PROJECT DOCUMENTATION
FINAL REPORT

Project Title:

Course Registration System

Course Title:

Software Engineering

Submitted To:

Hafiz Anzar Ahmed

Submitted By:

Name	Roll Number
Abriza Amin	BITF24M017
Mahnoor Ghani	BITF24M013
Fiza Majeed	BITF24M007

PROJECT REPORT

1. Introduction

Course registration is a fundamental academic process in universities, allowing the students to enroll in courses required for their academic progression. It involves maintaining student records, managing course offerings, and ensuring accurate enrollment of students in selected courses. In many educational institutions, traditional manual or semi-automated registration processes are still used, which often lead to inefficiency, errors, and increased administrative workload. To address these challenges, a structured and user-friendly Course Registration System is required.

2. Problem Statement

The manual management of course registration poses several problems for both students and administrators. Students face difficulties in viewing available courses, understanding course details, and ensuring successful enrollment. Administrators, on the other hand, struggle with maintaining accurate student records, managing course data, and organizing registration information efficiently. Manual processes are time-consuming, prone to duplication and errors, and lack proper data organization. As the number of students and courses increases, these issues become more critical, highlighting the need for an automated and centralized system.

3. Proposed Solution

The proposed **Course Registration System** is a software-based solution designed to automate and streamline the course registration process. The system provides a centralized platform where students can view available courses and register for them in a structured manner. Administrators are given the ability to manage student information, course details, and registration records through an intuitive graphical user interface.

The system consists of well-defined modules such as Student Management, Course Management, Course Registration, and View Registration. These modules work together to ensure accurate data handling, reduce manual workload, prevent registration inconsistencies and improve overall administrative efficiency.

4. System Benefits

The Course Registration System improves efficiency by automating repetitive administrative tasks and organizing academic data in a structured format. It minimizes human errors, ensures consistency in registration records, and provides quick access to student and course information. The user-friendly interface enhances usability for both students and administrators, while the modular design allows easy maintenance and future enhancements.

5. System Modules Overview

The Course Registration System is organized into multiple modules to ensure clarity, modularity, and ease of use. Each module is responsible for handling a specific set of functionalities within the system.

a) Student Management Module: It allows the administrator to add and maintain student information such as Student ID and Student Name and also add the Semester Details. It provides a structured view of all student records, enabling efficient management and reducing data redundancy.

b) Course Management Module: It is responsible for managing course-related data. This includes adding course details such as course code, course title, credit hours, and seat limit. The module displays all available courses in a tabular format, making it easier for administrators to manage course offerings.

c) Course Registration Module: It enables the registration process by allowing the selection of students and courses. It simulates the enrollment process through a user-friendly interface and confirms successful registration. This module ensures that registration data is properly recorded and displayed.

d) View Registration Module: It provides an overview of all registered courses. It displays student names along with their registered course codes and course titles, allowing administrators to monitor and verify registration records efficiently.

7. System Enhancements

The modular design of the Course Registration System allows easy maintenance and future improvements. Enhancements such as the addition of new modules, improved reporting capabilities, and scalability to support a larger number of users can be incorporated without major changes to the system. These features ensure long-term usability and adaptability of the system.

USE-CASE DOCUMENTATION

Use Case ID:

CRMS-01

Use Case Title:

Course Registration Management System

Actors:

Administrator, Student

Pre-Conditions:

Authorized user initiates the Course Registration Management System.

Post-Condition:

Course registration operations are successfully performed and academic records are updated and displayed.

MAIN FLOW:

1. User requests to access the Course Registration Management System.

2. System displays the main menu interface.

3. User selects the desired module from the main menu.

4. If the user is Administrator, the system displays administrative options.

4.1. Administrator selects the option to **manage semesters**.

4.1.1. System displays the Semester Management interface.

4.1.2. Administrator enters Semester ID and Semester Name.

4.1.3. System validates semester data.

4.1.3.1. If **semester already exists**:

4.1.3.1.1. System displays duplication error message.

4.1.3.1.2. Administrator re-enters semester details.

4.1.4. System saves semester information.

4.1.5. System marks the semester as active.

4.2. Administrator selects the option to **manage students**.

4.2.1. System displays the Student Management interface.

4.2.2. Administrator enters Student ID and Student Name.

4.2.3. System validates the entered student data.

4.2.3.1. If **required data is missing or invalid**:

4.2.3.1.1. System displays an error message.

4.2.3.1.2. Administrator corrects and re-enters the data.

4.2.3.2. If **Student ID already exists**:

4.2.3.2.1. System displays a duplicate student warning.

4.2.3.2.2. Administrator cancels or updates student entry.

4.2.4. System saves student information.

4.2.5. System displays student details in a table format.

4.3. Administrator selects the option to **manage courses**.

4.3.1. System displays the Course Management interface.

4.3.2. Administrator enters Course Code, Course Title, Credit Hours, Seat Limit, and Semester.

4.3.3. System validates course data.

4.3.3.1. If **required fields are missing**:

4.3.3.1.1. System displays a validation error message.

4.3.3.1.2. Administrator re-enters course data.

4.3.3.2. If **course code already exists for the same semester**:

4.3.3.2.1. System displays a duplicate course warning.

4.3.3.2.2. Administrator modifies course details.

4.3.4. System saves course information for the selected semester.

4.3.5. System displays available courses semester-wise.

4.4. Administrator selects the option to **register a course**.

4.4.1. System displays the Course Registration interface.

4.4.2. System displays a list of active semesters.

4.4.2.1. If **no active semester exists**:

4.4.2.1.1. System displays a "No Active Semester" message.

4.4.2.1.2. Administrator returns to Semester Management.

4.4.3. System displays students and courses for the selected semester.

4.4.4. Administrator selects a student.

4.4.5. Administrator selects a course.

4.4.6. System checks seat availability.

4.4.6.1. If **seat limit is exceeded**:

4.4.6.1.1. System displays a seat availability error.

4.4.6.1.2. Administrator selects another course.

4.4.6.2. If student is **already registered in the selected course for the same semester**:

4.4.6.2.1. System displays a duplicate registration warning.

4.4.6.2.2. Administrator cancels the registration.

4.4.7. System creates a semester-based registration record.

4.4.8. System displays successful registration confirmation.

4.5. Administrator selects the option to **view registrations**.

4.5.1. System displays the View Registration interface.

4.5.2. Administrator selects a semester.

4.5.3. System displays student names with course codes and titles for the selected semester.

5. If the user is Student, the system displays student options.

5.1. Student selects an active semester.

5.1.1. If **no active semester exists**:

5.1.1.1. System displays “Registration Closed” message.

5.1.1.2. Student exits the system.

5.2. System displays available courses for the selected semester.

5.3. Student selects a course for registration.

5.4. System validates the registration request.

5.4.1. If the **course is full**:

5.4.1.1. System displays seat unavailability message.

5.4.1.2. Student selects another course.

5.4.2. If the **student is already registered in the course**:

5.4.2.1. System displays duplicate registration warning.

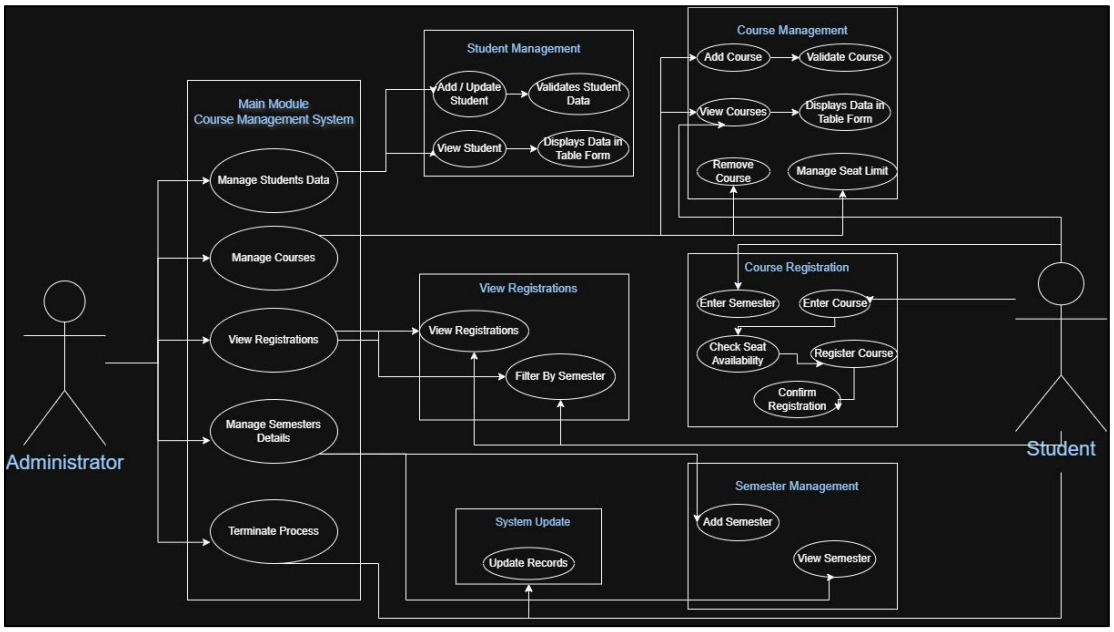
5.4.2.2. Student returns to course list.

