

OBJECT-ORIENTED SOFTWARE ENGINEERING (SE204n)
LAB FILE

Subject Code: SE204n

Subject Name: Object-Oriented Software Engineering

Branch: Software Engineering

Submitted by:

ASTHA AGARWAL

Submitted to:

Prof. Ruchika Malhotra

Head of Department

Department of Software Engineering

Delhi Technological University



Delhi Technological University
Shahbad Daultpur, Main Bawana Road, Delhi-110042

EXPERIMENT-1

AIM: Draft the Problem Statement for the course management system

Description:

The software should enable efficient management of all the courses a student has opted for online. The students should be able to view their attendance, view and download course material, check their grades and upload submissions. A faculty can upload their assignments, view grades and accept submissions with the help of the system.

Limitations of conventional course management system:

1. Lack of Attendance Visibility for Students:

In conventional course management systems, students do not have access to their attendance records. This lack of transparency makes it difficult for students to track their attendance status, potentially leading to missed opportunities for improvement and accountability.

2. Limited Networking Opportunities:

Traditional course management systems do not provide details about classmates, which hinders networking and collaboration among students. The absence of a shared platform for interaction makes it challenging for students to connect, discuss course materials, or form study groups.

3. Absence of Overall Grade Overview:

Students are often unable to view their overall grades and performance summaries in conventional systems. This limitation prevents them from assessing their progress throughout the course and identifying areas that require additional effort.

4. Lack of Performance Analytics for Teachers:

In existing systems, teachers do not have access to comprehensive performance analytics, such as overall grade distributions and average scores. As a result, it is difficult to identify trends and address common areas of difficulty.

Features of the proposed system:

1. Enhanced Attendance Tracking for Students:

The proposed system will provide students with real-time access to their attendance records. This feature will enhance transparency and allow students to monitor their attendance status, enabling them to stay informed and take necessary actions to maintain required attendance levels.

2. Enhanced Networking and Collaboration:

The new system will include a dedicated platform where students can view details of their classmates, enhancing peer-to-peer learning.

3. Comprehensive Grade Overview for Students:

The proposed system will offer students an overall view of their grades and

performance trends across various assessments. This feature will help them track their academic progress, set personal goals, and work on areas needing improvement.

4. Advanced Performance Analytics for Teachers:

Teachers will have access to detailed performance analytics, including overall grade distributions and average score trends. This feature will provide valuable insights into class performance, enabling educators to make data-driven decisions and offer targeted support to students.

Astha Agarwal

EXPERIMENT-2

AIM: Draft the Initial requirements document for the course management system

Title of the Project	Course Management System
Stakeholders involved in capturing requirements	Student, Instructor, Admin
Techniques used for requirement capturing	Brainstorming
Name of the persons along with designation	Vasavi Taneja - developer Astha Agarwal - developer
Date	January, 2025
Version	1.0
Consolidated list of requirements <ol style="list-style-type: none">1. A system is to be implemented which can run on the user's LAN.2. The system should be able to generate and maintain the login ID and password of all possible users.3. There are two types of members in the course management system - students and faculty.4. The administrator should be able to maintain the details of all the members of the course management system.5. The administrator should be able to maintain the details of all the courses.6. The faculty should be able to upload assignments, accept and grade submissions.7. The faculty should be able to upload course material.8. The faculty should be able to view student details.9. The faculty should be able to update attendance of students.10. The faculty should be able to calculate the average performance of the class/11. The maximum number of files that can be uploaded for each submission is 3.12. The student should be able to view their attendance for each course.13. The student should be able to view courses they have registered for.14. The student should be able to view and edit profile details.15. The student should be able to download course material.16. The student should be able to make submissions and check grades.17. The student should be able to view details of classmates (email ID) and network with other students.18. The system should be able to generate reports like:<ol style="list-style-type: none">i) Details of all students and the courses they have enrolled inii) Attendance details of all students for a given course to the facultyiii) Details of submissions made by students	

EXPERIMENT-3

Software Requirements Specification Document for Library Management System

Problem Statement

A software is to be developed to enable efficient management of all the courses a student has opted online. The students should be able to view their attendance, view and download course material, check their grades and upload submissions. A faculty can upload their assignments, view grades and accept submissions with the help of the system.

The Course Management System performs the following functions:

1. Course Material and Assignment Management

- Upload Course Material (Faculty):
 - Faculty can browse and upload files to specific courses
 - System validates course ID and checks file size constraints
 - A deadline for assignment submission can be set
 - Faculty have the option to undo uploads if needed.
- Download Course Material (Student):
 - Students can download material uploaded for their registered courses.
 - System validates the course ID and ensures only enrolled students access the content.
 - Only materials uploaded up to the selected lecture are available.
- Upload Assignments (Faculty):
 - Faculty can upload assignment files to the portal for students.
 - Upload process includes file validation and deadline setting.
 - Unauthorized faculty (invalid membership) are blocked from uploading.

2. Student Submissions and Grading

- Make Submissions (Student):
 - Students can upload assignment files for respective courses.
 - System enforces file size limits and submission deadlines.
 - Invalid course ID or late submissions are rejected with an error message.
 - Students can undo submissions before the deadline.
- Grade Submissions (Faculty):
 - Faculty can access a list of student submissions per course.
 - Submissions can be viewed and graded directly through the portal.
 - Defaulters (non-submitters) are also listed separately.
 - Invalid course ID or student ID results in an error and restart of the process.

3. Academic Monitoring and Access

- View Attendance (Student):

- Students can view attendance records for their registered courses.
- Attendance is shown up to the last recorded lecture.
- Invalid course ID or premature exit from the process is handled appropriately.
- View Classmate Details (Student):
 - Students can access contact details (e.g., college email) of classmates in a course.
 - Details are managed and uploaded by the administrator.
 - Access is restricted to valid course registrations only.
- Calculate Average Performance (Faculty):
 - Faculty can compute the average grade performance of a class.
 - The system calculates based on the grades uploaded.
 - If no grades are uploaded or course ID is invalid, the operation fails gracefully.

4. Maintain Details (Student, Faculty, and Course Information)

- Maintain Student Details:
 - The system stores comprehensive student records including name, student ID, email, registered courses, and submission history.
 - Administrators can add, update, or delete student records.
 - Updates ensure synchronization with current course enrollment and grade records.
- Maintain Faculty Details:
 - Faculty records include name, faculty ID, department, and assigned courses.
 - Faculty membership is verified for upload and grading permissions.
 - Admins can update faculty information or revoke access (e.g., on expiry of membership).
- Maintain Course Details:
 - Each course includes course ID, course name, assigned faculty, and list of enrolled students.
 - Courses can be added or modified by administrators.
 - All course-related operations (uploads, downloads, grades, attendance) are linked through validated course IDs.

Contents

Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, Acronyms, and Abbreviations
- 1.4 References

System Description

- 2.1 User Perspective
- 2.2 System Interfaces
- 2.3 User Interfaces
- 2.4 Hardware Interfaces
- 2.5 Software Interfaces
- 2.6 Communication Interfaces
- 2.7 Memory Constraints
- 2.8 Operations
- 2.9 Site Adaptation Requirements
- 2.10 Product Functions

- 2.11 User Characteristics
- 2.12 Constraints
- 2.13 Assumptions and Dependencies
- 2.14 Apportioning of Requirements

Specific Requirements

- 3.1 External Interface Requirements
 - 3.1.1 User Interfaces
 - 3.1.2 Hardware Interfaces
 - 3.1.3 Software Interfaces
 - 3.1.4 Communication Interfaces
- 3.2 Functional Requirements
- 3.3 Performance Requirements
- 3.4 Design Constraints
- 3.5 Software System Attributes
 - 3.5.1 Reliability
 - 3.5.2 Availability
 - 3.5.3 Security
 - 3.5.4 Maintainability
 - 3.5.5 Portability
- 3.6 Logical Database Requirements
- 3.7 Other Requirements

1.Introduction

The purpose of this document is to outline the software requirements for the Course Management System (CMS). The CMS aims to streamline and automate academic course-related activities for students, faculty, and administrators, including course enrollment, assignment submission, material distribution, attendance management, and report generation.

