

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Hệ thống học tập trực tuyến trên nền tảng Web

NGUYỄN QUANG NINH

ninh.nq198321@sis.hust.edu.vn

Ngành kĩ thuật phần mềm

Giảng viên hướng dẫn: TS. Trần Văn Đặng

Chữ ký GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 12/2024

# LỜI CẢM ƠN

Em muốn gửi lời cảm ơn tới gia đình, bạn bè, thầy cô, và chính bản thân mình vì đã chăm chỉ và quyết tâm thực hiện ĐATN để đạt kết quả tốt nhất.

Đầu tiên, em muốn gửi lời cảm ơn tới thầy Trần Văn Đặng, người đã dành thời gian để hướng dẫn, trao đổi và góp ý cũng như truyền đạt những kiến thức cần thiết cho đề tài em đang làm. Và chính những điều đó đã giúp em rất nhiều trong quá trình thực hiện đồ án này. Đồ án của em đã từng bước hoàn thiện sau mỗi lần trao đổi với thầy, các vấn đề dần được giải quyết một cách hiệu quả.

Ngoài ra, em cũng xin gửi lời cảm ơn đến gia đình cũng như bạn bè, những người đã luôn sẵn sàng chia sẻ kinh nghiệm, giúp đỡ em trong suốt quá trình thực hiện đồ án. Những sự hỗ trợ từ bạn bè, sự quan tâm từ gia đình tất cả những điều đó đã giúp em rất nhiều để được kết quả này.

Một lần nữa, em xin gửi lời cảm ơn chân thành nhất đến tất cả mọi người đã đồng hành và giúp đỡ em trong suốt quá trình làm đồ án này. Em xin ghi nhớ những tình cảm này trong quá trình làm cũng như tại thời điểm viết báo cáo này và xa hơn là trên con đường học tập và làm việc và phát triển hơn trong tương lai.

# TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong bối cảnh giáo dục hiện đại, nhu cầu học tập trực tuyến ngày càng tăng cao. Tuy nhiên, nhiều nền tảng hiện tại vẫn chưa đáp ứng đầy đủ các yêu cầu về tính tương tác, quản lý học tập và hỗ trợ người dùng.

Đồ án này nhằm giải quyết các hạn chế đó bằng cách phát triển một website học tập trực tuyến hoàn thiện hơn. Phát triển một nền tảng học tập trực tuyến với các tính năng như quản lý khóa học, bài giảng, bài tập trực tuyến. Với mong muốn tạo ra một môi trường học tập linh hoạt, tiện lợi và hiệu quả cho người dùng.Nền tảng bao gồm việc xây dựng một hệ thống quản lý học tập với các chức năng chính như: Đăng ký và quản lý tài khoản người dùng để người dùng có thể đăng ký tài khoản, đăng nhập và quản lý thông tin cá nhân. Tạo và quản lý khóa học, bài giảng để giáo viên có thể tạo các khóa học, bài giảng và tài liệu học tập.Người học có thể truy cập và tham gia các khóa học này. Hỗ trợ tương tác, trao đổi, thảo luận giữa giáo viên và người học. Quản lý bài tập trực tuyến với chức năng này thì giáo viên có thể tạo và quản lý các bài tập trực tuyến và người học có thể làm bài và nộp bài trực tuyến. Theo dõi tiến độ học tập và đánh giá kết quả thông qua các báo cáo, thống kê về tiến độ học tập và kết quả của người học.

Hệ thống học tập trực tuyến đã đưa ra được các chức năng cần thiết trong quá trình học tập và quản lí đồng thời giúp cải thiện trải nghiệm học tập của người dùng. Tăng cường tính tương tác và hỗ trợ người dùng: Hệ thống cung cấp các công cụ tương tác giữa giáo viên và người học. Hệ thống giúp giáo viên và người học quản lý các hoạt động học tập một cách dễ dàng và hiệu quả. Đáp ứng nhu cầu học tập trực tuyến trong bối cảnh hiện đại(trực tuyến) giúp người dùng có thể học tập mọi lúc, mọi nơi

Sinh viên thực hiện  
(Ký và ghi rõ họ tên)

## MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>1</b>
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án .....	2
<b>CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....</b>	<b>3</b>
2.1 Khảo sát hiện trạng .....	3
2.2 Tổng quan chức năng .....	5
2.2.1 Biểu đồ use case tổng quát .....	5
2.2.2 Biểu đồ use case phân rã Quản lý khóa học .....	6
2.2.3 Biểu đồ use case phân rã Quản lý tài khoản.....	6
2.2.4 Biểu đồ use case phân rã Quản lý điểm .....	7
2.2.5 Biểu đồ use case phân rã Quản lý câu hỏi.....	7
2.2.6 Biểu đồ use case phân rã Quản lý bài tập.....	8
2.2.7 Quy trình nghiệp vụ .....	8
2.3 Đặc tả chức năng .....	9
2.3.1 Đặc tả use case Đăng nhập.....	9
2.3.2 Đặc tả use case Đăng kí .....	9
2.3.3 Đặc tả use case Quản lý khóa học.....	11
2.3.4 Đặc tả use case Quản lý tài khoản.....	12
2.3.5 Đặc tả use case Đăng kí khóa học .....	14
2.3.6 Đặc tả use case Quản lý câu hỏi .....	15
2.3.7 Đặc tả use case Làm bài tập .....	16
2.3.8 Đặc tả use case Quản lý bài tập .....	17

2.3.9 Đặc tả use case Quản lí điểm.....	18
<b>2.4 Biểu đồ hoạt động.....</b>	<b>19</b>
2.4.1 Biểu đồ hoạt động cho usecase Đăng nhập .....	19
2.4.2 Biểu đồ hoạt động cho usecase Đăng kí.....	19
2.4.3 Biểu đồ hoạt động cho usecase Quản lí khóa học .....	20
2.4.4 Biểu đồ hoạt động cho usecase Quản lí tài khoản .....	20
2.4.5 Biểu đồ hoạt động cho usecase Quản lí câu hỏi .....	21
2.4.6 Biểu đồ hoạt động cho usecase Quản lí điểm .....	21
2.4.7 Biểu đồ hoạt động cho usecase Quản lí bài tập.....	22
<b>2.5 Yêu cầu phi chức năng .....</b>	<b>22</b>
<b>CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....</b>	<b>23</b>
3.1 HTML (HyperText Markup Language) .....	23
3.2 CSS (Cascading Style Sheets).....	23
3.3 JavaScript .....	23
3.4 Ngôn ngữ PHP .....	23
3.5 Mô hình MVC.....	24
3.6 Laravel Framework .....	24
3.7 Hệ quản trị cơ sở dữ liệu SQL .....	25
3.8 AJAX .....	26
<b>CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG ....</b>	<b>28</b>
4.1 Thiết kế kiến trúc.....	28
4.1.1 Lựa chọn kiến trúc phần mềm .....	28
4.1.2 Thiết kế gói .....	30
4.2 Thiết kế chi tiết.....	31
4.2.1 Thiết kế giao diện .....	31
4.2.2 Thiết kế lớp .....	33

4.2.3 Thiết kế cơ sở dữ liệu .....	35
4.3 Xây dựng ứng dụng.....	41
4.3.1 Thư viện và công cụ sử dụng.....	41
4.3.2 Kết quả đạt được .....	41
4.3.3 Minh họa các chức năng chính .....	42
4.4 Kiểm thử.....	47
4.5 Triển khai .....	48
<b>CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>50</b>
5.1 Một số vấn đề và giải pháp .....	50
5.2 Kết luận .....	50
5.3 Hướng phát triển.....	51



## DANH MỤC HÌNH VẼ

Hình 2.1	Usecase tổng quan hệ thống . . . . .	5
Hình 2.2	Use case phân rã Quản lý khóa học . . . . .	6
Hình 2.3	Use case phân rã Quản lý tài khoản . . . . .	6
Hình 2.4	Use case phân rã Quản lý điểm . . . . .	7
Hình 2.5	Use case phân rã Quản lý câu hỏi . . . . .	7
Hình 2.6	Use case phân rã Quản lý câu hỏi . . . . .	8
Hình 2.7	Biểu đồ hoạt động cho đăng nhập . . . . .	19
Hình 2.8	Biểu đồ hoạt động cho đăng ký . . . . .	19
Hình 2.9	Biểu đồ hoạt động cho Quản lý khóa học . . . . .	20
Hình 2.10	Biểu đồ hoạt động cho Quản lý tài khoản . . . . .	20
Hình 2.11	Biểu đồ hoạt động cho Quản lý câu hỏi . . . . .	21
Hình 2.12	Biểu đồ hoạt động cho Quản lý điểm . . . . .	21
Hình 2.13	Biểu đồ hoạt động cho Quản lý câu hỏi . . . . .	22
Hình 3.1	Mô hình MVC . . . . .	24
Hình 4.1	Thiết kế trang đăng nhập . . . . .	32
Hình 4.2	Thiết kế danh sách khóa học . . . . .	32
Hình 4.3	Thiết kế trang quản lý khóa học . . . . .	33
Hình 4.4	Thiết kế trang Tạo khóa học . . . . .	33
Hình 4.5	Biểu đồ lớp cho hệ thống . . . . .	34
Hình 4.6	Biểu đồ hoạt động đăng ký khóa học . . . . .	34
Hình 4.7	Biểu đồ hoạt động làm bài tập . . . . .	35
Hình 4.8	Biểu đồ thực thể liên kết . . . . .	36
Hình 4.9	Lưu thông tin khóa học . . . . .	36
Hình 4.10	Lưu thông tin vai trò người dùng . . . . .	37
Hình 4.11	Lưu điểm và đáp án . . . . .	37
Hình 4.12	Trang chủ . . . . .	43
Hình 4.13	Trang chủ2 . . . . .	43
Hình 4.14	Màn hình sau khi đăng nhập . . . . .	44
Hình 4.15	Trang làm bài tập . . . . .	44
Hình 4.16	Trang quản lý khóa học . . . . .	45
Hình 4.17	Trang tạo khóa học . . . . .	45
Hình 4.18	Trang chat khóa học . . . . .	46
Hình 4.19	Trang Thanh toán . . . . .	46

## **DANH MỤC BẢNG BIỂU**

Bảng 2.1	Đặc tả use case Đăng nhập . . . . .	9
Bảng 2.2	Đặc tả use case Đăng kí . . . . .	10
Bảng 2.3	Đặc tả use case Quản lí khóa học . . . . .	11
Bảng 2.4	Đặc tả use case Quản lí tài khoản . . . . .	13
Bảng 2.5	Đặc tả use case Đăng kí khóa học . . . . .	14
Bảng 2.6	Đặc tả use case Quản lí câu hỏi . . . . .	15
Bảng 2.7	Đặc tả use case Làm bài tập . . . . .	16
Bảng 2.8	Đặc tả use case Quản lí bài tập . . . . .	17
Bảng 2.9	Đặc tả use case Quản lí điểm . . . . .	18
Bảng 4.1	Bảng người dùng . . . . .	38
Bảng 4.2	Bảng khóa học . . . . .	38
Bảng 4.3	Bảng lưu điểm . . . . .	38
Bảng 4.4	Bảng câu hỏi . . . . .	39
Bảng 4.5	Bảng vai trò . . . . .	39
Bảng 4.6	Bảng quyền theo từng vai trò . . . . .	39
Bảng 4.7	Bảng nội dung khóa học . . . . .	40
Bảng 4.8	Bảng File của khóa học . . . . .	40
Bảng 4.9	Bảng bài tập . . . . .	41
Bảng 4.10	Thư viện và công cụ sử dụng . . . . .	41
Bảng 4.11	Kiểm thử chức năng đăng nhập, đăng kí . . . . .	47
Bảng 4.12	Kiểm thử chức năng nộp bài tập . . . . .	47
Bảng 4.13	Kiểm thử chức năng thanh toán . . . . .	48
Bảng 4.14	Kiểm thử chức năng tạo khóa học mới . . . . .	48
Bảng 4.15	Cấu hình máy tính triển khai hệ thống . . . . .	49

## **DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT**

<b>Thuật ngữ</b>	<b>Ý nghĩa</b>
API	Giao diện lập trình ứng dụng (Application Programming Interface)
CSS	Ngôn ngữ định kiểu theo tầng (Cascading Style Sheets)
EUD	Phát triển ứng dụng người dùng cuối(End-User Development)
GWT	Công cụ lập trình Javascript bằng Java của Google (Google Web Toolkit)
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
JSON	Định dạng dữ liệu JavaScript Object Notation
SQL	Ngôn ngữ truy vấn có cấu trúc (Structured Query Language)
UI	Giao diện người dùng (User Interface)
UX	Trải nghiệm người dùng (User Experience)

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

Trong bối cảnh giáo dục hiện đại, nhu cầu học tập trực tuyến ngày càng tăng cao. Sự phát triển của công nghệ thông tin và internet đã mở ra nhiều cơ hội mới cho việc học tập và giảng dạy. Tuy nhiên, nhiều nền tảng học tập trực tuyến hiện tại vẫn chưa đáp ứng đầy đủ các yêu cầu về tính tương tác, quản lý học tập và hỗ trợ người dùng. Các hệ thống hiện có thường gặp phải các hạn chế như giao diện phức tạp, thiếu tính năng hỗ trợ tương tác giữa giáo viên và học sinh, và khó khăn trong việc quản lý tiến độ học tập. Bởi vậy, việc đưa ra một hệ thống học tập trực tuyến có tính tương tác giữa giáo viên và người học là cần thiết. Điều này giúp cho việc áp dụng học trực tuyến vào các môi trường học tập trường học ở các cấp một cách hiệu quả hơn. Giúp giảm thời gian học cố định và tăng sự chủ động trong việc học cho người học như học sinh hay sinh viên hay có thể với người đang đi làm.

## 1.2 Mục tiêu và phạm vi đề tài

Phát triển một hệ thống học tập trực tuyến trên nền tảng website nhằm cung cấp một môi trường học tập linh hoạt, tiện lợi và hiệu quả cho người dùng. Hệ thống này sẽ giúp cải thiện trải nghiệm học tập của người học và giáo viên, đồng thời nâng cao hiệu quả quản lý học tập. Hiện tại, có nhiều nền tảng học tập trực tuyến như Coursera, Udemy, và Edx đã được phát triển và sử dụng rộng rãi. Mỗi nền tảng đều có những ưu điểm và nhược điểm riêng:

Coursera: Cung cấp nhiều khóa học từ các trường đại học hàng đầu, nhưng giao diện phức tạp và một số khóa học với chi phí cao.

Udemy: Đa dạng về nội dung khóa học, nhưng tính năng hỗ trợ tương tác giữa giáo viên và người học còn hạn chế.

Edx: Cung cấp các khóa học chất lượng cao, nhưng khó khăn trong việc quản lý tiến độ học tập.

Các hướng tiếp cận hiện tại chủ yếu tập trung vào việc cung cấp nội dung học tập và quản lý khóa học, nhưng chưa chú trọng đến tính tương tác và hỗ trợ người dùng. Trước những vấn đề trên, công việc của đồ án là đưa ra được hệ thống học tập trực tuyến giúp người học dễ dàng hơn trong việc quản lý tiến độ học tập, tăng sự tương tác giữa người học và giáo viên và chi phí được đưa ra với các khóa học là phù hợp và đảm bảo về chất lượng khóa học luôn cải thiện.

### 1.3 Định hướng giải pháp

Về phần tìm hiểu bài toán, nghiên cứu về cơ chế hoạt động và các thành phần liên quan đến lĩnh vực học tập trực tuyến. Tìm hiểu về các hệ thống học tập trực tuyến hiện đang được sử dụng. Về phần triển khai, tìm hiểu về quy trình phát triển phần mềm và công nghệ sử dụng cho đề tài và mô hình tổ chức mã nguồn sao cho hiệu quả. Sau khi tìm hiểu em quyết định sử dụng ngôn ngữ PHP để phát triển website. Sử dụng Laravel framework để lập trình PHP, Laravel là một framework được sử dụng phổ biến và hiệu quả trong việc quản lý mã nguồn cũng như hỗ trợ tương tác với cơ sở dữ liệu một cách dễ dàng. Được xây dựng dựa trên mô hình MVC(Model-View-Controller). Về CSDL, em sử dụng SQL Server để quản lý dữ liệu. Các đóng góp chính mà hệ thống học tập trực tuyến mang lại như: quản lý tài khoản người dùng, quản lý khóa học, bài giảng, hỗ trợ tương tác giữa người học và giáo viên, quản lý bài tập và theo dõi tiến độ học tập

### 1.4 Bố cục đồ án

Chương 2: Khảo sát và phân tích yêu cầu của đề tài. Tìm hiểu và các nội dung liên quan đến đề tài từ đó phân tích và đưa ra các biểu đồ dạng phân tích, nhằm hiểu chi tiết hơn về đề tài đang triển khai.

Chương 3: Trong chương này, em sẽ giới thiệu về công nghệ và nền tảng được sử dụng trong đồ án, bao gồm về ngôn ngữ lập trình, cơ sở dữ liệu và các công cụ khác giúp phát triển hệ thống.

Chương 4: Trong chương này, em trình bày chi tiết về thiết kế của hệ thống, thiết kế các lớp và quá trình kiểm thử và triển khai hệ thống.

Chương 5: Ở chương cuối, em tổng kết lại toàn bộ quá trình thực hiện đồ án, những kết quả đạt được, những kinh nghiệm cũng bài học rút ra. Đồng thời đề xuất các hướng phát triển của hệ thống trong tương lai.

## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

### 2.1 Khảo sát hiện trạng

#### 0.1.1 Đánh giá các phương pháp giảng dạy

##### \*Phương pháp giảng truyền thống

Người học tham gia trực tiếp tại lớp học theo thời khóa biểu đã được sắp xếp để thực hiện hoạt động học tập. Ngoài ra người học có thể giành thêm thời gian ở nhà để hoàn thành bài tập về nhà được giao hoặc tự học.

**Ưu điểm:** Duy trì sự giao tiếp giữa người dạy và người học hoặc giữa những người học (bạn bè). Ngoài ra, phương pháp học tập truyền thống giúp nâng cao động lực học tập cho người học. Động lực là một yếu tố cốt lõi dẫn đến thành công của hoạt động học tập. Vì vậy đây vẫn là một phương pháp học có hiệu quả cao nên vẫn được áp dụng phổ thông.

**Nhược điểm:** Một số nhược điểm có thể kể đến như việc cá nhân hóa hoạt động học tập và chi phí tổ chức môi trường học tập (trường học, trung tâm). Trong học tập truyền thống, người học phải tuân thủ lịch học đã sắp xếp cũng như quy định tại môi trường học. Điều này có thể gây khó khăn khi áp dụng cho tất cả người tham gia học tập. Về vấn đề chi phí tổ chức học tập cũng là yếu tố gây khó khăn khi triển khai các lớp học truyền thống. Một trong số các vấn đề chủ yếu là do số lượng người học trong một lớp. Nếu số lượng người học quá đông thì sẽ ảnh hưởng đến hiệu quả học tập. Ngược lại, nếu số lượng người học phù hợp (ít hơn) thì chi phí tổ chức lớp học sẽ tăng lên. Dẫn đến học phí người học phải trả sẽ cao hơn để đến các lớp học.Thêm vào đó, khi chia nhỏ người học thì phải đáp ứng thêm cơ sở hạ tầng (phòng học) cho người học.

Ở trên là một số nhược điểm cơ bản của phương pháp học tập truyền thống. Hiện nay với sự phát triển của khoa học công nghệ thì việc sử dụng kết hợp phương pháp học tập trực tuyến là biện pháp hiệu quả.

##### \*Phương pháp học tập trực tuyến

Đối với phương pháp này, người học không phải di chuyển đến các lớp học mà thực hiện hoạt động học tập từ xa thông qua các thiết bị giao tiếp có kết nối Internet.

**Ưu điểm:** Người học có thể tự lựa chọn, tự sắp xếp lịch học một cách phù hợp nhất với điều kiện của mình. Và việc lựa chọn thời gian học tập phù hợp cũng là một yếu tố quan trọng giúp nâng cao hiệu quả học tập của người học. Một ưu điểm tiếp theo, phương pháp này giúp giảm thiểu chi phí tổ chức học tập, đồng thời tiếp

cận với một số lượng lớn người học và không bị phụ thuộc vào vị trí, vùng miền.

