

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Xây dựng hệ thống tìm kiếm và cho thuê chung cư mini

NGUYỄN ĐÌNH SƠN

son.nd194363@sis.hust.edu.vn

Ngành Kỹ thuật máy tính

Giảng viên hướng dẫn: TS. Phạm Huy Hoàng

Chữ ký GVHD

Khoa: Kỹ thuật máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 01/2025

LỜI CẢM ƠN

Lời đầu tiên, em xin bày tỏ lòng biết ơn sâu sắc tới thầy TS. Phạm Huy Hoàng, giảng viên hướng dẫn. Thầy đã tận tình chỉ bảo, hỗ trợ em rất nhiều trong suốt quá trình học tập và hoàn thiện đồ án tốt nghiệp này.

Em xin gửi lời cảm ơn chân thành tới quý thầy cô trường Đại học Bách Khoa Hà Nội và Trường Công nghệ thông tin Truyền thông. Nhờ có sự dùu dắt tận tình của thầy cô, em đã được trang bị những kiến thức nền tảng quý báu, góp phần quan trọng vào việc hoàn thành đồ án này.

Em cũng muốn gửi lời tri ân tới gia đình và bạn bè, những người luôn kề vai sát cánh, động viên và giúp đỡ em vượt qua mọi khó khăn trong quá trình thực hiện đồ án cũng như trong quá trình học tập tại trường Đại học Bách Khoa Hà Nội.

Em hiểu rằng đồ án này chắc chắn vẫn còn những thiếu sót do thời gian và kiến thức bản thân còn hạn chế. Vì vậy, em rất mong nhận được những góp ý quý báu từ quý thầy cô để có thể hoàn thiện bản thân hơn nữa, trau dồi thêm kiến thức và kỹ năng, sẵn sàng áp dụng vào công việc thực tế sau này.

Em xin chân thành cảm ơn!

TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong thời đại công nghệ thông tin phát triển như vũ bão hiện nay, cho thuê nhà online đang dần trở thành một xu hướng tất yếu. So với cách thức cho thuê nhà truyền thống, hình thức online mang đến nhiều lợi ích vượt trội như giúp em tiết kiệm thời gian, tiếp cận được với nhiều khách hàng tiềm năng hơn, đồng thời cung cấp thông tin về căn hộ một cách đầy đủ và chi tiết.

Em nhận thấy nhu cầu tìm kiếm nhà ở tại các thành phố lớn đang ngày một tăng cao, nhất là đối với những căn hộ chung cư mini. Chính vì vậy, em đã quyết định lựa chọn đề tài "Xây dựng hệ thống tìm kiếm và cho thuê chung cư mini" cho đồ án tốt nghiệp của mình.

Hệ thống này được em thiết kế để tập trung giải quyết các vấn đề then chốt như: giúp người dùng tìm kiếm căn hộ phù hợp với nhu cầu của họ, tạo điều kiện cho chủ nhà đăng tin cho thuê/bán căn hộ và quản lý các tin đăng một cách hiệu quả. Để xây dựng hệ thống, em đã sử dụng những công nghệ tiên tiến như NodeJS, ExpressJS, MongoDB cho Backend và ReactJS, Tailwind, Ant Design cho Frontend. Không chỉ dừng lại ở đó, em còn tích hợp thêm Google Map vào hệ thống để người dùng có thể dễ dàng xác định vị trí của căn hộ, cùng với các tính năng như chat online và nhiều phương thức liên lạc hiện đại khác nhằm mang lại sự tiện lợi tối đa cho cả người thuê nhà lẫn chủ nhà.

Đặc biệt, em cũng rất chú trọng vào việc tối ưu hóa giao diện người dùng và tối ưu hóa công cụ tìm kiếm (SEO) để hệ thống có thể mang lại trải nghiệm tốt nhất cho người dùng, giúp họ dễ dàng tìm thấy thông tin trên các công cụ tìm kiếm phổ biến hiện nay.

Hà Nội, ngày 6 tháng 1 năm 2025

Sinh viên thực hiện

Nguyễn Đình Sơn

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 Bố cục đồ án	2
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	3
2.1 Khảo sát hiện trạng	3
2.2 Tổng quan chức năng	3
2.2.1 Phân tích tác nhân và vai trò.....	3
2.2.2 Biểu đồ use case tổng quát	4
2.2.3 Biểu đồ use case phân rã của Người dùng.....	5
2.2.4 Biểu đồ use case phân rã của Admin	6
2.2.5 Quy trình nghiệp vụ	6
2.3 Đặc tả chức năng	6
2.3.1 Đặc tả use case "Đăng ký"	6
2.3.2 Đặc tả use case "Đăng nhập"	7
2.3.3 Đặc tả use case "Đăng xuất".....	8
2.3.4 Đặc tả use case "Quên mật khẩu"	9
2.3.5 Đặc tả use case "Cập nhật thông tin"	9
2.3.6 Đặc tả use case "Nhận thông báo"	10
2.3.7 Đặc tả use case "Tìm kiếm chung cư mini"	11
2.3.8 Đặc tả use case "Xem bài đăng"	12
2.3.9 Đặc tả use case "Xem thông tin người dùng"	12
2.3.10 Đặc tả use case "Yêu thích bài đăng"	12
2.3.11 Đặc tả use case "Chat với người dùng"	13

2.3.12 Đặc tả use case "Xem danh sách bài đăng"	14
2.3.13 Đặc tả use case "Đăng tin"	14
2.3.14 Đặc tả use case "Sửa bài đăng"	15
2.3.15 Đặc tả use case "Xóa bài đăng"	16
2.3.16 Đặc tả use case "Xem danh sách bài đăng"	17
2.3.17 Đặc tả use case "Xem danh sách bài đăng cần duyệt".....	17
2.3.18 Đặc tả use case "Tìm kiếm".....	18
2.3.19 Đặc tả use case "Xem chi tiết bài đăng"	18
2.3.20 Đặc tả use case "Duyệt bài đăng"	19
2.3.21 Đặc tả use case "Khóa bài đăng"	20
2.3.22 Đặc tả use case "Xem danh sách người dùng"	20
2.3.23 Đặc tả use case "Tìm kiếm".....	20
2.3.24 Đặc tả use case "Xem chi tiết người dùng"	21
2.3.25 Đặc tả use case "Khóa tài khoản".....	21
2.3.26 Đặc tả use case "Mở tài khoản"	22
2.4 Mô hình hóa tương tác - Các biểu đồ tuần tự	23
2.4.1 Biểu đồ tuần tự "Đăng ký"	23
2.4.2 Biểu đồ tuần tự "Đăng nhập"	24
2.4.3 Biểu đồ tuần tự "Đăng xuất"	24
2.4.4 Biểu đồ tuần tự "Quên mật khẩu"	25
2.4.5 Biểu đồ tuần tự "Cập nhật thông tin".....	25
2.4.6 Biểu đồ tuần tự "Nhận thông báo"	26
2.4.7 Biểu đồ tuần tự "Tìm kiếm chung cư mini"	26
2.4.8 Biểu đồ tuần tự "Xem bài đăng"	27
2.4.9 Biểu đồ tuần tự "Xem thông tin người dùng"	27
2.4.10 Biểu đồ tuần tự "Yêu thích bài đăng".....	28

2.4.11 Biểu đồ tuần tự "Chat với người dùng"	28
2.4.12 Biểu đồ tuần tự "Xem danh sách bài đăng"	29
2.4.13 Biểu đồ tuần tự "Đăng tin"	29
2.4.14 Biểu đồ tuần tự "Sửa bài đăng"	30
2.4.15 Biểu đồ tuần tự "Xóa bài đăng".....	30
2.4.16 Biểu đồ tuần tự "Xem danh sách bài đăng"	31
2.4.17 Biểu đồ tuần tự "Xem danh sách bài đăng cần duyệt"	31
2.4.18 Biểu đồ tuần tự "Tìm kiếm".....	32
2.4.19 Biểu đồ tuần tự "Xem chi tiết bài đăng"	32
2.4.20 Biểu đồ tuần tự "Duyệt bài đăng"	33
2.4.21 Biểu đồ tuần tự "Khóa bài đăng".....	33
2.4.22 Biểu đồ tuần tự "Xem danh sách người dùng"	34
2.4.23 Biểu đồ tuần tự "Tìm kiếm".....	34
2.4.24 Biểu đồ tuần tự "Xem chi tiết người dùng"	35
2.4.25 Biểu đồ tuần tự "Khóa tài khoản".....	35
2.4.26 Biểu đồ tuần tự "Mở tài khoản"	36
2.5 Yêu cầu phi chức năng	36
2.5.1 Yêu cầu về bảo mật	36
2.5.2 Yêu cầu về hiệu năng.....	36
2.5.3 Yêu cầu về giao diện.....	36
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	37
3.1 Các công cụ sử dụng trong hệ thống	37
3.1.1 Postman	37
3.1.2 MongoDB Compass	37
3.1.3 draw.io	37
3.2 Visual Studio Code	38

3.3 Ngôn ngữ lập trình Javascript	38
3.4 Framework Backend	39
3.4.1 NodeJS	39
3.4.2 ExpressJS.....	39
3.5 MongoDB.....	40
3.6 Framework Frontend.....	40
3.6.1 ReactJS	40
3.7 Framework Tailwind CSS.....	41
3.8 Thư viện Ant-Design	42
3.9 Google Firebase Storage.....	42
3.10 Google Map API.....	43

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG 44

4.1 Thiết kế kiến trúc.....	44
4.1.1 Lựa chọn kiến trúc phần mềm	44
4.1.2 Thiết kế tổng quan.....	46
4.2 Thiết kế cơ sở dữ liệu	49
4.2.1 Biểu đồ quan hệ thực thể (ERD)	49
4.2.2 Thiết kế Bảng Users	50
4.2.3 Thiết kế Bảng Listings.....	51
4.2.4 Thiết kế Bảng Districts	51
4.2.5 Thiết kế Bảng Cities.....	52
4.2.6 Thiết kế Bảng Notifications	52
4.2.7 Thiết kế Bảng RoomChats	52
4.2.8 Thiết kế Bảng Messages	53
4.2.9 Thiết kế Bảng ForgetPasswords	53

4.3 Thiết kế giao diện	54
4.3.1 Giao diện đăng ký	54
4.3.2 Giao diện đăng nhập.....	54
4.3.3 Giao diện trang chủ	55
4.3.4 Giao diện bài đăng	55
4.3.5 Giao diện Profile người đăng.....	56
4.3.6 Giao diện Setting người đăng	56
4.4 Xây dựng ứng dụng.....	56
4.4.1 Thư viện và công cụ sử dụng.....	56
4.4.2 Kết quả đạt được	58
4.4.3 Minh họa các chức năng chính	61
4.5 Kiểm thử.....	63
4.5.1 Kiểm thử xác thực	63
4.5.2 Kiểm thử người dùng.....	64
4.5.3 Kiểm thử admin	65
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	66
5.1 Kết luận	66
5.2 Hướng phát triển.....	66

DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ use case tổng quan	4
Hình 2.2	Phân rã use case của Người dùng	5
Hình 2.3	Phân rã use case của Admin	6
Hình 2.4	Biểu đồ tuần tự "Đăng ký"	23
Hình 2.5	Biểu đồ tuần tự "Đăng nhập"	24
Hình 2.6	Biểu đồ tuần tự "Đăng xuất"	24
Hình 2.7	Biểu đồ tuần tự "Quên mật khẩu"	25
Hình 2.8	Biểu đồ tuần tự "Cập nhật thông tin"	25
Hình 2.9	Biểu đồ tuần tự "Nhận thông báo"	26
Hình 2.10	Biểu đồ tuần tự "Tìm kiếm chung cư mini"	26
Hình 2.11	Biểu đồ tuần tự "Xem bài đăng"	27
Hình 2.12	Biểu đồ tuần tự "Xem thông tin người dùng"	27
Hình 2.13	Biểu đồ tuần tự "Yêu thích bài đăng"	28
Hình 2.14	Biểu đồ tuần tự "Chat với người dùng"	28
Hình 2.15	Biểu đồ tuần tự "Xem danh sách bài đăng"	29
Hình 2.16	Biểu đồ tuần tự "Đăng tin"	29
Hình 2.17	Biểu đồ tuần tự "Sửa bài đăng"	30
Hình 2.18	Biểu đồ tuần tự "Xóa bài đăng"	30
Hình 2.19	Biểu đồ tuần tự "Xem danh sách bài đăng"	31
Hình 2.20	Biểu đồ tuần tự "Xem danh sách bài đăng cần duyệt"	31
Hình 2.21	Biểu đồ tuần tự "Tìm kiếm"	32
Hình 2.22	Biểu đồ tuần tự "Xem chi tiết bài đăng"	32
Hình 2.23	Biểu đồ tuần tự "Duyệt bài đăng"	33
Hình 2.24	Biểu đồ tuần tự "Khóa bài đăng"	33
Hình 2.25	Biểu đồ tuần tự "Xem danh sách người dùng"	34
Hình 2.26	Biểu đồ tuần tự "Tìm kiếm"	34
Hình 2.27	Biểu đồ tuần tự "Xem chi tiết người dùng"	35
Hình 2.28	Biểu đồ tuần tự "Khóa tài khoản"	35
Hình 2.29	Biểu đồ tuần tự "Mở tài khoản"	36
Hình 4.1	Kiến trúc MVC	44
Hình 4.2	Kiến trúc hệ thống	45
Hình 4.3	Kiến trúc tổng thể	46
Hình 4.4	Biểu đồ quan hệ thực thể	49
Hình 4.5	GD-Đăng ký	54

Hình 4.6	GD-Đăng nhập	54
Hình 4.7	GD-Trang chủ	55
Hình 4.8	GD-Bài đăng	55
Hình 4.9	GD-Profile	56
Hình 4.10	GD-Profile	56
Hình 4.11	GD-Đăng ký	58
Hình 4.12	GD-Đăng nhập	59
Hình 4.13	GD-Trang chủ	59
Hình 4.14	GD-Bài đăng	60
Hình 4.15	GD-Profile	60
Hình 4.16	GD-Setting	61
Hình 4.17	Tìm kiếm bài đăng	61
Hình 4.18	Tạo bài đăng	62
Hình 4.19	Cập nhật bài đăng	62
Hình 4.20	Quản lý người dùng	63
Hình 4.21	Quản lý bài đăng	63

DANH MỤC BẢNG BIỂU

Bảng 2.1	Đặc tả use case "Đăng ký"	7
Bảng 2.2	Đặc tả use case "Đăng nhập"	8
Bảng 2.3	Đặc tả use case "Đăng xuất"	8
Bảng 2.4	Đặc tả use case "Quên mật khẩu"	9
Bảng 2.5	Đặc tả use case "Cập nhật thông tin"	10
Bảng 2.6	Đặc tả use case "Nhận thông báo"	11
Bảng 2.7	Đặc tả use case "Tìm kiếm chung cư mini"	11
Bảng 2.8	Đặc tả use case "Xem bài đăng"	12
Bảng 2.9	Đặc tả use case "Xem thông tin người dùng"	12
Bảng 2.10	Đặc tả use case "Yêu thích bài đăng"	13
Bảng 2.11	Đặc tả use case "Chat với người dùng"	13
Bảng 2.12	Đặc tả use case "Xem danh sách bài đăng"	14
Bảng 2.13	Đặc tả use case "Đăng tin"	15
Bảng 2.14	Đặc tả use case "Sửa bài đăng"	16
Bảng 2.15	Đặc tả use case "Xóa bài đăng"	17
Bảng 2.16	Đặc tả use case "Xem danh sách bài đăng"	17
Bảng 2.17	Đặc tả use case "Xem danh sách bài đăng cần duyệt"	18
Bảng 2.18	Đặc tả use case "Tìm kiếm"	18
Bảng 2.19	Đặc tả use case "Xem chi tiết bài đăng"	19
Bảng 2.20	Đặc tả use case "Duyệt bài đăng"	19
Bảng 2.21	Đặc tả use case "Khóa bài đăng"	20
Bảng 2.22	Đặc tả use case "Xem danh sách người dùng"	20
Bảng 2.23	Đặc tả use case "Tìm kiếm"	21
Bảng 2.24	Đặc tả use case "Xem chi tiết người dùng"	21
Bảng 2.25	Đặc tả use case "Khóa tài khoản"	22
Bảng 2.26	Đặc tả use case "Mở tài khoản"	22
Bảng 3.1	So sánh Tailwind CSS với các framework CSS truyền thống .	42
Bảng 4.1	Bảng dữ liệu Users	50
Bảng 4.2	Bảng dữ liệu Listings	51
Bảng 4.3	Bảng dữ liệu Districts	51
Bảng 4.4	Bảng dữ liệu Cities	52
Bảng 4.5	Bảng dữ liệu Notifications	52
Bảng 4.6	Bảng dữ liệu RoomChats	52
Bảng 4.7	Bảng dữ liệu Messages	53

Bảng 4.8	Bảng dữ liệu RoomChats	53
Bảng 4.9	Danh sách thư viện và công cụ sử dụng	57

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API	Giao diện lập trình ứng dụng (Application Programming Interface)
EUD	Phát triển ứng dụng người dùng cuối(End-User Development)
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

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong bối cảnh đô thị hóa ngày càng phát triển, nhu cầu về nhà ở tại các thành phố lớn không ngừng gia tăng. Đặc biệt, phân khúc chung cư mini đã trở thành một lựa chọn phổ biến nhờ chi phí hợp lý, vị trí thuận lợi và khả năng đáp ứng nhu cầu của nhiều đối tượng như sinh viên, người lao động và các gia đình nhỏ. Tuy nhiên, việc tìm kiếm và cho thuê chung cư mini hiện nay vẫn gặp nhiều khó khăn do thiếu một nền tảng chuyên biệt hỗ trợ người dùng.

Người thuê nhà thường phải dành nhiều thời gian để tra cứu thông tin trên các trang, hội nhóm facebook hoặc thông qua các kênh truyền thống như môi giới, vốn kốn kém và không minh bạch. Ngược lại, chủ nhà cũng gặp khó khăn trong việc tiếp cận đối tượng thuê phù hợp và quảng bá hiệu quả thông tin về căn hộ của mình. Điều này dẫn đến sự lãng phí về thời gian, nguồn lực và giảm hiệu quả trong quá trình tìm kiếm và cho thuê nhà.

Xuất phát từ thực tế này, việc xây dựng một hệ thống tìm kiếm và cho thuê chung cư mini thông minh, tiện lợi, và thân thiện với người dùng trở nên cần thiết. Hệ thống sẽ không chỉ giúp kết nối trực tiếp giữa người thuê và chủ nhà mà còn cung cấp các chức năng hỗ trợ như tìm kiếm theo tiêu chí, đánh giá, xác thực thông tin, nhằm tạo ra một môi trường minh bạch và hiệu quả hơn. Hệ thống được tích hợp thêm Google map giúp người dùng xác định rõ vị trí căn hộ và tính năng chat với các người dùng/nhà môi giới khác nhằm tăng khả năng kết nối.

Với những lý do trên, em quyết định chọn đề tài "Xây dựng hệ thống tìm kiếm và cho thuê chung cư mini" làm đề tài cho đồ án tốt nghiệp của mình. Đề tài được thực hiện với mục tiêu tạo ra một nền tảng trực tuyến hiện đại, dễ sử dụng, và tối ưu hóa trải nghiệm của người dùng và các nhà môi giới.

1.2 Mục tiêu và phạm vi đề tài

Mục tiêu chung của đồ án: thiết kế, xây dựng hệ thống tìm kiếm và cho thuê chung cư mini với giao diện, tính năng mới nhằm tối ưu cho người dùng và các nhà môi giới/chủ nhà trên cả nước

Những mục tiêu chính của hệ thống:

- Xây dựng một hệ thống có giao diện thân thiện, dễ dùng, trực quan bằng các công nghệ xây dựng giao diện mới nhất
- Tối ưu hóa chức năng cho người dùng từ việc tìm kiếm căn hộ, thêm vào danh

sách yêu thích đến việc đăng tin và quản lý bài đăng của mình

- Hệ thống được xây dựng trên nền tảng bảo mật tốt, đảm bảo an toàn thông tin cho người sử dụng

1.3 Bố cục đồ án

Đồ án tốt nghiệp của em bao gồm 5 chương:

Chương 1: Giới thiệu đề tài, chương này giới thiệu và xác định mục tiêu phạm vi cho đề tài, cung cấp cho người dùng cái nhìn tổng quan về hệ thống tìm kiếm và cho thuê chung cư mini cũng như các tài liệu mô tả, tài liệu liên quan về hệ thống.

Chương 2: Khảo sát và phân tích yêu cầu phần mềm, chương này trình bày use case tổng quan và các nhóm use case của hệ thống, đặc tả chức năng và các yêu cầu phi chức năng, mô hình hóa cấu trúc hệ thống, mô hình hóa sự tương tác.

Chương 3: Công nghệ sử dụng, chương này giới thiệu các công nghệ sử dụng cho hệ thống tìm kiếm và cho thuê chung cư mini từ framework Backend đến Frontend cùng các thư viện bên thứ ba.

Chương 4: Thiết kế, xây dựng và thử nghiệm hệ thống, chương này mô tả kiến trúc tổng thể của hệ thống, thực hiện thiết kế các lớp, thiết kế cơ sở dữ liệu của hệ thống, thiết kế giao diện, xây dựng hệ thống rồi đi đến kiểm thử, đánh giá hiệu năng chương trình.

Chương 5: Kết luận và hướng phát triển, chương này trình bày tóm lược về đồ án tốt nghiệp, những kết quả đạt được và nêu lên hướng phát triển trong tương lai đối với đồ án hiện tại.

Cuối cùng là danh mục tài liệu tham khảo.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

2.1 Khảo sát hiện trạng

Hiện nay, nhu cầu thuê và cho thuê nhà ở, đặc biệt là chung cư mini, đang ngày càng gia tăng tại các thành phố lớn. Với ưu điểm chi phí hợp lý, vị trí thuận lợi, chung cư mini trở thành lựa chọn phổ biến của sinh viên, người lao động và các gia đình nhỏ. Tuy nhiên, quá trình tìm kiếm và cho thuê vẫn gặp nhiều khó khăn.

Các hình thức tìm kiếm truyền thống như qua môi giới, tờ rơi hoặc các nhóm trên mạng xã hội như facebook thường gây tốn thời gian, thiếu sự minh bạch, chưa trực quan hóa cho người dùng để dễ dàng so sánh. Các nền tảng trực tuyến hiện nay tuy đã hỗ trợ người dùng đăng tin và tìm kiếm bất động sản nhưng chưa tối ưu hóa cho phân khúc chung cư mini. Người thuê nhà khó tìm kiếm theo các tiêu chí chi tiết như giá cả, diện tích, tiện ích, và thông tin bài đăng đôi khi không được xác thực, dẫn đến nguy cơ gặp phải thông tin sai lệch.

Người dùng hiện nay thường ưa chuộng các hệ thống có giao diện thân thiện dễ sử dụng và có tích hợp nhiều tính năng hỗ trợ như tìm kiếm theo các tiêu chí, khả năng tương tác trực tiếp với nhà môi giới như chat, gọi zalo.

