

# Tóm tắt nội dung Project

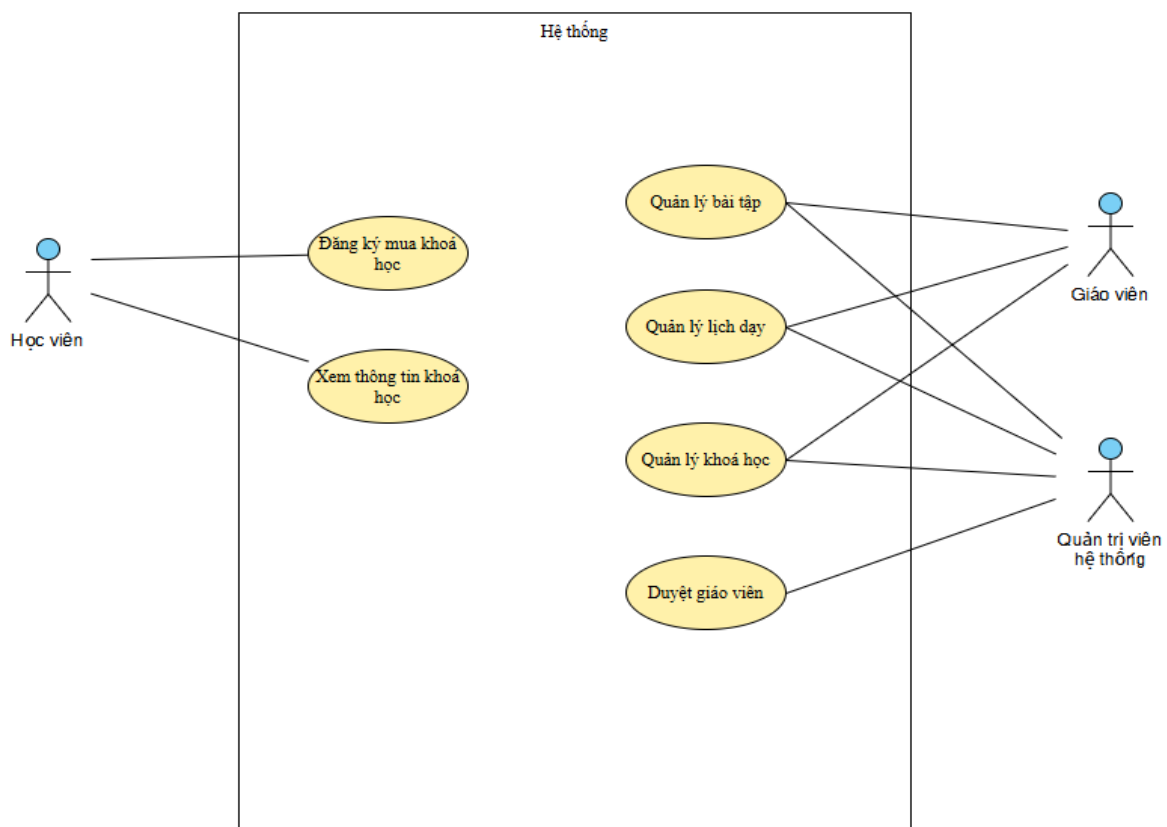
## CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1. Đặt vấn đề
2. Mục tiêu và phạm vi đề tài

## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH BÀI TOÁN

1. Tổng quan
2. Khảo sát hiện trạng tìm khóa học online
3. Tổng quan chức năng
  - 3.1. Phân tích các tác nhân trong hệ thống

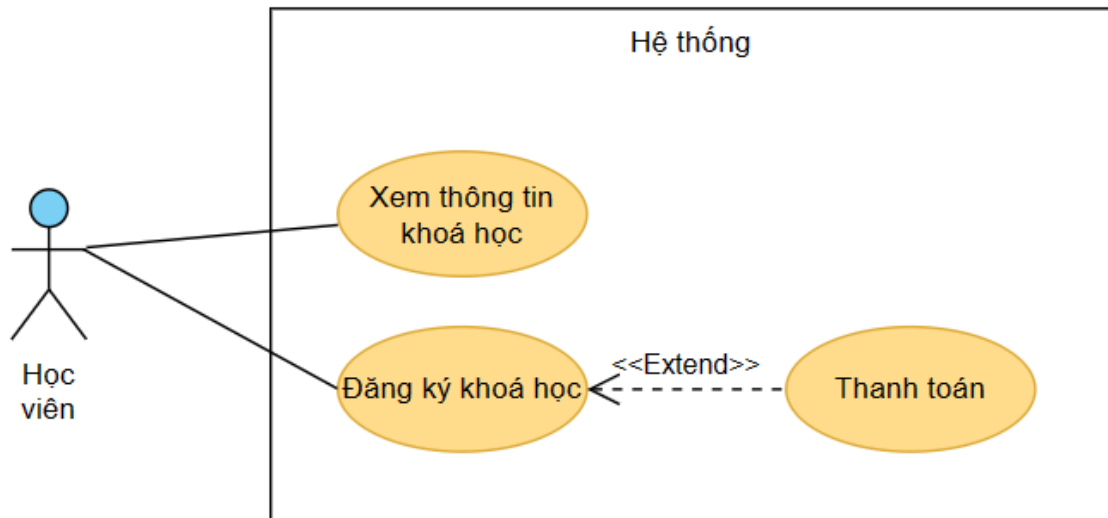
Hệ thống chia đối tượng sử dụng thành ba người dùng chính. Người dùng I là học viên, được chia nhỏ thành người dùng có tài khoản để học tập và khách vãng lai có mong muốn tìm khóa học phù hợp. Người dùng II là giáo viên, là người có thể đăng ký tài khoản dạy học trên web và bán khóa học cũng như quản lý các khóa học của mình. Cuối cùng là người quản lý hệ thống (admin) có các chức năng liên quan đến việc quản lý khách hàng sử dụng hệ thống, phân tích dữ liệu các khóa học, tình trạng phát triển của hệ thống và phân quyền. Biểu đồ use case tổng quát



Hình 1 Usecase tổng quát

### 3.2. Biểu đồ use case phân rã

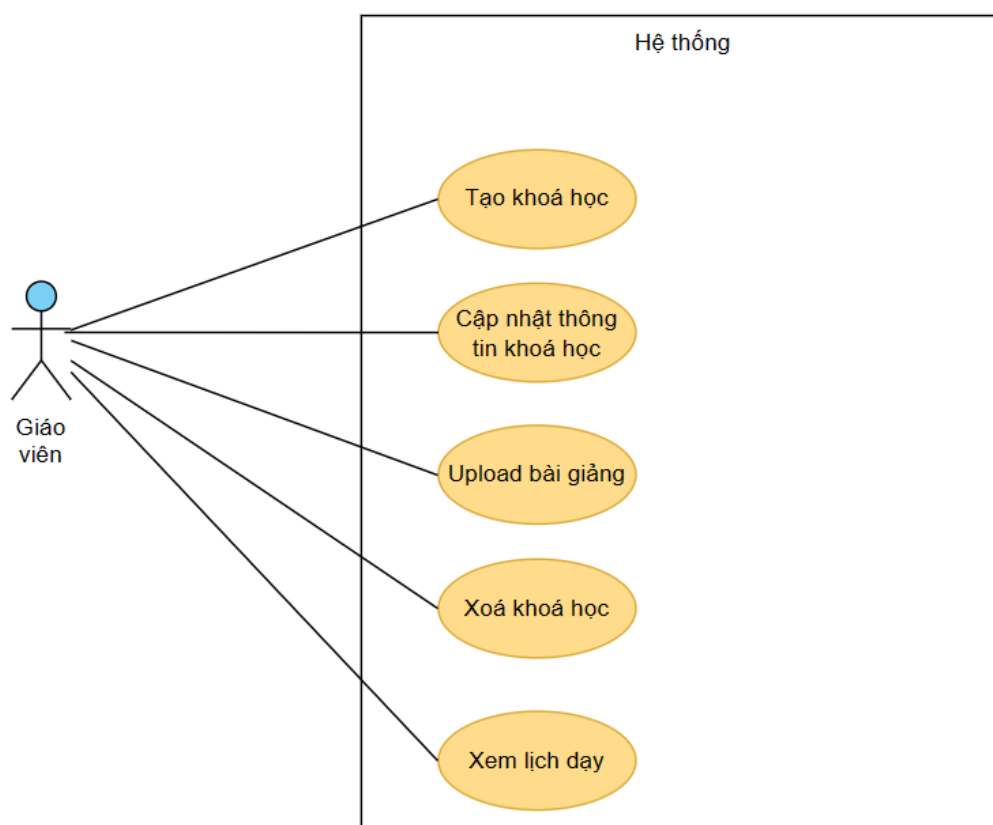
#### 3.2.1. Biểu đồ usecase đăng ký khoá học ( Học viên )



Hình 2 Usecase đăng ký khoá học

Hình 2 mô tả use case phân rã đăng ký khoá học. Trong đó học viên có thể xem thông tin chi tiết của khóa học. Học viên cũng có thể đăng ký khóa học và thanh toán qua VNpay.

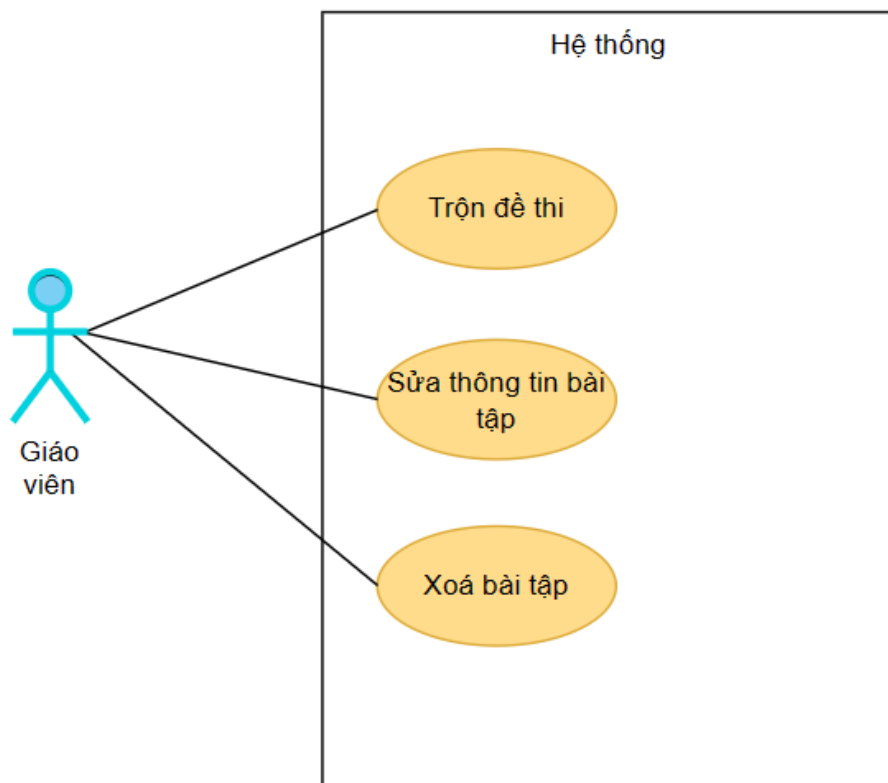
### 3.2.2. Biểu đồ use case quản lý khoá học ( Giáo viên )



Hình 3 Usecase quản lý khoá học

Hình 3 mô tả use case phân rã quản lý khoá học. Giáo viên dùng các chức năng như tạo khóa học mới, cập nhật thông tin khóa học, upload bài giảng như file tài liệu, video, xoá khóa học và xem lịch dạy.

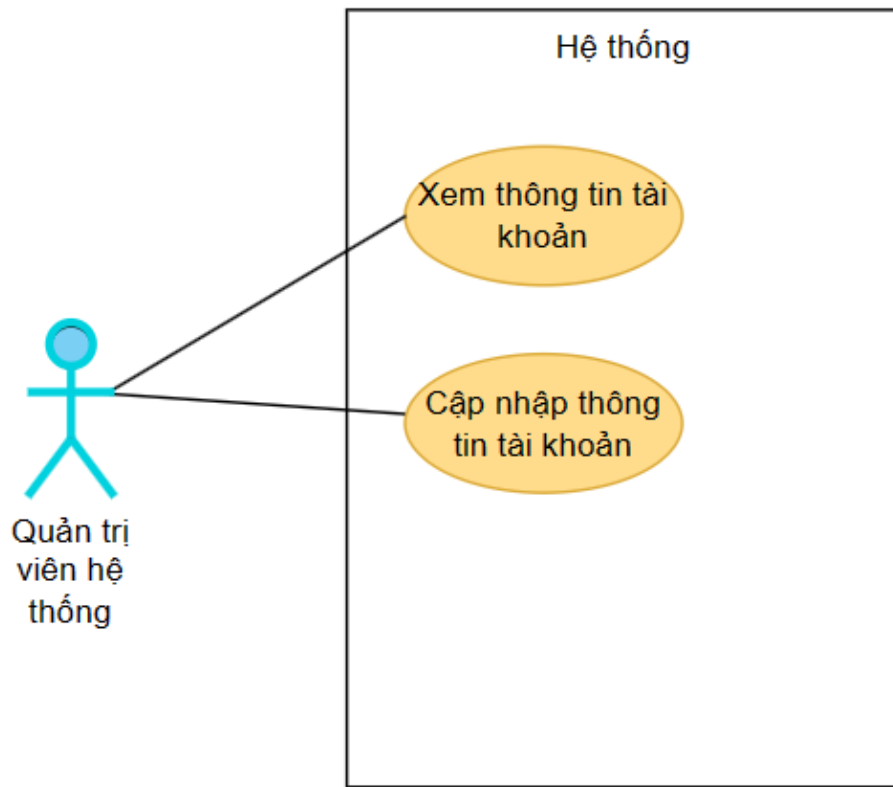
### 3.2.3. Biểu đồ use case quản lý bài tập ( Giáo viên )



*Hình 4 Usecase quản lý bài tập*

Hình 4 mô tả use case phân rõ quản lý bài tập. Giáo viên có thể dùng các chức năng như tạo bài tập mới từ ngân hàng câu hỏi upload lên, hay thông tin chi tiết bài tập và xoá bài tập.

#### 3.2.4. Biểu đồ use case quản lý tài khoản ( Quản trị viên hệ thống )



Hình 5 Usecase quản lý tài khoản

Hình 5 mô tả use case phân rã quản lý tài khoản người dùng. Quản trị viên có thể xem thông tin chi tiết của các tài khoản và cập nhật thông tin tài khoản cũng như phân quyền cho tài khoản đó.

### 3.3. Đặc tả chức năng

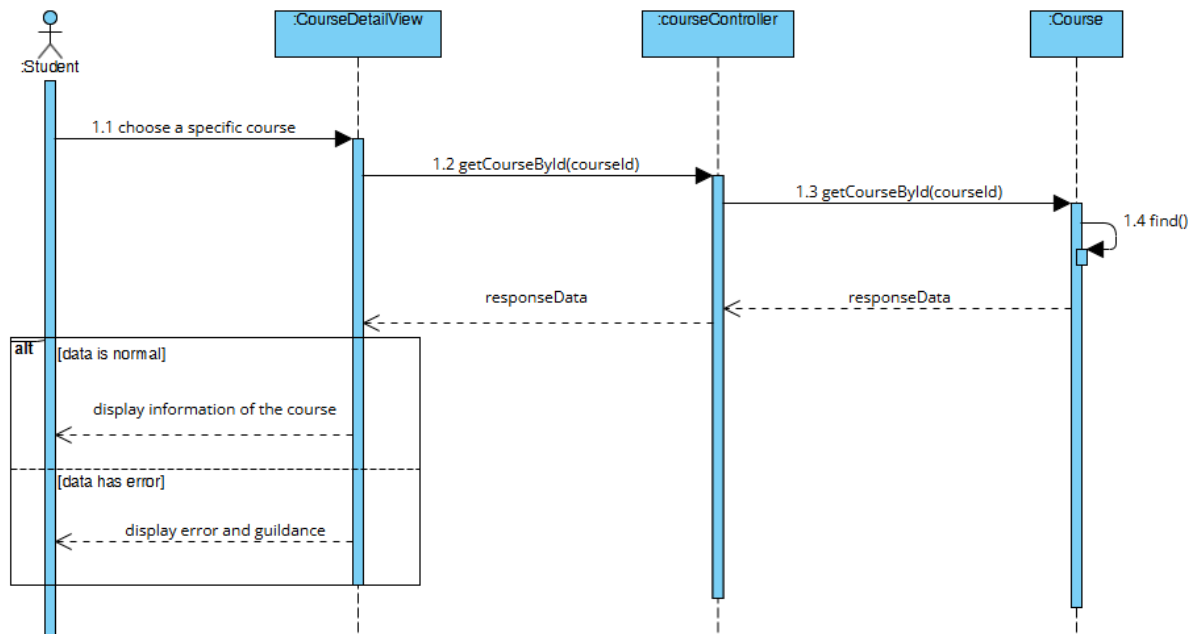
#### 3.3.1. Đặc tả chức năng xem thông tin khoá học

<b>Mã Use case</b>	UC01
<b>Tên Use case</b>	Xem thông tin khóa học
<b>Mô tả</b>	Người học có thể xem thông tin chi tiết về một khóa học cụ thể trên web.

<b>Tác nhân thực hiện</b>	Học viên
<b>Tên luồng</b>	Mô tả
<b>Luồng sự kiện chính</b>	<p>1. Học viên xem mô tả khái quát về khóa học, sau đó click vào thẻ khóa học muốn được xem chi tiết.</p> <p>2. Hệ thống sẽ hiển thị giao diện với các thông tin như nhiều hình ảnh mô tả về khóa học, thông tin chi tiết hơn về khóa học (giáo viên, giá khóa học, lịch học, mô tả khóa học, chứng chỉ của khóa học, và các yêu cầu tiên quyết để tham gia khóa học (nếu có)).</p> <ul style="list-style-type: none"> <li>• Người học có thể xem mô tả các chương, bài giảng, bài kiểm tra, và tài liệu liên quan đến khóa học.</li> <li>• Nếu khóa học được chia thành các bài giảng, người học có thể chọn xem bài giảng cụ thể. Người học có thể quay lại trang chính hoặc tiếp tục tương tác với các chức năng khác của hệ thống.</li> </ul>
<b>Luồng sự kiện thay thế.</b>	<p>2a. Học viên quyết định đăng ký khóa học ngay từ trang chi tiết, họ có thể thực hiện quy trình đăng ký trực tiếp từ trang này. Nếu khóa học có tính năng lớp học trực tuyến, người học có thể chọn tham gia ngay từ trang chi tiết.</p> <p>2b. Người học có thể chọn viết đánh giá về khóa học từ trang chi tiết nếu họ đã tham gia và hoàn thành nó.</p>

<b>Luồng sự kiện ngoại lệ</b>	2b. Nếu khóa học bị lỗi (lỗi do nhà cung cấp, do mạng) thì hệ thống hiển thị trang 404 Not Found và yêu cầu thử tìm lại khóa học khác.
-------------------------------	--

● Sequence diagram



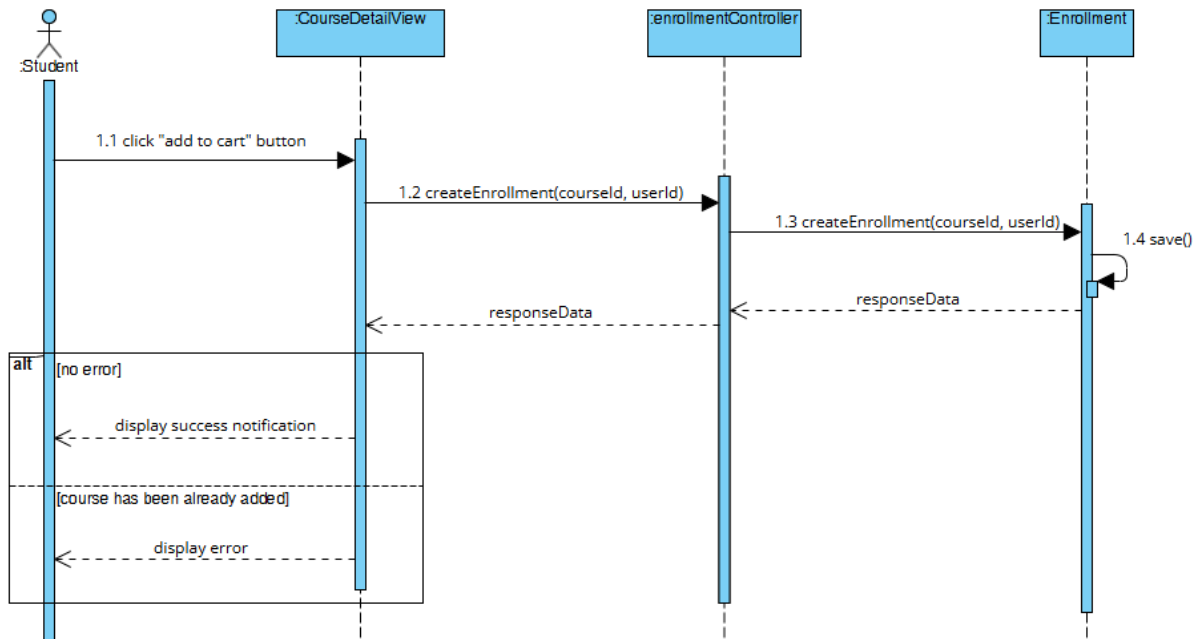
3.3.2. Đặc tả chức năng đăng ký khoá học

<b>Mã Use case</b>	UC02
<b>Tên Use case</b>	Đăng ký khóa học
<b>Mô tả</b>	Người học có thể tham gia khóa học mà họ mong muốn
<b>Tác nhân thực hiện</b>	Học viên

<b>Tiền điều kiện</b>	Học viên đã đăng nhập vào hệ thống, khóa học này chưa được học viên đăng ký từ trước, khóa học còn chỗ trống (nếu là khóa học offline).
<b>Tên luồng</b>	<b>Mô tả</b>
<b>Luồng sự kiện chính</b>	<p>1. Người học chọn tùy chọn "Tham gia" trên trang chi tiết khóa học.</p> <p>2. Hệ thống hiển thị một cửa sổ xác nhận với thông tin chi tiết về khóa học và yêu cầu xác nhận từ người học. Sau khi xác nhận, hệ thống hiển thị thông báo xác nhận đăng ký thành công và người học có thể xem các bài giảng khóa học và sử dụng tài liệu, bài thi được cung cấp trong khóa học (online) hoặc tham dự lớp học offline. Người học đã đăng ký thành công có thể quay lại trang chính hoặc tiếp tục tham gia các khóa học khác.</p>
<b>Luồng sự kiện thay thế.</b>	2a. Nếu khóa học có phí, sau khi xác nhận đăng ký, hệ thống chuyển người dùng tới bước thanh toán. Sau khi thực hiện xong bước thanh toán, người dùng sẽ tiếp tục những bước tiếp theo.
<b>Luồng sự kiện ngoại lệ</b>	2a. Nếu có lỗi xảy ra trong quá trình đăng ký (ví dụ: lỗi hệ thống, không đủ chỗ trống), hệ thống hiển thị thông báo lỗi và hướng dẫn người học thử lại hoặc liên hệ hỗ trợ.



