

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



LÊ VĂN CƯỜNG - 52100171

LÊ MINH HOÀNG - 52200036

WEB QUẢN LÝ BẤT ĐỘNG SẢN

**BÁO CÁO CUỐI KÌ
NHẬP MÔN
CÔNG NGHỆ JAVA**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**LÊ VĂN CƯỜNG - 52100171
HỌ TÊN - MSSV**

WEB QUẢN LÝ BẤT ĐỘNG SẢN

**BÁO CÁO CUỐI KÌ
NHẬP MÔN
CÔNG NGHỆ JAVA**

**Người hướng dẫn
TS. VÕ VĂN THÀNH**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024

LỜI CẢM ƠN

Trong suốt quá trình học tập và rèn luyện, chúng em đã nhận được rất nhiều sự giúp đỡ tận tình, sự quan tâm, chăm sóc của thầy **Võ Văn Thành**. Ngoài ra, chúng em còn được thầy truyền đạt những kiến thức, phương pháp mới về toán hay ho và thú vị, thầy còn giúp sinh viên có được nhiều niềm vui trong việc học và cảm thấy thoải mái, ... Chúng em xin chân thành cảm ơn thầy rất nhiều trong suốt quá trình học tập này!

Bởi lượng kiến thức của chúng em còn hạn hẹp và gặp nhiều vấn đề trong quá trình học nên báo cáo này sẽ còn nhiều thiếu sót và cần được học hỏi thêm. Chúng em rất mong sẽ nhận được sự góp ý của thầy về bài báo cáo này để chúng em rút kinh nghiệm trong những môn học sắp tới. Cuối cùng, chúng em xin chân thành cảm ơn thầy .

TP. Hồ Chí Minh, ngày 26 tháng 11 năm 2024.

Tác giả

(Ký tên và ghi rõ họ tên)

Lê Văn Cường

Lê Minh Hoàng

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng tôi xin cam đoan đây là công trình nghiên cứu của riêng chúng tôi và được sự hướng dẫn khoa học của thầy Võ Văn Thành. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 26 tháng 11 năm 2024.

Tác giả

(Ký tên và ghi rõ họ tên)

Lê Văn Cường

Lê Minh Hoàng

WEB QUẢN LÝ BẤT ĐỘNG SẢN

TÓM TẮT

Đề tài của chúng ta xoay quanh việc phát triển một hệ thống quản lý bất động sản trực tuyến. Hệ thống này sẽ cung cấp các dịch vụ như niêm yết bất động sản, giao dịch mua bán và cho thuê, thanh toán an toàn, và quản lý bảo mật. Người dùng sẽ có khả năng xem và tìm kiếm các bất động sản, liên hệ với nhà môi giới, thực hiện giao dịch, và quản lý tài khoản của họ. Hệ thống sẽ bao gồm các tính năng như xác thực người dùng, quản lý danh sách bất động sản, xử lý giao dịch, và quản lý thanh toán. Mục tiêu của dự án là cung cấp một nền tảng thuận tiện và an toàn cho người dùng để tìm kiếm, mua bán và cho thuê bất động sản một cách hiệu quả.

MỤC LỤC

LỜI CẢM ƠN	i
TÓM TẮT iii	
DANH MỤC HÌNH VẼ	vii
DANH MỤC BẢNG BIỂU	ix
DANH MỤC CÁC CHỮ VIẾT TẮT.....	x
CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI.....	1
1.1 Lý do chọn đề tài web bất động sản với Spring Boot	1
1.2 Các Nguyên Tắc, Mô Hình Phát Triển Phần Mềm và Thực Tiễn Đang Được Áp Dụng	1
1.2.1 Nguyên tắc SOLID	1
1.2.2 Mô hình MVC (Model-View-Controller)	2
1.2.3 Security (Spring Security)	2
1.2.4 Spring Boot	3
1.2.5 Thymleaf.....	3
1.2.6 Maven.....	3
1.2.7 Lombok.....	3
1.2.8 Mapstruct	3
1.3 Cấu trúc SpringBoot trong dự án	4
CHƯƠNG 2. PHÂN TÍCH HỆ THỐNG	6
2.1 Tác nhân hệ thống	6
2.2 Các chức năng của hệ thống.....	6
2.2.1 Người quản trị.....	7

2.2.2 Người mua giới	7
2.2.3 Người mua, khách hàng	8
2.3 Các yêu cầu phi chức năng.....	9
CHƯƠNG 3. Thiết kế hệ thống.....	9
3.1 UseCase tổng quát.....	9
3.2 Mô tả UseCase	10
3.2.1 UseCase đăng nhập	10
3.2.2 UseCase đăng ký.....	12
3.2.3 UseCase đổi mật khẩu	14
3.2.4 Usecase xem danh sách bất động sản.....	15
3.2.5 Usecase thêm bất động sản.....	17
3.2.6 Usecase xóa bất động sản.....	18
3.2.7 Usecase liên hệ người dùng	20
3.2.8 Usecase xử lý thuê/mua bất động sản.....	22
3.2.9 Usecase quản lý giao dịch	24
3.2.10 Usecase xử lý giao dịch	25
3.2.11 Usecase tìm kiếm bất động sản.....	27
3.2.12 Usecase xem chi tiết bất động sản	29
3.2.13 Usecase thuê/mua bất động sản.....	31
3.2.14 Usecase thanh toán.....	33
3.3 Class diagrams	34
3.3.1 Mô tả	34
3.3.2 Sơ đồ class diagram.....	35

3.4 Sequence Diagram	36
3.4.1 Đăng ký cung cấp dịch vụ.....	36
3.4.2 Đăng ký dịch vụ	36
3.4.3 Hủy dịch vụ	37
3.4.4 Xem danh sách dịch vụ	37
3.4.5 Xem dịch vụ đang đăng ký	38
3.5 Sơ đồ ERD	38
CHƯƠNG 4. THỰC HIỆN HỆ THỐNG.....	39
4.1 Trang Admin	39
4.2 Trang Agent	42
4.3 Trang Customer.....	45
CHƯƠNG 5. THUẬN LỢI VÀ KHÓ KHĂN	51
5.1 Thuận lợi	51
5.2 Khó khăn	51
CHƯƠNG 6. PHÂN CÔNG NHIỆM VỤ	52
CHƯƠNG 7. TÀI LIỆU THAM KHẢO.....	53

DANH MỤC HÌNH VẼ

Hình 1.1: Cấu trúc các Layer trong SpringBoot	6
Hình 3.1: Usecase tổng quát của hệ thống	10
Hình 3.2: Sơ đồ class diagram	36
Hình 3.3: Đăng ký cung cấp dịch vụ.....	36
Hình 3.4: Đăng ký dịch vụ	37
Hình 3.5: Hủy dịch vụ.....	37
Hình 3.6: Xem danh sách dịch vụ	38
Hình 3.7: Xem dịch vụ đang đăng ký	38
Hình 3.8: Sơ đồ ERD hệ thống	39
Hình 4.1: Trang dashboard của Admin.....	39
Hình 4.2: Trang quản lý người dùng của Admin	40
Hình 4.3: Trang quản lý khách hàng của Admin	40
Hình 4.4: Trang quản lý nhà môi giới của Admin	41
Hình 4.5: Trang quản lý bất động sản của Admin	41
Hình 4.6: Trang xem chi tiết bất động sản của Admin	42
Hình 4.7: Trang quản lý giao dịch của Admin.....	42
Hình 4.8: Trang dashboard của Agent	43
Hình 4.9: Trang tạo bất động sản của Agent.....	43
Hình 4.10: Trang quản lý bất động sản của Agent.....	44
Hình 4.11: Trang cập nhật bất động sản của Agent	44
Hình 4.12: Trang quản lý giao dịch với khách hàng của Agent	45
Hình 4.13: Trang giao diện chính của khách hàng	46

Hình 4.14: Trang giao diện chi tiết của bất động sản	47
Hình 4.15: Trang giao diện khách hàng muốn liên hệ với nhà môi giới	48
Hình 4.16: Trang giao diện giới thiệu các nhà môi giới	49
Hình 4.17: Trang giao diện liên hệ.....	50

DANH MỤC BẢNG BIỂU

Bảng 3.1: UseCase đăng nhập.....	12
Bảng 3.2: UseCase đăng ký	14
Bảng 3.3: Usecase đổi mật khẩu	15
Bảng 3.4: Usecase xem danh sách bất động sản	17
Bảng 3.5: Usecase thêm bất động sản.....	18
Bảng 3.6: Usecase xóa bất động sản	20
Bảng 3.7: Usecase liên hệ người dùng.....	22
Bảng 3.8: Usecase xử lý thuê/mua bất động sản.....	24
Bảng 3.9: Usecase quản lý giao dịch	25
Bảng 3.10: Usecase xử lý giao dịch	27
Bảng 3.11: Usecase tìm kiếm bất động sản	29
Bảng 3.12: Usecase xem chi tiết bất động sản	31
Bảng 3.13: Usecase thuê/mua bất động sản	32
Bảng 3.14: Usecase thanh toán	33

DANH MỤC CÁC CHỮ VIẾT TẮT

FE	Font End
BE	Back End

CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI

1.1 Lý do chọn đề tài web bất động sản với Spring Boot

- **Nhu cầu thị trường:** Bất động sản là một ngành quan trọng, có ảnh hưởng lớn đến nền kinh tế và cuộc sống con người. Với sự phát triển mạnh mẽ của công nghệ, các dịch vụ bất động sản trực tuyến ngày càng được ưa chuộng. Một hệ thống web bất động sản sẽ giúp người dùng dễ dàng tìm kiếm, đăng tin, giao dịch và quản lý các bất động sản.

- **Spring Boot và tính linh hoạt:** Spring Boot là một framework mạnh mẽ và dễ sử dụng trong phát triển ứng dụng web. Nó hỗ trợ phát triển nhanh chóng các ứng dụng với cấu trúc rõ ràng và mở rộng dễ dàng, phù hợp với các yêu cầu của một ứng dụng bất động sản, từ việc đăng tin, tìm kiếm, đến quản lý giao dịch.

- **Quản lý dữ liệu hiệu quả:** Trong ngành bất động sản, việc quản lý và xử lý dữ liệu như thông tin về bất động sản, người dùng, giao dịch, v.v. rất quan trọng. Spring Boot cùng với các công nghệ như JPA (Java Persistence API) giúp xây dựng một hệ thống quản lý dữ liệu mạnh mẽ, an toàn và dễ dàng mở rộng.

- **Tính bảo mật cao:** Spring Security giúp bảo vệ ứng dụng web khỏi các nguy cơ bảo mật, từ việc đăng nhập của người dùng, phân quyền đến bảo vệ các dữ liệu nhạy cảm như thông tin cá nhân và giao dịch bất động sản.

- **Khả năng mở rộng và bảo trì:** Với Spring Boot, việc phát triển và bảo trì hệ thống trở nên dễ dàng hơn nhờ vào các tính năng như tự động cấu hình, tích hợp với các công nghệ mới, và khả năng mở rộng linh hoạt.

- **Tạo cơ hội nghề nghiệp:** Việc phát triển ứng dụng bất động sản sử dụng Spring Boot cũng giúp người phát triển nâng cao kỹ năng lập trình Java, Spring Framework, và các công nghệ web hiện đại, mở ra cơ hội nghề nghiệp trong ngành công nghệ thông tin và phát triển phần mềm.

1.2 Các Nguyên Tắc, Mô Hình Phát Triển Phần Mềm và Thực Tiễn Đang Được Áp Dụng

1.2.1 Nguyên tắc SOLID

Đảm bảo thiết kế các lớp và phương thức tuân thủ:

- **S:** Single Responsibility Principle - Mỗi lớp chỉ chịu trách nhiệm một nhiệm vụ duy nhất.
- **O:** Open/Closed Principle - Các thành phần có thể mở rộng mà không thay đổi mã hiện có.
- **L:** Liskov Substitution Principle - Các lớp dẫn xuất có thể thay thế lớp cơ sở mà không gây lỗi.
- **I:** Interface Segregation Principle - Các giao diện được tách biệt để tránh các phụ thuộc không cần thiết.
- **D:** Dependency Inversion Principle - Ưu tiên phụ thuộc vào các abstractions thay vì các chi tiết cụ thể.

1.2.2 Mô hình MVC (Model-View-Controller)

Model:

- Quản lý dữ liệu và logic xử lý dữ liệu.
- Ví dụ: Các lớp như Property, User, Transaction, ... để xử lý đất đai, người dùng, giao dịch,...

View:

- Đại diện cho giao diện người dùng.
- Ví dụ: Trang web hiển thị danh sách sản phẩm, thông tin.

Controller:

- Điều phối tương tác giữa Model và View.
- Ví dụ: Các lớp điều khiển như PropertyController và UserController.

1.2.3 Security (Spring Security)

Sử dụng Spring Security để:

- Quản lý xác thực (Authentication).
- Quản lý ủy quyền (Authorization).
- Bảo vệ các endpoint API của ứng dụng.

1.2.4 Spring Boot

Cung cấp cấu hình mặc định giúp khởi chạy ứng dụng nhanh chóng.

Hỗ trợ phát triển các ứng dụng phức tạp với cấu trúc đơn giản hơn.

1.2.5 Thymleaf

Công cụ tạo giao diện web với khả năng tích hợp liền mạch với Spring Boot.

Hỗ trợ tùy chỉnh giao diện cho các trang hiển thị sản phẩm và giao dịch.

1.2.6 Maven

Quản lý các thư viện phụ thuộc (dependencies).

Hỗ trợ quy trình xây dựng và triển khai dự án dễ dàng hơn.

1.2.7 Lombok

Sử dụng để:

- Tự động sinh các phương thức getter, setter, và constructor.
- Làm giảm độ phức tạp của mã nguồn.
- Đơn giản hóa các lớp POJO (Plain Old Java Object).

1.2.8 Mapstruct

- Các tính năng chính của MapStruct

- Chuyển đổi giữa các lớp (DTOs, Entities)
- Hiệu suất cao
- Hỗ trợ cấu hình tùy chỉnh
- Tích hợp tốt với các framework khác
- Giao diện dựa trên annotation

1.3 Cấu trúc SpringBoot trong dự án

Trong dự án này cấu trúc bao gồm các layer sau:

Configuration: Lớp cấu hình dùng để thiết lập các cấu hình chung cho ứng dụng, chẳng hạn như cấu hình bảo mật, cấu hình dữ liệu, và các cài đặt khác cần thiết cho ứng dụng. Các lớp cấu hình có thể bao gồm cấu hình kết nối cơ sở dữ liệu, cấu hình bảo mật Spring Security, cấu hình API, và các cấu hình cụ thể của ứng dụng.

Controller: Lớp Controller chịu trách nhiệm nhận và xử lý các yêu cầu HTTP từ client. Controller tiếp nhận các yêu cầu, xử lý chúng (có thể thông qua các dịch vụ), và trả về các kết quả phù hợp cho người dùng. Controller cũng có thể thực hiện việc xác thực và phân quyền cho các yêu cầu dựa trên các cấu hình bảo mật của ứng dụng.

