

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

**VIỆN ĐIỆN TỬ - VIỄN THÔNG**

✯✯✯✯✯✯✯✯✯



**ĐỒ ÁN**

**TỐT NGHIỆP ĐẠI HỌC**

**Đề tài:**

**THIẾT KẾ HỆ THỐNG TÌM KIẾM**

**KHÁCH SẠN, NHÀ TRỌ**

Nhóm sinh viên thực hiện:

|  |  |  |
| --- | --- | --- |
| Họ tên | Mã số sinh viên | Lớp |
| Trần Quang Huy | 20131798 | ĐTTT 03 – K58 |
| Tạ Anh Tú | 20134509 | ĐTTT 04 – K58 |

Giáo viên hướng dẫn: TS. Nguyễn Tiến Hòa

Hà Nội, 6/2018



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

**VIỆN ĐIỆN TỬ - VIỄN THÔNG**

✯✯✯✯✯✯✯✯✯



**ĐỒ ÁN**

**TỐT NGHIỆP ĐẠI HỌC**

**Đề tài:**

**THIẾT KẾ HỆ THỐNG TÌM KIẾM**

**KHÁCH SẠN, NHÀ TRỌ**

Nhóm sinh viên thực hiện:

|  |  |  |
| --- | --- | --- |
| Họ tên | Mã số sinh viên | Lớp |
| Trần Quang Huy | 20131798 | ĐTTT 03 – K58 |
| Tạ Anh Tú | 20134509 | ĐTTT 04 – K58 |

Giáo viên hướng dẫn: TS. Nguyễn Tiến Hòa

Cán bộ phản biện:

Hà Nội, 6/2018

**Đánh giá quyển đồ án tốt nghiệp**

**(Dùng cho giảng viên hướng dẫn)**

Giảng viên đánh giá: **TS. Nguyễn Tiến Hòa**

Họ và tên Sinh viên: **Tạ Anh Tú** MSSV: **20134509**

Tên đồ án: ***Thiết kế hệ thống tìm kiếm khách sạn, nhà trọ***

***Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:***

***Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Có sự kết hợp giữa lý thuyết và thực hành (20)** | | | | | | | |
| 1 | Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án | | 1 | 2 | 3 | 4 | 5 |
| 2 | Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế) | | 1 | 2 | 3 | 4 | 5 |
| 3 | Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề | | 1 | 2 | 3 | 4 | 5 |
| 4 | Có kết quả mô phỏng/thưc nghiệm và trình bày rõ ràng kết quả đạt được | | 1 | 2 | 3 | 4 | 5 |
| **Có khả năng phân tích và đánh giá kết quả (15)** | | | | | | | |
| 5 | Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống | | 1 | 2 | 3 | 4 | 5 |
| 6 | Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng. | | 1 | 2 | 3 | 4 | 5 |
| 7 | Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai. | | 1 | 2 | 3 | 4 | 5 |
| **Kỹ năng viết (10)** | | | | | | | |
| 8 | Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định | | 1 | 2 | 3 | 4 | 5 |
| 9 | Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.) | | 1 | 2 | 3 | 4 | 5 |
| **Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)** | | | | | | | |
| 10a | | Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế | 5 | | | | |
| 10b | | Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest. | 2 | | | | |
| 10c | | Không có thành tích về nghiên cứu khoa học | 0 | | | | |
| **Điểm tổng** | | | **/50** | | | | |
| **Điểm tổng quy đổi về thang 10** | | |  | | | | |

***\* Nhận xét thêm của Thầy/Cô (****giảng viên hướng dẫn nhận xét về thái độ và tinh thần làm việc của sinh viên****)***

............................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

Ngày: / /2018

Người nhận xét

(Ký và ghi rõ họ tên)

**Đánh giá quyển đồ án tốt nghiệp**

**(Dùng cho cán bộ phản biện)**

Giảng viên đánh giá:................................................................................................

Họ và tên Sinh viên: **Tạ Anh Tú**  MSSV: **20134509**

Tên đồ án: ***Thiết kế hệ thống tìm kiếm khách sạn, nhà trọ***

***Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:***

***Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Có sự kết hợp giữa lý thuyết và thực hành (20)** | | | | | | | |
| 1 | Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án | | 1 | 2 | 3 | 4 | 5 |
| 2 | Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế) | | 1 | 2 | 3 | 4 | 5 |
| 3 | Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề | | 1 | 2 | 3 | 4 | 5 |
| 4 | Có kết quả mô phỏng/thưc nghiệm và trình bày rõ ràng kết quả đạt được | | 1 | 2 | 3 | 4 | 5 |
| **Có khả năng phân tích và đánh giá kết quả (15)** | | | | | | | |
| 5 | Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống | | 1 | 2 | 3 | 4 | 5 |
| 6 | Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng. | | 1 | 2 | 3 | 4 | 5 |
| 7 | Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai. | | 1 | 2 | 3 | 4 | 5 |
| **Kỹ năng viết (10)** | | | | | | | |
| 8 | Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có căn lề, dấu cách sau dấu chấm, dấu phẩy v.v), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định | | 1 | 2 | 3 | 4 | 5 |
| 9 | Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.) | | 1 | 2 | 3 | 4 | 5 |
| **Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)** | | | | | | | |
| 10a | | Có bài báo khoa học được đăng hoặc chấp nhận đăng/đạt giải SVNC khoa học giải 3 cấp Viện trở lên/các giải thưởng khoa học (quốc tế/trong nước) từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế | 5 | | | | |
| 10b | | Được báo cáo tại hội đồng cấp Viện trong hội nghị sinh viên nghiên cứu khoa học nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành như TI contest. | 2 | | | | |
| 10c | | Không có thành tích về nghiên cứu khoa học | 0 | | | | |
| **Điểm tổng** | | | **/50** | | | | |
| **Điểm tổng quy đổi về thang 10** | | |  | | | | |

***\* Nhận xét thêm của Thầy/Cô***

............................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................ Ngày: / /2018

Người nhận xét

(Ký và ghi rõ họ tên)

# **Lời nói đầu**

Hiện nay, nhu cầu du lịch của mọi người ngày một tăng, đặc biệt là các bạn trẻ, những người luôn muốn đặt chân tới những vùng đất mới để tìm hiểu và học hỏi. Với những người không có thời gian đặt chỗ trước thì việc tìm kiếm một khách sạn hay phòng trọ sau khi đặt chân tới là một việc không hề dễ dàng, vì nhiều người còn phải xem xét tới nhiều yếu tố như diện tích, địa điểm, cảnh quan đẹp, giá cả… có phù hợp hay không. Xuất phát từ yêu cầu đó, em và các bạn khác trong nhóm đã chọn đề tài "Thiết kế hệ thống tìm khách sạn, nhà trọ" để giúp những người như vậy dễ dàng tìm được chỗ ở ưng ý trong thời gian du lịch.

Hệ thống tìm kiếm nhà trọ có thể được sử dụng trên các thiết bị di động thông minh hoặc nền tảng web, phù hợp và thuận tiện với mọi người sử dụng.

Em xin chân thành cảm ơn thầy Nguyễn Tiến Hòa, người đã nhiệt tình hướng dẫn em hoàn thành đồ án này. Xin cảm ơn anh Dương Mạnh Tình ở bộ phận Mobile Solution thuộc công ty SMVC đã gợi ý và giúp đỡ em về mặt ý tưởng, các công cụ hỗ trợ trong quá trình làm đồ án.

Do thời gian nghiên cứu, tìm tòi không nhiều nên chắc hẳn hệ thống không thể tránh khỏi sai sót, mong thầy cô góp ý chân thành để em có thể hoàn chỉnh hơn nữa sản phẩm của mình.

Tạ Anh Tú

# **TÓM TẮT ĐỒ ÁN**

Đồ án này có tên đề tài là: "thiết kế hệ thống tìm kiếm khách sạn, nhà trọ". Sau đây là mục đích chính của đồ án:

1. Thu thập dữ liệu từ các website khác nhau về khách sạn, nhà trọ, các trang cho thuê nhà khác. Việc thu thập này dùng để xây dựng cơ sở dữ liệu cho hệ thống, giúp người dùng có thể dễ dàng so sách các nhà trọ từ nhiều nguồn khác nhau.
2. Xây dựng giao diện tiện ích trên 2 nền tảng khác nhau là ứng dụng di dộng (hệ điều hành Android) và nền tảng web. Điều này giúp cho người dùng có thể truy cập hệ thống ở bất kỳ nơi đâu với bất kỳ thiết bị điện thoại hay máy tính gì.
3. Xây dựng chức năng tìm kiếm khách sạn, nhà trọ (dựa trên vị trí hiện tại hoặc dựa theo tên) và chức năng chỉ đường, giúp người dùng dễ dàng tìm thấy nhà trọ hay khách sạn mà mình mong muốn.
4. Xây dựng chức năng cho phép người dùng đăng tải thông tin thuê nhà, giúp người dùng, đặc biệt là chủ các nhà nghỉ, nhà trọ dễ dàng quảng bá nhà trọ mà họ cần cho thuê.
5. Xây dựng chức năng đánh giá, phản hồi, cho phép người dùng đánh giá và nhận xét về bất kỳ dịch vụ nhà trọ mà họ đã dùng. Điều này sẽ khiến mọi người dễ dàng cảm nhận được chất lượng dịch vụ của nhà trọ, khách sạn mà họ đang tìm kiếm, dù họ chưa tới đó bao giờ.

Đồ án này gồm 5 chương. Chương đầu tiên sẽ là giới thiệu khái quát về đề tài và lý do chọn; kịch bản hoạt động, quy mô của đồ án và phân công công việc. Chương 2 đề cập đến cơ sở lý thuyết, những công nghệ sẽ được sử dụng trong quá trình xây dựng hệ thống. Chương 3 sẽ trình bày về khâu phân tích hệ thống thông qua việc sử dụng ngôn ngữ mô hình hóa UML (bao gồm các sơ đồ use case, sơ đồ tuần tự, sơ đồ trạng thái, biểu đồ hoạt động, biểu đồ lớp…). Chương 4 sẽ đưa ra các bước thiết kế hệ thống. Chương cuối cùng sẽ là phần demo kết quả đạt được và phương hướng phát triển hệ thống trong tương lai.

# **ABSTRACTION**

This project is entitled "finding hotel, guesthouse system". The following is the main purpose of the project:

1. Crawl data from various websites about hotels, guesthouses, other rental sites. This crawling is used to build the database for the system, making it easy for users to compare hotels, guesthouses from multiple sources.
2. Build a user interface on two different platforms, the mobile application (Android operating system) and the web platform. This gives users access to the system anywhere with any device or computer.
3. Build a search function for hotels, guesthouses (based on current location or by name) and navigation functions, making it easy to find the desired guesthouse or hotel.
4. Allows users to post rental information, helping users, especially landlords, to easily promote the guesthouse they need to rent.
5. Build review, feedback functions that allows users to review and comment on any guesthouse services they have used. This will make it easy for people to feel the quality of the guesthouse, the hotel they are looking for, even if they have not been there yet.

This project consists of 5 chapters. The first chapter will give an overview of the topic and the reasons for the selection; operational scenario, project size and job assignment. Chapter 2 mentions about theoretical foundations, what technologies will be used in the design process. Chapter 3 discusses about system analysis using UML modeling language (including use case diagrams, sequence diagrams, state diagrams, activity diagrams, class diagrams, etc). Chapter 4 is about how to design the system. The final chapter will be a demo of the results and future development of the system.

Mục lục

[**Lời nói đầu** 1](#_Toc516214677)

[**TÓM TẮT ĐỒ ÁN** 2](#_Toc516214678)

[**ABSTRACTION** 3](#_Toc516214679)

[**DANH MỤC HÌNH ẢNH** 8](#_Toc516214680)

[**THUẬT NGỮ VÀ VIẾT TẮT** 10](#_Toc516214681)

[**CHƯƠNG 1. GIỚI THIỆU** 11](#_Toc516214682)

[**1.1. Đặt vấn đề** 11](#_Toc516214683)

[**1.2. Kịch bản hoạt động** 12](#_Toc516214684)

[*1.2.1 Quy mô* 12](#_Toc516214685)

[*1.2.2 Người sử dụng* 12](#_Toc516214686)

[*1.2.3 Tìm kiếm* 12](#_Toc516214687)

[**1.3. Phân công công việc** 12](#_Toc516214688)

[**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT** 14](#_Toc516214689)

[**2.1. Hệ quản trị cơ sở dữ liệu MySQL** 14](#_Toc516214690)

[**2.2. Spring Boot và Spring Security** 14](#_Toc516214691)

[**2.3. GraphQL** 15](#_Toc516214692)

[*2.3.1. Query* 16](#_Toc516214693)

[*2.3.2. Mutation* 17](#_Toc516214694)

[*2.3.3. Schema và Type* 18](#_Toc516214695)

[**2.4. Google map API** 18](#_Toc516214696)

[*2.4.1. Giới thiệu về google map* 18](#_Toc516214697)

[*2.4.2. Giới thiệu về google map API* 19](#_Toc516214698)

[*2.4.3. Giới thiệu về google place API* 19](#_Toc516214699)

[*2.4.4. Giới thiệu về google place API* 19](#_Toc516214700)

[*2.4.5. Google Direction API* 20](#_Toc516214701)

[*2.4.6. Google Geocoding API* 20](#_Toc516214702)

[**2.5. Selenium WebDriver** 21](#_Toc516214703)

[*2.5.1. Giới thiệu* 21](#_Toc516214704)

[*2.5.2. Sử dụng Selenium WebDriver* 21](#_Toc516214705)

[**CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG** 23](#_Toc516214706)

[**3.1. Yêu cầu chức năng** 23](#_Toc516214707)

[**3.2. Yêu cầu phi chức năng** 23](#_Toc516214708)

[**3.3. Sơ đồ use case của hệ thống (Use case Diagram)** 23](#_Toc516214709)

[**3.4. Sơ đồ phân rã use case** 25](#_Toc516214710)

[*3.4.1. Use case đăng ký tài khoản* 25](#_Toc516214711)

[*3.4.2. Use case đăng nhập* 26](#_Toc516214712)

[*3.4.3. Use case tìm kiếm nhà trọ* 27](#_Toc516214713)

[*3.4.4. Use case xem thông tin nhà trọ* 28](#_Toc516214714)

[*3.4.5. Use case sửa thông tin* 29](#_Toc516214715)

[*3.4.6. Use case chỉ đường* 30](#_Toc516214716)

[*3.4.7. Use case đăng xuất* 30](#_Toc516214717)

[*3.4.8. Use case đánh giá, phản hồi* 30](#_Toc516214718)

[*3.4.9. Use case xóa bình luận* 31](#_Toc516214719)

[*3.4.10. Use case báo cáo thống kê* 31](#_Toc516214720)

[*3.4.11. Use case phê duyệt bài đăng* 32](#_Toc516214721)

[**3.5. CRC Cards (Class-Responsibility-Collaborators)** 32](#_Toc516214722)

[**3.6. Biểu đồ lớp (Class Diagram)** 35](#_Toc516214723)

[**3.7. Biểu đồ hoạt động (Activity Diagram)** 36](#_Toc516214724)

[*3.7.1. Chức năng đăng nhập* 36](#_Toc516214725)

[*3.7.2. Chức năng đăng ký* 37](#_Toc516214726)

[*3.7.3. Chức năng tìm kiếm nhà trọ* 38](#_Toc516214727)

[*3.7.4. Chức năng xem thông tin nhà trọ* 39](#_Toc516214728)

[*3.7.5. Chức năng đăng tin thuê nhà* 40](#_Toc516214729)

[**3.8. Biểu đồ tuần tự (Sequence diagram)** 41](#_Toc516214730)

