

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

ResearchHub - Mạng xã hội chia sẻ báo cáo nghiên cứu khoa học

Lại Văn Hoàn

hoan.lv187167@sis.hust.edu.vn

**Ngành Công nghệ thông tin và Truyền thông
Chuyên ngành Hệ thống thông tin**

Giảng viên hướng dẫn: TS. Đỗ Quốc Huy

Chữ kí GVHD

Giảng viên hướng dẫn: Th.S. Nguyễn Duy Hiệp

Chữ kí GVHD

Trường:

Công nghệ thông tin và truyền thông

HÀ NỘI, 08/2023

LỜI CẢM ƠN

Đầu tiên, xin chân thành cảm ơn Nhà trường, Ban giám hiệu, và tất cả các thầy cô giáo đã tạo điều kiện và cung cấp môi trường học tập tốt cho chúng tôi. Nhờ vào sự hỗ trợ và cống hiến của quý thầy cô, chúng tôi đã có cơ hội tiếp thu kiến thức, trang bị kỹ năng và phát triển bản thân một cách toàn diện.

Đặc biệt, chúng tôi xin gửi lời cảm ơn chân thành tới giảng viên hướng dẫn TS. Đỗ Quốc Huy. Thầy đã tận tình hướng dẫn từ những ngày đầu tiên của đồ án. Trong thời điểm bế tắc đối với đề tài cũ và không tìm được hướng phát triển mới, thầy đã ủng hộ và định hướng cho đồ án sang với đề tài mới và cũng là đề tài của đồ án này. Với sự định hướng và chỉ dạy của thầy, đồ án đã được hoàn thành và đạt được những yêu cầu đề ra.

Cuối cùng, tôi xin chân thành tri ân đến người thân và bạn bè, những người đã luôn đứng về phía tôi, cổ vũ, và hỗ trợ chúng tôi trên mọi hành trình. Sự ủng hộ và tình cảm của họ đã là nguồn động lực vô cùng quan trọng giúp chúng tôi vượt qua khó khăn và tiến lên phía trước.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong thế giới nghiên cứu khoa học, việc chia sẻ thông tin và kết quả nghiên cứu là rất quan trọng. Tuy nhiên, một số bài báo khoa học được đăng tải trên các trang web chỉ có thể được truy cập bằng cách trả phí hoặc đăng ký thành viên. Điều này gây khó khăn cho các nhà nghiên cứu muốn truy cập tài liệu cần thiết cho đề tài của mình. Đồ án này đề xuất xây dựng một mạng xã hội chia sẻ báo cáo nghiên cứu khoa học giúp cho các nhà nghiên cứu có thể tìm kiếm và truy cập tài liệu khoa học một cách dễ dàng và hiệu quả.

Hướng tiếp cận mà tôi lựa chọn để phát triển ứng dụng này là sử dụng công nghệ web để tạo ra một môi trường chân thực cho người dùng. Công nghệ sử dụng trong đề tài bao gồm các ngôn ngữ lập trình web như HTML, CSS và JavaScript. Sử dụng NodeJS(ExpressJS) để xây dựng phần back-end của trang web, quản lý cơ sở dữ liệu và xử lý các yêu cầu từ người dùng. Để tăng tính tương tác và tiện ích cho người dùng, đồ án cũng triển khai sử dụng socket để cung cấp tỉ lệ phản hồi nhanh chóng đối với các hoạt động trực tuyến như đăng tải báo cáo, bình luận và thông báo.

Sau khi hoàn thành, ứng dụng được triển khai trên domain render.com để người dùng có thể truy cập và sử dụng một cách thuận tiện.

Kết quả đồ án tốt nghiệp là một mạng xã hội chia sẻ báo cáo nghiên cứu khoa học hữu ích, giúp các nhà nghiên cứu tìm kiếm, truy cập và chia sẻ tài liệu khoa học một cách dễ dàng và hiệu quả. Tính năng socket cung cấp trải nghiệm tương tác tốt hơn cho người dùng, tạo ra một cộng đồng mạng xã hội chia sẻ kiến thức sôi động và tích cực. Đồ án sẽ tiếp tục được phát triển và cải tiến dựa trên phản hồi từ người dùng, nhằm tạo ra một môi trường mạng xã hội chia sẻ báo cáo khoa học hoàn thiện và phù hợp với nhu cầu của cộng đồng nghiên cứu khoa học.

Hà Nội, ngày tháng năm

Tác giả ĐATN

Lại Văn Hoàn

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án	2
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	4
2.1 Khảo sát hiện trạng	4
2.1.1 Các trang mạng xã hội hiện có	4
2.2 Tổng quan chức năng	5
2.2.1 Biểu đồ use case tổng quát	5
2.2.2 Biểu đồ use case phân rã quản lý bài đăng	6
2.2.3 Biểu đồ use case phân rã tìm kiếm thông tin	7
2.2.4 Biểu đồ use case phân rã quản lý thông tin	8
2.3 Đặc tả chức năng	9
2.3.1 Đặc tả use case đăng bài viết mới	9
2.3.2 Đặc tả use case chỉnh sửa bài đăng.....	10
2.3.3 Đặc tả use case xóa bài viết.....	11
2.3.4 Đặc tả use case thích bài viết.....	12
2.3.5 Đặc tả use case bình luận bài viết	13
2.3.6 Đặc tả use case gửi tin nhắn	14
2.3.7 Đặc tả use case gợi ý theo dõi người dùng	14
2.3.8 Đặc tả use case tìm kiếm thông tin.....	15
2.4 Biểu đồ Trình tự	16
2.4.1 Biểu đồ trình tự tạo bài đăng mới.....	16

2.4.2 Biểu đồ trình tự chỉnh sửa bài đăng.....	17
2.4.3 Biểu đồ trình tự xóa bài đăng	18
2.4.4 Biểu đồ trình tự xem bài đăng	19
2.4.5 Biểu đồ trình tự thích/ bỏ thích bài đăng.....	20
2.4.6 Biểu đồ trình tự bình luận bài đăng	21
2.4.7 Biểu đồ trình tự chỉnh sửa bình luận.....	22
2.4.8 Biểu đồ trình tự xóa bình luận	23
2.4.9 Biểu đồ trình tự chỉnh sửa thông tin người dùng.....	24
2.4.10 Biểu đồ trình tự xem thông tin người dùng.....	25
2.4.11 Biểu đồ trình tự gợi ý người dùng	25
2.4.12 Biểu đồ trình tự theo dõi người dùng	26
2.4.13 Biểu đồ trình tự xem danh sách người theo dõi.....	27
2.4.14 Biểu đồ trình tự xem danh sách nhắn tin	27
2.4.15 Biểu đồ trình tự xem tin nhắn	28
2.4.16 Biểu đồ trình tự gửi tin nhắn.....	29
2.4.17 Biểu đồ trình tự tìm kiếm thông tin	30
2.5 Biểu đồ Hoạt động.....	31
2.5.1 Biểu đồ hoạt động xem thông tin người dùng.....	31
2.5.2 Biểu đồ hoạt động gửi tin nhắn.....	32
2.5.3 Biểu đồ hoạt động xem danh sách nhắn tin	33
2.5.4 Biểu đồ hoạt động xem tin nhắn với người dùng khác	33
2.5.5 Biểu đồ hoạt động xem danh sách người theo dõi	34
2.5.6 Biểu đồ hoạt động xem danh sách người dùng gợi ý	34
2.5.7 Biểu đồ hoạt động bình luận bài đăng	35
2.5.8 Biểu đồ hoạt động chỉnh sửa bình luận.....	36
2.5.9 Biểu đồ hoạt động xóa bình luận.....	37

2.6 Yêu cầu phi chức năng	38
2.6.1 Yêu cầu về kỹ thuật	38
2.6.2 Yêu cầu về thiết kế	38
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	39
3.1 Javascript.....	39
3.2 Frontend	39
3.3 Lý do lựa chọn sự kết hợp cả ReactJS, Redux và Axios	40
3.4 Backend - NodeJS/Express	41
3.5 Database - MongoDB	41
3.6 Deploy - Render.com	42
CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ	43
4.0.1 Lựa chọn kiến trúc phần mềm	43
4.0.2 Thiết kế tổng quan.....	44
4.0.3 Thiết kế chi tiết gói	44
4.1 Thiết kế chi tiết.....	45
4.1.1 Thiết kế giao diện	45
4.1.2 Thiết kế lớp	48
4.1.3 Thiết kế cơ sở dữ liệu	53
4.2 Xây dựng ứng dụng.....	56
4.2.1 Thư viện và công cụ sử dụng.....	56
4.2.2 Kết quả đạt được	56
4.2.3 Minh họa các chức năng chính	57
4.3 Kiểm thử.....	60
4.4 Triển khai	61

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT	62
5.1 Hiệu năng trang web	62
5.1.1 Vấn đề gặp phải về tốc độ trang web	62
5.1.2 Quá trình giải quyết.....	62
5.1.3 Kết quả	62
5.2 Khó khăn khi triển khai chức năng search với MongoDB Text Search.....	63
5.2.1 Khó khăn gặp phải	63
5.2.2 Giải pháp	63
5.2.3 Kết quả	64
5.3 Tăng trải nghiệm người dùng với Socket.IO.....	64
5.3.1 Vấn đề cập nhật thông báo	64
5.3.2 Giải pháp	64
5.3.3 Kết quả	65
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	66
6.1 Kết luận	66
6.2 Hướng phát triển.....	66
6.2.1 Hoàn thiện các chức năng/nhiệm vụ đã làm	66
6.2.2 Phân tích các hướng đi mới	67
TÀI LIỆU THAM KHẢO.....	68

DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ Use Case tổng quát.	5
Hình 2.2	Biểu đồ use case phân rã quản lý bài đăng	6
Hình 2.3	Biểu đồ use case phân rã tìm kiếm thông tin	7
Hình 2.4	Biểu đồ use case phân rã quản lý thông tin	8
Hình 2.5	Biểu đồ trình tự tạo bài đăng mới	16
Hình 2.6	Biểu đồ trình tự chỉnh sửa bài đăng	17
Hình 2.7	Biểu đồ trình tự xóa bài đăng	18
Hình 2.8	Biểu đồ trình tự xem bài đăng	19
Hình 2.9	Biểu đồ trình tự thích/ bỏ thích bài đăng	20
Hình 2.10	Biểu đồ trình tự bình luận bài đăng	21
Hình 2.11	Biểu đồ trình tự chỉnh sửa bình luận	22
Hình 2.12	Biểu đồ trình tự xóa bình luận	23
Hình 2.13	Biểu đồ trình tự chỉnh sửa thông tin người dùng	24
Hình 2.14	Biểu đồ trình tự xem thông tin người dùng	25
Hình 2.15	Biểu đồ trình tự gợi ý người dùng	25
Hình 2.16	Biểu đồ trình tự theo dõi người dùng	26
Hình 2.17	Biểu đồ trình tự xem danh sách người theo dõi	27
Hình 2.18	Biểu đồ trình tự xem danh sách nhắn tin	27
Hình 2.19	Biểu đồ trình tự xem tin nhắn	28
Hình 2.20	Biểu đồ trình tự gửi tin nhắn	29
Hình 2.21	Biểu đồ trình tự tìm kiếm thông tin	30
Hình 2.22	Biểu đồ hoạt động xem thông tin người dùng	31
Hình 2.23	Biểu đồ hoạt động gửi tin nhắn	32
Hình 2.24	Biểu đồ hoạt động xem tin nhắn	33
Hình 2.25	Biểu đồ hoạt động xem tin nhắn với người dùng khác	33
Hình 2.26	Biểu đồ hoạt động xem danh sách người theo dõi	34
Hình 2.27	Biểu đồ hoạt động xem danh sách người dùng gợi ý	34
Hình 2.28	Biểu đồ hoạt động bình luận bài đăng	35
Hình 2.29	Biểu đồ hoạt động chỉnh sửa bình luận	36
Hình 2.30	Biểu đồ hoạt động xóa bình luận	37
Hình 3.1	Danh sách công nghệ áp dụng	39
Hình 4.1	Biểu đồ phụ thuộc gói	44
Hình 4.2	Thiết kế chi tiết gói	44
Hình 4.3	Thiết kế giao diện màn hình chính của trang web	46

Hình 4.4	Thiết kế giao diện profile page	46
Hình 4.5	Thiết kế giao diện chat	47
Hình 4.6	Thiết kế giao diện chi tiết báo cáo khoa học	48
Hình 4.7	Biểu đồ lớp liên quan đến chức năng quản lý các bài đăng . . .	48
Hình 4.8	Biểu đồ trình tự đăng bài	52
Hình 4.9	ERD	53
Hình 4.10	Màn hình chính	57
Hình 4.11	Màn hình trang cá nhân người dùng	58
Hình 4.12	Màn hình trang cá nhân người dùng(publication tab)	59
Hình 4.13	Màn hình hiển thị chi tiết báo cáo	59
Hình 5.1	Hình ảnh tải danh sách bài viết đã được phân trang	62
Hình 5.2	Thực hiện đánh index cho trường content trong MongoDB để sử dụng text search	64
Hình 5.3	Hình ảnh thông báo thời gian thực khi sử dụng socket.io	65

DANH MỤC BẢNG BIỂU

Bảng 2.1	Mô tả chức năng trong UseCase tổng quát	5
Bảng 2.2	Mô tả chức năng trong use case phân rã quản lý bài đăng . . .	6
Bảng 2.3	Mô tả chức năng trong use case phân rã tìm kiếm thông tin . .	7
Bảng 2.4	Mô tả chức năng trong use case phân rã quản lý thông tin . . .	8
Bảng 2.5	Bảng đặc tả use case đăng bài viết mới	9
Bảng 2.6	Bảng đặc tả use case Chỉnh sửa bài viết	10
Bảng 2.7	Bảng đặc tả use case Xóa bài đăng	11
Bảng 2.8	Bảng đặc tả use case thích bài viết	12
Bảng 2.9	Bảng đặc tả use case bình luận bài viết	13
Bảng 2.10	Bảng đặc tả use case Gửi tin nhắn	14
Bảng 2.11	Bảng đặc tả UseCase gợi ý theo dõi người dùng	14
Bảng 2.12	Bảng đặc tả use case tìm kiếm thông tin	15
Bảng 4.1	Thông tin thiết kế giao diện	45
Bảng 4.2	Đặc tả lớp User	49
Bảng 4.3	Đặc tả lớp Post	50
Bảng 4.4	Đặc tả lớp Comment	51
Bảng 4.5	Đặc tả lớp Category	52
Bảng 4.6	Các trường trong các collection của cơ sở dữ liệu	54
Bảng 4.7	Danh sách thư viện và công cụ sử dụng	56
Bảng 4.8	Thông tin project	56
Bảng 4.9	Các trường hợp kiểm thử cho trang web mạng xã hội chia sẻ báo cáo khoa học	60
Bảng 4.10	Cấu hình tối thiểu để truy cập trang web	61

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
DATN	Đồ án tốt nghiệp
API	Giao diện lập trình ứng dụng (Application Programming Interface)
CNTT	Công nghệ thông tin
EUD	Phát triển ứng dụng người dùng cuối (End-User Development)
Framework	Khung phần mềm, cung cấp chức năng chung, có thể được thay đổi có chọn lọc bằng mã bổ sung do người dùng viết
GWT	Công cụ lập trình Javascript bằng Java của Google (Google Web Toolkit)
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
IaaS	Dịch vụ hạ tầng (Infrastructure as a Service)
Platform	Nền tảng
User	Người dùng

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong thời đại công nghệ số hiện nay, mạng xã hội đã trở thành một phần không thể thiếu trong cuộc sống của con người. Tuy nhiên, việc sử dụng mạng xã hội vẫn chưa được tối ưu hoá để phục vụ cho các nhu cầu chuyên môn của các nhà khoa học. Do đó, việc xây dựng một mạng xã hội dành riêng cho người nghiên cứu khoa học chia sẻ báo cáo nghiên cứu khoa học sẽ là một bước tiến lớn đối với lĩnh vực nghiên cứu khoa học.

Mạng xã hội là một công cụ mạnh mẽ giúp cho các nhà nghiên cứu khoa học trao đổi thông tin và chia sẻ các nghiên cứu của mình. Tuy nhiên, hiện nay trên thị trường vẫn chưa có một mạng xã hội chuyên dành cho các nhà nghiên cứu khoa học. Việc xây dựng một mạng xã hội như vậy sẽ mang lại nhiều lợi ích cho cộng đồng nghiên cứu khoa học.

Đề tài này nhằm mục đích xây dựng một website mạng xã hội cho các nhà nghiên cứu chia sẻ và tìm kiếm các báo cáo khoa học. Đây là một nhu cầu cấp thiết của cộng đồng nhà nghiên cứu để tạo ra một nền tảng đơn giản và dễ sử dụng cho việc chia sẻ thông tin khoa học và cải thiện quá trình tìm kiếm các báo cáo khoa học cho người dùng.

1.2 Mục tiêu và phạm vi đề tài

Trong phần này tôi sẽ nêu ra các hạn chế của mạng xã hội hiện nay đối với việc chia sẻ tài nguyên báo cáo nghiên cứu:

Mạng xã hội cho các nhà nghiên cứu khoa học sẽ tạo ra một môi trường chia sẻ thông tin và tài liệu nghiên cứu. Các nhà nghiên cứu có thể đăng tải các bài báo, bài viết, tài liệu nghiên cứu của mình lên mạng xã hội và chia sẻ với các thành viên khác trong cộng đồng. Nhờ đó, các nghiên cứu sẽ được quảng bá rộng rãi hơn và thu hút sự quan tâm từ nhiều nhà nghiên cứu khác.

Mạng xã hội cho các nhà nghiên cứu khoa học sẽ cung cấp một nền tảng để trao đổi ý kiến và thảo luận về các nghiên cứu. Các thành viên trong cộng đồng có thể đăng tải các câu hỏi, ý kiến và nhận được phản hồi từ các nhà nghiên cứu khác. Điều này sẽ giúp các nhà nghiên cứu trao đổi thông tin với nhau một cách nhanh chóng và hiệu quả hơn.

Mạng xã hội cho các nhà nghiên cứu khoa học còn là một công cụ quản lý tài liệu nghiên cứu hiệu quả. Nhờ đó, các nhà nghiên cứu có thể quản lý và lưu trữ tài liệu nghiên cứu của mình một cách tiện lợi và dễ dàng. Họ có thể truy cập vào các

tài liệu này bất cứ lúc nào và ở bất kỳ đâu trên thế giới.

1.3 Định hướng giải pháp

Từ việc xác định rõ nhiệm vụ cần giải quyết ở phần 1.2, đề xuất giải pháp cho vấn đề trên:

Định hướng giải quyết vấn đề: Sử dụng công nghệ web development để xây dựng một website mạng xã hội cho các nhà nghiên cứu chia sẻ và tìm kiếm các báo cáo khoa học.

Giải pháp: Sử dụng các công nghệ web development phổ biến như HTML, CSS, JavaScript(ReactJS) để thiết kế giao diện và tương tác người dùng trên trình duyệt web. Sử dụng ngôn ngữ lập trình phía máy chủ như Javascript(NodeJS/Express) để xử lý các yêu cầu từ phía người dùng và lưu trữ các thông tin người dùng, báo cáo khoa học. Sử dụng hệ quản trị cơ sở dữ liệu MongoDB để lưu trữ và truy xuất thông tin.

Đóng góp chính và kết quả đạt được: Đề án này sẽ giúp đáp ứng nhu cầu chia sẻ và tìm kiếm các báo cáo khoa học của cộng đồng nhà nghiên cứu. Nó sẽ tạo ra một nền tảng mạng xã hội đơn giản và dễ sử dụng cho các nhà nghiên cứu chia sẻ thông tin khoa học của họ. Đồng thời, nó sẽ giúp cải thiện quá trình tìm kiếm các báo cáo khoa học cho người dùng bằng cách cung cấp các công cụ tìm kiếm tiên tiến và các tính năng tương tác như phản hồi và đánh giá.

Mạng xã hội dành cho người nghiên cứu khoa học sẽ giúp tạo ra một cộng đồng nghiên cứu khoa học chuyên nghiệp và có nhu cầu trao đổi, hỗ trợ và tạo ra sự đóng góp tích cực cho lĩnh vực nghiên cứu. Điều này giúp cho việc trao đổi thông tin chuyên môn, đóng góp tích cực cho cộng đồng nghiên cứu và tăng khả năng phát triển của lĩnh vực nghiên cứu. Ngoài ra, mạng xã hội này cũng giúp các nhà khoa học kết nối và tìm kiếm những đồng nghiệp có cùng lĩnh vực quan tâm, tạo ra sự đa dạng và phong phú hơn trong quá trình nghiên cứu khoa học.

Qua đó, chúng ta có thể thấy được những lợi ích mà một mạng xã hội mang lại cho con người. Chuyên sâu hơn, chúng ta sẽ phát triển một mạng xã hội cho các nhà nghiên cứu khoa học có thể dễ dàng chia sẻ và trao đổi nghiên cứu với nhau.

1.4 Bố cục đề án

Phần còn lại của báo cáo đề án tốt nghiệp này được tổ chức như sau.

Chương 2 có nhiệm vụ giới thiệu và so sánh các trang hiện có từ đó chọn ra các chức năng chính và được người dùng sử dụng nhiều nhất và các phần được chọn ra từ khảo sát yêu cầu người dùng và sẽ phân tích chúng theo các biểu đồ của UML.

Chương 3 trình bày về những công nghệ được sử dụng trong việc phát triển đồ án gồm có Javascript(ReactJS, NodeJS, MongoDB), khái quát, giới thiệu về công nghệ cũng như lý do lựa chọn công nghệ đó, giải thích một số thuật ngữ.

Chương 4 phân tích thiết kế chi tiết cho dự án, trình bày về kiến trúc được sử dụng, thiết kế giao diện. Sau đó là kết quả đã đạt được cùng với những ý kiến đóng góp của người dùng.

Chương 5 trình bày những khó khăn đã gặp phải, giải pháp và đóng góp nổi bật của đồ án.

Chương cuối cùng, kết quả đạt được sau quá trình nghiên cứu và xây dựng, cùng với đó là những gì cần được cải thiện và định hướng phát triển tiếp trong tương lai.

Sau đây là chương 2, trình bày về thiết kế của trang web.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Chương 2 này sẽ có nhiệm vụ giới thiệu và so sánh các sản phẩm hiện có để phát triển, tiếp theo trình bày về những chức năng chính mà người dùng có thể thực hiện được với trang web và phân tích chúng theo các biểu đồ của UML.