1.1 Purpose

The software should enable efficient management of all the courses a student has opted for online. The students should be able to view their attendance, view and download course material, check their grades and upload submissions. A faculty can upload their assignments, view grades and accept submissions with the help of the system.

1.2 Scope

This system will support academic institutions in managing student and faculty activities effectively over a secure LAN network. Key functionalities include:

- User authentication and role-based access
- Uploading and managing course materials and assignments
- Student submissions and faculty grading
- Attendance management
- Performance calculations
- Viewing and managing course, student, and faculty details
- Report generation for analytics and record keeping

1.3 Definitions and Acronyms

- CMS: Course Management System
- UI: User Interface

- LAN: Local Area Network
- SRS: Software Requirements Specification
- DB: Database

1.4 References

- (a) Software Engineering by K.K. Aggarwal & Yogesh Singh
- (b) Software Engineering by Ruchika Malhotra & Yogesh Singh
- (c) IEEE Recommended Practice for Software Requirements Specifications—IEEE Std. 830-1998.

2. Overall Description

2.1 Product Perspective

The CMS is a standalone application deployed on an institution's internal LAN. It interacts with a central database and provides role-based interfaces for students, faculty, and administrators. The system is designed to be scalable and adaptable to different educational institutions.

2.2 System Interfaces

- DBMS for persistent storage (e.g., MySQL/PostgreSQL)
- Authentication service (for login/password validation)
- Local institutional servers for deployment

2.3 User Interfaces

- Student Dashboard: View courses, submissions, grades, attendance, and download materials
 - Faculty Dashboard: Upload assignments, manage attendance, grade submissions
 - Admin Dashboard: Manage users, courses, and reports
- All interfaces are web-based with responsive design for accessibility across devices.

2.4 Hardware Interfaces

- Server: Intel i5/i7 processor, 16 GB RAM, 1 TB HDD
- Clients: Any modern device with a browser (PC, tablet, etc.)

2.5 Software Interfaces

- Backend: Python
- Database: MySQL
- Frontend: HTML5, CSS, JS

2.6 Communication Interfaces

- System runs on LAN; communication over TCP/IP
- Secure transmission via HTTPS for client-server interaction

2.7 Memory Constraints

- Server should support at least 100 concurrent users with 16 GB RAM
- Disk space requirements scale with course material and user count

2.8 Operations

- System backup daily at midnight

- Maintenance window scheduled weekly for updates
- Logs maintained for user activities

2.9 Site Adaptation Requirements

- System adaptable to other LAN environments with minimal configuration
- Site-specific data like institution name, logo, etc., configurable via admin panel

2.10 Product Functions

The Course Management System will support the following key functions:

- **Authentication:** Secure login for students, faculty, and administrators.
- **User Management:** Add, update, and delete user profiles.
- **Course Management:** Create and maintain course details.
- **Material Upload:** Faculty can upload lecture notes and other materials.
- **Assignment Management:** Upload assignments, accept submissions, and assign grades.
- **Attendance Management:** Faculty can update and students can view attendance.
- **Report Generation:** Generate detailed reports for admin and faculty on student enrollment, attendance, and submissions.
- **Peer Networking:** Students can view basic classmate details to foster communication.

2.11 User Characteristics

- **Students:** Basic computer and internet usage skills. Will primarily use the system to view/download materials, submit assignments, and track academic progress.
- **Faculty:** Moderate technical knowledge. Will upload content, grade submissions, manage attendance, and view reports.
- **Administrators:** Technically proficient. Responsible for managing users, courses, and generating institutional-level reports.

2.12 Constraints

- The system must operate on a secure institutional LAN.
- Users must not upload more than three files per assignment submission.
- All user passwords must be stored in encrypted form.
- The system should be operable only on devices with an internet browser.
- Limited English language UI in the first release.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Role-based dashboards

- Form-based inputs with validations
- Report generation interface with filters

3.1.2 Hardware Interfaces

- Compatible with standard server and client hardware

3.1.3 Software Interfaces

- Compatible with standard DBMS

3.1.4 Communication Interfaces

- All modules interact through secure internal APIs
- Encrypted data transmission over LAN

3.3 Performance Requirements

- The system should handle up to 1000 concurrent users efficiently
- Attendance and performance reports should generate in less than 5 seconds

3.4 Design Constraints

- Must support modular and scalable architecture
- System must be browser compatible (Chrome, Firefox, etc)

3.5 Software System Attributes

3.5.1 Reliability

- 99.9% uptime during working hours
- Fault-tolerant design for critical operation

3.5.2 Availability

- System available on LAN 24/7 except during maintenance

3.5.3 Security

- Role-based access
- Password encryption
- SQL injection prevention and input sanitization

3.5.4 Maintainability

- Modular codebase with inline documentation
- Admin panel for configuration and logs

3.5.5 Portability

- Deployable on any Linux/Windows server
- Frontend accessible on all standard browsers

3.6 Logical Database Requirements

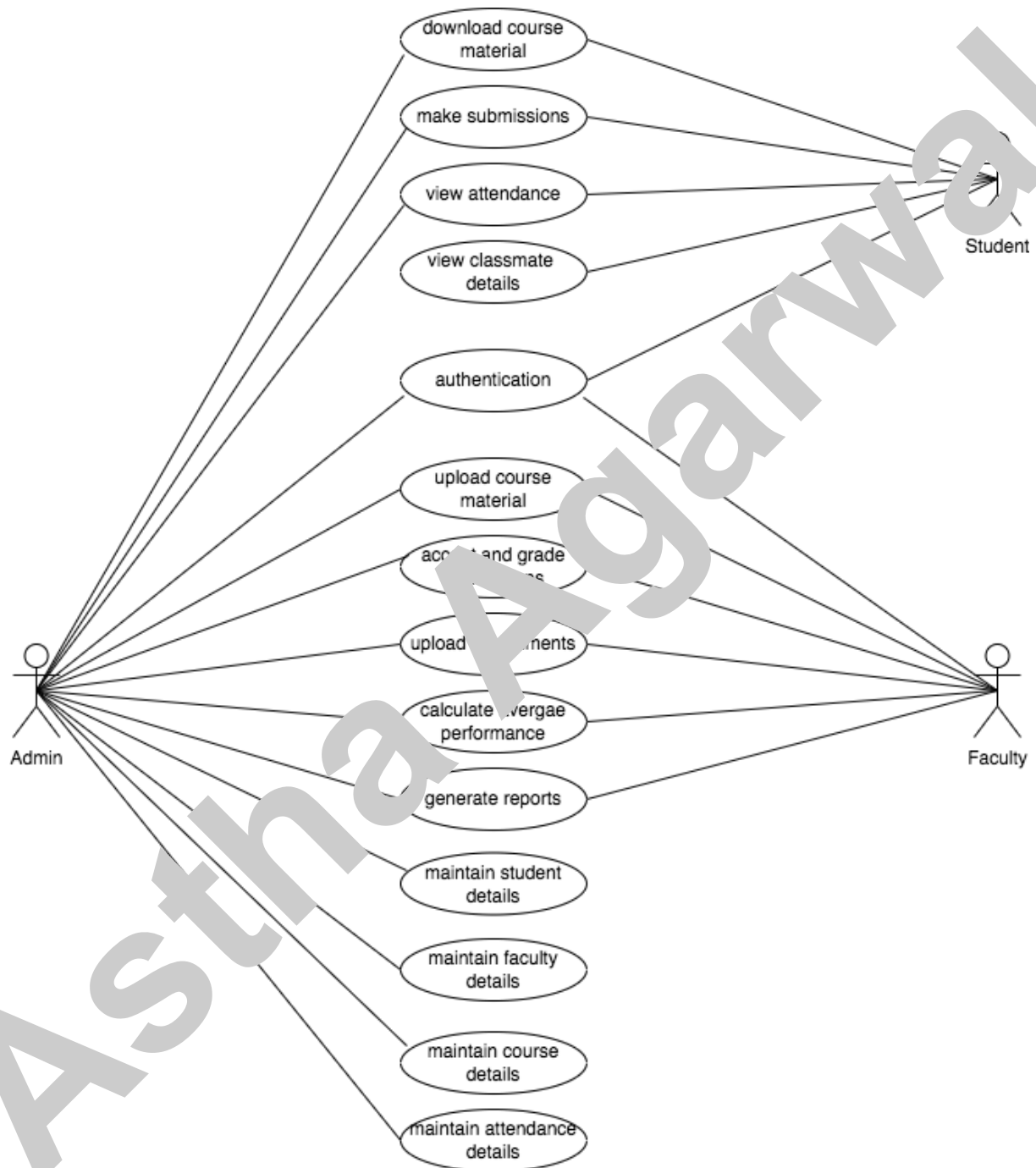
- User table (students, faculty, admins)
- Courses, Assignments, Submissions, Grades
- Attendance records
- Relational integrity with foreign key constraints