Nhược điểm: Việc duy trì động lực học tập cho người học trong môi trường trực tuyến là một vấn đề khó khăn. Nếu việc thiết kế và triển khai các khóa học chưa tốt hoặc tính năng theo dõi tiến độ không hiệu quả thì có thể sẽ ảnh hưởng nhiều đến kết quả người học hoặc nếu nghiêm trọng hơn sẽ không thể hoàn thành khóa học.

Hệ thống được triển khai sẽ khắc phục các nhược điểm trên giúp các khóa học sinh động và phong phú với các dạng tài liệu học khác nhau. Ngoài ra, hệ thống theo dõi tiến độ người học được thiết kế sẽ bám sát lộ trình học của người học giúp người học nắm bắt được quá trình rèn luyện học tập của mình một cách chủ động.

### 0.1.2. Tìm hiểu một số nền tảng hỗ trợ học tập trực tuyến

\*Google Classroom Hệ thống tổ chức lớp học thông qua việc hỗ trợ ba tính năng chính: thông báo, bài tập và lưu trữ. Thông báo và gửi phản hồi và tích hợp thêm email. Giáo viên đính kèm thêm tài liệu ở trong các thông báo. Việc giao bài tập và nhận bài tập được thực hiện thông qua ứng dụng Google Drive chia sẻ chung. Mỗi lớp học được tổ chức vào trong một thư mục riêng, mỗi học sinh sẽ được tự động tạo một thư mục cho phần bài tập của mình.

Ưu điểm: Giao diện được thiết kế đơn giản, quen thuộc với những người đã sử dụng qua các sản phẩm khác trước đó của Google. Việc thiết kế khóa học, đăng ký và vận hành diễn ra khá đơn giản và dễ dàng.

Nhược điểm: Với việc thiết kế đơn giản, ít tính năng dẫn đến một số hạn chế trong quá trình sử dụng. Sử dụng email thuộc gói Google Education để đăng ký cũng là 1 rào cản khiến cho nền tảng không dễ được phổ biến.

\*Coursera Coursera là một công ty công nghệ giáo dục cung cấp các khóa học trực tuyến mở. Coursera cộng tác với các trường đại học và một số tổ chức giáo dục khác để xây dựng các khóa học về nhiều lĩnh vực khác nhau. Các khóa học trên Coursera được cung cấp dưới dạng miễn phí. Sau đó, người học có thể trả tiền để được cấp chứng chỉ hoàn thành đối với khóa học đó. Các chứng chỉ này khá uy tín và có thể được sử dụng vào mục đích nghề nghiệp.

Ưu điểm: Với sự cộng tác của các trường Đại học hàng đầu nên khóa học rất chất lượng và uy tín.

Nhược điểm: Chỉ có các đối tác của Coursera mới đưa được khóa học lên nền tảng này, do đó hiện tại chưa thể áp dụng phổ biến trong các tổ chức giáo dục khác.

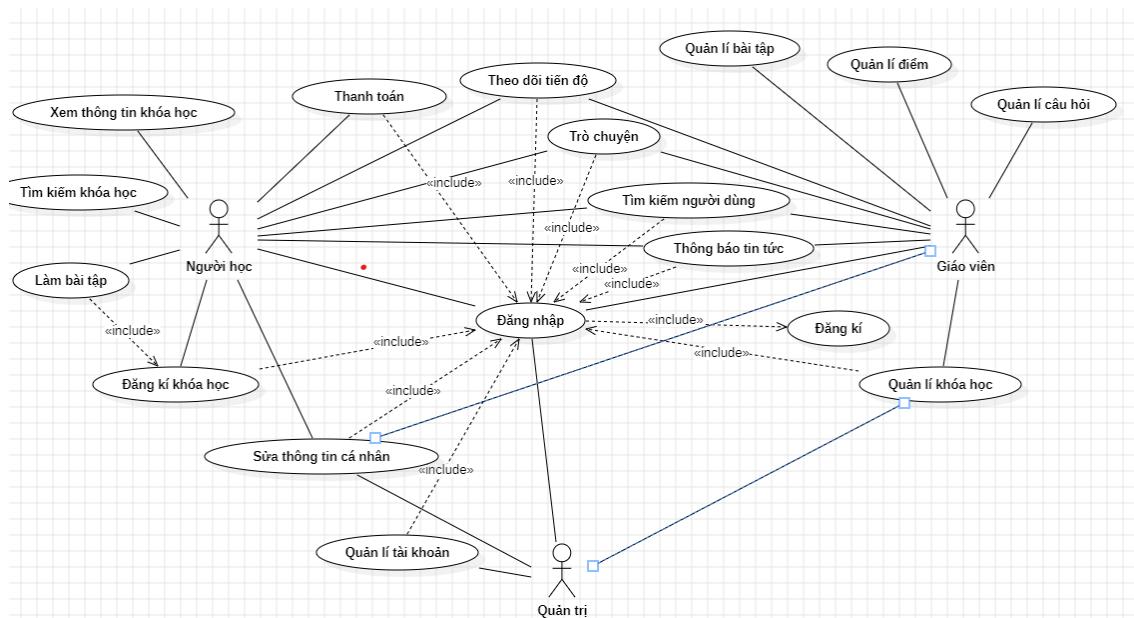
\*Udemy

Udemy là một nền tảng học trực tuyến rất phong phú về chủng loại và đến từ nhiều ngành nghề, ngôn ngữ khác nhau dành cho nhiều đối tượng. Cơ chế hỗ trợ giáo viên xây dựng các khóa học của Udemy cũng được thực hiện rất tốt.

Từ những phân tích trên, Các tính năng cơ bản cần được quan tâm như quản lý khóa học bao gồm tạo các tài liệu, bài tập, quản lí tiến độ người học, phần trao đổi, giải đáp liên quan đến khóa học, môn học và về phía người quản lí bao gồm việc quản lí người dùng như người học giáo viên và các tính năng của hệ thống.

## 2.2 Tổng quan chức năng

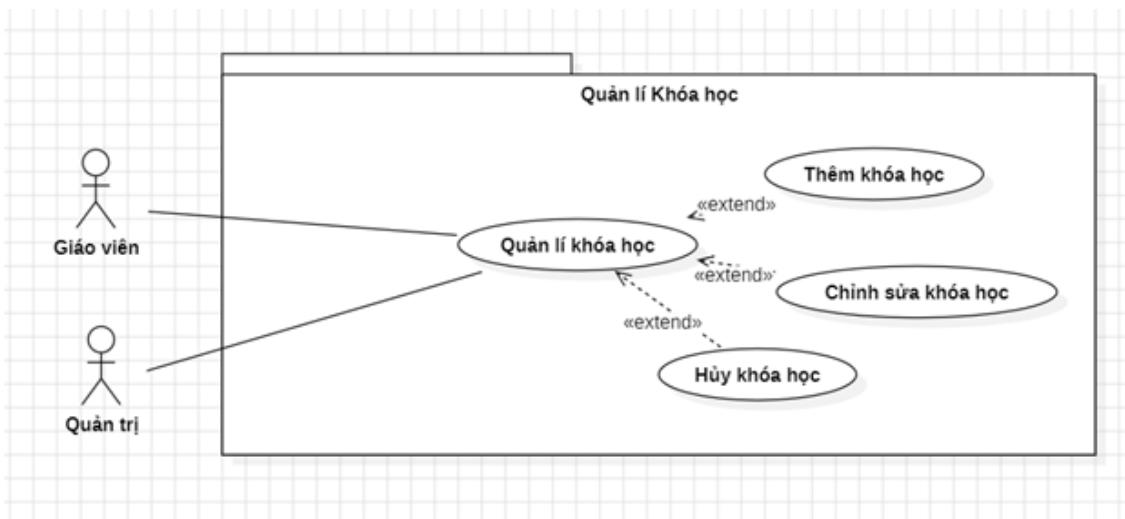
### 2.2.1 Biểu đồ use case tổng quát



Hình 2.1: Usecase tổng quan hệ thống

Biểu đồ use case tổng quan xác định các tác nhân chính của hệ thống là người học, giáo viên và quản trị. Người học có các hoạt động trên hệ thống như xem thông tin khóa học, tìm kiếm khóa học. Khi người học đã đăng kí và đăng nhập thành công vào hệ thống thì có thể chỉnh sửa thông tin cá nhân, tìm kiếm người dùng, nhận được các thông báo, theo dõi tiến độ học tập. Khi đã đăng kí khóa học thì người học sẽ phải thanh toán nếu khóa học có phí, có thể trò chuyện, trao đổi trong khóa học. Giáo viên có thể quản lí khóa học, quản lí các học viên trong khóa học, trao đổi thông qua khung chat, tìm kiếm người dùng, tạo và quản lí bài tập, câu hỏi và điểm của học viên trong khóa học. Với người quản trị có thể thực hiện các chức năng quản lí tài khoản người dùng, quản lí các khóa học

### 2.2.2 Biểu đồ use case phân rã Quản lý khóa học



**Hình 2.2:** Use case phân rã Quản lí khóa học

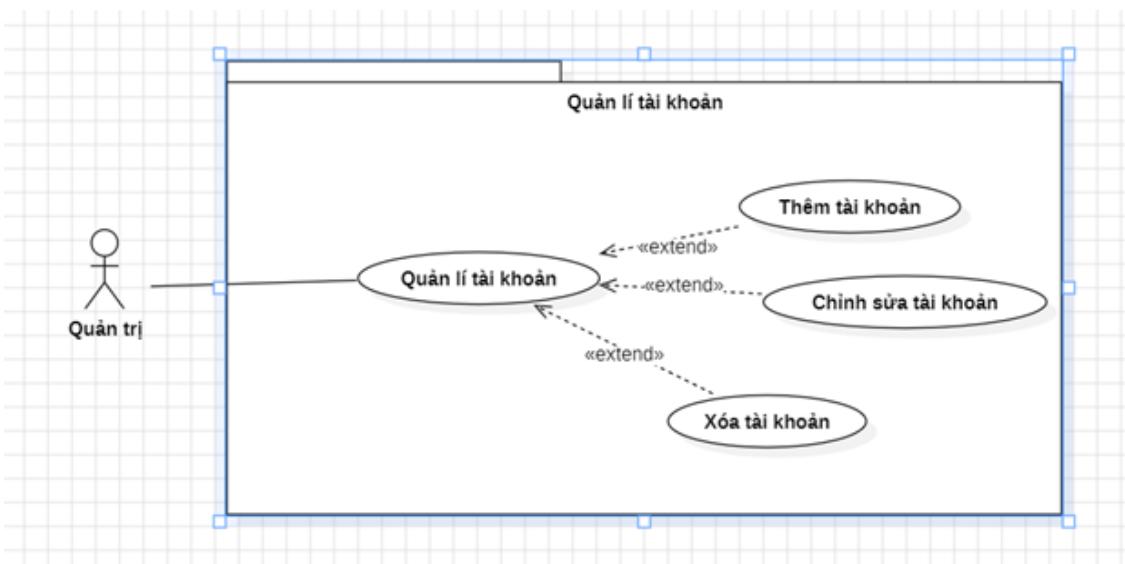
Phân rã use case quản lý khóa học có các use case con cụ thể như:

Thêm (tạo mới) khóa học: Giáo viên, quản trị nhập thông tin khóa học để tạo khóa học mới.

Chỉnh sửa khóa học: Giáo viên, quản trị chỉnh sửa thông tin của khóa học. Hệ thống lưu lại các thay đổi.

Xóa khóa học: Giáo viên, quản trị yêu cầu xóa khóa học khỏi hệ thống. Hệ thống xác nhận và xóa khóa học khỏi cơ sở dữ liệu.

### 2.2.3 Biểu đồ use case phân rã Quản lý tài khoản



**Hình 2.3:** Use case phân rã Quản lí tài khoản

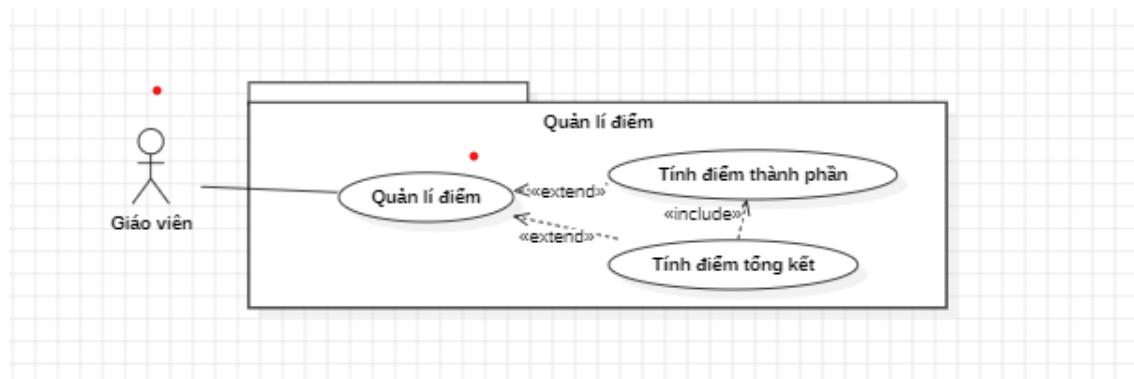
## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Phân rã use case quản lý tài khoản có các use case con cụ thể hơn. Thêm (Đăng ký) tài khoản: Người dùng nhập thông tin cá nhân. Hệ thống kiểm tra thông tin và tạo tài khoản mới.

Chỉnh sửa tài khoản: Người dùng chỉnh sửa thông tin cá nhân. Hệ thống lưu lại các thay đổi.

Xóa tài khoản: Người dùng yêu cầu xóa tài khoản. Hệ thống xác nhận và xóa tài khoản khỏi cơ sở dữ liệu.

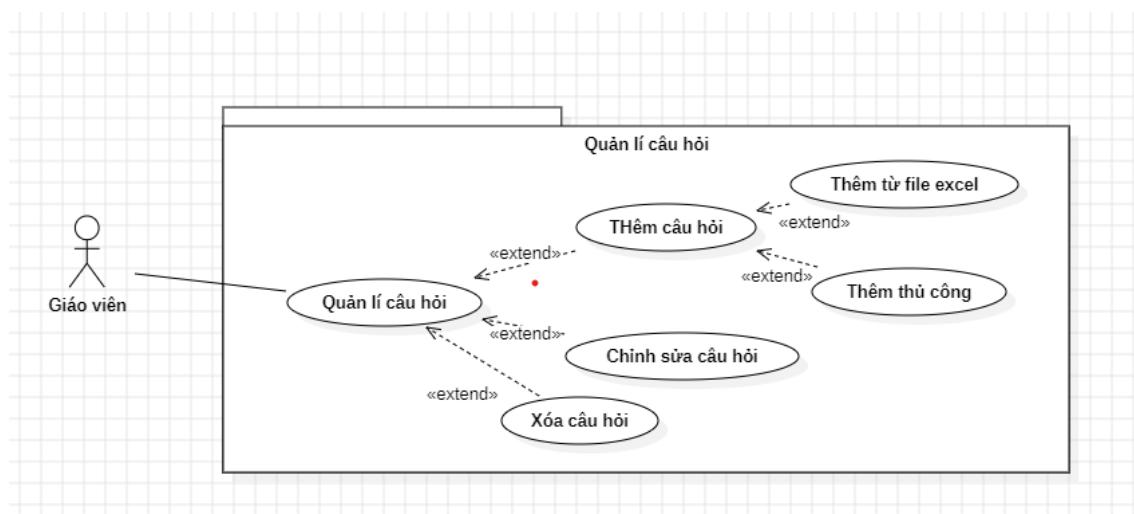
### 2.2.4 Biểu đồ use case phân rã Quản lý điểm



Hình 2.4: Use case phân rã Quản lý điểm

Phân rã use case quản lý điểm có các use case con cụ thể hơn. Giáo viên quản lý các điểm thành phần thông qua các bài tập và tính ra được điểm tổng kết.

### 2.2.5 Biểu đồ use case phân rã Quản lý câu hỏi



Hình 2.5: Use case phân rã Quản lý câu hỏi

Phân rã use case quản lý câu hỏi có các use case con cụ thể hơn.

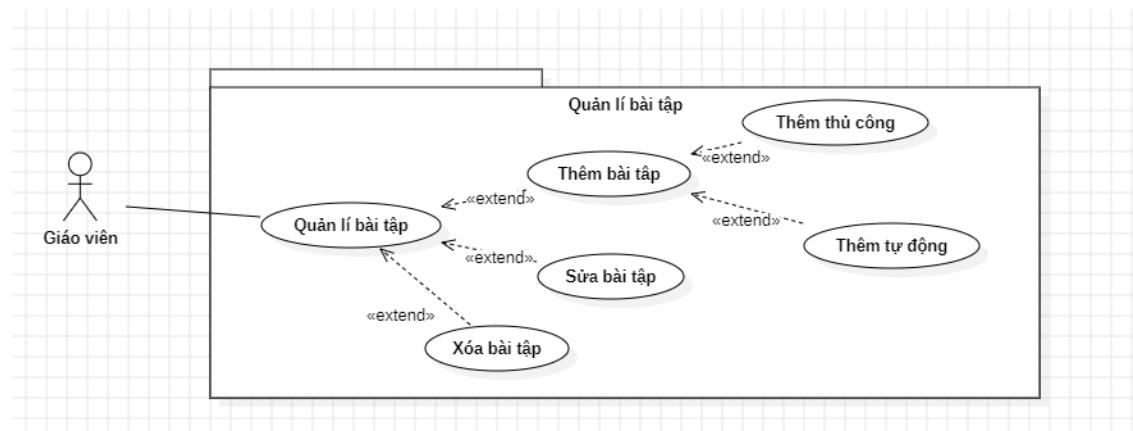
Thêm câu hỏi mới cho khóa học tương ứng: câu hỏi được thêm thủ công hoặc

import từ file excel.

Chỉnh sửa câu hỏi: Giáo viên chỉnh sửa các câu hỏi và đáp án, hệ thống sẽ lưu lại các thay đổi.

Xóa câu hỏi: Giáo viên chọn xóa câu hỏi và hệ thống yêu cầu xác nhận xóa câu hỏi tương ứng.

### 2.2.6 Biểu đồ use case phân rã Quản lý bài tập



**Hình 2.6:** Use case phân rã Quản lý câu hỏi

Phân rã use case quản lý bài tập có các use case con cụ thể hơn.

Thêm bài tập mới cho khóa học tương ứng: bài tập được thêm tự động hoặc thêm thủ công.

Chỉnh sửa bài tập: Giáo viên chỉnh sửa các bài tập bằng việc thay đổi số lượng câu hỏi, hệ thống sẽ lưu lại các thay đổi.

Xóa bài tập: Giáo viên chọn xóa bài tập và hệ thống yêu cầu xác nhận xóa bài tập tương ứng.

### 2.2.7 Quy trình nghiệp vụ

Quy trình nghiệp vụ đăng ký khóa học: Người học truy cập vào hệ thống để tìm kiếm các khóa học cần đăng kí. Người học có thể xem thông tin mô tả về khóa học. Nếu chưa có tài khoản, người học cần tạo một tài khoản. Sau khi đăng nhập, người học chọn khóa học muốn tham gia từ danh sách khóa học có sẵn. Nếu khóa học có phí thì người học thanh toán phí. Sau khi thanh toán thành công, người học sẽ nhận được email hoặc thông báo xác nhận đăng ký khóa học.