[*3.8.1. Chức năng đăng ký* 41](#_Toc516214731)

[*3.8.2. Chức năng đăng nhập* 41](#_Toc516214732)

[*3.8.3. Chức năng tìm nhà trọ* 42](#_Toc516214733)

[*3.8.4. Chức năng xem thông tin nhà trọ* 43](#_Toc516214734)

[*3.8.5. Chức năng đăng ký nhà trọ (đăng tin)* 44](#_Toc516214735)

[**3.9. Mô hình thực thể liên kết** 44](#_Toc516214736)

[**CHƯƠNG 4. THIẾT KẾ HỆ THỐNG** 46](#_Toc516214737)

[**4.1. Sơ đồ khối của hệ thống** 46](#_Toc516214738)

[**4.2. Mô hình MVC cho phía backend** 47](#_Toc516214739)

[**4.3. Thiết kế cơ sở dựa liệu** 48](#_Toc516214740)

[**4.4. Thiết kế giao diện người dùng** 50](#_Toc516214741)

[*4.4.1. Giao diện đăng nhập* 50](#_Toc516214742)

[*4.4.2. Giao diện đăng ký* 50](#_Toc516214743)

[*4.4.3. Giao diện trang chủ* 51](#_Toc516214744)

[*4.4.4. Giao diện đăng bài* 52](#_Toc516214745)

[*4.4.5. Giao diện từng khách sạn* 52](#_Toc516214746)

[**CHƯƠNG 5. KẾT QUẢ VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN** 56](#_Toc516214747)

[**5.1. Kết quả đạt được** 56](#_Toc516214748)

[*5.1.1. Ứng dụng trên nền tảng web* 56](#_Toc516214749)

[*5.1.2. Ứng dụng trên nền tảng Android* 61](#_Toc516214750)

[**5.2. Kết luận** 67](#_Toc516214751)

[**5.3. Phương hướng phát triển** 68](#_Toc516214752)

[**TÀI LIỆU THAM KHẢO** 69](#_Toc516214753)

**DANH MỤC HÌNH ẢNH**

[Hình 1-1. Kịch bản hoạt động của hệ thống 12](#_Toc516184164)

[Hình 2-1. Hệ quản trị cơ sở dữ liệu MySQL 14](#_Toc516214754)

[Hình 2-2. Framework Spring boot 14](#_Toc516214755)

[Hình 2-3. GraphQL 15](#_Toc516214756)

[Hình 3-1. Sơ đồ use case tổng quát của hệ thống tìm kiếm nhà trọ 24](#_Toc516214757)

[Hình 3-2. Phân rã use case đăng ký 25](#_Toc516214758)

[Hình 3-3. Phân rã use case đăng nhập 26](#_Toc516214759)

[Hình 3-4. Phân rã use case tìm nhà trọ 27](#_Toc516214760)

[Hình 3-5. Phân rã use case xem thông tin nhà trọ 28](#_Toc516214761)

[Hình 3-6. Phân rã use case sửa thông tin 29](#_Toc516214762)

[Hình 3-7. Phân rã use case đánh giá, phản hồi 30](#_Toc516214763)

[Hình 3-8. Phân rã use case báo cáo thống kê 31](#_Toc516214764)

[Hình 3-9. Sơ đồ lớp của hệ thống 35](#_Toc516214765)

[Hình 3-10. Biểu đồ hoạt động dành cho chức năng đăng nhập 36](#_Toc516214766)

[Hình 3-11. Biểu đồ hoạt động dành cho chức năng đăng ký 37](#_Toc516214767)

[Hình 3-12. Biểu đồ hoạt động dành cho chức năng tìm nhà trọ 38](#_Toc516214768)

[Hình 3-13. Biểu đồ hoạt động dành cho chức năng xem thông tin nhà trọ 39](#_Toc516214769)

[Hình 3-14. Biểu đồ hoạt động dành cho chức năng đăng tin thuê nhà 40](#_Toc516214770)

[Hình 3-15. Biểu đồ tuần tự của chức năng đăng ký tài khoản 41](#_Toc516214771)

[Hình 3-16. Biểu đồ tuần tự của chức năng đăng nhập 41](#_Toc516214772)

[Hình 3-17. Biểu đồ tuần tự của chức năng tìm nhà trọ 42](#_Toc516214773)

[Hình 3-18. Biểu đồ tuần tự của chức năng xem thông tin nhà trọ 43](#_Toc516214774)

[Hình 3-19. Biểu đồ tuần tự của chức năng đăng ký nhà trọ 44](#_Toc516214775)

[Hình 3-20. Mô hình thực thể liên kết 44](#_Toc516214776)

[Hình 4-1. Sơ đồ khối của hệ thống 46](#_Toc516214777)

[Hình 4-2. Mô hình MVC 47](#_Toc516214778)

[Hình 4-3. Các bảng trong cơ sở dữ liệu của hệ thống 48](#_Toc516214779)

[Hình 4-4. Giao diện đăng nhập 50](#_Toc516214780)

[Hình 4-5. Giao diện đăng ký 50](#_Toc516214781)

[Hình 4-6. Giao diện tìm khách sạn 51](#_Toc516214782)

[Hình 4-7. Giao diện tìm nhà trọ 51](#_Toc516214783)

[Hình 4-8. Giao diện đăng bài 52](#_Toc516214784)

[Hình 4-9. Giao diện tổng quát của từng khách sạn 52](#_Toc516214785)

[Hình 4-10. Chi tiết hình ảnh của khách sạn 53](#_Toc516214786)

[Hình 4-11. Chi tiết thông tin, vị trí của khách sạn 53](#_Toc516214787)

[Hình 4-12. Người dùng thêm nhận xét về khách sạn 54](#_Toc516214788)

[Hình 4-13. Hiển thị các nhận xét của mọi người 54](#_Toc516214789)

[Hình 5-1. Trang chủ tìm khách sạn 56](#_Toc516214790)

[Hình 5-2. Trang chủ tìm nhà trọ 56](#_Toc516214791)

[Hình 5-3. Kết quả tìm kiếm khách sạn 57](#_Toc516214792)

[Hình 5-4. Hình ảnh của khách sạn 57](#_Toc516214793)

[Hình 5-5. Vị trí của khách sạn trên bản đồ 58](#_Toc516214794)

[Hình 5-6. Thêm nhận xét cho khách sạn 58](#_Toc516214795)

[Hình 5-7. Hiển thị các nhận xét gần nhất 59](#_Toc516214796)

[Hình 5-8. Kết quả tìm kiếm nhà trọ 59](#_Toc516214797)

[Hình 5-9. Khi click vào xem chi tiết 60](#_Toc516214798)

[Hình 5-10. Màn hình tìm kiếm đầu tiên trên ứng dụng 61](#_Toc516214799)

[Hình 5-11. Màn hình thực hiện tìm kiếm 62](#_Toc516214800)

[Hình 5-12. Màn hình kết quả tìm kiếm 63](#_Toc516214801)

[Hình 5-13. Màn hình bộ lọc, sắp xếp kết quả tìm kiếm 64](#_Toc516214802)

[Hình 5-14. Màn hình hiển thị vị trí trên bản đồ 65](#_Toc516214803)

[Hình 5-15. Màn hình đăng tin nhà trọ 66](#_Toc516214804)

# **THUẬT NGỮ VÀ VIẾT TẮT**

|  |  |
| --- | --- |
| UML | Unified Modeling Language |
| HTTP | Hypertext Transfer Protocol |
| JSON | Javascript Object Notation |
| SQL | Structured Query Language |
| API | Application Programing Interface |
| HTTPs | Hypertext Transfer Protocol Secure |
| reST | REpresentational State Transfer |
| crawl data | Quá trình lấy dữ liệu từ website khác và lưu vào hệ thống của mình |

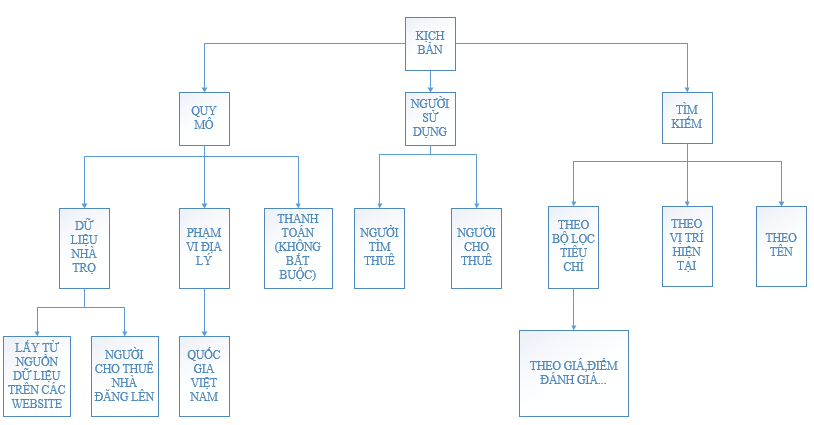
# **CHƯƠNG 1. GIỚI THIỆU**

## **1.1. Đặt vấn đề**

Trong thời đại bùng nổ công nghệ thông tin và cách mạng công nghiệp 4.0, cụ thể là Internet of Things, mọi vật đều được kết nối qua internet. Vai trò của internet là cực kì quan trọng đối với con người. Một trong những lợi ích cơ bản nhất của internet là giúp con người truy cập thông tin một cách dễ dàng nhất. Hiện nay, nhu cầu của người dùng internet là có thể có được những thông tin về một vấn đề cần thiết hoặc kết nối với nhau một cách dễ dàng, thuận tiện thông qua những website hoặc ứng dụng di động. Từ đó, những nhà phát triển phần mềm ứng dụng cần phải nắm bắt được xu thế và tạo ra những sản phẩm đúng với mong muốn của người dùng.

Hiện nay, khi người đi du lịch có thể dễ dàng tìm kiếm địa chỉ khách sạn qua các website và ứng dụng của nhiều nhà cung cấp như traveloka, trivago, booking.com, tripadvisor… Những sản phẩm trên đều là những sản phẩm có chất lượng tốt và được sử dụng rộng rãi, nhưng các sản phẩm đều hướng đến người dùng có nhu cầu ở khách sạn. Đối với sinh viên hay những người có thu nhập trung bình thì việc thuê khách sạn để ở lại vài ngày là một việc khá xa xỉ, nên họ thường có xu hướng tìm thuê các phòng trọ ở nơi mình đặt chân tới. Nhưng không phải ai cũng may mắn tìm được nhà rẻ, đặc biệt với những người du lịch nhiều nơi một lúc, họ thường đi đến những địa điểm nổi tiếng mà chưa đặt phòng hoặc tìm chỗ thuê phòng trọ trước, nên lúc tới nơi khá vất vả trong việc tìm được 1 chỗ ở ưng ý. Xuất phát từ yêu cầu đó, em và các bạn trong nhóm đã chọn đề tài "Xây dựng dịch vụ tìm nhà trọ" để giúp những người như vậy dễ dàng tìm được chỗ ở ưng ý trong thời gian du lịch.

## **1.2. Kịch bản hoạt động**



Hình 1-1. Kịch bản hoạt động của hệ thống

### *1.2.1 Quy mô*

* Dữ liệu nhà trọ: Dữ liệu nhà trọ được lấy từ trang trivago
* Người dùng có thể xem tin cho thuê nhà trọ và đăng tin thuê nhà.
* Phạm vi địa lý: Quốc gia Việt Nam.
* Thanh toán: Có thể có.

### *1.2.2 Người sử dụng*

* Người muốn thuê nhà trọ, khách sạn.
* Người muốn cho thuê nhà trọ, khách sạn.

### *1.2.3 Tìm kiếm*

* Tìm kiếm theo tên nhà trọ, khách sạn.
* Tìm kiếm theo vị trí hiện tại.
* Tìm kiếm theo tiêu chí lọc, sắp xếp: theo giá, theo điểm đánh giá …

## **1.3. Phân công công việc**

Đồ án này được hoàn thành bởi một nhóm gồm 2 thành viên, trong đó có em, thành viên còn lại tên là Trần Quang Huy, lớp ĐTTT03 – k58.

Sau đây là bảng phân công công việc của nhóm:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Task group** | **Task name** | **Person In Charge** |
|
|
| 1 | Lập kế hoạch | Lập nhóm, đề xuất ý tưởng và tổng hợp các đề tài | Cả nhóm |
| 2 | Đánh giá, lựa chọn đề tài | Cả nhóm |
| 3 | Xác định quy mô dự án | Huy |
| 4 | Xây dựng và tạo bảng kế hoạch | Tú |
| 5 | Phân công công việc | Tú |
| 6 | Phân tích | Phân tích các yêu cầu chức năng, phi chức năng | Cả nhóm |
| 7 | Xây dựng kịch bản sử dụng (use case) | Tú |
| 8 | Đặc tả các use case | Tú |
| 9 | Thiết kế CRC cards | Huy |
| 10 | Xây dựng biểu đồ lớp (Class diagram) | Tú |
| 11 | Xây dựng biểu đồ hoạt động (Activity diagram) | Huy |
| 12 | Xây dựng biểu đồ tuần tự (Sequence diagram) | Huy |
| 13 | Xây dựng mô hình thực thể và liên kết (Entity Relationship Model) | Huy |
| 14 | Thiết kế | Thiết kế sơ đồ khối của hệ thống | Huy |
| 15 | Thiết kế cơ sở dữ liệu | Tú |
| 16 | Thiết kế giao diện người dùng | Cả nhóm |
| 17 | Triển khai | Crawl dữ liệu | Tú |
| 18 | Xây dựng website backend | Tú |
| 19 | Xây dựng webservice | Tú |
| 20 | Xây dựng ứng dụng android | Huy |

Tóm tắt chương:

Chương này đã giới thiệu sơ qua về đề tài, quy mô của đề tài và phân công công việc giữa các thành viên trong nhóm.

# **CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

Sau đây là một số công nghệ, công cụ được sử dụng trong hệ thống.

## **2.1. Hệ quản trị cơ sở dữ liệu MySQL**



Hình 2-1. Hệ quản trị cơ sở dữ liệu MySQL

Đa số các ứng dụng ngày nay đều phải thao tác tới cơ sở dữ liệu – nơi lưu trữ toàn bộ dữ liệu của hệ thống. Có nhiều cách lưu trữ dữ liệu khác nhau, trong đó có cách lưu theo dạng các bảng quan hệ. MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở lớn nhất trên thế giới, sử dụng ngôn ngữ truy vấn có cấu trúc (SQL). Lý do khiến MySQL rất phổ biến là bởi vì: tốc độ truy vấn nhanh, ổn định, dễ sử dụng, tính bảo mật cao, có tính khả chuyển, nhẹ, dễ dàng cài đặt, hoạt động trên nhiều hệ điều hành, được hỗ trợ bởi nhiều ngôn ngữ lập trình và cung cấp một hệ thống lớn các hàm tiện ích rất mạnh [[1].](#tltk1)

## **2.2. Spring Boot và Spring Security**



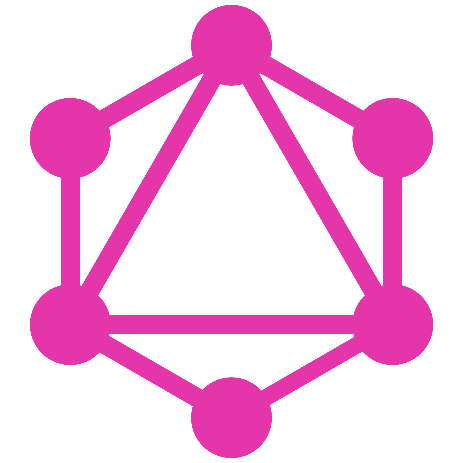
Hình 2-2. Framework Spring boot

Spring là một framework mã nguồn mở rất phổ biến để xây dựng các ứng dụng doanh nghiệp viết bằng ngôn ngữ Java. Spring Framework là một tập hợp của nhiều dự án con, chẳng hạn như Spring MVC, Spring Boot, Spring Data, Spring Clound, Spring Security…

Spring Boot là điểm bắt đầu để xây dựng các ứng dụng dựa trên Spring. Spring boot được thiết kế để giúp lập trình viên có thể xây dựng và chạy ứng dụng nhanh nhất có thể, với việc tối thiểu hóa các cấu hình [[2].](#tltk2)

Spring Security tập trung vào việc xác thực và ủy quyền trong các ứng dụng viết bằng ngôn ngữ Java. Spring Security giúp bảo vệ ứng dụng khỏi các tấn công như: session fixation, clickjacking, cross site request forgery… [[3]](#tltk3)

## **2.3. GraphQL**



Hình 2-3. GraphQL

Được Facebook giới thiệu cùng với Relay tại React.js Conf 2015, GraphQL là một ngôn ngữ truy vấn dữ liệu được tạo ra để miêu tả các phụ thuộc dữ liệu phức tạp, lồng nhau của các ứng dụng hiện đại [[4]](#tltk4).