3.7 Other Requirements

- Maximum file upload per submission: 3 files
- Submissions must be timestamped

EXPERIMENT-4

AIM: Draft the Use Case Diagram for the Course Management System



EXPERIMENT-5

AIM: Draft the Use Case Descriptions for the Course Management System

Introduction: This use case documents the steps that must be followed in order to login to the system
Actors: Student/faculty/Admin
Preconditions: The user must have valid login ID and password
Postconditions: if the use case is successful, the user will be able to login and use the system.
Event Flow Basic Flow <ol style="list-style-type: none">1. The user enters valid Login Id2. The user enters valid password3. The user confirms login
Alternate flows: Alternative Flow 1: Unauthorized user If the system does not validate the login details, then the user is prompted to retry. Alternative Flow 2: User exits This allows the user to exit at any time during the use case. The use case ends.
Special requirements: None
Associated use cases: None

Introduction: This use case documents the steps that must be followed in order to download course material.
Actors: Student
Preconditions: The student must be logged onto the system before the use case begins
Postconditions: if the use case is successful, the student will be able to view their course registration details, otherwise the system remains unchanged.
Event Flow Basic Flow <ol style="list-style-type: none">1. The student selects the option to download course material for the specified course.2. The course material up till the date of the last lecture uploaded by the faculty is displayed.
Alternate flows:

Alternative Flow 1: Invalid courseID

If the system does not validate the courseID, then an error message is flagged and the use case returns to the beginning of the basic flow.

Alternative Flow 2: User exits

This allows the user to exit at any time during the use case. The use case ends.

Special requirements: None

Associated use cases: Login

Introduction: This use case documents the steps that must be followed in order to upload course material

Actors: Faculty

Preconditions: The faculty must be logged onto the system before the use case begins

Postconditions: If the use case is successful, the faculty will be able to upload their respective course material to the system for students to see. Else, the system remains unchanged

Event Flow

Basic Flow

1. The faculty selects the option to upload assignments for the specified course.
2. The interface allows the faculty to select the contents of the desktop to upload files.
3. The faculty selects the file to be uploaded and uploads the file. The faculty also sets the deadline for assignment submission.
4. The faculty selects the "Upload" option.
5. The database of course material is updated.
6. The faculty still has the choice to undo the material upload.

Alternate flows:

Alternate flow 1: File uploaded is too large

The upload is unsuccessful. Error message is displayed. Use case ends and returns to the beginning of the basic flow.

Alternative Flow 2: Invalid courseID

If the system does not validate the courseID, then an error message is flagged and the use case returns to the beginning of the basic flow.

Alternative Flow 3: User exits

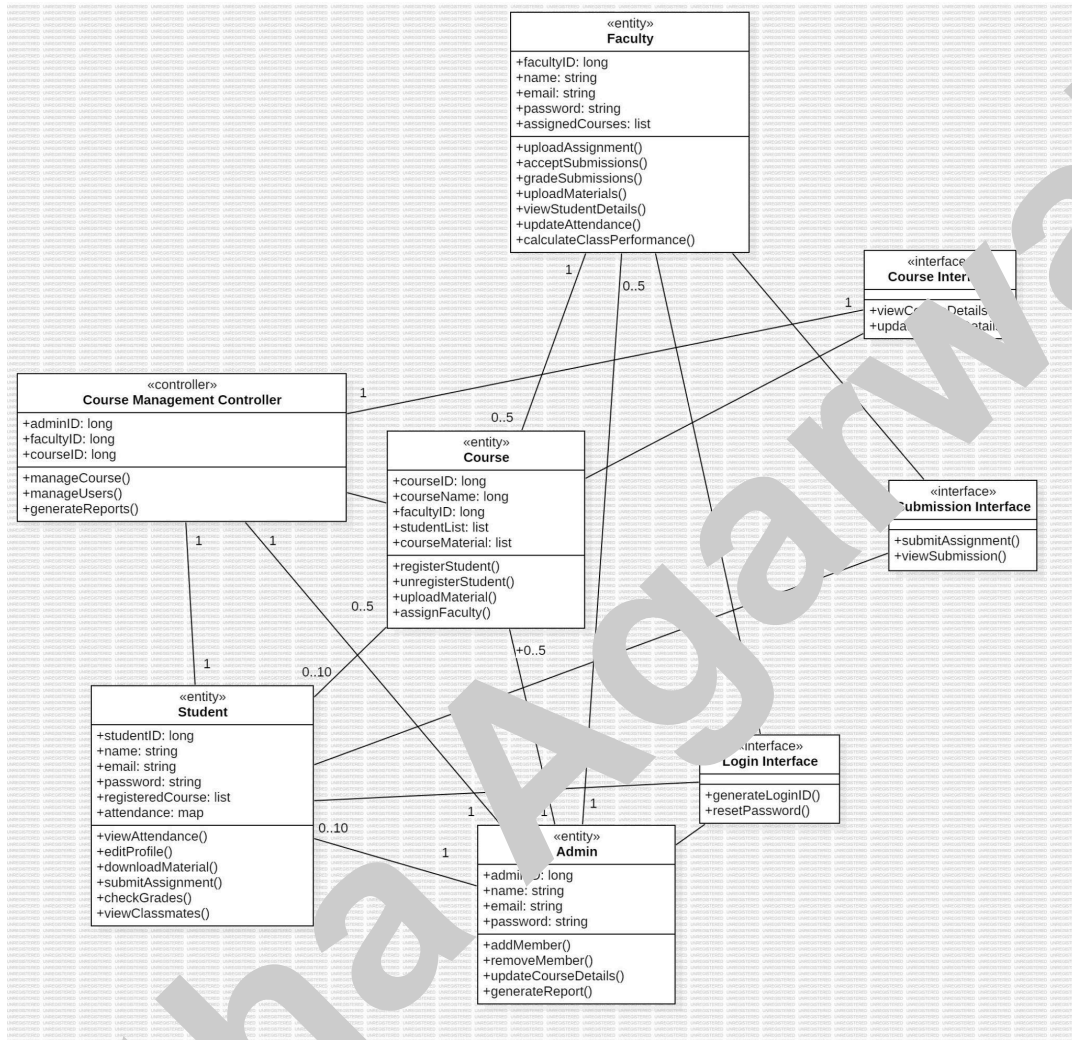
This allows the user to exit at any time during the use case. The use case ends.

