
Software Requirements and Design Document

for

<GradeHub>

Prepared by <Abeerah Aamir, Shahzaib Dars, Ali Haider>

<FAST Islamabad>

<25th November,2024>

1. Introduction.....	3
1.1 Purpose.....	3
1.2 Product Scope.....	3
1.3 Title.....	4
1.4 Objectives.....	4
1.5 Problem Statement.....	5
2. Overall Description.....	5
2.1 Product Perspective.....	5
2.2 Product Functions.....	5
2.3 List of Use Cases.....	7
2.4 Extended Use Cases.....	7
3. Other Nonfunctional Requirements.....	36
3.1 Performance Requirements.....	36
3.2 Safety Requirements.....	37
3.3 Security Requirements.....	37
3.4 Software Quality Attributes.....	37
3.5 Business Rules.....	37
3.6 Operating Environment.....	38
• Client System: Windows 10/11.....	38
• Server Environment: MySQL Database with Java runtime environment.....	38
3.7 User Interfaces.....	38
4. Domain Model.....	44
5. System Sequence Diagram.....	45
6. Sequence Diagram.....	52
7. Class Diagram.....	59
8. Deployment Diagram.....	60
9. Package Diagram.....	61
10. Component Diagram.....	61

1. Introduction

1.1 Purpose

The purpose of GradeHub is to provide university students, faculty, and administrators with a centralized platform to manage academic records, attendance, and course-related tasks. The system aims to streamline administrative processes, improve accessibility, and enhance transparency.

1.2 Product Scope

GradeHub is a JavaFX-based desktop application integrated with a MySQL database backend. It allows students to view grades, attendance, register/withdraw from courses, and access fee challans and admit cards. Faculty can manage marks and attendance, while administrators handle user and course management. The platform ensures scalability, security, and ease of use.

1.3 Title

GradeHub: Simplifying Academic Management for Universities

1.4 Objectives

1. User Registration and Role-Based Access:

- Enable students, faculty, and administrators to register and access GradeHub based on their specific roles.
- Ensure secure user verification during the registration process.

2. Attendance and Marks Management:

- Allow faculty to upload, update, and manage attendance records and marks for quizzes, assignments, and exams.
- Provide students with real-time access to view their attendance and marks.

3. Course Registration and Management:

- Enable students to register for courses, withdraw from courses, and view their registered course details.
- Allow administrators to manage courses, including adding, removing, and updating course information.

4. Admit Card and Fee Challan Generation:

- Facilitate students in generating admit cards for exams, ensuring eligibility based on attendance.
- Provide a feature to generate, view, and download fee challans for payments.

5. User and Course Management:

- Allow administrators to add and remove users (students and faculty) with detailed input validation.
- Ensure seamless management of courses, including assigning faculty and updating course information.

6. Performance Analytics:

- Provide students with graphical and tabular representations of their academic performance trends.
- Allow administrators to analyze system-wide usage statistics for better decision-making.

7. Integration and Accessibility:

- Integrate a role-based dashboard to streamline navigation and provide personalized user experiences.
- Ensure accessibility of the platform across devices with an intuitive and responsive design.

1.5 Problem Statement

Educational institutions often face significant challenges in managing academic processes efficiently. Students struggle to track their attendance, grades, and course registrations, while faculty members find it cumbersome to manage attendance and marks for multiple courses. Administrators face difficulties in maintaining user records, managing courses, and ensuring seamless communication across the system. GradeHub addresses these issues by providing a comprehensive academic management platform that streamlines attendance and grade tracking, facilitates course registration and management, and offers essential tools like fee challan and admit card generation. By bridging the gap between students, faculty, and administrators, GradeHub simplifies academic processes, fosters transparency, and enhances the user experience for all stakeholders in the educational ecosystem.

2. Overall Description

2.1 Product Perspective

GradeHub integrates seamlessly into the existing academic framework of universities, providing essential functionalities for students, faculty, and administrators. The platform ensures scalability and supports integration with future modules, like online examination management or alumni tracking.

2.2 Product Functions

- **User Management:**
 - Enable student registration, course enrollment, and secure authentication.
 - Provide faculty and administrator accounts with role-based access.
- **Attendance Management:**
 - Allow faculty to upload attendance records for courses.
 - Enable students to view attendance summaries and details.
- **Grade Tracking:**
 - Faculty can upload grades for quizzes, assignments, and exams.
 - Students can view detailed grade breakdowns for each course.
- **Course Management:**
 - Facilitate course registration and withdrawal for students.
 - Provide administrators with tools to add, update, or remove courses.
- **Fee and Admit Card Management:**
 - Generate and download fee challans in printable format.
 - Allow students to preview and download their exam admit cards based on eligibility criteria.
- **Administrative Oversight:**
 - Enable administrators to add, update, or remove students and faculty.
 - Maintain real-time records of system activities.

- **System Validation:**

- Ensure seamless error handling for invalid input, duplicate entries, or scheduling conflicts.
- Validate attendance and grades for eligibility during admit card issuance.

- **Centralized Reporting:**

- Provide students with access to their academic reports, attendance charts, and grades.
- Allow faculty and administrators to generate system-wide performance metrics.

- **User Experience:**

- Offer an intuitive, centralized platform with role-specific dashboards for students, faculty, and administrators.
- Include streamlined navigation and real-time updates for all users.

2.3 List of Use Cases

- View Grades
- View Attendance
- Register for Courses
- Withdraw from Courses
- Upload Marks
- Upload Attendance
- Add/Remove Students
- Add/Remove Faculty

2.4 Extended Use Cases

Use Case 1: System Validation (Students, Faculty, Admin)

a. **Use Case Name:** System Validation

b. **Scope:** University Data Management System

c. **Level:** System

d. Primary Actor: Faculty, Students, Admin

e. Stakeholders and Interests:

- **Faculty:** Needs accurate marks and attendance records.
- **Students:** Want to ensure correct records.
- **Admin:** Wants to maintain the integrity of the system and resolve any data issues.

f. Preconditions:

1. User (faculty, student, admin) is logged in.
2. System has data to validate.

g. Postconditions:

- User successfully validates data, or system reports validation issues.

h. Main Success Scenario (Two-Column Format)

User Action	System Response
1. User (faculty, student, admin) logs into the system.	1. System authenticates the user and displays the dashboard.
2. User initiates the validation process or system auto-triggers it.	2. System checks the integrity of uploaded data (marks, attendance, etc.).
	3. System validates user credentials (roles: faculty, student, admin).

4. System checks for data inconsistencies or errors.
3. User views the validation results. 6. System displays validation results or error details.
7. If no issues are found, system confirms successful validation.

Use Case 2: Upload Marks (Faculty)

a. Use Case Name: Upload Marks

b. Scope: University Grading System

c. Level: User Goal

d. Primary Actor: Faculty

e. Stakeholders and Interests:

- **Faculty:** Wants to ensure correct submission of student marks.
- **Students:** Need to access their marks.
- **Admin:** Needs proper record maintenance.

f. Preconditions:

1. Faculty is logged in.
2. The course and section are correctly assigned to the faculty.

g. Postconditions:

- Marks are uploaded successfully, and students can view them.

h. Main Success Scenario:

User Action	System Response
1. Faculty logs into the system.	1. System authenticates the faculty and displays the dashboard.
2. Faculty selects the course for which marks are to be uploaded.	2. System loads the course and section details.
3. Faculty selects the assessment (quiz, assignment, exam).	3. System displays the assessment type and marks entry interface.
4. Faculty inputs or uploads the marks (file or manually).	4. System validates the marks format (e.g., numerical values).
5. Faculty confirms the upload.	5. System checks the data and updates the student database with the marks.
6. Faculty submits the changes.	6. System confirms the successful upload and updates student profiles.

i. Extensions:

Condition	Handling
4a. Invalid file format is uploaded.	4a. System displays an error message and requests a valid format.
4b. Marks exceed the valid range.	4b. System alerts the faculty to correct the values.

Use Case 3: Upload Attendance (Faculty)

a. **Use Case Name:** Upload Attendance

b. **Scope:** University Attendance Tracking System

c. **Level:** User Goal

d. **Primary Actor:** Faculty

e. **Stakeholders and Interests:**

- **Faculty:** Wants to maintain accurate attendance records.
- **Students:** Need to track their attendance status.
- **Admin:** Needs accurate record-keeping for compliance.

f. **Preconditions:**

1. Faculty is logged in.
2. Class session has occurred, and attendance is ready to be entered.

g. **Postconditions:**

- Attendance is uploaded successfully, and students can view it.

h. **Main Success Scenario:**

User Action

System Response

1. Faculty logs into the system.

1. System authenticates the faculty and displays the dashboard.

2. Faculty selects the course and session for attendance upload.

2. System displays the session and attendance interface.

- | | |
|--|--|
| 3. Faculty inputs or uploads attendance for the session. | 3. System validates the attendance data (e.g., correct format). |
| 4. Faculty confirms the upload. | 4. System checks and updates the student attendance records. |
| 5. Faculty submits the changes. | 5. System confirms successful upload and updates student profiles. |

i. Extensions

Condition	Handling
3a. Invalid attendance format is uploaded.	3a. System displays an error message and requests correct format.
3b. Attendance exceeds maximum number of students allowed.	3b. System alerts faculty to correct the data.

Use Case 4: View Marks (Student)

a. Use Case Name: View Marks (for quizzes, assignments, and exams)

b. Scope: University Grading System

c. Level: User Goal

d. Primary Actor: Student

e. Stakeholders and Interests:

- **Student:** Wants to access accurate marks for quizzes, assignments, and exams.

- **Faculty:** Wants to ensure students can view marks correctly.
- **Admin:** Wants to maintain a smooth marks management system.

f. Preconditions:

1. The student is logged in.
2. The student has enrolled in courses with available marks for assessments (quizzes, assignments, exams).

g. Postconditions:

- The student can view their marks for all assessments.

h. Main Success Scenario

User Action	System Response
1. Student logs into the system.	1. System authenticates the student and displays the student dashboard.
2. Student navigates to the "Marks" section.	2. System displays the list of enrolled courses for the student.
3. Student selects a specific course.	3. System shows the available assessments (quizzes, assignments, exams) and corresponding marks.
4. Student selects the assessment (e.g., quiz).	4. System displays the marks for the selected assessment.

- | | |
|--|--|
| 5. Student views marks for other assessments (optional). | 5. System shows marks for the requested assessments. |
|--|--|

i. Extensions

Condition	Handling
4a. No marks are available for the selected assessment.	4a. System shows a message indicating marks are not yet available.
2a. Student is not enrolled in any courses.	2a. System displays a message showing no courses found.

Use Case 5: View Attendance (Student)

a. Use Case Name: View Attendance

b. Scope: University Attendance Tracking System

c. Level: User Goal

d. Primary Actor: Student

e. Stakeholders and Interests:

- **Student:** Wants to access accurate attendance records.
- **Faculty:** Wants students to have access to attendance records.
- **Admin:** Wants to ensure proper tracking and viewing of attendance records

f. Preconditions:

1. The student is logged in.
2. Attendance records are available for the courses the student is enrolled in.

g. Postconditions:

- The student can view their attendance records for all courses.

h. Main Success Scenario

User Action	System Response
1. Student logs into the system.	1. System authenticates the student and displays the student dashboard.
2. Student navigates to the "Attendance" section.	2. System displays the list of courses the student is enrolled in.
3. Student selects a course.	3. System shows the attendance records for the selected course (e.g., days attended/absent).
4. Student views detailed attendance (optional).	4. System provides a detailed breakdown of the attendance for individual sessions (e.g., specific dates, marked present/absent).

i. Extensions

Condition	Handling
3a. No attendance data is available for the selected course.	3a. System shows a message indicating that attendance data is not yet available.
2a. Student is not enrolled in any courses.	2a. System displays a message showing no courses found.

Use Case 6: Add Student (Administration)

a. Use Case Name: Add Student

b. Scope: University Student Portal (administration functionality)

c. Level: User-goal (high level)

d. Primary Actor: Administration (Admin staff who manage student enrollments)

e. Stakeholders and Interests:

1. Administration:

Ensures that students are properly added to the system so they can access the portal and receive updates.

2. Students:

Needs their information to be added correctly so they can register for courses, view results, and other portal activities.

3. University:

Ensures records are accurate for institutional management and reporting.

f. Preconditions:

1. The admin must be logged into the system with appropriate permissions to add new students.
2. The student's information (such as personal details, academic background) must be available.

g. Postconditions:

- Success: The student is successfully added to the portal and can now access the system.
- Failure: If there is an error (e.g., incomplete data or validation failure), the student is not added, and an error message is displayed.

h. Main Success Scenario:

1. The admin logs into the student portal.
2. The system validates the admin's credentials and permissions.
3. The admin navigates to the Add Student section.
4. The admin enters all required information (student ID, name, contact details, etc.).
5. The system validates the entered data (e.g., ensuring that required fields are filled).

6. The admin submits the form.
7. The system saves the student's data to the database and confirms successful enrollment.
8. The student is notified via email that their account has been created.

i. Extensions

Condition	Handling
The admin enters incomplete or invalid student information (e.g., missing mandatory fields).	The system flags the errors and prompts the admin to correct the data. The process cannot proceed until valid data is entered.
The system detects a duplicate entry (e.g., the student is already in the system).	The system alerts the admin that the student already exists and prevents adding a duplicate record.
The system experiences a technical issue (e.g., database failure).	The system logs the error and notifies the admin of a temporary failure, requesting them to retry later.

Use Case 7: Remove Teacher (Administration)

a. Use Case Name: Remove Teacher

b. Scope: University Student Portal (administration functionality)

c. Level: User-goal

d. Primary Actor: Administration (Admin staff who manage faculty records)

e. Stakeholders and Interests:

1. Administration:

Needs to ensure that teacher data is up to date and teachers no longer part of the institution are removed.

2. Teachers:

Need their accounts to be removed when they resign or retire to avoid any further association with the institution.

3. University:

Requires an accurate record of active faculty members for management and reporting purposes.

f. Preconditions:

- The admin must be logged into the system with permissions to remove faculty members.
- The teacher to be removed must have an active record in the system.

g. Postconditions:

- **Success:** The teacher's account is successfully removed from the system.
- **Failure:** If the teacher's account cannot be found or there is a system error, the removal process is halted, and an error is reported.

h. Main Success Scenario:

1. The admin logs into the portal.
2. The system validates admin credentials and permissions.
3. The admin navigates to the Remove Teacher section.
4. The admin searches for the teacher by name or ID.
5. The system retrieves the teacher's information.
6. The admin selects the option to remove the teacher.
7. The system validates the action (e.g., confirms whether the teacher has ongoing assignments or courses).
8. The admin confirms the removal.
9. The system deletes the teacher's record from the system and confirms the removal.

i. Extensions**Condition****Handling**

The teacher is currently assigned to courses or has pending grades to submit.	The system alerts the admin that the teacher cannot be removed until their responsibilities are reassigned or fulfilled.
The teacher's ID is not found in the system (e.g., due to a search error or incorrect data).	The system prompts the admin to verify the entered ID or name, suggesting they retry the search.
A network or system error occurs during the removal process.	The system logs the error and notifies the admin of a temporary failure, asking them to try again later.

Use Case 8: Add Teacher

a. Use Case Name: Add Teacher

b. Scope: University Student Portal (Administration functionality)

c. Level: User-goal (high level)

d. Primary Actor: Administration (Admin staff who manage faculty)

e. Stakeholders and Interests:

1. Administration:

Needs to add teacher information accurately to ensure faculty can manage courses and interact with students.

2. **Teachers:** Teachers need to be added to the system so they can be assigned courses and manage their responsibilities.

