Software Design Specifications

Flex Student Portal

Version: 1.0

# Project Code

Not Applicable

# Supervisor

Ms Fizza Mansoor

# Co Supervisor

Not Applicable

# Project Team

22k5024 Hadi  
22k5018 Aheed  
22k5144 Lucky

# Submission Date

4/28/2025

# Document History

| Version | Name of Person | Date | Description of change |
| --- | --- | --- | --- |
| 1.0 | 22k5024 Hadi | 12-5-2024 | Document Created |

# Distribution List

| Name | Role |
| --- | --- |
| Ms Fizza Mansoor | Supervisor |
| Co Supervisor | Co Supervisor |

# Document Sign-Off

| Version | Sign-off Authority | Project Role | Signature |
| --- | --- | --- | --- |
| 1.0 | Supervisor | Project Supervisor | [Signature] |

# Document Information

| Customer | FAST-NU |
| --- | --- |
| Project | Flex Student Portal |
| Document | Software Design Specification |
| Document Version | 1.0 |
| Status | Completed |
| Author(s) | 22k5024 Hadi, 22k5018 Aheed, 22k5144 Lucky |
| Distribution | Advisor, Project Coordinator’s Office (through Advisor) |

### Table of Contents

1. Introduction  
     1.1 Purpose of Document  
     1.2 Intended Audience  
     1.3 Document Convention  
     1.4 Project Overview  
     1.5 Scope
2. Design Considerations  
     2.1 Assumptions and Dependencies  
     2.2 Risks and Volatile Areas
3. System Architecture  
     3.1 System Level Architecture  
     3.2 Software Architecture
4. Design Strategy
5. Detailed System Design  
     5.1 Database Design  
      5.1.1 ER Diagram  
      5.1.2 Data Dictionary  
     5.2 Application Design  
      5.2.1 Sequence Diagram  
      5.2.2 State Diagram  
      5.2.3 Activity Diagram
6. References
7. Appendices  
     - Error Handling Terms  
     - Communication Diagram  
     - Entity Relationship Diagram  
     - Use Case Diagram

# 1 Introduction

## 1.1 Purpose of Document

This document outlines the software design specifications for the Flex Student Portal project. It provides a detailed description of the intended functionality and the Object-Oriented design methodology that will be used. The document is intended for developers, project managers, supervisors, and future maintainers of the system.

## 1.2 Intended Audience

This document is intended for:  
- Developers  
- Project Managers  
- Academic Supervisors  
- Students and Teachers  
- Documentation Writers

## 1.3 Document Convention

Headings: Arial, Bold, 14pt  
Body Text: Cambria, Regular, 12pt

## 1.4 Project Overview

The Flex Student Portal is an online platform facilitating communication between students and teachers. Core features include attendance management, timetable viewing, course registration, marks distribution, transcript generation, and feedback collection. The system will use a modern web stack with ReactJS frontend, Node.js backend, and PostgreSQL database.

## 1.5 Scope

The Flex Student Portal will:  
- Provide student-teacher interaction modules  
- Manage attendance, marks, timetable, transcript  
- Allow secure login and password management  
- Enable feedback collection  
- Exclude administrative functionalities (reserved for future phases)

# 2 Design Considerations

## 2.1 Assumptions and Dependencies

Users will have internet access and compatible devices. System depends on availability of ReactJS, Node.js libraries, and PostgreSQL database. Institutional IT infrastructure must support server deployment.

## 2.2 Risks and Volatile Areas

Changes in course structure or evaluation criteria may require frequent system updates. Security vulnerabilities in third-party libraries. Potential server downtime or cloud hosting changes. Future integration with other institutional systems (LMS, SIS).