Special requirements: None

Associated use cases: Login

EXPERIMENT-6

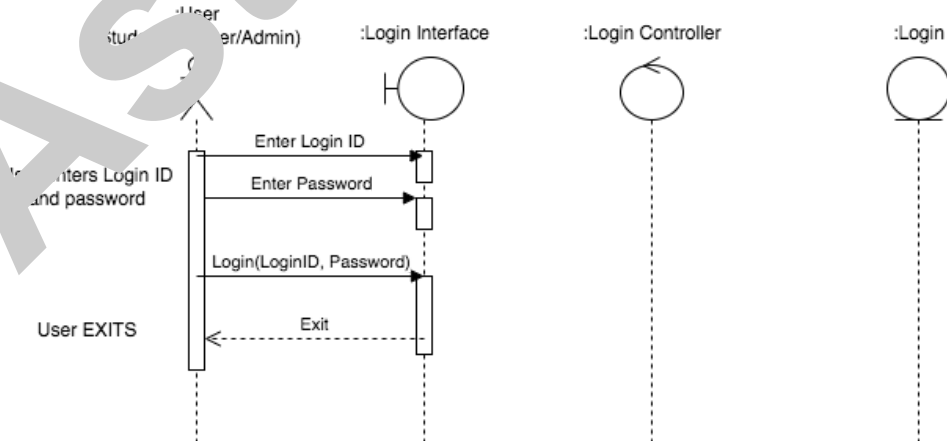
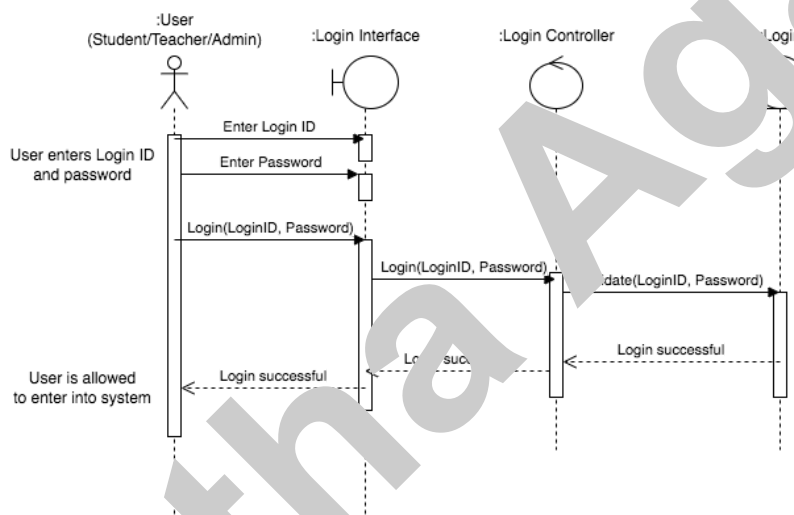
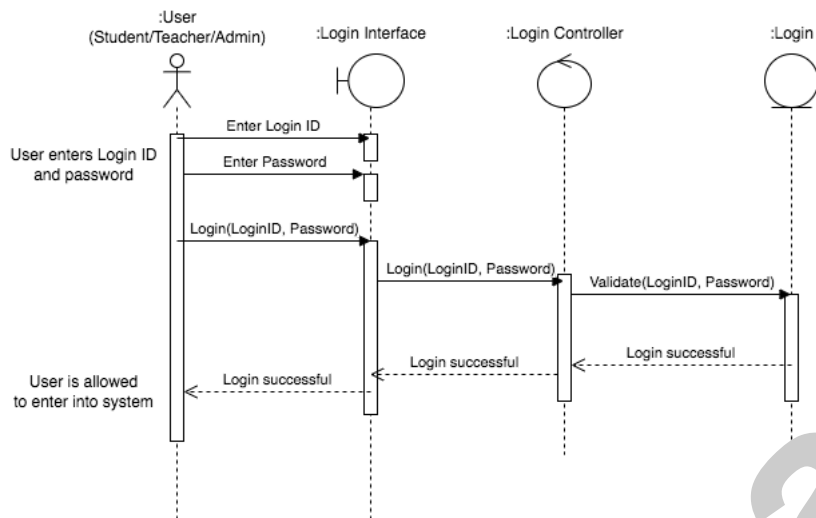
Aim: To design the class diagram of course management system



