

TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN
**THIẾT KẾ KIẾN TRÚC PHẦN
MỀM**
**XÂY DỰNG WEBSITE BLOG CÁ
NHÂN**

Nhóm thực hiện : Nhóm 24

Sinh viên thực hiện: **Nguyễn Duy Quang**

Mã sinh viên: 165865

Lớp quản lý: **65KSPM**

Giảng viên hướng dẫn: **Phạm Hữu Tùng**

HÀ NỘI 05/2024

Mục lục:

Contents

| | |
|--|----------|
| Chương 1. Giới thiệu đề tài..... | 4 |
| 1.1. Lý do chọn đề tài..... | 4 |
| 1.2. Mục tiêu của đề án..... | 4 |
| 1.3. Phạm vi và giới hạn | 5 |
| 1.4. Phương pháp thực hiện..... | 6 |
| Chương 2. Phân tích hệ thống..... | 7 |
| 2.1. Mô tả bài toán | 7 |
| 2.2. Yêu cầu chức năng..... | 8 |
| 2.3. Yêu cầu phi chức năng..... | 9 |
| 2.4. Use Case Diagram tổng quát | 10 |
| 2.5. Mô tả các Use Case chính..... | 10 |
| a. Đăng nhập..... | 10 |
| b. Đăng bài viết | 11 |
| c. Xem bài viết..... | 12 |
| d. Bình luận bài viết..... | 13 |
| e. Quản lý bài viết | 14 |
| f. Tìm kiếm bài viết..... | 15 |
| 2.6. Biểu đồ hoạt động (<i>Các chức năng chính</i>)..... | 16 |
| a. Đăng nhập..... | 16 |
| b. Đăng ký..... | 17 |
| c. Đăng xuất..... | 18 |
| d. Tạo bài viết..... | 19 |
| e. Chỉnh sửa bài viết | 20 |
| f. Xóa bài viết..... | 21 |
| g. Sửa comment bài viết | 23 |
| h. Xóa comment bài viết..... | 24 |

| | |
|---|-----------|
| i. Cập nhật thông tin người dùng..... | 25 |
| Chương 3. Thiết kế hệ thống..... | 25 |
| 3.1. Lựa chọn mô hình kiến trúc (Monolithic, Microservices, MVC, Layered, Clean Architecture, ...) | 25 |
| 3.2. Biểu đồ tuần tự (các chức năng chính) | 26 |
| a. Đăng nhập..... | 26 |
| b. Đăng ký..... | 27 |
| c. Đăng xuất..... | 27 |
| d. Cập nhật thông tin tài khoản..... | 28 |
| 3.3. Thiết kế lớp (Class Diagram) | 29 |
| Chương 4. Cài đặt và triển khai hệ thống..... | 29 |
| 4.1. Môi trường triển khai..... | 29 |
| 4.2. Cài đặt hệ thống..... | 30 |
| 4.2.1 Yêu cầu hệ thống..... | 30 |
| 4.2.2 Các bước cài đặt chương trình..... | 30 |
| Chương 5. Kết quả và đánh giá..... | 31 |
| 5.1. Kết quả thử nghiệm hệ thống..... | 31 |
| 5.2. Đánh giá hiệu quả hệ thống..... | 41 |
| a. Kết quả đạt được..... | 41 |
| b. Ưu điểm..... | 42 |
| c. Hạn chế..... | 42 |
| d. Kiến nghị cải tiến..... | 42 |
| Chương 6. Tài liệu tham khảo..... | 43 |

Chương 1. Giới thiệu đề tài

1.1. Lý do chọn đề tài

Trong thời đại công nghệ số phát triển mạnh mẽ, việc chia sẻ thông tin, kiến thức và kết nối với cộng đồng thông qua internet trở nên phổ biến và cần thiết. Blog cá nhân không chỉ là nơi người dùng thể hiện quan điểm, lưu trữ trải nghiệm mà còn đóng vai trò như một kênh xây dựng thương hiệu cá nhân và phát triển nghề nghiệp.

Hiện nay, phần lớn người dùng sử dụng mạng xã hội như Facebook, Instagram để chia sẻ nội dung, tuy nhiên các nền tảng này bị hạn chế về khả năng tùy biến, tổ chức nội dung và kiểm soát quyền riêng tư. Vì vậy, nhu cầu xây dựng một hệ thống blog riêng, do chính người dùng quản lý, với đầy đủ tính năng như đăng ký, đăng nhập, viết bài, phân loại nội dung, tương tác qua bình luận,... đang trở nên cấp thiết.

Do đó, em chọn thực hiện đề tài "Xây dựng hệ thống web blog cá nhân" với mong muốn phát triển một nền tảng có thể phục vụ nhu cầu viết bài, phân loại, quản lý nội dung, tương tác người dùng thông qua bình luận, đồng thời đảm bảo giao diện thân thiện, dễ sử dụng và hoàn toàn bằng tiếng Việt. Đề tài vừa mang tính thực tiễn, vừa giúp em củng cố kỹ năng nghề nghiệp cần thiết cho công việc sau này.

1.2. Mục tiêu của đồ án

Mục tiêu của đồ án là xây dựng một hệ thống website blog cá nhân hoàn chỉnh, thân thiện với người dùng Việt Nam, có khả năng hoạt động ổn định và hỗ trợ đầy đủ các chức năng cơ bản của một nền tảng blog hiện đại. Cụ thể:

- Cung cấp giao diện người dùng đơn giản, dễ sử dụng, hỗ trợ đầy đủ tiếng Việt, có thể truy cập tốt trên cả máy tính và thiết bị di động.
- Cho phép người dùng đăng ký, đăng nhập, đăng xuất và quản lý tài khoản cá nhân một cách bảo mật và tiện lợi.
- Hỗ trợ viết bài, chỉnh sửa, xóa bài viết, phục vụ nhu cầu chia sẻ thông tin cá nhân, kinh nghiệm, kiến thức.
- Phân loại bài viết theo danh mục, giúp tổ chức nội dung một cách có hệ thống.
- Hiện thị danh sách bài viết mới nhất, chi tiết từng bài viết, kèm theo chức năng tìm kiếm và lọc theo từ khóa hoặc danh mục.

- Tích hợp chức năng bình luận nhằm tạo môi trường tương tác văn minh giữa người đọc và người viết.
- Ghi nhận lượt xem bài viết và hiển thị thống kê, phục vụ cho mục đích đánh giá mức độ phổ biến của nội dung.
- Xây dựng hệ thống theo mô hình MVC, sử dụng các công nghệ như PHP (Laravel Framework), HTML, CSS, JavaScript và cơ sở dữ liệu MySQL.

Thông qua việc thực hiện đề tài, em cũng đặt mục tiêu nâng cao kỹ năng phân tích, thiết kế hệ thống phần mềm, triển khai ứng dụng thực tế, từ đó chuẩn bị tốt hơn cho công việc sau khi tốt nghiệp.

1.3. Phạm vi và giới hạn

Phạm vi thực hiện

Đề tài tập trung xây dựng một hệ thống web blog cá nhân với các chức năng chính bao gồm:

- Đăng ký, đăng nhập, đăng xuất tài khoản người dùng.
- Tạo, chỉnh sửa, xóa và lưu bài viết.
- Phân loại bài viết theo danh mục (ví dụ: Công nghệ, Du lịch, Sách, Thủ thuật...).
- Hiển thị danh sách bài viết, chi tiết bài viết và chức năng tìm kiếm.
- Quản lý bình luận: người dùng có thể để lại bình luận.
- Theo dõi lượt xem bài viết và hiển thị thống kê.
- Quản trị viên có quyền duyệt nội dung, quản lý người dùng, quản lý danh mục, bài viết.

Giới hạn của hệ thống

- Chưa tích hợp các tính năng nâng cao như: gửi email tự động, chia sẻ lên mạng xã hội.
- Hệ thống chỉ hỗ trợ tiếng Việt và hướng đến người dùng trong nước.
- Tính năng quản trị mới dừng lại ở mức cơ bản, chưa có phân quyền chi tiết theo vai trò.

- Không triển khai chức năng thanh toán, quảng cáo hay lưu trữ đa phương tiện (video/audio).

1.4. Phương pháp thực hiện

Để xây dựng hệ thống blog cá nhân, nhóm sử dụng phương pháp phát triển phần mềm theo hướng tiếp cận Module hóa và hướng đối tượng, kết hợp với mô hình MVC (Model – View – Controller) của Laravel. Phương pháp thực hiện cụ thể như sau:

- **Phân tích yêu cầu hệ thống:** Dựa trên nhu cầu thực tế và chức năng mong muốn từ phía người dùng, nhóm phân tích các chức năng chính như quản lý bài viết, người dùng, bình luận, và thống kê lượt xem.

- **Thiết kế hệ thống:**

Thiết kế sơ đồ Use Case, sơ đồ lớp (Class Diagram) và cơ sở dữ liệu quan hệ để xác định cấu trúc logic và mối quan hệ giữa các thực thể như: users, posts, categories, comments, contacts.

- **Xây dựng hệ thống:**

+ Ngôn ngữ lập trình: PHP sử dụng Laravel Framework với các thành phần đặc trưng như:

- Hệ thống route (routes/web.php), controller (app/Http/Controllers), model (app/Models) và view (resources/views/web) theo mô hình MVC.
- Sử dụng Middleware và hệ thống Laravel Auth để xử lý phân quyền người dùng.

- **Giao diện người dùng:** Xây dựng bằng HTML, CSS, JavaScript và sử dụng Bootstrap để hỗ trợ responsive trên các thiết bị. Ngoài ra, nhóm sử dụng: File custom.js để xử lý các tương tác trên giao diện.

- **Cơ sở dữ liệu:** Sử dụng hệ quản trị cơ sở dữ liệu MySQL, quản lý qua công cụ DBeaver. Cấu trúc được thiết lập thông qua thư mục database/migrations, bao gồm các bảng:

- + users: quản lý tài khoản người dùng
- + categories: phân loại bài viết

- + posts: nội dung bài viết
- + comments: lưu bình luận của người dùng
- + contacts: lưu thông tin liên hệ
- + Các mối quan hệ giữa bảng như: posts liên kết với categories, comments liên kết với posts.

- Công cụ hỗ trợ phát triển:

- + Visual Studio Code: viết và tổ chức mã nguồn.
- + DBeaver: quản lý cơ sở dữ liệu MySQL.

Thông qua việc triển khai theo cách tiếp cận module và phân lớp rõ ràng, hệ thống dễ dàng bảo trì, nâng cấp và mở rộng trong tương lai.

Chương 2. Phân tích hệ thống

2.1. Mô tả bài toán

Trong thời đại công nghệ phát triển mạnh mẽ như hiện nay, việc xây dựng và duy trì một nền tảng blog cá nhân không chỉ đơn thuần là việc chia sẻ cảm xúc hay kiến thức mà còn là một công cụ quan trọng giúp người dùng thể hiện bản thân, kết nối với cộng đồng và phát triển nghề nghiệp. Đặc biệt, trong bối cảnh mà mạng xã hội đang chiếm lĩnh không gian trực tuyến, việc tạo ra một trang web blog cá nhân không chỉ cần đáp ứng nhu cầu về nội dung mà còn phải đảm bảo tính năng tiện dụng, dễ dàng tương tác và quản lý hiệu quả. Chính vì vậy, chúng ta cần một hệ thống blog hoàn chỉnh với các chức năng cơ bản như đăng ký, đăng nhập, quản lý bài viết, phân loại nội dung và tương tác qua bình luận. Hệ thống cần phải thân thiện với người sử dụng, đặc biệt là người dùng Việt Nam, với giao diện hoàn toàn bằng tiếng Việt và khả năng tối ưu hóa trên tất cả các thiết bị.

Trong xu hướng hiện đại, các yêu cầu về bảo mật, hiệu suất và tính tương tác cao ngày càng trở nên quan trọng. Việc người dùng có thể dễ dàng tạo và quản lý các bài viết, phân loại bài viết theo các danh mục như công nghệ, du lịch, hay sách, đồng thời quản lý bình luận và theo dõi lượt xem, không chỉ giúp nâng cao trải nghiệm người dùng mà còn tạo ra một không gian chia sẻ thông tin, kiến thức hữu ích. Từ đó, hệ thống này sẽ không chỉ đơn thuần là một blog cá nhân mà còn là một công cụ giúp người dùng phát triển bản thân, giao lưu

và học hỏi từ cộng đồng. Để đáp ứng những yêu cầu này, nhóm chúng tôi đã quyết định xây dựng một hệ thống blog với đầy đủ tính năng từ đăng ký, đăng nhập cho đến quản lý bài viết, bình luận và lượt xem, đồng thời đảm bảo tính bảo mật cao, chính xác và dễ sử dụng. Với sự hỗ trợ của công nghệ hiện đại, hệ thống này sẽ giúp người dùng dễ dàng quản lý nội dung cá nhân và tối ưu hóa trải nghiệm trực tuyến, từ đó nâng cao sự hiện diện và kết nối của họ trên nền tảng mạng xã hội.

