ONLINE COURSE PORTAL SYSTEM

SUBMITTED BY:

NAME: ALDRIN ROGER S
DEPT&SEC: CSE-A (IIND YEAR)
ROLL NO: 2116220701023

TABLE OF CONTENTS

OVERVIEW OF THE PROJECT

The primary goal of this online course portal system is to provide a comprehensive platform for users to access and manage online courses effectively, streamlining course enrolment, content delivery, and administration. Encompassing functionalities such as course management, user authentication, feedback collection, and administrative tasks, the system aims to create an efficient, user-friendly environment for both students and educators. Teachers can create and manage courses, upload various materials, and organize curricula, while students can browse, enroll in, and manage courses through a user-friendly interface. The portal supports multiple content formats, progress tracking, secure registration and login, and role-based access control. A feedback system allows students to provide course and instructor feedback, and teachers to create surveys and quizzes. Administrators can manage user accounts, generate automated reports, and send notifications and announcements. Detailed course information, including syllabus and instructor details, is accessible to students, who can also use advanced search and filtering options to find specific courses. The system enhances efficiency, accessibility, and engagement by automating administrative tasks and offering flexible access to course materials. Robust tracking and reporting tools provide valuable insights into student performance and course effectiveness. Key implementation considerations include scalability, security, user experience, and integration with other systems. By leveraging technology to create a more accessible, efficient, and engaging learning environment, this portal modernizes educational processes and contributes to a more effective educational ecosystem.

SOFTWARE REQUIREMENTS SPECIFICATION(SRS)

EX_NO: 1 DATE: 20-02-24

1.1 INTRODUCTION

The purpose of this online portal system is to provide a platform for users to access and manage online courses. This system aims to facilitate course enrolment and content delivery and administration of courses. This system will cover course management, user authentication, feedback and administrative functionalities. It facilitates to access the information of a particular course. The information is provided by the teacher for a particular course. The purpose of developing software is to computerized the tradition way of taking class. The project aims at creating a courses portal for a campus/organization.

1.2 SCOPE

This project will create an Online Course Portal System for easy course enrollment, content sharing, and management. Features include user registration, course setup, content upload, discussion forums, assignment submission, quizzes, progress tracking, user profiles, and notifications. The system will be fast, secure, reliable, easy to use, and work on various devices and browsers while following data protection rules.

1.3 FUNCTIONAL REQUIREMENT

1.3.1 User registration and authentication

Users must be able to register the system using a valid mail address. User authentication should be secure and include password recovery options.

1.3.2 Course creation and management

Instructors should be able to create new courses with the title description and other relevant information. Course content, including lectures and resources, can be uploaded and managed.

1.3.3 Content Upload and Delivery

Instructors can upload multimedia content, documents, and presentations for course delivery. Support for various file formats and media types.

1.3.4 Discussion Forums

Each course should have a discussion forum for students and instructors to interact. Threaded discussions with the ability to attach files and multimedia.

1.3.5 Assignment Submission and Grading

Instructors can create assignments with due dates. Students should be able to submit assignments, and instructors can grade them with feedback.

1.3.6 Quiz and Exam Module

Instructors can create quizzes and exams with various question types. Automated grading for objective questions and manual grading for subjective questions.

1.3.7 Progress Tracking and Reporting

Users can track their progress within a course. Instructors can generate reports on student performance.

1.3.8 User Profile Management

Users can update their profiles, including personal information and profile pictures. Instructors can view and manage student profiles within their courses.

1.3.9 Notifications and Announcements

Users should receive notifications for course updates, announcements, and assignment deadlines.

1.4 NON-FUNCTIONAL REQUIREMENT

1.4.1 Performance

The system should handle a large number of simultaneous users. Response time for user interactions should be within acceptable limits.

1.4.2 Security

User data should be stored securely. Access to sensitive information should be restricted based on user roles.

1.4.3 Reliability

The system should be available 24/7 with minimal downtime for maintenance. Regular data backups should be performed.

1.4.4 Usability

The user interface should be intuitive and user-friendly. Support for different devices and screen sizes.

1.4.5 Compatibility

The system should be compatible with popular web browsers (Chrome, Firefox, Safari). Mobile responsiveness for accessibility on smartphones and tablets.

AGILE SCRUM METHODOLOGY

EX_NO: 2 DATE: 01-03-24

2.1 Sprint 1 Backlog

1. User Authentication

- User Story: As a user, I want to securely register and log in to the online course portal so that I can access my account.
- Tasks:
    - Implement user registration form with email and password.
    - Develop email verification feature.
    - Implement login functionality
    - Create password recovery feature.

2. Basic UI Setup

- User Story: As a user, I want a user-friendly interface so that I can easily navigate the system.
- Tasks:
    - Set up the basic layout and navigation
    - Design login, registration, and dashboard screens.

