

**BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**ĐỒ ÁN ĐẠI HỌC HỆ THỐNG THÔNG TIN**

**XÂY DỰNG WEBSITE TIN TỨC THỂ THAO SỬ DỤNG PHP**

**CBHD: TS. Lê Thị Anh**

**Sinh viên: Vũ Đình Mạnh**

**Mã số sinh viên: 2021604240**

**Hà Nội – Năm 2024**

**HỌ VÀ TÊN SINH VIÊN: VŨ ĐÌNH MẠNH**

**NGÀNH: HỆ THỐNG THÔNG TIN**

## MỤC LỤC

LỜI MỞ ĐẦU .....	i
DANH MỤC HÌNH VẼ.....	iii
DANH MỤC BẢNG BIỂU .....	v
CHƯƠNG 1. TỔNG QUAN VỀ CÔNG NGHỆ VÀ CƠ SỞ LÝ THUYẾT .....	1
1.1. Ngôn ngữ lập trình PHP .....	1
1.1.1. Giới thiệu về PHP .....	1
1.1.2. Ưu điểm của PHP.....	1
1.1.3. Nhược điểm của PHP.....	2
1.2. Laravel Framework .....	2
1.2.1. Giới thiệu về Laravel .....	2
1.1.2. Ưu điểm của Laravel.....	3
1.1.3. Nhược điểm của Laravel.....	3
1.2.4. Sử dụng Laravel trong đồ án.....	4
1.3. Blade Template trong Laravel .....	6
1.3.1. Giới thiệu về Blade Template .....	6
1.3.2. Sử dụng Blade Template trong đồ án .....	6
1.4. Ngôn ngữ lập trình Javascript .....	7
1.4.1. Giới thiệu về Javascript .....	7
1.4.2. Ưu điểm của javascript .....	7
1.4.3. Nhược điểm của javascript .....	7
1.4.4. Sử dụng javascript trong đồ án .....	8
1.5. Hệ quản trị cơ sở dữ liệu MySQL.....	8
1.5.1. Giới thiệu về MySQL .....	8
1.5.2. Ưu điểm của MySQL.....	9
1.5.3. Nhược điểm của MySQL.....	9
1.5.4. Sử dụng MySQL trong đồ án.....	10
1.6. Kết luận chương 1 .....	10

CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU .....	11
2.1. Xác định yêu cầu chính của hệ thống .....	11
2.1.1. Phân tích các yêu cầu.....	11
2.2.2. Các yêu cầu phi chức năng .....	11
2.2. Xác định biểu đồ use case .....	12
2.2.1. Các tác nhân .....	12
2.2.2. Các use case của hệ thống.....	13
2.2.3. Biểu đồ use case.....	13
2.2.4. Mô tả chi tiết các use case .....	14
2.2.4.1. Mô tả use case Đăng nhập (phía quản trị) .....	14
2.2.4.2. Mô tả use case Quản lý nhóm thành viên.....	15
2.2.4.3. Mô tả use case Quản lý thành viên .....	18
2.2.4.4. Mô tả use case Quản lý bài viết.....	23
2.2.4.5. Mô tả use case Quản lý nhóm bài viết.....	30
2.2.4.6. Mô tả use case Đăng nhập (phía người dùng) .....	35
2.2.4.7. Mô tả use case Xem bài viết theo danh mục .....	37
2.2.4.8. Mô tả use case Xem chi tiết bài viết.....	39
2.2.4.9. Mô tả use case Bình luận về bài viết .....	40
2.2.4.10. Mô tả use case Lưu bài viết .....	42
2.3. Xây dựng cơ sở dữ liệu .....	44
2.3.1. Biểu đồ Entity Relationship Diagram.....	44
2.3.2. Mô tả các bảng trong cơ sở dữ liệu.....	45
2.4. Thiết kế màn hình.....	56
2.4.1. Màn hình đăng nhập (phía quản trị).....	56
2.4.2. Màn hình Trang chủ.....	56
2.4.3. Màn hình Xem bài viết theo danh mục.....	57
2.4.4. Màn hình Xem chi tiết bài viết .....	58
2.4.5. Màn hình Quản lý bài viết .....	59
2.4.6. Màn hình Thêm mới nhóm bài viết .....	59

2.4.7. Màn hình Thêm mới thành viên.....	60
2.4.8. Màn hình xoá thành viên.....	61
2.5. Kết luận chương 2 .....	61
CHƯƠNG 3. KẾT QUẢ THỰC HIỆN VÀ KIỂM THỬ.....	62
3.1. Kết quả thực hiện .....	62
3.1.1. Giao diện đăng nhập (phía người dùng) .....	62
3.1.2. Giao diện trang chủ.....	62
3.1.3. Giao diện xem bài viết theo danh mục .....	64
3.1.4. Giao diện xem chi tiết bài viết.....	64
3.1.5. Giao diện thông tin cá nhân .....	65
3.1.6. Giao diện đăng nhập (phía quản trị) .....	66
3.1.7. Giao diện Dashboard.....	66
3.1.8. Giao diện quản lý nhóm bài viết.....	67
3.1.9. Giao diện quản lý bài viết.....	67
3.1.10. Giao diện quản lý nhóm thành viên .....	68
3.1.11. Giao diện quản lý thành viên .....	69
3.2. Kiểm thử hệ thống.....	70
3.2.1. Kế hoạch kiểm thử.....	70
3.2.2. Kiểm thử chức năng.....	70
3.2.3. Kết quả kiểm thử.....	74
3.3. Kết luận chương 3 .....	74
KẾT LUẬN .....	75
TÀI LIỆU THAM KHẢO.....	76

# LỜI MỞ ĐẦU

## 1. Lý do chọn đề tài

Hiện nay, công nghệ thông tin đang phát triển mạnh mẽ và hiện đại, kéo theo sự ra đời của nhiều nền tảng số đa dạng, giúp người dùng dễ dàng tiếp cận thông tin. Trong đó, các trang web đọc tin tức online ngày càng trở nên phổ biến và được ứng dụng rộng rãi nhờ sự tiện lợi, nhanh chóng và đa dạng trong nội dung cung cấp.

Xuất phát từ mong muốn củng cố và nâng cao kiến thức về lập trình, đồng thời áp dụng những kiến thức đã học vào thực tiễn, em đã quyết định chọn đề tài “Xây dựng website tin tức thể thao sử dụng PHP” làm đồ án tốt nghiệp. Đây không chỉ là cơ hội để em rèn luyện kỹ năng kỹ thuật, mà còn giúp em hiểu rõ hơn về quy trình phát triển một sản phẩm công nghệ hoàn chỉnh, từ giai đoạn lên ý tưởng, phân tích, thiết kế, đến thực hiện và triển khai.

Website tin tức thể thao là một nền tảng hữu ích, không chỉ cung cấp thông tin nhanh chóng về các sự kiện thể thao, kết quả và tin tức bên lề mà còn là nơi kết nối cộng đồng yêu thích thể thao. Với mục tiêu xây dựng một sản phẩm có giao diện thân thiện, dễ sử dụng, em hy vọng website này sẽ đáp ứng tốt nhu cầu tra cứu và giải trí của người dùng.

Để hoàn thành được đồ án tốt nghiệp này, em xin gửi lời cảm ơn chân thành đến các thầy cô trong Khoa Công nghệ thông tin, Trường Đại học Công Nghiệp Hà Nội, đặc biệt là cô giáo hướng dẫn đề tài – Tiến sĩ Lê Thị Anh, người đã tận tâm giúp đỡ, hướng dẫn và chỉ bảo em trong suốt quá trình thực hiện đồ án. Sự tận tình và chuyên môn sâu sắc của cô đã giúp em vượt qua những khó khăn, hoàn thiện kiến thức và kỹ năng để hoàn thành tốt đồ án này.

Em xin chân thành cảm ơn!

**Vũ Đình Mạnh**

## **2. Mục tiêu của đề tài**

Xây dựng website tin tức thể thao hoàn thiện với các module chức năng. Phát triển hệ thống quản lý nội dung: Tạo chức năng để quản trị viên có thể đăng, sửa, xóa và quản lý các bài viết liên quan đến tin tức thể thao. Hiểu cách thiết kế và triển khai hệ thống web hoàn chỉnh từ giao diện đến backend.

## **3. Nội dung nghiên cứu**

- Khảo sát, phân tích nghiệp vụ
- Nested set model để quản lý menu
- Nghiên cứu tích hợp đăng nhập Google
- Lập trình Backend với Laravel
- Lập trình giao diện với Blade Templates

## **4. Bố cục đề tài**

Ngoài các phần Mở đầu, Kết luận và Tài liệu tham khảo, báo cáo đồ án được bố cục thành ba chương chính sau.

- Chương 1: Tổng quan về công nghệ và cơ sở lý thuyết.
- Chương 2: Phân tích thiết kế hệ thống và thiết kế cơ sở dữ liệu.
- Chương 3: Kết quả thực hiện và kiểm thử.

## DANH MỤC HÌNH VẼ

Hình 2.1: Các use case phía người dùng.....	13
Hình 2.2: Các use phía quản trị viên.....	13
Hình 2.3: Biểu đồ trình tự use case Đăng nhập (phía quản trị) .....	15
Hình 2.4: Biểu đồ trình tự use case Quản lý nhóm thành viên .....	18
Hình 2.5: Biểu đồ trình tự use case Quản lý thành viên .....	23
Hình 2.6: Biểu đồ trình tự use case Quản lý bài viết .....	30
Hình 2.7: Biểu đồ trình tự use case Quản lý nhóm bài viết.....	35
Hình 2.8: Biểu đồ trình tự use case Đăng nhập (phía quản trị) .....	37
Hình 2.9: Biểu đồ trình tự use case Xem bài viết theo danh mục.....	39
Hình 2.10: Biểu đồ trình tự use case Xem chi tiết bài viết .....	40
Hình 2.11: Biểu đồ trình tự use case Bình luận về bài viết .....	42
Hình 2.12: Biểu đồ trình tự use case Lưu bài viết .....	44
Hình 2.13: Biểu đồ ERD .....	45
Hình 2.14: Thiết kế màn hình Đăng nhập (phía quản trị).....	56
Hình 2.15: Thiết kế màn hình Trang chủ.....	57
Hình 2.16: Thiết kế màn hình Xem bài viết theo danh mục .....	58
Hình 2.17: Thiết kế màn hình Xem chi tiết bài viết.....	58
Hình 2.18: Thiết kế màn hình Quản lý bài viết.....	59
Hình 2.19: Thiết kế màn hình Thêm mới nhóm bài viết.....	60
Hình 2.20: Thiết kế màn hình Thêm mới thành viên.....	60
Hình 2.21: Thiết kế màn hình Xóa thành viên.....	61
Hình 3.22: Giao diện đăng nhập (phía người dùng) .....	62
Hình 3.23: Giao diện trang chủ .....	64

Hình 3.24: Giao diện xem bài viết theo danh mục.....	64
Hình 3.25: Giao diện xem chi tiết bài viết .....	65
Hình 3.26: Giao diện thông tin cá nhân .....	66
Hình 3.27: Giao diện đăng nhập (phía quản trị) .....	66
Hình 3.28: Giao diện Dashboad .....	67
Hình 3.29: Giao diện quản lý nhóm bài viết .....	67
Hình 3.30: Giao diện quản lý bài viết .....	68
Hình 3.31: Giao diện quản lý nhóm thành viên .....	69
Hình 3.32: Giao diện quản lý thành viên .....	69



## DANH MỤC BẢNG BIỂU

Bảng 2.1: Các use case của hệ thống .....	12
Bảng 2.2: Chi tiết bảng USERS .....	45
Bảng 2.3: Chi tiết bảng PROVINCES .....	47
Bảng 2.4: Chi tiết bảng DISTRICTS .....	48
Bảng 2.5: Chi tiết bảng WARDS .....	48
Bảng 2.6: Chi tiết bảng USER_CATALOGUE.....	49
Bảng 2.7: Chi tiết bảng USER_CATALOGUE_PERMISSION.....	50
Bảng 2.8: Chi tiết bảng PERMISSIONS .....	50
Bảng 2.9: Chi tiết bảng SAVE_POSTS .....	51
Bảng 2.10: Chi tiết bảng COMMENTS.....	51
Bảng 2.11: Chi tiết bảng POSTS.....	52
Bảng 2.12: Chi tiết bảng POST_LANGGUAGE .....	53
Bảng 2.13: Chi tiết bảng POST_CATALOGUES .....	54
Bảng 2.14: Chi tiết bảng POST_CATALOGUES_POST .....	54
Bảng 2.15: Chi tiết bảng POST_CATALOGUE_LANGUAGE.....	55
Bảng 2.16: Chi tiết bảng LANGUAGES .....	55
Bảng 3.17: Kiểm thử chức năng.....	70

# CHƯƠNG 1. TỔNG QUAN VỀ CÔNG NGHỆ VÀ CƠ SỞ LÝ THUYẾT

## 1.1. Ngôn ngữ lập trình PHP

### 1.1.1. Giới thiệu về PHP

PHP: Hypertext Preprocessor, thường được viết tắt thành PHP là một ngôn ngữ lập trình kịch bản hay một loại mã lệnh chủ yếu được dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở, dùng cho mục đích tổng quát. Nó rất thích hợp với web và có thể dễ dàng nhúng vào trang HTML. Do được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một ngôn ngữ lập trình web phổ biến nhất thế giới.

Ngôn ngữ PHP thông thường sẽ phù hợp với việc lập trình website bởi nó có thể dễ dàng kết nối với các website khác có sử dụng HTML để chạy trên các trình duyệt web. Vì vậy, đây là ngôn ngữ lập trình được người dùng đánh giá là khá dễ đọc. Ngôn ngữ PHP cũng trở thành ngôn ngữ web dev phổ biến mà các PHP Developer phải học trước khi bắt đầu vào nghề.

### 1.1.2. Ưu điểm của PHP

- **Đơn giản, linh động:** Từ thiết kế của PHP, cộng với việc PHP là ngôn ngữ Script và cú pháp khá thoải mái nên nó rất linh động. Cú pháp PHP cũng rất dễ học nên rất nhiều người biết PHP. Với một người chưa biết gì về lập trình chỉ cần một khóa học vài ba tháng có thể bắt đầu với PHP và thậm chí có thể bắt đầu... kiếm tiền được rồi. Chính vì sự dễ dàng đó nên số lượng developer rất đông và hung hãn.
- **Support bởi cộng đồng lớn:** PHP sở hữu một trong những cộng đồng developer lớn nhất. Số lượng job về PHP cũng luôn thuộc hàng top. Chính vì có cộng đồng rất lớn như vậy nên hầu như vấn đề kỹ thuật nào bạn gặp phải cũng có thể có người hỗ trợ ngay lập tức. Nếu bạn đã từng sử dụng

những ngôn ngữ lập trình ít phổ biến thì sẽ thấy tầm quan trọng của cộng đồng lớn đến thế nào.

- **Có rất nhiều Framework, thư viện:** Cùng với việc sở hữu cộng đồng lớn, PHP cũng sở hữu vô số bộ thư viện, extension, rất nhiều Framework. Do vậy PHP có thể giải quyết rất nhiều bài toán khác nhau. Hầu như nói đến vấn đề gì cũng có thể có những thư viện liên quan để PHP. Do vậy đôi khi người ta còn tưởng PHP là lời giải cho mọi vấn đề.

### 1.1.3. Nhược điểm của PHP

- **Không chia sẻ tài nguyên:** Vấn đề đầu tiên cũng là vấn đề hạn chế lớn nhất của PHP chính là việc không chia sẻ tài nguyên giữa các tiến trình. Việc đóng khung các tiến trình giúp ích rất lớn cho việc PHP không phải đối mặt với những vấn đề như: quản lý bộ nhớ (chả mấy khi PHP Developer quan tâm đến vấn đề này), crash hệ thống (một hai tiến trình chết thì chả ảnh hưởng gì đến hệ thống). Tuy nhiên việc này lại dẫn đến rất nhiều hạn chế khác. Đầu tiên là việc không chia sẻ tài nguyên khiến cho tài nguyên sử dụng (RAM, CPU, I/O connection) tăng lên rất nhanh.
- **Khả năng mở rộng hạn chế:** PHP không lý tưởng để xây dựng các hệ thống yêu cầu khả năng mở rộng lớn, ví dụ: các dịch vụ microservices hoặc các hệ thống phân tán lớn.

## 1.2. Laravel Framework

### 1.2.1. Giới thiệu về Laravel

Laravel là một framework PHP mã nguồn mở miễn phí được phát hành lần đầu tiên vào năm 2011. Taylor Otwell đã tạo ra Laravel nhằm giúp các Developer xây dựng các ứng dụng web bằng cách sử dụng mẫu kiến trúc Model-View-Controller (MVC).

Laravel là một framework back-end, được viết bằng ngôn ngữ lập trình PHP. Laravel cung cấp các công cụ và cấu trúc để xây dựng phần logic của ứng dụng web ví dụ như: Xử lý yêu cầu và phản hồi HTTP, kết nối và thao tác với cơ

sở dữ liệu, xác thực và phân quyền người dùng, quản lý cache, gửi email và nhiều tính năng khác.

Laravel không phải là framework front-end. Laravel không cung cấp các công cụ để xây dựng giao diện người dùng (UI) của ứng dụng web, chẳng hạn như HTML, CSS và JavaScript. Tuy nhiên, Laravel có thể được tích hợp với các framework front-end khác để tạo ra các ứng dụng web hoàn chỉnh. Một số framework front-end phổ biến được sử dụng với Laravel có thể kể đến như: Vue.js, React, Angular...

### ***1.1.2. Ưu điểm của Laravel***

- **Cú pháp rõ ràng và dễ đọc:** Laravel sử dụng cú pháp rõ ràng và dễ đọc, giúp người phát triển tương tác và hiểu mã nguồn dễ dàng.
- **Mô hình MVC:** Sử dụng mô hình MVC giúp tách biệt code logic, dữ liệu và giao diện người dùng, từ đó dễ dàng bảo trì và mở rộng ứng dụng.
- **Eloquent ORM:** Laravel cung cấp Eloquent ORM để tương tác với cơ sở dữ liệu thông qua mô hình và query builder, giúp việc làm việc với cơ sở dữ liệu trở nên dễ dàng và linh hoạt.
- **Hệ thống template Blade:** Blade là một hệ thống template mạnh mẽ cho phép người phát triển xây dựng giao diện người dùng một cách đơn giản, linh hoạt và mạnh mẽ.
- **Tài liệu và cộng đồng phong phú:** Laravel cung cấp tài liệu hướng dẫn đầy đủ và có cộng đồng phong phú, giúp người phát triển dễ dàng tìm kiếm sự trợ giúp và hỗ trợ khi cần thiết.

### ***1.1.3. Nhược điểm của Laravel***

- **Kích thước lớn:** Laravel là một framework khá lớn, điều này có thể gây ra khó khăn trong việc triển khai và sử dụng trên các dịch vụ hosting có giới hạn tài nguyên.

- **Tính năng quá phức tạp:** Laravel cung cấp rất nhiều tính năng và package hỗ trợ, tuy nhiên, việc sử dụng đồng thời nhiều tính năng này có thể làm cho ứng dụng của bạn trở nên quá phức tạp và khó bảo trì.
- **Thư viện không hoàn toàn tương thích với các phiên bản mới nhất của PHP:** Một số phiên bản mới nhất của PHP có thể không hoàn toàn tương thích với các thư viện của Laravel, điều này có thể gây ra khó khăn trong việc phát triển và bảo trì ứng dụng.

#### ***1.2.4. Sử dụng Laravel trong đồ án***

Thư mục app: Chứa mã nguồn chính của ứng dụng. Đây là nơi bạn định nghĩa các lớp (classes) chính như Controllers, Models, Middleware, Requests, Providers:

- app/Http/Controllers/: Chứa các controller, nơi xử lý logic cho các request.
- app/Middleware/: Chứa các lớp middleware để lọc các HTTP requests.
- app/Requests/: Chứa các form request dùng để validate dữ liệu đầu vào.
- app/Models/: Chứa các model tương tác với cơ sở dữ liệu.
- app/Providers/: Providers là nơi chứa các Service Providers của ứng dụng, đóng vai trò quan trọng trong việc khởi tạo các thành phần chính của ứng dụng Laravel.
- app/Services/: Services thường được sử dụng để chứa các lớp xử lý nghiệp vụ phức tạp của ứng dụng. Giúp tách biệt logic nghiệp vụ ra khỏi các lớp Controller, giúp mã nguồn trở nên rõ ràng, dễ bảo trì và mở rộng.
- app/Repositories/: Repositories sử dụng Repository Pattern, giúp trừu tượng hóa các thao tác với cơ sở dữ liệu khỏi các lớp khác như Controller. Việc sử dụng Repositories giúp dễ dàng thay đổi lớp truy xuất dữ liệu mà không ảnh hưởng đến logic nghiệp vụ.

Thư mục bootstrap chứa những file khởi động của framework và những file cấu hình autoloading. Ngoài ra nó còn có thư mục cache chứa những file mà framework sinh ra để tăng hiệu năng như route và services cache files.

Thư mục config, đúng như cái tên của nó, chứa tất cả những file cấu hình.

Thư mục database chứa những file database migration và seeds:

- database/factories/: Chứa các file factory để tạo dữ liệu giả phục vụ cho quá trình thử nghiệm.
- database/migrations/: Chứa các file migration để tạo hoặc chỉnh sửa bảng trong cơ sở dữ liệu.
- database/seeder/: Chứa các file seeder để khởi tạo dữ liệu mẫu cho bảng.

Thư mục public chứa file index.php, đó là điểm mấu chốt cho tất cả các request vào ứng dụng. Ngoài ra nó còn chứa một số tài nguyên như ảnh, JavaScript, và CSS.

Thư mục resources chứa những view và raw, những tài nguyên chưa compiled như LESS, SASS, hoặc JavaScript. Nó còn chứa tất cả các file ngôn ngữ trong ứng dụng của bạn:

- resources/views/: Chứa các file Blade template (.blade.php) cho giao diện người dùng.
- resources/js/: Chứa các file JavaScript tùy chỉnh.
- resources/css/: Chứa các file CSS tùy chỉnh.
- resources/lang/: Chứa các file ngôn ngữ cho localization.

Thư mục routes chứa tất cả các định nghĩa route trong ứng dụng của bạn. Mặc định, có ba file route được thêm vào với Laravel: web.php, api.php, và console.php:

- api.php: Định nghĩa các route cho API.
- web.php: Định nghĩa các route cho ứng dụng web.
- console.php: Định nghĩa các lệnh Artisan tùy chỉnh.
- channels.php: Định nghĩa các kênh cho hệ thống thông báo (Broadcast).

Thư mục vendor/: Chứa các gói (packages) bên thứ ba được cài đặt qua Composer.

File env: Đây là File cấu hình môi trường, chứa các biến môi trường như thông tin kết nối database, mail, API keys.

### **1.3. Blade Template trong Laravel**

#### ***1.3.1. Giới thiệu về Blade Template***

Blade là một templating engine đơn giản nhưng mạnh mẽ được cung cấp bởi Laravel. Khác với hầu hết các templating engine khác, Blade không giới hạn bạn trong việc sử dụng mã PHP thuần trong các View. Tất cả các Blade View sẽ được biên dịch thành mã PHP và được lưu vào bộ đệm cho đến khi chúng được sửa đổi. Các file Blade View có phần mở rộng là `.blade.php` và được lưu trong thư mục `resources/views`.

Hệ thống Blade hỗ trợ cú pháp rất linh hoạt để xử lý điều kiện, vòng lặp, kế thừa layout, và tái sử dụng các phần tử giao diện như components. Blade cũng cung cấp các chỉ thị đặc biệt cho việc xử lý form, bảo mật và các thao tác khác mà không cần phải viết mã PHP phức tạp.

#### ***1.3.2. Sử dụng Blade Template trong đồ án***

Trong đồ án này, Blade Template sẽ được sử dụng để tạo ra các giao diện người dùng (UI) cho hệ thống. Blade là một templating engine mạnh mẽ của Laravel, giúp tách biệt phần giao diện và logic của ứng dụng, đồng thời cho phép dễ dàng tái sử dụng các phần giao diện trong suốt ứng dụng.

Để phân biệt giữa các giao diện quản trị (backend) và giao diện người dùng (frontend), em sẽ chia các view thành hai thư mục riêng biệt trong thư mục `resources/views`. Cấu trúc này giúp quản lý và phân loại các view một cách rõ ràng, dễ dàng cho việc phát triển và bảo trì sau này.

- `resources/views/backend/`: Chứa các view phục vụ cho phần quản trị (backend) của ứng dụng. Đây là nơi các quản trị viên hoặc cộng tác viên sẽ tương tác với hệ thống, như trang quản lý, báo cáo, thống kê, và các chức năng admin khác...

- `resources/views/frontend/`: Chứa các view dành cho người dùng cuối (frontend). Các giao diện trong thư mục này thường liên quan đến các trang như trang chủ, liên hệ, thông tin về tin tức, và các giao diện tương tác với người dùng...

## 1.4. Ngôn ngữ lập trình Javascript

### 1.4.1. Giới thiệu về Javascript

JavaScript là ngôn ngữ lập trình website phổ biến hiện nay, nó được tích hợp và nhúng vào HTML giúp website trở nên sống động hơn. JavaScript đóng vai trò như là một phần của trang web, thực thi cho phép Client-side script từ phía người dùng cũng như phía máy chủ (Nodejs) tạo ra các trang web động.

Nhiệm vụ của Javascript là xử lý những đối tượng HTML trên trình duyệt. Nó có thể can thiệp với các hành động như thêm / xóa / sửa các thuộc tính CSS và các thẻ HTML một cách dễ dàng. Hay nói cách khác, Javascript là một ngôn ngữ lập trình trên trình duyệt ở phía client.

### 1.4.2. Ưu điểm của javascript

- **Dễ học và dễ sử dụng:** Cú pháp đơn giản, dễ hiểu, phù hợp cho cả người mới học lập trình. JavaScript có tài liệu phong phú và cộng đồng hỗ trợ lớn.
- **Hỗ trợ nhiều framework và thư viện:** Các framework như React, Angular, và Vue.js giúp tăng tốc phát triển ứng dụng. Thư viện phong phú như jQuery, D3.js giúp Khả năng mở rộng
- **Khả năng mở rộng:** Có thể sử dụng JavaScript ở cả phía client (trình duyệt) và phía server (với Node.js), giúp xây dựng ứng dụng toàn diện. xử lý DOM, đồ họa và hoạt ảnh dễ dàng hơn.

### 1.4.3. Nhược điểm của javascript

- **Khó khăn trong việc gỡ lỗi:** Tính động và lỏng lẻo về kiểu dữ liệu (loosely-typed) của JavaScript khiến lỗi khó phát hiện sớm. Một số lỗi chỉ xuất hiện khi mã chạy, đặc biệt là trong các ứng dụng phức tạp.



