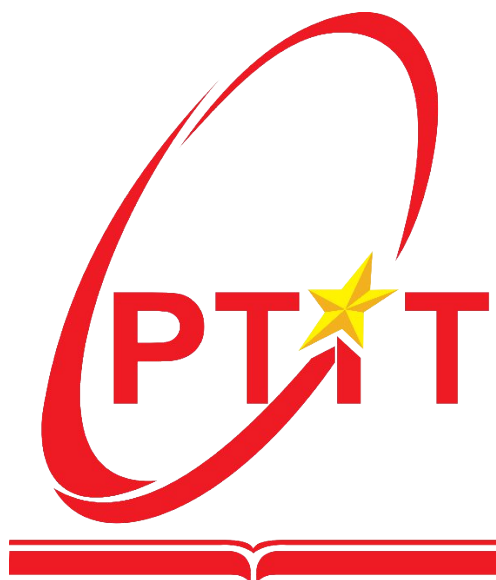


**BỘ KHOA HỌC VÀ CÔNG NGHỆ**  
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----o0o-----



**BÁO CÁO KẾT THÚC MÔN HỌC**  
**THỰC TẬP CƠ SỞ**

**Đề tài: Xây dựng website dạy học**  
**trực tuyến Techedu**

**Nhóm 1**

**Sinh viên: Trần Văn Hoàng**

**Giảng viên hướng dẫn: Kim Ngọc Bách**

**Mã sinh viên: B22DCAT129**

**Lớp: E22CQCN05-B**

**Ngày sinh: 11/11/2004**

**Số điện thoại: 0983235326**

**Hà Nội, 2025**

## Mục lục

LỜI CẢM ƠN.....	5
LỜI NÓI ĐẦU.....	6
CHƯƠNG I: GIỚI THIỆU TỔNG QUAN VỀ DỰ ÁN.....	7
1. Giới thiệu dự án.....	7
1.1. Tên dự án.....	7
1.2. Mục tiêu dự án.....	7
1.3. Nhiệm vụ nghiên cứu.....	7
1.4. Đối tượng nghiên cứu.....	7
2. Công nghệ sử dụng.....	7
CHƯƠNG II: CƠ SỞ LÝ THUYẾT.....	9
1. Tổng quan về website.....	9
1.1. Khái niệm Website.....	9
1.2. Các thành phần cơ bản của một Website.....	9
1.3. Phân loại Website.....	10
1.4. Lợi ích của Website.....	10
1.5. Quy trình phát triển Website.....	10
2. Tổng quan về công nghệ sử dụng.....	11
2.1. Công nghệ phía Client (Frontend).....	11
2.1.1. HTML (HyperText Markup Language).....	11
2.1.2. CSS (Cascading Style Sheets).....	12
2.1.3. JavaScript (JS).....	14
2.2. Công nghệ phía Server (Backend).....	15
2.2.1. Ngôn ngữ lập trình Python.....	15
2.2.2. Framework Django.....	15
3. Tổng quan về cơ sở dữ liệu SQLite.....	17
3.1. Khái niệm Cơ sở dữ liệu (CSDL) và Hệ quản trị CSDL (HQTCSDL).....	17
3.2. Giới thiệu về SQLite.....	17
3.3. Kiến trúc của SQLite.....	18
3.4. Các kiểu dữ liệu trong SQLite (Storage Classes).....	18
3.5. Ưu điểm của SQLite.....	19
3.6. Nhược điểm và Trường hợp không nên dùng SQLite.....	19
3.7. SQLite trong phát triển ứng dụng Django.....	20
CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	21
1. Glossary list.....	21

2. Mô tả hệ thống .....	23
2.1. Phạm vi phần mềm .....	23
2.2. Người dùng và chức năng của người dùng.....	23
2.3. Thông tin các đối tượng cần xử lý.....	24
2.4. Quan hệ giữa các đối tượng cần xử lý.....	24
2.5. Sơ đồ quan hệ thực thể:.....	25
Bảng mô tả cơ sở dữ liệu.....	25
<b>CHƯƠNG IV: MÔ TẢ NGHIỆP VỤ CHI TIẾT CỦA CÁC MODULE.....</b>	<b>30</b>
1. Mô tả các module .....	30
1.1. Module Quản lý người dùng .....	30
1.2. Module Quản lý khóa học .....	30
1.3. Module Quản lý blog.....	31
2. Các yêu cầu phi chức năng.....	31
3. Xây dựng các chức năng hệ thống.....	31
3.1. Biểu đồ Usecase tổng quan.....	32
3.2. Thiết kế các module .....	32
3.2.1. Module Quản lý người dùng .....	32
3.2.1.1 Usecase đăng ký tài khoản.....	33
3.2.1.2. Usecase đăng nhập tài khoản.....	37
3.2.1.5. Usecase đăng xuất.....	40
3.2.1.4. Usecase Quản lý hồ sơ cá nhân(bao gồm cả mật khẩu).....	42
3.2.2. Module Quản lý khóa học .....	46
3.2.2.1. Usecase Tạo khóa học .....	47
3.2.2.2. Usecase Chỉnh sửa khóa học .....	49
3.2.2.3. Usecase Xóa khóa học.....	52
3.2.2.4. Usecase Xem danh sách khóa học.....	55
3.2.2.5. Usecase Xem chi tiết khóa học và đăng ký khóa học.....	57
3.2.2.5. Usecase Học bài giảng/quiz.....	60
3.2.3. Module Quản lý blog.....	62
3.2.3.1. Usecase xem blog.....	63
3.2.3.2. Usecase Đăng bài blog.....	65
3.2.3.3. Bình luận blog.....	67
3.2.3.4.     Tìm kiếm blog.....	69
3.2.3.5.     Chỉnh sửa blog.....	72
3.2.3.6.     Xóa blog.....	75
<b>CHƯƠNG V: CÀI ĐẶT VÀ CHẠY CHƯƠNG TRÌNH.....</b>	<b>78</b>

1. Cài đặt chương trình.....	78
2. Cài đặt các thư viện cần thiết.....	78
3. Chạy chương trình.....	78

## LỜI CẢM ƠN

Em xin chân thành cảm ơn Khoa Công nghệ Thông tin 1, Học viện Công nghệ Bưu chính Viễn thông đã tạo điều kiện tốt cho em thực hiện đề tài này.

Em xin chân thành cảm ơn thầy Kim Ngọc Bách, là giảng viên hướng dẫn em, giúp em giải quyết các vấn đề, chỉ bảo em trong suốt thời gian thực hiện đề tài.

Em cũng xin gửi lời cảm ơn sâu sắc đến quý Thầy Cô trong Khoa Công nghệ Thông tin Chất lượng cao đã tận tình giảng dạy, trang bị cho em những kiến thức nền tảng quan trọng trong những năm học vừa qua. Mặc dù đã cố gắng trong phạm vi và khả năng cho phép, nhưng chắc chắn sẽ không tránh khỏi những thiếu sót, kính mong sự cảm thông và tận tình chỉ bảo của quý Thầy Cô!

Hà Nội, ngày 28 tháng 05 năm 2025

## LỜI NÓI ĐẦU

Ngày nay, công nghệ thông tin đã và đang được áp dụng rộng rãi vào nhiều lĩnh vực, bao gồm cả ngành giáo dục. Đặc biệt, ngành giáo dục trực tuyến đang chứng kiến sự bùng nổ mạnh mẽ với sự ra đời của nhiều nền tảng cung cấp các khóa học trực tiếp. Những nền tảng này không chỉ đáp ứng nhu cầu tiếp thu kiến thức của người dùng mà còn mở ra những cơ hội mới cho việc phân phối nội dung giáo dục.

Trong bối cảnh đó, dự án xây dựng website dạy và học Techedu ra đời nhằm giúp người dùng có thể tiếp cận kiến thức dễ dàng hơn

Dự án gồm:

### **Chương I: Giới thiệu dự án.**

Giới thiệu sơ lược về website dạy học trực tuyến Techedu.

### **Chương II: Cơ sở lý thuyết**

Tìm hiểu về framework lập trình web python django, html, css và javascript

### **Chương III: Xây dựng website học tập Techedu**

Xây dựng website học tập và giảng dạy giúp người học tiếp cận với kiến thức một cách dễ dàng và hiệu quả.

### **Chương IV. Kết luận**

Đưa ra các kết quả đã đạt được của dự án, những điểm còn tồn tại và hướng phát triển của ứng dụng trong tương lai.

# CHƯƠNG I: GIỚI THIỆU TỔNG QUAN VỀ DỰ ÁN

## 1. Giới thiệu dự án

### 1.1. Tên dự án

Xây dựng Website dạy học trực tuyến Techedu.

### 1.2. Mục tiêu dự án

Xây dựng website hỗ trợ quản lý, tổ chức khóa học hiệu quả. Giúp cắt giảm thời gian di chuyển, linh động trong việc học tập cho học sinh, sinh viên. Giúp giáo viên và học sinh chủ động hơn trong việc dạy và học. Giúp tiếp cận kiến thức một cách miễn phí và chủ động.

Tính ứng dụng cao: Website dạy học trực tuyến không những mang tính giáo dục và tiện lợi mà còn hỗ trợ những nhà phân phối khóa học có thể quảng bá hình ảnh, phân phối khóa học và giúp open source công nghệ và kỹ thuật.

### 1.3. Nhiệm vụ nghiên cứu

Phát triển trang website thuận tiện cho người dùng: Website dạy học trực tuyến Techedu được tạo ra với sự thuận tiện tối đa cho người dùng, giúp người học dễ dàng tìm kiếm, lựa chọn và học các khóa học phù hợp với nhu cầu của bản thân.

Tăng cường trải nghiệm người dùng: Nghiên cứu và phát triển các tính năng nâng cao trải nghiệm người dùng bao gồm gợi ý dựa trên những khóa học nổi tiếng, có khả năng tương tác với người dùng khác trong cộng đồng sử dụng website.

### 1.4. Đối tượng nghiên cứu

Người học có khả năng sử dụng thiết bị điện tử và mong muốn tiếp cận những khóa học chất lượng miễn phí.

## 2. Công nghệ sử dụng

Ngôn ngữ lập trình:

- HTML(HyperText Markup Language - Ngôn ngữ Đánh dấu Siêu văn bản)
  - Vai trò chính: Xây dựng cấu trúc và nội dung của một trang web.
  - Miêu tả: HTML không phải là ngôn ngữ lập trình mà là ngôn ngữ đánh dấu. Nó sử dụng các "thẻ" (tags) để định nghĩa các thành phần khác nhau trên trang, ví dụ: tiêu đề (<h1>), đoạn văn (<p>), hình ảnh (<img>), liên kết (<a>), danh sách (<ul>, <ol>), bảng (<table>), biểu mẫu (<form>), v.v.
  - Tưởng tượng: HTML giống như khung xương của một ngôi nhà, xác định các phòng, cửa sổ, cửa ra vào ở đâu.
- CSS (Cascading Style Sheets - Bảng kiểu theo tầng)
  - Vai trò chính: Định dạng giao diện và trình bày cho nội dung HTML.
  - Miêu tả: CSS cho phép bạn kiểm soát màu sắc, phông chữ, kích thước, khoảng cách, vị trí và cách bố trí của các phần tử HTML. Nó

giúp trang web trở nên trực quan hấp dẫn và dễ sử dụng hơn. CSS có thể được viết trực tiếp trong file HTML, trong thẻ <style>, hoặc phổ biến nhất là trong các file .css riêng biệt và liên kết vào HTML.

- Tưởng tượng: CSS giống như nội thất, sơn tường, trang trí cho ngôi nhà (được xây dựng bằng HTML), làm cho nó đẹp hơn.
- Javascript:
  - Vai trò chính: Tạo ra tính tương tác và hành vi động cho trang web phía trình duyệt (client-side).
  - Miêu tả: JavaScript là một ngôn ngữ lập trình kịch bản. Nó cho phép bạn:
    - Thay đổi nội dung HTML và thuộc tính CSS một cách linh động.
    - Phản hồi lại các hành động của người dùng (click chuột, nhập liệu, di chuyển chuột).
    - Xác thực dữ liệu người dùng nhập vào biểu mẫu.
    - Thực hiện các yêu cầu dữ liệu đến máy chủ mà không cần tải lại toàn bộ trang (sử dụng AJAX/Fetch API).
    - Tạo hiệu ứng động, slideshow, menu thả xuống, v.v.
  - Tưởng tượng: JavaScript giống như hệ thống điện, nước, các thiết bị thông minh trong ngôi nhà, cho phép cửa tự động mở, đèn bật tắt, TV hoạt động.
- Python (framework django):
  - Python: Là ngôn ngữ lập trình đa năng, dễ đọc, được dùng để viết logic cho ứng dụng.
  - Django: Là một framework (bộ khung) web viết bằng Python, giúp xây dựng phần backend (máy chủ) của ứng dụng web một cách có tổ chức và hiệu quả. Nó xử lý các yêu cầu từ người dùng, tương tác với cơ sở dữ liệu, thực hiện logic nghiệp vụ, và thường tạo ra HTML động để gửi về trình duyệt.
  - Tưởng tượng: Bộ não, nhà bếp, hoặc phòng điều khiển trung tâm của ngôi nhà, xử lý mọi yêu cầu và chuẩn bị "thức ăn" (dữ liệu/HTML) để phục vụ.



## CHƯƠNG II: CƠ SỞ LÝ THUYẾT

Chương này trình bày các kiến thức lý thuyết nền tảng liên quan đến đề tài, bao gồm tổng quan về website, các công nghệ được sử dụng để phát triển ứng dụng web, và đặc biệt là hệ quản trị cơ sở dữ liệu SQLite. Việc nắm vững các khái niệm này là cơ sở quan trọng để hiểu rõ hơn về quy trình thiết kế, xây dựng và triển khai một ứng dụng web hoàn chỉnh.

### 1. Tổng quan về website

Trong kỷ nguyên số hiện nay, website đã trở thành một công cụ không thể thiếu cho cá nhân, tổ chức và doanh nghiệp trong việc chia sẻ thông tin, quảng bá thương hiệu, cung cấp dịch vụ và tương tác với người dùng trên toàn cầu.

#### 1.1. Khái niệm Website

Website (thường được gọi tắt là site) là một tập hợp các trang web (web pages) có liên quan, chứa nội dung văn bản, hình ảnh, video, âm thanh, và các tập tin đa phương tiện khác, được lưu trữ trên ít nhất một máy chủ web (web server) và có thể truy cập thông qua mạng Internet bằng một địa chỉ duy nhất được gọi là URL (Uniform Resource Locator). Người dùng có thể xem website thông qua một chương trình ứng dụng gọi là trình duyệt web (web browser) như Google Chrome, Mozilla Firefox, Safari, Microsoft Edge.

Mỗi trang web trong một website thường được viết bằng ngôn ngữ đánh dấu siêu văn bản HTML (HyperText Markup Language) hoặc các biến thể của nó như XHTML, HTML5. Các trang này được liên kết với nhau thông qua các siêu liên kết (hyperlinks), cho phép người dùng điều hướng dễ dàng giữa các nội dung khác nhau.

#### 1.2. Các thành phần cơ bản của một Website

Một website hoàn chỉnh thường bao gồm ba thành phần chính:

**Frontend (Giao diện người dùng - Client-side):** Là phần mà người dùng nhìn thấy và tương tác trực tiếp trên trình duyệt. Nó bao gồm tất cả các yếu tố giao diện như bố cục, màu sắc, font chữ, hình ảnh, nút bấm, biểu mẫu, hiệu ứng động. Công nghệ chủ yếu để xây dựng frontend là HTML (cấu trúc), CSS (trình bày) và JavaScript (tương tác). Mục tiêu của frontend là tạo ra trải nghiệm người dùng (User Experience - UX) tốt và giao diện người dùng (User Interface - UI) hấp dẫn, dễ sử dụng.

**Backend (Máy chủ ứng dụng - Server-side):** Là phần xử lý logic nghiệp vụ, quản lý dữ liệu và tương tác với cơ sở dữ liệu, hoạt động trên máy chủ. Người dùng không trực tiếp nhìn thấy hay tương tác với backend. Backend nhận yêu cầu từ frontend, xử lý chúng (ví dụ: xác thực người dùng, truy vấn dữ liệu, tính toán), và gửi phản hồi trở lại cho frontend để hiển thị cho người dùng. Các ngôn ngữ lập trình phổ biến cho backend bao gồm Python (với các framework như Django, Flask), Java (Spring), PHP (Laravel, Symfony), Ruby (Ruby on Rails), Node.js (Express.js).

**Cơ sở dữ liệu (Database):** Nơi lưu trữ và quản lý toàn bộ dữ liệu của website một cách có tổ chức, ví dụ như thông tin người dùng, sản phẩm, bài viết, đơn hàng. Backend sẽ tương tác với cơ sở dữ liệu để đọc, ghi, cập nhật hoặc xóa dữ liệu khi cần thiết. Có nhiều loại cơ sở dữ liệu khác nhau như cơ sở dữ liệu quan hệ (MySQL, PostgreSQL, SQL Server, SQLite) và NoSQL (MongoDB, Cassandra, Redis).

### 1.3. Phân loại Website

Website có thể được phân loại theo nhiều tiêu chí khác nhau chủ yếu dựa trên tính tương tác và nội dung:

- Website tĩnh (Static Website): Là website mà nội dung các trang được xây dựng sẵn và không thay đổi theo tương tác của người dùng. Mỗi khi người dùng yêu cầu một trang, máy chủ sẽ gửi trực tiếp file HTML đã được định sẵn đó cho trình duyệt. Website tĩnh thường đơn giản, dễ phát triển, tốc độ tải nhanh và chi phí thấp. Chúng phù hợp cho các trang giới thiệu công ty, portfolio cá nhân, hoặc các trang thông tin ít khi cập nhật.
- Website động (Dynamic Website): Là website có nội dung được tạo ra một cách linh động dựa trên yêu cầu của người dùng, tương tác với cơ sở dữ liệu và các yếu tố khác. Các trang web động được tạo ra "on-the-fly" bởi ứng dụng backend. Chúng cho phép người dùng tương tác mạnh mẽ hơn (ví dụ: đăng nhập, bình luận, tìm kiếm, mua hàng). Hầu hết các website hiện đại như trang thương mại điện tử, mạng xã hội, diễn đàn, blog có hệ thống quản trị nội dung (CMS) đều là website động. Website dạy và học Techedu là loại website này.

### 1.4. Lợi ích của Website

Việc sở hữu một website mang lại nhiều lợi ích quan trọng:

- Tăng cường sự hiện diện trực tuyến: Giúp cá nhân, doanh nghiệp tiếp cận được lượng lớn khách hàng tiềm năng trên Internet.
- Cung cấp thông tin 24/7: Người dùng có thể truy cập thông tin mọi lúc, mọi nơi.
- Công cụ marketing hiệu quả: Quảng bá thương hiệu, sản phẩm, dịch vụ với chi phí tối ưu hơn so với các phương tiện truyền thống.
- Nâng cao uy tín và tính chuyên nghiệp: Một website được thiết kế tốt sẽ tạo ấn tượng tích cực với khách hàng.
- Tương tác và thu thập phản hồi: Dễ dàng giao tiếp, nhận phản hồi từ người dùng để cải thiện sản phẩm, dịch vụ.
- Mở rộng thị trường: Không bị giới hạn bởi vị trí địa lý.
- Tự động hóa quy trình: Ví dụ như đặt hàng, thanh toán trực tuyến, hỗ trợ khách hàng qua chatbot.

### 1.5. Quy trình phát triển Website

Một quy trình phát triển website điển hình thường bao gồm các bước sau:

1. Thu thập yêu cầu và lập kế hoạch: Xác định mục tiêu, đối tượng người dùng, chức năng cần có, phạm vi dự án.
2. Thiết kế (Design):
  - Thiết kế kiến trúc thông tin (Information Architecture): Tổ chức cấu trúc nội dung, luồng người dùng.
  - Thiết kế Wireframe và Mockup: Tạo bản phác thảo bố cục và giao diện cơ bản.
  - Thiết kế giao diện người dùng (UI Design): Hoàn thiện giao diện trực quan, màu sắc, font chữ, hình ảnh.
  - Thiết kế trải nghiệm người dùng (UX Design): Đảm bảo website dễ sử dụng, thân thiện và mang lại trải nghiệm tốt.
3. Lập trình Frontend (Frontend Development): Chuyển đổi thiết kế UI thành các trang web tương tác bằng HTML, CSS, JavaScript.
4. Lập trình Backend (Backend Development): Xây dựng logic nghiệp vụ, API, quản lý cơ sở dữ liệu bằng các ngôn ngữ và framework phía máy chủ.
5. Phát triển cơ sở dữ liệu (Database Development): Thiết kế, tạo và quản lý cơ sở dữ liệu.
6. Tích hợp và Kiểm thử (Integration & Testing): Kết hợp các thành phần frontend, backend, cơ sở dữ liệu và kiểm tra lỗi, đảm bảo các chức năng hoạt động đúng như yêu cầu.
7. Triển khai (Deployment): Đưa website lên máy chủ web để người dùng có thể truy cập trực tuyến.
8. Bảo trì và Nâng cấp (Maintenance & Updates): Theo dõi hoạt động, sửa lỗi, cập nhật nội dung, nâng cấp tính năng và bảo mật theo thời gian.

## **2. Tổng quan về công nghệ sử dụng**

Để xây dựng một website động hoàn chỉnh, cần sự kết hợp của nhiều công nghệ khác nhau, bao gồm các công nghệ phía client (trình duyệt) và phía server (máy chủ).

### **2.1. Công nghệ phía Client (Frontend)**

Đây là những công nghệ chạy trực tiếp trên trình duyệt của người dùng, chịu trách nhiệm hiển thị giao diện và xử lý các tương tác cơ bản.

#### **2.1.1. HTML (HyperText Markup Language)**

Khái niệm và Vai trò: HTML là ngôn ngữ đánh dấu tiêu chuẩn được sử dụng để tạo và cấu trúc nội dung của các trang web. Nó không phải là một ngôn ngữ lập trình mà là một ngôn ngữ mô tả, sử dụng các "thẻ" (tags) để bao bọc và định danh các phần tử khác nhau của trang như tiêu đề, đoạn văn, hình ảnh, liên kết, bảng biểu, biểu mẫu, v.v. Trình duyệt web đọc các file HTML và diễn giải các thẻ này để hiển thị nội dung lên màn hình.

Đặc điểm chính:

Cấu trúc dựa trên thẻ: Mỗi thẻ HTML có một ý nghĩa ngữ nghĩa riêng (ví dụ: `

` cho đoạn văn, `

# ` cho tiêu đề cấp 1, `` cho hình ảnh).

Phiên bản phổ biến HTML5: Mang đến nhiều thẻ ngữ nghĩa mới (như ``<article>``, ``<section>``, ``<nav>``, ``<header>``, ``<footer>``), hỗ trợ tốt hơn cho đa phương tiện (thẻ ``<audio>``, ``<video>``), đồ họa (thẻ ``<canvas>``, ``<svg>``), và các API cho ứng dụng web offline, lưu trữ cục bộ.

Nền tảng của web: Mọi trang web đều bắt đầu với HTML.

Cấu trúc cơ bản của một tài liệu HTML:

```
``html
<!DOCTYPE html> <!-- Khai báo loại tài liệu là HTML5 -->
<html lang="vi"> <!-- Thẻ gốc, khai báo ngôn ngữ là tiếng Việt -->
<head>
  <meta charset="UTF-8"> <!-- Khai báo bộ ký tự UTF-8 -->
  <meta name="viewport" content="width=device-width, initial-
scale=1.0"> <!-- Thiết lập cho responsive design -->
  <title>Tiêu đề trang web</title> <!-- Tiêu đề hiển thị trên tab trình duyệt
-->
  <!-- Liên kết tới file CSS, các thẻ meta khác -->
</head>
<body>
  <!-- Nội dung chính của trang web sẽ hiển thị ở đây -->
  <header>
    <h1>Tiêu đề chính của trang</h1>
  </header>
  <nav>
    <ul>
      <li><a href="#">Trang chủ</a></li>
      <li><a href="#">Giới thiệu</a></li>
    </ul>
  </nav>
  <main>
    <article>
      <p>Đây là một đoạn văn bản.</p>
    </article>
  </main>
  <footer>
    <p>© 2025 Bản quyền thuộc về...</p>
  </footer>
  <!-- Liên kết tới file JavaScript -->
</body>
</html>
``
```

### 2.1.2. CSS (Cascading Style Sheets)

Khái niệm và Vai trò: CSS là một ngôn ngữ bảng kiểu được sử dụng để mô tả cách trình bày (style) của các tài liệu HTML (hoặc XML). Nó cho phép tách biệt phần nội dung (HTML) và phần trình bày (CSS), giúp mã nguồn dễ quản lý, bảo

trì và tái sử dụng hơn. CSS kiểm soát màu sắc, font chữ, kích thước, lề, khoảng cách, vị trí, nền, và nhiều thuộc tính trực quan khác của các phần tử HTML.

Đặc điểm chính:

Cascading (Theo tầng): Các quy tắc CSS được áp dụng theo một thứ tự ưu tiên nhất định. Các quy tắc từ nhiều nguồn (style của trình duyệt, style của người dùng, style của tác giả) sẽ được kết hợp lại. Nếu có xung đột, quy tắc có độ ưu tiên cao hơn hoặc được định nghĩa sau (trong một số trường hợp) sẽ ghi đè lên quy tắc trước đó.