GraphQL ra đời để thay thế cho RESTful API bởi sự hiệu quả, mạnh mẽ và linh hoạt hơn. Trước đây, khi client muốn lấy dữ liệu từ phía server, cần phải thông qua một webservice. REST là một loại web service rất phổ biến, tuy nhiên việc trả dữ liệu về cho client lại quá cồng kềnh hoặc quá nhiều end-point, khiến cho việc thiết kế rất phức tạp. GraphQL ra đời để xóa bỏ nhược điểm này. Nếu như trước đây, client chỉ muốn lấy một vài giá trị của một đối tượng nào đó từ phía server, thì REST vẫn trả về tất cả các dữ liệu liên quan đến đối tượng đó cho client, và client chỉ lấy những thứ mình cần, còn lại thì bỏ đi! Như vậy việc truyền tải nhiều thông tin dư thừa như vậy rõ rảng không hiệu quả chút nào. GraphQL thì khác phục nhược điểm này bằng cách chỉ trả về cho phía client đúng dữ liệu theo định dạng mà client đã gửi tới, giúp việc truyền tải nhanh hơn và không lãng phý.

Sau đây là 1 số cú pháp viết câu lệnh truy vấn trong GraphQL:

### *2.3.1. Query*

Hoạt động truy vấn trong GraphQL có cú pháp như sau [[5]](#tltk5):

query {

object (argument:”arg\_val”) {

parameter1

parameter2

}

}

Trong đó object là đối tượng cần lấy dữ liệu, các parameter là các tham số của nó, argument là một đối số, dữ liệu trả về dưới dạng JSON

{

“data”:{

“object”:{

“parameter1”:”value1”

“parameter2”:”value2”

}

}

}

Truyền biến số vào hoạt động query:

Biến số:

{

“variable”:”var\_val”

}

Câu lệnh query như sau:

query OporationName($variable: Variable) {

Object(argument: $variable) {

parameter

}

}

### *2.3.2. Mutation*

Những thay đổi dữ liệu bên server sẽ được thực hiện qua mutation. Câu lệnh mutation như sau [[6]](#tltk6):

Câu lệnh khai báo biến số:

{

“parameter1”: ”val1”,

“parameter2”: {

“parameter3”:”val2”

“parameter4”:”val3”

}

}

Câu lệnh mutation như sau:

mutation CreateParameterForObject(

$variable1: Variable1!, $variable2: Variable2!){

createNew(parameter1: $variable1, parameter2: $ variable2) {

parameter3

parameter4

}

}

Kết quả trả về:

{

“data”: {

“createNew”: {

“parameter3”:”val2”

“parameter4”:”val3”

}

}

}

### *2.3.3. Schema và Type*

* Type:

Type là đối tượng của GraphQL, mô tả đối tượng JSON trả về.

Cấu trúc 1 type cơ bản như sau:

Type Type {

field1: object1!

field2: [object2]!

}

Type là kiểu đối tượng GraphQL, là kiểu với một số trường. Đa số các trường là kiểu đối tượng.

field1, field2 là các trường thuộc kiểu Type. Câu lệnh query trên kiểu Type chỉ có thể xuất hiện các trường này.

Object1! Là đối tượng, dấu chấm than là quy định trường không được có giá trị null.

[Object2]! Là mảng các đối tượng.

* Kiểu query và mutation

Một schema luôn có kiểu query, có hoặc không có kiểu mutation.

schema {

query: Query

mutation: Mutation

}

Với một câu lệnh query cần một kiểu Query tương ứng.

## **2.4. Google map API**

### *2.4.1. Giới thiệu về google map*

Google Maps là một ứng dụng bản đồ trực tuyến trên web được cung cấp miễn phí bởi Google. Nó có thể dùng để nhúng vào các trang web của bên thứ ba thông qua Google Maps API. Nó cung cấp rất nhiều những tiện ích nâng cao giúp người dùng tìm kiếm đường đi,các địa điểm, các địa danh... Google phát triển Google Maps API dành cho 4 mảng chính đó là:

* Các ứng dụng trên Android
* Các ứng dụng trên IOS
* Các ứng dụng web
* Các ứng dụng webservice

Hàng triệu trang web và ứng dụng sử dụng Google Maps API để cung cấp kinh nghiệm tìm kiếm vị trí cho người dùng của họ.Ngoài ra google map API còn hoạt động tốt trên các thiết bị di động như IOS, Android...

### *2.4.2. Giới thiệu về google map API*

Người dùng có thêm thêm bản đồ dựa trên dữ liệu Google Maps vào ứng dụng của mình. API tự động xử lý truy cập đến Google Maps servers, dữ liệu tải,dữ liệu hiển thị, và phản hồi những thay đổi của bản đồ. Người dùng cũng có thể dùng API để thêm điểm đánh dấu, hình vẽ cho một bản đồ cơ bản, thay đổi góc nhìn bản đồ của một khu vực cụ thể. Những đối tượng này cung cấp thêm thông tin cho bản đồ, đồng thời cho phép người dùng tương tác với bản đồ [[7]](#tltk7).

### *2.4.3. Giới thiệu về google place API*

Google place API cho phép truy vấn và tìm các thông tin trên bản đồ theo nhiều thể loại như: tìm nhà hàng, khách sạn, trường học, các địa điểm du lịch… Với google place API ta có thể tìm các địa điểm bằng những từ khóa thân thiện và gần gũi. Kết quả trả về của quá trình tìm kiếm là 1 chỉ dẫn và giới thiệu ngắn gọn về địa điểm và chi tiết về địa điểm đó [[8]](#TLTK8).

Một số tính năng quan trọng của place API:

* Place picker: người dùng có thể chọn địa điểm trên bản đồ.
* Place autocomplete: tự động điền tên và địa chỉ theo nội dung nhập của người dùng.
* Place IDs and Details: trả về thông tin chi tiết của địa điểm.

### *2.4.4. Giới thiệu về google place API*

Google place web service là dịch vụ trả về thông tin về địa điểm sử dụng yêu cầu HTTP. Những địa điểm được định nghĩa trong API này là những tổ chức, địa điểm địa lý … [[9]](#TLTK9)

Một số dịch vụ yêu cầu trên web service:

* Place Search: trả về danh sách địa điểm dựa trên vị trí người dùng hoặc từ khóa tìm kiếm.
* Place Details: trả về thông tin chi tiết về địa điểm
* Place Autocomplete: trả về kết quả là địa điểm, địa chỉ gợi ý có kí tự giống với kí tự người dùng nhập vào.
* Query Autocomplete: dịch vụ dự đoán địa chỉ, địa điểm dựa vào hạng mục người dùng nhập, ví dụ người dùng nhập “Nhà hàng tại Hà Nội”, kết quả trả về là những nhà hàng ở Hà Nội.

Ứng dụng sử dụng dịch vụ Place autocomplete để hiển thị kết quả gợi ý khi người dùng nhập tên địa điểm. Một yêu cầu https Place autocomplete có dạng như sau:

https://maps.googleapis.com/maps/api/place/autocomplete/json?input=YOUR\_STRING&key=YOUR\_API\_KEY

Trong đó YOUR\_STRING là chuỗi kí tự người dùng nhập vào, YOUR\_API\_KEY là API key của ứng dụng. Kết quả trả về là mảng đối tượng json “prediction”.

### *2.4.5. Google Direction API*

Google location API là dịch vụ tính toán đường đi giữa các địa điểm sử dụng yêu cầu HTTP. Với Direction API, người dùng có thể:

* Tìm đường đi ở nhiều cách di chuyển như sử dụng ô tô, đi bộ, đi xe đạp …
* Nhận được kết quả nhiều cách đi khác nhau.
* Địa điểm được thể hiện dưới dạng tên, tọa độ địa lý, ID của địa điểm.

Ứng dụng sử dụng Direction API để tính toán khoảng cách, thời gian giữa 2 địa điểm, từ đó đưa ra kết quả cho tùy chọn lọc địa điểm. Một yêu cầu https Direction API có dạng như sau:

http://maps.googleapis.com/maps/api/directions/json?origin=DIEM\_XUAT\_PHAT&destination=DICH\_DEN&key=YOUR\_API\_KEY

Trong đó DIEM\_XUAT\_PHAT là địa điểm xuất phát, DICH\_DEN là địa điểm đến, YOUR\_API\_KEY là API key của ứng dụng. Khoảng cách được trả về trong thuộc tính “distance”, thời gian trong thuộc tính “duration” [[10]](#TLTK10).

### *2.4.6. Google Geocoding API*

Google Geocoding API cung cấp 2 dịch vụ:

* Geocoding: chuyển địa chỉ của địa điểm sang tọa độ địa lý.
* Reverse geocoding: chuyển tọa độ địa lý sang địa chỉ của địa điểm.

Ứng dụng sử dụng Geocoding API để xác đinh tọa độ địa lý của địa điểm, từ đó hiển thị lên bản đồ. Một yêu cầu https Geocoding API có dạng như sau:

https://maps.googleapis.com/maps/api/geocode/json?address=YOUR\_ADDRESS&key=YOUR\_API\_KEY

Trong đó YOUR\_ADDRESS là địa chỉ cần chuyển sang tọa độ địa lý và YOUR\_API\_KEY là API key của ứng dụng. Kết quả trả về dưới dạng file JSON chứa vĩ độ, tung độ ở 2 thuộc tính “lat”, “lng” [[11]](#TLTK11).

## **2.5. Selenium WebDriver**

### *2.5.1. Giới thiệu*

Selenium WebDriver là công cụ phần mềm giúp thực thi các hành động lên trang web một cách tự động theo yêu cầu của người lập trình. Mục đích chính của công cụ là để thực thi những phép thử tự động.

Do Selenium WebDriver có khả năng giả lập sử dụng trình duyệt web, nhóm em sẽ sử dụng WebDriver này để tự động các thao tác truy cập website, từ đó lấy dữ liệu từ những website đó (crawl data).

### *2.5.2. Sử dụng Selenium WebDriver*

Câu lệnh cơ bản sử dụng Selenium để lấy những thành phần HTML như sau:

element = driver.findElement(option);

Trong đó:

* element là đối tượng WebElement, là đối tượng thành phần HTML được cung cấp bởi Selenium.
* Driver là đối tượng WebDriver.
* Option gồm các lựa chọn sau:
* By.name(): Tìm phần tử dựa theo tên
* By.xpath(): tìm phần tử dựa vào cấu trúc xpath

Cấu trúc xpath như sau:

Xpath=//tagname[@attribute='value']

Trong đó:

* //: lựa chọn phần tử có tagname tương ứng trong toàn bộ trang.
* tagname: tên node hiện tại(input, div, img…).
* attribute: thuộc tính(id, class, …).
* .//: lựa chọn phần từ có tagname tương ứng từ node hiện tại hoặc dưới nó.
* By.classname(): tìm phần tử theo classname
* By.id(): tìm phần tử theo id

Phương thức lấy dữ liệu: getAttribute(attribute)

Tóm tắt chương:

Chương này đã đề cập đến những công cụ, công nghệ được sử dụng trong quá trình thực hiện đề tài, bao gồm: hệ quản trị MySQL, framework Spring boot, webservice GraphQL, google APIs, Selenium web driver.

# **CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG**

## **3.1. Yêu cầu chức năng**

Yêu cầu chức năng của hệ thống như sau:

* Có chức năng xác định vị trí của người dùng và gợi ý các địa điểm nhà trọ, khách sạn, ở các khu vực xung quanh.
* Tìm kiếm khách sạn, nhà trọ dựa theo các tiêu chí như: tên, địa điểm, khoảng giá phòng, loại phòng và các dịch vụ khác…(Tìm kiếm nâng cao)
* Hệ thống có chức năng cho phép người dùng xem thông tin khách sạn, phòng, dịch vụ…được chọn.
* Hệ thống có chức năng chỉ đường tới nhà nghỉ, khách sạn được chọn lựa (optional).
* Có chức năng cho phép người dùng đánh giá chất lượng dịch vụ của nhà trọ, khách sạn (yêu cầu người dùng đăng nhập).
* Hệ thống cho phép người quản trị quản lý các dữ liệu về nhà trọ, phòng, dịch vụ, người dùng truy cập hệ thống.
* Hệ thống có chức năng thống kê, báo cáo lượt truy cập…

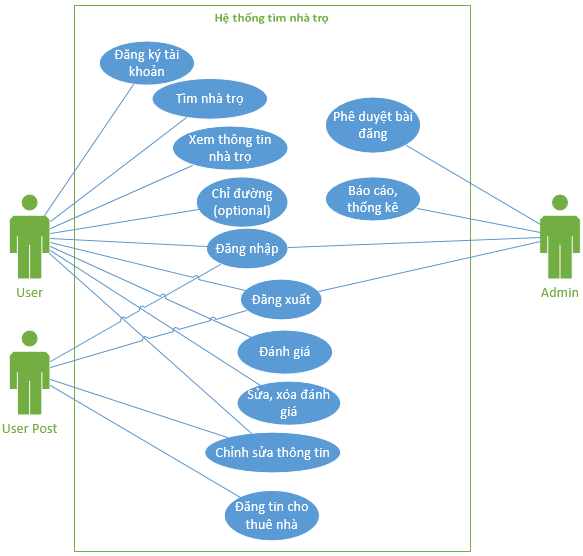
## **3.2. Yêu cầu phi chức năng**

Yêu cầu phi chức năng của hệ thống như sau:

* Hệ thống có giao diện thân thiện, dễ sử dụng, đảm bảo phục vụ được số lượng lớn người dùng truy cập đồng thời.
* Có thể chạy trên nhiều nền tảng khác nhau (ứng dụng Android hoặc trên web)
* Không xung đột với các ứng dụng khác
* Dễ dàng bảo trì, cập nhật

## **3.3. Sơ đồ use case của hệ thống (Use case Diagram)**

Sau đây là sơ đồ use case tổng quát của hệ thống

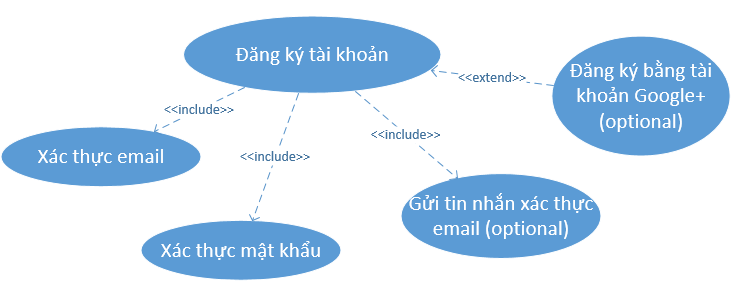


Hình 3-1. Sơ đồ use case tổng quát của hệ thống tìm kiếm nhà trọ

Hệ thống này gồm cho 3 tác nhân chính là User (người sử dụng), Admin (quản trị hệ thống) và User Post (người đăng tin cho thuê nhà) trong đó User là tác nhân chính.

