



**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: User Registration & Authentication

Prepared By: Policios, Andre C.

Date of Submission: 02/11/2026

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

The purpose of this document is to define the functional and non-functional requirements of the **User Registration & Authentication System**. This document is intended for instructors, developers, and stakeholders who need a clear understanding of the system's behavior, features, and constraints.

### 1.2. Scope

The system provides secure user account management functionalities including **user registration, login, logout, and profile viewing**. It ensures proper authentication and authorization of users before granting access to protected system features. The system is limited to user identity management and does not cover advanced features such as payment processing or third-party integrations.

### 1.3. Definitions, Acronyms, and Abbreviations

- Authentication - Process of verifying user identity
- Authorization - Granting access to system resources
- JWT - JSON Web Token used for authentication
- API - Application Programming Interface
- UI - User Interface

## 2. Overall Description

### 2.1. System Perspective

The system follows a **client-server architecture** where the frontend (React UI) communicates with a backend (Spring Boot REST API), which interacts with a relational database for user data storage. Authentication tokens are used to manage secure user sessions.

### 2.2. User Classes and Characteristics

User Type	Description
Guest User	Can register and log in
Authenticated User	Can view profile and log out

### 2.3. Operating Environment

- **Frontend:** ReactJS
- **Backend:** Spring Boot (Java)
- **Database:** MySQL
- **Authentication:** JWT-based token authentication
- **Platform:** Web-based system accessed via modern browsers

### 2.4. Assumptions and Dependencies

- Users have internet access and a compatible web browser
- The backend server and database are available
- JWT authentication library is properly configured
- Passwords are stored using secure hashing mechanisms

## 3. System Features and Functional Requirements

### 3.1. Feature 1: User Registration

Description: Allows a new user to create an account by providing valid credentials.

Functional Requirements:

- The system shall allow users to register using a username, email, and password
- The system shall validate user input before saving
- The system shall store passwords in encrypted form

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

Description: Allows registered users to log in securely and terminate sessions when logging out.

Functional Requirements:

- The system shall authenticate users using email and password
- The system shall generate a token upon successful login
- The system shall invalidate the token upon logout

## 4. Non-Functional Requirements

**Security:** Passwords must be encrypted and tokens must be securely generated

**Performance:** Authentication responses must be processed within acceptable time limits

**Usability:** The interface must be simple and easy to understand

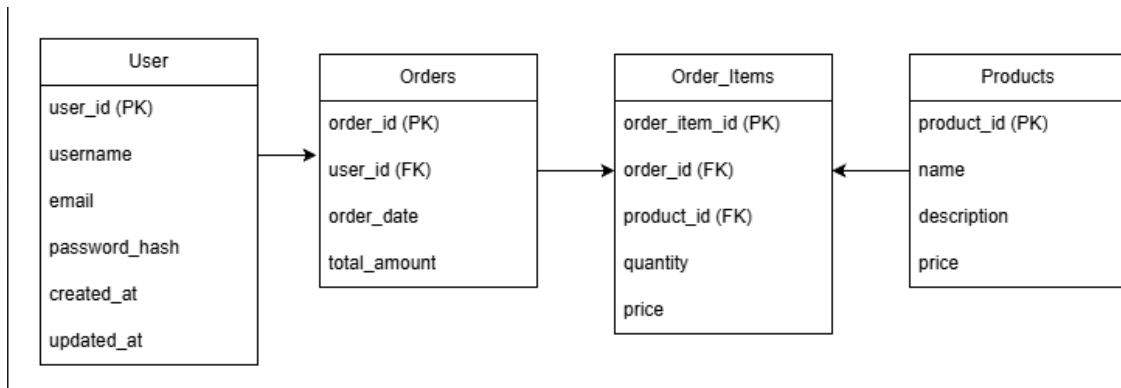
**Reliability:** System must handle incorrect credentials gracefully

