Software Requirements and Design Document

for

University Management System

Prepared by: I222553 M Shuja Uddin, I228759 Amna Hassan

National University Of Computer and Emerging Sciences

June, 06, 2024

Contents

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document specifies the software requirements for University Management System, version 5. This document outlines the functionalities, constraints, and capabilities of the software product. It serves as a comprehensive guide for the development, maintenance, and future updates of the system.

1.1.1 Product Overview

Developed with Scene Builder and JDBC SQL, the University Management System is a JavaFX application intended to effectively handle the essential functions of an administrator, student, and faculty member in a university environment. Enhancing interactions between students and teachers, streamlining administrative procedures, and managing university operations more effectively are the goals of the software.

1.1.2 Scope of the Product

The scope of this SRS covers the entire University Management System, including its core functionalities and features. Specifically, this document addresses the following components and subsystems:

Admin Subsystem:
- Description: Manages administrative tasks such as user management, course management, and fee structure updates.
- Key Features: User registration, course addition, fee management, viewing courses, viewing users.

Student Subsystem:
- Description: Allows students to enroll in courses, view attendance, check fee details, access study plan and view academic progress.
- Key Features: Course enrollment, viewing grades and marks, fee details.

Faculty Subsystem:
- Description: Enables faculty members to manage their courses, interact with students, and update academic records.
- Key Features: Update marks, grade submission, attendance tracking.

This SRS does not cover the following aspects of the University Management System:

- Third-party integration modules
- Hardware specifications

- Future enhancements beyond the current release

1.1.3 Intended Audience

The intended audience for this SRS includes:

- **Developers**: To understand the detailed requirements and functionalities they need to implement.
- **Testers**: To create test cases and plans that validate the software against its requirements.
- **Project Managers**: To plan, monitor, and manage the development process.
- **Stakeholders**: To ensure that the software meets their needs and expectations.

Throughout the development lifecycle, this document acts as a fundamental resource, ensuring that all parties involved have a mutual understanding of the requirements for the system.

1.2 Product Scope

The University Management System is a software solution developed using JavaFX, Scene Builder, and JDBC SQL. Its primary purpose is to streamline university operations by efficiently managing the roles of administrators, students, and faculty members.

1.2.1 Description and Purpose

The system addresses key university needs by:

- **For Administrators**: Managing user accounts, courses, and fee structures.
- **For Students**: Enrolling in courses, viewing attendance, checking fee details, accessing study plan and viewing academic progress.
- **For Faculty Members**: Tracking Attendance, Managing Grades, Updating Marks, and maintaining academic records.

1.2.2 Benefits

- **Efficiency**: Automates administrative tasks, reducing workload and errors.
- **Data Management**: Centralizes data storage and access using Database management.
- **User Interface**: Offers an intuitive interface using JavaFX and scenebuilder.
- **Real-Time Updates**: Provides up-to-date information on courses, grades, and fees.

1.2.3 Objectives

- **Streamline Processes**: Simplify user registration, course management, and fee handling.
- **Enhance Experience**: Improve user experience for students and faculty.
- **Support Functions**: Facilitate efficient course and academic management.
- **Ensure Security**: Protect sensitive information and ensure data accuracy.

The objectives of the university, which include increasing operational effectiveness, raising user satisfaction, and using technology to deliver high-caliber educational services, are all in line with this system.

For more detailed information, please refer to the vision document available online.

D-G09-i228759-i222553.docx

1.3 Title

AcademiaStream: University Management System

The goal of AcademiaStream is to use Scene Builder, JavaFX, and JDBC SQL to create a solid university management system. The goal of this system is to better serve the needs of faculty members, administrators, and students by streamlining and improving the administration of university activities.

1.4 Objectives

1.4.1 Admin Subsystem

Description:

The Admin Subsystem manages crucial administrative tasks within the University Management System. It oversees user management, course administration, and fee structure updates to ensure efficient operation of the academic institution.

Key Features:

- **User Registration:** Register new users into the system with relevant details and types.
- **Course Addition:** Add new courses to the system, specifying details such as course code, name, credits, instructor and prerequisites.
- **Fee Management:** Manage fee structures, including setting fee amounts per credit hour.
- **Viewing Courses:** Access and view details of all available courses within the system.
- **Viewing Users:** View and manage information about registered users, including students, faculty, and administrators.

1.4.2 Student Subsystem

Description:

The Student Subsystem provides essential functionalities tailored for student use within the University Management System. It supports activities related to course registration, academic progress monitoring, viewing attendance, accessing study plan and total fee.

Key Features:

- **Course Registration:** Enables students to register in courses offered by the university for each semester or academic term.
- **Viewing Grades and Marks:** Access and review grades and marks obtained in various courses and assessments.
- **Fee Details:** View detailed information about fee structures and view total fee.
- **Attendance Tracking:** Monitor and track attendance records for enrolled courses to ensure compliance with university attendance policies.
- **Study Plan Access:** View and manage study plans, including scheduled courses and academic requirements.

1.4.3 Faculty Subsystem

Description:

The Faculty Subsystem empowers faculty members with tools to maintain accurate academic records and track student attendance within the University Management System.

Key Features:

- **Update Marks:** Enter and update marks or grades for assessments, assignments, and exams conducted during the academic term.
- **Attendance Tracking:** Monitor and record student attendance for courses taught to ensure compliance with attendance policies.
- **Manage Academic Records:** Maintain comprehensive records of student performance, attendance, and academic progress.

The goal of AcademiaStream is to use Scene Builder, JavaFX, and JDBC SQL to create a solid university management system. The goal of this system is to better serve the needs of faculty, administrators, and students by streamlining and improving the administration of university activities.

1.5 Problem Statement

Problem	Description
The Problem of	fragmented university management in the current universities system is a significant issue.
Affects	the aspect of academic data interoperability, administrator's processes smoothness and the availability of resources in a timely manner. Lack of communication, incompatible information systems and not much interactivity affect the productivity among the educational institutions.

The impact of which	is an educational system characterized by information silos, decreased collaboration among stakeholders, and compromised educational outcomes.
a successful solution would be	deals with the UMS- the development of a single and user-friendly approach that will help to overcome the problems mentioned. The UMS do this via its communication and data systems that streamline communication and integration, and in doing so, it equips educational professionals with all the information and tools needed to arrive at well researched decisions. This technology is a step forward into a future educational system which is not only interconnected, efficient, but also student-centered.

The existing fragmented university administration systems have issues with academic data interoperability, inefficient administrative practices, and delayed access to essential resources. This fragmentation leads to information silos, a decrease in stakeholder participation, and a compromise in academic outcomes. The lack of cohesive information systems and communication platforms exacerbates these problems by impeding effective operations and decision-making.

Feasibility

To solve these issues, the University Management System (UMS) project aims to develop a consolidated, user-friendly platform. The UMS will simplify administrative tasks like student enrollment, course administration, and fee processing by utilizing technologies like Scene Builder, JavaFX, and JDBC SQL. By centralizing data and enhancing system interoperability, the UMS is supposed to improve strategic planning, resource allocation, and transparency inside educational institutions. Ultimately, the UMS aims to furnish teachers, staff, and students with the necessary tools to collaborate and make informed choices, leading to an enhanced and student-focused learning environment.

2. Overall Description

2.1 Product Perspective

This SRS describes the University Management System (UMS), a stand-alone software program that was created totally inside the local system environment. It doesn't rely on external connectors or connections to other systems apart from the user's PC.

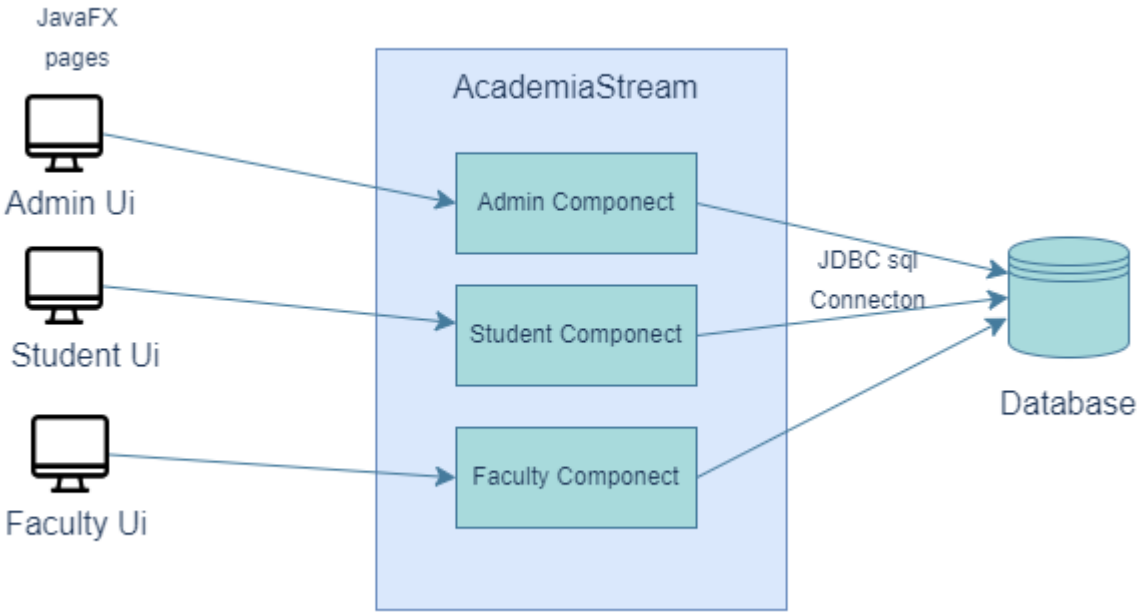
Context and Origin:

- **Origin:** The UMS is developed to address internal operational challenges within educational institutions, aiming to streamline administrative processes without requiring external dependencies other than JDBC connector .

- **Context:** It operates within a closed system environment on a single computer or local network, serving as a centralized tool for managing academic operations.

System Characteristics:

- **Self-contained System:** The UMS is self-contained, meaning all its functionalities, including user management, course administration, fee handling, and academic record keeping, are integrated within the software itself.
- **No External Interfaces:** Since it operates independently, there are no interfaces or connections with external databases, APIs, or online services. All data and operations are managed within the local system.



The UMS offers a centralized platform for managing academic activities locally and is an independent software solution designed to fulfill the unique requirements of educational institutions. It contributes to increased administrative effectiveness and improved educational outcomes by ensuring data confidentiality, operational control, and convenience of use through the elimination of external dependencies.

2.2 Product Functions

Admin Subsystem Functions:

- User management: Register and delete user accounts.
- Course management: Add and delete courses.
- Fee management: Update fee structures.
- View courses and users: Display lists and details of courses and users.

Student Subsystem Functions:

- Course enrollment: Allow students to register for courses.
- View attendance: Display attendance records for enrolled courses.

- Check fee details: View total fee and details.
- Access study plan: Provide access to study schedules and academic plans.
- View academic progress: Display grades, marks, and academic performance.

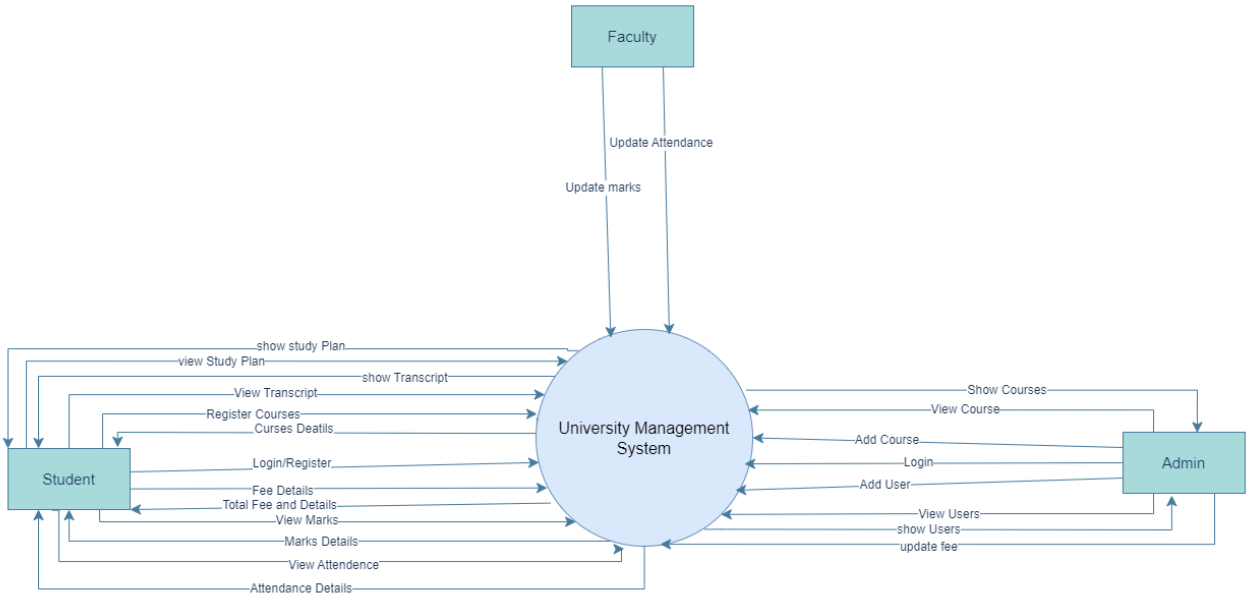
Faculty Subsystem Functions:

- Update academic records: Record and update student marks of quizzes, assignments, sessional, project and final.
- Update attendance: Record the attendance of registered students.

Common Functions:

- Authentication and authorization: Secure login and access control for users based on roles.
- Data management: Ensure reliable storage and retrieval of academic and user data.
- System maintenance: Allow for system configuration, updates, and maintenance tasks.

Top-Level Data Flow Diagram



2.3 List of Use Cases

2.3.1 Manage User Accounts
2.3.2 Manage Courses
2.3.3 Register Courses
2.3.4 Track Attendance
2.3.5 Manage Marks
2.3.6 Access Academic Record
2.3.7 Access Study Plan
2.3.8 View Fee Details
2.3.9 Manage Library
2.3.10 Manage Fees

2.4 Extended Use Cases

2.4.1 Use Case 1

<table>
<tr><th>Element</th><td>Description</td></tr>
<tr><td>Use Case ID</td><td>UC-001</td></tr>
<tr><td>Use Case Name</td><td>Manage User Accounts</td></tr>
<tr><td>Scope</td><td>User Management System</td></tr>
<tr><td>Level</td><td>User Goal</td></tr>
<tr><td>Primary Actor</td><td>Administrator</td></tr>
<tr><td>Stakeholders and Interests</td><td>Administrator: will manage all the managing operations which are adding a new user, removing a user or updating information of any user.
Users: expects the UMS to accurately store their data in a simple and understandable manner.</td></tr>
<tr><td>Preconditions</td><td>Administrators have to log in first into the system.
Administrators have to be allowed to manage (add, delete or update) and edit accounts.</td></tr>
<tr><td>Success Guarantee (Post Conditions)</td><td>The accounts are managed properly whether it is added, updated or removed. The action should be correctly performed. All the actions of management i-e adding, updating and deleting are done when called.</td></tr>
<tr><td>Main Success Scenario (two-column format)</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. Administrator, goes to add new user page</td><td></td></tr>
<tr><td></td><td>2. System Displays a form</td></tr>
<tr><td>3. Admin fills the data and user type in form to add a new user</td><td></td></tr>
<tr><td></td><td>4. System will check the data and make-up a new user account and add in the database</td></tr>
<tr><td></td><td>5. shows success message to user that the action was taken.</td></tr>
<tr><td>5. Administrator, for updating, navigates to view user page</td><td></td></tr>
<tr><td></td><td>6. System will display a list of users</td></tr>
</table>
</td></tr>
</table>

	7. Admin select a user to update and clicks the update button.	
		8. System shows a form to update the data of that specific user.
	9. Admin fills and submits the form	
		9. System updates the data in database.
		10. System Displays a success message.
	10. Admin, to delete user, navigates to view users page	
		11. System displays a list of users.
	12. Admin select a user	
	13. Admin Clicks the remove button.	
		14. System removes that specific user from the database
		15. System displays a success message.
Extensions	6a. Adding user without filling the form: 1. Display an error screen and account will not be added 2. Administrator re-enters the account details and submit. 8a. Data validation error or not according to system when updating account: 1. System finds an invalid data format (e.g., invalid email format) and alerts Administrator by error. 2. Administrator re-enters the information and submit to update. 9a. deleting or updating without selecting a user: 1. System will show the error and will not delete any account. 2. Admin selects the user and then update or remove the user	
Special Requirements	• To monitor any online activity related to the user, the user's accounts must be logged.	

	• Before anything is done, all the user-provided data must satisfy with the validity standards.
Technology and Data Variations List	• To manage user accounts, the system might enable "bulk actions," which would let administrators handle "multiple accounts" at once. • The data may be kept in many formats, each requiring different data management (e.g. SQL).
Frequency of Occurrence	The policies must be constantly reviewed and revised several times a day, given the high number of users and the frequent creation of new accounts.
Open Issues	• When an administrator deletes a user account that is being used for a transaction or other operation, the system should be more concerned with, what to do with those accounts? • Security concerns that must be identified to protect user data while it is being managed, potentially with the use of security measures

2.4.2 Use Case 2

Element	Description
Use Case ID	UC-002
Use Case Name	Manage Courses
Scope	Course Management System
Level	User Goal
Primary Actor	Administrator
Stakeholders and Interests	• ***Administrator need*** a system which can easily handle the course management i-e the adding, deletion and updating of course details and other related functions • ***Students*** *want a system where they can easily access the courses information according to the university management and decide their enrollments.*

<table>
<tr><td>Preconditions</td><td>Administrator must login to the system before any action.
Administrator must be allowed from University Administration to manage courses.</td></tr>
<tr><td>Success Guarantee (Post Conditions)</td><td>Courses must be managed according to the actions of the administrator without any errors. The functions of adding, deletion and updation of courses should perform accordingly on each action</td></tr>
<tr><td>Main Success Scenario (two-column format)</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. Administrator, goes to add new course page</td><td></td></tr>
<tr><td></td><td>2. System Displays a form</td></tr>
<tr><td>3. Admin fills the data and course details in form to add a new course</td><td></td></tr>
<tr><td></td><td>4. System will check the data and make-up a new course and add in the database</td></tr>
<tr><td></td><td>5. shows success message to user that the action was taken.</td></tr>
<tr><td>5. Administrator, for updating, navigates to view courses page</td><td></td></tr>
<tr><td></td><td>6. System will display a list of courses</td></tr>
<tr><td>7. Admin select a course to update and clicks the update button.</td><td></td></tr>
<tr><td></td><td>8. System shows a form to update the data of that specific course.</td></tr>
<tr><td>9. Admin fills and submits the form</td><td></td></tr>
<tr><td></td><td>9. System updates the data in database.</td></tr>
<tr><td></td><td>10. System Displays a success message.</td></tr>
<tr><td>10. Admin, to delete course, navigates to view course page</td><td></td></tr>
<tr><td></td><td>11. System displays a list of courses.</td></tr>
<tr><td>12. Admin select a course</td><td></td></tr>
<tr><td>13. Admin Clicks the remove button.</td><td></td></tr>
</table>
</td></tr>
</table>

	14. System removes that specific course from the database
	15. System displays a success message.
Extensions	6a. Adding Course without filling the form: 3. Display an error screen and course will not be added 4. Administrator re-enters the course details and submit. 8a. Data validation error or not according to system when updating account: 3. System finds an invalid data format and alerts Administrator by error. 4. Administrator re-enters the information and submit to update. 9a. deleting or updating without selecting a course : 3. System will show the error and will not delete any course. 4. Admin selects the user and then update or remove the course
Special Requirements	• Course changes must be according to the policy of HEC. • Course data must comply with validation rules of university Administration and faculty before any action in system
Technology and Data Variations List	• The system may support printing the details of multiple courses together for academic purposes. • Course data might be stored in various formats, requiring specific handling for each type (e.g., SQL, and physical database).
Frequency of Occurrence	Regularly or at the beginning of session, depending on the academic calendar and as needed for course updates or introductions.
Open Issues	• If a course is to be deleted, how will the system manage those with current enrollments? • a description of security procedures to safeguard course data while managing operations.

2.4.3 Use Case 3

Element	Description
Use Case ID	UC-003
Use Case Name	Register Course

<table>
<tr><td>Scope</td><td>Education Management System</td></tr>
<tr><td>Level</td><td>User Goal</td></tr>
<tr><td>Primary Actor</td><td>Student</td></tr>
<tr><td>Stakeholders and Interests</td><td>• Students need a system so that they can easily register in the selected courses, or the courses offered.
• Administrator wants a system where they can easily manage student enrollments, course registrations, and major selections.
• Faculty memberswant to make sure that choosing a major and registering in courses go without incident, with proper resource allocation and class numbers.</td></tr>
<tr><td>Preconditions</td><td>• The student has must taken admission in the university.
• The student has an account and access to UMS.
• Student is registering when Registration for the term is open.</td></tr>
<tr><td>Success Guarantee (Post Conditions)</td><td>The student has chosen a major, successfully registered for courses, and is enrolled in classes for the forthcoming term.</td></tr>
<tr><td>Main Success Scenario (two-column format)</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. Student navigate to register page to register a course</td><td></td></tr>
<tr><td></td><td>2. System will display a list of courses</td></tr>
<tr><td>3.Student selects a course from list.</td><td></td></tr>
<tr><td>4. Student clicks register button.</td><td></td></tr>
<tr><td></td><td>5. System registers the student in the selected course.</td></tr>
<tr><td></td><td>6. System Display the total credit hours of student.</td></tr>
<tr><td></td><td>7. System displays a success message of course registration.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>1: When the Prerequisites are not met

6a. Student registering a course does not meet course prerequisites for that specific course:</td></tr>
</table>

	6a1. System displays the error that prerequisites are not met. 6a2. The student will enroll in any other course or will complete the requirement during a later term. 2: Register button is clicked without selecting a course System displays an error message and student reselect the course and register 3: Credit Hours is full Student cannot register in any course as credit hours is exceeding and shows an error message
Special Requirements	• Updates in real time on course status should be provided by the system. • Information about registration and enrollment data must be managed confidentially by the system. • Accessibility features should be suitable for all students, including disables.
Technology and Data Variations List	• The system may allow printing services, like printing the details of courses for academic purpose. • Considering different devices have access to data, responsive design is necessary for PCs, tablets, and smartphones.
Frequency of Occurrence	Mostly two times a year, after each session courses are offered for registration
Open Issues	• How will the system behave when it will integrate to the financial assistance and scholarships program outside the system. • It is necessary to specify the procedure for managing and handling over enrollment in popular courses.

2.4.4 Track Attendance

Element	Description

<table>
<tr><td>Use Case ID</td><td>UC-004</td></tr>
<tr><td>Use Case Name</td><td>Track Attendance</td></tr>
<tr><td>Scope</td><td>Education Management System</td></tr>
<tr><td>Level</td><td>User Goal</td></tr>
<tr><td>Primary Actor</td><td>Faculty member/Administrator</td></tr>
<tr><td>Stakeholders and Interests</td><td>• Faculty member/administrator: Must track absences generate attendance reports, and mark attendance effectively.
• Pupils: Insist that their attendance be accurately tracked and reported.
• Administration: Interested in keeping an eye on attendance trends and efficiently controlling absence.</td></tr>
<tr><td>Preconditions</td><td>• After completing the authentication process, the faculty member or administrator can access the attendance monitoring area.
• The planned class or session is used to monitor attendance.</td></tr>
<tr><td>Success Guarantee (Post Conditions)</td><td>Absenteeism is successfully monitored as well as controlled, and student attendance is correctly recorded, and reports are made available.</td></tr>
<tr><td>Main Success Scenario (two-column format)</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. Faculty selects course and navigates to attendance page to mark attendance</td><td></td></tr>
<tr><td></td><td>2. The system displays the attendance details and form with attendance details like date, attendance etc.</td></tr>
<tr><td>3. Faculty selects a student in a specific course</td><td></td></tr>
<tr><td>4. Faculty fills the attendance details</td><td></td></tr>
<tr><td></td><td>4. The system updates the attendance in database.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>1: attendance already exists in database
System shows an error to correct the attendance</td></tr>
</table>

	2: Sql exception occurs System handles the error and shows the relative message 3: Faculty sends an empty attendance System shows the error and faculty enters the attendance 4: No student is selected System shows error box and faculty selects the student and re-submits
Special Requirements	• The accuracy confidentiality of attendance data must be guaranteed by the system. • verification of attendance records to prevent mistakes. • the capacity to be used with ease by everyone, even those with disabilities.
Technology and Data Variations List	• The system may recommend a device that uses student ID swipes or scanners to automatically calculate a student's attendance. • Must be compatible with many platforms and devices, offering instructors, administrators, and students a responsive design.
Frequency of Occurrence	Every day, or as each course/session's academic calendar specifies.
Open Issues	• Monitoring of student performance, including academic achievement, is made possible by integrating the attendance tracking system with extra tools and online resources.

2.3.5 Manage Marks

Element	Description
Use Case ID	UC-005
Use Case Name	Manage Marks
Scope	Education Management System
Level	User Goal
Primary Actor	Faculty member/Administrator
Stakeholders and Interests	• **Faculty member**: Must manage marks efficiently and accurately. • **Students**: Expect accurate recording and reporting of their marks. • **Administration**: Requires oversight of student performance and accurate records for reporting and decision-making.

<table>
<tr><td>Preconditions</td><td>After completing the authentication process, the faculty member can access the marks monitoring area.
The planned assessment or project is used to update marks.</td></tr>
<tr><td>Success Guarantee (Post Conditions)</td><td>Marks are successfully recorded and stored in the database, and student performance reports are made available.</td></tr>
<tr><td>Main Success Scenario (two-column format)</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. Faculty selects course and navigates to marks page to upload marks.</td><td></td></tr>
<tr><td></td><td>2. System Displays a list of students with their courses and input form for marks.</td></tr>
<tr><td>3. Faculty selects the student and their respective course</td><td></td></tr>
<tr><td>4. Faculty select the assessment type.</td><td></td></tr>
<tr><td>5. Faculty enter the assessment id, total and obtained marks</td><td></td></tr>
<tr><td>6. Faculty click the save marks button.</td><td></td></tr>
<tr><td></td><td>7. System saves the marks for respective assessment for selected student and course in the database.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>1a. Assessment data already exists in the database.
1a1. The system shows an error to correct the marks.

2a. SQL exception occurs.
2a1. The system handles the error and shows a relevant message.

3a. Faculty sends an empty marks form.
3a1. The system shows an error and prompts the faculty to enter the marks.

4a. No student is selected.
4a1. The system shows an error box and prompts the faculty to select a student and re-submit.</td></tr>
</table>

	5a. Obtained marks are greater than Total marks The system shows an error and prompts the faculty to enter the marks 6a: marks are not numerical terms 6a1: The system shows an error and prompts the faculty to re-enter the marks
Special Requirements	• The accuracy and confidentiality of marks data must be guaranteed by the system. • Verification of marks records to prevent mistakes. • The capacity to be used with ease by everyone, even those with disabilities.
Technology and Data Variations List	• The system may recommend devices or software that integrate with the marks management system for automated data entry. • Must be compatible with many platforms and devices, offering instructors, administrators, and students a responsive design.
Frequency of Occurrence	Every academic term or as specified by the academic calendar for each course/session and whenever a quiz or assesment is conducted.
Open Issues	• Monitoring of student performance, including academic achievement, is made possible by integrating the marks management system with additional tools and online resources.

2.3.6 Access Academic Record

Element	Description
Use Case ID	UC-006
Use Case Name	Access Academic Record
Scope	Education Management System
Level	User Goal
Primary Actor	Student
Stakeholders and Interests	• **Student:** Needs access to their academic records to track progress, attendance and plan future courses.
Preconditions	• After completing the authentication process, the student can access the marks and transcript.
Success Guarantee	Academic records are accurately retrieved and displayed to the user. The user can review the marks of each assessment and also the transcripts.

<table>
<tr><td>(Post Conditions)</td><td></td></tr>
<tr><td>Main Success Scenario (two-column format)</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. Student navigates to marks page to view marks of assessment.</td><td></td></tr>
<tr><td></td><td>2. System Displays a the marks page with options of registered courses and assessment types.</td></tr>
<tr><td>3. Student selects the course</td><td></td></tr>
<tr><td>4. Student select the assessment type or grand total.</td><td></td></tr>
<tr><td></td><td>5.System fetch the data from database and displays the marks.</td></tr>
<tr><td>6. Student navigates to transcript page to transcript.</td><td></td></tr>
<tr><td></td><td>7.System displays the transcript pagewith options of semesters</td></tr>
<tr><td>8. Student selects the semester.</td><td></td></tr>
<tr><td></td><td>9. System displays the transcript.</td></tr>
<tr><td>10. Student navigate to the attendance page</td><td></td></tr>
<tr><td></td><td>11. System Displays attendance page with option of registered courses</td></tr>
<tr><td>11. Student Selects a course</td><td></td></tr>
<tr><td></td><td>12. System displays the attendance in that course</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>2a. SQL exception occurs.
2a1. The system handles the error and shows a relevant message.</td></tr>
<tr><td>Special Requirements</td><td>• The accuracy and confidentiality of academic records must be guaranteed by the system.
• Verification of academic records to prevent mistakes.
• The capacity to be used with ease by everyone, even those with disabilities.</td></tr>
<tr><td>Technology and Data Variations List</td><td>The system may recommend devices or software that integrate with the academic records management system for automated data entry.Must be compatible with many platforms and devices, offering instructors, administrators, and students a responsive design</td></tr>
</table>

Frequency of Occurrence	As needed by students for academic review, planning, and reporting.
Open Issues	Integration of academic record systems with additional tools and online resources to monitor student performance and provide comprehensive academic support.

