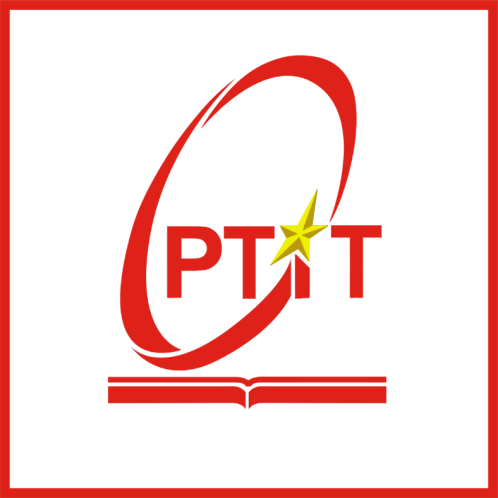
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO CUỐI KỲ**

**KỸ THUẬT LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Đề tài: “XÂY DỰNG ỨNG DỤNG BLOG ĐÁP ỨNG THỜI GIAN THỰC”**

**Giảng viên hướng dẫn : ThS. Huỳnh Trung Trụ**

|  |  |
| --- | --- |
| **Tên** | **MSSV** |
| **Trần Đăng Khoa** | **N18DCCN101** |
| **Nguyễn Phan Nhựt Trường** | **N20DCCN082** |
| **Lê Trọng Chiến** | **N20DCCN090** |

**Tp. Hồ Chí Minh - 12/2024**

**Mục lục**

[CHƯƠNG 1: CÔNG NGHỆ 1](#_Toc32407)

[1.1 Spring Boot 1](#_Toc4219)

[1.2. Spring Security và JWT 1](#_Toc21895)

[1.3. Spring Data JPA (Hibernate) 1](#_Toc3501)

[1.4. MySQL 2](#_Toc10506)

[1.5. ReactJS 2](#_Toc15987)

[1.6. Socket.IO 2](#_Toc28391)

[1.7. RabbitMQ 3](#_Toc21996)

[CHƯƠNG 2: KHẢO SÁT HIỆN TRẠNG 4](#_Toc6253)

[2.1 Hiện trạng thực tế 4](#_Toc2095)

[2.2 Giải pháp cho các vấn đề 4](#_Toc31053)

[2.3 Mô hình nghiệp vụ hệ thống 5](#_Toc29237)

[2.3.1 Chức năng từng bộ phận 5](#_Toc14950)

[2.3.2 Quy trình nghiệp vụ 5](#_Toc16904)

[2.3.2.1. Quy trình đăng ký và đăng nhập: 6](#_Toc24503)

[2.3.2.2. Quy trình quản lý bài đăng: 6](#_Toc10585)

[2.3.2.3. Quy trình quản lý bạn bè và theo dõi: 6](#_Toc25362)

[2.3.2.4. Quy trình nhắn tin thời gian thực: 7](#_Toc13814)

[2.3.2.5. Quy trình quản lý người dùng (dành cho admin): 7](#_Toc12622)

[2.3.2.6. Quy trình quản lý bài đăng và bình luận: 7](#_Toc18096)

[2.3.2.7. Quy trình tương tác với bài đăng: 8](#_Toc1531)

[2.4 Yêu cầu chức năng 8](#_Toc18611)

[2.4.1. Chức năng dành cho người dùng (User) 8](#_Toc16885)

[2.4.1.1. Quản lý tài khoản 8](#_Toc2142)

[2.4.1.2. Quản lý bài viết 8](#_Toc31556)

[2.4.1.3. Tương tác với bài viết 8](#_Toc31261)

[2.4.1.4. Kết bạn và tương tác xã hội 9](#_Toc14626)

[2.4.1.5. Nhắn tin thời gian thực 9](#_Toc17109)

[2.4.1.6. Tìm kiếm người dùng và bài viết 9](#_Toc9511)

[2.4.2. Chức năng dành cho quản trị viên (Admin) 9](#_Toc17629)

[2.4.2.1. Quản lý người dùng 9](#_Toc748)

[2.4.2.2. Quản lý bài đăng 9](#_Toc23964)

[2.4.2.3. Quản lý bình luận 10](#_Toc2466)

[2.4.2.4. Quản lý tag 10](#_Toc8518)

[2.4.2.5. Quản lý yêu cầu kết bạn 10](#_Toc22374)

[2.4.3. Chức năng bảo mật 10](#_Toc3406)

[2.4.4. Chức năng hiệu suất 10](#_Toc13420)

[2.5 Yêu cầu phi chức năng: 10](#_Toc12159)

[2.6 Yêu cầu hệ thống: 10](#_Toc19500)

[2.7. Mục đích nghiên cứu đề tài 11](#_Toc28389)

[2.8. Mục tiêu nghiên cứu đề tài 11](#_Toc29998)

[2.9. Phương pháp tiến hành 12](#_Toc19685)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 14](#_Toc9275)

[3.1 Sơ đồ phân cấp chức năng 14](#_Toc26680)

[3.1.1 Sơ đồ phân rã chức năng mức đỉnh 14](#_Toc19952)

[3.1.2 Sơ đồ phân rã chức năng 14](#_Toc6315)

[3.2 Mô hình luồng dữ liệu DFD 18](#_Toc20998)

[3.2.1 Biểu đồ DFD mức ngữ cảnh 18](#_Toc29123)

[3.2.2 Biểu đồ DFD phân rã cấp 0 (chức năng chính) 18](#_Toc9681)

[3.3 Mô hình Use Case 19](#_Toc11203)

[3.3.1 Xác định Actor 19](#_Toc24518)

[3.3.2 Xác định Use Case 19](#_Toc11687)

[3.3.3 Xây dựng Use Case tổng quát 20](#_Toc5116)

[3.3.3.1 Use case tổng quát của người dùng 20](#_Toc2050)

[3.3.3.2 Use case tổng quát của quản trị viên 20](#_Toc12817)

[3.3.4 Đặc tả Use Case 20](#_Toc28657)

[3.3.4.1 Use case đăng nhập 20](#_Toc22231)

[3.3.4.2 Use case đăng kí 21](#_Toc639)

[3.3.4.3 Use case đăng bài 22](#_Toc22702)

[3.3.4.4 Use case quản lý bài viết 22](#_Toc7199)

[3.3.4.5 Use case tìm người dung 23](#_Toc19326)

[3.3.4.6 Use case quản lý danh sách bạn bè 24](#_Toc11580)

[3.3.4.7 Use case quản lý tin nhắn 24](#_Toc25000)

[3.3.4.8 Use case quản lý người dùng 25](#_Toc31713)

[3.3.4.9 Use case quản lý tag 26](#_Toc31087)

[3.3.5 Sơ đồ tuần tự 27](#_Toc1032)

[3.3.5.1 Sơ đồ tuần tự chức năng đăng ký 27](#_Toc25249)

[3.3.5.2 Sơ đồ tuần tự chức năng đăng nhập 27](#_Toc29468)

[3.3.5.3 Sơ đồ tuần tự chức năng đăng bài 28](#_Toc1671)

[3.3.5.4 Sơ đồ tuần tự chức năng cấp quyền cho người dùng 29](#_Toc24739)

[3.3.5.5 Sơ đồ tuần tự chức năng kết nối người dùng 29](#_Toc11369)

[3.4 Xác định thực thể 30](#_Toc13620)

[3.5 Mô hình ERD 31](#_Toc10029)

[3.6 Mô hình cơ sở dữ liệu quan hệ 31](#_Toc31268)

[3.7 Mô hình dữ liệu 35](#_Toc23759)

[3.7.1 Xét quan hệ USER-USER\_ROLE 35](#_Toc24459)

[3.7.2 Xét quan hệ USER-POST 35](#_Toc24269)

[3.7.3 Xét quan hệ USER-USER\_FRIEND 35](#_Toc29592)

[3.7.4 Xét quan hệ USER-USER\_FOLLOWING 35](#_Toc17079)

[3.7.5 Xét quan hệ USER-FRIEND\_REQUEST 36](#_Toc7493)

[3.7.6 Xét quan hệ USER-POST\_LIKE 36](#_Toc9352)

[3.7.7 Xét quan hệ USER-POST\_DISLIKE 36](#_Toc15852)

[3.7.8 Xét quan hệ USER-COMMENT\_LIKE 36](#_Toc18541)

[3.7.9 Xét quan hệ USER-COMMENT\_DISLIKE 36](#_Toc14415)

[3.7.10 Xét quan hệ USER-MESSAGE 36](#_Toc16425)

[3.7.10 Xét quan hệ POST-POST\_TAG 37](#_Toc20749)

[3.7.10 Xét quan hệ POST-COMMENT 37](#_Toc22585)

[3.7.10 Xét quan hệ POST-POST\_LIKE 37](#_Toc15263)

[3.7.10 Xét quan hệ POST- POST\_DISLIKE 37](#_Toc29531)

[3.8 Phân tích và hoàn thiện mô hình quan hệ 38](#_Toc77)

[CHƯƠNG 4: GIAO DIỆN 38](#_Toc28468)

[4.1. Giao diện đối với user chưa tạo tài khoản 38](#_Toc2749)

[4.1.1. Trang chủ 38](#_Toc17178)

[4.1.2. Trang đăng nhập 40](#_Toc28169)

[4.1.3. Trang đăng ký 40](#_Toc21081)

[4.1.4. Trang thông tin tác giả bài viết 41](#_Toc4738)

[4.1.5. Trang chi tiết bài viết 41](#_Toc30056)

[4.2. Giao diện đối với user có ROLE\_USER 42](#_Toc25224)

[4.2.1. Trang tạo bài đăng mới 42](#_Toc6623)

[4.2.2. Trang chi tiết bài đăng với nhiều tính năng hơn 43](#_Toc30495)

[4.2.3. Trang thông tin cá nhân của chính mình 44](#_Toc5953)

[4.2.4. Trang đổi mật khẩu 45](#_Toc2599)

[4.2.5. Trang thông tin của user khác với nhiều tín năng hơn 45](#_Toc1735)

[4.2.6. Trang nhắn tin với user khác 46](#_Toc25442)

[4.2.7. Trang Following Feed - nơi hiển thị bài viết của tác giả yêu thích 47](#_Toc7888)

[4.3. Giao diện đối với user có ROLE\_ADMIN 47](#_Toc24243)

[4.3.1. Trang chủ với nhiều tín năng hơn 47](#_Toc22193)

[4.3.2. Trang xem danh sách all users 48](#_Toc23861)

[4.3.3. Trang thông tin cá nhân của chính mình với nhiều tín năng hơn 49](#_Toc32227)

[4.3.4. Trang thông tin cá nhân của user khác với nhiều tín năng hơn 50](#_Toc16407)

[4.3.5. Trang chi tiết bài viết với nhiều tín năng hơn 50](#_Toc2547)

[4.4. Đánh giá chung 51](#_Toc17256)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 52](#_Toc14493)

[5.1. Kết luận 52](#_Toc55)

[5.2. Hướng phát triển 52](#_Toc20460)

[TÀI LIỆU THAM KHẢO 53](#_Toc22704)

# CHƯƠNG 1: CÔNG NGHỆ

## 1.1 Spring Boot

Spring Boot là một framework mã nguồn mở dựa trên Spring Framework, được thiết kế để đơn giản hóa việc tạo và triển khai các ứng dụng Java. Với Spring Boot, nhà phát triển có thể tập trung vào viết mã thay vì phải quản lý cấu hình phức tạp. Spring Boot cung cấp cơ chế tự động cấu hình, giúp thiết lập các thành phần ứng dụng một cách nhanh chóng và dễ dàng. Nó cũng hỗ trợ tích hợp với các công nghệ khác như Spring Security, Spring Data và Spring Cloud, tạo điều kiện thuận lợi cho việc xây dựng các ứng dụng microservices. Bằng cách sử dụng Spring Boot, bạn có thể phát triển và triển khai ứng dụng một cách nhanh chóng, giảm thiểu thời gian phát triển và tối ưu hóa hiệu suất.[1]

## 1.2. Spring Security và JWT

Spring Security là một framework mạnh mẽ dùng để bảo mật các ứng dụng web Java, cung cấp các tính năng như xác thực, phân quyền người dùng và bảo vệ chống lại các cuộc tấn công phổ biến như CSRF và XSS. JWT (JSON Web Token) là một tiêu chuẩn mở để truyền tải thông tin giữa các bên một cách an toàn dưới dạng token. Trong ứng dụng blog này, Spring Security được sử dụng để bảo vệ các API và JWT được sử dụng để quản lý phiên đăng nhập người dùng. Khi người dùng đăng nhập thành công, hệ thống sẽ tạo ra một JWT chứa thông tin xác thực của người dùng, giúp giảm tải cho máy chủ bằng cách loại bỏ nhu cầu kiểm tra trạng thái phiên liên tục.[2]

## 1.3. Spring Data JPA (Hibernate)

Spring Data JPA là một phần của Spring Data, cung cấp cách tiếp cận dễ dàng và mạnh mẽ để làm việc với các cơ sở dữ liệu quan hệ. Nó tích hợp chặt chẽ với Hibernate, một framework ORM (Object-Relational Mapping) phổ biến, giúp chuyển đổi giữa các đối tượng Java và các bảng cơ sở dữ liệu một cách tự động. Spring Data JPA giảm thiểu lượng mã nguồn phải viết để thao tác với cơ sở dữ liệu bằng cách cung cấp các repository có sẵn cho các thao tác CRUD (Create, Read, Update, Delete). Điều này giúp nhà phát triển tập trung vào logic nghiệp vụ thay vì phải viết các câu lệnh SQL phức tạp.[3]

## 1.4. MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ và phổ biến, được sử dụng rộng rãi trong các ứng dụng web. Với MySQL, bạn có thể quản lý và truy vấn dữ liệu một cách hiệu quả, hỗ trợ đa người dùng và đa nhiệm. MySQL tích hợp tốt với nhiều ngôn ngữ lập trình và framework, bao gồm cả Spring Boot, giúp bạn dễ dàng quản lý cơ sở dữ liệu của ứng dụng blog. MySQL cung cấp hiệu suất cao, đáng tin cậy và khả năng mở rộng, là lựa chọn lý tưởng cho các ứng dụng yêu cầu cơ sở dữ liệu quan hệ lớn.

## 1.5. ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, dùng để xây dựng giao diện người dùng (UI). ReactJS sử dụng kiến trúc component-based, cho phép xây dựng các thành phần UI độc lập, tái sử dụng. Một trong những tính năng nổi bật của ReactJS là Virtual DOM, giúp cải thiện hiệu suất ứng dụng bằng cách cập nhật DOM một cách hiệu quả. JSX, một cú pháp mở rộng của JavaScript, giúp viết mã React dễ hiểu và dễ duy trì. Với ReactJS, bạn có thể tạo ra các ứng dụng web phức tạp với hiệu suất cao và trải nghiệm người dùng mượt mà.

## 1.6. Socket.IO

Socket.IO là một thư viện JavaScript phổ biến được sử dụng để xây dựng các ứng dụng thời gian thực. Trong ngữ cảnh của tính năng nhắn tin thời gian thực trong ứng dụng blog, Socket.IO đảm bảo rằng các tin nhắn được gửi và nhận ngay lập tức giữa các người dùng, tạo ra trải nghiệm tương tác liền mạch. Điểm mạnh của Socket.IO là khả năng thiết lập kết nối hai chiều (bi-directional) giữa máy khách và máy chủ, cho phép dữ liệu được truyền tải một cách nhanh chóng và hiệu quả. Socket.IO hoạt động trên nền tảng WebSocket nhưng bổ sung thêm nhiều tính năng quan trọng như fallback (dự phòng) qua các giao thức khác (như HTTP long polling) khi WebSocket không khả dụng, đảm bảo rằng kết nối luôn được duy trì một cách ổn định bất kể môi trường mạng.

Trong ứng dụng blog, khi một người dùng gửi tin nhắn, Socket.IO đảm bảo rằng tin nhắn này được truyền tải ngay lập tức đến tất cả người nhận. Điều này giúp cải thiện đáng kể trải nghiệm người dùng, đặc biệt trong các trường hợp cần tính tức thì như nhắn tin trực tiếp giữa các người dùng. Thêm vào đó, Socket.IO cung cấp các cơ chế quản lý phiên, giúp xử lý các kết nối đồng thời và đảm bảo rằng thông tin liên lạc giữa các người dùng được duy trì ổn định.[4]

## 1.7. RabbitMQ

RabbitMQ là một hệ thống hàng đợi tin nhắn (message broker) mã nguồn mở, được sử dụng để quản lý và điều phối luồng dữ liệu giữa các dịch vụ trong một ứng dụng phân tán. Trong bối cảnh ứng dụng blog có nhiều người dùng cùng tương tác và thay đổi các thuộc tính trong cơ sở dữ liệu đồng thời, RabbitMQ đóng vai trò quan trọng trong việc tránh xung đột dữ liệu và đảm bảo tính nhất quán.

RabbitMQ hoạt động bằng cách nhận các thông điệp từ các producer (nhà sản xuất), sau đó phân phối chúng đến các consumer (người tiêu thụ) thông qua các hàng đợi tin nhắn. Khi nhiều người dùng cùng thực hiện các thay đổi lên cùng một thuộc tính trong bảng cơ sở dữ liệu, RabbitMQ có thể được sử dụng để điều phối các yêu cầu này, xếp chúng vào hàng đợi và xử lý từng yêu cầu một cách tuần tự. Điều này giúp ngăn chặn các tình huống xung đột dữ liệu, nơi mà một thuộc tính có thể bị ghi đè bởi các giá trị khác nhau từ nhiều người dùng.

Ngoài ra, RabbitMQ còn cung cấp các cơ chế để xác nhận thông điệp đã được xử lý thành công, cũng như khả năng xử lý các thông điệp bị lỗi hoặc không thể xử lý, giúp đảm bảo tính tin cậy và chính xác của dữ liệu trong hệ thống. Điều này đặc biệt quan trọng trong các ứng dụng yêu cầu độ chính xác cao.[5]

# CHƯƠNG 2: KHẢO SÁT HIỆN TRẠNG

## 2.1 Hiện trạng thực tế

Do nhu cầu của người dùng trên các nền tảng trực tuyến ngày càng cao, đặc biệt là trong việc nhận phản hồi ngay lập tức khi nhắn tin hoặc cập nhật nội dung, các hệ thống truyền thống thường không đáp ứng được kỳ vọng này, khiến người dùng phải đợi lâu. Khi lượng nội dung được chia sẻ ngày càng đa dạng, từ văn bản, hình ảnh đến video, các hệ thống phải đảm bảo khả năng xử lý nhanh chóng mà không làm giảm hiệu suất. Thêm vào đó, việc bảo mật dữ liệu cá nhân là một yếu tố ngày càng quan trọng do các rủi ro bảo mật đang gia tăng. Người dùng mong muốn hệ thống bảo vệ thông tin cá nhân và tài khoản của họ một cách an toàn. Đồng thời, khi có nhiều người dùng tương tác đồng thời trên nền tảng, việc quản lý dữ liệu có thể trở nên phức tạp, dễ dẫn đến sự không đồng bộ và không chính xác. Điều này đòi hỏi hệ thống cần có cơ chế xử lý thông minh để đảm bảo tính chính xác của dữ liệu. Không những vậy, việc quản lý lưu lượng truy cập lớn cũng là thách thức khi hệ thống phải đảm bảo hiệu suất ổn định, đặc biệt khi số lượng người dùng và dữ liệu tăng cao. Những vấn đề này đã dẫn đến sự cần thiết phải xây dựng các ứng dụng blog hiện đại với khả năng đáp ứng thời gian thực, đảm bảo trải nghiệm người dùng liền mạch và an toàn, đồng thời cải thiện hiệu suất và khả năng mở rộng của hệ thống.

## 2.2 Giải pháp cho các vấn đề

|  |  |
| --- | --- |
| Vấn đề | Giải pháp |
| Nhu cầu phản hồi tức thời trong việc nhắn tin và cập nhật nội dung | Tích hợp cơ chế giao tiếp thời gian thực để nhận phản hồi ngay lập tức mà không cần tải lại trang. |
| Xử lý nội dung đa dạng (văn bản, hình ảnh, video) mà vẫn đảm bảo hiệu suất | Quản lý dữ liệu hiệu quả để xử lý và lưu trữ các bài viết, hình ảnh, video, kết hợp cơ chế lưu trữ tạm thời để tăng tốc độ truy xuất |
| Bảo mật dữ liệu cá nhân và tài khoản người dùng | Bảo mật dữ liệu cá nhân và tài khoản người dùng |
| Quản lý dữ liệu không đồng bộ khi có nhiều người dùng tương tác đồng thời | Xử lý sự kiện và thông điệp không đồng bộ, đảm bảo dữ liệu được xử lý chính xác khi nhiều người dùng tương tác cùng lúc |
| Quản lý lưu lượng truy cập lớn và khả năng mở rộng | Thiết kế hệ thống có khả năng mở rộng và phân tải khi có lượng truy cập lớn. |
| Đảm bảo trải nghiệm liền mạch và không làm giảm hiệu suất | Tối ưu hóa giao diện người dùng, chỉ tải các thành phần cần thiết khi được yêu cầu |

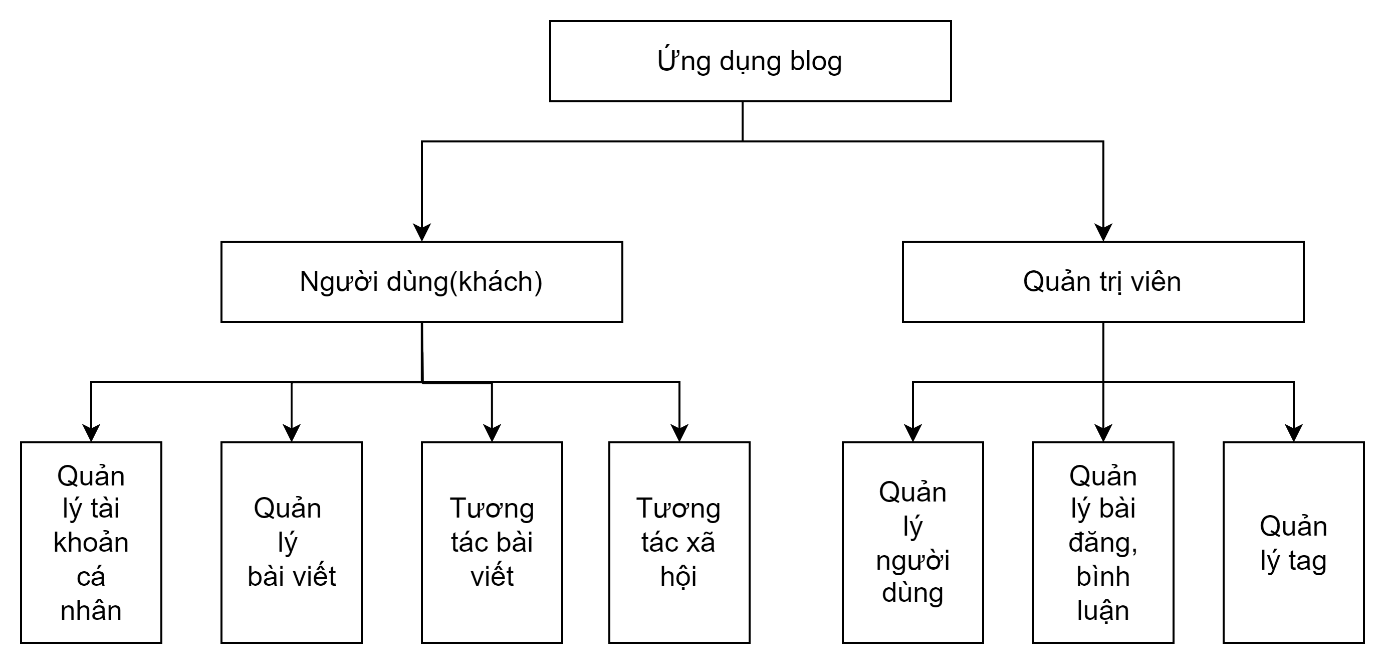
## 2.3 Mô hình nghiệp vụ hệ thống

### 2.3.1 Chức năng từng bộ phận

Bao gồm quản trị viên và người sử dụng:

- Admin (Quản trị viên): Quản trị viên có quyền quản lý toàn bộ hệ thống. Họ có thể quản lý người dùng, kiểm soát và điều hành các bài đăng, xử lý các yêu cầu liên quan đến bảo mật và phân quyền. Ngoài ra, admin còn có quyền cấp hoặc hạ quyền của người dùng, quản lý danh mục bài đăng, tags và các tài liệu liên quan.

