

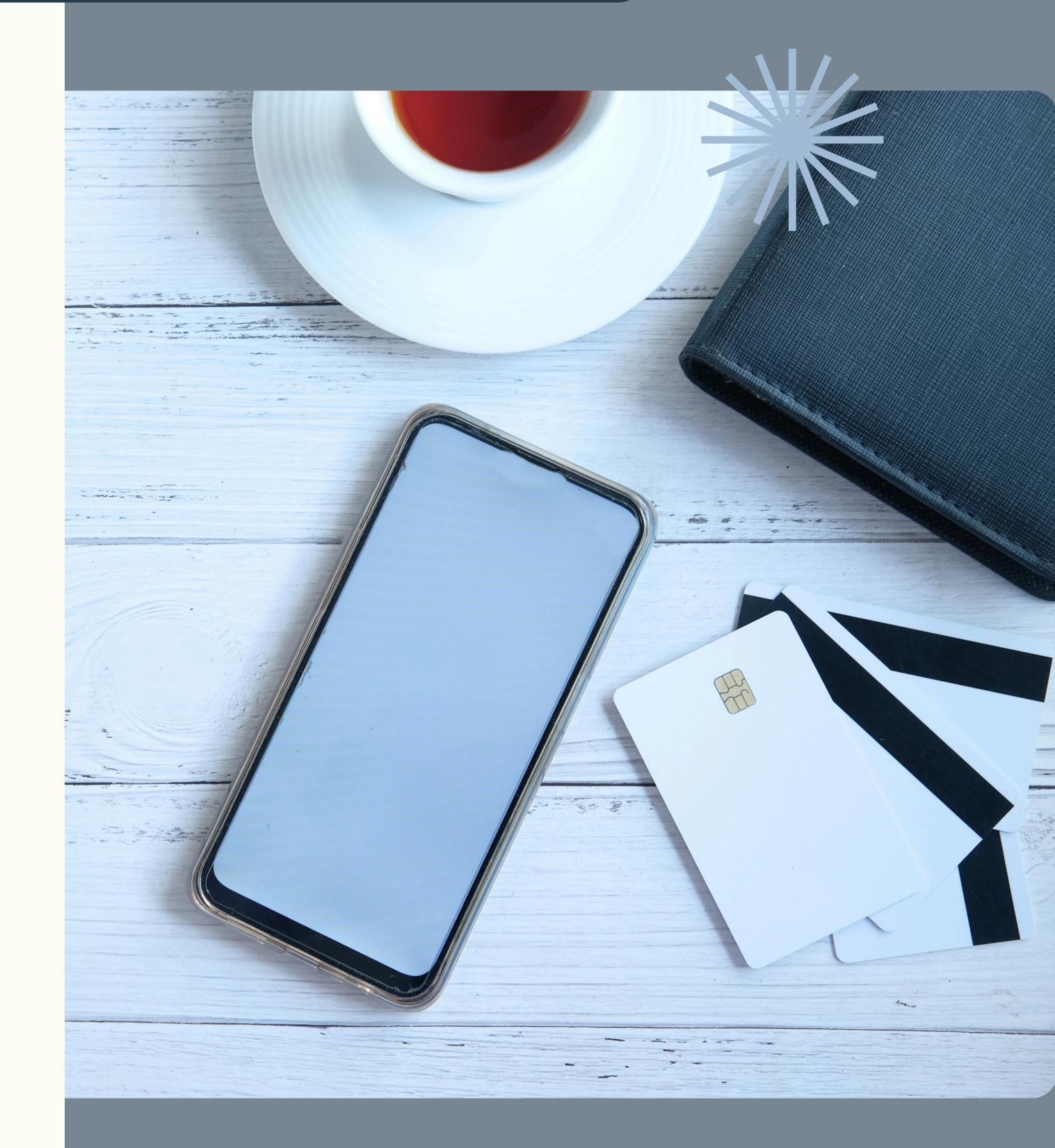
HỌC PHẦN PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM

ĐỀ TÀI: HỆ THỐNG QUẢN LÝ SINH VIÊN

1. Nguyễn Cao Chiến - 22010014
2. Phạm Hoàng Anh - 22010477
3. Nguyễn Đức Anh - 22010353
4. Nguyễn Hoàng Sơn - 22010249

 NHÓM 3

GV hướng dẫn: Ths. Vũ Quang Dũng



START PRESENTATION

CONTENT

01

GIỚI THIỆU ĐỀ TÀI

02

MÔ HÌNH USE CASE

03

PHÂN TÍCH

04

THIẾT KẾ

05

CÀI ĐẶT

06

DEMO

07

KẾT LUẬN

1. Giới thiệu đề tài

1.1 Mô tả vấn đề

- Việc quản lý thông tin sinh viên tại các trường đại học gặp nhiều khó khăn do dữ liệu lớn, dễ sai sót và mất thời gian xử lý. Phương pháp thủ công hoặc hệ thống rời rạc gây trùng lặp, khó tra cứu và thiếu tự động hóa, làm giảm hiệu suất làm việc.
- Cần một hệ thống phần mềm quản lý sinh viên để tối ưu quy trình, giảm sai sót và nâng cao hiệu quả. Hệ thống này giúp lưu trữ, tra cứu, xử lý dữ liệu khoa học, đồng thời hỗ trợ phân tích và báo cáo, giúp nhà trường quản lý hiệu quả hơn.



1. Giới thiệu đề tài

1.2 Giải pháp

Xây dựng hệ thống lưu trữ và quản lý thông tin sinh viên tập trung, đảm bảo an toàn và dễ dàng truy cập.

