Software Requirements Specification for

ONLINE COURSE RESERVATION SYSTEM

Submitted To: Savitha Ramesh

Prepared By:

S.PAVAN KUMAR – 22BD1A12B3

Tanmay Jain – 22BD1A12C0

Sampath – 22BD1A12C7

M.Krishna Vamshi – 22BD1A1292

V M Sathwik - 22BD1A12C2

Kandi Bhargav – 22BD1A1285

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Week-1	15-09-20	SRS Creation (Introduction)	1.0

Problem Statement

The Online Course Reservation System is designed to simplify course registration for students and administrators by providing a user-friendly, web-based platform. Students can easily browse available courses, check prerequisites, and reserve seats with real-time seat availability updates. Administrators will have tools to manage course offerings, faculty assignments, and enrollment limits, significantly reducing manual errors and improving efficiency. The system will also integrate with existing student information systems, automate waitlisting, and send confirmation emails for successful reservations.

Additionally, the system will enhance accessibility with mobile compatibility, enabling students to register from anywhere. Features like personalized course recommendations, search filters, and calendar synchronization will further streamline the process, allowing students to efficiently plan and manage their academic schedules. This will ultimately reduce administrative overhead and provide a smoother experience for all users involved.

1.INTRODUCTION

1.1 Purpose

This document outlines the software requirements for an online course reservation system. It is intended for designers, developers, and maintainers of the course reservation platform. These requirements were developed in response to the challenges of manual course registration, aiming to improve efficiency, reduce errors, and enhance the user experience.

For students, the system provides a convenient platform to browse and reserve courses, with features like personalized recommendations and mobile access. For administrators, it offers comprehensive tools for managing course offerings, faculty assignments, and enrollment capacities. Ultimately, the system's purpose is to create a seamless, accessible, and efficient registration process, benefiting both students and the academic administration.

1.2 Document Conventions

The Online Course Reservation System is a web-based application that allows students to check course availability, reserve seats, and manage their registrations online. The system is designed for all registered users, providing access to courses and registration features once they have gained authorization from the administrator

1.3 Definitions, Abbreviations

- OCRS – Online Course Reservation System
- CN – Course Number
- Reservation: The process of securing a seat in a specific course.
- Waitlist: A queue where students are placed when a course is full, pending seat availability.
- SID – Student Identification Number
- Report – Summary of a particular course or registration activity
- Admin Staff – Administrative personnel responsible for managing user access, course offerings, and system settings
- Prerequisite: A requirement that must be met before a student can register for a specific course.
- User Access: The permission granted to students, faculty, or administrators to use the system's features based on their role

1.4 Project Scope

The Online Course Reservation System consists of three modules. The first module allows students to inquire about the availability of courses on specific dates, the second module enables them to reserve a seat in a chosen course, and the third module allows for the cancellation of a previously reserved course. The current registration process typically relies on manual methods for enrolling in courses, which can lead to inefficiencies and errors in managing student registrations.

2.Overview

2.1 Online Course Reservation System

During the offline era, course registration posed various challenges for students and educational institutions. Offline methods limited students' ability to explore different options based on their academic criteria and created additional administrative burdens for institutions. Moreover, educational institutions struggled to effectively monitor course enrollment and availability.

Many small and medium educational organizations lack their own online course reservation systems. The Online Course Reservation System is a comprehensive web-based platform that facilitates in-house management of course offerings, registrations, reporting, and other administrative functions associated with student enrollment.

The Online Course Reservation System provides enhanced services to both educational institutions and students, including:

- Capture of student information such as name, address, phone number, and email address
- Course availability and pricing information
- Course rankings or ratings
- Course seating or capacity charts
- Search functionality for easy navigation
- Payment processing for course fees
- Institutional branding, including logos, slogans, and contact information
- A comments and suggestions section for user feedback
- Comprehensive reporting features for enrollment and performance tracking

3.Overall Description

3.1 Product

The Online Course Reservation System is a self-contained platform that manages the activities related to course registrations, including student information and enrollment processes. Various stakeholders are involved in the Online Course Reservation System, including students, faculty, administrators, and support staff, each playing a crucial role in ensuring the smooth operation of the system and enhancing the overall user experience.

]3.2 Product Functionality

➢ Registration

When a student seeks to register for a course, the system first checks to see if the student is already registered using their details. If not, a new Student Identification Number (SID) is assigned to the student. The student's information, including date of birth, address, and mobile number, is then captured and stored in the system.