## **3.4. Sơ đồ phân rã use case**

### *3.4.1. Use case đăng ký tài khoản*

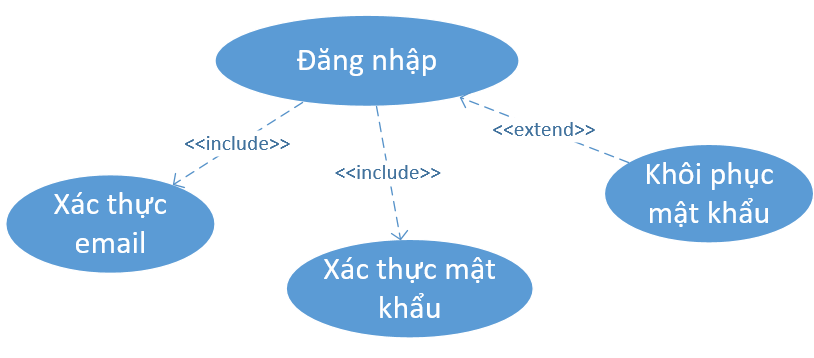


Hình 3-2. Phân rã use case đăng ký

* Đặc tả use case:
* Tên use case: Đăng ký tài khoản
* Actor: người dùng (user) chưa có tải khoản
* Mục tiêu: Cho phép người dùng đăng ký tài khoản để sử dụng các chức năng: đăng bài, nhận xét… của hệ thống
* Điều kiện xảy ra (trigger): người dùng click vào nút "đăng ký" trên website hoặc trên ứng dụng
* Điều kiện tiên quyết: người dùng phải vào trang web hoặc ứng dụng
* Mô tả chi tiết:
* Use case đăng ký bao gồm 3 use case con như trên hình vẽ. Hệ thống cần phải xác thực email, mật khẩu và gửi tin nhắn xác nhận email thì việc đăng ký tài khoản mới thành công.
* Use case đăng ký cho phép người dùng tạo một tài khoản mới. Khách hàng cần có một địa chỉ email để hệ thống có thể xác thực. Sau khi nhập email và mật khẩu (mật khẩu cần nhập 2 lần) thì hệ thống sẽ check xem email đó đã được dùng để đăng ký lần nào chưa. Nếu chưa thì một tin nhắn sẽ được gửi đến địa chỉ email của người dùng, yêu cầu người dùng click vào để kích hoạt tài khoản, lúc này việc tạo tài khoản thành công. Nếu email đó đã có người dùng để tạo tài khoản trước đó rồi, thì hệ thống sẽ thông báo email đã tồn tại.
* Mật khẩu phải nhập lần thứ 2 để xác minh lại, đề phòng trường hợp người dùng nhập không chính xác.
* Người dùng có thể sử dụng tài khoản Google+ để đăng ký (optional). Trường hợp này người dùng không cần nhập mật khẩu, do đó sẽ tiện lợi và nhanh chóng hơn.

### *3.4.2. Use case đăng nhập*

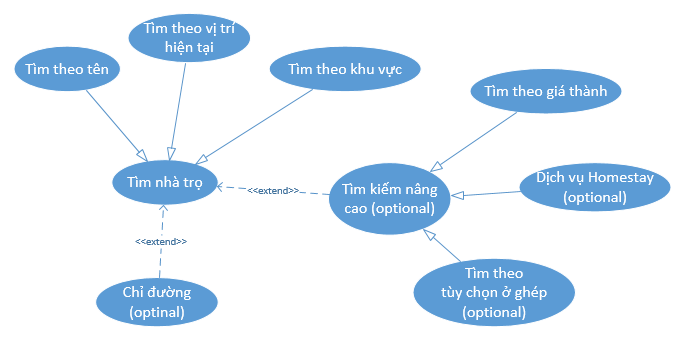
Nếu người dùng muốn nhận xét, đánh giá, hay đăng bài thì cần phải đăng nhập. Sau đây là sơ đồ use case chi tiết của quá trình đăng nhập



Hình 3-3. Phân rã use case đăng nhập

* Đặc tả use case:
* Tên use case: Đăng nhập
* Actor: người dùng đã có tải khoản
* Mục tiêu: Cho phép người dùng đăng nhập để sử dụng các chức năng: đăng bài, nhận xét… của hệ thống
* Điều kiện xảy ra (trigger): người dùng click vào nút "đăng nhập" trên website hoặc trên ứng dụng, hoặc khi người dùng muốn đăng bài hoặc bình luận mà chưa đăng nhập
* Điều kiện tiên quyết: người dùng phải vào trang web hoặc ứng dụng
* Sau đây là mô tả chi tiết:
* Nếu người dùng đã đăng ký tải khoản rồi thì có thể dùng tài khoản đó để đăng nhập. Người dùng cần nhập chính xác địa chỉ email đã dùng để đăng ký và mật khẩu.
* Nếu người dùng quên mật khẩu có thể dùng địa chỉ email trên để khôi phục lại tài khoản.
* Nếu người dùng chưa có tài khoản thì phải đăng ký trước rồi mới đăng nhập.

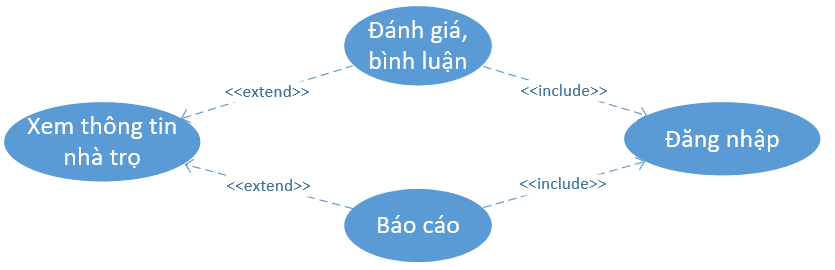
### *3.4.3. Use case tìm kiếm nhà trọ*



Hình 3-4. Phân rã use case tìm nhà trọ

* Đặc tả use case:
* Tên use case: tìm khách sạn/nhà trọ
* Actor: người dùng bất kỳ
* Mục tiêu: Cho phép người dùng tìm kiếm khách sạn/nhà trọ theo ý mình
* Điều kiện xảy ra (trigger): người dùng nhập tên, địa điểm vào ô tìm kiếm vào ấn nút "tìm" trên website hoặc ứng dụng
* Điều kiện tiên quyết: người dùng phải vào trang tìm kiếm của website hoặc ứng dụng
* Sau đây là mô tả chi tiết:
* Use case "tìm nhà trọ" là 1 use case trừu tượng, có 3 use case cụ thể kế thừa từ nó. 3 use case đó mô tả 3 tiêu chí để người dùng tìm kiếm nhà trọ: dựa vào vị trí hiện tại của họ, dựa theo tên nhà trọ hoặc dựa theo khu vực.
* Phần tìm kiếm nâng cao (đây cũng là một use case trừu tượng), người dùng có thể tìm nhà trọ theo mức giá, tìm nhà trọ muốn ở ghép hoặc các dịch vụ homestay (dịch vụ mà du khách sẽ ở trọ trong chính nhà của một người dân bản địa, và sinh hoạt như một thành viên trong gia đình) (optional).
* Sau khi tìm kiếm, hệ thống có thể chỉ đường tới nhà trọ từ vị trí hiện tại nếu muốn (optional). Use case này mở rộng từ use case "tìm nhà trọ", nên nó có thể được thực hiện hoặc không.

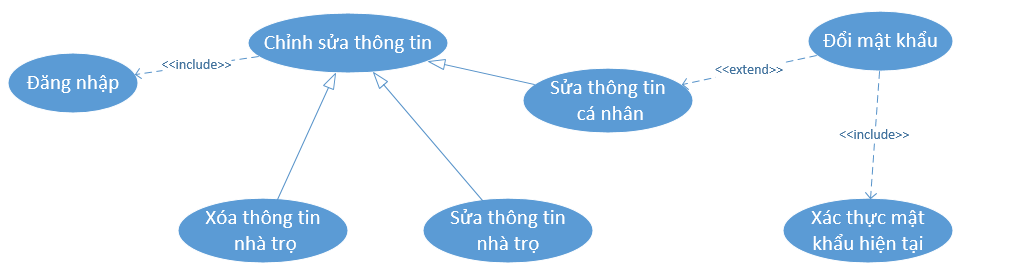
### *3.4.4. Use case xem thông tin nhà trọ*



Hình 3-5. Phân rã use case xem thông tin nhà trọ

* Đặc tả use case:
* Tên use case: xem thông tin khách sạn/nhà trọ
* Actor: người dùng bất kỳ
* Mục tiêu: Cho phép người dùng xem chi tiết thông tin về khách sạn, nhà trọ mà họ quan tâm
* Điều kiện xảy ra (trigger): người dùng click vào khách sạn/nhà trọ bất kì trên website hoặc trên ứng dụng
* Điều kiện tiên quyết: người dùng phải tìm kiếm khách sạn/nhà trọ và hệ thống phải trả về danh sách các khách sạn/nhà trọ mà người dùng muốn
* Sau đây là mô tả chi tiết:
* Người dùng có thể xem các thông tin liên quan đến nhà trọ như: diện tích, chi phí thuê, vị trí, nhận xét của người khác…
* Người dùng cũng có thể đánh giá, bình luận (không bắt buộc) về chất lượng của nhà trọ. Việc này giúp người khác dễ dàng đánh giá được chất lượng nhà trọ khi tìm kiếm. Nhưng nếu muốn làm điều này thì người dùng cần phải đăng nhập, do đó use case này sẽ include use case đăng nhập.
* Người dùng có thể báo cáo nhà trọ có thông tin sai sự thật, hoặc spam với admin. Use case này cũng yêu cầu người dùng phải đăng nhập.

### *3.4.5. Use case sửa thông tin*



Hình 3-6. Phân rã use case sửa thông tin

* Đặc tả use case:
* Tên use case: sửa thông tin
* Actor: người dùng đã có tài khoản
* Mục tiêu: Cho phép người dùng sửa đổi thông tin cá nhân của họ
* Điều kiện xảy ra (trigger): người dùng click vào mục "sửa thông tin" trên website hoặc trên ứng dụng
* Điều kiện tiên quyết: người dùng đã có tài khoản rồi
* Sau đây là mô tả chi tiết:
* Use case chỉnh sửa thông tin là 1 use case trừu tượng, có 3 use case cụ thể kế thừa từ nó. Use case này yêu cầu đăng nhập được thực hiện trước, do đó nó include use case đăng nhập.
* Sửa thông tin cá nhân: thông tin cá nhân của người dùng gồm họ tên và mật khẩu. Nếu người dùng muốn đổi mật khẩu thì bắt buộc phải nhập mật khẩu cũ để xác thực.
* Ngoài ra người dùng còn có thể sửa thông tin về nhà trọ mà họ đăng lên.

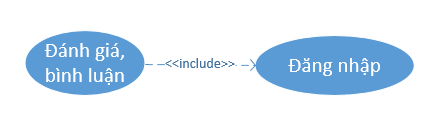
### *3.4.6. Use case chỉ đường*

* Đặc tả use case:
* Tên use case: chỉ đường
* Actor: người dùng
* Mục tiêu: Cho phép người dùng tìm đường từ vị trí hiện tại (hoặc từ bất kỳ chỗ nào) tới khách sạn mà người dùng vừa tìm kiếm
* Điều kiện xảy ra (trigger): người dùng click vào nút "chỉ đường" sau khi xem chi tiết thông tin của khách sạn
* Điều kiện tiên quyết: người dùng phải xem chi tiết thông tin của khách sạn

### *3.4.7. Use case đăng xuất*

* Đặc tả use case:
* Tên use case: đăng xuất
* Actor: người dùng đã có tài khoản
* Mục tiêu: Cho phép người dùng đăng xuất khỏi hệ thống
* Điều kiện xảy ra (trigger): người dùng click vào nút "đăng xuất" trên website hoặc trên ứng dụng
* Điều kiện tiên quyết: người dùng phải đã đăng nhập rồi

### *3.4.8. Use case đánh giá, phản hồi*



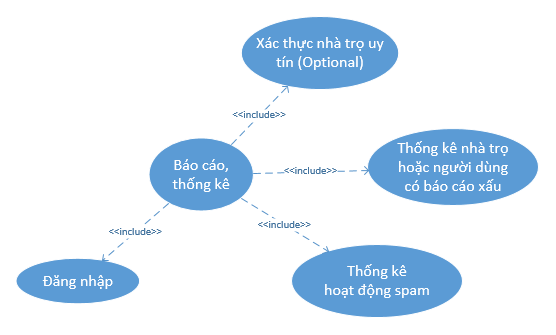
Hình 3-7. Phân rã use case đánh giá, phản hồi

* Đặc tả use case:
* Tên use case: đánh giá, phản hồi
* Actor: người dùng đã có tài khoản
* Mục tiêu: Cho phép người dùng đánh giá, phản hồi, bình luận về chất lượng dịch vụ của khách sạn mà người dùng muốn
* Điều kiện xảy ra (trigger): người dùng click vào nút "bình luận" trên website hoặc trên ứng dụng sau khi đã nhập đủ nội dung cần bình luận
* Điều kiện tiên quyết: người dùng phải đăng nhập trước

### *3.4.9. Use case xóa bình luận*

* Đặc tả use case:
* Tên use case: xóa bình luận
* Actor: người dùng đã có tài khoản
* Mục tiêu: Cho phép người dùng xóa bình luận, nhận xét của mình đối với khách sạn mà họ đã đánh giá trước đó
* Điều kiện xảy ra (trigger): người dùng click vào nút "xóa nhận xét" trên website hoặc trên ứng dụng
* Điều kiện tiên quyết: người dùng phải đã đăng nhập rồi

### *3.4.10. Use case báo cáo thống kê*



Hình 3-8. Phân rã use case báo cáo thống kê

* Đặc tả use case:
* Tên use case: báo cáo, thống kê (optional)
* Actor: admin
* Mục tiêu: Cho phép admin thống kê về hệ thống như: số lượng người dùng truy cập trang web, số lượng các bài đăng spam, những người đăng tin xấu nhằm phá hoại trang web…
* Điều kiện xảy ra (trigger): admin vào địa chỉ báo cáo, thống kê trên website
* Điều kiện tiên quyết: người dùng phải đăng nhập với vai trò admin

### *3.4.11. Use case phê duyệt bài đăng*

* Đặc tả use case:
* Tên use case: phê duyệt bài đăng
* Actor: admin
* Mục tiêu: Cho phép admin xem xét các bài đăng tin thuê nhà mà người dùng đăng lên hệ thống, để admin phê duyệt xem bài đăng đó có hợp lệ hay không. Nếu hợp lệ thì cho phép bài đăng đó hiển thị trên hệ thống khi người dùng tìm kiếm, ngược lại thì không.
* Điều kiện xảy ra (trigger): admin vào mục phê duyệt bài đăng trên website
* Điều kiện tiên quyết: người dùng phải đăng nhập với vai trò là admin.

## **3.5. CRC Cards (Class-Responsibility-Collaborators)**

Phân tích các use case để xác định các lớp thông qua việc xác định các đối tượng có trong hệ thống.