- **Khác biệt giữa các trình duyệt:** JavaScript có thể hoạt động khác nhau trên các trình duyệt, dẫn đến sự không đồng nhất. Nhà phát triển thường phải kiểm tra và tối ưu riêng cho từng trình duyệt, gây tốn thời gian
- **Hiệu suất trên mạng chậm:** JavaScript thường thực hiện các thao tác DOM, có thể chậm khi chạy trên mạng yếu hoặc các thiết bị cũ. Điều này ảnh hưởng đến trải nghiệm người dùng.

#### ***1.4.4. Sử dụng javascript trong đồ án***

Ở đồ án này JavaScript được sử dụng chủ yếu để tạo sự tương tác động và mượt mà cho người dùng và phần giao diện quản lý bên Admin. Cụ thể, JavaScript giúp xử lý các sự kiện như click, hover, và nhập liệu, thay đổi nội dung và kiểu dáng của trang web thông qua DOM, và giao tiếp với server mà không làm mới trang bằng AJAX. Nó còn được sử dụng để quản lý trạng thái ứng dụng trên phía client, ví dụ như lưu trữ thông tin người dùng đăng nhập các tùy chọn mà người dùng đã chọn trong phiên làm việc hiện tại..

### **1.5. Hệ quản trị cơ sở dữ liệu MySQL**

#### ***1.5.1. Giới thiệu về MySQL***

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) mã nguồn mở, được sử dụng rộng rãi để lưu trữ, quản lý và truy xuất dữ liệu. MySQL hoàn toàn miễn phí nằm trong nhóm LAMP (Linux – Apache – MySQL – PHP). Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,...

MySQL được sử dụng cho việc hỗ trợ Node.js, PHP, Perl, và nhiều ngôn ngữ khác, là nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl,...

Hiện nay, những website lớn, hoạt động với lượng người dùng đông đảo như mạng xã hội Facebook, Yahoo, hay Google, Twitter,... đều đang sử dụng hệ

quản trị dữ liệu MySQL để hỗ trợ cho việc lưu trữ thông tin được thực hiện hiệu quả. Từ đó có thể thấy tính hữu dụng, phổ biến của MySQL trong lĩnh vực thiết kế và phát triển website.

### ***1.5.2. Ưu điểm của MySQL***

- **Dễ dàng sử dụng:** là hệ thống khá dễ sử dụng, khiến nó trở thành sự chọn lựa khá phổ biến dành cho các nhà phát triển ở tất cả cấp độ kỹ năng. Sử dụng ngôn ngữ truy vấn đơn giản sẽ giúp dễ dàng truy xuất và thao tác dữ liệu.
- **Khả năng tương thích rộng:** MySQL tương thích với nhiều nền tảng, bao gồm Windows, Linux và macOS.
- **Hiệu suất cao:** được thiết kế cho hiệu suất cao, có nghĩa là nó có thể xử lý lượng lớn dữ liệu và lưu lượng truy cập mà không bị chậm. Nó sử dụng các kỹ thuật lập chỉ mục nâng cao và cơ chế lưu vào bộ nhớ đệm để tối ưu hóa hiệu suất
- **Hỗ trợ từ cộng đồng:** Có một cộng đồng lớn và tích cực bao gồm các nhà phát triển và người dùng, những người đóng góp cho sự phát triển của nó và cung cấp hỗ trợ cho những người khác sử dụng nó.

### ***1.5.3. Nhược điểm của MySQL***

- **Hạn chế dung lượng:** Trong trường hợp số lượng bản ghi ngày càng gia tăng, quá trình truy xuất dữ liệu sẽ trở nên khó khăn. Để giải quyết vấn đề này, bạn có thể áp dụng nhiều biện pháp như tạo cache MySQL hoặc phân tải cơ sở dữ liệu ra nhiều máy chủ.
- **Độ tin cậy thấp:** Một nhược điểm khác của MySQL là trong quá trình thực hiện các chức năng cụ thể như kiểm toán, giao dịch và quản lý tài liệu tham khảo, MySQL có thể trở nên kém tin cậy hơn một số hệ quản trị cơ sở dữ liệu quan hệ khác.

- **Giới hạn chức năng:** MySQL không thiết kế để thực hiện toàn bộ các chức năng và nó đi kèm với những hạn chế về chức năng mà một số ứng dụng có thể cần.

#### ***1.5.4. Sử dụng MySQL trong đồ án***

Trong đồ án này, MySQL được sử dụng như một hệ quản trị cơ sở dữ liệu (DBMS) phổ biến, cung cấp các tính năng mạnh mẽ và ổn định để lưu trữ và truy xuất dữ liệu. MySQL hỗ trợ các thao tác như lưu trữ, tìm kiếm, cập nhật và xóa dữ liệu.

Laravel cung cấp một hệ thống quản lý cơ sở dữ liệu mạnh mẽ thông qua Eloquent ORM (Object-Relational Mapping), cho phép dễ dàng tương tác với MySQL thông qua các mô hình (models). Các thao tác với cơ sở dữ liệu như tạo bảng, thực thi câu lệnh SQL, thêm, sửa, xóa bản ghi hoặc tạo quan hệ giữa các bảng được thực hiện thông qua các phương thức của Eloquent hoặc Query Builder.

Laravel hỗ trợ MySQL thông qua cấu hình trong file `.env`, nơi người phát triển có thể định nghĩa các thông tin kết nối như tên cơ sở dữ liệu, người dùng, mật khẩu và các tùy chỉnh khác. Các migration trong Laravel giúp tạo và quản lý cấu trúc cơ sở dữ liệu, giữ cho mã nguồn luôn đồng bộ với cơ sở dữ liệu trong quá trình phát triển ứng dụng.

### **1.6. Kết luận chương 1**

Trong chương này, em đã trình bày và lựa chọn các công nghệ cần thiết để xây dựng hệ thống. Qua đó, em đã hiểu rõ yêu cầu của dự án và những công nghệ sẽ được áp dụng để phát triển các chức năng của hệ thống một cách hiệu quả. Kết quả của quá trình nghiên cứu này sẽ là cơ sở để triển khai các bước tiếp theo trong việc xây dựng và hoàn thiện hệ thống.

## **CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU**

### **2.1. Xác định yêu cầu chính của hệ thống**

#### **2.1.1. Phân tích các yêu cầu**

##### **Những quyền của người dùng**

- Xem các tin tức: Người dùng có quyền xem nội dung các tin tức trên trang web.
- Đăng nhập: Người dùng có thể đăng nhập hệ thống.
- Bình luận: Người dùng có thể bình luận vào bài viết đã xem sau khi đăng nhập.
- Lưu tin: Người dùng có thể lưu tin yêu thích để có thể xem sau.
- Thay đổi thông tin cá nhân: Người dùng có thể sửa thông tin cá nhân của mình.

##### **Những quyền của quản trị viên**

- Quản lý thành viên: thêm, sửa, xoá thành viên.
- Quản lý nhóm thành viên: thêm, sửa, xoá nhóm thành viên, phân quyền cho các thành viên.
- Quản lý bài viết: xem, thêm, sửa xoá bài viết.
- Quản lý nhóm bài viết: xem, thêm, sửa xoá nhóm bài viết.
- Thay đổi ngôn ngữ của hệ thống.
- Xem thống kê về các bài viết.

##### **Những quyền của cộng tác viên**

- Quản lý bài viết: xem, thêm, sửa xoá bài viết.
- Quản lý nhóm bài viết: xem, thêm, sửa xoá nhóm bài viết.
- Thay đổi ngôn ngữ của hệ thống

#### **2.2.2. Các yêu cầu phi chức năng**

- Hiệu năng cao, trải nghiệm mượt mà, ổn định.

- Giao diện thân thiện.
- Đáp ứng được lượng lớn truy cập đồng thời.
- Dễ dàng bảo trì và mở rộng.

## 2.2. Xác định biểu đồ use case

### 2.2.1. Các tác nhân

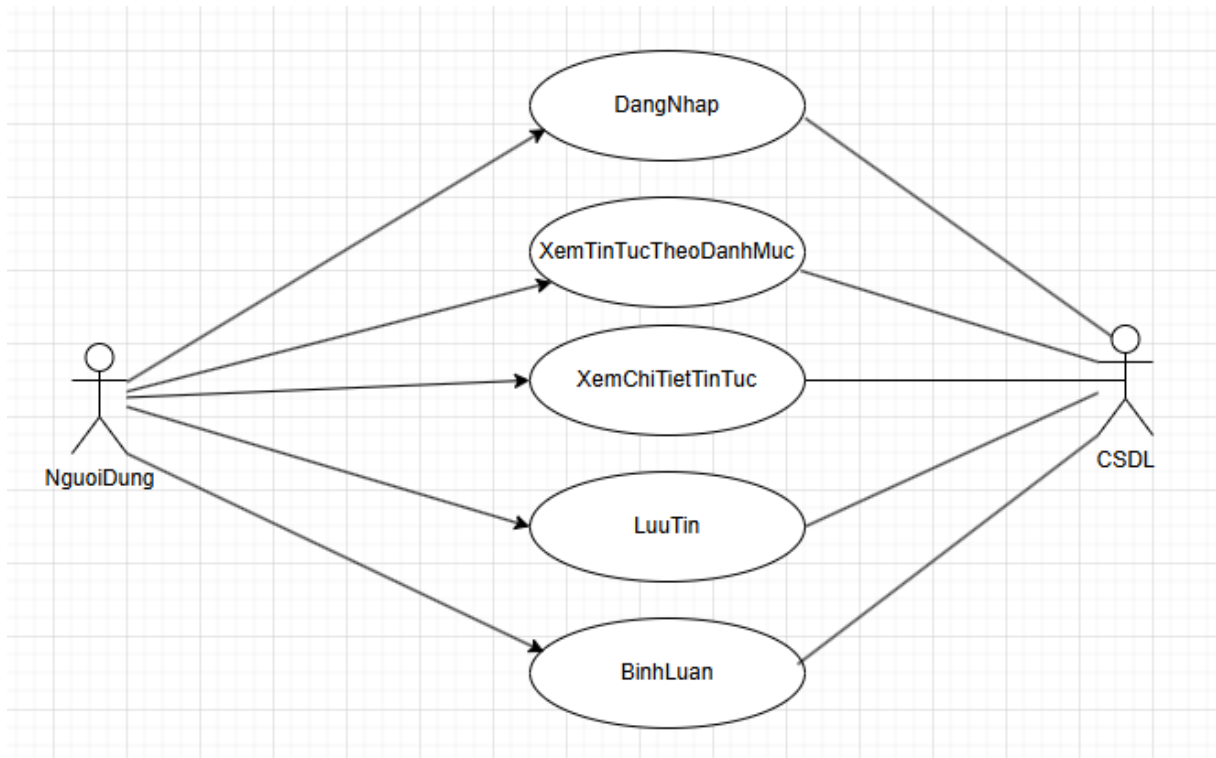
- **Người dùng**
- **Quản trị viên**
- **Cộng tác viên**

*Bảng 2.1: Các use case của hệ thống*

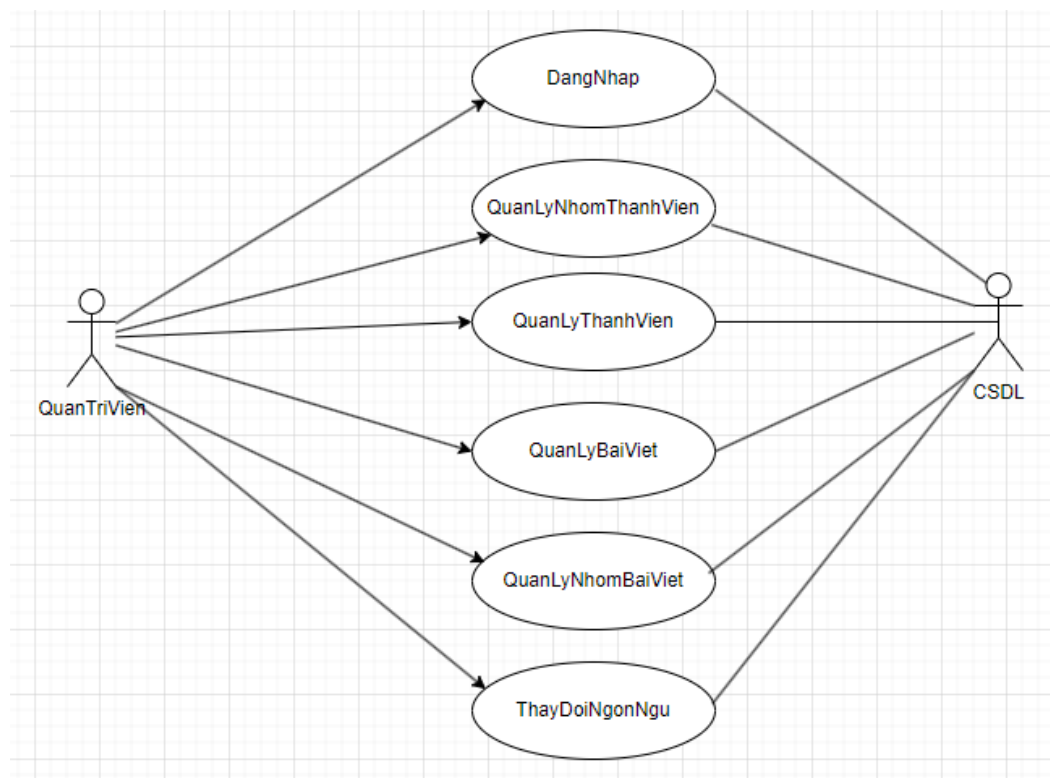
<b>Tác nhân</b>	<b>Các case sử dụng</b>
Người dùng	Đăng nhập Xem bài viết theo danh mục Xem chi tiết bài viết Bình luận về bài viết Lưu bài viết
Quản trị viên	Đăng nhập Quản lý nhóm thành viên Quản lý thành viên Quản lý bài viết Quản lý nhóm bài viết
Công tác viên	Đăng nhập Quản lý bài viết Quản lý nhóm bài viết

### 2.2.2. Các use case của hệ thống

#### 2.2.3. Biểu đồ use case



Hình 2.1: Các use case phía người dùng



Hình 2.2: Các use phía quản trị viên

#### **2.2.4. Mô tả chi tiết các use case**

##### **2.2.4.1. Mô tả use case Đăng nhập (phía quản trị)**

###### **a. Mô tả use case**

**Mô tả vắn tắt:** Use case này cho phép quản trị viên hoặc cộng tác viên đăng nhập vào hệ thống.

###### **Luồng sự kiện**

###### **Luồng cơ bản**

1. Use case này bắt đầu khi Admin hoặc cộng tác viên nhấn vào nút đăng nhập. Hệ thống sẽ yêu cầu nhập tên tài khoản và mật khẩu.
2. Quản trị viên hoặc cộng tác viên sẽ nhập email và mật khẩu sau đó nhấn vào nút “Login”. Hệ thống sẽ kiểm tra tên tài khoản, mật khẩu và kiểm tra trường user\_catalogue\_id trong bảng USERS. Nếu có và user\_catalogue\_id = 1 thì sẽ đăng nhập với vai trò là quản trị viên, còn nếu user\_catalogue\_id = 2 thì sẽ đăng nhập với vai trò là cộng tác viên. Sau đó hệ thống sẽ thông báo đăng nhập thành công và chuyển hướng đến trang quản trị. Use case kết thúc.

###### **Luồng rẽ nhánh**

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
2. Tại bước 2 trong luồng cơ bản khi quản trị viên hoặc cộng tác viên nhập thiếu thông tin hoặc thông tin không hợp lệ hoặc không tìm thấy bản ghi trong bảng USERS. Hệ thống sẽ thông báo lỗi và yêu cầu nhập lại.

###### **Các yêu cầu đặc biệt**

Không có.

###### **Tiền điều kiện**

Không có.

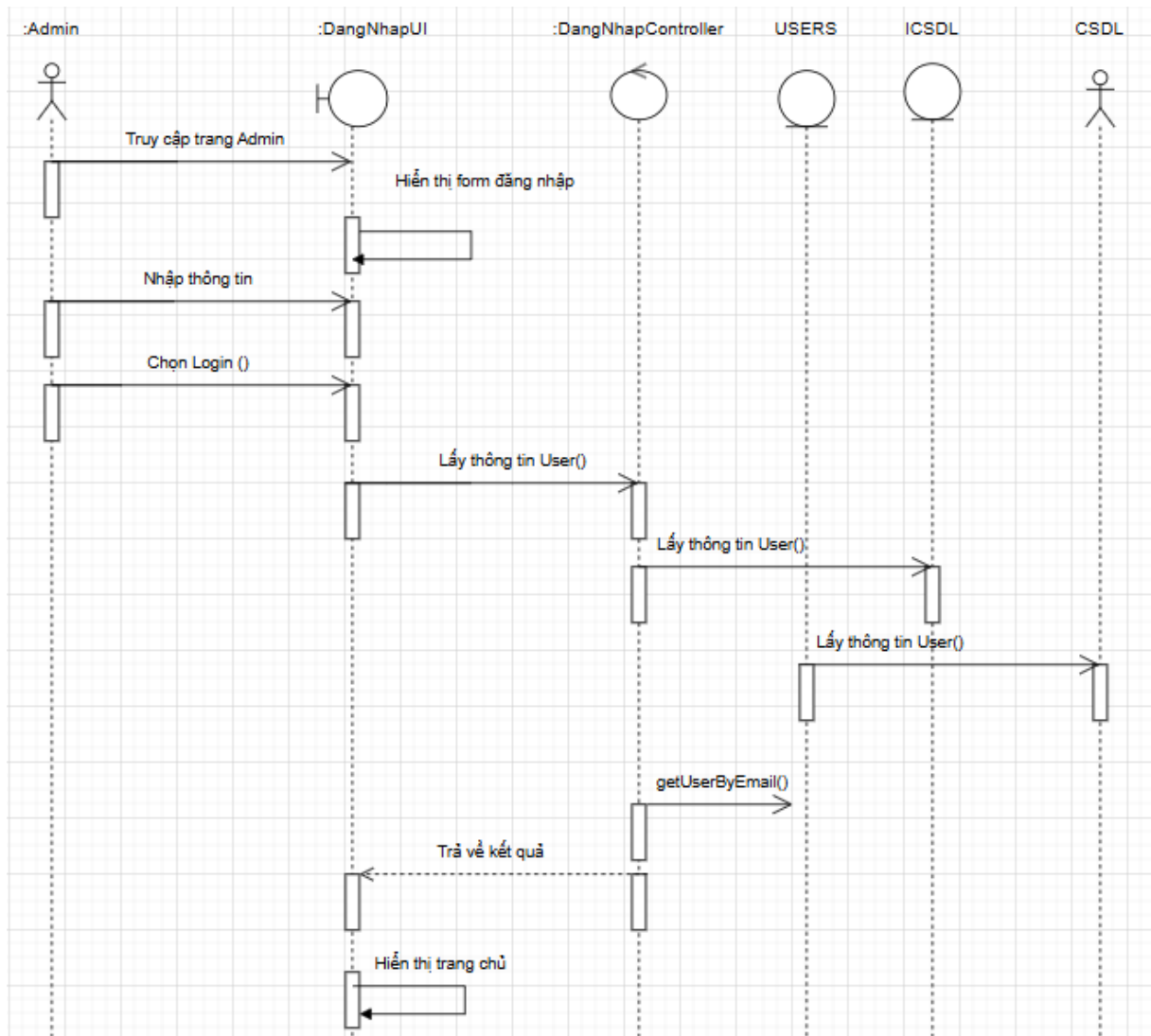
###### **Hậu điều kiện**

Sau khi use case kết thúc thành công thì lưu các thông tin vào bảng USERS.

## Diễn mở rộng

Không có.

### b. Biểu đồ trình tự



Hình 2.3: Biểu đồ trình tự use case Đăng nhập (phía quản trị)

#### 2.2.4.2. Mô tả use case Quản lý nhóm thành viên

##### a. Mô tả use case

**Mô tả vắn tắt:** Use case này cho phép quản trị viên xem, sửa, xóa nhóm thành viên và cập nhật quyền của thành viên.

##### Luồng sự kiện

##### Luồng cơ bản



1. Use case này bắt đầu khi quản trị viên nhấn vào nút “QL Thành Viên” ở menu quản trị và sau đó ấn vào nút “QL Nhóm Thành Viên”. Hệ thống sẽ lấy dữ liệu trong bảng USER\_CATALOGUES gồm các thông tin: name, description và lấy dữ liệu trong bảng USERS các thông tin user\_catalogue\_id. Sau khi lấy được các thông tin hệ thống sẽ hiển thị 1 bảng gồm danh sách các nhóm thành viên bao gồm các thông tin: Tên nhóm thành viên, mô tả, số thành viên và hiển thị lên màn hình.

## 2. Sửa nhóm thành viên

a. Người quản trị nhấn vào hình icon bút màu xanh ở cột thao tác. Hệ thống lấy thông tin ở bảng USER\_CATALOGUES gồm các trường dữ liệu: name, description và hiển thị lên màn hình.

b. Người quản trị sẽ sửa các thông tin của nhóm thành viên bao gồm tên nhóm thành viên và ghi chú sau đó chọn nút “Lưu lại” để lưu thông tin vừa sửa. Hệ thống sẽ hiển thị popup cập nhật thành công lên màn hình và sau đó sẽ cập nhật thông tin vào bảng USER\_CATALOGUES. Usecase kết thúc.

## 3. Cập nhật quyền

a. Người quản trị nhấn vào nút “Cập nhật quyền” ở sau tiêu đề của bảng. Hệ thống sẽ lấy dữ liệu trong bảng USER\_CATALOGUE\_PERMISSION gồm các thông tin là: user\_catalogue\_id và permission\_id và lấy dữ liệu trong bảng PERMISSIONS gồm các thông tin là name. Sau đó hệ thống sẽ hiển thị tất các các quyền và các thành viên.

b. Quản trị viên sẽ tích vào ô checkbox để chọn các quyền cho các thành viên. Sau đó ấn vào nút “Lưu lại” để phân quyền cho các thành viên. Hệ thống sẽ hiển thị popup cập nhật quyền thành công. Use case kết thúc.

## Luồng rẽ nhánh

1. Tại thời điểm bất kỳ trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

## Các yêu cầu đặc biệt

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

### Tiền điều kiện

Người quản trị cần đăng nhập tài khoản quản viên trước khi thực hiện use case.