2.3.7 Access Study Plan

Element	Description
Use Case ID	UC-007
Use Case Name	Access Academic Record
Scope	Education Management System
Level	User Goal
Primary Actor	Student
Stakeholders and Interests	• **Student:** Needs access to study plan to plan future courses.
Preconditions	• After completing the authentication process, the student can access the study plan.
Success Guarantee (Post Conditions)	Study plan information is successfully retrieved and displayed to the user. The user can review .
Main Success Scenario (two-column format)	

Actor Action	System Responsibility
1. Student navigates to study plan page to view study plan.	
	2. System Displays the study plan page with options of semester.
3. Student selects the semester	
	4.System fetch the courses of that semester from database and displays the courses.

Extensions	2a. SQL exception occurs. 2a1. The system handles the error and shows a relevant message.
Special Requirements	• The accuracy and completeness of study plan data must be ensured by the system.

	• Verification of study plans to prevent errors in course sequencing and requirements. • The system should be accessible and usable by all users, including those with disabilities.
Technology and Data Variations List	• The system may recommend features like course registration integration and academic advising tools to enhance study plan management. • Must support compatibility across various platforms and devices, offering a responsive design for ease of access.
Frequency of Occurrence	Users access study plans typically at the start of each semester or academic year to plan courses and track progress.
Open Issues	Integration of study plan management with academic advising systems to provide personalized guidance and support for students. Exploring options for dynamic study plans that can adapt based on student progress and changing academic requirements.

2.3.8 View Fee Details

Element	Description
Use Case ID	UC-008
Use Case Name	View Fee Details
Scope	Education Management System
Level	User Goal
Primary Actor	Student
Stakeholders and Interests	• **Student:** Needs to view fee details to understand payment obligations and deadlines.
Preconditions	• After completing the authentication process, the student can access the fee details.
Success Guarantee (Post Conditions)	fee details and total fee is successfully retrieved and displayed to the user. The user can review .
Main Success Scenario (two-column format)	

Actor Action	System Responsibility
1. Student navigates to fee page to view fee details.	
	2. System Displays the fee page with details of registered courses, credit hours and total fee.

Extensions	2a. SQL exception occurs. 2a1. The system handles the error and shows a relevant message.
Special Requirements	• The accuracy and confidentiality of fee details must be ensured by the system. • The system should be accessible and usable by all users, including those with disabilities
Technology and Data Variations List	The system may recommend features like online payment integration and fee calculation tools to enhance fee management. Must support compatibility across various platforms and devices, offering a responsive design for ease of access.
Frequency of Occurrence	Users access fee details regularly, especially at the beginning of each semester or academic term, to plan payments and understand financial obligations.
Open Issues	Integration of fee details with student accounts for seamless payment processing and real-time updates. Exploring options for automated fee reminders and notifications to improve fee collection efficiency and reduce late payments.

2.3.9 Manage Library

Element	Description
Use Case ID	UC-009
Use Case Name	Manage library
Scope	Library Management System
Level	Admin Goal
Primary Actor	Administrator, Students, Faculty
Stakeholders and Interests	• **Student:** Needs to access library books • **Admin:** Needs to manage library • **Faculty:** Needs to access library resources
Preconditions	• administrator is authenticated and has access to the library system. Library resources are cataloged.
Success Guarantee	The selected resource is booked. Admin added , updated , deleted the resource successfully.

<table>
<tr><td>(Post Conditions)</td><td></td></tr>
<tr><td>Main Success Scenario (two-column format)</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. Student/Faculty navigates to library to book resources.</td><td></td></tr>
<tr><td></td><td>2. System Displays the library page with details of resources.</td></tr>
<tr><td>3. Student/Faculty selects a resource and book it.</td><td></td></tr>
<tr><td></td><td>4. System lend the resource and update in the databse</td></tr>
<tr><td>5. Admin visits library page</td><td></td></tr>
<tr><td></td><td>6. System Displays the library page with details of resources.</td></tr>
<tr><td>7. Admin adds, updates, deletes a resource.</td><td></td></tr>
<tr><td></td><td>8. System updates the database</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>2a. SQL exception occurs.
2a1. The system handles the error and shows a relevant message.</td></tr>
<tr><td>Special Requirements</td><td>Accurate book records, integration with library automation systems, accessibility features.</td></tr>
<tr><td>Technology and Data Variations List</td><td>RFID/barcode integration, compatibility with library management software, mobile-friendly interface</td></tr>
<tr><td>Frequency of Occurrence</td><td>Daily updates to catalog, regular user account management, monitoring of library usage.</td></tr>
<tr><td>Open Issues</td><td>Digital lending options, analytics integration for resource optimization, LMS integration for seamless access.</td></tr>
</table>

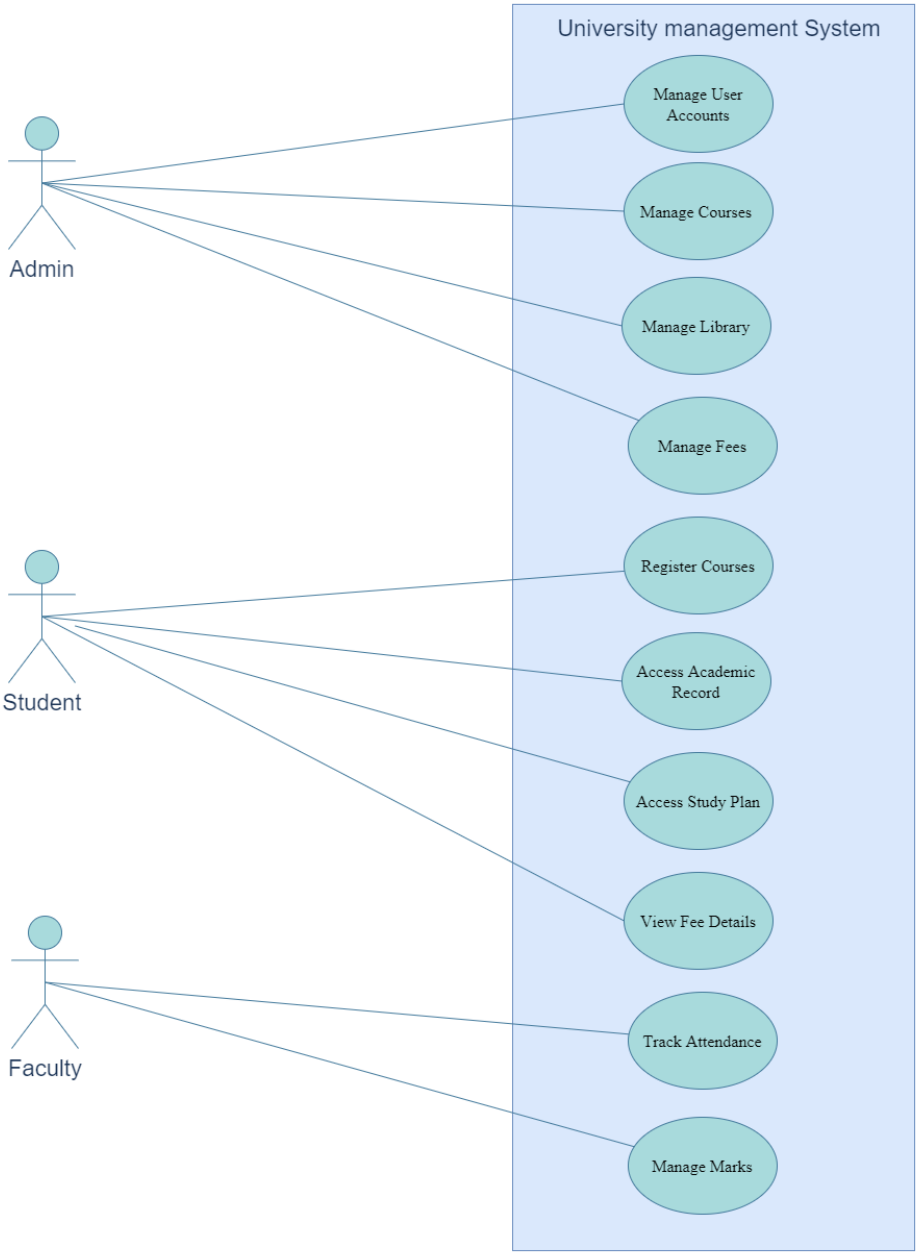
2.3.10 Manage Fees

Element	Description
Use Case ID	UC-009
Use Case Name	Manage library
Scope	Finance Management System
Level	Admin Goal
Primary Actor	Administrator

<table>
<tr><td>Stakeholders and Interests</td><td>• Admin: Needs to manage the fees and finance</td></tr>
<tr><td>Preconditions</td><td>• administrator is authenticated and has access to the finance system. .</td></tr>
<tr><td>Success Guarantee (Post Conditions)</td><td>Fee is collected successfully displayed to the admin.</td></tr>
<tr><td>Main Success Scenario (two-column format)</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. Administrator navigates to fee management.</td><td></td></tr>
<tr><td>2. Administrator selects a student’s fee record.</td><td></td></tr>
<tr><td></td><td>3. System retrieves and displays the student’s fee details.</td></tr>
<tr><td></td><td>4. System lend the resource and update in the databse</td></tr>
<tr><td>5. Administrator records fee payment details.</td><td></td></tr>
<tr><td></td><td>6. System prompts for payment method and updates fee status</td></tr>
<tr><td>7. Administrator confirms fee acceptance.</td><td></td></tr>
<tr><td></td><td>8. System stores payment confirmation and updates student records.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>• Payment method validation fails. System prompts for correct payment details.
• Student fee record not found. System alerts administrator to check student details.</td></tr>
<tr><td>Special Requirements</td><td>• Secure payment processing to maintain confidentiality.
• Audit trails for financial transactions.
• Accessibility features for ease of use.</td></tr>
<tr><td>Technology and Data Variations List</td><td>• Integration with payment gateways for real-time updates.
• Compatibility with various banking systems and currencies</td></tr>
</table>

Frequency of Occurrence	• Daily transactions during fee payment periods. • Ad-hoc payments throughout the academic year.
Open Issues	• Integration with student information systems for seamless fee tracking. • Compliance with financial regulations and reporting standards.

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

3.1.2 Response Time:

- **Requirement:** Under typical load levels, the system should react to user input in two seconds.
- **Rationale:** Quick response times improve productivity and user happiness, particularly during periods of high usage when several users may be logged in at once.

3.1.2 Database Query Performance:

- **Requirement:** 95% of requests for student and course information should be fulfilled by database queries in less than a second.
- **Rationale:** Optimizing database performance guarantees prompt access to vital information, facilitating decision-making procedures and enhancing overall operational effectiveness.

3.1.3 Concurrent User Handling:

- **Requirement:** Up to 500 concurrent users should be supported by the system without experiencing noticeably worse performance.
- **Rationale:** Peak usage of education management systems occurs during exam periods, registration periods, and fee payment deadlines. Managing multiple users at once guarantees dependability and continuous service.

3.1.4 Batch Processing Efficiency:

- **Requirement:** For big datasets (e.g., 10,000 records), batch processing operations (e.g., creating attendance reports, computing student grades) should be finished in 30 minutes or less.
- **Rationale:** Batch processing that is both efficient and supportive of administrative chores allows for timely reporting and analysis without interfering with operational workflows.

3.1.5 System Availability:

- **Requirement:** The system should maintain at least 99.9% uptime, excluding scheduled maintenance windows.
- **Rationale:** When not including regular maintenance windows, the system should maintain at least 99.9% uptime.

3.1.6 Scalability:

- **Requirement:** It should be possible for the system to grow both the user base and the volume of data by 20% annually.
- **Rationale:** Scalability guarantees long-term viability and cost-effectiveness by facilitating future growth and user uptake without necessitating major infrastructure modifications or redesign.

3.1.7 Security Performance:

- **Requirement:** Within 500 milliseconds, user credentials should be validated by authentication and authorization processes.
- **Rationale:** By allowing authorized users to access the system quickly and blocking unauthorized attempts to do so, swift authentication improves system security.

3.2 Safety Requirements

3.2.1 Data Security and Privacy:

- **Requirement:** In order to handle student and faculty personal data, the system must adhere to GDPR (General Data Protection Regulation) rules.
- **Action:** Encrypt sensitive data while it's in transit and at rest. Achieve user consent before processing their data, and follow retention guidelines.

3.2.2 Backup and Disaster Recovery:

- **Requirement:** Every day, regular backups of the entire system's data must be made and safely stored offshore.
- **Action:** To guarantee data availability and integrity in the event of hardware malfunctions or natural disasters, put automatic backup processes into place and periodically practice disaster recovery exercises.

3.2.3 User Access Control:

- **Requirement:** Access to sensitive administrative functions (such as financial transactions and student records) must be restricted using the principles of role-based access control (RBAC).
- **Action:** To monitor and trace access to crucial system components, use robust authentication mechanisms (such as multi-factor authentication) and audit logs.

3.2.4 Physical Security:

- **Requirement:** The system's hosting servers have to be kept in safe, industry-compliant buildings with restricted access.
- **Action:** Take action by limiting physical access to authorised staff only. Use fire suppression systems and environmental controls (such as humidity and temperature) to protect hardware.

3.2.5 Emergency Procedures:

- **Requirement:** It is necessary to establish explicit protocols and rules for handling security issues, such as data breaches, and system malfunctions.

- **Action:** To reduce effect and quickly return to normal operations, maintain an incident response plan that outlines notification procedures, containment strategies, and recovery actions.

3.2.6 Compliance with Regulatory Standards:

- **Requirement:** Local and international laws pertaining to cybersecurity and educational data management must be followed by the system.
- **Action:** Obtain the required certifications for data handling and security procedures and conduct routine audits and assessments to verify compliance with pertinent standards (such as FERPA, HIPAA, etc.).

3.2.7 User Training and Awareness:

- **Requirement:** Users (Faculty, students, administrators) must be trained on data security best practices and their responsibilities for protecting sensitive data.
- **Action:** To improve overall security posture, provide training modules covering subjects like phishing awareness, password management, and reporting security events.

By reducing the possibility of data loss, illegal access, and operational interruptions, these safety regulations hope to provide a dependable and safe environment for all parties engaging with the education management system.

3.3 Security Requirements

3.3.1 User Authentication:

- **Requirement:** In order to access the system, users must authenticate.
- **Action:** Where appropriate, put in place safe authentication methods such multi-factor authentication (MFA), username/password, or biometric authentication.

3.3.2 Data Encryption:

- **Requirement:** Encryption is required for all sensitive data, including financial records and personal information, both in transit and at rest.
- **Action:** To prevent unwanted access, use robust encryption standards (like AES-256) for data transit and storage over networks.

3.3.3 Access Control:

- **Requirement:** Role-based access control (RBAC) must be implemented in order to limit user roles and responsibilities-based access to system resources.

- **Action:** Establish regulations for access that restrict privileged actions (such changing data or configuring a system) to individuals who are permitted.

3.3.4 Data Integrity:

- **Requirement:** Maintaining data integrity throughout its lifecycle is necessary to stop illegal alteration or tampering.
- **Action:** Put in place safeguards like audit trails, digital signatures, and checksums to identify and counteract efforts at data manipulation.

3.3.5 Security Monitoring and Logging:

- **Requirement:** Enable thorough logging and monitoring of all system activity, including administrative and user actions.
- **Action:** Put in place logging systems that record pertinent security events and abnormalities in order to quickly identify and address possible security incidents.

3.3.6 Incident Response and Management:

- **Requirement:** Maintaining an incident response strategy is necessary to quickly handle security events, data breaches, and other security-related issues.
- **Action:** Establish protocols for the identification, eradication, containment, and recovery of incidents. Assign members of the incident response team positions and duties.

3.3.7 Compliance with Privacy Regulations:

- **Requirement:** Compliance with privacy laws and guidelines that apply to student data (such as FERPA and GDPR) is mandatory.
- **Action:** Take action to safeguard personal data by putting in place privacy measures like data minimization, consent management, and privacy impact assessments.

3.3.8 Physical Security:

- **Requirement:** Ensure physical security measures are in place to protect server rooms and data centers hosting the system.
- **Action:** Implement access controls, surveillance systems, and environmental controls (e.g., temperature, humidity) to safeguard physical infrastructure.

3.3.9 Security Awareness and Training:

- **Requirement:** To encourage best practices in data protection and cybersecurity, regularly teach users on security awareness.
- **Action:** To lower security risks, offer training on subjects including phishing prevention, password hygiene, and identifying social engineering techniques.

3.3.10 Security Testing and Vulnerability Management:

- **Requirement:** Perform regular security assessments, vulnerability scans, and penetration testing of the system.
- **Action:** Remediate identified vulnerabilities promptly and update security controls based on emerging threats and best practices.

These security specifications are meant to maintain user confidence in data security and privacy while guaranteeing regulatory compliance and safeguarding the availability, confidentiality, and integrity of data within the education management system.

3.4 Software Quality Attributes

3.4.1 Reliability:

- **Description:** The system should perform its functions consistently and accurately under normal operating conditions.
- **Requirement:** Achieve a system uptime of at least 99.9% availability per month, ensuring minimal downtime for users.
- **Verification:** Monitor system uptime using automated tools and report downtime incidents for analysis and improvement.

3.4.2 Usability:

- **Description:** The system should be intuitive and easy to use, minimizing the learning curve for new users.
- **Requirement:** Achieve a usability score of at least 80% on standardized usability testing metrics (e.g., SUS - System Usability Scale).
- **Verification:** Conduct usability testing sessions with representative users to gather feedback and improve interface design.

3.4.3 Maintainability:

- **Description:** The system should be easy to maintain and modify to accommodate future enhancements or bug fixes.
- **Requirement:** Ensure that any code modification or bug fix does not exceed an average of 2 hours per issue.
- **Verification:** Track development and maintenance time for changes using version control and issue tracking systems.

3.4.4 Security:

- **Description:** The system should protect data and resources from unauthorized access, breaches, and vulnerabilities.

- **Requirement:** Achieve compliance with industry standards (e.g., OWASP Top 10) and regulatory requirements (e.g., GDPR, FERPA).
- **Verification:** Conduct regular security audits, penetration tests, and vulnerability assessments to identify and mitigate risks.

3.4.5 Performance Efficiency:

- **Description:** The system should handle a high volume of concurrent users and transactions efficiently without degradation.
- **Requirement:** Support a minimum of 1000 concurrent users with an average response time of under 1 second for critical operations.
- **Verification:** Perform load testing and stress testing scenarios to evaluate system performance under peak loads.

3.4.6 Scalability:

- **Description:** The system should scale horizontally or vertically to accommodate increasing data and user demands over time.
- **Requirement:** Demonstrate scalability by supporting a 20% annual increase in users and data volume without significant performance degradation.
- **Verification:** Conduct scalability testing to measure system performance and capacity under increased load conditions.

3.4.7 Interoperability:

- **Description:** The system should integrate seamlessly with external systems, services, and APIs used within the educational ecosystem.
- **Requirement:** Ensure compatibility with standard protocols and data formats (e.g., RESTful APIs, JSON) for interoperable data exchange.
- **Verification:** Validate interoperability through integration testing with external systems and services.

3.4.8 Testability:

- **Description:** The system should facilitate effective testing strategies to validate functionality, performance, and security.
- **Requirement:** Achieve at least 80% code coverage in unit tests and automate regression testing for critical features.
- **Verification:** Monitor test coverage metrics and automate test execution using continuous integration (CI) pipelines.

These software quality attributes aim to ensure that the education management system meets high standards of reliability, usability, security, performance, maintainability, scalability, interoperability, and testability. Each attribute is designed to enhance the overall quality of the system, meeting both customer expectations and developer needs for efficiency and effectiveness.

3.5 Business Rules

3.5.1 User Roles and Permissions:

- **Description:** The system is only accessible to logged-in users with valid credentials.
- **Rule:** All modules' data, user accounts, and system settings are fully manageable by administrators.
- **Rule:** Faculty have the authority to oversee all course-related tasks, including keeping track of grades, verifying attendance, and gaining access to student information.
- **Rule:** Students are able to check their grades, fees, academic records, and library resources, but they are not able to edit system data.

3.5.2 Data Integrity and Validation:

- **Description:** Assure consistency and correctness of data across the system.
- **Rule:** Before being entered into the database, any user input—including attendance logs, grades, and fee payments—must pass validation checks.
- **Rule:** Keep records of attendance, grades, and fee payments free of duplicate entries for the same student.

3.5.3 Academic Policies:

- **Description:** Enforce systemic policies and processes related to academia.
- **Rule:** Faculty members are required to mark attendance and turn in marks in accordance with the academic schedule.
- **Rule:** Changes to student enrollments or academic records can only be approved by authorized staff members, such as administrators or academic advisers.

3.5.4 Fee Payment and Financial Rules:

- **Description:** Safely handle financial transactions and fee payments.
- **Rule:** Students are required to follow the payment schedules and cost plans that are laid out by the school.
- **Rule:** Using the system, administrators can create fee invoices, monitor payments, and provide receipts.

3.5.5 Security and Confidentiality:

- **Description:** Preserve private information about users and safeguard important data.
- **Rule:** Only authorized personnel have access to student records, financial information, and personal data.
- **Rule:** To prevent unwanted access to sensitive data, use secure authentication procedures and encryption protocols.

3.5.6 Audit and Compliance:

- **Description:** Verify adherence to internal policies and regulatory requirements.
- **Rule:** Keep audit records for every important system operation, such as configuration changes, user logins, and data alterations.
- **Rule:** To reduce risks and vulnerabilities, review and update security protocols on a regular basis.

These business rules control how the education management system functions and behaves. They also guarantee security, operational integrity, and adherence to institutional norms and procedures by dictating how users engage with the system.

3.6 Operating Environment

The software will operate in a typical educational institution environment, which includes:

3.6.1 Hardware Platform:

- Standard desktop and laptop computers
- Mobile devices (tablets, smartphones) for user accessibility

3.6.2 Operating Systems:

- Windows 7 and above
- macOS 10.15 and above
- Linux distributions (Ubuntu, CentOS, etc.)

3.6.3 Database:

- MySQL 8.0 and above
- SQLite 3.0 and above
- MySQL Workbench 8.0 and above

3.6.4 Frameworks and Libraries:

- Java Runtime Environment (JRE) 11 and above
- JavaFX 11 and above for GUI development
- My-sql-connector 8.2 and above

3.6.5 Network Requirements:

- Internet connectivity for accessing online resources and updates
- Local area network (LAN) for internal communication within the institution

3.6.6 Integration with Other Systems:

- Integration with the student information systems (SIS) already in place at the institution
- Suitability for smooth data interchange with learning management systems (LMS)
- compatibility for user management with authentication systems (LDAP, Active Directory)

3.6.7 Security Protocols:

- Secure Socket Layer (SSL) encryption for data transmission over the network
- User authentication mechanisms to ensure data security and privacy compliance

3.6.8 User Accessibility:

- Support for accessibility standards (WCAG 2.0) to accommodate users with disabilities
- Responsive design to optimize user experience across different screen sizes and devices

3.6.9 Regulatory Compliance:

- Adherence to data protection regulations (GDPR, CCPA) for handling personal information
- Compliance with educational standards and policies set by regulatory bodies

The operating system of the software is made to be adaptable and work with a variety of hardware and software configurations that are frequently used in educational settings. In order to facilitate the effective administration of academic and administrative procedures, security, accessibility, and integration capabilities are given top priority.

3.7 User Interfaces

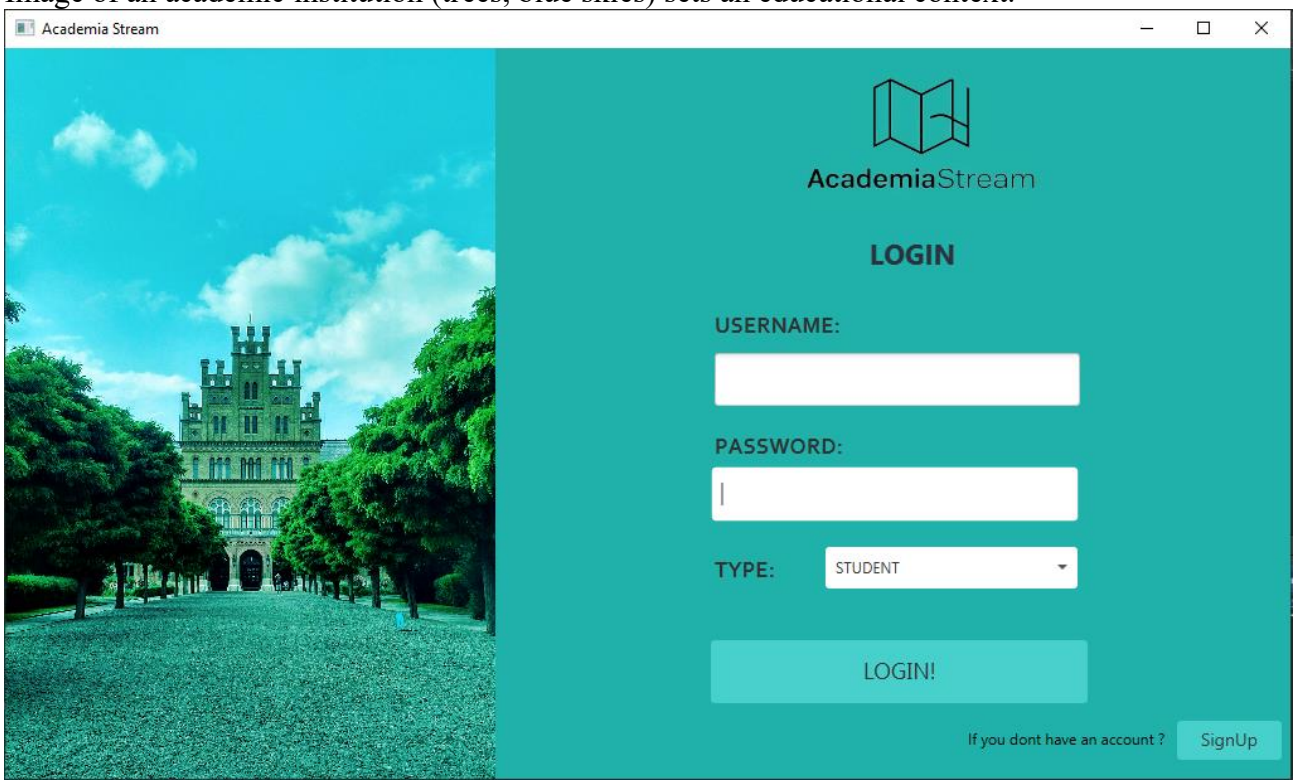
3.7.1 Login And Register

Login Interface:
Components:
Username Field: Allows users to input their username.
Password Field: Allows users to input their password.
Dropdown Menu (TYPE): Offers options (e.g., “STUDENT”) for users to select their role.
LOGIN Button: Initiates the login process.
Sign Up Link: Provides an option for users who don’t have an account.
Layout Constraints:

Central alignment of elements for focus on user interaction points.

Visual Context:

Image of an academic institution (trees, blue skies) sets an educational context.

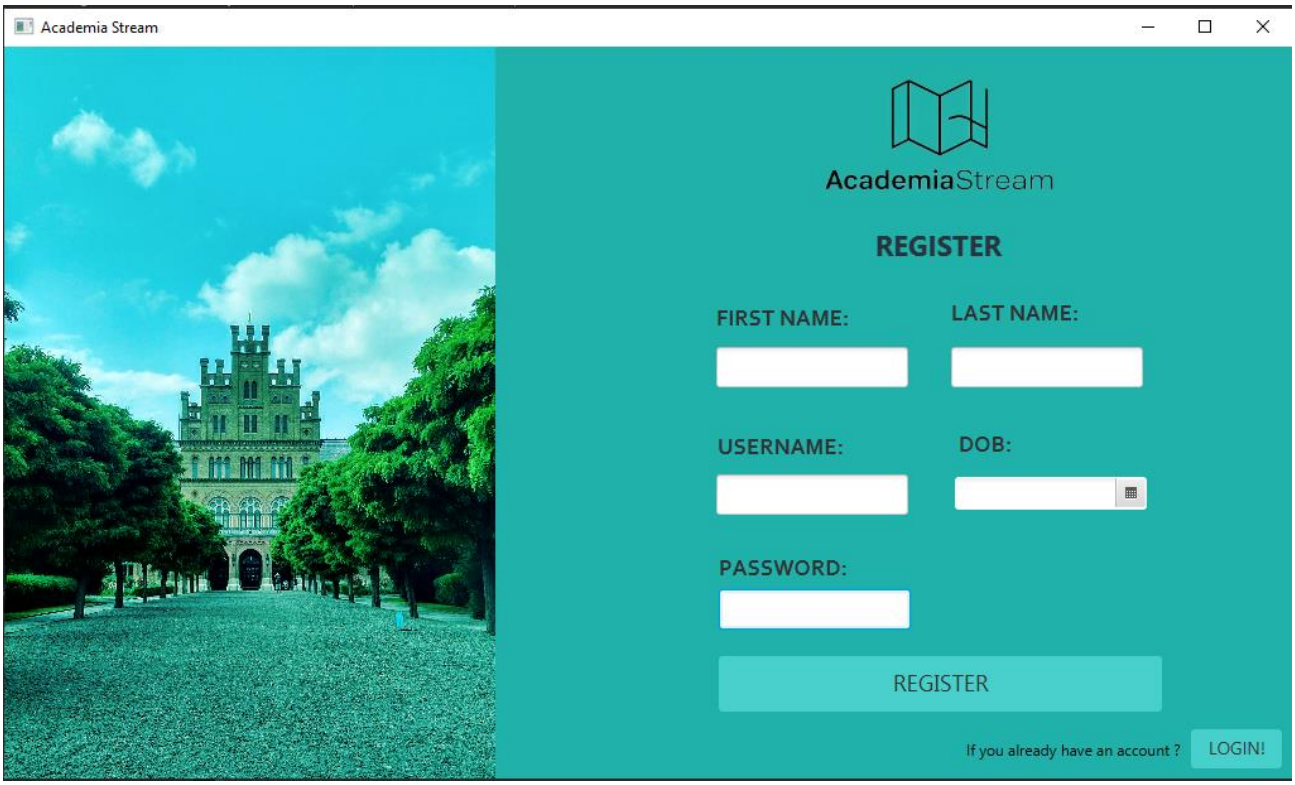


User Registration Interface:

- **Components**: The interface includes input fields for “First Name,” “Last Name,” “Username,” “Date of Birth (DOB),” and “Password.”
- **Layout Constraints**: The layout is organized with rectangular input boxes for each field, aligned vertically.
- **Buttons**: Two buttons are provided: “REGISTER” (highlighted in green) and a secondary option to “LOGIN” for existing users.
- **Visual Elements**: The background features an image of a grand building, suggesting an educational context.