- User (Người dùng): Người dùng có thể đăng ký, đăng nhập và sử dụng các tính năng cơ bản như tạo, chỉnh sửa và xóa bài viết. Họ cũng có thể tương tác với các người dùng khác thông qua các tính năng như theo dõi, gửi yêu cầu kết bạn, nhắn tin thời gian thực và bình luận trên bài viết. Người dùng cũng có thể thích (like), không thích (dislike) và chia sẻ nội dung từ hệ thống.



*Hình 1: Sơ đồ tổng quan hệ thống*

### 2.3.2 Quy trình nghiệp vụ

Về quy trình nghiệp vụ trong ứng dụng blog, có thể liệt kê như sau:

#### 2.3.2.1. Quy trình đăng ký và đăng nhập:

* **Đăng ký**:
  + Người dùng truy cập trang đăng ký.
  + Điền các thông tin cơ bản như tên, email, mật khẩu.
  + Hệ thống xác thực thông tin và tạo tài khoản mới.
* **Đăng nhập**:
  + Người dùng nhập email và mật khẩu để đăng nhập.
  + Hệ thống xác thực thông tin đăng nhập và cấp quyền truy cập dựa trên vai trò (user hoặc admin).

#### 2.3.2.2. Quy trình quản lý bài đăng:

**Tạo bài đăng**:

* + Người dùng hoặc admin truy cập trang tạo bài đăng mới.
  + Nhập tiêu đề, nội dung và chọn các tag liên quan.
  + Hệ thống lưu trữ bài đăng và hiển thị trên giao diện người dùng.

**Chỉnh sửa bài đăng**:

* + Người dùng hoặc admin chỉnh sửa nội dung bài đăng của mình thông qua giao diện chỉnh sửa.
  + Hệ thống cập nhật bài đăng và lưu các thay đổi.

**Xóa bài đăng**:

* + Người dùng hoặc admin có thể xóa bài đăng của mình.
  + Bài đăng được xóa khỏi hệ thống và không hiển thị nữa.

#### 2.3.2.3. Quy trình quản lý bạn bè và theo dõi:

* **Gửi yêu cầu kết bạn**:
  + Người dùng gửi yêu cầu kết bạn tới một người dùng khác.
  + Người nhận có thể chấp nhận hoặc từ chối yêu cầu.
* **Theo dõi một người dùng**:
  + Người dùng có thể theo dõi các bài đăng của người khác.
  + Hệ thống sẽ hiển thị bài viết từ các tài khoản đang theo dõi trên trang cá nhân của người dùng.

#### 2.3.2.4. Quy trình nhắn tin thời gian thực:

* **Gửi tin nhắn**:
  + Người dùng chọn một người để gửi tin nhắn trực tiếp.
  + Tin nhắn được truyền tải ngay lập tức thông qua hệ thống thời gian thực.
* **Nhận tin nhắn**:
  + Người nhận nhận được tin nhắn ngay lập tức mà không cần tải lại trang.

#### 2.3.2.5. Quy trình quản lý người dùng (dành cho admin):

* **Quản lý danh sách người dùng**:
  + Admin có thể xem toàn bộ danh sách người dùng, phân quyền và điều chỉnh các thông tin tài khoản nếu cần.
* **Cấp quyền admin**:
  + Admin có thể cấp quyền quản trị cho người dùng khác hoặc hạ quyền khi cần thiết.

#### 2.3.2.6. Quy trình quản lý bài đăng và bình luận:

* **Quản lý bình luận**:
  + Người dùng có thể thêm bình luận hoặc trả lời bình luận trên các bài đăng.
  + Admin có thể xóa bình luận nếu cần thiết.
* **Quản lý tags**:
  + Admin có thể tạo mới, chỉnh sửa hoặc xóa các tag không còn sử dụng.

#### 2.3.2.7. Quy trình tương tác với bài đăng:

* **Like/Dislike**:
  + Người dùng có thể thích (like) hoặc không thích (dislike) bài đăng của người khác.
* **Xem bài đăng theo tag hoặc người theo dõi**:
  + Người dùng có thể lọc bài viết theo tag hoặc xem bài đăng từ những người họ theo dõi.

## 2.4 Yêu cầu chức năng

### 2.4.1. Chức năng dành cho người dùng (User)

#### 2.4.1.1. ****Quản lý tài khoản****

* **Đăng ký**: Người dùng mới có thể đăng ký tài khoản với thông tin cá nhân cơ bản.
* **Đăng nhập**: Người dùng có thể đăng nhập vào hệ thống với tài khoản đã đăng ký.
* **Đổi mật khẩu**: Người dùng có thể thay đổi mật khẩu khi cần thiết.
* **Cập nhật thông tin cá nhân**: Cho phép người dùng chỉnh sửa thông tin cá nhân.

#### 2.4.1.2. ****Quản lý bài viết****

* **Tạo bài đăng**: Người dùng có thể tạo mới bài viết với nội dung, hình ảnh hoặc video.
* **Chỉnh sửa bài đăng**: Người dùng có thể chỉnh sửa bài viết đã tạo trước đó.
* **Xóa bài đăng**: Người dùng có thể xóa bài viết của chính mình.
* **Xem bài viết**: Người dùng có thể xem danh sách các bài viết trên hệ thống.
* **Xem bài viết theo tác giả**: Lọc và xem các bài viết được đăng bởi người dùng cụ thể.

#### 2.4.1.3. ****Tương tác với bài viết****

* **Like/Dislike bài viết**: Người dùng có thể thích hoặc không thích một bài viết.
* **Bình luận và trả lời bình luận**: Người dùng có thể thêm bình luận vào bài viết và trả lời các bình luận khác.
* **Xem danh sách người đã like bài viết**: Xem những người đã thích bài viết.
* **Bỏ like/dislike bài viết**: Người dùng có thể bỏ thích hoặc bỏ không thích.

#### 2.4.1.4. ****Kết bạn và tương tác xã hội****

* **Gửi yêu cầu kết bạn**: Người dùng có thể gửi yêu cầu kết bạn với người khác.
* **Chấp nhận/từ chối yêu cầu kết bạn**: Người dùng có thể chấp nhận hoặc từ chối yêu cầu kết bạn.
* **Theo dõi/unfollow người dùng khác**: Người dùng có thể theo dõi người dùng khác để xem bài viết của họ và hủy theo dõi khi không muốn nữa.
* **Xem danh sách bạn bè**: Người dùng có thể xem danh sách những người bạn đã kết bạn.
* **Xem danh sách following và followers**: Người dùng có thể xem danh sách những người mình đang theo dõi và danh sách người theo dõi mình.

#### 2.4.1.5. ****Nhắn tin thời gian thực****

* **Gửi tin nhắn**: Người dùng có thể gửi tin nhắn riêng tư cho người dùng khác.
* **Xem lịch sử tin nhắn**: Người dùng có thể xem lại các tin nhắn đã gửi và nhận.

#### 2.4.1.6. ****Tìm kiếm người dùng và bài viết****

* **Tìm kiếm người dùng**: Người dùng có thể tìm kiếm các tài khoản khác dựa trên tên người dùng.
* **Tìm kiếm bài viết**: Người dùng có thể tìm kiếm bài viết theo tiêu đề hoặc tag.

### 2.4.2. ****Chức năng dành cho quản trị viên (Admin)****

#### 2.4.2.1. ****Quản lý người dùng****

* **Xem danh sách người dùng**: Admin có thể xem danh sách tất cả các người dùng trong hệ thống.
* **Cấp quyền quản trị**: Admin có thể cấp quyền quản trị cho một tài khoản người dùng.
* **Hạ quyền quản trị**: Admin có thể hạ quyền quản trị từ một tài khoản admin xuống quyền user thường.
* **Xóa người dùng**: Admin có thể xóa tài khoản người dùng vi phạm quy tắc.

#### 2.4.2.2. ****Quản lý bài đăng****

* **Xem danh sách bài đăng**: Admin có thể xem tất cả các bài đăng trong hệ thống.
* **Chỉnh sửa bài đăng**: Admin có thể chỉnh sửa bài đăng trong trường hợp cần thiết.
* **Xóa bài đăng**: Admin có quyền xóa các bài đăng không phù hợp hoặc vi phạm quy tắc.

#### 2.4.2.3. ****Quản lý bình luận****

* **Xem danh sách bình luận**: Admin có thể xem tất cả các bình luận được đăng trên hệ thống.
* **Xóa bình luận**: Admin có thể xóa các bình luận có nội dung không phù hợp.

#### 2.4.2.4. ****Quản lý tag****

* **Tạo mới tag**: Admin có thể tạo mới các tag để gắn vào bài viết.
* **Xóa tag**: Admin có thể xóa các tag không còn được sử dụng hoặc liên quan.

#### 2.4.2.5. ****Quản lý yêu cầu kết bạn****

* **Xem danh sách yêu cầu kết bạn**: Admin có thể xem và quản lý các yêu cầu kết bạn trong hệ thống.

### 2.4.3. ****Chức năng bảo mật****

* **Xác thực người dùng**: Hệ thống phải xác thực tất cả người dùng trước khi cho phép truy cập các tính năng.
* **Phân quyền**: Phân biệt rõ ràng các chức năng giữa người dùng thường (user) và quản trị viên (admin) nhằm bảo vệ dữ liệu và tài nguyên hệ thống.

### 2.4.4. ****Chức năng hiệu suất****

* **Xử lý thời gian thực**: Các thao tác như nhắn tin, gửi yêu cầu kết bạn, và cập nhật bài viết phải được thực hiện ngay lập tức mà không cần tải lại trang.
* **Xử lý đồng bộ**: Hệ thống cần đảm bảo dữ liệu được xử lý chính xác khi có nhiều người dùng tương tác cùng lúc trên một đối tượng dữ liệu.

## 2.5 Yêu cầu phi chức năng:

- Hệ thống chạy đúng, tốt, ổn định.

- Giao diện thân thiện, dễ sử dụng.

- Đáp ứng hầu hết các yêu cầu của người dùng.

## 2.6 Yêu cầu hệ thống:

- Cung cấp phân quyền cho các loại người dùng có trách nhiệm: Quản trị viên và Người sử dụng

## 2.7. Mục đích nghiên cứu đề tài

Mục đích của đề tài "Xây dựng ứng dụng blog đáp ứng thời gian thực" là nghiên cứu và triển khai một ứng dụng web hiện đại, cho phép người dùng tương tác với hệ thống một cách mượt mà và nhanh chóng, đặc biệt trong các tình huống yêu cầu phản hồi ngay lập tức như nhắn tin và cập nhật bài viết. Trong bối cảnh công nghệ phát triển nhanh chóng, người dùng ngày càng kỳ vọng vào những trải nghiệm tức thời và liền mạch trên các nền tảng trực tuyến. Vì vậy, đề tài này nhằm khám phá và tích hợp các công nghệ tiên tiến như Spring Boot, Spring Security, JWT, Spring Data JPA (Hibernate), MySQL, ReactJS, Socket.IO, RabbitMQ cùng nhiều công nghệ khác để xây dựng một hệ thống blog không chỉ đảm bảo hiệu suất cao mà còn đáp ứng được nhu cầu về tính năng thời gian thực.

Ứng dụng blog sẽ phục vụ như một nền tảng nơi người dùng có thể tạo, chỉnh sửa, xóa và quản lý các bài viết của mình, đồng thời giao tiếp với các người dùng khác thông qua tính năng nhắn tin thời gian thực. Việc tích hợp các công nghệ như Socket.IO sẽ giúp các thông điệp được truyền tải ngay lập tức mà không cần tải lại trang, trong khi RabbitMQ sẽ đảm bảo rằng dữ liệu được xử lý một cách chính xác và hiệu quả ngay cả khi có nhiều người dùng tương tác cùng lúc trên cùng một đối tượng dữ liệu. Ngoài ra, việc sử dụng Spring Security kết hợp với JWT sẽ đảm bảo tính bảo mật cho hệ thống, chỉ cho phép những người dùng có quyền truy cập vào các tài nguyên nhạy cảm.

## 2.8. Mục tiêu nghiên cứu đề tài

Mục tiêu chính của đề tài bao gồm:

- Phát triển một nền tảng blog hoàn chỉnh: Ứng dụng blog sẽ được phát triển với đầy đủ các chức năng như đăng bài viết, chỉnh sửa, xóa, quản lý danh mục, và phân quyền người dùng. Hệ thống sẽ hỗ trợ nhiều loại nội dung, bao gồm văn bản, hình ảnh và video, tạo điều kiện cho người dùng tạo ra các bài viết phong phú và đa dạng.

- Tích hợp tính năng thời gian thực: Một trong những điểm nhấn của ứng dụng là tính năng nhắn tin thời gian thực, cho phép người dùng tương tác với nhau mà không cần tải lại trang. Socket.IO sẽ đóng vai trò là công cụ chính trong việc truyền tải dữ liệu thời gian thực giữa các người dùng, giúp cải thiện trải nghiệm người dùng và nâng cao hiệu quả giao tiếp.

- Bảo mật hệ thống: Đảm bảo rằng hệ thống có thể bảo vệ dữ liệu cá nhân của người dùng khỏi các nguy cơ bảo mật thông qua việc sử dụng Spring Security và JWT. Hệ thống sẽ yêu cầu người dùng đăng nhập và xác thực thông qua các token bảo mật, chỉ cho phép truy cập vào các tài nguyên nhất định khi người dùng đã được xác thực.

- Quản lý dữ liệu phức tạp: RabbitMQ sẽ được tích hợp để xử lý các tác vụ liên quan đến quản lý dữ liệu phức tạp, chẳng hạn như việc cập nhật đồng thời trên cùng một dữ liệu bởi nhiều người dùng. Điều này sẽ đảm bảo rằng dữ liệu trong cơ sở dữ liệu MySQL luôn được cập nhật chính xác và nhất quán.

## 2.9. Phương pháp tiến hành

Để đạt được các mục tiêu đề ra, đề tài sẽ sử dụng một loạt các phương pháp tiến hành như sau:

- Phân tích yêu cầu: Tiến hành khảo sát và thu thập yêu cầu từ người dùng cuối, sau đó phân tích và xác định các yêu cầu chức năng và phi chức năng của hệ thống. Điều này sẽ giúp đảm bảo rằng hệ thống đáp ứng đúng và đủ các nhu cầu thực tế của người dùng.

- Thiết kế kiến trúc hệ thống: Xây dựng mô hình kiến trúc tổng thể cho hệ thống, xác định các thành phần chính như backend, frontend, cơ sở dữ liệu và các dịch vụ hỗ trợ. Mỗi thành phần sẽ được thiết kế chi tiết với các công nghệ phù hợp, đảm bảo tính linh hoạt và khả năng mở rộng trong tương lai.

- Phát triển phần mềm: Sử dụng phương pháp phát triển phần mềm Agile để phát triển hệ thống theo từng giai đoạn nhỏ, mỗi giai đoạn sẽ bao gồm lập trình, kiểm thử và điều chỉnh dựa trên phản hồi của người dùng. Việc này giúp tối ưu hóa quy trình phát triển, đồng thời giảm thiểu rủi ro và chi phí phát sinh.

- Kiểm thử và triển khai: Sau khi hoàn thành việc phát triển, hệ thống sẽ được kiểm thử toàn diện bao gồm các bài kiểm tra chức năng, phi chức năng, kiểm tra hiệu suất và bảo mật. Sau khi vượt qua các bài kiểm tra, hệ thống sẽ được triển khai trên môi trường thực tế.

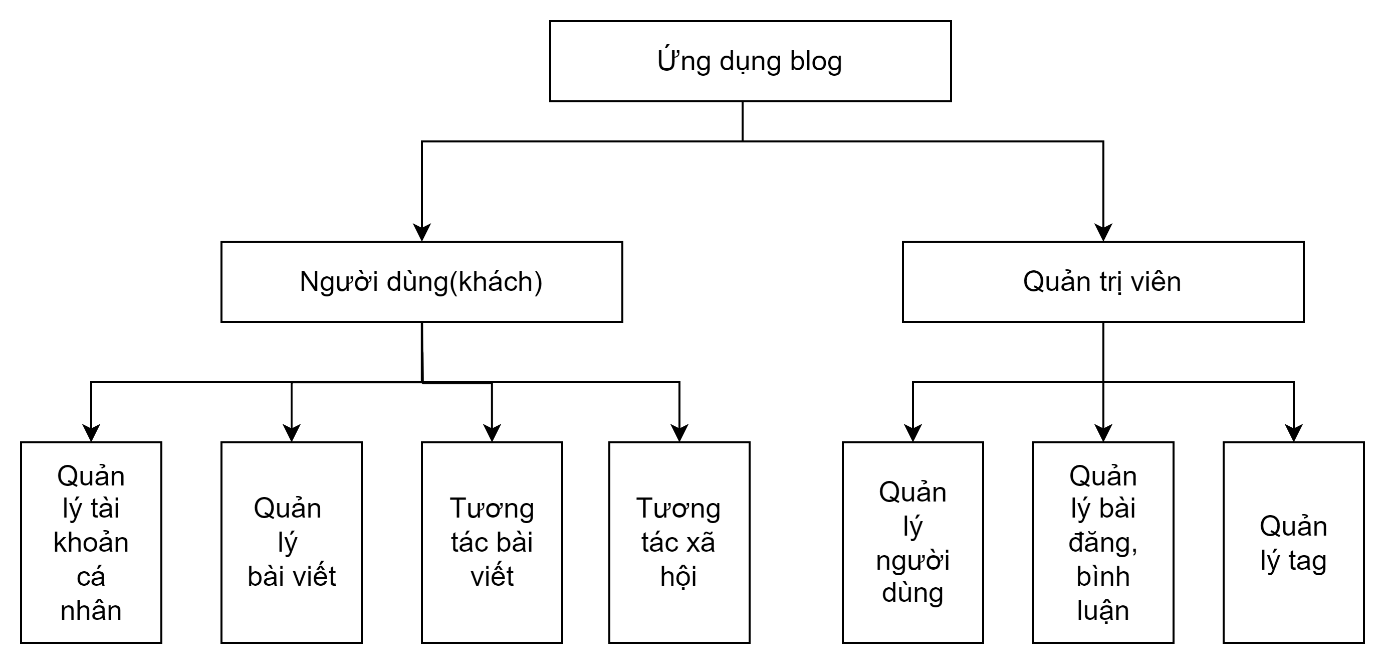
- Bảo trì và cập nhật: Sau khi triển khai, hệ thống sẽ được theo dõi và bảo trì định kỳ để đảm bảo hoạt động ổn định. Ngoài ra, các bản cập nhật và nâng cấp sẽ được thực hiện khi cần thiết để cải thiện tính năng và đáp ứng các yêu cầu mới từ người dùng.