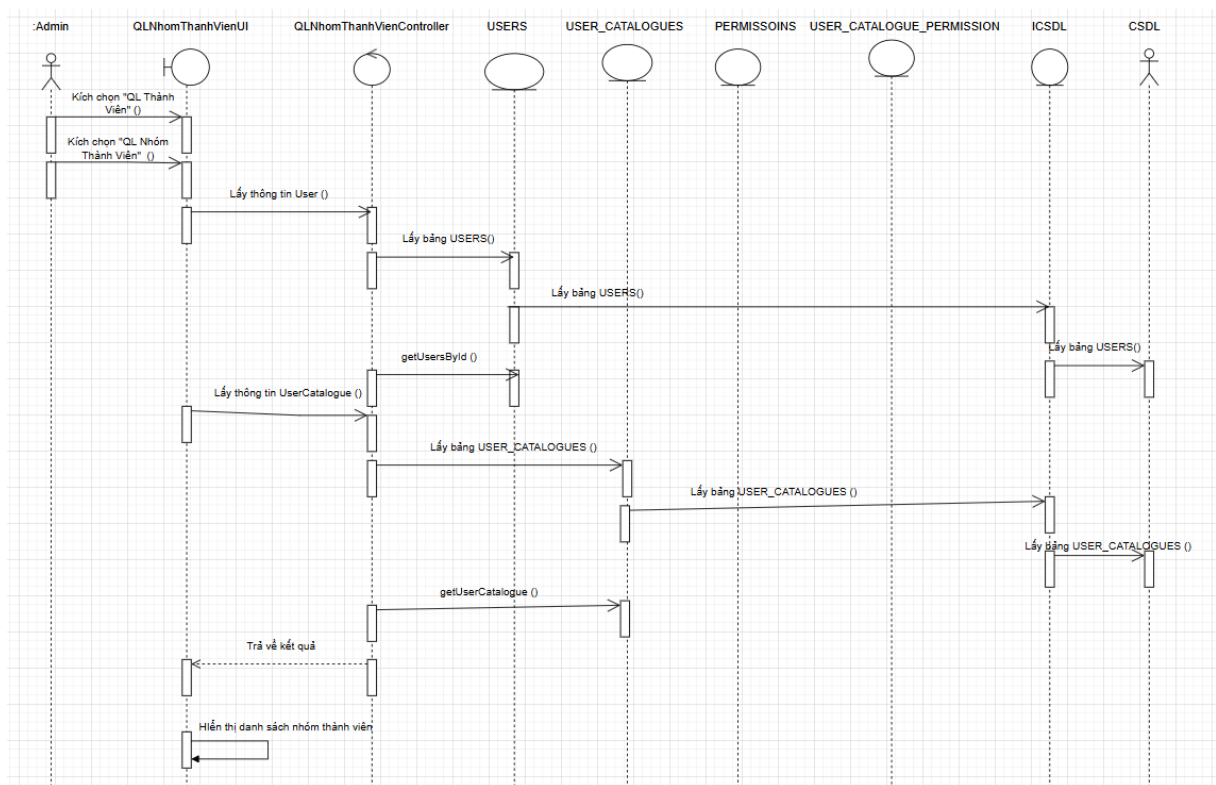
### Hậu điều kiện

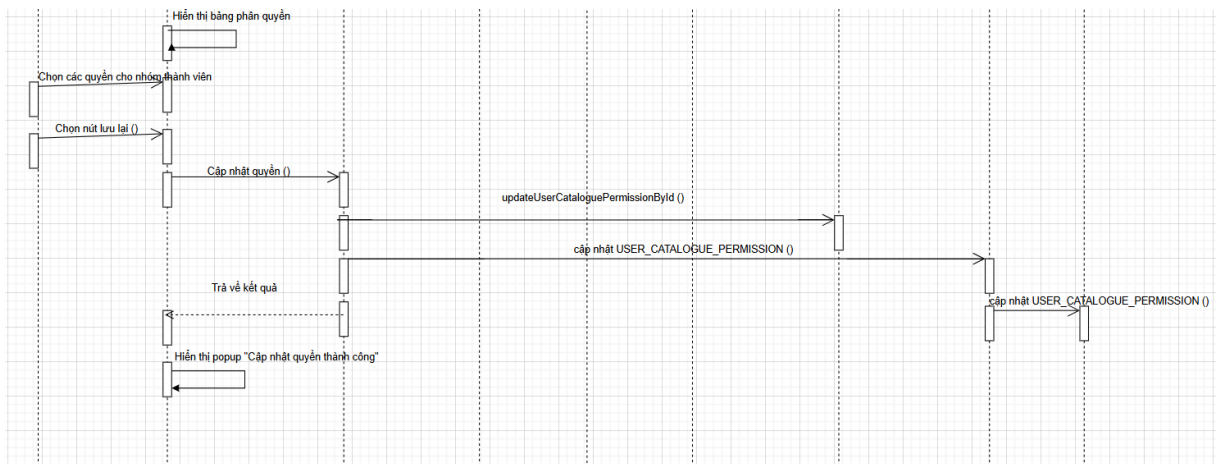
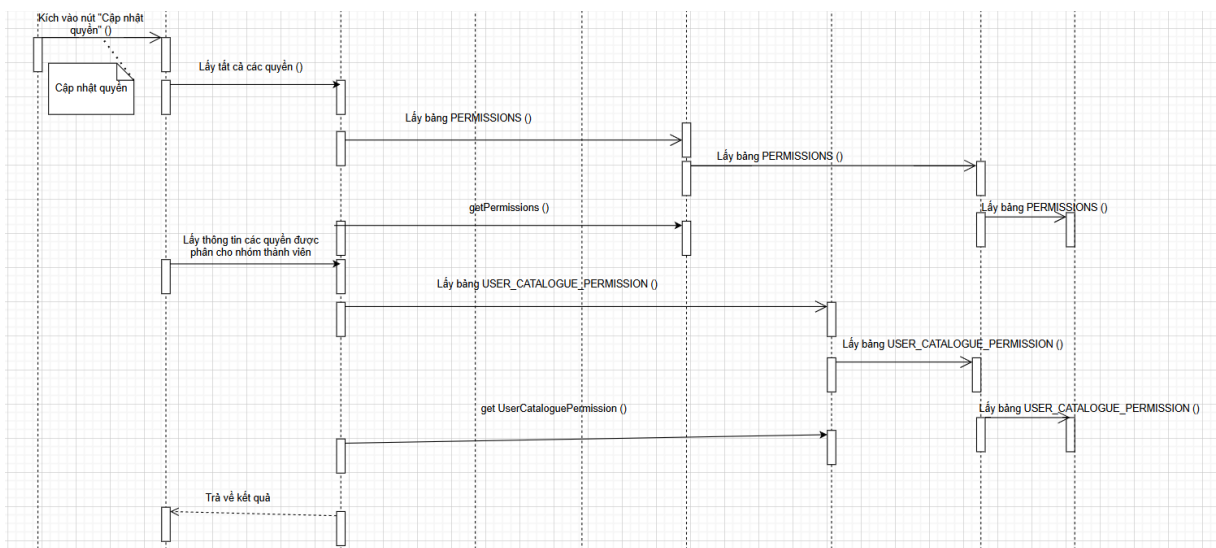
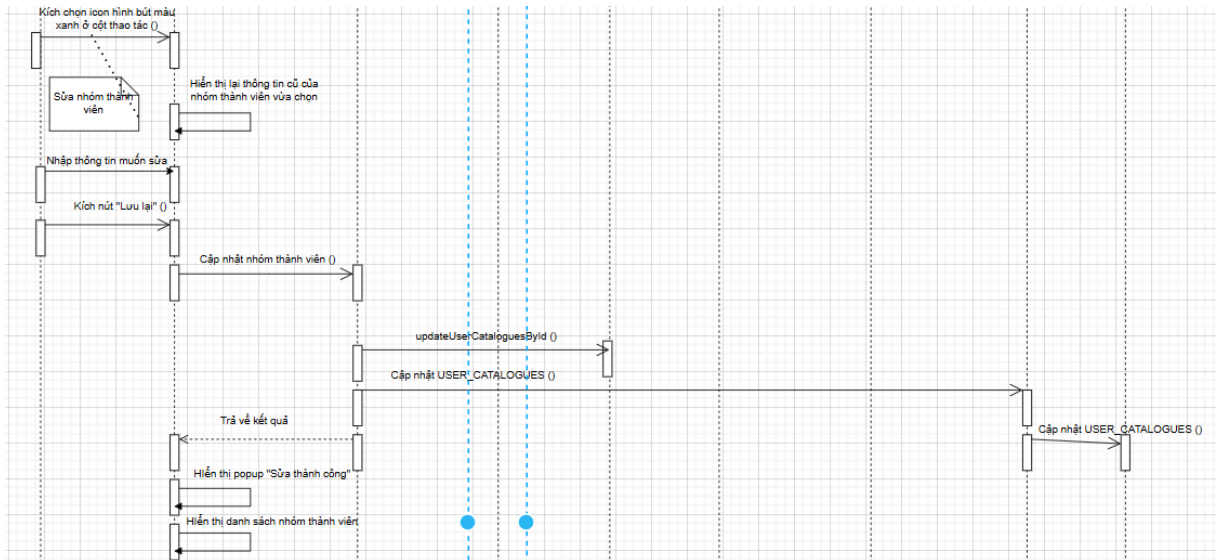
Sau khi use case kết thúc thì lưu thông tin vừa thực hiện trong bảng USER\_CATALOGUE\_PERMISSION và bảng USER\_CATALOGUES.

### Điểm mở rộng

Không có.

## b. Biểu đồ trình tự





Hình 2.4: Biểu đồ trình tự use case Quản lý nhóm thành viên

### 2.2.4.3. Mô tả use case Quản lý thành viên

#### a. Mô tả use case

**Mô tả vắn tắt:** Use case này cho phép người quản trị xem, thêm, tìm, sửa, xoá các thành viên.

## **Luồng sự kiện**

### **Luồng cơ bản**

1. Use case này bắt đầu khi người quản trị chọn nút “QL Thành Viên” và sau đó chọn nút “QL Thành Viên” ở menu quản trị. Hệ thống sẽ lấy thông tin từ bảng USERS gồm các thông tin: name, email, phone và lấy dữ liệu trong bảng USER\_CATALOGUES gồm các thông tin: name. Sau khi lấy được dữ liệu từ cơ sở dữ liệu, hệ thống sẽ hiển thị bảng danh sách thành viên bao gồm các thông tin: Họ tên, email, số điện thoại, vai trò.

### **2. Thêm thành viên**

a. Người quản trị nhấn vào nút “Thêm mới thành viên” sau đó hệ thống sẽ hiển thị form nhập các thông tin của thành viên gồm: Email, họ tên, nhóm thành viên, ngày sinh, mật khẩu, thành phố, quận/huyện, phường/xã, địa chỉ, số điện thoại, ghi chú.

b. Người quản trị nhập thông tin cho thành viên mới sau đó nhấn vào nút “Lưu lại”. Sau đó hệ thống sẽ quay lại danh sách thành viên và hiển thị popup thêm nhân viên thành công. Hệ thống sẽ thêm 1 bản ghi mới vào bảng USERS. Use case kết thúc.

### **3. Tìm thành viên**

a. Người quản trị nhập thông tin muốn tìm kiếm vào thanh tìm kiếm. Sau đó kích vào nút tìm kiếm. Hệ thống sẽ lấy tìm thông tin trong bảng USERS và sau đó hiển thị các thành viên mà bạn muốn tìm. Usecase kết thúc.

b. Người quản trị nhấn vào nút “Chọn nhóm thành viên” sau đó sẽ hiển thị combobox hiển thị ra các nhóm thành viên. Người quản trị chọn nhóm thành viên muốn tìm và nhấn vào nút “tìm kiếm”. Hệ thống sẽ kiểm tra trong bảng USERS và hiển thị các thành viên mà bạn muốn tìm. Use case kết thúc.

### **4. Sửa thành viên**

a. Người quản trị nhấn vào icon hình bút màu xanh ở cột thao tác. Hệ thống sẽ dữ liệu trong bảng USERS gồm các thông tin: name, phone, province\_id, district\_id, ward\_id, birthday, description, email và lấy dữ liệu trong bảng USER\_CATALOGUES gồm thông tin: name và bảng PROVINCES sau đó hiển thị form cập nhật thành viên lên màn hình quản trị.

b. Người quản trị sửa lại các thông tin của thành viên sau đó ấn vào nút “Lưu lại”. Hệ thống sẽ cập nhật thông tin vào bảng USERS sau đó hiển thị popup “Cập nhật thành công” và hiển thị lại danh sách các thành viên. Use case kết thúc.

## 5. Xóa thành viên

a. Người quản trị nhấn vào icon thùng rác màu đỏ ở cột thao tác. Hệ thống sẽ lấy dữ liệu từ bảng USERS gồm thông tin: email và bảng USER\_CATALOGUES gồm thông tin: name. Sau đó hiển thị thông tin lên màn hình gồm email và vai trò và hiển thị form xác nhận xóa.

b. Người quản trị nhấn vào nút “Xóa”, hệ thống sẽ thực hiện xóa người dùng đã chọn khỏi bảng USERS. Use case kết thúc.

## Luồng rẽ nhánh

1. Tại thời điểm bất kỳ trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

2. Tại hai bước 2b và 4b trong luồng cơ bản nếu người quản trị nhập thông tin không hợp lệ, hệ thống sẽ thông báo lỗi yêu cầu người quản trị nhập lại hoặc nhấn “Huỷ” để kết thúc.

3. Tại bước 5b trong luồng cơ bản nếu người quản trị nhấn vào nút “Huỷ” thì hệ thống sẽ bỏ qua thao tác và hiển thị danh sách thành viên và use case kết thúc.

## Các yêu cầu đặc biệt

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

## Tiền điều kiện

Người quản trị cần đăng nhập với quyền quản trị trước khi thực hiện use case.

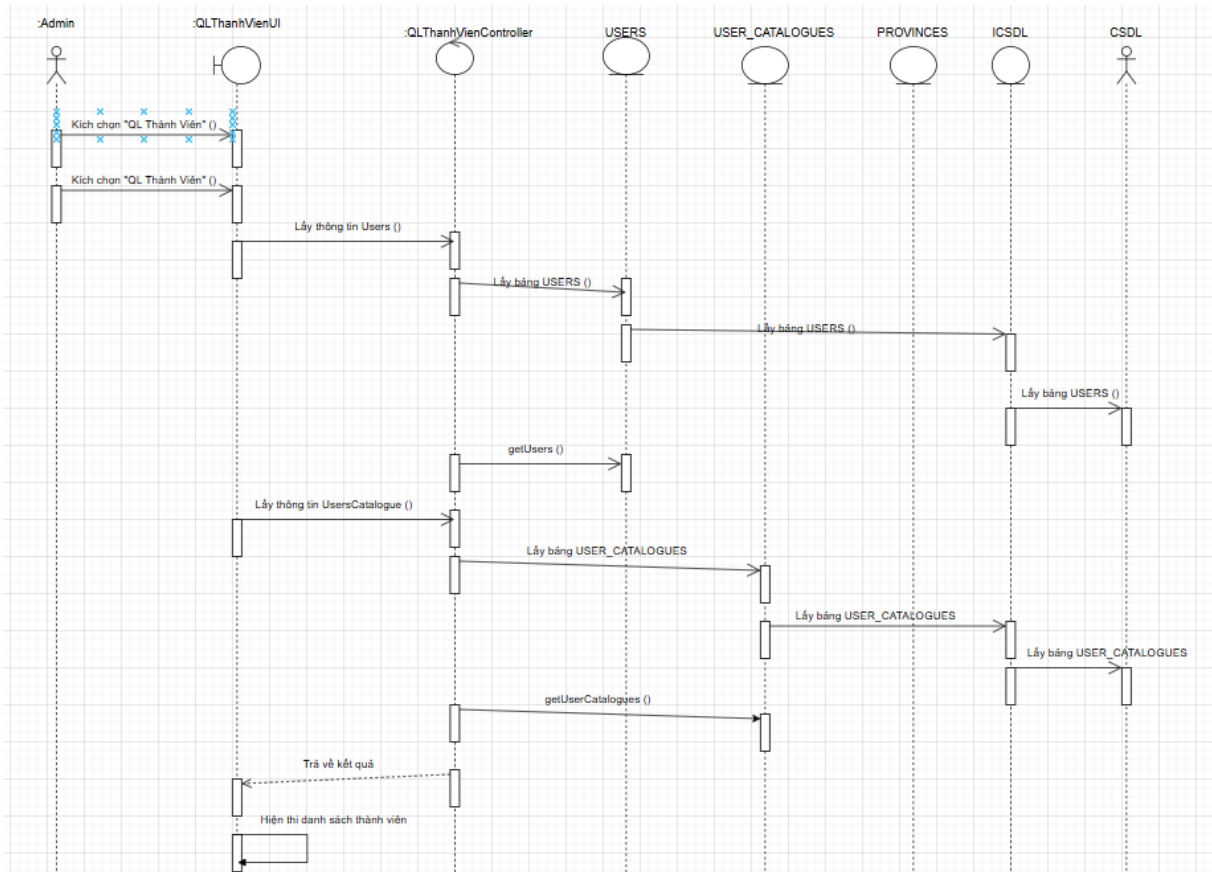
## Hậu điều kiện

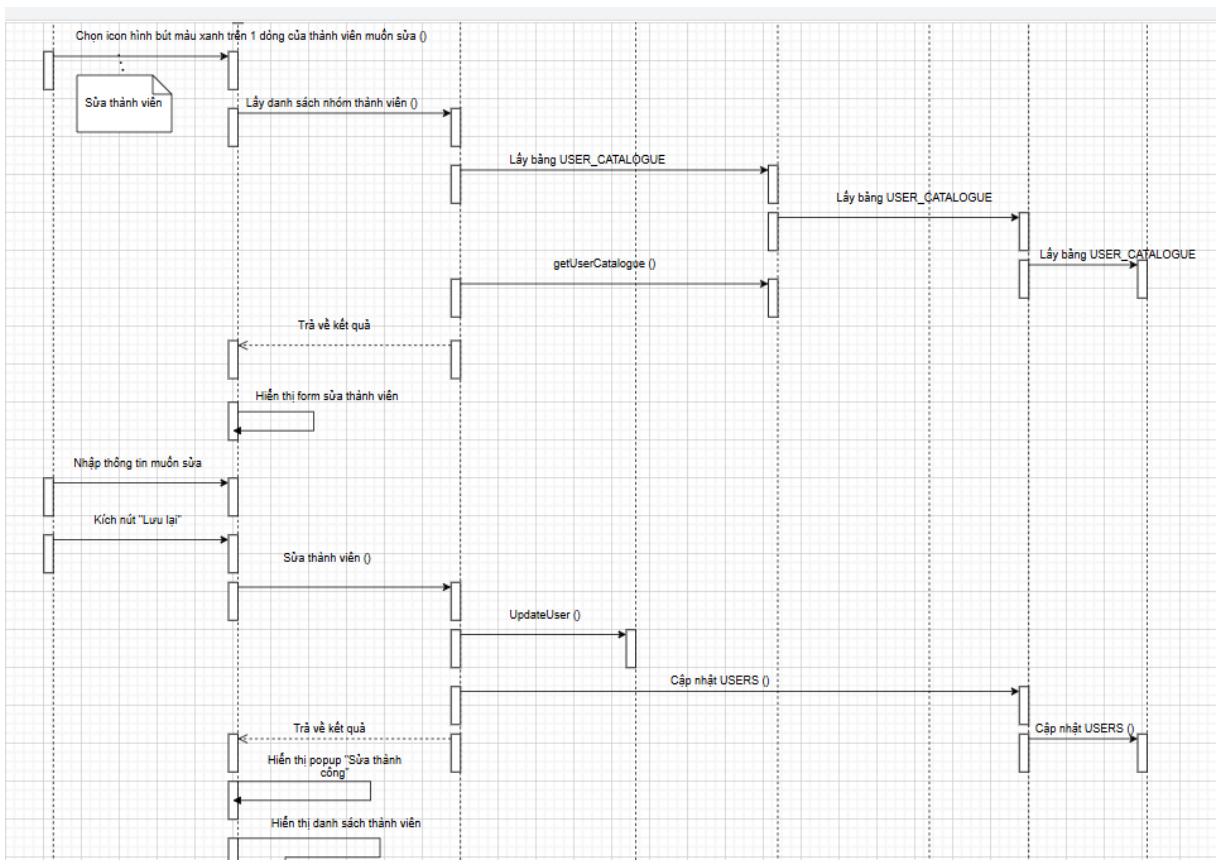
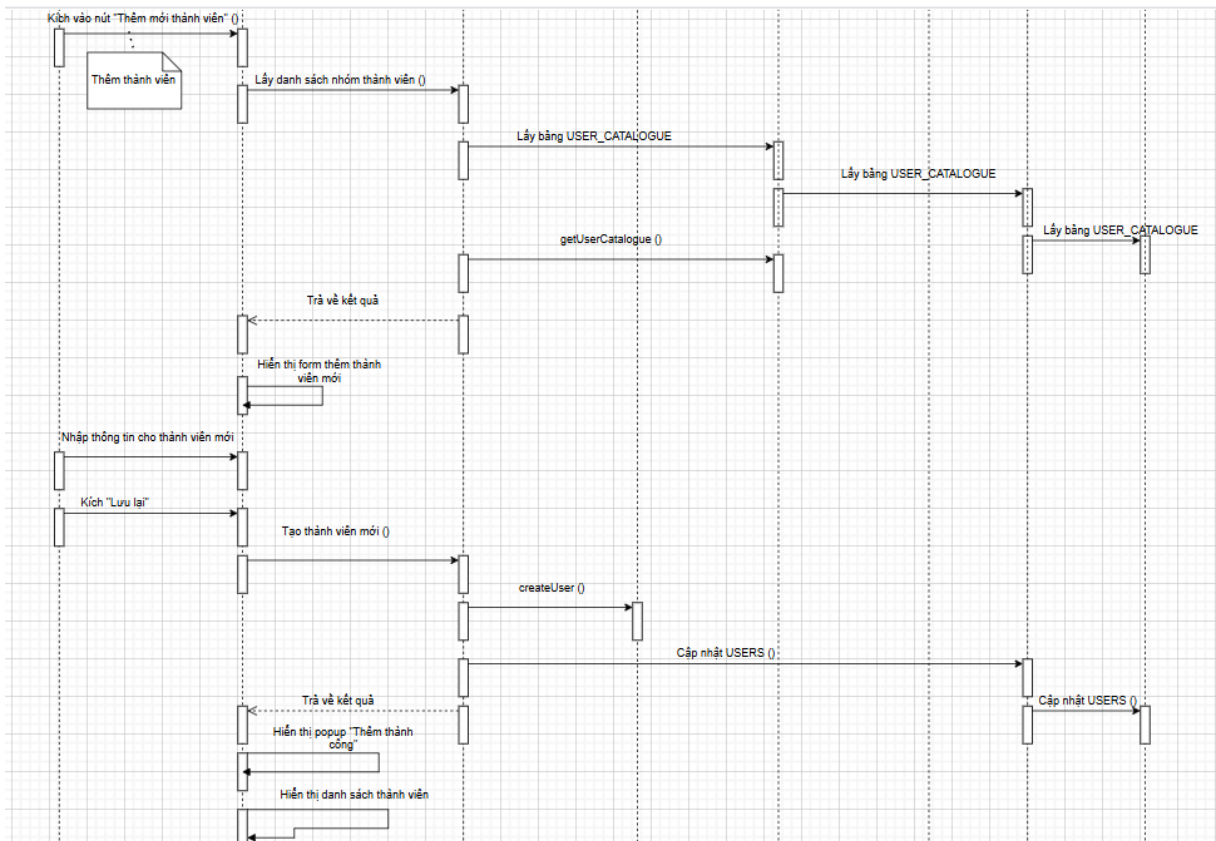
Sau khi use case kết thúc thành công thì lưu các thông tin vào bảng USERS.

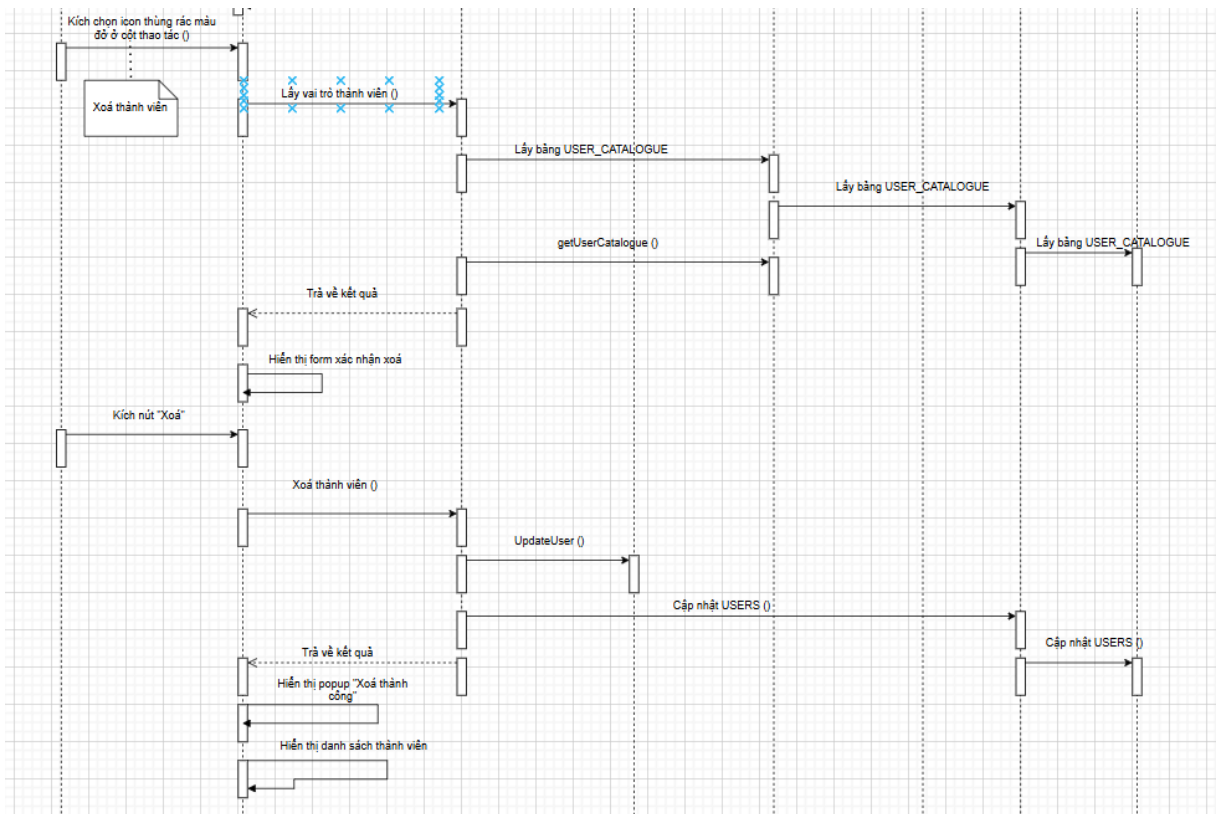
## Diễn mở rộng

Không.

## b. Biểu đồ trình tự







Hình 2.5: Biểu đồ trình tự use case Quản lý thành viên

#### 2.2.4.4. Mô tả use case Quản lý bài viết

##### a. Mô tả use case

**Mô tả vắn tắt:** Use case này cho phép người quản trị hoặc cộng tác viên thêm, sửa, xóa bài viết.

##### Luồng sự kiện

##### Luồng cơ bản

1. Use case này bắt đầu khi người quản trị hoặc cộng tác viên ấn vào nút “QL Bài viết” và sau đó ấn vào nút “QL Bài viết” ở bên trái menu quản trị. Hệ thống sẽ lấy từ bảng POSTS và bảng POST\_LANGGUAGE và bảng POST\_CATALOGUES và bảng POST\_CATALOGUES\_POST và bảng POST\_CATALOGUES và bảng POST\_CATALOGUE\_LANGUAGE sau đó lấy thông tin của bài viết bao gồm: tiêu đề, mô tả ngắn, nội dung, danh mục, ảnh đại diện, tình trạng và hiển thị thông tin của bài viết hiện tại lên màn hình.

2. Thêm bài viết mới



a. Người quản trị ấn vào nút “Thêm bài viết mới”. Hệ thống sẽ hiển thị form để người quản trị nhập thông tin bao gồm: tiêu đề, mô tả ngắn, nội dung, danh mục, ảnh đại diện, tình trạng.

b. Người quản trị nhập thông tin và sau đó ấn vào nút “Lưu lại”. Hệ thống sẽ hiển thị lên popup “Thêm thành công” và sẽ thêm 1 bản ghi mới vào bảng POST\_LANGUAGE. Use case kết thúc.

### 3. Sửa bài viết

a. Người quản trị ấn vào icon hình bút màu xanh ở cột thao tác trên 1 dòng của bài viết muốn sửa. Hệ thống sẽ lấy từ bảng POSTS và bảng POST\_LANGUAGE và bảng POST\_CATALOGUES và bảng POST\_CATALOGUES\_POST và bảng POST\_CATALOGUE và bảng POST\_CATALOGUE\_LANGUAGE sau đó lấy thông tin của bài viết bao gồm: tiêu đề, mô tả ngắn, nội dung, danh mục, ảnh đại diện, tình trạng và hiển thị thông tin của bài viết hiện tại lên màn hình.

b. Người quản trị sẽ sửa những thông tin muốn sửa và sau đó nhấn vào nút “Lưu lại”. Hệ thống sẽ hiển thị popup “Sửa thành công” và cập nhật thông tin vào bảng POST\_LANGUAGE. Use case kết thúc.

### 4. Xóa bài viết

a. Người quản trị nhấn vào biểu tượng thùng rác màu đỏ trên một dòng của bài viết. Hệ thống sẽ hiển thị form xác nhận xóa.

b. Người quản trị nhấn vào nút “Xóa”, hệ thống sẽ hiển thị popup “Xóa thành công” và thực hiện xóa bài viết đã chọn khỏi bảng POST\_LANGUAGE và bảng POST\_CATALOGUE\_POST và bảng POSTS. Use case kết thúc.

### **Luồng rẽ nhánh**

1. Tại thời điểm bất kỳ trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

2. Tại bước 1 trong luồng cơ bản nếu không có bản ghi nào trong cơ sở dữ liệu hệ thống sẽ hiển thị chưa có bài viết nào và use case kết thúc.

3. Tại bước 2b và 3b trong luồng cơ bản, khi quản trị viên nhập thông tin bài viết không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi yêu cầu người quản trị nhập lại, quản trị viên tiếp tục nhập lại hoặc nhấn trở về để kết thúc.

4. Tại bước 4b trong luồng cơ bản nếu người quản trị nhấn vào nút “Huỷ” thì hệ thống sẽ bỏ qua thao tác và hiển thị danh sách các bài viết lên màn hình và use case kết thúc.

#### **Các yêu cầu đặc biệt**

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

#### **Tiền điều kiện**

Người quản trị cần đăng nhập với quyền quản trị trước khi thực hiện use case.