2.1 Khảo sát hiện trạng

2.1.1 Các trang mạng xã hội hiện có

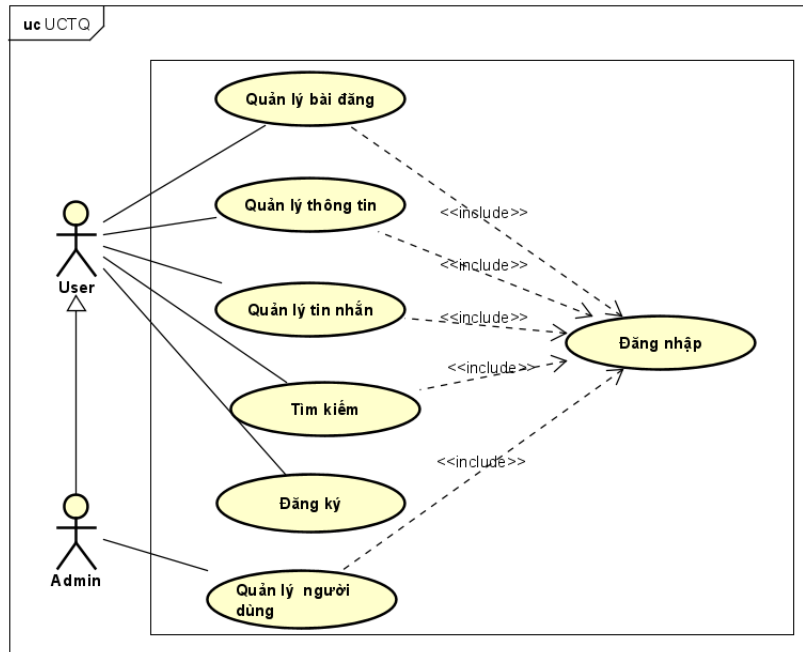
Hiện nay có rất nhiều loại mạng xã hội khác nhau, tuy nhiên tôi sẽ liệt kê và phân loại chúng theo nhóm chính như sau:

1. Mạng xã hội chia sẻ hình ảnh: Chúng là những mạng xã hội nơi người dùng có thể chia sẻ hình ảnh và video như Instagram, Snapchat, TikTok, Pinterest.
2. Mạng xã hội tương tác: Các mạng xã hội này cung cấp nền tảng cho người dùng để trao đổi thông tin và tương tác với nhau. Các ví dụ bao gồm Facebook, Twitter, LinkedIn, Reddit.
3. Mạng xã hội dành cho người yêu thích nội dung đa phương tiện: Các mạng xã hội này tập trung vào việc chia sẻ và truy cập vào nội dung đa phương tiện như YouTube, Vimeo, SoundCloud.
4. Mạng xã hội chia sẻ tài liệu: Các mạng xã hội này cung cấp cho người dùng chia sẻ và truy cập vào các tài liệu như Google Drive, Dropbox, OneDrive.

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quát

Từ các trang mạng xã hội trên, hình 2.1 dưới đây là use case tổng quan biểu diễn các chức năng chính của hệ thống:



Hình 2.1: Biểu đồ Use Case tổng quát.

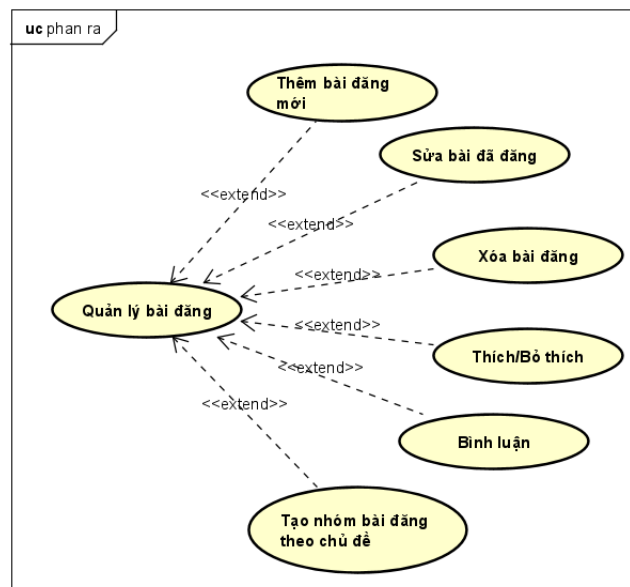
Hình 2.1 có tác nhân User vào vai trò là người dùng hệ thống, người mà vào hệ thống và có thể thực hiện được các chức năng ở trên như: quản lý bài đăng, quản lý thông tin cá nhân, quản lý tin nhắn, tìm kiếm thông tin(người dùng, tài liệu). Bảng 2.1 mô tả chức năng trong use case tổng quát:

Tên chức năng	Mô tả chức năng
Quản lý bài đăng	Người dùng hệ thống có thể đăng bài lên trên nền tảng mạng xã hội và sau khi thêm có thể chỉnh sửa hoặc có thể xóa bài đã đăng. Gom nhóm các bài đăng thành một bộ sưu tập các chủ đề mà người dùng mong muốn
Quản lý thông tin cá nhân	Người dùng có thể chỉnh sửa thông tin cá nhân của mình trong trang cá nhân của chính mình(thông tin cá nhân cơ bản, thông tin về tổ chức, kĩ năng mà người dùng có)
Quản lý tin nhắn	Người dùng có thể nhắn tin cho người dùng khác trên nền tảng mạng xã hội này,

Bảng 2.1: Mô tả chức năng trong UseCase tổng quát

2.2.2 Biểu đồ use case phân rã quản lý bài đăng

Hình 2.2 dưới đây là hình vẽ biểu diễn các chức năng thuộc chức năng quản lý bài đăng. Mô tả chi tiết được thể hiện trong bảng 2.4



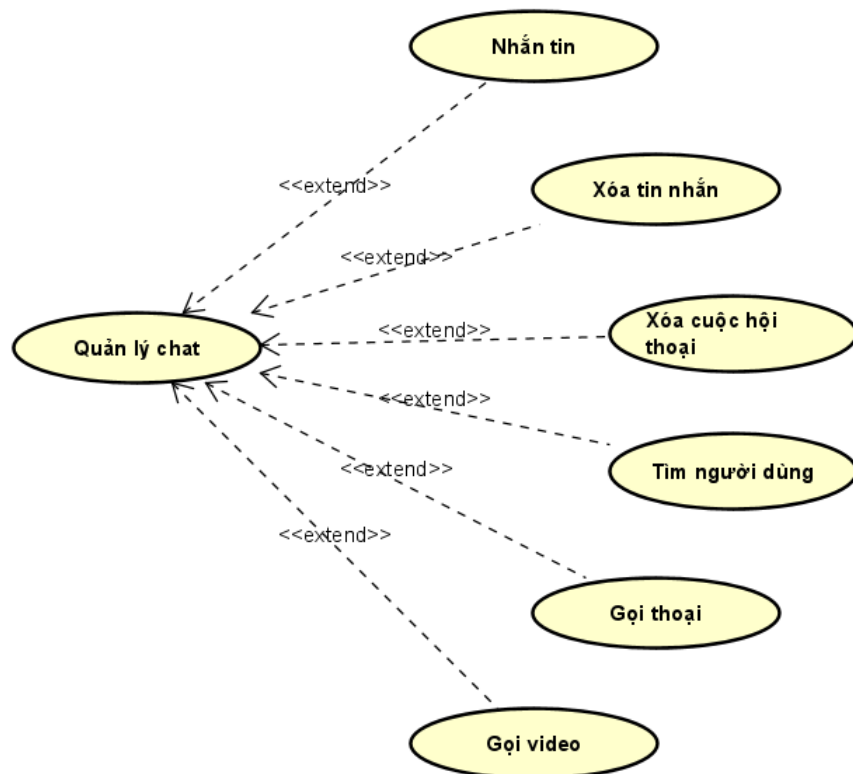
Hình 2.2: Biểu đồ use case phân rã quản lý bài đăng

Tên chức năng	Mô tả chức năng
Đăng bài mới	Khi người dùng muốn đăng bài mới, trang web sẽ hiển thị các trường thông tin cần nhập như nội dung(content), file(ảnh, video, pdf ...)
Chỉnh sửa bài đã đăng	Khi người dùng muốn chỉnh sửa lại bài đã đăng trước đó, người dùng cần phải vào mục chỉnh sửa bài đăng ở phần mở rộng của bài đăng, chỉnh sửa xong cần lưu lại mục đã chỉnh sửa
Xóa bài đã đăng	Khi người dùng muốn xóa một bài đã được đăng lên, người dùng cần chọn chức năng xóa bài viết trong menu của bài đăng đó
Thích/ Bỏ thích bài đăng	Người dùng có thể thể hiện tương tác giữa người dùng với bài đăng bằng cách thả cảm xúc(thích) bài đăng đó
Gom nhóm bài đăng theo chủ đề	Khi người dùng muốn gom nhóm các bài có cùng chủ đề đã đăng trên nền tảng, chọn các bài đăng muốn thêm vào chủ đề chỉ định

Bảng 2.2: Mô tả chức năng trong use case phân rã quản lý bài đăng

2.2.3 Biểu đồ use case phân rã tìm kiếm thông tin

Hình ?? dưới đây là hình vẽ biểu diễn các chức năng thuộc chức năng tìm kiếm thông tin. Mô tả chi tiết được thể hiện trong bảng 2.3



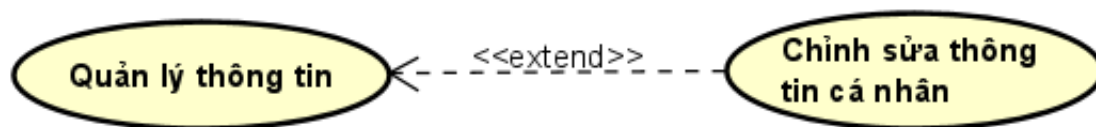
Hình 2.3: Biểu đồ use case phân rã tìm kiếm thông tin

Tên chức năng	Mô tả chức năng
Tìm kiếm thông tin người dùng	Khi người dùng muốn tìm kiếm thông tin về một nhà nghiên cứu thông qua trang web, chỉ cần gõ tên vào mục search phần header và các kết quả trả về thông tin đại diện cho người dùng đó
Tìm kiếm thông tin bài đăng thông qua từ khóa	Khi người dùng muốn tìm kiếm thông tin các bài đăng mà mình quan tâm, gõ từ khóa vào ô search và tìm kiếm trong phần "research" các bài đăng liên quan đến từ khóa sẽ hiển thị trong trang tìm kiếm
Tìm kiếm theo hashtag	Khi người dùng muốn tìm kiếm các bài đăng theo hashtag, click vào hashtag của bài đăng tương tự sẽ ra kết quả các bài đăng có chứa hashtag đó

Bảng 2.3: Mô tả chức năng trong use case phân rã tìm kiếm thông tin

2.2.4 Biểu đồ use case phân rã quản lý thông tin

Sau đây là phần trình bày về biểu đồ use case phân rã quản lý thông tin được biểu diễn bằng hình vẽ 2.4



Hình 2.4: Biểu đồ use case phân rã quản lý thông tin

Bảng 2.4 mô tả chức năng trong use case phân rã quản lý thông tin với hai chức năng chính là xem thông tin người dùng và chỉnh sửa thông tin cá nhân

Tên chức năng	Mô tả chức năng
Xem thông tin người dùng	Khi người dùng muốn xem thông tin của chính mình hoặc muốn xem thông tin của người dùng khác trong hệ thống, trang web hiển thị trang cá nhân chứa thông tin người dùng đó
Chỉnh sửa thông tin cá nhân	Người dùng muốn thay đổi thông tin cá nhân(tên, tổ chức, website, giới thiệu bản thân,...)

Bảng 2.4: Mô tả chức năng trong use case phân rã quản lý thông tin

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case đăng bài viết mới

Bảng 2.5 đặc tả use case đăng bài viết mới trên hệ thống:

Mã UseCase	UC01
Tên UseCase	Đăng bài viết mới
Tác nhân	Người dùng hệ thống
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng hệ thống: Ấn vào thanh status bar 2. Hệ thống: Hiển thị giao diện các trường cần điền: mô tả ngắn gọn, thể loại, ngày công bố, hashtag, các file media(ảnh, video, pdf) 3. Người dùng thêm các trường cần điền 4. Đăng tải bài viết mới bằng cách nhấn "Upload" 5. Người dùng thoát ra khỏi form đăng tải bài viết mới chuyển sang luồng rẽ nhánh A1 6. UseCase kết thúc.
Luồng sự kiện rẽ nhánh	<p>Luồng rẽ nhánh A1:</p> <ol style="list-style-type: none"> 1. Hệ thống: thông báo cho người dùng đăng tải bài viết không thành công 2. Người dùng hệ thống: Ấn "Cancel" hoặc "x" thoát khỏi modal đăng tải bài viết mới <p>Usecase dừng lại</p>
Hậu điều kiện	Đăng tải bài viết mới thành công

Bảng 2.5: Bảng đặc tả use case đăng bài viết mới

2.3.2 Đặc tả use case chỉnh sửa bài đăng

Bảng 2.6 đặc tả use case chỉnh sửa bài viết trên hệ thống:

Mã UseCase	UC02
Tên UseCase	Chỉnh sửa bài đăng
Tác nhân	Người dùng hệ thống
Tiền điều kiện	1. Người dùng đã đăng nhập vào hệ thống 2. Người dùng có quyền chỉnh sửa bài đăng đó 3. Bài đăng cần chỉnh sửa có tồn tại
Luồng sự kiện chính	1. Người dùng hệ thống: Chọn menu của bài đăng cần chỉnh sửa 2. Hệ thống: Hiển thị giao diện các trường cần điền: mô tả ngắn gọn, thể loại, ngày công bố, các file media(ảnh, video, pdf) 3. Người dùng chỉnh sửa các trường cần điền 4. Đăng tải bài viết đã chỉnh sửa bằng cách nhấn "Upload" 5. Người dùng thoát ra khỏi form đăng tải bài viết mới chuyển sang luồng rẽ nhánh A1 6. UseCase kết thúc.
Luồng sự kiện rẽ nhánh	Luồng rẽ nhánh A1: 1. Hệ thống: thông báo cho người dùng cập nhật bài viết không thành công 2. Người dùng hệ thống: Ấn "Cancel" hoặc "x" thoát khỏi phần chỉnh sửa UseCase dừng lại
Hậu điều kiện	Chỉnh sửa bài đăng thành công Hiển thị bài viết với các trường đã được chỉnh sửa

Bảng 2.6: Bảng đặc tả use case Chỉnh sửa bài viết

2.3.3 Đặc tả use case xóa bài viết

Đặc tả use case xóa bài viết được thể hiện ở bảng 2.7

Mã UseCase	UC03
Tên UseCase	Xóa bài viết
Tác nhân	Người dùng hệ thống
Tiền điều kiện	1. Người dùng đã đăng nhập vào hệ thống 2. Người dùng có quyền xóa bài đăng đó 3. Bài đăng cần xóa có tồn tại
Luồng sự kiện chính	1. Người dùng hệ thống: Chọn menu của bài đăng muốn xóa 2. Hệ thống: Hiển thị thông báo xóa bài viết 3. Người dùng thoát ra khỏi thông báo xóa bài viết chuyển sang luồng rẽ nhánh A1 4. UseCase kết thúc.
Luồng sự kiện rẽ nhánh	Luồng rẽ nhánh A1: 1. Hệ thống: thông báo cho người dùng xóa bài viết không thành công 2. Người dùng hệ thống: Ấn "Cancel" hoặc "x" thoát khỏi thông báo xóa bài UseCase dừng lại
Hậu điều kiện	Xóa bài đăng thành công

Bảng 2.7: Bảng đặc tả use case Xóa bài đăng

2.3.4 Đặc tả use case thích bài viết

Bảng 2.8 đặc tả use case thích bài viết:

Mã UseCase	UC04
Tên UseCase	Thích bài viết
Tác nhân	Người dùng hệ thống
Tiền điều kiện	1. Người dùng đã đăng nhập vào hệ thống 2. Bài đăng có tồn tại
Luồng sự kiện chính	1. Người dùng hệ thống: Chọn nút "thích" của bài đăng 2. Hệ thống: Chuyển đổi trạng thái "thích" của bài đăng khi hiển thị với người dùng 3. Người dùng đã thích bài viết trước đó, chuyển sang luồng rẽ nhánh A1 4. UseCase kết thúc.
Luồng sự kiện rẽ nhánh	Luồng rẽ nhánh A1: 1. Hệ thống: thông báo cho người dùng cập nhật lại số lượng "thích" bài viết 2. Người dùng hệ thống: Ấn "thích" để xóa lượt "thích" của mình khỏi bài viết Usecase dừng lại
Hậu điều kiện	Thích bài đăng thành công Dữ liệu bài viết cập nhật lại số lượng "thích" bài

Bảng 2.8: Bảng đặc tả use case thích bài viết

2.3.5 Đặc tả use case bình luận bài viết

Bảng 2.9 đặc tả use case bình luận tại một bài đăng trên hệ thống như dưới đây:

Mã UseCase	UC05
Tên UseCase	Bình luận bài viết
Tác nhân	Người dùng hệ thống
Tiền điều kiện	1. Người dùng đã đăng nhập vào hệ thống 2. Bài đăng có tồn tại
Luồng sự kiện chính	1. Người dùng hệ thống: Nhập bình luận vào phần "Comment" của bài đăng 2. Sau khi nhập xong ấn "Gửi" hoặc "Enter" để đăng bình luận về bài viết đó 3. Người dùng trước khi đăng đã mất kết nối mạng chuyển sang luồng rẽ nhánh A1 4. UseCase kết thúc.
Luồng sự kiện rẽ nhánh	Luồng rẽ nhánh A1: 1. Hệ thống: Thông báo đăng bình luận không thành công 2. Người dùng hệ thống: Đăng lại bình luận bằng cách thực hiện lại luồng chính Usecase dừng lại
Hậu điều kiện	Đăng bình luận bài viết thành công Dữ liệu bài viết cập nhật lại số lượng bình luận của bài và hiển thị danh sách các bình luận

Bảng 2.9: Bảng đặc tả use case bình luận bài viết

2.3.6 Đặc tả use case gửi tin nhắn

Tiếp theo là bảng 2.10 đặc tả use case gửi tin nhắn cho người dùng khác trong hệ thống

Mã UseCase	UC06
Tên UseCase	Gửi tin nhắn
Tác nhân	Người dùng hệ thống
Tiền điều kiện	1. Người dùng đã đăng nhập vào hệ thống 2. Hệ thống hiển thị chi tiết cuộc trò chuyện
Luồng sự kiện chính	1. Người dùng hệ thống: nhập nội dung tin nhắn và gửi 2. Hệ thống: xác nhận gửi tin nhắn thành công 3. Hệ thống cập nhật lại tin nhắn mới trên giao diện nhắn tin 4. UseCase kết thúc.
Luồng sự kiện rẽ nhánh	Luồng rẽ nhánh A1: Hệ thống: thông báo cho người dùng gửi tin nhắn không thành công Usecase dừng lại
Hậu điều kiện	Gửi tin nhắn thành công Hiển thị tin nhắn mới được cập nhật

Bảng 2.10: Bảng đặc tả use case Gửi tin nhắn

2.3.7 Đặc tả use case gợi ý theo dõi người dùng

Mã UseCase	UC07
Tên UseCase	Gợi ý theo dõi người dùng
Tác nhân	Người dùng hệ thống
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống
Luồng sự kiện chính	1. Hệ thống lấy ra danh sách những người dùng có điểm chung để lấy ra và đưa vào danh sách gợi ý theo dõi cho người dùng(cùng tổ chức) 2. Hệ thống: hiển thị danh sách gợi ý theo dõi 3. Người dùng bị mất kết nối mạng chuyển sang luồng rẽ nhánh A1 4. UseCase kết thúc.
Luồng sự kiện rẽ nhánh	Luồng rẽ nhánh A1: Hệ thống: không hiển thị danh sách gợi ý Usecase dừng lại
Hậu điều kiện	Lấy ra danh sách gợi ý theo dõi Hiển thị danh sách người dùng mới khi người dùng tải lại

Bảng 2.11: Bảng đặc tả UseCase gợi ý theo dõi người dùng

Bảng 2.11 đặc tả use case gợi ý theo dõi người dùng dựa trên danh sách người dùng có chung các thông tin như tổ chức và kỹ năng được công khai trên trang cá nhân hiện tại

2.3.8 Đặc tả use case tìm kiếm thông tin

Bảng 2.12 đặc tả use case tìm kiếm thông tin các bài báo cáo hoặc người dùng có trong hệ thống. Use case được mô tả như sau:

Mã UseCase	UC08
Tên UseCase	Tìm kiếm thông tin
Tác nhân	Người dùng hệ thống
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng hệ thống: nhập từ khóa cần tìm và ấn "submit" 2. Hệ thống: hiển thị danh sách các danh mục tìm kiếm dựa theo từ khóa mà người dùng cung cấp 3. Hệ thống không tìm ra được các bài viết hoặc người dùng liên quan chuyển sang luồng A1 3. UseCase kết thúc.
Luồng sự kiện rẽ nhánh	Luồng rẽ nhánh A1: Hệ thống không lấy ra được các từ khóa đó, hiển thị thông báo cho người dùng tìm kiếm không thành công Usecase dừng lại
Hậu điều kiện	Tìm kiếm các bài viết có chứa từ khóa liên qua Hiển thị danh sách các bài đăng hoặc người dùng với từ khóa đó

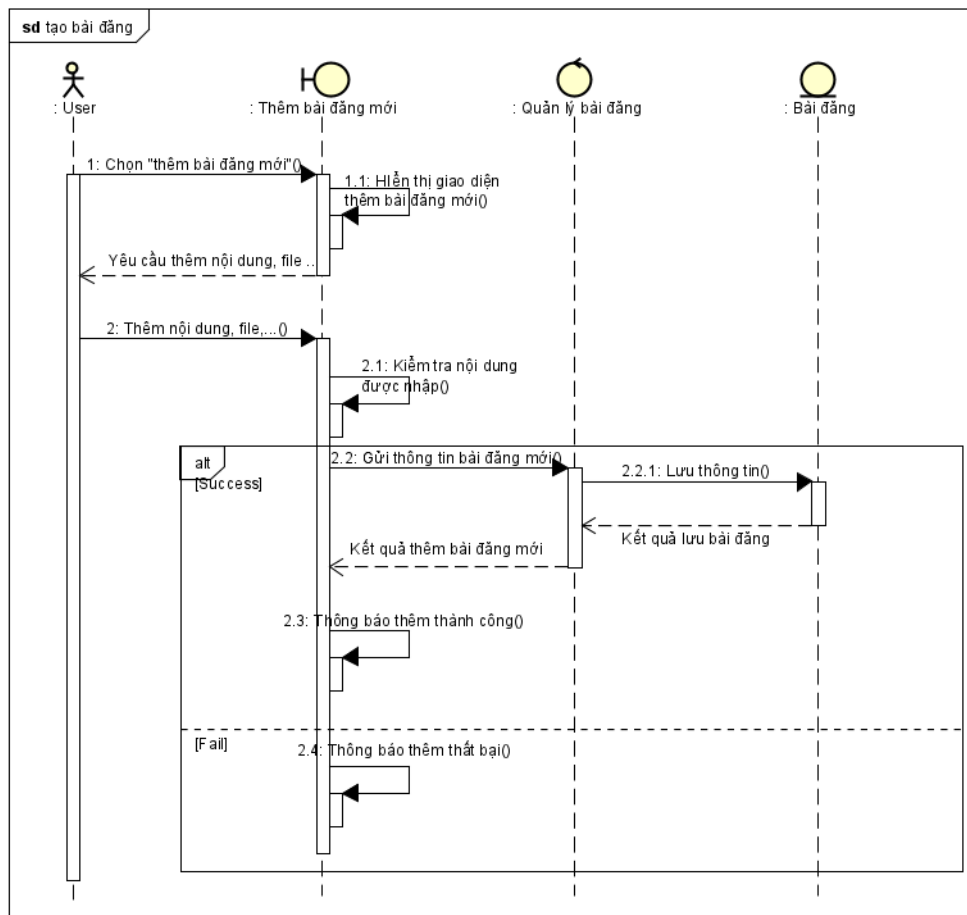
Bảng 2.12: Bảng đặc tả use case tìm kiếm thông tin

2.4 Biểu đồ Trình tự

Từ đặc tả ca sử dụng ở phần 2.3 sau đây là các biểu đồ trình tự biểu diễn luồng sự kiện cho từng chức năng.

