

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

# **ĐỒ ÁN TỐT NGHIỆP**

## **Website thuê nhà trọ**

**Phương Trung Đức**

duc.pt173030@sis.hust.edu.vn

**Ngành Khoa học máy tính**

**Chuyên ngành Công nghệ phần mềm**

**Giảng viên hướng dẫn:** TS. Trịnh Tuấn Đạt

**Bộ môn:** Công nghệ phần mềm

**Viện:** Công nghệ thông tin – Truyền thông

**HÀ NỘI, 07/2022**

# 1. Lời cam kết

Họ và tên sinh viên: Phương Trung Đức

Điện thoại liên lạc: 0979336552 Email: duc.pt173030@sis.hust.edu.vn

Lớp: KHMT.04-K62 Hệ đào tạo: Kỹ sư chính quy

Tôi – *Phương Trung Đức* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Trịnh Tuấn Đạt*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

	<i>Hà Nội, ngày    tháng    năm</i>  Tác giả ĐATN  <i>Họ và tên sinh viên</i>
--	---

## **2.Lời cảm ơn**

Lời đầu tiên, em xin gửi lời cảm ơn tới TS.Trịnh Tuấn Đạt đã tận tình hướng dẫn, giúp đỡ em trong quá trình thực hiện Đồ án tốt nghiệp của mình.

Em xin chân thành cảm ơn các thầy, cô giáo Viện Công nghệ thông tin và Truyền thông của trường Đại học Bách Khoa Hà Nội đã dìu dắt, dạy dỗ em cả về kiến thức chuyên môn và tinh thần học tập để em có được những kiến thức thực hiện đồ án tốt nghiệp của mình.

Em xin chân thành cảm ơn các thầy cô trong ban giám hiệu nhà trường, các phòng ban đã giúp đỡ tạo điều kiện tốt nhất cho em trong suốt thời gian học tập tại trường. Tuy có nhiều cố gắng trong quá trình học tập hay trong quá trình làm đồ án tốt nghiệp, bản thân em cũng không thể tránh khỏi những thiếu sót, em rất mong được sự góp ý, nhận xét của tất cả các thầy cô giáo và mọi người để bản thân em của em được hoàn thiện hơn.

### 3. Tóm tắt

Đã từ lâu nhu cầu thuê và cho thuê nhà trọ luôn là một trong những vấn đề thường trực trong cuộc sống. Cùng với nhu cầu thuê và cho thuê nhà trọ, việc chia sẻ thông tin về nhà trọ cũng trở thành một vấn đề của xã hội. Trong hai năm trở lại đây, đại dịch covid-19 khiến cho rất nhiều nhà trọ bị bỏ trống. Vì vậy, khi xã hội trở lại trạng thái bình thường mới, nhu cầu nhà trọ thực sự trở thành một vấn đề nóng. Tuy nhiên, việc chia sẻ thông tin về nhà trọ giữa người cho thuê và người đi thuê vẫn gặp rất nhiều khó khăn.

Đã có rất nhiều giải pháp cho việc chia sẻ thông tin nhà trọ được sử dụng như: rải tờ rơi, dán quảng cáo, liên hệ trung gian ... Tuy nhiên những cách này vẫn tồn tại rất nhiều nhược điểm, nhất là việc số lượng người tiếp nhận được thông tin này chưa cao. Gần đây, rất nhiều trang web cho thuê nhà trọ ra đời, giúp cho nhu cầu chia sẻ thông tin nhà trọ phần nào được giải quyết. Có thể thấy, việc chia sẻ thông tin nhà trọ thông qua internet mà cụ thể là qua một trang web giúp cho người cho thuê cũng như người thuê trọ tiết kiệm được rất nhiều thời gian trong việc chia sẻ cũng như tiếp cận các thông tin về nhà trọ, tối ưu hóa được quá trình thuê nhà.

Trên cơ sở đó, ĐATN này hướng tới việc xây dựng một trang web cho thuê nhà trọ, giải quyết được nhu cầu chia sẻ thông tin giữa hai bên thuê trọ và bên cho thuê. Ngoài ra, còn đáp ứng các nhu cầu khác như đảm bảo việc liên lạc giữa hai bên, cung cấp các thông tin chi tiết về nhà trọ, các khu vực lân cận... Đây sẽ là các nội dung chính mà em sẽ trình bày trong ĐATN này.

## **4. Abstract**

Mục này khuyến khích sinh viên viết lại mục “Tóm tắt” đồ án tốt nghiệp ở trang trước bằng tiếng Anh. Phần này phải có đầy đủ các nội dung như trong phần tóm tắt bằng tiếng Việt.

Sinh viên không nhất thiết phải trình bày mục này. Nhưng nếu lựa chọn trình bày, sinh viên cần đảm bảo câu từ và ngữ pháp chuẩn tắc, nếu không sẽ có tác dụng ngược, gây phản cảm.

## 5. Mục lục

<b>Lời cam kết</b>	<b>ii</b>
<b>Lời cảm ơn</b>	<b>iii</b>
<b>Tóm tắt</b>	<b>iv</b>
<b>Abstract</b>	<b>v</b>
<b>MỤC LỤC</b>	<b>vi</b>
<b>Danh mục hình vẽ</b>	<b>xi</b>
<b>DANH MỤC BẢNG</b>	<b>xiii</b>
<b>Danh mục các từ viết tắt</b>	<b>xiv</b>
<b>Danh mục thuật ngữ</b>	<b>xv</b>
<b>Chương 1 Giới thiệu đề tài</b>	<b>1</b>
1.1 Đặt vấn đề	1
1.2 Mục tiêu và phạm vi đề tài	2
1.3 Định hướng giải pháp	2
1.4 Bố cục đồ án	3
<b>Chương 2 Khảo sát và phân tích yêu cầu</b>	<b>4</b>
2.1 Khảo sát hiện trạng	4
2.2 Tổng quan chức năng	4
2.2.1 Biểu đồ usecase tổng quan	4
	6

2.2.2 Biểu đồ usecase phân rã quản lý bài đăng	7
2.2.3 Biểu đồ usecase phân rã quản lý người dùng	8
2.2.4 Biểu đồ usecase phân rã thanh toán phí bài đăng	9
2.2.5 Biểu đồ usecase phân rã tìm kiếm bài đăng	10
2.2.6 Biểu đồ usecase phân rã Quản lý liên hệ	11
2.2.7 Biểu đồ usecase phân rã Quản lý danh sách bài đăng yêu thích (Wishlist)	12
2.2.8 Biểu đồ usecase phân rã Quản lý tài khoản	13
2.2.9 Quy trình nghiệp vụ	13
2.2.9.1 Quy trình thêm bài đăng mới	13
2.2.9.2 Quy trình liên hệ với người sở hữu bài đăng	15
2.2.9.3 Quy trình chỉnh sửa bài đăng	17
2.3 Đặc tả chức năng	19
2.3.1 Đặc tả usecase Liên hệ với người đăng bài	19
2.3.2 Đặc tả usecase Chỉnh sửa bài đăng	21
2.3.3 Đặc tả usecase Thêm bài đăng	22
2.4 Yêu cầu phi chức năng	22
2.4.1 Yêu cầu chung	22
2.4.2 Yêu cầu về giao diện người dùng	23
2.4.3 Yêu cầu về bảo mật	23
2.4.4 Yêu cầu khác	23
<b>Chương 3 Công nghệ sử dụng</b>	<b>24</b>
3.1 Lập trình front-end	24
3.1.1 Javascript	24
3.1.2 VueJS	25
3.2 Lập trình back-end	25
3.2.1 ASP.NET Core	25

3.2.2 Entity Framework	26
3.2.3 Sql Server	28
<b>Chương 4 Phát triển và triển khai ứng dụng</b>	<b>30</b>
4.1 Thiết kế kiến trúc	30
4.1.1 Lựa chọn kiến trúc phần mềm	30
4.1.1.1 Kiến trúc client-server	30
4.1.1.2 Kiến trúc MVVM (Model-View-Viewmodel)	31
4.1.1.3 Kiến trúc 3-layer	32
4.1.2 Thiết kế tổng quan	34
4.1.2.1 Biểu đồ phụ thuộc gói với ứng dụng bên phía client	34
4.1.2.2 Biểu đồ phụ thuộc gói với ứng dụng bên phía server	35
4.1.3 Thiết kế chi tiết gói	36
4.1.3.1 Thiết kế chi tiết gói phía server	36
4.2 Thiết kế chi tiết	36
4.2.1 Thiết kế giao diện	36
4.2.1.1 4.2.1.1. Giao diện đăng kí	39
4.2.1.2 4.2.1.2: Giao diện trang chủ	39
4.2.1.3 4.2.1.3: Giao diện thông tin chi tiết nhà trọ	40
4.2.1.4 4.2.1.4: Giao diện quản lý tài khoản	40
4.2.1.5 4.2.1.5: Giao diện quản lý bài đăng	41
4.2.1.6 4.2.1.6: Giao diện thông báo	41
4.2.2 Thiết kế lớp	42
4.2.2.1 Thiết kế lớp cho các lớp trong trình tự thêm nhà vào danh sách Wishlist	42
4.2.2.2 Thiết kế lớp cho các lớp trong trình tự đăng nhập	43



4.2.2.3 Thiết kế lớp cho các lớp trong trình tự liên hệ với người bán	44
4.2.3 Thiết kế cơ sở dữ liệu	46
4.3 Xây dựng ứng dụng	51
4.3.1 Thư viện và công cụ sử dụng	51
4.3.2 Kết quả đạt được	52
4.3.3 Minh hoạ các chức năng chính	53
4.4 Kiểm thử	56
4.5 Triển khai	57
<b>Chương 5 Các giải pháp và đóng góp nổi bật</b>	<b>58</b>
5.1 Giải pháp cho tính năng yêu cầu thời gian thực (real-time)	58
5.1.1 Đặt vấn đề	58
5.1.2 Giải pháp	58
5.1.3 Kết quả đạt được	60
5.2 Giải pháp cho việc triển khai thiết kế kiến trúc 3-layer	61
5.2.1 Đặt vấn đề	61
5.2.2 Giải pháp	61
5.2.2.1 Presentation Layer	61
5.2.2.2 Business Logic Layer (BLL)	62
5.2.2.3 Data Access Layer (DAL)	63
5.2.3 Kết quả đạt được	66
5.3 Giải pháp cho yêu cầu mở rộng trong tương lai	66
5.3.1 Đặt vấn đề	66
5.3.2 Giải pháp	67
5.3.3 Kết quả đạt được	68
5.4 Giải pháp cho việc đảm bảo công thái học, thân thiện với người dùng	68
5.4.1 Đặt vấn đề	68

5.4.2 Giải pháp	68
<b>Chương 6 Kết luận và hướng phát triển</b>	<b>69</b>
6.1 Kết luận	69
6.2 Hướng phát triển	69
<b>Tài liệu tham khảo</b>	<b>71</b>

## 6. Danh mục hình vẽ

HÌNH 1: USECASE TỔNG QUAN	5
HÌNH 2: PHÂN RÃ USECASE QUẢN LÝ BÀI ĐĂNG	7
HÌNH 3: PHÂN RÃ USE CASE QUẢN LÝ NGƯỜI DÙNG	8
HÌNH 4: PHÂN RÃ USE CASE THANH TOÁN PHÍ BÀI ĐĂNG	9
HÌNH 5: PHÂN RÃ USE CASE TÌM KIẾM BÀI ĐĂNG	10
HÌNH 6: PHÂN RÃ USE CASE QUẢN LÝ LIÊN HỆ	11
HÌNH 7: PHÂN RÃ USE CASE QUẢN LÝ DANH SÁCH BÀI ĐĂNG YÊU THÍCH (WISHLIST)	12
HÌNH 8: PHÂN RÃ USE CASE QUẢN LÝ TÀI KHOẢN	13
HÌNH 9: QUY TRÌNH THÊM BÀI ĐĂNG MỚI	15
HÌNH 10: QUY TRÌNH LIÊN HỆ VỚI NGƯỜI SỞ HỮU BÀI ĐĂNG	17
HÌNH 11: QUY TRÌNH CHỈNH SỬA BÀI ĐĂNG	19
HÌNH 12: THỐNG KÊ CÁC NGÔN NGỮ LẬP TRÌNH PHỔ BIẾN NHẤT THẾ GIỚI NĂM 2022	24
HÌNH 13: CẤU TRÚC CỦA ENTITY FRAMEWORK	27
HÌNH 14: THỐNG KÊ CÁC HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU ĐƯỢC SỬ DỤNG NHIỀU NHẤT NĂM 2022	29
HÌNH 15: KIẾN TRÚC CLIENT-SERVER	30
HÌNH 16: KIẾN TRÚC MVVM (MODEL-VIEW-VIEWMODEL)	31
HÌNH 17: MÔ HÌNH 3-LAYER	32
HÌNH 18: BIỂU ĐỒ PHỤ THUỘC GÓI VỚI ỨNG DỤNG BÊN PHÍA CLIENT	34
HÌNH 19: BIỂU ĐỒ PHỤ THUỘC GÓI VỚI ỨNG DỤNG BÊN PHÍA SERVER	35
HÌNH 20: THIẾT KẾ CHI TIẾT GÓI PHÍA SERVER	36
HÌNH 21: LAYOUT CHUNG CỦA WEB	37
HÌNH 22: GIAO DIỆN ĐĂNG KÍ	39
HÌNH 23: GIAO DIỆN TRANG CHỦ	39
HÌNH 24: GIAO DIỆN THÔNG TIN CHI TIẾT NHÀ TRÒ	40
HÌNH 25: GIAO DIỆN QUẢN LÝ TÀI KHOẢN	40
HÌNH 26: GIAO DIỆN QUẢN LÝ BÀI ĐĂNG	41
HÌNH 27: GIAO DIỆN THÔNG BÁO	42
HÌNH 28: BIỂU ĐỒ CÁC LỚP TRONG TRÌNH TỰ THÊM NHÀ VÀO DANH SÁCH WISHLIST	43
HÌNH 29: BIỂU ĐỒ TRÌNH TỰ BIỂU DIỄN TRÌNH TỰ THÊM NHÀ VÀO DANH SÁCH WISHLIST CỦA MỘT USER	43
HÌNH 30: BIỂU ĐỒ CÁC LỚP TRONG TRÌNH TỰ ĐĂNG NHẬP	44
HÌNH 31: BIỂU ĐỒ TRÌNH TỰ BIỂU DIỄN TRÌNH TỰ ĐĂNG NHẬP	44
HÌNH 32: BIỂU ĐỒ CÁC LỚP TRONG TRÌNH TỰ LIÊN HỆ VỚI NGƯỜI BÁN	45
HÌNH 33: BIỂU ĐỒ TRÌNH TỰ BIỂU DIỄN TRÌNH TỰ LIÊN HỆ VỚI NGƯỜI BÁN	45
HÌNH 34: BIỂU ĐỒ THỰC THỂ LIÊN KẾT	46
HÌNH 35: TỔNG QUAN THIẾT KẾ DATABASE	48

HÌNH 36: GIAO DIỆN CHAT	53
HÌNH 37: GIAO DIỆN XEM CHI TIẾT BÀI ĐĂNG	53
HÌNH 38: GIAO DIỆN ĐĂNG KÝ TÀI KHOẢN	54
HÌNH 39: GIAO DIỆN TRANG CHỦ	55
HÌNH 40: GIAO DIỆN CHI TIẾT BÀI ĐĂNG	56
HÌNH 41: KẾT QUẢ KIỂM THỬ CODE VỚI SONARQUBE	57
HÌNH 42: MÔ HÌNH LÀM VIỆC CỦA SINALR	59
HÌNH 43: THIẾT KẾ LỚP ỨNG DỤNG HUB	60
HÌNH 44: THƯ MỤC PRESENTATION LAYER	61
HÌNH 45: BIỂU ĐỒ LỚP CHO PRESENTATION LAYER	62
HÌNH 46: THƯ MỤC BUSINESS LOGIC LAYER	62
HÌNH 47: BIỂU ĐỒ LỚP CHO BUSINESS LOGIC LAYER	63
HÌNH 48: THƯ MỤC DATA ACCESS LAYER	64
HÌNH 49: BIỂU ĐỒ LỚP CHO DATA ACCESS LAYER	64
HÌNH 50: DEPENDENCY INJECTION TRONG CLASS STARTUP	66
HÌNH 51: CHUẨN RESTFUL	67



## 7. Danh mục bảng

BẢNG 1 SO SÁNH HAI WEBSITE CHO THUÊ NHÀ	4
BẢNG 2: MÔ TẢ THÊM VỀ CÁC USECASE	5
BẢNG 3: QUY TRÌNH THÊM BÀI ĐĂNG MỚI	13
BẢNG 4: QUY TRÌNH LIÊN HỆ VỚI NGƯỜI SỞ HỮU BÀI ĐĂNG	15
BẢNG 5: QUY TRÌNH SỬA BÀI ĐĂNG	17
BẢNG 6: ĐẶC TẢ USECASE LIÊN HỆ VỚI NGƯỜI ĐĂNG BÀI	19
BẢNG 7: ĐẶC TẢ USECASE CHỈNH SỬA BÀI ĐĂNG	21
BẢNG 8: ĐẶC TẢ USECASE THÊM BÀI ĐĂNG	22
BẢNG 9: TIÊU CHÍ VÀ QUY ĐỊNH THIẾT KẾ	37
BẢNG 10: TIÊU CHÍ VÀ QUY ĐỊNH VỀ MÀU SẮC	37
BẢNG 11: MÔ TẢ VỀ THỰC THỂ MÔ HÌNH LIÊN KẾT	46
BẢNG 12: USER	48
BẢNG 13: FILE	49
BẢNG 14: HOUSE	49
BẢNG 15: HOUSETYPE	49
BẢNG 16: MESSAGE	50
BẢNG 17: POST	50
BẢNG 18: WISHLIST	50
BẢNG 19: DANH SÁCH THƯ VIỆN VÀ CÔNG CỤ SỬ DỤNG	51
BẢNG 20: THỐNG KÊ MÃ NGUỒN	52
BẢNG 21: MÔI TRƯỜNG TRIỂN KHAI CỤ THỂ	57

## 8. Danh mục các từ viết tắt

<b>API</b>	Application Programming Interface Giao diện lập trình ứng dụng
<b>EUD</b>	End-User Development Phát triển ứng dụng người dùng cuối
<b>GWT</b>	Google Web Toolkit Công cụ lập trình Javascript bằng Java của Google
<b>HTML</b>	HyperText Markup Language Ngôn ngữ đánh dấu siêu văn bản
<b>CNTT</b>	Công nghệ thông tin
<b>ĐATN</b>	Đồ án tốt nghiệp
<b>SV</b>	Sinh viên

## 9. Danh mục thuật ngữ

Browser	Trình duyệt
<b>Cache memory</b>	Bộ nhớ đệm
<b>E-commerce</b>	Thương mại điện tử
<b>Bloatware</b>	Ứng dụng nhà sản xuất tích hợp vào thiết bị
<b>Interpreter</b>	Trình thông dịch
<b>Compiler</b>	Trình biên dịch



# Chương 1 Giới thiệu đề tài

## 1.1 Đặt vấn đề

Tìm kiếm nhà trọ hiện nay là một nhu cầu thiết yếu của rất nhiều người, đặc biệt là với hai đối tượng chính là công nhân và sinh viên. Ở Hà Nội và các thành phố lớn, không khó để bắt gặp được những tờ rơi quảng cáo nhà trọ hay hình ảnh các sinh viên đi khắp nơi tìm nhà trọ phù hợp với nhu cầu cá nhân.

Trong bối cảnh công nghiệp hóa và hiện đại hóa của nước ta hiện nay, các khu công nghiệp và các công ty, xí nghiệp mới được liên tục thành lập. Đi cùng với đó là nhu cầu rất lớn của các tổ chức này về công nhân - lực lượng lao động sản xuất chính của họ. Theo thống kê của bộ xây dựng<sup>1</sup>, Hà Nội hiện có khoảng 326.000 doanh nghiệp với trên 2,5 triệu lao động; trong đó có 9 khu công nghiệp, khu chế xuất và khu công nghệ cao Hòa Lạc với 661 doanh nghiệp, 165.000 lao động, phần lớn là lao động ngoại tỉnh (chiếm trên 60%). Tuy nhiên, hiện nay mới có 3 khu công nghiệp: Thạch Thất-Quốc Oai, Thăng Long (Đông Anh), Phú Nghĩa (Chương Mỹ) có dự án nhà ở đáp ứng một phần nhu cầu của công nhân. Các khu công nghiệp còn lại đều chưa có nhà ở cho công nhân. Phần lớn công nhân lao động phải đi thuê và sống trong các phòng trọ.

Mặt khác, các trường đại học lớn cũng thường tập trung ở các thành phố lớn như Hà Nội, Hồ Chí Minh, Đà Nẵng. Theo thống kê của Bộ Giáo dục và Đào tạo<sup>2</sup>, năm học 2019-2020 có thêm đến gần 500.000 sinh viên mới. Theo đó, nhu cầu về nhà trọ cho lượng sinh viên này cũng tăng cao. Đặc biệt, trong bối cảnh dịch bệnh Covid, nhiều khóa sinh viên dù đã được tiếp nhận bởi các trường đại học nhưng học tập theo hình thức trực tuyến và chỉ mới bắt đầu trở lại giảng đường gần đây. Điều này làm cho nhu cầu về nhà trọ không những tăng mà còn tăng đột biến.

Thực trạng thuê nhà trọ hiện nay đang có nhiều vấn đề bất cập. Người sở hữu nhà trọ cũng như người đi thuê nhà trọ dù đều có nhu cầu thiết thực nhưng gặp vấn đề trong việc trao đổi

---

<sup>1</sup> <https://moc.gov.vn/vn/tin-tuc/1184/72048/som-giai-quyet-nhu-cau-nha-o-de-cong-nhan-%E2%80%98an-cu-lac-nghep%E2%80%99.aspx> truy cập lần cuối 29/07/2022

<sup>2</sup> <https://moet.gov.vn/thong-ke/Pages/thong-ko-giao-duc-dai-hoc.aspx?ItemID=7389> truy cập lần cuối 29/07/2022

thông tin. Rất nhiều chủ nhà trọ sở hữu nhiều căn phòng trọ cho thuê nhưng không biết phải chia sẻ thông tin nhà trọ qua đâu. Cách họ thường làm là chia sẻ thông tin qua các mạng xã hội như Facebook, Zalo ... Tuy nhiên, cách này chỉ giúp họ tiếp cận với số lượng nhỏ khách hàng tiềm năng. Bên cạnh đó, họ cũng thường phát tờ rơi hoặc dán quảng cáo trên tường quanh các khu phố lớn để quảng cáo thông tin nhà trọ. Nói chung, các cách làm này đều chưa đạt được hiệu quả tối đa. Với sự phát triển của các trang cho thuê nhà trọ gần đây, rất nhiều người cũng đã tìm đến các trang này để chia sẻ thông tin về nhà trọ nhưng số lượng không nhiều do hình thức này vẫn chưa thực sự phổ biến.

Trong khi đó, người đi thuê trọ gặp rất nhiều khó khăn trong việc tìm kiếm thông tin. Họ không biết phải tìm các thông tin về nhà trọ ở đâu hay liên hệ thế nào với các chủ nhà trọ. Họ cũng khó tìm được nhà trọ vừa gần nơi học tập và làm việc, vừa gần các khu vực có tập trung nhiều quán ăn, chợ, và siêu thị để thuận lợi hơn cho cuộc sống hàng ngày. Rất nhiều người còn phải lựa chọn cách trực tiếp đi dò tìm các nhà trọ trên các tuyến phố gần nơi làm việc, trường học để tự mình thu thập các thông tin cụ thể về nhà trọ. Nhiều trường hợp không tìm được nhà trọ phải ở ghép với người lạ hoặc ở nhờ nhà người thân, gây nhiều bất tiện trong cuộc sống.

Một trang web cho thuê nhà sẽ giải quyết được các vấn đề nêu trên. Đó là nơi mà người sở hữu nhà trọ cho thuê tìm đến để chia sẻ thông tin nhà trọ của họ, cũng là nơi mà người muốn thuê trọ ghé thăm để tìm kiếm được nhà trọ phù hợp với mình.

## 1.2 Mục tiêu và phạm vi đề tài

Đã có rất nhiều trang web cho thuê nhà trọ đã được phát triển. Hai trang web nổi tiếng nhất hiện nay về cho thuê nhà là [muonnha.com.vn](http://muonnha.com.vn)<sup>3</sup> và [phongtro123.com](http://phongtro123.com)<sup>4</sup>. Các trang web còn lại đều phát triển theo hình mẫu của hai trang web này, từ hình thức cho đến nội dung.

Hai trang web này đều đã đáp ứng được các yêu cầu cho một trang web thuê trọ khá tốt, đảm bảo tính năng quan trọng nhất cần có của một trang thuê nhà là chia sẻ thông tin về nhà trọ. Tuy nhiên, vẫn tồn tại những nhược điểm ở hai trang web này. Hai trang web đều chưa cung cấp cho người dùng phương thức liên hệ trực tiếp trên trang, chưa có chỉ dẫn cho người dùng về đường đi đến địa chỉ nhà trọ và các thông tin lân cận khu trọ. Đây đều là các thông tin người dùng thực sự quan tâm trong việc thuê nhà trọ.

Trên cơ sở đó, ĐATN này hướng tới việc xây dựng một ứng dụng web giúp người dùng chia sẻ thông tin về nhà trọ. Ứng dụng ngoài đáp ứng nhu cầu cơ bản của một trang web cho thuê

---

<sup>3</sup> <http://muonnha.com.vn/> truy cập lần cuối 29/07/2022

<sup>4</sup> <https://phongtro123.com> truy cập lần cuối 29/07/2022

nhà, giúp người dùng có thể tiếp cận được các thông tin về nhà trọ cũng như có thể đăng tải các thông tin nếu có nhu cầu cho thuê trọ. Mà còn phải đáp ứng nhu cầu giúp người dùng có thể liên hệ với chủ nhà. Hỗ trợ người dùng trong việc tìm kiếm địa chỉ nhà trọ và đảm bảo cung cấp các thông tin về khu vực lân cận. Từ đó, đáp ứng đầy đủ quá trình thuê nhà từ khi tìm kiếm thông tin, lựa chọn cho đến chốt phương án thuê.

Ngoài đảm bảo các chức năng cơ bản của một website thuê nhà, ĐATN cũng hướng tới xây dựng một trang web có giao diện thân thiện và hỗ trợ các chức năng để một người dùng mới cũng dễ dàng có thể tiếp cận với trang web.

### **1.3 Định hướng giải pháp**

Với sự hỗ trợ mạnh mẽ từ các thiết bị phần cứng cũng như các trình duyệt, các ứng dụng web đang là xu thế hàng đầu trong việc phát triển ứng dụng. Người dùng có thể sử dụng linh hoạt các ứng dụng này trên nhiều nền tảng thiết bị khác nhau như máy tính, điện thoại, tablet.... Chỉ cần có internet, người dùng có thể truy cập qua bất kì trình duyệt nào. Vì vậy, ĐATN này sẽ hướng tới xây dựng một ứng dụng web. trong khuôn khổ đề án chỉ dừng lại ở việc đáp ứng việc truy cập qua máy tính, nhưng vẫn đảm bảo đáp ứng nhu cầu mở rộng trong tương lai.

Với việc xây dựng một ứng dụng web, ĐATN này lựa chọn là mô hình Client-Server, với hai phần tách biệt là Server xử lý dữ liệu sử dụng công nghệ ASP.Net Core và cơ sở dữ liệu SQL Server. Phần giao diện sử dụng VueJS là một framework javascript xây dựng giao diện.

Để đáp ứng nhu cầu liên hệ giữa người tìm nhà trọ và người cho thuê trọ trong ứng dụng, công nghệ websocket được sử dụng và được triển khai cụ thể bằng thư viện SignalR của ASP.Net Core.

Ngoài ra, ứng dụng này sử dụng các API được cung cấp bởi Google Map nhằm hỗ trợ người dùng trong việc tìm kiếm địa chỉ trên bản đồ, cung cấp thông tin về một vị trí cụ thể và các địa điểm lân cận.

### **1.4 Bố cục đề án**

Phần còn lại của báo cáo đề án tốt nghiệp gồm 5 chương:

Chương 2, tiến hành phân tích sâu về các yêu cầu chức năng đối với từng tác nhân tham gia nghiệp vụ bằng các usecase và nêu ra các yêu cầu phi chức năng như: hiệu năng, độ tin cậy, tính dễ dùng, tính dễ bảo trì, hoặc các yêu cầu về mặt kỹ thuật như về CSDL, công nghệ sử dụng ...

Chương 3, sau khi phân tích các yêu cầu chức năng cho hệ thống, tiến hành lựa chọn công nghệ phù hợp và giới thiệu, mô tả chúng. Cụ thể, ĐATN xây dựng website theo mô hình

Client-Server. Phía front-end sử dụng framework VueJS để xây dựng giao diện. Phía back-end, sử dụng framework ASP.NET core tích hợp Entity Framework và sử dụng database SQL Server. Và còn một số công nghệ khác sẽ trình bày chi tiết hơn trong phần này.

Chương 4, đây là chương sẽ tiến hành thiết kế hệ thống dựa trên những phân tích nghiệp vụ ở chương 2 và phù hợp triển khai với những công nghệ lựa chọn ở chương 3. Tiến hành thiết kế kiến trúc phần mềm, thiết kế chi tiết giao diện, lớp, cơ sở dữ liệu sau đó xây dựng ứng dụng, thể hiện kết quả đạt được và minh họa các chức năng chính.

Chương 5, trình bày các giải pháp, những khó khăn và hướng giải pháp, cũng như những đóng góp nổi bật của đồ án.

Chương 6, đây là chương cuối cùng của báo cáo đồ án tốt nghiệp. Đưa ra kết luận về sản phẩm đã thực hiện, và hướng phát triển dựa trên tiêu chí trải nghiệm người dùng.