Dto: DTO là các đối tượng dữ liệu dùng để truyền tải thông tin giữa các lớp của ứng dụng, đặc biệt là giữa lớp Controller và lớp Service. DTO giúp tách biệt dữ liệu truyền tải ra khỏi các đối tượng Entity (thường được sử dụng để tương tác với cơ sở dữ liệu). DTO bao gồm hai loại:

- **Request:** Dùng để nhận dữ liệu từ người dùng khi họ gửi yêu cầu đến hệ thống (ví dụ: thông tin người dùng, sản phẩm, v.v.).
- **Response:** Dùng để gửi lại dữ liệu cho người dùng, chứa các kết quả xử lý từ hệ thống.

Entity: Entity đại diện cho các đối tượng trong cơ sở dữ liệu. Đây là các lớp được ánh xạ trực tiếp với các bảng trong cơ sở dữ liệu thông qua JPA (Java Persistence API). Các Entity giúp lưu trữ và truy vấn dữ liệu từ cơ sở dữ liệu một cách hiệu quả.

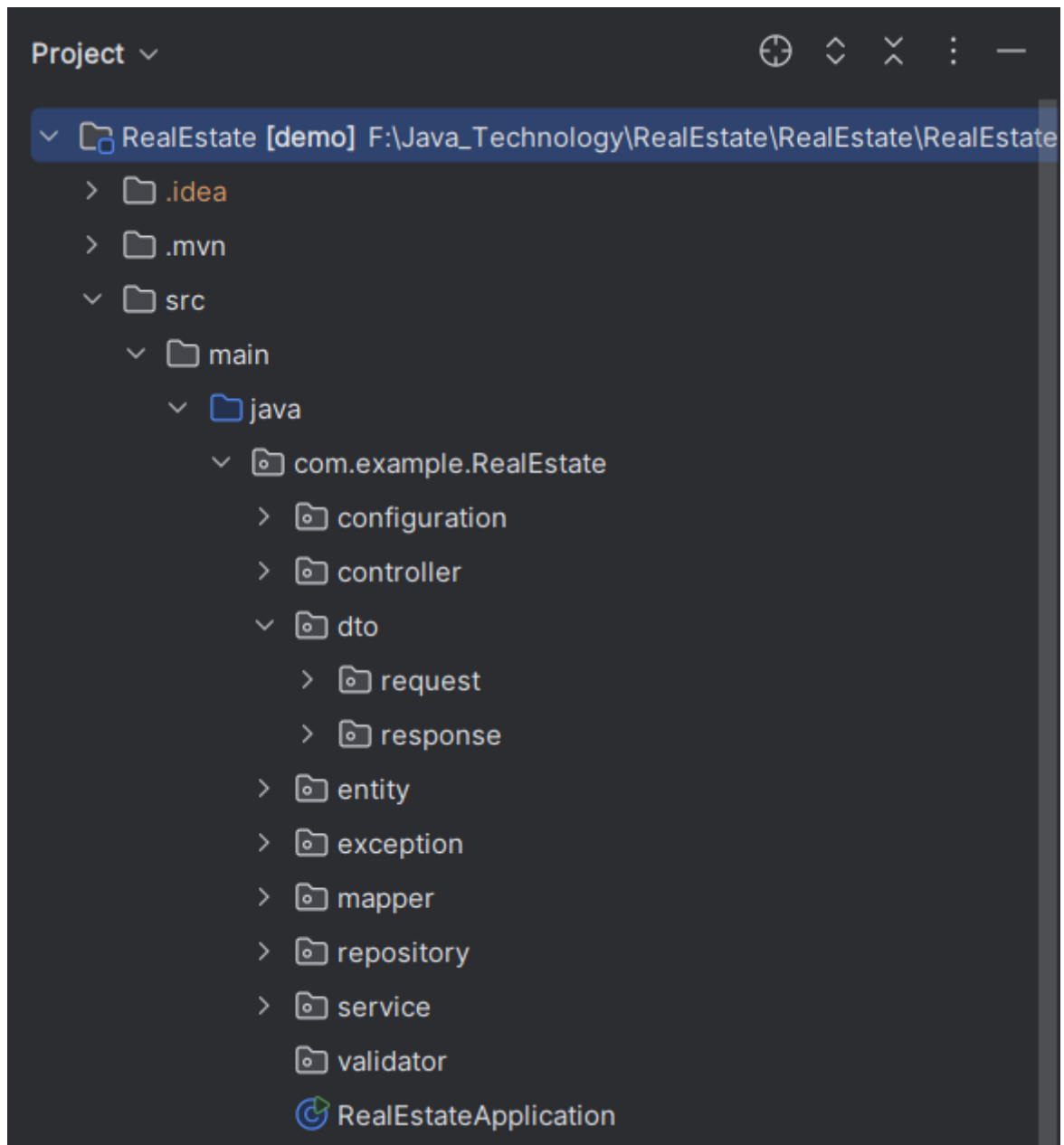
Exception: Các lớp exception chịu trách nhiệm xử lý các ngoại lệ trong quá trình xử lý yêu cầu. Các exception này có thể được sử dụng để báo cáo lỗi khi có vấn đề trong quá trình xử lý (ví dụ: tài nguyên không tìm thấy, lỗi đầu vào từ người dùng). Mỗi loại ngoại lệ có thể được ánh xạ với mã lỗi HTTP phù hợp để người dùng có thể nhận được thông báo lỗi chính xác.

Mapper: Mapper dùng để chuyển đổi giữa các đối tượng khác nhau, ví dụ như chuyển từ Entity sang DTO và ngược lại. Lớp Mapper giúp việc chuyển đổi dữ liệu giữa các lớp trở nên tự động và dễ dàng hơn, giảm thiểu việc viết mã thủ công.

Repository: Repository là lớp trung gian giữa ứng dụng và cơ sở dữ liệu, giúp thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên các Entity. Repository thường sử dụng Spring Data JPA hoặc các framework tương tự để thực hiện các truy vấn cơ sở dữ liệu mà không cần phải viết nhiều SQL thủ công.

Service: Service chứa các logic nghiệp vụ của ứng dụng, bao gồm xử lý các yêu cầu từ Controller, gọi các phương thức của Repository để thao tác với dữ liệu, và thực hiện các logic nghiệp vụ phức tạp. Service là nơi xử lý các yêu cầu và đảm bảo dữ liệu được xử lý chính xác trước khi trả về kết quả cho người dùng.

Validator: Validator là lớp kiểm tra tính hợp lệ của dữ liệu trước khi đưa vào xử lý trong các lớp Service. Đây là nơi thực hiện các kiểm tra như kiểm tra dữ liệu đầu vào của người dùng (ví dụ: kiểm tra xem trường dữ liệu có bị bỏ trống, kiểm tra định dạng email, v.v.). Nếu dữ liệu không hợp lệ, Validator sẽ thông báo lỗi hoặc ném ra một exception.



Hình 1.1: Cấu trúc các Layer trong SpringBoot

CHƯƠNG 2. PHÂN TÍCH HỆ THỐNG

2.1 Tác nhân hệ thống

- Người quản trị (Admin)
- Người mua giới (Agent)
- Người mua, khách hàng (Customer)

2.2 Các chức năng của hệ thống

2.2.1 Người quản trị

Người quản trị là người có quyền cao nhất trong hệ thống, chịu trách nhiệm quản lý toàn bộ các chức năng liên quan đến người dùng và nội dung trên website. Các chức năng của người quản trị bao gồm:

1. Quản lý tài khoản người dùng:

- Duyệt và phê duyệt tài khoản của người mới giới khi họ đăng ký.
- Cấp, chỉnh sửa, hoặc thu hồi quyền truy cập của người dùng.
- Xóa tài khoản người dùng nếu vi phạm chính sách.

2. Quản lý bài đăng bất động sản:

- Phê duyệt bài đăng bất động sản trước khi hiển thị lên website.
- Xóa hoặc chỉnh sửa bài đăng không phù hợp.
- Xem và quản lý lịch sử các bài đăng.

3. Quản lý danh mục:

- Thêm, chỉnh sửa hoặc xóa các danh mục bất động sản (ví dụ: nhà phố, chung cư, đất nền).
- Quản lý danh sách khu vực, tỉnh/thành phố.

4. Quản lý giao dịch:

- Theo dõi và phê duyệt các giao dịch giữa người mua và người bán.
- Xem báo cáo doanh thu và lịch sử giao dịch.

5. Quản lý thông tin hệ thống:

- Quản lý các thông tin liên hệ, chính sách bảo mật, và điều khoản sử dụng của website.
- Xem báo cáo thống kê số liệu truy cập, hoạt động người dùng và hiệu quả kinh doanh.

2.2.2 Người mua giới

Người môi giới là những cá nhân hoặc tổ chức có vai trò đăng bài quảng cáo bất động sản để kết nối với khách hàng. Các chức năng của người môi giới bao gồm:

1. Quản lý tài khoản cá nhân:

- Đăng ký tài khoản và chờ phê duyệt từ quản trị viên.

- Chỉnh sửa thông tin cá nhân, hồ sơ chuyên nghiệp.

2. Đăng bài bất động sản:

- Đăng bài quảng cáo bất động sản kèm thông tin chi tiết (hình ảnh, giá cả, diện tích, vị trí).
- Chỉnh sửa hoặc xóa bài đăng bất động sản.
- Theo dõi trạng thái phê duyệt của bài đăng.

3. Quản lý danh sách bất động sản:

- Xem danh sách các bất động sản đã đăng.
- Theo dõi số lượt lượt xem, liên hệ từ khách hàng.

4. Tương tác với khách hàng:

- Trả lời câu hỏi hoặc yêu cầu từ khách hàng liên quan đến bất động sản.
- Cập nhật trạng thái bất động sản (đã bán, còn trống, đang thương lượng).

5. Thống kê hiệu quả hoạt động:

- Xem báo cáo về lượt xem bài đăng, số lượt liên hệ.
- Theo dõi lịch sử giao dịch liên quan đến các bất động sản đã đăng.

2.2.3 Người mua, khách hàng

Người mua hoặc khách hàng là những người tìm kiếm và mua bất động sản trên website. Các chức năng của họ bao gồm:

1. Tìm kiếm bất động sản:

- Tìm kiếm bất động sản theo các tiêu chí: loại hình, giá cả, vị trí, diện tích.
- Lọc và sắp xếp kết quả tìm kiếm theo các yếu tố như giá tăng/giảm, diện tích.

2. Xem thông tin chi tiết:

- Xem chi tiết thông tin bất động sản, bao gồm hình ảnh, mô tả, giá bán, và thông tin người môi giới.

3. Liên hệ người môi giới:

- Gửi tin nhắn hoặc yêu cầu liên hệ với người môi giới thông qua hệ thống.
- Lưu thông tin bất động sản yêu thích vào danh sách riêng.

4. **Quản lý tài khoản cá nhân:**

- Đăng ký và đăng nhập tài khoản để theo dõi lịch sử hoạt động.
- Chỉnh sửa thông tin cá nhân, thay đổi mật khẩu.

5. **Tham gia giao dịch:**

- Gửi yêu cầu mua bất động sản thông qua hệ thống.
- Theo dõi tiến trình giao dịch, bao gồm các bước thương lượng, ký hợp đồng, và thanh toán.

6. **Đánh giá và phản hồi:**

- Đánh giá chất lượng dịch vụ của người môi giới.
- Gửi phản hồi hoặc khiếu nại đến quản trị viên nếu gặp vấn đề.

2.3 Các yêu cầu phi chức năng

Tích hợp Thanh toán: Cung cấp khả năng thanh toán trực tuyến an toàn và thuận tiện cho người dùng khi họ muốn mua hoặc thuê bất động sản.

Tích hợp Bản đồ: Hiện thị bản đồ vị trí của các bất động sản để người dùng có thể dễ dàng xác định vị trí chính xác.

Hỗ trợ Tìm kiếm nâng cao: Cho phép người dùng tìm kiếm bất động sản theo các tiêu chí cụ thể như diện tích, giá, loại hình, vv.

CHƯƠNG 3. Thiết kế hệ thống

3.1 UseCase tổng quát

3.2 Mô tả UseCase

3.2.1 UseCase đăng nhập

Use Case Name	Đăng nhập
Triggering Event	Người dùng muốn truy cập vào hệ thống để sử dụng các tính năng của trang web.

Description	Chức năng này cho phép người dùng nhập thông tin đăng nhập của họ để truy cập vào tài khoản cá nhân trên trang web.	
Actor(s)	Người dùng	
Pre-Condition(s)	Trang web hiển thị giao diện đăng nhập. Người dùng đã đăng ký tài khoản trên trang web.	
Post-Condition(s)	Người dùng đã đăng nhập thành công và được chuyển hướng đến trang chính của tài khoản cá nhân. Hoặc thông báo lỗi nếu thông tin đăng nhập không chính xác.	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người dùng: Nhấn vào nút "Đăng nhập" trên trang web. 2. Người dùng: Nhập tên đăng nhập và mật khẩu vào các trường tương ứng 	<ol style="list-style-type: none"> 1. Hệ thống: Hiển thị giao diện đăng nhập yêu cầu người dùng nhập tên đăng nhập và mật khẩu. 2. Hệ thống: Chấp nhận thông tin đăng nhập và kiểm tra tính hợp lệ của chúng.

	<p>trên giao diện đăng nhập.</p> <p>3. Người dùng: Nhấn nút "Đăng nhập".</p> <p>4. Người dùng: Truy cập vào trang chính của tài khoản cá nhân.</p>	<p>3. Hệ thống: Xác nhận tính hợp lệ của thông tin đăng nhập.</p> <p>4. Hệ thống: Nếu thông tin đăng nhập chính xác, đăng nhập người dùng vào hệ thống và chuyển hướng họ đến trang chính của tài khoản cá nhân.</p>
Exception	<p>Nếu hệ thống gặp sự cố kỹ thuật hoặc lỗi, nó sẽ hiển thị thông báo lỗi và yêu cầu người dùng thử lại sau.</p> <p>Nếu người dùng quên mật khẩu, họ có thể sử dụng chức năng "Quên mật khẩu" để đặt lại mật khẩu mới.</p>	

Bảng 3.1: UseCase đăng nhập

3.2.2 UseCase đăng ký

Use Case Name	Đăng kí
Triggering Event	Người dùng muốn tạo một tài khoản mới trên trang web.
Description	Chức năng này cho phép người dùng tạo một tài khoản mới bằng cách cung cấp thông tin cá nhân cần thiết.

Actor(s)	Người dùng	
Pre-Condition(s)	Trang web hiển thị giao diện đăng nhập. Người dùng đã đăng ký tài khoản trên trang web.	
Post-Condition(s)	Người dùng đã tạo thành công một tài khoản mới và có thể sử dụng để đăng nhập vào hệ thống.	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người dùng: Truy cập vào trang web và nhấp vào liên kết "Đăng ký". 2. Người dùng: Nhập thông tin cá nhân cần thiết vào các trường trên giao diện đăng ký. 3. Người dùng: Nhấn nút "Đăng ký".chính của tài khoản cá nhân. 	<ol style="list-style-type: none"> 1. Hệ thống: Hiển thị giao diện đăng ký yêu cầu người dùng nhập thông tin cá nhân cần thiết để tạo tài khoản mới. 2. Hệ thống: Chấp nhận thông tin đăng ký và kiểm tra tính hợp lệ của chúng. 3. Hệ thống: Tạo một tài khoản mới nếu thông tin đăng ký hợp lệ và thông

		báo cho người dùng biết.
Exception	Nếu thông tin đăng ký không hợp lệ (ví dụ: tên đăng nhập đã tồn tại), hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin đăng ký.	