2.2. Yêu cầu chức năng

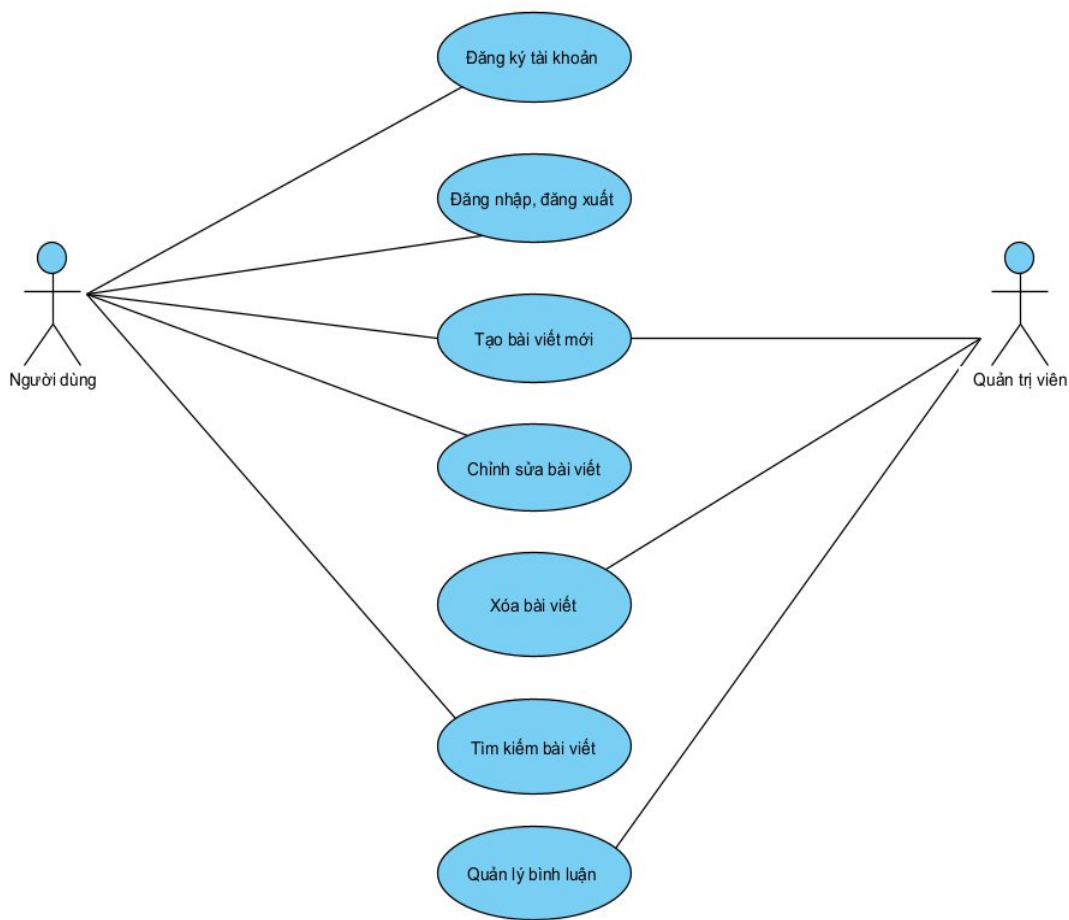
| STT | Chức năng | Mô tả |
|-----|--------------------|---|
| 1 | Đăng ký/Đăng nhập | Người dùng tạo tài khoản, đăng nhập hệ thống |
| 2 | Quản lý bài viết | Tạo, sửa, xóa bài viết |
| 3 | Phân loại bài viết | Gắn bài viết vào danh mục cụ thể (VD: công nghệ, ẩm thực, du lịch...) |
| 4 | Hiển thị bài viết | Xem danh sách bài viết mới nhất, chi tiết bài viết |
| 5 | Tìm kiếm bài viết | Tìm kiếm theo từ khóa hoặc danh mục |
| 6 | Quản lý bình luận | Người dùng bình luận, admin duyệt bình luận |
| 7 | Theo dõi lượt xem | Ghi nhận và thống kê số lượt xem bài viết |
| 8 | Quản trị hệ thống | Quản lý người dùng, danh mục, bài viết |

2.3. Yêu cầu phi chức năng

- Khả năng mở rộng: Hệ thống dễ dàng thêm mới các chức năng như đánh giá bài viết, phân quyền chi tiết,...
- Hiệu suất: Hệ thống phản hồi nhanh, tải trang trong thời gian ngắn.
- Tính tương thích: Chạy ổn định trên các trình duyệt phổ biến (Chrome, Firefox, Edge).
- Bảo mật: Mã hóa mật khẩu.

- Thân thiện với người dùng: Giao diện tiếng Việt, bố cục rõ ràng, dễ sử dụng với người mới.
- Khả năng bảo trì: Cấu trúc rõ ràng, sử dụng Laravel theo chuẩn giúp dễ bảo trì, nâng cấp sau này.

2.4. Use Case Diagram tổng quát



2.5. Mô tả các Use Case chính

a. Đăng nhập

| | |
|------------------|--|
| Name | Đăng nhập |
| ID | UC01 |
| Description | Cho phép người dùng đăng nhập vào hệ thống bằng tên đăng nhập và mật khẩu. |
| Actor | Người dùng |
| Priority | Độ ưu tiên: 1 |
| Pre-Condition | Điều kiện để Use Case thực hiện thành công: - Người dùng đã có tài khoản hợp lệ. |
| Post-Condition | Những thứ sẽ xuất hiện nếu Use Case thực hiện thành công: - Người dùng được chuyển đến trang chính sau khi đăng nhập thành công. |
| Basic Flow | Luồng tương tác chính giữa các Actor và System để Use case thực hiện thành công: - Truy cập trang đăng nhập. - Nhập thông tin tài khoản. - Nhấn “Đăng nhập”. - Hệ thống xác thực và chuyển trang nếu đúng. |
| Alternative Flow | Luồng thay thế giữa Actor và hệ thống: |
| Exception Flow | Luồng tương tác ngoại lệ giữa các Actor và System mà Use case thực hiện thất bại: - Sai mật khẩu hoặc tài khoản → hiển thị lỗi. - Tài khoản bị khóa → thông báo lỗi. |

b. Đăng bài viết

| | |
|------------------|---|
| Name | Đăng bài viết |
| ID | UC02 |
| Description | Người dùng có thể đăng bài viết mới gồm tiêu đề, nội dung, danh mục và tag. |
| Actor | Người dùng |
| Priority | Độ ưu tiên: 1 |
| Pre-Condition | Điều kiện để Use Case thực hiện thành công: <ul style="list-style-type: none">- Đã đăng nhập |
| Post-Condition | Những thứ sẽ xuất hiện nếu Use Case thực hiện thành công: <ul style="list-style-type: none">- Bài viết được lưu và hiển thị trên hệ thống. |
| Basic Flow | Luồng tương tác chính giữa các Actor và System để Use case thực hiện thành công: <ul style="list-style-type: none">- Vào trang đăng bài.- Nhập thông tin.- Nhấn “Đăng bài”.- Hệ thống lưu bài viết. |
| Alternative Flow | Luồng thay thế giữa Actor và hệ thống: <ul style="list-style-type: none">- Nhấn 'Lưu nháp' → bài viết lưu ở trạng thái nháp. |
| Exception Flow | Luồng tương tác ngoại lệ giữa các Actor và System mà Use case thực hiện thất bại: <ul style="list-style-type: none">-Thiếu tiêu đề/nội dung → hiển thị lỗi. |

c. Xem bài viết

| | |
|------------------|---|
| Name | Xem bài viết |
| ID | UC03 |
| Description | Người dùng có thể xem chi tiết một bài viết. |
| Actor | Người dùng |
| Priority | Độ ưu tiên: 2 |
| Pre-Condition | Điều kiện để Use Case thực hiện thành công: - Hệ thống có bài viết hợp lệ. |
| Post-Condition | Những thứ sẽ xuất hiện nếu Use Case thực hiện thành công: - Lượt xem tăng lên 1. |
| Basic Flow | Luồng tương tác chính giữa các Actor và System để Use case thực hiện thành công: - Chọn bài viết từ danh sách. - Hệ thống hiển thị nội dung bài viết. |
| Alternative Flow | Luồng thay thế giữa các Actor và System khi UseCase thực hiện thành công: - Bài viết bị xóa hoặc không tồn tại → thông báo lỗi. |

d. Bình luận bài viết

| | |
|------------------|---|
| Name | Bình luận bài viết |
| ID | UC04 |
| Description | Người dùng đã đăng nhập có thể để lại bình luận dưới bài viết |
| Actor | Người dùng |
| Priority | Mức độ ưu tiên: 2 |
| Pre-Condition | Điều kiện để Use Case thực hiện thành công: -Đã đăng nhập, bài viết tồn tại. |
| Post-Condition | Những thứ sẽ xuất hiện nếu Use Case thực hiện thành công: - Bình luận được hiển thị. |
| Basic Flow | Luồng tương tác chính giữa các Actor và System để Use case thực hiện thành công: - Nhập bình luận ở phần bài viết. - Nhấn “Gửi bình luận”. - Hệ thống lưu và hiển thị bình luận. |
| Alternative Flow | Luồng thay thế giữa các Actor và System khi UseCase thực hiện thành công: |
| Exception Flow | Luồng tương tác ngoại lệ giữa các Actor và System mà Use case thực hiện thất bại: - Nội dung trống → hiển thị lỗi - Hệ thống lỗi DB → thông báo thất bại. |

e. Quản lý bài viết

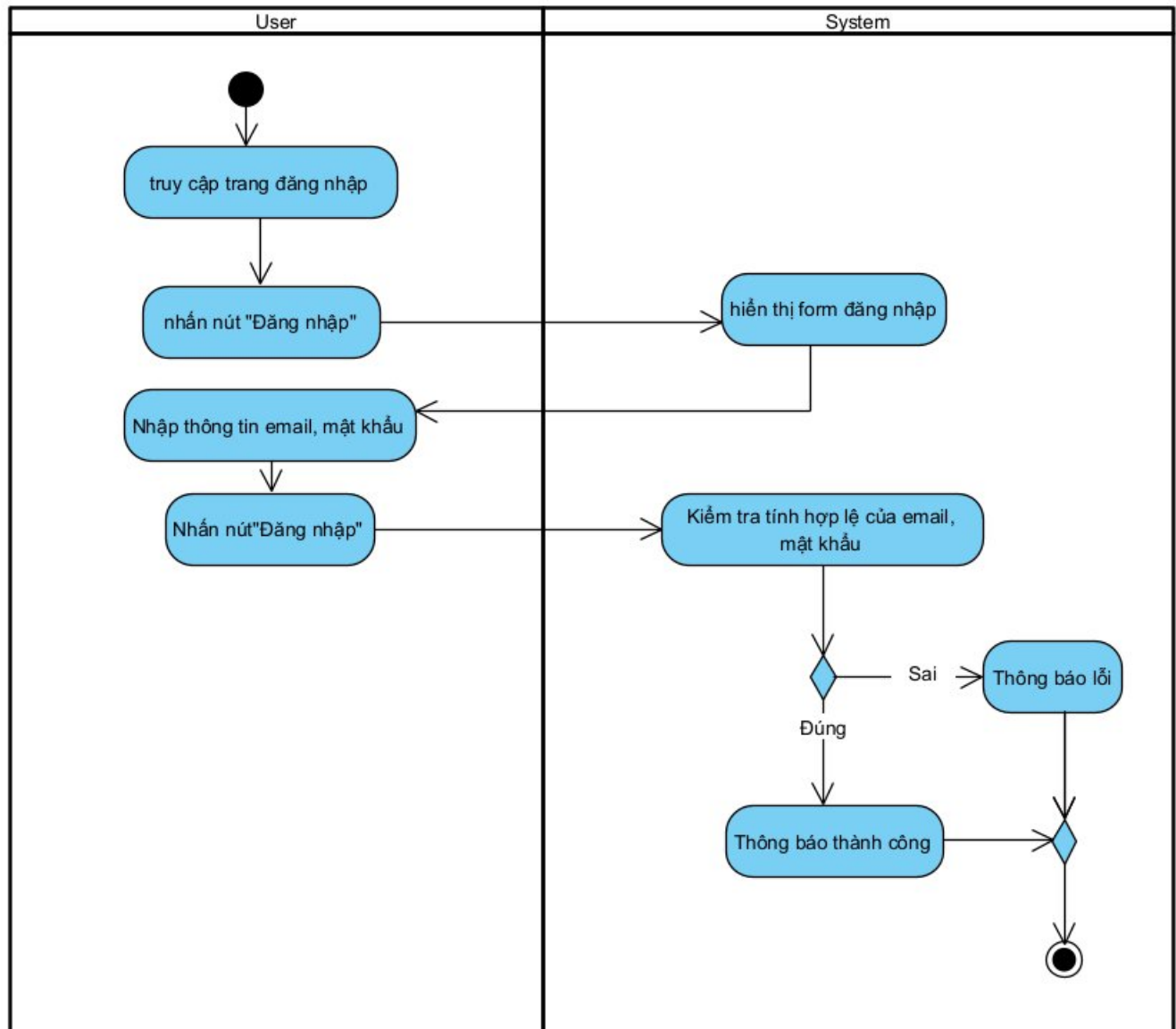
| | |
|------------------|---|
| Name | Quản lý bài viết |
| ID | UC05 |
| Description | Quản trị viên có thể xem danh sách, chỉnh sửa, xóa các bài viết. |
| Actor | Quản trị viên |
| Priority | Độ ưu tiên: 1 |
| Pre-Condition | Điều kiện để Use Case thực hiện thành công: <ul style="list-style-type: none">- Đăng nhập với vai trò Admin (Người quản lý) |
| Post-Condition | Những thứ sẽ xuất hiện nếu Use Case thực hiện thành công: <ul style="list-style-type: none">- Thao tác được lưu. |
| Basic Flow | Luồng tương tác chính giữa các Actor và System để Use case thực hiện thành công: <ul style="list-style-type: none">- Vào trang quản lý bài viết.- Chọn bài viết.- Thực hiện thao tác (Sửa/Xóa).- Hệ thống cập nhật dữ liệu. Người |
| Alternative Flow | Luồng thay thế giữa các Actor và System khi UseCase thực hiện thành công: |