Quy trình nghiệp vụ tạo khóa học: Giáo viên thiết kế nội dung bao gồm tài liệu, video, thời gian hoàn thành khóa học và các tài nguyên khác mà người học sử dụng. Mỗi khóa học có thể được quản lý bởi nhiều giáo viên, tính năng có thể giúp các

giáo viên hợp tác và chia sẻ trong quá trình xây dựng cũng như triển khai khóa học

### **2.3 Đặc tả chức năng**

#### **2.3.1 Đặc tả use case Đăng nhập**

Mã usecase	UC-01
Tên usecase	Đăng nhập
Mục tiêu	Người dùng có thể đăng nhập vào hệ thống bằng tài khoản đã đăng ký
Tác nhân chính	Người học, giáo viên, quản trị
Tiền điều kiện	Người dùng đã đăng ký tài khoản
Hậu điều kiện	Người dùng được xác thực và chuyển đến trang tùy theo vai trò của tài khoản
Luồng sự kiện chính	1. Người dùng chọn chức năng đăng nhập. 2. Hệ thống nhận và xử lý yêu cầu và sau đó trả về trang đăng nhập 3. Người dùng nhập tên tài khoản và mật khẩu đã đăng ký và gửi thông tin đăng nhập tới hệ thống 4. Hệ thống kiểm tra thông tin người dùng đăng nhập [e4.1] 5. Hệ thống trả về trang tương ứng với vai trò của tài khoản
Luồng sự kiện ngoại lệ	e4.1. Hệ thống kiểm tra thông tin đăng nhập không hợp lệ e4.2 Hệ thống trả về thông báo lỗi, yêu cầu người dùng đăng nhập lại. Quay lại bước 3

**Bảng 2.1: Đặc tả use case Đăng nhập**

#### **2.3.2 Đặc tả use case Đăng kí**

## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Mã usecase	UC-02
Tên usecase	Đăng kí
Mục tiêu	Người dùng có thể đăng kí tài khoản
Tác nhân chính	Người học, giáo viên
Tiền điều kiện	Người dùng muốn đăng nhập để sử dụng các chức năng của hệ thống
Hậu điều kiện	Người dùng đăng kí thành công và chuyển sang trang đăng nhập
Luồng sự kiện chính	1. Người dùng chọn chức năng đăng kí. 2. Hệ thống nhận và xử lý yêu cầu và sau đó trả về trang đăng kí 3. Người dùng nhập thông tin theo yêu cầu và gửi thông tin đăng kí tới hệ thống 4. Hệ thống nhận và kiểm tra thông tin. [e4.1] 5. Hệ thống xác nhận thông tin hợp lệ, chuyển đến trang đăng nhập
Luồng sự kiện ngoại lệ	e4.1. Hệ thống kiểm tra thông tin đăng kí không hợp lệ e4.2 Hệ thống trả về thông báo lỗi, yêu cầu người dùng nhập lại thông tin. Quay lại bước 3

**Bảng 2.2:** Đặc tả use case Đăng kí

### 2.3.3 Đặc tả use case Quản lí khóa học

Mã use-case	UC-03
Tên use-case	Quản lí khóa học
Mục tiêu	Người dùng có thể thao tác với các chức năng liên quan đến quản lí khóa học
Tác nhân chính	Giáo viên, quản trị
Tiền điều kiện	Người dùng với vai trò là giáo viên hoặc quản trị
Hậu điều kiện	Thông tin khóa học được thêm mới hoặc chỉnh sửa hoặc hủy thành công.
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Tác nhân chọn chức năng quản lý khóa học.</li> <li>2. Hệ thống xác nhận và xử lý yêu cầu và trả về trang quản lí khóa học bao gồm danh sách các khóa học hiện có và các tùy chọn thêm mới, chỉnh sửa, xóa khóa học</li> <li>3. Tác nhân chọn thêm mới khóa học</li> <li>4. Hệ thống trả về trang tạo khóa học</li> <li>5. Người dùng nhập thông tin theo yêu cầu và gửi thông tin tới hệ thống</li> <li>6. Hệ thống nhận và kiểm tra thông tin. [e6.1]</li> <li>7. Hệ thống xác nhận thông tin hợp lệ và khóa học được thêm thành công</li> <li>8. Cập nhập lại danh sách các khóa học hiện có</li> </ol>
Luồng sự kiện thay thế	alt3.1.1 Tác nhân chọn chức năng chỉnh sửa khóa học alt3.1.2 Hệ thống trả về trang chỉnh sửa khóa học tương ứng alt3.1.3 Tác nhân chỉnh sửa các thông tin cần thiết và gửi yêu cầu cập nhập tới hệ thống alt3.1.4 Hệ thống nhận và kiểm tra thông tin. [e3.1.4] alt3.1.5 Hệ thống xác nhận thông tin hợp lệ và khóa học được chỉnh sửa thành công alt3.1.6 Cập nhập lại thông tin của khóa học hiện được chỉnh sửa alt3.2.1 Tác nhân chọn chức năng xóa khóa học alt3.2.2 Hệ thống hiển thị thông báo xác nhận lại “Bạn muốn xóa khóa này ?” alt3.2.3 Tác nhân xác nhận xóa khóa học [e3.2.3.1] alt3.2.4 Hệ thống nhận và thực hiện xóa khóa học alt3.2.5 Hệ thống cập nhập lại danh sách khóa học

Bảng 2.3: Đặc tả use case Quản lí khóa học

Luồng sự kiện ngoại lệ

e6.1. Hệ thống kiểm tra thông tin không hợp lệ

e6.2 Hệ thống trả về thông báo lỗi, yêu cầu người dùng nhập lại thông tin

Quay lại bước 5

e3.1.4.1 Hệ thống kiểm tra thông tin muốn chỉnh sửa không hợp lệ

e3.1.4.2 Hệ thống trả về thông báo lỗi, yêu cầu người dùng nhập lại thông tin

Quay lại bước 3.1.3

e3.2.3.1 Tác nhân chọn hủy yêu cầu xóa khóa học

e3.2.3.2 Hệ thống hủy yêu cầu xóa khóa học

### **2.3.4 Đặc tả use case Quản lý tài khoản**

## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Mã use-case	UC-04
Tên use-case	Quản lí tài khoản
Mục tiêu	Người dùng có thể thao tác với các chức năng liên quan đến quản lí tài khoản
Tác nhân chính	Quản trị
Tiền điều kiện	Người dùng với vai trò là quản trị
Hậu điều kiện	Thông tin tài khoản được thêm mới hoặc chỉnh sửa hoặc hủy thành công.
Luồng sự kiện chính	<ul style="list-style-type: none"> <li>1. Tác nhân chọn chức năng quản lý tài khoản.</li> <li>2. Hệ thống xác nhận và xử lý yêu cầu và trả về trang quản lí tài khoản bao gồm danh sách các tài khoản hiện có và các tùy chọn thêm mới, chỉnh sửa, xóa tài khoản</li> <li>3. Tác nhân chọn thêm mới tài khoản</li> <li>4. Hệ thống trả về trang tạo tài khoản</li> <li>5. Tác nhân nhập thông tin theo yêu cầu và gửi thông tin tới hệ thống</li> <li>6. Hệ thống nhận và kiểm tra thông tin. [e6.1]</li> <li>7. Hệ thống xác nhận thông tin hợp lệ và tài khoản được thêm thành công</li> <li>8. Cập nhập lại danh sách các tài khoản hiện có</li> </ul>
Luồng sự kiện thay thế	<ul style="list-style-type: none"> <li>alt3.1.1 Tác nhân chọn chức năng chỉnh sửa tài khoản</li> <li>alt3.1.2 Hệ thống trả về trang chỉnh sửa tài khoản tương ứng</li> <li>alt3.1.3 Tác nhân chỉnh sửa các thông tin cần thiết và gửi yêu cầu cập nhập tới hệ thống</li> <li>alt3.1.4 Hệ thống nhận và kiểm tra thông tin. [e3.1.4]</li> <li>alt3.1.5 Hệ thống xác nhận thông tin hợp lệ và tài khoản được chỉnh sửa thành công</li> <li>alt3.1.6 Cập nhập lại thông tin của tài khoản hiện được chỉnh sửa</li> <li>alt3.2.1 Tác nhân chọn chức năng xóa tài khoản</li> <li>alt3.2.2 Hệ thống hiển thị thông báo xác nhận lại “Bạn muốn xóa tài khoản này ?”</li> <li>alt3.2.3 Tác nhân xác nhận xóa tài khoản [e3.2.3.1]</li> <li>alt3.2.4 Hệ thống nhận và thực hiện xóa tài khoản</li> <li>alt3.2.5 Hệ thống cập nhật lại danh sách tài khoản</li> </ul>

**Bảng 2.4:** Đặc tả use case Quản lí tài khoản

Luồng sự kiện ngoại lệ

e6.1. Hệ thống kiểm tra thông tin không hợp lệ

e6.2 Hệ thống trả về thông báo lỗi, yêu cầu người dùng nhập lại thông tin

Quay lại bước 5

e3.1.4.1 Hệ thống kiểm tra thông tin muốn chỉnh sửa không hợp lệ

e3.1.4.2 Hệ thống trả về thông báo lỗi, yêu cầu người dùng nhập lại thông tin

Quay lại bước 3.1.3

e3.2.3.1 Tác nhân chọn hủy yêu cầu xóa tài khoản

e3.2.3.2 Hệ thống hủy yêu cầu xóa tài khoản

### **2.3.5 Đặc tả use case Đăng kí khóa học**

Mã usecase	UC-05
Tên usecase	Đăng kí khóa học
Mục tiêu	Người học có thể đăng kí các khóa học hiện mở để tham gia
Tác nhân chính	Người dùng
Tiền điều kiện	Người học đã đăng nhập và chọn được khóa học muốn đăng kí
Hậu điều kiện	Người học đăng kí khóa học thành công
Luồng sự kiện chính	<ul style="list-style-type: none"> <li>1. Tác nhân chọn chức năng đăng kí khóa học</li> <li>2. Hệ thống gửi thông báo xác nhận yêu cầu đăng kí</li> <li>3. Tác nhân chọn xác nhận để tiếp tục [e3.1]</li> <li>4. Hệ thống kiểm tra khóa học có phí và trả về trang thanh toán [alt4.1]</li> <li>5. Tác nhân nhập thông tin thanh toán hợp lệ</li> <li>6. Ngân hàng xác thực thông tin và hoàn tất giao dịch và trả về trạng thái cho hệ thống[e.6.1]</li> <li>7. Hệ thống lưu lại các thông tin giao dịch</li> <li>8. Hệ thống thông báo thành công và thêm khóa học vào mục khóa học đã đăng kí</li> </ul>
Luồng sự kiện thay thế	<ul style="list-style-type: none"> <li>Alt4.1.1 Hệ thống kiểm tra khóa học miễn phí</li> <li>Alt4.1.2 Hệ thống gửi yêu cầu đăng kí đến trang quản lý khóa học</li> <li>Alt4.1.3 Người quản lý khóa học chấp nhận yêu cầu đăng kí</li> </ul> <p>Tiếp tục bước 8</p>
Luồng sự kiện ngoại lệ	<ul style="list-style-type: none"> <li>e3.1 Tác nhân chọn hủy yêu cầu đăng kí</li> <li>e3.2 Hệ thống hủy yêu cầu</li> <li>e6.1 Thông tin thanh toán không hợp lệ</li> </ul> <p>Quay lại bước 5</p>

**Bảng 2.5:** Đặc tả use case Đăng kí khóa học

### 2.3.6 Đặc tả use case Quản lí câu hỏi

Mã usecase	UC-06
Tên usecase	Quản lí câu hỏi
Mục tiêu	Người học có thể tạo các câu hỏi với các khóa học tương ứng
Tác nhân chính	Giáo viên, quản trị
Tiền kiện	Tác nhân muốn thêm hoặc chỉnh sửa hoặc xóa câu hỏi
Hậu kiện	Câu hỏi được thêm mới hoặc chỉnh sửa hoặc xóa thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Tác nhân chọn chức năng quản lý câu hỏi.</li> <li>2. Hệ thống xác nhận và xử lý yêu cầu và trả về trang quản lí câu hỏi bao gồm danh sách các câu hỏi hiện có và các tùy chọn thêm mới, chỉnh sửa, xóa câu hỏi</li> <li>3. Tác nhân chọn thêm mới câu hỏi[alt3.1, alt3.2]</li> <li>4. Tác nhân chọn thêm câu hỏi thủ công [alt4.1]</li> <li>5. Hệ thống trả về trang tạo câu hỏi</li> <li>6. Tác nhân nhập thông tin theo yêu cầu và gửi thông tin tới hệ thống</li> <li>7. Hệ thống nhận và kiểm tra thông tin. [e7.1]</li> <li>8. Hệ thống xác nhận thông tin hợp lệ và câu hỏi được thêm thành công</li> <li>9. Cập nhật lại danh sách các câu hỏi hiện có</li> </ol>
Luồng sự kiện thay thế	alt3.1.1 Tác nhân chọn chức năng chỉnh sửa câu hỏi alt3.1.2 Hệ thống trả về trang chỉnh sửa câu hỏi tương ứng alt3.1.3 Tác nhân chỉnh sửa các thông tin cần thiết và gửi yêu cầu cập nhật tới hệ thống alt3.1.4 Hệ thống nhận và kiểm tra thông tin. [e3.1.4] alt3.1.5 Hệ thống xác nhận thông tin hợp lệ và câu hỏi được chỉnh sửa thành công alt3.1.6 Cập nhật lại thông tin của câu hỏi hiện được chỉnh sửa alt3.2.1 Tác nhân chọn chức năng xóa câu hỏi alt3.2.2 Hệ thống hiển thị thông báo xác nhận lại “Bạn muốn xóa câu hỏi này ?” alt3.2.3 Tác nhân xác nhận xóa câu hỏi [e3.2.3.1] alt3.2.4 Hệ thống nhận và thực hiện xóa câu hỏi alt3.2.5 Hệ thống cập nhật lại danh sách câu hỏi alt4.1.1 Tác nhân chọn thêm câu hỏi từ file excel alt4.1.2 Tác nhân nhập dữ liệu vào file excel theo định dạng có sẵn tiếp tục bước 7

Bảng 2.6: Đặc tả use case Quản lí câu hỏi

Luồng sự kiện ngoại lệ

E7.1. Hệ thống kiểm tra thông tin không hợp lệ

E7.2 Hệ thống trả về thông báo lỗi, yêu cầu người dùng nhập lại thông tin

Quay lại bước 5

E3.1.4.1 Hệ thống kiểm tra thông tin muốn chỉnh sửa không hợp lệ

E3.1.4.2 Hệ thống trả về thông báo lỗi, yêu cầu người dùng nhập lại thông tin

Quay lại bước 3.1.3

e3.2.3.1 Tác nhân chọn hủy yêu cầu xóa câu hỏi

e3.2.3.2 Hệ thống hủy yêu cầu xóa câu hỏi

### **2.3.7 Đặc tả use case Làm bài tập**

Mã usecase	UC-07
Tên usecase	Làm bài tập
Mục tiêu	Người học thực hiện việc làm bài tập dưới hình thức câu hỏi trắc nghiệm
Tác nhân chính	Người học
Tiền điều kiện	Tác nhân đã tham gia ít nhất 1 khóa học
Hậu điều kiện	Tác nhân hoàn thành và nộp bài tập thành công
Luồng sự kiện chính	<ul style="list-style-type: none"> <li>1. Tác nhân chọn bài tập cần làm</li> <li>2. Hệ thống nhận yêu cầu và xử lý và sau đó hiển thị đề bài với các câu hỏi tương ứng</li> <li>3. Tác nhân thực hiện làm bài dưới dạng câu hỏi trắc nghiệm.</li> <li>4. Tác nhân chọn nộp bài</li> <li>5. Hệ thống thông báo xác nhận lại tác nhân muốn nộp bài</li> <li>6. Tác nhân xác nhận nộp</li> <li>7. Hệ thống kiểm tra và hiển thị thông báo nộp bài thành công [e7.1]</li> </ul>
Luồng sự kiện ngoại lệ	<p>E7.1. Hệ thống kiểm tra chưa hoàn thành hết tất cả các câu hỏi</p> <p>E7.2 Hệ thống trả về thông báo, yêu cầu tác nhân hoàn thành hết các câu hỏi</p> <p>Quay lại bước 3</p>

**Bảng 2.7: Đặc tả use case Làm bài tập**

### 2.3.8 Đặc tả use case Quản lí bài tập

Mã usecase	UC-08
Tên usecase	Quản lí bài tập
Mục tiêu	Người học có thể tạo các bài tập với các khóa học tương ứng
Tác nhân chính	Giáo viên, quản trị
Tiền điều kiện	Tác nhân muốn thêm hoặc chỉnh sửa hoặc xóa bài tập
Hậu điều kiện	Bài tập được thêm mới hoặc chỉnh sửa hoặc xóa thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Tác nhân chọn chức năng quản lý bài tập.</li> <li>2. Hệ thống xác nhận và xử lý yêu cầu và trả về trang quản lí bài tập bao gồm danh sách các bài tập hiện có và các tùy chọn thêm mới, chỉnh sửa, xóa bài tập</li> <li>3. Tác nhân chọn thêm mới bài tập[alt3.1, alt3.2]</li> <li>4. Tác nhân chọn thêm bài tập thủ công [alt4.1]</li> <li>5. Hệ thống trả về trang tạo bài tập</li> <li>6. Tác nhân nhập chọn các câu hỏi và gửi thông tin tới hệ thống</li> <li>7. Hệ thống nhận và kiểm tra thông tin. [e7.1]</li> <li>8. Hệ thống xác nhận thông tin hợp lệ và bài tập được thêm thành công</li> <li>9. Cập nhập lại danh sách các bài tập hiện có</li> </ol>
Luồng sự kiện thay thế	alt3.1.1 Tác nhân chọn chức năng chỉnh sửa bài tập alt3.1.2 Hệ thống trả về trang chỉnh sửa bài tập tương ứng alt3.1.3 Tác nhân chỉnh sửa các thông tin cần thiết và gửi yêu cầu cập nhật tới hệ thống alt3.1.4 Hệ thống nhận và kiểm tra thông tin. [e3.1.4] alt3.1.5 Hệ thống xác nhận thông tin hợp lệ và bài tập được chỉnh sửa thành công alt3.1.6 Cập nhập lại thông tin của bài tập hiện được chỉnh sửa alt3.2.1 Tác nhân chọn chức năng xóa bài tập alt3.2.2 Hệ thống hiển thị thông báo xác nhận lại “Bạn muốn xóa bài tập này ?” alt3.2.3 Tác nhân xác nhận xóa bài tập [e3.2.3.1] alt3.2.4 Hệ thống nhận và thực hiện xóa bài tập alt3.2.5 Hệ thống cập nhật lại danh sách bài tập alt4.1.1 Tác nhân chọn thêm bài tập tự động alt4.1.2 Hệ thống yêu cầu tác nhân nhập số lượng câu hỏi alt4.1.3 Tác nhân nhập số lượng câu hỏi mong muốn alt4.1.4 Hệ thống chọn ngẫu nhiên các câu hỏi để cấu thành bài tập quay lại bước 9

**Bảng 2.8:** Đặc tả use case Quản lí bài tập

Luồng sự kiện ngoại lệ

E7.1. Hệ thống kiểm tra thông tin không hợp lệ

E7.2 Hệ thống trả về thông báo lỗi, yêu cầu tác nhân thực hiện lại

Quay lại bước 6

E3.1.4.1 Hệ thống kiểm tra thông tin muốn chỉnh sửa không hợp lệ

E3.1.4.2 Hệ thống trả về thông báo lỗi, yêu cầu tác nhân thực hiện lại

Quay lại bước 3.1.3

e3.2.3.1 Tác nhân chọn hủy yêu cầu xóa bài tập

e3.2.3.2 Hệ thống hủy yêu cầu xóa bài tập

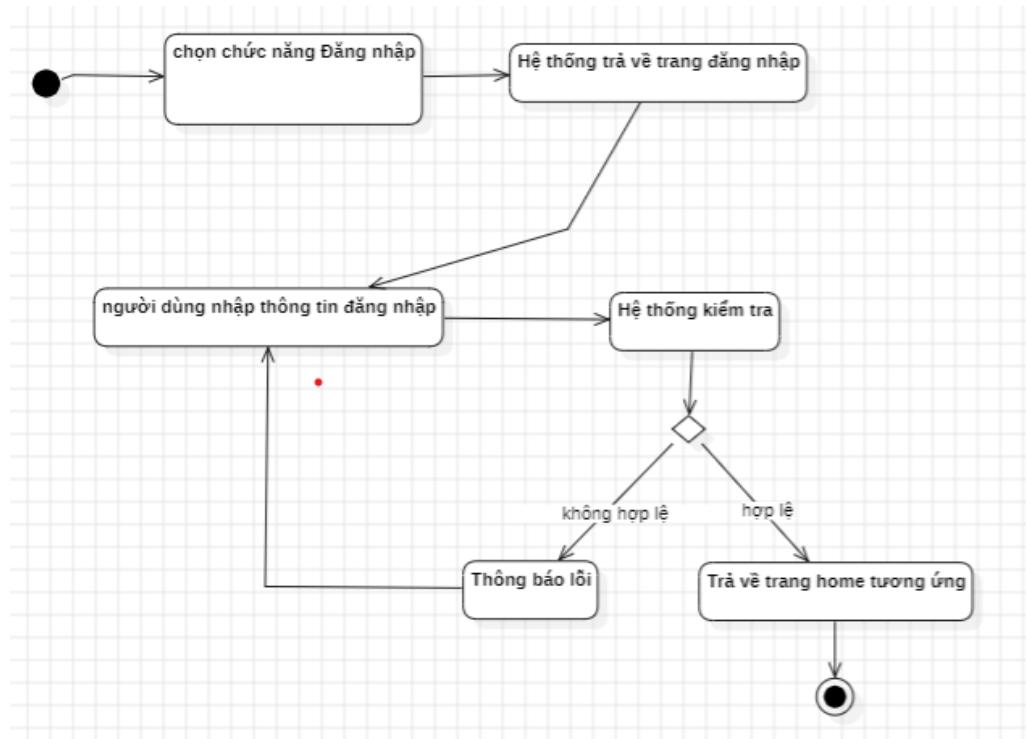
### **2.3.9 Đặc tả use case Quản lý điểm**