3. **University:**

The institution requires up-to-date records of all faculty members for proper course assignment, payroll, and reporting.

f. Preconditions:

- The admin is logged in with the necessary permissions to add new faculty members.
- Teacher information (such as name, qualifications, contact information) is available.

g. Postconditions:

- **Success:** The teacher is successfully added to the system and can now log in, be assigned courses, and view relevant information.
- **Failure:** If the teacher's data is incomplete or there is a system error, the addition process fails, and an error message is displayed.

h. Main Success Scenario:

1. The admin logs into the university portal.
2. The system validates the admin's credentials and ensures they have the right permissions to add a new teacher.
3. The admin navigates to the Add Teacher section.
4. The admin inputs all the necessary details (name, department, contact info, qualifications).
5. The system validates the entered information (ensures that required fields are filled, no duplicates, etc.).

6. The admin submits the form.
7. The system saves the new teacher's data to the database.
8. The system confirms the teacher has been added and sends a confirmation email with login details to the teacher.

i. Extensions

Condition	Handling
The admin enters incomplete or incorrect data (e.g., missing mandatory fields, wrong format).	The system displays an error message and prompts the admin to correct the data. The teacher will not be added until all the required information is valid.
The system detects a duplicate entry (i.e., a teacher with the same name or credentials already exists).	The system alerts the admin about the duplicate and prevents the new entry from being created.
The system encounters a technical issue (e.g., database connectivity problem).	The system logs the error and notifies the admin that the operation has failed. It suggests retrying the process later.

Use Case 9: Remove Student (Administration)

a. Use Case Name: Remove Student

b. Scope: University Student Portal (Administration functionality)

c. Level: User-goal (high level)

d. Primary Actor: Administration (Admin staff responsible for managing student records)

e. Stakeholders and Interests:

1. Administration:

Needs to ensure that students who have graduated, dropped out, or been expelled are properly removed from the system.

2. Students:

In case of graduation or voluntary withdrawal, the student needs to have their access and data removed to close their account.

3. University:

The institution must keep accurate records, ensuring that only active students are part of the system and that former students' data is archived or removed.

f. Preconditions:

- The admin is logged into the system with permissions to remove students.
- The student to be removed must have an existing record in the system.

g. Postconditions:

- Success: The student's record is successfully removed, and their access to the portal is terminated.
- Failure: If the student's record cannot be found or there is a system issue, the removal process fails, and an error message is displayed.

h. Main Success Scenario:

1. The admin logs into the portal and selects the Remove Student option.
2. The system validates the admin's credentials and permissions to ensure they have the rights to remove a student.
3. The admin searches for the student by ID or name.
4. The system retrieves the student's information and displays it for confirmation.
5. The admin confirms the removal.
6. The system checks if the student has any pending dues, ongoing courses, or unresolved issues.
7. If there are no pending issues, the system removes the student's profile and logs the action.
8. The system notifies the admin and the student (if applicable) that the account has been successfully removed.

i. Extensions

Condition	Handling
The student has pending dues or ongoing courses.	The system alerts the admin that the student cannot be removed due to unresolved issues. The removal is halted until the dues are cleared, or the courses are completed or transferred.
The student's record cannot be found (e.g., due to a typo or incorrect data input).	
There's a technical issue during the removal process (e.g., system or database failure).	<p>The system prompts the admin to verify the entered details (ID or name) and retry the search.</p> <p>The system logs the error and notifies the admin of the failure. It suggests retrying the process after the issue is resolved.</p>

Use Case 10: Register Courses (Student)

a. Use Case Name: Register Courses

b. Scope: University Course Registration System

c. Level: User Goal

d. Primary Actor: Student

e. Stakeholders and Interests:

- **Faculty:** Need to manage student enrollments in their courses.
- **Students:** Want to register for courses to complete their academic program.
- **Admin:** Needs accurate enrollment records for compliance and scheduling.

f. Preconditions:

1. Students are logged into the system.
2. System has data to validate.

g. Postconditions:

- Courses are successfully registered, and the student can view their updated course list.
- Faculty can view the updated roster of enrolled students.

h. Main Success Scenario (Two-Column Format)

User Action	System Response
1. Student logs into the system.	1. System authenticates the student and displays the dashboard.
2. Student navigates to the "Register Courses" section.	2. System displays available courses based on the student's program and year.
3. Student selects courses to register for.	3. System checks the availability of seats in the selected courses.
4. Student confirms the course selection.	4. System validates the selection (e.g., prerequisites, schedule conflicts).
5. Student submits the registration.	5. System finalizes registration, updates the student's course list, and confirms success.

i. Extensions

Condition	Handling
3a. Course is full.	3a. System displays a message that the course is full and suggests alternatives.
3b. Student has a schedule conflict with the selected course.	3b. System alerts the student about the conflict and requests resolution.
4a. Student does not meet prerequisites for a course.	4a. System displays an error message and disallows registration for that course.
4b. Maximum credit hours exceeded.	4b. System prompts the student to drop a course or reduce credit load.

Use Case 11: Withdraw Course (Student)

a. **Use Case Name:** Withdraw Course

b. **Scope:** University Course Withdrawal System

c. **Level:** User Goal

d. **Primary Actor:** Student

e. **Stakeholders and Interests:**

- **Students:** Want to withdraw from courses they no longer wish to continue.
- **Faculty:** Needs to manage student enrollment changes in their courses.
- **Admin:** Needs accurate records for course withdrawal deadlines and student academic progress.

f. **Preconditions:**

1. Student is logged into the system.

2. The withdrawal period for the course is active.

g. Postconditions:

- Course is successfully withdrawn, and the student can view their updated course list.
- The course is removed from the student's academic load, and faculty sees the updated roster.

h. Main Success Scenario:

User Action	System Response
1. Student logs into the system.	1. System authenticates the student and displays the dashboard.
2. Student navigates to the "My Courses" or "Course Withdrawal" section.	2. System displays the list of currently registered courses.
3. Student selects the course they wish to withdraw from.	3. System confirms eligibility for withdrawal (e.g., within allowed period).
4. Student confirms the course withdrawal.	4. System checks if there are any outstanding issues (e.g., pending grades).
5. Student submits the withdrawal request.	5. System finalizes the withdrawal, updates the student's course list, and confirms success.

i. Extensions:

Condition	Handling
-----------	----------

- | | |
|--|---|
| 3a. Withdrawal period for the course has passed. | 3a. System displays an error message stating the withdrawal deadline has passed. |
| 4a. Pending fees or issues prevent withdrawal. | 4a. System alerts the student and requests resolution of pending issues. |
| 4b. Withdrawal causes the student to drop below minimum credits. | 4b. System warns the student and requests confirmation before proceeding. |
| 4c. Withdrawal limits exceeded. | 4c. System alerts the student if they exceed the allowable number of withdrawals. |

Use Case 12: Print Fee Challan (Student)

- a. **Use Case Name:** Print Fee Challan
- b. **Scope:** University Fee Management System
- c. **Level:** User Goal
- d. **Primary Actor:** Student
- e. **Stakeholders and Interests:**
 - **Students:** Want to access and print fee challans for timely payment.
 - **Admin/Accounts Department:** Needs students to pay fees on time for smooth financial operations.
 - **University:** Ensures all fee payments are processed and recorded accurately.
- f. **Preconditions:**
 - 1. Student is logged into the system.
 - 2. Fee challan is generated and available for the current semester/period.
- g. **Postconditions:**

- The fee challan is successfully generated and printed or downloaded.
- The student can use the fee challan for payment at the designated bank or payment portal.

h. Main Success Scenario:

User Action	System Response
1. Student logs into the system.	1. System authenticates the student and displays the dashboard.
2. Student navigates to the "Fee Challan" section.	2. System displays the available fee challans for the current semester/period.
3. Student selects the relevant fee challan for printing.	3. System retrieves the fee details and displays a preview of the challan.
4. Student clicks "Print" or "Download" option.	4. System generates the fee challan in printable format (e.g., PDF) and prompts the user to print or download it.
5. Student confirms print or download.	5. System confirms the action and logs the download/print for record-keeping.

i. **Extensions:**

Condition	Handling
3a. Fee challan is not yet generated.	3a. System displays an error message indicating that no challan is available.
4a. Student has outstanding issues preventing challan generation.	4a. System displays a message outlining the issues (e.g., pending documents, penalties) and requests resolution.
4b. System is unable to generate challan due to a technical issue.	4b. System displays an error message and requests the student to try again later.

Use Case 13: Print Admit Card (Student)

a. **Use Case Name:** Print Admit Card

b. **Scope:** University Examination System

c. **Level:** User Goal

d. **Primary Actor:** Student

e. **Stakeholders and Interests:**

- **Students:** Want to print their admit cards for final exams to be eligible to sit in exams.
- **Faculty:** Ensures that students meet attendance requirements for each course.
- **Admin/Examination Department:** Needs to ensure that only eligible students (with sufficient attendance) can take the final exams.

f. Preconditions:

1. Student is logged into the system.
2. Final exam dates are announced, and the admit card generation period is open.
3. Student has attendance records for all registered courses.

g. Postconditions:

- Admit card is successfully printed or downloaded if the student meets the attendance criteria.
- Students are allowed to sit for the exams based on the printed admit card.

h. Main Success Scenario:

User Action	System Response
1. Student logs into the system.	1. System authenticates the student and displays the dashboard.
2. Student navigates to the "Admit Card" section.	2. System displays the list of courses registered for the current semester.
3. Student selects "Generate Admit Card" for final exams.	3. System checks the attendance percentage for each registered course.
4. System confirms eligibility (attendance \geq 80%).	4. System displays a preview of the admit card with the eligible courses.
5. Student clicks "Print" or "Download" option.	5. System generates the admit card in printable format (e.g., PDF) and prompts the user to print or download it.

6. Student confirms print or download.

6. System logs the action and marks the admit card as issued for record-keeping.

i. Extensions:

Condition

Handling

3a. Attendance for any course is below 80%.

3a. System displays an alert indicating the courses for which the student is ineligible due to low attendance and removes them from the admit card.

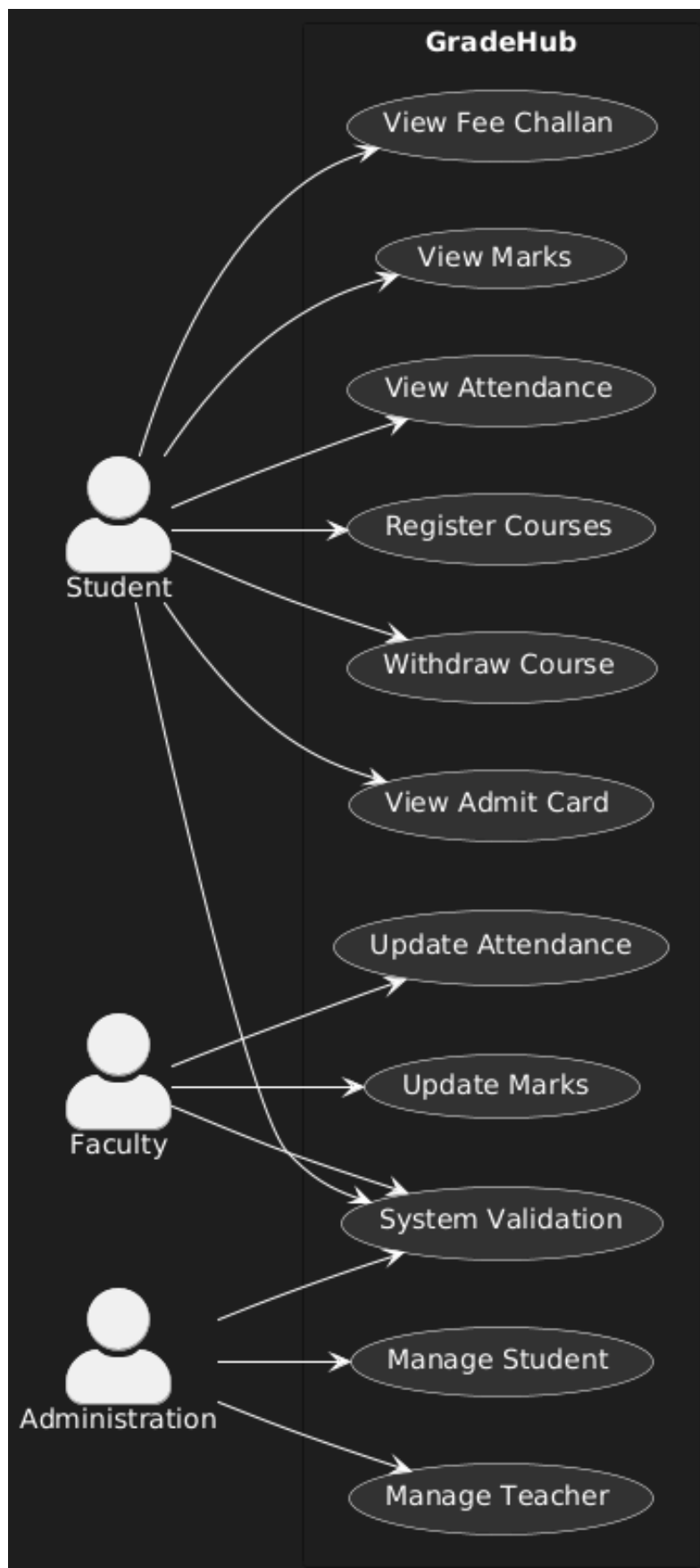
4a. No courses meet the attendance requirement.

4a. System prevents admit card generation and displays a message indicating ineligibility for all courses.

5a. System is unable to generate the admit card due to a technical issue.

5a. System displays an error message and requests the student to try again later.

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

1. UI Performance:

- Ensure a responsive and consistent layout for all screens.
- Maintain adherence to design protocols for smooth navigation and interaction.
- Optimize screen rendering to minimize delays during page transitions.

2. Concurrent Users:

- Support simultaneous access for multiple students, faculty, and administrators without system lag.

3. Module Usage Optimization:

- Specify the active modules to prioritize resource allocation and prevent performance bottlenecks.

4. Database Capacity:

- Ensure sufficient database capacity to handle large volumes of student records, course details, grades, and attendance data.

5. Efficient Data Access:

- Reduce delays in fetching grades, attendance, or fee-related information by optimizing database queries.
- Prevent excessive or redundant database operations to maintain smooth performance.

6. Network Efficiency:

- Minimize network latency by employing optimized communication protocols for accessing and updating data.

3.2 Safety Requirements

- The system must detect unsafe states, prevent unauthorized access, and alert users.

- Limit critical functions to authorized personnel to maintain control and security.
- Provide feedback for user actions to confirm successful processing and ensure accuracy.
- Regularly back up data to prevent loss due to malfunctions.
- Obtain user consent for manual or on-demand backup processes.
- Verify user identity and input details before critical actions to protect database integrity.

3.3 Security Requirements

- Secure login using encrypted credentials.
- Role-based access control to restrict functionalities by user type.

3.4 Software Quality Attributes

- **Usability:** Intuitive interfaces designed with JavaFX.
- **Maintainability:** Modular code for easy updates and bug fixes.
- **Scalability:** Supports future additions like exam management modules.

