

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



ĐỒ ÁN TỐT NGHIỆP

**ĐỀ TÀI: TÌM HIỂU VÀ XÂY DỰNG ỨNG
DỤNG MẠNG XÃ HỘI TRÊN NỀN TẢNG
REACT NATIVE**

SINH VIÊN : VŨ ĐẠI AN

MÃ SINH VIÊN : 1451020002

LỚP : CNTT 14 - 05

NGƯỜI HƯỚNG DẪN: TS. ĐẬU HẢI PHONG

HÀ NỘI - 2024

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



VŨ ĐẠI AN

**TÌM HIỂU VÀ XÂY DỰNG ỨNG DỤNG
MẠNG XÃ HỘI TRÊN NỀN TẢNG REACT
NATIVE**

**CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN
MÃ SỐ : 74.80.201**

NGƯỜI HƯỚNG DẪN: TS. ĐẬU HẢI PHONG

HÀ NỘI - 2024

LỜI CAM ĐOAN

Em tên là Vũ Đại An cam đoan rằng báo cáo đồ án tốt nghiệp về đề tài “Tìm hiểu và xây dựng ứng dụng mạng xã hội trên nền tảng react native” là sản phẩm của bản thân trong quá trình thực hiện đồ án tốt nghiệp dưới sự hướng dẫn của TS. Đậu Hải Phong.

Em cam đoan rằng trong quá trình thực hiện báo cáo này, em đã tuân thủ các quy định về nghiên cứu khoa học và trích dẫn tài liệu. Mọi thông tin, số liệu và kết quả được trình bày trong báo cáo đều được thu thập và xử lý một cách trung thực và đáng tin cậy.

Em xin chịu trách nhiệm về tính chính xác và độ tin cậy của báo cáo này. Báo cáo được trình bày dưới dạng một tài liệu tham khảo cho mục đích học tập và nghiên cứu.

Hà Nội, ngày 16 tháng 06 năm 2024

Sinh viên thực hiện

Vũ Đại An

LỜI CẢM ƠN

Để hoàn thành báo cáo đồ án tốt nghiệp này, trước hết em chân thành cảm ơn các cá nhân và tổ chức đã tạo điều kiện hỗ trợ, giúp đỡ em trong suốt quá trình học tập và nghiên cứu đề tài này. Em xin cảm ơn đến các thầy cô giáo, những người đã dành thời gian và tâm huyết để hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đồ án tốt nghiệp. Nhờ sự chỉ dẫn và động viên của thầy cô, em đã có cơ hội không chỉ làm việc một cách hiệu quả mà còn phát triển kỹ năng và kiến thức của mình.

Đặc biệt, em xin gửi đến thầy Đậu Hải Phong người đã tận tình hướng dẫn, giúp đỡ để em có thể hoàn thành tốt báo cáo thực tập trong thời gian qua.

Cuối cùng, em cảm ơn trường Đại học Đại Nam đã tạo cơ hội cho sinh viên chúng em được tiếp cận và áp dụng chính kiến thức thực tế, từ đó phát triển kỹ năng và chuẩn bị cho sự nghiệp của tương lai.

Do bản thân còn những hạn chế nhất định về chuyên môn và thời gian nên sẽ khó tránh khỏi những sai sót trong quá trình thực hiện. Em mong nhận được sự góp ý của các thầy/cô trong khoa và các bạn để bài báo cáo của em được hoàn thiện tốt hơn.

Em xin chân thành cảm ơn!

LỜI NÓI ĐẦU

Trong bối cảnh công nghệ thông tin không ngừng phát triển, mạng xã hội đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của con người. Từ việc kết nối bạn bè, chia sẻ thông tin, đến việc kinh doanh trực tuyến, mạng xã hội đã và đang đóng vai trò quan trọng trong mọi khía cạnh của xã hội hiện đại. Để xây dựng một ứng dụng mạng xã hội hoàn chỉnh, việc lựa chọn công nghệ phù hợp là điều vô cùng quan trọng.

Trong đề tài này, em tập trung vào việc sử dụng React Native, NestJS và ReactJS – những công nghệ hàng đầu hiện nay trong lĩnh vực phát triển ứng dụng web và di động. React Native cho phép phát triển ứng dụng di động đa nền tảng với hiệu suất cao, trong khi đó NestJS mang lại sự mạnh mẽ và linh hoạt cho phía server, còn ReactJS giúp xây dựng giao diện người dùng phong phú và tương tác. Mục tiêu của đề tài này là nghiên cứu và xây dựng một ứng dụng mạng xã hội với các tính năng cơ bản như đăng ký, đăng nhập, tạo bài viết, bình luận, và kết nối bạn bè. Qua đó, em mong muốn đem đến cho người dùng trải nghiệm mượt mà và hiệu quả, đồng thời tạo ra một nền tảng vững chắc để mở rộng và phát triển thêm nhiều tính năng mới trong tương lai.

Em hy vọng rằng, qua đề tài này, các bạn sẽ hiểu rõ hơn về quá trình phát triển một ứng dụng mạng xã hội, cũng như cách sử dụng các công nghệ hiện đại như React Native, NestJS và ReactJS một cách hiệu quả. Bên cạnh đó, đề tài này cũng là cơ hội để em áp dụng những kiến thức đã học vào thực tiễn, nâng cao kỹ năng lập trình và giải quyết vấn đề.

[illegible]

DANH MỤC VIẾT TẮT

Từ viết tắt	Lý giải từ viết tắt
API	Application Programming Interface (Giao diện lập trình ứng dụng)
BSON	Binary JSON (Binary JavaScript Object Notation)
CDN	Cloudinary
CLI	Command Line Interface (Giao diện dòng lệnh)
CSDL	Cơ sở dữ liệu
JSON	JavaScript Object Notation (Định dạng đối tượng JavaScript)
MVC	Model-View-Controller (Mô hình - Giao diện - Bộ điều khiển)
OTT	Over-The-Top
UC	Use case
UI/UX	User Interface/ User Experience (Giao diện người dùng/ Trải nghiệm người dùng)

DANH MỤC HÌNH ẢNH

Hình 2.1. Reactjs lifecycle.....	7
Hình 2.2. Reactjs native lifecycle.....	8
Hình 2.3. Nestjs lifecycle	11
Hình 2.4. Cấu trúc database trong mongodb	13
Hình 3.1. Sơ đồ hệ thống.....	19
Hình 3.2. Biểu đồ use case tổng quát	20
Hình 3.3. Biểu đồ use case đăng nhập và đăng ký	21
Hình 3.4. Biểu đồ use case quản lý bài viết.....	21
Hình 3.5. Biểu đồ use case quản lý người dùng	22
Hình 3.6. Biểu đồ use case quản lý bài viết.....	22
Hình 3.7. Biểu đồ use case case bình luận	23
Hình 3.8. Biểu đồ use case case thích bài viết	23
Hình 3.9. Biểu đồ use case case tìm kiếm nội dung	24
Hình 3.10. Biểu đồ use case case kết bạn.....	24
Hình 3.11. Biểu đồ use case case nhắn tin	25
Hình 3.12. Biểu đồ use case case xem tin nhắn	25
Hình 3.13. Biểu đồ use case case chỉnh sửa trang cá nhân	25
Hình 3.14. Biểu đồ tuần tự đăng nhập.....	36
Hình 3.15. Biểu đồ tuần tự đăng ký.....	37
Hình 3.16. Biểu đồ tuần tự đăng xuất.....	37
Hình 3.17. Biểu đồ tuần tự xem bài viết.....	38
Hình 3.18. Biểu đồ tuần tự đăng bài viết.....	38
Hình 3.19. Biểu đồ tuần tự thích bài viết	39
Hình 3.20. Biểu đồ tuần tự bình luận bài viết	39
Hình 3.21. Biểu đồ tuần tự tìm kiếm	40

Hình 3.23. Biểu đồ tuần tự huỷ kết bạn.....	41
Hình 3.24. Biểu đồ hoạt động đăng nhập.....	41
Hình 3.25. Biểu đồ hoạt động đăng ký.....	42
Hình 3.26. Biểu đồ hoạt động đăng bài viết.....	42
Hình 3.27. Biểu đồ hoạt động xem bài viết.....	43
Hình 3.28. Biểu đồ hoạt động nhấn tin.....	43
Hình 3.29. Biểu đồ hoạt động chỉnh sửa thông tin cá nhân	44
Hình 3.30. Biểu đồ hoạt động xem thông báo.....	44
Hình 3.31. Cơ sở dữ liệu.....	45
Hình 4.1. Giao diện trang đăng nhập.....	59
Hình 4.2. Giao diện trang đăng ký	59
Hình 4.3. Giao diện trang chủ	60
Hình 4.4. Giao diện trang bạn bè.....	61
Hình 4.5. Giao diện bình luận	61
Hình 4.6. Giao diện thêm bài viết.....	62
Hình 4.7. Giao diện trang cá nhân.....	62
Hình 4.8. Giao diện trang nhấn tin	63
Hình 4.9. Giao diện trang thống kê	63
Hình 4.10. Giao diện trang quản lý người dùng.....	64
Hình 4.11. Giao diện trang quản lý bài viết	64
Hình 4.12. Giao diện màn hình đăng nhập trên mobile	65
Hình 4.13. Giao diện màn hình xem bài viết trên mobile	65
Hình 4.14. Giao diện màn hình cá nhân trên mobile.....	66
Hình 4.15. Giao diện màn hình nhấn tin trên mobile	66

DANH MỤC BẢNG BIỂU

Bảng 3.1. Danh sách các actor.....	18
Bảng 3.2. Danh sách các tác nhân	19
Bảng 3.3. Mô tả ca sử dụng use case đăng nhập	26
Bảng 3.4. Mô tả ca sử dụng use case đăng ký	27
Bảng 3.5. Mô tả ca sử dụng use case đăng xuất	27
Bảng 3.6. Mô tả ca sử dụng use case xem bài viết.....	28
Bảng 3.7. Mô tả ca sử dụng use case bình luận bài viết.....	29
Bảng 3.8. Mô tả ca sử dụng use case thích bài viết.....	30
Bảng 3.9. Mô tả ca sử dụng use case xem thông báo	31
Bảng 3.11. Mô tả ca sử dụng use case kết bạn	33
Bảng 3.12. Mô tả ca sử dụng use case huỷ kết bạn	33
Bảng 3.13. Mô tả ca sử dụng use case nhắn tin.....	34
Bảng 3.14. Mô tả ca sử dụng use case sửa thông tin cá nhân	35
Bảng 3.15. Bảng user.....	45
Bảng 3.16. Bảng post.....	46
Bảng 3.16. Bảng chat.....	47
Bảng 3.17. Bảng comment	47
Bảng 3.18. Bảng friendship	47
Bảng 4.1. Kiểm thử api đăng nhập và đăng ký	48
Bảng 4.2. Kiểm thử api đăng bài viết	49
Bảng 4.3. Kiểm thử chức năng đăng nhập và đăng ký	50
Bảng 4.4. Kiểm thử chức năng đăng bài viết	53
Bảng 4.5. Kiểm thử chức năng kết bạn	56

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI	1
1.1. Giới thiệu chung về đề tài lựa chọn	1
1.1.1. Tổng quan về mạng xã hội	1
1.1.2. Sự phát triển của mạng xã hội.....	2
1.2. Lý do chọn đề tài.....	3
1.3. Mục tiêu	3
1.4. Phạm vi nghiên cứu của đề tài	4
1.5. Phương pháp nghiên cứu	5
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG	6
2.1. Các chức năng cơ bản của mạng xã hội.....	6
2.2. Công nghệ sử dụng	6
2.2.1. Frontend	6
2.2.2. Backend (NestJS)	10
2.2.3. Database và lưu trữ.....	12
CHƯƠNG 3. XÂY DỰNG PHẦN MỀM	18
3.1. Đặc tả phần mềm	18
3.1.1. Xác định các tác nhân.....	18
3.1.2. Sơ đồ hệ thống.....	19
3.2. Phân tích thiết kế.....	19
3.2.1. Danh sách các use case.....	19
3.2.2. Biểu đồ Use Case	20
3.2.3. Mô tả các ca sử dụng.....	26
3.2.4. Biểu đồ tuần tự chức năng.....	36
3.2.5. Biểu đồ hoạt động	41
3.2.6. Cơ sở dữ liệu	45

CHƯƠNG 4. KẾT QUẢ VÀ ĐÁNH GIÁ	48
4.1. Kiểm thử	48
4.1.1. Kiểm thử API	48
4.1.2. Kiểm thử chức năng	50
4.2. Kết quả đạt được	59
4.3. Hạn chế	67
4.4. Công việc tương lai.....	68

CHƯƠNG 1.

GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

1.1. Giới thiệu chung về đề tài lựa chọn

1.1.1. Tổng quan về mạng xã hội

Mạng xã hội (social network) là hệ thống thông tin cung cấp cho cộng đồng người sử dụng mạng các dịch vụ lưu trữ, cung cấp, sử dụng, tìm kiếm, chia sẻ và trao đổi thông tin với nhau, bao gồm dịch vụ tạo trang thông tin điện tử cá nhân, trò chuyện (chat) trực tuyến, chia sẻ âm thanh, hình ảnh và các hình thức dịch vụ tương tự khác. Có thể hiểu một cách đơn giản đây là hệ thống (mạng lưới) giúp con người kết nối với những người khác. Thông qua mạng xã hội, mọi người có thể chia sẻ thông tin, hình ảnh, âm thanh... tìm kiếm bạn bè, kết nối với những người khác,...v.v

Hiện nay, tất cả mọi người đều có thể dễ dàng truy cập vào mạng xã hội bất kỳ thông qua điện thoại, máy tính, máy tính bảng,...v.v. Tuy nhiên, để sử dụng được (đăng bài, kết nối với người khác...), người dùng phải tạo một tài khoản bằng số điện thoại, email... (tùy từng loại mạng xã hội yêu cầu thế nào).

Các mạng xã hội phổ biến hiện nay:

- Facebook: đây là mạng xã hội phổ biến nhất hiện nay, với số lượng người dùng rất rất lớn, được phát hành miễn phí và có thể sử dụng được trên nhiều thiết bị như điện thoại, máy tính bảng, máy tính,... và trên nhiều nền tảng như IOS, Android,...
- Zalo: là một trong những ứng dụng cực kỳ phổ biến ở Việt Nam. Là một ứng dụng OTT, cho phép người dùng nhắn tin, gọi điện cho những người dùng khác trên nền tảng di động và máy tính có sử dụng kết nối internet.
- Instagram: Instagram là một mạng xã hội hoàn toàn khác với hai mạng xã hội trên. Đây là mạng xã hội chuyên được sử dụng để chia sẻ ảnh và tạo những tin video ngắn, lưu trữ những hình ảnh, video đẹp của mình và bạn bè. Ngoài ra, ứng dụng này còn có nhiều công cụ, hiệu ứng tạo ảnh, chỉnh sửa ảnh đẹp, được giới trẻ rất ưa chuộng. Ứng dụng này cũng tích hợp ứng dụng nhắn tin, comment trong từng ảnh, video được chia sẻ, giúp bạn bè có thể giao lưu, kết nối với nhau.

- Youtube: Mạng xã hội Youtube là một trong các sản phẩm của Google, là mạng xã hội chuyên dùng để chia sẻ các video. Tại đây, người dùng có thể đăng tải nhiều video với các dung lượng khác nhau. Thông qua Youtube, người dùng có thể tìm kiếm nhiều video ở nhiều mảng khác nhau: Phim ảnh, âm thực, trend... Hiện, Youtube là mạng xã hội phổ biến trên toàn thế giới. Đặc biệt, người dùng đăng tải video trên Youtube có thể bật kiếm tiền từ các quảng cáo trên trang Youtube và các video của Youtube.
- Tiktok: là mạng xã hội còn khá “non trẻ” so với các mạng xã hội khác, nhưng Tiktok lại có sự phát triển vô cùng mạnh mẽ bởi người dùng dễ dàng tạo tài khoản bằng số điện thoại hoặc email, mã QR hoặc bằng liên kết với các mạng xã hội khác như Facebook, Twitter, Line, Instagram,...v.v. Khi sử dụng Tiktok, người dùng sẽ dễ dàng tạo ra những video ngắn với kho nhạc miễn phí khổng lồ cùng với hiệu ứng cực đẹp và dán nhãn phong phú, đa dạng kết hợp với nhiều bộ lọc màu đẹp.

1.1.2. Sự phát triển của mạng xã hội

Mạng xã hội đang trở thành một phần không thể thiếu của cuộc sống hiện đại, và sự phát triển của chúng đang diễn ra với tốc độ nhanh chóng và đa dạng. Trong thời gian gần đây, chúng ta đã chứng kiến sự bùng nổ của video và live streaming trên các nền tảng như TikTok, Instagram và Facebook. Việc này không chỉ thúc đẩy sự sáng tạo và tương tác trực tiếp giữa người dùng mà còn mở ra một kênh mới cho doanh nghiệp và cá nhân để chia sẻ nội dung và tiếp cận đối tượng khách hàng tiềm năng.

Ngoài ra, mạng xã hội đang ngày càng chú trọng vào trải nghiệm người dùng di động, với việc phát triển các ứng dụng di động và tối ưu hóa giao diện để thu hút người dùng trên thiết bị di động. Tính năng thương mại điện tử cũng đang trở nên phổ biến hơn, khi các nền tảng mạng xã hội tích hợp các cửa hàng trực tuyến cho phép người dùng mua sắm trực tiếp từ nền tảng của họ. Trend của việc chia sẻ nội dung ngắn cũng đang dần trở nên phổ biến hơn, với sự phát triển của các tính năng như Stories trên Instagram và Facebook. Điều này phản ánh nhu cầu của người dùng trong việc tiêu thụ nội dung một cách nhanh chóng và dễ dàng trên các thiết bị di động.

Tuy nhiên, điều quan trọng mà các nền tảng mạng xã hội đang phải đối mặt là bảo vệ dữ liệu cá nhân và quyền riêng tư của người dùng. Trong bối cảnh nhiều vụ việc liên quan

đến việc sử dụng dữ liệu cá nhân một cách không đúng mục đích, các nền tảng này đang đối diện với áp lực từ phía cộng đồng và các cơ quan quản lý để tăng cường bảo vệ thông tin cá nhân của người dùng.

Mạng xã hội không chỉ là nơi chia sẻ văn bản và hình ảnh mà còn là nơi mà người dùng có thể chia sẻ âm thanh, video và nội dung đa phương tiện khác. Điều này tạo ra một môi trường phong phú và đa dạng, đồng thời tăng cường sự tương tác và kết nối giữa các thành viên trong cộng đồng mạng. Tóm lại, mạng xã hội hiện nay đang phát triển với sự tập trung vào trải nghiệm người dùng, tính năng video, thương mại điện tử, bảo vệ dữ liệu cá nhân và tương tác thời gian thực, đồng thời mở ra những cơ hội mới cho sự sáng tạo và kết nối xã hội.

1.2. Lý do chọn đề tài

Cùng với sự phát triển của xã hội, phương tiện giao tiếp giữa mọi người với nhau ngày càng đa dạng, trong đó mạng xã hội đang trở thành xu hướng. Tiềm năng của một mạng xã hội là rất lớn, áp dụng cho nhiều đối tượng, nhiều mục đích sử dụng, với sự bùng nổ của internet, mạng xã hội giờ đang không thể thiếu với đời sống mọi người.

Nắm bắt được vấn đề đó, em đã chọn đề tài “Tìm hiểu và xây dựng ứng dụng mạng xã hội trên nền tảng react native”.

1.3. Mục tiêu

Xây dựng các chức năng cơ bản: Hệ thống sẽ cung cấp một loạt các chức năng cần thiết để tạo ra một mạng xã hội hoàn chỉnh. Các chức năng này bao gồm đăng nhập, đăng ký tài khoản mới, và đăng xuất. Người dùng sẽ có thể tạo bài viết mới, chỉnh sửa hoặc xóa các bài viết của mình. Ngoài ra, hệ thống còn cho phép người dùng bình luận, thích các bài viết của nhau, tạo ra một môi trường tương tác và giao lưu sôi nổi.

Giao diện đẹp và dễ sử dụng: Một trong những mục tiêu quan trọng của dự án là thiết kế một giao diện người dùng thân thiện, hiện đại và hấp dẫn. Giao diện phải đảm bảo dễ sử dụng, trực quan để người dùng ở mọi lứa tuổi và trình độ công nghệ khác nhau đều có thể dễ dàng tiếp cận và sử dụng ứng dụng.

Nền tảng công nghệ hiện đại: Ứng dụng sẽ được xây dựng dựa trên các công nghệ tiên tiến và phổ biến hiện nay, bao gồm ReactJS cho giao diện web, React Native cho ứng

dụng di động, NestJS cho phần backend và MongoDB cho cơ sở dữ liệu. ReactJS và React Native sẽ giúp tối ưu hóa quá trình phát triển ứng dụng trên cả hai nền tảng web và di động, giúp tiết kiệm thời gian và công sức. NestJS, một framework mạnh mẽ và linh hoạt, sẽ đảm bảo rằng phần backend của ứng dụng có thể xử lý được các yêu cầu phức tạp và quy mô lớn một cách hiệu quả. MongoDB, với khả năng lưu trữ dữ liệu dạng NoSQL, sẽ cung cấp một hệ thống cơ sở dữ liệu linh hoạt, dễ mở rộng và quản lý.

Tối ưu hóa hiệu suất và bảo mật: Ngoài việc xây dựng các chức năng cơ bản, một mục tiêu quan trọng khác là tối ưu hóa hiệu suất của ứng dụng. Điều này bao gồm việc đảm bảo ứng dụng chạy mượt mà trên các thiết bị khác nhau, từ máy tính để bàn đến điện thoại di động, và có thể xử lý được số lượng lớn người dùng đồng thời. Hệ thống cũng phải được thiết kế với các biện pháp bảo mật nghiêm ngặt để bảo vệ thông tin người dùng, ngăn chặn các hành vi tấn công mạng và đảm bảo tính toàn vẹn của dữ liệu.

1.4. Phạm vi nghiên cứu của đề tài

Đối tượng: Tất cả người dùng có nhu cầu kết nối, chia sẻ thông tin và liên lạc qua mạng xã hội. Nhóm người dùng cá nhân bao gồm các cá nhân ở mọi lứa tuổi, nghề nghiệp và sở thích khác nhau. Những người dùng này sử dụng mạng xã hội để duy trì mối quan hệ cá nhân, chia sẻ những khoảnh khắc trong cuộc sống và tìm kiếm thông tin hữu ích từ cộng đồng. Nhóm người dùng chuyên nghiệp bao gồm các chuyên gia, doanh nhân và những người làm việc trong các ngành nghề khác nhau. Họ sử dụng mạng xã hội để kết nối với đồng nghiệp, mở rộng mạng lưới nghề nghiệp, chia sẻ kiến thức chuyên môn và tìm kiếm cơ hội kinh doanh. Nhóm người dùng trẻ tuổi, họ thường sử dụng các nền tảng này để giải trí, kết bạn mới, học hỏi những điều mới mẻ và tham gia vào các cộng đồng trực tuyến theo sở thích.

Phạm vi nghiên cứu:

- Trải nghiệm người dùng: nghiên cứu về trải nghiệm người dùng trong việc sử dụng mạng xã hội bao gồm khả năng tương tác, tiện ích, giao diện người dùng,...
- Phát triển ứng dụng: nghiên cứu và phát triển ứng dụng bao gồm lập kế hoạch, lập trình, kiểm thử, và triển khai
- Quản lý hệ thống và dữ liệu: nghiên cứu quản lý hệ thống, bảo mật dữ liệu người dùng, xử lý dữ liệu và quản lý dữ liệu.

1.5. Phương pháp nghiên cứu

Để thực hiện đề tài "Tìm hiểu và xây dựng ứng dụng mạng xã hội trên nền tảng React Native", em đã tiến hành một số các phương pháp nghiên cứu. Trước hết, tập trung vào việc nghiên cứu tài liệu, bao gồm các sách, bài báo khoa học và tài liệu trực tuyến liên quan đến React Native, nhằm tìm hiểu về khái niệm, lịch sử phát triển, ưu nhược điểm và các thành phần cơ bản của nền tảng này.

Đồng thời, nghiên cứu các tính năng cơ bản của mạng xã hội như đăng bài, bình luận, chia sẻ, kết bạn và nhắn tin cũng được thực hiện. Bên cạnh đó, phân tích các ứng dụng mạng xã hội phổ biến hiện nay như Facebook, Instagram, Twitter giúp hiểu rõ hơn về yêu cầu và xu hướng của người dùng.

Tiếp theo, quá trình phân tích yêu cầu được tiến hành thông qua việc khảo sát và thu thập ý kiến từ người dùng tiềm năng, nhằm xác định các tính năng quan trọng và cần thiết cho ứng dụng. Kết quả của quá trình nghiên cứu này sẽ là nền tảng để thiết kế và phát triển ứng dụng mạng xã hội trên nền tảng React Native một cách hiệu quả và đáp ứng được nhu cầu của người dùng.

CHƯƠNG 2.

CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG

2.1. Các chức năng cơ bản của mạng xã hội

Các chức năng của mạng xã hội:

- Đăng nhập: Chức năng cho phép người dùng đăng nhập vào hệ thống bằng tên người dùng và mật khẩu.
- Đăng ký: Chức năng cho phép người dùng đăng ký tài khoản mới bằng cách cung cấp các thông tin như tên, email và mật khẩu.
- Đăng bài viết: Chức năng cho phép người dùng đăng bài viết mới trên hệ thống.
- Bình luận bài viết: Chức năng cho phép người dùng bình luận về các bài viết đã được đăng trên hệ thống.
- Thích bài viết: Chức năng cho phép người dùng thể hiện sự quan tâm bằng cách thích các bài viết.
- Nhắn tin: Chức năng cho phép người dùng gửi và nhận tin nhắn từ các người dùng khác trên hệ thống.
- Kết bạn: Chức năng cho phép người dùng kết bạn với nhau trên hệ thống.
- Hủy kết bạn: Chức năng cho phép người dùng hủy kết bạn với người khác trong hệ thống.
- Quản lý người dùng: Chức năng cho phép quản trị viên quản lý người dùng trong hệ thống.
- Quản lý bài viết: Chức năng cho phép quản trị viên quản lý bài viết trong hệ thống.
- Thống kê: Chức năng cho phép quản trị viên xem được các thống kê về bài viết và người dùng.