# 3 System Architecture

## 3.1 System Level Architecture

Frontend: ReactJS for UI  
Backend: Node.js with Express.js API  
Database: PostgreSQL  
Server: Hosted locally or on cloud (AWS/Azure)  
Major Modules:  
- User Authentication  
- Attendance Management  
- Marks Management  
- Course Registration  
- Transcript Management  
- Feedback Collection

## 3.2 Software Architecture

The system will follow a three-layered architecture:  
- User Interface Layer: Frontend ReactJS Application  
- Middle Tier: Node.js/Express.js handling business logic and APIs  
- Data Access Layer: PostgreSQL database accessed via server APIs



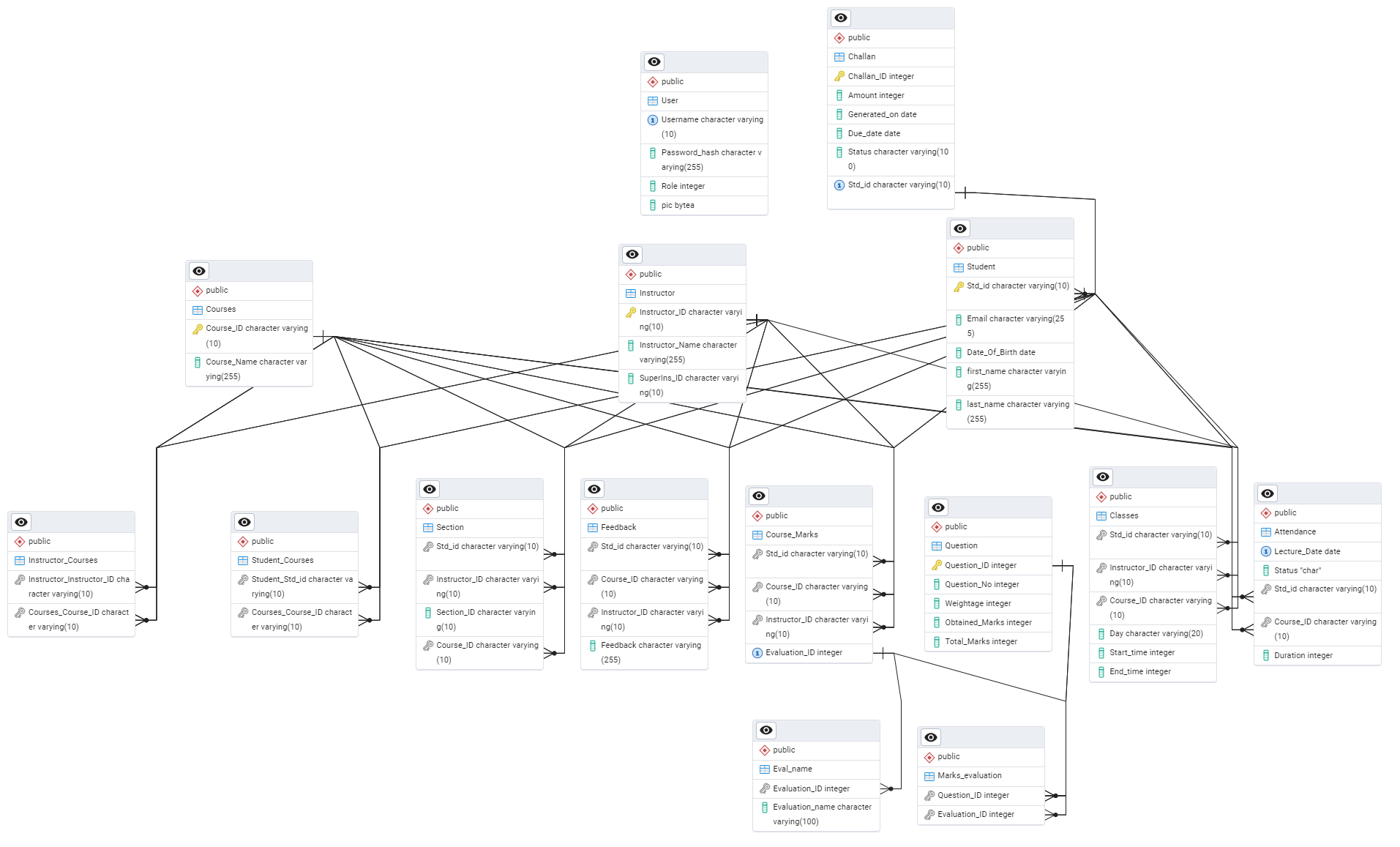
# 4 Design Strategy

Extendibility: New features like administrative controls can be added in future phases.  
System Reuse: Modular components for course and marks management can be reused.  
UI Paradigm: Responsive design principles.  
Data Management: Data persistence through PostgreSQL with regular backups.  
Concurrency: Node.js async handling for multiple simultaneous users.  
Security: Password hashing, SSL/TLS encryption.