3. User Profile Management

- User Story: As a user, I want to manage my profile so that my personal information is up-to-date.
- Tasks:
    - Create user profile page.
    - Allow users to update their personal information and profile picture.

2.2 Sprint 2 Backlog

4. Course Creation and Management

- User Story: As an instructor, I want to create and manage my courses so that I can provide course content to students.
- Tasks:
    - Allow instructors to edit and delete courses.
    - Develop course management dashboard for instructors.

5. Content Upload and Delivery

- User Story: As an instructor, I want to upload multimedia content and documents so that I can deliver course materials to students.
- Tasks:
    - Support for various file formats (PDF, DOC, MP4, etc.).
    - Implement content upload functionality.
    - Create content delivery interface for students.

6. Notifications and Announcements

- User Story: As a user, I want to receive notifications for course updates and announcements so that I stay informed.
- Tasks:
    - Implement notification system.
    - Allow instructors to create and send announcements.
    - Display notifications on user dashboard.

2.3 Sprint 3 Backlog

7. Discussion Forums

- User Story: As a student, I want to participate in course discussions so that I can interact with instructors and peers.
- Tasks:
    - Implement discussion forum for each course.
    - Allow threaded discussions with file attachments.
    - Enable moderation features for instructors.

8. Assignment Submission and Grading

- User Story: As a student, I want to submit assignments so that I can receive feedback and grades.
- Tasks:
    - Create assignment submission form.
    - Allow file uploads for assignments.
    - Develop grading interface for instructors with feedback options.

9. Quiz and Exam Module

- User Story: As an instructor, I want to create quizzes and exams so that I can assess student learning.
- Tasks:

- Implement quiz and exam creation with various question types (multiple choice, short answer, etc.).
- Develop automated grading for objective questions.
- Create manual grading interface for subjective questions.

2.4 Sprint 4 Backlog

10. Progress Tracking and responding

- User Story: As a student, I want to track my progress within a course so that I know how well I am doing.
- Tasks:
  - Implement progress tracking features.
  - Allow students to view progress reports.
  - Develop performance reporting tools for instructors.

11. Role-Based Access Control

- User Story: As an administrator, I want to manage user roles and permissions so that only authorized users can perform certain actions.
- Tasks:
  - Implement role-based access control system.
  - Assign roles (admin, instructor, student) to users.
  - Set permissions for each role.

12. Testing and Bug Fixes

- User Story: As a developer, I want to ensure the system is reliable and bug-free so that users have a smooth experience.
- Tasks:
  - Conduct unit and integration testing.
  - Perform user acceptance testing.
  - Fix identified bugs and issues.

USER STORIES

EX_NO: 3 DATE: 12-03-24

User Story 1: As a Student, I want to browse and enroll in courses easily so that I can start learning quickly and efficiently.

Acceptance Criteria:

- The system displays a comprehensive list of available courses with detailed descriptions, syllabi, and instructor profiles.
- Students can filter courses by category, difficulty level, and instructor.
- The system provides a clear and guided enrollment process, including secure payment options if necessary.
- An enrollment confirmation is sent via email or notification within the system.

User Story 2: As a Student, I want to access course materials from any device, so I can study at my convenience and pace.

Acceptance Criteria:

- The system supports access to course materials on various devices (PC, tablet, smartphone).
- Course materials are available in multiple formats (video, documents, slides).
- Students can download materials for offline access.

User Story 3: As a Student, I want to submit assignments online, receive timely feedback, and track my progress to improve my understanding and performance.

Acceptance Criteria:

- The system allows for online submission of assignments.
- Instructors can provide feedback and grades directly in the system.
- Students can view their progress and feedback in real-time.

User Story 4: As a Student, I want to participate in discussion forums for each course to interact with my peers and instructors and enhance my learning experience.

Acceptance Criteria:

- Each course has a dedicated discussion forum.
- Students and instructors can start and reply to threads.