Từ những khảo sát chung về hiện trạng nhu cầu người dùng và hệ thống tìm kiếm chung cư như trên, em đã xác định được một số tính năng phần mềm quan trọng cần phát triển trong đồ án tốt nghiệp này:

- Cho phép người dùng đăng nhập, đăng ký vào hệ thống, ngoài ra chức năng quên mật khẩu cho phép người dùng lấy lại mật khẩu.
- Tìm kiếm chung cư mini theo các tiêu chí (tên bài, địa điểm, diện tích, giá cả, tiện ích)
- Xem thông tin chi tiết bài đăng và thông tin người đăng bài, tích hợp tính năng yêu thích để lưu bài đăng và tính năng chat để liên hệ giữa các người dùng.
- Đăng tin cho thuê chung cư, quản lý bài đăng tin các chung cư cho người dùng.

2.2 Tổng quan chức năng

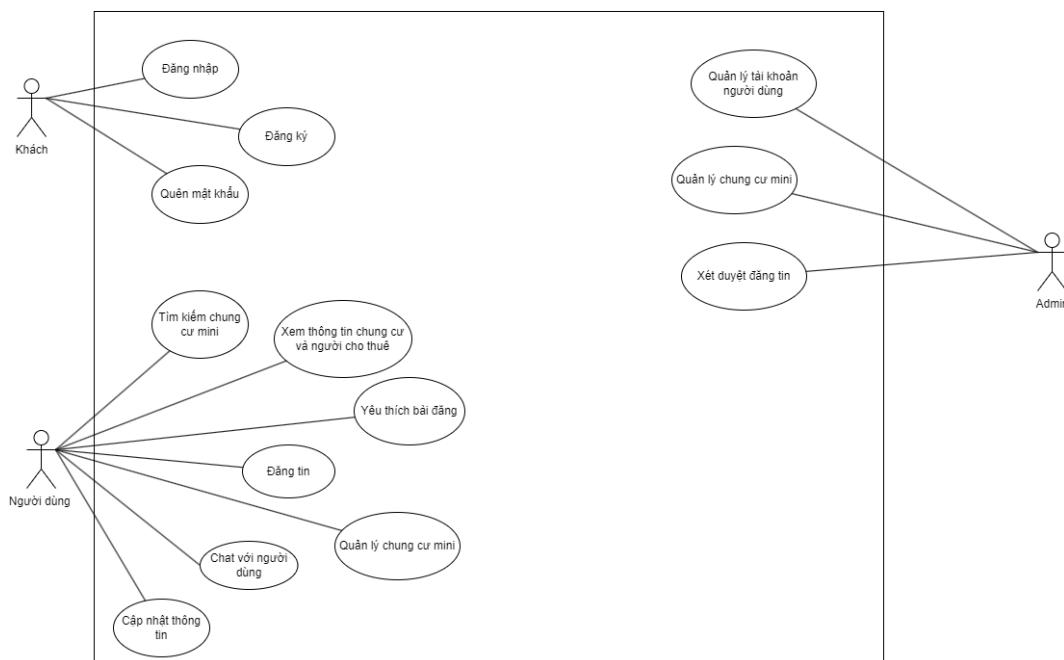
2.2.1 Phân tích tác nhân và vai trò

Trong hệ thống này, người dùng là tác nhân lớn nhất, có nhiều chức năng nhất. Bên cạnh đó còn có admin đóng vai trò xét duyệt và quản lý hệ thống. Cụ thể như sau:

- Khách: là tác nhân khi chưa đăng nhập, khách bị giới hạn quyền tương tác với website. Khách có thể thao tác đăng ký, đăng nhập, quên mật khẩu.

- Người dùng: là tác nhân sau khi đã đăng nhập vào thành công. Người dùng có thể thực hiện tất cả các chức năng của hệ thống.
- Admin: là tác nhân quản lý tài khoản người dùng và bài đăng chung cư mini, xét duyệt đăng tin cho người dùng.

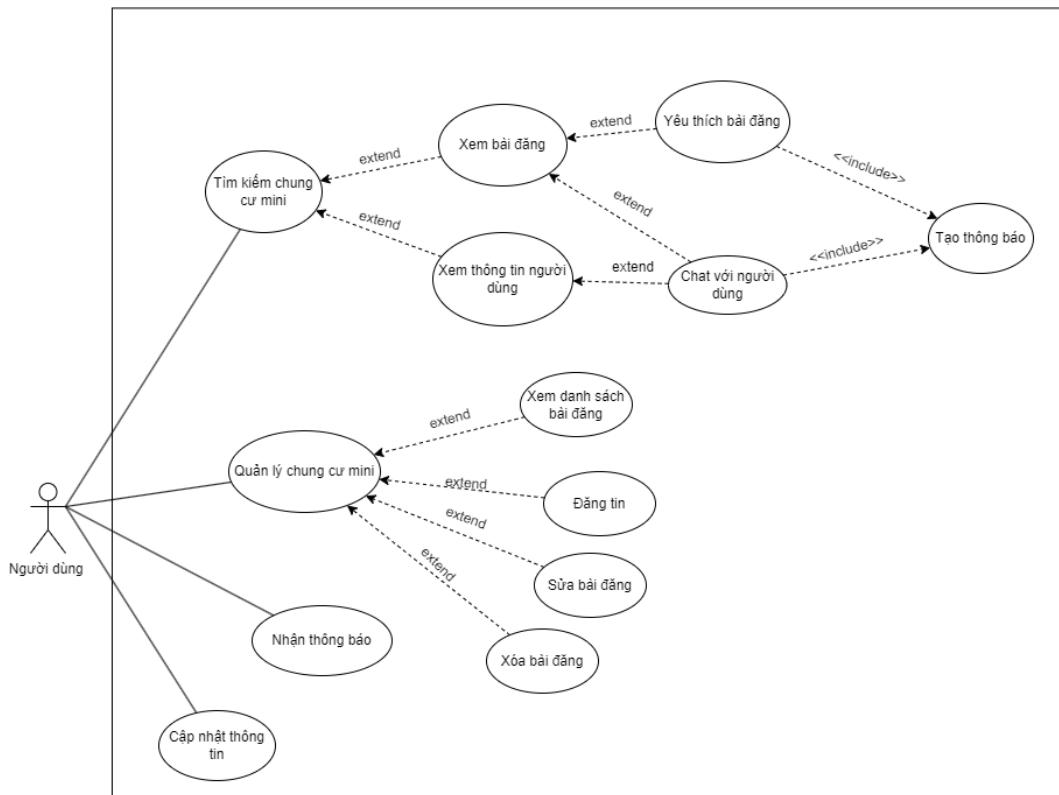
2.2.2 Biểu đồ use case tổng quát



Hình 2.1: Biểu đồ use case tổng quát

Hình 2.1 mô tả các tác nhân chính là "Khách", "Người dùng" và "Admin". Tác nhân "khách" có thể thực hiện các chức năng đăng ký, đăng nhập, khôi phục mật khẩu tài khoản. Đối với "người dùng" đã đăng nhập hệ thống có thể tìm kiếm bài đăng, xem thông tin bài và người đăng, liên hệ người đăng và lưu lại bài viết. Ngoài ra "người dùng" có thể quản lý các bài đăng tin, tạo đăng tin về chung cư mini của mình và đợi "admin" xét duyệt. "Người dùng" cập nhật thông tin cá nhân để người xem khác có thể liên lạc được. Cuối cùng, "admin" quản lý tất cả bài đăng và tài khoản của người dùng.

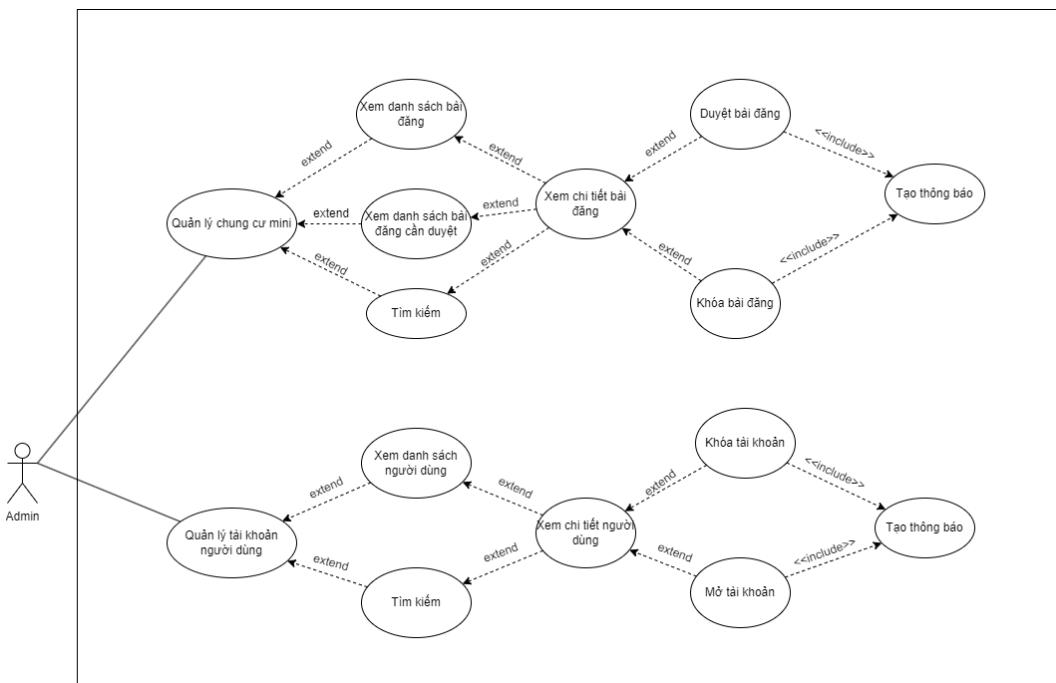
2.2.3 Biểu đồ use case phân rã của Người dùng



Hình 2.2: Phân rã use case của Người dùng

Các chức năng chính của "người dùng": Tìm kiếm bài đăng hiện có trên hệ thống, quản lý bài đăng của mình. Sau khi tìm kiếm được bài đăng mong muốn, "người dùng" có thể xem chi tiết bài đăng và thông tin người đăng, sau đó có thể lưu lại và liên lạc cho người đăng qua chat hoặc số điện thoại. Ngoài ra "người dùng" có thể tự tạo bài đăng và quản lý bài đăng của mình, bao gồm chỉnh sửa và xóa bài đăng.

2.2.4 Biểu đồ use case phân rã của Admin



Hình 2.3: Phân rã use case của Admin

Các chức năng chính của "admin": Xét duyệt cho bài đăng, quản lý các bài đăng và tài khoản người dùng trong hệ thống. Khi "người dùng" muốn đăng tin thì bài đăng sẽ ở trạng thái pending, sau khi được "admin" duyệt bài đăng sẽ sang trạng thái approved và hiển thị trên website của hệ thống. "Admin" có quyền xem danh sách bài đăng, danh sách người dùng và các hoạt động với danh sách này.

2.2.5 Quy trình nghiệp vụ

Quy trình nghiệp vụ chính của hệ thống tìm kiếm và cho thuê chung cư mini như sau:

Khách đăng ký tài khoản rồi đăng nhập vào website. Sau khi đăng nhập, người dùng có thể tìm kiếm chung cư mini theo các tiêu chí mong muốn như địa điểm, diện tích, giá cả, dịch vụ. Sau khi tìm được bài đăng mong muốn, người dùng có xem thông tin bài đăng và thông tin người đăng và lưu trữ, liên lạc với người đăng như chat, gửi email hoặc qua zalo nếu có nhu cầu. Ngoài ra, người dùng cũng có thể là một nhà môi giới với chức năng đăng bài và quản lý bài đăng, bài đăng khi được tạo sẽ ở trạng thái pending và cần được admin xét duyệt để có thể lên trang chủ của hệ thống.

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case "Đăng ký"

Mã use case	UC001
Tên use case	Đăng ký
Tác nhân	Khách
Mục đích sử dụng	Khách đăng ký vào hệ thống.
Sự kiện kích hoạt	Khách lựa chọn chức năng đăng ký.
Điều kiện tiên quyết	Không
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Khách chọn chức năng đăng ký. 2. Hệ thống hiển thị giao diện đăng ký. 3. Khách nhập các trường yêu cầu bắt buộc và thực hiện thao tác yêu cầu đăng ký. 4. Hệ thống kiểm tra các trường bắt buộc. 5. Hệ thống kiểm tra tài khoản hợp lệ, trùng lặp. 6. Hệ thống thông báo đăng ký thành công.
Luồng sự kiện thay thế	<ol style="list-style-type: none"> 4a. Hệ thống thông báo đăng ký lỗi. 5. Hệ thống thông báo tài khoản không hợp lệ. 6a. Hệ thống thông báo đăng ký không thành công.
Hậu điều kiện	Hệ thống thông báo đăng ký thành công và chuyển tiếp trang giao diện.

Bảng 2.1: Đặc tả use case "Đăng ký"

2.3.2 Đặc tả use case "Đăng nhập"

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Mã use case	UC002
Tên use case	Đăng nhập
Tác nhân	Khách
Mục đích sử dụng	Khách đăng nhập vào hệ thống.
Sự kiện kích hoạt	Khách lựa chọn chức năng đăng nhập.
Điều kiện tiên quyết	Không
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Khách chọn chức năng đăng nhập. 2. Hệ thống hiển thị giao diện đăng nhập. 3. Khách nhập các trường yêu cầu bắt buộc và thực hiện thao tác yêu cầu đăng nhập. 4. Hệ thống kiểm tra các trường bắt buộc và tài khoản hợp lệ hay không. 5. Hệ thống thông báo đăng nhập thành công.
Luồng sự kiện thay thế	<ol style="list-style-type: none"> 4a. Hệ thống thông báo các trường bắt buộc nhập. 5a. Hệ thống thông báo đăng ký không thành công.
Hậu điều kiện	Hệ thống thông báo đăng nhập thành công và chuyển tiếp trang giao diện.

Bảng 2.2: Đặc tả use case "Đăng nhập"

2.3.3 Đặc tả use case "Đăng xuất"

Mã use case	UC002
Tên use case	Đăng xuất
Tác nhân	Khách
Mục đích sử dụng	Người dùng đăng xuất khỏi hệ thống.
Sự kiện kích hoạt	Người dùng lựa chọn chức năng đăng xuất.
Điều kiện tiên quyết	Người dùng đã đăng nhập
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Khách chọn chức năng đăng xuất. 2. Hệ thống chuyển hướng về trang giao diện đăng nhập.
Luồng sự kiện thay thế	Không
Hậu điều kiện	<ol style="list-style-type: none"> 1. Hệ thống đăng xuất người dùng thành công 2. Trở về trang đăng nhập

Bảng 2.3: Đặc tả use case "Đăng xuất"

2.3.4 Đặc tả use case "Quên mật khẩu"

Mã use case	UC004
Tên use case	Quên mật khẩu
Tác nhân	Khách
Mục đích sử dụng	Người dùng quên mật khẩu, muốn khôi phục mật khẩu mới cho tài khoản.
Sự kiện kích hoạt	Khách lựa chọn chức năng quên mật khẩu.
Điều kiện tiên quyết	Không
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Khách nhập email đã đăng ký tài khoản. 2. Hệ thống gửi email xác nhận khôi phục tài khoản đến email đã nhập. 3. Khách truy cập đường dẫn được gửi trong email. 4. Hệ thống điều hướng khách đến trang khôi phục mật khẩu. 5. Khách nhập mật khẩu mới và xác nhận khôi phục mật khẩu.
Luồng sự kiện thay thế	<ol style="list-style-type: none"> 2a. Khách nhập email không tồn tại, sẽ không nhận được email xác thực. 5a. Khách nhập mật khẩu lần 2 không khớp với mật khẩu lần 1, hệ thống báo lỗi mật khẩu không trùng khớp.
Hậu điều kiện	Không

Bảng 2.4: Đặc tả use case "Quên mật khẩu"

2.3.5 Đặc tả use case "Cập nhật thông tin"

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Mã use case	UC005
Tên use case	Cập nhật thông tin
Tác nhân	Người dùng
Mục đích sử dụng	Cập nhật thông tin cá nhân của người đang sử dụng tài khoản.
Sự kiện kích hoạt	Người dùng lựa chọn chức năng cập nhật thông tin cá nhân.
Điều kiện tiên quyết	<ol style="list-style-type: none"> 1. Cập nhật thông tin cá nhân trên tài khoản người dùng đăng nhập. 2. Đã đăng nhập thành công.
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng lựa chọn chức năng cập nhật thông tin cá nhân. 2. Hệ thống hiển thị giao diện cập nhật thông tin. 3. Người dùng nhập các trường yêu cầu bắt buộc và thực hiện thao tác yêu cầu cập nhật. 4. Hệ thống kiểm tra các trường bắt buộc. 5. Hệ thống thông báo thành công, chuyển về giao diện trước đó.
Luồng sự kiện thay thế	<ol style="list-style-type: none"> 4a. Hệ thống thông báo nhập thiếu trường. 5a. Hệ thống giữ nguyên thông tin ban đầu khi người dùng xác nhận hủy cập nhật thông tin.
Hậu điều kiện	Hệ thống thông báo thành công, chuyển về giao diện trước đó.

Bảng 2.5: Đặc tả use case "Cập nhật thông tin"

2.3.6 Đặc tả use case "Nhận thông báo"

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Mã use case	UC006
Tên use case	Nhận thông báo
Tác nhân	Người dùng
Mục đích sử dụng	Xem các thông báo gửi đến người dùng
Sự kiện kích hoạt	Người dùng lựa chọn chức năng Xem thông báo.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Người dùng đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Người dùng lựa chọn chức năng xem thông báo 2. Hệ thống hiển thị giao diện các thông báo.
Luồng sự kiện thay thế	Không
Hậu điều kiện	Không

Bảng 2.6: Đặc tả use case "Nhận thông báo"

2.3.7 Đặc tả use case "Tìm kiếm chung cư mini"

Mã use case	UC007
Tên use case	Tìm kiếm chung cư mini
Tác nhân	Người dùng
Mục đích sử dụng	Tìm chung cư mini theo nhu cầu
Sự kiện kích hoạt	Người dùng lựa chọn chức năng tìm kiếm chung cư mini.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Người dùng đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Người dùng lựa chọn chức năng tìm kiếm chung cư mini. 2. Hệ thống hiển thị giao diện tìm kiếm cùng các tiêu chí lọc. 3. Người dùng lọc các chung cư mini theo các tiêu chí mong muốn và nhập cụm từ tìm kiếm liên quan. 4. Hệ thống hiển thị các chung cư theo các tiêu chí đã lọc.
Luồng sự kiện thay thế	<ul style="list-style-type: none"> 4a. Hệ thống hiển thị giao diện không tìm thấy do không tìm được chung cư theo tiêu chí đã lọc.
Hậu điều kiện	Hệ thống hiển thị giao diện

Bảng 2.7: Đặc tả use case "Tìm kiếm chung cư mini"

2.3.8 Đặc tả use case "Xem bài đăng"

Mã use case	UC008
Tên use case	Xem bài đăng
Tác nhân	Người dùng
Mục đích sử dụng	Xem chi tiết bài đăng
Sự kiện kích hoạt	Người dùng click vào bài đăng.
Điều kiện tiên quyết	1. Người dùng đã đăng nhập.
Luồng sự kiện chính	1. Người dùng click vào bài đăng muốn xem 2. Hệ thống hiển thị trang thông tin của bài đăng cùng thông tin liên lạc người dùng
Luồng sự kiện thay thế	Không
Hậu điều kiện	Hệ thống hiển thị trang thông tin

Bảng 2.8: Đặc tả use case "Xem bài đăng"

2.3.9 Đặc tả use case "Xem thông tin người dùng"

Mã use case	UC009
Tên use case	Xem thông tin người dùng
Tác nhân	Người dùng
Mục đích sử dụng	Xem thông tin người dùng
Sự kiện kích hoạt	Người dùng click vào phần thông tin người dùng.
Điều kiện tiên quyết	1. Người dùng đã đăng nhập.
Luồng sự kiện chính	1. Người dùng click vào phần thông tin người dùng 2. Hệ thống hiển thị trang cá nhân của người dùng bao gồm thông tin liên lạc và các bài đăng của người đó.
Luồng sự kiện thay thế	Không
Hậu điều kiện	Hệ thống hiển thị trang thông tin

Bảng 2.9: Đặc tả use case "Xem thông tin người dùng"

2.3.10 Đặc tả use case "Yêu thích bài đăng"

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Mã use case	UC010
Tên use case	Yêu thích bài đăng
Tác nhân	Người dùng
Mục đích sử dụng	Yêu thích và lưu trữ bài đăng
Sự kiện kích hoạt	Người dùng click vào icon yêu thích.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Người dùng đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Người dùng click vào icon yêu thích. 2. Hệ thống sẽ tăng 1 lượt yêu thích cho bài đăng. 3. Hệ thống thêm bài đăng vào mục yêu thích của người dùng. 4. Hệ thống gửi thông báo đến người đăng tin.
Luồng sự kiện thay thế	Không
Hậu điều kiện	Hệ thống hiển thị thông báo thêm vào mục yêu thích thành công

Bảng 2.10: Đặc tả use case "Yêu thích bài đăng"

2.3.11 Đặc tả use case "Chat với người dùng"

Mã use case	UC011
Tên use case	Chat với người dùng
Tác nhân	Người dùng
Mục đích sử dụng	Chat với người dùng
Sự kiện kích hoạt	Người dùng click vào phần chat với người dùng.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Người dùng đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Người dùng click vào phần chat với người dùng. 2. Hệ thống mở một giao diện chat với người dùng khác bao gồm các tin nhắn giữa 2 người. 3. Người dùng nhập nội dung tin nhắn và bấm gửi. 4. Hệ thống lưu trữ tin nhắn và cập nhật phòng chat của 2 người 5. Hệ thống gửi thông báo đến người dùng còn lại.
Luồng sự kiện thay thế	Không
Hậu điều kiện	Hệ thống hiển thị giao diện chat

Bảng 2.11: Đặc tả use case "Chat với người dùng"

2.3.12 Đặc tả use case "Xem danh sách bài đăng"

Mã use case	UC012
Tên use case	Xem danh sách bài đăng
Tác nhân	Người dùng
Mục đích sử dụng	Xem danh sách bài đăng của người dùng
Sự kiện kích hoạt	Người dùng click vào phần xem danh sách bài đăng.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Người dùng đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Người dùng click vào phần xem danh sách bài đăng. 2. Hệ thống mở một giao diện danh sách các bài đăng của người dùng
Luồng sự kiện thay thế	Không
Hậu điều kiện	Hệ thống hiển thị giao diện danh sách các bài đăng của người dùng

Bảng 2.12: Đặc tả use case "Xem danh sách bài đăng"

2.3.13 Đặc tả use case "Đăng tin"

Mã use case	UC013
Tên use case	Đăng tin
Tác nhân	Người dùng
Mục đích sử dụng	Đăng tin cho thuê chung cư
Sự kiện kích hoạt	Người dùng click vào phần đăng tin.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Người dùng đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Người dùng click vào phần đăng tin. 2. Hệ thống mở giao diện đăng tin. 3. Người dùng nhập các trường yêu cầu bắt buộc và thực hiện thao tác yêu cầu đăng tin. 4. Hệ thống kiểm tra các trường bắt buộc. 5. Hệ thống thông báo tạo bài đăng tin thành công và đợi xét duyệt
Luồng sự kiện thay thế	<ul style="list-style-type: none"> 4a Hệ thống thông báo lỗi khi tạo bài đăng 6a Hệ thống thông báo tạo bài đăng không thành công
Hậu điều kiện	Hệ thống hiển thị thông báo tạo bài đăng thành công

Bảng 2.13: Đặc tả use case "Đăng tin"

