

**BỘ GIÁO DỤC ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



**ĐỒ ÁN TỐT NGHIỆP**  
**XÂY DỰNG ỨNG DỤNG WEBSITE KẾT NỐI**  
**VIỆC THUÊ VÀ CHO THUÊ NHÀ Ở DỰA**  
**TRÊN CÔNG CỤ ASP NET CORE API VÀ**  
**VUEJS**

**SINH VIÊN THỰC HIỆN : NGUYỄN THỊ LINH**

**MÃ SINH VIÊN : 1451020141**

**KHOA : CÔNG NGHỆ THÔNG TIN**

**HÀ NỘI – 2024**

**BỘ GIÁO DỤC ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC ĐẠI NAM**



**NGUYỄN THỊ LINH**

**XÂY DỰNG ỨNG DỤNG WEBSITE KẾT NỐI  
VIỆC THUÊ VÀ CHO THUÊ NHÀ Ở DỰA  
TRÊN CÔNG CỤ ASP NET CORE API VÀ  
VUEJS**

**CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN**  
**MÃ SỐ : 74.80.201**

**NGƯỜI HƯỚNG DẪN : ThS. ĐỖ NGỌC ĐIỆP**

**HÀ NỘI – 2024**

## LỜI CAM ĐOAN

Với đề tài “**Xây dựng ứng dụng website kết nối việc thuê và cho thuê nhà ở dựa trên công cụ Asp Net Core Api và VueJS**” này do em thực hiện và là sản phẩm duy nhất, được xây dựng và hoàn thiện minh bạch và có sự hỗ trợ của giảng viên Đỗ Ngọc Điệp.

Tất cả các nội dung trong báo cáo đều được em nghiên cứu, tổng hợp và viết dựa trên kiến thức đã được học, tài liệu tham khảo và kinh nghiệm thực tế trong quá trình học tập.

Em xin cam đoan rằng báo cáo không vi phạm đạo đức học thuật, không sao chép hay đạo văn từ bất kỳ nguồn nào khác.

Trường hợp phát hiện vi phạm, em xin chịu hoàn toàn trách nhiệm.

Kính mong Hội đồng xem xét và đánh giá.

Hà Nội, ngày 11 tháng 06 năm 2024

Người thực hiện

Nguyễn Thị Linh

## LỜI CẢM ƠN

Trong lời đầu tiên của báo cáo “**Xây dựng ứng dụng website kết nối việc thuê và cho thuê nhà ở dựa trên công cụ Asp Net Core Api và VueJS**”, em muốn gửi những lời cảm ơn và biết ơn chân thành nhất của mình tới tất cả những người đã hỗ trợ, giúp đỡ em về kiến thức và tinh thần trong quá trình thực hiện bài làm.

Em xin chân thành gửi lời cảm ơn tới các thầy, cô giáo trong Trường Đại Học Đại Nam nói chung và các thầy cô giáo trong Khoa Công nghệ thông tin nói riêng đã tận tình giảng dạy, truyền đạt cho chúng em những kiến thức cũng như kinh nghiệm quý báu trong suốt quá trình học tập. Đặc biệt, em xin gửi lời cảm ơn đến giảng viên hướng dẫn – Thầy Đỗ Ngọc Điệp. Thầy đã tận tình theo sát giúp đỡ, trực tiếp chỉ bảo, hướng dẫn trong suốt quá trình nghiên cứu của em.

Do thời gian thực hiện có hạn kiến thức còn nhiều hạn chế nên bài làm của em chắc chắn không tránh khỏi những thiếu sót nhất định. Em rất mong nhận được ý kiến đóng góp của thầy, cô giáo và các bạn để em có thêm kinh nghiệm và tiếp tục hoàn thiện đề tài của mình.

Em xin chân thành cảm ơn.

[illegible]

## LỜI NÓI ĐẦU

Công nghệ thông tin ngày càng phát triển mạnh mẽ, mang lại sự hiện đại và đổi mới không ngừng. Trong xu hướng này, Internet đóng vai trò quan trọng nhất, ảnh hưởng đến mọi lĩnh vực của cuộc sống, đặc biệt là trong việc truyền tải và trao đổi dữ liệu. Sự kết nối Internet đã mở ra không gian rộng lớn của thông tin và tài nguyên, không chỉ giới hạn trong quốc gia mà còn trên toàn thế giới, và không chỉ qua ngôn ngữ mà còn thông qua hình ảnh và âm thanh.

Lợi ích từ Internet đã thúc đẩy sự phát triển mạnh mẽ của thương mại điện tử, làm thay đổi cách sản xuất, kinh doanh và các ngành công nghiệp khác hoạt động. Cuộc sống ngày càng phát triển, nhu cầu sống của con người ngày càng được nâng cao, bao gồm những yếu tố cơ bản như ăn, ở, mặc và du lịch.

Trong bối cảnh này, để đáp ứng nhu cầu ngày càng cao về dịch vụ ở, em đã chọn đề tài " Xây dựng ứng dụng website kết nối việc thuê và cho thuê nhà ở dựa trên công cụ Asp Net Core Api và VueJS". Việc này nhằm mục đích giải quyết các thách thức của cuộc sống hiện đại bằng cách tận dụng ưu điểm của Internet và công nghệ thông tin. Em muốn tạo ra một nền tảng trực tuyến giúp người dùng dễ dàng tìm kiếm nhà ở một cách thuận tiện từ bất kỳ đâu, giúp họ tiết kiệm thời gian và chi phí.

Website sẽ không chỉ mang lại lợi ích cho người sử dụng mà còn giúp quản trị viên dễ dàng quản lý thông tin, quản lý khách hàng và đơn đặt hàng. Em tin rằng việc xây dựng nền tảng này sẽ đóng góp tích cực vào việc cải thiện trải nghiệm của người tiêu dùng và nâng cao hiệu suất quản lý trong lĩnh vực tìm kiếm nhà ở.

## DANH MỤC BẢNG BIỂU

Bảng 1: Bảng các tác nhân .....	9
Bảng 2: Bảng danh sách use case .....	10
Bảng 3: Bảng phân quyền.....	28
Bảng 4: Bảng tài khoản .....	29
Bảng 5: Bảng nhân viên .....	29
Bảng 6: Bảng khách hàng.....	30
Bảng 7: Bảng nhà ở .....	31
Bảng 8: Bảng khu vực .....	32
Bảng 9: Bảng trạng thái .....	32
Bảng 10: Bảng loại nhà .....	33

## DANH MỤC HÌNH ẢNH

Hình 1: Biểu đồ use case tổng quát .....	10
Hình 2: Biểu đồ phân rã của use case Quản lý khu vực .....	11
Hình 3: Biểu đồ phân rã của use case Quản lý loại nhà .....	11
Hình 4: Biểu đồ phân rã của use case Quản lý loại nhà .....	12
Hình 5: Biểu đồ phân rã của use case Quản lý khách hàng .....	12
Hình 6: Biểu đồ phân rã của use case Quản lý nhân viên .....	13
Hình 7: Biểu đồ phân rã của use case Tìm kiếm nhà ở .....	13
Hình 8: Biểu đồ phân rã của use case Đăng ký .....	14
Hình 9: Kiến trúc phần mềm .....	14
Hình 10: Biểu đồ lớp thực thể .....	22
Hình 11: Biểu đồ phân tích ca sử dụng của use case Đăng nhập .....	23
Hình 12: Biểu đồ phân tích ca sử dụng của use case Nhân viên .....	23
Hình 13: Biểu đồ phân tích ca sử dụng của use case Khu vực .....	24
Hình 14: Biểu đồ phân tích ca sử dụng của use case Khách hàng .....	24
Hình 15: Biểu đồ phân tích ca sử dụng của use case Loại nhà .....	25
Hình 16: Biểu đồ phân tích ca sử dụng của use case Nhà ở .....	25
Hình 17: Biểu đồ tuần tự quản lý thêm nhân viên .....	26
Hình 18: Biểu đồ tuần tự quản lý thêm khu vực .....	26
Hình 19: Biểu đồ tuần tự quản lý thêm loại nhà .....	27
Hình 20: Biểu đồ tuần tự quản lý thêm nhà ở .....	27
Hình 21: Biểu đồ tuần tự quản lý xem danh sách khách hàng .....	28
Hình 22: Database Diagrams .....	33
Hình 23: Giao diện trang chủ .....	34
Hình 24: Giao diện trang yêu thích .....	34
Hình 25: Giao diện trang chi tiết .....	35



Hình 26: Giao diện trang tài khoản của tôi.....	35
Hình 27: Giao diện trang đăng nhập.....	36
Hình 28: Giao diện trang đăng bài .....	36
Hình 29: Giao diện danh sách bài đăng.....	37
Hình 30: Giao diện trang danh sách loại nhà .....	37
Hình 31: Giao diện tranh sách khu vực .....	37

## MỤC LỤC

Chương 1: GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI .....	1
1.1. Giới thiệu về đề tài. ....	1
1.2. Lý do chọn đề tài .....	1
1.3. Phân tích tính cấp thiết của đề tài.....	2
1.4. Đối tượng nghiên cứu .....	2
1.5. Mục đích và phạm vi nghiên cứu. ....	2
1.5.1. Mục đích nghiên cứu .....	2
1.5.2. Phạm vi nghiên cứu .....	3
1.6. Phương pháp nghiên cứu.....	4
1.7. Giới thiệu về Asp Net Core Api .....	4
1.8. Giới thiệu về VueJS.....	5
1.9. Thư viện sử dụng .....	6
1.9.1. Axios.....	6
1.9.2. Vuex .....	7
1.10. Bố cục của đồ án.....	7
Tiểu kết: .....	7
Chương 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG WEBSITE CHO THUÊ NHÀ .....	8
2.1. Yêu cầu bài toán .....	8
2.2. Phân tích hệ thống .....	9
2.2.1. Các tác nhân .....	9
2.2.2. Các yêu cầu chức năng.....	9
2.3. Thiết kế hệ thống .....	14
2.3.1. Kiến trúc phần mềm .....	14
2.3.2. Đặc tả yêu cầu hệ thống .....	16
2.3.3. Biểu đồ lớp thực thể .....	22

2.3.4. Biểu đồ phân tích ca sử dụng .....	22
2.3.5. Biểu đồ tương tác .....	25
2.3.6. Thiết kế cơ sở dữ liệu .....	28
Tiểu kết: .....	33
<b>Chương 3: KẾT QUẢ CHƯƠNG TRÌNH</b> .....	34
3.1. Giao diện khách hàng .....	34
3.1.1. Giao diện trang chủ .....	34
3.1.2. Giao diện trang yêu thích .....	34
3.1.3. Giao diện trang chi tiết nhà ở .....	35
3.1.4. Giao diện trang tài khoản của tôi.....	35
3.1.5. Giao diện trang đăng nhập.....	36
3.2. Giao diện admin .....	36
3.2.1. Giao diện trang đăng bài nhà ở.....	36
3.2.2. Giao diện danh sách bài đăng.....	36
3.2.3. Giao diện trang danh sách loại nhà .....	37
3.2.4. Giao diện trang danh sách khu vực .....	37
Tiểu kết: .....	37
<b>KẾT LUẬN</b> .....	38
1. Kết quả đạt được.....	38
2. Hạn chế của đề tài.....	38
3. Hướng phát triển.....	38
4. Bổ sung thêm một số chức năng.....	38
<b>TÀI LIỆU THAM KHẢO</b> .....	39

## **Chương 1: GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI**

### **1.1. Giới thiệu về đề tài.**

Trong bối cảnh hiện đại hóa và số hóa ngày càng phát triển, nhu cầu tìm kiếm và cho thuê nhà ở qua các nền tảng trực tuyến đang trở nên phổ biến và cần thiết hơn. Việc kết nối người cho thuê và người thuê thông qua một ứng dụng web tiện lợi, nhanh chóng và an toàn không chỉ giúp tiết kiệm thời gian và công sức mà còn mang lại nhiều giá trị cho cả hai bên. Tuy nhiên, vẫn còn nhiều hạn chế về tính năng, giao diện người dùng và độ bảo mật.

Đề tài "Xây dựng ứng dụng website kết nối việc thuê và cho thuê nhà ở dựa trên công cụ ASP.NET Core API và VueJS" được đề xuất nhằm tạo ra một nền tảng toàn diện, hiện đại và hiệu quả. Ứng dụng sẽ sử dụng các công nghệ tiên tiến như ASP.NET Core cho phần backend và VueJS cho phần frontend để đảm bảo hiệu suất, tính linh hoạt và khả năng mở rộng.

### **1.2. Lý do chọn đề tài**

Xu hướng thị trường và nhu cầu thực tiễn: Với sự phát triển mạnh mẽ của Internet và công nghệ số, việc tìm kiếm và cho thuê nhà ở qua các nền tảng trực tuyến ngày càng trở nên phổ biến. Người dùng có xu hướng tìm kiếm các giải pháp nhanh chóng, tiện lợi và an toàn, do đó, một ứng dụng web hiệu quả là nhu cầu thiết yếu.

