

TRƯỜNG ĐẠI HỌC ĐÀ LẠT
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN CHUYÊN NGÀNH
ĐỀ TÀI: HỆ THỐNG NHÀ TRỢ ĐÀ LẠT

Giảng viên hướng dẫn: Thầy Lê Gia Công
Sinh viên thực hiện: 2113005 – Lê Hà Hiếu Nghĩa
2115277 – Nguyễn Trung Tín
2116977 – Trần Hữu Bằng

Đà Lạt, tháng 6/2024

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Đà Lạt, ngày... tháng... năm 2024

Giảng viên hướng dẫn

[Ký và ghi rõ họ tên]

MỤC LỤC

MỤC LỤC.....	2
LỜI CẢM ƠN.....	4
DANH MỤC HÌNH.....	8
DANH MỤC BẢNG.....	8
ĐẶT VẤN ĐỀ.....	9
CHƯƠNG 1: GIỚI THIỆU VỀ ĐỀ TÀI.....	10
1.1. Tổng quan	10
1.2. Đối tượng và phương pháp nghiên cứu	10
1.3. Cấu trúc của đồ án	10
CHƯƠNG 2: LÝ THUYẾT LIÊN QUAN.....	11
2.1. Nghiên cứu liên quan.....	11
2.1.1. Vấn đề vĩ mô và thực trạng nói chung	11
2.1.2. Các khu trọ ở Đà Lạt.....	13
2.1.3. Giải pháp marketing cho vấn đề cho thuê trọ	14
2.2. Công cụ và công nghệ sử dụng	15
2.2.1. ReactJS.....	15
2.2.2. Visual Studio Code.....	16
2.2.3. MongoDB.....	17
2.2.4. Một số thư viện hỗ trợ.....	18
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	19
3.1. Khảo sát thị trường, tập hợp thông tin.....	19
3.2. Phác thảo ban đầu	19
3.3. Thiết kế Wireframe	20
3.4. Biểu đồ Use case	21
3.4.1. Chi tiết về các use case.....	22

3.4.2.	Tìm kiếm phòng trọ theo các yếu tố.....	25
3.4.3.	Quản lý người thuê.....	25
3.5.	Biểu đồ lớp	26
3.6.	Cơ sở dữ liệu, bảng contract operation.....	31
3.7.	Biểu đồ hoạt động.....	31
3.8.	Biểu đồ tuần tự.....	32
3.9.	Yêu cầu và ràng buộc phi chức năng.....	34
CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC		34
4.1.	Kết quả đạt được.....	34
4.1.1.	Giao diện chính.....	35
4.1.2.	Hiển thị thông tin chi tiết.....	38
4.2.	Mã nguồn	39
4.2.1.	googlemap.js	39
4.2.2.	menu.js	40
4.2.3.	feedback.js	42
4.2.4.	Home.router.js	42
4.2.5.	Header.js.....	42
4.2.6.	HomePage.controller.ts	44
4.2.7.	Phongtro.router.js.....	47
4.2.8.	tb_city_phongtro.ts	47
4.3.	Kết quả kiểm thử.....	48
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....		51
TÀI LIỆU THAM KHẢO		52

LỜI CẢM ƠN

Lời đầu tiên chúng em xin chân thành cảm ơn giảng viên hướng dẫn, thầy Lê Gia Công – người hướng dẫn chúng em tận tình trong thời gian học tập và làm bài. Những sự chỉ dẫn của thầy là hướng đi chính xác để đề tài của nhóm em triển khai và hoàn thiện đáp ứng những tiêu chí mà đề tài đã đưa ra.

Chúng em cũng xin chân thành cảm ơn đến quý thầy/cô Khoa Công nghệ Thông tin – Trường Đại Học Đà Lạt đã dành rất nhiều tâm huyết tận tình giảng dạy, truyền đạt những kiến thức cần thiết và tạo điều kiện tốt nhất cho em thực hiện đề tài này. Trong quá trình thực hiện đồ án, chúng em đã có cơ hội được tiếp cận với những công nghệ mới, đồng thời cũng giúp cho chúng em có thêm kinh nghiệm, nền tảng quý báu cho bản thân để tiếp tục phát triển trên con đường học tập của mình.

Dưới đây là kết quả của quá trình tìm hiểu và nghiên cứu mà nhóm em đã đạt được trong thời gian vừa qua. Trong quá trình thực hiện, do khả năng còn hạn chế và chưa có được nhiều kinh nghiệm thực tế nên chúng em không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự thông cảm và những góp ý chỉ bảo tận tình của quý thầy, cô.

Trong khuôn khổ của đồ án, do thời gian và kinh nghiệm thực tế còn hạn chế nên có những phần thực hiện chưa được tốt, chúng em rất mong nhận được sự góp ý của thầy và các bạn.

Chúng em xin chân thành cảm ơn !

Trường Đại học Đà Lạt
Khoa Công nghệ Thông tin



ĐỀ CƯƠNG THỰC HIỆN ĐỒ ÁN

Tên đề tài: Hệ thống nhà trọ Đà Lạt

Sinh viên thực hiện:

STT	Họ và tên	MSSV	Lớp	Email
1	Lê Hà Hiếu Nghĩa	2113005	CTK45B	2113005@dlu.edu.vn
2	Nguyễn Trung Tín	2115277	CTK45B	2115277@dlu.edu.vn
3	Trần Hữu Bằng	2116977	CTK45B	2116977@dlu.edu.vn

Giảng viên hướng dẫn: Thầy Lê Gia Công

I. Mục tiêu đề tài

Xây dựng lên một trang web tìm nhà trọ nhằm đáp ứng cho các sinh viên cũng như những người lao động từ những tỉnh thành khác trên cả nước khi đến đà lạt nhập học và làm việc. Trang web giúp người dùng tiết kiệm được thời gian cũng như kinh phí sau khi sử dụng ứng dụng web này.

II. Nội dung thực hiện đề tài:

- Khảo sát các nhà trọ trên địa bàn Đà Lạt
- Khảo sát các hệ thống trực tuyến hỗ trợ tìm và thuê nhà trọ
- Khảo sát các hệ thống quản lý phòng trọ trực tuyến
- Xây dựng hệ thống có các chức năng sau:
 - Cho phép cung cấp thông tin địa chỉ của nhà trọ khi người dùng tìm kiếm
 - Cho phép người dùng có thể trao đổi nắm bắt thông tin của chủ trọ
 - Cho phép người dùng có thể tìm kiếm nhà trọ với giá hợp lý
 - Tích hợp các giải pháp thông minh vào hệ thống
 - Lược đồ use case, lược đồ hoạt động(activity)
 - Thiết kế cơ sở dữ liệu.

- Thiết kế giao diện website
- Thực hiện xây dựng các chức năng
- Triển khai và bảo trì
- Tổng hợp góp ý và phản hồi người dùng
- Sửa chữa và cập nhật dựa trên góp ý, phản hồi người dùng

III. Phần mềm và công cụ sử dụng

- Công cụ
 - Github
 - Figma
 - Visual Studio Code
- Công nghệ
 - ReactJS
 - JavaScript
 - MongoDB

IV. Dự kiến kết quả đạt được:

- Kết quả mong muốn đạt được của nhóm:
 - Áp dụng kiến thức để xây dựng một hệ thống quản lý nhà trọ mở ở Đà Lạt (không gian kết nối và website)
 - Áp dụng kiến thức để xây dựng một hệ thống nhà trọ mở ở Đà Lạt (không gian kết nối và website)
 - Cung cấp thông tin địa chỉ của nhà trọ khi người dùng tìm kiếm
 - Cho phép người dùng có thể trao đổi nắm bắt thông tin của chủ trọ
 - Tích hợp các giải pháp thông minh vào hệ thống
 - Cho phép người dùng có thể tìm kiếm nhà trọ với giá hợp lý
 - Tích lũy kinh nghiệm quản lý dự án và giải quyết vấn đề.
 - Kỹ năng làm việc nhóm, giao tiếp
 - Tổng hợp góp ý và phản hồi người dùng

V. Tài liệu tham khảo chính

1. “PhongTro123.” *PhongTro123*, <https://phongtro123.com>
2. (“Cho Thuê Phòng Trọ Giá Rẻ, Đầy Đủ Tiện Nghi T3/2024”)

Đà Lạt, ngày 02 tháng 01 năm 2024

Giảng viên hướng dẫn

(Ký tên)

Sinh viên thực hiện

(Ký tên)

BCN Khoa

(Ký tên)

Tổ trưởng bộ môn

(Ký tên)

DANH MỤC HÌNH

Hình 1: Tháp nhu cầu của Maslow.....	12
Hình 2: Bản đồ địa chính phường 8.....	13
Hình 3: Hình ảnh thực tế một số nhà cho thuê ở Đà Lạt	14
Hình 4: Icon của ReactJS	15
Hình 5: Logo của Visual Studio Code	16
Hình 6: Logo của MongoDB	17
Hình 7: BootStrap và AJAX.....	18
Hình 8: Cách thức hoạt động của backend MVC và RestAPI.....	18
Hình 9: Mô hình thác nước.....	20
Hình 10: Sơ đồ Use-case cho hệ thống	22
Hình 11: Mô tả thao tác tìm thông tin với đối tượng cụ thể là sinh viên	25
Hình 12: Use case cho việc cập nhật thông tin người cư trú.....	25
Hình 13: Biểu đồ lớp cho HomeController.....	26
Hình 14: Biểu đồ lớp cho userController	26
Hình 15: Biểu đồ cho lớp phongtro.controller	27
Hình 16: Bảng cơ sở dữ liệu cho web.....	31
Hình 17: Sơ đồ hoạt động chung cho toàn hệ thống	32
Hình 18: Sơ đồ hoạt động cho thao tác đăng ký.....	32
Hình 19: Biểu đồ cho chức năng đăng nhập	33
Hình 20: Biểu đồ tuần tự cho chức năng tìm kiếm thông tin	33
Hình 21: Biểu đồ tuần tự cho chức năng quản lý thông tin (Cập nhật)	34
Hình 22: Giao diện chính của trang web.....	35
Hình 23: Thanh tìm kiếm theo ba yếu tố chính.....	36
Hình 24: Các bài đăng nổi bật.....	36
Hình 25: Thanh Footer của web.....	37
Hình 26: Thông tin chi tiết về phòng của người đăng (cho thuê)	38

DANH MỤC BẢNG

Bảng 1: Mô tả cho UC đăng nhập.....	23
Bảng 2: Mô tả cho UC đăng ký.....	23
Bảng 3: Mô tả cho UC Đăng thông tin.....	24

Bảng 4: Mô tả cho UC tìm kiếm thông tin	24
Bảng 5: Mô tả cho UC quản lý (cập nhật).....	24
Bảng 6: Contract Operation của web.....	31
Bảng 7: Các yêu cầu chức năng của hệ thống.....	34
Bảng 8: Bảng các test scenario trong bước Manual Testing.....	49
Bảng 9: Điều kiện tiên quyết với thao tác kiểm thử giao diện	49
Bảng 10: Điều kiện tiên quyết với thao tác đăng nhập.....	49
Bảng 11: Điều kiện tiên quyết đối với thao tác đăng ký.....	50
Bảng 12: Điều kiện tiên quyết với thao tác lấy lại mật khẩu.....	51

ĐẶT VẤN ĐỀ

Hiện nay, Đà Lạt có một số lượng lớn sinh viên ngoại tỉnh và sinh viên từ các huyện trên địa bàn tỉnh Lâm Đồng, đa số các sinh viên này phải ở trọ ngoài vì ký túc xá chỉ đáp ứng được 20-30% nhu cầu của sinh viên. Tìm kiếm phòng trọ như ý theo cách thức truyền thống là một việc rất tốn công sức và thời gian. Với những sinh viên năm nhất thì đây là vấn đề khó khăn.

Khi không tìm được phòng trọ, sinh viên buộc phải tìm đến các trung tâm môi giới. Thế nhưng chi phí có thể là cao cho lựa chọn này và có thể phát sinh những vấn đề liên quan khác.

Xây dựng một website với mục đích giúp cho việc tìm nhà trọ đối với sinh viên - đặc biệt hơn đối với tân sinh viên trở nên dễ dàng thuận tiện. Các nguồn phòng được đăng trên web sẽ được đảm bảo về độ tin cậy với đầy đủ các thông tin cần thiết cho việc đi tìm một phòng trọ như: giá phòng, diện tích, cấu tạo sàn nhà, trần, ảnh chụp toàn cảnh khu trọ sẽ ở, khoảng cách đến chợ, trường,.. cùng với ý kiến và yêu cầu đến từ chủ nhà.

Từ những lý do trên và có sự góp ý của các thành viên trong lớp CTK45B, nhóm đã quyết định chọn đề tài: “**Hệ thống phòng trọ Đà Lạt**” . Trong đề tài này, nhóm sẽ trình bày cái nhìn cơ bản thực trạng thuê phòng trọ hiện nay trên địa bàn Đà Lạt và xây dựng trang web tương ứng với yêu cầu đề tài.

CHƯƠNG 1: GIỚI THIỆU VỀ ĐỀ TÀI

1.1. Tổng quan

Bối cảnh thực tế của đề tài: với sinh viên, việc tìm kiếm phòng trọ bằng các phương pháp truyền thống như rao vặt, truyền miệng thiếu chính xác và có thể tốn nhiều thời gian và tiền bạc. Nhóm thực hiện đề tài mong muốn khắc phục điều đó bằng cách áp dụng công nghệ thông tin vào công việc tìm kiếm phòng trọ.

Vấn đề cần giải quyết: Sử dụng các nền tảng công nghệ để hỗ trợ việc tìm kiếm phòng trọ để khắc phục các nhược điểm của việc tìm trọ truyền thống: khả năng tiếp cận các thông tin phòng trọ còn hạn chế, thông tin thiếu chính xác, việc xác thực thông tin (xem trọ, liên hệ với chủ trọ) còn tốn nhiều thời gian và một số rủi ro (lừa đảo – gây mất tiền bạc). Cung cấp chức năng đăng tải thông tin phòng trọ cho các chủ phòng trọ.

Đề xuất nội dung thực hiện: nghiên cứu và xây dựng một website có chức năng đăng tải, hiển thị danh sách, thông tin phòng trọ chính xác, được cập nhật liên tục.

Đây là trang web thuộc lĩnh vực thương mại điện tử và xuất phát từ mong muốn áp dụng những kiến thức đã học để áp dụng vào thực tế, muốn thoát ra vùng an toàn, thử sức trên thị trường.

1.2. Đối tượng và phương pháp nghiên cứu

Đề tài tập trung nghiên cứu vào các đối tượng là sinh viên, người lao động đến từ ngoài tỉnh có nhu cầu tìm trọ phù hợp với điều kiện.

