OLS : Online Learning System

Project Description

The Online Learning System is a web-based platform designed to enhance the learning experience by providing a structured and accessible environment for students, instructors, and administrators. Built using ASP.NET Core MVC and SQL Server, the system enables students to enroll in courses, take quizzes, and earn certificates. Instructors can create and manage courses, while administrators oversee the entire platform through user and course management functionalities. The project aims to provide a scalable, role-based system that supports interactive learning and efficient course delivery. The outcomes include a functional system with secure authentication, streamlined course management, and integrated payment and reporting features that collectively support modern e-learning needs.

Team Members

Ahmed Mostafa Mohamed Mostafa ID:22100920

Mohamed Ahmed Saed Hasonah ID:22101312

Github

https://github.com/AhmedxMostafa1/OnlineLearningS

📖 Introduction

Background and Motivation

With the rapid digital transformation in education, online learning platforms have become essential tools for delivering knowledge efficiently. Traditional learning methods are limited by physical presence and scheduling constraints, while e-learning systems provide flexibility and accessibility to a wider audience. The motivation for this project stems from the increasing demand for affordable, accessible, and user-friendly platforms where students, instructors, and administrators can interact seamlessly.

Problem Statement

Existing online learning platforms are often either too complex or too limited in functionality, making them less suitable for small institutions or independent instructors. Key challenges include lack of role-based access, limited content management features, and poor integration of assessments . There is a need for a system that balances functionality with usability while ensuring scalability for future growth.

Objectives of the Project

The main objectives of the Online Learning System are:

To design and implement a web-based platform for managing courses, students, and instructors.

To provide role-specific access: students (enroll & learn), instructors (create courses), and admins (manage the platform).

To integrate quizzes for effective assessment of learning outcomes.

To ensure secure user authentication and support payment integration.

To provide an admin dashboard with insights into users, courses, and financials.

Scope of the System

The system targets students seeking flexible learning opportunities, instructors wanting to deliver content online, and administrators managing the platform. The scope covers:

Students: Course enrollment, quizzes, progress tracking,payments.

Instructors: Course creation, content management, quiz design, and student progress tracking.

Admins: User management, instructor approval, course and enrollment oversight, and payment tracking.

The project is scoped for small-to-medium institutions or training providers but is scalable to support larger deployments in the future.

Functional Requirements:

1 Authentication & Authorization
1.1 The system shall support user registration and login using email and password.
1.2 The system shall provide role-based access control with three roles: 1: Student 2: Instructor 3: Admin

2. Student Features
2.1 Students shall be able to browse all available courses..
2.2 Students shall be able to enroll in free or available courses.
2.3 Students shall be able to access course content (videos, PDFs, quizzes).

3. Instructor Features
3.1 Instructors shall be able to create new courses.
3.2 Instructors shall be able to upload course content, including: Video links and PDFs
3.3 Instructors shall be able to edit or delete their own courses.

4. Admin Features
4.1 Admins shall be able to manage users (students, instructors).
4.2 Admins shall be able to view all courses.
4.3 Admins shall have access to a dashboard showing basic metrics.

5. Course Management
5.1 Each course shall have: Title, Description, Instructor, Category, Content.
5.2 The system shall support categories (e.g., IT, Business, Design).
5.3 Each course shall consist of modules, and each module shall include lessons.
5.3 Each course module shall include a quiz.
5.3 The system shall support payment integration for premium courses.

Non-Functional Requirements:

1.User-friendly interface for ease of use

2.Secure data storage and protection of user information

3.The system shall be built using ASP.NET Core MVC.

4.The system shall use SQL Server as the database.

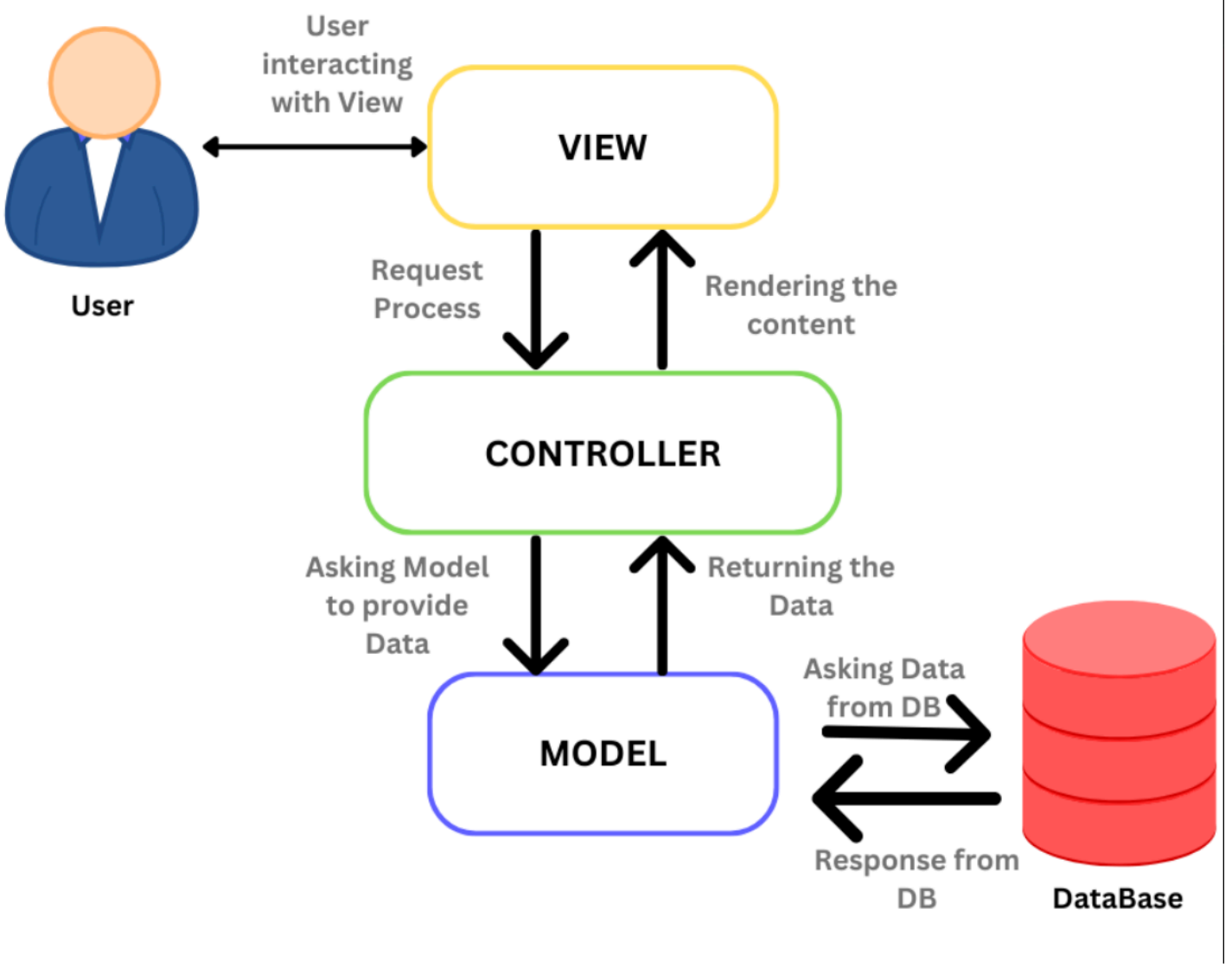
5.The system shall use Entity Framework as the ORM.