Giải quyết các hạn chế hiện tại: Nhiều nền tảng hiện tại còn gặp phải các vấn đề như giao diện người dùng chưa thân thiện, tính năng chưa đầy đủ và bảo mật chưa cao. Đề tài này nhằm mục đích khắc phục những hạn chế đó bằng cách sử dụng các công nghệ tiên tiến, tạo ra một trải nghiệm tốt hơn cho người dùng.

Sự kết hợp giữa công nghệ hiện đại: ASP.NET Core API và VueJS là hai công nghệ hiện đại, mạnh mẽ và linh hoạt. ASP.NET Core API cung cấp khả năng xây dựng backend vững chắc, bảo mật và hiệu suất cao. VueJS mang đến trải nghiệm người dùng mượt mà và dễ dàng mở rộng. Sự kết hợp này sẽ tạo ra một ứng dụng toàn diện, đáp ứng tốt các yêu cầu của người dùng.

Tiềm năng mở rộng và phát triển: Một ứng dụng web được xây dựng trên nền tảng ASP.NET Core API và VueJS có khả năng mở rộng và phát triển dễ dàng trong tương lai.

Điều này cho phép tích hợp thêm nhiều tính năng mới, cải tiến giao diện người dùng và nâng cao bảo mật, đáp ứng nhu cầu ngày càng cao của thị trường.

**Đóng góp cho cộng đồng và xã hội:** Một nền tảng kết nối việc thuê và cho thuê nhà ở hiệu quả không chỉ mang lại lợi ích kinh tế mà còn giúp cải thiện chất lượng cuộc sống. Nó giúp người thuê dễ dàng tìm được nơi ở phù hợp, còn người cho thuê nhanh chóng tìm được khách hàng, từ đó thúc đẩy sự phát triển của thị trường bất động sản.

**Kinh nghiệm và kỹ năng phát triển:** Việc thực hiện đề tài này giúp người phát triển nâng cao kỹ năng lập trình, hiểu sâu hơn về các công nghệ hiện đại và quy trình phát triển một ứng dụng web hoàn chỉnh. Đây là cơ hội để học hỏi và tích lũy kinh nghiệm quý báu cho sự nghiệp sau này.

### **1.3. Phân tích tính cấp thiết của đề tài.**

Do nhu cầu thuê và cho thuê nhà ở ngày càng cao, đặc biệt tại thành phố lớn như Hà Nội. Việc tìm kiếm nhà ở phù hợp qua các loại hình như: báo, tờ rơi, môi giới,... gặp nhiều hạn chế và mất thời gian.

Nhờ sự phát triển của công nghệ mà mọi người có thể truy cập vào các trang mạng, đã tạo điều kiện cho việc phát triển các ứng dụng web kết nối với nhu cầu thuê và cho thuê nhà ở.

### **1.4. Đối tượng nghiên cứu**

**Người thuê nhà:** Những cá nhân hoặc gia đình đang tìm kiếm các tùy chọn nhà ở để thuê. Nghiên cứu nhu cầu, hành vi tìm kiếm, các tiêu chí lựa chọn nhà và các tính năng hỗ trợ người thuê như bộ lọc tìm kiếm, bản đồ, đánh giá từ người thuê trước.

**Người cho thuê nhà:** Những cá nhân hoặc tổ chức có bất động sản cần cho thuê. Tìm hiểu các yêu cầu của họ đối với một nền tảng quản lý tin đăng, cách thức họ quản lý tài sản và tương tác với người thuê.

### **1.5. Mục đích và phạm vi nghiên cứu.**

#### **1.5.1. Mục đích nghiên cứu**

Xây dựng ứng dụng website kết nối việc thuê và cho thuê nhà ở dựa trên công cụ Asp Net Core Api và VueJS để thực hiện mục tiêu tạo ra một ứng dụng có giao diện trực

quan, dễ sử dụng và hiệu quả, giúp người dùng dễ dàng tìm kiếm và đặt thuê nhà ở theo nhu cầu của mình.

**Xây dựng một nền tảng trực tuyến hiệu quả:** Tạo ra một ứng dụng web cho phép người dùng dễ dàng tìm kiếm và cho thuê nhà ở. Ứng dụng sẽ tập trung vào việc cung cấp trải nghiệm người dùng tốt nhất với giao diện thân thiện và dễ sử dụng.

**Ứng dụng công nghệ hiện đại:** Sử dụng các công nghệ tiên tiến như ASP.NET Core API cho phần backend và VueJS cho phần frontend nhằm đảm bảo hiệu suất, bảo mật và khả năng mở rộng của ứng dụng.

**Cải thiện bảo mật và an toàn thông tin:** Xây dựng hệ thống bảo mật mạnh mẽ để bảo vệ dữ liệu người dùng và ngăn chặn các nguy cơ xâm nhập từ bên ngoài.

**Tạo ra giá trị cho người dùng:** Đáp ứng nhu cầu của cả người thuê và người cho thuê nhà bằng cách cung cấp các tính năng hữu ích như tìm kiếm nâng cao, đánh giá và phản hồi, quản lý danh sách nhà cho thuê, và liên lạc trực tiếp giữa người thuê và chủ nhà.

**Thúc đẩy sự phát triển của thị trường bất động sản:** Góp phần vào sự phát triển của thị trường bất động sản bằng cách tạo ra một kênh giao dịch tiện lợi, minh bạch và hiệu quả.

### ***1.5.2. Phạm vi nghiên cứu***

- Công nghệ và nền tảng:
  - ASP.NET Core API: Nghiên cứu và áp dụng công nghệ ASP.NET Core API để xây dựng phần backend của ứng dụng, bao gồm quản lý dữ liệu, bảo mật, và các dịch vụ API.
  - VueJS: Nghiên cứu và áp dụng VueJS để xây dựng phần frontend, tạo ra giao diện người dùng tương tác, dễ sử dụng và thân thiện.
- Tính năng ứng dụng:
  - Đăng ký và đăng nhập: Xây dựng hệ thống đăng ký và đăng nhập an toàn cho người dùng.
  - Quản lý tài khoản: Cho phép người dùng quản lý thông tin cá nhân, danh sách nhà cho thuê và các yêu cầu thuê nhà.

- Tìm kiếm và lọc: Phát triển tính năng tìm kiếm và lọc nâng cao giúp người dùng dễ dàng tìm thấy nhà ở phù hợp.
- Liên lạc trực tiếp: Cung cấp tính năng liên lạc trực tiếp giữa người thuê và chủ nhà để trao đổi thông tin chi tiết.

### 1.6. Phương pháp nghiên cứu.

Phân tích yêu cầu và thiết kế hệ thống: Xác định các chức năng cần thiết, thiết kế cơ sở dữ liệu và giao diện người dùng.

Phát triển backend: Sử dụng Asp Net Core Api để quản lý người dùng, nhà cho thuê, tìm kiếm.

Phát triển frontend: Sử dụng VueJS để xây dựng giao diện người dùng.

Thử nghiệm và sửa lỗi ứng dụng.

### 1.7. Giới thiệu về Asp Net Core Api

- ASP.NET đã được sử dụng từ nhiều năm để phát triển các ứng dụng web. Kể từ đó, framework này đã trải qua một sự thay đổi tiến hóa đều đặn và cuối cùng đã đưa chúng ta đến với ASP.NET Core 1.0 hậu duệ mới gần đây nhất của nó.
  - ASP.NET Core không phải là phiên bản tiếp theo của ASP.NET 4.6
  - Đó là một framework hoàn toàn mới, may mắn rằng nó một dự án side-by-side tương tự với mọi thứ mà chúng ta biết.
  - Nó thực ra được viết lại trên framework ASP.NET 4.6 hiện tại những kích thước nhỏ hơn và nhiều modular hơn.
- Asp Net Core là một nền tảng mã nguồn mở, cross-platform framework cho việc xây dựng và kết nối các ứng dụng web hiện đại chẳng hạn như ứng dụng web, Internet of Thing, Mobile Backend - Web API ASP.NET Core cung cấp một kiến trúc để tối ưu hóa việc xây dựng các ứng dụng đám mây (microsoft azure) hoặc các ứng dụng độc lập.
- API (Application Programming Interface) là cách để hai hay nhiều chương trình máy tính có thể giao tiếp với nhau. Một số điểm nổi bật:
  - Tích hợp dễ dàng: API cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Nó giúp bạn nhúng nội dung từ bất kỳ trang web hoặc ứng dụng nào hiệu quả hơn.

- Kết nối ứng dụng: API là cầu nối cho hai hoặc nhiều ứng dụng trao đổi, tương tác với nhau. Nó cho phép các ứng dụng giao tiếp mà không cần sự can thiệp của người dùng.
  - Tính linh hoạt và tiện lợi: API có thể truy cập các thành phần ứng dụng và cung cấp dịch vụ linh hoạt hơn.
  - Quản lý và đo lường: API được quản lý và đo lường, giúp dễ dàng theo dõi và kiểm soát việc tương tác giữa các ứng dụng.
- API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.
  - Một số điểm quan trọng về Asp Net Core Api:
    - Ứng dụng client/server qua HTTP: Trong các ứng dụng web truyền thống, giao diện người dùng (HTML + JS + CSS) được server sinh ra và gửi cho trình duyệt qua HTTP. Tuy nhiên, hiện nay rất phổ biến các loại ứng dụng trong đó phần giao diện người dùng là một ứng dụng độc lập chạy trên một thiết bị khác và tương tác với ứng dụng server qua giao thức HTTP. Phần giao diện người dùng được gọi là client. ASP.NET Core Web API chịu trách nhiệm cung cấp dữ liệu, chủ yếu ở dạng JSON và XML cho client. Điều này dẫn đến Web API trở thành một phần vô cùng quan trọng trong các hệ thống ứng dụng hiện đại.
    - Đa dạng ứng dụng client: Web API có thể đồng thời được sử dụng bởi ứng dụng đơn trang (SPA), ứng dụng mobile, desktop và các ứng dụng khác. Điều này giúp tối ưu hóa việc cung cấp dữ liệu và tương tác qua HTTP.
    - Tạo dự án ASP.NET Core Web API: Có thể tạo dự án ASP.NET Core Web API bằng cách sử dụng Visual Studio hoặc Visual Studio Code. Sau khi tạo dự án, bạn có thể thêm các API Controller để xử lý các yêu cầu từ client.

## 1.8. Giới thiệu về VueJS

- Vue là một framework progressive dùng để xây dựng giao diện người dùng. Phiên bản đầu tiên được release vào tháng 12/2014.
- Khi phát triển lớp giao diện (view layer), người dùng chỉ cần dùng thư viện lõi (core library) của Vue, vốn rất dễ học và tích hợp với các thư viện hoặc dự án có sẵn.
- Vue cũng đáp ứng được dễ dàng nhu cầu xây dựng những ứng dụng một trang (SPA - Single-Page Applications) với độ phức tạp cao.



- Ưu điểm:

- Kích thước nhỏ: Tập zip của Vue.js chỉ nặng khoảng 18 KB, giúp cài đặt nhanh và tối ưu hóa SEO và trải nghiệm người dùng.
- DOM ảo: Vue.js sử dụng DOM ảo để cải thiện hiệu suất ứng dụng. Thay vì cập nhật toàn bộ DOM, Vue.js chỉ cập nhật những phần tử cần thiết, giúp hiển thị trang nhanh chóng.
- Ràng buộc dữ liệu: Vue.js hỗ trợ ràng buộc dữ liệu giữa mô hình dữ liệu và giao diện người dùng. Bạn có thể dễ dàng liên kết dữ liệu và làm cho ứng dụng trở nên “reactive”.

- Nhược điểm:

- Rào cản ngôn ngữ: Việc áp dụng VueJS của các doanh nghiệp như Xiaomi và Alibaba đã giúp phổ biến framework và tạo ra nhu cầu trên thị trường lao động. Với việc Vue.js ngày càng phổ biến ở Trung Quốc, một phần quan trọng trong nội dung và các cuộc thảo luận của nó đều là tiếng Trung.
- Thiếu hỗ trợ cho các dự án quy mô lớn: Quy mô nhóm phát triển và cộng đồng của VueJS vẫn không thể so sánh với Angular hay React. Framework này cũng không được hỗ trợ tài chính từ các doanh nghiệp lớn. Để được áp dụng trong các dự án quy mô lớn, công nghệ phải ổn định và được hỗ trợ mạnh mẽ để các vấn đề có thể được giải quyết nhanh chóng. Mặc dù VueJS không gặp nhiều vấn đề và thậm chí còn có nhu cầu đến từ các doanh nghiệp như IBM và Adobe, nhưng nó chủ yếu được sử dụng trong các dự án tương đối nhỏ.
- Nguồn tài nguyên giới hạn: Mặc dù hệ sinh thái khá rộng và có tất cả các công cụ cần thiết để bắt đầu phát triển với VueJS, nhưng framework này không lớn bằng React hay Angular.

## 1.9. Thư viện sử dụng

### 1.9.1. Axios

Axios là một thư viện HTTP Client dựa trên Promise dành cho node.js và trình duyệt. Nó có tính đẳng hình (tức là cùng codebase có thể chạy trong cả trình duyệt và node.js). Ở phía server thì nó sử dụng native module http trong node.js, còn ở phía client (trình duyệt) thì nó sử dụng XMLHttpRequest.