Phương pháp nghiên cứu

- Thu thập và nghiên cứu các tài liệu có liên quan đến đề tài. Tổng hợp và chọn lọc các nghiên cứu để đưa vào đồ án và trích dẫn nguồn tham khảo.
- Phân tích các chức năng và yêu cầu cần có của một trang web.
- Tiến hành xây dựng trang web dựa trên các yêu cầu
- Bảo trì, kiểm thử hệ thống

1.3. Cấu trúc của đồ án

Đồ án có cấu trúc bao gồm các chương:

- Chương 1: Giới thiệu về đề tài

- Chương 2: Lý thuyết liên quan
- Chương 3: Phân tích thiết kế hệ thống
- Chương 4: Kết quả đạt được
- Chương 5: Kết luận và hướng phát triển
- Phần cuối: Danh mục tài liệu tham khảo

CHƯƠNG 2: LÝ THUYẾT LIÊN QUAN

2.1. Nghiên cứu liên quan

2.1.1. Vấn đề vĩ mô và thực trạng nói chung

Ngay từ khi nhận được thông tin trúng tuyển đại học, các bạn sinh viên đã phải lo tới chuyện tìm nhà trọ. Tại hầu hết các tỉnh / thành phố, nhà trọ đều tập trung quanh các trường Đại học, Cao đẳng và tỷ lệ sinh viên sống tại đây cao hơn những khu vực khác trên địa bàn.

Thực trạng tại một số thành phố lớn: Giá thuê phòng từ 300000 - 500000/ ngày với diện tích 22m². Hay có nơi là 1 triệu, cách các trường như Bách khoa, Kinh tế Quốc Dân,... từ 300 – 500m [1]. Ở TP.HCM, giá trọ trong tháng 8/2023 đã tăng 10-40% (từ 500.000 – 2.000.000/tháng [2] [3]). Xác định nơi cư trú trong một khu vực mới được biết đến bị ảnh hưởng bởi nhiều yếu tố như vị trí, cơ sở vật chất, hệ thống hợp đồng và giá cả. Dịch vụ xung quanh các khu trọ đều khá đa dạng, đáp ứng nhu cầu của sinh viên từ lương thực, quán ăn, chợ, cho tới các nhu cầu về thông tin.

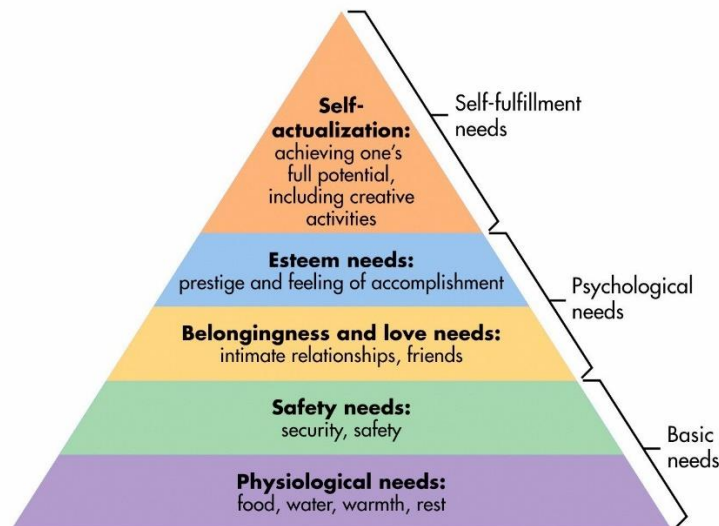
Kinh doanh phòng trọ cũng không phải là con đường dễ dàng mà một số cá nhân đang lầm tưởng. Vay lãi cao lên đến 15%/năm chẳng khác nào tạo áp lực tài chính đè lên vai nhà đầu tư, chủ trọ. Tổng doanh thu của các chủ kinh doanh có tới 10% thuế nhà nước.

Thêm nữa, thị trường thuê nhà cũng đang chịu ảnh hưởng từ các yếu tố vĩ mô khác như biến đổi khí hậu và đại dịch COVID-19. Biến đổi khí hậu có thể gây ra các thiên tai tự nhiên, làm suy giảm nguồn cung nhà ở và đồng thời làm tăng giá thuê. Trong khi đó, đại dịch COVID-19 đã tạo ra những biến động mạnh mẽ trong thị trường bất động sản, làm cho một số người không còn khả năng trả tiền thuê nhà, dẫn đến việc tăng mức nợ cho chủ sở hữu nhà và các tổ chức tài chính.

Thách thức và rủi ro: gần một nửa phòng trọ trên địa bàn TP.HCM có diện tích

nhỏ hơn 10m²; nhiều phòng trọ không có cửa sổ, không được lát gạch, nhà vệ sinh riêng. Thậm chí, không ít nhà trọ dưới 5 tầng không có hồ sơ thiết kế lẫn hệ thống, thiết bị phòng cháy chữa cháy [4].

Chính những rủi ro trên đánh vào tâm lý khách hàng - người thuê trọ, họ cần đảm bảo rằng những quan điểm thẩm mang tính phải trùng khớp với ý nghĩa của mô hình tháp nhu cầu của Maslow.



Hình 1: Tháp nhu cầu của Maslow

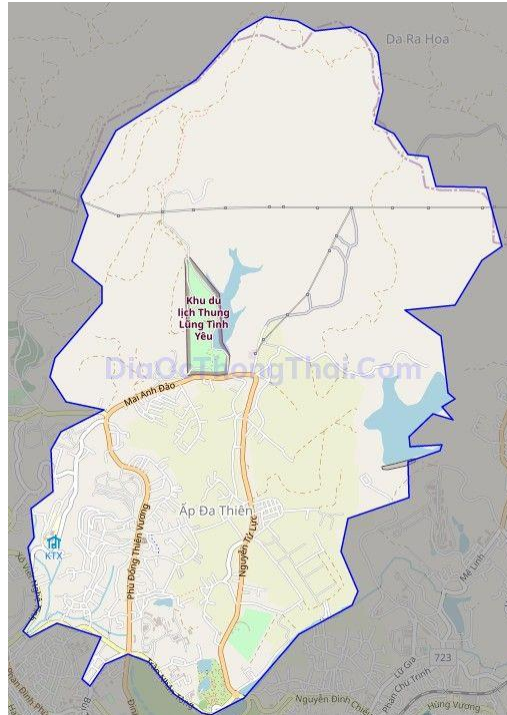
Theo tháp nhu cầu của Abraham Maslow (nhà tâm lý học 62 tuổi người Mỹ), có tới năm cấp bậc minh họa những nhu cầu cơ bản của con người và là mô hình có tính lan truyền về mặt nhận thức nhất trong khoa học hành vi. Theo tác giả của mô hình này, “*nhu cầu của con người tự sắp xếp theo thứ bậc trước tiên năng*” [5] [6]. Như vậy, có thể tin tưởng rằng, nhu cầu về an ninh và sự an toàn nằm ở tầng thứ hai đối với họ. Đảm bảo được an ninh sẽ dẫn đến an toàn sinh mạng cũng được đảm bảo và là tiền đề cho các yếu tố khác. Thực tế đã chứng minh, góc ngách, hẻm nhỏ sâu bên trong, dẫn đến việc phát sinh các vấn đề an ninh (người thuê trọ lại là đối tượng có tâm lý chủ quan trong một số thời điểm như lễ, Tết). Nhiều vụ mất cắp tài sản xảy ra khiến yêu cầu về an ninh lại càng được quan tâm hơn trong việc tìm kiếm của đối tượng.

Vị trí có tác động mạnh mẽ đến quyết định lựa chọn phòng trọ của sinh viên ngoại tỉnh học tập. Không phải sinh viên nào khi lên thành phố học cũng có thể chủ động trong phương tiện đi lại cũng như thành thạo đường xá. Thêm vào đó, việc phải chịu các chi phí phòng hay điện nước hàng tháng khiến họ không muốn chịu

thêm phí đi lại thêm nữa, đó lại là gánh nặng đè lên vai sinh viên.

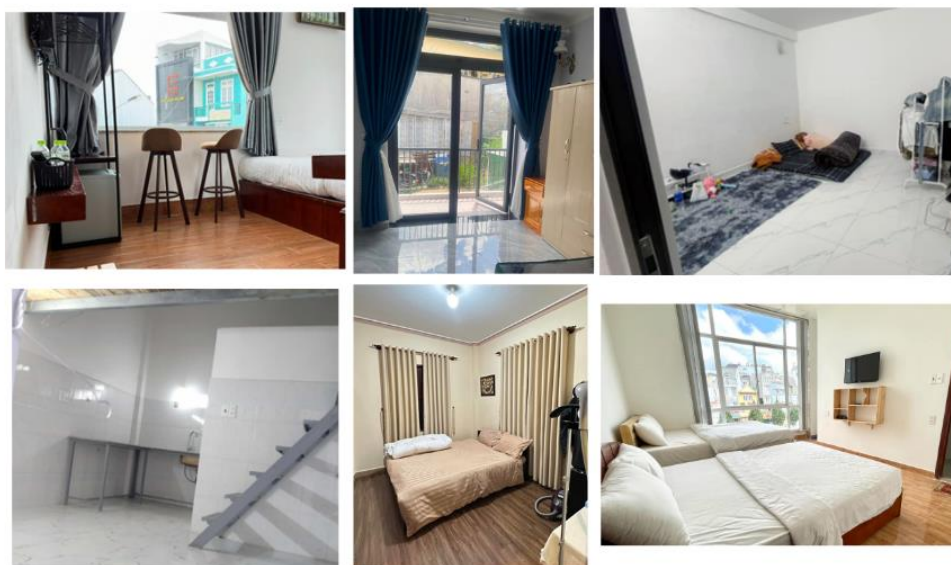
2.1.2. Các khu trọ ở Đà Lạt

Theo thống kê mới nhất, hiện trên địa bàn thành phố có tới 1.448 nhà trọ với công suất khoảng 7.000 phòng [7]. Một báo cáo do UNDP ngày 12/4/2022, tỉnh Lâm Đồng ta là điểm hấp dẫn người nhập cư đến nhiều thứ tư [8].



Hình 2: Bản đồ địa chính phường 8

Khu vực này có trên 600 khu nhà trọ, trong đó khoảng 450 khu nhà trọ dành cho sinh viên. Hầu hết các khu trọ đều tập trung ở khu vực Trần Khánh Dư, đường Phù Đổng Thiên Vương, đường Võ Trường Toản, khu vực Hasfarm trên đường Nguyễn Tử Lực. Bên cạnh đó, còn có sự đóng góp của các khu dãy trọ ở đường Bùi Thị Xuân và Nguyễn Công Trứ đã tạo thành một ma trận của những nhà trọ trên địa bàn Đà Lạt. Các khu trọ đều được xây với nhiều phong cách khác nhau.



Hình 3: Hình ảnh thực tế một số nhà cho thuê ở Đà Lạt

Một số ví dụ về các nhà trọ trên địa bàn Đà Lạt: các mức giá có thể là 1tr4, 1tr7, 2tr, 1tr6, 1tr9, 2tr8,... Thực tế, theo quan sát của nhóm, tới 90% đối tượng mà nhóm quen biết có xu hướng chọn lựa vị trí thuê trọ gần trường, các quán ăn, quầy thuốc, tiệm photocopy, chỉ có số ít là có nhu cầu khác. Nhu cầu thuê trọ cũng chỉ ở mức bình dân. Đường xá thì thuận tiện, chỗ thuê hợp lý nên tỷ lệ người ở lại trọ trên một năm gần như đạt hơn 1,5 năm. Không chỉ có sinh viên năm nhất, những đối tượng từ các khu vực khác chuyển sang,...

2.1.3. Giải pháp marketing cho vấn đề cho thuê trọ

Thuê trọ là một vấn đề thiết yếu và cần được giải quyết. Tuy nhiên, để tìm ra giải pháp phù hợp, cần phải xem xét nhiều yếu tố khác nhau, bao gồm chính sách, kinh tế, xã hội và văn hóa.

Một trong những giải pháp có thể giúp giải quyết vấn đề nhà trọ là marketing. Có nhiều diễn giải khác nhau về marketing, nhưng có thể hiểu như sau: “*Marketing bao gồm những nỗ lực ảnh hưởng đến việc chuyển giao quyền sở hữu hàng hóa và chăm sóc phân phối vật lý của chúng*” [9] và bao gồm tất cả các hoạt động liên quan đến việc tạo ra các tiện ích địa điểm, thời gian và sở hữu. Một tác giả khác đã định nghĩa khái niệm này là “*Khi hoạt động tiếp thị được phát triển đúng cách, điều đó có nghĩa là bạn tạo ra những trải nghiệm ấn tượng xung quanh thương hiệu hoặc sản phẩm của mình. Bằng cách này, những người không kết nối với thương hiệu của bạn muốn nói với người khác rằng bạn tuyệt vời như thế nào và chu kỳ đó sẽ bắt đầu lại*” [10].

Marketing có thể giúp tăng cường nhận thức của công chúng về vấn đề nhà trọ và thu hút sự quan tâm của các nhà đầu tư, các tổ chức phi lợi nhuận và các chính phủ địa phương. Ngoài ra, marketing cũng có thể giúp tăng cường sự nhận thức của người dân về các chính sách và chương trình hỗ trợ nhà ở có sẵn.

Tuy nhiên, để thành công trong việc giải quyết vấn đề nhà trọ, chúng ta cần phải xem xét nhiều giải pháp khác nhau và tìm ra giải pháp phù hợp nhất cho từng trường hợp cụ thể.

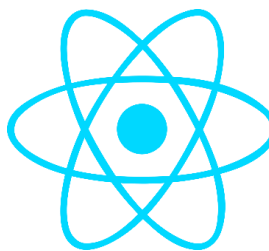
Mặt khác, công nghệ đang ngày càng trở thành một phần không thể thiếu trong việc tiếp cận và tương tác với khách hàng. Các ứng dụng di động, trang web và mạng xã hội có thể được sử dụng để quảng cáo dịch vụ, cung cấp thông tin và tương tác với khách hàng tiềm năng. Hơn nữa, việc sử dụng công nghệ trong quản lý và vận hành cũng giúp tối ưu hóa hiệu suất và tạo ra trải nghiệm tốt hơn cho cả người cho thuê và người thuê.

Việc xây dựng mối quan hệ vững chắc với cộng đồng địa phương là một phần quan trọng của chiến lược marketing thành công. Các nhà cung cấp dịch vụ thuê trọ có thể tham gia vào các hoạt động cộng đồng, hỗ trợ các dự án cộng đồng và tham gia vào các sự kiện địa phương để tạo dựng hình ảnh tích cực và tạo lòng tin từ cộng đồng.

Việc xây dựng và quảng bá thương hiệu là một phần quan trọng của chiến lược marketing. Các nhà cung cấp dịch vụ thuê trọ cần phải tạo ra một hình ảnh thương hiệu mạnh mẽ và độc đáo, từ logo đến thông điệp và giá trị cốt lõi của mình. Việc này giúp họ nổi bật giữa đám đông và tạo ra ấn tượng sâu sắc trong tâm trí của khách hàng.

2.2. Công cụ và công nghệ sử dụng

2.2.1. ReactJS



Hình 4: Icon của ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở được thiết kế bởi Facebook để tạo ra những ứng dụng web hấp dẫn, nhanh và hiệu quả, nền tảng của nhiều ứng

dụng và là framework được nhiều front-end dev theo đuổi [11]. Ngoài ra, nó tạo ra những ứng dụng web với UI tốt hơn để nâng cao trải nghiệm người dùng.