## Chương 2 Khảo sát và phân tích yêu cầu

### 2.1 Khảo sát hiện trạng

Như đã phân tích ở chương 1, hiện nay vẫn rất nhiều người lựa chọn cách tìm thông tin thuê nhà và chia sẻ thông tin thuê nhà bằng những cách truyền thống như dán tờ rơi quảng cáo, sử dụng facebook đăng bài, hay tìm kiếm qua môi giới.

Tuy nhiên, rất nhiều trang web cho thuê nhà cũng đã được ra đời trong thời gian gần đây. Ở đây chỉ đề cập đến hai website lớn nhất và nổi tiếng nhất đó là: muonnha.com.vn<sup>5</sup> và phongtro123.com<sup>6</sup>. Bảng 1 so sánh các điểm chính của hai website này bao gồm các thông tin như ứng dụng có hỗ trợ trên nhiều nền tảng khác nhau không (máy tính, điện thoại, tab...), thêm bài đăng miễn phí hay mất phí, ứng dụng có hỗ trợ liên hệ giữa các người dùng, ứng dụng có hiển thị các tiện ích lân cận khu vực nhà trọ hay không và có đảm bảo tính năng chính cho phép thêm bài đăng về các thông tin nhà trọ hay không.

Bảng 1 So sánh hai website cho thuê nhà

	muonnha.com.vn	phongtro123.com
Ứng dụng hỗ trợ trên nhiều nền tảng	Không	Không
Chi phí khi đăng bài	Không	Không
Hỗ trợ liên hệ giữa các người dùng ngay trên trang	Không	Không
Hiển thị các tiện ích lân cận khu vực nhà trọ	Không	Không

<sup>5</sup> <http://muonnha.com.vn/> truy cập lần cuối 29/07/2022

<sup>6</sup> <https://phongtro123.com> truy cập lần cuối 29/07/2022

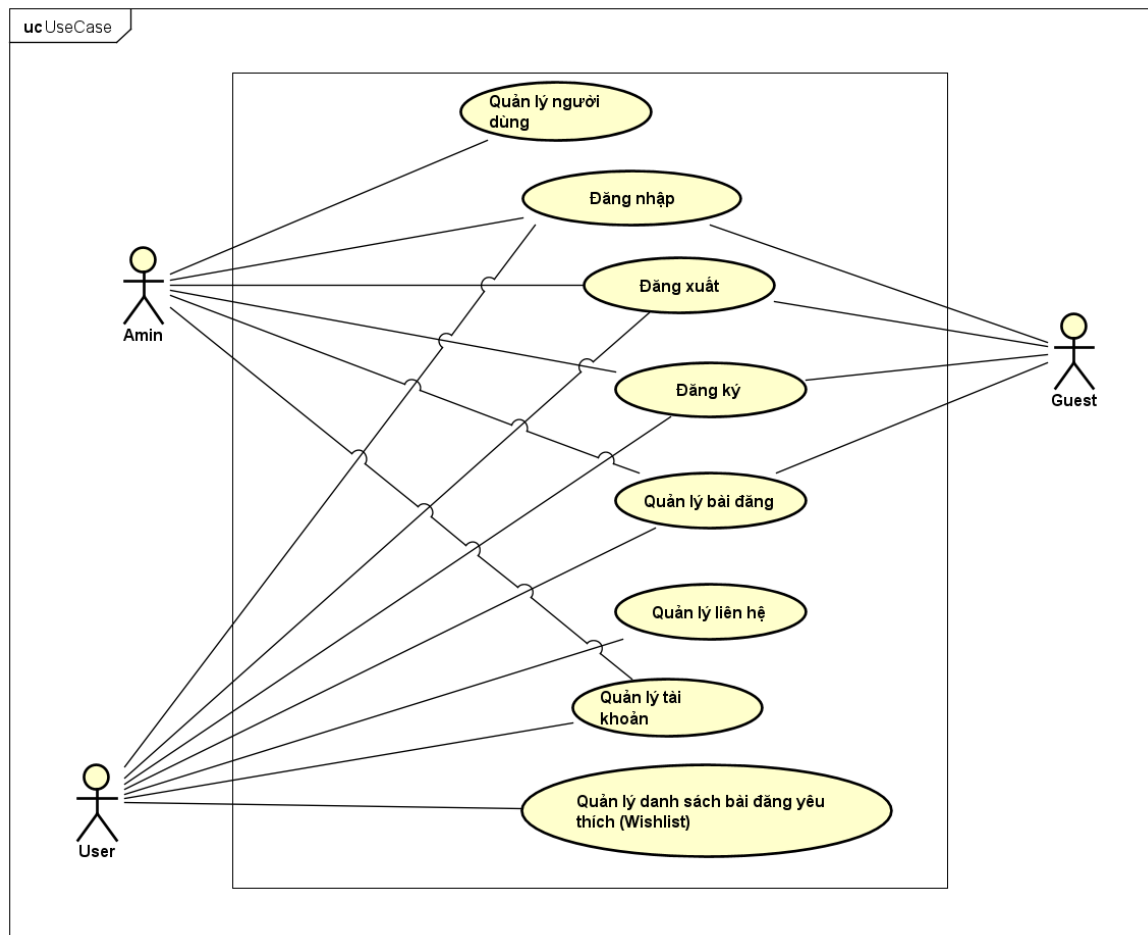
Cho phép thêm bài đăng về thông tin thuê nhà	Có	Có
--	----	----

Qua những điểm trên và trên cơ sở tìm hiểu hai trang web, ĐATN này sẽ kế thừa các ưu điểm của các ứng dụng đồng thời cũng khắc phục các nhược điểm để cho ra được kết quả là một ứng dụng tối ưu nhất.

## 2.2 Tổng quan chức năng

### 2.2.1 Biểu đồ usecase tổng quan

Hệ thống có 3 tác nhân chính: Guest (khách là người dùng của hệ thống khi chưa đăng nhập), User (là người dùng đã đăng nhập), Admin (người dùng đã đăng nhập với vai trò là admin)



Hình 1: Usecase tổng quan

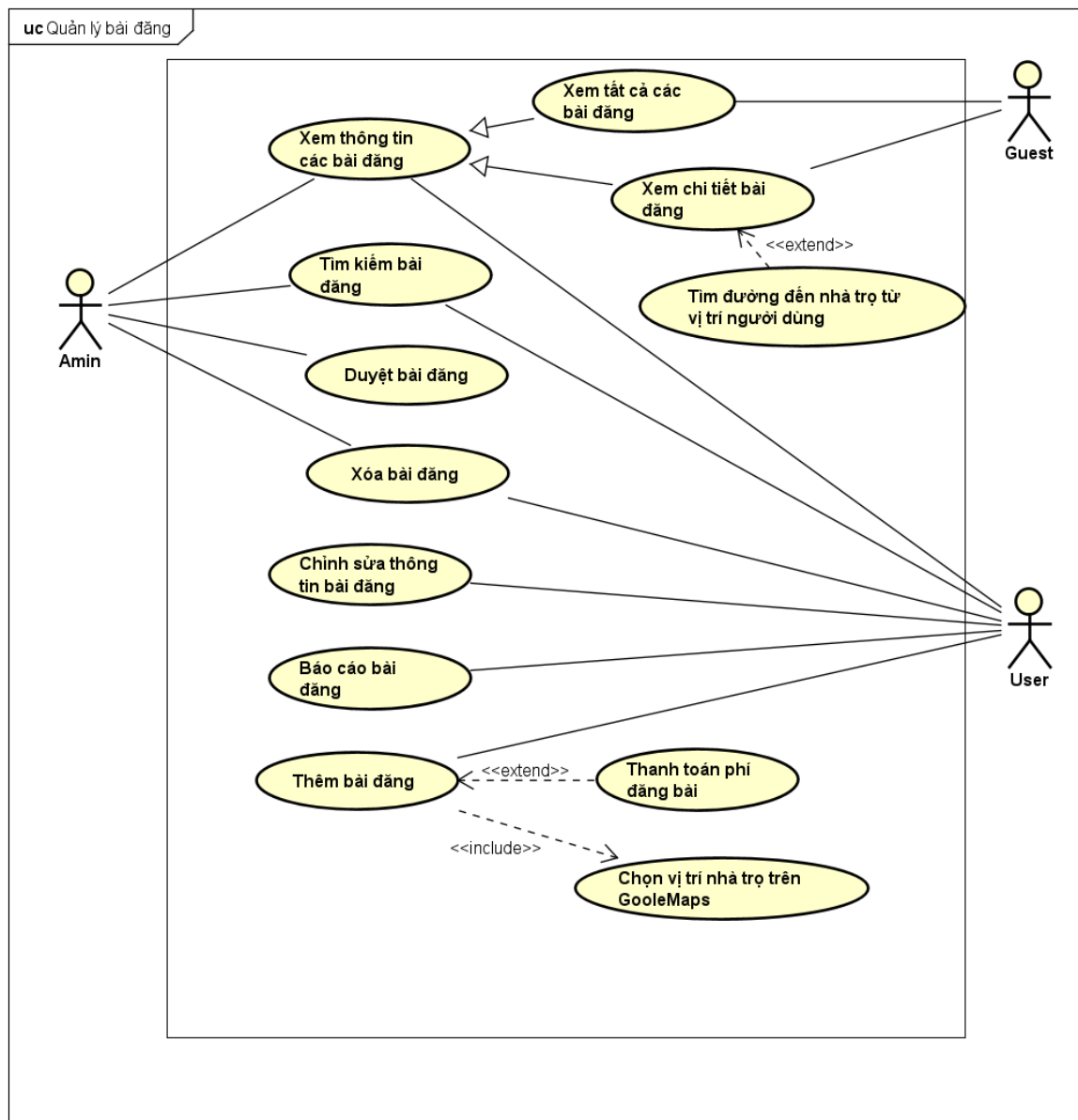
Mô tả thêm về các usecase:

Bảng 2: Mô tả thêm về các Usecase

Usecase	Mô tả
Quản lý bài đăng	Bao gồm các tính năng: duyệt bài đăng, xóa bài đăng, chỉnh sửa thông tin bài đăng, báo cáo bài đăng, thêm bài đăng, xem bài đăng...
Quản lý danh sách bài đăng yêu thích	Bao gồm các tính năng: Thêm bài đăng vào danh sách yêu thích, xóa bài khỏi danh sách yêu thích.
Quản lý liên hệ	Bao gồm các tính năng: Liên hệ với chủ sở hữu, xem danh sách tin nhắn của người dùng.
Quản lý người dùng	Bao gồm các tính năng: Xóa người dùng, Chặn người dùng đăng bài.
Quản lý tài khoản	Bao gồm các tính năng: Thay đổi mật khẩu tài khoản, thay đổi avatar người dùng.
Thanh toán phí đăng bài	Bao gồm các tính năng: Thanh toán qua Momo, thanh toán qua Zalo.
Tìm kiếm bài đăng	Bao gồm các tính năng: Tìm kiếm bài đăng theo tiêu đề, tìm kiếm bài đăng theo tên người đăng, tìm kiếm bài đăng theo tên đăng nhập người dùng.

Sau đây là phân phân rã các usecase đã phân tích ở trên.

## 2.2.2 Biểu đồ usecase phân rã quản lý bài đăng



Hình 2: Phân rã usecase quản lý bài đăng

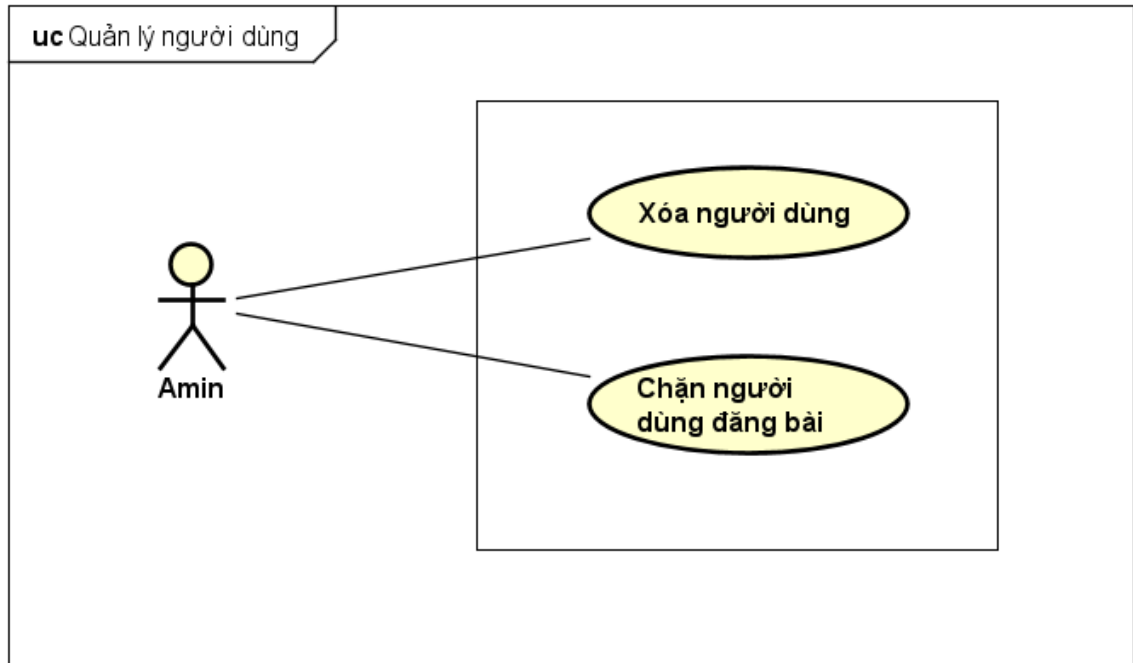
Admin có thể xem bài đăng, duyệt bài đăng, xem các bài đăng đã có trên hệ thống và xóa bài đăng.

User có thể xóa các bài đăng, chỉnh sửa các thông tin các bài đăng mà mình đã đăng, báo cáo bài đăng, thêm bài đăng và xem các bài đăng trên hệ thống. Với usecase thêm bài đăng, địa chỉ nhà trọ là thông tin bắt buộc nhập nên luôn phải bao gồm việc chọn vị trí nhà trọ trên GoogleMaps.



Guest chỉ có thể xem các bài đăng và xem chi tiết một bài đăng cụ thể. Với usecase xem chi tiết một bài đăng, người dùng có thể tìm đường đến địa chỉ nhà trọ từ vị trí người dùng dựa trên GoogleMaps.

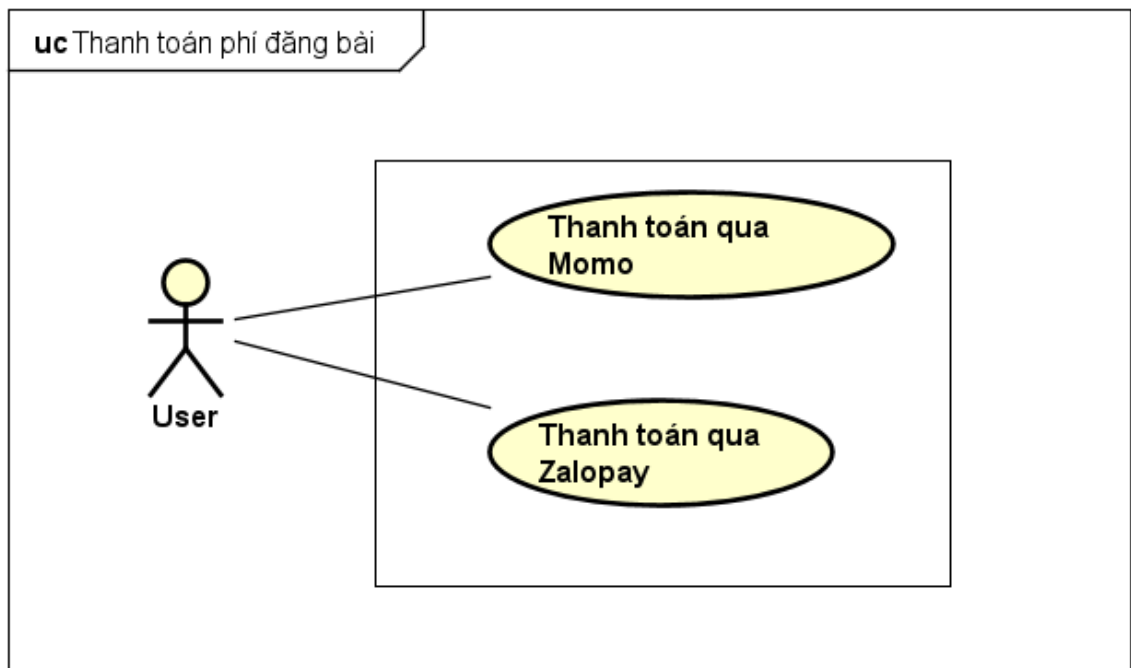
### 2.2.3 Biểu đồ usecase phân rã quản lý người dùng



Hình 3: Phân rã use case quản lý người dùng

Admin có thể xóa người dùng hoặc chặn người dùng. Khi người dùng bị chặn sẽ không thể thêm bài đăng được.

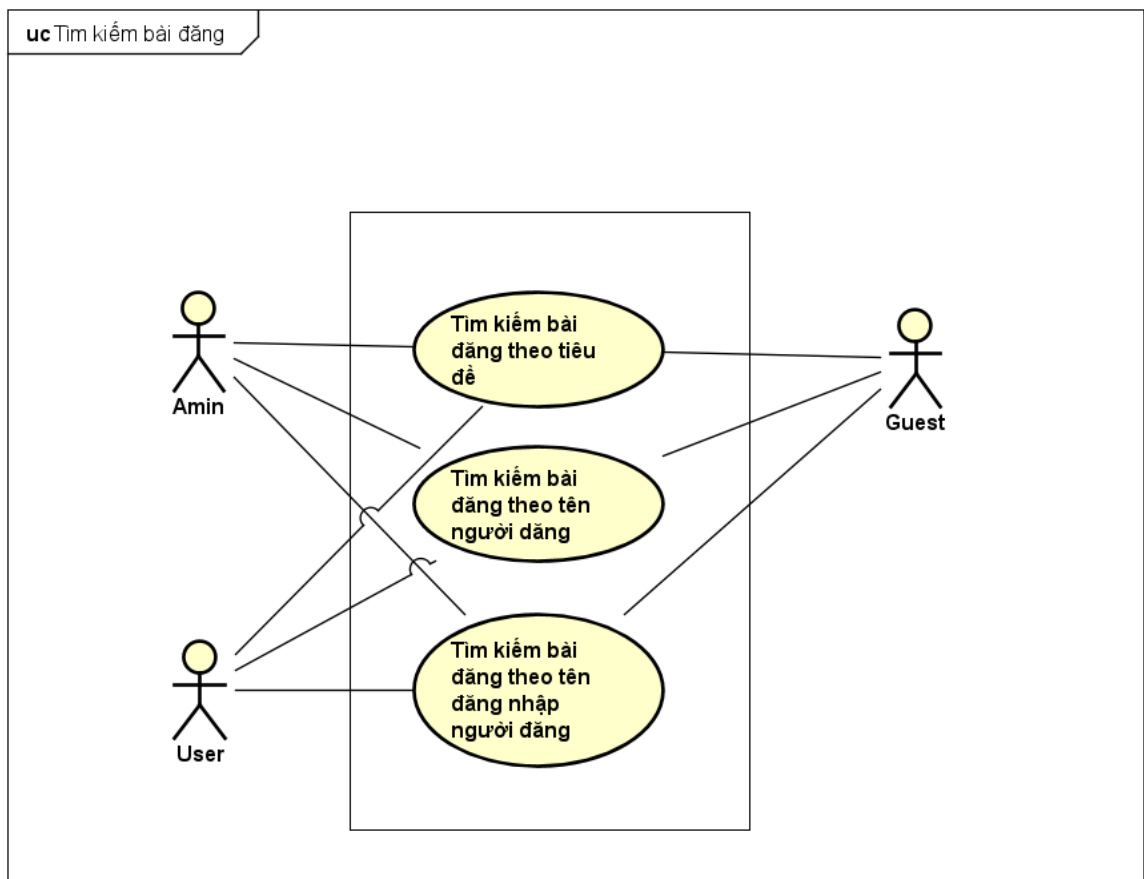
#### 2.2.4 Biểu đồ usecase phân rã thanh toán phí bài đăng



Hình 4: Phân rã use case thanh toán phí bài đăng

Sau khi thực hiện thêm bài đăng, bài đăng sẽ ở trạng thái chưa thanh toán, muốn bài đăng được hiển thị với người dùng, User phải thực hiện thanh toán phí đăng bài qua Momo hoặc Zalopay.

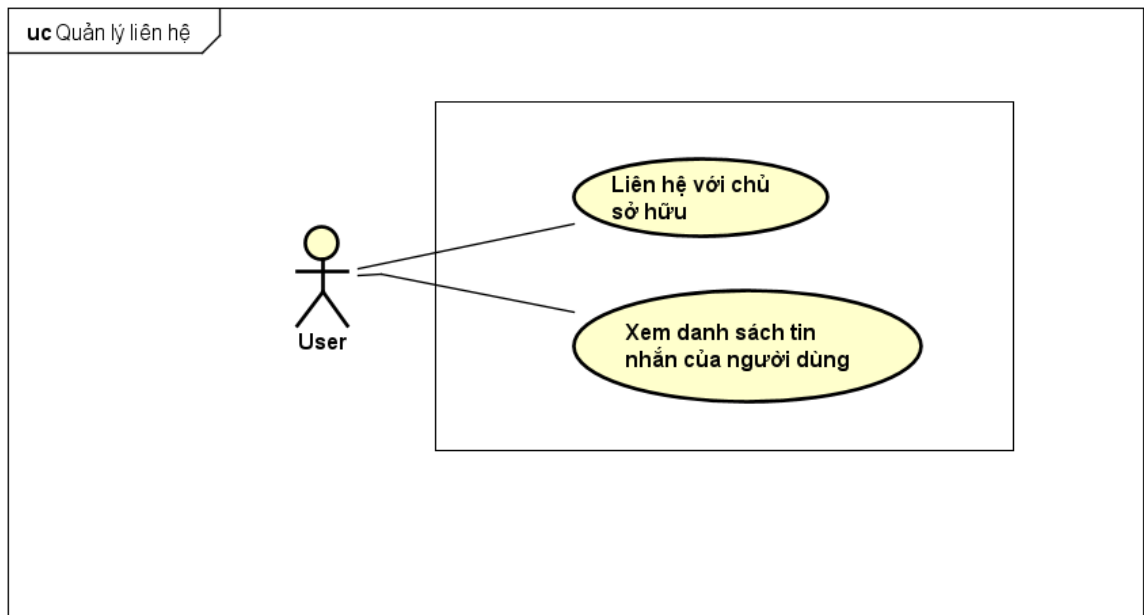
### 2.2.5 Biểu đồ usecase phân rã tìm kiếm bài đăng



Hình 5: Phân rã use case tìm kiếm bài đăng

Cả ba tác nhân chính của hệ thống đều có thể tìm kiếm bài đăng theo tiêu đề bài đăng, tìm kiếm bài đăng theo tên người đăng hoặc theo tên đăng nhập của người đăng.

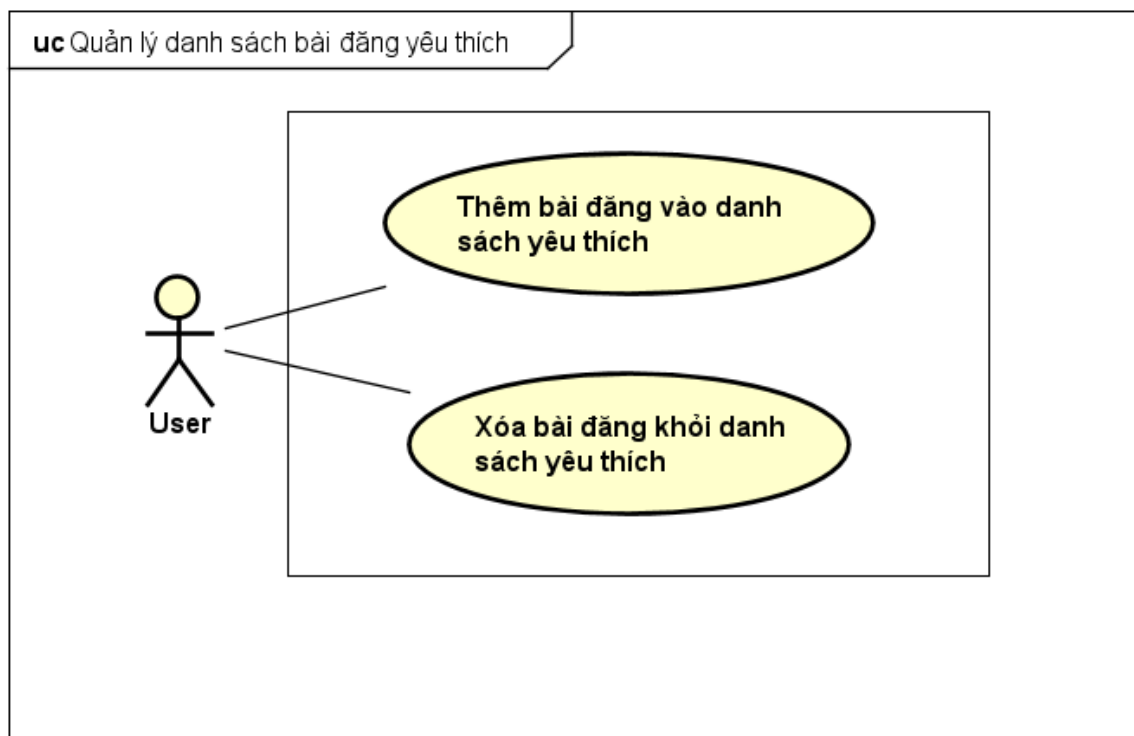
### 2.2.6 Biểu đồ usecase phân rã Quản lý liên hệ



Hình 6: Phân rã use case quản lý liên hệ

User có thể liên hệ với chủ sở hữu bài đăng, xem danh sách tin nhắn của mình.

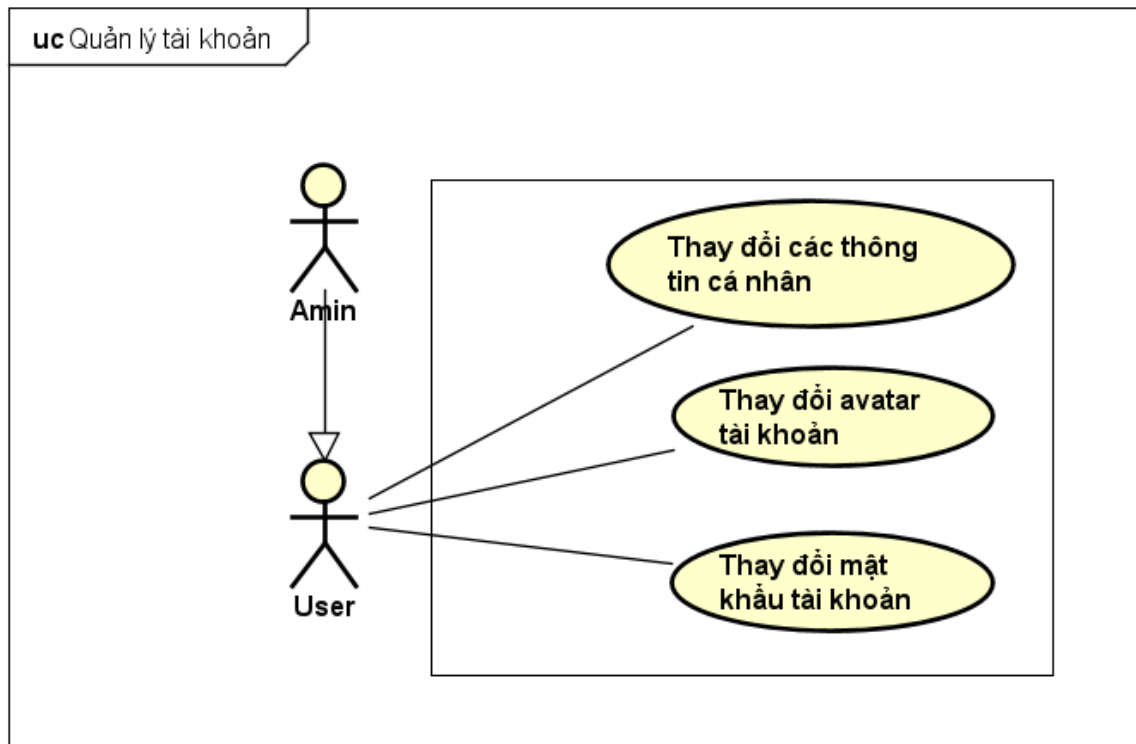
### 2.2.7 Biểu đồ usecase phân rã Quản lý danh sách bài đăng yêu thích (Wishlist)



Hình 7: Phân rã use case quản lý danh sách bài đăng yêu thích (Wishlist)

User có thể lưu bài đăng vào danh sách yêu thích và xóa các bài đăng đã lưu khỏi danh sách yêu thích.

### 2.2.8 Biểu đồ usecase phân rã Quản lý tài khoản



Hình 8: Phân rã use case quản lý tài khoản

Trong usecase quản lý tài khoản, Admin đóng vai trò cũng giống như một User, họ có thể thay đổi các thông tin về tài khoản, thay đổi avatar hoặc thay đổi mật khẩu tài khoản.