* Phân tích use case Đăng kí tài khoản, xác định được đối tượng Người dùng.
* Phân tích use case Tìm nhà trọ, xác định được đối tượng Nhà trọ.
* Phân tích use case Đánh giá, phản hồi, xác định đối tượng Đánh giá, Phản hồi.
* Phân tích use case Đăng tin cho thuê nhà, xác định đối tượng Bài đăng.

|  |  |
| --- | --- |
| Class name : User | |
| Description : người sử dụng dịch vụ | |
| Responsibilities:   * Chỉnh sửa thông tin * Chỉnh sửa mật khẩu * Khôi phục mật khẩu | Collaborators:   * Role * Review * Report * Post |
| Attibutes :   * ID * Username * Password * Email * Full\_name * Enabled | |
| Relationships :  Generalization :  Aggregation:Role  Other association :Review, Report, Post | |

|  |  |
| --- | --- |
| Class name : Role | |
| Description : Phân loại vai trò người sử dụng | |
| Responsibilities:   * Phân loại người dùng | Collaborators:   * User |
| Attibutes :   * ID * Name | |
| Relationships :  Generalization :  Aggregation:User  Other association : | |

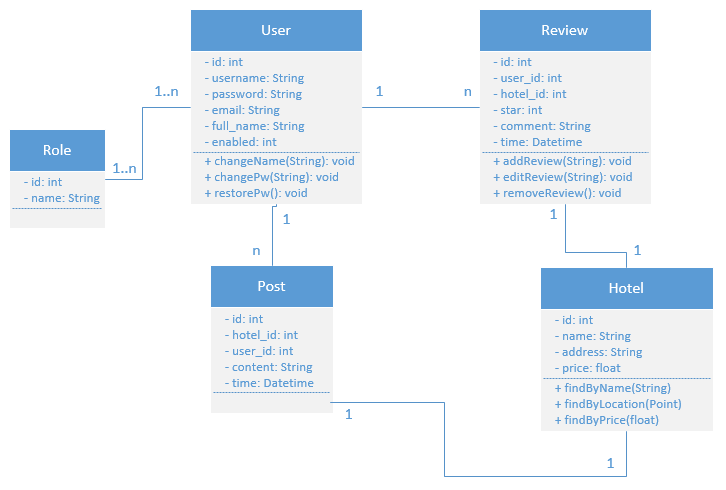
|  |  |
| --- | --- |
| Class name : Review | |
| Description : Bình luận của người sử dụng | |
| Responsibilities:   * Thêm bình luận * Sửa bình luận * Xóa bình luận | Collaborators:   * User * Hotel |
| Attibutes :   * ID * User\_ID * Hotel\_ID * Star * Comment * Time | |
| Relationships :  Generalization :  Aggregation: User  Other association : Hotel | |

|  |  |
| --- | --- |
| Class name : Post | |
| Description : bài đăng của người cho thuê nhà | |
| Responsibilities:   * Thêm bài đăng | Collaborators:   * User |
| Attibutes :   * ID * User\_id * description * time | |
| Relationships :  Generalization :  Aggregation: User  Other association : | |

|  |  |
| --- | --- |
| Class name : Hotel | |
| Description : Khách sạn, nhà trọ | |
| Responsibilities:   * Tìm theo tên * Tìm theo địa điểm * Tìm theo giá | Collaborators:   * Review |
| Attibutes :   * ID * Name * Address * Price * Area | |
| Relationships :  Generalization :  Aggregation: Review  Other association : | |

## **3.6. Biểu đồ lớp (Class Diagram)**

Sau đây là sơ đồ lớp của hệ thống



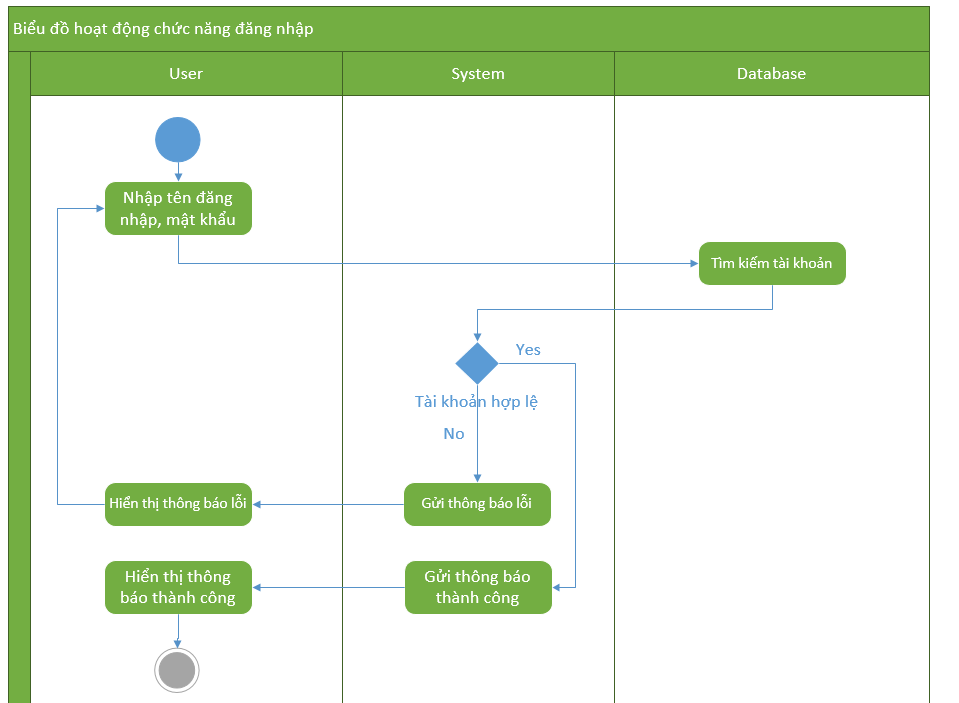
Hình 3-9. Sơ đồ lớp của hệ thống

Quan hệ giữa các lớp:

* Lớp User có quan hệ m – n với lớp Role, bởi vì một người dùng có thể có 1 hoặc 2 vai trò (ROLE\_USER hoặc/và ROLE\_ADMIN).
* Lớp User có quan hệ 1 – n với lớp Post vì mỗi 1 người dùng có thể đăng nhiều bài viết cho thuê nhà, nhưng mỗi bài viết chỉ thuộc về 1 người.
* Tương tự, lớp User có quan hệ 1 – n với lớp Review, diễn giải như trên.
* Mỗi 1 review chỉ liên quan đến 1 khách sạn, do đó 2 lớp này đều có quan hệ 1 – 1 với lớp Hotel.

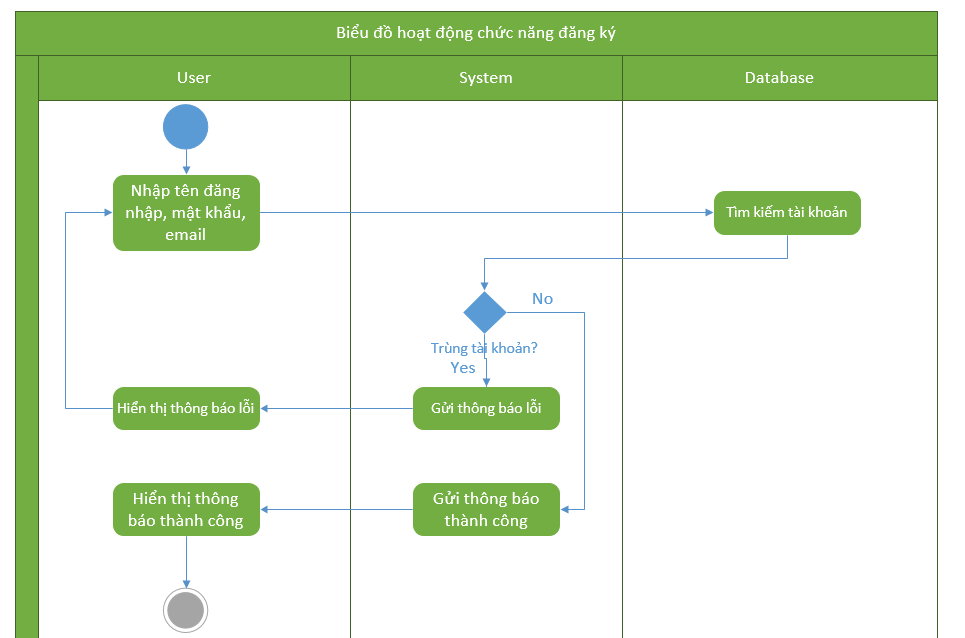
## **3.7. Biểu đồ hoạt động (Activity Diagram)**

### *3.7.1. Chức năng đăng nhập*



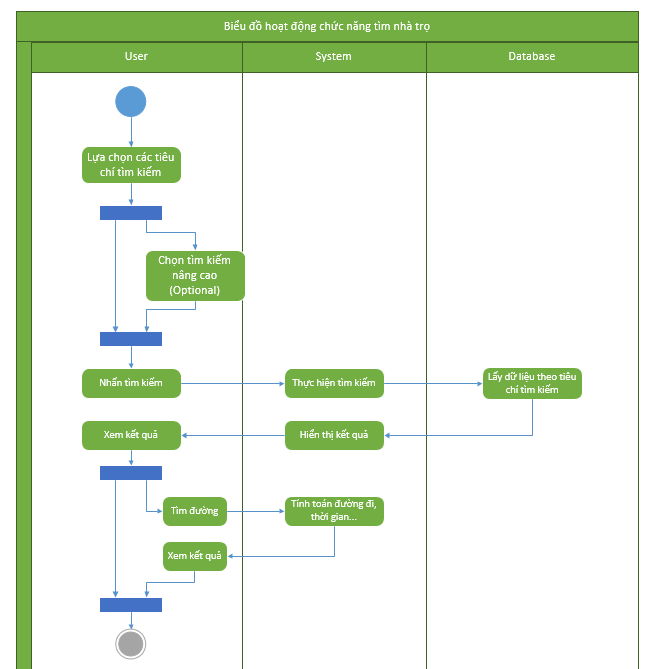
Hình 3-10. Biểu đồ hoạt động dành cho chức năng đăng nhập

### *3.7.2. Chức năng đăng ký*



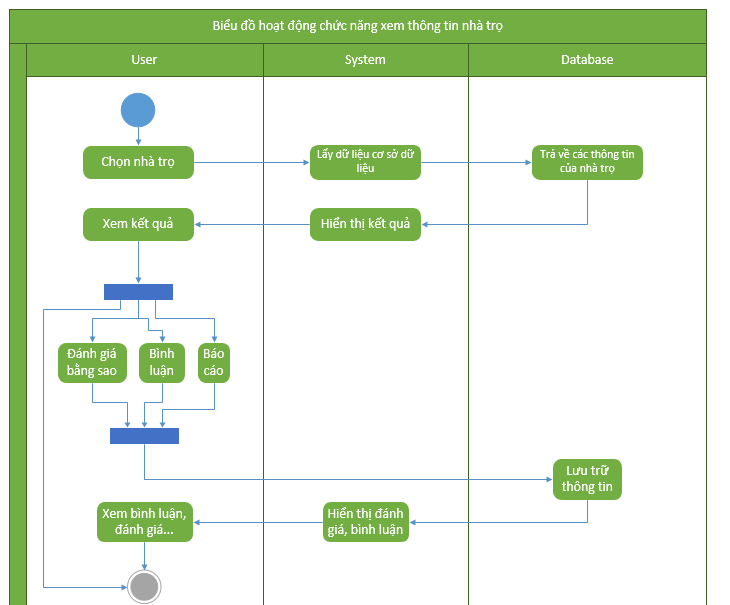
Hình 3-11. Biểu đồ hoạt động dành cho chức năng đăng ký

### *3.7.3. Chức năng tìm kiếm nhà trọ*



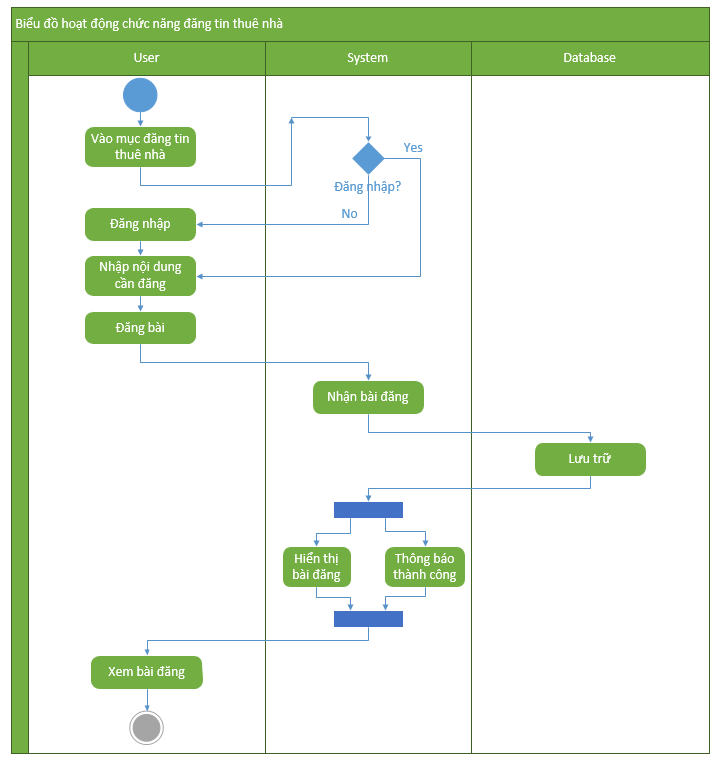
Hình 3-12. Biểu đồ hoạt động dành cho chức năng tìm nhà trọ

### *3.7.4. Chức năng xem thông tin nhà trọ*



Hình 3-13. Biểu đồ hoạt động dành cho chức năng xem thông tin nhà trọ

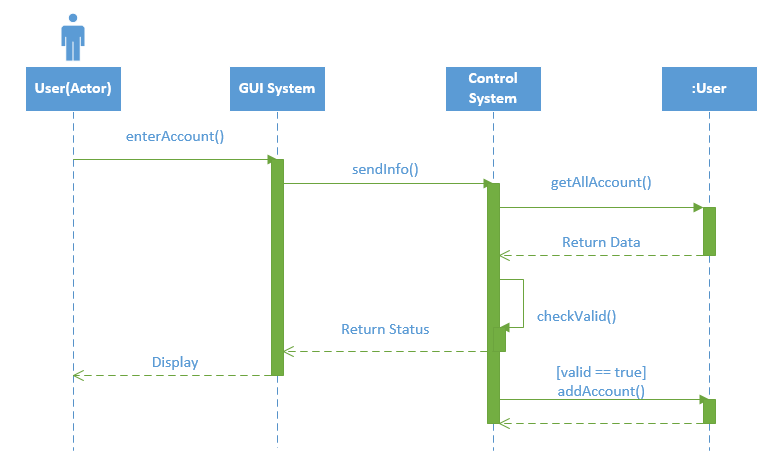
### *3.7.5. Chức năng đăng tin thuê nhà*