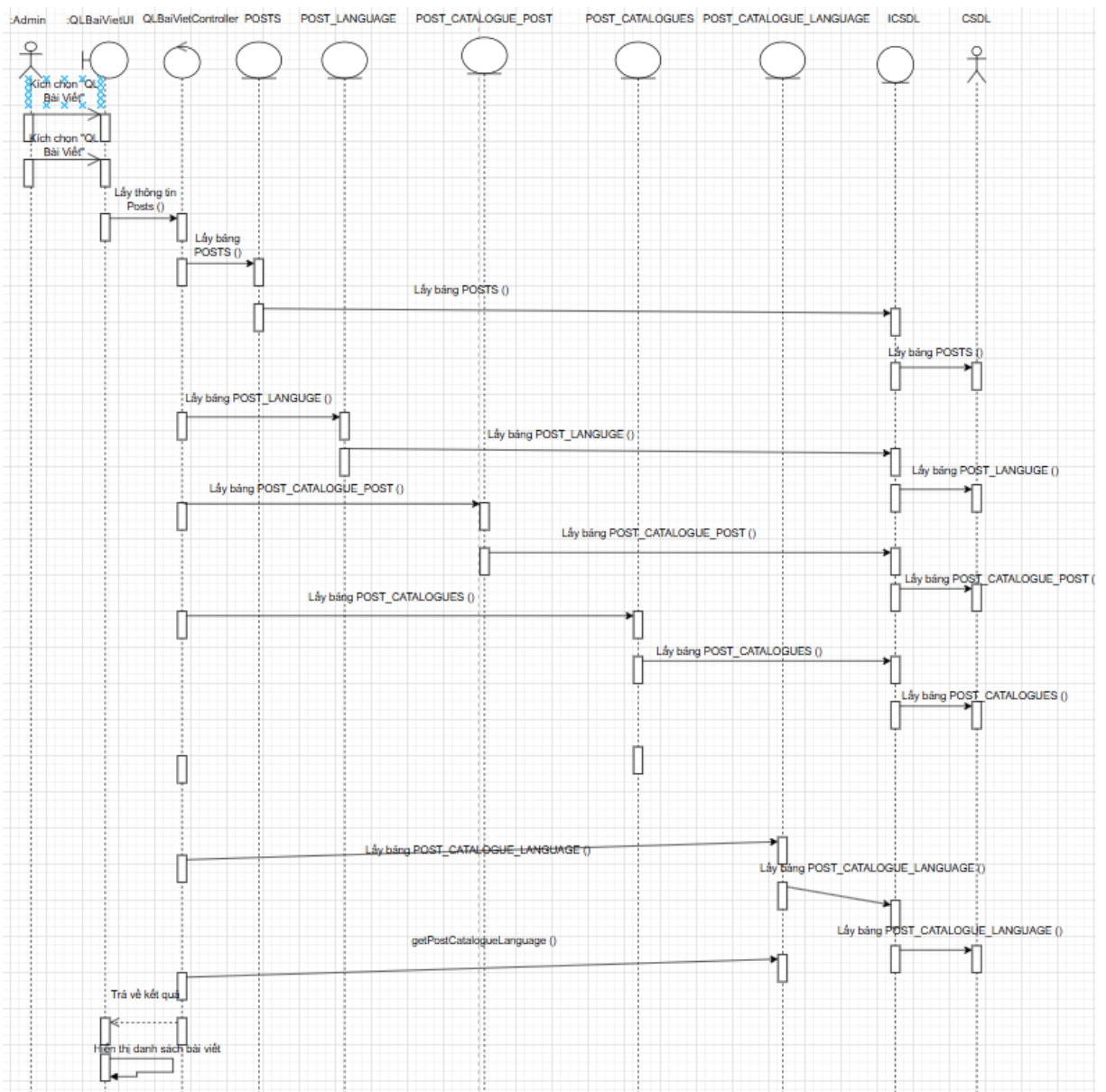
#### **Hậu điều kiện**

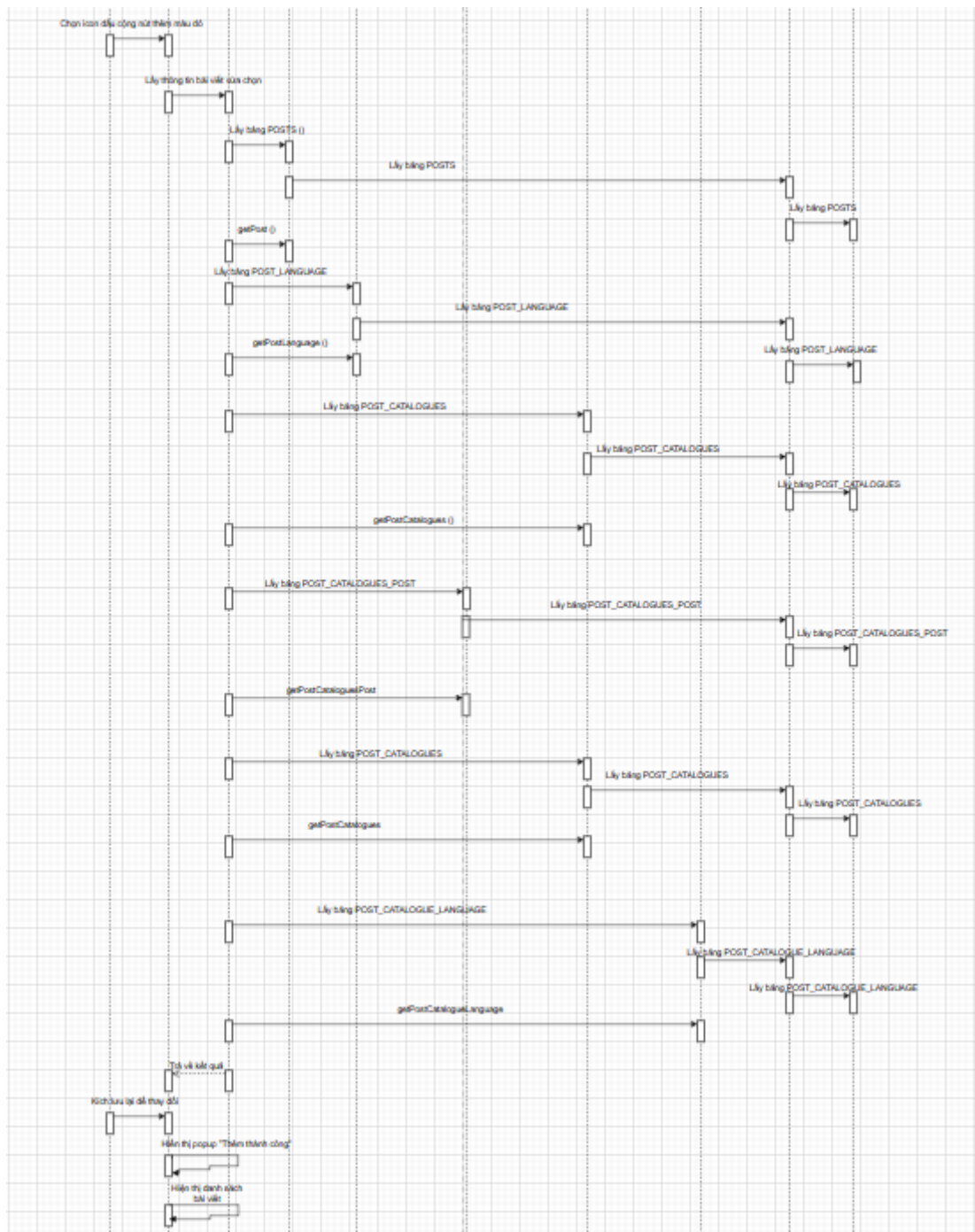
Không có.

#### **Điểm mở rộng**

Không có.

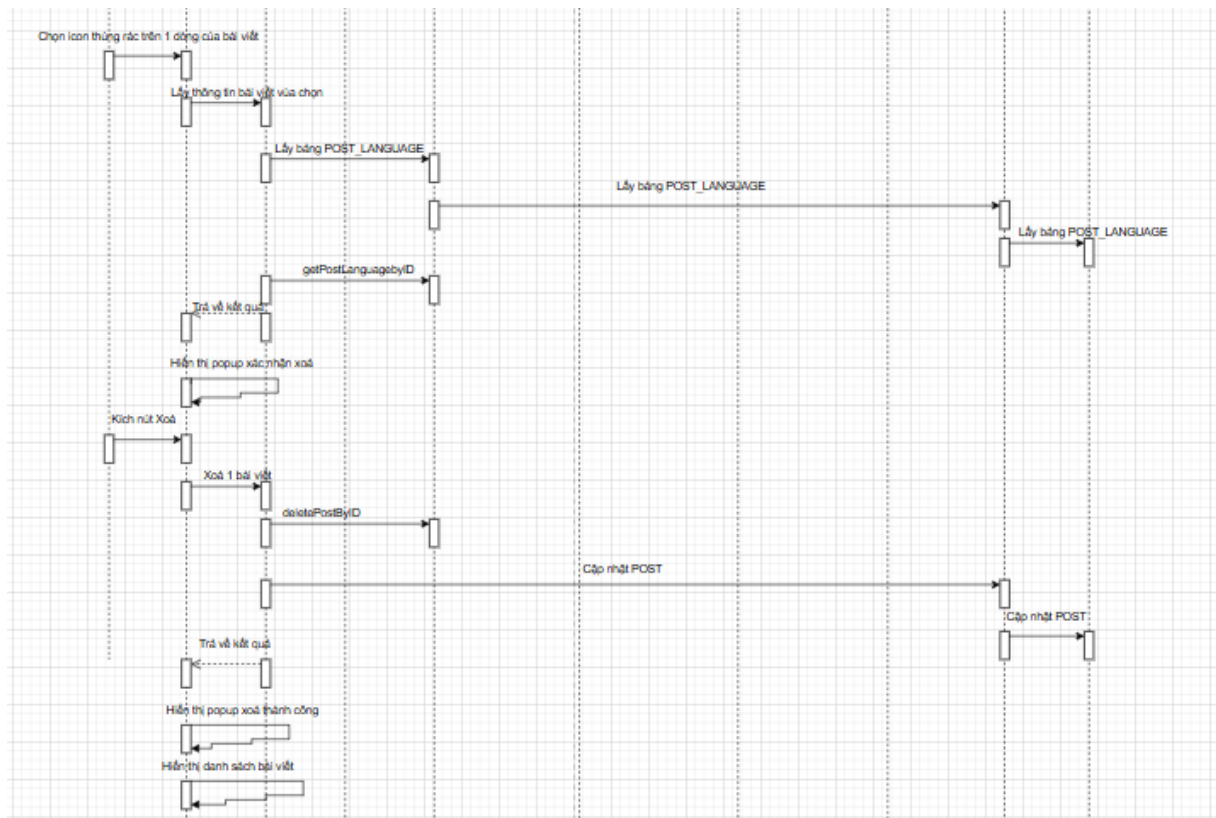
#### **b. Biểu đồ trình tự**











Hình 2.6: Biểu đồ trình tự use case Quản lý bài viết

#### 2.2.4.5. Mô tả use case Quản lý nhóm bài viết

##### a. Mô tả use case

**Mô tả vắn tắt:** Use case này cho phép người quản trị thêm, sửa, xóa nhóm bài viết.

##### Luồng sự kiện

Luồng cơ bản

1. Use case này bắt đầu khi người quản trị ấn vào nút “QL Bài viết” sau đó ấn vào nút “QL Nhóm Bài Viết” ở menu quản trị. Hệ thống sẽ lấy trong bảng POST\_CATALOGUES và POST\_CATALOGUE\_LANGUAGE gồm các thông tin: name, publish, level và hiển thị lên màn hình.

2. Thêm nhóm bài viết

a. Người quản trị nhấn vào nút “Thêm mới nhóm bài viết”. Hệ thống sẽ hiển thị form yêu cầu nhập thông tin cho nhóm bài viết bao gồm: Tiêu đề, mô tả ngắn, chọn danh mục cha, ảnh đại diện, tình trạng.

b. Người quản trị nhập thông tin cho nhóm bài viết mới và nhấn vào nút “Lưu lại”. Hệ thống sẽ hiển thị popup “Thêm thành công” và thêm bản ghi mới vào POST\_CATALOGUE. Use case kết thúc.

### 3. Sửa nhóm bài viết

a. Người quản trị nhấn vào icon hình bút màu xanh trên 1 dòng. Hệ thống sẽ lấy dữ liệu từ bảng POST\_CATALOGUES bao gồm: id, parentid, image, publish, lft, rgt, level và bảng POST\_CATALOGUE\_LANGUAGE bao gồm: description, content và hiển thị lên màn hình.

b. Người quản trị sẽ sửa các thông tin của nhóm bài viết sau đó nhấn vào nút “Lưu lại”. Hệ thống sẽ hiển thị popup “Sửa thành công” cập nhập thông tin vào bảng POST\_CATALOGUE. Use case kết thúc

### 4. Xóa nhóm bài viết

a. Người quản trị nhấn vào biểu tượng thùng rác màu đỏ ở trên 1 dòng của nhóm bài viết. Hệ thống sẽ hiển thị form xác nhận xóa.

b. Người quản trị nhấn vào nút “Xóa”, hệ thống sẽ hiển thị popup “Xóa thành công” và xóa 1 bản ghi từ bảng POST\_CATALOGUE sau đó hiển thị lại danh sách nhóm bài viết lên màn hình. Use case kết thúc.

## **Luồng rẽ nhánh**

1. Tại thời điểm bất kỳ trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

2. Tại bước 1 trong luồng cơ bản, nếu không có bản ghi nào thì hệ thống sẽ không hiển thị nhóm bài viết và use case kết thúc.

3. Tại bước 2b và 3b trong luồng cơ bản, khi quản trị viên nhập thông tin sản phẩm không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi yêu cầu người quản trị nhập lại, quản trị viên có thể tiếp tục nhập lại hoặc nhấn quay lại để kết thúc.



4. Tại bước 4b trong luồng cơ bản, nếu người quản trị nhấn vào nút huỷ thì hệ thống sẽ bỏ qua thao tác xóa và hiển thị lại danh sách các nhóm bài viết lên màn hình.

### Các yêu cầu đặc biệt

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

### Tiền điều kiện

Người quản trị cần đăng nhập với quyền quản trị trước khi thực hiện use case.

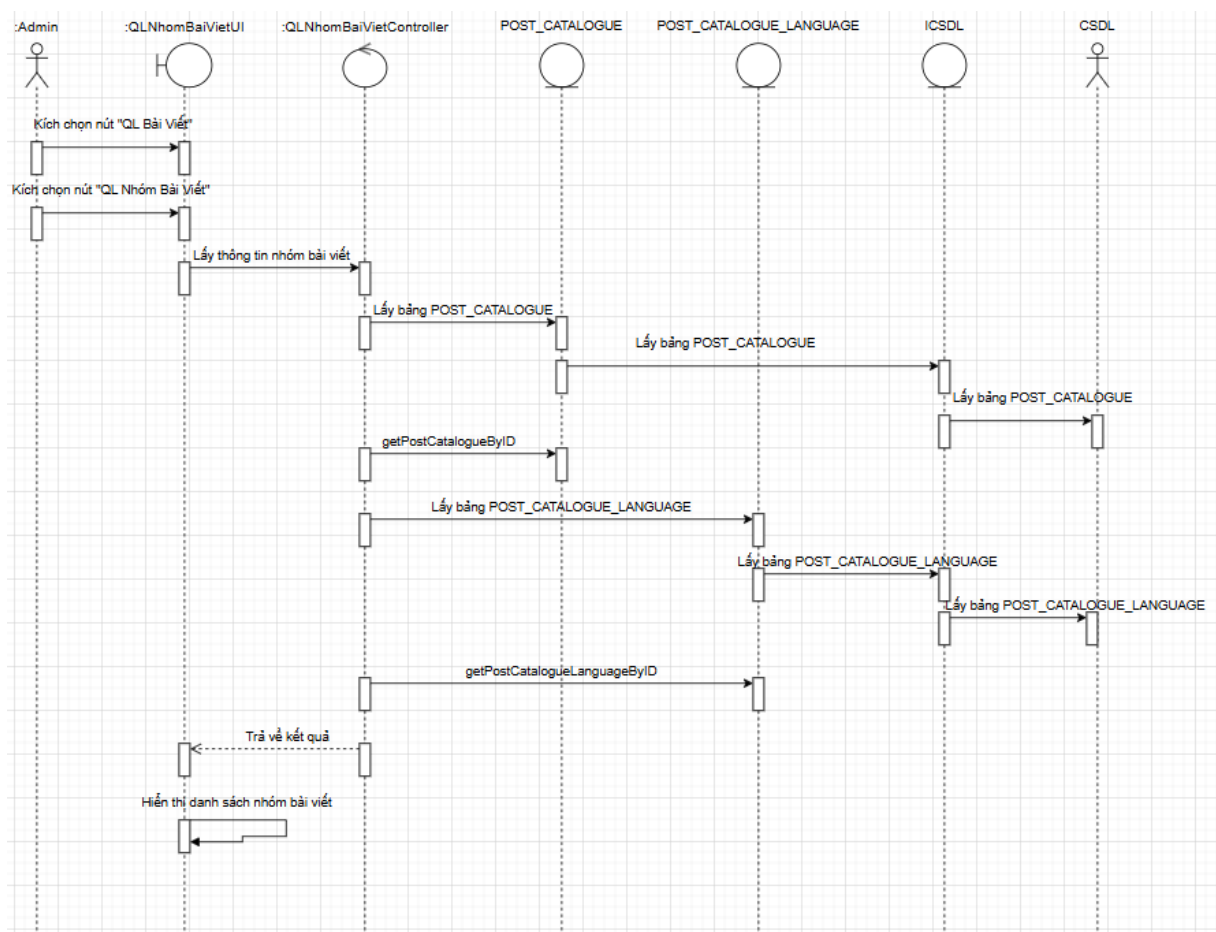
### Hậu điều kiện

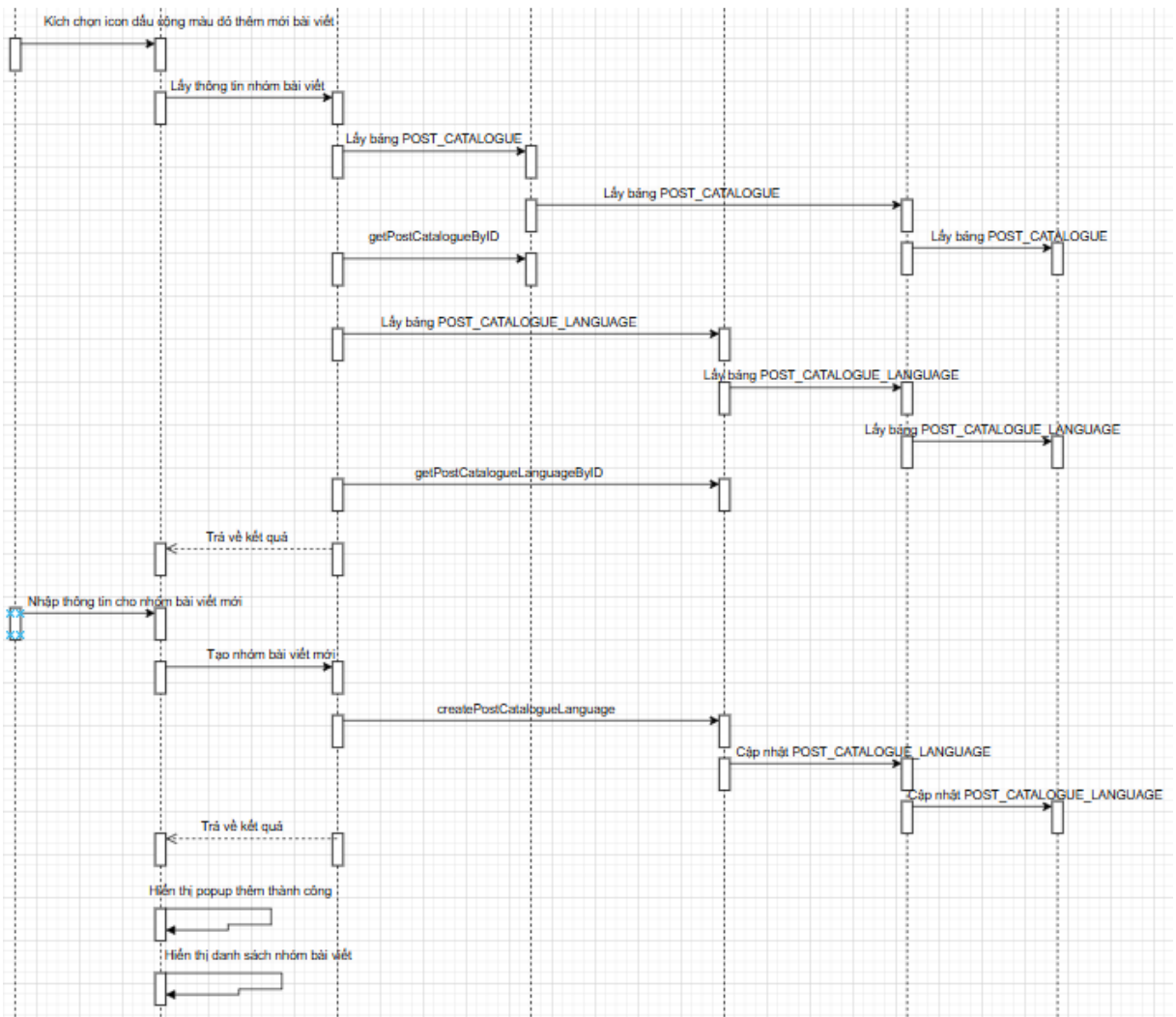
Sau khi use case kết thúc thành công thì lưu các thông tin vào bảng POST\_CATALOGUE.

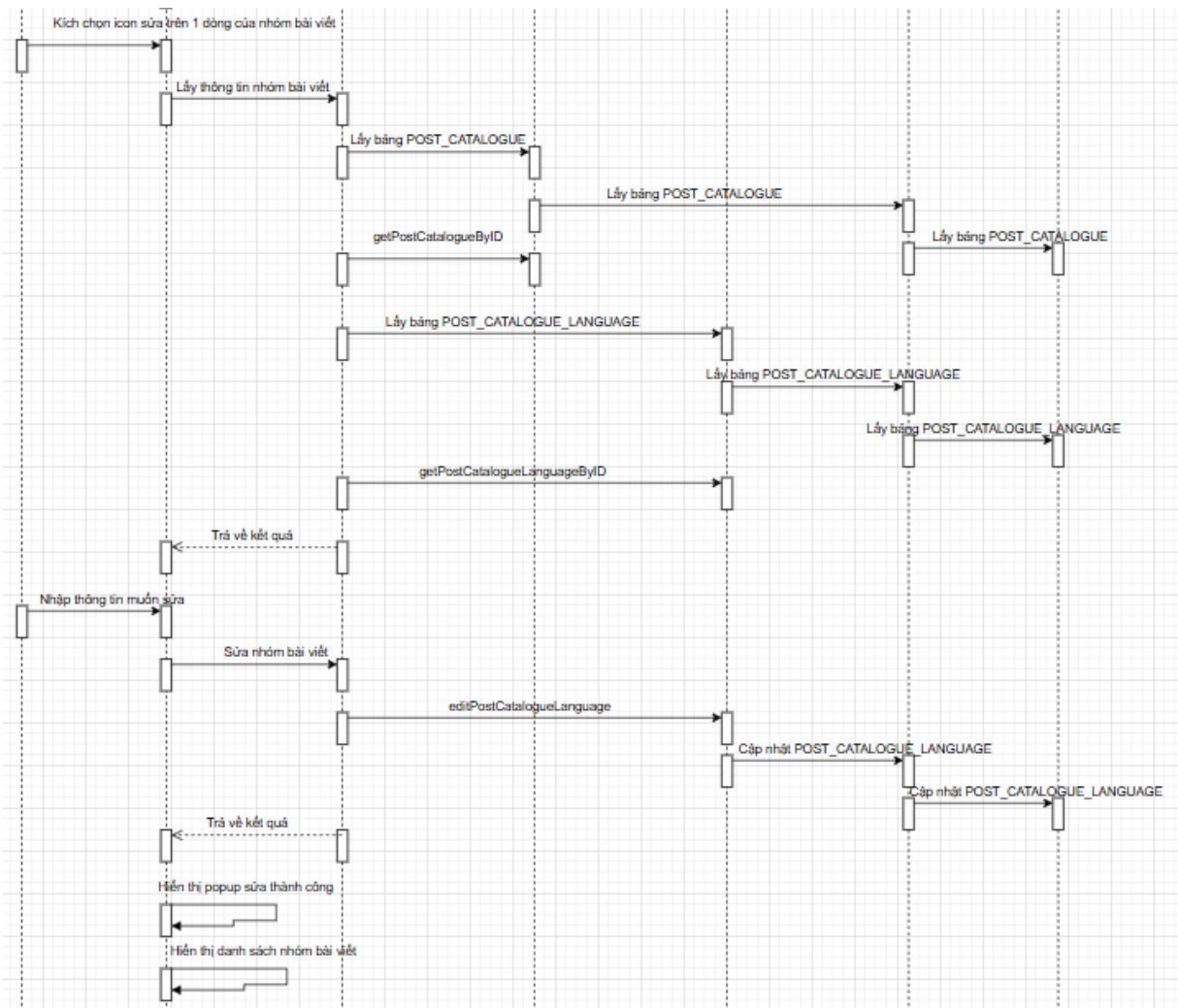
### Điểm mở rộng

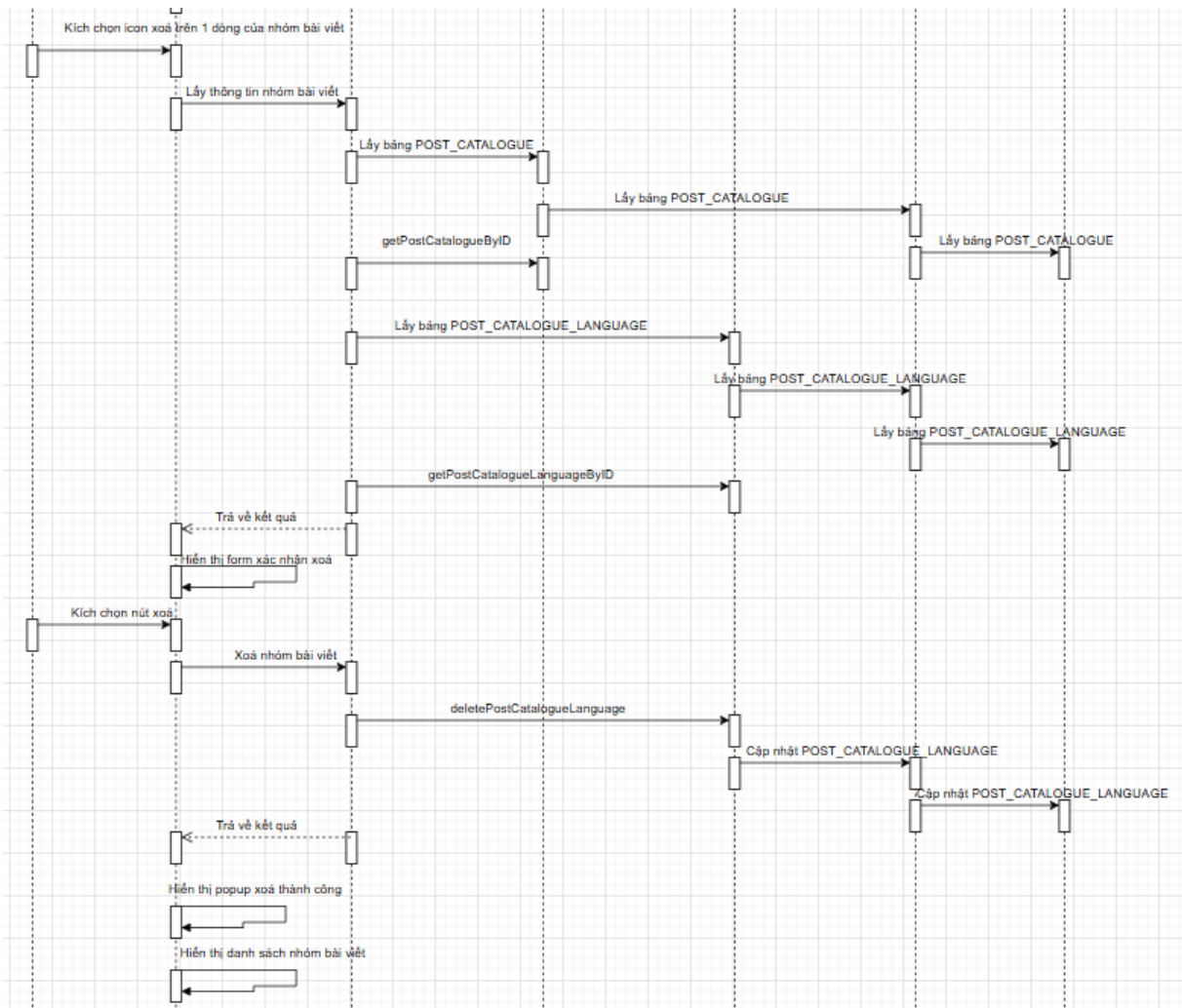
Không có.