2.3.14 Đặc tả use case "Sửa bài đăng"

Mã use case	UC014
Tên use case	Sửa bài đăng
Tác nhân	Người dùng
Mục đích sử dụng	Sửa bài đăng cho người dùng
Sự kiện kích hoạt	Người dùng click vào phần đăng tin.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Người dùng đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Người dùng click vào phần sửa bài đăng. 2. Hệ thống mở giao diện sửa bài đăng với các thông tin cũ của bài. 3. Người dùng nhập các thay đổi về bài đăng của mình 4. Hệ thống kiểm tra các trường bắt buộc. 5. Hệ thống thông báo sửa bài đăng tin thành công.
Luồng sự kiện thay thế	<ul style="list-style-type: none"> 4a Hệ thống thông báo lỗi khi sửa bài đăng 6a Hệ thống thông báo sửa bài đăng không thành công
Hậu điều kiện	Hệ thống hiển thị thông báo sửa bài đăng thành công

Bảng 2.14: Đặc tả use case "Sửa bài đăng"

2.3.15 Đặc tả use case "Xóa bài đăng"

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Mã use case	UC015
Tên use case	Xóa bài đăng
Tác nhân	Người dùng
Mục đích sử dụng	Xóa bài đăng cho người dùng
Sự kiện kích hoạt	Người dùng click vào phần xóa bài đăng.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Người dùng đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Người dùng click vào phần xóa bài đăng 2. Hệ thống mở thông báo cho người dùng liệu có muốn xóa bài đăng 3. Người dùng xác nhận xóa bài đăng 4. Hệ thống thông báo xóa bài đăng tin thành công.
Luồng sự kiện thay thế	4a Hệ thống thông báo hủy xóa bài
Hậu điều kiện	Hệ thống hiển thị thông báo xóa bài đăng thành công

Bảng 2.15: Đặc tả use case "Xóa bài đăng"

2.3.16 Đặc tả use case "Xem danh sách bài đăng"

Mã use case	UC016
Tên use case	Xem danh sách bài đăng
Tác nhân	Admin
Mục đích sử dụng	Xem danh sách bài đăng trong hệ thống
Sự kiện kích hoạt	Admin click vào phần xem danh sách bài đăng.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Admin đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Admin click vào phần xem danh sách bài đăng 2. Hệ thống mở trang giao diện danh sách bài đăng
Luồng sự kiện thay thế	Không
Hậu điều kiện	Hệ thống hiển thị trang giao diện danh sách bài đăng

Bảng 2.16: Đặc tả use case "Xem danh sách bài đăng"

2.3.17 Đặc tả use case "Xem danh sách bài đăng cần duyệt"

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Mã use case	UC017
Tên use case	Xem danh sách bài đăng cần duyệt
Tác nhân	Admin
Mục đích sử dụng	Xem danh sách bài đăng cần duyệt trong hệ thống
Sự kiện kích hoạt	Admin click vào phần xem danh sách bài đăng cần duyệt.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Admin đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Admin click vào phần xem danh sách bài đăng cần duyệt 2. Hệ thống mở trang giao diện danh sách bài đăng cần duyệt
Luồng sự kiện thay thế	Không
Hậu điều kiện	Hệ thống hiển thị trang giao diện danh sách bài đăng cần duyệt

Bảng 2.17: Đặc tả use case "Xem danh sách bài đăng cần duyệt"

2.3.18 Đặc tả use case "Tìm kiếm"

Mã use case	UC018
Tên use case	Tìm kiếm
Tác nhân	Admin
Mục đích sử dụng	Tìm kiếm bài đăng có trong hệ thống
Sự kiện kích hoạt	Admin nhập thông tin bài đăng cần tìm.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Admin đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Admin nhập thông tin bài đăng cần tìm. 2. Hệ thống hiển thị bài đăng từ thông tin đã nhập từ admin
Luồng sự kiện thay thế	2a Hệ thống thông báo không tìm thấy bài đăng
Hậu điều kiện	Hệ thống hiển thị bài đăng cần tìm

Bảng 2.18: Đặc tả use case "Tìm kiếm"

2.3.19 Đặc tả use case "Xem chi tiết bài đăng"

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Mã use case	UC019
Tên use case	Xem chi tiết bài đăng
Tác nhân	Admin
Mục đích sử dụng	Xem chi tiết bài đăng
Sự kiện kích hoạt	Admin click vào bài đăng.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Admin đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Admin click vào bài đăng muốn xem 2. Hệ thống hiển thị trang thông tin của bài đăng cùng thông tin liên lạc người dùng
Luồng sự kiện thay thế	Không
Hậu điều kiện	Hệ thống hiển thị trang thông tin của bài đăng

Bảng 2.19: Đặc tả use case "Xem chi tiết bài đăng"

2.3.20 Đặc tả use case "Duyệt bài đăng"

Mã use case	UC020
Tên use case	Duyệt bài đăng
Tác nhân	Admin
Mục đích sử dụng	Duyệt bài đăng
Sự kiện kích hoạt	Admin click vào phần duyệt bài đăng.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Admin đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Admin click vào phần duyệt bài đăng 2. Hệ thống thông báo duyệt bài đăng thành công 3. Hệ thống tạo thông báo tới người dùng về bài đăng được duyệt 4. Hệ thống cập nhật bài đăng lên trang chủ
Luồng sự kiện thay thế	<ul style="list-style-type: none"> 1a Admin click vào phần từ chối duyệt bài đăng 2a Hệ thống thông báo từ chối duyệt thành công 3a Hệ thống tạo thông báo tới người dùng về bài đăng không được duyệt
Hậu điều kiện	Hệ thống hiển thị thông báo duyệt cho admin

Bảng 2.20: Đặc tả use case "Duyệt bài đăng"

2.3.21 Đặc tả use case "Khóa bài đăng"

Mã use case	UC021
Tên use case	Khóa bài đăng
Tác nhân	Admin
Mục đích sử dụng	khóa bài đăng
Sự kiện kích hoạt	Admin click vào phần khóa bài đăng.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Admin đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Admin click vào phần khóa bài đăng 2. Hệ thống mở form thông báo lý do khóa bài đăng 3. Admin nhập lý do khóa 4. Hệ thống thông báo khóa bài đăng thành công 5. Hệ thống gửi thông báo khóa bài đăng tới người dùng cùng lý do
Luồng sự kiện thay thế	Không
Hậu điều kiện	Hệ thống hiển thị thông báo khóa thành công cho admin

Bảng 2.21: Đặc tả use case "Khóa bài đăng"

2.3.22 Đặc tả use case "Xem danh sách người dùng"

Mã use case	UC022
Tên use case	Xem danh sách người dùng
Tác nhân	Admin
Mục đích sử dụng	Xem danh sách người dùng trong hệ thống
Sự kiện kích hoạt	Admin click vào phần xem danh sách người dùng.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Admin đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Admin click vào phần xem danh sách người dùng 2. Hệ thống mở trang giao diện danh sách người dùng
Luồng sự kiện thay thế	Không
Hậu điều kiện	Hệ thống hiển thị trang giao diện danh sách người dùng

Bảng 2.22: Đặc tả use case "Xem danh sách người dùng"

2.3.23 Đặc tả use case "Tìm kiếm"

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Mã use case	UC023
Tên use case	Tìm kiếm
Tác nhân	Admin
Mục đích sử dụng	Tìm kiếm người dùng có trong hệ thống
Sự kiện kích hoạt	Admin nhập thông tin người dùng.
Điều kiện tiên quyết	1. Admin đã đăng nhập.
Luồng sự kiện chính	1. Admin nhập thông tin người dùng cần tìm. 2. Hệ thống hiển thị bài đăng từ thông tin đã nhập từ admin
Luồng sự kiện thay thế	2a Hệ thống thông báo không tìm thấy người dùng
Hậu điều kiện	Hệ thống hiển thị người dùng cần tìm

Bảng 2.23: Đặc tả use case "Tìm kiếm"

2.3.24 Đặc tả use case "Xem chi tiết người dùng"

Mã use case	UC024
Tên use case	Xem chi tiết người dùng
Tác nhân	Admin
Mục đích sử dụng	Xem chi tiết người dùng
Sự kiện kích hoạt	Admin click vào phần người dùng.
Điều kiện tiên quyết	1. Admin đã đăng nhập.
Luồng sự kiện chính	1. Admin click vào người dùng muốn xem 2. Hệ thống hiển thị trang thông tin của người dùng gồm thông tin liên lạc, giới thiệu và các bài đăng, các bài đăng yêu thích
Luồng sự kiện thay thế	Không
Hậu điều kiện	Hệ thống hiển thị trang thông tin của người dùng

Bảng 2.24: Đặc tả use case "Xem chi tiết người dùng"

2.3.25 Đặc tả use case "Khóa tài khoản"

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Mã use case	UC025
Tên use case	Khóa tài khoản
Tác nhân	Admin
Mục đích sử dụng	khóa tài khoản
Sự kiện kích hoạt	Admin click vào phần khóa tài khoản.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Admin đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Admin click vào phần khóa tài khoản 2. Hệ thống mở form thông báo lý do khóa tài khoản 3. Admin nhập lý do khóa 4. Hệ thống thông báo khóa tài khoản thành công 5. Hệ thống gửi thông báo khóa tài khoản tới người dùng cùng lý do
Luồng sự kiện thay thế	Không
Hậu điều kiện	Hệ thống hiển thị thông báo khóa thành công cho admin

Bảng 2.25: ĐẶC TẢ USE CASE "KHÓA TÀI KHOẢN"

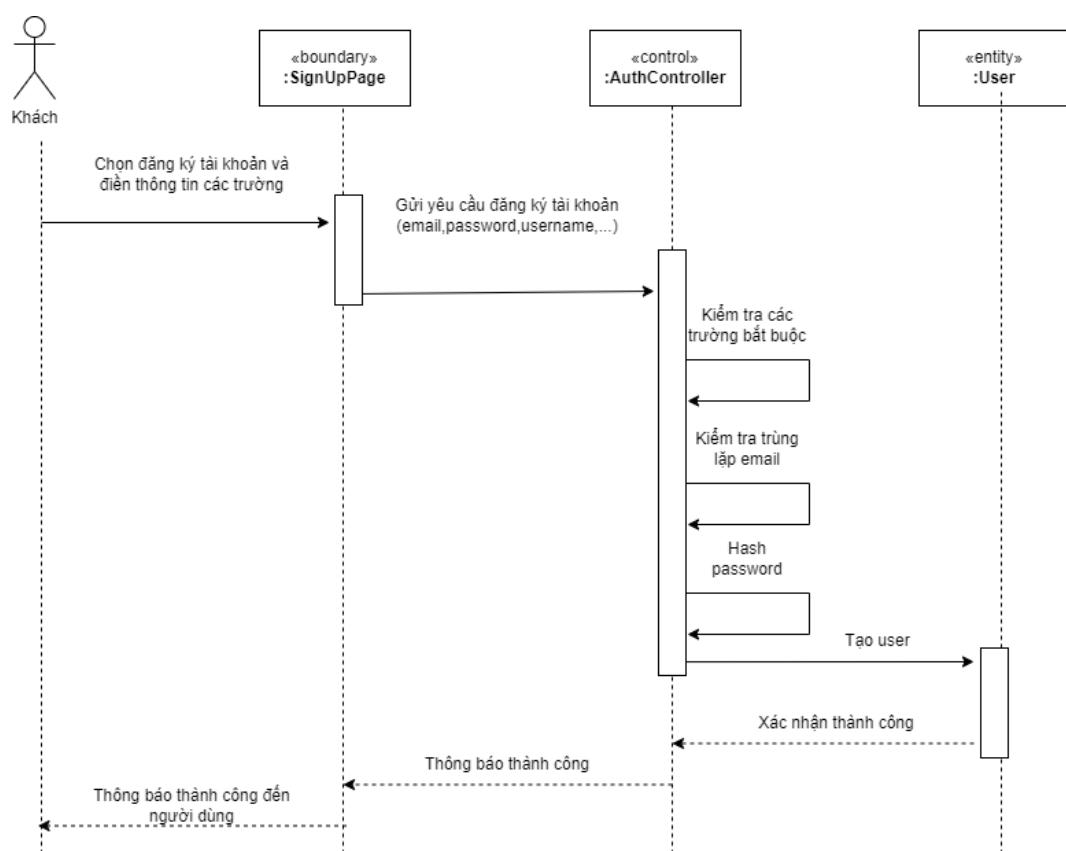
2.3.26 ĐẶC TẢ USE CASE "MỞ TÀI KHOẢN"

Mã use case	UC026
Tên use case	Mở tài khoản
Tác nhân	Admin
Mục đích sử dụng	Mở tài khoản
Sự kiện kích hoạt	Admin click vào phần mở tài khoản.
Điều kiện tiên quyết	<ul style="list-style-type: none"> 1. Admin đã đăng nhập.
Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Admin click vào phần mở tài khoản 2. Hệ thống thông báo mở tài khoản thành công 3. Hệ thống gửi thông báo khóa tài khoản tới người dùng
Luồng sự kiện thay thế	Không
Hậu điều kiện	Hệ thống hiển thị thông báo mở thành công cho admin

Bảng 2.26: ĐẶC TẢ USE CASE "MỞ TÀI KHOẢN"

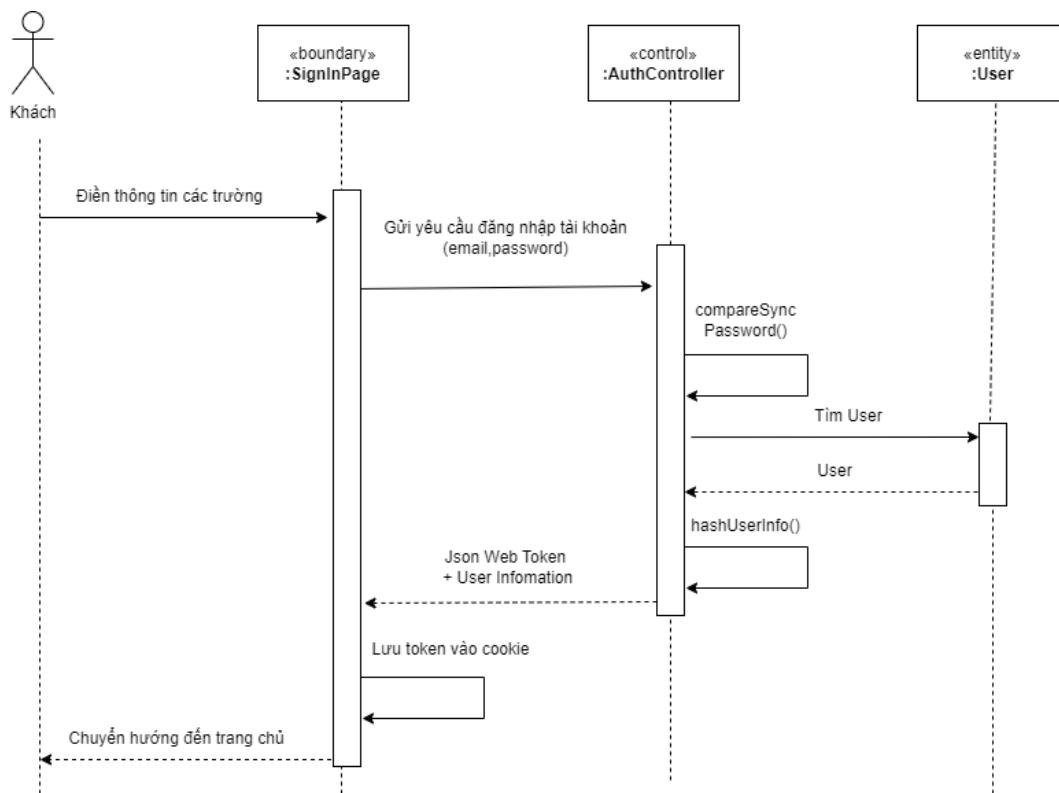
2.4 Mô hình hóa tương tác - Các biểu đồ tuần tự

2.4.1 Biểu đồ tuần tự "Đăng ký"



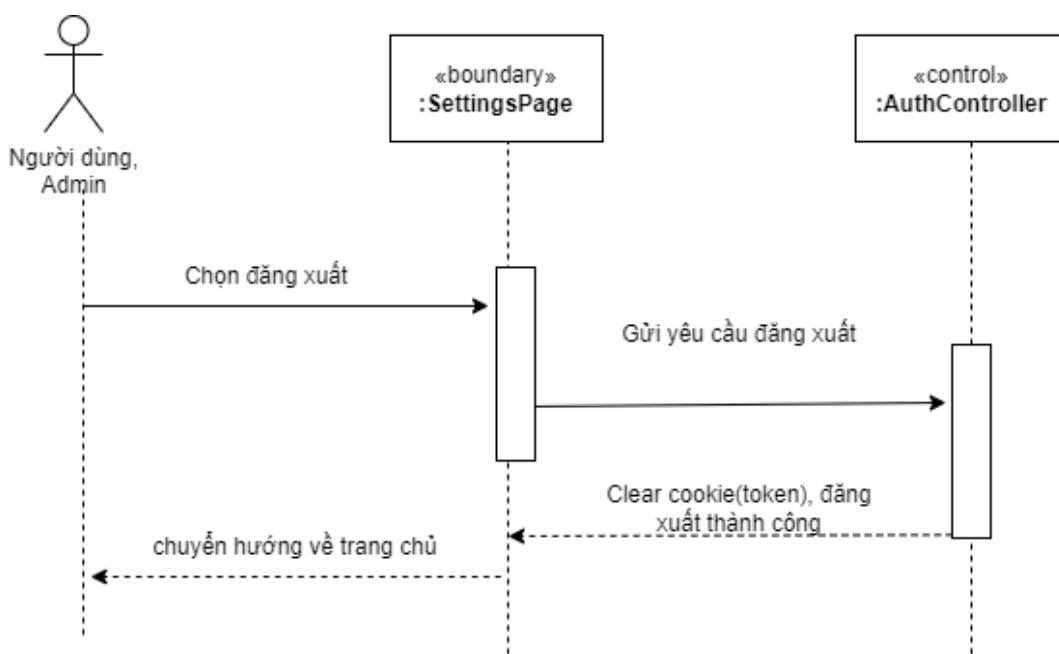
Hình 2.4: Biểu đồ tuần tự "Đăng ký"

2.4.2 Biểu đồ tuần tự "Đăng nhập"



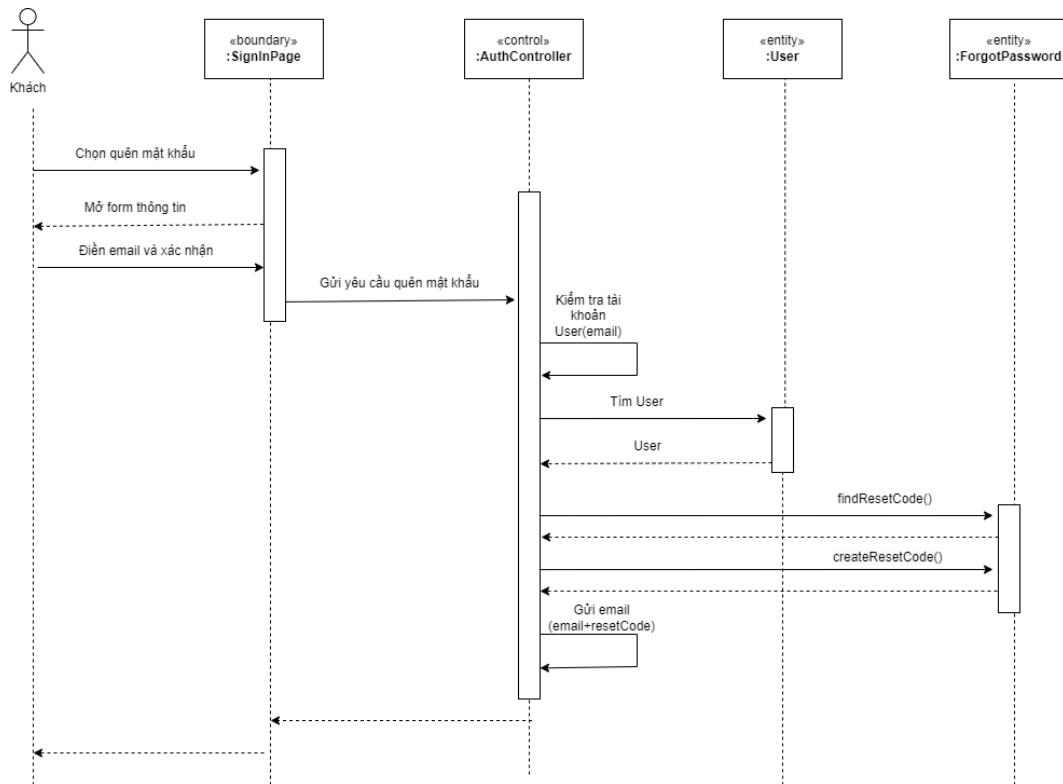
Hình 2.5: Biểu đồ tuần tự "Đăng nhập"

2.4.3 Biểu đồ tuần tự "Đăng xuất"



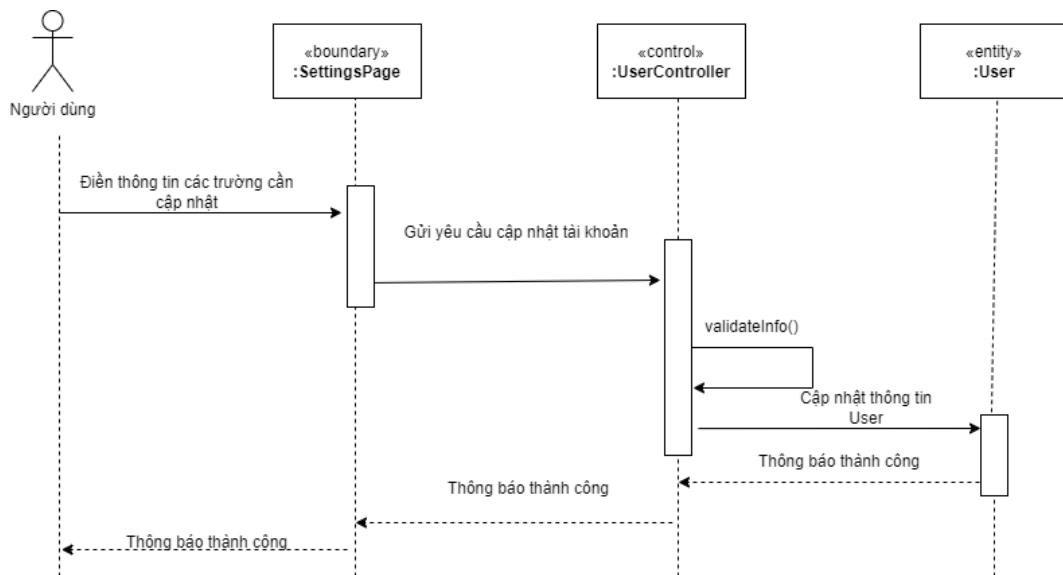
Hình 2.6: Biểu đồ tuần tự "Đăng xuất"