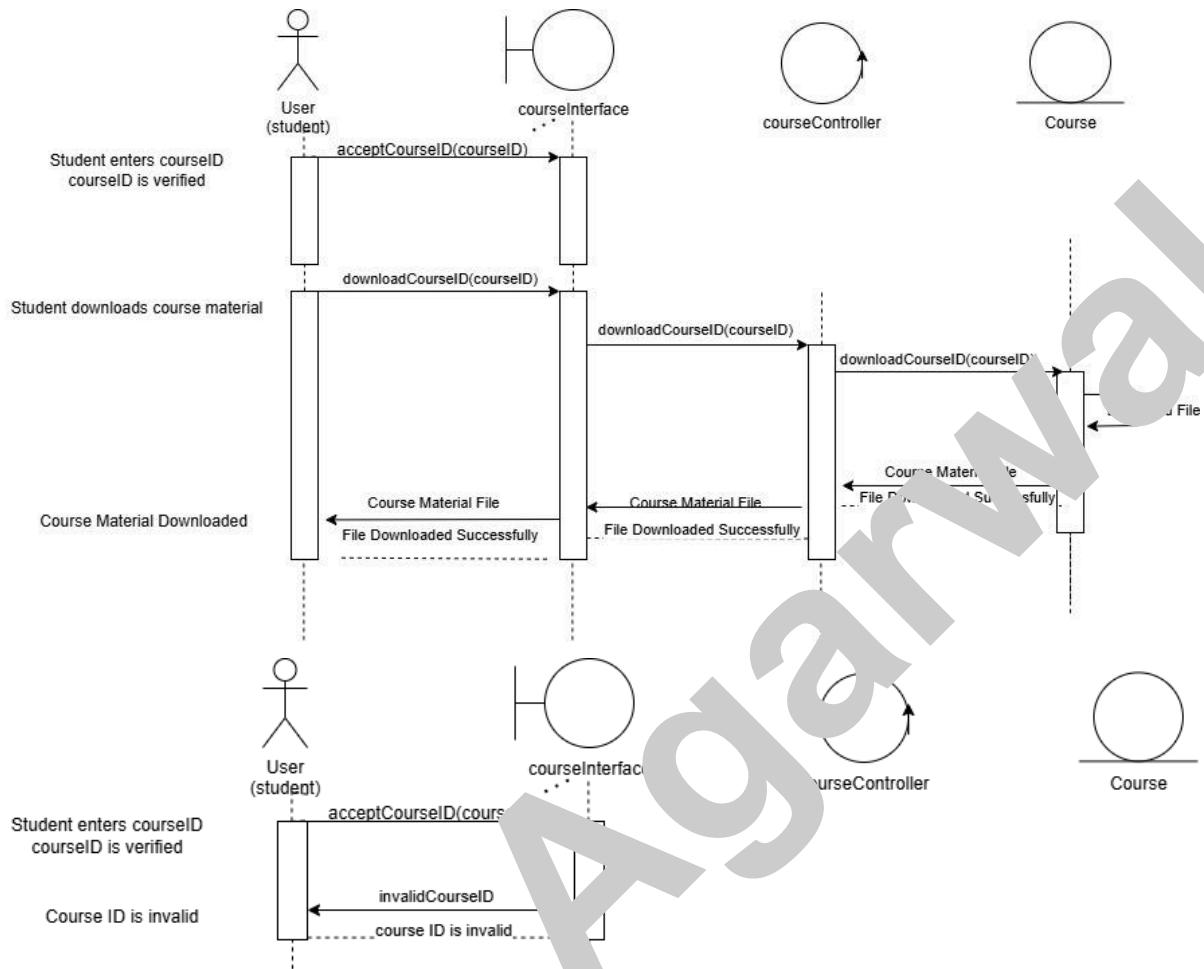
EXPERIMENT-7

Aim: To design the sequence diagrams of course management systems

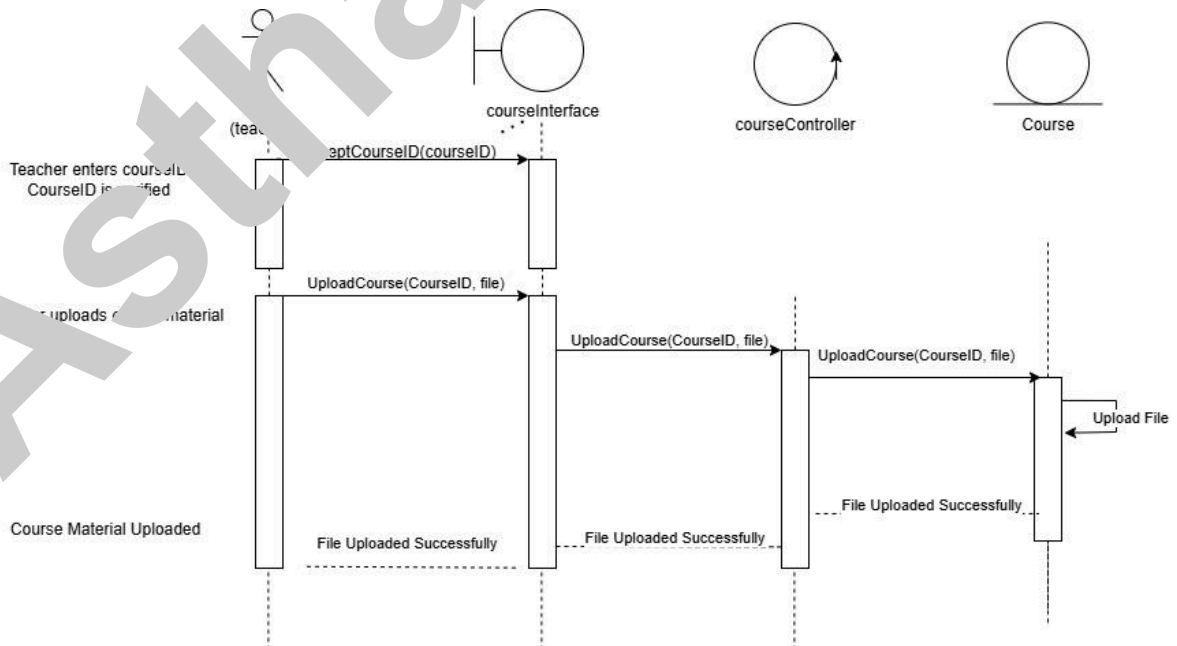
1. Login

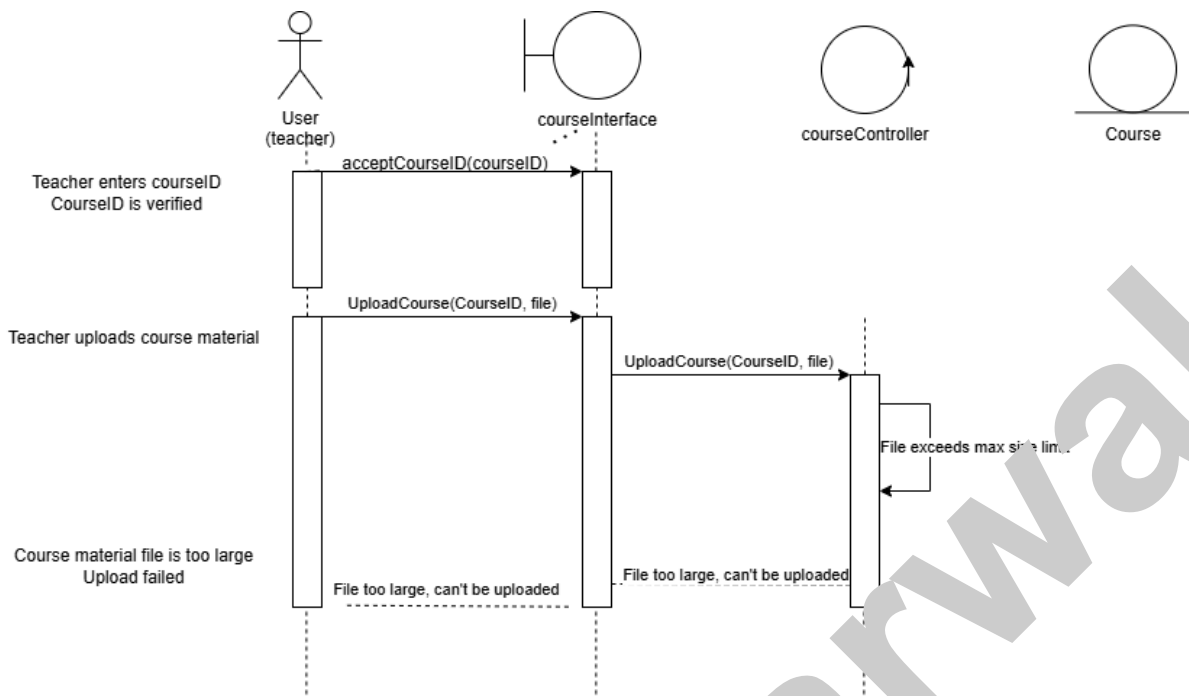


2. Downloading course material

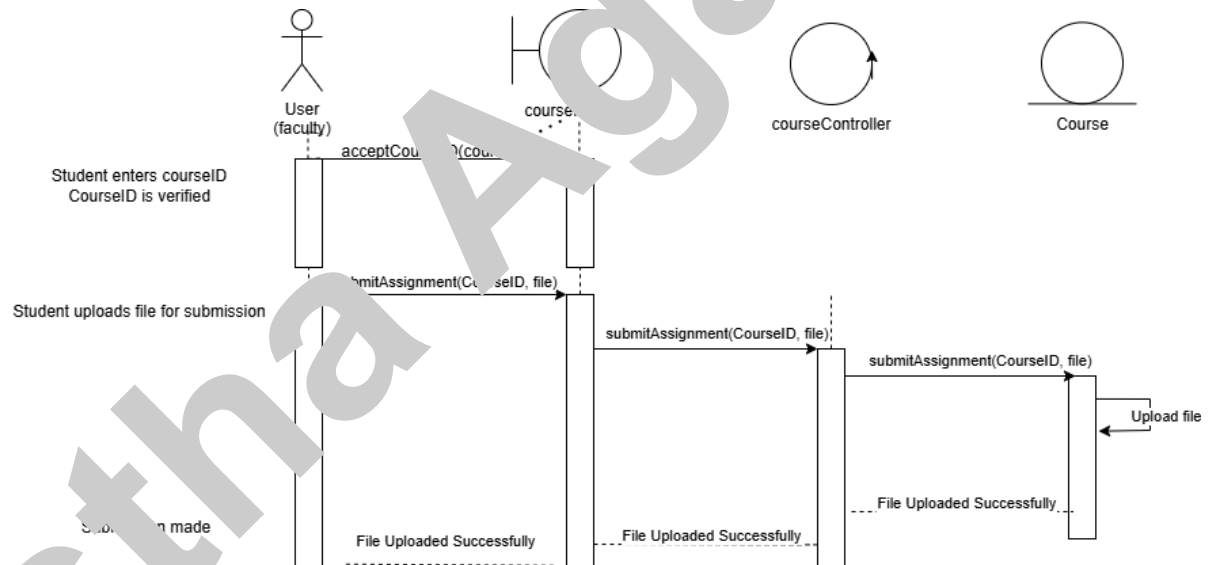


3. Uploading course material

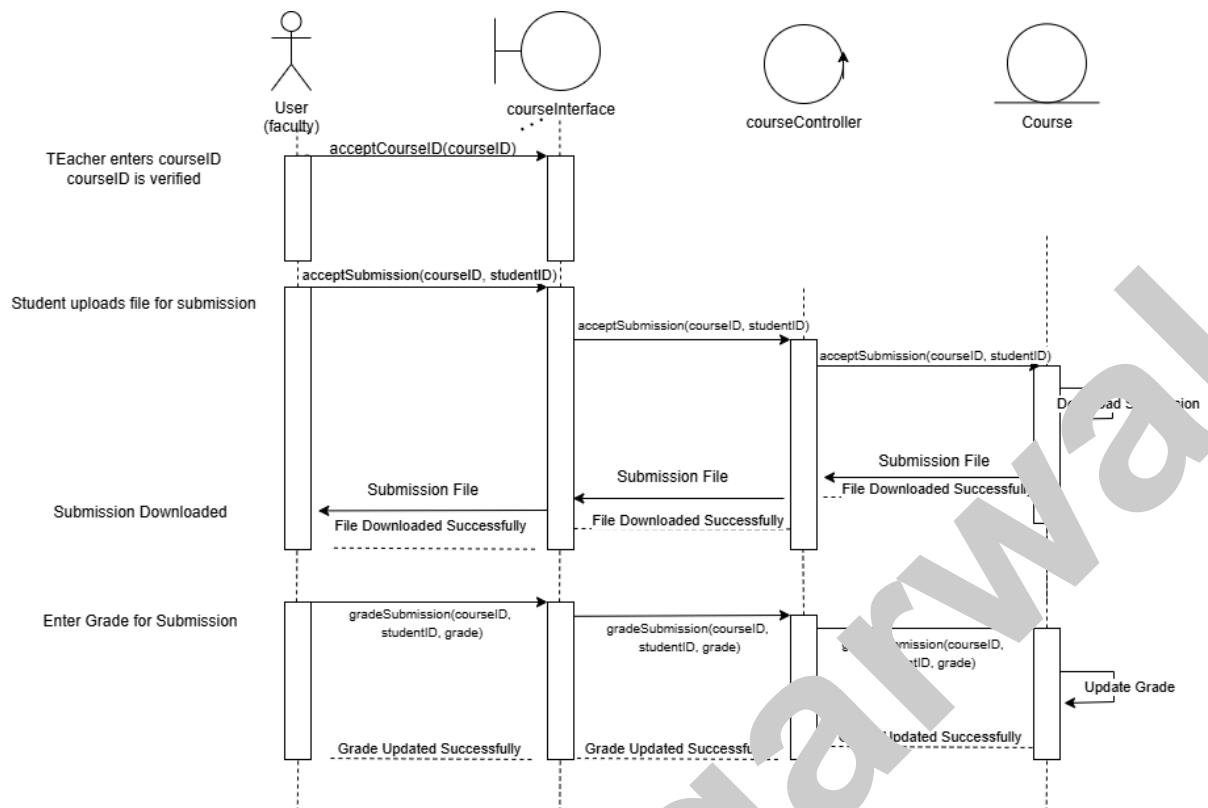




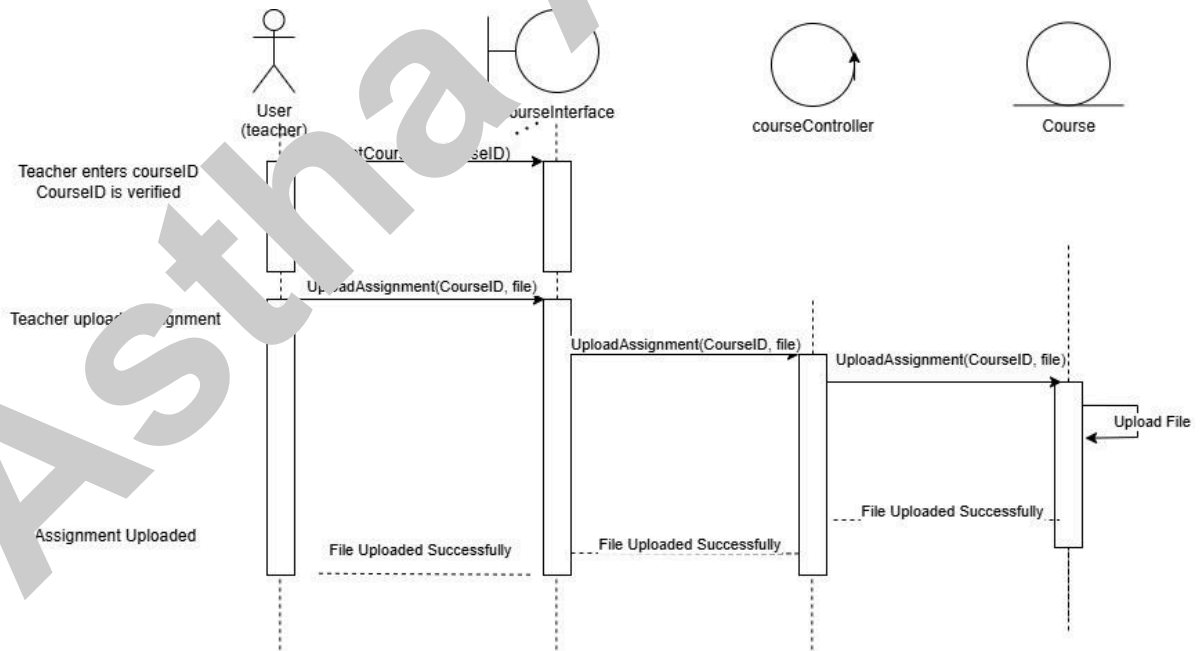
4. Make submission



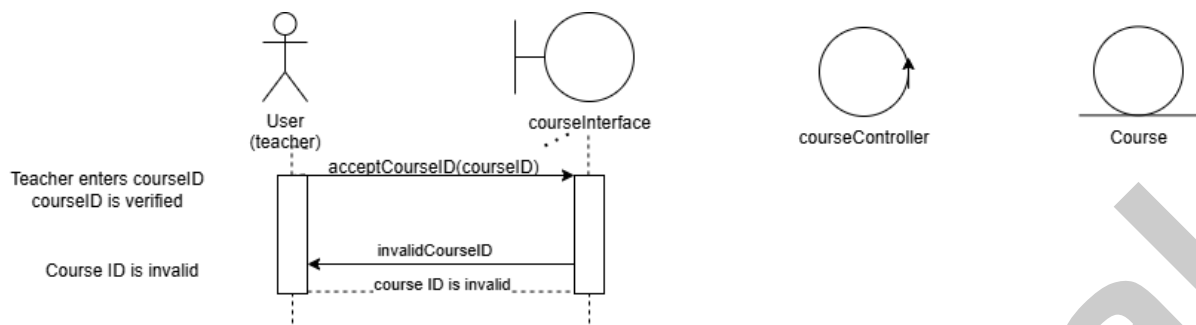
5. Accept and grade submissions



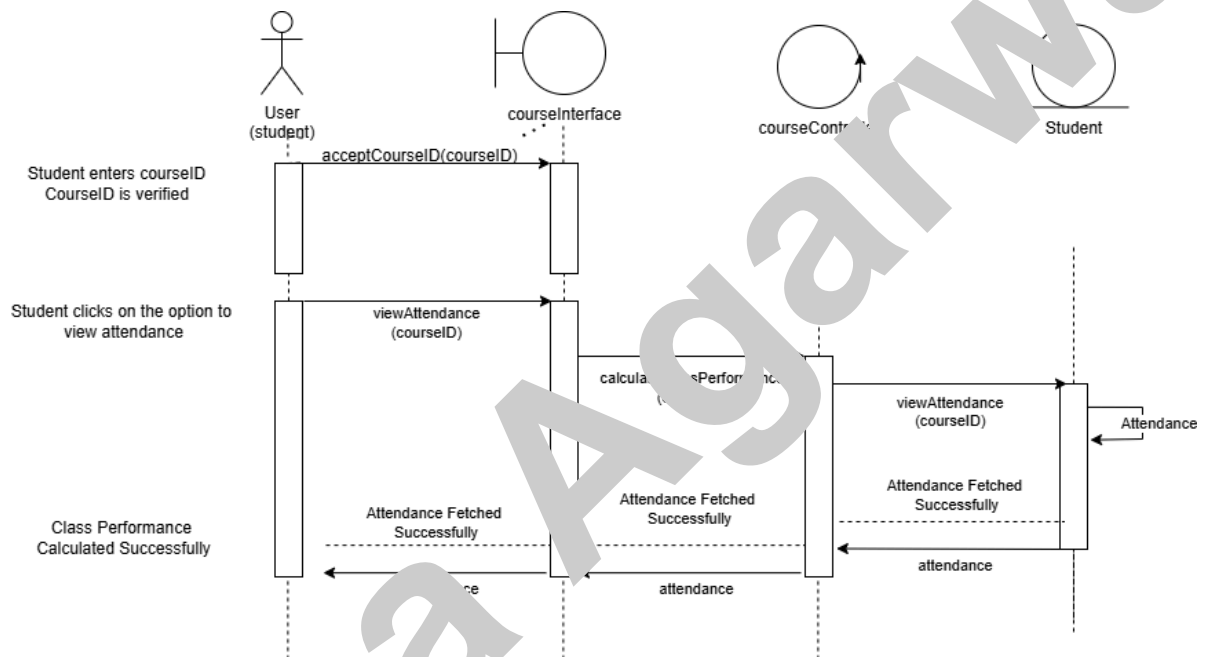
6. Upload assignments



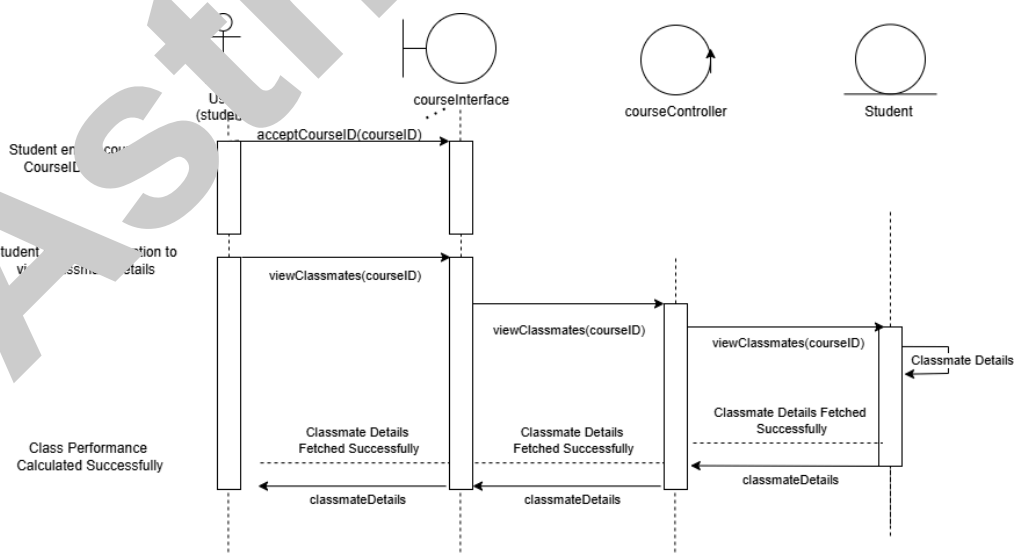
7. Calculate average performance



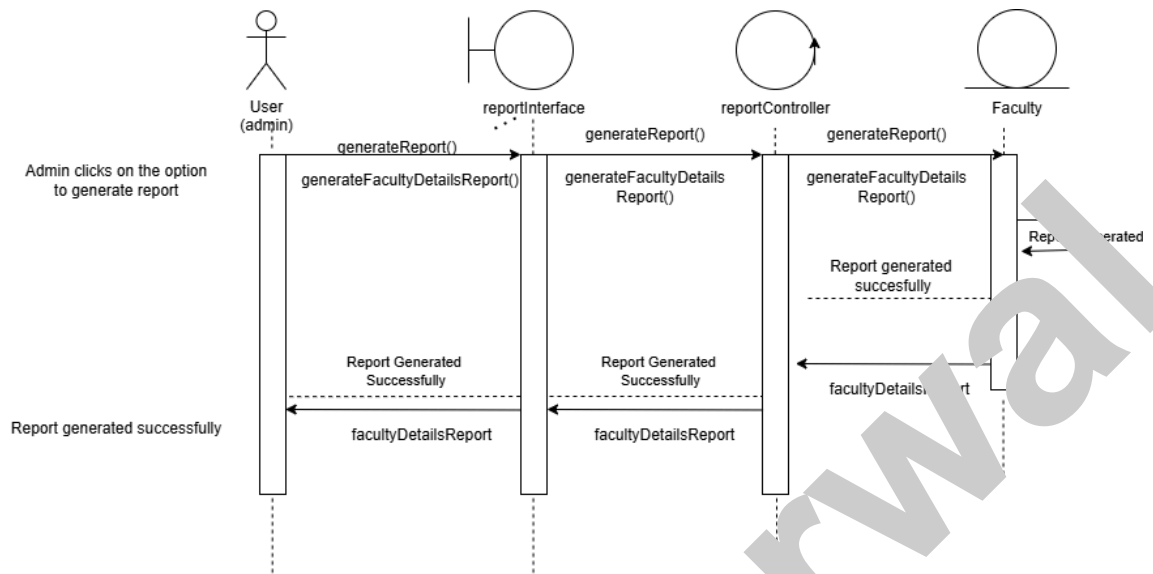
8. View attendance



9. View classmate details

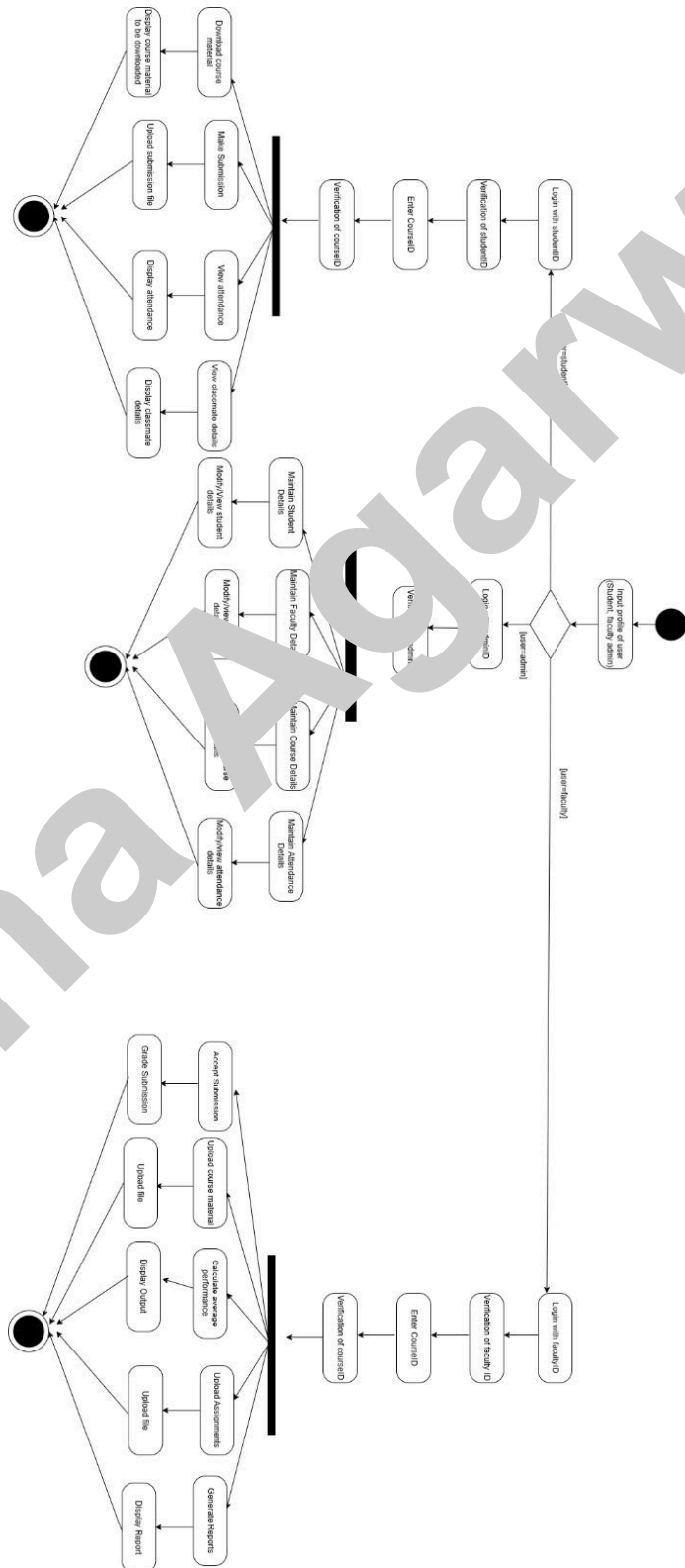


10. Generate reports



EXPERIMENT-8

Aim: To design the activity diagram of the course management system



EXPERIMENT-9

Aim: To design the test case matrices of course management system

1. Maintain student details

Test Case ID	Scenario and Description	Student ID	Name	Class	Section	Phone	Address	Mother's Name	Father's Name	Email	Update Confirmed	Deletion Confirmed	Expected Result	Remarks (if any)
TC ₁	Add a Student (All valid inputs)	Valid	Valid	Valid	Valid	Valid	Valid	Valid	Valid	Valid	n/a	n/a	Student is added successfully	-
TC ₂	Add a Student (Invalid Student ID)	Invalid	Valid	Valid	Valid	Valid	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Student ID format incorrect or missing
TC ₃	Add a Student (Invalid Name)	Valid	Invalid	Valid	Valid	Valid	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Name format incorrect or missing
TC ₄	Add a Student (Invalid Class)	Valid	Valid	Invalid	Valid	Valid	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Class field missing or incorrect
TC ₅	Add a Student (Invalid Section)	Valid	Valid	Valid	Invalid	Valid	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Section field missing or incorrect
TC ₆	Add a Student (Invalid Phone)	Valid	Valid	Valid	Valid	Invalid	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Phone number format incorrect
TC ₇	Add a Student (Invalid Address)	Valid	Valid	Valid	Valid	Valid	Invalid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Address field missing
TC ₈	Add a Student (Invalid Mother's Name)	Valid	Valid	Valid	Valid	Valid	Valid	Invalid	Valid	Valid	n/a	n/a	Error message displayed	Mother's name missing