### b. Biểu đồ trình tự









Hình 2.7: Biểu đồ trình tự use case Quản lý nhóm bài viết

#### 2.2.4.6. Mô tả use case Đăng nhập (phía người dùng)

##### a. Mô tả use case

**Mô tả vắn tắt:** Use case này cho phép người dùng đăng nhập vào hệ thống.

##### Luồng sự kiện

##### Luồng cơ bản

1. Use case này bắt đầu khi người dùng kích vào nút đăng nhập với Google. Hệ thống sẽ chuyển sang trang đăng nhập bằng google.
2. Người dùng sẽ nhập thông tin tài khoản google của mình. Hệ thống sẽ kiểm tra tài khoản google nếu đúng sẽ thêm thông tin của người dùng vào bảng USERS và thông báo “Chào mừng bạn đến với trang tin tức” và chuyển hướng đến trang tin tức. Use case kết thúc.

### **Luồng rẽ nhánh**

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
2. Tại bước 2 trong luồng cơ bản khi người dùng nhập thiếu thông tin hoặc thông tin không hợp lệ hoặc không tìm thấy tài khoản Google. Hệ thống sẽ thông báo lỗi và yêu cầu nhập lại.

### **Các yêu cầu đặc biệt**

Không có.

### **Tiền điều kiện**

Không có.

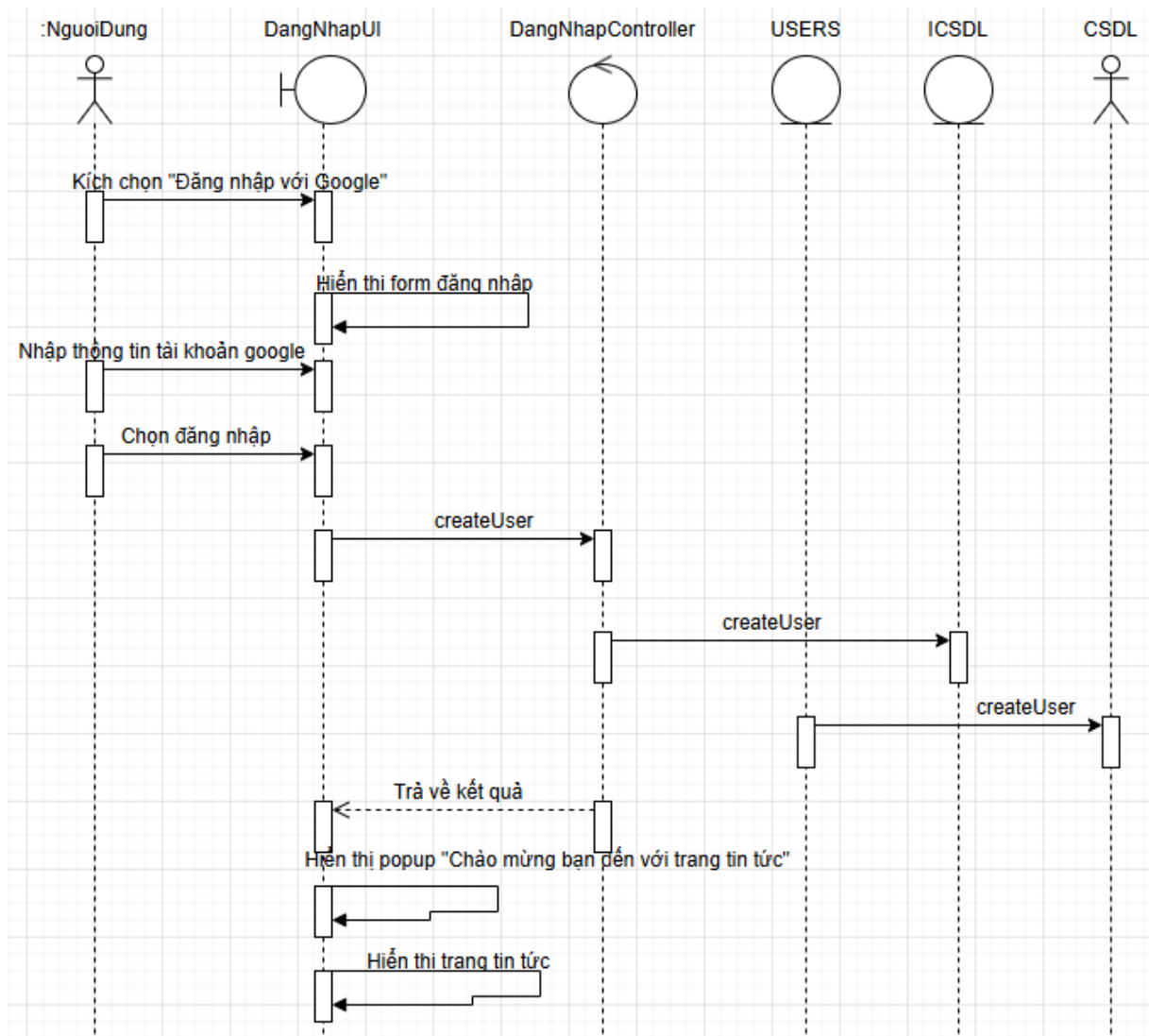
### **Điểm mở rộng**

Không có.

### **Hậu điều kiện**

Không có.

### **b. Biểu đồ trình tự**



Hình 2.8: Biểu đồ trình tự use case Đăng nhập (phía quản trị)

#### 2.2.4.7. Mô tả use case Xem bài viết theo danh mục

##### a. Mô tả use case

**Mô tả vắn tắt:** Use case này cho phép người dùng xem bài viết theo danh mục.

##### Luồng sự kiện

Luồng cơ bản

1. Use case này bắt đầu khi người dùng nhấn vào danh mục muốn xem ở trang tin tức từ thông tin bảng POSTS, bảng POST\_CATALOGUE\_POST, bảng LANGUAGES, bảng POST\_CATALOGUE\_LANGUAGE, bảng POST\_CATALOGUES, bảng USERS và sau đó hiển thị lên màn hình các thông tin: tên danh mục, tiêu đề bài viết, mô tả ngắn, ngày đăng, tác giả, số lượt xem và số lượng bình luận lên màn hình. Use case kết thúc.

Luồng rẽ nhánh

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với cơ sở dữ liệu thì hệ thống hiển thị thông báo lỗi. Use case kết thúc.
2. Tại bước 1 trong luồng cơ bản nếu trong danh mục chưa có bài viết nào hệ thống sẽ hiển thị thông báo “Chúng tôi sẽ cập nhật thêm tin tức trong thời gian sắp tới”. Use case kết thúc.

**Các yêu cầu đặc biệt**

Không có.

**Tiền điều kiện**

Không có.

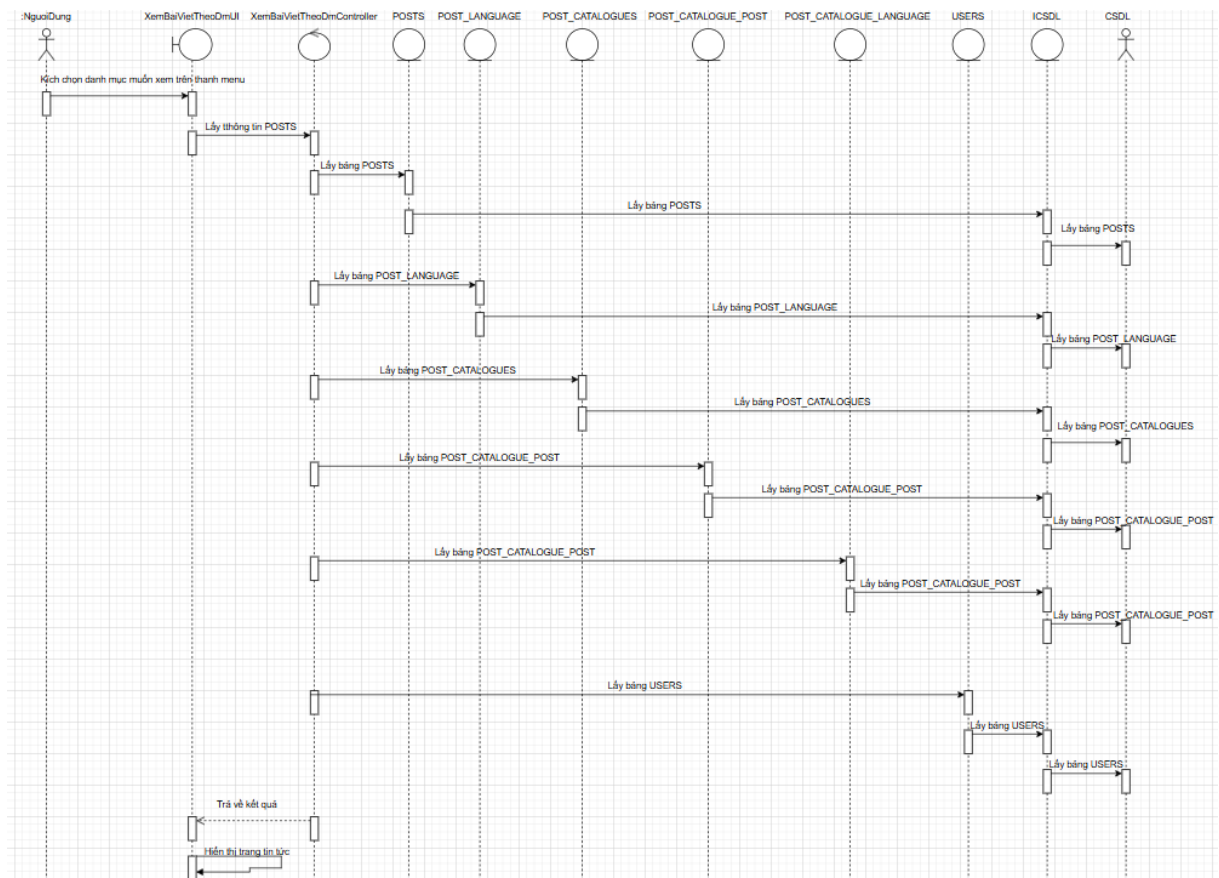
**Điểm mở rộng**

Không có.

**Hậu điều kiện**

Không có.

**b. Biểu đồ trình tự**



Hình 2.9: Biểu đồ trình tự use case Xem bài viết theo danh mục

#### 2.2.4.8. Mô tả use case Xem chi tiết bài viết

##### a. Mô tả use case

**Mô tả vắn tắt:** Use case này cho phép người dùng xem chi tiết thông tin của bài viết.

##### Luồng sự kiện

##### Luồng cơ bản

1. Use case này bắt đầu khi người dùng chọn một danh mục sau đó chọn vào một bài viết lấy từ các bảng POSTS, bảng POST\_CATALOGUE\_POST, bảng LANGUAGES, bảng POST\_CATALOGUE\_LANGUAGE, bảng POST\_CATALOGUES, bảng USERS trong cơ sở dữ liệu và hiển thị lên màn hình các thông tin: danh mục, ngày đăng, tiêu đề, mô tả ngắn, nội dung, tác giả. Use case kết thúc.

##### Luồng rẽ nhánh



1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với cơ sở dữ liệu thì hệ thống hiển thị thông báo lỗi. Use case kết thúc.

### Các yêu cầu đặc biệt

Không có.

### Tiền điều kiện

Không có.

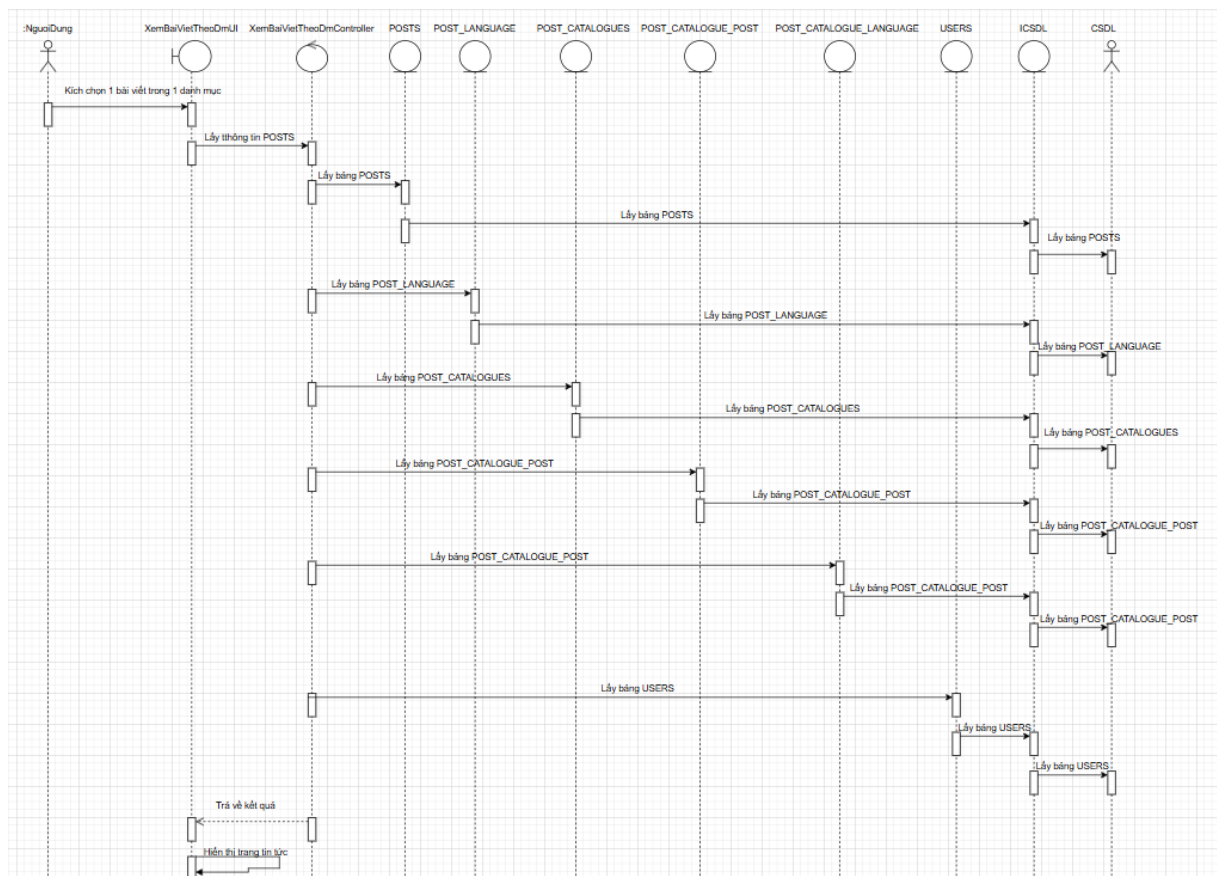
### Điểm mở rộng

Không có.

### Hậu điều kiện

Không có.

## b. Biểu đồ trình tự



Hình 2.10: Biểu đồ trình tự use case Xem chi tiết bài viết

## 2.2.4.9. Mô tả use case Bình luận về bài viết

### a. Mô tả use case

**Mô tả vắn tắt:** Use case này cho phép người dùng bình luận về bài viết.

### **Luồng sự kiện**

Luồng cơ bản

1. Use case này bắt đầu khi người dùng viết bình luận ở phần bình luận cuối bài viết và sau đó kích vào nút gửi. Hệ thống sẽ thêm mới bình luận vào bảng COMMENTS và hiển thị lên màn hình sau đó hiển thị popup thông báo “Bình luận của bạn đã được ghi nhận”. Use case kết thúc.

2. Xoá bình luận

a. Người dùng kích chuột vào nút “Xoá” để xoá bình luận của mình. Hệ thống sẽ xoá bình luận của bạn trong bảng COMMENTS và hiển thị lên màn hình popup thông báo “Xoá bình luận thành công”. Use case kết thúc.

Luồng rẽ nhánh

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với cơ sở dữ liệu thì hệ thống hiển thị thông báo lỗi. Use case kết thúc.

### **Các yêu cầu đặc biệt**

Không có.

### **Tiền điều kiện**

Người dùng cần đăng nhập để thực hiện chức năng này.

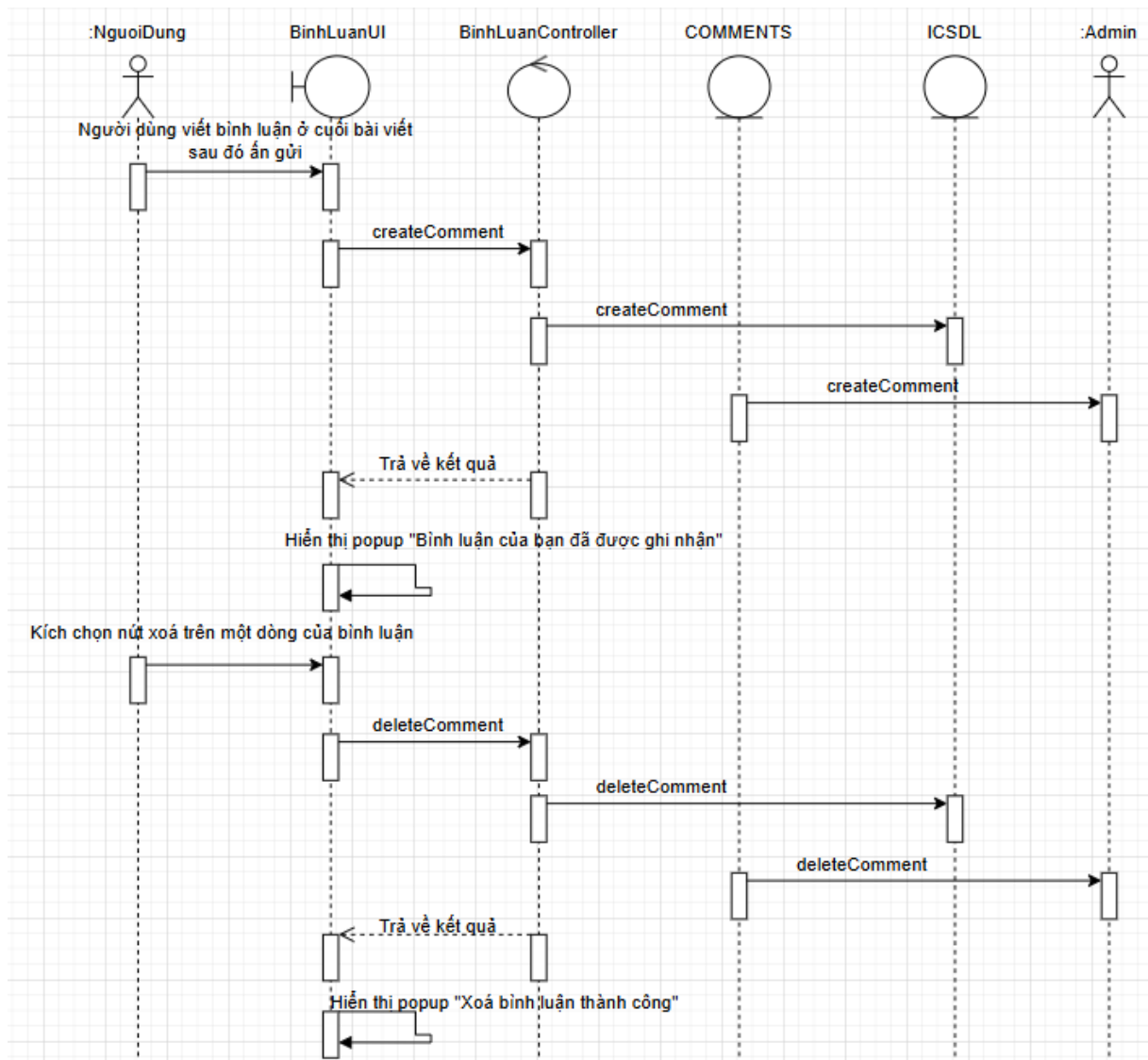
### **Điểm mở rộng**

Không có.

### **Hậu điều kiện**

Không có.

### **b. Biểu đồ trình tự**



Hình 2.11: Biểu đồ trình tự use case Bình luận về bài viết

#### 2.2.4.10. Mô tả use case Lưu bài viết

**a. Mô tả use case:** Use case này cho phép người dùng lưu tin tức để xem sau.

##### Mô tả vắn tắt

##### Luồng sự kiện

##### Luồng cơ bản

1. Use case bắt đầu khi người dùng ấn chuột vào nút “Lưu tin tức” ở cuối bài viết. Hệ thống sẽ lưu thông tin của bài viết vào bảng SAVE\_POSTS gồm các thông tin: id, user\_id, post\_id. Sau đó hiển thị popup “Tin tức đã được lưu”. Use case kết thúc.

2. Bỏ lưu bài viết

a. Use case bắt đầu khi người dùng kích chuột chọn thông tin cá nhân. Hệ thống sẽ hiển thị các bài viết đã lưu. Người dùng kích chọn “Bỏ lưu tin tức” ở dưới phần mô tả ngắn của mỗi bài viết để bỏ lưu 1 bài viết. Hệ thống sẽ xóa 1 bản ghi trong bảng SAVE\_POSTS. Use case kết thúc.

#### **Luồng rẽ nhánh**

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với cơ sở dữ liệu thì hệ thống hiển thị thông báo lỗi. Use case kết thúc.
2. Ở bước 2 của luồng cơ bản nếu người dùng chưa lưu tin tức nào sẽ hiển thị thông báo “Bạn chưa lưu tin tức nào”.

#### **Các yêu cầu đặc biệt**

Không có.

#### **Tiền điều kiện**

Người dùng cần đăng nhập để thực hiện chức năng này.

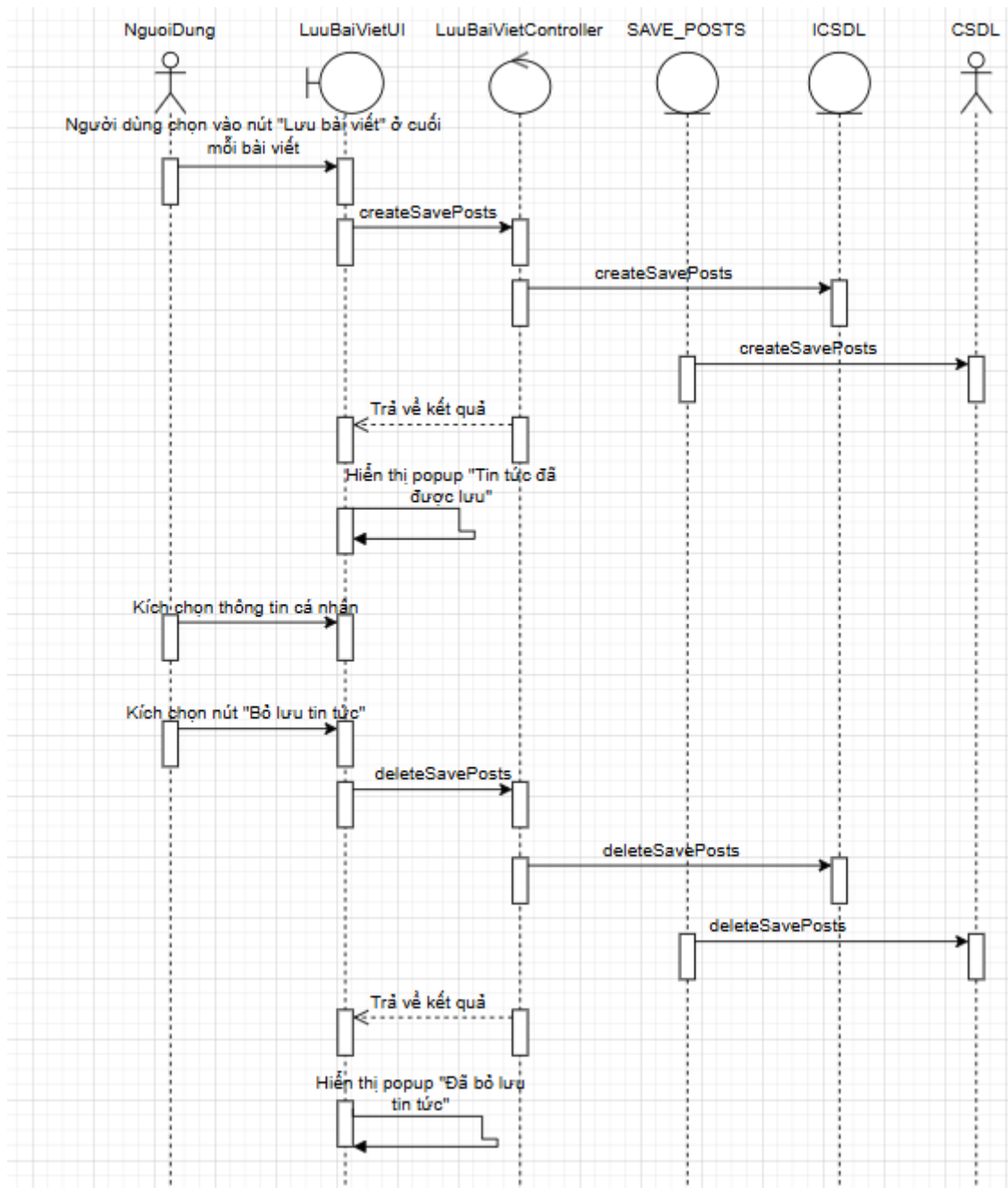
#### **Điểm mở rộng**

Không có.

#### **Hậu điều kiện**

Không có.

#### **b. Biểu đồ trình tự**



Hình 2.12: Biểu đồ trình tự use case Lưu bài viết

## 2.3. Xây dựng cơ sở dữ liệu

### 2.3.1. Biểu đồ Entity Relationship Diagram

Biểu đồ mô tả cấu trúc cơ sở dữ liệu được trình bày chi tiết ở Hình 2.13.