Standard Elements:

- **Input Fields**: Users can enter their personal information (first name, last name, etc.).
- **Buttons**: “REGISTER” and “LOGIN” buttons.
- **Error Messages**: Standards for displaying errors (e.g., invalid input, missing fields).



3.7.2 Dashboards

Student Dashboard
Software Name and Window Controls: At the top, the software's name "AcademiaStream" is displayed on the left. On the right, there are standard window control buttons (minimize, maximize, close).

Navigation Panel: A vertical navigation panel is present on the left side with icons representing different functionalities. These include menu options, "Dashboard","Home," "Registration," "Fee Details," "Attendance," "Marks," "Transcript," and "Study Plan.

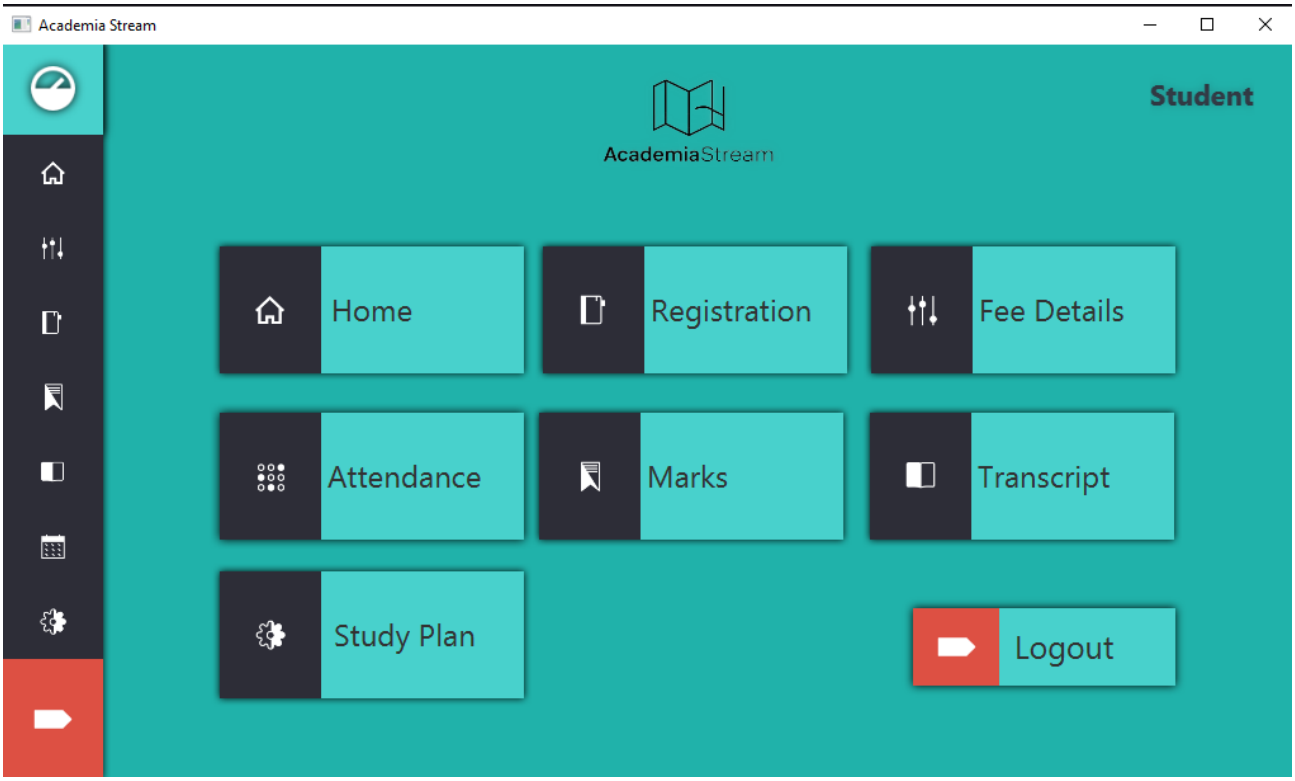
Main Interface: The main part of the interface consists of rectangular tiles arranged in two rows. Each tile represents different sections or functions available to the user: "Home," "Registration," "Fee Details," "Attendance," "Marks," "Transcript," and "Study Plan." These tiles have icons corresponding to their functions.

Logout Button: At the bottom right corner, there's a 'Logout' button. Its red color stands out from other elements, emphasizing its importance as an action button.

GUI Standards: The interface adheres to certain GUI standards such as consistent icon usage for intuitive navigation and clear demarcation of actionable items like logout buttons. The dark theme and color scheme provide visual segmentation and ease of use.

Software Components: The software components for which a user interface is needed include the Home, Registration, Fee Details, Attendance, Marks, Transcript, and Study Plan sections. Each of

these sections has its own set of functionalities and interactions that would be detailed in a separate user interface specification.



Admin Dashboard
Software Name and Window Controls: At the top, the website's name "AcademiaStream" is displayed on the left. On the right, there are standard window control buttons (minimize, maximize, close).

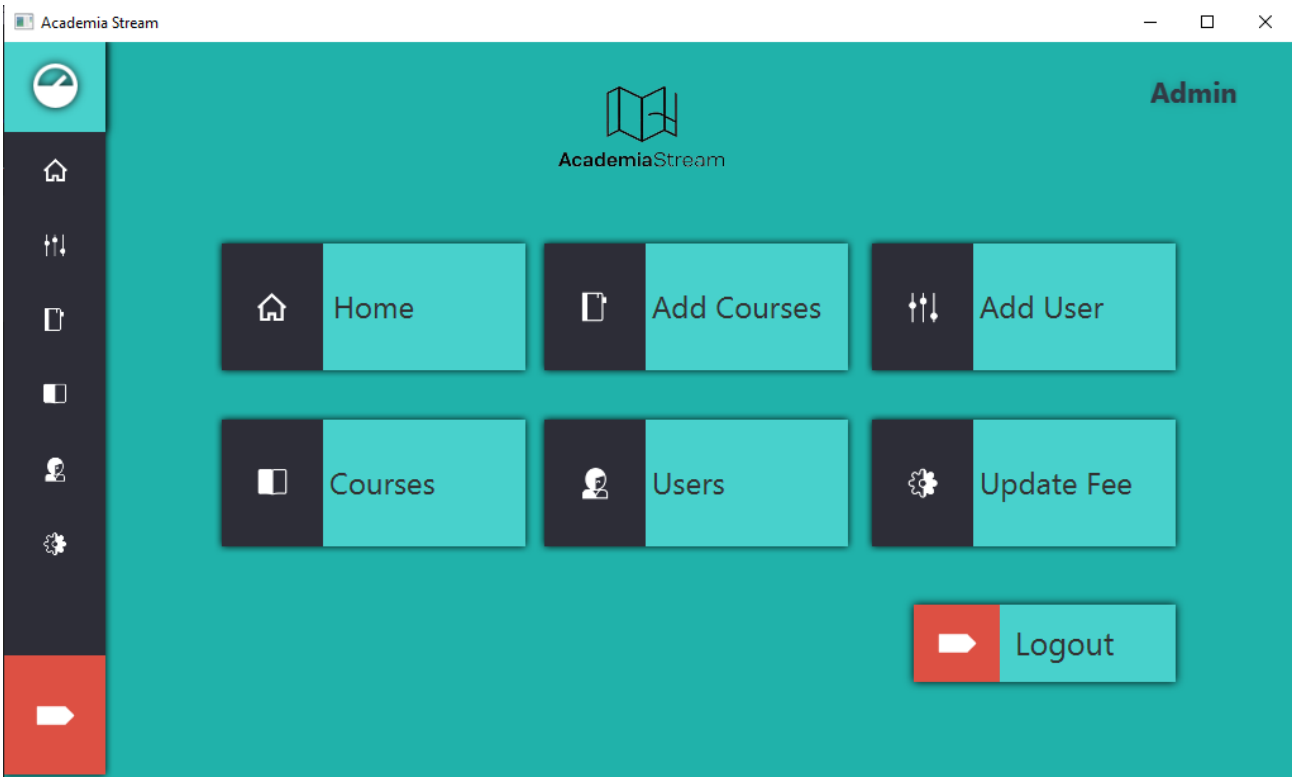
Navigation Panel: A vertical navigation panel is present on the left side with icons representing different functionalities. These include menu options, "Home," "Add Courses," "Add User," "Courses," "Users," and "Update Fee."

Main Interface: The main part of the interface consists of rectangular tiles arranged in two rows. Each tile represents different sections or functions available to the user: "Home," "Add Courses," "Add User," "Courses," "Users," and "Update Fee." These tiles have icons corresponding to their functions.

Logout Button: At the bottom right corner, there's a 'Logout' button. Its red color stands out from other elements, emphasizing its importance as an action button.

GUI Standards: The interface adheres to certain GUI standards such as consistent icon usage for intuitive navigation and clear demarcation of actionable items like logout buttons. The dark theme and color scheme provide visual segmentation and ease of use.

Software Components: The software components for which a user interface is needed include the Home, Add Courses, Add User, Courses, Users, and Update Fee sections. Each of these sections has its own set of functionalities and interactions that would be detailed in a separate user interface specification.



Faculty Dashboard

Software Name and Window Controls: At the top, the website's name "AcademiaStream" is displayed on the left. On the right, there are standard window control buttons (minimize, maximize, close).

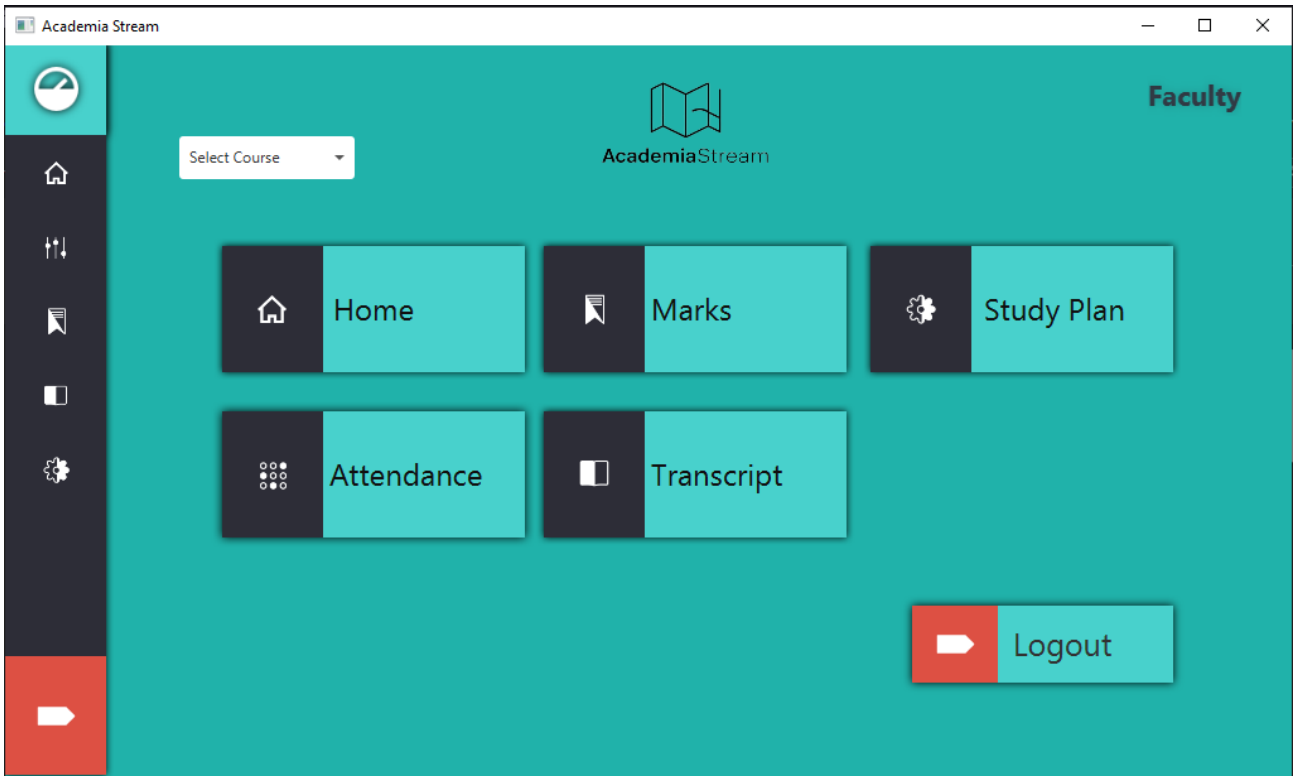
Navigation Panel: A vertical navigation panel is present on the left side with icons representing different functionalities. These include menu options, "Home," "Marks," "Study Plan," "Attendance," "Transcript,"

Main Interface: The main part of the interface consists of rectangular tiles arranged in two rows. Each tile represents different sections or functions available to the user: "Home," "Marks," "Study Plan," "Attendance," "Transcript," and an unlabelled red button which is for 'Logout'. These tiles have icons corresponding to their functions.

Logout Button: At the bottom right corner, there's a 'Logout' button. Its red color stands out from other elements, emphasizing its importance as an action button.

GUI Standards: The interface adheres to certain GUI standards such as consistent icon usage for intuitive navigation and clear demarcation of actionable items like logout buttons. The dark theme and color scheme provide visual segmentation and ease of use.

Software Components: The software components for which a user interface is needed include the Home, Marks, Study Plan, Attendance, Transcript, and Logout sections. Each of these sections likely has its own set of functionalities and interactions that would be detailed in a separate user interface specification.



3.7.3 Student Interfaces

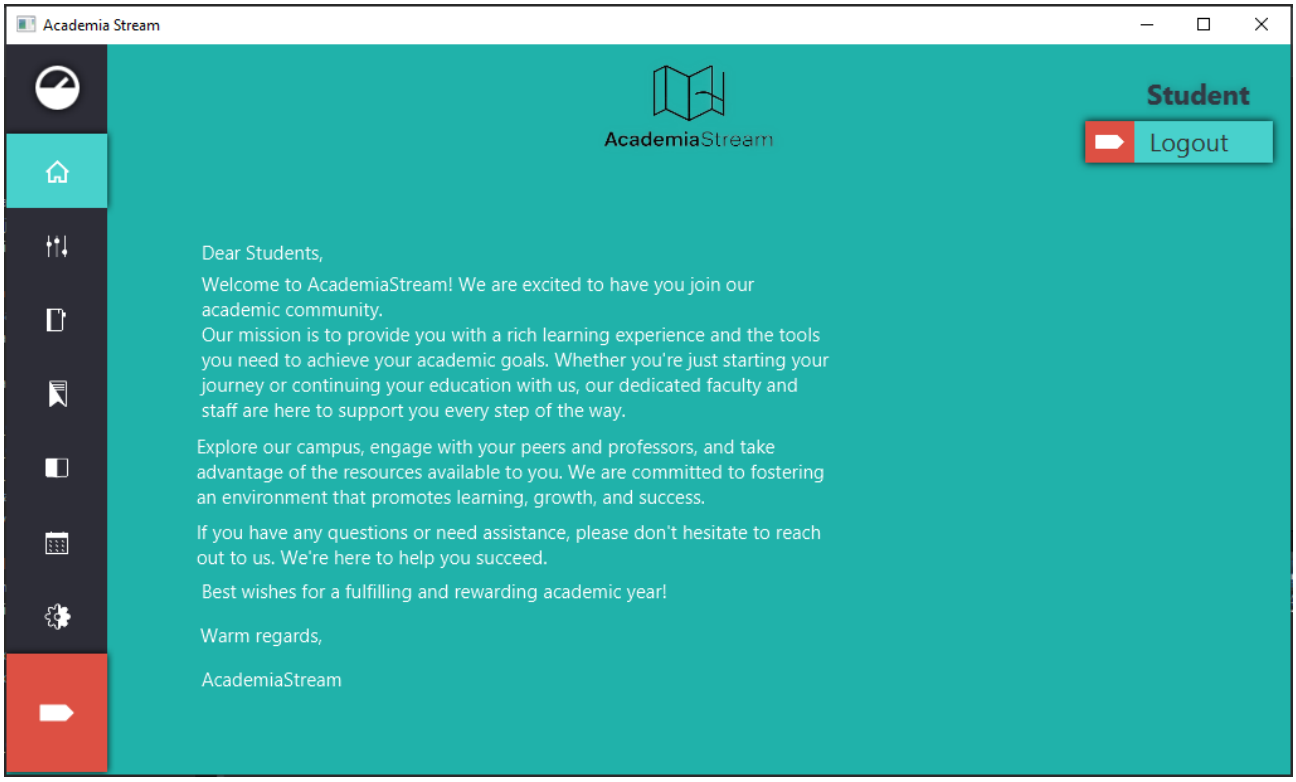
Home Page
Software Name and Window Controls: At the top, the website's name "AcademiaStream" is displayed on the left. On the right, there are standard window control buttons (minimize, maximize, close).
Navigation Panel: A vertical navigation panel is present on the left side with icons representing different functionalities. These include menu options, "Dashboard", "Home," "Registration," "Fee Details," "Attendance," "Marks," "Transcript," and "Study Plan.

Main Interface: The main part of the interface consists a welcome note for students.

Logout Button: At the bottom right corner, there's a 'Logout' button. Its red color stands out from other elements, emphasizing its importance as an action button.

GUI Standards: The interface adheres to certain GUI standards such as consistent icon usage for intuitive navigation and clear demarcation of actionable items like logout buttons. The dark theme and color scheme provide visual segmentation and ease of use.

Software Components: The software components for which a user interface is needed include the Home, Registration, Fee Details, Attendance, Marks, Transcript, and Study Plan sections. Each of these sections has its own set of functionalities and interactions that would be detailed in a separate user interface specification.



Fee Deatail Page

Page Title: The page is titled "FEE Details," indicating that it provides information about the fees associated with different courses.

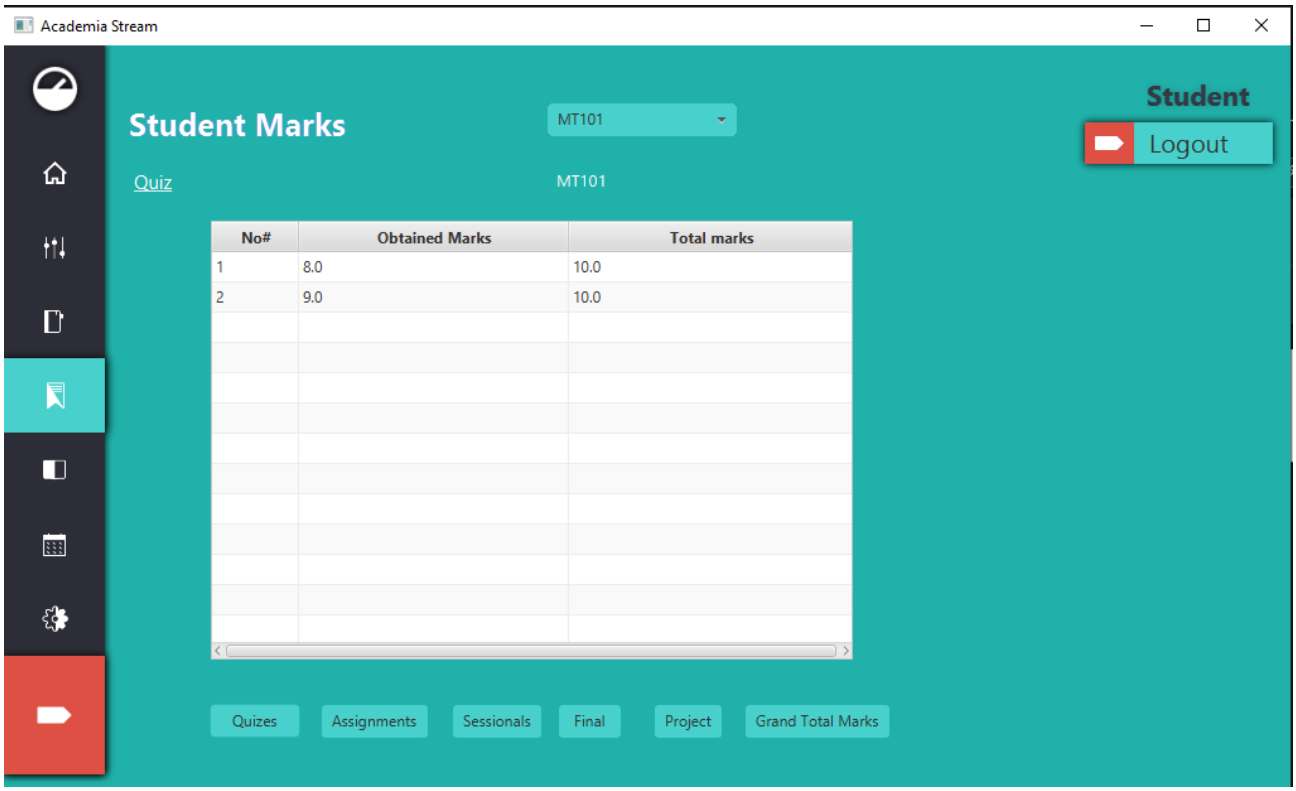
Table: The page includes a table with columns for Course ID, Course Name, and Credit Hrs. The table lists four courses with their respective IDs, names, and credit hours.

Total Calculated Fee: At the bottom of the table, there's a statement "Total Calculated FEE:" followed by "Rs. 66600". This indicates the total fee for the listed courses.

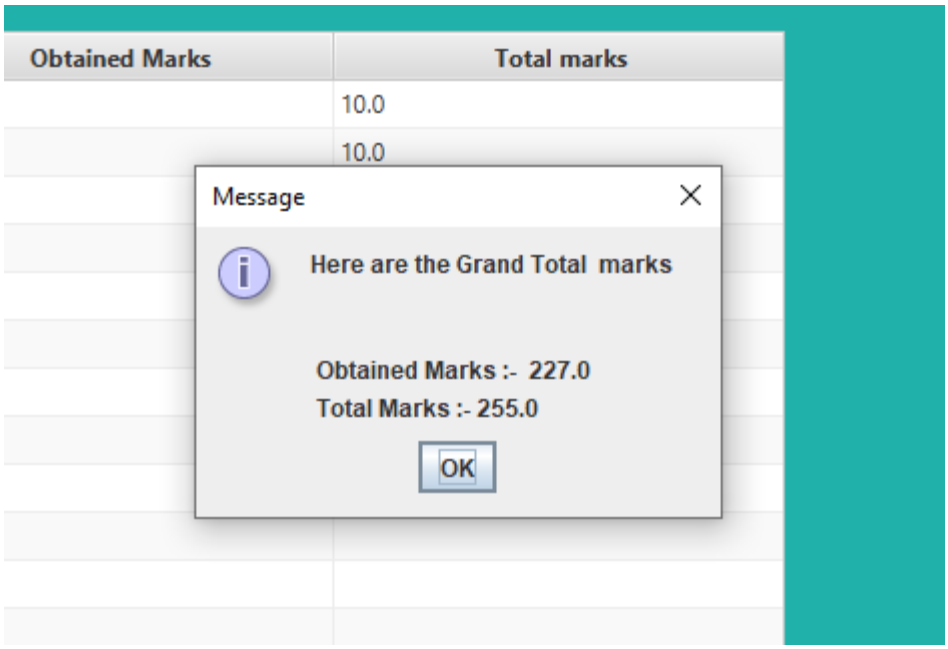
Logout Option: On the top right corner of the page, there's a 'Logout' option, indicating that this page is likely viewed from a user's account.

Software Components: The software components for which a user interface is needed include the table displaying the course details and the total calculated fee.

Marks Page
Page Title: The page is titled "Student Marks," indicating that it provides information about the marks obtained by a student in different courses.
Course Selection: There are dropdown menus for selecting courses labeled "MT:101" and "MT103."
Marks Table: The page includes a table with columns titled "No#," "Obtained Marks," and "Total marks." The table lists quiz scores for one course. The first row shows the number '1', an obtained mark of '80', and total marks of '100'. The second row is empty, indicating space for additional entries.
Bottom Panel: On the left side of the interface, there are various icons that seem to represent different functions or sections such as quizzes, assignments, sessionals, finals, projects, and grand total marks.
Logout Option: On the top right corner of the page, there's a 'Logout' option, indicating that this page is likely viewed from a student's account.
Software Components: The software components for which a user interface is needed include the table displaying the marks details, the course selection dropdowns, and the left side panel with various icons.


Academia Stream
Student Marks
MT101
Student
Logout
Quiz
MT101
No#
Obtained Marks
Total marks
1
8.0
10.0
2
9.0
10.0
Quizes
Assignments
Sessionals
Final
Project
Grand Total Marks

Grand Total



Transcript Page
Page Title: The page is titled "Student Transcript," indicating that it provides a transcript of the student's academic performance.

Student Information Section: This section includes:

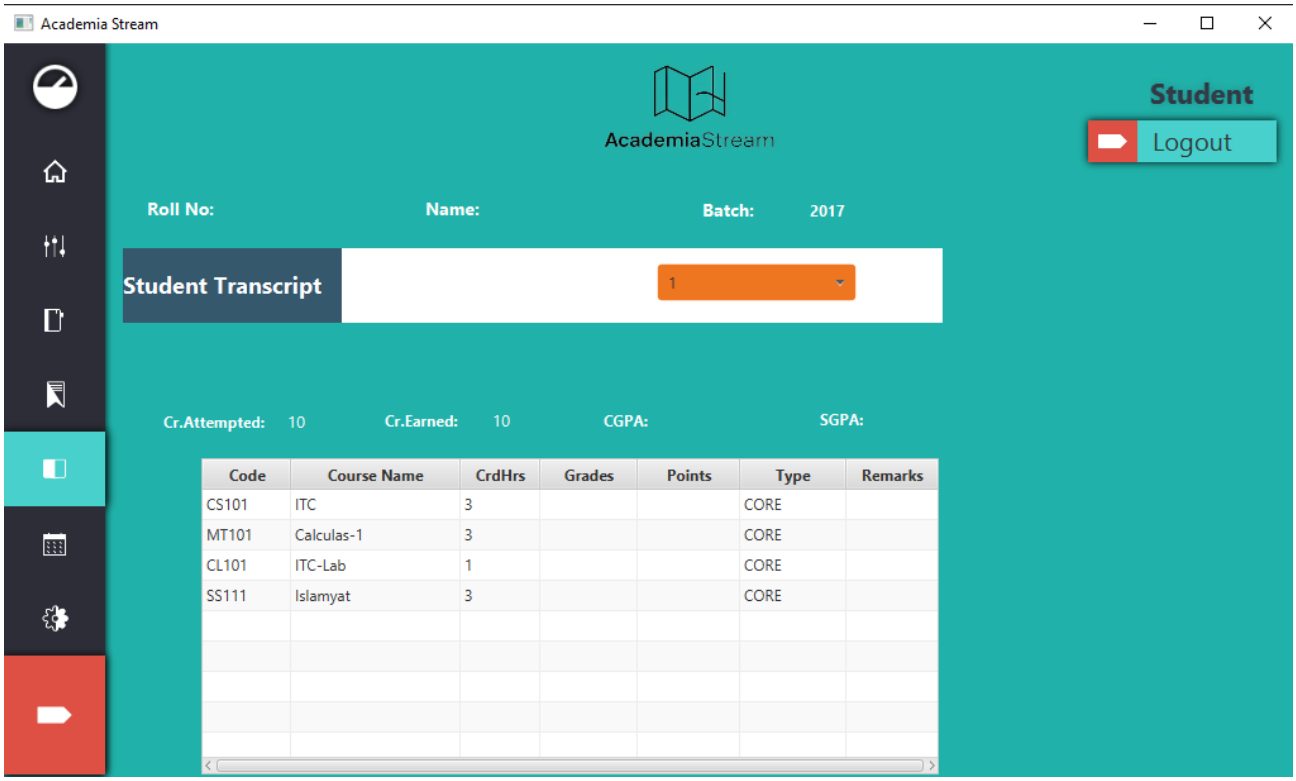
- "Roll No:" field with an input box for the student's roll number.
- "Name:" field with an input box for the student's name.
- "Batch:" field with a dropdown menu for selecting the batch year. The year 2017 is currently visible.