**Scalability:** System should support future user growth

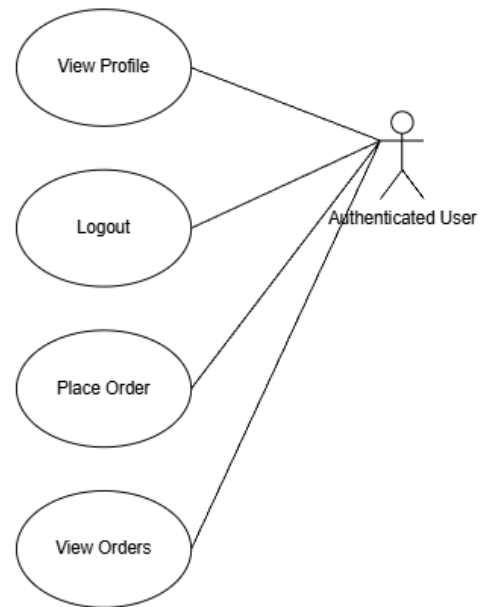
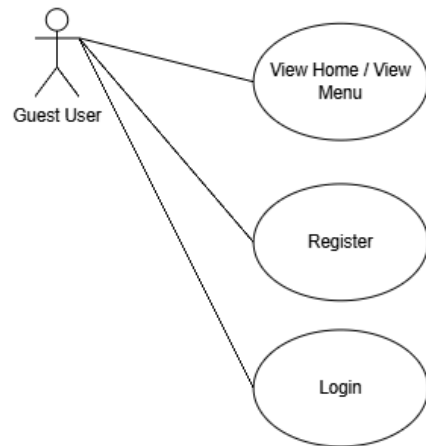
## 5. System Models (Diagrams)