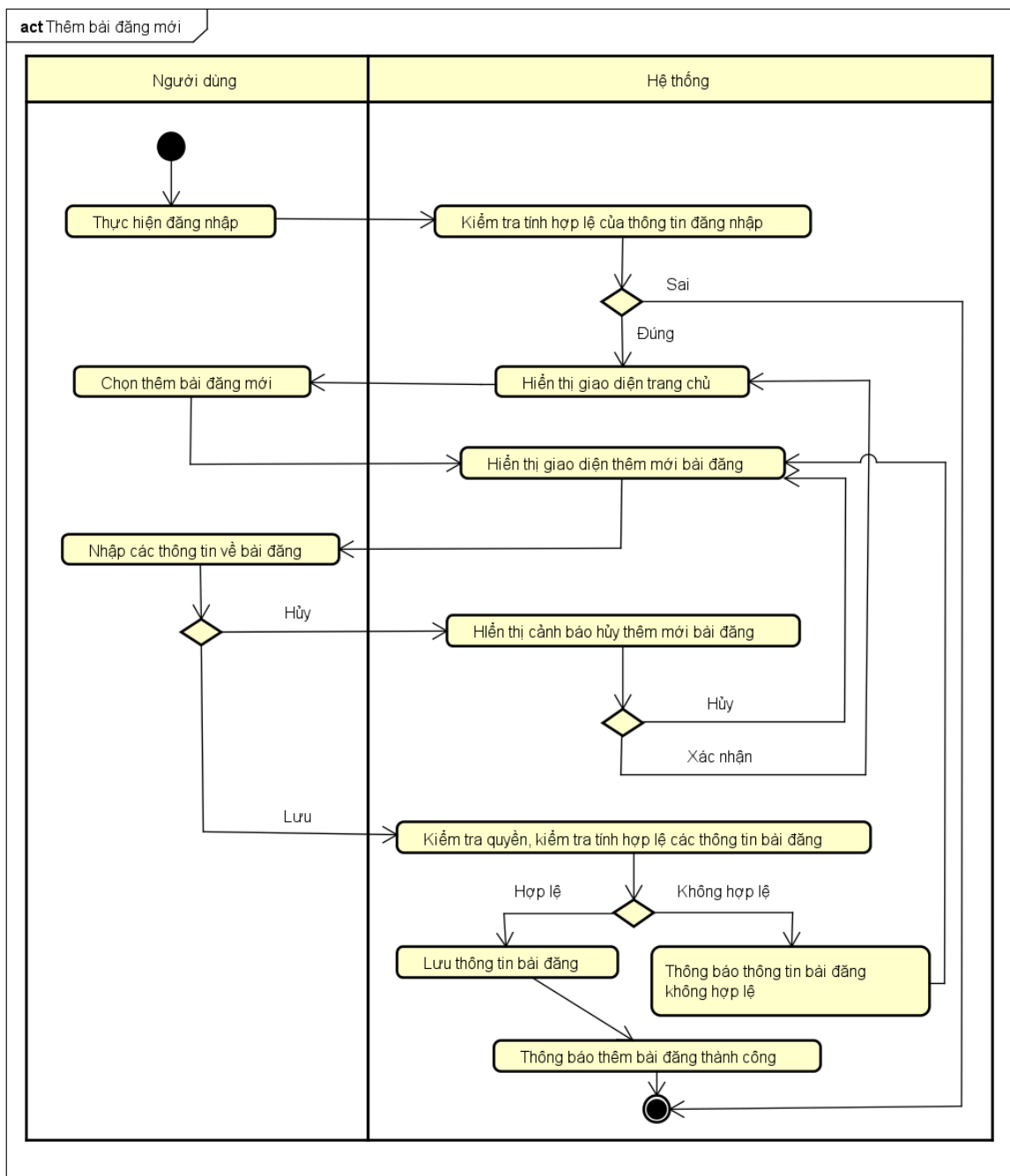
### 2.2.9 Quy trình nghiệp vụ

#### 2.2.9.1 Quy trình thêm bài đăng mới

Bảng 3: Quy trình thêm bài đăng mới

	Thêm bài đăng mới
<b>Mục đích quy trình</b>	Người dùng thêm một bài đăng mới
<b>Phạm vi quy trình</b>	Người dùng sở hữu bài đăng
<b>Đầu vào quy trình</b>	<ul style="list-style-type: none"><li>- Người dùng đã đăng nhập</li><li>- Các thông tin về bài đăng: Tiêu đề bài đăng, mô tả chi tiết, các thông tin về nhà trọ trong bài đăng (ảnh nhà, số tầng, số phòng ngủ...)</li></ul>

<b>Flow Quy trình</b>	<ul style="list-style-type: none"> <li>- Người dùng đăng nhập</li> <li>- Hệ thống hiển thị giao diện trang chủ</li> <li>- Người dùng yêu cầu thêm bài đăng mới</li> <li>- Hệ thống hiển thị giao diện thêm mới bài đăng</li> <li>- Người dùng nhập các thông tin về bài đăng</li> <li>- Người dùng Lưu thông tin bài đăng</li> <li>- Hệ thống kiểm tra quyền, kiểm tra tính hợp lệ các thông tin bài đăng</li> <li>- Hệ thống lưu lại thông tin và thực hiện thông báo trên giao diện người dùng</li> </ul>
<b>Đầu ra Quy trình</b>	<ul style="list-style-type: none"> <li>- Thông báo thêm mới bài đăng thành công cho người dùng</li> </ul>
<b>Quy trình đặc biệt</b>	Không



Hình 9: Quy trình thêm bài đăng mới

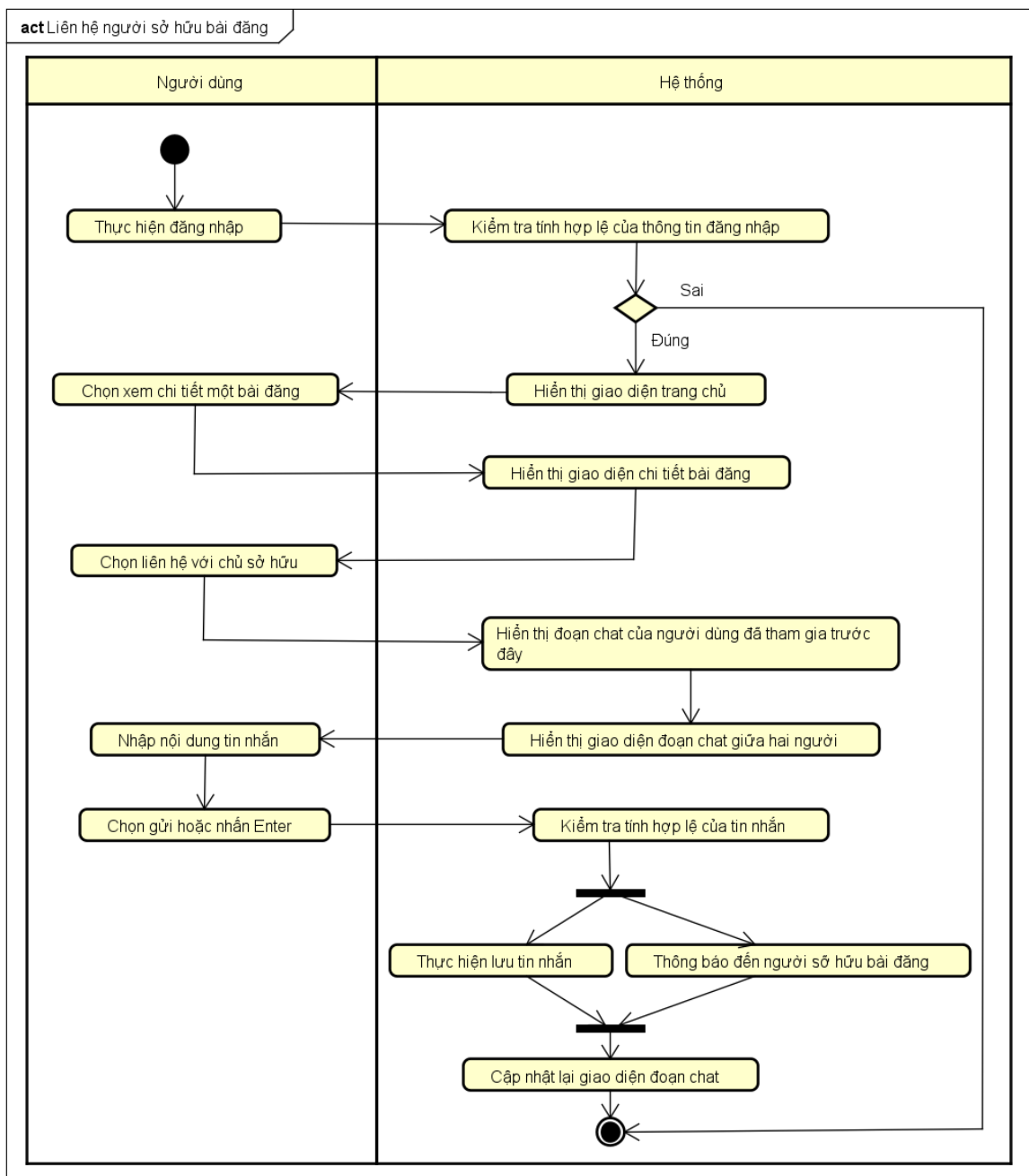
### 2.2.9.2 Quy trình liên hệ với người sở hữu bài đăng

Bảng 4: Quy trình liên hệ với người sở hữu bài đăng

	Liên hệ với người sở hữu bài đăng
<b>Mục đích quy trình</b>	Người dùng sau khi xem bài đăng muốn liên hệ với người sở hữu bài đăng



<b>Phạm vi quy trình</b>	Người dùng sở hữu bài đăng, người dùng xem bài đăng
<b>Đầu vào quy trình</b>	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập</li> <li>- Nội dung tin nhắn muốn gửi đến người sở hữu bài đăng</li> </ul>
<b>Flow Quy trình</b>	<ul style="list-style-type: none"> <li>- Người dùng đăng nhập</li> <li>- Hệ thống hiển thị giao diện trang chủ</li> <li>- Người dùng chọn xem chi tiết một bài đăng</li> <li>- Người dùng chọn liên hệ với chủ sở hữu</li> <li>- Hệ thống hiển thị giao diện đoạn chat giữa hai người và các đoạn chat của người dùng đã tham gia trước đây</li> <li>- Người dùng nhập vào nội dung tin nhắn</li> <li>- Người dùng ấn nút gửi hoặc ấn enter trên bàn phím</li> <li>- Hệ thống kiểm tra tính hợp lệ của tin nhắn</li> <li>- Hệ thống thực hiện lưu tin nhắn và gửi thông báo cho người sở hữu bài đăng</li> <li>- Hệ thống cập nhật lại giao diện đoạn chat</li> </ul>
<b>Đầu ra Quy trình</b>	<ul style="list-style-type: none"> <li>- Tin nhắn được gửi cho người sở hữu bài đăng thành công và cập nhật lên giao diện đoạn chat</li> <li>- Có thông báo tin nhắn mới cho người sở hữu bài đăng</li> </ul>
<b>Quy trình đặc biệt</b>	Không



Hình 10: Quy trình liên hệ với người sở hữu bài đăng

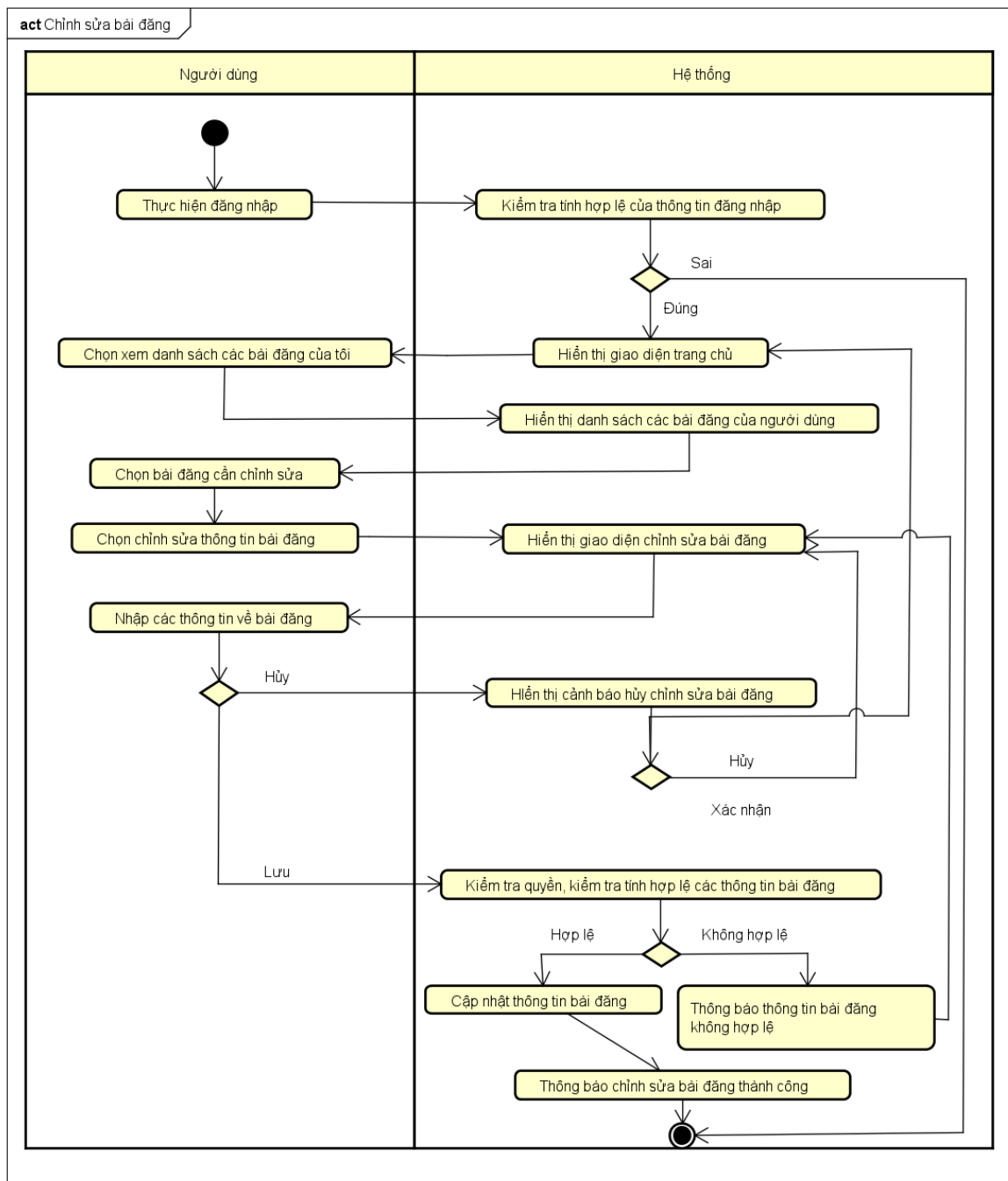
### 2.2.9.3 Quy trình chỉnh sửa bài đăng

- Có hai cách để chỉnh sửa bài đăng: Trong giao diện các bài đã đăng, người dùng có thể lựa chọn chỉnh sửa bài đăng hoặc trong giao diện xem chi tiết bài đăng, người dùng ấn nút chọn chỉnh sửa bài đăng. Ở đây em mô tả nghiệp vụ theo cách số 1

Bảng 5: Quy trình sửa bài đăng

	Chỉnh sửa bài đăng
--	--------------------

<b>Mục đích quy trình</b>	Người dùng chỉnh sửa thông tin bài đăng của mình
<b>Phạm vi quy trình</b>	Người dùng sở hữu bài đăng
<b>Đầu vào quy trình</b>	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập</li> <li>- Các thông tin về bài đăng: Tiêu đề bài đăng, mô tả chi tiết, các thông tin về nhà trọ trong bài đăng (ảnh nhà, số tầng, số phòng ngủ...)</li> </ul>
<b>Flow Quy trình</b>	<ul style="list-style-type: none"> <li>- Người dùng đăng nhập</li> <li>- Người dùng chọn Xem danh sách các bài đăng của tôi</li> <li>- Hệ thống hiển thị danh sách các bài đăng của người dùng</li> <li>- Người dùng chọn bài đăng cần chỉnh sửa và chọn chỉnh sửa thông tin bài đăng</li> <li>- Hệ thống hiển thị giao diện chỉnh sửa chi tiết bài đăng</li> <li>- Người dùng nhập các thông tin về bài đăng</li> <li>- Người dùng Lưu thông tin bài đăng</li> <li>- Hệ thống kiểm tra quyền, kiểm tra tính hợp lệ các thông tin bài đăng</li> <li>- Hệ thống lưu lại thông tin và thực hiện thông báo trên giao diện người dùng</li> </ul>
<b>Đầu ra Quy trình</b>	<ul style="list-style-type: none"> <li>- Thông báo chỉnh sửa bài đăng thành công cho người dùng</li> </ul>
<b>Quy trình đặc biệt</b>	Không



Hình 11: Quy trình chỉnh sửa bài đăng

## 2.3 Đặc tả chức năng

### 2.3.1 Đặc tả usecase Liên hệ với người đăng bài

Bảng 6: Đặc tả Usecase Liên hệ với người đăng bài

Mã Usecase	UC01	Tên Usecase	Liên hệ với người đăng bài
Tác nhân hệ thống	User, Guest		

Tiền điều kiện	Không		
Luồng sự kiện chính	STT	Thực hiện	Hành động
	1.1	User, Guest	Trên trang chủ lựa chọn xem chi tiết một bài đăng
	1.2	Hệ thống	Lấy thông tin chi tiết theo bài đăng đã được chọn
	1.3	Hệ thống	Hiển thị chi tiết bài đăng
	2.1	User, Guest	Ấn vào nút “Liên hệ với chủ sở hữu”
	2.2	Hệ thống	Kiểm tra người dùng đã thực hiện đăng nhập hay chưa
	2.3	Hệ thống	Thực hiện chuyển hướng người dùng đến giao diện chat
	2.4	Hệ thống	Lấy thông tin về cuộc hội thoại giữa người dùng và chủ sở hữu
	2.5	Hệ thống	Lấy thông tin về danh sách các cuộc hội thoại của người dùng
	2.6	Hệ thống	Hiển thị giao diện cuộc hội thoại và danh sách các cuộc hội thoại của người dùng
	3.1	User	Nhập tin nhắn và ấn nút gửi hoặc ấn enter
	3.2	Hệ thống	Kiểm tra tin nhắn của người dùng
	3.3	Hệ thống	Cập nhật cuộc hội thoại và cập nhật lại giao diện cho người dùng
Luồng sự kiện thay thế	STT	Thực hiện	Hành động
	2.3a	Hệ thống	Hiển thị thông báo chức năng cần đăng nhập và cho người dùng xác nhận
	2.4a	User	Chọn xác nhận
	2.5a	Hệ thống	Thực hiện chuyển hướng sang giao diện đăng nhập
	2.4b	User	Chọn hủy bỏ
	2.5b	Hệ thống	Quay lại giao diện chi tiết bài đăng
	3.2a	Hệ thống	Nếu tin nhắn rỗng thì không thực hiện gì tiếp theo
Hậu điều kiện	Không		

### 2.3.2 Đặc tả usecase Chỉnh sửa bài đăng

Bảng 7: Đặc tả usecase chỉnh sửa bài đăng

Mã Usecase	UC02	Tên Usecase	Chỉnh sửa bài đăng
Tác nhân hệ thống	User		
Tiền điều kiện	Đăng nhập thành công với quyền là User		
Luồng sự kiện chính	STT	Thực hiện	Hành động
	1.1	User	Trên trang chủ lựa chọn xem chi tiết một bài đăng
	1.2	Hệ thống	Lấy thông tin chi tiết theo bài đăng đã được chọn
	1.3	Hệ thống	Kiểm tra xem người đăng bài có phải người dùng đang đăng nhập
	1.4	Hệ thống	Hiển thị nút chỉnh sửa bài đăng
	2.1	User	Ấn vào nút chỉnh sửa bài đăng
	2.2	Hệ thống	Hiển thị giao diện chỉnh sửa bài đăng
	3.1	User	Nhập các thông tin chỉnh sửa bài đăng
	4.1	User	Ấn nút Lưu
	4.2	Hệ thống	Kiểm tra tính hợp lệ các thông tin bài đăng
	4.3	Hệ thống	Hiển thị thông báo chỉnh sửa bài đăng thành công và load lại giao diện chi tiết bài đăng
Luồng sự kiện thay thế	STT	Thực hiện	Hành động
	4.1a	User	Ấn nút Hủy
	4.2a	Hệ thống	Thực hiện chuyển hướng về giao diện chi tiết bài đăng
	4.3a	Hệ thống	Hiển thị các cảnh báo nhập liệu không chính xác
Hậu điều kiện	Không		

### 2.3.3 Đặc tả usecase Thêm bài đăng

Bảng 8: Đặc tả usecase Thêm bài đăng

Mã Usecase	UC03	Tên Usecase	Thêm bài đăng
Tác nhân hệ thống	User		
Tiền điều kiện	Đăng nhập thành công với quyền là User		
Luồng sự kiện chính	STT	Thực hiện	Hành động
	1.1	User	Trên trang chủ lựa “Thêm” trên thanh header
	2.1	User	Lựa chọn “Thêm bài đăng mới”
	2.2	Hệ thống	Thực hiện chuyển hướng đến giao diện thêm bài đăng mới
	3.1	User	Nhập các thông tin về bài đăng
	4.1	User	Ấn nút Lưu
	4.2	Hệ thống	Kiểm tra tính hợp lệ các thông tin bài đăng
	4.3	Hệ thống	Hiển thị thông báo thêm bài đăng thành công và chuyển hướng đến giao diện chi tiết bài đăng
Luồng sự kiện thay thế	STT	Thực hiện	Hành động
	4.1a	User	Ấn nút Hủy
	4.2a	Hệ thống	Thực hiện chuyển hướng về giao diện trang chủ
	4.2b	Hệ thống	Hiển thị các cảnh báo nhập liệu không chính xác
Hậu điều kiện	Không		

## 2.4 Yêu cầu phi chức năng

### 2.4.1 Yêu cầu chung

Ứng dụng có cơ chế phân quyền quản trị viên theo từng chức năng, vai trò đảm bảo từng tài khoản đăng nhập chỉ được phép sử dụng các tính năng theo quyền hạn của mình. Ngoài ra website hỗ trợ quản trị viên các phương thức tìm kiếm và truy vấn nhanh dữ liệu. Về cấu trúc, việc xây dựng ứng dụng theo từng module cho phép mở rộng dễ dàng. Ứng dụng có tính khả chuyển khi cho phép quản trị sử dụng trên nhiều trình duyệt khác nhau như Google Chrome, Microsoft Edge, Cốc Cốc v.v.

### **2.4.2 Yêu cầu về giao diện người dùng**

Ứng dụng luôn đặt yếu tố dễ dùng và có tính tương tác cao, đơn giản hóa các thao tác và gợi ý cho người dùng về các thành phần chức năng trên giao diện bằng cách sử dụng các icon, button v.v nhằm tăng khả năng tương tác với website.

### **2.4.3 Yêu cầu về bảo mật**

Hệ thống đảm bảo được các yêu cầu về bảo mật như (i) dùng phương thức sinh số ngẫu nhiên (number random generator encryption) để mã hóa thông tin password đăng nhập của người dùng và đăng nhập người dùng theo phiên (ii) các API được đảm bảo khỏi các truy cập giả mạo, chưa được xác thực, (iii) xác thực phân quyền cho các request gửi đến server trước khi xử lý.

### **2.4.4 Yêu cầu khác**

Ngoài ra ứng dụng xây dựng đáp ứng các yêu cầu (i) tính khả thi - xây dựng ứng dụng đáp ứng các yêu cầu từ phía người dùng, (ii) tính linh động - ứng dụng dễ dàng thay đổi theo yêu cầu người dùng.



## Chương 3 Công nghệ sử dụng

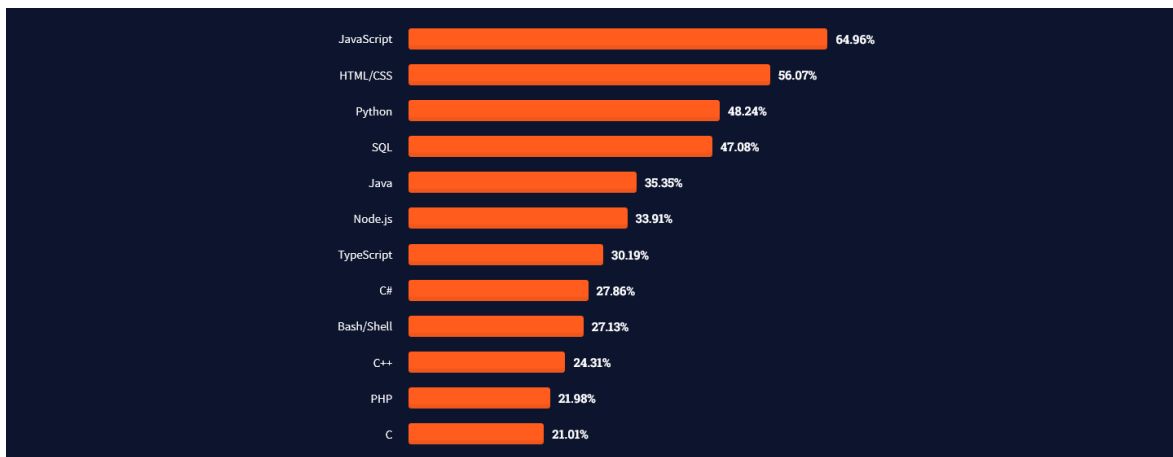
Dựa trên các phân tích trên, để xây dựng trang web đáp ứng đủ các yêu cầu, ĐATN này sử dụng những công nghệ sau: với phần front end xây dựng giao diện, sử dụng VueJS – là một framework javascript tiên tiến trong việc xây dựng giao diện người dùng. Về phía backend, sử dụng công nghệ ASP.NET core – một sản phẩm của Microsoft. ASP.NET core là một framework sử dụng ngôn ngữ C# trong lập trình. Để phù hợp với nghiệp vụ thuê nhà và các dữ liệu liên quan đến thuê nhà, ĐATN sử dụng hệ quản trị cơ sở dữ liệu SQL Server.

### 3.1 Lập trình front-end

#### 3.1.1 Javascript

JavaScript, theo phiên bản hiện hành, là một ngôn ngữ lập trình thông dịch được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web (phía người dùng) cũng như phía máy chủ (với Nodejs). Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript.

Javascript hiện nay là ngôn ngữ lập trình phổ biến nhất thế giới. Hình 12 là thống kê của StackOverFlow về số lượng người dùng các ngôn ngữ lập trình, trong đó Javascript được sử dụng nhiều nhất chiếm 64,96%.



Hình 12: Thống kê các ngôn ngữ lập trình phổ biến nhất thế giới năm 2022

Javascript sinh ra với mục đích chạy trên môi trường run-time của các trình duyệt, tuy nhiên hiện nay với sự ra đời của Nodejs, javascript còn có thể chạy trên các ứng dụng máy chủ. Các ưu điểm đáng chú ý của Javascript:

- Có tốc độ xử lý nhanh nhờ cơ chế xử lý bất đồng bộ (non-blocking). Bạn có thể dễ dàng xử lý hàng ngàn kết nối trong khoảng thời gian ngắn nhất.
- Dễ dàng mở rộng khi có nhu cầu phát triển website.
- Nhận và xử lý nhiều kết nối chỉ với một single-thread. Nhờ đó, hệ thống xử lý sẽ sử dụng ít lượng RAM nhất và giúp quá trình xử lý Nodejs lý nhanh hơn rất nhiều.
- Có khả năng xử lý nhiều Request/s cùng một lúc trong thời gian ngắn nhất.
- Có khả năng xử lý hàng ngàn Process cho hiệu suất đạt mức tối ưu nhất.

### 3.1.2 VueJS

Vue.js là một framework linh động (nguyên bản tiếng Anh: progressive – tiệm tiến) dùng để xây dựng giao diện người dùng (user interfaces). Khác với các framework nguyên khối (monolithic), Vue được thiết kế từ đầu theo hướng cho phép và khuyến khích việc phát triển ứng dụng theo từng bước. Khi phát triển lớp giao diện (view layer), người dùng chỉ cần dùng thư viện lõi (core library) của Vue, dễ tích hợp với các thư viện hoặc dự án có sẵn. Cùng lúc đó, nếu kết hợp với những kỹ thuật hiện đại như SFC (single file components) và các thư viện hỗ trợ, 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 hơn nhiều. Các ưu điểm của VueJS:

- Trang web nhẹ, tốc độ xử lý cực nhanh bởi được Render, xử lý bằng Javascript.
- Đơn giản, dễ học sử dụng.
- Có khả năng xử lý các render thuộc server thành file JS tĩnh.
- Kho thư viện lớn, hỗ trợ xây dựng giao diện một cách nhanh chóng, hoàn hảo.

- Dung lượng tải thấp, giúp tốc độ tải trang nhanh hơn.
- Sàng lọc, tích hợp các tính năng ưu việt của nhiều framework đối thủ, giúp tối ưu hóa hiệu suất làm việc.

## 3.2 Lập trình back-end

### 3.2.1 ASP.NET Core

Bản phát hành đầu tiên của ASP.NET đã xuất hiện cách đây 15 năm trước, nó là một phần của .NET Framework. Từ đó, hàng triệu lập trình viên đã sử dụng nó để xây dựng những ứng dụng web tuyệt vời, và trên những năm đó Microsoft đã phát triển thêm nhiều tính năng mới.

ASP.NET Core có một số thay đổi kiến trúc lớn, đó là kết quả của việc học hỏi rất nhiều từ các framework module hóa khác. ASP.NET Core không còn dựa trên System.Web.dll nữa. Nó được dựa trên một tập hợp các gói, các module hay cũng được gọi là các Nuget packages.