- **Quản lý thông tin sinh viên:** Cập nhật thông tin cá nhân, học lực, và lịch sử học tập của từng sinh viên.
- **Đăng ký môn học:** Sinh viên có thể tự chọn và đăng ký các môn học theo kế hoạch.
- **Theo dõi kết quả học tập:** Sinh viên và nhà quản lý có thể xem điểm số và kết quả học tập chi tiết.



1. Giới thiệu đề tài

1.3 Bảo mật

- Phân quyền truy cập rõ ràng và bảo vệ thông tin quan trọng.
- Hệ thống hoạt động ổn định, xử lý nhanh các tác vụ, giúp công việc luôn đảm bảo liên tục và hiệu quả.

1.4 Thông số kỹ thuật bổ sung

- Hiệu năng
- Tốc độ
- Độ tin cậy
- Tính an toàn và bảo mật
- Khả năng bảo trì
- Khả năng sử dụng
- Khả năng mở rộng
- Tính khả dụng

2. Mô hình Use Case

2.1 Các yêu cầu chức năng



Chức năng dành cho Sinh viên

- **Đăng ký môn học:** Sinh viên có thể chọn và đăng ký môn học cho các kỳ học.
- **Tra cứu điểm:** Sinh viên có thể xem điểm của mình trong hệ thống.
- **Quản lý thông tin cá nhân:** Sinh viên có thể cập nhật hoặc xem thông tin cá nhân của mình.

2. Mô hình Use Case

2.1 Các yêu cầu chức năng



Chức năng dành cho Quản lý Khoa:

- **Quản lý thông tin sinh viên:** Quản lý khoa có thể theo dõi và cập nhật thông tin sinh viên.
- **Quản lý điểm sinh viên:** Quản lý khoa có quyền nhập hoặc chỉnh sửa điểm số của sinh viên.

2. Mô hình Use Case

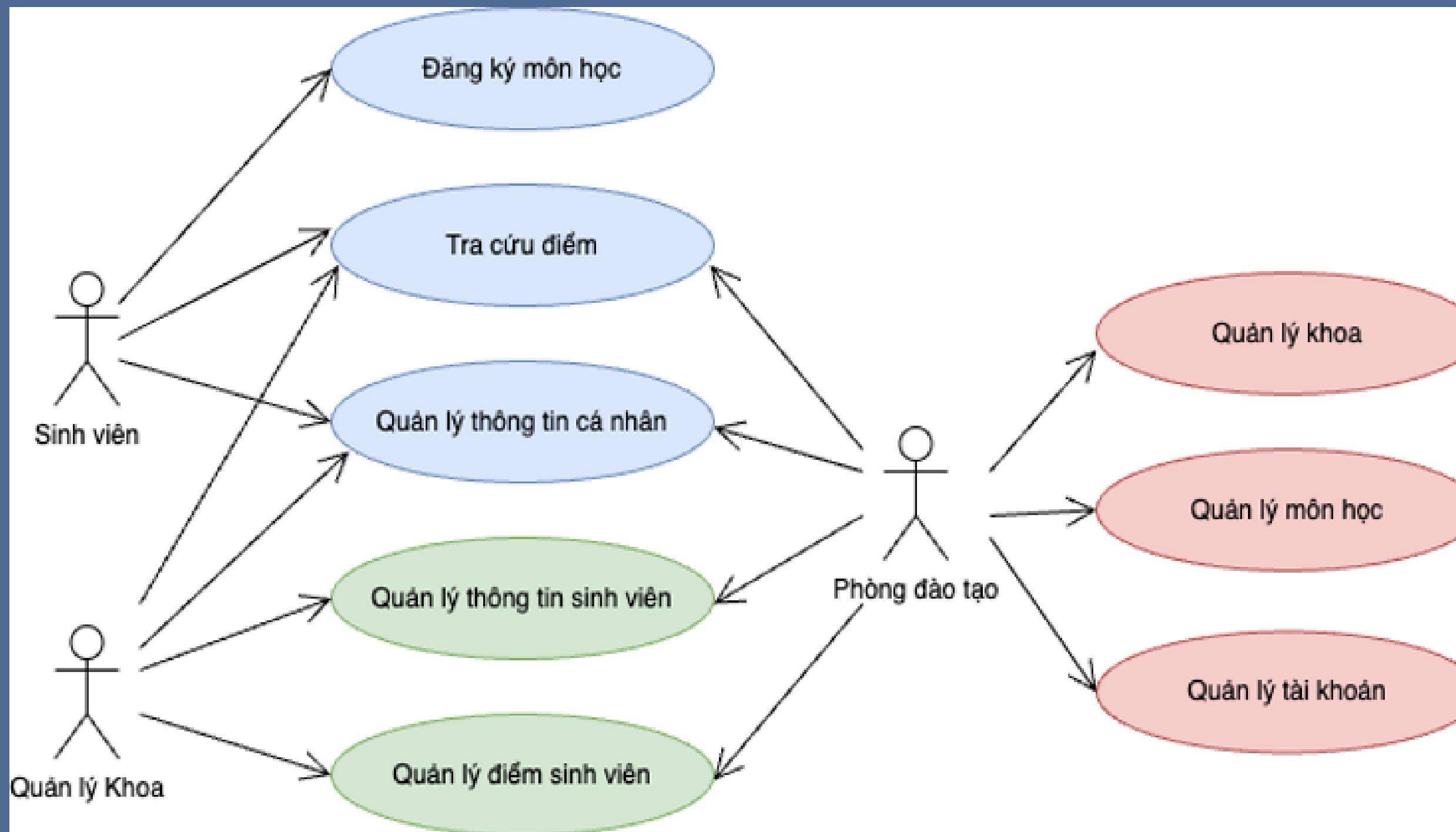
2.1 Các yêu cầu chức năng



Chức năng dành cho Phòng Đào Tạo:

- **Quản lý khoa:** Chịu trách nhiệm quản lý thông tin các khoa trong trường.
- **Quản lý môn học:** Chịu trách nhiệm thêm, sửa, xóa, và quản lý danh sách môn học.
- **Quản lý tài khoản:** Chịu trách nhiệm quản lý tài khoản của sinh viên, giảng viên, và các nhân viên liên quan.