f. Tìm kiếm bài viết

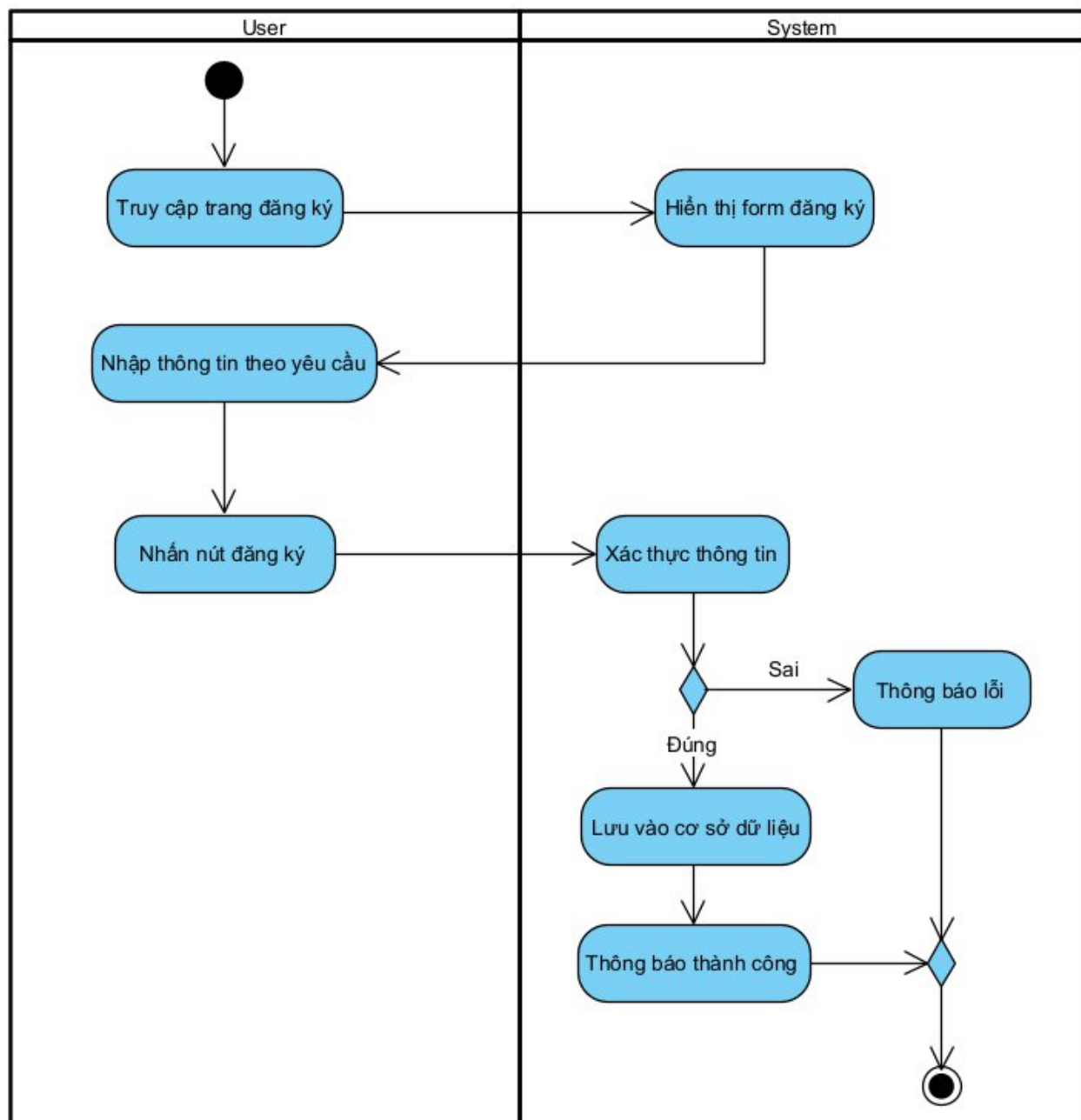
| | |
|------------------|--|
| Name | Tìm kiếm bài viết |
| ID | UC06 |
| Description | Người dùng có thể tìm bài viết theo từ khóa, tag hoặc danh mục |
| Actor | Người dùng |
| Priority | Độ ưu tiên: 2 |
| Pre-Condition | Điều kiện để Use Case thực hiện thành công: -Hệ thống có bài viết đã đăng. |
| Post-Condition | Những thứ sẽ xuất hiện nếu Use Case thực hiện thành công: - Hiện thị danh sách kết quả phù hợp |
| Basic Flow | Luồng tương tác chính giữa các Actor và System để Use case thực hiện thành công: - Nhập từ khóa vào thanh tìm kiếm. - Nhấn “Tìm”. - Hệ thống hiển thị kết quả |
| Alternative Flow | - Luồng thay thế giữa Actor và hệ thống: |
| Exception Flow | Luồng tương tác ngoại lệ giữa các Actor và System mà Use case thực hiện thất bại: |

