

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**Thiết kế và xây dựng trang web thuê và cho thuê  
homestay**

**NGUYỄN THỊ THU TRÀ**  
tra.ntt176890@sis.hust.edu.vn

**Ngành Công nghệ thông tin  
Chương trình Việt Nhật**

**Giảng viên hướng dẫn:** TS. Trần Hải Anh

Chữ ký GVHD

**Khoa:** Kỹ thuật máy tính

**Trường:** Công nghệ thông tin và Truyền thông

**HÀ NỘI, 08/2022**

# **LỜI CẢM ƠN**

Lời đầu tiên em xin được gửi lời cảm ơn chân thành nhất tới thầy giáo, TS.Trần Hải Anh đã tận tình hướng dẫn em trong suốt quá trình thực hiện đồ án tốt nghiệp vừa qua. Thầy không những hướng dẫn, tư vấn định hướng đề tài, các chức năng của hệ thống, mà còn tạo điều kiện thuận lợi cho em tiếp xúc, học hỏi thêm các kiến thức công nghệ mới thông qua những buổi báo cáo, chia sẻ rất gần gũi. Em chúc thầy luôn mạnh khỏe, công tác tốt, tiếp tục truyền động lực học tập và làm việc cho các bạn sinh viên.

Em cũng xin gửi lời cảm ơn đến các thầy cô giáo thuộc trường Đại học Bách khoa Hà Nội nói chung đã truyền đạt những kiến thức bổ ích không chỉ trong phạm vi môn học mà còn là cả những kinh nghiệm quý báu trong cuộc sống của mình, giúp em có được những kiến thức và kinh nghiệm như ngày hôm nay.

Tiếp đó xin được gửi lời cảm ơn tới những người bạn, những anh chị em trong khoa HEDSPI trường Đại học Bách Khoa Hà Nội đã cùng mình sống trọn quãng thời gian năm năm làm sinh viên. Cùng mình vượt qua những kỳ thi khốc liệt, những môn học khó nhằn, truyền động lực cho mình trong suốt quá trình làm đồ án.

# TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong những năm gần đây, khi kinh tế ngày càng phát triển thì con người càng có nhu cầu nâng cao chất lượng cuộc sống của bản thân, chính vì lẽ đó mà nhu cầu du lịch cũng ngày càng tăng cao. Có thể thấy không ít những website cho phép chúng ta có thể tìm kiếm và đặt được những căn phòng nghỉ dưỡng có chất lượng và dịch vụ tốt của khách sạn chỉ bằng những thao tác đơn giản. Nhưng với xu thế giới trẻ hiện nay, khách sạn không còn là mô hình duy nhất được lựa chọn mà thay vào đó là các kiểu homestay, nhà nghỉ du lịch kiểu gia đình... Các nền tảng tiêu biểu trên thế giới có thể kể đến như Booking hay Airbnb. Tuy nhiên chúng đều hướng đến lượng người dùng toàn cầu, không có dành riêng cho người dùng Việt Nam. Ngoài ra, việc trải nghiệm du lịch tự túc không giống như việc đi theo tour du lịch, mà cần phải tìm hiểu kỹ, tham khảo ý kiến của những người đi trước về nơi ở, địa điểm để tạo ra một lịch trình du lịch lý tưởng. Việc này đối với những người du lịch tự túc hiện nay là tương đối khó khăn. Nhận thấy vấn đề này, đồ án này tập trung phát triển một hệ thống nhằm kết nối giữa chủ nhà, chủ homestay và khách lưu trú giúp họ có thể liên hệ với nhau khi có nhu cầu về nhà trọ, nơi nghỉ dưỡng. Một mặt, hệ thống giúp cá nhân những người có nhu cầu dễ dàng tiếp cận các thông tin của những địa điểm tham quan hay tìm cho mình một lịch trình du lịch phù hợp mỗi khi đến một địa điểm bất kỳ. Mặt khác, hệ thống giúp các chủ nhà có thể quảng bá thông tin nơi ở của mình rộng rãi tới các cá nhân.

Để đạt được điều này, hướng tiếp cận của đồ án là đưa ra một hệ thống mà người dùng có thể dễ dàng tìm kiếm thông tin về nơi nghỉ ngơi phù hợp với tiêu chí của bản thân. Bên cạnh đó, đồ án cũng cung cấp một hệ thống quản lý dành cho quản trị viên và chủ nhà khi muốn đăng tải, quản lý thông tin về nơi nghỉ của mình. Điểm đặc biệt của đồ án này là việc người dùng có thể chia sẻ lịch trình du lịch cá nhân của mình cho người khác cùng tham khảo khi họ đã từng lưu trú ở địa điểm du lịch đó, từ đó giúp nâng cao chất lượng mỗi chuyến du lịch của mọi người.

## MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>1</b>
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án .....	2
<b>CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....</b>	<b>4</b>
2.1 Khảo sát hiện trạng .....	4
2.2 Tổng quan chức năng .....	5
2.2.1 Biểu đồ use case tổng quát .....	5
2.2.2 Biểu đồ use case phân rã.....	6
2.3 Đặc tả chức năng .....	10
2.3.1 Đặc tả use case "Tìm kiếm homestay" .....	10
2.3.2 Đặc tả use case "Đặt phòng" .....	11
2.3.3 Đặc tả use case "Thay đổi trạng thái đơn đặt phòng" .....	12
2.3.4 Đặc tả use case "Đánh giá homestay" .....	12
2.3.5 Đặc tả use case "Review địa điểm du lịch".....	13
2.3.6 Đặc tả use case "Quản lý homestay" .....	14
2.4 Yêu cầu phi chức năng .....	15
<b>CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....</b>	<b>16</b>
3.1 Công nghệ sử dụng cho phần Frontend .....	16
3.1.1 Thư viện ReactJS .....	16
3.2 Công nghệ sử dụng cho phần Backend.....	18
3.2.1 NodeJS và Framework Express.....	18
3.2.2 REST API .....	21

3.2.3 MongoDB .....	21
<b>CHƯƠNG 4. THIẾT KẾ VÀ TRIỂN KHAI ỨNG DỤNG .....</b>	<b>23</b>
4.1 Thiết kế kiến trúc.....	23
4.1.1 Lựa chọn kiến trúc phần mềm .....	23
4.1.2 Kiến trúc Frontend .....	24
4.1.3 Kiến trúc Backend.....	25
4.2 Thiết kế chi tiết.....	26
4.2.1 Thiết kế giao diện .....	26
4.2.2 Thiết kế cơ sở dữ liệu .....	27
4.3 Xây dựng ứng dụng.....	33
4.3.1 Thư viện và công cụ sử dụng .....	33
4.3.2 Kết quả đạt được .....	33
4.3.3 Minh họa các chức năng chính .....	33
4.4 Kiểm thử.....	39
4.5 Triển khai .....	40
<b>CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....</b>	<b>41</b>
5.1 Thiết kế và xây dựng frontend và backend.....	41
5.2 Mô tả một số chức năng độc đáo .....	41
5.2.1 Cơ chế tổng hợp các bài nhận xét theo địa điểm. ....	41
5.2.2 Cơ chế tìm phòng trống theo các tiêu chí khác nhau .....	43
5.2.3 Cơ chế trao đổi thông tin trực tiếp giữa người dùng và host .....	46
<b>CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>49</b>
6.1 Kết luận .....	49
6.2 Hướng phát triển.....	50
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>51</b>
<b>PHỤ LỤC.....</b>	<b>51</b>

## DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ use case tổng quan . . . . .	5
Hình 2.2	Biểu đồ use case phân rã “Quản lý thông tin cá nhân” . . . . .	6
Hình 2.3	Biểu đồ use case phân rã “Tìm kiếm homestay” . . . . .	7
Hình 2.4	Biểu đồ use case phân rã “Quản lý homestay của tôi” . . . . .	8
Hình 2.5	Biểu đồ use case phân rã “Quản lý lịch sử đặt phòng” . . . . .	8
Hình 2.6	Biểu đồ use case phân rã “Quản lý hệ thống” . . . . .	9
Hình 3.1	Cây cấu trúc DOM của một trang Web . . . . .	16
Hình 3.2	Phân chia component trong Web . . . . .	17
Hình 3.3	DOM ảo và quá trình cập nhật dữ liệu . . . . .	18
Hình 3.4	Xử lý luồng của web truyền thống . . . . .	19
Hình 3.5	Xử lý luồng của NodeJS . . . . .	19
Hình 3.6	Cấu trúc của express . . . . .	20
Hình 3.7	Rest api . . . . .	21
Hình 4.1	Kiến trúc tổng quan của hệ thống . . . . .	23
Hình 4.2	Kiến trúc Frontend trang Người dùng và Host . . . . .	24
Hình 4.3	Kiến trúc Frontend trang trị viên . . . . .	25
Hình 4.4	Kiến trúc Backend . . . . .	25
Hình 4.5	Giao diện mockup Desktop . . . . .	26
Hình 4.6	Giao diện thực tế Desktop . . . . .	26
Hình 4.7	Giao diện mockup Desktop . . . . .	27
Hình 4.8	Giao diện thực tế Desktop . . . . .	27
Hình 4.9	Sơ đồ thiết kế cơ sở dữ liệu . . . . .	27
Hình 4.10	Giao diện trang tìm kiếm phòng . . . . .	34
Hình 4.11	Giao diện trang lựa chọn phòng . . . . .	34
Hình 4.12	Giao diện biểu mẫu thông tin đặt phòng . . . . .	35
Hình 4.13	Giao diện chọn phương thức thanh toán . . . . .	35
Hình 4.14	Giao diện nhập thông tin thanh toán . . . . .	35
Hình 4.15	Giao diện yêu cầu chọn phương thức thanh toán . . . . .	36
Hình 4.16	Giao diện yêu cầu nhập thông tin thanh toán . . . . .	36
Hình 4.17	Giao diện sau khi đặt phòng thành công . . . . .	36
Hình 4.18	Giao diện danh sách các đơn đặt phòng . . . . .	36
Hình 4.19	Giao diện sau chi tiết đơn đặt phòng . . . . .	37
Hình 4.20	Giao diện biểu mẫu nhập thông tin đánh giá . . . . .	37
Hình 4.21	Giao diện biểu mẫu nhập thông tin review . . . . .	38

Hình 5.1	Phân chia các bài review theo từng tỉnh thành . . . . .	42
Hình 5.2	Ví dụ kết quả khi chọn xem review của một tỉnh thành . . . . .	42
Hình 5.3	Tìm kiếm theo thời gian và địa điểm . . . . .	44
Hình 5.4	Sắp xếp giảm dần theo sao . . . . .	44
Hình 5.5	Sắp xếp tăng dần theo giá . . . . .	45
Hình 5.6	Tìm kiếm theo tên . . . . .	45
Hình 5.8	Boxchat cá nhân . . . . .	47
Hình 5.9	Boxchat của Host . . . . .	47

## DANH MỤC BẢNG BIỂU

Bảng 2.1	So sánh các hệ thống đặt phòng . . . . .	4
Bảng 2.2	Đặc tả use case “Tìm kiếm homestay” . . . . .	10
Bảng 2.3	Đặc tả use case “Đặt phòng” . . . . .	11
Bảng 2.4	Đặc tả use case “Thay đổi trạng thái đơn đặt phòng” . . . . .	12
Bảng 2.5	Đặc tả use case “Đánh giá homestay” . . . . .	12
Bảng 2.6	Đặc tả use case “Review địa điểm du lịch” . . . . .	13
Bảng 2.7	Đặc tả use case “Quản lý homestay” . . . . .	14
Bảng 3.1	So sánh tên gọi giữa CSDL quan hệ và CSDL MongoDB . . .	22
Bảng 4.1	Cấu trúc collection user . . . . .	28
Bảng 4.2	Cấu trúc collection homestay . . . . .	28
Bảng 4.3	Cấu trúc collection category . . . . .	29
Bảng 4.4	Cấu trúc collection room . . . . .	29
Bảng 4.5	Cấu trúc collection order . . . . .	30
Bảng 4.6	Cấu trúc collection comment . . . . .	30
Bảng 4.7	Cấu trúc collection review . . . . .	31
Bảng 4.8	Cấu trúc collection likeReview . . . . .	31
Bảng 4.9	Cấu trúc collection conversation . . . . .	32
Bảng 4.10	Cấu trúc collection message . . . . .	32
Bảng 4.11	Danh sách thư viện và công cụ sử dụng . . . . .	33
Bảng 4.12	Các thông tin về hệ thống . . . . .	33
Bảng 4.13	Kết quả kiểm thử trên các trình duyệt máy tính . . . . .	39
Bảng 5.1	Quy luật tạo thông báo/hành động . . . . .	43
Bảng 5.2	Các loại boxchat . . . . .	46
Bảng 5.3	So sánh ưu nhược điểm của Firebase . . . . .	48

# DANH MỤC CÁC TỪ VIẾT TẮT

Viết tắt	Tên tiếng Anh	Tên tiếng Việt
<b>API</b>	Application Programming Interface	Giao diện lập trình ứng dụng
<b>HTML</b>	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
<b>CRUD</b>	Create/Read/Update/Delete	Thao tác xem thêm sửa xoá cơ bản
<b>CSDL</b>		Cơ sở dữ liệu
<b>Host</b>		Chủ nhà, chủ homestay
<b>CNTT</b>		Công nghệ thông tin
<b>DATN</b>		Đồ án tốt nghiệp

# DANH MỤC THUẬT NGỮ

Viết tắt	Tên tiếng Anh
<b>Browser</b>	Trình duyệt
<b>Firebase</b>	Dịch vụ cơ sở dữ liệu được hoạt động ở trên nền tảng đám mây (Cloud)
<b>API</b>	Application Programming Interface – giao diện lập trình ứng dụng, là cơ chế cho phép 2 thành phần phần mềm giao tiếp với nhau bằng một tập hợp các định nghĩa và giao thức
<b>Backend</b>	Tầng truy cập xử lý dữ liệu, xử lý nghiệp vụ của phần mềm
<b>Frontend</b>	Tầng trình diễn (giao diện người dùng)
<b>Component</b>	Các thành phần giao diện độc lập có thể tái sử dụng trong thư viện ReactJS
<b>DOM</b>	Document Object Model, được dùng để truy xuất và thao tác trên các tài liệu có cấu trúc dạng HTML hay XML
<b>Review</b>	Đánh giá, bài nhận xét, phê bình

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

Ngày nay nhu cầu giải tỏa sau những ngày làm việc hay những giờ học căng thẳng bằng những chuyến du lịch gần xa đang tăng lên đáng kể. Không chỉ những dịch vụ trọn gói, nơi nghỉ dưỡng xa hoa đắt đỏ mà những chuyến du lịch tự túc phù hợp với sở thích, kinh tế của cá nhân lại đang được mọi người ưa chuộng. Tuy nhiên, việc tìm kiếm các thông tin về nơi ở hay một lộ trình du lịch phù hợp còn tồn đọng một số bất cập như sau:

**Đối với những người đi du lịch tự túc:** Việc tìm kiếm các thông tin liên quan đến nơi ở và các điểm du lịch nên đi tại nơi đó mất nhiều thời gian và công sức do các thông tin thường rời rạc, không tập trung tại một chỗ nhất định. Ngoài ra, các thông tin về lộ trình du lịch thường được cá nhân đăng tải tự do trên các nền tảng mà chưa được kiểm chứng liệu người đó đã thực sự đến địa điểm đó hay chưa, điều này gây khó khăn trong việc phán đoán tính xác thực của thông tin.

**Đối với chủ nhà:** Với những gia đình có nhiều phòng trống không sử dụng và muốn tận dụng không gian đó để kiếm thêm thu nhập. Họ cần một nơi để đăng tải các thông tin về nơi ở của mình để làm sao tiếp cận nhanh nhất đến với người dùng.

Nếu giải quyết được những vấn đề trên thì những người đi du lịch tự túc sẽ dễ dàng hơn trong việc tìm kiếm, lên kế hoạch cho một chuyến đi. Đồng thời, chủ nhà cũng dễ dàng mang hình ảnh homestay của mình phổ biến đến với mọi người.

## 1.2 Mục tiêu và phạm vi đề tài

Từ những vấn đề nêu ở phần 1.1, em nhận thấy rằng nhu cầu có một hệ thống kết nối chủ nhà và người du lịch, đồng thời là nơi những người du lịch tự túc tìm kiếm hay chia sẻ thông tin về chính chuyến đi của mình là cần thiết. Hệ thống sẽ tập trung giải quyết hai nhu cầu chính tương ứng với hai đối tượng người dùng như sau:

**Đối với người du lịch tự túc:** Hệ thống sẽ là nơi tập trung tổng hợp thông tin về các homestay, những cơ chế để giúp họ tìm và thuê được nơi nghỉ phù hợp với các tiêu chí của bản thân. Ngoài ra, họ cũng có thể chia sẻ chuyến đi của mình một cách chi tiết cho những người đi sau tham khảo.