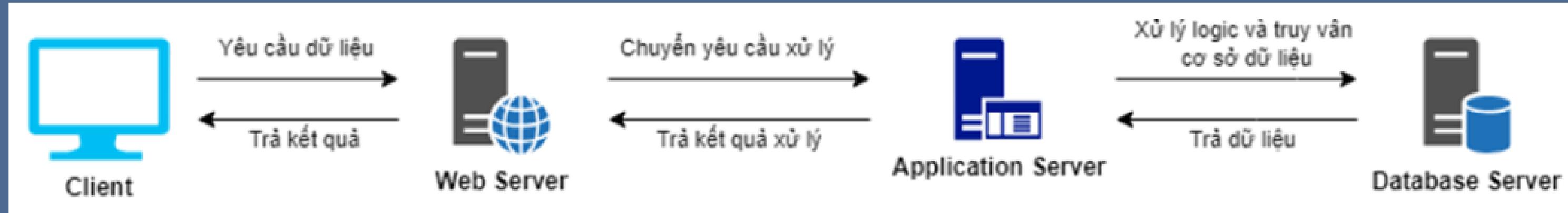
2. Mô hình Use Case



SƠ
ĐỒ
USE
CASE

3. Phân tích

3.1 Phân tích kiến trúc hệ thống



Kiến trúc mức cao của hệ thống

VAI TRÒ	Client	Web Server	Application Server	Database Server
	Thiết bị đầu cuối người dùng (PC, điện thoại) gửi yêu cầu và nhận kết quả từ hệ thống.	Cầu nối giữa Client và Application Server, tiếp nhận yêu cầu HTTP và phản hồi.	Xử lý nghiệp vụ chính và điều phối yêu cầu đến Database Server.	Lưu trữ dữ liệu hệ thống, đảm bảo truy xuất và cung cấp dữ liệu chính xác.