2.2. Công nghệ sử dụng

2.2.1. Frontend

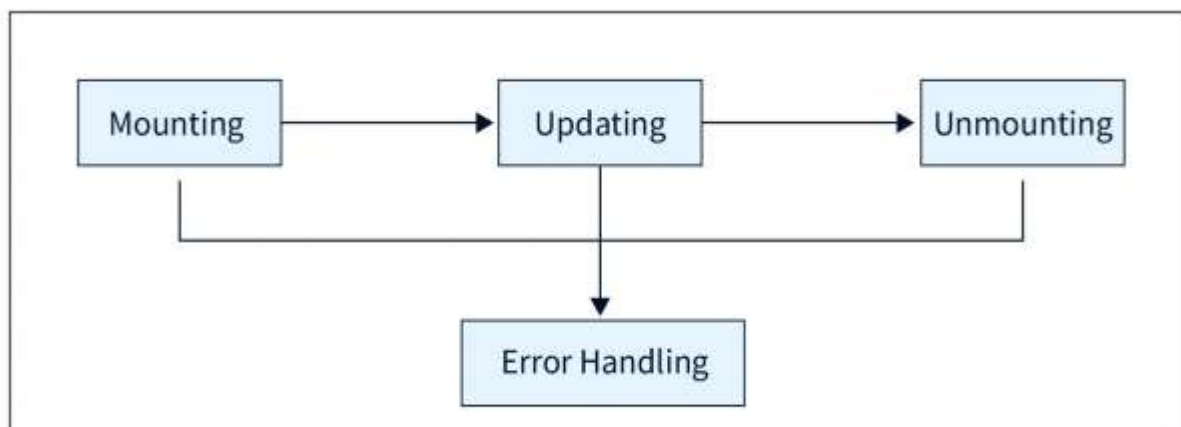
2.2.1.1. ReactJS

React là một thư viện JavaScript được sử dụng rộng rãi để xây dựng giao diện người dùng (UI) cho các ứng dụng web. Với phương pháp mới để render trang web, React mang lại tốc độ phản hồi nhanh cho người dùng khi nhập liệu.

React được phát triển bởi Facebook và được giới thiệu như một công cụ mã nguồn mở vào năm 2013. Hiện nay, React đã vượt qua các thư viện JavaScript khác như Angular và Bootstrap để trở thành thư viện phổ biến nhất cho việc xây dựng giao diện người dùng.

Với tính năng tái sử dụng cao và khả năng tích hợp với các thư viện khác, React giúp tăng tốc độ phát triển và bảo trì ứng dụng web. Ngoài ra, việc quản lý trạng thái đơn giản và hiệu quả của React cũng giúp cho các nhà phát triển front-end dễ dàng phát triển các tính năng phức tạp cho giao diện người dùng.

React hoạt động như thế nào?



Hình 2.1. Reactjs lifecycle

React hoạt động dựa trên các components. Mỗi component đại diện cho một phần của giao diện người dùng và có thể được sử dụng lại ở nhiều nơi khác nhau trong ứng dụng. Khi trạng thái của component thay đổi, React sẽ tạo ra một bản sao mới của Virtual DOM và so sánh nó với bản sao cũ để tìm ra những thay đổi cần được cập nhật. Sau đó, React chỉ cập nhật những phần thay đổi này trên DOM thật, giúp tăng tốc độ render.

Trong React, thay vì sử dụng JavaScript để thiết kế bố cục trang web, ta sử dụng JSX. JSX là một phần mở rộng của JavaScript, cho phép trích dẫn HTML và sử dụng các cú pháp thẻ HTML để render các subcomponent. JSX được đánh giá là sử dụng đơn giản hơn JavaScript và giúp tối ưu hóa code khi biên soạn, giúp ứng dụng chạy nhanh hơn so với code JavaScript tương đương. Với JSX, ta có thể viết code có cấu trúc gần giống với

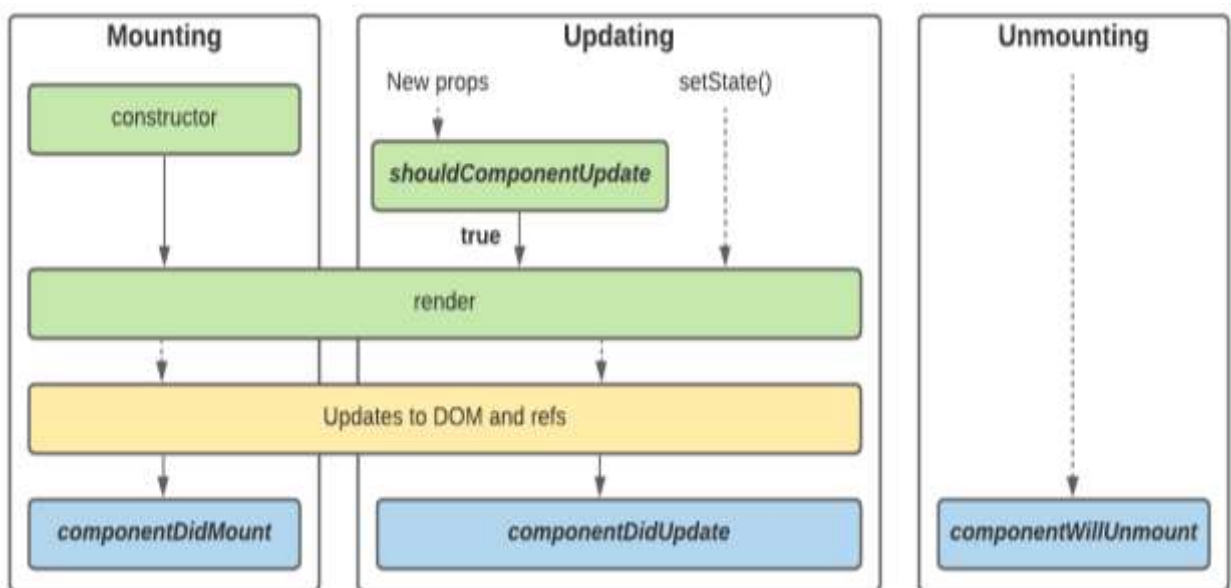
HTML, giúp cho việc đọc và hiểu code dễ dàng hơn. Ngoài ra, JSX cũng giúp tối ưu hóa việc sử dụng các component và tạo ra một cách tiếp cận trực quan hơn cho việc phát triển giao diện người dùng trong React.

Các Framework sử dụng Virtual-DOM như ReactJS cho phép thực hiện các thay đổi trên giao diện người dùng một cách hiệu quả hơn. Khi có sự thay đổi trên Virtual-DOM, các thay đổi này sẽ được phản ánh trên giao diện người dùng mà không cần phải thao tác trực tiếp với DOM trong View. Virtual-DOM đóng vai trò là Model và đồng thời đóng vai trò là View, cho phép mọi sự thay đổi trên Model đều được áp dụng trên View và ngược lại. Điều này cho phép thực hiện cơ chế Data-binding một cách dễ dàng và hiệu quả hơn, giúp tăng tốc độ ứng dụng lên đáng kể.

Đây là một lợi thế không thể tuyệt vời hơn khi sử dụng Virtual-DOM trong các Framework như ReactJS, giúp tối ưu hóa hiệu suất và cải thiện trải nghiệm người dùng trên các ứng dụng web.

2.2.1.2. React native

React Native là một framework mã nguồn mở được sáng tạo bởi Facebook. Nó được sử dụng để phát triển ứng dụng di động Android, iOS, Web và UWP bằng cách cho phép các nhà phát triển sử dụng React cùng với môi trường ứng dụng gốc. React Native được phát triển nhằm mục đích giải quyết bài toán hiệu năng của Hybrid và bài toán chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động.



Hình 2.2. Reactjs native lifecycle

Ưu điểm:

- React Native có thể viết được cả App Android và IOS, điều này sẽ giúp: Tiết kiệm thời gian và giảm chi phí phát triển của một ứng dụng. Tận dụng nguồn nhân lực tốt hơn. Duy trì ít code hơn, ít bugs hơn. Các tính năng trong cả 2 platforms cũng tương tự nhau.
- React Native cho phép các developer có thể tái sử dụng code trong khi phát triển các ứng dụng đa nền tảng. Đặc biệt, developer có thể tái sử dụng hầu như 80-90% các đoạn code thay vì phải viết và tạo các ứng dụng riêng biệt cho các nền tảng khác nhau.
- Cộng đồng người dùng lớn: React Native được đánh giá là một trong những Framework được yêu thích nhất, bạn có thể dễ dàng fix các lỗi thông qua các diễn đàn lớn nhỏ. Đồng thời có rất nhiều thư viện hỗ trợ.
- Tính ổn định và tối ưu: Được phát triển bởi Facebook, React Native có hiệu năng ổn định khá cao. Mã React Native giúp đơn giản hóa quá trình xử lý dữ liệu.
- *Nhược điểm:*
- Yêu cầu Native code. Hiệu năng kém hơn so với Native App.
- Bảo mật chưa thật sự tốt do dùng JavaScript. Do sử dụng JavaScript, người dùng cũng sẽ bị ảnh hưởng bởi những đặc điểm của JavaScript: dễ làm dễ sai, dẫn đến khó duy trì về sau.
- Quản lý bộ nhớ tùy biến chưa thật sự tốt ở một số module.
- Không thích hợp cho các app cần năng lực tính toán cao (hash, crypto, etc).

2.2.1.3. Tailwind CSS

Tailwind css là một utility-first CSS framework nó hỗ trợ phát triển xây dựng nhanh chóng giao diện người dùng, nó cũng có điểm chung giống như Bootstrap & điểm làm nó nổi bật hơn cả đó là chúng ta có thể tùy biến phát triển css theo cách mà chúng ta định nghĩa ra.

Cách cài đặt và sử dụng:

- Cài đặt thông qua sử dụng link CDN: Ở phiên bản tailwind css hiện tại là V3.0.0 thì có thể sử dụng link viết theo dạng script:

`<script src="https://cdn.tailwindcss.com" > </script>`

hoặc có thể dùng link CDN theo dạng link css ở phiên bản V2

`<link href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css" rel="stylesheet">`

- Cài tailwind css trên project (Tailwind CLI): Cài đặt tailwind css qua npm & yarn : `npm install -D tailwind css` hoặc `yarn add tailwind css`.
- Cài đặt file tailwind.config.js dùng để sử dụng tạo cấu hình tailwind css cơ bản: `npx tailwind css init`.

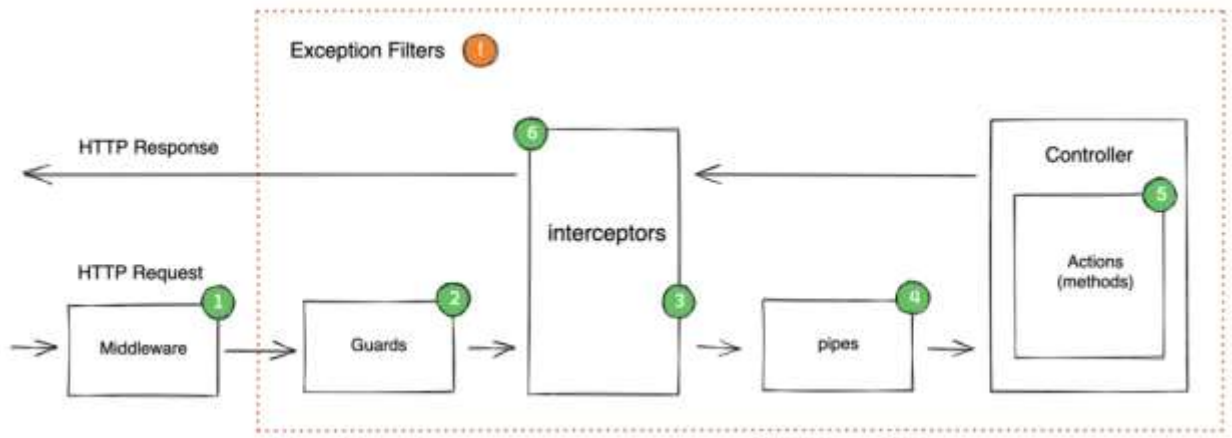
Một số framework được tailwind hỗ trợ: AngularJS, ReactJS, VueJS, NextJS, Laravel, Gatsby, Nuxt.js.

2.2.2. Backend (NestJS)

NestJS là một framework Node.js được xây dựng dựa trên nguyên tắc của Angular với cấu trúc tương tự. Bằng việc kết hợp TypeScript và các tính năng mạnh mẽ của Node.js, NestJS giúp xây dựng các ứng dụng server-side hiệu quả và dễ bảo trì. Ngoài ra, NestJS còn hỗ trợ các cơ sở dữ liệu như: MongoDB, MySQL, PostgreSQL.

Cấu trúc của NestJS:

- Controller
- Providers
- Modules
- Middleware
- Interceptors
- Testing



Hình 2.3. NestJS lifecycle

NestJS là một framework phát triển backend được nhiều người dùng sử dụng, bởi nó mang cho mình những tính năng vượt trội, hữu ích đối với người dùng. Cụ thể như sau:

- NestJS thúc đẩy sự phát triển của TypeScript.
- Mã nguồn mở của Node.js được phát triển mạnh mẽ trong thời gian gần đây.
- Sử dụng ngôn ngữ lập trình TypeScript và JavaScript Không tốn nhiều thời gian cho việc cài đặt và nghiên cứu sử dụng.
- Phát triển mạnh mẽ nhờ giao diện của dòng lệnh CLI.
- Tài liệu phong phú, đa dạng, đầy đủ.
- Khả năng mở rộng, phát triển và bảo trì cơ sở mã được thực hiện.
- Hỗ trợ các modules giúp ứng dụng được tích hợp các công nghệ và khái niệm như: Caching, WebSockets, Validation, TypeORM,...v.v

NestJS làm cho các ứng dụng của bạn có thể mở rộng, nhanh chóng, hiệu quả và dễ bảo trì. Trong ba năm qua, NestJS đã là khung hỗ trợ thúc đẩy sự phát triển vượt trội của Node.js. NestJS kết hợp phát triển front-end và middleware, điều không thể thực hiện được ở hầu hết các ngôn ngữ khác. Sử dụng TypeScript, cho phép ngôn ngữ lập trình này phản ứng nhanh với những thay đổi trong JavaScript. Tài liệu tốt, quy trình thanh toán dễ dàng. NestJS được sử dụng để xây dựng các ứng dụng cho các doanh nghiệp lớn. NestJS cung cấp cho các nhà phát triển các ứng dụng độc lập để xây dựng các ứng dụng dễ kiểm tra, cập nhật và bảo trì. NestJS cho phép bạn xây dựng các ứng dụng Rest API, MVC, Microservices, Websockets, GraphQL hoặc CRON Job.

Cấu trúc của NestJS chủ yếu dựa trên Angular, cho phép bạn tập trung vào việc thiết kế các điểm cuối thay vì cấu trúc của ứng dụng. NestJS cung cấp các mô-đun (Modules), nhà cung cấp (Providers) và bộ điều khiển (Controllers) giống như Angular giúp ứng dụng của bạn có thể mở rộng, kiểm tra và liên kết lỏng lẻo hơn, không giống như một số framework Node.js như Express và Koa.

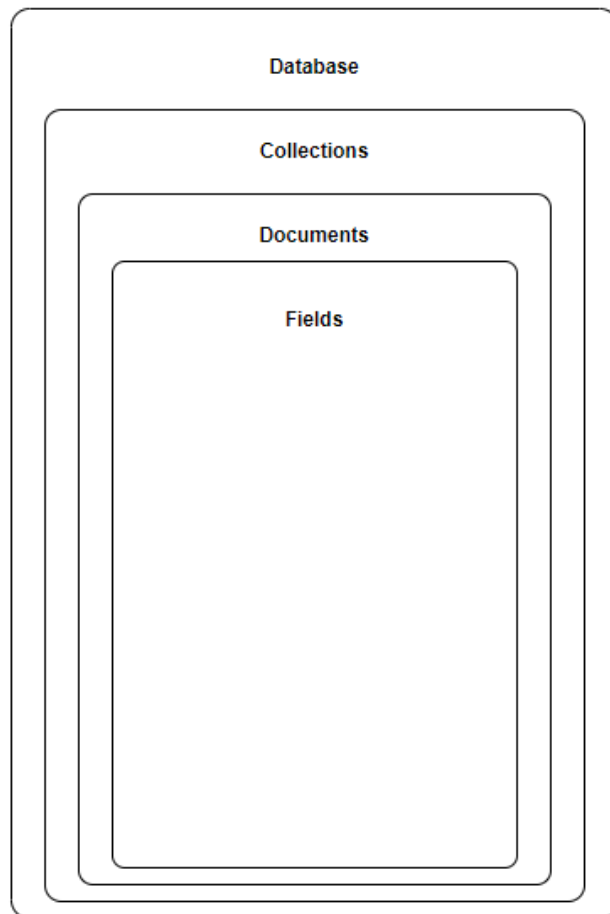
2.2.3. Database và lưu trữ

2.2.3.1. MongoDB

MongoDB là một phần mềm mã nguồn mở dùng để quản trị cơ sở dữ liệu NoSQL, được thiết kế để lưu trữ và truy vấn dữ liệu dưới dạng tài liệu (document). NoSQL (Not only SQL) được sử dụng thay thế cho cơ sở dữ liệu quan hệ (Relational Database – RDB) truyền thống. Cơ sở dữ liệu NoSQL khá hữu ích trong khi làm việc với các tập dữ liệu phân tán lớn. MongoDB là một công cụ có thể quản lý thông tin hướng document cũng như lưu trữ hoặc truy xuất thông tin.

Mỗi collection trong MongoDB chứa các document có kích thước và cấu trúc khác nhau. Sử dụng các document JSON cho lưu trữ dữ liệu giúp cho việc truy vấn cực kỳ nhanh chóng và hiệu quả. Các thuật ngữ hay sử dụng trong MongoDB. MongoDB hoạt động theo cơ chế lưu trữ và truy vấn dữ liệu dưới dạng tài liệu (document) có cấu trúc linh hoạt. Nó lưu trữ dữ liệu dưới dạng BSON (Binary JSON), một định dạng tương tự như JSON nhưng có thêm các tính năng như hỗ trợ kiểu dữ liệu nhị phân và chạy nhanh hơn.

Trong MongoDB, dữ liệu được tổ chức thành các collection, tương tự như bảng trong hệ thống quản lý cơ sở dữ liệu quan hệ. Mỗi collection chứa nhiều document, mỗi document là một tài liệu độc lập có chứa các trường và giá trị tương ứng. Các document trong cùng một collection không bắt buộc phải có cấu trúc giống nhau, cho phép lưu trữ dữ liệu linh hoạt hơn. MongoDB cung cấp giao diện lập trình (API) cho phép thao tác với dữ liệu trong MongoDB. API này hỗ trợ các tính năng như insert, update, delete và query để thao tác với các document trong collection. MongoDB cung cấp một ngôn ngữ truy vấn dựa trên JSON để cho phép truy vấn dữ liệu theo các tiêu chí nhất định. MongoDB cũng hỗ trợ các tính năng như replica set, sharding và indexing để cải thiện tính sẵn sàng và hiệu suất của hệ thống.



Hình 2.4. Cấu trúc database trong mongodb

Các thuật ngữ MongoDB thường dùng:

- **_id**: là một trường bắt buộc trong mọi document của MongoDB. **_id** được sử dụng để đại diện cho tính duy nhất của một document trong một collection. Trường **_id** hoạt động giống như khóa chính (primary key) của document.
- **Document**: là đơn vị lưu trữ dữ liệu cơ bản trong cơ sở dữ liệu MongoDB. Document mang vai trò tương tự như row trong các hệ thống cơ sở dữ liệu quan hệ truyền thống.
- **Collection**: là một tập hợp các document MongoDB. Collection tương tự như table trong hệ thống cơ sở dữ liệu quan hệ. Các collection có tính chất schema less, do đó các document trong cùng một collection có thể có các trường khác nhau.
- **Database**: là một container vật lý chứa tập hợp các collection. Một database có thể chứa 0 collection hoặc nhiều collection.

Ưu điểm của mongodb:

- Linh hoạt về cấu trúc dữ liệu: MongoDB không yêu cầu schema cố định, cho phép lưu trữ các tài liệu có cấu trúc khác nhau trong cùng một collection. Điều này giúp dễ dàng điều chỉnh và mở rộng cấu trúc dữ liệu khi cần thiết.
- Hiệu suất cao: MongoDB có khả năng truy vấn nhanh và xử lý khối lượng dữ liệu lớn một cách hiệu quả nhờ vào mô hình tài liệu và các cơ chế như indexing và sharding.
- Mở rộng ngang: Khả năng sharding của MongoDB cho phép phân phối dữ liệu trên nhiều máy chủ, giúp dễ dàng mở rộng hệ thống khi nhu cầu tăng lên mà không ảnh hưởng đến hiệu suất.
- Replication và tính khả dụng cao: Replica sets giúp tăng cường tính khả dụng của dữ liệu và đảm bảo khôi phục dữ liệu khi có sự cố xảy ra.
- Ngôn ngữ truy vấn phong phú: Hỗ trợ các truy vấn đa dạng và phức tạp, bao gồm tìm kiếm, lọc, sắp xếp và tổng hợp dữ liệu.
- Hỗ trợ dữ liệu không gian địa lý: MongoDB có khả năng lưu trữ và truy vấn dữ liệu không gian địa lý, rất hữu ích cho các ứng dụng liên quan đến bản đồ và vị trí.
- Khả năng tích hợp cao: Dễ dàng tích hợp với các công cụ và nền tảng khác, bao gồm các hệ thống big data, công cụ phân tích, và các ngôn ngữ lập trình phổ biến.

Nhược điểm của mongodb:

- Thiếu hỗ trợ giao dịch ACID toàn diện: Trước phiên bản 4.0, MongoDB không hỗ trợ giao dịch ACID đa tài liệu. Mặc dù phiên bản mới đã cải thiện điều này, nhưng khả năng vẫn còn hạn chế so với các hệ thống RDBMS truyền thống.
- Tiêu tốn bộ nhớ cao: MongoDB yêu cầu bộ nhớ lớn hơn do việc lưu trữ dữ liệu dưới dạng BSON và sử dụng các index.
- Vấn đề với dữ liệu phức tạp: Mặc dù MongoDB rất linh hoạt với cấu trúc tài liệu, nhưng khi dữ liệu trở nên phức tạp và có nhiều mối quan hệ, việc quản lý và truy vấn có thể trở nên khó khăn hơn.

- Công cụ quản lý hạn chế: So với các hệ quản trị cơ sở dữ liệu quan hệ, các công cụ quản lý và giám sát cho MongoDB có thể chưa hoàn thiện và đa dạng bằng.
- Bảo mật: Mặc dù MongoDB đã cải thiện nhiều về bảo mật, nhưng vẫn cần phải cấu hình cẩn thận để tránh các lỗ hổng bảo mật, đặc biệt khi triển khai trên môi trường mạng mở.
- Chỉ số và hiệu suất truy vấn: Cần phải thiết kế cẩn thận hệ thống chỉ số (index) để đảm bảo hiệu suất truy vấn tốt. Việc sử dụng không đúng có thể dẫn đến giảm hiệu suất và tăng chi phí lưu trữ.

2.2.3.2. Cloudinary

Cloudinary là một cloud-based service, nó cung cấp một giải pháp quản lý hình ảnh bao gồm upload, lưu trữ, thao tác, tối ưu hóa và delivery. Với cloudinary bạn có thể dễ dàng upload ảnh lên cloud, tự động thực thi các thao tác với ảnh một cách thông minh mà không cần phải cài đặt bất kỳ một phần mềm phức tạp nào khác. Cloudinary cung cấp các APIs toàn diện và màn hình quản lý giúp chúng ta dễ dàng tích hợp vào các trang web và ứng dụng di động.

Cloudinary cung cấp nền tảng trải nghiệm đa phương tiện cấp độ doanh nghiệp cho tất cả các loại dữ liệu. Bằng cách sử dụng AI, tự động hóa và khả năng xử lý hình ảnh và video nâng cao, Cloudinary loại bỏ rắc rối của các yêu cầu phương tiện kỹ thuật số thủ công và cung cấp một lộ trình rõ ràng để các công ty tạo ra kho hình ảnh ấn tượng trên quy mô lớn.

Cloudinary cung cấp một Ruby Gem để dễ dàng cho việc tương tác với các app viết bởi ngôn ngữ Ruby với một số framework như là Rails hay Sinatra với các tương tác chính sau:

- Xây dựng các URL để chuyển đổi và thao tác với hình ảnh.
- Rails view helper cho việc nhúng và thay đổi hình ảnh.
- API wrappers: upload image, quản lý và nhiều thứ khác.
- Upload image trực tiếp từ trình duyệt sử dụng một jQuery plugin.
- Tích hợp với Active Record.

- CarrierWave plugin.
- Hình ảnh tĩnh đồng bộ với CDN delivery.
- Migration tool.

Ưu điểm của Cloudinary:

- Quản lý và xử lý hình ảnh mạnh mẽ: Cloudinary cung cấp nhiều công cụ và API mạnh mẽ để quản lý, chỉnh sửa, và tối ưu hóa hình ảnh và video, bao gồm cắt, xoay, thay đổi kích thước, và áp dụng các bộ lọc.
- CDN tích hợp: Dịch vụ của Cloudinary được tích hợp với mạng phân phối nội dung (CDN), giúp tải hình ảnh và video nhanh hơn cho người dùng toàn cầu.
- Tối ưu hóa hiệu suất: Cloudinary tự động tối ưu hóa hình ảnh và video để giảm kích thước tệp mà không làm giảm chất lượng, giúp cải thiện hiệu suất trang web và trải nghiệm người dùng.
- Hỗ trợ nhiều định dạng: Cloudinary hỗ trợ nhiều định dạng hình ảnh và video khác nhau, bao gồm cả các định dạng mới như WebP và AVIF, giúp cải thiện hiệu suất và chất lượng.
- Tích hợp dễ dàng: Cloudinary cung cấp nhiều SDK và plugin cho các ngôn ngữ lập trình và nền tảng phổ biến như JavaScript, Python, Ruby, PHP, và các hệ thống quản lý nội dung như WordPress, giúp việc tích hợp trở nên dễ dàng.
- Tính năng bảo mật: Cloudinary cung cấp các tính năng bảo mật như kiểm soát truy cập, xác thực, và các tùy chọn bảo mật cao cấp để bảo vệ tài nguyên.
- Hỗ trợ lưu trữ và sao lưu: Dịch vụ lưu trữ đám mây đảm bảo an toàn và sao lưu dữ liệu tự động, giảm thiểu nguy cơ mất mát dữ liệu.

Nhược điểm của Cloudinary:

- Chi phí: Cloudinary có thể trở nên đắt đỏ khi sử dụng với quy mô lớn hoặc khi yêu cầu nhiều tính năng cao cấp. Chi phí tăng lên dựa trên dung lượng lưu trữ, băng thông sử dụng, và số lượng thao tác xử lý hình ảnh/video.

- Phụ thuộc vào bên thứ ba: Việc phụ thuộc vào dịch vụ của bên thứ ba có thể gây ra rủi ro về mặt kiểm soát dữ liệu và tính khả dụng. Nếu Cloudinary gặp sự cố, dịch vụ của bạn cũng có thể bị ảnh hưởng.
- Độ phức tạp của API: Mặc dù API của Cloudinary rất mạnh mẽ, nhưng cũng có thể phức tạp đối với người mới bắt đầu hoặc những người không có nhiều kinh nghiệm về lập trình.
- Quản lý tài nguyên lớn: Khi số lượng hình ảnh và video tăng lên, việc quản lý và tổ chức các tài nguyên có thể trở nên phức tạp và tốn thời gian.
- Phụ thuộc vào kết nối internet: Do là dịch vụ đám mây, Cloudinary yêu cầu kết nối internet ổn định. Nếu kết nối internet kém, hiệu suất truy cập và quản lý tài nguyên có thể bị ảnh hưởng.