React cho phép các nhà phát triển tạo ra các thành phần giao diện người dùng độc lập, tái sử dụng và mở rộng được. Điều này giúp giảm thiểu thời gian và công sức để phát triển và bảo trì ứng dụng web, cũng như giảm thiểu khả năng xảy ra lỗi và tăng tính ổn định của ứng dụng. Ngoài ra, nó được tích hợp một cách dễ dàng, sử dụng với các công nghệ web khác như Node.js và Redux để tạo ra các ứng dụng web phức tạp và khối lượng lớn.

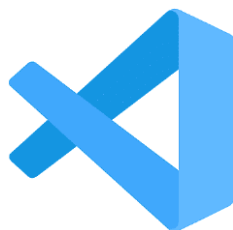
Lệnh cài đặt ReactJS trên VS Code [12]:

- Nhập lệnh sau vào cửa sổ dòng lệnh: `npx create-react-app hello-world`
- Sau đó mở trong Visual Studio Code bằng câu lệnh: `cd hello-world && code .`
- Mở terminal (`Ctrl + Shift + ``) và nhập lệnh ***npm start*** để khởi động máy chủ Node và chạy ứng dụng
- Để kiểm tra, mở trình duyệt và truy cập vào `localhost:3000`

Trong VSCode có hỗ trợ một số tool cần thiết cho ReactJS [13]:

- ES7 React/Redux/GraphQL/React-Native snippets
- VSCode React Refactor
- GitLens
- Jumpy (đi từ một dòng code đến một từ khóa cụ thể mà chỉ cách đó vài dòng hay khoảng trắng)

2.2.2. Visual Studio Code



Hình 5: Logo của Visual Studio Code

Visual Studio Code là một trình mã biên tập được phát triển bởi Microsoft. Ngoài ra nó còn hỗ trợ đa dạng các chức năng Debug, đi kèm với Git, có Syntax Highlighting. Đặc biệt là tự hoàn thành mã thông minh, Snippets, và khả năng cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép các lập

trình viên thay đổi Theme, phím tắt, và đa dạng các tùy chọn khác. Mặc dù trình soạn thảo Code này tương đối nhẹ, nhưng lại bao gồm các tính năng mạnh mẽ.

2.2.3. MongoDB



Hình 6: Logo của MongoDB

MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh. Bên cạnh đó, nó có hiệu năng cao, tính sẵn dùng cao, dễ dàng co giãn.

MongoDB schema hoạt động khác với SQL. Với kiểu thiết kế cơ sở dữ liệu theo MongoDB, chúng ta sẽ: không có quy trình chính thức, những thuật toán và cả những quy tắc [14]. Nó còn hỗ trợ các mô hình mở rộng như sharding, cho phép bạn chia nhỏ dữ liệu và phân phối chúng trên nhiều máy chủ, giúp tăng hiệu suất và khả năng chịu tải của hệ thống cũng như cung cấp một ngôn ngữ truy vấn mạnh mẽ cho phép bạn thực hiện các truy vấn phức tạp và linh hoạt trên dữ liệu.

Ngoài ra, mongoDB có thể được sử dụng trong các trường hợp sau: [15]

- Tích hợp một lượng lớn dữ liệu đa dạng
- Mô tả các cấu trúc dữ liệu phức tạp, biến hoá
- Cung cấp dữ liệu cho các ứng dụng hiệu suất cao
- Hỗ trợ các ứng dụng đám mây lai và đa đám mây
- Hỗ trợ phương pháp phát triển Agile

Một số câu lệnh cơ bản:

- Tạo cơ sở dữ liệu mới, thực hiện lệnh sau: `use DATABASE_NAME`
- Loại bỏ CSDL đã chọn: `db.dropDatabase();`
- Loại bỏ bộ sưu tập: `db.COLLECTION_NAME.drop();`

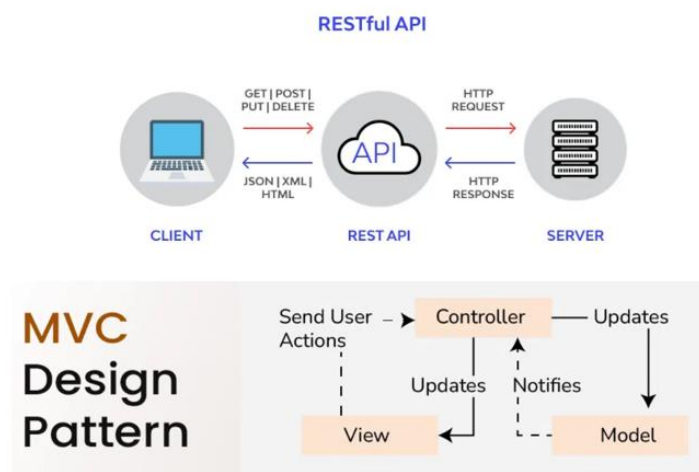
2.2.4. Một số thư viện hỗ trợ



Hình 7: Bootstrap và AJAX

Bootstrap là một bộ công cụ mạnh mẽ – một tập hợp các công cụ HTML, CSS và JavaScript để tạo và xây dựng các trang web và ứng dụng web. Đây là một dự án mã nguồn mở và miễn phí, được tạo bởi Twitter và lưu trữ trên GitHub. Đây còn là một collection hoàn toàn free. Bộ sưu tập này gồm có các mã nguồn mở, hay các công cụ giúp bạn có thể tạo thành một website đầy đủ với các thành phần [16].

AJAX (Asynchronous JavaScript and XML) là một kỹ thuật lập trình web cho phép gửi và nhận dữ liệu giữa trang web và máy chủ mà không cần tải lại trang. Kỹ thuật này sử dụng các công nghệ web như JavaScript và XML (hoặc JSON) để giao tiếp với máy chủ và cập nhật dữ liệu trên trang web một cách động. Với Ajax, người dùng có thể tương tác với trang web một cách mượt mà và nhanh chóng hơn, mà không bị gián đoạn bởi các trang web phải tải lại. Ajax được sử dụng rộng rãi trong các ứng dụng web động như Gmail, Facebook, Twitter và Google Maps [17].



Hình 8: Cách thức hoạt động của backend MVC và RestAPI

Các design pattern sẽ được áp dụng: RESTful API và Backend MVC (mẫu thiết kế tách biệt lớp View khỏi Model và Controller), Express.js (xử lý các loại yêu cầu HTTP như GET, POST, PUT, DELETE)

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1. Khảo sát thị trường, tập hợp thông tin

Từ những khảo sát thực tế bên ngoài, nhóm đã xác định được các thông tin sau:

Đặc điểm nhân khẩu học:

- Sinh viên
- Công nhân, người lao động
- Người mới chuyển đến thành phố
- Người cần thuê nhà ở tạm thời (đi du lịch hoặc công tác ngắn hạn)

Nhu cầu và mong muốn:

- Nhận được thông tin cần thiết liên quan: giá thuê hằng tháng, khoảng cách, vị trí
- Mục đích thuê trọ: làm thuê xa nhà

Từ việc xác định đối tượng người dùng, ta có thể minh họa bức tranh về chân dung người dùng (user persona) như sau:

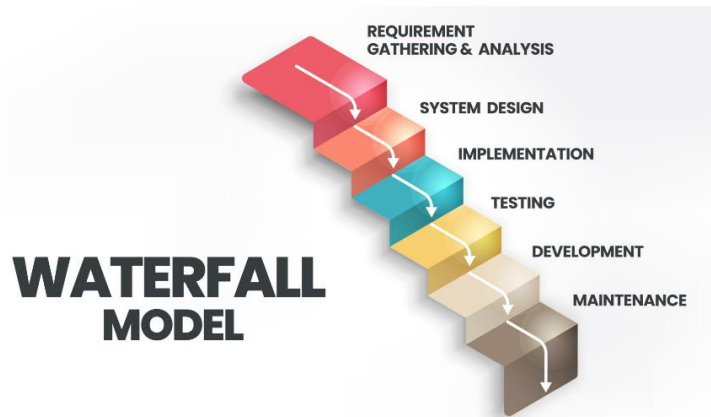
- Sinh viên: là nhóm người dùng có nhu cầu chiếm đến 80% trong toàn bộ nhân khẩu. Đây là đối tượng có thu nhập không cao, cần tiếp cận với một khu trọ có giá cả phù hợp với ngân sách cá nhân.
- Công nhân, người lao động: là nhóm đối tượng có thể là cư dân trên địa bàn hoặc là người từ nơi khác chuyển đến. Độ tuổi nằm trong khoảng từ 18-50. Họ có thể đến từ những nơi khác.

3.2. Phác thảo ban đầu

Tính năng cần có trên trang web: Cho phép người dùng tìm kiếm thông tin trên trang web. Thông tin liên quan bao gồm: khoảng cách, giá, vị trí. Scenarios: Người dùng nhập từ khóa tìm kiếm và xem kết quả phù hợp. Stories: Người dùng muốn tìm thông tin về các khu trọ nằm trên một cung đường cụ thể, nhập thông tin tìm kiếm, từ đó nhận lại kết quả liên quan.

Sử dụng hình ảnh phong phú, phản ánh đúng đặc trưng, liên quan đến phòng trọ, căn hộ. Cần đảm bảo hình ảnh được chọn phải tương thích với màu sắc chủ đạo để tạo nên một giao diện đồng nhất, dễ nhìn.

Mô hình được sử dụng để phát triển web là mô hình thác nước do có sự chú trọng vào sự logic hóa của các bước thực hiện trong suốt vòng đời phát triển của một phần mềm [18].



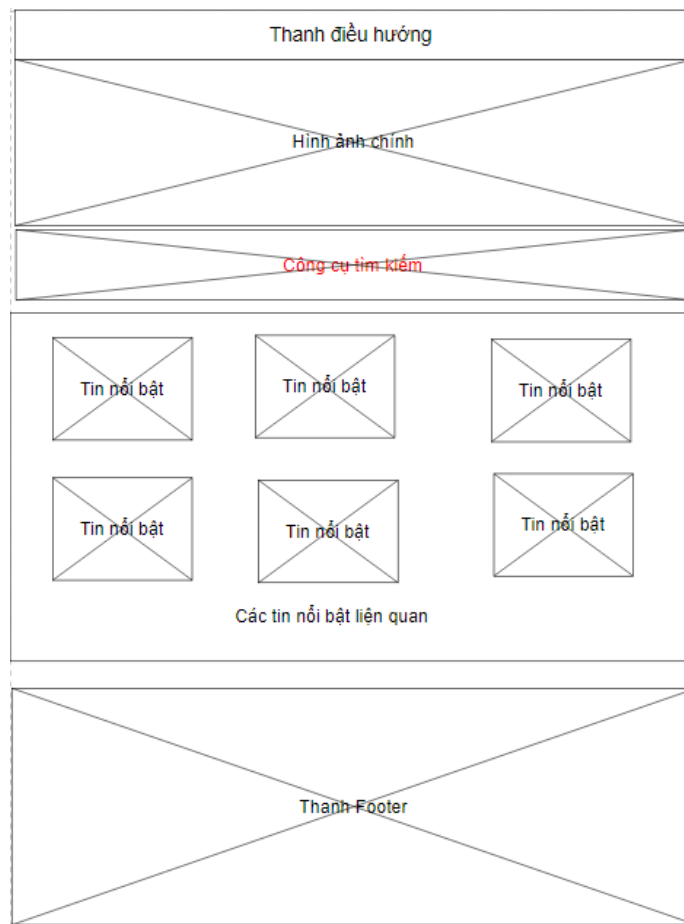
Hình 9: Mô hình thác nước

3.3. Thiết kế Wireframe

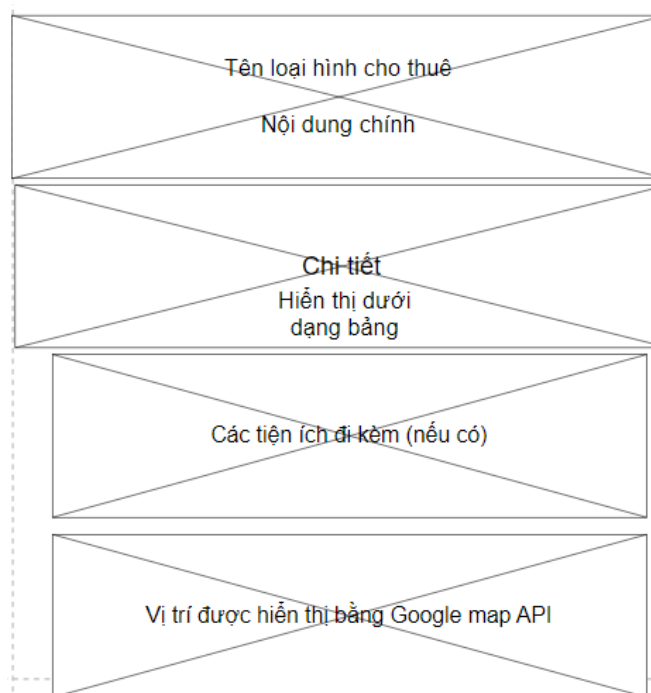
Wireframe là các bố cục đen trắng được phác thảo dựa trên vị trí và có kích thước cụ thể. Chúng tương ứng với tính năng trang, từng thành phần, các khu vực chuyển đổi hay mỗi sự điều hướng cho website của bạn. Chúng không có màu sắc, logo, sự lựa chọn về phông chữ hay bất kỳ yếu tố design nào khác.

Điều Wireframe hướng tới chính là cấu trúc của website.