2.6. Biểu đồ hoạt động (Các chức năng chính)

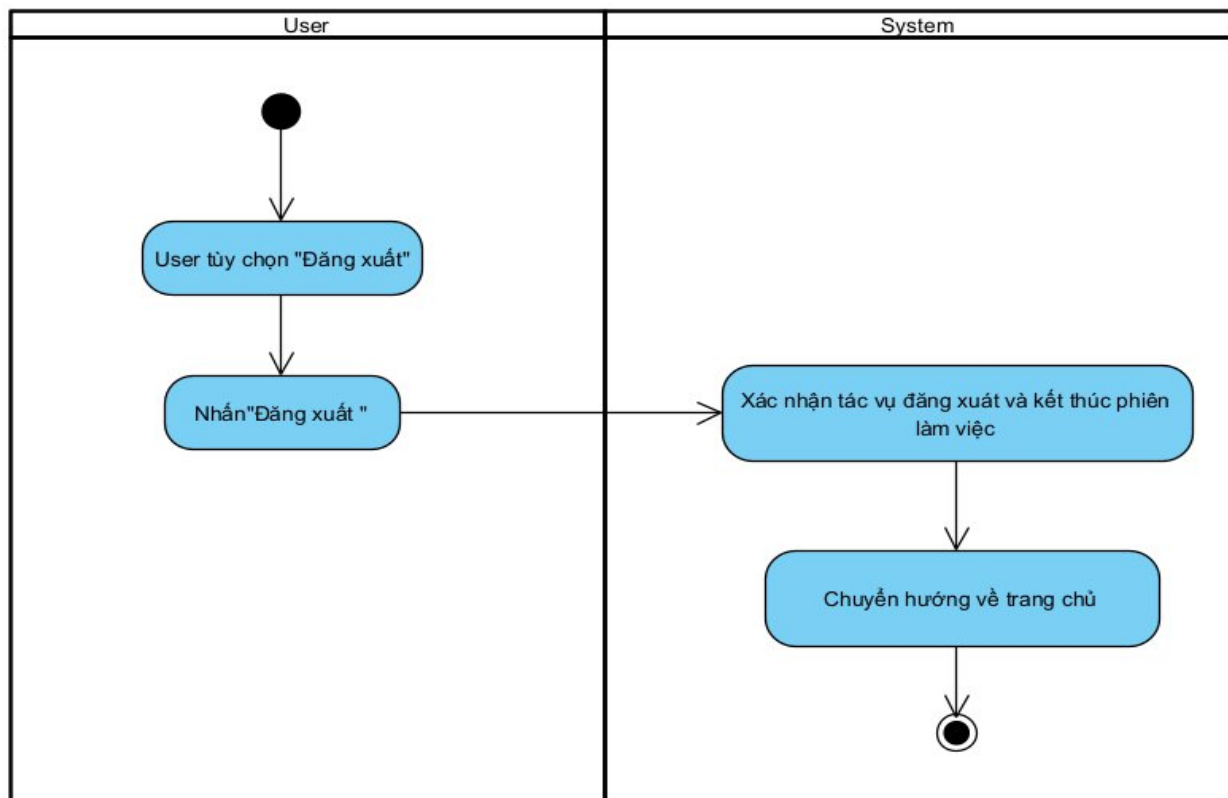
a. Đăng nhập



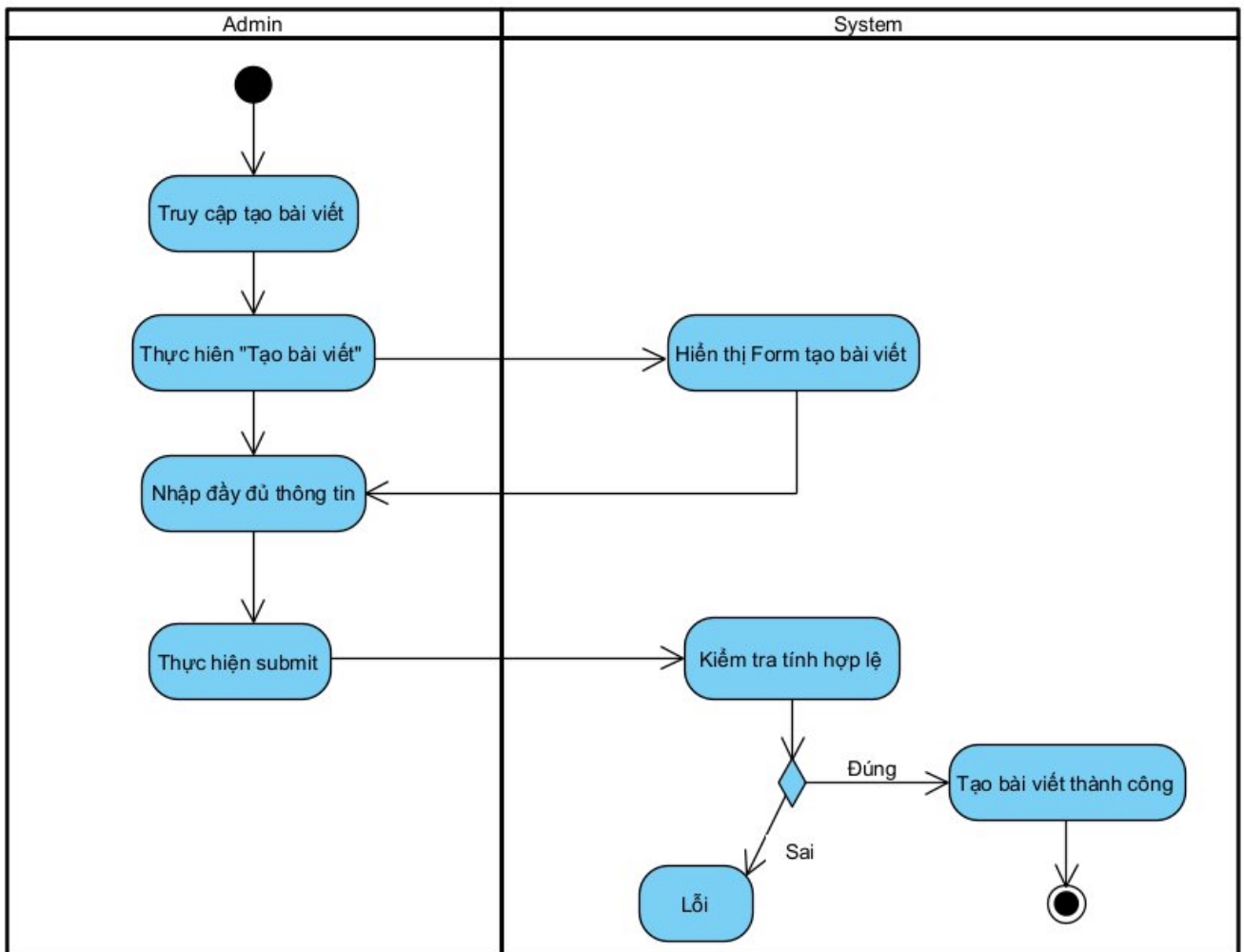
b. Đăng ký



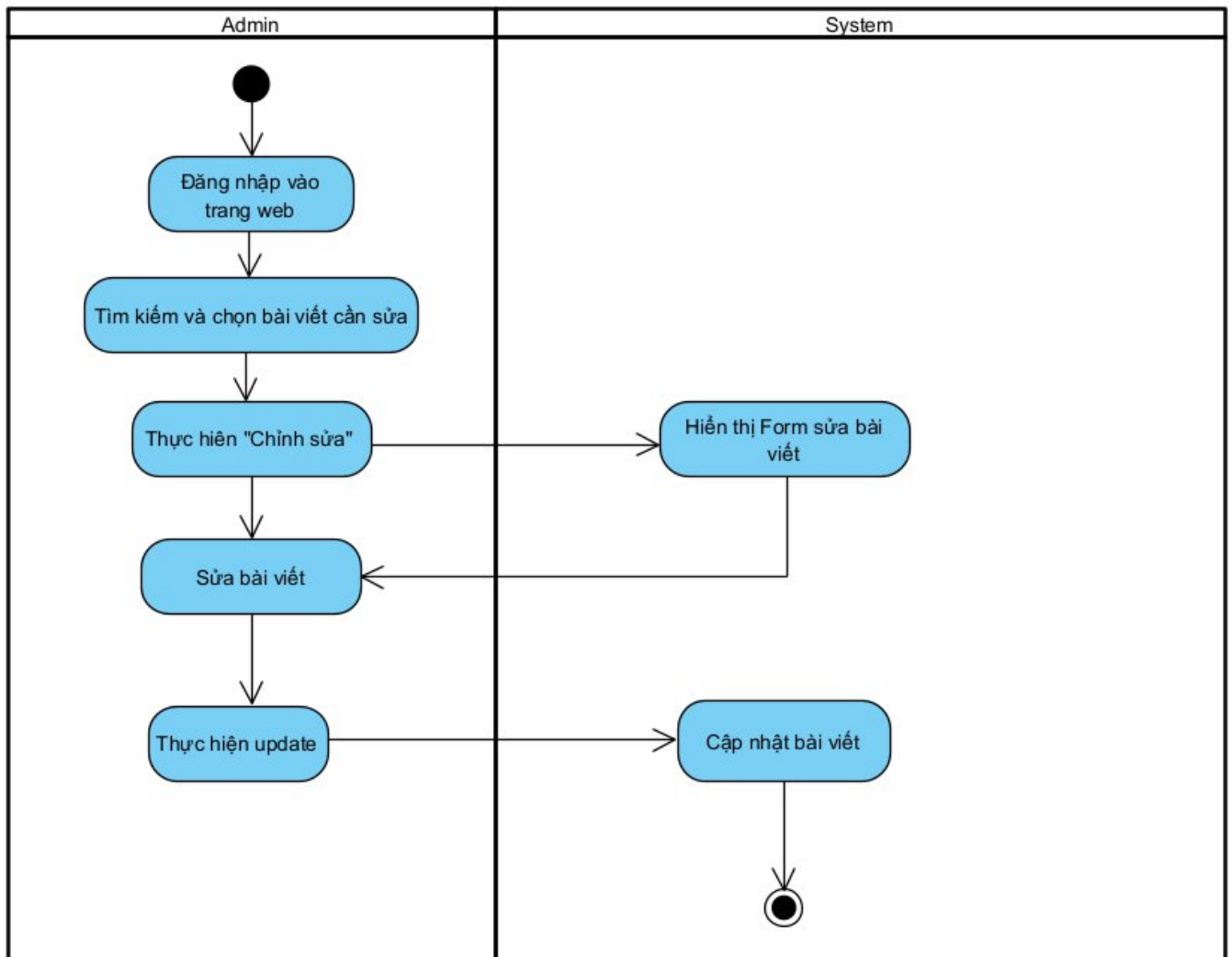
c. Đăng xuất



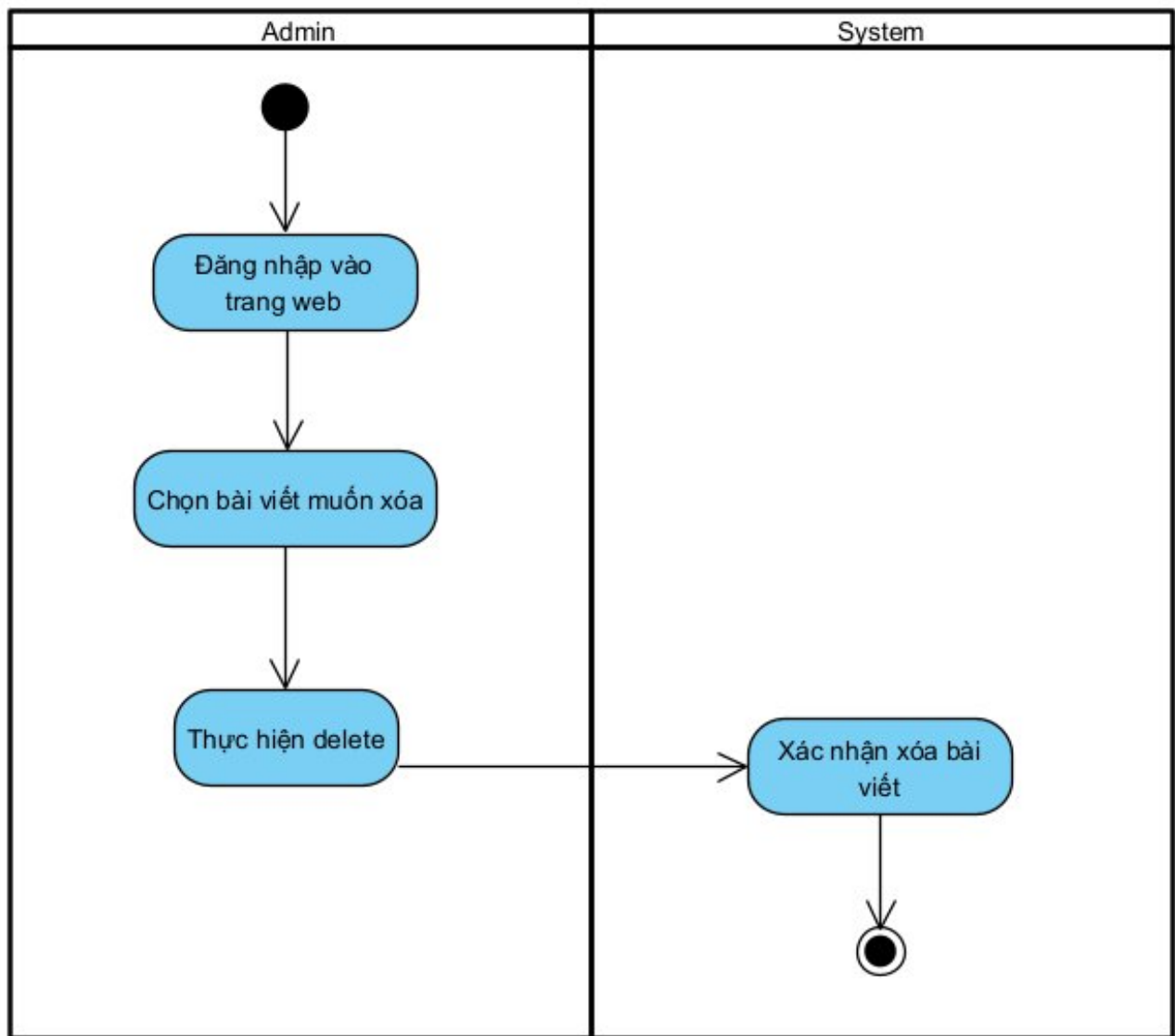
d. Tạo bài viết



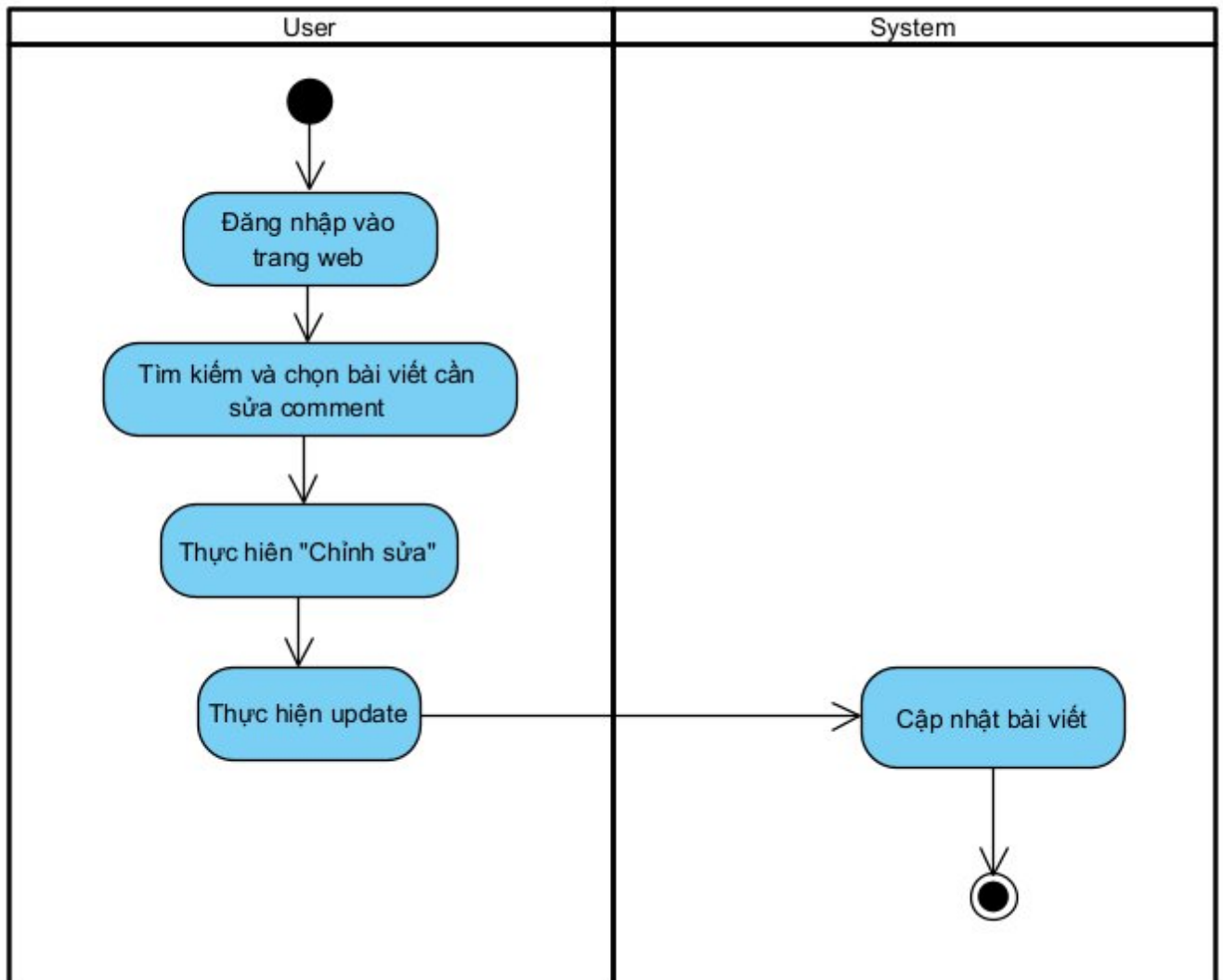
e. Chỉnh sửa bài viết



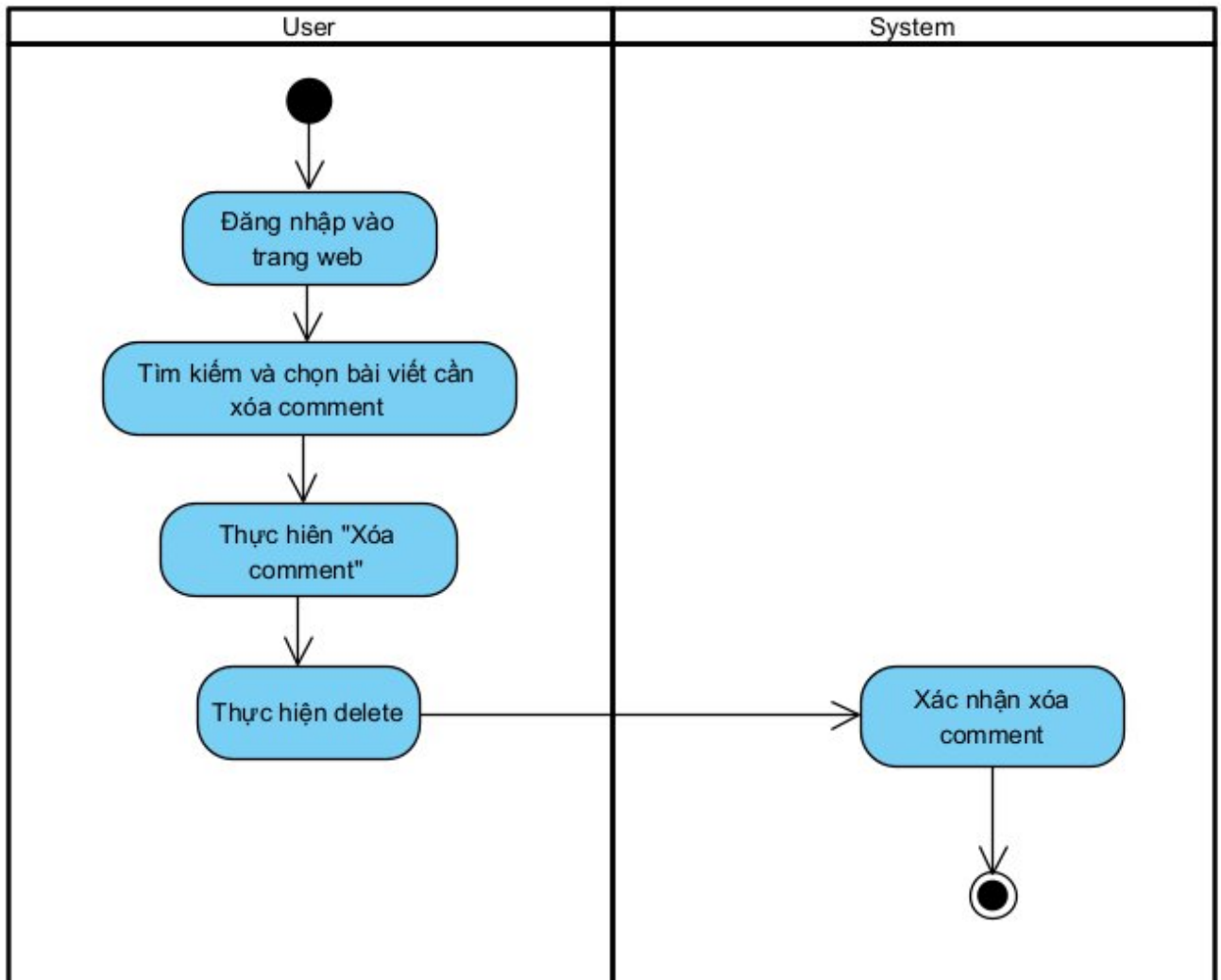
f. Xóa bài viết



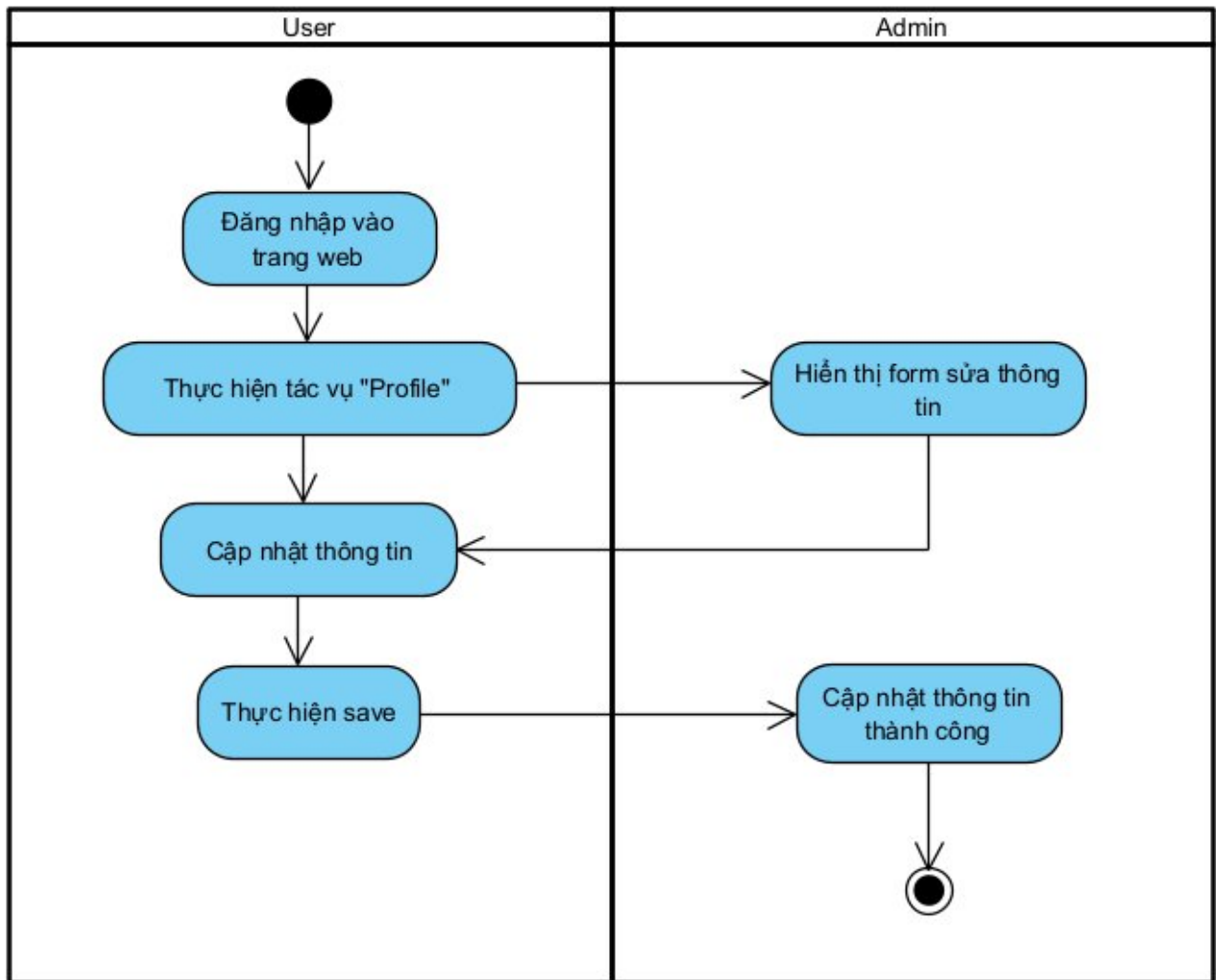
g. Sửa comment bài viết



h. Xóa comment bài viết



i. Cập nhật thông tin người dùng



Chương 3. Thiết kế hệ thống

3.1. Lựa chọn mô hình kiến trúc (Monolithic, Microservices, MVC, Layered, Clean Architecture, ...)

Để đảm bảo khả năng phát triển, mở rộng và bảo trì hệ thống một cách hiệu quả, nhóm lựa chọn mô hình kiến trúc MVC (Model – View – Controller) kết hợp với Laravel Framework.

