



**Department of Computer Science**

**Quaid-e-Azam University, Islamabad**

**Assignment 3**

**Student Course Registration System**

Prepared by:

**HURR MEHDI**

**M. SEHAL BILAL**

**M. ASAD (Leader)**

**ZOHAIB AHMED**

Submitted to:

**DR. ONAIZA MAQBOOL**

17 March 2025

|  |  |
| --- | --- |
| Use case Description & System Sequence Diagram | **Student Course Registration System** |
| Created by | Hurr Mehdi (04072312048) |
| Date Created | 17-03-2025 |

## Use Case: View Academic Progress

**Primary Actor(s):**

* *Student*

**Use Case Description:**

*This use case allows a student to view their academic progress, including courses they have passed, failed, and pending requirements.*

**Stakeholders and Interests:**

* *Student: Wants to track their academic performance and plan future registrations.*
* *University Administration: Ensures accurate record-keeping and compliance with academic policies.*
* *Course Coordinator: Uses progress data to advise students and update curriculum requirements.*

**Preconditions:**

* *The student must be logged into the system.*
* *The student’s academic records (course registrations, grades) must exist in the system.*

**Postconditions:**

* *The student’s academic progress is displayed.*
* *No changes are made to the database (read-only action).*

**Inputs:**

* *StudentID*

**Outputs:**

* *AcademicProgressDetails (passed courses, failed courses, pending prerequisites, GPA).*

**Main Success Scenario:**

* *The student navigates to the Academic Progress section.*
* *The system retrieves the student’s academic records (registrations, grades, prerequisites).*
* *The system generates a summary of:*
* *Passed Courses: Courses marked as "Pass."*
* *Failed Courses: Courses marked as "Fail."*
* *Pending Requirements: Prerequisites not yet completed.*
* *The system displays the academic progress summary to the student in a tabular or visual format.*
* *The student reviews the information and exits the section.*

**Alternative Scenarios:**

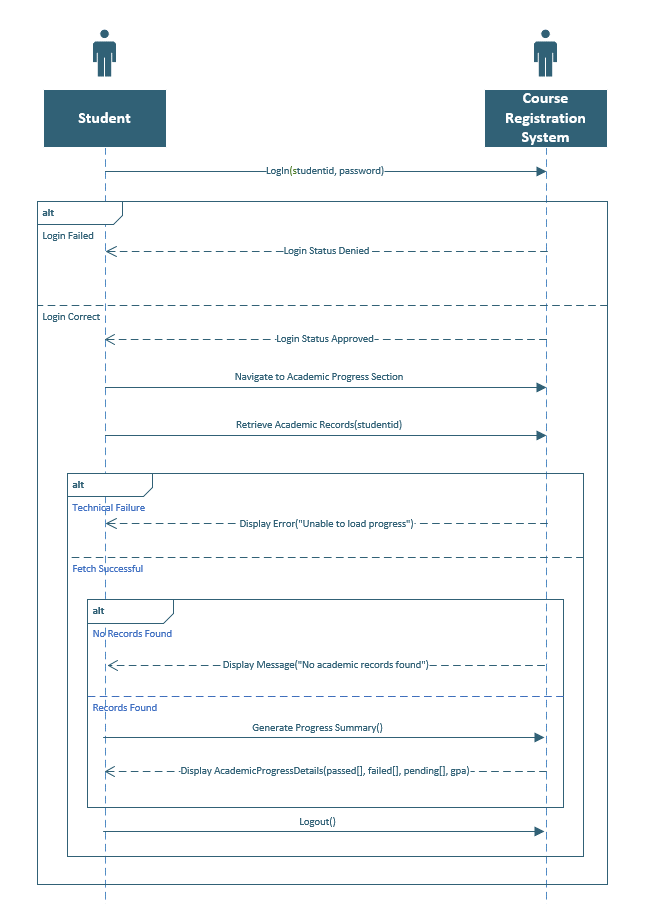
* *No Academic Records Found:*

*The system displays a message: “No academic records available. Please register for courses first.”*

* *Technical Failure:*

*If the system fails to retrieve data, an error message is displayed: “Unable to load academic progress. Please try again later.”*

**Sequence Diagram: View Academic Progress**



## UML Diagram

|  |  |
| --- | --- |
|  | The purpose of this use case diagram is to provide a visual representation of the functional requirements and interactions between users (actors) and the Student Course Registration System, as outlined in Case Study #3. It identifies key actors, their primary use cases, and the relationships between them.This use case diagram was designed using Argo UML, a UML modeling tool, to adhere to ensure clarity in system design. It serves as a foundational blueprint for upcoming development phases. |