**Đối với các chủ nhà:** Hệ thống sẽ hỗ trợ cho các chủ nhà việc đăng tải thông tin nơi ở của mình và đưa tới những người du lịch. Đồng thời cung cấp các cơ chế để chủ nhà quản lý những lượt thuê phòng tại homestay của mình.

Ngoài ra, hệ thống còn đóng vai trò trung gian giữa chủ nhà và người dùng, đảm bảo thu tiền khách thuê homestay và trả tiền cho chủ nhà. Những cơ chế giúp người dùng để lại nhận xét, đánh giá cho homestay và phản hồi của chủ nhà đối với những đánh giá đó.

### 1.3 Định hướng giải pháp

Từ việc xác định rõ nhiệm vụ cần giải quyết ở phần 1.2, giải pháp em đặt ra là xây dựng một hệ thống Web tích hợp dành cho cả người du lịch và chủ homestay. Hệ thống sẽ phát triển tách biệt riêng phần Frontend và phần Backend. Trong đó phía Backend sẽ cung cấp API thực hiện các chức năng chính như: tìm kiếm thông tin homestay, tìm kiếm phòng nghỉ phù hợp với tiêu chí của người dùng, tìm kiếm các lịch trình được review trước đó... Frontend sẽ bao gồm 2 trang web, một trang web dành cho chủ homestay và người có nhu cầu tìm kiếm thông tin về homestay, lịch trình du lịch và một trang web dành cho quản trị viên hệ thống. Nội dung cụ thể của hai trang web này như sau:

**Tại trang web dành cho người dùng:** mỗi người dùng khi đăng ký một tài khoản trên hệ thống sẽ có thể đảm nhiệm 1 vai trò (người dùng hoặc chủ nhà) hoặc cả 2 vai trò (người dùng và chủ nhà).

Khi ở trong vai trò người dùng: người dùng có thể tìm kiếm homestay theo các tiêu chí phù hợp và vào các trang hiển thị thông tin chi tiết của từng homestay. Khi đã tìm cho mình được homestay và phòng phù hợp, người dùng có thể tiến hành đặt phòng và thanh toán giá trị đơn đặt phòng. Sau khi đặt phòng và trải nghiệm thực tế, người dùng có thể đánh giá, cho điểm và bình luận đóng góp ý kiến cho homestay. Ngoài ra, hệ thống cũng cung cấp chức năng để người dùng có thể đăng tải, chia sẻ chuyến đi của mình cho mọi người.

Khi ở trong vai trò chủ nhà: hệ thống cung cấp chức năng thêm/sửa thông tin của một homestay (như là địa điểm, các loại phòng, giá phòng...) và tắt trạng thái hoạt động của homestay khi cần thiết. Thay đổi trạng thái của các đơn đặt phòng được gửi đến từ người dùng.

**Tại trang web dành cho quản trị viên hệ thống:** hệ thống cung cấp quyền quản lý các nội dung được đăng tải trên trang web.

### 1.4 Bố cục đồ án

Sau khi xác định được vấn đề, mục tiêu, phạm vi đề tài và định hướng giải pháp, phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Trong Chương 2, em sẽ trình bày chi tiết về quá trình phân tích thiết kế hệ thống. Quá trình này bao gồm việc khảo sát hiện trạng thực tế từ đó làm rõ các tác nhân

và các chức năng chính của hệ thống thông qua các biểu đồ use case tổng quan và biểu đồ use case phân rã.

Trong Chương 3 em sẽ trình bày các công nghệ đã sử dụng ở từng phần Frontend, Backend. Sau đó là đưa ra những lý do tại sao bản thân lại lựa chọn những công nghệ này.

Sau khi xác định được những chức năng cụ thể và những công nghệ được sử dụng, Chương 4 trình bày về cách kết hợp những công nghệ đã trình bày ở chương 3 để tạo nên kiến trúc phần cho toàn bộ hệ thống. Phần Backend và Frontend sẽ được phát triển độc lập và giao tiếp với nhau thông qua API. Ngoài ra, chương này cũng sẽ trình bày các vấn đề về công việc thiết kế như là thiết kế giao diện, cơ sở dữ liệu và một vài công việc liên quan khác.

Trong Chương 5, em sẽ trình bày các giải pháp và những nội dung đóng góp mà mình thấy tâm đắc nhất trong suốt quá trình làm ĐATN.

Cuối cùng, Chương 6 là tổng hợp lại quá trình phát triển ứng dụng, những kết quả đạt được, những điều chưa làm được, những bài học kinh nghiệm cần rút ra. Nêu ra định hướng phát triển của ứng dụng trong tương lai, bao gồm việc hoàn thiện các tính năng hiện tại, phát triển các tính năng mới và đưa hệ thống đi vào sử dụng thực tế trong cuộc sống.

## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Trước khi tiến hành xây dựng và phát triển hệ thống, em đã tiến hành khảo sát hiện trạng của các hệ thống đặt phòng đã có, các ứng dụng tương tự và lấy ý kiến từ chính những người có khả năng trở thành người dùng cuối. Từ đó rút ra những chức năng cơ bản cần thiết phải có cho trang web.

### 2.1 Khảo sát hiện trạng

Hiện nay có rất nhiều trang web và ứng dụng liên quan đến việc hỗ trợ cho người dùng tìm kiếm và đặt phòng. Em đã khảo sát những trang web tương tự, trong đó tiêu biểu phải kể đến Booking, Agoda và Airbnb. Chi tiết kết quả khảo sát được so sánh ở **Bảng 2.1**.

	<u>Booking</u>	<u>Agoda</u>	<u>Airbnb</u>
Đối tượng	Toàn cầu	Toàn cầu	Toàn cầu
Tìm kiếm	Có	Có	Có
Bình luận, đánh giá	Có	Có	Có
Nhắn tin	Phải đặt phòng xong mới được nhắn tin trực tiếp với chủ nhà	Tin nhắn từ box chat không được gửi trực tiếp mà gửi qua mail	Có
Review lịch trình, các địa điểm du lịch	Review của hệ thống chứ không phải của người dùng thực tế	Không	Không
Biết đầy đủ thông tin (địa chỉ, giá..)	Có	Có	Phải đặt phòng xong thì mới được biết địa chỉ chính xác

**Bảng 2.1:** So sánh các hệ thống đặt phòng

Từ **Bảng 2.1**, em thấy rằng các hệ thống đặt phòng tuy đã có khá nhiều, ngoài các hệ thống nêu trên là những hệ thống được nhiều người biết tới thông qua khảo sát thì việc tìm kiếm thông tin liên quan trên các nền tảng mạng xã hội như là Facebook cũng không còn xa lạ. Tuy nhiên các hệ thống này đều hướng tới lượng người dùng toàn cầu với đa dạng các công năng khác nhau ngoài tìm kiếm và đặt phòng. Ngoài ra, có 2 trang web khác hướng đến đối tượng người Việt Nam là Luxstay và Ivivu. Tuy nhiên, 2 trang này vẫn tồn đọng nhược điểm như là: chỉ có boxchat của hệ thống (Luxstay), có review nhưng là review của hệ thống chứ không phải của người dùng thực tế hay cần phải có tài khoản mới xem được giá (Ivivu). Từ đó, em nêu ra các nhóm chức năng cần có trong hệ thống:

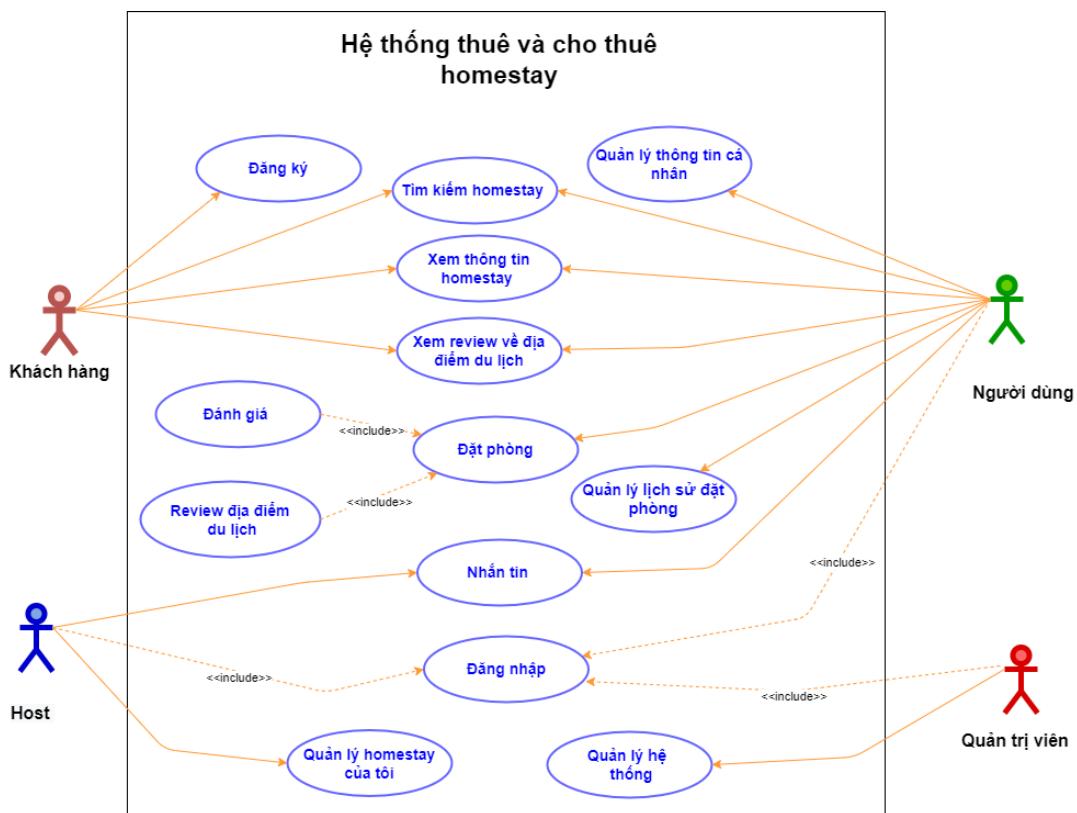
- Nhóm chức năng đăng ký, đăng nhập.
- Nhóm chức năng xem thông tin: xem thông tin homestay, thông tin phòng, thông tin đơn đặt phòng, các bài review điểm đến, bình luận và đánh giá, thông tin cá nhân...
- Nhóm chức năng quản lý thông tin: quản lý thông tin cá nhân, quản lý danh sách người dùng, quản lý danh mục homestay, quản lý danh sách đơn đặt phòng...
- Nhóm chức năng đặt phòng, thanh toán, tiếp nhận, xử lý đơn đặt phòng.
- Nhóm chức năng phản hồi: nhận xét và đánh giá phòng, review điểm đến.

## 2.2 Tổng quan chức năng

Phần 2.2 này em sẽ tóm tắt các chức năng của phần mềm. Còn phần đặc tả chi tiết em sẽ trình bày trong phần 2.3.

### 2.2.1 Biểu đồ use case tổng quát

Tổng quan về hệ thống được mô tả trong **Hình 2.1**. Hệ thống gồm có 4 tác nhân: Khách hàng, Người dùng (Khách hàng đăng ký tài khoản để trở thành Người dùng), Host (Người dùng đăng tải thông tin về homestay của mình để trở thành Host), và Quản trị viên. Ứng với mỗi tác nhân là những vai trò, chức năng cụ thể trong hệ thống.



**Hình 2.1:** Biểu đồ use case tổng quát

**Khách hàng:** có thể tạo tài khoản cá nhân, xem và tìm kiếm homestay phù hợp theo nhiều cách khác nhau, đọc những review về các địa điểm du lịch được người dùng khác chia sẻ.

**Người dùng:** có thể đặt phòng, quản lý các đơn đặt phòng của mình, đánh giá các homestay mình từng lưu trú và viết review lịch trình du lịch của mình cho người dùng khác tham khảo.

**Host:** có thể đăng tải, quản lý thông tin homestay, quản lý các đơn đặt phòng trong homestay của mình.

**Quản trị viên:** sẽ có quyền quản lý các tài khoản của người dùng, các thông tin được đăng tải trên hệ thống.

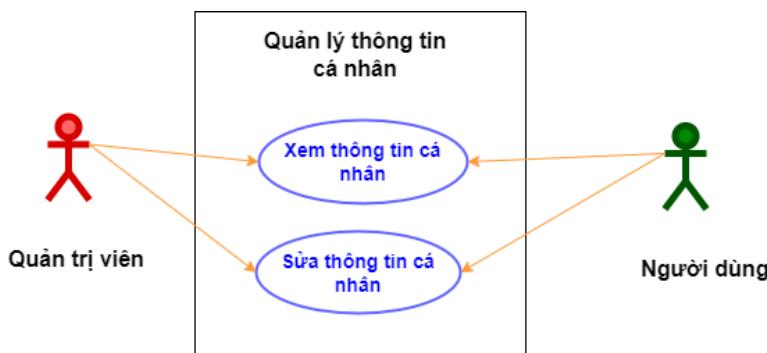
Ở hệ thống quản lý thì 3 tác nhân là Quản trị viên, Người dùng và Host cần đăng nhập để có thể sử dụng được các chức năng của hệ thống. Host và Người dùng chỉ có quyền quản lý thông tin homestay và đơn đặt phòng của mình.

Bên trên là những mô tả tổng quan về hệ thống bao gồm các tác nhân và các use case chính. Phần sau em sẽ trình bày chi tiết các use case mức cao bằng các biểu đồ use case phân rã.

### 2.2.2 Biểu đồ use case phân rã

#### a, Biểu đồ use case phân rã “Quản lý thông tin cá nhân”

Use case “Quản lý thông tin cá nhân” được mô tả thông qua **Hình 2.2** với sự tham gia của 2 tác nhân chính là Quản trị viên và Người dùng. Use case này bao gồm 2 use case con.

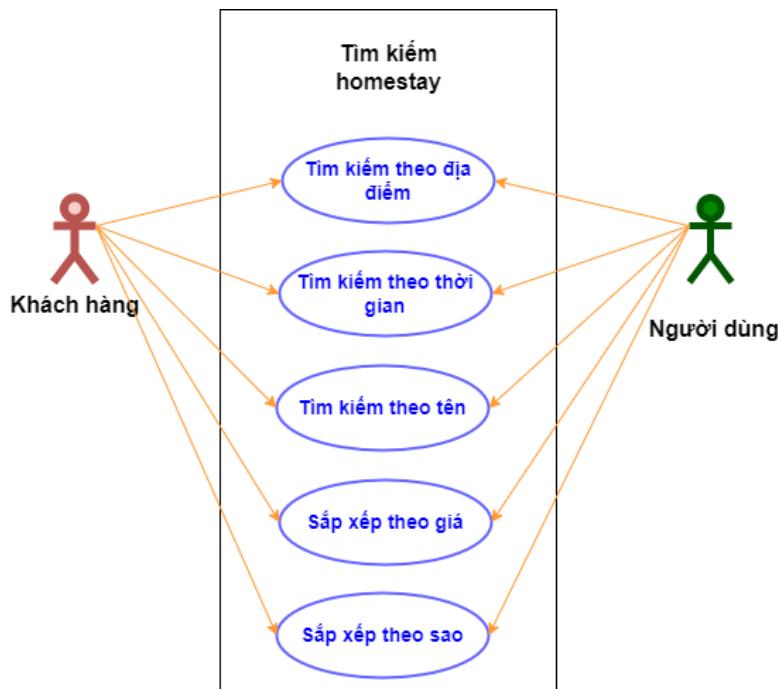


**Hình 2.2:** Biểu đồ use case phân rã “Quản lý thông tin cá nhân”

Ngoài những thông tin cố định, Quản trị viên và Người dùng có thể thay đổi thông tin cá nhân của mình như tên, giới tính, số điện thoại, ảnh đại diện.

### b, Biểu đồ use case phân rã “Tìm kiếm homestay”

Use case “Tìm kiếm homestay” được thể hiện thông qua **Hình 2.3** với sự tham gia của 2 tác nhân chính là Khách hàng và Người dùng. Use case này bao gồm 5 use case con.

