CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-G4
SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: Quantix

Prepared By: Shinely Marie R. Embalsado

Date of Submission: 02/11/2026

Version: 1

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
   3.3. Feature 3:....3
   3.4. Feature 4:....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

This document serves as the functional and design specification for the Quantix system. It outlines the system's architecture, functional requirements, and non-functional constraints to guide development and ensure alignment with CIT-U laboratory operations.

1.2. Scope

In Scope:

- Microsoft OAuth login and register using institutional email
- Allow users to view and update their personal profile information
- Role-based access control for NAS and laboratory coordinators
- Allow users to view their transaction history
- Enable Non-Academic Scholars (NAS) to process inventory transactions
- Asset categorization, condition, and status tracking
- Allow laboratory coordinators to add, update, and manage laboratory equipment records
- Maintain records of equipment, users, and transaction history

Out of Scope:

- Handle procurement or purchasing of equipment
- Email or SMS notifications
- Perform physical inspection or maintenance of equipment
- Financial tracking or accounting features

1.3. Definitions, Acronyms, and Abbreviations

- **Quantix** – The laboratory equipment inventory management system
- **CIT-U** – Cebu Institute of Technology – University
- **User** – Any individual who interacts with the system
- **NAS** – Non-Academic Scholars assigned to laboratory offices
- **Laboratory Coordinator** – Faculty or staff responsible for overseeing laboratory operations
- **API** – Application Programming Interface
- **OAuth** – Open standard for access delegation used for authentication

2. Overall Description

2.1. System Perspective

Quantix is a centralized inventory management system that operates within CIT-U. It integrates with Microsoft OAuth for authentication and uses a three-tier architecture consisting of:

- Backend services (business logic and data management)
- Web application for coordinators
- Mobile application for NAS

The system communicates through RESTful APIs and is accessible via secure internet connections within the university environment.

2.2. User Classes and Characteristics

- **Non-Academic Scholars (NAS)**
  - Handle inventory transactions
  - Update equipment availability status
  - View user transaction records
  - Moderate system proficiency
- **Laboratory Coordinators**
  - Add, update, and manage laboratory equipment records
  - Monitor transactions
  - View system-wide transaction reports and logs
  - Monitoring and supervisory role

2.3. Operating Environment

Backend: Spring Boot (Java)
Web Application: React (JS)
Mobile Application: Android
Database: Relational databaseAuthentication: Microsoft OAuth
Client Devices: Desktop, laptop, and Android mobile devices
Network: Stable internet connection within CIT-U

2.4. Assumptions and Dependencies

- Users possess valid CIT-U institutional Microsoft accounts
- Microsoft OAuth services are available and operational
- NAS personnel are officially assigned by the university
- Laboratory coordinators have monitoring authority only
- Stable internet connectivity is available within CIT-U
- Equipment data is properly maintained and updated

3. System Features and Functional Requirements

3.1. Feature 1: Authentication and Account Access

Description: Allows users to securely access the system using institutional credentials.
Functional Requirements:

- Users shall be able to log in using Microsoft OAuth
- The system shall validate institutional email accounts
- Access restrictions shall be applied based on user roles

3.2. Feature 2: User Profile Management

Description: Allows users to manage their personal account information and view activity.
Functional Requirements:

- View their profile information
- Update allowed profile details
- View their transaction history
- Users shall not modify role or authentication credentials

3.3. Feature 3: Equipment Management

Description: Allows laboratory coordinators to manage laboratory equipment records.
Functional Requirements:

- Coordinators shall add new equipment records
- Coordinators shall update equipment details
- The system shall store equipment information

3.4. Feature 4: Transaction Processing and Monitoring

Description: Allows NAS to process transactions and coordinators to monitor them.
Functional Requirements:

- The system shall log all transactions with timestamps and user details
- NAS and coordinators shall view transaction logs and reports