### 1.9.2. Vuex

Vuex là một mẫu quản lý trạng thái (state) + thư viện cho các ứng dụng Vue.js, lấy cảm hứng từ Flux, Redux và kiến trúc Elm nhưng được thiết kế và điều chỉnh đặc biệt để tích hợp tốt với Vue.js và tận dụng Reactivity của Vue.

Tất cả các thư viện này đều nhằm giải quyết một vấn đề đơn giản: Khi có trạng thái được chia sẻ trên nhiều thành phần, đặc biệt là các thành phần là anh chị em hoặc theo view rất khác nhau, việc quản lý phân phối và cập nhật trạng thái đó là một thách thức.

Các thư viện như Vuex cho phép quản lý trạng thái chia sẻ giữa các thành phần theo cách có cấu trúc và có thể duy trì, bằng cách tạo một cây trạng thái toàn cầu có thể được truy cập và cập nhật bởi mọi thành phần theo cách có cấu trúc.

Nó phục vụ như một cửa hàng tập trung cho tất cả các thành phần trong một ứng dụng, với các quy tắc đảm bảo rằng trạng thái chỉ có thể bị đột biến theo kiểu có thể dự đoán được.

Nó cũng tích hợp với tiện ích mở rộng devtools chính thức của Vue để cung cấp các tính năng nâng cao như gỡ lỗi thời gian không cấu hình và xuất/nhập ảnh chụp nhanh trạng thái

## 1.10. Bố cục của đề án

Chương 1: Giới thiệu tổng quan về đề tài

Chương 2: Phân tích thiết kế hệ thống website cho thuê nhà

Chương 3: Kết quả chương trình

**Tiểu kết:** Trong bối cảnh nhu cầu thuê và cho thuê nhà ở ngày càng cao, đặc biệt tại các thành phố lớn như Hà Nội, việc xây dựng một ứng dụng website kết nối người cho thuê và người thuê trở nên cấp thiết hơn bao giờ hết. Đề tài "Xây dựng ứng dụng website kết nối việc thuê và cho thuê nhà ở dựa trên công cụ ASP.NET Core API và VueJS" được đề xuất nhằm đáp ứng nhu cầu này một cách toàn diện và hiệu quả.

Ứng dụng sẽ sử dụng công nghệ ASP.NET Core API cho phần backend và VueJS cho phần frontend, đảm bảo hiệu suất, tính linh hoạt và khả năng mở rộng. Nghiên cứu này tập trung vào hai đối tượng chính: người thuê nhà và người cho thuê nhà, với mục tiêu cung cấp một nền tảng tiện lợi, an toàn và dễ sử dụng cho cả hai bên.

## **Chương 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG WEBSITE CHO THUÊ NHÀ**

### **2.1. Yêu cầu bài toán**

Xây dựng Website cho phép tìm kiếm nhà ở online. Website cho phép khách hàng có thể xem chi tiết nhà ở, tìm kiếm, liên hệ bên cho thuê; nhân viên có thể thêm bài đăng nhà ở, khu vực và trạng thái nhà ở; admin quản lý nhân viên. Các yêu cầu của hệ thống được mô tả như sau:

- Admin sẽ quản lý nhân viên gồm các thông tin (Mã nhân viên, tên nhân viên, số điện thoại, địa chỉ, email, quyền). Khi thêm một nhân viên mới thì admin có nhiệm vụ cập nhật thông tin nhân viên vào trong hệ thống, nếu thông tin có sai sót thì có thể chỉnh sửa hoặc xóa thông tin về nhân viên đó.
- Để biết khu vực nhà ở đâu thì nhân viên sẽ có nhiệm vụ thêm thông tin khu vực (Mã khu vực, tên khu vực). Khi thêm một khu vực thì nhân viên có nhiệm vụ cập nhật thông tin khu vực vào trong hệ thống, nếu thông tin có sai sót thì nhân viên có thể chỉnh sửa hoặc xóa thông tin về khu vực đó.
- Nhiệm vụ của nhân viên còn có quản lý trạng thái nhà ở gồm thông tin (Mã trạng thái, tên trạng thái). Khi thêm một trạng thái thì nhân viên có nhiệm vụ cập nhật thông tin trạng thái vào trong hệ thống, nếu thông tin có sai sót thì nhân viên có thể chỉnh sửa hoặc xóa thông tin về trạng thái đó.
- Nhân viên có thể xem thông tin khách hàng và không thể sửa hay xóa thông tin khách hàng.
- Để biết nhà ở thuộc loại nhà nào thì nhân viên phải thêm thông tin loại nhà (Mã loại nhà, tên loại nhà). Khi thêm một loại nhà thì nhân viên có nhiệm vụ cập nhật thông tin loại nhà vào trong hệ thống, nếu thông tin có sai sót thì nhân viên có thể chỉnh sửa hoặc xóa thông tin về loại nhà đó.
- Nhân viên đăng bài gồm thông tin (Mã nhà ở, địa chỉ, giá, diện tích, hình ảnh, mô tả, khu vực, trạng thái, loại nhà ở). Khi thêm một bài đăng thì nhân viên có nhiệm vụ cập nhật thông tin bài đăng vào trong hệ thống, nếu thông tin có sai sót thì nhân viên có thể chỉnh sửa hoặc xóa thông tin về bài đăng đó.

## 2.2. Phân tích hệ thống

### 2.2.1. Các tác nhân

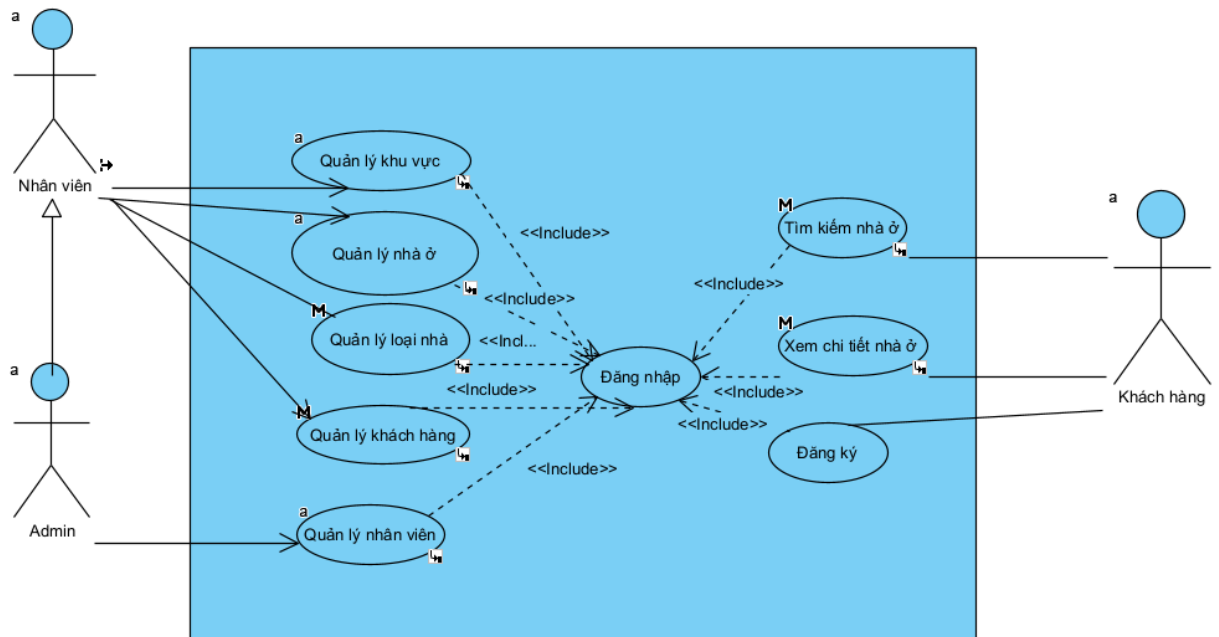
*Bảng 1: Bảng các tác nhân*

STT	Tên tác nhân	Mô tả
1	Admin	Thao tác trên chức năng quản lý nhân viên
2	Nhân viên	Thao tác trên tất cả các chức năng của trang web: quản lý khu vực, loại nhà, nhà ở, đăng bài, khách hàng
3	Khách hàng	Thao tác trên chức năng tìm kiếm nhà ở.

### 2.2.2. Các yêu cầu chức năng

#### a. Biểu đồ use case tổng quát

Hệ thống gồm hai trang chính. Một trang cho khách hàng, một trang cho quản trị và nhân viên. Tác nhân tham gia vào trang khách hàng là khách hàng khi thực hiện vào website xem thông tin nhà ở. Tác nhân tham gia vào trang quản trị thông tin của website và quản lý thông tin nhà ở.



Hình 1: Biểu đồ use case tổng quát

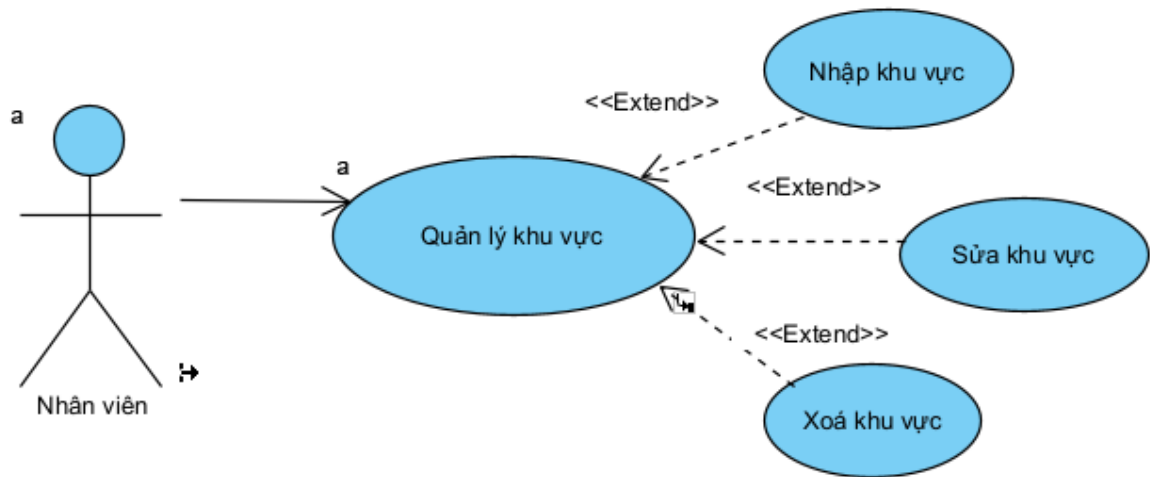
b. Danh sách các use case

Bảng 2: Bảng danh sách use case

STT	Tên use case	Ý nghĩa/ Ghi chú
1	Quản lý khu vực	Thêm, sửa, xóa khu vực nhà ở
2	Quản lý loại nhà ở	Thêm, sửa, xóa loại nhà ở
3	Quản lý nhà ở	Thêm, sửa, xóa nhà ở
4	Quản lý khách hàng	Xem danh sách khách hàng
5	Quản lý nhân viên	Thêm, sửa, xóa nhân viên
6	Tìm kiếm nhà ở	Tìm kiếm nhà ở bằng khu vực muốn tìm
7	Xem chi tiết nhà ở	Xem chi tiết thông tin nhà ở (khu vực, giá, diện tích,...)
8	Đăng ký	Cho phép người dùng đăng ký tài khoản

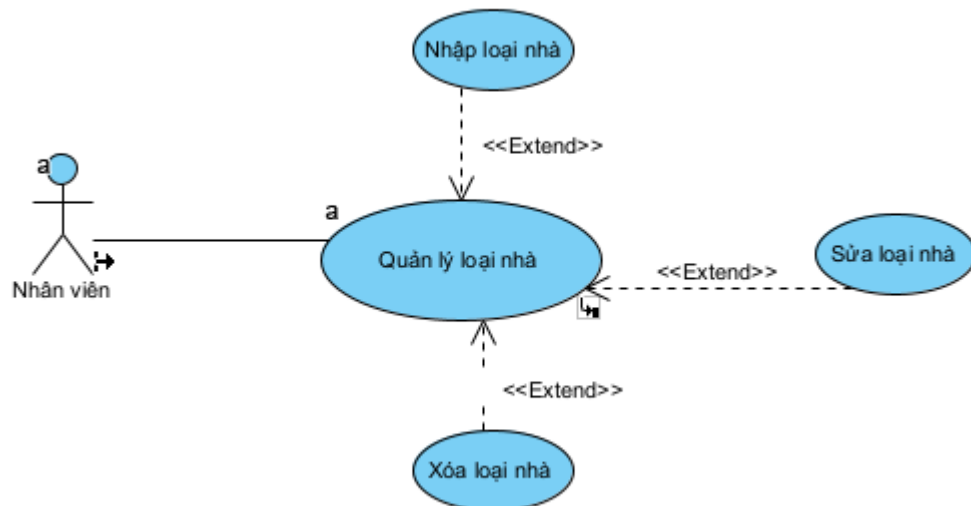
Sau khi xây dựng biểu đồ ca sử dụng mức tổng quát, thì tiếp tục xây dựng các biểu đồ ca sử dụng phân rã.