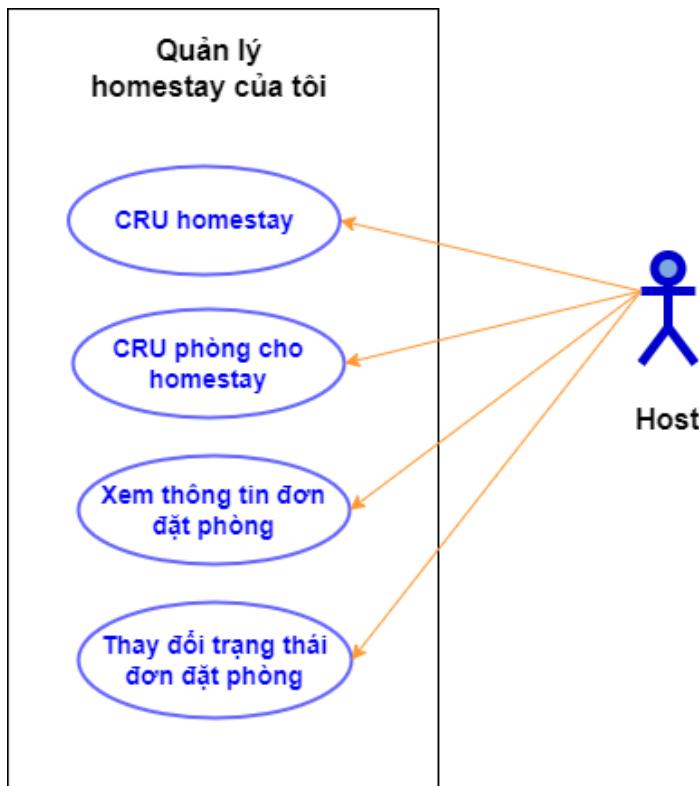


**Hình 2.3:** Biểu đồ use case phân rã “Tìm kiếm homestay”

Để tìm kiếm thông tin các homestay còn phòng trống, người dùng có 5 cách lần lượt là “Tìm kiếm theo địa điểm” như là Hà Nội, Ba Vì.., “Tìm kiếm theo thời gian” như là thời gian đi-thời gian về, “Tìm kiếm theo tên” của homestay, “Sắp xếp theo giá” và “Sắp xếp theo sao” của các homestay tìm được theo trình tự từ thấp đến cao hoặc từ cao xuống thấp (với sao là trung bình cộng số sao được đánh giá từ những người sử dụng trước đó).

### c, Biểu đồ use case phân rã “Quản lý homestay của tôi”

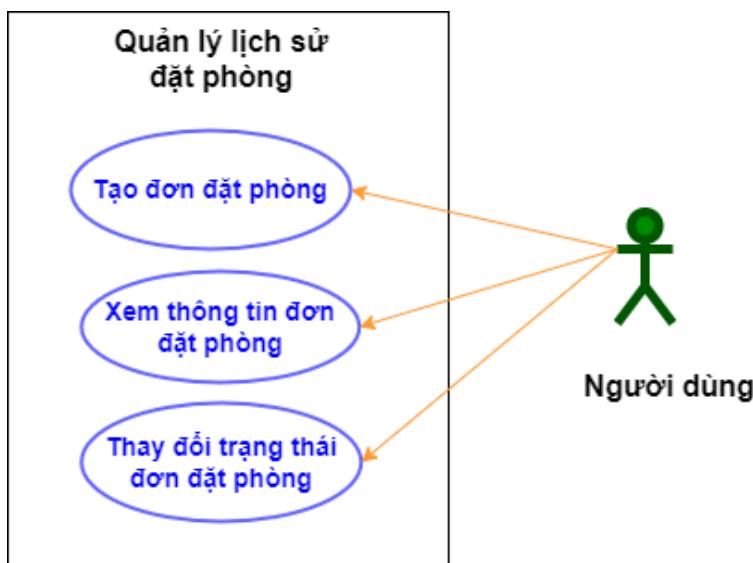
Use case “Quản lý thông tin homestay của tôi” được thể hiện thông qua **Hình 2.4** với sự tham gia của 1 tác nhân chính là Host. Use case này bao gồm 4 use case con. Host có thể xem, thêm, sửa thông tin về homestay, các phòng trong homestay và tắt trạng thái hoạt động của homestay hay các phòng của homestay khi cần thiết. Hơn nữa Host có thể xem, thay đổi trạng thái các đơn đặt phòng được gửi từ người dùng (chấp nhận hoặc từ chối).



**Hình 2.4:** Biểu đồ use case phân rã “Quản lý homestay của tôi”

#### d, Biểu đồ use case phân rã “Quản lý lịch sử đặt phòng”

Use case “Quản lý thông tin lịch sử đặt phòng” được thể hiện thông qua **Hình 2.5** với sự tham gia của 1 tác nhân chính là Người dùng. Use case này bao gồm 3 use case con.

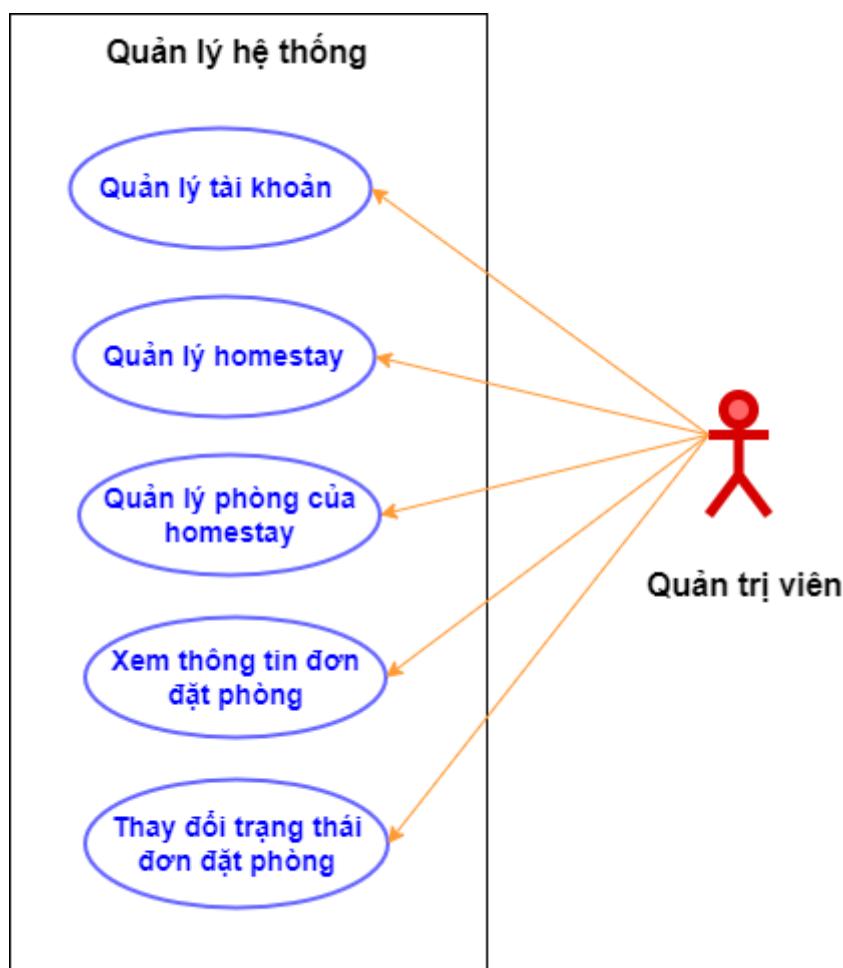


**Hình 2.5:** Biểu đồ use case phân rã ‘Quản lý lịch sử đặt phòng’

Người dùng sau khi tìm được homestay/phòng ưng ý có thể tiến hành tạo đơn đặt phòng. Ngoài ra, sau khi đặt, nếu người dùng không còn mong muốn thuê phòng nữa thì có thể thay đổi trạng thái đơn thành hủy (chỉ được hủy trước khi Host duyệt đơn đặt phòng).

#### e, Biểu đồ use case phân rã “Quản lý hệ thống”

Use case “Quản lý hệ thống” được thể hiện thông qua **Hình 2.6** với sự tham gia của tác nhân chính là Quản trị viên. Use case này bao gồm các thao tác quản lý là thao tác thêm/sửa/thay đổi trạng thái hoạt động của các đối tượng và bao gồm 5 use case con.



**Hình 2.6:** Biểu đồ use case phân rã “Quản lý hệ thống”

## 2.3 Đặc tả chức năng

Tiếp theo em sẽ trình bày chi tiết về các use case. Trong phần này, em xin phép không trình bày tất cả các chức năng như là đăng ký tài khoản, thêm mới người dùng, thêm mới phòng cho homestay, sửa thông tin người dùng, sửa thông tin phòng... mà chỉ trình bày một số chức năng chính của hệ thống.

### 2.3.1 Đặc tả use case "Tìm kiếm homestay"

Chức năng “Tìm kiếm homestay” dành cho những người du lịch đã xác định được mình sẽ đến địa điểm nào trong khoảng thời gian nào... Chi tiết được đặc tả ở **Bảng 2.2**.

Tên	Tìm kiếm homestay		
Tác nhân	Người dùng (Người dùng, Khách hàng)		
Mô tả	Cho phép người dùng tìm kiếm homestay theo ý muốn		
Tiền điều kiện	Không		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Đến chức năng tìm kiếm
	2	Người dùng	Lựa chọn một hoặc nhiều điều kiện rồi ấn nút “Tìm kiếm”
	3	Hệ thống	Hiển thị danh sách các homestay phù hợp dựa theo các tiêu chí mà người dùng đã chọn
	4	Người dùng	Chọn homestay
	5	Hệ thống	Đưa người dùng đi đến trang chi tiết các thông tin của homestay đó
Luồng sự kiện rẽ nhánh	2a	Người dùng	Không chọn điều kiện nào mà ấn “Tìm kiếm”
	3a	Hệ thống	Vẫn hiển thị tất cả các homestay có trong hệ thống
	5a	Hệ thống	Hiển thị ô nhập ngày check-in ngày check-out (điều kiện bắt buộc) và quay lại bước 2
Hậu điều kiện	Không		

**Bảng 2.2:** Đặc tả use case “Tìm kiếm homestay”

### 2.3.2 ĐẶC TẢ USE CASE "ĐẶT PHÒNG"

Chức năng “Đặt phòng” dành cho những người du lịch đã tìm được cho mình nơi ở phù hợp. Chi tiết được đặc tả ở **Bảng 2.3**.

Tên	Đặt phòng		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng đặt phòng		
Tiền điều kiện	Không		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Vào trang chi tiết thông tin của homestay, chọn phòng, số lượng phòng mong muốn
	2	Hệ thống	Hiển thị thông tin về số lượng phòng và tổng số tiền cần phải thanh toán
	3	Người dùng	Nhấn “Tôi sẽ đặt”
	4	Hệ thống	Hiển thị khung gồm các thông tin liên quan đến người dùng và đơn đặt hàng
	5	Người dùng	Xem và sửa lại các thông tin nếu cần thiết rồi nhấn “Đặt phòng”
	6	Hệ thống	Đưa người dùng đi đến trang lựa chọn phương thức thanh toán
	7	Người dùng	Lựa chọn phương thức thanh toán phù hợp
	8	Hệ thống	Hiển thị ô chọn và nhập thông tin
	9	Người dùng	Chọn và nhập thông tin phù hợp rồi ấn “Thanh toán”
	10	Hệ thống	Thông báo đặt phòng thành công và đưa người dùng về trang “Lịch sử đặt phòng”
Luồng sự kiện rẽ nhánh	10a	Hệ thống	Hiển thị thông báo thông tin bị sai và quay lại bước 9
Hậu điều kiện	Thông tin đơn đặt phòng được lưu vào cơ sở dữ liệu của hệ thống		

**Bảng 2.3:** Đặc tả use case “Đặt phòng”

### 2.3.3 ĐẶC TẢ USE CASE "THAY ĐỔI TRẠNG THÁI ĐƠN ĐẶT PHÒNG"

Chức năng này giúp Host quản lý các đơn đặt phòng trong homestay của mình. Chi tiết được đặc tả ở **Bảng 2.4**.

Tên	Thay đổi trạng thái đơn đặt phòng		
Tác nhân	Host		
Mô tả	Cho phép Host thay đổi trạng thái đơn đặt phòng của Người dùng		
Tiền điều kiện	Không		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Host	Vào trang quản lý đơn đặt phòng
	2	Hệ thống	Hiển thị danh sách các đơn đặt phòng của homestay
	3	Host	Nhấn vào icon xem chi tiết thông tin đơn đặt phòng
	4	Hệ thống	Đưa Host đến trang chi tiết
	5	Host	Chọn trạng thái muốn thay đổi (chấp nhận hoặc từ chối)
	6	Hệ thống	Hiển thị hộp thoại yêu cầu xác nhận thay đổi
	7	Host	Chọn “Đồng ý”
Luồng sự kiện rẽ nhánh	8	Hệ thống	Hiển thị thông báo cập nhật đơn hàng thành công
	7a	Host	Chọn “Không”
Hậu điều kiện	8a	Hệ thống	Tắt hộp thoại yêu cầu xác nhận
	Trạng thái mới của đơn đặt phòng được cập nhật lại vào CSDL của hệ thống		

**Bảng 2.4:** Đặc tả use case “Thay đổi trạng thái đơn đặt phòng”

### 2.3.4 ĐẶC TẢ USE CASE "ĐÁNH GIÁ HOMESTAY"

Chức năng này giúp người dùng để lại ý kiến cá nhân khi lưu trú tại homestay.

Tên	Đánh giá homestay		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng đánh giá homestay mình đã lưu trú		
Tiền điều kiện	Đơn đặt phòng ở trạng thái đã xác nhận và người dùng đã lưu trú thành công (thời điểm muốn đánh giá phải lớn hơn hoặc bằng thời gian check in)		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn mục đánh giá ở các đơn đặt phòng thỏa mãn điều kiện
	2	Hệ thống	Hiển thị biểu mẫu nhập thông tin đánh giá
	3	Người dùng	Nhập thông tin đánh giá cần thiết vào biểu mẫu và nhấn nút “Đăng bài”
	4	Hệ thống	Thông báo thêm đánh giá thành công
	5	Hệ thống	Cập nhật thông tin của bài đánh giá vào CSDL
Luồng sự kiện rẽ nhánh	3a	Người dùng	Không nhập gì hoặc nhập thiếu thông tin
	4a	Hệ thống	Hiển thị yêu cầu người dùng nhập thông tin. Quay lại bước 3 của luồng sự kiện
Hậu điều kiện	Dữ liệu bài đánh giá được lưu vào CSDL của hệ thống		

**Bảng 2.5:** Đặc tả use case “Đánh giá homestay”

### 2.3.5 Đặc tả use case "Review địa điểm du lịch"

Chức năng “Review địa điểm du lịch” dành cho những người du lịch muốn chia sẻ, đăng tải lịch trình du lịch của mình cho những người khác tham khảo. Chi tiết được đặc tả ở **Bảng 2.6**.

Tên	Review địa điểm du lịch		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng chia sẻ/review về chuyến đi của mình		
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và đã từng lưu trú tại địa điểm đó		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn “Thêm review của bạn” tại trang Review điểm đến
	2	Hệ thống	Hiển thị khung điền thông tin
	3	Người dùng	Điền đầy đủ thông tin trong khung có sẵn và nhấn “Đăng bài”
	4	Hệ thống	Hiển thị thông báo thành công và đưa người dùng đến trang hiển thị các review của địa điểm du lịch đó
Luồng sự kiện rẽ nhánh	2a	Hệ thống	Hiển thị thông báo “Bạn chưa từng đến địa điểm nào” nếu người dùng chưa từng đặt phòng trên hệ thống
	2b	Hệ thống	Hiển thị thông báo “Bạn chưa đăng nhập” nếu người dùng chưa đăng nhập vào hệ thống
	3c	Người dùng	Ấn “Đăng bài” khi chưa điền đủ thông tin cần thiết
	4c	Hệ thống	Hiển thị thông báo để yêu cầu người dùng tiếp tục nhập thông tin và quay lại bước 3
Hậu điều kiện	Thông tin bài review được lưu vào cơ sở dữ liệu của hệ thống		

**Bảng 2.6:** Đặc tả use case “Review địa điểm du lịch”

### 2.3.6 ĐẶC TẢ USE CASE "Quản lý homestay"

Chức năng “Quản lý thông tin homestay” là chức năng quản lý thêm sửa, tắt trạng thái hoạt động của homestay. Chi tiết được đặc tả ở **Bảng 2.7**.