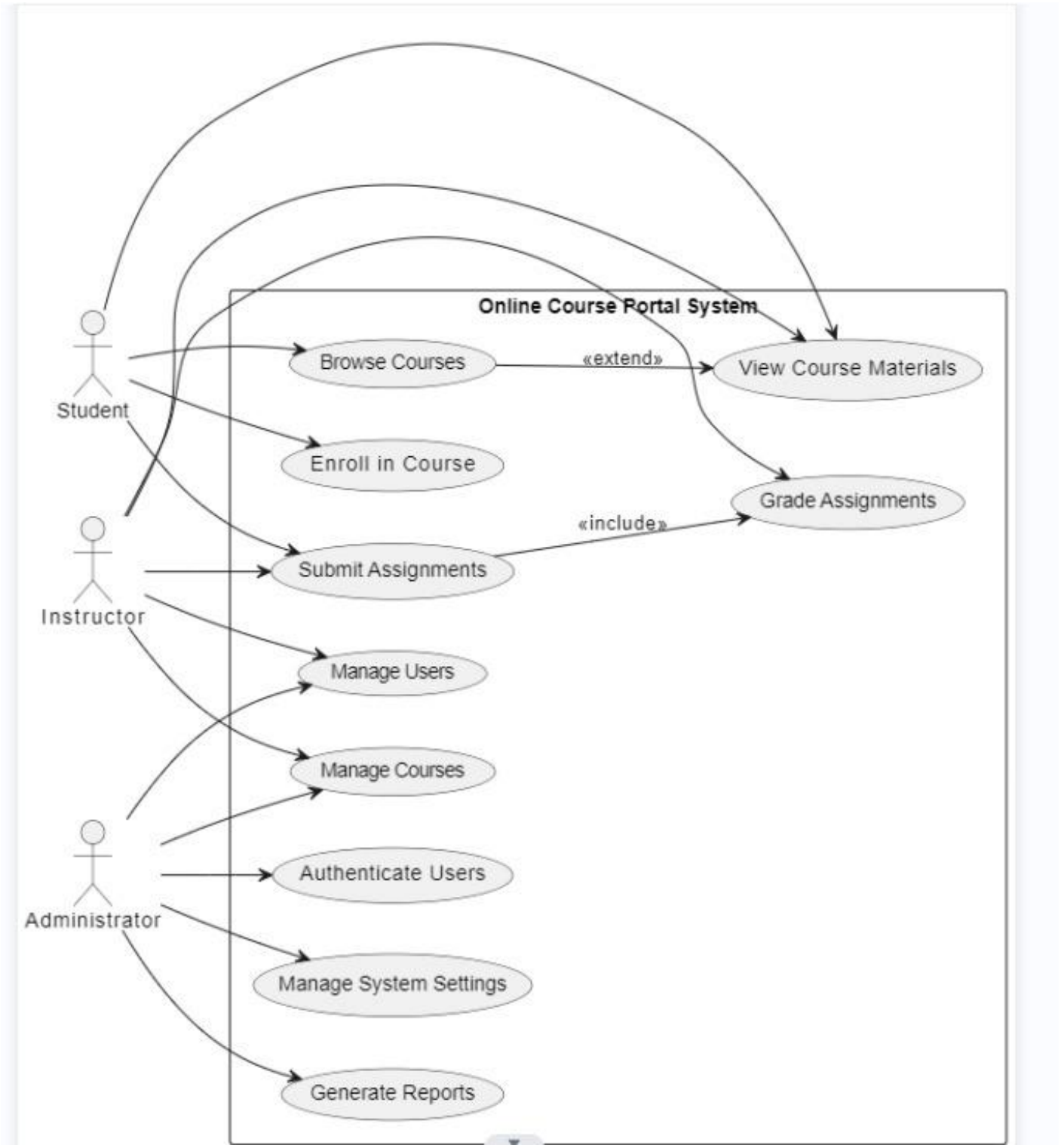
User Story 5: As an Instructor, I want to create and manage courses, track student progress, and provide feedback efficiently to deliver high-quality education.

Acceptance Criteria:

- Instructors can create new courses and input relevant details.
- The system allows for easy uploading and organizing of course materials.
- Instructors can create, assign, and grade assignments within the platform.
- The system generates detailed progress and performance reports, allowing instructors to view individual and overall student performance.
- Reports include metrics such as assignment grades, quiz scores, and participation in discussions.

USE-CASE DIAGRAM

EX_NO: 4 DATE: 19-03-24



1. ***Actors:***
    - Student: A user who enrolls in and participates in courses.
    - Instructor: A user who creates and manages courses, including grading assignments.
    - Administrator: A user who oversees the system, managing users and system settings.

2. ***Use Cases:***
    - Browse Courses: Allows students to search for and view available courses.
    - Enroll in Course: Enables students to enroll in chosen courses.
    - View Course Materials: Lets students access lectures, slides, and other resources once enrolled in a course.
    - Submit Assignments: Allows students to submit their assignments online.
    - Grade Assignments: Instructors can review and grade student submissions.
    - Manage Users: Instructors and administrators can add, update, or delete user information.
    - Manage Courses: Instructors can create, update, and manage course content.
    - Authenticate Users: Administrators ensure secure login and access control.
    - Manage System Settings: Administrators configure system preferences and settings.
    - Generate Reports: Administrators can create and view reports on various system metrics.

3. ***Relationships:***
    - Extend Relationship («extend»):
      Browse Courses extends to View Course Materials: This indicates that browsing courses can lead to viewing course materials, but viewing course materials is a broader activity that encompasses more actions.
    - Include Relationship («include»):

Enroll in Course includes Grade Assignments: This signifies that grading assignments is a necessary part of the enrollment process. Once students enroll, their assignments must be graded as part of their course participation.