Wireframe chung cho trang chính

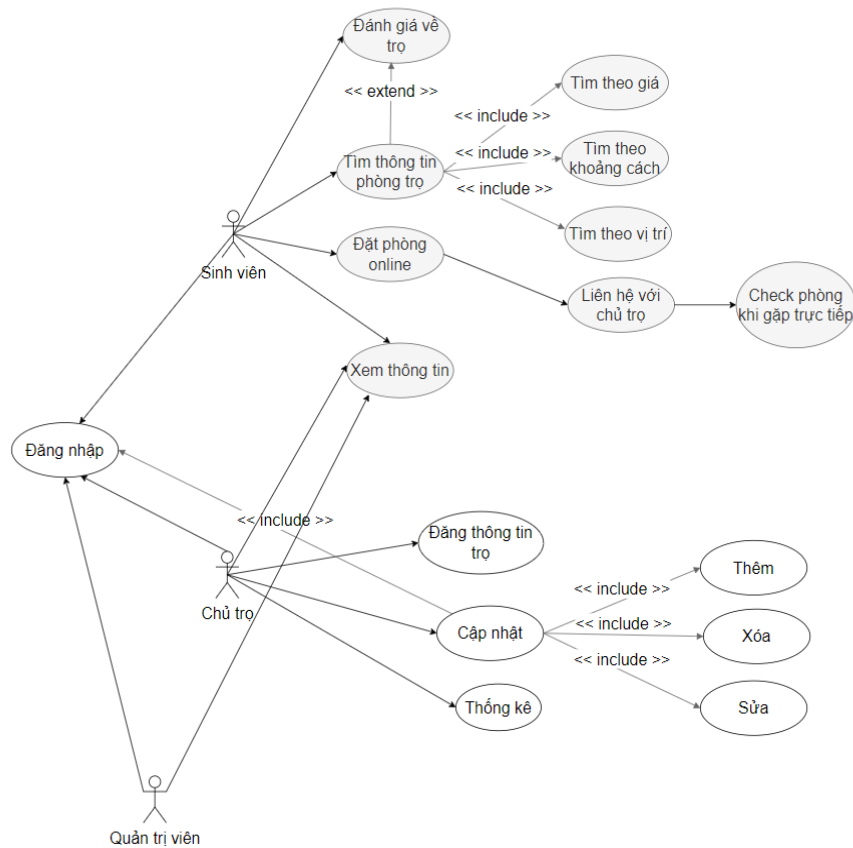


Wireframe cho màn hình thông tin chi tiết



3.4. Biểu đồ Use case

Sơ đồ use case tổng quát và chi tiết về các yếu tố



Hình 10: Sơ đồ Use-case cho hệ thống

3.4.1. Chi tiết về các use case

Hệ thống giới thiệu chỗ ở cung cấp thông tin từ cả hai phía

- Các chủ cho thuê có nhu cầu giới thiệu nhà trọ/ căn hộ / nhà nguyên căn
- Sinh viên có nhu cầu thuê

Use Case UC1: Đăng nhập	
Phạm vi truy cập	Tất cả các actor
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Hệ thống hiển thị màn hình đăng nhập 2. Người dùng nhập username, password 3. Nếu thành công, hệ thống thể hiện việc đăng nhập đã hoàn tất 4. Kết thúc use case
Luồng sự kiện phụ	<p>1a Mật khẩu không hợp lệ (khi nhập sai username hoặc password)</p> <ul style="list-style-type: none"> - Bắt nhập lại thông tin, đi kèm đó là thông báo nhập sai và yêu cầu nhập lại - Quay lại bước 2 trong luồng sự kiện chính <p>1b. Quên mật khẩu: khi người dùng chọn chức năng</p>

	quên mật khẩu và có thể lấy lại mật khẩu(thực hiện ở 1c) - Hiện thị màn hình 1c. Lấy lại mật khẩu - Nhấn vào nút quên mật khẩu tại form đăng nhập - Chỉ cần 1 trong ba loại mật khẩu: cũ, hiện tại và xác nhận sai => bắt người dùng nhập lại từ đầu
--	---

Bảng 1: Mô tả cho UC đăng nhập

Đăng ký

Use Case UC2: Đăng ký	
Phạm vi truy cập	Tất cả các actor
Luồng sự kiện chính	1a: Truy cập trang web 1. Thực hiện việc đăng nhập 2. Đưa ra form chứa các textbox liên quan, người chủ chỉ cần nhập thông tin liên quan vào - Nếu chưa nhập đủ thông tin => Yêu cầu nhập hết thông tin 3. Nhấn xác nhận 4. Gửi thông báo thành công, thông tin mới đó được hiển thị lên form 5. Kết thúc thao tác (có thể load lại trang)
Luồng sự kiện phụ	- Nếu chưa đăng nhập thì phải thực hiện lại bước 1 trong luồng sự kiện chính

Bảng 2: Mô tả cho UC đăng ký

Đăng thông tin

Use Case UC3: Đăng thông tin	
Phạm vi truy cập	ChuTro
Luồng sự kiện chính	1a: Tạo mới 1. Thực hiện việc đăng nhập 2. Đưa ra form chứa các textbox liên quan, người chủ chỉ cần nhập thông tin liên quan vào 3. Nhấn xác nhận 4. Gửi thông báo thành công, thông tin mới đó được hiển thị lên form 5. Kết thúc thao tác (có thể load lại trang)
Luồng sự kiện phụ	- Nếu chưa đăng nhập thì phải thực hiện lại bước 1 trong luồng sự kiện chính

Bảng 3: Mô tả cho UC Đăng thông tin

Tìm kiếm thông tin

Use Case UC4: Tìm kiếm thông tin	
Phạm vi truy cập	SinhVien
Luồng sự kiện chính	1a: <ol style="list-style-type: none"> 1. Truy cập trang web 2. Có thể để lại comment, để owner của bài đăng đó có thể nhìn thấy.
Luồng sự kiện phụ	- Có thể không cần đăng nhập

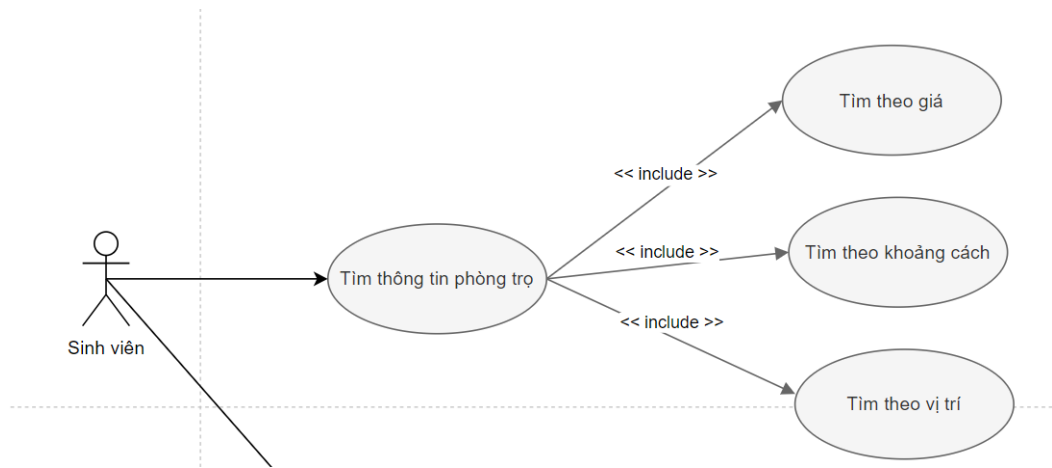
Bảng 4: Mô tả cho UC tìm kiếm thông tin

Cập nhật (thêm, xóa, sửa)

Use Case UC5: Cập nhật(Thêm xóa, sửa)	
Phạm vi truy cập	ChuTro
Điều kiện tiên quyết	- Phải hoàn thành việc đăng nhập vào web
Luồng sự kiện chính	1a: Chức năng thêm <ol style="list-style-type: none"> 1. Hệ thống hiển thị màn hình quản lý (điền thông tin của người mới đến thuê) 2. ChuTro nhập thông tin liên quan 3. Nếu thành công, hiển thị thêm thông tin thành công, cập nhật trong db 4. Kết thúc thao tác 1b: Chức năng chỉnh sửa hoặc xóa <ol style="list-style-type: none"> 1. Hệ thống hiển thị màn hình quản lý 2. Nhấn vào nút edit hoặc delete ở góc bên phải 3. Hiển thị thông tin để user có thể thực hiện thao tác 4. Hệ thống gửi form xác nhận có muốn thay đổi hay không? 5. Nếu thành công, hiển thị thông tin đã chỉnh sửa hoặc xóa thành công 6. Kết thúc thao tác

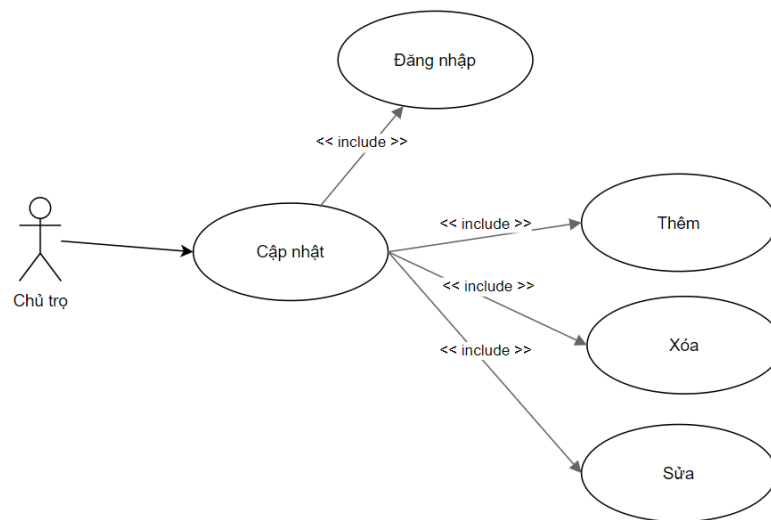
Bảng 5: Mô tả cho UC quản lý (cập nhật)

3.4.2. Tìm kiếm phòng trọ theo các yếu tố



Hình 11: Mô tả thao tác tìm thông tin với đối tượng cụ thể là sinh viên

3.4.3. Quản lý người thuê

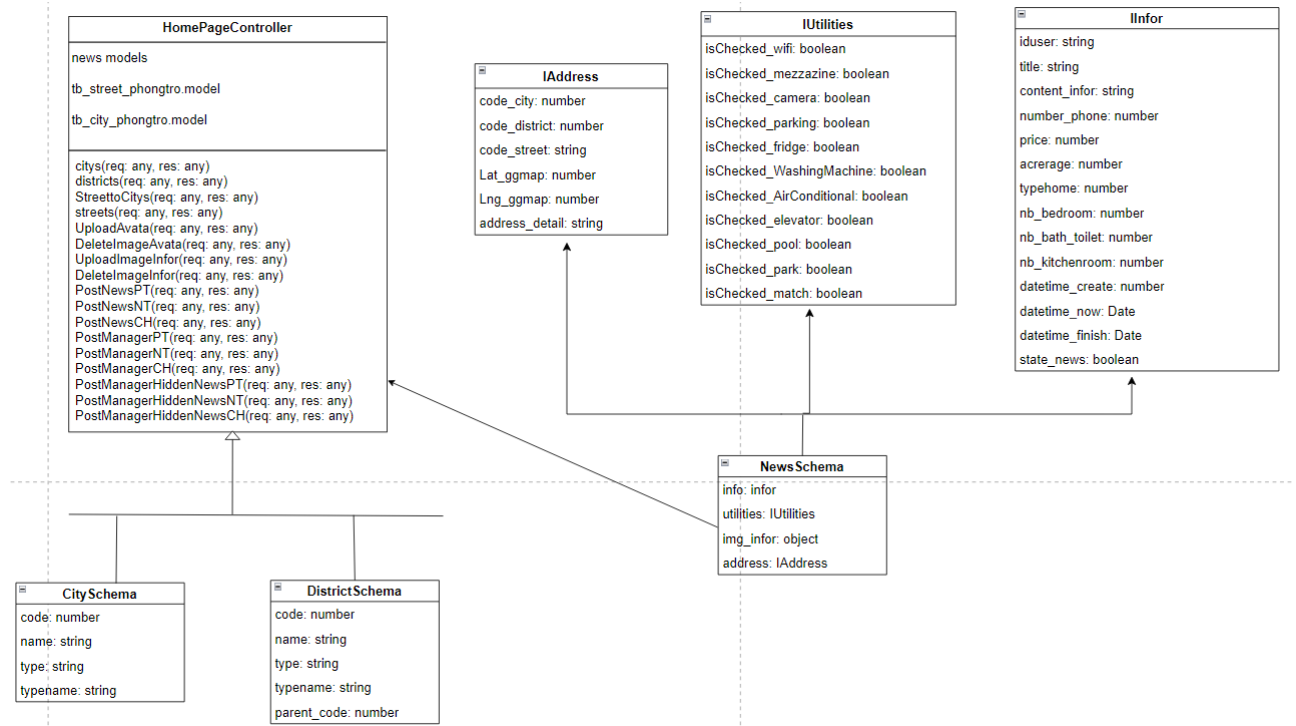


Hình 12: Use case cho việc cập nhật thông tin người cư trú

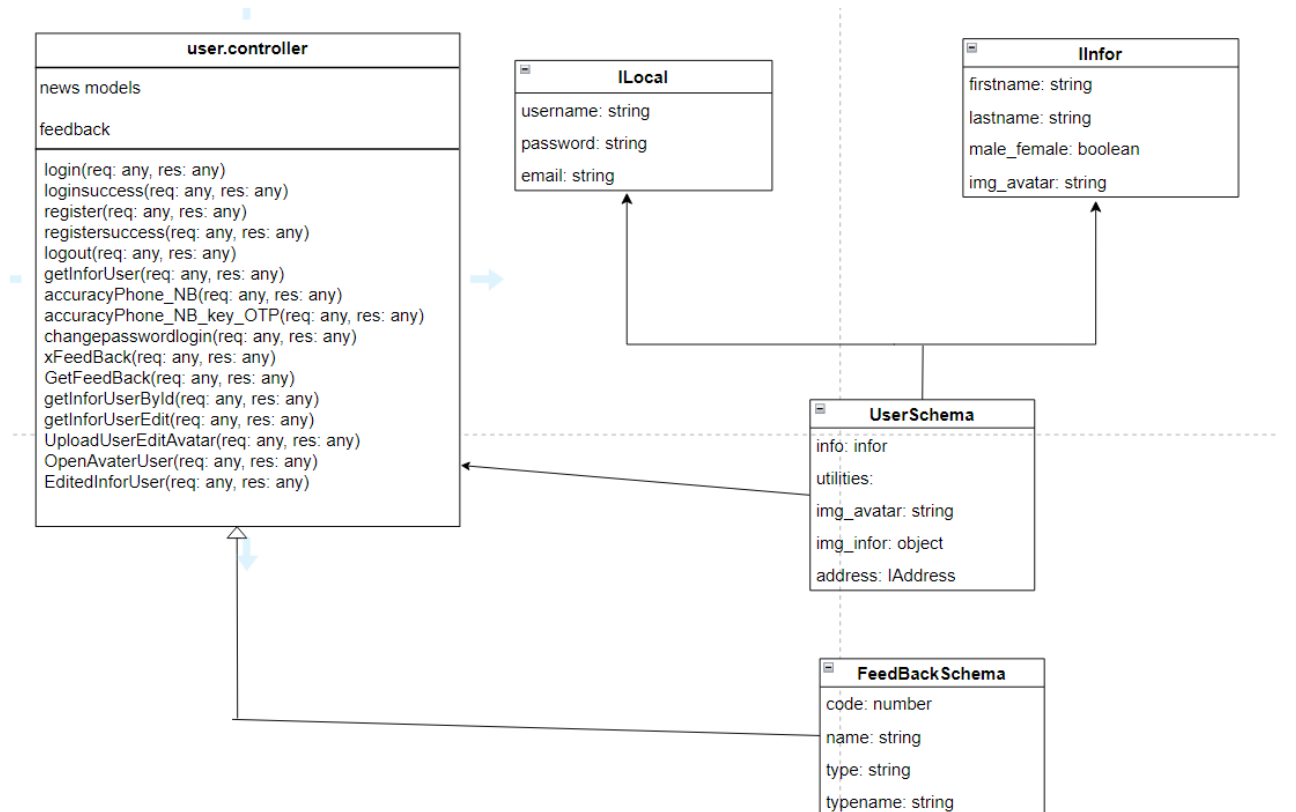
Mô tả: tác vụ này dành cho các chủ trọ. Có thể thêm, sửa, xóa danh sách những người đến tạm trú.