# 5 Detailed System Design

## 5.1 Database Design

### 5.1.1 ER Diagram



### 5.1.2 Data Dictionary

### Attendance

| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| attendance\_id | Unique attendance ID | Integer | - | No | None | PK |
| student\_id | Reference to student | Integer | - | No | None | FK |
| class\_id | Reference to class | Integer | - | No | None | FK |
| date | Date of attendance | Date | - | No | None |  |
| status | Attendance status (Present/Absent) | Varchar | 10 | No | None |  |

### Challan

| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| challan\_id | Unique challan ID | Integer | - | No | None | PK |
| student\_id | Reference to student | Integer | - | No | None | FK |
| fee\_amount | Total fee amount | Integer | - | No | None |  |
| due\_date | Due date for payment | Date | - | No | None |  |
| status | Paid/Unpaid status | Varchar | 10 | No | None |  |

### Classes

| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| class\_id | Unique class ID | Integer | - | No | None | PK |
| section\_id | Reference to section | Integer | - | No | None | FK |
| course\_id | Reference to course | Integer | - | No | None | FK |
| class\_day | Day of the week | Varchar | 10 | No | None |  |
| start\_time | Class start time | Time | - | No | None |  |
| end\_time | Class end time | Time | - | No | None |  |

### Course\_Marks

| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| std\_id | Student ID | Integer | - | No | None | PK, FK |
| course\_id | Course ID | Integer | - | No | None | PK, FK |
| instructor\_id | Instructor ID | Integer | - | No | None | FK |
| evaluation\_id | Evaluation ID | Integer | - | No | None | FK |

### Courses

| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| course\_id | Unique course ID | Integer | - | No | None | PK |
| course\_name | Name of the course | Varchar | 100 | No | None |  |

### Eval\_name

| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| evaluation\_id | Evaluation ID | Integer | - | No | None | PK |
| evaluation\_name | Name of evaluation type | Varchar | 50 | No | None |  |

### Feedback

| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| feedback\_id | Unique feedback ID | Integer | - | No | None | PK |
| student\_id | Student providing feedback | Integer | - | No | None | FK |
| course\_id | Course related to feedback | Integer | - | No | None | FK |
| feedback\_text | Text of feedback | Varchar | 500 | No | None |  |

### Instructor

| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| instructor\_id | Unique instructor ID | Integer | - | No | None | PK |
| instructor\_name | Name of instructor | Varchar | 100 | No | None |  |
| email | Instructor email | Varchar | 100 | No | None |  |

### Instructor\_Courses

| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| instructor\_id | Reference to instructor | Integer | - | No | None | PK, FK |
| course\_id | Reference to course | Integer | - | No | None | PK, FK |

### Marks\_Evaluation

| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| evaluation\_id | Reference to evaluation | Integer | - | No | None | PK, FK |
| question\_id | Reference to question | Integer | - | No | None | PK, FK |

### Question

| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| question\_id | Unique question ID | Integer | - | No | None | PK |
| question\_number | Number of question | Integer | - | No | None |  |
| weightage | Weightage of question | Integer | - | No | None |  |
| obtained\_marks | Marks obtained | Integer | - | Yes | Null |  |
| total\_marks | Total marks | Integer | - | No | None |  |

### Section

| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| section\_id | Unique section ID | Integer | - | No | None | PK |
| section\_name | Section name | Varchar | 20 | No | None |  |

### Student

| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| student\_id | Unique student ID | Integer | - | No | None | PK |
| student\_name | Student name | Varchar | 100 | No | None |  |
| email | Student email | Varchar | 100 | No | None |  |
| section\_id | Reference to section | Integer | - | No | None | FK |