- Sequence diagram

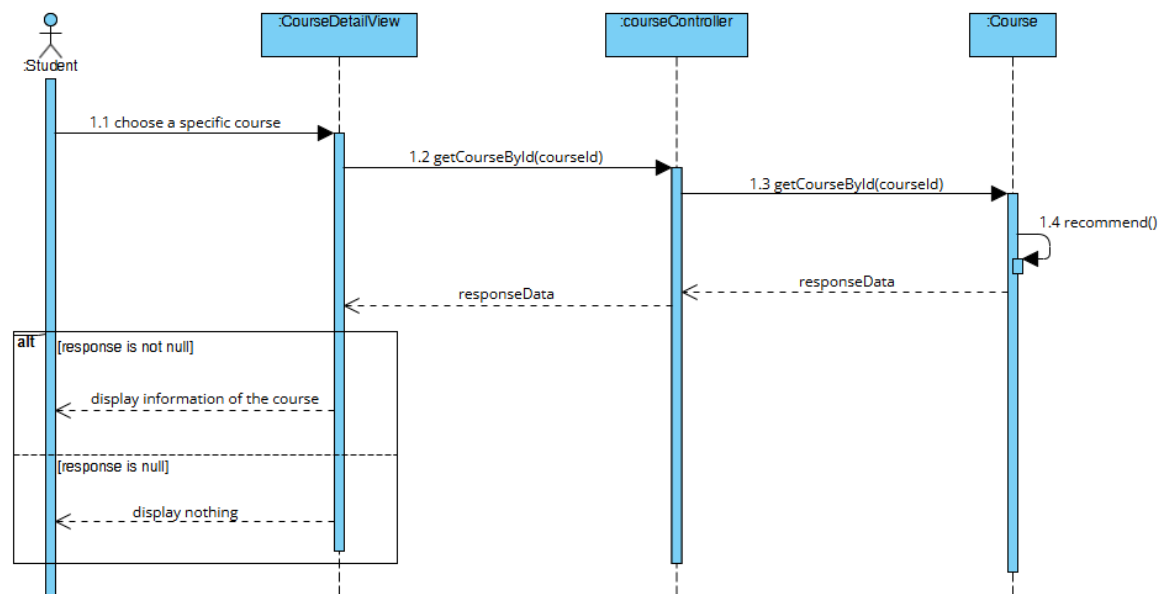


### 3.3.3. Đặc tả chức năng gợi ý khóa học

<b>Mã Use case</b>	UC03
<b>Tên Use case</b>	Gợi ý khóa học
<b>Mô tả</b>	Học viên được gợi ý các khóa học tương đồng với khóa học đang xem khi mua một khóa học bất kì.
<b>Tác nhân thực hiện</b>	Học viên
<b>Tiền điều kiện</b>	Học viên đăng nhập vào trang web
<b>Hậu điều kiện</b>	

Tên luồng	Mô tả
<b>Luồng sự kiện chính</b>	<p>B1: Sau khi người dùng mua 1 khóa học bất kì thành công, hệ thống sẽ đưa ra gợi ý các khóa học tương đồng, hoặc đưa ra một lộ trình học tập cụ thể dựa vào khóa học đã mua.</p> <p>B2: Người dùng có thể chọn xem từng khóa học mà hệ thống đã gợi ý.</p>
<b>Luồng sự kiện thay thế.</b>	1a. Khi người dùng nhấn vào xem một khóa học bất kì, hệ thống cũng sẽ đưa ra gợi ý các khóa học tương đồng ở bên dưới khóa học đó.
<b>Luồng sự kiện ngoại lệ</b>	1a. Không có khóa học nào liên quan đến khóa học người dùng đang xem thì hệ thống không gợi ý.

#### ● Sequence diagram



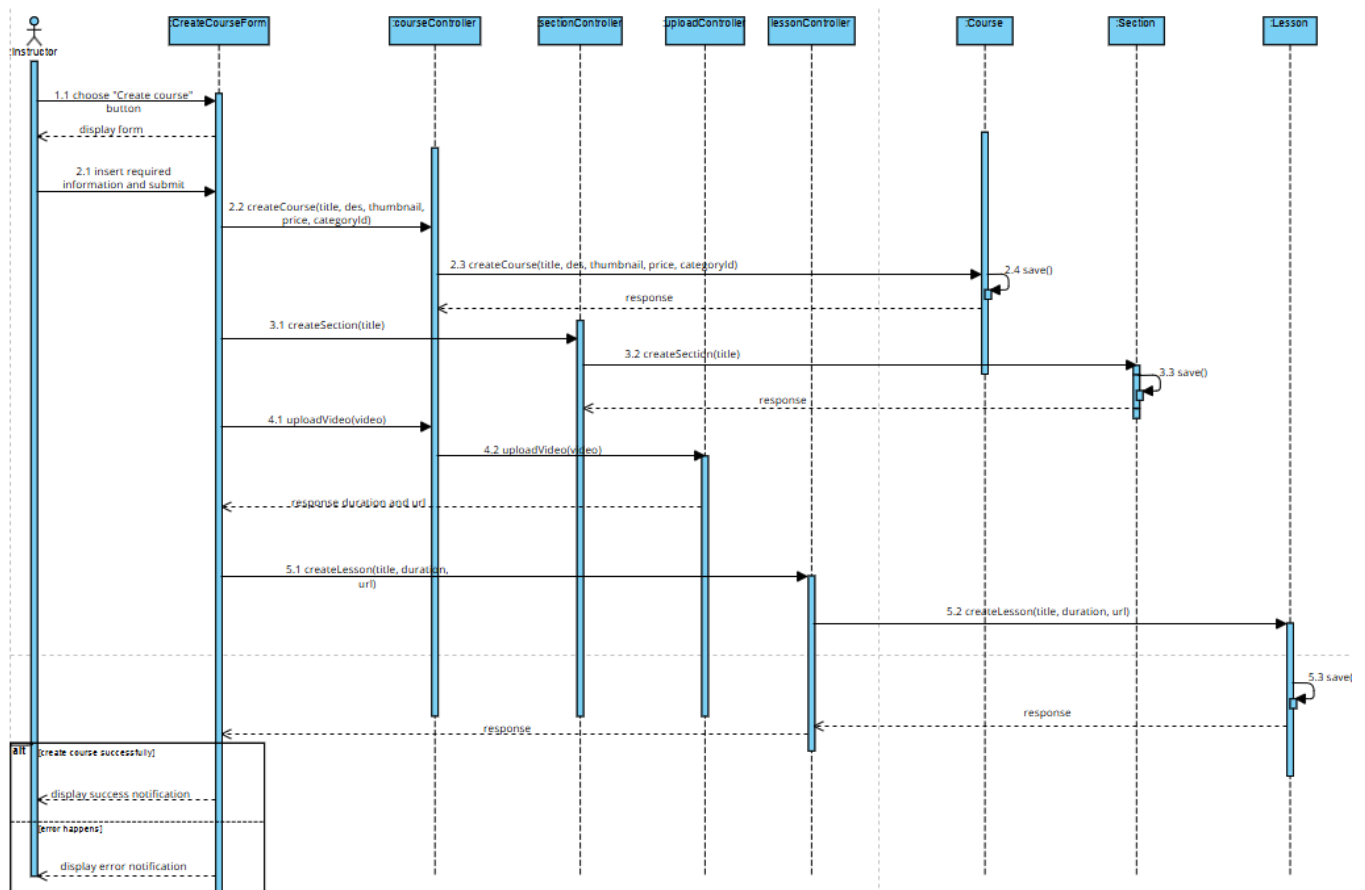
#### 3.3.4. Đặc tả chức năng tạo khóa học

<b>Mã Use case</b>	UC4
--------------------	-----

<b>Tên Use case</b>	Tạo khóa học online		
<b>Mô tả</b>	Thực hiện quá trình tạo một khóa học mới trên nền tảng giáo dục trực tuyến. giáo viên có thể thêm nội dung, quản lý lịch trình, và đánh giá học viên thông qua chức năng này.		
<b>Tác nhân thực hiện</b>	Giáo viên		
<b>Tiền điều kiện</b>	Người giáo viên đã đăng nhập vào tài khoản của mình trên nền tảng giáo dục trực tuyến.		
<b>Hậu điều kiện</b>	Khóa học phải có ít nhất 30 phút nội dung toàn khóa học, bao gồm ít nhất 5 bài giảng riêng biệt và video phải có chất lượng HD.		
<table><tr><td><b>Tên luồng</b></td><td>Mô tả</td></tr></table>		<b>Tên luồng</b>	Mô tả
<b>Tên luồng</b>	Mô tả		

<p><b>Luồng sự kiện chính</b></p>	<p>1. Người giáo viên chọn tùy chọn "Tạo Khóa Học" từ trang quản lý giáo viên.</p> <p>2. Hệ thống hiện ra form điền các thông tin cơ bản về khóa học bao gồm tên khóa học, thể loại khóa học, mô tả ngắn, và các chương của khóa học, cũng như coupon giảm giá khóa học. Người giáo viên thêm nội dung cho khóa học, bao gồm các bài giảng, bài đọc, video, và bài tập.</p> <p>Trong đó trang tạo nội dung khóa học bao gồm các phần với yêu cầu cụ thể như sau:</p> <ul style="list-style-type: none"> <li>- Tên tiêu đề không có kí tự đặc biệt, dưới 30 kí tự</li> <li>- Thể loại khóa học phải chọn trong list danh sách thể loại khóa học</li> <li>- Mục tiêu khóa học, yêu cầu, mô tả khóa học, đối tượng khóa học ở dạng text</li> <li>- Chi phí của khóa học, tạo coupon cho khóa học</li> <li>- Các chương của khóa học, người giáo viên tạo từng chương của khóa học, trong mỗi chương có thể thêm video hoặc tài liệu khóa học vào, yêu cầu tối thiểu lần tạo khóa học phải đăng 5 video.</li> </ul> <p>Sau khi đáp ứng được yêu cầu, giáo viên nhấn nút tạo khóa học, yêu cầu sẽ được gửi tới đến quản trị viên.</p> <p>3. Nếu quản trị viên chấp nhận khóa học, giáo viên sẽ nhận được thông tin đã nhận được khóa học thành công.</p>
<p><b>Luồng sự kiện thay thế.</b></p>	<p>2a. Người giáo viên có thể hủy bỏ quá trình tạo khóa học bất cứ lúc nào nếu họ không muốn tiếp tục.</p>
<p><b>Luồng sự kiện ngoại lệ</b></p>	<p>2a. Nếu có lỗi xuất hiện trong quá trình tạo khóa học, người giáo viên sẽ nhận được thông báo lỗi và hướng dẫn cách sửa.</p>

- Sequence diagram

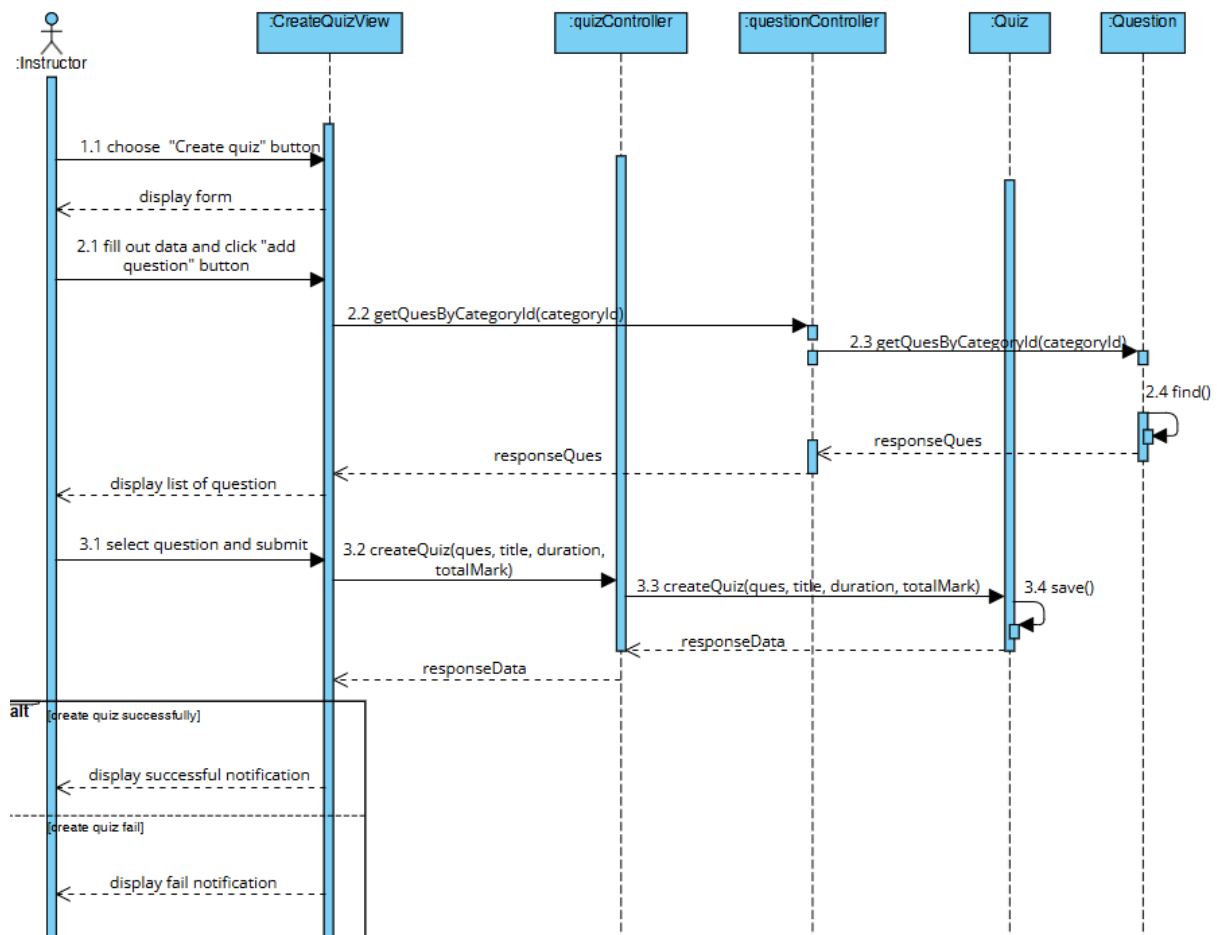


### 3.3.5. Đặc tả chức năng tạo bài tập thủ công

<b>Mã Use case</b>	UC5
<b>Tên Use case</b>	Tạo bài tập
<b>Mô tả</b>	Giáo viên có thể tạo bài tập theo chương trong khóa học.
<b>Tác nhân thực hiện</b>	Giảng viên
<b>Tiền điều kiện</b>	Người giáo viên đã đăng nhập vào tài khoản của mình trên nền tảng giáo dục trực tuyến.
<b>Hậu điều kiện</b>	

Tên luồng	Mô tả
<b>Luồng sự kiện chính</b>	<p>1. Giáo viên chọn tùy chọn “Tạo bài tập” trên trang quản lý bài tập của mình.</p> <p>2. Hệ thống hiển thị form điền thông tin.</p> <p>3. Giáo viên điền thông tin yêu cầu và chọn nút “Tạo câu hỏi mới”,</p> <p>4. Hệ thống sẽ hiển thị một giao diện danh sách các câu hỏi khi người dùng chọn mục tương ứng. Người dùng sau đó thêm câu hỏi mong muốn và nhấn nút “Lưu” để có thể tạo được bài thi như mong muốn.</p> <p>4. Hệ thống kiểm tra các câu hỏi xem hợp lệ chưa và rồi tiến hành tạo bài thi trên khóa học.</p>
<b>Luồng sự kiện thay thế.</b>	
<b>Luồng sự kiện ngoại lệ</b>	3a. Nếu có lỗi xuất hiện trong quá trình tạo bài thi, người giáo viên sẽ nhận được thông báo lỗi và hướng dẫn cách sửa.

