



# Assignment 2

09.03.2025

---

CS-322 SC

Ahmed Nazir (04072312025)

Ahsen Rasheed (04072312012)

Muhammad Yamaan (04072312021)

Submitted To:  
Dr. Onaiza Maqbool

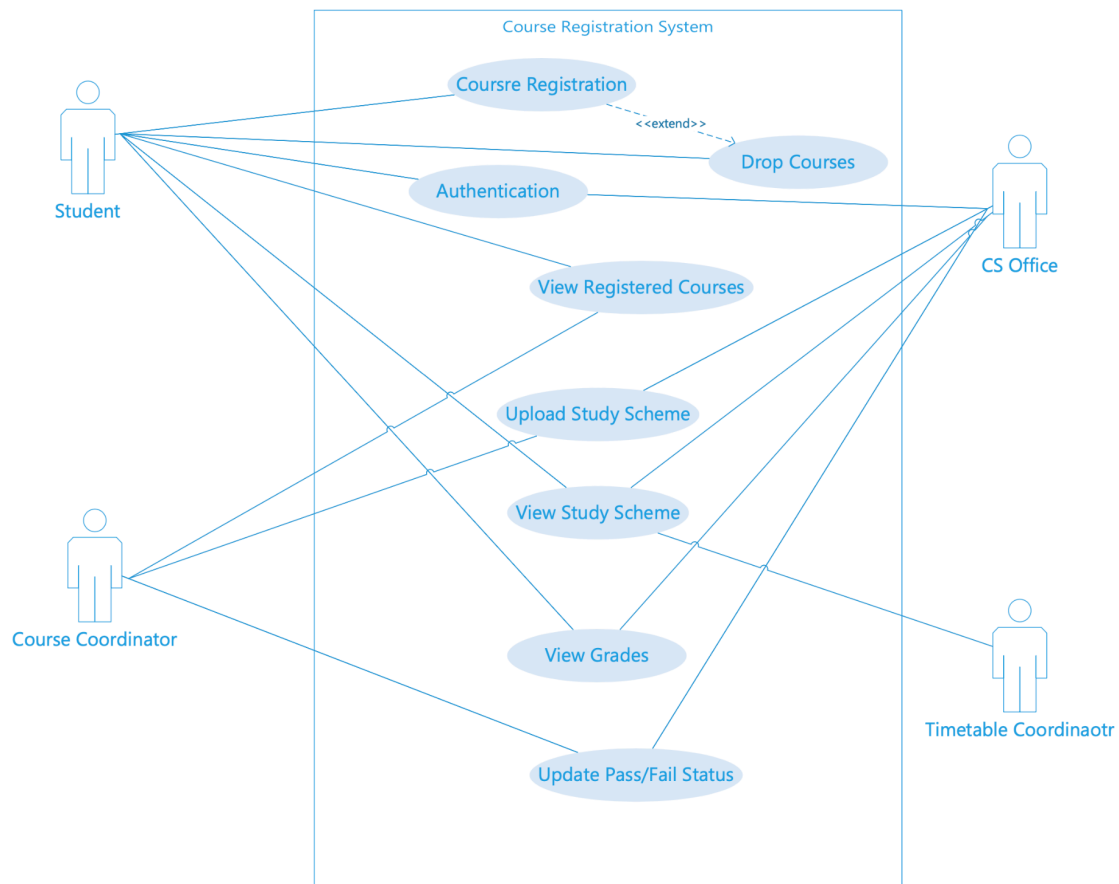
## Overview

This document describes the use case diagram for a Student Course Registration System. It explains the interactions between various users also called actors and the functionalities of our system that are use cases.

## Use case diagram

### Actors:

1. Student
2. Course coordinator
3. Timetable coordinator
4. CS department



## Use Case Descriptions

### Authentication

**Actor:** Student, CS Office

**Description:** This is how users login to the system.

**Working:**

1. The user enters their username and password.
2. The system validates the credentials against the stored data.
3. If the credentials are valid, the user is granted access to the system.
4. If the credentials are invalid, an error message is displayed, and the user is prompted to re-enter their credentials.