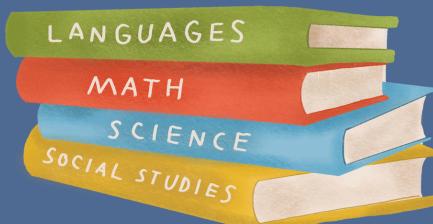
3. Phân tích

3.1 Phân tích kiến trúc hệ thống

Các đối tượng trừu tượng chính của hệ thống



Sinh viên



Môn học



Điểm số



Khoa



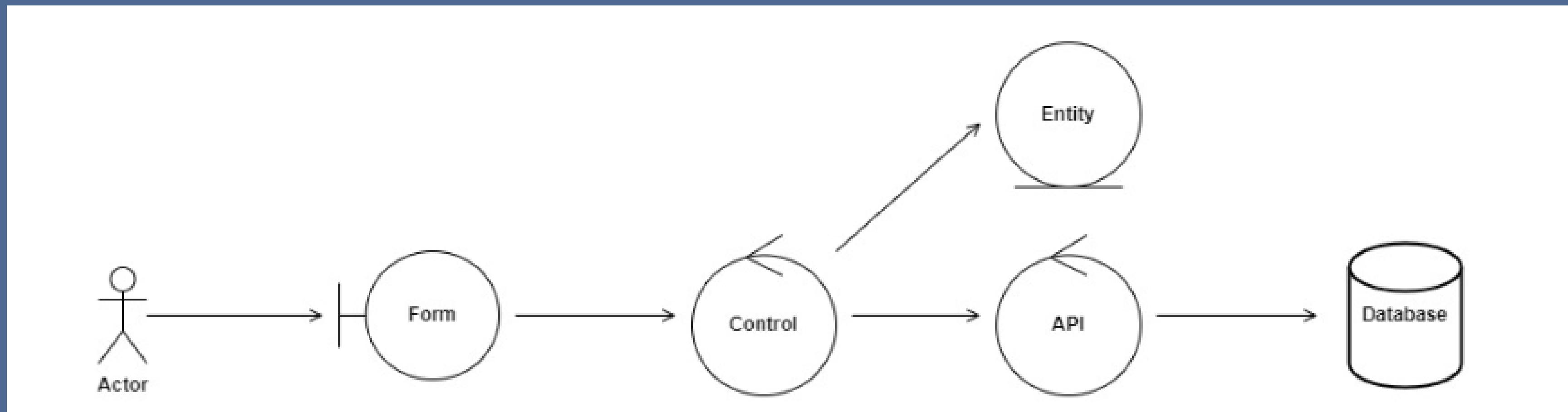
Phòng đào tạo



Tài khoản người dùng

3. Phân tích

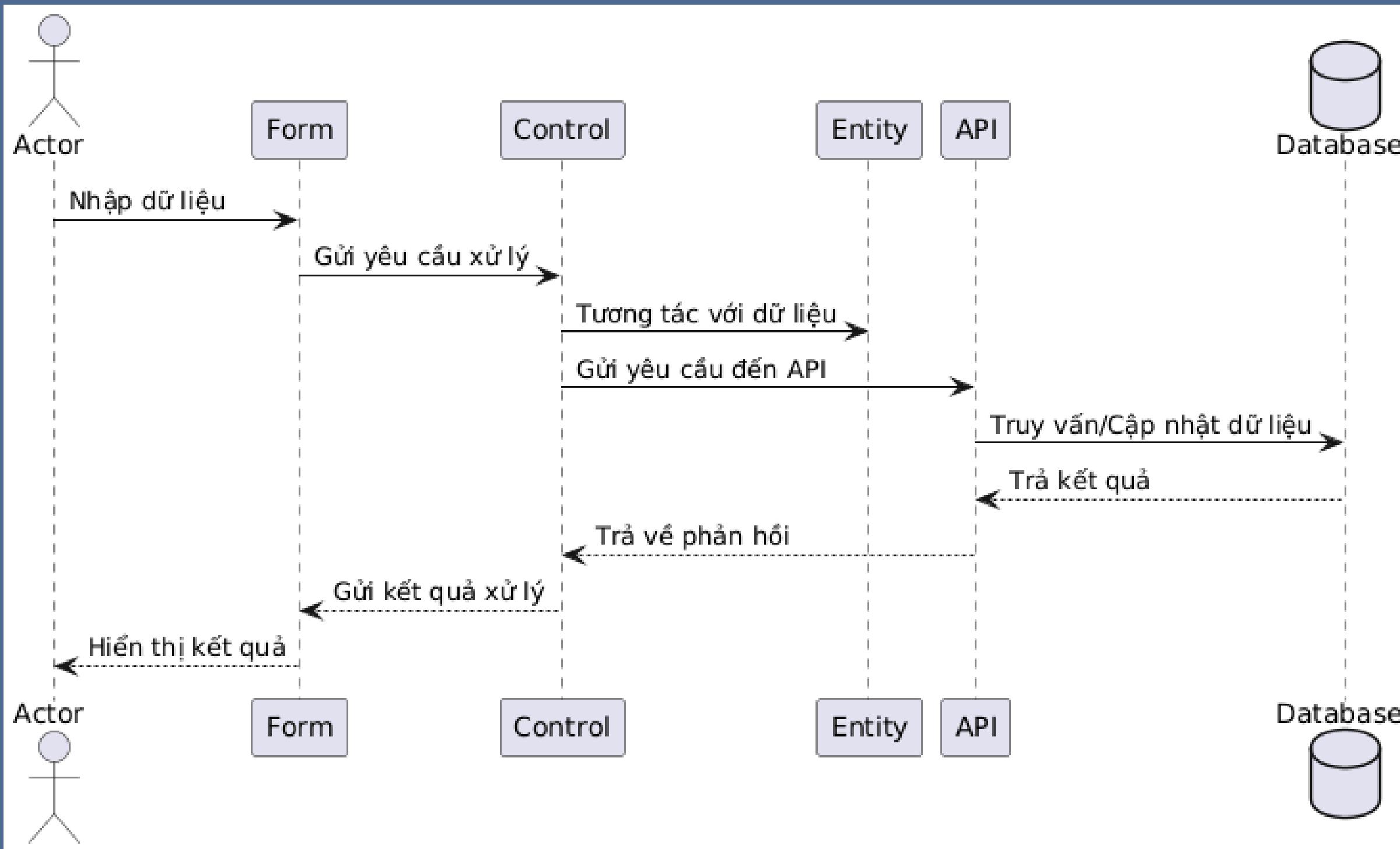
3.2 Thực thi trường hợp sử dụng



Sơ đồ lớp phân tích chung

3. Phân tích

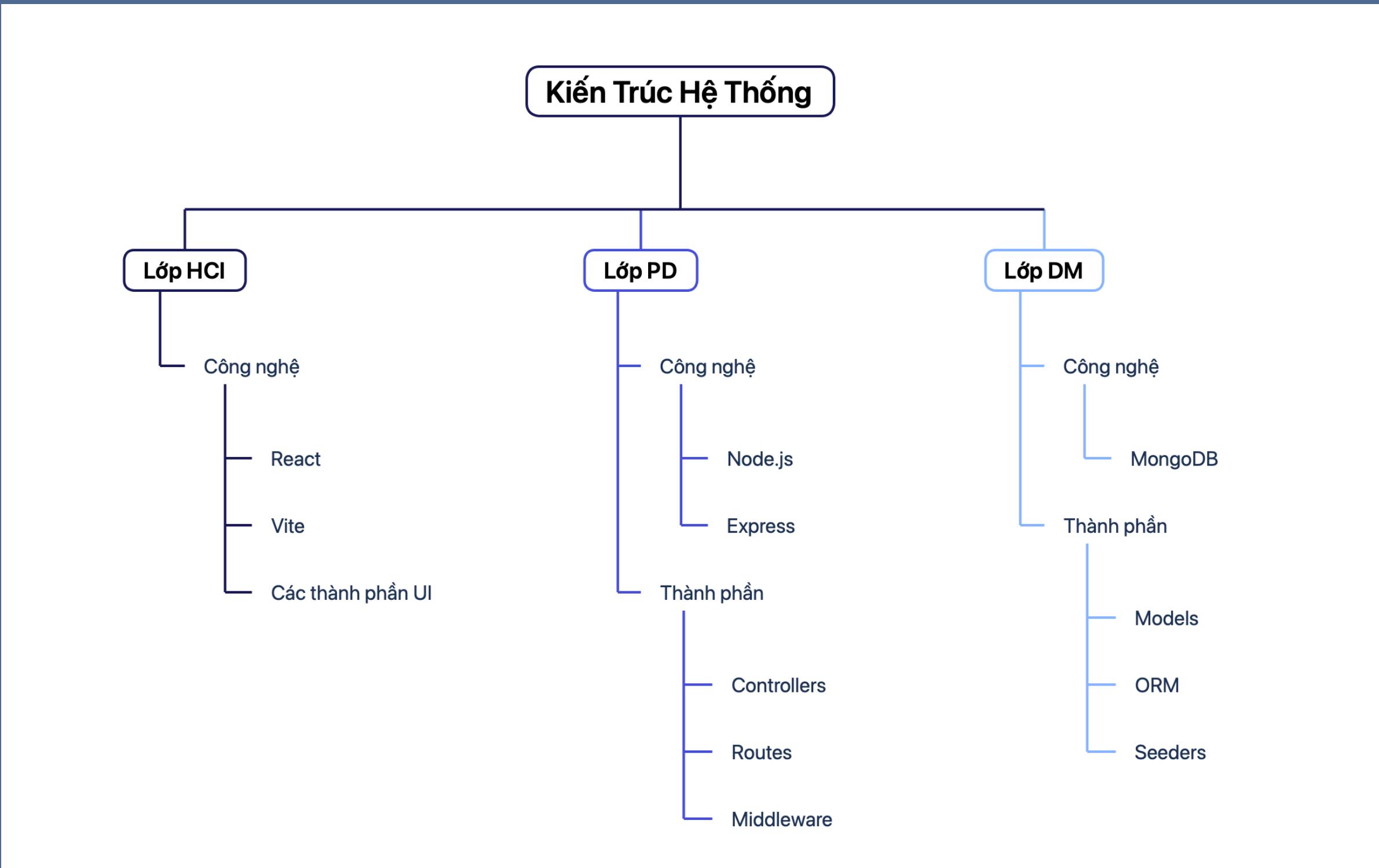