● Sequence diagram



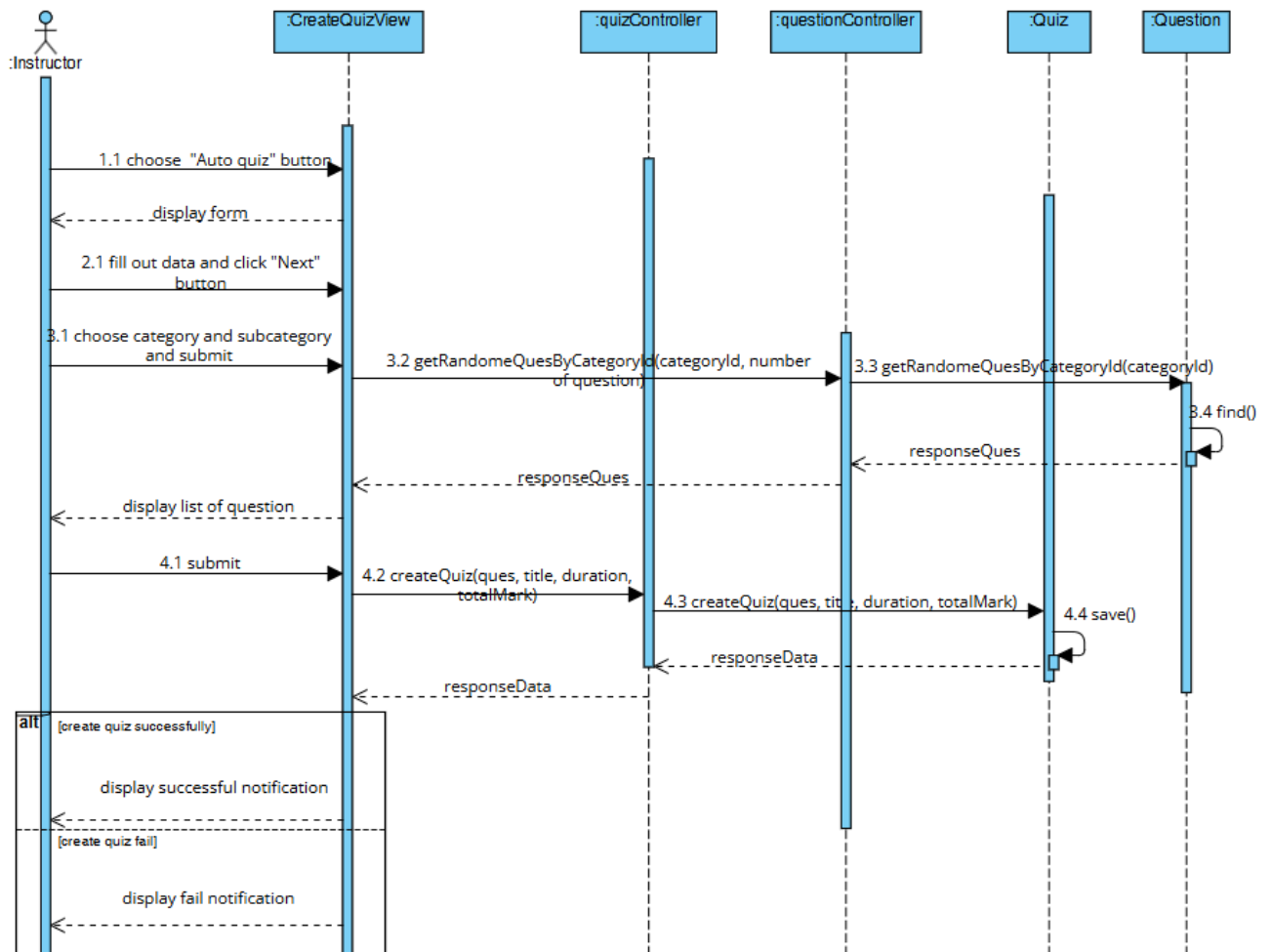
### 3.3.6. Đặc tả chức năng tạo bài tập tự động

<b>Mã Use case</b>	UC6
<b>Tên Use case</b>	Tạo bài tập tự động
<b>Mô tả</b>	Giáo viên có thể tạo bài tập theo chương trong khóa học tự động.
<b>Tác nhân thực hiện</b>	Giảng viên
<b>Tiền điều kiện</b>	Người giáo viên đã đăng nhập vào tài khoản của mình trên nền tảng giáo dục trực tuyến.
<b>Hậu điều kiện</b>	

Tên luồng	Mô tả
<b>Luồng sự kiện chính</b>	<p>1. Giáo viên chọn tùy chọn “Tạo bài tập tự động” trên trang quản lý bài tập của mình.</p> <p>2. Hệ thống hiển thị form điền thông tin.</p> <p>3. Giáo viên điền thông tin yêu cầu và chọn nút “Tự động tạo câu hỏi mới”,</p> <p>4. Hệ thống sẽ hiển thị danh sách câu hỏi theo khoá học tương ứng. Người dùng sau đó nhấn nút “Lưu” để có thể tạo được bài thi như mong muốn.</p> <p>4. Hệ thống kiểm tra các câu hỏi xem hợp lệ chưa và rồi tiến hành tạo bài thi trên khóa học.</p>
<b>Luồng sự kiện thay thế.</b>	
<b>Luồng sự kiện ngoại lệ</b>	<p>3a. Nếu có lỗi xuất hiện trong quá trình tạo bài thi, người giáo viên sẽ nhận được thông báo lỗi và hướng dẫn cách sửa.</p>

- Sequence diagram



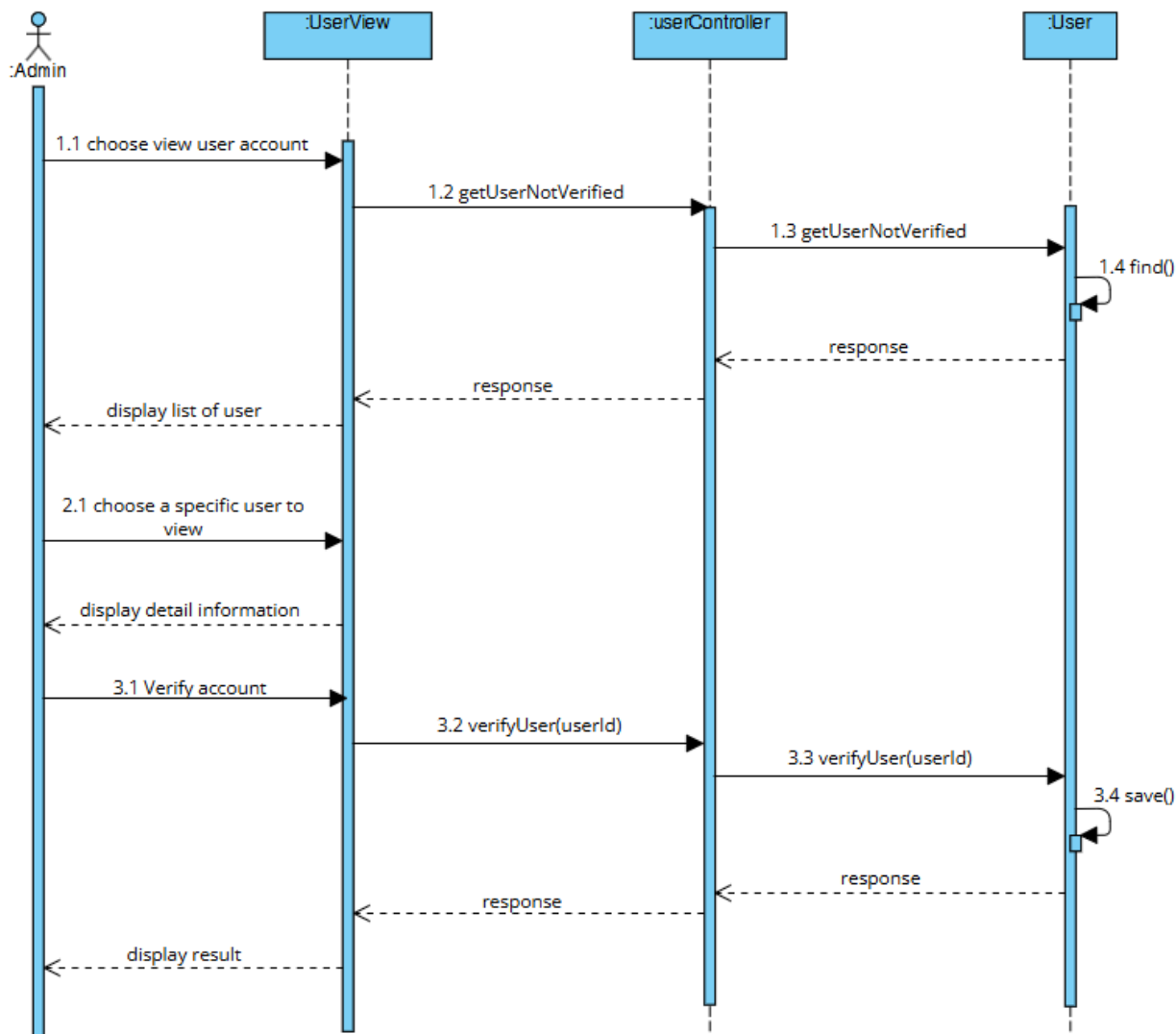


### 3.3.7. Đặc tả chức năng duyệt giáo viên

<b>Mã Use case</b>	UC7
<b>Tên Use case</b>	Duyệt giáo viên
<b>Mô tả</b>	Quản trị viên có thể xem cũng như duyệt tài khoản học sinh bình thường thành giáo viên
<b>Tác nhân thực hiện</b>	Quản trị viên hệ thống
<b>Tiền điều kiện</b>	Quản trị viên đã đăng nhập vào tài khoản của mình trên nền tảng giáo dục trực tuyến.

<b>Hậu điều kiện</b>	
<b>Tên luồng</b>	<b>Mô tả</b>
<b>Luồng sự kiện chính</b>	<p>1. Quản trị viên chọn tùy chọn "Quản Lý Tài Khoản" từ trang quản lý của mình trên web.</p> <p>2. Hệ thống hiển thị danh sách tài khoản đã đăng ký và chưa được duyệt.</p> <p>3. Quản trị viên có thể chọn một tài khoản cụ thể để xem chi tiết về tài khoản đó như số điện thoại, họ và tên và ảnh minh chứng. Ngoài ra quản trị viên có thể xét duyệt và phân quyền cho tài khoản đó thành giáo viên.</p>
<b>Luồng sự kiện thay thế.</b>	
<b>Luồng sự kiện ngoại lệ</b>	3a. Nếu có lỗi xuất hiện trong quá trình xem chi tiết tài khoản người dùng, quản trị viên sẽ nhận được thông báo lỗi và hướng dẫn cách sửa.

- Sequence diagram



## CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

### 1. Tổng quan

Tổng quan Trong chương trước, đã đề cập tới các nội dung về phân tích và khảo sát bài toán được giải quyết, kèm theo đó là những giải pháp thiết kế để giải quyết bài toán cùng với phân tích nghiệp vụ. Vì vậy chương này sẽ giới thiệu những công nghệ, phần mềm được sử dụng để phát triển và giải quyết bài toán trong đề án này cùng với những nguyên do lựa chọn chúng để giải quyết bài toán hiệu quả.

### 2. Công nghệ lưu trữ dữ liệu

## 2.1. Hệ quản trị cơ sở dữ liệu

Hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) là một loại hệ quản trị cơ sở dữ liệu không sử dụng mô hình quan hệ truyền thống của cơ sở dữ liệu SQL. Thay vì sử dụng các bảng, hàng và cột, hệ quản trị cơ sở dữ liệu phi quan hệ tập trung vào việc lưu trữ và truy xuất dữ liệu dựa trên các mô hình dữ liệu linh hoạt và không cấu trúc.

Một trong những đặc điểm nổi bật của hệ quản trị cơ sở dữ liệu phi quan hệ là khả năng lưu trữ dữ liệu không cấu trúc. Điều này có nghĩa là không cần phải định nghĩa một schema cố định trước khi lưu trữ dữ liệu. Thay vào đó, hệ quản trị cơ sở dữ liệu phi quan hệ cho phép lưu trữ dữ liệu với các định dạng linh hoạt như tài liệu JSON, cặp khóa-giá trị (key-value), hoặc các cấu trúc dữ liệu khác. Điều này cho phép lưu trữ và truy xuất dữ liệu có cấu trúc động, và thậm chí có thể thay đổi cấu trúc dữ liệu dễ dàng theo nhu cầu của ứng dụng.

Hệ quản trị cơ sở dữ liệu phi quan hệ cũng có khả năng mở rộng dễ dàng. Thay vì phải tăng cường phần cứng hoặc phân vùng dữ liệu như trong cơ sở dữ liệu SQL truyền thống, hệ quản trị cơ sở dữ liệu phi quan hệ cho phép mở rộng theo chiều ngang. Điều này có nghĩa là có thể phân tán dữ liệu trên nhiều máy chủ và xử lý khối lượng dữ liệu lớn một cách hiệu quả. Việc mở rộng ngang giúp tăng cường hiệu suất và khả năng chịu tải của hệ thống, đồng thời giúp giảm thiểu sự gián đoạn và đảm bảo tính sẵn sàng của cơ sở dữ liệu.

Hệ quản trị cơ sở dữ liệu phi quan hệ cũng hỗ trợ các tính năng truy vấn linh hoạt. Mỗi hệ quản trị cơ sở dữ liệu phi quan hệ có ngôn ngữ truy vấn riêng, cho phép thực hiện các truy vấn phức tạp và linh hoạt trên dữ liệu. Có thể sử dụng các truy vấn điều kiện, truy vấn sắp xếp, truy vấn kết hợp và truy vấn theo các thuộc tính khác nhau để tìm kiếm, lọc và sắp xếp dữ liệu theo ý muốn. Hệ quản trị cơ sở dữ liệu phi quan hệ cũng hỗ trợ các phương pháp tìm kiếm và truy vấn nhanh chóng trên khối lượng dữ liệu lớn.

Một ưu điểm nổi bật của hệ quản trị cơ sở dữ liệu phi quan hệ là khả năng xử lý dữ liệu có tính mở rộng và phân tán. Với khả năng mở rộng ngang và linh hoạt trong việc phân tán dữ liệu, hệ quản trị cơ sở dữ liệu phi quan hệ có thể đáp ứng yêu cầu của

các ứng dụng đòi hỏi hiệu suất cao và khả năng mở rộng. Điều này đặc biệt hữu ích trong các lĩnh vực như trực tuyến lớn (big data), Internet of Things (IoT) và các ứng dụng web quy mô lớn.

Hệ quản trị cơ sở dữ liệu phi quan hệ cũng giúp đơn giản hóa quy trình phát triển phần mềm. Với cấu trúc linh hoạt, bạn có thể thay đổi cấu trúc dữ liệu một cách dễ dàng khi yêu cầu của ứng dụng thay đổi. Điều này giúp giảm thiểu thời gian và công sức cần thiết cho việc thiết kế và triển khai cơ sở dữ liệu. Hơn nữa, hệ quản trị cơ sở dữ liệu phi quan hệ thường cung cấp các công cụ và giao diện lập trình ứng dụng (API) phong phú, giúp nhà phát triển tương tác với cơ sở dữ liệu một cách dễ dàng và hiệu quả.

Tuy nhiên, hệ quản trị cơ sở dữ liệu phi quan hệ cũng có nhược điểm. Một số nguyên tắc của hệ quản trị cơ sở dữ liệu SQL truyền thống như tính toàn vẹn, kiểm soát và khả năng tham chiếu khóa ngoại không được áp dụng trong hệ quản trị cơ sở dữ liệu phi quan hệ. Do đó, việc duy trì tính toàn vẹn và quản lý mối quan hệ giữa các bảng (collections) trong hệ quản trị cơ sở dữ liệu phi quan hệ có thể phức tạp hơn. Hơn nữa, trong một số trường hợp, hiệu suất truy vấn có thể bị ảnh hưởng do cấu trúc không cấu trúc của dữ liệu và sự phân tán trên nhiều máy chủ.

Để tổng kết, hệ quản trị cơ sở dữ liệu phi quan hệ là một sự thay đổi đáng kể so với hệ quản trị cơ sở dữ liệu SQL truyền thống. Với khả năng lưu trữ dữ liệu không cấu trúc, khả năng mở rộng ngang và tính linh hoạt trong truy vấn, hệ quản trị cơ sở dữ liệu phi quan hệ đã trở thành lựa chọn phổ biến trong việc xây dựng các ứng dụng hiệu suất cao và đáp ứng yêu cầu của các lĩnh vực công nghệ tiên tiến như big data và IoT. Mặc dù có nhược điểm riêng, hệ quản trị cơ sở dữ liệu phi quan hệ tiếp tục phát triển và được cải tiến để đáp ứng nhu cầu ngày càng tăng của các ứng dụng hiện đại.

## 2.2. Hệ quản trị cơ sở dữ liệu MongoDB



**Hình 1. Logo của mongoDB**

MongoDB là một hệ quản trị cơ sở dữ liệu phân tán (DBMS) không sử dụng cấu trúc dữ liệu cố định, mà thay vào đó sử dụng mô hình dữ liệu JSON như một cơ sở để lưu trữ thông tin. Với sự phát triển mạnh mẽ trong thời gian gần đây, MongoDB đã trở thành một trong những công cụ phổ biến và mạnh mẽ nhất trong lĩnh vực quản lý cơ sở dữ liệu phi quan hệ.

Một trong những đặc điểm nổi bật của MongoDB là khả năng lưu trữ dữ liệu theo mô hình không cấu trúc. Thay vì sử dụng các bảng, hàng và cột như trong cơ sở dữ liệu quan hệ, MongoDB sử dụng tài liệu JSON để lưu trữ dữ liệu. Điều này cho phép lưu trữ dữ liệu có cấu trúc linh hoạt, với khả năng mở rộng dễ dàng khi cần thiết.

Với MongoDB, có thể lưu trữ các tài liệu có cấu trúc khác nhau trong cùng một bảng (collection), mà không cần tuân theo một schema cố định. Điều này đồng nghĩa với việc bạn có thể thay đổi cấu trúc dữ liệu theo nhu cầu và phát triển ứng dụng một cách linh hoạt mà không cần phải thay đổi cấu trúc cơ sở dữ liệu.

MongoDB cung cấp khả năng mở rộng ngang (horizontal scaling) để xử lý khối lượng dữ liệu lớn. Bằng cách chia nhỏ dữ liệu thành các phân đoạn (shard), MongoDB có thể phân tán dữ liệu trên nhiều máy chủ, giúp tăng khả năng chịu

tải và hiệu suất của hệ thống. Điều này rất hữu ích khi bạn đang xây dựng ứng dụng có khối lượng truy vấn lớn hoặc phải xử lý dữ liệu với tốc độ cao.

Một tính năng quan trọng khác của MongoDB là khả năng truy vấn dữ liệu linh hoạt. MongoDB cung cấp một ngôn ngữ truy vấn mạnh mẽ và linh hoạt, cho phép bạn

tìm kiếm và lọc dữ liệu theo các tiêu chí phức tạp. Có thể sử dụng các truy vấn điều kiện, truy vấn với các toán tử so sánh, truy vấn trên các phần tử của tài liệu JSON và thậm chí cả truy vấn sắp xếp và truy vấn kết hợp giữa các bảng (collection).

MongoDB cũng hỗ trợ các tính năng quan trọng khác như replica set và sharding. Replica set cho phép sao lưu và đồng bộ dữ liệu giữa các máy chủ, đảm bảo tính sẵn sàng và bảo mật của dữ liệu. Sharding, như đã đề cập trước đó, cho phép phân tán dữ liệu trên nhiều máy chủ, giúp tăng khả năng chịu tải và khả năng mở rộng của hệ thống.

MongoDB cũng hỗ trợ các giao thức truy cập linh hoạt như RESTful API, JDBC và ODBC, giúp các nhà phát triển tích hợp dễ dàng MongoDB vào ứng dụng của họ. Ngoài ra, MongoDB cũng có các thư viện và driver hỗ trợ cho nhiều ngôn ngữ lập trình phổ biến như Java, Python, Node.js, và C#, giúp việc phát triển ứng dụng trở nên thuận tiện và nhanh chóng.

Với cộng đồng người dùng rộng lớn và sự phát triển liên tục, MongoDB có sự hỗ trợ tốt từ cộng đồng và nhà phát triển. Có thể tìm thấy tài liệu, hướng dẫn và ví dụ phong phú trên trang web chính thức của MongoDB và các nguồn thông tin khác để học và nắm bắt kiến thức về MongoDB.

Tổng kết lại, MongoDB là một hệ quản trị cơ sở dữ liệu phân tán mạnh mẽ, linh hoạt và dễ sử dụng. Với khả năng lưu trữ dữ liệu không cấu trúc, khả năng mở rộng ngang và tính năng truy vấn linh hoạt, MongoDB là lựa chọn lý tưởng cho việc xây dựng các ứng dụng hiệu suất cao và có khả năng mở rộng.

### 3. Ngôn ngữ lập trình và framework sử dụng

#### 3.1. Lập trình hướng đối tượng

Lập trình hướng đối tượng (Object-Oriented Programming – OOP) là một mô thức lập trình dựa trên nền tảng là hướng đối tượng để mô hình, thiết kế các sản phẩm sao cho chúng hoạt động giống như hoạt động thật ngoài thực tế. Mô thức lập trình này khi ra đời đã giải quyết được một số vấn đề còn tồn đọng trước đó trong giai đoạn phương pháp lập trình cấu trúc đang phổ biến. Việc lập trình cấu trúc không giải quyết tốt các bài toán phức tạp, hay không tái sử dụng được các mô đun đã lập trình trong cùng sản phẩm dẫn tới sự ra đời của lập trình hướng đối tượng. Hướng đối tượng là một kỹ thuật, trong đó kỹ

thuật này cho phép việc mô hình hóa các đối tượng trong thực tế vào trong một phần mềm để mô tả lại thế giới thực, vì vậy trọng tâm của kỹ thuật này chính là đối tượng.