2.4.1 Biểu đồ trình tự tạo bài đăng mới

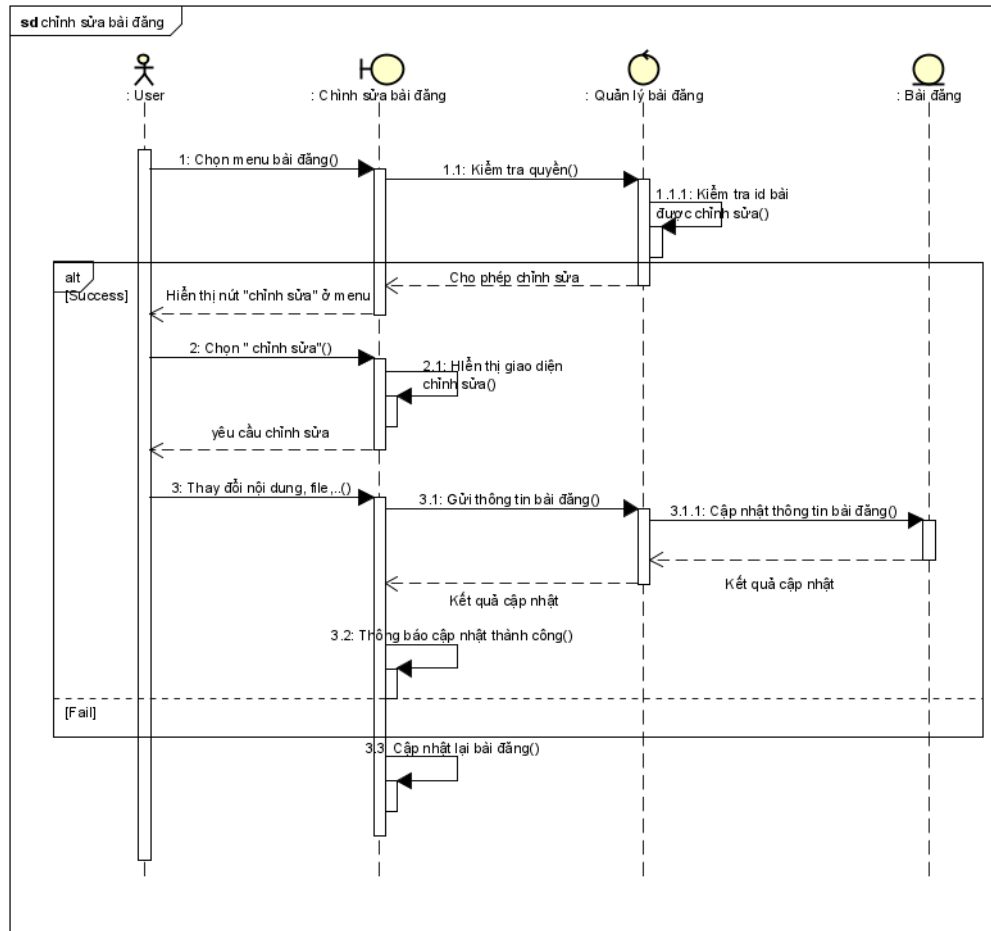
Hình 2.5 biểu diễn luồng sự kiện chính khi thực hiện thao tác đăng bài viết mới trên hệ thống.



Hình 2.5: Biểu đồ trình tự tạo bài đăng mới

2.4.2 Biểu đồ trình tự chỉnh sửa bài đăng

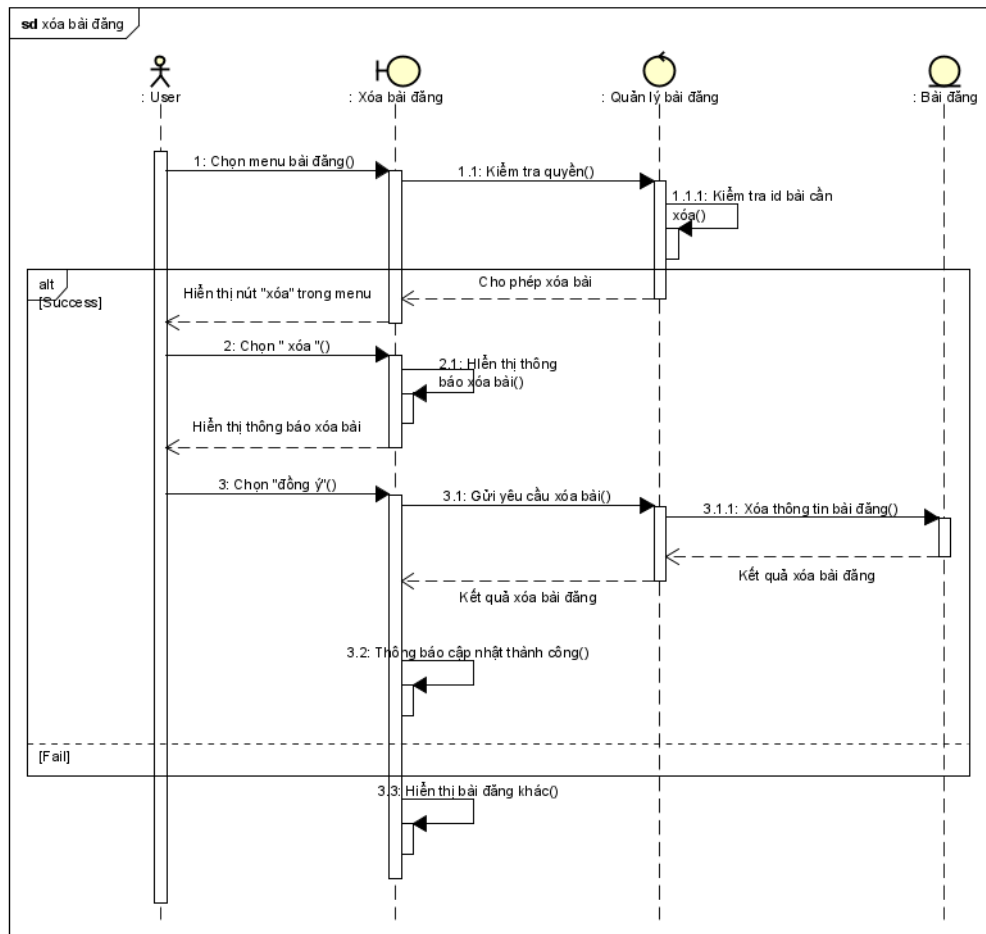
Biểu đồ trình tự chỉnh sửa bài đăng được biểu diễn như hình 2.6:



Hình 2.6: Biểu đồ trình tự chỉnh sửa bài đăng

2.4.3 Biểu đồ trình tự xóa bài đăng

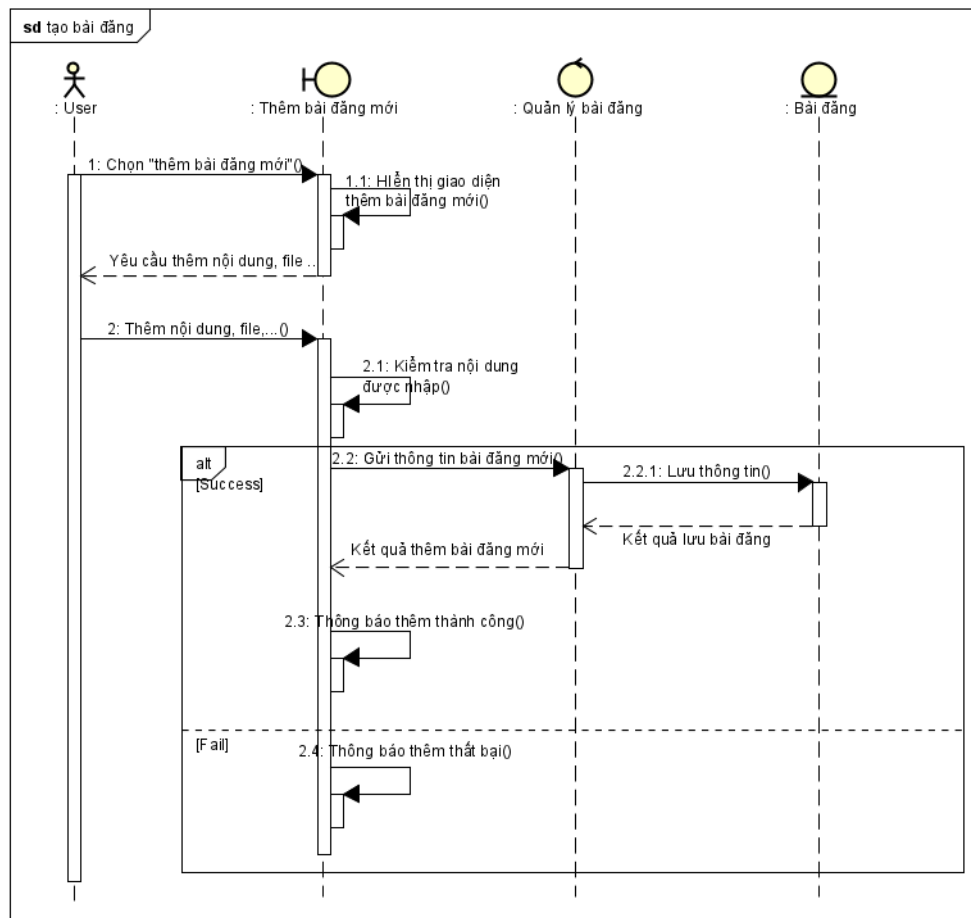
Chức năng xóa bài đăng đã được tạo ra trên hệ thống tuân theo luồng sự kiện như hình 2.7:



Hình 2.7: Biểu đồ trình tự xóa bài đăng

2.4.4 Biểu đồ trình tự xem bài đăng

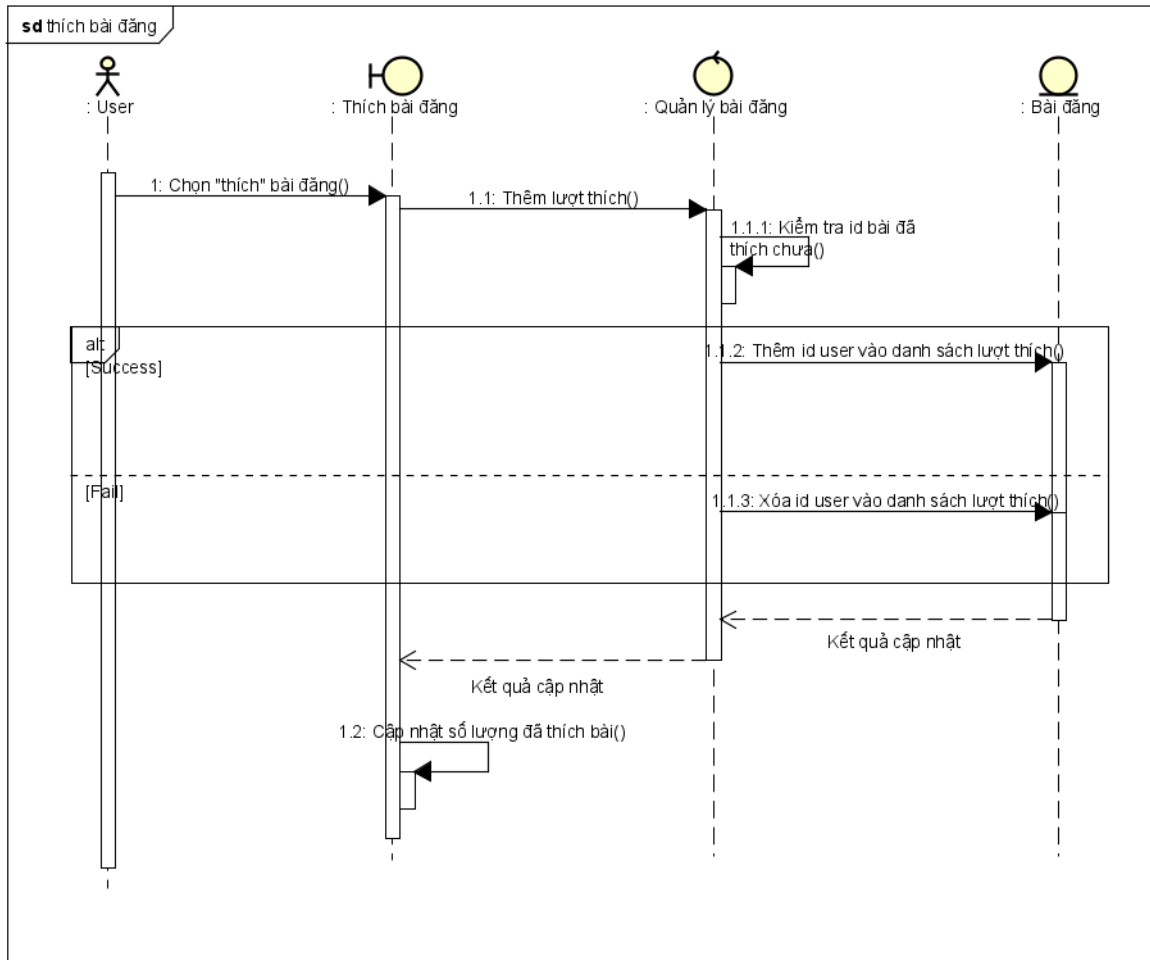
Các bước xem các bài đăng được biểu diễn như hình 2.8



Hình 2.8: Biểu đồ trình tự xem bài đăng

2.4.5 Biểu đồ trình tự thích/ bỏ thích bài đăng

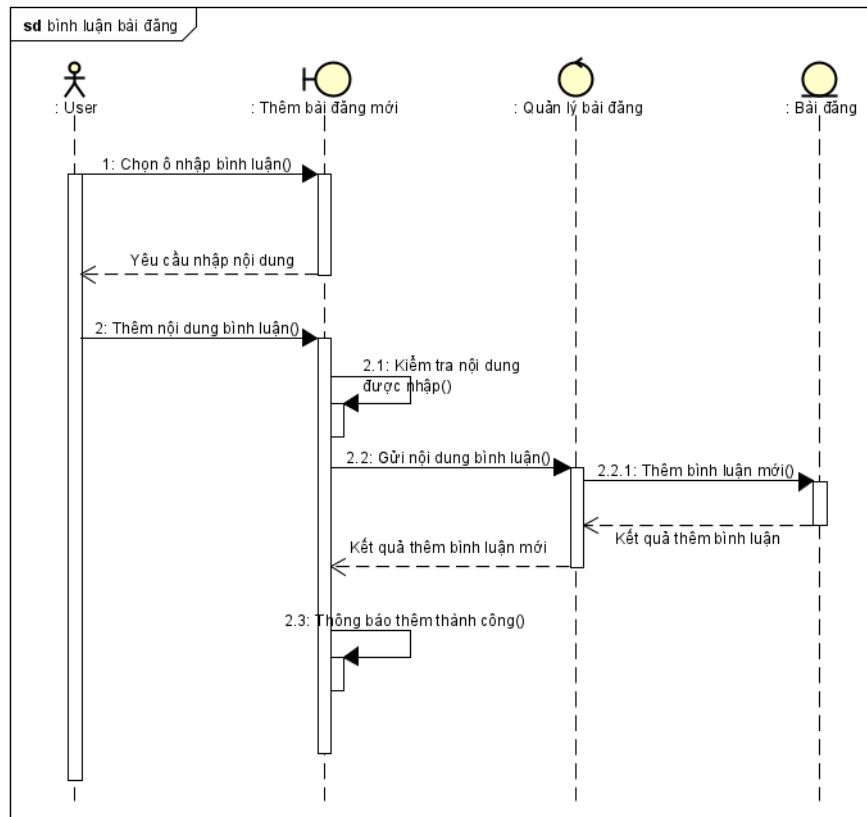
Hình 2.9 là biểu đồ trình tự thích/ bỏ thích bài đăng



Hình 2.9: Biểu đồ trình tự thích/ bỏ thích bài đăng

2.4.6 Biểu đồ trình tự bình luận bài đăng

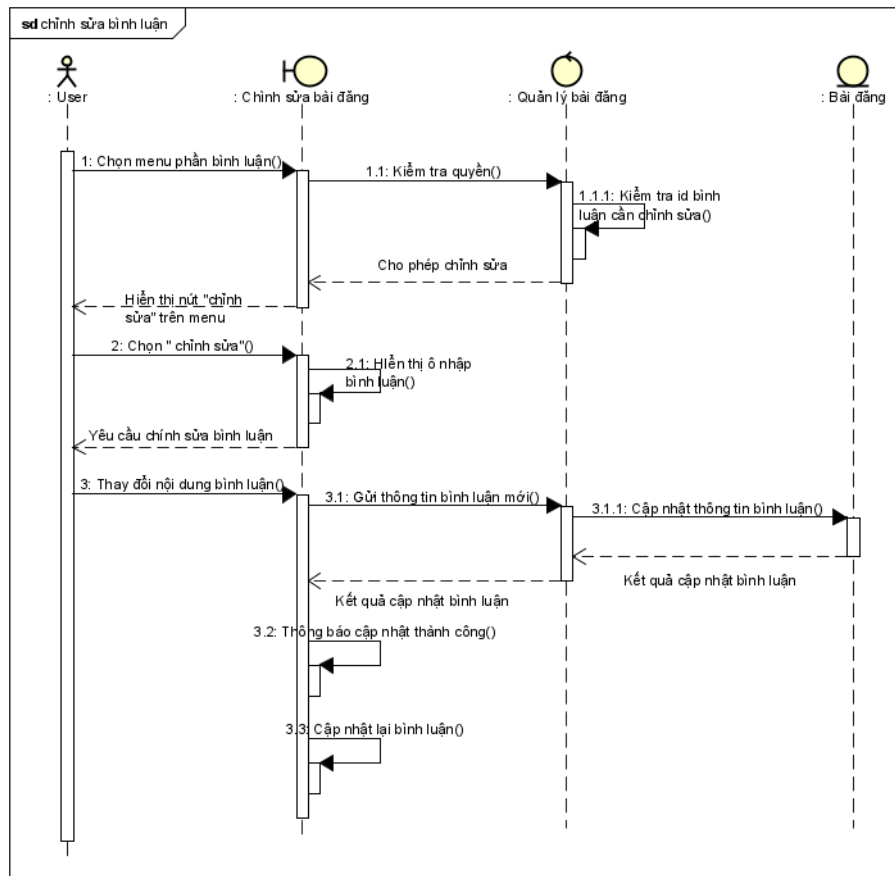
Bình luận bài đăng như đặc tả ở phần trước đã được mô tả dưới dạng biểu đồ trình tự như hình 2.10



Hình 2.10: Biểu đồ trình tự bình luận bài đăng

2.4.7 Biểu đồ trình tự chỉnh sửa bình luận

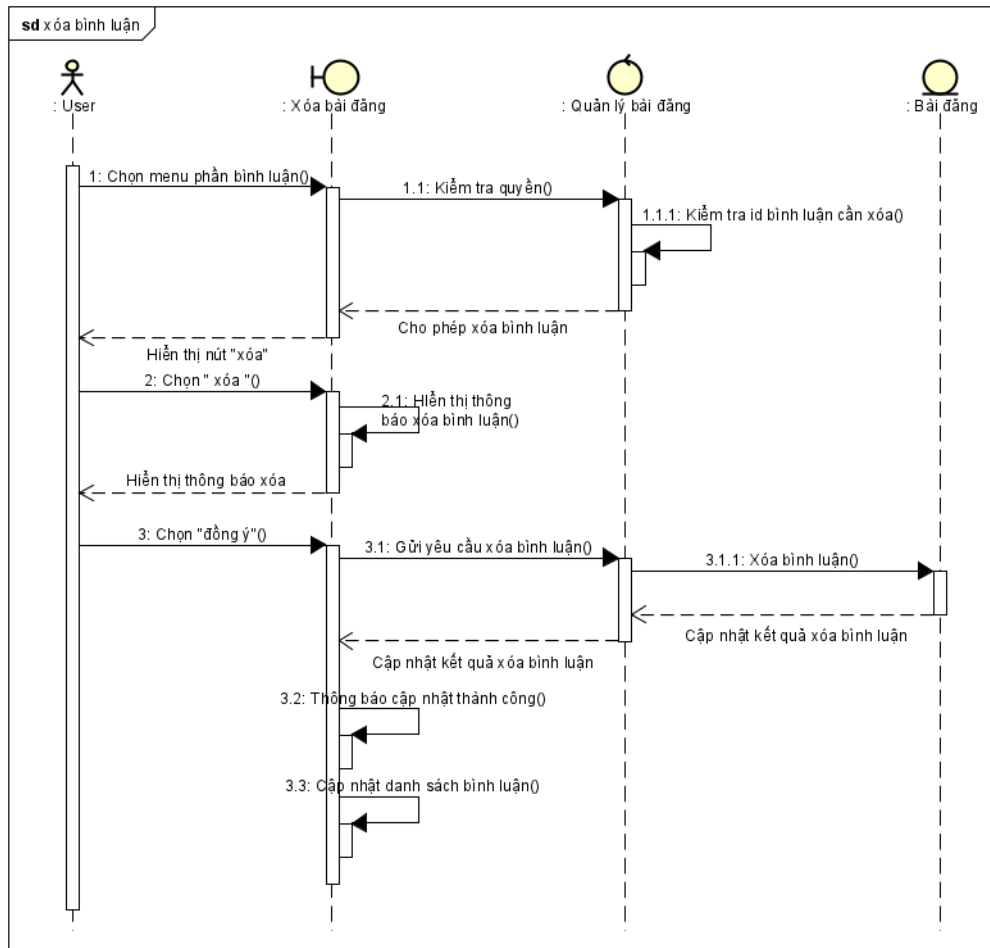
Chỉnh sửa bình luận được biểu diễn dưới dạng biểu đồ trình tự như hình 2.11



Hình 2.11: Biểu đồ trình tự chỉnh sửa bình luận

2.4.8 Biểu đồ trình tự xóa bình luận

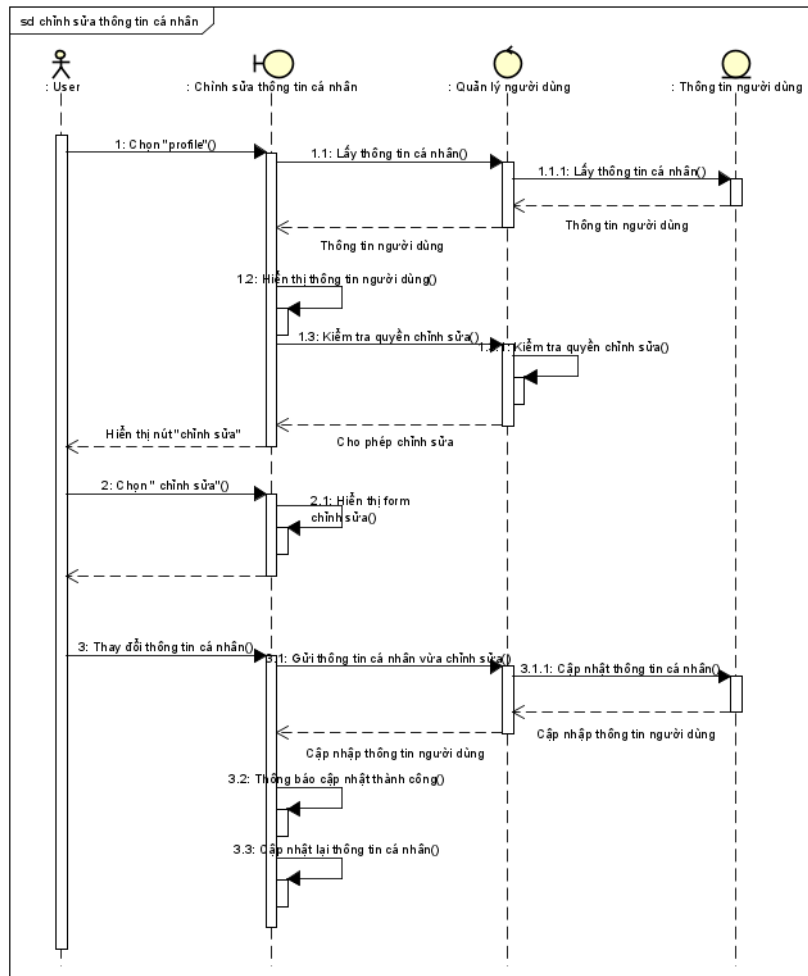
Hình 2.12 là biểu đồ trình tự xóa bình luận được hiểu diễn như sau;