Selectors (Bộ chọn): CSS sử dụng các bộ chọn để "chọn" các phần tử HTML cụ thể mà bạn muốn áp dụng style. Ví dụ: chọn theo tên thẻ (`p`), theo class (`.my-class`), theo ID (`#my-id`), theo thuộc tính (`[type="text"]`), hoặc các bộ chọn phức tạp hơn.

Box Model (Mô hình hộp): Mỗi phần tử HTML được coi như một hộp chữ nhật, bao gồm nội dung (content), đệm (padding), viền (border), và lề (margin). CSS cho phép kiểm soát kích thước và khoảng cách của các thành phần này.

Responsive Web Design: CSS3 giới thiệu các kỹ thuật quan trọng như Media Queries, Flexbox, Grid Layout giúp tạo ra các trang web có khả năng tự điều chỉnh bố cục để hiển thị tốt trên nhiều kích thước màn hình khác nhau (desktop, tablet, mobile).

Các cách nhúng CSS vào HTML:

1. Inline CSS: Viết trực tiếp trong thuộc tính `style` của thẻ HTML. Ít được khuyến khích vì khó quản lý.

```
<p style="color: blue; font-size: 16px;">Đoạn văn màu xanh.</p>
```

2. Internal CSS (Embedded CSS): Viết trong thẻ `<style>` bên trong phần `<head>` của tài liệu HTML. Phù hợp cho một trang HTML duy nhất.

```
``html
<head>
  <style>
    p {
      color: blue;
      font-size: 16px;
    }
  </style>
</head>
``
```

3. External CSS: Viết trong một file `.css` riêng biệt và liên kết vào tài liệu HTML bằng thẻ `<link>` trong phần `<head>`. Đây là cách phổ biến và được khuyến khích nhất vì tính module hóa, dễ quản lý và tái sử dụng.

```
``html
<head>
  <link rel="stylesheet" href="styles.css">
</head>
``
```

### 2.1.3. JavaScript (JS)

**Khái niệm và Vai trò:** JavaScript là một ngôn ngữ lập trình kịch bản (scripting language) hoặc ngôn ngữ lập trình động, đa năng, thường được sử dụng ở phía client để tạo ra các trang web tương tác. Nó cho phép thay đổi nội dung HTML và thuộc tính CSS một cách linh động, phản hồi lại các hành động của người dùng (click chuột, di chuột, gõ phím), xác thực dữ liệu trên biểu mẫu, thực hiện các yêu cầu bất đồng bộ (AJAX) đến máy chủ mà không cần tải lại toàn bộ trang, tạo hiệu ứng động, và nhiều hơn nữa.

**Đặc điểm chính:**

**Client-side Scripting:** Mã JavaScript chủ yếu được thực thi bởi trình duyệt web của người dùng.

**Dynamic Typing:** Kiểu dữ liệu của biến được xác định tại thời điểm chạy.

**Object-Oriented (Prototype-based):** Hỗ trợ lập trình hướng đối tượng dựa trên prototype.

**Event-driven:** Thực thi mã dựa trên các sự kiện do người dùng hoặc trình duyệt tạo ra (ví dụ: `onclick`, `onload`, `onmouseover`).

**DOM Manipulation:** JavaScript có thể truy cập và thay đổi Cấu trúc Đối tượng Tài liệu (Document Object Model - DOM) của một trang web, cho phép thêm, xóa, hoặc sửa đổi các phần tử HTML và nội dung của chúng.

**Asynchronous Operations:** Hỗ trợ các hoạt động bất đồng bộ thông qua Callbacks, Promises, Async/Await, cho phép thực hiện các tác vụ tốn thời gian (như gọi API) mà không làm "treo" giao diện người dùng.

**Ứng dụng phổ biến:**

- Tạo slideshow, menu thả xuống, hiệu ứng cuộn.
- Xác thực dữ liệu biểu mẫu trước khi gửi lên server.
- Tải dữ liệu động (ví dụ: cập nhật bình luận, tải thêm sản phẩm).
- Tương tác với bản đồ, biểu đồ.
- Xây dựng các Single Page Applications (SPA) phức tạp với sự trợ giúp của các thư viện/framework như React, Angular, Vue.js.

**Nhúng JavaScript vào HTML:**

- Trong thẻ ``<script>`` ở phần ``<head>`` hoặc cuối phần ``<body>`` (khuyến khích đặt cuối ``<body>`` để trang tải nội dung HTML/CSS trước).

```
```html
<body>
  <!-- Nội dung HTML -->
  <script>
    // Mã JavaScript ở đây
    console.log("Hello from inline JavaScript!");
  </script>
</body>
```
```

- Trong file `.js` bên ngoài và liên kết bằng thuộc tính `src` của thẻ ``<script>``.

```
```html
```