Đối tượng (Object) chính là khái niệm trung tâm trong lập trình hướng đối tượng, dùng để mô hình hóa lại một thực thể trong hiện thực. Để mô hình hóa được thực thể, đối tượng cần phải có những thuộc tính bao gồm những thông tin, những đặc tính về thực thể đó, và đối tượng cần có phương thức (hay thông điệp) để đối tượng có thể trao đổi, giao tiếp với các đối tượng khác. Tiếp đến là khái niệm Lớp (Class) chính là tập hợp những thuộc tính, những phương thức chung của một nhóm các đối tượng có liên hệ gần với nhau. Có thể nói Lớp chính là khái quát hóa của Đối tượng.

Các tính chất chính của lập trình hướng đối tượng đem lại ưu thế trước các mô thức lập trình khác bao gồm. Tính đóng gói (encapsulation) là tính chất quy định việc trừu tượng hóa các thuộc tính, các hành vi liên quan tới nhau được đóng gói lại trong một khái niệm là lớp, sau khi đóng gói thì đối tượng có thể thực hiện các hành vi, tính năng cần thiết trong nội bộ lớp đó, và nhờ có việc này nên đảm bảo được việc chặn những truy cập trái phép từ bên ngoài, và sinh ra tính che giấu thông tin chi tiết của đối tượng với bên ngoài, bên ngoài chỉ có thể tương tác với đối tượng thông qua những phương thức mà đối tượng cung cấp công khai ra, còn bên trong xử lý như thế nào sẽ được che giấu hoàn toàn. Tính chất thứ hai được kể tới của lập trình hướng đối tượng là tính kế thừa (inheritance), đây là tính chất mô tả quan hệ phân cấp giữa các lớp, cho phép các lớp dẫn xuất được kế thừa lại các tính chất, các phương thức của lớp cơ sở để tái sử dụng lại các tính chất, phương thức đã được định nghĩa trước. Nhờ có tính chất này mà lập trình hướng đối tượng có khả năng tái sử dụng phần mềm, tái sử dụng các mô đun để làm giảm bớt dư thừa, tránh lặp trong quá trình phát triển. Tính chất kế đến là tính đa hình (polymorphism), là một tính chất đi cùng với tính kế thừa. Tính chất này cho phép các đối tượng khác nhau có thể nhận cùng một thông điệp giống nhau nhưng được thực thi một cách khác nhau theo từng

đối tượng. Để có thể dùng tính chất này, các lớp chứa các đối tượng đó cần phải kế thừa một lớp cơ sở, và từ đó ghi đè các phương thức của lớp cơ sở đó để có thể hành xử đúng như mong muốn cho từng lớp dẫn xuất này. Tính chất này đem lại khả năng tổng quát hóa cao trong quá trình thiết kế và xây dựng phần mềm.



Hiện nay có nhiều ngôn ngữ lập trình được xây dựng theo hướng lập trình hướng đối tượng và được sử dụng phổ biến trong lập trình xây dựng các sản phẩm. Một vài ngôn ngữ lập trình có hỗ trợ lập trình hướng đối tượng có thể kể ra như C++, Java, .NET, PHP, Ruby, Python, . . . Nhờ những ưu điểm trên đây, cùng với sự phù hợp để mô hình hóa các thực thể trong thực tế trở thành các đối tượng trong sản phẩm được phát triển, em lựa chọn lập trình hướng đối tượng là phương pháp để phát triển sản phẩm trong đồ án này.

### 3.2. Ngôn ngữ lập trình: Javascript



**Hình 1. Logo của Javascript**

Javascript là một ngôn ngữ lập trình phổ biến, được sử dụng rộng rãi trên web để thêm tính năng động vào các trang web. Ngôn ngữ này được tạo ra bởi Brendan Eich vào năm 1995 và được phát triển bởi Netscape Communications Corporation. Ngày nay, Javascript được sử dụng không chỉ trên web mà còn trên các ứng dụng di động, máy tính để bàn và các thiết bị khác.

Javascript là một ngôn ngữ lập trình đa nền tảng, cho phép các lập trình viên tạo ra các ứng dụng web động, tương tác và thân thiện với người dùng. Nó được sử dụng để tạo ra các chức năng tương tác như kiểm tra thông tin đăng nhập, hiển thị thông báo, và đưa ra các hành động phản hồi từ người dùng. Một trong những lợi ích của Javascript là nó cho phép các lập trình viên tạo ra các ứng dụng động không cần phải tải lại trang web, giúp cải thiện trải nghiệm người dùng.

Javascript có thể được sử dụng để tạo ra các chức năng cơ bản như điều khiển phần tử trên trang web, tạo hiệu ứng và tương tác đơn giản. Tuy nhiên, với sự phát triển

của các framework như React, Vue, Angular và Nodejs, các lập trình viên có thể sử dụng Javascript để tạo ra các ứng dụng phức tạp, động và tương tác cao.

Javascript cũng cho phép tương tác với các API của bên thứ ba, cho phép các lập trình viên tạo ra các ứng dụng kết nối với các dịch vụ web khác, chẳng hạn như các dịch vụ định vị, dịch vụ thanh toán và các dịch vụ mạng xã hội.

Các lập trình viên có thể sử dụng các trình duyệt web để phát triển Javascript, hoặc sử dụng các trình biên dịch Javascript để tạo ra các ứng dụng độc lập. Các trình biên dịch Javascript như TypeScript và CoffeeScript cung cấp các tính năng mở rộng và cải tiến cho ngôn ngữ Javascript gốc.

Javascript cũng có thể được sử dụng để tạo ra các ứng dụng di động với các framework như React Native và Ionic. Ngoài ra, Javascript cũng được sử dụng để tạo ra các ứng dụng máy tính như các ứng dụng desktop và các ứng dụng trên các thiết bị di động, như Raspberry Pi và Arduino.

Ngôn ngữ Javascript được thiết kế để dễ học và có cú pháp đơn giản. Nó cũng hỗ trợ các tính năng lập trình hướng đối tượng, giúp các lập trình viên có thể tạo ra các ứng dụng phức tạp mà không cần phải viết lại mã nhiều lần.

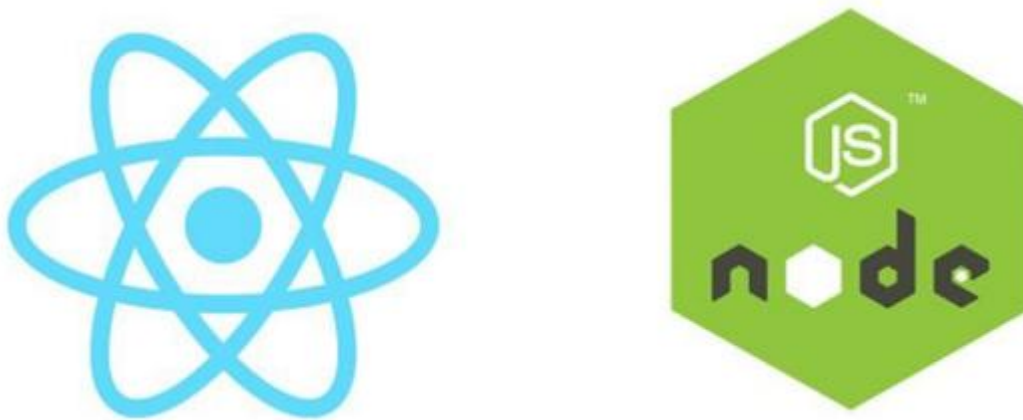
Một trong những thế mạnh của Javascript là khả năng tương tác với HTML và CSS, cho phép các lập trình viên thao tác với các thành phần trên trang web và tạo ra các hiệu ứng đẹp mắt cho người dùng. Ngoài ra, Javascript còn có thể tương tác với các thư viện Javascript khác như jQuery, AngularJS và ReactJS, giúp các lập trình viên dễ dàng phát triển các ứng dụng web.

Trong quá trình phát triển, Javascript đã trải qua nhiều thay đổi và cải tiến. Hiện nay, phiên bản mới nhất của Javascript là ES2022, với nhiều tính năng mới được thêm vào như toán tử tìm kiếm và cập nhật của mảng. Với sự phát triển không ngừng của công nghệ và sự yêu cầu ngày càng cao của người dùng, Javascript sẽ tiếp tục được cải tiến và phát triển để đáp ứng nhu cầu của thị trường.

Tóm lại, Javascript là một ngôn ngữ lập trình rất quan trọng và phổ biến trong lĩnh vực web. Nó cho phép các lập trình viên tạo ra các ứng dụng web động, tương tác và thân thiện với người dùng. Với tính năng đa nền tảng và khả năng tương tác với các

API của bên thứ ba, Javascript được sử dụng rộng rãi trên nhiều thiết bị và ứng dụng khác nhau. Các lập trình viên có thể sử dụng Javascript để tạo ra các ứng dụng đơn giản hoặc phức tạp, và ngôn ngữ này còn được cải tiến và phát triển để đáp ứng các yêu cầu ngày càng cao của thị trường.

### 3.3. Framework: NodeJS + React JS + ReactNative



**Hình 3. Logo của React JS và Node JS**

Node.js, React và React Native là ba framework phổ biến trong cộng đồng phát triển ứng dụng web và di động. Mỗi framework đóng vai trò quan trọng trong việc xây dựng ứng dụng hiệu suất cao, tương tác và dễ dàng mở rộng.

Node.js là một môi trường chạy mã JavaScript ở phía server, cho phép xây dựng các ứng dụng web đa nền tảng và chạy trên máy chủ. Node.js có thể xử lý các yêu cầu web đồng thời, xây dựng API RESTful, quản lý trạng thái phi ngắn hạn của ứng dụng và sử dụng JavaScript cả ở phía server và phía client. Node.js cung cấp một hệ sinh thái mạnh mẽ của các thư viện và công cụ hỗ trợ, giúp giảm thiểu thời gian và công sức phát triển.

React là một thư viện JavaScript phát triển bởi Facebook, được sử dụng để xây dựng giao diện người dùng động và tương tác trên web. React sử dụng cơ chế viết lại (reconciliation) thông minh để cập nhật và hiển thị các thành phần (components) chỉ khi cần thiết, giúp tăng hiệu suất của ứng dụng. React cũng hỗ trợ việc tái sử dụng thành phần, cho phép bạn xây dựng giao diện module và dễ dàng quản lý trạng thái của ứng dụng. React đã trở thành một trong những lựa

chọn phổ biến nhất cho phát triển giao diện người dùng trong việc xây dựng các ứng dụng web đơn trang (single-page applications) và ứng dụng di động. Trong đồ án này, em sử dụng React như một công cụ để xây dựng lên giao diện phía Admin và giao diện phía quản lý bãi đỗ. Điều này cho phép họ có thể thuận tiện sử dụng cả điện thoại hoặc máy tính để tương tác với phần mềm.

React Native là một framework xây dựng ứng dụng di động đa nền tảng sử dụng JavaScript và React. Với React Native, có thể phát triển ứng dụng di động cho cả iOS và Android từ một mã nguồn duy nhất. Framework này cung cấp các thành phần (components) gốc của hệ điều hành di động, cho phép xây dựng giao diện di động chất lượng cao và tương tác mượt mà. React cũng hỗ trợ truy cập vào các tính năng phần cứng của điện thoại như máy ảnh, cảm biến và GPS. Điều này giúp giảm thiểu công sức phát triển và đảm bảo tính đa nền tảng của ứng dụng. Em đã sử dụng React Native để xây dựng phần tương tác của người dùng dưới dạng application, cho phép người dùng tương tác với phần mềm dễ dàng hơn thông qua điện thoại. Điều này sẽ mang lại trải nghiệm tuyệt vời và tương tác tốt người dùng.

Tổng kết lại, Node.js, React và React Native là ba framework quan trọng trong lĩnh vực phát triển ứng dụng web và di động. Node.js cho phép xây dựng ứng dụng server-side hiệu suất cao, trong khi React và React Native tập trung vào việc xây dựng giao diện người dùng tương tác và đa nền tảng. Sử dụng các framework này, có thể tận dụng sức mạnh của JavaScript để phát triển ứng dụng chất lượng cao, tối ưu và dễ dàng mở rộng. Với cộng đồng phát triển lớn và hỗ trợ từ các công cụ và thư viện khác nhau, Node.js, React và React Native đóng vai trò quan trọng trong việc xây dựng các ứng dụng hiện đại và đáp ứng nhu cầu ngày càng tăng của người dùng.

### 3.4. Kết chương

Trong chương này, nhóm em đã trình bày lý thuyết và các công nghệ được sử dụng cùng với lý do lựa chọn các công nghệ này trong quá trình phát triển sản phẩm. Những công nghệ đó được áp dụng và phát triển sản phẩm, dẫn đến một sản phẩm hoàn chỉnh sẽ được trình bày trong chương tiếp theo

## CHƯƠNG 4. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

### 1. Tổng quan

Trong chương trước em đã nêu ra và phân tích nguyên do lựa chọn các công nghệ, thư viện, tính năng được áp dụng trong phát triển sản phẩm cho project này.

Tiếp theo trong chương này em sẽ đi vào phần thiết kế chi tiết hệ thống và sử dụng những công nghệ trên để giải quyết những vấn đề và tiến hành xây dựng hệ thống sản phẩm trong các phần kế tiếp.

### 2. Thiết kế kiến trúc

#### 2.1. Lựa chọn kiến trúc phần mềm

Kiến trúc em lựa chọn để làm phần mềm là kiến trúc MVC. Mẫu thiết kế MVC (Model-View-Controller) được phát triển bởi Trygve Reenskaug tại Xerox Palo Alto Research Center (PARC) trong quá trình phát triển ngôn ngữ lập trình Smalltalk-76.

MVC là viết tắt của Model-View-Controller, một mẫu thiết kế phần mềm nhằm tách biệt các phần của một ứng dụng thành ba thành phần chính:

- **Model (M):**

- Đại diện cho dữ liệu của ứng dụng và logic nghiệp vụ.
- Quản lý trạng thái của ứng dụng và thường tương tác với cơ sở dữ liệu.
- Model không biết gì về View hay Controller.

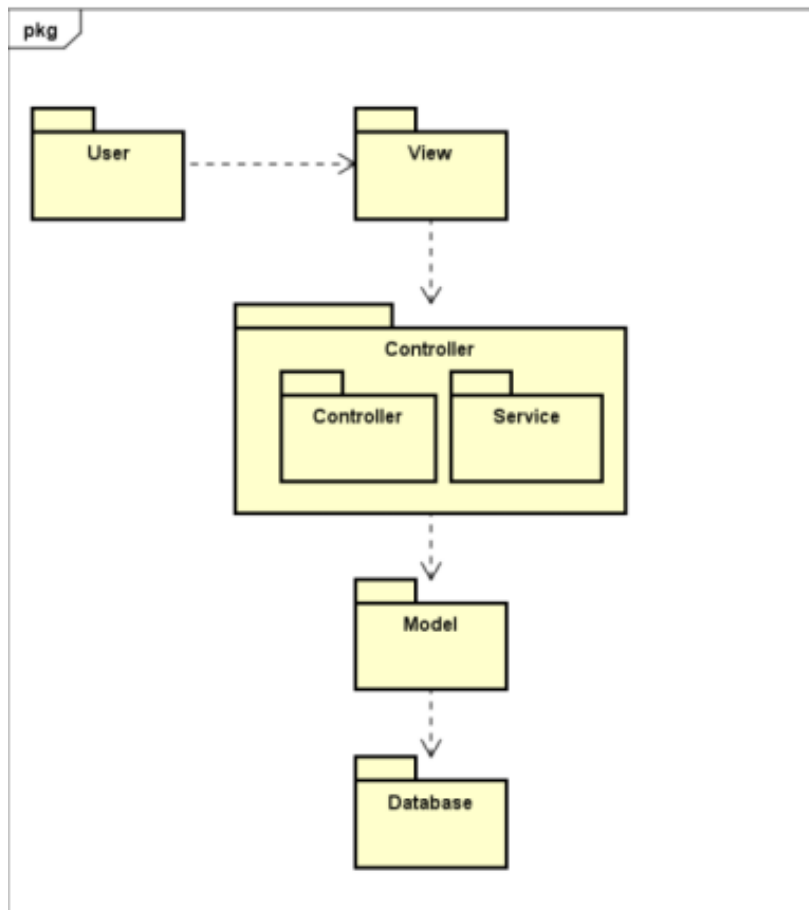
- **View (V):**

- Đại diện cho giao diện người dùng của ứng dụng.
- Hiển thị dữ liệu từ Model cho người dùng và gửi các yêu cầu đầu vào từ người dùng đến Controller.
- View không chứa logic nghiệp vụ.

- **Controller (C):**

- Xử lý đầu vào từ người dùng và tương tác với Model.
- Quyết định dữ liệu nào từ Model sẽ được hiển thị trong View.
- Controller đóng vai trò trung gian giữa Model và View.

## 2.2. Thiết kế tổng quan



**Hình 1. Hình ảnh thiết kế tổng quan của MVC**

Đầu tiên là gói về User, gói này không nằm trong hệ thống mà đại diện cho nhóm đối tượng người sử dụng hệ thống, có khả năng tương tác với hệ thống dựa trên những gì được đưa ra bởi gói View trong quá trình hoạt động.

Kế đến là gói View, gói này có nhiệm vụ lưu trữ, quản lý các đối tượng giao diện như nút bấm, biểu mẫu, . . . Tiếp đến gói này cung cấp khả năng tương tác cho các đối tượng người sử dụng được mô tả trong gói User, nhận những yêu

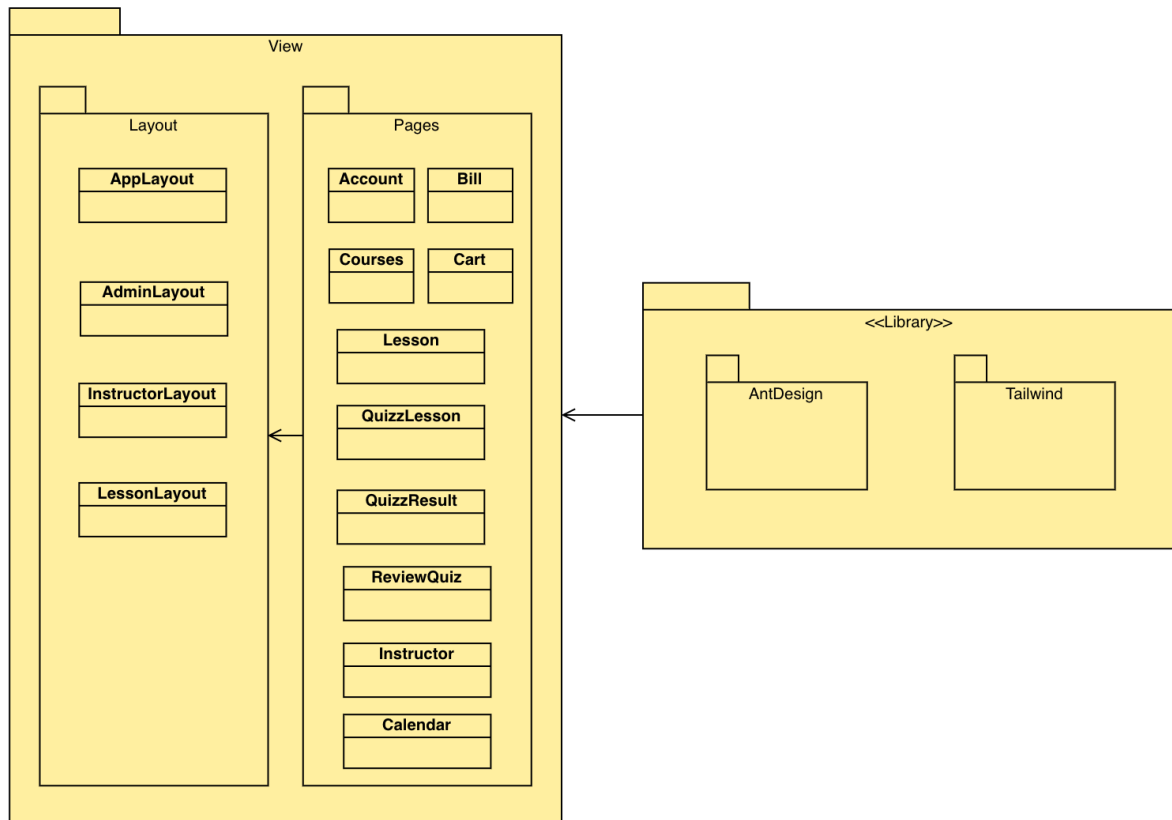
cầu từ người sử dụng chuyển xuống cho gói Controller xử lý, và ngược lại nhận những phản hồi, những dữ liệu từ gói Controller để xử lý hiển thị ra cho người sử dụng. Gói này phụ thuộc vào gói Controller để xử lý được hoạt động của mình.