➢ Course Reservation

Once the student logs in with their credentials, they are prompted to select their desired course, including the course code, date, and time. The system checks the database for seat availability in the chosen course. If a seat is available, the reservation is confirmed through e-payment, and the student receives a confirmation with the course details, including seat number, course code, date, and time, along with a digital ticket blueprint.

➢ Course Cancellation

In the event that a student needs to cancel their reservation, the system allows for easy cancellation within a specified time frame. The student can log in to their account and view their current reservations, selecting the course they wish to cancel. Upon cancellation, a confirmation message is sent, and any applicable refunds are processed according to the institution's policies.

➢ Feedback and Support

Students can provide feedback on their course experience through the system, allowing for continuous improvement in course offerings. The system also features a support module where students can submit inquiries or issues they encounter, ensuring prompt assistance from the administration team. This creates a supportive learning environment and enhances student satisfaction

➢ Report Generation

The system generates various reports, including detailed information regarding the courses offered, enrollment statistics, and student registrations, providing valuable insights for administrators and faculty to manage course offerings effectively.

4.Specific Requirements

4.1 System Requirements

Hardware: Intel Core 2 Duo System or equivalent Minimum 15 TB of storage capa

Software: HTML for structuring web content

CSS for styling and layout design

JavaScript for interactive features and functionality

MySQL for database management and data storage

4.2 Functional Requirements

➢ **Registration**

- **Assign PIN:** The Online Course Reservation System (OCRS) allows administrators to assign each student a unique Student Identification Number (SID) during the registration process. This ID will be used by the student throughout their academic journey within the system.

➢ **Course Reservation**

- **Sign In:** Students must first sign in to the system using the username and password they provided during registration. The system will validate the credentials, allowing access only if the information is correct.
- **Check Availability:** Once logged in, students should be able to view all available courses, including relevant details such as course descriptions, timings, and prerequisites.
- **Reserve Course:** If a seat is available, the system will book it under the student's SID. The reservation should be locked to that student until they decide to cancel or until the course begins, preventing double booking.

➢ **Course Cancellation**

- Cancel Reservation: Students should have the ability to cancel their course reservations through their account. Upon cancellation, the seat should become available for other students, and a confirmation of the cancellation should be sent to the student.

➢ **Report Generation**

- Seat Availability Reports: The OCRS shall generate reports regarding seat availability, including details such as course code, seat number, and occupancy status (occupied/unoccupied).
- Enrollment Statistics: The system will generate reports on overall enrollment data, including the number of students per course and demographic information, helping administrators make informed decisions about course offerings.

➢ **Database**

- Student-Related Information: Each student shall have the following mandatory information stored: first name, last name, phone number, Student ID, address, postal code, city, username, and password.
- Course-Related Information: Each course should have the following information: course code, course title, number of available seats, course type, and schedule details
- Payment Information: The system must securely store payment-related information for each transaction, ensuring compliance with data protection regulations.

➢ **Feedback Mechanism**

Course Feedback: Students shall have the ability to provide feedback on courses they have completed. This feedback will be stored in the system and can be used for future improvements to course offerings.

Non-Functional Requirements

➢ **Performance Requirements**

- **Response Time**: The system shall provide responses within 1 second after processing passenger information to ensure a smooth user experience
- **Capacity**: The system must support simultaneous access for up to 1000 users, accommodating peak usage without degradation in performance.
- **Scalability:** The system should be designed to scale efficiently, allowing for increased user load or additional functionalities without significant rework.

➢ **Safety and Security Requirements**

- **Student Identification:** The system requires each student to verify their identity using their unique Student Identification Number (SID) during registration and booking processes.

- **Data Modification:** Any modifications (insert, update, delete) in the database shall be synchronized and allowed only by authorized administrators to maintain data integrity.
- **Administrator's Rights**: Administrators shall have comprehensive access to view and modify all information in the system, including user data and course offerings.
- **Data Encryption:** Sensitive information, such as passwords and payment details, must be encrypted during transmission and storage to safeguard against unauthorized access.
- **User Authentication:** The system shall implement robust authentication mechanisms to prevent unauthorized access to student accounts.