4. ***Flow of Interaction:***
    - Students: They start by browsing available courses.
    - They can enroll in courses of interest.
    - Upon enrolling, they can view course materials.
    - They submit assignments as part of their coursework.

5. ***Instructors:***
    - They manage users (e.g., students in their courses).
    - They create and manage the courses.
    - They grade assignments submitted by students.

6. ***Administrators:***
    - They authenticate users to ensure secure access.
    - They manage overall system settings.
    - They generate reports to monitor system usage and performance.

NON-FUNCTIONAL REQUIREMENTS (NFR)

EX_NO: 5 DATE: 29-03-24

1. ***Performance and Scalability:***
    - Response Time: The system must handle user interactions and deliver course materials within 2 seconds.
    - Scalability: The system should support up to 15,000 concurrent users during peak times.

2. ***Security:***

    Data Protection: Encrypt all user data both at rest and in transit using industry-standard techniques.
    - Access Control: Implement role-based access control (RBAC) to ensure that only authorized users can access sensitive information and perform administrative functions.

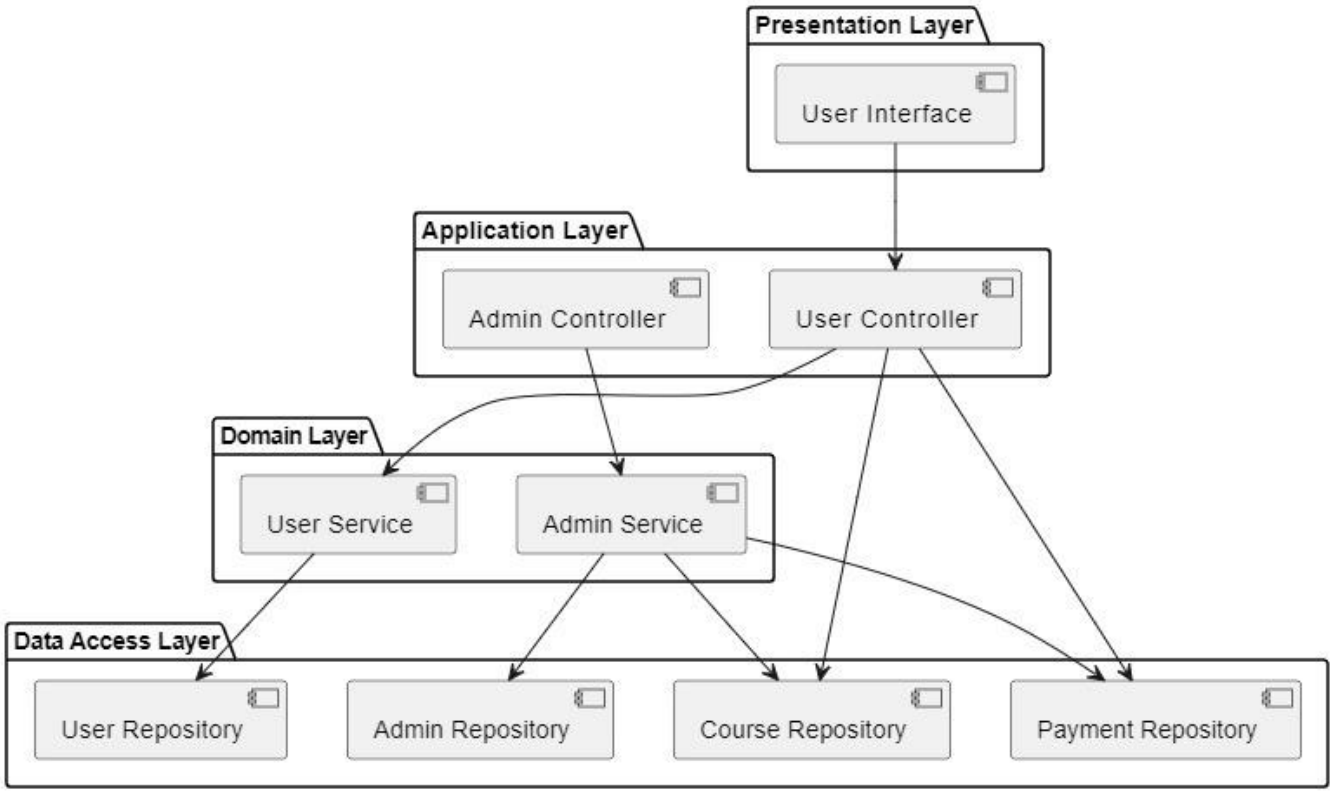
3. ***Availability and Reliability:***
    - Uptime: Maintain an uptime of 99.9%, ensuring high availability.
    - Backup and Recovery: Perform data backups every 12 hours and have a disaster recovery plan to restore services within 1 hour of a major failure.