ASP.NET Core đạt được rất nhiều nền tảng cải tiến: Hợp nhất việc xây dựng web UI và web APIs. ASP.Net Core hỗ trợ việc xây dựng cả APIs và web UI trong một dự án, với những cải tiến mới tích hợp những client-side frameworks hiện đại và những luồng phát triển mới, người phát triển sẽ tích hợp được những frameworks UI mới nhất vào dự án ASP.Net Core một cách dễ dàng. Việc triển khai ASP.Net Core cũng đạt được nhiều cải tiến đáng kể, có thể host trên IIS hoặc self-host trong các process, hệ thống cũng có thể cấu hình dựa trên môi trường đám mây thật sự. ASP.Net core cũng rất tiện dụng cho các lập trình viên với kỹ thuật dependency injection được xây dựng sẵn, việc khai báo cũng như sử dụng rất nhanh và tiện dụng. Trong DATN này, ứng dụng cũng áp dụng sử dụng dependence injection và sẽ được trình bày cụ thể trong chương 5. Ngoài ra, ASP.Net Core cũng hỗ trợ xây dựng và chạy trên đa nền tảng (Windows, Mac và Linux), với việc chuyển các thực thể, thành phần, module thành các thành phần như những NuGet packages, việc quản lý mã nguồn và các packages trên các nền tảng đều trở nên đơn giản hơn. Đặc biệt, ASP.Net Core là một mã nguồn mở, đang dần tập trung hơn vào cộng đồng.

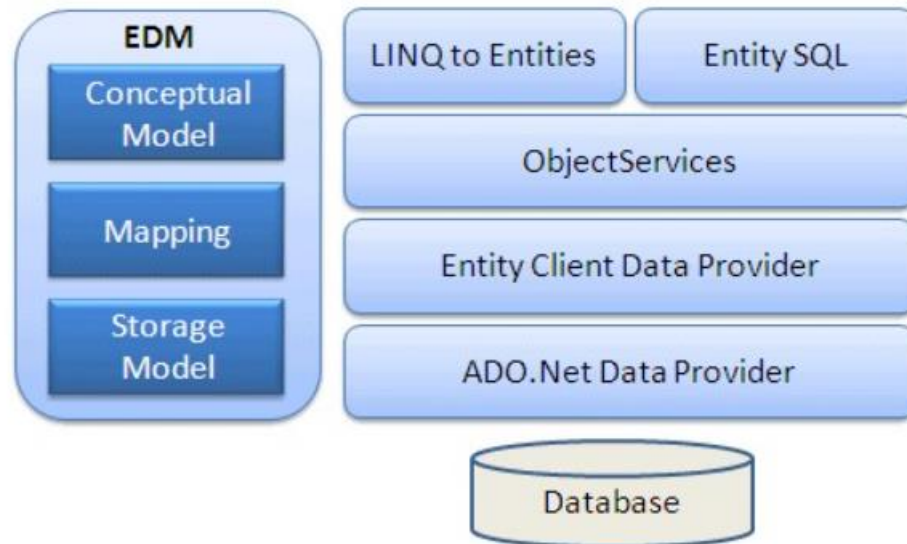
### 3.2.2 Entity Framework

Entity Framework ra đời nhằm hỗ trợ sự tương tác giữa các ứng dụng trên nền tảng .NET với các cơ sở dữ liệu quan hệ. Hay, Entity Framework chính là công cụ giúp ánh xạ giữa các đối tượng trong ứng dụng, phần mềm của bạn với các bảng của một cơ sở dữ liệu quan hệ.

Entity Framework giúp các nhà phát triển Web tương tác với dữ liệu quan hệ theo phương pháp hướng đối tượng với ít mã hơn so với các ứng dụng truyền thống. Lợi ích

lớn nhất của nó là giúp lập trình viên giảm thiểu việc lập trình mã nguồn để thực hiện truy cập và tương tác với cơ sở dữ liệu.

Cấu trúc của Entity Framework:



Hình 13: Cấu trúc của Entity Framework

- EDM (Entity Data Model): EDM bao gồm 3 phần:
  - Conceptual Model: Chứa các model class và các quan hệ của nó. Nó độc lập với thiết kế bảng CSDL của bạn.
  - Mapping: gồm có thông tin về cách Conceptual model kết nối với Storage model.
  - Storage Model: Đây là database design model bao gồm các bảng, views, stored procedures, và những quan hệ và các khóa.
- LINQ to Entities: Là ngôn ngữ truy vấn sử dụng để viết các truy vấn tới object model. Trả về các thực thể được định nghĩa bên trong Conceptual model.
- Entity SQL: Đây là một ngôn ngữ truy vấn khác, nó giống LINQ to Entities.
- Object Service: Object service sẽ có trách nhiệm trong việc cụ thể hóa quá trình chuyển đổi dữ liệu trả về từ một entity client data provider tới một entity object structure.
- Entity Client Data Provider: Nhiệm vụ chính của nó là chuyển đổi L2E hoặc những truy vấn Entity SQL vào một truy vấn SQL. Nó được hiểu bởi CSDL cơ bản. Giao tiếp với ADO.Net data provider lần lượt gửi và nhận dữ liệu từ CSDL.
- ADO.Net Data Provider: Thực hiện giao tiếp với CSDL bằng sử dụng chuẩn ADO.Net.

Những tính năng đáng chú ý nhất của Entity framework:

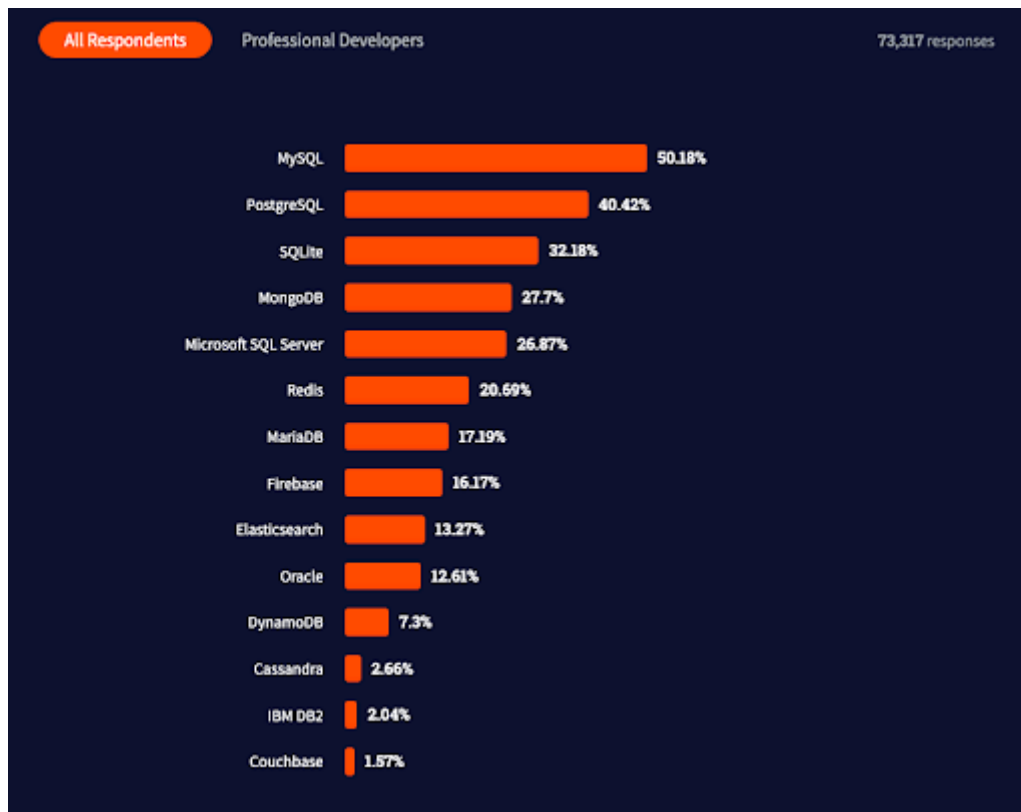
- Entity framework hỗ trợ stored procedure.
- Nó làm việc với bất kỳ cơ sở dữ liệu quan hệ nào có Entity Framework hợp lệ.
- Entity Framework tạo truy vấn SQL từ LINQ to Entities.
- Nó được phát triển như một sản phẩm mã nguồn mở. Entity framework là một sản phẩm của Microsoft.
- EF cho phép tạo những câu lệnh thêm, xóa, cập nhật.
- Theo dõi những thay đổi của các đối tượng trong bộ nhớ.

Sử dụng Entity Framework sẽ đem lại những lợi ích sau:

- Hỗ trợ việc thực hiện truy vấn dữ liệu. Linq to Entities hỗ trợ trực tiếp việc thao tác với objects được sinh ra từ Entity Framework, đồng nghĩa với việc có thể sử dụng nó để thao tác với dữ liệu mà không cần viết code SQL.
- Việc update các classes, commands dễ dàng mỗi khi cơ sở dữ liệu có sự thay đổi, điều này giúp bạn tiết kiệm thời gian đáng kể.
- Entity Framework sẽ tự động tạo ra các classes, commands tương ứng cho việc select, insert, update, delete dữ liệu từ cơ sở dữ liệu quan hệ.
- Entity Framework tự động tạo ra các classes cho việc truy suất cơ sở dữ liệu giúp lập trình viên giảm được thời gian viết code thao tác với database. Hỗ trợ bạn không phải mất quá nhiều thời gian cho việc viết code để thao tác với database.

### 3.2.3 Microsoft Sql Server

Microsoft SQL Server là một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) hỗ trợ nhiều ứng dụng xử lý giao dịch, nghiệp vụ thông minh và phân tích trong môi trường doanh nghiệp CNTT. Microsoft SQL Server là một trong năm hệ quản trị cơ sở dữ liệu được sử dụng nhiều nhất năm 2022. Hình ... thống kê các hệ quản trị dữ liệu được sử dụng nhiều nhất năm 2022 theo số liệu của LearnSQL.com, trong đó Microsoft SQL Server đứng ở vị trí thứ 5 thể hiện vẫn là một trong những hệ quản trị cơ sở dữ liệu được sử dụng nhiều nhất trên thế giới hiện tại.



Hình 14: Thống kê các hệ quản trị cơ sở dữ liệu được sử dụng nhiều nhất năm 2022 theo LearnSQL.com

Giống như các hệ quản trị cơ sở dữ liệu quan hệ khác, Microsoft SQL Server được xây dựng dựa trên SQL, một ngôn ngữ lập trình tiêu chuẩn hóa sử dụng để quản lý cơ sở dữ liệu và truy vấn dữ liệu mà chúng chứa. SQL Server được liên kết với Transact-SQL (T-SQL), một triển khai SQL của Microsoft bổ sung một tập hợp các phần mở rộng lập trình độc quyền cho ngôn ngữ chuẩn.

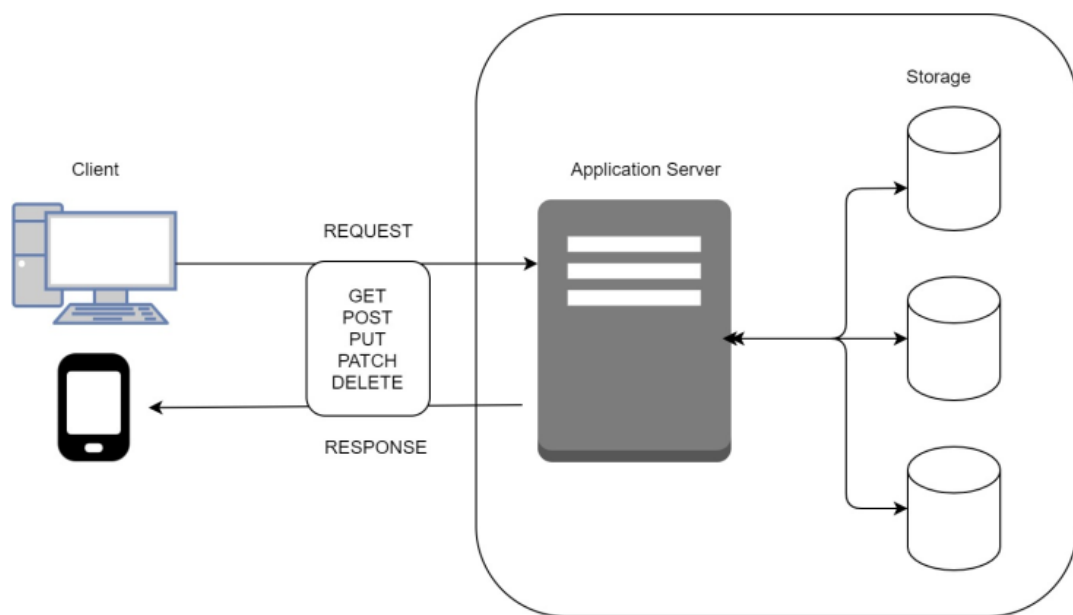
Cũng giống như các hệ quản trị cơ sở dữ liệu quan hệ khác, SQL Server được xây dựng dựa trên cấu trúc dựa vào các dòng dữ liệu, liên kết các thành phần dữ liệu liên quan đến nhau trong các bảng dữ liệu với các bảng khác. Trong DATN này, các dữ liệu đều có thể tổ chức phù hợp dưới dạng bảng, vì vậy SQL Server sẽ là hệ quản trị cơ sở dữ liệu phù hợp với yêu cầu đặt ra của đề án.

# Chương 4 Phát triển và triển khai ứng dụng

## 4.1 Thiết kế kiến trúc

### 4.1.1 Lựa chọn kiến trúc phần mềm

#### 4.1.1.1 Kiến trúc client-server



Hình 15: Kiến trúc Client-Server

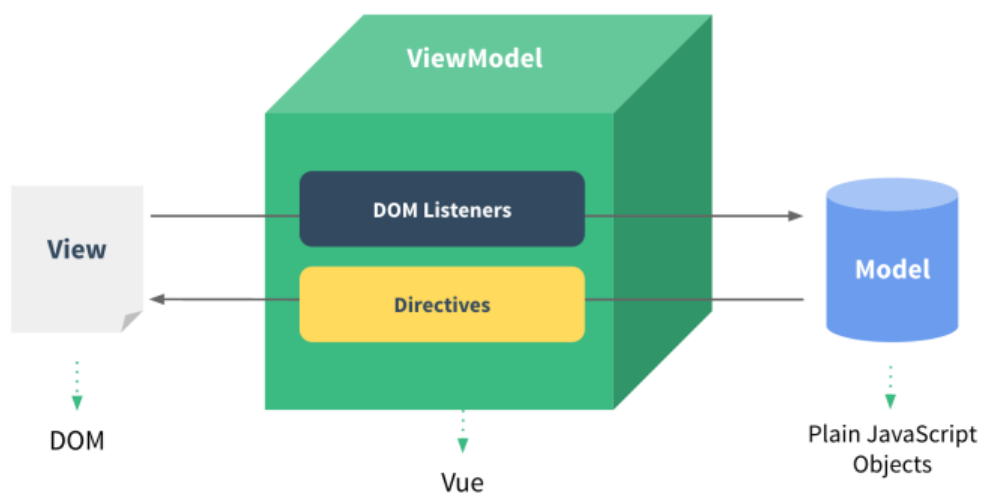
Hệ thống ứng dụng sẽ bao gồm 2 phần chính: Phần giao diện người dùng (Front-end hay Web UI) và phần Lưu trữ, xử lý Logic (Back-end), giao tiếp với nhau qua API (Application Programming Interface) với chuẩn RESTful.

Các quy tắc của một hệ thống chuẩn RESTful API: Sử dụng giao diện thống nhất (UI). Tài nguyên phải được nhận dạng duy nhất thông qua một URL và chỉ bằng cách sử dụng các phương thức cơ bản của giao thức mạng, chẳng hạn như DELETE, PUT, POST và GET với HTTP, nếu có thể thao tác tài nguyên. Hệ thống hoạt động theo mô hình client-server, trong đó server là tập hợp các service nhỏ lắng nghe các request từ client. Với từng request khác nhau thì có thể một hoặc nhiều service xử lý. Server và client không lưu trạng thái của nhau, mỗi request lên server thì client phải đóng gói thông tin đầy đủ để server hiểu

được. Hệ thống phát triển theo hướng này có ưu điểm nhưng cũng tồn tại những khuyết điểm điển hình là gia tăng lượng thông tin cần truyền tải giữa client và server. Các response có thể lấy ra từ cache (khả năng caching). Bằng cách cache các response, server giảm tải việc xử lý request, còn client cũng nhận được thông tin nhanh hơn. Trong hệ thống REST chia tách các thành phần hệ thống theo từng lớp, mỗi lớp chỉ sử dụng lớp ở dưới nó và giao tiếp với lớp ở ngay trên nó mà thôi. Điều này giúp giảm độ phức tạp của hệ thống, giúp các thành phần tách biệt nhau từ đó dễ dàng mở rộng từng thành phần.

Với nhu cầu đáp ứng đa nền tảng trong tương lai, ứng dụng được xây dựng theo kiến trúc client-server với chuẩn RESTful API đảm bảo tách biệt hai phần client và server với client chịu trách nhiệm cung cấp giao diện cho người dùng, server thực hiện xử lý dữ liệu được lưu trữ. Với việc tách biệt như vậy, khi có nhu cầu phát triển ứng dụng trên một nền tảng khác như mobile hay tablet, việc phát triển sẽ chỉ cần tập trung vào thay đổi giao diện từ web thành mobile.

#### 4.1.1.2 Kiến trúc MVVM (Model-View-Viewmodel)



Hình 16: Kiến trúc MVVM (Model-View-Viewmodel)

**View:** Tương tự như trong mô hình MVC, View là phần giao diện của ứng dụng để hiển thị dữ liệu và nhận tương tác của người dùng. Một điểm khác biệt so với các ứng dụng truyền thống là View trong mô hình này tích cực hơn. Nó có khả năng thực hiện các hành vi và phản hồi lại người dùng thông qua tính năng binding, command.

**Model:** Cũng tương tự như trong mô hình MVC. Model là các đối tượng giúp truy xuất và thao tác trên dữ liệu thực sự.

**ViewModel:** Lớp trung gian giữa View và Model. ViewModel có thể được xem là thành phần thay thế cho Controller trong mô hình MVC. Nó chứa các mã lệnh cần thiết để thực



hiện ràng buộc dữ liệu (data binding). Ưu điểm của mô hình này là ViewModel không hề biết gì về View, một ViewModel có thể được sử dụng cho nhiều View (one-to-many). ViewModel sử dụng Observer design pattern để liên lạc với View (thường được gọi là binding data, có thể là 1 chiều hoặc 2 chiều tùy nhu cầu ứng dụng). Chính đặc điểm này MVVM thường được phối hợp với các thư viện hỗ trợ Reactive Programming hay Event/Data Stream, đây là triết lý lập trình hiện đại và hiệu quả phát triển rất mạnh trong những năm gần đây.

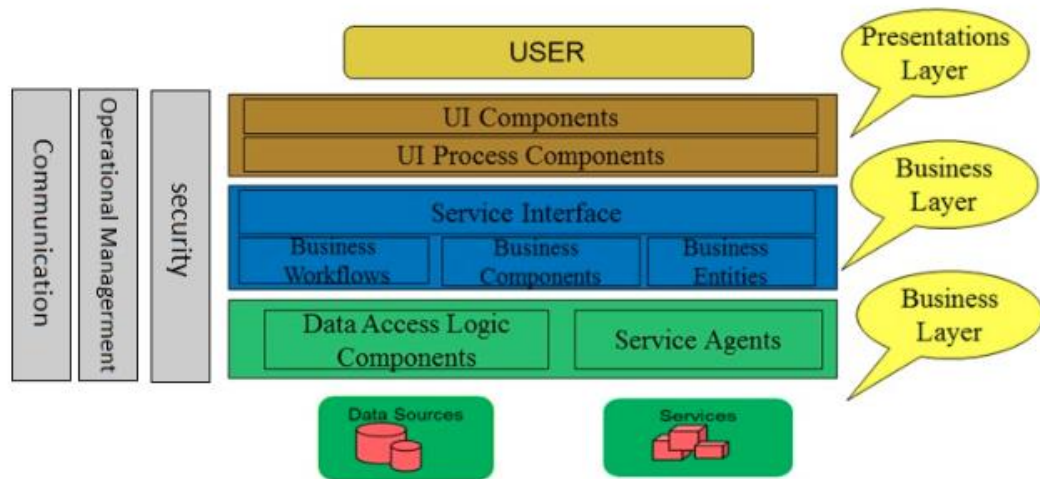
Đây là mô hình được framework VueJS sử dụng trong việc xây dựng giao diện, vì vậy với việc sử dụng framework này, tính tái sử dụng của mã nguồn được nâng cao do việc xây dựng một ViewModel có thể sử dụng lại ở nhiều nơi. Cơ chế của Observer Pattern cũng cho phép VueJS cập nhật lại giao diện ngay khi có thay đổi của các dữ liệu đã được gán vào giao diện (data binding).

#### **4.1.1.3 Kiến trúc 3-layer**

3- layer là một kiến trúc kiểu client/server mà trong đó giao diện người dùng (UI-user interface), các quy tắc xử lý (BR-business rule hay BL-business logic), và việc lưu trữ dữ liệu được phát triển như những module độc lập, và hầu hết là được duy trì trên các nền tảng độc lập, và mô hình 3 tầng (3-layer) được coi là một kiến trúc phần mềm và là một mẫu thiết kế.”

Đây là kiến trúc triển khai ứng dụng ở mức vật lý. Kiến trúc gồm 3 module chính và riêng biệt:

- Tầng Presentation: hiển thị các thành phần giao diện để tương tác với người dùng như tiếp nhận thông tin, thông báo lỗi, ...
- Tầng Business Logic: thực hiện các hành động nghiệp vụ của phần mềm như tính toán, đánh giá tính hợp lệ của thông tin, ... Tầng này còn di chuyển, xử lý thông tin giữa 2 tầng trên dưới.
- Tầng Data: nơi lưu trữ và trích xuất dữ liệu từ các hệ quản trị CSDL hay các file trong hệ thống. Cho phép tầng Business logic thực hiện các truy vấn dữ liệu.



Hình 17: Mô hình 3-layer

Mô hình 3-layer gồm có 3 phần chính:

- Presentation Layer (GUI): Lớp này có nhiệm vụ chính giao tiếp với người dùng. Nó gồm các thành phần giao diện (win form, web form...) và thực hiện các công việc như nhập liệu, hiển thị dữ liệu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp Business Logic Layer (BLL).
- Business Logic Layer (BLL): Layer này phân ra 2 thành nhiệm vụ:
  - Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Access Layer và lưu xuống hệ quản trị CSDL.
  - Đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer.
- Data Access Layer (DAL): Lớp này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu (tìm kiếm, thêm, xóa, sửa...).

Ưu điểm

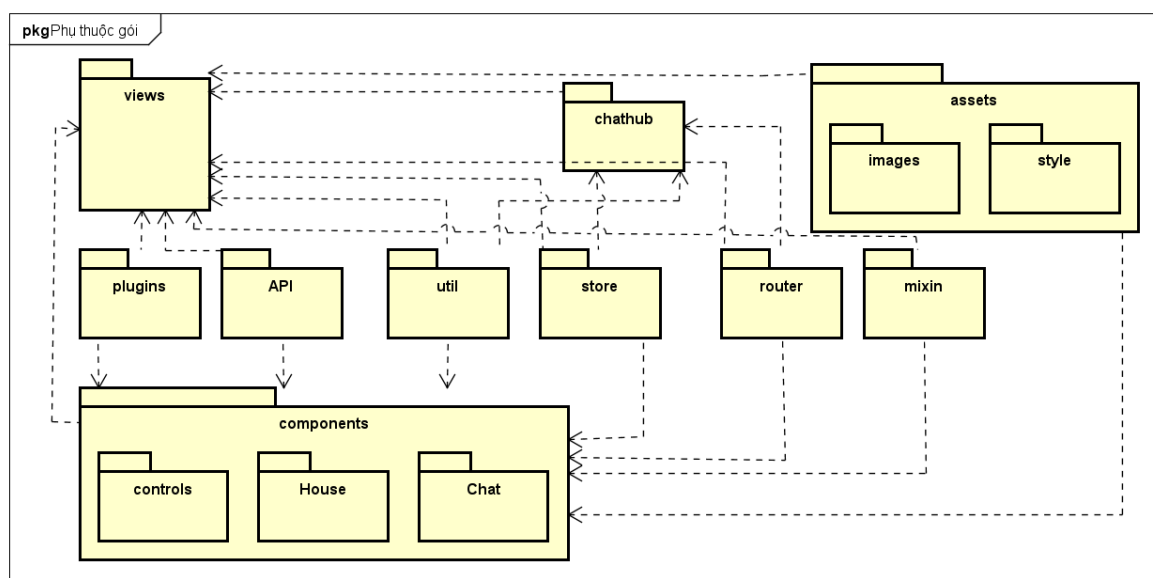
- Việc phân chia thành từng lớp giúp cho code được tường minh hơn. Nhờ vào việc chia ra từng lớp đảm nhận các chức năng khác nhau và riêng biệt như giao diện, xử lý, truy vấn thay vì để tất cả lại một chỗ. Nhằm giảm sự kết dính.
- Dễ bảo trì khi được phân chia, thì một thành phần của hệ thống sẽ dễ thay đổi. Việc thay đổi này có thể được cô lập trong 1 lớp, hoặc ảnh hưởng đến lớp gần nhất mà không ảnh hưởng đến cả chương trình.

- Dễ phát triển, tái sử dụng: khi chúng ta muốn thêm một chức năng nào đó thì việc lập trình theo một mô hình sẽ dễ dàng hơn vì chúng ta đã có chuẩn để tuân theo. Và việc sử dụng lại khi có sự thay đổi giữa hai môi trường (Winform sang Webfrom) thì chỉ việc thay đổi lại lớp GUI.
- Dễ bàn giao. Nếu mọi người đều theo một quy chuẩn đã được định sẵn, thì công việc bàn giao, tương tác với nhau sẽ dễ dàng hơn và tiết kiệm được nhiều thời gian.
- Dễ phân phối khối lượng công việc. Mỗi một nhóm, một bộ phận sẽ nhận một nhiệm vụ trong mô hình 3 lớp. Việc phân chia rõ ràng như thế sẽ giúp các lập trình viên kiểm soát được khối lượng công việc của mình.

Trên đây là các định nghĩa cơ bản giúp hình dung ra được mô hình 3-layer, về việc áp dụng mô hình này sẽ được trình bày rõ hơn ở phần 5.2.

## 4.1.2 Thiết kế tổng quan

### 4.1.2.1 Biểu đồ phụ thuộc gói với ứng dụng bên phía client

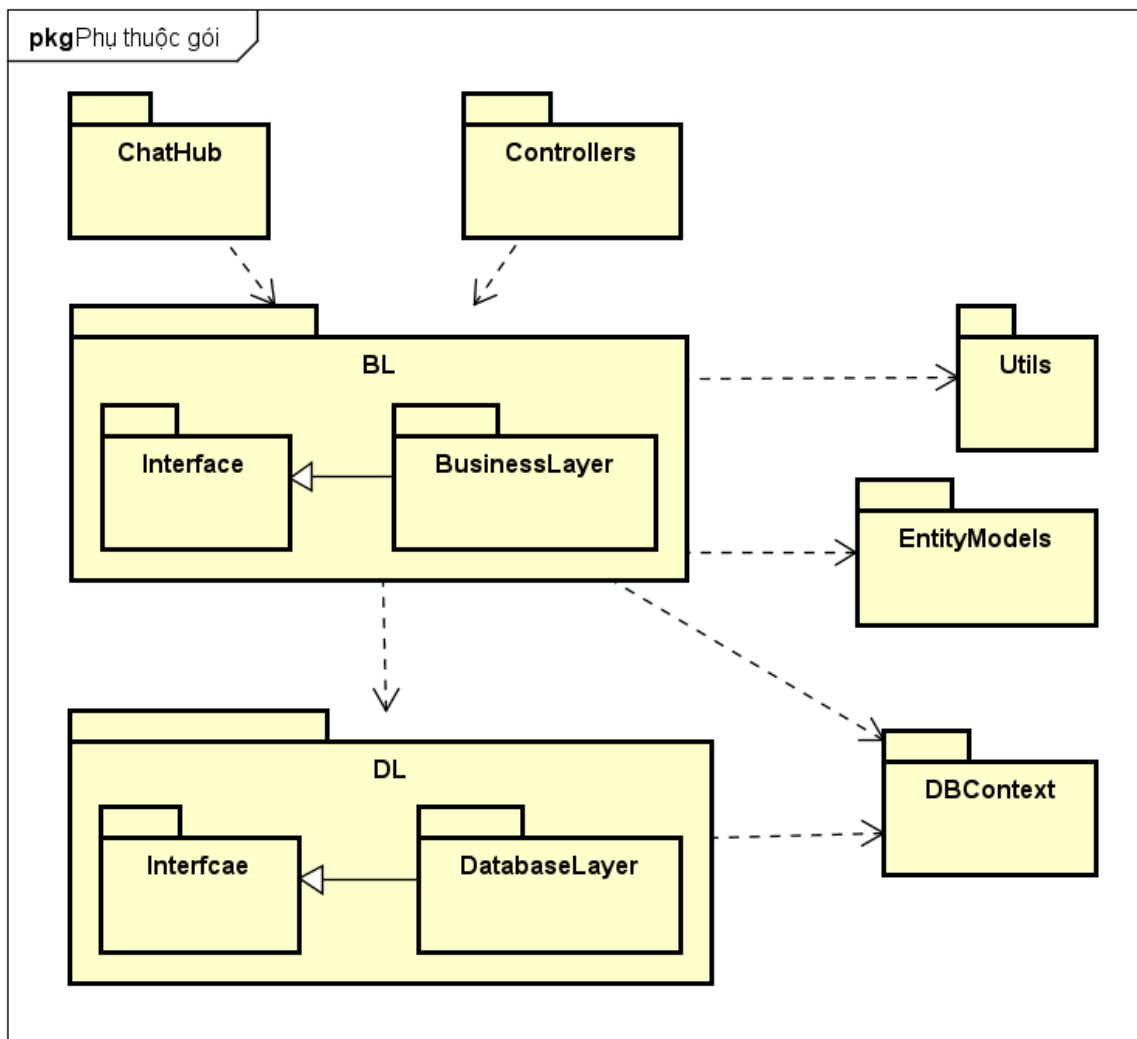


Hình 18: Biểu đồ phụ thuộc gói với ứng dụng bên phía client

- views: chứa các class hiển thị giao diện cho người dùng
- chathub: chứa các class kết nối với server chat
- assets: chứa các nội dung tĩnh như ảnh, các file css, scss dùng chung cho nhiều màn hình
- plugins: chứa các class kết nối với các thư viện ngoài như vuetify, fontawesome...
- API: chứa các class thực hiện các phương thức gọi đến server
- util: bao gồm class util chứa các function sử dụng chung ở nhiều nơi như convert số thành dạng tiền tệ, lưu cookie ...