Tiếp đến là gói Controller, gói này đảm nhận vai trò nhận yêu cầu được chuyển về từ người dùng, tiến hành điều hướng, xử lý các nghiệp vụ logic cần thiết, và chuyển hướng các yêu cầu xuống dưới cho gói Model phụ trách tiếp các yêu cầu đó. Sau đó, gói sẽ nhận các trả về từ gói Model là các dữ liệu cần thiết, tiếp tục xử lý nghiệp vụ logic và chuyển kết quả lên phía trên cho gói View thực hiện tiếp nhiệm vụ của mình. Trong gói Controller được chia nhỏ thành hai gói Controller và gói Service, trong đó gói Controller bên trong có chức năng giống với gói Controller bên ngoài, và gói Service sẽ trợ giúp gói Controller bên trong thông qua việc xử lý một số tác vụ khó khăn, hay tác vụ chạy ngầm. Gói này phụ thuộc vào gói Model để xử lý được hoạt động của mình.

Phía dưới là gói Model, gói này có vai trò là nơi thiết lập liên kết giữa hệ thống với hệ quản trị cơ sở dữ liệu, là nơi xử lý những thao tác liên quan đến dữ liệu như truy vấn nội dung, chỉnh sửa thông tin dữ liệu trong hệ thống. Gói này sẽ nhận những yêu cầu từ gói Controller và sau đó tiến hành các truy vấn, các yêu cầu thực hiện đúng mục đích chuyển xuống cho gói Database xử lý, sau đó nhận lại kết quả, xử lý dữ liệu và chuyển lên phía trên cho gói Controller. Gói này phụ thuộc vào gói Database để xử lý được hoạt động của mình.

Cuối cùng là gói Database, gói này chứa hệ quản trị cơ sở dữ liệu, có vai trò lưu trữ và thực hiện các truy vấn dữ liệu. Gói sẽ nhận những yêu cầu truy vấn từ gói Model, sau đó thực thi các truy vấn và gửi trả lại kết quả lên cho gói Model tiếp tục xử lý.

### 2.3. Thiết kế chi tiết gói

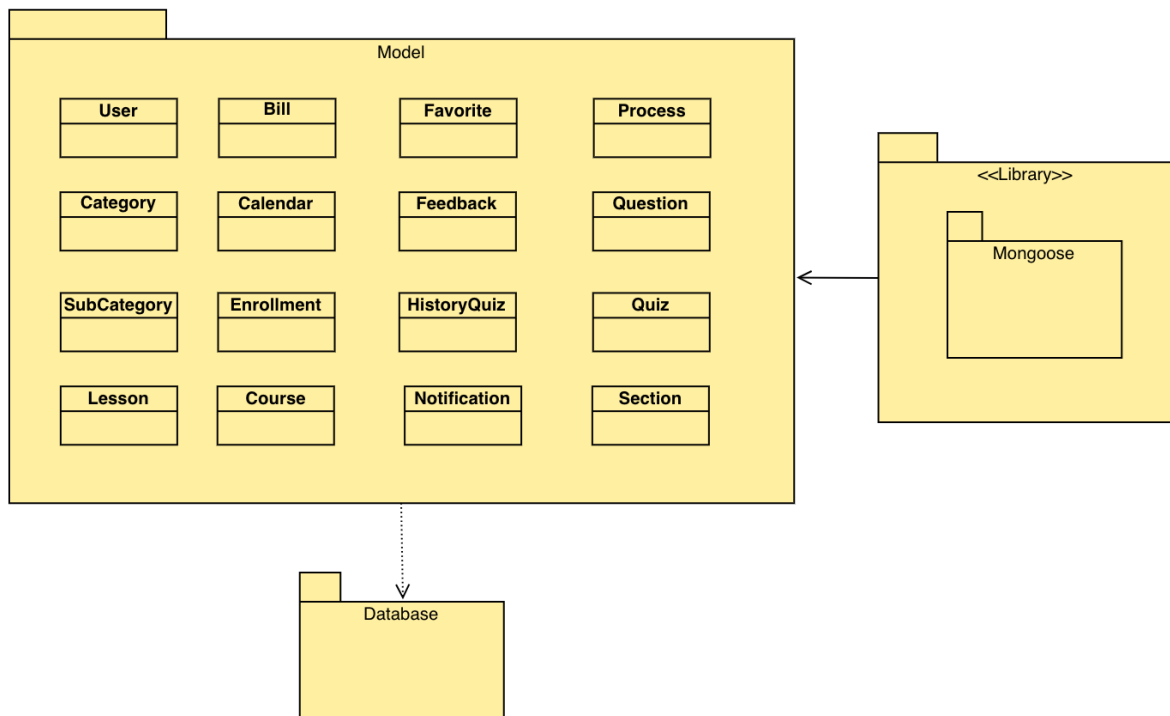


**Hình 2. Hình ảnh chi tiết gói View**

Trong hình 2 này mô tả biểu đồ gói chi tiết cho gói View trong hệ thống. Gói View này có phụ thuộc vào một số thư viện bên ngoài như thư viện Tailwind và thư viện Antd để giúp tạo giao diện nhanh và đồng nhất, hỗ trợ sử dụng và tạo ra những tính năng dùng Javascript nhanh chóng và dễ dàng hơn.

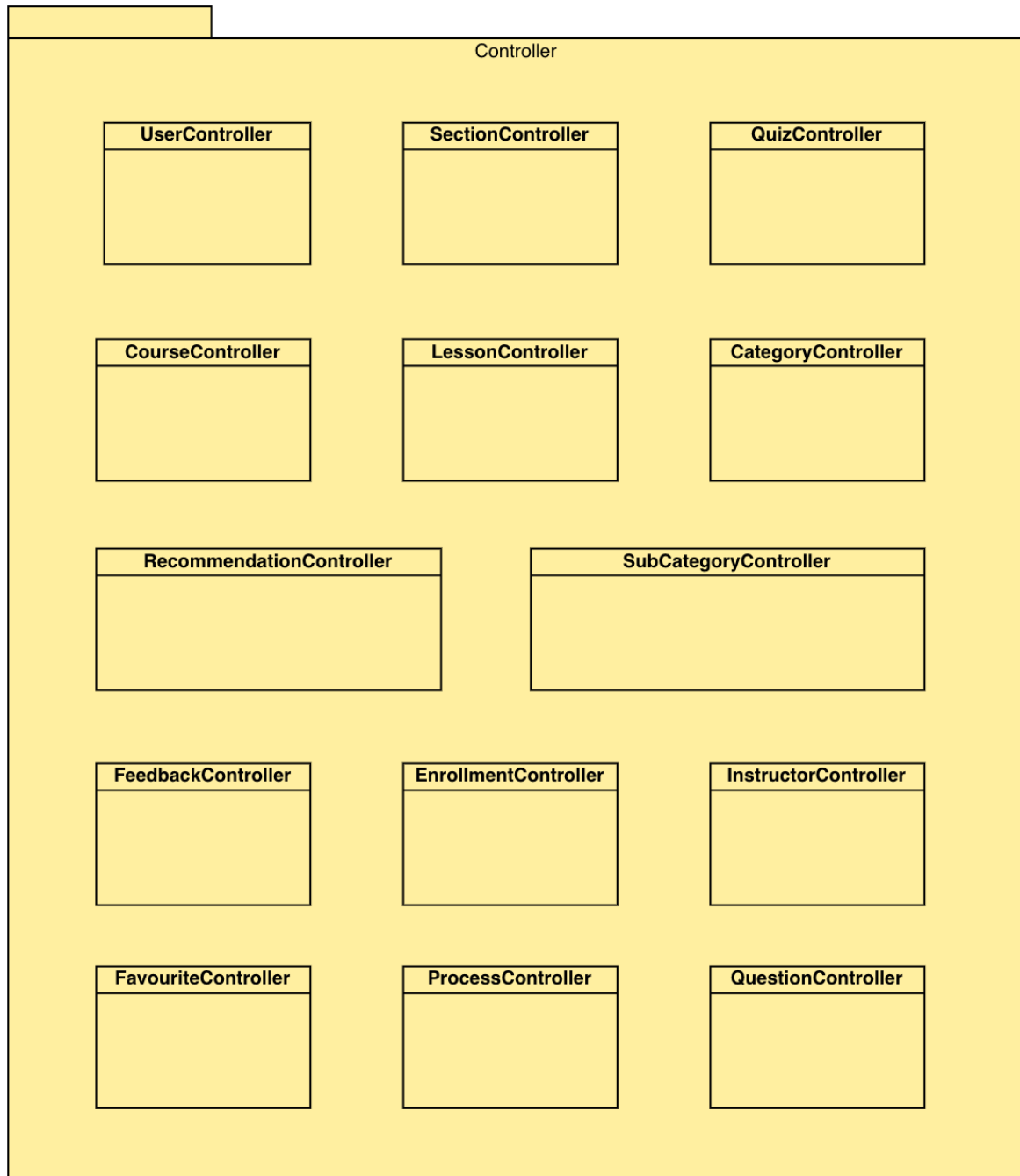
Gói View này được chia ra các lớp theo các nhóm đối tượng hiển thị. Trong đó có một lớp LayoutView bao gồm khuôn mẫu định dạng hiển thị tiêu chuẩn, các lớp giao diện khác sẽ kế thừa và chỉnh sửa các thành phần nội dung phù hợp với lớp đó. Tại đây có lớp LoginView về giao diện đăng nhập sẽ hợp thành bên trong MapView là giao diện web tìm kiếm.





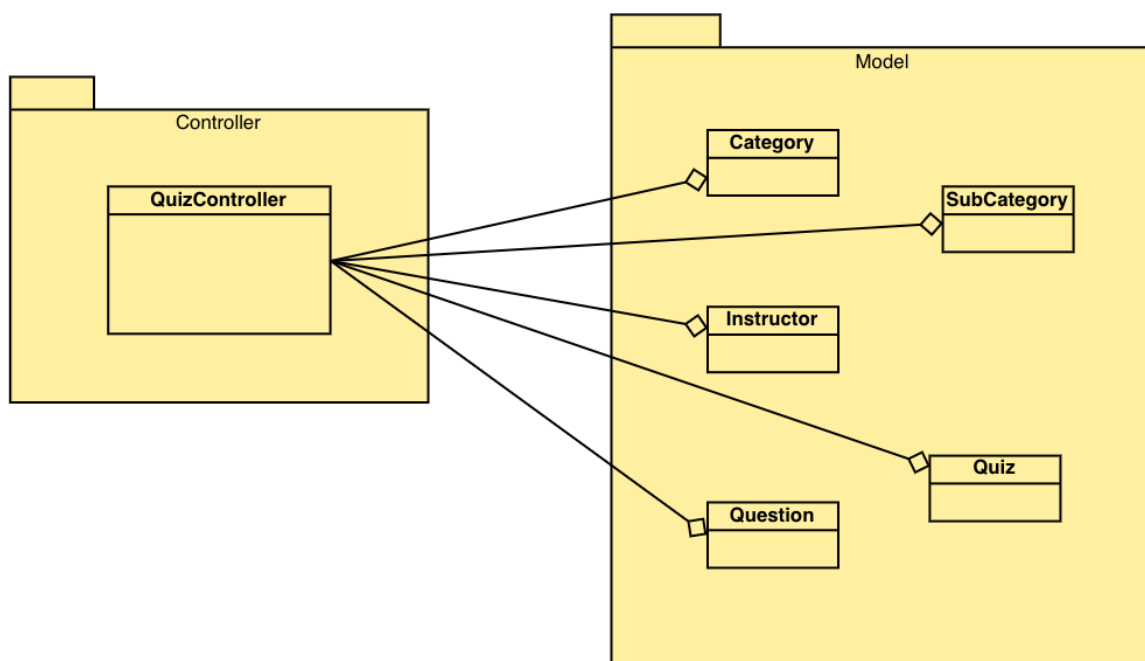
**Hình 3. Hình ảnh chi tiết gói Model**

Kế tiếp là thiết kế chi tiết cho biểu đồ gói Model được minh họa trên hình 3. Gói Model có phụ thuộc vào một số thư viện, trong đó một thư viện quan trọng là mongoose, thư viện này có vai trò giúp thiết lập và tạo kết nối giữa gói Model và gói Database để có khả năng truy cập đồng thời thực hiện các truy vấn với hệ quản trị cơ sở dữ liệu MongoDB được đặt nằm trong gói Database này.



**Hình 4. Hình ảnh chi tiết gói Controller**

Cuối cùng là biểu đồ gói chi tiết cho gói Controller, được minh họa trên hình 4. Gói này có các lớp có tính tương ứng với các lớp trong gói View và gói Model có liên hệ với gói. Ví dụ lớp UserController là lớp điều khiển luồng xử lý logic của việc làm, có mối liên hệ với lớp MapView bên gói View và liên hệ với lớp User trong lớp Model để tạo thành một liên kết xuyên suốt. Gói Controller bên trong chứa những lớp xử lý nghiệp vụ logic cơ bản ứng với các lớp bên dưới gói Model và các lớp trên bên gói View.



**Hình 5. Thiết kế gói cho chức năng tạo bài tập**

### 3. Thiết kế chi tiết

#### 3.1. Thiết kế giao diện

Các thông số của màn hình mà ứng dụng hướng tới được đặc tả. Các màn hình được sử dụng bao gồm cả màn hình máy tính để bàn, màn hình máy tính xách tay, và màn hình điện thoại di động, màn hình máy tính bảng.

Tính chất	Giá trị
Kích cỡ	6,1 inch; 14 inch, 15.6 inch, 19 inch, 20 inch
Độ phân giải	1280x720, 1366x768, 1920x1080
Tỉ lệ màn hình	18:9, 19:9, 19.5:9, 20:9, 16:9, 4:3
Số lượng màu sắc hỗ trợ	16 triệu
Tần số quét	60Hz, 120Hz

## Đặc tả thông tin về màn hình

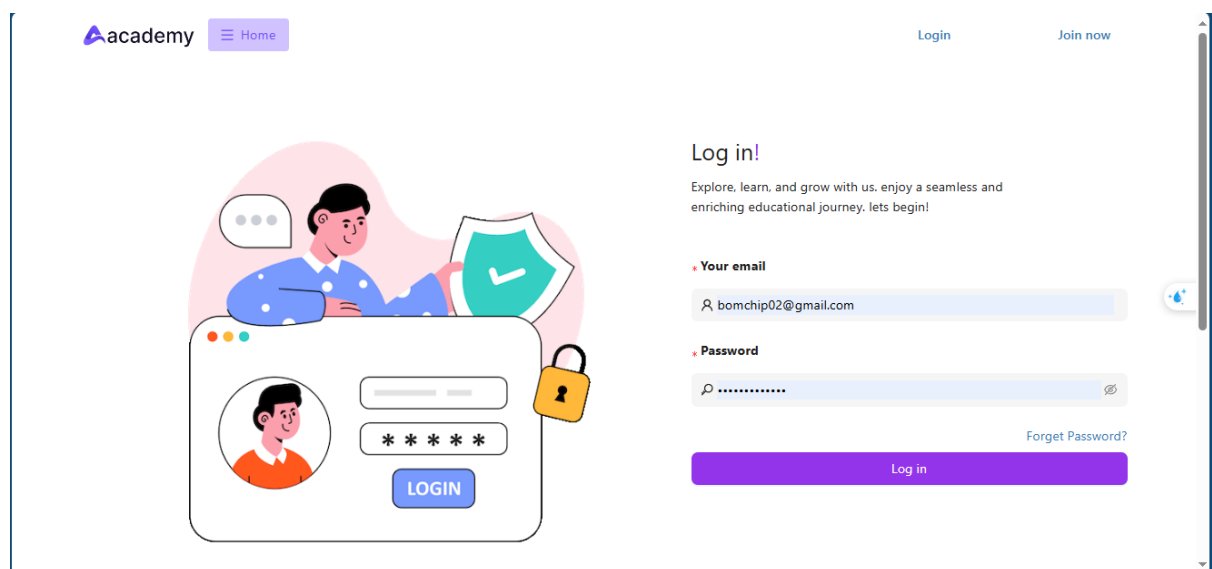
Các chuẩn hóa thiết kế giao diện với các đối tượng được mô tả dưới đây.

Về nút bấm, nút bấm có kích thước phù hợp với nội dung bên trong nút bấm. Màu sắc của nút bấm được thể hiện cho các chức năng tương ứng, trong đó màu đỏ bao gồm các chức năng liên quan đến xác nhận, chuyển trạng thái, nút bấm màu xám đặc trưng cho không thực hiện hành động và thể hiện việc thực hiện các hành động như xóa hay từ chối một nội dung nào đó.

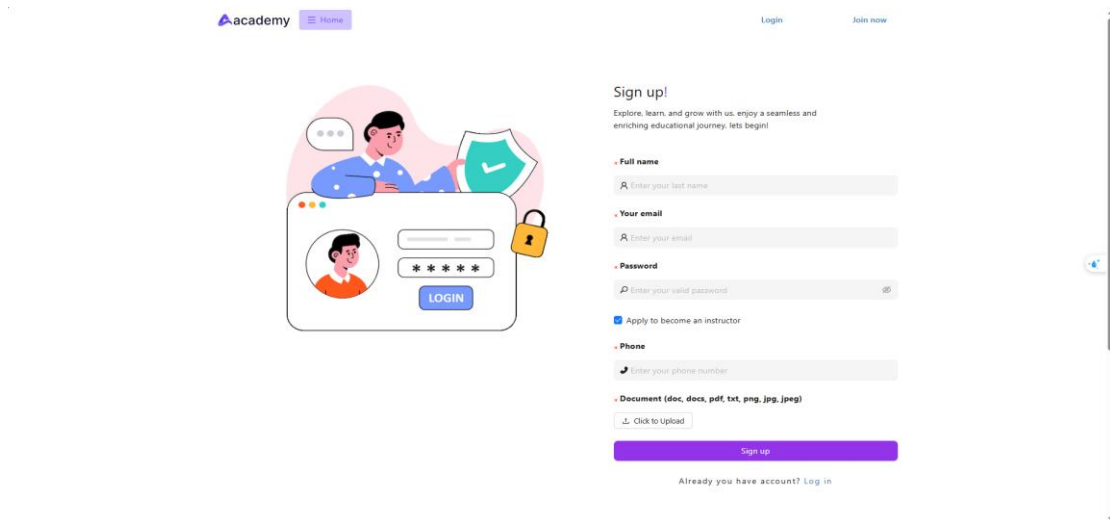
Các thông báo được đưa ra màn hình ở góc trên bên phải. Các thông báo này chỉ xuất hiện trong vòng vài giây và tự động biến mất. Màu sắc của thông báo thể hiện được kết quả của các hành động trước đó. Màu đỏ thể hiện hành động trước đó đã được thực thi thành công, popup thể hiện hành động trước đó đã thất bại hoặc có xảy ra lỗi trong quá trình thực thi.

Giao diện có màu sắc tươi sáng, tông màu chính là màu trắng, một số màu phụ đạo chủ yếu là màu đỏ nhẹ để giúp cho hiển thị trở nên rõ ràng. Các biểu mẫu được thiết kế căn lề phân tách giữa cột tên các trường và cột điền nội dung. Một số biểu mẫu được thiết kế dưới dạng một popup hiển thị để không cần chuyển trang trong quá trình sử dụng hệ thống.