Bằng cách áp dụng những phương pháp này, đề tài "Xây dựng ứng dụng blog đáp ứng thời gian thực" không chỉ đạt được các mục tiêu kỹ thuật mà còn góp phần vào việc nâng cao kỹ năng phát triển phần mềm của người thực hiện, đặc biệt là trong việc áp dụng các công nghệ hiện đại để giải quyết các bài toán phức tạp trong thực tế

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 3.1 Sơ đồ phân cấp chức năng

### 3.1.1 Sơ đồ phân rã chức năng mức đỉnh

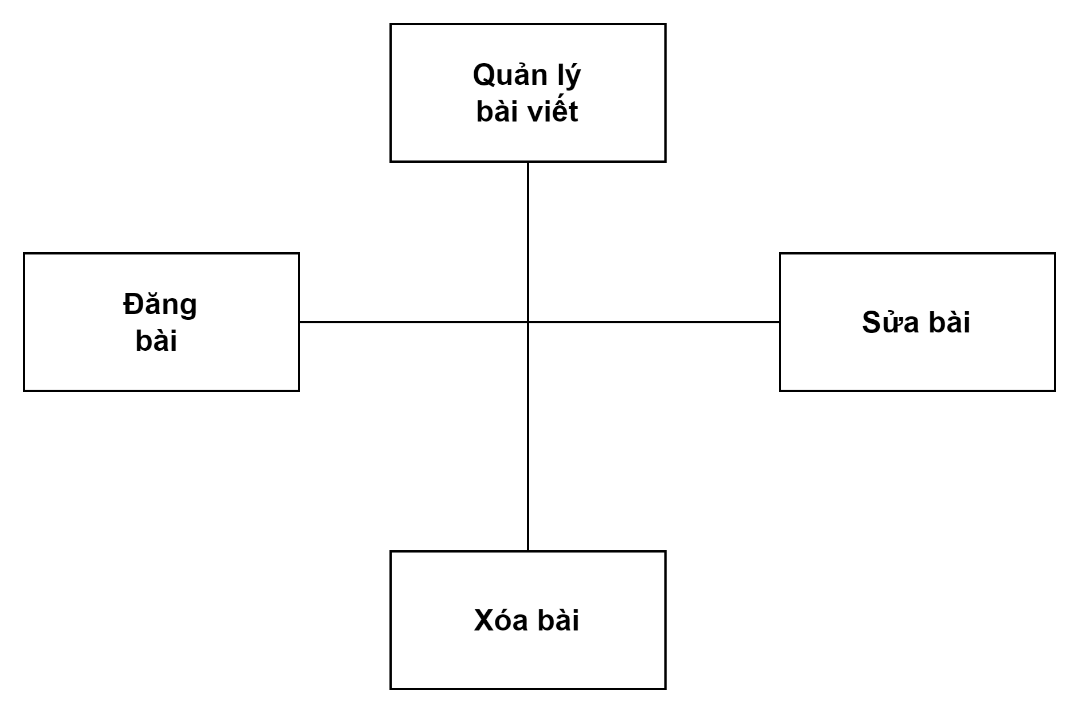


*Hình 2:* *Sơ đồ phân rã chức năng mức đỉnh*

Hệ thống nghiệp vụ được xét gồm các chức năng chính: với quản trị viên sẽ có các chức năng: quản lý người dung, quản lý bài đăng, bình luận, tag. Với người dung sẽ có các chức năng: quản lý tài khoản cá nhân, quản lý bài viết, tương tác với bài viết và xã hội.

### 3.1.2 Sơ đồ phân rã chức năng

**3.1.2.1 Chức năng quản lý bài đăng**



*Hình 3: Sơ đồ phân rã chức năng quản lý bài đăng*

**Tạo bài đăng**: Người dùng có thể tạo mới bài viết với nội dung, hình ảnh hoặc video.

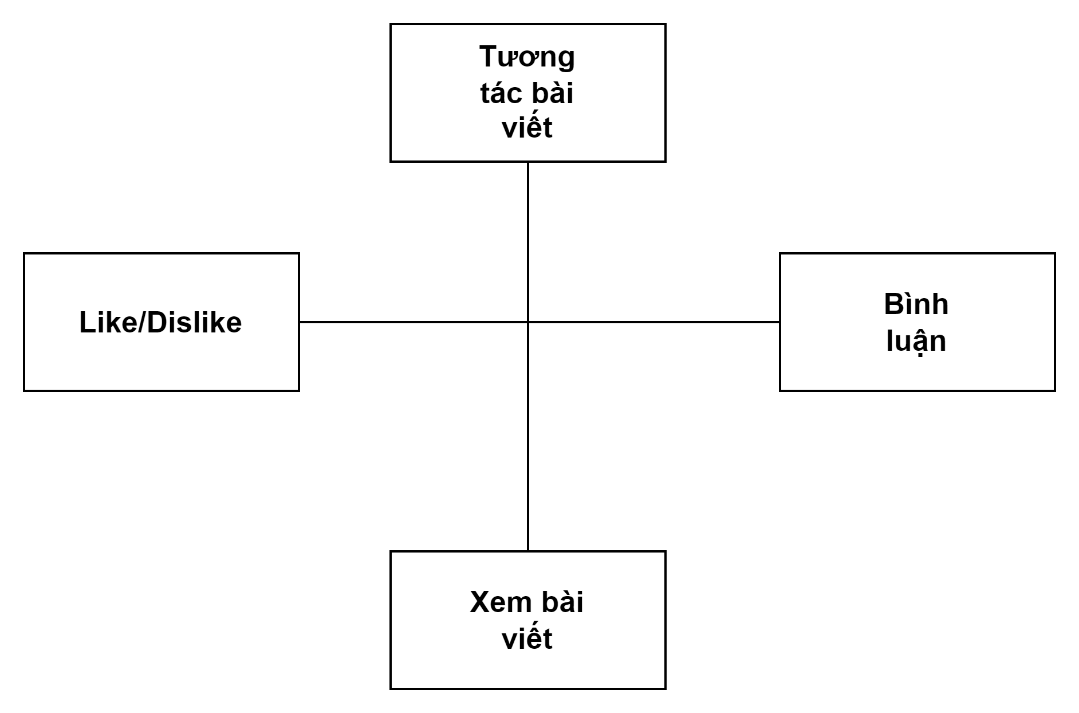
**Chỉnh sửa bài đăng**: Người dùng có thể chỉnh sửa bài viết đã tạo trước đó.

**Xóa bài đăng**: Người dùng có thể xóa bài viết của chính mình.

**Xem bài viết**: Người dùng có thể xem danh sách các bài viết trên hệ thống.

**Xem bài viết theo tác giả**: Lọc và xem các bài viết được đăng bởi người dùng cụ thể.

**3.1.2.2 Chức năng tương tác bài đăng**



*Hình 4: Sơ đồ phân rã chức năng tương tác với bài đăng*

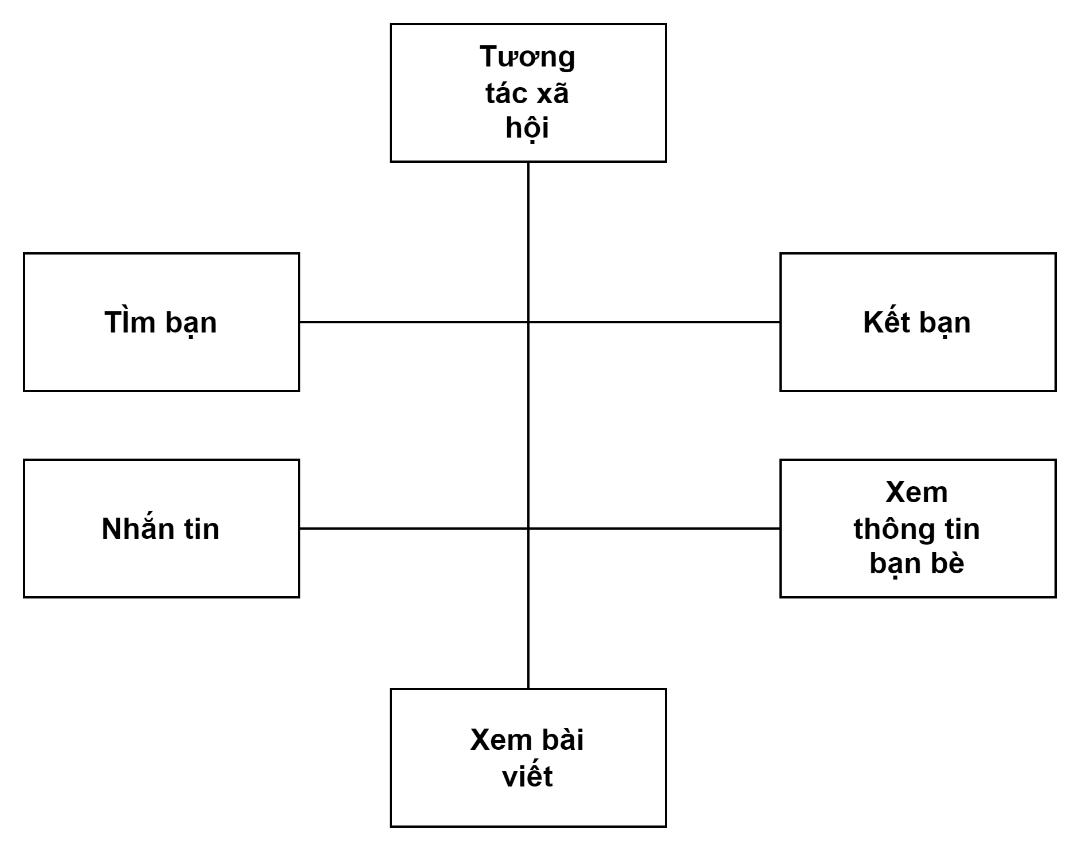
**Like/Dislike bài viết**: Người dùng có thể thích hoặc không thích một bài viết.

**Bình luận và trả lời bình luận**: Người dùng có thể thêm bình luận vào bài viết và trả lời các bình luận khác.

**Xem danh sách người đã like bài viết**: Xem những người đã thích bài viết.

**Bỏ like/dislike bài viết**: Người dùng có thể bỏ thích hoặc bỏ không thích.

**3.1.2.3 Chức năng tương tác xã hội**



*Hình 5: Sơ đồ phân rã chức năng tương tác với xã hội*

**Gửi yêu cầu kết bạn**: Người dùng có thể gửi yêu cầu kết bạn với người khác.

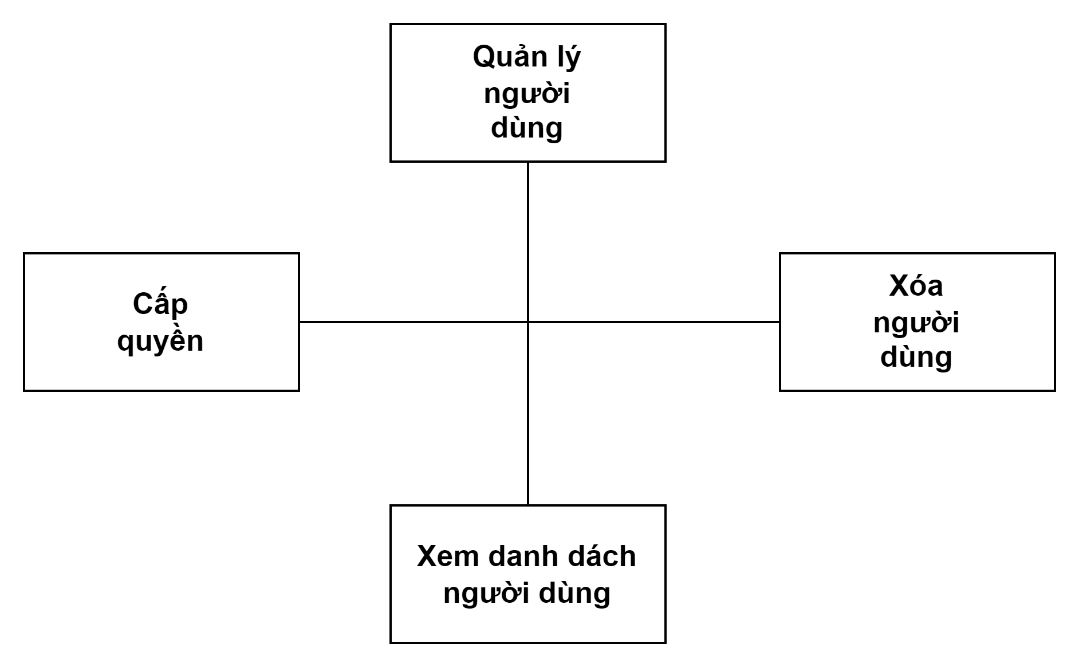
**Chấp nhận/từ chối yêu cầu kết bạn**: Người dùng có thể chấp nhận hoặc từ chối yêu cầu kết bạn.

**Theo dõi/unfollow người dùng khác**: Người dùng có thể theo dõi người dùng khác để xem bài viết của họ và hủy theo dõi khi không muốn nữa.

**Xem danh sách bạn bè**: Người dùng có thể xem danh sách những người bạn đã kết bạn.

**Xem danh sách following và followers**: Người dùng có thể xem danh sách những người mình đang theo dõi và danh sách người theo dõi mình.

**3.1.2.4 Chức năng quản lý người dùng**



*Hình 6: Sơ đồ phân rã chức năng quản lý người dùng*

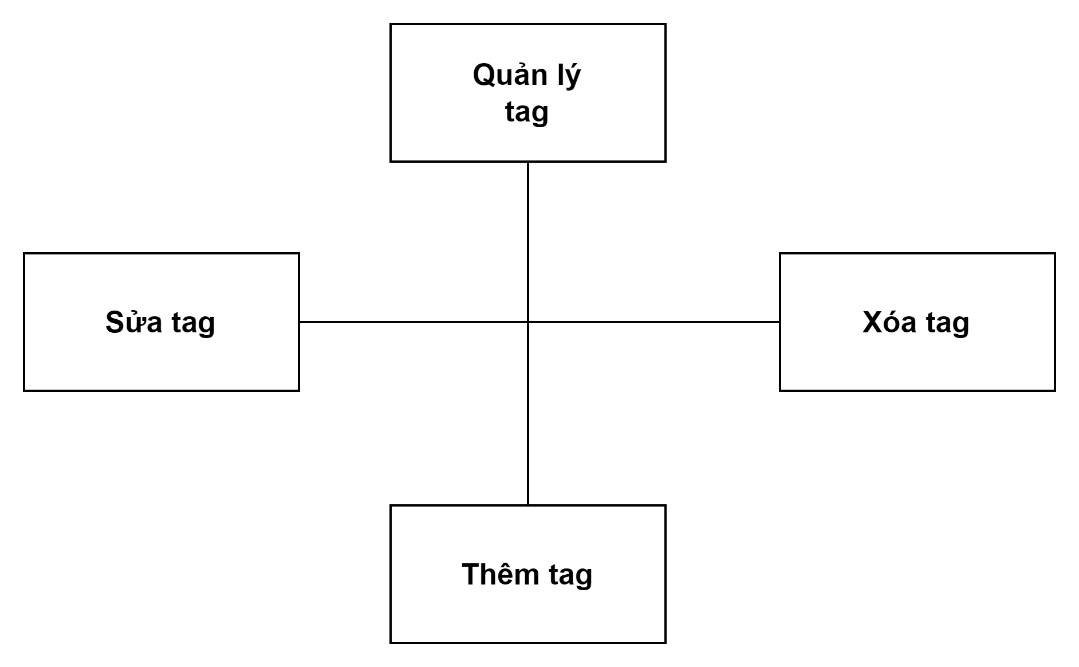
**Xem danh sách người dùng**: Admin có thể xem danh sách tất cả các người dùng trong hệ thống.

**Cấp quyền quản trị**: Admin có thể cấp quyền quản trị cho một tài khoản người dùng.

**Hạ quyền quản trị**: Admin có thể hạ quyền quản trị từ một tài khoản admin xuống quyền user thường.

**Xóa người dùng**: Admin có thể xóa tài khoản người dùng vi phạm quy tắc.

**3.1.2.5 Chức năng quản lý tag**



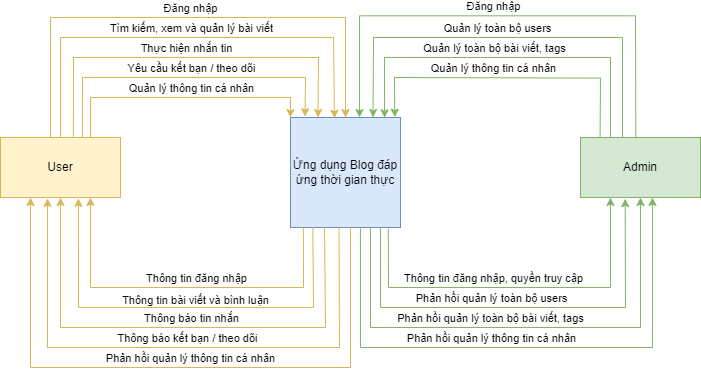
*Hình 7: Sơ đồ phân rã chức năng quản lý tag*

**Tạo mới tag**: Admin có thể tạo mới các tag để gắn vào bài viết.

**Xóa tag**: Admin có thể xóa các tag không còn được sử dụng hoặc liên quan.

## 3.2 Mô hình luồng dữ liệu DFD

### 3.2.1 Biểu đồ DFD mức ngữ cảnh



*Hình 8: Biểu đồ luồng dữ liệu mức ngữ cảnh*

### 3.2.2 Biểu đồ DFD phân rã cấp 0 (chức năng chính)

## 3.3 Mô hình Use Case

### 3.3.1 Xác định Actor

Dựa vào yêu cầu của đề tài, ta có các actor: người dùng(khách) và quản trị viên.

### 3.3.2 Xác định Use Case

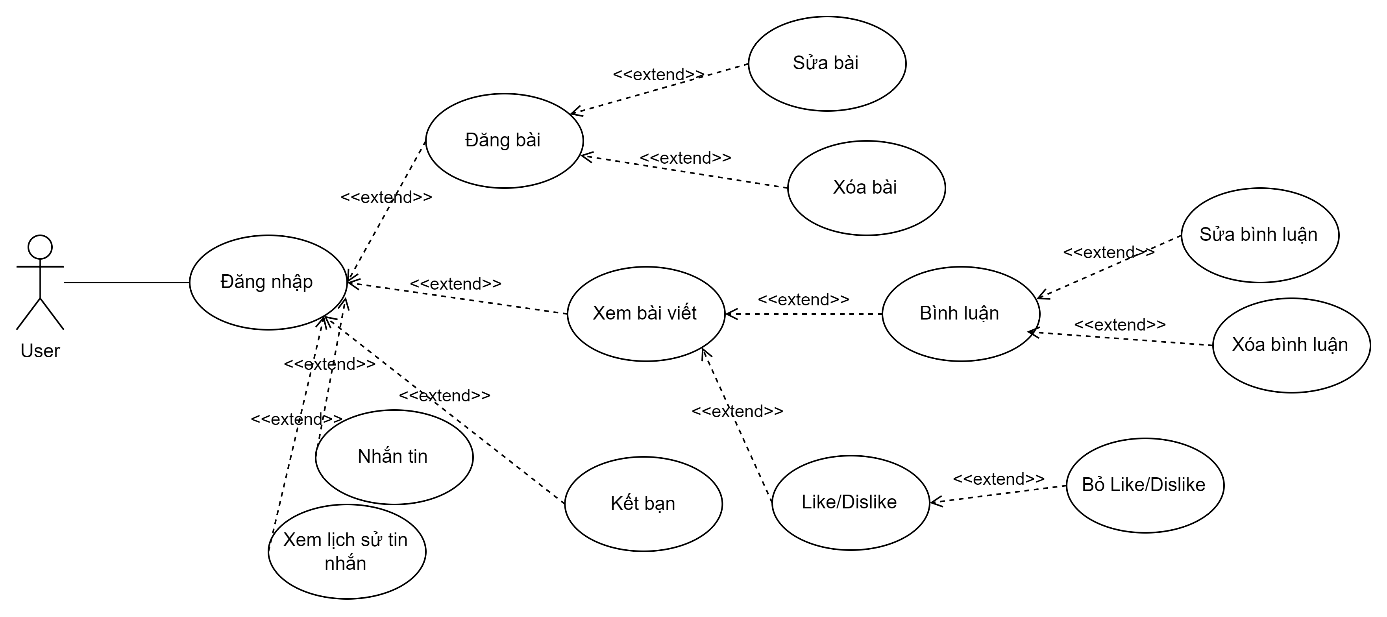
Từ yêu cầu chức năng ứng với từng actor ta có thể xác định được các use

case như sau:

|  |  |
| --- | --- |
| Actor | Use case |
| Người dùng(khách) | Đăng nhập, đăng kí, quản bài viết, quản lý tài khoản cá nhân, bình luận, nhắn tin |
| Quản trị viên | Đăng nhập, quản lý người dung, quản lý bài viết, quản lý tag, quản lý bình luận |

### 3.3.3 Xây dựng Use Case tổng quát

#### 3.3.3.1 Use case tổng quát của người dùng



*Hình 9: Use case tổng quát của người dùng*

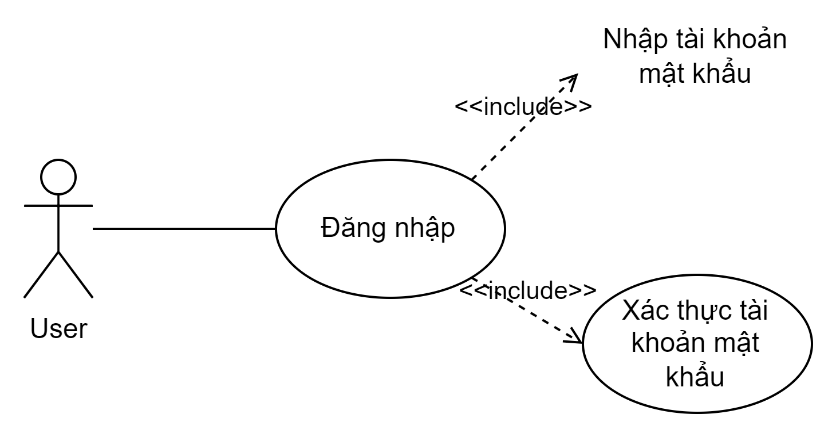
#### 3.3.3.2 Use case tổng quát của quản trị viên



*Hình 10: Use case tổng quát của quản trị viên*

### 3.3.4 Đặc tả Use Case

#### 3.3.4.1 Use case đăng nhập



**Mục đích:** Cho phép người dùng đăng nhập vào hệ thống.

**Mô tả:** Người dùng sẽ nhập tài khoản và mật khẩu của họ để xác thực và truy cập vào hệ thống.

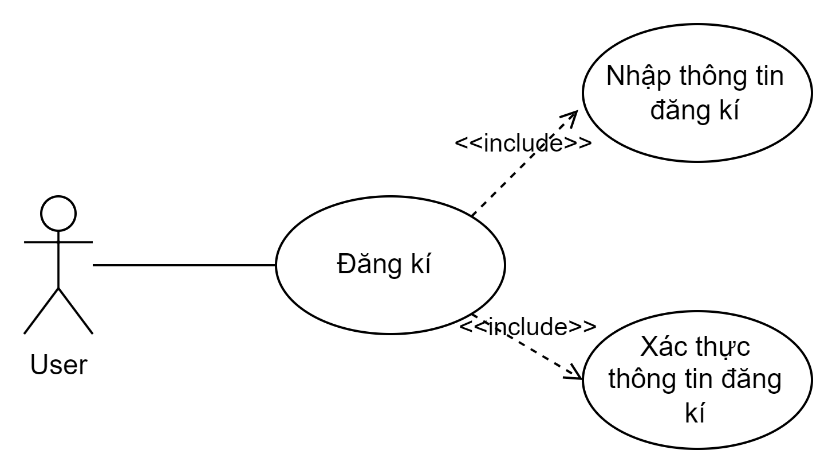
**Tác nhân (Actor):** Người dùng.