6.The system shall follow a layered architecture (separating UI, business logic, and data access).

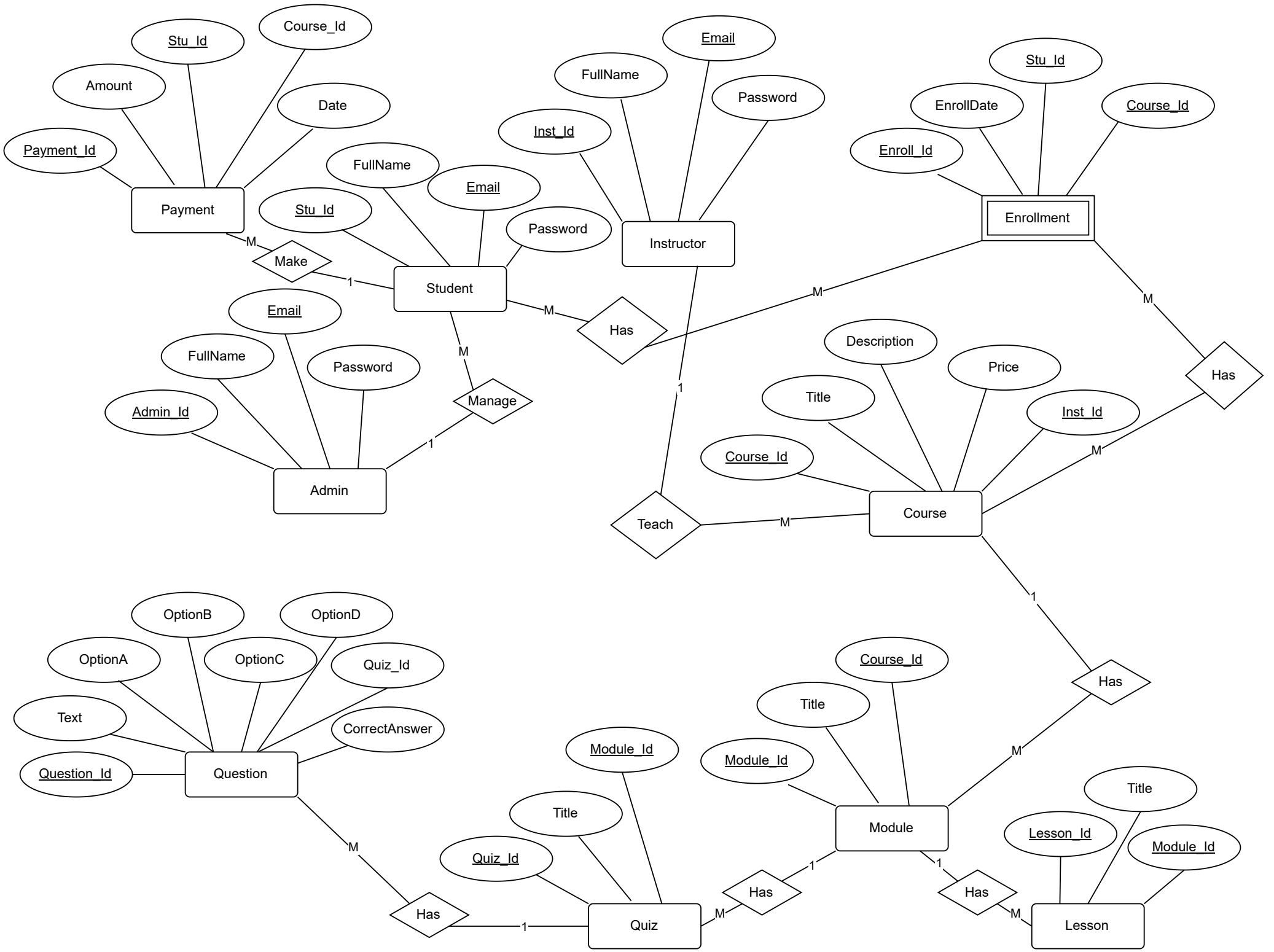
7.The system shall follow best practices for security (password hashing, input validation).