Hình 2.12: Biểu đồ trình tự xóa bình luận

2.4.9 Biểu đồ trình tự chỉnh sửa thông tin người dùng

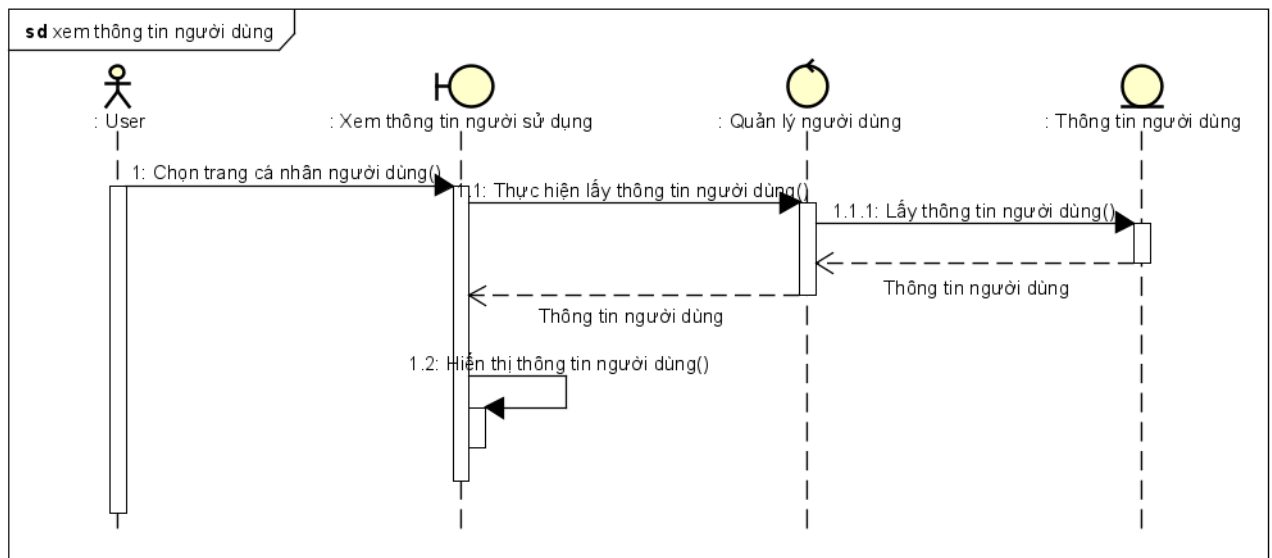
Biểu đồ trình tự chỉnh sửa thông tin người dùng mô tả quá trình chỉnh sửa thông tin của chính người dùng đó như hình 2.13



Hình 2.13: Biểu đồ trình tự chỉnh sửa thông tin người dùng

2.4.10 Biểu đồ trình tự xem thông tin người dùng

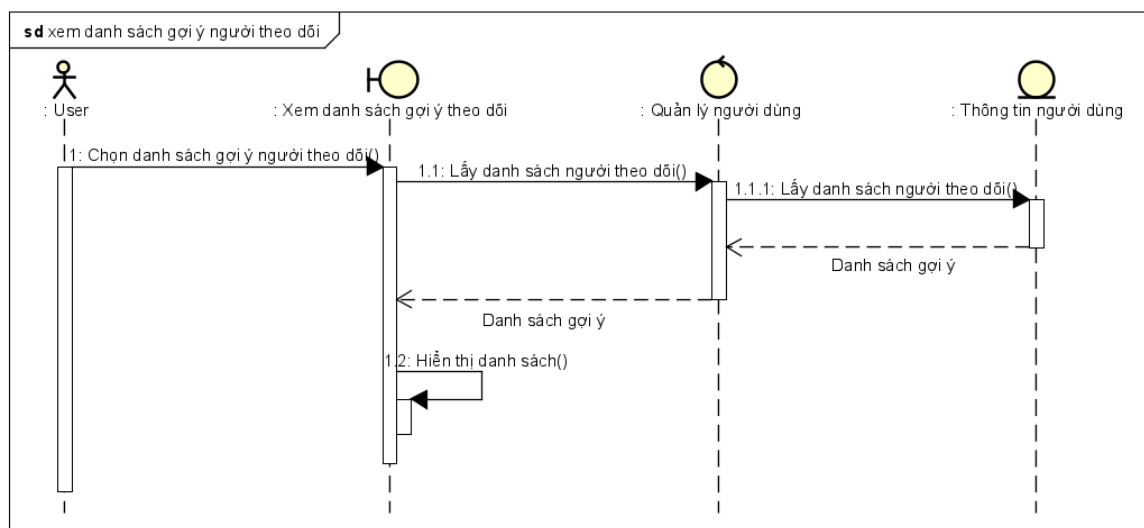
Sau đây là biểu đồ trình tự xem thông tin người dùng được biểu diễn như hình 2.14



Hình 2.14: Biểu đồ trình tự xem thông tin người dùng

2.4.11 Biểu đồ trình tự gợi ý người dùng

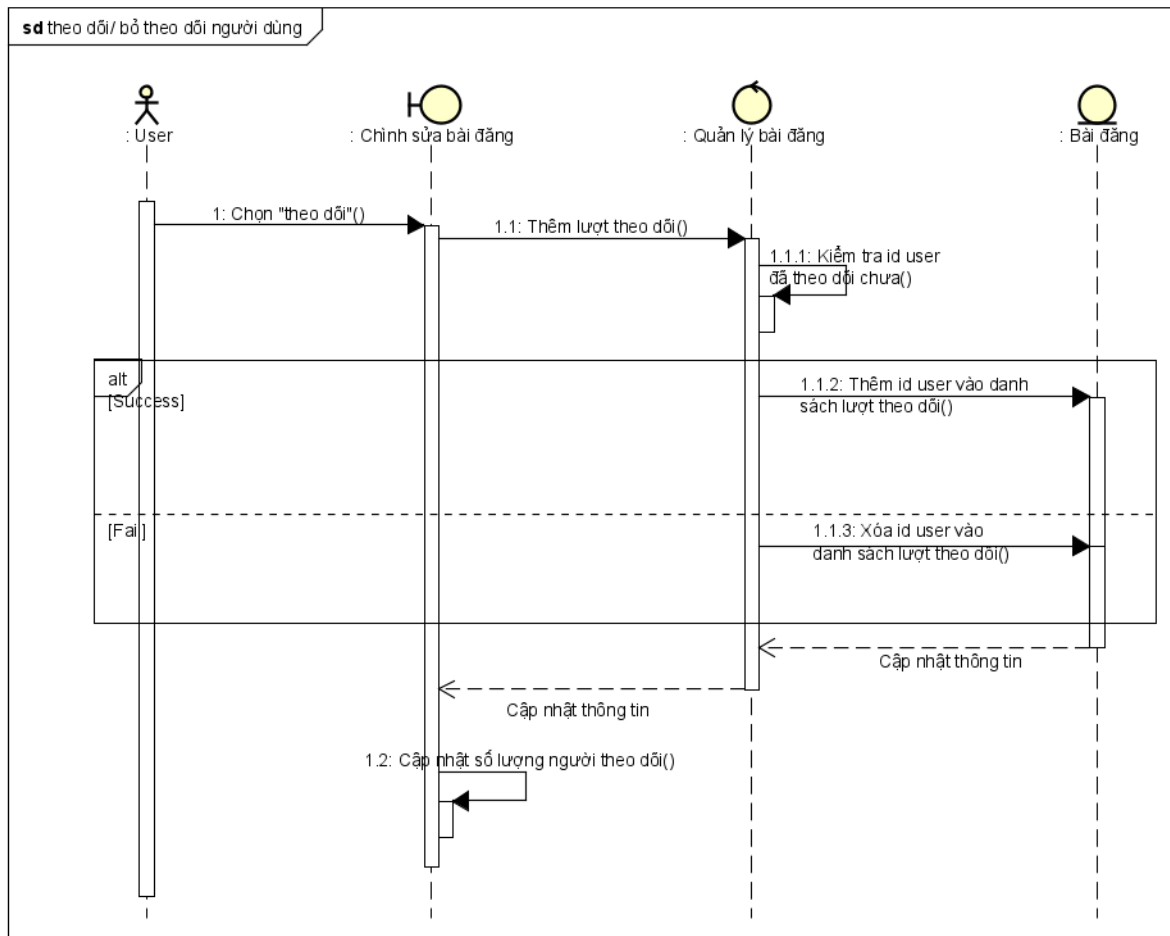
Hình 2.15 biểu diễn biểu đồ trình tự gợi ý người dùng



Hình 2.15: Biểu đồ trình tự gợi ý người dùng

2.4.12 Biểu đồ trình tự theo dõi người dùng

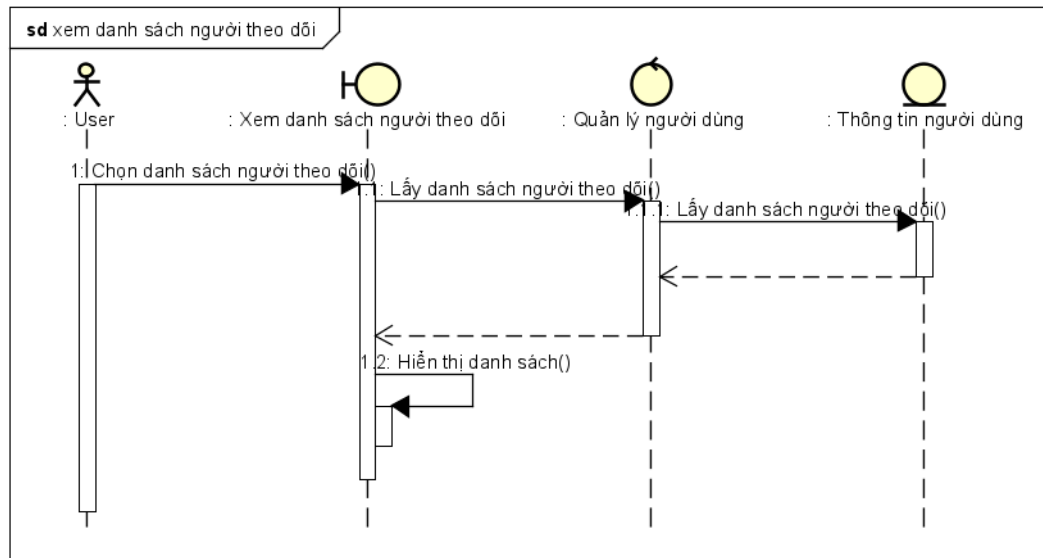
Hình 2.16 biểu diễn biểu đồ trình tự theo dõi người dùng khác trên hệ thống



Hình 2.16: Biểu đồ trình tự theo dõi người dùng

2.4.13 Biểu đồ trình tự xem danh sách người theo dõi

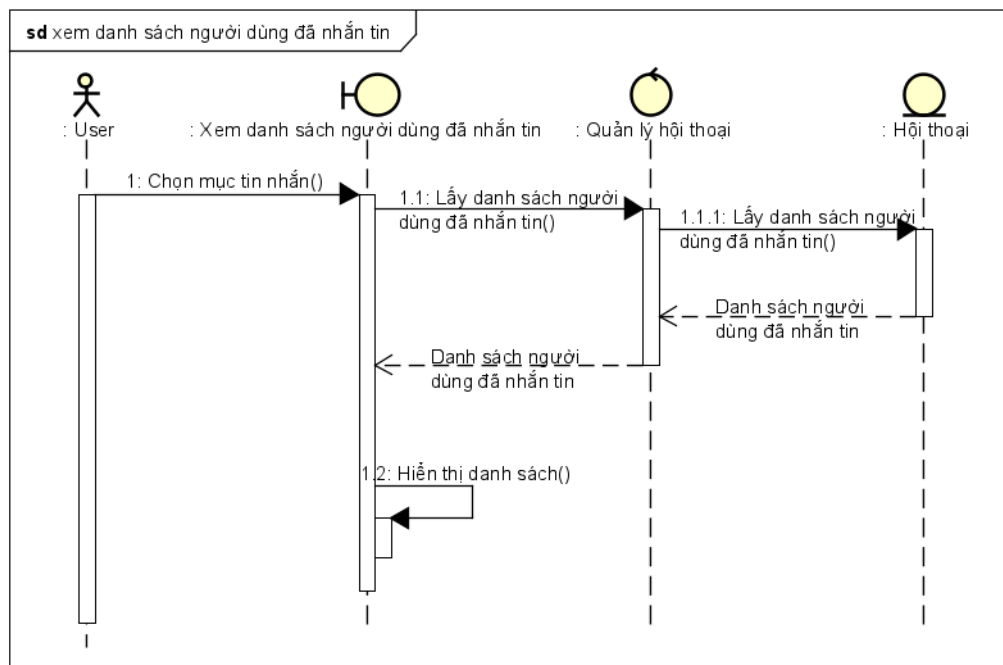
Hình 2.17 biểu diễn biểu đồ trình tự theo dõi người dùng khác trên hệ thống



Hình 2.17: Biểu đồ trình tự xem danh sách người theo dõi

2.4.14 Biểu đồ trình tự xem danh sách nhấn tin

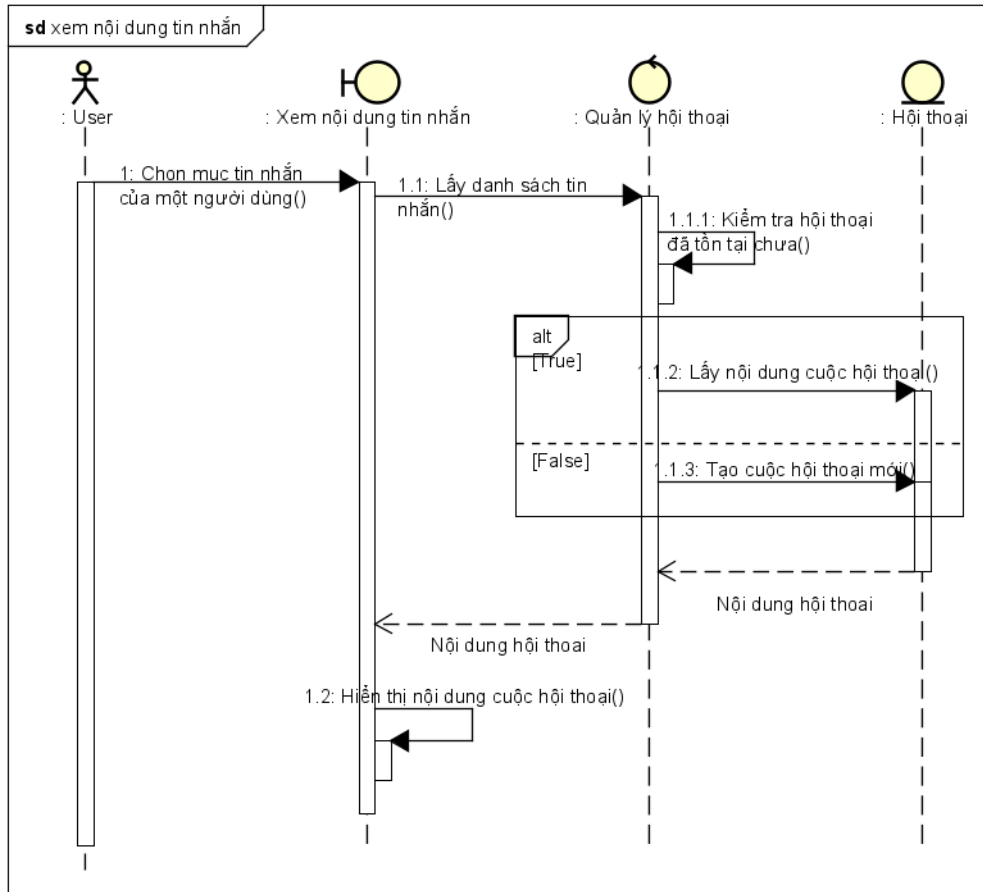
Biểu đồ trình tự xem danh sách nhấn tin mô tả quá trình xem danh sách nhấn tin của chính mình như hình 2.18



Hình 2.18: Biểu đồ trình tự xem danh sách nhấn tin

2.4.15 Biểu đồ trình tự xem tin nhắn

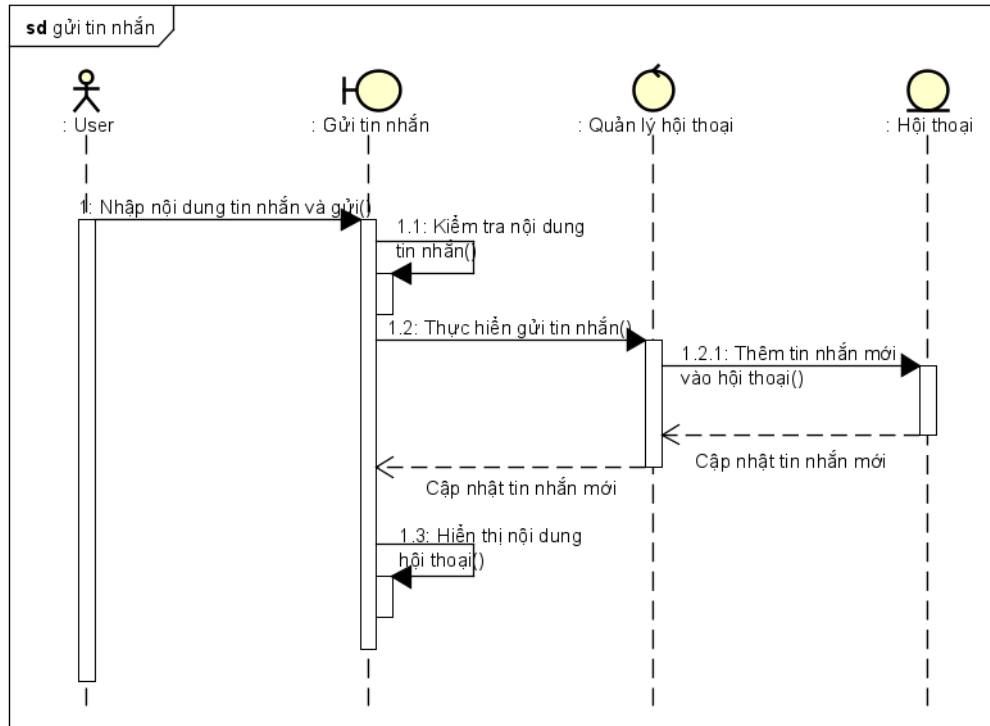
Hình 2.19 biểu diễn biểu đồ trình tự xem tin nhắn của bản thân với một người dùng khác trong hệ thống bằng cách kiểm tra xem giữa hai người đã tồn tại cuộc hội thoại nào chưa, nếu có rồi thì sẽ hiển thị danh sách các tin nhắn còn nếu chưa có thì sẽ tạo ra một cuộc hội thoại mới giữa hai người dùng:



Hình 2.19: Biểu đồ trình tự xem tin nhắn

2.4.16 Biểu đồ trình tự gửi tin nhắn

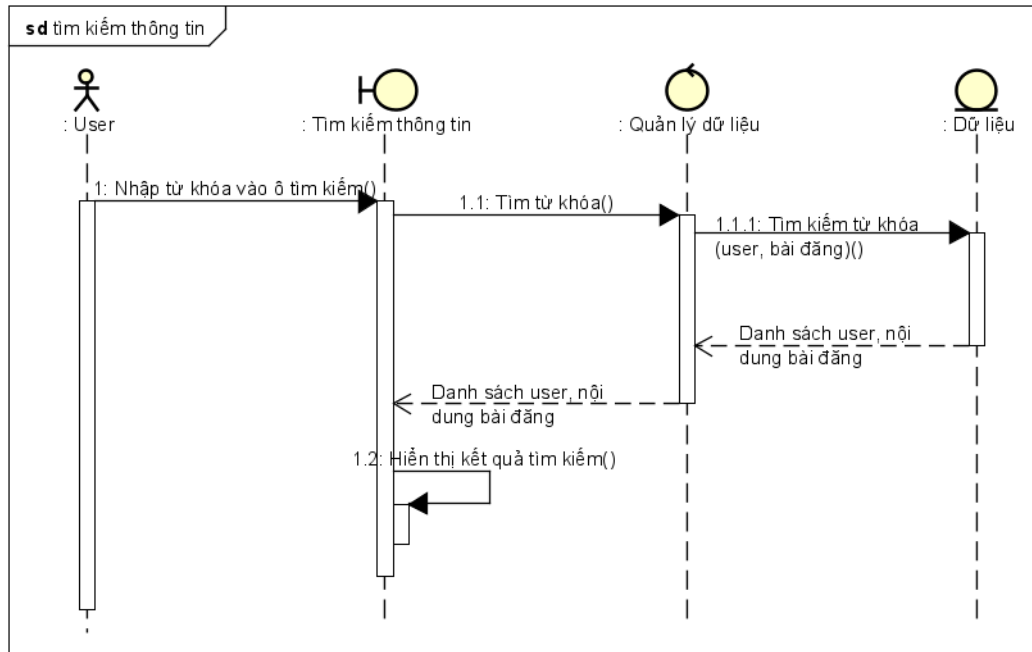
Biểu đồ trình tự gửi tin nhắn mô tả trình tự nhắn tin cho người dùng khác bằng cách thêm nội dung tin nhắn của người gửi vào cuộc hội thoại, hình 2.20 dưới đây mô tả quá trình đó:



Hình 2.20: Biểu đồ trình tự gửi tin nhắn

2.4.17 Biểu đồ trình tự tìm kiếm thông tin

Trình tự tìm kiếm thông tin được mô tả dưới dạng biểu đồ trình tự như hình 2.21

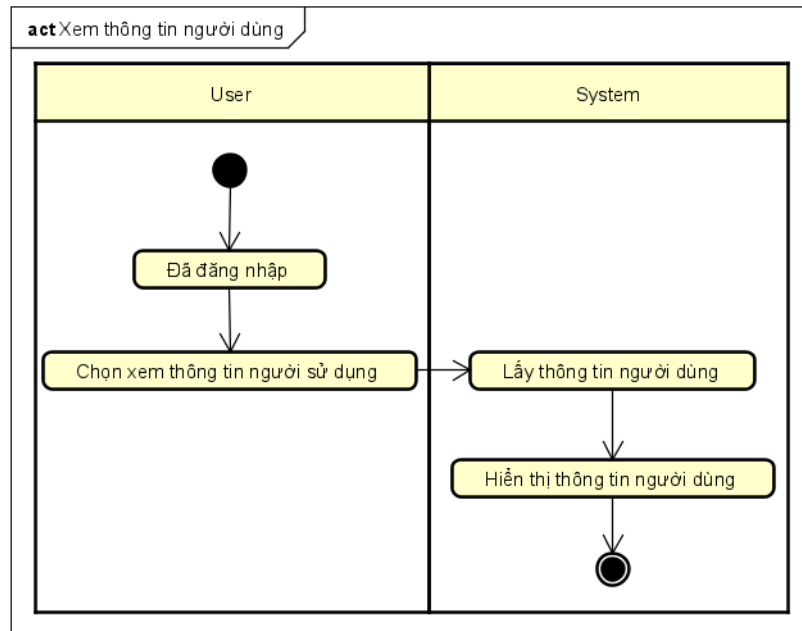


Hình 2.21: Biểu đồ trình tự tìm kiếm thông tin

2.5 Biểu đồ Hoạt động

2.5.1 Biểu đồ hoạt động xem thông tin người dùng

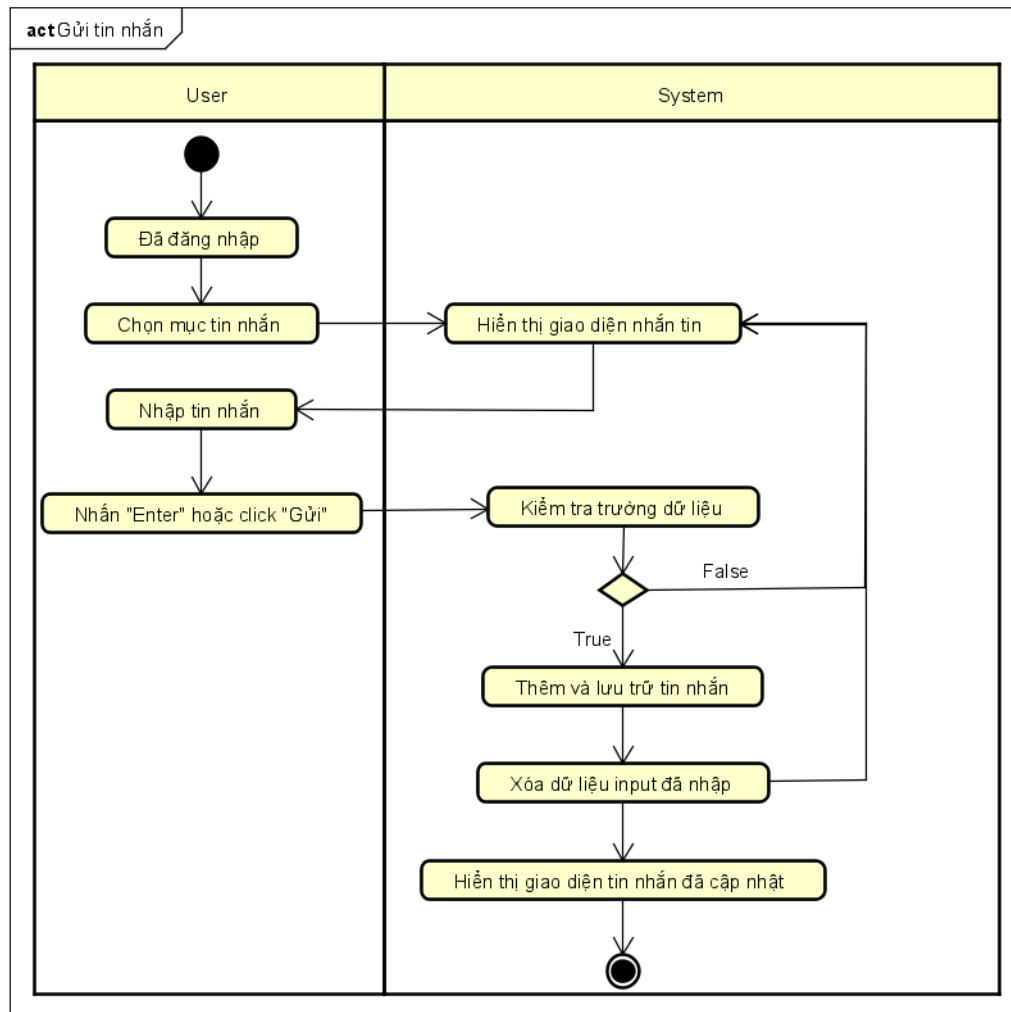
Từ hình 2.14 mô tả ở trên, sau đây là hình 2.22 biểu diễn biểu đồ hoạt động xem thông tin người dùng hệ thống



Hình 2.22: Biểu đồ hoạt động xem thông tin người dùng

2.5.2 Biểu đồ hoạt động gửi tin nhắn

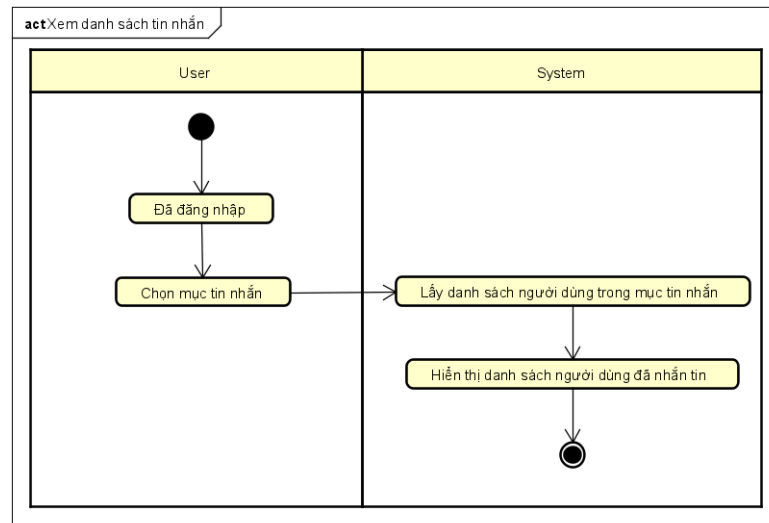
Từ hình 2.20 mô tả biểu đồ trình tự ở trên, sau đây là hình 2.23 biểu diễn biểu đồ hoạt động gửi tin nhắn



Hình 2.23: Biểu đồ hoạt động gửi tin nhắn

2.5.3 Biểu đồ hoạt động xem danh sách nhắn tin

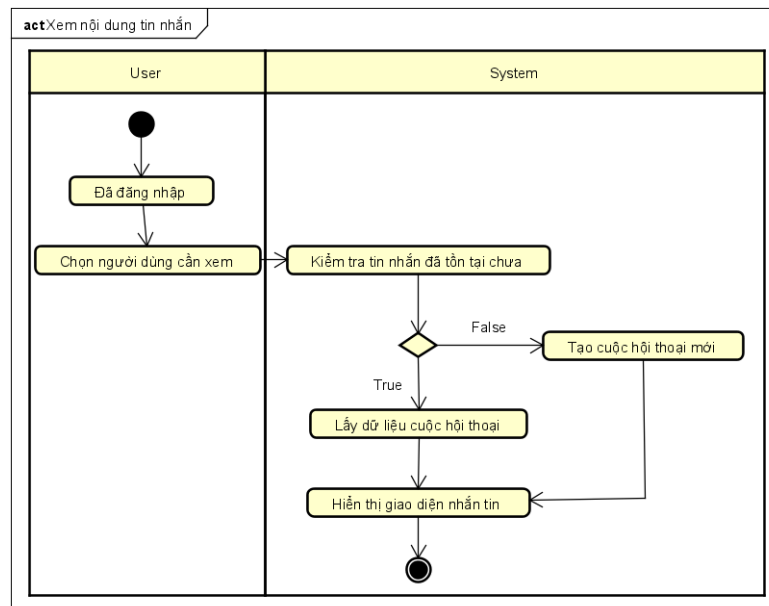
Từ hình 2.18 mô tả biểu đồ trình tự ở trên, sau đây là hình 2.24 biểu diễn biểu đồ hoạt động xem danh sách nhắn tin



Hình 2.24: Biểu đồ hoạt động xem tin nhắn

2.5.4 Biểu đồ hoạt động xem tin nhắn với người dùng khác

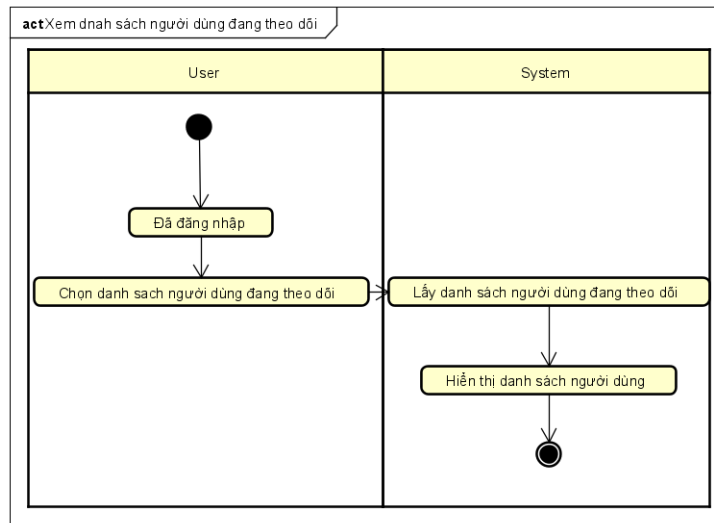
Từ hình 2.19 mô tả biểu đồ trình tự ở trên, sau đây là hình 2.25 biểu diễn biểu đồ hoạt động xem tin nhắn với người dùng khác khi đã tồn tại cuộc hội thoại



Hình 2.25: Biểu đồ hoạt động xem tin nhắn với người dùng khác

2.5.5 Biểu đồ hoạt động xem danh sách người theo dõi

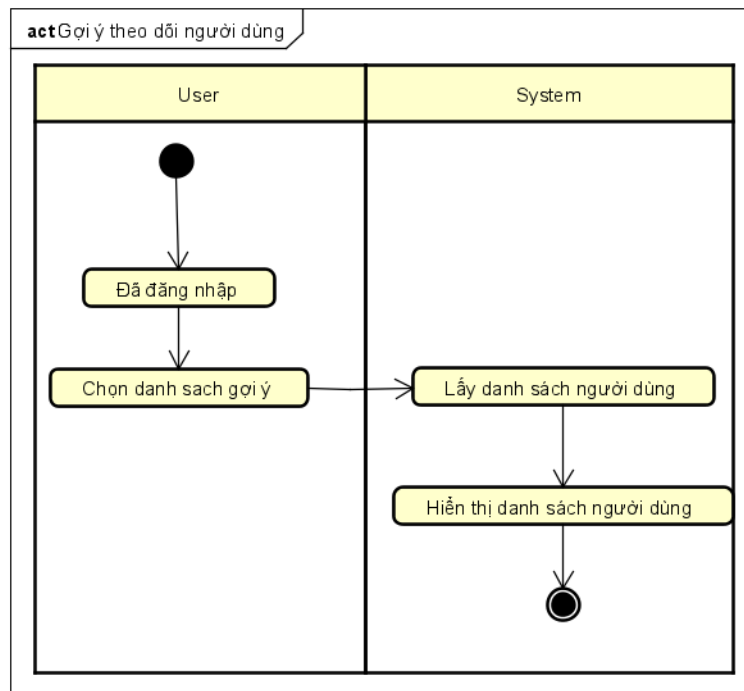
Từ hình 2.17 mô tả biểu đồ trình tự ở trên, sau đây là hình 2.26 biểu diễn biểu đồ hoạt động xem danh sách người dùng đang theo dõi



Hình 2.26: Biểu đồ hoạt động xem danh sách người theo dõi

2.5.6 Biểu đồ hoạt động xem danh sách người dùng gợi ý

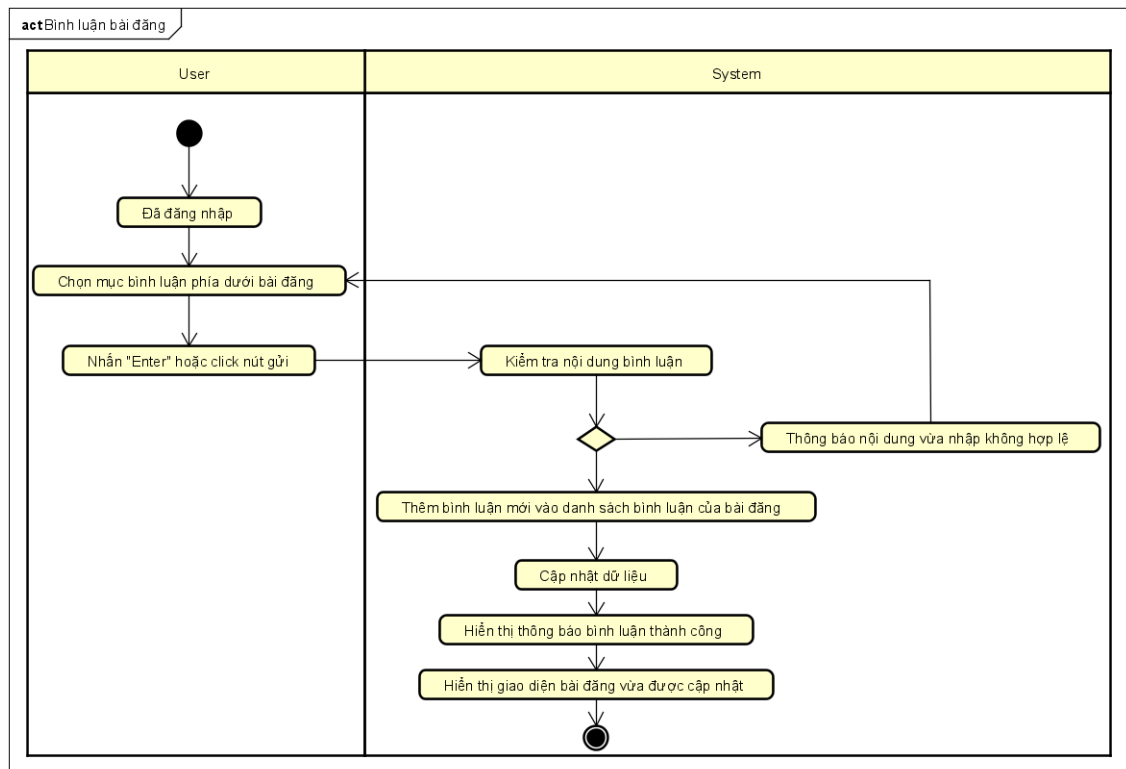
Từ hình 2.15 mô tả biểu đồ trình tự ở trên, sau đây là hình 2.27 biểu diễn biểu đồ hoạt động xem danh sách người dùng gợi ý



Hình 2.27: Biểu đồ hoạt động xem danh sách người dùng gợi ý

2.5.7 Biểu đồ hoạt động bình luận bài đăng

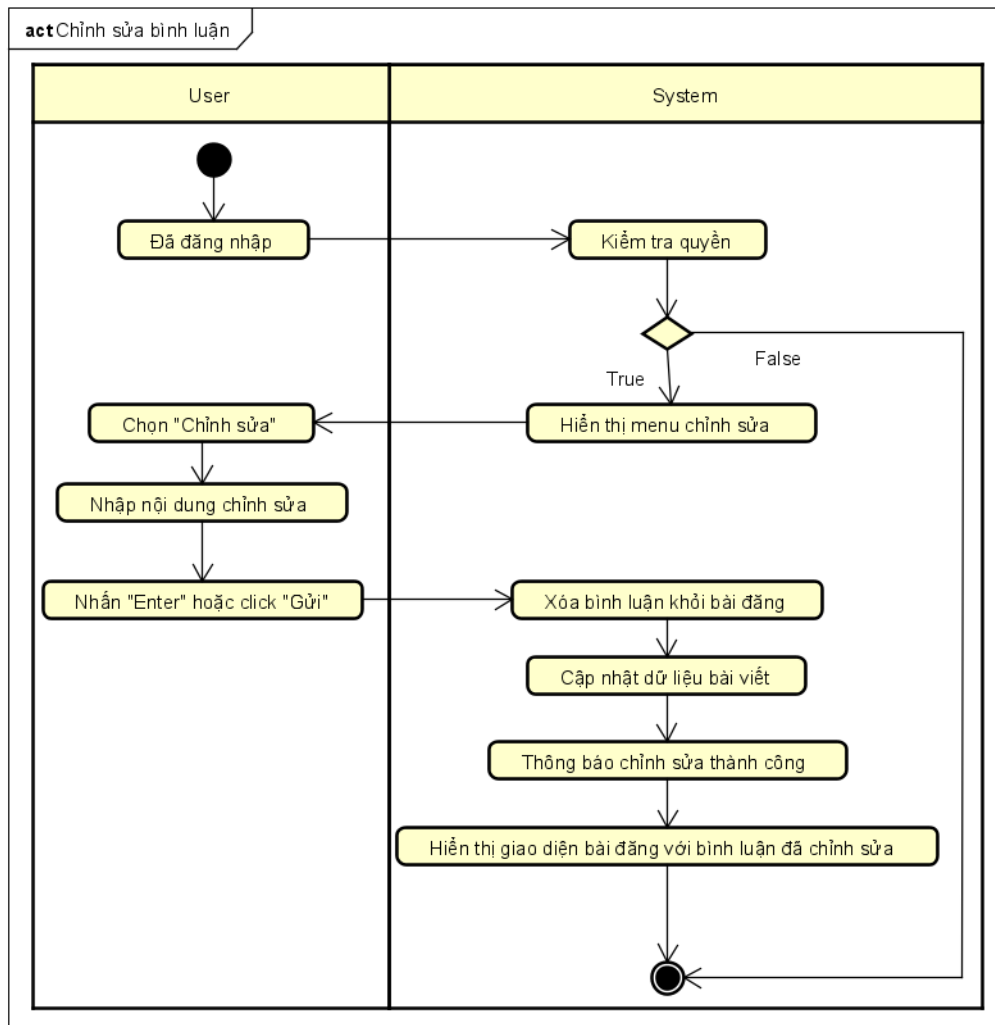
Từ hình 2.10 mô tả biểu đồ trình tự ở trên, sau đây là hình 2.28 biểu diễn biểu đồ hoạt động thêm bình luận mới vào bài đăng



Hình 2.28: Biểu đồ hoạt động bình luận bài đăng

2.5.8 Biểu đồ hoạt động chỉnh sửa bình luận

Từ sự mô tả biểu đồ trình tự chỉnh sửa bài viết ở trên phần 2.3 biểu đồ hoạt động chỉnh sửa bình luận được biểu diễn như hình ảnh 2.29



Hình 2.29: Biểu đồ hoạt động chỉnh sửa bình luận

2.6 Yêu cầu phi chức năng

2.6.1 Yêu cầu về kỹ thuật

Bảo mật : Áp dụng các biện pháp bảo mật như mã hóa dữ liệu, xác thực người dùng, hạn chế truy cập trái phép và giảm thiểu các lỗ hổng bảo mật tiềm ẩn.

Quản lý phiên: Đảm bảo quản lý phiên đăng nhập an toàn để bảo vệ dữ liệu và thông tin người dùng.

Độc lập với trình duyệt: Đảm bảo trang web hoạt động một cách tương thích trên các trình duyệt phổ biến như Google Chrome, Firefox, Safari và Microsoft Edge.

Tốc độ tải trang: Tối ưu hóa mã nguồn và tài nguyên để đảm bảo thời gian tải trang nhanh chóng và cải thiện trải nghiệm người dùng.

Hỗ trợ và bảo trì: Đảm bảo cung cấp hỗ trợ và bảo trì liên tục để giải quyết sự cố, nâng cấp và cập nhật hệ thống.

2.6.2 Yêu cầu về thiết kế

Thiết kế đơn giản và dễ sử dụng: Tạo ra một giao diện người dùng đơn giản, trực quan và dễ sử dụng để người dùng có thể tìm kiếm, duyệt và tải xuống các báo cáo một cách dễ dàng. Tương thích di động: Đảm bảo trang web được thiết kế đáp ứng (responsive) để hiển thị tốt trên các thiết bị di động như điện thoại di động và máy tính bảng.

Tính năng tìm kiếm nâng cao: Cung cấp một công cụ tìm kiếm mạnh mẽ để người dùng có thể lọc và tìm kiếm các báo cáo dựa trên tiêu đề, tác giả, từ khóa, ngày đăng, và các tiêu chí khác.

Trang cá nhân người dùng: Cung cấp cho người dùng một trang cá nhân để quản lý và hiển thị thông tin cá nhân, bài viết và báo cáo đã đăng.

Trang chi tiết báo cáo: Cung cấp một trang chi tiết cho mỗi báo cáo nghiên cứu, hiển thị thông tin chi tiết về báo cáo, tác giả, tóm tắt, và các tài liệu liên quan.

Chia sẻ và kết nối mạng xã hội: Tích hợp các nút chia sẻ xã hội để người dùng có thể dễ dàng chia sẻ bài viết và báo cáo thông qua các mạng xã hội phổ biến như Facebook, Twitter, LinkedIn.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

Chương 2 đã trình bày hiện trạng và phân tích thiết kế hệ thống. Chương 3 sẽ trình bày các công nghệ sử dụng để xây dựng trang web này. Các công cụ sử dụng bao gồm: ngôn ngữ lập trình Javascript(Frontend - ReactJS, Backend - NodeJS/-Express, Database - MongoDB, Deploy - Render.com)

Mạng xã hội chia sẻ báo cáo nghiên cứu cho các nhà nghiên cứu khoa học		
Client	Server	Deploy
React	Node.js	render.com
Bootstrap 4	ExpressJS	
Redux	MongoDB	
Axios		

Hình 3.1: Danh sách công nghệ áp dụng

Hình 3.1 đưa ra một cái nhìn tổng quan về công nghệ sử dụng trong đồ án này, chi tiết về công nghệ sẽ được trình bày trong phần ???. Sau đây là phần trình bày về các ngôn ngữ được nêu ra trong hình và lý do lựa chọn. Dưới đây là phần trình bày về các ngôn ngữ lập trình JavaScript và các platform/framework tương ứng, cùng lý do lựa chọn chúng.

Dưới đây là phần trình bày về các ngôn ngữ lập trình JavaScript và các platform/framework tương ứng, cùng lý do lựa chọn chúng.

3.1 Javascript

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web. Là ngôn ngữ kịch bản phía máy khách(client), JavaScript là một trong những công nghệ cốt lõi của World Wide Web. Ví dụ: khi duyệt internet, bất cứ khi nào bạn thấy quảng cáo quay vòng dạng hình ảnh, menu thả xuống nhấp để hiển thị hoặc màu sắc phần tử thay đổi động trên trang web cũng chính là lúc bạn thấy các hiệu ứng của JavaScript. [1]

3.2 Frontend

ReactJS là một thư viện JavaScript phổ biến được phát triển bởi Facebook vào năm 2013. Nó được sử dụng để xây dựng giao diện người dùng tương tác trên trang web. Điểm mạnh của ReactJS là cách tiếp cận khác biệt trong việc cập nhật

giao diện, đảm bảo hiệu suất cao và tăng trải nghiệm người dùng. ReactJS sử dụng Virtual DOM (Document Object Model) để giảm thiểu số lần truy cập tới DOM thực sự, giúp cải thiện tốc độ render và hiệu năng tổng thể của ứng dụng. Thêm vào đó, cộng đồng phong phú và nhiều thư viện hỗ trợ hơn nữa giúp phát triển ứng dụng dễ dàng và nhanh chóng. [2]

Lý do lựa chọn ReactJS: ReactJS được lựa chọn vì tính hiệu quả trong việc xây dựng giao diện tương tác và khả năng tối ưu hóa hiệu suất render. Sự hỗ trợ mạnh mẽ từ cộng đồng và hệ sinh thái của ReactJS cũng giúp đơn giản hóa quá trình phát triển và đảm bảo tính linh hoạt trong việc mở rộng ứng dụng.