**Điều kiện tiên quyết:** Người dùng đã có tài khoản hợp lệ trong hệ thống.

**Dòng sự kiện chính:**

1. Người dùng nhấn vào nút "Đăng nhập".
2. Hệ thống hiển thị biểu mẫu yêu cầu nhập tài khoản và mật khẩu.
3. Người dùng nhập thông tin tài khoản và mật khẩu.
4. Hệ thống xác thực thông tin đăng nhập.
5. Nếu thông tin chính xác, người dùng được phép truy cập vào hệ thống.
6. Nếu thông tin sai, hệ thống thông báo lỗi và yêu cầu nhập lại.

#### 3.3.4.2 Use case đăng kí



**Mục đích:** Cho phép người dùng mới tạo tài khoản trên hệ thống.

**Mô tả:** Người dùng sẽ nhập thông tin đăng ký để tạo một tài khoản mới.

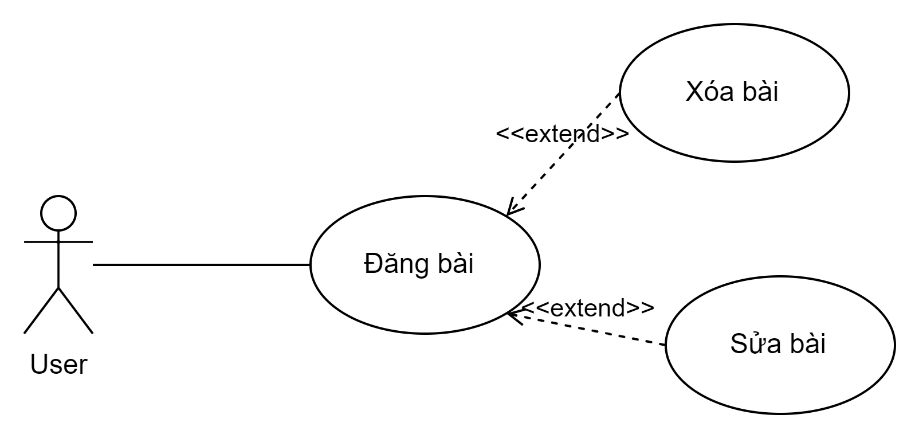
**Tác nhân (Actor):** Người dùng.

**Điều kiện tiên quyết:** Người dùng chưa có tài khoản trong hệ thống.

**Dòng sự kiện chính:**

* + 1. Người dùng nhấn vào nút "Đăng ký".
    2. Hệ thống hiển thị biểu mẫu yêu cầu nhập thông tin đăng ký (ví dụ: tên, email, mật khẩu).
    3. Người dùng nhập thông tin đăng ký.
    4. Hệ thống xác thực thông tin đăng ký.
    5. Nếu thông tin hợp lệ, tài khoản mới được tạo và người dùng có thể sử dụng tài khoản.
    6. Nếu thông tin không hợp lệ (ví dụ: email đã được sử dụng), hệ thống thông báo lỗi và yêu cầu nhập lại thông tin.

#### 3.3.4.3 Use case đăng bài



**Mục đích:** Cho phép người dùng xóa một bài viết đã đăng.

**Mô tả:** Người dùng có thể xóa bài viết mà họ đã tạo.

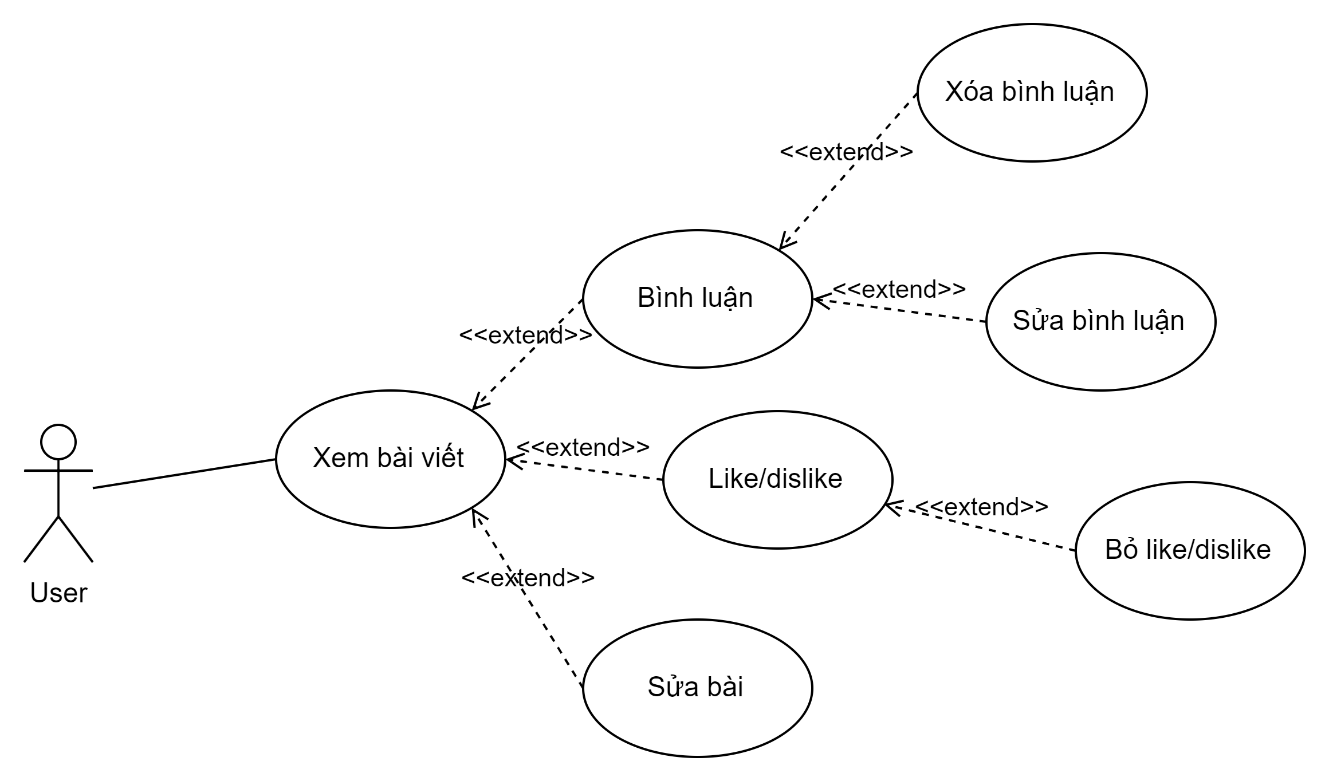
**Tác nhân (Actor):** Người dùng.

**Điều kiện tiên quyết:** Người dùng đã đăng bài viết trước đó và đang đăng nhập vào hệ thống.

**Dòng sự kiện chính:**

1. Người dùng chọn bài viết mà họ muốn xóa.
2. Hệ thống hiển thị cảnh báo xác nhận xóa bài viết.
3. Người dùng xác nhận xóa.
4. Hệ thống xóa bài viết khỏi cơ sở dữ liệu.
5. Bài viết không còn hiển thị trên hệ thống.

#### 3.3.4.4 Use case quản lý bài viết



**Mục đích:** Cho phép người dùng xem chi tiết một bài viết.

**Mô tả:** Người dùng có thể truy cập vào bài viết để xem nội dung.

**Tác nhân (Actor):** Người dùng.

**Điều kiện tiên quyết:** Người dùng phải đăng nhập.

**Dòng sự kiện chính:**

1. Người dùng nhấn vào tiêu đề bài viết.
2. Hệ thống hiển thị toàn bộ nội dung bài viết cùng các bình luận liên quan.
3. Người dùng có thể đọc bài viết và bình luận.

#### 3.3.4.5 Use case tìm người dung



**Mô tả:** Cho phép người dùng tìm kiếm những người dùng khác trên hệ thống.

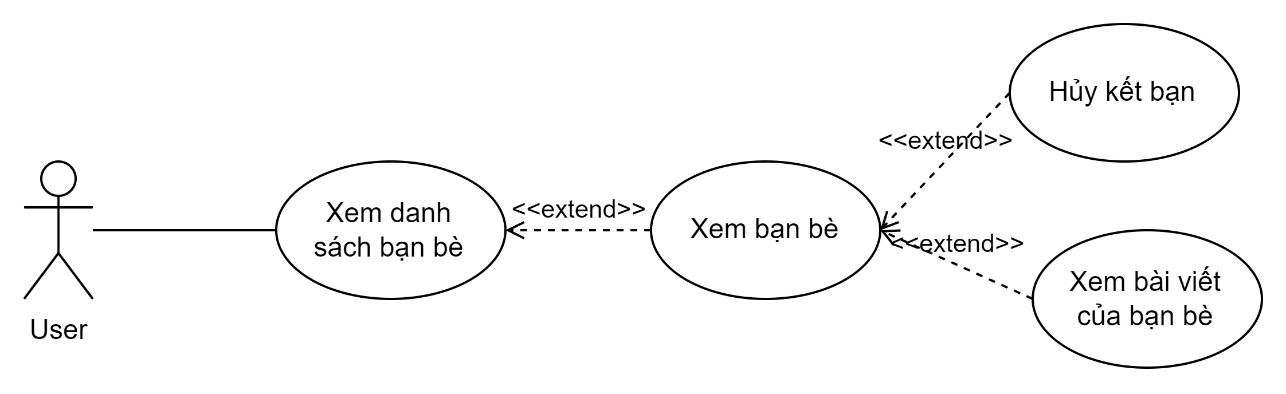
**Tác nhân chính:** Người dùng.

**Dòng sự kiện chính:**

1. Người dùng truy cập tính năng tìm kiếm.
2. Người dùng nhập từ khóa liên quan đến người dùng khác.
3. Hệ thống tìm kiếm trong cơ sở dữ liệu người dùng.
4. Hệ thống hiển thị danh sách người dùng phù hợp.

**Kết quả:** Danh sách người dùng được hiển thị cho người tìm kiếm.

#### 3.3.4.6 Use case quản lý danh sách bạn bè



**Mô tả:** Người dùng có thể hủy kết bạn với một người dùng đã kết bạn.

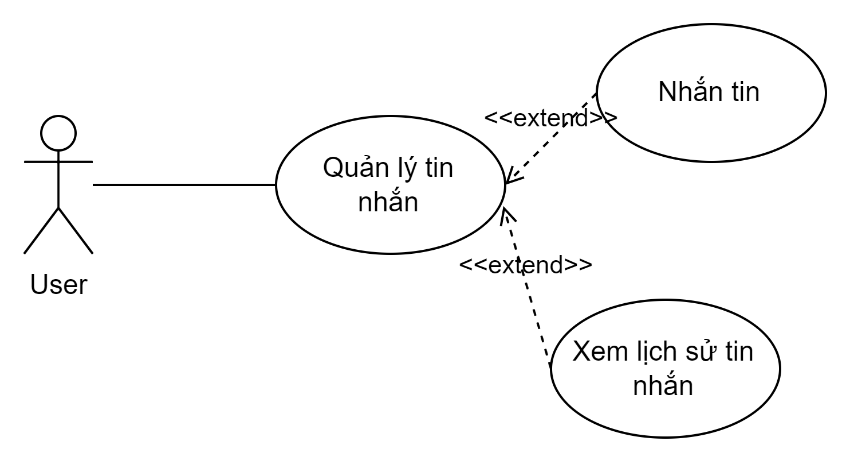
**Tác nhân chính:** Người dùng.

**Dòng sự kiện chính:**

1. Người dùng truy cập danh sách bạn bè.
2. Người dùng chọn người dùng cần hủy kết bạn.
3. Người dùng chọn tùy chọn "Hủy kết bạn".
4. Hệ thống hủy kết bạn giữa hai người dùng.
5. Hệ thống hiển thị thông báo xác nhận đã hủy kết bạn.

**Kết quả:** Quan hệ kết bạn giữa hai người dùng bị hủy.

#### 3.3.4.7 Use case quản lý tin nhắn



**Mô tả:** Người dùng có thể gửi tin nhắn đến người dùng khác.

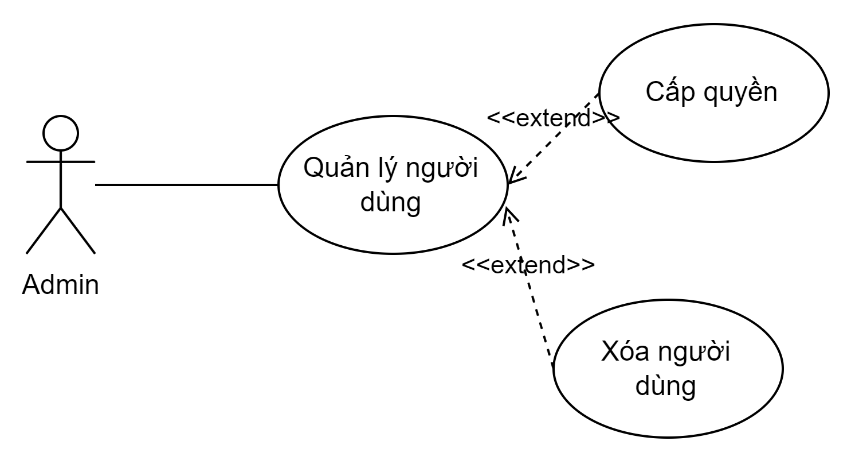
**Tác nhân chính:** Người dùng.

**Dòng sự kiện chính:**

1. Người dùng truy cập vào danh sách bạn bè hoặc tìm kiếm người dùng cần nhắn tin.
2. Người dùng chọn tùy chọn "Nhắn tin".
3. Hệ thống hiển thị giao diện nhắn tin.
4. Người dùng nhập tin nhắn vào hộp văn bản.
5. Người dùng chọn "Gửi" để gửi tin nhắn.
6. Hệ thống gửi tin nhắn đến người dùng nhận và lưu trữ vào cơ sở dữ liệu.
7. Hệ thống hiển thị thông báo rằng tin nhắn đã được gửi thành công.

**Kết quả:** Tin nhắn được gửi và hiển thị cho người nhận.

#### 3.3.4.8 Use case quản lý người dùng



**Mô tả:** Quản trị viên có thể quản lý người dùng trong hệ thống (bao gồm việc xóa người dùng, cấp quyền quản trị).

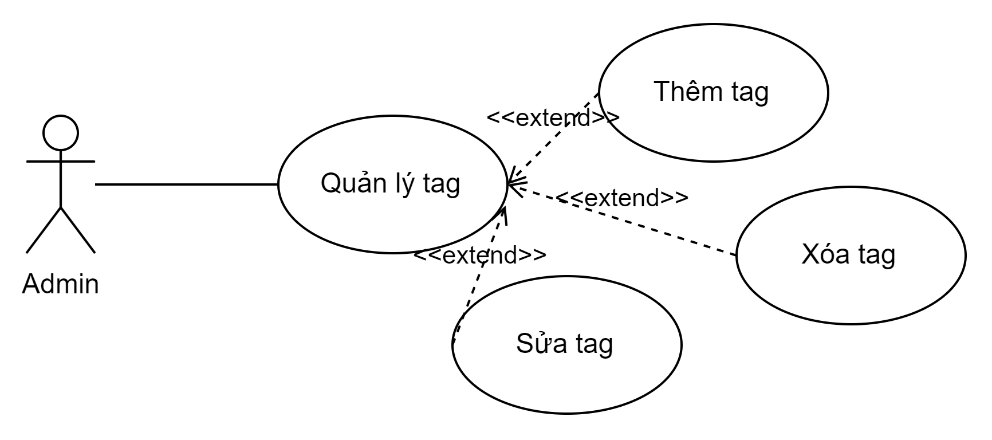
**Tác nhân chính:** Quản trị viên.

**Dòng sự kiện chính:**

1. Quản trị viên truy cập vào trang quản lý người dùng.
2. Quản trị viên xem danh sách người dùng.
3. Quản trị viên chọn một người dùng cụ thể.
4. Quản trị viên có thể thực hiện một trong các hành động:
5. Xóa người dùng:
6. Quản trị viên chọn tùy chọn "Xóa người dùng".
7. Hệ thống xác nhận hành động.
8. Hệ thống xóa người dùng khỏi cơ sở dữ liệu.
9. Cấp quyền:
10. Quản trị viên chọn tùy chọn "Cấp quyền".
11. Quản trị viên chọn loại quyền cần cấp (quản trị viên, người dùng thường).
12. Hệ thống cập nhật quyền hạn của người dùng.
13. Hệ thống thông báo thành công.

**Kết quả:** Người dùng được quản lý (xóa hoặc cấp quyền) theo yêu cầu của quản trị viên.

#### 3.3.4.9 Use case quản lý tag



**Mô tả:** Quản trị viên có thể quản lý (thêm, xóa, sửa) các tag trong hệ thống.

**Tác nhân chính:** Quản trị viên.

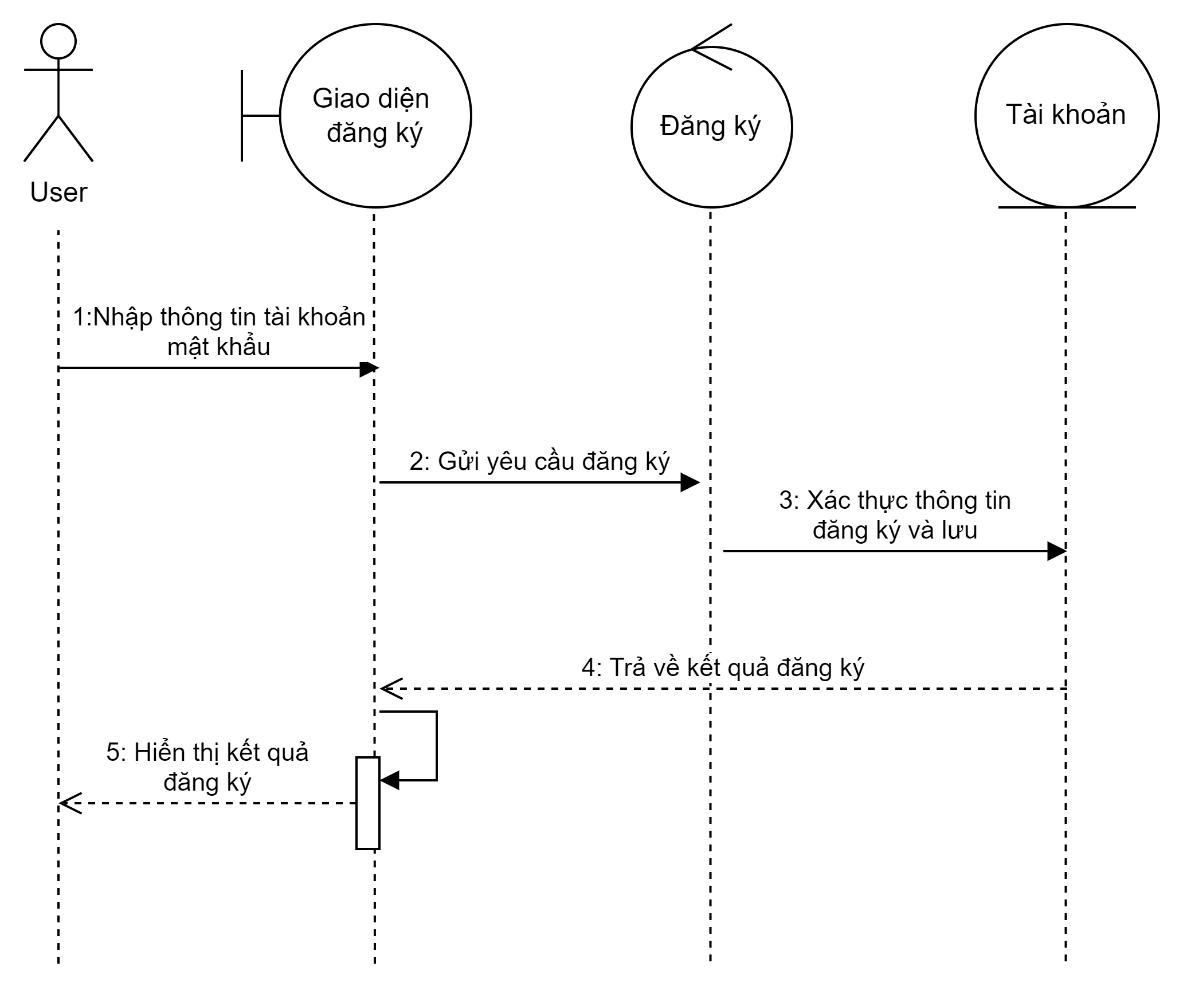
**Dòng sự kiện chính:**

1. Quản trị viên truy cập vào phần quản lý tag.
2. Quản trị viên có thể thực hiện một trong các hành động sau:
3. Thêm tag:
4. Quản trị viên chọn tùy chọn "Thêm tag".
5. Quản trị viên nhập thông tin về tag mới (tên, mô tả).
6. Hệ thống lưu tag mới vào cơ sở dữ liệu.
7. Xóa tag:
8. Quản trị viên chọn một tag cần xóa từ danh sách tag.
9. Quản trị viên chọn tùy chọn "Xóa tag".
10. Hệ thống xác nhận việc xóa.
11. Hệ thống xóa tag khỏi cơ sở dữ liệu.
12. Sửa tag:
13. Quản trị viên chọn một tag từ danh sách tag.
14. Quản trị viên chọn tùy chọn "Sửa tag".
15. Quản trị viên thay đổi thông tin của tag (tên, mô tả).
16. Hệ thống cập nhật tag trong cơ sở dữ liệu.
17. Hệ thống thông báo kết quả hành động.