Use Case Diagram

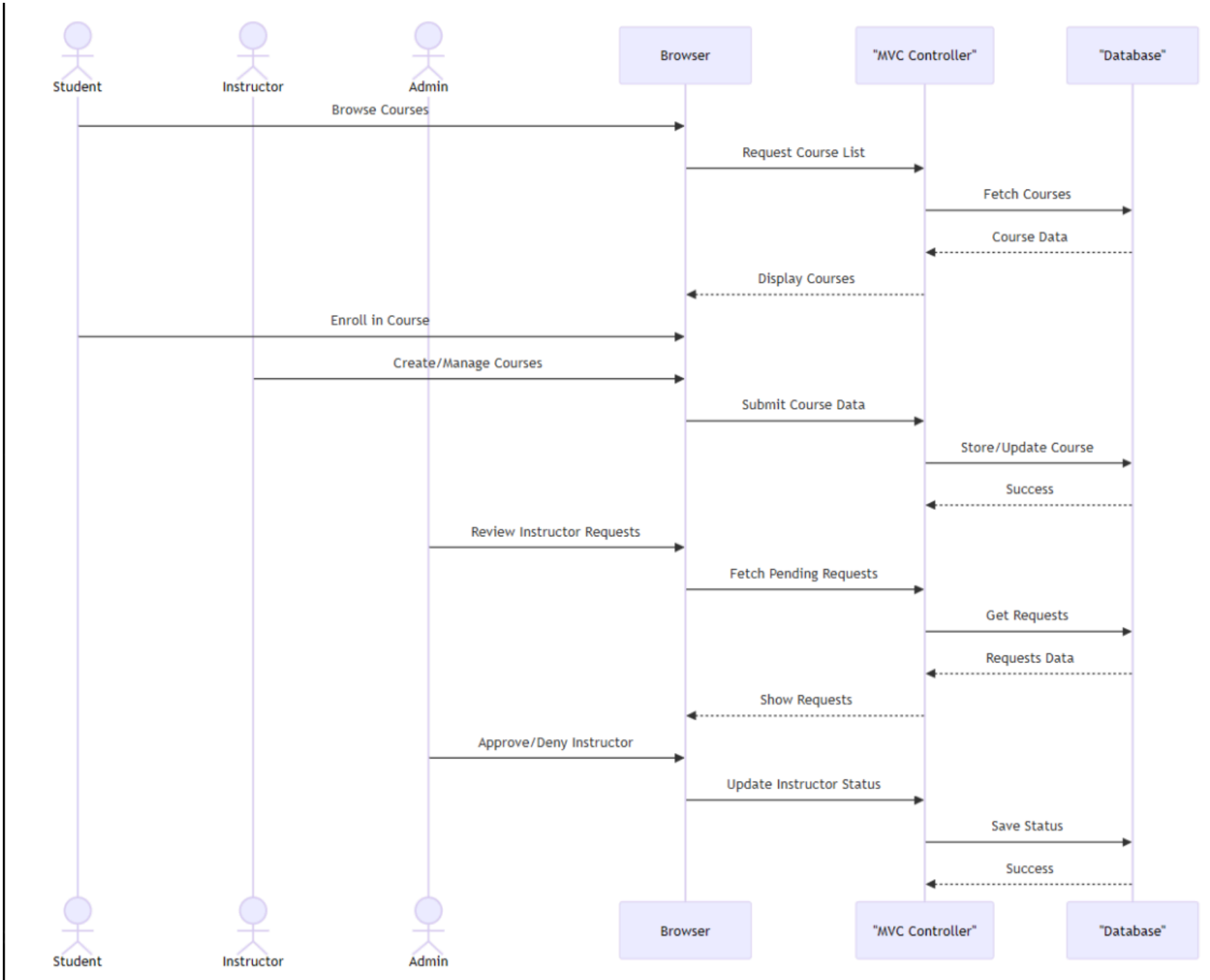
Architecture (MVC structure)



