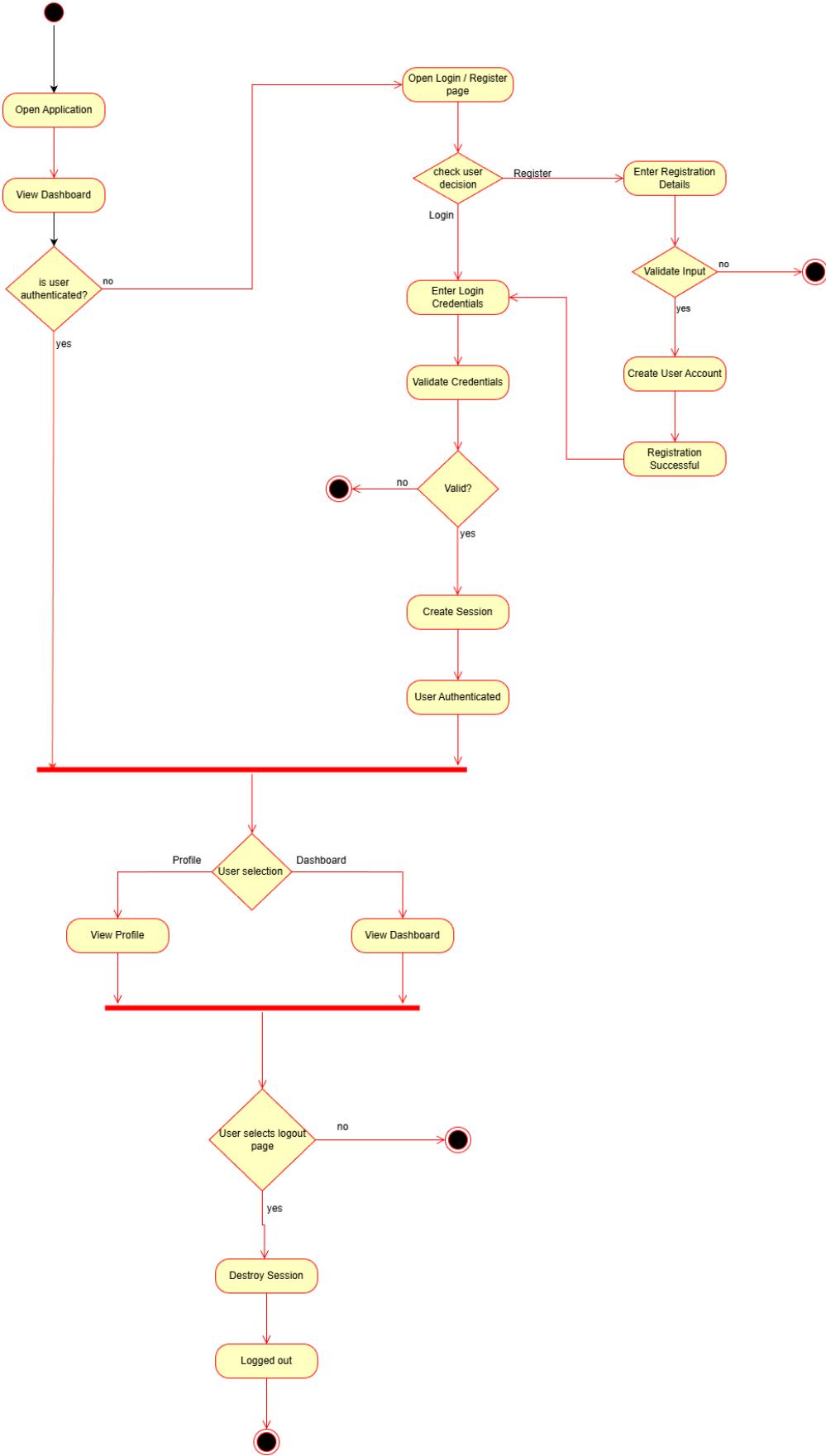
5.1. ERD



Use Case Diagram



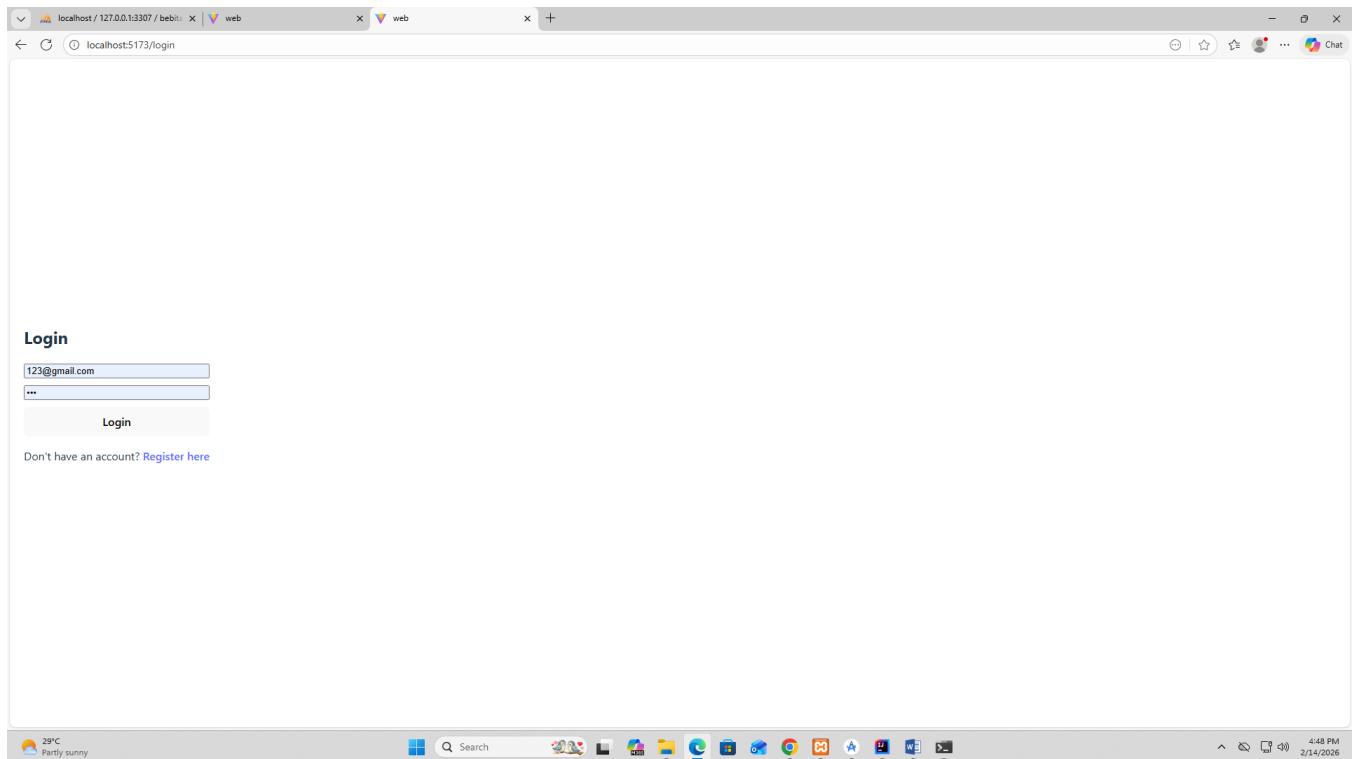
5.2. Activity Diagram



5.3. Class Diagram N/A

5.4. Sequence Diagram N/A

6. Appendices



localhost / 127.0.0.1:3307 / bebiti web web

localhost:5173/register

Registration Test

First Name

Last Name

123@gmail.com

Email

...

Student

Register User



localhost / 127.0.0.1:3307 / bebiti web web

localhost:5173/dashboard

Welcome, Kent!

Role: Student

Email: 123@gmail.com

Student Dashboard

Here you can view your schedule, grades, etc.



localhost / 127.0.0.1:3307 / bebita_lab_1 / web / web

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=bebita_lab_1&table=users

Server: 127.0.0.1:3307 > Database: bebita_lab_1 > Table: users

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Showing rows 0 - 1 (2 total). Query took 0.0004 seconds.

SELECT * FROM `users`

Profile [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	id	email	first_name	id_number	last_name	password_hash	user_type