3.5 Business Rules

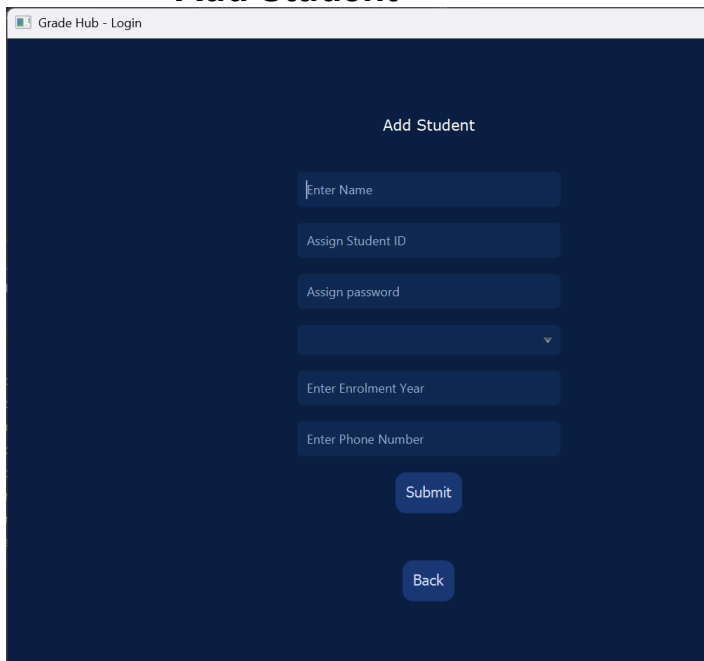
- **Registration Rules:** A student can register for multiple courses but cannot exceed the maximum number of courses allowed per semester.
- **Course Enrollment:** Only courses with available seats can be selected for registration.
- **Eligibility Check:** Registration and withdrawal actions will only be processed if the student meets all academic prerequisites.
- **Course Prerequisite:** A student must complete the prerequisite courses before registering for advanced-level courses.
- **Payment Rules:** Any pending fees must be cleared before finalizing registration or withdrawing from courses.

3.6 Operating Environment

- **Client System:** Windows 10/11.
- **Server Environment:** MySQL Database with Java runtime environment.

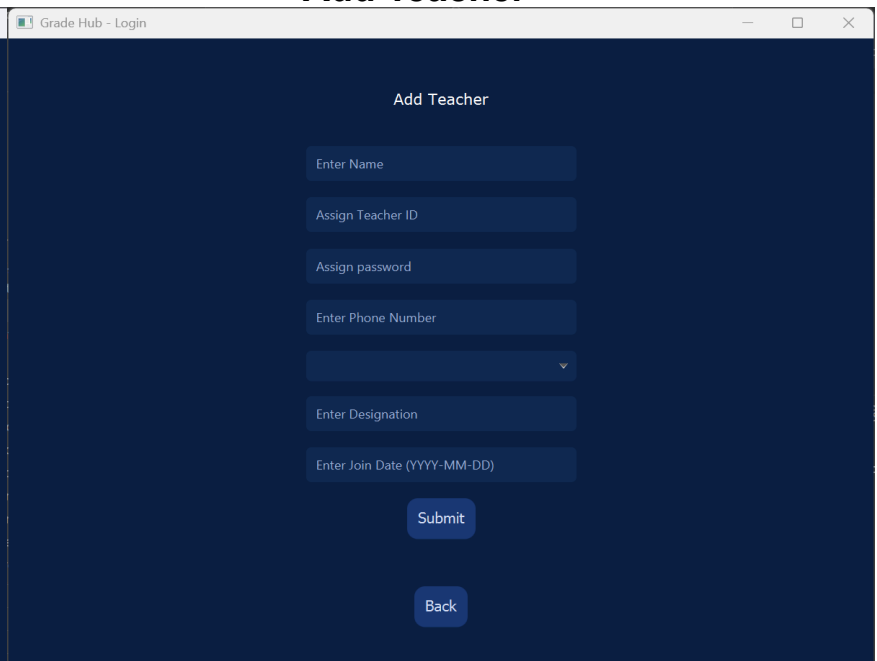
3.7 User Interfaces

Add Student



The 'Add Student' form is displayed in a browser window titled 'Grade Hub - Login'. It features a dark blue background with white text. The form fields are arranged vertically: 'Enter Name', 'Assign Student ID', 'Assign password', a dropdown menu, 'Enter Enrolment Year', and 'Enter Phone Number'. Below the fields are two buttons: 'Submit' and 'Back'.

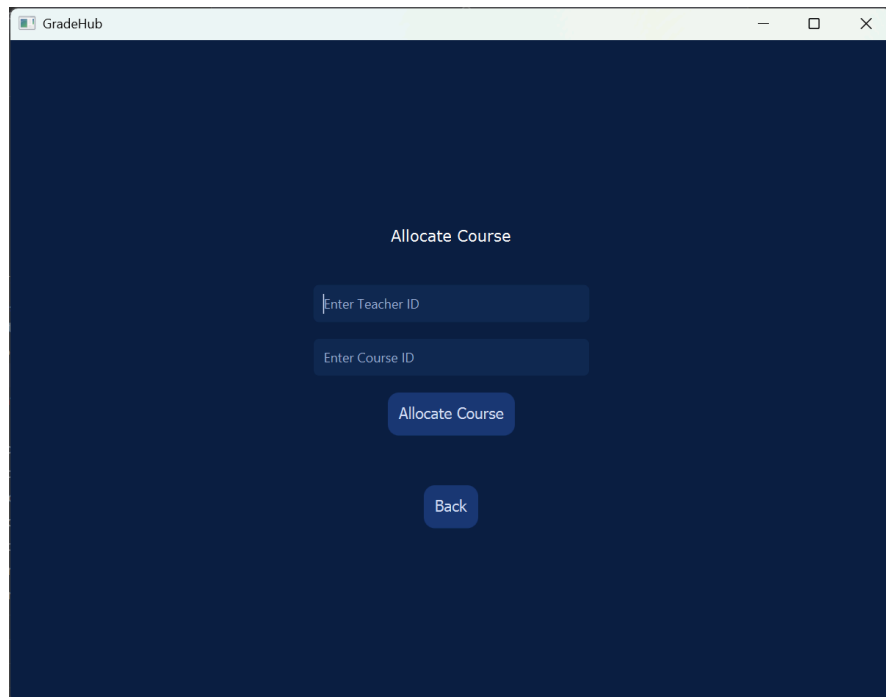
Add Teacher



The 'Add Teacher' form is displayed in a browser window titled 'Grade Hub - Login'. It features a dark blue background with white text. The form fields are arranged vertically: 'Enter Name', 'Assign Teacher ID', 'Assign password', 'Enter Phone Number', a dropdown menu, 'Enter Designation', and 'Enter Join Date (YYYY-MM-DD)'. Below the fields are two buttons: 'Submit' and 'Back'.

Admin enters the teacher/student details into the fields which get saved into the Database

Allocate Courses



A screenshot of a web browser window titled "GradeHub". The page has a dark blue background and is titled "Allocate Course" in white text. It contains two input fields: "Enter Teacher ID" and "Enter Course ID", both with white text and a white cursor. Below the input fields are two buttons: "Allocate Course" and "Back", both with white text on a dark blue background.

GradeHub

Allocate Course

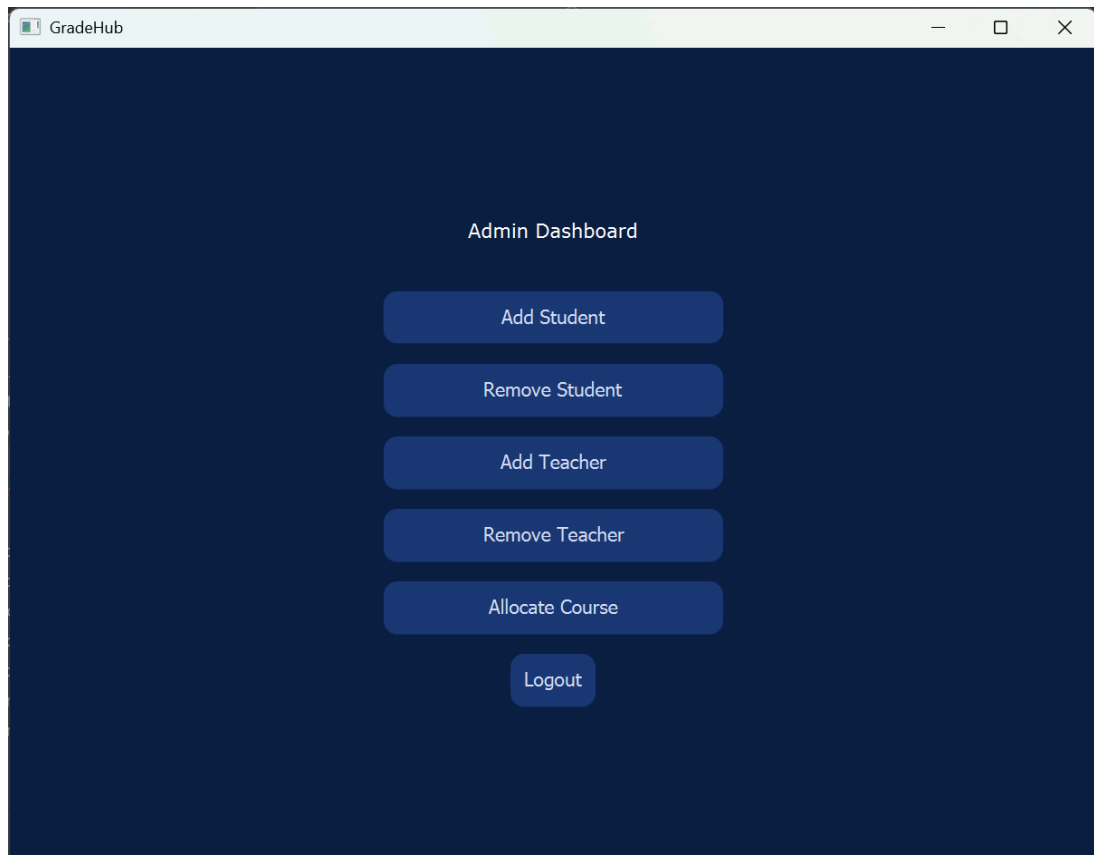
Enter Teacher ID

Enter Course ID

Allocate Course

Back

Admin Dashboard



A screenshot of a web browser window titled "GradeHub". The page has a dark blue background and is titled "Admin Dashboard" in white text. It contains six buttons: "Add Student", "Remove Student", "Add Teacher", "Remove Teacher", "Allocate Course", and "Logout", all with white text on a dark blue background.

GradeHub

Admin Dashboard

Add Student

Remove Student

Add Teacher

Remove Teacher

Allocate Course

Logout

Print Admit Card

The screenshot shows a web browser window titled "Grade Hub - Login". The main content area has a dark blue background. In the center, there is a white rectangular box representing the admit card. Above this box, the text "Admit Card:" is displayed. The box itself contains the following text: "Admit Card - Final Exams", a horizontal dashed line, "Eligible Courses:", "Introduction to Programming", and "Please bring this admit card to your exams." Below the white box, there is a blue button with the text "Back".

Print Fee Challan

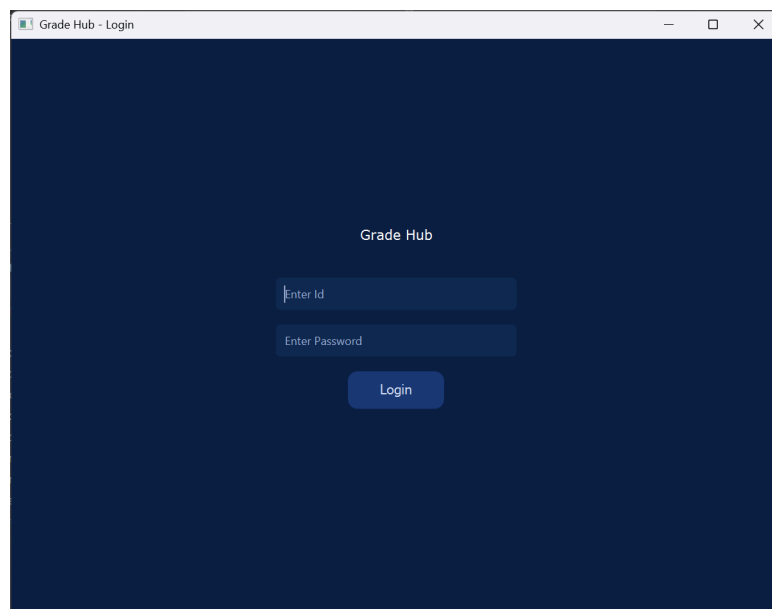
The screenshot shows a web browser window titled "Grade Hub - Login". The main content area has a dark blue background. In the center, there is a white rectangular box representing the fee challan. Above this box, the text "Fee Challan:" is displayed. The box itself contains the following text: "Credit Hours: 11", "Amount: PKR110000", and "Payable at: Faysal Bank". Below the white box, there is a blue button with the text "Back".

Student Admit Card is printed on basis of eligibility and attendance $\geq 80\%$

Faculty Dashboard

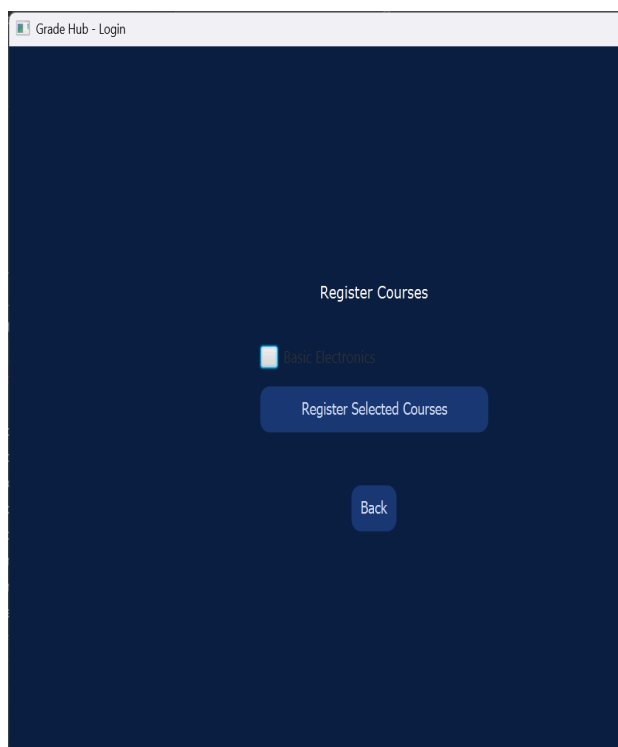
The screenshot shows a web browser window titled "Grade Hub - Login". The main content area has a dark blue background. In the center, the text "Faculty Dashboard" is displayed. Below this text, there are three blue buttons stacked vertically: "Upload Marks", "Upload Attendance", and "Logout".