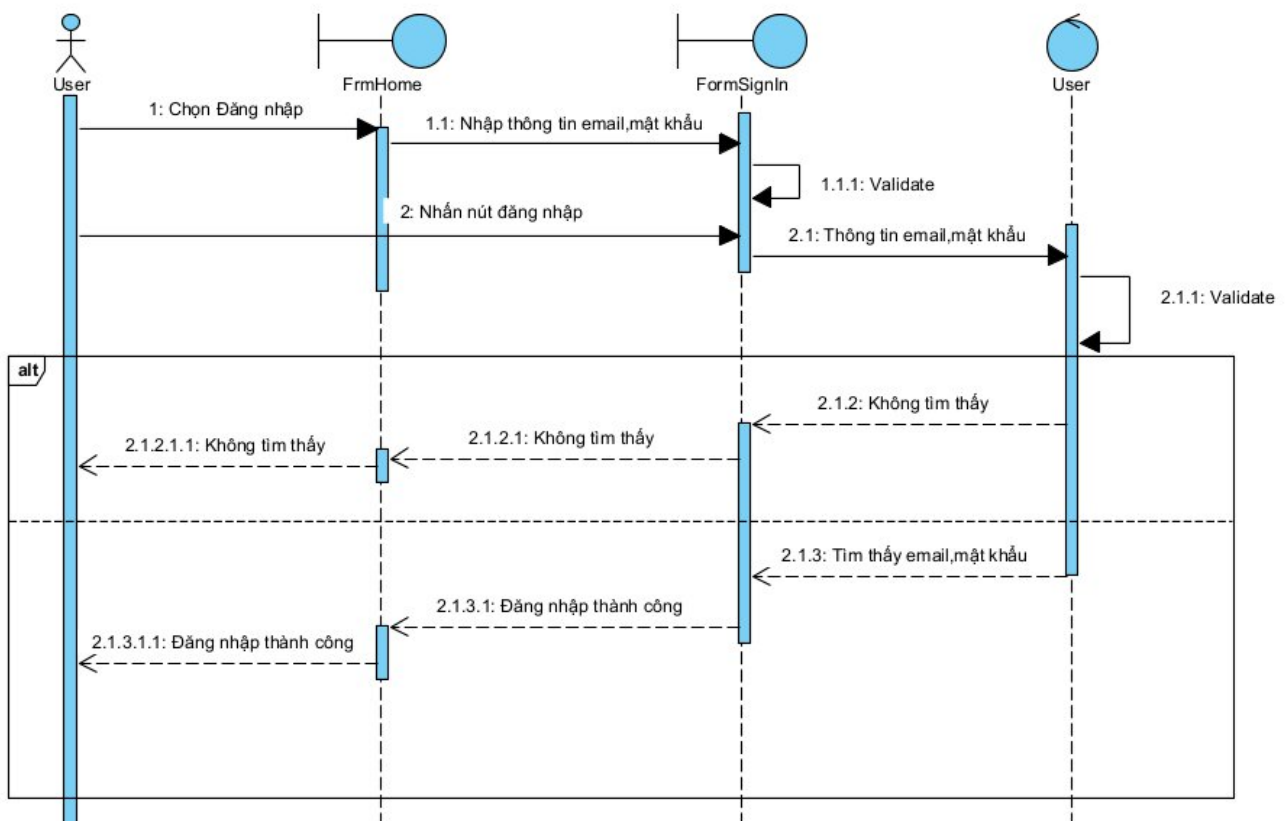
- Model: Chứa các lớp đại diện cho dữ liệu và xử lý nghiệp vụ, bao gồm các model như User, Post, Category, Comment, Contact.

- View: Sử dụng Blade Template Engine để hiển thị giao diện người dùng; các file được đặt trong thư mục resources/views/web.
- Controller: Xử lý các yêu cầu từ người dùng, tương tác với model và trả kết quả về cho view. Được tổ chức trong các thư mục như:
 - + Web: giao diện chính của người dùng
 - + Admin: giao diện quản trị viên
 - + Auth: xử lý xác thực người dùng

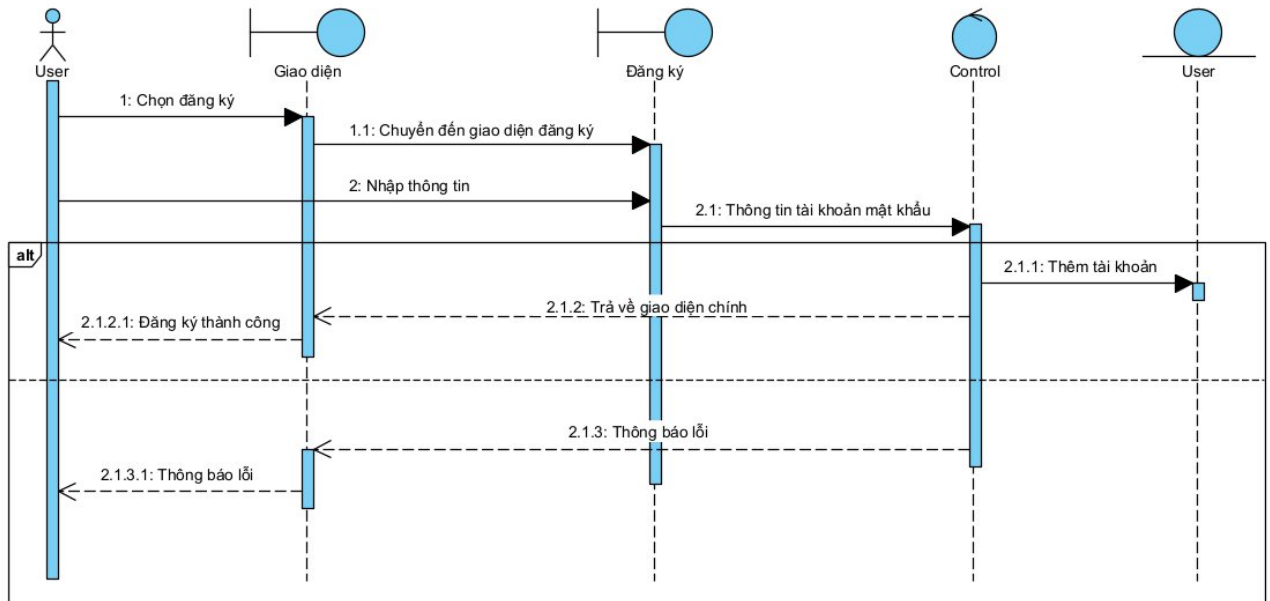
Laravel hỗ trợ đầy đủ các thành phần cần thiết như hệ thống định tuyến, middleware, bảo mật CSRF, validation và ORM (Eloquent), giúp hệ thống dễ mở rộng, tái sử dụng và đảm bảo an toàn.