<b>name</b>	varchar (191)	DEFAULT NULL	Tên của người dùng
<b>phone</b>	varchar (20)	DEFAULT NULL	Số điện thoại của người dùng
<b>province_id</b>	varchar (10)	DEFAULT NULL	Địa chỉ cấp tỉnh/thành phố của người dùng
<b>district_id</b>	varchar (10)	DEFAULT NULL	Địa chỉ cấp quận/huyện của người dùng
<b>ward_id</b>	varchar (10)	DEFAULT NULL	Địa chỉ cấp phường/xã của người dùng
<b>address</b>	varchar(191)	DEFAULT NULL	Địa chỉ cụ thể của người dùng
<b>birthday</b>	datetime	DEFAULT NULL	Ngày sinh của người dùng
<b>description</b>	varchar(191)	DEFAULT NULL	Ghi chú
<b>email</b>	varchar(191)	NOT NULL	Email của người dùng

<b>password</b>	varchar(191)	NOT NULL	Mật khẩu của người dùng
<b>user_catalogue_id</b>	bigint (20)	DEFAULT 2	Id của nhóm thành viên
<b>created_at</b>	timestamp	DEFAULT NULL	Thời gian tạo user
<b>updated_at</b>	timestamp	DEFAULT NULL	Thời gian cập nhật user
<b>deleted_at</b>	timestamp	DEFAULT NULL	Thời gian xóa user

*Bảng 2.3: Chi tiết bảng PROVINCES*

<b>Tên cột</b>	<b>Kiểu dữ liệu</b>	<b>Phương thức lưu trữ</b>	<b>Mô tả</b>
<b>code</b>	varchar (20)	NOT NULL	Mã của tỉnh/thành phố
<b>name</b>	varchar(255)	NOT NULL	Tên tiếng việt của tỉnh/thành phố
<b>name_en</b>	varchar(255)	DEFAULT NULL	Tên tiếng anh của tỉnh/thành phố
<b>full_name</b>	varchar(255)	NOT NULL	Tên đầy đủ tiếng việt của tỉnh/thành phố



<b>full_name_en</b>	varchar(255)	DEFAULT NULL	Tên đầy đủ tiếng anh của tỉnh/thành phố
<b>code_name</b>	varchar(255)	DEFAULT NULL	Mã code của tỉnh/thành phố

*Bảng 2.4: Chi tiết bảng DISTRICTS*

<b>Tên cột</b>	<b>Kiểu dữ liệu</b>	<b>Phương thức lưu trữ</b>	<b>Mô tả</b>
<b>code</b>	varchar (20)	NOT NULL AUTO_INCREMENT	Mã của quận/huyện
<b>name</b>	varchar (255)	NOT NULL	Tên tiếng việt của quận/huyện
<b>name_en</b>	varchar (255)	DEFAULT NULL	Tên tiếng anh của quận/huyện
<b>full_name</b>	varchar (255)	DEFAULT NULL	Tên tiếng việt đầy đủ của quận/huyện
<b>full_name_en</b>	varchar (255)	DEFAULT NULL	Tên tiếng anh đầy đủ của quận/huyện
<b>code_name</b>	varchar (255)	DEFAULT NULL	Mã code của quận/huyện
<b>province_code</b>	varchar (20)	DEFAULT NULL	Mã của tỉnh/thành phố

*Bảng 2.5: Chi tiết bảng WARDS*

<b>Tên cột</b>	<b>Kiểu dữ liệu</b>	<b>Phương thức lưu trữ</b>	<b>Mô tả</b>
<b>code</b>	varchar (20)	NOT NULL AUTO_INCREMENT	Mã của phường/huyện
<b>name</b>	varchar (255)	NOT NULL	Tên tiếng việt của phường/huyện
<b>name_en</b>	varchar (255)	DEFAULT NULL	Tên tiếng anh của phường/huyện
<b>full_name</b>	varchar (255)	DEFAULT NULL	Tên tiếng việt đầy đủ của phường/huyện
<b>full_name_en</b>	varchar (255)	DEFAULT NULL	Tên tiếng anh đầy đủ của phường/huyện
<b>code_name</b>	varchar (255)	DEFAULT NULL	Mã code của phường
<b>district_code</b>	varchar (20)	DEFAULT NULL	Mã của quận huyện

*Bảng 2.6: Chi tiết bảng USER\_CATALOGUE*

<b>Tên cột</b>	<b>Kiểu dữ liệu</b>	<b>Phương thức lưu trữ</b>	<b>Mô tả</b>
<b>id</b>	bigint (20)	NOT NULL AUTO_INCREMENT	Mã của nhóm thành viên

<b>name</b>	varchar (191)	NOT NULL	Tên của nhóm thành viên
<b>description</b>	text	DEFAULT NULL	Mô tả nhóm thành viên
<b>deleted_at</b>	timestamp	DEFAULT NULL	Thời gian xóa của nhóm thành viên
<b>created_at</b>	timestamp	DEFAULT NULL	Thời gian tạo của nhóm thành viên
<b>updated_at</b>	timestamp	DEFAULT NULL	Thời gian cập nhật của nhóm thành viên
<b>publish</b>	tinyint (4)	DEFAULT 0	Tình trạng của nhóm thành viên

*Bảng 2.7: Chi tiết bảng USER\_CATALOGUE\_PERMISSION*

<b>Tên cột</b>	<b>Kiểu dữ liệu</b>	<b>Phương thức lưu trữ</b>	<b>Mô tả</b>
<b>user_catalogue_id</b>	bigint (20)	NOT NULL	id của nhóm thành viên
<b>permission_id</b>	bigint (20)	NOT NULL	id của quyền

*Bảng 2.8: Chi tiết bảng PERMISSIONS*

Tên cột	Kiểu dữ liệu	Phương thức lưu trữ	Mô tả
<b>id</b>	bigint (20)	NOT NULL AUTO_INCREMENT	id của quyền
<b>name</b>	varchar (191)	NOT NULL	Tên của quyền
<b>canonical</b>	varchar (191)	NOT NULL	Đường dẫn của quyền
<b>created_at</b>	timestamp	DEFAULT NULL	Thời gian tạo quyền
<b>updated_at</b>	timestamp	DEFAULT NULL	Thời gian cập nhật quyền

*Bảng 2.9: Chi tiết bảng SAVE\_POSTS*

Tên cột	Kiểu dữ liệu	Phương thức lưu trữ	Mô tả
<b>id</b>	bigint (20)	NOT NULL AUTO_INCREMENT	Mã của bài viết đã lưu
<b>user_id</b>	bigint (20)	NOT NULL	id của người dùng
<b>post_id</b>	bigint (20)	NOT NULL	id của bài viết
<b>created_at</b>	timestamp	NOT NULL	Thời gian lưu bài
<b>updated_at</b>	timestamp	NOT NULL	Thời gian cập nhật bài viết

*Bảng 2.10: Chi tiết bảng COMMENTS*

Tên cột	Kiểu dữ liệu	Phương thức lưu trữ	Mô tả
<b>id</b>	bigint (20)	NOT NULL AUTO_INCREMENT	Mã của bình luận
<b>user_id</b>	bigint (20)	NOT NULL	Mã của người dùng đã bình luận
<b>post_id</b>	bigint (20)	NOT NULL	Mã của bài viết
<b>content</b>	varchar (191)	NOT NULL	Nội dung bình luận
<b>created_at</b>	timestamp	NOT NULL	Thời gian bình luận
<b>updated_at</b>	timestamp	NOT NULL	Thời gian cập nhật bình luận

*Bảng 2.11: Chi tiết bảng POSTS*

Tên cột	Kiểu dữ liệu	Phương thức lưu trữ	Mô tả
<b>id</b>	bigint (20)	NOT NULL AUTO_INCREMENT	Mã của bài viết
<b>post_catalogue_id</b>	int (11)	DEFAULT 0	Mã của nhóm bài viết
<b>image</b>	varchar (191)	DEFAULT NULL	Ảnh của bài viết
<b>publish</b>	tinyint(4)	DEFAULT 1	Tình trạng bài viết

<b>user_id</b>	bigint(20)	NOT NULL	Mã của người đăng bài
<b>deleted_at</b>	timestamp	DEFAULT NULL	Thời gian xóa bài
<b>created_at</b>	timestamp	DEFAULT NULL	Thời gian tạo bài
<b>updated_at</b>	timestamp	DEFAULT NULL	Thời gian cập nhật bài
<b>n_views</b>	bigint(20)	DEFAULT 0	Số lượng lượt xem của bài viết
<b>n_comments</b>	bigint(20)	DEFAULT 0	Số lượng bình luận của bài viết

*Bảng 2.12: Chi tiết bảng POST\_LANGGUAGE*

<b>Tên cột</b>	<b>Kiểu dữ liệu</b>	<b>Phương thức lưu trữ</b>	<b>Mô tả</b>
<b>post_id</b>	bigint (20)	NOT NULL	Mã của bài viết
<b>language_id</b>	bigint (20)	NOT NULL	Mã của ngôn ngữ
<b>name</b>	varchar (191)	NOT NULL	Tiêu đề của bài viết
<b>description</b>	text	DEFAULT NULL	Mô tả ngắn của bài viết
<b>content</b>	longtext	NOT NULL	Nội dung bài viết

<b>created_at</b>	timestamp	DEFAULT NULL	Thời gian tạo bài viết
<b>updated_at</b>	timestamp	DEFAULT NULL	Thời gian cập nhật bài viết

*Bảng 2.13: Chi tiết bảng POST\_CATALOGUES*

<b>Tên cột</b>	<b>Kiểu dữ liệu</b>	<b>Phương thức lưu trữ</b>	<b>Mô tả</b>
<b>id</b>	bigint (20)	NOT NULL AUTO_INCREMENT	Mã của danh mục
<b>parentid</b>	int (11)	DEFAULT 0	Mã của danh mục cha
<b>lft</b>	int (11)	DEFAULT 0	Node trong cây thư mục
<b>rgt</b>	int (11)	DEFAULT 0	Node trong cây thư mục
<b>level</b>	int (11)	DEFAULT 0	Cấp của danh mục
<b>publish</b>	tinyint(4)	DEFAULT 1	Tình trạng của danh mục
<b>user_id</b>	bigint (20)	NOT NULL	Mã của user
<b>deleted_at</b>	timestamp	DEFAULT NULL	Ngày xóa của thư mục
<b>created_at</b>	timestamp	DEFAULT NULL	Ngày tạo thư mục
<b>updated_at</b>	timestamp	DEFAULT NULL	Ngày cập nhật thư mục

*Bảng 2.14: Chi tiết bảng POST\_CATALOGUES\_POST*

<b>Tên cột</b>	<b>Kiểu dữ liệu</b>	<b>Phương thức lưu trữ</b>	<b>Mô tả</b>
----------------	---------------------	----------------------------	--------------

<b>post_catalogue_id</b>	bigint (20)	NOT NULL	Mã của danh mục
<b>post_id</b>	bigint (20)	NOT NULL	Mã của bài viết

*Bảng 2.15: Chi tiết bảng POST\_CATALOGUE\_LANGUAGE*

Tên cột	Kiểu dữ liệu	Phương thức lưu trữ	Mô tả
<b>post_catalogue_id</b>	bigint (20)	NOT NULL	Mã của danh mục
<b>language_id</b>	bigint (20)	NOT NULL	Mã của ngôn ngữ
<b>name</b>	varchar (191)	NOT NULL	Tên danh mục
<b>description</b>	Text	DEFAULT NULL	Mô tả ngắn danh mục
<b>created_at</b>	timestamp	DEFAULT NULL	Thời gian tạo danh mục
<b>updated_at</b>	timestamp	DEFAULT NULL	Thời gian sửa danh mục

*Bảng 2.16: Chi tiết bảng LANGUAGES*

Tên cột	Kiểu dữ liệu	Phương thức lưu trữ	Mô tả
<b>id</b>	bigint (20)	NOT NULL AUTO_INCREMENT	Mã của ngôn ngữ
<b>name</b>	varchar (50)	NOT NULL	Tên ngôn ngữ
<b>user_id</b>	bigint (20)	NOT NULL	Mã của user

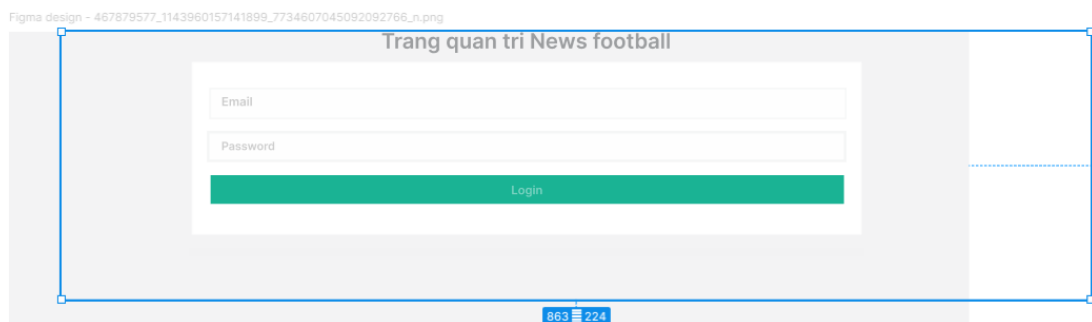


<b>publish</b>	tinyint (4)	DEFAULT NULL	Tình trạng ngôn ngữ
<b>current</b>	tinyint (4)	DEFAULT NULL	Ngôn ngữ hiện tại

## 2.4. Thiết kế màn hình

### 2.4.1. Màn hình đăng nhập (phía quản trị)

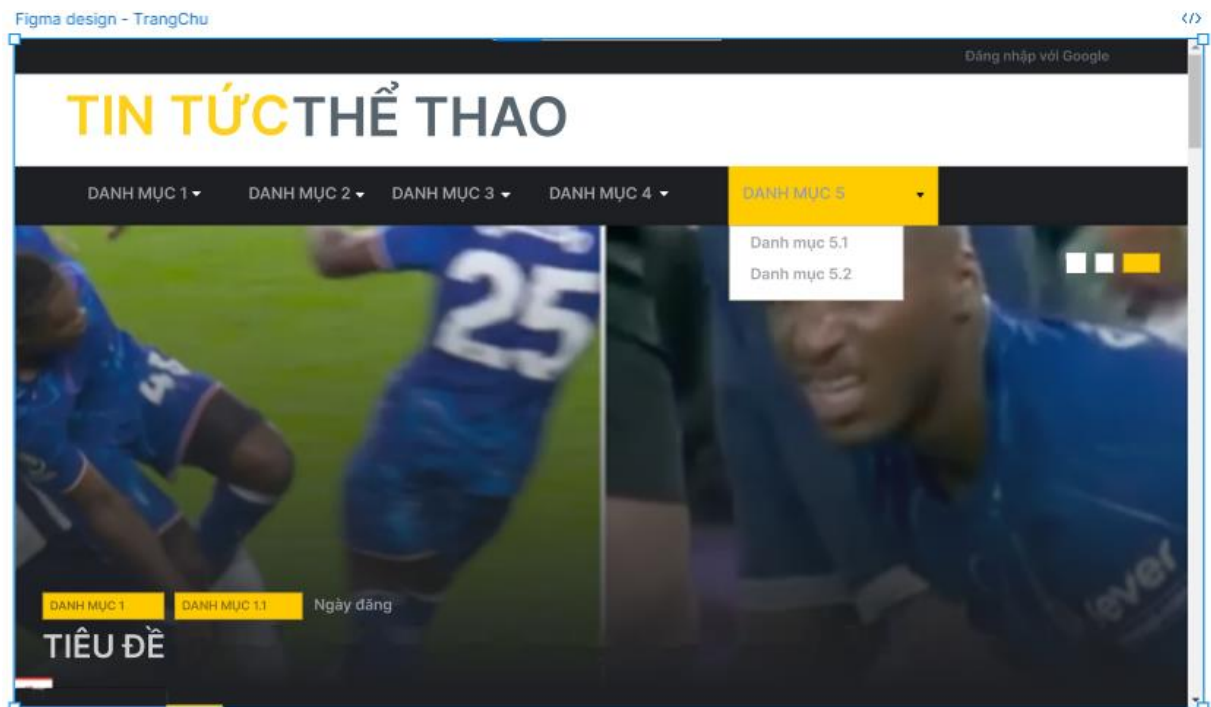
Màn hình đăng nhập phía quản trị được thiết kế như Hình 2.14.

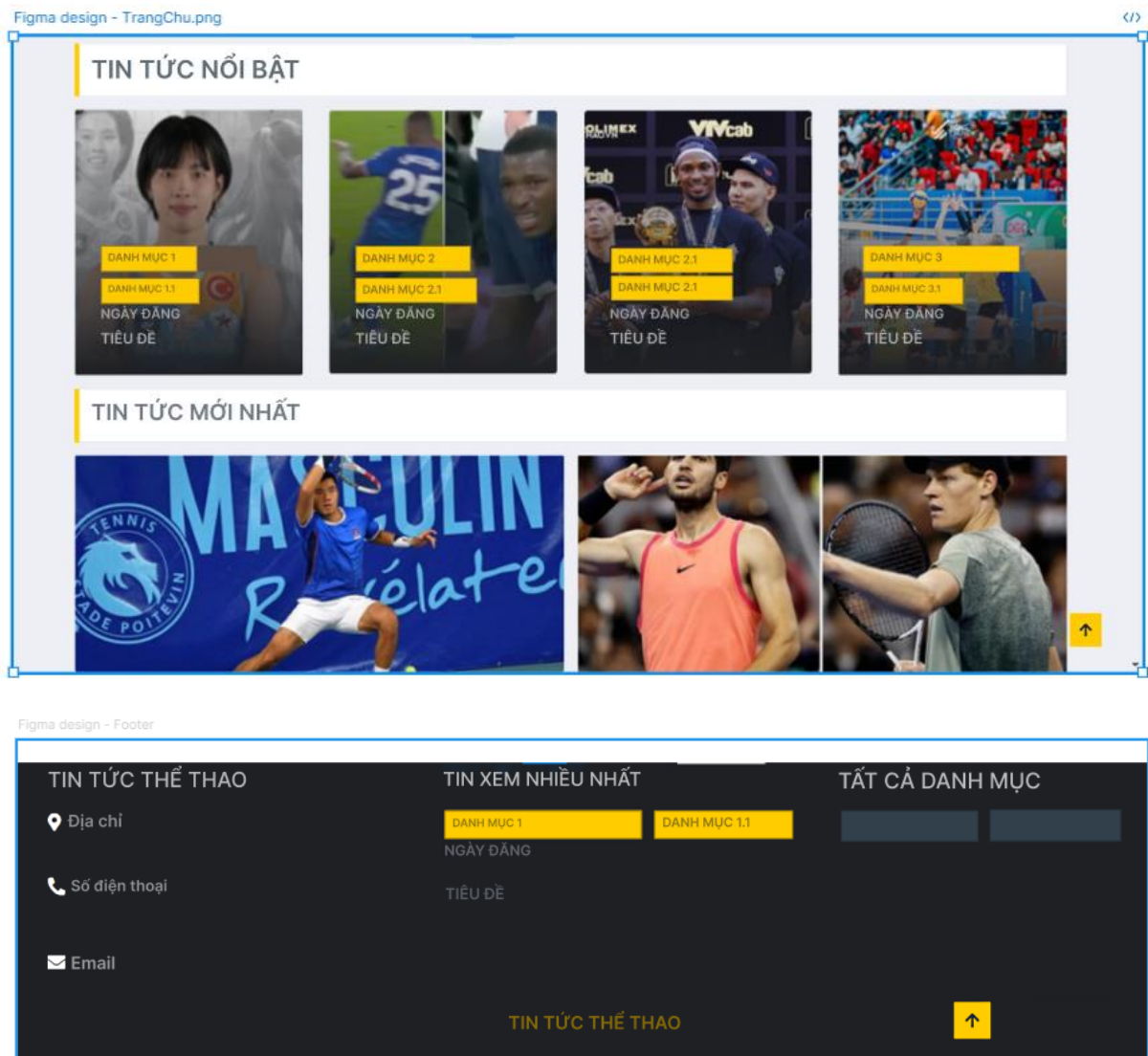


Hình 2.14: Thiết kế màn hình Đăng nhập (phía quản trị)

### 2.4.2. Màn hình Trang chủ

Màn hình trang chủ được thiết kế như Hình 2.15.

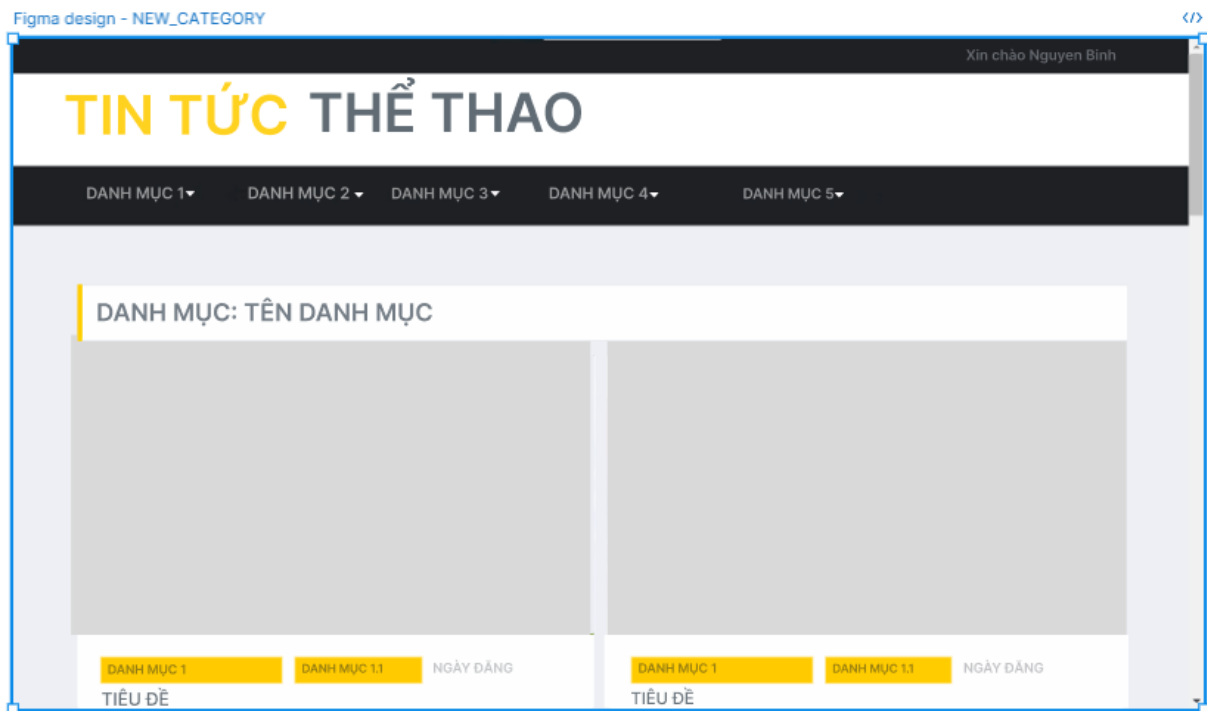




Hình 2.15: Thiết kế màn hình Trang chủ

### 2.4.3. Màn hình Xem bài viết theo danh mục

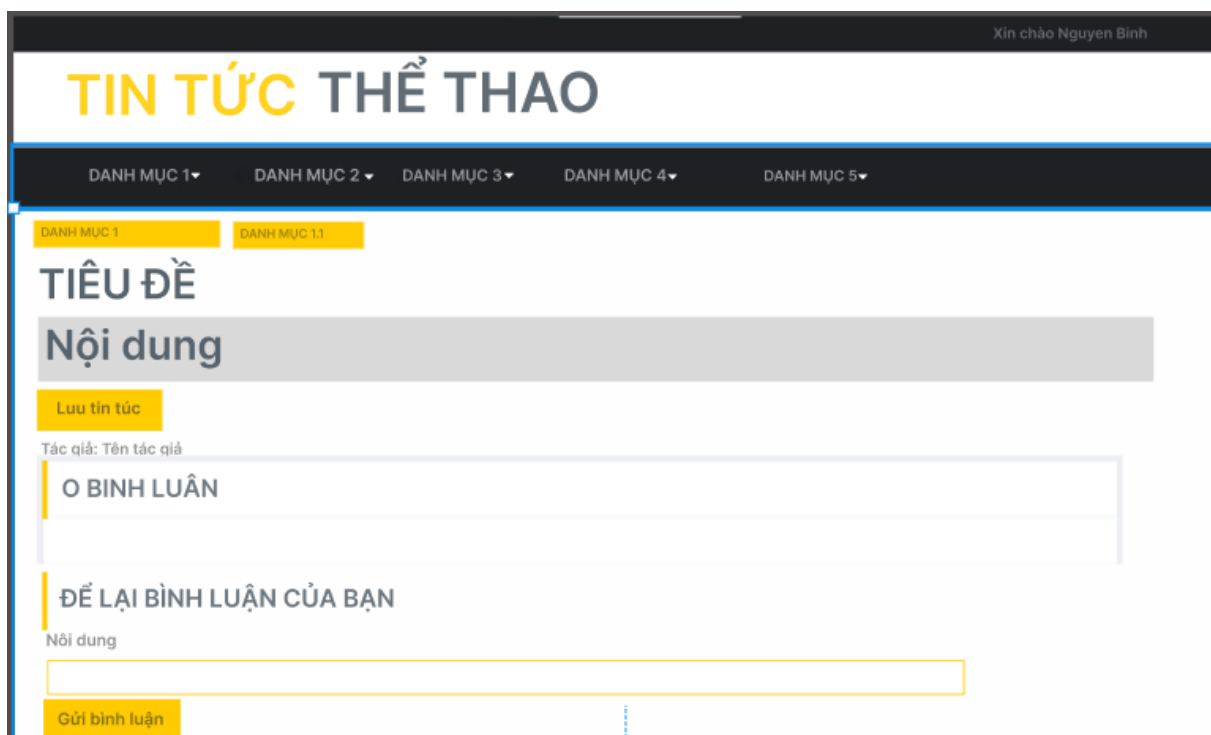
Màn hình xem bài viết theo danh mục được thiết kế như Hình 2.16.