Login



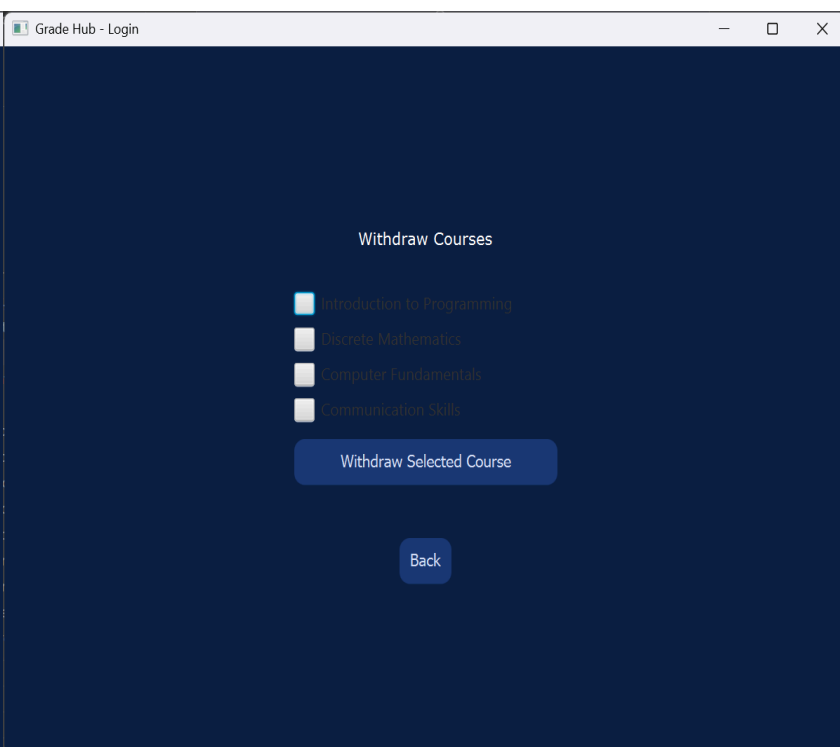
A screenshot of a web browser window titled "Grade Hub - Login". The page has a dark blue background. In the center, the text "Grade Hub" is displayed. Below it, there are two input fields: "Enter Id" and "Enter Password". A blue "Login" button is positioned below the password field.

Register Courses



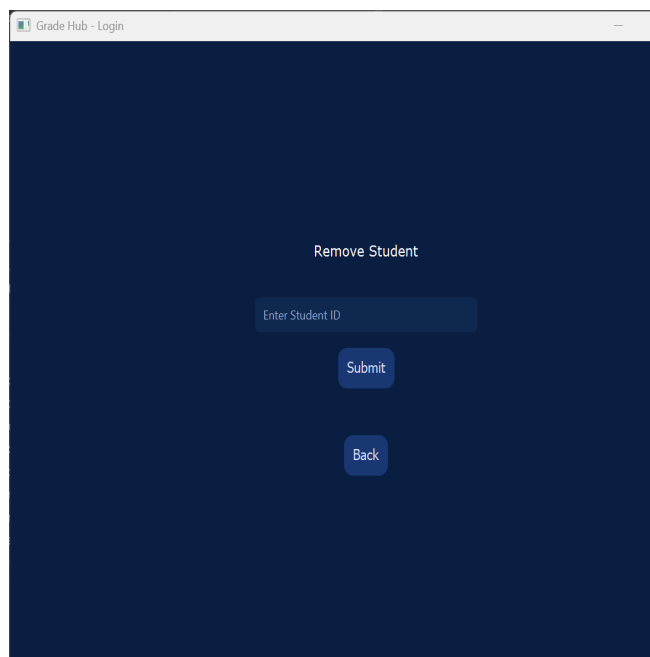
A screenshot of a web browser window titled "Grade Hub - Login". The page has a dark blue background. In the center, the text "Register Courses" is displayed. Below it, there is a list of courses with checkboxes: "Basic Electronics" (checked), "Discrete Mathematics", "Computer Fundamentals", and "Communication Skills". A blue "Register Selected Courses" button is positioned below the list. A blue "Back" button is located at the bottom center.

Withdraw Course



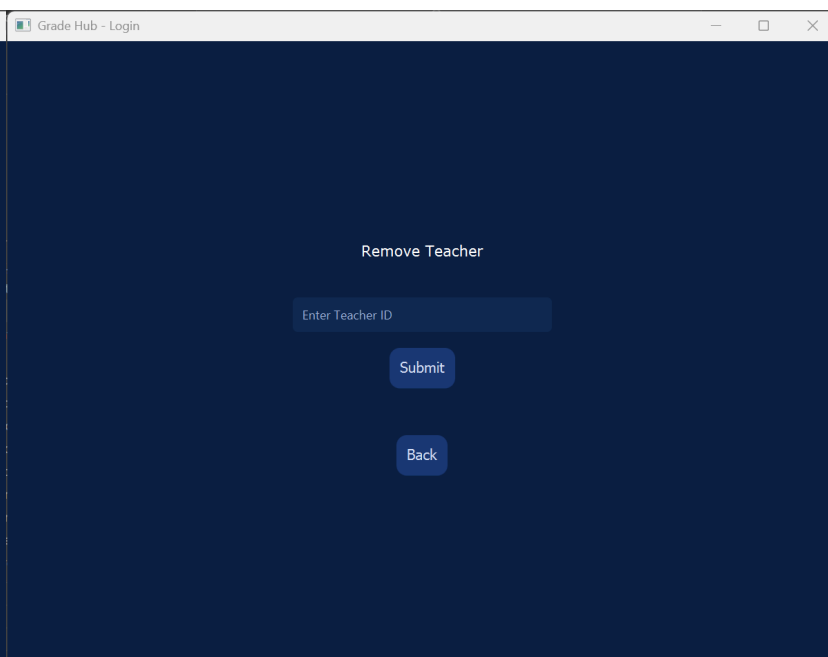
A screenshot of a web browser window titled "Grade Hub - Login". The page has a dark blue background. In the center, the text "Withdraw Courses" is displayed. Below it, there is a list of courses with checkboxes: "Introduction to Programming" (checked), "Discrete Mathematics", "Computer Fundamentals", and "Communication Skills". A blue "Withdraw Selected Course" button is positioned below the list. A blue "Back" button is located at the bottom center.

Remove Student



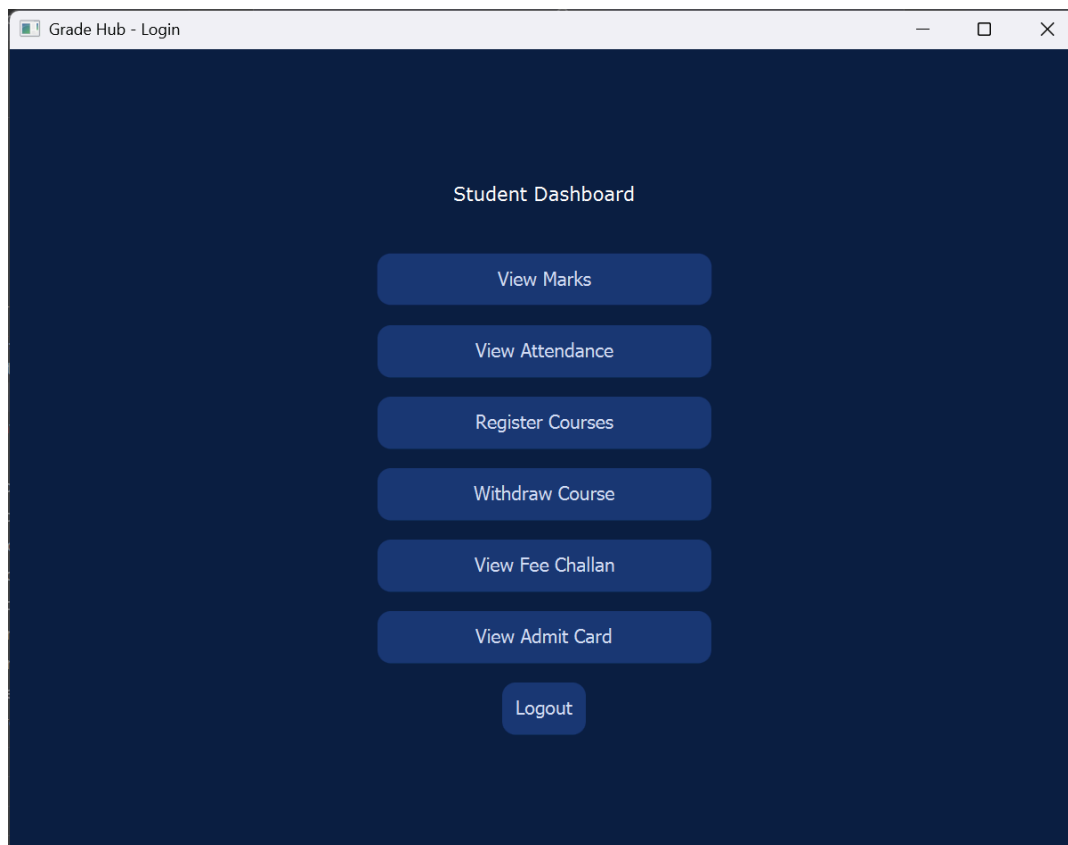
The screenshot shows a web browser window titled "Grade Hub - Login". The main content area has a dark blue background. At the top center, the text "Remove Student" is displayed. Below it is a text input field with the placeholder "Enter Student ID". Underneath the input field are two buttons: "Submit" and "Back", both in a lighter blue color.

Remove Teacher



The screenshot shows a web browser window titled "Grade Hub - Login". The main content area has a dark blue background. At the top center, the text "Remove Teacher" is displayed. Below it is a text input field with the placeholder "Enter Teacher ID". Underneath the input field are two buttons: "Submit" and "Back", both in a lighter blue color.

Student Dashboard



The screenshot shows a web browser window titled "Grade Hub - Login". The main content area has a dark blue background. At the top center, the text "Student Dashboard" is displayed. Below it are seven buttons stacked vertically: "View Marks", "View Attendance", "Register Courses", "Withdraw Course", "View Fee Challan", "View Admit Card", and "Logout". All buttons are in a lighter blue color.

Upload Attendance

Grade Hub - Login

Upload Attendance

Introduction to Programming ▾

Enter Date (yyyy-MM-dd)

Enter Student ID

PRESENT ▾

Submit

Back

Upload Marks

Grade Hub - Login

Upload Marks

Introduction to Programming ▾

QUIZ ▾

Enter Student ID

Enter Marks

Submit

Back

View Attendance

Grade Hub - Login

View Attendance

Introduction to Programming ▾

View Attendance

Attendance: 100.00000%

Date: 2024-03-02 - PRESENT

Back

View Marks

Grade Hub - Login

View Marks

Introduction to Programming ▾

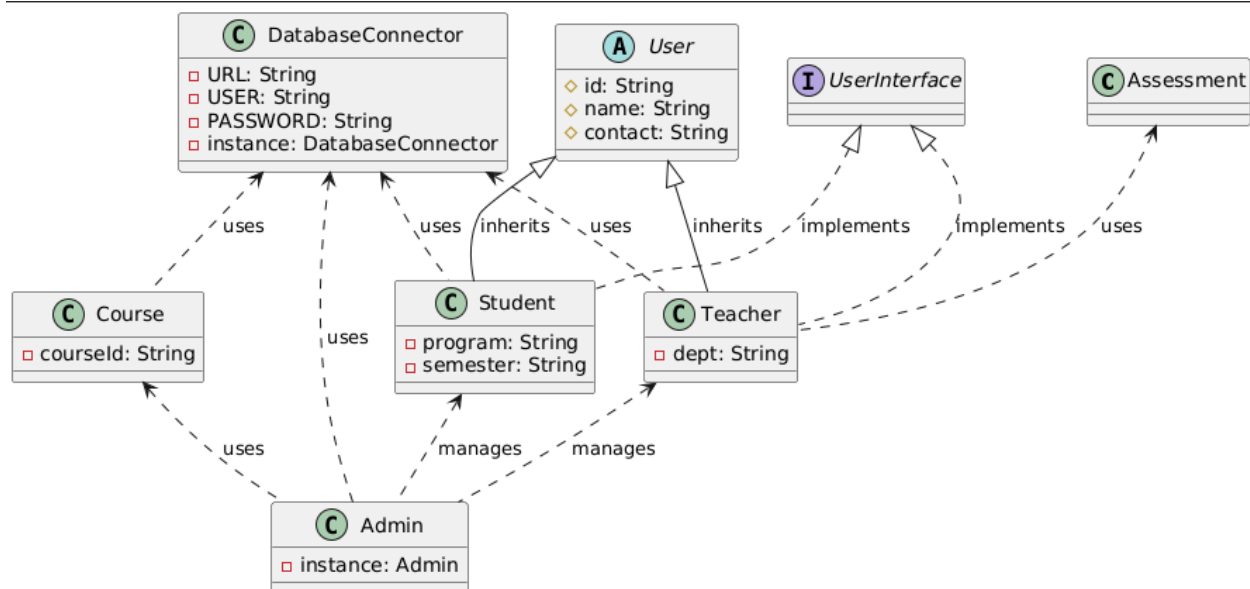
QUIZ ▾

View Marks

Marks for QUIZ in Introduction to Programming: 12.0/20.0

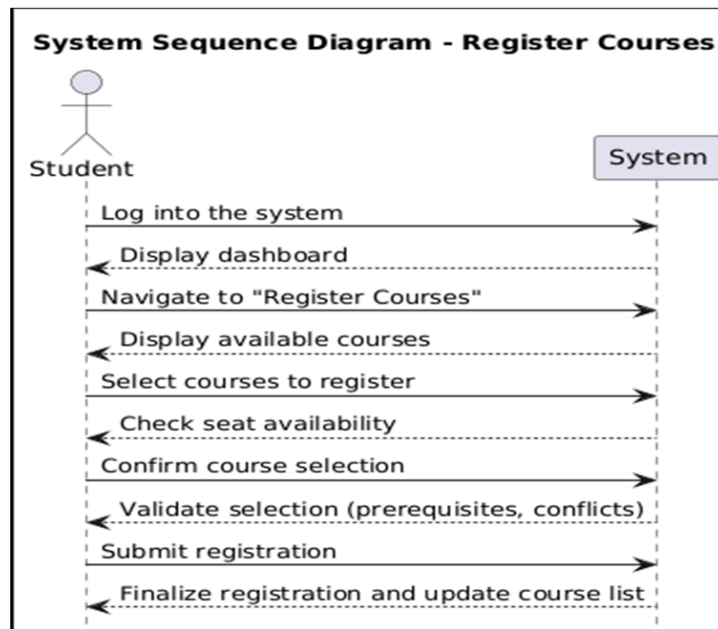
Back

4. Domain Model



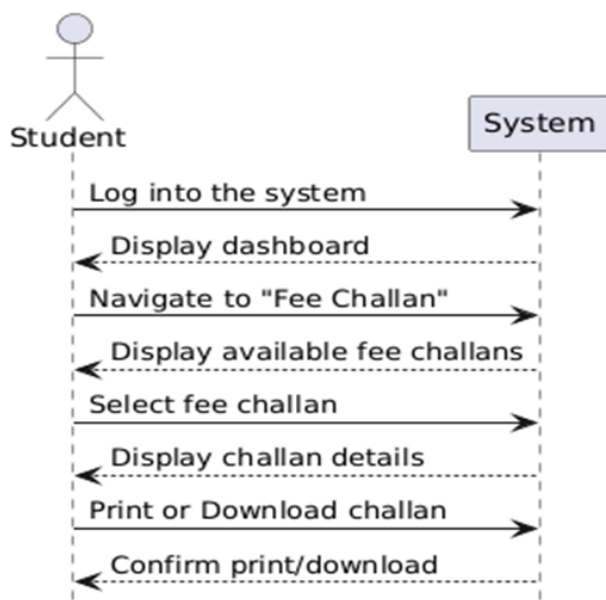
5. System Sequence Diagram

SSD 1: Register Course (Shahzaib Dars)