### Student\_Courses

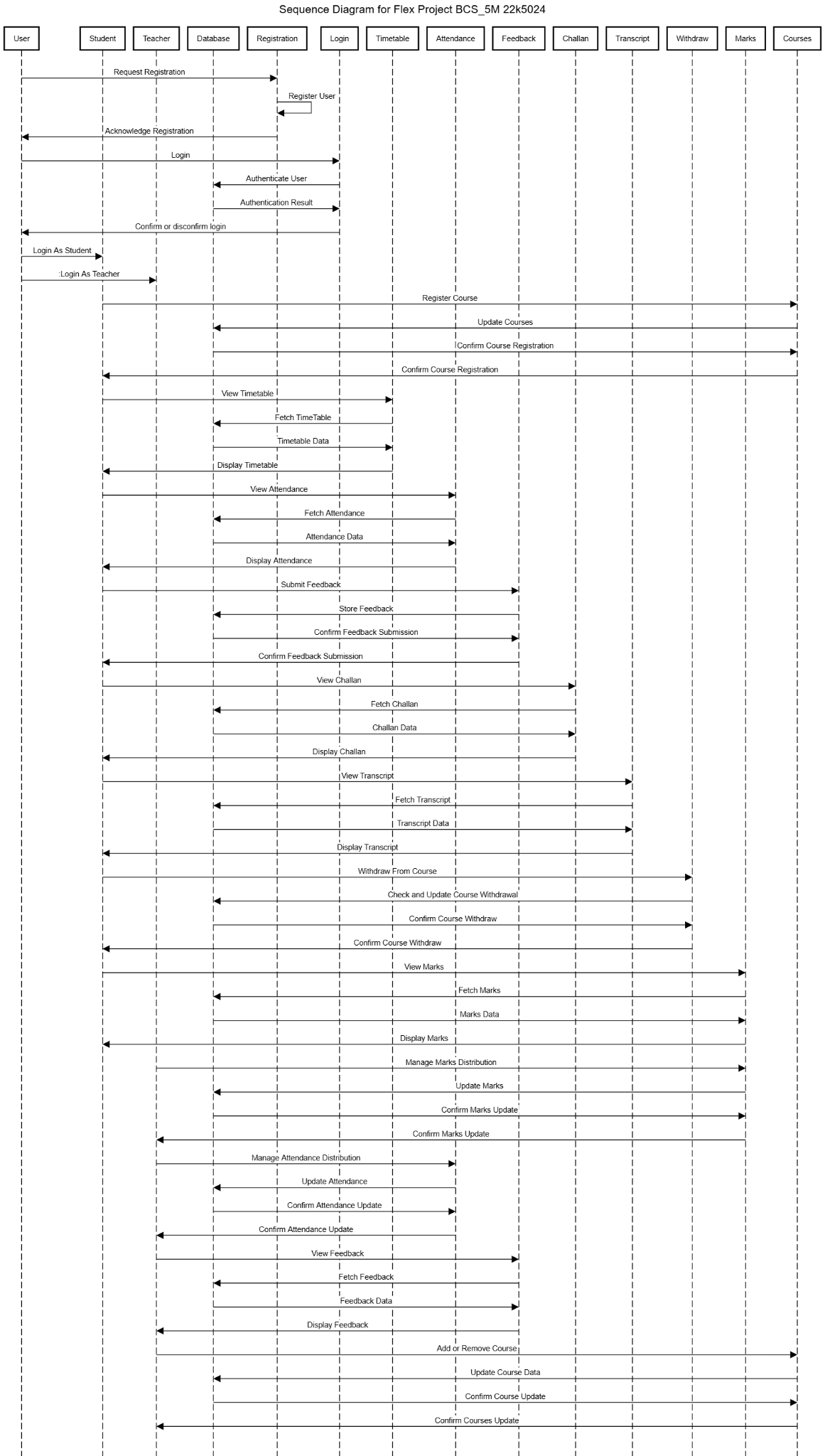
| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| student\_id | Reference to student | Integer | - | No | None | PK, FK |
| course\_id | Reference to course | Integer | - | No | None | PK, FK |

