|  |  |
| --- | --- |
|  |  |
| VŨ QUỐC HÙNG | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
| Logo truong |
| **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **NGÀNH CÔNG NGHỆ THÔNG TIN** |
|  |
| **XÂY DỰNG WEBSITE CUNG CẤP DỊCH VỤ CHO THUÊ NHÀ TRỌ TRÊN ĐỊA BÀN HÀ NỘI** |
|  |
|  |
| **CBHD : TS. Nguyễn Văn Tỉnh** |
| CÔNG NGHỆ THÔNG TIN | **Sinh viên : Vũ Quốc Hùng** |
| **Mã số sinh viên : 2020605491** |
|  |
|  |
|  |
| Hà Nội – Năm 2024 |

**MỤC LỤC**

[LỜI NÓI ĐẦU ii](#_Toc21361)

[DANH MỤC CÁC CHỮ VIẾT TẮT iv](#_Toc25326)

[DANH MỤC BẢNG BIỂU v](#_Toc9416)

[DANH MỤC HÌNH ẢNH vi](#_Toc7995)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT, CÔNG CỤ SỬ DỤNG VÀ KHẢO SÁT HỆ THỐNG 1](#_Toc32013)

[1.1. Cơ sở lý thuyết 1](#_Toc5254)

[1.2. Công cụ sử dụng 14](#_Toc12801)

[1.3. Khảo sát hệ thống 14](#_Toc2508)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾT HỆ THỐNG 17](#_Toc7652)

[2.1. Phân tích hệ thống 17](#_Toc19109)

[2.2. Xây dựng mô hình use case 18](#_Toc5882)

[2.3. Phân tích use case 34](#_Toc20769)

[2.4. Thiết kế cơ sở dữ liệu 47](#_Toc16492)

[2.5. Thiết kế giao diện 51](#_Toc26553)

[CHƯƠNG 3. KẾT QUẢ, CÀI ĐẶT VÀ KIỂM THỬ 53](#_Toc26361)

[3.1. Kết quả đạt được 53](#_Toc19378)

[3.2. Cài đặt chương trình 63](#_Toc28748)

[3.3. Kiểm thử chương trình 64](#_Toc23454)

[KẾT LUẬN 68](#_Toc23033)

[TÀI LIỆU THAM KHẢO 69](#_Toc31457)

# LỜI NÓI ĐẦU

Trong thời đại hiện đại, việc sử dụng công nghệ thông tin cho việc tìm kiếm nhà ở đang trở thành một xu hướng không thể phủ nhận. Sự tiến bộ của công nghệ đã mở ra những cơ hội vô cùng lớn lao để cải thiện chất lượng cuộc sống và giúp mọi người có thể dễ dàng tìm kiếm và lựa chọn những căn phòng hoặc căn hộ phù hợp với nhu cầu và ngân sách của mình thông qua các bộ lọc thông minh, đặc biệt là đối với những người đang ở xa có thể nhận được thông tin và tìm được những căn phòng ưng ý nhất, những người có phòng cho thuê có thể dễ dàng tìm kiếm người thuê. Tại Hà Nội, nhu cầu thuê nhà trọ đang ngày càng gia tăng mạnh mẽ do sự phát triển nhanh chóng của thành phố và sự gia nhập của nhiều người mới từ các vùng khác. Với lượng dân cư đông đúc và sự bùng nổ của các khu công nghiệp, trường học và cơ sở kinh doanh, nhu cầu về chỗ ở tiện nghi và giá cả phải chăng luôn là một vấn đề quan trọng. Các sinh viên, người lao động, và cả các gia đình trẻ đều đang tìm kiếm những giải pháp thuê trọ phù hợp với điều kiện tài chính và vị trí thuận lợi. Sự phát triển của thị trường cho thuê nhà trọ ở Hà Nội không chỉ phản ánh nhu cầu ngày càng cao về chỗ ở mà còn mở ra cơ hội lớn cho các dịch vụ và nền tảng cung cấp giải pháp nhà trọ đáp ứng nhu cầu đa dạng của cư dân thành phố. Công nghệ không chỉ giúp cho việc tìm trọ trở nên dễ dàng hơn mà còn tạo ra một kênh giao tiếp thuận lợi giữa người có nhu cầu thuê trọ và những người có nhu cầu cho thuê trọ.

Sau khi tìm hiểu và dưới sự hướng dẫn của thầy Nguyễn Văn Tỉnh, em đã phân tích và thiết kế được **Website cung cấp dịch vụ cho thuê nhà trọ trên địa bàn Hà Nội**. Để giúp người sử dụng có thể kết nối với những người cho thuê. Đó cũng là lý do mà em chọn đề tài này.

Để hoàn thành được đồ án tốt nghiệp này, em xin được gửi lời cảm ơn chân thành đến các thầy cô trong khoa Công nghệ thông tin Trường Đại học Công Nghiệp Hà Nội đã tận tình giảng dạy và trang bị kiến thức cho em trong suốt thời gian em học tập tại trường. Thầy giáo hướng dẫn đề tài - **Tiến Sĩ Nguyễn Văn Tỉnh**, giảng viên Khoa Công nghệ Thông tin Trường Đại học Công Nghiệp Hà Nội – đã tận tụy hết lòng giúp đỡ, hướng dẫn, chỉ dẫn tận tình để giúp em hoàn thành được đồ án. Bạn bè đã góp ý và giúp đỡ, bên cạnh động viên và khích lệ em trong suốt thời gian em làm đồ án.