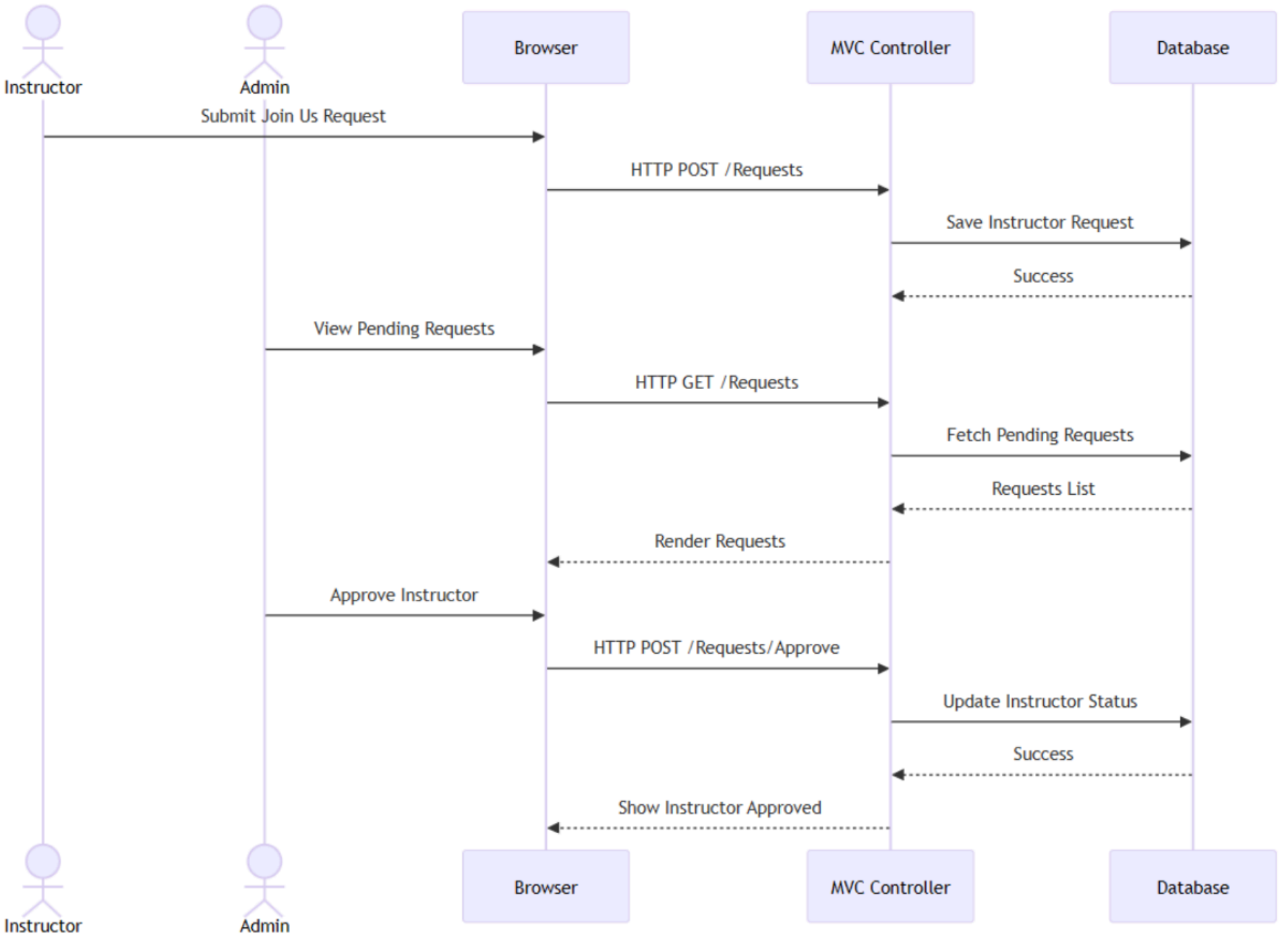
ERD

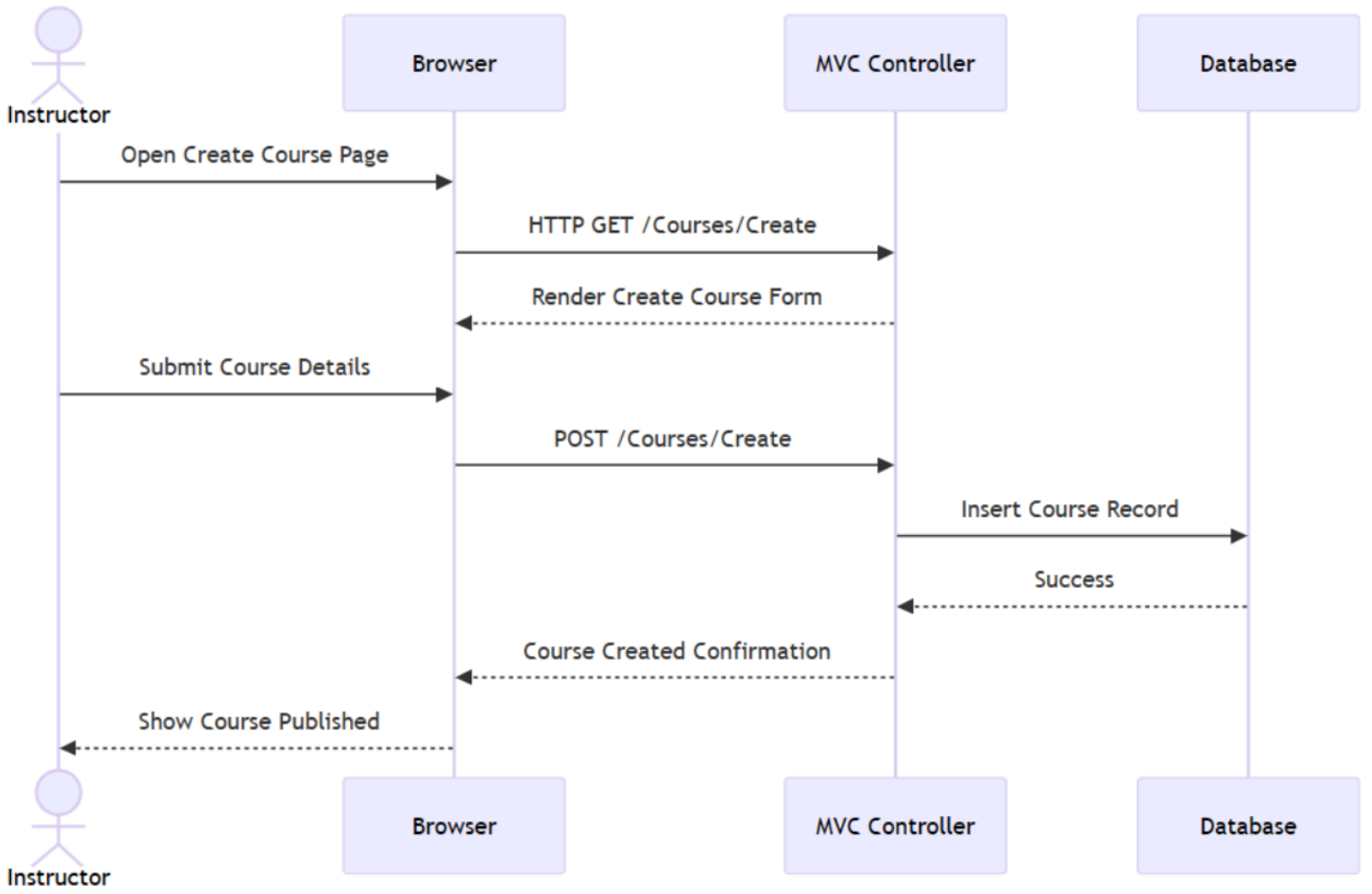

Stu_Id
Course_Id
Amount
Date
Payment_Id
Payment
M
Make
1
Email
FullName
Stu_Id
Password
Student
M
Has
M
Email
FullName
Password
Admin_Id
Manage
1
Admin
Email
FullName
Password
Inst_Id
Instructor
1
Stu_Id
EnrollDate
Course_Id
Enroll_Id
Enrollment
M
M
Has
Description
Price
Title
Inst_Id
Course_Id
Course
M
Teach
M
1
OptionB
OptionD
OptionA
OptionC
Quiz_Id
Text
CorrectAnswer
Question_Id
Question
M
Has
1
Course_Id
Title
Module_Id
Module_Id
Module
M
Has
1
Has
1
Title
Lesson_Id
Module_Id
Lesson
M
Title
Quiz_Id
Quiz
M