Tên	Quản lý homestay		
Tác nhân	Host		
Mô tả	Cho phép người dùng xem, thêm, sửa thông tin liên quan đến homestay		
Tiền điều kiện	Thành viên đã đăng nhập vào hệ thống		
Luồng xem	STT	Thực hiện bởi	Hành động
	1	Host	Chọn “Homestay của tôi” trên thanh menu
Luồng tạo	2	Hệ thống	Hiển thị thông tin về homestay
	STT	Thực hiện bởi	Hành động
	1	Host	Chọn “Homestay của tôi” trên thanh menu
	2	Hệ thống	Hiển thị thông tin về homestay
	3	Host	Nhấn nút “Tạo homestay”
	4	Hệ thống	Hiển thị form nhập thông tin homestay
	5	Host	Nhập đầy đủ thông tin về homestay rồi nhấn nút “Lưu”
	6	Hệ thống	Hiển thị thông báo tạo homestay thành công, lưu thông tin vào CSDL và chuyển về trang “Homestay của tôi”
Luồng sửa	5a	Host	Chưa nhập đầy đủ thông tin về homestay mà nhấn nút “Lưu”
	6a	Hệ thống	Hiển thị thông báo tại nơi các thông tin chưa được điền và quay lại bước 5
	STT	Thực hiện bởi	Hành động
	1	Host	Chọn “Homestay của tôi” trên thanh menu
	2	Hệ thống	Hiển thị thông tin về homestay
	3	Host	Nhấn icon “Sửa thông tin”
Luồng thay đổi trạng thái	4	Hệ thống	Hiển thị form sửa thông tin homestay
	4	Host	Chỉnh sửa thông tin homestay và ấn nút “Lưu”
	5	Hệ thống	Hiển thị thông báo sửa thành công, sửa lại thông tin trong CSDL và chuyển về trang “Homestay của tôi”
	1	Host	Chọn “Homestay của tôi” trên thanh menu
	2	Hệ thống	Hiển thị thông tin về homestay
Luồng thay đổi trạng thái	3	Host	Nhấn nút thay đổi trạng thái
	2	Hệ thống	Hiển thị thông báo “Bạn muốn dừng (mở lại) hoạt động của homestay”
	3	Host	Nhấn icon “Đồng ý”
	4	Hệ thống	Thay đổi trạng thái hoạt động của homestay

**Bảng 2.7:** Đặc tả use case “Quản lý homestay”

## 2.4 Yêu cầu phi chức năng

Bên cạnh các yêu cầu về chức năng, trang web còn đưa ra các yêu cầu phi chức năng như sau:

### Tính dễ dùng:

Để trang web có giao diện đẹp mà vẫn đơn giản, dễ dàng sử dụng, em đã có đưa ra các quy tắc để thống nhất màu sắc, vị trí, hình dáng của các phần tử trong một trang và giữa các trang với nhau. Ngoài ra em cũng bố trí các chức năng một cách hợp lý, thuận tiện theo thao tác thông thường để tránh các lỗi người dùng xuất hiện trên hệ thống.

### Khả năng tương thích:

Trang web hoạt động ổn định, tương thích với các trình duyệt hiện nay như Chrome, Cốc Cốc trên thiết bị máy tính.

### Tính bảo mật:

Thông tin quan trọng của người dùng được mã hóa trước khi gửi đi.

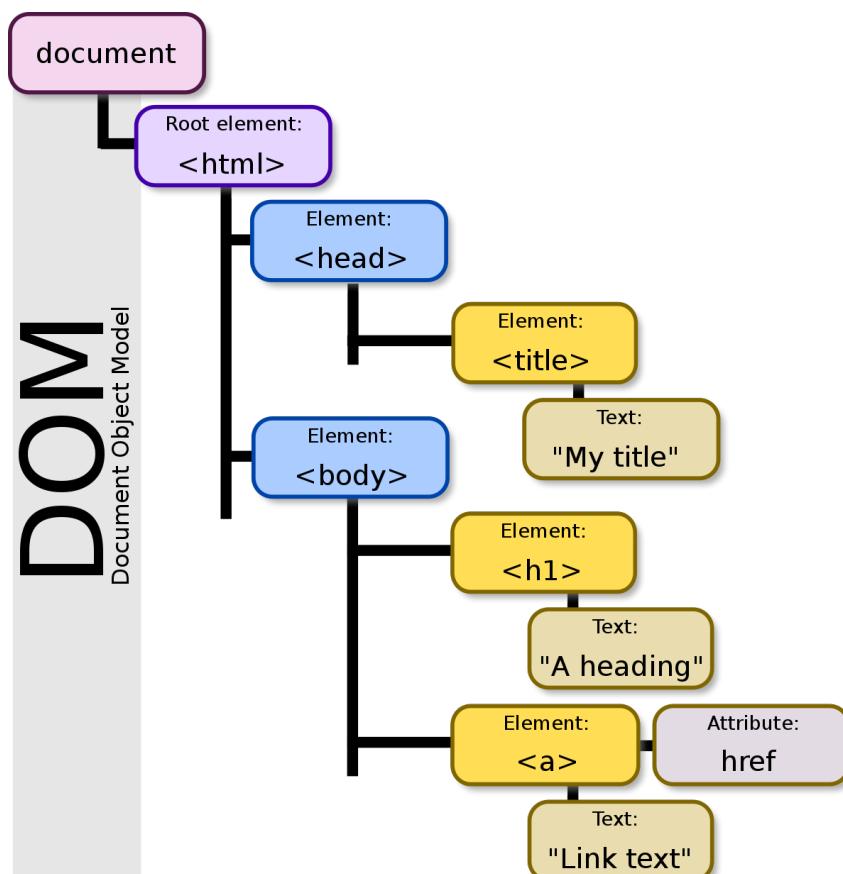
## CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

### 3.1 Công nghệ sử dụng cho phần Frontend

Ở trong phần này em sẽ trình bày về các công nghệ sử dụng cho phần Frontend của hệ thống, bao gồm cả trang cho Người dùng và trang dành cho Quản trị viên hệ thống.

#### 3.1.1 Thư viện ReactJS

**ReactJS**[1] [2] là một thư viện phát triển ứng dụng phía client được Facebook xây dựng và công bố lần đầu tiên vào năm 2011. Cho đến hiện tại, React được nhiều hệ thống lớn sử dụng tiêu biểu như là Facebook, Shopee. Trước khi có ReactJS, việc tương tác và xử lý giao diện chủ yếu sử dụng Javascript thuần (vanilla Javascript) và JQuery để xây dựng giao diện người dùng. Các tương tác này sẽ được thực hiện lên DOM (document object model), một cấu trúc trình duyệt Web sử dụng để tạo ra giao diện cho người dùng. **Hình 3.1** thể hiện cấu trúc của 1 trang web khi được render trên trình duyệt.

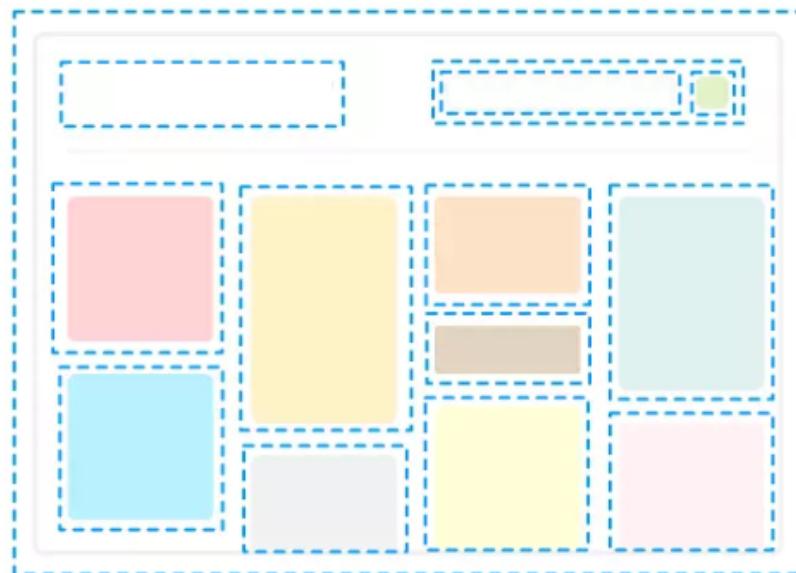


**Hình 3.1:** Cây cấu trúc DOM của một trang Web (Tham khảo từ Internet)

Thế nhưng với các trang web thông thường, với mỗi lần tương tác thay đổi khi

người dùng gửi yêu cầu dữ liệu mới thì phía server sẽ kiểm tra, tính toán và trả về toàn bộ cây DOM cho trang web. Việc thường xuyên thao tác lên DOM sẽ gây giảm hiệu suất, làm ứng dụng chậm hơn. Thực tế có một số phương pháp như sử dụng Ajax, thư viện Jquery,... đã phần nào giải quyết được vấn đề này nhưng đối với những hệ thống lớn thì việc mở rộng phát triển, bảo trì còn gặp nhiều khó khăn. Vì vậy ReactJS tạo ra với mục đích chính là cải thiện quá trình phát triển giao diện người dùng, làm cho việc tương tác với các thành phần trong ứng dụng trở nên dễ dàng hơn. Để làm được như vậy, ReactJS đã đưa ra 2 khái niệm quan trọng bao gồm : JSX và Virtual DOM.

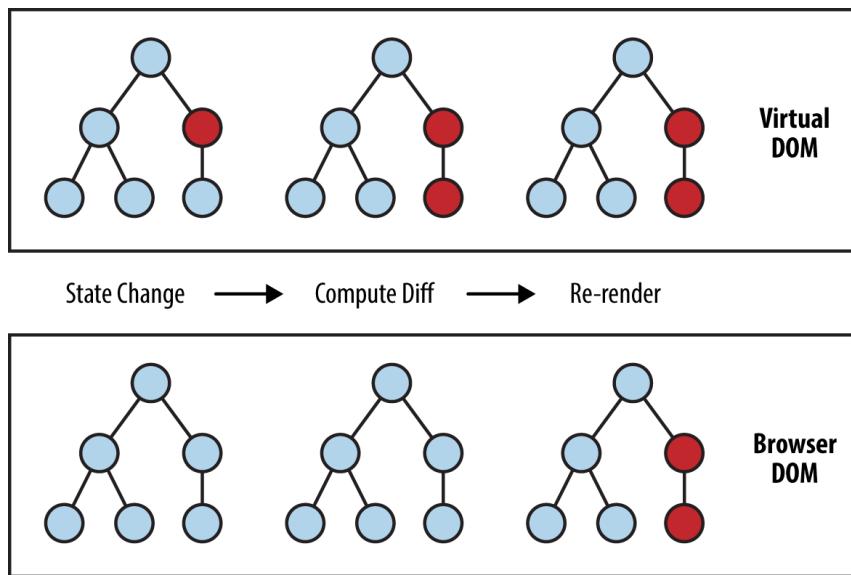
**JSX:** là một bộ cú pháp mở rộng của JavaScript, cho phép lập trình viên viết HTML trong ReactJS một cách dễ dàng. Các đoạn code JavaScript tạo ra các phần tử HTML thông qua jsx có thể tái sử dụng được gọi là Component (Thành phần). Tính năng này giúp cho tốc độ phát triển giao diện người dùng trở lên nhanh gọn hơn. Việc chia giao diện thành nhiều các component nhỏ lẻ là một cách hiệu quả giúp cho mã nguồn có thể dễ dàng mở rộng và bảo trì hơn. **Hình 3.2** mô phỏng một trang web được chia thành nhiều component. Mỗi ô trong nét đứt đầy tương đương với một component.



**Hình 3.2:** Phân chia component trong Web (Tham khảo từ Internet)

**Virtual DOM:** Virtual DOM sinh ra để giải quyết vấn đề nêu trên của cây DOM thật. ReactJS sinh ra và sử dụng DOM ảo (Virtual DOM) để nâng cao hiệu suất của nó. Các thay đổi bất kì phần tử nào được thay đổi trên trang web sẽ được tương tác lên trên DOM ảo. Khi tất cả các thay đổi được cập nhật trên DOM ảo, React dùng thuật toán so sánh diff của React để kiểm tra xem thay đổi nào cần áp dụng vào DOM gốc, và sau đó nó áp dụng thay đổi này vào chính xác chỗ cần thay đổi. Điều

này sẽ giúp tối ưu những cập nhật trên DOM thật, do đó giảm hao tốn hiệu suất của việc cập nhật DOM thật. **Hình 3.3** cho thấy cây DOM ảo và quá trình cập nhật dữ liệu.



**Hình 3.3:** DOM ảo và quá trình cập nhật dữ liệu (Tham khảo từ Internet)

Các vòng tròn đỏ đại diện cho các phần tử giao diện người dùng đã thay đổi trạng thái. React sẽ tính toán sự khác nhau giữa phiên bản trước của cây DOM ảo so với hiện tại. Khi React biết các đối tượng DOM ảo nào đã thay đổi, sẽ tiến hành cập nhật duy nhất đối tượng đó vào DOM thật. Điều này làm cho hiệu xuất nhanh hơn đáng kể khi so sánh với thao tác trực tiếp với DOM thực, làm cho React nổi bật như một thư viện JavaScript hiệu suất cao. Cho tới thời điểm hiện tại, React được rất nhiều lập trình viên ưa chuộng, chính vì vậy cộng đồng hỗ trợ và hỏi đáp rất đông đảo. Đi kèm với ReactJS cũng kéo theo rất nhiều công cụ phát triển giúp cho việc debug code một cách dễ dàng hơn. Theo thời gian mà hầu hết các chức năng và hiệu ứng cơ bản về phía giao diện người dùng đều đã có thư viện hỗ trợ.

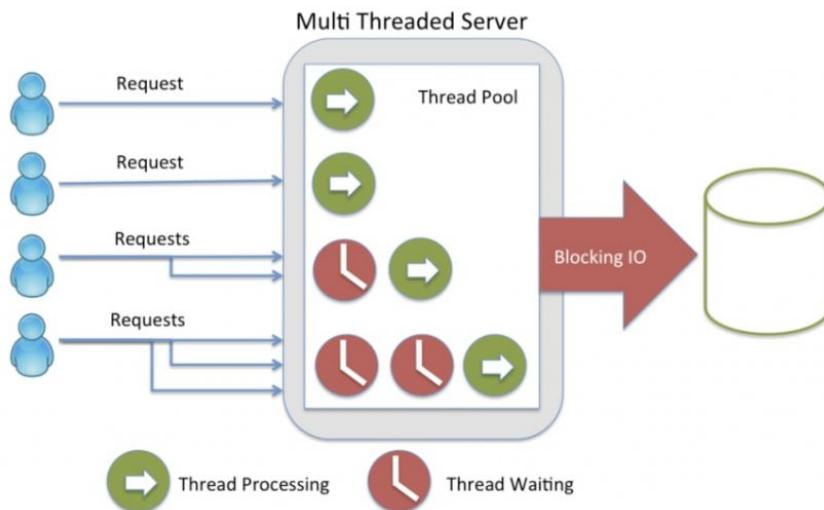
### 3.2 Công nghệ sử dụng cho phần Backend

Trong phần này em sẽ trình bày về các công nghệ đã sử dụng cho phần Backend của hệ thống.

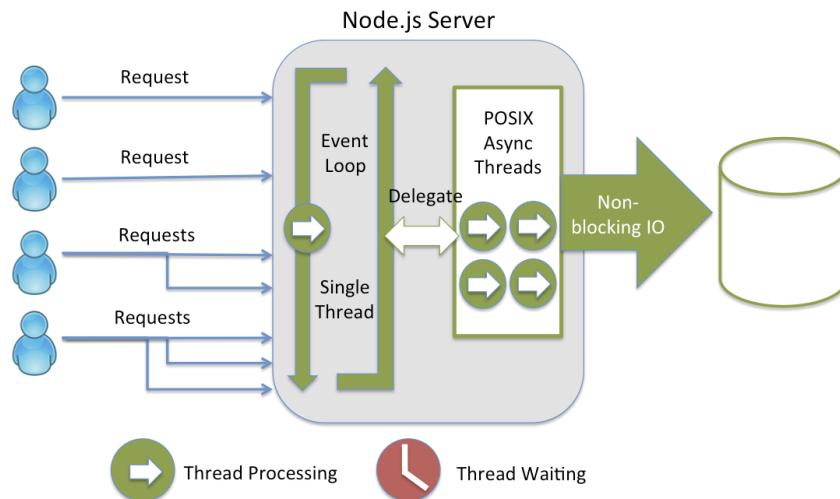
#### 3.2.1 NodeJS và Framework Express

##### NodeJS

**Nodejs[3]** là một nền tảng JavaScript mạnh mẽ được xây dựng trên JavaScript V8 Engine của Google Chrome. Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao giúp tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.



**Hình 3.4:** Xử lý luồng của web truyền thống (Tham khảo từ Internet)



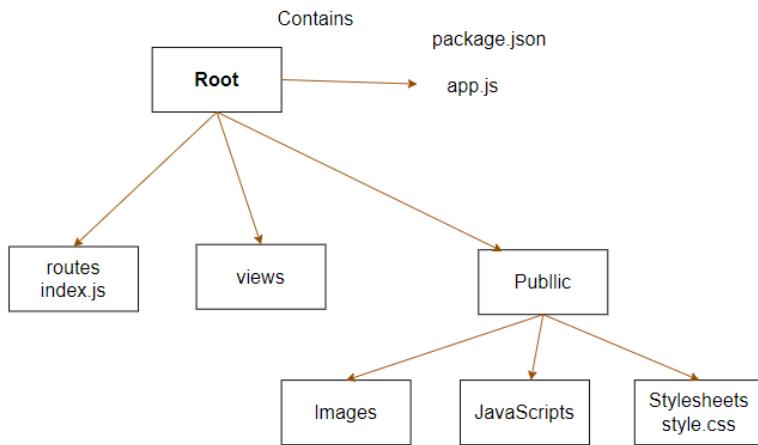
**Hình 3.5:** Xử lý luồng của NodeJS (Tham khảo từ Internet)

Với các ứng dụng web truyền thống, mỗi request sẽ tạo ra một luồng xử lý yêu cầu mới và chiếm RAM của hệ thống như **Hình 3.4**. Vì vậy dẫn đến việc tài nguyên của hệ thống sẽ không được sử dụng hiệu quả. Chính vì như vậy mà NodeJS sử dụng đơn luồng (Single-Threaded) kết hợp với non-blocking I/O để thực thi các request như **Hình 3.5** cho phép hỗ trợ hàng chục nghìn kết nối đồng thời. Đồng thời hệ thống nén của NodeJS-npm cũng là hệ thống thư viện nguồn mở lớn nhất trên thế giới.