2.4.4 Biểu đồ tuần tự "Quên mật khẩu"



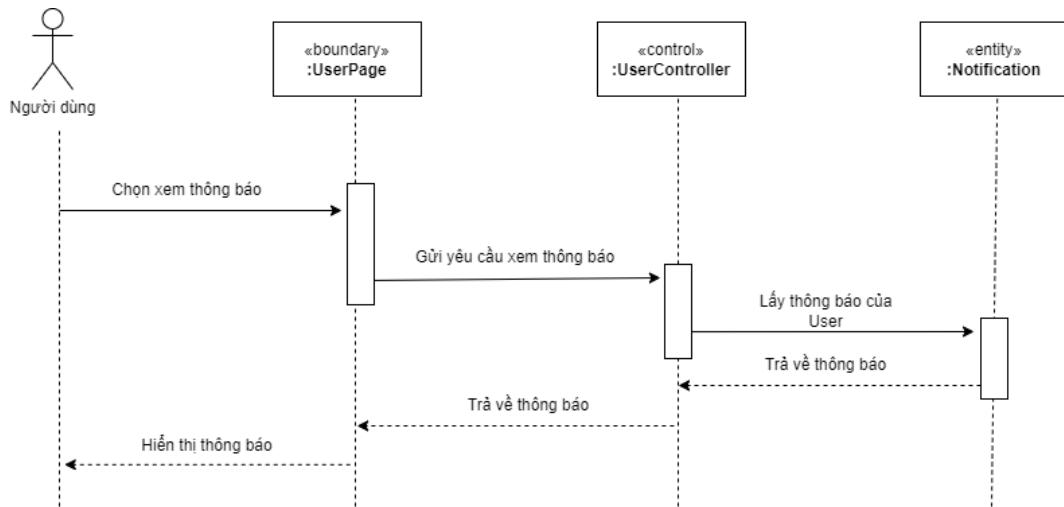
Hình 2.7: Biểu đồ tuần tự "Quên mật khẩu"

2.4.5 Biểu đồ tuần tự "Cập nhật thông tin"



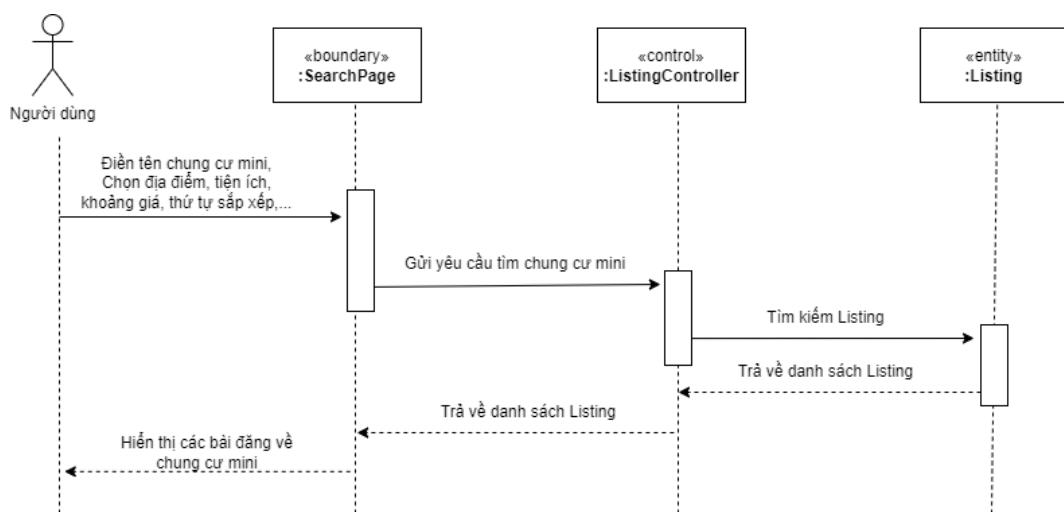
Hình 2.8: Biểu đồ tuần tự "Cập nhật thông tin"

2.4.6 Biểu đồ tuần tự "Nhận thông báo"



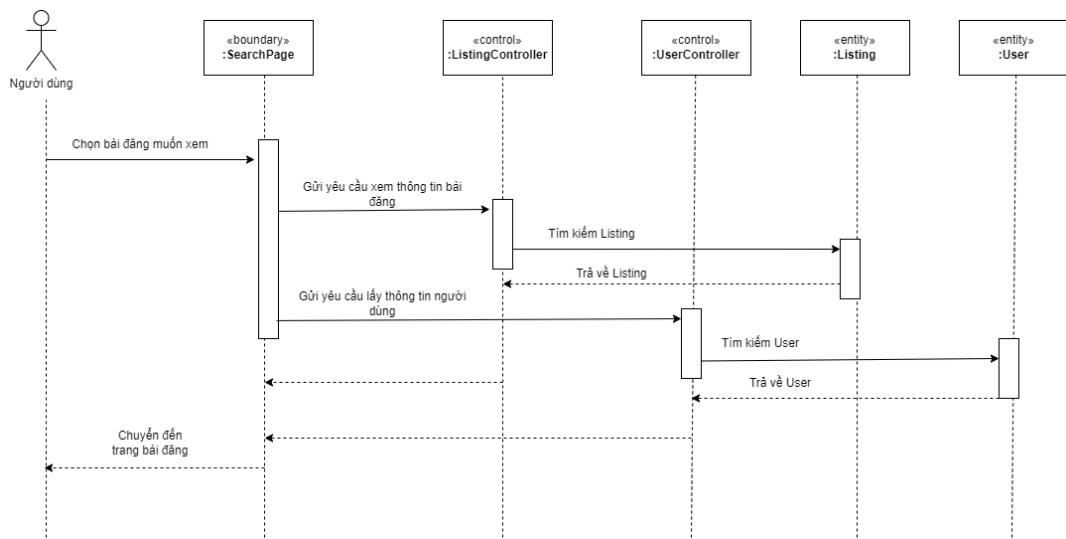
Hình 2.9: Biểu đồ tuần tự "Nhận thông báo"

2.4.7 Biểu đồ tuần tự "Tìm kiếm chung cư mini"



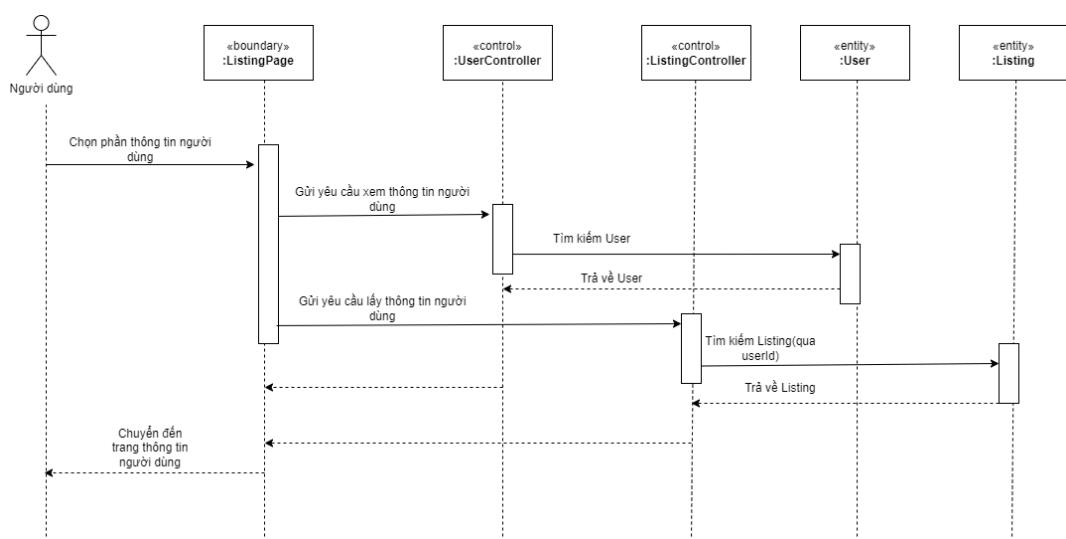
Hình 2.10: Biểu đồ tuần tự "Tìm kiếm chung cư mini"

2.4.8 Biểu đồ tuần tự "Xem bài đăng"



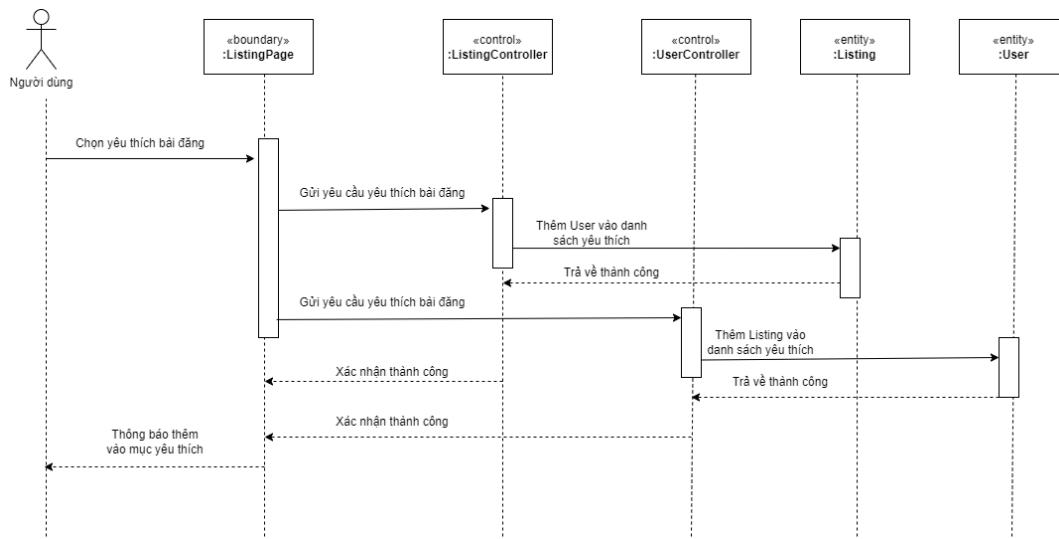
Hình 2.11: Biểu đồ tuần tự "Xem bài đăng"

2.4.9 Biểu đồ tuần tự "Xem thông tin người dùng"



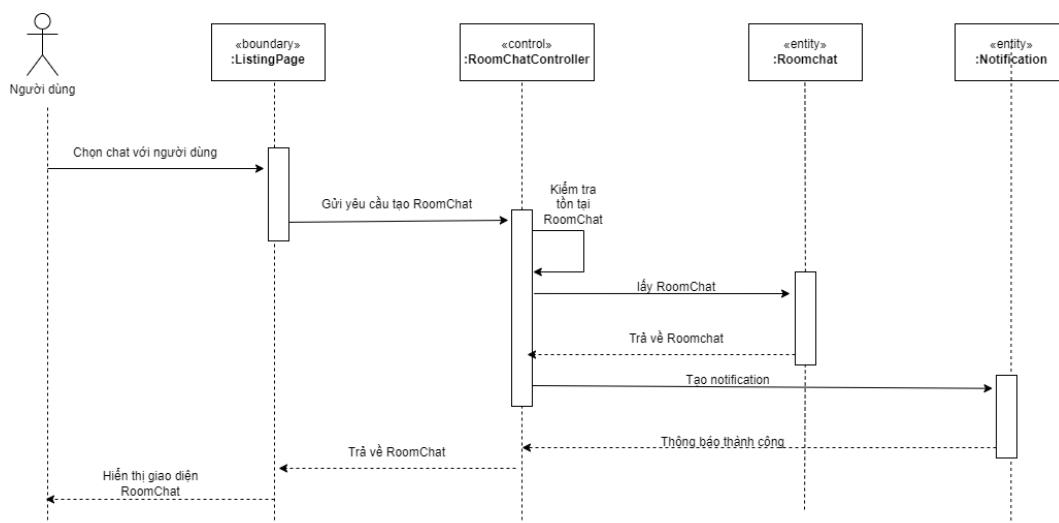
Hình 2.12: Biểu đồ tuần tự "Xem thông tin người dùng"

2.4.10 Biểu đồ tuần tự "Yêu thích bài đăng"



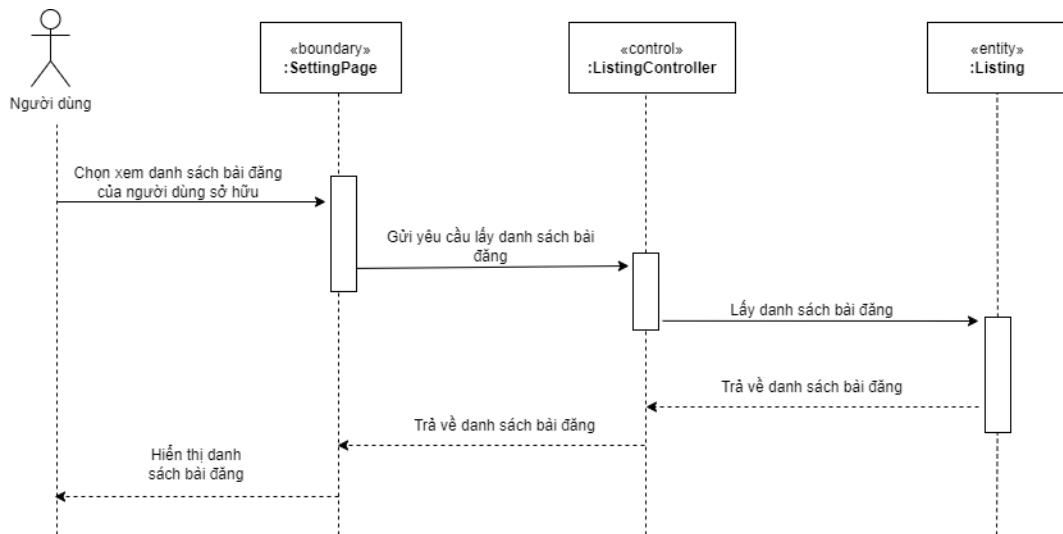
Hình 2.13: Biểu đồ tuần tự "Yêu thích bài đăng"

2.4.11 Biểu đồ tuần tự "Chat với người dùng"



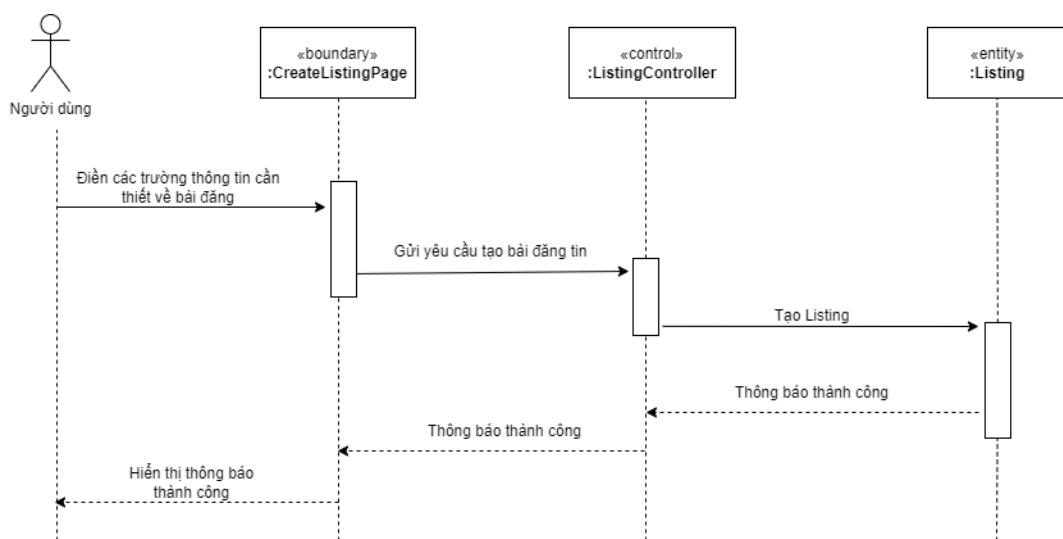
Hình 2.14: Biểu đồ tuần tự "Chat với người dùng"

2.4.12 Biểu đồ tuần tự "Xem danh sách bài đăng"



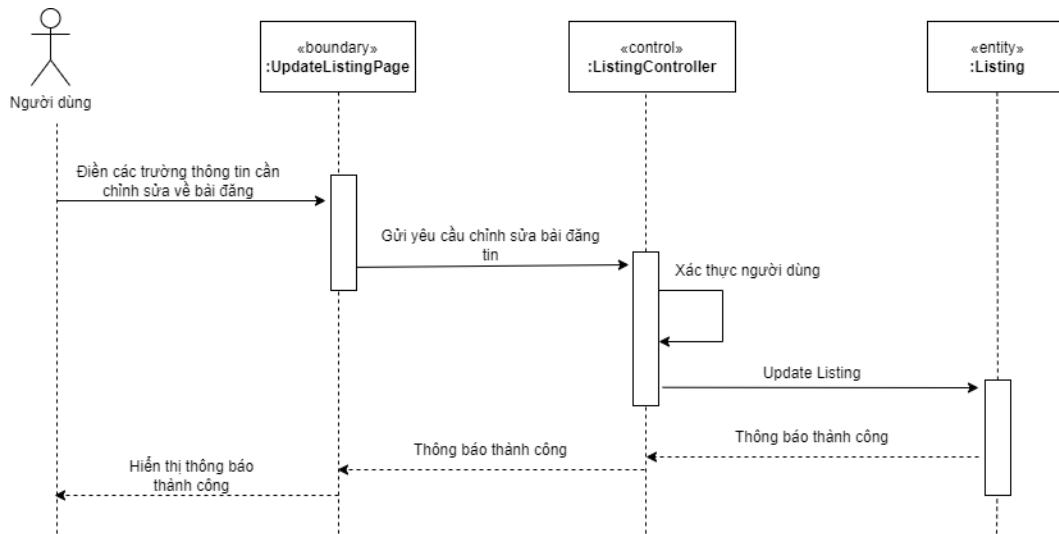
Hình 2.15: Biểu đồ tuần tự "Xem danh sách bài đăng"

2.4.13 Biểu đồ tuần tự "Đăng tin"



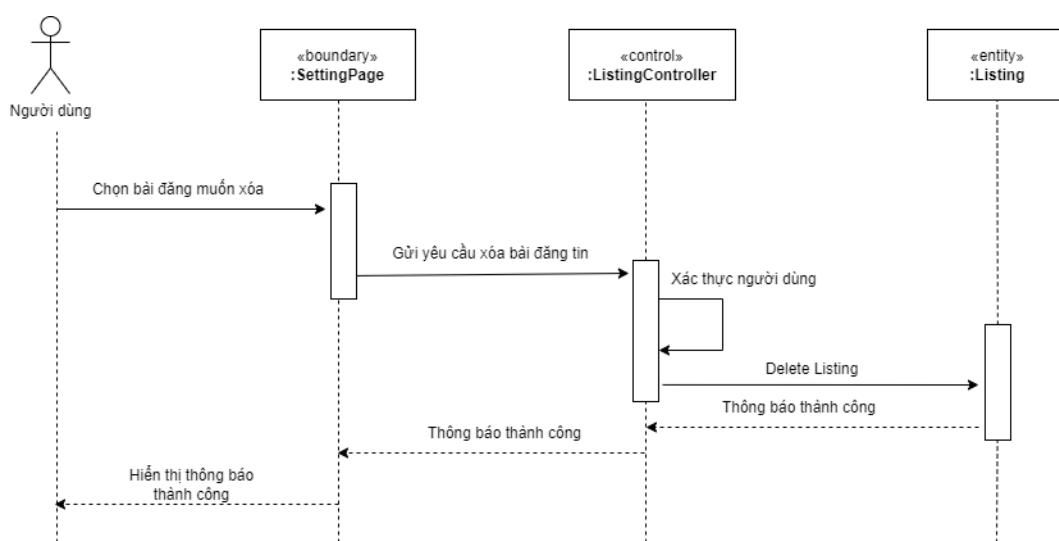
Hình 2.16: Biểu đồ tuần tự "Đăng tin"

2.4.14 Biểu đồ tuần tự "Sửa bài đăng"



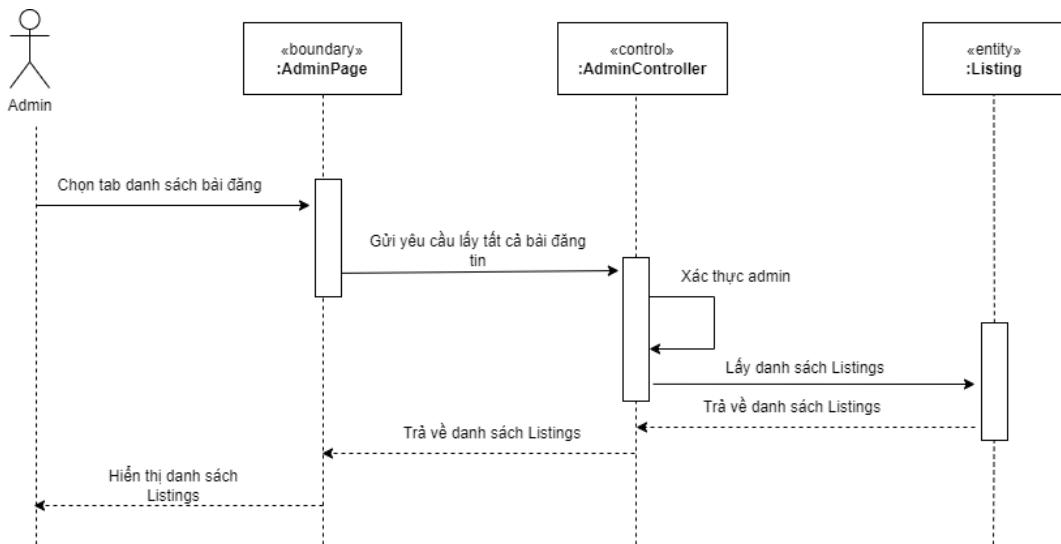
Hình 2.17: Biểu đồ tuần tự "Sửa bài đăng"

2.4.15 Biểu đồ tuần tự "Xóa bài đăng"



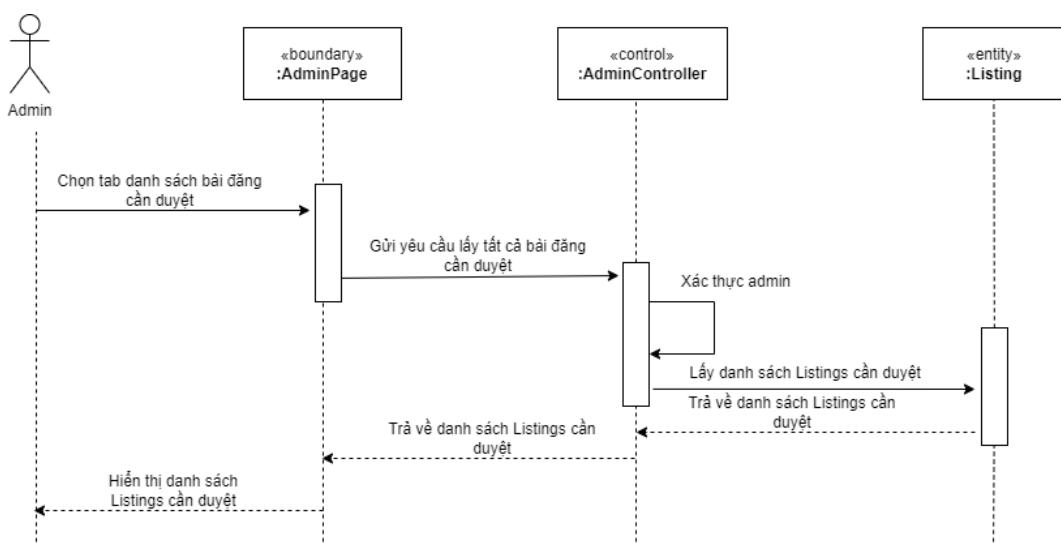
Hình 2.18: Biểu đồ tuần tự "Xóa bài đăng"