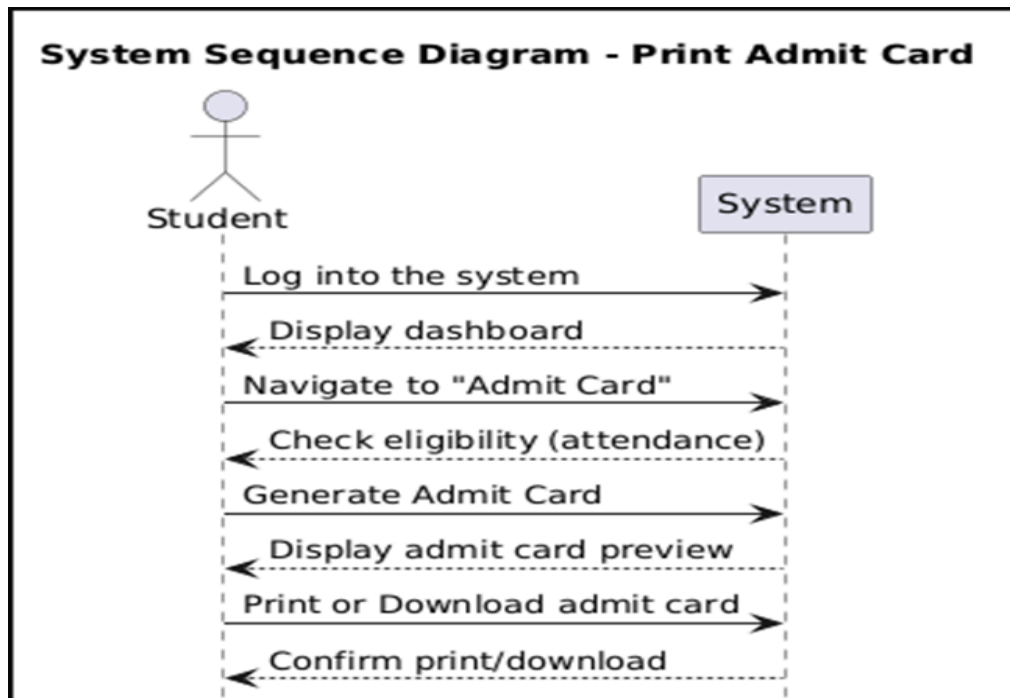


SSD 2: Print Fee Challan (Shahzaib Dars)

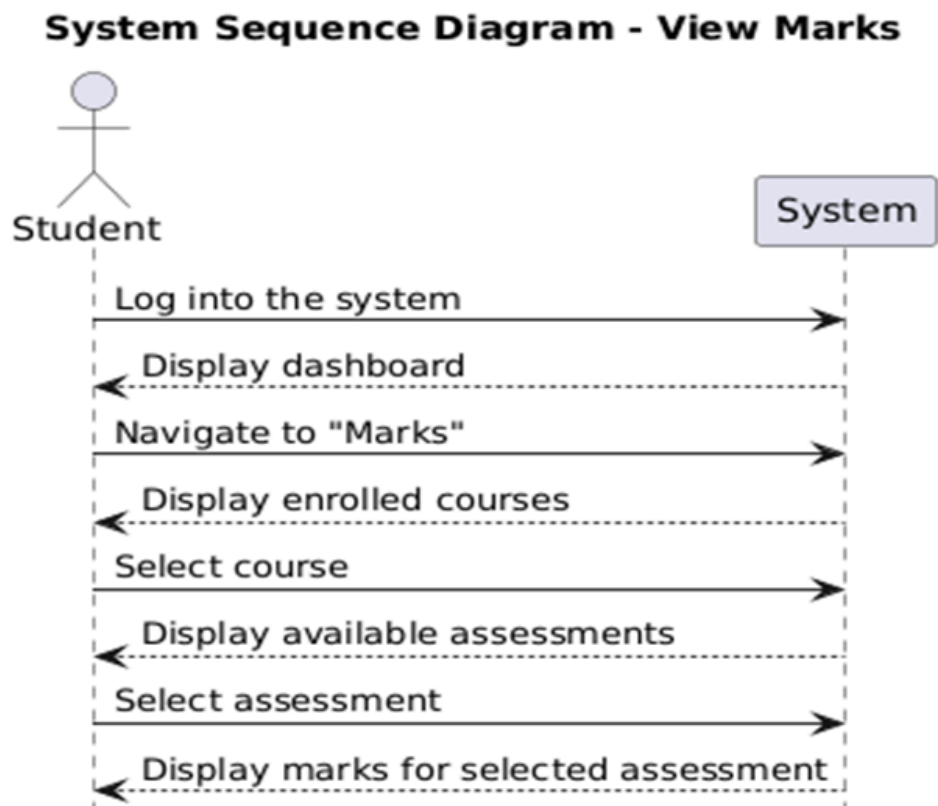
System Sequence Diagram - Print Fee Challan



SSD 3: Print Admit Card (Shahzaib Dars)

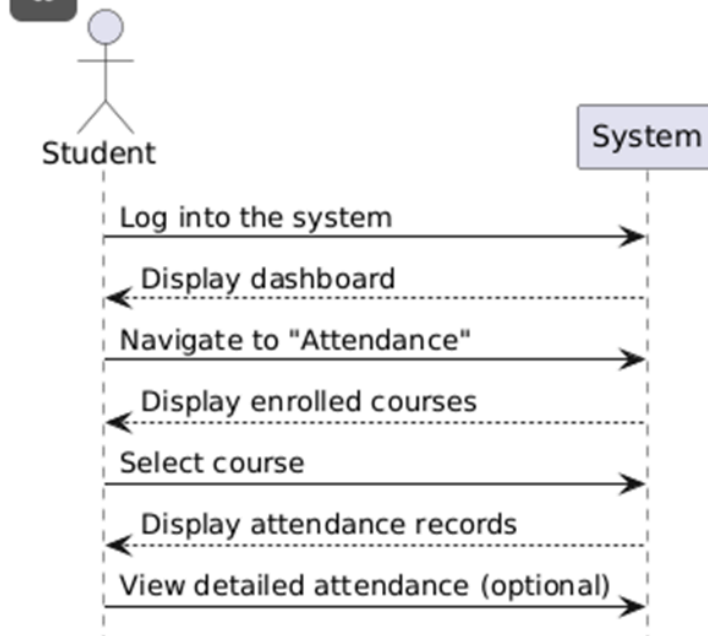


SSD 4: View Marks (Abeerah Aamir)



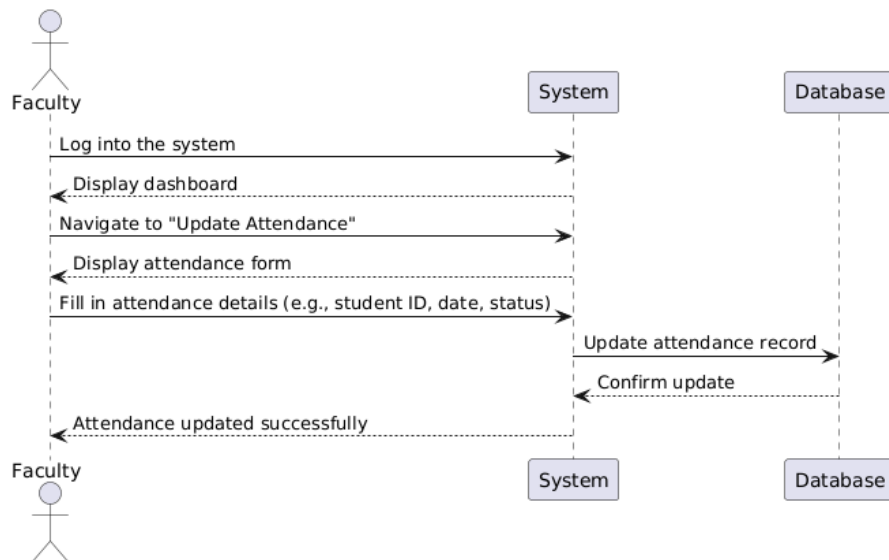
SSD 5: View Attendance(Abeerah Aamir)

System Sequence Diagram - View Attendance



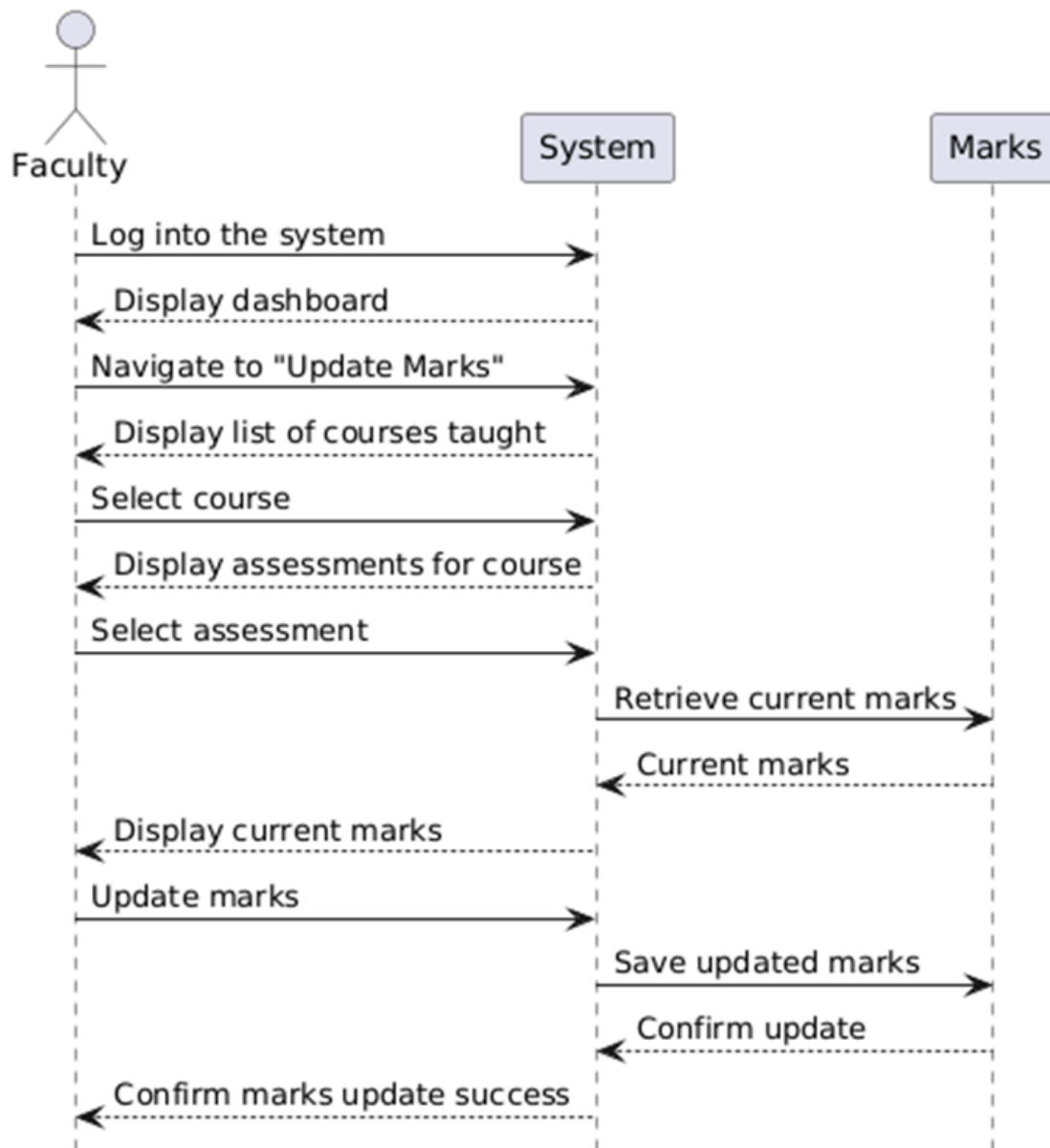
SSD 6: Update Attendance(Abeerah Aamir)

System Sequence Diagram - Update Attendance



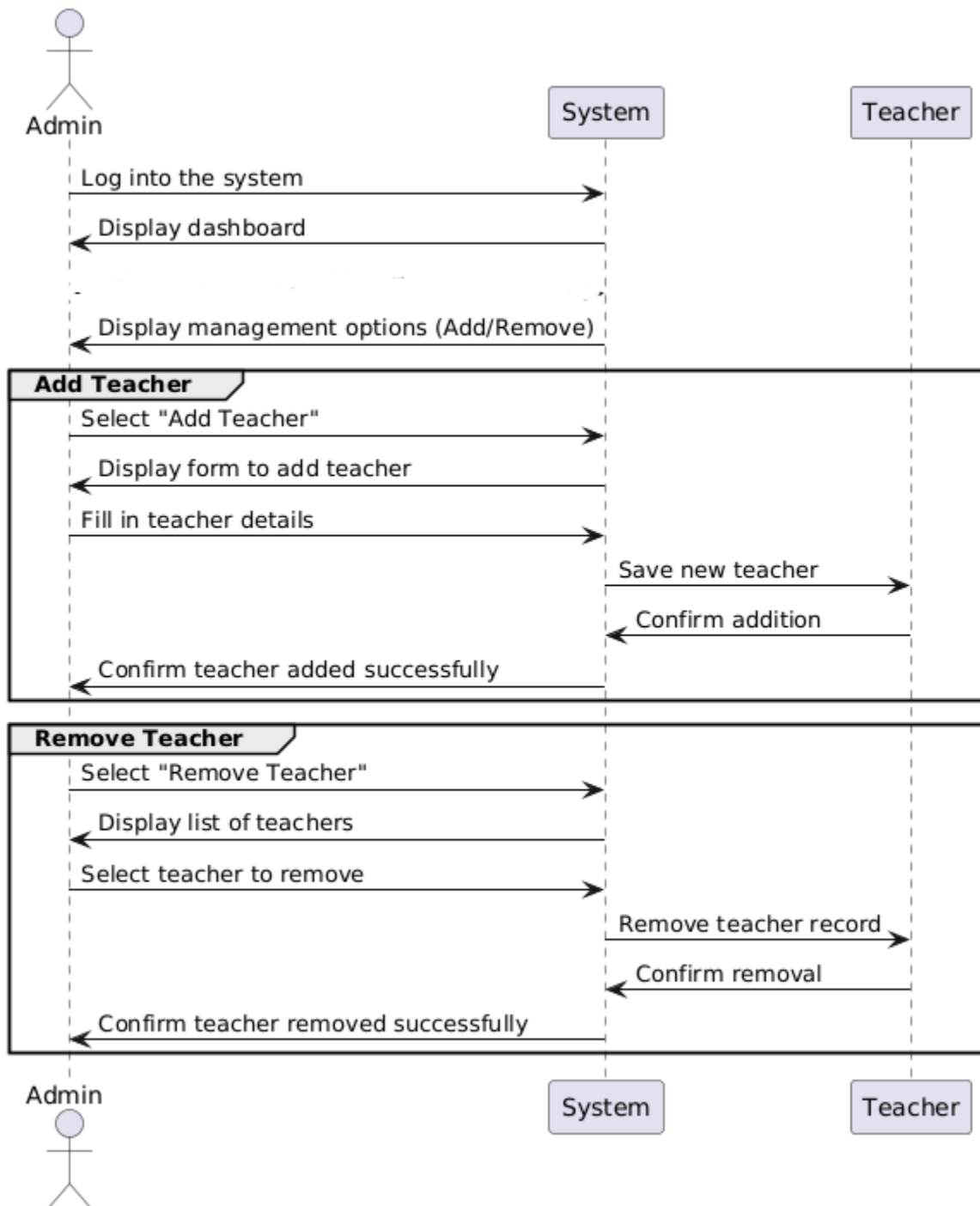
SSD 7: Update Marks (Abeerah Aamir)