- Quản lý khu vực



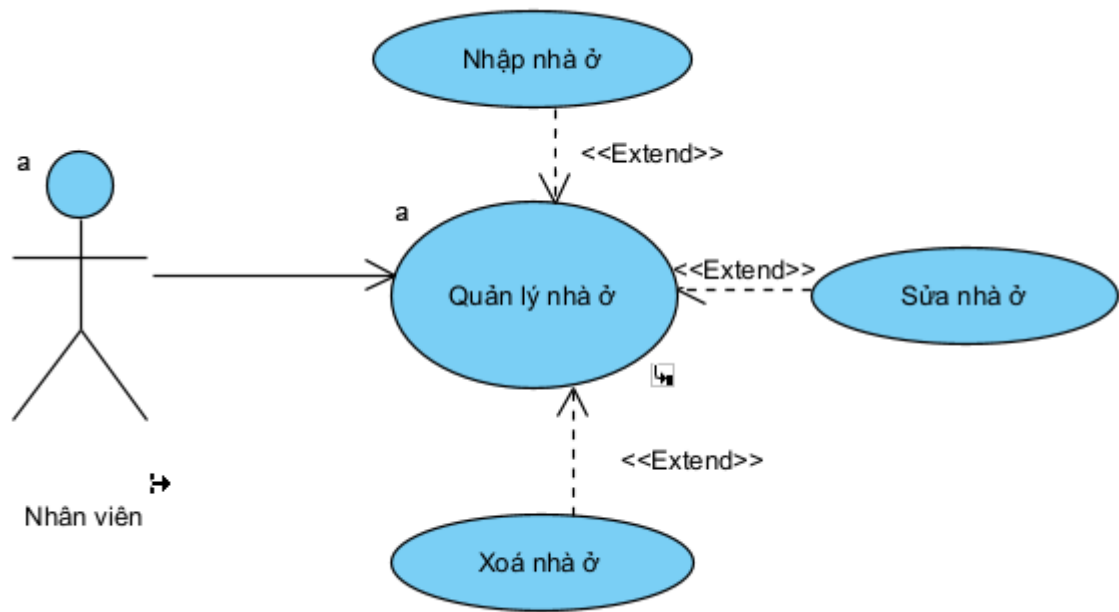
Hình 2: Biểu đồ phân rã của use case Quản lý khu vực

- Quản lý loại nhà ở



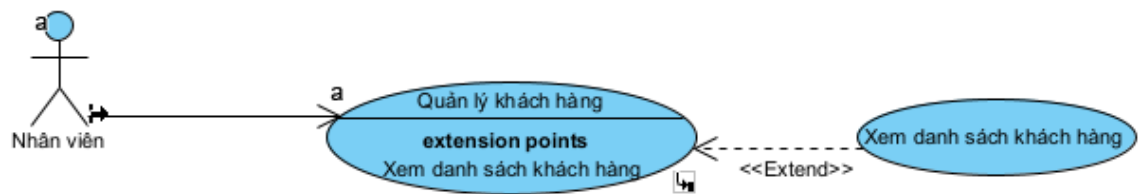
Hình 3: Biểu đồ phân rã của use case Quản lý loại nhà ở

- Quản lý nhà ở



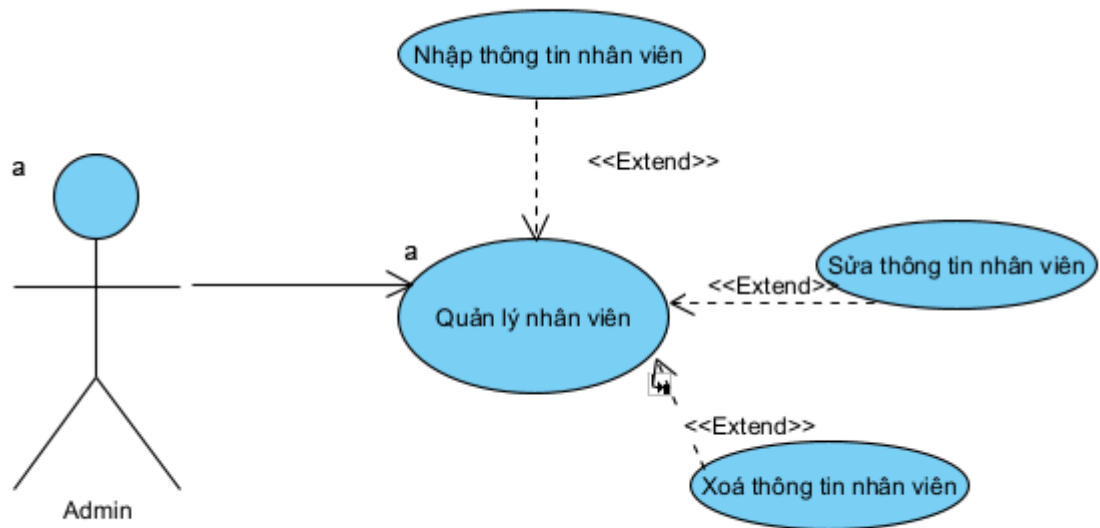
Hình 4: Biểu đồ phân rã của use case Quản lý loại nhà

- Quản lý khách hàng



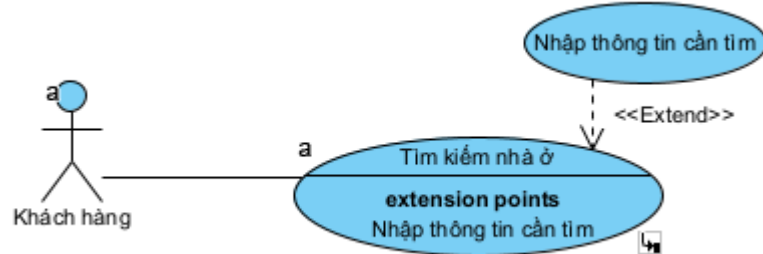
Hình 5: Biểu đồ phân rã của use case Quản lý khách hàng

- Quản lý nhân viên



Hình 6: Biểu đồ phân rã của use case Quản lý nhân viên

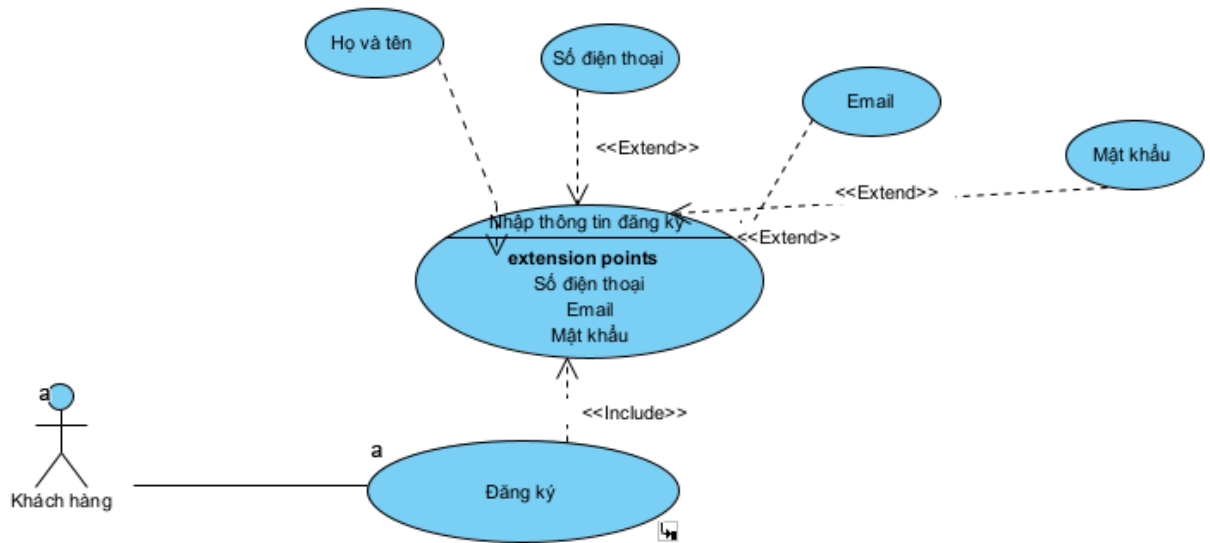
- Tìm kiếm nhà ở



Hình 7: Biểu đồ phân rã của use case Tìm kiếm nhà ở



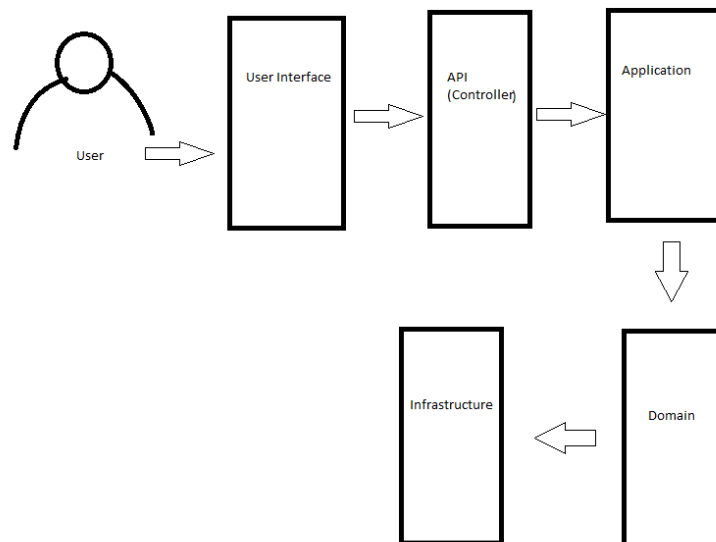
- Đăng ký



Hình 8: Biểu đồ phân rã của use case Đăng ký

## 2.3. Thiết kế hệ thống

### 2.3.1. Kiến trúc phần mềm



Hình 9: Kiến trúc phần mềm

Có một số cách tiếp cận khác nhau có thể được thực hiện để thiết kế kiến trúc của một hệ thống. Dưới đây là một cái nhìn cấp cao có thể có về kiến trúc của hệ thống:

- User Interface:

- Mô tả: Đây là phần nơi người sử dụng tương tác với hệ thống. Giao diện người dùng có thể bao gồm các trang web, ứng dụng di động hoặc bất kỳ công cụ nào người dùng sử dụng để tương tác với hệ thống.
- Chức năng: Hiển thị thông tin nhà ở, cho phép tìm kiếm, quản lý tài khoản người dùng và thực hiện các chức năng tương tự.
- API (Controllers):
  - Mô tả: Đây là lớp chứa controllers, nơi xử lý các yêu cầu từ giao diện người sử dụng và gọi các dịch vụ cần thiết từ tầng Application.
  - Chức năng: Xử lý bài đăng, quản lý khu vực, loại nhà ở, nhà ở, nhân viên, danh sách khách hàng.
- Application:
  - Mô tả: Đây là tầng chứa các Services và Modules. Các Services thực hiện các nhiệm vụ cụ thể, trong khi Modules có thể chia nhỏ chức năng thành các phần nhỏ hơn để dễ quản lý.
  - Chức năng: Xử lý logic quản lý nhà ở, trạng thái nhà ở, bài đăng và giao tiếp với tầng Domain.
- Domain:
  - Mô tả: Chứa các Entities và Repositories, biểu diễn các khái niệm cốt lõi của miền. Entities là các đối tượng có ý nghĩa trong ngữ cảnh, trong khi Repositories làm nhiệm vụ truy xuất và lưu trữ dữ liệu liên quan đến Entities.
  - Chức năng: Biểu diễn thông tin về nhà ở, người dùng trong miền
- Infrastructure:
  - Mô tả: Chứa Context và Repositories, nơi thực hiện các nhiệm vụ liên quan đến dữ liệu và lưu trữ. Context thường đại diện cho cơ sở dữ liệu hoặc cơ sở dữ liệu và cung cấp một cách để tương tác với nó.
  - Chức năng: Quản lý kết nối và truy vấn đến cơ sở dữ liệu, cung cấp cơ sở hạ tầng cho tầng Domain và Application.

Tóm lại mô hình DDD triển khai các nguyên tắc, nơi logic kinh doanh được tập trung ở tầng Domain và Application, còn tầng Infrastructure chịu trách nhiệm về quản lý dữ liệu và kết nối đến cơ sở dữ liệu. Cấu trúc thư mục được tổ chức logic và dễ quản lý.

### **2.3.2. Đặc tả yêu cầu hệ thống**

Sau khi phân rã các ca sử dụng trong hệ thống, thì xây dựng tài liệu đặc tả chi tiết cho ca sử dụng. Các thông tin mô tả ca sử dụng gồm một số các thông tin chung như tên chức năng, đường dẫn, mô tả chức năng.

#### *a. Chức năng đăng nhập*

- Tên chức năng: Đăng nhập
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập
- Mô tả chức năng: Người dùng đăng nhập vào hệ thống
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Người dùng nhập đầy đủ thông tin rồi ấn nút đăng nhập. Nếu người dùng bỏ trống 1 trong 2 ô thì hệ thống sẽ không thực hiện chức năng đăng nhập.