CHƯƠNG 3. XÂY DỰNG PHẦN MỀM

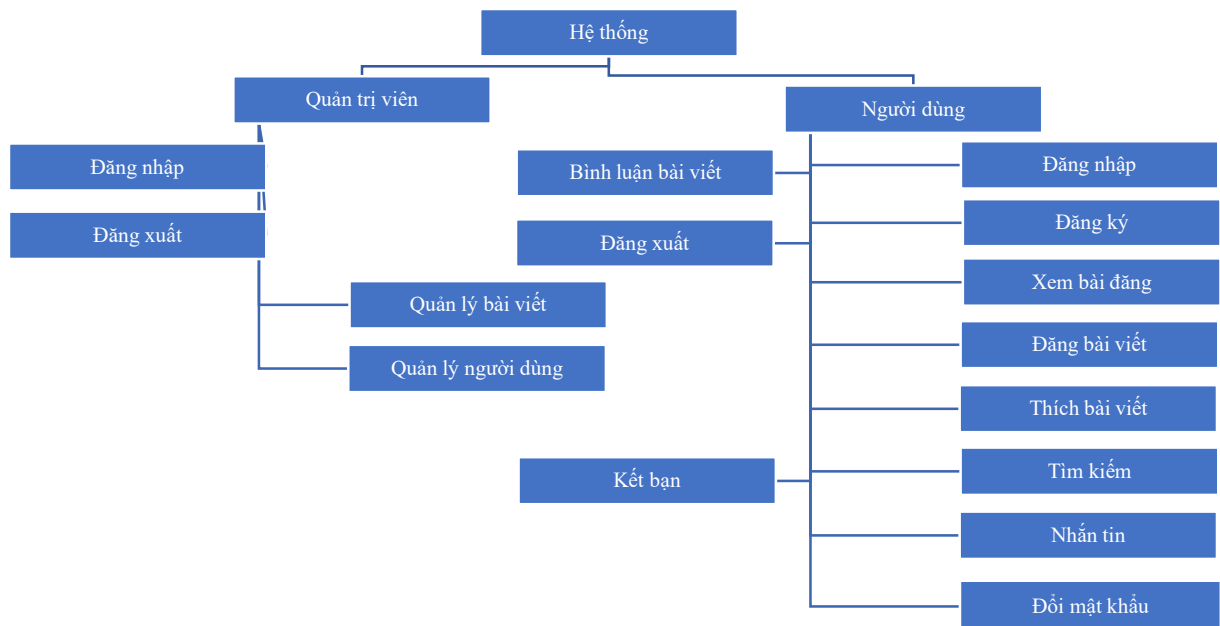
3.1. Đặc tả phần mềm

3.1.1. Xác định các tác nhân

Bảng 3.1. Danh sách các actor

STT	Actor	Mô tả Actor	Chức năng của Actor
1	Người dùng	Là người dùng hệ thống, sở hữu tài khoản để đăng nhập vào hệ thống.	Đăng nhập Đăng ký Đăng xuất Xem các bài viết Đăng các bài viết Bình luận các bài viết Thích các bài viết Nhắn tin Kết bạn Chỉnh sửa thông tin cá nhân Tìm kiếm
2	Quản trị viên	Là người dùng hệ thống cao cấp nhưng cũng phải có tài khoản để đăng nhập	Đăng nhập Thống kê Quản lý bài viết Quản lý người dùng

3.1.2. Sơ đồ hệ thống



Hình 3.1. Sơ đồ hệ thống

3.2. Phân tích thiết kế

3.2.1. Danh sách các use case

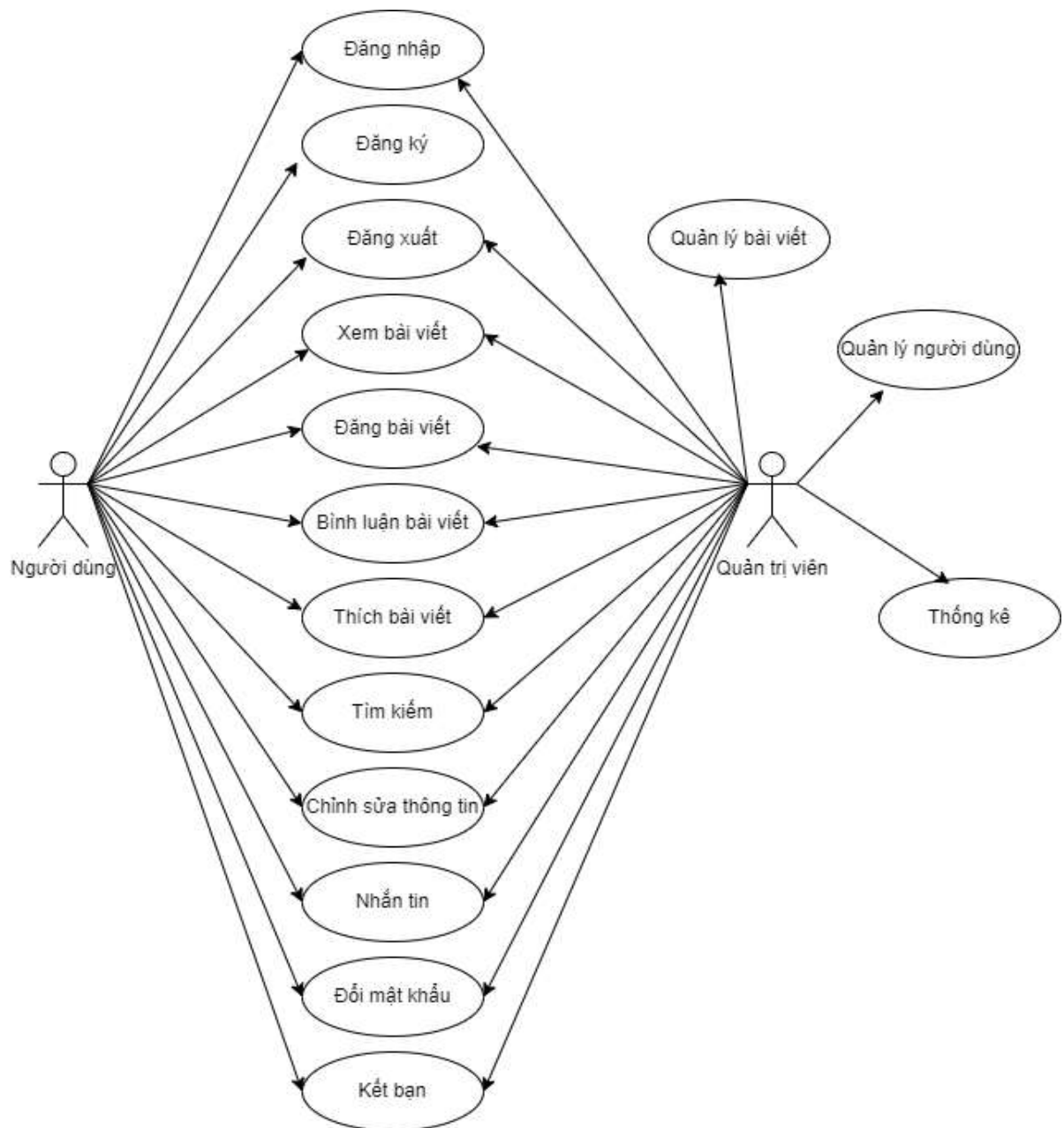
Bảng 3.2. Danh sách các tác nhân

STT	Actor	Tên use case
1	Quản trị viên	Đăng nhập
		Đăng xuất
		Quản lý bài viết
		Quản lý người dùng
2	Người dùng	Đăng nhập
		Đăng xuất
		Đăng ký
		Xem bài đăng
		Đăng bài viết

STT	Actor	Tên use case
		Bình luận bài viết
		Like bài viết
		Tìm kiếm
		Nhắn tin
		Đổi mật khẩu

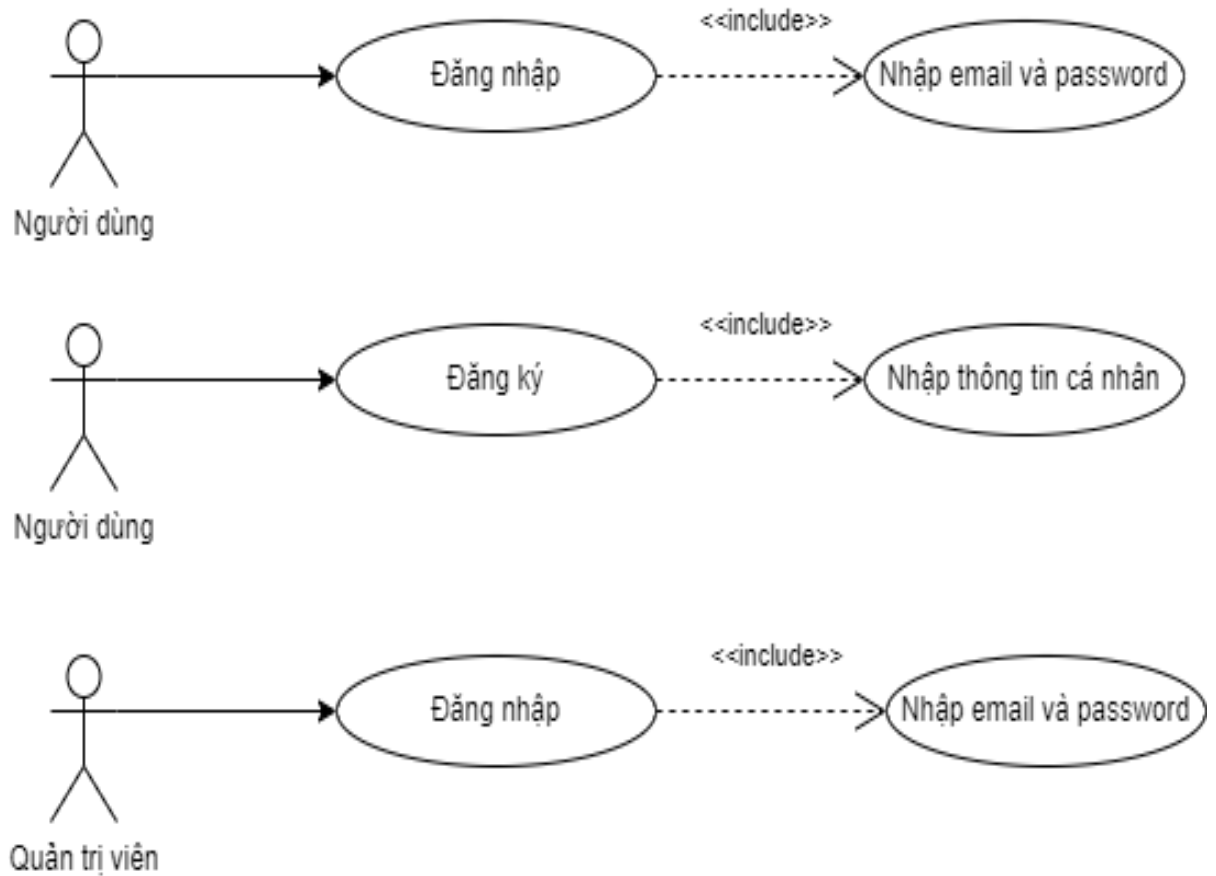
3.2.2. Biểu đồ Use Case

- Biểu đồ use case tổng quát



Hình 3.2. Biểu đồ use case tổng quát

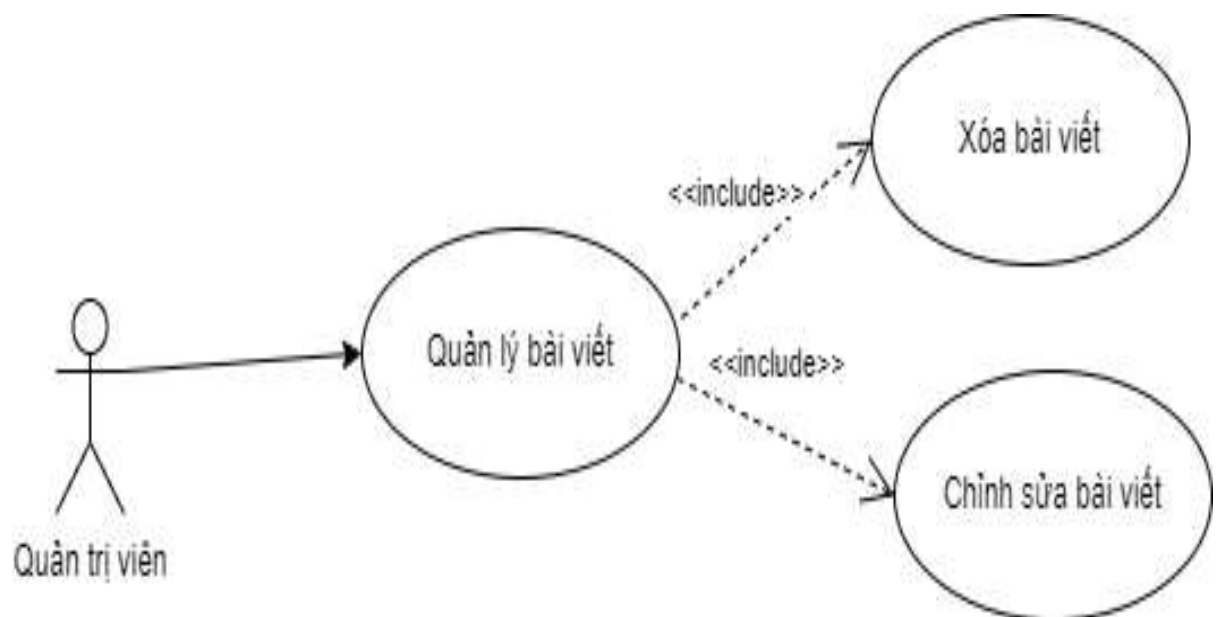
- Biểu đồ use case đăng nhập và đăng ký



Hình 3.3. Biểu đồ use case đăng nhập và đăng ký

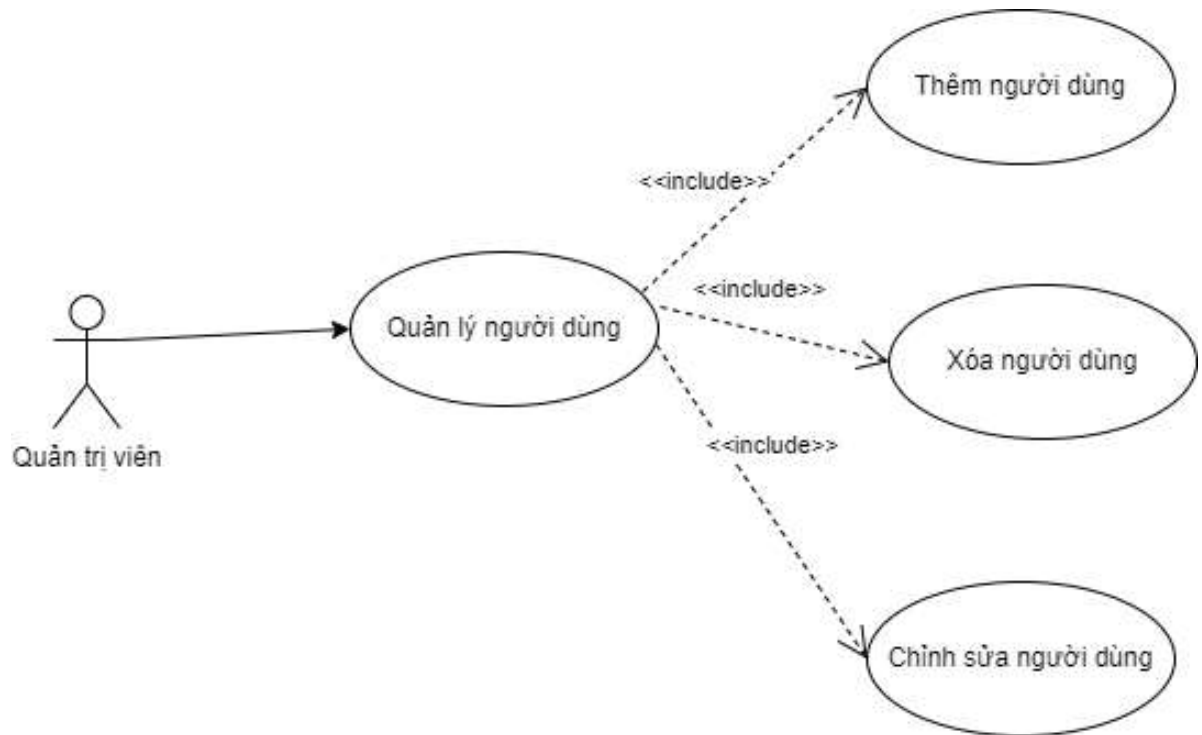
Use case cho quản trị viên

- Quản lý bài viết



Hình 3.4. Biểu đồ use case quản lý bài viết

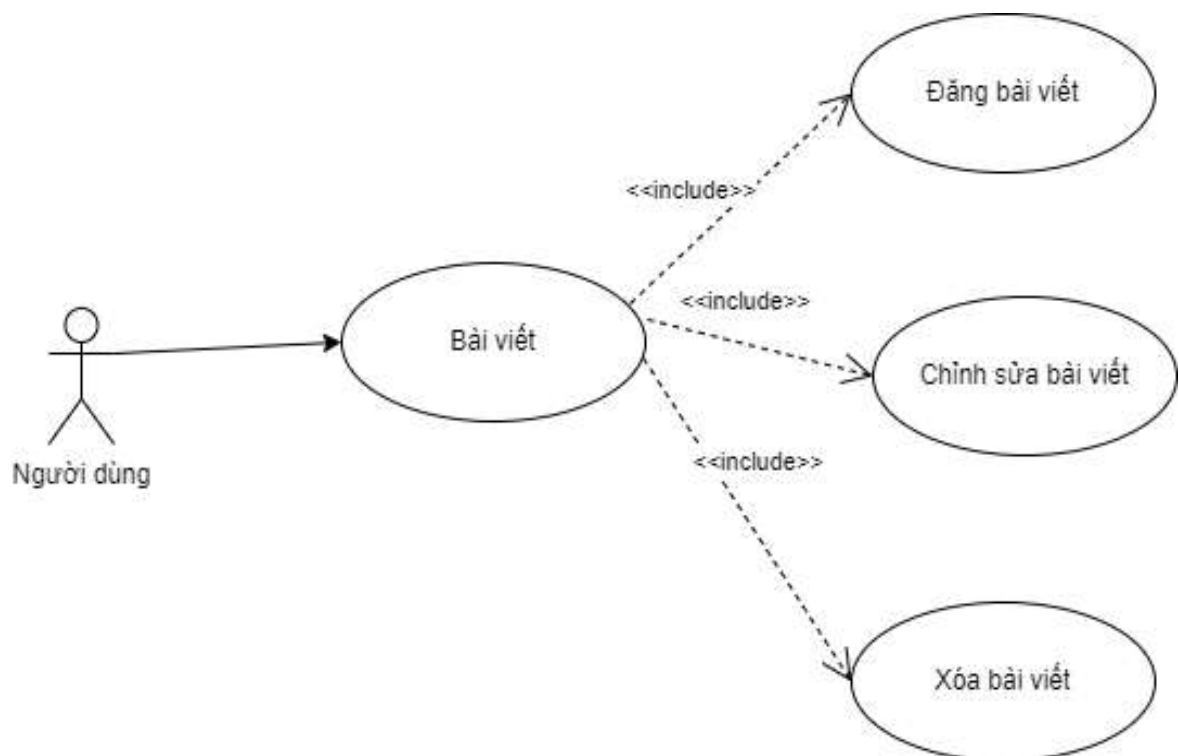
- Quản lý người dùng



Hình 3.5. Biểu đồ use case quản lý người dùng

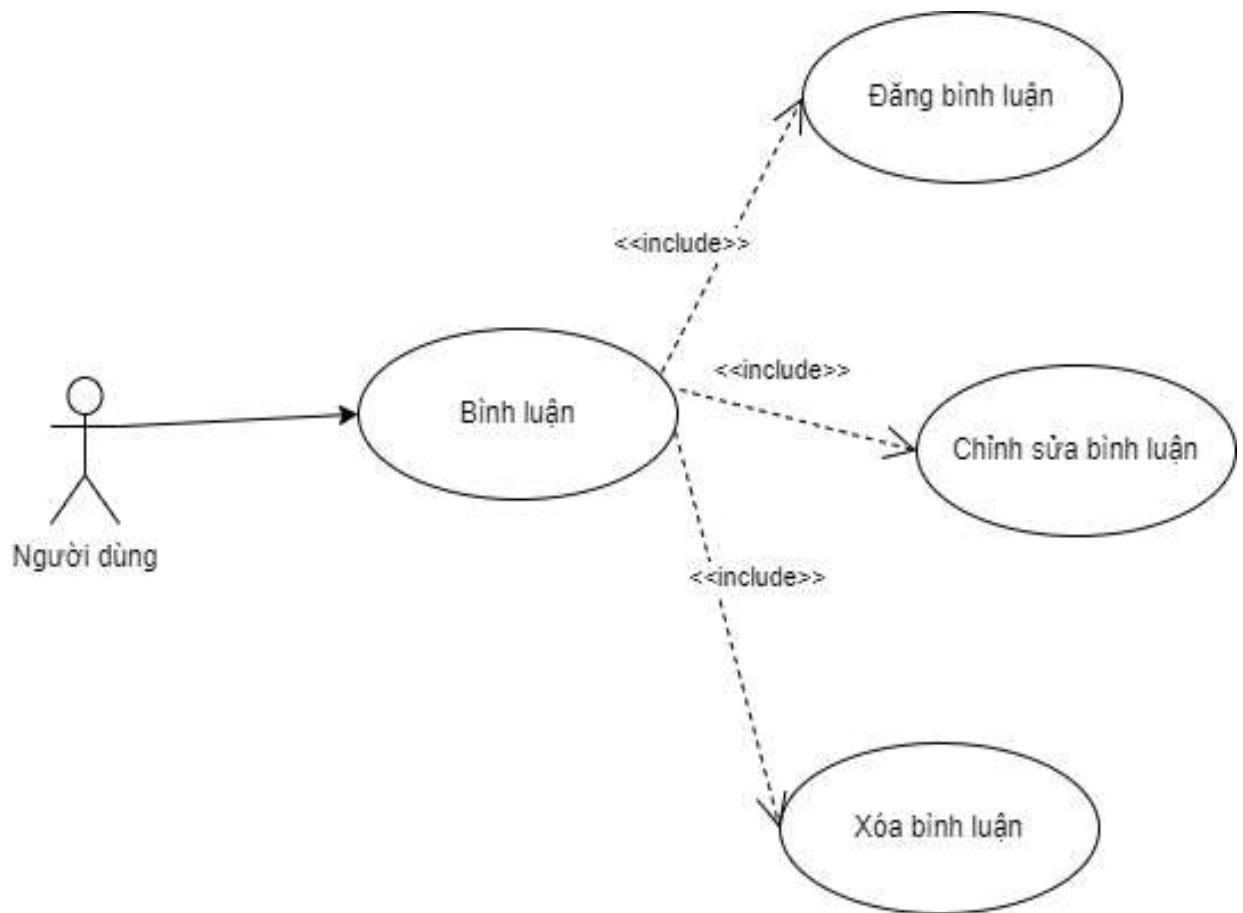
Use case cho người dùng

- Biểu đồ use case bài viết



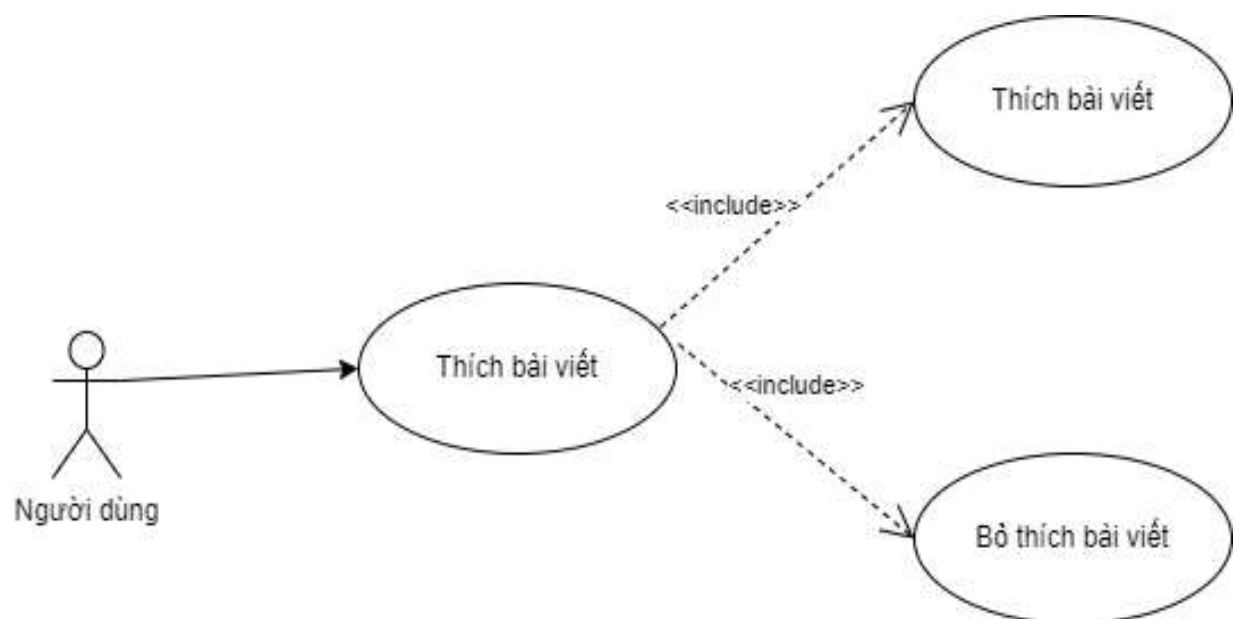
Hình 3.6. Biểu đồ use case quản lý bài viết