2.4.16 Biểu đồ tuần tự "Xem danh sách bài đăng"



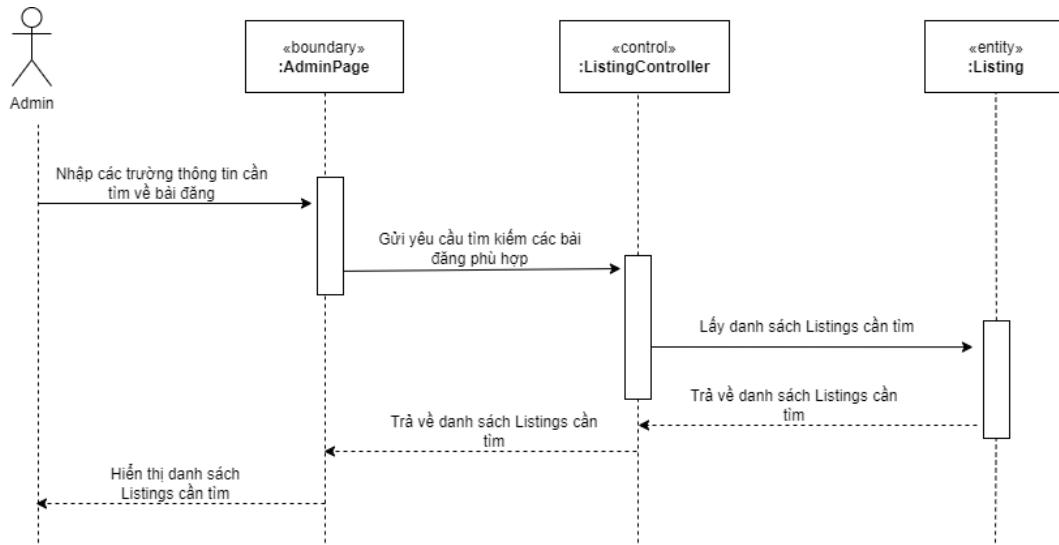
Hình 2.19: Biểu đồ tuần tự "Xem danh sách bài đăng"

2.4.17 Biểu đồ tuần tự "Xem danh sách bài đăng cần duyệt"



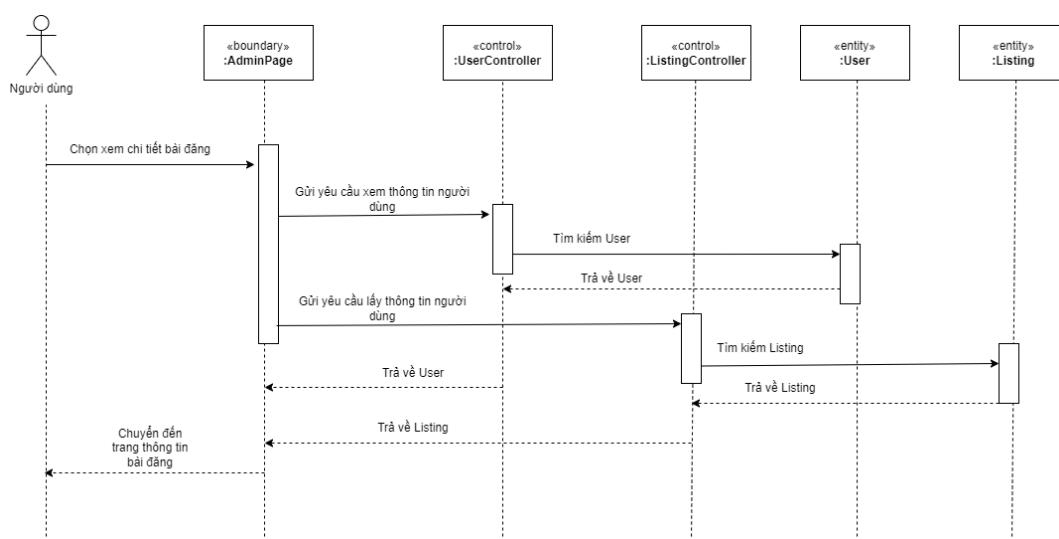
Hình 2.20: Biểu đồ tuần tự "Xem danh sách bài đăng cần duyệt"

2.4.18 Biểu đồ tuần tự "Tìm kiếm"



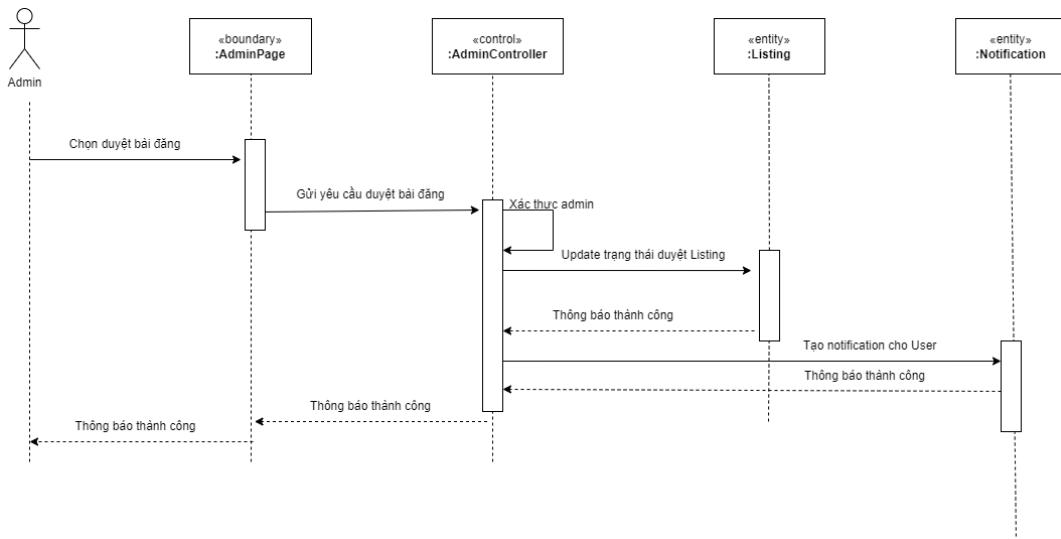
Hình 2.21: Biểu đồ tuần tự "Tìm kiếm"

2.4.19 Biểu đồ tuần tự "Xem chi tiết bài đăng"



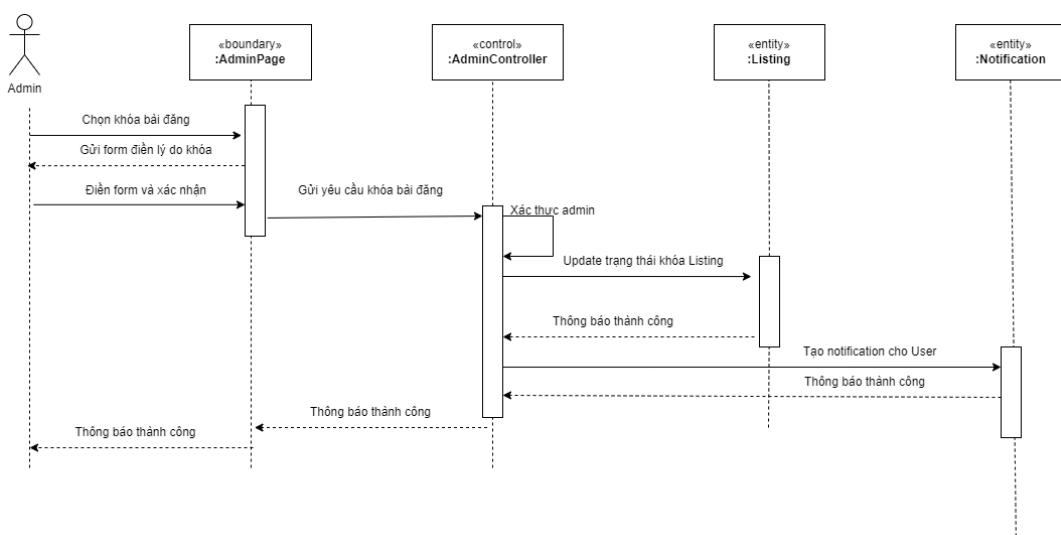
Hình 2.22: Biểu đồ tuần tự "Xem chi tiết bài đăng"

2.4.20 Biểu đồ tuần tự "Duyệt bài đăng"



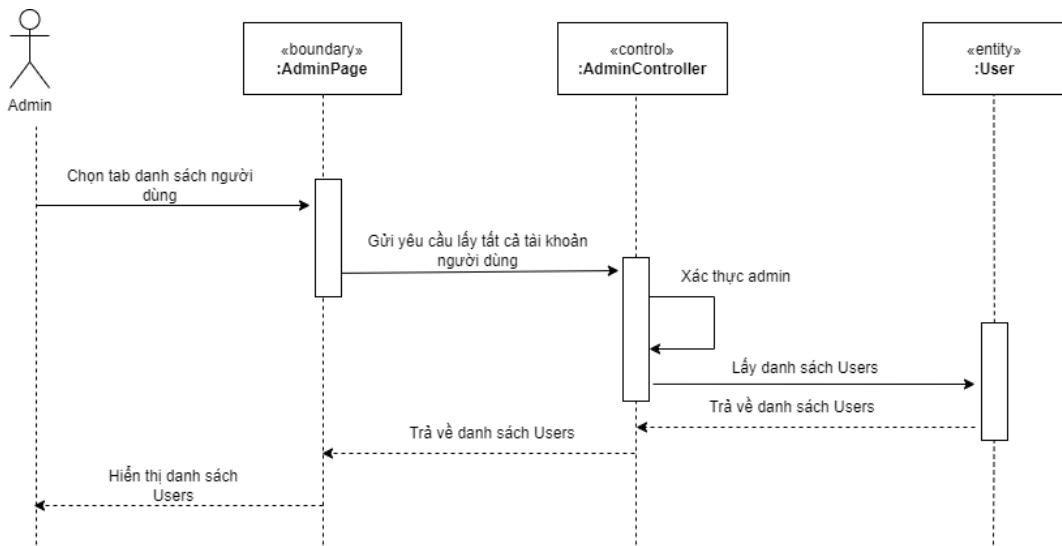
Hình 2.23: Biểu đồ tuần tự "Duyệt bài đăng"

2.4.21 Biểu đồ tuần tự "Khóa bài đăng"



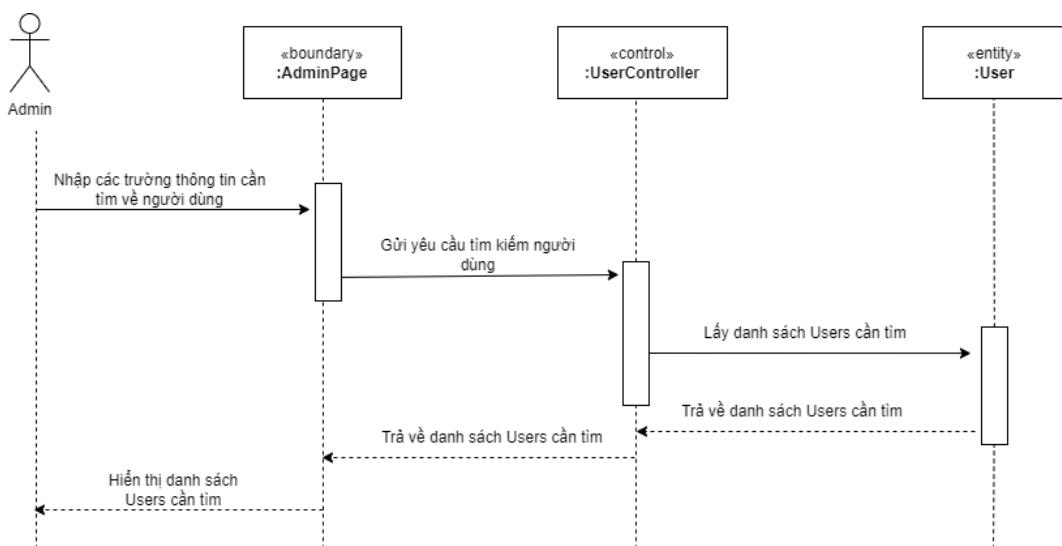
Hình 2.24: Biểu đồ tuần tự "Khóa bài đăng"

2.4.22 Biểu đồ tuần tự "Xem danh sách người dùng"



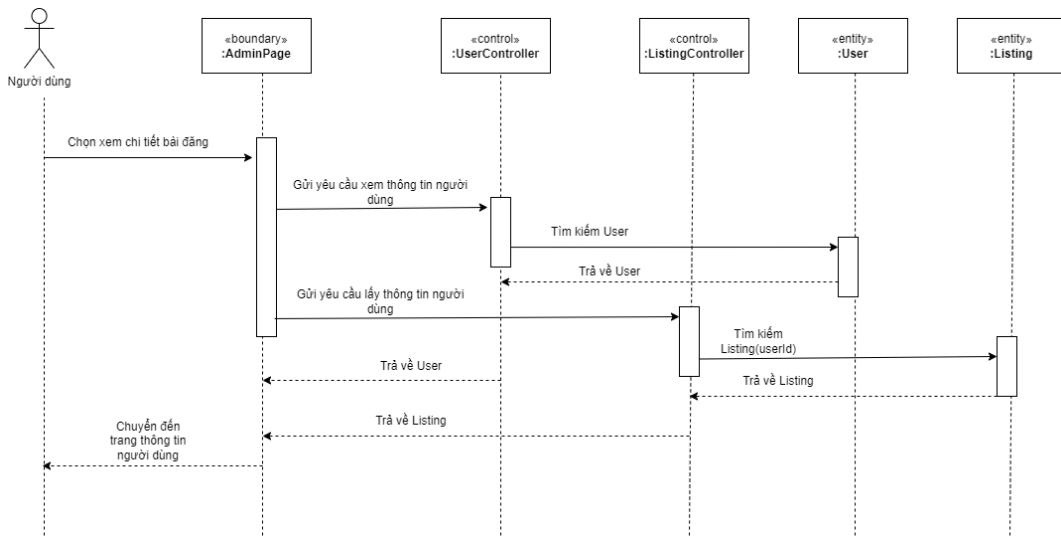
Hình 2.25: Biểu đồ tuần tự "Xem danh sách người dùng"

2.4.23 Biểu đồ tuần tự "Tìm kiếm"



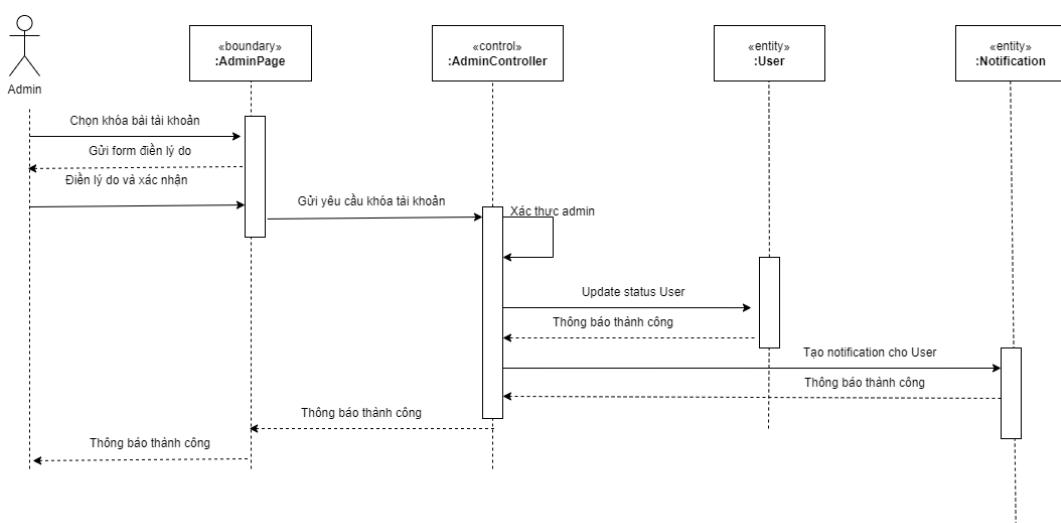
Hình 2.26: Biểu đồ tuần tự "Tìm kiếm"

2.4.24 Biểu đồ tuần tự "Xem chi tiết người dùng"



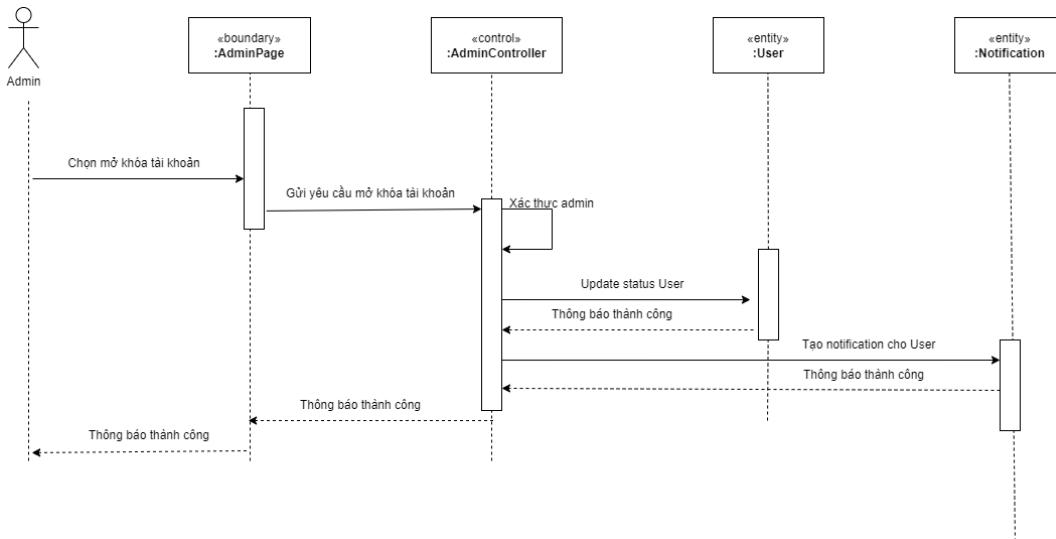
Hình 2.27: Biểu đồ tuần tự "Xem chi tiết người dùng"

2.4.25 Biểu đồ tuần tự "Khóa tài khoản"



Hình 2.28: Biểu đồ tuần tự "Khóa tài khoản"

2.4.26 Biểu đồ tuần tự "Mở tài khoản"



Hình 2.29: Biểu đồ tuần tự "Mở tài khoản"

2.5 Yêu cầu phi chức năng

2.5.1 Yêu cầu về bảo mật

Các hệ thống tìm kiếm bất động sản nói chung và chung cư mini nói riêng hiện nay yêu cầu về bảo mật, hiệu năng và giao diện khá cao nên em đề ra các yêu cầu cho hệ thống mình như sau:

- Các usecase do người dùng, admin thì khách cần đăng nhập với vai trò tương ứng.
- Phương thức đăng nhập sử dụng token và hàm băm SHA256 để chuyển đổi mật khẩu thành ký tự có độ dài cố định

2.5.2 Yêu cầu về hiệu năng

- Thời gian phản hồi của hệ thống tối đa 3s (cùng với xử lý loading phía Frontend của người dùng).
- Định dạng ngày được sử dụng trong hệ thống là dd/mm/yyyy.
- Đảm bảo tính năng thông báo, chat theo thời gian thực

2.5.3 Yêu cầu về giao diện

- Giao diện web thân thiện, trực quan.
- Đáp ứng, thích nghi tốt với các loại màn hình kích cỡ khác nhau.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

3.1 Các công cụ sử dụng trong hệ thống

3.1.1 Postman

Postman là một công cụ mạnh mẽ để kiểm tra, phát triển và test các API (Application Programming Interface). Postman cho phép:

- Gửi các request HTTP: Bạn có thể gửi mọi loại request HTTP (GET, POST, PUT, DELETE,...) đến bất kỳ API nào.
- Quản lý các request: Tổ chức các request vào các collections để dễ dàng quản lý và tái sử dụng.
- Tạo các environment: Đặt các biến môi trường để thay đổi các thông số trong request mà không cần chỉnh sửa thủ công.
- Test và debug API: Kiểm tra xem API có hoạt động đúng như mong đợi hay không, xem response trả về và debug các lỗi.

Với các tính năng trên, Postman như là một công cụ không thể thiếu giúp em trong quá trình xây dựng hệ thống.

3.1.2 MongoDB Compass

MongoDB Compass là một GUI (Graphical User Interface) để quản lý và truy vấn cơ sở dữ liệu MongoDB. Nó hỗ trợ người dùng trong việc quản lý cơ sở dữ liệu MongoDB một cách trực quan và hiệu quả, kiểm tra và debug dữ liệu trong quá trình xây dựng hệ thống. Cụ thể như:

- Xem dữ liệu: Người dùng có thể dễ dàng duyệt qua các collections và xem dữ liệu bên trong.
- Tìm kiếm dữ liệu: Sử dụng các câu lệnh tìm kiếm để tìm ra dữ liệu mong muốn.
- Chỉnh sửa dữ liệu: Thêm, sửa hoặc xóa dữ liệu trực tiếp trong giao diện.
- Visualize dữ liệu: Hiển thị dữ liệu dưới dạng biểu đồ để dễ dàng phân tích.

3.1.3 draw.io

draw.io là một công cụ vẽ đồ thị trực tuyến miễn phí cho phép người dùng tạo nhiều loại sơ đồ, bao gồm sơ đồ luồng, sơ đồ mạng, sơ đồ UML, maquette và hơn thế nữa. Công cụ này cung cấp một thư viện lớn các hình dạng và mẫu có sẵn, người dùng có thể lưu các sơ đồ của mình ở nhiều định dạng khác nhau, bao gồm XML, PNG, JPG và PDF. Draw.io có thể được sử dụng để:

- Vẽ các biểu đồ use case, biểu đồ tuần tự, quy trình nghiệp vụ.
- Vẽ thiết kế cấu trúc hệ thống, các biểu đồ lớp, biểu đồ quan hệ thực thể, biểu đồ thiết kế cơ sở dữ liệu.
- Vẽ mẫu giao diện cho các trang cho client.

3.2 Visual Studio Code

Visual Studio Code (thường được gọi tắt là VS Code) là một trình soạn thảo mã nguồn mạnh mẽ và phổ biến do Microsoft phát triển. Nó được yêu thích bởi các nhà phát triển trên toàn thế giới nhờ sự kết hợp tuyệt vời giữa tính đơn giản và các tính năng mạnh mẽ, hỗ trợ đa nền tảng (Windows, macOS, Linux) và hoàn toàn miễn phí.

Một số điểm nổi bật có thể kể đến của VS Code:

- Hỗ trợ đa ngôn ngữ: VS Code hỗ trợ rất nhiều ngôn ngữ lập trình, từ JavaScript, Python, Java đến C++, Go, PHP và nhiều ngôn ngữ khác.
- IntelliSense: Tính năng tự động hoàn thành mã thông minh giúp người sử dụng viết code nhanh hơn và ít lỗi hơn. IntelliSense cung cấp các gợi ý dựa trên ngữ cảnh, bao gồm các biến, hàm và các thành phần khác.
- Debugging: VS Code tích hợp sẵn trình gõ lỗi mạnh mẽ, cho phép tìm và sửa lỗi trong mã một cách hiệu quả. Người dùng có thể đặt breakpoint, kiểm tra biến và bước qua từng dòng code.
- Tích hợp Git: VS Code có tích hợp sẵn Git, người sử dụng dễ dàng quản lý mã nguồn và cộng tác với những người khác. Ngoài ra, người dùng có thể thực hiện các thao tác Git phổ biến ngay trong VS Code, chẳng hạn như commit, push, pull và merge.
- Extensions: VS Code có một kho tiện ích mở rộng phong phú, cho phép tùy chỉnh và mở rộng chức năng của nó từ hỗ trợ ngôn ngữ lập trình mới đến các công cụ để cải thiện năng suất.
- Giao diện tùy chỉnh: VS Code cho phép tùy chỉnh giao diện theo ý thích, bao gồm chủ đề màu sắc, font chữ và bố cục.
- Terminal tích hợp: VS Code có một terminal tích hợp sẵn, cho phép chạy các lệnh mà không cần phải rời khỏi trình soạn thảo.