Mã usecase	UC-09
Tên usecase	Quản lý điểm
Mục tiêu	Giáo viên có thể quản lý điểm từng bài tập với khóa học tương ứng
Tác nhân chính	Giáo viên, quản trị
Tiền điều kiện	Tác nhân muốn quản lý điểm của người học
Hậu điều kiện	Tác nhân có thể thao tác với điểm của người học
Luồng sự kiện chính	1. Tác nhân chọn chức năng quản lý điểm của người học 2. Hệ thống xác nhận và xử lý yêu cầu và trả về trang quản lý điểm bao gồm danh sách các bài tập mà các người học đã thực hiện, và tùy chọn hiển thị điểm chi tiết của người học 3. Tác nhân chọn xem chi tiết điểm của người học 4. Tác nhân chọn xem chi tiết đáp án mà người học đã lựa chọn 5. Hệ thống hiển thị trang bài tập và hiển thị đáp án mà người học đã lựa chọn

**Bảng 2.9:** Đặc tả use case Quản lý điểm

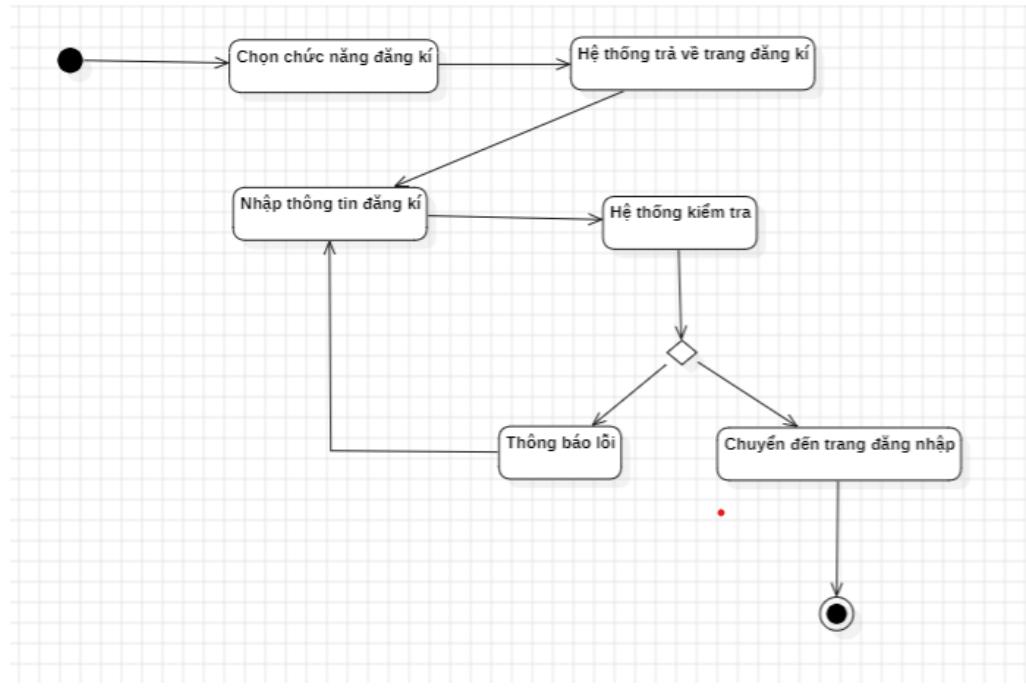
## 2.4 Biểu đồ hoạt động

### 2.4.1 Biểu đồ hoạt động cho usecase Đăng nhập



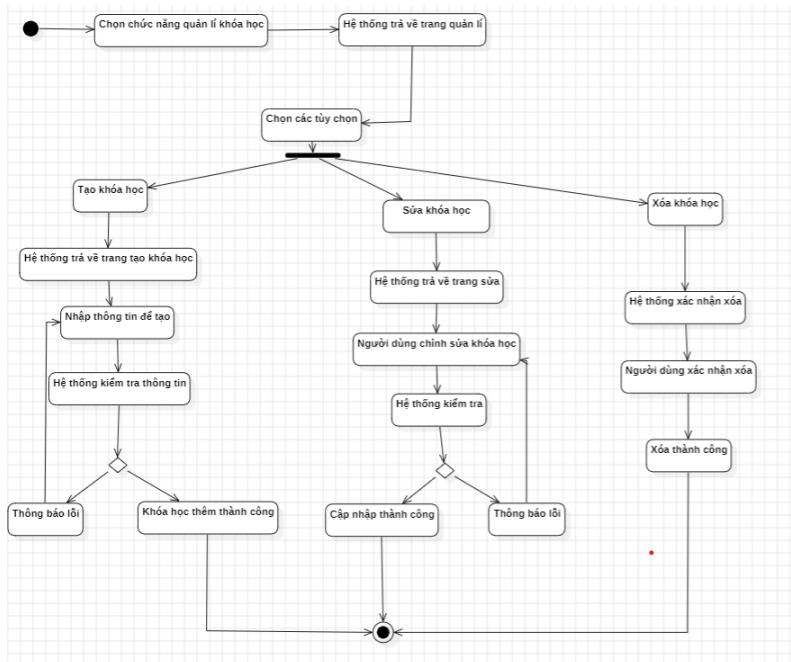
Hình 2.7: Biểu đồ hoạt động cho đăng nhập

### 2.4.2 Biểu đồ hoạt động cho usecase Đăng ký



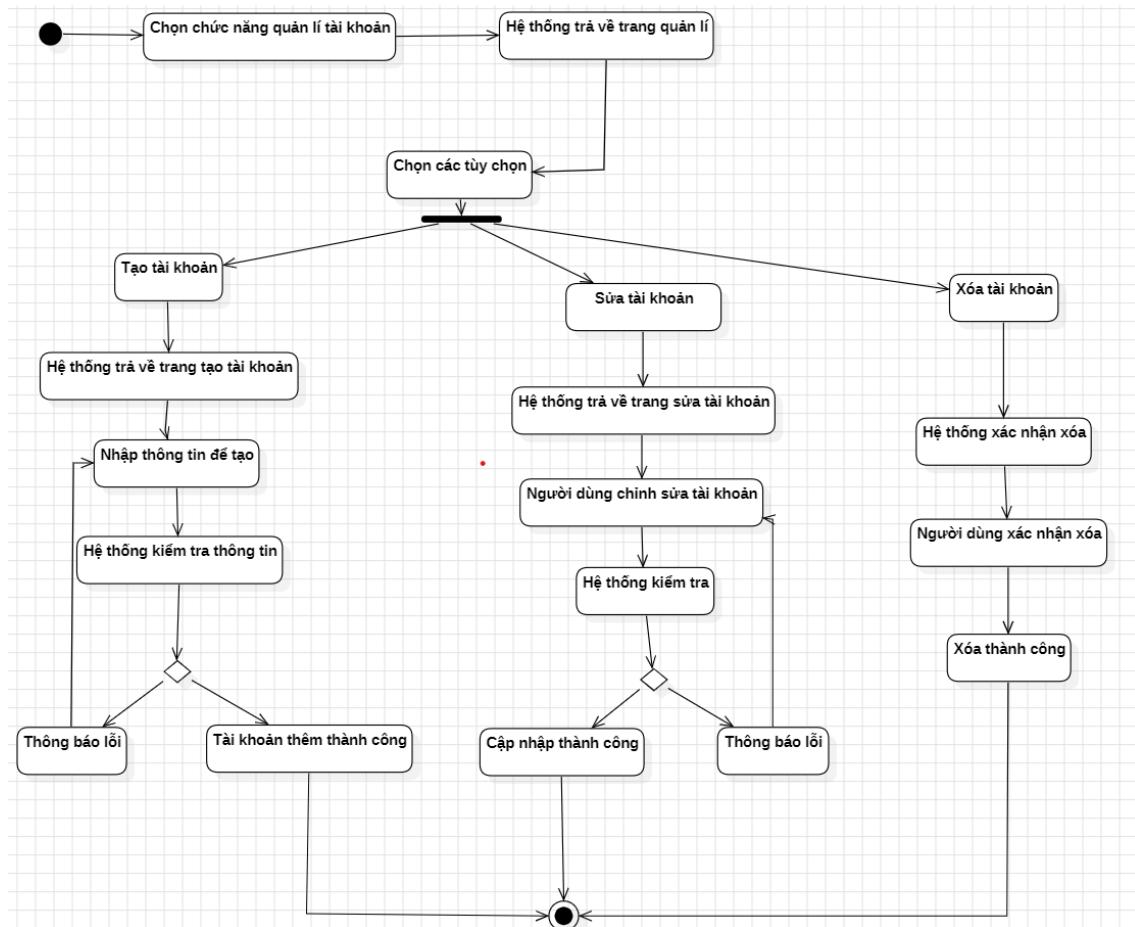
Hình 2.8: Biểu đồ hoạt động cho đăng ký

### 2.4.3 Biểu đồ hoạt động cho usecase Quản lí khóa học



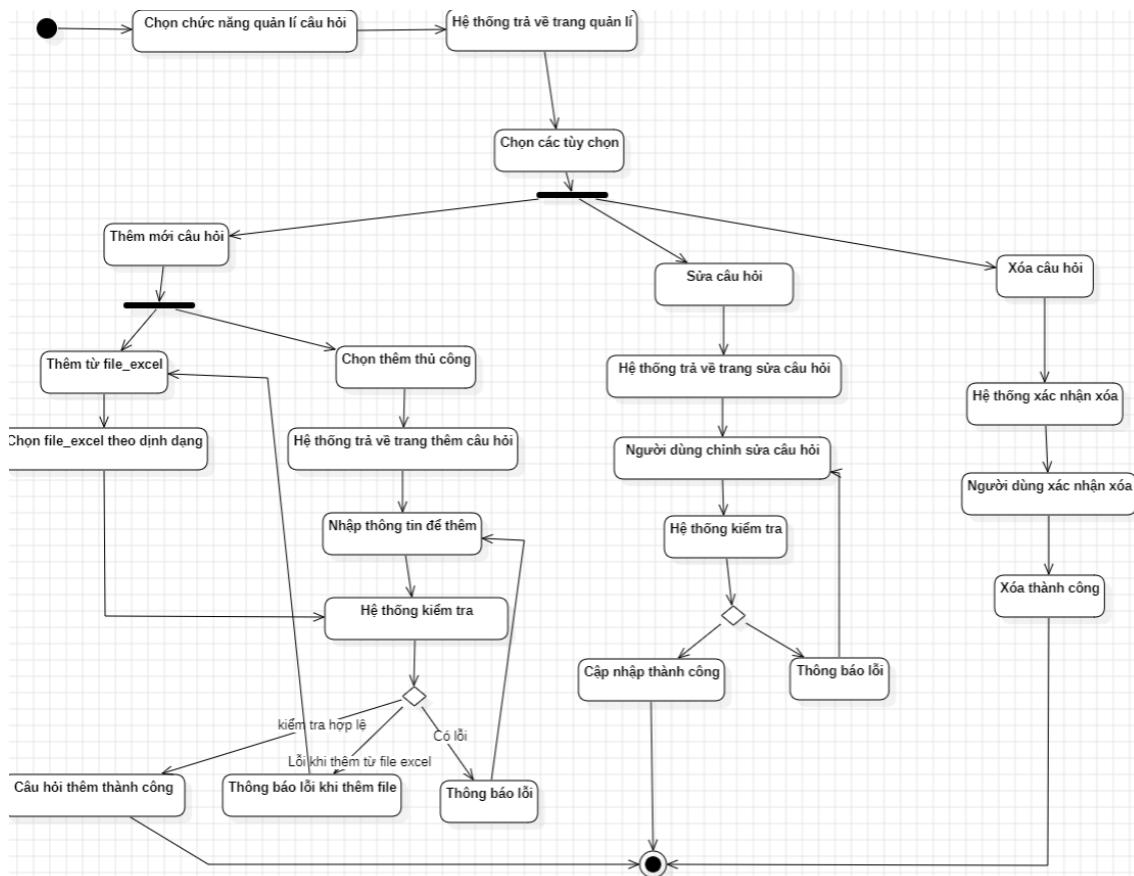
Hình 2.9: Biểu đồ hoạt động cho Quản lí khóa học

### 2.4.4 Biểu đồ hoạt động cho usecase Quản lí tài khoản



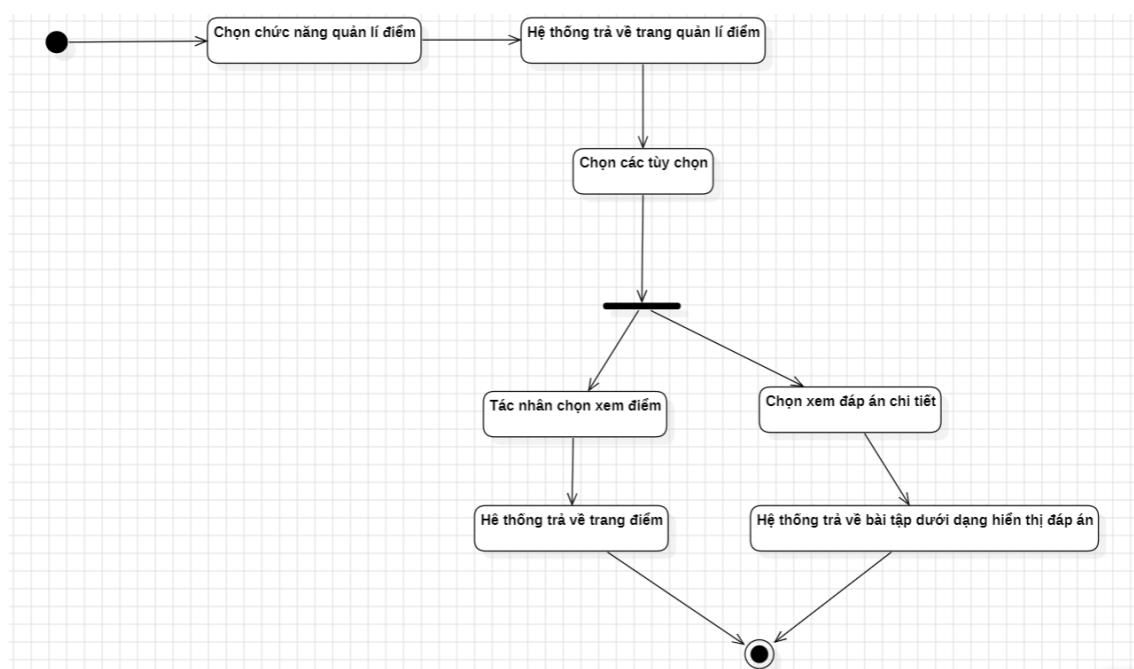
Hình 2.10: Biểu đồ hoạt động cho Quản lí tài khoản

#### 2.4.5 Biểu đồ hoạt động cho usecase Quản lý câu hỏi



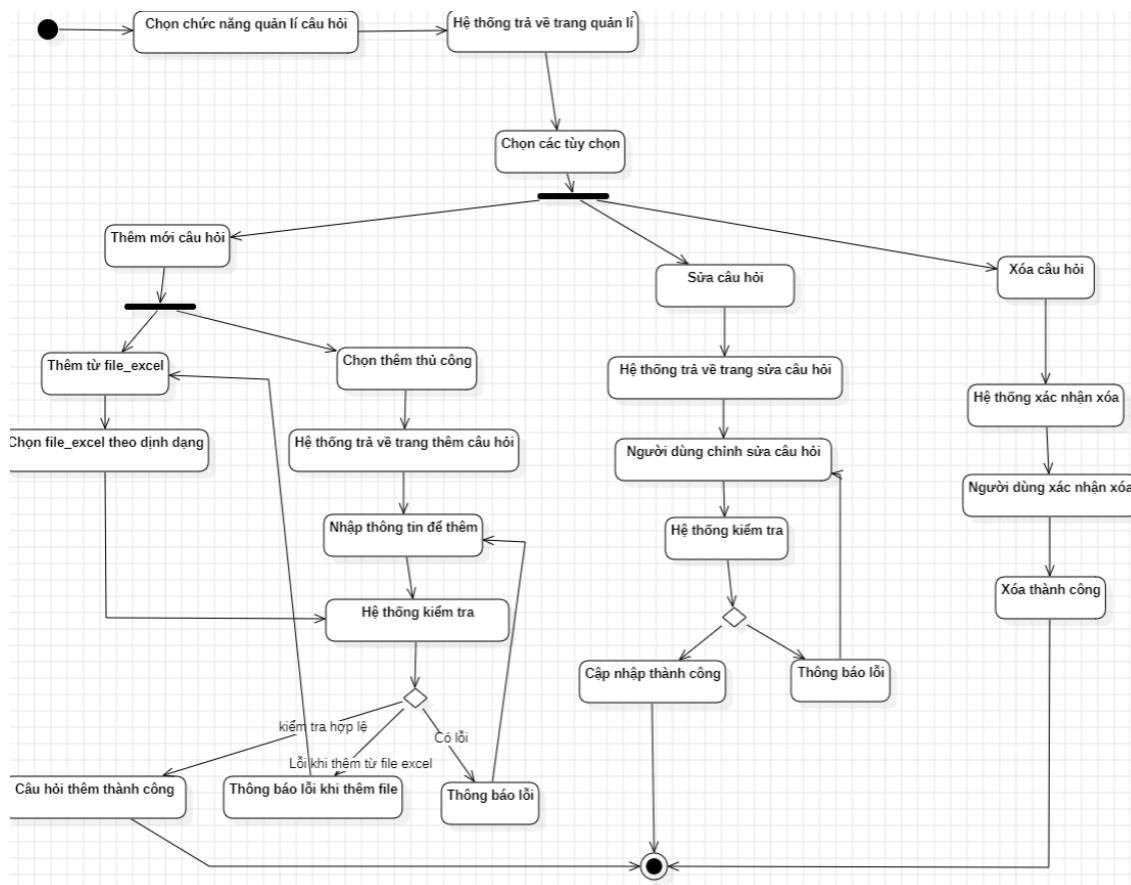
Hình 2.11: Biểu đồ hoạt động cho Quản lý câu hỏi

#### 2.4.6 Biểu đồ hoạt động cho usecase Quản lý điểm



Hình 2.12: Biểu đồ hoạt động cho Quản lý điểm

### 2.4.7 Biểu đồ hoạt động cho usecase Quản lý bài tập



Hình 2.13: Biểu đồ hoạt động cho Quản lý câu hỏi

## 2.5 Yêu cầu phi chức năng

Bảo mật: Bảo vệ thông tin người dùng, mã hóa dữ liệu nhạy cảm, và tuân thủ các quy định về bảo mật thông tin.

Hiệu suất: Hệ thống cần hoạt động ổn định, xử lý đồng thời nhiều người dùng mà không bị lag.

Khả năng mở rộng: Dễ dàng thêm người dùng, khóa học và tính năng mới trong tương lai.

Tương thích: Hệ thống cần tương thích với nhiều trình duyệt và thiết bị (máy tính, tablet, điện thoại).

## CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

Các công nghệ sử dụng phát triển hệ thống: Frontend(Giao diện): HTML, CSS, Javascript và một số thư viện khác. Backend : sử dụng ngôn ngữ PHP để phát triển hệ thống và framework Laravel. Và sử dụng SQL Server để lưu trữ và quản lý dữ liệu cho hệ thống.

### 3.1 HTML (HyperText Markup Language)

HTML là ngôn ngữ đánh dấu được sử dụng để tạo cấu trúc và nội dung cho trang web. Nó bao gồm các thẻ (tags) và thuộc tính (attributes) để định nghĩa các phần tử như tiêu đề, đoạn văn, hình ảnh, liên kết, và nhiều hơn nữa.

### 3.2 CSS (Cascading Style Sheets)

CSS là ngôn ngữ dùng để mô tả cách hiển thị của các phần tử HTML trên trang web. Nó cho phép bạn thay đổi màu sắc, phông chữ, bố cục và nhiều thuộc tính khác của các phần tử HTML.

### 3.3 JavaScript

JavaScript là ngôn ngữ lập trình được sử dụng để tạo ra các tương tác động trên trang web. Nó cho phép bạn thay đổi nội dung, xử lý sự kiện, và tương tác với người dùng mà không cần tải lại trang.

### 3.4 Ngôn ngữ PHP

PHP(Hypertext Preprocessor) là một ngôn ngữ lập trình kịch bản phía máy chủ, được sử dụng rộng rãi trong phát triển web. PHP được tạo ra bởi Rasmus Lerdorf vào năm 1994. Ban đầu, nó được gọi là Personal Home Page, nhưng sau đó đã được đổi tên thành Hypertext Preprocessor2.

Cấu trúc cơ bản: PHP cung cấp nhiều hàm tích hợp sẵn để xử lý dữ liệu. Có thể tạo hàm tùy chỉnh bằng cách sử dụng từ khóa function. Câu lệnh điều kiện: Sử dụng các câu lệnh if, else, elseif để kiểm tra điều kiện và thực hiện các hành động tương ứng. Vòng lặp: PHP hỗ trợ các vòng lặp for, while, do-while và foreach để lặp qua các phần tử của mảng hoặc thực hiện các hành động lặp lại. PHP thường được sử dụng để phát triển các ứng dụng web động và tương tác với cơ sở dữ liệu.

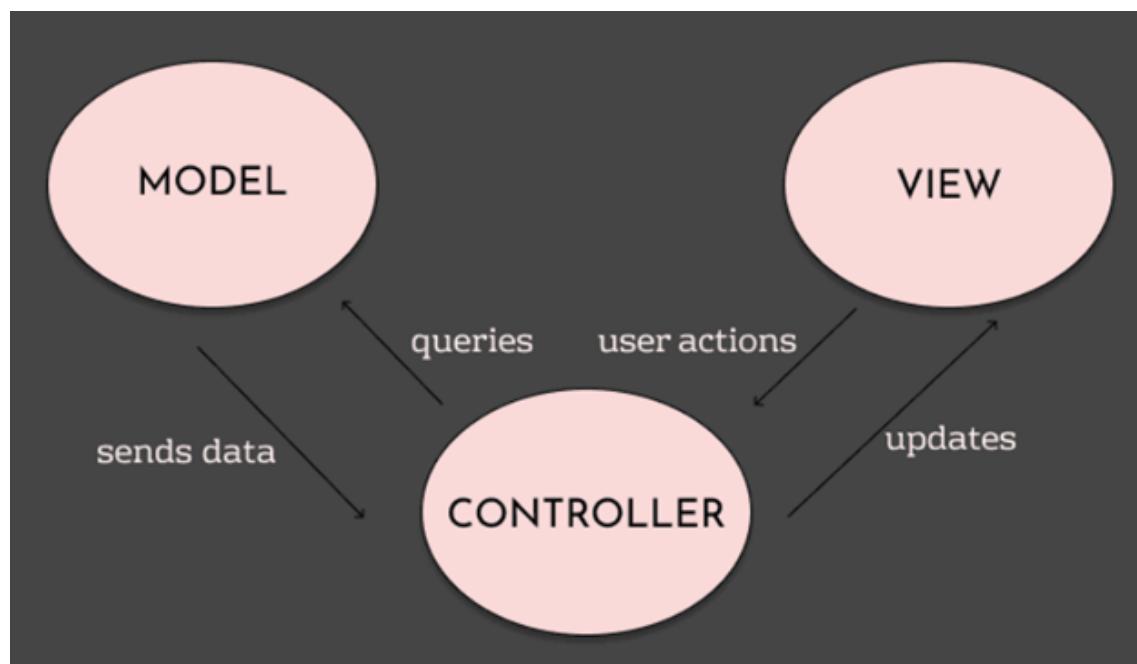
Một số ứng dụng phổ biến của PHP bao gồm: Phát triển website: PHP có thể nhúng vào HTML để tạo ra các trang web động. Quản lý cơ sở dữ liệu: PHP thường được sử dụng cùng với MySQL để quản lý cơ sở dữ liệu. Xử lý biểu mẫu: PHP có thể xử lý dữ liệu từ các biểu mẫu HTML và lưu trữ chúng vào cơ sở dữ liệu. Tạo API: PHP có thể được sử dụng để tạo các API RESTful cho các ứng dụng web và

di động.

**Ưu Điểm** của PHP: Mã nguồn mở và miễn phí: PHP là ngôn ngữ mã nguồn mở và miễn phí, giúp tiết kiệm chi phí phát triển. Dễ học và sử dụng: PHP có cú pháp đơn giản và dễ hiểu, phù hợp cho cả người mới bắt đầu và lập trình viên có kinh nghiệm. Tương thích cao: PHP tương thích với nhiều hệ điều hành như Windows, MacOS, và Linux. Cộng đồng lớn: PHP có một cộng đồng lớn và năng động, cung cấp nhiều tài liệu, hướng dẫn và hỗ trợ.

**Nhược Điểm** của PHP: PHP có thể gặp phải các vấn đề bảo mật nếu không được cấu hình và lập trình đúng cách. PHP có thể không hiệu quả bằng một số ngôn ngữ lập trình khác trong các ứng dụng yêu cầu hiệu năng cao.

### 3.5 Mô hình MVC



**Hình 3.1:** Mô hình MVC

**Model:** Quản lý dữ liệu và logic nghiệp vụ của ứng dụng. Model tương tác với cơ sở dữ liệu để truy xuất và lưu trữ dữ liệu.

**View** là các tệp HTML/CSS/JavaScript được sử dụng để tạo giao diện người dùng. Hiển thị dữ liệu cho người dùng.

**Controller:** Xử lý các yêu cầu từ người dùng, tương tác với Model để lấy dữ liệu và trả về View để hiển thị.

### 3.6 Laravel Framework

Laravel là một framework PHP mã nguồn mở, được thiết kế để phát triển các ứng dụng web theo mô hình MVC (Model-View-Controller).

#### Các tính năng chính của Laravel

Routing: Laravel cung cấp một hệ thống định tuyến mạnh mẽ, cho phép bạn định nghĩa các URL và ánh xạ chúng tới các hành động cụ thể trong Controller.

Eloquent ORM: Hệ thống ORM (Object-Relational Mapping) của Laravel, giúp tương tác với cơ sở dữ liệu một cách dễ dàng và trực quan.

#### Blade Templating

Engine: Công cụ tạo mẫu Blade của Laravel cho phép bạn sử dụng các cấu trúc điều khiển và biến trong các tệp HTML.

Middleware: Cho phép bạn lọc các yêu cầu HTTP vào ứng dụng của mình, ví dụ như xác thực người dùng hoặc kiểm tra quyền truy cập.

Artisan CLI: Công cụ dòng lệnh của Laravel, cung cấp nhiều lệnh hữu ích để phát triển và quản lý ứng dụng.

#### Ưu điểm của Laravel:

Dễ học và sử dụng: Laravel có cú pháp rõ ràng và dễ hiểu, giúp lập trình viên nhanh chóng làm quen và phát triển ứng dụng.

Cộng đồng lớn: Laravel có một cộng đồng lớn và năng động, cung cấp nhiều tài liệu, hướng dẫn và hỗ trợ.

Tích hợp tốt với các công cụ khác: Laravel tích hợp tốt với các công cụ như Redis, Memcached, và các dịch vụ email, giúp bạn xây dựng ứng dụng nhanh chóng và hiệu quả.

### 3.7 Hệ quản trị cơ sở dữ liệu SQL

Server SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft. Nó hỗ trợ nhiều tính năng mạnh mẽ để quản lý và truy vấn dữ liệu.

Cấu Trúc Cơ Bản: Database: (Là một tập hợp các bảng, chỉ mục, và các đối tượng khác liên quan đến dữ liệu), Table: (Là một tập hợp các hàng và cột, nơi dữ liệu được lưu trữ), Column: (Là một trường trong bảng, đại diện cho một loại dữ liệu cụ thể), Row: (Là một bản ghi trong bảng, chứa dữ liệu cho một mục cụ thể). Các loại dữ liệu: SQL Server hỗ trợ nhiều loại dữ liệu khác nhau như: INT(Số nguyên), VARCHAR(Chuỗi ký tự có độ dài thay đổi), DATETIME(Ngày và giờ), FLOAT(Số thực). Một số khái niệm nâng cao như: Stored Procedures: (Là các tập lệnh SQL được lưu trữ và có thể tái sử dụng), Triggers: Là các thủ tục tự động thực thi khi có sự kiện cụ thể xảy ra trong cơ sở dữ liệu, Views: (Là các truy vấn SQL

được lưu trữ dưới dạng bảng ảo), Indexes: (Là các cấu trúc dữ liệu giúp tăng tốc độ truy vấn dữ liệu)

Các Phép Toán Tập Hợp: SQLServer hỗ trợ các phép toán tập hợp như UNION, INTERSECT và EXCEPT, giúp kết hợp và so sánh các tập dữ liệu2. Quản Lý Giao Dịch: SQL Server cung cấp các tính năng quản lý giao dịch để đảm bảo tính toàn vẹn và nhất quán của dữ liệu. Các giao dịch có thể được bắt đầu, cam kết hoặc hủy bỏ bằng các lệnh BEGIN TRANSACTION, COMMIT, và ROLLBACK.

Bảo Mật: SQL Server cung cấp nhiều cơ chế bảo mật để bảo vệ dữ liệu, mã hóa dữ liệu.

Hiệu Năng và Tối Ưu Hóa: SQL Server cung cấp nhiều công cụ và tính năng để tối ưu hóa hiệu năng, bao gồm lập chỉ mục, tối ưu hóa truy vấn, và quản lý bộ nhớ.

Sao Lưu và Phục Hồi: SQL Server hỗ trợ các tính năng sao lưu và phục hồi dữ liệu để đảm bảo an toàn dữ liệu trong trường hợp xảy ra sự cố.

### 3.8 AJAX

AJAX (Asynchronous JavaScript and XML) là một kỹ thuật phát triển web cho phép các trang web cập nhật nội dung một cách động mà không cần tải lại toàn bộ trang.

Cách hoạt động của AJAX: AJAX hoạt động bằng cách gửi các yêu cầu HTTP không đồng bộ tới máy chủ và nhận phản hồi từ máy chủ mà không làm gián đoạn trải nghiệm người dùng. Quá trình này bao gồm các bước sau: Sự kiện người dùng: Người dùng thực hiện một hành động trên trang web. Gửi yêu cầu: JavaScript gửi một yêu cầu HTTP không đồng bộ tới máy chủ. Xử lý yêu cầu: Máy chủ xử lý yêu cầu và trả về dữ liệu (thường là XML, JSON hoặc HTML). Cập nhật trang: JavaScript nhận dữ liệu từ máy chủ và cập nhật nội dung trang web mà không cần tải lại toàn bộ trang.

Các Thành Phần Chính của AJAX: XMLHttpRequest: Đối tượng này được sử dụng để gửi các yêu cầu HTTP tới máy chủ và nhận phản hồi. JavaScript: Ngôn ngữ lập trình được sử dụng để thao tác với đối tượng XMLHttpRequest và cập nhật nội dung trang web. Dữ liệu: Dữ liệu được trao đổi giữa máy chủ và trình duyệt, thường ở định dạng XML, JSON hoặc HTML.

Ưu điểm của AJAX Tăng trải nghiệm người dùng: AJAX cho phép cập nhật nội dung trang web một cách mượt mà mà không cần tải lại toàn bộ trang. Tiết kiệm băng thông: Chỉ các phần dữ liệu cần thiết được trao đổi giữa máy chủ và trình duyệt, giúp tiết kiệm băng thông. Tăng hiệu suất: Giảm thời gian tải trang và tăng

hiệu suất của ứng dụng web.

Nhược điểm của AJAX: Phụ thuộc vào JavaScript: AJAX yêu cầu JavaScript để hoạt động, do đó nếu người dùng tắt JavaScript, ứng dụng sẽ không hoạt động đúng cách. Vấn đề bảo mật: AJAX có thể gặp phải các vấn đề bảo mật như tấn công XSS (Cross-Site Scripting) nếu không được cấu hình đúng cách.

Việc kết hợp sử dụng HTML, CSS Javascript cùng với các thư viện khác để tạo phần giao diện cho hệ thống cùng với Laravel để phát triển hệ thống theo mô hình MVC. Với các tính năng mạnh mẽ, laravel giúp phát triển nhanh chóng và hiệu quả. Cơ sở dữ liệu SQL hỗ trợ việc lưu trữ dữ liệu với nhiều tính năng hỗ trợ. Sử dụng AJAX gửi yêu cầu như xóa các đối tượng, hỗ trợ chức năng chat realtime mà không phải tải lại trang web giúp trải nghiệm người dùng tốt hơn và tiết kiệm băng thông.

## **CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG**

### **4.1 Thiết kế kiến trúc**

#### **4.1.1 Lựa chọn kiến trúc phần mềm**

Kiến trúc MVC (Model-View-Controller) là một phương pháp thiết kế phổ biến bởi khả năng tách biệt rõ ràng giữa các thành phần khác nhau của ứng dụng. Điều này không chỉ giúp cho việc phát triển và bảo trì ứng dụng trở nên dễ dàng hơn mà còn giúp cho hoạt động làm việc nhóm được hiệu quả hơn bởi mỗi thành viên có thể được chia theo các công việc rõ ràng. Trong đồ án tốt nghiệp này, em sẽ áp dụng kiến trúc MVC để xây dựng một hệ thống học tập trực tuyến, với mục tiêu mang lại lợi ích 2 phía cả về đáp ứng yêu cầu người dùng và cả về việc mở rộng, hay bảo trì hệ thống.

Kiến trúc MVC được chia thành ba thành phần chính: Model, View và Controller. Mỗi thành phần có vai trò và chức năng riêng biệt, giúp tách biệt các khía cạnh khác nhau của ứng dụng, từ đó tăng tính linh hoạt và khả năng mở rộng.

#### **Model**

Model chịu trách nhiệm quản lý dữ liệu và logic nghiệp vụ của hệ thống. Model tương tác với cơ sở dữ liệu để truy xuất và lưu trữ dữ liệu, đồng thời thực hiện các phép tính và xử lý logic cần thiết.

#### **Vai trò của Model:**

**Quản lý dữ liệu:** Model lưu trữ và quản lý dữ liệu của hệ thống. Nó có thể bao gồm các đối tượng dữ liệu, các lớp (classes) và các phương thức (methods) để thao tác với dữ liệu.

**Tương tác với cơ sở dữ liệu:** Model thực hiện các thao tác CRUD (Create, Read, Update, Delete) để tương tác với cơ sở dữ liệu. Nó sử dụng các truy vấn SQL hoặc ORM(Object-Relational Mapping) để truy xuất và lưu trữ dữ liệu.

**Xử lý logic nghiệp vụ:** Model chứa các quy tắc và logic nghiệp vụ của ứng dụng. Nó thực hiện các phép tính, kiểm tra điều kiện và xử lý dữ liệu trước khi gửi đến View hoặc Controller.

#### **Cấu trúc của Model:**

**Entities:** Các đối tượng dữ liệu đại diện cho các bảng trong cơ sở dữ liệu. Mỗi entity thường tương ứng với một bảng và chứa các thuộc tính (fields) đại diện cho các cột trong bảng.

**Repositories:** Các lớp hoặc đối tượng chịu trách nhiệm truy xuất và lưu trữ dữ liệu từ cơ sở dữ liệu. Repositories cung cấp các phương thức để thực hiện các thao tác CRUD.

**Services:** Các lớp hoặc đối tượng chứa logic nghiệp vụ của ứng dụng. Services sử dụng các repositories để truy xuất và lưu trữ dữ liệu, đồng thời thực hiện các phép tính và xử lý logic cần thiết. Những model cơ bản của hệ thống triển khai như Course(Khoa học), User(người dùng), Exercise(Bài tập), Question(Câu hỏi), Section(Tài liệu khóa học)

### View

View chịu trách nhiệm hiển thị dữ liệu cho người dùng. View là các tệp HTML, CSS, và JavaScript được sử dụng để tạo giao diện người dùng. Nó nhận dữ liệu từ Controller và hiển thị dữ liệu đó dưới dạng dễ hiểu và dễ tương tác. Ngoài ra, trong Laravel framework các view được tạo ra với đuôi blade.php. Blade là một công cụ tạo mẫu (templating engine) mạnh mẽ và linh hoạt được tích hợp sẵn trong Laravel. Blade cho phép bạn sử dụng các cấu trúc điều khiển và biến trong các tệp HTML một cách dễ dàng và hiệu quả.

#### Vai trò của View:

**Hiển thị dữ liệu:** View nhận dữ liệu từ Controller và hiển thị dữ liệu đó cho người dùng. Dữ liệu có thể là văn bản, hình ảnh, biểu đồ, hoặc bất kỳ loại thông tin nào khác.

**Tương tác với người dùng:** View cung cấp các phần tử giao diện như biểu mẫu, nút bấm, và liên kết để người dùng tương tác với ứng dụng. **Tách biệt giao diện và logic nghiệp vụ:** View giúp tách biệt giao diện người dùng khỏi logic nghiệp vụ, giúp dễ dàng bảo trì và mở rộng ứng dụng.

#### Cấu trúc của View:

View bao gồm các tệp HTML, CSS, Javascript nên cấu trúc của nó được tạo ra từ các tệp này. HTML giúp tạo cấu trúc cơ bản của trang web, bao gồm các phần tử như tiêu đề, đoạn văn, hình ảnh, và biểu mẫu. CSS giúp tạo kiểu dáng cho trang web, bao gồm màu sắc, phông chữ, bố cục, và các hiệu ứng. JavaScript giúp tạo các tương tác động trên trang web, chẳng hạn như xử lý sự kiện, xác thực biểu mẫu, và cập nhật nội dung mà không cần tải lại trang.

Trong hệ thống đang triển khai sẽ có các view bao gồm các trang như trang quản lí khóa học(quản trị), quản lí người dùng(quản trị), trang chủ, danh sách khóa học

### Controller

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Controller đóng vai trò trung gian giữa model và view giúp tách biệt logic nghiệp vụ và giao diện người dùng. Nó chịu trách nhiệm xử lý các yêu cầu từ người dùng từ view, tương tác với Model để lấy dữ liệu và trả về View để hiển thị.

Vai trò của Controller:

Xử lý yêu cầu từ người dùng: Controller nhận các yêu cầu HTTP từ người dùng (ví dụ: GET, POST, PUT, DELETE) và xác định hành động cần thực hiện.

Tương tác với Model: Controller gọi các phương thức của Model để truy xuất, lưu trữ hoặc cập nhật dữ liệu. Trả về View: Sau khi lấy dữ liệu từ Model, Controller trả về View để hiển thị dữ liệu cho người dùng.

Cấu trúc của Controller:

Phương thức (Methods): Controller chứa các phương thức để xử lý các yêu cầu cụ thể từ người dùng. Mỗi phương thức thường tương ứng với một hành động cụ thể (ví dụ: hiển thị danh sách, thêm mới, cập nhật, xóa).

Định tuyến (Routing): Controller được ánh xạ tới các URL cụ thể thông qua hệ thống định tuyến của framework. Điều này cho phép người dùng truy cập các hành động của Controller thông qua các URL tương ứng. Trong hệ thống đang triển khai sẽ có các controller như CourseController(Thêm(), chỉnh sửa(), xóa(), tìm kiếm khóa học()), UserController(Thêm(), cập nhập thông tin(), xóa(), tìm kiếm người dùng())

Lợi ích của Kiến trúc MVC

Tách biệt trách nhiệm: Giúp tách biệt các phần khác nhau của ứng dụng, làm cho việc phát triển và bảo trì dễ dàng hơn.