4. ***Usability:***
    - User Interface: Ensure the interface is intuitive and user-friendly, allowing easy navigation with minimal training.
    - Accessibility: Comply with WCAG 2.1 standards to ensure the system is usable by people with disabilities.

5. ***Compatibility and Maintainability:***
    - Web and Device Compatibility: Ensure compatibility with popular web browsers (Chrome, Firefox, Safari) and responsive design for desktops, tablets, and smartphones.
    - Maintainability: Follow coding standards and best practices, and provide comprehensive documentation for all system components to facilitate maintenance and future upgrades.

OVERALL PROJECT ARCHITECTURE

EX_NO: 6 DATE: 09-04-24



1. ***User Interface (UI)***
    - Description: This package contains modules that interact directly with the users of the system.
    - Components:
        - Login Module: Manages user authentication and authorization.
        - Course Enrollment Module: Allows users to browse, select, and enroll in courses.
        - Payment Module: Facilitates payment for course enrollment.

2. ***Authentication & Authorization***
    - Description: This package handles the login process and ensures that users and admins are properly authenticated and authorized.
    - Components:
        - User Database: Stores user credentials and information.
        - Admin Database: Stores administrator credentials and information.
    - Interaction: The Login Module within the UI interacts with both User and Admin Databases to verify credentials.

3. ***Course Management***
    - Description: This package is responsible for creating, managing, and delivering courses.
    - Components:
        - Course Database: Stores details of all available courses.
        - Content Management System (CMS): Manages course content, including videos, quizzes, and reading materials.
    - Interaction: The Course Enrollment Module within the UI accesses the Course Database to display course information and the CMS to deliver content.

4. ***Payment Processing***
    - Description: This package deals with the financial transactions related to course payments.
    - Components:
        - Payment Gateway: External service for processing payments.
        - Payment Database: Stores details of completed transactions and payment status.
    - Interaction: The Payment Module within the UI communicates with the Payment Gateway for transaction processing and updates the Payment Database.

5. *Administration*

- Description: This package provides administrative functionalities, allowing administrators to manage courses, users, and generate reports.
- Components:
  - Admin Dashboard: Central interface for administrators to manage the system.
  - User Management Module: Manages user accounts and roles.
  - Course Management Module: Manages course creation and updates.
  - Reporting Module: Generates various reports related to courses and payments.
  - Course Database: Accessed for managing and reviewing course details.
  - Payment Database: Accessed for reviewing and updating payment records.
- Interaction: The Admin Dashboard interacts with various databases and modules to perform administrative tasks.

6. *Database*

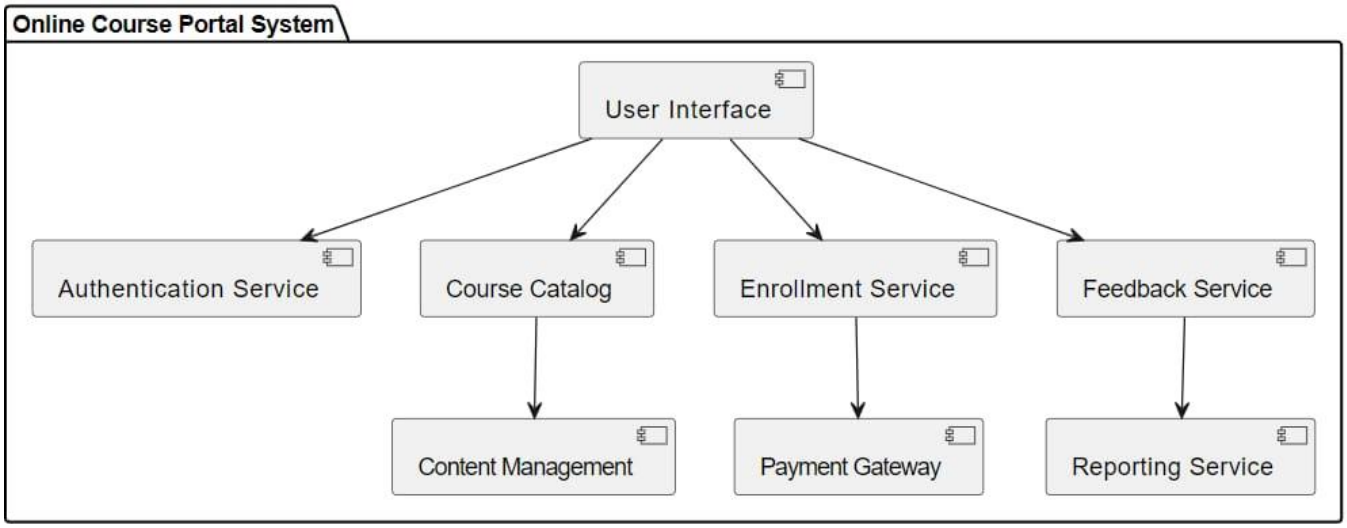
- Description: This package encompasses all the databases used by the system, containing essential data for its operation.
- Components:
  - User Database: Stores user-related information.
  - Admin Database: Stores administrator-related information.
  - Course Database: Contains records of all courses.
  - Payment Database: Contains records of all payments.
- Interaction: All other packages (Authentication & Authorization, Course Management, Payment Processing, and Administration) interact with the Database package to store and retrieve necessary data.