<input type="checkbox"/>	1	testuser@gmail.com	Kent	123123123	Bebita	\$2a\$10\$WUqHujnFe8L4VDlgBBuuG8GrSzA3deR4rb0pp...	Student
<input type="checkbox"/>	2	123@gmail.com	Kent	123123	Babita	\$2a\$10\$MxkHx03Zg8NUG/bhIUS5u8kINE.cmZmfGF9WwnB...	Student

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label: Let every user access this bookmark

Bookmark this SQL query

Console

29°C Partly sunny

Search

4:48 PM 2/14/2026

The screenshot shows the phpMyAdmin interface for a MySQL database named 'bebita_lab_1'. The current table is 'users'. The interface includes a sidebar with a tree view of databases and tables, and a main panel for managing the 'users' table. The table has columns: id, email, first_name, id_number, last_name, password_hash, and user_type. Two rows are present in the table:

	id	email	first_name	id_number	last_name	password_hash	user_type
<input type="checkbox"/>	1	testuser@gmail.com	Kent	123123123	Bebita	\$2a\$10\$WUqHujnFe8L4VDlgBBuuG8GrSzA3deR4rb0pp...	Student
<input type="checkbox"/>	2	123@gmail.com	Kent	123123	Babita	\$2a\$10\$MxkHx03Zg8NUG/bhIUS5u8kINE.cmZmfGF9WwnB...	Student