### Framework Express

Khi nói đến các framework NodeJS phổ biến nhất thì **Express**[4] [5] là lựa chọn được nhiều người ưa chuộng hiện tại. Framework Express nhẹ, cực kì nhanh, cung cấp các tiện ích mở rộng để hỗ trợ nhiều chức năng cho nó. Express rất đơn giản để

tùy chỉnh và sử dụng theo nhu cầu. Nó cung cấp cho chúng ta đầy đủ tiện ích để có thể xây dựng một ứng dụng web hay một API. **Hình 3.6** là cấu trúc tổng quát của một ứng dụng theo framework Express.

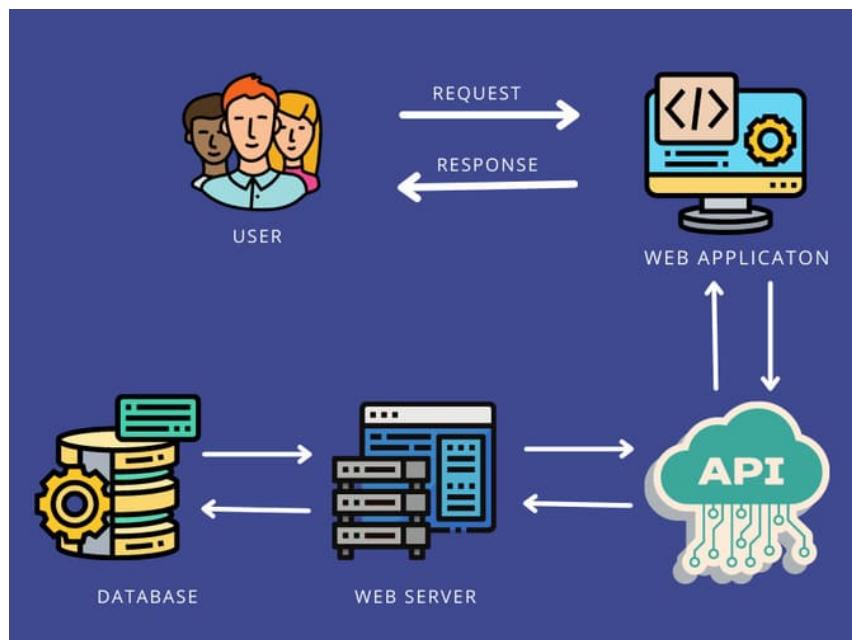


**Hình 3.6:** Cấu trúc của express

Một số ưu điểm của framework Express có thể kể đến như hỗ trợ chúng ta phát triển ứng dụng theo mô hình MVC, cho phép định nghĩa các middleware, định nghĩa rõ ràng các request methods trong router và hỗ trợ mạnh về REST API.

### 3.2.2 REST API

REST API là một tập hợp các tiêu chuẩn dùng trong việc xây dựng và thiết kế API cho web service để quản lý các Resource trở lên dễ dàng hơn. Nó có những phương thức kết nối với các ứng dụng khác nhau. Thay vì sử dụng URL cho việc xử lý thông tin, REST API sẽ dùng các HTTP method như GET, PATCH, POST, DELETE,... để xử lý dữ liệu. Mô tả trong **Hình 3.7**



**Hình 3.7:** Rest api (Tham khảo từ Internet)

Với xu hướng phát triển ứng dụng phần mềm hiện nay thì việc sử dụng chuẩn REST API để xây dựng API trở lên phổ biến hơn do REST API sử dụng ít băng thông và linh hoạt. Ngoài ra, có rất nhiều trang web sử dụng REST API để cho phép việc kết nối dữ liệu từ họ dễ dàng hơn. Và trong số đó, thì ông lớn facebook cũng đang cung cấp cho các Rest API để ứng dụng bên ngoài vẫn có thể kết nối đến với dữ liệu của họ.

### 3.2.3 MongoDB

**MongoDB**[6] là hệ quản trị cơ sở dữ liệu mã nguồn mở, là cơ sở dữ liệu kiểu NoSQL (None-Relational SQL hoặc Not-Only SQL) và hướng tài liệu. Ngoài ra, MongoDB là một cơ sở dữ liệu đa nền tảng, các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ, hoạt động trên các khái niệm Collection và Document, nó cũng là một trong những kiến trúc CSDL dùng để lưu trữ dữ liệu lớn (Big Data), hỗ trợ phân tán, dễ dàng co giãn mở rộng, tốc độ truy cập nhanh, không có ràng buộc mối quan hệ giữa các bảng.

**Bảng 3.1** so sánh tên gọi giữa CSDL quan hệ và CSDL MongoDB.

Cơ sở dữ liệu quan hệ	MongoDB
Database	Database
Table	Collection
Row	Document (BSON document)
Index	Index
Join	\$lookup
Foreign Key	Reference

**Bảng 3.1:** So sánh tên gọi giữa CSDL quan hệ và CSDL MongoDB

Mỗi Server được gọi là 1 Cluster, 1 Cluster có nhiều Cơ sở dữ liệu, mỗi cơ sở dữ liệu có nhiều Collection (bảng), mỗi Collection có nhiều Document (dòng). Document là đối tượng Json, tùy biến theo người thiết kế.

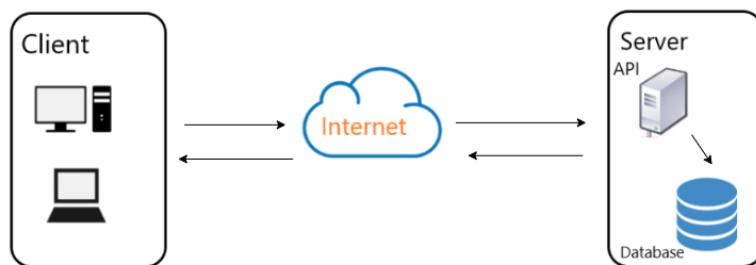
## CHƯƠNG 4. THIẾT KẾ VÀ TRIỂN KHAI ỨNG DỤNG

### 4.1 Thiết kế kiến trúc

#### 4.1.1 Lựa chọn kiến trúc phần mềm

Trong hệ thống này em chọn sử dụng mô hình Client-Server.

Mô hình Client-Server là một mô hình nổi tiếng trong mạng máy tính, được áp dụng rất rộng rãi và là mô hình của nhiều trang web hiện có. Trong mô hình này các máy tính client được đóng vai trò như một máy khách (client), chúng sẽ gửi yêu cầu (request) đến các máy chủ (server) để máy chủ xử lý những yêu cầu đó và trả kết quả về cho client. Kiến trúc tổng quan của hệ thống được mô tả trong **Hình 4.1**



**Hình 4.1:** Kiến trúc tổng quan của hệ thống

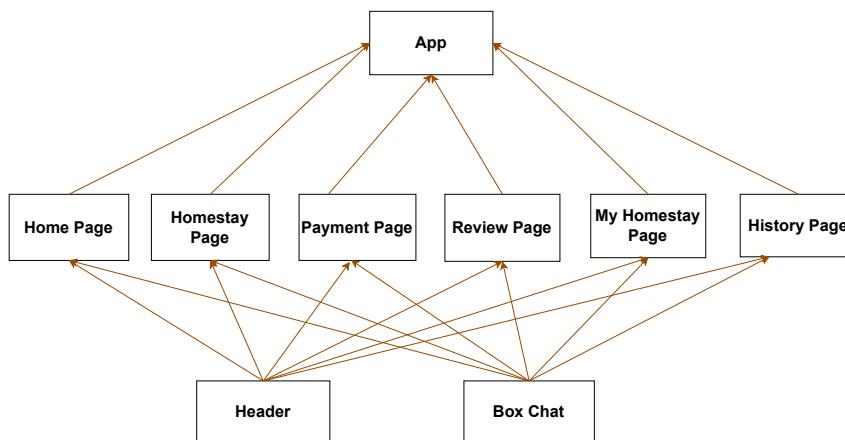
Cụ thể trong hệ thống thuê và cho thuê homestay này sẽ có 2 Client tương ứng với 2 trang web khác nhau. Một trang dành cho người dùng, một trang dành cho quản trị viên trang web. Phía server sẽ gồm các API. Trong đó Server sẽ cung cấp các API lấy dữ liệu của các homestay, các phòng, các bình luận, các bài review, thông tin người dùng,...từ CSDL cho Client.

### 4.1.2 Kiến trúc Frontend

Ở phía frontend em sử dụng thư viện React nên cấu trúc frontend em sẽ tập trung vào việc trình bày cấu trúc Component trong hệ thống. Do số lượng Component trong hệ thống là tương đối nhiều nên em sẽ chỉ trình bày những Component chính nhất.

**Trang web cho Người dùng và Host:**

**Hình 4.2** mô tả kiến trúc Frontend trang Người dùng và Host.

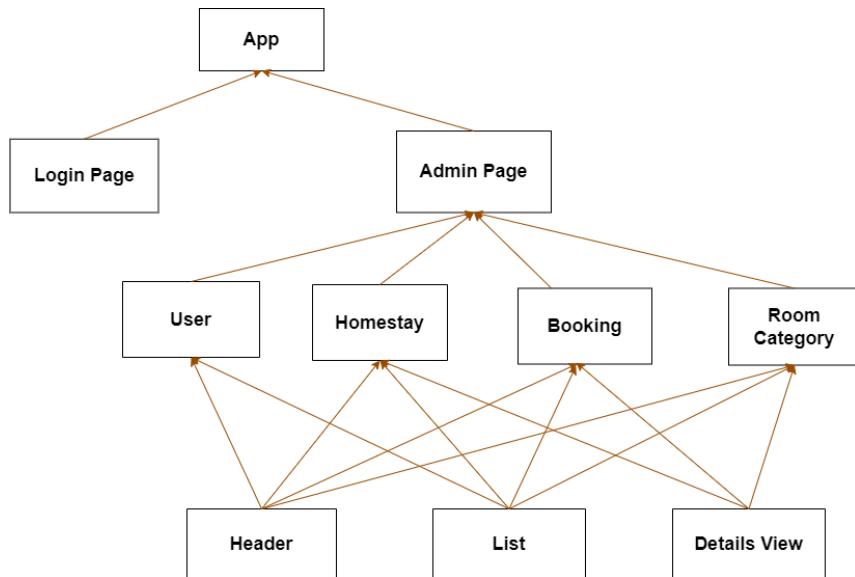


**Hình 4.2:** Kiến trúc Frontend trang Người dùng và Host

Trong đó **App** là Component tổng trong ReactJS, bao gồm 6 trang chính là : **Home Page** (trang chủ), **Homestay Page** (Danh sách homestay), **History Page** (Lịch sử đặt phòng), **Payment Page** (Chọn phương thức thanh toán), **Review Page** (Danh sách review địa điểm), **My Homestay Page** (Trang quản lý Homestay của Host). Ở đây em nhận thấy giao diện mỗi trang đều có phần **Header**, **Boxchat** nên em viết tách riêng các Component này và tái sử dụng chúng ở trong các trang trên.

**Trang web cho quản trị viên:**

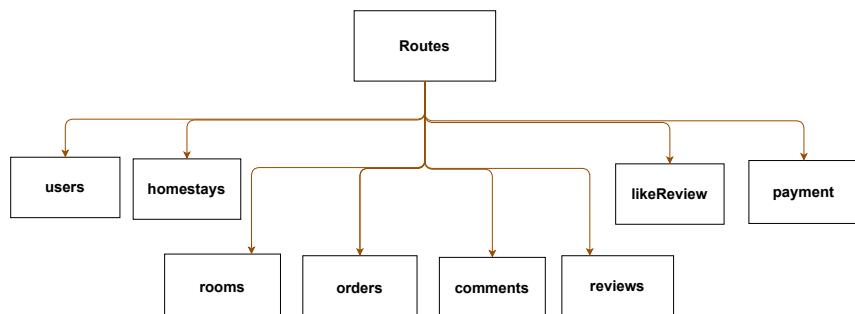
**Hình 4.3** là kiến trúc Frontend trang cho quản trị viên. Trong **App** của trang quản trị viên, em chia thành 2 trang chính : **Login Page** (Trang đăng nhập) và **Admin Page** (Trang quản trị). Trong đó, trang quản trị sẽ bao gồm các Component nhỏ hơn là **User** (Tài khoản người dùng), **Homestay**, **Booking** (Đơn đặt phòng), **Room Category** (Các loại phòng trong homestay). Và các Component dùng chung cho 4 component trên là **Header**, **List** (Xem danh sách), **Details View** (Xem chi tiết), **Update** (Chỉnh sửa).

**Hình 4.3:** Kiến trúc Frontend trang trí viên

#### 4.1.3 Kiến trúc Backend

Như đã trình bày trong phần 3.2.1, phần API em sẽ sử dụng framework Express. Ứng dụng sẽ chia thành 3 phần chính là **Controller**, **Route** và **Model**. **Model** mô tả giao diện collection hiển thị như thế nào (nó bao gồm thuộc tính và phương thức để định nghĩa kiểu dữ liệu cho các thuộc tính đó). **Route** để xác định cách mà ứng dụng phản hồi các yêu cầu của máy khách tại một endpoint cụ thể, là một URL (hoặc đường dẫn) và một phương thức request HTTP cụ thể (GET, POST, ...). Mỗi route có một hàm handler khác nhau, được thực thi khi route được khớp nhau. **Controller** là nơi xử lý tất cả các logic nghiệp vụ. Nói cách khác, Route sẽ tiếp nhận yêu cầu từ phía frontend, điều hướng tới Controller để xử lý tất cả các logic nghiệp vụ.

Do số lượng route nhiều nên em sẽ chỉ trình bày một vài route chính có trong hệ thống và mô tả chúng trong **Hình 4.4**.

**Hình 4.4:** Kiến trúc Backend

Có 8 route chính là : **users**, **homestays**, **rooms**, **orders**, **comments**, **payment**,

**reviews, likeReview.** Mỗi route sẽ có controller và model tương ứng để xử lý logic nghiệp vụ. Việc chia ra từng route với controller và model riêng biệt sẽ giúp ta quản lý các logic dễ dàng hơn. Khi hệ thống cần phát triển lớn yêu cầu những logic nghiệp vụ phức tạp hơn thì đây cũng là kiến trúc khiến cho ứng dụng dễ mở rộng và bảo trì đối với lập trình viên.

## 4.2 Thiết kế chi tiết

### 4.2.1 Thiết kế giao diện

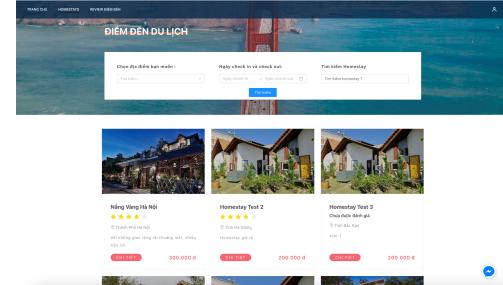
#### Thông tin màn hình

Đối với giao diện trên hệ thống, em thiết kế sao cho người dùng dễ dàng sử dụng, bố trí các nút chức năng hợp lý và phối hợp màu sắc thống nhất theo quy tắc như sau:

- Với các nút đăng ký, đăng nhập và thêm mới sẽ có nền màu xanh dương, chữ màu trắng.
- Với mỗi thao tác thành công sẽ hiển thị thông báo màu xanh lá.
- Những lỗi xảy ra do thao tác của người dùng sẽ hiện thông báo màu đỏ.
- Những ô nhập dữ liệu bắt buộc thì trước tiêu đề của ô đó sẽ có biểu tượng dấu hoa thị màu đỏ.

Giao diện Desktop hiện tại đang hoạt động ổn định với 3 kích cỡ màn hình là 1366x768, 1440x695, 1920x937. **Hình 4.5, 4.6, 4.7, 4.8** là giao diện mockup và giao diện thực tế trên màn hình Desktop ở một vài trang tiêu biểu trong hệ thống.