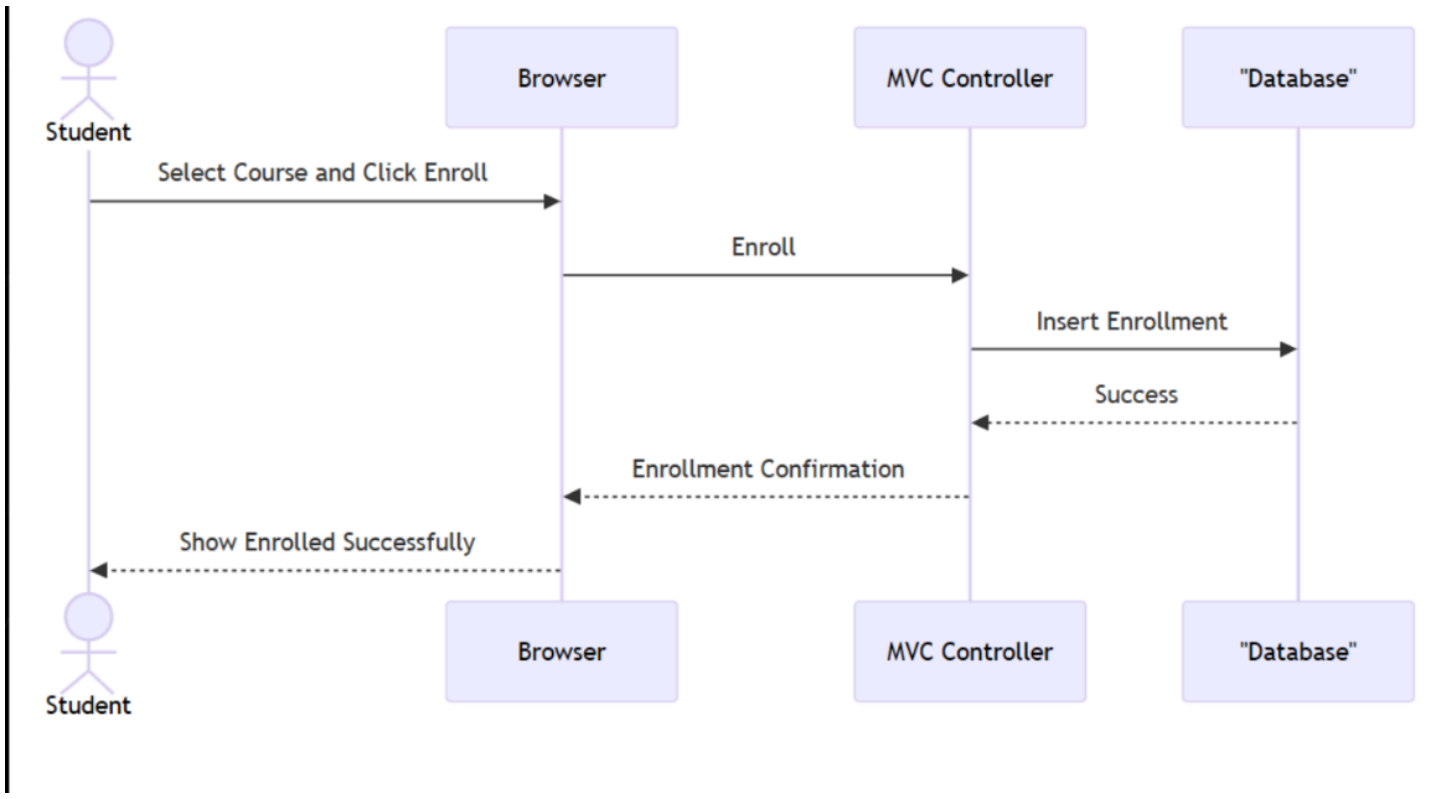
Sequence diagrams



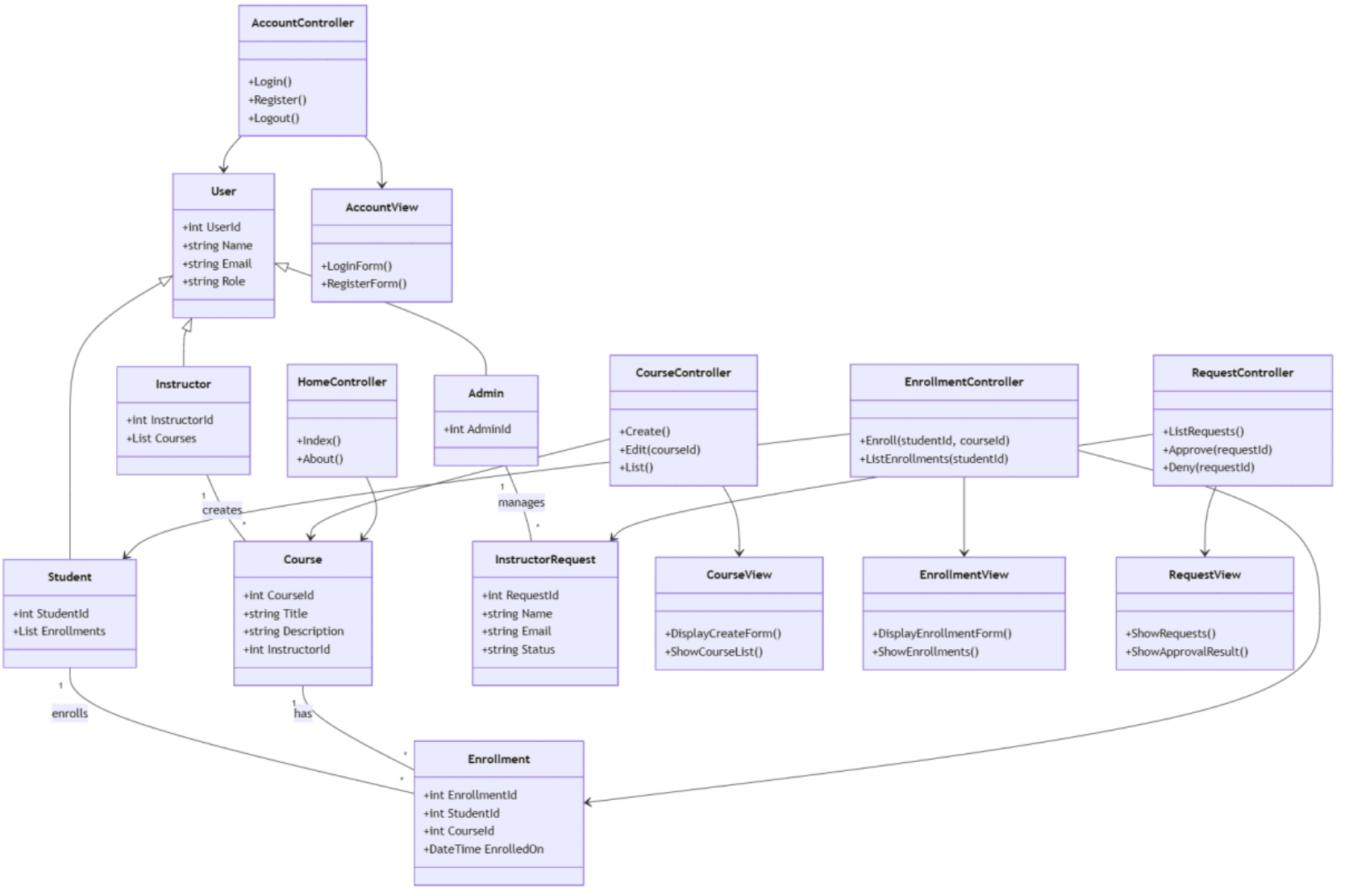
Sequence diagrams




Instructor
Browser
MVC Controller
Database
Open Create Course Page
HTTP GET /Courses/Create
Render Create Course Form
Submit Course Details
POST /Courses/Create
Insert Course Record
Success
