



**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: KIN

Prepared By: Naranjo, Ana Claire Ellen R.

Date of Submission: February 06, 2026

Version: Draft

# 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 system is to provide a secure, centralized authentication gateway for the Knowledge Institutional Network (KIN). This document is intended for project evaluators and developers to understand the integration between the Spring Boot backend and the ReactJS frontend.

### 1.2. Scope

The system handles user registration, secure login, and session management. It is bounded by institutional email validation (@cit.edu) and provides a protected dashboard for authenticated students.

### 1.3. Definitions, Acronyms, and Abbreviations

- **KIN:** Knowledge Institutional Network.
- **BCrypt:** The hashing algorithm used for password encryption.
- **CORS:** Cross-Origin Resource Sharing, enabled to allow React to communicate with Spring Boot.
- **REST:** Representational State Transfer, the architecture for the API endpoints.

## 2. Overall Description

### 2.1. System Perspective

KIN operates as a decoupled full-stack application. The Spring Boot backend acts as a RESTful API provider, while the ReactJS frontend serves as the user interface.

### 2.2. User Classes and Characteristics

**Student:** The primary user who must register with an institutional email to access the network dashboard.

### 2.3. Operating Environment

- **Backend:** Java Spring Boot, MySQL Database.
- **Frontend:** ReactJS, Axios for API calls.
- **Styling:** CSS with a Maroon and Gold institutional theme.

### 2.4. Assumptions and Dependencies

- The system assumes the user has a valid @cit.edu email address.
- Connectivity to a MySQL instance named kin\_db is required.

## 3. System Features and Functional Requirements

### 3.1. Feature 1: Secure User Registration

Description: Allows new users to create an account specifically within the institutional domain.

Functional Requirements:

- The system shall validate that the email ends with @cit.edu.
- The system shall encrypt passwords using PasswordEncoder before database storage.
- The system shall prevent duplicate registrations for the same email address.

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

Description: Validates credentials and manages the user session.

Functional Requirements:

The system shall verify the email and password against the MySQL database.

- The system shall store the user's email in localStorage upon successful login.
- The logout function shall clear the localStorage and redirect the user to the login page.

### 3.3 Feature 3: Protected Dashboard

Description: A restricted area accessible only to logged-in users.

Functional Requirements:

The system shall verify the email and password against the MySQL database.

- The system shall display the logged-in user's email and role.
- The system shall use React Router to prevent unauthenticated access to the /dashboard route.