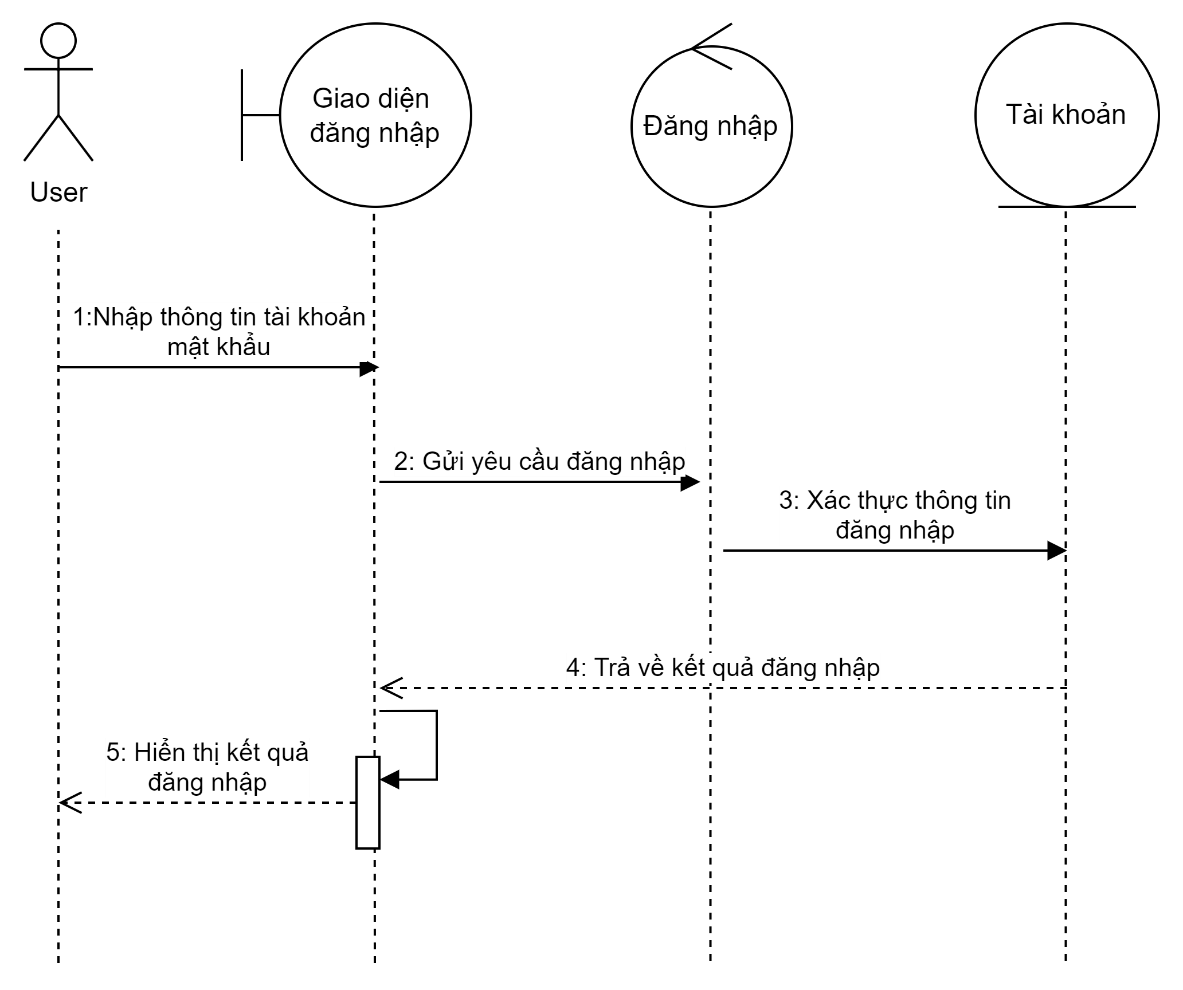
**Kết quả:** Tag được thêm, xóa, hoặc sửa thành công.

### 3.3.5 Sơ đồ tuần tự

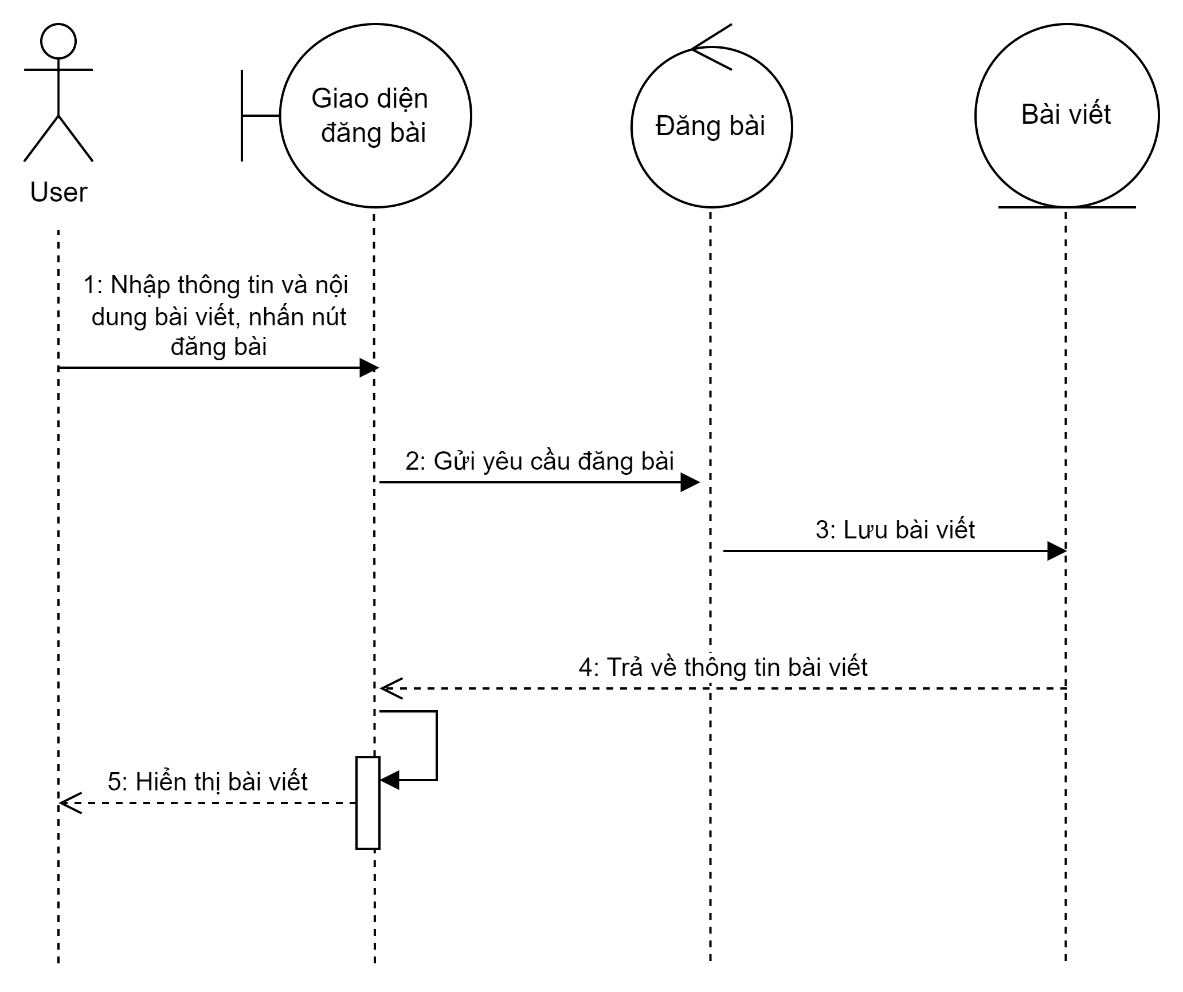
#### 3.3.5.1 Sơ đồ tuần tự chức năng đăng ký



#### 3.3.5.2 Sơ đồ tuần tự chức năng đăng nhập



#### 3.3.5.3 Sơ đồ tuần tự chức năng đăng bài



**Quy trình Đăng Bài**

1. Nhập thông tin: Người dùng nhập thông tin và nội dung bài viết vào giao diện đăng bài.

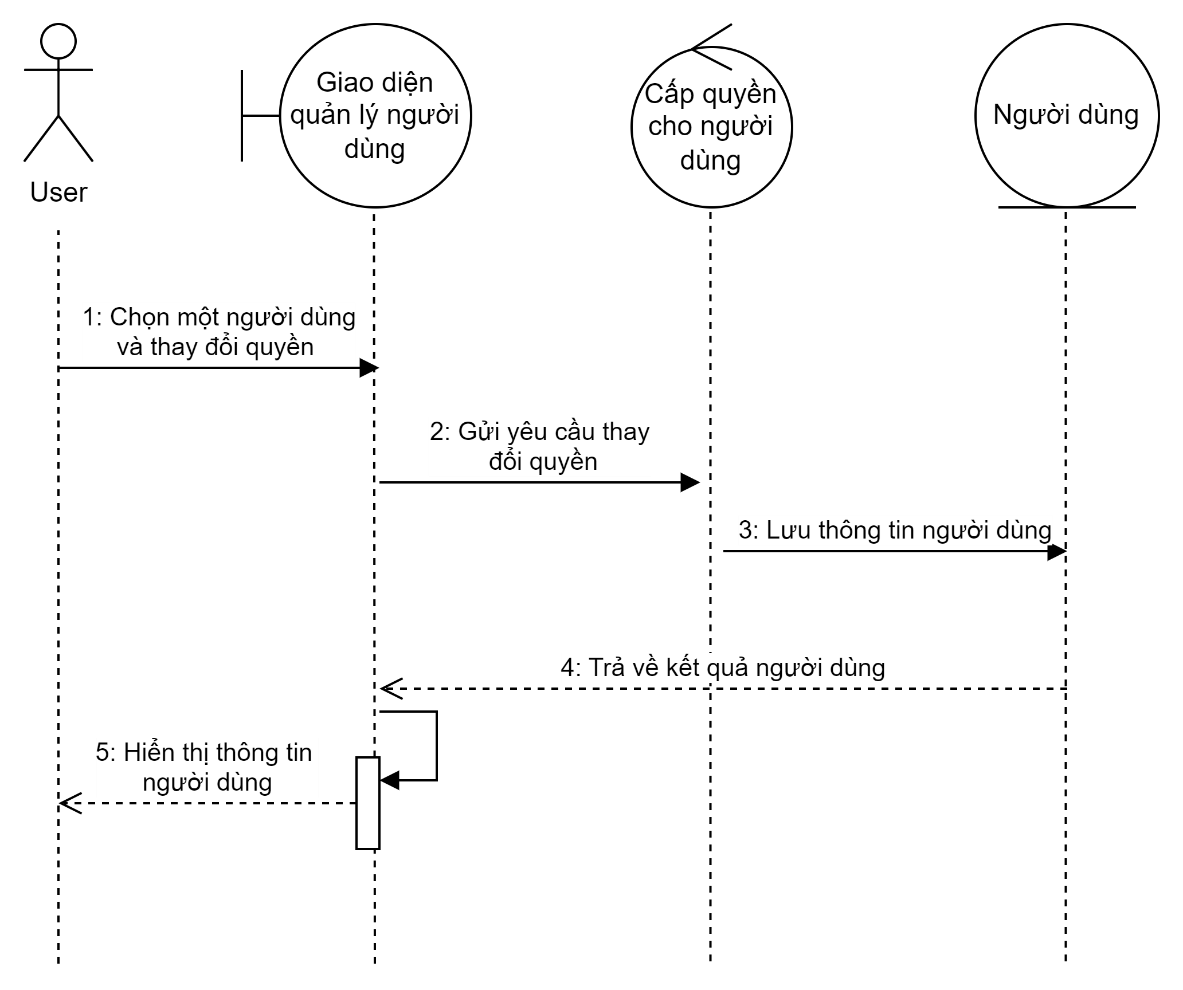
2. Gửi yêu cầu đăng bài: Sau khi nhập thông tin, người dùng nhấn nút để gửi yêu cầu đăng bài.

3. Lưu bài viết: Hệ thống lưu bài viết vào cơ sở dữ liệu hoặc hệ thống quản lý nội dung.

4. Trả về thông tin bài viết: Hệ thống trả lại thông tin về bài viết vừa được đăng.

5. Hiển thị bài viết: Bài viết sau đó được hiển thị trên giao diện người dùng.

#### 3.3.5.4 Sơ đồ tuần tự chức năng cấp quyền cho người dùng



**Quy trình Quản Lý Người Dùng**

1. Chọn một người dùng: Người quản trị chọn một người dùng và xác định quyền hạn cần thay đổi.

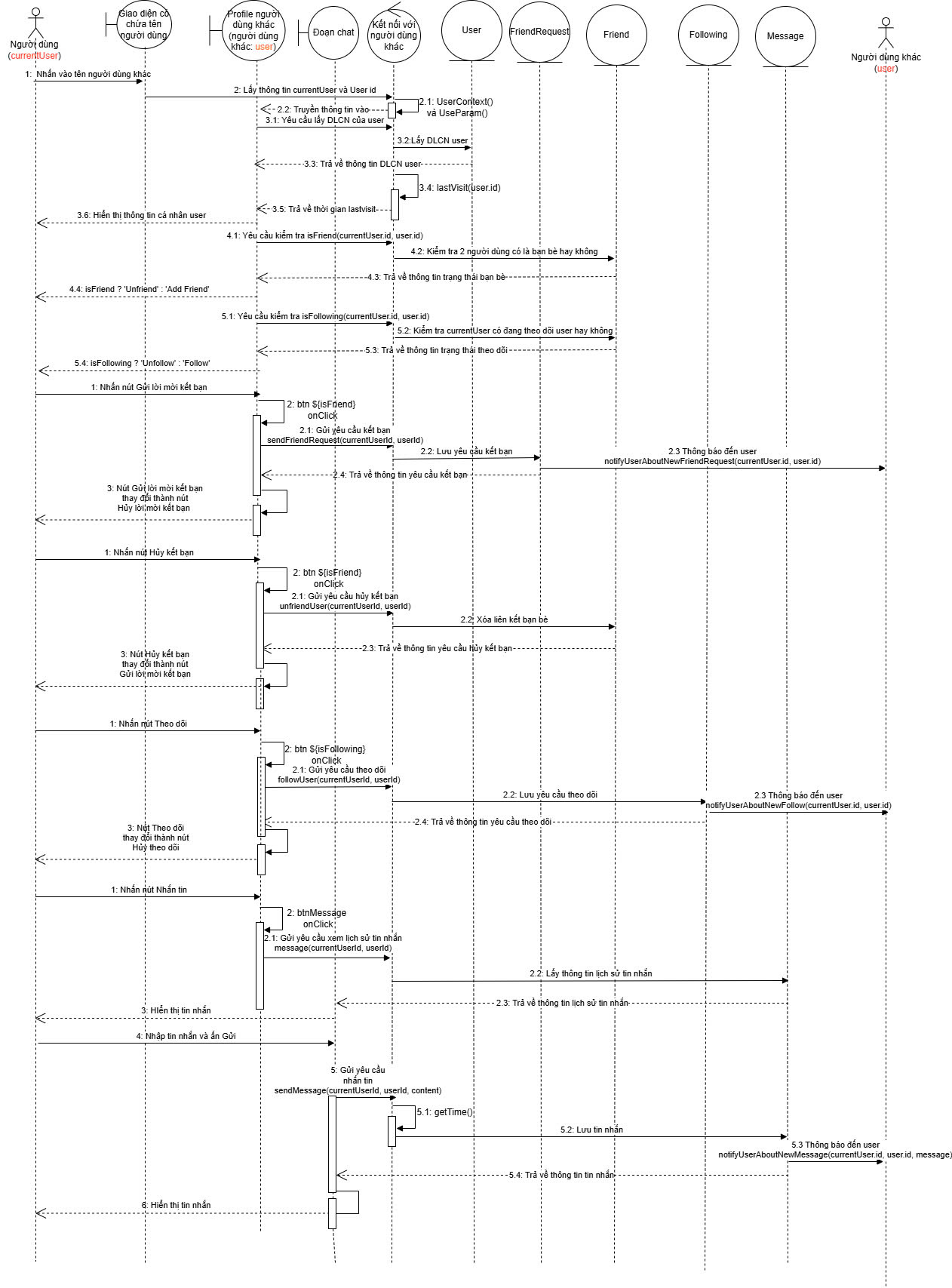
2. Gửi yêu cầu thay đổi quyền: Yêu cầu thay đổi quyền được gửi đến hệ thống.

3. Lưu thông tin người dùng: Hệ thống lưu thông tin về quyền hạn của người dùng.

4. Trả về kết quả người dùng: Hệ thống trả về kết quả về việc thay đổi quyền.

5. Hiển thị thông tin người dùng: Thông tin người dùng, bao gồm các quyền hạn hiện tại, được hiển thị cho người quản trị.

#### 3.3.5.5 Sơ đồ tuần tự chức năng kết nối người dùng



## 3.4 Xác định thực thể

- USER(**ID**, TEN, USERNAME, PASSWORD, EMAIL, NGAYTAO)

- ROLE(**ID**, TEN)

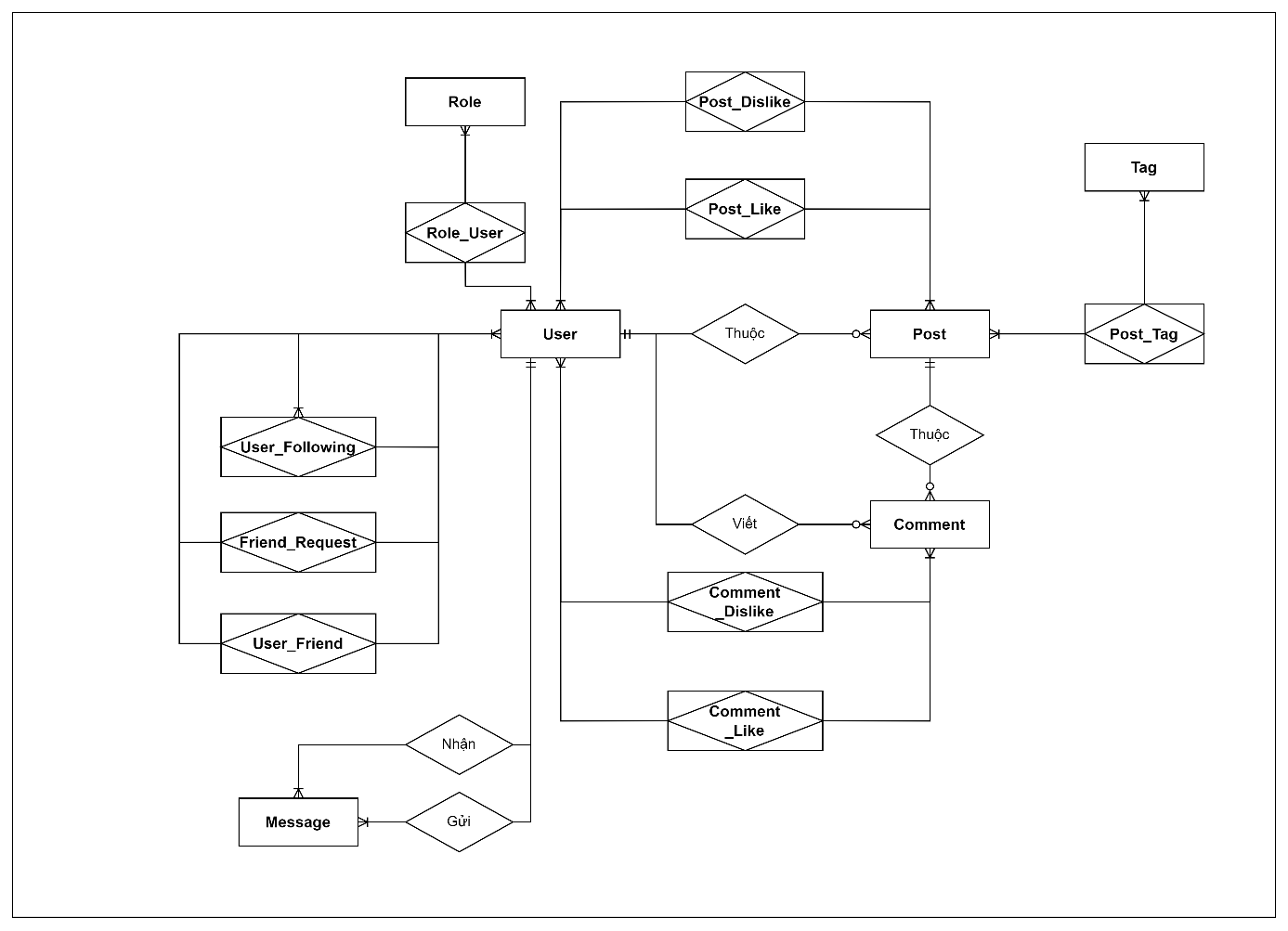
- POST(**ID**, IDUSER, TIEUDE, NOIDUNG, NGAYTAO)

- TAG(**ID**, TEN)

- MESSAGE(**ID**, IDNGUOIGUI, IDNGUOINHAN, NOIDUNG, NGAYGUI)

- COMMENT(**ID**, IDPOST, IDUSER, NOIDUNG, IDCHA, NGAYTAO)

## 3.5 Mô hình ERD



## 3.6 Mô hình cơ sở dữ liệu quan hệ

- KHÓA CHÍNH: **IN ĐẬM GẠCH CHÂN**

- KHÓA NGOẠI: GẠCH CHÂN

- USER(**ID**, TEN, USERNAME, PASSWORD, EMAIL, NGAYTAO)

- ROLE(**ID**, TEN)

- USER\_ROLE(IDUSER, IDROLE)

- POST(**ID**, IDUSER, TIEUDE, NOIDUNG, NGAYTAO)

- TAG(**ID**, TEN)

- MESSAGE(**ID**, IDNGUOIGUI, IDNGUOINHAN, NOIDUNG, NGAYGUI)

- POST\_TAG(IDPOST, IDTAG)

- POST\_DISLIKE(IDPOST, IDUSER)

- POST\_LIKE(IDPOST, IDUSER)

- COMMENT(**ID**, IDPOST, IDUSER, NOIDUNG, IDCHA, NGAYTAO)

- COMMENT\_LIKE(IDCOMMENT, IDUSER)

- COMMENT\_DISLIKE(IDCOMMENT, IDUSER)

- FRIEND\_REQUEST(IDNGUOIGUI, IDNGUOINHAN)

- USER\_FRIEND(IDUSER, IDFRIEND)

- USER\_FOLLOWING(IDUSER, IDUSERFOLLOW)

* Thực thể USER: Thông tin user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | ID | Id | bigint | Khóa chính | X |
| 2 | TEN | Tên người dùng | varchar(255) |  | X |
| 3 | USERNAME | Tên đăng nhập | varchar(20) |  | X |
| 4 | PASSWORD | Mật khẩu | varchar(64) |  | X |
| 5 | EMAIL | Email | varchar(320) |  |  |
| 6 | NGAYTAO | Ngày tạo | datetime |  | X |