Logout Button: On the top right corner of the page, there's a 'Logout' button, indicating that this page is likely viewed from a student's account.

Transcript Table: The main part of the page is a table with the following column headers and entries:

- "Code" column: Lists the course codes for the courses the student has taken. The courses listed are MT101, CS101, and SS111.
- "Course Name" column: Lists the names of the courses. The courses listed are Calculus I, C++, and History.
- "Credits" column: Lists the number of credits for each course. Each course is listed as having 3 credits.
- "Grades" column: Lists the grades obtained in each course. The grades listed are B for Calculus I and C++ and A for History.
- "Points Scored" column: Lists the points scored in each course. "Remarks" column: Lists remarks for each course.

Total Marks Section: At the bottom of the table, there are totals for Attempted Credits (10), Earned Credits (10), CGPA (3.0), and SGPA. The CGPA and SGPA fields are currently empty.



Attendance Page
Page Title: The page is titled "Registered Courses," indicating that it provides information about the attendance of a student in different courses.

Student Information Section: This section includes:
"Courses" Dropdown with an ID number "MT201," indicating that this page is currently showing MT201 attendance
"Logout" button: On the top right corner of the page, there's a 'Logout' button.

Registered Courses Section: Below the header, there is a section titled "Registered Courses," which includes a dropdown menu for selecting courses.

Attendance Percentage: Next to the "Registered Courses" section, there is an orange rectangle displaying the text "Attendance Percentage: 66 %." Means Attendance is 66% currently.

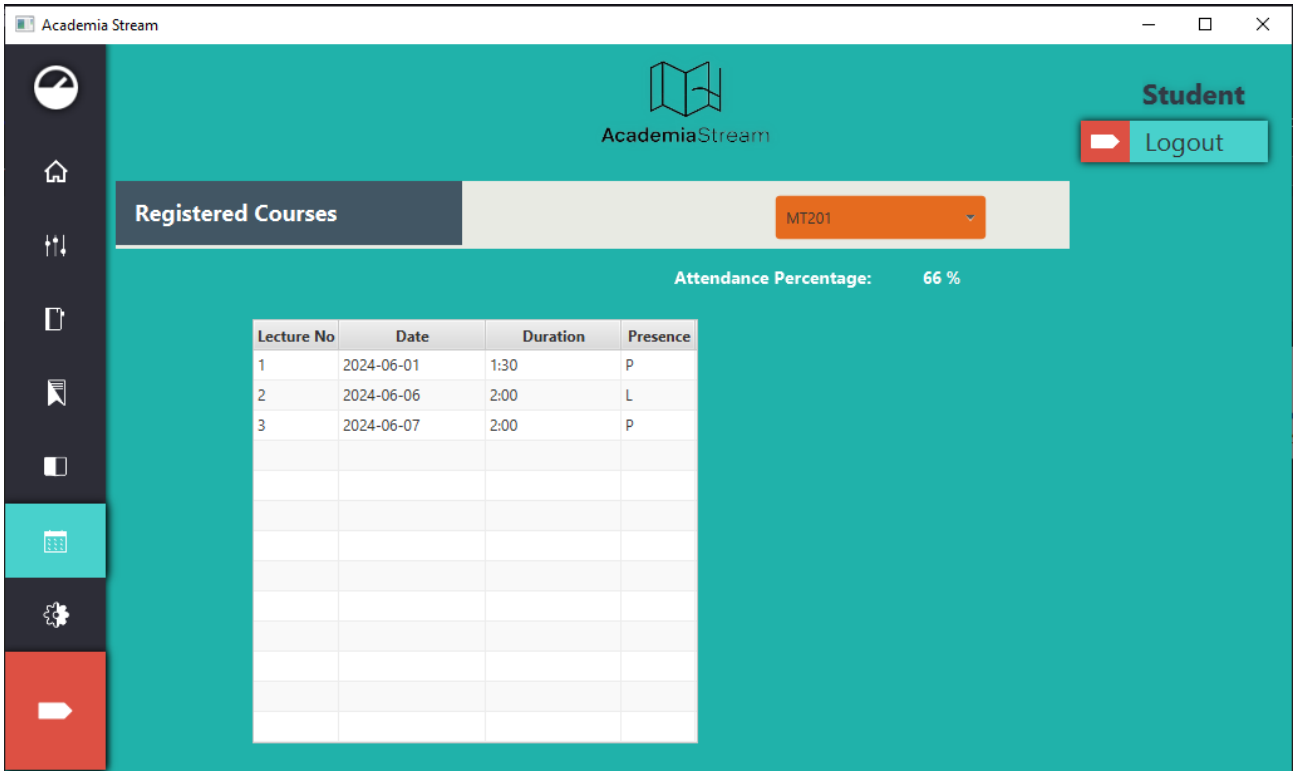
Attendance Table: The main part of the page is a table with the following column headers and entries:

- "Lecture No." column: Lists the lecture numbers as L1, L2, and L3.

- "Date" column: Lists dates in YYYY-MM-DD format: 2021-07-01, 2021-07-02, and 2021-07-03.
- "Duration" column: Lists durations as 1H 45M, 2H 30M, and 2H respectively.
- "Presence" column: This column is filled with the presence status of the student in each lecture.

Left Side Panel: On the left side of the interface is a vertical navigation bar with icons representing different functions or pages. Each icon is accompanied by a label indicating its function.

Design: The design uses contrasting colors like green for headers and navigation highlights; orange for important information like attendance percentage; white for table background; and gray for page background. This color scheme aids in distinguishing different sections of the interface.



Study Plan Page
Page Title: The page is titled "AcademiaStream" indicating that it's part of the University Management system

Student Information Section: This section includes:
"Logout" button: On the top right corner of the page, there's a 'Logout' button.

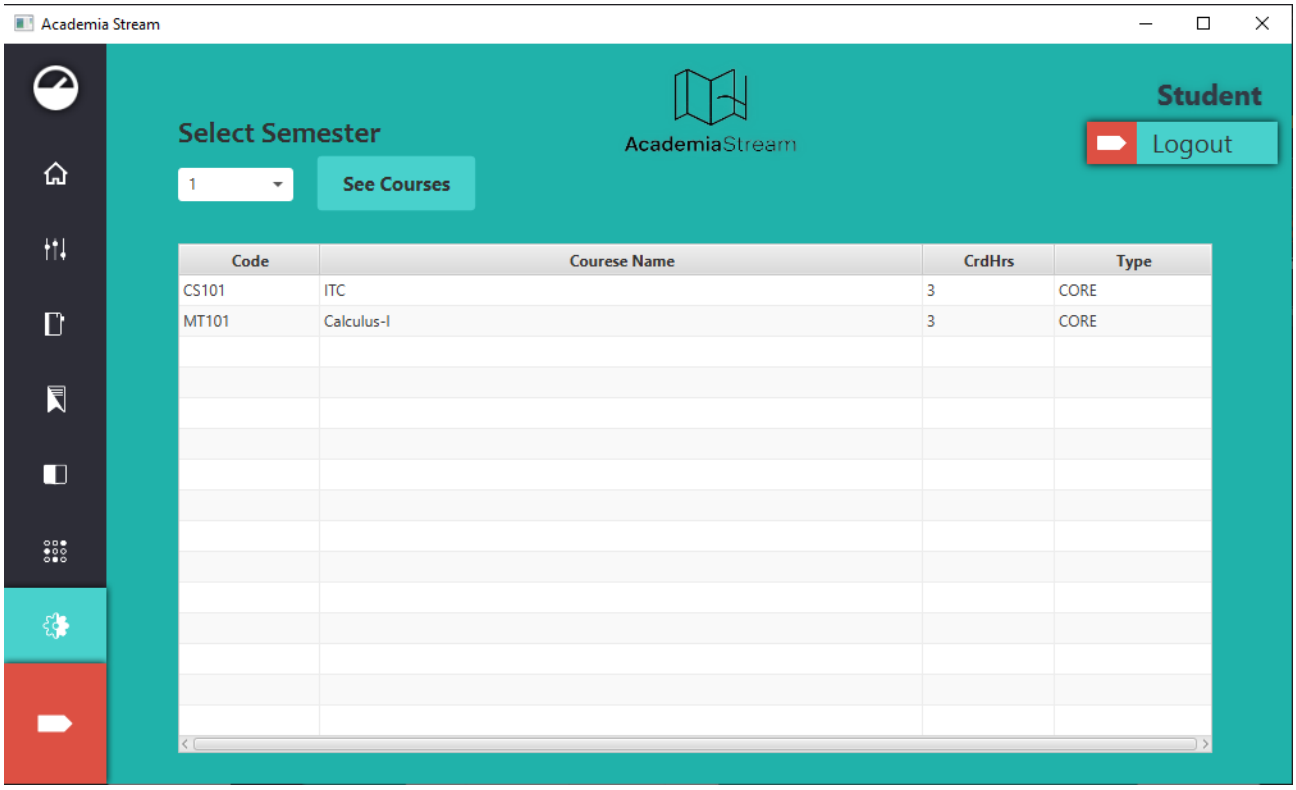
Select Semester Section: Below the header, there is a section titled "Select Semester," which includes a dropdown menu for selecting semesters. The dropdown menu is currently set to "1."

See Courses Button: Next to the "Select Semester" dropdown, there is a button labeled "See Courses." This button displays the courses available for the selected semester when clicked.

Registered Courses Table: The main part of the page is a table with the following column headers and entries:
"Code" column: Lists the course codes for the courses the student has registered for. The courses listed are CS101 and IDC.
"Courses Name" column: Lists the names of the courses. The courses listed are Calculus-I.
"Credits" column: Lists the number of credits for each course. Each course is listed as having 3 credits.
"Type" column: Lists the type of each course. All courses are marked as CORE currently.

Left Side Panel: On the left side of the interface is a vertical navigation bar with icons representing different functions or pages. Each icon is accompanied by a label indicating its function.

Design: The design uses contrasting colors like green for headers and navigation highlights; orange for important information like attendance percentage; white for table background; and gray for page background. This color scheme aids in distinguishing different sections of the interface.



3.7.4 Admin Interfaces

Home Page
Page Title: The page is titled "AcademiaStream," indicating that it's part of the University Management System.

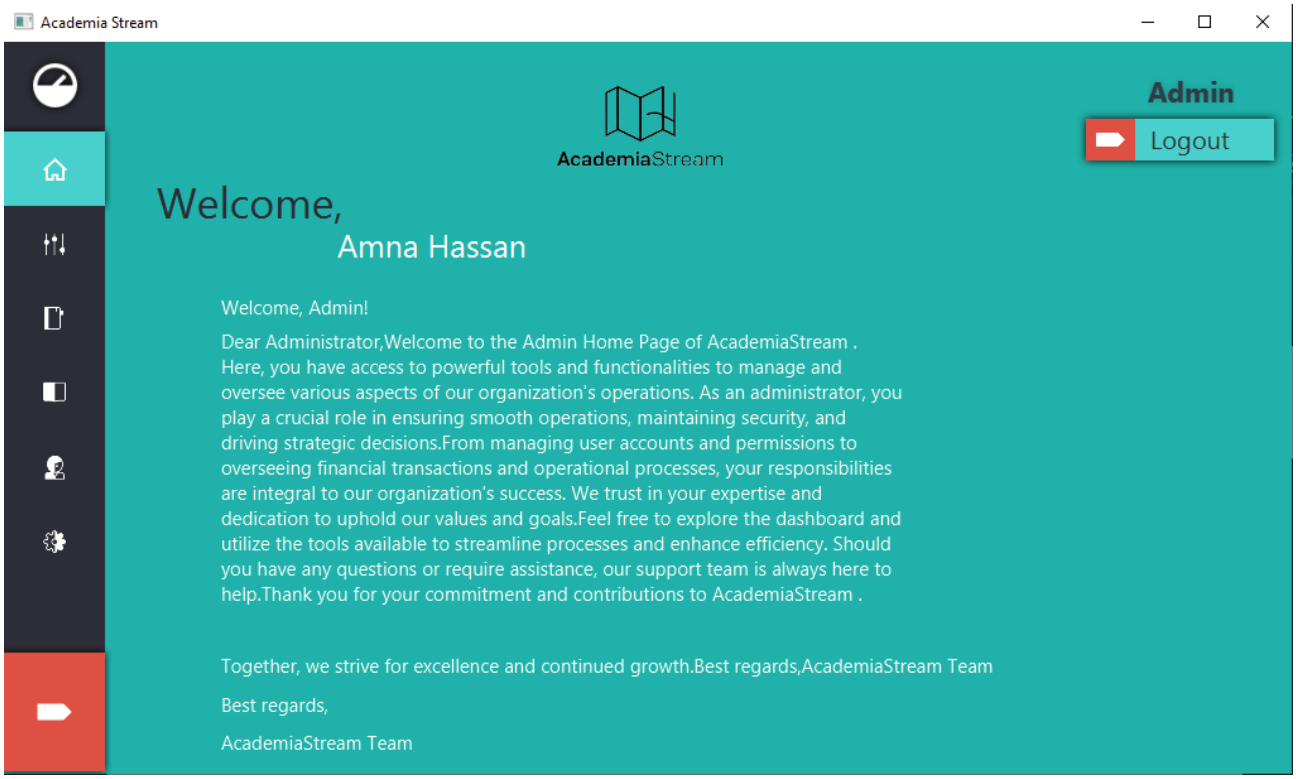
Admin Information Section: This section includes:

“Admin Name ” , indicating that this page is viewed from an admin’s account.
“Logout” button: On the top right corner of the page, there’s a ‘Logout’ button.

Welcome Message: Below the header, there is a welcome message addressing “Admin Hassan.” The message provides a detailed description of the services and responsibilities of an administrator on the AcademiaStream platform, including maintaining security, overseeing institutional networks, managing user accounts, and ensuring optimal performance. The text emphasizes the importance of the administrator’s role in shaping strategies from analytics and insights to drive institutional decisions. There is also gratitude expressed for contributions to AcademiaStream.

Left Side Panel: On the left side of the interface is a vertical navigation bar with icons representing different functions or pages. Each icon is accompanied by a label indicating its function.

Design: The design uses contrasting colors like green for headers and navigation highlights; orange for important information like attendance percentage; white for table background; and gray for page background. This color scheme aids in distinguishing different sections of the interface.



Add User Page
Page Title: The page is titled “AcademiaStream,”

Admin Information Section: This section includes:
“Admin” label, indicating that this page is viewed from an admin’s account.
“Logout” button: On the top right corner of the page, there’s a ‘Logout’ button.

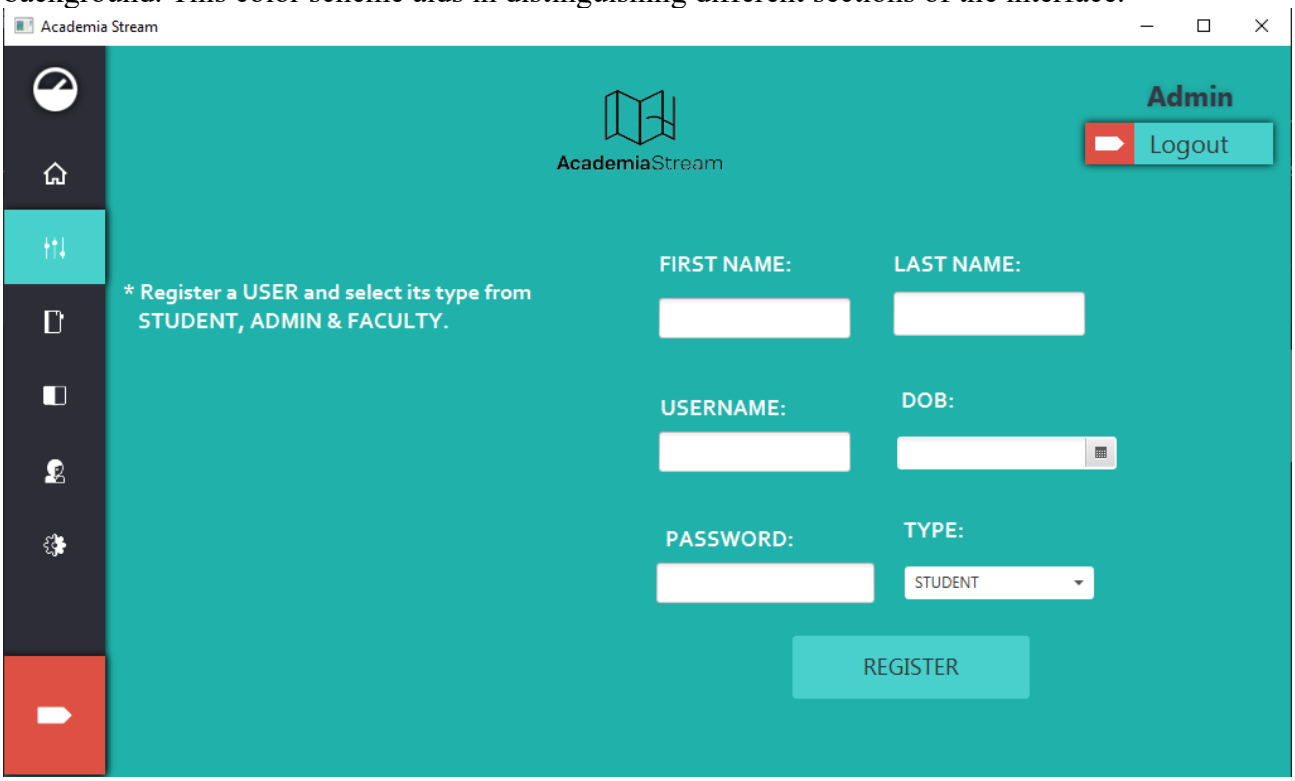
Add User Section: Below the header, there is a section titled "Register a USER and select its type from STUDENT, ADMIN & FACULTY." This section includes:

- "First Name" field with an input box.
- "Last Name" field with an input box.
- "Username" field with an input box.
- "Date of Birth (DOB)" field with an input box.
- "Password" field with an input box.
- "Type" field with a dropdown menu for selecting the user type (Student, Admin, or Faculty).

Register Button: Below the "Add User" section, there is a button labeled "REGISTER." This button likely creates a new user account with the entered details when clicked.

Left Side Panel: On the left side of the interface is a vertical navigation bar with icons representing different functions or pages. Each icon is accompanied by a label indicating its function.

Design: The design uses contrasting colors like green for headers and navigation highlights; orange for important information like attendance percentage; white for table background; and gray for page background. This color scheme aids in distinguishing different sections of the interface.



Add Course Page
Page Title: The page is titled "AcademiaStream,".

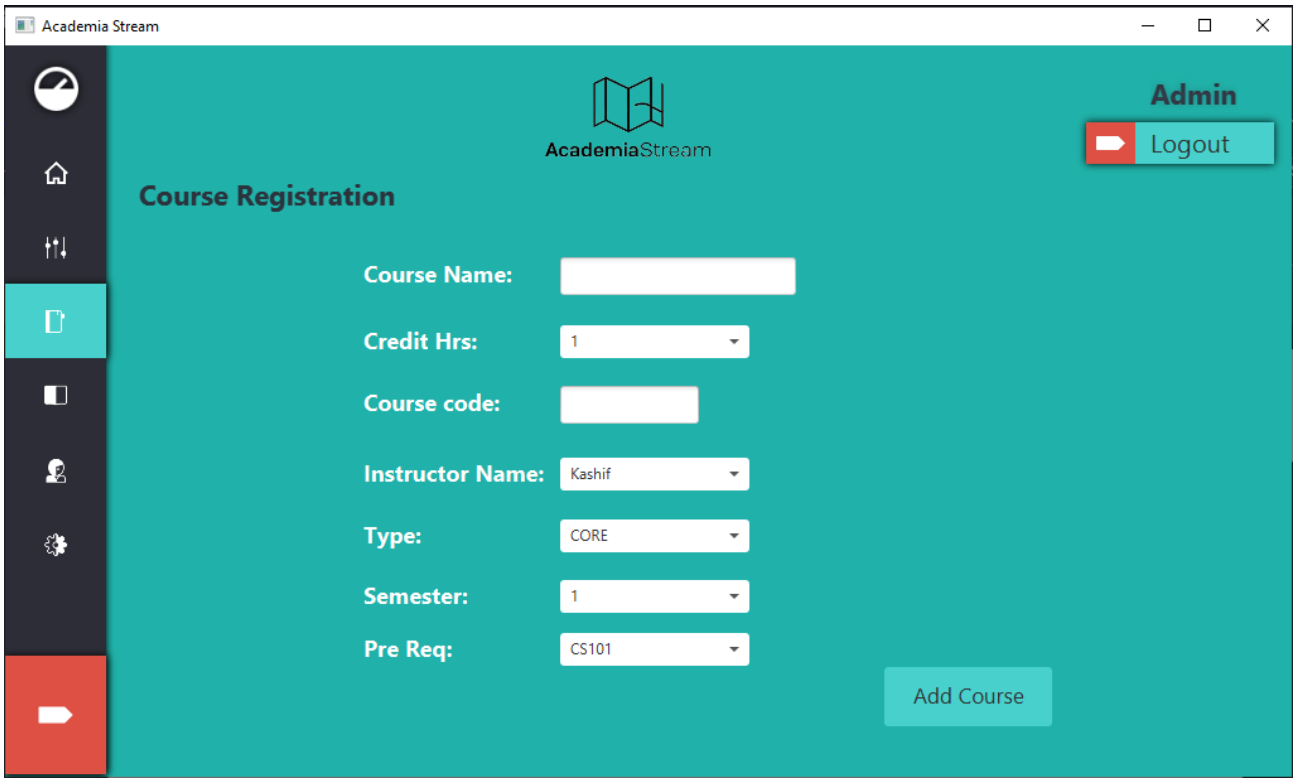
Admin Information Section: This section includes:
"Admin" label, indicating that this page is viewed from an admin's account.
"Logout" button: On the top right corner of the page, there's a 'Logout' button.

Course Registration Section: Below the header, there is a section titled "Course Registration," which includes:

- "Course Name" field with an input box.
- "Credit Hours" field with a dropdown menu.
- "Course Code" field with an input box.
- "Instructor Name" field with an input box pre-filled with 'Kashif'.
- "Type" field with a dropdown menu for selecting the course type pre-filled as CORE.
- "Semester" field with an input box.
- "Pre Req (Pre Requisites)" field with an input box pre-filled with 'CS101'.

Add Course Button: Below the "Course Registration" section, there is a button labeled "Add Course." This button creates a new course with the entered details when clicked.

Left Side Panel: On the left side of the interface is a vertical navigation bar with icons representing different functions or pages. Each icon is accompanied by a label indicating its function.
Design: The design uses contrasting colors like green for headers and navigation highlights; orange for important information like attendance percentage; white for table background; and gray for page background. This color scheme aids in distinguishing different sections of the interface.



Update / Remove Courses
Page Title: The page is titled "AcademiaStream,"

Admin Information Section: This section includes:

- "Admin" label, indicating that this page is likely viewed from an admin's account.

- "Logout" button: On the top right corner of the page, there's a 'Logout' button.

Course Update Section: Below the header, there is a section for updating or removing courses. This section includes:
A table with the following column headers and entries:

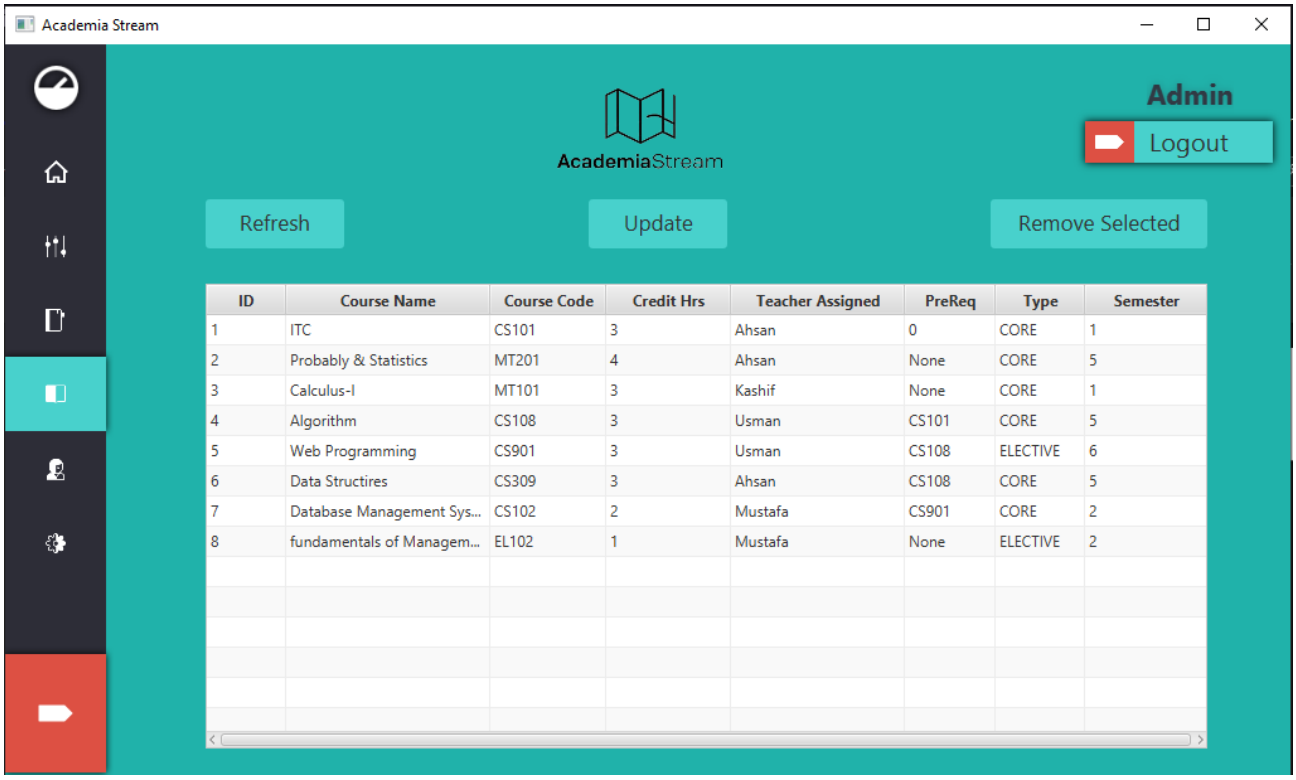
- "ID" column: Lists the course IDs.
- "Course Name" column: Lists the names of the courses.
- "Course Code" column: Lists the course codes.
- "Credit Hours" column: Lists the number of credit hours for each course.
- "Teacher Assigned" column: Lists the teachers assigned to each course.
- "Prerequisite" column: Lists the prerequisites for each course.
- "Semester" column: Lists the semester in which each course is offered.

Checkbox: Each row has a checkbox on the left side, likely used to select courses for updating or removing.

- "Refresh" button: refreshes the page or the table.
- "Update" button: updates the selected course(s) based on changes made by the admin.
- "Remove Selected" button: removes the selected course(s) from the system.

Left Side Panel: On the left side of the interface is a vertical navigation bar with icons representing different functions or pages. Each icon is accompanied by a label indicating its function.

Design: The design uses contrasting colors like green for headers and navigation highlights; orange for important information like attendance percentage; white for table background; and gray for page background. This color scheme aids in distinguishing different sections of the interface.



ID	Course Name	Course Code	Credit Hrs	Teacher Assigned	PreReq	Type	Semester
1	ITC	CS101	3	Ahsan	0	CORE	1
2	Probably & Statistics	MT201	4	Ahsan	None	CORE	5
3	Calculus-I	MT101	3	Kashif	None	CORE	1
4	Algorithm	CS108	3	Usman	CS101	CORE	5
5	Web Programming	CS901	3	Usman	CS108	ELECTIVE	6
6	Data Structires	CS309	3	Ahsan	CS108	CORE	5
7	Database Management Sys...	CS102	2	Mustafa	CS901	CORE	2
8	fundamentals of Managem...	EL102	1	Mustafa	None	ELECTIVE	2

Update Form
Form Title: The form is titled "Update Course," indicating that it's used to update the details of a course.

Course Name: This field is for entering the name of the course. In the provided image, it's filled with "Algorithm."
Course Code: This field is for entering the code of the course. In the provided image, it's filled with "CS108."

Assigned Teacher: This field is for entering the name of the teacher assigned to the course. In the provided image, it's filled with "Usman."

Type: This field is for selecting the type of the course. It's a dropdown menu with "CORE" selected in the provided image.