7. *Inter-Package Interactions*

- User Interface: Acts as the primary access point, interfacing with all core packages: Authentication & Authorization, Course Management, Payment Processing, and Administration.
- Course Management: Interfaces with the Database to manage and deliver courses.
- Payment Processing: Interfaces with the Payment Gateway for transactions and updates the Payment Database accordingly.

BUSINESS ARCHITECTURE DIAGRAM

EX_NO: 7 DATE: 09-04-24



1. ***Actors***
    - Admin: Manages courses, users, and oversees the system.
    - Student: Enrolls in and takes courses.
    - Instructor: Creates and manages course content.

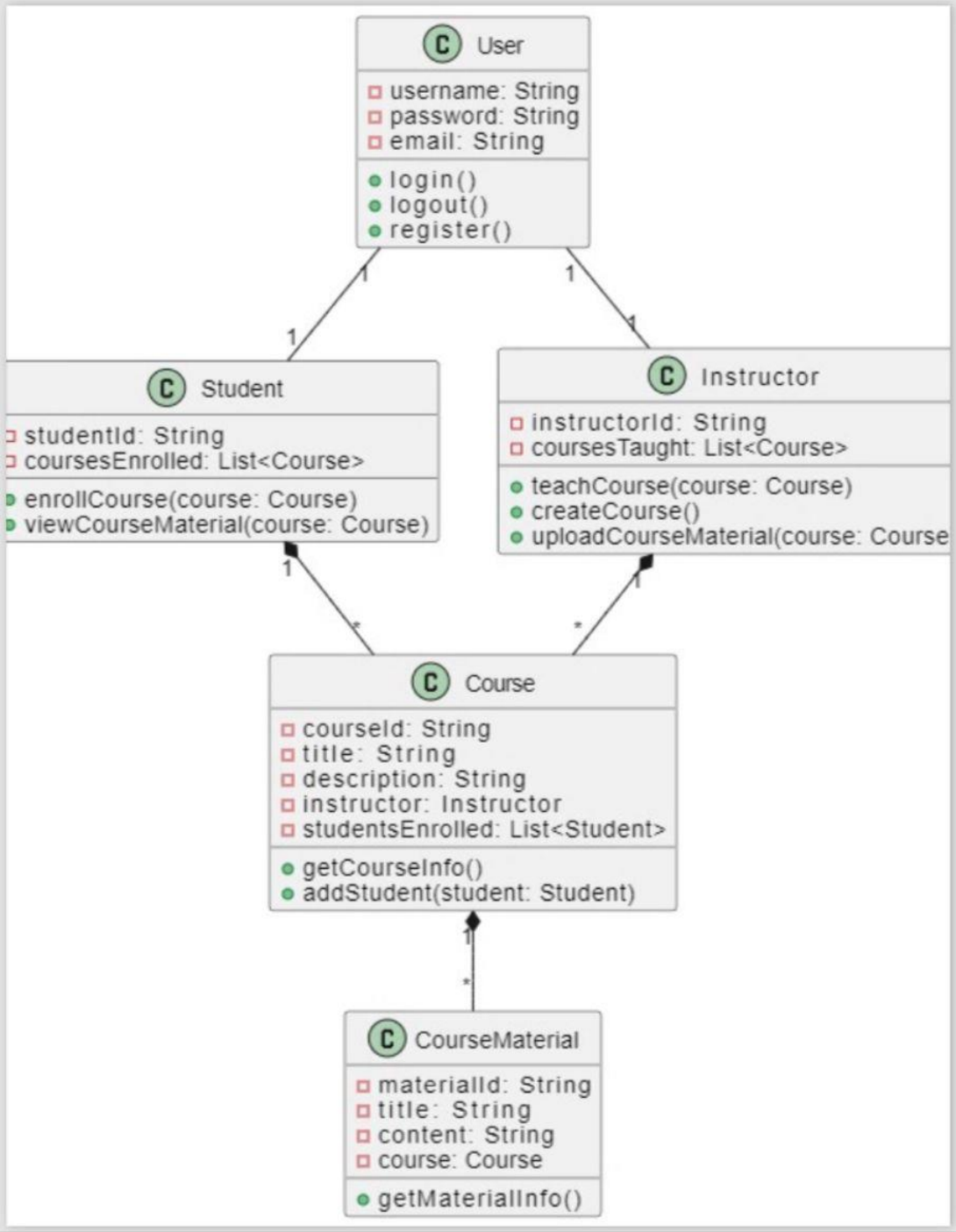
2. ***Frontend Use Cases***
    - Login: Log into the system.
    - Register: Register as a new user.
    - Dashboard: Main interface post-login.
    - Browse Courses: View available courses.
    - Enroll in Course: Enroll in courses.
    - Access Course Content: View and participate in course activities.
    - Submit Feedback: Provide feedback on courses.
    - Manage Course: Create and update course content.
    - View Reports: View reports on system usage, enrollments, and feedback.