* Thực thể ROLE: Thông tin các quyền

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | ID | Id | bigint | Khóa chính | X |
| 2 | TEN | Tên quyền | varchar(20) |  | X |

* Thực thể USER\_ROLE: Quyền của user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | IDUSER | Id user | bigint | Khóa ngoại | X |
| 2 | IDROLE | Id quyền | bigint | Khóa ngoại | X |

* Thực thể POST: Thông tin bài viết

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | ID | Id bài viết | bigint | Khóa chính | X |
| 2 | IDUSER | Id người đăng | bigint | Khóa ngoại | X |
| 3 | TIEUDE | Tiêu đề | varchar(80) |  | X |
| 4 | NOIDUNG | Nội dung | text |  | X |
| 5 | NGAYTAO | Ngày tạo | datetime |  | X |

* Thực thể TAG: Thông tin tag

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | ID | Id | bigint | Khóa chính | X |
| 2 | TEN | Tên tag | varchar(255) |  | X |

* Thực thể MESSAGE: Thông tin tin nhắn

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | ID | Id tin nhắn | bigint | Khóa chính | X |
| 2 | IDNGUOIGUI | Id người gửi | bigint | Khóa ngoại | X |
| 3 | IDNGUOINHAN | Id người nhận | bigint | Khóa ngoại | X |
| 4 | NOIDUNG | Nội dung | text |  | X |
| 5 | NGAYTAO | Ngày tạo | datetime |  | X |

* Thực thể POST\_TAG: Tag của bài viết

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | IDPOST | Id bài viết | bigint | Khóa ngoại | X |
| 2 | IDTAG | Id tag | bigint | Khóa ngoại | X |

* Thực thể POST\_LIKE: Like của bài viết

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | IDPOST | Id bài viết | bigint | Khóa ngoại | X |
| 2 | IDUSER | Id người like | bigint | Khóa ngoại | X |

* Thực thể POST\_DISLIKE: Dislike của bài viết

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | IDPOST | Id bài viết | bigint | Khóa ngoại | X |
| 2 | IDUSER | Id người dislike | bigint | Khóa ngoại | X |

* Thực thể COMMENT: Thông tin bình luận

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | ID | Id bình luận | bigint | Khóa chính | X |
| 2 | IDUSER | Id người đăng | bigint | Khóa ngoại | X |
| 3 | IDCHA | Id bình luận cha | bigint | Khóa ngoại | X |
| 4 | NOIDUNG | Nội dung | text |  | X |
| 5 | NGAYTAO | Ngày tạo | datetime |  | X |

* Thực thể COMMENT\_LIKE: Like của bình luận

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | IDCOMMENT | Id bình luận | bigint | Khóa ngoại | X |
| 2 | IDUSER | Id người like | bigint | Khóa ngoại | X |

* Thực thể COMMENT\_DISLIKE: Dislike của bình luận

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | IDCOMMENT | Id bình luận | bigint | Khóa ngoại | X |
| 2 | IDUSER | Id người dislike | bigint | Khóa ngoại | X |

* Thực thể FRIEND\_REQUEST: Lời mời kết bạn

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | IDNGUOIGUI | Id người gửi | bigint | Khóa ngoại | X |
| 2 | IDNGUOINHAN | Id người nhận | bigint | Khóa ngoại | X |

* Thực thể USER\_FRIEND: Bạn bè

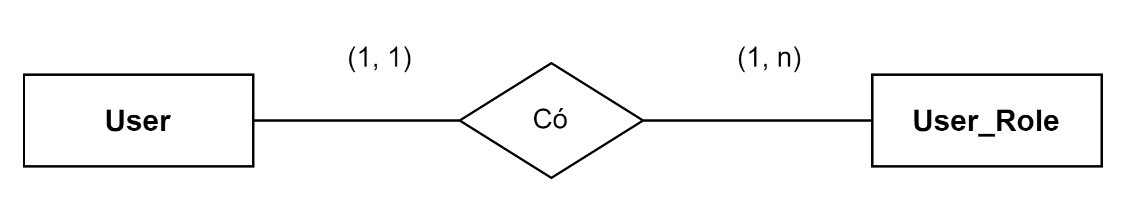
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | IDUSER1 | Id người dung | bigint | Khóa ngoại | X |
| 2 | IDFIEND | Id bạn bè | bigint | Khóa ngoại | X |

* Thực thể USER\_FOLLOWING: Theo dõi

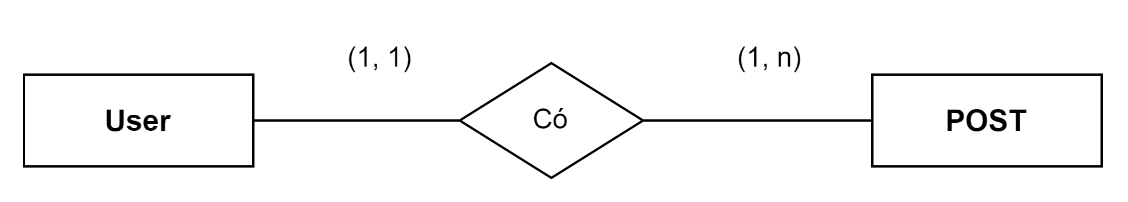
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** | **Notnull** |
| 1 | IDUSER | Id người dung | bigint | Khóa ngoại | X |
| 2 | IDUSERFOLLOW | Id người người follow | bigint | Khóa ngoại | X |

## 3.7 Mô hình dữ liệu

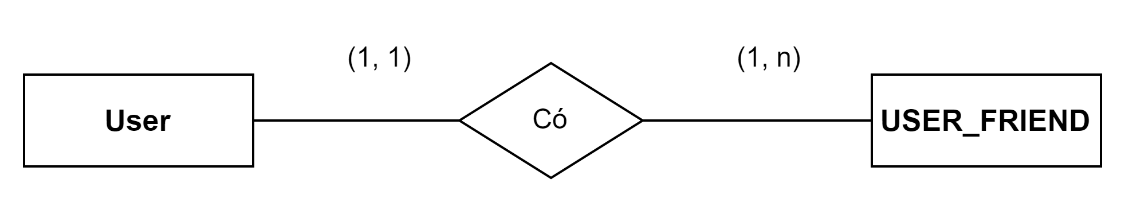
### 3.7.1 Xét quan hệ USER-USER\_ROLE



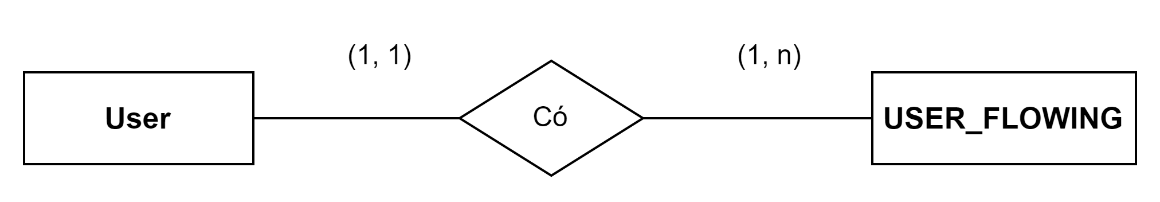
### 3.7.2 Xét quan hệ USER-POST



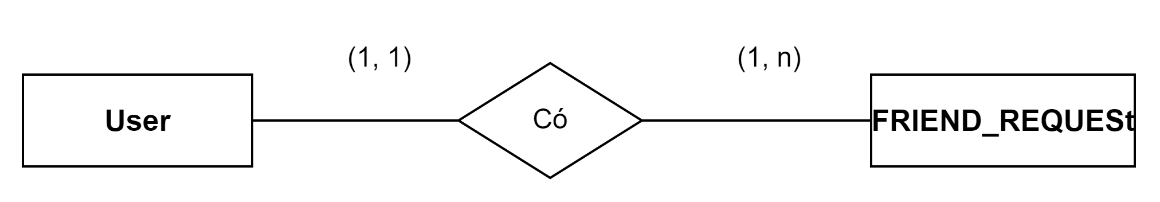
### 3.7.3 Xét quan hệ USER-USER\_FRIEND



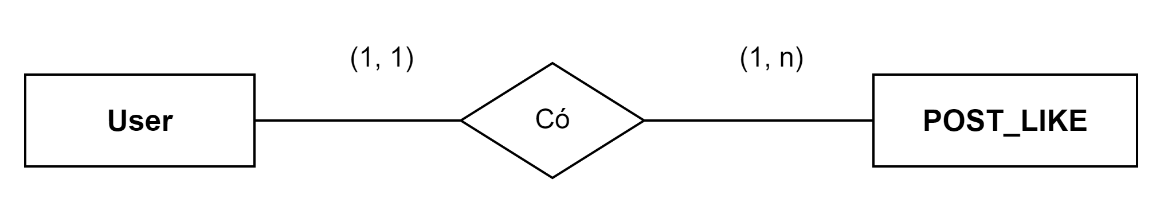
### 3.7.4 Xét quan hệ USER-USER\_FOLLOWING



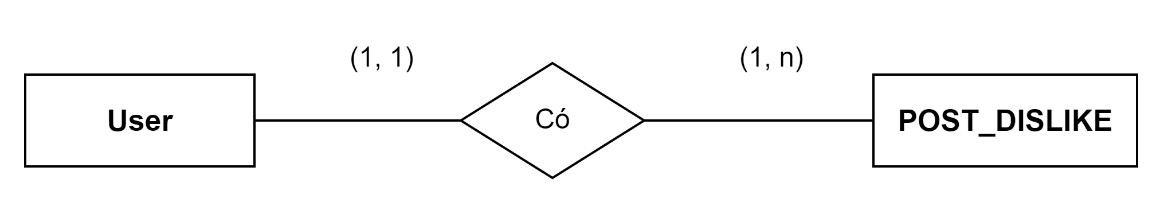
### 3.7.5 Xét quan hệ USER-FRIEND\_REQUEST



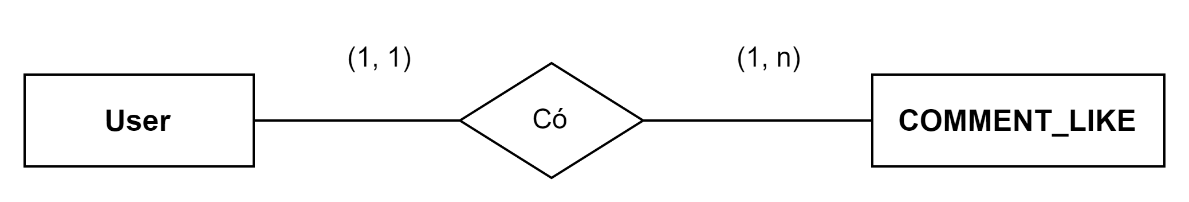
### 3.7.6 Xét quan hệ USER-POST\_LIKE



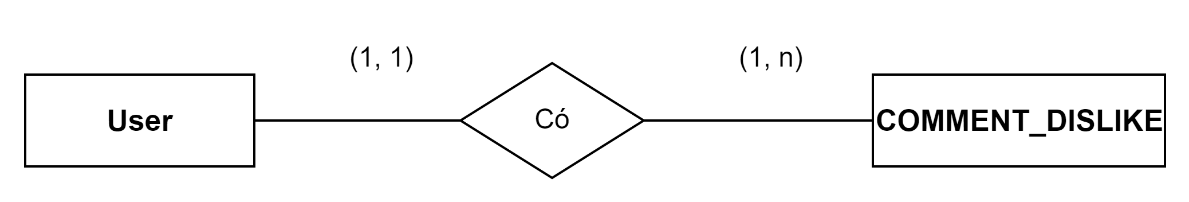
### 3.7.7 Xét quan hệ USER-POST\_DISLIKE



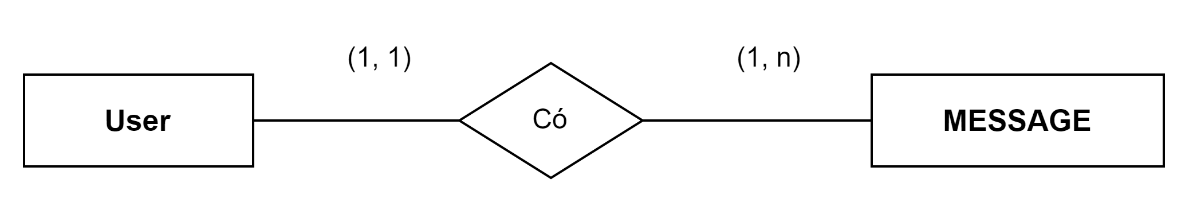
### 3.7.8 Xét quan hệ USER-COMMENT\_LIKE



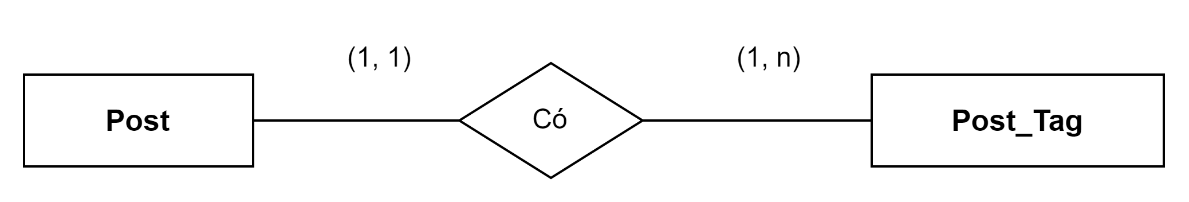
### 3.7.9 Xét quan hệ USER-COMMENT\_DISLIKE



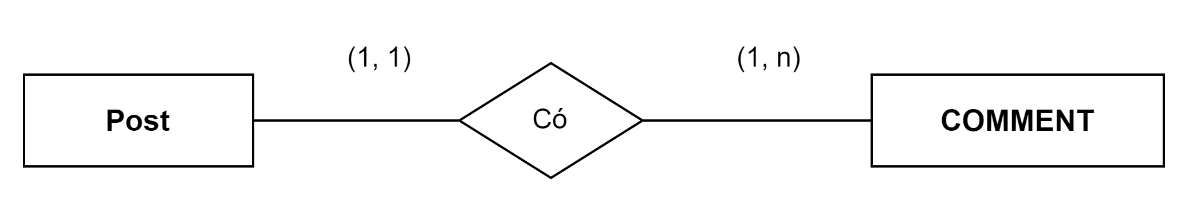
### 3.7.10 Xét quan hệ USER-MESSAGE



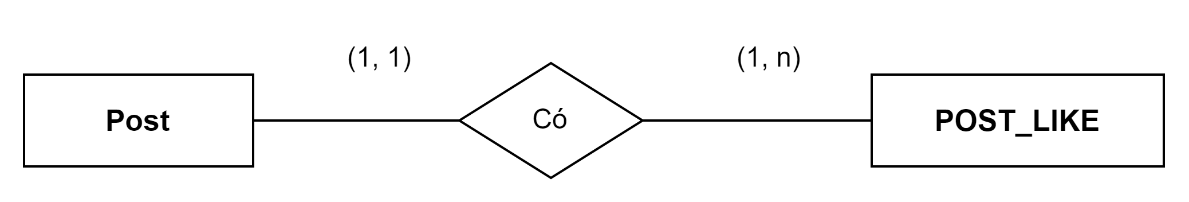
### 3.7.10 Xét quan hệ POST-POST\_TAG



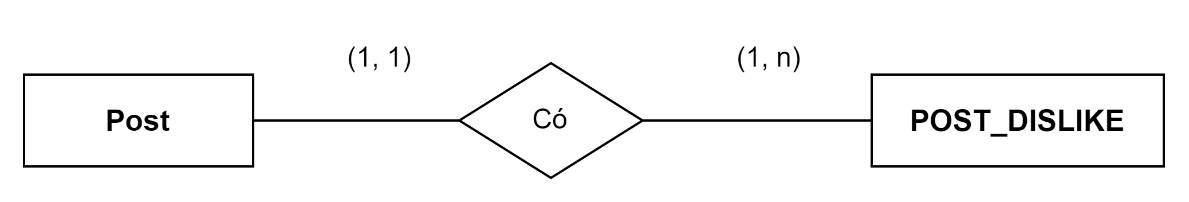
### 3.7.10 Xét quan hệ POST-COMMENT



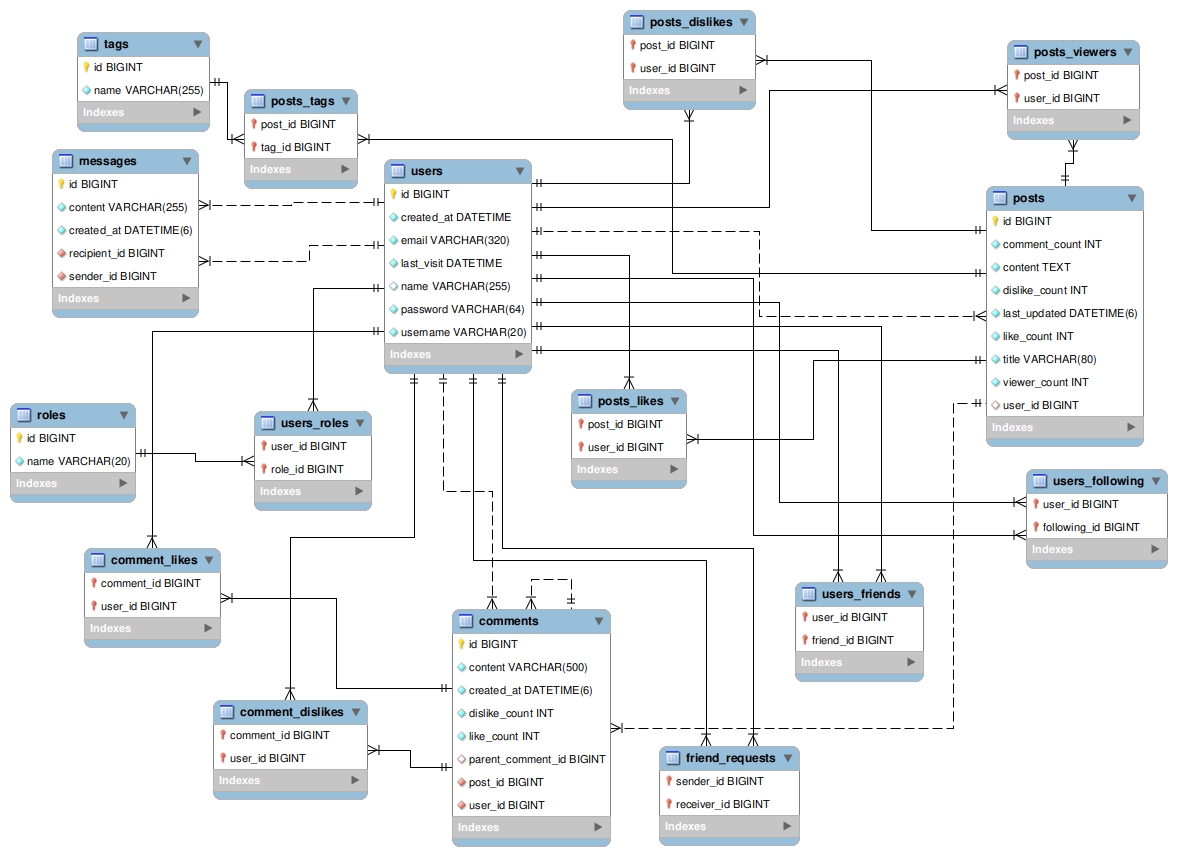
### 3.7.10 Xét quan hệ POST-POST\_LIKE



### 3.7.10 Xét quan hệ POST- POST\_DISLIKE



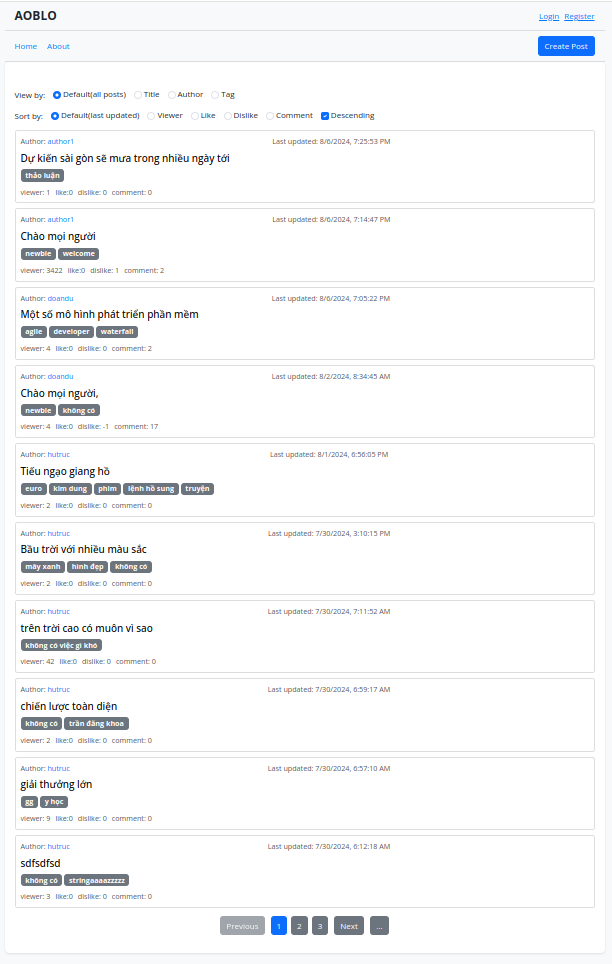
## 3.8 Phân tích và hoàn thiện mô hình quan hệ



# CHƯƠNG 4: GIAO DIỆN

**4.1. Giao diện đối với user chưa tạo tài khoản**

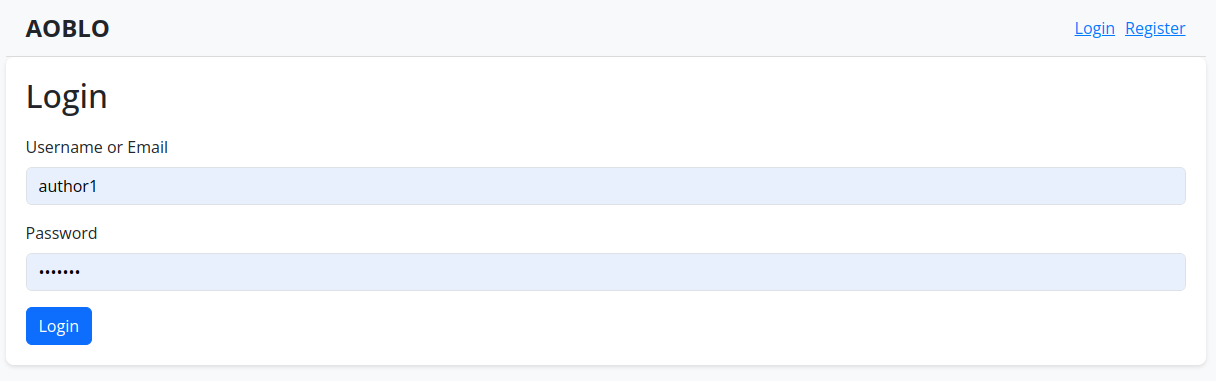
**4.1.1. Trang chủ**

Hình 4.1 – Hình trang chủ

Trang chủ nơi chứa danh sách tất cả bài đăng của tất cả các tác giả được sắp xếp theo thời gian. Trang chủ được thiết kế theo phong cách tối giản, phù hợp với giao diện mà một trangweb blog nên có. Ở trên thanh header có chứa nút create post được làm nổi bật để làm bật lên 2 chức năng chủ đạo của blog là viết bài và xem bài viết.