```
<body>
  <!-- Nội dung HTML -->
  <script src="myscript.js"></script>
</body>
...
```

## 2.2. Công nghệ phía Server (Backend)

Đây là những công nghệ chạy trên máy chủ web, chịu trách nhiệm xử lý logic nghiệp vụ, tương tác với cơ sở dữ liệu, xác thực người dùng và tạo ra nội dung động cho client.

### 2.2.1. Ngôn ngữ lập trình Python

Khái niệm: Python là một ngôn ngữ lập trình bậc cao, thông dịch, đa mục đích, được tạo ra bởi Guido van Rossum và ra mắt lần đầu vào năm 1991. Python nổi tiếng với cú pháp rõ ràng, dễ đọc, dễ học, giúp lập trình viên thể hiện ý tưởng bằng ít dòng mã hơn so với các ngôn ngữ khác như C++ hay Java.

Đặc điểm nổi bật:

- Dễ đọc, dễ học: Cú pháp gần với ngôn ngữ tự nhiên.
- Thông dịch (Interpreted): Mã được thực thi từng dòng một, giúp dễ dàng gỡ lỗi.
- Đa năng (Versatile): Sử dụng trong nhiều lĩnh vực: phát triển web, khoa học dữ liệu, trí tuệ nhân tạo, học máy, tự động hóa, phát triển game, ứng dụng desktop.
- Thư viện chuẩn lớn (Large Standard Library): Cung cấp nhiều module và hàm dựng sẵn cho các tác vụ phổ biến.
- Cộng đồng lớn và tích cực: Nguồn tài liệu phong phú, nhiều thư viện bên thứ ba mạnh mẽ.
- Hướng đối tượng (Object-Oriented Programming - OOP): Hỗ trợ đầy đủ các khái niệm OOP như lớp, đối tượng, kế thừa, đa hình.
- Hỗ trợ nhiều mô hình lập trình: Ngoài OOP, Python còn hỗ trợ lập trình thủ tục (procedural) và lập trình hàm (functional).
- Tính di động (Portable): Mã Python có thể chạy trên nhiều hệ điều hành khác nhau (Windows, macOS, Linux) mà không cần thay đổi nhiều.
- Ứng dụng trong phát triển Web: Python trở thành lựa chọn phổ biến cho phát triển backend nhờ vào các framework mạnh mẽ như Django và Flask. Chúng giúp đơn giản hóa và tăng tốc quá trình xây dựng các ứng dụng web phức tạp.

### 2.2.2. Framework Django

Khái niệm Framework: Trong phát triển phần mềm, framework là một bộ khung (hoặc một tập hợp các thư viện và công cụ) cung cấp một cấu trúc chuẩn để xây dựng ứng dụng. Nó giúp lập trình viên không phải "phát minh lại bánh xe" bằng cách cung cấp các giải pháp đã được kiểm chứng cho các vấn đề phổ biến, từ đó tăng tốc độ phát triển, đảm bảo tính nhất quán và chất lượng của mã nguồn.

**Giới thiệu Django:** Django là một framework web Python bậc cao, miễn phí và mã nguồn mở, tuân theo mô hình kiến trúc MTV (Model-Template-View), tương tự như MVC (Model-View-Controller) trong nhiều framework khác.

**Model (Mô hình):** Đại diện cho dữ liệu của ứng dụng. Django cung cấp một ORM (Object-Relational Mapper) mạnh mẽ cho phép tương tác với cơ sở dữ liệu bằng các đối tượng Python thay vì viết các câu lệnh SQL phức tạp. Model định nghĩa cấu trúc của dữ liệu, các mối quan hệ giữa chúng và các phương thức để truy cập, thao tác dữ liệu.

**Template (Giao diện mẫu):** Đại diện cho lớp trình bày, tức là cách dữ liệu được hiển thị cho người dùng. Django có một hệ thống template riêng, cho phép nhúng các biến và logic đơn giản từ Python vào các tệp HTML để tạo ra các trang web động.

**View (Khung nhìn/Bộ điều khiển):** Là thành phần xử lý logic. View nhận các yêu cầu HTTP từ người dùng, tương tác với Model để lấy hoặc lưu trữ dữ liệu, sau đó chọn một Template phù hợp và truyền dữ liệu vào đó để tạo ra phản hồi HTTP (thường là một trang HTML) gửi về cho trình duyệt của người dùng.

Các thành phần và tính năng chính của Django:

**ORM (Object-Relational Mapper):** Cho phép định nghĩa mô hình dữ liệu bằng Python classes và Django sẽ tự động ánh xạ chúng sang các bảng trong cơ sở dữ liệu. Hỗ trợ nhiều hệ quản trị CSDL khác nhau (PostgreSQL, MySQL, SQLite, Oracle).

**URL Dispatcher (Bộ định tuyến URL):** Cho phép định nghĩa các mẫu URL và ánh xạ chúng tới các View tương ứng một cách linh hoạt.

**Template Engine (Hệ thống Template):** Cung cấp cú pháp mạnh mẽ để tạo HTML động, hỗ trợ kế thừa template, tags và filters tùy chỉnh.

**Forms Framework (Khung xử lý biểu mẫu):** Giúp dễ dàng tạo, xác thực và xử lý dữ liệu từ các biểu mẫu HTML, bao gồm cả việc bảo vệ chống lại các tấn công CSRF.

**Admin Interface (Giao diện quản trị tự động):** Một trong những tính năng nổi bật nhất của Django. Nó tự động tạo ra một giao diện quản trị web chuyên nghiệp, cho phép quản lý dữ liệu của ứng dụng (thêm, sửa, xóa các bản ghi trong Model) mà không cần viết thêm nhiều mã.

**Security (Bảo mật):** Django tích hợp sẵn các biện pháp bảo vệ chống lại nhiều lỗ hổng web phổ biến như Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), SQL Injection, Clickjacking.

**Authentication System (Hệ thống xác thực):** Cung cấp sẵn cơ chế quản lý người dùng, nhóm, quyền hạn.

**Internationalization and Localization (Quốc tế hóa và Địa phương hóa):** Hỗ trợ xây dựng ứng dụng đa ngôn ngữ.

**"Batteries-included" Philosophy:** Django cung cấp rất nhiều thành phần sẵn có, giúp lập trình viên tập trung vào logic nghiệp vụ riêng của ứng dụng thay vì phải xây dựng lại các thành phần cơ bản.

Ưu điểm của Django:



- Phát triển nhanh (Rapid Development): Nhờ các thành phần sẵn có và cấu trúc rõ ràng.
- An toàn và bảo mật (Secure): Tích hợp nhiều cơ chế bảo mật.
- Khả năng mở rộng (Scalable): Có thể đáp ứng lượng truy cập lớn khi được thiết kế và triển khai đúng cách.
- Cộng đồng lớn và tài liệu tốt: Dễ dàng tìm kiếm sự hỗ trợ và học hỏi.
- Đa năng: Phù hợp cho nhiều loại ứng dụng web, từ blog đơn giản đến các hệ thống phức tạp như mạng xã hội, trang thương mại điện tử, CMS.

### 3. Tổng quan về cơ sở dữ liệu SQLite

Cơ sở dữ liệu là một thành phần quan trọng không thể thiếu trong hầu hết các ứng dụng web động, nơi nó lưu trữ và quản lý dữ liệu của ứng dụng. SQLite là một lựa chọn phổ biến cho nhiều trường hợp nhờ tính đơn giản và tiện lợi của nó.

#### 3.1. Khái niệm Cơ sở dữ liệu (CSDL) và Hệ quản trị CSDL (HQTCSDDL)

- Cơ sở dữ liệu (Database - DB): Là một tập hợp các dữ liệu có tổ chức, được lưu trữ và truy cập điện tử từ một hệ thống máy tính. Dữ liệu trong CSDL được cấu trúc theo một mô hình nhất định (ví dụ: mô hình quan hệ, mô hình đối tượng, mô hình tài liệu) để dễ dàng quản lý, truy vấn và cập nhật.
- Hệ quản trị cơ sở dữ liệu (Database Management System - DBMS):\*\* Là một phần mềm hoặc hệ thống phần mềm được thiết kế để cho phép người dùng tạo, định nghĩa, truy vấn, cập nhật và quản lý cơ sở dữ liệu. DBMS đóng vai trò trung gian giữa người dùng/ứng dụng và cơ sở dữ liệu vật lý. Ví dụ: MySQL, PostgreSQL, Microsoft SQL Server, Oracle Database, MongoDB, và SQLite.

#### 3.2. Giới thiệu về SQLite

- Định nghĩa và Nguồn gốc: SQLite là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) nhúng, gọn nhẹ, không cần máy chủ (serverless), không cần cấu hình (zero-configuration), và tự chứa trong một thư viện C. Nó được D. Richard Hipp thiết kế và phát hành lần đầu vào tháng 8 năm 2000. "Lite" trong SQLite ám chỉ sự nhẹ nhàng về tài nguyên, dễ cài đặt và quản lý.
- Đặc điểm nổi bật:
  - Serverless (Không cần máy chủ): Khác với các HQTCSDDL client-server truyền thống như MySQL hay PostgreSQL, SQLite không yêu cầu một tiến trình máy chủ riêng biệt để hoạt động. Thay vào đó, thư viện SQLite được liên kết trực tiếp vào ứng dụng, và các hàm của thư viện được gọi để thao tác trực tiếp với tệp cơ sở dữ liệu trên đĩa.
  - Self-contained (Tự chứa): Toàn bộ cơ sở dữ liệu (bao gồm định nghĩa bảng, chỉ mục, dữ liệu) được lưu trữ trong một tệp duy nhất trên máy chủ nơi ứng dụng chạy. Điều này giúp dễ dàng di chuyển, sao lưu và chia sẻ cơ sở dữ liệu.

- Zero-configuration (Không cần cấu hình): Không cần cài đặt phức tạp hay cấu hình máy chủ. Để sử dụng SQLite, bạn chỉ cần có thư viện SQLite và tạo một tệp cơ sở dữ liệu.
- Transactional (Hỗ trợ giao dịch ACID): SQLite đảm bảo các giao dịch tuân thủ đầy đủ thuộc tính ACID (Atomicity, Consistency, Isolation, Durability), giúp đảm bảo tính toàn vẹn dữ liệu ngay cả khi xảy ra lỗi hệ thống, mất điện.