Redux là một thư viện quản lý trạng thái phổ biến trong ứng dụng React. Nó giúp tổ chức và điều phối trạng thái của ứng dụng một cách dễ dàng và hiệu quả. Redux hoạt động dựa trên kiến trúc Flux và sử dụng một store duy nhất để lưu trữ toàn bộ trạng thái của ứng dụng. Các thành phần React có thể truy cập và cập nhật trạng thái từ store này thông qua các hàm được cung cấp bởi Redux. [3]

Lý do lựa chọn Redux: Redux được lựa chọn để quản lý trạng thái của ứng dụng vì tính đơn giản và hiệu quả trong việc quản lý trạng thái phức tạp. Sự tích hợp giữa React và Redux giúp giảm thiểu các vấn đề về quản lý trạng thái và tạo sự đồng nhất trong cả quá trình phát triển.

Axios là một thư viện HTTP client phổ biến và được sử dụng trong ứng dụng React để thực hiện các yêu cầu HTTP giữa máy khách và máy chủ. Nó hỗ trợ các phương thức HTTP như GET, POST, PUT, DELETE, vv và cung cấp các tính năng xử lý yêu cầu và phản hồi một cách dễ dàng. Axios giúp đơn giản hóa việc làm việc với API và tương tác với cơ sở dữ liệu hoặc các dịch vụ bên ngoài. [4]

Lý do lựa chọn Axios: Axios được lựa chọn để xử lý các yêu cầu HTTP giữa frontend và backend vì tính đơn giản và khả năng hỗ trợ đa dạng các phương thức HTTP. Sự tích hợp giữa Axios và React giúp tạo sự dễ dàng trong việc giao tiếp và trao đổi dữ liệu giữa các thành phần của ứng dụng.

3.3 Lý do lựa chọn sự kết hợp cả ReactJS, Redux và Axios

Việc kết hợp cả ReactJS, Redux và Axios mang lại nhiều lợi ích cho việc xây dựng ứng dụng web:

Quản lý trạng thái hiệu quả: Redux giúp quản lý trạng thái ứng dụng một cách dễ dàng và đồng nhất, giúp giảm thiểu lỗi và đảm bảo tính chính xác của dữ liệu hiển thị.

Tối ưu hóa hiệu suất: Sử dụng ReactJS và Virtual DOM giúp tối ưu hóa tốc độ render và cải thiện hiệu suất tổng thể của ứng dụng.

Giao tiếp với máy chủ: Axios hỗ trợ việc gửi yêu cầu HTTP giữa frontend và backend, giúp tương tác với cơ sở dữ liệu và các dịch vụ bên ngoài một cách linh hoạt và dễ dàng.

Linh hoạt và mở rộng: Kết hợp cả ba công nghệ giúp tạo sự linh hoạt và khả năng mở rộng trong việc phát triển và mở rộng ứng dụng theo nhu cầu của người dùng.

Tóm lại, sự kết hợp cả ReactJS, Redux và Axios giúp tối ưu hóa hiệu suất và quản lý trạng thái của ứng dụng một cách dễ dàng và hiệu quả, đồng thời cung cấp khả năng tương tác với máy chủ và tích hợp với các dịch vụ bên ngoài.

3.4 Backend - NodeJS/Express

Node.js là một nền tảng phát triển độc lập được xây dựng trên JavaScript Runtime của trình duyệt Chrome. Điều này cho phép chúng ta xây dựng các ứng dụng mạng phía máy chủ (server-side) một cách nhanh chóng và hiệu quả. Node.js hỗ trợ mô hình lập trình không đồng bộ (asynchronous programming), giúp tối ưu hoá hiệu suất và xử lý đồng thời hàng ngàn kết nối. Node.js cũng có một cộng đồng lớn và sự hỗ trợ mạnh mẽ từ các công cụ và thư viện bên thứ ba, giúp việc phát triển ứng dụng dễ dàng và linh hoạt. [5]

Express là một framework được xây dựng trên Node.js, giúp đơn giản hóa việc xây dựng các ứng dụng web phức tạp. Nó cung cấp các tính năng mạnh mẽ như định tuyến (routing), middleware, xử lý yêu cầu và phản hồi (request-response), giúp việc xây dựng API và ứng dụng web trở nên dễ dàng và hiệu quả. [6]

Lý do lựa chọn: Node.js và Express được lựa chọn vì khả năng xử lý đa luồng (thông qua event và callback), giúp ứng dụng đáp ứng tốt trong các tình huống có nhiều kết nối cùng lúc. Ngoài ra, việc sử dụng JavaScript cả phía máy chủ và phía máy khách giúp đơn giản hóa quản lý mã nguồn và tạo sự đồng nhất trong cả quá trình phát triển.

3.5 Database - MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở, được xây dựng đa nền tảng và viết bằng C++. MongoDB lưu trữ dữ liệu dưới dạng văn bản dưới dạng các cặp giá trị và trường, tương tự như đối tượng JSON. Điều này giúp MongoDB linh hoạt và dễ dàng mở rộng theo nhu cầu của ứng dụng. Hơn nữa, MongoDB hỗ trợ tính năng đa luồng và có khả năng xử lý hàng ngàn yêu cầu cùng lúc, giúp cải thiện hiệu suất và đáp ứng tốt trong các ứng dụng có lượng truy vấn lớn. [7]

Lý do lựa chọn: MongoDB được lựa chọn vì tính linh hoạt và khả năng mở

rộng cao. Sự lưu trữ dữ liệu dưới dạng JSON giúp đơn giản hóa việc làm việc với dữ liệu và dễ dàng tích hợp với các ứng dụng JavaScript. Khả năng xử lý đa luồng cùng với khả năng mở rộng theo chiều ngang giúp đáp ứng nhanh chóng với các yêu cầu của ứng dụng.

3.6 Deploy - Render.com

Render.com là một nền tảng dịch vụ đám mây (cloud service platform) giúp việc triển khai ứng dụng web trở nên đơn giản và hiệu quả. **render.com** hỗ trợ việc triển khai các ứng dụng frontend và backend dựa trên các công nghệ như ReactJS, Node.js và MongoDB. Nền tảng này cung cấp các công cụ dễ dàng để quản lý ứng dụng, mở rộng quy mô, và đảm bảo hiệu suất cao cho các ứng dụng trực tuyến. [8]

Lý do lựa chọn: Render.com được lựa chọn vì tính đơn giản và tiện lợi trong việc triển khai và quản lý ứng dụng. Nền tảng này hỗ trợ triển khai các công nghệ mà chúng ta đã lựa chọn (ReactJS, Node.js và MongoDB), giúp tối ưu hoá quá trình triển khai và giảm thiểu công việc phức tạp trong việc quản lý hệ thống.

Tóm lại, việc lựa chọn Node.js, React và MongoDB là một quyết định mang lại sự linh hoạt, hiệu suất và khả năng mở rộng cho việc phát triển ứng dụng và làm việc nhóm trên nền tảng web. Sự hỗ trợ từ Render.com cũng đảm bảo việc triển khai và quản lý ứng dụng trở nên đơn giản và hiệu quả.

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Chương 3 đã trình bày về những công nghệ được sử dụng để xây dựng đồ án. Trong chương 4 này sẽ trình bày về thiết kiến trúc, thiết kế lớp, thiết kế dữ liệu, giao diện, kết quả đạt được cùng những ý kiến đóng góp.

4.0.1 Lựa chọn kiến trúc phần mềm

Trong thiết kế kiến trúc của ứng dụng mạng xã hội, tôi đã lựa chọn mô hình phân tầng (Layered Architecture), với sự tách biệt giữa phần client (gồm các thành phần View) và phần server (bao gồm các thành phần Controller và Model). Dưới đây là mô tả chi tiết về kiến trúc phân tầng cho ứng dụng của tôi.

Phần Client (View): Gồm các thành phần giao diện người dùng, được triển khai bằng HTML, CSS và JavaScript. Phần này có trách nhiệm hiển thị thông tin và tương tác với người dùng. Nó nhận thông tin từ phần Controller và hiển thị nó cho người dùng, đồng thời gửi các sự kiện từ người dùng đến phần Controller để xử lý.

Phần Server (Controller và Model): Gồm hai thành phần chính:

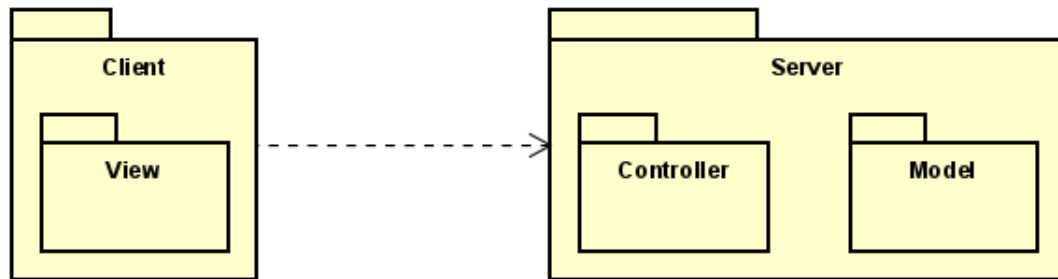
Controller: Được triển khai bằng các mã nguồn phía server (ví dụ: Node.js). Controller là thành phần trung gian giữa phần Client và phần Model. Nó nhận các yêu cầu từ phần Client, xử lý logic ứng dụng và gửi yêu cầu tương ứng đến phần Model để truy xuất, thêm, sửa, xóa dữ liệu. Sau đó, Controller trả về kết quả cho phần Client để hiển thị cho người dùng.

Model: Đại diện cho dữ liệu và logic xử lý liên quan đến dữ liệu. Model bao gồm các lớp và đối tượng để thao tác với cơ sở dữ liệu và lưu trữ thông tin về người dùng, bài viết, bình luận, v.v. Model cung cấp các phương thức để truy xuất, thêm, sửa, xóa dữ liệu và xử lý các yêu cầu từ phần Controller.

Sự tách biệt rõ ràng giữa phần Client và phần Server trong kiến trúc phân tầng giúp tăng tính tổ chức, dễ bảo trì và mở rộng ứng dụng mạng xã hội. Nó cho phép phát triển độc lập giữa giao diện người dùng và logic xử lý, cũng như tạo điều kiện cho việc mở rộng các tính năng và tích hợp với các hệ thống khác trong tương lai.

4.0.2 Thiết kế tổng quan

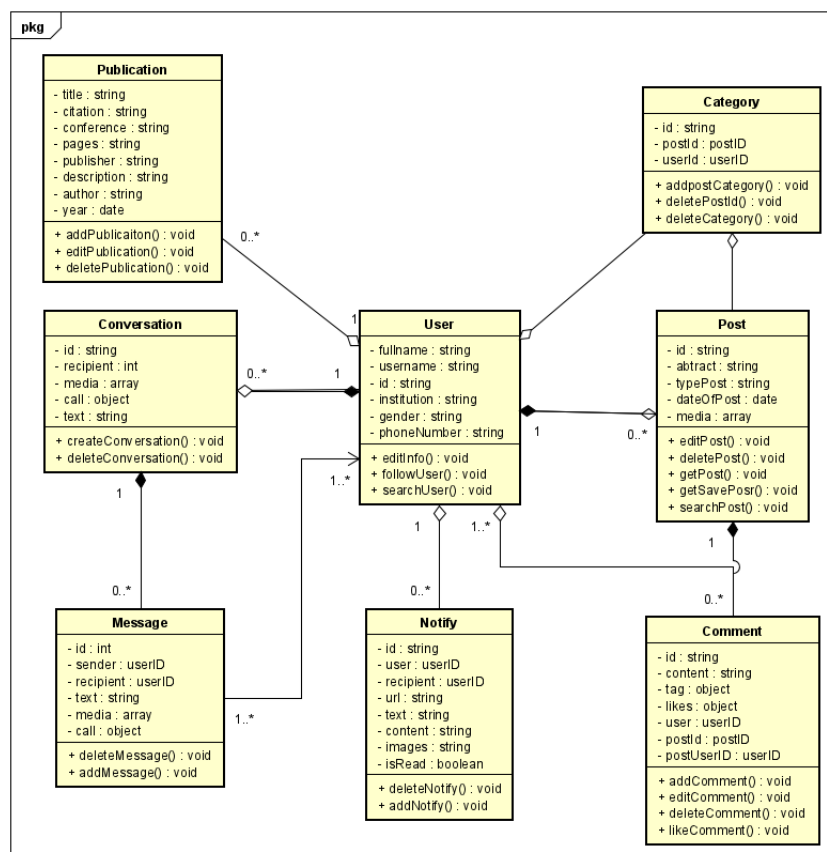
Hình 4.2 dưới đây là hình ảnh minh họa cho phần thiết kế gói cho ứng dụng của tôi như đã trình bày ở 4.0.1



Hình 4.1: Biểu đồ phụ thuộc gói

4.0.3 Thiết kế chi tiết gói

Sau đây là phần thiết kế chi tiết gói được minh họa cho biểu đồ phụ thuộc gói 4.1 ở trên



Hình 4.2: Thiết kế chi tiết gói

4.1 Thiết kế chi tiết

4.1.1 Thiết kế giao diện

Bảng 4.1 dưới đây là bảng thông tin thiết kế màn hình

Thông tin	Thông số
Độ phân giải màn hình	HD (720p). Full HD (1080p). QHD (1440p).
Kích thước màn hình	Từ 1366 x 768 trở lên. Tối ưu với 1920 x 1080.
Chế độ màn hình	Toàn màn hình.
Font chữ	Tiêu đề và từ khóa: Arial. Các nút bấm : SansBold

Bảng 4.1: Thông tin thiết kế giao diện

Thiết kế màn hình

Phần thiết kế màn hình được dựng bằng phần mềm Photoshop. Một số hình ảnh mang tính chất tham khảo. Phần thiết kế hình ảnh này không phải là giao diện cuối cùng của trang web trò chơi.

Thiết kế giao diện chính:

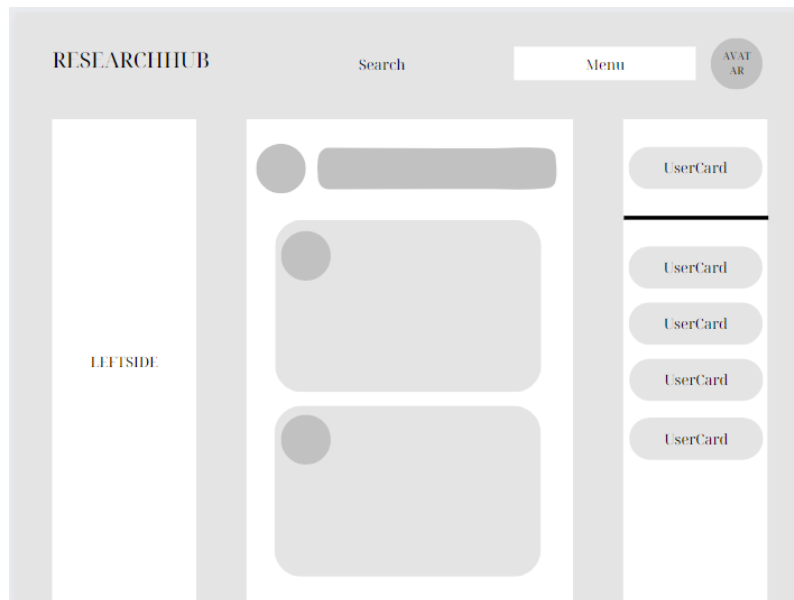
Hình 4.3 dưới đây mô tả màn hình thiết kế giao diện chính bao gồm các thành phần sau:

Header: Chứa các nút điều hướng đến các thành phần khác của trang web. Một số nút có chức năng như: home(trang chính), chat(trang chi tiết mục tin nhắn), discover(mục chứa các bài đăng của những người dùng khác mà người dùng không theo dõi có thể khám phá), notify(mục thông báo), cuối cùng là mục thông tin của user(profile page) và nút logout.

Left side bar: thiết kế chính của mục này dành cho những tác vụ thuộc về phần user(cá nhân hóa). Newfeed, chat, profile page. Một số tác vụ khác có thể mở rộng khi nhu cầu sử dụng tăng lên sẽ được thực hiện mở rộng ở phần left side bar này.

Giao diện chính: Hiển thị chức năng đăng bài dưới dạng thanh trạng thái. Chứa các bài đăng thuộc phần home post của người dùng.(danh sách các bài đăng được lấy từ những người mà người dùng đã theo dõi để có thể cập nhật thông tin của người theo dõi liên tục)

Right side bar: phần này hiển thị danh sách gợi ý người dùng tương đương trong hệ thống(có cùng tổ chức, kĩ năng..)

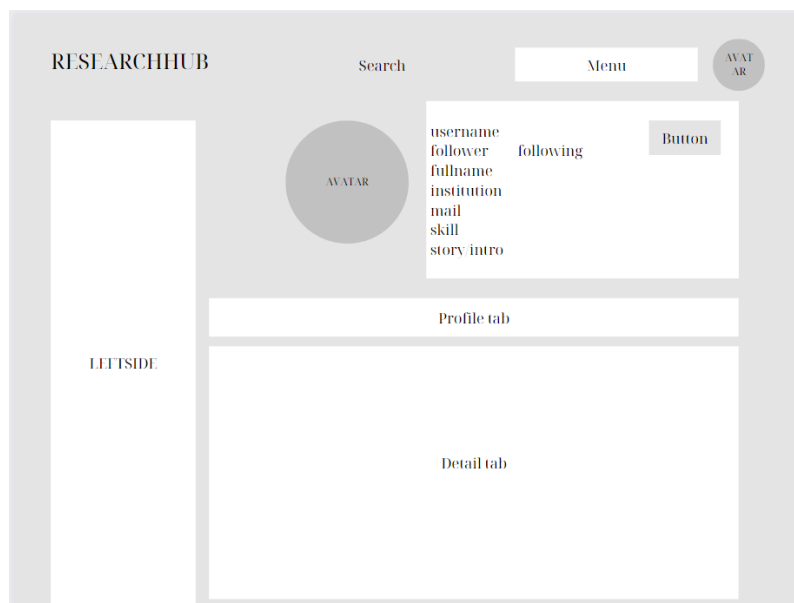


Hình 4.3: Thiết kế giao diện màn hình chính của trang web

Thiết kế giao diện profile page

Gồm 2 phần chính: Thông tin của người dùng hiển thị ở phần "info" và "profile tab". Thông tin user chứa các thông tin mà người dùng cung cấp có thể được chỉnh sửa theo ý muốn. Các phần trong "profile tab" hiển thị các bài đăng, bài đã lưu, các chủ đề mà người dùng có thể tùy chỉnh, và các bài báo đã được công bố mà người dùng có thể thêm vào.

Hình 4.4 là hình minh họa về thiết kế giao diện profile page

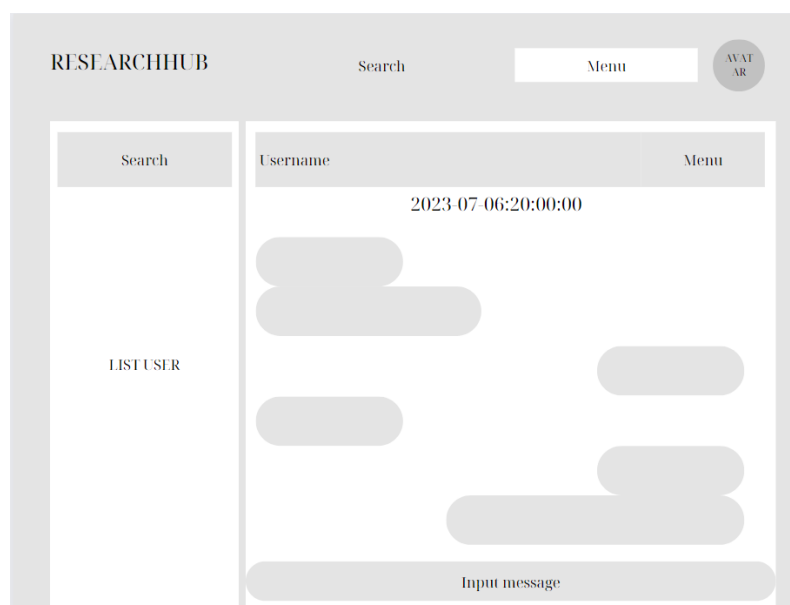


Hình 4.4: Thiết kế giao diện profile page

Thiết kế giao diện chat:

Gồm 2 phần: Leftside: chứa thanh tìm kiếm người dùng nếu như trong phần userlist chưa có tin nhắn với người dùng đó. Phần chi tiết tin nhắn: hiển thị tên người dùng đang nhắn tin ở phần header của phần chi tiết, các menu của header(voice call, video call, delete conversation). Hiển thị danh sách tin nhắn của 2 người dùng, chứa thông tin về các tin nhắn đã gửi(thời gian, nội dung...). Phần dưới cùng là phần nhập tin nhắn.

Hình 4.5 là hình minh họa về thiết kế giao diện chat



Hình 4.5: Thiết kế giao diện chat

Thiết kế giao diện chi tiết báo cáo khoa học đã được công khai:

Hình 4.6 là hình minh họa về thiết kế giao diện chi tiết báo cáo khoa học gồm 2 phần: phần bên trái là thông tin tác giả đăng tải báo cáo khoa học bao gồm ảnh đại diện và username của tác giả. Phần bên phải là chi tiết về báo cáo khoa học bao gồm: tiêu đề báo cáo, các thông tin liên quan đến báo cáo như: tác giả tham gia, ngày công bố, nhà xuất bản, hội nghị, mô tả... được người dùng hệ thống(các nhà nghiên cứu) đăng tải lên đã được công bố trên các tạp chí, hội nghị khoa học.

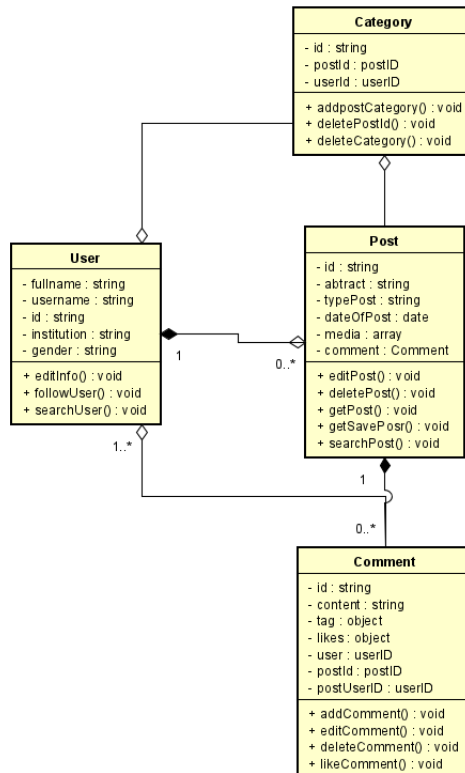


Hình 4.6: Thiết kế giao diện chi tiết báo cáo khoa học

4.1.2 Thiết kế lớp

Trong phần này sẽ trình bày về các lớp liên quan đến phần tạo các bài đăng, phần chính của trang mạng xã hội chia sẻ báo cáo khoa học.