4. Non-Functional Requirements

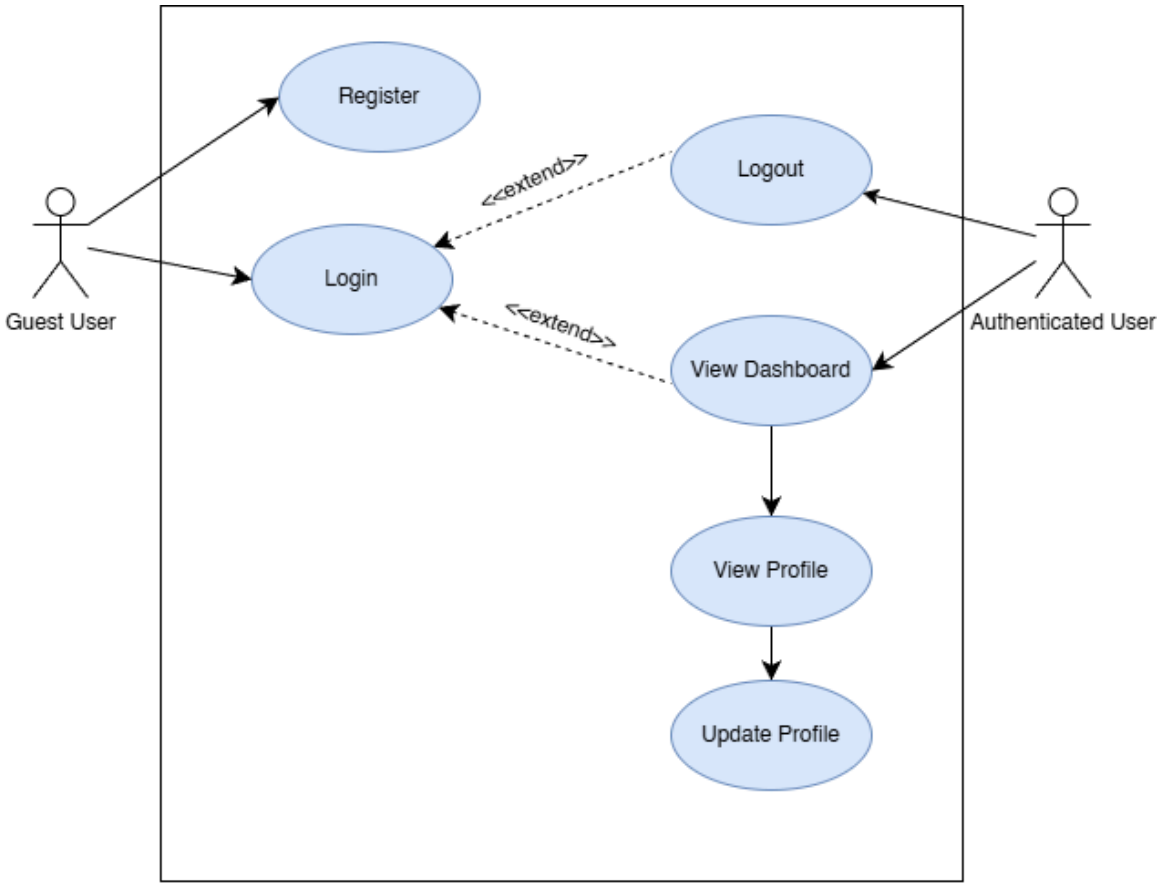
4.1. **Performance:** The system shall respond to user actions within acceptable time limits.
4.2. **Security:** The system shall use Microsoft OAuth, enforce role-based access control, and protect user and transaction data.
4.3. **Usability:** The system shall provide a clear and user-friendly interface across platforms.
4.4. **Reliability:** The system shall maintain accurate and consistent records.
4.5. **Availability:** The system shall be accessible during laboratory operating hours.