Bảng 3.2: UseCase đăng ký

3.2.3 UseCase đổi mật khẩu

Use Case Name	Đổi mật khẩu
Triggering Event	Người dùng click chức năng “đổi mật khẩu” trong trang web.
Description	Người dùng muốn thay đổi mật khẩu cho tài khoản của mình.
Actor(s)	Người dùng
Pre-Condition(s)	Đã đăng nhập vào trang web. Thiết bị của người dùng đã được kết nối internet khi thực hiện đổi mật khẩu.

Post-Condition(s)	Tài khoản của người dùng đã được đổi mật khẩu. Hiển thị thông báo đổi mật khẩu thành công.	
Flow of Event	Actor	System
	1 Người dùng truy cập ứng dụng. 2 Người dùng chọn chức năng đổi mật khẩu. 3 Nhập các thông tin cần thiết: + Mật khẩu cũ. + Mật khẩu mới. + Xác nhận mật khẩu. 4 Chọn nút lưu.	1. Hiển thị giao diện người dùng. 2. Hiển thị form thay đổi mật khẩu. 3.1 Hệ thống kiểm tra thông tin đổi mật khẩu thành công và cho phép người dùng truy cập ứng dụng. 3.2 Hiển thị thanh tác thành công. 4.1 Hệ thống sẽ cập nhật lại mật khẩu của tài khoản.
Exception	Hệ thống xác thực đổi mật khẩu không thành công và hiển thị thông báo. Người dùng không nhấn lưu đổi mật khẩu thì hệ thống sẽ giữ nguyên hiện trạng.	

Bảng 3.3: Usecase đổi mật khẩu

3.2.4 Usecase xem danh sách bất động sản

Use Case Name	Xem toàn bộ danh sách	
Triggering Event	Người dùng muốn xem tất cả bất động sản trong trang web.	
Description	Người dùng chọn nút xem tất cả để xem.	
Actor(s)	Người dùng.	
Pre-Condition(s)	Thiết bị của người dùng đã được kết nối internet khi thực hiện tìm kiếm.	
Post-Condition(s)	Người dùng xem tất cả sản phẩm thành công.	
Flow of Event	Actor	System
	1 Người dùng truy cập trang web. 2 Người dùng chọn nút xem tất cả.	1. Hiện thị giao diện trang web. 2.1 Hệ thống kiểm tra lấy danh sách từ cơ sở dữ liệu. 2.2 Hệ thống hiển thị.

Exception	Không có sản phẩm để hiển thị.	

Bảng 3.4: Usecase xem danh sách bất động sản

3.2.5 Usecase thêm bất động sản

Use Case	Thêm bất động sản
Triggering Event	Quản trị viên nhập thông tin của bất động sản và bấm nút thêm
Description	Người mua giới đăng nhập vào trang. Sau đó chọn post, nhập thông tin của bất động sản sau đó bấm thêm.
Actor(s)	Người mua giới
Pre-Condition(s)	Thiết bị đã được kết nối internet. Đã có tài khoản.

Post-Condition(s)	Bất động sản sẽ được thêm vào danh sách	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người mua giới truy cập và đăng nhập vào hệ thống 2. Người mua giới nhấn vào Quản lý sản phẩm. 3. Người mua giới chọn Add. Người mua giới nhập đầy đủ thông tin sản phẩm rồi bấm thêm. 	<ol style="list-style-type: none"> 1. Hiển thị giao diện admin. 2. Hiển thị giao diện Quản lý bất động sản. 3. Hiển thị giao diện danh sách bất động sản có trong cửa hàng 4. Hiển thị giao diện nhập thông tin sách bất động sản 5. Nếu không nhập đầy đủ sẽ không thể sách bất động sản và hệ thống sẽ báo lỗi. 6. Bất động sản được thêm vào và xem lại trong danh sách
Exception	Người mua giới không nhập đầy đủ thông tin khi thêm sản phẩm	

Bảng 3.5: Usecase thêm bất động sản

3.2.6 Usecase xóa bất động sản

Use Case Name	Xóa bất động sản	
Triggering Event	Người mua giới chọn bất động muốn xóa và click “Xóa”	
Description	<p>Khi không còn bán sản phẩm đó nữa, Người mua giới cần xóa đó ra khỏi ứng dụng</p> <p>Hệ thống đánh giá sản phẩm đã được bán thì sẽ xóa.</p>	
Actor(s)	Người mua giới, hệ thống	
Pre-Condition(s)	<p>Thiết bị đã được kết nối internet.</p> <p>Đã có tài khoản.</p>	
Post-Condition(s)	Bất động sản sẽ xóa khỏi cửa hàng	
Flow of Event	Actor	System
	1 Người mua giới truy cập và đăng nhập vào hệ thống	<p>1. Hiện thị giao diện .</p> <p>2. Hiện thị giao diện Quản lý</p> <p>3. Hiện thị giao diện danh sách</p>

	<p>2 Người mua giới nhân vào Quản lý sản phẩm</p> <p>3 Người mua giới chọn delete tại sản phẩm cần xóa.</p>	4. Hệ thống xóa bất động sản đã được chọn ra khỏi danh sách post
Exception	Người mua chọn bất động sản không có trong danh sách của họ post	

Bảng 3.6: Usecase xóa bất động sản

3.2.7 Usecase liên hệ người dùng

Use Case Name	Liên Hệ giữa người dùng và nhà môi giới
Triggering Event	Người dùng hoặc nhà môi giới muốn liên hệ với đối tác của mình để thảo luận về các giao dịch bất động sản.
Description	Chức năng này cho phép người dùng và nhà môi giới có thể liên hệ với nhau để thảo luận về các giao dịch bất động sản, bao gồm việc xem và mua bán bất động sản.
Actor(s)	Người dùng, Nhà môi giới
Pre-Condition(s)	Người dùng hoặc nhà môi giới đã đăng nhập vào tài khoản của mình trên trang web bất động sản.

	Cả hai đối tác (người dùng và nhà môi giới) đều có thông tin liên hệ hiển thị trên hệ thống.	
Post-Condition(s)	Sau khi liên hệ, cả người dùng và nhà môi giới đều có thể tiếp tục thảo luận và tiến hành các bước tiếp theo trong quá trình giao dịch bất động sản.	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người dùng hoặc nhà môi giới muốn liên hệ với đối tác của mình để thảo luận về các giao dịch bất động sản. 	<ol style="list-style-type: none"> 1. Hệ thống hiển thị tùy chọn "Liên Hệ" trên trang thông tin bất động sản cụ thể. 2. Hệ thống hiển thị giao diện liên hệ cho phép người dùng hoặc nhà môi giới nhập tin nhắn hoặc thông điệp liên hệ. 3. Hệ thống gửi thông điệp liên hệ đến đối tác của mình. 4. Hệ thống cập nhật trạng thái của tin nhắn và thông báo cho người gửi khi có phản hồi từ đối tác.

		5. Hệ thống cung cấp khả năng trò chuyện trực tiếp trên trang web giữa hai bên nếu cần.
Exception	Nếu thông tin liên hệ không chính xác hoặc không đầy đủ, hệ thống có thể hiển thị thông báo lỗi và yêu cầu người dùng hoặc nhà môi giới cập nhật thông tin liên hệ của họ.	

Bảng 3.7: Usecase liên hệ người dùng

3.2.8 Usecase xử lý thuê/mua bất động sản

Use Case Name	Xử lý yêu cầu thuê/mua
Triggering Event	Nhà môi giới nhận yêu cầu thuê/mua từ người dùng.
Description	Chức năng này cho phép nhà môi giới xử lý các yêu cầu thuê/mua bất động sản từ người dùng, bao gồm việc thực hiện các bước giao dịch như đàm phán, ký hợp đồng và thanh toán.
Actor(s)	Nhà môi giới

Pre-Condition(s)	Nhà môi giới đã đăng nhập vào hệ thống. Người dùng đã gửi yêu cầu thuê/mua bất động sản.	
Post-Condition(s)	Nhà môi giới đã thực hiện các bước giao dịch và hoàn tất yêu cầu thuê/mua bất động sản từ người dùng.	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Nhận yêu cầu thuê/mua từ người dùng thông qua hệ thống. 2. Tiếp tục đàm phán với người dùng về các điều khoản và điều kiện của giao dịch. 3. Hỗ trợ quá trình ký kết hợp đồng thuê/mua bất động sản. 4. Thực hiện các biện pháp cần thiết để hoàn tất giao dịch. 	<ol style="list-style-type: none"> 1. Nhận yêu cầu thuê/mua từ nhà môi giới thông qua giao diện hệ thống. 2. Cung cấp thông tin liên quan và hỗ trợ đàm phán cho nhà môi giới. 3. Ghi nhận và lưu trữ thông tin về giao dịch và hợp đồng. 4. Cập nhật trạng thái của giao dịch và thông báo cho nhà môi giới khi cần thiết.

Exception	Nếu có sự cố kỹ thuật hoặc trục trặc trong quá trình giao dịch, nhà môi giới có thể thông báo cho người dùng và thực hiện các biện pháp khắc phục.
------------------	--

Bảng 3.8: Usecase xử lý thuê/mua bất động sản

3.2.9 Usecase quản lý giao dịch

Use Case Name	Quản lý giao dịch
Triggering Event	Yêu cầu giao dịch từ nhà mua giới và khách hàng
Description	Chức năng này cho phép nhà môi giới xử lý các yêu cầu thuê/mua bất động sản từ người dùng, bao gồm việc thực hiện các bước giao dịch như đàm phán, ký hợp đồng và thanh toán.
Actor(s)	Người mua, Nhà môi giới, Hệ thống
Pre-Condition(s)	Người mua và nhà môi giới đã đăng nhập vào hệ thống. Các thông tin liên quan đến giao dịch đã được nhập và lưu trữ trong hệ thống.

Post-Condition(s)	Giao dịch được hoàn tất thành công và thông tin liên quan được cập nhật trong hệ thống.	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người mua tiến hành các thao tác liên quan đến giao dịch, bao gồm đàm phán với nhà môi giới, ký kết hợp đồng và thanh toán. 2. Nhà môi giới hỗ trợ người mua trong quá trình giao dịch, bao gồm cung cấp thông tin, đàm phán và hỗ trợ ký kết hợp đồng. 	<ol style="list-style-type: none"> 1. Hệ thống ghi nhận và lưu trữ thông tin liên quan đến giao dịch, bao gồm thông tin về giao dịch, hợp đồng và thanh toán. 2. Hệ thống cập nhật trạng thái của giao dịch và thông báo cho người mua và nhà môi giới khi cần thiết.
Exception	Nếu có sự cố kỹ thuật hoặc trục trặc trong quá trình giao dịch, hệ thống có thể thông báo lỗi và hỗ trợ giải quyết vấn đề.	

Bảng 3.9: Usecase quản lý giao dịch

3.2.10 Usecase xử lý giao dịch

Use Case Name	Xử lý giao dịch bán/thuê	
Triggering Event	Yêu cầu từ khách hàng	
Description	Chức năng này cho phép hệ thống xác nhận yêu cầu mua/thuê bất động sản từ người dùng và liên lạc với người mua để hoàn tất giao dịch bán hàng.	
Actor(s)	Hệ thống	
Pre-Condition(s)	<p>Hệ thống đã nhận được yêu cầu mua bất động sản từ người dùng.</p> <p>Thông tin liên lạc của người mua đã được lưu trữ trong hệ thống.</p>	
Post-Condition(s)	Giao dịch bán hàng được hoàn tất và thông tin liên quan được cập nhật trong hệ thống.	
	Actor	System

Flow of Event	<p>1. Người dùng yêu cầu mua/thuê bất động sản</p>	<p>1. Hệ thống xác nhận yêu cầu mua bất động sản từ người dùng, kiểm tra tính hợp lệ của yêu cầu.</p> <p>2. Sau khi xác nhận, hệ thống liên lạc với người mua bằng cách sử dụng thông tin liên lạc đã được lưu trữ trong hệ thống.</p> <p>3. Hệ thống hỗ trợ và hướng dẫn người mua qua các bước cần thiết để hoàn tất giao dịch bán hàng.</p>
Exception	<p>Nếu thông tin thanh toán không hợp lệ hoặc không đầy đủ, hệ thống có thể hiển thị thông báo lỗi và yêu cầu người dùng cung cấp hoặc cập nhật thông tin thanh toán của họ.</p>	

Bảng 3.10: Usecase xử lý giao dịch

3.2.11 Usecase tìm kiếm bất động sản

Use Case Name:	Tìm kiếm bất động sản
-----------------------	-----------------------

Scenario:	Khách hàng muốn tìm kiếm bất động sản dựa trên các tiêu chí cụ thể.	
Triggering Event:	Khách hàng muốn tìm kiếm một bất động sản cụ thể.	
Brief Description:	Khách hàng có thể tìm kiếm bất động sản dựa trên từ khóa hoặc sử dụng các tiêu chí chi tiết như loại, vị trí, giá cả.	
Actor:	Khách hàng	
Preconditions:	Khách hàng đã truy cập vào hệ thống bất động sản và đã đăng nhập vào tài khoản (nếu yêu cầu).	
Postconditions:	Khách hàng tìm thấy các bất động sản phù hợp với nhu cầu của mình.	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Khách hàng truy cập hệ thống 2. Khách hàng chọn chức năng "Tìm kiếm bất động sản". 	<ol style="list-style-type: none"> 1. Hệ thống hiển thị giao diện trang chủ. 2. Hệ thống hiển thị trang tìm kiếm.

	3. Khách hàng nhập từ khóa hoặc chọn các tiêu chí tìm kiếm chi tiết.	3. Hệ thống hiển thị kết quả tìm kiếm dựa trên các tiêu chí được cung cấp.
Exception	Nếu không có bất động sản nào phù hợp với tiêu chí tìm kiếm: Hệ thống hiển thị thông báo rằng không tìm thấy bất động sản phù hợp.	

Bảng 3.11: Usecase tìm kiếm bất động sản

3.2.12 Usecase xem chi tiết bất động sản

Use Case Name:	Xem chi tiết bất động sản
Scenario:	Khách hàng muốn xem thông tin chi tiết của một bất động sản cụ thể.
Triggering Event:	Khách hàng chọn xem thông tin chi tiết của một bất động sản trên danh sách.
Brief Description:	Khách hàng có thể xem thông tin chi tiết, hình ảnh và mô tả của một bất động sản cụ thể. Họ cũng có thể xem thông tin liên lạc với nhà môi giới.

Actor:	Khách hàng	
Preconditions:	Khách hàng đã xem danh sách bất động sản và đã chọn một bất động sản cụ thể để xem chi tiết.	
Postconditions:	Khách hàng đã hiểu rõ thông tin về bất động sản và có thể liên hệ với nhà môi giới nếu cần.	
Flow of events:	Actor	Flow of events:
	1. Khách hàng truy cập hệ thống 2. Khách hàng chọn một bất động sản từ danh sách để xem chi tiết. Khách hàng có thể xem thông tin liên lạc với nhà môi giới.	4. Hệ thống hiển thị giao diện trang chủ. 5. Hệ thống hiển thị trang tìm kiếm. <ul style="list-style-type: none"> Hệ thống hiển thị kết quả tìm kiếm dựa trên các tiêu chí được cung cấp.
Exception Conditions:	Nếu bất động sản được chọn không tồn tại trong cơ sở dữ liệu: Hệ thống hiển thị thông báo lỗi cho người dùng rằng không tìm thấy thông tin về bất động sản này.	