#### *b. Chức năng xem chi tiết nhà ở*

- Tên chức năng: Xem chi tiết nhà ở
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập >> Điều hướng >> Chọn một nhà ở xem chi tiết
- Mô tả chức năng: Người dùng xem chi tiết một nhà ở

Người dùng chọn một nhà ở muốn xem chi tiết thông tin.

#### *c. Chức năng thêm nhà ở vào yêu thích*

- Tên chức năng: Thêm nhà ở vào yêu thích
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập >> Chọn một nhà ở vào yêu thích
- Mô tả chức năng: Người dùng thêm nhà ở vào yêu thích

Người dùng thêm nhà ở vào mục yêu thích.

*d. Chức năng thêm bài đăng*

- Tên chức năng: Thêm bài đăng
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Bài đăng >> Thêm bài đăng
- Mô tả chức năng: Nhân viên đăng nhập thêm bài đăng
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Nhân viên thêm bài đăng nhà ở mới.

*e. Chức năng sửa bài đăng*

- Tên chức năng: Sửa bài đăng
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Bài đăng >> Sửa bài đăng
- Mô tả chức năng: Nhân viên đăng nhập sửa bài đăng
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Nhân viên chọn bài đăng nhà ở cần sửa rồi ấn nút sửa. Sửa thông tin cần sửa và lưu.

*f. Chức năng xóa bài đăng*

- Tên chức năng: Xóa bài đăng
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Bài đăng >> Xóa bài đăng
- Mô tả chức năng: Nhân viên đăng nhập xóa bài đăng
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Nhân viên chọn bài đăng cần xóa rồi ấn nút xóa.

*g. Chức năng xem danh sách khách hàng*

- Tên chức năng: Xem danh sách khách hàng
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Danh sách khách hàng
- Mô tả chức năng: Nhân viên đăng nhập xem danh sách khách hàng
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Nhân viên chỉ được xem danh sách khách hàng và không được sửa thông tin khách hàng.

*h. Chức năng thêm nhân viên*

- Tên chức năng: Thêm nhân viên
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Nhân viên >> Điều hướng >> Thêm nhân viên
- Mô tả chức năng: Admin đăng nhập thêm nhân viên
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Admin thêm nhân viên mới.

*i. Chức năng sửa nhân viên*

- Tên chức năng: Sửa nhân viên
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Nhân viên >> Điều hướng >> Sửa nhân viên
- Mô tả chức năng: Admin đăng nhập sửa nhân viên
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Admin sửa thông tin nhân viên.

### *k. Chức năng xóa nhân viên*

- Tên chức năng: Xóa nhân viên
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Nhân viên >> Điều hướng >> Xóa nhân viên
- Mô tả chức năng: Admin đăng nhập xóa nhân viên
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Admin xóa nhân viên cần xóa.

### *l. Chức năng thêm khu vực*

- Tên chức năng: Thêm khu vực
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Khu vực >> Điều hướng >> Thêm khu vực
- Mô tả chức năng: Nhân viên đăng nhập thêm khu vực nhà ở
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Nhân viên có thể thêm thông tin khu vực nhà ở các quận của Hà Nội.

### *m. Chức năng sửa khu vực*

- Tên chức năng: Sửa khu vực
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Khu vực >> Điều hướng >> Sửa khu vực
- Mô tả chức năng: Nhân viên đăng nhập sửa khu vực nhà ở
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Nhân viên sửa thông tin khu vực nhà ở các quận của Hà Nội.

*n. Chức năng xóa khu vực*

- Tên chức năng: Xóa khu vực
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Khu vực >> Điều hướng >> Xóa khu vực
- Mô tả chức năng: Nhân viên đăng nhập xóa khu vực nhà ở
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Nhân viên xóa thông tin khu vực nhà ở các quận của Hà Nội.

*o. Chức năng thêm trạng thái*

- Tên chức năng: Thêm trạng thái
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Trạng thái >> Điều hướng >> Thêm trạng thái
- Mô tả chức năng: Nhân viên đăng nhập thêm trạng thái nhà ở
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Nhân viên thêm thông tin trạng thái nhà ở.

*p. Chức năng sửa trạng thái*

- Tên chức năng: Sửa trạng thái
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Trạng thái >> Điều hướng >> Sửa trạng thái
- Mô tả chức năng: Nhân viên đăng nhập sửa trạng thái nhà ở
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Nhân viên có thể sửa thông tin trạng thái nhà ở.

*q. Chức năng xóa trạng thái*

- Tên chức năng: Xóa trạng thái
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Trạng thái >> Điều hướng >> Xóa trạng thái
- Mô tả chức năng: Nhân viên đăng nhập xóa trạng thái nhà ở
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Nhân viên có thể xóa thông tin trạng thái nhà ở.

*r. Chức năng thêm loại nhà ở*

- Tên chức năng: Thêm loại nhà
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Loại nhà >> Điều hướng >> Thêm loại nhà
- Mô tả chức năng: Nhân viên đăng nhập thêm loại nhà ở
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Nhân viên thêm thông tin loại nhà ở.

*s. Chức năng sửa loại nhà ở*

- Tên chức năng: Sửa loại nhà
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Loại nhà >> Điều hướng >> Sửa loại nhà
- Mô tả chức năng: Nhân viên đăng nhập sửa loại nhà ở
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Nhân viên sửa thông tin loại nhà ở.

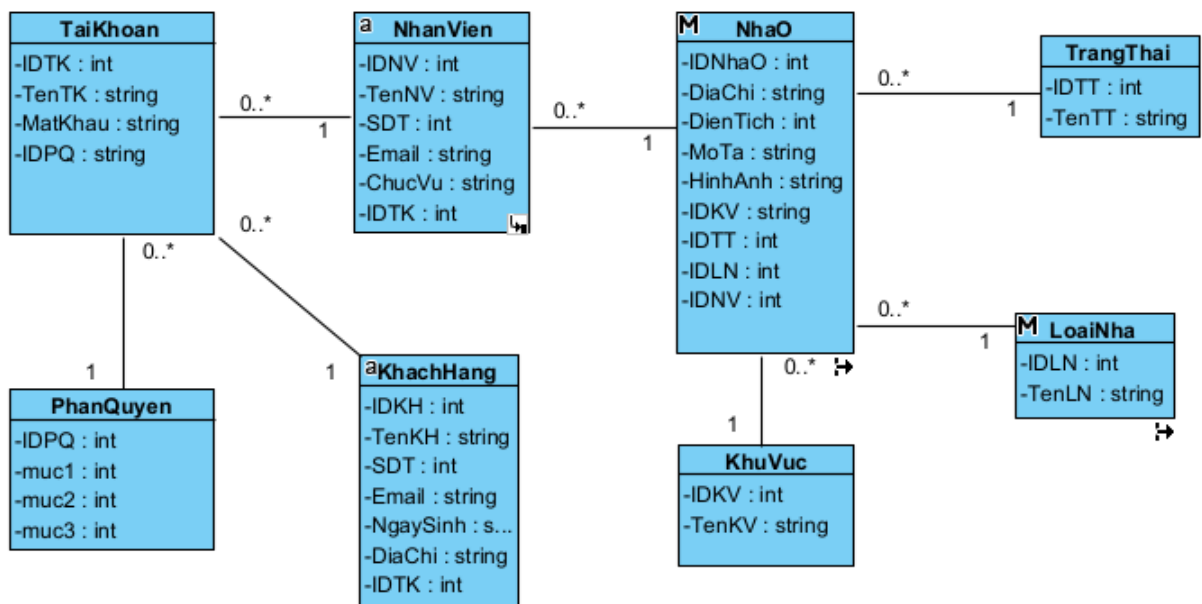


*t. Chức năng xóa loại nhà ở*

- Tên chức năng: Xóa loại nhà
- Đường dẫn: Trang chủ >> Điều hướng >> Đăng nhập admin >> Điều hướng >> Loại nhà >> Điều hướng >> Xóa loại nhà
- Mô tả chức năng: Nhân viên đăng nhập xóa loại nhà ở
  - Thông tin đăng nhập:
    - Tài khoản
    - Mật khẩu
  - Nhân viên xóa thông tin loại nhà ở.

### 2.3.3. Biểu đồ lớp thực thể

Phân tích yêu cầu người dùng, xác định được các thực thể lưu trữ dữ liệu trong hệ thống và mô hình hóa bằng biểu đồ lớp thực thể.

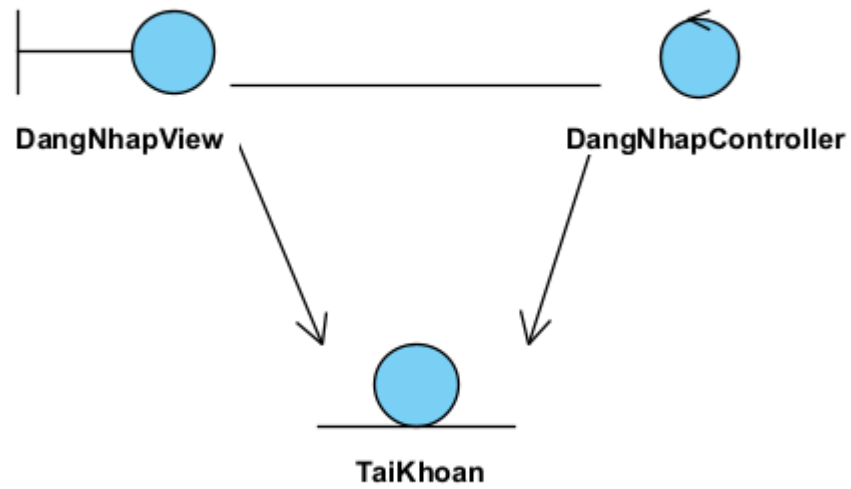


Hình 10: Biểu đồ lớp thực thể

### 2.3.4. Biểu đồ phân tích ca sử dụng

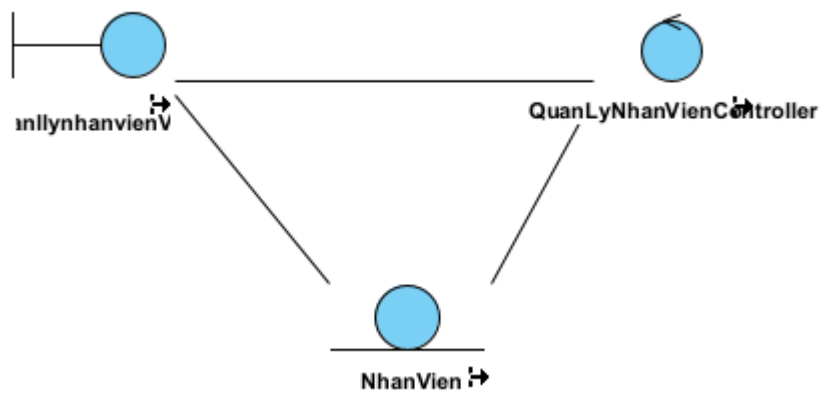
Tiếp theo tiến hành phân tích lần lượt từng ca sử dụng, từ ca sử dụng tổng quát đến ca sử dụng phân rã. Từ kịch bản luồng sự kiện trong mô tả chi tiết của từng ca sử dụng để xác định các lớp tham gia thực ca sử dụng và mô hình hóa bằng biểu đồ lớp VOPC cho từng ca sử dụng.