➢ **Software Quality Attributes**

- **Back Up:** he system shall provide regular data backup capabilities to prevent information loss in the event of system failures or outages.
- **Error Logging:** The system shall maintain a log of all errors and exceptions, enabling administrators to monitor, diagnose, and resolve issues effectively.
- **Availability:** The system shall aim for high availability, targeting 99.9% uptime, ensuring users can access the platform at any time without interruption.
- **Usability:** The system shall feature user-friendly interfaces to ensure easy navigation for all users, requiring minimal training or technical support**.**
- **Maintainability:** The system shall be built with a modular architecture to facilitate easy updates and maintenance, allowing for the addition of new features with minimal downtime.
- **Performance Monitoring:** The system shall include performance monitoring tools to track system responsiveness and load, ensuring proactive management of resources.

5. Conclusion:

In conclusion, the Online Course Reservation System project involves the design and development of a software solution that enables students to register for courses online efficiently. The project's scope includes creating a system that is user-friendly, secure, scalable, and accessible to a diverse range of users, including students, faculty, and administrators. Key functional requirements of the system encompass features such as course registration, seat selection, course cancellation, and secure payment processing. Additionally, non-functional requirements focus on performance, security, accessibility, and overall user experience.

To ensure the success of the project, the development team must also address system constraints, including platform and technology choices, integration with existing educational systems, and ongoing maintenance and support. Collaboration with educational institutions is crucial to ensure that the system aligns with their specific requirements and regulatory standards. By prioritizing user needs and adhering to best practices in software development, the project aims to deliver a robust solution that enhances the course registration experience for all stakeholders involved.

UML DIAGRAMS:

Conceptual Model of UML (Unified Modeling Language)

The conceptual model in UML helps represent the structure and behavior of a system using various diagrams. UML offers a standard way to visualize the design of a system and is widely used in object-oriented software development. It typically involves three main components:

1. **Objects**: These represent real-world entities.
2. **Relationships**: These define how objects interact with each other.
3. **Actions/Behaviors**: This defines the functionality or behavior of the system.

USE CASE DIAGRAM:-

A use case diagram illustrates the interactions between users (actors) and a system. It visually represents how users interact with the system's functionalities.

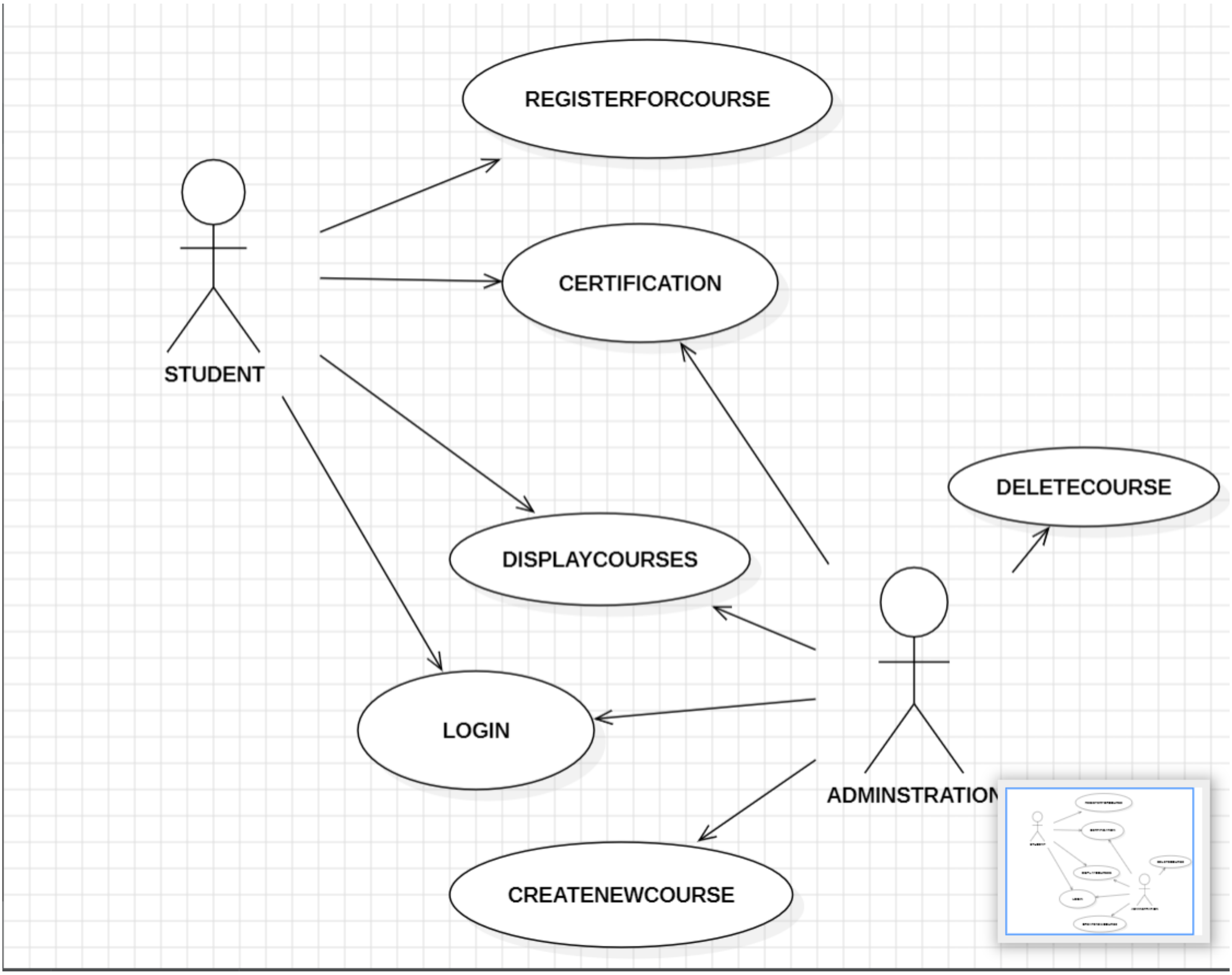
Elements in a Use Case Diagram:

- **Actors**: Represent users or external systems that interact with the system. They can be a person, another system, or an organization.
- **Use Cases**: These represent the specific actions the system can perform. For example, in an "Online Course Reservation System," use cases might include:
  - **Register for a Course**
  - **View Course Details**
  - **Cancel Reservation**
- **Relationships**:
  - **Association**: A line between an actor and a use case that shows interaction.
  - **Include**: A use case that includes the behavior of another use case.
  - **Extend**: A use case that extends the behavior of another use case, typically representing optional actions.
  - **Generalization**: A relationship between actors or use cases where one is a more specific version of the other.

Example (Use Case Diagram for an Online Course Reservation System):

- **Actors**:
    - Student
    - Admin
    - Instructor
- **Use Cases**:
    - Student: Register for Course, View Available Courses, Cancel Reservation.
    - Admin: Add/Remove Courses, View Registrations.
    - Instructor: Update Course Details.

Use Case Diagram:-



CLASS DIAGRAM:-

A class diagram describes the structure of a system by showing its classes, attributes, operations (or methods), and the relationships between objects.

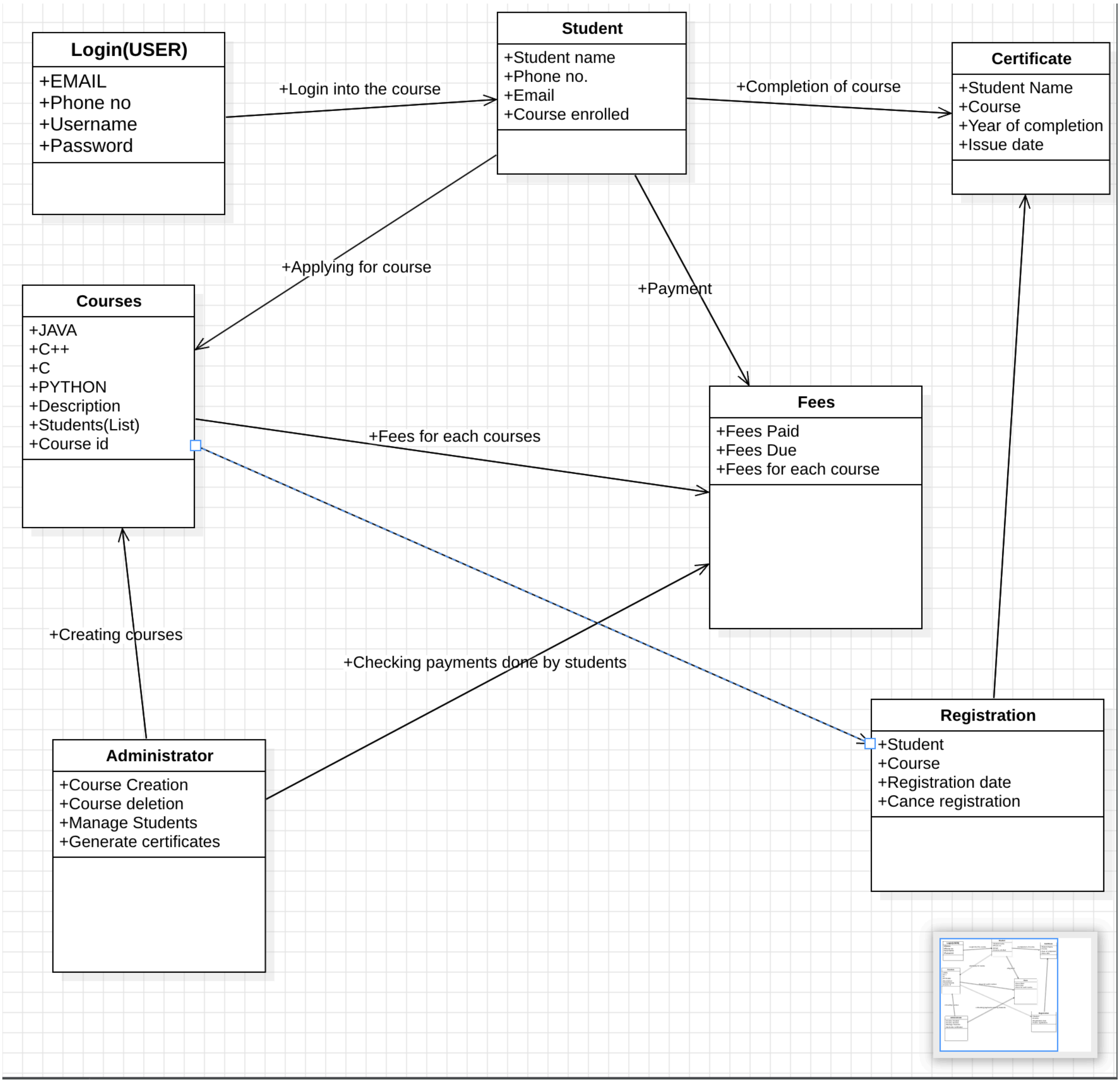
Elements in a Class Diagram:

- **Class**: Represents a blueprint for objects, containing attributes and methods.
- **Attributes**: Characteristics or properties of a class.
- **Methods**: Actions or functions that a class can perform.
- **Relationships**:
  - **Association**: Represents a link between two classes.