Bảng 3.12: Usecase xem chi tiết bất động sản

3.2.13 Usecase thuê/mua bất động sản

Use Case Name:	Thuê/mua bất động sản	
Scenario:	Khách hàng muốn thuê hoặc mua một bất động sản sau khi quan tâm.	
Triggering Event:	Khách hàng quyết định thuê hoặc mua một bất động sản sau khi xem chi tiết và liên hệ với nhà môi giới.	
Brief Description:	Khách hàng gửi yêu cầu thuê hoặc mua tới nhà môi giới và sau đó thực hiện các bước giao dịch như đàm phán, ký hợp đồng và thanh toán.	
Actor:	Khách hàng	
Preconditions:	Khách hàng đã xem thông tin chi tiết của một bất động sản, đã liên hệ với nhà môi giới và quyết định thuê hoặc mua.	
Postconditions:	Khách hàng đã bắt đầu quá trình thuê hoặc mua bất động sản và đang tiến hành các bước giao dịch.	
Flow of events:	Actor	System
	<p>3. chọn chức năng "Thuê/mua bất động sản" từ trang chi tiết của bất động sản</p> <p>4. Khách hàng bắt đầu điền thông tin cần thiết và gửi yêu cầu.</p>	<p>1.1 Hệ thống hiển thị biểu mẫu hoặc cung cấp phương tiện để người mua gửi yêu cầu thuê/mua tới nhà môi giới.</p> <p>2.1 Hệ thống hiển thị trang để điền thông tin cần thiết và gửi yêu cầu.</p>
Exception Conditions:	<p>Nếu bất động sản đã được thuê hoặc mua bởi một khách hàng khác trước đó:</p> <p>Hệ thống thông báo cho khách hàng rằng bất động sản này đã không còn có sẵn.</p>	

Bảng 3.13: Usecase thuê/mua bất động sản

3.2.14 Usecase thanh toán

Use Case Name:	Thực hiện thanh toán	
Scenario:	Khách hàng muốn thực hiện thanh toán cho bất động sản đã thuê/mua.	
Triggering Event:	Khách hàng đã hoàn tất các bước đàm phán và ký hợp đồng với nhà môi giới.	
Description:	Khách hàng chọn phương thức thanh toán và nhập thông tin thanh toán để hoàn tất giao dịch.	
Actor:	Khách hàng	
Preconditions:	Khách hàng đã hoàn tất các bước đàm phán và ký hợp đồng với nhà môi giới.	
Postconditions:	Thanh toán đã được thực hiện và giao dịch đã hoàn tất.	
Flow of events:	Actor	System
	1. Khách hàng chọn chức năng "Thực hiện thanh toán" sau khi hoàn tất các bước đàm phán và ký hợp đồng. 2. Khách hàng chọn phương thức thanh toán (ví dụ: chuyển khoản, thẻ tín dụng) và nhập thông tin thanh toán cần thiết. 3. Khách hàng xác nhận và hoàn tất giao dịch.	1.1 Hệ thống hiển thị các phương thức thanh toán có sẵn. 2.1 Hệ thống hiển thị trang để điền thông tin thanh toán cần thiết. 3.1 Hệ thống tự động chuyển sang trang chi tiết thanh toán và xác nhận giao dịch thành công
Exception Conditions:	Nếu hệ thống không thể xác thực hoặc xử lý thông tin thanh toán: Hệ thống thông báo cho khách hàng rằng thanh toán không thể thực hiện được và yêu cầu khách hàng thử lại sau.	

Bảng 3.14: Usecase thanh toán

3.3 Class diagrams

3.3.1 Mô tả

3.3.1.1 Role

Mô tả: Vai trò của người dùng trong hệ thống, quyết định quyền hạn mà người dùng có thể thực hiện.

Các vai trò phổ biến:

Admin: Quản lý toàn bộ hệ thống, phê duyệt tài khoản và bài đăng.

Agent: Đăng bài quảng cáo bất động sản, quản lý bất động sản của riêng mình.

Customer: Tìm kiếm, xem thông tin và liên hệ với người môi giới.

Chức năng chính:

Gán vai trò cho người dùng.

Phân quyền dựa trên vai trò.

3.3.1.2 Permission

Mô tả: Quyền truy cập chi tiết của một vai trò, xác định các hành động cụ thể mà người dùng có thể thực hiện.

Ví dụ về quyền hạn:

Xem danh sách bất động sản.

Thêm, sửa, xóa bài đăng.

Quản lý giao dịch.

Chức năng chính:

Gán quyền cho từng vai trò.

Quản lý quyền hạn của hệ thống theo yêu cầu.

3.3.1.3 User

Mô tả: Người dùng trong hệ thống, bao gồm Admin, Agent, và Customer.

Các thông tin cơ bản:

Họ tên, email, số điện thoại.

Vai trò và trạng thái tài khoản (đang hoạt động, bị khóa).

Chức năng chính:

Đăng ký và đăng nhập.

Quản lý thông tin cá nhân.

Thực hiện các chức năng tương ứng với vai trò.

3.3.1.4 Property

Mô tả: Các bất động sản được đăng bán hoặc cho thuê trong hệ thống.

Các thông tin chính:

Tên bất động sản, địa chỉ, mô tả.

Giá, diện tích, loại hình (nhà phố, chung cư, đất nền, v.v.).

Hình ảnh minh họa.

Chức năng chính:

Đăng bài quảng cáo.

Tìm kiếm và lọc bất động sản.

Quản lý trạng thái (đã bán, còn trống).

3.3.1.5 Transaction

Mô tả: Quá trình mua bán hoặc cho thuê bất động sản giữa người mua và người môi giới.

Các thông tin chính:

Người mua, người môi giới.

Bất động sản liên quan.

Giá trị giao dịch, ngày thực hiện.

Trạng thái giao dịch (đang xử lý, hoàn tất, hủy bỏ).

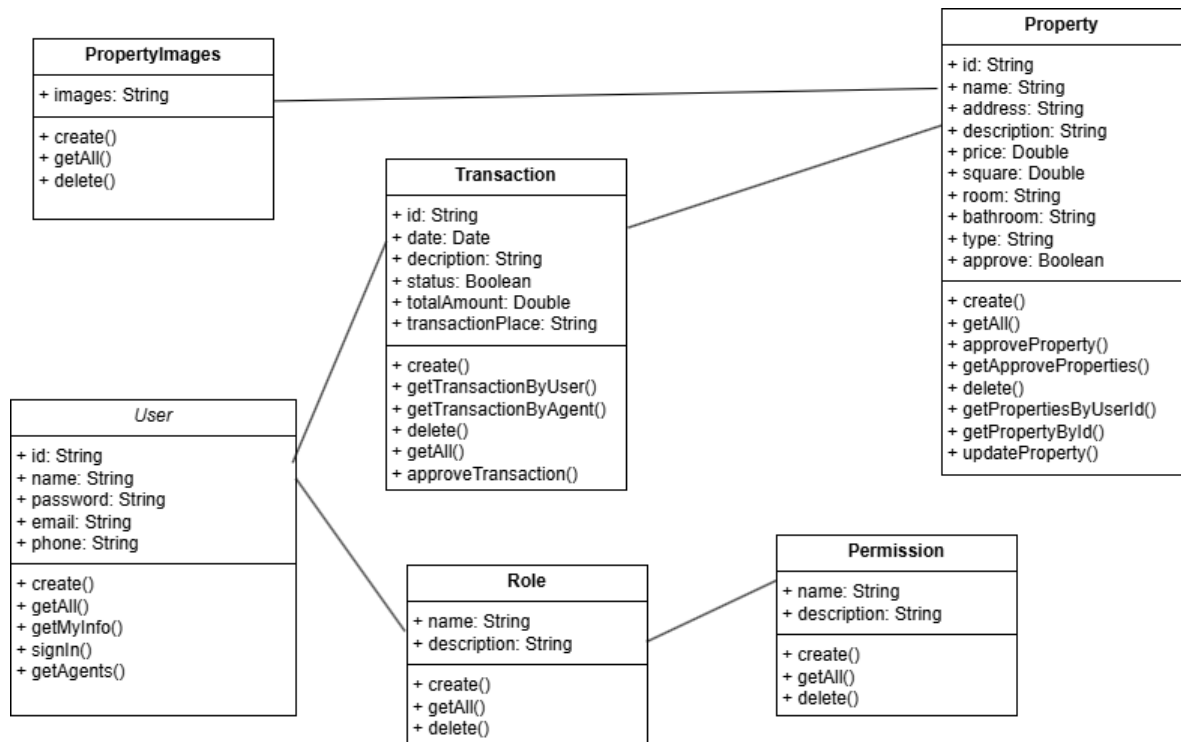
Chức năng chính:

Tạo yêu cầu giao dịch.

Quản lý tiến trình giao dịch.

Lưu lịch sử các giao dịch.