Tái sử dụng mã nguồn: Các thành phần có thể được tái sử dụng trong nhiều phần khác nhau của ứng dụng.

Dễ dàng kiểm thử: Từng thành phần có thể được kiểm thử một cách độc lập, giúp nâng cao chất lượng phần mềm.

Phát triển song song: Các nhà phát triển có thể làm việc trên Model, View, và Controller cùng một lúc mà ít ảnh hưởng lẫn nhau

### 4.1.2 Thiết kế gói

Với gói User: mục đích chứa các đối tượng người dùng trong hệ thống như quản trị, giáo viên, người học giúp quản lý thông tin người dùng và xác thực người dùng.

Course package: quản lý thông tin khóa học và các thành phần liên quan đến khóa học như nội dung tài liệu, bài tập, câu hỏi

Role package: quản lý thông tin về vai trò của từng loại người dùng trong hệ thống mỗi vai trò được gắn với các quyền tương ứng.

Payment package: quản lý việc thanh toán khóa học nếu khóa học trả phí giúp giáo viên và quản trị viên nắm được thông tin về nguồn thu từ khóa học.

Chat package: quản lý việc trao đổi thông tin giữa người dùng cũng như trong khóa học và phần giao tiếp 2 chiều giữa người học và giáo viên.

Score package: giúp lưu thông tin về điểm các lần làm của người học và kèm theo đáp án tương ứng. Giáo viên có thể quản lý điểm của người học và người học cũng theo dõi được tiến độ học của bản thân.

### 4.2 Thiết kế chi tiết

#### 4.2.1 Thiết kế giao diện

Trong quá trình phát triển hệ thống, việc thiết kế giao diện người dùng (UI) đóng vai trò quan trọng nhằm mang lại trải nghiệm tốt nhất cho người dùng. Thiết kế UI không chỉ đảm bảo tính thẩm mỹ mà còn phải đáp ứng và tính dễ sử dụng, nhất quán và thân thiện với người dùng. Các vấn đề trong thiết kế giao diện như thông số kỹ thuật của màn hình mục tiêu mà hệ thống hướng tới, từ độ phân giải, kích thước màn hình, số lượng màu sắc, quy tắc về nút bấm, điều khiển, vị trí hiển thị thông điệp phản hồi... Những thông số này giúp định hướng quá trình thiết kế.

Đặc tả thông tin về màn hình:

Độ phân giải màn hình: Được thiết kế hướng tập trung trên Desktop/Laptop với độ phân giải Full HD (1920x1080 pixels)

Kích thước màn hình: Màn hình từ 13 inch đến 27 inch

Số lượng màu sắc hỗ trợ: Hệ thống sẽ hỗ trợ hiển thị hầu như số lượng màu để đảm bảo chất lượng hình ảnh và giao diện người dùng sắc nét.

Chuẩn hóa thiết kế giao diện:

Thiết kế nút: Hình chữ nhật với góc bo tròn, chiều cao và chiều rộng tùy thuộc vào nội dung.

Điều khiển: Các thanh cuộn với màu nền đen, phía bên phải, hoạt động khi quá chiều cao tối đa, Các form với chiều cao tự động theo nội dung.

Vị trí hiển thị thông điệp phản hồi: Thông điệp sẽ được hiển thị ở phía trên và ở giữa màn hình.

Phối màu: Sử dụng chủ yếu nền trắng để nổi bật nội dung và có thể đổi qua nền đen.

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

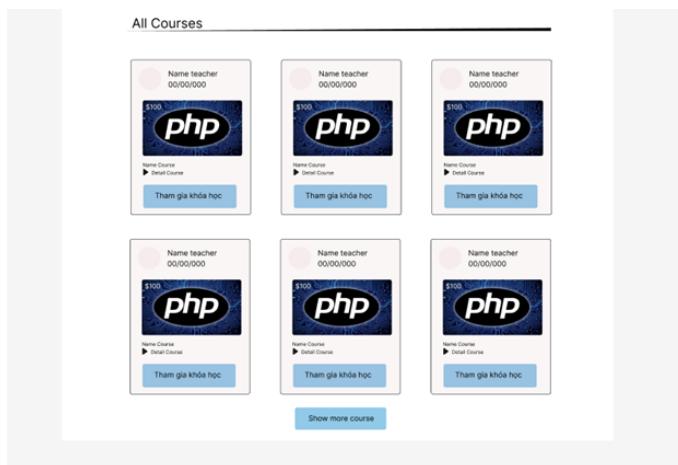
\* Thiết kế trang Đăng nhập

The image shows a login form with a light gray background. At the top, it says "Tên đăng nhập" (Username) and has a text input field with placeholder text "Tên đăng nhập". Below that is a "Mật khẩu" (Password) field with placeholder text "Mật khẩu". To the right of the password field are two buttons: "Ghi nhớ đăng nhập" (Remember login) with a checkbox and "Quên mật khẩu" (Forgot password). At the bottom center is a green "Đăng nhập" (Login) button. Below the button, there's a link "Chưa có tài khoản? [Đăng ký](#)" (Don't have an account? [Sign up](#)) in blue.

**Hình 4.1:** Thiết kế trang đăng nhập

Trang đăng nhập được thiết kế với các trường nhập cho tên đăng nhập, mật khẩu và một nút để gửi yêu cầu đăng nhập. Ngoài ra còn có ô check ghi nhớ đăng nhập và liên kết đến trang đăng kí nếu chưa có tài khoản và 1 liên kết đến chức năng quên mật khẩu.

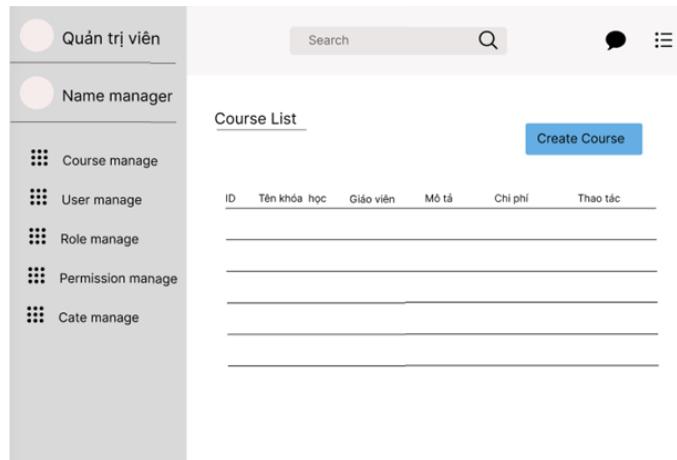
\* Thiết kế trang danh sách khóa học



**Hình 4.2:** Thiết kế danh sách khóa học

Giao diện hiển thị danh sách các khóa học hiện có và các thông tin đi kèm trong 1 thẻ khóa học gồm ảnh đại diện của giáo viên và ảnh đại diện khóa học kèm học phí khóa học (nếu có) gắn kèm ở góc trái của ảnh cùng với tên khóa học và mô tả về khóa học và có một nút để gửi yêu cầu đăng kí khóa học.

\* Thiết kế trang quản lý khóa học



**Hình 4.3:** Thiết kế trang quản lý khóa học

Giao diện gồm 3 phần: thanh bên trái gồm avatar và vai trò của người đăng nhập là quản trị hay giáo viên cùng với avatar và tên của người đăng nhập và ở dưới là danh sách các chức năng của người quản lí có thể chọn để sử dụng, Thanh ở trên gồm ôtìm kiếm như tìm kiếm khóa học hoặc giáo viên và 2 biểu tượng tin nhắn và ô menu thu gọn. Phần còn lại chứa nội dung tương ứng với chức năng đã chọn.

#### \* Thiết kế trang Tạo khóa học

 A screenshot of a 'Create New Course' form. The sidebar on the left is identical to the one in Figure 4.3. The main form has fields for 'Tên khóa học' (Input: Nhập tên khóa học), 'Mô Tả Khóa học' (Input: Mô tả), 'Giáo viên quản lý' (Input: Thêm giáo viên quản lý), 'Ảnh đại diện' (File input: Chọn File, selected file: avatar.png), and 'Chi phí khóa học(nếu có)' (Input: Nhập phí khóa học). At the bottom is a green 'Tạo khóa học' (Create course) button.

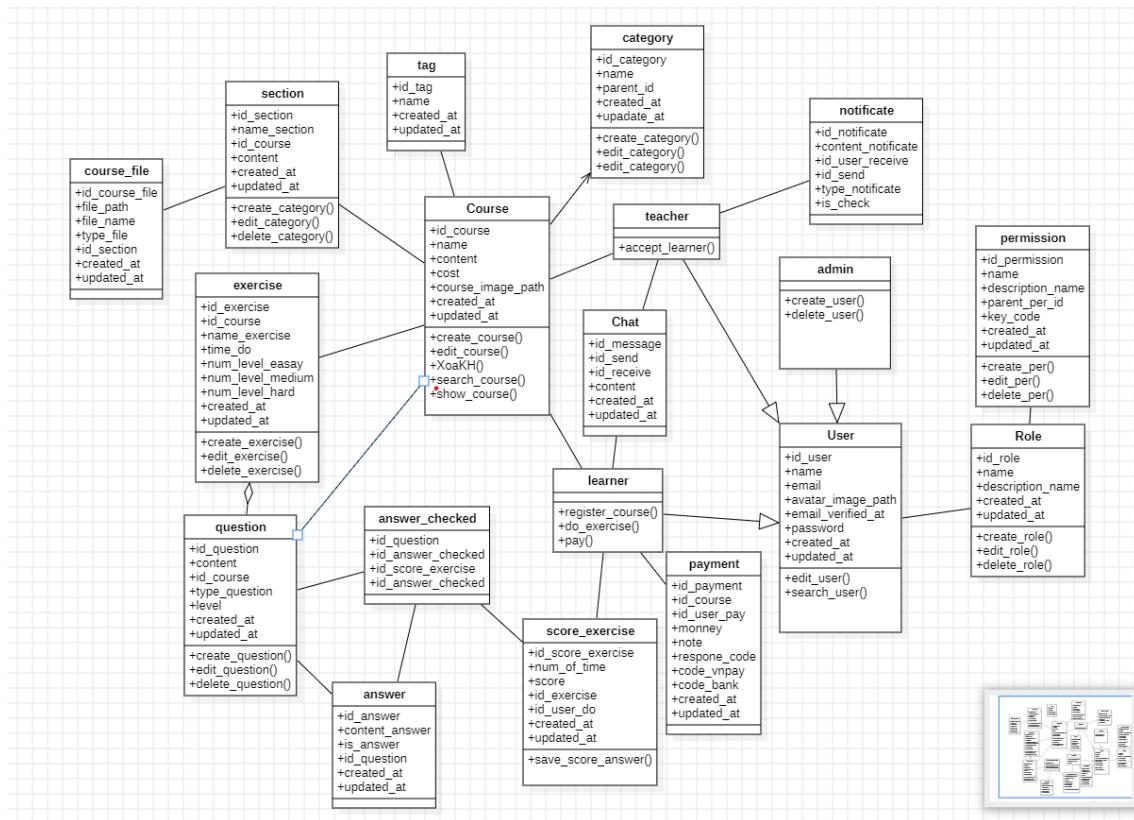
**Hình 4.4:** Thiết kế trang Tạo khóa học

Giao diện tạo khóa học bao gồm form với các trường cần thiết để tạo một khóa học và 1 nút để gửi yêu cầu tạo khóa học

#### 4.2.2 Thiết kế lớp

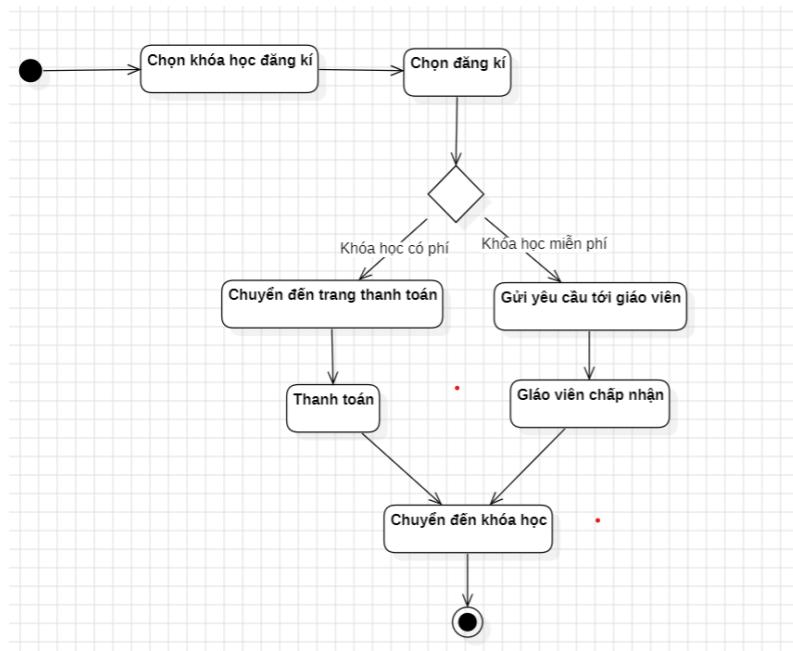
Minh họa một số phương thức qua biểu đồ hoạt động

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



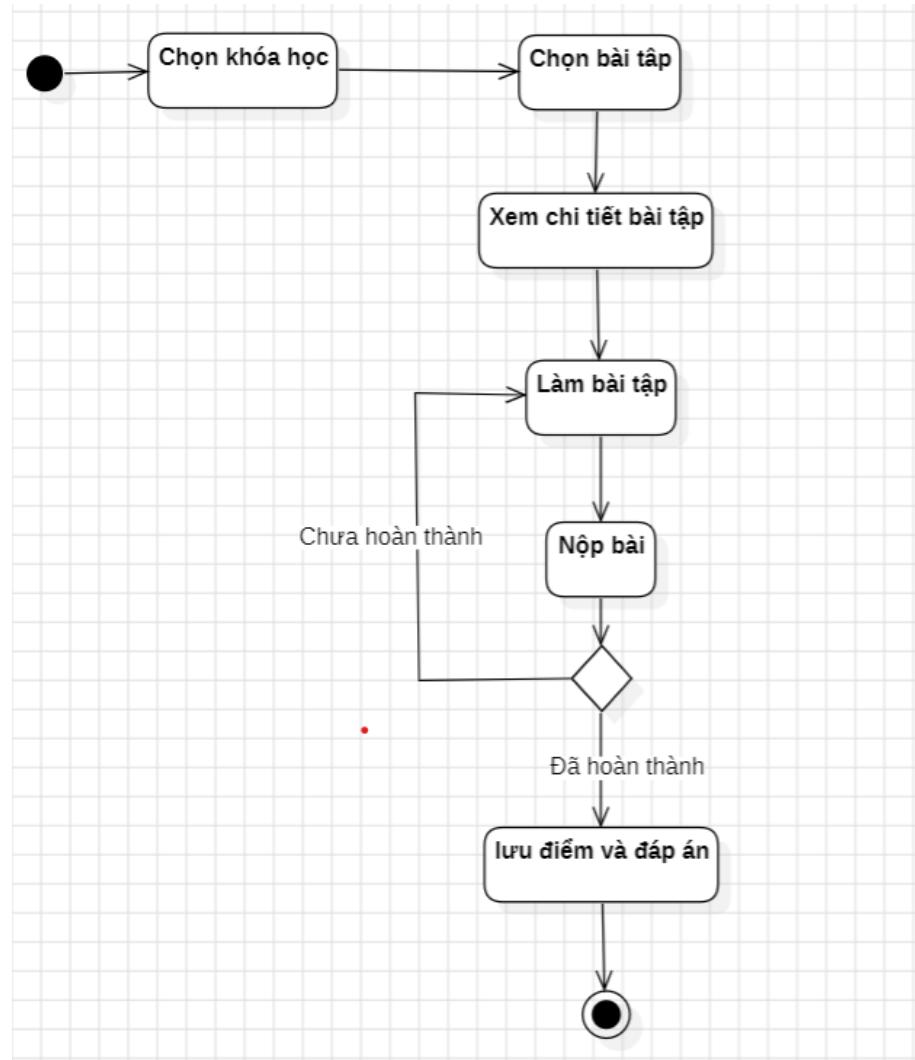
**Hình 4.5:** Biểu đồ lớp cho hệ thống

\* Phương thức đăng kí khóa học: Người học chọn khóa học và gửi yêu cầu đăng kí. Nếu khóa học miễn phí thì yêu cầu được gửi đến giáo viên quản lí để xác nhận. Nếu khóa học có phí thì gọi phương thức thanh toán khóa học. Sau khi đăng kí thành công thì người học sẽ được thêm vào khóa học



**Hình 4.6:** Biểu đồ hoạt động đăng kí khóa học

\* Phương thức làm bài tập: Người học sau khi được thêm vào khóa học và chọn chức năng làm bài tập. Hệ thống chuyển đến giao diện bài tập tương ứng. Người học sau khi làm bài và chọn nộp bài. Hệ thống xác nhận xem người học đã hoàn thành hết các câu hỏi chưa. Nếu đã hoàn thành thì xác nhận nộp bài thành công và tiến hành nộp bài và lưu điểm. Nếu chưa thì yêu cầu hoàn thành tiếp



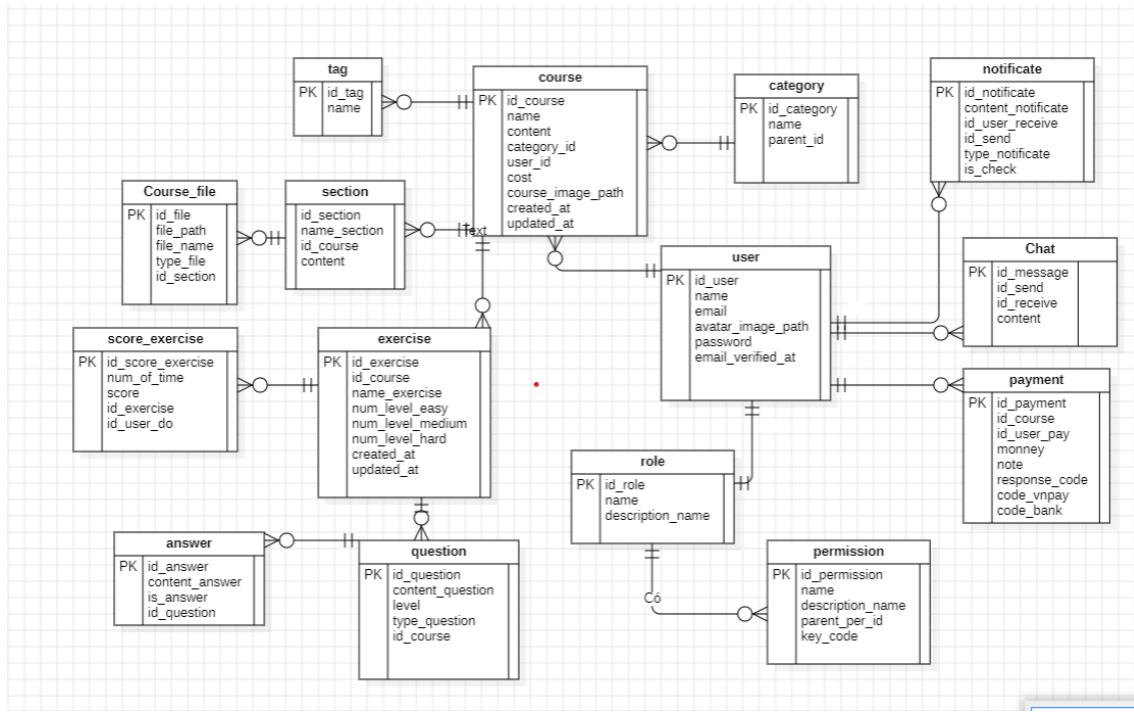
**Hình 4.7:** Biểu đồ hoạt động làm bài tập

#### 4.2.3 Thiết kế cơ sở dữ liệu

\*Biểu đồ thực thể liên kết

Các thực thể chủ đạo như course (khóa học) và user (người dùng). Thực thể user được phân quyền theo từng vai trò nhất, mỗi loại người dùng thì có một vai trò tương ứng. Trong mỗi vai trò thì lại có các quyền khác nhau để thao tác. Ngoài ra đối với loại người dùng là người học thì các thanh toán học phí phát sinh và mỗi người dùng có các thông báo và tin nhắn đến và đi. Đối với thực thể khóa học, mỗi khóa học thuộc về một danh mục lĩnh vực lớn và ngược lại mỗi danh mục thì bao

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



**Hình 4.8:** Biểu đồ thực thể liên kết

gồm nhiều khóa học. Về phần học liệu thì mỗi khóa học lại có nhiều phần nội dung tài liệu và các câu hỏi và bài tập tương ứng.