***Hà Nội, 29 Ngày 9 tháng năm 2024***

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| Từ viết tắt | Giải thích |
| API | Application Programming Interface |
| OOP | Object Oriented Programming |
| IBM | International Business Machines |
| PK | Primary Key |
| FK | Foreign Key |

# DANH MỤC BẢNG BIỂU

[Bảng 3.1: Kiểm thử chức năng phía người dùng 66](#_Toc18852)

# DANH MỤC HÌNH ẢNH

[Hình 1.1: ASP.NET Core 4](#_Toc11579)

[Hình 1.2: Mô hình Model-View-Controller 5](#_Toc9507)

[Hình 1.3: Cơ chế hoạt động của ASP.NET 6](#_Toc22211)

[Hình 1.4: Kiến trúc của Entity Framework 8](#_Toc10631)

[Hình 1.5: Database first 10](#_Toc7999)

[Hình 1.6: Model-first 10](#_Toc17728)

[Hình 1.7: Code first 11](#_Toc2610)

[Hình 2.1: Biểu đồ use case phía Admin 19](#_Toc28967)

[Hình 2.2: Biểu đồ use case phía người dùng 20](#_Toc19159)

[Hình 2.3: Biểu đồ trình tự chức năng đăng nhập 34](#_Toc32204)

[Hình 2.4: Biểu đồ lớp chức năng đăng nhập 35](#_Toc5980)

[Hình 2.5: Biểu đồ trình tự chức năng đăng ký 35](#_Toc4466)

[Hình 2.6: Biểu đồ lớp chức năng đăng nhập 36](#_Toc23057)

[Hình 2.7: Biểu đồ trình tự chức năng tìm kiếm nhà trọ 37](#_Toc26933)

[Hình 2.8: Biểu đồ lớp chức năng tìm kiếm nhà trọ 38](#_Toc30772)

[Hình 2.9: Biểu đồ trình tự chức năng xem thông tin bài đăng 39](#_Toc5161)

[Hình 2.10: Biều đồ lớp chức năng xem thông tin bài đăng 40](#_Toc10977)

[Hình 2.11: Biểu đồ trình tự của chức năng sửa thông tin cá nhân 40](#_Toc5794)

[Hình 2.12: Biểu đồ lớp chức năng sửa thông tin cá nhân 41](#_Toc13464)

[Hình 2.13: Biểu đồ trình tự chức năng đăng bài 41](#_Toc9272)

[Hình 2.14: Biểu đồ lớp chức năng đăng bài 42](#_Toc28990)

[Hình 2.15: Biểu đồ trình tự chức năng nhắn tin 43](#_Toc3371)

[Hình 2.16: Biểu đồ lớp chức năng nhắn tin 44](#_Toc27738)

[Hình 2.17: Biểu đồ trình tự chức năng thanh toán trực tuyến 44](#_Toc16925)

[Hình 2.18: Biểu đồ trình tự chức năng xem lịch sử bài đăng 45](#_Toc595)

[Hình 2.19: Biểu đồ lớp chức năng xem lịch sử bài đăng 45](#_Toc27361)

[Hình 2.20: Biểu đồ trình tự chức năng quản lý user 46](#_Toc11412)

[Hình 2.21: Biểu đồ lớp chức năng quản lý user 46](#_Toc20629)

[Hình 2.22: Biểu đồ trình tự chức năng đổi quyền user 47](#_Toc22453)

[Hình 2.23: Biểu đồ lớp chức năng đổi quyền user 47](#_Toc7573)

[Hình 2.24: Biểu đồ trình tự chức năng quản lý bài đăng 48](#_Toc23591)

[Hình 2.25: Biểu đồ lớp chức năng quản lý bài đăng 48](#_Toc18961)

[Hình 2.26: Biểu đồ miêu tả cấu trúc cơ sở dữ liệu 49](#_Toc13546)

[Hình 2.27: Bảng user 50](#_Toc585)

[Hình 2.28: Bảng boardinghouses 50](#_Toc11612)

[Hình 2.29: Bảng districts 50](#_Toc31681)

[Hình 2.30: Bảng houses 50](#_Toc17960)

[Hình 2.31: Bảng imageurls 50](#_Toc6372)

[Hình 2.32: Bảng messages 50](#_Toc21195)

[Hình 2.33: Bảng payments 51](#_Toc30449)

[Hình 2.34: Bảng lands 51](#_Toc7262)

[Hình 2.35: Bảng postsaves 51](#_Toc8234)

[Hình 2.36: Bảng provinces 51](#_Toc31612)

[Hình 2.37: Bảng Realestates 52](#_Toc11416)

[Hình 2.38: Màn hình chính 53](#_Toc10873)

[Hình 2.39: Giao diện màn hình đăng nhập 54](#_Toc11754)

[Hình 2.40: Giao diện màn hình đăng ký 55](#_Toc23117)

[Hình 3.1: Màn hình chức năng đăng nhập 56](#_Toc26414)

[Hình 3.2: Màn hình chức năng đăng ký 56](#_Toc6009)

[Hình 3.3: Màn hình tìm kiếm nhà trọ 57](#_Toc15507)

[Hình 3.4: Màn hình xem thông tin chi tiết 57](#_Toc5823)

[Hình 3.5: Màn hình đăng bài 58](#_Toc31808)

[Hình 3.6: Màn hình nhắn tin 59](#_Toc15968)

[Hình 3.7: Màn hình thanh toán bước 1 59](#_Toc29700)

[Hình 3.8: Màn hình thanh toán bước 2 60](#_Toc9022)

[Hình 3.9: Màn hình thanh toán thành công 60](#_Toc6287)

[Hình 3.10: Màn hình sửa thông tin cá nhân 61](#_Toc5458)

[Hình 3.11: Màn hình xem lịch sử bài đăng 61](#_Toc6101)

[Hình 3.12: Màn hình trước khi xóa user 62](#_Toc14619)