Hình 3-14. Biểu đồ hoạt động dành cho chức năng đăng tin thuê nhà

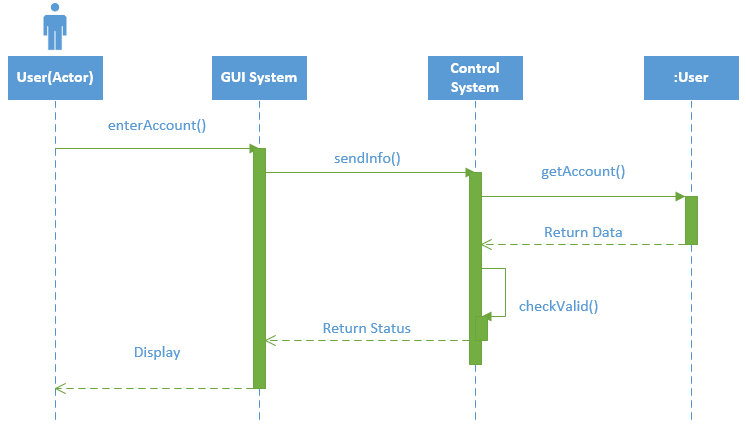
## **3.8. Biểu đồ tuần tự (Sequence diagram)**

### *3.8.1. Chức năng đăng ký*



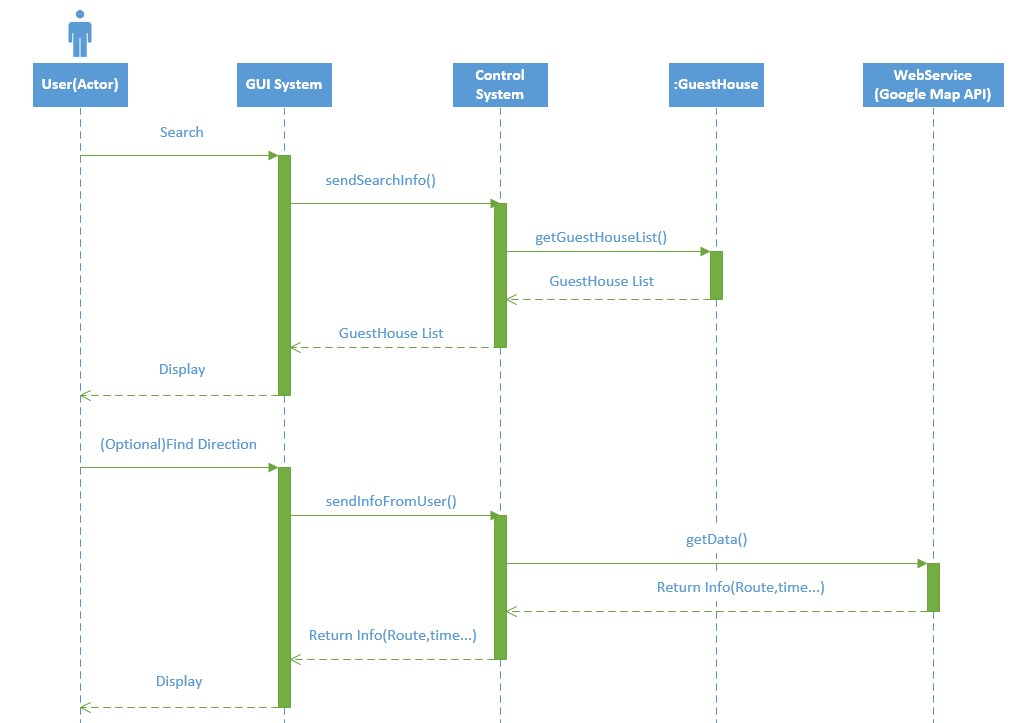
Hình 3-15. Biểu đồ tuần tự của chức năng đăng ký tài khoản

### *3.8.2. Chức năng đăng nhập*



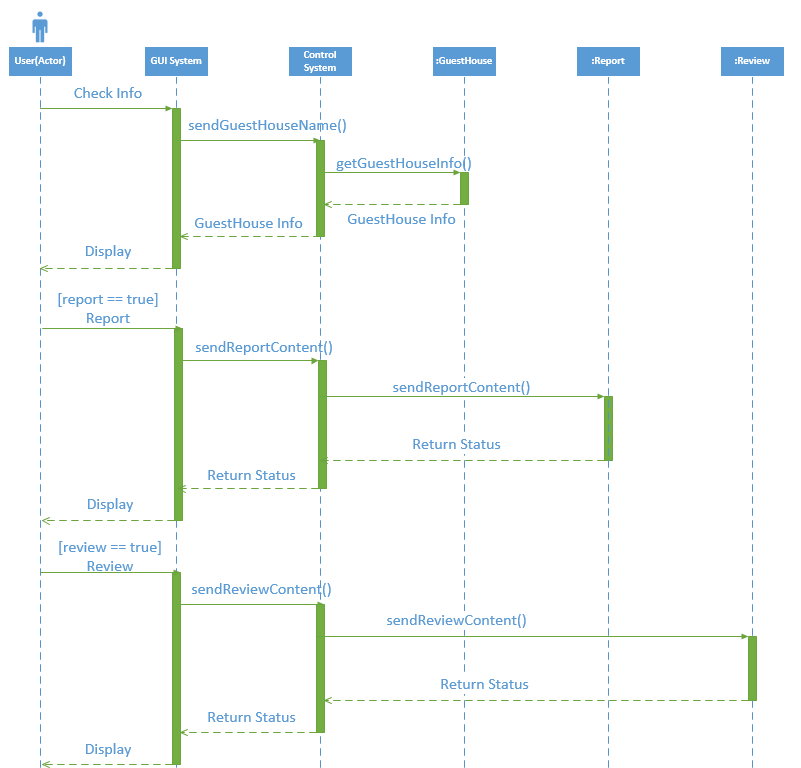
Hình 3-16. Biểu đồ tuần tự của chức năng đăng nhập

### *3.8.3. Chức năng tìm nhà trọ*



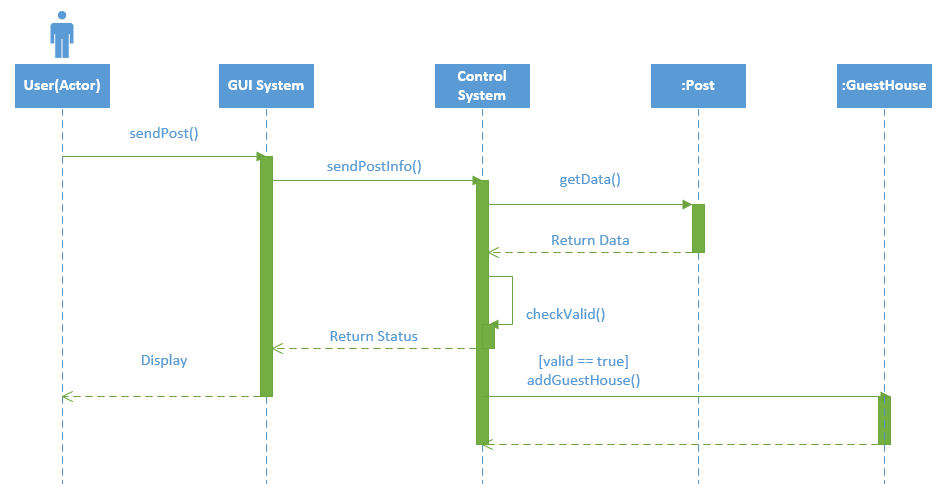
Hình 3-17. Biểu đồ tuần tự của chức năng tìm nhà trọ

### *3.8.4. Chức năng xem thông tin nhà trọ*



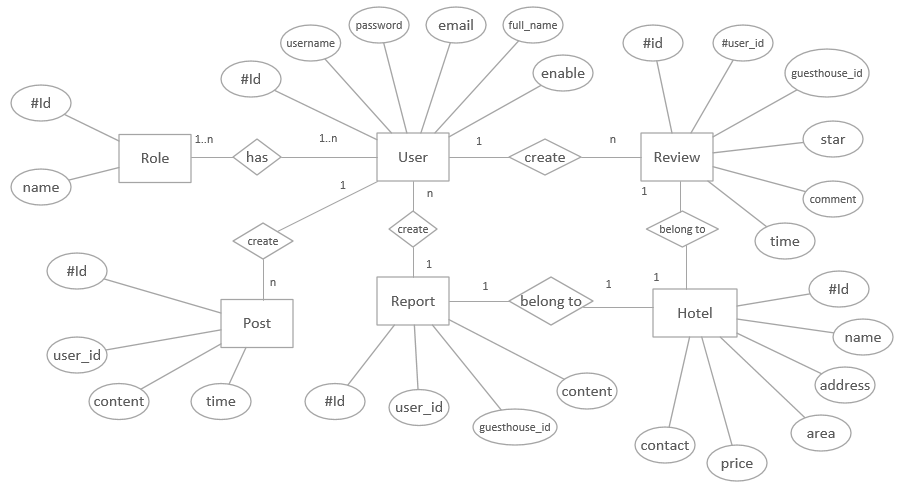
Hình 3-18. Biểu đồ tuần tự của chức năng xem thông tin nhà trọ

### *3.8.5. Chức năng đăng ký nhà trọ (đăng tin)*



Hình 3-19. Biểu đồ tuần tự của chức năng đăng ký nhà trọ

## **3.9. Mô hình thực thể liên kết**



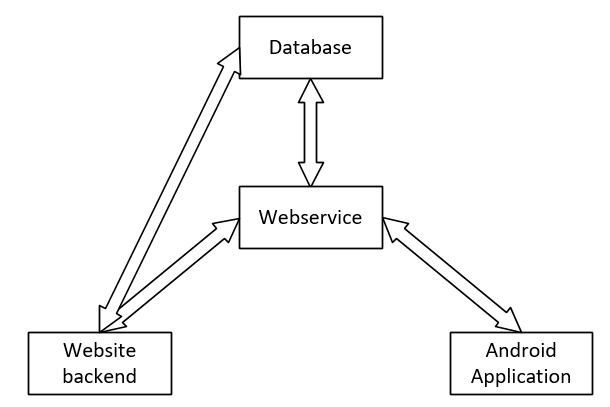
Hình 3-20. Mô hình thực thể liên kết

Tóm tắt chương:

Chương này đã phân tích hệ thống, từ việc phân tích yêu cầu chức năng và phi chức năng, sau đó tới việc phân tích các use case của hệ thống. Tiếp đó là việc xây dựng các biểu đồ lớp, biểu đồ hoạt động và biểu đồ tuần tự cho hệ thống. Cuối cùng là xây dựng mô hình thực thể liên kết.

# **CHƯƠNG 4. THIẾT KẾ HỆ THỐNG**

## **4.1. Sơ đồ khối của hệ thống**

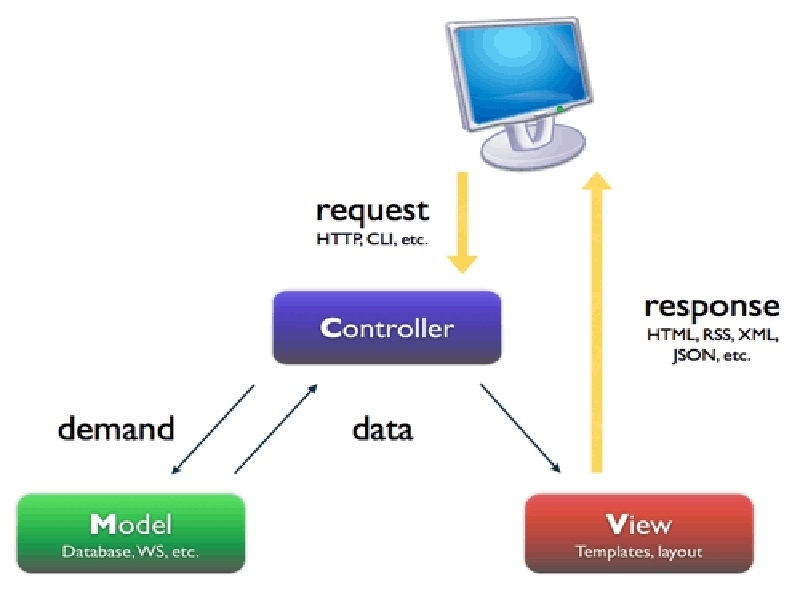


Hình 4-1. Sơ đồ khối của hệ thống

Hệ thống sẽ sử dụng kiến trúc client-server. Hệ thống sẽ sử dụng Internet và các giao thức web cơ bản để truyền tải dữ liệu giữa web service và ứng dụng Android. Sau đây là 4 khối của hệ thống:

* Database: Đây là khối lưu trữ tất cả dữ liệu của hệ thống. Trong đồ án này thì hệ quản trị MySQL sẽ được sử dụng để quản lý dữ liệu.
* Website backend: Phía backend sẽ sử dụng ngôn ngữ phía server (Java, PHP, C#...) để xây dựng một ứng dụng trên nền tảng web, giúp xử lý các request từ phía người dùng gửi tới, sau đó trả về cho người dùng các trang giao diện (phía front-end) là các thẻ html. Phần backend có thể lấy dữ liệu trực tiếp từ database hoặc cũng có thể thông qua khối webservice. Phần này sẽ được lập trình dựa bằng framework Spring (viết bằng ngôn ngữ Java) và sử dụng kiến trúc MVC.
* Webservice: Khối này cung cấp các dịch vụ cho ứng dụng Android sử dụng hoặc các client khác. Nó sẽ được viết bằng ngôn ngữ NodeJS để truyền tải các dữ liệu dưới dạng JSON tới ứng dụng Android (khối website backend cũng có thể dùng được dữ liệu này).
* Android application: Ứng dụng viết trên nền tảng Android, nó sẽ lấy dữ liệu từ hệ thống thông qua khối webservice.

## **4.2. Mô hình MVC cho phía backend**



Hình 4-2. Mô hình MVC

Các module phía backend của hệ thống được thiết kế theo mô hình 3 lớp MVC như sau:

* View Tier (Tầng giao diện): Là tầng tương tác với người dùng. Nhận các hành động của người dùng và gửi cho controller, để controller xử lý và trả về kết quả cho view, sau đó view hiện thị kết quả cho người dung. Tầng này thông thường là các màn hình như: login, quản lý, tìm kiếm…
* Controller Tier (Tầng xử lý nghiệp vụ): đóng vài trò trung gian giữa Model và View. Đây là tầng logic xử lý nghiệp vụ của chương trình. Nó có nhiệm vụ tiếp nhận yêu cầu từ client sau đó xử lý request, load model tương ứng và gửi data qua view tương ứng rồi trả kết quả về cho client. Tầng này đưa ra các tính toán, thống kê, load dữ liệu, xử lý các hành động của người dùng... Chú ý rằng nếu controller muốn truy cập dữ liệu thì phải lấy từ tầng model.
* Model Tier (Tầng dữ liệu): có nhiệm vụ thao tác với cơ sở dữ liệu, nghĩa là nó sẽ chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu (thao tác với database, files... như thêm mới, cập nhật, xóa…) và controller sẽ thông qua các hàm, phương thức đó để lấy dữ liệu rồi gửi qua View. Nó cũng có thể thông qua webservice để thao tác với dữ liệu được lưu trong database (như đã nói ở trên).