5.5. System confirms successful enrollment.

6. System saves all student, course, semester, and registration data.

7. User navigates back to the main menu or exits the system.

USE CASE DIAGRAM



DOMAIN MODEL

The **student** initiates the **system** by accessing the course registration interface. After logging in, the **student** views a list of available **courses** offered for the current semester. Each **course** displays essential information such as course code, course title, credit hours, and seat limit, helping the **student** make an informed decision.

Once the desired **course** is selected, the **student** proceeds to enroll through the **registration** process. The **system** verifies the availability of seats and confirms the successful **registration**. A confirmation message is displayed, indicating that the **student** has been enrolled in the selected **course**.

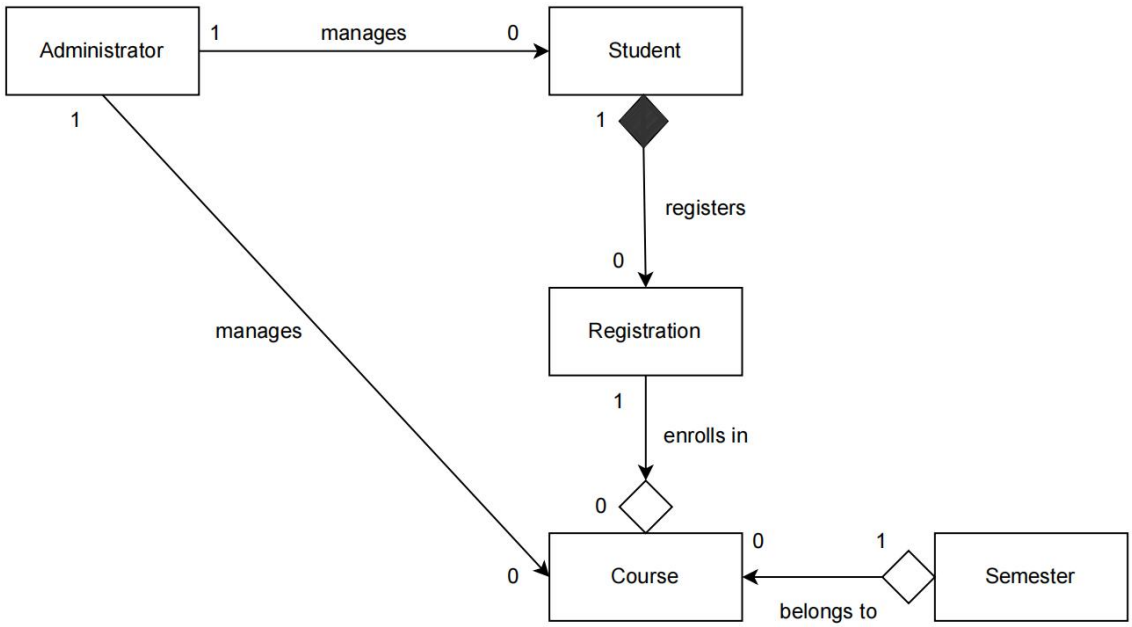
The **administrator** plays a crucial role in maintaining the system. The **administrator** manages **student** records, adds and updates **course** details, and monitors all **registrations**. By overseeing these activities, the **administrator** ensures data accuracy and smooth functioning of the course registration process.

The **registration** entity acts as a link between the **student** and the **course**. A single **student** may have multiple **registrations**, while each **course** can also be associated with many **registrations**. This structure allows the **system** to efficiently track enrollment information.

Entities:

Candidate Entities	Discarded Entities
Student	System
Course	Interface
Registration	Message
Administrator	Process
Semester	Information

Domain Model Diagram:



ACITIVITY DIAGRAM