  - **Aggregation**: A "whole-part" relationship between two classes, where the part can exist independently of the whole.
  - **Composition**: A stronger form of aggregation where the part cannot exist without the whole.
  - **Inheritance**: A relationship where one class (subclass) inherits the properties of another class (superclass).
  - **Multiplicity**: Defines the number of instances in the relationship between classes.

Example (Class Diagram for an Online Course Reservation System):

- **Classes**:
  - **Student**:
    - Attributes: StudentID, Name, Email.
    - Methods: Register(), ViewCourses(), CancelReservation().
  - **Course**:
    - Attributes: CourseID, CourseName, Instructor, Capacity.
    - Methods: UpdateCourseDetails(), GetAvailableSeats().
  - **Instructor**:
    - Attributes: InstructorID, Name, Email.
    - Methods: AssignCourse(), UpdateCourseDetails().
  - **Admin**:
    - Attributes: AdminID, Name, Email.
    - Methods: AddCourse(), RemoveCourse().
- **Relationships**:
  - **Student registers for a Course** (association).
  - **Instructor manages a Course** (aggregation).
  - **Admin adds/removes a Course** (association).

Class Diagram:-


Login(USER)
+EMAIL
+Phone no
+Username
+Password
+Login into the course
Student
+Student name
+Phone no.
+Email
+Course enrolled
+Completion of course
Certificate
+Student Name
+Course
+Year of completion
+Issue date
+Applying for course
+Payment
Courses
+JAVA
+C++
+C
+PYTHON
+Description
+Students(List)
+Course id
+Fees for each courses
Fees
+Fees Paid
+Fees Due
+Fees for each course
+Creating courses
+Checking payments done by students
Registration
+Student
+Course
+Registration date
+Cance registration
Administrator
+Course Creation
+Course deletion
+Manage Students
+Generate certificates