TC ₉	Add a Student (Invalid Father's Name)	Valid	Valid	Valid	Valid	Valid	Valid	Valid	Invalid	Valid	n/a	n/a	Error message displayed	Father's name missing
TC ₁₀	Add a Student (Invalid Email)	Valid	Valid	Valid	Valid	Valid	Valid	Valid	Valid	Invalid	n/a	n/a	Error message displayed	Email format incorrect
TC ₁₁	Add a Student (Student Already Exists)	Existing Student ID	Valid	Valid	Valid	Valid	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Student ID already exists
TC ₁₂	Update a Student (All valid inputs)	Valid	Valid	Valid	Valid	Valid	Valid	Valid	Valid	Valid	Yes	n/a	Student details updated successfully	-
TC ₁₃	Update a Student (Invalid Student ID)	Invalid	Valid	Valid	Valid	Valid	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Student ID format incorrect or missing
TC ₁₄	Update a Student (Student Not Found)	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Error message displayed	No record found with this Student ID
TC ₁₅	Update a Student (Update Cancelled)	Valid	Valid	Valid	Valid	Valid	Valid	Valid	Valid	Valid	No	n/a	Main screen appears	Update operation cancelled
TC ₁₆	Delete a Student	Valid	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Yes	Student record deleted	-
TC ₁₇	Delete a Student (Student Not Found)	Non-existent Student ID	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Error message displayed	No record found with this Student ID

TC ₁₈	Delete a Student (Delete Cancelled)	Valid	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	No	Main screen appears	Delete operation cancelled	
TC ₁₉	View a Student	Valid	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Student details displayed	-
TC ₂₀	View a Student (Student Not Found)	Non-existent Student ID	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Error message displayed	No record found with this Student ID

2. Maintain faculty details

Test Case ID	Scenario and Description	Faculty ID	Name	Department	Designation	Phone	Address	Email	Update Confirmed	Deletion Confirmed	Expected result	Remarks (if any)
TC ₁	Add a Faculty Member (All valid inputs)	Valid	Valid	Valid	Valid	Valid	Valid	Valid	n/a	n/a	Faculty member is added successfully	-
TC ₂	Add a Faculty Member (Invalid Faculty ID)	Invalid	Valid	Valid	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Faculty ID format incorrect or missing
TC ₃	Add a Faculty Member (Invalid Name)	Valid	Invalid	Valid	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Name format incorrect or missing