System Sequence Diagram - Update Marks



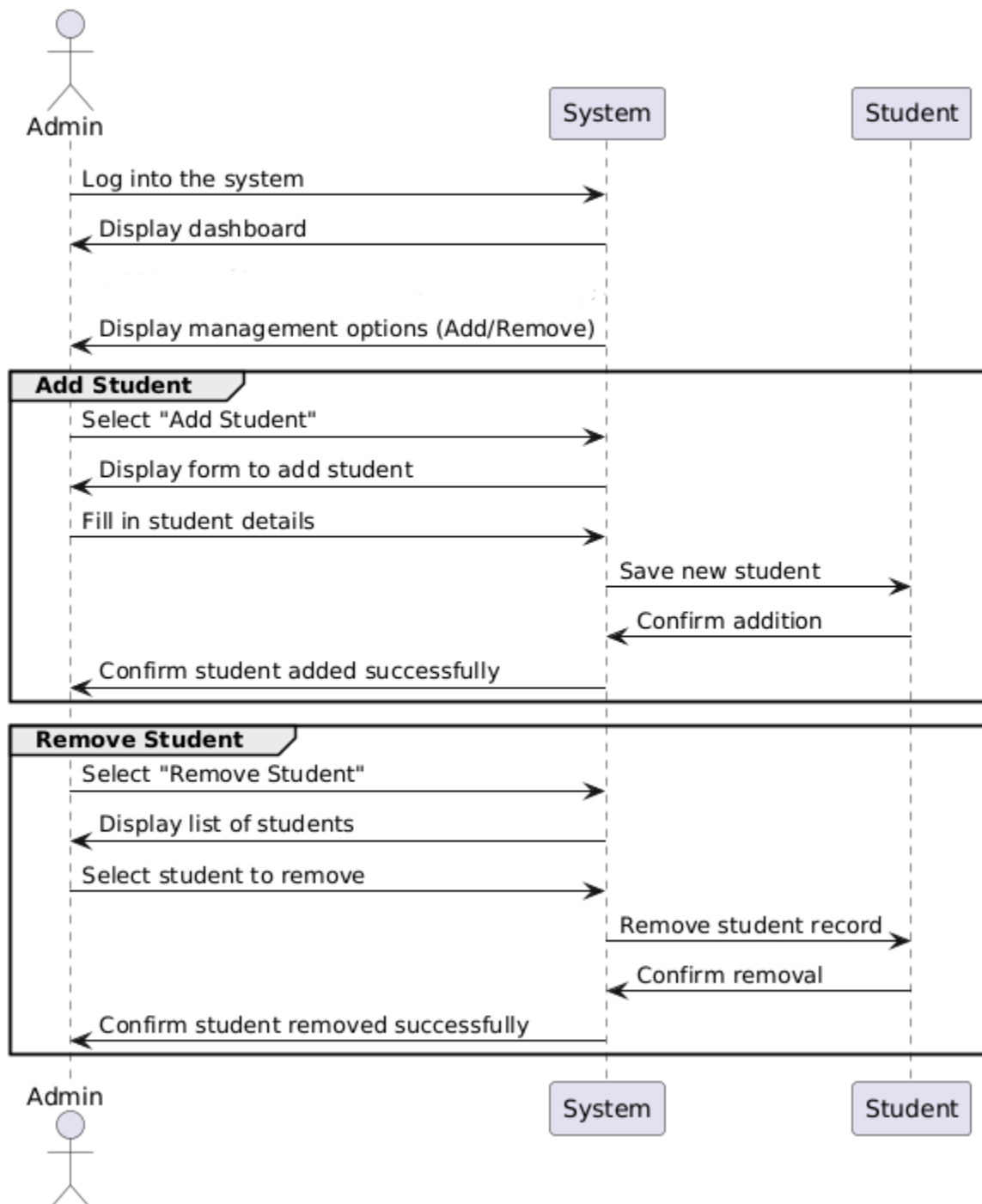
SSD 8: Manage Teacher (Ali Haider)

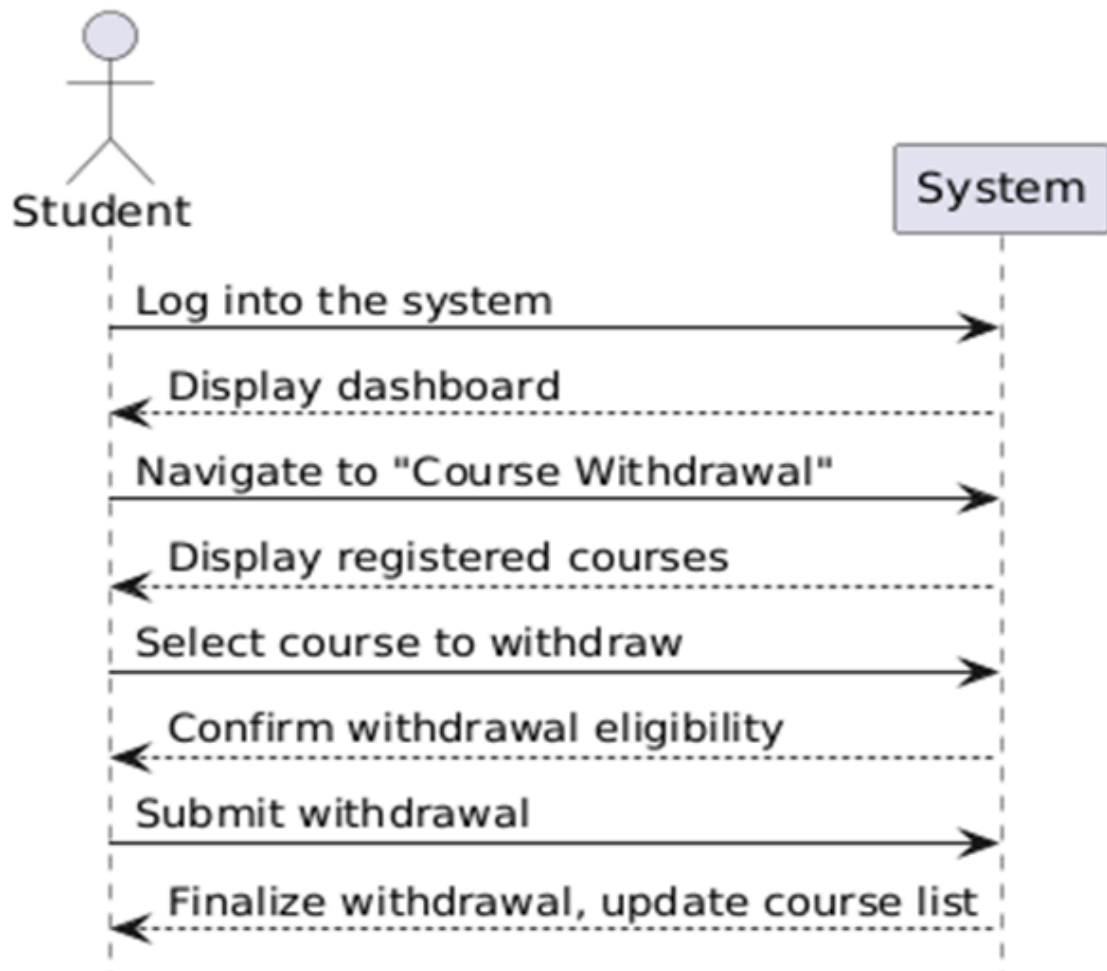
System Sequence Diagram - Manage Teacher



SSD 9: Manage Student (Ali Haider)

System Sequence Diagram - Manage Student

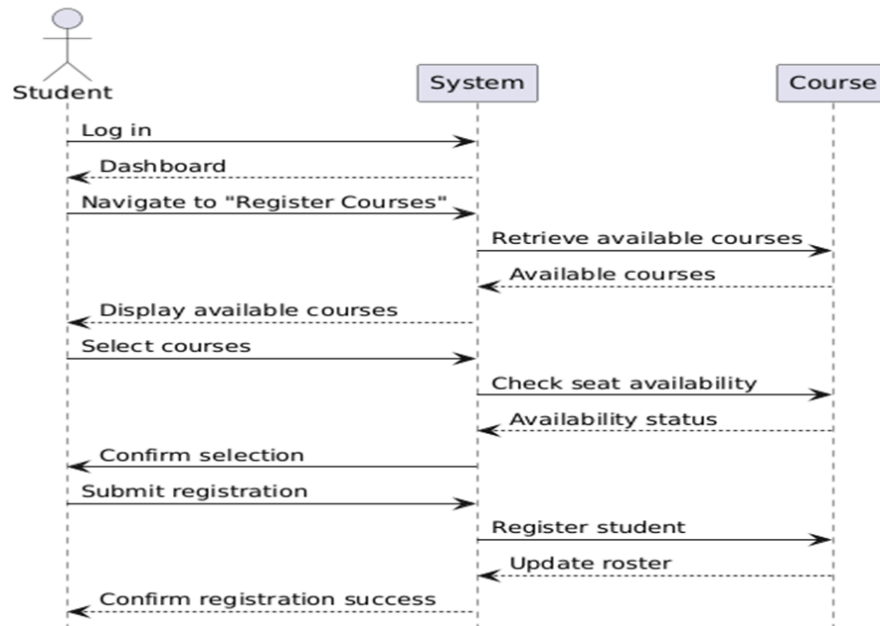


SSD 10: Withdraw Course (Shahzaib Dars)**System Sequence Diagram - Withdraw Course**

6. Sequence Diagram

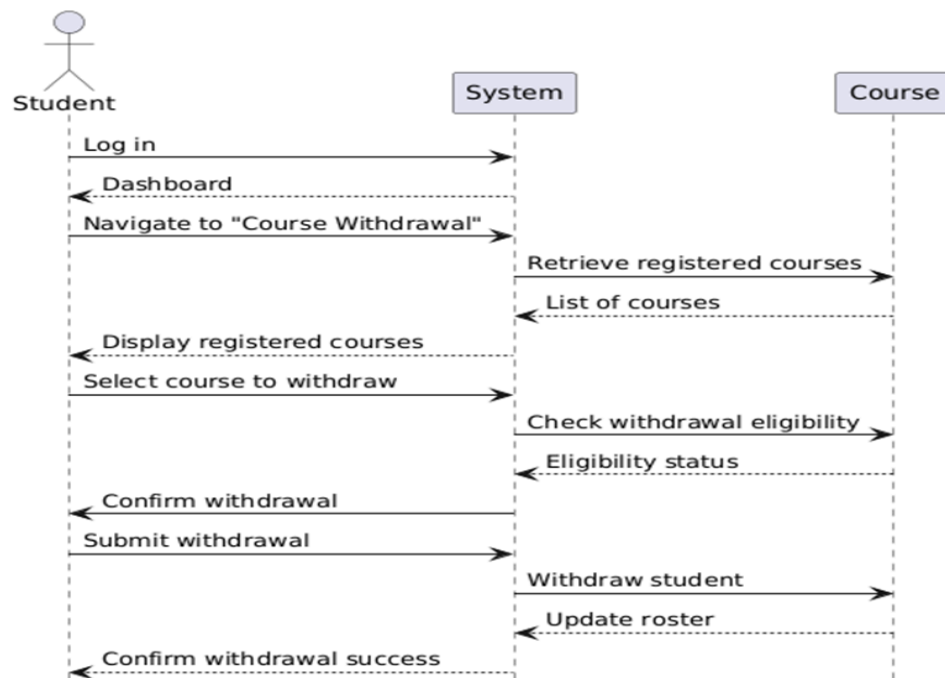
SD 1: Register Course (Shahzaib Dars)