- Cross-platform (Đa nền tảng): Tệp cơ sở dữ liệu SQLite hoàn toàn tương thích trên các hệ điều hành 32-bit và 64-bit, cũng như các kiến trúc big-endian và little-endian.
- Public Domain (Phạm vi công cộng): Mã nguồn của SQLite là phạm vi công cộng, nghĩa là bất kỳ ai cũng có thể sử dụng, sao chép, sửa đổi và phân phối nó cho bất kỳ mục đích nào, kể cả thương mại, mà không cần giấy phép.

### 3.3. Kiến trúc của SQLite

SQLite có kiến trúc module hóa, bao gồm các thành phần chính sau:

- Interface (Giao diện): Cung cấp các API (chủ yếu là API C) để ứng dụng tương tác với thư viện SQLite.
- Tokenizer & Parser (Bộ phân tích từ vựng và cú pháp): Phân tích các câu lệnh SQL thành các token và sau đó xây dựng cây cú pháp.
- Code Generator (Bộ sinh mã): Chuyển đổi cây cú pháp thành một dạng mã máy ảo (bytecode) đặc thù của SQLite.
- Virtual Machine (Máy ảo - VDBE - Virtual Database Engine): Thực thi mã bytecode được tạo ra để thực hiện các thao tác trên cơ sở dữ liệu.
- B-tree Module: Quản lý cấu trúc lưu trữ dữ liệu và chỉ mục dưới dạng cây B-tree.
- Pager Module: Quản lý việc đọc/ghi dữ liệu theo từng trang (page) giữa bộ nhớ và tệp cơ sở dữ liệu trên đĩa, đồng thời xử lý cache và rollback giao dịch.
- OS Interface (Giao diện Hệ điều hành): Cung cấp một lớp trừu tượng để SQLite tương tác với các tính năng của hệ điều hành (như hệ thống tệp, khóa tệp).

### 3.4. Các kiểu dữ liệu trong SQLite (Storage Classes)

SQLite sử dụng một hệ thống kiểu dữ liệu động và linh hoạt hơn so với hầu hết các RDBMS khác. Thay vì gán kiểu dữ liệu nghiêm ngặt cho cột, SQLite gán kiểu cho từng giá trị riêng lẻ. Tuy nhiên, khi định nghĩa cột, bạn vẫn có thể khai báo "kiểu dữ liệu ưu tiên" (type affinity). Các lớp lưu trữ (storage classes) cơ bản trong SQLite bao gồm:

- NULL: Giá trị là NULL.
- INTEGER: Giá trị là một số nguyên có dấu, được lưu trữ dưới dạng 1, 2, 3, 4, 6, hoặc 8 byte tùy thuộc vào độ lớn của giá trị.

- **REAL:** Giá trị là một số thực dấu phẩy động, được lưu trữ dưới dạng số thực dấu phẩy động 8 byte theo chuẩn IEEE.
- **TEXT:** Giá trị là một chuỗi văn bản, được lưu trữ bằng cách sử dụng mã hóa cơ sở dữ liệu (UTF-8, UTF-16BE hoặc UTF-16LE).
- **BLOB (Binary Large Object):** Giá trị là một khối dữ liệu nhị phân, được lưu trữ chính xác như khi nhập vào.

Type affinity (kiểu ưu tiên) cho cột bao gồm: 'TEXT', 'NUMERIC', 'INTEGER', 'REAL', 'BLOB' (hoặc 'NONE'). Khi bạn chèn dữ liệu, SQLite sẽ cố gắng chuyển đổi giá trị về kiểu ưu tiên của cột đó nếu có thể.

### 3.5. Ưu điểm của SQLite

- Đơn giản và dễ sử dụng: Không cần cài đặt, không cần quản trị viên cơ sở dữ liệu.
- Gọn nhẹ: Thư viện nhỏ, tệp cơ sở dữ liệu đơn lẻ.
- Tính di động cao: Dễ dàng sao chép, di chuyển, chia sẻ tệp cơ sở dữ liệu.
- Không cần máy chủ riêng: Tiết kiệm tài nguyên hệ thống.
- Nhanh chóng cho các thao tác đọc và ghi cục bộ: Hiệu suất tốt cho các ứng dụng có lượng truy cập vừa phải và chủ yếu là đọc.
- Lựa chọn tốt cho phát triển và kiểm thử: Do tính đơn giản, SQLite thường được sử dụng làm cơ sở dữ liệu mặc định trong giai đoạn phát triển của nhiều framework web (bao gồm cả Django).
- Tuyệt vời cho các ứng dụng nhúng: Sử dụng trong điện thoại di động, trình duyệt web, thiết bị IoT.
- Miễn phí và mã nguồn mở.

### 3.6. Nhược điểm và Trường hợp không nên dùng SQLite

Mặc dù có nhiều ưu điểm, SQLite cũng có những hạn chế và không phải lúc nào cũng là lựa chọn tốt nhất:

- Khả năng chịu tải ghi đồng thời thấp: SQLite sử dụng cơ chế khóa ở cấp độ tệp, điều này có nghĩa là tại một thời điểm chỉ có một tiến trình có thể ghi vào cơ sở dữ liệu. Điều này làm hạn chế khả năng mở rộng cho các ứng dụng có lượng ghi lớn và đồng thời cao.
- Không phù hợp cho các ứng dụng client-server quy mô lớn: Vì là serverless, nó không được thiết kế để xử lý nhiều kết nối mạng đồng thời từ nhiều client như các HQTCSDB chuyên dụng (MySQL, PostgreSQL).
- Ít tính năng nâng cao: So với các RDBMS lớn, SQLite thiếu một số tính năng nâng cao như stored procedures, quyền người dùng phức tạp, replication.
- Kích thước cơ sở dữ liệu lớn có thể ảnh hưởng hiệu suất: Mặc dù SQLite có thể xử lý các tệp cơ sở dữ liệu rất lớn (terabytes), hiệu suất có thể giảm khi kích thước tệp tăng lên đáng kể, đặc biệt là trên các hệ thống tệp mạng.
- Không có quản lý người dùng và bảo mật mạng tích hợp: Vì là tệp cục bộ, bảo mật phụ thuộc vào quyền truy cập tệp của hệ điều hành.

Do đó, SQLite thường không được khuyến nghị cho các ứng dụng web có lượng truy cập cao, yêu cầu ghi đồng thời lớn, hoặc cần các tính năng quản trị và bảo mật phức tạp của một HQTCSĐL client-server.

### **3.7. SQLite trong phát triển ứng dụng Django**

Django hỗ trợ SQLite "out-of-the-box" và sử dụng nó làm cơ sở dữ liệu mặc định cho các dự án mới. Điều này rất tiện lợi cho việc bắt đầu phát triển nhanh chóng mà không cần phải cài đặt và cấu hình một hệ quản trị cơ sở dữ liệu riêng biệt. Lập trình viên có thể dễ dàng phát triển và kiểm thử ứng dụng với SQLite, sau đó khi cần triển khai lên môi trường production với yêu cầu cao hơn, có thể chuyển sang các HQTCSĐL mạnh mẽ hơn như PostgreSQL hoặc MySQL bằng cách thay đổi cấu hình trong file `settings.py` của Django mà không cần thay đổi nhiều trong mã nguồn Model (nhờ vào tính năng ORM của Django).

## CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 1. Glossary list

Thuật ngữ	Mô tả
Đăng ký tài khoản	Quá trình người dùng tạo tài khoản mới trên hệ thống.
Xác thực email	Gửi mã xác thực đến email người dùng để xác nhận tài khoản.
Đăng nhập	Người dùng nhập thông tin để truy cập hệ thống.
Đăng xuất	Người dùng kết thúc phiên làm việc, rời khỏi hệ thống.
Quên mật khẩu	Chức năng gửi mật khẩu mới hoặc liên kết đặt lại mật khẩu qua email cho người dùng.
Chỉnh sửa hồ sơ cá nhân	Người dùng cập nhật thông tin cá nhân như tên, email, ảnh đại diện, mật khẩu, v.v.
Quản trị viên	Người có quyền quản lý toàn bộ hệ thống, bao gồm người dùng, khóa học, blog, v.v.
Giảng viên	Người tạo, chỉnh sửa, xóa và quản lý các khóa học, bài giảng.
Học viên	Người đăng ký và tham gia các khóa học.
Tạo khóa học	Chức năng cho phép giảng viên hoặc admin tạo mới một khóa học với các thông tin chi tiết.
Chỉnh sửa khóa học	Cập nhật thông tin khóa học đã tạo (tên, mô tả, ảnh, chương, bài học, v.v.).
Xóa khóa học	Xóa khóa học khỏi hệ thống sau khi xác nhận.
Xem danh sách khóa học	Hiển thị tất cả các khóa học hiện có, hỗ trợ lọc, tìm kiếm theo nhiều tiêu chí.
Xem chi tiết khóa học	Hiển thị thông tin chi tiết về một khóa học

	cụ thể.
Đăng ký khóa học	Người dùng đăng ký tham gia một khóa học, trở thành học viên của khóa học đó.
Học bài giảng/quiz	Người dùng học các bài lý thuyết hoặc làm quiz trong khóa học.
Quiz	Bài kiểm tra trắc nghiệm trong mỗi khóa học.
Đăng bài blog	Người dùng tạo và đăng tải bài viết blog mới lên hệ thống.
Xem danh sách blog	Xem danh sách các bài blog đã đăng trên hệ thống.
Xem chi tiết blog	Xem nội dung chi tiết, bình luận, số lượt xem của một bài blog.
Bình luận blog	Người dùng nhập và gửi bình luận dưới mỗi bài blog.
Tìm kiếm blog/khóa học	Tìm kiếm các bài blog hoặc khóa học theo từ khóa, tiêu chí lọc.
Chỉnh sửa blog	Người dùng hoặc admin cập nhật nội dung bài blog đã đăng.
Xóa blog	Người dùng hoặc admin xóa bài blog khỏi hệ thống.
Quản lý người dùng	Chức năng cho phép admin xem, chỉnh sửa, xóa tài khoản người dùng.
Quản lý blog	Chức năng cho phép admin chỉnh sửa, xóa các bài blog hoặc bình luận vi phạm.
Quản lý khóa học	Chức năng cho phép admin hoặc giảng viên tạo, chỉnh sửa, xóa, quản lý khóa học.
Sao lưu dữ liệu	Quá trình lưu trữ dữ liệu hệ thống định kỳ để đảm bảo an toàn và khôi phục khi có sự cố.
Gửi email tự động	Hệ thống tự động gửi email xác thực, quên mật khẩu, thông báo cho người dùng.

Phân quyền truy cập	Cơ chế kiểm soát quyền hạn của từng loại người dùng (admin, giảng viên, học viên).
Khả năng mở rộng	Hệ thống dễ dàng bổ sung module hoặc chức năng mới trong tương lai.