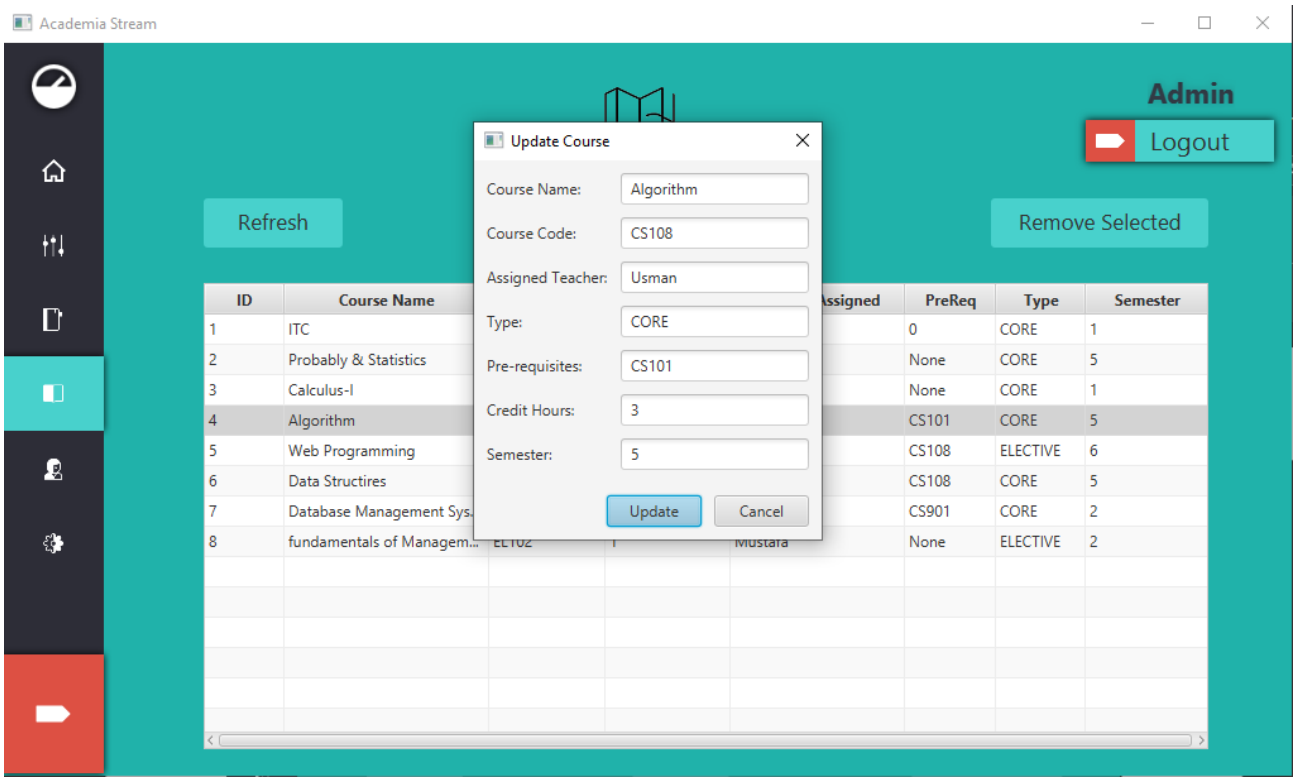
Pre-requisites: This field is for entering the prerequisites for the course. In the provided image, it's filled with "CS101."

Credit Hours: This field is for entering the number of credit hours of the course. In the provided image, it's set to '3.'

Semester: This field is for entering the semester in which the course is offered. In the provided image, it's set to '5.'

Update Button: This button is used to submit the form and update the course details in the system.

Cancel Button: This button is used to cancel the operation and go back to the previous page.



Update/Remove User

Page Title: The page is titled "AcademiaStream,"Admin Information Section: This section includes:

- "Admin" label, indicating that this page is likely viewed from an admin's account.
- "Logout" button: On the top right corner of the page, there's a 'Logout' button.

User Table: The main part of the page is a table with the following column headers and entries:

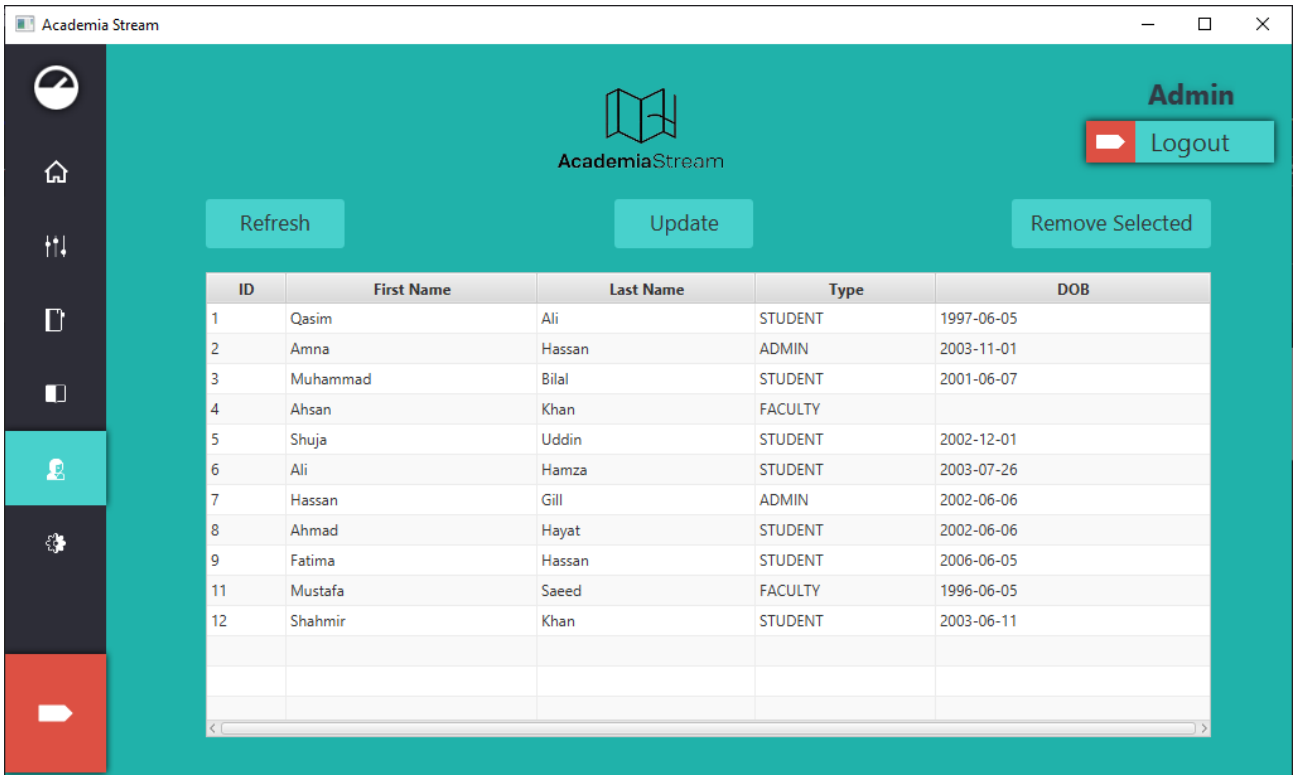
- "ID" column: Lists the user IDs.
- "First Name" column: Lists the first names of the users.
- "Last Name" column: Lists the last names of the users.
- "Type" column: Lists the types of the users (ADMIN or STUDENT).
- "DOB" column: Lists the dates of birth of the users.

Checkbox: Each row has a checkbox on the left side, likely used to select users for updating or removing.

Refresh Button: This button is used to refresh the page or the table.

Update Button: This button is used to update the details of the selected user(s) in the system.

Remove Selected Button: This button is used to remove the selected user(s) from the system.



User Update Form
Form Title: The form is titled "Update User,"

First Name: This field is for entering the first name of the user. In the provided image, it's filled with "Ahsan."

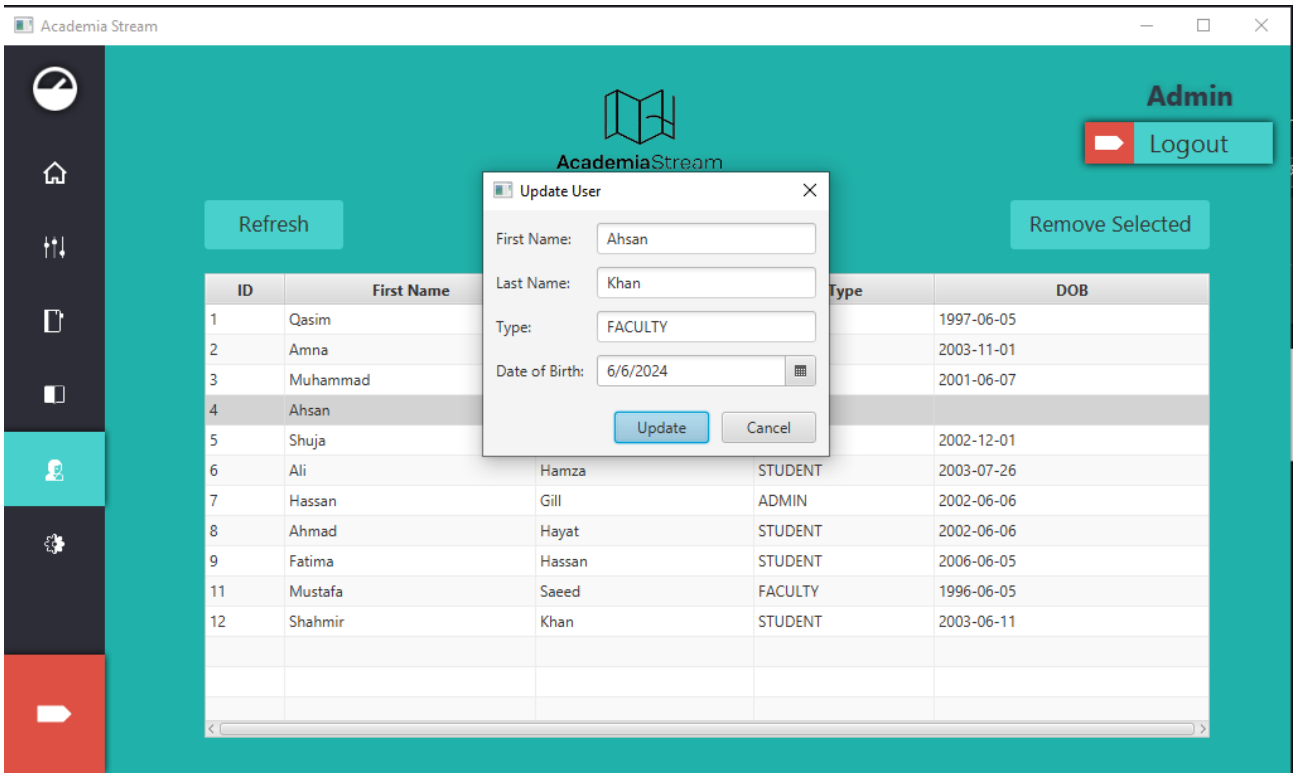
Last Name: This field is for entering the last name of the user. In the provided image, it's filled with "Khan."

Type: This field is for selecting the type of the user. It's a dropdown menu with "FACULTY" selected in the provided image.

Date of Birth (DOB): This field is for entering the date of birth of the user. In the provided image, it's set to '6/6/2024.'

Update Button: This button is likely used to submit the form and update the user details in the system.

Cancel Button: This button is likely used to cancel the operation and go back to the previous page.

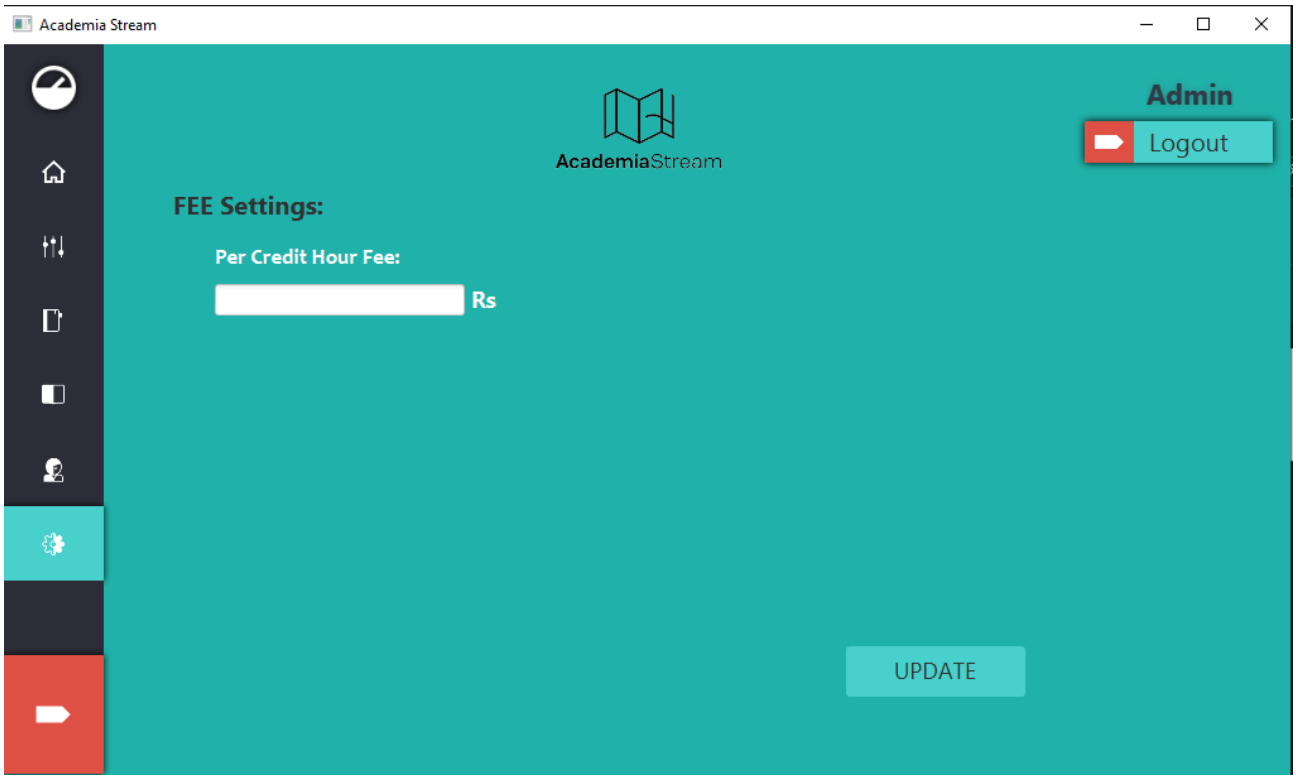


Set Fee Page
Page Title: The page is titled "AcademiaStream,"

Admin Information Section: This section includes:

- "Admin" label, indicating that this page is likely viewed from an admin's account.
- "Logout" button: On the top right corner of the page, there's a 'Logout' button.

Fee Settings Section: Below the header, there is a section titled "FEE Settings," which includes: "Per Credit Hour Fee" field with an input box for entering the fee per credit hour in Rupees (Rs).
Update Button: Below the "FEE Settings" section, there is a button labeled "UPDATE." This button is likely used to submit the form and update the fee settings in the system.



3.7.5 Faculty Interfaces

Assign Marks Page
Page Title: The page is titled "AcademiaStream,"

Admin Information Section: This section includes:

- "Admin" label, indicating that this page is likely viewed from an admin's account.
- "Logout" button: On the top right corner of the page, there's a 'Logout' button.

Marks Management Section: Below the header, there is a section titled "Marks Management," which includes:

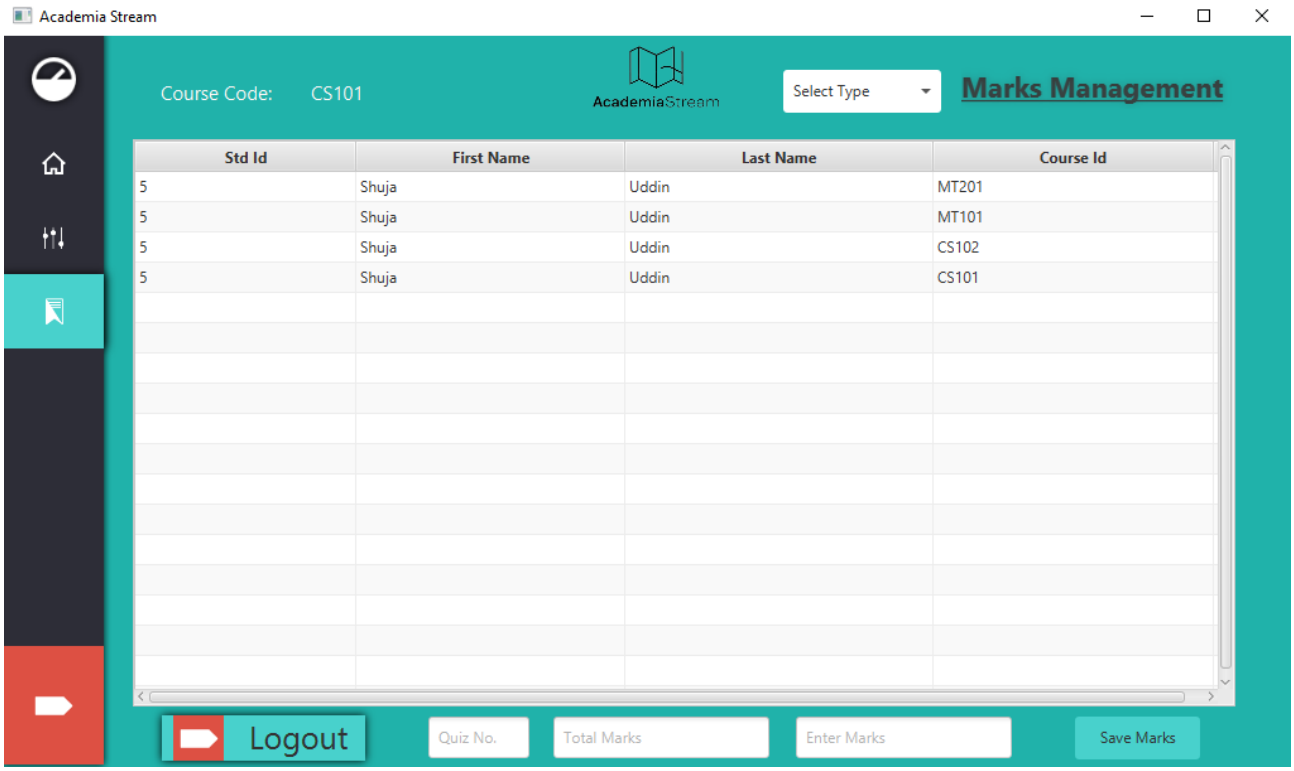
- "Course Code" field with an input box.
- "Academic Year" field with an input box.
- "Subject Type" field with an dropdown box.
-

User Table: The main part of the page is a table with the following column headers and entries:

- "Std Id" column: Lists the student IDs.
- "First Name" column: Lists the first names of the students.
- "Last Name" column: Lists the last names of the students.
- "Course Id" column: Lists the course IDs for the courses the students are enrolled in.

Save Marks Button: This button is likely used to save the marks assigned to the students for the assessments.

Left Side Panel: On the left side of the interface is a vertical navigation bar with icons representing different functions or pages. Each icon is accompanied by a label indicating its function.



Attendance Marking Page
Page Title: The page is titled "AcademiaStream,"

Admin Information Section: This section includes:

- "Admin" button, indicating that this page is likely viewed from an admin's account.
- "Logout" button: On the top right corner of the page, there's a 'Logout' button.
- Course Code Section: Below the header, there is a section titled "Course Code," which includes:
- "Course Code" field with an input box. In the provided image, it's filled with "CS309."

Select Type Section: This section includes:

- "Select Type" field with a dropdown menu for selecting the type of the attendance.

Attendance Section: This section includes:

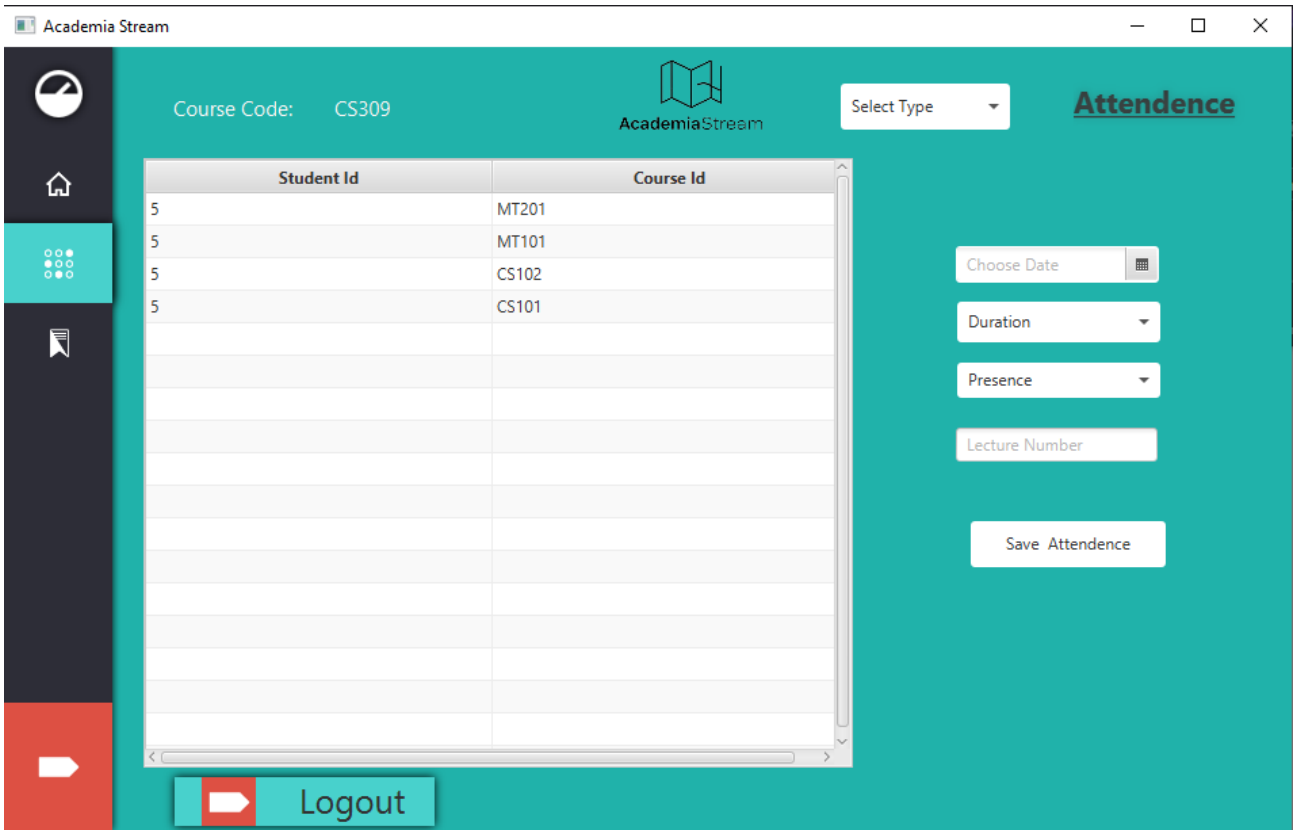
- "Attendance" field with a dropdown menu for selecting the attendance status.
- Student Table: The main part of the page is a table with the following column headers and entries:
- "Std Id" column: Lists the student IDs.
- "First Name" column: Lists the first names of the students.
- "Last Name" column: Lists the last names of the students.
- "Course Id" column: Lists the course IDs for the courses the students are enrolled in.

Duration Field: This field is for entering the duration of the lecture.

Choose Date Field: This field is for selecting the date of the lecture.

Presence Number Field: This field is for entering the presence number.

Save Attendance Button: This button is used to save the attendance details in the system



3.7.6 Error And Success Alerts

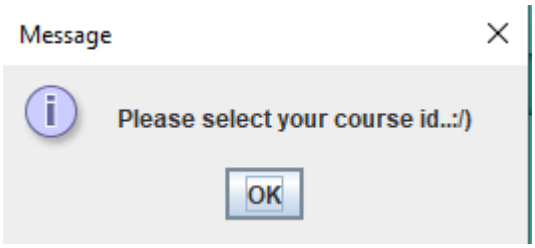
Success Messages: These messages are displayed when an operation or task has been successfully completed. They typically include a confirmation of the action taken. For example:

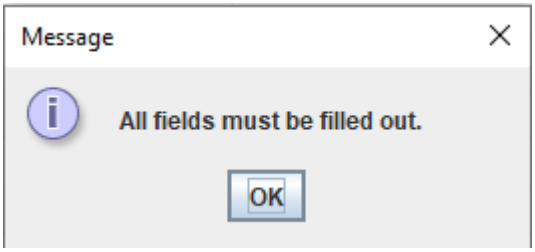
- "New course registered successfully!" This message confirms that a new course has been added to the system.
- "Saved Successfully!" This message confirms that changes made by the user have been saved.

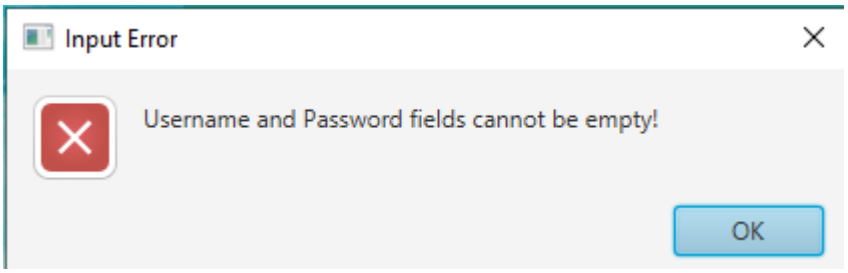
Error Messages: These messages are displayed when an operation or task could not be completed due to some error or invalid input. They typically include a description of the issue and how to resolve it. For example:

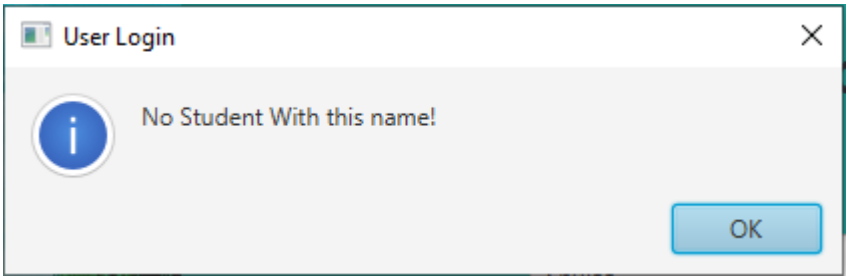
- "Obtained marks cannot be greater than total marks." This message indicates that the marks entered are invalid because the obtained marks exceed the total marks.

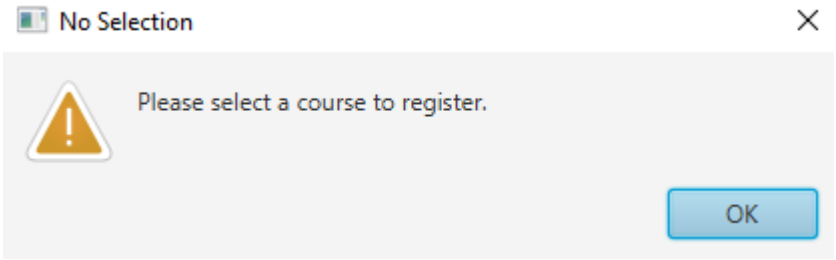
- "Marks and Total Marks must be numeric values." This message indicates that the input is invalid because non-numeric values were entered for marks.
- "Please select a student and try again before saving selection." This message indicates that no student was selected before attempting to save.