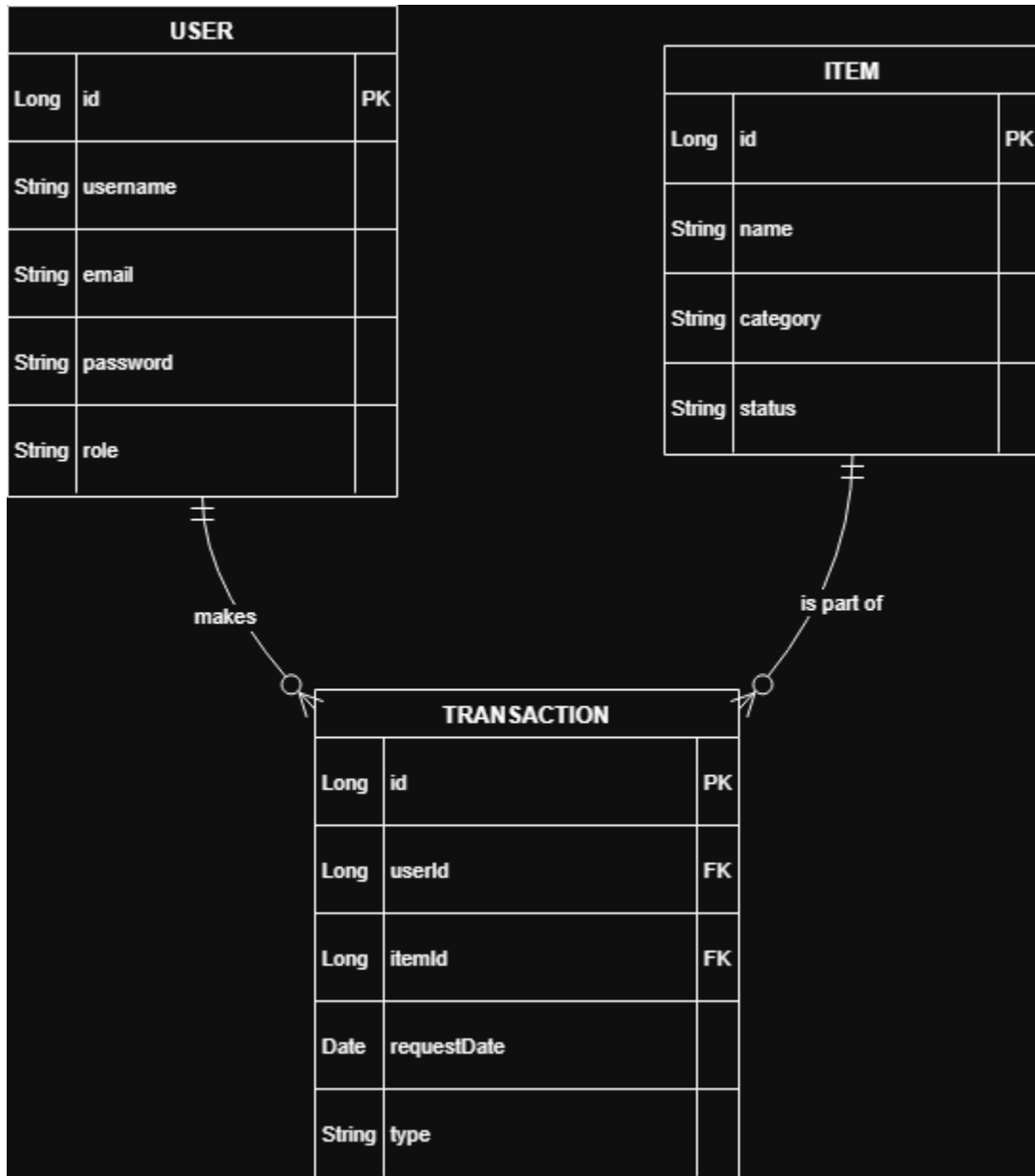
## 4. Non-Functional Requirements

- **Security:** Passwords must never be stored in plain text.
- **Usability:** The interface must adhere to the Maroon and Gold institutional color scheme.
- **Performance:** API authentication responses should be processed in under 2 seconds.

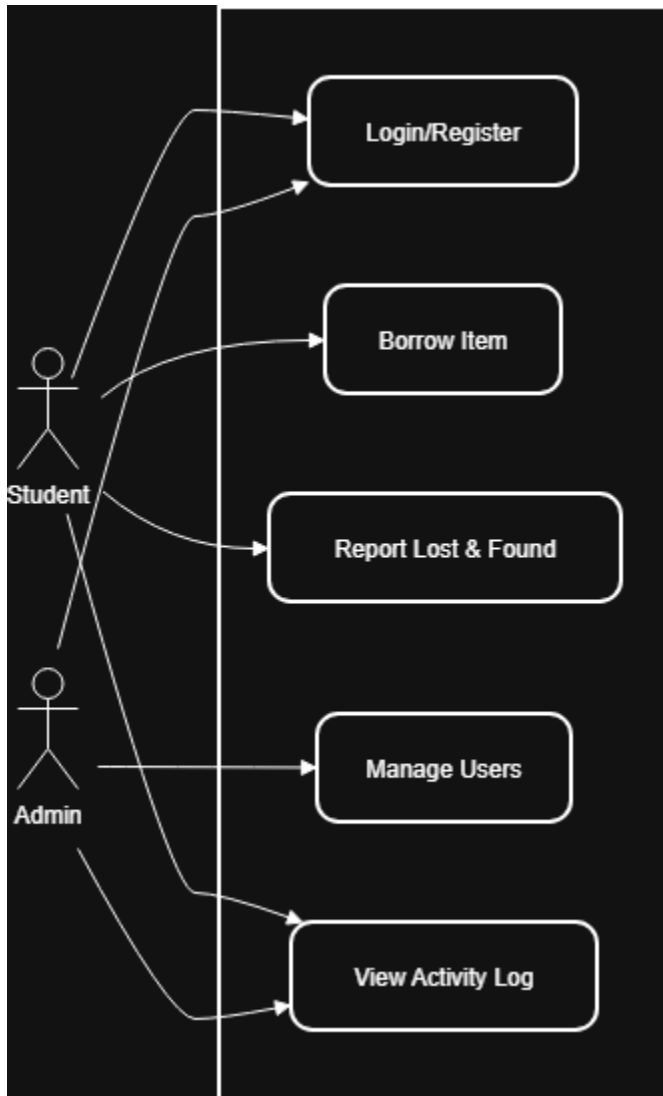
## 5. System Models (Diagrams)