- Biểu đồ use case bình luận bài viết



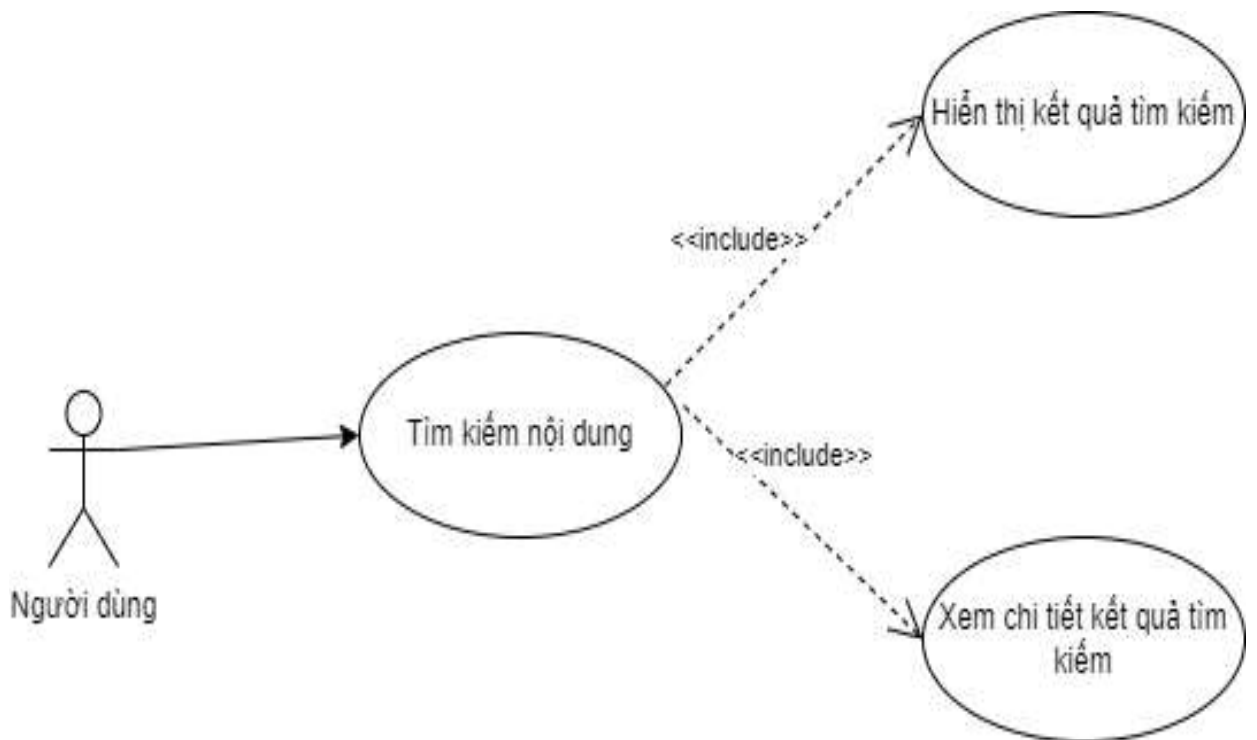
Hình 3.7. Biểu đồ use case case bình luận

- Biểu đồ use case thích bài viết



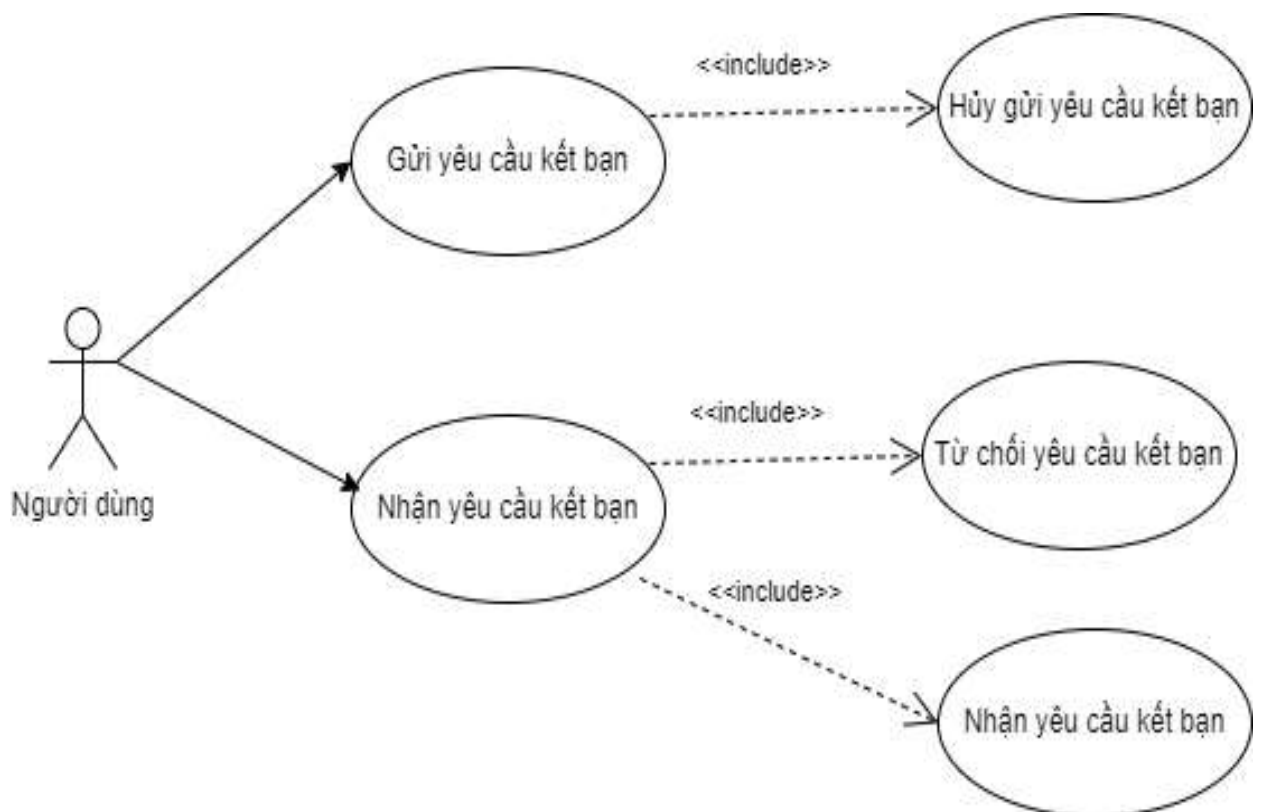
Hình 3.8. Biểu đồ use case case thích bài viết

- Biểu đồ use case tìm kiếm



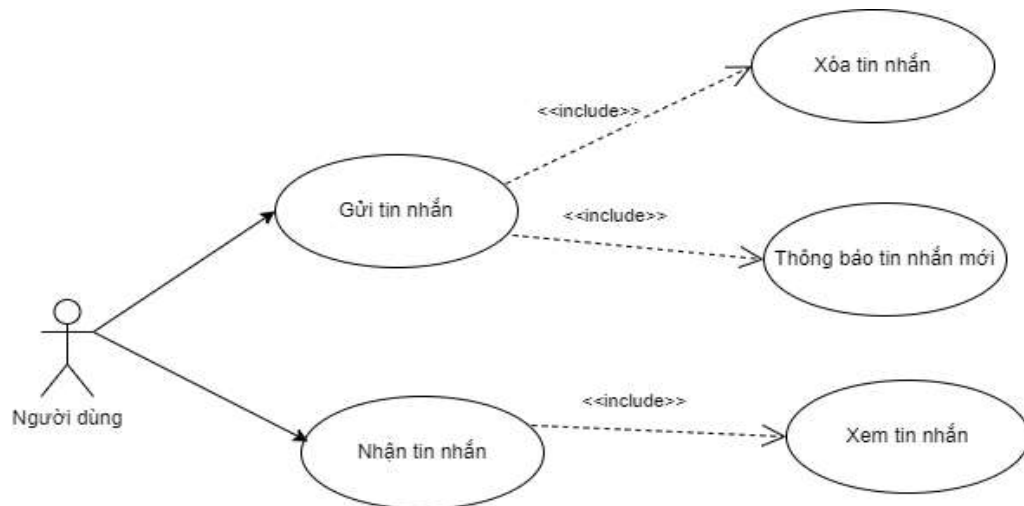
Hình 3.9. Biểu đồ use case case tìm kiếm nội dung

- Biểu đồ use case kết bạn



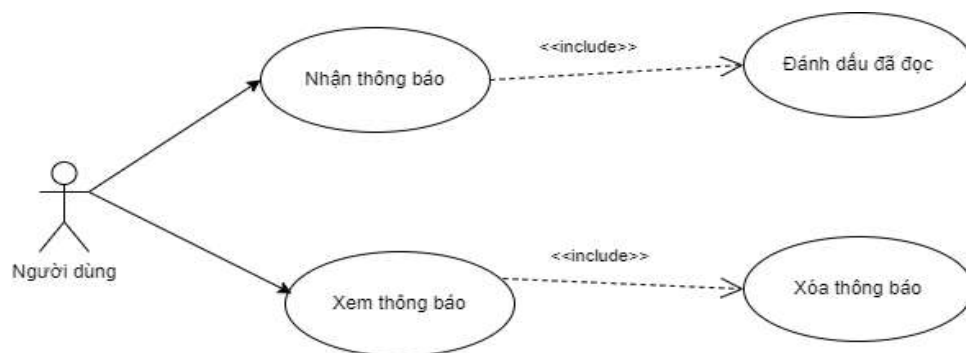
Hình 3.10. Biểu đồ use case case kết bạn

- Biểu đồ use case nhắn tin



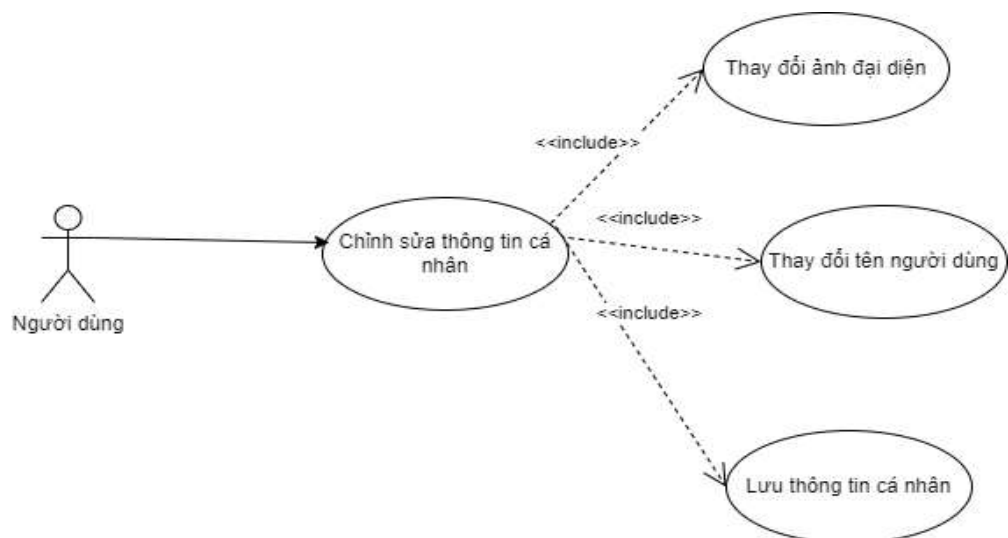
Hình 3.11. Biểu đồ use case case nhắn tin

- Use case xem thông báo



Hình 3.12. Biểu đồ use case case xem tin nhắn

- Use case chỉnh sửa trang cá nhân



Hình 3.13. Biểu đồ use case case chỉnh sửa trang cá nhân

3.2.3. Mô tả các ca sử dụng

- Use case đăng nhập

Bảng 3.3. Mô tả ca sử dụng use case đăng nhập

STT	Thành phần	Nội dung
1	Tóm tắt	Use case mô tả cách người dùng đăng nhập vào hệ thống.
2	Actor	Người dùng và quản trị viên
3	Dòng sự kiện	<p>Dòng sự kiện chính: Use case này bắt đầu khi người dùng cuối muốn đăng nhập vào ứng dụng</p> <ul style="list-style-type: none">- Người dùng nhập tên đăng nhập và mật khẩu- Hệ thống kiểm tra thông tin đăng nhập, nếu tài khoản và mật khẩu chính xác thì chuyển qua màn hình chính. <p>Dòng sự kiện khác: Nếu trong dòng sự kiện chính, người dùng nhập, mật khẩu thì hệ thống sẽ hiển thị thông báo lỗi. Người dùng có thể chọn trở về đầu dòng sự kiện chính hoặc hủy bỏ việc đăng nhập, lúc này use case kết thúc.</p>
4	Các yêu cầu đặc biệt	Không có.
5	Trạng thái hệ thống trước khi thực hiện use	Đứng ở màn hình đăng nhập, các chức năng không được hiển thị.
6	Trạng thái hệ thống sau khi thực hiện use	Nếu use case thành công, người dùng có thể sử dụng các chức năng của website.
7	Điểm mở rộng	Không có.

- Use case xem đăng ký

Bảng 3.4. Mô tả ca sử dụng use case đăng ký

STT	Thành phần	Nội dung
1	Tóm tắt	Use case mô tả người dùng đăng ký tài khoản hệ thống.
2	Actor	Người dùng.
3	Dòng sự kiện	<p>Dòng sự kiện chính: Use case này bắt đầu khi người dùng cuối muốn đăng ký một tài khoản trên hệ thống.</p> <ul style="list-style-type: none"> - Người dùng nhấn tin đăng ký, sau đó nhập những thông tin cần thiết. - Hệ thống kiểm tra thông tin đăng ký, thông tin hợp lệ thì lưu vào CSDL và chuyển người dùng về trang đăng nhập. <p>Dòng sự kiện khác: Nếu trong dòng sự kiện chính, người dùng nhập dữ liệu không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi. Người dùng có thể chọn trở về đầu dòng sự kiện chính hoặc hủy bỏ việc đăng nhập, lúc này use case kết thúc.</p>
4	Các yêu cầu đặc biệt	Không có.
5	Trạng thái hệ thống trước khi thực hiện use	Đứng ở màn hình đăng nhập, các chức năng không được hiển thị.
6	Trạng thái hệ thống sau khi thực hiện use	Nếu use case thành công, người dùng có thể đăng nhập và sử dụng các chức năng của website.
7	Điểm mở rộng	Không có.

- Use case xem đăng xuất

Bảng 3.5. Mô tả ca sử dụng use case đăng xuất

STT	Thành phần	Nội dung
1	Tóm tắt	Use case mô tả người dùng đăng ký tài khoản hệ thống.

STT	Thành phần	Nội dung
2	Actor	Người dùng và quản trị viên
3	Dòng sự kiện	<p>Dòng sự kiện chính: Use case này bắt đầu khi người dùng cuối muốn đăng ký một tài khoản trên hệ thống.</p> <ul style="list-style-type: none"> - Người dùng nhấn tin đăng ký, sau đó nhập những thông tin cần thiết. - Hệ thống kiểm tra thông tin đăng ký, thông tin hợp lệ thì lưu vào CSDL và chuyển người dùng về trang đăng nhập. <p>Dòng sự kiện khác: Nếu trong dòng sự kiện chính, người dùng nhập dữ liệu không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi. Người dùng có thể chọn trở về đầu dòng sự kiện chính hoặc hủy bỏ việc đăng nhập, lúc này use case kết thúc.</p>
4	Các yêu cầu đặc biệt	Không có.
5	Trạng thái hệ thống trước khi thực hiện use	Đứng ở màn hình đăng nhập, các chức năng không được hiển thị.
6	Trạng thái hệ thống sau khi thực hiện use	Nếu use case thành công, người dùng có thể đăng nhập và sử dụng các chức năng của hệ thống.
7	Điểm mở rộng	Không có.

- Use case xem bài viết

Bảng 3.6. Mô tả ca sử dụng use case xem bài viết

STT	Thành phần	Nội dung
1	Tóm tắt	Use case mô tả cách người dùng xem bài viết trên bản tin
2	Actor	Người dùng

STT	Thành phần	Nội dung
3	Dòng sự kiện	<p>Dòng sự kiện chính: Use case này bắt đầu khi người dùng muốn xem các bài viết trên bản tin.</p> <ul style="list-style-type: none"> - Người dùng đăng nhập vào website. - Các bài viết sẽ được hiển thị tại trang chủ của website. <p>Dòng sự kiện khác: Không.</p>
4	Các yêu cầu đặc biệt	Không có.
5	Trạng thái hệ thống trước khi thực hiện use	Đứng ở màn hình đăng nhập, các chức năng chưa được hiển thị.
6	Trạng thái hệ thống sau khi thực hiện use	Nếu use case thành công, người dùng sẽ được chuyển về trang chủ.
7	Điểm mở rộng	Không có.

- Use case bình luận bài viết

Bảng 3.7. Mô tả ca sử dụng use case bình luận bài viết

STT	Thành phần	Nội dung
1	Tóm tắt	Use case mô tả cách người dùng bình luận một bài viết trong hệ thống
2	Actor	Người dùng
3	Dòng sự kiện	<p>Dòng sự kiện chính: Use case này bắt đầu khi người dùng muốn bình luận một bài viết nào đó.</p> <ul style="list-style-type: none"> - Tại màn hình chính, người dùng tìm đến bài viết muốn bình luận và nhập nội dung bình luận rồi nhấn gửi. - Hệ thống sẽ xử lý và lưu trữ dữ liệu vào CSDL, bình luận sẽ được hiển thị bên dưới bài viết.

STT	Thành phần	Nội dung
		Dòng sự kiện khác: Không.
4	Các yêu cầu đặc biệt	Không có.
5	Trạng thái hệ thống trước khi thực hiện use	Đứng ở màn hình chính, các chức năng được hiển thị.
6	Trạng thái hệ thống sau khi thực hiện use	Nếu use case thành công, dữ liệu bình luận của bài viết sẽ được cập nhật.
7	Điểm mở rộng	Không có

- Use case thích bài viết

Bảng 3.8. Mô tả ca sử dụng use case thích bài viết

STT	Thành phần	Nội dung
1	Tóm tắt	Use case mô tả cách người dùng thích một bài viết trong hệ thống
2	Actor	Người dùng
3	Dòng sự kiện	<p>Dòng sự kiện chính: Use case này bắt đầu khi người dùng muốn thích một bài viết nào đó.</p> <ul style="list-style-type: none"> - Tại màn hình chính, người dùng tìm đến bài viết muốn thích và nhấn vào biểu tượng like. - Biểu tượng like sẽ chuyển sang màu xanh dương và số người thích cũng sẽ cập nhật. <p>Dòng sự kiện khác: Không.</p>
4	Các yêu cầu đặc biệt	Không có.
5	Trạng thái hệ thống trước khi thực hiện use	Đứng ở màn hình chính, các chức năng được hiển thị.

STT	Thành phần	Nội dung
6	Trạng thái hệ thống sau khi thực hiện use	Nếu use case thành công, dữ liệu người thích của bài viết sẽ được cập nhật.
7	Điểm mở rộng	Không có.

- Use case xem thông báo

Bảng 3.9. Mô tả ca sử dụng use case xem thông báo

STT	Thành phần	Nội dung
1	Tóm tắt	Use case mô tả cách người dùng xem tất cả thông báo.
2	Actor	Người dùng.
3	Dòng sự kiện	<p>Dòng sự kiện chính: Use case này bắt đầu khi người dùng muốn xem tất cả thông báo.</p> <ul style="list-style-type: none"> - Tại màn hình chính, người dùng ấn vào biểu tượng thông báo, chọn xem tất cả thông báo. - Hệ thống xử lý và tải dữ liệu lên. - Người dùng sẽ được chuyển đến trang xem tất cả thông báo. <p>Dòng sự kiện khác: Không.</p>
4	Các yêu cầu đặc biệt	Không có.
5	Trạng thái hệ thống trước khi thực hiện use	Đứng ở màn hình chính, các chức năng được hiển thị.
6	Trạng thái hệ thống sau khi thực hiện use	Nếu use case thành công, trang xem tất cả thông báo sẽ được hiển thị.
7	Điểm mở rộng	Không có.

- Use case tìm kiếm

Bảng 3.10. Mô tả ca sử dụng use case tìm kiếm

STT	Thành phần	Nội dung
1	Tóm tắt	Use case mô tả cách người dùng tìm kiếm 1 bài viết hoặc 1 tài khoản khác trong hệ thống.
2	Actor	Người dùng.
3	Dòng sự kiện	<p>Dòng sự kiện chính: Use case này bắt đầu khi người dùng muốn tìm kiếm 1 bài viết hoặc 1 tài khoản khác trong hệ thống.</p> <ul style="list-style-type: none"> - Tại màn hình chính, người dùng ấn vào biểu tượng tìm kiếm, ô tìm kiếm sẽ hiện ra, sau đó nhập tiêu đề bài viết hoặc tên tài khoản muốn tìm rồi nhấn Enter. - Hệ thống xử lý dữ liệu và trả về. - Người dùng sẽ được chuyển đến trang hiển thị kết quả tìm kiếm. <p>Dòng sự kiện khác: Nếu tên tài khoản hoặc không có bài viết nào phù hợp với keyword tìm kiếm thì thông báo không tìm thấy và trở về trang chủ.</p>
4	Các yêu cầu đặc biệt	Không có.
5	Trạng thái hệ thống trước khi thực hiện use	Đứng ở màn hình chính, các chức năng được hiển thị.
6	Trạng thái hệ thống sau khi thực hiện use	Nếu use case thành công, người dùng sẽ được chuyển đến trang kết quả tìm kiếm.
7	Điểm mở rộng	Không có

- Use case kết bạn

Bảng 3.11. Mô tả ca sử dụng use case kết bạn

STT	Thành phần	Nội dung
1	Tóm tắt	Use case mô tả cách người dùng kết bạn với một tài khoản khác.
2	Actor	Người dùng.
3	Dòng sự kiện	<p>Dòng sự kiện chính: Use case này bắt đầu khi người dùng muốn kết bạn với một tài khoản khác.</p> <ul style="list-style-type: none"> - Tại màn hình chính, người dùng tìm kiếm tài khoản muốn kết bạn và nhấn vào nút kết bạn nếu chưa kết bạn tại trang cá nhân của tài khoản đó. - Hệ thống sẽ xử lý và lưu trữ dữ liệu vào danh sách kết bạn của người dùng đó, nút kết bạn chuyển thành nút hủy kết bạn. <p>Dòng sự kiện khác: Không.</p>
4	Các yêu cầu đặc biệt	Không có.
5	Trạng thái hệ thống trước khi thực hiện use	Đứng ở màn hình chính, các chức năng được hiển thị.
6	Trạng thái hệ thống sau khi thực hiện use	Nếu use case thành công, dữ liệu người thích của bài viết sẽ được cập nhật.
7	Điểm mở rộng	Không có.

- Use case hủy kết bạn

Bảng 3.12. Mô tả ca sử dụng use case hủy kết bạn

STT	Thành phần	Nội dung
1	Tóm tắt	Use case mô tả cách người dùng hủy kết bạn với một tài khoản khác.
2	Actor	Người dùng

STT	Thành phần	Nội dung
3	Dòng sự kiện	<p>Dòng sự kiện chính: Use case này bắt đầu khi người dùng muốn hủy kết bạn với một tài khoản khác.</p> <ul style="list-style-type: none"> - Tại màn trang cá nhân, người dùng vào tab bạn bè, tại đây người dùng tìm kiếm tài khoản muốn hủy kết bạn và nhấn Hủy kết bạn. - Hệ thống xử lý và lưu dữ liệu vào hệ thống - Danh sách bạn bè của người dùng sẽ được cập nhập lại, nút hủy kết bạn sẽ chuyển thành kết bạn <p>Dòng sự kiện khác: Không.</p>
4	Các yêu cầu đặc biệt	Không có
5	Trạng thái hệ thống trước khi thực hiện use	Đứng ở trang cá nhân, các chức năng được hiển thị.
6	Trạng thái hệ thống sau khi thực hiện use	Nếu use case thành công, danh sách kết bạn của người dùng sẽ được cập nhật.
7	Điểm mở rộng	Không có.

- Use case nhắn tin

Bảng 3.13. Mô tả ca sử dụng use case nhắn tin

STT	Thành phần	Nội dung
1	Tóm tắt	Use case mô tả cách người dùng nhắn tin với một tài khoản khác.
2	Actor	Người dùng.
3	Dòng sự kiện	<p>Dòng sự kiện chính: Use case này bắt đầu khi người dùng muốn nhắn tin với một tài khoản khác.</p> <ul style="list-style-type: none"> - Tại màn hình chính, người dùng tìm kiếm tài khoản muốn nhắn tin hoặc vào mục nhắn tin

STT	Thành phần	Nội dung
		<p>để chọn. Trang tin nhắn sẽ hiện ra, người dùng nhập tin nhắn và nhấn gửi.</p> <ul style="list-style-type: none"> - Hệ thống xử lý và lưu dữ liệu vào hệ thống. - Tin nhắn gửi đi sẽ được hiển thị trong cửa sổ nhắn tin. <p>Dòng sự kiện khác: Không</p>
4	Các yêu cầu đặc biệt	Không có.
5	Trạng thái hệ thống trước khi thực hiện use	Đứng ở màn hình chính, các chức năng được hiển thị.
6	Trạng thái hệ thống sau khi thực hiện use	Nếu use case thành công, tin nhắn mới sẽ cập nhật và người dùng gửi tin nhắn thành công.
7	Điểm mở rộng	Không có

- Use case sửa thông tin cá nhân

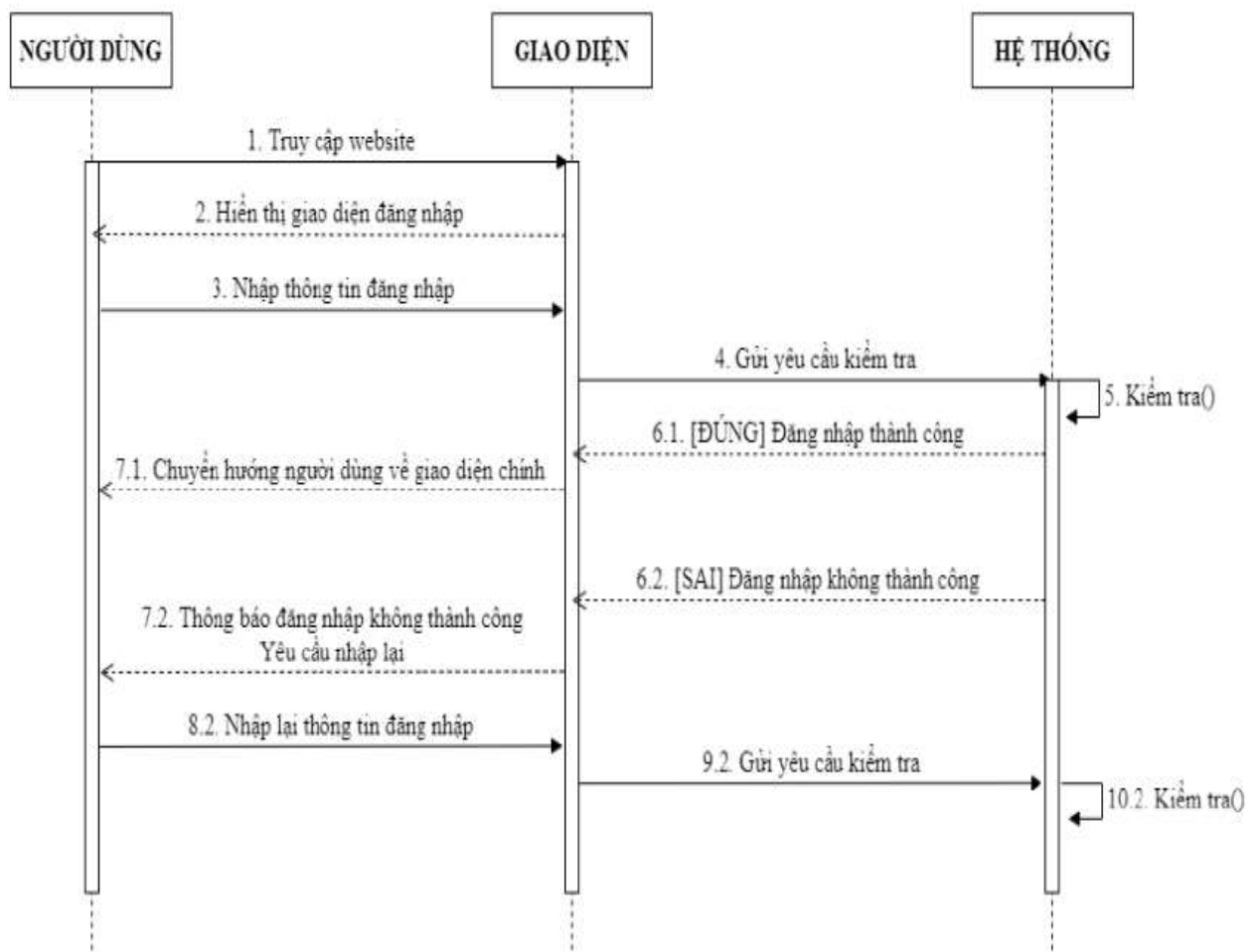
Bảng 3.14. Mô tả ca sử dụng use case sửa thông tin cá nhân