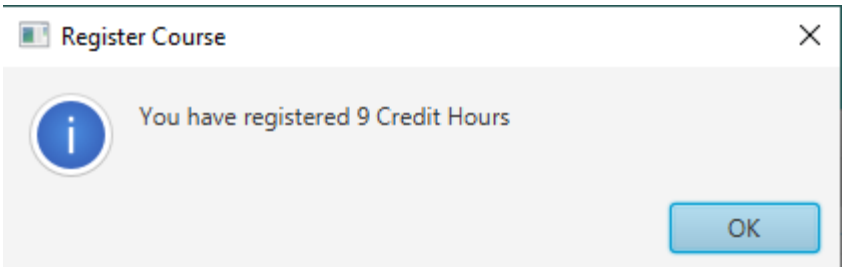


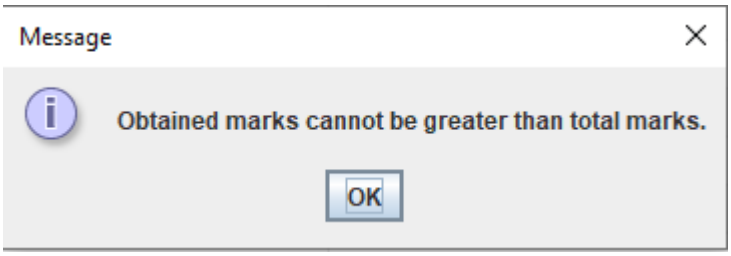


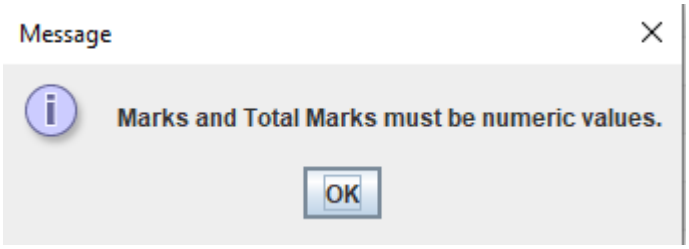


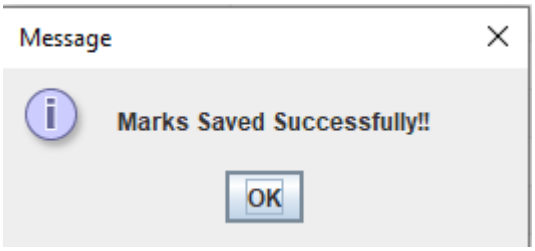


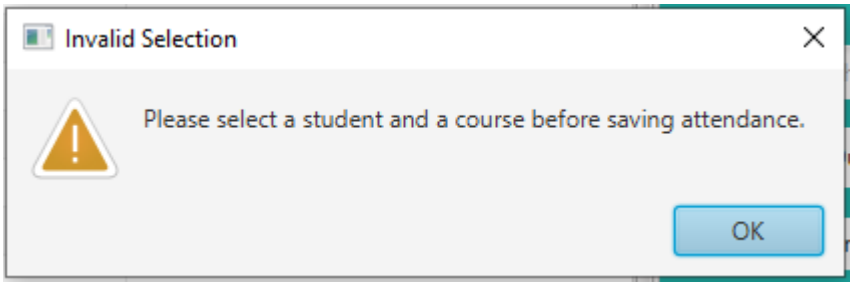


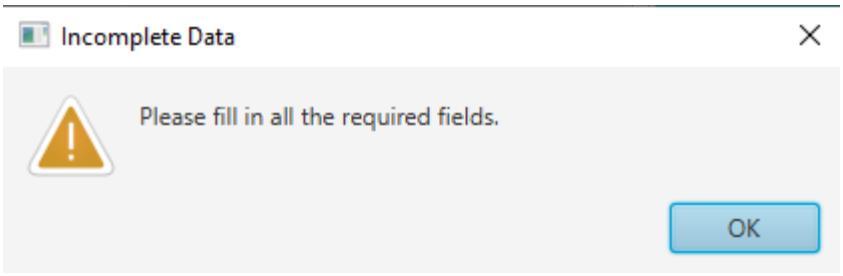

Register Course
You have registered 9 Credit Hours
OK

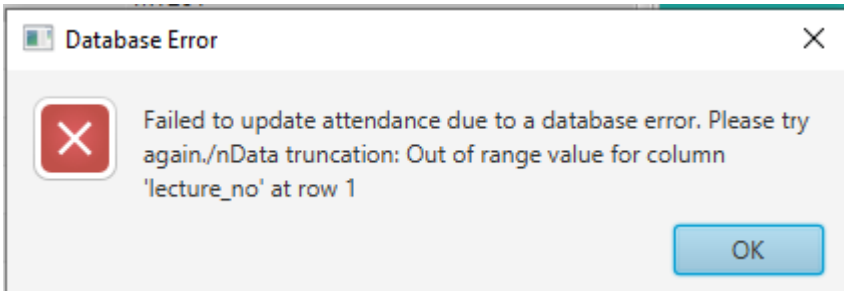

Message
Obtained marks cannot be greater than total marks.
OK

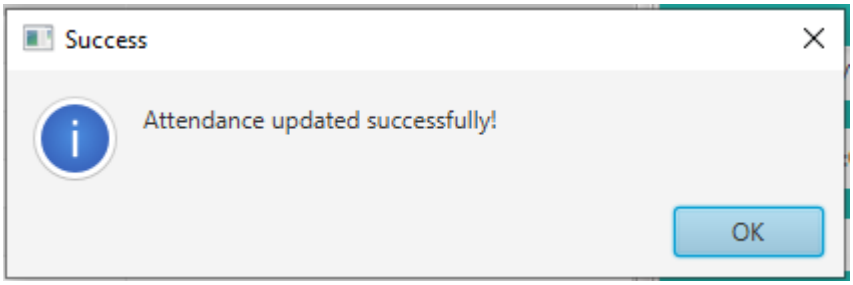

Message
Marks and Total Marks must be numeric values.
OK


Message
Marks Saved Successfully!!
OK

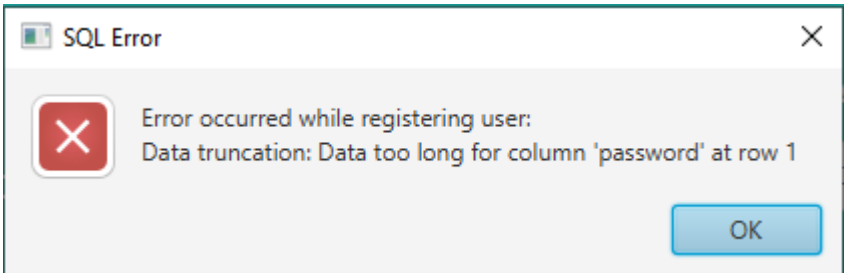

Invalid Selection
Please select a student and a course before saving attendance.
OK

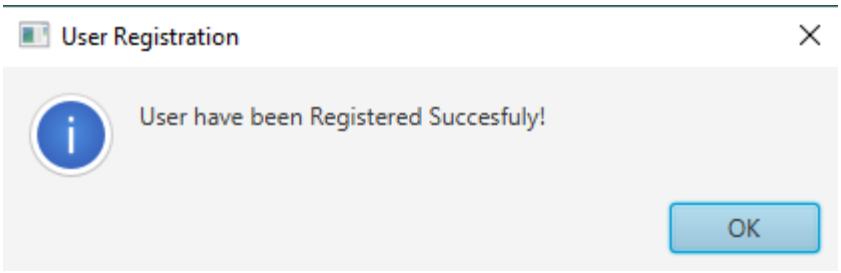

Incomplete Data
Please fill in all the required fields.
OK

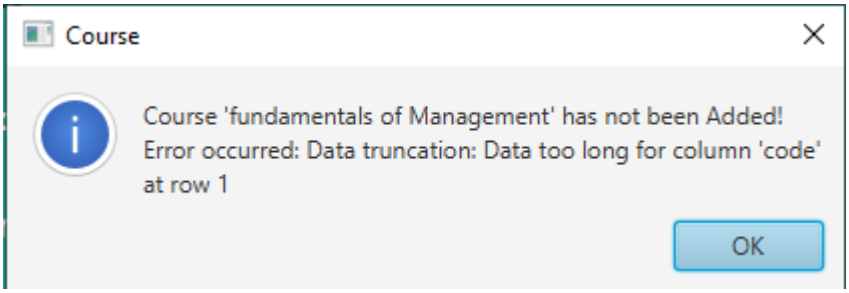

Database Error
Failed to update attendance due to a database error. Please try again./nData truncation: Out of range value for column 'lecture_no' at row 1
OK

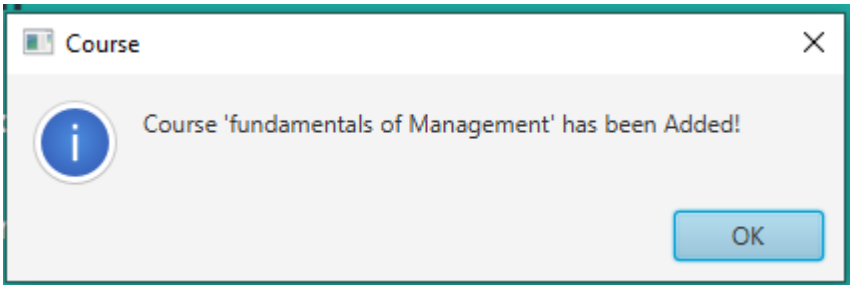

Success
Attendance updated successfully!
OK


User Registration
Fill All the Fields! :(
OK


SQL Error
Error occurred while registering user:
Data truncation: Data too long for column 'password' at row 1
OK







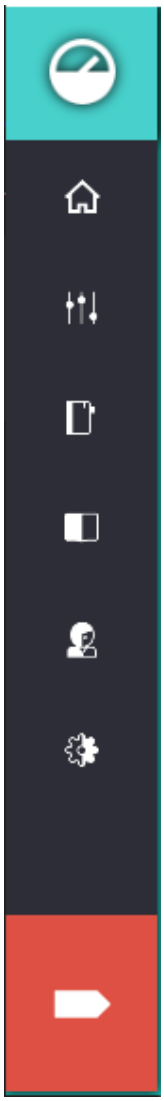
3.7.7 Navigation Bars

Admin Nav Bar
Dashboard Icon: The topmost icon, which is typically used to represent the home page or dashboard. Clicking this would likely take the admin back to the main admin page or dashboard.

Additional Icons: Below the home icon, there are five additional icons. While the specific functions of these icons aren't clear from the image, they typically represent different sections or functionalities of the admin interface. These could include sections for Add User, Add Course, View Users, View Courses, Update Fee and Home.

Logout Button: At the bottom of the navigation bar, there's a button that likely serves as the logout function. Admins can use this to securely log out of their admin session.

Navigation Bar Design: The navigation bar has a dark background, which helps to distinguish it from the rest of the page. The icons are light-colored for contrast, making them stand out against the dark background.



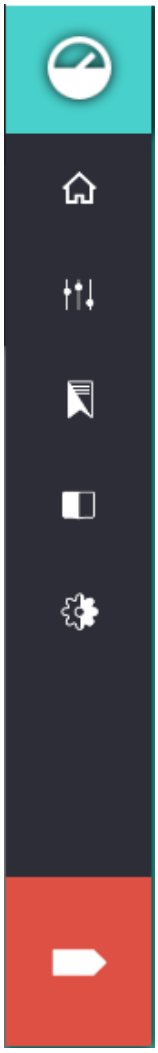
Faculty Nav Bar

Dashboard Icon: The topmost icon, which is typically used to represent the home page or dashboard. Clicking this would likely take the admin back to the main admin page or dashboard.

Additional Icons: Below the home icon, there are five additional icons. While the specific functions of these icons aren't clear from the image, they typically represent different sections or functionalities of the admin interface. These could include sections Marking attendance, Assign marks, view transcript, And Study Plan

Logout Button: At the bottom of the navigation bar, there's a button that likely serves as the logout function. Admins can use this to securely log out of their admin session.

Navigation Bar Design: The navigation bar has a dark background, which helps to distinguish it from the rest of the page. The icons are light-colored for contrast, making them stand out against the dark background.

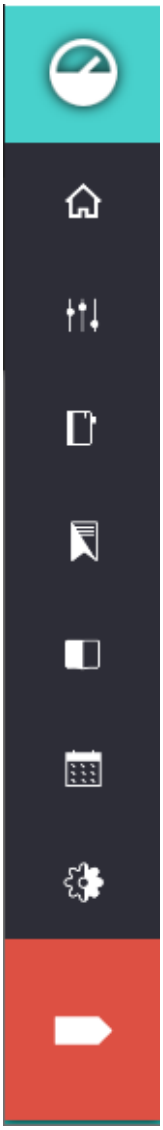


Faculty Nav Bar
Dashboard Icon: The topmost icon, which is typically used to represent the home page or dashboard. Clicking this would likely take the admin back to the main admin page or dashboard.

Additional Icons: Below the home icon, there are five additional icons. While the specific functions of these icons aren't clear from the image, they typically represent different sections or functionalities of the admin interface. These could include sections for Add User, Add Course, View Users, View Courses, Update Fee and Home.

Logout Button: At the bottom of the navigation bar, there's a button that likely serves as the logout function. Admins can use this to securely log out of their admin session.

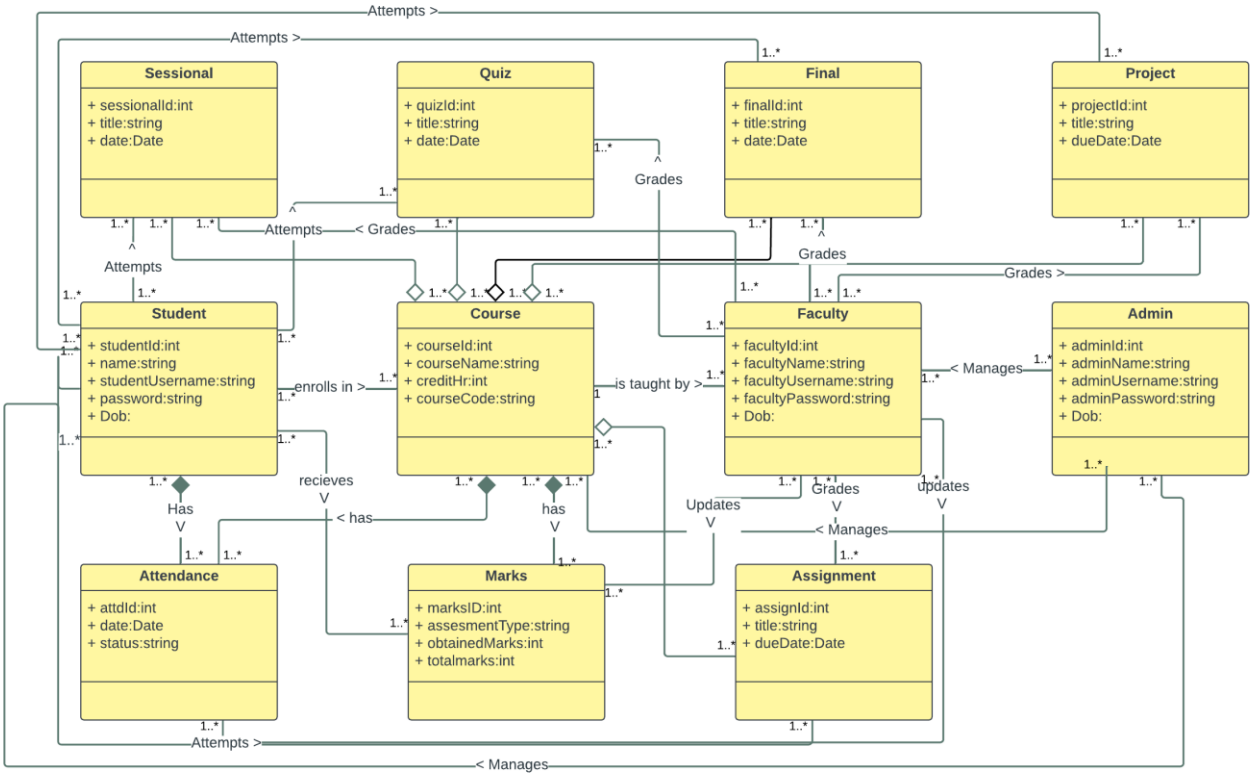
Navigation Bar Design: The navigation bar has a dark background, which helps to distinguish it from the rest of the page. The icons are light-colored for contrast, making them stand out against the dark background.



This description offers a thorough examination of every component, label, and button on a particular software application page. A separate user interface specification document would be required for a more thorough design specification. This document would provide particular information about each component and how it interacts with the user, as well as any guidelines or style standards related to the GUI that must be followed.

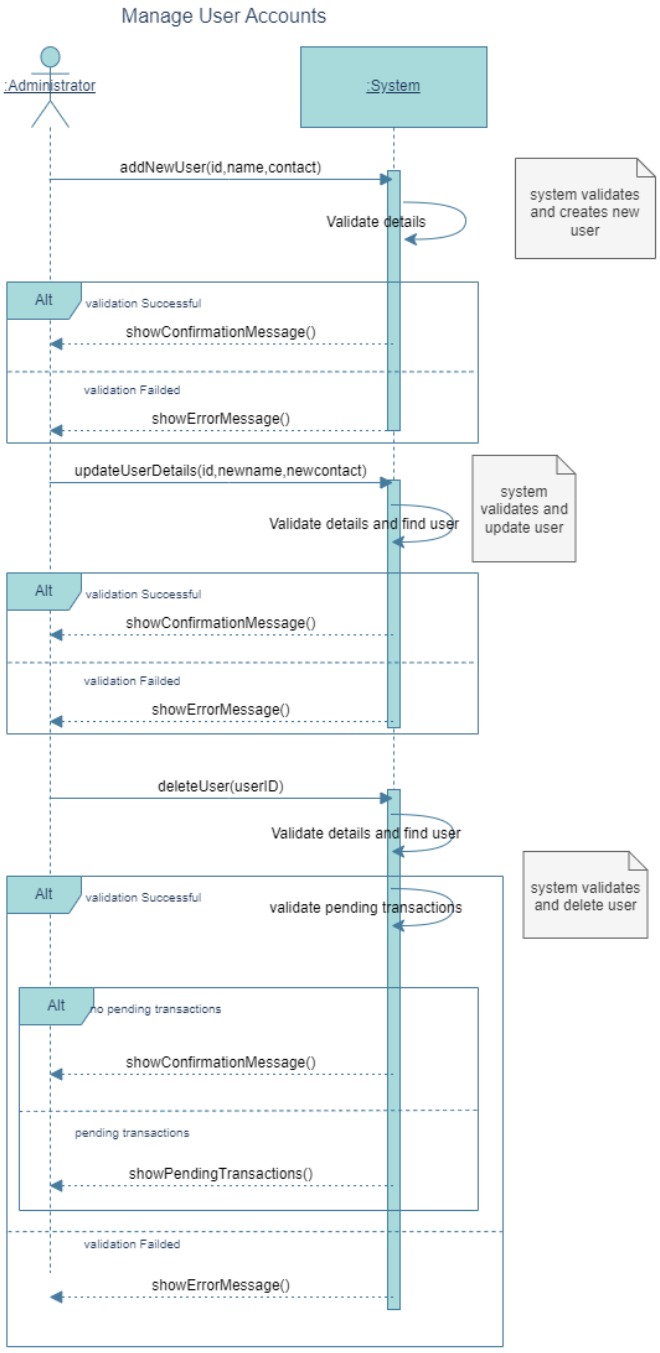
Note that this is a high-level summary based on a picture that has been supplied.