Course Created Confirmation
Show Course Published
Instructor
Browser
MVC Controller
Database

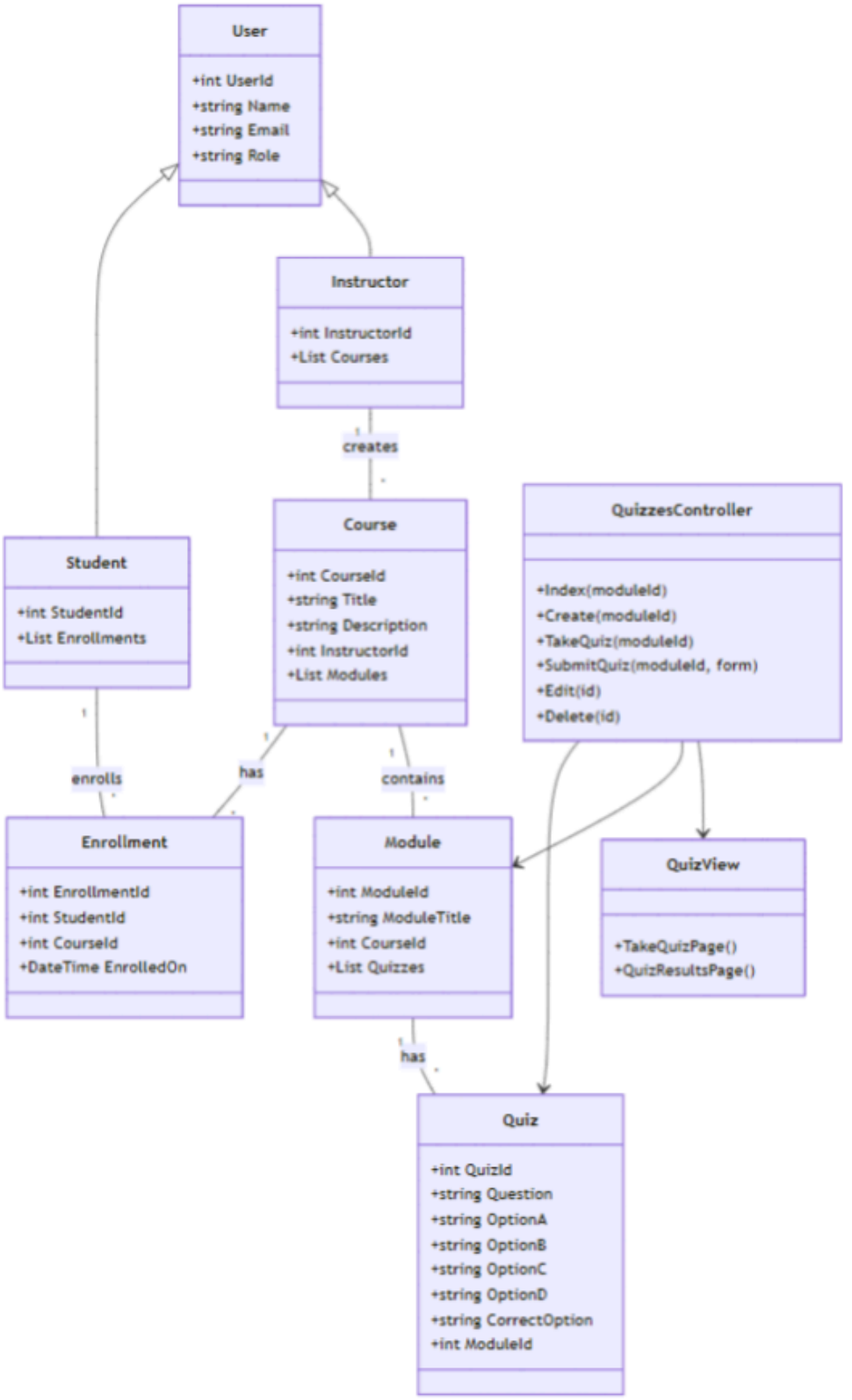

Student
Browser
MVC Controller
"Database"
Select Course and Click Enroll
Enroll
Insert Enrollment
Success
Enrollment Confirmation
Show Enrolled Successfully
Student
Browser
MVC Controller
"Database"