[Hình 3.13: Màn hình sau khi xóa user 63](#_Toc24907)

[Hình 3.14: Màn hình trước khi đổi quyền user 64](#_Toc29025)

[Hình 3.15: Màn hình sau khi đổi quyền user 65](#_Toc24863)

[Hình 3.16: Màn hình quản lý bài đăng 65](#_Toc17431)

Hình 3.17[: Kiểm thử chức năng phía người dùng 66](#_Toc3587)

# CƠ SỞ LÝ THUYẾT, CÔNG CỤ SỬ DỤNG VÀ KHẢO SÁT HỆ THỐNG

## Cơ sở lý thuyết

### Giới thiệu về Vue.js

**Vue.js** là một **framework JavaScript** được sử dụng để xây dựng giao diện người dùng (UI) và các ứng dụng web đơn trang (Single Page Applications - SPA). Nó được phát triển bởi **Evan You** vào năm 2014 và hiện nay là một trong những framework phổ biến nhất cho việc phát triển web front-end, cùng với React và Angular.

#### **Đặc điểm nổi bật của Vue.js**

**Reactive data binding**: Vue.js sử dụng cơ chế "two-way data binding" giúp tự động đồng bộ hóa dữ liệu giữa mô hình (model) và giao diện (view), giúp cho việc phát triển trở nên dễ dàng hơn.

**Component-based Architecture**: Vue.js cho phép chia ứng dụng thành các thành phần (component) độc lập, tái sử dụng được. Mỗi component là một khối chức năng UI tự chứa, giúp code dễ bảo trì và mở rộng.

**Virtual DOM**: Vue.js sử dụng Virtual DOM để tối ưu hóa quá trình cập nhật giao diện, giúp cho các thay đổi được thực hiện nhanh chóng và hiệu quả hơn.

**Ease of Integration**: Vue.js có thể dễ dàng tích hợp vào các dự án hiện có mà không cần phải viết lại toàn bộ ứng dụng, thích hợp cho cả dự án nhỏ và lớn.

#### **Lợi ích của Vue.js**

**Dễ học và sử dụng**: Vue.js có cú pháp đơn giản, dễ hiểu với các tài liệu phong phú, giúp các lập trình viên mới có thể tiếp cận nhanh chóng.

**Nhẹ và nhanh**: Kích thước nhỏ gọn của Vue.js (~30KB), giúp ứng dụng tải nhanh hơn, đặc biệt là trên các thiết bị có kết nối chậm.

**Khả năng mở rộng**: Vue.js có một hệ sinh thái mạnh mẽ với nhiều công cụ và thư viện mở rộng như Vue Router (để xử lý routing) và Vuex (quản lý trạng thái toàn cục của ứng dụng).

#### **Cách hoạt động cơ bản**

Vue.js hoạt động bằng cách liên kết dữ liệu với giao diện. Khi dữ liệu thay đổi, Vue sẽ tự động cập nhật lại giao diện tương ứng mà không cần phải thao tác thủ công.

#### **Hệ sinh thái của Vue.js**

**Vue CLI**: Công cụ dòng lệnh giúp khởi tạo và phát triển ứng dụng Vue.js nhanh chóng.

**Vue Router**: Thư viện quản lý định tuyến, giúp xây dựng các ứng dụng đơn trang dễ dàng.

**Vuex**: Thư viện quản lý trạng thái toàn cục, giúp đồng bộ hóa dữ liệu giữa các component.

**Nuxt.js**: Một framework xây dựng trên Vue.js, cung cấp nhiều tính năng bổ sung như server-side rendering (SSR) và static site generation (SSG).

#### **Kết luận**

Vue.js là một lựa chọn lý tưởng cho cả những người mới bắt đầu và các dự án lớn nhờ tính dễ học, tính linh hoạt và khả năng mở rộng mạnh mẽ. Với cộng đồng người dùng và hệ sinh thái phát triển liên tục, Vue.js là một trong những công cụ quan trọng cho phát triển ứng dụng web hiện đại.

### ASP.NET

ASP.NET là một nền tảng web được sử dụng để tạo các trang web, ứng dụng và dịch vụ web. Đây là sản phẩm tích hợp của các ngôn ngữ HTML, CSS và JavaScript. ASP.NET được ra đời vào năm 2002 và đến nay đã được nâng cấp thành nhiều phiên bản khác nhau.

ASP.NET hoạt động trên HTTP, sử dụng các lệnh và chính sách của HTTP để trình duyệt có thể trở thành giao tiếp song phương của máy chủ.

ASP.NET cung cấp những phát triển để tạo ứng dụng web như:

Web Forms: ASP.NET cung cấp tương tác hướng sự kiện cho các ứng dụng web, nó được sử dụng để phát triển ứng dụng có quyền truy cập dữ liệu. Đồng thời, kiểu phát triển web này cũng sẽ cung cấp dữ liệu cho máy chủ và sự kiện để tạo nên ứng dụng phù hợp. Web Forms được phát triển một cách nhanh chóng nhờ vào sử dụng thư viện điều khiển phong phú đánh dấu HTML.

ASP.NET MVC: cung cấp các mẫu MVC (Model View Controller) để xây dựng lên trang web động. Những mẫu này được sử dụng cho dữ liệu (Model), giao diện người dùng (View) và logic ứng dụng (Controller). Mô hình này sẽ đảm nhận nhiệm vụ duy trì dữ liệu và logic của ứng dụng. Chế độ xem sẽ là giao diện người dùng và hiển thị dữ liệu. Bộ điều khiển sẽ xử lý yêu cầu của người dùng về phần View và Model này.