Hình 4.7 dưới đây minh họa một phần nhỏ về biểu đồ lớp liên quan đến các chức năng mà user có thể sử dụng để đăng tải báo cáo khoa học



Hình 4.7: Biểu đồ lớp liên quan đến chức năng quản lý các bài đăng

	Tên	Kiểu dữ liệu	Vai trò
Thuộc tính	id	string	id người dùng hệ thống(unique)
	fullname	string	tên người dùng đầy đủ
	username	string	tên người dùng(duy nhất) dùng để tìm kiếm
	institution	string	tổ chức mà người dùng
Phương thức	editInfo()	void	Hàm này sử dụng để chỉnh sửa thông tin người dùng.
	followUser()	void	Hàm này thực hiện chức năng theo dõi người dùng hệ thống bởi một người dùng khác
	searchUser()	void	Hàm này thực hiện chức năng tìm kiếm người dùng

Bảng 4.2: Đặc tả lớp User

	Tên	Kiểu dữ liệu	Vai trò
Thuộc tính	id	string	id bài đăng(unique)
	abstract	string	nội dung bài đăng
	typePost	string	thể loại bài đăng
	dateOfPublication	string	ngày công bố báo cáo
	media	array	mảng các file ảnh, video, pdf...
	comment	comment	các bình luận trong bài đăng
Phương thức	editPost()	void	Hàm này sử dụng để chỉnh sửa nội dung bài đăng.
	deletePost()	void	Hàm này thực hiện chức năng xóa bài đăng
	getPost()	void	Hàm này thực hiện chức năng lấy ra bài đăng đã được công bố
	getSavePost()	void	Hàm này thực hiện chức năng lấy ra bài đăng đã được lưu lại
	searchpost()	void	Hàm này thực hiện chức năng tìm kiếm bài đăng

Bảng 4.3: Đặc tả lớp Post

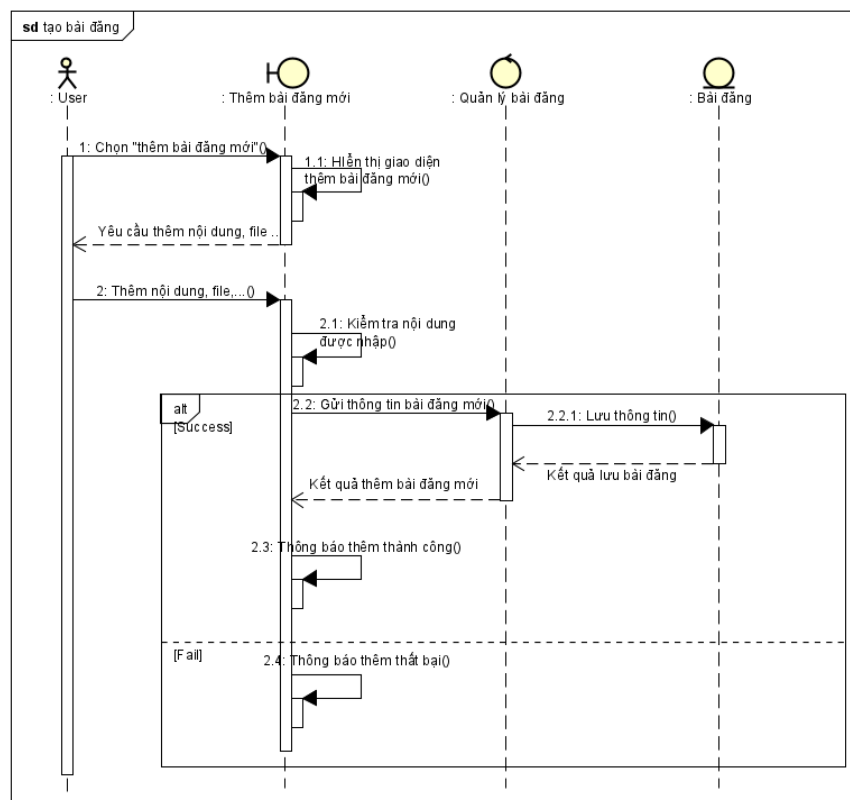
	Tên	Kiểu dữ liệu	Vai trò
Thuộc tính	id	string	id bình luận(unique)
	content	string	nội dung bình luận
	tag	string	hashtag cho bài đăng để có thể tìm kiếm dựa vào hashtag đó
	likes	string	số lượng người thích bình luận
	user	array	người đăng bình luận
	postId	postID	id của bài đăng chứa bình luận
	postUserId	userID	id của người đăng tải bài viết
Phương thức	addComment()	void	Hàm này thêm nội dung bình luận vào bài viết.
	editComment()	void	Hàm này thực hiện chức năng chỉnh sửa bình luận
	deleteComment()	void	Hàm này xóa bình luận được chọn
	likeComment()	void	Kiểm tra và lấy ra số lượt thích bình luận

Bảng 4.4: Đặc tả lớp Comment

	Tên	Kiểu dữ liệu	Vai trò
Thuộc tính	id	string	id chủ đề(unique)
	postId	postID	id bài đăng
	userId	userID	id người dùng sở hữu
Phương thức	addCategory()	void	Hàm này cho phép người dùng có thêm vào trang cá nhân của mình bộ sưu tập danh sách các bài có cùng chủ đề
	deleteCategory()	void	Hàm này thực hiện chức năng xóa bộ sưu tập
	editCategory()	void	Chỉnh sửa danh sách các bài đăng hoặc sửa lại chủ đề đã có

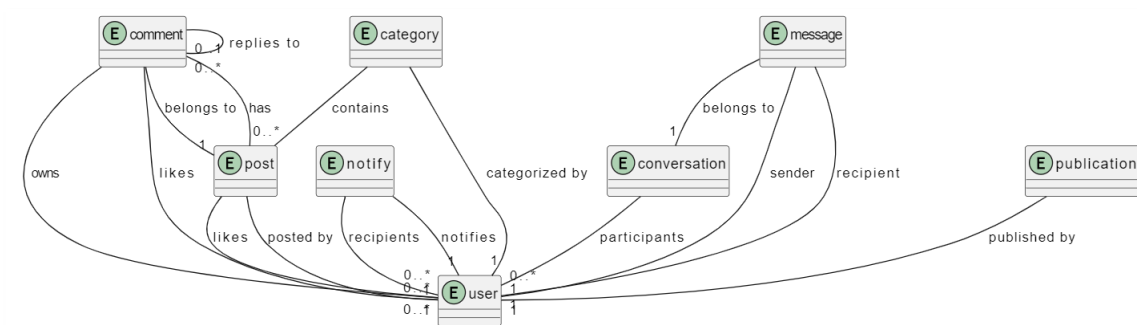
Bảng 4.5: Đặc tả lớp Category

Hình 4.8 dưới đây là biểu đồ thể hiện trình tự đăng tải bài viết



Hình 4.8: Biểu đồ trình tự đăng bài

4.1.3 Thiết kế cơ sở dữ liệu



Hình 4.9: ERD

Giải thích sơ đồ thực thể liên kết được minh họa ở hình 4.9

Trong hệ thống, có các thực thể chính bao gồm "comment" (bình luận), "notify" (thông báo), "message" (tin nhắn), "post" (bài viết), "conversation" (cuộc trò chuyện), "user" (người dùng), "category" (danh mục chủ đề), và "publication" (danh sách báo cáo khoa học).

Quan hệ giữa các thực thể được mô tả như sau:

Mỗi "comment" (bình luận) thuộc về một "user" (người dùng) duy nhất.

Một "user" (người dùng) có thể thích nhiều "comment" (bình luận).

Một "comment" (bình luận) có thể trả lời cho một "comment" (bình luận) khác.

Mỗi "comment" (bình luận) thuộc về một "post" (bài viết) duy nhất.

Mỗi "notify" (thông báo) được gửi cho một "user" (người dùng) duy nhất.

Một "user" (người dùng) có thể nhận được nhiều "notify" (thông báo).

Mỗi "message" (tin nhắn) thuộc về một "conversation" (cuộc trò chuyện) duy nhất.

Một "message" (tin nhắn) được gửi bởi một "user" (người dùng) duy nhất.

Một "message" (tin nhắn) được nhận bởi một "user" (người dùng) duy nhất.

Mỗi "post" (bài viết) được đăng bởi một "user" (người dùng) duy nhất.

Một "user" (người dùng) có thể thích nhiều "post" (bài viết).

Một "post" (bài viết) có thể có nhiều "comment" (bình luận).

Một "conversation" (cuộc trò chuyện) có thể có nhiều "user" (người dùng) tham gia.

Một "category" (danh mục) được phân loại bởi một "user" (người dùng) duy nhất.

Một "category" (danh mục) có thể chứa nhiều "post" (bài viết).

Các quan hệ trên thể hiện mối liên kết và phụ thuộc giữa các thực thể trong hệ thống.

Bảng 4.6: Các trường trong các collection của cơ sở dữ liệu

Collection	Trường	Mô tả
Comment	content tag reply likes user postId postUserId	Nội dung của bình luận Gắn thẻ người dùng liên quan đến bình luận ObjectId của bình luận phản hồi Mảng ObjectId của người dùng thích bình luận ObjectId của người dùng tạo bình luận ObjectId của bài viết liên quan ObjectId của người dùng tạo bài viết
Notify	user recipients url text	ObjectId của người dùng nhận thông báo Mảng ObjectId của người dùng được thông báo Đường dẫn liên kết trong thông báo Nội dung văn bản của thông báo
	content image isRead	Nội dung chi tiết của thông báo Đường dẫn hình ảnh trong thông báo Trạng thái đánh dấu đã đọc
Message	conversation sender recipient text media call	ObjectId của cuộc trò chuyện ObjectId của người gửi tin nhắn ObjectId của người nhận tin nhắn Nội dung văn bản của tin nhắn Mảng các tệp đa phương tiện trong tin nhắn Thông tin về cuộc gọi
Post	content images typePost dateOfPublication hashtag likes comments user	Nội dung của bài viết Mảng đường dẫn hình ảnh trong bài viết Loại bài viết Ngày công bố bài báo cáo Mảng các hashtag liên quan Mảng ObjectId của người dùng thích bài viết Mảng ObjectId của bình luận trong bài viết ObjectId của người dùng tạo bài viết
Conversation	recipients	Mảng ObjectId của người dùng trong cuộc trò chuyện

Bảng 4.6: Các trường trong các collection của cơ sở dữ liệu (tiếp theo)

Collection	Trường	Mô tả
	text media call	Nội dung văn bản của cuộc trò chuyện Mảng các tệp đa phương tiện trong cuộc trò chuyện Thông tin về cuộc gọi
User	fullname username email password avatar skill gender institution story website followers following saved	Họ và tên đầy đủ của người dùng Tên đăng nhập của người dùng Địa chỉ email của người dùng Mật khẩu của người dùng Đường dẫn hình đại diện của người dùng Mảng các kỹ năng của người dùng Giới tính của người dùng Tên tổ chức của người dùng Thông tin giới thiệu của người dùng Địa chỉ website cá nhân của người dùng Mảng ObjectId của người dùng theo dõi Mảng ObjectId của người dùng được theo dõi Mảng ObjectId của người dùng đã lưu
Category	userID postID topic	Thông tin người tạo Mảng chứa id các bài viết Chủ đề
Publication	title citation conference pages publisher description author year	Tiêu đề xuất bản Trích dẫn Hội nghị Số trang Nhà xuất bản Mô tả Tác giả Năm Xuất bản

4.2 Xây dựng ứng dụng

4.2.1 Thư viện và công cụ sử dụng

Sau đây là danh sách các thư viện và công cụ sử dụng để thực hiện xây dựng ứng dụng. Chi tiết được thể hiện trong bảng 4.7:

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Visual Studio Code	https://code.visualstudio.com/
Test API	Postman	https://www.postman.com/
Thiết kế hệ thống	AstahUML	https://astah.net/
Thiết kế cơ sở dữ liệu	Dataedo	https://dataedo.com/

Bảng 4.7: Danh sách thư viện và công cụ sử dụng

4.2.2 Kết quả đạt được

Sau 4 tháng phát triển và tìm hiểu, trang mạng xã hội dành cho các nhà nghiên cứu hoàn thành phiên bản đầu tiên, phiên bản đầu tiên này tập trung triển khai những yêu cầu và những chức năng như đã phân tích ở Chương 2. Phiên bản đầu này đã đầy đủ các chức năng chính, những chức năng cơ sở để có thể phát triển cho những phiên bản tiếp theo.

Bảng 4.8 dưới đây là thông kê về thông tin của project

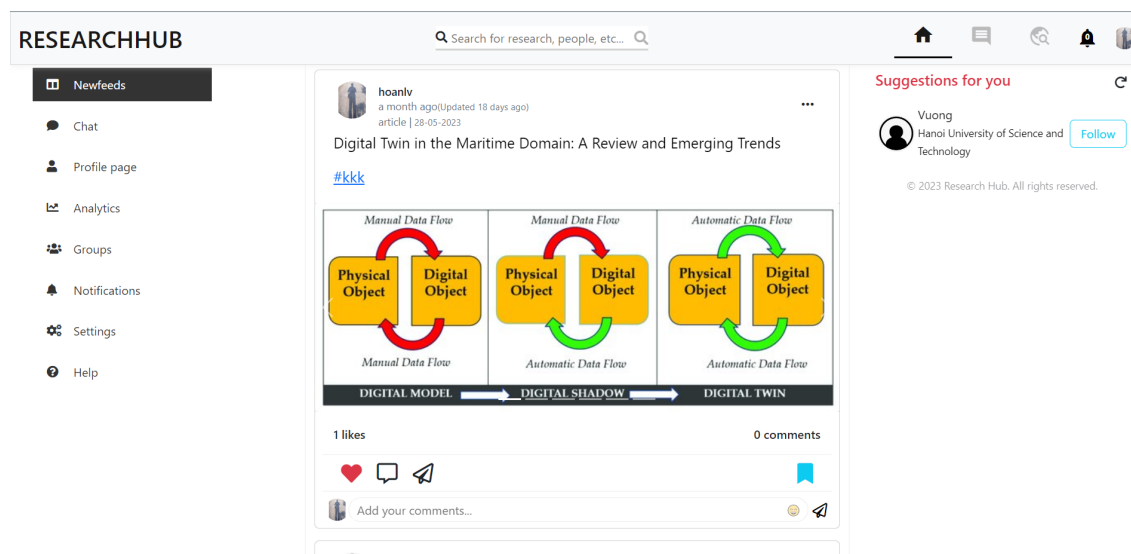
Thông tin	Thông kê
Tổng dung lượng dự án	1.01GB
Tổng số dòng mã nguồn	56780 dòng
Số lớp quản lý	6
Số lớp UI	10
Số lớp đối tượng	8
Số lớp dữ liệu	8

Bảng 4.8: Thông tin project

4.2.3 Minh họa các chức năng chính

Các hình ảnh dưới đây là minh họa cho một số chức năng chính của trang web

Màn hình chính của trang web



Hình 4.10: Màn hình chính

Hình 4.10 là màn hình chính của trang web bao gồm các phần chính sau:

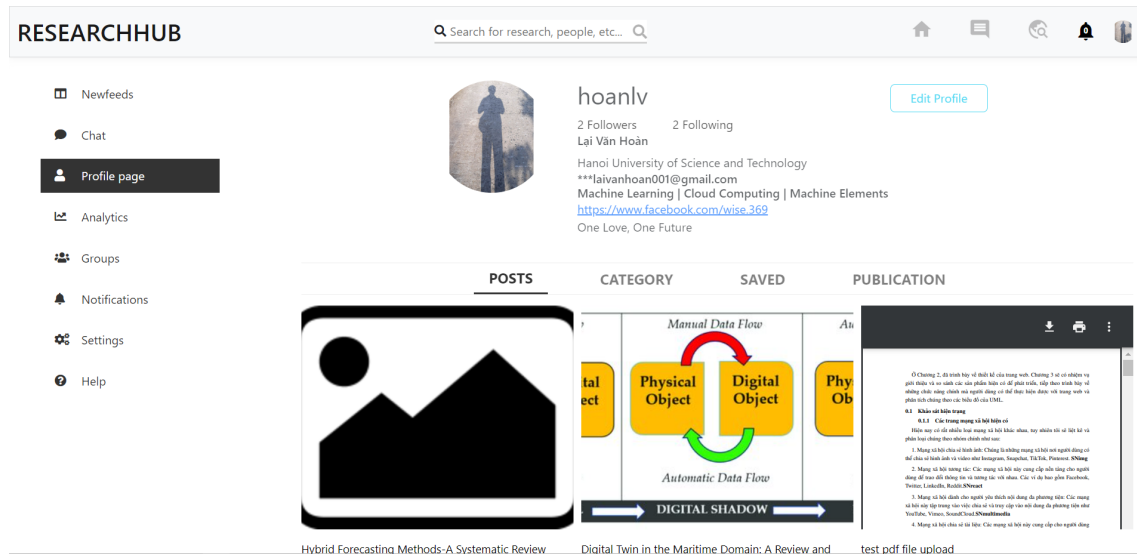
Header: chứa logo trang web, chức năng tìm kiếm trong trang, các menu điều hướng cho trang: home page, chat, discover page, notify và profile page.

Left-side-bar: chứa các thanh điều hướng đến các phần khác trong trang web. đây là một phần mở rộng cho phiên bản sau của trang web hướng đến việc cá nhân hóa cho việc sử dụng trang web.

Main: chứa thanh tạo bài viết mới, các bài đăng thuộc home page(các bài đăng của người dùng đã theo dõi). Trong các bài đăng chứa các thông tin như sau: tác giả đăng bài, thời gian đăng bài, thể loại, ngày công bố, nội dung và hashtag(sử dụng để tìm kiếm các bài đăng có cùng hashtag).

Right-side-bar: thể hiện danh sách người dùng có thể theo dõi(người dùng có thể theo dõi dựa trên thông tin tổ chức(institution) và kỹ năng(skill) của người dùng hiện tại)

Màn hình trang cá nhân người dùng



Hình 4.11: Màn hình trang cá nhân người dùng

Hình 4.12 thể hiện các thông tin người dùng của hệ thống

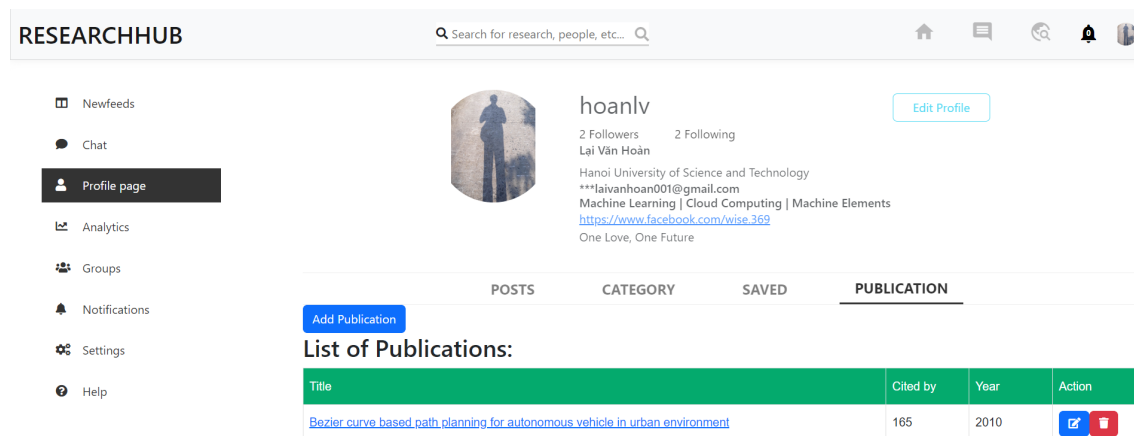
Thông tin của người dùng hiển thị ở phần "info" và "profile tab". Thông tin user chứa các thông tin mà người dùng cung cấp có thể được chỉnh sửa theo ý muốn. Các phần trong "profile tab" hiển thị các bài đăng, bài đã lưu, các chủ đề mà người dùng có thể tùy chỉnh, và các bài báo đã được công bố mà người dùng có thể thêm vào.

Profile tab bao gồm 4 tab: các bài đăng đã upload(posts), các bài đăng đã lưu(saved), chủ đề(category), báo cáo khoa học đã công bố(publication)

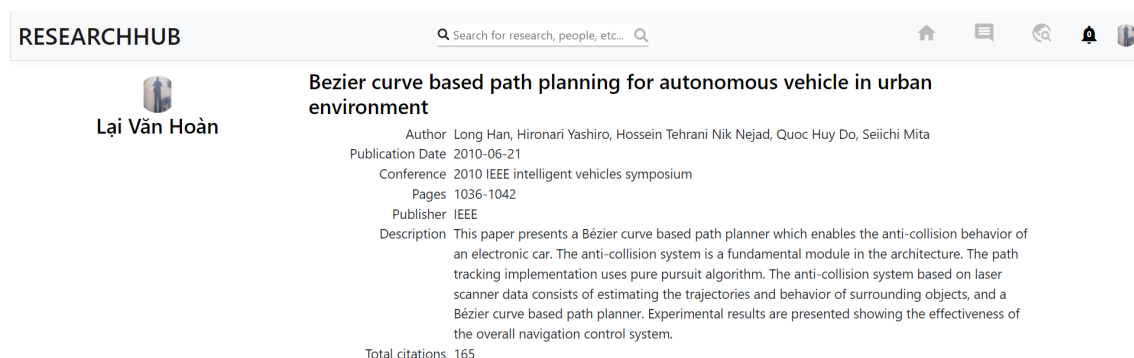
Mỗi tab là một phần chi tiết khác nhưng trong phần này tôi chỉ trình bày tab nổi bật là publication tab:

Màn hình trang cá nhân người dùng(publicaiton tab)

Bao gồm 2 phần: Nút thêm bài báo mới(chỉ có người sở hữu mới được đăng tải tại mục này) còn hiển thị với những người dùng khác thì chỉ hiển thị danh sách các bài đăng dưới dạng bảng. Phần 2 hiển thị danh sách các bài báo(liên kết tới mục chi tiết của báo cáo). Hình ảnh 4.12 là màn hình trang cá nhân người dùng ở publication tab



Hình 4.12: Màn hình trang cá nhân người dùng(publication tab)



Hình 4.13: Màn hình hiển thị chi tiết báo cáo

Hình 4.13 là màn hình hiển thị chi tiết báo cáo bao gồm thông tin chi tiết về báo cáo khoa học của người dùng và thông tin của người dùng đăng tải

4.3 Kiểm thử

STT	Trường hợp	Quy trình kiểm thử	Kết quả mong đợi	Kết quả
1	Đăng nhập	Kiểm tra việc đăng nhập thành công với tên người dùng và mật khẩu hợp lệ	Người dùng được đăng nhập thành công vào hệ thống	Đạt
2	Đăng nhập	Kiểm tra việc đăng nhập thất bại với tên người dùng và/hoặc mật khẩu không chính xác	Người dùng không thể đăng nhập với tên người dùng và/hoặc mật khẩu không chính xác	Đạt
3	Tạo bài đăng	Kiểm tra việc tạo bài đăng thành công với nội dung hợp lệ	Bài đăng được tạo thành công và hiển thị trên trang web	Đạt
4	Tạo bài đăng	Kiểm tra việc tạo bài đăng thất bại khi không có nội dung	Hệ thống không cho phép tạo bài đăng khi không có nội dung	Đạt
5	Tìm kiếm người dùng hoặc bài đăng	Kiểm tra việc tìm kiếm người dùng thành công với từ khóa hợp lệ	Hệ thống hiển thị kết quả tìm kiếm chính xác với từ khóa hợp lệ	Đạt
6	Tìm kiếm người dùng hoặc bài đăng	Kiểm tra việc tìm kiếm người dùng không thành công khi không có kết quả phù hợp	Hệ thống hiển thị thông báo không tìm thấy kết quả phù hợp khi không có người dùng phù hợp với từ khóa tìm kiếm	Đạt
7	Chỉnh sửa các thông tin về các bài thuộc quyền sở hữu của người dùng	Xác thực người dùng có quyền chỉnh sửa các chi tiết mà người dùng có quyền mới có thể tác động đến.	Hệ thống kiểm tra quyền(xác thực người dùng có sở hữu nó không mới được chỉnh sửa	Đạt
8	Thông báo cho người theo dõi khi có bài đăng mới	Kiểm tra việc gửi thông báo cho người theo dõi mỗi khi có bài đăng mới từ người dùng mà họ đang theo dõi	Người theo dõi nhận được thông báo về bài đăng mới	Đạt
9	Thông báo cho người theo dõi khi có bình luận mới	Kiểm tra việc gửi thông báo cho người theo dõi mỗi khi có bình luận mới trên bài đăng mà họ đã theo dõi	Người theo dõi nhận được thông báo về bình luận mới	Đạt

Bảng 4.9: Các trường hợp kiểm thử cho trang web mạng xã hội chia sẻ báo cáo khoa học

4.4 Triển khai

Trang web được xây dựng trên nền tảng website, và đang được phát hành triển khai trên server miễn phí render.com với link đến trang web như sau: **research-hubbe.onrender.com**

Bảng 4.10 dưới đây là yêu cầu cấu hình hệ thống tối thiểu

Linh kiện	Cấu hình
Hệ điều hành	Windows / MacOS / Linux / Android
Vi xử lý	1.8 GHz Dual-Core 64-bit CPU
Dung lượng RAM	2GB RAM
Đồ họa	Không yêu cầu đồ họa đặc biệt
Bộ nhớ	200MB dung lượng trống

Bảng 4.10: Cấu hình tối thiểu để truy cập trang web

Sau khi triển khai và thu thập ý kiến phản hồi về quy cách sử dụng cũng như trải nghiệm người dùng. Có thể tổng hợp các ý kiến phản hồi thành các ý chính như sau:

Tiếp nhận thông tin và đăng tải nhanh chóng tiện lợi: Trang web cho phép người dùng dễ dàng đăng tải và chia sẻ báo cáo khoa học một cách nhanh chóng và tiện lợi. Quy trình đăng tải bài đăng được thiết kế đơn giản và trực quan, giúp người dùng tiết kiệm thời gian trong việc chia sẻ kiến thức của họ với cộng đồng.