3.2 Thực thi trường hợp sử dụng



Sơ đồ tuân tự chung

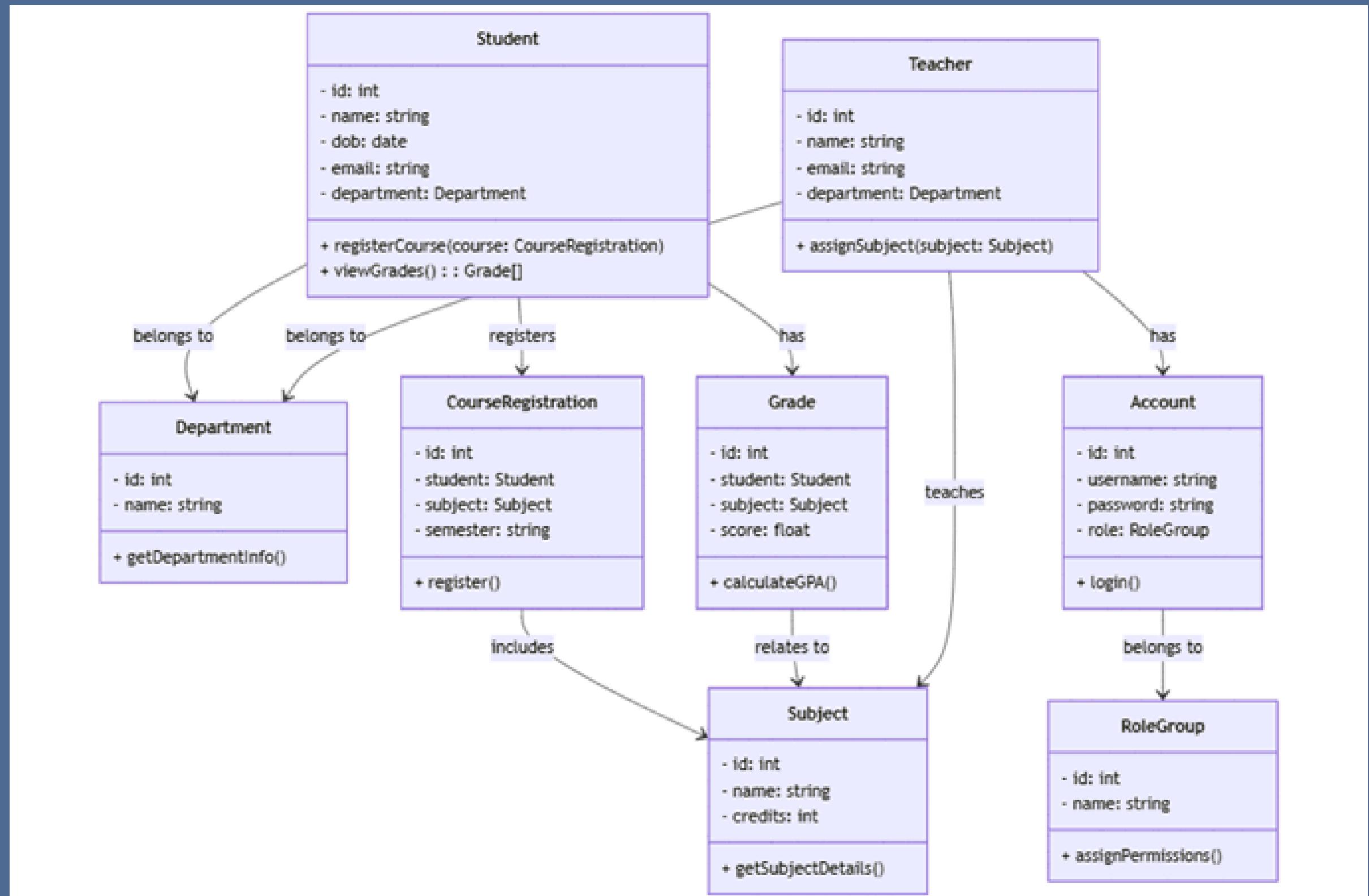
4. Thiết kế

SƠ ĐỒ
PACKAGE
TỔNG QUÁT
CỦA HỆ THỐNG



4. Thiết kế

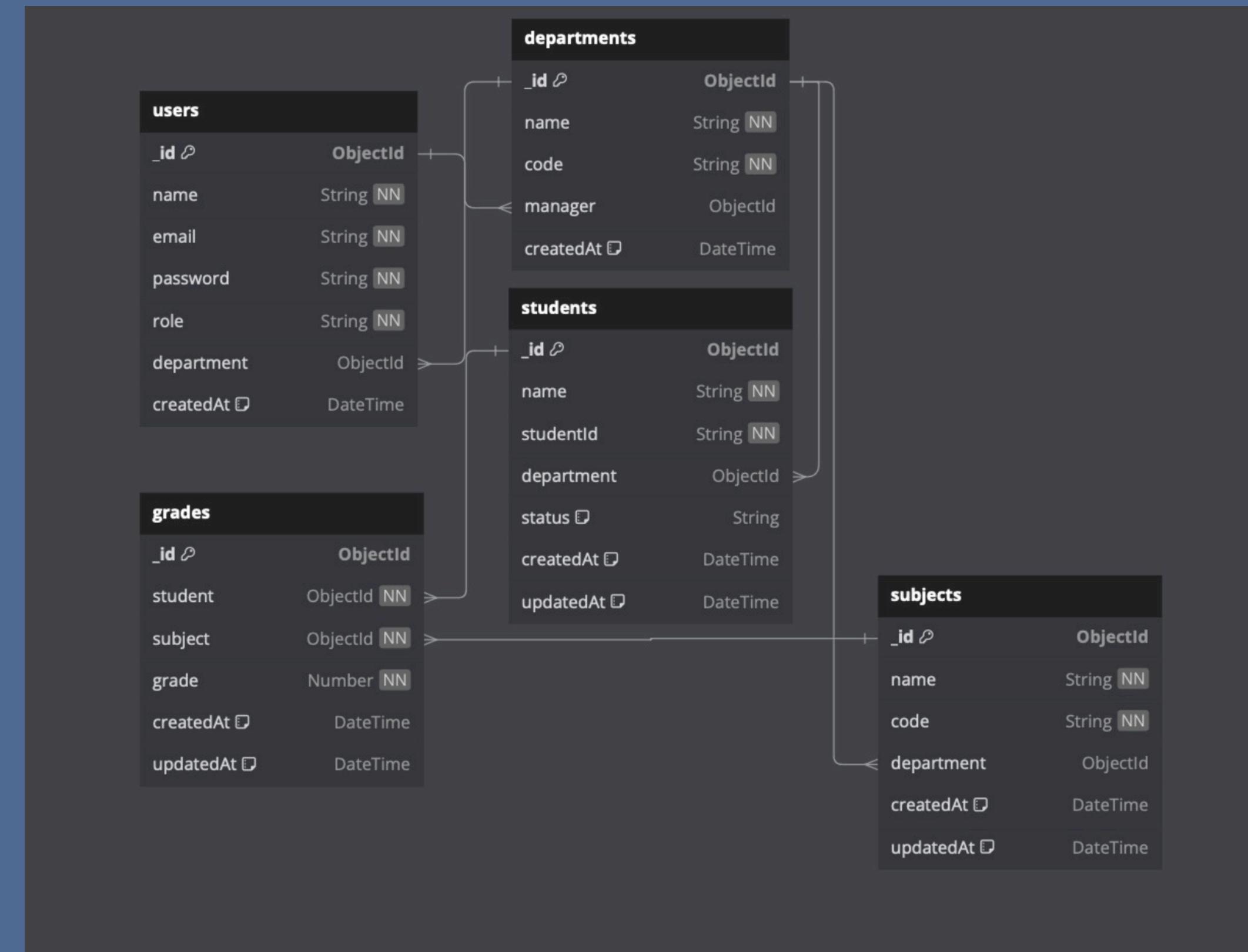
○ Thiết kế Biểu đồ Lớp



Class Diagram

4. Thiết kế

○ Thiết kế CSDL



Lược đồ Cơ Sở Dữ Liệu

4. Thiết kế

Thiết kế API

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

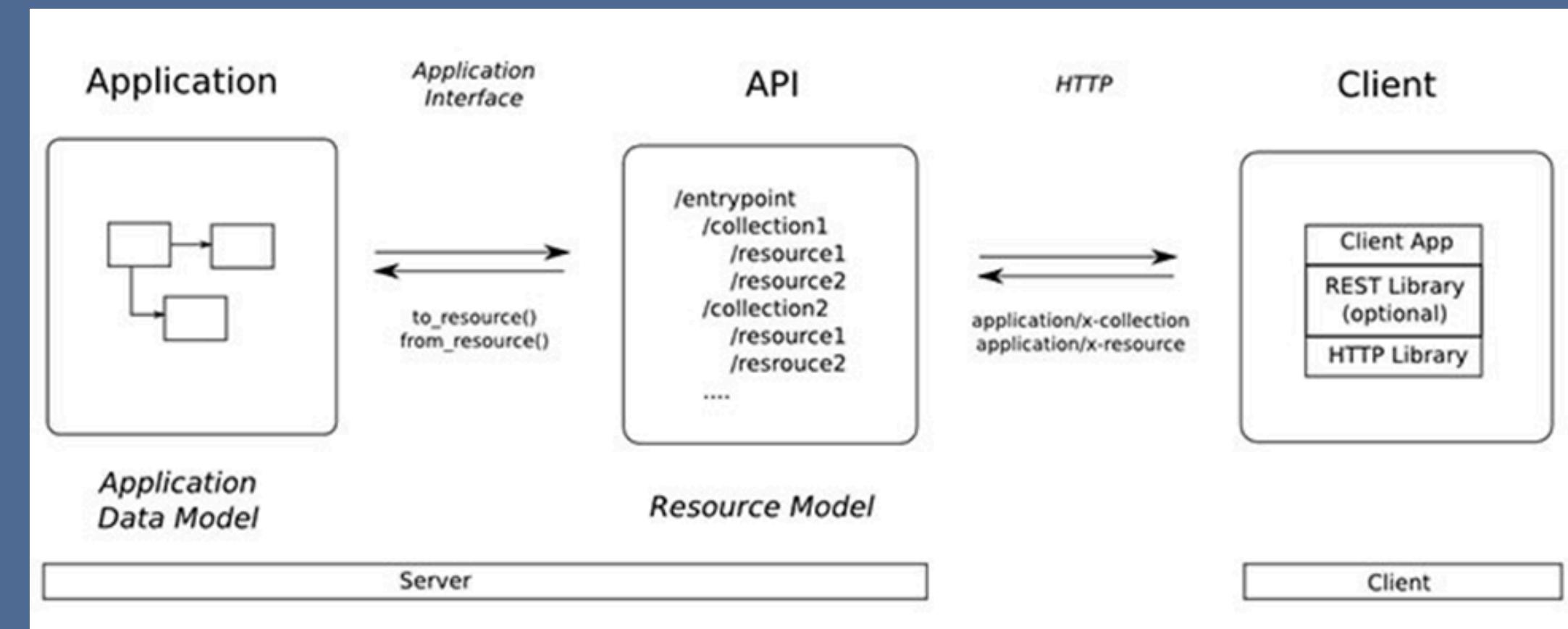