## 

## Actors and Use Cases

|  |  |
| --- | --- |
| Use case Description & System Sequence Diagram | **Student Course Registration System** |
| Created by | M. Sehal Bilal (04072312015) |
| Date Created | 17-03-2025 |

## Use Case: Register Courses

**Primary Actor(s):**

* *Student*

**Use Case Description:**

*This use case allows a student to register for courses during the course registration period. The system ensures that the student meets prerequisites before enrolling them in a course.*

**Stakeholders and Interests:**

* *Student: Wants to successfully register for courses without conflicts.*
* *University Administration: Ensures students enroll in the correct courses.*
* *Course Coordinator: Monitors course enrollments and prerequisite compliance.*
* *Timetable Coordinator: Manages scheduling conflicts and class capacities.*

**Preconditions:**

* *The student must be logged into the system.*
* *The course registration period must be active.*

**Postconditions:**

* *The registered course is stored in the database.*
* *The system updates the student's registered courses.*

**Inputs:**

* *StudentID*
* *CourseID*

**Outputs:**

* *RegistrationStatus (Confirmed or Denied)*
* *UpdatedCourseList*

**Main Success Scenario:**

* *The student navigates to the course registration section.*
* *The student enters the course ID for the desired course.*
* *The system checks if the student meets the course prerequisites.*
* *If prerequisites are met, the system registers the student.*
* *The system updates the student’s registered courses.*
* *The system confirms the registration to the student by displaying a ✔ (checkmark) or 'True' status next to the registered course ID.*

**Alternative Scenarios:**

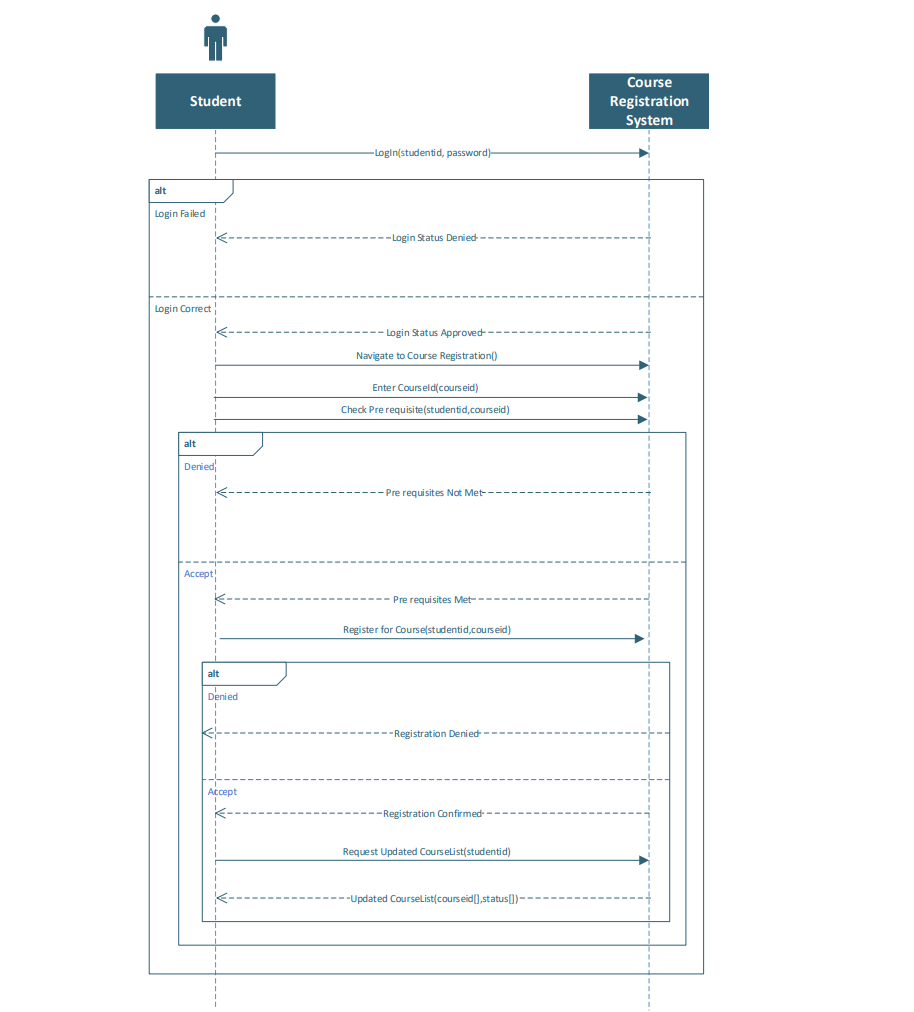
* *Prerequisites Not Met*

*The system notifies the student that they do not meet the prerequisites.The student is unable to register for the course.*