TC ₄	Add a Faculty Member (Invalid Department)	Valid	Valid	Invalid	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Department field missing or incorrect
TC ₅	Add a Faculty Member (Invalid Designation)	Valid	Valid	Valid	Invalid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Designation field missing or incorrect
TC ₆	Add a Faculty Member (Invalid Phone)	Valid	Valid	Valid	Valid	Invalid	Valid	Valid	n/a	n/a	Error message displayed	Phone number format incorrect
TC ₇	Add a Faculty Member (Invalid Address)	Valid	Valid	Valid	Valid	Valid	Invalid	Valid	n/a	n/a	Error message displayed	Address field missing
TC ₈	Add a Faculty Member (Invalid Email)	Valid	Valid	Valid	Valid	Valid	Valid	Invalid	n/a	n/a	Error message displayed	Email format incorrect
TC ₉	Add a Faculty Member (Duplicate Faculty ID)	Valid	Valid	Valid	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Faculty ID already exists
TC ₁₀	Update a Faculty Member (All valid inputs)	Valid	Valid	Valid	Valid	Valid	Valid	Valid	Yes	n/a	Faculty details updated successfully	-
TC ₁₁	Update a Faculty Member (Invalid Faculty ID)	Invalid	Valid	Valid	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Faculty ID format incorrect or missing

TC ₁₂	Update a Faculty Member (Faculty Not Found)	Non-existent Faculty ID	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Error message displayed	No record found with this Faculty ID
TC ₁₃	Update a Faculty Member (Update Cancelled)	Valid	Valid	Valid	Valid	Valid	Valid	Valid	No	n/a	Main screen appears	Update operation cancelled
TC ₁₄	Delete a Faculty Member	Valid	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Yes	Faculty details deleted	-
TC ₁₅	Delete a Faculty Member (Faculty Not Found)	Non-existent Faculty ID	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Error message displayed	No record found with this Faculty ID
TC ₁₆	Delete a Faculty Member (Delete Cancelled)	Valid	n/a	n/a	n/a	n/a	n/a	n/a	n/a	No	Main screen appears	Delete operation cancelled
TC ₁₇	View a Faculty Member	Valid	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Faculty details displayed	-
TC ₁₈	View a Faculty Member (Faculty Not Found)	Non-existent Faculty ID	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Error message displayed	No record found with this Faculty ID

3. Maintain course details