3.3 Ngôn ngữ lập trình Javascript

JavaScript là một ngôn ngữ lập trình phổ biến, đóng vai trò quan trọng trong việc phát triển web hiện đại. Nó được biết đến với khả năng mang lại sự tương tác

và động lực cho các trang web, biến chúng từ những trang tĩnh thành những trải nghiệm trực tuyến hấp dẫn. Một số đặc điểm nổi bật của JavaScript có thể kể đến như: Cú pháp đơn giản, là ngôn ngữ kịch bản phía máy khách nên rất phù hợp để xây dựng client project. Ngoài ra, JavaScript được hỗ trợ bởi cộng đồng lớn các nhà phát triển, từ đó có thể dễ dàng tìm thấy sự hỗ trợ, tài liệu và thư viện mã nguồn mở khi cần thiết.

Ứng dụng phổ biến của JavaScript:

- Xây dựng giao diện người dùng web.
- Phát triển ứng dụng di động sử dụng các framework như React Native hay Ionic.
- Phát triển game bằng các thư viện như Phaser và PixiJS.
- Xử lý dữ liệu phía máy chủ, xây dựng API và các ứng dụng backend với Node.js.

3.4 Framework Backend

3.4.1 NodeJS

Node.js là một môi trường thực thi JavaScript mã nguồn mở, đa nền tảng, cho phép chạy mã JavaScript bên ngoài trình duyệt web. Điều này mở ra cánh cửa cho việc sử dụng JavaScript để xây dựng ứng dụng phía máy chủ, tương tác với hệ thống tệp, cơ sở dữ liệu và mạng. Node.js sử dụng mô hình I/O không đồng bộ (non-blocking I/O) và kiến trúc hướng sự kiện, cho phép xử lý nhiều yêu cầu đồng thời mà không bị chặn. Điều này làm cho Node.js trở nên lý tưởng cho các ứng dụng thời gian thực, yêu cầu hiệu suất cao như ứng dụng chat, streaming, và API. Ngoài ra Node.js đi kèm với một thư viện mô-đun phong phú, cung cấp các công cụ và chức năng cần thiết để xây dựng các ứng dụng web, từ xử lý HTTP, làm việc với hệ thống tệp, đến kết nối cơ sở dữ liệu.

3.4.2 ExpressJS

Express.js là một framework web tối giản, được xây dựng trên nền tảng Node.js. Nó cung cấp một tập hợp các tính năng và công cụ mạnh mẽ để đơn giản hóa việc xây dựng ứng dụng web và API. Express.js cung cấp một hệ thống định tuyến linh hoạt, cho phép người dùng xác định các tuyến đường (routes) để xử lý các yêu cầu HTTP khác nhau đến các phần khác nhau của ứng dụng. Bên cạnh đó Express.js sử dụng middleware, là các hàm được thực thi theo thứ tự để xử lý yêu cầu và phản hồi. Middleware có thể được sử dụng cho nhiều mục đích, chẳng hạn như xác thực, ghi log, và xử lý lỗi. ExpressJS đóng vai trò quan trọng trong việc phát triển server, cụ thể là:

- Tạo cấu trúc ứng dụng Server rõ ràng.
- Xử lý yêu cầu và phản hồi HTTP một cách dễ dàng.
- Khả năng kết nối với các cơ sở dữ liệu khác nhau, đặc biệt là MongoDB.

3.5 MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở, nổi tiếng với tính linh hoạt, khả năng mở rộng và hiệu suất cao. Nó được thiết kế để xử lý các tập dữ liệu lớn, có cấu trúc linh hoạt và thường được sử dụng trong các ứng dụng web hiện đại, ứng dụng di động và các hệ thống lớn.

Các đặc điểm nổi bật:

- Hướng tài liệu: MongoDB lưu trữ dữ liệu dưới dạng các tài liệu JSON (JavaScript Object Notation). Các tài liệu này có cấu trúc linh hoạt, cho phép người dùng lưu trữ các loại dữ liệu khác nhau trong cùng một bộ sưu tập mà không cần phải xác định lược đồ cố định trước.
- Schema-less: Không giống như các cơ sở dữ liệu quan hệ truyền thống, MongoDB không yêu cầu người dùng phải định nghĩa lược đồ trước khi lưu trữ dữ liệu. Điều này mang lại sự linh hoạt cao, cho phép thay đổi cấu trúc dữ liệu khi cần thiết.
- Khả năng mở rộng: MongoDB được thiết kế để mở rộng theo chiều ngang, cho phép người dùng dễ dàng thêm nhiều máy chủ vào hệ thống để xử lý lượng dữ liệu và lưu lượng truy cập ngày càng tăng.
- Hiệu suất cao: MongoDB cung cấp hiệu suất cao cho các hoạt động đọc và ghi dữ liệu. Nó sử dụng kỹ thuật lưu trữ dữ liệu hiệu quả và hỗ trợ các chỉ mục để tăng tốc độ truy vấn.
- Truy vấn linh hoạt: MongoDB hỗ trợ một ngôn ngữ truy vấn phong phú, cho phép người dùng thực hiện các truy vấn phức tạp và tìm kiếm dữ liệu theo nhiều tiêu chí khác nhau.

3.6 Framework Frontend

3.6.1 ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, được sử dụng rộng rãi để xây dựng giao diện người dùng (UI) tương tác và động cho các ứng dụng web. Nó tập trung vào việc tạo ra các thành phần UI có thể tái sử dụng, giúp đơn giản hóa quá trình phát triển và bảo trì ứng dụng.

Các đặc điểm chính của ReactJS:

- Component-based: ReactJS cho phép chia nhỏ giao diện người dùng thành các thành phần nhỏ, độc lập và có thể tái sử dụng. Mỗi thành phần có logic và giao diện riêng, giúp dễ dàng quản lý và bảo trì code.
- Virtual DOM: ReactJS sử dụng Virtual DOM (DOM ảo) để tối ưu hóa hiệu suất. Khi có sự thay đổi dữ liệu, ReactJS sẽ cập nhật Virtual DOM trước, sau đó so sánh với DOM thật và chỉ cập nhật những phần thay đổi trên DOM thật. Điều này giúp giảm thiểu số lượng thao tác trực tiếp trên DOM, từ đó tăng hiệu suất ứng dụng.
- JSX: ReactJS sử dụng JSX, một cú pháp mở rộng cho phép bạn viết HTML trực tiếp trong JavaScript. Điều này giúp code dễ đọc và dễ viết hơn.
- Hỗ trợ bởi cộng đồng lớn: ReactJS có một cộng đồng nhà phát triển đông đảo và tích cực, cung cấp nhiều hỗ trợ, tài liệu và thư viện mã nguồn mở.

Từ những đặc điểm trên, em quyết định chọn ReactJS làm framework để xây dựng ứng dụng Client Website trong đồ án của mình.

3.7 Framework Tailwind CSS

Tailwind CSS là một framework CSS "utility-first" đang ngày càng phổ biến, mang đến cách tiếp cận khác biệt so với các framework truyền thống như Bootstrap hay Foundation. Thay vì cung cấp các thành phần UI được thiết kế sẵn, Tailwind CSS cung cấp một tập hợp khổng lồ các lớp tiện ích (utility classes) cấp thấp, cho phép người dùng xây dựng giao diện người dùng trực tiếp trong HTML.

Các đặc điểm nổi bật của Tailwind CSS:

- Tiện ích là trên hết (Utility-first): Tailwind CSS tập trung vào việc cung cấp các lớp tiện ích nhỏ, có mục đích cụ thể, ví dụ như p-4 (padding 4), text-blue-500 (màu chữ xanh lam), rounded-lg (bo tròn góc lớn). Bạn kết hợp các lớp này trực tiếp trong HTML để tạo kiểu cho các phần tử.
- Tùy chỉnh cao: Tailwind CSS cho phép tùy chỉnh mọi thứ, từ màu sắc, font chữ, kích thước, khoảng cách đến breakpoint, thông qua file cấu hình tailwind.config.js.
- Hiệu suất cao: Tailwind CSS loại bỏ CSS không sử dụng trong quá trình build, giúp giảm kích thước file CSS cuối cùng và tăng tốc độ tải trang.
- Dễ sử dụng: Cú pháp đơn giản, dễ nhớ và dễ sử dụng, giúp bạn nhanh chóng làm quen và xây dựng giao diện.
-

So sánh Tailwind và các Framework CSS truyền thống:

Đặc điểm	Tailwind CSS	Framework truyền thống (ví dụ: Bootstrap)
Cách tiếp cận	Utility-first	Component-based
Tùy chỉnh	Cao	Hạn chế
Kích thước file CSS	Nhỏ gọn	Lớn
Tốc độ phát triển	Nhanh	Trung bình
Khả năng bảo trì	Cao	Trung bình

Bảng 3.1: So sánh Tailwind CSS với các framework CSS truyền thống

3.8 Thư viện Ant-Design

Ant Design là một hệ thống thiết kế (Design System) mã nguồn mở phổ biến, được phát triển bởi Ant Financial (thuộc Alibaba Group). Nó cung cấp một bộ sưu tập phong phú các thành phần UI (User Interface) chất lượng cao, được thiết kế sẵn và có thể tùy chỉnh, giúp bạn xây dựng giao diện người dùng đẹp mắt, nhất quán và thân thiện với người dùng cho các ứng dụng web.

Các đặc điểm nổi bật của Ant Design:

- Hệ thống thiết kế hoàn chỉnh: Ant Design không chỉ là một thư viện UI đơn thuần, mà còn cung cấp một hệ thống thiết kế đầy đủ, bao gồm các nguyên tắc thiết kế, các mẫu thiết kế, các thành phần UI và các công cụ hỗ trợ.
- Nhiều thành phần UI: Ant Design cung cấp một bộ sưu tập lớn các thành phần UI, từ các thành phần cơ bản như button, input, icon đến các thành phần phức tạp như table, form, modal, v.v.
- Thiết kế đẹp mắt và hiện đại: Các thành phần UI của Ant Design được thiết kế theo phong cách hiện đại, tối giản và dễ sử dụng, mang lại trải nghiệm người dùng tốt.
- Hỗ trợ nhiều framework: Ant Design hỗ trợ nhiều framework JavaScript phổ biến như React, Angular, Vue.js.

3.9 Google Firebase Storage

Google Firebase Storage là một dịch vụ lưu trữ đối tượng mạnh mẽ, đơn giản và tiết kiệm chi phí, được xây dựng trên nền tảng Google Cloud Storage. Nó cung cấp một giải pháp lưu trữ an toàn và khả năng mở rộng cho các ứng dụng di động và web, giúp người dùng lưu trữ và phân phối các tệp như hình ảnh, video, âm thanh và các tệp khác.

Các tính năng chính của Firebase Storage:

- Tải lên và tải xuống các tệp: Dễ dàng tải lên và tải xuống các tệp từ ứng dụng của bạn.

- Quản lý các tệp: Tổ chức các tệp của bạn vào các thư mục và thực hiện các hoạt động như xóa, di chuyển, đổi tên.
- Kiểm soát truy cập: Thiết lập các quy tắc truy cập để xác định ai có thể đọc, viết và xóa các tệp.
- Mã hóa dữ liệu: Mã hóa dữ liệu khi đang lưu trữ và truyền đi để bảo vệ dữ liệu của bạn.
- Tích hợp với các thư viện bên thứ ba: Firebase Storage hỗ trợ tích hợp với các thư viện bên thứ ba để thực hiện các tác vụ xử lý hình ảnh, video.

3.10 Google Map API

Google Maps API là một tập hợp các giao diện lập trình ứng dụng (API) do Google phát triển, cho phép các nhà phát triển tích hợp bản đồ Google và các tính năng liên quan vào các ứng dụng web và di động của họ. API này cung cấp nhiều chức năng phong phú, từ hiển thị bản đồ đơn giản đến tạo ra các ứng dụng bản đồ phức tạp với các tính năng tùy chỉnh.

Các tính năng chính của Google Maps API:

- Hiển thị bản đồ: Hiển thị bản đồ với các tùy chọn tùy chỉnh về kiểu dáng, chế độ xem, khu vực hiển thị.
- Đánh dấu vị trí: Đánh dấu các vị trí cụ thể trên bản đồ bằng cách sử dụng điểm đánh dấu (marker), thêm thông tin chi tiết cho từng điểm.
- Tìm kiếm địa điểm: Tìm kiếm địa điểm theo tên, địa chỉ hoặc loại hình kinh doanh.
- Chỉ đường: Xác định tuyến đường giữa hai hoặc nhiều điểm, hiển thị thông tin về khoảng cách, thời gian di chuyển và hướng dẫn chi tiết.

Google Maps API có thể được tích hợp vào hệ thống tìm kiếm chung cư mini để cung cấp nhiều tính năng hữu ích, nâng cao trải nghiệm người dùng và tăng tính hiệu quả của hệ thống như hiển thị vị trí chung cư trên bản đồ, chỉ đường đến chung cư, tìm kiếm chung cư xung quanh theo vị trí.

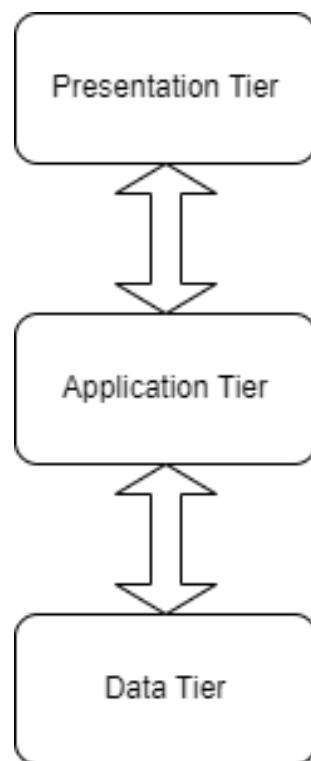
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

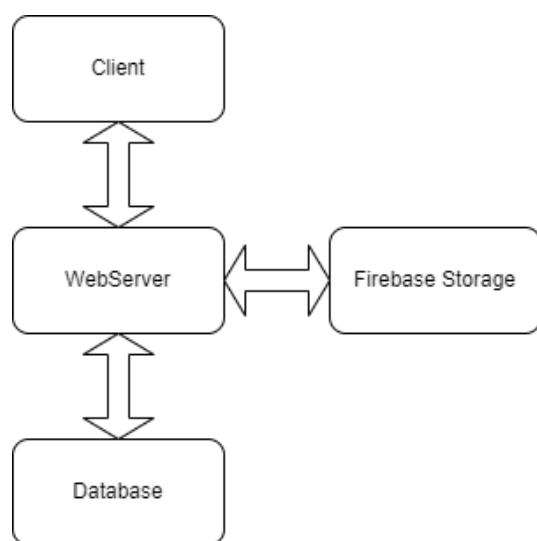
Xuyên suốt quá trình xây dựng hệ thống, em đã tuân theo mô hình kiến trúc ba tầng (Three-Tier). Kiến trúc ba tầng là một kiến trúc ứng dụng phần mềm được thiết kế để tổ chức ứng dụng, chia ứng dụng thành các tầng chạy trên cơ sở hạ tầng của riêng mình, mỗi tầng có thể được phát triển đồng thời bởi một nhóm phát triển mà không ảnh hưởng đến tầng khác. Kiến trúc ba tầng trong Hình 4.1 bao gồm các thành phần:

- Tầng trình bày (Presentation tier): là giao diện người dùng và tầng giao tiếp của ứng dụng.
- Tầng ứng dụng (Application tier): là tầng logic và trung tâm của ứng dụng. Nơi thu nhận thông tin và xử lý nghiệp vụ. Tầng ứng dụng có thể tương tác sửa đổi dữ liệu trong tầng ứng dụng. Giao tiếp với tầng ứng dụng bằng cách gọi API.
- Tầng dữ liệu (Data tier): là tầng truy cập dữ liệu, nơi lưu trữ và quản lý thông tin được ứng dụng xử lý. Tầng trình bày và tầng dữ liệu không thể giao tiếp trực tiếp với nhau.



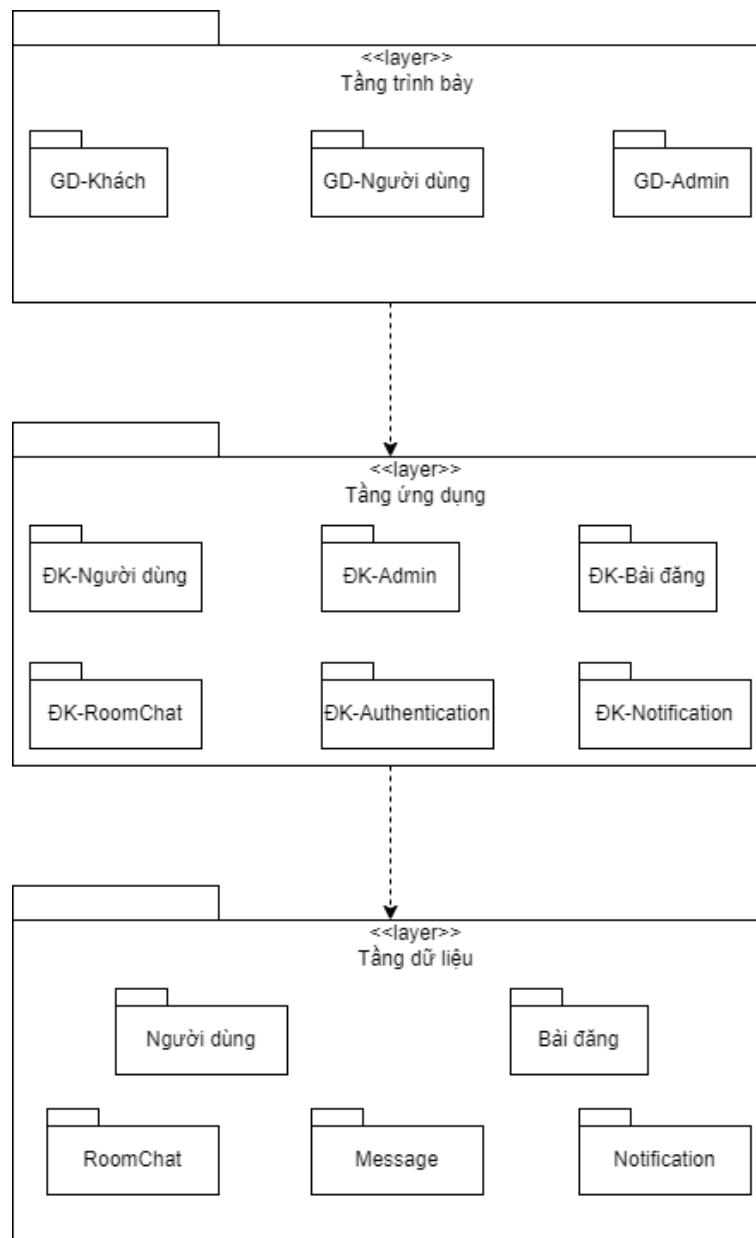
Hình 4.1: Kiến trúc MVC

Dựa trên kiến trúc ba tầng, Hình 4.2 mô tả kiến trúc hệ thống trong đề tài em xây dựng. Tầng Client có vai trò như tầng trình bày, là nơi người dùng giao tiếp với ứng dụng. Tầng ứng dụng tương ứng với Web Server tương ứng chứa các nghiệp vụ về hệ thống tìm kiếm và cho thuê chung cư mini. Tầng Database có vai trò như tầng dữ liệu, là nơi lưu trữ dữ liệu của hệ thống. Qua đó, Client phải giao tiếp với tầng ứng dụng chứa dịch vụ xử lý nghiệp vụ thông qua các API và Client không thể tương tác trực tiếp.



Hình 4.2: Kiến trúc hệ thống

4.1.2 Thiết kế tổng quan



Hình 4.3: Kiến trúc tổng thể

Tầng trình bày gồm

- GD-Khách:

- Đăng nhập
- Đăng ký
- Quên mật khẩu

- GD-Người dùng:

- Tìm kiếm bài đăng
- Trang thông tin bài đăng

- Trang thông tin người dùng
- Cập nhật thông tin người dùng
- Quản lý bài đăng

- GD-Admin:

- Quản lý bài đăng
- Quản lý bài đăng cần duyệt
- Quản lý tài khoản người dùng

Tầng ứng dụng gồm

- ĐK-Người dùng:

- Lấy thông tin người dùng
- Cập nhật thông tin người dùng
- Xóa tài khoản người dùng
- Lấy danh sách bài đăng của người dùng
- Lấy danh sách bài đăng yêu thích của người dùng

- ĐK-Admin:

- Lấy danh sách tài khoản trong hệ thống
- Lấy danh sách bài đăng trong hệ thống
- Lấy danh sách bài đăng cần duyệt trong hệ thống
- Duyệt/Từ chối duyệt bài đăng
- Khóa/Mở tài khoản người dùng
- Khóa/Mở bài đăng người dùng

- ĐK-Bài đăng:

- Tạo mới bài đăng
- Cập nhật thông tin bài đăng
- Xóa bài đăng
- Lấy thông tin bài đăng cụ thể của người dùng
- Lấy danh sách bài đăng của người dùng
- Lấy danh sách bài đăng yêu thích của người dùng

- ĐK-RoomChat:

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

- Tạo mới Room Chat
- Lấy thông tin Room Chat
- Xóa Room Chat
- Tạo mới tin nhắn
- Lấy các tin nhắn của Room Chat

- DK-Authentication:

- Đăng nhập
- Đăng ký
- Đăng xuất
- Đăng nhập bằng tài khoản Google

- DK-Notification:

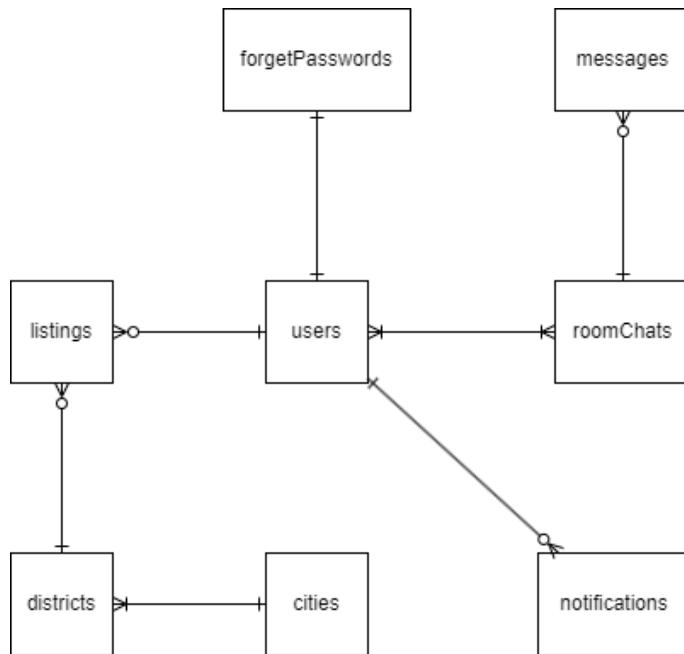
- Tạo mới thông báo
- Lấy thông tin thông báo
- Xóa thông báo

Tầng dữ liệu gồm

- Người dùng
- Bài đăng
- RoomChat
- Message
- Notification

4.2 Thiết kế cơ sở dữ liệu

4.2.1 Biểu đồ quan hệ thực thể (ERD)



Hình 4.4: Biểu đồ quan hệ thực thể

Biểu đồ quan hệ thực thể của hệ thống tại Hình 4.4 thể hiện các mối quan hệ của thực thể trong cơ sở dữ liệu của hệ thống. Mô tả quan hệ các thực thể:

- Một người dùng có thể có nhiều bài đăng hoặc không có.
- Một người dùng có thể có nhiều thông báo hoặc không có.
- Một mã khôi phục mật khẩu chỉ tương ứng với một và chỉ một người dùng.
- Một người dùng có thể có nhiều roomChat và ngược lại.
- Một RoomChat có thể có nhiều tin nhắn hoặc không có.
- Một tỉnh/thành phố có thể có nhiều quận/huyện.
- Một quận/huyện có thể có nhiều bài đăng hoặc không có.