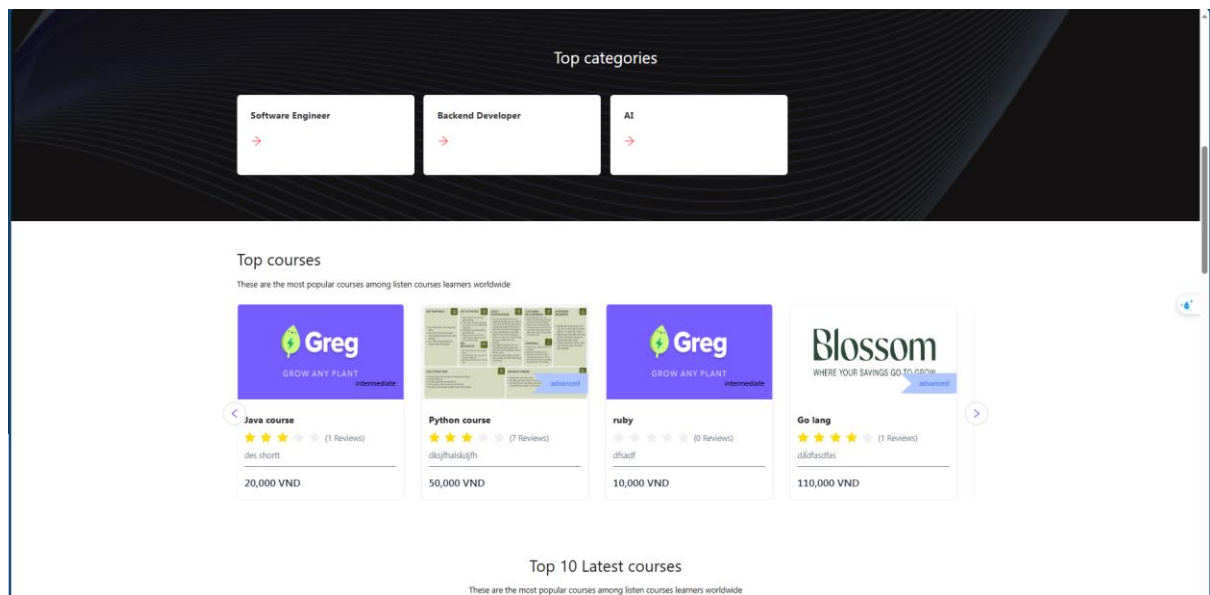
Dưới đây là một số hình ảnh minh họa thiết kế giao diện.



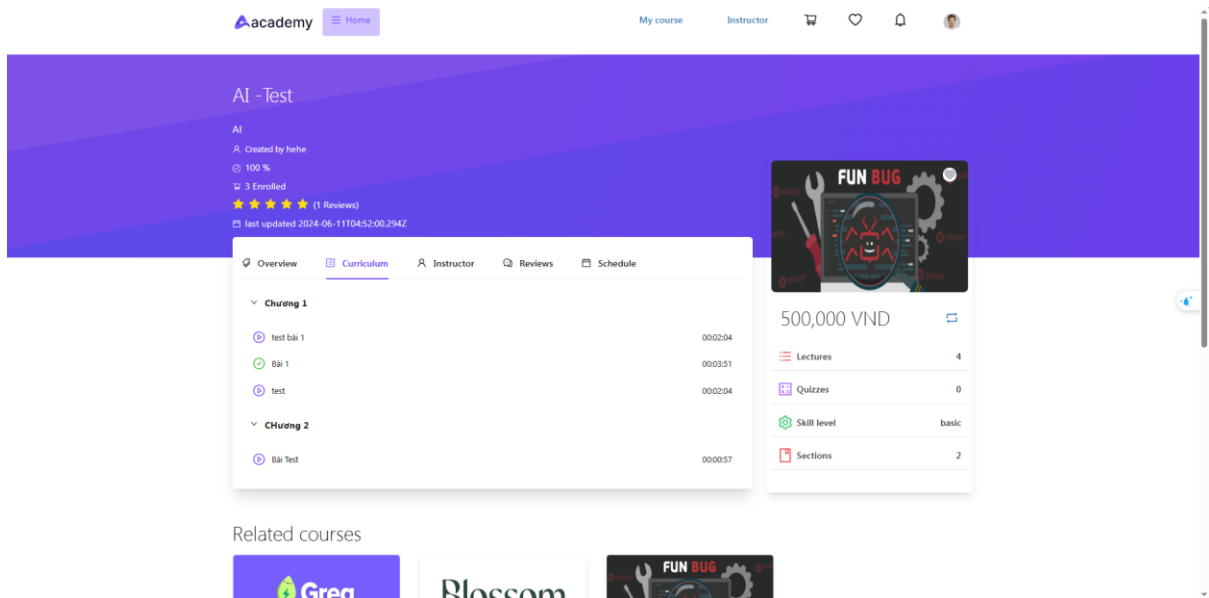
**Hình 5. Minh họa giao diện phần đăng nhập**



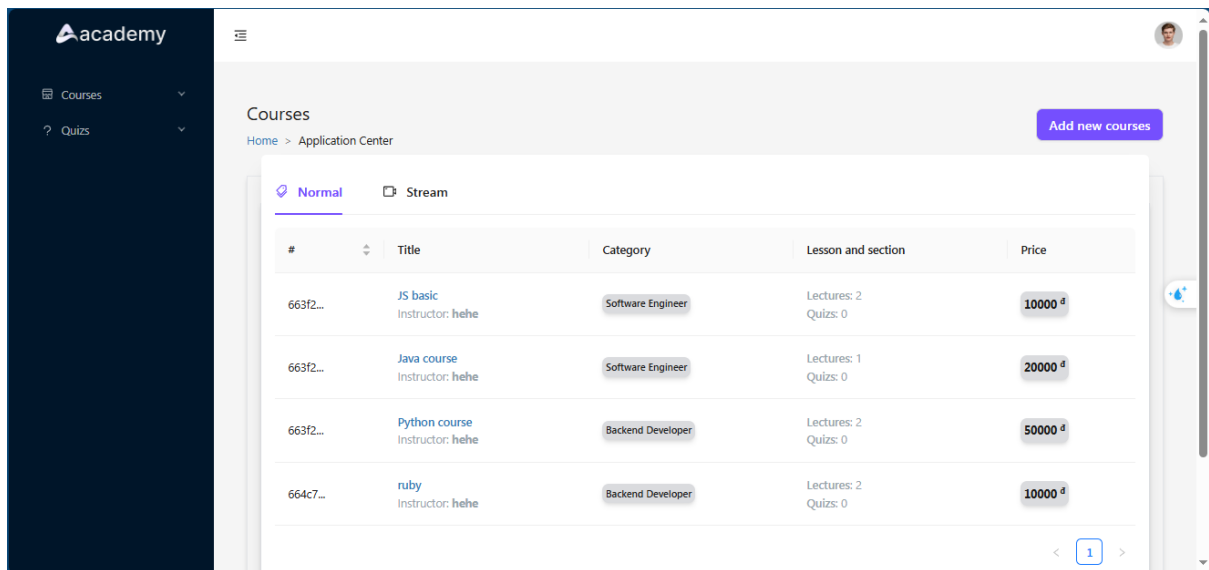
**Hình 6. Minh hoạ giao diện phần đăng ký**



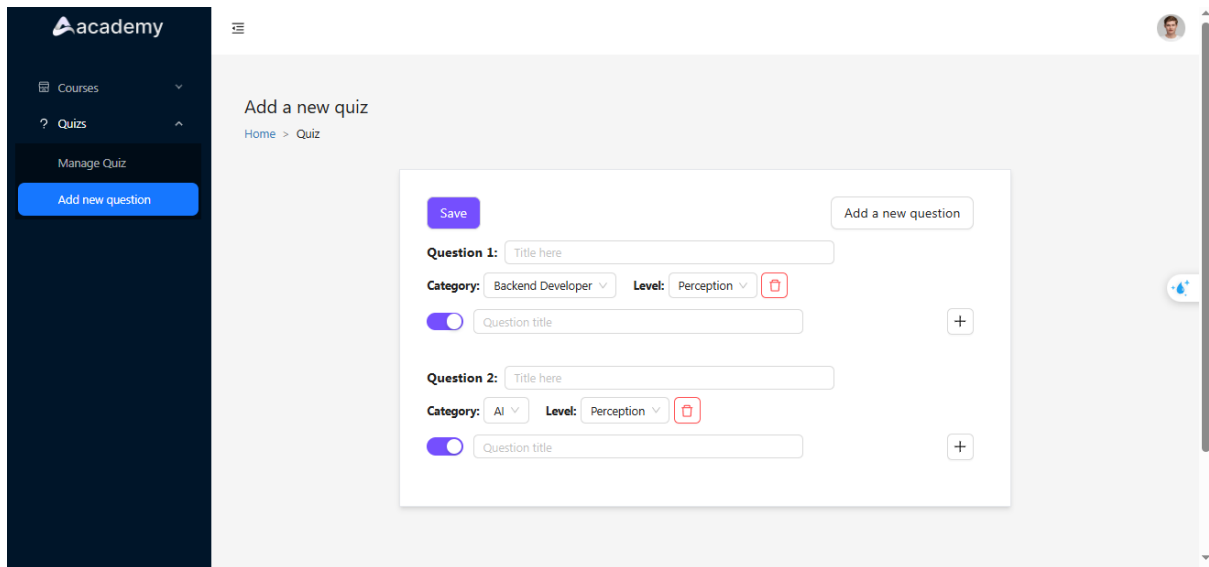
**Hình 7. Minh hoạ giao diện trang chủ**



**Hình 8. Minh hoạ giao diện chi tiết khóa học**

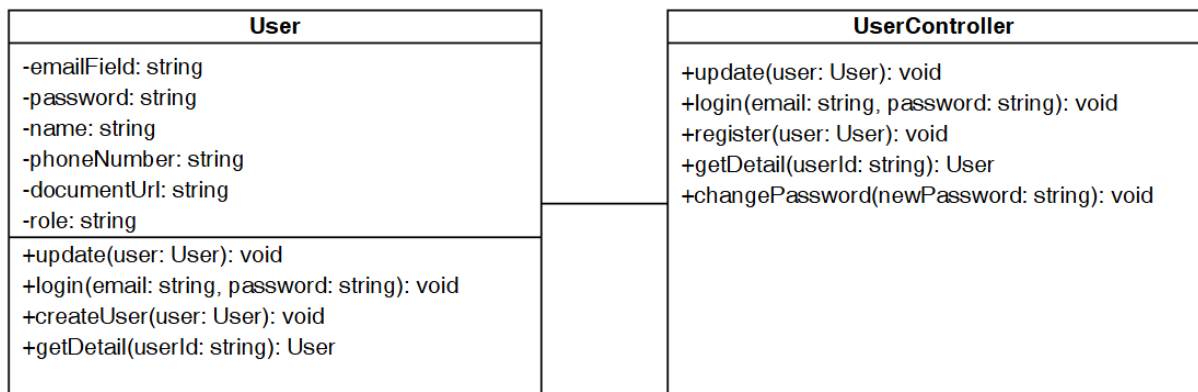


**Hình 9. Minh hoạ giao diện quản lý khoá học**



**Hình 10. Minh hoạ giao diện thêm câu hỏi vào ngân hàng đề**

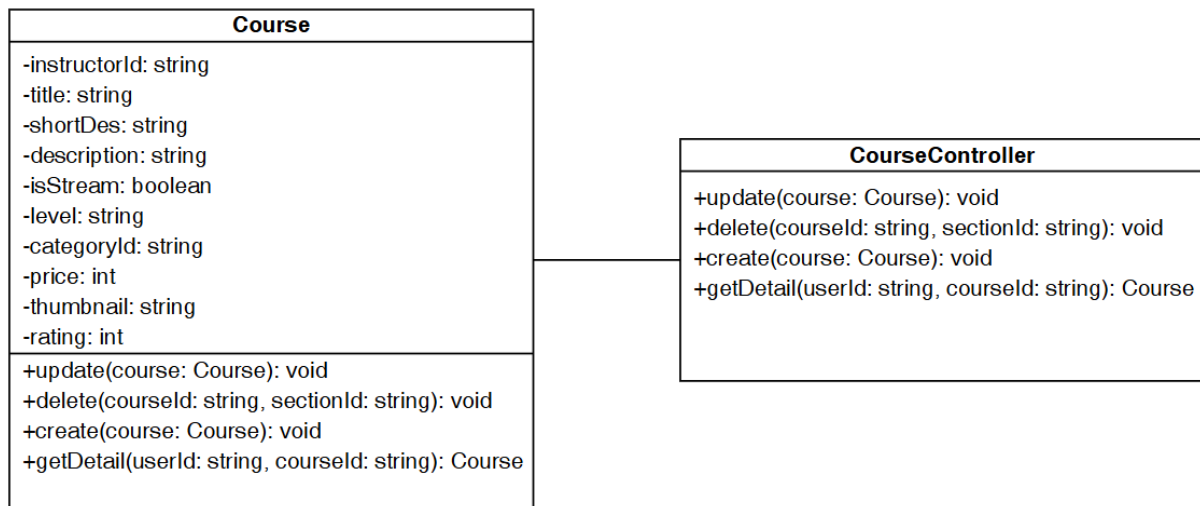
### 3.2. Thiết kế lớp



**Hình 11. Hình ảnh thiết kế lớp User**

Hình 5.11 là thiết kế chi tiết cho lớp User và UserController bao gồm các thuộc tính và các phương thức được sử dụng trong lớp này. Trong đó lớp UserController chứa các phương thức để xác thực tài khoản người dùng, đăng ký tài khoản, lấy ra thông tin chi tiết của người dùng, thay đổi mật khẩu của người dùng và cuối cùng là cập nhập thông tin.

Lớp User là lớp mô hình hóa thực thể người dùng hệ thống, lớp này có những thuộc tính mô tả các thông tin cơ bản của một người dùng bao gồm email, password và phone number. Các trường khác như role để lưu trữ phân quyền người dùng và documentUrl để lưu trữ minh chứng bằng cấp giáo viên nếu người dùng là giáo viên.



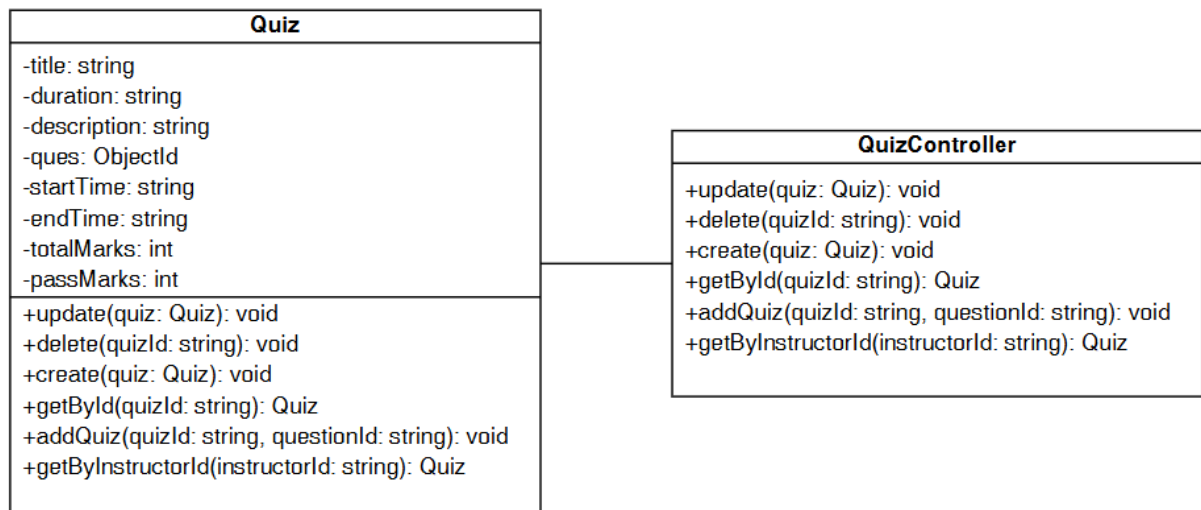
**Hình 12. Hình ảnh thiết kế lớp Course**

Hình 12 là thiết kế chi tiết cho lớp Course và CourseController bao gồm các thuộc tính và các phương thức được sử dụng trong lớp này. Trong đó lớp CourseController chứa các phương thức để quản lý khóa học, bao gồm tạo khóa học mới, lấy thông tin chi tiết của khóa học, cập nhật thông tin khóa học, và xóa khóa học. Lớp này cũng hỗ trợ việc thêm và xóa các phần học (section) trong khóa học.

Lớp Course là lớp mô hình hóa thực thể khóa học trong hệ thống, lớp này có những thuộc tính mô tả các thông tin cơ bản của một khóa học bao gồm:

- instructorId: ID của người hướng dẫn (giáo viên) khóa học, liên kết tới mô hình User.
- title: Tiêu đề của khóa học, phải là duy nhất và bắt buộc.
- shortDes: Mô tả ngắn gọn về khóa học.
- description: Mô tả chi tiết về khóa học, bắt buộc phải có.
- isStream: Cờ xác định khóa học có phải là khóa học trực tuyến hay không.
- categoryId: ID của danh mục khóa học, liên kết tới mô hình Category.
- level: Mức độ khó của khóa học, với các giá trị có thể là 'basic', 'intermediate', 'advanced', 'specialized', mặc định là 'basic'.
- price: Giá của khóa học, mặc định là 0.
- courseVideo: URL của video khóa học (không bắt buộc).
- tags: Danh sách các thẻ liên quan đến khóa học.
- sections: Danh sách các ID của các phần học (section) thuộc khóa học, liên kết tới mô hình Section.
- thumbnail: URL của hình thu nhỏ của khóa học, bắt buộc phải có.
- rating: Đánh giá của khóa học, giá trị nằm trong khoảng từ 0 đến 5, mặc định là 0.
- date\_created: Ngày tạo khóa học.
- date\_updated: Ngày cập nhật thông tin khóa học.



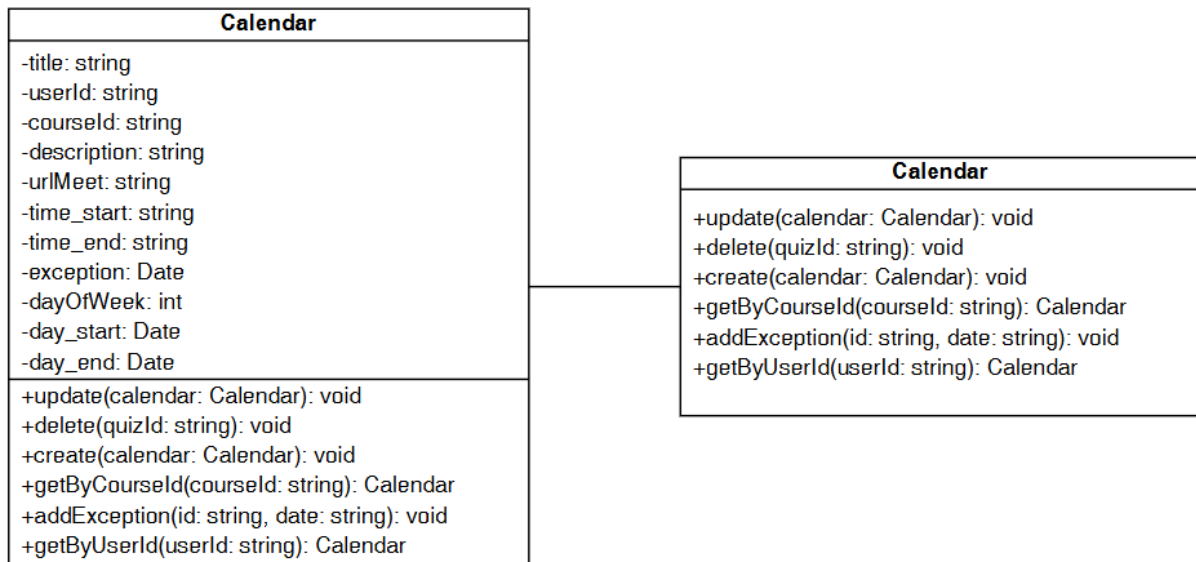


**Hình 13. Hình ảnh thiết kế lớp Quiz**

Hình 13 là thiết kế chi tiết cho lớp Quiz và QuizController bao gồm các thuộc tính và các phương thức được sử dụng trong lớp này. Trong đó lớp QuizController chứa các phương thức để quản lý bài kiểm tra (quiz), bao gồm tạo bài kiểm tra mới, lấy thông tin chi tiết của bài kiểm tra, cập nhật thông tin bài kiểm tra, và xóa bài kiểm tra. Lớp này cũng hỗ trợ việc thêm và xóa các câu hỏi (question) trong bài kiểm tra.