Tìm kiếm tài liệu nhanh chóng: Chức năng tìm kiếm trên trang web hoạt động hiệu quả và nhanh chóng. Người dùng có thể dễ dàng tìm kiếm các báo cáo khoa học cụ thể hoặc từ khóa liên quan đến lĩnh vực nghiên cứu của họ.

Tỉ lệ phản hồi nhanh dựa vào socket: Triển khai socket cho phép trang web cung cấp tỉ lệ phản hồi nhanh chóng đối với các hoạt động trực tuyến như bình luận, thảo luận và thông báo. Điều này tạo ra trải nghiệm người dùng tích cực và giúp duy trì tính tương tác cao trong cộng đồng người dùng.

Bên cạnh đó giao diện hiển thị trên mobile cần chú ý: Do thời gian chuẩn bị gấp rút, giao diện của trang web khi truy cập từ thiết bị di động cần được cải thiện. Người dùng ghi nhận rằng giao diện hiện tại có thể không hoàn toàn tương thích với các thiết bị di động, dẫn đến trải nghiệm người dùng không tối ưu.

Những ý kiến phản hồi tích cực và tiêu cực từ người dùng sẽ được sử dụng để nâng cao chất lượng và tính năng của trang web, để đáp ứng tốt hơn nhu cầu chia sẻ tài liệu và tạo môi trường mạng xã hội hữu ích cho cộng đồng nhà nghiên cứu khoa học.

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Trong chương trước, chúng ta đã thiết kế các thành phần, giao diện và kết quả đạt được trong quá trình phát triển. Trong chương này, chúng ta sẽ trình bày về những khó khăn gặp phải và cách giải quyết.

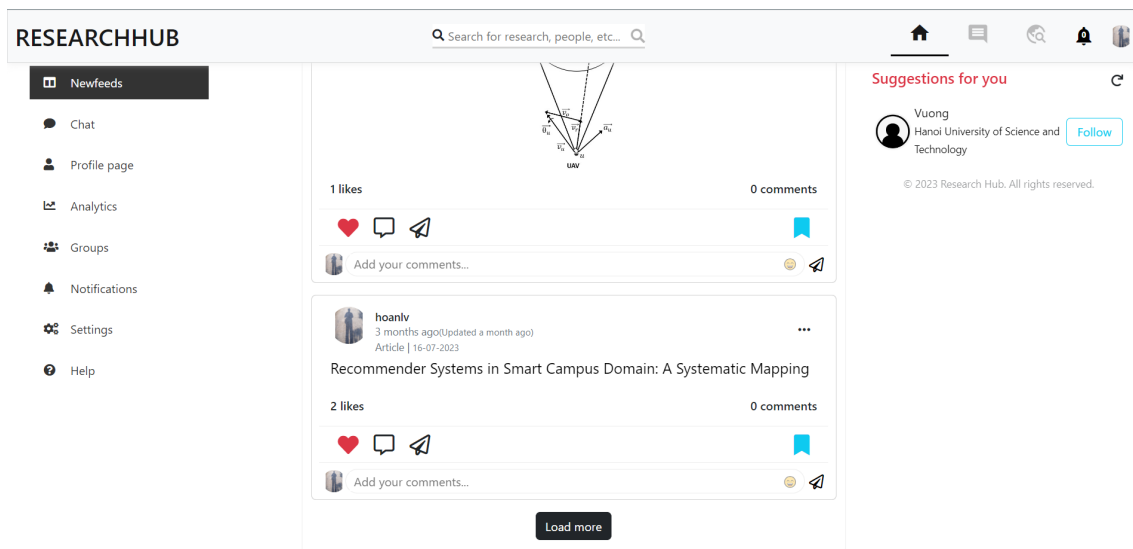
5.1 Hiệu năng trang web

5.1.1 Vấn đề gặp phải về tốc độ trang web

Trong quá trình phát triển đồ án, chúng ta đã gặp vấn đề về tốc độ tải trang web do chưa áp dụng phân trang cho trang home ban đầu. Ban đầu, trang home hiển thị toàn bộ bài viết, gây ra tải về một lượng lớn dữ liệu cùng một lúc, làm chậm tốc độ tải trang và ảnh hưởng đến trải nghiệm người dùng.

5.1.2 Quá trình giải quyết

Để giải quyết vấn đề này, chúng ta đã thực hiện phân trang cho trang home. Thay vì hiển thị toàn bộ bài viết, chúng ta chỉ hiển thị một số bài viết và tải thêm khi người dùng cuộn xuống cuối trang. Điều này giới hạn số lượng bài viết hiển thị ban đầu, giảm thiểu lượng dữ liệu cần tải về và tăng tốc độ tải trang.



Hình 5.1: Hình ảnh tải danh sách bài viết đã được phân trang

5.1.3 Kết quả

Việc áp dụng phân trang cho trang home và các component khác đã giúp giải quyết vấn đề về tốc độ tải trang chậm và cải thiện trải nghiệm người dùng. Chúng ta đã đạt được các kết quả sau:

Tăng tốc độ tải trang: Giới hạn số lượng bài viết hiển thị ban đầu giúp giảm thời gian tải trang, cải thiện trải nghiệm người dùng.

Tiết kiệm tài nguyên mạng: Chỉ tải về một số bài viết ban đầu giúp giảm lượng dữ liệu truyền từ máy chủ đến trình duyệt, tiết kiệm băng thông và giảm tải cho máy chủ.

Tối ưu hóa trang web: Cách phân trang giúp tối ưu hóa mã nguồn và tài nguyên của trang web. Chỉ tải về những bài viết cần thiết giúp trang web trở nên nhẹ nhàng và dễ quản lý.

5.2 Khó khăn khi triển khai chức năng search với MongoDB Text Search

Trong quá trình triển khai chức năng search cho ứng dụng mạng xã hội, chúng ta đã gặp phải một số khó khăn khi sử dụng MongoDB Text Search. MongoDB Text Search là một tính năng mạnh mẽ cho phép chúng ta tìm kiếm văn bản trong các trường có chỉ mục văn bản. Tuy nhiên, việc triển khai và sử dụng chức năng này cũng đòi hỏi sự quan tâm đến một số điểm quan trọng.

5.2.1 Khó khăn gặp phải

Khả năng tìm kiếm chính xác

Mặc dù MongoDB Text Search cung cấp khả năng tìm kiếm văn bản dựa trên chỉ mục, nhưng việc đảm bảo tính chính xác của kết quả tìm kiếm không phải lúc nào cũng dễ dàng. MongoDB sử dụng các quy tắc ngữ pháp và ngôn ngữ để xác định các từ khóa và truy vấn liên quan đến kết quả tìm kiếm. Tuy nhiên, điều này có thể dẫn đến kết quả không chính xác hoặc không đáp ứng được nhu cầu tìm kiếm của người dùng.

Hiệu suất và tối ưu hóa

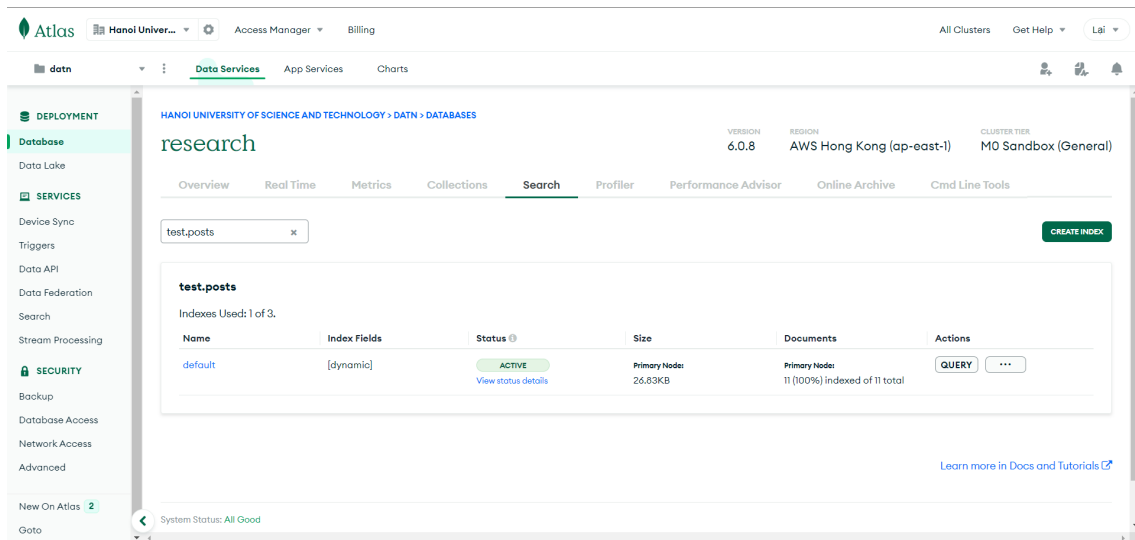
Khi triển khai chức năng search với MongoDB Text Search, việc hiệu suất và tối ưu hóa cũng là một thách thức. Với số lượng bài viết và dữ liệu người dùng lớn, việc tìm kiếm văn bản có thể ảnh hưởng đến hiệu suất của cơ sở dữ liệu và ứng dụng. Điều này đặc biệt quan trọng khi ứng dụng phải đáp ứng cho hàng ngàn yêu cầu tìm kiếm cùng lúc.

5.2.2 Giải pháp

Để giải quyết các khó khăn trên, tôi đã áp dụng giải pháp sau:

Tối ưu hóa chỉ mục

Để cải thiện hiệu suất của truy vấn tìm kiếm, tôi đã tối ưu hóa chỉ mục văn bản trong MongoDB như trong hình 5.2. MongoDB cung cấp các tùy chọn chỉ mục phù hợp cho từng trường văn bản, như đánh trọng số, điều kiện tìm kiếm và ngôn ngữ. Chúng ta đã chọn các tùy chọn chỉ mục phù hợp để đảm bảo tối ưu hóa truy vấn tìm kiếm và giảm thiểu thời gian tìm kiếm.



Hình 5.2: Thực hiện đánh index cho trường **content** trong MongoDB để sử dụng text search

5.2.3 Kết quả

Nhờ áp dụng các giải pháp trên, chúng ta đã cải thiện được tính chính xác và hiệu suất của chức năng tìm kiếm trong ứng dụng. Người dùng có thể tìm kiếm văn bản chính xác và nhanh chóng, đáp ứng tốt nhu cầu tìm kiếm của họ. Đồng thời, việc tối ưu hóa chỉ mục và tiền xử lý dữ liệu giúp ứng dụng hoạt động ổn định và hiệu quả, đáp ứng được lượng truy vấn lớn từ người dùng.

Trên đây là phần trình bày về các khó khăn gặp phải trong quá trình xây dựng đồ án và cách giải quyết. Chương tiếp theo, cũng là chương cuối cùng, sẽ là lời kết luận cho quá trình xây dựng đồ án.

5.3 Tăng trải nghiệm người dùng với Socket.IO

5.3.1 Vấn đề cập nhật thông báo

Trong quá trình phát triển ứng dụng, một yếu tố quan trọng để cải thiện trải nghiệm người dùng là đảm bảo cập nhật thông báo ngay lập tức theo thời gian thực. Điều này giúp người dùng nhận được thông báo mới nhất và phản hồi nhanh chóng đối với các tương tác từ người dùng khác.

5.3.2 Giải pháp

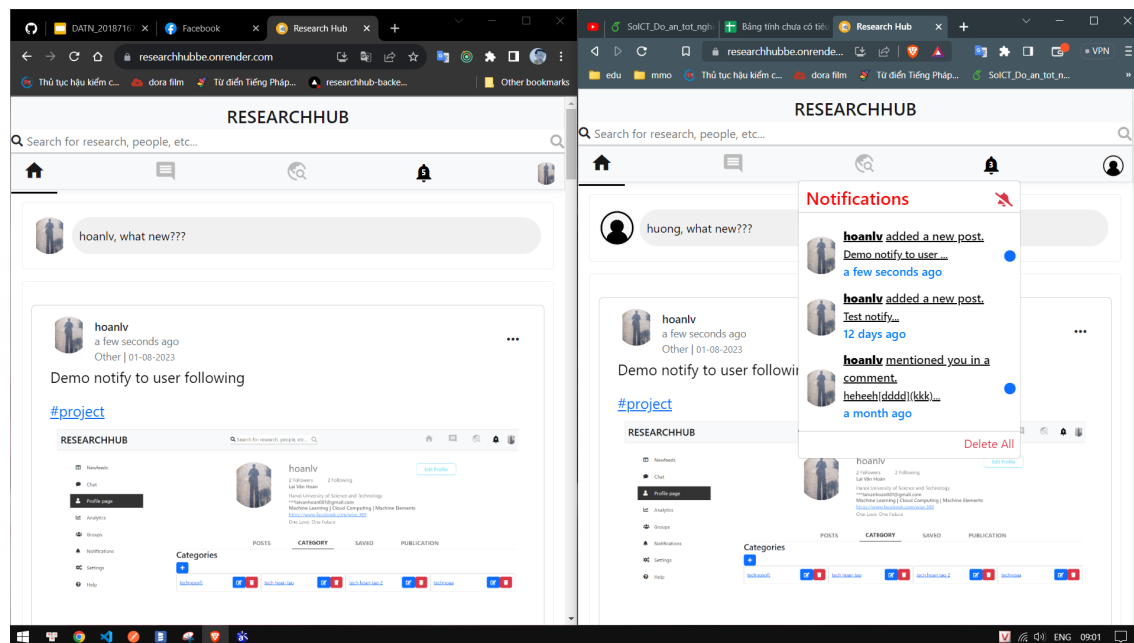
Để giải quyết vấn đề này, chúng ta đã sử dụng thư viện Socket.IO trong ứng dụng. Socket.IO là một thư viện JavaScript cho phép truyền thông tin hai chiều và xây dựng ứng dụng theo thời gian thực, hoạt động dựa trên giao thức WebSocket.

Khi có thông báo mới (ví dụ: bài viết mới, bình luận, lượt thích), máy chủ gửi thông điệp tới các máy khách đã kết nối thông qua Socket.IO. Các máy khách nhận thông điệp và cập nhật giao diện người dùng ngay lập tức để hiển thị thông báo

mới. Điều này đảm bảo người dùng có thể nhìn thấy thông báo mới mà không cần làm mới trang hoặc thực hiện thao tác thủ công.

5.3.3 Kết quả

Hình 5.3 là kết quả đạt được sau khi sử dụng **socket.io** để nhận thông báo ngay lập tức khi người mà người dùng đang theo dõi đăng tải bài viết mới trên trang cá nhân của họ



Hình 5.3: Hình ảnh thông báo thời gian thực khi sử dụng socket.io

Sử dụng Socket.IO để cung cấp cập nhật thông báo theo thời gian thực đã tăng trải nghiệm người dùng trong ứng dụng mạng xã hội. Người dùng có thể nhận được thông báo mới ngay lập tức và tương tác nhanh chóng với người dùng khác trên nền tảng. Điều này tạo ra một trải nghiệm tốt hơn và thu hút người dùng hoạt động nhiều hơn trên ứng dụng.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong chương trước đã trình bày về những vấn đề gặp phải khi xây dựng đồ án và quá trình giải quyết vấn đề. Chương cuối này sẽ là lời kết luận cho đồ án, những gì đã làm được và định hướng.

6.1 Kết luận

Trong suốt quá trình thực hiện đồ án web chia sẻ báo cáo nghiên cứu khoa học, chúng tôi đã có những thành tựu đáng kể nhờ vào sự định hướng và chỉ dẫn tận tâm của TS. Đỗ Quốc Huy. Đồ án này đã giúp chúng tôi học hỏi và áp dụng những kiến thức và kỹ năng trong lĩnh vực phát triển trang web.

Trước tiên, chúng tôi đã thiết kế và triển khai các thành phần cần thiết cho trang web chia sẻ báo cáo nghiên cứu. Chúng tôi đã tạo ra giao diện người dùng hấp dẫn và thân thiện, cho phép người dùng tìm kiếm, xem và tải xuống các báo cáo nghiên cứu một cách dễ dàng.

Chúng tôi đã đạt được mục tiêu về hiệu năng trang web thông qua việc tối ưu hóa và cải thiện tốc độ tải trang. Những giải pháp như tối ưu hóa hình ảnh, sử dụng bộ nhớ cache và sử dụng công nghệ CDN đã giúp trang web hoạt động một cách mượt mà và nhanh chóng.

Tuy nhiên, trong quá trình phát triển, chúng tôi cũng đã gặp phải một số khó khăn. Nhưng nhờ vào sự nỗ lực và kiên nhẫn, chúng tôi đã tìm ra các giải pháp thích hợp để vượt qua những khó khăn đó. Quá trình giải quyết vấn đề đã giúp chúng tôi hiểu rõ hơn về cách hoạt động và tối ưu hóa trang web.

Tổng kết lại, đồ án website chia sẻ báo cáo nghiên cứu khoa học đã mang lại cho chúng tôi nhiều kiến thức và kinh nghiệm trong việc phát triển trang web. Chúng tôi rất biết ơn TS. Đỗ Quốc Huy vì sự hướng dẫn và động viên của thầy trong suốt quá trình thực hiện đồ án. Đồ án đã đạt được kết quả như mong đợi và góp phần nâng cao trải nghiệm người dùng trong việc chia sẻ và truy cập các báo cáo nghiên cứu khoa học.

6.2 Hướng phát triển

Trong phần này, tôi sẽ trình bày về định hướng công việc trong tương lai để hoàn thiện sản phẩm và nghiên cứu của mình.

6.2.1 Hoàn thiện các chức năng/nhiệm vụ đã làm

Tối ưu hóa hiệu năng: Tiếp tục nghiên cứu và áp dụng các phương pháp tối ưu hóa để cải thiện hiệu suất và tốc độ tải trang. Kiểm tra và điều chỉnh cấu hình máy

chủ, cơ sở dữ liệu, và cơ chế caching để đảm bảo trang web hoạt động một cách hiệu quả và nhanh chóng.

Tăng cường tính bảo mật: Nghiên cứu và triển khai các biện pháp bảo mật để đảm bảo an toàn thông tin trên trang web. Áp dụng các kỹ thuật mã hóa, kiểm tra đầu vào, và quản lý phiên để ngăn chặn các cuộc tấn công và lợi dụng lỗ hổng bảo mật.

Cải thiện giao diện người dùng: Tiếp tục tối ưu hóa giao diện người dùng để mang lại trải nghiệm tốt hơn cho người dùng. Tích hợp các tính năng tương tác, cải thiện tính thẩm mỹ và đảm bảo tính responsive trên các thiết bị khác nhau.

6.2.2 Phân tích các hướng đi mới

Phát triển tính năng phân tích dữ liệu: Xây dựng và triển khai các công cụ và thuật toán phân tích dữ liệu để hiểu và khai thác thông tin từ các báo cáo nghiên cứu. Áp dụng các phương pháp học máy và khai phá dữ liệu để phân loại, tóm tắt và tìm ra mối quan hệ giữa các báo cáo.

Từ những công việc cần thiết để hoàn thiện chức năng đã làm và các hướng đi mới, tôi hy vọng rằng sản phẩm và nghiên cứu của mình sẽ tiếp tục phát triển và đóng góp vào việc chia sẻ và truyền đạt kiến thức khoa học một cách hiệu quả và tiện lợi.

Trên đây là phần trình bày về kết luận những gì làm được, chưa làm được, những điều đã học hỏi được qua quá trình xây dựng đồ án. Cùng với đó là những hướng phát triển cho đồ án trong tương lai. Đây cũng là chương cuối cùng và là sự tổng kết cho báo cáo và toàn bộ đồ án.

TÀI LIỆU THAM KHẢO

- [1] Javascript, *Javascript documentation*. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (visited on 07/21/2023).
- [2] React, *React documentation*. [Online]. Available: <https://legacy.reactjs.org/docs/getting-started.html> (visited on 07/21/2023).
- [3] Redux, *Redux package documentation*. [Online]. Available: <https://www.npmjs.com/package/redux> (visited on 07/21/2023).
- [4] Axios, *Axios package documentation*. [Online]. Available: <https://www.npmjs.com/package/axios> (visited on 03/21/2023).
- [5] Node.js, *Node.js documentation*. [Online]. Available: <https://nodejs.org/en/docs> (visited on 07/21/2023).
- [6] Express, *Express package documentation*. [Online]. Available: <https://www.npmjs.com/package/express> (visited on 07/21/2023).
- [7] MongoDB, *Mongodb documentation*. [Online]. Available: <https://www.mongodb.com/docs/> (visited on 07/21/2023).
- [8] Render, *Render hosting docs and intergration*. [Online]. Available: <https://socket.io/docs/v4/> (visited on 07/21/2023).