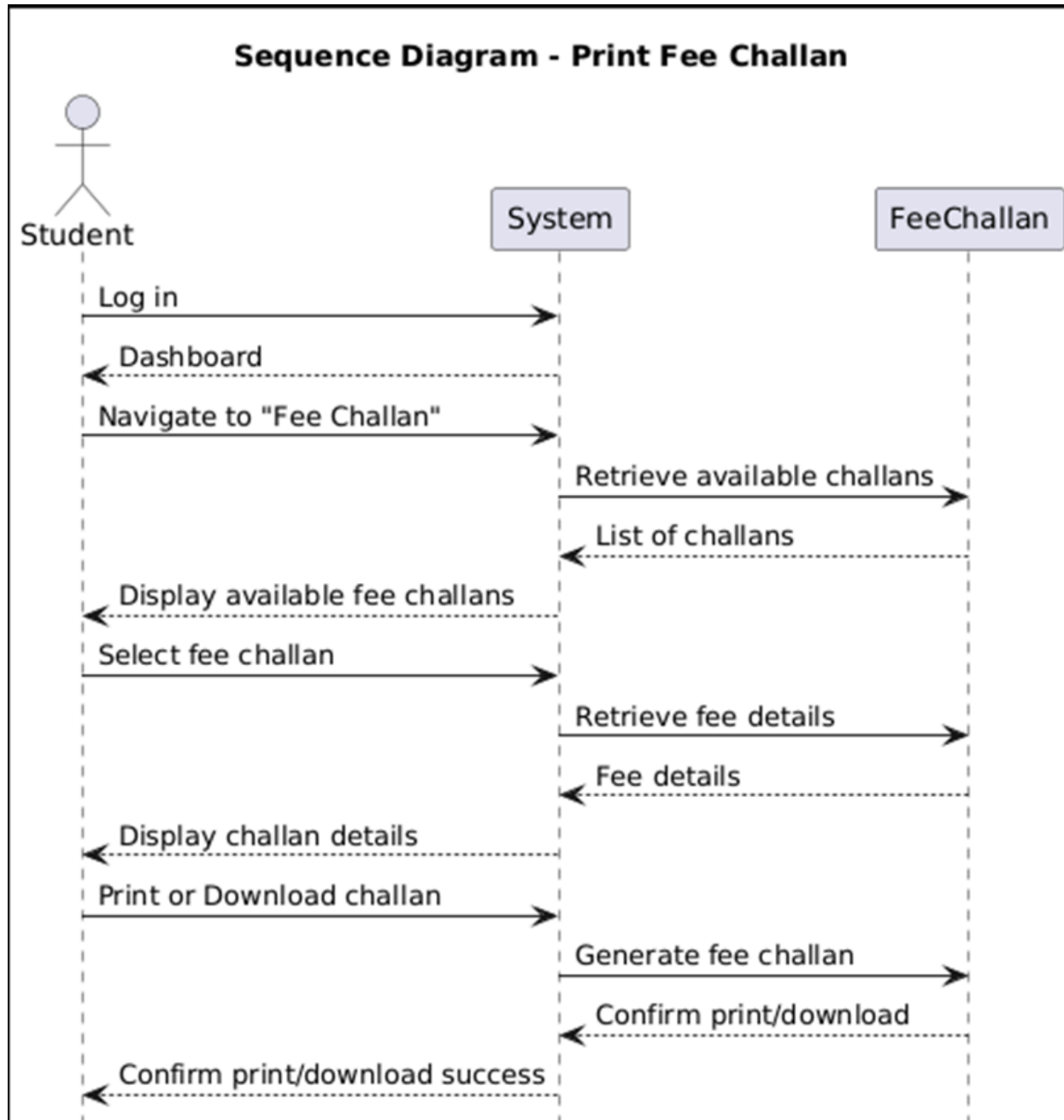
Sequence Diagram - Register Courses

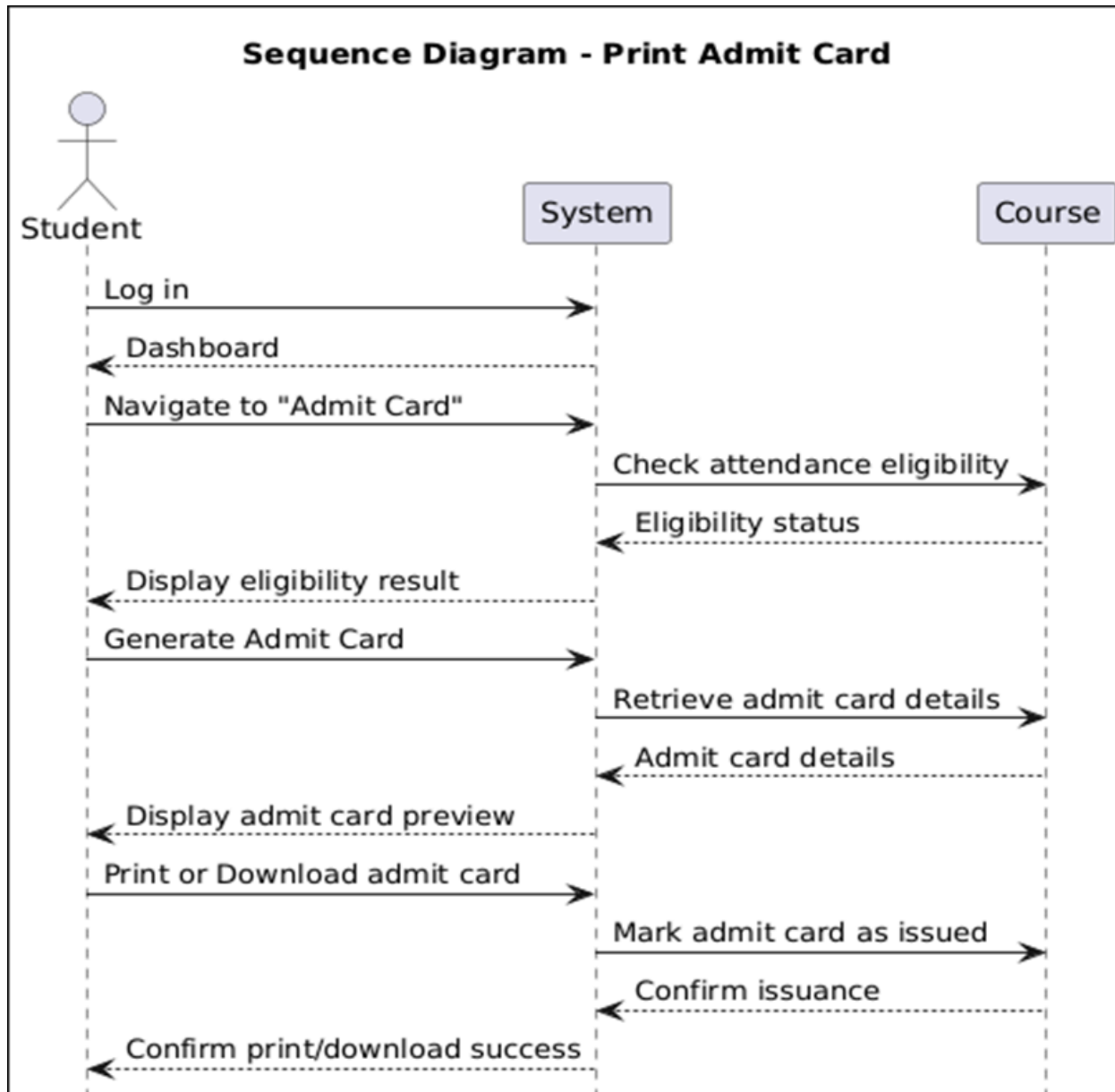


SD 2: Withdraw Course Course (Shahzaib Dars)

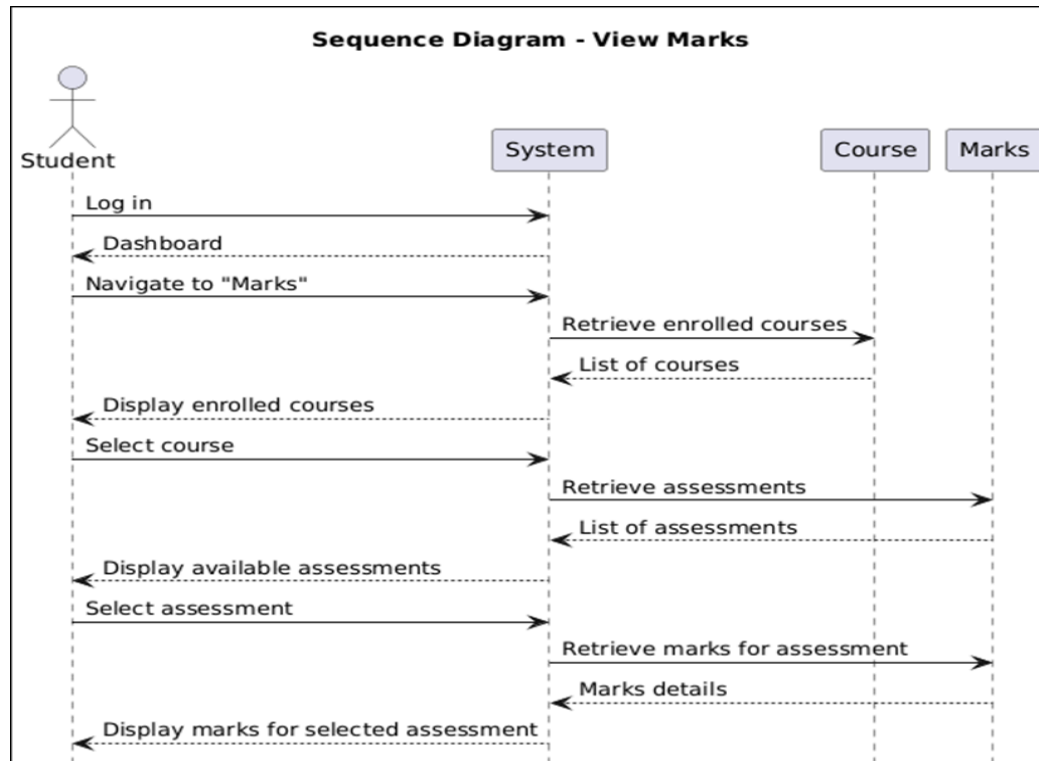
Sequence Diagram - Withdraw Course



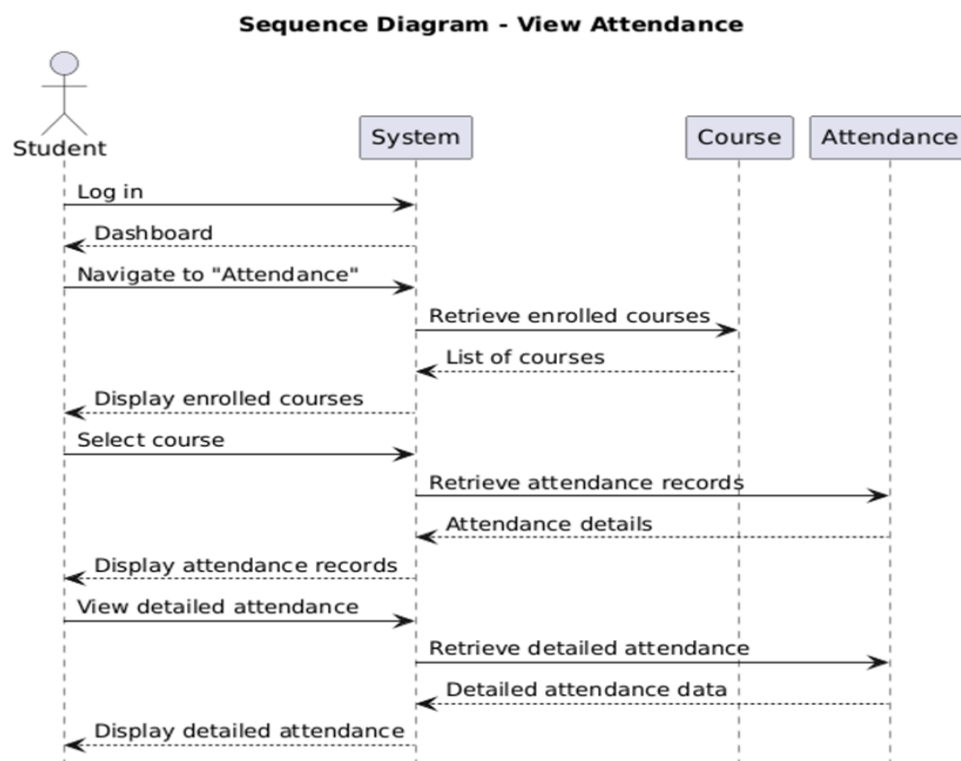
SD 3: Print Fee Challan (Shahzaib Dars)

SD 4: Print Admit Card (Shahzaib Dars)

SD 5: View Marks (Abeerah Aamir)

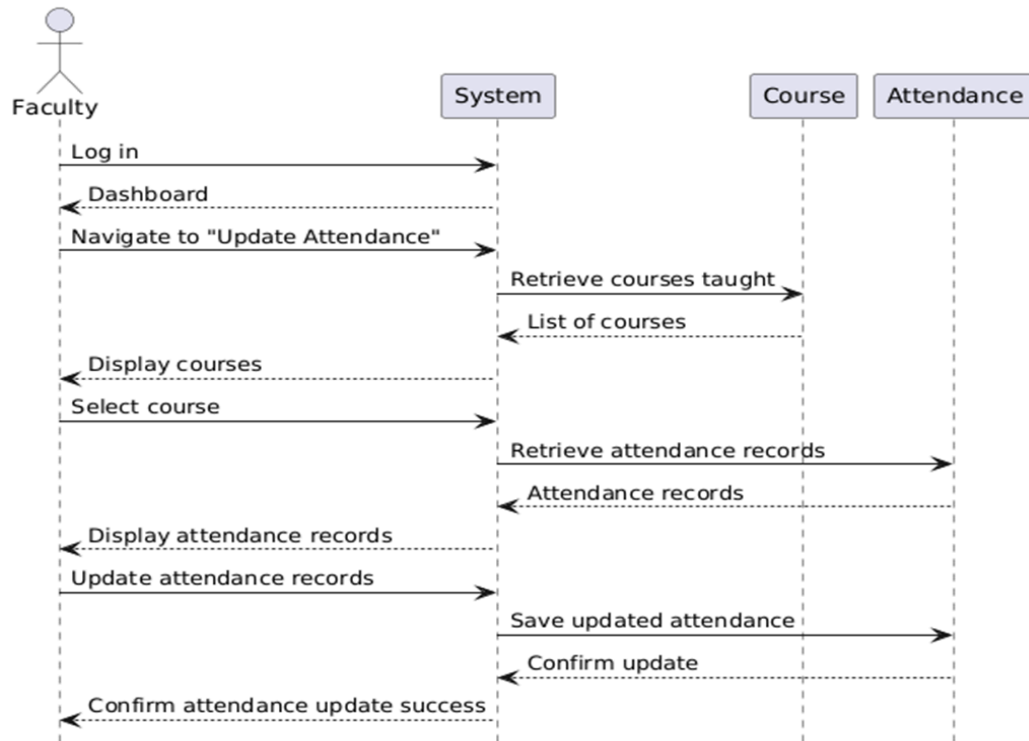


SD 6: View Attendance (Abeerah Aamir)



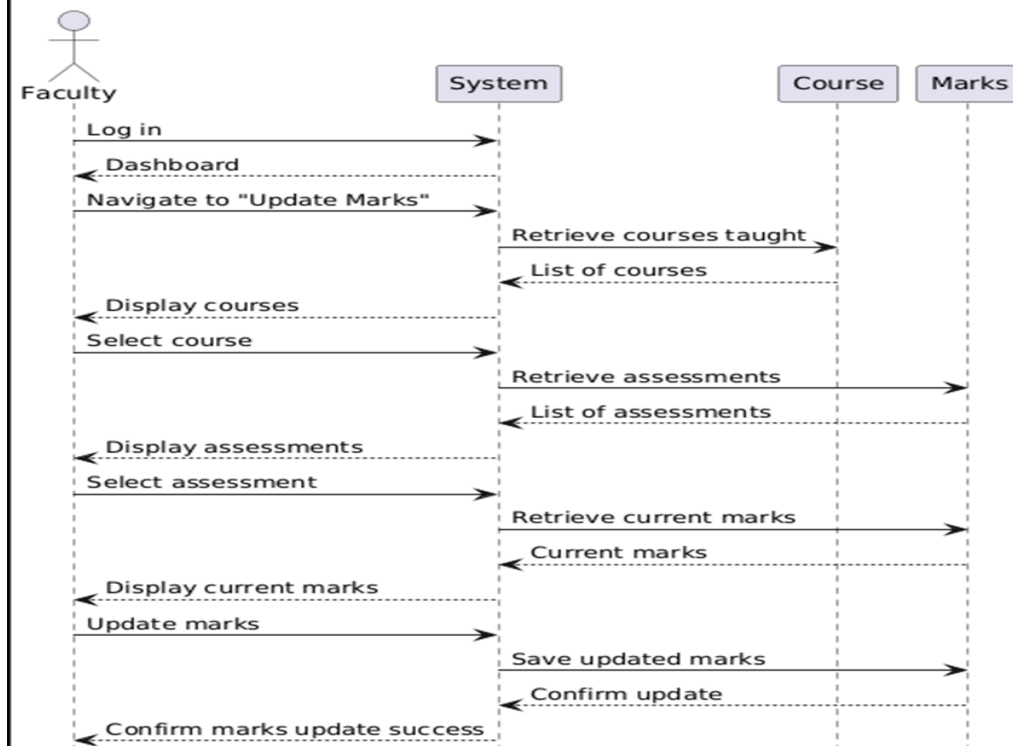
SD 7: Update Attendance (Abeerah Aamir)

Sequence Diagram - Update Attendance

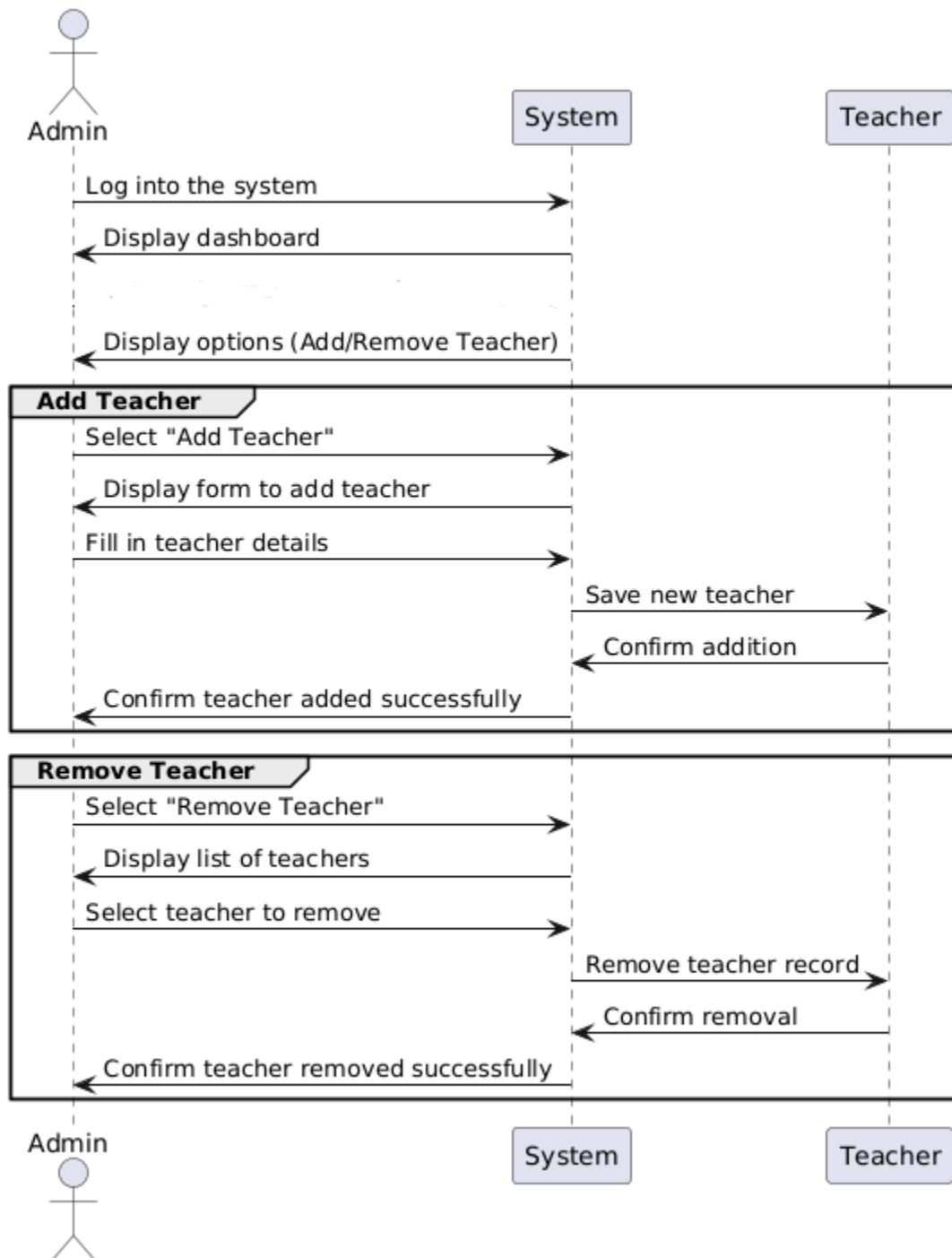


SD 8: Update Marks(Abeerah Aamir)