-GET (SELECT): Trả về một Resource hoặc một danh sách Resource.

-POST (CREATE): Tạo mới một Resource.

-PUT (UPDATE): Cập nhật thông tin cho Resource.

-DELETE (DELETE): Xoá một Resource.

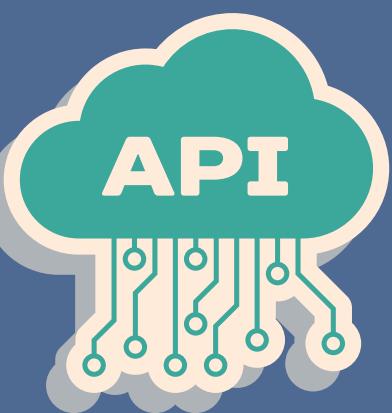
=> Những phương thức hay hoạt động này thường được gọi là CRUD



Cấu trúc thư mục của API

4. Thiết kế

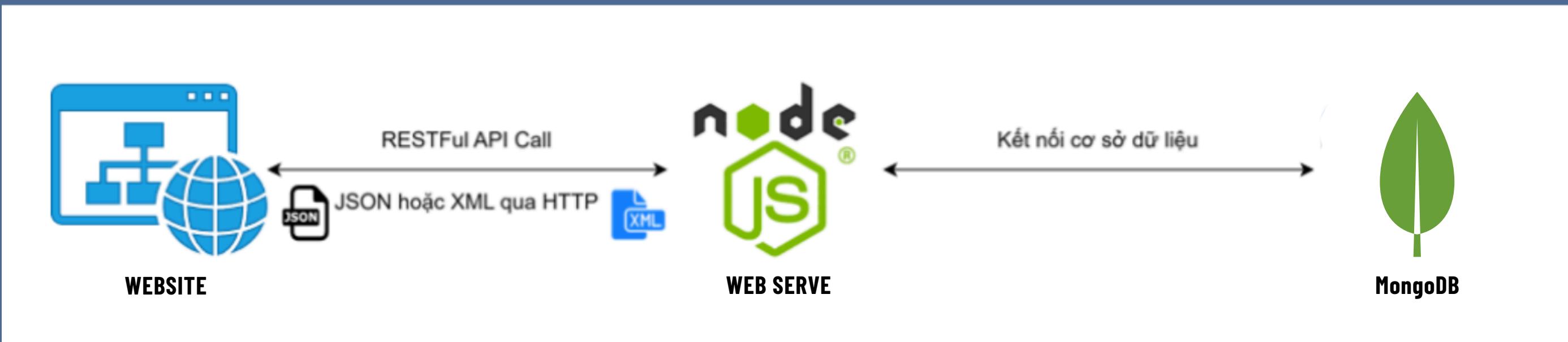
Danh sách các API



STT	Chức năng	URL	Phương thức
1	Xác thực người dùng	/api/auth	POST, GET
2	Quản lý người dùng	/api/users	GET, POST, PUT, DELETE
3	Quản lý khoa/bộ môn	/api/departments	GET, POST, PUT, DELETE
4	Quản lý sinh viên	/api/students	GET, POST, PUT, DELETE
5	Quản lý môn học	/api/subjects	GET, POST, PUT, DELETE
6	Quản lý điểm số	/api/grades	GET, POST, PUT, DELETE
7	Đăng ký khóa học	/api/courseRegistrations	GET, POST, DELETE