5. System Models (Diagrams)

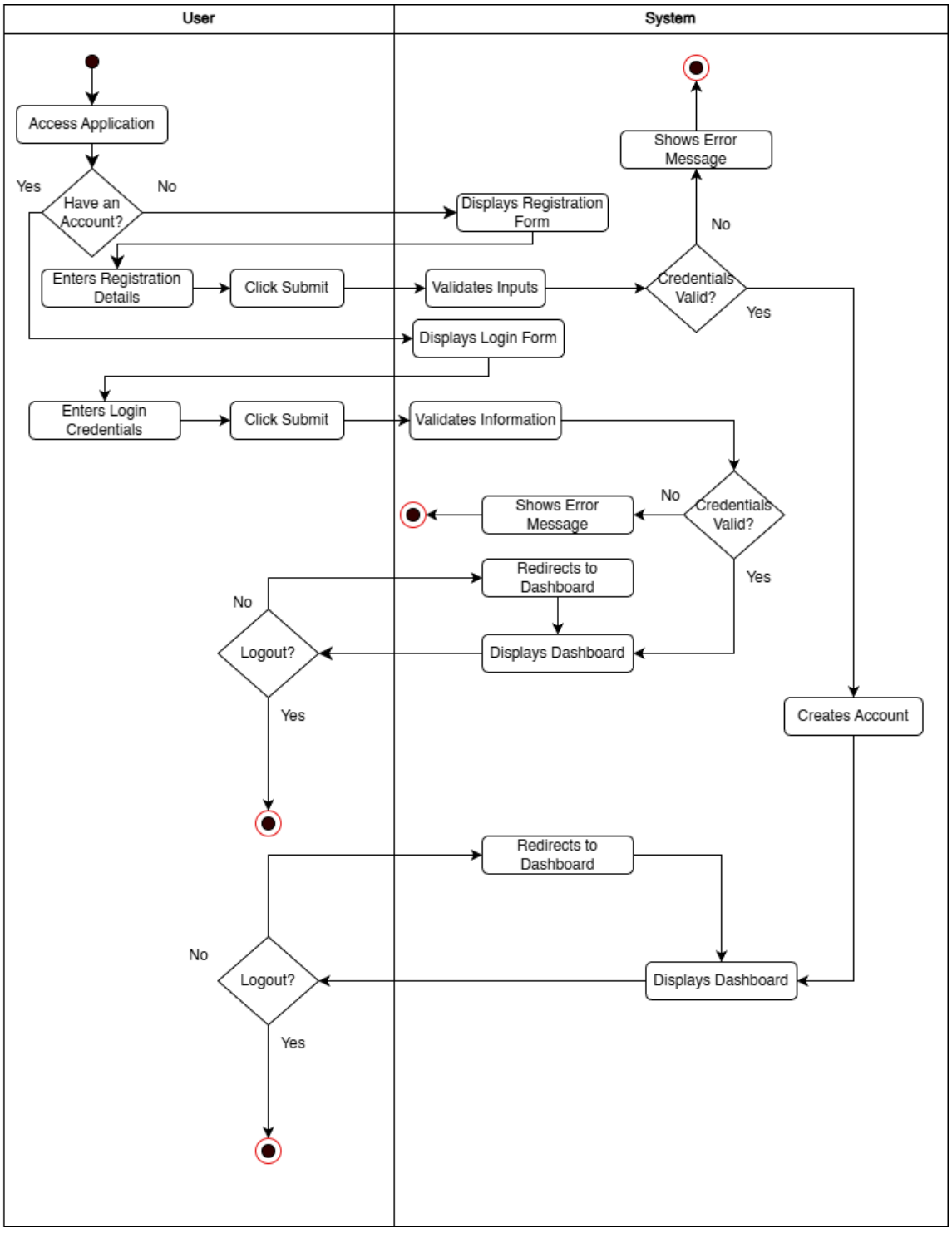
5.1. ERD

User
(PK) id: BIGINT
email: VARCHAR(255) UNIQUE NOT NULL
username: VARCHAR(100) UNIQUE NOT NULL
password: VARCHAR(255) NOT NULL
first_name: VARCHAR(70)
last_name: VARCHAR(70)
avatar: VARCHAR(255)
role_id: BIGINT FOREIGN KEY REFERENCES user_roles(id)
is_active: BOOLEAN DEFAULT TRUE
created_at: TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
updated_at: TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP

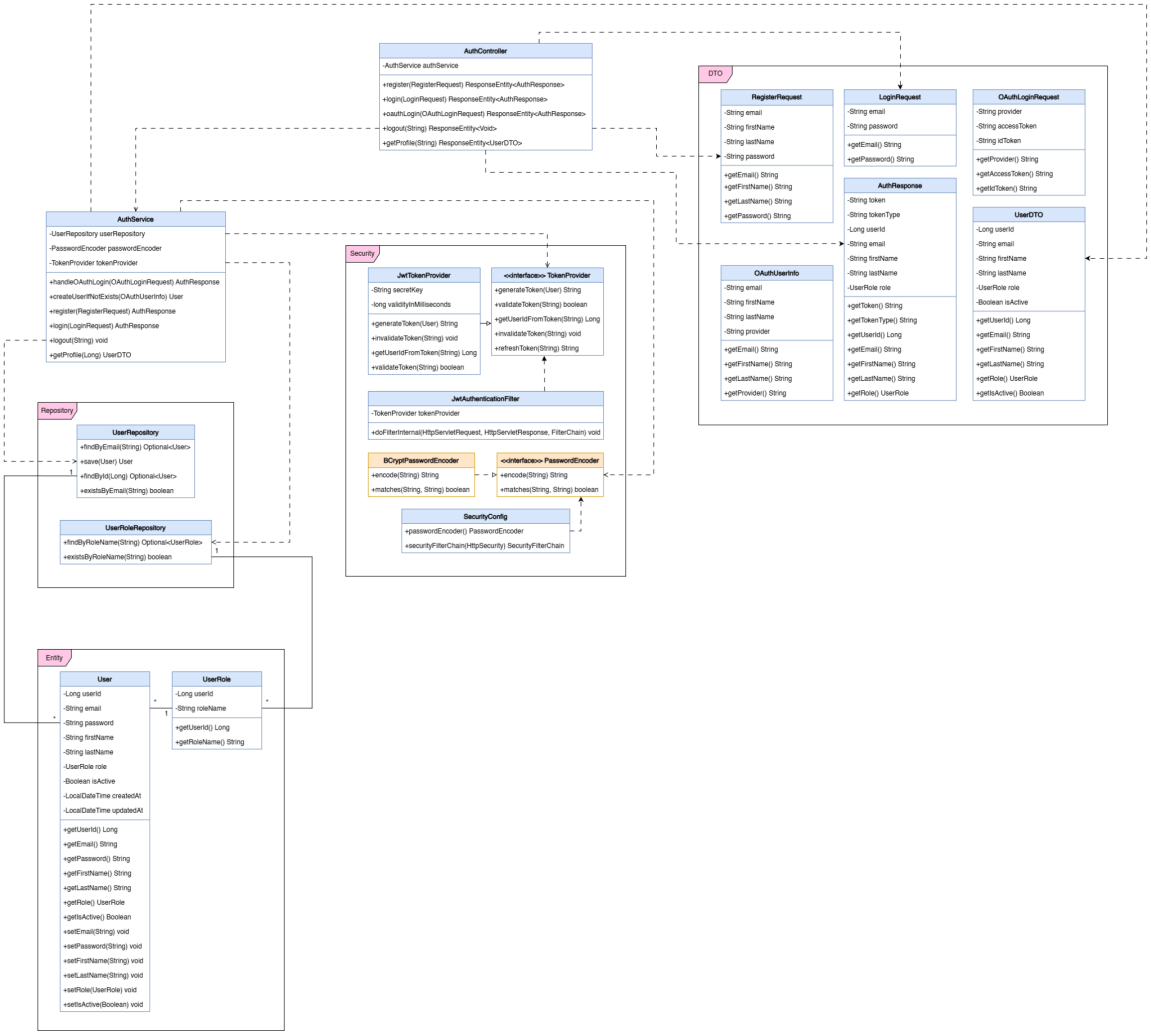
5.2. Use Case Diagram



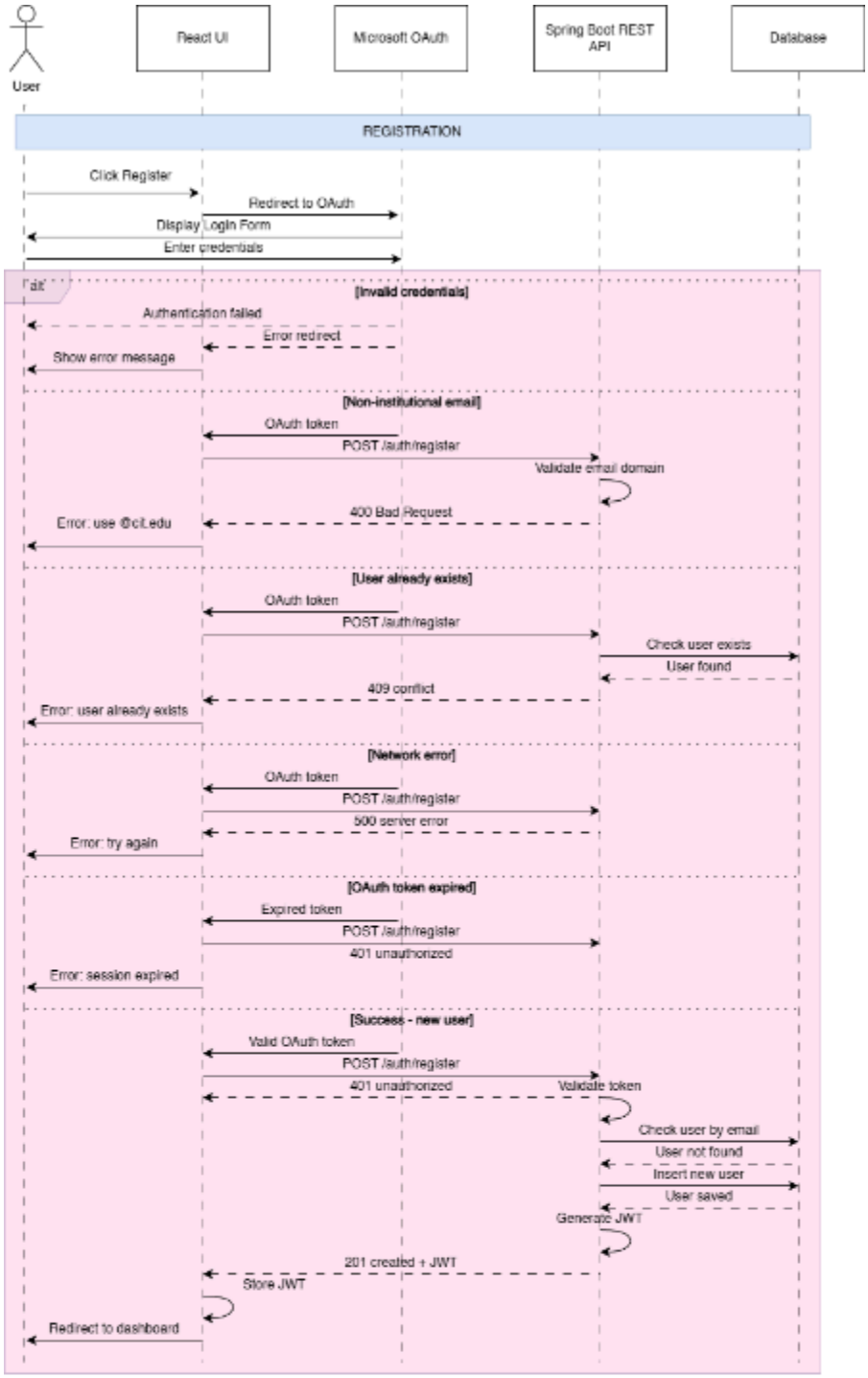
5.3. Activity Diagram

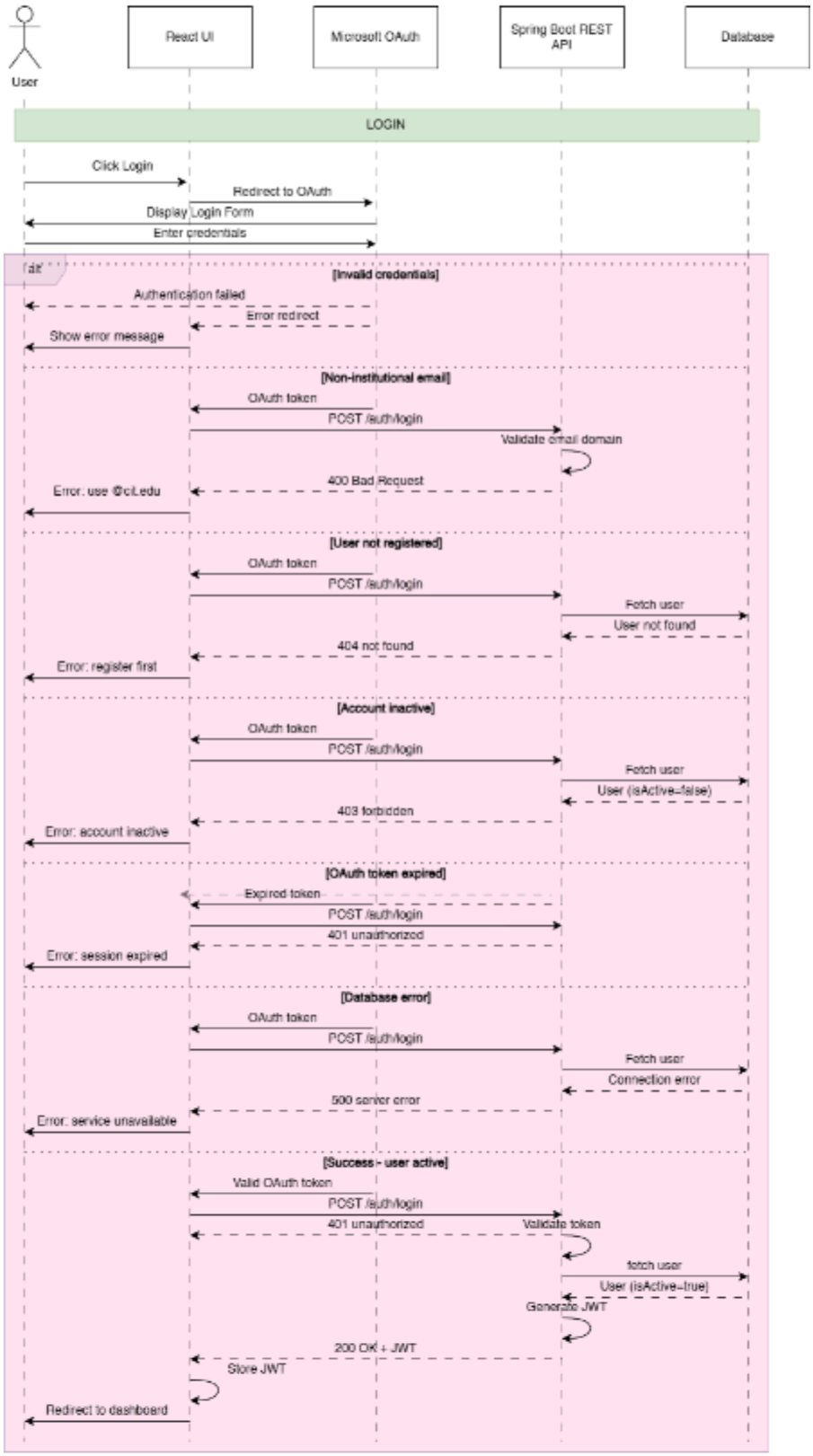


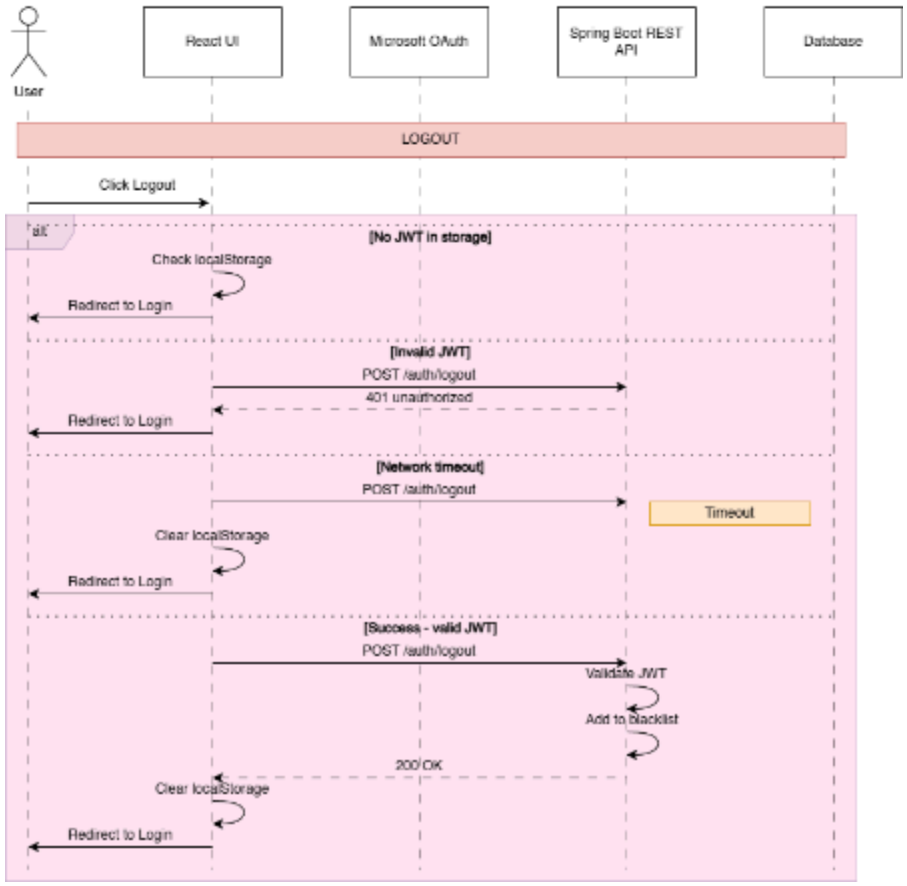
5.4. Class Diagram



5.5. Sequence Diagram




User
React UI
Microsoft OAuth
Spring Boot REST API
Database
LOGIN
Click Login
Redirect to OAuth
Display Login Form
Enter credentials
alt
[Invalid credentials]
Authentication failed
Error redirect
Show error message
[Non-institutional email]
OAuth token
POST /auth/login
Validate email domain
400 Bad Request
Error: use @ci.edu
[User not registered]
OAuth token
POST /auth/login
Fetch user
User not found
404 not found
Error: register first
[Account inactive]
OAuth token
POST /auth/login
Fetch user
User (isActive=false)
403 forbidden
Error: account inactive
[OAuth token expired]
Expired token
POST /auth/login
401 unauthorized
Error: session expired
[Database error]
OAuth token
POST /auth/login
Fetch user
Connection error
500 server error
Error: service unavailable
[Success - user active]
Valid OAuth token
POST /auth/login
401 unauthorized
Validate token
fetch user
User (isActive=true)
Generate JWT
200 OK + JWT
Store JWT
Redirect to dashboard



6. Appendices