## 2. Mô tả hệ thống

Hệ thống LMS Website là một nền tảng học trực tuyến cho phép người dùng đăng ký, tham gia các khóa học, làm bài kiểm tra, tương tác qua blog và quản lý thông tin cá nhân. Hệ thống hỗ trợ nhiều vai trò người dùng như học viên, giảng viên và quản trị viên với các chức năng phù hợp từng vai trò.

### 2.1. Phạm vi phần mềm

- Quản lý người dùng: Đăng ký, đăng nhập, quản lý hồ sơ cá nhân, quên mật khẩu.
- Quản lý khóa học: Tạo, chỉnh sửa, xóa, đăng ký, học bài giảng, làm quiz.
- Quản lý blog: Đăng bài, xem, bình luận, tìm kiếm blog.
- Quản lý liên hệ: Gửi thông tin liên hệ, hỗ trợ người dùng.
- Quản trị viên có thể quản lý toàn bộ hệ thống, giảng viên quản lý khóa học của mình, học viên tham gia học tập.

### 2.2. Người dùng và chức năng của người dùng

- Học viên (User):
  - + Đăng ký, đăng nhập, chỉnh sửa hồ sơ, quên mật khẩu
  - + Xem, đăng ký, học các khóa học, làm quiz
  - + Xem, đăng bài, bình luận blog
  - + Gửi thông tin liên hệ
- Giảng viên (Instructor):
  - + Tất cả chức năng của học viên
  - + Tạo, chỉnh sửa, xóa khóa học do mình tạo

- Quản trị viên (Admin):

+ Toàn quyền quản lý hệ thống, quản lý người dùng, khóa học, blog

### **2.3. Thông tin các đối tượng cần xử lý**

- Người dùng (User): Thông tin cá nhân, vai trò, email, mật khẩu, avatar, mô tả bản thân

- Khóa học (Course): Tên, mô tả, ảnh, giá, thời lượng, cấp độ, ngôn ngữ, người tạo, danh sách học viên, chương/bài học

- Bài học (Lesson): Tiêu đề, loại (lý thuyết/quiz), nội dung, video, câu hỏi

- Blog: Tiêu đề, nội dung, tác giả, ngày đăng, bình luận

- Bình luận (Comment): Nội dung, người viết, thời gian

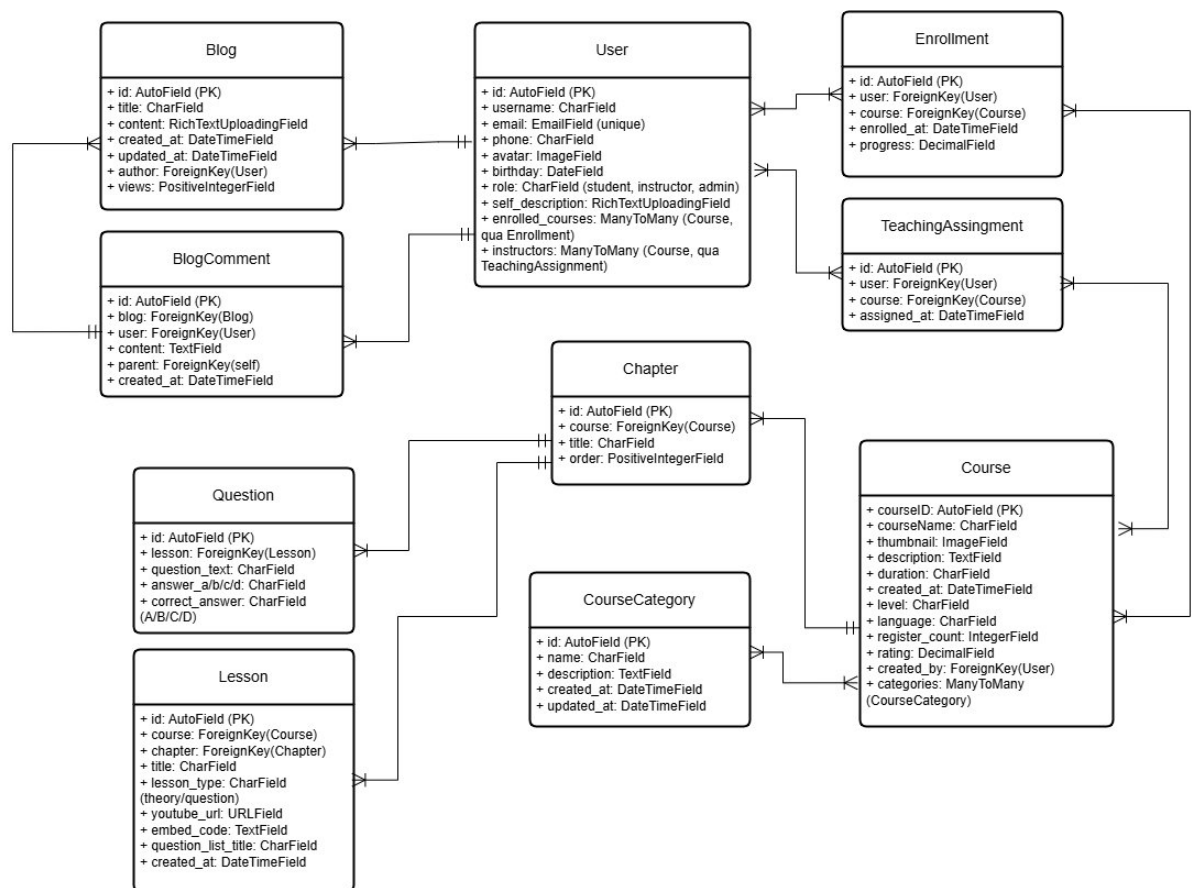
- Đăng ký khóa học (Enrollment): Người dùng, khóa học, tiến độ

### **2.4. Quan hệ giữa các đối tượng cần xử lý**

- Một khóa học (Course) có thể bao gồm nhiều bài học (Lesson), mỗi bài học thuộc về một khóa học duy nhất.
- Một người dùng (User) có thể đăng ký nhiều khóa học (Enrollment), và một khóa học có thể có nhiều học viên (User) đăng ký.
- Một người dùng (User) có thể là tác giả của nhiều bài blog (Blog), mỗi blog chỉ có một tác giả.
- Một bài blog (Blog) có thể có nhiều bình luận (Comment), mỗi bình luận thuộc về một blog duy nhất và do một người dùng viết.
- Một bài học (Lesson) có thể chứa nhiều câu hỏi (Question), mỗi câu hỏi thuộc về một bài học.
- Một người dùng (User) có thể là giảng viên của nhiều khóa học (Course), mỗi khóa học có một giảng viên chính (created\_by).
- Một người dùng (User) có thể là quản trị viên, giảng viên hoặc học viên, mỗi vai trò có quyền thao tác khác nhau trên các đối tượng.
- Một khóa học (Course) có thể thuộc nhiều danh mục (Category), và một danh mục có thể chứa nhiều khóa học.
- Một học viên (User) có thể có nhiều tiến độ học tập (progress) ở các khóa học khác nhau.



## 2.5. Sơ đồ quan hệ thực thể:



## Bảng mô tả cơ sở dữ liệu

- Bảng Users

Column	Datatype	Description
id	AutoField (PK)	Khóa chính, mã người dùng duy nhất
username	CharField	Tên đăng nhập
email	EmailField	Địa chỉ email của người dùng (duy nhất)
phone	CharField	Số điện thoại của người dùng
avatar	ImageField	Ảnh đại diện người dùng
birthday	DateField	Ngày sinh của người dùng
role	CharField	Vai trò người dùng(Student, Instructor, Admin)

self_description	RichTextUploadingField	Mô tả bản thân của người dùng
enrolled_courses	ManyToMany (Course, qua Enrollment)	Các khóa học người dùng đã đăng ký
instructors	ManyToMany (Course, qua TeachingAssignment)	Các khóa học người dùng giảng dạy

- Bảng Courses

Column	Datatype	Description
courseID	AutoField (PK)	Khóa chính, mã khóa học duy nhất
coursename	CharField	Tên khóa học
thumbnail	ImageField	Ảnh đại diện khóa học
description	TextField	Mô tả khóa học
duration	CharField	Thời lượng khóa học
created_at	DateField	Ngày tạo khóa học
level	CharField	Cấp độ khóa học (ví dụ: cơ bản, nâng cao)
language	CharField	Ngôn ngữ giảng dạy
register_count	IntegerField	Số lượng học viên đăng ký
created_by	ForeignKey(User)	Người tạo khóa học
rating	DecimalField	Đánh giá khóa học
categories	ManyToMany (CourseCategory)	Danh mục khóa học

- Bảng Lessons

Column	Datatype	Description
id	AutoField (PK)	Khóa chính, mã bài học duy nhất

course	ForeignKey(Course)	Khóa học chứa bài học
chapter	ForeignKey(Chapter)	Chương chứa bài học
title	TextField	Tiêu đề bài học
lesson_type	CharField	Loại bài học (lý thuyết/câu hỏi)
youtube_url	DateField	Đường dẫn video YouTube
embed_code	CharField	Mã nhúng video
question_list_title	CharField	Tiêu đề danh sách câu hỏi
created_at	IntegerField	Ngày tạo bài học

- Bảng Chapters

Column	Datatype	Description
id	AutoField (PK)	Khóa chính, mã chương duy nhất
course	ForeignKey(Course)	Khóa học chứa chương
tittle	CharField	Tiêu đề chương
order	PositiveIntegerField	Thứ tự chương trong khóa học

- Bảng Questions

Column	Datatype	Description
id	AutoField (PK)	Khóa chính, mã câu hỏi duy nhất
lesson	ForeignKey(Lesson)	Bài học chứa câu hỏi
question_text	CharField	Nội dung câu hỏi
answer_a	CharField	Đáp án A
answer_b	CharField	Đáp án B
answer_c	CharField	Đáp án C

answer_d	CharField	Đáp án D
correct_answer	CharField	Đáp án đúng (A/B/C/D)

- Bảng Enrollments

Column	Datatype	Description
id	AutoField (PK)	Khóa chính, mã đăng ký duy nhất
user	ForeignKey(Lesson)	Người dùng đăng ký
course	ForeignKey(Course)	Khóa học đã đăng ký
enrolled_at	DateTimeField	Ngày đăng ký
progress	CharField	Tiến trình học của người dùng

- Bảng TeachingAssignments

Column	Datatype	Description
id	AutoField (PK)	Khóa chính, mã tạo khóa học duy nhất
user	ForeignKey(Lesson)	Giảng viên tạo khóa học
course	ForeignKey(Course)	Khóa học được tạo
assigned_at	DateTimeField	Ngày tạo khóa học

- Bảng CourseCategory

Column	Datatype	Description
id	AutoField (PK)	Khóa chính, mã danh mục duy nhất
name	CharField	Tên danh mục

description	TextField	Mô tả danh mục
created_at	DateTimeField	Ngày tạo danh mục
updated_at	DateTimeField	Ngày cập nhật danh mục

- Bảng Blogs

Column	Datatype	Description
id	AutoField (PK)	Khóa chính, mã blog duy nhất
title	CharField	Tiêu đề bài blog
content	RichTextUploadingField	Nội dung bài blog
created_at	DateTimeField	Ngày tạo bài blog
updated_at	DateTimeField	Ngày cập nhật bài blog
author	ForeignKey(User)	Tác giả bài blog
views	PositiveIntegerField	Số lượt xem

- Bảng BlogComments

Column	Datatype	Description
id	AutoField (PK)	Khóa chính, mã bình luận duy nhất
blog	ForeignKey(Blog)	Bài blog được bình luận
user	ForeignKey(User)	Người bình luận
content	TextField	Nội dung bình luận
parent	ForeignKey(self)	Bình luận cha (nếu là trả lời bình luận khác)
create_at	DateTimeField	Ngày bình luận

## **CHƯƠNG IV: MÔ TẢ NGHIỆP VỤ CHI TIẾT CỦA CÁC MODULE**

### **1. Mô tả các module**

#### **1.1. Module Quản lý người dùng**

- Đăng ký tài khoản, xác thực email, đăng nhập, đăng xuất, quên mật khẩu (gửi mật khẩu mới qua email), chỉnh sửa hồ sơ cá nhân.
- Quản trị viên có thể quản lý danh sách người dùng.

#### **1.2. Module Quản lý khóa học**

- Tạo khóa học: Giảng viên hoặc admin chọn chức năng "Tạo khóa học". Giao diện form nhập thông tin khóa học hiện ra với các trường: tên, mô tả, ảnh, giá, thời lượng, cấp độ, ngôn ngữ, danh mục, chương/bài học. Sau khi nhập đầy đủ thông tin, nhấn "Lưu" để tạo khóa học mới. Hệ thống kiểm tra trùng tên, thông tin hợp lệ trước khi lưu.
- Chỉnh sửa khóa học: Giảng viên hoặc admin tìm kiếm khóa học do mình tạo, chọn chức năng "Chỉnh sửa". Giao diện cập nhật thông tin hiện ra với dữ liệu cũ đã có sẵn. Giảng viên cập nhật các trường cần thiết và nhấn "Cập nhật". Hệ thống kiểm tra quyền sở hữu và hợp lệ trước khi lưu.
- Xóa khóa học: Giảng viên hoặc admin tìm kiếm khóa học do mình tạo, chọn chức năng "Xóa". Hệ thống hiển thị hộp thoại xác nhận. Sau khi xác nhận, hệ thống xóa khóa học khỏi cơ sở dữ liệu.
- Xem danh sách khóa học: Người dùng truy cập tab "Khóa học" để xem danh sách các khóa học hiện có. Có thể lọc, tìm kiếm theo tên, danh mục, cấp độ, v.v.