* *Technical Failure*

*If the system crashes or encounters an error, the student cannot complete registration.The system logs the failure, and the student is advised to retry later.*

**Sequence Diagram: Register Courses**



|  |  |
| --- | --- |
| Use case Description & System Sequence Diagram | **Student Course Registration System** |
| Created by | Zohaib Ahmed (04072213001) |
| Date Created | 17-03-2025 |

## Use Case: View Course Registrations

**Primary Actor(s):**

* *Timetable Coordinator*

**Use Case Description:**

*This use case allows the timetable coordinator to view student course registrations to assist in class scheduling and avoid conflicts. The system retrieves and displays enrollment data for different courses.*

**Stakeholders and Interests:**

* *Timetable Coordinator: Needs registration data to plan class schedules efficiently.*
* *Students: Benefit from properly managed class schedules with no conflicts.*
* *Course Coordinator: May need to adjust course offerings based on registration trends.*

**Preconditions:**

* *The timetable coordinator must be logged into the system.*
* *Course registration data must be available.*

**Postconditions:**

* *The system displays up-to-date student enrollment data for different courses.*

**Inputs:**

* *Semester*
* *Course Name*

**Outputs:**

* *List of students registered for the course.*

**Main Success Scenario:**

* *The timetable coordinator navigates to the "View Course Registrations" section.*
* *The coordinator enters search criteria (e.g., semester or specific courses).*
* *The system retrieves and displays a list of students registered in each course.*
* *The timetable coordinator reviews the data for scheduling adjustments.*

**Alternative Scenarios:**

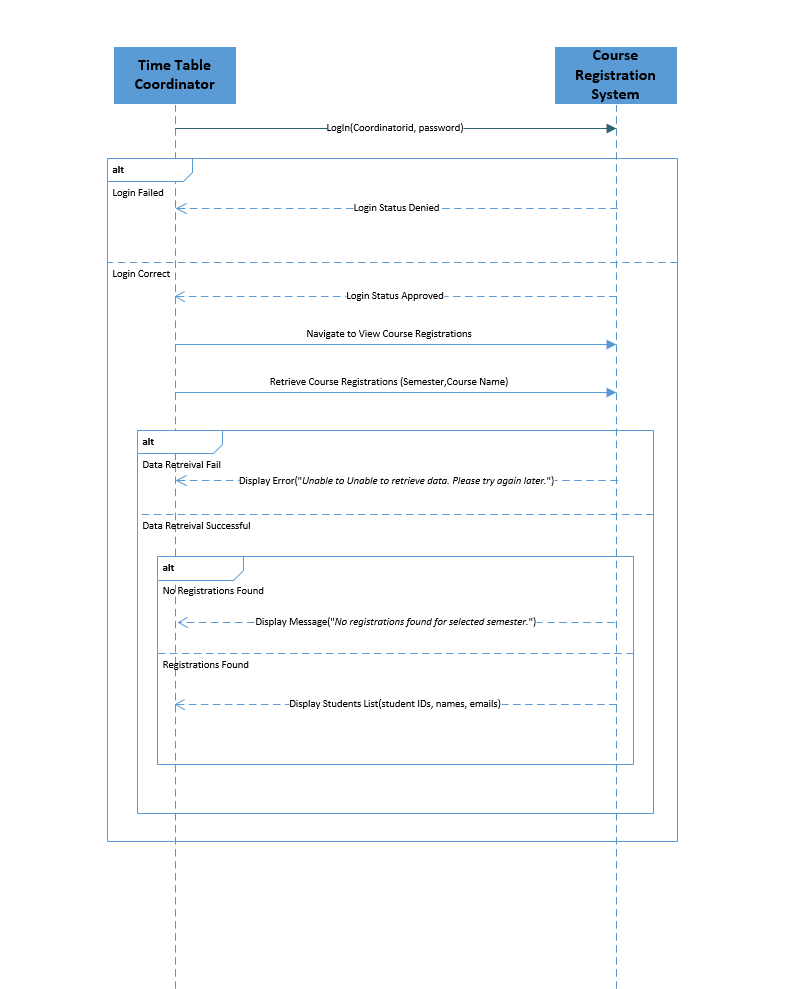
* *No Registrations Found*

*The system notifies the coordinator that no registrations exist for the selected criteria.*

* *Data Retrieval Error*

*The system displays an error message, and the coordinator is advised to try again later.*

**Sequence Diagram: View Course Registrations**



|  |  |
| --- | --- |
| Use case Description & System Sequence Diagram | **Student Course Registration System** |
| Created by | Muhammad Asad (04072312023) |
| Date Created | 17-03-2025 |

## Use Case: Edit Courses

**Primary Actor(s):**

* *Course Coordinator*

**Use Case Description:**

*This use case allows the Course Coordinator to edit the details of a course, such as course name, prerequisites, or other relevant information. The system ensures that the changes are validated and updated in the database.*

**Stakeholders and Interests:**

* *Course Coordinator: Wants to update course details accurately and efficiently.*
* *University Administration: Ensures that course information is up-to-date and accurate.*
* *Students: Rely on accurate course information for registration and academic planning.*
* *Timetable Coordinator: Needs updated course details to manage scheduling conflicts.*

**Preconditions:**

* *The Course Coordinator must be logged into the system.*
* *The course to be edited must exist in the system.*

**Postconditions:**

* *The course details are updated in the database.*
* *The system reflects the updated course information for all users.*

**Inputs:**

* *CourseID*
* *Updated course details (e.g., course name, prerequisites, etc.)*

**Outputs:**

* *EditStatus (Confirmed or Denied)*
* *UpdatedCourseDetails.*

**Main Success Scenario:**

* *The Course Coordinator navigates to the course management section.*
* *The Course Coordinator selects the course to be edited by entering the CourseID.*
* *The system retrieves the current course details.*
* *The Course Coordinator updates the course details (e.g., course name, prerequisites, etc.).*
* *The system validates the updated course details.*
* *If the validation is successful, the system updates the course details in the database.*