**Exceptions:**

- **Invalid Credentials:** If the user enters incorrect credentials multiple times, the system may temporarily lock the account to prevent unauthorized access.
- **Forgotten Password:** The system provides a "forgotten password" functionality that allows the user to reset their password through a secure process, such as email verification.

## Course Registration

**Actor:** Student, CS Office

**Description:** This use case allows students to register for courses.

**Working:**

1. The student logs into the system.
2. The student views the list of available courses for the current semester.
3. The student selects the courses they want to register for.
4. The system checks for any prerequisites or restrictions for the selected courses.
5. If the student meets the requirements, the system registers the student for the selected courses.
6. The system then updates the student's record.

**Exceptions:**

- **Credit Hours Full:** If a student meets 18 credit hours, the system notifies the student that you can't register more than 18.
- **Clashes:** If the selected courses have overlapping schedules, the system tells the student that they can't register their respective course due to classes at the same time.
- **Prerequisite Not Met:** If the student has not fulfilled the prerequisites for a course, the system prevents registration and informs the student of the missing requirements.

## Remove/Drop Courses

**Actor:** Student, CS Office

**Description:** This use case allows students to drop or remove courses they have registered for in the current semester.

**Working:**

1. The student logs into the system.
2. The student views their list of registered courses for the current semester.
3. The student selects the course they want to remove.
4. The system checks if the removal is possible (eg. within the deadline etc).
5. If possible, the system removes the student from the course.
6. The system then updates the student's record.

**Exceptions:**

- **Removal Deadline Passed:** If the deadline for course removal has passed, the system prevents the removal and notifies the student.
- **Remove With Fine:** If removing the course requires fine, the system informs the student and requires the confirmation.

## View Registered Courses

**Actor:** Student, Course Coordinator

**Description:** This use case allows students and course coordinators to view the courses registered by students.

**Working:**

1. The user (student or course coordinator) logs into the system.
2. The system displays the current semester (Fall or Spring).
3. The system displays the list of courses the student is registered for.
4. The user can view course details, such as schedules, instructors, and locations.

## Upload Study Scheme

**Actor:** Course Coordinator, CS Office

**Description:** This use case allows the course coordinator and the CS Office to upload the study scheme or curriculum for courses in the current semester.

**Basic Flow:**

1. The user (course coordinator or CS Office staff) logs into the system.
2. The user selects the course for which to upload the study scheme.
3. The user uploads the study scheme document, specifying the current semester.

## View Study Scheme

**Actor:** Course Coordinator

**Description:** This use case allows students to view the study scheme for their registered courses in the current semester.

**Working:**

1. The student logs into the system.
2. The student selects the course for which they want to view the study scheme.
3. The system retrieves and displays the study scheme document for the current semester.

## Update Pass/Fail Status

**Actor:** Course Coordinator, CS Office

**Description:** This use case allows the course coordinator and the CS Office to update the pass/fail status of students for a course in the current semester.

**Working:**

1. The user (course coordinator or CS Office staff) logs into the system.
2. The user selects the relevant course and the student.
3. The user updates the pass/fail status for the selected student for the current semester.
4. The system updates the student's record with the new status.

## View Grades

**Actor:** Student, CS Office

**Description:** This use case allows students and the CS Office to view student grades..

**Working:**

1. The user (student or CS Office staff) logs into the system.
2. The user selects the course/semester for which they want to view grades.
3. The system retrieves and displays the student's grades for that course/semester..

## Conclusions

This document provides a comprehensive overview of the use cases for the Student Course Registration System by explaining the interactions between various actors (Student, Course Coordinator, Timetable Coordinator, CS Office) and the system's functionalities. It provides a valuable resource for understanding the system's capabilities.