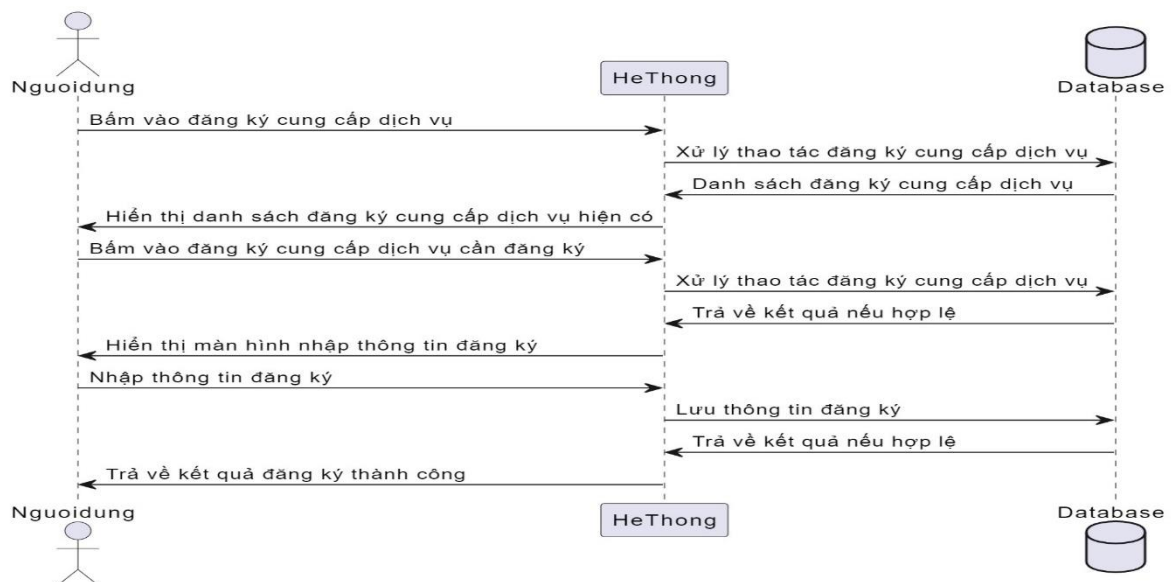
3.3.2 Sơ đồ class diagram



Hình 3.2: Sơ đồ class diagram

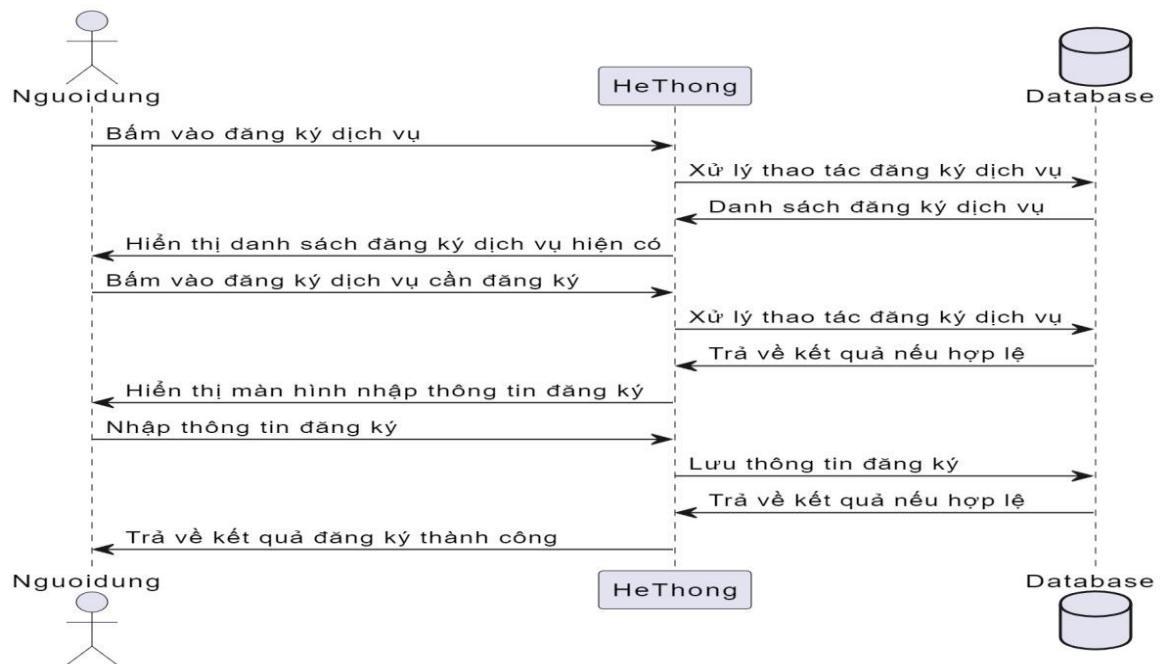
3.4 Sequence Diagram

3.4.1 Đăng ký cung cấp dịch vụ



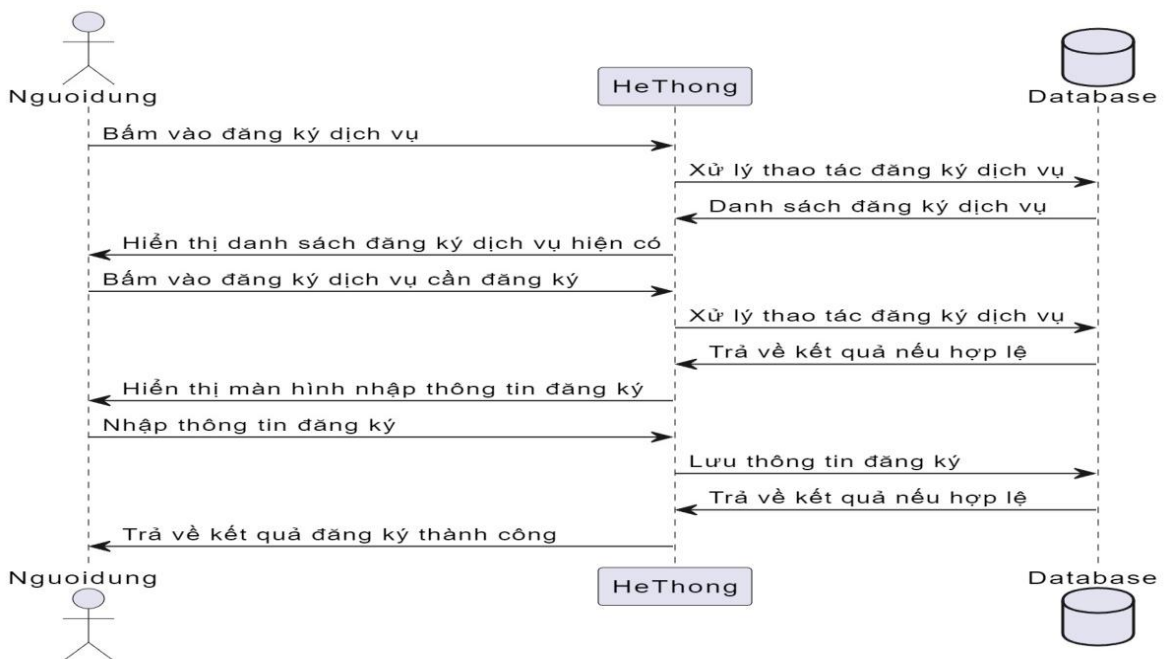
Hình 3.3: Đăng ký cung cấp dịch vụ

3.4.2 Đăng ký dịch vụ



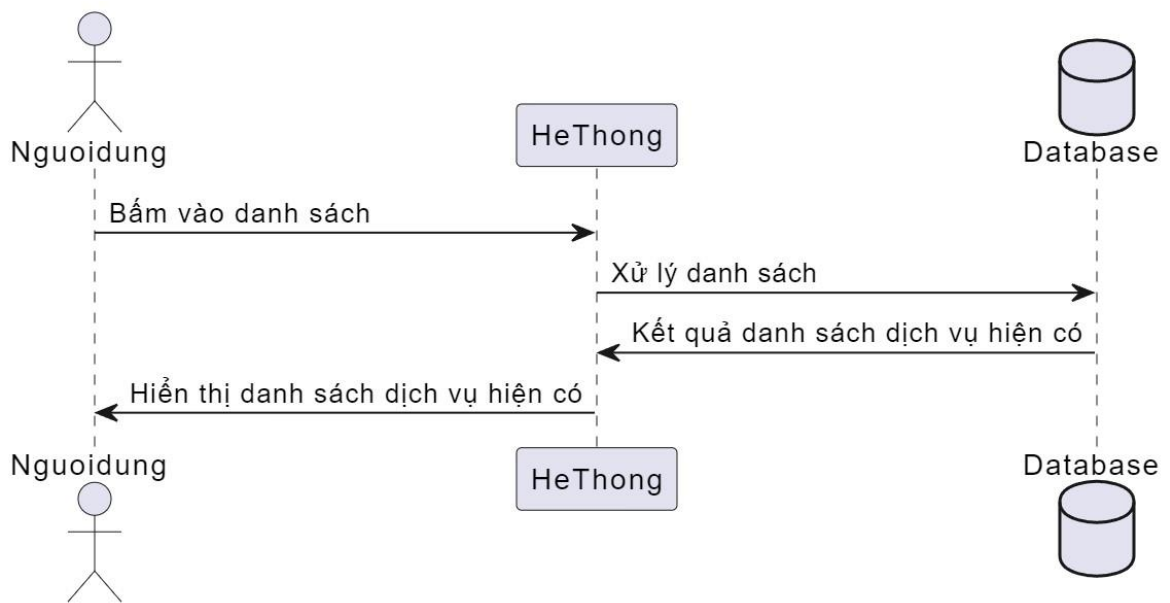
Hình 3.4: Đăng ký dịch vụ

3.4.3 Hủy dịch vụ



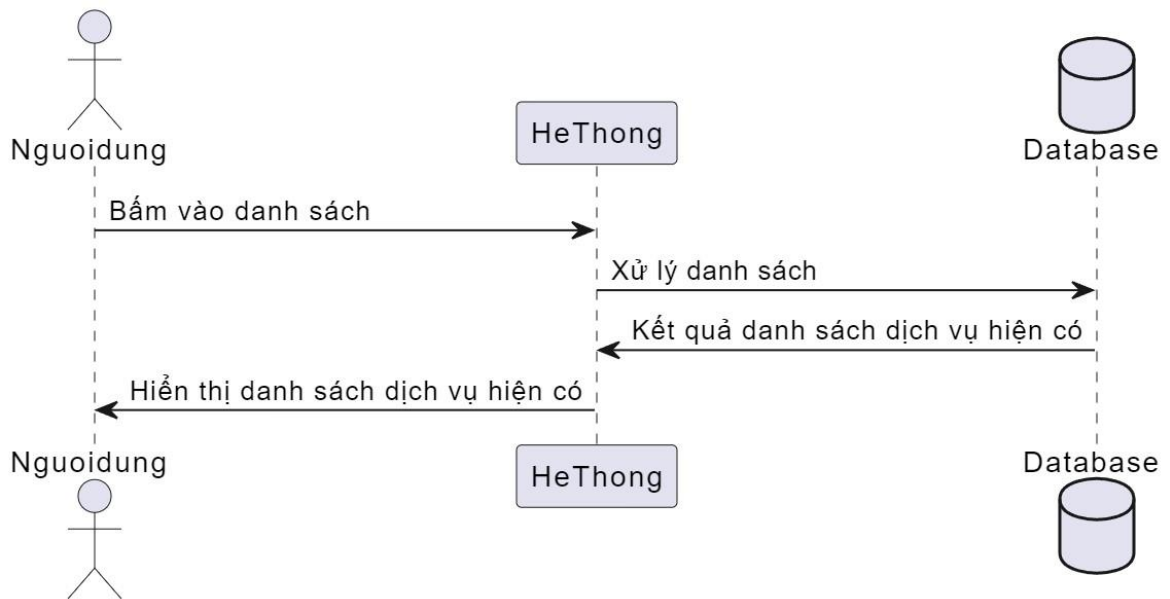
Hình 3.5: Hủy dịch vụ

3.4.4 Xem danh sách dịch vụ



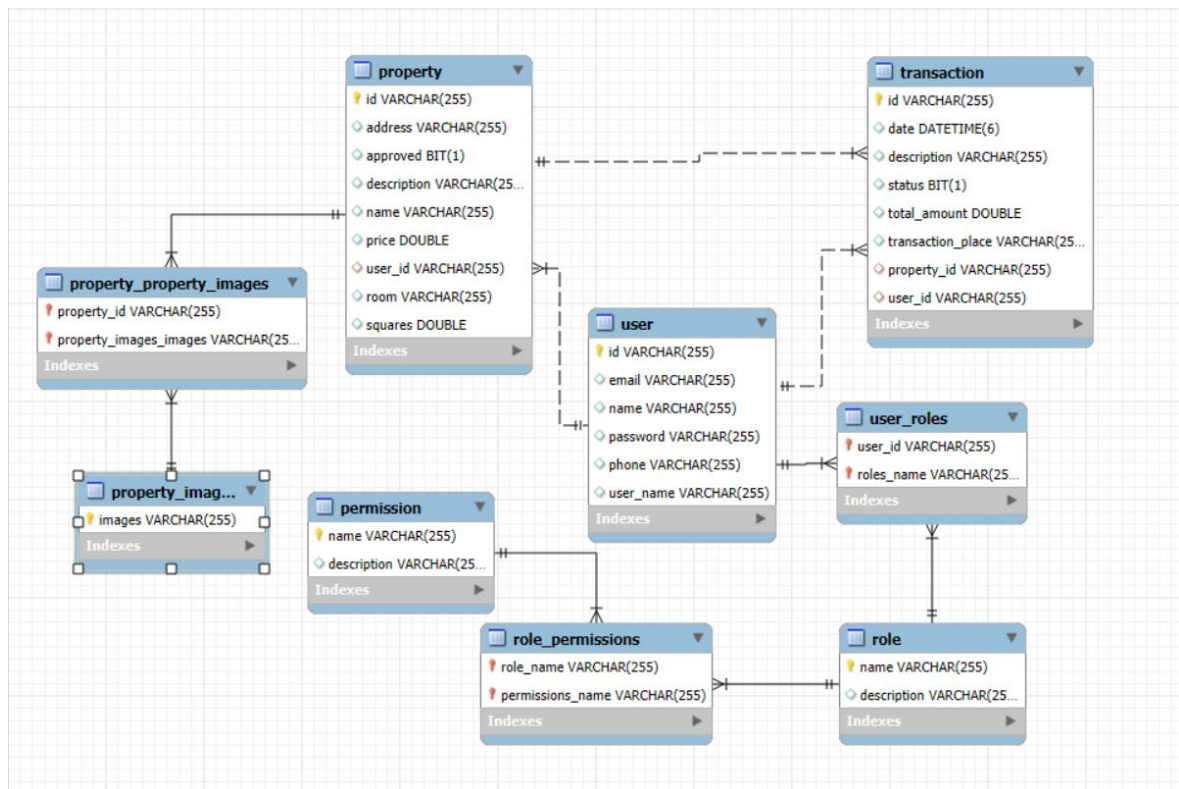
Hình 3.6: Xem danh sách dịch vụ

3.4.5 Xem dịch vụ đang đăng ký



Hình 3.7: Xem dịch vụ đang đăng ký

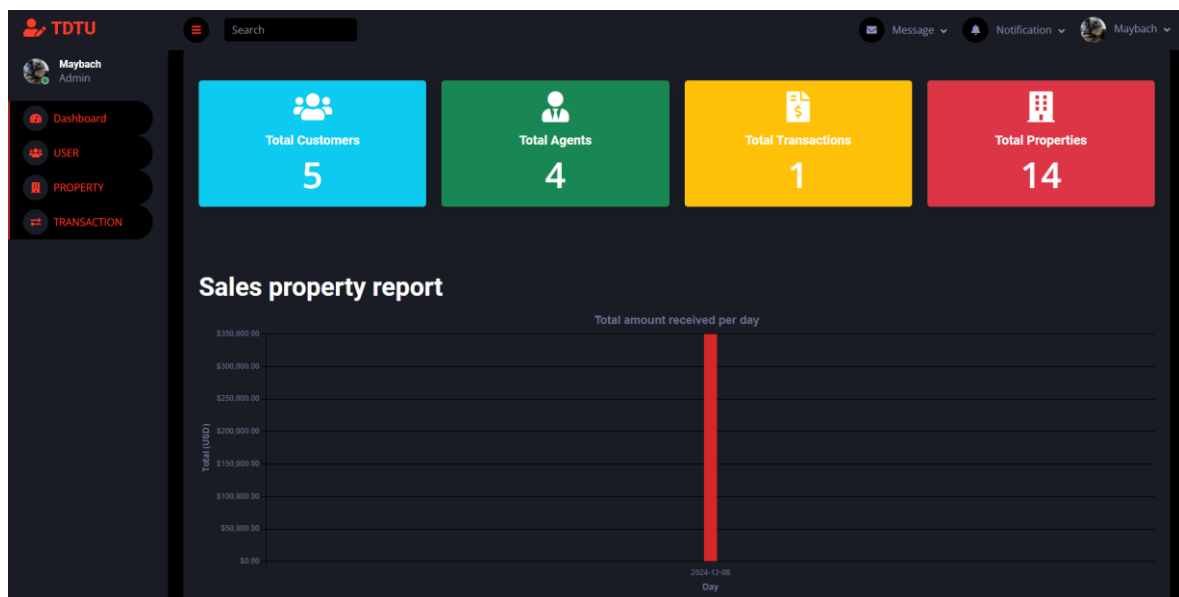
3.5 Sơ đồ ERD



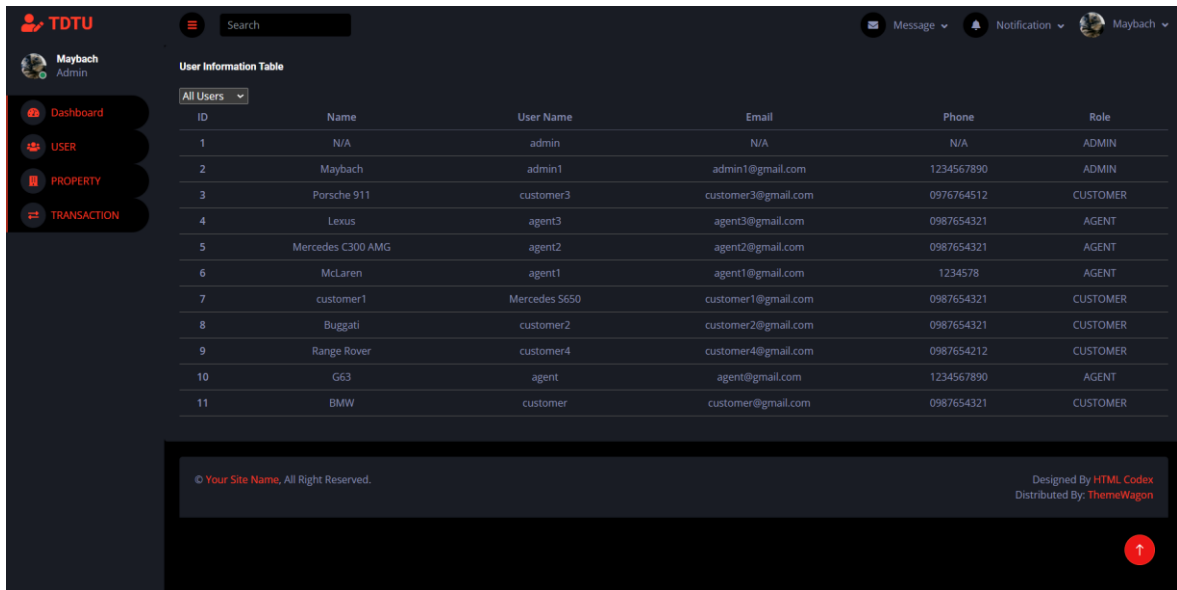
Hình 3.8: Sơ đồ ERD hệ thống

CHƯƠNG 4. THỰC HIỆN HỆ THỐNG

4.1 Trang Admin



Hình 4.1: Trang dashboard của Admin



TDTU Maybach Admin

Search

Message Notification Maybach

User Information Table

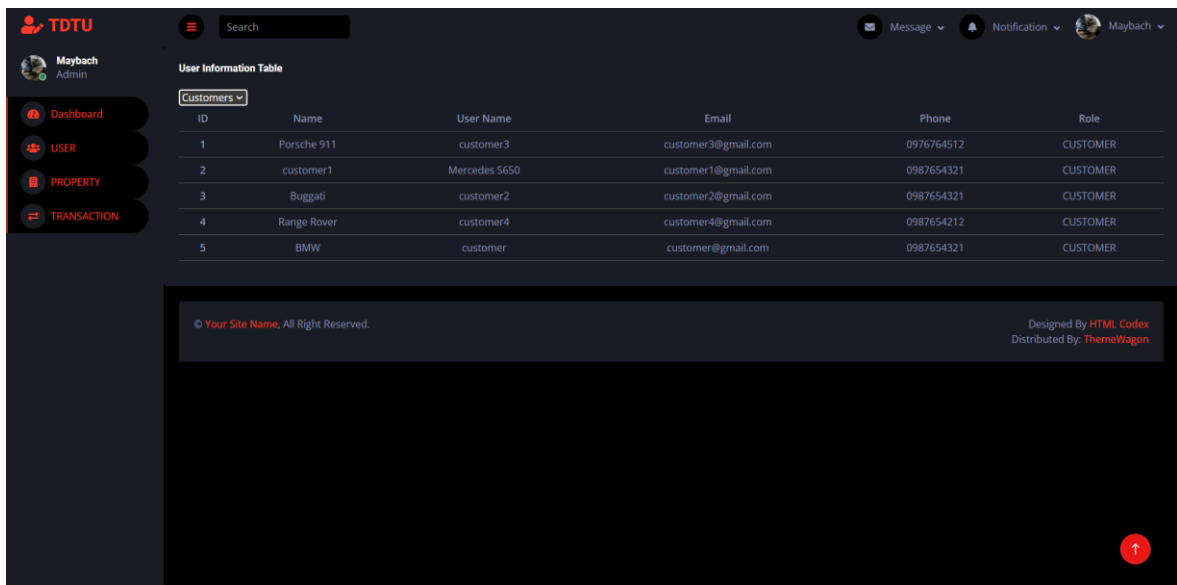
All Users

ID	Name	User Name	Email	Phone	Role
1	N/A	admin	N/A	N/A	ADMIN
2	Maybach	admin1	admin1@gmail.com	1234567890	ADMIN
3	Porsche 911	customer3	customer3@gmail.com	0976764512	CUSTOMER
4	Lexus	agent3	agent3@gmail.com	0987654321	AGENT
5	Mercedes C300 AMG	agent2	agent2@gmail.com	0987654321	AGENT
6	McLaren	agent1	agent1@gmail.com	1234578	AGENT
7	customer1	Mercedes S650	customer1@gmail.com	0987654321	CUSTOMER
8	Bugatti	customer2	customer2@gmail.com	0987654321	CUSTOMER
9	Range Rover	customer4	customer4@gmail.com	0987654212	CUSTOMER
10	G63	agent	agent@gmail.com	1234567890	AGENT
11	BMW	customer	customer@gmail.com	0987654321	CUSTOMER

© Your Site Name, All Right Reserved.