STT	Thành phần	Nội dung
1	Tóm tắt	Use case mô tả cách người dùng chỉnh sửa thông tin cá nhân
2	Actor	Người dùng
3	Dòng sự kiện	<p>Dòng sự kiện chính: Use case này bắt đầu khi người dùng muốn chỉnh sửa thông tin cá nhân của mình</p> <ul style="list-style-type: none"> - Tại trang cá nhân, người dùng nhấn vào nút chỉnh sửa trang cá nhân. Trang chỉnh sửa trang cá nhân sẽ hiển thị, người dùng nhập các thông tin cần chỉnh sửa và nhấn cập nhật - Hệ thống xử lý và lưu vào CSDL - Thành công, người dùng sẽ được chuyển đến trang cá nhân với dữ liệu đã được chỉnh sửa

STT	Thành phần	Nội dung
		Dòng sự kiện khác: Nếu dữ liệu người dùng nhập vào không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại.
4	Các yêu cầu đặc biệt	Không có
5	Trạng thái hệ thống trước khi thực hiện use	Đứng ở trang cá nhân, các chức năng được hiển thị.
6	Trạng thái hệ thống sau khi thực hiện use	Nếu use case thành công, người dùng sẽ được chuyển về trang cá nhân
7	Điểm mở rộng	Không có

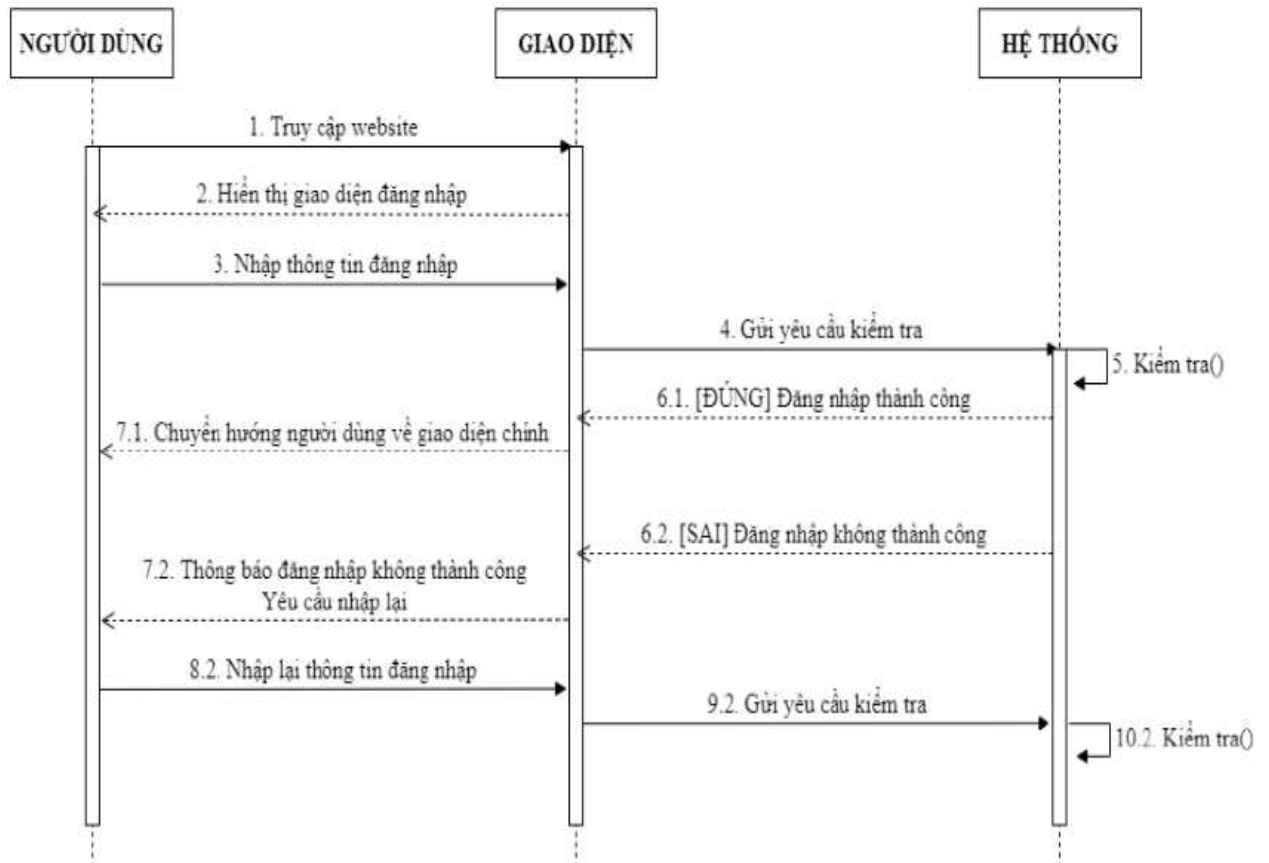
3.2.4. Biểu đồ tuần tự chức năng

- Biểu đồ tuần tự đăng nhập



Hình 3.14. Biểu đồ tuần tự đăng nhập

- Biểu đồ tuần tự đăng ký



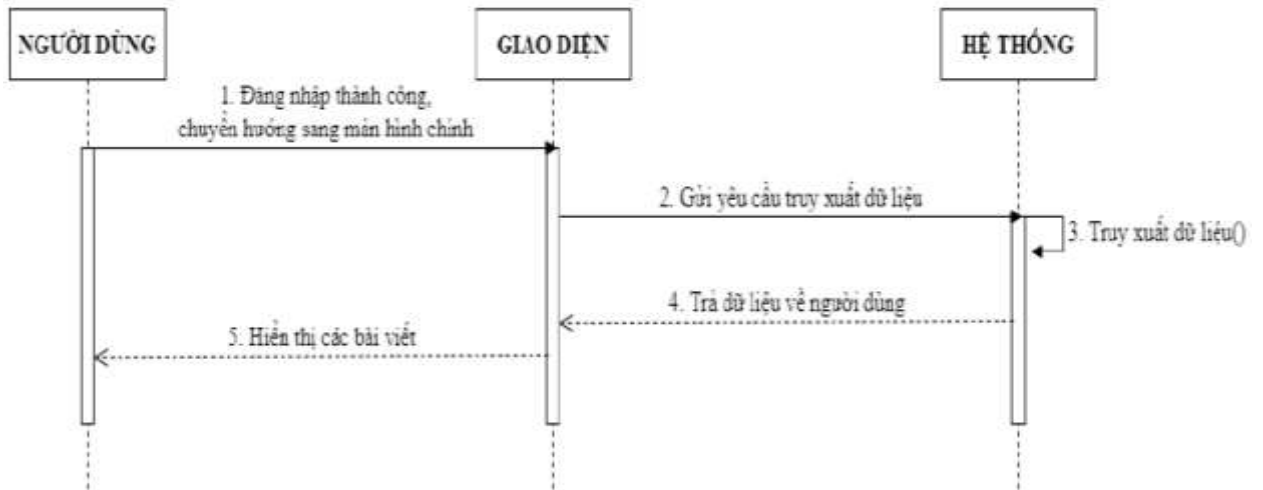
Hình 3.15. Biểu đồ tuần tự đăng ký

- Biểu đồ tuần tự đăng xuất



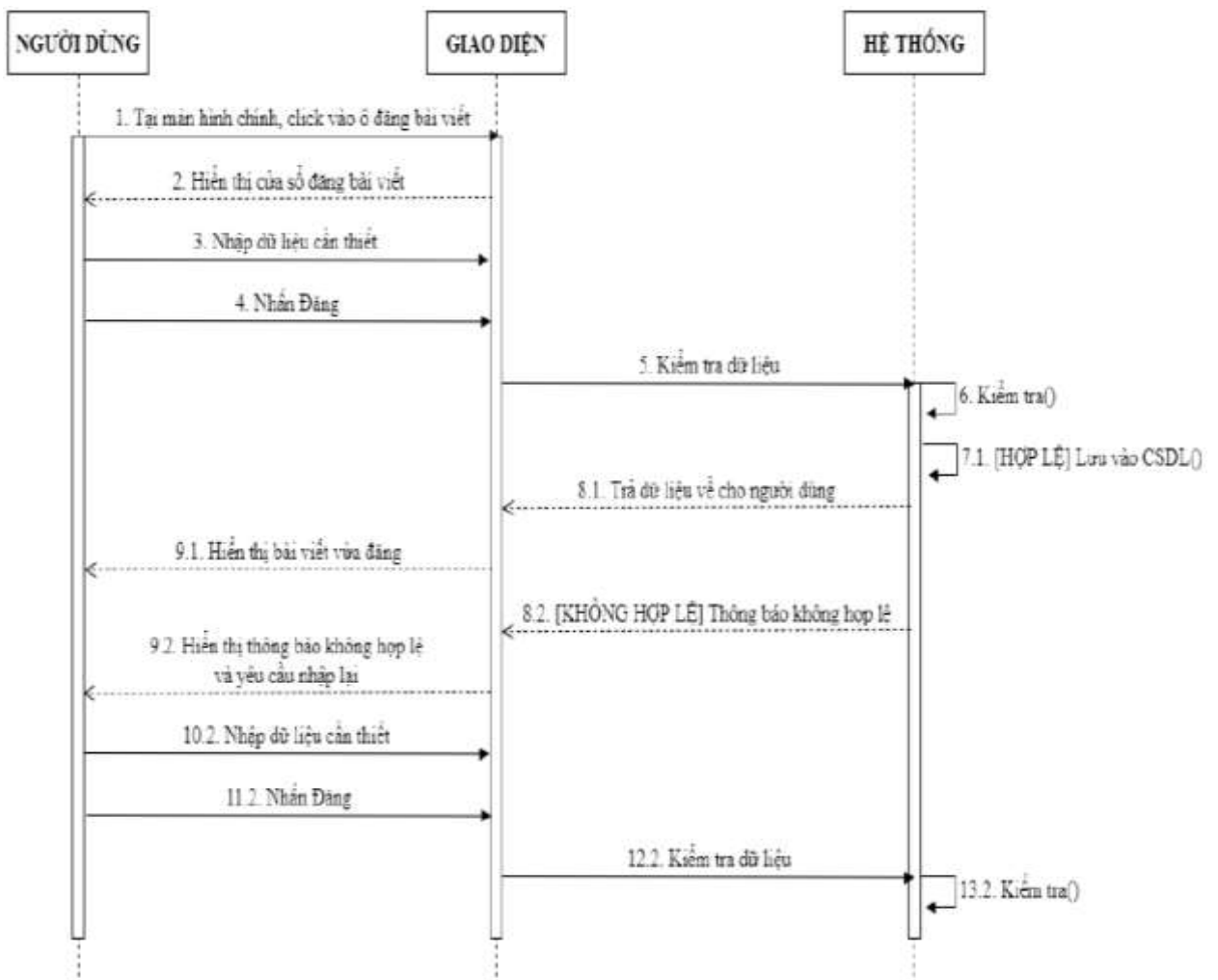
Hình 3.16. Biểu đồ tuần tự đăng xuất

- Biểu đồ tuần tự xem bài viết



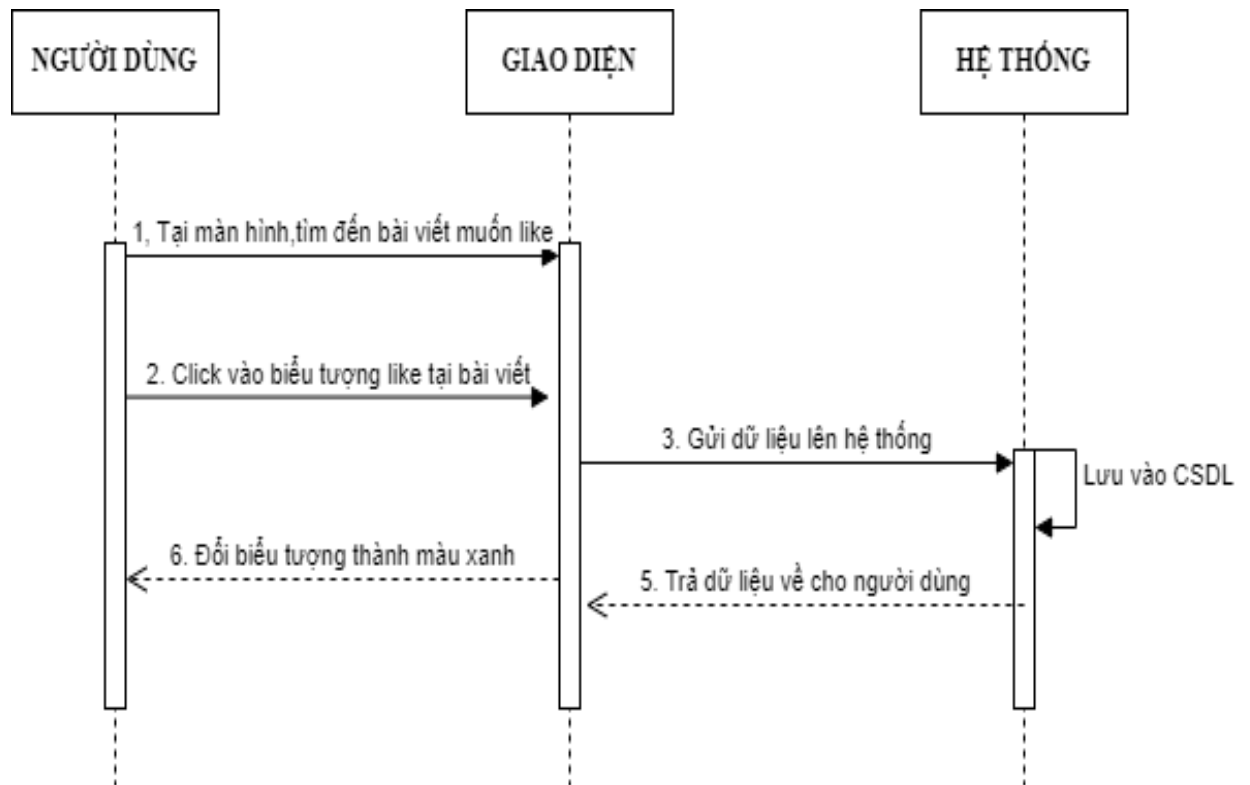
Hình 3.17. Biểu đồ tuần tự xem bài viết

- Biểu đồ tuần tự đăng bài viết



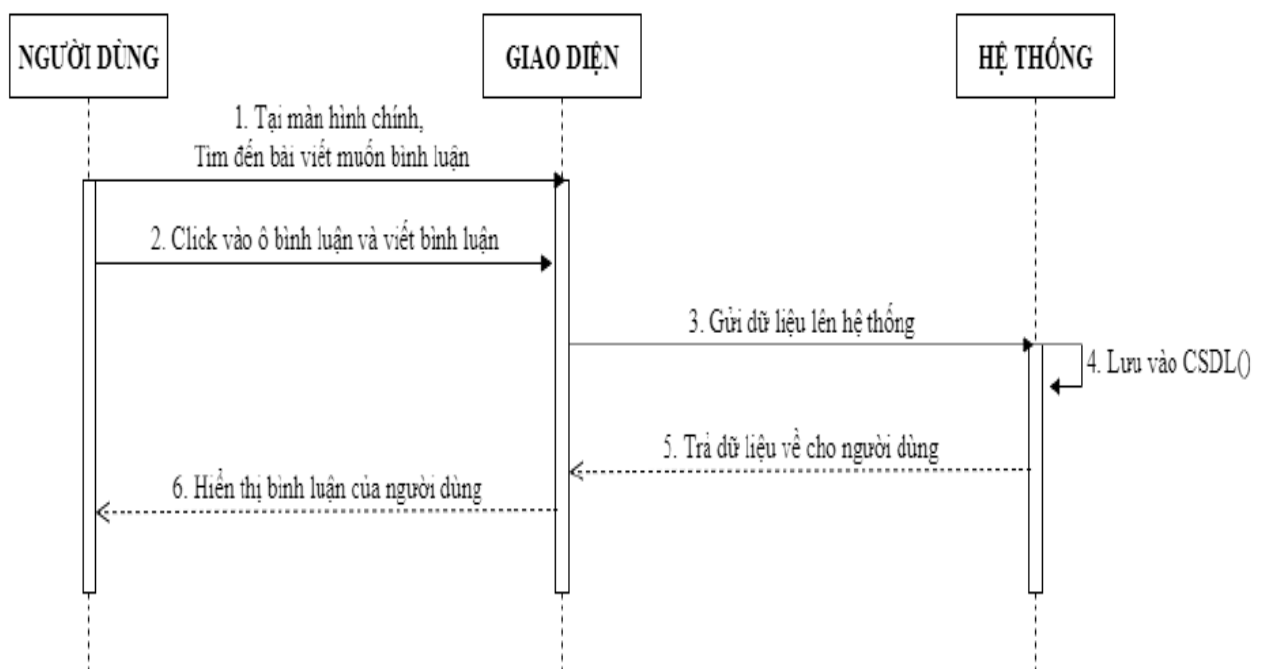
Hình 3.18. Biểu đồ tuần tự đăng bài viết

- Biểu đồ tuần tự thích bài viết



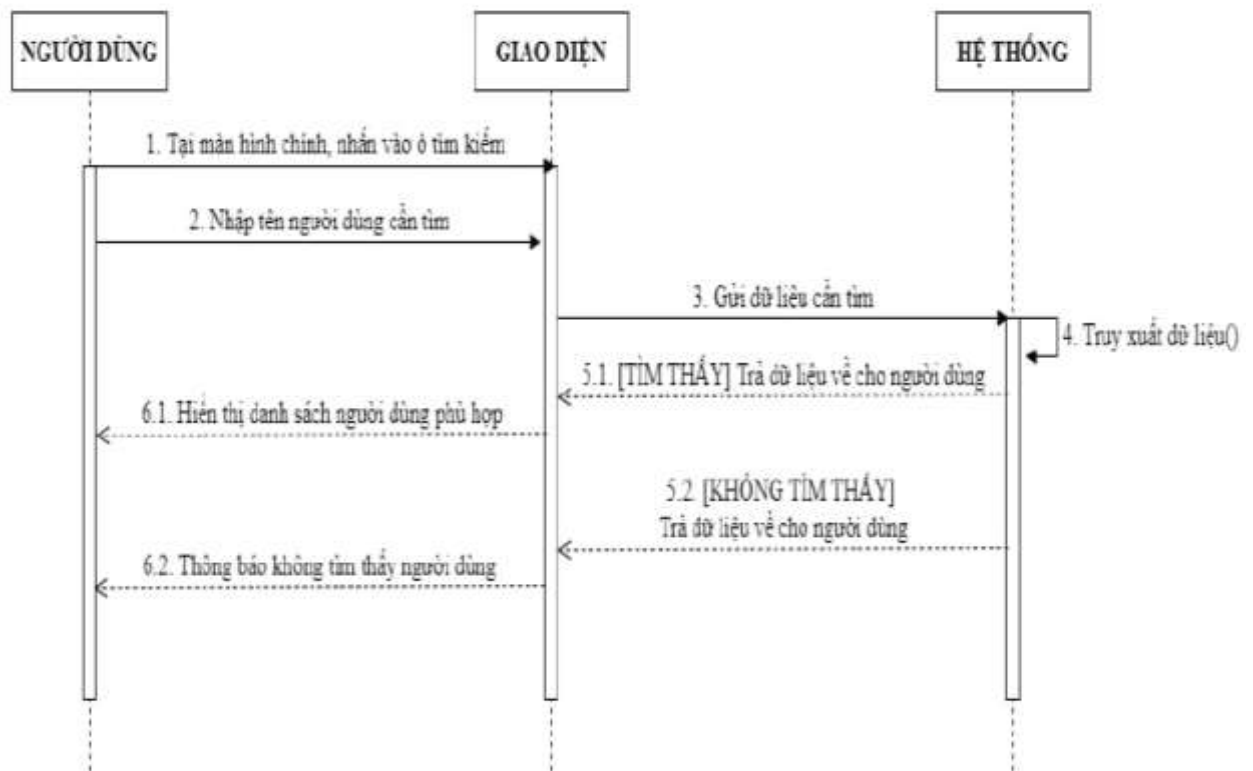
Hình 3.19. Biểu đồ tuần tự thích bài viết

- Biểu đồ tuần tự bình luận bài viết



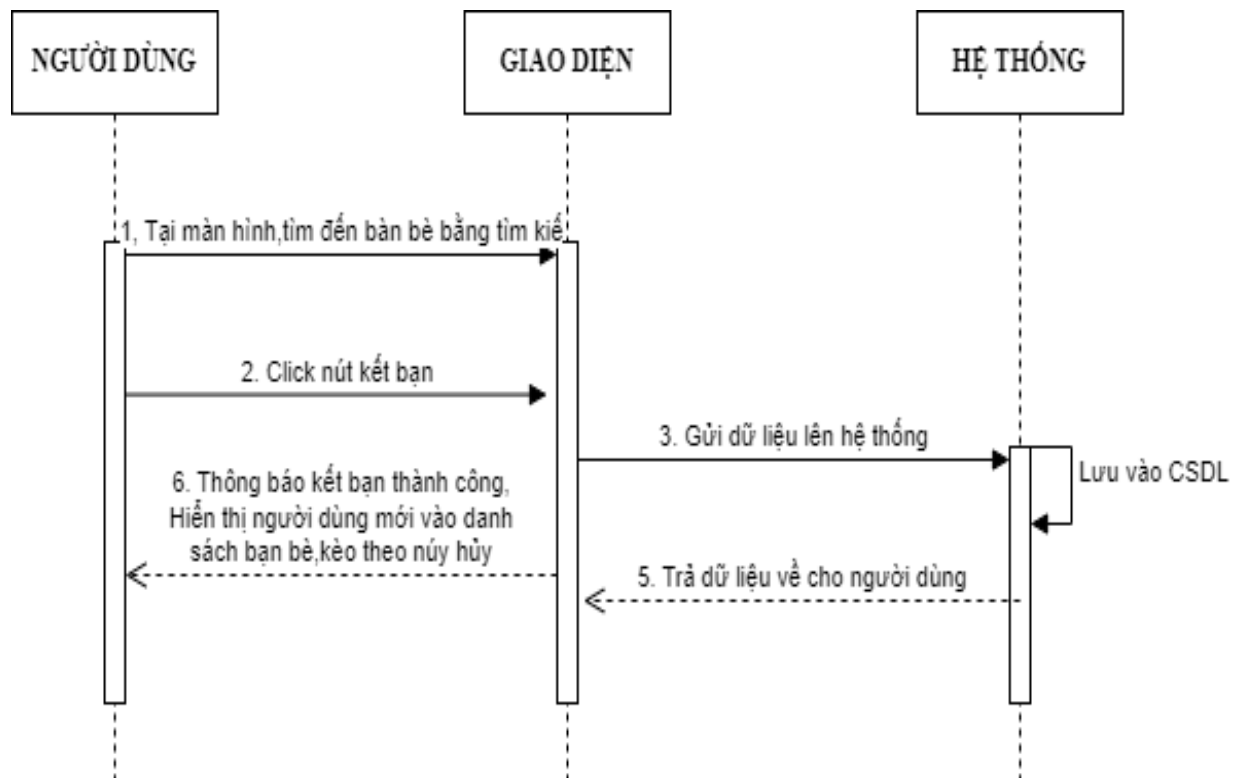
Hình 3.20. Biểu đồ tuần tự bình luận bài viết