4. Domain Model

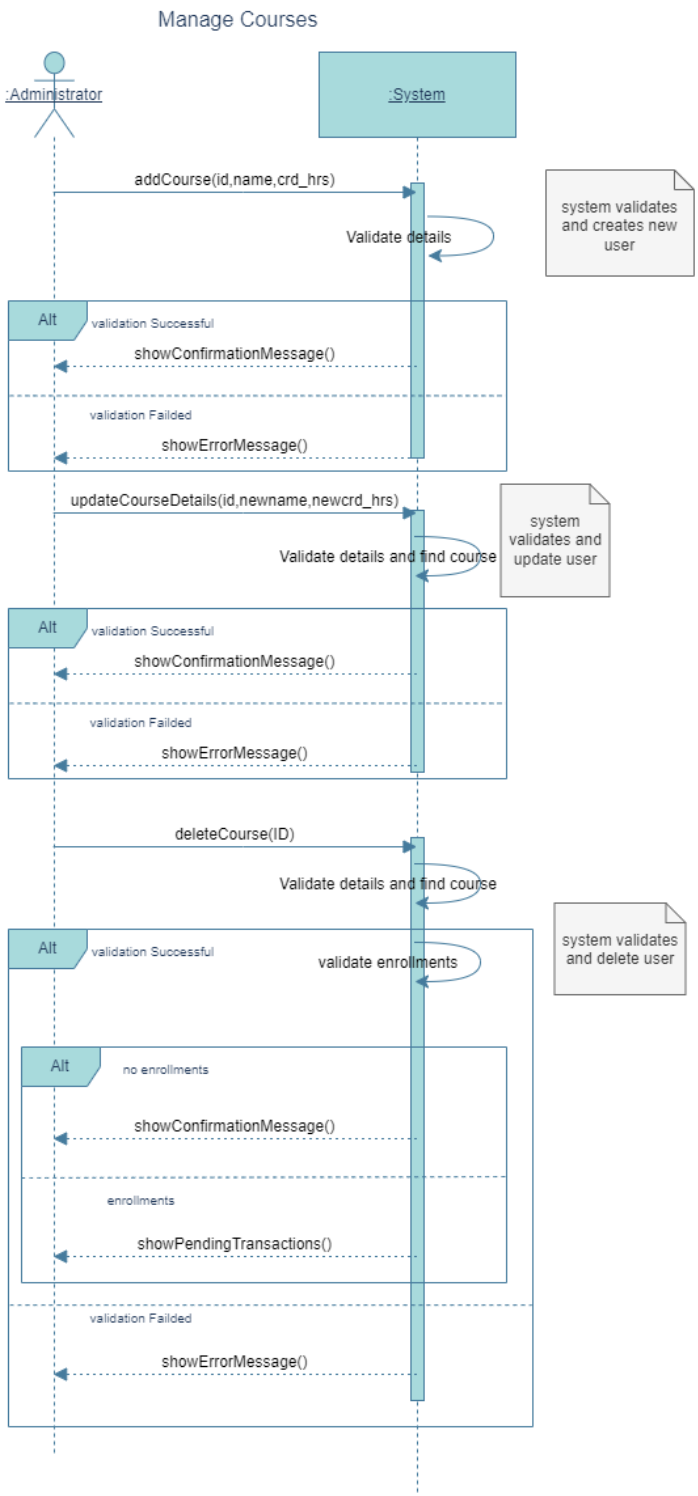


5. System Sequence Diagram

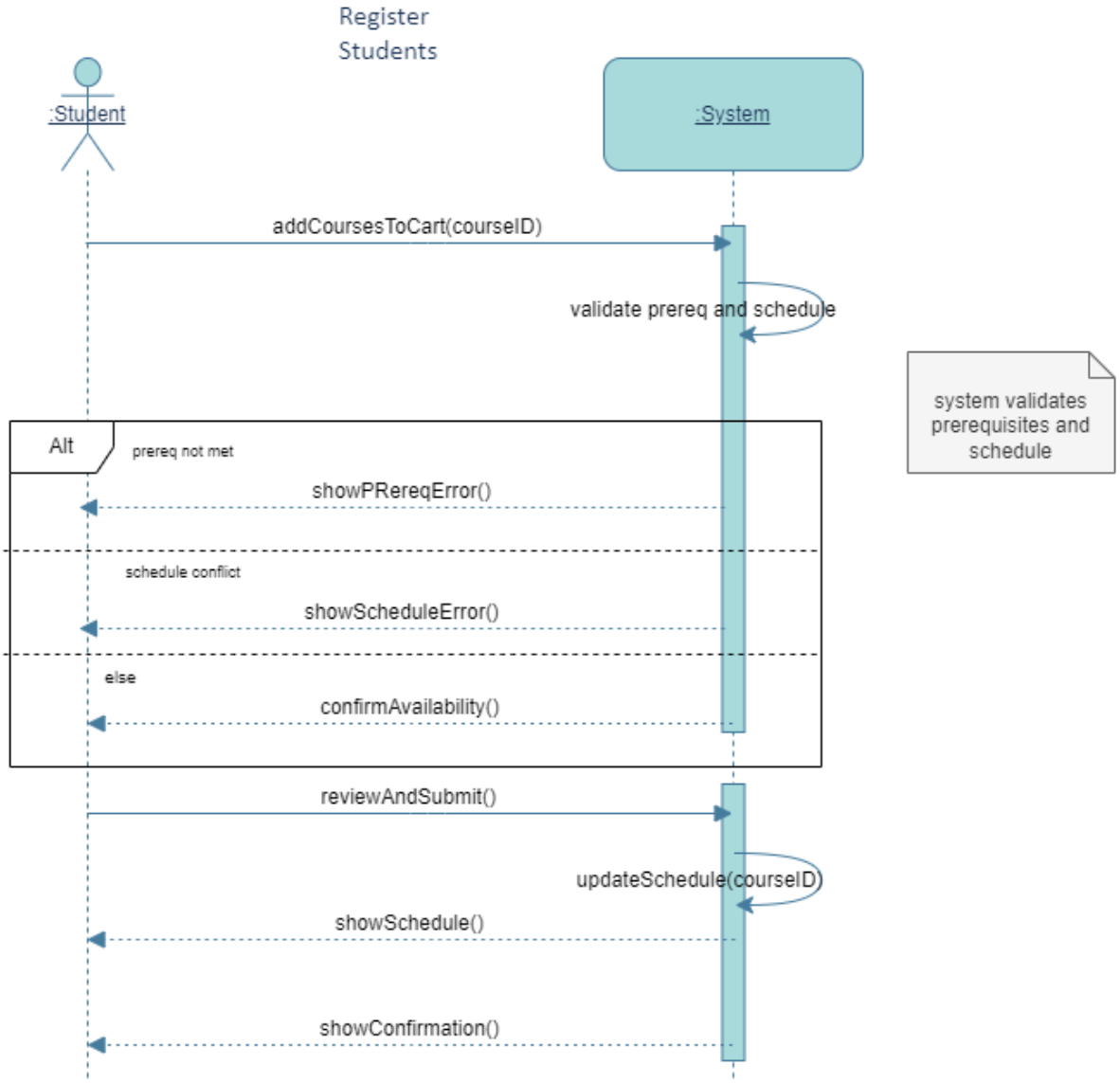
5.1 Manage User Accounts



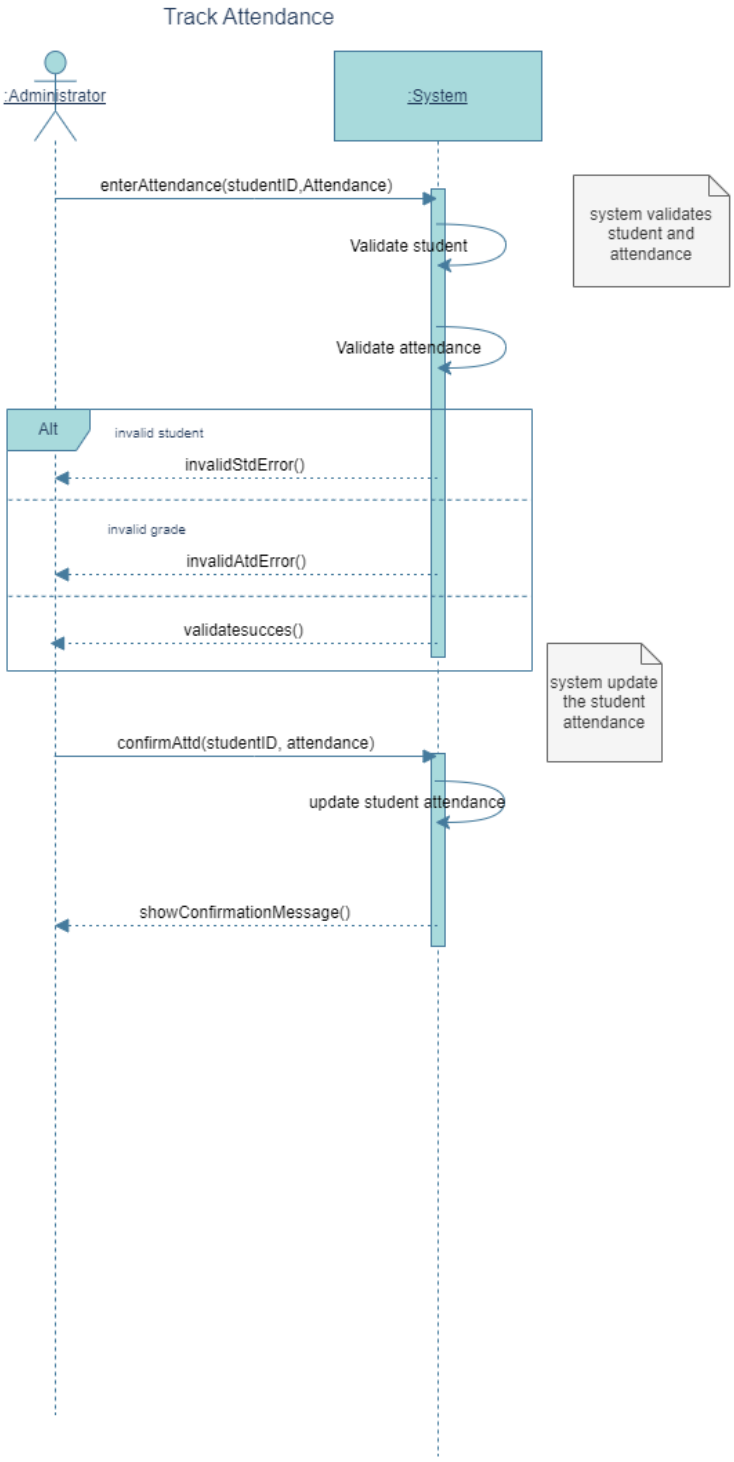
5.2 Manage Courses



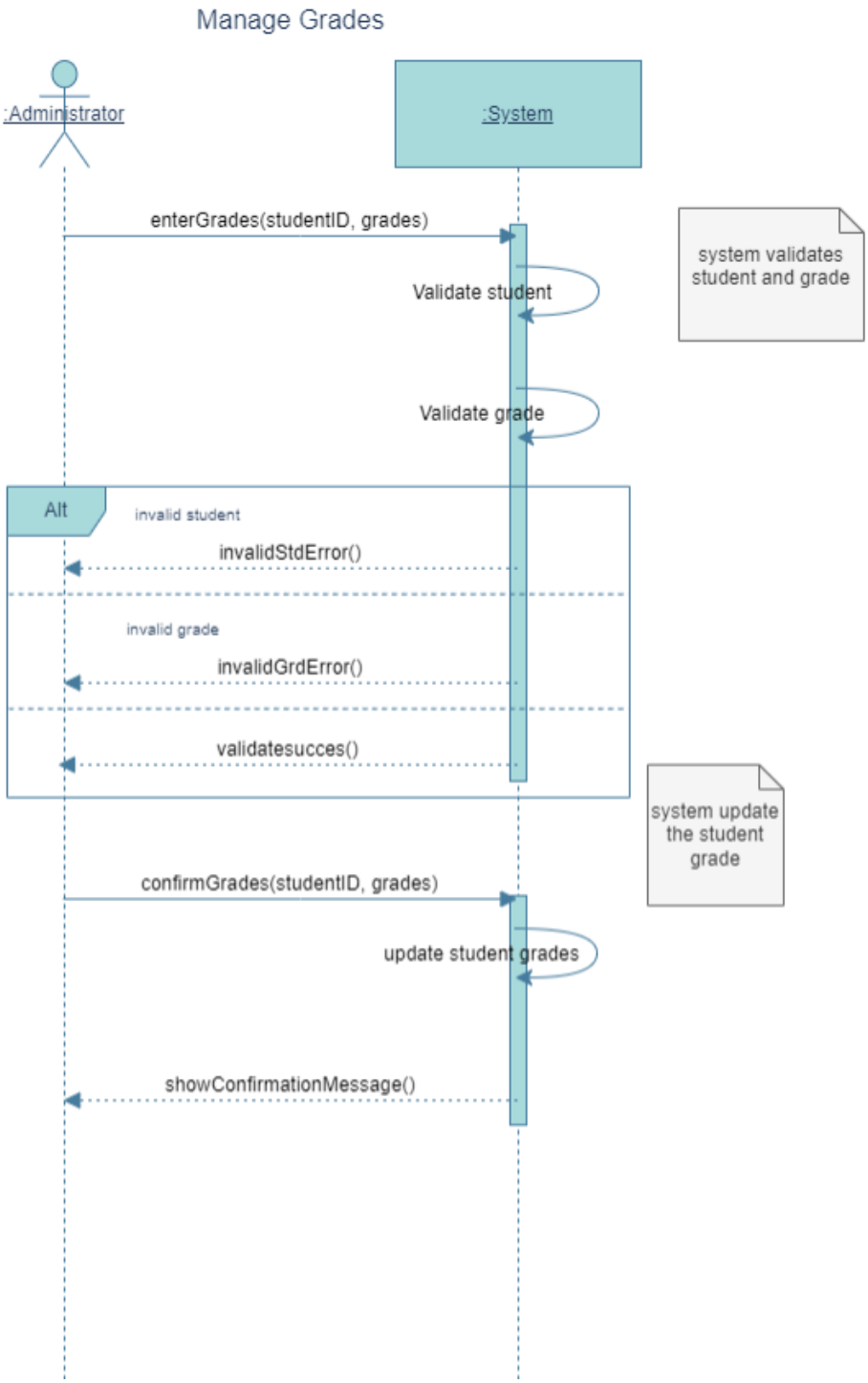
5.3 Register Courses



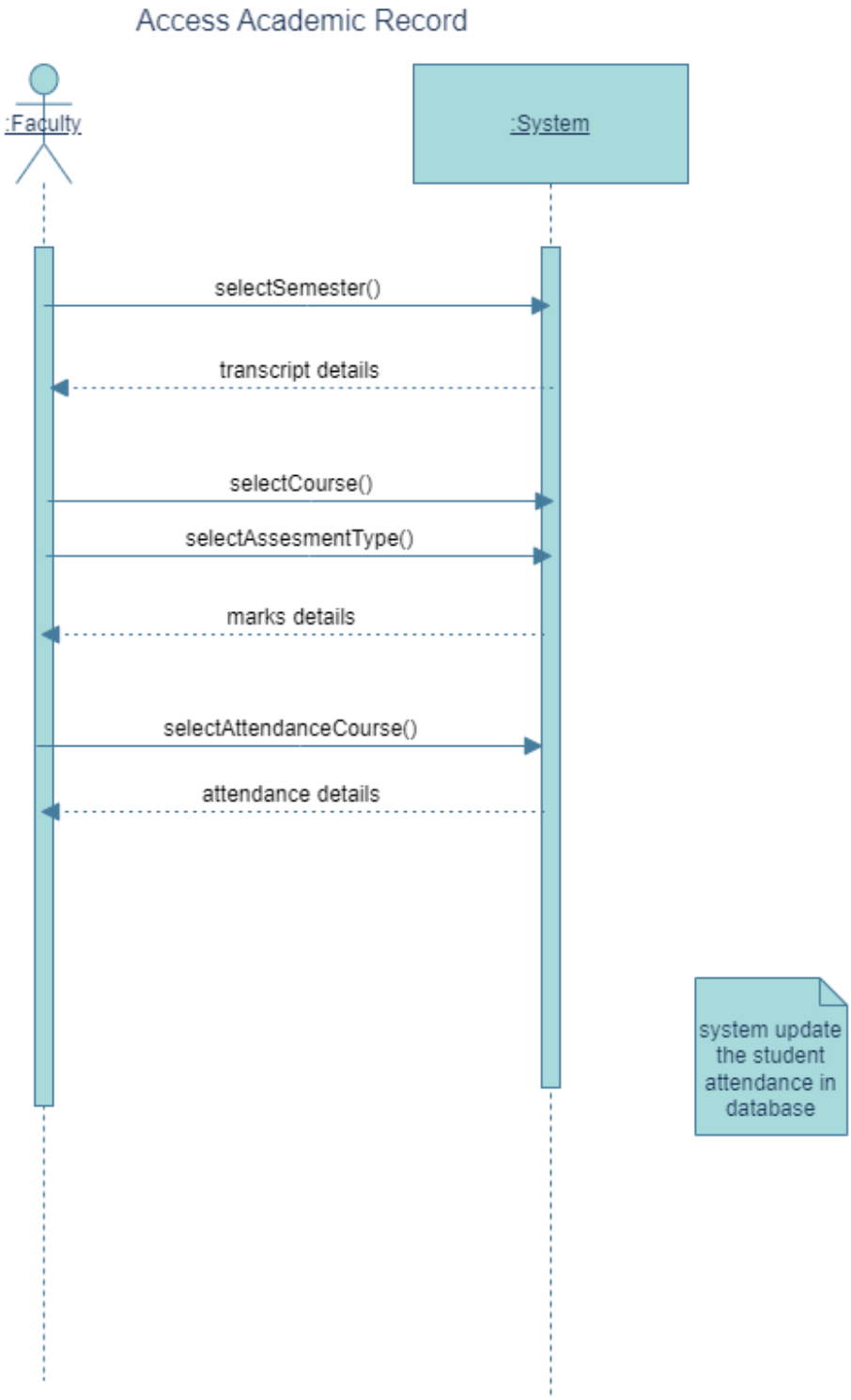
5.4 Track Attendance



5.5 Manage Marks



5.6 Access Academic Record



Extra Imlementaton

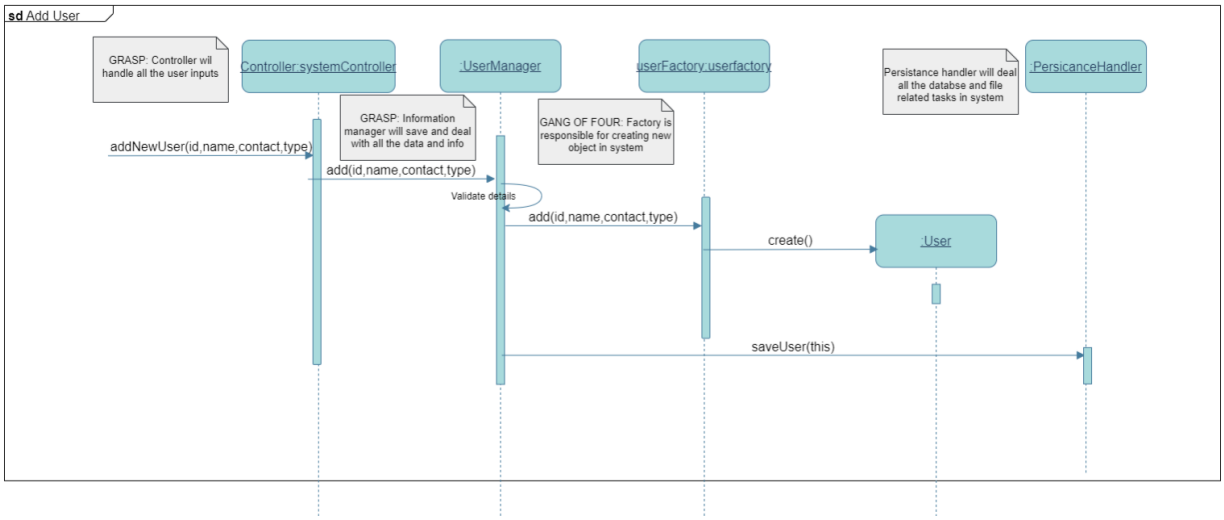
5.7 Access Study Plan

5.8 View Fee Details

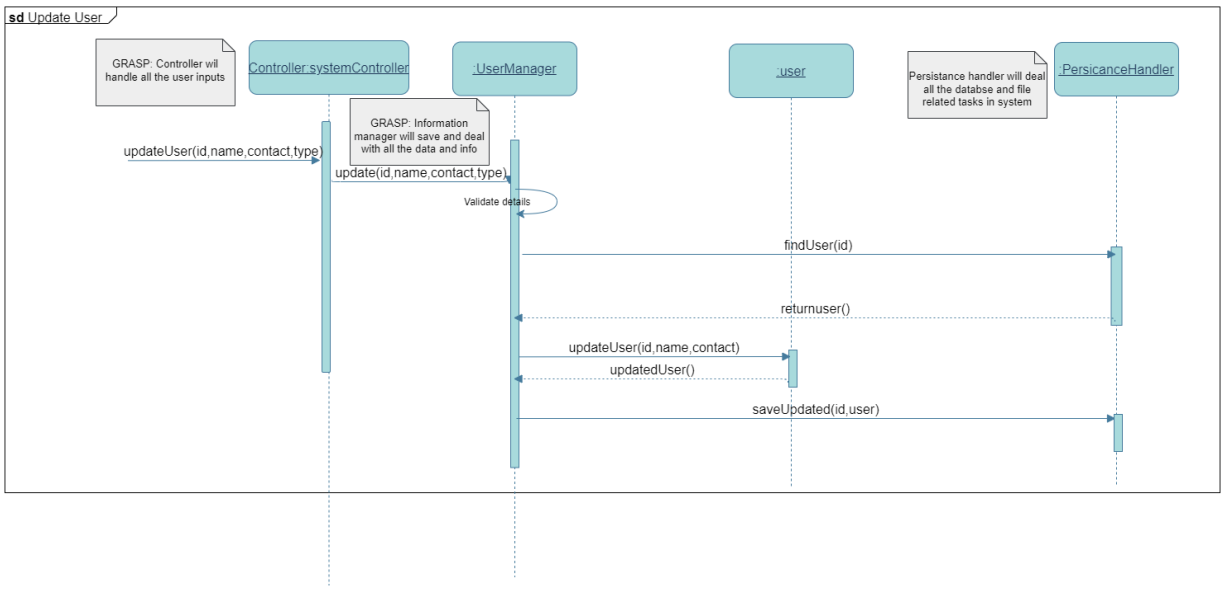
6. Sequence Diagram

6.1 Manage User Accounts

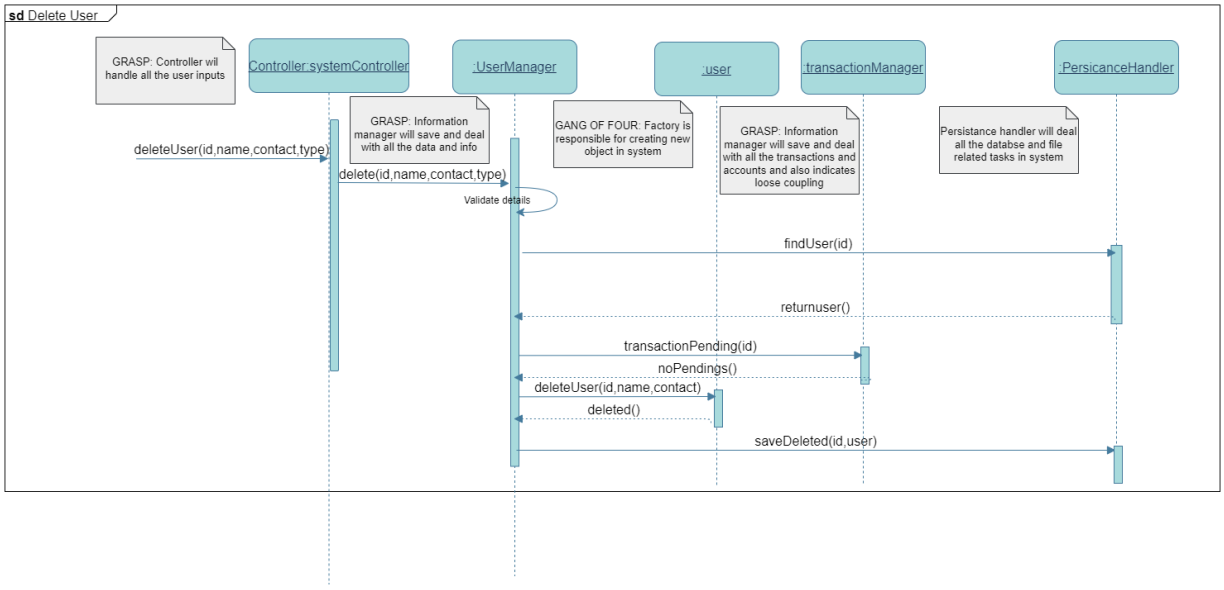
Adding A User Event



Updating A User Event

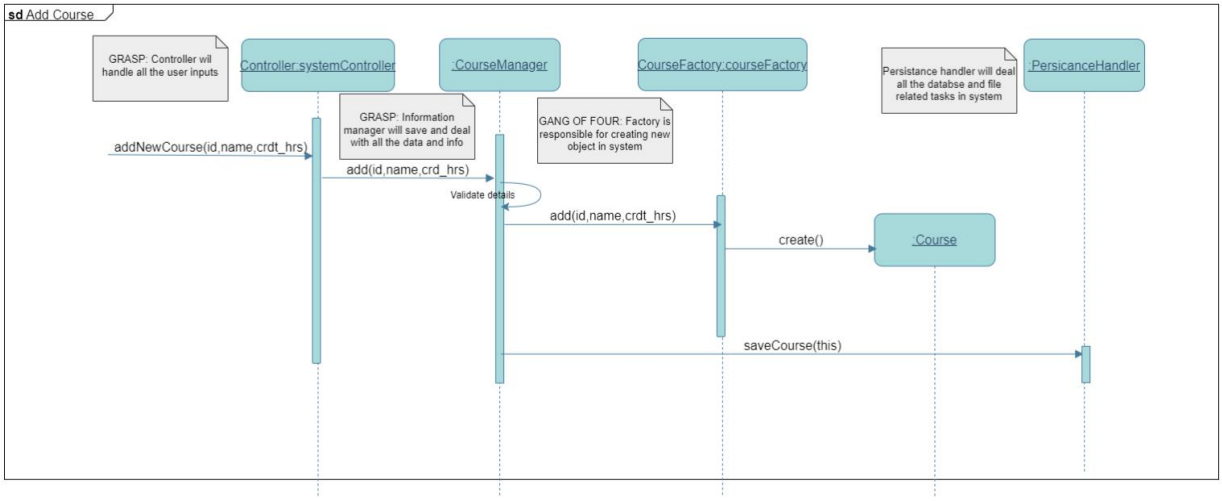


Deleting A User Event

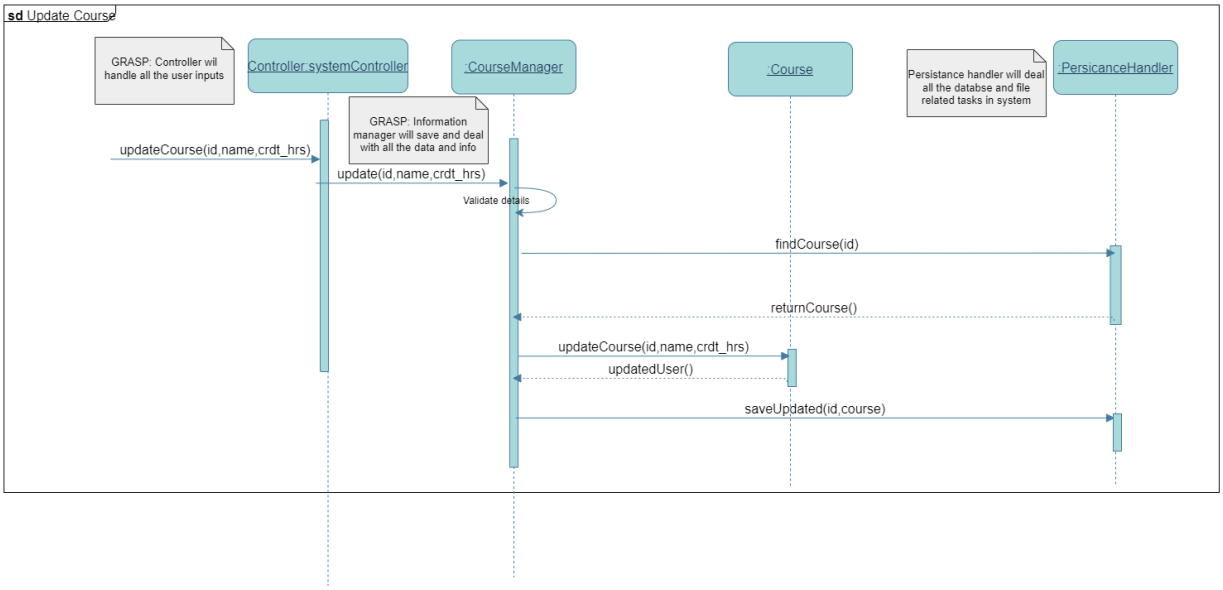


6.2 Manage Courses

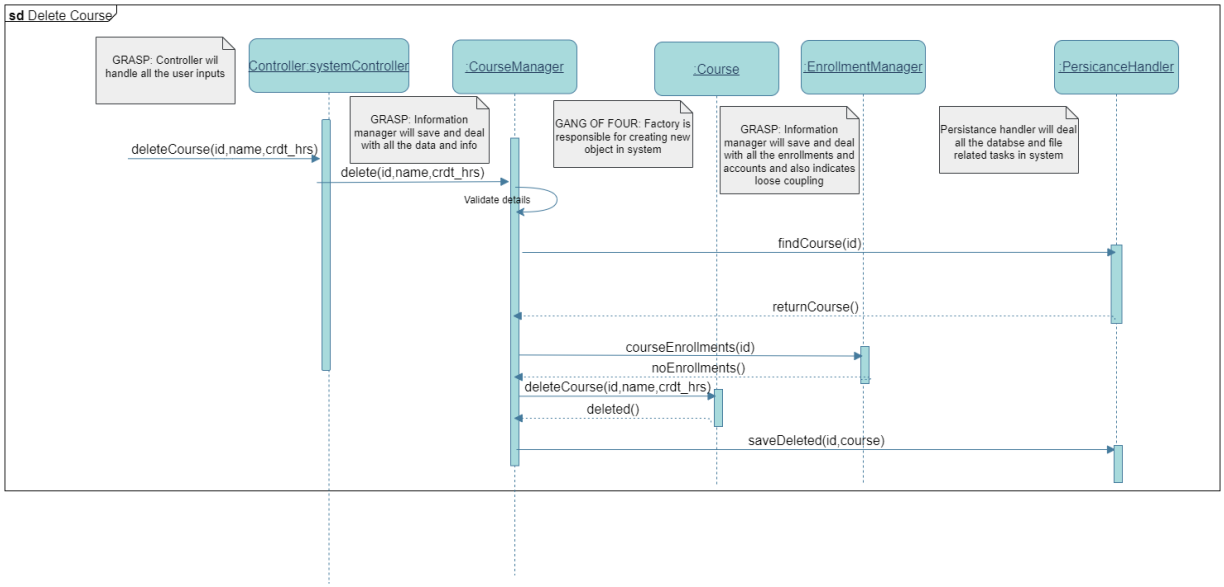
Add New Course Event



Update Course Event

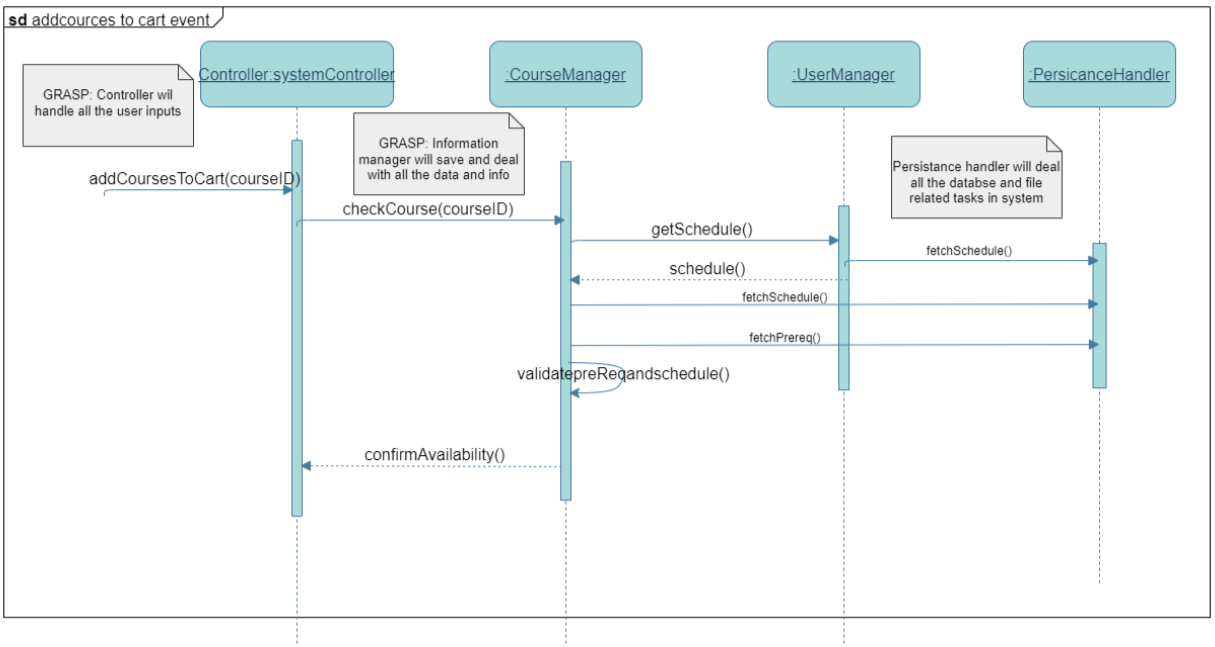


Course Deletion Event

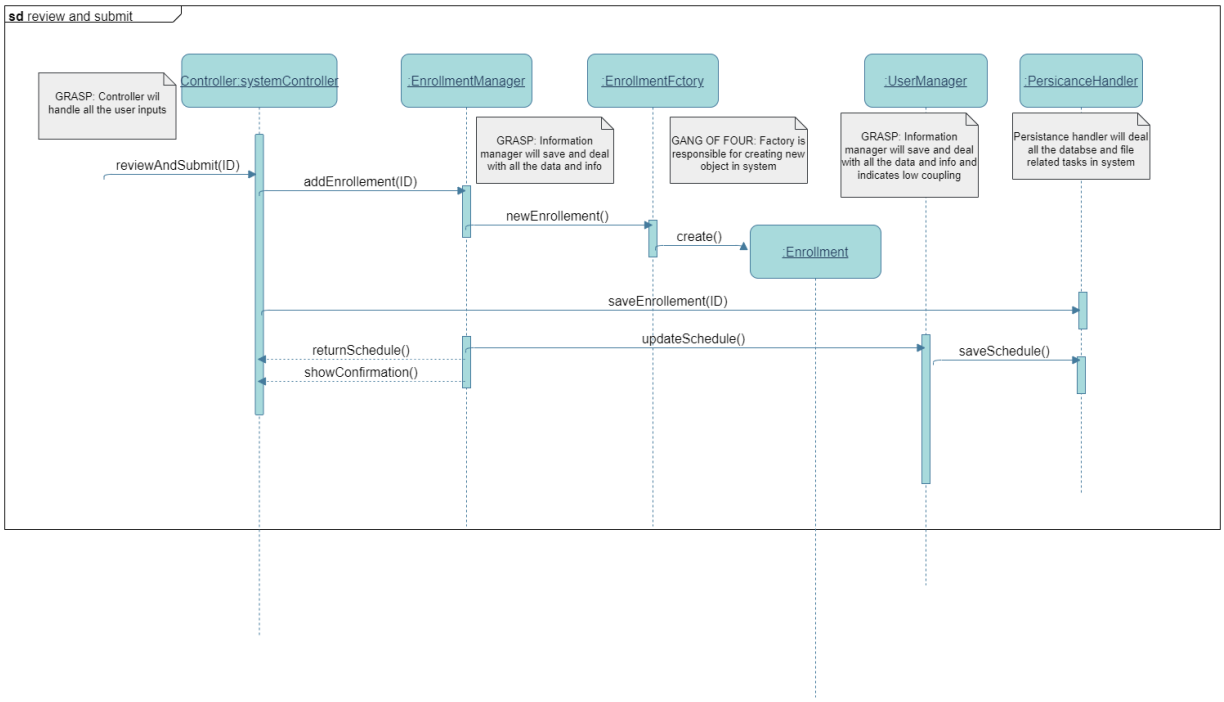


6.3 Register Courses

Add courses to cart event

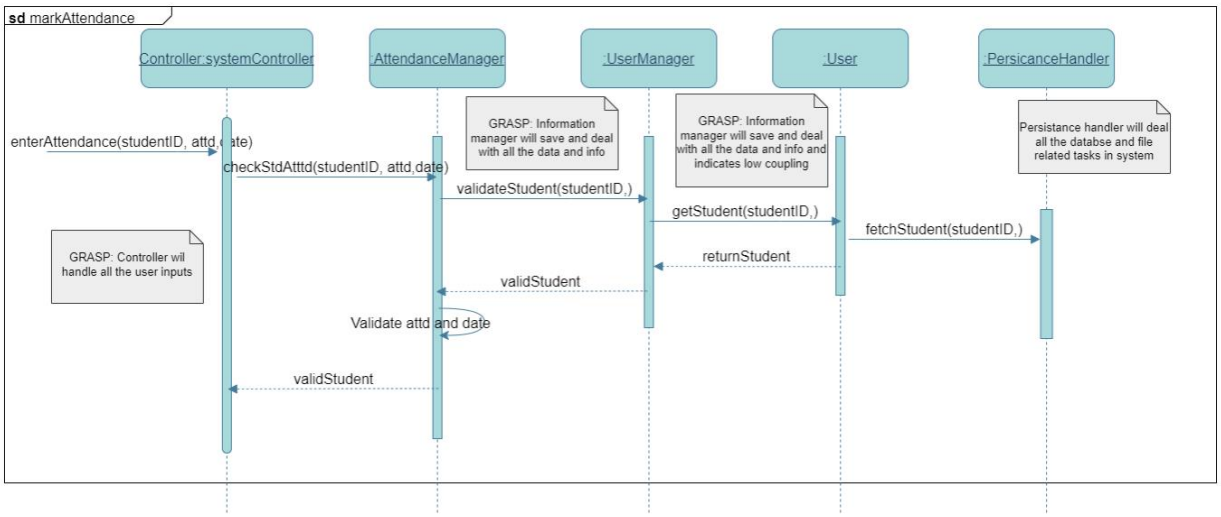


Review and submit event

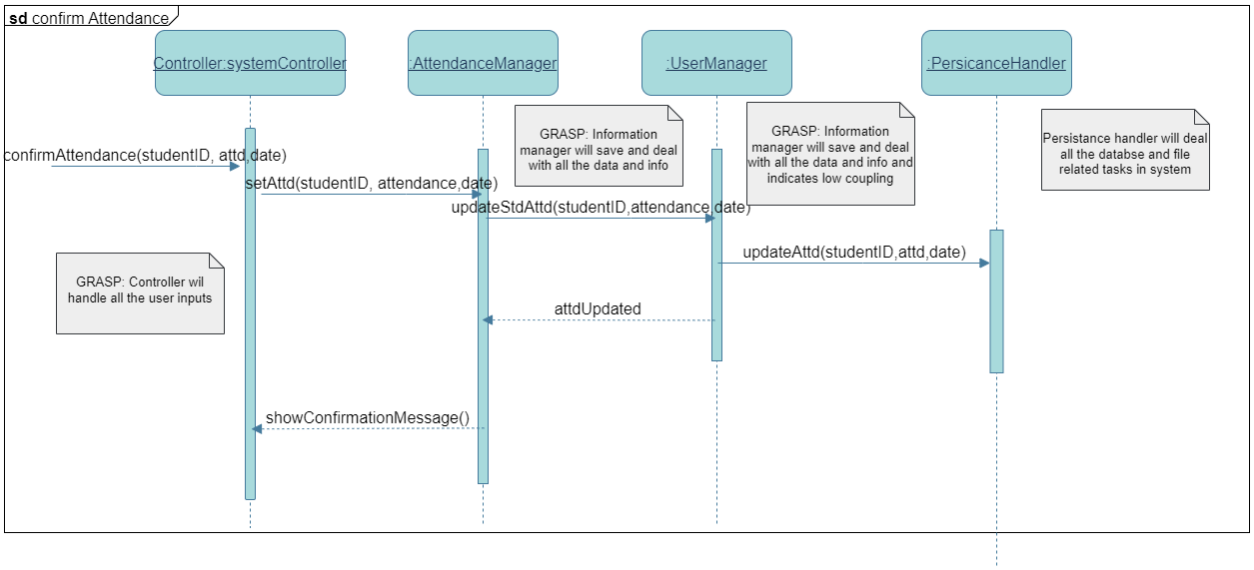


6.4 Track Attendance

Mark Attendance Event

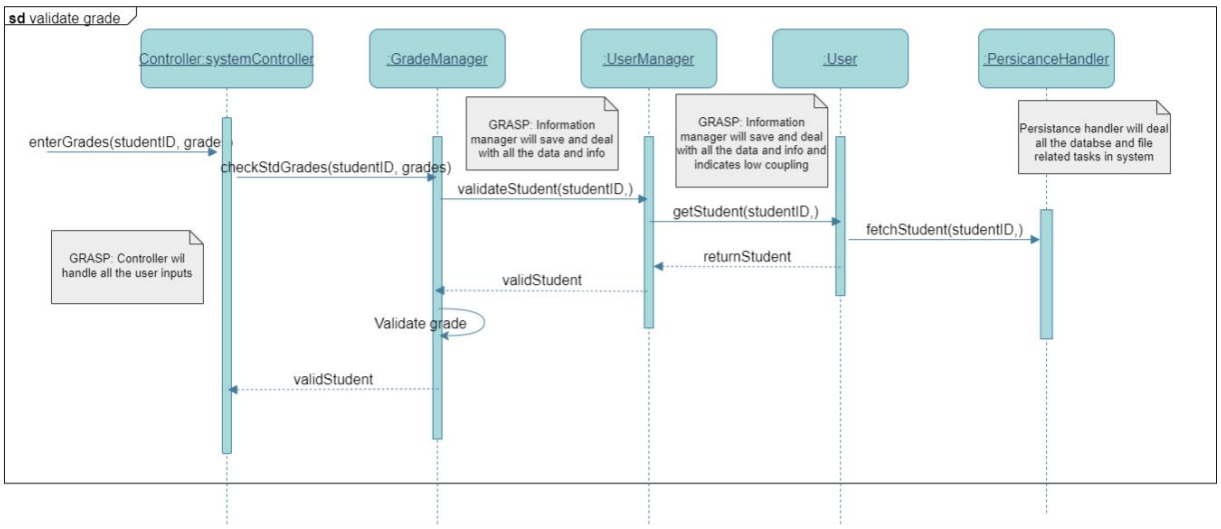


Final Attendance Submission

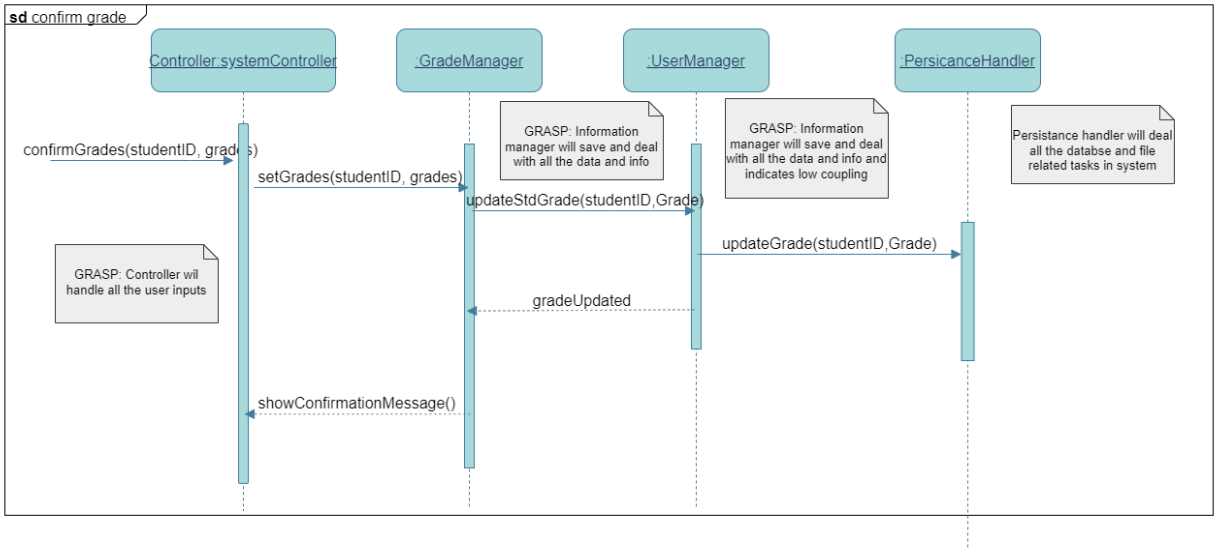


6.5 Manage Marks

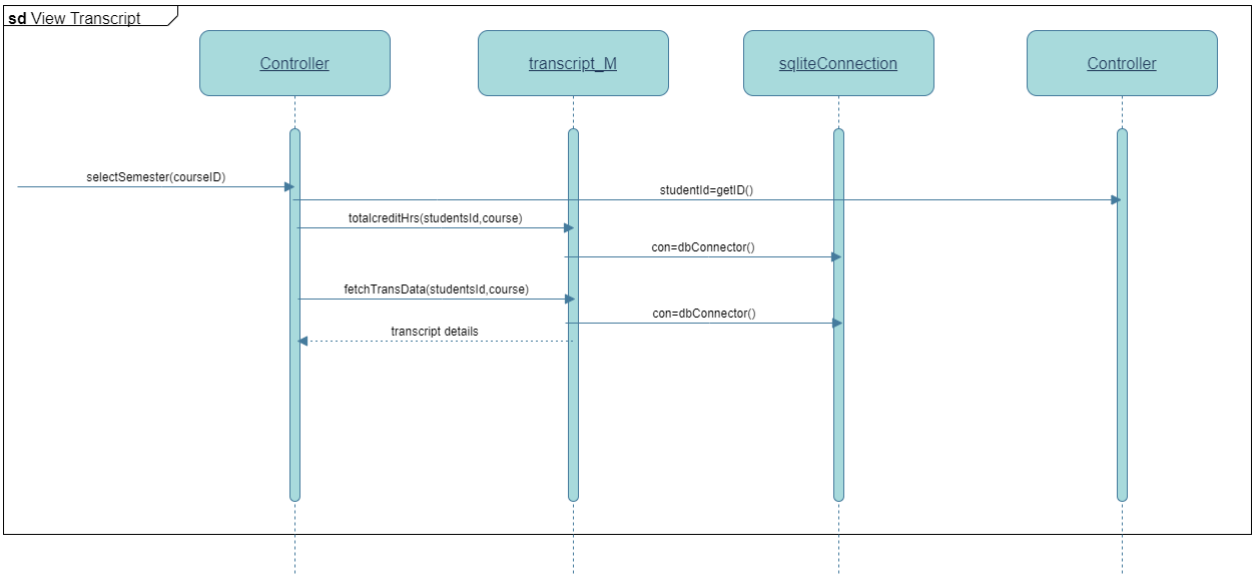
Validate Grade event

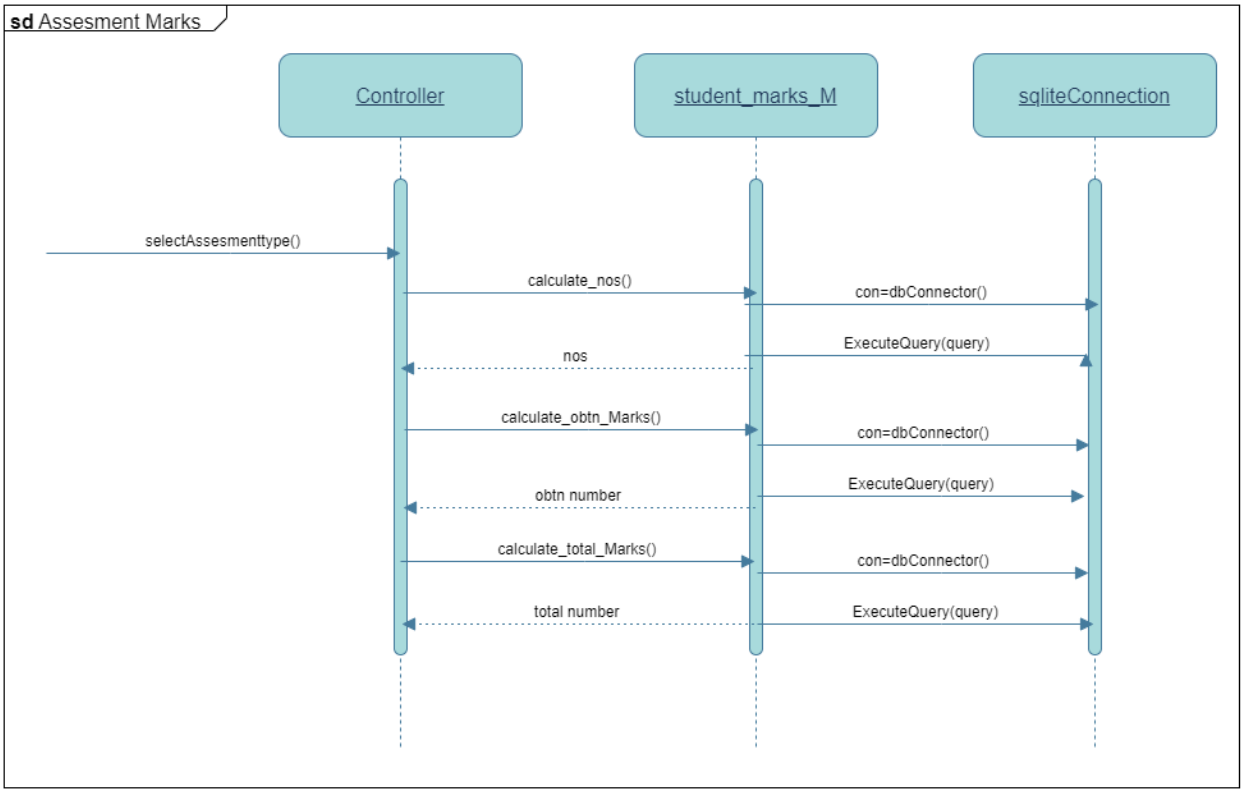


Confirm Grade event



6.6 Access Academic Record



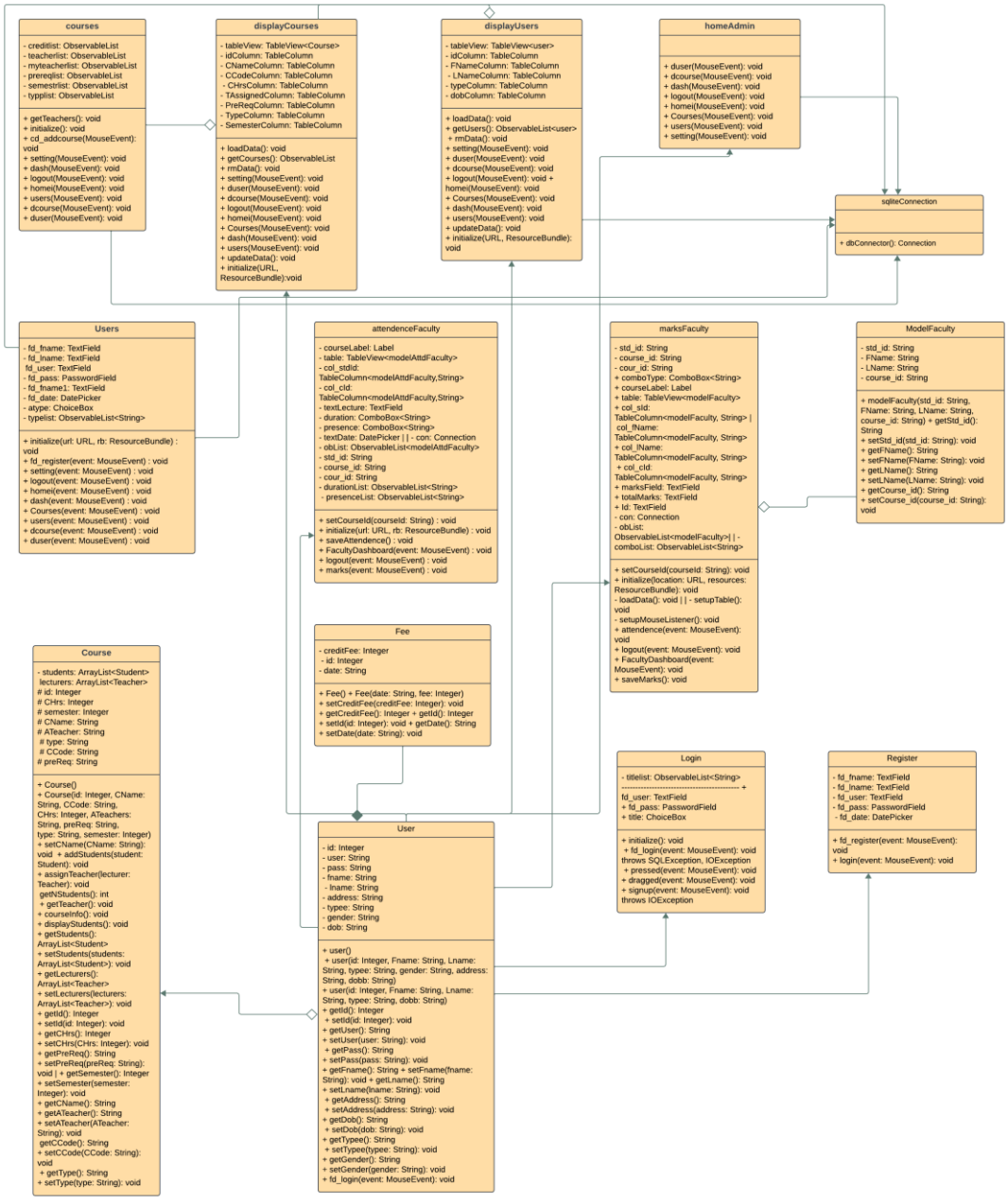


Extra Features Implemented

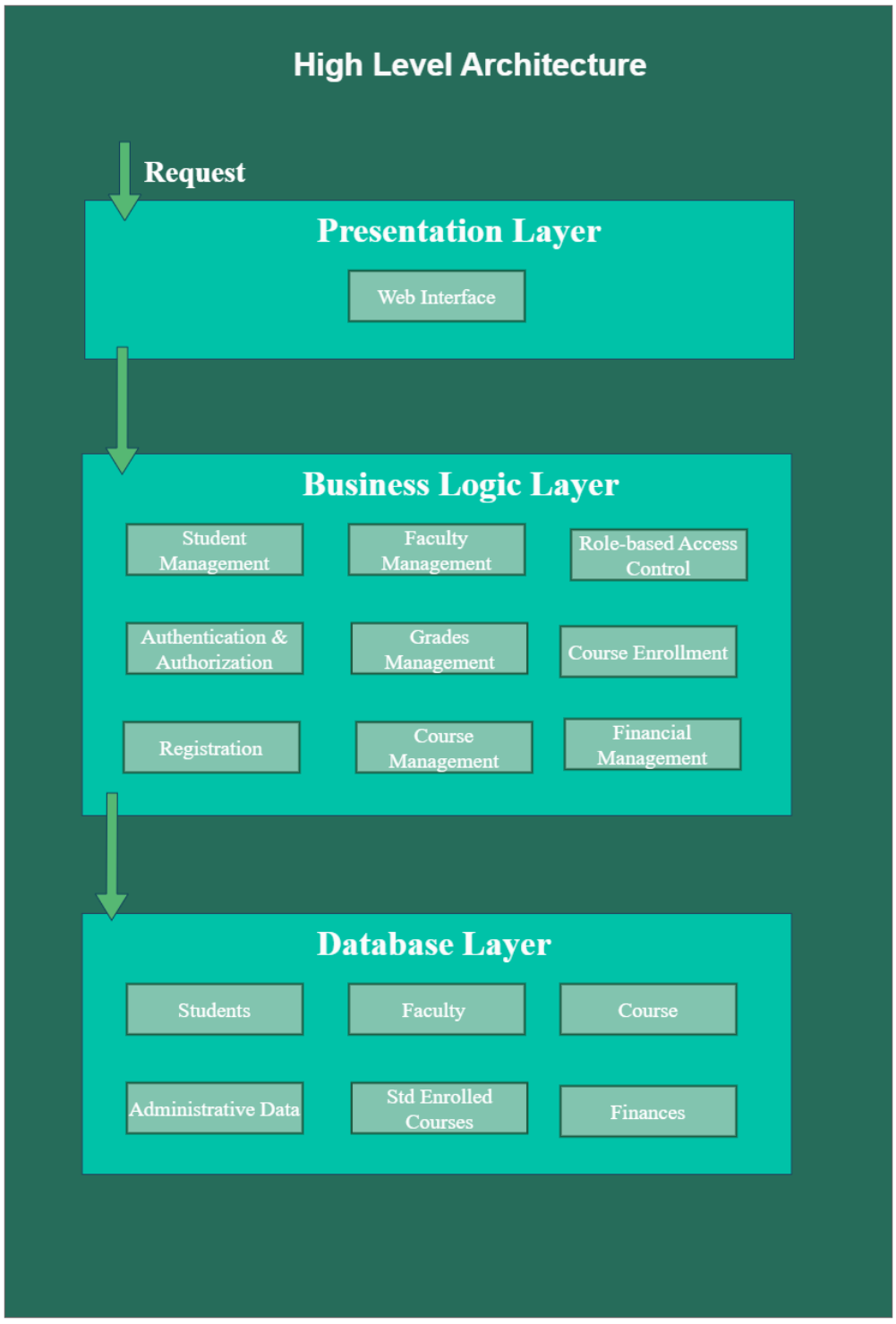
6.7 Access Study Plan

6.8 View Fee Details

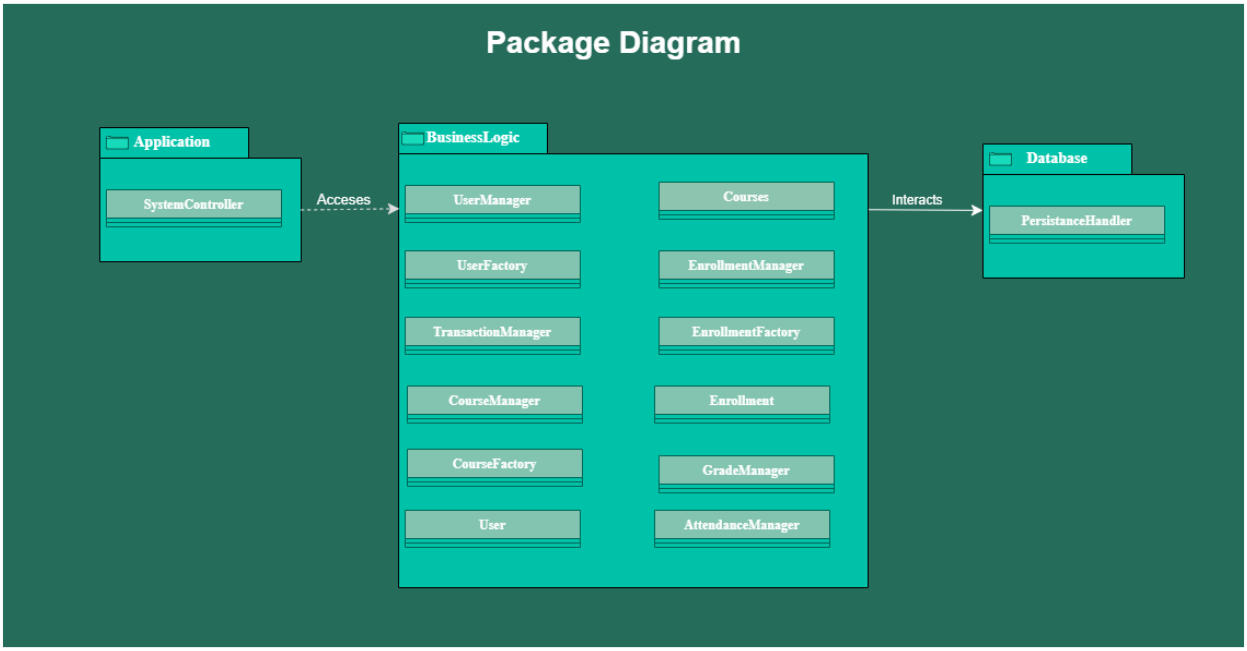
7. Class Diagram



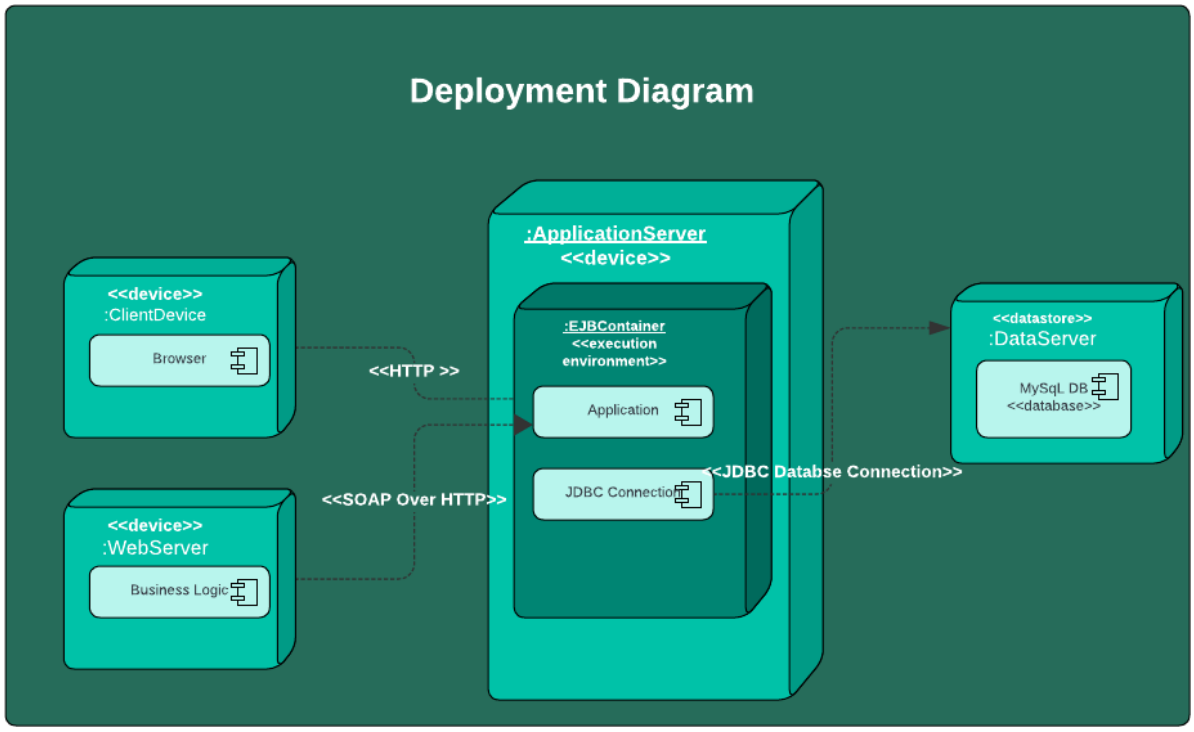
8. High-level Architecture



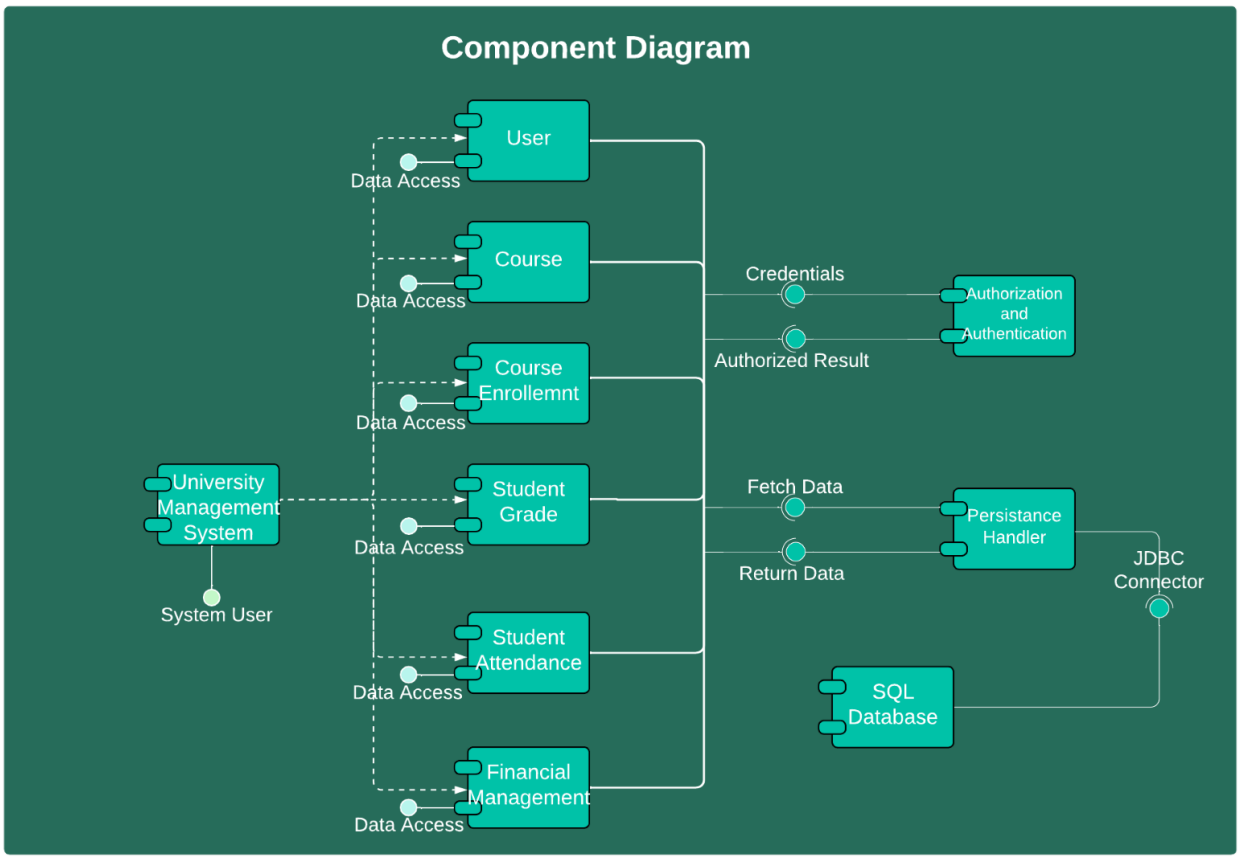
8.1 Package Diagram



8.2 Deployment Diagram



8.3 Component Diagram



9. Work Division Table

S/N	Task	Description	22I2553	22I8759
1	**Project Setup**			
1.1	- Project Initialization	Set up project structure, repository, and dependencies	✅	
1.2	- Database Design	Design and implement the database schema		✅
2	**Admin Subsystem**			
2.1	- User Management	Implement user registration, update, delete functionalities		✅
2.2	- Course Management	Implement course addition, update, delete functionalities		✅