Designed By HTML Codex
Distributed By: ThemeWagon

Hình 4.2: Trang quản lý người dùng của Admin



TDTU Maybach Admin

Search

Message Notification Maybach

User Information Table

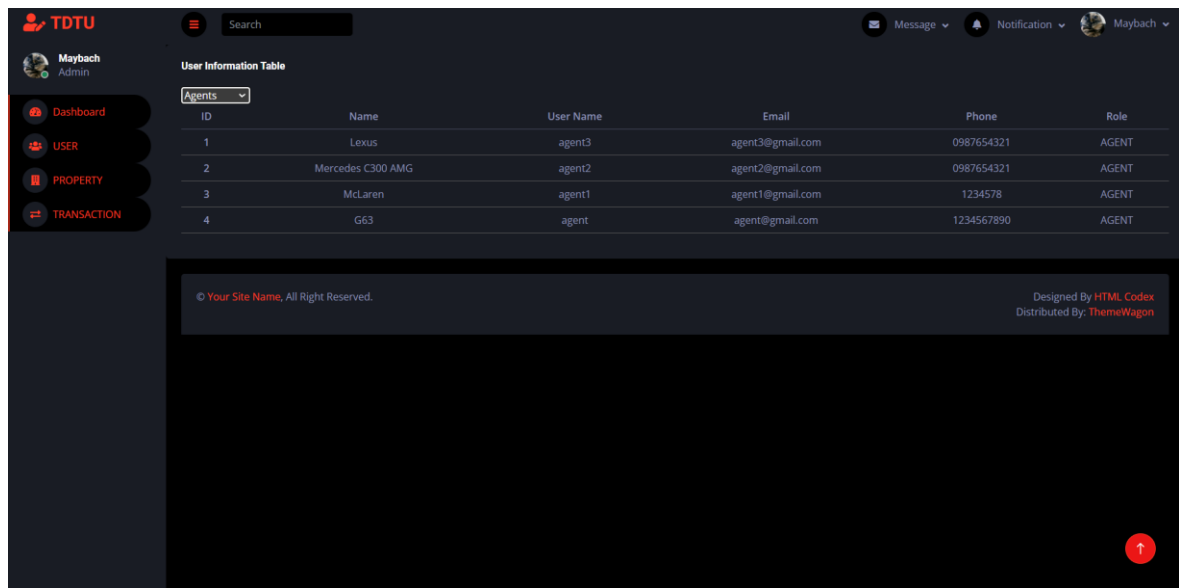
Customers

ID	Name	User Name	Email	Phone	Role
1	Porsche 911	customer3	customer3@gmail.com	0976764512	CUSTOMER
2	customer1	Mercedes S650	customer1@gmail.com	0987654321	CUSTOMER
3	Bugatti	customer2	customer2@gmail.com	0987654321	CUSTOMER
4	Range Rover	customer4	customer4@gmail.com	0987654212	CUSTOMER
5	BMW	customer	customer@gmail.com	0987654321	CUSTOMER

© Your Site Name, All Right Reserved.

Designed By HTML Codex
Distributed By: ThemeWagon

Hình 4.3: Trang quản lý khách hàng của Admin

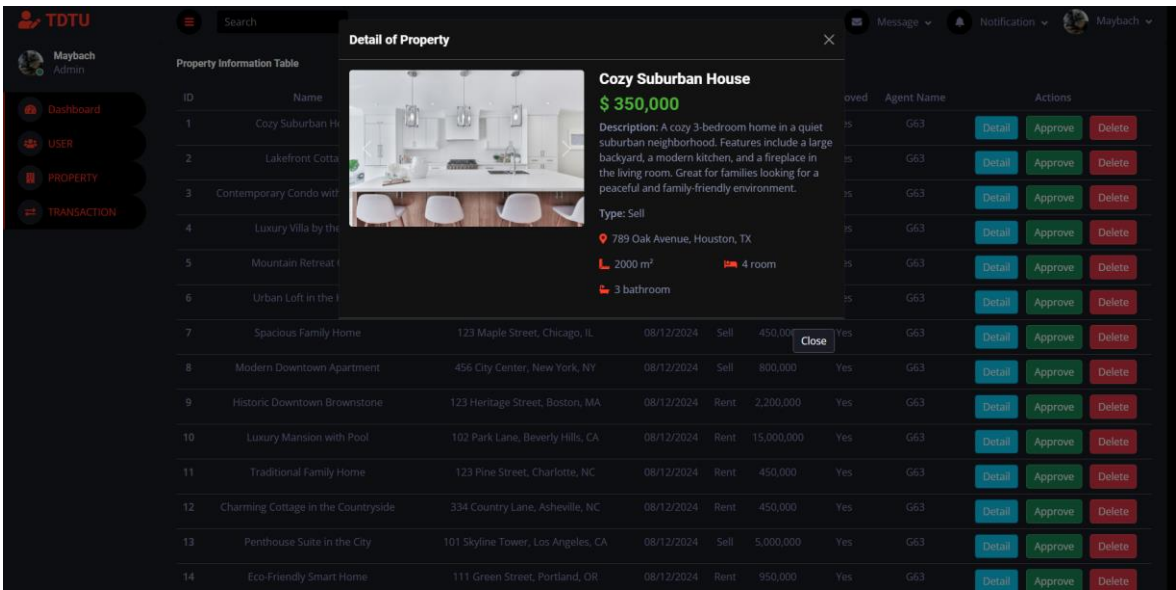


Hình 4.4: Trang quản lý nhà môi giới của Admin

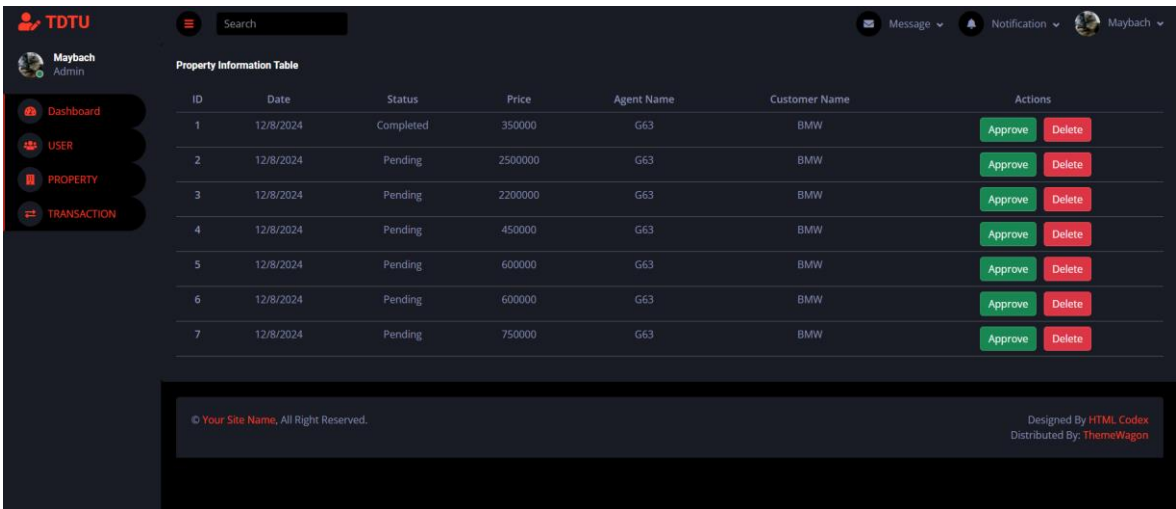
The screenshot displays the TDTU Admin Dashboard with the 'Property Information Table' selected. The table lists 14 properties with columns for ID, Name, Address, Date, Type, Price, Approved, Agent Name, and Actions. Each property entry includes a 'Detail' button, an 'Approve' button, and a 'Delete' button. The footer of the dashboard is the same as in the previous image.

ID	Name	Address	Date	Type	Price	Approved	Agent Name	Actions
1	Cozy Suburban House	789 Oak Avenue, Houston, TX	08/12/2024	Sell	350,000	Yes	G63	Detail Approve Delete
2	Lakefront Cottage	102 Lakeview Drive, Seattle, WA	08/12/2024	Sell	600,000	Yes	G63	Detail Approve Delete
3	Contemporary Condo with Ocean View	456 Ocean Boulevard, San Francisco, CA	08/12/2024	Sell	1,200,000	Yes	G63	Detail Approve Delete
4	Luxury Villa by the Sea	123 Beach Road, Miami, FL	08/12/2024	Rent	2,500,000	Yes	G63	Detail Approve Delete
5	Mountain Retreat Cabin	789 Mountain Road, Aspen, CO	08/12/2024	Sell	750,000	Yes	G63	Detail Approve Delete
6	Urban Loft in the Heart	789 Industrial Avenue, Detroit, MI	08/12/2024	Rent	600,000	Yes	G63	Detail Approve Delete
7	Spacious Family Home	123 Maple Street, Chicago, IL	08/12/2024	Sell	450,000	Yes	G63	Detail Approve Delete
8	Modern Downtown Apartment	456 City Center, New York, NY	08/12/2024	Sell	800,000	Yes	G63	Detail Approve Delete
9	Historic Downtown Brownstone	123 Heritage Street, Boston, MA	08/12/2024	Rent	2,200,000	Yes	G63	Detail Approve Delete
10	Luxury Mansion with Pool	102 Park Lane, Beverly Hills, CA	08/12/2024	Rent	15,000,000	Yes	G63	Detail Approve Delete
11	Traditional Family Home	123 Pine Street, Charlotte, NC	08/12/2024	Rent	450,000	Yes	G63	Detail Approve Delete
12	Charming Cottage in the Countryside	334 Country Lane, Asheville, NC	08/12/2024	Rent	450,000	Yes	G63	Detail Approve Delete
13	Penthouse Suite in the City	101 Skyline Tower, Los Angeles, CA	08/12/2024	Sell	5,000,000	Yes	G63	Detail Approve Delete
14	Eco-Friendly Smart Home	111 Green Street, Portland, OR	08/12/2024	Rent	950,000	Yes	G63	Detail Approve Delete

Hình 4.5: Trang quản lý bất động sản của Admin

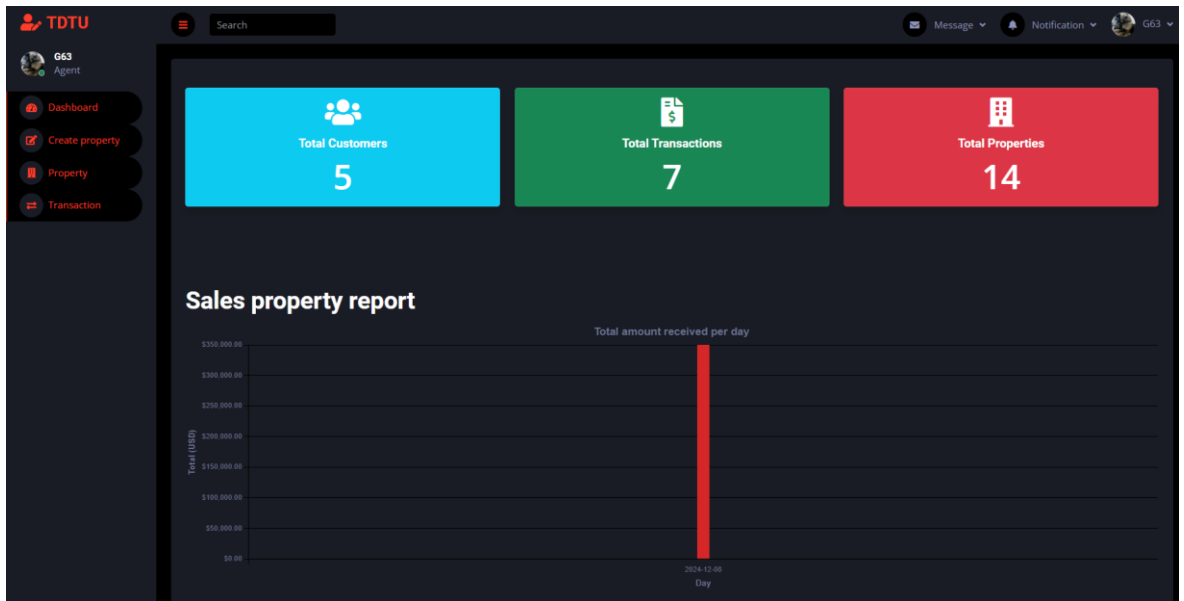


Hình 4.6: Trang xem chi tiết bất động sản của Admin



Hình 4.7: Trang quản lý giao dịch của Admin

4.2 Trang Agent



Hình 4.8: Trang dashboard của Agent

The screenshot shows the 'Property Form' in the TDTU Agent Dashboard. The form is titled 'Property Form' and contains the following fields and options:

- Name: Enter Property Name
- Address: Enter Property Address
- Description: Enter Property Description
- Type: Radio buttons for 'For Sell' (selected) and 'For Rent'
- Price: Enter Property Price
- Squares: Enter Property Squares
- Room: Enter number of room
- Bathroom: Enter number of bathroom
- Property Images: A file upload section with a 'Choose Files' button and 'No file chosen' text.

A red 'Submit' button is located at the bottom of the form.