a. Biểu đồ phân tích ca sử dụng của use case Đăng nhập



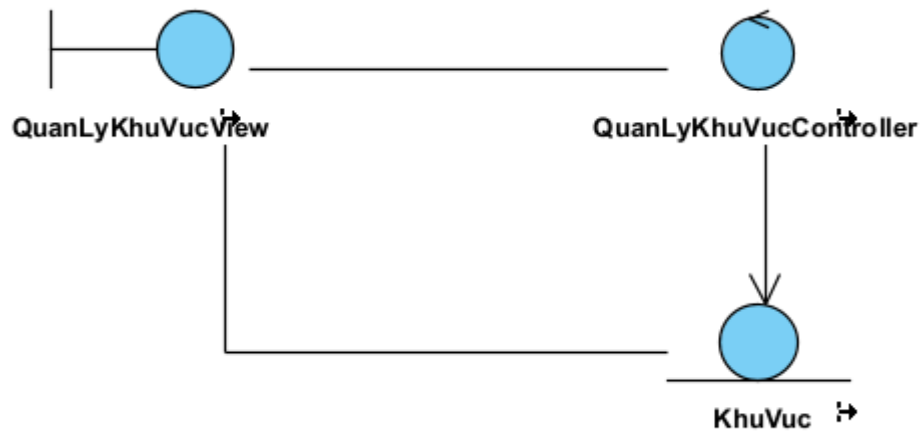
Hình 11: Biểu đồ phân tích ca sử dụng của use case Đăng nhập

b. Biểu đồ phân tích ca sử dụng của use case Nhân viên



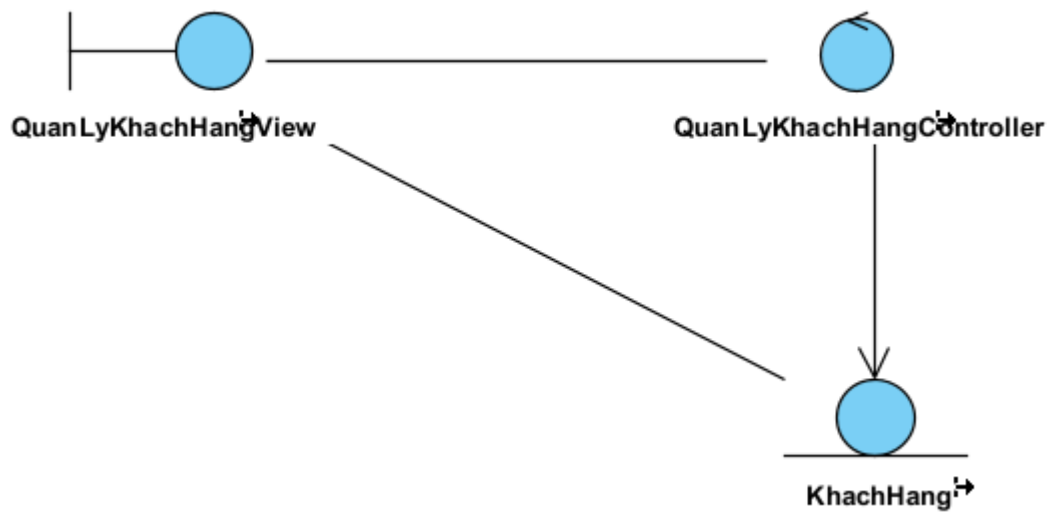
Hình 12: Biểu đồ phân tích ca sử dụng của use case Nhân viên

c. Biểu đồ phân tích ca sử dụng của use case Khu vực



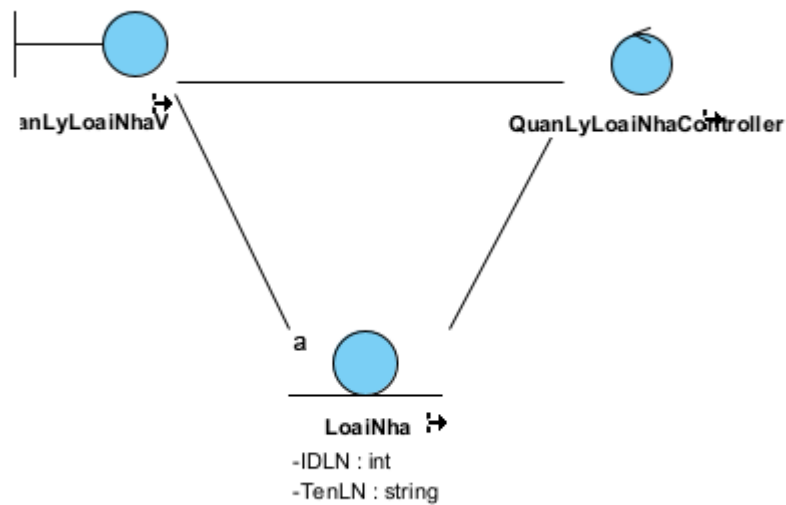
Hình 13: Biểu đồ phân tích ca sử dụng của use case Khu vực

d. Biểu đồ phân tích ca sử dụng của use case Khách hàng



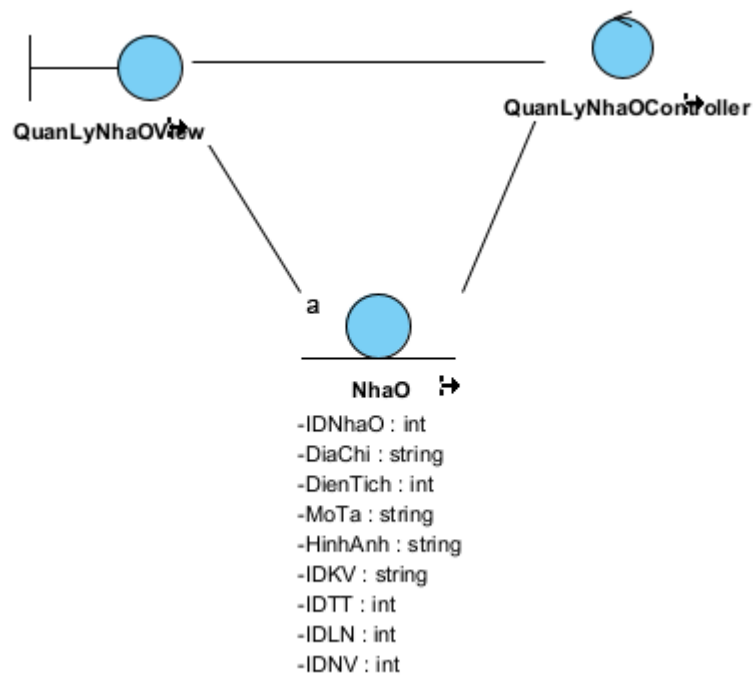
Hình 14: Biểu đồ phân tích ca sử dụng của use case Khách hàng

e. Biểu đồ phân tích ca sử dụng của use case Loại nhà



Hình 15: Biểu đồ phân tích ca sử dụng của use case Loại nhà

f. Biểu đồ phân tích ca sử dụng của use case Nhà ở

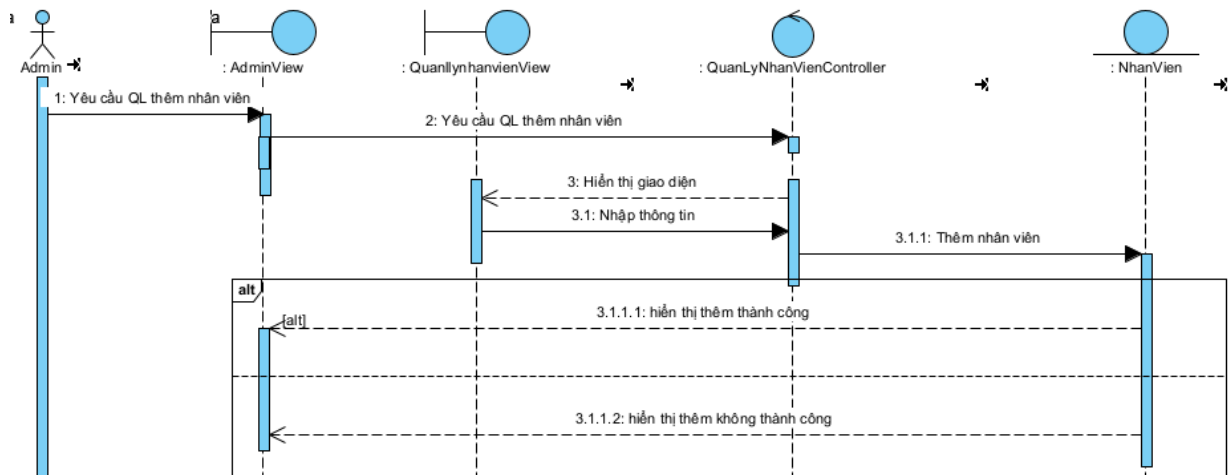


Hình 16: Biểu đồ phân tích ca sử dụng của use case Nhà ở

### 2.3.5. Biểu đồ tương tác

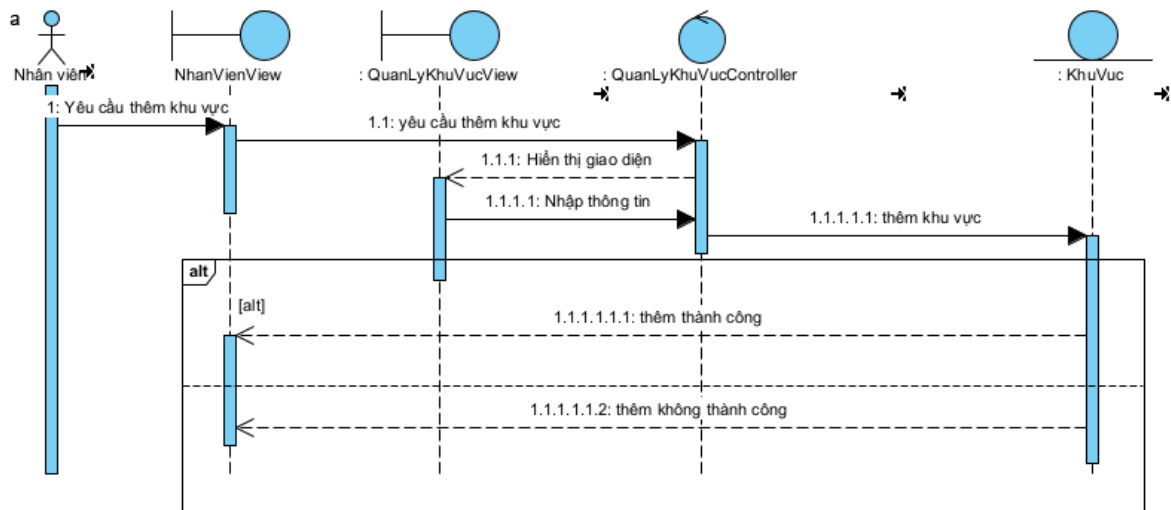
Tiếp theo biểu đồ tuần tự được sử dụng để mô hình hóa các chuỗi tương tác giữa các đối tượng là thể hiện của các lớp tham gia thực hiện ca sử dụng.

*a. Quản lý thêm nhân viên*



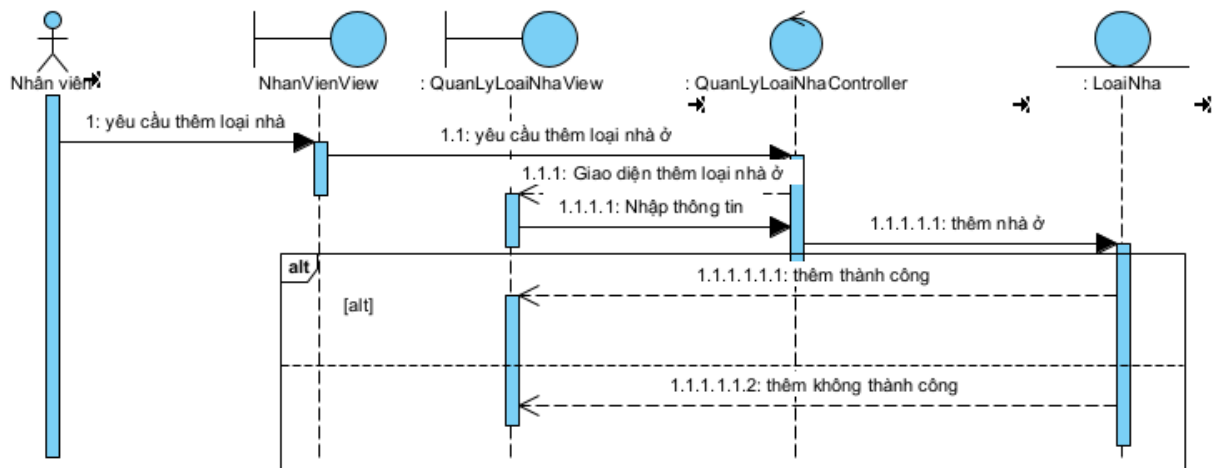
Hình 17: Biểu đồ tuần tự quản lý thêm nhân viên