### User

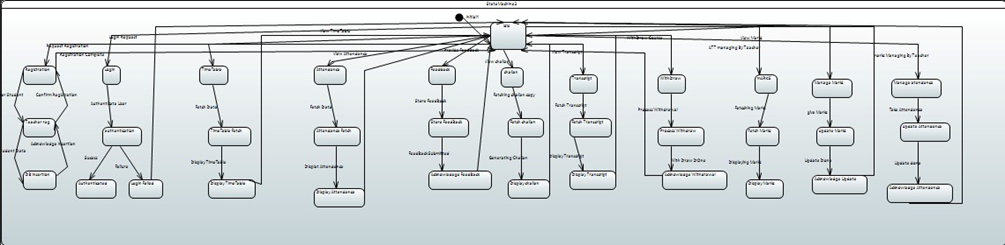
| Column Name | Description | Type | Length | Nullable | Default Value | Key Type |
| --- | --- | --- | --- | --- | --- | --- |
| user\_id | Unique user ID | Integer | - | No | None | PK |
| username | Login username | Varchar | 50 | No | None |  |
| password\_hash | Hashed password | Varchar | 255 | No | None |  |
| profile\_picture | Profile picture (Binary) | Bytea | - | Yes | Null |  |

## 5.2 Application Design

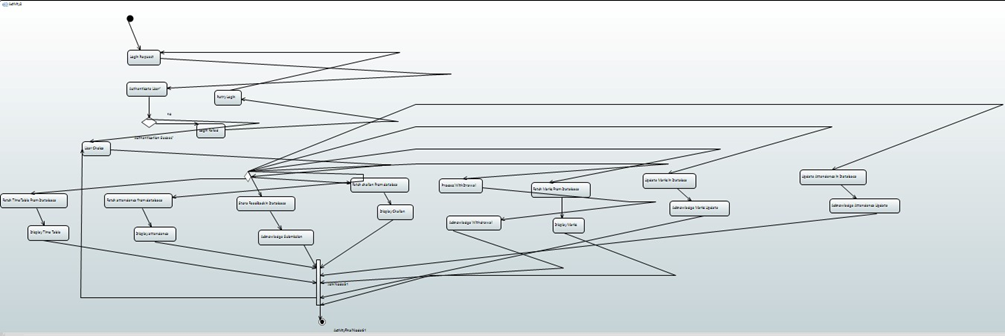
### 5.2.1 Sequence Diagram



### 5.2.2 State Diagram



### 5.2.3 Activity Diagram



# 6 References

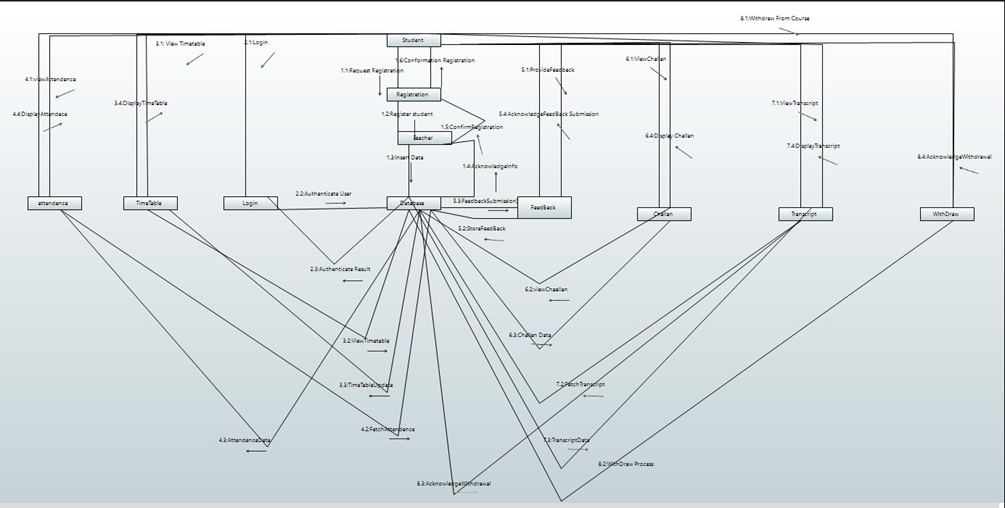
PostgreSQL Documentation  
ReactJS Documentation  
Node.js Official Documentation

# 7 Appendices

## Error Handling Terms:

1. Error 500 (Server Error): Indicates an internal server issue.
2. Error 400 (Bad Request): Indicates an issue with the client request, such as missing or incorrect data.
3. ON CONFLICT: A PostgreSQL clause to handle duplicate key violations during INSERT queries.
4. EXCLUDED: A PostgreSQL keyword used in ON CONFLICT clauses to reference the conflicting rows.

## Communication Diagram:



## Entity Relationship Diagram:

## Use Case Diagram:

