

19TH MARCH 2025

ENROLLMATE

ONLINE COURSE REGISTRATION SYSTEM

Prepared by:
Zainab Fatima (22i-1064)
Shayane Zainab (22i-1049)

02

INTRODUCTION

EnrollMate is a course enrollment and registration system designed to streamline the process of enrolling and dropping courses for students. It ensures that students can only enroll in classes with available seats and provides a user-friendly interface for course management.

PURPOSE OF THIS PROJECT

EnrollMate aims to automate and simplify the course enrollment process for students, faculty, and administrators. The system ensures efficient course registration, seat allocation, and real-time updates, reducing manual effort and administrative burden while improving accessibility and ease of use.

SCOPE OF THIS PROJECT

The system enables students to view, enroll in, and drop courses effortlessly while ensuring that faculty members can manage course enrollments and communicate important announcements to students. Administrators have the ability to add, update, and remove courses while monitoring seat availability in real time.

03

Additionally, the system prevents over-enrollment by displaying seat limits and sending notifications for enrollment deadlines. The project is designed as a web-based platform with role-based access to ensure secure and efficient course management.

OVERVIEW

EnrollMate consists of three primary user roles:

- **Students:** Can search, filter, enroll in, and drop courses.
- **Administrators:** Can manage course offerings and generate reports.

INTERFACE REQUIREMENTS

The system will feature a responsive web-based UI that is accessible from desktops, tablets, and mobile devices. It will include a student dashboard where users can view their enrolled courses and upcoming deadlines. Faculty members will have access to a dedicated portal for managing their courses and student rosters. Administrators will utilize a panel for adding, updating, and removing courses efficiently. Additionally, the system will provide real-time notifications and status updates to keep users informed of critical information.

04

FUNCTIONAL REQUIREMENTS

1. **Course Viewing:** The system shall allow students to view available courses, including seat availability, faculty details, and schedules.
2. **Course Enrollment:** The system shall enable students to enroll in a course if seats are available, ensuring proper validation of prerequisites.
3. **Course Drop:** The system shall allow students to drop a course before the deadline and update available seat counts accordingly.
4. **Faculty Course Management:** The system shall enable faculty members to manage course rosters, view student enrollments, and send announcements.
5. **Administrator Course Control:** Administrators shall have the ability to add, update, and remove courses while monitoring enrollment trends in real time.
6. **Seat Availability Notifications:** The system shall display a "Seats Full" message when a course reaches capacity and prevent additional enrollments.
7. **Enrollment Notifications:** The system shall send automated notifications to students regarding enrollment deadlines, seat availability updates, and course-related announcements.

PERFORMANCE REQUIREMENTS

1. The system should process enrollment requests within 2 seconds.
2. Must support up to 10,000 concurrent users.
3. Course availability should update in real time.
4. Data retrieval should not exceed 3 seconds for any query.
5. The system should maintain 99.9% uptime.

05

USER STORIES

Students:

- 1.As a student, I want to view available courses so that I can choose which ones to enroll in.
- 2.As a student, I want to enroll in a course so that I can attend classes and earn credits.
- 3.As a student, I want to drop a course so that I can adjust my schedule as needed.
- 4.As a student, I want to receive notifications and reminders about enrollment deadlines so that I don't miss important dates.
- 5.As a student, I want to view my enrolled courses in a dashboard so that I can easily track my schedule.

Administrator:

- 1.As an administrator, I want to manage course seat limits so that classes do not exceed capacity.
- 2.As an administrator, I want to generate reports on student enrollments so that I can analyze trends and make decisions.
- 3.As an administrator, I want to add, update, or remove courses from the system so that the course catalog remains current.
- 4.As an administrator, I want to oversee student enrollment trends so that I can predict demand for future semesters.
- 5.As an administrator, I want to set enrollment deadlines and rules so that students follow structured registration processes.

06

USE CASES

1. View Available Courses

- **Actor:** Student
- **Description:** Allows students to browse available courses with details like schedule and seat availability.
- **Preconditions:** The student must be logged in.
- **Postconditions:** The system displays the list of available courses.

2. Enroll in a Course

- **Actor:** Student
- **Description:** Enables students to enroll in a course with available seats.
- **Preconditions:** The student must be logged in and meet course prerequisites.
- **Postconditions:** The student is enrolled in the selected course.

3. Drop a Course

- **Actor:** Student
- **Description:** Allows students to drop an enrolled course before the deadline.
- **Preconditions:** The student must be enrolled in the course.
- **Postconditions:** The course is removed from the student's schedule, and the seat count is updated.

4. View Enrolled Courses Dashboard

- **Actor:** Student
- **Description:** Displays a student's enrolled courses in a dashboard.
- **Preconditions:** The student must be logged in.
- **Postconditions:** The system presents the enrolled courses.

07

USE CASES

5. User Registration

- **Actor:** Student, Administrator
- **Description:** Allows new users to register an account in the system.
- **Preconditions:** User must provide valid credentials and required information.
- **Postconditions:** A new account is created, and the user can log in.

6. User Login

- **Actor:** Student, Administrator
- **Description:** Enables users to log into the system securely.
- **Preconditions:** The user must be registered in the system.
- **Postconditions:** The user gains access to their respective dashboard.

7. Add a Course

- **Actor:** Administrator
- **Description:** Allows administrators to add a new course to the catalog.
- **Preconditions:** Administrator must be logged in.
- **Postconditions:** A new course is added and made available for enrollment.

8. Update a Course

- **Actor:** Administrator
- **Description:** Allows administrators to modify existing course details.
- **Preconditions:** The course must exist in the system.
- **Postconditions:** The course details are updated and saved.

08

USE CASES

9. Manage Course Seat Limits

- **Actor:** Administrator
- **Description:** Allows administrators to set and update course seat limits.
- **Preconditions:** The course must exist in the system.
- **Postconditions:** Seat limits are updated and enforced.

10. Generate Enrollment Reports

- **Actor:** Administrator
- **Description:** Provides reports on student enrollments and trends.
- **Preconditions:** The system must have stored enrollment data.
- **Postconditions:** Reports are generated and displayed.

11. Remove a Course

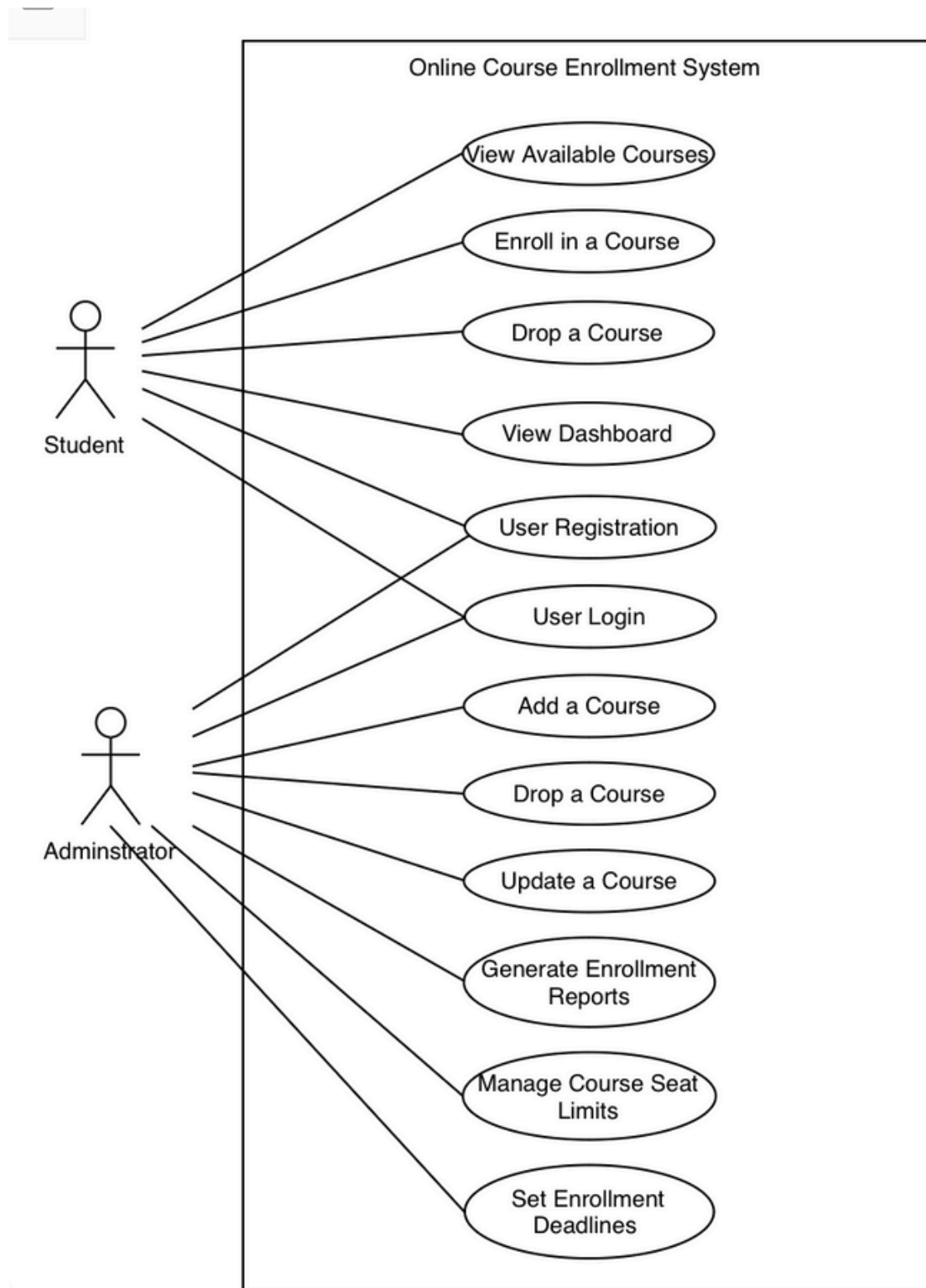
- **Actor:** Administrator
- **Description:** Allows administrators to remove a course from the catalog.
- **Preconditions:** The course must exist in the system, and no students should be enrolled in it.
- **Postconditions:** The course is deleted from the catalog.

12. Set Enrollment Deadlines and Rules

- **Actor:** Administrator
- **Description:** Allows administrators to configure registration deadlines and policies.
- **Preconditions:** The administrator must have system access.
- **Postconditions:** The system enforces registration deadlines for students.

09

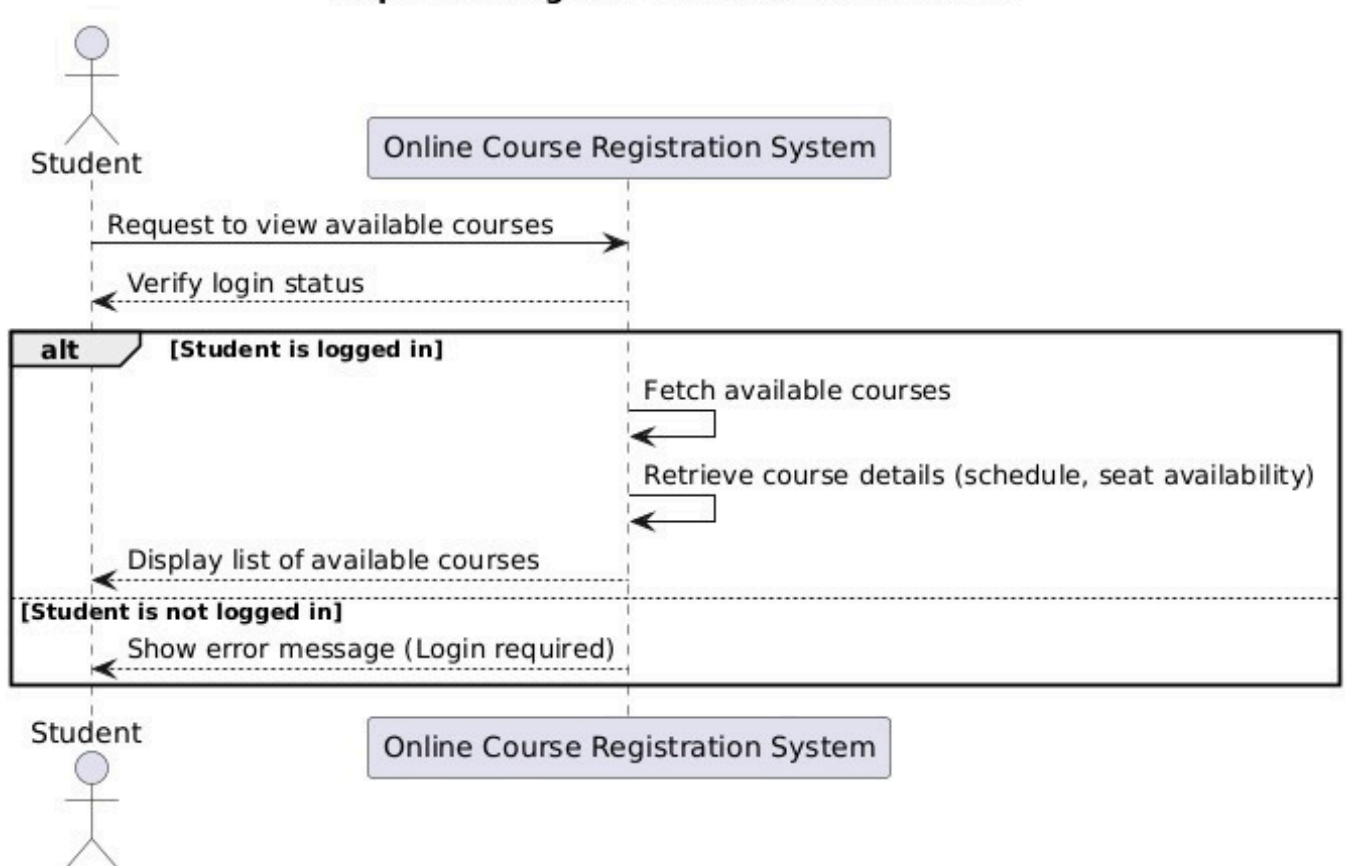
USE CASE DIAGRAM



10

SEQUENCE DIAGRAM: USE CASE 1

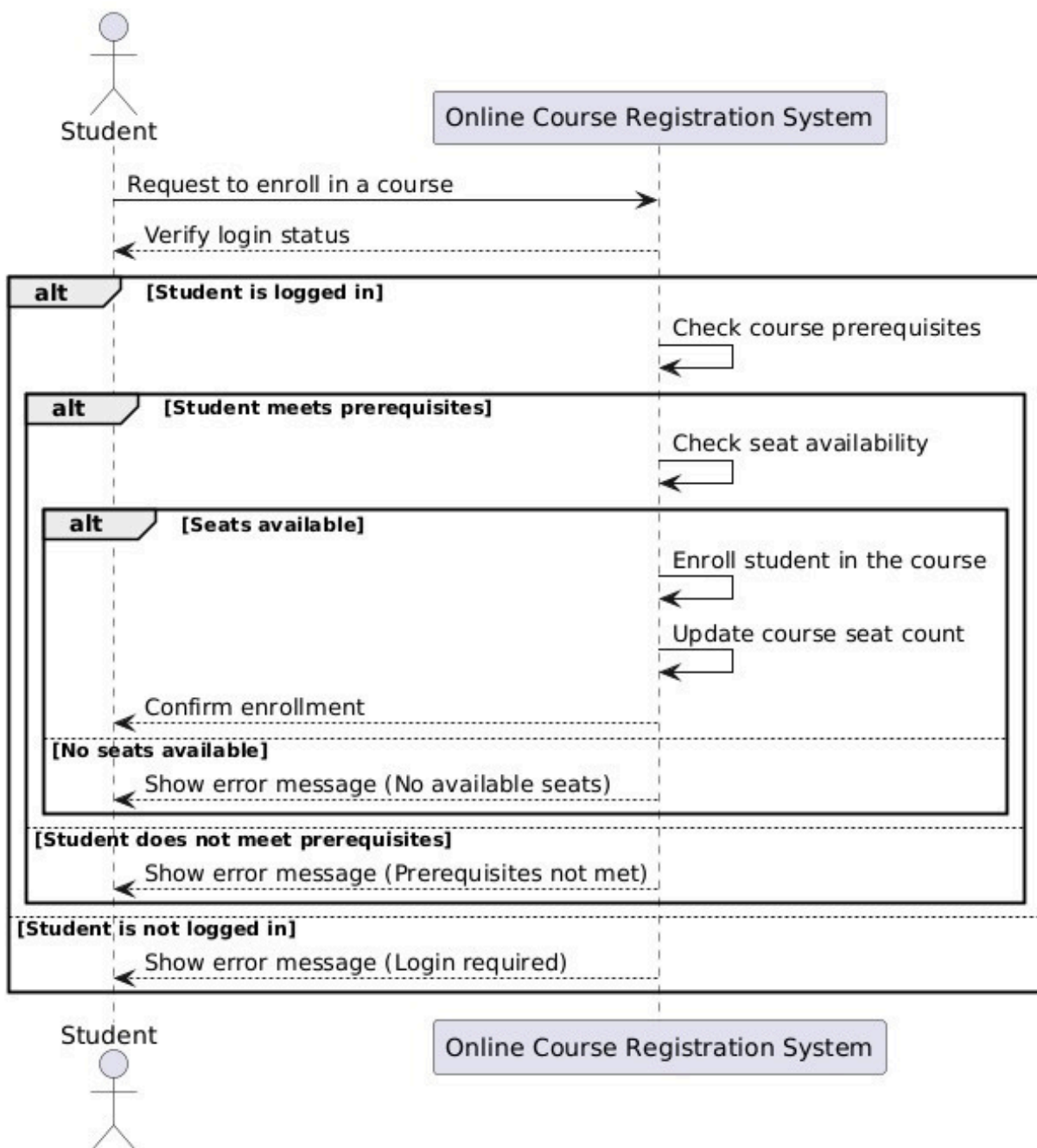
Sequence Diagram: View Available Courses



11

SEQUENCE DIAGRAM: USE CASE 2

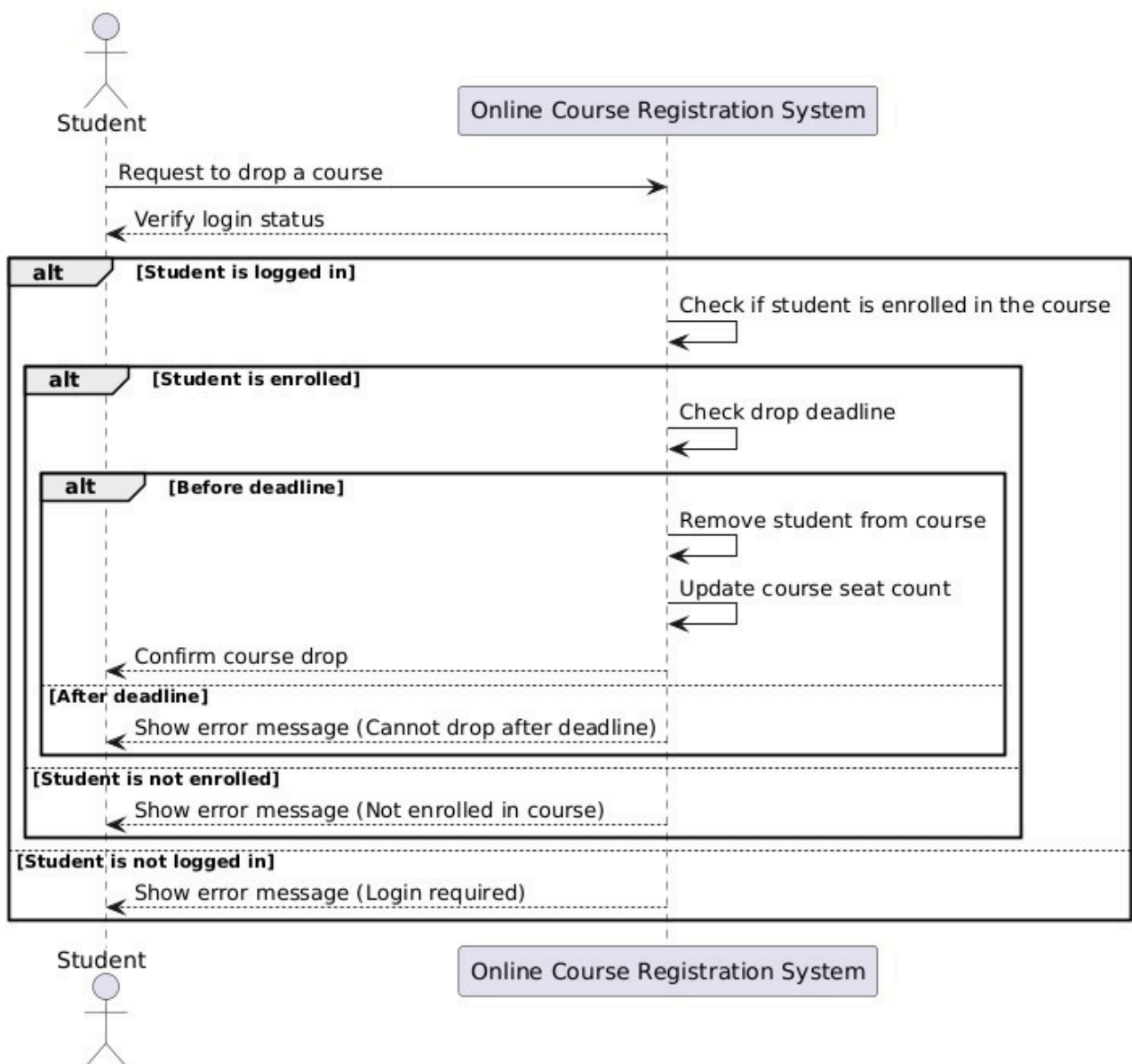
Sequence Diagram: Enroll in a Course



12

SEQUENCE DIAGRAM: USE CASE 3

Sequence Diagram: Drop a Course



13

UML CLASS DIAGRAM