*b. Quản lý thêm khu vực*



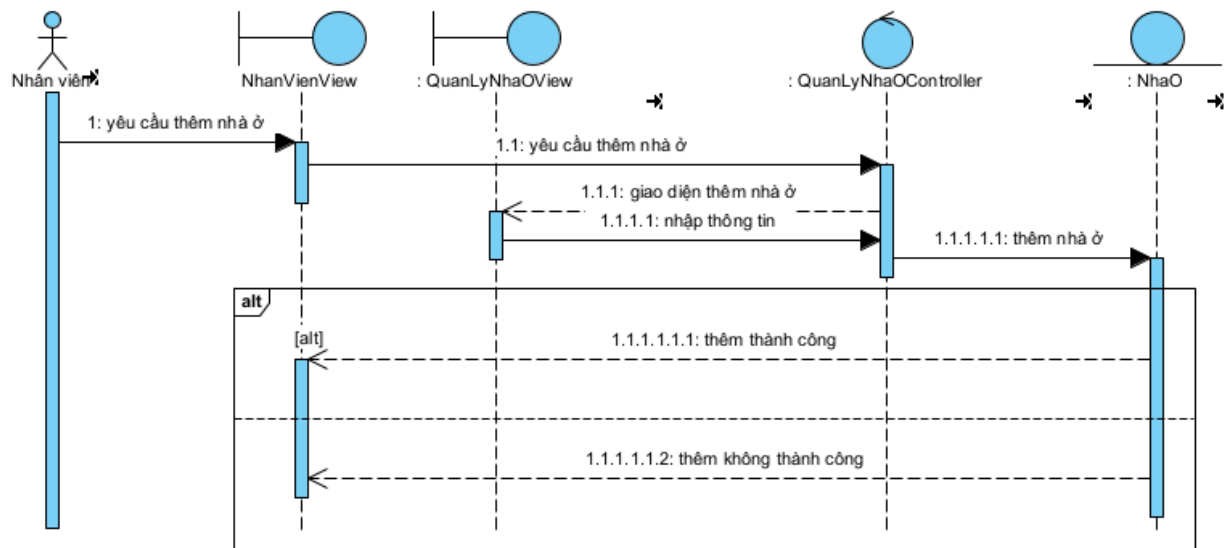
Hình 18: Biểu đồ tuần tự quản lý thêm khu vực

c. Quản lý thêm loại nhà



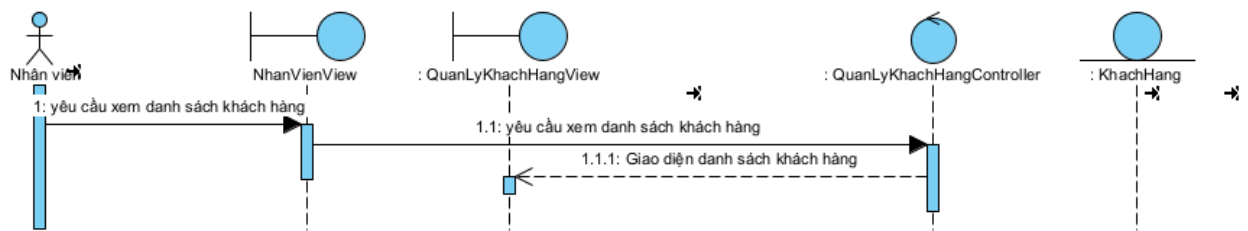
Hình 19: Biểu đồ tuần tự quản lý thêm loại nhà

d. Quản lý thêm nhà ở



Hình 20: Biểu đồ tuần tự quản lý thêm nhà ở

e. Quản lý xem danh sách khách hàng



Hình 21: Biểu đồ tuần tự quản lý xem danh sách khách hàng

### 2.3.6. Thiết kế cơ sở dữ liệu

a. Bảng phân quyền

Bảng phân quyền là để phân ra khách hàng, nhân viên và admin sẽ có các quyền vào các trang khác nhau.

Bảng 3: Bảng phân quyền

Tên trường	Kiểu dữ liệu	Kích thước	Mô tả
IDPQ	Int		Tự tăng
Muc1	Int		Mức 1
Muc2	Int		Mức 2
Muc3	Int		Mức 3

*b. Bảng tài khoản*

Bảng tài khoản lưu thông tin đăng nhập của người dùng.

*Bảng 4: Bảng tài khoản*

Tên trường	Kiểu dữ liệu	Kích thước	Mô tả
IDTK	Int		Tự tăng
TenTK	Nvarchar	50	Tên tài khoản
MatKhau	Nvarchar	255	Mật khẩu
IDPQ	Int		Id bảng phân quyền

*c. Bảng nhân viên*

Bảng nhân viên để lưu thông tin của nhân viên.

*Bảng 5: Bảng nhân viên*

Tên trường	Kiểu dữ liệu	Kích thước	Mô tả
IDNV	Int		Tự tăng
TenNV	Nvarchar	50	Tên nhân viên
SĐT	Nvarchar	12	Số điện thoại
Email	Nvarchar	255	Email
ChucVu	Nvarchar	50	Chức vụ



<b>Tên trường</b>	<b>Kiểu dữ liệu</b>	<b>Kích thước</b>	<b>Mô tả</b>
IDTK	Int		Id bảng tài khoản

*d. Bảng khách hàng*

Bảng khách hàng để lưu thông tin của khách hàng khi đăng ký.

*Bảng 6: Bảng khách hàng*

<b>Tên trường</b>	<b>Kiểu dữ liệu</b>	<b>Kích thước</b>	<b>Mô tả</b>
IDKH	Int		Tự tăng
TenKH	Nvarchar	50	Tên khách hàng
SDT	Nvarchar	12	Số điện thoại
Email	Nvarchar	255	Email
NgaySinh	Nvarchar	Max	Ngày sinh
DiaChi	Nvarchar	100	Địa chỉ
IDTK	Int		Id bảng tài khoản

## e. Bảng nhà ở

Bảng nhà ở lưu thông tin nhà ở để đăng bài lên trang web.

Bảng 7: Bảng nhà ở

Tên trường	Kiểu dữ liệu	Kích thước	Mô tả
IDNhaO	Int		Tự tăng
DiaChi	Nvarchar	100	Địa chỉ
DienTich	Float		Diện tích
MoTa	Nvarchar	Max	Mô tả
HinhAnh	Nvarchar	Max	Hình ảnh
IDNV	Int		Id bảng nhân viên
IDKV	Int		Id bảng khu vực
IDTT	Int		Id bảng trạng thái
IDLN	Int		Id bảng loại nhà

*f. Bảng khu vực*

Bảng khu vực để có thể phân nhà ở ra từng khu vực phục vụ cho việc lọc nhà dễ dàng hơn.

*Bảng 8: Bảng khu vực*

Tên trường	Kiểu dữ liệu	Kích thước	Mô tả
IDKV	Int		Tự tăng
TenKV	Nvarchar	100	Tên khu vực

*g. Bảng trạng thái*

Bảng trạng thái để người dùng biết được nhà ở này đang ở trạng thái chưa cho thuê hay đang cho thuê để dễ dàng tìm nhà nhanh chóng.

*Bảng 9: Bảng trạng thái*

Tên trường	Kiểu dữ liệu	Kích thước	Mô tả
IDTT	Int		Tự tăng
TenTT	Nvarchar	100	Tên trạng thái

*h. Bảng loại nhà*

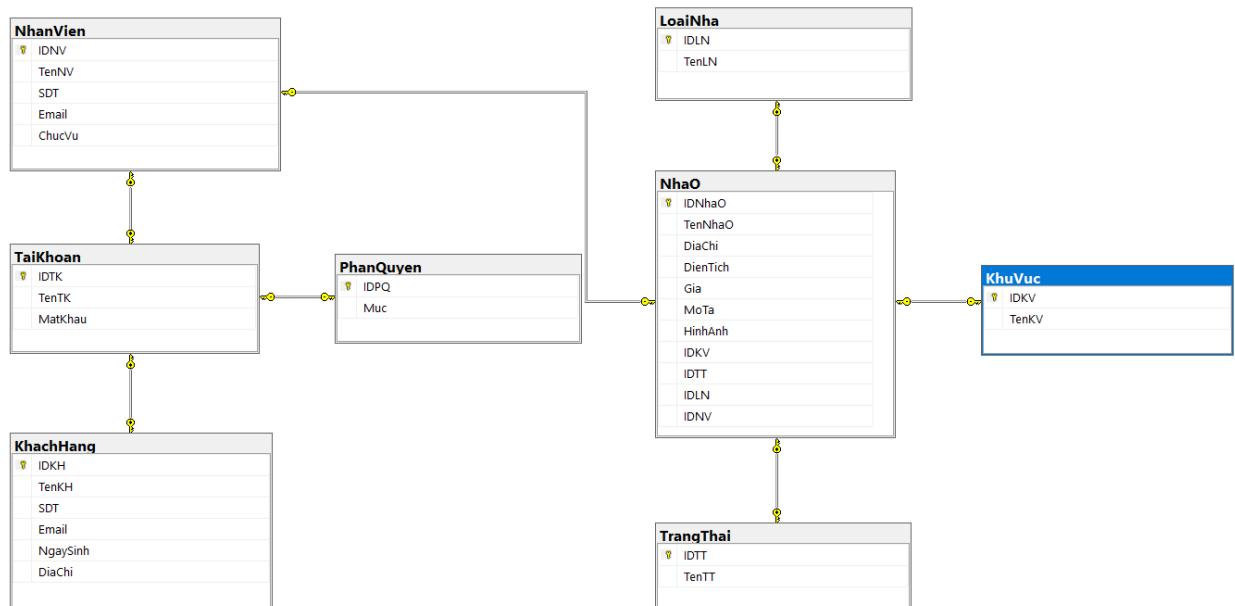
Bảng loại nhà để người dùng có thể lọc nhà theo yêu cầu như nhà chung cư hay biệt thự hoặc nhà trong ngõ, nhà mặt đường,... dễ dàng đáp ứng yêu cầu của khách hàng tìm kiếm nhanh chóng.

Bảng 10: Bảng loại nhà

Tên trường	Kiểu dữ liệu	Kích thước	Mô tả
IDLN	Int		Tự tăng
TenLN	Nvarchar	100	Tên loại nhà

### i. Database Diagrams

Đầu vào để thiết kế cơ sở dữ liệu là biểu đồ lớp thực thể và bảng cơ sở dữ liệu.



Hình 22: Database Diagrams

**Tiểu kết:** Chương 2 đã trình bày về quá trình phân tích và thiết kế hệ thống website cho thuê nhà. Đầu tiên, các yêu cầu bài toán được xác định rõ ràng, bao gồm các chức năng cần thiết như đăng tin cho thuê, tìm kiếm nhà, quản lý tài khoản người dùng. Tiếp theo, hệ thống được phân tích kỹ lưỡng để hiểu rõ các luồng dữ liệu, tương tác giữa các thành phần và các nghiệp vụ chính của ứng dụng.

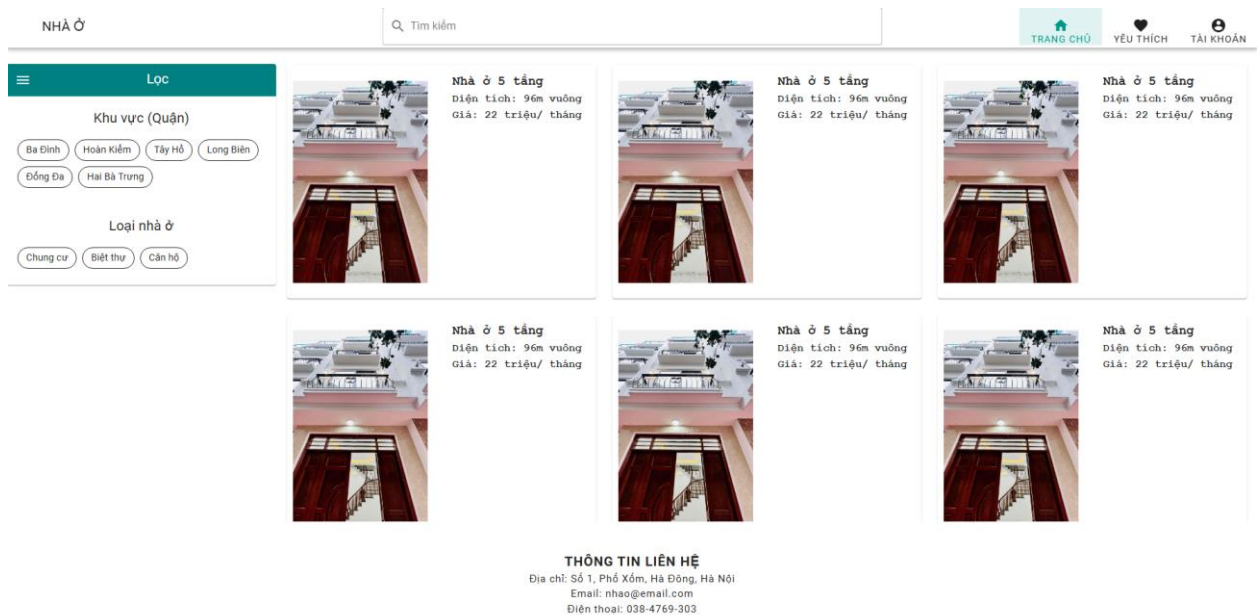
## Chương 3: KẾT QUẢ CHƯƠNG TRÌNH

### 3.1. Giao diện khách hàng

Một số giao diện mà khách hàng sẽ sử dụng.

#### 3.1.1. Giao diện trang chủ

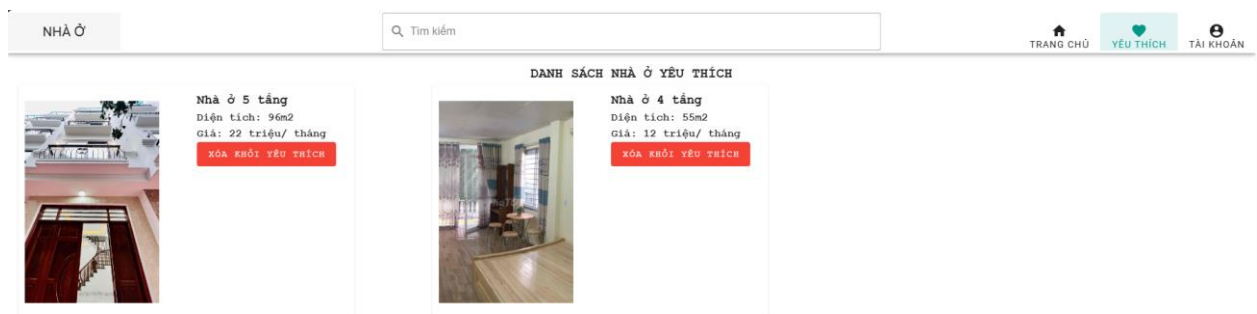
Hình ảnh trang chủ mà khách hàng sẽ nhìn thấy đầu tiên. Người dùng có thể tìm kiếm nhà ở theo nhu cầu hoặc lọc nhà ở theo khu vực hoặc loại nhà, có thể xem nhà ở đã thêm vào yêu thích, xem thông tin tài khoản của mình.