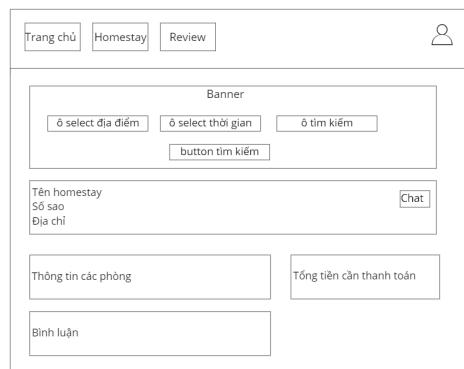
#### Trang danh sách Homestay



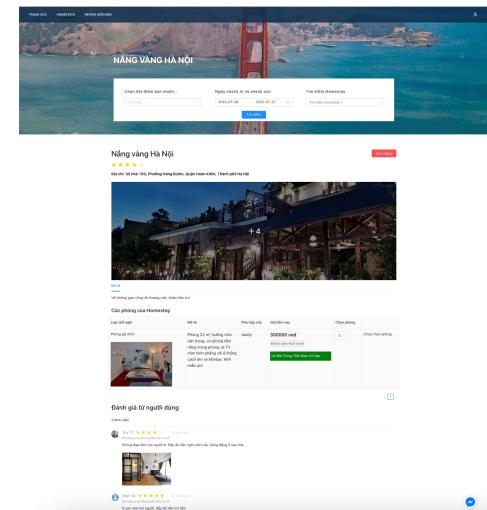
**Hình 4.6:** Giao diện thực tế Desktop

**Hình 4.5:** Giao diện mockup Desktop

#### Trang Homestay Detail



**Hình 4.7:** Giao diện mockup Desktop

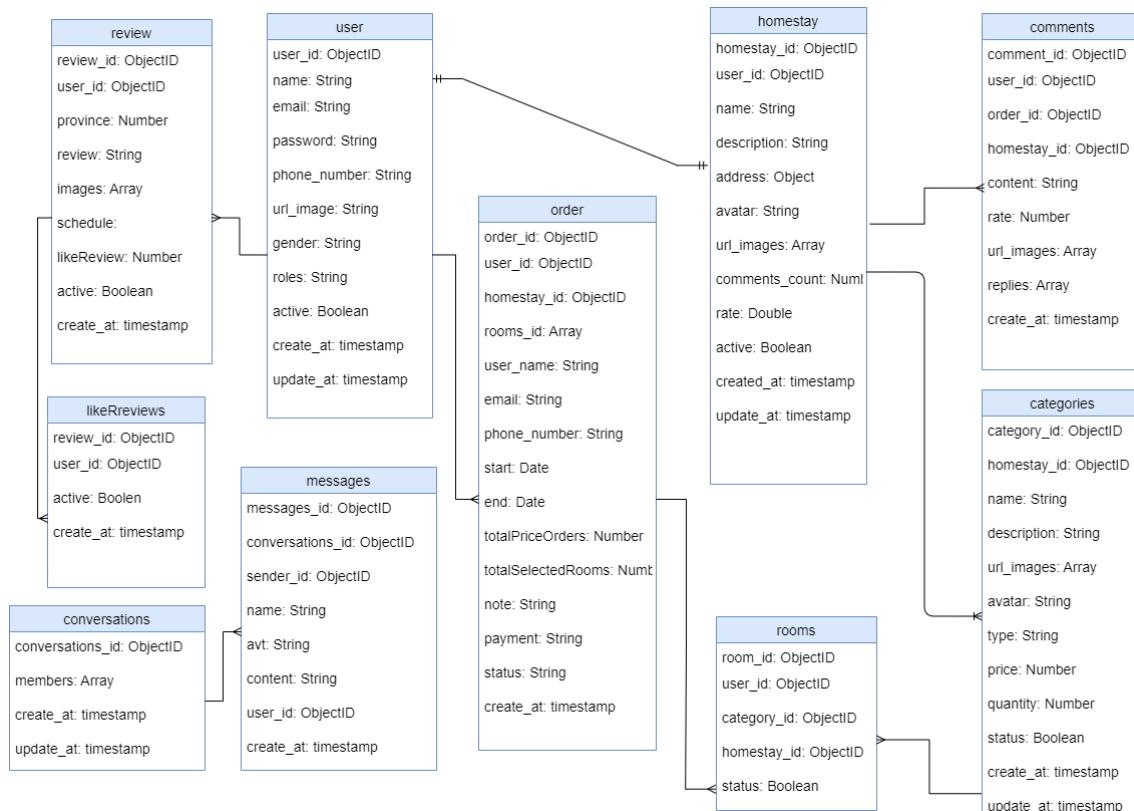


**Hình 4.8:** Giao diện thực tế Desktop

## 4.2.2 Thiết kế cơ sở dữ liệu

### a, Sơ đồ thiết kế cơ sở dữ liệu

Em thiết kế CSDL sử dụng hệ quản trị CSDL MongoDB gồm những collection như **Hình 4.9**



**Hình 4.9:** Sơ đồ thiết kế cơ sở dữ liệu

### b, Mô tả chi tiết các trường trong từng collection

**Bảng 4.1:** Collection chứa thông tin của người dùng

Thuộc tính	Mô tả
user_id	Id người dùng
name	Tên của người dùng
email	Địa chỉ email của người dùng
password	Mật khẩu khi đăng nhập
phone_number	Số điện thoại của người dùng
url_image	Địa chỉ hình ảnh đại diện
gender	Giới tính
roles	Quyền người dùng (Admin, user)
active	Trạng thái tài khoản
created_at	Thời điểm tạo tài khoản
update_at	Thời điểm chỉnh sửa thông tin tài khoản

**Bảng 4.1:** Cấu trúc collection user**Bảng 4.2:** Collection chứa thông tin homestay của người dùng

Thuộc tính	Mô tả
homestay_id	Id của homestay
user_id	Id của người dùng, có chức năng như là khóa ngoại liên kết với bảng user
name	Tên homestay
description	Mô tả của homestay
address	Địa chỉ của homestay
avatar	Đường dẫn ảnh mô tả ảnh bìa của homestay
url_image	Đường dẫn ảnh mô tả của homestay
comments_counts	Số lượng bình luận của homestay
rate	Trung bình đánh giá của người dùng cho homestay
active	Trạng thái hoạt động của homestay
created_at	Thời điểm tạo homestay
update_date	Thời điểm chỉnh sửa thông tin homestay

**Bảng 4.2:** Cấu trúc collection homestay

**Bảng 4.3:** Collection chứa thông tin của các loại phòng của homestay

Thuộc tính	Mô tả
category_id	Id của category
user_id	Id của người dùng
homestay_id	Id của homestay, có chức năng như là khóa ngoại liên kết với bảng homestay
name	Tên category
quantity	Số lượng phòng có trong category
price	Giá phòng của category
type	Loại phòng của category
description	Mô tả của category
avatar	Đường dẫn của ảnh bìa category
url_image	Đường dẫn của ảnh mô tả category
comments_count	Số lượng comment của category
rate	Số sao/số điểm của category
status	Trạng thái hoạt động của category
created_at	Thời điểm tạo category
update_at	Thời điểm thay đổi thông tin category

**Bảng 4.3:** Cấu trúc collection category**Bảng 4.4:** Collection chứa thông tin của các phòng thuộc các category của homestay

Thuộc tính	Mô tả
room_id	Id của phòng
user_id	Id người dùng
homestay_id	Id của homestay
category_id	Id của category, có chức năng như là khóa ngoại liên kết với bảng category
price	Giá phòng
status	Trạng thái hoạt động của phòng
created_at	Thời điểm tạo phòng
update_at	Thời điểm chỉnh sửa thông tin của phòng

**Bảng 4.4:** Cấu trúc collection room

**Bảng 4.5:** Collection chứa thông tin của các đơn đặt phòng

Thuộc tính	Mô tả
order_id	Id của order
order	Mảng các order (danh sách và số lượng các category đặt trong order)
rooms_id	Id của người dùng
homestay_id	Id của homestay
nameHomestay	Tên của homestay
rooms_id	Danh sách các room được đặt trong order
user_name	Tên của người dùng
email	Địa chỉ email của người dùng
phone_number	Số điện thoại của người dùng
check_in	Thời gian check in
check_out	Thời gian check out
totalPriceOrders	Tổng số tiền của đơn
totalSelectedRooms	Tổng số phòng đã đặt trong đơn đặt phòng
note	Ghi chú riêng của người dùng cho đơn đặt phòng của mình
payment	Phương thức thanh toán
status	Trạng thái của order (bao gồm: Chưa xác nhận, Đã xác nhận, Hủy, Từ chối)
created_at	Thời điểm tạo order

**Bảng 4.5:** Cấu trúc collection order**Bảng 4.20:** Collection chứa thông tin đánh giá của người dùng cho homestay

Thuộc tính	Mô tả
comment_id	Id của comment
user_id	Id người dùng
order_id	Id của đơn đặt phòng
homestay_id	Id của homestay
content	Nội dung comment
rate	Số sao/số điểm đánh giá homestay
url_image	Đường dẫn ảnh mô tả của comment
replies	Phản hồi của Host đối với comment của người dùng
created_at	Thời điểm tạo comment
update_at	Thời điểm chỉnh sửa comment

**Bảng 4.6:** Cấu trúc collection comment

**Bảng 4.7:** Collection chứa thông tin bài viết review của người dùng đối với điểm đến

Thuộc tính	Mô tả
review_id	Id bài review
user_id	Id người dùng
province	Mã số của điểm đến
content	Nội dung review
url_image	Đường dẫn ảnh mô tả của review
schedule	Lịch trình du lịch của người dùng
likeReview	Số lượt yêu thích bài review
created_at	Thời điểm tao review
update_at	Thời điểm chỉnh sửa review

**Bảng 4.7:** Cấu trúc collection review

**Bảng 4.8:** Collection chứa thông tin về like của bài review

Thuộc tính	Mô tả
likeReview_id	Id của like
review_id	Id bài review
user_id	Id người dùng
status	Trạng thái
created_at	Thời điểm like review
update_at	Thời điểm chỉnh sửa like review

**Bảng 4.8:** Cấu trúc collection likeReview

**Bảng 4.9:** Collection chứa thông tin về cuộc trò chuyện

Thuộc tính	Mô tả
conversation_id	Id cuộc hội thoại
members	Danh sách id thành viên trong cuộc hội thoại
created_at	Thời điểm gửi tin nhắn
update_at	Thời gian tin nhắn mới nhất được gửi

**Bảng 4.9:** Cấu trúc collection conversation**Bảng 4.10:** Collection chứa thông tin về tin nhắn trong cuộc trò chuyện

Thuộc tính	Mô tả
messages_id	Id của tin nhắn
conversations_id	Id cuộc hội thoại
sender_id	Id của người gửi
name	Tên người gửi
avt	Đường dẫn ảnh đại diện của người gửi
content	Nội dung tin nhắn
user_id	Id của tài khoản đang đăng nhập
created_at	Thời điểm gửi tin nhắn

**Bảng 4.10:** Cấu trúc collection message

### 4.3 Xây dựng ứng dụng

#### 4.3.1 Thư viện và công cụ sử dụng

Để hoàn thành trang web, em đã sử dụng các thư viện và công cụ hỗ trợ như ở **Bảng 4.11**

Mục đích	Công cụ	Địa chỉ URL
Cơ sở dữ liệu	MongoDB Atlas	<a href="https://www.mongodb.com">https://www.mongodb.com</a>
Cơ sở dữ liệu	Firebase Cloud Firestore (9.8.1)	<a href="https://firebase.google.com">https://firebase.google.com</a>
Frontend framework	ReactJS ( 17.0.2)	<a href="https://reactjs.org/">https://reactjs.org/</a>
Backend framework	Express ( 4.17.1 )	<a href="https://expressjs.com/">https://expressjs.com/</a>
File Storage	Cloudinary	<a href="https://cloudinary.com/">https://cloudinary.com/</a>
Thư viện ODM	Mongoose (5.12.11)[7]	<a href="https://mongoosejs.com/">https://mongoosejs.com/</a>
Thư viện frontend	Ant Design (4.15.1)[8]	<a href="https://ant.design/">https://ant.design/</a>
IDE lập trình	Visual Studio Code	<a href="https://code.visualstudio.com/">https://code.visualstudio.com/</a>

**Bảng 4.11:** Danh sách thư viện và công cụ sử dụng

#### 4.3.2 Kết quả đạt được

Hệ thống được đóng gói thành 3 phần lần lượt là Trang web cho người dùng, Trang web cho quản trị viên và các API. **Bảng 4.12** thống kê số dòng code, số gói và dung lượng của các thành phần có trong hệ thống.

Ứng dụng	Số dòng code	Số gói	Dung lượng
Trang web cho người dùng	14356	23	3.72MB
Trang web cho quản trị viên	9240	21	3.3MB
API	2635	27	87KB

**Bảng 4.12:** Các thông tin về hệ thống

#### 4.3.3 Minh họa các chức năng chính

Do các chức năng đăng ký, đăng nhập hay thêm sửa xoá là các chức năng đã quen thuộc và tương đối cơ bản nên trong phần này em sẽ minh họa 4 chức năng tìm kiếm phòng theo điều kiện, nhắn tin, đặt phòng và bình luận.

### a, Tìm kiếm phòng theo điều kiện:

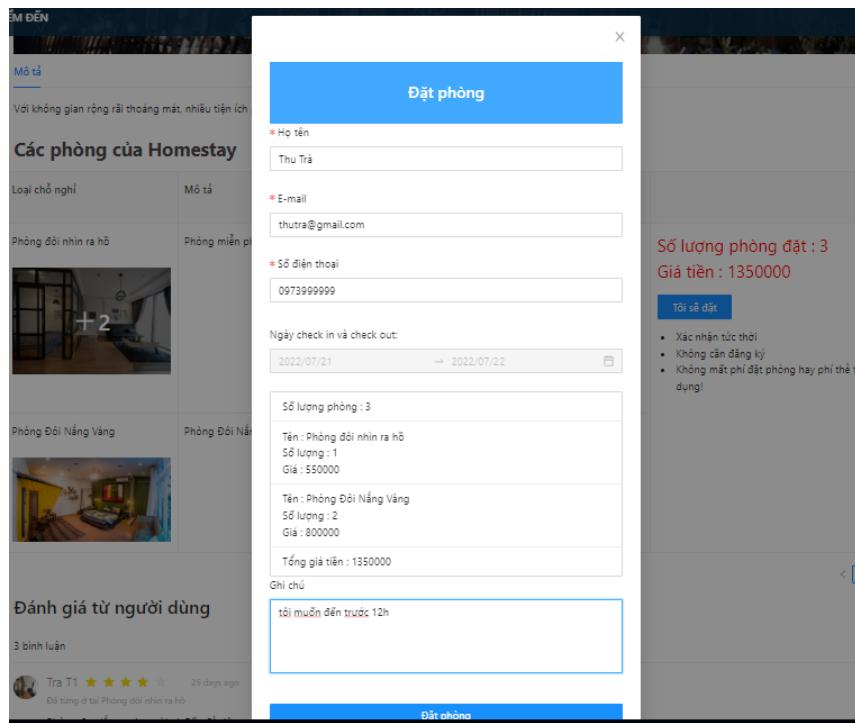
**Hình 4.10** là giao diện trang để tìm kiếm phòng trống, bao gồm thanh tìm kiếm để nhập các điều kiện, nút tìm kiếm, ô chọn kiểu sắp xếp và danh sách các homestay.

**Hình 4.10:** Giao diện trang tìm kiếm phòng

**Hình 4.11** là giao diện trang chi tiết homestay, ở đây bao gồm các thông tin liên quan đến homestay như là thông tin các phòng, các bình luận và đánh giá từ người dùng... Tại đây ta sẽ lựa chọn số lượng phòng mong muốn và hệ thống sẽ hiển thị số lượng phòng đã chọn và tổng số tiền cần thanh toán.

**Hình 4.11:** Giao diện trang lựa chọn phòng

Khi bấm vào nút “Tôi sẽ đặt”, biểu mẫu nhập thông tin hiện ra như **Hình 4.12**, người dùng có thể xem lại các thông tin liên quan đến đơn đặt phòng và chỉnh sửa thông tin cá nhân hay để lại ghi chú (nếu muốn).



**Hình 4.12:** Giao diện biểu mẫu thông tin đặt phòng

Các **Hình 4.13, 4.14** lần lượt là các trang lựa chọn hình thức thanh toán, nhập thông tin thẻ ngân hàng để tiến hành thanh toán trực tuyến.

**Hình 4.13:** Giao diện chọn phương thức thanh toán

**Hình 4.14:** Giao diện nhập thông tin thanh toán

Với các thao tác nhập sai hay nhập thiếu thông tin thì các thông báo đỏ sẽ được hiển thị như **Hình 4.15, 4.16** (đây là giao diện do chính cổng thanh toán VNPay cung cấp, em chỉ tích hợp chúng vào trang web của mình).