Hình 2.16: Thiết kế màn hình Xem bài viết theo danh mục

#### 2.4.4. Màn hình Xem chi tiết bài viết

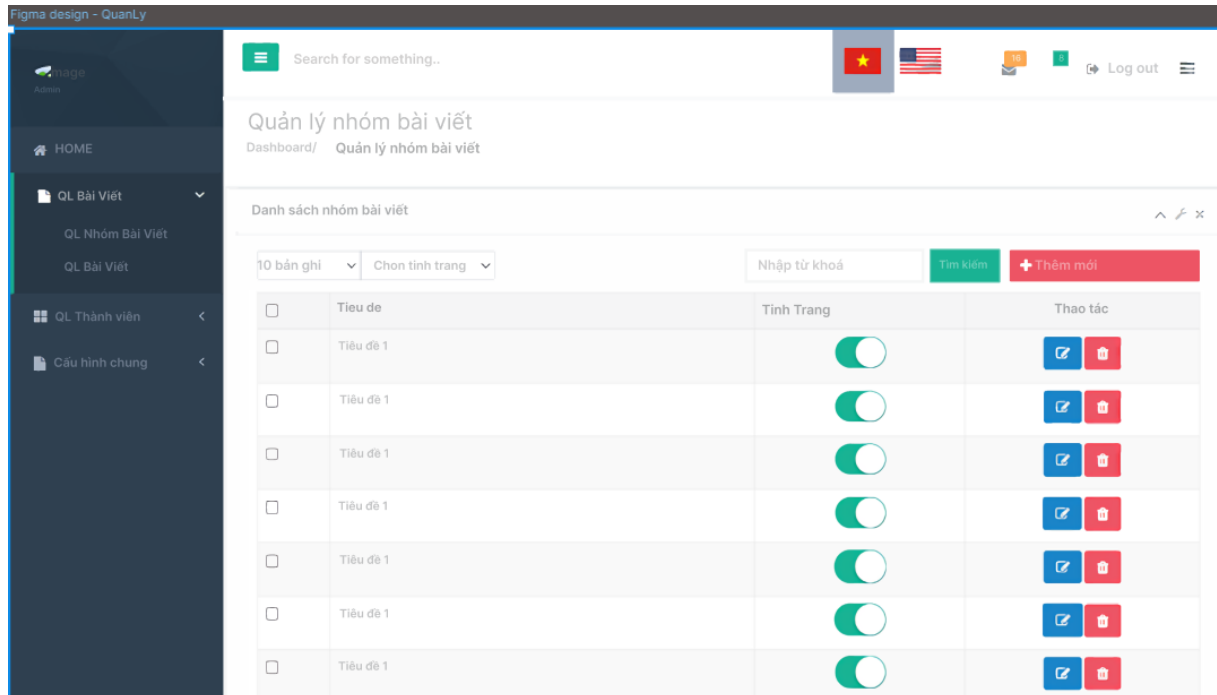
Màn hình xem chi tiết bài viết được thiết kế như Hình 2.17.



Hình 2.17: Thiết kế màn hình Xem chi tiết bài viết

### 2.4.5. Màn hình Quản lý bài viết

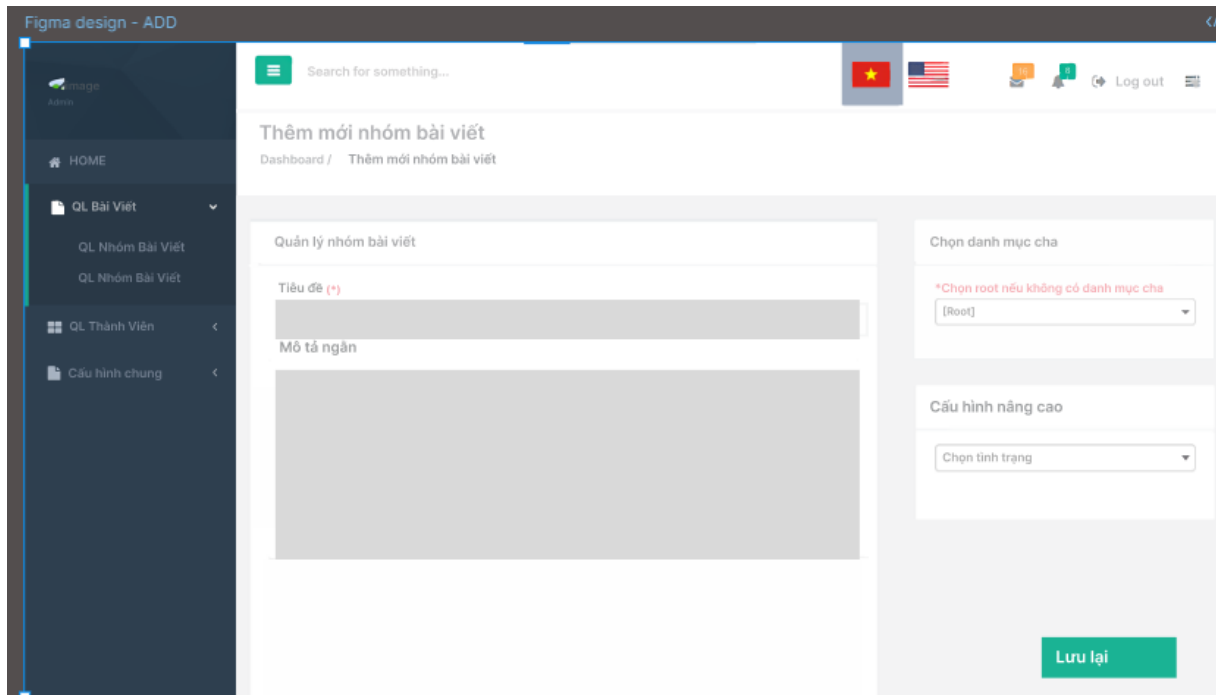
Màn hình quản lý bài viết được thiết kế như Hình 2.18 và các màn hình quản lý nhóm bài viết, quản lý nhóm thành viên, quản lý thành viên được thiết kế tương tự chỉ thay đổi phần tiêu đề.



Hình 2.18: Thiết kế màn hình Quản lý bài viết

### 2.4.6. Màn hình Thêm mới nhóm bài viết

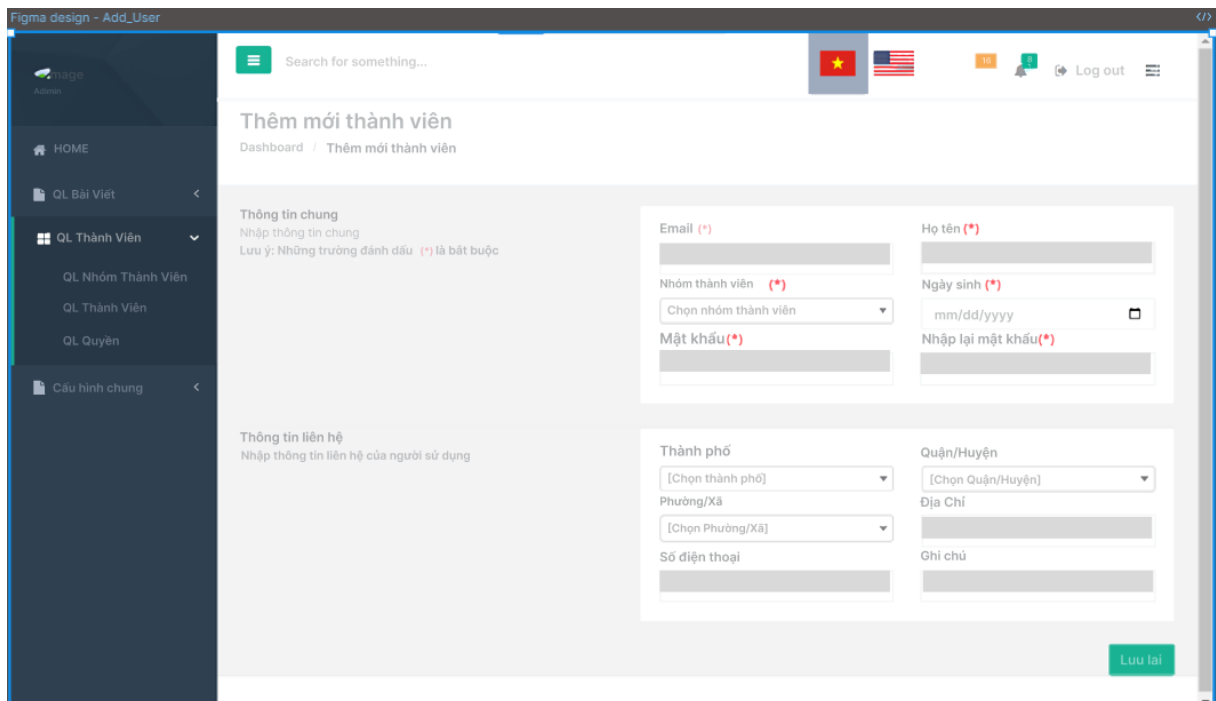
Màn hình thêm mới nhóm bài viết được thiết kế như Hình 2.19 và các màn hình thêm mới bài viết, sửa bài viết được thiết kế tương tự chỉ thay đổi phần tiêu đề.



Hình 2.19: Thiết kế màn hình Thêm mới nhóm bài viết

#### 2.4.7. Màn hình Thêm mới thành viên

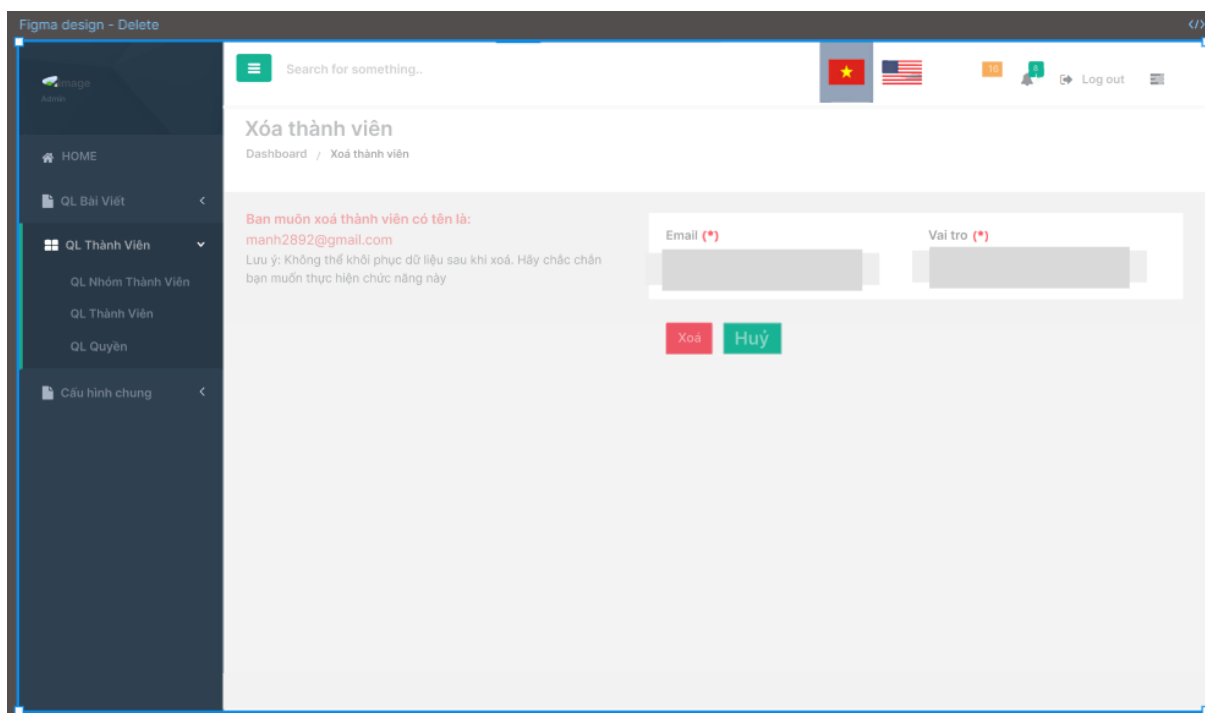
Màn hình thêm mới thành viên được thiết kế như Hình 2.20 và màn hình thêm mới nhóm thành viên được thiết kế tương tự.



Hình 2.20: Thiết kế màn hình Thêm mới thành viên

## 2.4.8. Màn hình xoá thành viên

Màn hình xoá thành viên được thiết kế như Hình 2.21 và các màn hình xoá bài viết, xoá nhóm bài viết, xoá nhóm thành viên được thiết kế tương tự.



Hình 2.21: Thiết kế màn hình Xoá thành viên

## 2.5. Kết luận chương 2

Trong chương này, em đã thực hiện phân tích chi tiết các tác nhân tham gia vào hệ thống, qua đó làm rõ vai trò và nhiệm vụ của từng đối tượng. Biểu đồ use case tổng quát giúp hình dung rõ nét các chức năng chính mà hệ thống cần đáp ứng. Bên cạnh đó, việc đặc tả chi tiết từng use case hỗ trợ xác định cụ thể các bước thực hiện, đảm bảo tính chính xác và toàn vẹn trong quá trình phát triển.

Ngoài ra, em cũng đã tiến hành thiết kế giao diện màn hình, tập trung vào việc đảm bảo tính trực quan và thân thiện với người dùng. Các màn hình được thiết kế bao gồm các yếu tố cần thiết để đáp ứng các chức năng chính của hệ thống, đồng thời thể hiện rõ luồng tương tác giữa người dùng và hệ thống.

Cuối cùng, phần mô tả cơ sở dữ liệu cung cấp một cái nhìn tổng thể về cấu trúc thông tin cần quản lý, tạo nền tảng vững chắc cho việc triển khai hệ thống ở các chương tiếp theo.

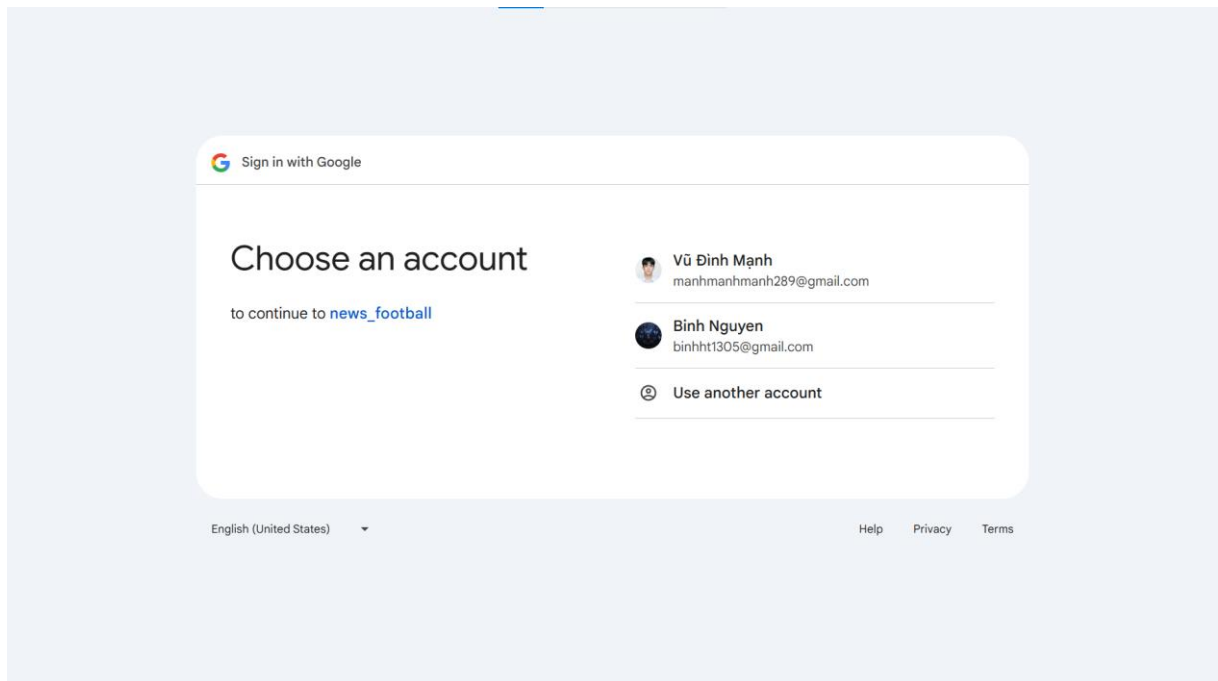
## CHƯƠNG 3. KẾT QUẢ THỰC HIỆN VÀ KIỂM THỬ

### 3.1. Kết quả thực hiện

Kết quả xây dựng website tin tức thể thao sử dụng PHP được thể hiện thông qua giao diện các chức năng được trình bày chi tiết ở các đề mục thông qua các hình vẽ bên dưới.

#### 3.1.1. Giao diện đăng nhập (phía người dùng)

Giao diện đăng nhập (phía người dùng) được thể hiện chi tiết trong Hình 3.22.



Hình 3.22: Giao diện đăng nhập (phía người dùng)

#### 3.1.2. Giao diện trang chủ

Giao diện trang chủ được thể hiện chi tiết trong Hình 3.23.

# TIN TỨC THỂ THAO

BÓNG ĐÁ QUỐC TẾ ▾

BÓNG RỔ ▾

BÓNG CHUYỀN ▾

BÓNG ĐÁ VIỆT NAM ▾

MÔN THỂ THAO KHÁC ▾

Pickleball

Tennis - Quần vợt

NGOẠI HẠNG ANH

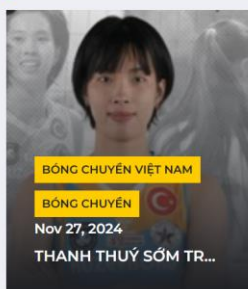
BÓNG ĐÁ QUỐC TẾ

Dec 10, 2024

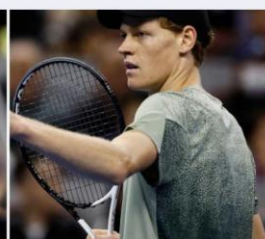
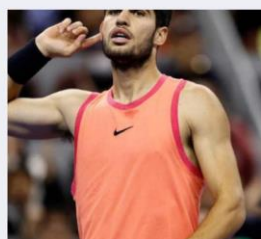
## CAICEDO THOÁT THẺ ĐỎ, BTC NGOẠI HẠNG ANH NÓI GÌ?

127.0.0.1:8000/categories/90

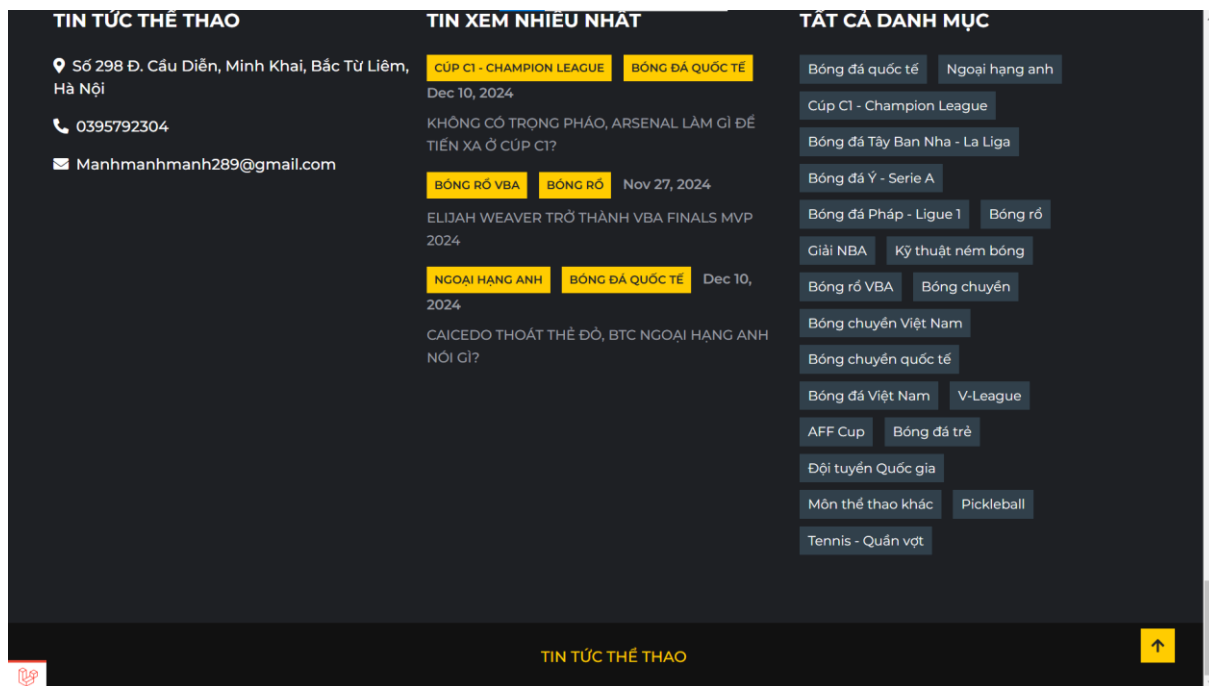
### TIN TỨC NỔI BẬT



### TIN TỨC MỚI NHẤT



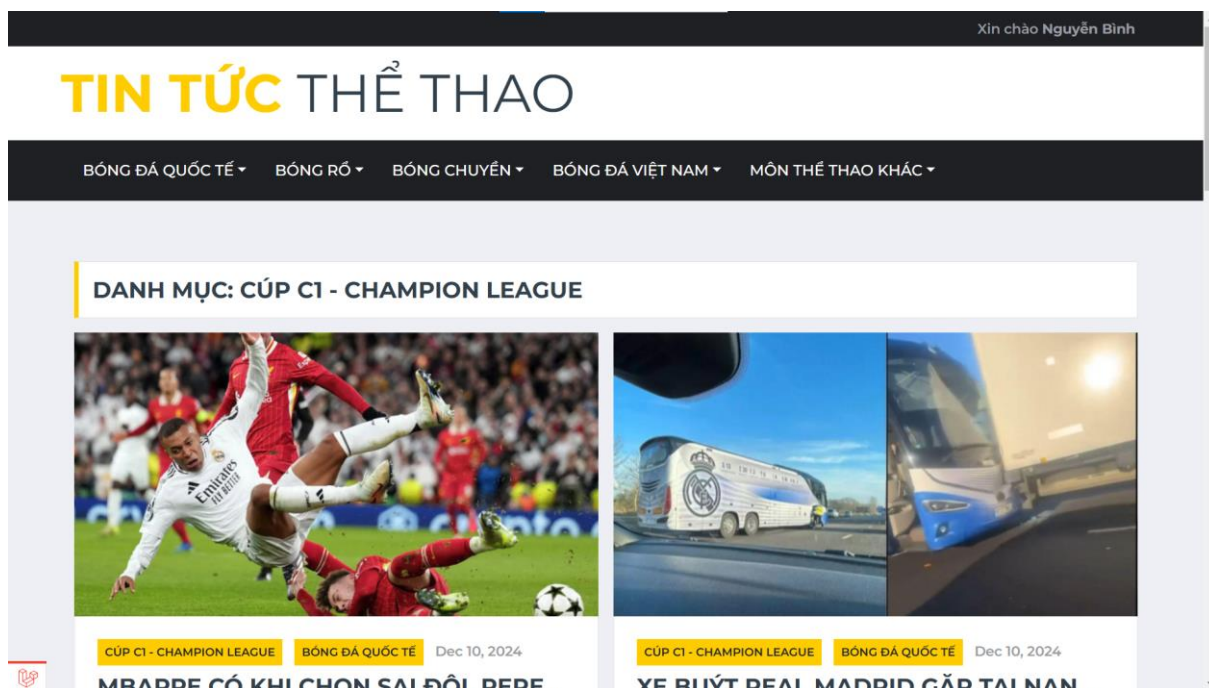




Hình 3.23: Giao diện trang chủ

### 3.1.3. Giao diện xem bài viết theo danh mục

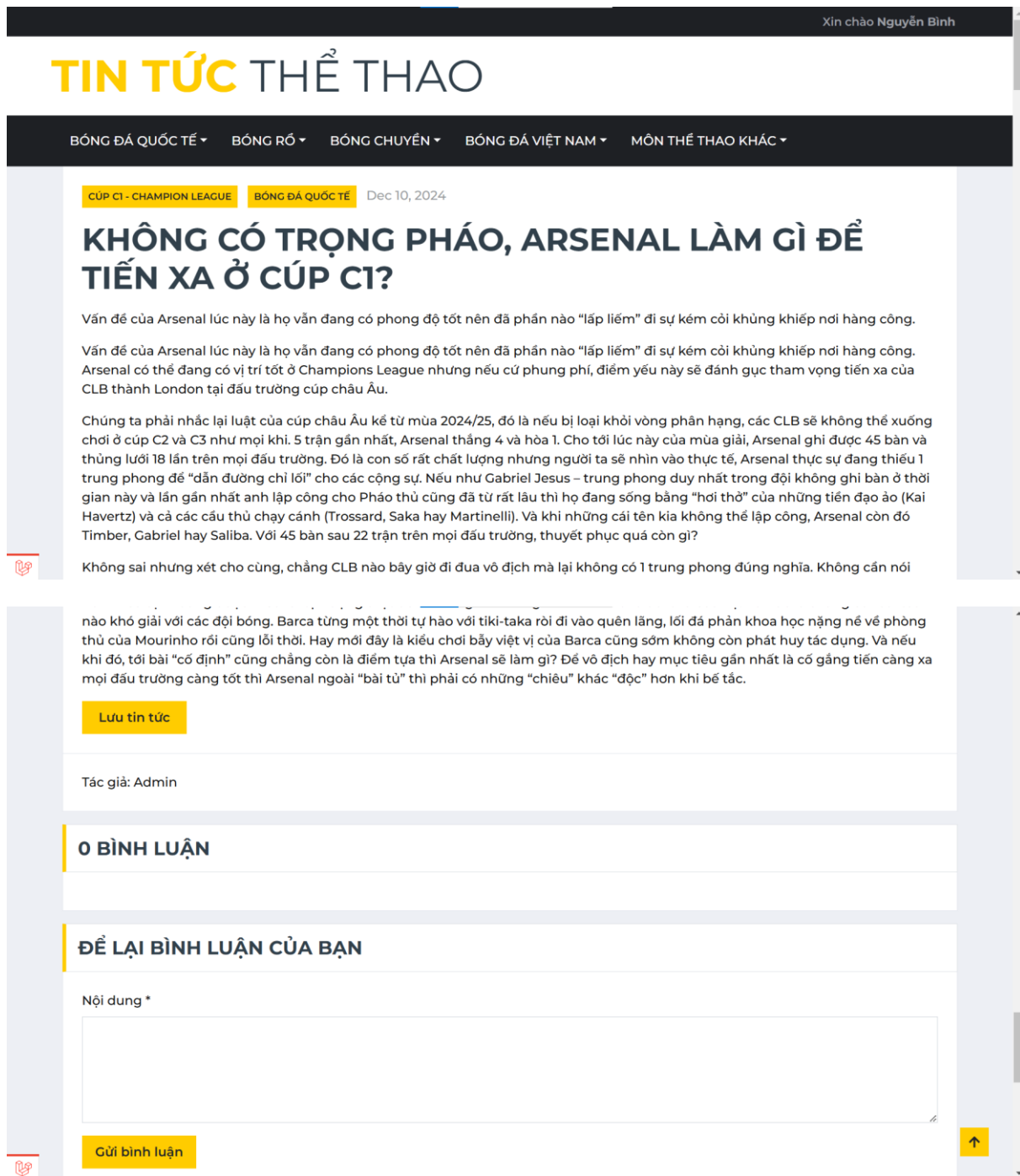
Giao diện xem bài viết theo danh mục được thể hiện chi tiết trong Hình 3.24.



Hình 3.24: Giao diện xem bài viết theo danh mục

### 3.1.4. Giao diện xem chi tiết bài viết

Giao diện xem chi tiết bài viết được thể hiện chi tiết trong Hình 3.25.



Hình 3.25: Giao diện xem chi tiết bài viết

### 3.1.5. Giao diện thông tin cá nhân

Giao diện thông tin cá nhân được thể hiện chi tiết trong Hình 3.26.