3.2. Biểu đồ tuần tự (các chức năng chính)

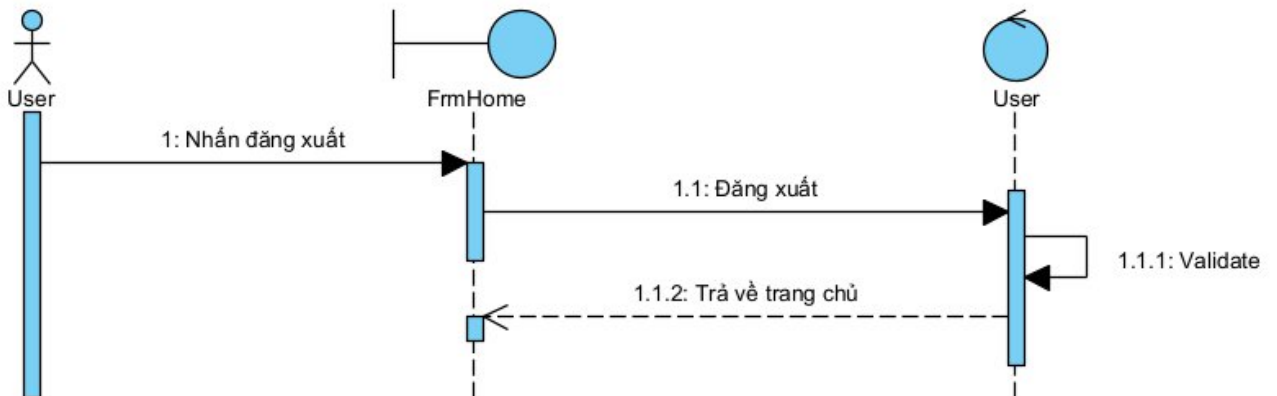
a. Đăng nhập



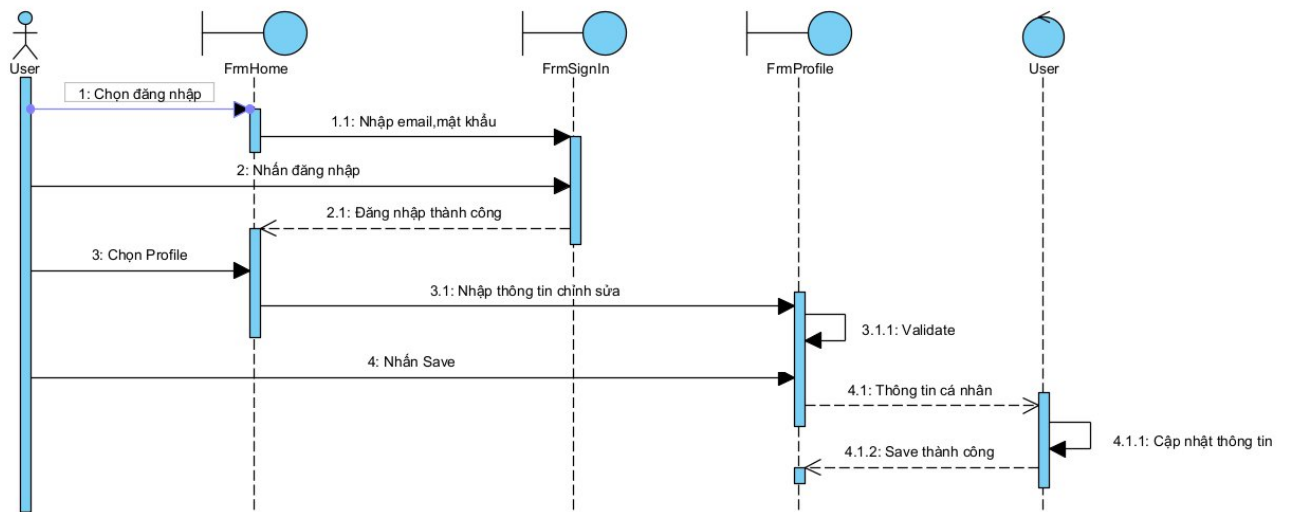
b. Đăng ký



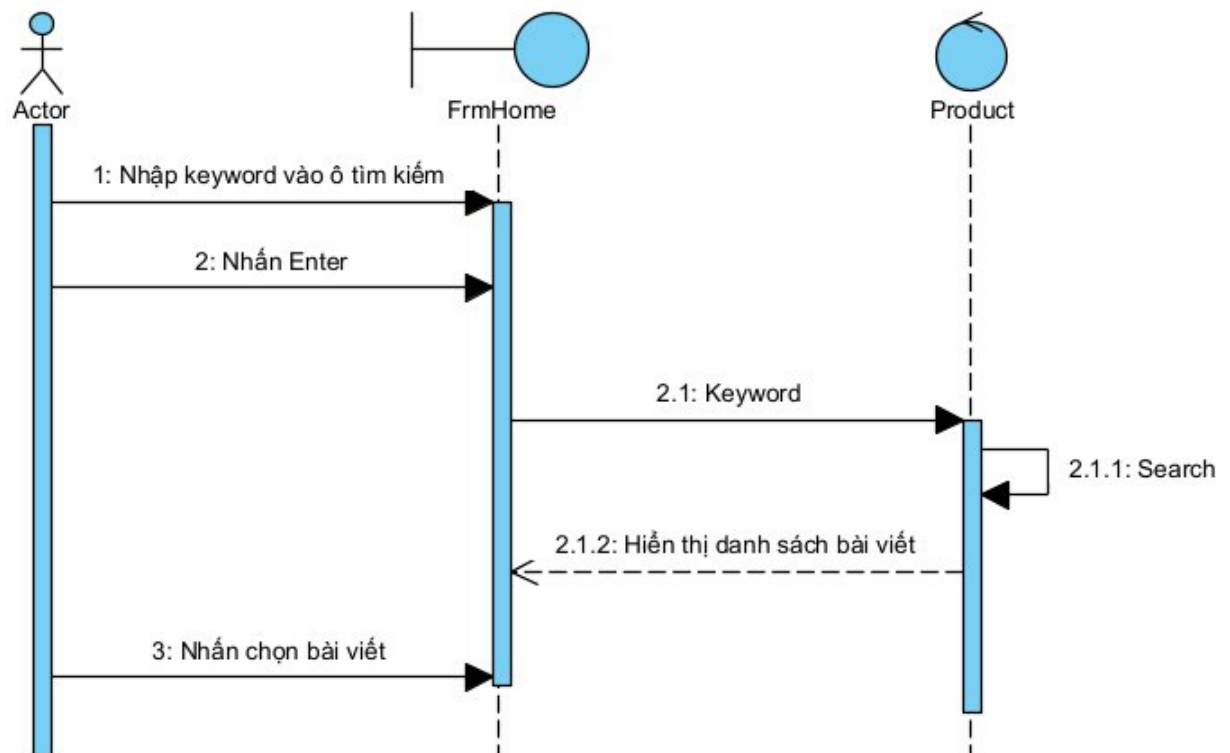
c. Đăng xuất



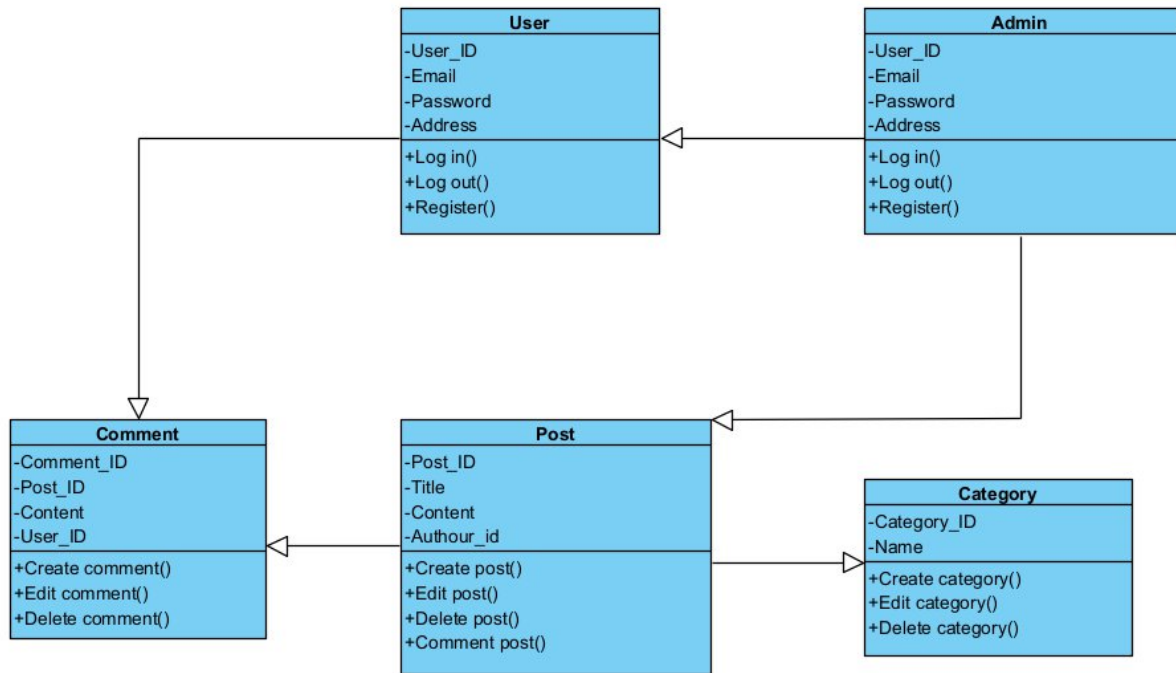
d. Cập nhật thông tin tài khoản



e. Xem bài viết



3.3. Thiết kế lớp (Class Diagram)



Chương 4. Cài đặt và triển khai hệ thống

4.1. Môi trường triển khai

Hệ thống được xây dựng trên nền tảng Laravel Framework và yêu cầu các môi trường sau:

- PHP: Version 8.0 trở lên
- MySQL: Version 8.0 trở lên
- Web Server: Apache/Nginx
- Composer: Package manager của PHP
- Node.js: Version 14.17.0 trở lên
- NPM: Package manager của Node.js

4.2. Cài đặt hệ thống

4.2.1 Yêu cầu hệ thống

a. Yêu cầu về phần cứng:

- CPU: 2.0 GHz trở lên
- RAM: 4GB trở lên
- Ổ cứng: 10GB trống trở lên
- Mạng: Kết nối Internet ổn định

b. Yêu cầu về phần mềm:

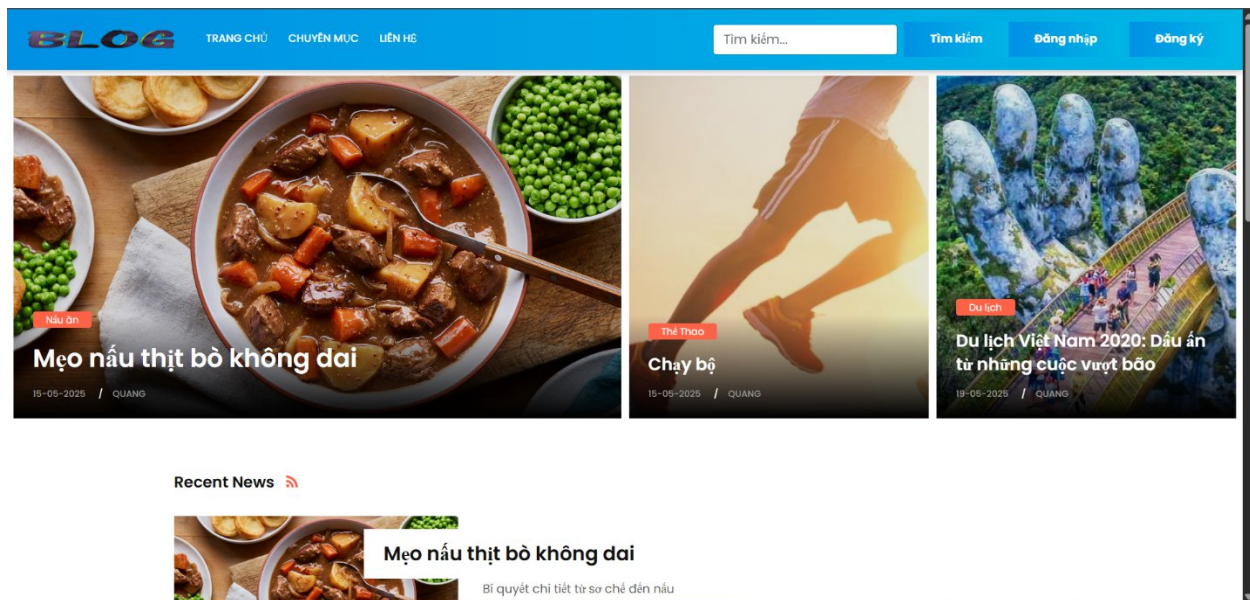
- Hệ điều hành: Windows 10/11
- PHP 8.0 trở lên
- MySQL 8.0 trở lên
- Composer
- Node.js 14.17.0 trở lên
- NPM
- Web Server

4.2.2 Các bước cài đặt chương trình

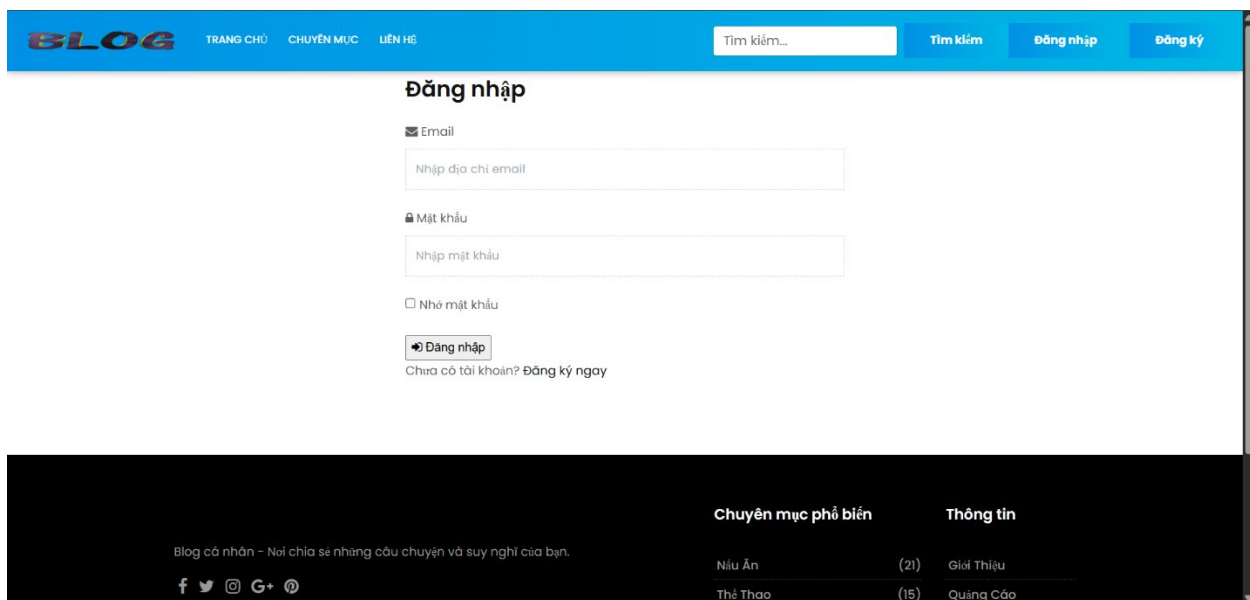
- Mở folder trong visual studio code.
- Nếu chưa có folder “ vendor ” thì bật terminal chạy lệnh “ composer intall ” để cài đặt các thư viện khai báo ở file “ composer.json ”
- Bật terminal chạy lệnh “ php artisan serve ”.
- Nhấp vào đường link “ http://127.0.0.1:8000 ” để chạy chương trình.
- Đăng ký tài khoản để tạo bài viết.
- Để truy cập quyền admin hãy vào đường dẫn “ http://127.0.0.1:8000/admin/login ”, đăng nhập với tài khoản: “ admin@example.com ” và mật khẩu: “ password ”

Chương 5. Kết quả và đánh giá

5.1. Kết quả thử nghiệm hệ thống



Hình 1. Trang chủ



Hình 2. Giao diện đăng nhập

BLOG

TRANG CHỦ

CHUYÊN MỤC

LIÊN HỆ

Tìm kiếm...