ASP.NET web Pages: được sừ dụng để tạo ra các trang web động kết hợp máy chủ với HTML một cách nhanh chóng. ASP.NET web Pages sẽ đánh dấu mã của bạn và HTML cùng nhau trong cùng một tệp.



Hình 1.1: ASP.NET Core

#### Các thành phần của ASP.NET

Các thành phần chính của nền tảng ASP.NET cơ bản gồm:

Ngôn ngữ: ASP.NET sử dụng nhiều ngôn ngữ lập trình khác nhau như VB.NET và C#.

Thư viện: ASP.NET có bộ thư viện chuẩn bao gồm các giao diện, các lớp và kiểu giá trị. Bộ thư viện này có thể sử dụng lại cho quá trình phát triển ASP.NET và xây dựng các chức năng của hệ thống.

Thời gian chạy ngôn ngữ chung (CLR): CLR - Common Language Runtime được sử dụng để thực hiện các hoạt động mã. Các hoạt động này sẽ thực hiện xử lý các ngoại lệ và thu gom rác.

#### Tổng quan mô hình MVC

A diagram of a person

Description automatically generated with medium confidence

Hình 1.2: Mô hình Model-View-Controller

MVC (Model - View - Controller) là một design partern đã tồn tại rất lâu trong ngành công nghệ phần mềm. Một ứng dụng viết theo mô hình MVC sẽ bao gồm 3 thành phần tách biệt nhau đó là Model, View, Controller. Mô hình MVC giúp tách biệt 3 tầng trong mô hình lập trình web, vì vậy giúp tối ưu ứng dụng, dễ dàng thêm mới và chỉnh sửa code hoặc giao diện.

Model

* Có nhiệm vụ thao tác với Database
* Nó chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu
* Controller sẽ thông qua các hàm, phương thức đó để lấy dữ liệu rồi gửi qua View

View

* Là giao diện người dùng (User Interface)
* Chứa các thành phần tương tác với người dùng như menu, button, image, text, ...
* Nơi nhận dữ liệu từ Controller và hiển thị

Controller

* Là thành phần trung gian giữa Model và View
* Đảm nhận vai trò tiếp nhận yêu cầu từ người dùng, thông qua Model để lấy dữ liệu sau đó thông qua View để hiển thị cho người dùng.

#### Cơ chế hoạt động của ASP.NET

Ảnh có chứa biểu đồ

Mô tả được tạo tự động

Hình 1.3: Cơ chế hoạt động của ASP.NET

Dựa vào hình mô phỏng phía trên này thì ta cơ chế hoạt động của ASP.Net.

• User gửi 1 yêu cầu tới server bằng cách truyền vào 1 URL trong browser.

• Yêu cầu đó được gửi tới controller đầu tiên, controller sẽ xử lý yêu cầu, nếu yêu cầu cần truy xuất dữ liệu thì controller sẽ chuyển qua tầng model.

• Tại tầng model, dữ liệu được truy xuất từ database và sau đó truyền qua view thông qua controller.

• Controller sẽ giúp dữ liệu được chuyển từ model qua view

• View là tầng cuối cùng giao tiếp với User, mọi dữ liệu sẽ được hiển thị cho User thông qua tầng View.

### Entity Framework

Entity Framework ra đời từ 2008 cùng với .NET 3.5. Phiên bản hiện nay là 6.0. Có một số mốc đặc biệt trong lịch sử framework này. Phiên bản đầu tiên chỉ hỗ trợ hướng tiếp cận Database-first. Nghĩa là EF lúc đó chỉ có thể làm việc với một cơ sở dữ liệu sẵn có. EF giúp sinh code các lớp model bằng C# hoặc VB.NET. Đến phiên bản 4 xuất hiện hướng tiếp cận Model-first, giúp thiết kế các lớp thực thể trước sử dụng giao diện đồ họa. Phiên bản 4.1 đưa thêm hướng tiếp cận Code-first, nghĩa là từ giờ mọi thứ được viết bằng code VB.NET hoặc C# ngay từ đầu, một hướng tiếp cận được lập trình viên hoan nghênh đặc biệt.

Entity Framework là một ORM. ORM là viết tắt của Object – Relational Mapper (hay Mapping, tùy từng tình huống). ORM là một tập hợp các công nghệ cho phép làm việc với các hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System, hay RDBMS) từ các ngôn ngữ hướng đối tượng và không cần trực tiếp xử lý truy vấn SQL.

Các ORM như Entity Framework hỗ trợ người lập trình ánh xạ (hai chiều): class với cấu trúc bảng, object với bản ghi (hay hàng) trong bảng, thuộc tính (property) với cột của bảng, tập hợp object với tập hợp các bản ghi, tham chiếu sang object với quan hệ giữa các bảng. Trong quá trình này, tất cả các truy vấn SQL được ORM tự động sinh ra và thực thi. Người lập trình chỉ cần làm việc với các khái niệm quen thuộc của ngôn ngữ lập trình hướng đối tượng.

Trong quá trình ánh xạ này, ORM hỗ trợ tự động xử lý sự vênh nhau giữa ngôn ngữ lập trình hướng đối tượng và cơ sở dữ liệu quan hệ. Sự vênh nhau này bao gồm: vênh về kiểu giữa .NET và RDBMS, vênh về cách lưu trữ (giữa bảng và object trong bộ nhớ). Mối quan hệ phức tạp giữa các object và quan hệ giữa các bảng. Các vấn đề này thường gọi chung là impedance mismatch.

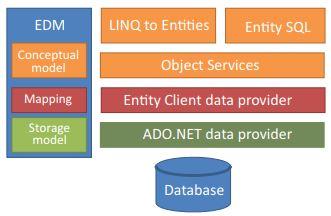
Entity Framework có đặc điểm gì?