Về thông tin của mỗi bài đăng thì được hiển thị trong danh sách thì có lượng thông tin vừa đủ, chứa tác giả(có gắn link đến tác giả), bên cạnh có chứa thời gian cập nhật lần cuối của bài đăng được đặt cùng dòng được căn lề, màu sắc, kích cỡ rất phù hợp, không làm nhiễu thông tin quan trong nhất đó là tiêu đề của bài viết, tiêu đề của bài viết được in đâm, có kích thước lớn, ngay dưới tiêu đề là danh sách các tags được gắn với tiêu đề đó, giúp người xem hình dung ngay được bài viết liên quan đến những keyword nào. Ngoài ra các thông tin như số lượt người xem bài đăng, số lượt like, số lượt dislike, số lượng comment, cũng phản ánh phần nào sự thu hút của bài viết và chất lượng của bài viết đó để tác giả đưa ra lựa chọn có nên xem không.

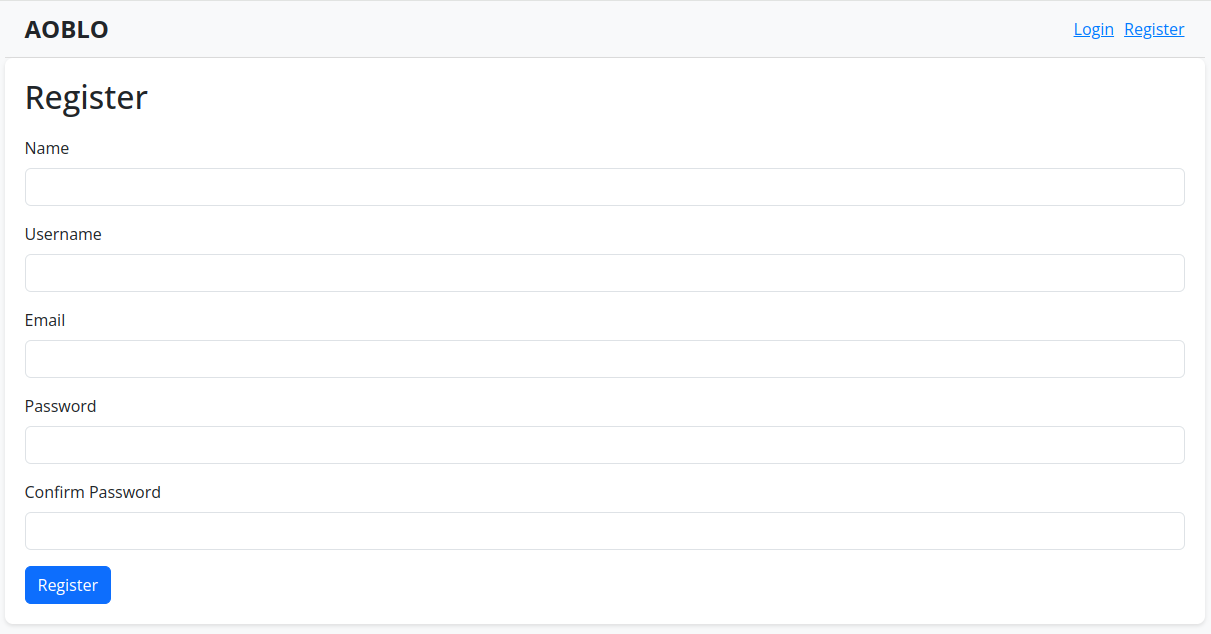
Tính năng có phần nổi bật hơn cả đó là filter có thể sắp xếp theo nhiều option, tìm kiếm theo tiêu đề, tác giả, tag rất thân thiên với người dùng, làm bật lên sự thực dụng của việc lựa chọn thiết kế giao diện theo phong cách này.

**4.1.2. Trang đăng nhập**

Hình 4.2 – Hình trang đăng nhập

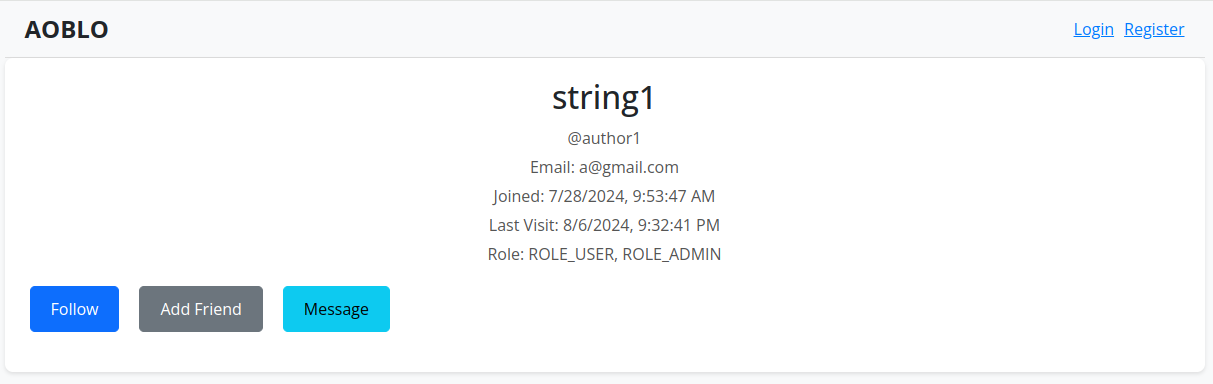
Với trang đăng nhập này bạn có thể dùng username hoặc email để đăng nhập, rất tiện lợi, nếu không may chỉ nhớ được 1 trong 2 thông tin đó thì bạn vẫn có thể đăng nhập. Về phần giao diện thì vẫn đồng bộ với các trang khác, theo phong cách tối giản mà một blog nên có.

**4.1.3. Trang đăng ký**

Hình 4.3 – Hình trang đăng ký

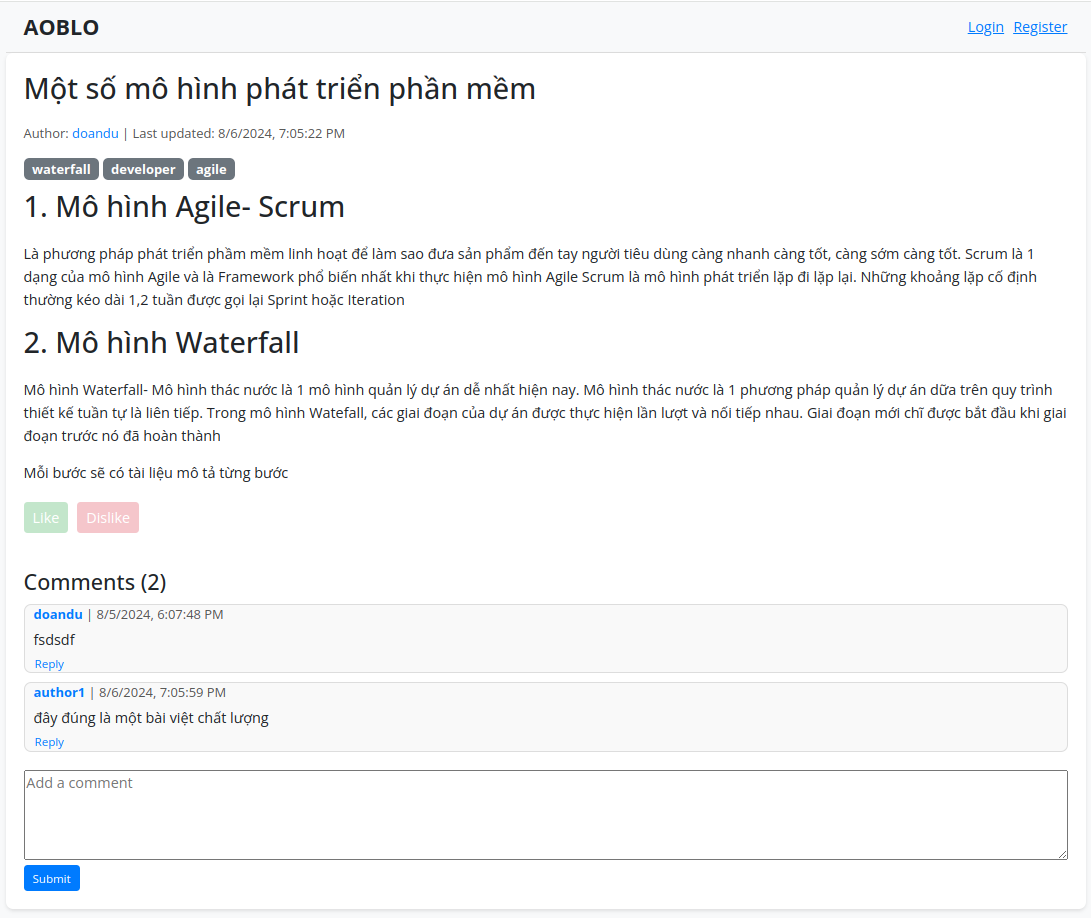
Trang đăng ký chỉ yêu cầu các thông tin rất cơ bản, có confirm password để tránh tính trạng lỡ tay gõ không đúng password mong muốn, và nếu bạn cũng không lưu password vào trình duyệt thì sẽ mất thời gian để có thể login. Về thiết kế giao diện cũng đồng bộ với trang login.

**4.1.4. Trang thông tin tác giả bài viết**

Hình 5 – Hình trang thông tin tác giả bài viết

Trang thông tin tác giả với giao diện hết sức đơn giản, thông tin nổi bật có lưu lại thời gian đăng nhập lần cuối là khi nào, và người đó có role là gì. Ba tính năng có trong trang này là follow, add friend, message. Nếu bạn chưa đăng nhập thì sẽ không thể thực hiện 3 tính năng này, tuy nhiên bạn nhìn thấy các nút tương ứng với các tính năng này để biết là blog có những tính năng đó, và nếu bạn muốn dùng các tính năng này thì hãy login.

**4.1.5. Trang chi tiết bài viết**

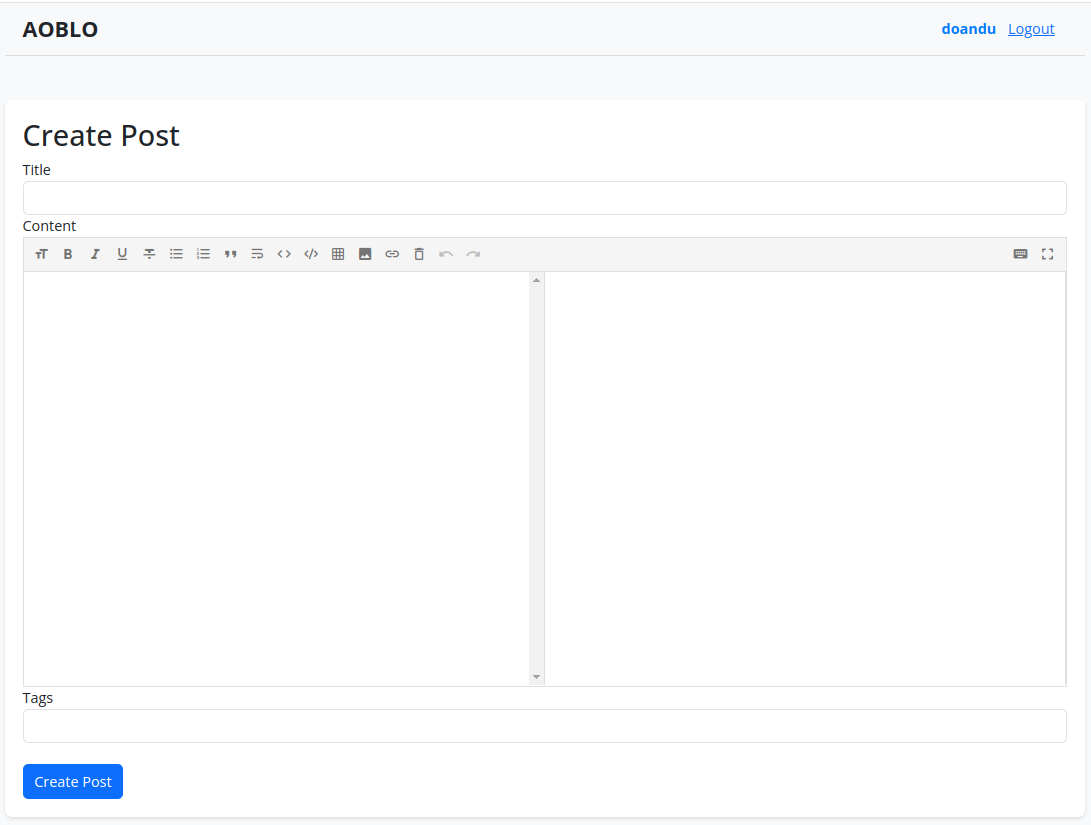
Hình 6 – Hình trang chi tiết bài viết

Trang chi tiết bài viết giao diện được thiết kế đơn giản, giống như các blog phổ biến khác hay có. Với việc bạn chưa login thì bạn có thể xem nội dung bài viết, đọc comment của người khác, xem thông tin tác giả. Với tính năng reply và comment thì bạn sẽ nhận được thông báo yêu cầu đăng nhập.

Điểm nổi bật ở trong chi tiết bài đăng, đó là sử dụng comment cây, tức là các comment sẽ hiển thị theo dạng cây, mỗi comment sẽ có parrent comment là null hoặc một comment khác. Comment con sẽ thụt vào so với comment cha giúp cho việc trao đổi thông tin suôn sẻ dễ hiểu hơn.

**4.2. Giao diện đối với user có ROLE\_USER**

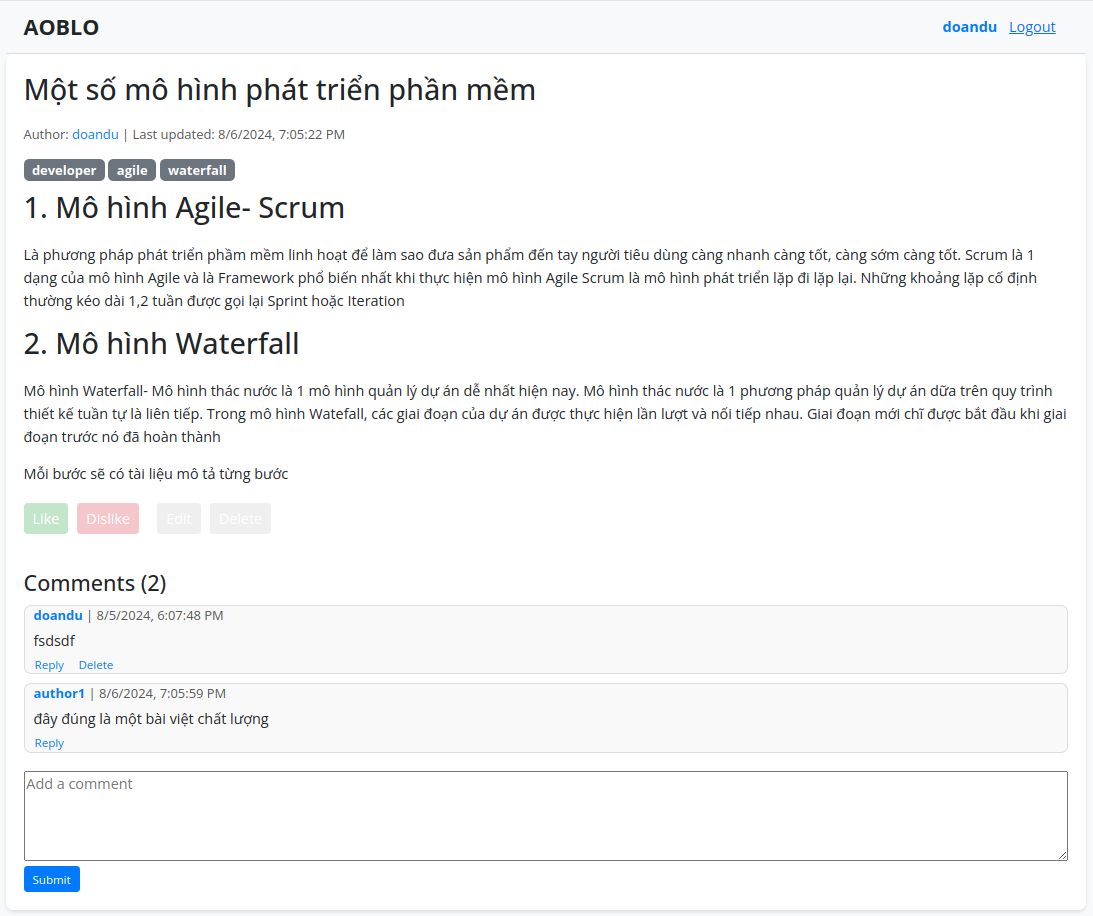
**4.2.1. Trang tạo bài đăng mới**

Hình 4.6 – Hình trang tạo bài viết

Với trang tạo bài đăng này, có 2 điểm đặc biệt: một là về trình soạn thảo nội dung, hai là tính năng gắn tag. Về trình soạn thảo thì có tích hợp soạn thảo markdown, bên trái là cú pháp markdown và bạn có preview ở bên phải. Với markdown bài viết sẽ có chất lượng hiển thị tốt hơn, có tiêu đề, có bảng, có hình ảnh, có link giúp bạn dễ dàng trong việc biểu đạt nội dung muốn truyền tải.

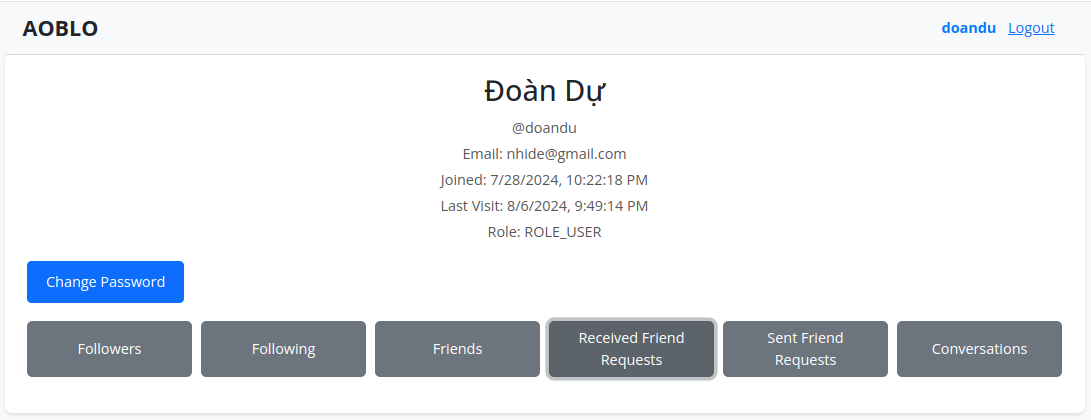
Về gắn tag thì khi bạn gõ vào thanh tag, nó sẽ tìm ra tối đa 5 tag liên quan đến nội dung bạn gõ, để chọn tag đó bạn có thể Enter hoặc gõ dấu “,” nó sẽ tự động chọn tag đó. Nếu bạn muốn bỏ tag đã chọn thì chỉ cần nhấp vào dấu x bên phải tag là được. Ngoài ra nó cũng chặn việc bạn thêm 2 tag giống nhau vào một bài viết.

**4.2.2. Trang chi tiết bài đăng với nhiều tính năng hơn**

Hình 4.7 – Hình trang chi tiết bài đăng (ROLE\_USER)

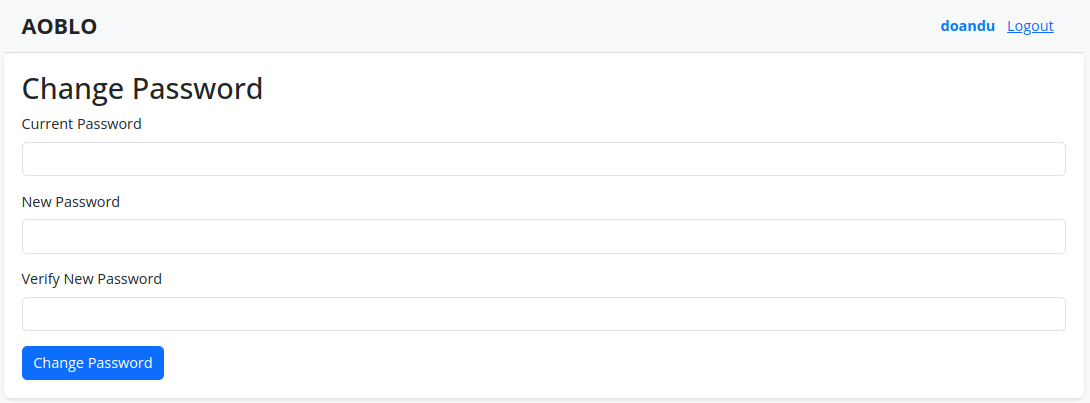
Có thêm tính năng Like, Dislike, Comment, Reply... Ngoài ra có thêm tín năng edit bài viết, xóa bài viết nếu bạn là tác giả. Tương tự có thể xóa comment nếu bạn là người viết comment đó. Khi bạn xóa comment thì các comment con của comment đó cũng sẽ biến mất theo, điều này rất phù hộp khi sử dụng comment cây.