- Biểu đồ tuần tự tìm kiếm



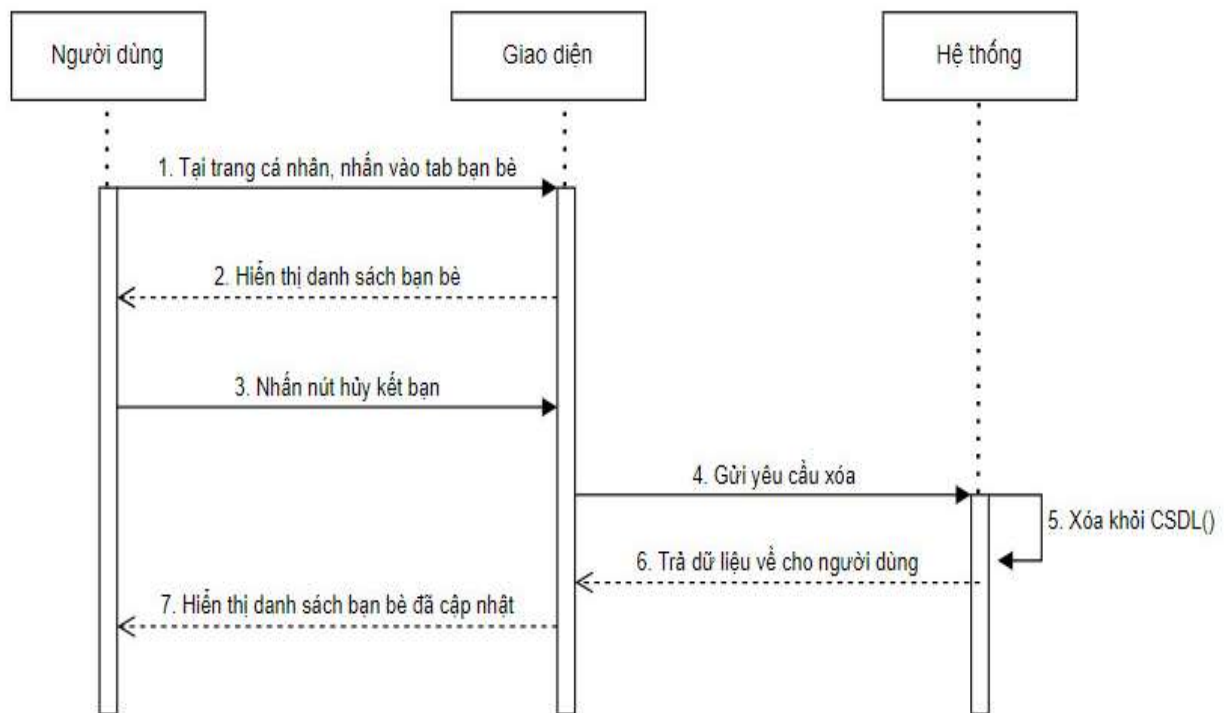
Hình 3.21. Biểu đồ tuần tự tìm kiếm

- Biểu đồ tuần tự kết bạn



Hình 3.22. Biểu đồ tuần tự kết bạn

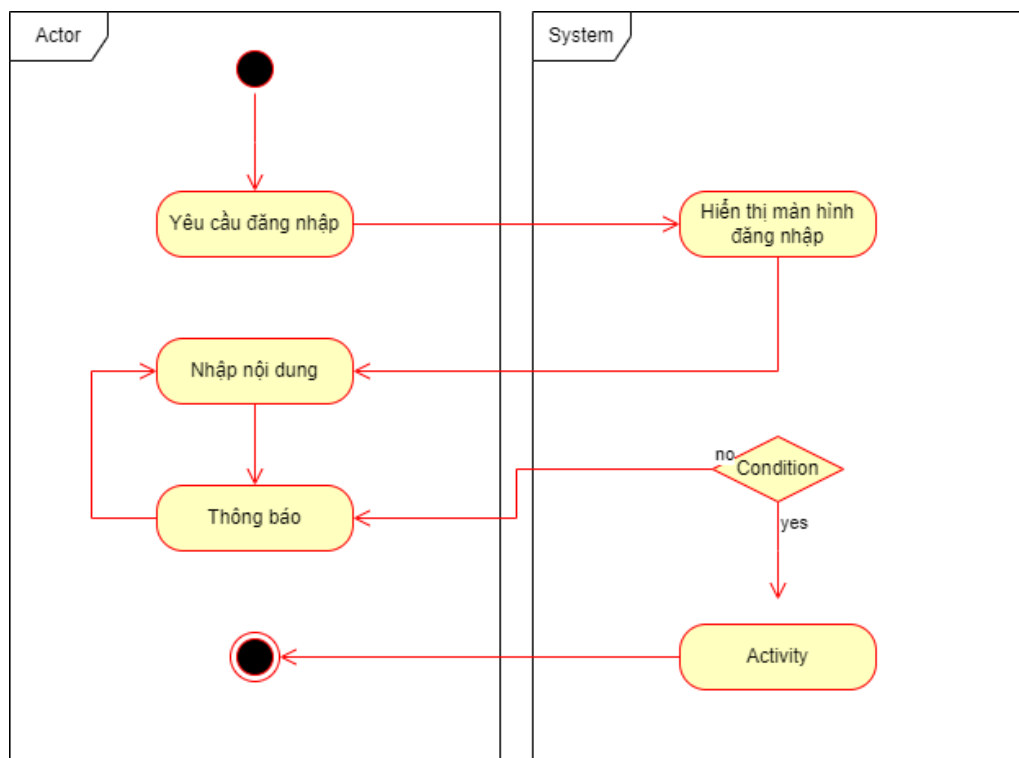
- Biểu đồ tuần tự hủy kết bạn



Hình 3.23. Biểu đồ tuần tự hủy kết bạn

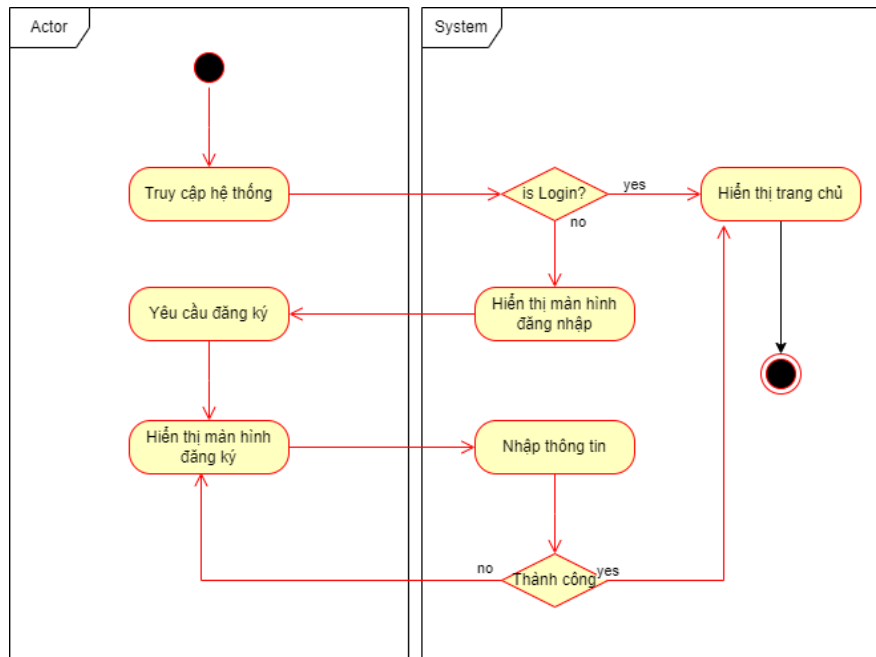
3.2.5. Biểu đồ hoạt động

- Biểu đồ hoạt động đăng nhập



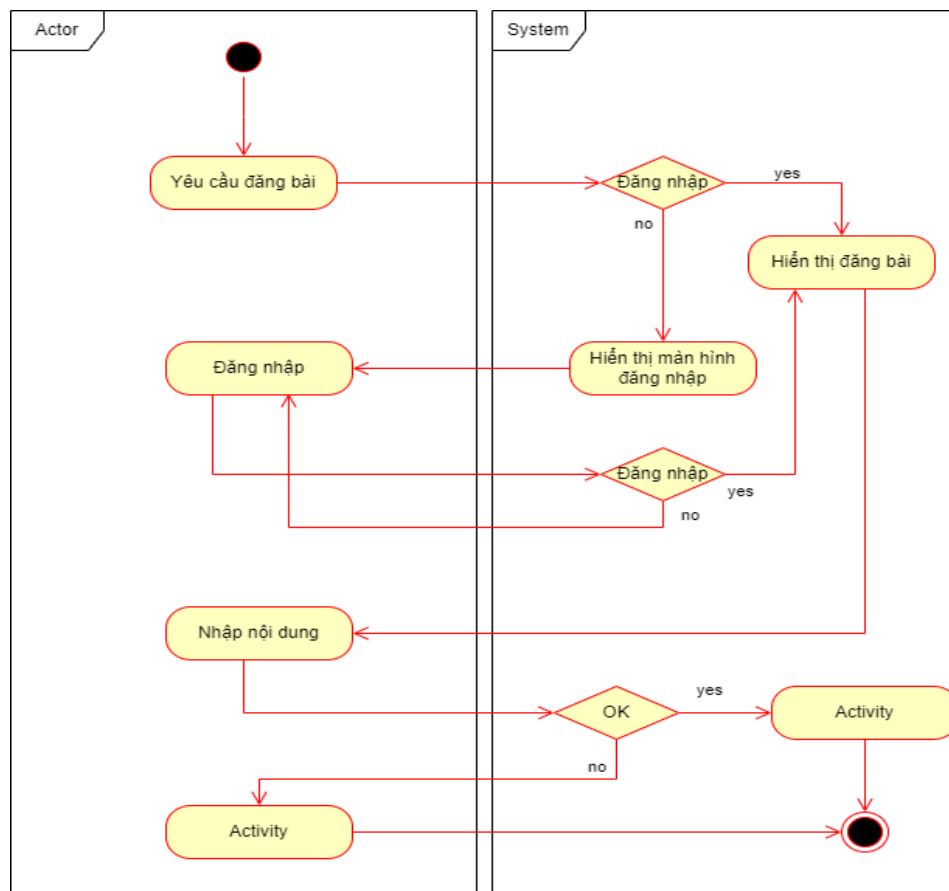
Hình 3.24. Biểu đồ hoạt động đăng nhập

- Biểu đồ hoạt động đăng ký



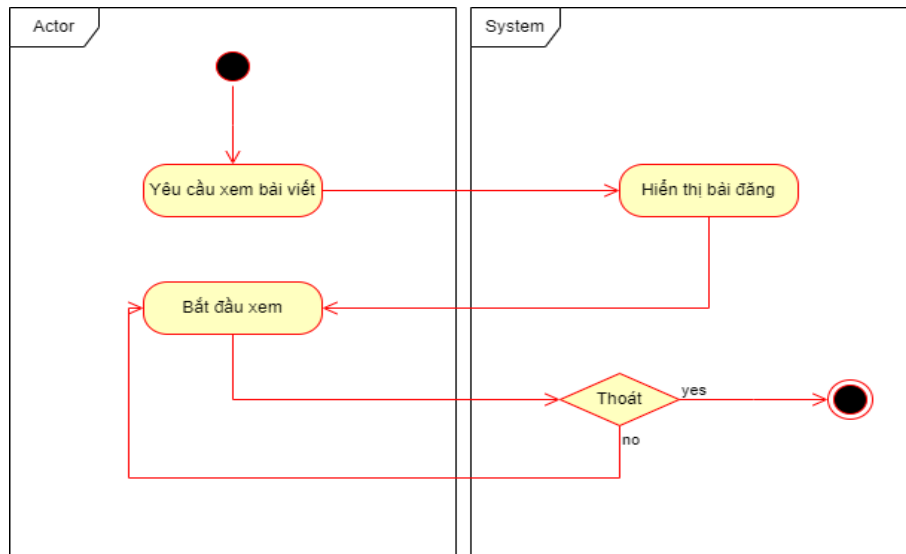
Hình 3.25. Biểu đồ hoạt động đăng ký

- Biểu đồ hoạt động đăng bài viết



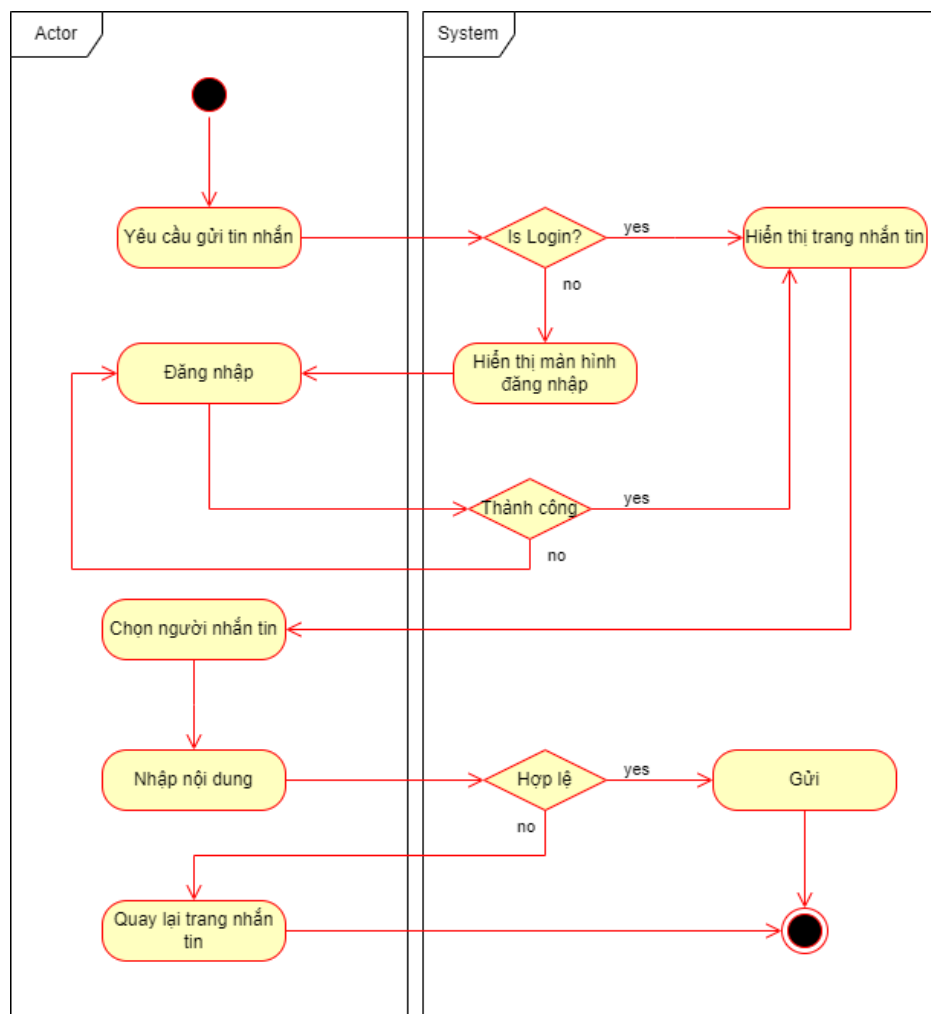
Hình 3.26. Biểu đồ hoạt động đăng bài viết

- Biểu đồ hoạt động xem bài viết



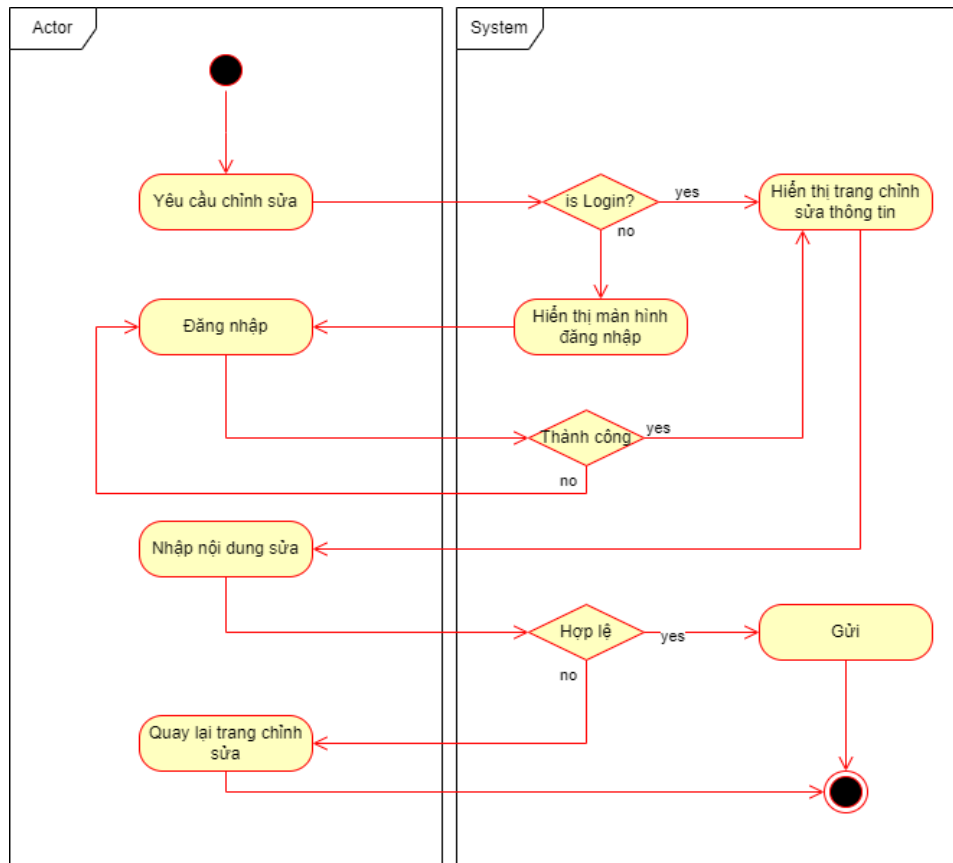
Hình 3.27. Biểu đồ hoạt động xem bài viết

- Biểu đồ hoạt động nhắn tin



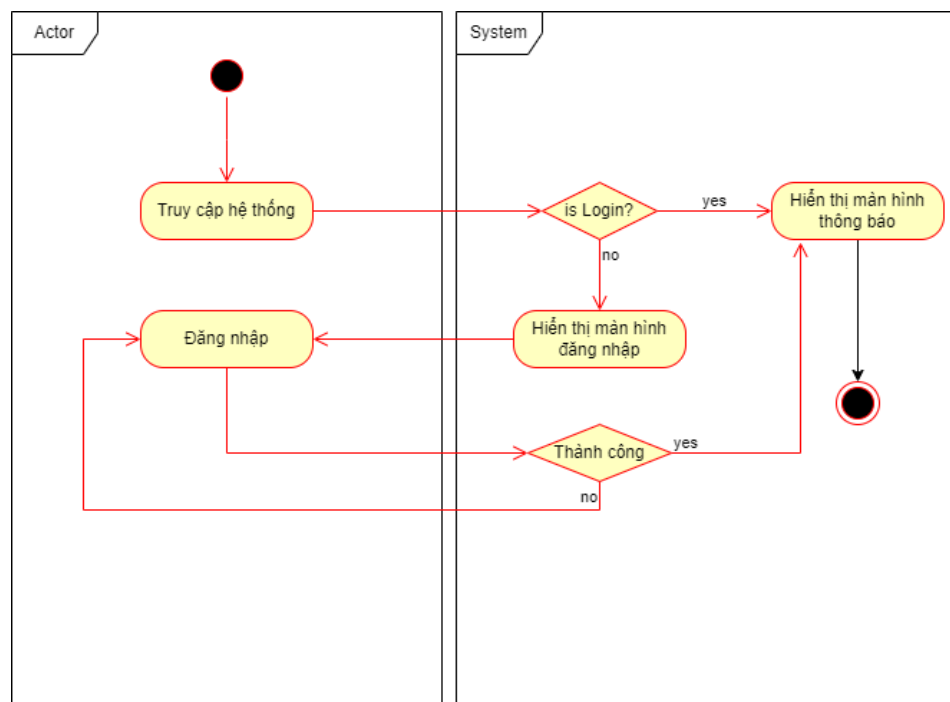
Hình 3.28. Biểu đồ hoạt động nhắn tin

- Biểu đồ hoạt động chỉnh sửa thông tin cá nhân



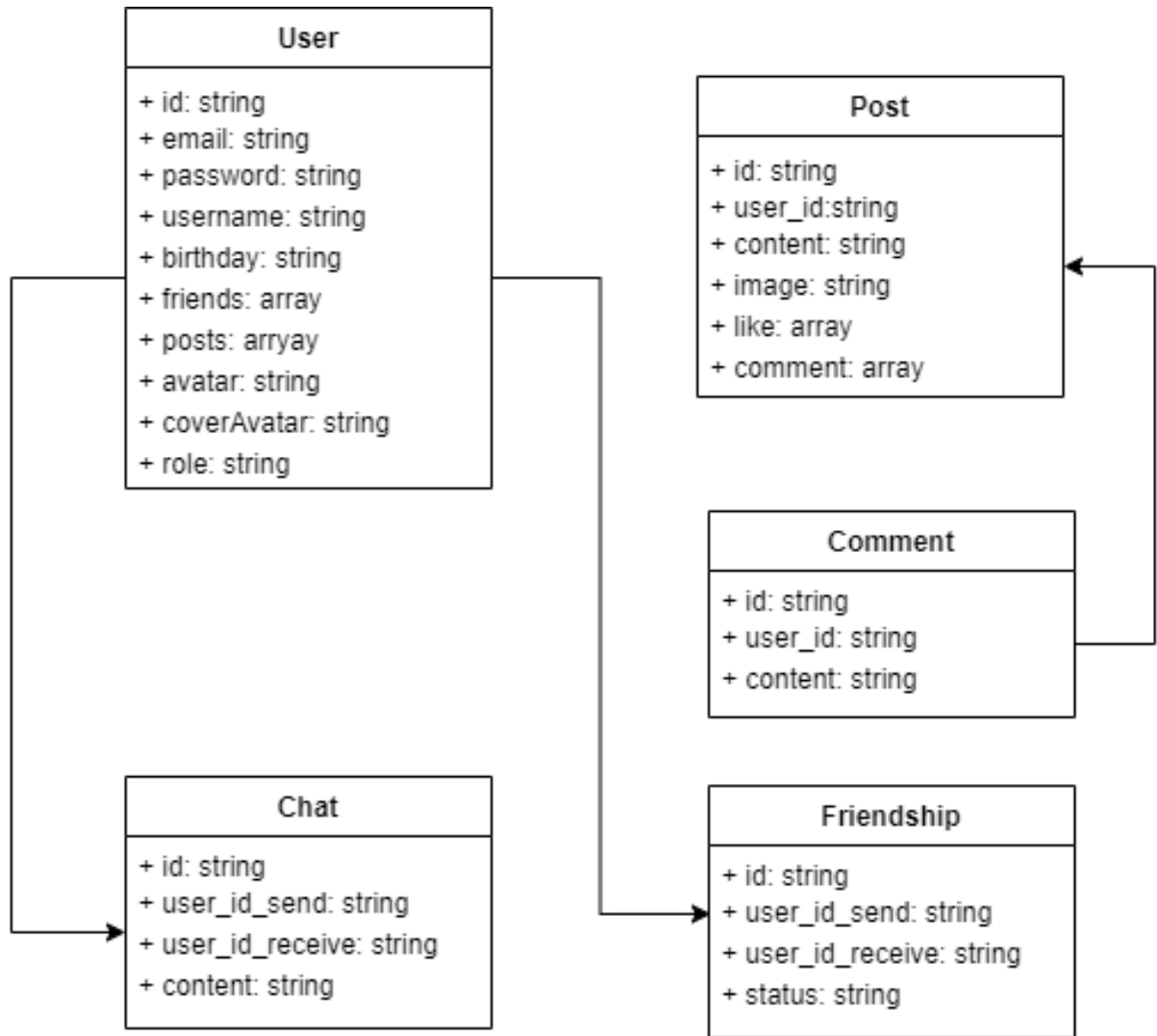
Hình 3.29. Biểu đồ hoạt động chỉnh sửa thông tin cá nhân

- Biểu đồ hoạt động xem thông báo



Hình 3.30. Biểu đồ hoạt động xem thông báo

3.2.6. Cơ sở dữ liệu



Hình 3.31. Cơ sở dữ liệu

Cách tổ chức dữ liệu:

- Bảng User

Bảng 3.15. Bảng User

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	string	ID duy nhất của người dùng, dùng để xác định một người dùng trong hệ thống.
email	string	Địa chỉ email của người dùng, dùng để đăng nhập và liên hệ.
password	string	Mật khẩu của người dùng, được mã hóa để bảo mật.

Tên thuộc tính	Kiểu dữ liệu	Mô tả
username	string	Tên người dùng, được sử dụng để hiển thị trên giao diện và có thể duy nhất.
birthday	string	Ngày sinh của người dùng, định dạng ngày tháng.
friends	array	Danh sách bạn bè của người dùng, chứa các ID của những người dùng khác.
posts	array	Danh sách các bài đăng của người dùng, chứa các ID các bài viết đã đăng.
avatar	string	Đường dẫn hoặc URL tới ảnh đại diện của người dùng.
coverAvatar	string	Đường dẫn hoặc URL tới ảnh bìa của người dùng.
role	string	Vai trò của người dùng trong hệ thống.

- Bảng Post

Bảng 3.16. Bảng Post

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	string	ID duy nhất của bài đăng, dùng để xác định một bài đăng trong hệ thống.
user_id	string	ID của người dùng đã tạo bài đăng, liên kết với bảng người dùng.
content	string	Nội dung văn bản của bài đăng.
image	string	Đường dẫn hoặc URL tới hình ảnh đính kèm trong bài đăng (nếu có).
likes	array	Danh sách các ID của người dùng đã thích bài đăng.
comments	array	Danh sách các ID của bình luận của người dùng trong bài đăng.

- Bảng Chat

Bảng 3.16. Bảng Chat

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	string	ID duy nhất của tin nhắn, dùng để xác định một tin nhắn trong hệ thống.
user_id_send	string	ID của người dùng gửi tin nhắn.
user_id_receive	string	ID của người dùng nhận tin nhắn.
content	string	Nội dung văn bản của tin nhắn.

- Bảng Comment

Bảng 3.17. Bảng Comment

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	string	ID duy nhất của bình luận, dùng để xác định một bình luận trong hệ thống.
user_id	string	ID của người dùng đã viết bình luận.
content	string	Nội dung văn bản của bình luận.

- Bảng Friendship

Bảng 3.18. Bảng Friendship

Tên thuộc tính	Kiểu dữ liệu	Mô tả
id	string	ID duy nhất của yêu cầu kết bạn, dùng để xác định một yêu cầu trong hệ thống.
user_id_send	string	ID của người dùng gửi yêu cầu kết bạn.
user_id_receive	string	ID của người dùng nhận yêu cầu kết bạn.
status	string	Trạng thái của yêu cầu kết bạn.

CHƯƠNG 4.