- store: chứa các class liên quan đến Vuex quản lý các trạng thái của toàn bộ trang
- router: chứa các class thực hiện việc điều hướng trên trang
- mixin: bao gồm class mixin chứa các function chung mà components, views nào cũng kế thừa lại
- components: chứa các class định nghĩa các thành phần được tái sử dụng nhiều lần ở các views bao gồm các package: control – chứa các thành phần điều khiển như các ô input, các form hiển thị ảnh ... , house – chứa các thành phần liên quan đến nhà như hình ảnh nhà ..., chat – chứa các thành phần liên quan đến nội dung chat

#### 4.1.2.2 Biểu đồ phụ thuộc gói với ứng dụng bên phía server



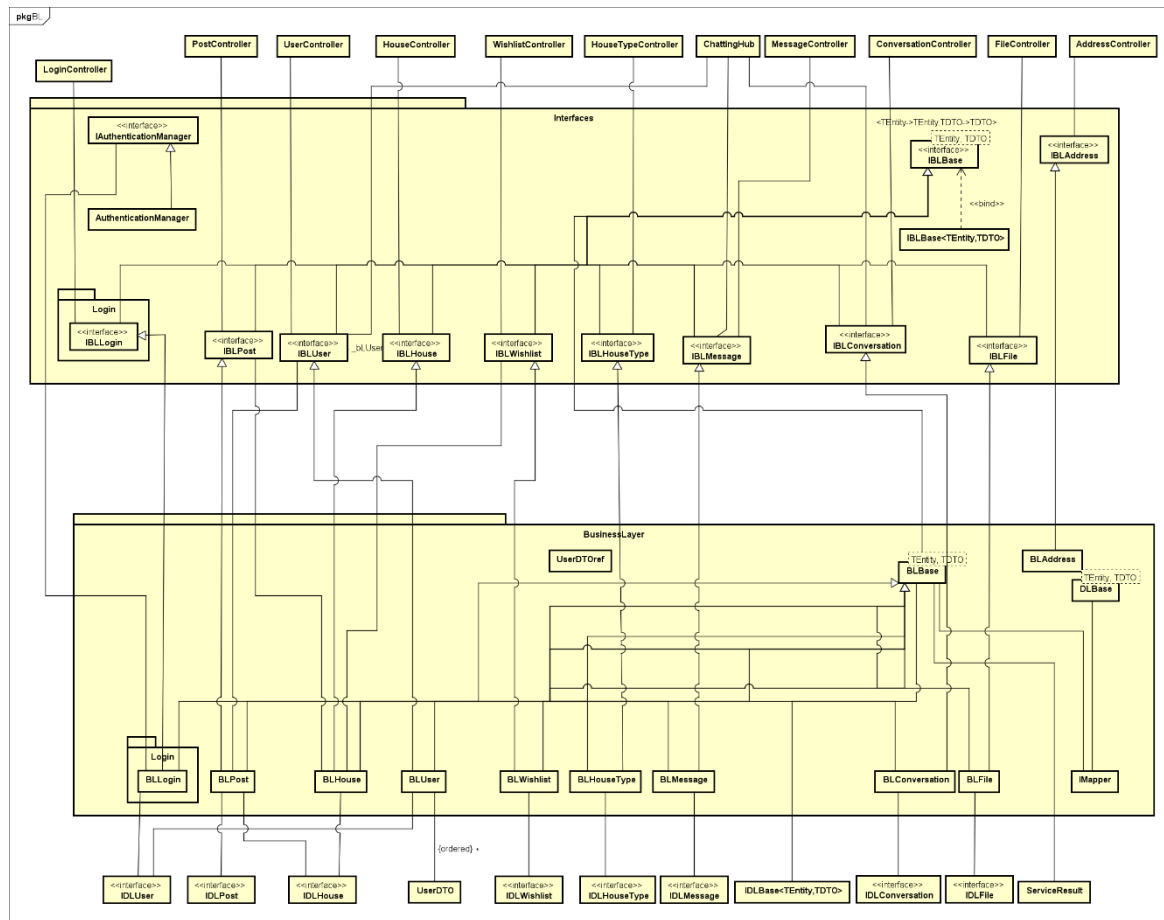
Hình 19: Biểu đồ phụ thuộc gói với ứng dụng bên phía server

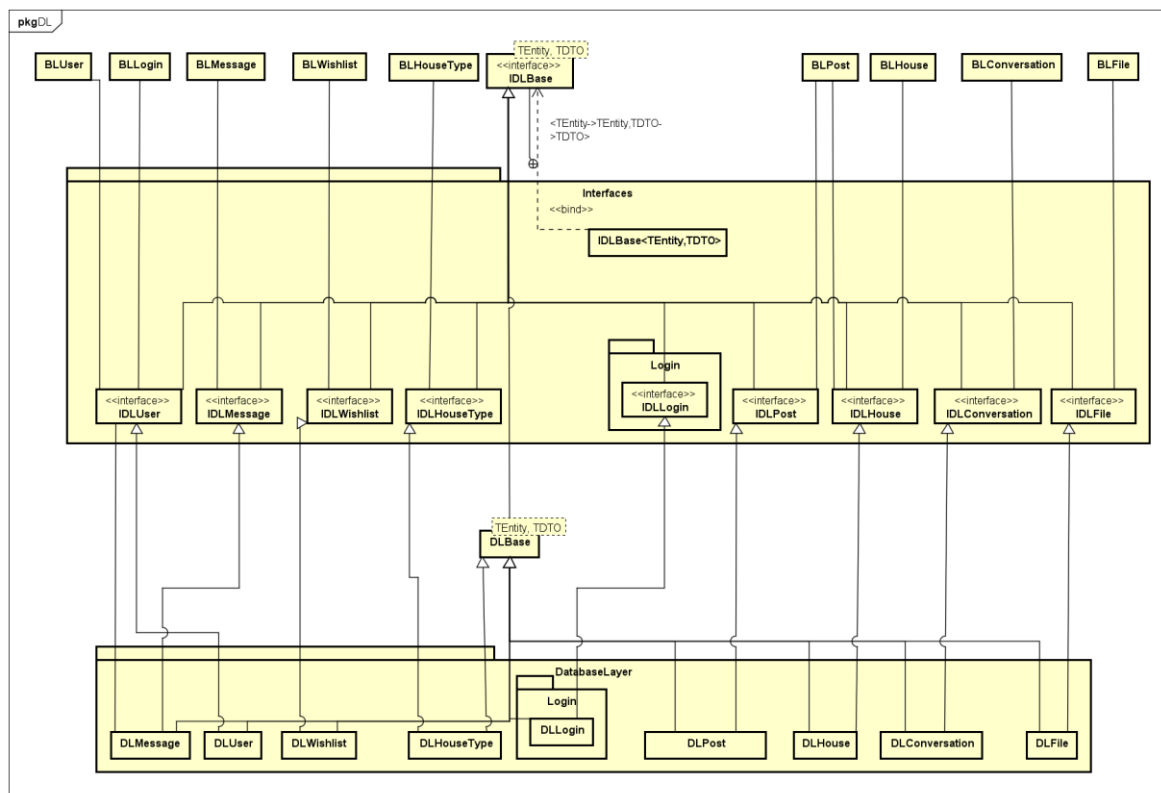
- Chathub: chứa các class thiết lập các kết nối chat, thực hiện các tác vụ liên quan đến chat
- Controllers: Chứa các đầu API là các điểm để client giao tiếp với server
- BL: chứa các inrterface và các class thực hiện công việc xử lý các nghiệp vụ

- DL: chứa các interface và các class thực hiện các công việc thao tác với cơ sở dữ liệu
- DBContext: chứa các model ánh xạ từ các bảng thuộc cơ sở dữ liệu SQLServer, chứa class thao tác với cơ sở dữ liệu bằng việc sử dụng EntityFramework
- EntityModels: Chứa các model sử dụng trong cả project, chứa các class DTO
- Utils: Chứa các class bao gồm các hàm sử dụng ở nhiều nơi trong toàn project

## 4.1.3 Thiết kế chi tiết gói

### 4.1.3.1 Thiết kế chi tiết gói phía server





Hình 20: Thiết kế chi tiết gói phía server

Hình 20 là biểu diễn chi tiết gói các package phía server. Như biểu diễn tổng quan thiết kế gói trình bày tại phần 4.1.2.2, phía server bao gồm 3 package chính Controller chứa các class đóng vai trò là các đầu API giúp giao tiếp giữa. Các controller này sẽ phụ thuộc vào các interface và class tại package BL bao gồm hai package Interface và BusinessLayer.

Package BusinessLayer chứa các class hiện thực hóa cho các interface nằm trong package Interface nhằm đảm bảo nhiệm vụ chung của package BL là thực hiện xử lý các nghiệp vụ của phần mềm.

Package DatabaseLayer chứa các class hiện thực hóa cho các interface nằm trong package Interface nhằm đảm bảo nhiệm vụ chung của package DL là thực hiện các thao tác chung với cơ sở dữ liệu.

## 4.2 Thiết kế chi tiết

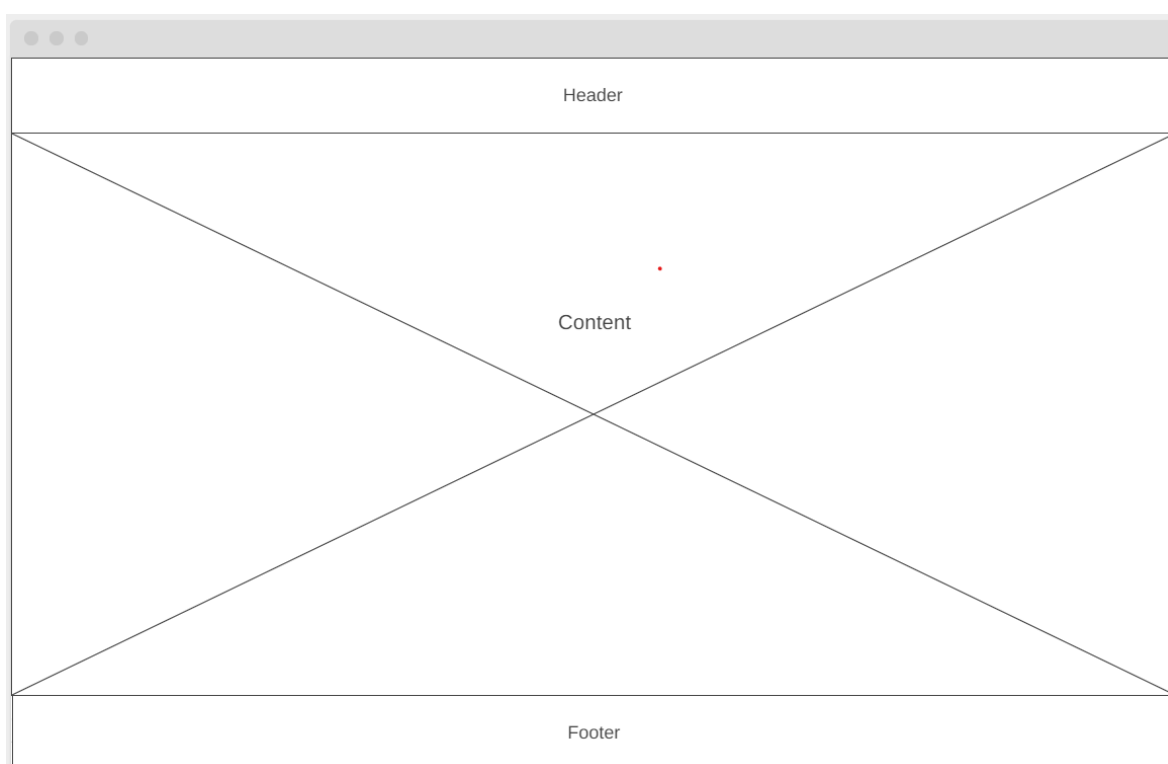
### 4.2.1 Thiết kế giao diện

Ứng dụng được thiết kế với một số các tiêu chí và quy định sau:

Bảng 9: Tiêu chí và quy định thiết kế

Tiêu chí	Nội dung
Độ phân giải	Đáp ứng tốt cho màn hình 1920x1080 hoặc 1366x768
Kích thước màn hình	Khuyến nghị màn hình từ 13inch trở lên để các thông tin, nội dung hiển thị tốt nhất

Nội dung trang web được tổ chức được tổ chức thành ba phần: phần header, phần nội dung và phần footer



Hình 21: Layout chung của web

Một số quy định về màu sắc, thiết kế trên trang web

Bảng 10: Tiêu chí và quy định về màu sắc

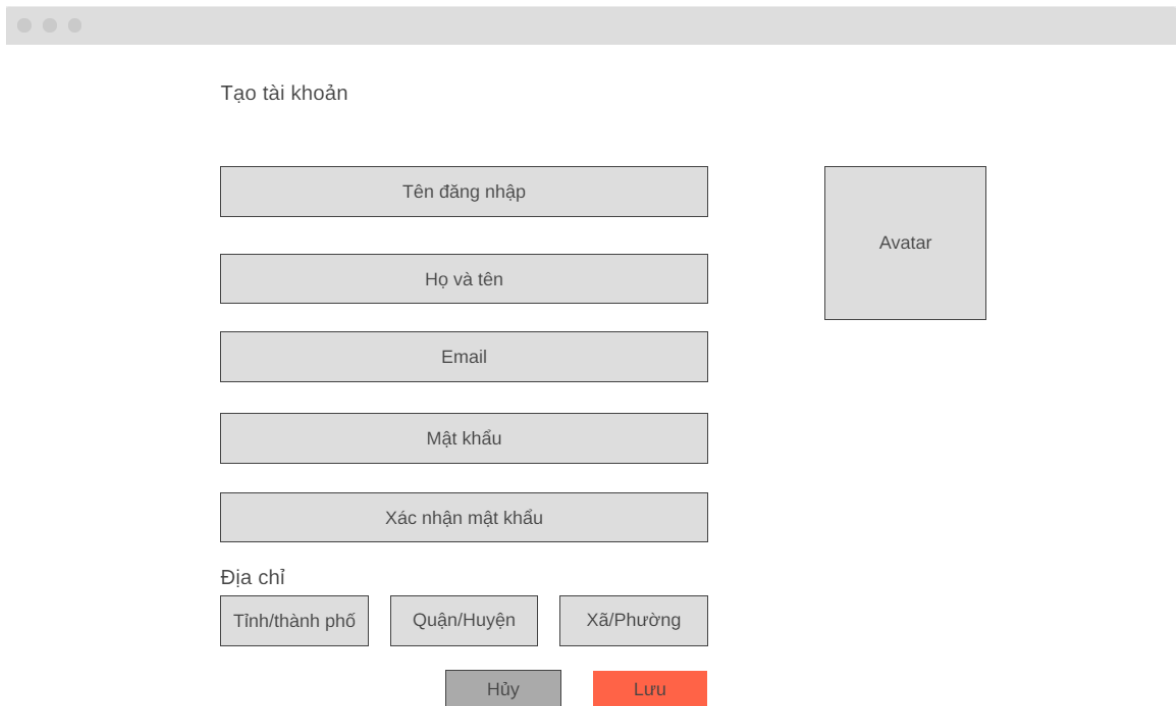
Tiêu chí	Nội dung
Màu nền phần nội dung	Màu trắng sáng (#fff)



Màu nền phần header và footer	Màu xám đen (#343a40)
Các phím chức năng	<p>Các phím có nội dung tạo mới: màu xanh lá #4caf50)</p> <p>Các phím có nội dung xóa: màu đỏ (#dc3545)</p>
Thông báo	<p>Với các ô nhập liệu, khi có lỗi, thông báo hiện ngay dưới ô input với màu chữ đỏ (#dc3545)</p> <p>Thông báo hệ thống hiện ra dạng popup ở giữa màn hình. Với các thông báo có nội dung tích cực, sẽ có icon dấu tích thể hiện nội dung tích cực. Ngược lại với các thông báo có nội dung</p>

Các thiết kế cụ thể được trình bày ở các mục dưới đây:

#### 4.2.1.1. Giao diện đăng kí



Tạo tài khoản

Tên đăng nhập

Họ và tên

Email

Mật khẩu

Xác nhận mật khẩu

Địa chỉ

Tỉnh/thành phố

Quận/Huyện

Xã/Phường

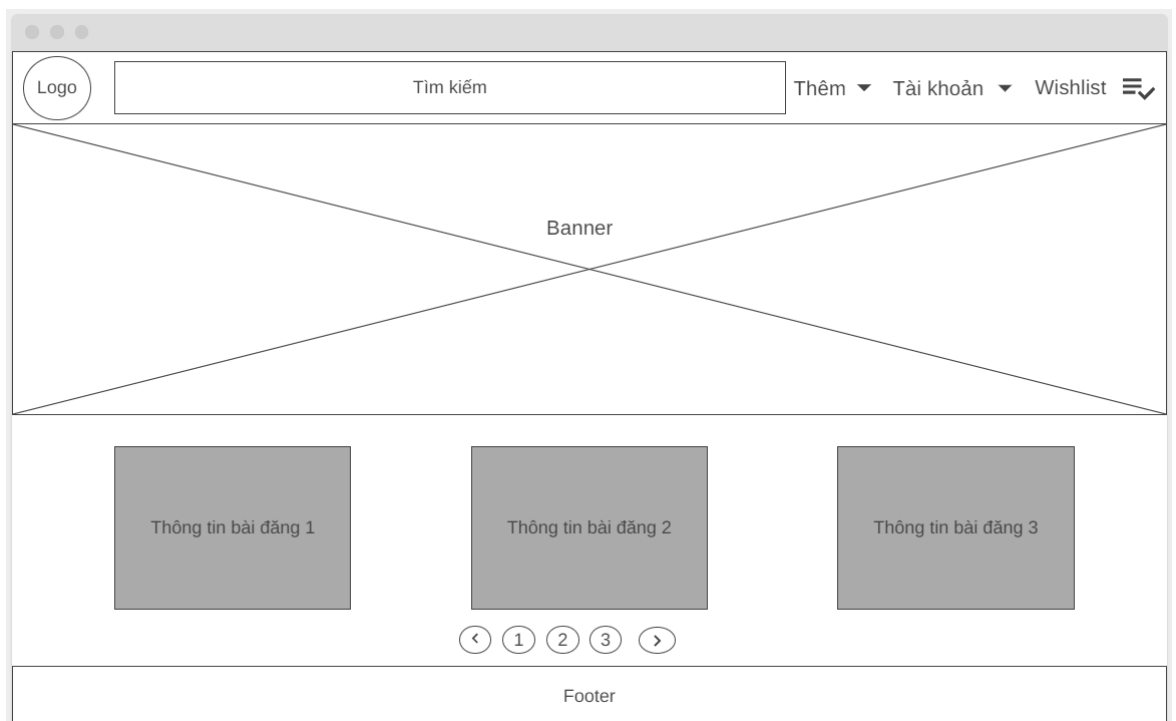
Hủy

Lưu

Avatar

Hình 22: Giao diện đăng kí

#### 4.2.1.2: Giao diện trang chủ



Logo

Tìm kiếm

Thêm ▾ Tài khoản ▾ Wishlist ≡

Banner

Thông tin bài đăng 1

Thông tin bài đăng 2

Thông tin bài đăng 3

< 1 2 3 >

Footer

Hình 23: Giao diện trang chủ

#### 4.2.1.1 Giao diện thông tin chi tiết nhà trọ



Hình 24: Giao diện thông tin chi tiết nhà trọ

#### 4.2.1.2 Giao diện quản lý tài khoản



Hình 25: Giao diện quản lý tài khoản

### 4.2.1.3 Giao diện quản lý bài đăng

Logo

Tìm kiếm

Thêm ▾ Tài khoản ▾

Danh sách bài đăng

Tên bài đăng	Tên nhà trọ	Tên chủ sở hữu	Trạng thái	
Tên bài đăng 1	Tên nhà trọ 1	Tên chủ sở hữu	Chưa phê duyệt	<div>Phê duyệt</div> <div>Xóa</div> <div>Lưu</div>
Tên bài đăng 1	Tên nhà trọ 1	Tên chủ sở hữu	Chờ thanh toán	<div>Xóa</div> <div>Lưu</div>
Tên bài đăng 1	Tên nhà trọ 1	Tên chủ sở hữu	Đã thanh toán	<div>Xóa</div> <div>Lưu</div>
Tên bài đăng 1	Tên nhà trọ 1	Tên chủ sở hữu	Chưa phê duyệt	<div>Phê duyệt</div> <div>Xóa</div> <div>Lưu</div>
Tên bài đăng 1	Tên nhà trọ 1	Tên chủ sở hữu	Chưa phê duyệt	<div>Phê duyệt</div> <div>Xóa</div> <div>Lưu</div>
Tên bài đăng 1	Tên nhà trọ 1	Tên chủ sở hữu	Chưa phê duyệt	<div>Phê duyệt</div> <div>Xóa</div> <div>Lưu</div>

<

1

2

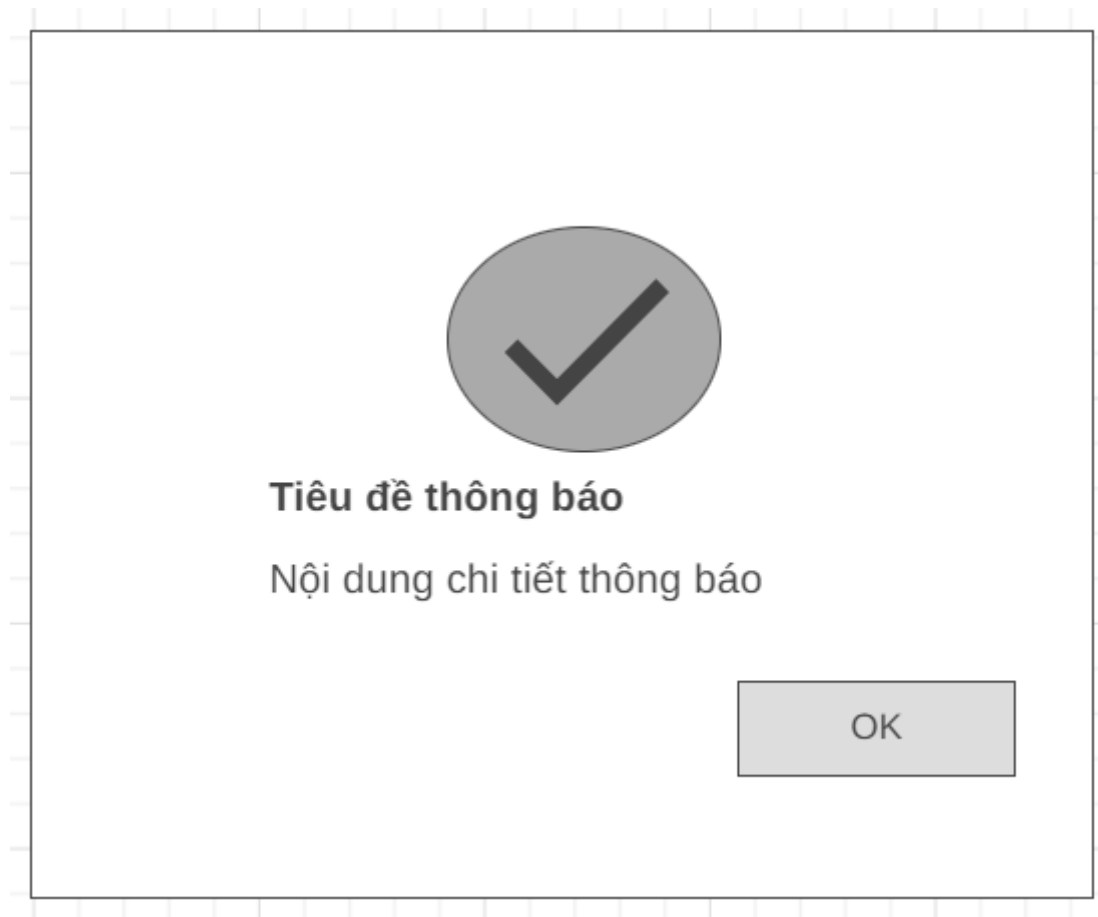
3

>

Footer

Hình 26: Giao diện quản lý bài đăng

### 4.2.1.4 Giao diện thông báo

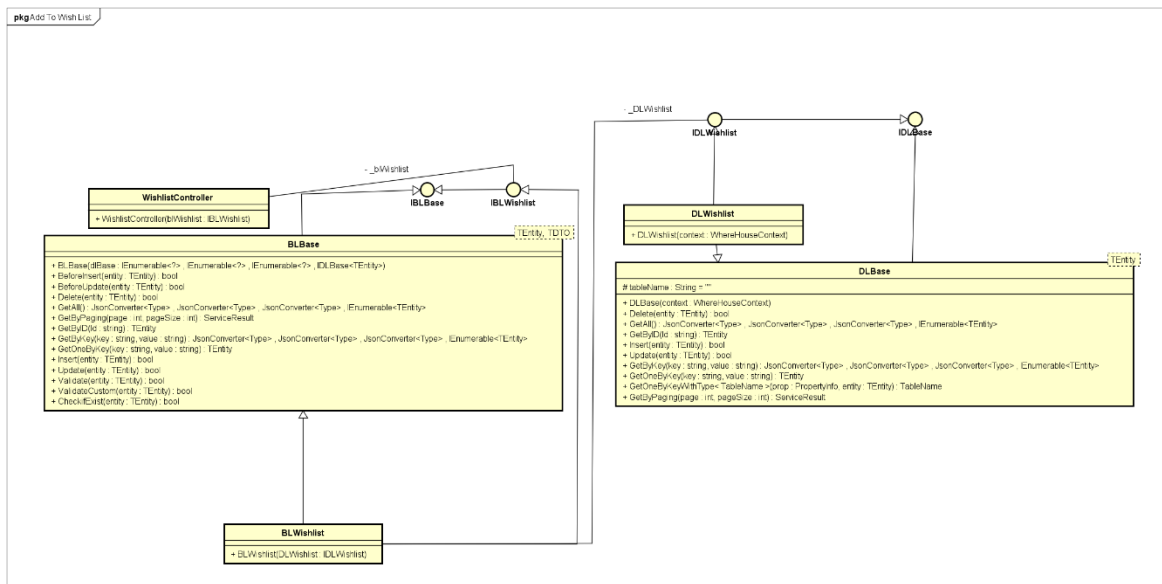


Hình 27: Giao diện thông báo

#### **4.2.2 Thiết kế lớp**

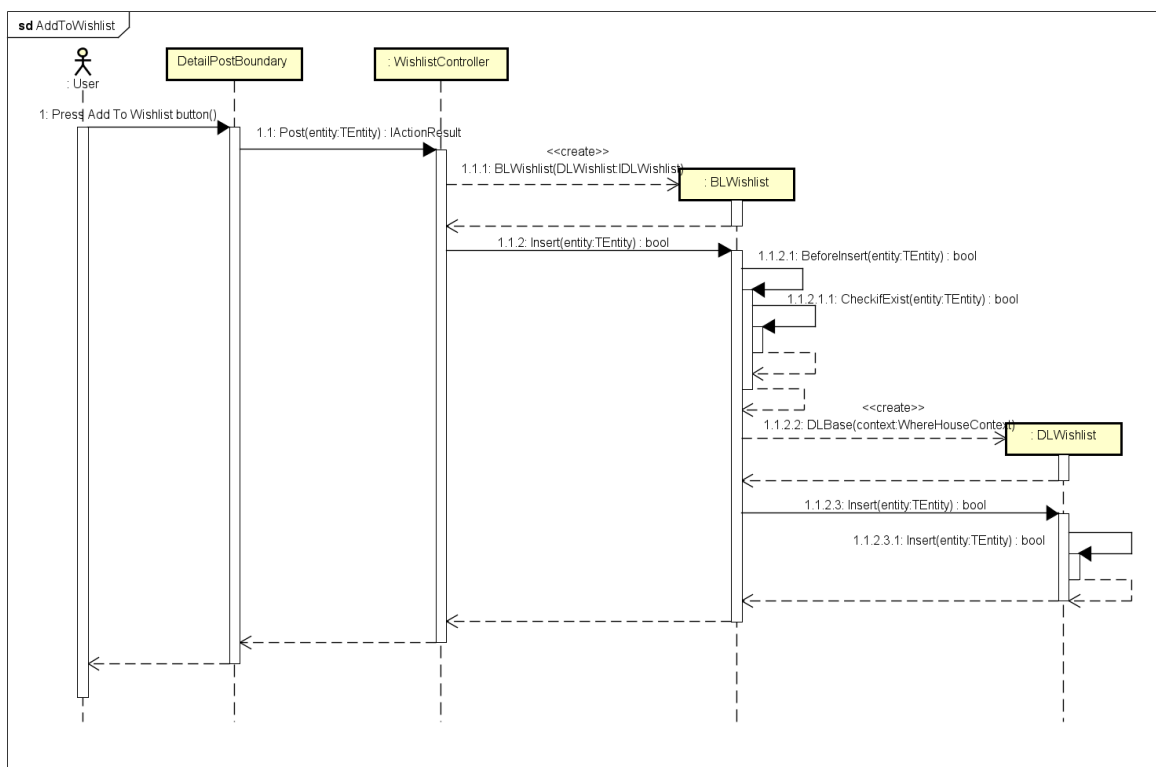
##### **4.2.2.1 Thiết kế lớp cho các lớp trong trình tự thêm nhà vào danh sách Wishlist**

- Biểu đồ lớp biểu diễn các lớp tham gia vào trình tự thêm nhà vào danh sách Wishlist của một User