• Entity Framework thể hiện cơ sở dữ liệu dưới dạng tập hợp của các object quen thuộc.

• Thứ hai, bạn có thể truy vấn dữ liệu sử dụng LINQ thay vì dùng SQL. Bạn có thể dễ dàng thực hiện các truy vấn CRUD hoàn toàn từ code C# với các class quen thuộc mà không cần viết dòng SQL nào.

• Thứ ba, bạn có thể thực hiện các thao tác với cấu trúc dữ liệu như tạo cơ sở dữ liệu, tạo bảng, thay đổi cấu trúc bảng một cách dễ dàng với công cụ Migration mà không làm mất dữ liệu.

#### Kiến trúc Entity Framework



Hình 1.4: Kiến trúc của Entity Framework

Như trong sơ đồ trên ta thấy Entity Framework phân chia làm các thành phần: Entity Data Model (EDM), Entity Client data provider, Object Services, LINQ to Entities và Entity SQL.

**Entity Data Model (EDM)**

EDM là thành phần lưu trữ thông tin ánh xạ (mapping) giữa các class và cơ sở dữ liệu. Nó đóng vai trò cầu nối giữa các lớp model và cơ sở dữ liệu nhưng lại giúp chương trình hoàn toàn tách biệt với cơ sở dữ liệu.

Cơ sở dữ liệu và các lớp model của chương trình là hai thứ khác nhau hoàn toàn về bản chất và cấu trúc. Nhờ có EDM, hai thứ khác nhau này được liên kết với nhau. Nếu bạn thay một cơ sở dữ liệu khác, EDM lại giúp bạn tạo cầu nối mới mà không phải thay đổi các lớp model. Như vậy chương trình không phụ thuộc vào cơ sở dữ liệu.

**EDM có 3 thành phần:**

• Conceptual model: chịu trách nhiệm lưu thông tin về các lớp model và quan hệ giữa chúng.

• Storage model: lưu thông tin về cơ sở dữ liệu, bao gồm các bảng, view, stored procedure, khóa, quan hệ.

• Mapping model: lưu thông tin ánh xạ giữa conceptual và storage model.

Thông tin từ các file này được sử dụng để cung cấp metadata cho các thành phần còn lại. EDM hầu như được tạo tự động. Tuy nhiên, vẫn có thể tự thay đổi cho phù hợp với yêu cầu của dự án.

**LINQ to Entities và Entity SQL**

Đây là hai ngôn ngữ dùng để truy vấn dữ liệu. LINQ to Entities được xem như là một “phương ngữ” của LINQ. Không phải mọi thứ trong LINQ đều sử dụng được trong LINQ to Entities.

Object services

Tầng này thực hiện một quá trình gọi là materialization – quá trình chuyển đổi dữ liệu dạng bảng nhận từ tầng dưới (Entity Client data provider) về dạng object. Lưu ý rằng, dữ liệu dạng bảng nhận từ tầng dưới không phải là dữ liệu dạng bảng thu trực tiếp từ cơ sở dữ liệu (gọi là database tabular structure) mà là một dạng trung gian có tên gọi là model tabular structure.

**Entity Client data provider**

Tầng này chịu trách nhiệm tương tác với Data provider của ADO.NET. Tầng này thực hiện chuyển đổi truy vấn LINQ to Entities hoặc Entity SQL về truy vấn SQL. Sau đó Data provider sẽ thực thi truy vấn SQL. Data provider chịu trách nhiệm tương tác về mặt vật lý với cơ sở dữ liệu.

**Các hướng tiếp cận khi xây dựng EDM**

Ảnh có chứa biểu đồ

Mô tả được tạo tự động

Hình 1.5: Database first

Database-first là hướng tiếp cận xuất hiện đầu tiên trong Entity Framework.

Hướng tiếp cận này sẽ giúp sinh ra model từ cơ sở dữ liệu có sẵn từ stored procedures hoặc views. Chỉ cần tinh chỉnh lại model (domain classes) theo nhu cầu của mình là xong. Việc tinh chỉnh này thực hiện qua một giao diện đồ họa (visual designer) rất tiện lợi. Nói cách khác, xuất phát từ Storage Model có sẵn, EF sẽ giúp sinh ra Conceptual Model và Mapping.

Ảnh có chứa biểu đồ

Mô tả được tạo tự động

Hình 1.6: Model-first

Model-first: Đây là hướng tiếp cận thứ hai xuất hiện từ EF4

Hướng tiếp cận này cho phép bạn sử dụng một giao diện đồ họa để thiết kế model trước. Sau đó trình thiết kế này sẽ sinh ra các lớp model tương ứng cũng như mã SQL để tạo cơ sở dữ liệu. Nói cách khác, bạn xuất phát từ một giao diện thiết kế riêng biệt, EF sẽ giúp bạn sinh ra cả ba thành phần của EDM.

Ảnh có chứa biểu đồ

Mô tả được tạo tự động

Hình 1.7: Code first

Code-first: Đây là hướng tiếp cận được lập trình viên hoan nghênh nhất. Code-first xuất hiện từ EF4.1.

Với vai trò lập trình viên, không cần quan tâm đến SQL hay giao diện thiết kế đồ họa nữa. Tất cả những gì cần làm là xây dựng các lớp model (domain classes) theo đúng kiểu hướng đối tượng mà bạn quen thuộc nhất. Entity Framework sẽ làm tất cả những gì còn lại để bạn có một cơ sở dữ liệu phù hợp. Nói theo kiểu khác, tự mình code Conceptual Model, EF sẽ giúp bạn sinh ra Mapping và Storage Model.