Tại sao lại sử dụng mô hình MVC? Mô hình MVC được sử dụng rộng rãi nhờ những tính năng ưu điểm sau:

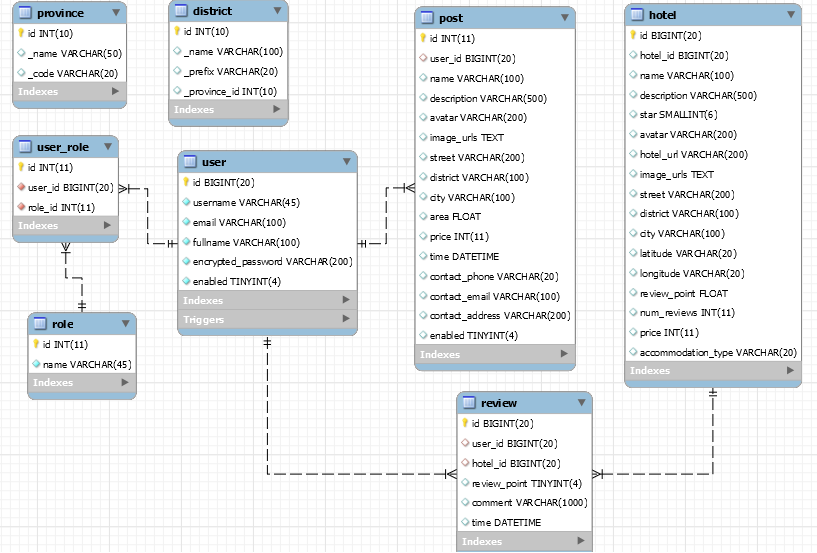
* Hệ thống phân ra từng phần nên phát triển dễ dàng, nhanh chóng
* Chia thành nhiều modun nhỏ và độc lập nên nhiều người có thể làm chung dự án
* Vấn đề bảo trì cũng tương đối đơn giản, dễ nâng cấp
* Dễ dàng debug trong quá trình xây dựng
* Thể hiện tính chuyên nghiệp trong thiết kế

Tuy vậy mô hình này vẫn tồn tại một vài nhược điểm sau:

* Kiến hệ thống cồng kềnh và phức tạp, khiến người mới sử dụng gặp nhiều khó khăn, đặc biệt là với các dự án nhỏ
* Dữ liệu được truyền qua lại giữa nhiều tầng trung gian, dẫn đến hiệu năng bị giảm
* Xây dựng cầu kỳ và tốn thời gian

Xét về ưu và nhược thì rõ ràng mô hình này có nhiều ưu điểm hơn và khuyến khích được sử dụng, đặc biệt với các hệ thống lớn

## **4.3. Thiết kế cơ sở dựa liệu**



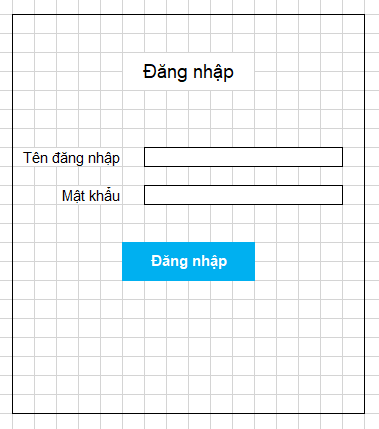
Hình 4-3. Các bảng trong cơ sở dữ liệu của hệ thống

Giải thích:

* Do user và role có quan hệ m-n nên ta tạo thêm một bảng user\_role biểu thị mối quan hệ đó. Bảng này có 2 thuộc tính lần lượt tham chiếu tới 2 bảng user và role.
* User có quan hệ 1-n với các bảng report, review và post, đã chỉ ra trong biều đồ lớp, mục 3.5.
* Mỗi 1 bài đăng (post) sẽ tương ứng với duy nhất một nhà trọ (hotel) nên quan hệ giữa 2 bảng này sẽ là 1-1.
* Mỗi một bản ghi trong bảng "post" tương ứng với 1 nhà trọ
* Các field "district", "city" trong bảng "hotel" có giá trị được fix cứng luôn chứ không tham chiếu tới 2 bảng "district" và "province", điều này sẽ dẫn đến việc dị thường dữ liệu khi sửa, xóa các bản ghi của bảng "district" và "province". Tuy nhiên do 2 bảng "district" và "province" chỉ để phục vụ việc lấy tên thành phố và quận huyện cho người dùng chọn nên có thể sẽ không cần liên quan đến bảng "hotel", và 2 bảng này cũng ít khi bị thay đổi, vì đây là các tỉnh thành của một quốc gia. Như vậy khi lấy dữ liệu từ bảng "hotel" ra, ta không cần phải join với bảng "district" và "province" nữa, điều này làm tăng tốc quá trình lấy dữ liệu (do chỉ cần lấy dữ liệu từ 1 bảng).

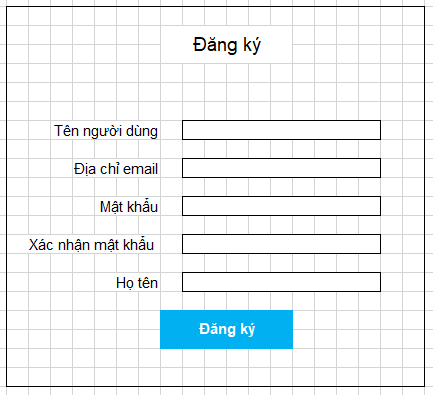
## **4.4. Thiết kế giao diện người dùng**

### *4.4.1. Giao diện đăng nhập*



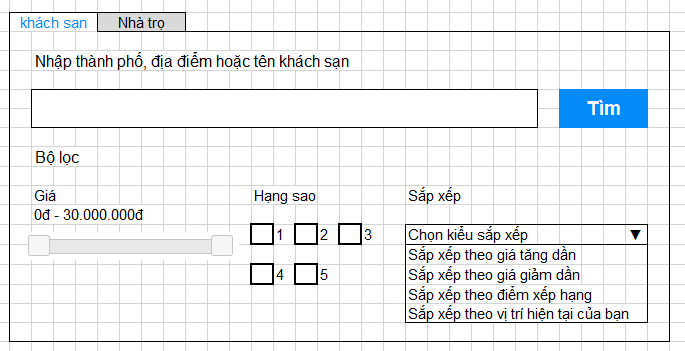
Hình 4-4. Giao diện đăng nhập

### *4.4.2. Giao diện đăng ký*

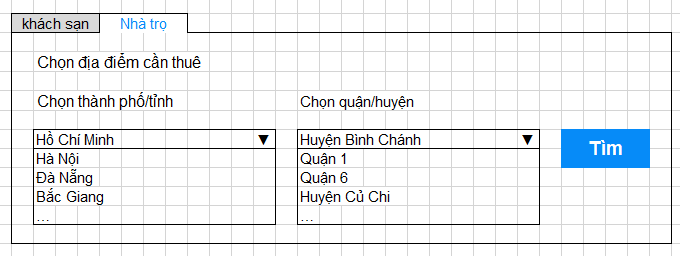


Hình 4-5. Giao diện đăng ký

### *4.4.3. Giao diện trang chủ*

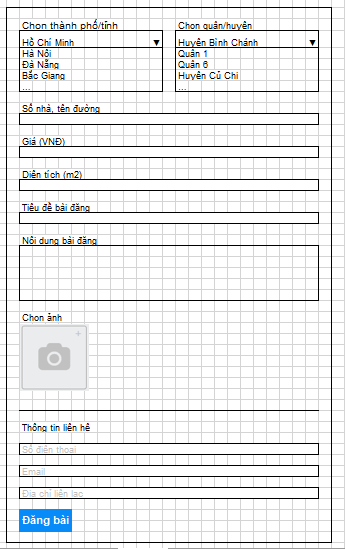


Hình 4-6. Giao diện tìm khách sạn



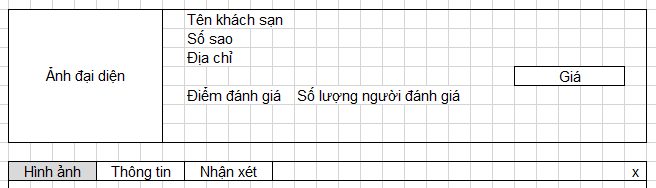
Hình 4-7. Giao diện tìm nhà trọ

### *4.4.4. Giao diện đăng bài*



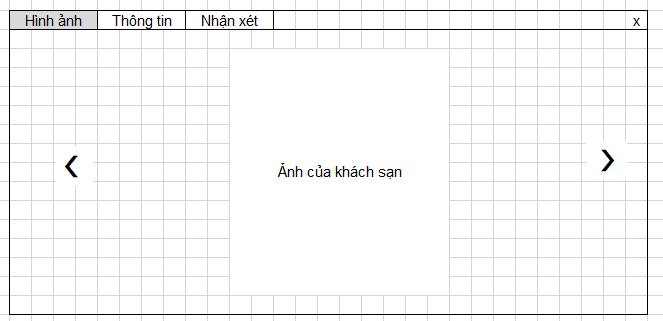
Hình 4-8. Giao diện đăng bài

### *4.4.5. Giao diện từng khách sạn*

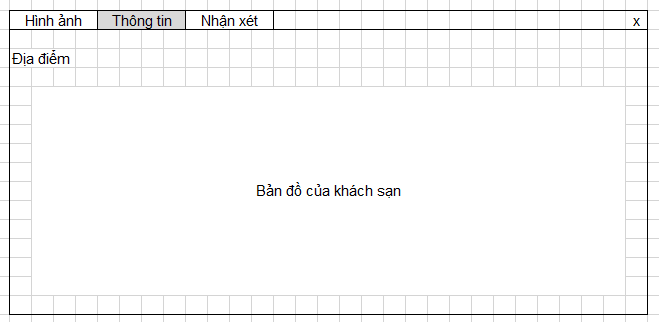


Hình 4-9. Giao diện tổng quát của từng khách sạn

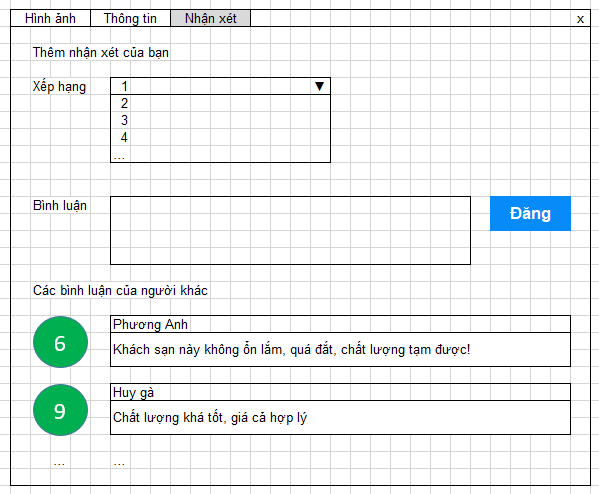
Khi người dùng click vào tên của khách sạn, thông tin chi tiết của khách sạn sẽ hiện ra với 3 tab dưới đây



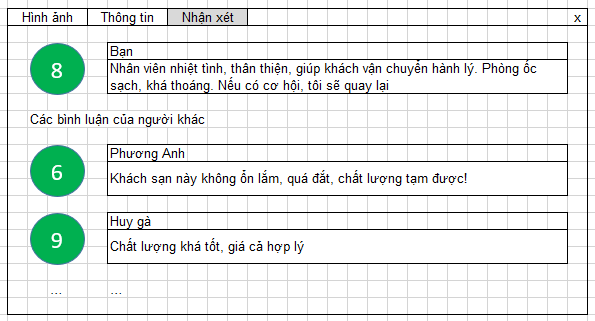
Hình 4-10. Chi tiết hình ảnh của khách sạn



Hình 4-11. Chi tiết thông tin, vị trí của khách sạn



Hình 4-12. Người dùng thêm nhận xét về khách sạn



Hình 4-13. Hiển thị các nhận xét của mọi người

Tóm tắt chương:

Chương này thiết kế hệ thống, bao gồm thiết kế cơ sở dữ liệu và giao diện người dùng. Bên cạnh đó cũng nêu lên mẫu thiết kế được sử dụng trong quá trình phân tích.

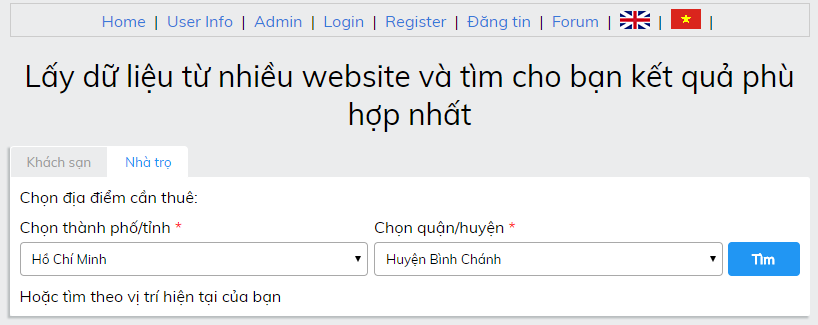
# **CHƯƠNG 5. KẾT QUẢ VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN**

## **5.1. Kết quả đạt được**

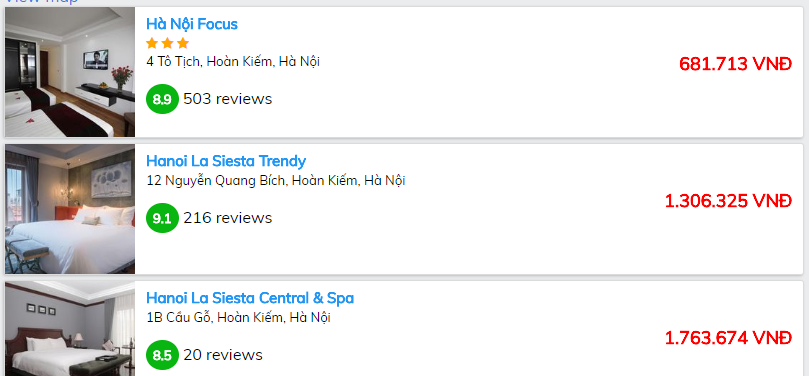
### *5.1.1. Ứng dụng trên nền tảng web*



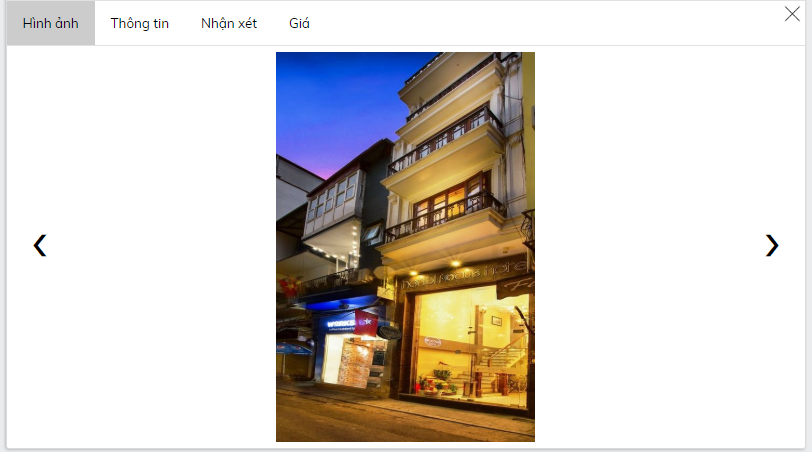
Hình 5-1. Trang chủ tìm khách sạn



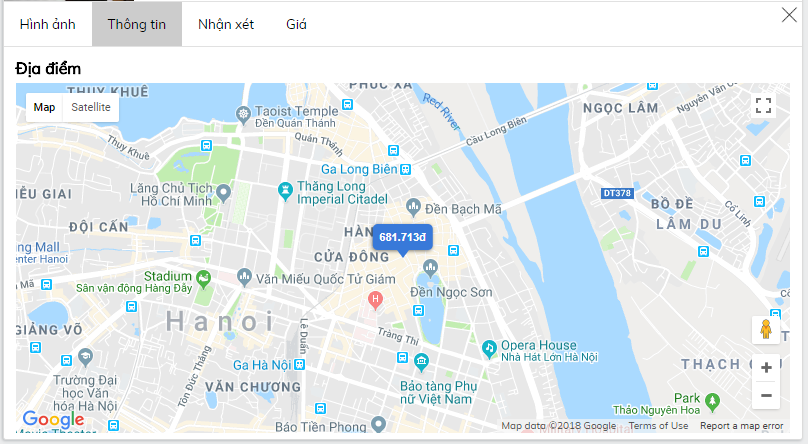
Hình 5-2. Trang chủ tìm nhà trọ



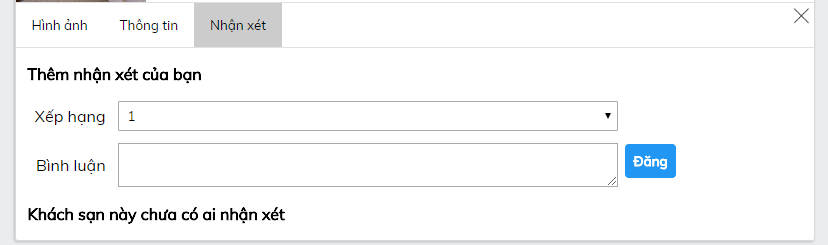
Hình 5-3. Kết quả tìm kiếm khách sạn



Hình 5-4. Hình ảnh của khách sạn



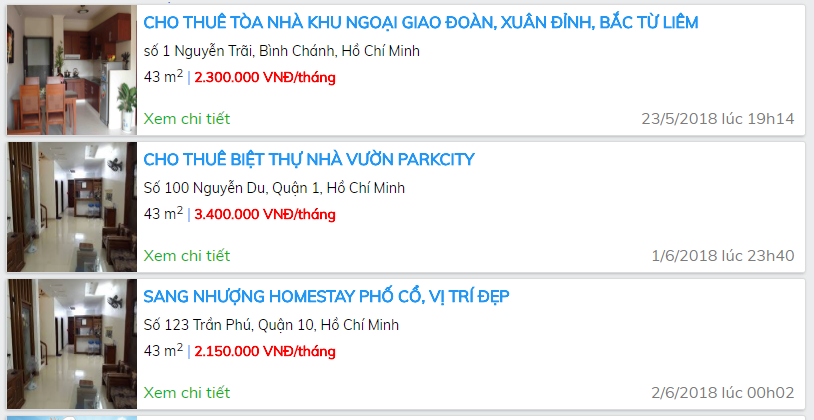
Hình 5-5. Vị trí của khách sạn trên bản đồ