Hình 28: Biểu đồ các lớp trong trình tự thêm nhà vào danh sách Wishlist

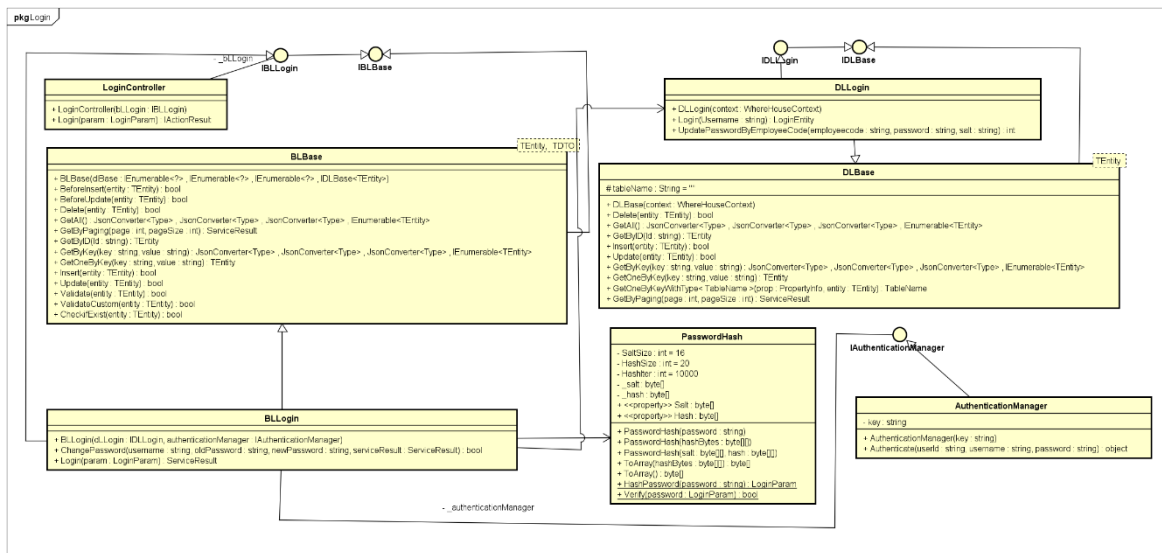
- Biểu đồ trình tự biểu diễn trình tự thêm nhà vào danh sách Wishlist của một User



Hình 29: Biểu đồ trình tự biểu diễn trình tự thêm nhà vào danh sách Wishlist của một User

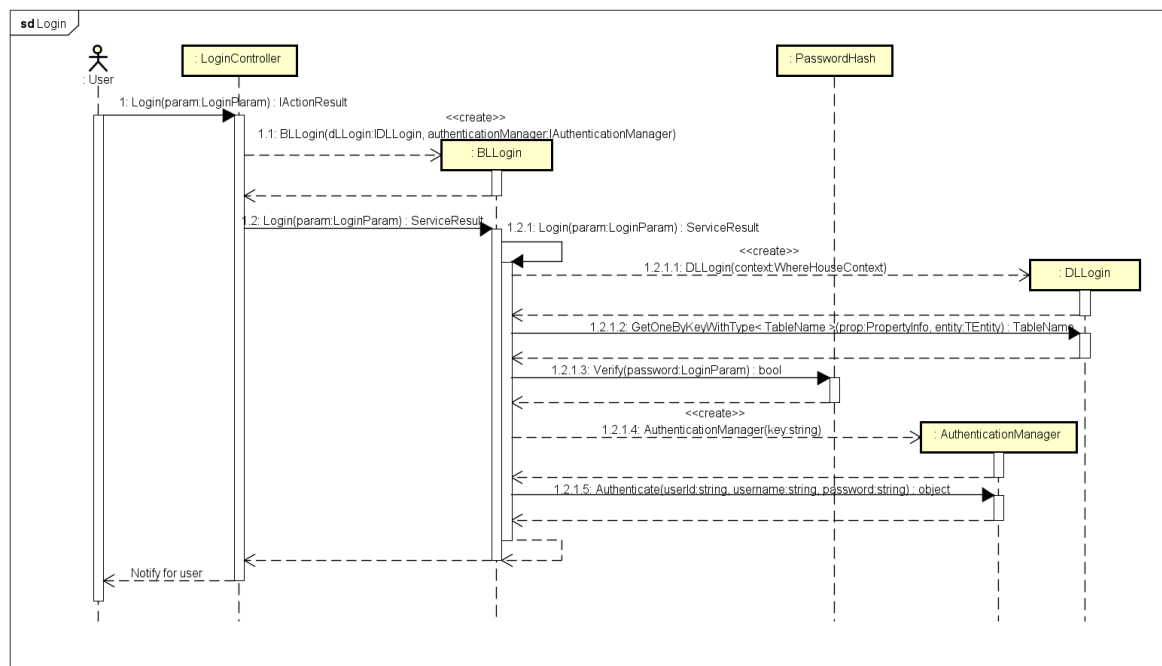
#### 4.2.2.2 Thiết kế lớp cho các lớp trong trình tự đăng nhập

- Biểu đồ lớp biểu diễn các lớp tham gia vào trình tự đăng nhập



Hình 30: Biểu đồ các lớp trong trình tự đăng nhập

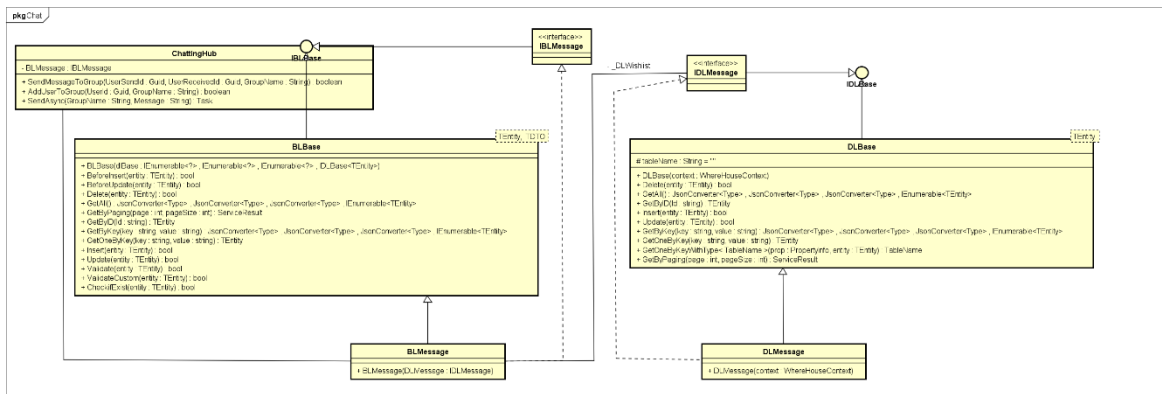
- Biểu đồ trình tự biểu diễn trình tự đăng nhập



Hình 31: Biểu đồ trình tự biểu diễn trình tự đăng nhập

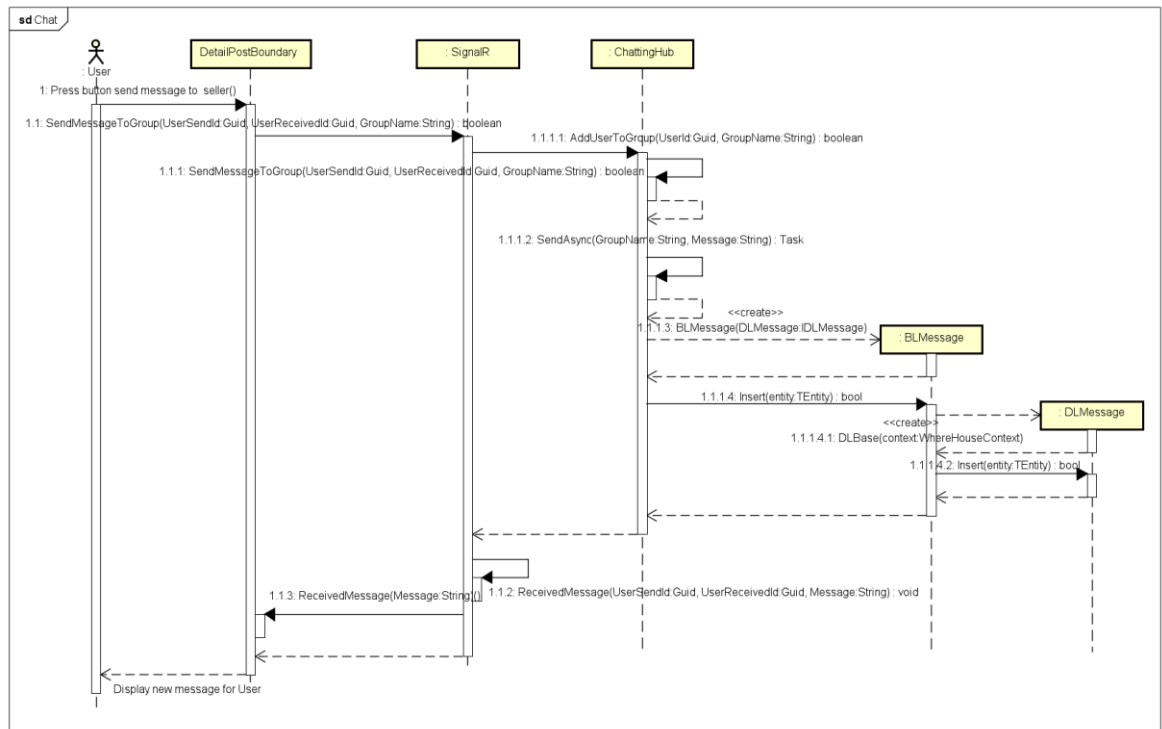
#### 4.2.2.3 Thiết kế lớp cho các lớp trong trình tự liên hệ với người bán

- Biểu đồ lớp biểu diễn các lớp tham gia vào trình tự liên hệ với người bán



Hình 32: Biểu đồ các lớp trong trình tự liên hệ với người bán

- Biểu đồ trình tự biểu diễn trình tự liên hệ với người bán



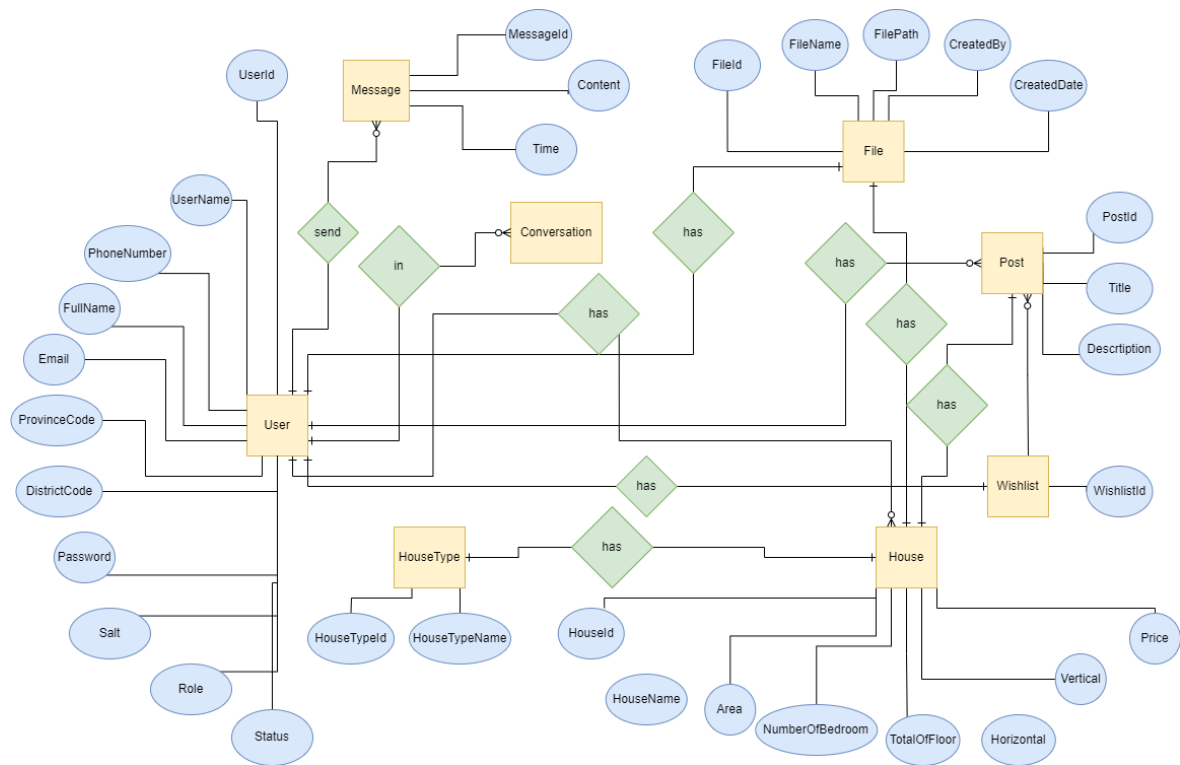
Hình 33: Biểu đồ trình tự biểu diễn trình tự liên hệ với người bán

- Trong trình tự liên hệ với người bán, việc liên hệ giữa người mua và người bán qua tin nhắn. Vì vậy, ứng dụng xây dựng một cơ chế real-time gửi và nhận tin nhắn giữa hai người. Áp dụng công nghệ WebSocket bằng việc sử dụng thư viện SignalR của ASP.Net. Chi tiết thông tin về việc áp dụng công nghệ vào trình tự liên hệ với người bán sẽ được trình bày trong chương 5.



### 4.2.3 Thiết kế cơ sở dữ liệu

Biểu đồ thực thể liên kết:



Hình 34: Biểu đồ thực thể liên kết

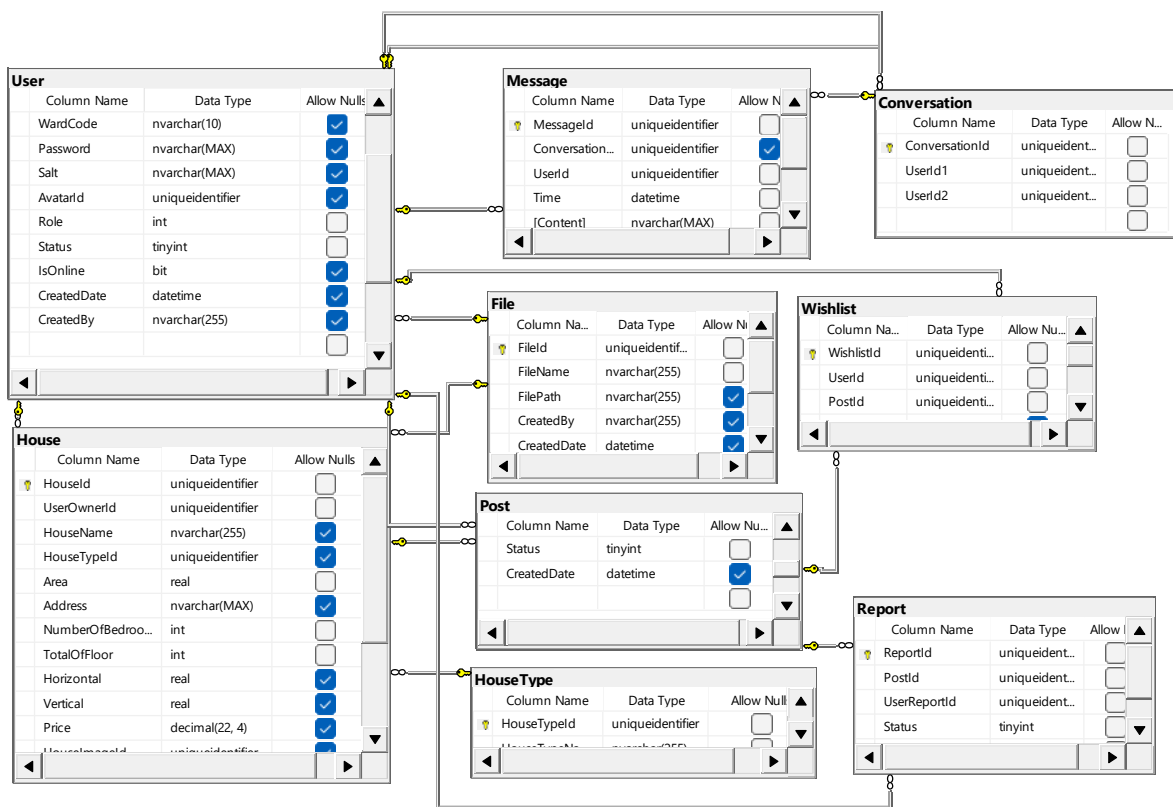
Mô tả về mô hình thực thể liên kết:

Bảng 11: Mô tả về thực thể mô hình liên kết

Thực thể	Mô tả
User	<p>Thông tin về người dùng</p> <ul style="list-style-type: none"> <li>- UserName: Tên đăng nhập</li> <li>- PhoneNumber: Số điện thoại</li> <li>- FullName: Họ và tên đầy đủ</li> <li>- Email: Email</li> <li>- ProvinceCode: Mã tỉnh/TP</li> <li>- DistrictCode: Mã quận/huyện</li> <li>- WardCode: Mã xã/phường</li> <li>- Password: Mật khẩu đăng nhập</li> <li>- Salt: Biến muối để mã hóa MK</li> <li>- Role: Vai trò</li> </ul>

	<ul style="list-style-type: none"> <li>- Status: Trạng thái người dùng</li> </ul>
House	<p>Thông tin về nhà trọ</p> <ul style="list-style-type: none"> <li>- HouseName: Tên ngôi nhà</li> <li>- Area: Diện tích (m2)</li> <li>- NumberOfBedroom: Số lượng phòng ngủ</li> <li>- TotalOfFloor: Tổng số tầng</li> <li>- Horizontal: Chiều rộng (m2)</li> <li>- Vertical: Chiều dài (m2)</li> <li>- Price: Giá thuê (/tháng)</li> </ul>
HouseType	<p>Thông tin về các loại nhà trọ</p> <ul style="list-style-type: none"> <li>- HouseTypeName: Tên loại nhà trọ</li> </ul>
File	<p>Thông tin về các file tải lên hệ thống</p> <ul style="list-style-type: none"> <li>- Filename: Tên file</li> <li>- FilePath: Đường dẫn file trên server</li> </ul>
Post	<p>Thông tin về bài đăng</p> <ul style="list-style-type: none"> <li>- Title: Tiêu đề bài đăng</li> <li>- Description: Mô tả chi tiết về bài đăng</li> </ul>
Message	<p>Thông tin về những tin nhắn giữa người thuê nhà và chủ nhà trọ</p> <ul style="list-style-type: none"> <li>- Time: Thời gian gửi tin nhắn</li> <li>- Content: Nội dung tin nhắn</li> </ul>
Wishlist	<p>Thông tin về danh sách yêu thích của người dùng (Chỉ bao gồm các nội dung tham chiếu đến thực thể khác)</p>
Conversation	<p>Thông tin về các đoạn hội thoại của người dùng (Chỉ bao gồm các nội dung tham chiếu đến thực thể khác)</p>

Từ mô hình thực thể-liên kết, database SQL Server với các bảng và liên kết giữa các bảng được xây dựng cụ thể như hình dưới đây.



Hình 35: Tổng quan thiết kế database

Trong đó, thông tin khóa chính và các khóa ngoại của các bảng cụ thể trình bày ở Bảng 12. Với các biểu diễn quy ước như sau: mỗi dòng có nền đỏ là khóa chính của bảng, dòng có nền xanh là khóa ngoại có tham chiếu đến khóa chính của một bảng khác được biểu diễn trên biểu đồ Hình 35.

Bảng 12: User

User	
<b><u>UserId</u></b>	<b><u>uniqueidentifier</u></b>
UserName	nvarchar
PhoneNumber	nvarchar
FullName	nvarchar
Email	nvarchar
ProvinceCode	nvarchar
DistrictCode	nvarchar
WardCode	nvarchar
Password	nvarchar
Salt	nvarchar
AvatarId	uniqueidentifier
Role	int

Status	tinyint
CreatedDate	datetime
CreatedBy	nvarchar

Bảng 13: File

File	
<b>FileId</b>	<b>uniqueidentifier</b>
FileName	nvarchar
FilePath	nvarchar
CreatedBy	nvarchar
CreatedDate	datetime

Bảng 14: House

House	
<b>HouseId</b>	<b>uniqueidentifier</b>
UserOwnerId	uniqueidentifier
HouseName	nvarchar
HouseTypeId	uniqueidentifier
Area	real
Address	nvarchar
NumberOfBedroom	int
TotalOfFloor	int
Horizontal	real
Vertical	real
Price	decimal
HouseImageId	uniqueidentifier
AddressByGoogle	nvarchar
CreatedDate	datetime
CreatedBy	nvarchar

Bảng 15: Housetype

HouseType	
<b>FileId</b>	<b>uniqueidentifier</b>

HouseTypeId	uniqueidentifier
HouseTypeName	nvarchar

Bảng 16: Message

Message	
<b>FileId</b>	<b>uniqueidentifier</b>
MessageId	uniqueidentifier
UserId	uniqueidentifier
Time	datetime
Content	nvarchar

Bảng 16: Conversation

Conversation	
<b>FileId</b>	<b>uniqueidentifier</b>
ConversationId	uniqueidentifier
UserId1	uniqueidentifier
UserId2	uniqueidentifier

Bảng 17: Post

Post	
<b>PostId</b>	<b>uniqueidentifier</b>
Title	nvarchar
Descrtiption	nvarchar
UserId	uniqueidentifier
HouseId	uniqueidentifier
Status	tinyint

Bảng 18: Wishlist

Wishlist	
<b>WishlistId</b>	<b>uniqueidentifier</b>
UserId	uniqueidentifier
PostId	uniqueidentifier
CreatedDate	datetime

CreatedTime	uniqueidentifier
-------------	------------------

Bảng 18: Report

Wishlist	
<b>ReportId</b>	<b>uniqueidentifier</b>
PostId	uniqueidentifier
UserReportId	uniqueidentifier
Status	tinyint

## 4.3 Xây dựng ứng dụng

### 4.3.1 Thư viện và công cụ sử dụng

Bảng 19: Danh sách thư viện và công cụ sử dụng

Mục đích	Công cụ	Địa chỉ URL	Phiên bản
IDE lập trình	Visual Studio Code	<a href="http://code.visualstudio.com/">http://code.visualstudio.com/</a>	1.69.0
	Microsoft Visual Studio Professional 2019	<a href="https://visualstudio.microsoft.com/vs/">https://visualstudio.microsoft.com/vs/</a>	16.11.2
Trình duyệt	Google Chrome	<a href="https://www.google.com/intl/vi_vn/chrome/">https://www.google.com/intl/vi_vn/chrome/</a>	
Quản lý phiên bản	Git		
Thao tác với database	Microsoft SQL Server Management Studio 18	<a href="https://docs.microsoft.com/en-us/sql/ssms/">https://docs.microsoft.com/en-us/sql/ssms/</a>	18.10
Cơ sở dữ liệu	Microsoft SQL Server	<a href="https://www.microsoft.com/en-us/sql-server/sql-server-downloads">https://www.microsoft.com/en-us/sql-server/sql-server-downloads</a>	15.0.2000.5
Hỗ trợ thao tác với cơ sở dữ liệu	SQL Prompt	<a href="https://www.red-gate.com/products/sql-development/sql-prompt/">https://www.red-gate.com/products/sql-development/sql-prompt/</a>	10.11.11.27632

Truy xuất cơ sở dữ liệu	Entity Framework	<a href="https://docs.microsoft.com/vi-vn/ef/ef6/?redirectedfrom=MSDN">https://docs.microsoft.com/vi-vn/ef/ef6/?redirectedfrom=MSDN</a>	5.0.0
Framework backend	ASP.Net Core		5.0
Quản lý kết nối chat	SignalR		
Framework xây dựng giao diện	VueJS	<a href="https://vuejs.org/">https://vuejs.org/</a>	2.6.11
Thư viện giao diện	Vuetify	<a href="https://vuetifyjs.com/en/">https://vuetifyjs.com/en/</a>	2.6.0
	Bootstrap	<a href="https://getbootstrap.com/">https://getbootstrap.com/</a>	5.1.3
Quản lý trạng thái trong VueJS	Vuex	<a href="https://vuex.vuejs.org/">https://vuex.vuejs.org/</a>	3.6.2
Kiểm soát chất lượng	Sonnar Qube	<a href="https://www.sonarqube.org">https://www.sonarqube.org</a>	9.5

#### 4.3.2 Kết quả đạt được

Sản phẩm được đóng gói gồm 3 phần chính

- Ứng dụng Server
- Ứng dụng client
- Mẫu Cơ sở dữ liệu(bao gồm các bảng đã đề cập ở trên cùng với tài khoản và mật khẩu)

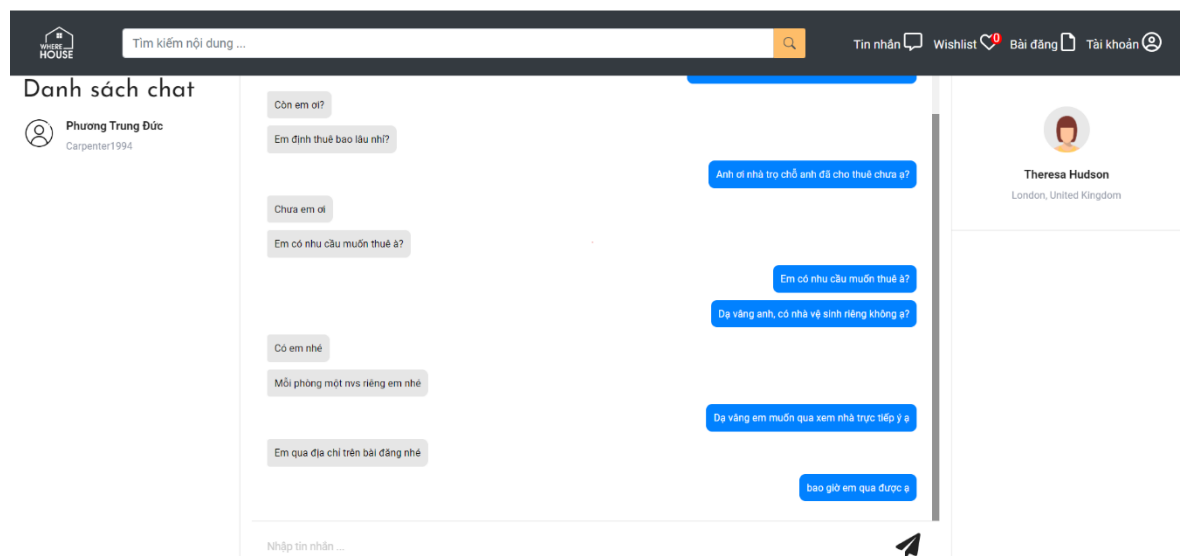
Thống kê về mã nguồn chi tiết trình bày ở Bảng 20.

Bảng 20: Thống kê mã nguồn

	Server	Client
--	--------	--------

Số lượng dòng code	2743	1891
Số lượng function	529	194
Số lượng class	98	Không thống kê được
Kích thước ứng dụng (Bao gồm cả cả thư viện đóng gói cùng)	47.7MB, 1221 Files	212Mb, 47622 Files

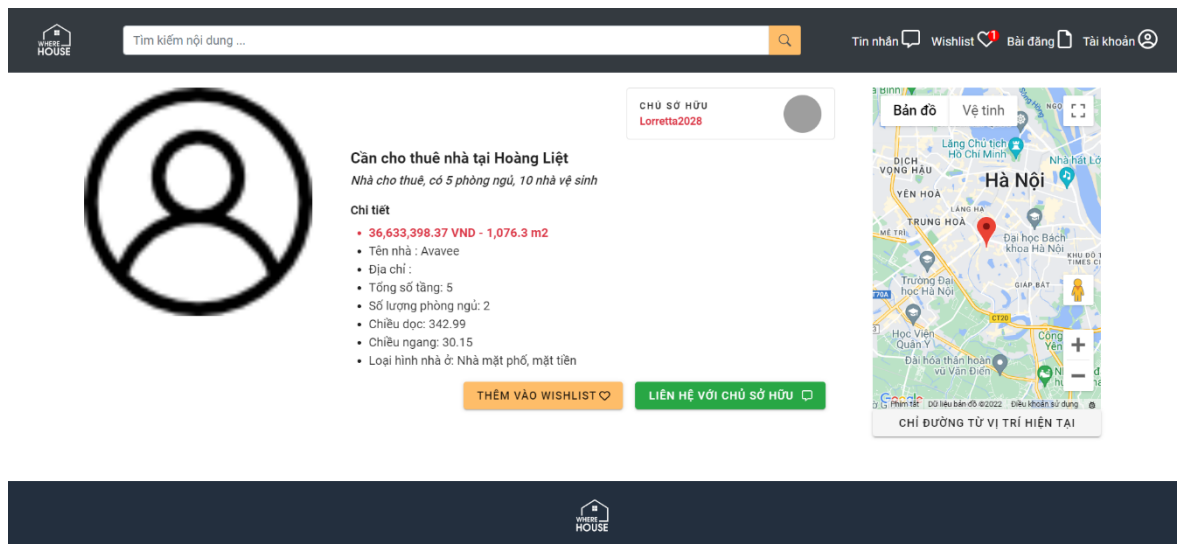
### 4.3.3 Minh hoạ các chức năng chính



Hình 36: Giao diện chat

Giao diện chat, khi người dùng ấn gửi, giao diện sẽ cập nhật tin lại đoạn hội thoại. Nếu người dùng nhấn chọn vào một trang các đoạn hội thoại đã tham gia thuộc phần Danh sách chat, hệ thống cũng hiển thị chi tiết đoạn hội thoại tương ứng.