Lưu ý: Actor phải là chủ trọ, đồng thời thực hiện đăng nhập thì mới có thể truy cập và thực hiện được các task được.

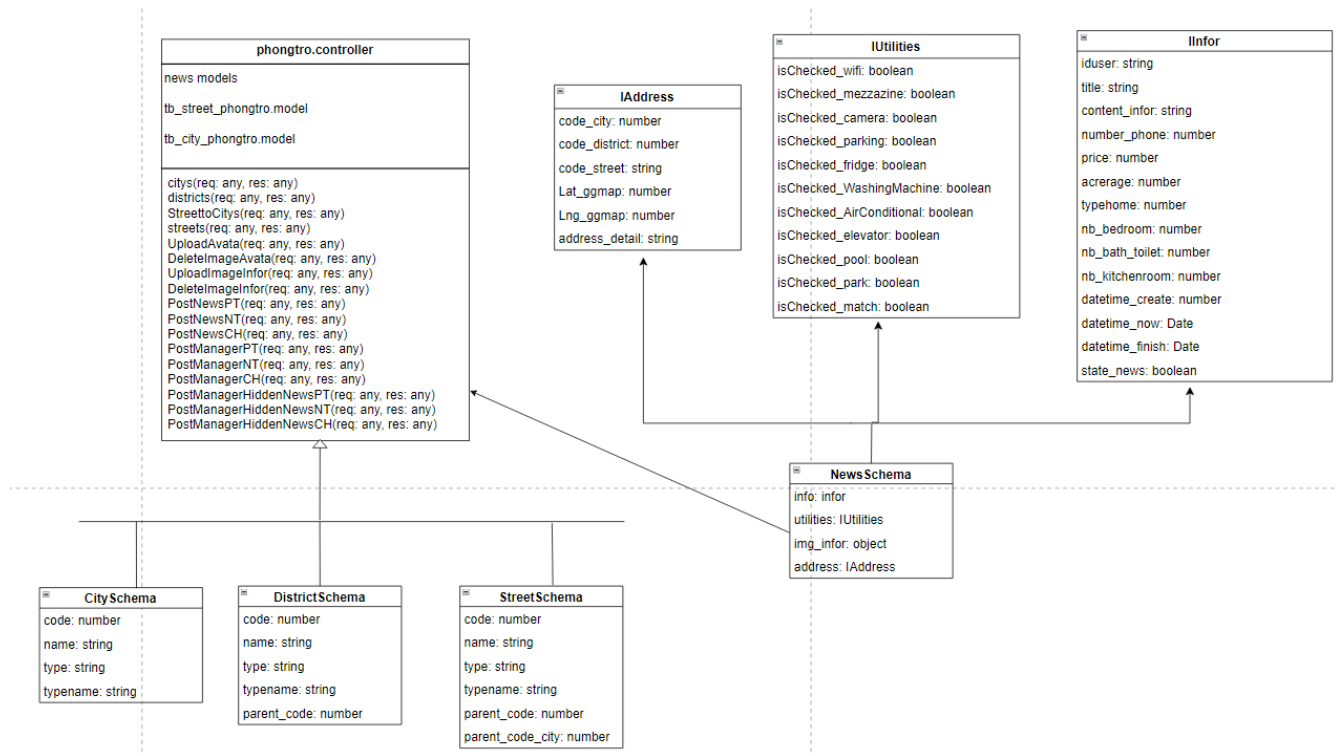
3.5. Biểu đồ lớp



Hình 13: Biểu đồ lớp cho HomePageController



Hình 14: Biểu đồ lớp cho userController



Hình 15: Biểu đồ cho lớp phongtro.controller

Chi tiết các bảng

Phương thức dành cho HomePage.controller

- news models
- tb_street_phongtro.model
- tb_city_phongtro.model
- cities(req: any, res: any)
- districts(req: any, res: any)
- StreettoCitys(req: any, res: any)
- streets(req: any, res: any)
- UploadAvata(req: any, res: any)
- DeleteImageAvata(req: any, res: any)
- UploadImageInfor(req: any, res: any)
- DeleteImageInfor(req: any, res: any)
- PostNewsPT(req: any, res: any)
- PostNewsNT(req: any, res: any)
- PostNewsCH(req: any, res: any)
- PostManagerPT(req: any, res: any)
- PostManagerNT(req: any, res: any)
- PostManagerCH(req: any, res: any)
- PostManagerHiddenNewsCH(req: any, res: any)
- PostManagerHiddenNewsPT(req: any, res: any)

- PostManagerHiddenNewsNT(req: any, res: any)

Phương thức dành cho phongtro.controller

- news models
- tb_street_phongtro.model
- tb_city_phongtro.model
- citys(req: any, res: any)
- districts(req: any, res: any)
- StreetCitys(req: any, res: any)
- streets(req: any, res: any)
- UploadAvata(req: any, res: any)
- DeleteImageAvata(req: any, res: any)
- UploadImageInfor(req: any, res: any)
- DeleteImageInfor(req: any, res: any)
- PostNewsPT(req: any, res: any)
- PostNewsNT(req: any, res: any)
- PostNewsCH(req: any, res: any)
- PostManagerPT(req: any, res: any)
- PostManagerNT(req: any, res: any)
- PostManagerCH(req: any, res: any)
- PostManagerHiddenNewsPT(req: any, res: any)
- PostManagerHiddenNewsNT(req: any, res: any)
- PostManagerHiddenNewsCH(req: any, res: any)

Các phương thức dành cho user.controller

- news models
- feedback
- login(req: any, res: any)
- loginsuccess(req: any, res: any)
- register(req: any, res: any)
- registersuccess(req: any, res: any)
- logout(req: any, res: any)
- getInforUser(req: any, res: any)
- accuracyPhone_NB(req: any, res: any)
- accuracyPhone_NB_key_OTP(req: any, res: any)
- changepasswordlogin(req: any, res: any)
- xFeedBack(req: any, res: any)
- GetFeedBack(req: any, res: any)
- getInforUserById(req: any, res: any)
- getInforUserEdit(req: any, res: any)
- UploadUserEditAvatar(req: any, res: any)

- OpenAvaterUser(req: any, res: any)
- EditedInforUser(req: any, res: any)

Bảng Infor: dành cho lớp userController

- firstname: string
- lastname: string
- male_female: boolean (Giới tính)
- img_avatar: string (Ảnh đại diện)

Bảng ILocal

- username: string
- password: string
- email: string

Bảng IAddress

- code_city: number
- code_district: number
- Lat_ggmap: number
- Lng_ggmap: number
- address_detail: string (chi tiết về địa chỉ)

Bảng IUtilities

- isChecked_wifi: boolean
- isChecked_mezzazine: boolean
- isChecked_camera: boolean
- isChecked_fridge: boolean
- isChecked_WashingMachine: boolean
- isChecked_AirConditional: boolean
- isChecked_pool: boolean
- isChecked_match: boolean

Bảng IInfor: dành cho lớp phongtro.controller

- iduser: string
- title: string (nội dung liên quan trong gói thông tin)
- content_infor: string
- price: number (Giá phòng)
- nb_bedroom: number
- nb_bath_toilet: number
- nb_kitchenroom: number
- typehome: number (loại hình cho thuê: thuê nguyên căn, phòng trọ)
- datetime_create: number
- datetime_now: Date
- datetime_finish: Date

Bảng NewsSchema

- code: number
- name: string
- type: string
- typename: string

Bảng FeedBackSchema

- info: Infor
- utilities: IUtilities
- img_infor: object (hình ảnh chi tiết trong các news)
- address: IAddress

Bảng CitySchema

- code: number
- name: string: (tên thành phố hoặc tỉnh thành)
- type:string
- typename: string

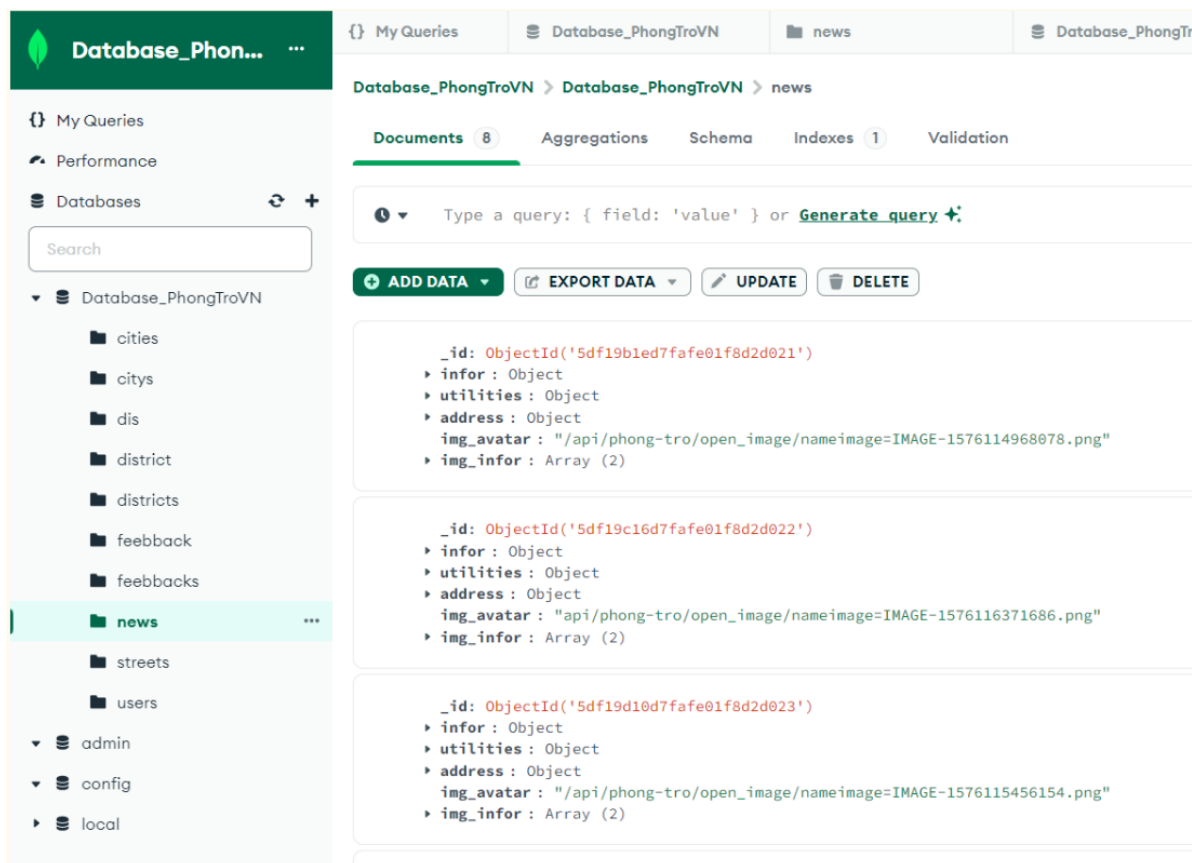
Bảng StreetSchema

- Code: number
- name: string (tên đường)
- type:string
- typename: string

Bảng DistrictSchema

- Code: number
- name: string: Địa chỉ cụ thể
- type:string
- typename: string
- parent_code: number

3.6. Cơ sở dữ liệu, bảng contract operation



Hình 16: Bảng cơ sở dữ liệu cho web

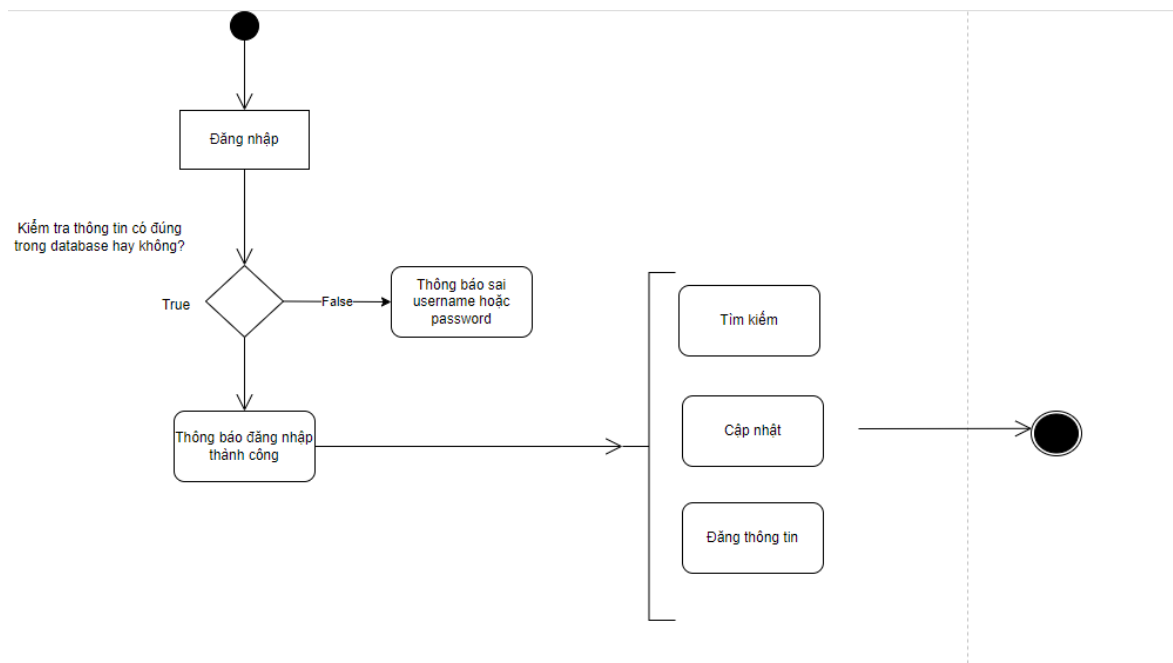
Hợp đồng hoạt động (Contract Operation)

Contract Operation ID	Mô tả
CO-01	Đăng nhập
CO-02	Đăng ký
CO-03	Hiện thị thông tin
CO-04	Cập nhật(thêm, xóa, sửa)
CO-05	Đăng xuất

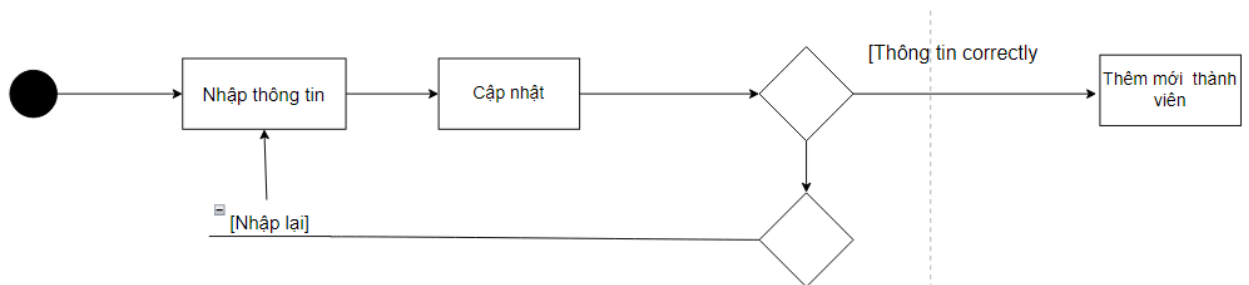
Bảng 6: Contract Operation của web

3.7. Biểu đồ hoạt động

Đây là bản vẽ tập trung vào mô tả các hoạt động, luồng xử lý bên trong hệ thống. Nó có thể được sử dụng để mô tả các quy trình nghiệp vụ trong hệ thống, các luồng của một chức năng hoặc các hoạt động của một đối tượng. Nó mô tả hoạt động chính và mối quan hệ giữa các hoạt động này trong quy trình. Hay có thể hiểu là mô tả cả luồng xử lý chính của hệ thống bao gồm các luồng con, luồng xử lý của các Use – Case gom lại mà thành.