Code-first from Database: Đây là một cách tiếp cận khá khác biệt. Xuất phát từ một cơ sở dữ liệu có sẵn, EF giúp bạn sinh ra Conceptual Model và Mapping. Tuy nhiên, cái thu được chỉ là code (của các domain class) chứ không có giao diện thiết kế (đồ họa) như kiểu tiếp cận Database-first.

### My SQL

**MySQL** là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS) mã nguồn mở, được sử dụng rộng rãi trên toàn thế giới để lưu trữ, quản lý và truy vấn dữ liệu. MySQL được phát triển lần đầu vào năm 1995 bởi **MySQL AB**, sau đó được Oracle Corporation mua lại vào năm 2010.

#### **Đặc điểm nổi bật của MySQL**

**Cơ sở dữ liệu quan hệ**: MySQL lưu trữ dữ liệu trong các bảng (table) theo mô hình quan hệ, mỗi bảng chứa các bản ghi (row) và các trường (column). Điều này giúp dữ liệu được tổ chức và truy vấn một cách dễ dàng.

**SQL (Structured Query Language)**: MySQL sử dụng ngôn ngữ SQL để thao tác với dữ liệu. SQL là ngôn ngữ tiêu chuẩn để truy vấn, cập nhật và quản lý cơ sở dữ liệu quan hệ.

**Mã nguồn mở**: MySQL là phần mềm mã nguồn mở, nghĩa là người dùng có thể tải về, chỉnh sửa và phân phối mà không phải trả phí. Tuy nhiên, Oracle cũng cung cấp các phiên bản thương mại với các tính năng nâng cao.

**Hiệu suất cao và độ ổn định**: MySQL nổi tiếng với khả năng xử lý các tác vụ truy vấn dữ liệu lớn với hiệu suất cao và độ ổn định tốt, điều này làm cho MySQL trở thành lựa chọn phổ biến cho các ứng dụng web và dịch vụ lớn.

#### **Lợi ích của MySQL**

**Dễ sử dụng**: MySQL có cú pháp đơn giản và dễ tiếp cận với những người mới bắt đầu học về cơ sở dữ liệu.

**Khả năng mở rộng**: MySQL có thể xử lý các cơ sở dữ liệu nhỏ cho đến các ứng dụng lớn với lượng dữ liệu và số lượng truy vấn lớn.

**Hỗ trợ nhiều nền tảng**: MySQL có thể hoạt động trên nhiều hệ điều hành như Windows, Linux, macOS, và được tích hợp tốt với nhiều ngôn ngữ lập trình như PHP, Java, Python, C#.

**Cộng đồng phát triển lớn**: Với cộng đồng người dùng đông đảo, MySQL có một nguồn tài liệu và hỗ trợ phong phú từ các diễn đàn, tài liệu hướng dẫn và các bài viết kỹ thuật.

#### **Cấu trúc cơ bản của MySQL**

MySQL sử dụng mô hình cơ sở dữ liệu quan hệ, trong đó:

**Database**: Tập hợp các bảng chứa dữ liệu liên quan.

**Table**: Một bảng trong cơ sở dữ liệu chứa các bản ghi và được chia thành các hàng (row) và cột (column).

**Row**: Một bản ghi dữ liệu cụ thể trong bảng.

**Column**: Một thuộc tính cụ thể của bảng.

#### **Các tính năng chính**

**Data Security**: MySQL cung cấp cơ chế bảo mật mạnh mẽ với các tính năng như xác thực người dùng, phân quyền truy cập, và mã hóa dữ liệu.

**Replication (Nhân bản dữ liệu)**: MySQL hỗ trợ nhân bản cơ sở dữ liệu, cho phép sao chép dữ liệu từ một server chính (master) sang các server phụ (slave), giúp cải thiện hiệu suất và đảm bảo tính sẵn sàng của dữ liệu.

**High Availability (Tính sẵn sàng cao)**: MySQL có thể được cấu hình với các kiến trúc phân tán và cân bằng tải để đảm bảo tính khả dụng cao cho các ứng dụng quan trọng.

**Stored Procedures & Triggers**: MySQL hỗ trợ các thủ tục lưu trữ và trigger, giúp thực hiện các hành động tự động dựa trên các sự kiện xảy ra trong cơ sở dữ liệu.

**Full-text Search**: MySQL cung cấp tính năng tìm kiếm toàn văn bản (full-text search), giúp cải thiện hiệu suất tìm kiếm trong các ứng dụng lớn.

#### **Các ứng dụng phổ biến của MySQL**

**Hệ thống quản lý nội dung (CMS)**: Nhiều hệ thống CMS phổ biến như WordPress, Joomla và Drupal sử dụng MySQL làm cơ sở dữ liệu.

**Thương mại điện tử**: Các nền tảng thương mại điện tử như Magento và WooCommerce thường tích hợp MySQL để quản lý sản phẩm, đơn hàng và thông tin khách hàng.

**Web Application**: MySQL thường được sử dụng cùng với ngôn ngữ lập trình PHP để xây dựng các ứng dụng web như diễn đàn, blog, và các hệ thống quản lý dữ liệu.

#### **Kết luận**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ phổ biến, mạnh mẽ và linh hoạt. Nó thích hợp cho nhiều loại dự án, từ các ứng dụng nhỏ đến các hệ thống lớn, yêu cầu xử lý dữ liệu nhanh và an toàn. Nhờ tính dễ sử dụng, hiệu suất cao và khả năng mở rộng, MySQL đã trở thành lựa chọn hàng đầu cho các nhà phát triển và các công ty trên toàn thế giới.

## Công cụ sử dụng

* Visual Studio Code: thiết kết API.
* Visual Code: thiết kế giao diện.