- Xem chi tiết khóa học: Người dùng chọn một khóa học từ danh sách để xem thông tin chi tiết, bao gồm mô tả, chương/bài học, giảng viên, đánh giá, số lượng học viên, v.v.
- Đăng ký khóa học: Người dùng nhấn nút "Đăng ký" trên trang chi tiết khóa học. Hệ thống kiểm tra điều kiện (đã đăng nhập, chưa đăng ký khóa học này), sau đó thêm người dùng vào danh sách học viên của khóa học.
- Học bài giảng/quiz: Sau khi đăng ký, người dùng truy cập các bài học lý thuyết hoặc quiz trong khóa học. Nếu là quiz, người dùng làm bài và nộp, hệ thống chấm điểm và cập nhật tiến độ học tập.

- Xem tiến độ học tập: Người dùng có thể xem tiến độ học tập của mình trong từng khóa học (số bài đã hoàn thành, điểm quiz, v.v.).

### **1.3. Module Quản lý blog**

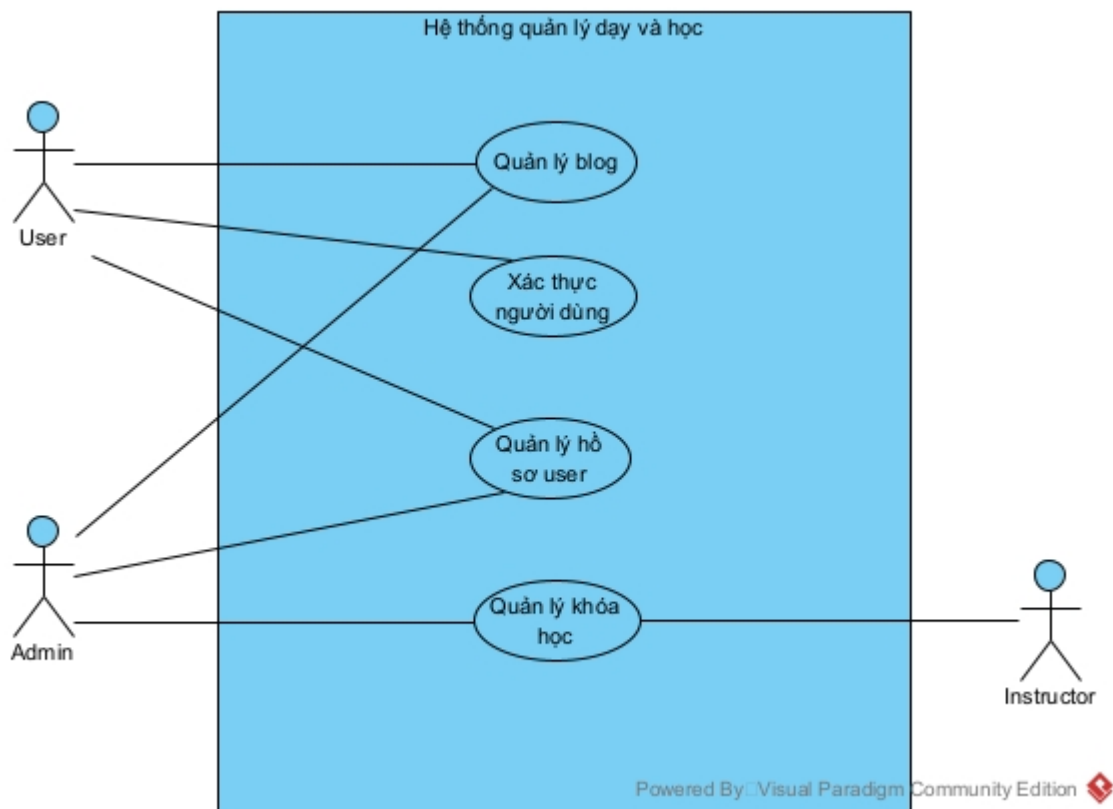
- Đăng bài blog: Người dùng chọn chức năng "Đăng bài blog". Giao diện form nhập tiêu đề, nội dung, ảnh (nếu có). Sau khi nhập xong, nhấn "Đăng" để lưu bài blog mới. Hệ thống kiểm tra hợp lệ trước khi lưu.
- Xem blog: Người dùng truy cập tab "Blog" để xem danh sách các bài blog. Có thể tìm kiếm, lọc theo từ khóa, tác giả, ngày đăng.
- Xem chi tiết blog: Người dùng chọn một bài blog để xem nội dung chi tiết, bình luận, số lượt xem, v.v.
- Bình luận blog: Người dùng nhập nội dung bình luận vào form dưới mỗi bài blog, nhấn "Gửi" để lưu bình luận. Hệ thống kiểm tra đã đăng nhập và nội dung hợp lệ trước khi lưu.
- Tìm kiếm blog: Người dùng nhập từ khóa vào ô tìm kiếm để lọc các bài blog phù hợp.
- Quản trị viên có thể chỉnh sửa, xóa bất kỳ bài blog hoặc bình luận nào nếu phát hiện vi phạm.

### **2. Các yêu cầu phi chức năng**

- Hệ thống phải bảo mật thông tin người dùng (mã hóa mật khẩu, xác thực email, phân quyền truy cập).
- Giao diện thân thiện, dễ sử dụng trên máy tính.
- Hệ thống phải đảm bảo hiệu năng khi có nhiều người dùng truy cập đồng thời.
- Dữ liệu phải được sao lưu định kỳ, đảm bảo an toàn và khôi phục khi có sự cố.
- Hệ thống hỗ trợ gửi email tự động (xác thực, quên mật khẩu, thông báo).
- Đảm bảo khả năng mở rộng để bổ sung thêm module hoặc chức năng mới trong tương lai.

### **3. Xây dựng các chức năng hệ thống**

### 3.1. Biểu đồ Usecase tổng quan

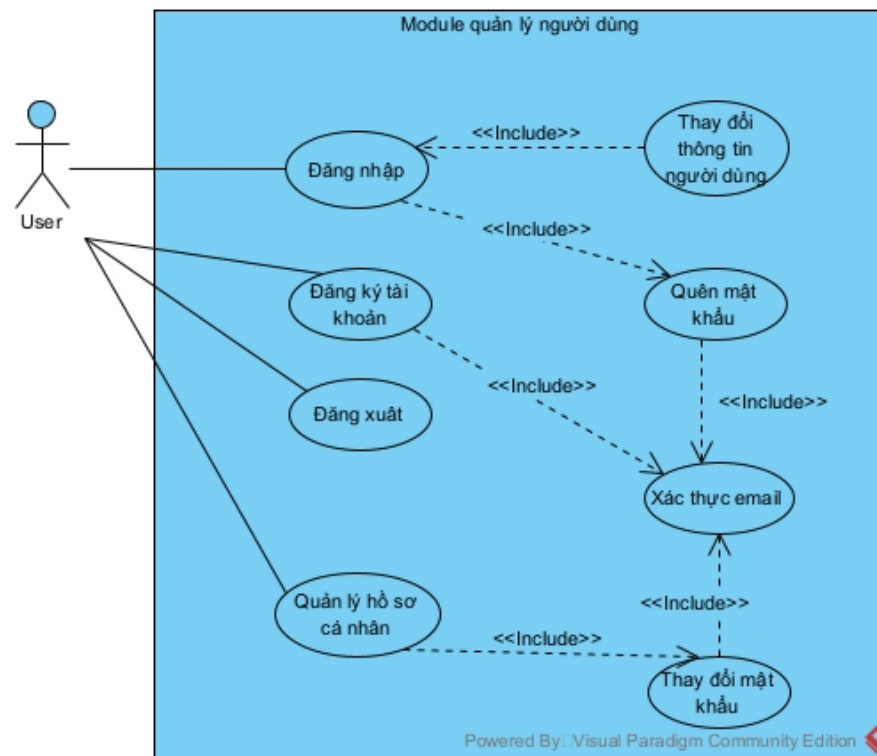


### 3.2. Thiết kế các module

#### 3.2.1. Module Quản lý người dùng

Biểu đồ usecase của module





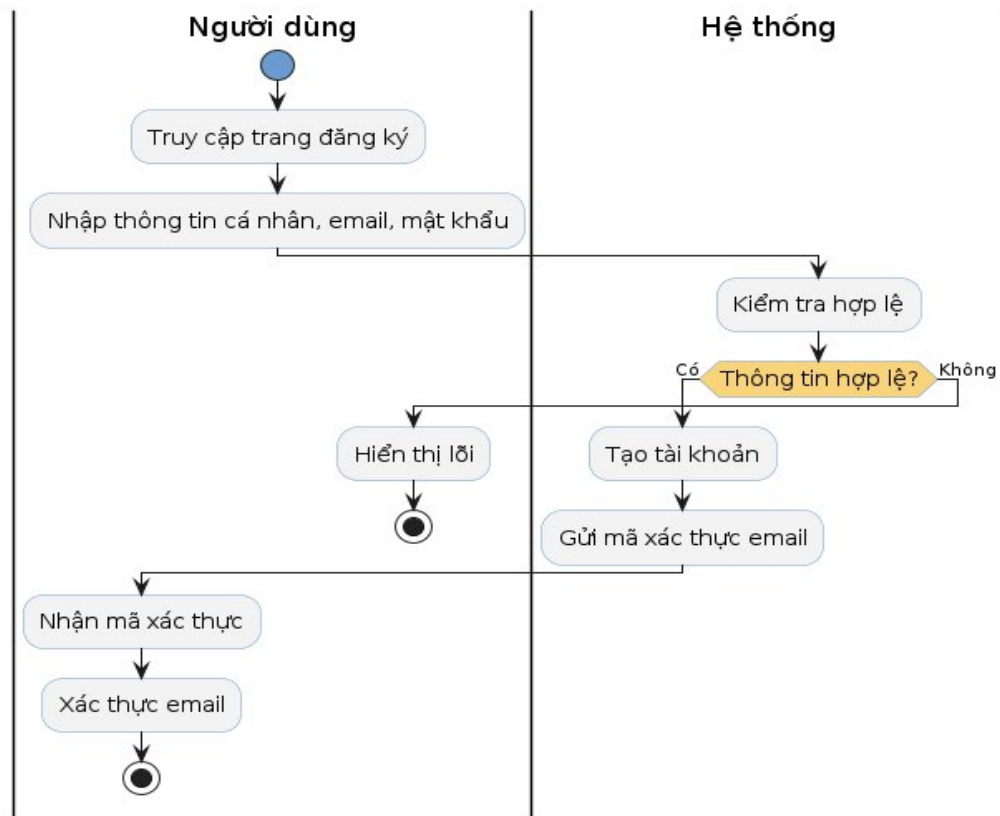
### 3.2.1.1 Usecase đăng ký tài khoản

- Đặc tả Usecase

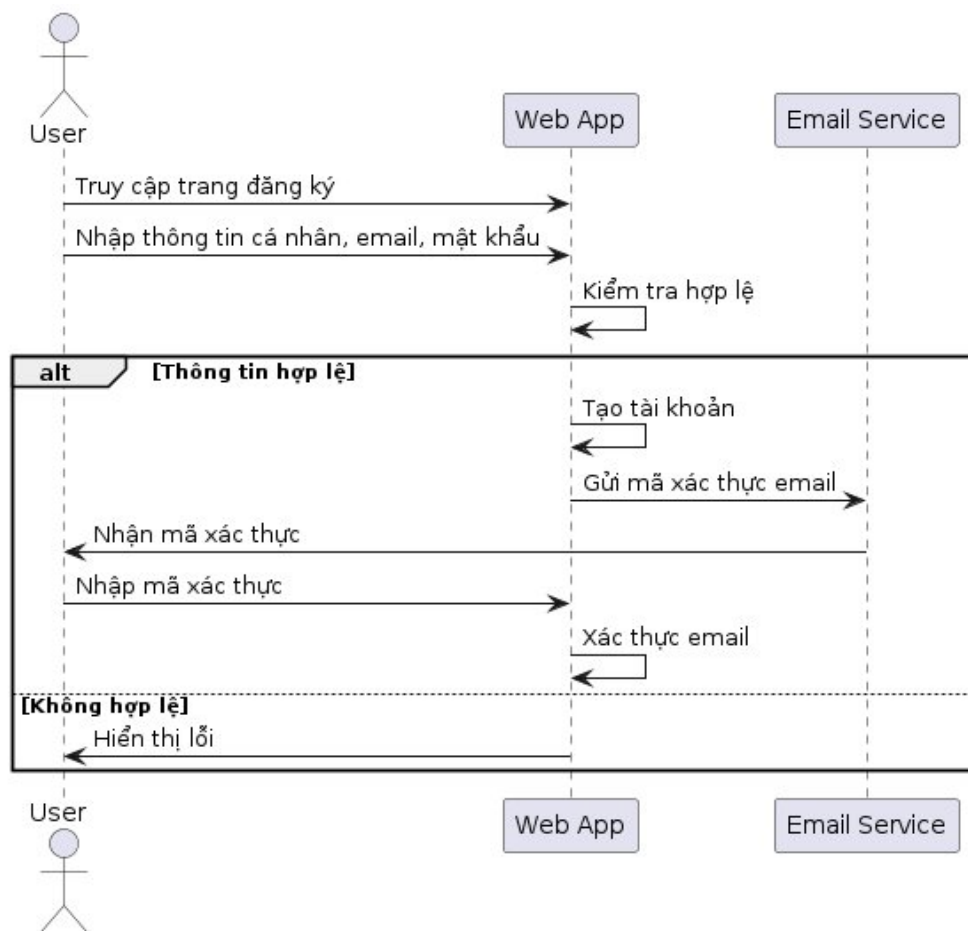
Usecase specification	
Use Case ID	UC01
Name	Đăng ký tài khoản
Created By:	Trần Văn Hoàng
Primary Actor	Người dùng
Secondary Actor	Hệ thống, người quản trị hệ thống
Trigger	Người dùng truy cập trang đăng ký
Description	Người dùng nhập thông tin để tạo tài khoản mới
Pre-condition	Người dùng chưa có tài khoản
Post-condition	Tài khoản mới được tạo, gửi mã xác thực email
Normal flow	<ol style="list-style-type: none"> <li>1. Người dùng truy cập trang đăng ký</li> <li>2. Nhập thông tin cá nhân, email, mật khẩu</li> <li>3. Hệ thống kiểm tra hợp lệ và tạo tài khoản</li> </ol>

	4. Gửi mã xác thực email 5. Người dùng xác thực email
Alternative flow	- Nếu email đã tồn tại, hiển thị lỗi - Nếu mật khẩu không khớp, hiển thị lỗi
Exception flow	- Lỗi kết nối, lỗi gửi email xác thực
Business rules	- Email phải là duy nhất - Mật khẩu phải đủ mạnh

## Biểu đồ hoạt động



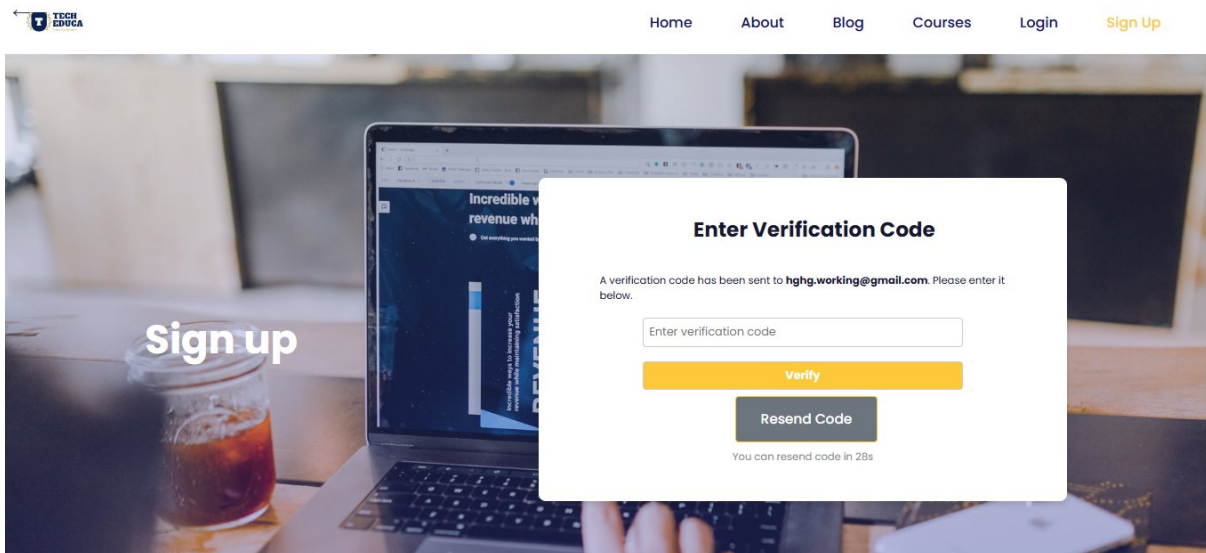
## Biểu đồ tuần tự



## Chi tiết giao diện

The screenshot shows a web application interface with a navigation bar at the top containing links: Home, About, Blog, Courses, Login, and Sign Up. The main content area features a 'Sign up' overlay form titled 'Create free account now'. The form includes input fields for first name (Tran), last name (Van Hoang), username (hoangtv12), email (bustren12@gmail.com), a role dropdown menu (Instructor), and two password fields. A yellow 'sign up' button is at the bottom of the form. Below the button, there is a link: 'Already have an account? [Login here](#)'.

Trang đăng ký với form nhập tên, email, mật khẩu, xác nhận mật khẩu. Có thông báo lỗi nếu nhập thiếu hoặc sai định dạng. Sau khi đăng ký, hiển thị thông báo xác thực email.



Giao diện nhập mã xác thực từ email. Hiện thị trạng thái xác thực thành công/thất bại.

Frontend: signup.html

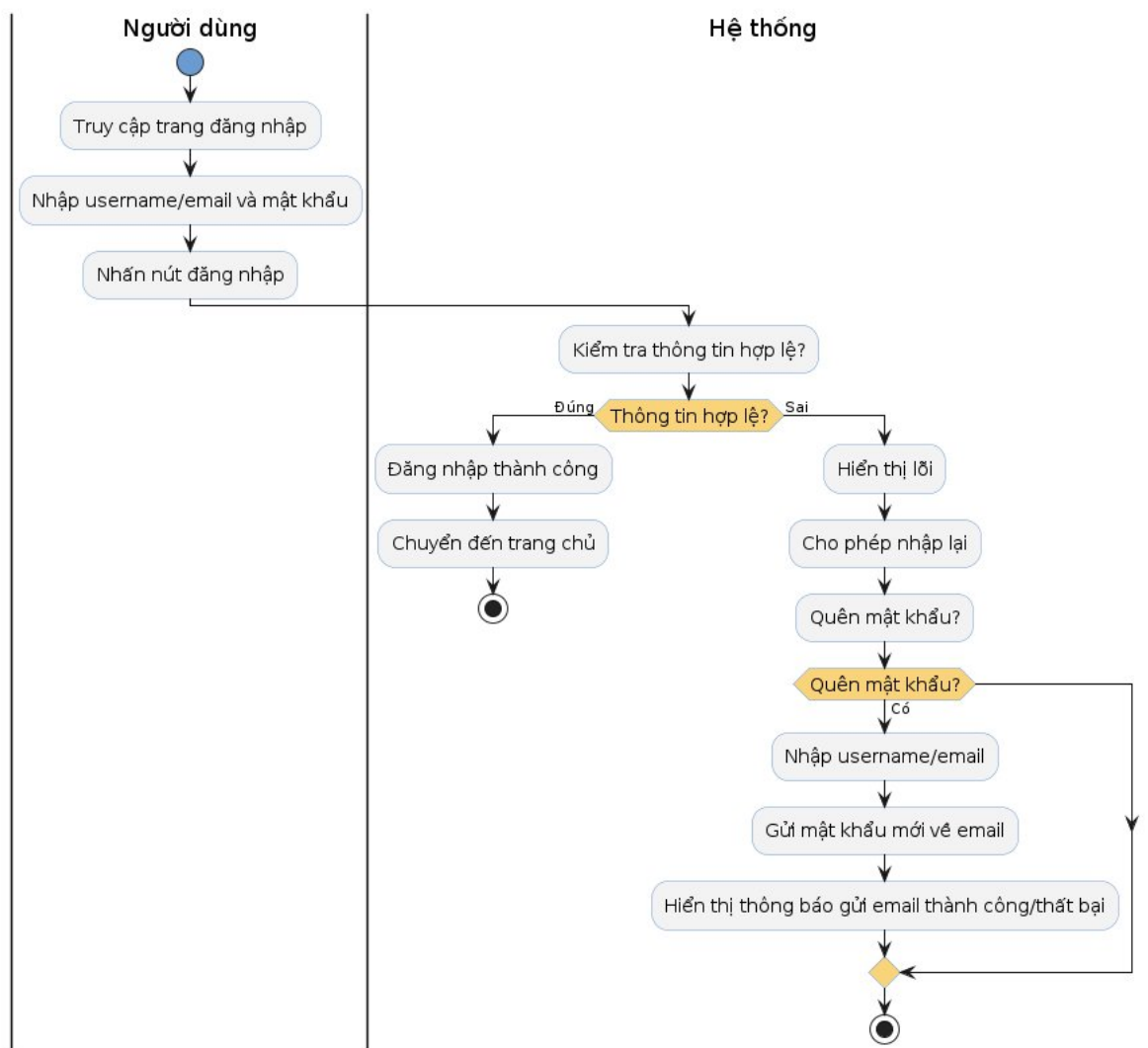
Backend: signup\_view

### 3.2.1.2. Usecase đăng nhập tài khoản

Usecase specification	
Use Case ID	UC02
Name	Đăng nhập
Created By:	Trần Văn Hoàng
Primary Actor	Người dùng, Người quản trị hệ thống
Secondary Actor	Hệ thống
Trigger	Người dùng truy cập trang đăng nhập
Description	Người dùng nhập thông tin để đăng nhập vào hệ thống
Pre-condition	Đã có tài khoản hợp lệ
Post-condition	Đăng nhập thành công, chuyển đến trang chủ
Normal flow	1. Người dùng nhập username/email và mật khẩu

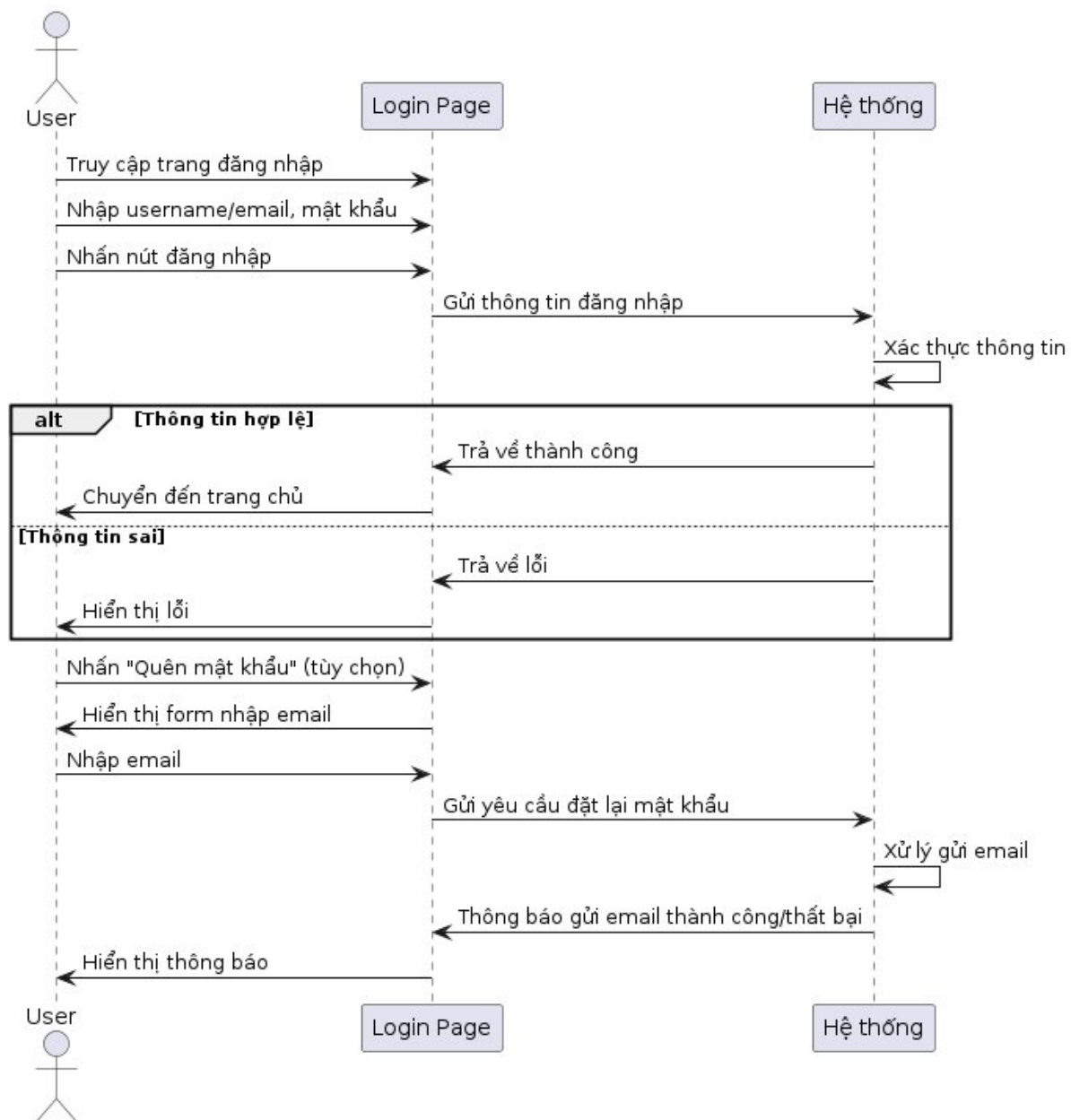
	2. Hệ thống xác thực thông tin 3. Nếu đúng, đăng nhập thành công 4. Nếu sai, hiển thị lỗi 5. Nếu quên mật khẩu, xác nhận lại username và email 6. Gửi mật khẩu mới về email
Alternative flow	- Nếu sai thông tin, hiển thị lỗi - Nếu không nhớ mật khẩu, xác thực lại
Exception flow	- Lỗi kết nối, tài khoản không tồn tại
Business rules	- Tài khoản phải được xác thực email

## Biểu đồ hoạt động



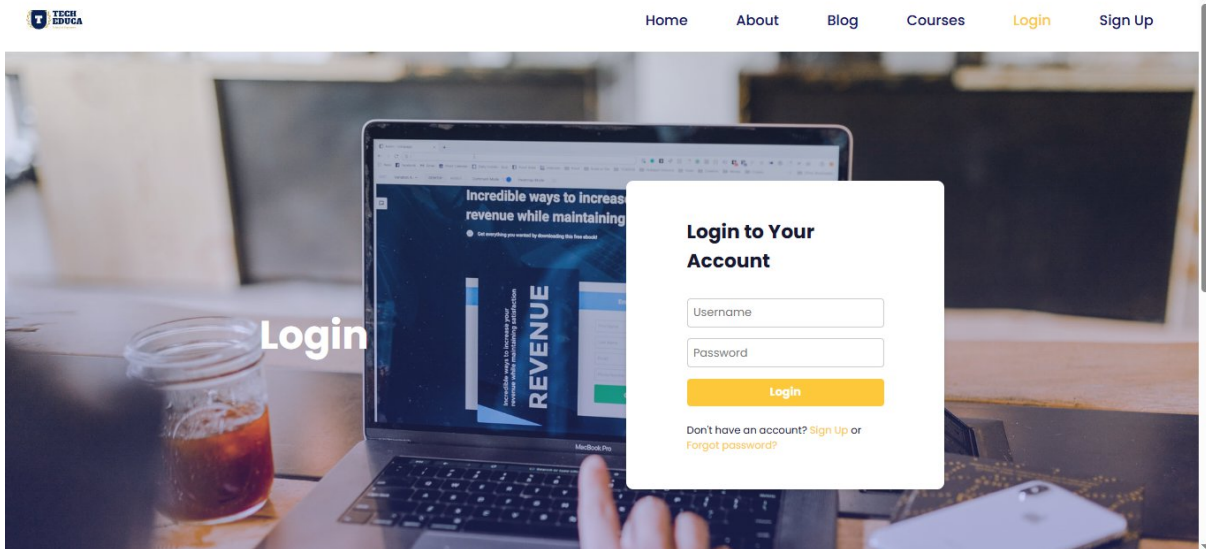


## Biểu đồ tuần tự

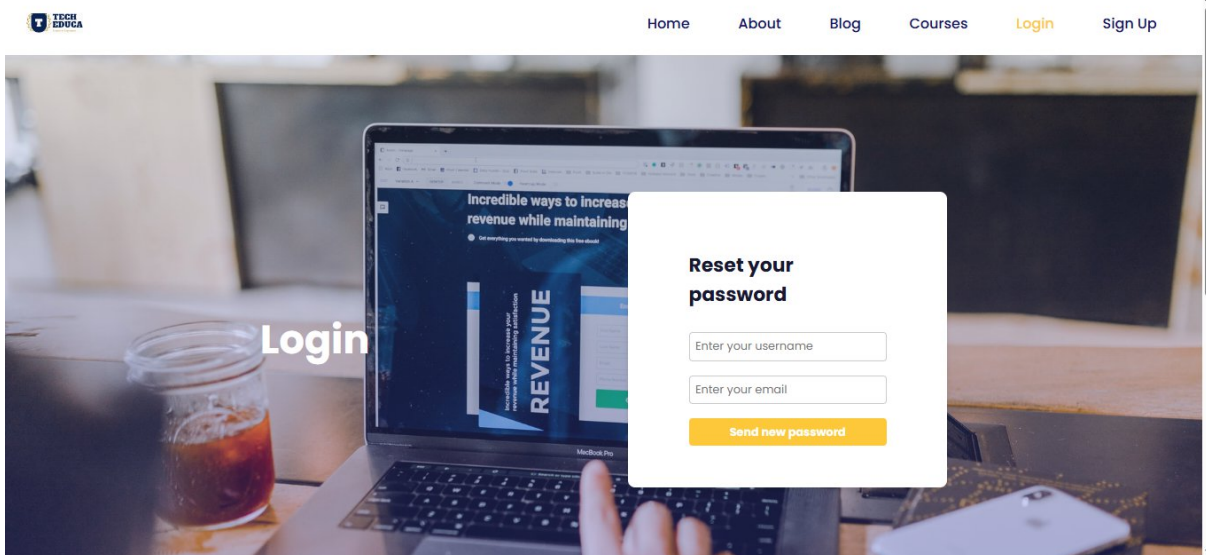


## Chi tiết giao diện





Trang đăng nhập với form nhập username/email và mật khẩu. Có nút "forgot password". Hiện thị lỗi nếu thông tin không đúng.



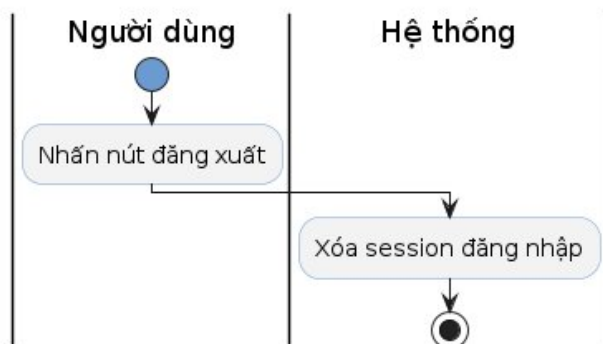
Form nhập email để nhận link/mã đặt lại mật khẩu. Thông báo gửi email thành công/thất bại

### 3.2.1.5. Usecase đăng xuất

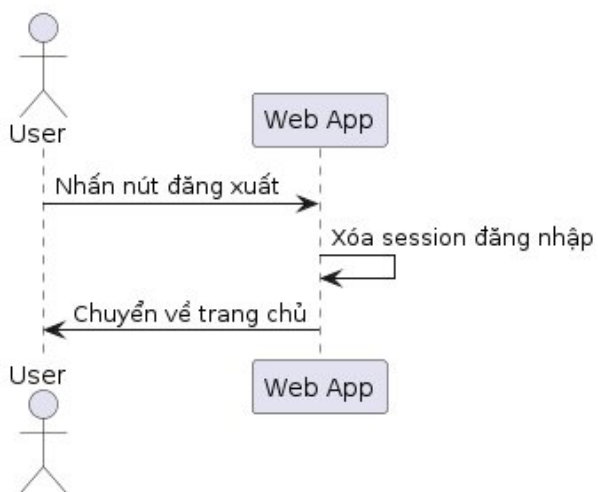
Usecase specification	
Use Case ID	UC03
Name	Đăng xuất
Created By:	Trần Văn Hoàng
Primary Actor	Người dùng

Secondary Actor	Hệ thống
Trigger	Người dùng chọn chức năng đăng xuất
Description	Người dùng đăng xuất khỏi hệ thống
Pre-condition	Đã đăng nhập
Post-condition	Phiên đăng nhập kết thúc
Normal flow	1. Người dùng nhấn nút đăng xuất 2. Hệ thống xóa session đăng nhập
Alternative flow	Không
Exception flow	Không
Business rules	Không

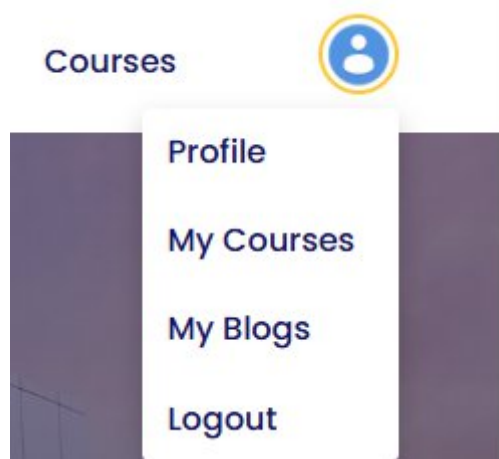
### Biểu đồ hoạt động



### Biểu đồ tuần tự



## Chi tiết giao diện



Nhấn nút “Logout” trong dropdown hiển thị ở dưới sẽ đăng xuất tài khoản.

Frontend: navigation.html

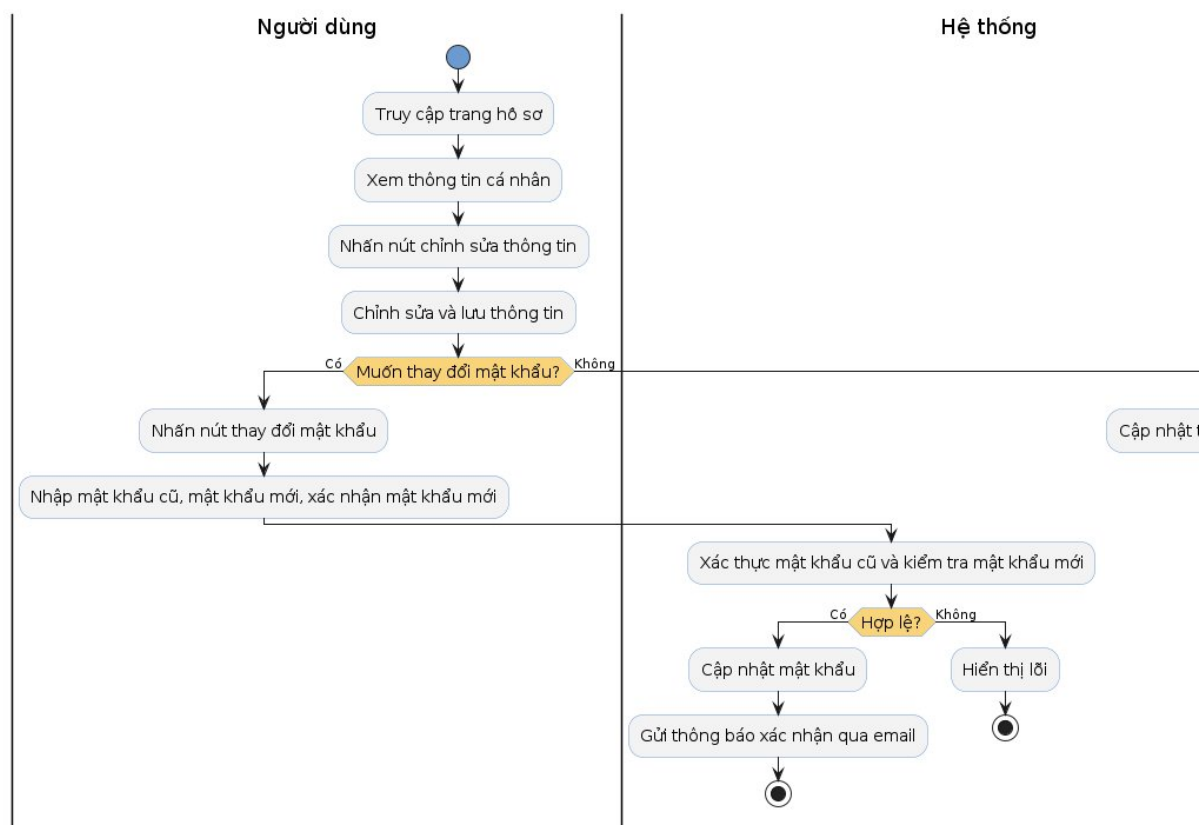
Backend: logout\_view

### 3.2.1.4. Usecase Quản lý hồ sơ cá nhân(bao gồm cả mật khẩu)

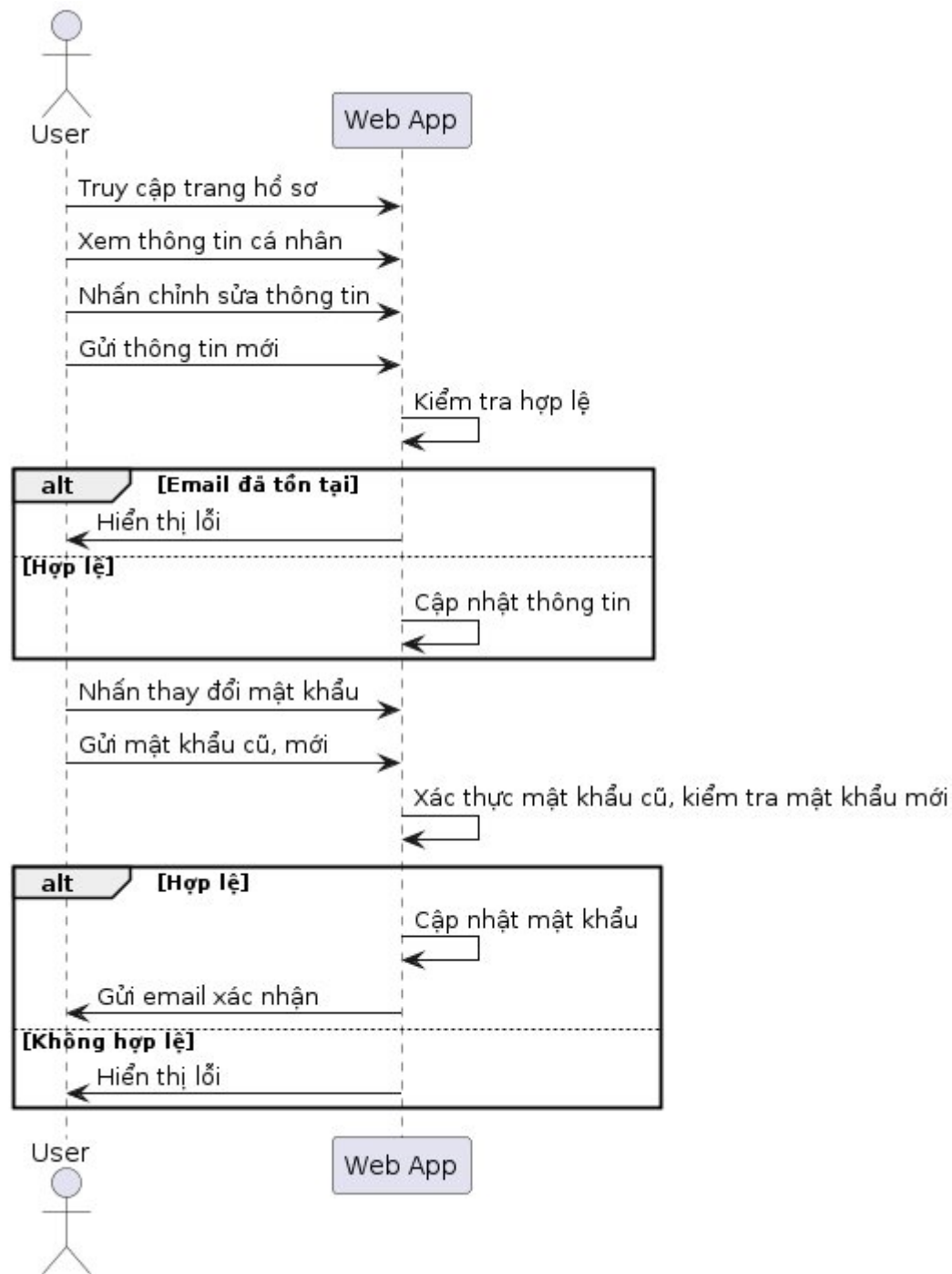
Usecase specification	
Use Case ID	UC04
Name	Quản lý hồ sơ cá nhân
Created By:	Trần Văn Hoàng
Primary Actor	Người dùng
Secondary Actor	Hệ thống, người quản trị hệ thống
Trigger	Người dùng truy cập trang hồ sơ
Description	Người dùng xem và chỉnh sửa thông tin cá nhân
Pre-condition	Đã đăng nhập
Post-condition	Thông tin cá nhân được cập nhật
Normal flow	<ol style="list-style-type: none"><li>1. Người dùng truy cập trang hồ sơ</li><li>2. Xem thông tin cá nhân</li><li>3. Nhấn nút chỉnh sửa thông tin</li><li>4. Chỉnh sửa và lưu thông tin</li><li>5. Nhấn nút thay đổi mật khẩu</li></ol>

	6. Nhập mật khẩu cũ và mới 7. Xác thực lại với email
Alternative flow	Không
Exception flow	Không
Business rules	Không

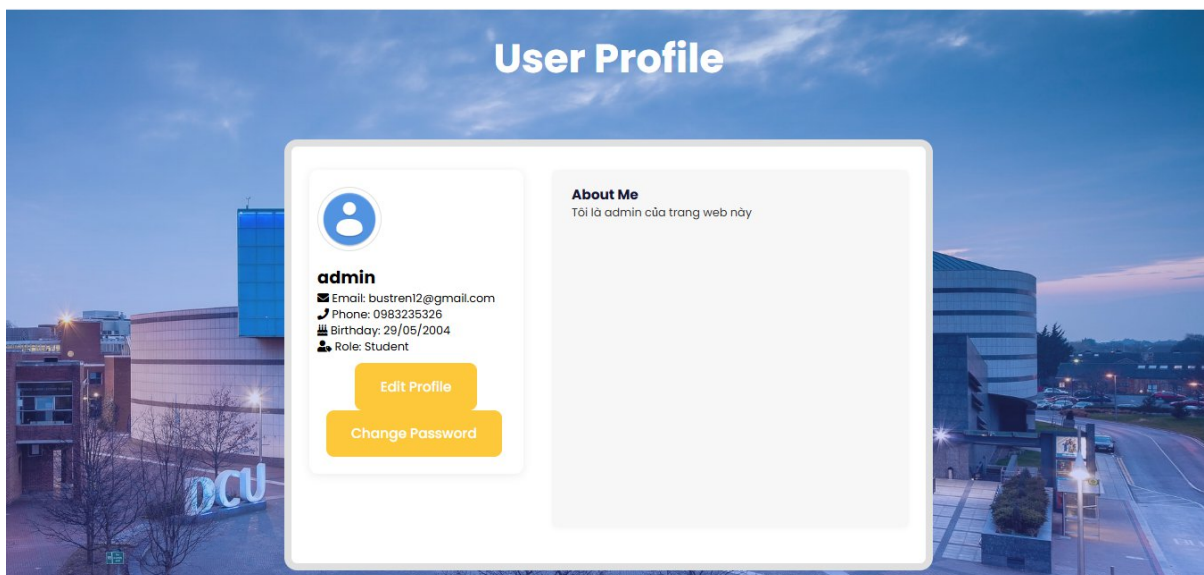
## Biểu đồ hoạt động



## Biểu đồ tuần tự



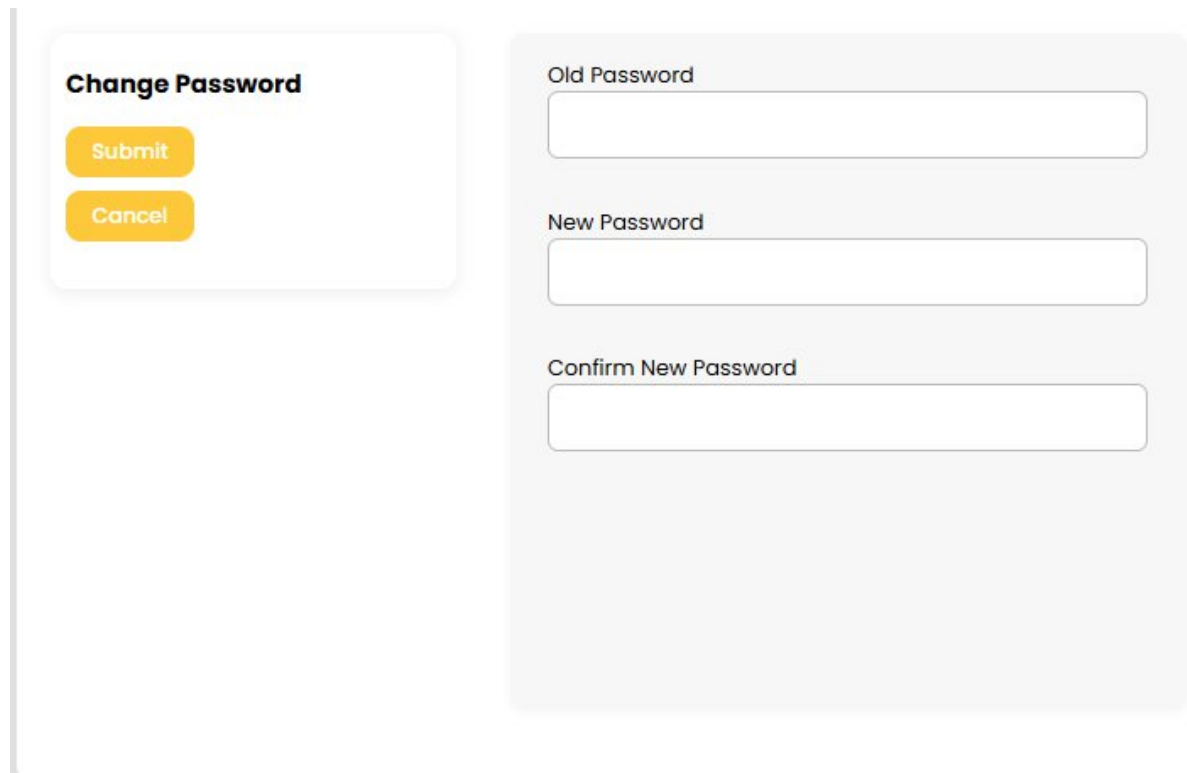
## Chi tiết giao diện



Trang profile hiển thị thông tin cá nhân, có nút chỉnh sửa. Form cập nhật thông tin, đổi mật khẩu, upload avatar. Hiển thị thông báo khi cập nhật thành công/thất bại.

The image shows the 'Edit Profile' form. On the left, there is a user card with a blue profile icon. The card contains the following fields: 'Phone' with the value '0983235326', 'Birthday' with the value '05/29/2004' and a calendar icon, and 'Role' with a dropdown menu showing 'Student'. To the right of the card is an 'Edit Description' section. This section has a rich text editor with a toolbar containing options for font, size, bold, italic, underline, text color, background color, bulleted list, numbered list, link, unlink, image, table, link icon, and source. The description text is 'Tôi là admin của trang web này'. At the bottom of the form are two yellow buttons: 'Update Profile' and 'Cancel'.