KẾT QUẢ VÀ ĐÁNH GIÁ

4.1. Kiểm thử

4.1.1. Kiểm thử API

- Chức năng đăng nhập và đăng ký

Bảng 4.1. Kiểm thử API đăng nhập và đăng ký

STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
1	Kiểm tra đăng nhập thành công	Gửi yêu cầu POST với thông tin đăng nhập hợp lệ (tên người dùng và mật khẩu)	Nhận về mã trạng thái 200 và thông tin người dùng đăng nhập thành công (mã token)	Nhận về mã trạng thái 200 và thông tin người dùng đăng nhập thành công (mã token)	Đạt
2	Kiểm tra đăng nhập thất bại với tên người dùng không tồn tại	Gửi yêu cầu POST với thông tin đăng nhập không hợp lệ (tên người dùng không tồn tại)	Nhận về mã trạng thái 401 hoặc 404 và thông báo lỗi "Tên người dùng không tồn tại"	Nhận về mã trạng thái 401 hoặc 404 và thông báo lỗi "Tên người dùng không tồn tại"	Đạt
3	Kiểm tra đăng nhập thất bại với mật khẩu không đúng	Gửi yêu cầu POST với thông tin đăng nhập không hợp lệ (mật khẩu không đúng)	Nhận về mã trạng thái 401 và thông báo lỗi "Mật khẩu không đúng"	Nhận về mã trạng thái 401 và thông báo lỗi "Mật khẩu không đúng"	Đạt

STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
4	Kiểm tra đăng ký thành công	Gửi yêu cầu POST với thông tin đăng ký hợp lệ	Nhận về mã trạng thái 201 và thông tin người dùng đã đăng ký thành công	Nhận về mã trạng thái 201 và thông tin người dùng đã đăng ký thành công	Đạt
5	Kiểm tra đăng ký thất bại với tên người dùng đã tồn tại	Gửi yêu cầu POST với thông tin đăng ký (tên người dùng đã tồn tại)	Nhận về mã trạng thái 409 và thông báo lỗi "Tên người dùng đã tồn tại"	Nhận về mã trạng thái 409 và thông báo lỗi "Tên người dùng đã tồn tại"	Đạt
6	Kiểm tra đăng ký thất bại với email đã tồn tại	Gửi yêu cầu POST với thông tin đăng ký (email đã tồn tại)	Nhận về mã trạng thái 409 và thông báo lỗi "Email đã tồn tại"	Nhận về mã trạng thái 409 và thông báo lỗi "Email đã tồn tại"	Đạt

- Chức năng đăng bài viết

Bảng 4.2. Kiểm thử API đăng bài viết

STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
1	Đăng bài viết với text và ảnh hợp lệ	1. Gửi yêu cầu POST với nội dung bài viết hợp lệ (bao gồm text và ảnh)	Nhận về mã trạng thái 201 và thông báo "Bài viết đã được đăng thành công"	Nhận về mã trạng thái 201 và thông báo "Bài viết đã được đăng thành công"	Đạt

STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
2	Đăng bài viết với tiêu đề rỗng và ảnh hợp lệ	1. Gửi yêu cầu POST chỉ với ảnh (tiêu đề rỗng)	Nhận về mã trạng thái 400 và thông báo lỗi "Tiêu đề bài viết không được để trống"	Nhận về mã trạng thái 400 và thông báo lỗi "Tiêu đề bài viết không được để trống"	Đạt
3	Đăng bài viết với tiêu đề hợp lệ và ảnh rỗng	1. Gửi yêu cầu POST chỉ với tiêu đề (ảnh rỗng)	Nhận về mã trạng thái 201 và thông báo "Bài viết đã được đăng thành công"	Nhận về mã trạng thái 201 và thông báo "Bài viết đã được đăng thành công"	Đạt
4	Đăng bài viết với cả tiêu đề và ảnh rỗng	1. Gửi yêu cầu POST không có tiêu đề và ảnh (cả hai đều trống)	Nhận về mã trạng thái 400 và thông báo lỗi "Tiêu đề bài viết không được để trống"	Nhận về mã trạng thái 400 và thông báo lỗi "Tiêu đề bài viết không được để trống"	Đạt

4.1.2. Kiểm thử chức năng

- Chức năng đăng nhập và đăng ký

Bảng 4.3. Kiểm thử chức năng đăng nhập và đăng ký

STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
1	Kiểm tra đăng nhập thành công với tên	1. Nhập tên người dùng hợp lệ vào trường tên người dùng	- Hiện thị thông báo "Đăng nhập thành công" -	Hiện thị thông báo "Đăng nhập thành công"	Đạt

STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
	người dùng và mật khẩu hợp lệ	2. Nhập mật khẩu hợp lệ vào trường mật khẩu 3. Nhấn nút Đăng nhập	- Chuyển hướng đến trang chính	Chuyển hướng đến trang chính	
2	Kiểm tra đăng nhập thất bại với tên người dùng không tồn tại	1. Nhập tên người dùng không tồn tại vào trường tên người dùng 2. Nhập mật khẩu hợp lệ vào trường mật khẩu 3. Nhấn nút Đăng nhập	Hiển thị thông báo "Tên người dùng hoặc mật khẩu không đúng"	Hiển thị thông báo "Tên người dùng hoặc mật khẩu không đúng"	Đạt
3	Kiểm tra đăng nhập thất bại với mật khẩu không đúng	1. Nhập tên người dùng hợp lệ vào trường tên người dùng 2. Nhập mật khẩu không đúng vào trường mật khẩu 3. Nhấn nút Đăng nhập	Hiển thị thông báo "Tên người dùng hoặc mật khẩu không đúng"	Hiển thị thông báo "Tên người dùng hoặc mật khẩu không đúng"	Đạt
4	Kiểm tra đăng ký thành công với thông tin hợp lệ	1. Nhập thông tin hợp lệ vào các trường đăng ký (tên người dùng, email, mật khẩu)	Hiển thị thông báo "Đăng ký thành công"	Hiển thị thông báo "Đăng ký thành công"	Đạt

STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
		2. Nhấn nút Đăng ký			
5	Kiểm tra đăng ký thất bại với tên người dùng đã tồn tại	1. Nhập tên người dùng đã tồn tại vào trường tên người dùng 2. Nhập thông tin hợp lệ cho các trường còn lại 3. Nhấn nút Đăng ký	Hiển thị thông báo "Tên người dùng đã tồn tại"	Hiển thị thông báo "Tên người dùng đã tồn tại"	Đạt
6	Kiểm tra đăng ký thất bại với email đã tồn tại	1. Nhập email đã tồn tại vào trường email 2. Nhập thông tin hợp lệ cho các trường còn lại 3. Nhấn nút Đăng ký	Hiển thị thông báo "Email đã tồn tại"	Hiển thị thông báo "Email đã tồn tại"	Đạt
7	Kiểm tra đăng ký thất bại với mật khẩu không đủ mạnh	1. Nhập thông tin hợp lệ vào các trường đăng ký (tên người dùng, email) 2. Nhập mật khẩu không đủ mạnh (ít hơn 6 ký tự)	Hiển thị thông báo "Mật khẩu phải có ít nhất 6 ký tự"	Hiển thị thông báo "Mật khẩu phải có ít nhất 6 ký tự"	Đạt

STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
		3. Nhấn nút Đăng ký			

- Chức năng đăng bài viết

Bảng 4.4. Kiểm thử chức năng đăng bài viết

STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
1	Kiểm tra đăng bài viết với tiêu đề và nội dung hợp lệ	1. Đăng nhập vào tài khoản hợp lệ 2. Nhấn vào nút "Đăng bài" 3. Nhập tiêu đề hợp lệ vào trường tiêu đề 4. Nhập nội dung hợp lệ vào trường nội dung 5. Nhấn nút "Đăng"	- Hiện thị thông báo "Bài viết đã được đăng thành công" - Bài viết xuất hiện trên trang chính	Hiện thị thông báo "Bài viết đã được đăng thành công" Bài viết xuất hiện trên trang chính	Đạt
2	Kiểm tra đăng bài viết mà không nhập tiêu đề	1. Đăng nhập vào tài khoản hợp lệ 2. Nhấn vào nút "Đăng bài" 3. Để trống trường tiêu đề 4. Nhập nội dung hợp lệ vào trường nội dung 5. Nhấn nút "Đăng"	- Hiện thị thông báo lỗi "Tiêu đề không được để trống"	Hiện thị thông báo lỗi "Tiêu đề không được để trống"	Đạt
3	Kiểm tra đăng bài viết mà không	1. Đăng nhập vào tài khoản hợp lệ 2. Nhấn	- Hiện thị thông báo lỗi "Nội dung	Hiện thị thông báo lỗi "Nội	Đạt

STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
	nhập nội dung	vào nút "Đăng bài" 3. Nhập tiêu đề hợp lệ vào trường tiêu đề 4. Để trống trường nội dung 5. Nhấn nút "Đăng"	không được để trống"	dung không được để trống"	
4	Kiểm tra đăng bài viết với tiêu đề dài hơn 255 ký tự	1. Đăng nhập vào tài khoản hợp lệ 2. Nhấn vào nút "Đăng bài" 3. Nhập tiêu đề dài hơn 255 ký tự vào trường tiêu đề 4. Nhập nội dung hợp lệ vào trường nội dung 5. Nhấn nút "Đăng"	- Hiện thị thông báo lỗi "Tiêu đề không được vượt quá 255 ký tự"	Hiện thị thông báo lỗi "Tiêu đề không được vượt quá 255 ký tự"	Đạt
5	Kiểm tra đăng bài viết với nội dung dài hơn 5000 ký tự	1. Đăng nhập vào tài khoản hợp lệ 2. Nhấn vào nút "Đăng bài" 3. Nhập tiêu đề hợp lệ vào trường tiêu đề 4. Nhập nội dung dài hơn 5000 ký tự vào trường nội dung 5. Nhấn nút "Đăng"	- Hiện thị thông báo lỗi "Nội dung không được vượt quá 5000 ký tự"	Hiện thị thông báo lỗi "Nội dung không được vượt quá 5000 ký tự"	Đạt

STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
6	Kiểm tra đăng bài viết với định dạng ảnh không hỗ trợ	1. Đăng nhập vào tài khoản hợp lệ 2. Nhấn vào nút "Đăng bài" 3. Nhập tiêu đề hợp lệ vào trường tiêu đề 4. Nhập nội dung hợp lệ vào trường nội dung 5. Thêm ảnh định dạng không hỗ trợ (ví dụ: .tiff) 6. Nhấn nút "Đăng"	- Hiện thị thông báo lỗi "Định dạng ảnh không được hỗ trợ"	Hiện thị thông báo lỗi "Định dạng ảnh không được hỗ trợ"	Đạt
7	Kiểm tra đăng bài viết với ảnh định dạng hợp lệ	1. Đăng nhập vào tài khoản hợp lệ 2. Nhấn vào nút "Đăng bài" 3. Nhập tiêu đề hợp lệ vào trường tiêu đề 4. Nhập nội dung hợp lệ vào trường nội dung 5. Thêm ảnh định dạng hợp lệ (ví dụ: .jpg, .png) 6. Nhấn nút "Đăng"	- Hiện thị thông báo "Bài viết đã được đăng thành công" - Bài viết với ảnh xuất hiện trên trang chính	Hiện thị thông báo "Bài viết đã được đăng thành công" Bài viết với ảnh xuất hiện trên trang chính	Đạt
8	Kiểm tra đăng bài viết khi không	1. Đăng nhập vào tài khoản hợp lệ 2. Ngắt	- Hiện thị thông báo lỗi "Không có	Hiện thị thông báo lỗi "Không	Đạt

STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
	có kết nối internet	kết nối internet 3. Nhấn vào nút "Đăng bài" 4. Nhập tiêu đề và nội dung hợp lệ 5. Nhấn nút "Đăng"	kết nối internet. Vui lòng thử lại sau."	có kết nối internet. Vui lòng thử lại sau."	
9	Kiểm tra hủy đăng bài viết	1. Đăng nhập vào tài khoản hợp lệ 2. Nhấn vào nút "Đăng bài" 3. Nhập tiêu đề và nội dung hợp lệ 4. Nhấn nút "Hủy"	- Không có bài viết mới được đăng - Quay lại trang chính hoặc trang đăng bài trống	Không có bài viết mới được đăng Quay lại trang chính hoặc trang đăng bài trống	Đạt

- Chức năng kết bạn

Bảng 4.5. Kiểm thử chức năng kết bạn

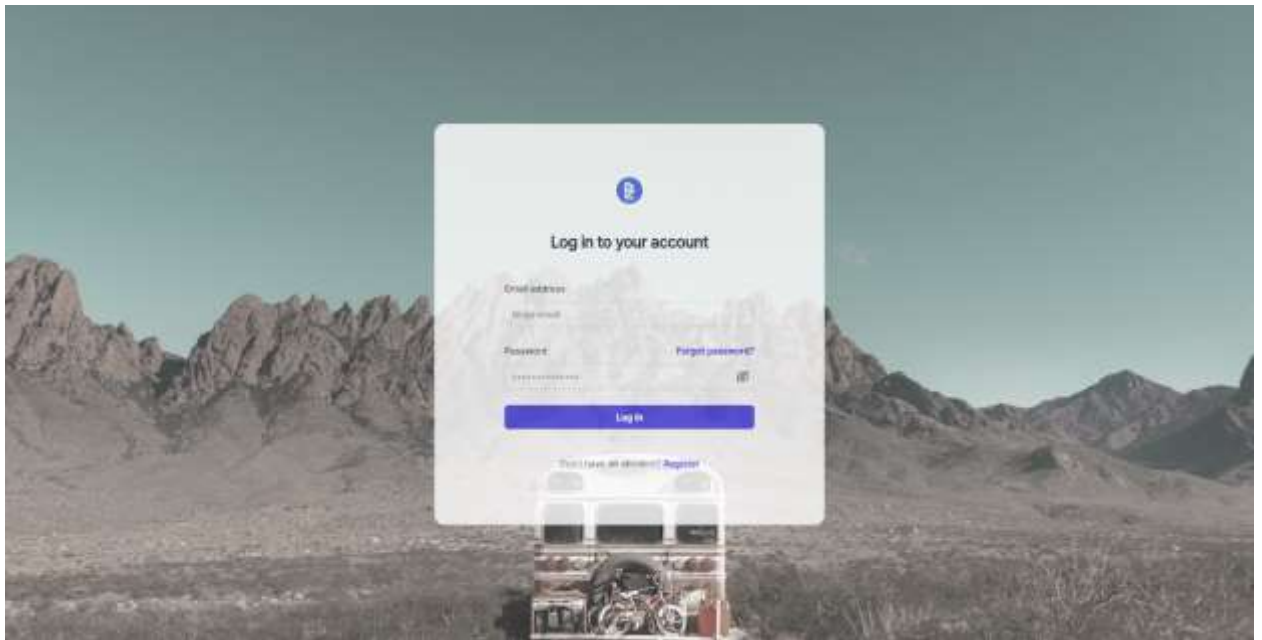
STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
1	Kiểm tra kết bạn thành công với người dùng tồn tại trong hệ thống	1. Đăng nhập vào tài khoản hợp lệ 2. Tìm kiếm người dùng cần kết bạn 3. Nhấn vào nút "Kết bạn"	- Hiện thị thông báo "Đã gửi lời mời kết bạn" - Người dùng nhận được thông báo mời kết bạn	Hiện thị thông báo "Đã gửi lời mời kết bạn" Người dùng nhận được thông báo mời kết bạn	Đạt
2	Kiểm tra kết bạn với người dùng không tồn	1. Đăng nhập vào tài khoản hợp lệ 2. Tìm kiếm người	- Hiện thị thông báo lỗi "Người dùng không tồn tại"	Hiện thị thông báo lỗi "Người	Đạt

STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
	tại trong hệ thống	dùng không tồn tại trong hệ thống 3. Nhấn vào nút "Kết bạn"		dùng không tồn tại"	
3	Kiểm tra kết bạn với chính tài khoản của mình	1. Đăng nhập vào tài khoản hợp lệ 2. Tìm kiếm tên tài khoản của mình 3. Nhấn vào nút "Kết bạn"	- Hiện thị thông báo lỗi "Không thể kết bạn với chính mình"	Hiện thị thông báo lỗi "Không thể kết bạn với chính mình"	Đạt
4	Kiểm tra kết bạn khi người dùng đã gửi lời mời kết bạn trước đó	1. Đăng nhập vào tài khoản hợp lệ 2. Tìm kiếm người dùng cần kết bạn 3. Nhấn vào nút "Kết bạn" lần đầu 4. Nhấn vào nút "Kết bạn" lần thứ hai	- Hiện thị thông báo lỗi "Đã gửi lời mời kết bạn trước đó"	Hiện thị thông báo lỗi "Đã gửi lời mời kết bạn trước đó"	Đạt
5	Kiểm tra chấp nhận lời mời kết bạn thành công	1. Đăng nhập vào tài khoản hợp lệ 2. Nhận thông báo lời mời kết bạn 3. Nhấn vào nút "Chấp nhận"	- Hiện thị thông báo "Đã kết bạn thành công" - Người dùng được thêm vào danh sách bạn bè	Hiện thị thông báo "Đã kết bạn thành công" Người dùng được thêm vào danh sách bạn bè	Đạt
6	Kiểm tra từ chối lời mời	1. Đăng nhập vào tài khoản	- Hiện thị thông báo	Hiện thị thông báo	Đạt

STT	Mô tả	Bước thực hiện	Kết quả dự kiến	Kết quả thực tế	Kết luận
	kết bạn thành công	hợp lệ 2. Nhận thông báo lời mời kết bạn 3. Nhấn vào nút "Từ chối"	"Đã từ chối lời mời kết bạn"	"Đã từ chối lời mời kết bạn"	
7	Kiểm tra hủy lời mời kết bạn thành công	1. Đăng nhập vào tài khoản hợp lệ 2. Gửi lời mời kết bạn đến người dùng khác 3. Trở lại danh sách lời mời kết bạn 4. Nhấn vào nút "Hủy"	- Hiện thị thông báo "Đã hủy lời mời kết bạn"	Hiện thị thông báo "Đã hủy lời mời kết bạn"	Đạt
8	Kiểm tra hủy kết bạn thành công	1. Đăng nhập vào tài khoản hợp lệ 2. Chọn một người bạn từ danh sách bạn bè 3. Nhấn vào nút "Hủy kết bạn"	- Hiện thị thông báo "Đã hủy kết bạn"	Hiện thị thông báo "Đã hủy kết bạn"	Đạt
9	Kiểm tra kết bạn thành công với người dùng tồn tại trong hệ thống	1. Đăng nhập vào tài khoản hợp lệ 2. Tìm kiếm người dùng cần kết bạn 3. Nhấn vào nút "Kết bạn"	- Hiện thị thông báo "Đã gửi lời mời kết bạn" - Người dùng nhận được thông báo mời kết bạn	Hiện thị thông báo "Đã gửi lời mời kết bạn" Người dùng nhận được thông báo mời kết bạn	Đạt

4.2. Kết quả đạt được

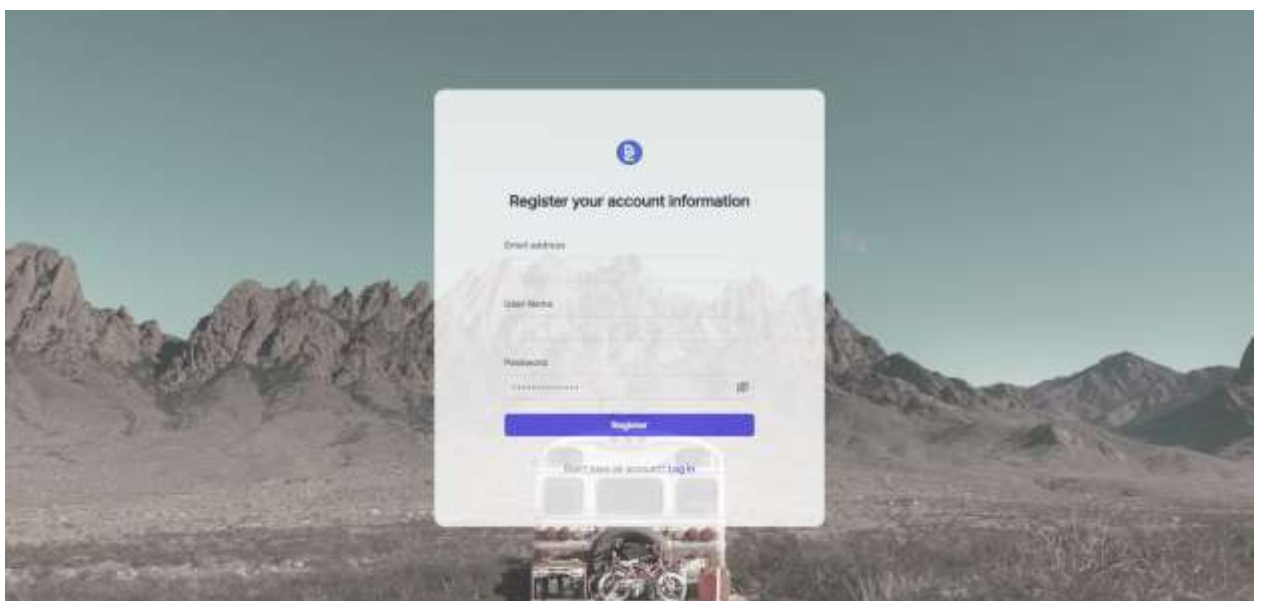
- Trang đăng nhập



Hình 4.1. Giao diện trang đăng nhập

Mô tả: Chức năng đăng nhập cho phép người dùng nhập email và mật khẩu để truy cập vào tài khoản cá nhân của họ. Khi thông tin được nhập và gửi đi qua nút "Đăng nhập", hệ thống sẽ xác thực thông tin này. Nếu thông tin chính xác, người dùng sẽ được chuyển đến trang chủ, nếu không, sẽ có thông báo lỗi.

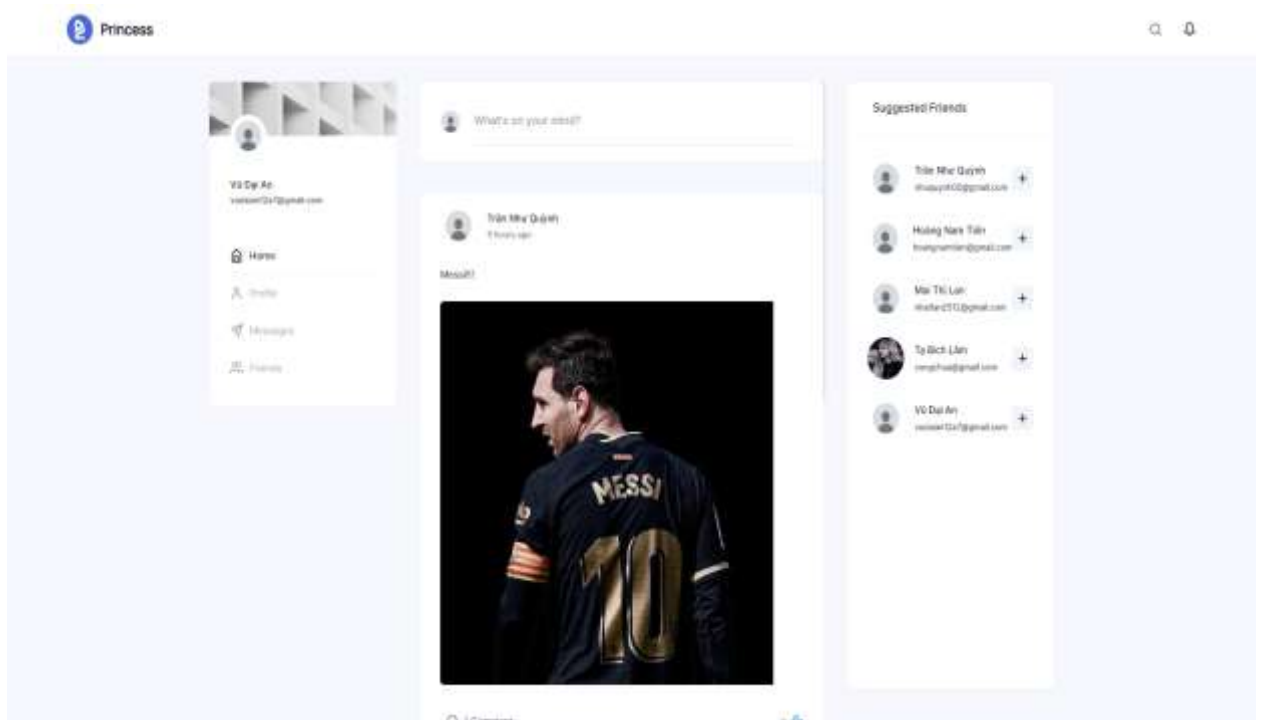
- Trang đăng ký



Hình 4.2. Giao diện trang đăng ký

Mô tả: Chức năng đăng ký của trang ứng dụng website mạng xã hội cho phép người dùng tạo tài khoản mới. Người dùng cần điền vào biểu mẫu đăng ký với các thông tin cá nhân như tên đăng nhập, email, mật khẩu. Sau khi hoàn tất, họ sẽ nhấn nút "Đăng ký" để gửi thông tin. Hệ thống sẽ kiểm tra tính hợp lệ của các thông tin này và tạo tài khoản mới nếu tất cả đều hợp lệ.

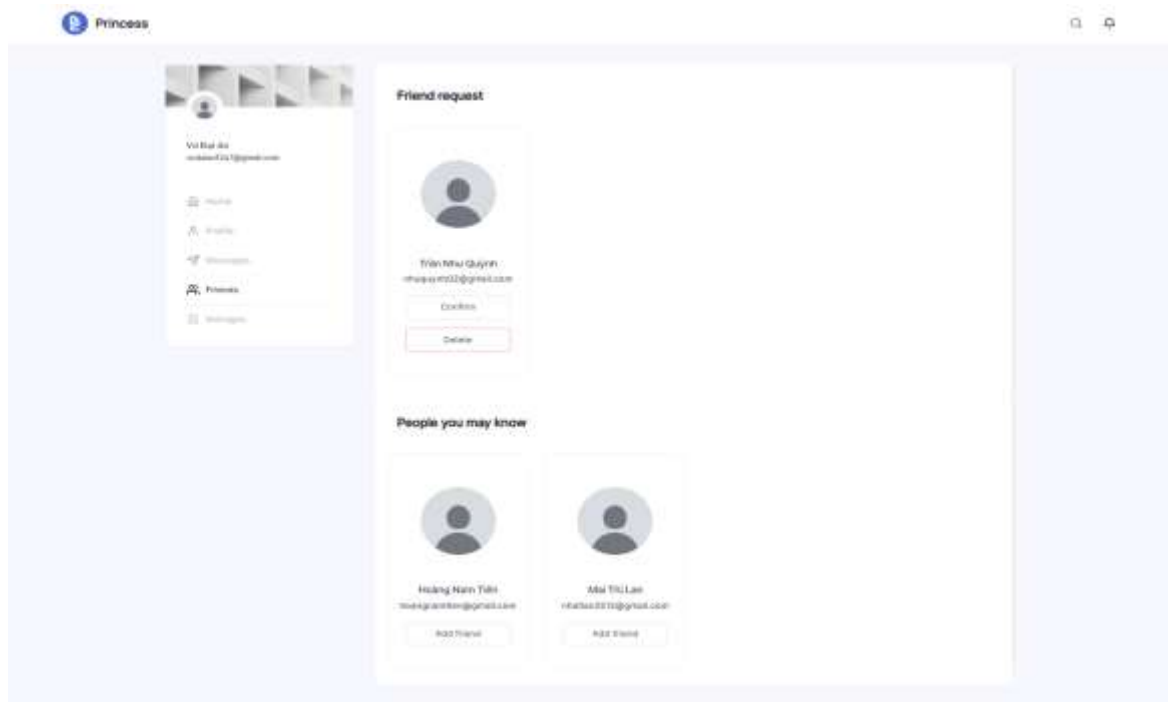
- Trang chủ



Hình 4.3. Giao diện trang chủ

Mô tả: Trang chủ của trang ứng dụng website mạng xã hội là nơi người dùng tiếp cận đầu tiên sau khi đăng nhập, cung cấp thông tin cập nhật và tương tác dễ dàng. Nó bao gồm bảng tin hiển thị các bài đăng mới nhất từ bạn bè, sắp xếp theo thứ tự thời gian đăng bài.

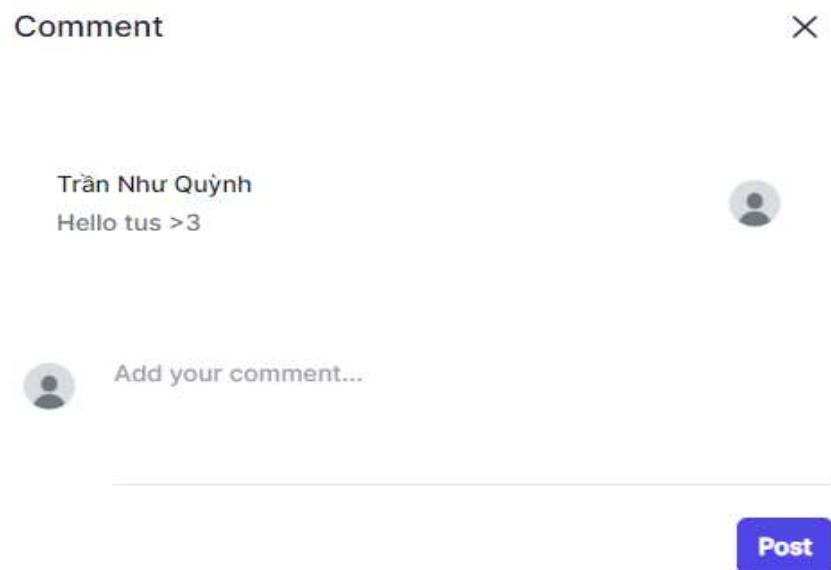
- Trang bạn bè



Hình 4.4. Giao diện trang bạn bè

Mô tả: Trang bạn bè của trang ứng dụng mạng xã hội cho phép người dùng quản lý và tương tác với danh sách bạn bè của mình. Nó hiển thị danh sách lời mời kết bạn, danh sách những người bạn có thể biết cùng thông tin cơ bản như tên và ảnh đại diện. Người dùng có thể xem và xử lý các lời mời kết bạn mới, nhận gợi ý kết bạn từ hệ thống.