4.2.2 Thiết kế Bảng Users

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	ObjectId	Mã tham chiếu người dùng
2	username	String	Username của người dùng
3	email	String	Email của người dùng
4	password	String	Password của người dùng
5	avatar	String	Avatar của người dùng
6	bio	String	Giới thiệu của người dùng
7	phone	Number	Số điện thoại của người dùng
8	role	String (Enum)	Role của người dùng
9	status	String (Enum)	Trạng thái mở/khóa tài khoản của người dùng
10	isOnline	Boolean	Trạng thái online/offline tài khoản của người dùng
11	socketId	String	ID Socket của người dùng
12	createdAt	DateTime	Ngày tạo
13	updatedAt	DateTime	Ngày cập nhật

Bảng 4.1: Bảng dữ liệu Users

4.2.3 Thiết kế Bảng Listings

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	ObjectId	Mã tham chiếu bài đăng
2	name	String	Tiêu đề bài đăng
3	description	String	Phản giội thiệu bài đăng
4	address	String	Địa chỉ của chung cư
5	regularPrice	Number	Giá gốc của căn chung cư
6	discountPrice	Number	Giá sau khi được giảm của căn chung cư
7	bathrooms	Number	Số phòng tắm của căn chung cư
8	bedrooms	Number	Số phòng ngủ của căn chung cư
9	furnished	Boolean	Được trang bị nội thất không
10	parking	Boolean	Có nơi để đồ xe hay không
11	type	String	Loại của chung cư để bán/cho thuê
12	offer	Boolean	Chung cư có đang được giảm giá không
13	imageUrls	Array	Danh sách ảnh mô tả của căn chung cư
14	userRef	String	Mã tham chiếu đến người đăng tin
15	districtRef	String	Mã tham chiếu đến quận/huyện của chung cư
16	favoriteUserList	Array	Danh sách người dùng yêu thích bài đăng
17	status	String (Enum)	Trạng thái của bài đăng
18	createdAt	DateTime	Ngày tạo
19	updateAt	DateTime	Ngày cập nhật

Bảng 4.2: Bảng dữ liệu Listings

4.2.4 Thiết kế Bảng Districts

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	ObjectId	Mã tham chiếu quận/huyện
2	districtName	String	Tên của quận/huyện
3	districtSlug	String	Slug của quận/ huyện thân thiện với URL
4	cityRef	String	Mã tham chiếu đến thành phố
5	createdAt	DateTime	Ngày tạo
6	updateAt	DateTime	Ngày cập nhật

Bảng 4.3: Bảng dữ liệu Districts

4.2.5 Thiết kế Bảng Cities

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	Object Id	Mã tham chiếu thành phố/tỉnh
2	cityName	String	Tên của tỉnh/thành phố
3	citySlug	String	Slug của tỉnh/thành phố thân thiện với URL
4	createdAt	DateTime	Ngày tạo
5	updateAt	DateTime	Ngày cập nhật

Bảng 4.4: Bảng dữ liệu Cities

4.2.6 Thiết kế Bảng Notifications

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	Object Id	Mã tham chiếu thông báo
2	from	String	Mã tham chiếu của người gửi
3	to	String	Mã tham chiếu của người nhận
4	content	String	Nội dung của thông báo
5	createdAt	DateTime	Ngày tạo
6	updateAt	DateTime	Ngày cập nhật

Bảng 4.5: Bảng dữ liệu Notifications

4.2.7 Thiết kế Bảng RoomChats

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	Object Id	Mã tham chiếu của room chat
2	users	Array	Danh sách người tham gia Room Chat
3	name	String	Tên của RoomChat
5	createdAt	DateTime	Ngày tạo
6	updateAt	DateTime	Ngày cập nhật

Bảng 4.6: Bảng dữ liệu RoomChats

4.2.8 Thiết kế Bảng Messages

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	Object Id	Mã tham chiếu gói tin nhắn
2	roomChatRef	Array	Mã tham chiếu Room Chat
3	size	Number	số tin nhắn trong Room Chat
4	messagesRoomChat	Object	1 Tin nhắn trong Room
	from	String	Mã tham chiếu đến người gửi
	content	String	Nội dung tin nhắn
	read	Boolean	Trạng thái đã xem
	createdAt	DateTime	Ngày tạo
	updatedAt	DateTime	Ngày cập nhật
5	createdAt	DateTime	Ngày tạo
6	updatedAt	DateTime	Ngày cập nhật

Bảng 4.7: Bảng dữ liệu Messages

4.2.9 Thiết kế Bảng ForgetPasswords

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	Object Id	Mã tham chiếu bản ghi khôi phục mật khẩu
2	email	String	Email người gửi yêu cầu khôi phục
3	resetCode	String	Chuỗi GUID (dữ liệu nhị phân 16 bytes) khôi phục mật khẩu
5	createdAt	DateTime	Ngày tạo
6	updatedAt	DateTime	Ngày cập nhật

Bảng 4.8: Bảng dữ liệu RoomChats

4.3 Thiết kế giao diện

4.3.1 Giao diện đăng ký

The screenshot shows a web page layout. At the top is a header section with a search bar labeled "Search Term" and a "Sign In" button. Below the header is a main content area titled "Sign Up". This area contains three input fields: "username...", "email...", and "password...". Below these fields are two buttons: a blue "SIGN UP" button and a red "CONTINUE WITH GOOGLE" button. At the bottom of the page is a footer section.

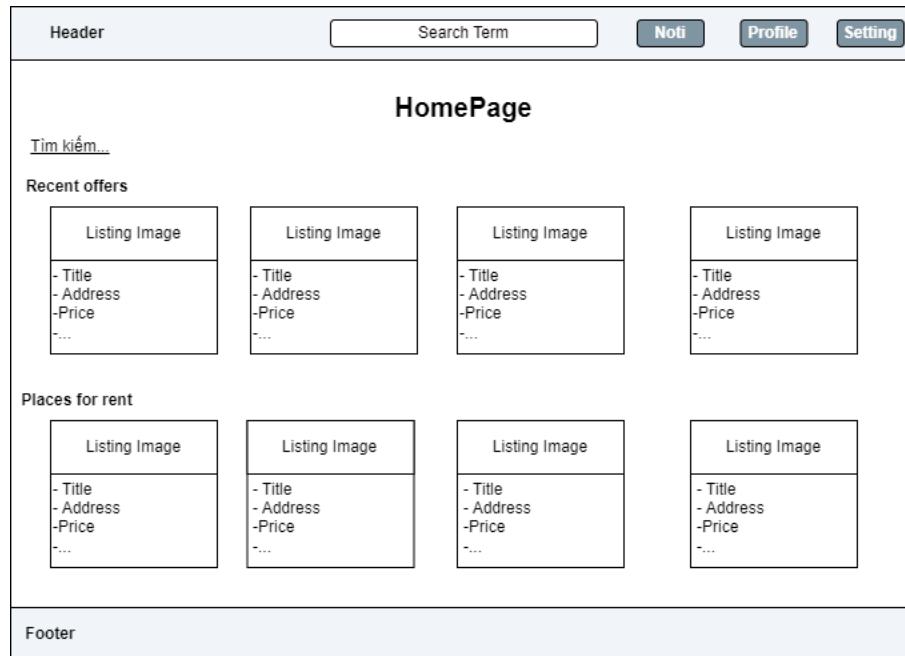
Hình 4.5: GD-Đăng ký

4.3.2 Giao diện đăng nhập

The screenshot shows a web page layout similar to the sign-up page. It features a header with a search bar and a "Sign Up" button. The main content area is titled "Sign In". It includes two input fields for "email..." and "password...". There are two buttons in this section: a blue "SIGN IN" button and a red "CONTINUE WITH GOOGLE" button. A footer section is located at the bottom of the page.

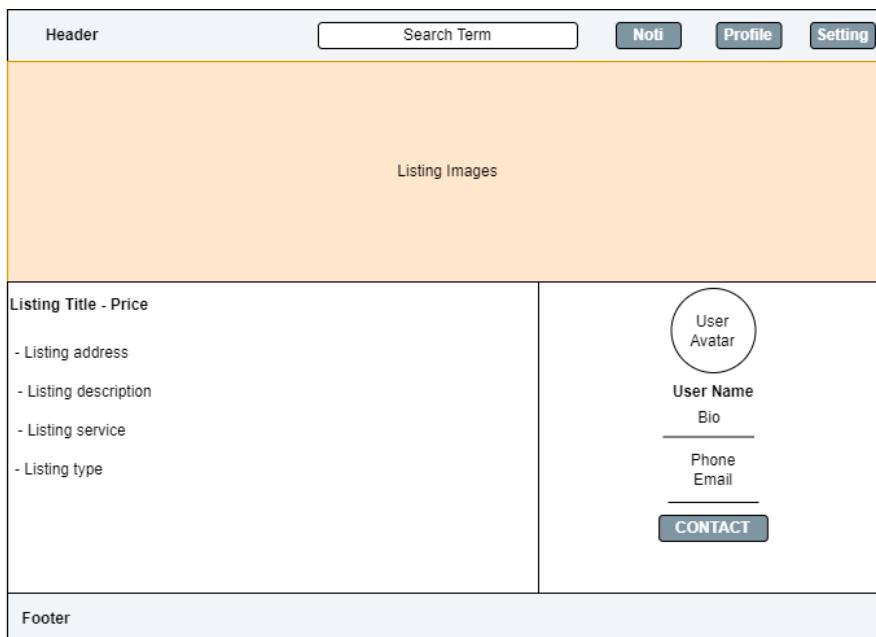
Hình 4.6: GD-Đăng nhập

4.3.3 Giao diện trang chủ



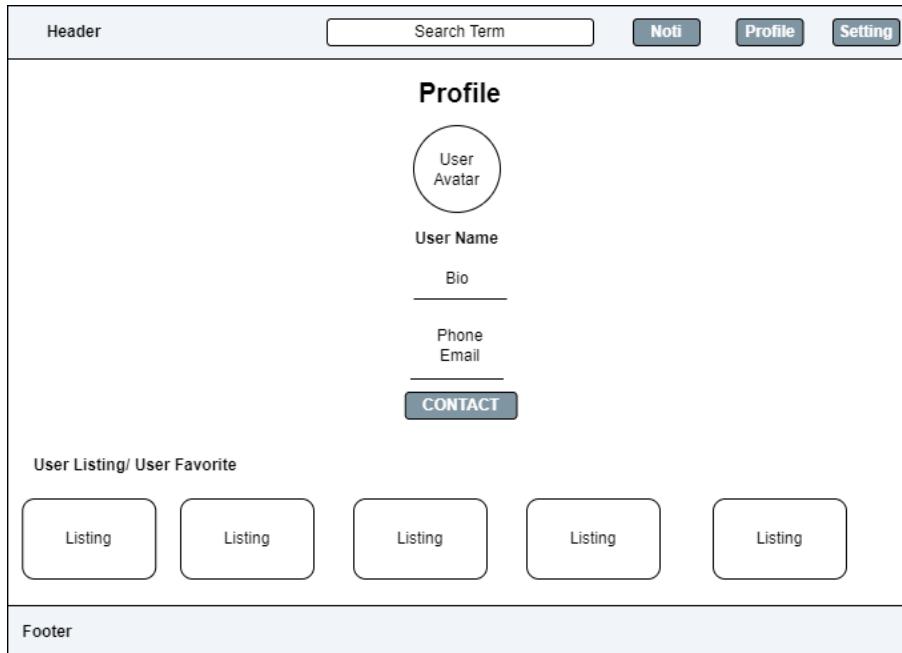
Hình 4.7: GD-Trang chủ

4.3.4 Giao diện bài đăng



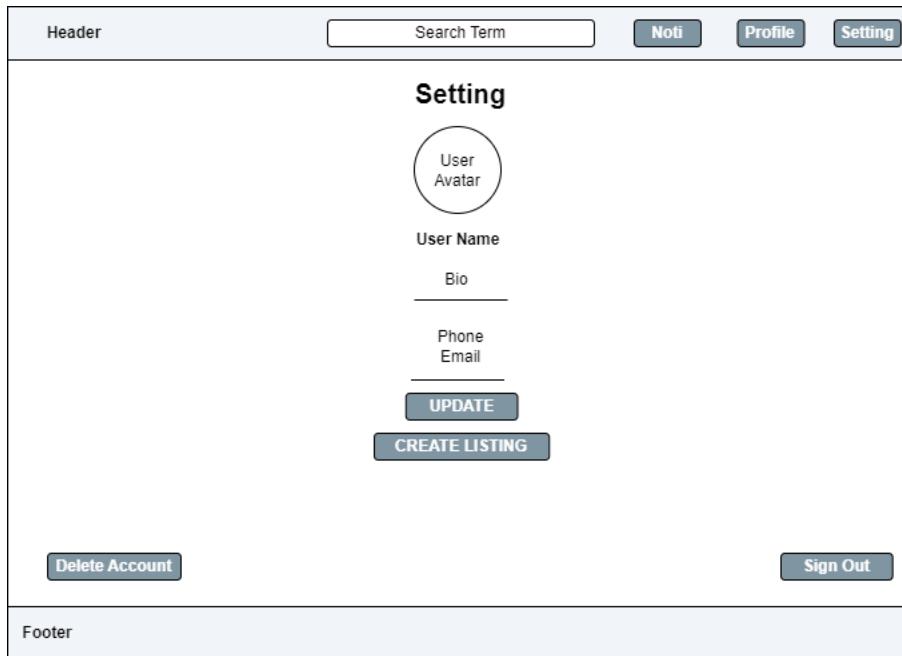
Hình 4.8: GD-Bài đăng

4.3.5 Giao diện Profile người đăng



Hình 4.9: GD-Profile

4.3.6 Giao diện Setting người đăng



Hình 4.10: GD-Profile

4.4 Xây dựng ứng dụng

4.4.1 Thư viện và công cụ sử dụng

Các công cụ, ngôn ngữ lập trình, API, thư viện, IDE, công cụ kiểm thử, mà em sử dụng để làm đồ án:

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Visual Studio Code	https://code.visualstudio.com/
Quản lý CSDL	MongoDB Compass	https://www.mongodb.com/products/tools/compass
Lưu trữ ảnh, video	Google Firebase	https://firebase.google.com/
Kiểm thử API	Postman	https://www.postman.com/
Vẽ biểu đồ thiết kế	draw.io	https://draw.io/

Bảng 4.9: Danh sách thư viện và công cụ sử dụng

Các công nghệ và Framework em sử dụng để xây dựng hệ thống:

a) Front-end

- ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, được sử dụng rộng rãi để xây dựng giao diện người dùng (UI) tương tác và động cho các ứng dụng web. Nó tập trung vào việc tạo ra các thành phần UI có thể tái sử dụng, giúp đơn giản hóa quá trình phát triển và bảo trì ứng dụng.

Các đặc điểm chính của ReactJS:

- Component-based: ReactJS cho phép chia nhỏ giao diện người dùng thành các thành phần nhỏ, độc lập và có thể tái sử dụng. Mỗi thành phần có logic và giao diện riêng, giúp dễ dàng quản lý và bảo trì code.
- Virtual DOM: ReactJS sử dụng Virtual DOM (DOM ảo) để tối ưu hóa hiệu suất. Khi có sự thay đổi dữ liệu, ReactJS sẽ cập nhật Virtual DOM trước, sau đó so sánh với DOM thật và chỉ cập nhật những phần thay đổi trên DOM thật. Điều này giúp giảm thiểu số lượng thao tác trực tiếp trên DOM, từ đó tăng hiệu suất ứng dụng.
- JSX: ReactJS sử dụng JSX, một cú pháp mở rộng cho phép bạn viết HTML trực tiếp trong JavaScript. Điều này giúp code dễ đọc và dễ viết hơn.
- Hỗ trợ bởi cộng đồng lớn: ReactJS có một cộng đồng nhà phát triển đông đảo và tích cực, cung cấp nhiều hỗ trợ, tài liệu và thư viện mã nguồn mở.

- Ngoài ra có Tailwind CSS

- Và Ant-Design

b) Back-end

- Express.js là một framework web tối giản, được xây dựng trên nền tảng Node.js. Nó cung cấp một tập hợp các tính năng và công cụ mạnh mẽ để đơn giản hóa việc xây dựng ứng dụng web và API. Express.js cung cấp một hệ thống định tuyến linh hoạt, cho phép người dùng xác định các tuyến đường (routes) để xử lý các yêu cầu HTTP khác nhau đến các phần khác nhau của ứng dụng. Bên cạnh đó Express.js sử dụng middleware, là các hàm được thực thi theo thứ tự để xử lý yêu cầu và phản

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

hồi. Middleware có thể được sử dụng cho nhiều mục đích, chẳng hạn như xác thực, ghi log, và xử lý lỗi. ExpressJS đóng vai trò quan trọng trong việc phát triển server, cụ thể là:

- Tạo cấu trúc ứng dụng Server rõ ràng.
- Xử lý yêu cầu và phản hồi HTTP một cách dễ dàng.
- Khả năng kết nối với các cơ sở dữ liệu khác nhau, đặc biệt là MongoDB.

c) Database

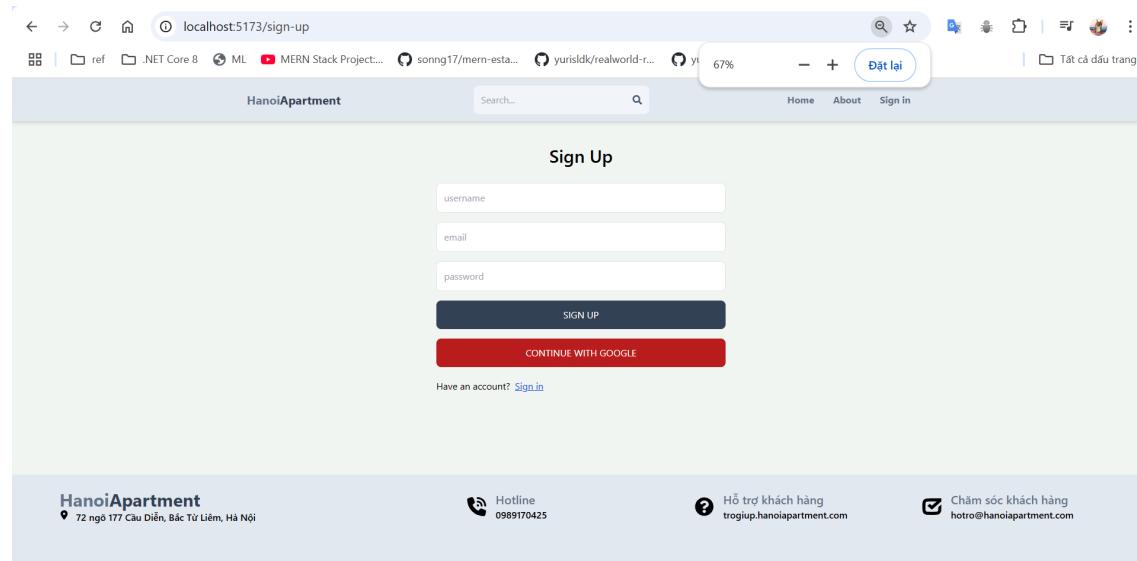
- MongoDB
- Google Firebase Storage

d) Socket.io

Để lập trình ứng dụng giao tiếp trong thời gian thực như thông báo, nhắn tin, em đã sử dụng module Socket.IO cho cả Front-End và Back-End.