Tìm kiếm

Đăng nhập

Đăng ký

Đăng ký tài khoản

Tên đầy đủ

Nhập tên đầy đủ

Email

Nhập địa chỉ email

Mật khẩu

Nhập mật khẩu

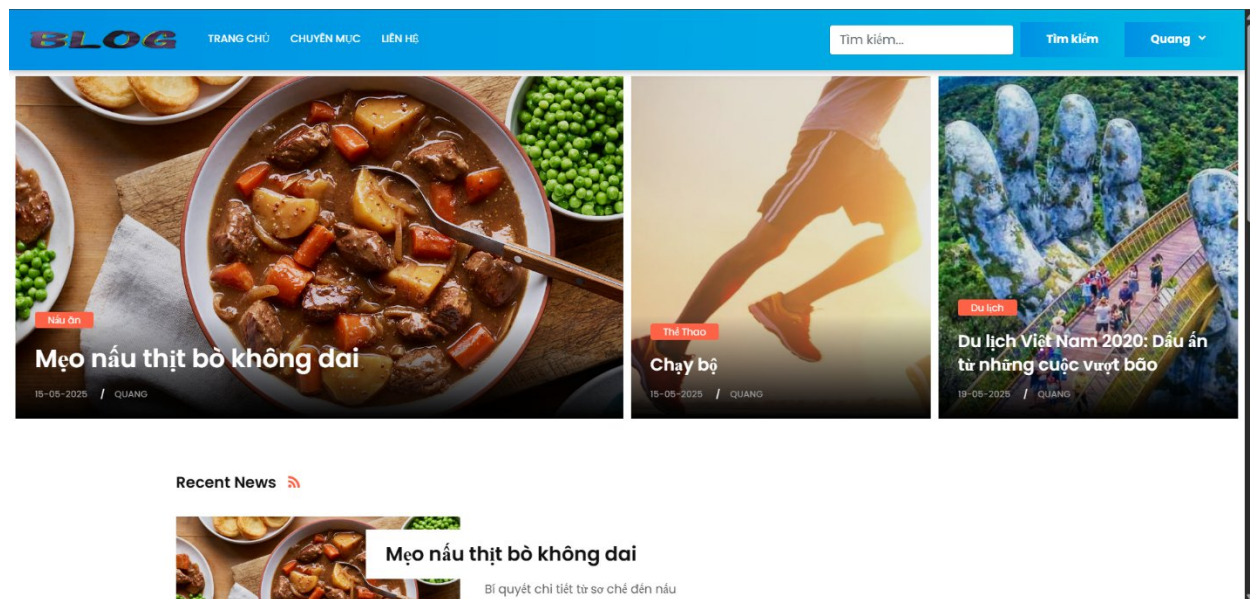
Xác nhận mật khẩu

Nhập lại mật khẩu

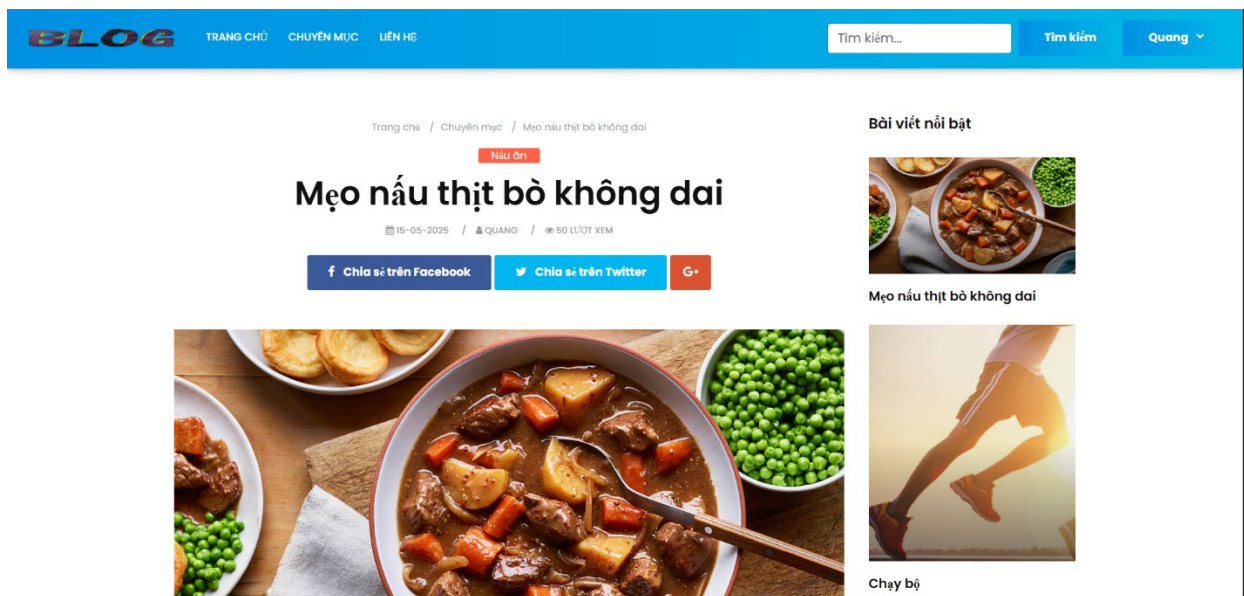
Đăng ký tài khoản

Đã có tài khoản? Đăng nhập ngay

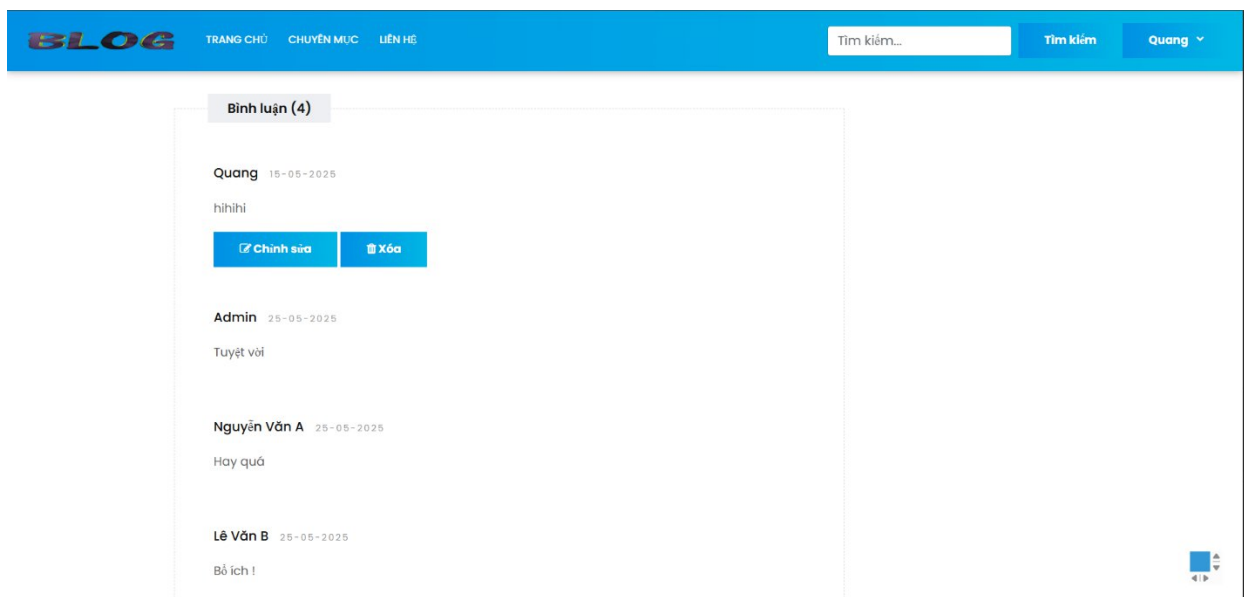
Hình 3. Giao diện đăng ký



Hình 4. Giao diện khi đăng nhập thành công



Hình 5. Giao diện khi xem 1 bài viết



Hình 6. Phần mọi người đã bình luận

BLOG

TRANG CHỦ

CHUYÊN MỤC

LIÊN HỆ

Tìm kiếm...

Tìm kiếm

Quang ▾

Chỉnh sửa bình luận

Nội dung bình luận

bihibi

Lưu thay đổi

Blog cá nhân - Nơi chia sẻ những câu chuyện và suy nghĩ của bạn.

f

t

@

G+

in

Chuyên mục phổ biến

Nấu Ăn (21)

Thế Thao (15)

Du Lịch (31)

Thông tin

Giới Thiệu

Quảng Cáo

Góp Ý

Hình 7. Phần chỉnh sửa bình luận

BLOG

TRANG CHỦ

CHUYÊN MỤC

LIÊN HỆ

Tìm kiếm...

Tìm kiếm

Quang ▾

Để lại bình luận

Viết bình luận

Gửi bình luận

Blog cá nhân - Nơi chia sẻ những câu chuyện và suy nghĩ của bạn.

f

t

@

G+

in

Chuyên mục phổ biến

Nấu Ăn (21)

Thế Thao (15)

Du Lịch (31)

Thông tin

Giới Thiệu

Quảng Cáo

Góp Ý

Hình 8. Phần bình luận bài viết

BLOG

TRANG CHỦ | CHUYÊN MỤC | LIÊN HỆ

Tìm kiếm...

Tìm kiếm

Quang ▾

Thông tin cá nhân

Tên hiển thị:

Quang

Email:

quang123@gmail.com

Giới thiệu:

Mật khẩu hiện tại:

Mật khẩu mới:

Xác nhận mật khẩu mới:

Lưu thay đổi

Bài viết của tôi

Tìm hiểu công nghệ màn hình Pixelsense của Microsoft

Công nghệ màn hình Pixelsense là gì?

19/05/2020 [Chỉnh sửa](#) [Xóa](#)

Chạy bộ

Hãy chạy bộ

15/05/2020 [Chỉnh sửa](#) [Xóa](#)

Mẹo nấu thịt bò không dai

Bí quyết chi tiết từ sư chế biến nấu

15/05/2020 [Chỉnh sửa](#) [Xóa](#)

Du lịch Việt Nam 2020: Dấu ấn tư năng cuộc vượt bão

Du lịch Việt Nam

19/05/2020 [Chỉnh sửa](#) [Xóa](#)

BLOG

TRANG CHỦCHUYỂN MẠCCLIÊN HỆ

Tìm kiếm...Tìm kiếmQuang

Tạo bài viết mới

Tiêu đề bài viết

Nhập tiêu đề bài viết...

Mô tả ngắn

Nhập mô tả ngắn cho bài viết...

Nội dung bài viết

Nhập nội dung bài viết...

Chuyển mục

Chọn chuyển mục

Hình ảnh đại diện

Chọn tệpKhông có tệp nào được chọn
Hỗ trợ định dạng: JPG, PNG, GIF. Kích thước tối đa: 2MB

Tạo bài viết

BLOG

TRANG CHỦ

CHUYÊN MỤC

LIÊN HỆ

Tìm kiếm...

Tìm kiếm

Nguyễn Văn A

Chỉnh sửa bài viết

Chỉnh sửa bài viết: Những kiến thức nhạc lý cơ bản

Tiêu đề bài viết

Những kiến thức nhạc lý cơ bản

Mô tả ngắn

Hệ thống âm nhạc cũng có những nguyên tắc riêng và để bắt đầu chuyển hành trình đến với âm nhạc ta cần làm quen với "bảng chữ cái" và cách "đánh vần"

Nội dung bài viết

Âm nhạc là một thế giới thú vị mà đa số mọi người đều cảm thấy bị thu hút hoặc muốn đắm chìm trong nó. Âm nhạc giúp ta thư giãn, xua lạnh tâm hồn. Mỗi khoảnh khắc trong cuộc đời, khi vui hay khi buồn ta đều tìm đến âm nhạc. Khi chúng ta thực sự muốn tìm hiểu nó, muốn tự tạo những giai điệu của chính mình, âm nhạc sẽ giúp ta khám phá bản thân. Không chỉ là một môn giải trí đơn thuần, để có những bản nhạc ta nghe hằng ngày, những tác phẩm âm nhạc để đời, người học và làm nhạc phải trải qua một quá trình học tập và làm việc chăm chỉ. Hệ thống âm nhạc cũng có những nguyên tắc riêng và để bắt đầu chuyển hành trình đến với âm nhạc ta cần làm quen với "bảng chữ cái" và cách "đánh vần". Chúng được gọi là lý thuyết âm nhạc hoặc nhạc lý.

1. Nhạc lý là gì?

Giống như mặt đũa tre bắt đầu học chữ, người học nhạc cũng có những bộ "alphabet" của riêng mình, thay vì A B C D E F G, bạn sẽ tập làm quen với Đô, Rê, Mi, Fa, Sol, La, Si. Những دانش که trên nĩa được thay bằng những دانش که không nhạc. Bả sẽ dần những دانش که thành một từ, một câu hoặc chỉnh và

Chuyên mục

Hình 11. Chỉnh sửa bài viết 1

BLOG

TRANG CHỦ

CHUYÊN MỤC

LIÊN HỆ

Tìm kiếm...

Tìm kiếm

Nguyễn Văn A

tác phẩm âm nhạc để đời, người học và làm nhạc phải trải qua một quá trình học tập và làm việc chăm chỉ. Hệ thống âm nhạc cũng có những nguyên tắc riêng và để bắt đầu chuyển hành trình đến với âm nhạc ta cần làm quen với "bảng chữ cái" và cách "đánh vần". Chúng được gọi là lý thuyết âm nhạc hoặc nhạc lý.

1. Nhạc lý là gì?

Giống như mặt đũa tre bắt đầu học chữ, người học nhạc cũng có những bộ "alphabet" của riêng mình, thay vì A B C D E F G, bạn sẽ tập làm quen với Đô, Rê, Mi, Fa, Sol, La, Si. Những دانش که trên nĩa được thay bằng những دانش که không nhạc. Bả sẽ dần những دانش که thành một từ, một câu hoặc chỉnh và

Chuyên mục

Âm nhạc

Hình ảnh đại diện

Chọn tệp

Không có tệp nào được chọn

Hỗ trợ định dạng: JPG, PNG, GIF. Kích thước tối đa: 2MB

Hình ảnh hiện tại

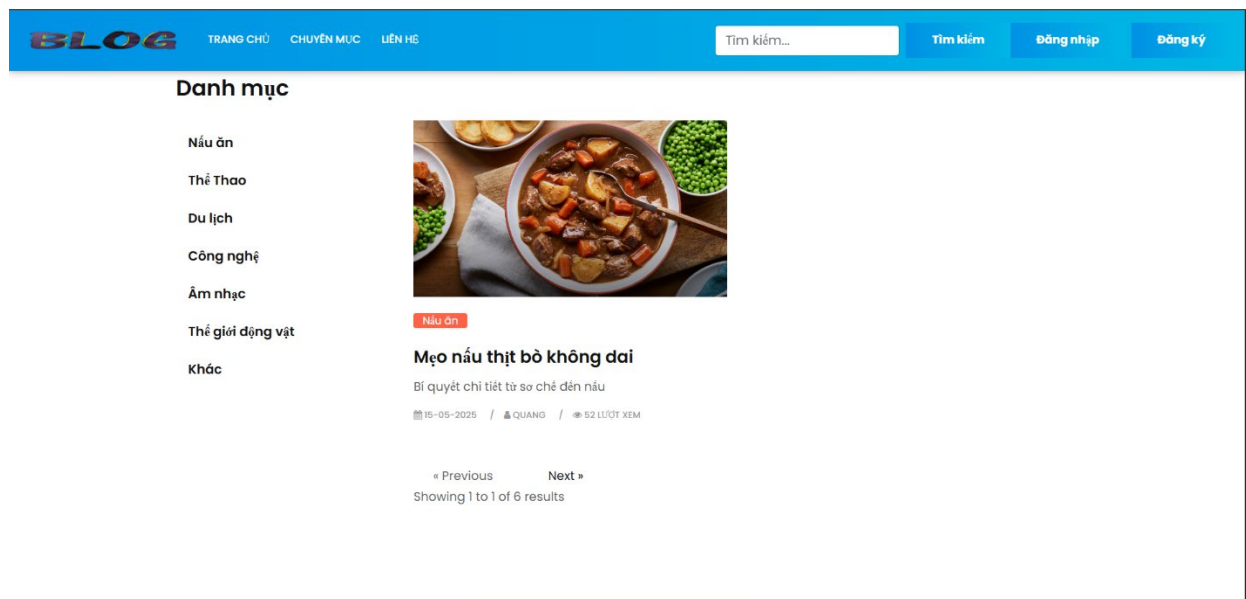
Lưu thay đổi

Xóa bài viết

Hình 12. Chỉnh sửa bài viết 2

Nhóm 24

35



Hình 13. Chuyên mục (phân loại danh mục)

The screenshot shows a contact form on a blog. The header is the same as in the previous image. The main content area is titled "Về chúng tôi" (About us). Below the title, there is a paragraph: "Blog Cá Nhân là nơi chia sẻ những câu chuyện, suy nghĩ và trải nghiệm từ các tác giả độc lập trên thế giới."