\* Cơ sở dữ liệu cho hệ thống

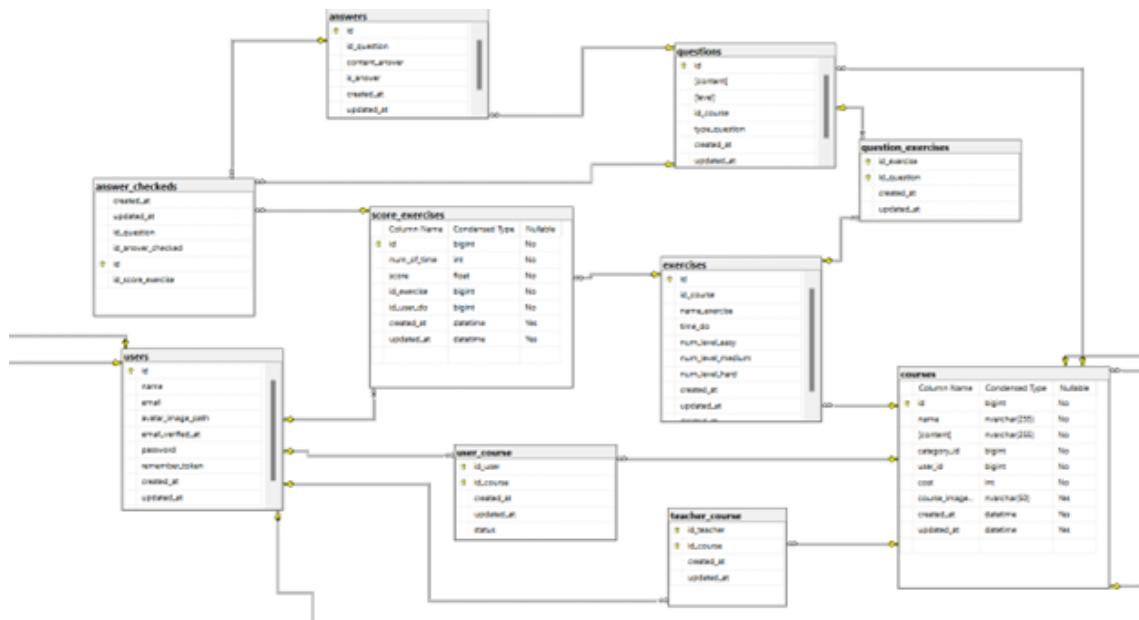


**Hình 4.9:** Lưu thông tin khóa học

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



Hình 4.10: Lưu thông tin vai trò người dùng



Hình 4.11: Lưu điểm và đáp án

\* Chi tiết các bảng

a, Bảng User

Lưu các thông tin của người dùng email, mật khẩu, tên, ảnh đại diện

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Tên cột	Kiểu dữ liệu
id	bigint
name	nvarchar(255)
email	nvarchar(255)
avatar_image_path	nvarchar(255)
email_verified_at	datetime
password	nvarchar(255)
remember_token	nvarchar(100)
created_at	datetime
updated_at	datetime

**Bảng 4.1:** Bảng người dùng

b, Bảng Course

Bảng course lưu thông tin cơ bản của khóa học như tên, mô tả khóa học, chi phí khóa học, ảnh đại diện khóa học.

Tên cột	Kiểu dữ liệu
id	bigint
name	nvarchar(255)
[content]	nvarchar(255)
category_id	bigint
user_id	bigint
cost	int
course_image_path	nvarchar(50)
created_at	datetime
updated_at	datetime

**Bảng 4.2:** Bảng khóa học

c, Bảng score\_exercise

Bảng score\_exercise lưu thông tin về điểm của các lần làm bài tập của người dùng.

Tên cột	Kiểu dữ liệu
id	bigint
num_of_time	int
score	float
id_exercise	bigint
id_user_do	bigint
created_at	datetime
updated_at	datetime

**Bảng 4.3:** Bảng lưu điểm

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

---

d, Bảng question Bảng câu hỏi lưu thông tin về các câu hỏi và có 1 bảng answer để ánh xạ sang bảng câu hỏi với khóa ngoài là id của câu hỏi.

Tên cột	Kiểu dữ liệu
id	bigint
content	nvarchar(255)
level	int
id_course	bigint
type_question	int
created_at	datetime
updated_at	datetime

**Bảng 4.4:** Bảng câu hỏi

e, Bảng role

Bảng vai trò lưu thông tin của các vai trò ứng với từng loại người dùng khác nhau.

Tên cột	Kiểu dữ liệu
id	bigint
name	nvarchar(255)
description_name	nvarchar(255)
created_at	datetime
updated_at	datetime

**Bảng 4.5:** Bảng vai trò

f, Bảng permission

Tên cột	Kiểu dữ liệu
id	bigint
name	nvarchar(255)
description_name	nvarchar(255)
parent_id	bigint
key_code	nvarchar(255)
created_at	datetime
updated_at	datetime

**Bảng 4.6:** Bảng quyền theo từng vai trò

g, Bảng section

Bảng lưu các nội dung tổng quát về nội dung của khóa học và mã của bảng section sẽ là khóa ngoài tới bảng lưu các file tương ứng.

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Tên cột	Kiểu dữ liệu
id	bigint
name_section	nvarchar(255)
id_course	bigint
content	nvarchar(1000)
created_at	datetime
updated_at	datetime

**Bảng 4.7:** Bảng nội dung khóa học

h, Bảng course\_file Bảng lưu thông tin về các file thuộc vào các khóa học tương ứng

Tên cột	Kiểu dữ liệu
id	bigint
file_path	nvarchar(255)
id_section	bigint
file_name	nvarchar(255)
type_file	varchar(20)
created_at	datetime
updated_at	datetime
deleted_at	datetime

**Bảng 4.8:** Bảng File của khóa học

i, Bảng exercise

Bảng bài tập lưu thông tin về các bài tập tương ứng với khóa học và mà bài tập sẽ là khóa ngoài ánh xạ tới bảng câu hỏi để lấy câu hỏi tương ứng.

Tên cột	Kiểu dữ liệu
id	bigint
id_course	bigint
name_exercise	nvarchar(255)
time_do	int
num_level_easy	int
num_level_medium	int
num_level_hard	int
created_at	datetime
updated_at	datetime
deleted_at	datetime

**Bảng 4.9:** Bảng bài tập

### 4.3 Xây dựng ứng dụng

#### 4.3.1 Thư viện và công cụ sử dụng

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	VS code	<a href="https://www.php.net">https://www.php.net</a>
Ngôn ngữ lập trình	PHP	<a href="https://apps.microsoft.com">https://apps.microsoft.com</a>
Thư viện hỗ trợ Laravel	Framework	<a href="https://laravel.com">https://laravel.com</a>
Gửi tin nhắn trong thời gian thực	Pusher	<a href="https://pusher.com">https://pusher.com</a>
Quản lý cơ sở dữ liệu	SQLServer management Studio	<a href="https://www.microsoft.com/en-us/sql-server">https://www.microsoft.com/en-us/sql server</a>

**Bảng 4.10:** Thư viện và công cụ sử dụng

#### 4.3.2 Kết quả đạt được

Sản phẩm của đồ án: Hệ thống học tập trực tuyến trên nền Website. Hệ thống cung cấp nền tảng trực tuyến cho người dùng có nhu cầu học tập và các giáo viên muôn tạo các khóa học online chất lượng hoặc có thể chia sẻ và trao đổi kiến thức qua những câu hỏi trong group cộng đồng. Người học có thể truy cập vào các khóa học từ bất kỳ đâu và bất kỳ lúc nào, chỉ cần có kết nối internet. Người học và giáo viên có thể chủ động hơn trong quá trình học và dạy của mình. Hệ thống giúp giảng

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

viên dễ dàng quản lý khóa học, theo dõi tiến độ học tập của học viên, và đánh giá kết quả học tập. Hệ thống có thể cung cấp các tính năng tương tác thời gian thực.

Học viên có thể học theo tốc độ của riêng mình và lựa chọn các khóa học phù hợp với nhu cầu và sở thích cá nhân. Giảm chi phí liên quan đến việc di chuyển, thuê phòng học, và in ấn tài liệu. Về mặt nghiệp vụ của hệ thống: Giao diện người dùng đơn giản, dễ sử dụng giúp tăng trải nghiệm người dùng. Đã đưa ra được các chức năng cần có so với nghiệp vụ, yêu cầu của hệ thống cần có. Về mặt dữ liệu được quản lý một cách hiệu quả đảm bảo tính nhất quán và toàn vẹn dữ liệu. Các chức năng chính của hệ thống như đăng ký khóa học, làm bài tập, chức năng thanh toán, chức tạo khóa học, chức năng chat realtime.

**Đăng ký khóa học:** Cho phép học viên chọn và đăng ký các khóa học mà họ quan tâm. Giúp học viên dễ dàng tiếp cận và tham gia vào các khóa học, đồng thời giúp giảng viên quản lý số lượng học viên tham gia.

**Làm bài tập:** Cung cấp cho học viên các bài tập để kiểm tra và củng cố kiến thức đã học. Giúp học viên thực hành và áp dụng kiến thức, đồng thời giúp giảng viên đánh giá tiến độ và hiệu quả học tập của học viên.

**Chức năng thanh toán:** Cho phép học viên thanh toán học phí (nếu có) và các khoản phí liên quan trực tuyến. Tạo điều kiện thuận lợi cho học viên trong việc thanh toán, đồng thời giúp hệ thống quản lý tài chính và thu chi một cách hiệu quả.

**Chức năng quản lý khóa học:** Cho phép giảng viên tạo và quản lý các khóa học mới. Giúp giảng viên dễ dàng thiết lập nội dung khóa học, lịch trình và các tài liệu học tập, đồng thời giúp hệ thống duy trì và cập nhật các khóa học mới.

**Chức năng chat realtime:** Cung cấp kênh giao tiếp thời gian thực giữa học viên và giảng viên. Tăng cường tương tác và hỗ trợ học viên kịp thời, giúp giải đáp thắc mắc và tạo môi trường học tập năng động và hiệu quả.

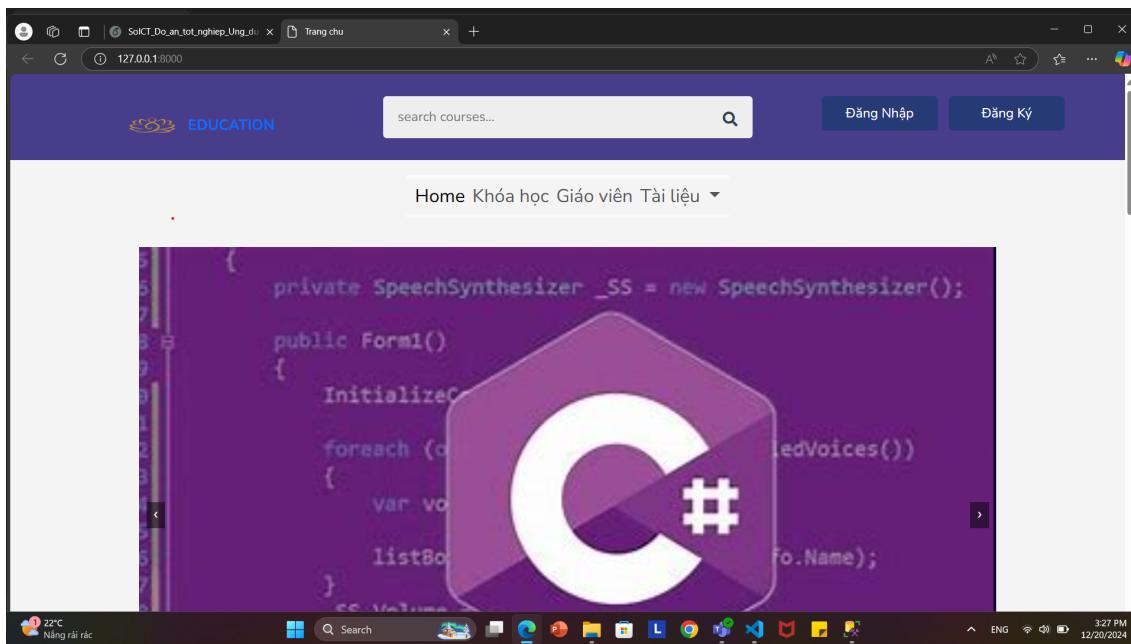
### 4.3.3 Minh họa các chức năng chính

Trong hệ thống, các màn hình đã thiết kế dựa trên các phân tích để đảm bảo người dùng dễ sử dụng, thao tác và vẫn đáp ứng đầy đủ các yêu cầu nghiệp vụ. Dưới đây là một số màn hình mô tả một số chức năng của hệ thống:

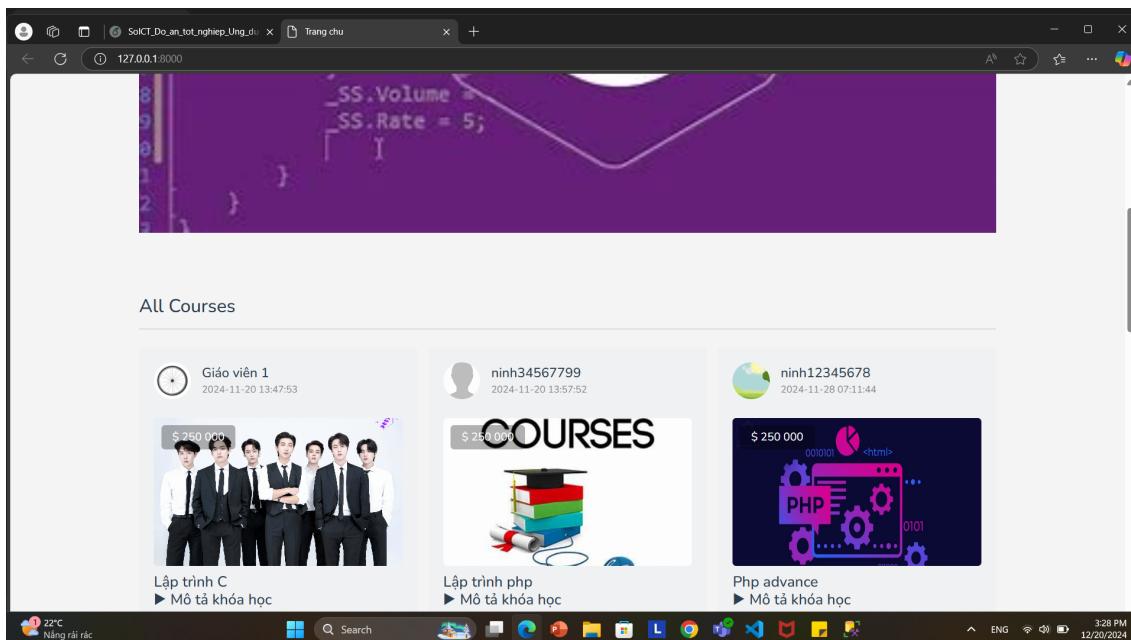
\* Màn hình trang chủ

Cung cấp các thông tin cơ bản của một hệ thống học tập như hiển thị các khóa học, phần đăng nhập cho người dùng và người dùng có thể tìm kiếm các khóa học mong muốn.

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



Hình 4.12: Trang chủ

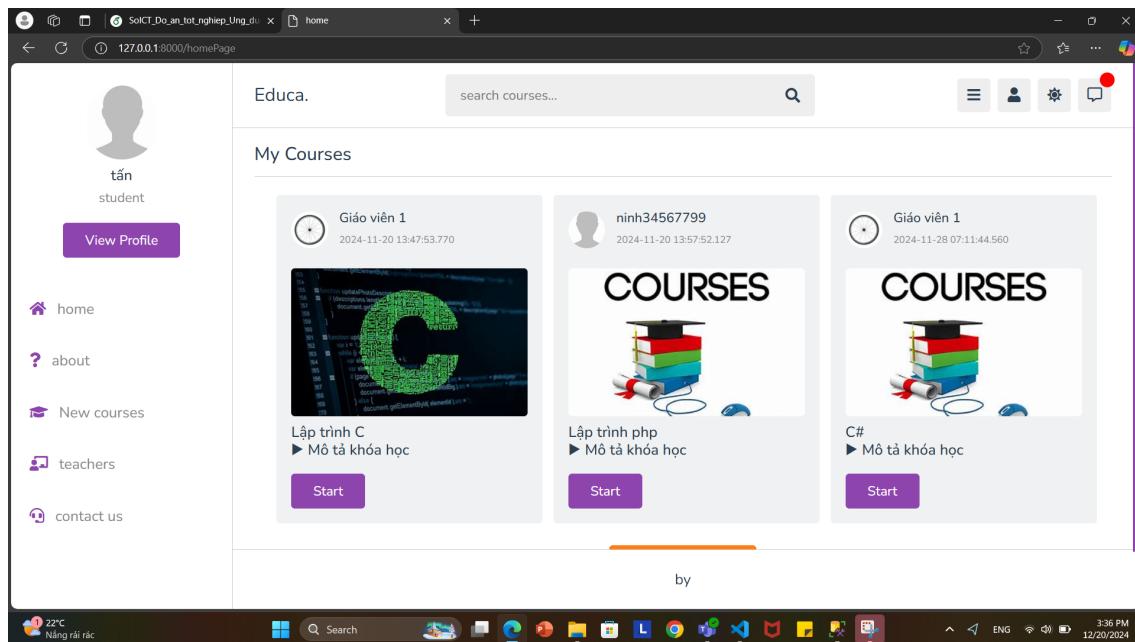


Hình 4.13: Trang chủ

\* Màn hình khi người dùng đã vào hệ thống

Người học sau khi đã đăng nhập vào hệ thống thì đưa đến trang chủ cá nhân. Các chức năng bao gồm như hiển thị các khóa học đang tham gia, có thể tìm kiếm các khóa học, nhận thông báo (nếu có), tìm kiếm giáo viên và chức năng cập nhật thông tin cá nhân.

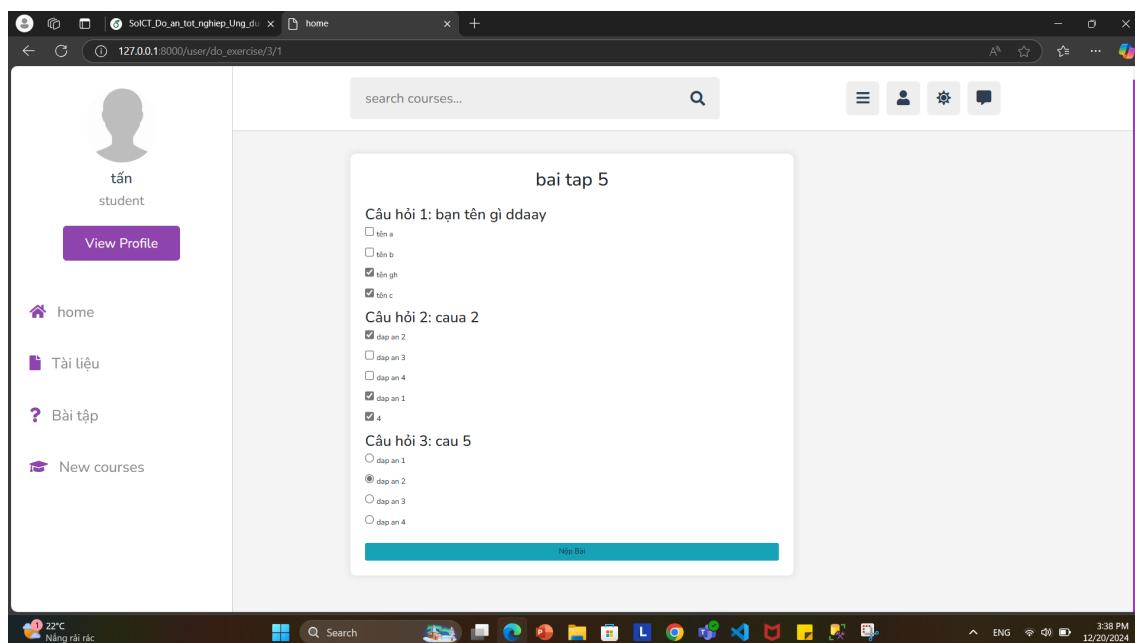
## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



**Hình 4.14:** Màn hình sau khi đăng nhập

\* Màn hình làm bài tập

Thành phần hiển thị gồm các câu hỏi và đáp án tương ứng với bài tập, các ô chọn đáp án sẽ tương ứng với loại câu hỏi( 1 hay nhiều đáp án ) giúp người học dễ dàng hơn trong việc học tập và củng cố và cuối cùng là nút gửi bài tập khi gửi hệ thống sẽ kiểm tra các câu hỏi đã được hoàn thành hết chưa và xác nhận lại yêu cầu nộp bài.