Test Case ID	Scenario and Description	Course ID	Course Name	Department	Credits	Schedule	Update Confirmed	Deletion Confirmed	Expected Result	Remarks (if any)
TC ₁	Add a Course (All valid inputs)	Valid	Valid	Valid	Valid	Valid	n/a	n/a	Course is added successfully	-
TC ₂	Add a Course (Invalid Course ID)	Invalid	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Course ID format incorrect or missing
TC ₃	Add a Course (Invalid Course Name)	Valid	Invalid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Course Name format incorrect or missing
TC ₄	Add a Course (Invalid Department)	Valid	Valid	Invalid	Valid	Valid	n/a	n/a	Error message displayed	Department field missing or incorrect
TC ₅	Add a Course (Invalid Credits)	Valid	Valid	Valid	Invalid	Valid	n/a	n/a	Error message displayed	Credits field missing or incorrect
TC ₆	Add a Course (Invalid Instructor)	Valid	Valid	Valid	Valid	Invalid	n/a	n/a	Error message displayed	Instructor field missing or incorrect
TC ₇	Add a Course (Invalid Schedule)	Valid	Valid	Valid	Valid	Invalid	n/a	n/a	Error message displayed	Schedule field missing or incorrect
TC ₈	Add a Course (Course Already Exists)	Existing Course ID	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Course ID already exists
TC ₉	Update a Course (All valid inputs)	Valid	Valid	Valid	Valid	Valid	Yes	n/a	Course details updated successfully	-
TC ₁₀	Update a Course (Invalid Course ID)	Invalid	Valid	Valid	Valid	Valid	n/a	n/a	Error message displayed	Course ID format incorrect or missing

TC ₁₁	Update a Course (Course Not Found)	Non-existent Course ID	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Error message displayed	No record found with this Course ID
TC ₁₂	Update a Course (Update Cancelled)	Valid	Valid	Valid	Valid	Valid	Valid	No	n/a	Main screen appears	Update operation cancelled
TC ₁₃	Delete a Course	Valid	n/a	n/a	n/a	n/a	n/a	n/a	Yes	Course record deleted	-
TC ₁₄	Delete a Course (Course Not Found)	Non-existent Course ID	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Error message displayed	No record found with this Course ID
TC ₁₅	Delete a Course (Delete Cancelled)	Valid	n/a	n/a	n/a	n/a	n/a	n/a	No	Main screen appears	Delete operation cancelled
TC ₁₆	View a Course	Valid	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Course details display	-
TC ₁₇	View a Course (Course Not Found)	Non-existent Course ID	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Error message displayed	No record found with this Course ID