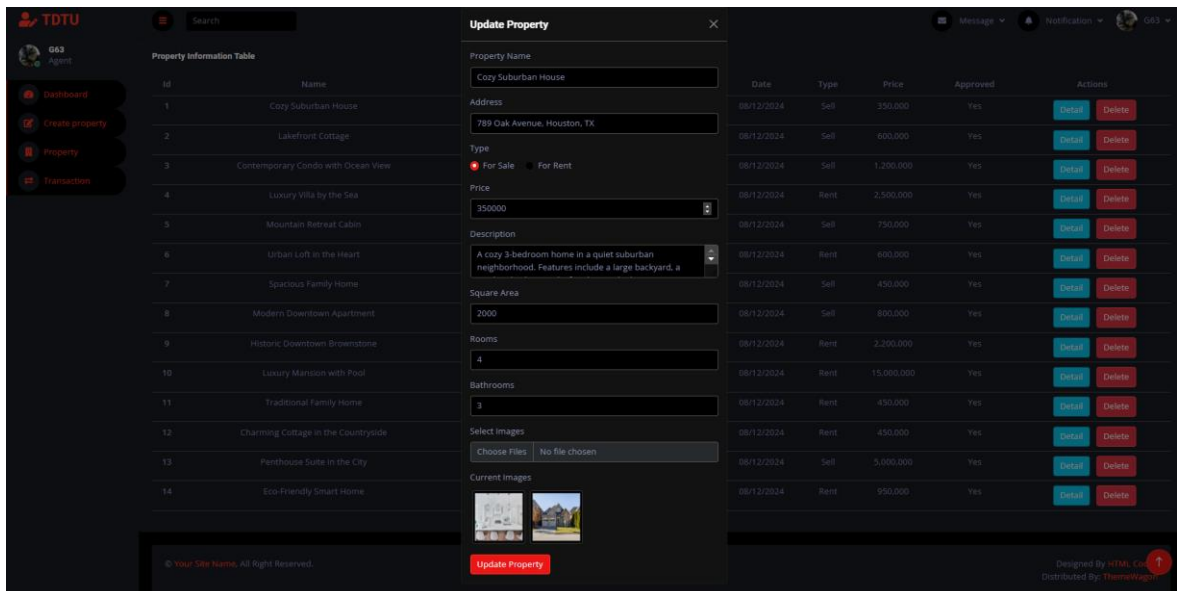
Hình 4.9: Trang tạo bất động sản của Agent



The screenshot shows the TDTU Agent dashboard. On the left is a sidebar with navigation links: Dashboard, Create property, Property, and Transaction. The main area displays a 'Property Information Table' with 14 rows of property data. Each row includes an ID, Name, Address, Date, Type, Price, Approved status, and Actions (Detail and Delete buttons).

Id	Name	Address	Date	Type	Price	Approved	Actions
1	Cozy Suburban House	789 Oak Avenue, Houston, TX	08/12/2024	Sell	350,000	Yes	Detail Delete
2	Lakefront Cottage	102 Lakeview Drive, Seattle, WA	08/12/2024	Sell	600,000	Yes	Detail Delete
3	Contemporary Condo with Ocean View	456 Ocean Boulevard, San Francisco, CA	08/12/2024	Sell	1,200,000	Yes	Detail Delete
4	Luxury Villa by the Sea	123 Beach Road, Miami, FL	08/12/2024	Rent	2,500,000	Yes	Detail Delete
5	Mountain Retreat Cabin	789 Mountain Road, Aspen, CO	08/12/2024	Sell	750,000	Yes	Detail Delete
6	Urban Loft in the Heart	789 Industrial Avenue, Detroit, MI	08/12/2024	Rent	600,000	Yes	Detail Delete
7	Spacious Family Home	123 Maple Street, Chicago, IL	08/12/2024	Sell	450,000	Yes	Detail Delete
8	Modern Downtown Apartment	456 City Center, New York, NY	08/12/2024	Sell	800,000	Yes	Detail Delete
9	Historic Downtown Brownstone	123 Heritage Street, Boston, MA	08/12/2024	Rent	2,200,000	Yes	Detail Delete
10	Luxury Mansion with Pool	102 Park Lane, Beverly Hills, CA	08/12/2024	Rent	15,000,000	Yes	Detail Delete
11	Traditional Family Home	123 Pine Street, Charlotte, NC	08/12/2024	Rent	450,000	Yes	Detail Delete
12	Charming Cottage in the Countryside	334 Country Lane, Asheville, NC	08/12/2024	Rent	450,000	Yes	Detail Delete
13	Penthouse Suite in the City	101 Skyline Tower, Los Angeles, CA	08/12/2024	Sell	5,000,000	Yes	Detail Delete
14	Eco-Friendly Smart Home	111 Green Street, Portland, OR	08/12/2024	Rent	950,000	Yes	Detail Delete

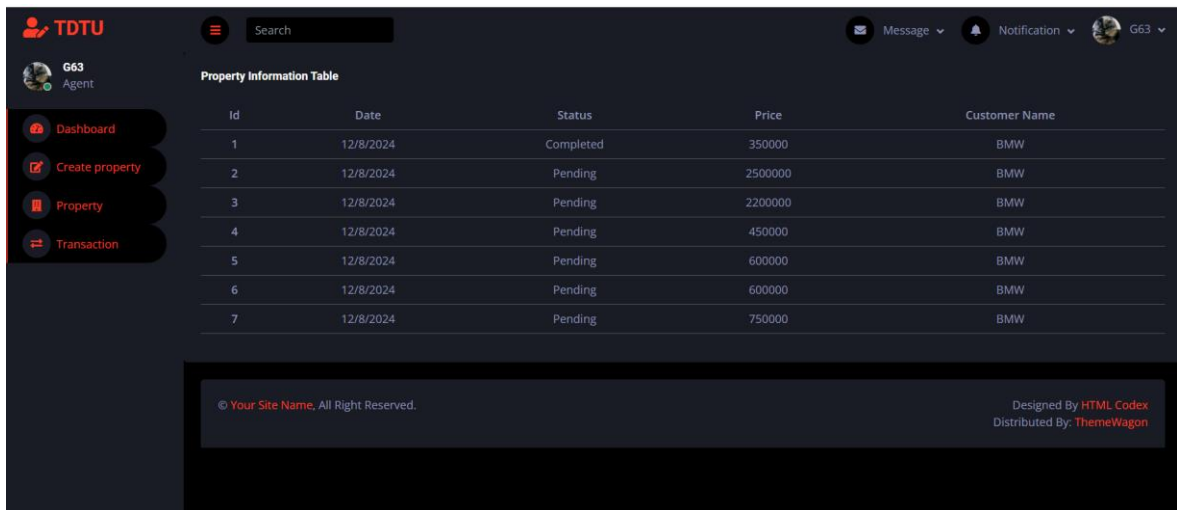
Hình 4.10: Trang quản lý bất động sản của Agent



The screenshot shows the TDTU Agent dashboard with the 'Update Property' modal open. The modal contains a form for updating property details. The 'Property Name' field is set to 'Cozy Suburban House'. The 'Address' field is set to '789 Oak Avenue, Houston, TX'. The 'Type' is set to 'For Sale'. The 'Price' is set to '350000'. The 'Description' is set to 'A cozy 3-bedroom home in a quiet suburban neighborhood. Features include a large backyard, a'. The 'Square Area' is set to '2000'. The 'Rooms' are set to '4'. The 'Bathrooms' are set to '3'. The 'Select Images' section shows 'Choose Files' and 'No file chosen'. The 'Current Images' section shows two images of the property. The 'Update Property' button is at the bottom of the modal. The background shows the 'Property Information Table' with 14 rows of property data.

Id	Name	Address	Date	Type	Price	Approved	Actions
1	Cozy Suburban House	789 Oak Avenue, Houston, TX	08/12/2024	Sell	350,000	Yes	Detail Delete
2	Lakefront Cottage	102 Lakeview Drive, Seattle, WA	08/12/2024	Sell	600,000	Yes	Detail Delete
3	Contemporary Condo with Ocean View	456 Ocean Boulevard, San Francisco, CA	08/12/2024	Sell	1,200,000	Yes	Detail Delete
4	Luxury Villa by the Sea	123 Beach Road, Miami, FL	08/12/2024	Rent	2,500,000	Yes	Detail Delete
5	Mountain Retreat Cabin	789 Mountain Road, Aspen, CO	08/12/2024	Sell	750,000	Yes	Detail Delete
6	Urban Loft in the Heart	789 Industrial Avenue, Detroit, MI	08/12/2024	Rent	600,000	Yes	Detail Delete
7	Spacious Family Home	123 Maple Street, Chicago, IL	08/12/2024	Sell	450,000	Yes	Detail Delete
8	Modern Downtown Apartment	456 City Center, New York, NY	08/12/2024	Sell	800,000	Yes	Detail Delete
9	Historic Downtown Brownstone	123 Heritage Street, Boston, MA	08/12/2024	Rent	2,200,000	Yes	Detail Delete
10	Luxury Mansion with Pool	102 Park Lane, Beverly Hills, CA	08/12/2024	Rent	15,000,000	Yes	Detail Delete
11	Traditional Family Home	123 Pine Street, Charlotte, NC	08/12/2024	Rent	450,000	Yes	Detail Delete
12	Charming Cottage in the Countryside	334 Country Lane, Asheville, NC	08/12/2024	Rent	450,000	Yes	Detail Delete
13	Penthouse Suite in the City	101 Skyline Tower, Los Angeles, CA	08/12/2024	Sell	5,000,000	Yes	Detail Delete
14	Eco-Friendly Smart Home	111 Green Street, Portland, OR	08/12/2024	Rent	950,000	Yes	Detail Delete

Hình 4.11: Trang cập nhật bất động sản của Agent



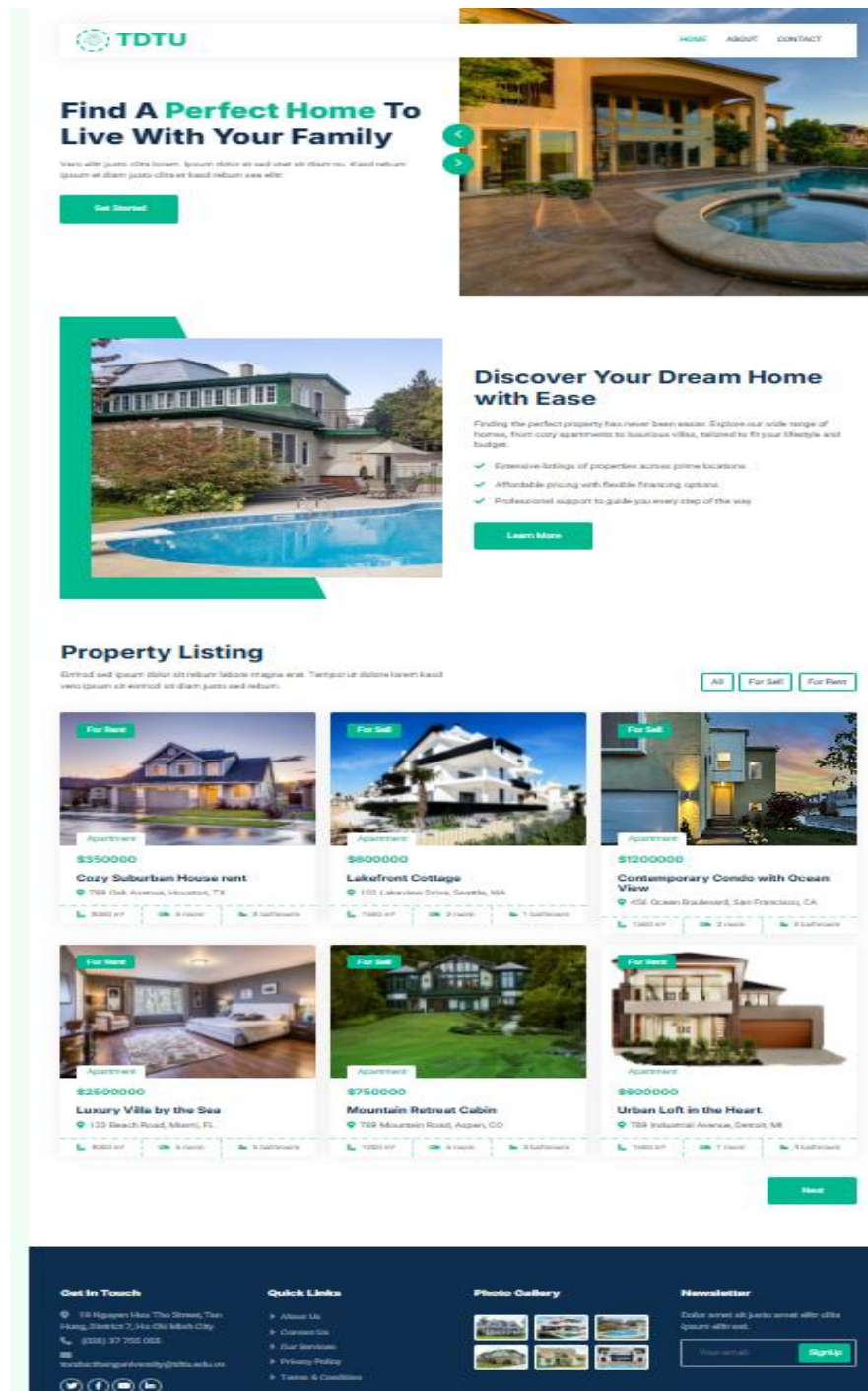
The screenshot displays the TDTU Agent dashboard. On the left is a sidebar with navigation links: Dashboard, Create property, Property, and Transaction. The main area features a 'Property Information Table' with the following data:

Id	Date	Status	Price	Customer Name
1	12/8/2024	Completed	350000	BMW
2	12/8/2024	Pending	2500000	BMW
3	12/8/2024	Pending	2200000	BMW
4	12/8/2024	Pending	450000	BMW
5	12/8/2024	Pending	600000	BMW
6	12/8/2024	Pending	600000	BMW
7	12/8/2024	Pending	750000	BMW


At the bottom of the dashboard, there is a footer area containing the text: © Your Site Name, All Right Reserved. and Designed By HTML Codex Distributed By: ThemeWagon.

Hình 4.12: Trang quản lý giao dịch với khách hàng của Agent

4.3 Trang Customer




Hình 4.13: Trang giao diện chính của khách hàng



TDTU

[HOME](#)
[ABOUT](#)
[CONTACT](#)

Property Detail





[HOME](#) / [PROPERTY DETAILS](#)






\$ 2500000

Luxury Villa by the Sea

 123 Beach Road, Miami, FL
 4000 m²
 5 room
 5 bathroom




[Make Appointment](#)

A beautiful luxury villa located by the beach with stunning ocean views. The villa features 5 spacious bedrooms, a private swimming pool, and an expansive garden. Perfect for those seeking a private and serene living space.