Sequence Diagram - Update Marks

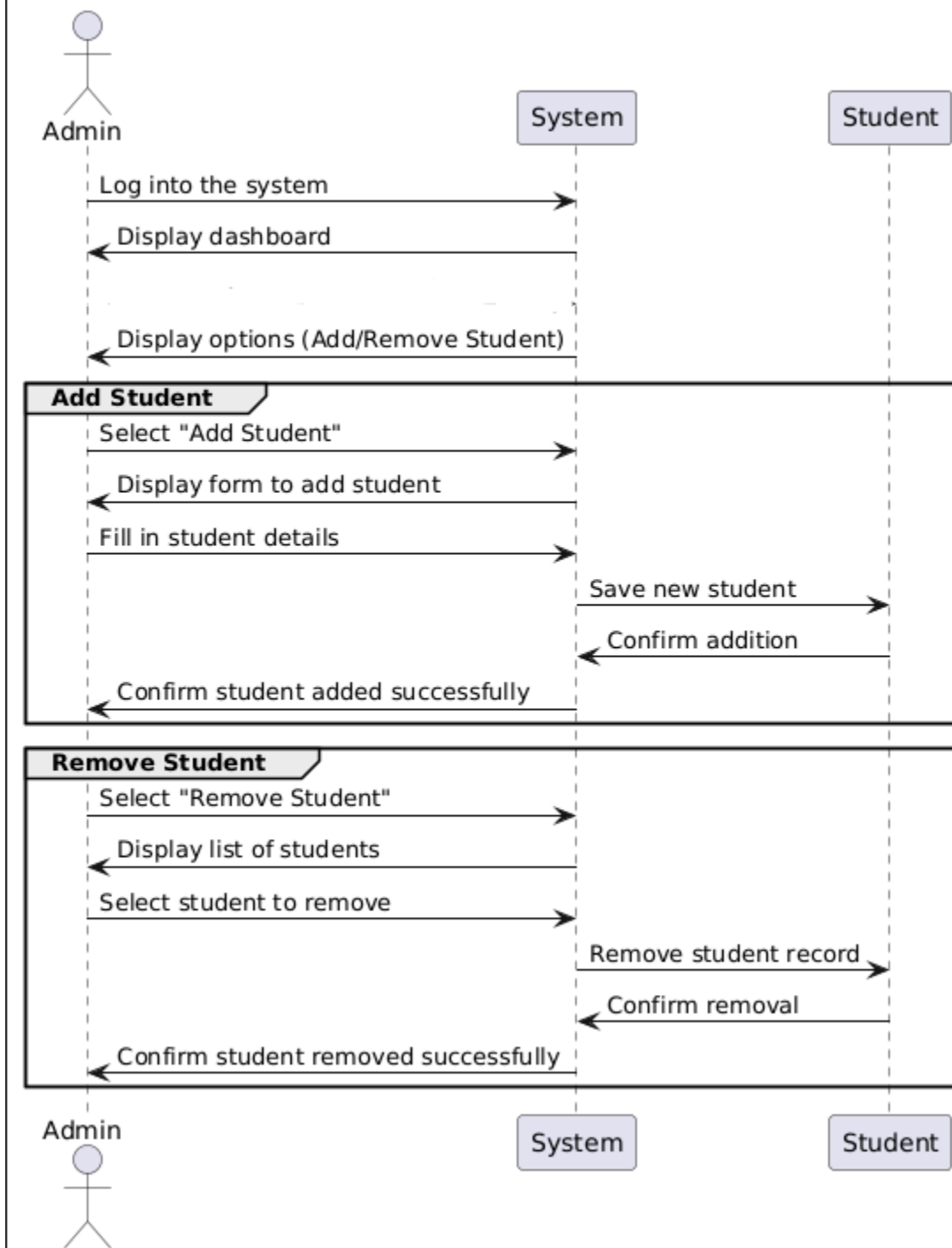


Sequence Diagram - Manage Teacher

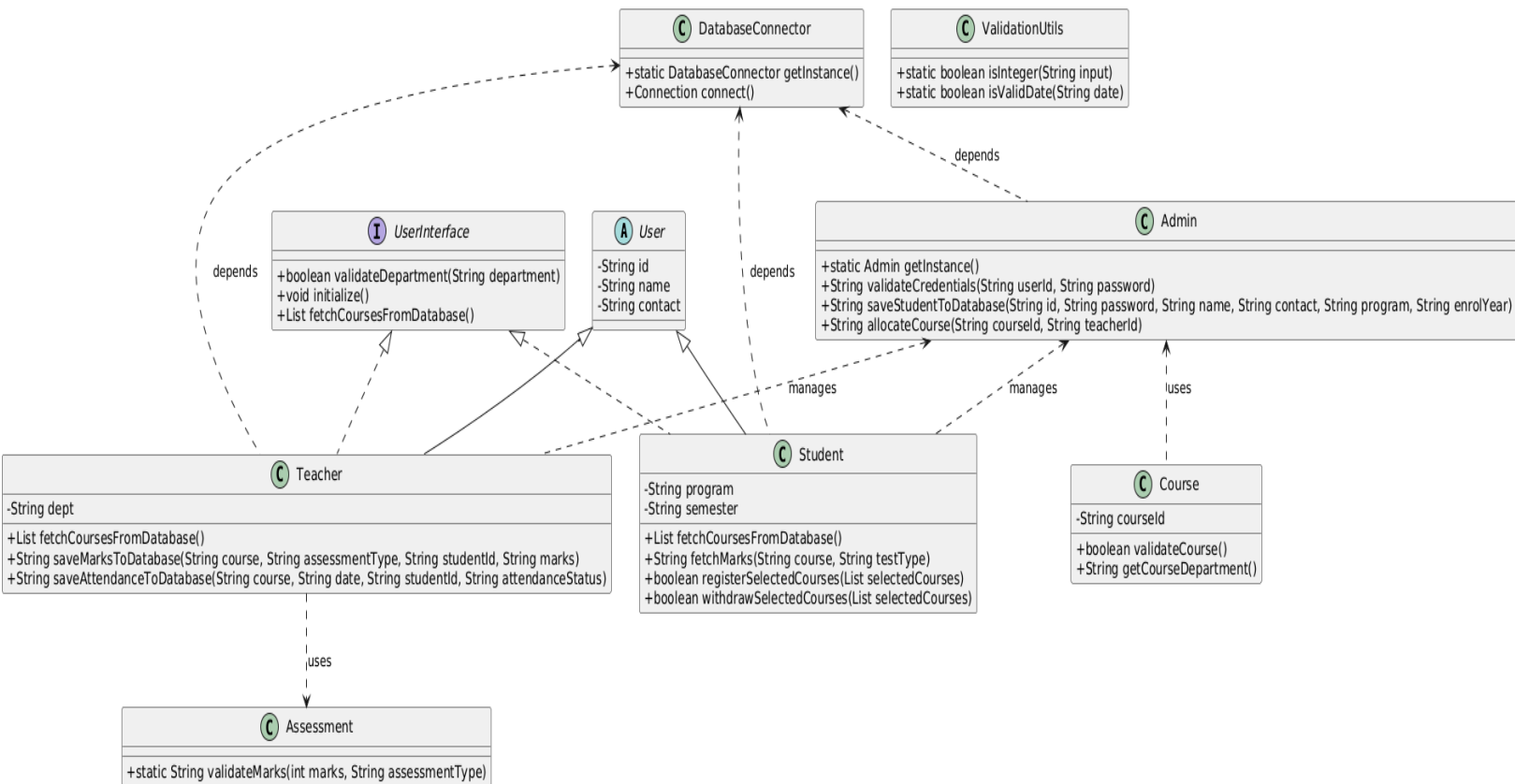


SD 10: Manage Student (Ali Haider)

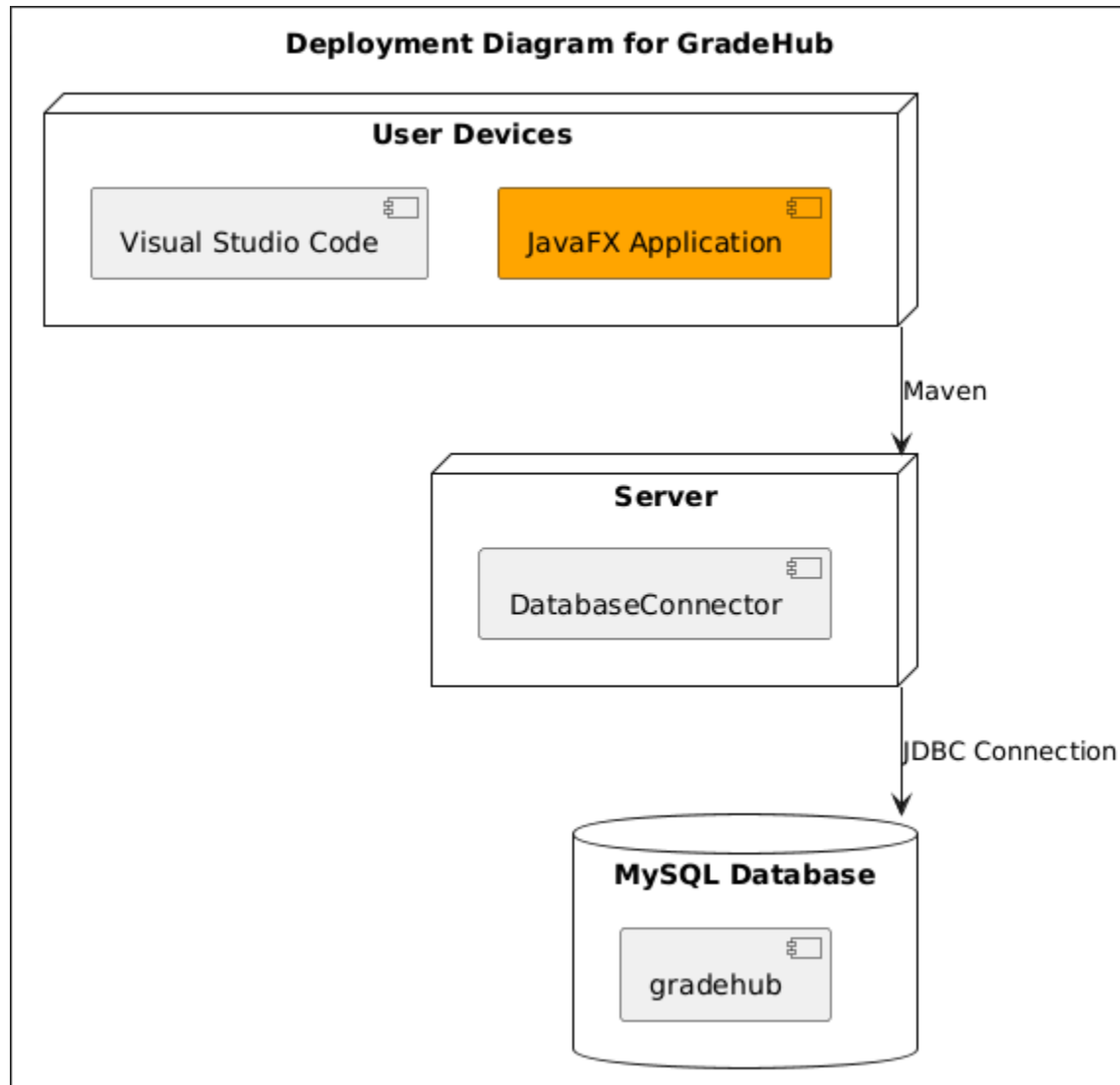
Sequence Diagram - Manage Student



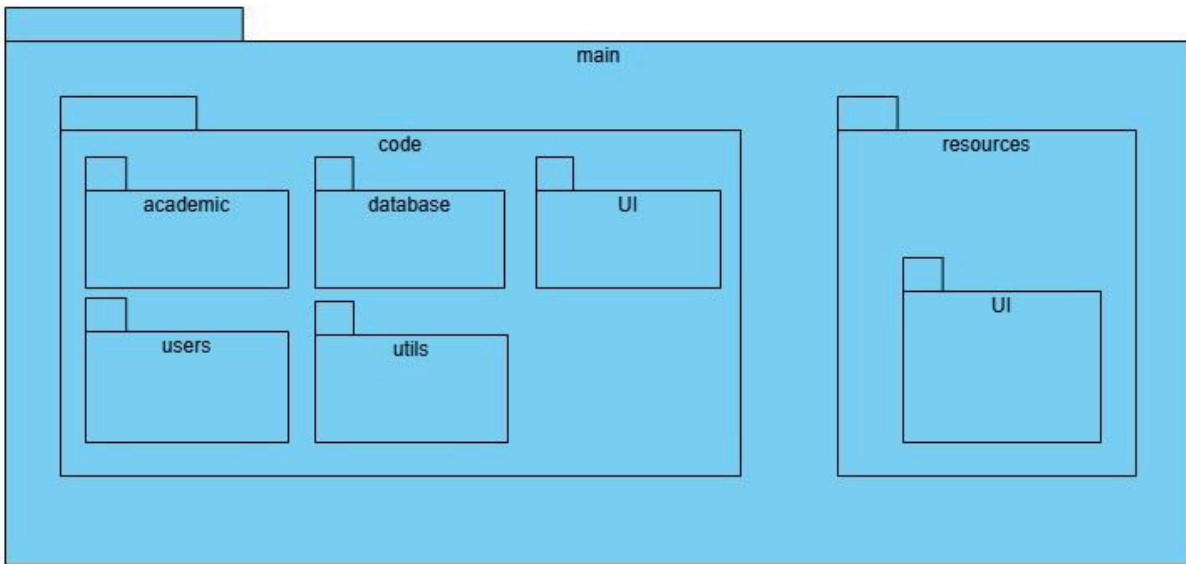
7. Class Diagram



8. Deployment Diagram



9. Package Diagram



10. Component Diagram

Component Diagram for GradeHub