Below this, there is a section titled "Chúng tôi hỗ trợ gì?" (What we support?). Below this, there is a paragraph: "Chúng tôi giúp bạn kết nối với cộng đồng người viết blog, chia sẻ kiến thức và trải nghiệm."

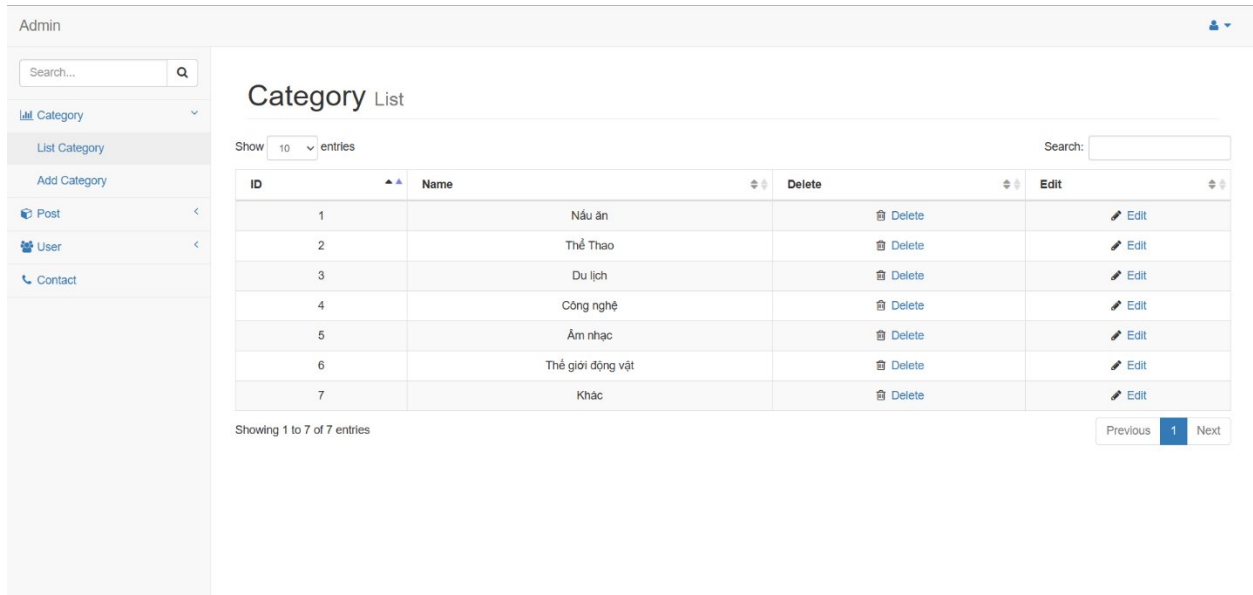
Below this, there is a section titled "Câu hỏi trước khi liên hệ" (Questions before contacting). Below this, there is a paragraph: "Vui lòng đọc kỹ các điều khoản và chính sách của chúng tôi trước khi liên hệ."

To the right of the text, there is a form with several input fields:

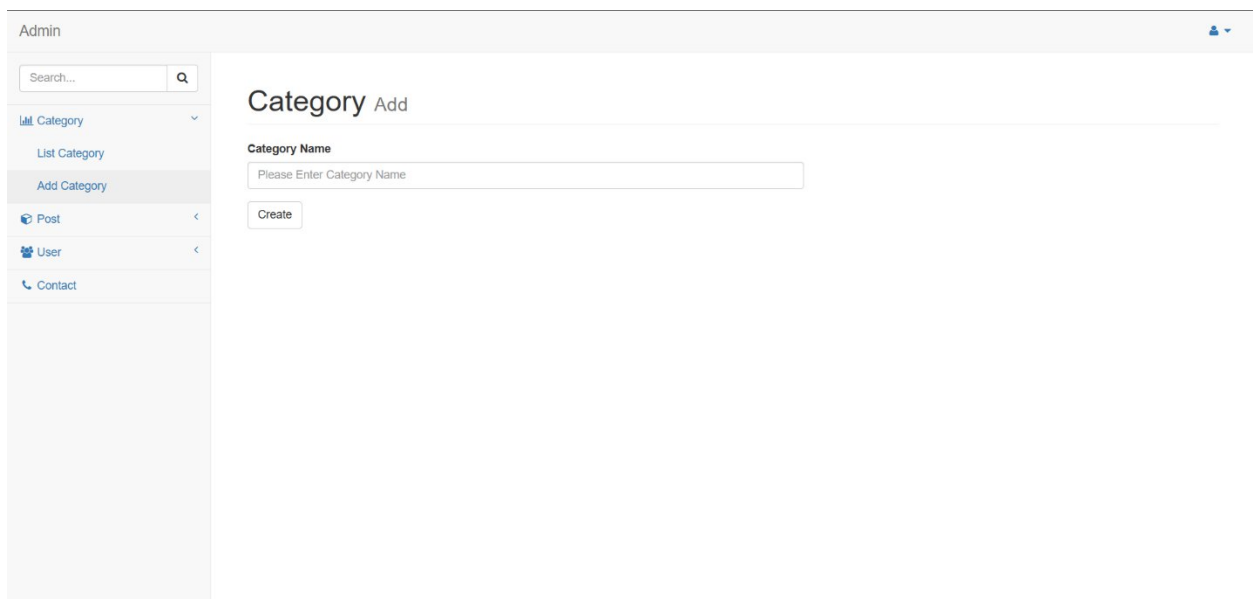
- Tên của bạn (Your name): Nhập tên của bạn
- Địa chỉ (Address): Nhập địa chỉ
- Số điện thoại (Phone number): Nhập số điện thoại
- Chủ đề (Subject): Nhập chủ đề
- Nội dung (Content): Nhập nội dung

At the bottom of the form, there is a blue button with a white envelope icon and the text "Gửi liên hệ" (Send contact).

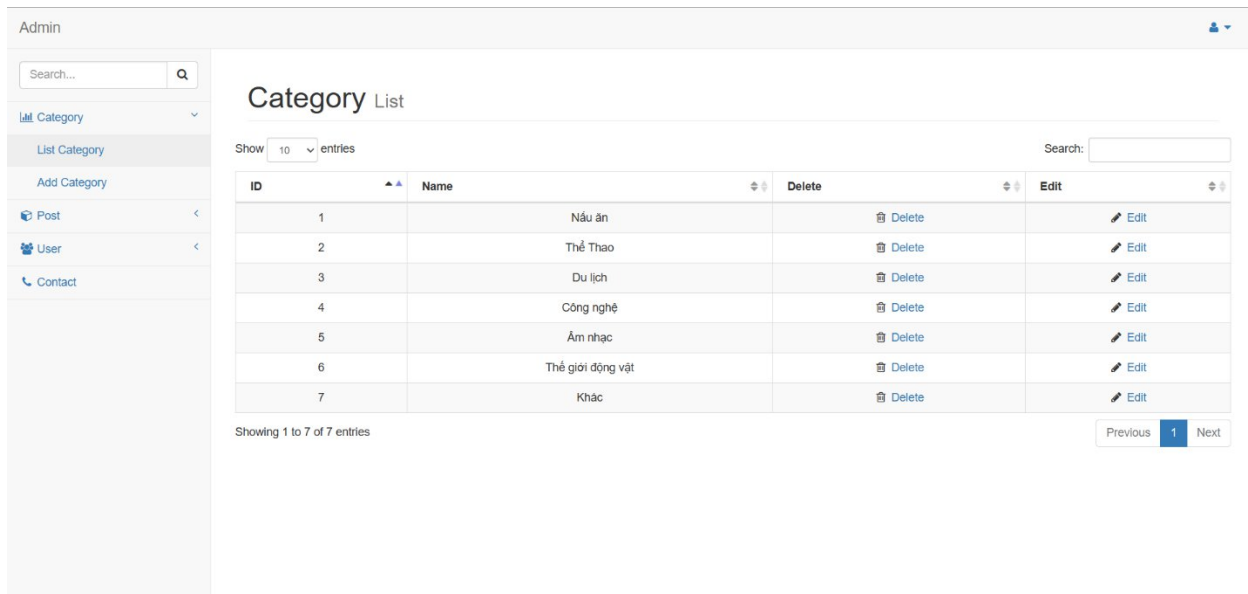
Hình 14. Liên hệ



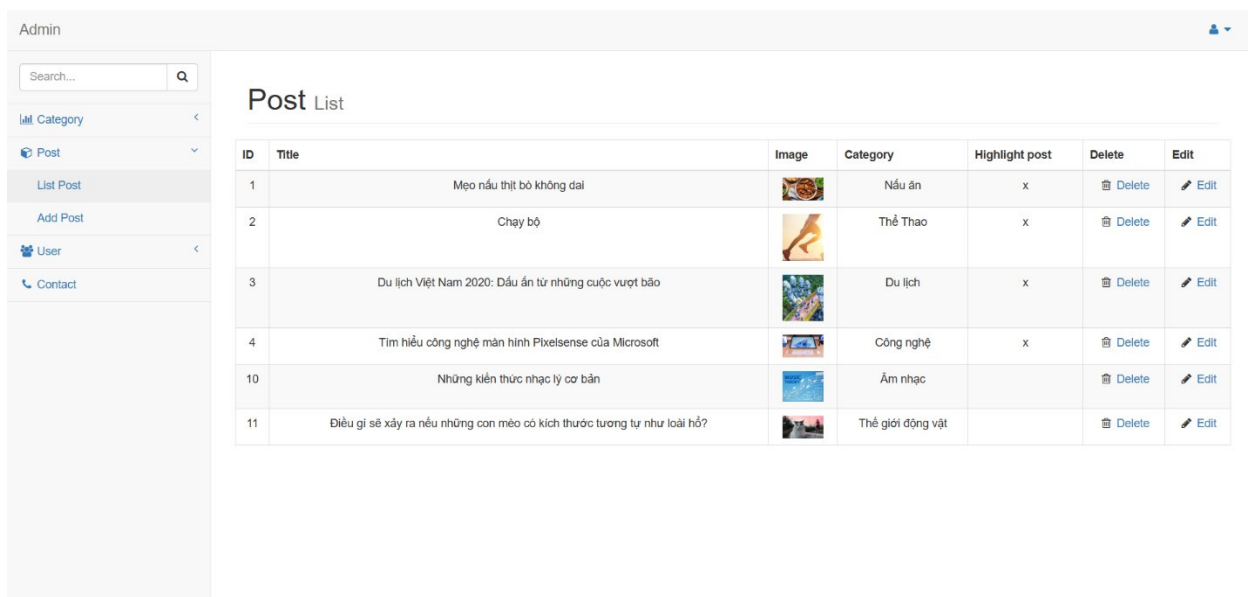
Hình 15. Giao diện khi đăng nhập thành công Admin



Hình 16. Thêm chuyên đề (danh mục) của Admin



Hình 17. Chỉnh sửa, xóa danh mục của Admin



Hình 18. Những bài viết đã tạo

Admin

Search...

Q

Category

Post

User

List User

Add User

Contact

User Add

Name

Please Enter Name

Email

Please Enter Email

Password

Please Enter Password

Confirm

Please Confirm Password

Role

☒ User

☐ Admin

Add

Hình 21. Thêm người dùng và phân quyền

Admin

Search...

Q

Category

Post

User

Contact

Contact List

| ID | Name | Address | Phone | Subject | Message | Delete |
|----|--------------|------------------------|------------|---------|--------------------|--------|
| 1 | Nguyễn Văn A | Ngõ 1153 Đường Lê Duẩn | 0123456789 | Chạy bộ | Cần chỉnh sửa thêm | Delete |

Hình 22. Liên hệ (Admin)

5.2. Đánh giá hiệu quả hệ thống

a. Kết quả đạt được

Sau quá trình phân tích, thiết kế và phát triển, hệ thống website blog cá nhân đã đạt được các chức năng cơ bản và nâng cao như sau:

- Giao diện người dùng trực quan, thân thiện.
- Chức năng đăng ký, đăng nhập, đăng xuất sử dụng hệ thống xác thực của Laravel.

- Người dùng có thể viết bài, chỉnh sửa, xóa bài viết.
- Mỗi bài viết có thể được phân loại theo danh mục.
- Tích hợp chức năng bình luận.
- Hệ thống ghi nhận lượt xem bài viết để thống kê bài viết phổ biến.
- Trang quản trị viên có thể quản lý bài viết, người dùng, bình luận, và xem báo cáo thống kê.
- Hệ thống hoạt động ổn định trên môi trường cục bộ với SQL Server.

b. Ưu điểm

- Sử dụng Laravel Framework giúp rút ngắn thời gian phát triển nhờ vào các tính năng sẵn có.
- Tách biệt rõ ràng giữa giao diện, xử lý và dữ liệu theo mô hình MVC, dễ bảo trì và mở rộng.
- Hệ thống có khả năng mở rộng với nhiều chức năng như chia sẻ bài viết, bookmark, theo dõi người dùng,...

c. Hạn chế

- Website mới chỉ được triển khai và kiểm thử trên môi trường cục bộ (localhost), chưa triển khai lên môi trường thực tế (hosting/VPS).
- Chưa tích hợp các chức năng nâng cao như: gửi email khi có bình luận mới, quản lý ảnh đa phương tiện nâng cao, hoặc tìm kiếm toàn văn.
- Chức năng phân quyền người dùng còn đơn giản (chỉ có người dùng thường và admin).

d. Kiến nghị cải tiến

Để hệ thống hoàn thiện hơn và phục vụ người dùng tốt hơn, có thể cải tiến các điểm sau:

- Triển khai hệ thống trên một hosting thực tế hoặc VPS để phục vụ người dùng thật.
- Bổ sung chức năng tìm kiếm nâng cao, có thể lọc theo nhiều tiêu chí.
- Phát triển thêm API mở, phục vụ ứng dụng di động trong tương lai.

Chương 6. Tài liệu tham khảo

- Visual Studio Code - <https://code.visualstudio.com/>
- Laravel Documentation - <https://laravel.com/docs>
- PHP Manual - <https://www.php.net/manual/en/>
- MySQL - <https://www.mysql.com/>
- Composer Dependency Manager - <https://getcomposer.org/>
- DBeaver Community Edition – <https://dbeaver.io/>