Phương thức thanh toán

VNPAY Thanh toán VNPay  
Chọn ngân hàng  
Xin vui lòng chọn ngân hàng!  
Ngân hàng chuyển khoản  
Xin vui lòng nhập nội dung chuyển khoản!  
Thẻ thanh toán quốc tế

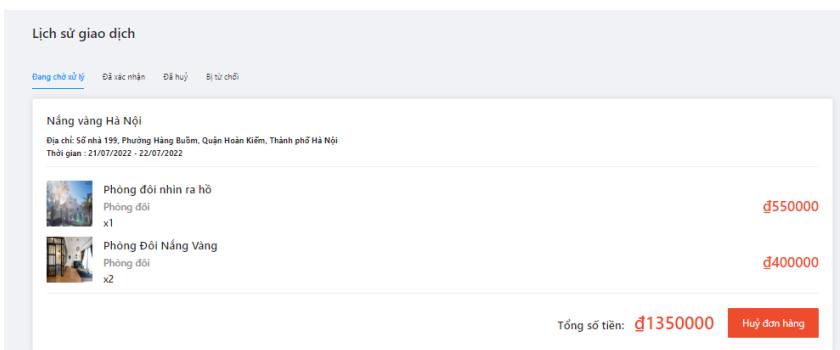
**Hình 4.15:** Giao diện yêu cầu chọn phương thức thanh toán

#### Thanh toán qua Ngân hàng NCB

Số thẻ  
Nhập số thẻ  
Bạn chưa nhập số thẻ  
Tên chủ thẻ  
Nhập tên chủ thẻ (không dấu)  
Bạn chưa nhập Tên chủ thẻ  
Ngày phát hành  
MM / YY  
Bạn chưa nhập ngày phát hành

**Hình 4.16:** Giao diện yêu cầu nhập thông tin thanh toán

Khi đặt phòng thành công, màn hình sẽ chuyển về trang Lịch sử order như **Hình 4.17.**



**Hình 4.17:** Giao diện sau khi đặt phòng thành công

### b, Quản lý đơn đặt phòng

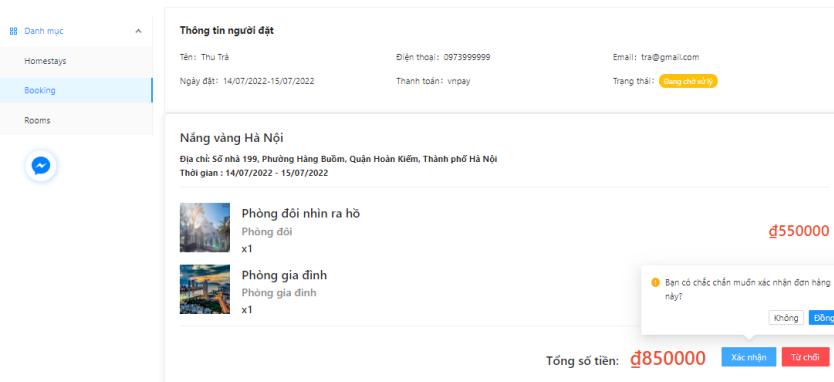
**Hình 4.18** là giao diện danh sách các đơn đặt phòng của 1 homestay. Tại đây Host có thể biết được thông tin khái quát về các đơn đặt phòng được gửi đến từ người dùng.

Danh sách order									
Mã đơn hàng	Số lượng	Tổng tiền	Thời gian	Thanh toán	Ngày tạo	Gf	Trạng thái	Thao tác	
62bc27a748beef30e402226d0	1 phòng	300000 VND	29/06/2022-30/06/2022	vnpay	29/06/2022 17:21	<span style="color: yellow;">Đang chờ xử lý</span>	<span style="color: green;">Đã xác nhận</span>	<span style="color: blue;">Xem</span>	
62bb823e10b8d1c1469213b	1 phòng	550000 VND	26/06/2022-27/06/2022	vnpay	26/06/2022 22:58	<span style="color: green;">Đã duyệt</span>	<span style="color: green;">Đã xác nhận</span>	<span style="color: blue;">Xem</span>	
62bb84c52da032604ffdec0	1 phòng	250000 VND	26/06/2022-27/06/2022	vnpay	26/06/2022 23:09	<span style="color: green;">Đã duyệt</span>	<span style="color: green;">Đã xác nhận</span>	<span style="color: blue;">Xem</span>	
62bb170eebacab1e16ec0fb	1 phòng	550000 VND	01/07/2022-02/07/2022	vnpay	28/06/2022 21:58	<span style="color: green;">Đã duyệt</span>	<span style="color: green;">Đã xác nhận</span>	<span style="color: blue;">Xem</span>	
62bb23feebacab1e16ec9b27	1 phòng	300000 VND	28/06/2022-29/06/2022	vnpay	28/06/2022 22:53	<span style="color: green;">Đã duyệt</span>	<span style="color: green;">Đã xác nhận</span>	<span style="color: blue;">Xem</span>	
62bb276932815228be5ed260	2 phòng	450000 VND	28/06/2022-29/06/2022	vnpay	28/06/2022 23:08	<span style="color: green;">Đã duyệt</span>	<span style="color: green;">Đã xác nhận</span>	<span style="color: blue;">Xem</span>	
62bc1ef09ee9631644cf4f30	1 phòng	300000 VND	03/07/2022-04/07/2022	vnpay	29/06/2022 1644	<span style="color: green;">Đã duyệt</span>	<span style="color: green;">Đã xác nhận</span>	<span style="color: blue;">Xem</span>	
62bc22c546beef30e40222508	1 phòng	200000 VND	29/06/2022-30/06/2022	vnpay	29/06/2022 17:00	<span style="color: green;">Đã duyệt</span>	<span style="color: green;">Đã xác nhận</span>	<span style="color: blue;">Xem</span>	
62bc5adacde0e099180373	1 phòng	300000 VND	29/06/2022-29/06/2022	vnpay	29/06/2022 20:59	<span style="color: green;">Đã duyệt</span>	<span style="color: green;">Đã xác nhận</span>	<span style="color: blue;">Xem</span>	
62bb85d32da032604ff0f38	1 phòng	500000 VND	28/06/2022-29/06/2022	vnpay	26/06/2022 23:14	<span style="color: blue;">Đã hủy</span>	<span style="color: blue;">Đã xác nhận</span>	<span style="color: blue;">Xem</span>	

< 1 2 3 > 10 / page

**Hình 4.18:** Giao diện sau danh sách các đơn đặt phòng

Để biết thông tin chi tiết của đơn, Host cần vào trang chi tiết đơn đặt phòng được mô tả trong **Hình 4.19**. Ở đây, Host có quyền phê duyệt các đơn đặt phòng thông qua 2 nút chấp nhận hoặc từ chối. Nếu ấn chấp nhận, yêu cầu sẽ xác nhận được hiển thị như hình.



**Hình 4.19:** Giao diện chi tiết đơn đặt phòng

### c, Đánh giá homestay

Khi đơn đặt phòng được Host xác nhận và người dùng đã lưu trú ở homestay thì người dùng có thể đánh giá chất lượng homestay đó theo thang điểm 5 và để lại cảm nghĩ của mình trong ô bình luận. **Hình 4.20** mô tả giao diện biểu mẫu nhập đánh giá.

The screenshot shows a review submission form for 'Nắng vàng Hà Nội'.

- Rating:** \* 1. Đánh giá: 4 stars.
- Comment:** \* 2. Nhận xét chỗ nghỉ này: An empty text area with a red border.
- Photo:** Thêm ảnh vào đánh giá của bạn: A file upload input field labeled 'Upload'.
- Post:** Đăng bài: A blue button at the bottom.

A note at the bottom left says: 'Vui lòng nhận xét chỗ nghỉ này'.

**Hình 4.20:** Giao diện biểu mẫu nhập thông tin đánh giá

#### d, Review điểm đến

Tương tự như chức năng đánh giá homestay, khi đơn đặt phòng được Host xác nhận và người dùng đã lưu trú ở homestay thì người dùng có thể chia sẻ, đăng bài review về lịch trình du lịch của cá nhân. **Hình 4.21** mô tả giao diện biểu mẫu nhập review.

**Review của bạn**

\* Chọn địa điểm: Tỉnh Ninh Bình

\* Review của bạn:

Vui lòng review điểm đến

Thêm ảnh: + Upload

Lịch trình của bạn

Ngày 1:

+ Thêm lịch trình

Đăng bài

**Hình 4.21:** Giao diện biểu mẫu nhập thông tin review

#### 4.4 Kiểm thử

**Bảng 4.13** là kết quả kiểm thử các chức năng của hệ thống trên các trình duyệt máy tính,

STT	Chức năng	Cốc cốc	Chrome
1	Đăng ký	Pass	Pass
2	Đăng nhập	Pass	Pass
3	Thêm/sửa/ẩn homestay của tôi	Pass	Pass
4	Thêm/sửa/ẩn phòng của homestay của tôi	Pass	Pass
5	Tìm kiếm phòng trống	Pass	Pass
6	Đặt phòng	Pass	Pass
7	Thay đổi trạng thái đơn đặt phòng	Pass	Pass
8	Nhắn tin trực tiếp	Pass	Pass
9	Đánh giá/comment cho homestay	Pass	Pass
10	Thêm review	Pass	Pass
11	Like bài review	Pass	Pass
12	Chỉnh sửa thông tin cá nhân	Pass	Pass

**Bảng 4.13:** Kết quả kiểm thử trên các trình duyệt máy tính

## 4.5 Triển khai

Hệ thống được triển khai trên môi trường local với:

- Thiết bị: Laptop
- Hệ điều hành: Window 10, MacOS
- Trình duyệt: Cốc cốc, Google Chrome
- Nền tảng: NodeJS Thực thi chạy chương trình từ terminal

Các bước cài đặt để chạy thử trang web trên môi trường local:

Bước 1. Cài đặt git

Bước 2. Cài đặt Nodejs (version 14.17.0)

Bước 3. Tải source code (mã nguồn)

Bước 4. Chạy lệnh npm install trên từng thư mục

Bước 5. Chạy lệnh npm start trên từng thư mục để khởi chạy trang web

## CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Chương này em xin trình bày những nội dung đóng góp mà mình thấy tâm đắc nhất trong suốt quá trình làm ĐATN.

### 5.1 Thiết kế và xây dựng frontend và backend

**Phía frontend:** Tự thiết kế giao diện trang web và xây dựng dựa trên các thành phần có sẵn trong thư viện Ant Design.

**Phía backend:** Xây dựng các chức năng phía backend phù hợp với đối tượng người dùng

- Khách hàng: chức năng đăng ký, tìm kiếm, xem thông tin homestay, các bài nhận xét...
- Người dùng: các chức năng quản lý thông tin cá nhân, đặt phòng, thanh toán, nhận xét và đánh giá phòng, nhận xét điểm đến...
- Host: chức năng quản lý homestay, tiếp nhận, xử lý đơn đặt phòng...
- Quản trị viên: quản lý danh sách người dùng, quản lý danh mục homestay, quản lý danh sách đơn đặt phòng...

### 5.2 Mô tả một số chức năng độc đáo

#### 5.2.1 Cơ chế tổng hợp các bài nhận xét theo địa điểm.

##### a, Vấn đề:

Từ chính bản thân mình và những bạn bè xung quanh, em nhận thấy rằng mọi người thường có thói quen sẽ tìm hiểu những thông tin về các địa điểm du lịch trước khi tự mình đến đó trải nghiệm. Từ đó dẫn đến hai vấn đề sau:

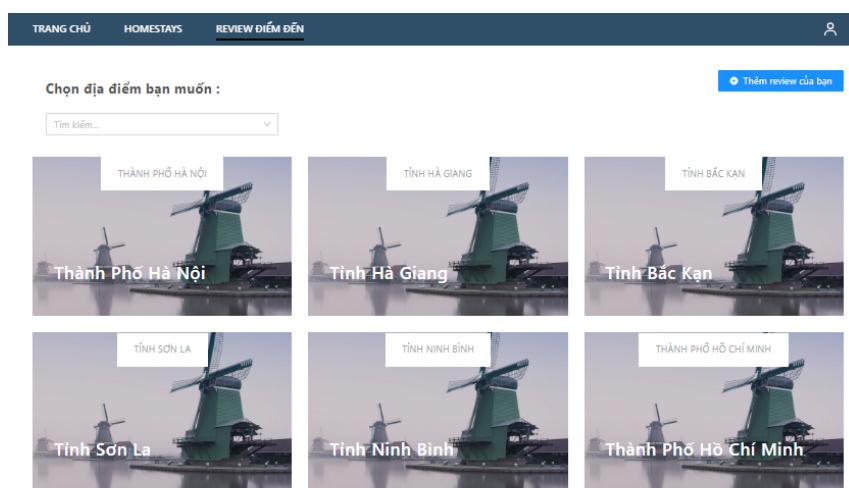
**Thông tin rác:** những thông tin người dùng muốn tìm hiểu thường không tập trung tại một nơi mà rải rác trên các nền tảng. Thông thường mọi người đều tra cứu trên mạng internet hay thông qua các bài viết trên mạng xã hội như là facebook... Trên internet thì đều là những thông tin mang tính khái quát, tổng thể. Trên facebook mặc dù có những nhóm riêng tại mỗi địa điểm du lịch nổi tiếng nhưng hầu hết các bài viết trong nhóm đều nhằm mục đích kinh doanh và hầu như rất ít bài review về địa điểm đó.

**Tính xác thực:** dù thông tin có được tại đâu đi nữa, nhưng do nền tảng internet, mạng xã hội nên bất kỳ ai cũng có thể đăng các thông tin lên đó mà không có bất kỳ ràng buộc nào. Thế nên chúng ta không thể biết được liệu thông tin trong các bài viết đó có đúng hay không, hay chỉ là một bài quảng cáo cho một homestay, địa điểm nào đó.

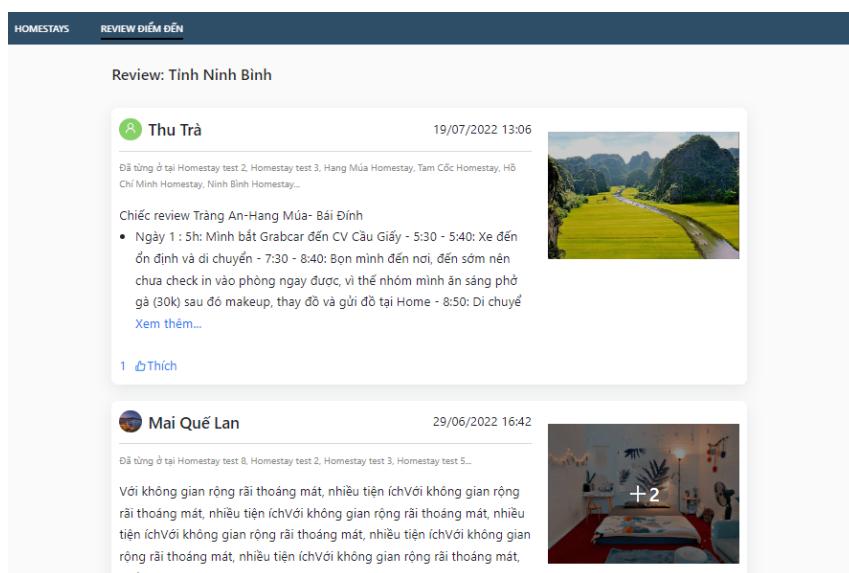
## b, Giải pháp:

Chính từ những vấn đề nêu trên, em đã xây dựng trong hệ thống của mình một cơ chế để tập hợp, nâng cao tính xác thực của những bài review.

**Đối với vấn đề thông tin nằm rải rác:** giải pháp của em chính là tập hợp tất cả các bài review trong hệ thống lại tại một nơi duy nhất là màn “Review điểm đến”, sau đó phân chia các bài review đó theo từng tỉnh thành cụ thể như **Hình 5.1**. Nhờ đó, người dùng bất kỳ chỉ cần chọn địa điểm mình mong muốn là có thể xem được tất cả các review của địa điểm đó. **Hình 5.2** là ví dụ một kết quả hiện ra các bài review theo lựa chọn của người dùng.



**Hình 5.1:** Phân chia các bài review theo từng tỉnh thành



**Hình 5.2:** Ví dụ kết quả khi chọn xem review của một tỉnh thành

**Đối với vấn đề xác thực , tính tin cậy trong các bài review:** để giải quyết được

vấn đề này, giải pháp hiện tại của em là đưa ra quy tắc bắt buộc rằng người dùng chỉ có thể đăng bài review trên hệ thống khi đảm bảo được 2 điều kiện như sau:

- Thứ nhất: cần có tài khoản và đăng nhập vào hệ thống
- Thứ hai: đã từng lưu trú tại ít nhất một nơi tại địa điểm muốn review

Để thuận tiện cho người dùng, em đưa ra các quy luật tạo thông báo/hành động của hệ thống theo từng trường hợp của người dùng khi muốn tao ra bài review cá nhân và mô tả chúng ở **Bảng 5.1**.

Trường hợp	Thông báo/Hành động
Chưa đăng nhập	Bạn cần đăng nhập để thực hiện chức năng này ?
Đã đăng nhập nhưng chưa từng lưu trú	Bạn chưa từng đến địa điểm nào
Đã đăng nhập và từng lưu trú	Hiển thị biểu mẫu nhập thông tin review

**Bảng 5.1:** Quy luật tạo thông báo/hành động

Ngoài ra để giúp người dùng có thêm thông tin về chất lượng các bài review, em có bổ sung thêm chức năng yêu thích (like) bài review như trong **Hình 5.2**. Lý do em bổ sung tính năng này chính là bởi vì một bài review khi có càng nhiều lượt yêu thích thì chứng tỏ bài review đó hay, chi tiết, mức độ tin cậy lớn, nên khi nhìn vào đó người dùng sẽ dễ dàng phán đoán hơn. Hơn nữa, với mỗi bài review, hệ thống cũng sẽ hiển thị thêm những homestay mà người dùng đã ở tại địa điểm đó.