Hình 17: Sơ đồ hoạt động chung cho toàn hệ thống



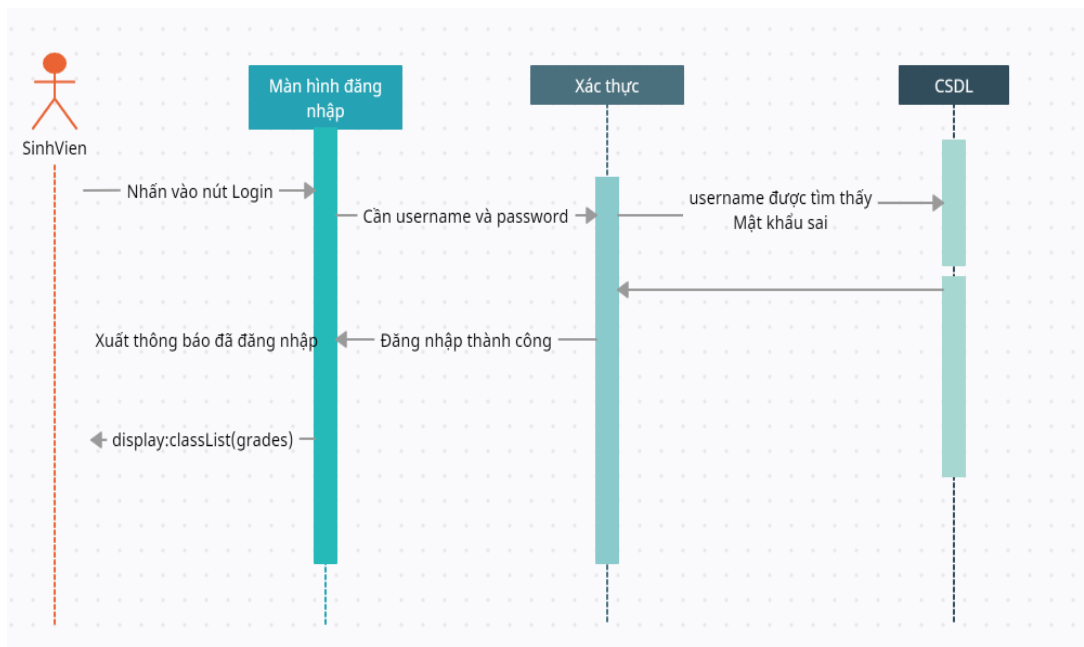
Hình 18: Sơ đồ hoạt động cho thao tác đăng ký

3.8. Biểu đồ tuần tự

Biểu đồ tuần tự là biểu đồ dùng để biểu diễn luồng dữ liệu của một UC theo trình tự thời gian [19]. Nó còn là bản vẽ xác định câu chuyện hậu trường của một chức năng. Câu chuyện hậu trường ở đây chính là sự tương tác giữa các nhóm đối tượng, các thông điệp được gửi và nhận giữa các đối tượng cũng như trình tự thời gian giữa những thông điệp đó.

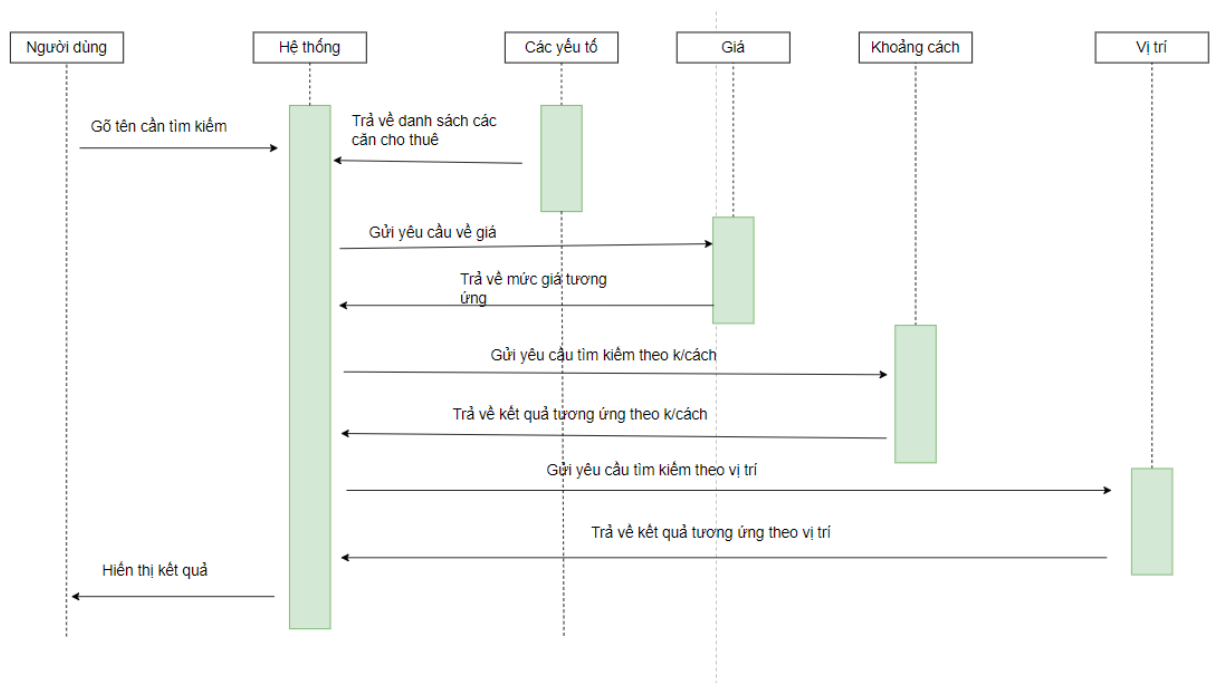
Mô tả chi tiết cho các task:

Đăng nhập



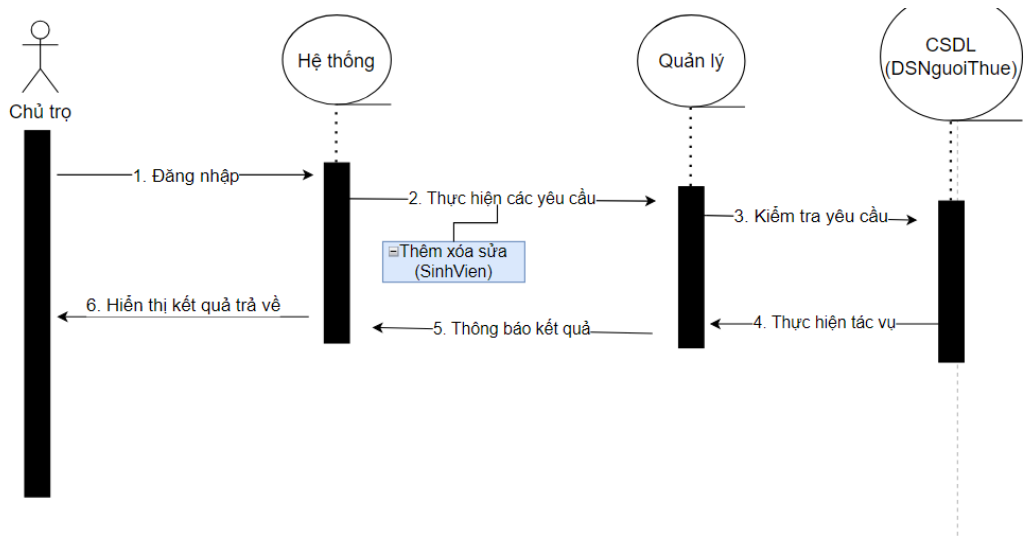
Hình 19: Biểu đồ cho chức năng đăng nhập

Tìm kiếm thông tin



Hình 20: Biểu đồ tuần tự cho chức năng tìm kiếm thông tin

Quản lý thông tin



Hình 21: Biểu đồ tuần tự cho chức năng quản lý thông tin (Cập nhật)

3.9. Yêu cầu và ràng buộc phi chức năng

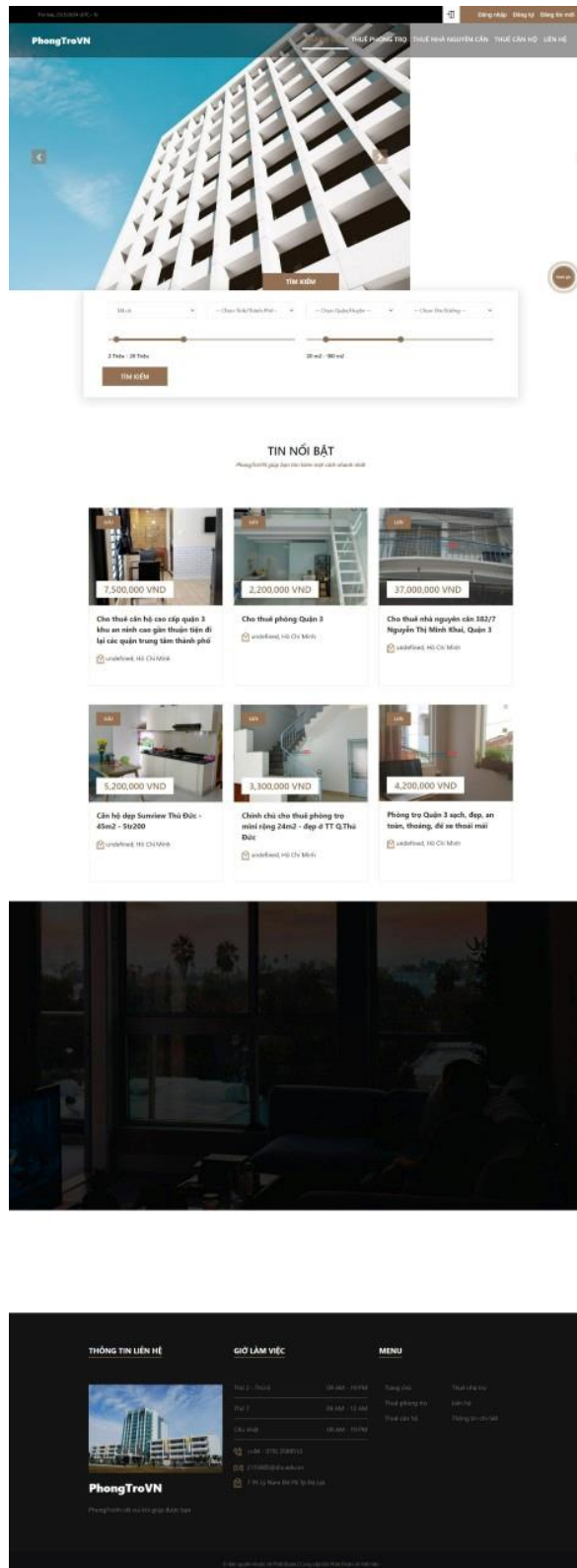
ID	Loại	Mô tả
NFA-01	Hiệu suất	Truy cập trang web nhanh
NFA-02	Khả năng sử dụng	Các đối tượng dùng web
NFA-03	Độ tin cậy	Thông tin chính xác

Bảng 7: Các yêu cầu chức năng của hệ thống

CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC

4.1. Kết quả đạt được

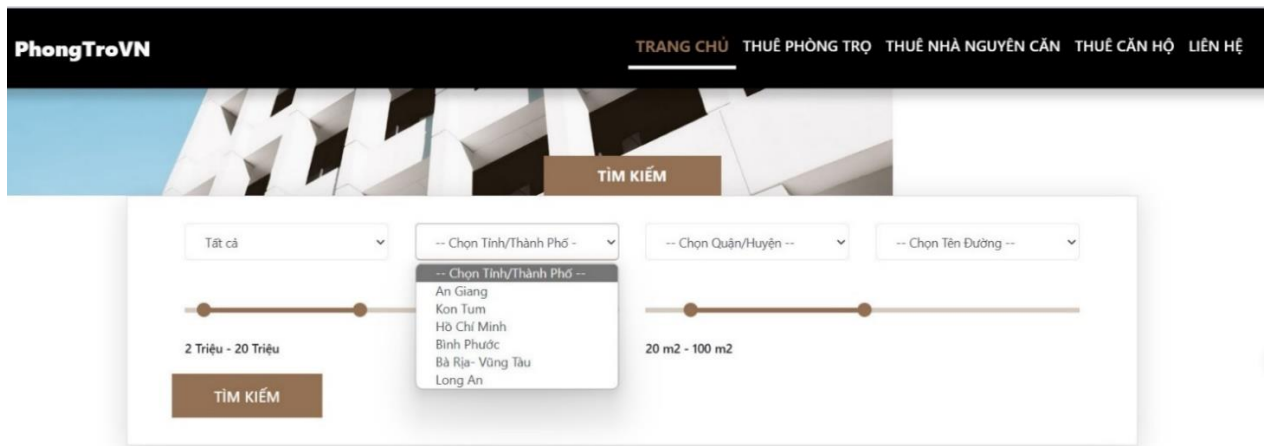
4.1.1. Giao diện chính



Hình 22: Giao diện chính của trang web

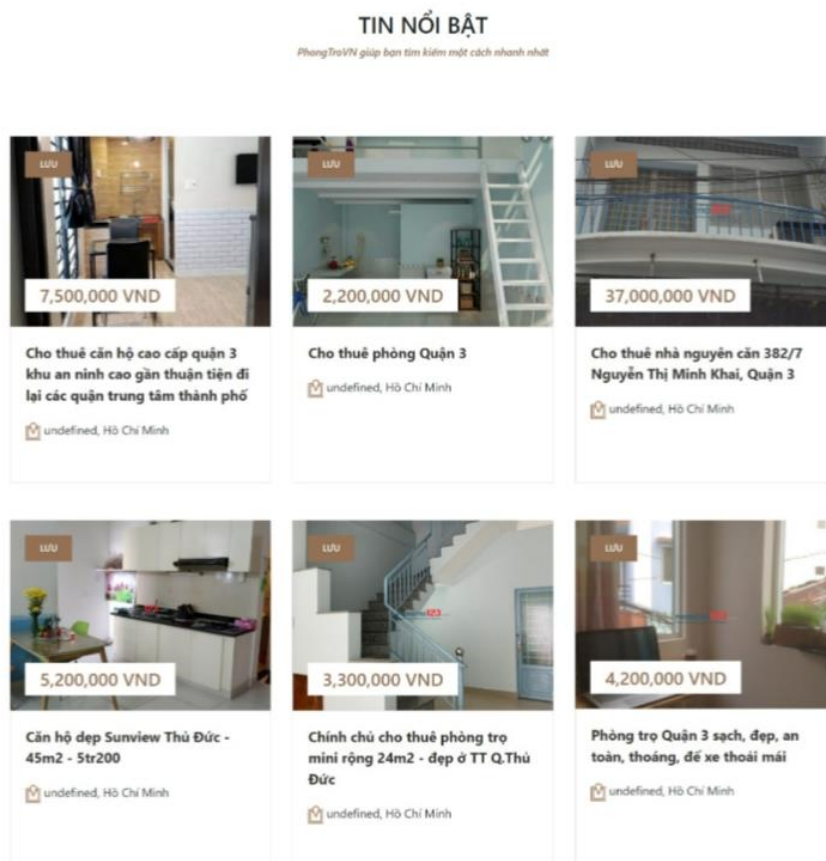
Ở phần cuối của trang chính có phần feedback cho phép client đánh giá nhờ các comment(với điều kiện đã thực hiện UC2 và hoàn thành UC1). Thể hiện Slogan cho trang web.