Lớp Quiz là lớp mô hình hóa thực thể bài kiểm tra trong hệ thống, lớp này có những thuộc tính mô tả các thông tin cơ bản của một bài kiểm tra bao gồm:

- title: Tiêu đề của bài kiểm tra, bắt buộc phải có.
- duration: Thời lượng của bài kiểm tra.
- ques: Danh sách các ID của các câu hỏi thuộc bài kiểm tra, liên kết tới mô hình Question.
- startTime: Thời gian bắt đầu của bài kiểm tra.
- endTime: Thời gian kết thúc của bài kiểm tra.
- totalMarks: Tổng số điểm của bài kiểm tra, mặc định là 10.
- passMarks: Số điểm cần thiết để vượt qua bài kiểm tra, mặc định là 10.
- date\_created: Ngày tạo bài kiểm tra.
- date\_updated: Ngày cập nhật thông tin bài kiểm tra.

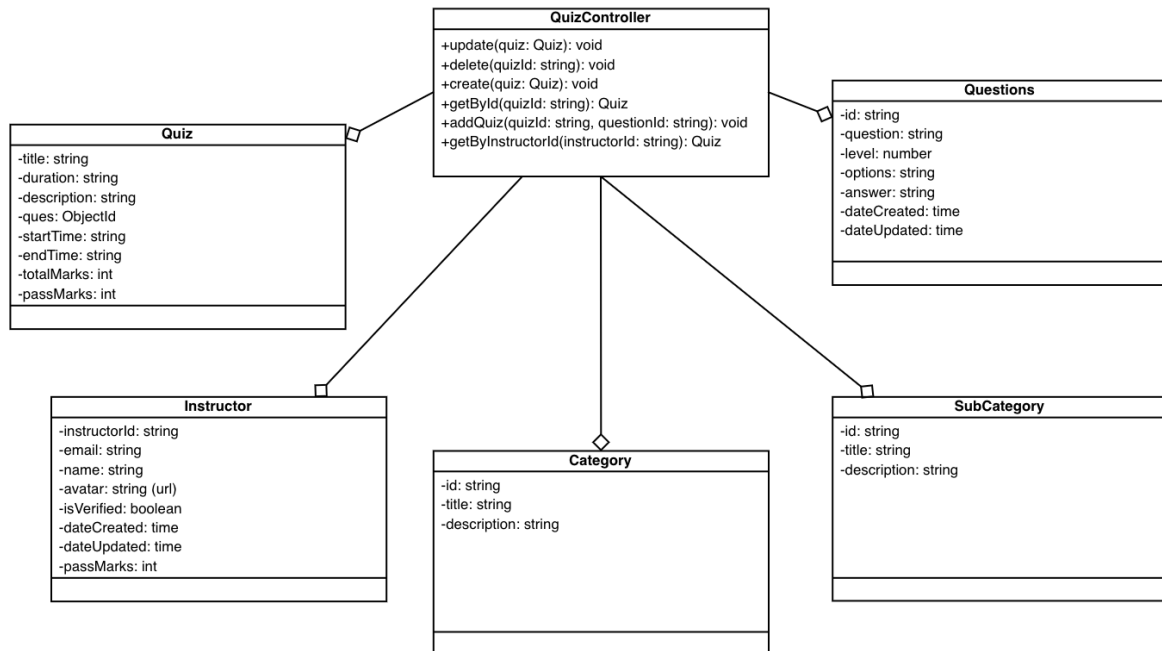


**Hình 14. Hình ảnh thiết kế lớp Calendar**

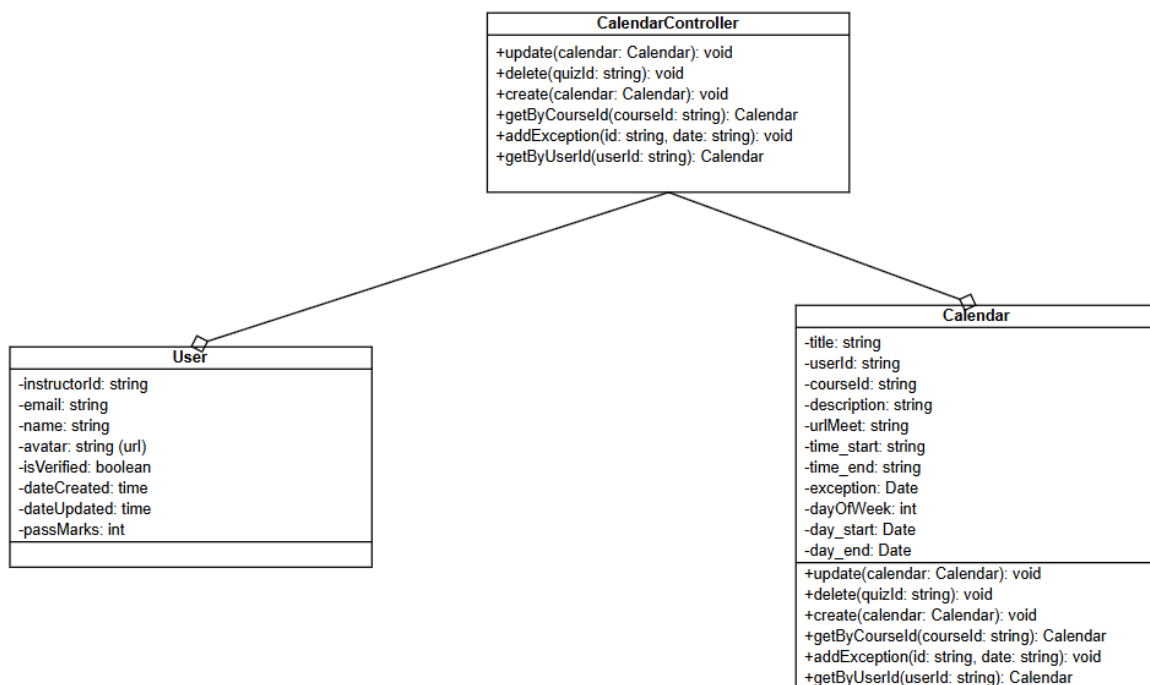
Hình 14 là thiết kế chi tiết cho lớp Calendar và CalendarController bao gồm các thuộc tính và các phương thức được sử dụng trong lớp này. Trong đó lớp CalendarController chứa các phương thức để quản lý lịch học, bao gồm tạo lịch học mới, lấy thông tin chi tiết của lịch học, cập nhật thông tin lịch học, và xóa lịch học. Lớp này cũng hỗ trợ việc thêm và xóa các ngày ngoại lệ trong lịch học.

Lớp Calendar là lớp mô hình hóa thực thể lịch học trong hệ thống, lớp này có những thuộc tính mô tả các thông tin cơ bản của một lịch học bao gồm:

- **userId:** ID của người dùng (giảng viên) tạo lịch học, liên kết tới mô hình User.
- **courseId:** ID của khóa học liên quan đến lịch học, liên kết tới mô hình Course.
- **title:** Tiêu đề của lịch học.
- **description:** Mô tả chi tiết của lịch học.
- **urlMeet:** URL của buổi học trực tuyến.
- **dayOfWeek:** Ngày trong tuần diễn ra buổi học (0: Chủ Nhật, 6: Thứ Bảy).
- **time\_start:** Thời gian bắt đầu của buổi học (định dạng HH:mm cho lịch học định kỳ).
- **time\_end:** Thời gian kết thúc của buổi học (định dạng HH:mm cho lịch học định kỳ).
- **day\_start:** Ngày bắt đầu của lịch học.
- **day\_end:** Ngày kết thúc của lịch học.
- **exceptions:** Danh sách các ngày ngoại lệ không có buổi học.
- **date\_created:** Ngày tạo lịch học.
- **date\_updated:** Ngày cập nhật thông tin lịch học.

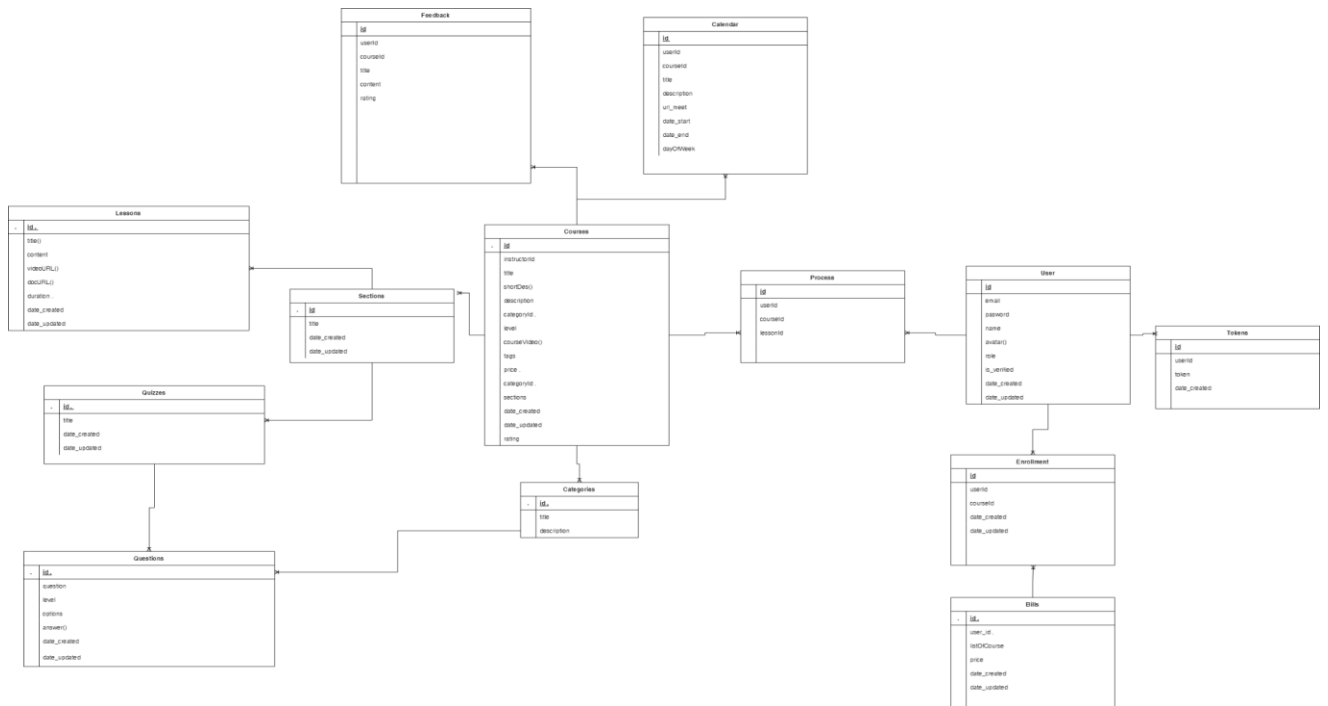


**Hình 15. Thiết kế lớp chi tiết cho chức năng tạo bài tập**



**Hình 16. Thiết kế lớp chi tiết cho chức năng tạo lịch học**

### 3.3. Thiết kế cơ sở dữ liệu



**Hình 16. Hình ảnh cơ sở dữ liệu**

Hình 15 thể hiện các thực thể và thuộc tính:

- **User (Người Dùng)**
  - id: Mã định danh duy nhất cho mỗi người dùng.
  - email: Địa chỉ email của người dùng (Duy nhất).
  - password: Mật khẩu để xác thực người dùng.
  - name: Tên đầy đủ của người dùng.
  - avatar: URL đến hình ảnh đại diện của người dùng.
  - role: Vai trò của người dùng (ví dụ: học sinh, giáo viên, quản trị viên).
  - is\_verified: Trạng thái xác minh email của người dùng.
  - date\_created: Thời gian tạo người dùng.
  - date\_updated: Thời gian cập nhật thông tin người dùng lần cuối.
- **Course (Khóa Học)**
  - id: Mã định danh duy nhất cho mỗi khóa học.
  - instructorId: Tham chiếu đến người dùng tạo khóa học.
  - title: Tiêu đề của khóa học.
  - shortDesc: Mô tả ngắn gọn của khóa học.
  - description: Mô tả chi tiết của khóa học.
  - categoryId: Tham chiếu đến danh mục của khóa học.
  - level: Cấp độ khó của khóa học (ví dụ: cơ bản, trung cấp, nâng cao).
  - courseVideo: URL đến video giới thiệu khóa học.

- tags: Các thẻ liên quan đến khóa học để dễ dàng tìm kiếm.
- price: Giá của khóa học.
- sections: Danh sách các phần trong khóa học.
- created\_at: Thời gian tạo khóa học.
- updated\_at: Thời gian cập nhật thông tin khóa học lần cuối.
- rating: Đánh giá trung bình của khóa học.
- Section (Phần)
  - id: Mã định danh duy nhất cho mỗi phần.
  - title: Tiêu đề của phần.
  - date\_created: Thời gian tạo phần.
  - date\_updated: Thời gian cập nhật thông tin phần lần cuối.
  - courseId: Tham chiếu đến khóa học mà phần này thuộc về.
- Lesson (Bài Học)
  - id: Mã định danh duy nhất cho mỗi bài học.
  - title: Tiêu đề của bài học.
  - content: Nội dung của bài học.
  - videoURL: URL đến video bài học.
  - docURL: URL đến tài liệu bài học.
  - duration: Thời lượng của bài học.
  - date\_created: Thời gian tạo bài học.
  - date\_updated: Thời gian cập nhật thông tin bài học lần cuối.
  - sectionId: Tham chiếu đến phần mà bài học thuộc về.
- Quiz (Bài Kiểm Tra)
  - id: Mã định danh duy nhất cho mỗi bài kiểm tra.
  - title: Tiêu đề của bài kiểm tra.
  - date\_created: Thời gian tạo bài kiểm tra.
  - date\_updated: Thời gian cập nhật thông tin bài kiểm tra lần cuối.
  - sectionId: Tham chiếu đến phần mà bài kiểm tra thuộc về.
- Question (Câu Hỏi)
  - id: Mã định danh duy nhất cho mỗi câu hỏi.
  - question: Nội dung của câu hỏi.
  - level: Cấp độ khó của câu hỏi.
  - options: Các tùy chọn trả lời cho câu hỏi.
  - answer: Đáp án đúng của câu hỏi.
  - date\_created: Thời gian tạo câu hỏi.
  - date\_updated: Thời gian cập nhật thông tin câu hỏi lần cuối.
  - quizId: Tham chiếu đến bài kiểm tra mà câu hỏi thuộc về.
- Category (Danh Mục)
  - id: Mã định danh duy nhất cho mỗi danh mục.
  - title: Tiêu đề của danh mục.

- description: Mô tả của danh mục.
- Process (Quá Trình)
  - id: Mã định danh duy nhất cho mỗi quá trình.
  - userId: Tham chiếu đến người dùng tham gia quá trình.
  - courseId: Tham chiếu đến khóa học liên quan đến quá trình.
  - sectionId: Tham chiếu đến phần liên quan đến quá trình.
- Enrollment (Ghi Danh)
  - id: Mã định danh duy nhất cho mỗi ghi danh.
  - userId: Tham chiếu đến người dùng đã ghi danh.
  - courseId: Tham chiếu đến khóa học mà người dùng đã ghi danh.
  - date\_created: Thời gian tạo ghi danh.
  - date\_updated: Thời gian cập nhật thông tin ghi danh lần cuối.
- Calendar (Lịch)
  - id: Mã định danh duy nhất cho mỗi mục lịch.
  - userId: Tham chiếu đến người dùng liên quan đến mục lịch.
  - courseId: Tham chiếu đến khóa học liên quan đến mục lịch.
  - title: Tiêu đề của mục lịch.
  - description: Mô tả của mục lịch.
  - url\_meet: URL của buổi họp.
  - date\_start: Ngày và giờ bắt đầu của mục lịch.
  - date\_end: Ngày và giờ kết thúc của mục lịch.
  - dayOfWeek: Ngày trong tuần cho các sự kiện định kỳ.
- Feedback (Phản Hồi)
  - id: Mã định danh duy nhất cho mỗi phản hồi.
  - userId: Tham chiếu đến người dùng đã đưa ra phản hồi.
  - courseId: Tham chiếu đến khóa học mà phản hồi liên quan.
  - title: Tiêu đề của phản hồi.
  - content: Nội dung của phản hồi.
  - rating: Đánh giá trong phản hồi.
- Token
  - id: Mã định danh duy nhất cho mỗi token.
  - userId: Tham chiếu đến người dùng liên quan đến token.
  - token: Giá trị của token.
  - date\_created: Thời gian tạo token.
- Bill (Hóa Đơn)
  - id: Mã định danh duy nhất cho mỗi hóa đơn.
  - user\_id: Tham chiếu đến người dùng liên quan đến hóa đơn.
  - ratioCourse: Tỷ lệ học phí.
  - price: Giá của hóa đơn.
  - date\_created: Thời gian tạo hóa đơn.

- date\_updated: Thời gian cập nhật thông tin hóa đơn lần cuối.

Mối quan hệ:

- User-Course (Một-Nhiều)
  - Một người dùng (giáo viên) có thể tạo nhiều khóa học.
- Course-Section (Một-Nhiều)
  - Một khóa học có thể có nhiều phần.
- Section-Lesson (Một-Nhiều)
  - Một phần có thể có nhiều bài học.
- Section-Quiz (Một-Nhiều)
  - Một phần có thể có nhiều bài kiểm tra.
- Quiz-Question (Một-Nhiều)
  - Một bài kiểm tra có thể có nhiều câu hỏi.
- Course-Category (Nhiều-Một)
  - Nhiều khóa học có thể thuộc về một danh mục.
- User-Process (Một-Nhiều)
  - Một người dùng có thể có nhiều quá trình.
- Course-Process (Một-Nhiều)
  - Một khóa học có thể có nhiều quá trình.
- User-Enrollment (Một-Nhiều)
  - Một người dùng có thể có nhiều ghi danh.
- Course-Enrollment (Một-Nhiều)
  - Một khóa học có thể có nhiều ghi danh.
- User-Calendar (Một-Nhiều)
  - Một người dùng có thể có nhiều mục lịch.
- Course-Calendar (Một-Nhiều)
  - Một khóa học có thể có nhiều mục lịch.
- User-Feedback (Một-Nhiều)
  - Một người dùng có thể đưa ra nhiều phản hồi.
- Course-Feedback (Một-Nhiều)
  - Một khóa học có thể có nhiều phản hồi.
- User-Token (Một-Nhiều)
  - Một người dùng có thể có nhiều token.
- User-Bill (Một-Nhiều)
  - Một người dùng có thể có nhiều hóa đơn.

#### 4. Kiểm thử chức năng

Để kiểm tra việc hệ thống hoạt động đúng đắn có như mong muốn theo thiết kế ban đầu hay không, chúng ta cần phải kiểm thử các chức năng của hệ thống được phát triển. Mục tiêu cần kiểm tra xem các chức năng có hoạt động đúng hay không, có gặp

phải lỗi trong quá trình vận hành hay không. Các kết quả hiển thị từ hệ thống có đúng như mong muốn hay không, các thông báo lỗi có chuẩn xác không. Một số chức năng quan trọng được kiểm thử sẽ được nêu ra trong các phần tiếp sau đây.

#### 4.1. Kiểm thử chức năng xem thông tin chi tiết khóa học

Trường hợp 1: Người dùng chưa đăng nhập cố gắng truy cập trang chi tiết khóa học. Kết quả mong đợi: Hệ thống chuyển hướng người dùng đến trang đăng nhập.

Trường hợp 2: Người dùng đăng nhập truy cập trang chi tiết khóa học. Kết quả mong đợi: Hệ thống hiển thị thông tin chi tiết của khóa học (tên, mô tả, nội dung, giảng viên, đánh giá, giá).

Trường hợp 3: Khóa học không tồn tại. Kết quả mong đợi: Hệ thống hiển thị thông báo lỗi "Khóa học không tồn tại".

Trường hợp 4: Khóa học có thông tin không đầy đủ. Kết quả mong đợi: Hệ thống hiển thị thông tin có sẵn và thông báo rằng một số thông tin có thể thiếu.

#### 4.2. Kiểm thử chức năng đăng ký khóa học

Trường hợp 1: Người dùng không đăng nhập cố gắng đăng ký khóa học. Kết quả mong đợi: Hệ thống chuyển hướng người dùng đến trang đăng nhập.