Rational Rose: thiết kế hệ thống.

## Khảo sát hệ thống

### Tên đề tài

**Xây dựng Website cung cấp dịch vụ cho thuê nhà trọ trên địa bàn Hà Nội.**

### Lý do chọn đề tài

Với sự phát triển nhanh chóng của đô thị và gia tăng số lượng sinh viên, người đi làm tại Hà Nội, nhu cầu tìm kiếm nhà trọ ngày càng cao. Ngoài ra Hà Nội là một thành phố lớn với dân số đông và đa dạng, từ sinh viên đến nhân viên văn phòng và người lao động, tất cả đều cần tìm nơi ở.

Việc phát triển một website cho dịch vụ cho thuê nhà trọ tại Hà Nội không chỉ đáp ứng nhu cầu cao trong thị trường hiện tại mà còn mang lại nhiều lợi ích đáng kể. Hà Nội, với sự gia tăng dân số và nhu cầu thuê nhà ngày càng lớn từ sinh viên và người đi làm, tạo điều kiện thuận lợi cho sự phát triển của một nền tảng trực tuyến. Website sẽ giúp người dùng dễ dàng tìm kiếm và lọc các lựa chọn nhà trọ theo nhu cầu cá nhân, đồng thời cung cấp thông tin chi tiết và chính xác để họ có thể đưa ra quyết định nhanh chóng và thông minh. Chủ nhà cũng sẽ được hưởng lợi từ việc quản lý tin đăng dễ dàng và tiếp cận đối tượng khách hàng rộng lớn hơn. Thêm vào đó, việc tích hợp công nghệ mới và tự động hóa quy trình sẽ nâng cao hiệu quả và cải thiện trải nghiệm người dùng, mở ra cơ hội kinh doanh mới từ quảng cáo và dịch vụ liên quan. Tóm lại, việc xây dựng website này không chỉ đáp ứng nhu cầu thị trường mà còn tạo ra cơ hội kinh doanh và nâng cao chất lượng dịch vụ cho thuê nhà trọ tại Hà Nội.

Nhận thức được sự quan trọng và cũng như để có thể áp dụng được những kiến thức đã được học và tìm hiểu, em xin được áp dụng những kiến thức đã được học và tìm hiểu đó để xây dựng một website cụ thể là: ***“Xây dựng Website cung cấp dịch vụ cho thuê nhà trọ trên địa bàn Hà Nội”.***

Đề tài: ***Xây dựng Website cung cấp dịch vụ cho thuê nhà trọ trên địa bàn Hà Nội*** đáp ứng được những mục tiêu:

* Hiểu biết những kiến thức cần thiết về xây dựng giao diện website, phân tích thiết kế hệ thống, cơ sở dữ liệu.
* Nắm được các kiến thức cơ bản về xây dựng giao diện và thiết kế hệ thống.
* Dễ sử dụng và thân thiện với người dùng: Website được thiết kế với giao diện đơn giản, dễ hiểu và dễ sử dụng, đảm bảo người dùng có thể tiếp cận và tương tác với nó một cách thuận tiện và không gặp khó khăn.
* Ứng dụng cung cấp tính năng kết nối trực tiếp giữa người đăng tin với người có nhu cầu thuê trọ qua tin nhắn, có thể trả phí bài đăng trực tuyến, giúp người dùng giải quyết vấn đề một cách nhanh chóng và thuận tiện.

### Đối tượng và phạm vi

Mọi người đều có thể sử dụng.

### Kết quả dự kiến đạt được

Ứng dụng bao gồm bao gồm những mục chính và chức năng chính:

* Đăng ký, đăng nhập tài khoản cá nhân.
* Thay đổi mật khẩu, cập nhật thông tin tài khoản.
* Tìm kiếm, lọc kết quả mà mình mong muốn.
* Xem thông tin bài đăng chi tiết.
* Liên hệ với người đăng bài qua tin nhắn trực tuyến.
* Thanh toán trực tuyến qua vnpay.

# PHÂN TÍCH VÀ THIẾT KẾT HỆ THỐNG

## Phân tích hệ thống

### Vai trò của người dùng

* Hệ thống chia làm 2 nhóm người dùng chính:
* User: là người có nhu cầu thuê trọ hoặc cần cho thuê trọ.
* Admin: người có quyền quản lý những bài đăng và đổi quyền những user khác.
* Mô tả nhóm người dùng.
* User: những người dùng đã có tài khoản đăng nhập trên website. Người dùng có thể tìm kiếm các thông tin bài đăng về phòng trọ hoặc có thể đăng bài về phòng trọ.
* Admin: là người có quyền quản lý giám sát những bài đăng và người dùng.

### Yêu cầu chức năng

User:

* Đăng ký
* Đăng nhập
* Đăng xuất
* Tìm kiếm nhà trọ
* Xem thông tin nhà trọ
* Đăng bài
* Nhắn tin
* Thanh toán trực tuyến
* Chỉnh sửa thông tin cá nhân
* Xem lịch sử đăng bài

Admin:

* Đăng nhập
* Sửa thông tin cá nhân
* Quản lý người dùng
* Đổi quyền người dùng
* Quản lý bài đăng
* Đăng xuất

### Yêu cầu phi chức năng

* Giao diện ứng dụng được thiết kế dễ sử dụng và có tính thẩm mỹ cao.
* Hiệu năng: tốc độ, khả năng và độ tin cậy của hệ thống.
* Website bảo mật phân quyền để thực hiện các chức năng của hệ thống.

## Xây dựng mô hình use case

### Xác định Actor và use case

#### Xác định Actor

Website có các Actor sau: Quản trị viên, Khách hàng.