Hình 37: Giao diện xem chi tiết bài đăng

Hình 37 giao diện xem chi tiết bài đăng, bao gồm các thông tin chi tiết về nhà trọ và một số thông tin về chủ sở hữu. Với trường hợp đây là bài đăng của một người dùng khác tài khoản đang đăng nhập, hệ thống hiển thị nút thêm vào wishlist và liên hệ với chủ sở hữu cho người dùng. Phần bản đồ hiển thị địa chỉ chi tiết địa chỉ của nhà trọ, người dùng có thể tìm đường từ vị trí hiện tại bằng nhấn vào nút Chỉ đường từ vị trí hiện tại, phần đường đi sẽ được hiển thị chi tiết trên bản đồ.

WHIT HOUSE

### Tạo tài khoản

Tên đăng nhập(\*)

Số điện thoại

Họ và tên

Email

Carpenter1994

Email không đúng định dạng.

Mật khẩu(\*)

\*\*\*\*\*

Xác nhận mật khẩu(\*)

Địa chỉ

Tỉnh/Thành ...

Quận/Huyện

Xã/Phường

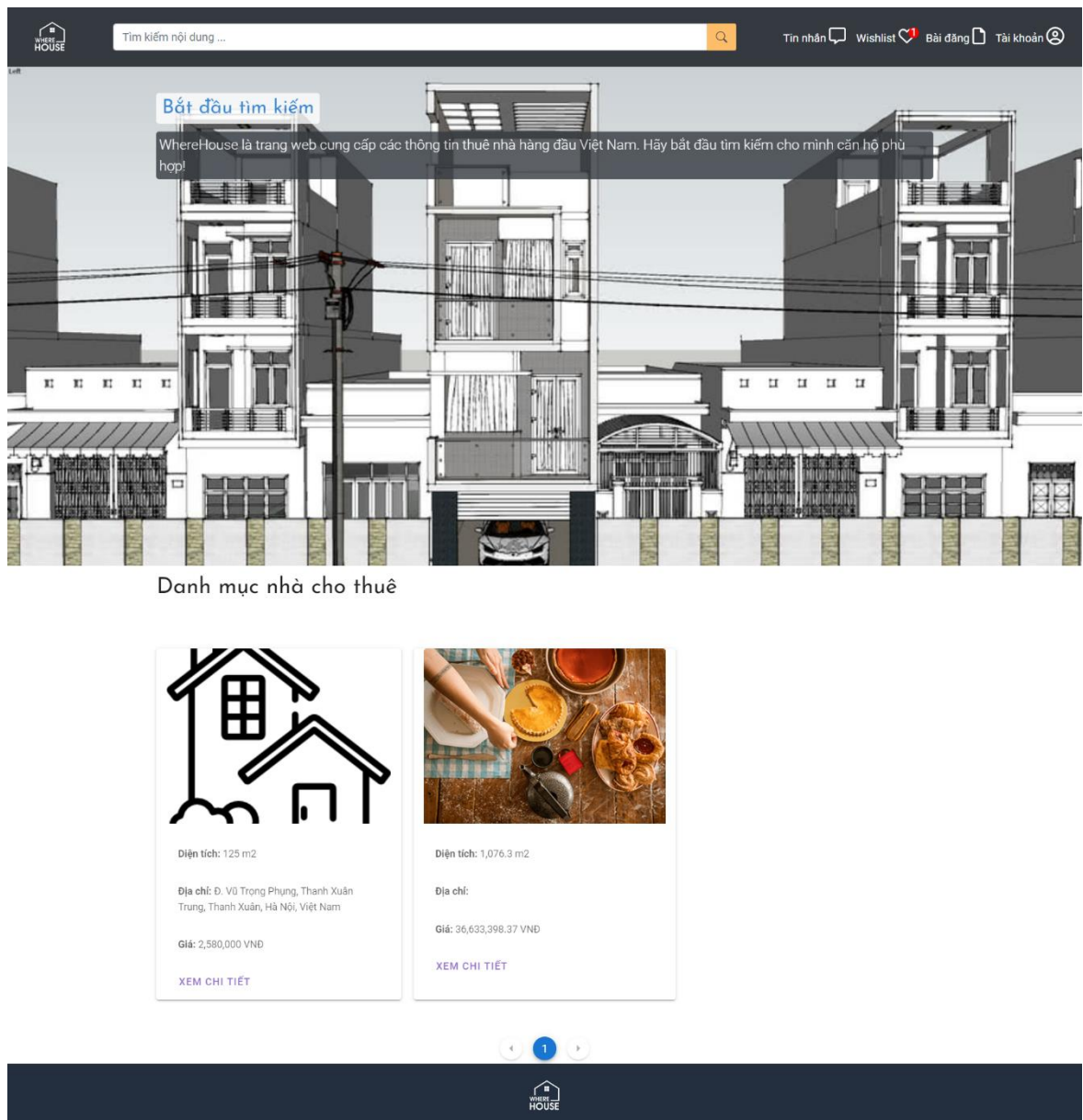
ĐĂNG KÝ

Bạn có một tài khoản?

ĐĂNG NHẬP TẠI ĐÂY

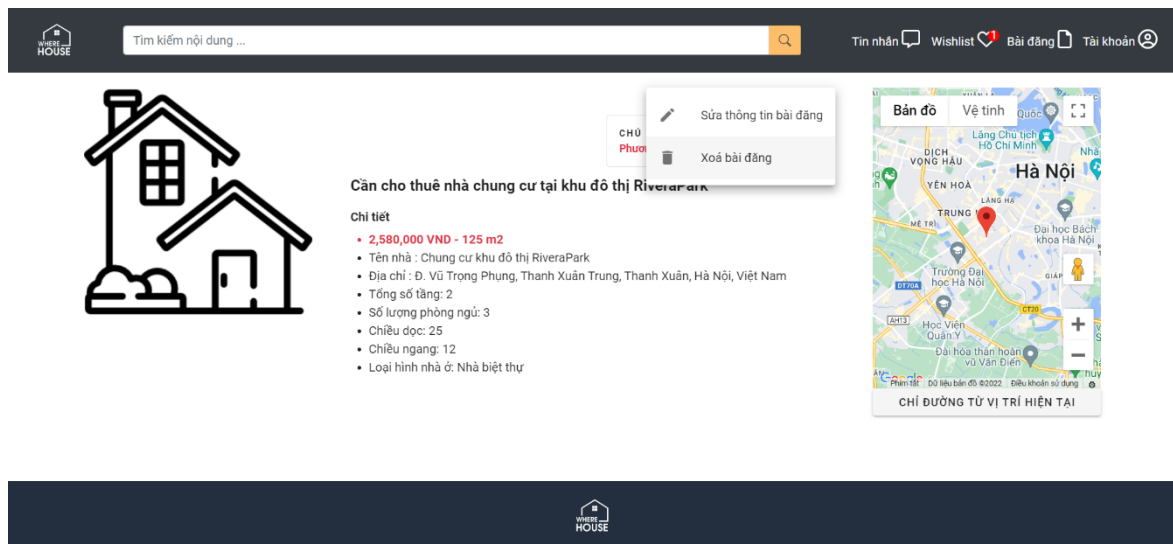
Hình 38: Giao diện đăng ký tài khoản

Với chức năng đăng ký, khi người dùng nhập liệu các ô input, nếu không đảm bảo điều kiện như độ dài tối thiểu, trường bắt buộc nhập ... sẽ có cảnh báo hiện ngay dưới các ô input và ô input chuyển đỏ cảnh báo. Việc tương tự cũng xảy ra khi người dùng ấn Đăng ký.



Hình 39: Giao diện trang chủ

Sau khi đăng nhập, người dùng được đưa tới giao diện trang chủ. Tại đây, người dùng có thể xem các bài đăng, nếu muốn xem thông tin chi tiết của các bài đăng, người dùng có thể lựa chọn Xem chi tiết ngay trên các bài đăng đó.



Hình 40: Giao diện chi tiết bài đăng

Với giao diện chi tiết bài đăng, nếu hệ thống kiểm tra bài đăng của người dùng hiện đang đăng nhập sẽ hiển thị chức năng chỉnh sửa và xóa bài đăng.

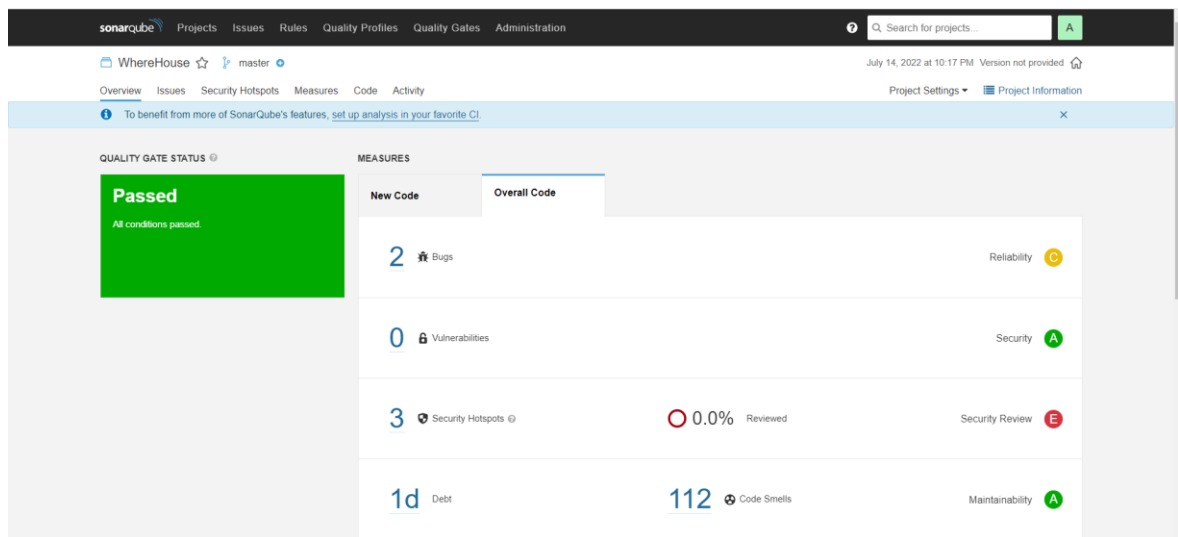
## 4.4 Kiểm thử

Thực hiện việc kiểm thử với ứng dụng Sonnar Qube - là công cụ phân tích và đánh giá chất lượng code dựa trên các tiêu chí về các lỗi (Bugs), lỗ hổng bảo mật (Security), khả năng bảo trì (Maintainability) và trùng lặp (Duplications) trong cú pháp, cấu trúc của code. Ngoài ra Sonnar Qube cũng có thể có thêm thông tin về khả năng bao phủ (Code Coverage) dựa trên các công cụ chạy kiểm thử đơn vị trong từng ngôn ngữ lập trình. Sonnar Qube không đánh giá được các lỗi liên quan đến logic của người lập trình.

Xếp hạng: A-B-C: Ổn định, chấp nhận để sử dụng

Loại E trở xuống: Có nhiều vấn đề, cần xem lại và cải thiện.

Kết quả cả hai ứng dụng client và server đều pass qua kiểm thử của ứng dụng:



Hình 41: Kết quả kiểm thử code với SonarQube

## 4.5 Triển khai

Hệ thống được triển khai trên môi trường máy local với cấu hình cụ thể:

Bảng 21: Môi trường triển khai cụ thể

Tiêu chí	Chi tiết
Hệ điều hành	Window
Ram	8GB
CPU	Intel Core i5-8300H 2.30 GHz
Màn hình	15.6 inch

Như đã đề cập, client và server và hai phần được chạy riêng biệt, hệ thống bao gồm cả giao diện và server chạy ổn định trên máy tính đã được triển khai.

# Chương 5 Các giải pháp và đóng góp nổi bật

## 5.1 Hiện thị các tiện ích ở khu vực lân cận nhà trọ

### 5.1.1 Đặt vấn đề

Sinh viên và người lao động đi thuê nhà trọ ngoài nhu cầu đảm bảo vị trí gần các trường đại học, khu công nghiệp... nơi làm việc, các thông tin về khu vực lân cận cũng là một yếu tố quan trọng mà người đi thuê nhà quan tâm đến. Nhà trọ cần phải đảm bảo gần các khu chợ, nhà hàng, quán ăn ... Vì vậy, ngoài cung cấp địa chỉ cho người dùng, ứng dụng cũng cần cung cấp được các thông tin về khu vực lân cận của nhà trọ như các cửa hàng, quán ăn, siêu thị.

### 5.1.2 Giải pháp

Với nhu cầu đặt ra yêu cầu ứng dụng cung cấp cho người dùng thông tin về địa chỉ nhà trọ, cũng như thông tin về các địa điểm tiện ích lân cận với nhà trọ. Ứng dụng sử dụng Google Maps cùng với các API cung cấp bởi Google Map API để đáp ứng yêu cầu đặt ra. Các thông tin địa chỉ như địa chỉ nhà trọ, Places API cung cấp các thông tin chính xác trên bản đồ như địa chỉ chi tiết, tọa độ ... Tuy nhiên, với một vị trí cụ thể, Google Map rất hạn chế hiển thị các thông tin lân cận để đảm bảo cho người dùng dễ quan sát, nên để lấy thêm các thông tin lân cận về một vị trí trên bản đồ ứng dụng sử dụng thêm Nearby Search API để truy vấn các thông tin về lân cận của khu trọ.

Google Maps là ứng dụng nổi bật và phổ biến của Google được hầu hết tất cả những người sử dụng internet tin dùng. Bởi nó cho phép người dùng có thể dễ dàng tìm kiếm các địa điểm mong muốn hay tìm đường đi một cách nhanh chóng cũng như cung cấp thông tin về mọi địa điểm xung quanh khu vực tìm kiếm. Tuy nhiên, để có thể sử dụng Maps trên các ứng dụng hoặc website phải cần đến sự hỗ trợ của Google Map API.

Google Map API có thể hiểu đơn giản là một phương thức hỗ trợ. Cho phép một ứng dụng có khả năng sử dụng dịch vụ hay hiển thị nội dung của một website khác mà Google Maps cung cấp như di chuyển, đánh dấu hay zoom trên bản đồ. Những ứng dụng nổi bật của Google Map API:

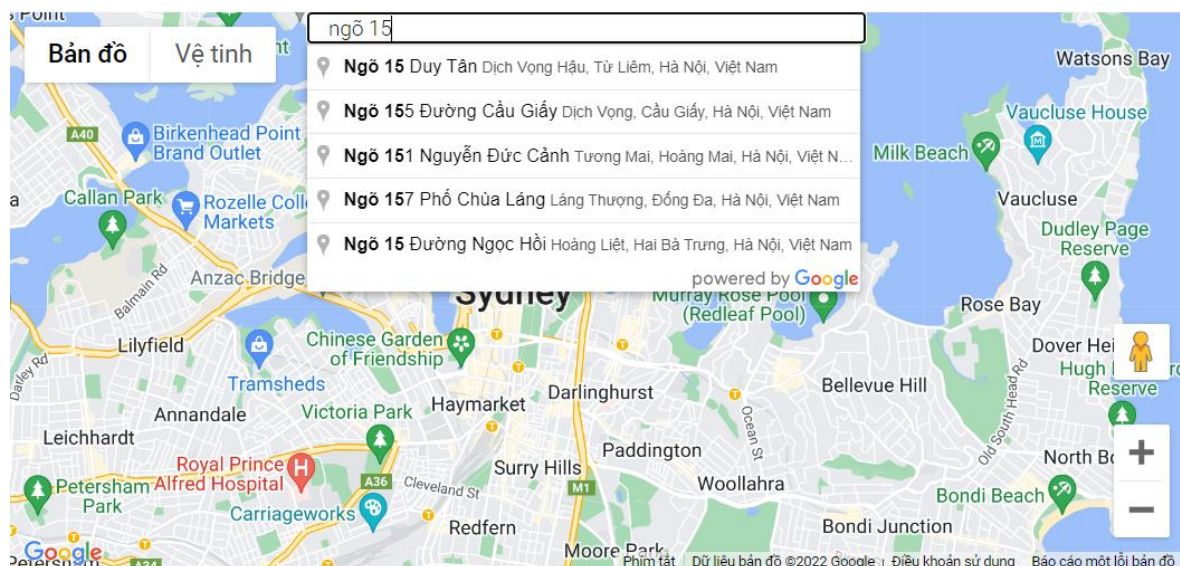
- Có thể dễ dàng đánh dấu được các vị trí, địa điểm với những thông tin có liên quan đến địa điểm đó trên bản đồ khi sử dụng Google Map API. Thông qua các dịch vụ

- Các vùng, các khu đặc biệt như khu ô nhiễm, khu sản xuất, khu đô thị đều dễ dàng nhận diện bằng cách khoanh vùng rõ ràng. Khi sử dụng Google Map API, có thể theo dõi lượng phương tiện giao thông xuất hiện tại khu vực đã được chỉ định.

```
[...]
```

```
{  
  "address_components": Array(5)  
    ▶ 0: {long_name: "Hố 15 Duy Tân", short_name: "Hố 15 Duy Tân", types: Array(1)}  
    ▶ 1: {long_name: "Địch Vọng Hào", short_name: "Địch Vọng Hào", types: Array(3)}  
    ▶ 2: {long_name: "Cầu Giấy", short_name: "Cầu Giấy", types: Array(2)}  
    ▶ 3: {long_name: "Hà Nội", short_name: "Hà Nội", types: Array(2)}  
    ▶ 4: {long_name: "Việt Nam", short_name: "VN", types: Array(2)}  
    length: 5  
    [[Prototype]]: Array(0)  
  , "adr_address": "<span class=\"street-address\">Hố 15 Duy Tân</span>, <span class=\"extended-address\">Địch Vọng Hào</span>, <span class=\"locality\">Cầu Giấy</span>, <span class=\"region\">Hà Nội</span>, <span class=\"country-name\">Việt Nam</span>  
  , "geometry":  
    ▶ location: _Ge  
      ▶ lat: f ()  
      ▶ lng: f ()  
      ▶ [[Prototype]]: Object  
    ▶ viewport: _Gf (vb: Bf, Sa: xf)  
    ▶ [[Prototype]]: Object  
  , "html_attributions": Array(0)  
    length: 0  
    [[Prototype]]: Array(0)  
  , "icon": "https://maps.gstatic.com/mapfiles/place_api/icons/v1/png_71/geocode-71.png"  
  , "icon_background_color": "#78909D"  
  , "icon_mask_base_url": "https://maps.gstatic.com/mapfiles/place_api/icons/v2/generic_pinlet" +  
    name: "Hố 15 Duy Tân"  
  , "place_id": "E6x0Z0B1DEIErIESbW63ulCBt4bnYzggVv7Jn5IEIjyqUe1llCBd4qdSBIae6pxk1EJbcBO4bu2z5wgVnl4GSho5IukKFAoc5tS1upPhLqES2wq4FQbcENHQEgntSDse5gi19Rip-cfgNlU3u"  
  , "reference": "E6x0Z0B1DEIErIESbW63ulCBt4bnYzggVv7Jn5IEIjyqUe1llCBd4qdSBIae6pxk1EJbcBO4bu2z5wgVnl4GSho5IukKFAoc5tS1upPhLqES2wq4FQbcENHQEgntSDse5gi19Rip-cfgNlU3u"  
  , "types": [ "route" ]  
  , "url": "https://www.google.com/maps/@33.05+15.04,35.00,15.00z&hl=vi&entry=point&f=false", "v": 1, "v2": 1, "v3": 1, "v4": 1, "v5": 1, "v6": 1, "v7": 1, "v8": 1, "v9": 1, "v10": 1, "v11": 1, "v12": 1, "v13": 1, "v14": 1, "v15": 1, "v16": 1, "v17": 1, "v18": 1, "v19": 1, "v20": 1, "v21": 1, "v22": 1, "v23": 1, "v24": 1, "v25": 1, "v26": 1, "v27": 1, "v28": 1, "v29": 1, "v30": 1, "v31": 1, "v32": 1, "v33": 1, "v34": 1, "v35": 1, "v36": 1, "v37": 1, "v38": 1, "v39": 1, "v40": 1, "v41": 1, "v42": 1, "v43": 1, "v44": 1, "v45": 1, "v46": 1, "v47": 1, "v48": 1, "v49": 1, "v50": 1, "v51": 1, "v52": 1, "v53": 1, "v54": 1, "v55": 1, "v56": 1, "v57": 1, "v58": 1, "v59": 1, "v60": 1, "v61": 1, "v62": 1, "v63": 1, "v64": 1, "v65": 1, "v66": 1, "v67": 1, "v68": 1, "v69": 1, "v70": 1, "v71": 1, "v72": 1, "v73": 1, "v74": 1, "v75": 1, "v76": 1, "v77": 1, "v78": 1, "v79": 1, "v80": 1, "v81": 1, "v82": 1, "v83": 1, "v84": 1, "v85": 1, "v86": 1, "v87": 1, "v88": 1, "v89": 1, "v90": 1, "v91": 1, "v92": 1, "v93": 1, "v94": 1, "v95": 1, "v96": 1, "v97": 1, "v98": 1, "v99": 1, "v100": 1, "v101": 1, "v102": 1, "v103": 1, "v104": 1, "v105": 1, "v106": 1, "v107": 1, "v108": 1, "v109": 1, "v110": 1, "v111": 1, "v112": 1, "v113": 1, "v114": 1, "v115": 1, "v116": 1, "v117": 1, "v118": 1, "v119": 1, "v120": 1, "v121": 1, "v122": 1, "v123": 1, "v124": 1, "v125": 1, "v126": 1, "v127": 1, "v128": 1, "v129": 1, "v130": 1, "v131": 1, "v132": 1, "v133": 1, "v134": 1, "v135": 1, "v136": 1, "v137": 1, "v138": 1, "v139": 1, "v140": 1, "v141": 1, "v142": 1, "v143": 1, "v144": 1, "v145": 1, "v146": 1, "v147": 1, "v148": 1, "v149": 1, "v150": 1, "v151": 1, "v152": 1, "v153": 1, "v154": 1, "v155": 1, "v156": 1, "v157": 1, "v158": 1, "v159": 1, "v160": 1, "v161": 1, "v162": 1, "v163": 1, "v164": 1, "v165": 1, "v166": 1, "v167": 1, "v168": 1, "v169": 1, "v170": 1, "v171": 1, "v172": 1, "v173": 1, "v174": 1, "v175": 1, "v176": 1, "v177": 1, "v178": 1, "v179": 1, "v180": 1, "v181": 1, "v182": 1, "v183": 1, "v184": 1, "v185": 1, "v186": 1, "v187": 1, "v188": 1, "v189": 1, "v190": 1, "v191": 1, "v192": 1, "v193": 1, "v194": 1, "v195": 1, "v196": 1, "v197": 1, "v198": 1, "v199": 1, "v200": 1, "v201": 1, "v202": 1, "v203": 1, "v204": 1, "v205": 1, "v206": 1, "v207": 1, "v208": 1, "v209": 1, "v210": 1, "v211": 1, "v212": 1, "v213": 1, "v214": 1, "v215": 1, "v216": 1, "v217": 1, "v218": 1, "v219": 1, "v220": 1, "v221": 1, "v222": 1, "v223": 1, "v224": 1, "v225": 1, "v226": 1, "v227": 1, "v228": 1, "v229": 1, "v230": 1, "v231": 1, "v232": 1, "v233": 1, "v234": 1, "v235": 1, "v236": 1, "v237": 1, "v238": 1, "v239": 1, "v240": 1, "v241": 1, "v242": 1, "v243": 1, "v244": 1, "v245": 1, "v246": 1, "v247": 1, "v248": 1, "v249": 1, "v250": 1, "v251": 1, "v252": 1, "v253": 1, "v254": 1, "v255": 1, "v256": 1, "v257": 1, "v258": 1, "v259": 1, "v260": 1, "v261": 1, "v262": 1, "v263": 1, "v264": 1, "v265": 1, "v266": 1, "v267": 1, "v268": 1, "v269": 1, "v270": 1, "v271": 1, "v272": 1, "v273": 1, "v274": 1, "v275": 1, "v276": 1, "v277": 1, "v278": 1, "v279": 1, "v280": 1, "v281": 1, "v282": 1, "v283": 1, "v284": 1, "v285": 1, "v286": 1, "v287": 1, "v288": 1, "v289": 1, "v290": 1, "v291": 1, "v292": 1, "v293": 1, "v294": 1, "v295": 1, "v296": 1, "v297": 1, "v298": 1, "v299": 1, "v300": 1, "v301": 1, "v302": 1, "v303": 1, "v304": 1, "v305": 1, "v306": 1, "v307": 1, "v308": 1, "v309": 1, "v310": 1, "v311": 1, "v312": 1, "v313": 1, "v314": 1, "v315": 1, "v316": 1, "v317": 1, "v318": 1, "v319": 1, "v320": 1, "v321": 1, "v322": 1, "v323": 1, "v324": 1, "v325": 1, "v326": 1, "v327": 1, "v328": 1, "v329": 1, "v330": 1, "v331": 1, "v332": 1, "v333": 1, "v334": 1, "v335": 1, "v336": 1, "v337": 1, "v338": 1, "v339": 1, "v340": 1, "v341": 1, "v342": 1, "v343": 1, "v344": 1, "v345": 1, "v346": 1, "v347": 1, "v348": 1, "v349": 1, "v350": 1, "v351": 1, "v352": 1, "v353": 1, "v354": 1, "v355": 1, "v356": 1, "v357": 1, "v358": 1, "v359": 1, "v360": 1, "v361": 1, "v362": 1, "v363": 1, "v364": 1, "v365": 1, "v366": 1, "v367": 1, "v368": 1, "v369": 1, "v370": 1, "v371": 1, "v372": 1, "v373": 1, "v374": 1, "v375": 1, "v376": 1, "v377": 1, "v
```

63



Hình 2: Giao diện ô tìm kiếm của Google Map API

Trong các thông tin trả về từ API của Google bao gồm 3 thông tin quan trọng cần thực hiện lưu trữ bao gồm: lat – Latitude (Vĩ độ), lng – Longitude (Kinh độ) và place\_id là id của vị trí đã chọn. Với 3 thông tin này, khi hiển thị vị trí chi tiết cho người dùng, tiếp tục sử dụng API PlacesService với tham số truyền vào là id hoặc lat và lng sẽ lấy được thông tin chi tiết về vị trí nhà trọ cùng các thông tin chi tiết về vị trí.

Ở đây, khi hiển thị vị trí nhà trọ trên bản đồ, như đã trình bày, để đảm bảo cho người dùng dễ dàng quan sát, Google Map hạn chế hiển thị thêm các thông tin lân cận của một địa điểm. Vì vậy với nhu cầu hiển thị các thông tin lân cận của khu trọ, cần thực hiện thêm việc truy vấn lấy thông tin bằng việc sử dụng API Nearby Search<sup>7</sup>. Danh sách các tham số được mô tả cụ thể trong tài liệu mô tả API của google. Cụ thể, với mong muốn lấy về thông tin nhà hàng, chợ, siêu thị, các cửa hàng, tham số type truyền vào để lấy dữ liệu sẽ có dạng: type=supermarket|store|restaurant. Từ đó sẽ nhận được kết quả trả về là thông tin chi tiết về các vị trí thỏa mãn tham số type đã truyền vào. Hệ thống sẽ thêm các vị trí này trên bản đồ tiện cho người dùng tiện theo dõi.

Ngoài ra, ứng dụng cũng hỗ trợ người dùng về việc tìm đường đến nhà trọ. Sử dụng đối tượng navigator hỗ trợ bởi HTML5 có thể lấy được vị trí hiện tại của người dùng. Từ vị trí hiện tại và vị trí của nhà trọ, ứng dụng sử dụng Direction Service<sup>8</sup> để lấy thêm thông tin

<sup>7</sup> [https://developers.google.com/maps/documentation/javascript/places#place\\_searches](https://developers.google.com/maps/documentation/javascript/places#place_searches) truy cập lần cuối 01/08/2022

<sup>8</sup> <https://developers.google.com/maps/documentation/javascript/directions> truy cập lần cuối 01/08/2022



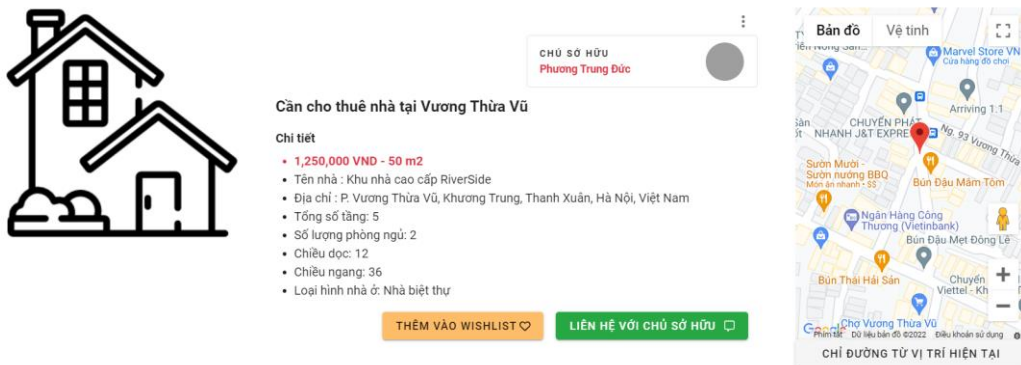
tuyến đường tối ưu và hiển thị thông tin này ngay trên bản đồ cho người dùng. Hình 3 là giao diện chi tiết khi áp dụng vào ứng dụng của ĐATN.



Hình 3: Giao diện dẫn đường cho người dùng đến địa chỉ nhà trọ

### 5.1.3 Kết quả đạt được

Với việc áp dụng Google Map API vào ứng dụng, nhu cầu hiển thị vị trí nhà trọ đồng thời hiển thị chi tiết thông tin lân cận khu trọ được triển khai tốt trên ứng dụng.