Bao gồm các thanh điều hướng giúp người dùng chuyển sang các hình thức (phòng trọ, nhà nguyên căn, căn hộ). Phía header của trang web sẽ có ba button thực hiện việc đăng nhập đăng ký, truy cập đến các task khác.



Hình 23: Thanh tìm kiếm theo ba yếu tố chính

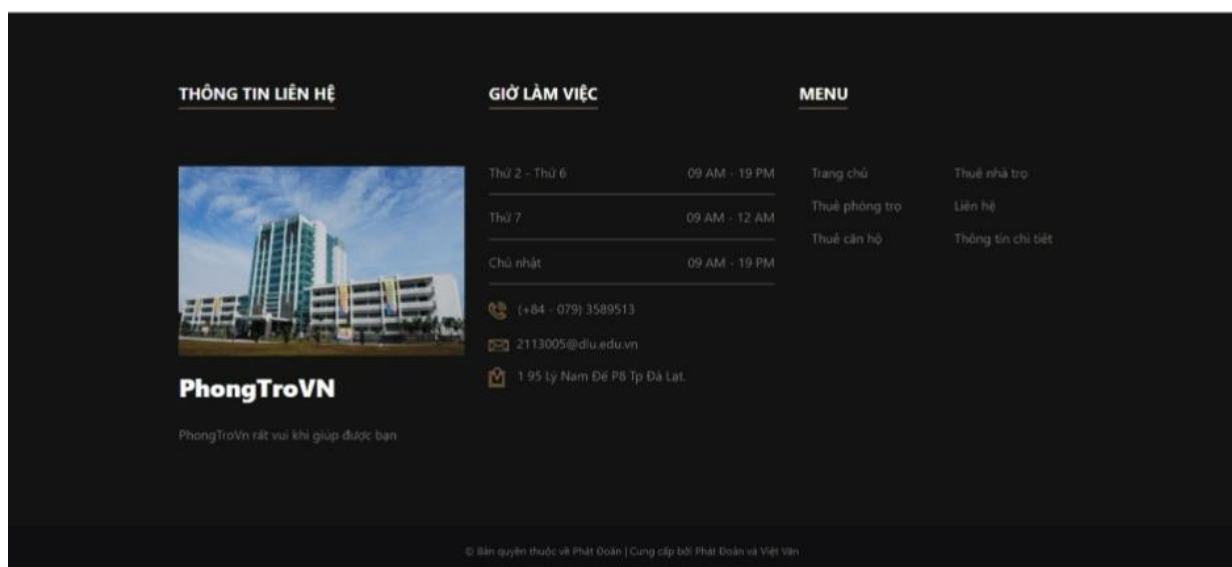
Đây là nơi cho phép tìm kiếm thông tin ngay ở UC2. Bao gồm việc tìm kiếm trên ba yếu tố chính: vị trí, giá, khoảng cách.



Hình 24: Các bài đăng nổi bật

Tác vụ tin nổi bật và tin phòng trọ cho phép người dùng có thể thấy ngay những

thông tin liên quan, giúp việc tra cứu trở nên dễ dàng hơn.



Hình 25: Thanh Footer của web

4.1.2. Hiện thị thông tin chi tiết

Thuê phòng trọ

Cho Thuê Phòng Đường Lý Nam Đế

Đà Lạt

Chi tiết

Địa chỉ	77 Lý Nam Đế .P8 Tp Đà Lạt		
Giá:	2,200,000 VND	Người đăng:	Lê Hà Hiếu Nghĩa
Diện tích:	Khoảng 20 m ²	Số điện thoại:	0975547178
Loại tin:	Thuê phòng trọ	Số phòng bếp:	1 Phòng
Số phòng ngủ:	1 Phòng	Số phòng toilet:	1 Phòng

Thông tin

Cho thuê phòng đường Lý Nam Đế cách KTX Đại Học Đà Lạt 100m Diện tích: 20m² Phòng sạch sẽ, thoáng mát, có ban công, wc trong phòng, giờ giấc tự do, cho nấu ăn, có chỗ để xe, wifi... Điện: 3,5k/kg Nước: 70k/người

Tiện ích

Wifi

Camera an ninh

Bản đồ

Hình 26: Thông tin chi tiết về phòng của người đăng (cho thuê)

Trang này mô tả các thông tin cơ bản về một căn hộ cho thuê: người đăng, giá, vị trí và các tiện ích đi kèm. Dưới phần này là một bản đồ hiển thị vị trí của căn hộ, với các đường phố và khu vực xung quanh. Bản đồ có dấu hiệu của Google và có lớp chồng lên và hiện lên “Chỉ dành cho mục đích phát triển.” Có một thông báo pop-up trên bản đồ cho biết “ *Trang này không thể tải Google Maps đúng cách.*”

4.2. Mã nguồn

Đường link dự án:

https://github.com/LeHaHieuNghia/QL_NhaTro_VN.git

4.2.1. googlemap.js

```
import React, { Component } from 'react';
import GoogleMapReact from 'google-map-react';
import icon_ggmap from './image_ggmap/icon_ggmap.png';
const AnyReactComponent = ({ text }) => <div>{text}</div>;
class GoogleMap extends Component {
  constructor(props) {
    super(props);
    this.state={
      center:{
        lat:10.8365648,
        lng:106.7560183
      },
      zoom:16
    }
  }
  async componentDidMount(){
    if (navigator.geolocation) {
      await
      navigator.geolocation.getCurrentPosition(this.getLocationForBrowser);
    } else {
      alert("Geolocation is not supported by this browser.");
    }
  }
  // getLocation={()=>{
  //   if (navigator.geolocation) {
  //
  //
  //
  //   navigator.geolocation.getCurrentPosition(this.getLocationForBrowser);
  //   } else {
  //     alert("Geolocation is not supported by this browser.");
  //   }
  // }
  // }
  getLocationForBrowser=(position)=>{
    this.setState({
      center:{
        lat:position.coords.latitude,
        lng:position.coords.longitude
      }
    })
    this.props.getLocationtoNewsNews(this.state.center)
  }
  handleApiLoaded=(map, maps)=>{
    new maps.Marker({
      position: this.state.center,
      map,
      title: 'Home Your!',
      icon:{
        url: icon_ggmap,

```



```

        scaledSize:
        {
            width:40,height:40
        }
    });
}
render() {
    return (
        <div style={{ height: '500px', width: '100%' }}>
            <GoogleMapReact
                bootstrapURLKeys={{key: 'AIzaSyDLhm8DHP3A6kMCIsiwQWUU-
pX5hSbyaQo'}}
                defaultCenter={this.state.center}
                defaultZoom={this.state.zoom}
                yesIWantToUseGoogleMapApiInternals
                onGoogleApiLoaded={({ map, maps }) =>
this.handleApiLoaded(map, maps)}
            >
            </GoogleMapReact>
            <AnyReactComponent
                lat={this.state.center.lat}
                lng={this.state.center.lng}
            />
        </div>
    );
}
}
export default GoogleMap;

```

4.2.2. menu.js

```

import React, { Component } from 'react';
import img_icon_menu from '../header/image_header/icons_menu.png'
import {NavLink,Link} from "react-router-dom";
class Menu extends Component {
    constructor(props) {
        super(props);
        this.state={
            scrolled:false,
        }
    }
    componentDidMount(){
        window.addEventListener('scroll',()=>{
            const isTop = window.scrollY < 150;
            if(isTop !==true) {
                this.setState({
                    scrolled:true
                })
            }
            else this.setState({scrolled:false})
        });
    }
    render() {
        return (
            <div className={!this.props.stateFiterandslide_imgApp

```

```

?
        "col-md-12 col-sm-12 col-xs-12 header2-menu "
        : "col-md-12 col-sm-12 col-xs-12 header2-
menu_detail_news" }>
                <nav className={!this.state.scrolled ?
"navbar navbar-expand-lg header2-menu-nav" : "navbar navbar-expand-lg
header2-menu-nav-fixed"} id="stickyHeader">
                        <Link className="navbar-
brand" to="/">PhongTroVN</Link>

                                <button className="navbar-
toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
                                        <span className="navbar-
toggler-icon">
                                                <img
src={img_icon_menu} alt="icomn_menu"></img>
                                                </span>
                                        </button>

                                <div className="collapse navbar-
collapse" id="navbarNav">
                                        <ul className="nav navbar-nav
ml-auto">
                                                <li className="nav-item
active">
                                                        <NavLink
className="nav-link header2-menu-nav-item" exact
activeClassName="header2-menu-nav-item-active" to="/"
onClick={this.props.StateFiterandslide_FuncApp}>Trang chủ</NavLink>
                                                        </li>
                                                        <li className="nav-item">
                                                                <NavLink
className="nav-link header2-menu-nav-item" to="/thue-phong-tro"
activeClassName="header2-menu-nav-item-active"
onClick={this.props.StateFiterandslide_FuncApp}>Thuê Phòng
trọ</NavLink>
                                                                </li>
                                                                <li className="nav-item">
                                                                        <NavLink
className="nav-link header2-menu-nav-item" activeClassName="header2-
menu-nav-item-active" to="/thue-can-ho"
onClick={this.props.StateFiterandslide_FuncApp}>Thuê căn hộ</NavLink>
                                                                        </li>
                                                                        <li className="nav-item">
                                                                                <NavLink
className="nav-link header2-menu-nav-item" activeClassName="header2-
menu-nav-item-active" to="/lien-he">Liên hệ</NavLink>
                                                                                </li>
                                                                        </ul>
                                        </div>
                                </nav>
        </div>

```

```

    );
  }
}
export default Menu;

```

4.2.3. feedback.js

```

var mongoose = require('mongoose');
var today = new Date();
var FeebBackSchema = new mongoose.Schema(
  {
    iduser:{
      type:String,
      required:true
    },
    titelfeedback:{
      type:String,
      required:true
    },
    contentfeedback:{
      type:String,
      required:true
    },
    timefeedback:{
      type:Date,
      default:today
    }
  },{
    versionKey:false
  }
);
var FeebBack= mongoose.model('FeebBack',FeebBackSchema,'FeedBack')
module.exports=FeebBack;

```

4.2.4. Home.router.js

```

var express = require("express")
var router = express.Router()
var controller= require('../controllers/HomePage.controller')
router.get("/tin-tong-hop",controller.News_All);
router.get("/tin-phong-tro",controller.News_RoomHome);
router.get("/tin-nha-tro",controller.News_HouseHome);
router.get("/tin-can-ho",controller.News_ApartmentHome);
router.get("/thong-tin-chi-tiet/:id",controller.News_Detail);
router.get("/thong-tin-chi-tiet/city/:code_city",controller.GetNameCity);
router.get("/thong-tin-chi-tiet/dictrict/:code_dictrict",controller.GetNameDictrict);
router.get("/tin-tuc-gan-do/:code_city/:code_dictrict",controller.NewsNears);
router.post("/tim-kiem/gia-tien/dien-tich",controller.NewsFilter);
module.exports=router;

```

4.2.5. Header.js

```

import React, { Component } from 'react';
import './header.css'
import Datenow from './datenow/datenow';

```

```

import img_icon_login from '../header/image_header/icon_login.png'
import LoginRegister from '../login_register/login_register';
import Menu from '../menu/menu';
import SlideHeader from '../slide_header/slide_header';
import Filter from './filter/filter';

class Header extends Component {
  render() {
    return (
      <div className="container-fluid">
        <div className="row header1">
          <div className="col-md-6 col-sm-6 col-xs-6
header1_info_left" >
            <Datenow/>
          </div>
          <div className="col-md-6 col-sm-6 col-xs-6
header1_info_right">
            <LoginRegister
clickPostNewstoApp={this.props.clickPostNewstoApp} clickMovedOnUserst
oApp={this.props.clickMovedOnUserstoApp}/>
            <div className="header1_info_right__div-img">
              <img src={img_icon_login}
alt="icon_login"/>
            </div>
          </div>
        </div>
        <div className="row header2">
          <div className="row header2-slide-menu"
id={!this.props.stateFiterandslide_imgApp ? "header2-menu" :""}>
            <Menu
StateFiterandslide_FuncApp={this.props.StateFiterandslide_FuncApp}
stateFiterandslide_imgApp={this.props.stateFit
erandslide_imgApp}/>
          </div>
          <div className="row header2-slide">
            <SlideHeader/>
          </div>
        </div>
        <div className="row">
          <div className="container form_filter">
            <Filter
StateFiterTyhomeNews_TF={this.props.StateFiterTyhomeNews_TF} // Hiện
hoặc ẩn fitter
GetNewsFiltertoApp={this.props.GetNewsFilterto
oApp}
GetTypeNewstoApp={this.props.GetTypeNewstoAp
// GetTypeNewstoApp
/>
          </div>
        </div>
      </div>
    )
  }
}

```

```

        </div>
    );
}
}
export default Header;

```

4.2.6. HomePage.controller.ts

```

import { Request, Response } from "express"
import News from "../models/News/news.model"
import Citys from "../models/tb_city.phongtro.model"
import Districts from "../models/tb_district.phongtro.model"

export const News_All = async (req: Request, res: Response) => {
  try {
    const result = await News.find({ "infor.state_news": true })
      .sort({ "infor.date_now": -1 })
      .limit(6)
      .exec()
    res.json({
      All_News: result,
    })
  } catch (err) {
    res.json({
      result: false,
      message: err,
    })
  }
}

export const News_Detail = async (req: Request, res: Response) => {
  try {
    const result = await News.find({ _id: req.params.id })
    res.json({
      news: result,
    })
  } catch (err) {
    res.json({
      result: false,
      message: err,
    })
  }
}

export const GetNameCity = async (req: Request, res: Response) => {
  try {
    const result = await Citys.findOne({ code: req.params.code_city
    })

    if (!result) throw new Error("not found")

    res.json({
      NameCity: result.name,
    })
  } catch (err) {