G63
Agent


Get In Touch

 19 Nguyen Huu Tho Street, Tan Hung, District 7, Ho Chi Minh City
 (028) 37 755 055
 tonducthanguniversity@tdtu.edu.vn

Quick Links

- [About Us](#)
- [Contact Us](#)
- [Our Services](#)
- [Privacy Policy](#)
- [Terms & Condition](#)

Photo Gallery

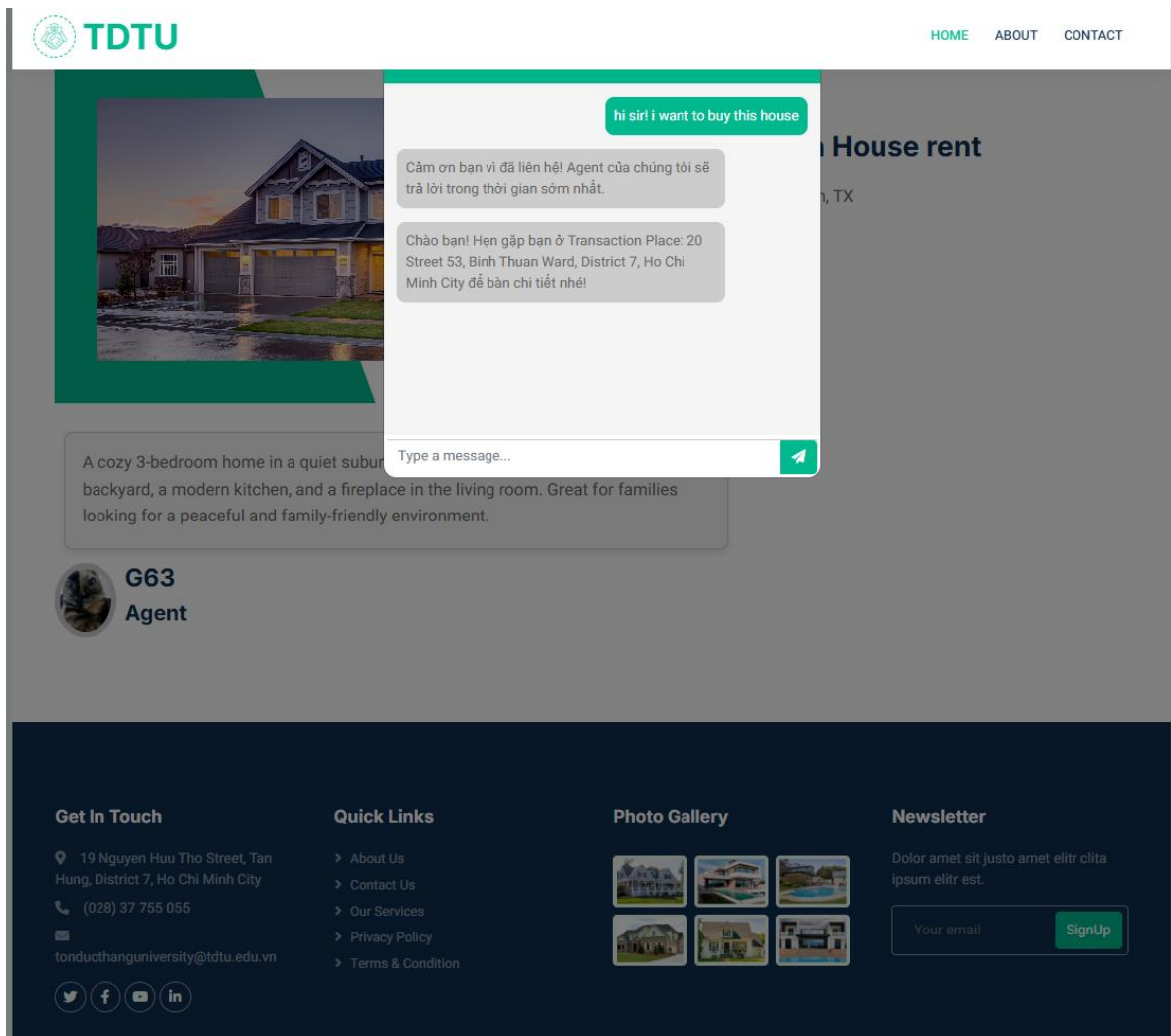


Newsletter

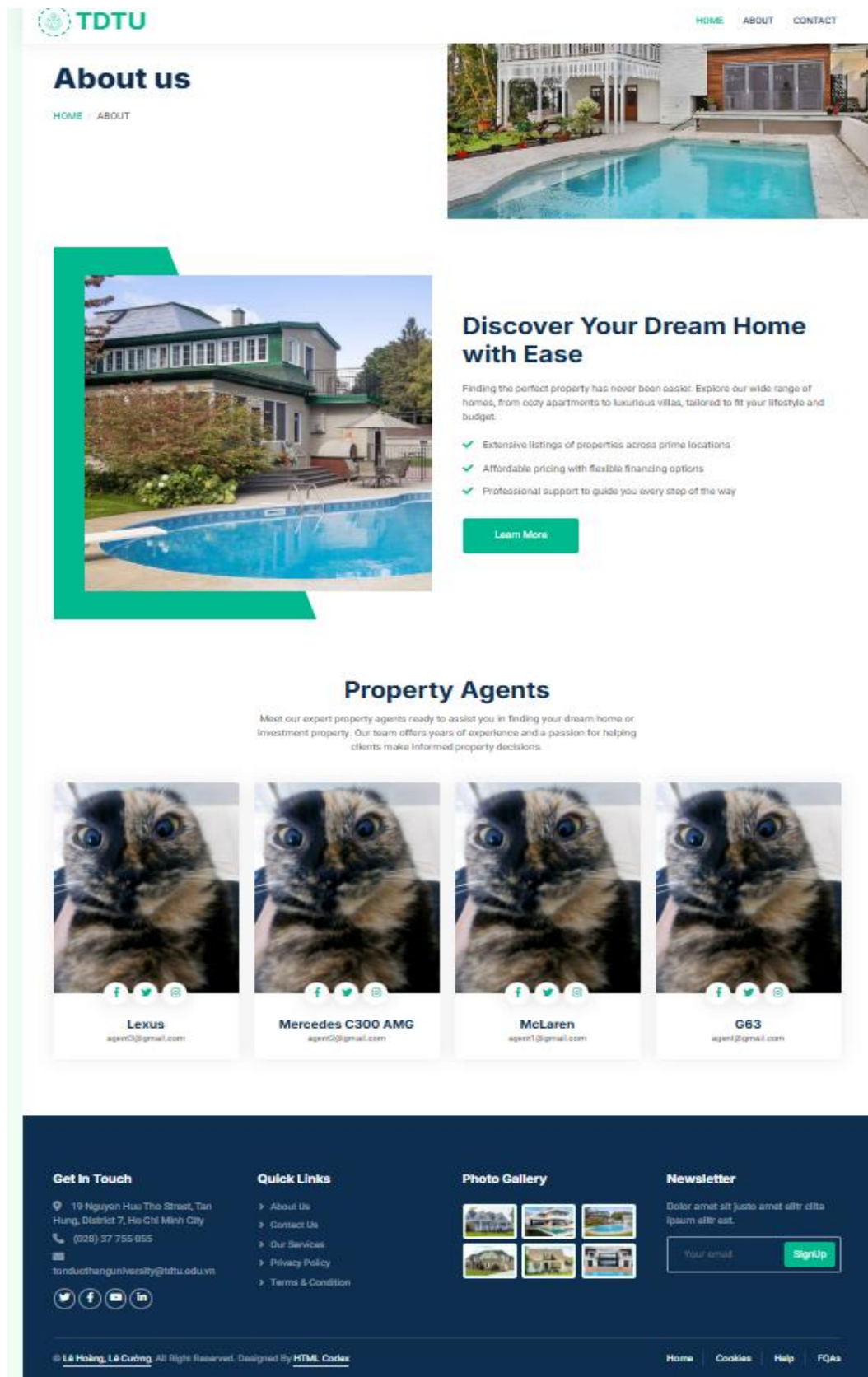
Dolor amet sit justo amet elittr clita ipsum elittr est.

[SignUp](#)

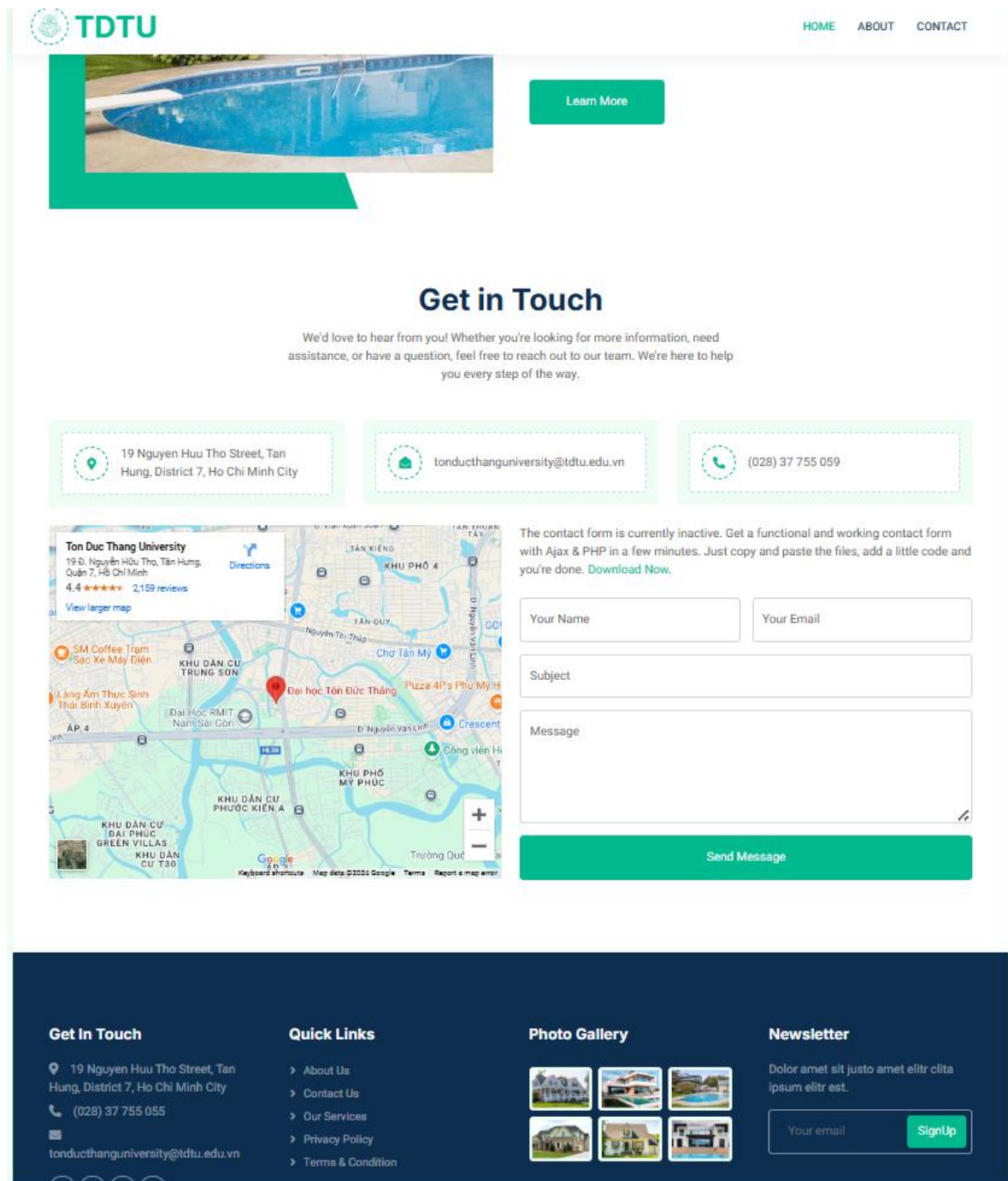
Hình 4.14: Trang giao diện chi tiết của bất động sản



Hình 4.15: Trang giao diện khách hàng muốn liên hệ với nhà môi giới



Hình 4.16: Trang giao diện giới thiệu các nhà môi giới



Hình 4.17: Trang giao diện liên hệ

CHƯƠNG 5. THUẬN LỢI VÀ KHÓ KHĂN

5.1 Thuận lợi

Công nghệ và Framework

- **Spring Boot dễ sử dụng:** Spring Boot cung cấp nhiều cấu hình mặc định, giúp bạn tiết kiệm thời gian khi thiết lập dự án và tập trung vào việc phát triển các tính năng chính.
- **Khả năng mở rộng:** Spring Boot dễ dàng mở rộng và tích hợp với các công cụ như Hibernate (JPA), Thymeleaf, MongoDB, Elasticsearch,...
- **Hỗ trợ tốt cho bảo mật:** Spring Security hỗ trợ việc phân quyền, xác thực, giúp đảm bảo an toàn dữ liệu, đặc biệt quan trọng khi xây dựng hệ thống quản lý bất động sản.
- **Khả năng tích hợp mạnh mẽ:** Dễ dàng kết nối với các dịch vụ bên ngoài (như thanh toán online, bản đồ Google Maps) qua REST API.

Quản lý dự án

- **Hỗ trợ nhiều công cụ phát triển:** IntelliJ IDEA hoặc Eclipse giúp việc viết code, kiểm tra và gỡ lỗi dễ dàng hơn.
- **Kiến trúc rõ ràng:** Sử dụng mô hình MVC (Model-View-Controller) giúp phân chia trách nhiệm rõ ràng, dễ bảo trì và mở rộng.

Tính năng nghiệp vụ

- **Tích hợp phân quyền người dùng:** Phân vai trò (admin, agent, customer) rõ ràng, giúp tăng tính chuyên nghiệp.
- **Khả năng tùy chỉnh linh hoạt:** Có thể tùy chỉnh các thuộc tính như giá bất động sản, hình ảnh, mô tả... phù hợp với nhu cầu khách hàng.
- **Quản lý tài nguyên:** Hỗ trợ lưu trữ và xử lý thông tin lớn như danh sách bất động sản, hình ảnh, giao dịch.

5.2 Khó khăn

Kỹ thuật

- **Cấu hình bảo mật phức tạp:** Spring Security yêu cầu cấu hình chi tiết, dễ dẫn đến lỗi nếu không kiểm tra kỹ (như frontend không thể truy cập).
- **Hiệu suất hệ thống:** Khi dữ liệu lớn (hình ảnh, giao dịch), cần tối ưu cơ sở dữ liệu và caching (Redis hoặc Ehcache) để tránh chậm trễ.
- **Xử lý ngoại lệ:** Cần xây dựng hệ thống quản lý ngoại lệ tốt để tránh lỗi khi người dùng nhập dữ liệu không hợp lệ.

Triển khai và vận hành

- **Triển khai trên server:** Cần xử lý các vấn đề liên quan đến build ứng dụng, deploy trên server hoặc cloud (như AWS, Azure).
- **Tương thích với frontend:** Nếu sử dụng frontend framework (React, Angular), cần đảm bảo API hoạt động đúng cách, đặc biệt khi xử lý CORS.

Nhiệm vụ

- **Tích hợp tính năng phức tạp:** Tích hợp tính năng duyệt bất động sản trước khi hiển thị hoặc quản lý trạng thái giao dịch đôi khi gây khó khăn khi xử lý logic.
- **Đáp ứng yêu cầu UI/UX:** Xây dựng giao diện thân thiện, trực quan (đặc biệt với người dùng không quen công nghệ) đòi hỏi đầu tư nhiều thời gian.
- **Quản lý dữ liệu**
- **Đảm bảo tính toàn vẹn dữ liệu:** Khi nhiều người dùng cùng thao tác (như cập nhật thông tin bất động sản), cần cơ chế khóa hoặc kiểm tra để tránh xung đột.
- **Phân tích dữ liệu:** Nếu muốn tích hợp phân tích dữ liệu (ví dụ: bất động sản nào được tìm kiếm nhiều), cần công cụ hỗ trợ (Elasticsearch, Kibana).

CHƯƠNG 6. PHÂN CÔNG NHIỆM VỤ

Họ và tên	Nhiệm vụ	Tiến độ
Lê Văn Cường	Code và viết báo cáo	100%
Lê Minh Hoàng	Code và viết báo cáo	100%

CHƯƠNG 7. TÀI LIỆU THAM KHẢO

<https://kungfutech.edu.vn/khoa-hoc/spring-boot>

<https://docs.spring.io/spring-security/reference/index.html>