*Insert the necessary diagrams for the system:*

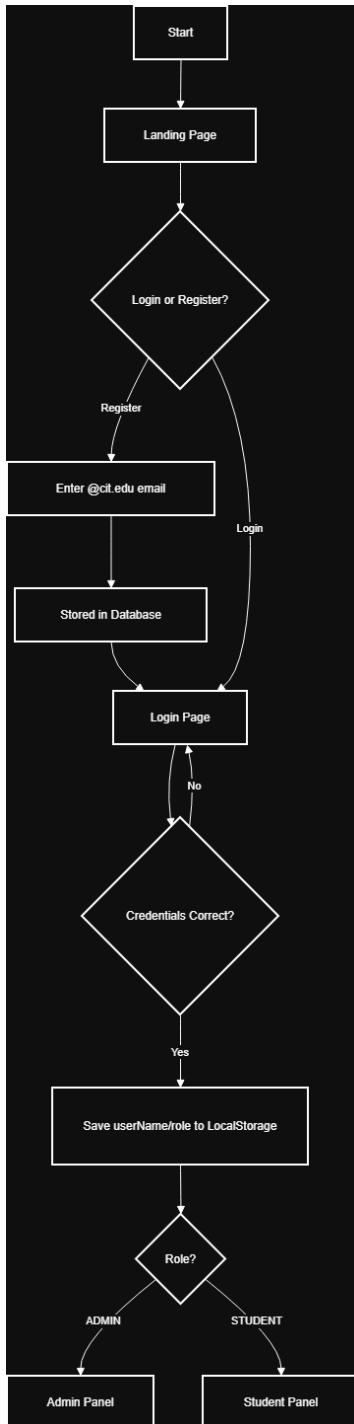
### 5.1. ERD



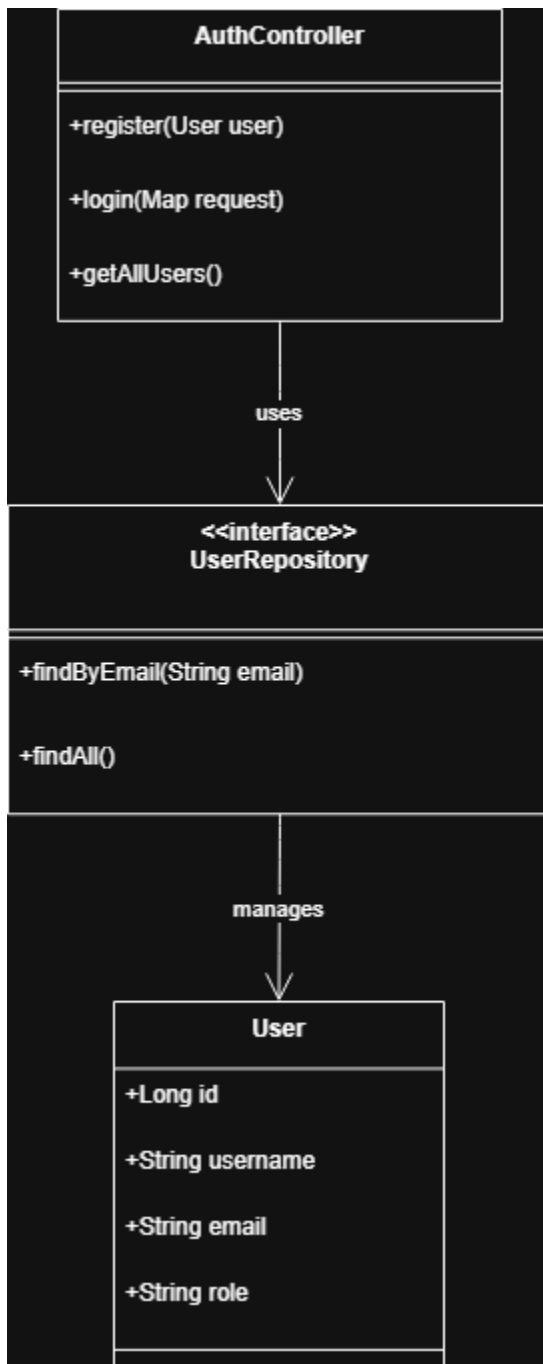
## 5.2. Use Case Diagram



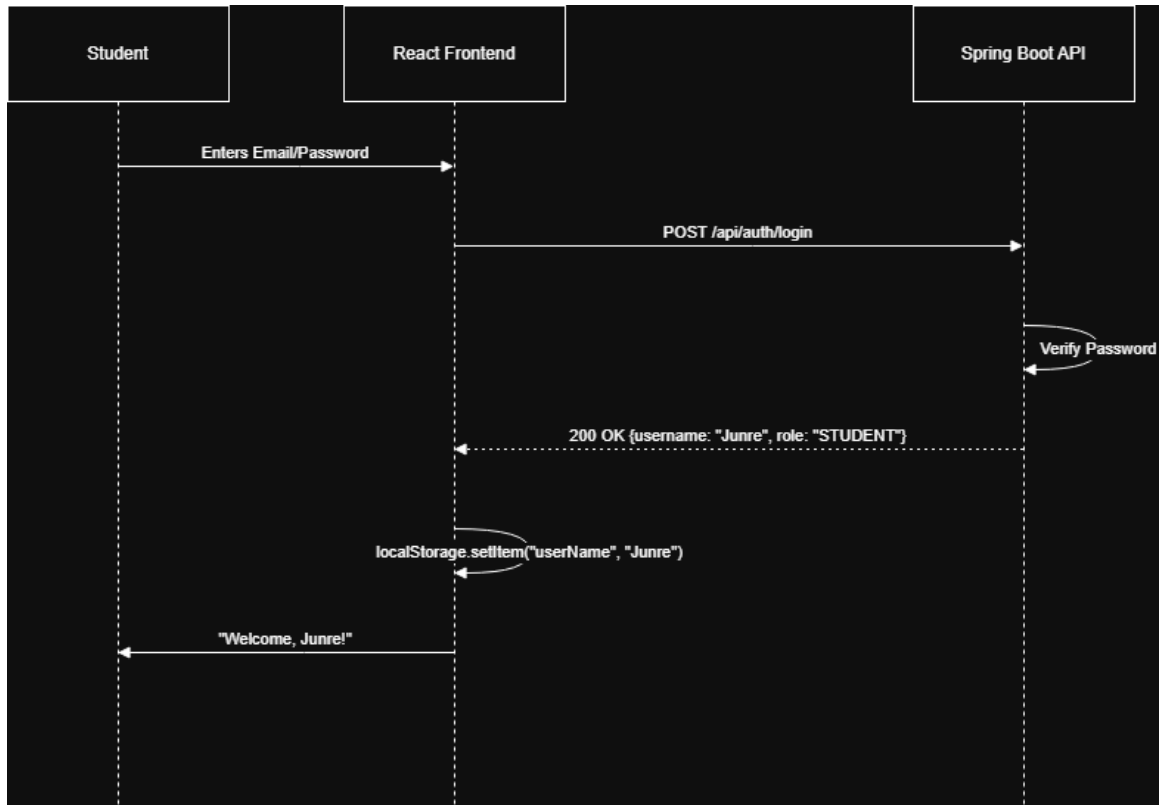
### 5.3. Activity Diagram



#### 5.4. Class Diagram



### 5.5. Sequence Diagram



## 6. Appendices

### 1. UI Color Palette

To ensure the "CIT-U" brand remains consistent as you add more features:

- **Primary Maroon:** #8B4444 (Used for buttons and headings).
- **Secondary Gold:** #C5A059 (Used for icons and borders).
- **Soft Background:** #F4F0F0 (Used to reduce eye strain on dashboards).

### 2. API Endpoints

Endpoint	Method	Purpose
/api/register	POST	Creates a new student account.

/api/auth/login	POST	Verifies identity and sends back the <b>username</b> .
/api/users	GET	(Admin Only) Retrieves the list of all registered users