Form chỉnh sửa thông tin



The image shows a 'Change Password' form. On the left, there is a white box with the title 'Change Password' and two yellow buttons labeled 'Submit' and 'Cancel'. To the right of this box is a larger, light gray box containing three input fields. The first field is labeled 'Old Password', the second 'New Password', and the third 'Confirm New Password'. Each field is a simple white rectangle with a thin gray border.

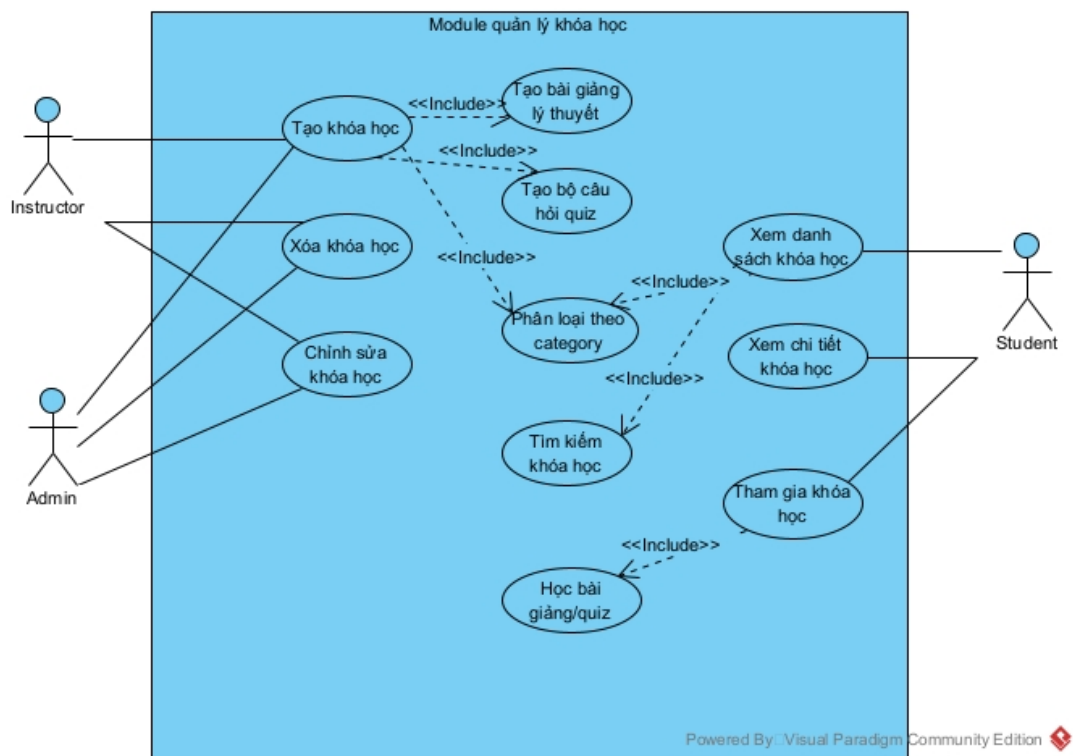
Form thay đổi mật khẩu

Frontend: profile.html

Backend: profile\_view

### 3.2.2. Module Quản lý khóa học

Sơ đồ Usecase tổng quan của module



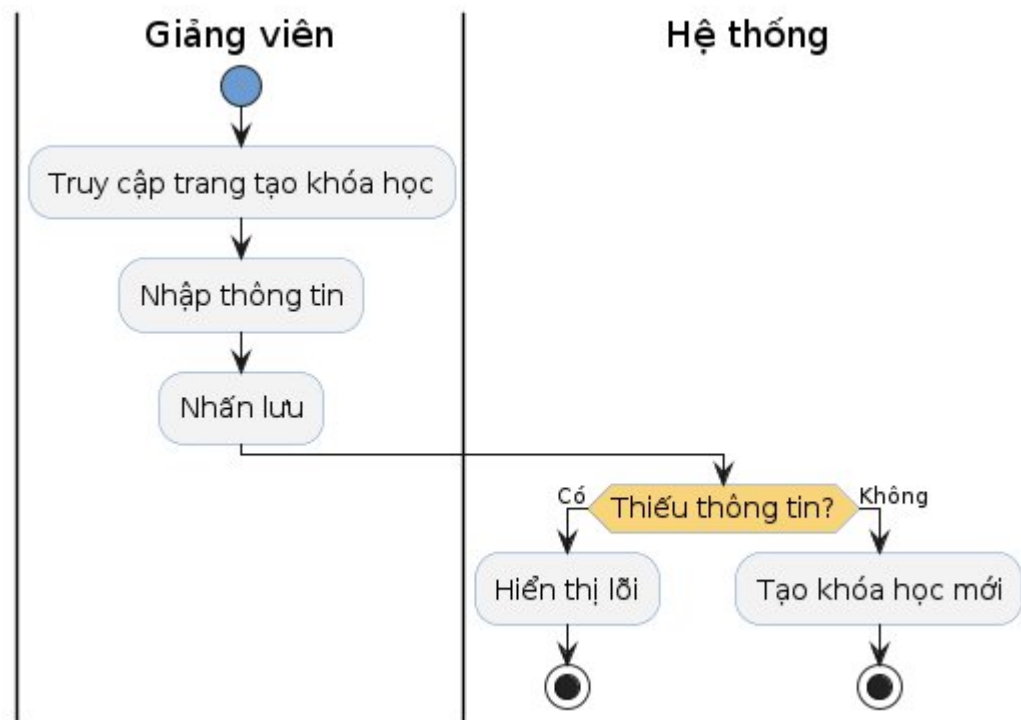
### 3.2.2.1. Usecase Tạo khóa học

Usecase specification	
Use Case ID	UC05
Name	Tạo khóa học
Created By:	Trần Văn Hoàng
Primary Actor	Người dùng có vai trò là giảng viên
Secondary Actor	Người quản trị hệ thống
Trigger	Giảng viên chọn chức năng tạo khóa học
Description	Giảng viên tạo mới một khóa học
Pre-condition	Đăng nhập với vai trò giảng viên hoặc admin
Post-condition	Khóa học mới được tạo
Normal flow	<ol style="list-style-type: none"> <li>1. Giảng viên truy cập trang tạo khóa học</li> <li>2. Nhập thông tin và lưu</li> <li>3. Hệ thống tạo khóa học mới</li> </ol>

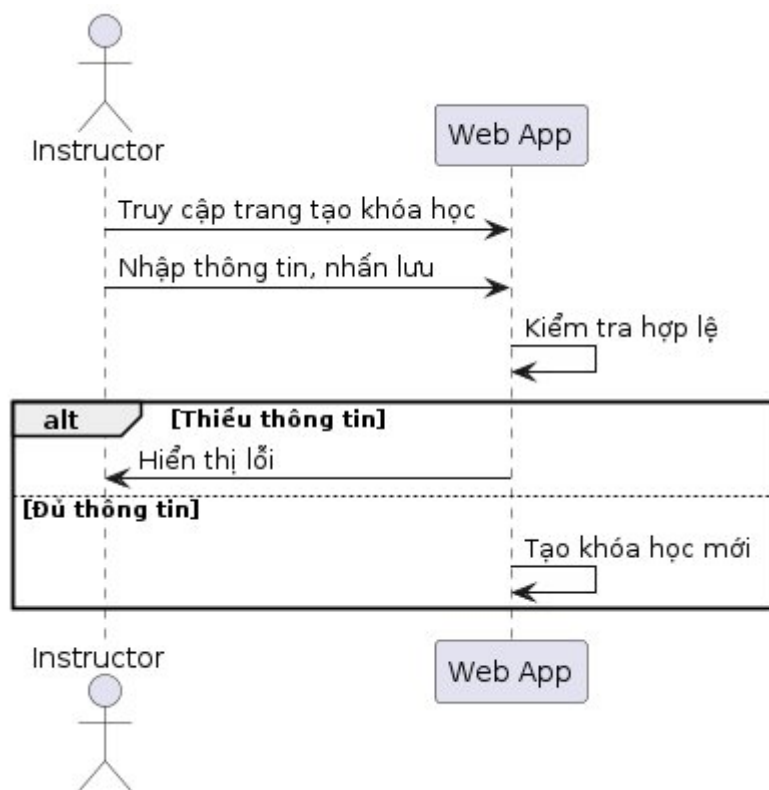


Alternative flow	- Nếu thiếu thông tin, hiển thị lỗi
Exception flow	- Lỗi lưu dữ liệu
Business rules	- Chỉ giảng viên hoặc admin được tạo khóa học

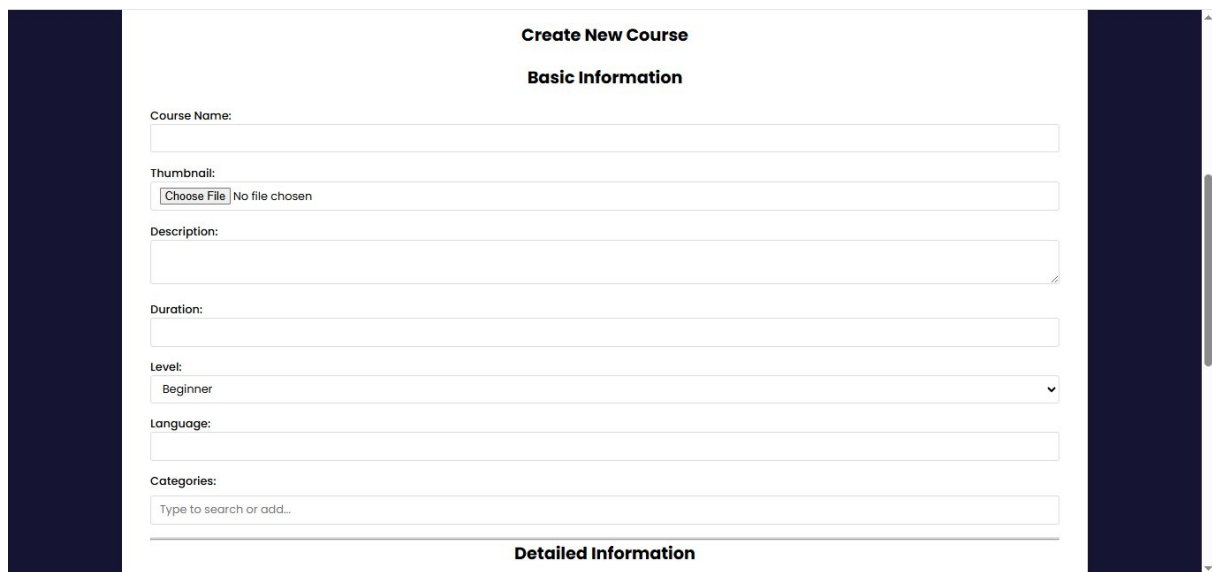
### Biểu đồ hoạt động



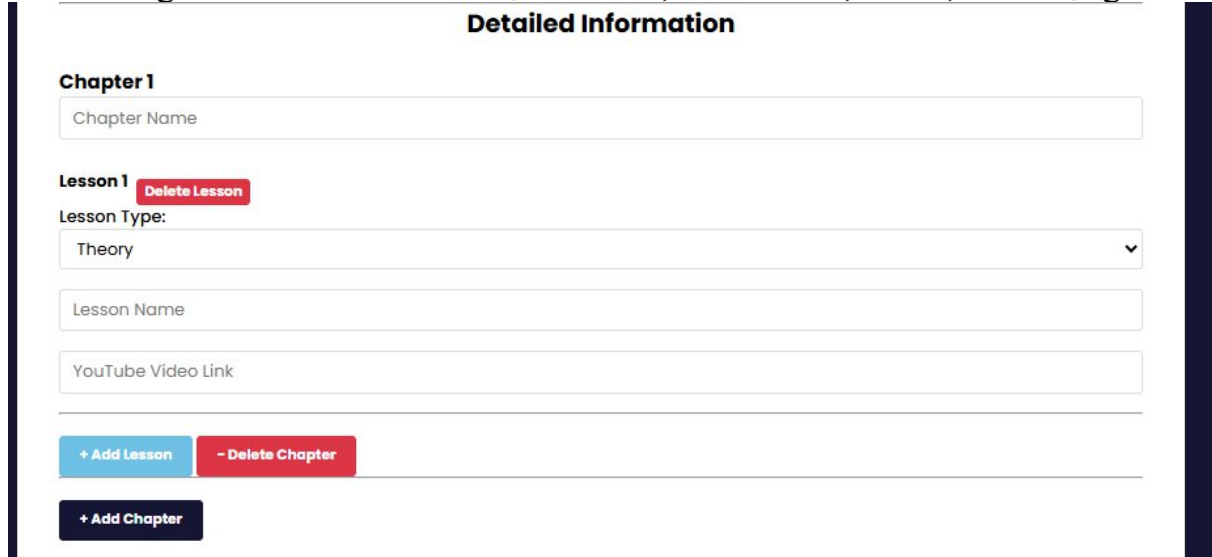
### Biểu đồ tuần tự



## Chi tiết giao diện



Điền thông tin cơ bản của khóa học như tên, ảnh tiêu đề, mô tả, thời lượng...



Tạo các chương và bài học cho khóa học, mỗi bài học có 2 dạng lý thuyết/ trắc nghiệm. Câu hỏi trắc nghiệm là các câu hỏi 4 đáp án.

Giao diện: create-course.html

Backend(views.py): create\_course\_view

### 3.2.2.2. Usecase Chỉnh sửa khóa học

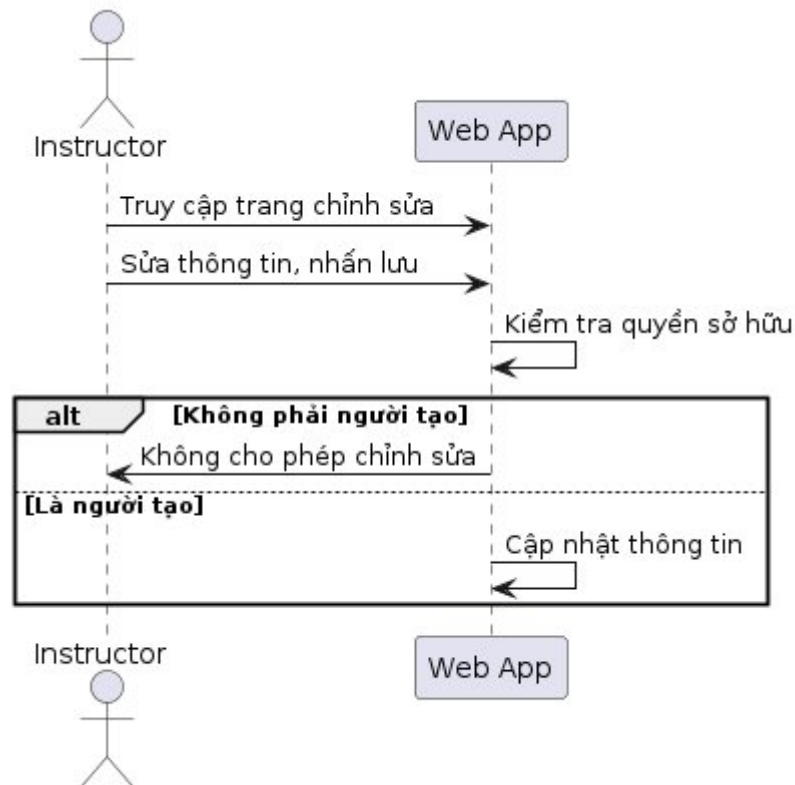
Usecase specification	
Use Case ID	UC06
Name	Chỉnh sửa khóa học

Created By:	Trần Văn Hoàng
Primary Actor	Người dùng có vai trò là giảng viên
Secondary Actor	Người quản trị hệ thống
Trigger	Giảng viên chọn chức năng chỉnh sửa khóa học
Description	Giảng viên chỉnh sửa khóa học của mình
Pre-condition	- Đăng nhập với vai trò giảng viên hoặc admin - Khóa học phải do giảng viên tạo ra
Post-condition	Chỉnh sửa khóa học đã có sẵn
Normal flow	1. Giảng viên truy cập trang khóa học cá nhân, vào mục khóa học đã tạo chọn chỉnh sửa 2. Sửa thông tin và lưu 3. Hệ thống cập nhật thông tin
Alternative flow	- Nếu không phải người tạo/ admin hệ thống, không cho phép chỉnh sửa
Exception flow	- Lỗi lưu dữ liệu
Business rules	- Chỉ người tạo và admin được chỉnh sửa

### Biểu đồ hoạt động



## Biểu đồ tuần tự



## Chi tiết giao diện

Edit Course

Course Name:

Project Management Essentials

Thumbnail:

Choose File

No file chosen

Description:

This is a detailed course covering the essentials of Project.

Price:

Duration:

4 weeks

Level:

Beginner

Language:

Vietnamese

Categories:

C++ Programming ✖

Type to search or add...

Trang cập nhật thông tin khóa học với dữ liệu cũ đã có sẵn. Có nút lưu, thông báo thành công/thất bại.

**Chapters & Lessons**

**Chapter 1**

**Lesson 1** Delete Lesson  
Lesson Type:

**Lesson 2** Delete Lesson  
Lesson Type:

**Lesson 3** Delete Lesson  
Lesson Type:

Chỉnh sửa phần khóa học

Giao diện: edit\_course.html

Backend(views.py): edit\_course\_view

### 3.2.2.3. Usecase Xóa khóa học

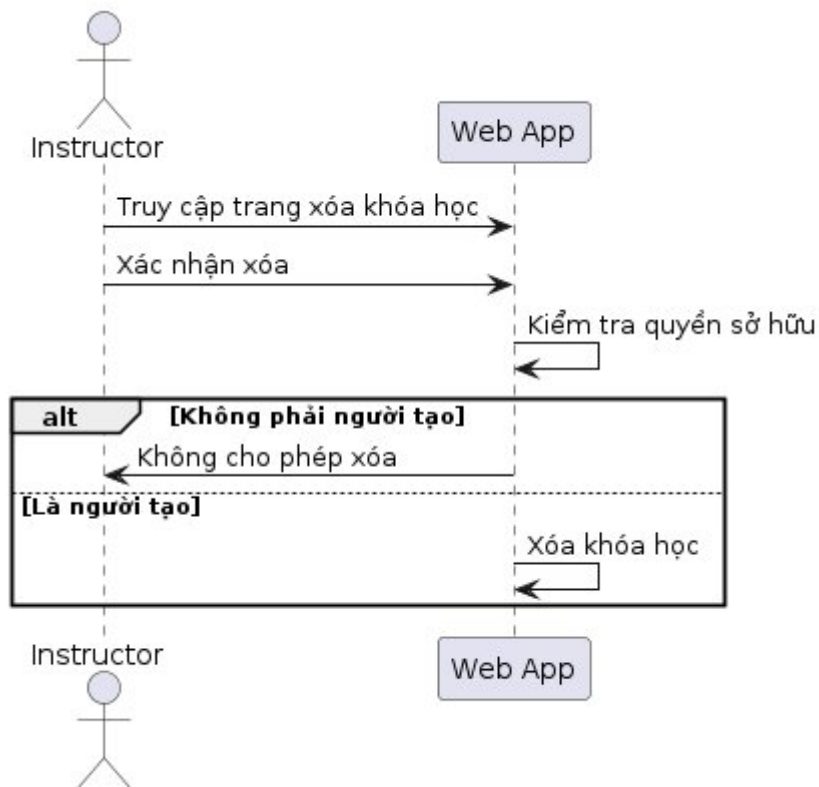
Usecase specification	
Use Case ID	UC07
Name	Xóa khóa học
Created By:	Trần Văn Hoàng
Primary Actor	Người dùng có vai trò là giảng viên
Secondary Actor	Người quản trị hệ thống
Trigger	Giảng viên chọn chức năng xóa khóa học
Description	Giảng viên xóa đi một khóa học
Pre-condition	Đăng nhập với vai trò giảng viên hoặc admin
Post-condition	Xóa khóa học mà giảng viên tạo hoặc bất kỳ khóa học nào với vai trò là admin
Normal flow	1. Giảng viên truy cập trang khóa học cá nhân phần khóa học đã tạo 2. Nhấn nút xóa khóa học 3. Hệ thống xóa khóa học

Alternative flow	Nếu không phải người tạo, không cho phép xóa
Exception flow	Lỗi xóa dữ liệu
Business rules	Chỉ người tạo hoặc admin được xóa

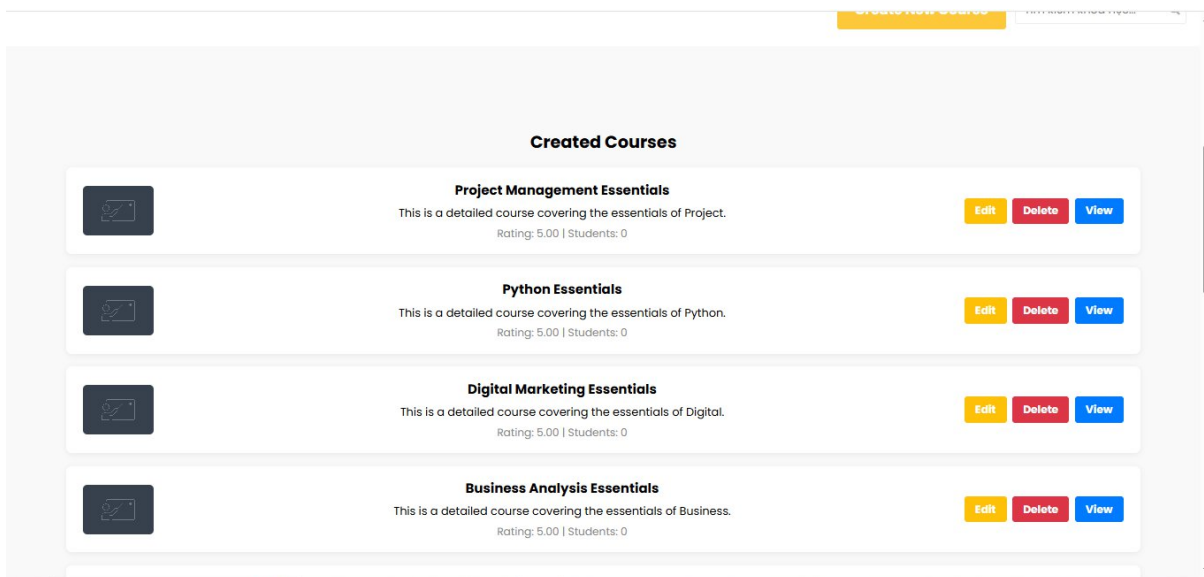
### Biểu đồ hoạt động



### Biểu đồ tuần tự

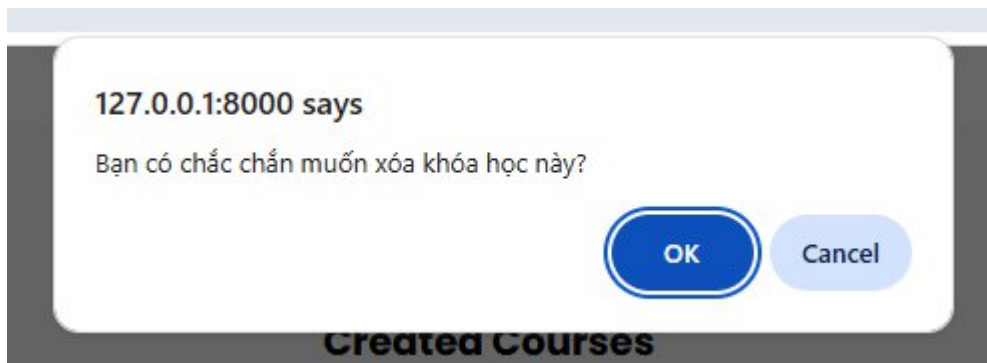


## Chi tiết giao diện



Những khóa học đã tạo của người dùng đều có nút delete để xóa khóa học





Thông báo xác nhận xóa khóa học

Frontend: my\_courses.html

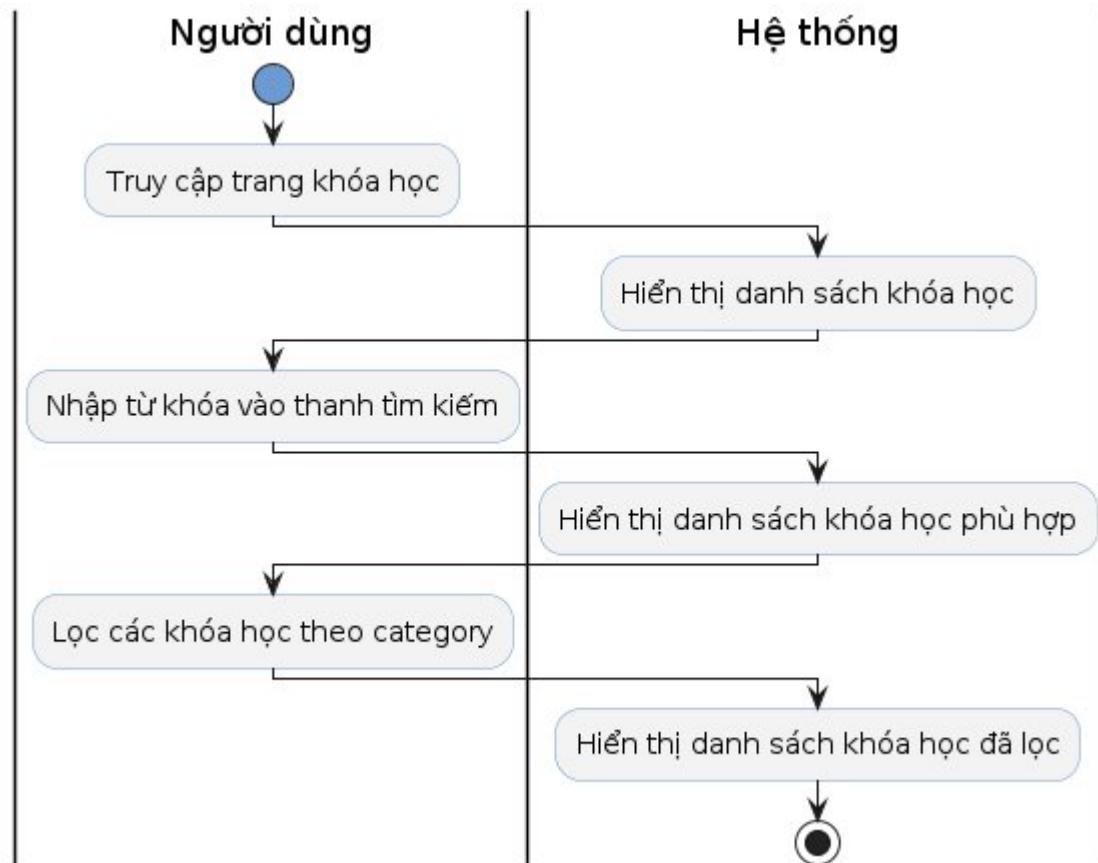
Backend: delete\_course\_view

#### 3.2.2.4. Usecase Xem danh sách khóa học

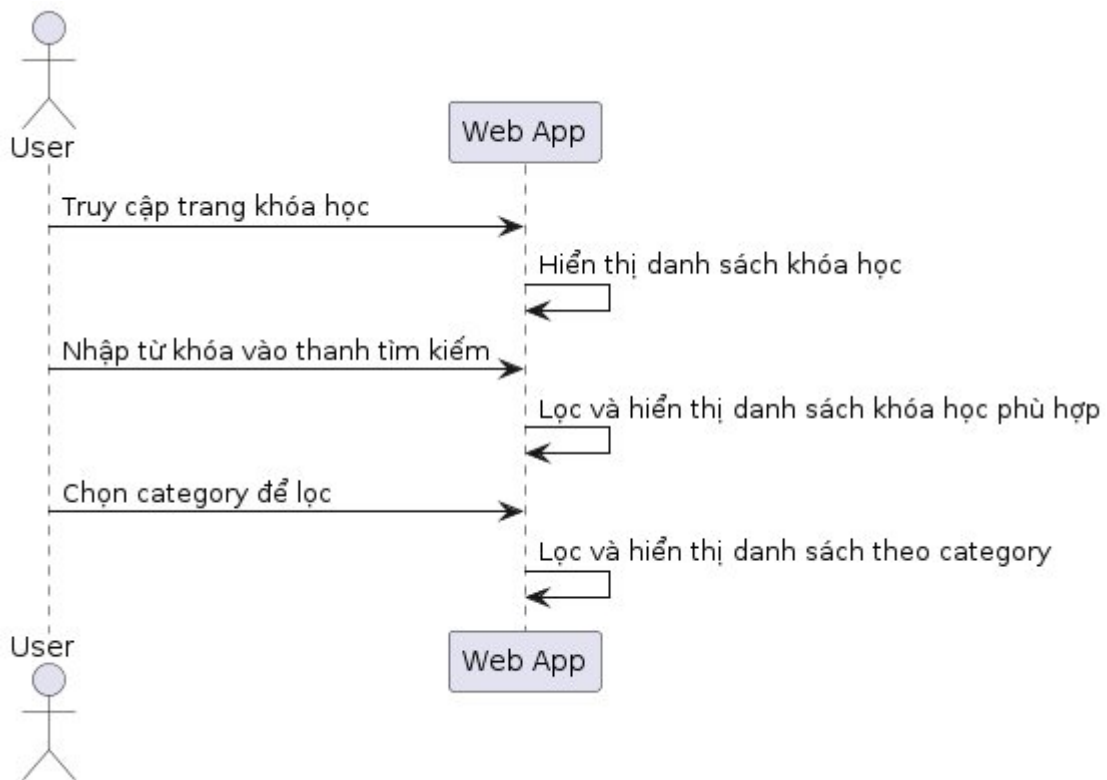
Usecase specification	
Use Case ID	UC08
Name	Xem danh sách khóa học
Created By:	Trần Văn Hoàng
Primary Actor	Người dùng
Secondary Actor	Người quản trị hệ thống
Trigger	Người dùng truy cập trang khóa học
Description	Hiển thị danh sách các khóa học, có thể chia theo category và tìm kiếm
Pre-condition	Người dùng đã đăng nhập
Post-condition	Danh sách khóa học được hiển thị
Normal flow	<ol style="list-style-type: none"> <li>1. Người dùng truy cập trang khóa học</li> <li>2. Hệ thống hiển thị danh sách khóa học</li> <li>3. Người dùng tìm kiếm theo tên khóa học bằng thanh tìm kiếm.</li> <li>4. Người dùng lọc các khóa học theo category phù hợp</li> </ol>
Alternative flow	Không

Exception flow	Không
Business rules	Không

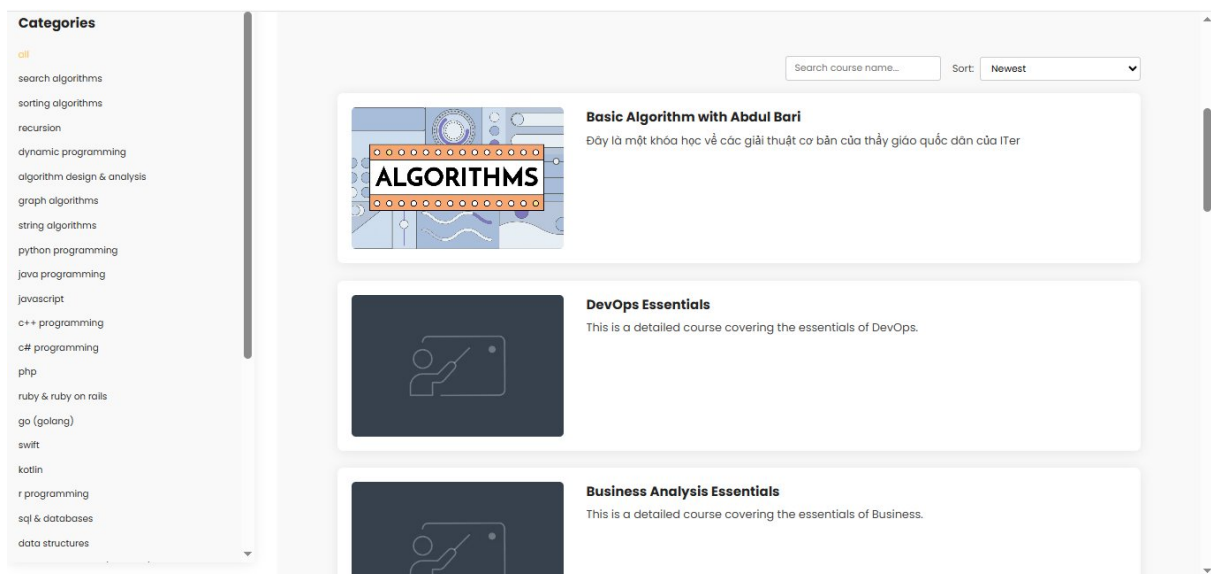
### Biểu đồ hoạt động



### Biểu đồ tuần tự



## Chi tiết giao diện



Danh sách khóa học bao gồm thanh tìm kiếm, thanh sắp xếp.

Frontend: `courses.html`

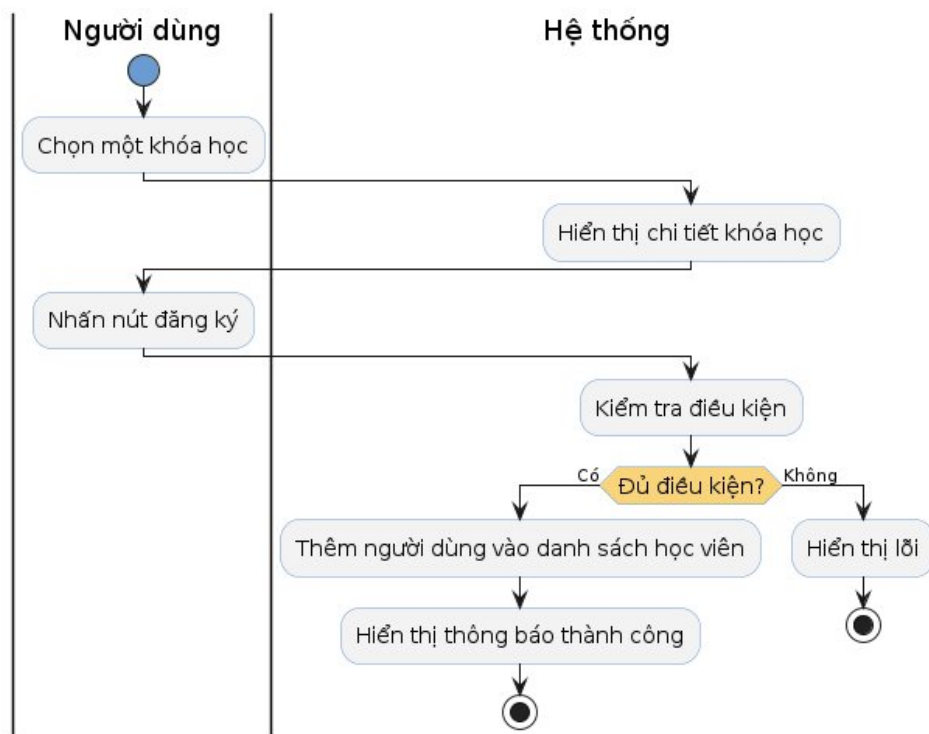
Backend: `course_view`

### 3.2.2.5. Usecase Xem chi tiết khóa học và đăng ký khóa học

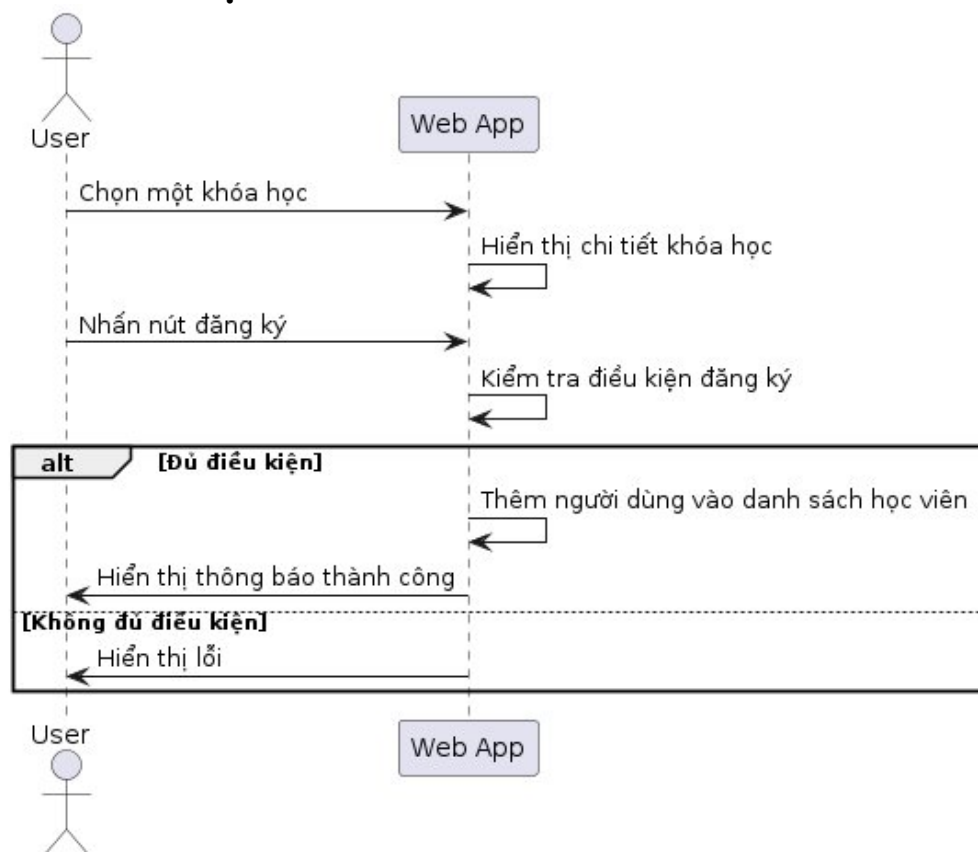
Usecase specification

Use Case ID	UC09
Name	Xem chi tiết khóa học và đăng ký khóa học
Created By:	Trần Văn Hoàng
Primary Actor	Người dùng đã đăng nhập vào hệ thống
Secondary Actor	Người quản trị hệ thống
Trigger	Người dùng xem và nhấn nút đăng ký khóa học
Description	Người dùng xem chi tiết đăng ký tham gia một khóa học
Pre-condition	Đã đăng nhập, chưa đăng ký khóa học đó
Post-condition	Người dùng được thêm vào danh sách học viên
Normal flow	<ol style="list-style-type: none"> <li>1. Người dùng chọn một khóa học</li> <li>2. Hệ thống hiển thị chi tiết khóa học</li> <li>3. Người dùng nhấn đăng ký</li> <li>4. Hệ thống kiểm tra điều kiện</li> <li>5. Thêm người dùng vào danh sách học viên</li> </ol>
Alternative flow	- Nếu thiếu thông tin, hiển thị lỗi
Exception flow	- Lỗi lưu dữ liệu
Business rules	- Chỉ giảng viên hoặc admin được tạo khóa học

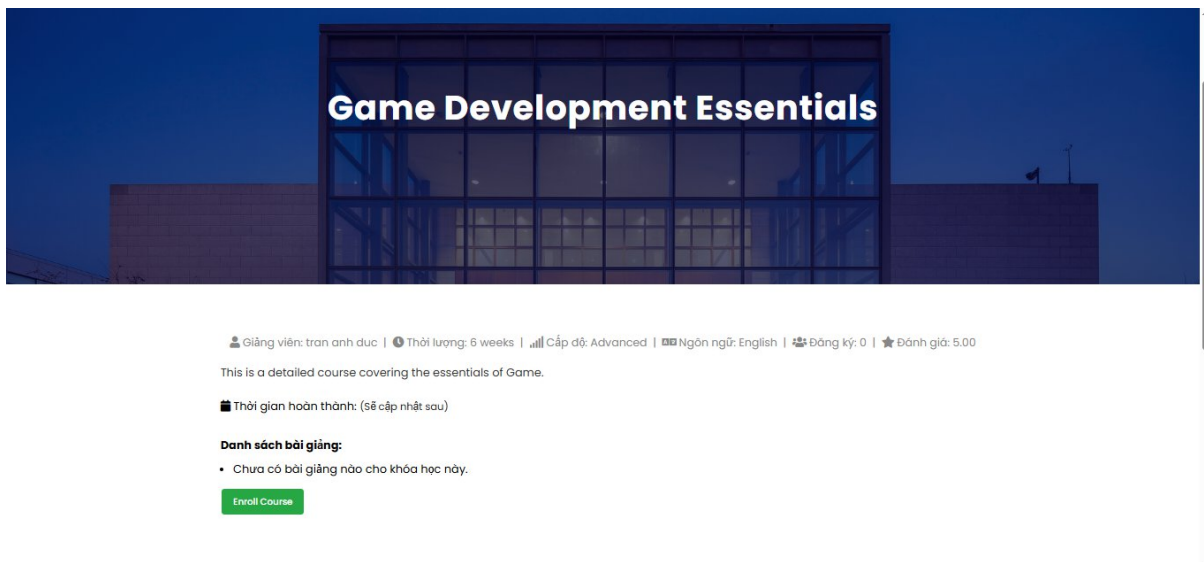
### **Biểu đồ hoạt động**



### Biểu đồ tuần tự



### Chi tiết giao diện



Nút "Đăng ký" trên trang chi tiết khóa học.

Trang giao diện: course-inner.html

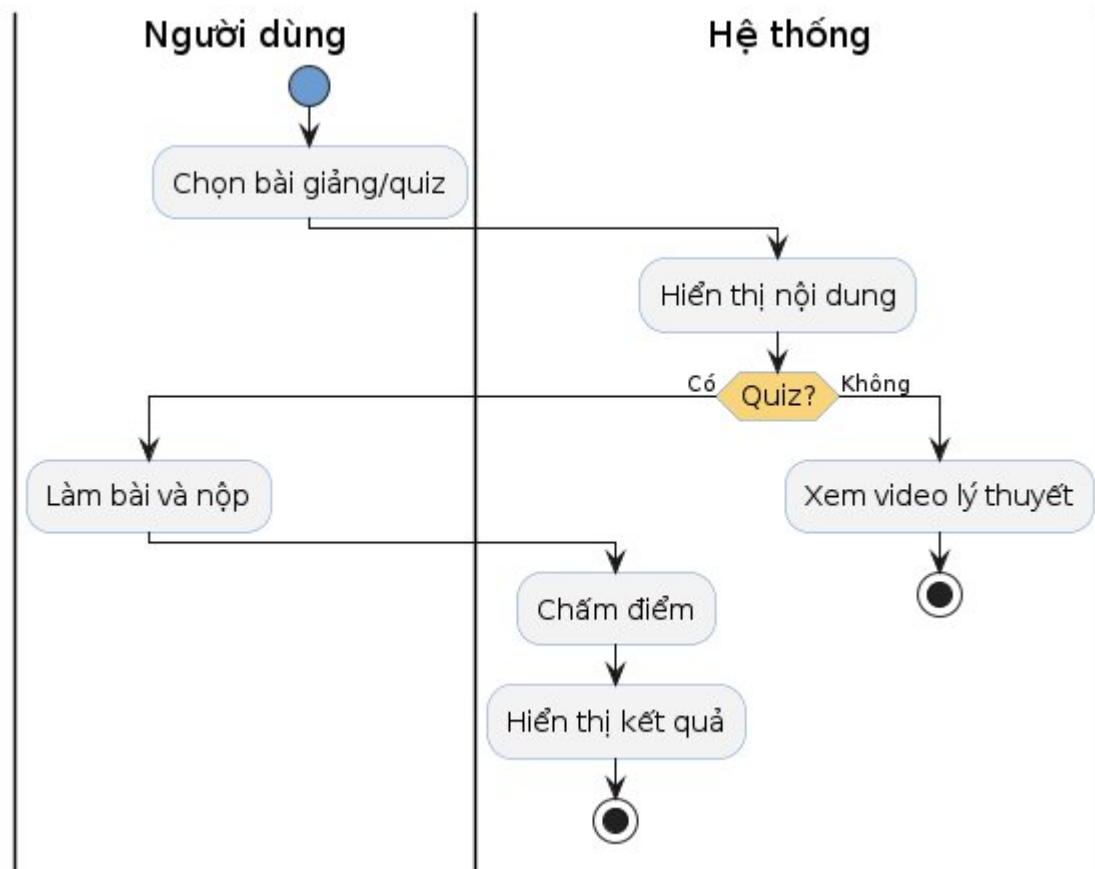
Backend: `def course_inner_view(request, courseID):`

### 3.2.2.5. Usecase Học bài giảng/quiz

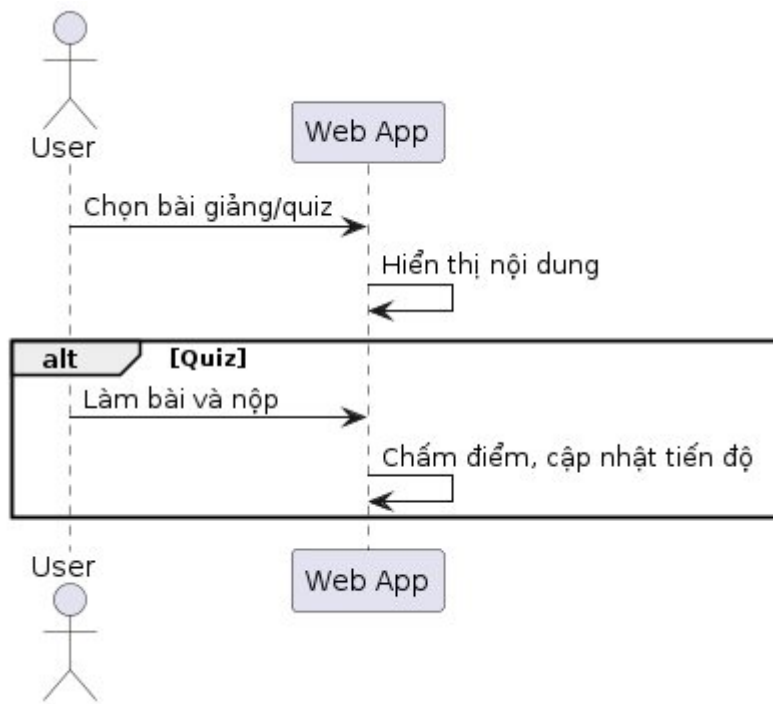
Usecase specification	
Use Case ID	UC11
Name	Học khóa học
Created By:	Trần Văn Hoàng
Primary Actor	Người dùng
Secondary Actor	Người quản trị hệ thống
Trigger	Người dùng chọn học khóa học rồi chọn bài giảng hoặc quiz
Description	Người dùng học bài giảng hoặc làm quiz trong khóa học
Pre-condition	Đã đăng ký khóa học
Post-condition	Cập nhật tiến độ học
Normal flow	<ol style="list-style-type: none"> <li>1. Người dùng chọn bài giảng/quiz</li> <li>2. Hệ thống hiển thị nội dung</li> <li>3. Nếu là quiz, người dùng làm bài và nộp</li> </ol>