**Alternative Scenarios:**

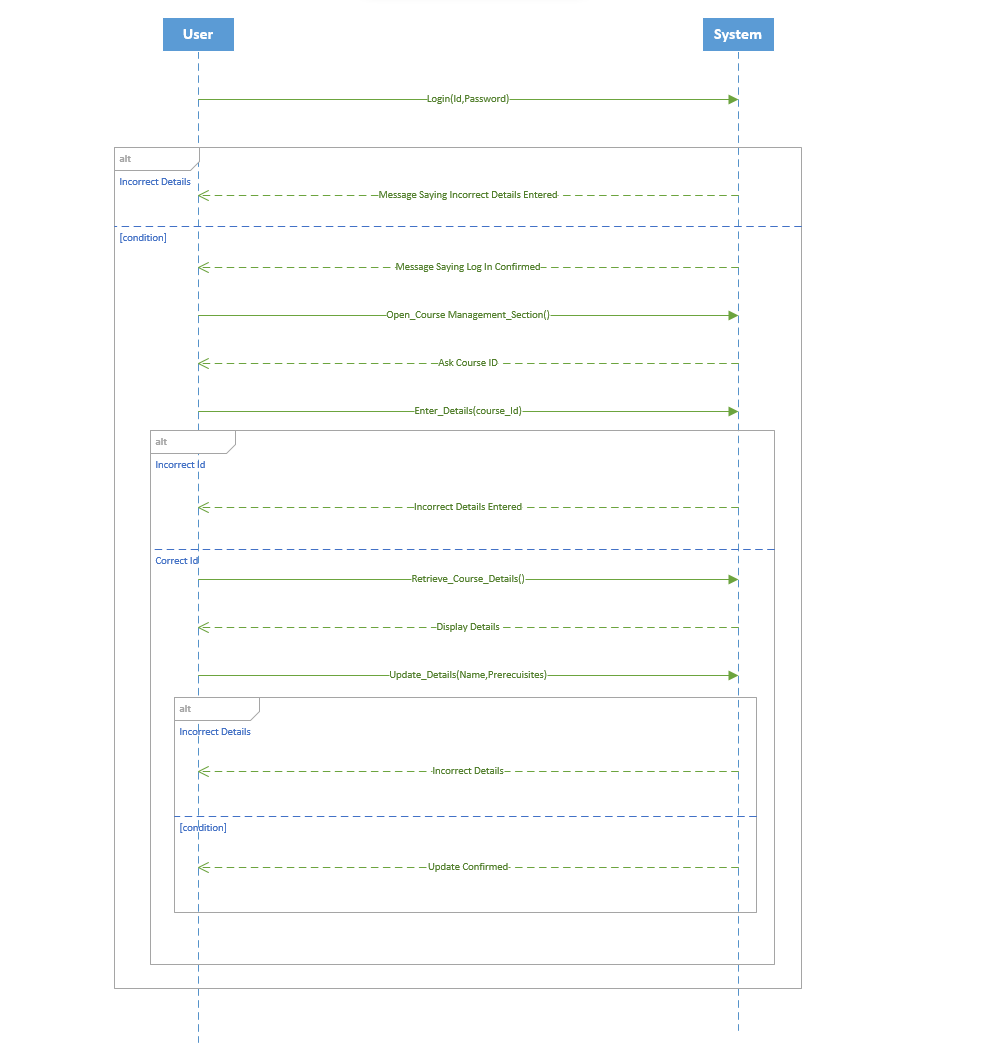
* *Validation Failure:*

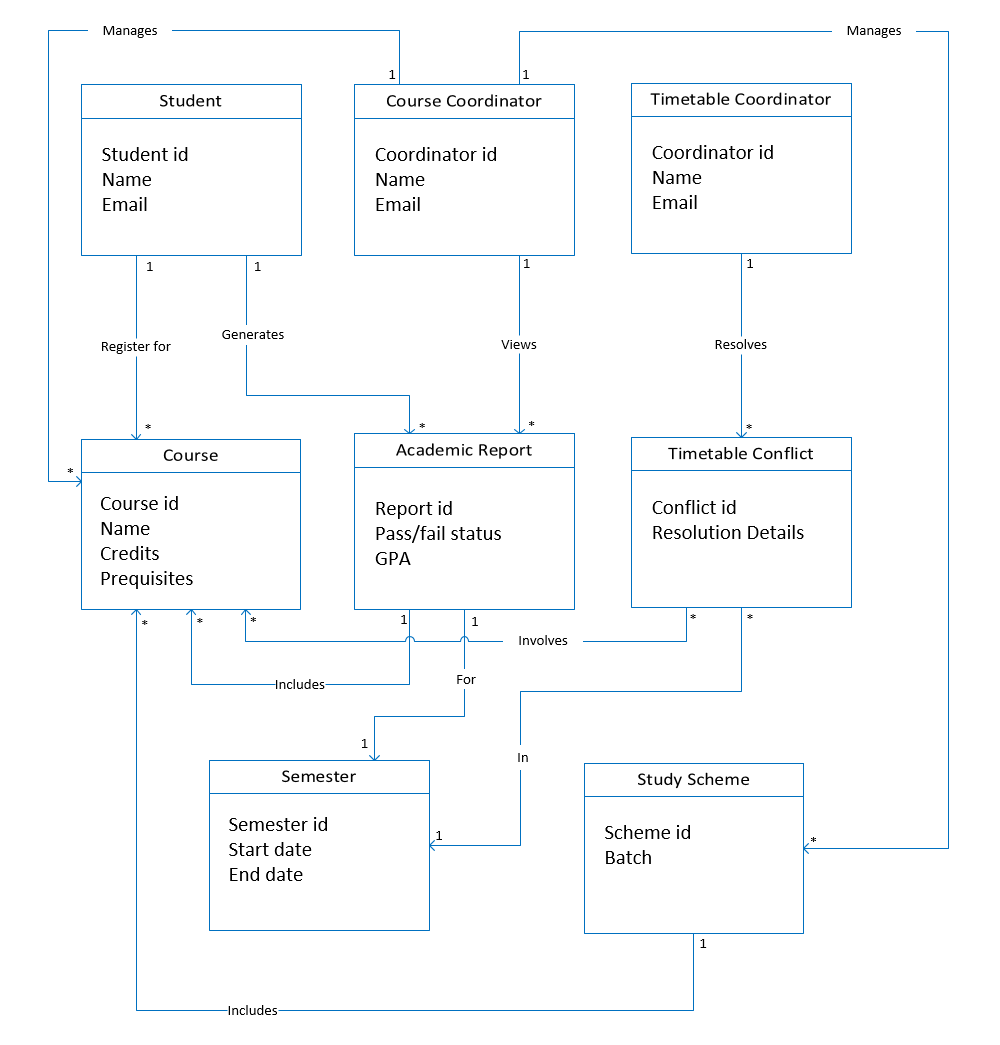
*If the updated course details fail validation (e.g., invalid prerequisites or missing information), the system notifies the Course Coordinator. The Course Coordinator is prompted to correct the details and resubmit.*

* *Technical Failure:*

*If the system crashes or encounters an error during the update process, the Course Coordinator cannot complete the edit. The system logs the failure, and the Course Coordinator is advised to retry later.*

**Sequence Diagram: View Academic Progress**





|  |  |
| --- | --- |
| Domain Model | **Student Course Registration System** |
| Created by | Group 8 |
| Date Created | 17-03-2025 |