### c, Định hướng phát triển:

Định hướng phát triển của em dành cho tính năng này là làm thế nào để người dùng tích cực tham gia vào việc chia sẻ review lịch trình du lịch của mình và dành tâm huyết để tạo ra một bài review chất lượng cao. Từ định hướng đó, một số hướng tiếp cận mà em đang hướng tới đó là sau khi đưa hệ thống đi vào sử dụng trong thực tế, với mỗi một bài review người dùng sẽ được tặng quà như là voucher, thẻ tích điểm..., hay với mỗi bài review có một lượng yêu thích nhất định ( 100, 1000... lượt) cũng sẽ nhận được những ưu đãi tương đương.

#### 5.2.2 Cơ chế tìm phòng trống theo các tiêu chí khác nhau

##### a, Vấn đề:

Điểm quan trọng nhất trong một hệ thống đặt phòng đó là tìm ra thuật toán tìm kiếm homestay còn phòng trống theo thời gian tại một địa điểm cho trước. Thế nhưng nếu chỉ có tìm kiếm theo thời gian và địa điểm thôi thì người dùng sẽ tốn thêm nhiều công sức để tìm ra được phòng phù hợp với các tiêu chí khác của bản thân. Vậy nên việc bổ sung thêm các điều kiện tìm kiếm khác là điều cần thiết.

### b, Giải pháp:

Để giải quyết vấn đề này, em có bổ sung 3 điều kiện tìm kiếm tích hợp cho hệ thống đó là: tìm kiếm theo tên, sắp xếp theo giá và sắp xếp theo sao (điểm người dùng đánh giá cho homestay). Người dùng có thể chọn tích hợp một số các điều kiện trên để tìm kiếm phòng mà mình mong muốn.

#### Tìm kiếm theo thời gian và địa điểm

Khi nhập thời gian và địa điểm du lịch (ví dụ: chọn Ninh Bình, từ 19-20/07/2022), hệ thống sẽ hiển thị 3 homestay thỏa mãn như **Hình 5.3**.

The screenshot shows a search interface for 'HOMESTAYS' with the title 'REVIEW ĐIỂM ĐẾN'. It includes fields for 'Chọn địa điểm bạn muốn:' (selected: Tỉnh Ninh Bình), 'Ngày check in và check out:' (selected: 2022-07-19 - 2022-07-20), and 'Tim kiếm Homestay' (Search). Below the search bar is a button labeled 'Tim kiếm' (Search). The results section is titled 'Sắp xếp theo : Mặc định' (Sort by: Default) and shows three homestay options:

- Hang Múa Homestay**: ★★★★☆ (4 stars), located in Ninh Bình, described as modern with wooden furniture. Price: CHỈ TỪ 280.000 ₫.
- Tam Cốc Homestay**: ★★★★☆ (4 stars), located in Ninh Bình, described as modern with wooden furniture. Price: CHỈ TỪ 400.000 ₫.
- Ninh Bình Homestay**: ★★★★☆ (4 stars), located in Ninh Bình, described as cheap. Price: CHỈ TỪ 320.000 ₫.

**Hình 5.3:** Tìm kiếm theo thời gian và địa điểm

#### Sắp xếp theo sao

Với chức năng này, người dùng có thể sắp xếp các homestay tìm được theo số sao từ thấp đến cao hoặc từ cao xuống thấp. **Hình 5.4** mô tả lựa chọn sắp xếp từ cao đến thấp.

The screenshot shows a search interface for 'HOMESTAYS' with the title 'REVIEW ĐIỂM ĐẾN'. It includes fields for 'Chọn địa điểm bạn muốn:' (selected: Tỉnh Ninh Bình), 'Ngày check in và check out:' (selected: 2022-07-19 - 2022-07-20), and 'Tim kiếm Homestay' (Search). Below the search bar is a button labeled 'Tim kiếm' (Search). The results section is titled 'Sắp xếp theo : Sao giảm dần' (Sort by: Star rating descending) and shows three homestay options:

- Hang Múa Homestay**: ★★★★☆ (4 stars), located in Ninh Bình, described as modern with wooden furniture. Price: CHỈ TỪ 280.000 ₫.
- Ninh Bình Homestay**: ★★★★☆ (4 stars), located in Ninh Bình, described as cheap. Price: CHỈ TỪ 320.000 ₫.
- Tam Cốc Homestay**: ★★★★☆ (4 stars), located in Ninh Bình, described as modern with wooden furniture. Price: CHỈ TỪ 400.000 ₫.

**Hình 5.4:** Sắp xếp giảm dần theo sao

## Sắp xếp theo giá

Tương tự chức năng sắp xếp theo sao, **Hình 5.5** mô tả lựa chọn sắp xếp giá từ cao đến thấp.

**Hình 5.5:** Sắp xếp tăng theo giá

## Tìm kiếm theo tên

**Hình 5.6:** Tìm kiếm theo tên

## c, Định hướng phát triển:

Ngoài các điều kiện đã được nêu, thì còn nhiều điều kiện khác sẽ hỗ trợ tối đa cho người dùng như là: tìm kiếm theo số lượng người lớn - trẻ em, theo khoảng giá (từ 300000đ đến 600000đ...),... Chính vì thế định hướng em dành cho chức năng này là bổ sung thêm nhiều điều kiện tìm kiếm sao cho phù hợp và có ích đối với người dùng.

### 5.2.3 Cơ chế trao đổi thông tin trực tiếp giữa người dùng và host

#### a, Vấn đề:

Trước khi quyết định có thuê phòng tại homestay hay không, sẽ có những trường hợp người dùng mong muốn biết thêm nhiều thông tin khác ngoài những thông tin được đăng tải trên hệ thống. Mặc dù có thể liên lạc trực tiếp qua số điện thoại để trao đổi nhưng việc này gây bất tiện và tốn thời gian cho cả người dùng và Host. Từ vấn đề đó, em nhận thấy việc xây dựng một chức năng trao đổi thông tin trực tiếp với Host là cần thiết.

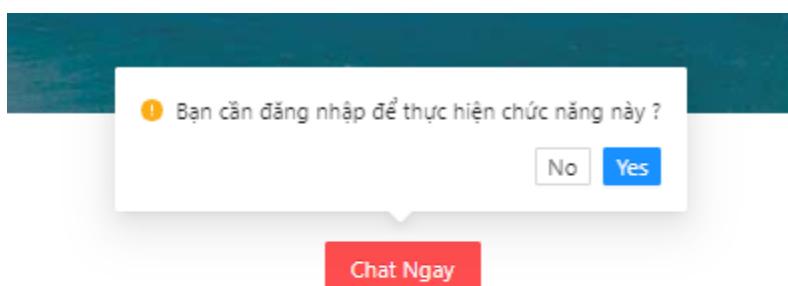
#### b, Giải pháp:

Để giải quyết vấn đề này, em đã tạo ra tính năng nhắn tin với 2 loại boxchat riêng biệt và mô tả chúng trong **Bảng 5.2**.

Loại	Đối tượng	Vị trí	Vai trò
1	Người dùng	Boxchat xuất hiện ở mọi trang có trong hệ thống	Đây là boxchat được người dùng sử dụng để liên lạc với Host khi muốn biết thêm thông tin về homestay
2	Host	Boxchat xuất hiện ở trang "Homestay của tôi"	Đây là boxchat được Host sử dụng để trả lời, giải đáp thắc mắc được gửi đến từ người dùng

**Bảng 5.2:** Các loại boxchat

Để sử dụng tính năng này, người dùng phải có tài khoản và đăng nhập vào hệ thống. Nếu chưa có, hệ thống sẽ hiện thông báo như **Hình 5.7**



**Hình 5.7:** Thông báo yêu cầu đăng nhập

Đối với lần đầu tiên liên lạc với Host, người dùng chỉ có thể liên lạc thông qua chức năng chat ngay trong trang chi tiết homestay.

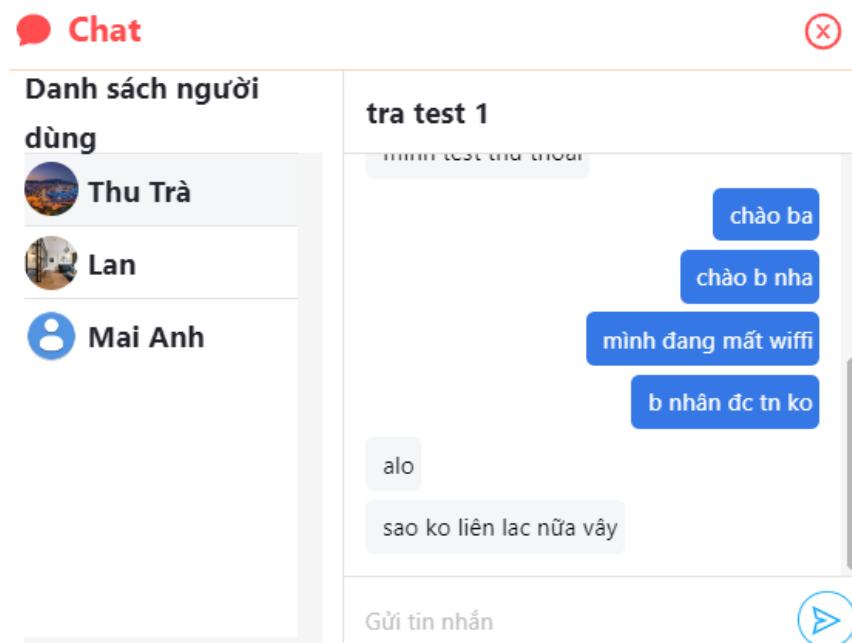
Các lần sau người dùng có thể liên lạc thông qua boxchat cá nhân của mình tại bất kỳ trang nào trên hệ thống. **Hình 5.8** minh họa cho boxchat cá nhân của người dùng

bao gồm danh sách homestay người dùng đã từng liên lạc, nội dung trò chuyện và ô nhập tin nhắn mới.



**Hình 5.8:** Boxchat cá nhân

Tương tự như boxchat cá nhân **Hình 5.9** minh họa cho boxchat của Host, bao gồm danh sách những người dùng đã từng liên lạc với homestay, nội dung trò chuyện và ô nhập tin nhắn mới



**Hình 5.9:** Boxchat của Host

Về công nghệ, em đã sử dụng riêng **Firebase**[9] [10] (là một dịch vụ cơ sở dữ liệu thời gian thực được cung cấp bởi Google và hoạt động trên nền tảng đám mây) cho chức năng nhắn tin. Ưu điểm và nhược điểm của Firebase được mô tả trong **Bảng 5.3**.

Ưu điểm	Nhược điểm
Xử lý mọi thứ trong thời gian thực dễ dàng, nhanh gọn.	Chi phí đắt đỏ - giá cả không ổn định do đặc tính firebase tính tiền dựa trên mức độ sử dụng nên việc giới hạn giá cả trở lên khó khăn và không rõ ràng, khó đoán được chi phí bỏ ra khi dữ liệu lớn lên.
Firebase đi kèm với kiến trúc không máy chủ, do vậy người dùng không cần suy nghĩ về cơ sở hạ tầng.	Firebase tổ chức cơ sở dữ liệu theo kiểu trees, parent-children chứ không phải kiểu table. Nên sẽ gây khó khăn với những ai đang quen với CSDL SQL.

**Bảng 5.3:** So sánh ưu nhược điểm của Firebase

Nhờ những ưu điểm trên mà chức năng này tương đối hoàn thiện, người dùng sẽ không cần phải tải lại trang để có thể xem tin nhắn mới từ đối phương. Hơn nữa, Firebase hỗ trợ cả cơ chế offline-sync DB, cơ chế này giúp cho người dùng trong tình huống khi đang trao đổi với nhau mà bị mất mạng, thì người dùng hay Host vẫn có thể tiếp tục gửi tin nhắn mới và khi mạng được kết nối lại nó sẽ tự cập nhật những tin nhắn đã gửi trong lúc đó.

Để khắc phục những nhược điểm được nêu ở trên, em không dùng Firebase để lưu trữ toàn bộ dữ liệu của trang web mà chỉ dùng để lưu dữ liệu liên quan đến chức năng nhắn tin nhằm giảm chi phí phải bỏ ra và sử dụng MongoDB để lưu trữ những dữ liệu khác.

### c, Định hướng phát triển:

Định hướng phát triển của em dành cho tính năng này đó là bổ sung thêm thông báo mỗi lần tin nhắn mới được gửi đến và thay đổi một chút giao diện để người dùng dễ dàng nhận ra những tin nhắn nào đã được đọc, đã được phản hồi.

## CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Đây là chương tổng kết kiến thức đã học, các kinh nghiệm rút ra và các kết quả mà em đã đạt được trong suốt quá trình làm ĐATN. Sau đó, em đưa ra một vài hướng phát triển bản thân chưa kịp làm để cải thiện, nâng cao chất lượng cho trang web.

### 6.1 Kết luận

Trong suốt quá trình thực hiện ĐATN, em đã hoàn thành trang web đặt phòng, thuê homestay đơn giản với những chức năng cơ bản của người dùng. Mặc dù chỉ mới là sản phẩm còn đơn sơ nhưng thông qua quá trình vừa làm vừa tìm hiểu em đã học hỏi và hoàn thiện được các kỹ năng như:

- Các công nghệ phát triển web: HTML, CSS, Javascript, ReactJS, NodeJS,... và một số thư viện hỗ trợ lập trình web khác như Ant Design...
- Sử dụng tương đối các câu truy vấn cơ sở dữ liệu.
- Biết dùng git để quản lý source code.
- Biết sử dụng kết hợp firebase để lưu trữ dữ liệu.

Đồng thời, em cũng rút ra được cho mình những bài học kinh nghiệm:

- Cần phải tìm hiểu thật kỹ đề tài mà mình chọn trước khi bắt tay vào thực hiện
- Việc suy xét kỹ lưỡng cấu trúc của cơ sở dữ liệu là điều vô cùng quan trọng. Nếu cơ sở dữ liệu không chính xác ngay từ đầu sẽ mất rất nhiều thời gian để sửa, thậm chí là phải bắt đầu lại từ đầu.
- Thiết kế giao diện không chỉ cần đẹp mà còn phải cần sự thống nhất toàn cục trong trang web, điều này giúp người dùng dễ dàng sử dụng và có trải nghiệm tốt hơn.
- Trước khi dùng thư viện, ngôn ngữ nào đó thì nên tìm hiểu sơ qua các thư viện, ngôn ngữ tương tự, so sánh ưu nhược điểm để tìm ra thư viện, ngôn ngữ phù hợp nhất với trang web của mình.

## 6.2 Hướng phát triển

Một số hướng phát triển, định hướng công việc trong tương lai mà em dành cho hệ thống:

- Trước tiên, để hoàn thiện các chức năng đã làm, cần đưa trang web vào sử dụng thực tế để lấy ý kiến, phản hồi của người dùng làm cơ sở để sửa đổi, bổ sung các tính năng cần thiết.
- Xây dựng tính năng giúp Host cài đặt giá phòng thay đổi theo thời gian mong muốn, để Host tạo ra và cung cấp các mã giảm giá của riêng homestay.
- Cải tiến hệ thống với chức năng quản lý không đơn thuần chỉ là quản lý thông tin mà có thể thống kê doanh thu, thống kê tỷ lệ lấp đầy phòng của homestay.
- Phát triển trang web thành các ứng dụng trên điện thoại di động để thuận tiện hơn với người dùng.

## TÀI LIỆU THAM KHẢO

- [1] S. Aggarwal, “Modern web-development using reactjs,” *International Journal of Recent Research Aspects*, vol. 5, no. 1, pp. 133–137, 2018.
- [2] *Reactjs documentation*. [Online]. Available: <https://reactjs.org/> (visited on 06/30/2022).
- [3] M. Satheesh, B. J. D’mello, and J. Krol, *Web development with MongoDB and NodeJs*. Packt Publishing Ltd, 2015.
- [4] E. Hahn, *Express in Action: Writing, building, and testing Node.js applications*. Simon and Schuster, 2016.
- [5] *Express documentation*. [Online]. Available: <https://expressjs.com/> (visited on 06/30/2022).
- [6] K. Banker, D. Garrett, P. Bakkum, and S. Verch, *MongoDB in action: covers MongoDB version 3.0*. Simon and Schuster, 2016.
- [7] A. Mardan, “Boosting node.js and mongodb with mongoose,” in *Practical Node.js*, Springer, 2018, pp. 239–276.
- [8] *Ant design documentation*. [Online]. Available: <https://ant.design/> (visited on 06/30/2022).
- [9] *Firebase documentation*. [Online]. Available: <https://firebase.google.com/> (visited on 06/30/2022).
- [10] L. Moroney, “The firebase realtime database,” in *The Definitive Guide to Firebase*, Springer, 2017, pp. 51–71.