3. ***Backend Use Cases***
    - Authenticate User: Handle user authentication.
    - Manage Users: Manage user data and registration.
    - Manage Courses: Create, update, and delete courses.
    - Store Course Data: Store course details and content.
    - Retrieve Course Data: Retrieve course details and content.
    - Process Enrollment: Manage student enrollments.
    - Handle Payment: Process payments for enrollments.
    - Send Notifications: Send updates about enrollments and courses.
    - Generate Reports: Produce reports on enrollments, feedback, and performance.

4. ***Relationships***
    - Admin, Student, and Instructor interact with various frontend use cases.
    - Each frontend use case has a corresponding backend functionality.
    - Include relationships (<<include>>) indicate actions involving multiple backend processes.

CLASS DIAGRAM

EX_NO: 8 DATE: 30-04-24



A class diagram is a type of UML(Unified Modeling Language) diagram that represents the structure and behavior of a system or application by illustrating the classes, attributes, methods, and relationships between objects. Class Diagrams are widely used in software development to visualize the static design of a system. The class diagram for the Online Course Portal System shows the main entities and their relationships, as follows:

1. ***User***
    - Attributes: username, password, email
    - Methods: login(), logout(), register()
    - Description: Represents generic users with common functionalities.

2. ***Student***
    - Attributes: studentId, coursesEnrolled
    - Methods: enrollCourse(), viewCourseMaterial()
    - Description: Extends User, representing students with specific methods for enrolling in and viewing courses.

3. ***Instructor***
    - Attributes: instructorId, coursesTaught
    - Methods: teachCourse(), createCourse(), uploadCourseMaterial()
    - Description: Extends User, representing instructors with methods for managing courses and materials.

4. ***Course***
    - Attributes: courseId, title, description, instructor, studentsEnrolled
    - Methods: getCourseInfo(), addStudent()
    - Description: Represents a course with details and lists of enrolled students and instructor.

5. ***CourseMaterial***
    - Attributes: materialId, title, content, course
    - Methods: getMaterialInfo()
    - Description: Represents materials associated with a course, like videos and readings.

6. ***Relationships***
    - User-Student and User-Instructor: Inheritance relationships showing that Student and Instructor are specialized types of User.
    - Course-Instructor: Each course is linked to one instructor.
    - Course-Student: Many-to-many relationship between courses and students.
    - Course-CourseMaterial: Composition relationship where each course includes multiple materials.

This structure allows efficient management of users, courses, and course materials within the online portal system.

SEQUENCE DIAGRAM

EX_NO: 9 DATE: 30-04-24

1. ***Login:***
    - User logs in by providing credentials.
    - CoursePortal verifies credentials with Database.

2. ***Validate Login:***
    - If credentials are valid:
    - CoursePortal grants access.
    - User can browse courses.
    - If credentials are invalid:
    - CoursePortal returns an error.

3. ***Browse Courses:***
    - User requests course list.
    - CoursePortal fetches list from CourseCatalog and displays it.

4. ***Enroll in a Course:***
    - User selects a course to enroll.
    - CoursePortal processes enrollment with Database.

5. ***Process Payment:***
    - User initiates payment.
    - CoursePortal processes payment with PaymentGateway.

6. ***Update User's Courses:***
    - CoursePortal confirms payment and updates user's courses in Database.