Hình 23: Giao diện trang chủ

#### 3.1.2. Giao diện trang yêu thích

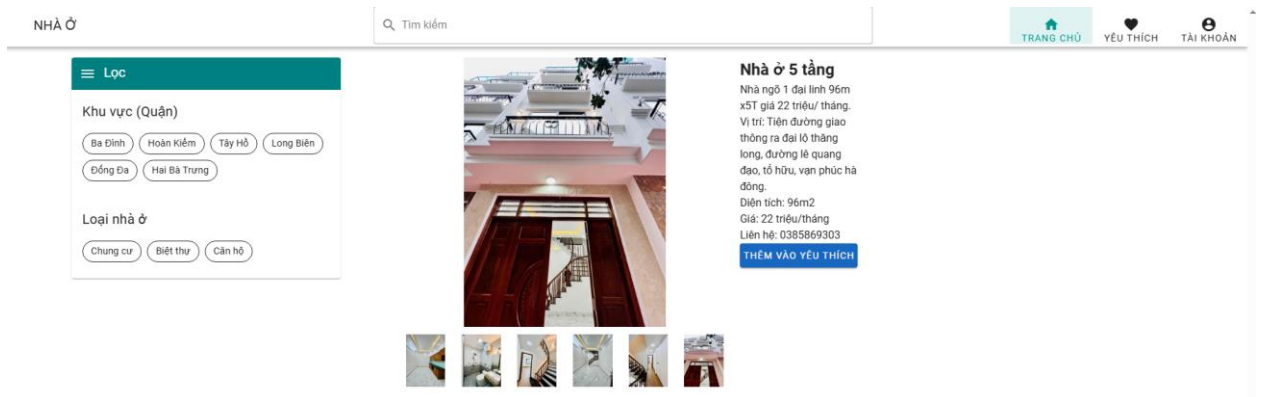
Khách hàng có thêm thêm nhà mà mình lưu ý vào thích và xóa những nhà không quan tâm nữa.



Hình 24: Giao diện trang yêu thích

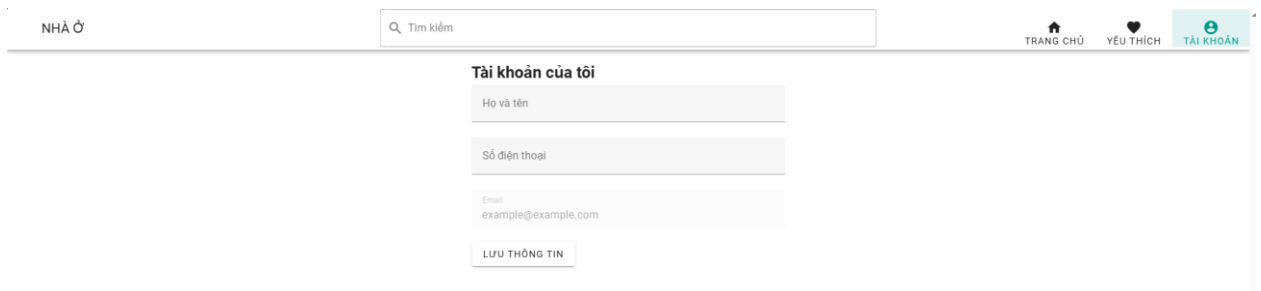
### 3.1.3. Giao diện trang chi tiết nhà ở

Xem chi tiết thông tin nhà ở và có thể liên hệ với chủ nhà. Trong đây có thông tin nhà gồm: Địa chỉ, giá, diện tích, liên hệ.



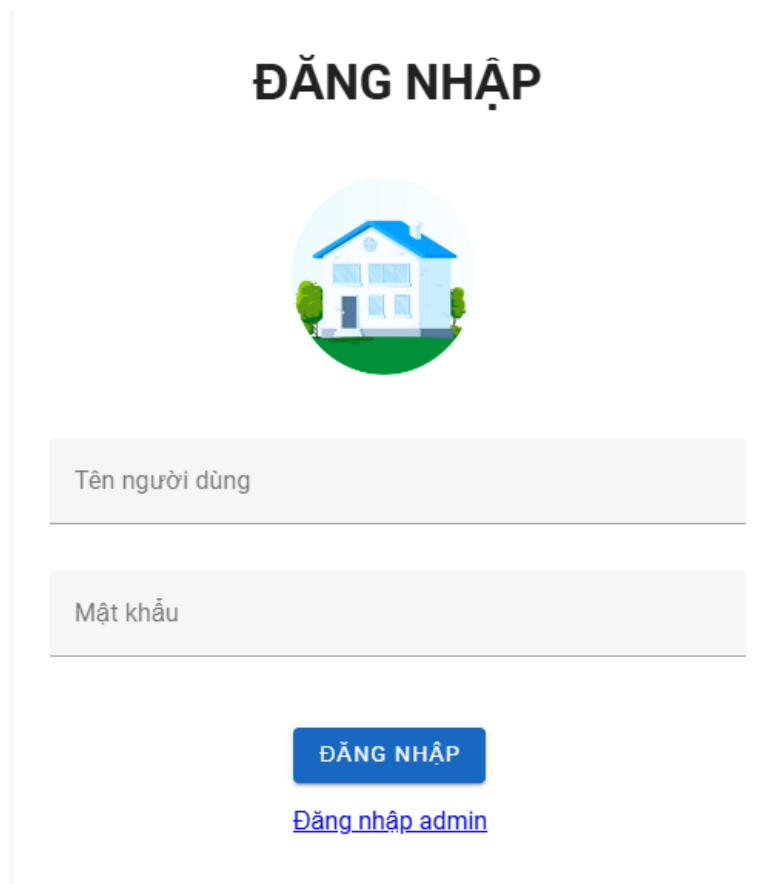
Hình 25: Giao diện trang chi tiết

### 3.1.4. Giao diện trang tài khoản của tôi




Hình 26: Giao diện trang tài khoản của tôi

### 3.1.5. Giao diện trang đăng nhập



**ĐĂNG NHẬP**



Tên người dùng

Mật khẩu

**ĐĂNG NHẬP**

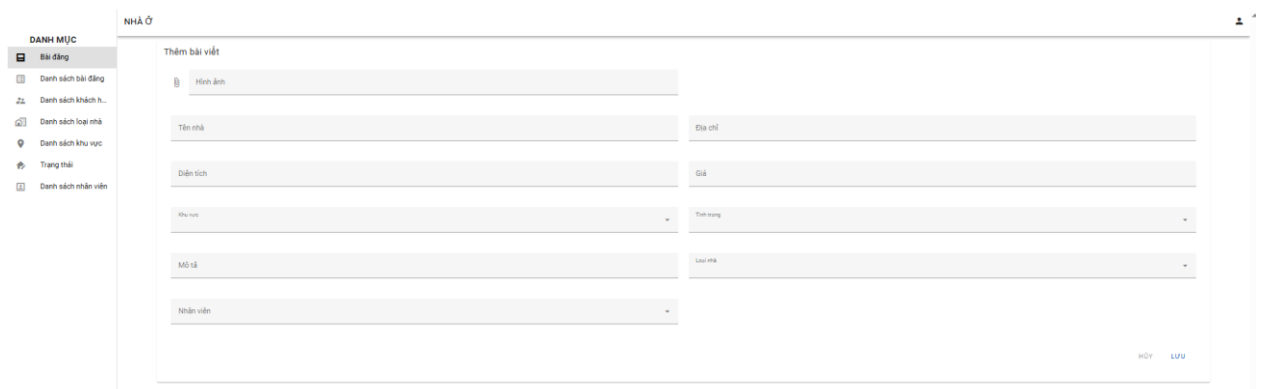
[Đăng nhập admin](#)

Hình 27: Giao diện trang đăng nhập

## 3.2. Giao diện admin

### 3.2.1. Giao diện trang đăng bài nhà ở

Nhân viên có thể đăng bài về nhà ở.



**DANH MỤC**

- Bài đăng**
- Danh sách bài đăng
- Danh sách khách h...
- Danh sách loại nhà
- Danh sách khu vực
- Trạng thái
- Danh sách nhân viên

**NHÀ Ở**

Thêm bài viết

Hình ảnh

Tên nhà

Địa chỉ

Diện tích

Giá

Phu vụ

Tình trạng

Mô tả

Loại nhà

Nhân viên

HỖY LƯU

Hình 28: Giao diện trang đăng bài

### 3.2.2. Giao diện danh sách bài đăng

Hiển thị danh sách các bài đăng đã đăng trên trang web

STT	Loại nhà	Tên nhà	Địa chỉ	Diện tích	Giá	Khu vực	Tình trạng nhà	Hình ảnh	Mã số	Chức năng
1	3	nhà 4 tầng	Đại Mỗ, Nam Từ Liêm, Hà Nội	50	12000000	1	1	nhà.jpg	Nhà 4 tầng	

Hình 29: Giao diện danh sách bài đăng

### 3.2.3. Giao diện trang danh sách loại nhà

Hiện thị danh sách loại nhà ở có thể thêm, sửa, xóa loại nhà.

Mã loại nhà	Loại nhà	Chức năng
1	Nhà chung cư	
2	Nhà trong ngõ	
3	Nhà mặt đường	
4	Biệt thự sân cỏ	

Hình 30: Giao diện trang danh sách loại nhà

### 3.2.4. Giao diện trang danh sách khu vực

Hiện thị danh sách khu vực nhà ở các quận – huyện ở Hà Nội.

Mã khu vực	Khu vực	Chức năng
1	Nam Từ Liêm	
2	Hà Đông	
3	BTL	

Hình 31: Giao diện trang danh sách khu vực

**Tiểu kết:** Chương 3 đã minh họa rõ ràng và cụ thể kết quả của quá trình phát triển ứng dụng, từ thiết kế giao diện đến các chức năng quan trọng.



## KẾT LUẬN

### 1. Kết quả đạt được

#### *a. Về kiến thức và học tập*

- Nắm vững kiến thức lý thuyết cơ bản phục vụ tốt cho việc thiết kế chương trình.
- Thiết kế được CSDL tương đối hoàn chỉnh, đáp ứng tốt cho việc viết chương trình
- Xây dựng được một trang web tương đối đầy đủ các chức năng cần thiết cho việc thuê nhà và cho thuê nhà ở.
- củng cố lại các kiến thức đã học được, đặc biệt là kỹ năng phân tích, giải quyết vấn đề.
- Biết cách áp dụng kiến thức lý thuyết vào ứng dụng thực tế.

#### *b. Về phần mềm*

- Hoàn thiện đầy đủ chức năng cơ bản của một phần mềm: Thêm, sửa, xóa, cập nhật thông tin, tìm kiếm căn bản.
- Giao diện chương trình thân thiện, dễ sử dụng, trực quan và linh hoạt

### 2. Hạn chế của đề tài

- Nghiên cứu các vấn đề còn chưa đầy đủ, do thời gian còn hạn chế.
- Kiến thức lập trình còn hạn chế, nên tính chuyên nghiệp của chương trình chưa cao, bắt lỗi chưa kỹ ở một số chỗ và một số tính năng chưa được tối ưu.
- Chương trình chưa thực sự đầy đủ các tính năng mà bản thân mong muốn như phân quyền người dùng giữa admin, nhân viên, khách hàng.

### 3. Hướng phát triển

- Thiết kế chương trình và chỉnh sửa giao diện mang tính chuyên nghiệp hơn.
- Tìm hiểu các biện pháp về bảo mật cho trang web, từ đó áp dụng cho bài.

### 4. Bổ sung thêm một số chức năng

Khách hàng có thêm ấn vào số điện là sẽ liên hệ được với bên cho thuê, tìm kiếm nâng cao.

## **TÀI LIỆU THAM KHẢO**

1. Tài liệu bài giảng VueJS cơ bản – ThS. Phạm Văn Tiệp
2. Tài liệu bài giảng RESTFul Service – ThS. Phạm Văn Tiệp
2. VueJS là gì? Ứng dụng và một vài tính năng nổi trội của VueJS ([glints.com](https://glints.com))
3. ASP.NET Core Là Gì? Tổng Quan Về ASP.NET Core - .NET Core VN
4. Vuex là gì và tại sao nên dùng nó ([viblo.asia](https://viblo.asia))
5. Axios là gì? Cơ bản về Axios trong VueJs ([viblo.asia](https://viblo.asia))
6. Giáo trình Phân tích thiết kế hướng đối tượng với UML – TS.Chu Thị Minh Huệ (Chủ biên), TS. Nguyễn Minh Tiến, TS. Nguyễn Minh Quý