The screenshot shows a web interface for a sports news website. At the top, there is a dark header with the text "Xin chào Nguyễn Bình" on the right. Below this is a large yellow and black logo that reads "TIN TỨC THỂ THAO". Underneath the logo is a dark navigation bar with several menu items: "BÓNG ĐÁ QUỐC TẾ", "BÓNG RỔ", "BÓNG CHUYỀN", "BÓNG ĐÁ VIỆT NAM", and "MÔN THỂ THAO KHÁC". The main content area has a light gray background. It features a section titled "THÔNG TIN CÁ NHÂN" in bold black text. Below this title are four input fields arranged in a 2x2 grid. The first field is labeled "Tên" and contains the text "Nguyễn Bình". The second field is labeled "Số điện thoại" and is empty. The third field is labeled "Địa chỉ" and is empty. The fourth field is labeled "Ngày sinh" and contains the date "12/12/2001". Below these fields is a yellow button labeled "Cập nhật". Underneath the button is another section titled "TIN TỨC ĐÃ LƯU" in bold black text. Below this title is a small image showing two people, one of whom is a woman with dark hair.

Hình 3.26: Giao diện thông tin cá nhân

### 3.1.6. Giao diện đăng nhập (phía quản trị)

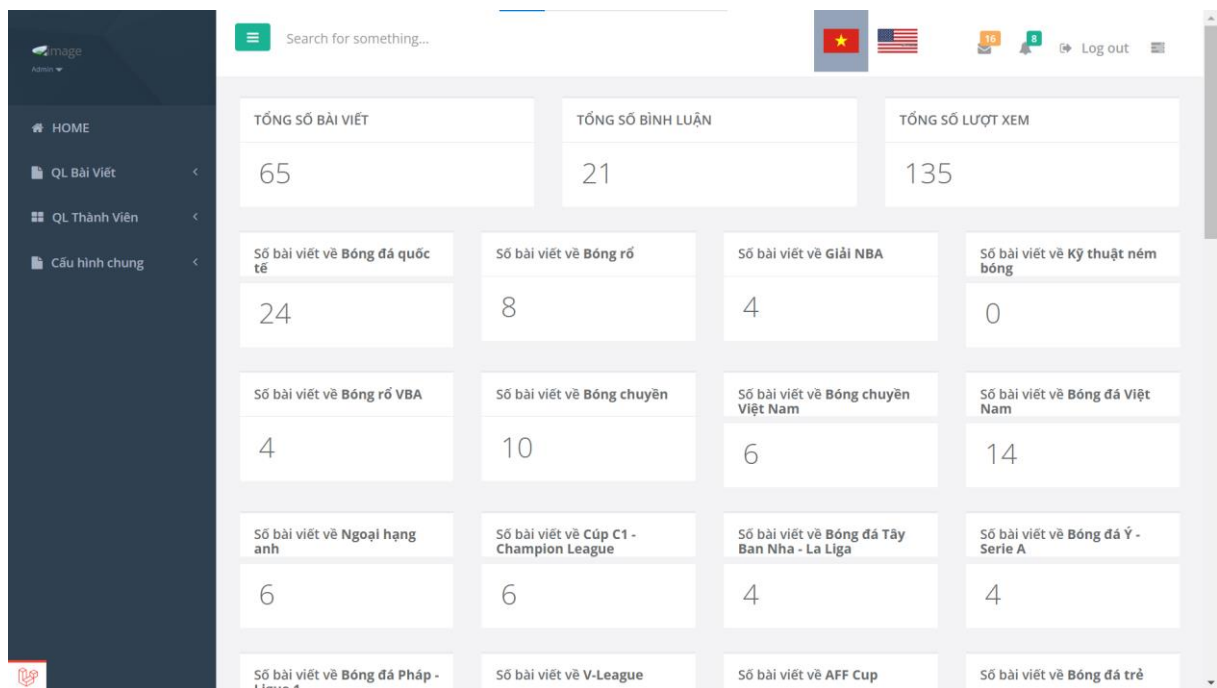
Giao diện đăng nhập (phía quản trị) được thể hiện chi tiết trong Hình 3.27.

The screenshot shows a login form for an admin interface. The form is titled "Trang quản trị News football" in a gray font. It contains two input fields: "Email" and "Password". Below these fields is a green button labeled "Login". The form is set against a light gray background.

Hình 3.27: Giao diện đăng nhập (phía quản trị)

### 3.1.7. Giao diện Dashboard

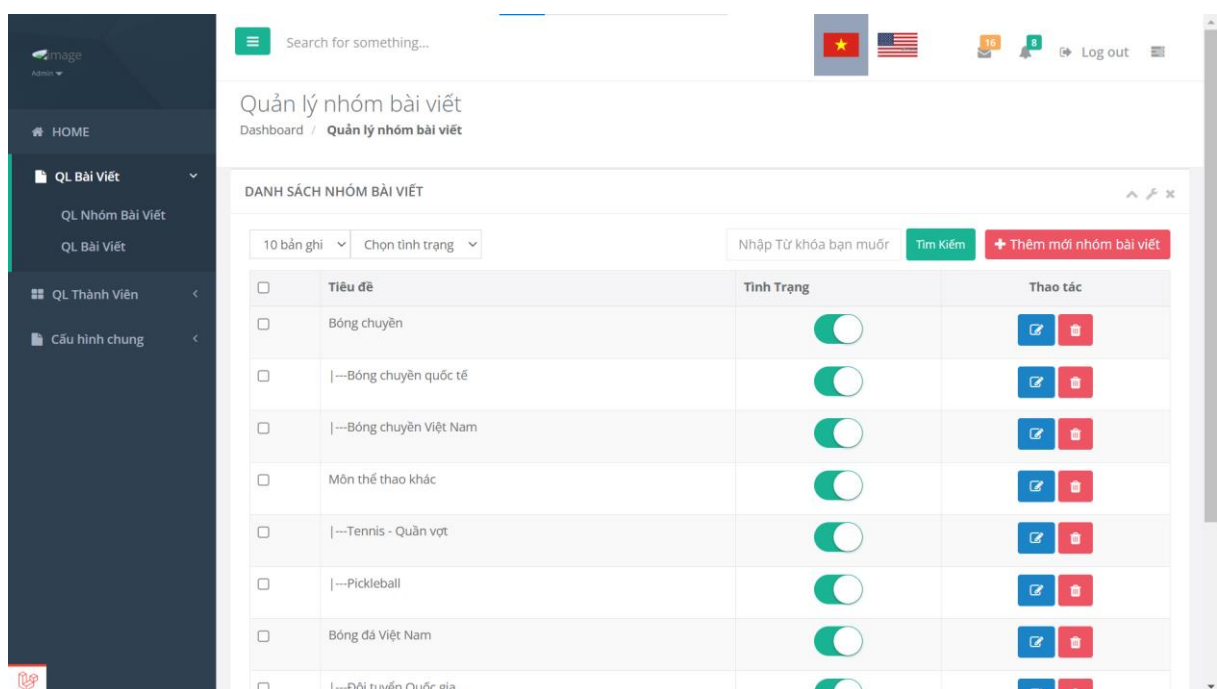
Giao diện Dashboard được thể hiện chi tiết trong hình 3.28.



Hình 3.28: Giao diện Dashboard

### 3.1.8. Giao diện quản lý nhóm bài viết

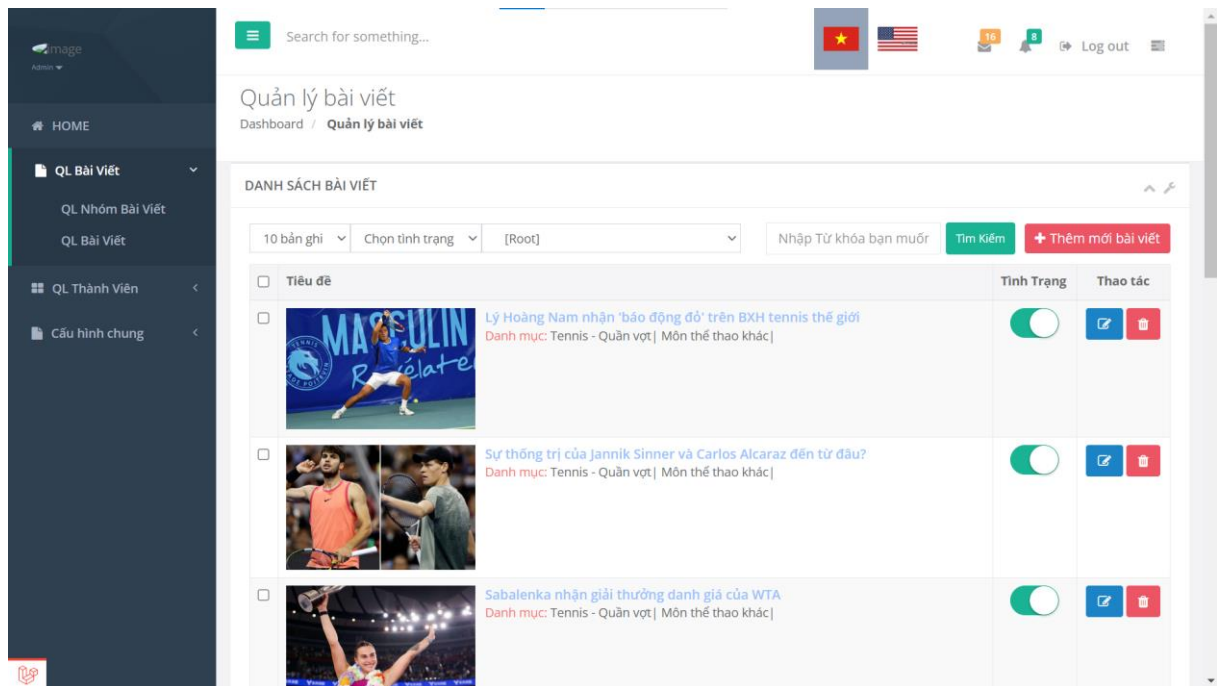
Giao diện quản lý nhóm bài viết được thể hiện trong Hình 3.29.



Hình 3.29: Giao diện quản lý nhóm bài viết

### 3.1.9. Giao diện quản lý bài viết

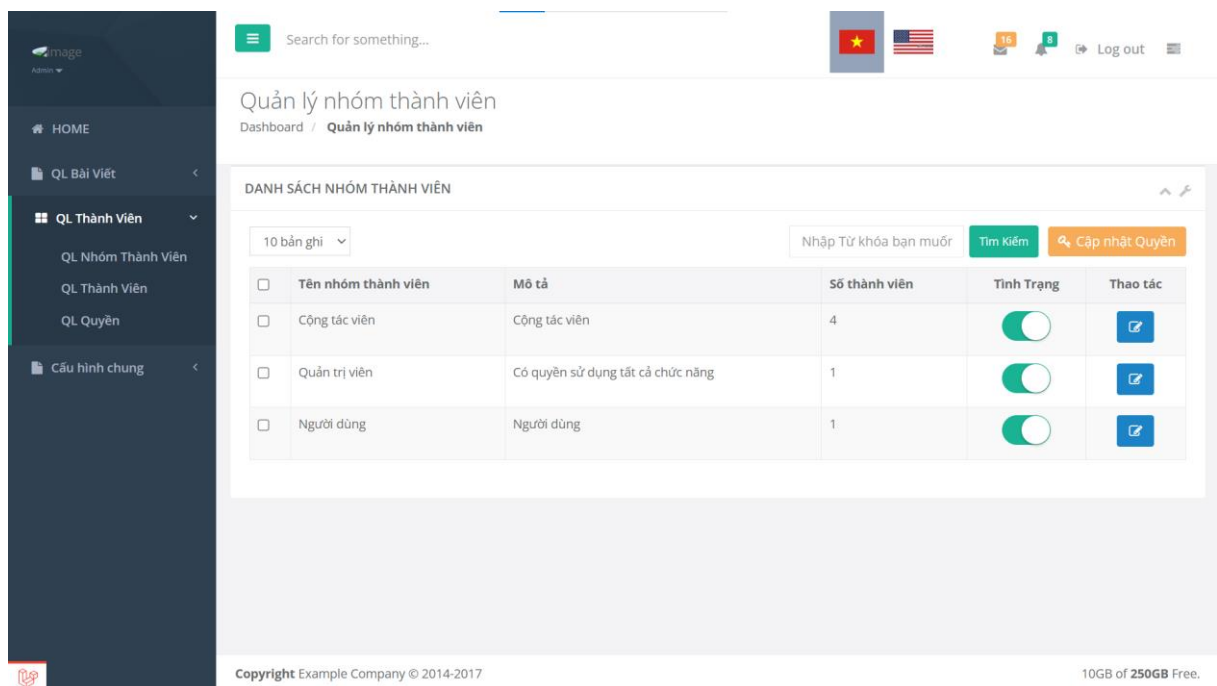
Giao diện quản lý bài viết được thể hiện trong Hình 3.30.

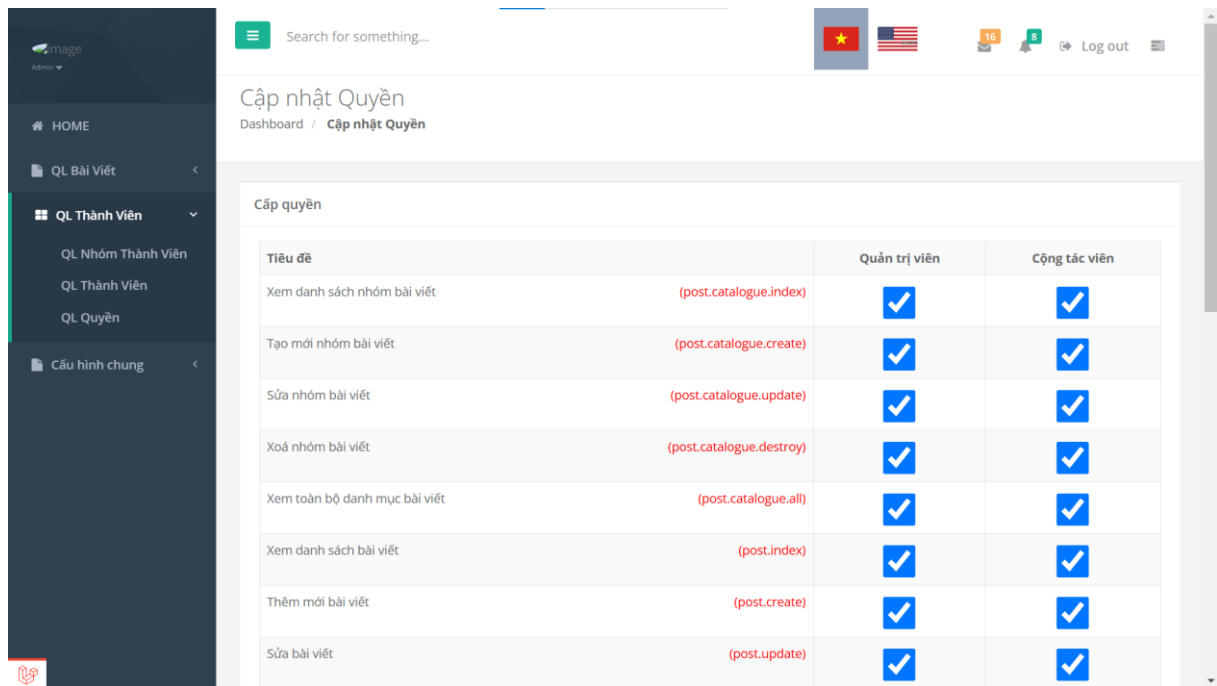


Hình 3.30: Giao diện quản lý bài viết

### 3.1.10. Giao diện quản lý nhóm thành viên

Giao diện quản lý nhóm thành viên được thể hiện trong Hình 3.31.

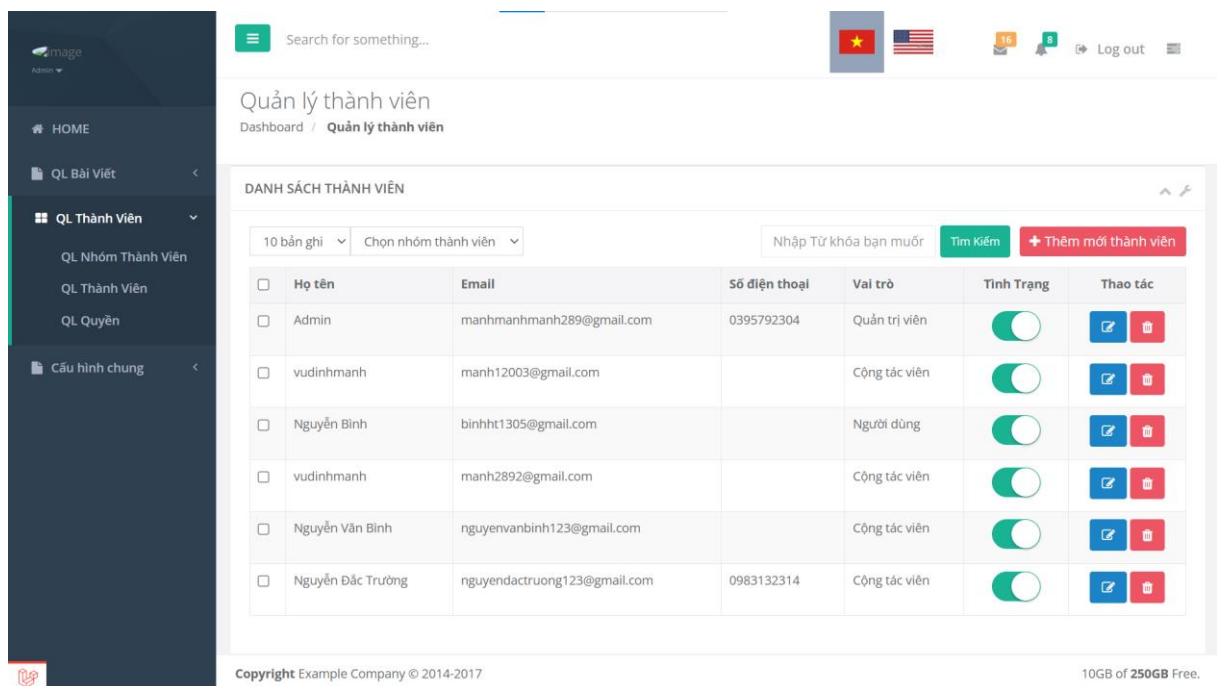




Hình 3.31: Giao diện quản lý nhóm thành viên

### 3.1.11. Giao diện quản lý thành viên

Giao diện quản lý thành viên được thể hiện trong Hình 3.32.



Hình 3.32: Giao diện quản lý thành viên

### 3.2. Kiểm thử hệ thống

#### 3.2.1. Kế hoạch kiểm thử

Thực hiện kiểm thử cho 11 chức năng

- Đăng nhập
- Quản lý nhóm bài viết
- Quản lý bài viết
- Quản lý nhóm thành viên
- Quản lý thành viên
- Xem bài viết theo danh mục
- Xem chi tiết bài viết
- Lưu tin tức
- Sửa thông tin cá nhân
- Bình luận
- Thay đổi ngôn ngữ

#### 3.2.2. Kiểm thử chức năng

Chi tiết các test case được mô tả theo Bảng 3.17.

*Bảng 3.17: Kiểm thử chức năng*

STT	Chức năng	Trường hợp kiểm thử	Đầu vào	Đầu ra thực tế	Kết quả
1	Đăng nhập	Đăng nhập (phía người dung).	Người dùng nhập tài khoản google của mình.	Hiển thị popup “Chào mừng bạn đến với tràn tin tức” sau đó chuyển hướng đến trang chủ.	Đạt

		Đăng nhập (phía quản trị)	Nhập dữ liệu hợp lệ.	Hiển thị popup “Đăng nhập thành công” và chuyển hướng đến trang quản trị.	Đạt
			Nhập dữ liệu không hợp lệ.	Hiển thị popup “Email hoặc mật khẩu không đúng”.	Đạt
2	Quản lý nhóm bài viết	Thêm nhóm bài viết	Nhập dữ liệu hợp lệ.	Hiển thị popup “Thêm thành công” sau đó hiển thị lại danh sách nhóm bài viết.	Đạt
			Nhập dữ liệu không hợp lệ.	Hiển thị thông báo và yêu cầu nhập lại.	Đạt
		Sửa nhóm bài viết	Click chuột vào icon sửa màu xanh.	Hiển thị popup “Sửa thành công” sau đó hiển thị lại danh sách nhóm bài viết.	Đạt
		Xoá nhóm bài viết	Click chuột vào icon thùng rác màu đỏ.	Hiển thị popup “Xoá thành công” sau đó hiển thị lại danh sách nhóm bài viết.	Đạt
3	Quản lý bài viết	Thêm bài viết	Nhập dữ liệu hợp lệ.	Hiển thị popup “Thêm thành công”	Đạt



				sau đó hiển thị lại danh sách bài viết.	
			Nhập dữ liệu không hợp lệ.	Hiển thị thông báo và yêu cầu nhập lại.	Đạt
		Sửa bài viết	Click chuột vào icon sửa màu xanh.	Hiển thị popup “Sửa thành công” sau đó hiển thị lại danh sách bài viết.	Đạt
		Xoá bài viết	Click chuột vào icon thùng rác màu đỏ.	Hiển thị popup “Xoá thành công” sau đó hiển thị lại danh sách bài viết.	Đạt
4	Quản lý nhóm thành viên	Cập nhật quyền	Click chuột vào nút “Cập nhật quyền”.	Hiển thị popup “Cập nhật quyền thành công” sau đó hiển thị lại danh sách nhóm thành viên.	Đạt
5	Quản lý thành viên	Thêm thành viên	Nhập dữ liệu hợp lệ.	Hiển thị popup “Thêm thành công” sau đó hiển thị lại danh sách thành viên.	Đạt
			Nhập dữ liệu không hợp lệ.	Hiển thị thông báo và yêu cầu nhập lại.	Đạt

		Sửa thành viên	Click chuột vào icon sửa màu xanh.	Hiển thị popup “Sửa thành công” sau đó hiển thị lại danh sách thành viên.	Đạt
		Xoá thành viên	Click chuột vào icon thùng rác màu đỏ.	Hiển thị popup “Xoá thành công” sau đó hiển thị lại danh sách thành viên.	Đạt
<b>6</b>	Xem bài viết theo danh mục	Xem bài viết theo danh mục	Người dùng chọn 1 danh mục bất kì.	Hiển thị tất cả bài viết theo danh mục vừa chọn.	Đạt
<b>7</b>	Xem chi tiết bài viết	Xem chi tiết bài viết	Người dùng chọn 1 bài viết bất kì.	Hiển thị chi tiết bài viết vừa chọn.	Đạt
<b>8</b>	Lưu tin tức	Lưu tin tức	Người dùng chọn nút lưu bài viết ở cuối mỗi bài viết.	Hiển thị chi tiết bài viết vừa chọn.	Đạt
<b>9</b>	Sửa thông tin cá nhân	Sửa thông tin cá nhân	Người dùng chọn vào nút thông tin cá nhân, sau đó điền thông tin muốn sửa.	Hiển thị popup “Sửa thành công” và quay lại trang tin tức.	Đạt

<b>10</b>	Bình luận	Bình luận	Người dùng viết nội dung muốn bình luận sau đó chọn nút “Gửi bình luận”	Hiện thị popup “Bình luận của bạn đã được ghi nhận”.	Đạt
<b>11</b>	Thay đổi ngôn ngữ	Thay đổi ngôn ngữ	Người quản trị nhấn chọn hình lá cờ để thay đổi ngôn ngữ	Ngôn ngữ ở phía quản trị đã được thay đổi.	Chưa đạt

### 3.2.3. Kết quả kiểm thử

Sau khi thực hiện 22 test case thì có 21 test đạt và 1 test chưa đạt. Tỷ lệ test case đạt: 95.5% và tỷ lệ test case thất bại: 4.5%. Hệ thống chạy ổn định trên các trình duyệt web khác nhau như Firefox, Google Chrome, Microsoft Edge.

### 3.3. Kết luận chương 3

Trong chương này, em đã trình bày chi tiết về các kết quả đạt được sau khi hoàn thành dự án, bao gồm giao diện người dùng và phần kiểm thử hệ thống. Giao diện người dùng được thiết kế với tiêu chí thân thiện và dễ sử dụng, giúp người dùng dễ dàng thao tác và trải nghiệm các chức năng của hệ thống.

Phần kiểm thử hệ thống được thực hiện kỹ lưỡng, nhằm đảm bảo mọi chức năng hoạt động đúng yêu cầu và không phát sinh lỗi trong quá trình sử dụng. Kết quả kiểm thử cho thấy hệ thống đáp ứng tốt các tiêu chí về hiệu năng, độ ổn định và tính chính xác, tạo tiền đề vững chắc cho việc triển khai thực tế.

## KẾT LUẬN

### 1. Kết quả đạt được

Qua đề tài “Xây dựng website thể thao sử dụng PHP” dựa trên những kiến thức đã học được trên ghế nhà trường em đã xây dựng được một website hoàn chỉnh với đầy đủ các module chức năng. Qua đó hiểu được quy trình thiết kế và xây dựng một website hoàn chỉnh. Ngoài ra, qua quá trình thực hiện, em đã rèn luyện được khả năng giải quyết vấn đề, tư duy logic và kỹ năng làm việc độc lập. Em cũng đã hiểu rõ hơn về quy trình phát triển phần mềm, từ việc phân tích yêu cầu, thiết kế cơ sở dữ liệu, đến triển khai và kiểm thử sản phẩm. Tất cả những thành tựu này đã giúp em tích lũy được nhiều kinh nghiệm quý báu và tự tin hơn trong công việc phát triển phần mềm sau này.

### 2. Hướng phát triển trong tương lai

- Tích hợp thêm tính năng tương tác người dùng
- Triển khai hệ thống bình luận, đánh giá bài viết và chức năng chia sẻ trên các nền tảng mạng xã hội để tăng tính tương tác và thu hút người dùng.
- Phát triển hệ thống thông báo đẩy để người dùng có thể nhận được các tin tức thể thao nóng hổi và kịp thời.
- Cập nhật thêm các tính năng như live stream, video highlights các sự kiện thể thao để người dùng có thể theo dõi trực tiếp các trận đấu và sự kiện thể thao.
- Cung cấp các phân tích và thống kê chi tiết về các trận đấu, cầu thủ, đội tuyển giúp người dùng dễ dàng theo dõi và đưa ra các nhận định.

## TÀI LIỆU THAM KHẢO

- [1]. Trường Đại học Công nghiệp Hà Nội. Bài giảng điện tử, Nhập môn công nghệ phần mềm, Tổng quan về công nghệ phần mềm.
- [2]. Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Thị Kim Phượng, Giáo trình phân tích thiết kế hệ thống, NXB Giáo dục VN.
- [3]. Laravel Document. (Truy cập ngày: 18/12/2024).  
<https://laravel.com/docs/10.x>
- [4]. Javascript Document. (Truy cập ngày: 5/12/2024).  
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [5]. Vietnv - Đăng nhập bằng tài khoản google với laravel. (Truy cập ngày: 2/12/2024).  
<https://blog.haposoft.com/dang-nhap-tai-khoan-google-laravel/>