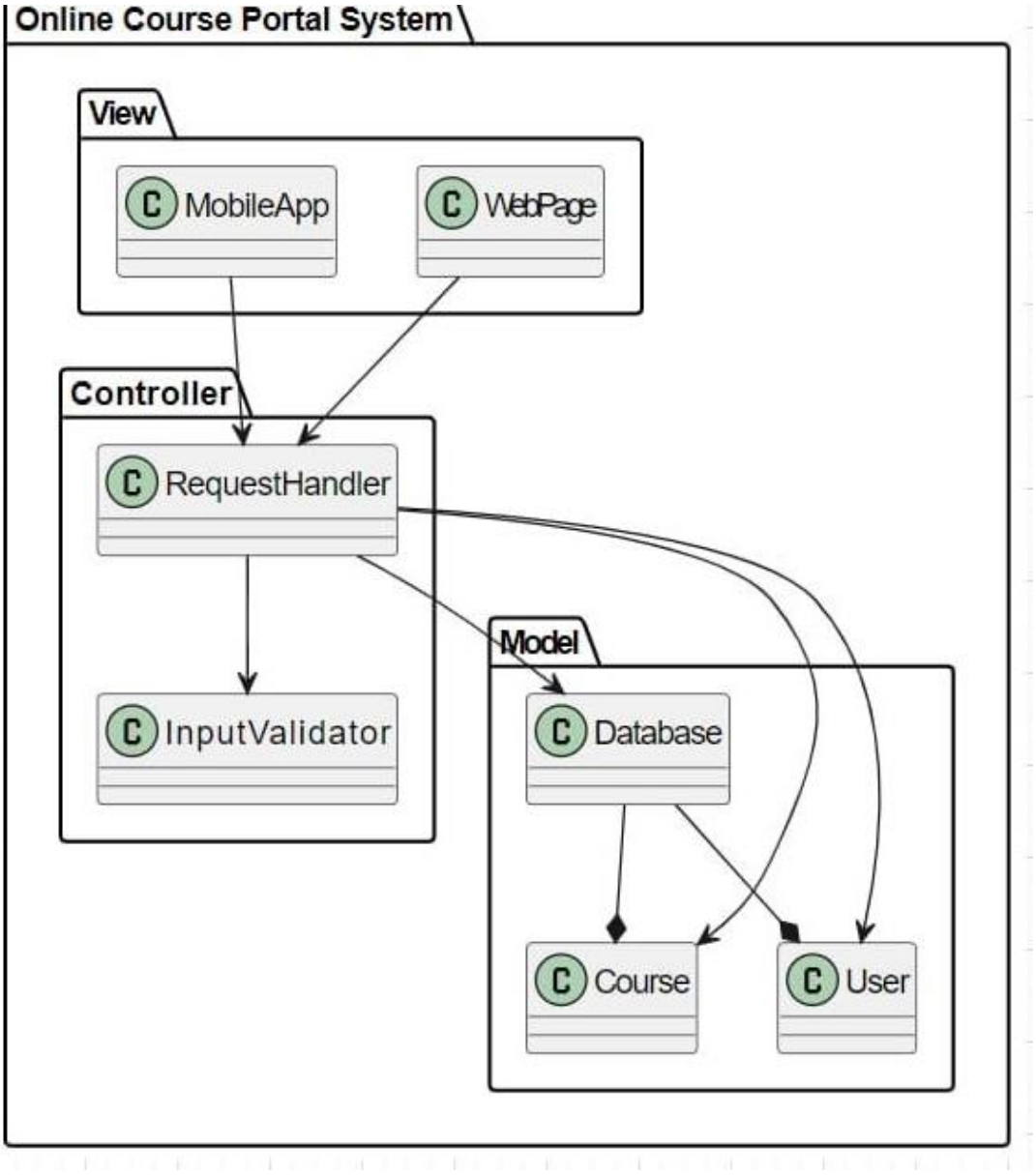
7. ***Course Enrollment Confirmation:***
    - CoursePortal sends confirmation to User.

8. ***Invalid Credentials:***
    - If login fails, CoursePortal returns an error message to User.

9. ***Logout:***
    - User logs out of the system.

ARCHITECTURAL PATTERN (MVC)

EX_NO: 10 DATE: 30-04-24



1. ***Model***
    - Course: Manages course data (e.g., courseID, courseName, courseFee). Methods: createCourse(), getCourseDetails(), updateCourse(), deleteCourse().
    - User: Manages user data (e.g., userID, username, email). Methods: createUser(), getUserDetails(), updateUser(), deleteUser().
    - Database: Central storage for courses and users. Methods: connect, execute queries, manage transactions.

2. View
    - MobileApp: Displays course details, user profile, and enrollment status.
    - WebPage: Displays course list, course details, and user information.

3. Controller
    - RequestHandler: Routes requests from views to models.
    - InputValidator: Validates user inputs (e.g., credentials, course selection, payment details).

4. Interactions
    - User Login:
        - User logs in via MobileApp or WebPage.
        - RequestHandler validates credentials with InputValidator.
        - Database fetches user details and displays them via the view.

5. Browse Courses:
    - User requests course list.
    - RequestHandler fetches course data from Database.
    - View displays the course list.

6. Enroll in a Course:
    - User selects and enrolls in a course.
    - RequestHandler validates selection with InputValidator.
    - Database updates enrolled courses.
    - View confirms enrollment.

7. Process Payment:
    - User initiates payment.
    - RequestHandler validates and processes payment.