**Hình 4.15:** Trang làm bài tập

\* Màn hình quản lí khóa học

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Hiển thị đầy đủ thông tin về các khóa học hiện có. Giúp người quản lý hoặc giáo viên dễ dàng quản lý hoặc chỉnh sửa nếu có nhầm lẫn. Các chức năng có bản như chỉnh sửa, xóa đi kèm với từng khóa học giúp quản lý dễ dàng chỉnh sửa hoặc xóa bỏ. Bên trái trang là các liên kết đến các chức năng mà người quản lý có thể theo dõi dễ dàng.

ID	Tên khóa học	Giáo viên quản lý	Số học viên	Chi phí khóa học	Danh Mục	Action
1	Lập trình C	ninh12345678	3	0	Công nghệ thông tin	<button>Edit</button> <button>Delete</button> <button>Detail</button>
5	Lập trình php	ninh12345678	1	0	Công nghệ thông tin	<button>Edit</button> <button>Delete</button> <button>Detail</button>
6	PHP advance	Giáo viên 1	1	0	Công nghệ thông tin	<button>Edit</button> <button>Delete</button> <button>Detail</button>
7	C#	lê nguyên	1	0	Công nghệ thông tin	<button>Edit</button> <button>Delete</button> <button>Detail</button>
12	test teacher677	ninh12345678	0	0	Công nghệ thông tin	<button>Edit</button> <button>Delete</button> <button>Detail</button>

Hình 4.16: Trang quản lí khóa học

### \* Màn hình Thêm khóa học

Hiển thị các trường cần thiết khi tạo một khóa học mới bao gồm tên, học phí, ảnh đại diện(nếu có). Trong ô nhập liệu giáo viên cùng quản lý, đưa ra danh sách các giáo viên để người tạo lựa chọn và nút để xác nhận gửi yêu cầu tạo khóa học với các nội dung đã khai báo.

The screenshot shows a user profile on the left with a placeholder image and the name 'tấn student'. Below it are navigation links: home, Tài liệu, Bài tập, and New courses. The main area displays a form titled 'bai tap 5' containing three questions:

- Câu hỏi 1: bạn tên gì ddaay  
 tên a  
 tên b  
 tên gh  
 tên c
- Câu hỏi 2: caua 2  
 đáp án 2  
 đáp án 3  
 đáp án 4  
 đáp án 1  
 4
- Câu hỏi 3: cau 5  
 đáp án 1  
 đáp án 2  
 đáp án 3  
 đáp án 4

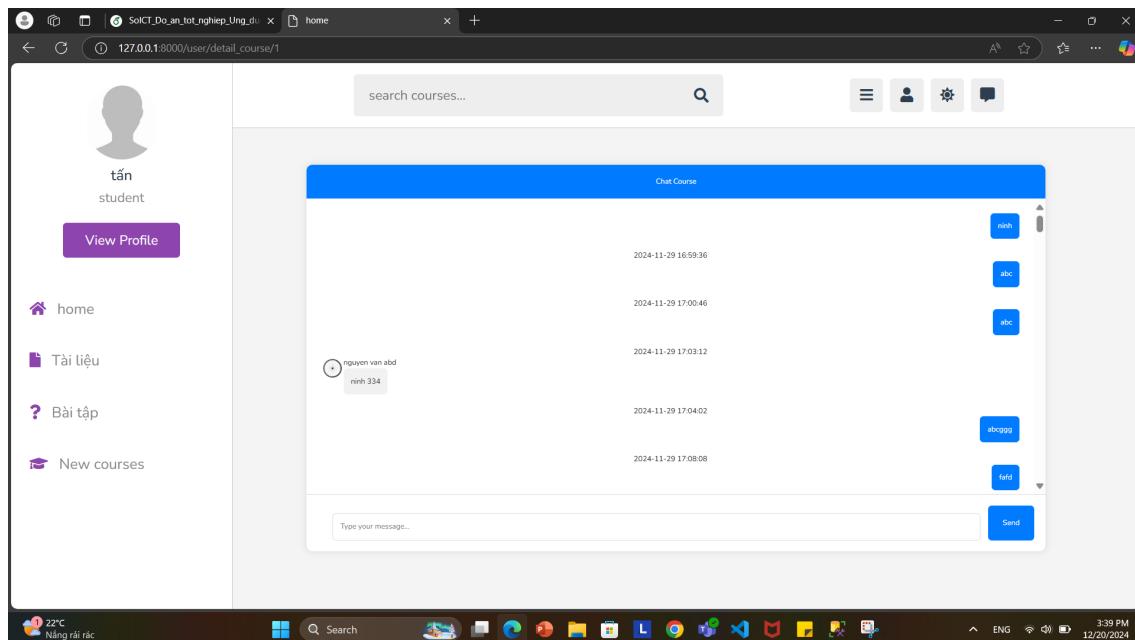
A large blue 'Nộp Bài' button is at the bottom of the form.

Hình 4.17: Trang tạo khóa học

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

### \* Màn hình chat của khóa học

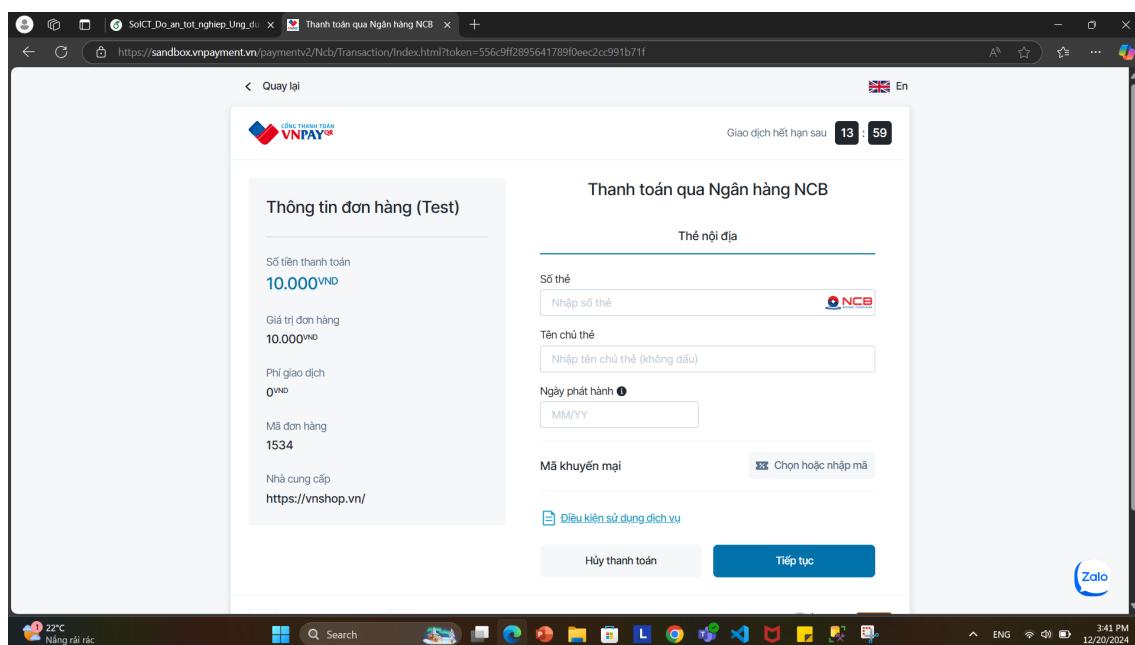
Giúp người học và giáo viên trao đổi các vấn đề thắc mắc liên quan trong khóa học tương ứng. Bao gồm các nội dung tên, nội dung người gửi và người nhận cùng thời gian gửi, nhận.



Hình 4.18: Trang chat khóa học

### \* Màn hình thanh toán

Trang thanh toán học phí của khóa học (nếu có). Hiển thị các nội dung cơ bản về thông tin thanh toán và nội dung thanh toán.



Hình 4.19: Trang Thanh toán

#### 4.4 Kiểm thử

Phần này có độ dài từ hai đến ba trang. Sinh viên thiết kế các trường hợp kiểm thử cho hai đến ba chức năng quan trọng nhất. Sinh viên cần chỉ rõ các kỹ thuật kiểm thử đã sử dụng. Chi tiết các trường hợp kiểm thử khác, nếu muốn trình bày, sinh viên đưa vào phần phụ lục. Sinh viên sau cùng tổng kết về số lượng các trường hợp kiểm thử và kết quả kiểm thử. Sinh viên cần phân tích lý do nếu kết quả kiểm thử không đạt

\* Kiểm thử chức năng Đăng nhập và đăng kí

STT	Mô tả	Dữ liệu đầu vào	Kết quả mong đợi	Kết quả thực tế
1	Đăng ký với email hợp lệ	Email: test@gmail.com Mật khẩu: 123456	Đăng ký thành công và chuyển đến trang đăng nhập	Đăng ký thành công
2	Đăng ký với email đã tồn tại	Email: test2@gmail.com Mật khẩu: 123456	Hiển thị thông báo lỗi "email đã tồn tại"	Hiển thị thông báo lỗi
3	Đăng nhập với thông tin đúng	Email: test@gmail.com Mật khẩu: 123456	Đăng nhập thành công và chuyển đến trang chủ	Đăng nhập thành công
4	Đăng nhập với tài khoản hoặc mật khẩu sai	Email: error@gmail.com Mật khẩu: 123	Thông báo tài khoản hoặc mật khẩu bị sai	Hiển thị lỗi

Bảng 4.11: Kiểm thử chức năng đăng nhập, đăng kí

STT	Mô tả	Dữ liệu đầu vào	Kết quả mong đợi	Kết quả thực tế
1	Người học chưa hoàn thành đủ các câu hỏi	Chọn thiếu câu trả lời	Thông báo người học chưa hoàn thành hết các câu hỏi	Thông báo chưa hoàn thành hết các câu hỏi
2	Hoàn thành tất cả các câu hỏi	Chọn hết các câu hỏi trong bài tập	Hệ thống hiển thị thông báo xác nhận nộp bài	Hệ thống hiển thị thông báo xác nhận nộp bài

Bảng 4.12: Kiểm thử chức năng nộp bài tập

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

STT	Mô tả	Dữ liệu đầu vào	Kết quả mong đợi	Kết quả thực tế
1	Nhập thông tin thanh toán sai	STK: 085430354, Ngày: 4/3, Tên TK: NGUYEN VAN A	Thông báo tài khoản không hợp lệ, yêu cầu nhập lại	Thông báo tài khoản không hợp lệ, yêu cầu nhập lại
2	Nhập đúng thông tin thanh toán	STK: 0854367887, Ngày: 07/15, Tên TK: NGUYEN VAN C	Hệ thống xác nhận hợp lệ và chuyển sang trang xác nhận	Hệ thống xác nhận hợp lệ và chuyển sang trang xác nhận

**Bảng 4.13:** Kiểm thử chức năng thanh toán

STT	Mô tả	Dữ liệu đầu vào	Kết quả mong đợi	Kết quả thực tế
1	Nhập thiếu trường bắt buộc	Nhập thiếu trường tên khóa học	Thiếu trường tên khóa học, yêu cầu nhập thêm	Thông báo thiếu trường tên khóa học
2	Nhập đủ trường theo yêu cầu	Nhập đủ các thông tin vào form	Hệ thống xác nhận thành công và quay lại trang danh sách khóa học	Hệ thống thêm thành công và quay lại trang danh sách khóa học

**Bảng 4.14:** Kiểm thử chức năng tạo khóa học mới

\*Phân tích kết quả kiểm thử Trong quá trình kiểm thử thì tất cả các trường hợp kiểm thử đều đạt kết quả mong đợi, không có lỗi xảy ra. Điều này chứng tỏ rằng hệ thống đã hoạt động ổn định và đúng như yêu cầu.

### 4.5 Triển khai

Hệ thống được triển khai và thử nghiệm thực tế trên máy tính cá nhân có những cấu hình và thông số như sau:

## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

---

Thông số	Cấu hình
Hệ điều hành	Win 11 64 bit
RAM	28 GB
CPU	12th Gen Intel(R) Core(TM) i5-12450HX 2.40 GHz
Cơ sở dữ liệu	SQL Server
Phần mềm khác	SQL Server , VS code

**Bảng 4.15:** Cấu hình máy tính triển khai hệ thống

## CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1 Một số vấn đề và giải pháp

Trong quá trình làm đồ án, em có gặp một vài vấn đề khó khăn như ở chức năng chat giữa người học và giáo viên làm cách nào để giáo viên và người học có thể trao đổi qua lại mà không phải tải lại trang website khi có một tin nhắn mới được gửi. Sau khi tìm hiểu em đã tìm ra giải pháp giúp giải quyết vấn đề trên. Em đã sử dụng pusher để xử lý các tác vụ thời gian thực. Cách hoạt động của pusher là khi có 1 tin nhắn được gửi hệ thống sẽ gửi dữ liệu đến pusher thông qua api của pusher. Sau đó máy chủ pusher nhận dữ liệu và xác định các kênh nhận thông báo. Máy chủ pusher phát sóng dữ liệu đến tất cả các client đã đăng ký. Các client nhận dữ liệu và cập nhập giao diện người dùng theo thời gian thực. Một vấn đề nữa có thể kể đến là việc khi thực hiện chức năng xóa 1 hàng dữ liệu thì việc phải tải lại cả trang gây ảnh hưởng đến trải nghiệm. Từ kinh nghiệm xây dựng hệ thống đã có thì em đã sử dụng ajax - một kỹ thuật phát triển web cho phép các trang web cập nhập nội dung một cách động mà không cần phải tải lại toàn bộ trang.

### 5.2 Kết luận

Trong quá trình thực hiện đồ án, em học thêm và tích lũy thêm nhiều kinh nghiệm về nghiên cứu, phân tích, thiết kế và triển khai hệ thống trên nền tảng website. Hệ thống cơ bản đã hoàn thành các chức năng cần thiết so với các hệ thống đã có. Hệ thống học tập trực tuyến đã chứng minh được tính hiệu quả và tiện lợi trong việc cung cấp kiến thức và đào tạo. Với khả năng tiếp cận rộng rãi, tính tương tác cao và khả năng tùy chỉnh nội dung học tập, hệ thống này đã mang lại nhiều lợi ích cho cả học viên và giảng viên. Chất lượng nội dung của các khóa học, tài liệu học tập được cung cấp đầy đủ từ các giáo viên. Ngoài ra, với các chức năng cần có của hệ thống học trực tuyến như đăng ký khóa học, làm bài tập, chức năng thanh toán, tạo khóa học và chức năng chat realtime.

Bên cạnh đó, trong quá trình phát triển và hoàn thiện hệ thống, em nhận thấy vẫn còn những mảng chưa thực sự hoặc cần hoàn thiện thêm. Một trong số những chức năng như cấp chứng chỉ cho người học khi hoàn thành khóa học. Vấn đề cần kết hợp với các giáo viên liên quan đến khóa học để thực hiện. Một chức năng nữa phải nói đến như phân tích dữ liệu người học để có thể phát triển các khóa học theo xu hướng các lĩnh vực được quan tâm nhiều tại từng thời điểm. Giúp người kết nối kịp thời giữa người chia sẻ kiến thức và người có nhu cầu học tập. Ngoài ra, em cũng nhận thấy cần thực hiện kiểm thử và đánh giá một cách tổng quan hơn hệ thống khi đưa vào bộ kiểm thử với số lượng người dùng lớn để kịp thời phát hiện những vấn

dề tiềm ẩn cũng như khả năng chịu tải của hệ thống. Mặc dù đã thực hiện các kiểm thử cơ bản và đánh giá hiệu suất nhưng chưa đủ chi tiết và bao quát để có thể đảm bảo được hệ thống có thể hoạt động ổn định và hiệu quả với tất cả các tình huống có thể xảy ra trong môi trường thực tế. Việc thực hiện kiểm thử chi tiết và đánh giá với số lượng người dùng lớn là cần thiết.

Trong suốt quá trình thực hiện đồ án, em đã học hỏi và tích lũy được thêm các kinh nghiệm về cả lĩnh vực chuyên môn cũng như lĩnh vực nghiệp vụ liên quan của đề tài. Đối với phần chuyên môn, em đã dày dặn hơn trong phần tạo và tổ chức các đề mục công việc liên quan một cách hiệu quả, phân bổ được thời gian một cách hiệu quả và hợp lý, rèn luyện được khả năng tự lãnh đạo và làm việc chủ động. Bên cạnh đó, em đã gặp phải nhiều vấn đề khó khăn trong quá trình triển khai và phát triển. Sau khi giải quyết các vấn đề, lỗi phát sinh em nhận thấy rằng việc phát triển kỹ năng sẽ song song với quá trình giải quyết các vấn đề từ cơ bản đến phức tạp. Từ việc giải quyết, xử lý vấn đề tìm được điểm thắt có thể giúp tối ưu và hoàn thiện hơn về một chức năng nói riêng cũng như một vấn đề nói chung.

### 5.3 Hướng phát triển

Dựa trên kết quả của quá trình phân tích và thiết kế hệ thống. Các chức năng đã được đưa ra để xây dựng một hệ thống hoàn chỉnh. Một số chức năng như: chức năng đăng ký khóa học dành cho người học, các công việc thiết kế giao diện gồm các danh sách khóa học và các nút chức năng, tạo các phương thức để lưu trữ các thông tin đăng ký vào cơ sở dữ liệu, tích hợp thanh toán nếu khóa học có phí, kiểm thử các phương thức liên quan để đảm bảo hệ thống hoạt động ổn định. Với chức năng thanh toán khóa học các công việc cần thiết như tích hợp các hình thức thanh toán, tạo phương thức để lưu các nội dung thanh toán vào cơ sở dữ liệu. Với tính năng chat thì các công việc được tạo như tạo giao diện hộp thoại chat, tạo phương thức lưu nội dung chat, tạo phương thức để gửi realtime nội dung tới người nhận. Các chức năng ở trên có thể được cải thiện hơn như tích hợp thêm các phương thức thanh toán để người dùng có thể dễ dàng lựa chọn. Với tính năng chat có thể đưa thêm các tùy chọn ảnh file có thể gửi giúp người học trao đổi được nhiều cách hơn.

Một hệ thống ổn định và hiệu quả là một hệ thống luôn được cập nhập và phát triển. Không ngoại lệ, đối với hệ thống được triển khai trong đồ án sẽ có các hướng phát triển sao cho hệ thống ngày càng được hoàn thiện trong tương lai. Hệ thống có thể mở rộng như phát triển thêm nhiều ngôn ngữ để hỗ trợ các học viên từ nhiều quốc gia khác nhau. Tích hợp thêm các công nghệ trí tuệ nhân tạo hỗ trợ học tập giúp người học có thể chủ động hơn trong quá trình trao đổi kiến thức. Phát triển thêm các loại bài tập với các dạng đặc thù hơn chẳng hạn như lập trình.

## **TÀI LIỆU THAM KHẢO**

- [1] SQL, <https://www.w3schools.com/sql/>
- [2] LaravelFramework, <https://laravel.com/>
- [3] Coursera, <https://www.coursera.org/>