4.4.2 Kết quả đạt được

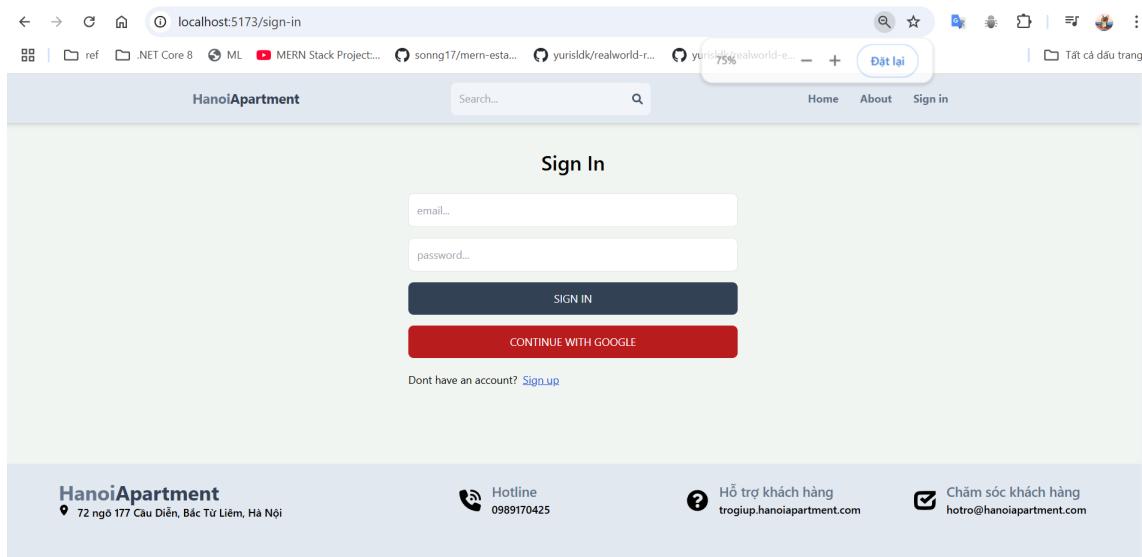
- Giao diện đăng ký:



Hình 4.11: GD-Đăng ký

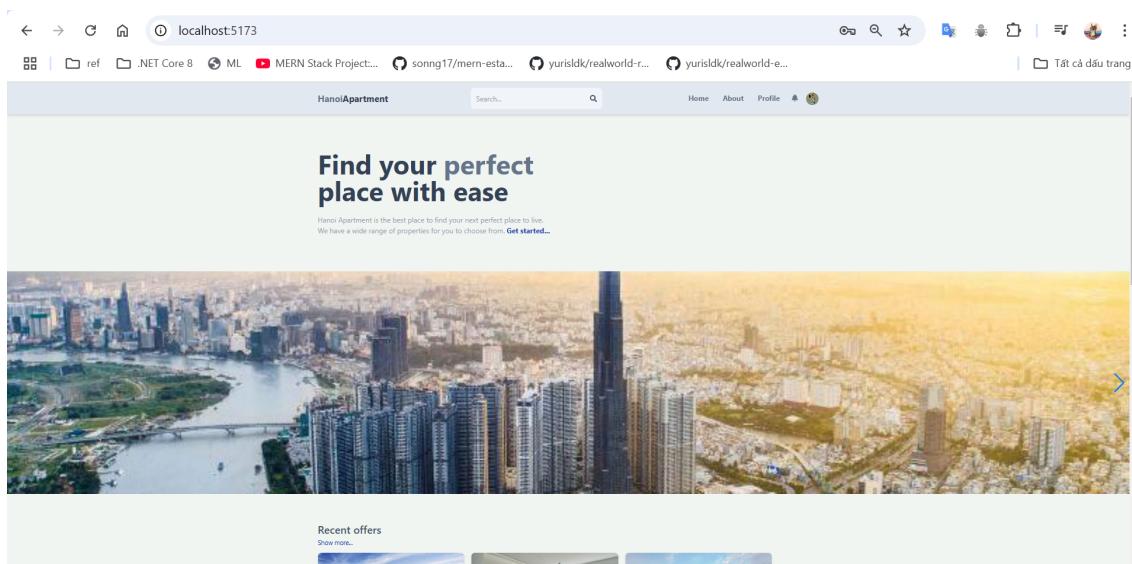
- Giao diện đăng nhập:

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



Hình 4.12: GD-Đăng nhập

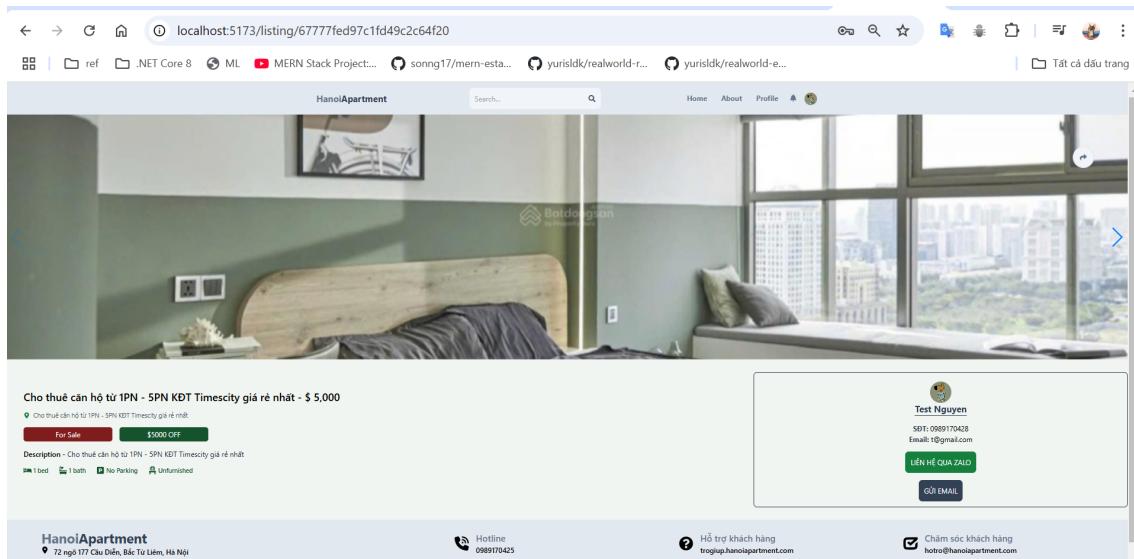
- Giao diện trang chủ:



Hình 4.13: GD-Trang chủ

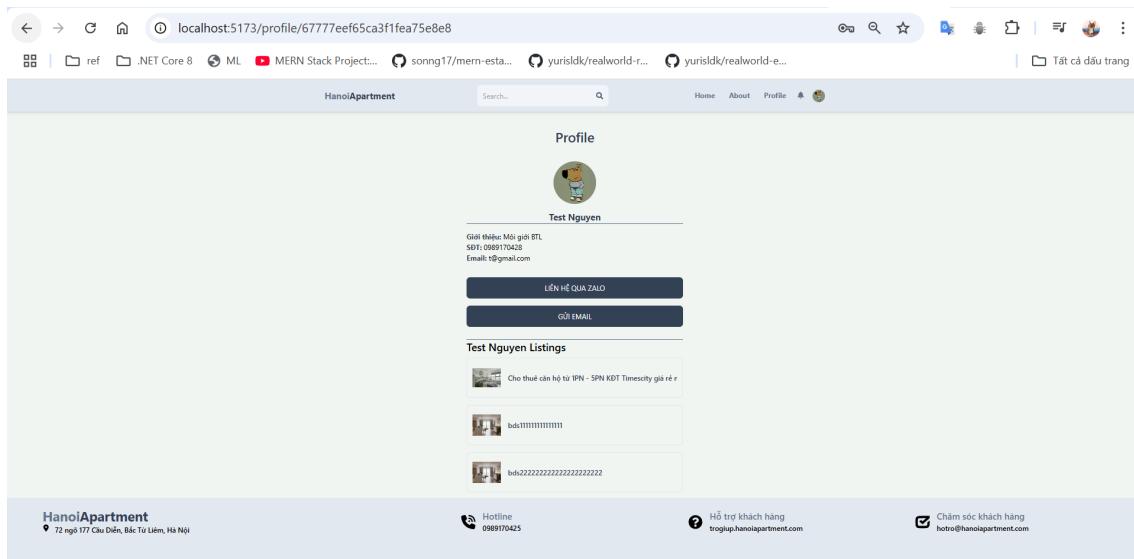
- Giao diện bài đăng:

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



Hình 4.14: GD-Bài đăng

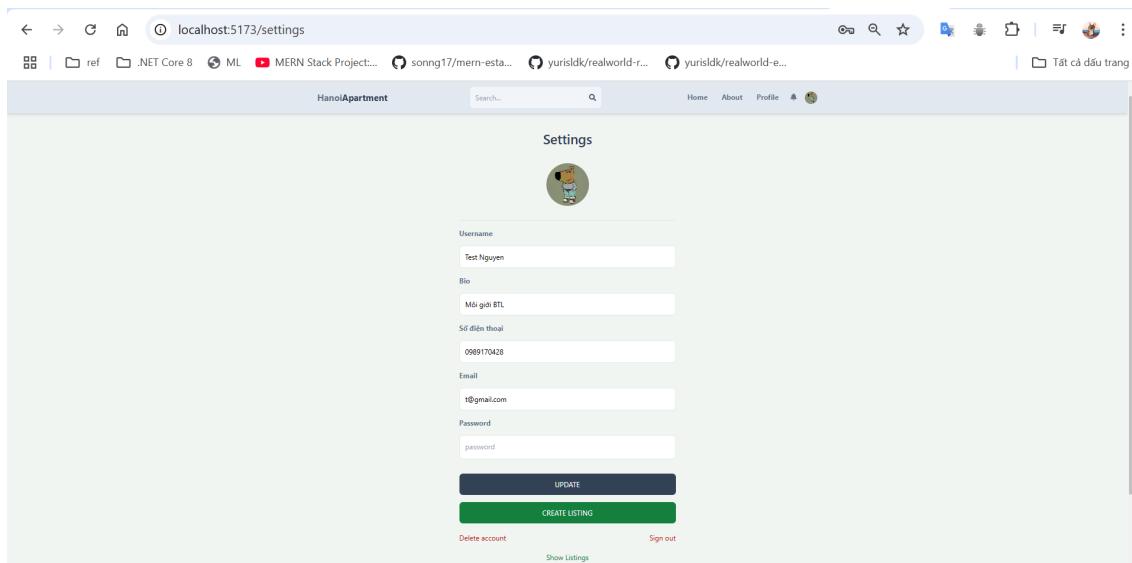
- Giao diện Profile:



Hình 4.15: GD-Profile

- Giao diện Setting:

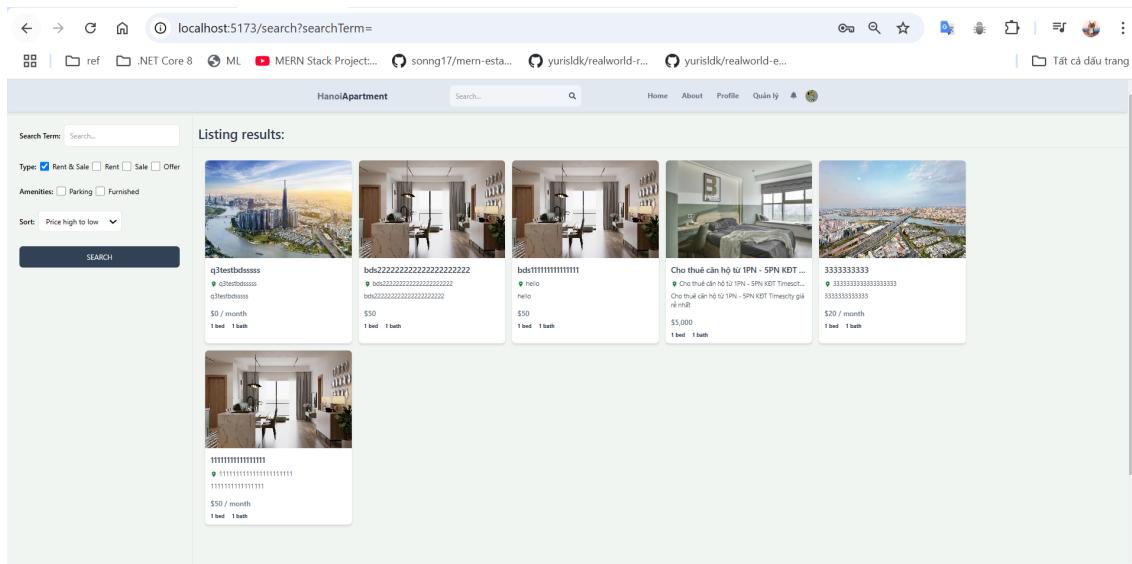
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



Hình 4.16: GD-Setting

4.4.3 Minh họa các chức năng chính

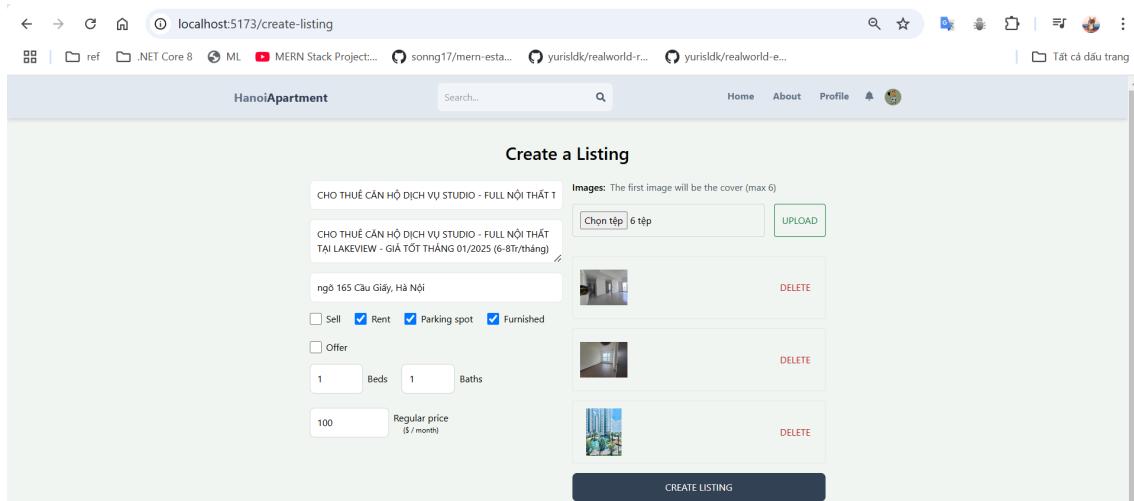
- Tìm kiếm bài đăng:



Hình 4.17: Tìm kiếm bài đăng

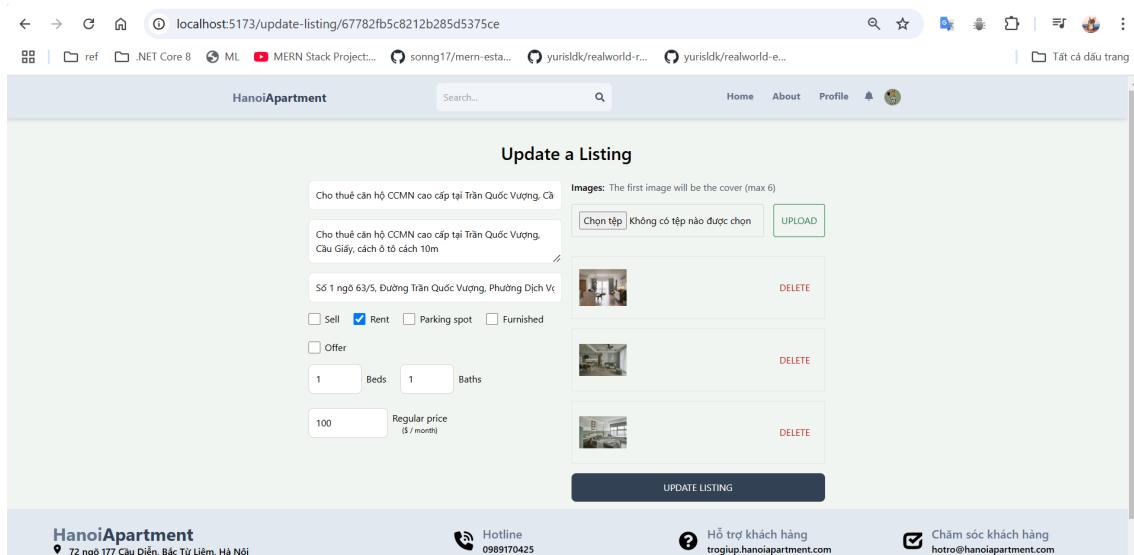
- Đăng bài:

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



Hình 4.18: Tạo bài đăng

- Cập nhật bài đăng:



Hình 4.19: Cập nhật bài đăng

- Quản lý người dùng:

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Avatar	Username	Email	Role	Actions
	test2	test2@gmail.com	user	<button>Get</button> <button>Update</button> <button>Delete</button>
	test4	test4@gmail.com	user	<button>Get</button> <button>Update</button> <button>Delete</button>
	test5	test5@gmail.com	user	<button>Get</button> <button>Update</button> <button>Delete</button>
	test3	test3@gmail.com	user	<button>Get</button> <button>Update</button> <button>Delete</button>
	tieuber1	tieuber1@gmail.com	user	<button>Get</button> <button>Update</button> <button>Delete</button>

Hình 4.20: Quản lý người dùng

- Quản lý bài đăng:

Name	Description	Address	Type	UserRef	status	Actions
1111111111111111	1111111111111111	1111111111111111	rent	6771997ad6b731ac806b7b3f	approved	<button>Get</button> <button>Update</button> <button>Delete</button>
3333333333333333	3333333333333333	3333333333333333	rent	6771997ad6b731ac806b7b3f	approved	<button>Get</button> <button>Update</button> <button>Delete</button>
Cho thuê căn hộ từ 1PN - 5PN KDT Timescity giá rẻ nhất	Cho thuê căn hộ từ 1PN - 5PN KDT Timescity giá rẻ nhất	Cho thuê căn hộ từ 1PN - 5PN KDT Timescity giá rẻ nhất	sale	67777ee65ca3f1fea75e8e8	approved	<button>Get</button> <button>Update</button> <button>Delete</button>
bds11111111111111	hello	hello	sale	67777ee65ca3f1fea75e8e8	approved	<button>Get</button> <button>Update</button> <button>Delete</button>
bds2222222222222222222222	bds22222222222222222222	bds22222222222222222222	sale	67777ee65ca3f1fea75e8e8	approved	<button>Get</button> <button>Update</button> <button>Delete</button>

Hình 4.21: Quản lý bài đăng

4.5 Kiểm thử

Phương pháp em áp dụng kiểm thử trong đồ án tốt nghiệp là kiểm thử thủ công. Sau đây là danh sách các trường hợp kiểm thử hệ thống:

4.5.1 Kiểm thử xác thực

- Kiểm thử đăng ký

Testcase 1: Nhập thiếu một trong các trường (username, email, mật khẩu).

Đầu vào: username: Nguyễn Đình Sơn, mật khẩu: 12345678

Kết quả mong đợi: cảnh báo các trường bắt buộc nhập không được bỏ trống.

Kết quả: Đạt

Testcase 2: Nhập sai định dạng email.

Đầu vào: nguyendinhson Kết quả mong đợi: cảnh báo sai định dạng email.

Kết quả: Đạt

Testcase 3: Nhập trùng email với tài khoản khác đã sử dụng.

Đầu vào: đã tồn tại tài khoản nguyendinhson@gmail.com; nhập email đăng ký là nguyendinhson@gmail.com

Kết quả mong đợi: cảnh báo email đã được sử dụng.

Kết quả: Đạt

- Kiểm thử đăng nhập

Testcase 1: Nhập email không đúng định dạng.

Đầu vào: nguyendinhson@.

Kết quả mong đợi: user not found!

Kết quả: Đạt

Testcase 2: Nhập email chưa đăng ký tài khoản.

Đầu vào: hệ thống chưa ghi nhận đăng ký tài khoản của sondeptraiqua@gmail.com, nhập sondeptraiqua@gmail.com đăng nhập.

Kết quả mong đợi: usernotfound!

Kết quả: Đạt

Testcase 3: Nhập email, không nhập mật khẩu hoặc sai mật khẩu.

Đầu vào: email: nguyendinhson@gmail.com, mật khẩu đúng: 12345678. không nhập hoặc nhập mật khẩu: 123.

Kết quả mong đợi: Thông báo sai thông tin người dùng.

Kết quả: Đạt

- Kiểm thử đăng xuất

Testcase 1: Đăng xuất.

Đầu vào: trạng thái đang đăng nhập, thực hiện đăng xuất khỏi hệ thống, sử dụng các chức năng quay lại trên trình duyệt

Kết quả mong đợi: hệ thống chặn quay trở lại các trang và điều hướng đến trang đăng nhập.

Kết quả: Đạt

4.5.2 Kiểm thử người dùng

- Kiểm thử hiển thị đầy đủ các trang dành cho người dùng

Testcase 1: Trang chủ.

Đầu vào: Truy cập trang chủ

Kết quả mong đợi: Hiển thị đầy đủ danh sách các loại bài đăng

Kết quả: Đạt

Testcase 1: Trang thông tin bài đăng.

Đầu vào: Chọn một bài đăng muốn xem

Kết quả mong đợi: Hiển thị đầy đủ thông tin về bài đăng và người dùng

Kết quả: Đạt

Testcase 1: Trang cá nhân

Đầu vào: Chọn xem trang cá nhân người dùng

Kết quả mong đợi: Hiển thị đầy đủ thông tin người dùng và các bài đăng người dùng đó

Kết quả: Đạt

Testcase 1: Trang setting.

Đầu vào: Truy cập trang setting

Kết quả mong đợi: Hiển thị danh sách bài đăng người dùng có thể tạo, chỉnh sửa, xóa, cập nhật thông tin người dùng

Kết quả: Đạt

4.5.3 Kiểm thử admin

- Kiểm thử quản lý danh sách bài đăng và người dùng

Testcase 1: Trang quản lý bài đăng của admin

Đầu vào: Truy cập trang quản lý bài đăng

Kết quả mong đợi: Hiển thị đầy đủ danh sách các loại bài đăng kèm thông tin và các hành động với chúng

Kết quả: Đạt

Testcase 2: Trang quản lý tài khoản người dùng của admin

Đầu vào: Truy cập trang quản lý tài khoản người dùng

Kết quả mong đợi: Hiển thị đầy đủ danh sách người dùng kèm thông tin và các hành động với tài khoản

Kết quả: Đạt

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Sau quá trình thực hiện đồ án "Xây dựng hệ thống tìm kiếm và cho thuê chung cư mini", em đã hoàn thành các mục tiêu chính đề ra từ ban đầu. Kết quả sản phẩm thử nghiệm đáp ứng được các tính năng cơ bản của một ứng dụng tìm kiếm và cho thuê chung cư mini cùng với một giao diện thân thiện, trực quan, cho phép người dùng dễ dàng tìm kiếm và thuê chung cư mini theo nhu cầu, Xây dựng API mạnh mẽ và bảo mật với Expressjs NodeJS để xử lý dữ liệu và hỗ trợ giao tiếp giữa các thành phần, tích hợp cơ sở dữ liệu MongoDB để quản lý thông tin khách hàng, chủ nhà, và các căn hộ cho thuê một cách hiệu quả. Hệ thống đã được kiểm tra và hoạt động ổn định với các tính năng cơ bản như tìm kiếm, lọc, và quản lý thông tin thuê/cho thuê. Kết quả này không chỉ đáp ứng yêu cầu đặt ra mà còn tạo cơ sở vững chắc cho các ứng dụng thực tế.

Mặc dù đã đạt được những kết quả nhất định, đồ án vẫn còn một số hạn chế do giới hạn về thời gian thực hiện. Cụ thể, hệ thống chưa tích hợp đầy đủ một số chức năng nâng cao như đánh giá/nhận xét, gợi ý thông minh, hay tích hợp thanh toán trực tuyến. Đây là những điểm cần được cải thiện trong tương lai để nâng cao trải nghiệm người dùng và giá trị thực tiễn của sản phẩm.

5.2 Hướng phát triển

Do thời gian làm đồ án tốt nghiệp có hạn nên ứng dụng vẫn còn thiếu sót một số chức năng cho hệ thống của em. Để cải thiện và mở rộng hệ thống, một số hướng phát triển tiềm năng em muốn thêm vào bao gồm:

- Phát triển ứng dụng trên nền tảng di động, máy tính bảng
- Bổ sung tính năng đánh giá bài đăng dựa trên số điểm
- Bổ sung tính năng thanh toán trực tuyến qua các cổng thanh toán
- Hệ thống gợi ý thông minh dựa trên lịch sử tìm kiếm và các tiêu chí ưu tiên của người dùng.
- Tăng cường hệ thống bảo mật và hỗ trợ đa ngôn ngữ

Bên cạnh các bổ sung trên, em nhận thấy hệ thống của em mới chỉ hoàn thiện ở mức cơ bản, do vậy trong tương lai em dự kiến sẽ hoàn thiện những tính năng cũ và update các tính năng, công nghệ mới cho cả Front-end, Back-end cho hệ thống của mình.

TÀI LIỆU THAM KHẢO

- [1] J. A. Jason, *Learning React: Functional Web Development with React and Redux*. O'Reilly Media, 2020.
- [2] R. Khan, *Full-Stack React, TypeScript, and Node: Build Modern Web Applications with React, Node.js, and TypeScript*. Packt Publishing, 2022.
- [3] MongoDB, Inc., *MongoDB Documentation*. [Online]. Available: <https://www.mongodb.com/docs/> (visited on 2021).
- [4] B. Wheeler, *Express.js: Web Application Framework for Node.js*. Packt Publishing, 2021.
- [5] MDN Web Docs, *JavaScript (Node.js) Documentation*. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (visited on 2023).
- [6] GeeksforGeeks, *Introduction to ReactJS*. [Online]. Available: <https://www.geeksforgeeks.org/introduction-to-reactjs/> (visited on 2021).