	4. Hệ thống chấm điểm và cập nhật tiến độ
Alternative flow	Nếu chưa đăng ký khóa học, chuyển hướng đăng ký
Exception flow	Lỗi tải nội dung
Business rules	Chỉ học viên đã đăng ký mới truy cập được

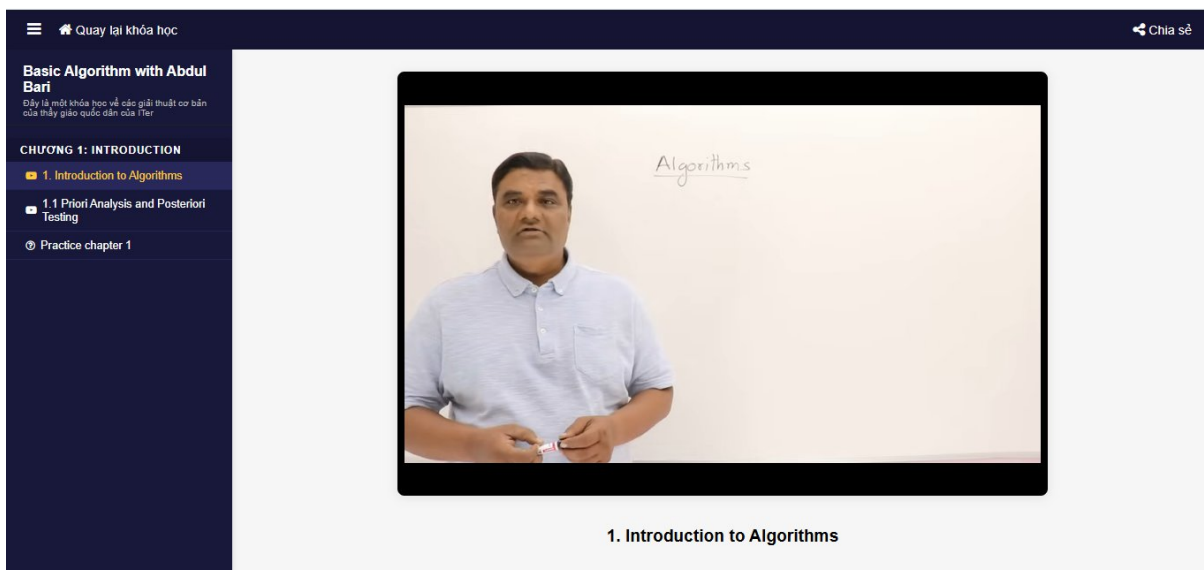
### Biểu đồ hoạt động



### Biểu đồ tuần tự

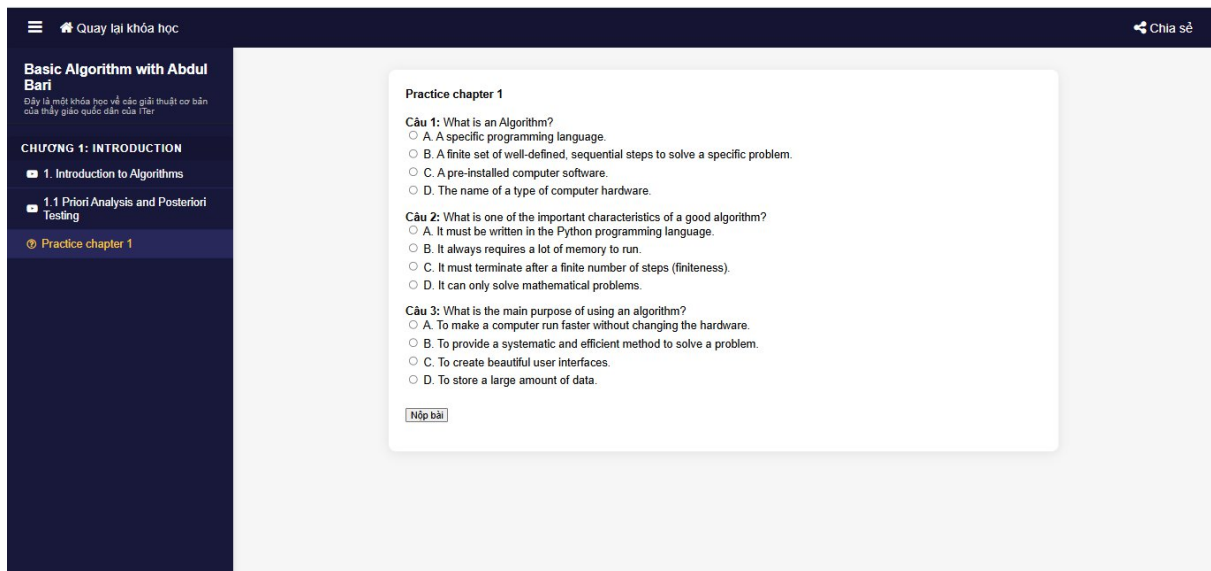


## Chi tiết giao diện



## Học bài giảng bằng video





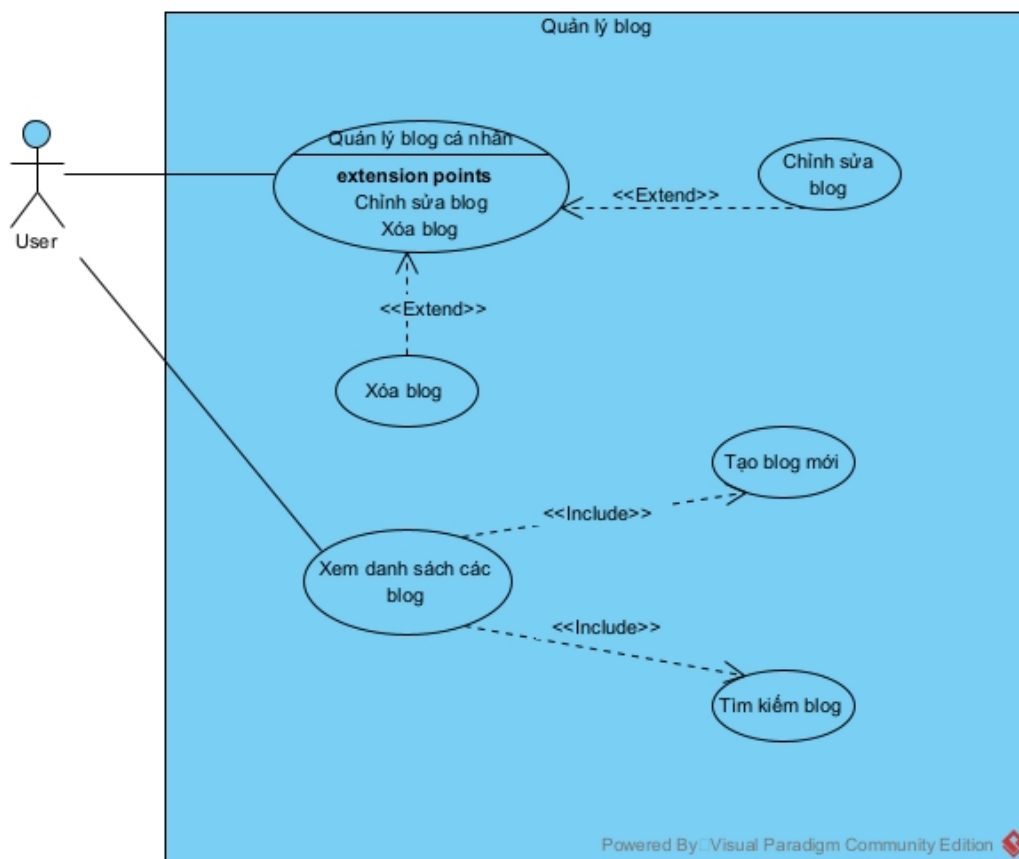
Học bài bằng câu hỏi trắc nghiệm

Frontend: study\_course.html

Backend: lesson\_view

### 3.2.3. Module Quản lý blog

Sơ đồ Usecase của module

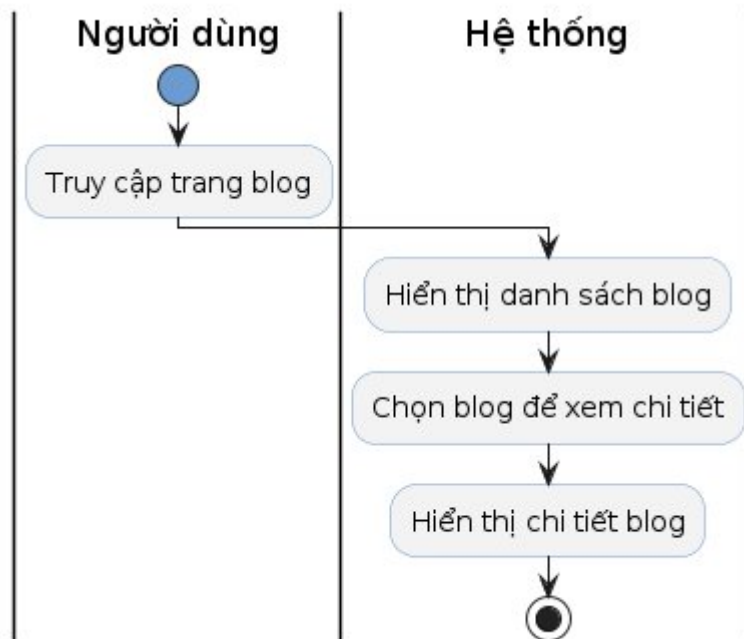


## Đặc tả Usecase cho module

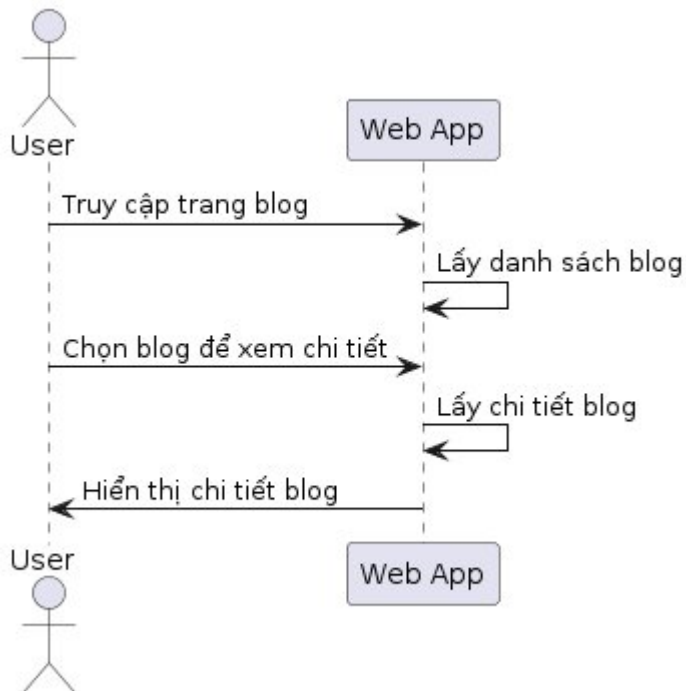
### 3.2.3.1. Usecase xem blog

Usecase specification	
Use Case ID	UC12
Name	Xem blog
Created By:	Trần Văn Hoàng
Primary Actor	Người dùng
Secondary Actor	Người quản trị hệ thống
Trigger	Người dùng truy cập vào trang blog
Description	Hiển thị danh sách và chi tiết các bài blog
Pre-condition	Đã đăng nhập
Post-condition	Xem được blog
Normal flow	1. Người dùng truy cập trang blog 2. Hệ thống hiển thị danh sách blog 3. Người dùng chọn blog để xem chi tiết
Alternative flow	Không có
Exception flow	Không có
Business rules	Không có

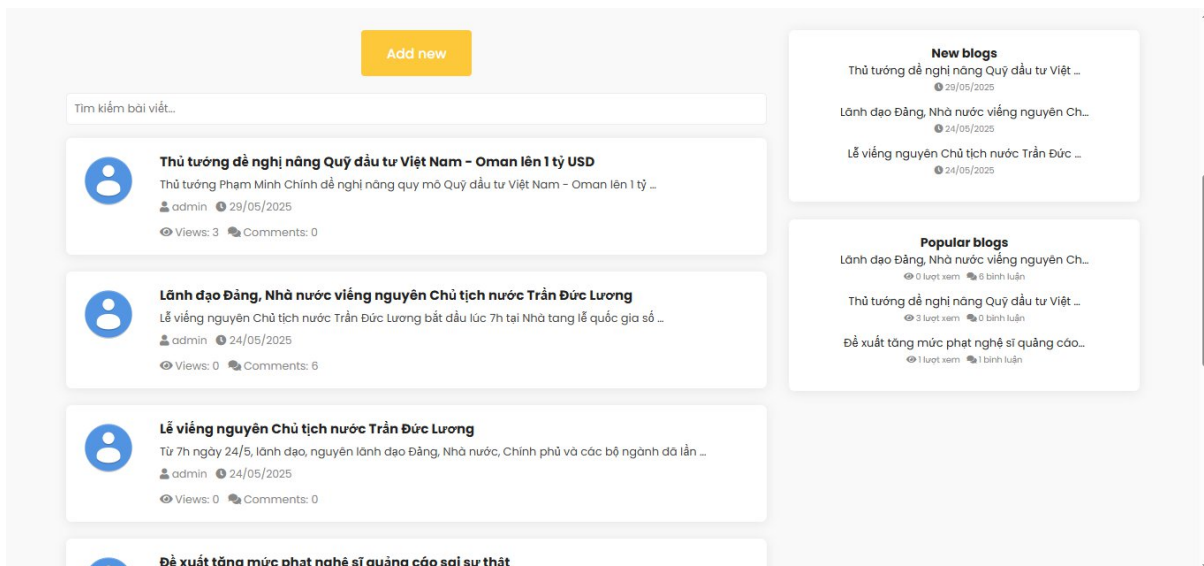
### Biểu đồ hoạt động



### Biểu đồ tuần tự



### Chi tiết giao diện



Trang danh sách blog với chức năng tìm kiếm, lọc theo từ khóa, tác giả, ngày đăng. Hiện thị trích đoạn nội dung, số lượt xem.



Trang chi tiết của các blog.

Frontend: danh sách blog: blog.html, chi tiết blog: post.html

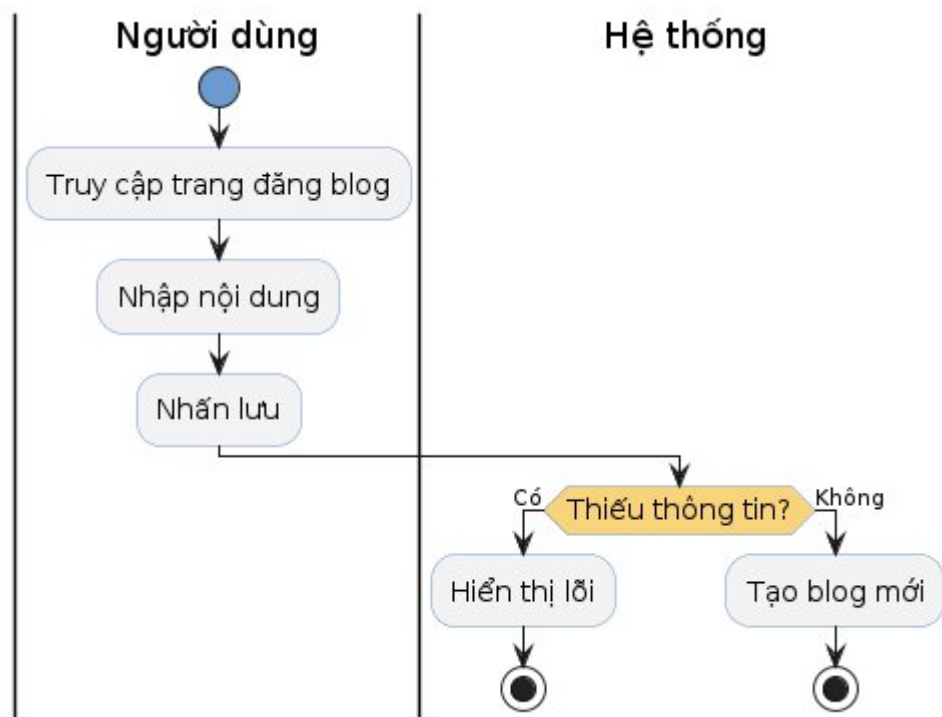
Backend: blog\_view ,blog\_detail\_view

### 3.2.3.2. Usecase Đăng bài blog

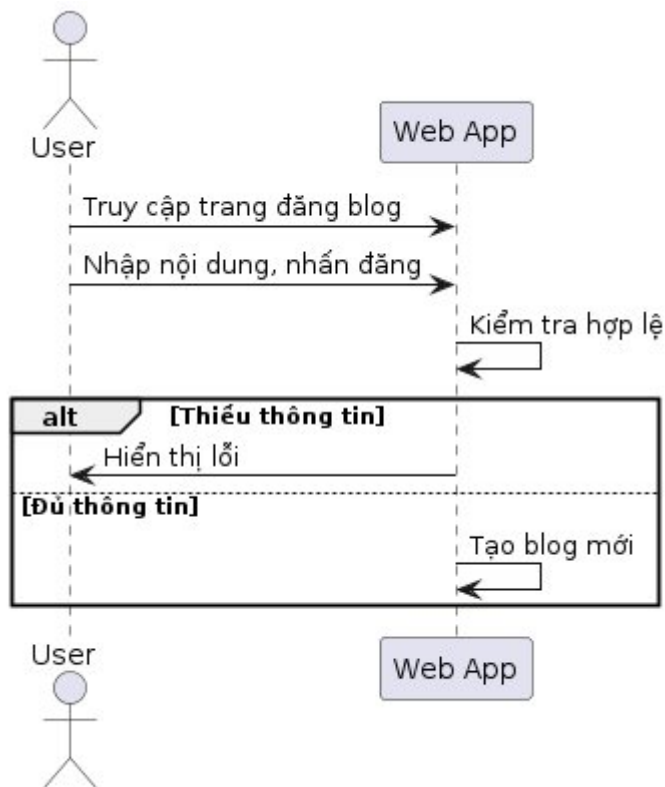
Usecase specification	
Use Case ID	UC13
Name	Đăng bài blog
Created By:	Trần Văn Hoàng

Primary Actor	Người dùng
Secondary Actor	Người quản trị hệ thống
Trigger	Người dùng chọn chức năng đăng bài blog
Description	Người dùng đăng bài blog mới
Pre-condition	Đã đăng nhập
Post-condition	Blog mới được tạo
Normal flow	1. Người dùng truy cập trang đăng blog 2. Nhập nội dung và lưu 3. Hệ thống tạo blog mới
Alternative flow	Nếu thiếu thông tin, hiển thị lỗi
Exception flow	Lỗi lưu dữ liệu
Business rules	Chỉ người dùng đã đăng nhập mới đăng blog

### Biểu đồ hoạt động



### Biểu đồ tuần tự



## Chi tiết giao diện

The screenshot shows a web form titled 'Thêm bài viết mới'. It includes a 'Title:' label followed by a text input field. Below this is a 'Content:' label followed by a rich text editor with a toolbar containing options for font, size, bold, italic, underline, link, unlink, list, and image. At the bottom of the form is a large yellow button labeled 'Đăng bài' (Post).

Trang form nhập tiêu đề, nội dung, upload ảnh. Có nút đăng, thông báo thành công/thất bại.

Trang giao diện: new-blog.html

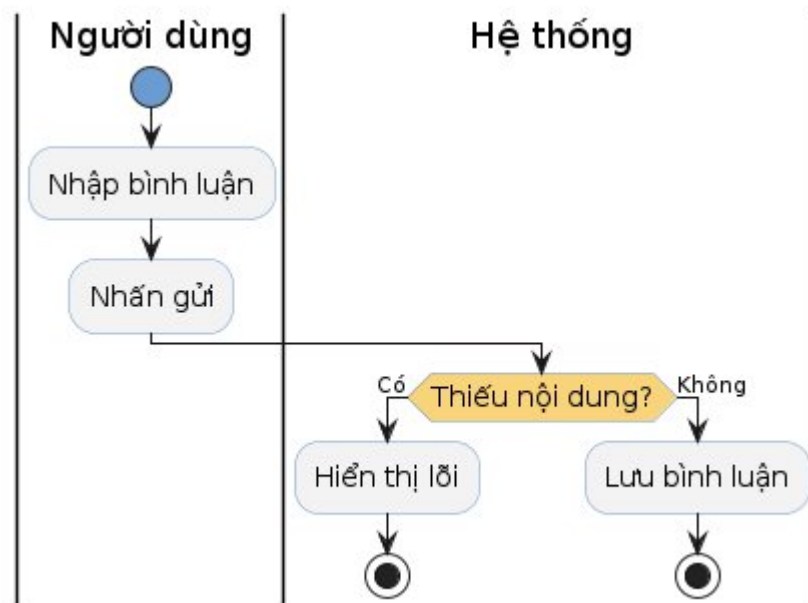
Backend: new\_blog\_view

### 3.2.3.3. Bình luận blog

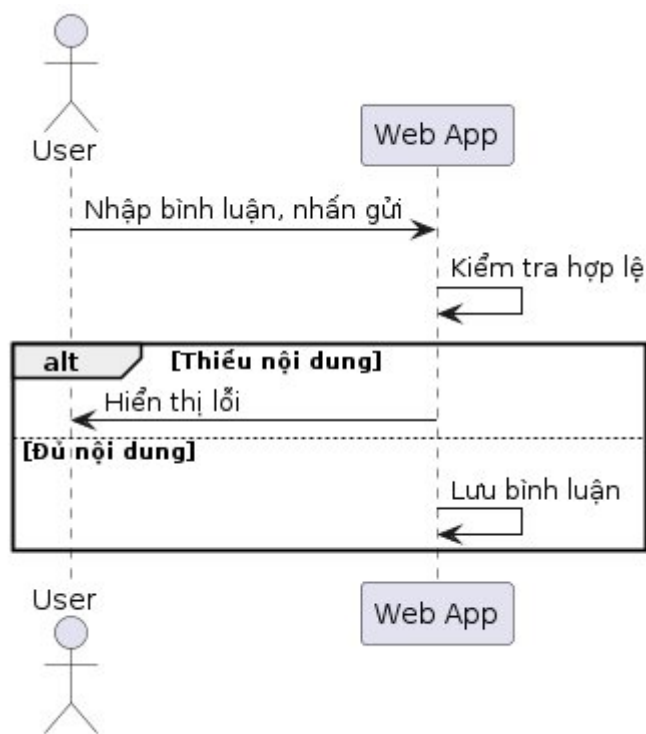
Usecase specification	
Use Case ID	UC14

Name	Bình luận blog
Created By:	Trần Văn Hoàng
Primary Actor	Người dùng
Secondary Actor	Người quản trị hệ thống
Trigger	Người dùng nhập bình luận vào blog
Description	Người dùng bình luận vào bài blog
Pre-condition	Đã đăng nhập, người dùng đang xem blog
Post-condition	Bình luận được thêm vào blog
Normal flow	1. Người dùng nhập bình luận 2. Hệ thống lưu bình luận
Alternative flow	Nếu thiếu nội dung, hiển thị lỗi
Exception flow	Lỗi lưu dữ liệu
Business rules	Chỉ người dùng đã đăng nhập mới có thể bình luận

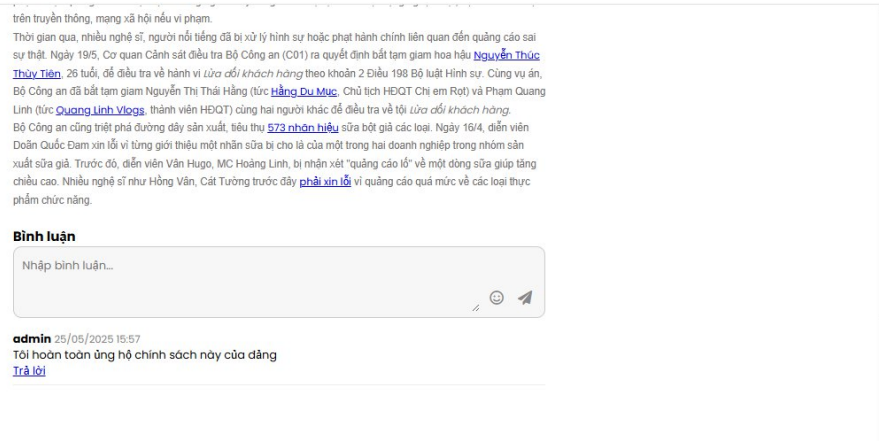
### Biểu đồ hoạt động



### Biểu đồ tuần tự



## Chi tiết giao diện



Form nhập bình luận dưới mỗi bài blog, nút gửi, hiển thị danh sách bình luận, hỗ trợ trả lời bình luận.

Frontend: post.html

Backend: blog\_comment\_view

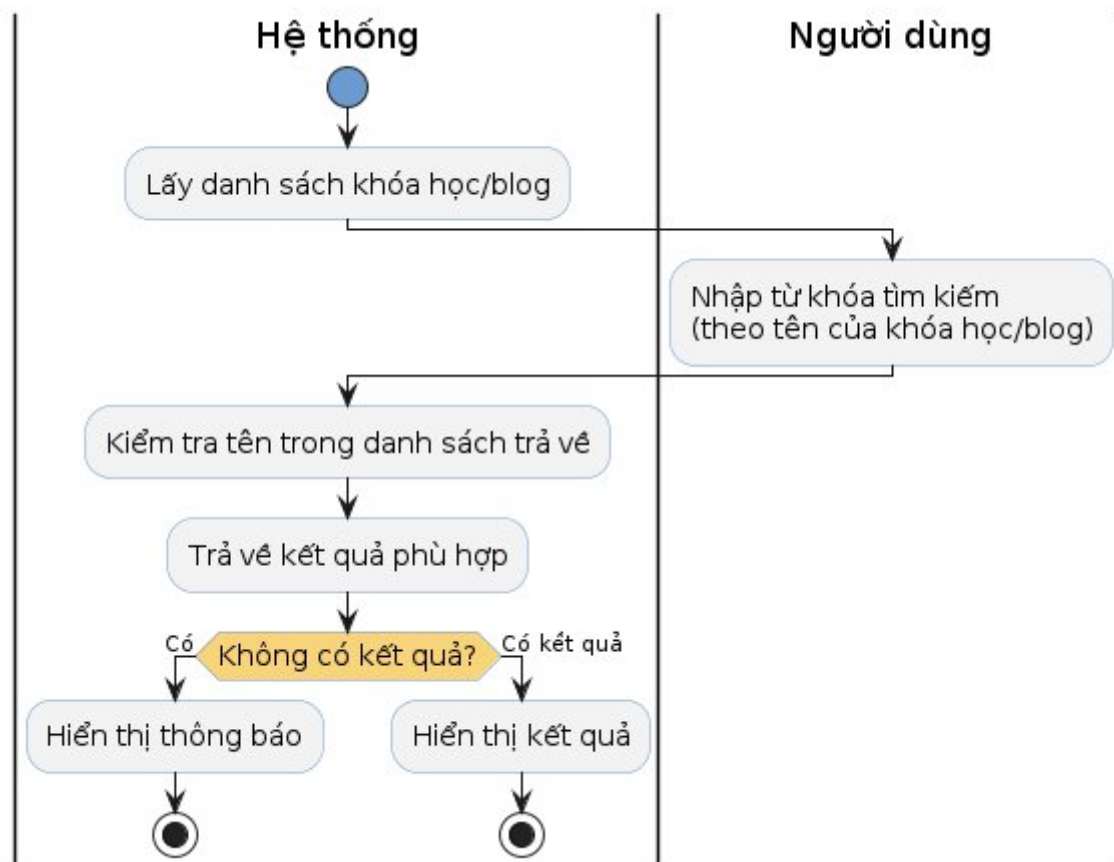
### 3.2.3.4. Tìm kiếm blog

Usecase specification	
Use Case ID	UC15
Name	Tìm kiếm blog
Created By:	Trần Văn Hoàng

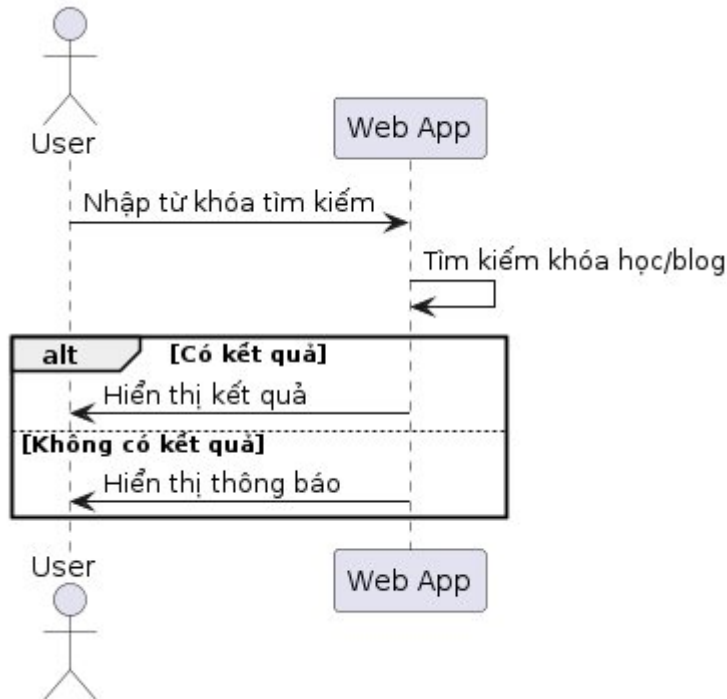


Primary Actor	Người dùng
Secondary Actor	Người quản trị hệ thống
Trigger	Người dùng nhập từ khóa tìm kiếm
Description	Tìm kiếm khóa học hoặc blog theo từ khóa
Pre-condition	Đã đăng nhập
Post-condition	Hiển thị kết quả tìm kiếm
Normal flow	1. Người dùng nhập từ khóa 2. Hệ thống trả về kết quả phù hợp
Alternative flow	Nếu không có kết quả, hiển thị thông báo
Exception flow	Lỗi truy vấn dữ liệu
Business rules	Không có

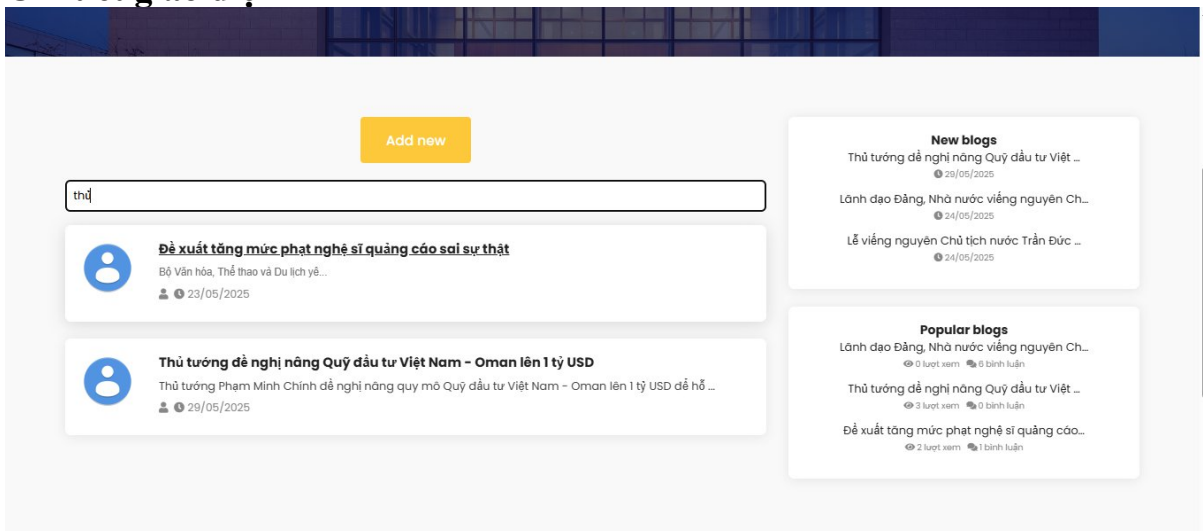
### Biểu đồ hoạt động



## Biểu đồ tuần tự



## Chi tiết giao diện



Ô tìm kiếm trên trang blog, lọc kết quả theo từ khóa. Khi nhập từ khóa, giao diện sẽ tự động lọc theo từ khóa tìm kiếm bằng kỹ thuật ajax.

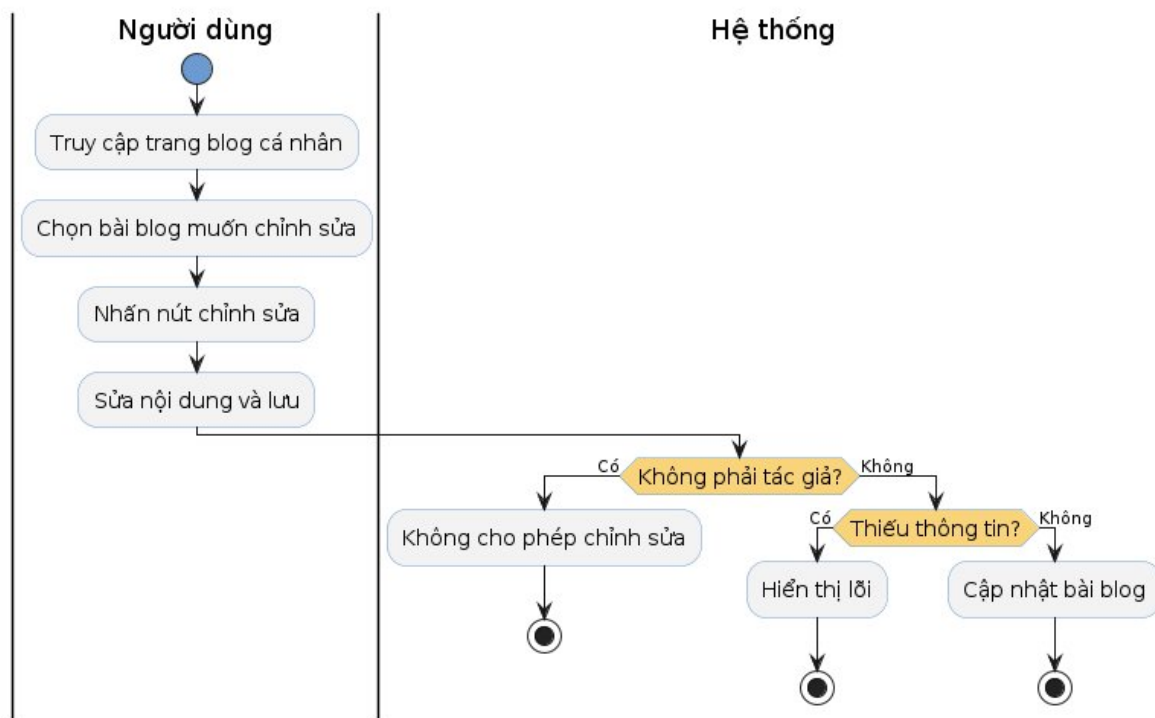
Frontend: blog.html

Backend: blog\_search\_ajax

### 3.2.3.5. Chỉnh sửa blog

Usecase specification	
Use Case ID	UC16
Name	Chỉnh sửa blog
Created By:	Trần Văn Hoàng
Primary Actor	Người dùng
Secondary Actor	Người quản trị hệ thống
Trigger	Người dùng chọn chức năng chỉnh sửa blog
Description	Chỉnh sửa bài blog của cá nhân người dùng
Pre-condition	Đã đăng nhập, là tác giả của blog
Post-condition	Bài blog được cập nhật
Normal flow	<ol style="list-style-type: none"><li>1. Người dùng truy cập trang blog cá nhân</li><li>2. Chọn bài blog muốn chỉnh sửa</li><li>3. Nhấn nút chỉnh sửa</li><li>4. Sửa nội dung và lưu</li><li>5. Hệ thống cập nhật bài blog</li></ol>
Alternative flow	<ul style="list-style-type: none"><li>- Nếu không phải tác giả, không cho phép chỉnh sửa</li><li>- Nếu thiếu thông tin, hiển thị lỗi</li></ul>
Exception flow	Lỗi lưu dữ liệu
Business rules	Chỉ tác giả mới được chỉnh sửa blog

### Biểu đồ hoạt động



**Biểu đồ tuần tự**



Frontend: edit-blog.html

Backend: edit\_blog

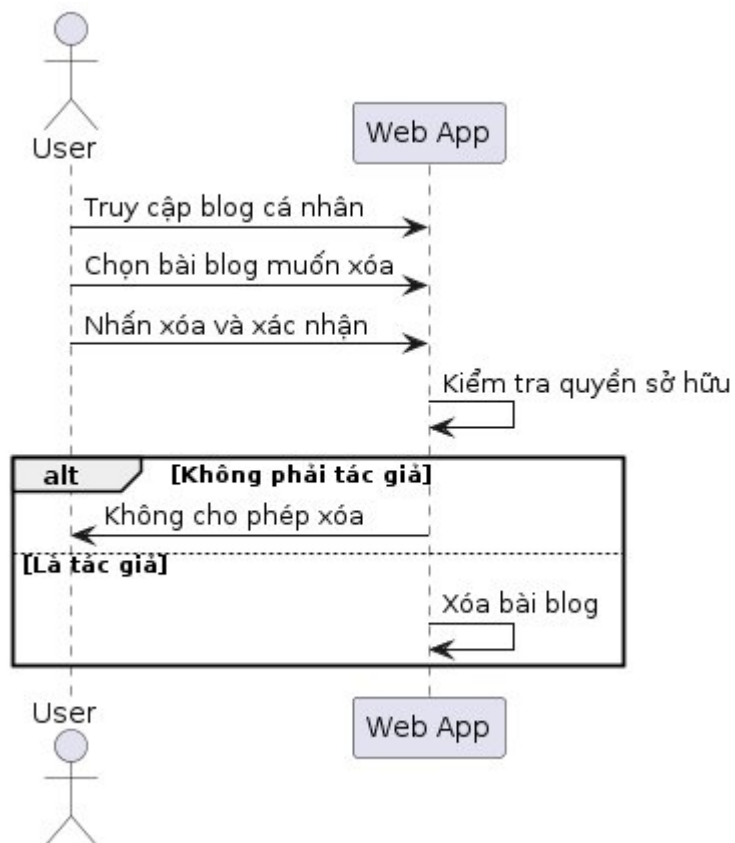
### 3.2.3.6. Xóa blog

Usecase specification	
Use Case ID	UC15
Name	Xóa blog
Created By:	Trần Văn Hoàng
Primary Actor	Người dùng
Secondary Actor	Người quản trị hệ thống
Trigger	Người dùng chọn chức năng xóa blog
Description	Tìm kiếm khóa học hoặc blog theo từ khóa
Pre-condition	Đã đăng nhập, là tác giả của bài blog
Post-condition	Bài blog bị xóa khỏi hệ thống
Normal flow	1. Người dùng truy cập trang blog cá nhân 2. Chọn bài blog muốn xóa 3. Nhấn nút xóa và xác nhận 4. Hệ thống xóa bài blog
Alternative flow	Nếu không phải tác giả, không cho phép xóa
Exception flow	Lỗi xóa dữ liệu
Business rules	Chỉ tác giả/admin mới được xóa blog

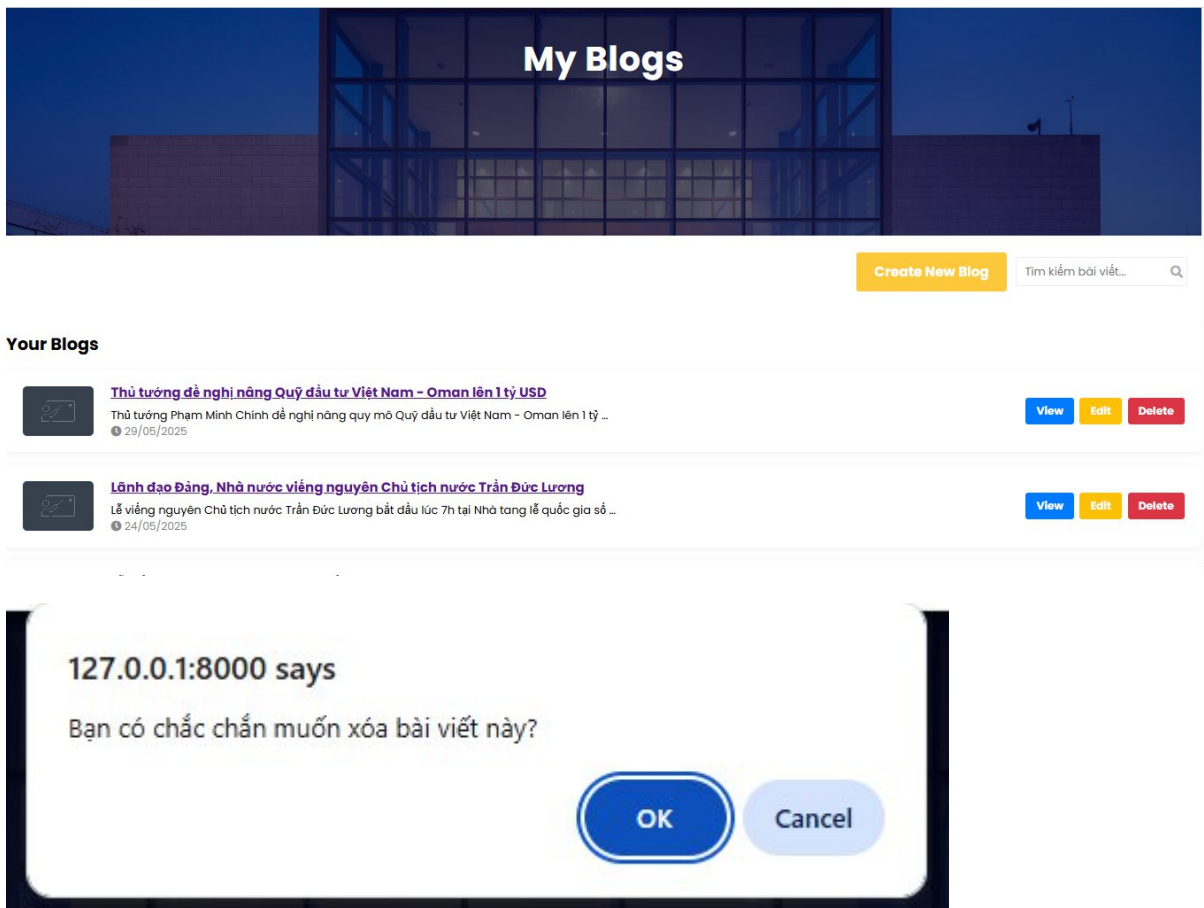
### Biểu đồ hoạt động



### Biểu đồ tuần tự



### Chi tiết giao diện



Nút xóa trên trang my blog(blog cá nhân), hộp thoại xác nhận trước khi xóa.

Frontend: my\_blog.html

Backend: delete\_blog



# CHƯƠNG V: CÀI ĐẶT VÀ CHẠY CHƯƠNG TRÌNH

## 1. Cài đặt chương trình

- Truy cập trang GitHub chứa mã nguồn hệ thống LMS Website ở sau.

`https://github.com/tvhoang12/ttcsPTIT`

- Nhấn nút "Code" và chọn "Download ZIP" hoặc sử dụng lệnh git clone:  
...

```
git clone https://github.com/tvhoang12/ttcsPTIT
...
```

- Giải nén (nếu tải file ZIP) hoặc chuyển vào thư mục dự án:  
...

```
cd sourceCode/lms_website
...
```

- Cài đặt Miniconda (<https://docs.conda.io/en/latest/miniconda.html>) nếu chưa có.

- Tạo và kích hoạt môi trường ảo:  
...

```
conda create -n lms_env python=3.10 -y
conda activate lms_env
...
```

## 2. Cài đặt các thư viện cần thiết

Cài đặt thủ công các thư viện cho code backend.  
...

```
pip install django djangorestframework drf-yasg drf-spectacular django-ckeditor
...
```

## 3. Chạy chương trình

Khởi tạo cơ sở dữ liệu:  
...

```
python manage.py makemigrations
python manage.py migrate
...
```

- (Tùy chọn) Tạo tài khoản admin:  
...

```
python manage.py createsuperuser
...
```

- Thu thập static files:  
...

```
python manage.py collectstatic  
'''
```

- Chạy server phát triển:  
'''

```
python manage.py runserver  
'''
```

- Truy cập hệ thống tại địa chỉ <http://127.0.0.1:8000> trên trình duyệt.