Class diagram

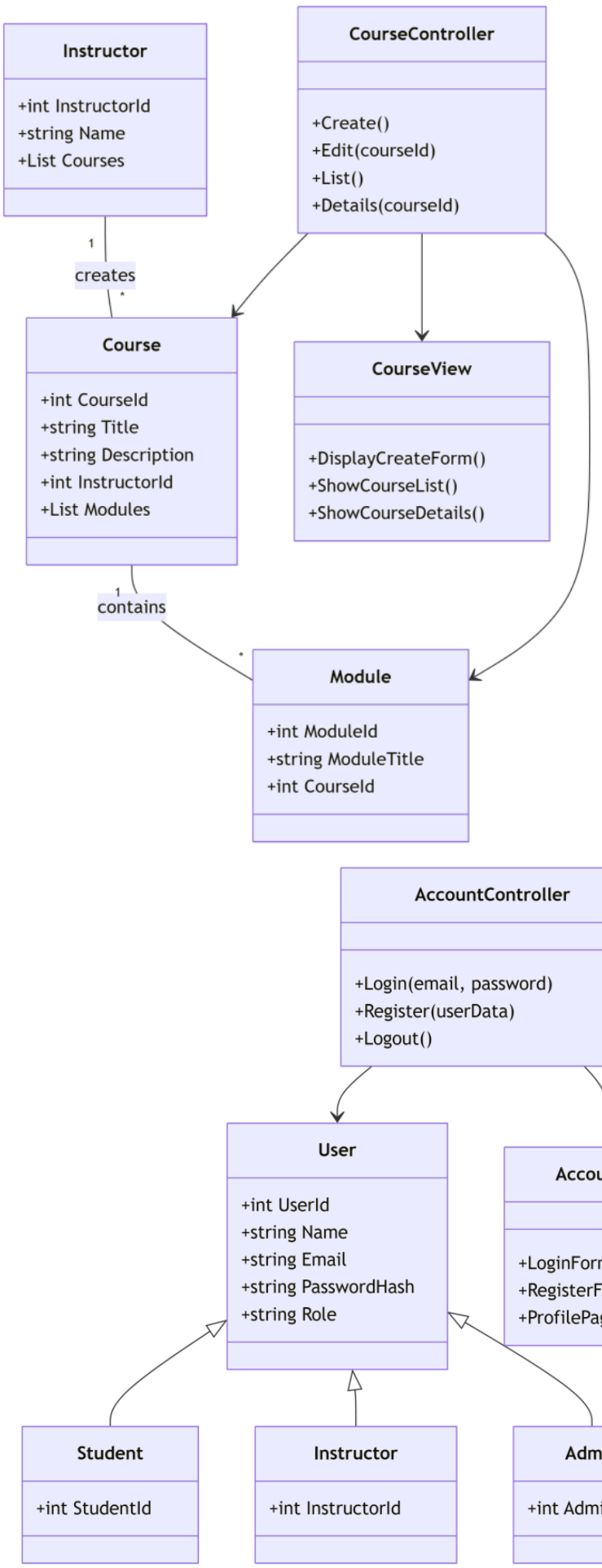


Class diagram

quizzes

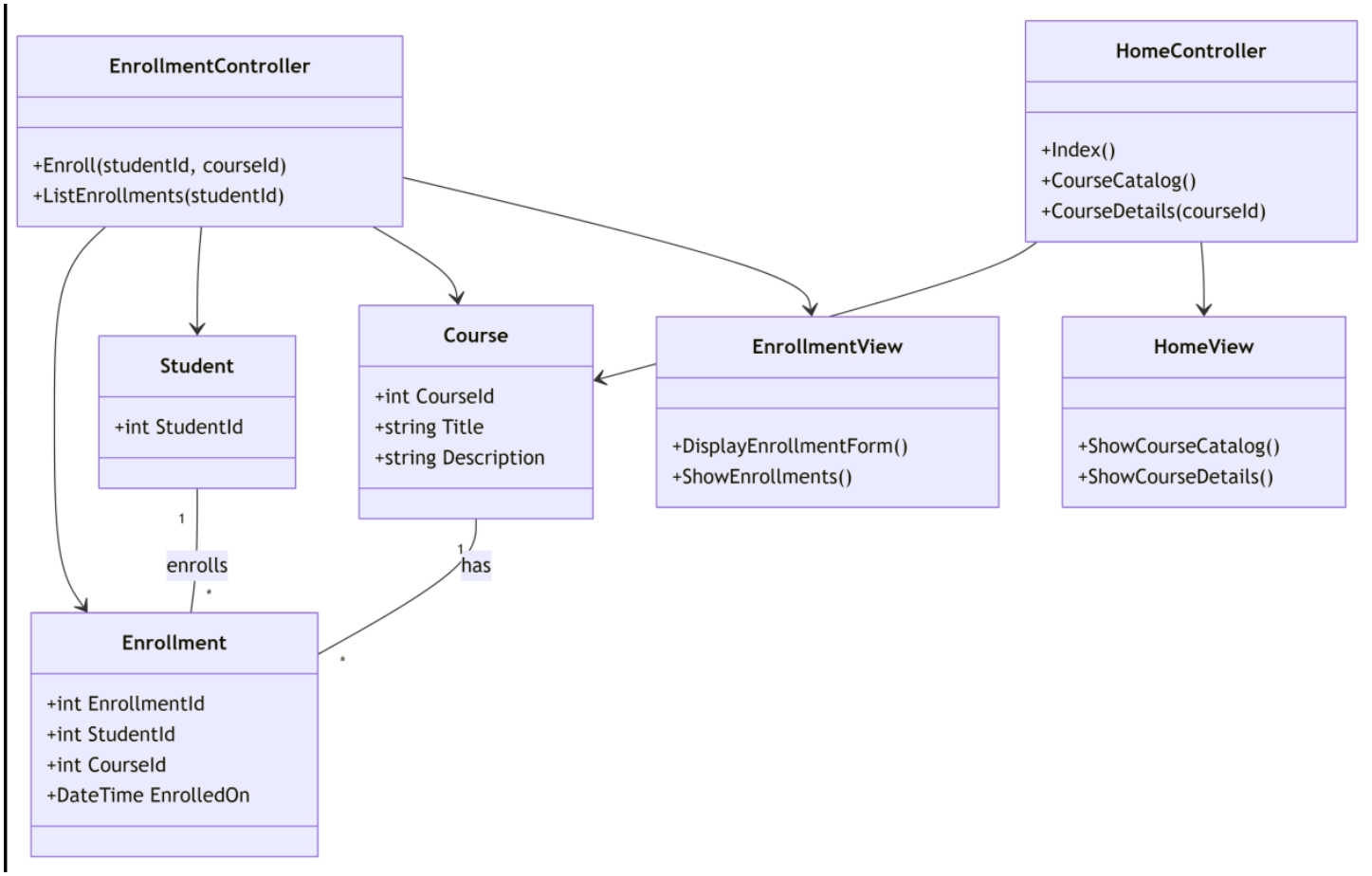


course creation

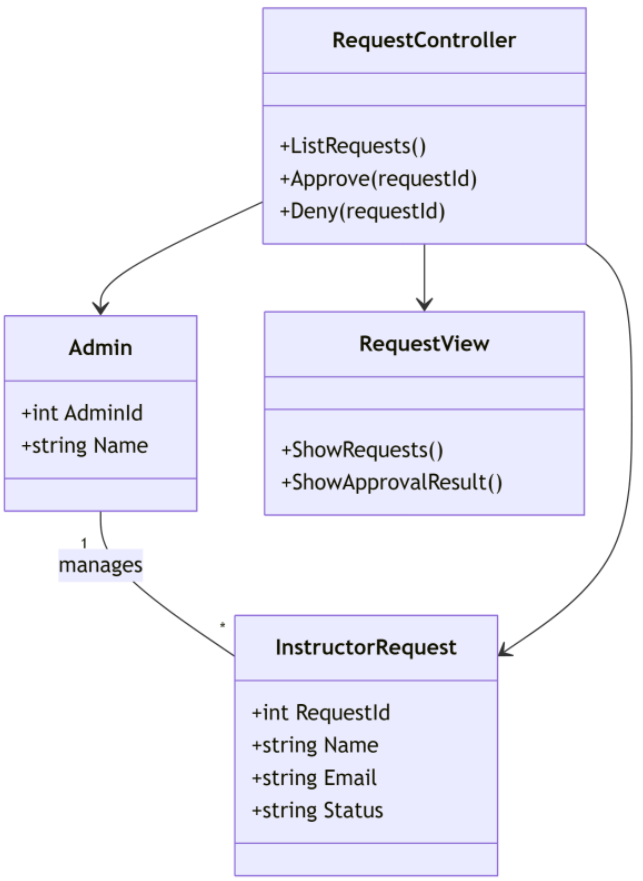


Account Creation

course enrollment



admin approve instructor



Implementation

Tech Stack Used

- Framework: ASP.NET Core MVC (Model-View-Controller pattern for clean separation of concerns).
- Backend: C# with .NET 6/7 (depending on project target).
- Database: Microsoft SQL Server.
- ORM: Entity Framework Core (for database-first approach, handling CRUD and relationships).
- Views/UI: Razor Pages and Razor Views for rendering dynamic content.
- Frontend Styling: Bootstrap (responsive UI), CSS.
- Authentication & Authorization: ASP.NET Identity for role-based access (Student, Instructor, Admin).
- Version Control: GitHub for collaboration and versioning.

Project Structure Explanation

The project follows the ASP.NET Core MVC architecture:

1.**Models**

- Represent the application's data (Users, Courses, Enrollments, Quizzes, etc.).
- Entity Framework Core maps these classes to SQL Server tables.

2.**Controllers**

- Handle requests, process business logic, and interact with Models.
- Example: CourseController manages course creation and listing.

3.Views

- Razor Views are used to render UI pages.
- Example: CourseView shows course lists and details.

Key Modules

1.User Management (Accounts & Roles)

- Registration, login, and logout system.
- Role-based access: Student, Instructor, Admin.
- Admin approves instructor requests.

2.Courses

- Instructors can create, edit, and manage their own courses.
- Students can browse, view, and enroll in courses.
- Supports course modules and details pages.

3.Quizzes

- Instructors can create quizzes linked to their courses.
- Questions and multiple-choice options included.
- Students can attempt quizzes, and their attempts are scored.

Results & Discussion

The developed **Online Learning Management System** successfully provides a structured platform for managing online courses, instructors, students, and quizzes. The system was implemented using **ASP.NET Core MVC** with **Entity Framework Core** for data access and **SQL Server** as the database. Razor Views and Bootstrap were used to create a responsive and user-friendly interface.

During testing, the system demonstrated:

- Correct handling of user registration and login (students, instructors, admins).