5. Cài đặt

Lựa chọn công nghệ



Sơ đồ triển khai hệ thống

Mô tả: Người dùng gửi yêu cầu qua ứng dụng Web, yêu cầu được gửi đến API trên Web Server để xử lý. Web Server kết nối với cơ sở dữ liệu, truy xuất dữ liệu và trả lại cho ứng dụng Web hiển thị kết quả cho người dùng.

5. Cài đặt

Giới thiệu các công nghệ sử dụng

- Công nghệ sử dụng tạo Server: NodeJs tạo các RESTFul API cho bên client gọi để lấy dữ liệu từ cơ sở dữ liệu.



Mô tả

- Được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau
- Node.js là một JavaScript runtime dựa trên V8 engine, sử dụng mô hình event-driven và non-blocking I/O, giúp xử lý hàng ngàn kết nối đồng thời mà vẫn nhẹ và hiệu quả
- Node.js có cộng đồng người dùng lớn, hỗ trợ mạnh mẽ, tốc độ xử lý cao, và khả năng mở rộng dễ dàng, phù hợp cho các ứng dụng web cần hiệu suất cao.

5. Cài đặt

Giới thiệu các công nghệ sử dụng

- Cơ sở dữ liệu sử dụng: MongoDB lưu trữ dữ liệu của phần mềm.



Mô tả

- MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL dạng document-oriented (hướng tài liệu), lưu trữ dữ liệu dưới dạng JSON/BSON linh hoạt thay vì bảng như SQL.
- MongoDB kết hợp với Node.js rất phù hợp cho các ứng dụng web, API RESTful, microservices và ứng dụng real-time.

5. Cài đặt

Giới thiệu các công nghệ sử dụng

- Công nghệ sử dụng xây dựng Web Client: ReactJs thiết kế giao diện bên phía người dùng trên ứng dụng Web.

Mô tả



- React là thư viện JavaScript giúp xây dựng giao diện người dùng hiệu quả và linh hoạt bằng cách tạo các components độc lập, tái sử dụng được.
- ReactJS sử dụng JSX giúp viết mã dễ dàng hơn, hỗ trợ tốt SEO, dễ kiểm thử, và có nhiều thư viện hỗ trợ, phù hợp cho các dự án mở rộng và có cộng đồng phát triển mạnh.

6. Demo

Link mã nguồn dự án:

<https://github.com/vintageboy/N06-PTTKPM-NHOM3>

7. Kết luận

1. Những điều đã làm được

- Hoàn thiện giao diện quản lý sinh viên với các chức năng cơ bản, hệ thống có thể lưu trữ và hiển thị thông tin sinh viên một cách chính xác, chức năng đăng ký môn học hoạt động ổn định.
- Hỗ trợ phân quyền tài khoản cơ bản cho sinh viên và Quản lý Khoa.

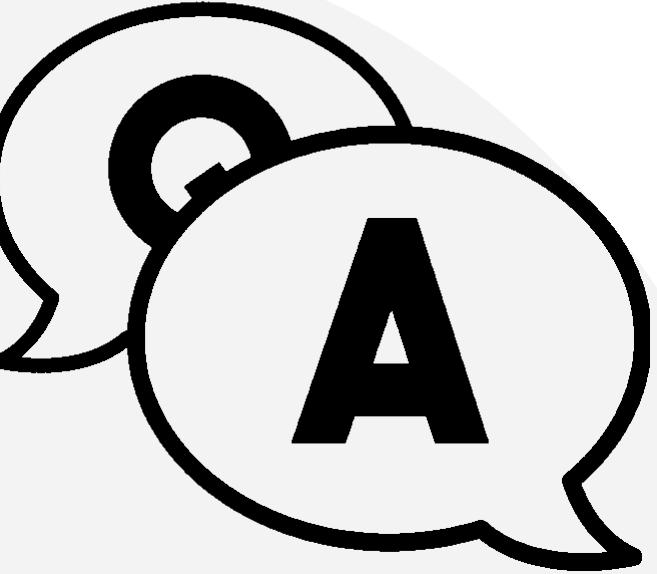
2. Những điều chưa làm được

- Hệ thống vẫn chưa có tính nghiệp vụ rõ ràng, chưa hỗ trợ sinh viên tra cứu lịch học và lịch thi,...
- Chưa tích hợp hệ thống xác thực mạnh mẽ để đảm bảo bảo mật cao hơn.
- Giao diện người dùng chưa tối ưu cho thiết bị di động.
- Chưa tích hợp hệ thống theo dõi tiến độ học tập theo từng kỳ.

3. Đề xuất tương lai

- Cải thiện bảo mật: Tích hợp xác thực hai yếu tố (2FA) và mã hóa dữ liệu.
- Mở rộng chức năng: Thêm tra cứu lịch học, lịch thi, thêm xuất báo cáo học tập
- Tối ưu hóa giao diện: Đảm bảo hệ thống hoạt động tốt trên cả desktop và thiết bị di động.

THANK YOU!



Hà Nội, 3-2025