*Insert the necessary diagrams for the system:*

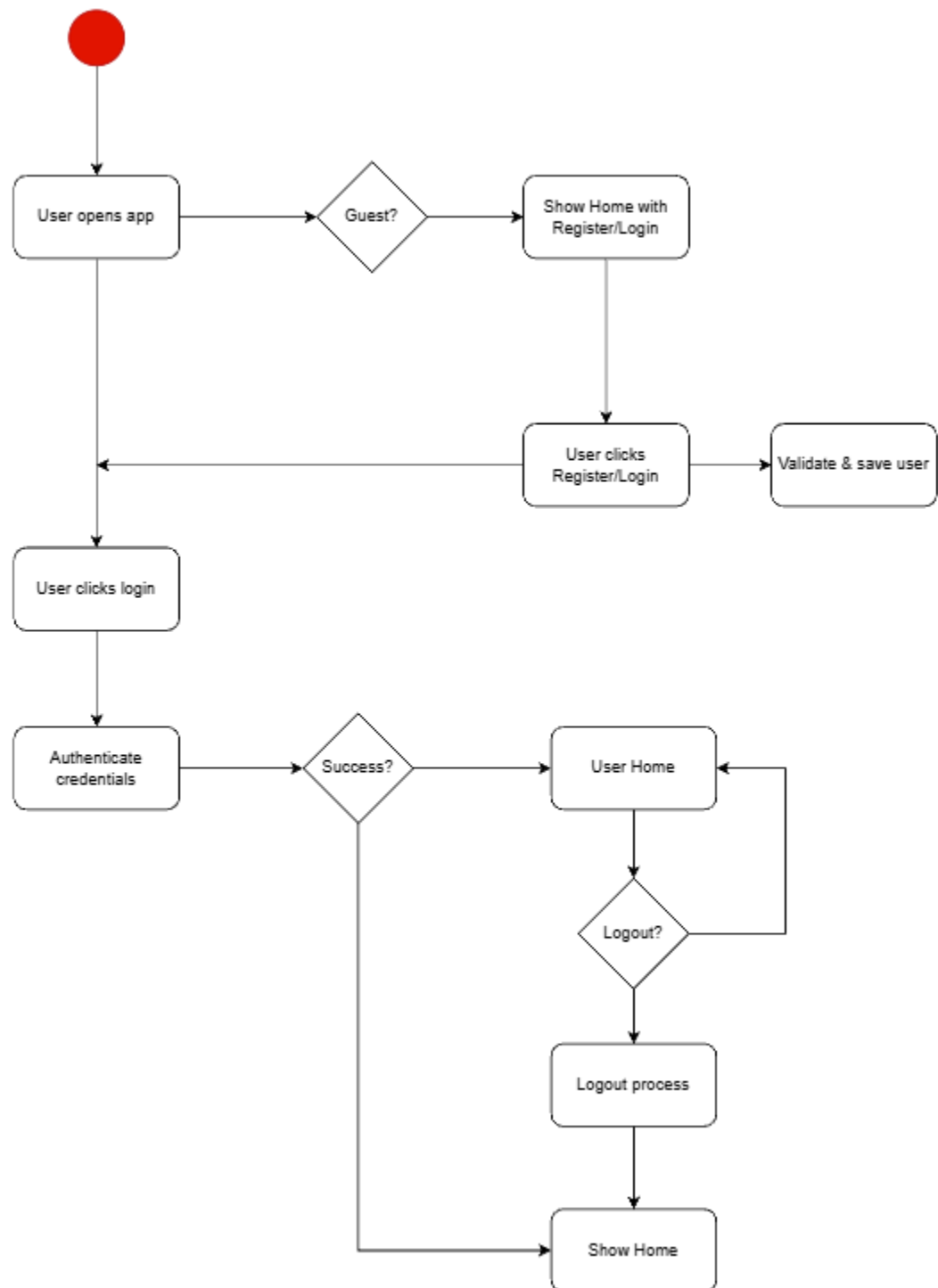
### 5.1. ERD



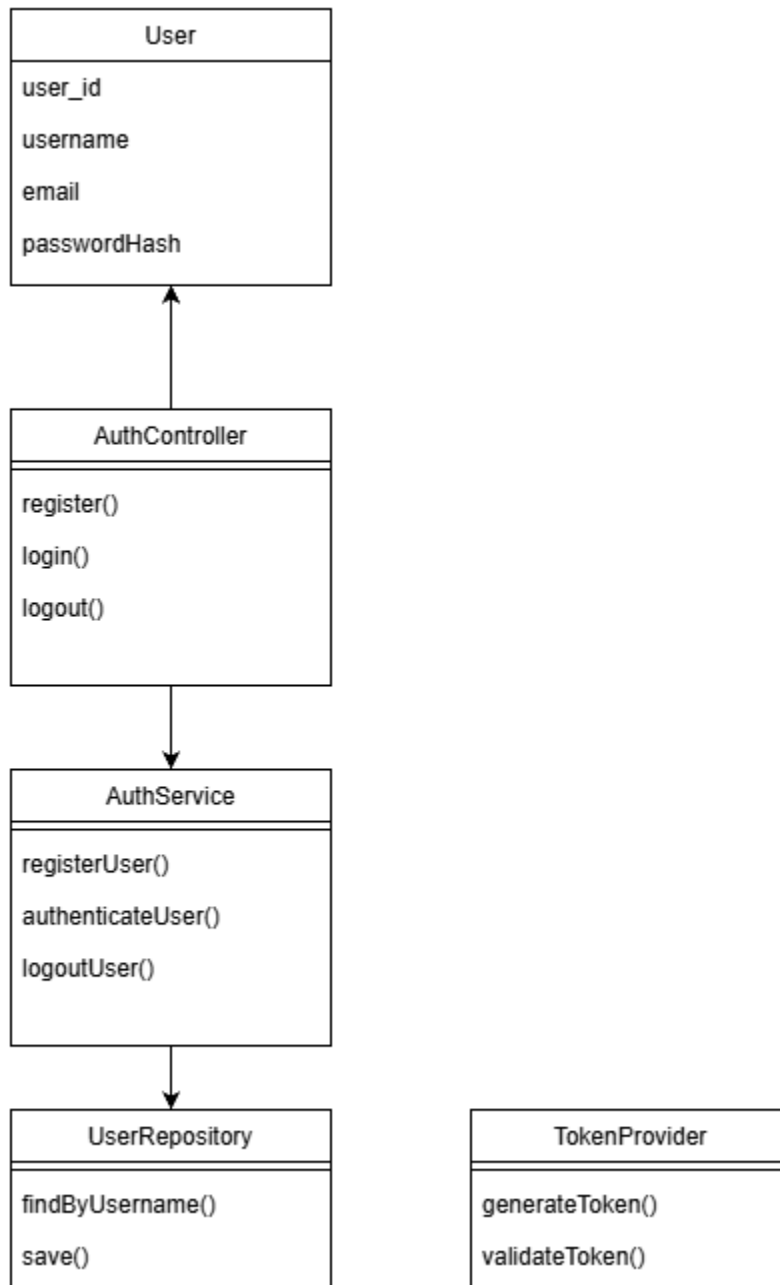
## 5.2. Use Case Diagram



### 