```

```

        res.json({
            result: false,
            message: err,
        })
    }
}

export const GetNameDictrict = async (req: Request, res: Response) => {
    {
        try {
            const result = await Districts.findOne({ code:
req.params.code_dictrict })
            if (!result) throw new Error("not found")
            res.json({
                NameDistricts: result.typename,
            })
        } catch (err) {
            res.json({
                result: false,
                message: err,
            })
        }
    }
}

export const NewsNears = async (req: Request, res: Response) => {
    try {
        const result = await News.find({
            "address.code_city": req.params.code_city,
            "address.code_dictrict": req.params.code_dictrict,
        })
        res.json({
            NewsNears: result,
        })
    } catch (err) {
        res.json({
            result: false,
            NewsNears: "",
            message: err,
        })
    }
}

export const News_RoomHome = async (req: Request, res: Response) => {
    try {
        const result = await News.find({
            "infor.state_news": 1,
            "infor.typehome": 1,
        })
        .sort({ "infor.date_now": -1 })
        .limit(6)
        .exec()
        res.json({
            NewsRoom: result,
        })
    } catch (err) {

```

```

        res.json({
          result: false,
          message: err,
        })
      }
    }
  }

export const News_HouseHome = async (req: Request, res: Response) =>
{
  try {
    const result = await News.find({
      "infor.state_news": 1,
      "infor.typehome": 2,
    })
    .sort({ "infor.date_now": -1 })
    .limit(6)
    .exec()
    res.json({
      HouseHome: result,
    })
  } catch (err) {
    res.json({
      result: false,
      message: err,
    })
  }
}

export const News_ApartmentHome = async (req: Request, res: Response)
=> {
  try {
    const result = await News.find({
      "infor.state_news": 1,
      "infor.typehome": 3,
    })
    .sort({ "infor.date_now": -1 })
    .limit(6)
    .exec()
    res.json({
      ApartmentHome: result,
    })
  } catch (err) {
    res.json({
      result: false,
      message: err,
    })
  }
}

export const NewsFilter = async (req: Request, res: Response) => {
  try {
    const { PriceMin, PriceMax, AcreageMin, AcreageMax } = req.body
    const result = await News.find({
      "infor.price": { $gte: PriceMin, $lte: PriceMax },
      "infor.acreage": { $gte: AcreageMin, $lte: AcreageMax },
    })
  }
}

```

```

        "infor.state_news": 1,
    }).exec()
    res.json({
        NewsFilter: result,
    })
} catch (err) {
    console.log(err)
    res.json({
        result: false,
        message: err,
    })
}
}
}

```

4.2.7. Phongtro.router.js

```

var express = require("express")
var router = express.Router()
var controller= require('../controllers/phongtro.controller')
//#####__Đăng tin mới_Address#####
router.post("/dang-tin-moi/chon-tinhTP",controller.citys);
router.get("/dang-tin-moi/danh-sach-duong/:code_city",controller.StreetstoCitys);
router.post("/dang-tin-moi/chon-QH",controller.districts);
router.post("/dang-tin-moi/chon-Duong",controller.streets);
//#####__Đăng tin upload_Image#####
router.post("/dang-tin-moi/upload_image",controller.UploadAvarata);
router.get("/open_image/nameimage=:imagenam",controller.getImageAvarata);
router.post("/dang-tin-moi/xoa-anh-dai-dien",controller.DeleteImageAvarata);
router.post("/dang-tin-moi/up-load_hinh-mota",controller.UploadImageInfor);
router.post("/dang-tin-moi/xoa-anh-mota",controller.DeleteImageInfor);
//#####__Đăng tin mới_Finish#####
router.post("/dang-tin-moi/phong-tro",controller.PostNewsPT);
router.post("/dang-tin-moi/nha-tro",controller.PostNewsNT);
router.post("/dang-tin-moi/can-ho",controller.PostNewsCH);
//#####__Phong_Tro#####
router.get("/quan-ly-tin-dang/phong-tro",controller.PostManagerPT);
router.get("/quan-ly-tin-dang/nha-tro",controller.PostManagerNT);
router.get("/quan-ly-tin-dang/can-ho",controller.PostManagerCH);
router.post("/quan-ly-tin-dang/an-tin-tuc-phong-tro",controller.PostManagerHiddenNewsPT);
router.post("/quan-ly-tin-dang/an-tin-tuc-nha-tro",controller.PostManagerHiddenNewsNT);
router.post("/quan-ly-tin-dang/an-tin-tuc-can-ho",controller.PostManagerHiddenNewsCH);
module.exports=router;

```

4.2.8. tb_city_phongtro.ts

```

import mongoose, { Document, Schema, Model, model} from 'mongoose';
interface ICity extends Document {
    code: number;
    name: string;

```



```

    type: string;
    typename: string;
  }
  const CitySchema = new Schema(
    {
      code: {
        type: Number,
        required: true
      },
      name: {
        type: String,
        required: true
      },
      type: {
        type: String,
        required: true
      },
      typename: {
        type: String,
        required: true
      }
    }, {
      versionKey: false
    }
  );
  export default model('citys', CitySchema);

```

4.3. Kết quả kiểm thử

Sau khi hoàn thiện được trang web, bước quan trọng là kiểm thử, giúp đánh giá tính chính xác của sản phẩm web, từ đó phát hiện lỗi sai và khắc phục liền. Nó đảm bảo sản phẩm phải đáp ứng chính xác theo các yêu cầu của khách hàng đề ra. Ngoài ra, nó còn tạo điều kiện cho người dùng tận dụng tối đa tư duy sáng tạo để bạn có thể phát hiện ra những điểm mà người khác chưa nhìn thấy.

Công cụ Postman được sử dụng như một công cụ phổ biến để đơn giản hóa quy trình làm việc API trong kiểm thử và phát triển.

Bảng dưới đây đề cập đến các trường hợp (test case) có thể xuất hiện

Test Scenario	Requirement	Test Scenario Description
ID001	S1	Kiểm thử giao diện
ID002	S2	Kiểm thử đăng nhập
ID003	S3	Kiểm thử chức năng đăng ký
ID004	S4	Kiểm thử quên và lấy lại mật khẩu

ID005	S5	Kiểm thử chức năng quản lý các trọ
ID006	S6	Kiểm thử chức năng cập nhật (thêm, xóa sửa)

Bảng 8: Bảng các test scenario trong bước Manual Testing

Test case 1: Kiểm thử giao diện

S #	Prerequisites:	Các tiêu chí
1	Access to Chrome Browser	font chữ: 20px in đậm cho các tiêu đề nổi bật
2	Navigate to some form in web	Màn hình 1920x1080
3	Check size of button, image, font-size, font-weight	

Bảng 9: Điều kiện tiên quyết với thao tác kiểm thử giao diện

Step#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Access to website	Page should open	As Expected	Pass
2	Changing screen size	website changes into smaller screen	As Expected	Not executed
3	View on smartphone	website changes into type web app	As Expected	Suspended

Test case 2: Kiểm thử đăng nhập

S #	Prerequisites:
1	Truy cập thành công trên Google Chrome
2	Chuyển đến form đăng nhập
3	Username và mật khẩu phải trùng khớp

Bảng 10: Điều kiện tiên quyết với thao tác đăng nhập

Step#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Nhập username và mật khẩu	Thông tin được đưa vào form	As Expected	Pass
2	Nhấn nút đăng nhập	Người dùng đăng nhập thành công	As Expected	Pass
3	In thông báo	In thông báo đăng nhập thành công, có thể tiếp cận các chức năng khác	As Expected	Suspended

Test case 3: Kiểm thử đăng ký

S #	Prerequisites	Các tiêu chí
1	Access to Chrome Browser	Họ tên: Nguyễn Văn A
2	Valid information	Địa chỉ: 37 Trần Quốc Toàn, phường 2, Quận Tân Bình, TP.HCM
3	Password must contain number, string and special character	Số điện thoại: 0927376893
		email: zomb9998@gmail.com
		password: 12fn736@#5&\$\$#

Bảng 11: Điều kiện tiên quyết đối với thao tác đăng ký

Step#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to Login form	Page should open	As Expected	Pass
2	Enter full name	Information has been entered	As Expected	Pass
3	Enter address	Information has been entered	As Expected	Pass
4	Enter phone number	Information has been entered	As Expected	Pass
5	Navigate to Login form	Information has been entered	As Expected	Pass Fail => if Prerequisites 3 is wrong
6	Enter full name	Customer can logged in	As Expected	Pass
7	Navigate to main page	Completing log in in previous stages	As Expected	Pass

Test case 4: Chức năng lấy lại mật khẩu

S #	Prerequisites	Các tiêu chí
1	Click on Forgot Password => navigate to scence change password	Họ tên: Nguyễn Văn A
2	Verify that the Current Password, New Password, and Confirm password fields are displayed on the screen ?	Số điện thoại: 0927376893
	Verify that user UI of currently password, new password and verify_password is correctly based on database?	Địa chỉ: 37 Trần Quốc Toàn, phường 2, Quận Tân Bình, TP.HCM
		email: zomb9998@gmail.com
		Mật khẩu cũ: 12fn736@#5&\$\$#
		Mật khẩu mới: cocaimatkhaucungquen

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong khuôn khổ của đề án này, nhóm đã thực hiện được nhiều nhóm công việc, từ xác định yêu cầu -> phân tích thiết kế -> viết mã nguồn -> hoàn thiện slide, báo cáo. Qua đó, các thành viên đã tích lũy được nhiều kỹ năng làm việc nhóm, nắm bắt các công nghệ mới và rút ra được các bài học kinh nghiệm cho bản thân.

Những kết quả mà nhóm đã đạt được:

- Xây dựng thành công một trang web đáp ứng nhu cầu của người dùng : thuê nhà
- Cho phép người dùng tìm kiếm nhà trọ trong phạm vi tính bằng mét và giá tiền
- Dùng API kết nối thành công CSDL

Bên cạnh đó, những hạn chế mà nhóm chưa thể khắc phục bao gồm:

- Tính bảo mật vẫn chưa đảm bảo
- Chưa tối ưu hóa được về mặt hiệu suất và chức năng
- Một số chức năng chưa hoàn thiện được theo yêu cầu chung

Trong quá trình thực hiện đề tài, nhóm chắc chắn không thể tránh khỏi thiếu sót kèm theo đó nhu cầu và công nghệ luôn thay đổi với tình hình mới, vậy nên, hướng phát triển trong tương lai đề tài sẽ bao gồm:

- Mở rộng thêm thao tác tìm theo các tỉnh thành phố khác, không chỉ trong phạm vi của tỉnh Lâm Đồng
- Hoàn thiện và thêm các chức năng mới, tăng hiệu suất và tính hiệu quả của website

TÀI LIỆU THAM KHẢO

- [1] H. Nhung, "Cảnh giác thuê phòng trọ qua mạng khi đi thi đại học," 16 6 2014. [Online]. Available: <https://giaoduc.net.vn/canh-giac-thue-phong-tro-quua-mang-khi-di-thi-dai-hoc-post146082.gd>.
- [2] PV, "Thị trường thuê trọ TP Hồ Chí Minh bắt đầu sôi nổi," 14 8 2023. [Online]. Available: <https://vtv.vn/goc-doanh-nghiep/thi-truong-thue-tro-tp-ho-chi-minh-bat-dau-soi-noi-20230814190049652.htm>.
- [3] NGUYỄN THỊ HỒNG PHÚC - LÊ MỘNG KHA, "Phân tích các yếu tố ảnh hưởng đến quyết định thuê nhà trọ của sinh viên Trường Đại học Trà Vinh," Trà Vinh, 2020.
- [4] Đ. Hòa, "TP Hồ Chí Minh có 37 ngàn phòng trọ nhỏ hơn 10m2," 25 1 2022. [Online]. Available: <https://baophapluat.vn/tp-ho-chi-minh-co-37-ngan-phong-tro-nho-hon-10m2-post431909.html>.
- [5] Douglas T. Kenrick, Vladas Griskevicius, Steven L. Neuberg, Mark Schaller, "Renovating the Pyramid of Needs: Contemporary Extensions Built Upon Ancient Foundations," *National Library of Medicine*, vol. 5, no. 3, pp. 292-314, 2 10 2011.
- [6] Omar Saeed, Ahmer Asif, "Classics in the History of Psychology," in *A Theory of Human Motivation*, vol. 50, Toronto, Ontario, pp. 370-396.
- [7] C.THÀNH, "Lâm Đồng online," 16 4 2020. [Online]. Available: <https://baolamdong.vn/doi-song/202004/da-lat-hon-2000-phong-giam-gia-cho-nguoi-thue-tro-2998982/>. [Accessed 16 4 2020].
- [8] V. Trọng, "Lâm Đồng đứng thứ 4 các tỉnh, thành trong nước được người dân muốn đến sống nhất," 15 8 2023. [Online]. Available: <https://baolamdong.vn/du-lich/202308/lam-dong-dung-thu-4-cac-tinh-thanh-trong-nuoc-duoc-nguoi-dan-muon-den-song-nhat-e071876/>.
- [9] Philip Kotler, Gary Armstrong, "Principles of Marketing," in *Principles of Marketing*, 2022, pp. 26-62.
- [10] P. Shankman, *Customer Service: New Rules for a Social Media World*, 2010.
- [11] "Hiring ReactJS Developers: The Best Practices For Strengthening Your Programming Team," 31 1 2023. [Online]. Available: <https://www.apac-insider.com/hiring-reactjs-developers-the-best-practices-for-strengthening-your-programming-team/>.
- [12] P. T. Dương, "Hướng dẫn chạy Reactjs trên Visual Studio Code," FPT Polytechnic, 4 3 2024. [Online]. Available: <https://caodang.fpt.edu.vn/tin-tuc-poly/huong-dan-chay-reactjs-tren-visual-studio-code.html>.
- [13] 200Lab, "Các VS Code extension dành cho React Developer," 200Lab Blog, 22 7 2022. [Online]. Available: <https://200lab.io/blog/vs-code-extension-cho-react/>.
- [14] D. T. Được, "Thiết kế cơ sở dữ liệu bằng MongoDB sao cho chuẩn," DTD, Hồ Chí Minh, 2023.
- [15] T. Uyen, "MongoDB là gì? Định nghĩa và Hiểu rõ A-Z về MongoDB," 29 6 2023. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK574574/#_article-137541_s2_.

- [16] T. D. Thanh, "Bootstrap là gì? 2 cách cài đặt Bootstrap cực đơn giản," 15 4 2022. [Trực tuyến]. Available: https://bkhost.vn/blog/bootstrap/#bootstrap_la_gi.
- [17] W. Rågstad, "How to use \$.ajax()," 20 1 2019. [Online]. Available: <https://dev.to/williamragstad/how-to-use-ajax-3b5e>.
- [18] H. Linh, "Mô hình Waterfall là gì? Khi nào sử dụng Waterfall?," [Online]. Available: <https://pma.edu.vn/blogs/mo-hinh-waterfall-la-gi-khi-nao-su-dung-waterfall/>. [Accessed 28 9 2022].
- [19] T. D. Linh, "Đồ thị UML và tầm quan trọng của đồ thị UML trong phát triển phần mềm[2]," 20 3 2020. [Online]. Available: <https://viblo.asia/p/do-thi-uml-va-tam-quan-trong-cua-do-thi-uml-trong-phat-trien-phan-mem2-djeZ1oGR5Wz>.