**4.2.3. Trang thông tin cá nhân của chính mình**

Hình 4.8 – Hình trang thông tin cá nhân(ROLE\_USER)

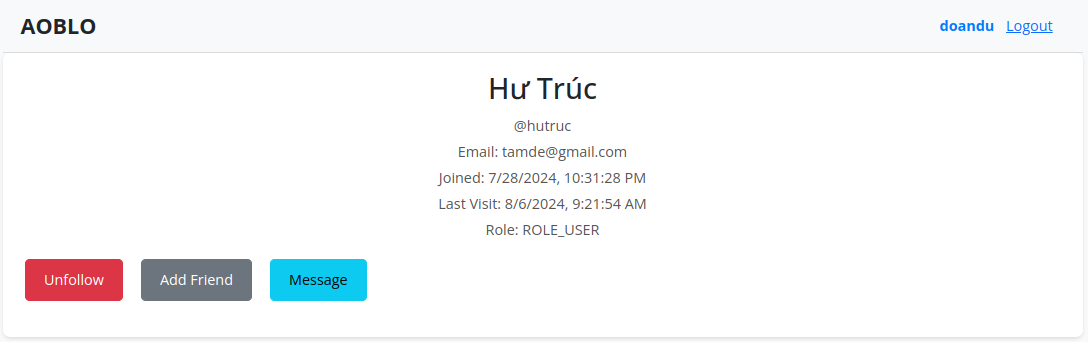
Trong trang thông tin cá nhân của chính mình bạn có các tính năng đổi mật khẩu, xem ai đang follow bạn, bạn đang follow ai, bạn đã kết bạn với ai rồi, ai đã gửi lời mời kết bạn đến bạn, bạn đã gửi lời mời kết bạn cho ai, có các cuộc hội thoại nào giữa bạn và người khác. Rất tiện lợi để nắm thông tin.

**4.2.4. Trang đổi mật khẩu**

Hình 4.9 – Hình trang đổi mật khẩu

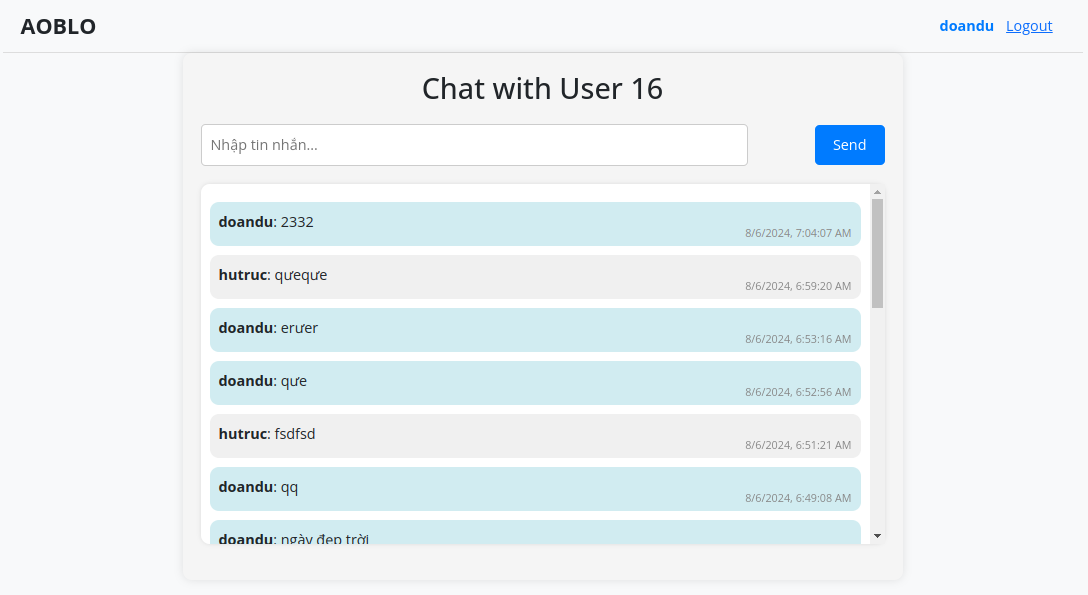
Như thông thường, trang đổi mật khẩu sẽ yêu cầu bạn nhập mật khẩu hiện tại. Đây là điều cần thiết để tránh việc bạn đang treo máy và ai đó vào đổi mật khẩu của bạn, chẳng hạn như vậy, và nhiều hữu ích trong nhiều tính huống khác.

**4.2.5. Trang thông tin của user khác với nhiều tín năng hơn**

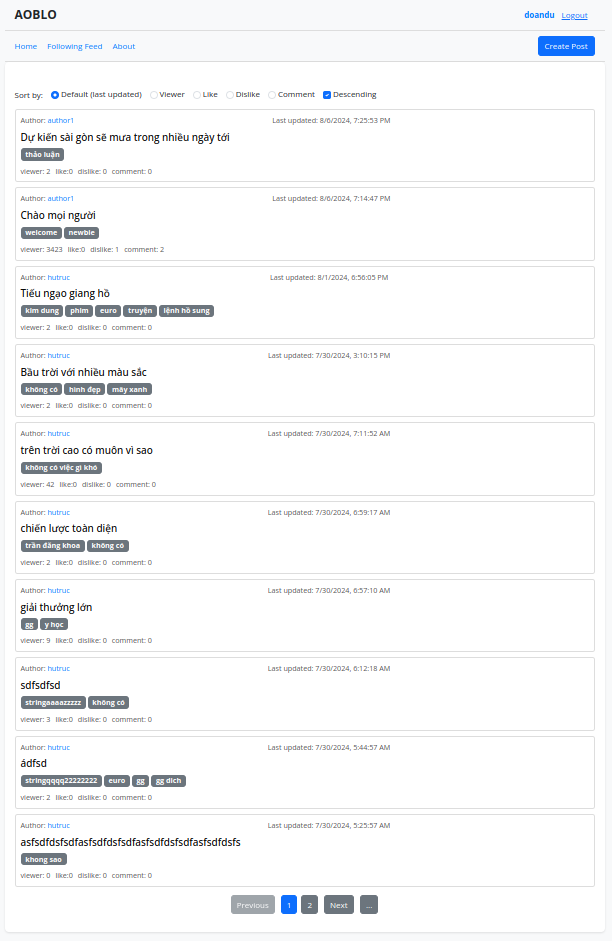
Hình 4.10 – Hình trang thông tin của user khác (ROLE\_USER)

Trang thông tin tác giả với giao diện hết sức đơn giản, thông tin nổi bật có lưu lại thời gian đăng nhập lần cuối là khi nào, và người đó có role là gì. Ba tính năng có trong trang này là follow, add friend, message. Với follow, nếu bạn follow tác giả này thì bài viết của tác giả này sẽ nằm ở trang following feed để bạn tiện theo dõi. Về add friend, bạn có thể gửi lời mời kết bạn, nếu cả 2 đã là bạn thì nút này sẽ đổi thành unfriend, và khi bạn click vào sẽ thực hiện tín năng hủy kết bạn, thay vì gửi lời mời kết bạn như hiện tại. Với Message, khi click vào bạn sẽ được chuyển tới trang nhắn tin với người đó, nơi đây có thể nhắn tin và xem lịch sử tin nhắn của 2 người.

**4.2.6. Trang nhắn tin với user khác**

Hình 4.11 – Hình trang nhắn tin với user khác

Bạn có thể nhắn tin realtime với đối phương, đối phương cũng sẽ nhận được tin nhắn khi cả 2 đều đang mở cuộc hội thoại này. Trong trường hợp chỉ có 1 người online cũng không sao cả, các tin nhắn đều được lưu vào database, đảm bảo bạn không bỏ sót tin nhắn nào, bạn sẽ vẫn thấy các tin nhắn khi vào lại cuộc hội thoại.

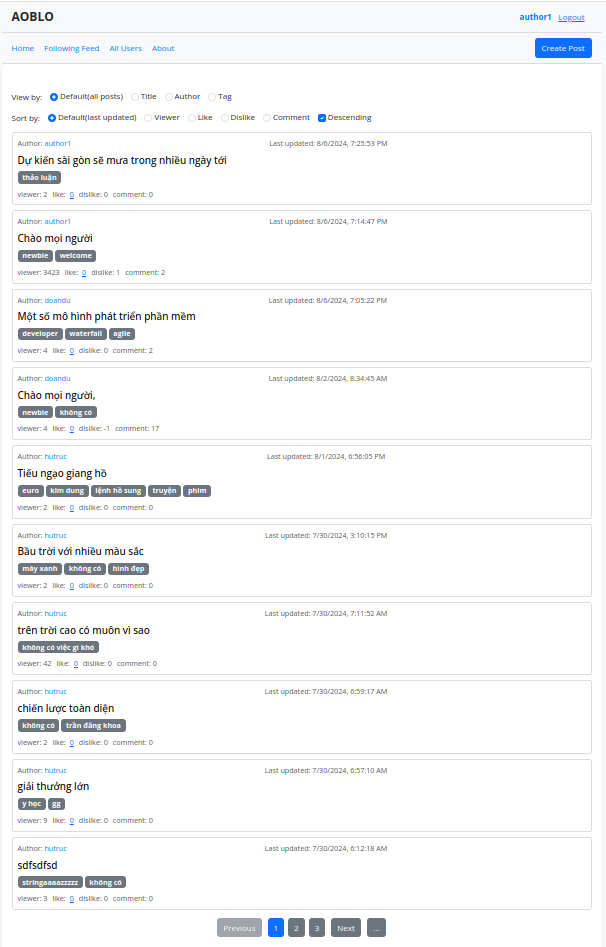
**4.2.7. Trang Following Feed - nơi hiển thị bài viết của tác giả yêu thích**

Hình 4.12 – Hình trang Following Feed

Sau khi bạn đã follow tác giả nào đó thì bạn có thể nhìn thấy bài đăng của họ bên trong Following Feed , nó rất tiện lợi phải không nào, nó có phần nào đó giống với bảng tin của facebook. Nó thật sự hữu ích, nó giúp bạn lọc nội dung tốt hơn là xem nội dung đủ các thể loại trên trang chủ, tất nhiên mỗi cái đều có ưu điểm riêng của nó.

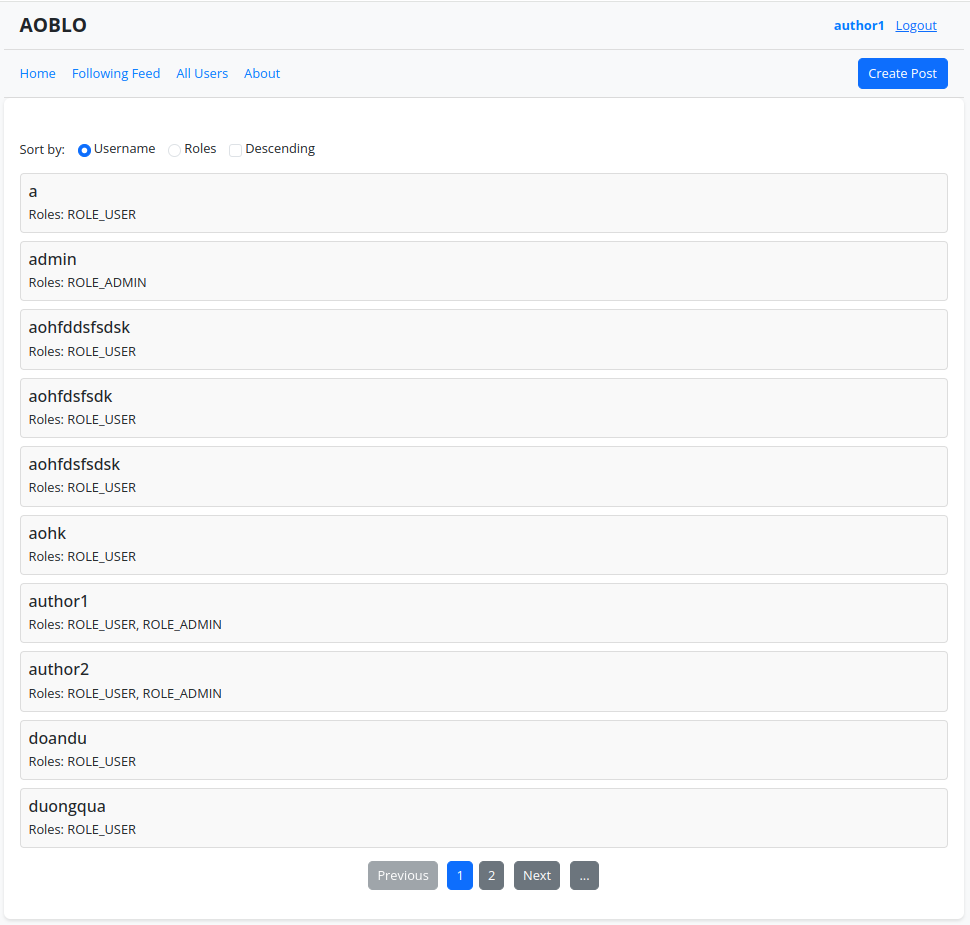
**4.3. Giao diện đối với user có ROLE\_ADMIN**

**4.3.1. Trang chủ với nhiều tín năng hơn**

Hình 4.13 – Hình trang chủ (ROLE\_ADMIN)

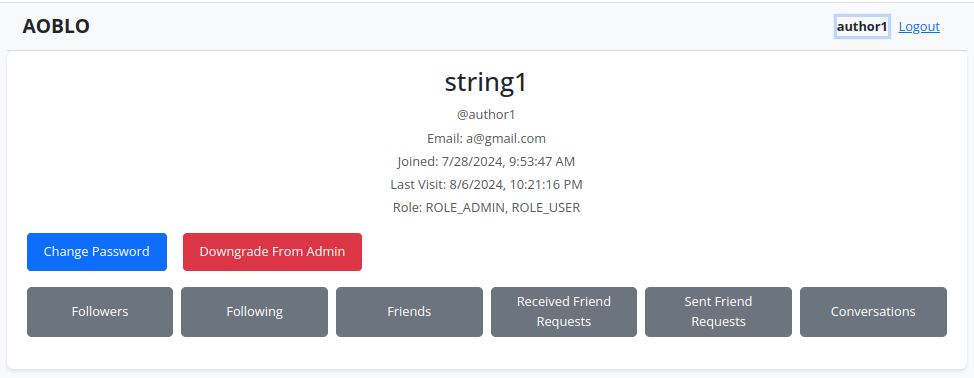
Tính năng mà ADMIN mới có đó là có thể xem danh sách users đã like bài đăng.

**4.3.2. Trang xem danh sách all users**

Hình 4.14 – Hình trang danh sách all users

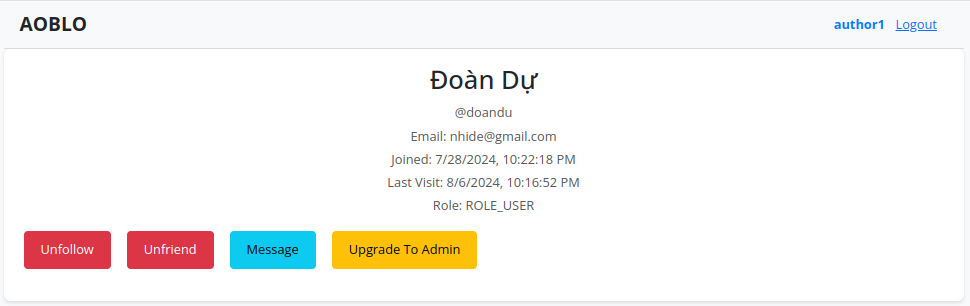
Chỉ có ROLE\_ADMIN mới có thể xem được trang tất cả users trong hệ thống. Điều này là cần thiết với ROLE\_ADMIN và nó không cần thiết với user thông thường, tôi nghĩ vậy. Trong trang này, bạn có thể sắp xếp để dễ dàng xem hơn.

**4.3.3. Trang thông tin cá nhân của chính mình với nhiều tín năng hơn**

Hình 4.15 – Hình trang thông tin cá nhân ROLE\_ADMIN

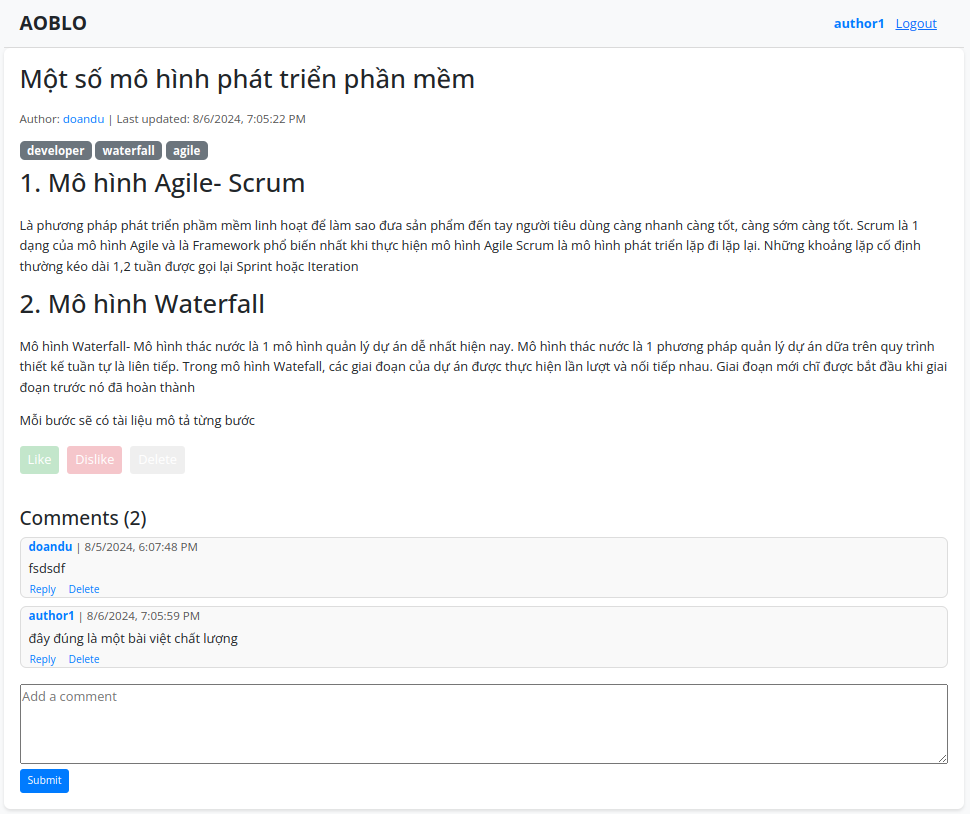
Tính năng được thêm vào đó là hạ quyền admin nếu bạn là admin. Việc nó xuất hiện ở đây là phù hợp.

**4.3.4. Trang thông tin cá nhân của user khác với nhiều tín năng hơn**

Hình 4.16 – Hình trang thông tin cá nhân của user khác (ROLE\_ADMIN)

Tính năng được bổ sung đó là nâng cấp quyền admin cho user. Việc tính năng này xuất hiện ở đây cũng là phù hợp.

**4.3.5. Trang chi tiết bài viết với nhiều tín năng hơn**

Hình 4.17 – Hình trang trang chi tiết bài viết (ROLE\_ADMIN)

ROLE\_ADMIN có thẻ xóa bài viết, comment của tất cả các user, điều này đảm bảo việc lọc nội dung rác giúp trangweb blog này chất lượng hơn.

**4.4. Đánh giá chung**

- Giao diện được thiết kế đơn giản, phù hợp với giao diện của một trang blog nên có.

- Giao diện các trang đồng bộ.

- Thác tác người dùng đơn giản mà không cần hướng dẫn sử dụng.

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**5.1. Kết luận**

Những chức năng chính theo yêu cầu của đề tài đã được hoàn thành.

Ứng dụng blog đáp ứng thời gian thực có tương đối đầy đủ các chức năng của một blog, và đặc biệt có realtime trong nhắn tin, điều mà không nhiều blog có. Giao diện đơn giản, dễ dùng, không làm người đọc rối, phân tâm.

Trangweb cung cấp môi trường để các tác giả viết bài, cung cấp môi trường để người đọc có nơi đọc nội dung. Có nơi để thảo luận, trao đổi về đủ các thể loại nội dung. Rất tiện lơi để trao đổi, kết nối, chia sẻ thông tin với nhau trực tuyến.

Trangweb còn giúp admin quản lý nội dung rác tiện lợi. Từ đó nâng cao chất lượng các bài viết trong ứng dụng, giúp cho trang web sớm trở thành nơi tra cứu uy tín, khi ngươi dùng cần nguồn tài liệu chất lượng để đọc, tham khảo. tìm hiểu.

**5.2. Hướng phát triển**

- Thêm tín năng donate cho tác giả, từ đó tác giả có thêm động lực viết bài, cung cấp các bài viết chất lượng.

- Thêm mục lục cho bài đăng để người đọc tiện theo dõi hơn

- Bổ sung thêm tính năng quên mật khẩu, cấp lại mật khẩu qua email.

- Bổ sung thời lượng dự kiến để đọc xong một bài đăng.

- Bổ sung thêm thông báo cho bất kì sự kiện nào liên quan đế user đó

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | “Spring,” [Online]. Available: https://spring.io/guides/gs/spring-boot. [Accessed 21 08 2024]. |
| [2] | Hach, "Spring Security 6 with JWT, Oauth2," Viblo, [Online]. Available: https://hocvienptit.edu.vn/tong-quan-ve-spring-security-va-cach-thuc-bao-mat-trong-ung-dung-java. [ Accessed 20 7 2024]. |
| [3] | N. H. Nam, "Hướng dẫn Spring Boot JPA + MySQL," Viblo, [Online]. Available: https://viblo.asia/p/spring-boot-11-huong-dan-spring-boot-jpa-mysql-GrLZD8dgZk0. [Accessed 21 8 2024]. |
| [4] | “SocketIO,” [Online]. Available: https://socket.io/docs/v4/tutorial/introduction. [Accessed 21 08 2024]. |
| [5] | “RabbitMQ,” [Online]. Available:https://www.rabbitmq.com/tutorials. [Accessed 21 08 2024]. |