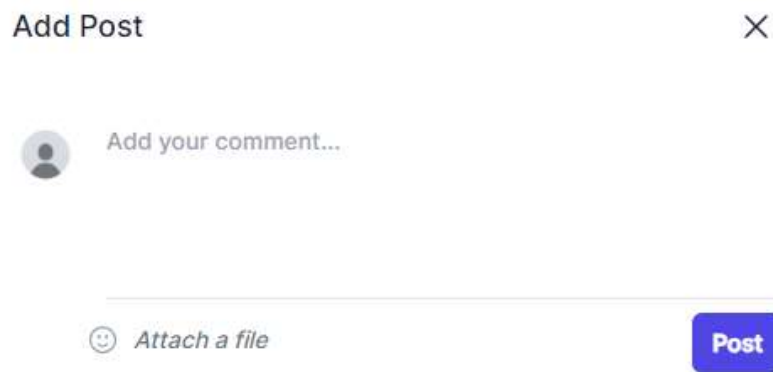
- Giao diện bình luận



Hình 4.5. Giao diện bình luận

Mô tả: Giao diện bình luận của trang ứng dụng mạng xã hội cho phép người dùng xem và tham gia vào các cuộc trò chuyện dưới các bài đăng. Nó hiển thị tất cả các bình luận của bài đăng theo thứ tự thời gian, bao gồm tên, ảnh đại diện. Người dùng có thể thêm bình luận mới qua ô nhập liệu.

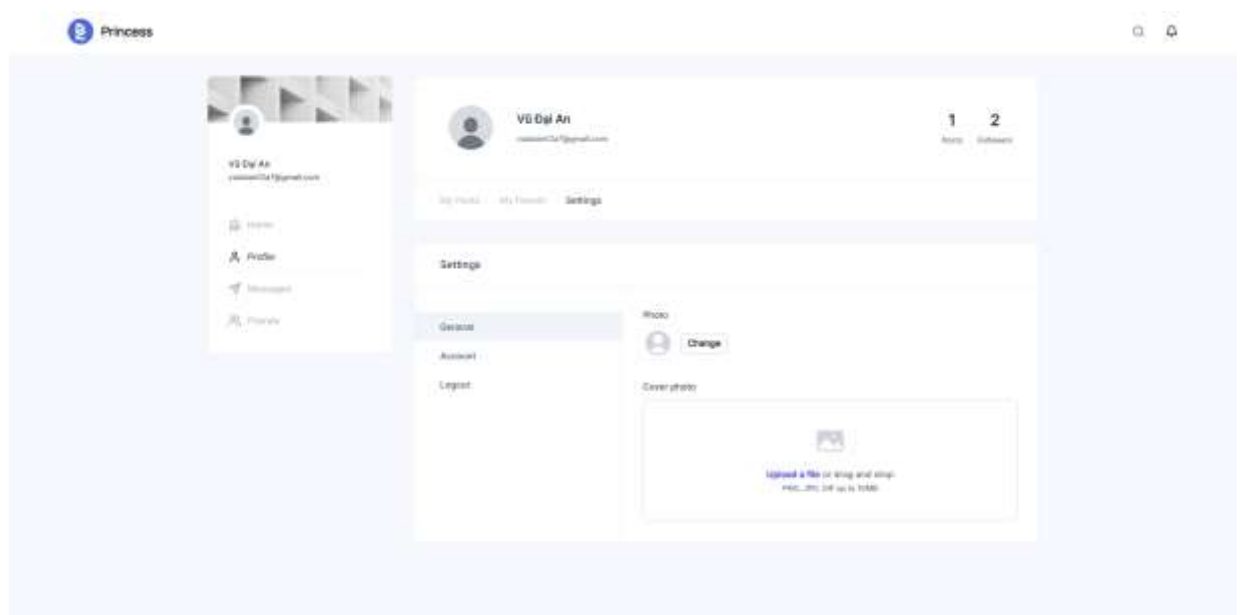
- Giao diện thêm bài viết



Hình 4.6. Giao diện thêm bài viết

Mô tả: Trang thêm bài viết của trang mạng xã hội là nơi người dùng có thể chia sẻ nội dung mới với cộng đồng của mình. Người dùng nội dung của bài viết trong biểu mẫu, có thể đính kèm hình ảnh. Đây là cách thuận tiện để người dùng chia sẻ ý kiến, trải nghiệm và thông tin mới với cộng đồng trực tuyến của họ.

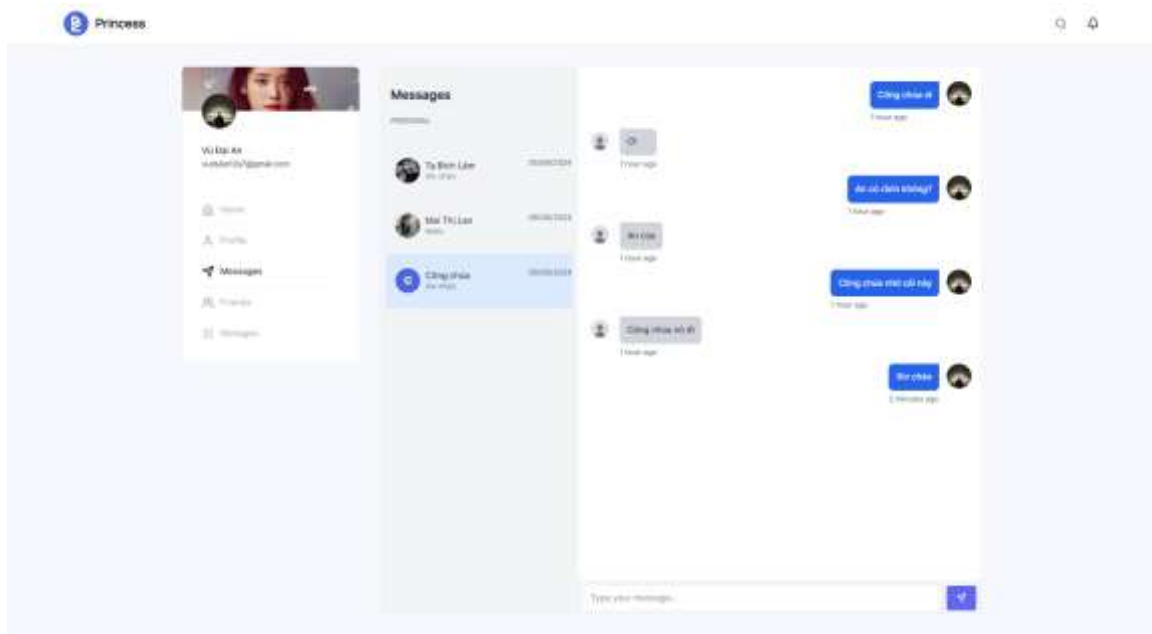
- Trang cá nhân



Hình 4.7. Giao diện trang cá nhân

Mô tả: quản lý các thông tin cá nhân như ảnh avatar, ảnh bìa, xóa vĩnh viễn tài khoản cá nhân, hiển thị số bài viết đã đăng, hiển thị số người đã kết bạn, thông tin chung như tên và gmail.

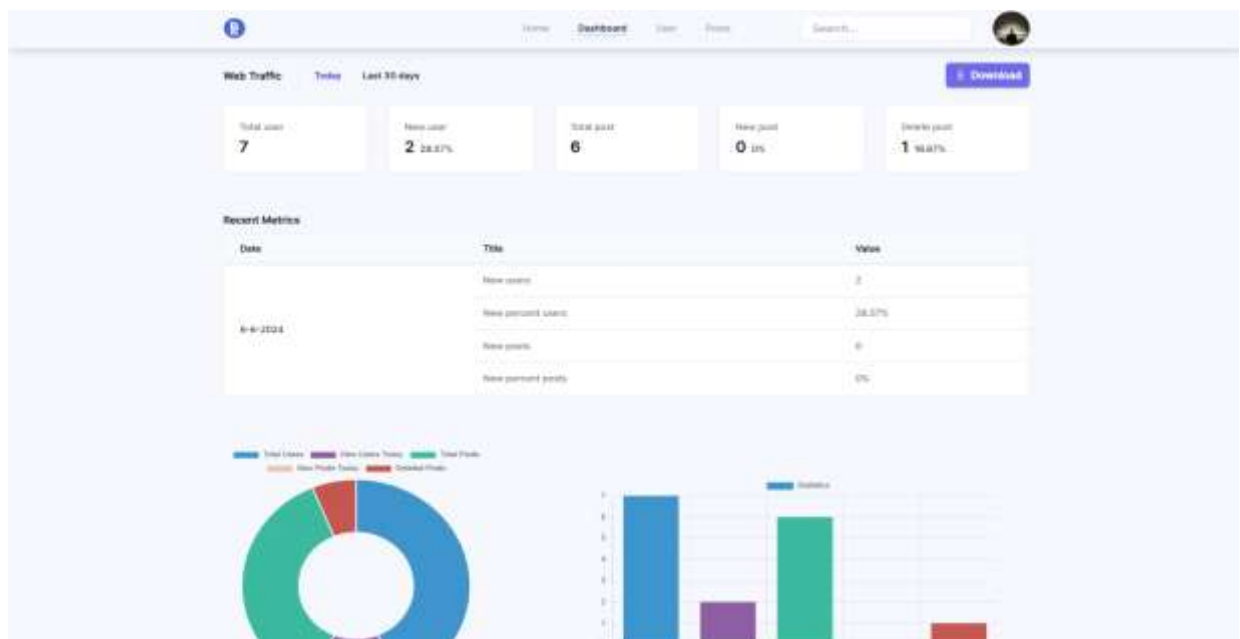
- Trang nhắn tin



Hình 4.8. Giao diện trang nhắn tin

Mô tả: Giao diện trang nhắn tin giúp người dùng gửi các tin nhắn văn bản tới những người dùng khác đã kết bạn trong hệ thống.

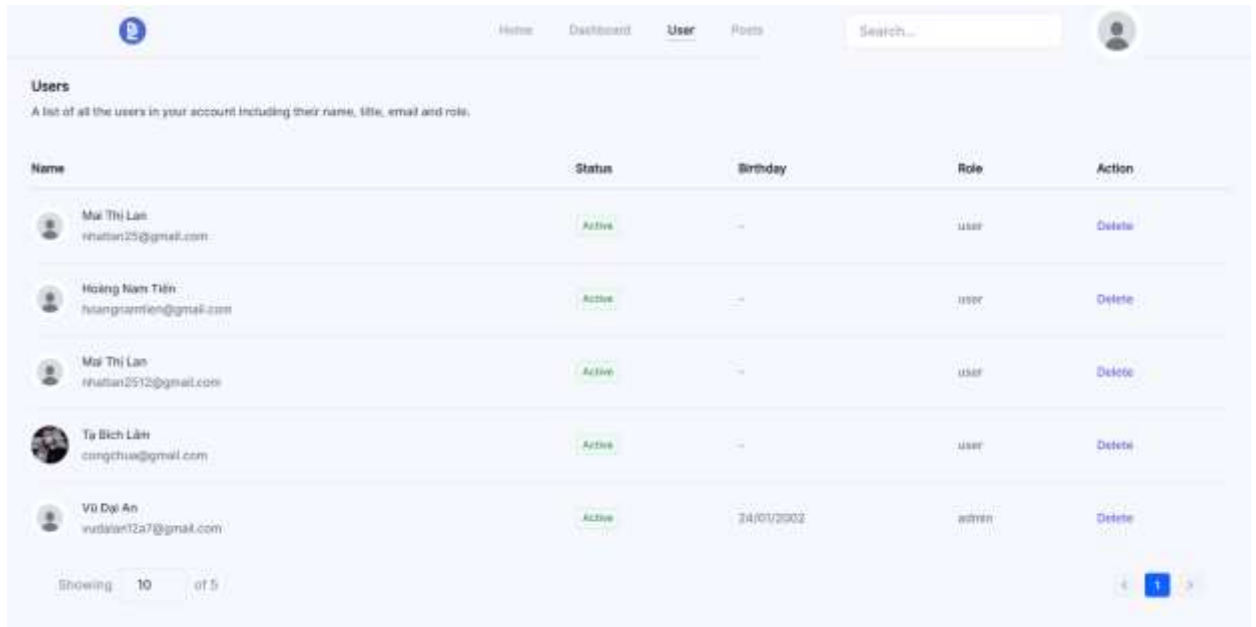
- Trang thống kê



Hình 4.9. Giao diện trang thống kê

Mô tả: Giao diện trang thống kê tổng người dùng, người dùng mới trong một ngày, một tháng, tổng số bài viết, tổng số bài viết mới trong một ngày, một tháng, tải file thống kê.

- Trang quản lý người dùng

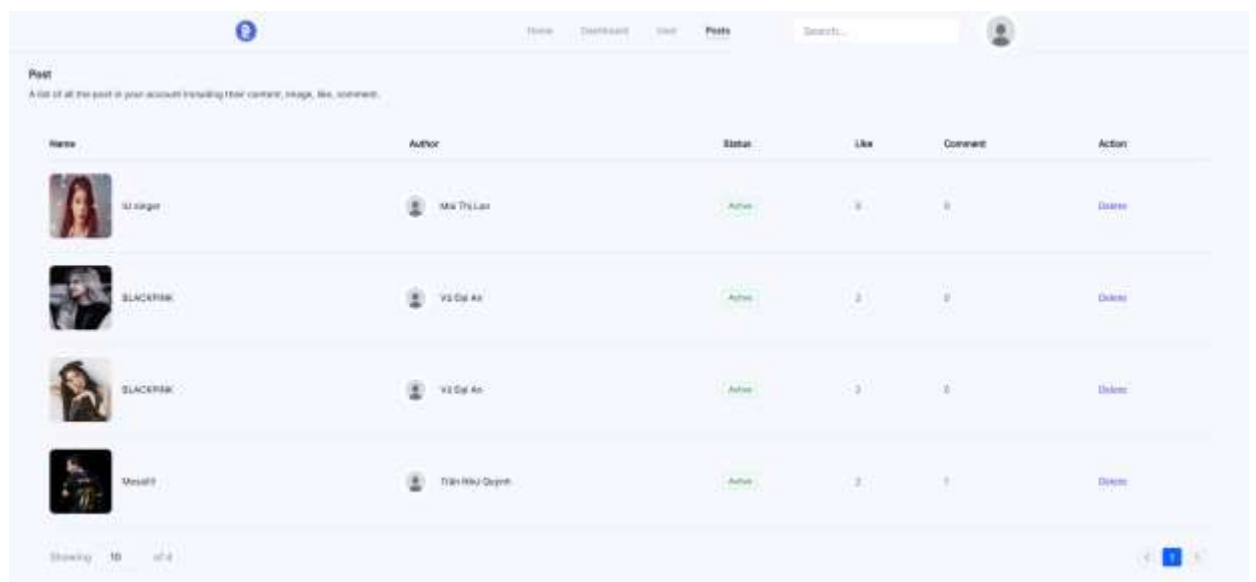


Name	Status	Birthday	Role	Action
Mai Thi Lan nhattan25@gmail.com	Active	--	USER	Delete
Hoàng Nam Tiến huangnamtien@gmail.com	Active	--	USER	Delete
Mai Thi Lan nhattan2512@gmail.com	Active	--	USER	Delete
Tạ Bích Liên congchua@gmail.com	Active	--	USER	Delete
Vũ Đại An vudalan12a7@gmail.com	Active	24/01/2002	ADMIN	Delete

Hình 4.10. Giao diện trang quản lý người dùng

Mô tả: Quản lý người dùng bao gồm hiển thị các thông tin người dùng bao gồm email, tên người dùng, ngày sinh, quyền, quản trị viên có thể xóa người dùng mà họ mong muốn hoặc tìm kiếm người dùng.

- Trang quản lý bài viết



Name	Author	Status	Like	Comment	Action
LƯU SINGER	Mai Thi Lan	Active	0	0	Delete
SLACKER	Vũ Đại An	Active	0	0	Delete
SLACKER	Vũ Đại An	Active	0	0	Delete
MUSIC	Tiến Nam Nguyễn	Active	0	0	Delete

Hình 4.11. Giao diện trang quản lý bài viết

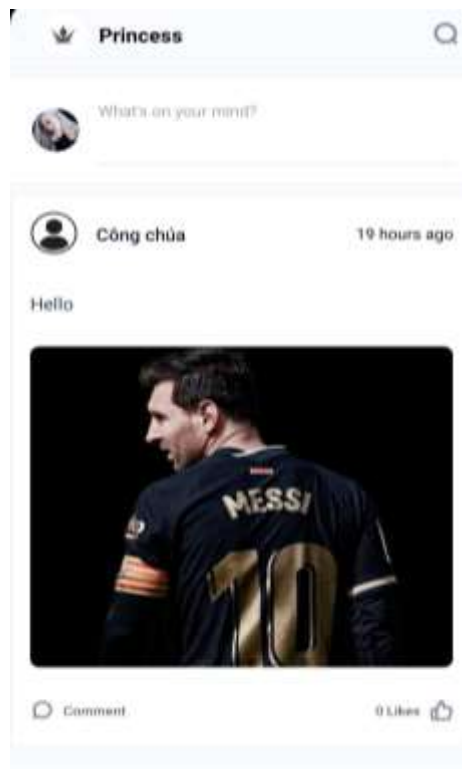
Mô tả: Quản lý các bài viết bao gồm hiển thị thông tin các bài viết, xóa các bài viết mà quản trị viên muốn, tìm kiếm các bài viết, quản lý bài viết giúp cho mạng xã hội có các bài viết lành mạnh hơn.

- Giao diện màn hình đăng nhập trên mobile



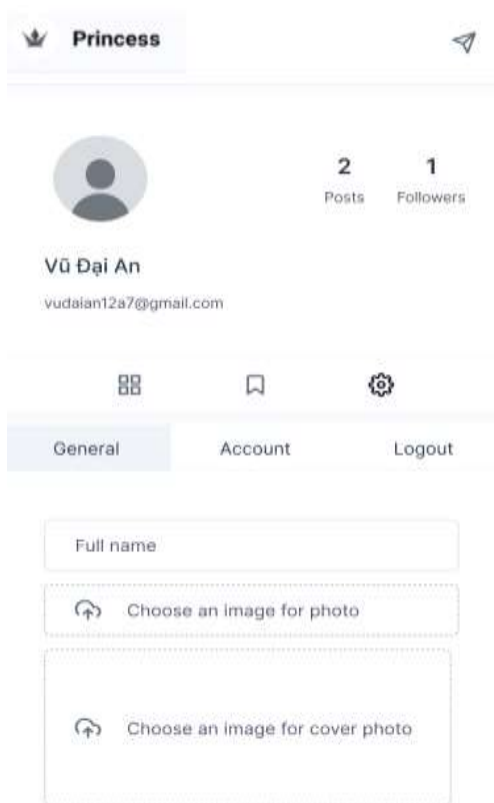
Hình 4.12. Giao diện màn hình đăng nhập trên mobile

- Giao diện màn hình xem bài viết trên mobile



Hình 4.13. Giao diện màn hình xem bài viết trên mobile

- Giao diện màn hình trang cá nhân trên mobile



Hình 4.14. Giao diện màn hình cá nhân trên mobile

- Giao diện màn hình trang nhắn tin trên mobile



Hình 4.15. Giao diện màn hình nhắn tin trên mobile

4.3. Hạn chế

- Hiệu suất

Quản lý bộ nhớ: React Native đôi khi gặp khó khăn trong việc quản lý bộ nhớ hiệu quả, đặc biệt là khi ứng dụng yêu cầu xử lý nhiều hình ảnh hoặc video.

Thời gian tải ứng dụng: Một số ứng dụng React Native có thể gặp vấn đề với thời gian khởi động chậm hơn so với ứng dụng native.

- Tính ổn định

Thư viện và plugin: Không phải tất cả các thư viện hoặc plugin bên thứ ba đều hỗ trợ tốt cho React Native, điều này có thể dẫn đến sự không tương thích hoặc lỗi khi sử dụng các tính năng đặc biệt.

Cập nhật thường xuyên: React Native và các thư viện liên quan thường xuyên được cập nhật, điều này có thể dẫn đến sự không tương thích và cần nhiều thời gian để cập nhật và sửa lỗi.

- Trải nghiệm người dùng (UX/UI)

Độ mượt mà của giao diện: Ứng dụng React Native có thể không mượt mà và phản hồi nhanh như ứng dụng native, đặc biệt là trên các thiết bị cũ hoặc có cấu hình thấp.

Giao diện người dùng phức tạp: Đối với các giao diện phức tạp hoặc yêu cầu tính tương tác cao, React Native có thể gặp khó khăn trong việc cung cấp trải nghiệm người dùng tốt như ứng dụng native.

- Khả năng tùy biến và tính năng đặc thù

Tính năng đặc thù của hệ điều hành: React Native có thể gặp khó khăn trong việc hỗ trợ các tính năng đặc thù của hệ điều hành hoặc phần cứng cụ thể, như các API chỉ có trên Android hoặc iOS.

Giới hạn của JavaScript: JavaScript, ngôn ngữ chính của React Native, có thể không mạnh mẽ và hiệu quả bằng các ngôn ngữ native như Swift (iOS) hay Kotlin (Android), đặc biệt khi xử lý các tác vụ nặng hoặc tính toán phức tạp.

- Kiến thức và kỹ năng của lập trình viên

Cần có kiến thức đa nền tảng: Lập trình viên cần hiểu biết về cả Android và iOS để xử lý các vấn đề cụ thể của từng nền tảng.

Kỹ năng React và JavaScript: Lập trình viên cần có kỹ năng tốt về React và JavaScript để phát triển và tối ưu hóa ứng dụng.

- Hỗ trợ kỹ thuật và cộng đồng

Tài liệu và hỗ trợ: Mặc dù cộng đồng React Native đang phát triển nhanh chóng, nhưng tài liệu và hỗ trợ cho các vấn đề phức tạp có thể không đầy đủ hoặc không kịp thời so với các nền tảng phát triển ứng dụng native.

4.4. Công việc tương lai

- Cải thiện hiệu suất và tối ưu hóa ứng dụng

Nâng cao hiệu suất: Tối ưu hóa mã nguồn, cải thiện quản lý bộ nhớ và giảm thiểu thời gian tải ứng dụng.

Sử dụng các kỹ thuật mới: Áp dụng các kỹ thuật như lazy loading, code splitting, và sử dụng các component tối ưu để cải thiện tốc độ và hiệu suất.

- Phát triển các tính năng tiên tiến

Tích hợp AI và Machine Learning: Tạo các tính năng thông minh như gợi ý bạn bè, tự động gắn thẻ (tag) trong ảnh, và phân tích tâm trạng từ văn bản.

Tăng cường bảo mật: Sử dụng các phương pháp bảo mật mạnh mẽ như xác thực hai yếu tố (2FA), mã hóa dữ liệu, và phát hiện xâm nhập.

- Cải thiện trải nghiệm người dùng (UX/UI)

Thiết kế giao diện hấp dẫn: Sử dụng các nguyên tắc thiết kế hiện đại để tạo ra giao diện thân thiện, trực quan và thẩm mỹ.

Tối ưu hóa trải nghiệm người dùng: Thực hiện thử nghiệm người dùng (user testing) để cải thiện trải nghiệm tổng thể và giảm thiểu các bước không cần thiết trong quá trình sử dụng ứng dụng.

- Mở rộng tính năng mạng xã hội

Tính năng tương tác xã hội: Phát triển các tính năng mới như livestream, phòng chat nhóm, và các cuộc thăm dò ý kiến (polls).

Hỗ trợ nội dung đa phương tiện: Cải thiện khả năng đăng tải và chia sẻ video, ảnh chất lượng cao, và âm thanh.

- Phát triển phiên bản cho nhiều nền tảng

Đồng bộ hóa đa nền tảng: Đảm bảo ứng dụng hoạt động mượt mà trên cả Android và IOS, cũng như trên Web.

Hỗ trợ thiết bị đeo thông minh: Phát triển các tính năng tích hợp với thiết bị đeo tay (smartwatch) và các thiết bị thông minh khác.

- Tạo cộng đồng và tương tác người dùng

Tạo tính năng nhóm cộng đồng: Cho phép người dùng tạo và tham gia các nhóm theo sở thích hoặc mục đích cụ thể.

Tăng cường tương tác: Phát triển các tính năng như thông báo tức thì, tin nhắn trực tiếp và phản hồi người dùng nhanh chóng.

KẾT LUẬN

Qua quá trình nghiên cứu và phát triển, đề tài "Tìm hiểu và xây dựng ứng dụng mạng xã hội trên nền tảng React Native" đã làm rõ các khái niệm cơ bản và quan trọng liên quan đến React Native, một framework mạnh mẽ và phổ biến cho việc phát triển ứng dụng di động.

Đề tài đã hoàn thành việc xây dựng một ứng dụng mạng xã hội cơ bản với các chức năng chính như đăng ký, đăng nhập, đăng bài viết, bình luận, và quản lý thông tin người dùng. Ứng dụng hoạt động ổn định trên cả hai nền tảng IOS và Android, thể hiện tính linh hoạt và hiệu quả của React Native.

Trong quá trình phát triển, em đã gặp phải một số khó khăn liên quan đến việc tích hợp các thư viện bên thứ ba, tối ưu hiệu suất và quản lý trạng thái phức tạp, cũng như việc xử lý các vấn đề về tương thích và hiệu suất trên nhiều thiết bị và hệ điều hành khác nhau.

Tuy nhiên, em đã đề xuất các giải pháp để cải thiện hiệu suất và tối ưu hóa trải nghiệm người dùng, tối ưu hóa hình ảnh, và quản lý bộ nhớ hiệu quả. Ứng dụng mạng xã hội này không chỉ giúp người dùng kết nối, chia sẻ thông tin mà còn là minh chứng cho tiềm năng của React Native trong việc phát triển các ứng dụng di động hiện đại.

Đề tài cũng mở ra cơ hội cho việc nghiên cứu và phát triển thêm các dự án công nghệ khác dựa trên nền tảng React Native, góp phần thúc đẩy sự phát triển của công nghệ thông tin và ứng dụng di động.

TÀI LIỆU THAM KHẢO

TIẾNG VIỆT

- [1]. Nguyễn Đức Anh, Lê Chí Luận, Phạm Thị Tố Nga, “*Công nghệ phần mềm*”, 2024, NXB Đại học Quốc gia Hà Nội.
- [2]. Nguyễn Văn Hùng, Trần Thị Minh Hòa, "Lập trình ReactJS cơ bản và nâng cao", 2023, NXB Đại học Quốc gia TP.HCM.
- [3]. Phạm Văn Cường, "MongoDB: Hệ quản trị cơ sở dữ liệu NoSQL", 2023, NXB Bách khoa Hà Nội.

WEBSITE

- [4]. <https://react.dev/learn>
- [5]. <https://www.mongodb.com/docs>
- [6]. <https://reactnative.dev/docs/environment-setup>