- Course management features where instructors could create, edit, and update courses.
- Student enrollment functionality with automatic record creation in the database.
- Quiz module, allowing instructors to create quizzes and students to attempt them.
- Admin panel where admin users can manage instructor requests and oversee the platform.

These results indicate that the system fulfills its main objectives of supporting digital learning workflows.

Achievements

1. Multi-role system: Clear separation of roles (Admin, Instructor, Student) with different privileges.
2. Course lifecycle management: From creation by instructors to enrollment by students.
3. Quiz integration: Basic assessment features for student evaluation.
4. Admin-controlled instructor approval: Prevents unauthorized course creation.
5. Scalable architecture: Built on ASP.NET Core MVC, making it extensible and maintainable.
6. Modern UI: Implemented with Bootstrap for responsive design.

Limitations

1. Limited quiz functionality: Only supports basic question/answer structure (no advanced types like drag-and-drop, coding tests, etc.).
2. No integrated payment system: While courses can be created and enrolled in, real-world monetization (via Stripe/PayPal) is not yet implemented.
3. Basic reporting: The system lacks advanced analytics (e.g., student performance tracking, instructor revenue summaries).
4. Email/notifications not integrated: Users are not notified about approvals, enrollments, or quiz results.
5. Deployment focused on local setup: Cloud-based CI/CD pipelines and scalability testing remain for future work.

Conclusion

This project successfully demonstrates the design and implementation of a web-based learning management platform. The use of ASP.NET Core MVC ensured a modular and maintainable structure, while Entity Framework Core simplified database interactions. The system achieves its goal of enabling students to enroll in courses, instructors to manage content, and admins to regulate platform activity. While some features such as payment integration, advanced quizzes, and analytics remain as future improvements, the current version serves as a solid foundation for an online learning platform.

In conclusion, the Online Learning System highlights the potential of combining modern web frameworks with database-driven architectures to create scalable and user-centered applications for education.