Trường hợp 2: Người dùng đăng nhập đăng ký khóa học thành công. Kết quả mong đợi: Hệ thống xác nhận đăng ký thành công và cập nhật thông tin khóa học trong tài khoản của người dùng.

Trường hợp 3: Người dùng đã đăng ký khóa học này cố gắng đăng ký lại. Kết quả mong đợi: Hệ thống thông báo "Bạn đã đăng ký khóa học này rồi".

Trường hợp 4: Khóa học yêu cầu thanh toán và người dùng thực hiện thanh toán thành công. Kết quả mong đợi: Hệ thống xác nhận thanh toán và đăng ký khóa học thành công.

Trường hợp 5: Khóa học yêu cầu thanh toán và người dùng hủy thanh toán. Kết quả mong đợi: Hệ thống hủy bỏ quá trình đăng ký và thông báo "Thanh toán bị hủy".

#### 4.3. Kiểm thử chức năng gợi ý khóa học

Trường hợp 1: Người dùng không đăng nhập truy cập trang gợi ý khóa học. Kết quả mong đợi: Hệ thống hiển thị các khóa học phổ biến.

Trường hợp 2: Người dùng đăng nhập với lịch sử học tập trống. Kết quả mong đợi: Hệ thống hiển thị các khóa học phổ biến hoặc mới nhất.

Trường hợp 3: Người dùng đăng nhập với lịch sử học tập cụ thể. Kết quả mong đợi: Hệ thống gợi ý các khóa học liên quan đến các khóa học đã tham gia.

Trường hợp 4: Người dùng tìm kiếm khóa học bằng từ khóa. Kết quả mong đợi: Hệ thống hiển thị danh sách các khóa học phù hợp với từ khóa tìm kiếm.

#### 4.4. Kiểm thử chức năng tạo khóa học mới



Trường hợp 1: Người dùng không đăng nhập cố gắng tạo khóa học mới. Kết quả mong đợi: Hệ thống chuyển hướng người dùng đến trang đăng nhập.

Trường hợp 2: Người dùng đăng nhập không có quyền giảng viên cố gắng tạo khóa học mới. Kết quả mong đợi: Hệ thống hiển thị thông báo "Bạn không có quyền tạo khóa học".

Trường hợp 3: Người dùng đăng nhập với quyền giảng viên và điền đầy đủ thông tin khóa học. Kết quả mong đợi: Hệ thống lưu khóa học mới và thông báo "Khóa học đã được tạo thành công".

Trường hợp 4: Người dùng đăng nhập với quyền giảng viên nhưng bỏ trống một số thông tin bắt buộc. Kết quả mong đợi: Hệ thống hiển thị thông báo yêu cầu điền đầy đủ thông tin.

#### 4.5. Kiểm thử chức năng tạo bài tập thủ công

Trường hợp 1: Giảng viên không đăng nhập cố gắng tạo bài tập. Kết quả mong đợi: Hệ thống chuyển hướng người dùng đến trang đăng nhập.

Trường hợp 2: Giảng viên đăng nhập và chọn khóa học để tạo bài tập thủ công. Kết quả mong đợi: Hệ thống hiển thị giao diện tạo bài tập.

Trường hợp 3: Giảng viên điền đầy đủ thông tin bài tập và lưu. Kết quả mong đợi: Hệ thống lưu bài tập và thông báo "Bài tập đã được tạo thành công".

Trường hợp 4: Giảng viên bỏ trống một số thông tin bắt buộc và cố gắng lưu bài tập. Kết quả mong đợi: Hệ thống hiển thị thông báo yêu cầu điền đầy đủ thông tin.

#### 4.6. Kiểm thử chức năng tạo bài tập tự động

Trường hợp 1: Giảng viên không đăng nhập cố gắng tạo bài tập. Kết quả mong đợi: Hệ thống chuyển hướng người dùng đến trang đăng nhập.

Trường hợp 2: Giảng viên đăng nhập và chọn khóa học để tạo bài tập tự động. Kết quả mong đợi: Hệ thống hiển thị giao diện tạo bài tập.

Trường hợp 3: Giảng viên cấu hình bài tập tự động và lưu. Kết quả mong đợi: Hệ thống lưu bài tập và thông báo "Bài tập đã được tạo thành công".

Trường hợp 4: Giảng viên bỏ trống một số thông tin bắt buộc và cố gắng lưu bài tập. Kết quả mong đợi: Hệ thống hiển thị thông báo yêu cầu điền đầy đủ thông tin.

Trường hợp 5: Giảng viên cấu hình không hợp lệ cho bài tập tự động. Kết quả mong đợi: Hệ thống hiển thị thông báo lỗi cấu hình không hợp lệ.

#### 4.7. Kiểm thử chức năng duyệt giáo viên

Trường hợp 1: Người dùng không đăng nhập cố gắng truy cập trang duyệt giáo viên. Kết quả mong đợi: Hệ thống chuyển hướng người dùng đến trang đăng nhập.

Trường hợp 2: Người dùng đăng nhập không có quyền quản trị cố gắng truy cập trang duyệt giáo viên. Kết quả mong đợi: Hệ thống hiển thị thông báo "Bạn không có quyền truy cập trang này."

Trường hợp 3: Quản trị viên đăng nhập và truy cập trang duyệt giáo viên. Kết quả mong đợi: Hệ thống hiển thị danh sách các giáo viên chờ duyệt và trạng thái hiện tại của họ.

Trường hợp 4: Quản trị viên duyệt giáo viên thành công. Kết quả mong đợi: Hệ thống cập nhật trạng thái của giáo viên thành "Đã duyệt" và thông báo "Giáo viên đã được duyệt thành công."

## CHƯƠNG 5: CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

### 1. Tổng quan

Giải thuật đề xuất khóa học là một phần quan trọng trong các hệ thống học trực tuyến hiện đại. Nó giúp cung cấp cho người dùng những khóa học phù hợp dựa trên sở thích và hành vi của họ cũng như của người dùng khác. Điều này giúp nâng cao trải nghiệm học tập và giữ chân người dùng.

### 2. Giới thiệu vấn đề

Trong các hệ thống học trực tuyến như Udemy, việc cung cấp các đề xuất khóa học phù hợp là vô cùng quan trọng để giữ chân người dùng và tăng cường trải nghiệm học tập của họ. Vấn đề chính cần giải quyết là làm thế nào để tìm ra các khóa học phù hợp nhất với mỗi người dùng dựa trên dữ liệu về sở thích cá nhân và đánh giá khóa học từ cộng đồng người dùng.

### 3. Cách thức hoạt động của thuật toán

#### 3.1. Mô hình hoá dữ liệu người dùng và khóa học

Thuật toán sử dụng các mô hình chính để lưu trữ thông tin:

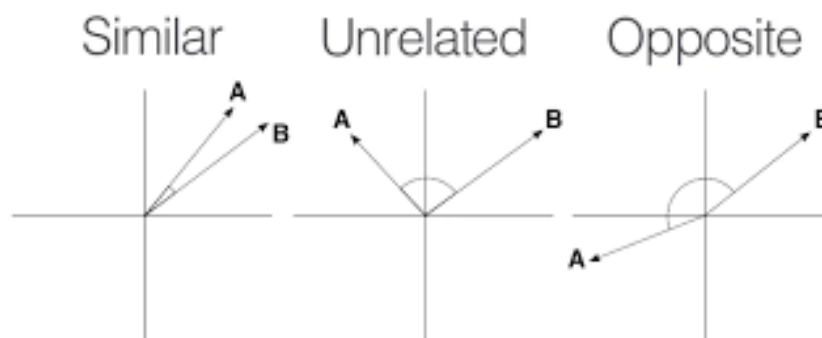
- **FavoriteModel:** Lưu trữ thông tin về các khóa học mà người dùng đã thêm vào danh sách yêu thích.
- **FeedbackModel:** Lưu trữ thông tin về đánh giá của người dùng đối với các khóa học.
- **CourseModel:** Lưu trữ thông tin chi tiết về các khóa học.

#### 3.2. Xây dựng vector đặc trưng khóa học

Sử dụng mô hình TF-IDF để xây dựng vector đặc trưng cho các khóa học dựa trên tiêu đề, mô tả và các thẻ liên quan của khóa học. Các vector này sẽ được sử dụng để tính toán độ tương đồng giữa các khóa học.

#### 3.3. Tính toán độ tương đồng giữa người dùng

Thuật toán sử dụng hàm **cosineSimilarity** để tính toán độ tương đồng giữa các người dùng dựa trên đánh giá của họ. Độ tương đồng cosine được sử dụng để đo lường sự tương đồng giữa hai vector đánh giá.



**Hình 3.1. Hình ảnh mô tả thuật toán cosine similarity**

Ý tưởng chính của độ tương đồng cosin là đo lường góc giữa hai vector trong không gian n-chiều. Nếu góc giữa hai vector nhỏ, giá trị cosin sẽ gần 1, cho thấy hai vector rất tương đồng. Nếu góc lớn, giá trị cosin sẽ gần 0, cho thấy hai vector ít tương đồng.

$$\text{similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

Trong đó:

- $A \cdot B$  là tích vô hướng của hai vector (dot product).
- $\|A\|$  và  $\|B\|$  là độ dài của hai vector (norm của vector).

#### 3.4. Đề xuất khoá học

Dựa trên độ tương đồng giữa các khóa học, thuật toán xác định các khóa học chưa được đánh giá bởi người dùng hiện tại nhưng có độ tương đồng cao với các khóa học mà người dùng đã yêu thích hoặc đánh giá cao. Các khóa học này được tính điểm đề xuất dựa trên mức độ tương đồng.

#### 3.5. Chuẩn hoá và trả về kết quả

Điểm đề xuất cho mỗi khóa học được chuẩn hóa bằng cách chia tổng điểm cho số lượng đóng góp để đạt được điểm trung bình. Kết quả cuối cùng là danh sách các khóa học được đề xuất cho người dùng.

### 4. Độ phức tạp của thuật toán

- **Xây dựng vector đặc trưng:**  $O(C)$  với  $C$  là số lượng khóa học.
- **Tính toán độ tương đồng cosin:**  $O(C^2)$  trong trường hợp xấu nhất (tính toán độ tương đồng giữa tất cả các khóa học).
- **Đề xuất khóa học:**  $O(C * U)$  với  $C$  là số lượng khóa học và  $U$  là số lượng người dùng.

5. Ví dụ minh hoạ
- Dữ liệu đầu vào:
    - Người dùng: U1, U2, U3, U4, U5
    - Khóa học: Course 1, Course 2, Course 3, Course 4.

Khoá học	Tiêu đề	Mô tả	Thẻ
Course1	"Python Basics"	"Learn Python"	["programming", "python"]
Course2	"Advanced Python"	"Deep dive into Python"	["programming", "python"]
Course3	"Data Science"	"Introduction to Data Science"	["data", "science"]
Course4	"Machine Learning"	"Basics of ML"	["machine learning", "ai"]

Hình 5.1 Bảng dữ liệu đầu vào

Chú thích:

- Course1: Khóa học về các khái niệm cơ bản của Python.
- Course2: Khóa học nâng cao về Python.
- Course3: Khóa học nhập môn về Khoa học dữ liệu.
- Course4: Khóa học về các khái niệm cơ bản của Máy học.

Khoá học	Vector đặc trưng
Course1	[0.5, 0.5, 0.5, 0, 0]
Course2	[0.6, 0.6, 0.6, 0, 0]
Course3	[0, 0, 0, 0.7, 0.7]
Course4	[0, 0, 0, 0.8, 0.8]

Hình 5.2 Hình ảnh vector đặc trưng cho các khoá học

Chú thích:

- Course1 Vector: Vector đặc trưng cho khóa học Python Basics, với các thành phần thể hiện sự xuất hiện của các từ khóa liên quan đến "programming" và "python".

- Course2 Vector: Vector đặc trưng cho khóa học Advanced Python, với các thành phần tương tự Course1 nhưng với giá trị cao hơn do mức độ chuyên sâu hơn.
- Course3 Vector: Vector đặc trưng cho khóa học Data Science, với các thành phần thể hiện sự xuất hiện của các từ khóa liên quan đến "data" và "science".
- Course4 Vector: Vector đặc trưng cho khóa học Machine Learning, với các thành phần thể hiện sự xuất hiện của các từ khóa liên quan đến "machine learning" và "ai".

**Cách tính vector đặc trưng. Giả sử ta có thông tin của người dùng như sau:**

Thông tin đánh giá của người dùng:

- U1: [("Course1", 5), ("Course3", 4)]
- U2: [("Course2", 5), ("Course4", 4)]

Thông tin yêu thích của người dùng:

- U1: ["Course1", "Course3"]
- U2: ["Course2", "Course4"]

**Xây dựng ma trận từ ngữ (Word Matrix)**

Tạo từ điển từ ngữ (token) từ mô tả khóa học:

- Tokens: ["introduction", "to", "programming", "advanced", "java", "web", "development", "javascript", "machine", "learning", "basics", "data", "science", "python"]

**Tính toán TF-IDF**

IDF của một từ là log của tổng số tài liệu chia cho số tài liệu chứa từ đó:

$$IDF(t) = \log\left(\frac{N}{n_t}\right)$$

trong đó:

- N là tổng số tài liệu (Bản ghi).
- $n_t$  là số tài liệu chứa từ t.

TF-IDF là tích của TF và IDF:  $TF-IDF(t,d)=TF(t,d)\times IDF(t)$

**Course 1: ["introduction", "programming", "beginner"]**

- Tokens trong bản ghi 1: ["introduction", "programming", "beginner"]

- TF của mỗi từ:
  - "introduction":  $1/3 \approx 0.333$
  - "programming":  $1/3 \approx 0.333$
  - "beginner":  $1/3 \approx 0.333$
- IDF của mỗi từ (giả sử mỗi từ xuất hiện trong 1 tài liệu khác nhau):
  - "introduction":  $\log_{f_0}(5/1) = \log_{f_0}(5) \approx 0.699$
  - "programming":  $\log_{f_0}(5/2) = \log_{f_0}(2.5) \approx 0.398$  (giả sử "programming" xuất hiện trong 2 tài liệu)
  - "beginner":  $\log_{f_0}(5/2) = \log_{f_0}(2.5) \approx 0.398$
- TF-IDF của mỗi từ trong Course1:
  - "introduction":  $0.333 \times 0.699 \approx 0.233$
  - "programming":  $0.333 \times 0.398 \approx 0.133$
  - "beginner":  $0.333 \times 0.398 \approx 0.133$

### Course 2: ["advanced", "java", "programming"]

- Tokens trong Course2: ["advanced", "java", "programming"]
- TF của mỗi từ:
  - "advanced":  $1/3 \approx 0.333$
  - "java":  $1/3 \approx 0.333$
  - "programming":  $1/3 \approx 0.333$
- IDF của mỗi từ (giả sử mỗi từ xuất hiện trong 1 tài liệu khác nhau):
  - "advanced":  $\log_{f_0}(5/1) = \log_{f_0}(5) \approx 0.699$
  - "java":  $\log_{f_0}(5/1) = \log_{f_0}(5) \approx 0.699$
  - "programming":  $\log_{f_0}(5/2) = \log_{f_0}(2.5) \approx 0.398$
- TF-IDF của mỗi từ trong Course2:
  - "advanced":  $0.333 \times 0.699 \approx 0.233$
  - "java":  $0.333 \times 0.699 \approx 0.233$
  - "programming":  $0.333 \times 0.398 \approx 0.133$

### Kết quả vector đặc trưng:

- Bản ghi 1: [0.233, 0.133, 0.133, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
- Bản ghi 2: [0, 0, 0.133, 0.233, 0.233, 0, 0, 0, 0, 0, 0, 0, 0, 0]
- Bản ghi 3: [0, 0, 0, 0, 0, 0.333, 0.333, 0.333, 0, 0, 0, 0, 0, 0]
- Bản ghi 4: [0, 0, 0, 0, 0, 0, 0, 0, 0.5, 0.5, 0.5, 0, 0, 0]
- Bản ghi 5: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.333, 0.333, 0.333]

### Tính toán độ tương đồng cosine giữa Course1 và các khoá học khác.

- **Cosine(Course1, Course2)** =  $(0.5 * 0.6 + 0.5 * 0.6 + 0.5 * 0.6) / (\sqrt{0.5^2 + 0.5^2 + 0.5^2} * \sqrt{0.6^2 + 0.6^2 + 0.6^2}) = 0.99$
- **Cosine(Course1, Course3)** =  $(0.5 * 0 + 0.5 * 0 + 0.5 * 0) / (\sqrt{0.5^2 + 0.5^2 + 0.5^2} * \sqrt{0.7^2 + 0.7^2 + 0.7^2}) = 0$
- **Cosine(Course1, Course4)** =  $(0.5 * 0 + 0.5 * 0 + 0.5 * 0) / (\sqrt{0.5^2 + 0.5^2 + 0.5^2} * \sqrt{0.8^2 + 0.8^2 + 0.8^2}) = 0$

Tổng hợp kết quả dựa trên tính toán cosine similarity, chúng ta có độ tương đồng giữa Course 1 và các khoá học khác như sau:

- **Cosine(Course1, Course2)**: Độ tương đồng giữa Course1 và Course2 là rất cao (0.99), cho thấy hai khóa học này có nội dung rất giống nhau.
- **Cosine(Course1, Course3)**: Độ tương đồng giữa Course1 và Course3 là 0, cho thấy hai khóa học này không có nội dung liên quan.
- **Cosine(Course1, Course4)**: Độ tương đồng giữa Course1 và Course4 cũng là 0, cho thấy hai khóa học này không có nội dung liên quan.

Dựa trên các giá trị này, giả sử người dùng U1 đã yêu thích Course1, thuật toán sẽ đề xuất Course2 cho U1 vì độ tương đồng giữa Course1 và Course2 là cao nhất (0.99).

FETCH user data (userId)

GET favorites FROM favoriteModel

GET feedbacks FROM feedbackModel

RETURN {favorites, feedbacks}

FETCH course data

GET courses FROM courseModel

RETURN courses

BUILD feature vectors (courses)

INITIALIZE tokenizer

INITIALIZE tfidf



```
FOR EACH course IN courses
```

```
    text = course.title + ' ' + course.description + ' ' + course.tags.join(' ')
```

```
    ADD document TO tfidf
```

```
END FOR
```

```
courseVectors = []
```

```
FOR EACH course IN courses
```

```
    text = course.title + ' ' + course.description + ' ' + course.tags.join(' ')
```

```
    vector = []
```

```
    tfidf.tfidf(text, (i, measure) => vector.push(measure))
```

```
    courseVectors.push({ course, vector })
```

```
END FOR
```

```
RETURN courseVectors
```

```
CALCULATE cosine similarity (vector1, vector2)
```

```
dotProduct = 0
```

```
normVector1 = 0
```

```
normVector2 = 0
```

```
FOR i FROM 0 TO LENGTH(vector1)
```

```
    dotProduct += vector1[i] * vector2[i]
```

```
    normVector1 += vector1[i] ** 2
```

```
    normVector2 += vector2[i] ** 2
```

```
END FOR
```

```
normVector1 = sqrt(normVector1)
```

```
normVector2 = sqrt(normVector2)
```

```
IF normVector1 > 0 AND normVector2 > 0
```

```

        RETURN dotProduct / (normVector1 * normVector2)

ELSE

    RETURN 0

RECOMMEND courses (userId)

    FETCH {favorites, feedbacks} FOR userId

    FETCH courses

    courseVectors = BUILD feature vectors (courses)

    userCourses = favorites.map(fav => fav.courseId) + feedbacks.map(fb =>
fb.courseId)

    userCourseVectors = FILTER courseVectors WHERE course._id IN
userCourses

    recommendedCourses = { }

    FOR EACH cv IN courseVectors

        IF cv.course._id NOT IN userCourses

            totalSimilarity = 0

            FOR EACH ucv IN userCourseVectors

                totalSimilarity += CALCULATE cosine similarity
(cv.vector, ucv.vector)

            END FOR

            averageSimilarity = totalSimilarity /
LENGTH(userCourseVectors)

            IF averageSimilarity > 0

                recommendedCourses[cv.course._id] = averageSimilarity

            END IF

        END IF

    END FOR

END FOR

```

RETURN recommendedCourses