2.3	- Fee Management	Implement fee structure updates	✅	
3	**Student Subsystem**			
3.1	- Course Enrollment	Implement course enrollment functionality		✅
3.2	- View Attendance	Implement functionality to view attendance records	✅	
3.3	- Fee Details	Implement functionality to view fee details	✅	
3.4	- Study Plan	Implement functionality to access and manage study plans	✅	

4	**Faculty Subsystem**			
4.1	- Mark Student Attendance	Implement functionality for marking attendance.	✅	
4.3	- Marks Management	Implement functionality for faculty to update marks of quizzes, assignments, sessional, final and project	✅	
5	**Common Functions**			
5.1	- Authentication/Authorization	Implement login and user role management functionalities	✅	
5.2	- Data Management	Implement data handling, CRUD operations for all subsystems		✅
5.3	- Validations and Error/exception handling	Implement code to validate all the inputs and handle any exceptions	✅	
6	**UI/UX Design**			
6.1	- Admin Interface	Design and implement the user interface for admin functionalities	✅	✅
6.2	- Student Interface	Design and implement the user interface for student functionalities	✅	✅
6.3	- Faculty Interface	Design and implement the user interface for faculty functionalities	✅	
7	**Diagrams**			
7.1	Top level Data flow diagram	Diagram to show the owerall		✅

		working of the system		
7.2	Use case diagram		✅	
7.3	Domain Model			✅
7.4	SSDs	Manage Courses, Manage Users, Course Registration		✅
7.5	SSDs	Track Attendance, Manage marks, View academic Record	✅	
7.6	SDs	Manage Courses, Manage Users, Course Registration		✅
7.7	SDs	Track Attendance, Manage marks, View academic Record	✅	
7.8	Class Diagram	To show the system Classes and interaction	✅	
7.9	High Level Architecture Diagram	To show the high level architecture of system		✅
7.10	Component Diagram	To show the components of system	✅	
7.11	Package Diagram	To show the packages of the system	✅	
7.4	Deployment Diagram	To show the physical view of 4+1 model		✅
8	**Documentation**			
8.1	- SRS Document	Draft and finalize the Software Requirements	✅	

		Specification document		
8.2	- Read me file	Prepare Read me file and guides for different users		✅
8.3	-Liscence File	To provide the liscence to the developers and further users of the system	✅	