5.3. Activity Diagram

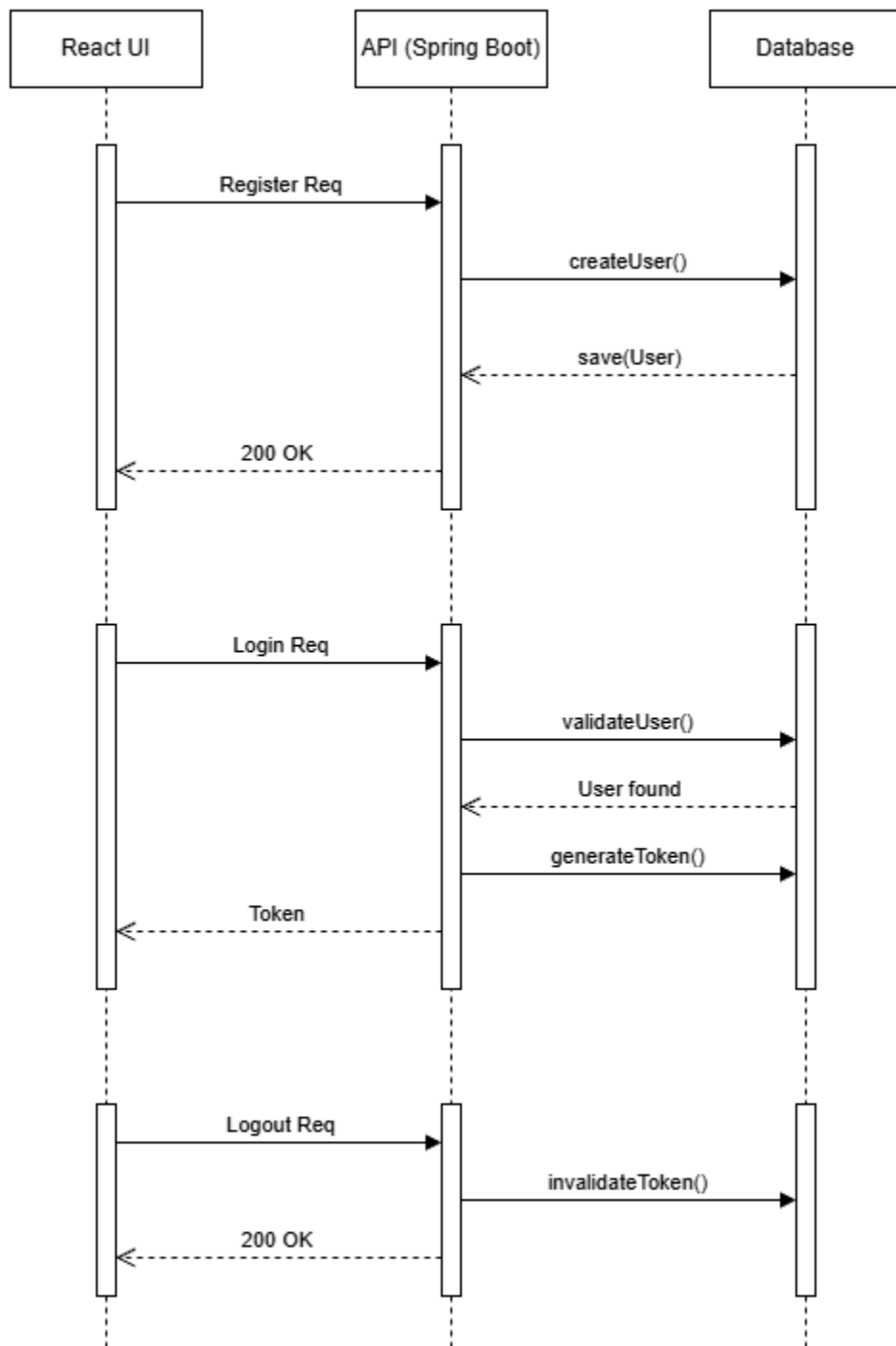


#### 5.4. Class Diagram



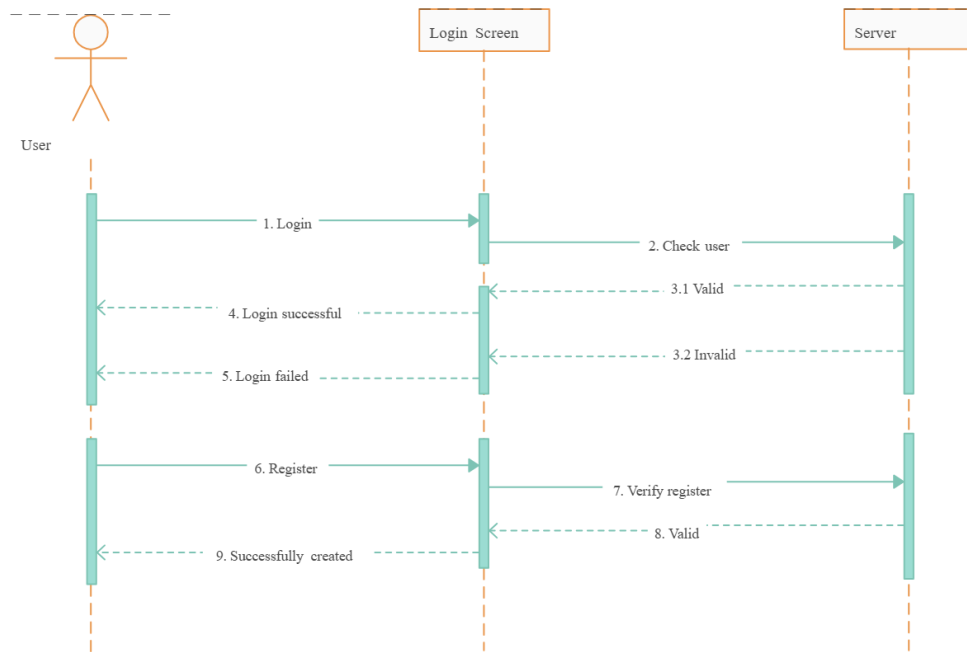


### 5.5. Sequence Diagram



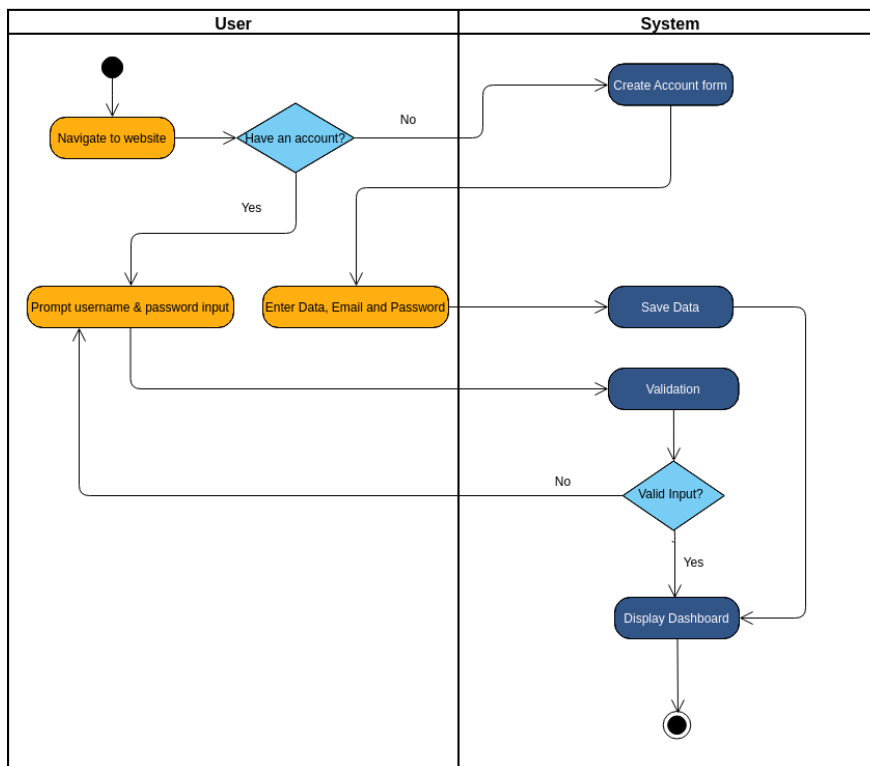
## 6. Appendices

### Appendix 1: Sequence Diagram

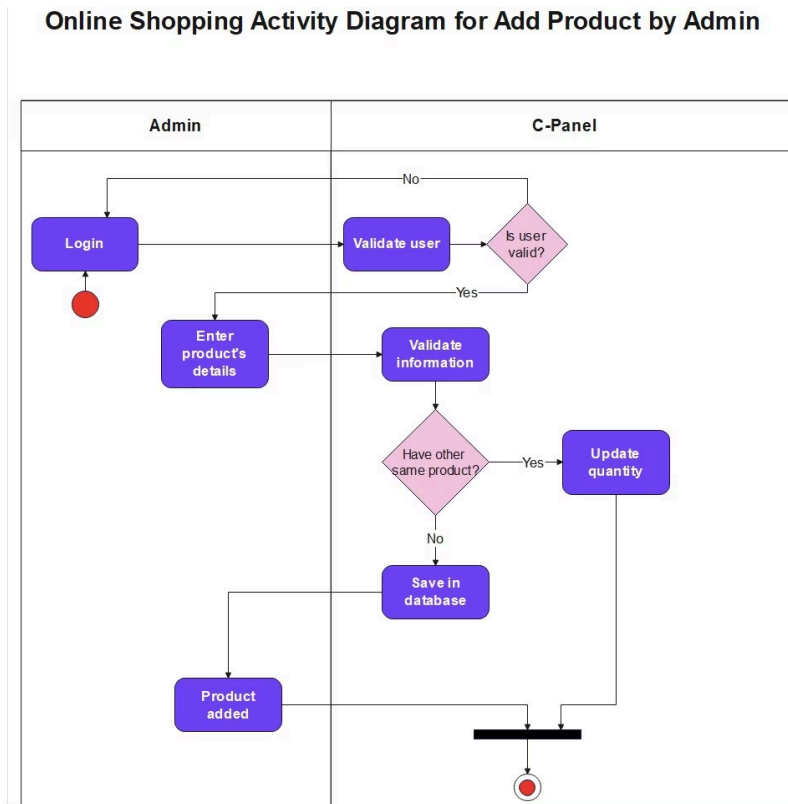


### Appendix 2: Activity Diagram

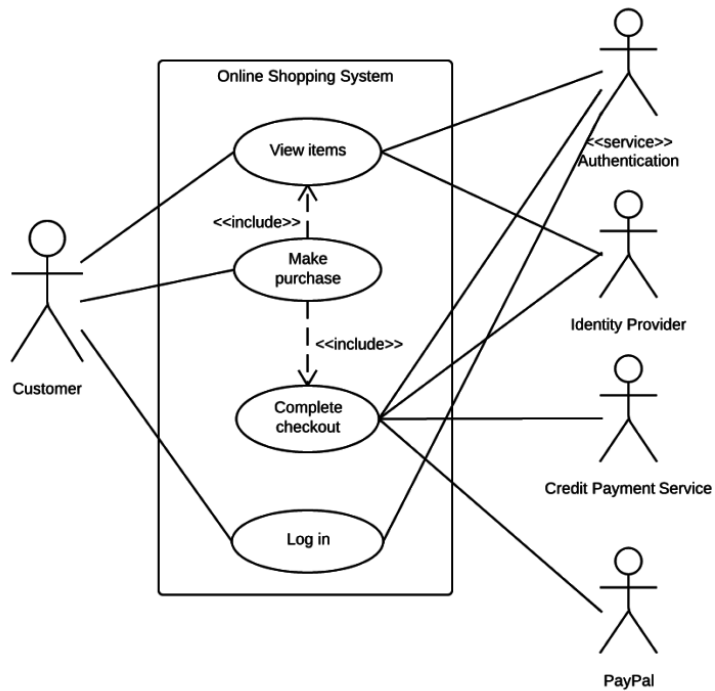
#### CBZ Website User Login and Registration Activity Diagram



### Appendix 3: Activity Diagram (2)



### Appendix 4: Use Case Diagram



## Appendix 5: ERD (Database Schema)