## 5.2 Giải pháp cho việc liên hệ giữa các người dùng

### 5.2.1 Đặt vấn đề

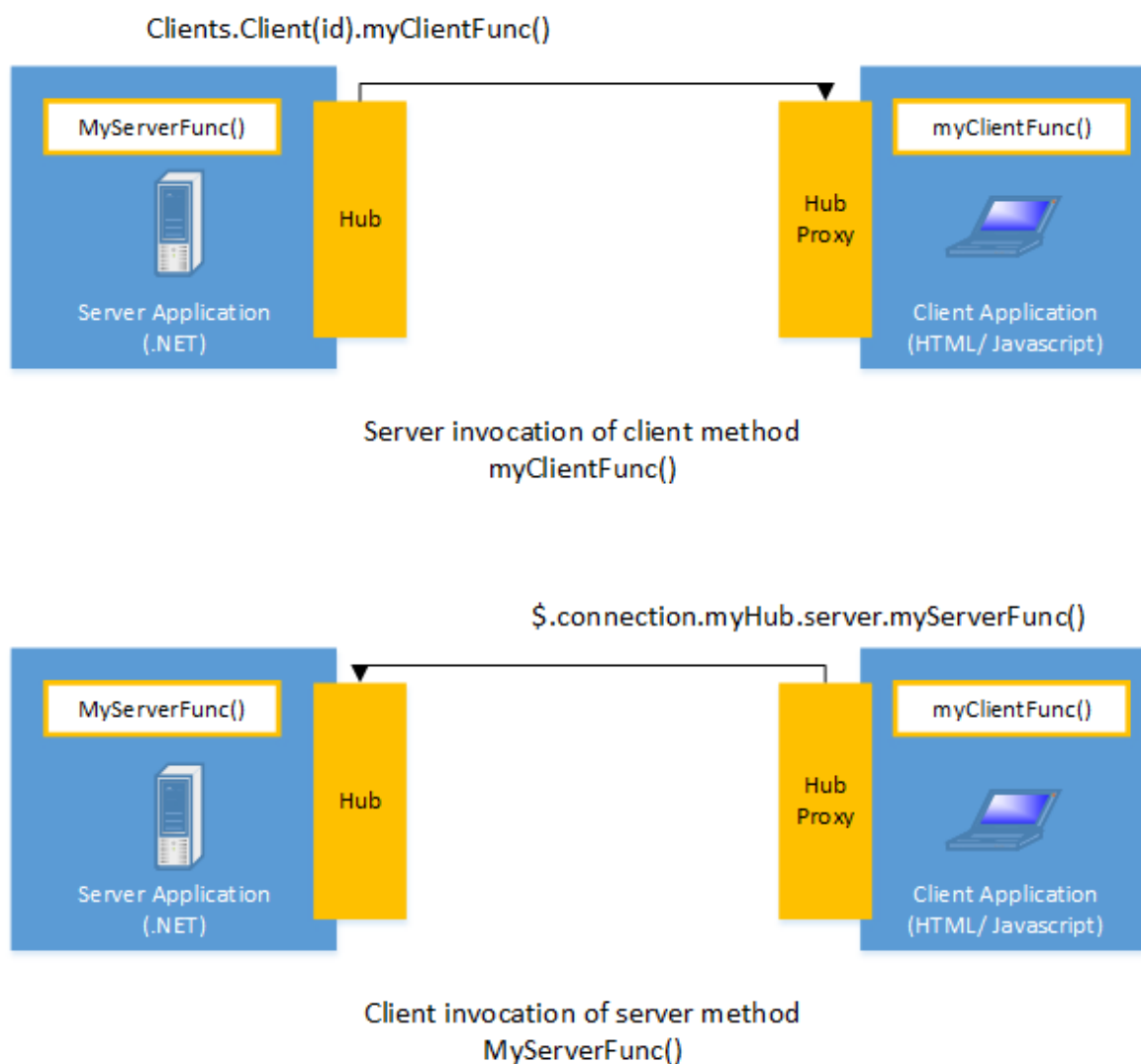
Ứng dụng cho thuê nhà cần phải kết nối được người cho thuê và người đi thuê nhà một cách hiệu quả nhất. Người thuê đôi khi muốn trao đổi chi tiết thêm về các thông tin nhà trọ với người cho thuê. Với các cách làm truyền thống như xem thông tin từ tờ rơi, bài đăng Facebook. Giữa hai bên phải liên hệ qua số điện thoại trên bài đăng hoặc một mạng xã hội trung gian như Zalo, Viber... Cách làm này tương đối bất tiện, như khi liên hệ qua Zalo cần phải nhận được sự đồng ý từ người dùng còn lại mới liên hệ được, đôi khi tin nhắn được gửi từ người lạ làm cho người cho thuê không nhận được tin nhắn của người quan tâm đến nhà trọ của mình.

Như vậy, việc kết nối trực tiếp qua kênh chat trên ứng dụng sẽ đem lại hiệu quả cao hơn. Với tính năng chat, hệ thống phải đảm bảo yêu cầu thời gian thực (real-time). Mỗi khi người dùng gửi tin nhắn cho chủ nhà, chủ nhà cũng phải ngay lập tức nhận được tin nhắn từ người dùng đã gửi.

### **5.2.2 Giải pháp**

Từ yêu cầu đó, ứng dụng đã sử dụng công nghệ Websocket. Websocket là 1 loại công nghệ hỗ trợ giao tiếp 2 chiều giữa Client và Server. Công nghệ này sử dụng giao thức TCP (Transmission Control Protocol) để kết nối thông tin với nhau trong môi trường Internet. Websocket cung cấp giao thức 2 chiều rất mạnh mẽ nên có độ trễ rất thấp. Thông tin trả về từ websocket nhanh chóng hơn so với HTTP request thông thường nên nó được sử dụng trong nhiều trường hợp cần thời gian thực như là chat, hiển thị biểu đồ hay thông tin chứng khoán... Với trường hợp cụ thể của ứng dụng là đảm bảo yêu cầu thời gian thực của tính năng chat, không thể dùng HTTP để kết nối vì nếu gửi lệnh AJAX liên tiếp tới server để lấy dữ liệu mới và cập nhật chúng lên màn hình sẽ tốn nhiều tài nguyên, traffic và thời gian trả lại không chính xác.

Với việc xây dựng server bằng công nghệ ASP.Net Core, ứng dụng sử dụng thư viện SignalR để triển khai tư tưởng trên. SignalR là một thư viện cho phép đơn giản hóa quá trình thêm chức năng web real-time trong phát triển ứng dụng. SignalR cung cấp một API đơn giản cho việc tạo server-to-client remote procedure call (RPC) để gọi những hàm javascript trong trình duyệt (và những nền tảng khác) từ code .Net của server-side. SignalR cũng bao gồm những API cho việc quản lý kết nối (connect và disconnect events) và những kết nối nhóm (Groups).



Hình 42: Mô hình làm việc của SinalR

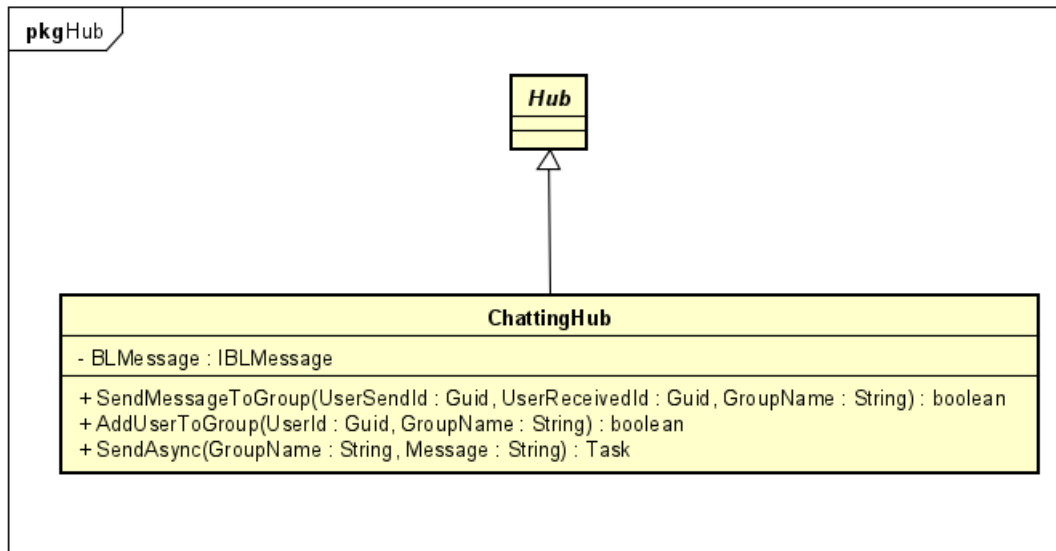
SignalR xử lý quản lý kết nối một cách tự động, và cho phép truyền đi thông điệp tới tất cả các client đã được kết nối một cách đồng loạt, giống như một chat room. Nó cũng hỗ trợ gửi những thông điệp tới những client được xác định. Kết nối giữa client và server là liên tục, không giống như kết nối HTTP cổ điển kết nối sẽ được lập lại sau mỗi lần giao tiếp.

SignalR API chứa hai mô hình cho việc giao tiếp giữa client và server: Persistent Connection và Hubs. Trong đó

- Connection đại diện một endpoint đơn giản cho việc gửi single-recipient, grouped, hoặc broadcast messages. Persistent Connection API đưa lập trình viên truy cập trực tiếp tới low-level của các giao thức giao tiếp mà SignalR hỗ trợ.
- Hub là một high-level đã xây dựng dựa trên Connection API mà cho phép client và server gọi những phương thức của nhau một cách trực tiếp. SignalR xử lý việc điều phối cho phép

clients gọi các phương thức trên server một cách dễ dàng như các phương thức cục bộ và ngược lại.

Ứng dụng sử dụng Hub. Với thiết kế cụ thể như Hình 43.



Hình 43: Thiết kế lớp áp dụng Hub

Trong đó, Hub là class abstract hỗ trợ sẵn của ASP.Net Core, ChattingHub kế thừa từ Hub sẽ là class chứa các đầu endpoint để giao tiếp giữa client và server. Như đã định nghĩa, với mô hình Hub, client có thể gọi trực tiếp các phương thức của server và ngược lại. Do đó, với một trình tự liên hệ với chủ sở hữu nhà trọ, từ client có thể gọi trực tiếp tới các phương thức của lớp ChattingHub để thiết lập các kết nối cần thiết cũng như gửi tin nhắn cho chủ sở hữu nhà trọ.

Điều đặc biệt của SignalR sử dụng Hub, nó linh hoạt sử dụng phương thức truyền tải WebSocket mới (các phương thức tự định nghĩa) và có thể trở lại với phương thức cũ hơn nơi cần thiết. Điều đó có ý nghĩa ứng dụng ngoài việc sử dụng WebSocket một cách trực tiếp, sử dụng SignalR nghĩa là rất nhiều chức năng mở rộng của WebSocket sẽ cần tự triển khai đã được làm sẵn hầu hết các phần quan trọng, mã nguồn sẽ được triển khai với những ưu điểm của WebSocket mà không phải lo lắng về việc phân chia code cho những client cũ hơn. SignalR cũng sẽ giúp cập nhật những thay đổi của WebSocket, từ đó SignalR sẽ tiếp tục được cập nhật để hỗ trợ những thay đổi trong truyền tải tầng bên dưới, cung cấp cho ứng dụng một giao diện thống nhất xuyên suốt các phiên bản của WebSocket.

### 5.2.3 Kết quả đạt được

Với giải pháp đã triển khai, người dùng đã có thể sử dụng tính năng liên hệ với chủ sở hữu qua tin nhắn (chat). Hệ thống đáp ứng được yêu cầu real-time, phản hồi ngay khi có tin nhắn từ người dùng.

## 5.3 Giải pháp cho việc triển khai thiết kế kiến trúc 3-layer

### 5.3.1 Đặt vấn đề

Trong việc phát triển phần mềm, một trong những yêu cầu đặt ra là mã nguồn phải đảm bảo các tiêu chí: dễ hiểu, dễ bảo trì, dễ dàng trong việc mở rộng trong tương lai. Như đã trình bày ở phần 4, với việc lựa chọn xây dựng ứng dụng với mô hình client-server và được triển khai bằng kiến trúc 3-layer. Những ưu điểm chính của mô hình này có thể kể ra như: mã nguồn tường minh, dễ bảo trì, dễ phát triển, tăng tính tái sử dụng của mã nguồn, dễ bàn giao dự án... Ngoài việc triển khai đúng mô hình, phần áp dụng mô hình 3-layer được kết hợp với áp dụng các design pattern và các kỹ thuật lập trình của ASP.Net Core tăng tính hiệu quả cho ứng dụng và mã nguồn dự án. Trong phần này, việc triển khai kiến trúc 3-layer sẽ được trình bày rõ hơn.

### 5.3.2 Giải pháp

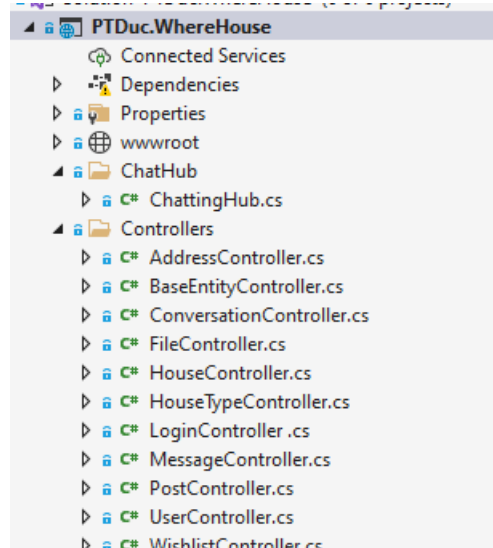
Trong DATN, ngoài việc áp dụng mô hình 3-layer thông thường, mã nguồn được triển khai linh hoạt bằng việc sử dụng các design pattern. Sử dụng kỹ thuật khởi tạo Scoped của C#, scoped được định nghĩa là một vòng đời HTTP request. Kỹ thuật này có dạng Singleton Pattern khởi tạo các đối tượng sử dụng trong một request duy nhất một lần, có nghĩa rằng các class sẽ được khởi tạo khi bắt đầu nhận request từ client và hủy bỏ khi trả lại kết quả response cho client đồng thời hủy kết nối HTTP. Điều này giúp tiết kiệm tài nguyên cho hệ thống, một đối tượng không phải khởi tạo nhiều lần trong một request.

Sử dụng kỹ thuật Dependence Injection của ASP.Net Core, hủy sự phụ thuộc của các class ở mức cao vào class ở mức thấp đảm bảo tính SOLID của mã nguồn. Dependence Injection là một kỹ thuật trong đó một đối tượng cung cấp các phụ thuộc của một đối tượng khác, ở đây có thể hiểu một class sẽ cung cấp các khởi tạo các class khác thông qua việc khai báo phụ thuộc giữa class được khởi tạo và interface nó triển khai. Khi áp dụng kỹ thuật này, class Startup của ASP.Net Core chính là một class như vậy, nó đóng vai trò như một Factory trong Factory Method Pattern, chịu trách nhiệm khởi tạo các class tương ứng với các interface đã chỉ định.

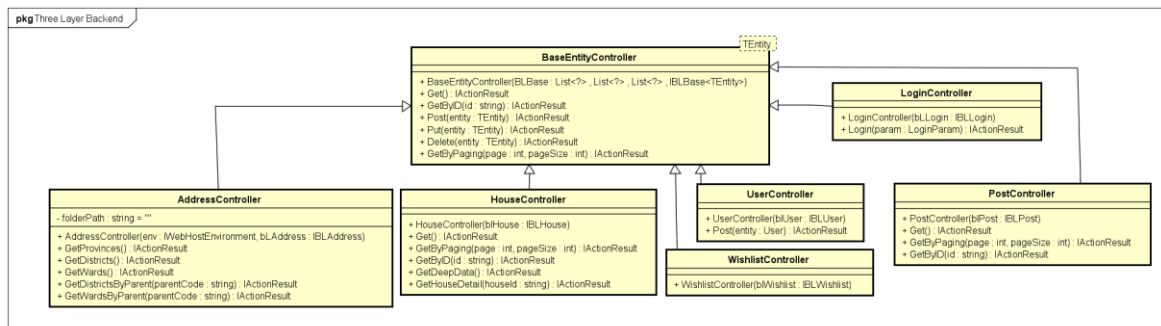
Triển khai cụ thể của 3 layer và việc áp dụng các Design Pattern sẽ được trình bày cụ thể dưới đây.

### 5.3.2.1 Presentation Layer

Với nhiệm vụ chính là giao tiếp với người dùng, nên ở layer chứa các module giao diện cùng các đầu controller – là các API để client có thể tiếp thực hiện giao tiếp với server. Hình 44 là triển khai chi tiết trong đồ án tốt nghiệp với các đầu API trong thư mục Controllers.



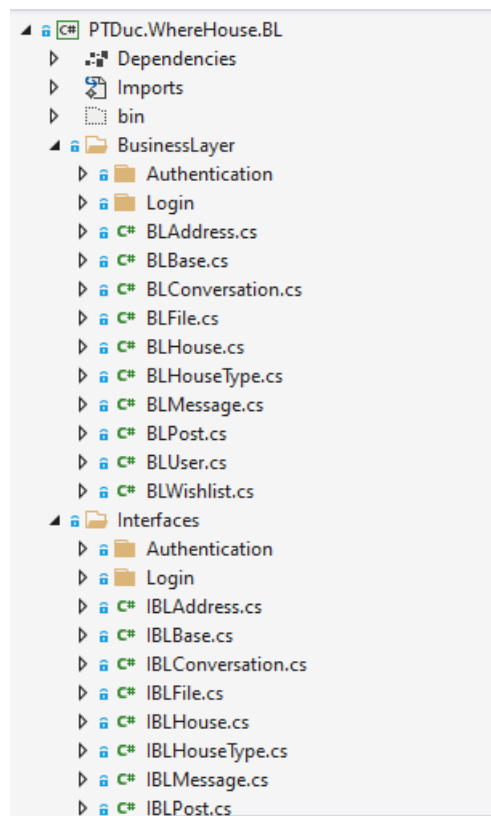
Hình 44: Thư mục Presentation Layer



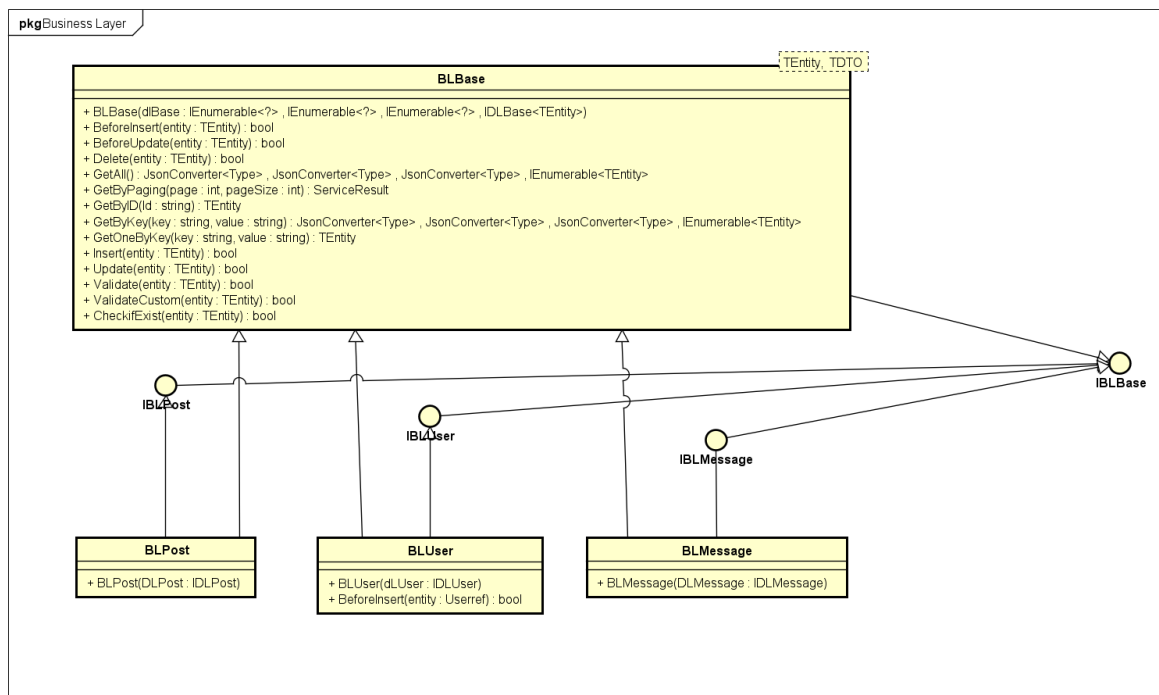
Hình 45: Biểu đồ lớp cho Presentation Layer

### 5.3.2.2 Business Logic Layer (BLL)

Với nhiệm vụ là xử lý các logic liên quan đến nghiệp vụ, xử lý các dữ liệu nhận được từ Presentation Layer nên các lớp ở layer này thực hiện một số các nhiệm vụ như: mã hóa mật khẩu khi thêm người dùng, kiểm tra tồn tại người dùng, kiểm tra mật khẩu đã nhập có đúng với người dùng đang thực hiện đăng nhập hay không ... Hình 46, triển khai cụ thể của business logic layer với các interface nằm trong thư mục Interfaces và các class BL nằm trong thư mục BusinessLayer là triển khai của các interface đã khai báo.



Hình 46: Thư mục Business Logic Layer



Hình 47: Biểu đồ lớp cho Business Logic Layer

Vì số lượng các lớp nhiều, nếu biểu diễn hết lên biểu đồ sẽ gây rối mắt, các lớp cũng có ý tưởng thiết kế tương tự nhau nên ở báo cáo này chỉ biểu diễn trên biểu đồ lớp 3 lớp. Các lớp còn lại cũng có ý tưởng thiết kế tương tự như ba lớp đã biểu diễn.

### 5.3.2.3 Data Access Layer (DAL)

Với nhiệm vụ giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu (tìm kiếm, thêm, xóa, sửa...). Các class thuộc layer này được xây dựng để thực hiện các thao tác thêm một bản ghi, thêm nhiều bản ghi, update các bản ghi, lấy dữ liệu phân trang, lấy dữ liệu theo ID ...

Với thiết kế cụ thể như trình bày trong Hình 48, triển khai cụ thể của data access layer với các interface nằm trong thư mục Interfaces và các class DL nằm trong thư mục DatabaseLayer là triển khai của các interface đã khai báo.





kiểu control trong thiết kế hướng đối tượng nhằm đạt được yêu cầu loose coupling. Với Dependence Injection nó cho phép việc khởi tạo các đối tượng mà một lớp phụ thuộc bên ngoài phạm vi lớp đó và cung cấp các đối tượng cho lớp thông qua các cách khác nhau.

Như đã trình bày, với dependence injection, một class hay cụ thể với dự án ASP.Net Core là class Startup sẽ đóng vai trò là class khởi tạo cung cấp phụ thuộc cho một class khác hay cụ thể hơn là cung cấp khởi tạo của một class ứng với một interface.

Có ba cách để áp dụng Dependence Injection:

- **Constructor Injection:** injector cung cấp việc khởi tạo đối tượng thông qua constructor của một lớp.
- **Property Injection:** injector cung cấp việc khởi tạo đối tượng thông qua một thuộc tính công khai (public property)
- **Method Injection:** khai báo một interface đã định nghĩa sẵn các hàm cung cấp các phương thức hay thuộc tính cần thiết, lớp bị phụ thuộc sẽ triển khai chính interface này.

Ứng dụng thuê nhà trong ĐATN này lựa chọn cách áp dụng Constructor Injection bằng việc khai báo một thuộc tính riêng tư (private property) của lớp và khởi tạo nó trong constructor của lớp đó. Thay vì phải truyền các đối tượng có kiểu khác vào các class phụ thuộc mỗi khi có sự thay đổi về nghiệp vụ cần phải tạo class mới, chỉ cần thêm khai báo vào class Startup, class này sẽ tự tạo ra class tương ứng mỗi khi có yêu cầu khởi tạo đối tượng.

Việc khởi tạo có thể không cần chỉ định trực tiếp, việc khởi tạo có thể rút gọn bằng các sử dụng phương thức AddScoped của C# trong class Startup.cs, class này sẽ được gọi khi khởi tạo ứng dụng, AddScoped sẽ nhận hai tham số là Interface và class triển khai interface đó. Sau khi được AddScoped, hiện tượng đa hình của lập trình hướng đối tượng sẽ bị loại bỏ do class triển khai interface đã được chỉ định. Mỗi khi có nâng cấp hoặc thay đổi, ta chỉ cần thêm một lớp mới triển khai interface đã có sẵn và thay thế class đã AddScoped trước đó. Nó có tác dụng như một Factory Method, khởi tạo class được chỉ định một cách rất linh hoạt. Đồng thời, ưu điểm của AddScoped là các class sẽ không được khởi tạo lại trong một HTTP request, nó sẽ chỉ được tạo ra khi nhận được request từ client. Điều này giúp tiết kiệm tài nguyên cho máy chủ. Hình ... là hình ảnh cụ thể của việc triển khai AddScoped trong class Startup mà ĐATN này đã sử dụng.

```
services.AddScoped<IBLUser, BLUser>();  
services.AddScoped<IDLUser, DLUser>();  
  
services.AddScoped<IBLLogin, BLLogin>();  
services.AddScoped<IDLLogin, DLLogin>();  
  
services.AddScoped<IBLHouse, BLHouse>();  
services.AddScoped<IDLHouse, DLHouse>();  
  
services.AddScoped<IBLAddress, BLAddress>();  
  
services.AddScoped<IBLFile, BLFile>();  
services.AddScoped<IDLFile, DLFile>();  
  
services.AddScoped<IDLWishlist, DLWishlist>();  
services.AddScoped<IBLWishlist, BLWishlist>();  
  
services.AddScoped<IDLPost, DLPost>();  
services.AddScoped<IBLPost, BLPost>();
```

Hình 50: Dependence Injection trong class Startup

### 5.3.3 Kết quả đạt được

Mã nguồn được triển khai đúng theo cấu trúc lý thuyết của 3-layer và áp dụng được linh hoạt kỹ thuật dependence injection cũng như addscoped của APS.Net Core.

# Chương 6 Kết luận và hướng phát triển

## 6.1 Kết luận

Tổng kết, trên các cơ sở đã phân tích, hệ thống em đã xây dựng đã có thể hoạt động và đáp ứng được các tính năng của một ứng dụng cho thuê nhà như. Đáp ứng việc chia sẻ thông tin về nhà thuê giữa người đi thuê nhà và chủ sở hữu nhà trọ, đảm bảo liên hệ giữa hai bên và ứng dụng hoạt động cũng đem lại nguồn thu để duy trì nếu đem triển khai trên môi trường thực tế.

Tuy nhiên, trên tư tưởng đó, hệ thống vẫn dừng lại ở mức tư tưởng, các chức năng và thông tin còn khá đơn giản, một số tính năng còn chưa thực sự tối ưu. Ngoài ra, nếu đem triển khai thực tế sẽ còn phát sinh thêm các lỗi và rủi ro trong quá trình triển khai.

Trong quá trình thực hiện đồ án, em đã phát huy hết mình tinh thần trách nhiệm, nghiêm túc của một người làm sản phẩm. Cùng với sự hướng dẫn tận tình của TS. Trịnh Tuấn Đạt, em đã tìm hiểu và áp dụng được rất nhiều công nghệ, mô hình mới mà trước đó chưa từng sử dụng qua.

Bên cạnh đó, việc xây dựng một ứng dụng từ đầu đến cuối, trải qua tất cả các quy trình giúp em có một cái nhìn thực tế hơn về phát triển phần mềm, những kiến thức này rất bổ ích và giúp em tự tin hơn trong môi trường công việc sau này.

## 6.2 Hướng phát triển

Hệ thống hiện tại chỉ dừng lại ở mức tư tưởng, vì vậy các chức năng và thông tin của ứng dụng còn rất đơn giản. Trong tương lai, cần phải làm giàu dữ liệu của ứng dụng. Việc này rất nhiều trang web tồn tại đang sử dụng, crawl các nguồn dữ liệu từ các website lớn như chợ tốt, batdongsan.com.vn, nhattrong.com... để bổ sung cho ứng dụng của mình.

Trên thực tế, cần chia sẻ rất nhiều thông tin của một ngôi nhà trọ hay cụ thể là cần nhiều ảnh về nhà trọ, ở đây hệ thống mới chỉ đáp ứng cho thêm duy nhất một ảnh. Vì vậy, để hoàn thiện hơn nữa trong tương lai cần để người dùng có thể tải lên nhiều ảnh cho một ngôi nhà.

Với chức năng thanh toán phí đăng bài, trong ứng dụng của mình em mới đáp ứng hai cách thanh toán qua Zalopay và Momo. Tuy nhiên trên thực tế, rất nhiều người lựa chọn thanh toán bằng các tài khoản ngân hàng đã có. Vì chỉ dừng lại ở việc triển khai local nên các chức

năng này em chưa tích hợp được vào hệ thống. Khi triển khai thực tế, cần bổ sung thêm việc thanh toán bằng các tài khoản ngân hàng.

Với chức năng đăng nhập, muốn đăng nhập trước hết người dùng cần đăng kí tài khoản. Quy trình này có thể mất thời gian và rất nhiều người dùng có thể chuyển sang sử dụng một ứng dụng khác ngay khi cảm thấy khó chịu. Cần tích hợp thêm các phương thức đăng nhập bằng các mạng xã hội như Facebook, Zalo, Github...

Ngoài ra, với nhu cầu đặt ra từ đầu, em muốn hướng tới ứng dụng Web Application của mình có thể chạy trên nhiều nền tảng nên để đáp ứng, trong tương lai cần phát triển giao diện theo hướng đa nền tảng (responsive) để người dùng có thể sử dụng được cả trên điện thoại hay tablet, tăng tính tiện dụng.

## 10. Tài liệu tham khảo

[1] Google Developers, Google, <https://developers.google.com/maps/documentation> truy cập lần cuối 29/7/2022.

[2] Microsoft, <https://docs.microsoft.com/vi-vn/aspnet/core> truy cập lần cuối 29/7/2022.