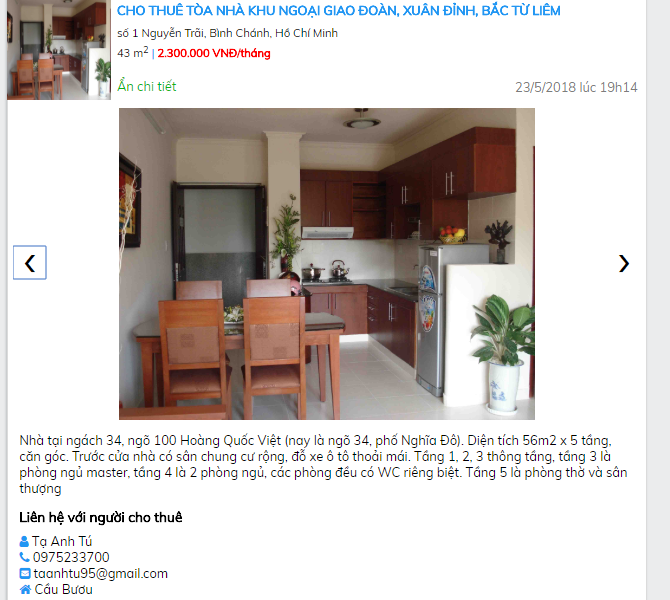
Hình 5-6. Thêm nhận xét cho khách sạn



Hình 5-7. Hiển thị các nhận xét gần nhất

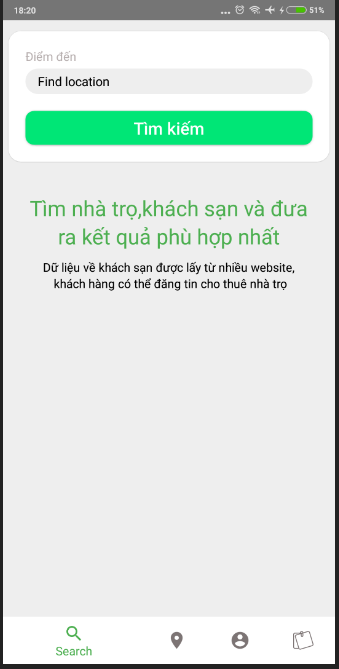


Hình 5-8. Kết quả tìm kiếm nhà trọ

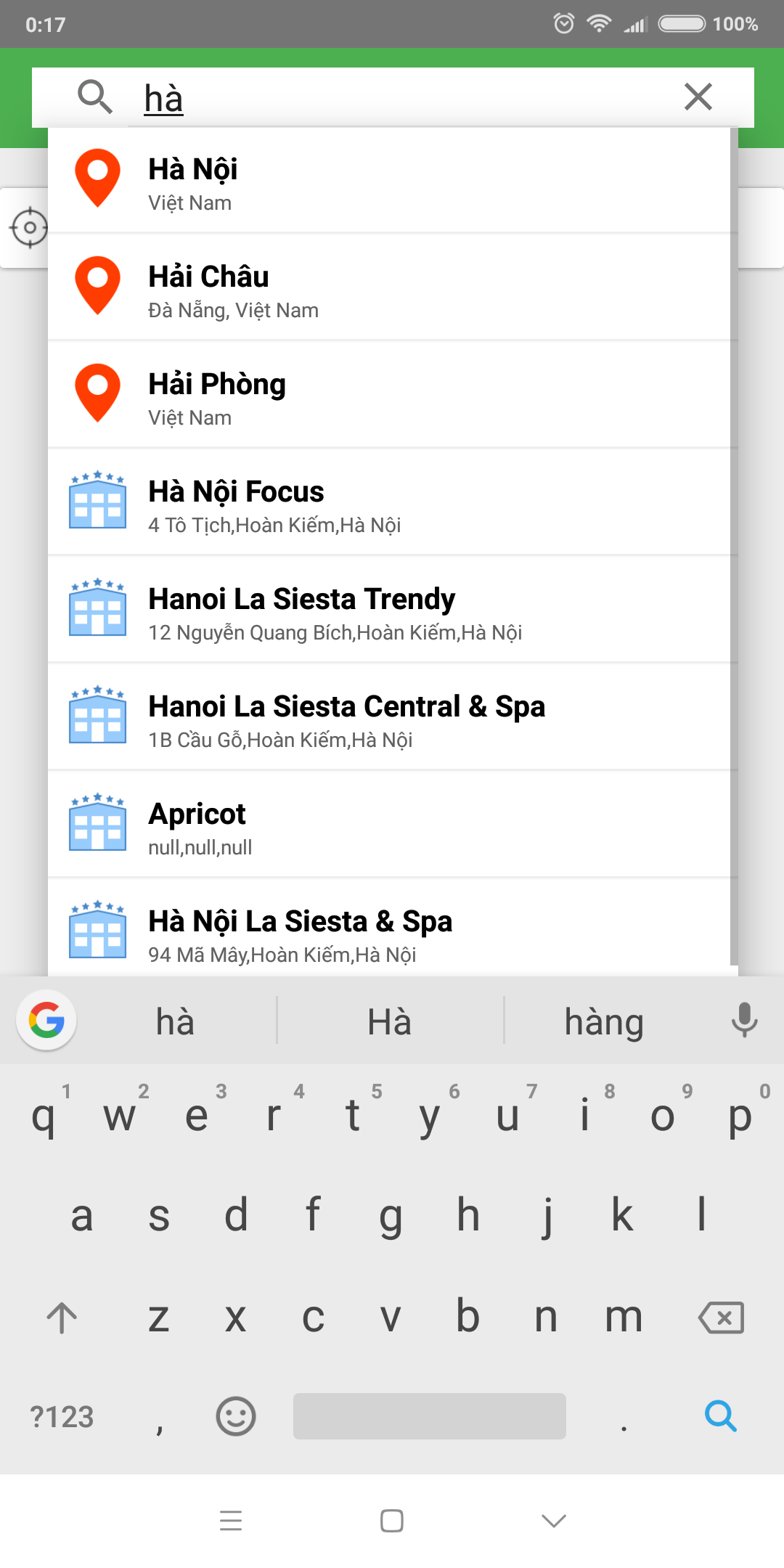


Hình 5-9. Khi click vào xem chi tiết

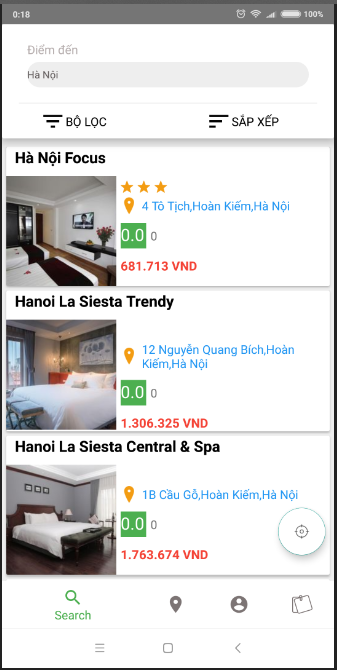
### *5.1.2. Ứng dụng trên nền tảng Android*



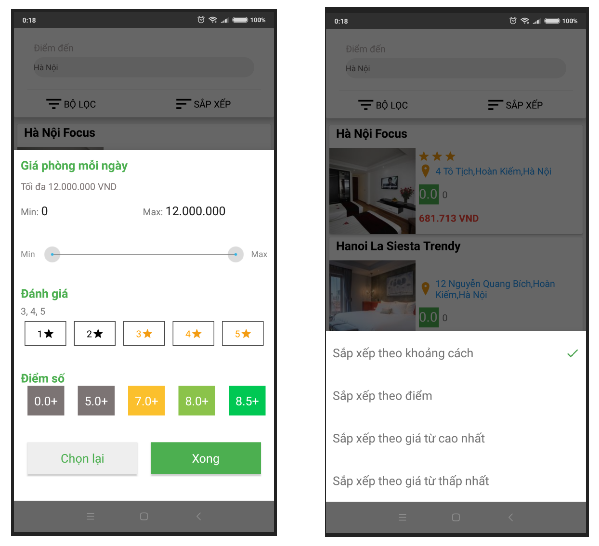
Hình 5-10. Màn hình tìm kiếm đầu tiên trên ứng dụng



Hình 5-11. Màn hình thực hiện tìm kiếm



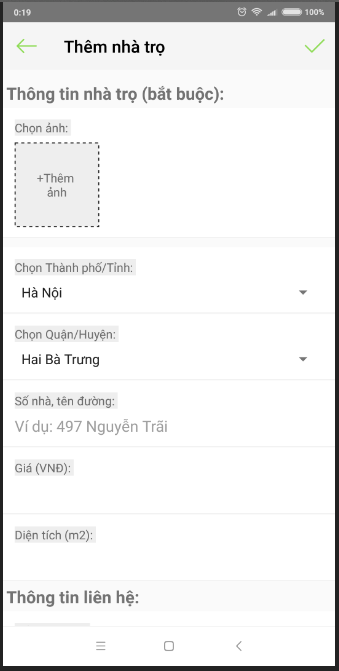
Hình 5-12. Màn hình kết quả tìm kiếm



Hình 5-13. Màn hình bộ lọc, sắp xếp kết quả tìm kiếm



Hình 5-14. Màn hình hiển thị vị trí trên bản đồ



Hình 5-15. Màn hình đăng tin nhà trọ

## **5.2. Kết luận**

Sau đây là những mục tiêu và kết quả mà em đã đạt được sau khi đồ án kết thúc

* Hệ thống được thiết kế và xây dựng đúng như theo quá trình phân tích lý thuyết. Các giao diện người dùng tương đối giống với các giao diện đã được vẽ và thiết kế trước đó.
* Hệ thống đã hoàn thiện được các chức năng cơ bản sau:
* Tìm kiếm khách sạn, nhà trọ
* Sắp xếp kết quả tìm kiếm theo giá, điểm xếp hạng
* Đăng tin thuê nhà
* Bình luận, đánh giá chất lượng của nhà trọ
* Đăng nhập, đăng ký, đăng xuất
* Hỗ trợ hai ngôn ngữ tiếng Anh và tiếng Việt (áp dụng với người dùng truy cập hệ thống bằng website)
* Cải thiện và nâng cao kỹ năng phân tích, thiết kế và triển khai một hệ thống.
* Tìm hiểu được thêm các công nghệ mới

Những khó khăn khi làm đồ án và những mục tiêu chưa đạt được

* Có khá nhiều công nghệ mới mà cả nhóm chưa biết, do đó cả nhóm đã tốn nhiều thời gian để học hỏi từ đầu
* Thời gian có hạn nên một số chức năng không kịp hoàn thiện
* Giao diện người dùng không sử dụng các template có sẵn nên không quá đẹp và responsive (không phù hợp với các loại kích thước màn hình khác nhau)
* Phần crawl dữ liệu từ các website khác còn làm thủ công nên chưa ứng dụng thực tế trên nhiều trang web khác nhau, thời gian crawl còn khá lâu
* Hệ thống chưa hoàn thành được những chức năng sau đây:
* Cho phép người dùng report những bài đăng xấu, không hợp lệ hoặc không đúng sự thật
* Đặt phòng, thanh toán trực tuyến
* Chỉ đường

## **5.3. Phương hướng phát triển**

Hệ thống có thể ứng dụng vào thực tế mà thương mại hóa nếu có thể phát triển được những chức năng sau:

* Cho phép người dùng đặt phòng, thanh toán trực tuyến
* Phần crawl dữ liệu có thể crawl được từ nhiều trang web, và đảm bảo việc không trùng lặp dữ liệu khi crawl từ nhiều nguồn khác nhau
* Giao diện người dùng phù hợp với mọi loại thiết bị
* …

Tóm tắt chương:

Chương này demo kết quả đạt được, những khó khăn gặp phải trong quá trình thực hiện đề tài, những phần mà chưa hoàn thiện được và phương hướng phát triển hệ thống trong tương lai.

# **TÀI LIỆU THAM KHẢO**

[1] Seyed M.M. “Saied” Tahaghoghi and Hugh E. Williams, *Learning MySQL*, O’Reilly, 2006, page 4-5.

[2] Introduction to Spring Boot: <https://projects.spring.io/spring-boot/>, truy cập lần cuối ngày 16/05/2018.

[3] Introduction to Spring Security: <https://projects.spring.io/spring-security/>, truy cập lần cuối ngày 16/05/2018.

[4] Introduction to GraphQL: <https://reactjs.org/blog/2015/02/20/introducing-relay-and-graphql.html>, truy cập lần cuối ngày 16/05/2018.

[5] GraphQL Queries and Mutations: <https://graphql.github.io/learn/queries/>, truy cập lần cuối ngày 07/06/2018.

[6] GraphQL Mutations: <https://graphql.github.io/learn/queries/#mutations> , truy cập lần cuối ngày 07/06/2018.

[7] Introductions to Google Map API: <https://developers.google.com/maps/documentation/android-sdk/intro>, truy cập lần cuối ngày 19/05/2018.

[8] Introductions to Google Places API:

<https://developers.google.com/places/android-sdk/intro>, truy cập lần cuối ngày 19/05/2018.

[9] Introductions to Google Places Web Service:

<https://developers.google.com/places/web-service/intro>, truy cập lần cuối ngày 19/05/2018.

[10] Introductions to Google Directions API:

<https://developers.google.com/maps/documentation/directions/intro>, truy cập lần cuối ngày 19/05/2018.

[11] Introductions to Google Geocoding API:

<https://developers.google.com/maps/documentation/geocoding/intro>, truy cập lần cuối ngày 19/05/2018.