#### Xác định use case

|  |  |
| --- | --- |
| Actor | Use Case |
| Quản trị viên(admin) | Đăng nhập, quản lý user, đổi quyền user và quản lý bài đăng. |
| Khách hàng(User) | Đăng nhập, đăng ký, tìm kiếm nhà trọ, đăng bài, xem thông tin chi tiết bài đăng, nhắn tin, thanh toán trực tuyến bài dăng, xem lịch sử đăng bài và chỉnh sửa thông tin cá nhân. |

### Biểu đồ use case

#### Biểu đồ use case phía Admin



Hình 2.1: Biểu đồ use case phía Admin

#### Biểu đồ use case phía người dùng



Hình 2.2: Biểu đồ use case phía người dùng

### Đặc tả use case

#### Use case Đăng nhập

* Đặc tả use case:

Use case này cho phép user đăng nhập để xác định quyền truy cập vào hệ thống.

* Luồng sự kiện:
* Luồng cơ bản:
* Use case này bắt đầu khi người dùng bấm vào nút “Đăng nhập” màn hình.
* Hệ thống yêu cầu người dùng nhập các thông tin bao gồm: Email và mật khẩu.
* Người dùng nhập các thông tin: email và mật khẩu và bấm nút “Đăng Nhập”.
* Use case kết thúc.
* Luồng rẽ nhánh:
* Sai email hoặc mật khẩu: tại bước 3 trong luồng cơ bản nếu người dùng nhập Email hoặc mật khẩu sai, hệ thống hiển thị thông báo lỗi. Người dùng có thể nhập lại hoặc bỏ qua thao tác, khi đó use case kết thúc.
* Đăng ký: tại bước 3 trong luồng cơ bản, nếu người dùng bấm nút “Tạo tài khoản” use case kết thúc.
* Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.
* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Nếu use case thành công, người dùng sẽ đăng nhập được vào hệ thống. Nếu không trạng thái của hệ thống không thay đổi.

* Điểm mở rộng:

Không có.

#### Use case Đăng ký

* Đặc tả use case:

Use case cho phép user đăng ký tài khoản trên hệ thống.

* Luồng sự kiện:
* Luồng cơ bản:
* Use case này bắt đầu khi user hàng kích vào nút “Tạo tài khoản mới” trên màn hình đăng nhập. Hệ thống sẽ hiển thị màn hình đăng ký.
* Người dùng nhập thông tin trên form đăng ký gồm Email, Họ và tên, Số điện thoai và mật khẩu rồi kích nút “Đăng ký. Hệ thống thêm một bản ghi mới vào trong bảng User trong cơ sở dữ liệu.
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bước 2 trong luồng cơ bản, khi nhập thông tin không thoả mãn, hệ thống thông báo lỗi. Use case kết thúc.
* Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.
* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

#### Use case Tìm nhà trọ

* Đặc tả use case:

Use case cho phép user tìm kiếm nhà trọ phù hợp theo từng khu vực.

* Luồng sự kiện:
* Luồng cơ bản:
* Use case này bắt đầu khi user click vào nút “Lọc”. Hệ thống hiển thị bảng thông tin gồm những thông tin như loại bài đăng, loại hình cho thuê, khoảng giá, diện tích.
* Người dùng nhập thông tin mà mình cần tìm kiếm và click vào nút “Lọc”. Hệ thống truy xuất đến các bảng realestates. Dựa vào thông tin người nhập hiển thi ra bản ghi,
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bước 1 nếu không có dữ liệu nào phù hợp hệ thống hiển thị thông báo không tìm thấy bài đăng nào và use case kết thúc.
* Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

#### Use case Xem thông tin bài đăng

* Đặc tả use case:

Use case cho phép user xem thông tin chi tiết về bài đăng.

* Luồng sự kiện:
* Luồng cơ bản:
* Use case này bắt đầu khi user kích vào bài đăng. Hệ thống sẽ lấy các thông tin về bài đăng bao gồm OwnerID, DistricID, Address,... dựa vào các bảng user, districs, realestates và hiển thị lên màn hình.
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

#### Use case Sửa thông tin cá nhân

* Đặc tả use case:

Use case cho phép user sửa thông tin cá nhân trong tài khoản.

* Luồng sự kiện:
* Luồng cơ bản:
* Use case này bắt đầu khi user kích vào mục “Chỉnh sửa thông tin” trên góc phải màn hình. Hệ thống sẽ hiển thị màn hình chỉnh sửa thông tin cá nhân.
* User nhập thông tin cần chỉnh sửa rồi kích nút cập nhật hồ sơ. Hệ thống cập nhật lại thông tin trên bảng User trong cơ sở dữ liệu.
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bước 2 ở luồng cơ bản, nếu user nhập sai định dạng, hệ thống thông báo lỗi. Use case kết thúc.
* Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

User đăng nhập thành công.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

#### User case Đăng bài

* Đặc tả use case:

Use case cho user đăng bài cho thuê hoặc mua bán.

* Luồng sự kiện:
* Luồng cơ bản:
* Use case này bắt đầu khi user click vào nút “Đăng tin” trong màn hình hiển thị.
* Hệ thống hiển thị form thông tin. User chọn loại hình cần đăng và thông tin về bài đăng rồi sau đó click nút “Đăng tin”.
* Hệ thống hiển thị form thanh toán giao dịch gồm phí bài đăng, mã bài đăng, phương thức thanh toán, ngôn ngữ. Người dùng chọn phương thức thanh toán rồi sau đó click vào nút thanh toán.
* Hệ thống hiển thị view thanh toán thành công. Hệ thống hiển thị thông báo thanh toán thành công và hệ thống tạo bảng ghi mới trong bảng payments.
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bước 1 trong luồng cơ bản, khi user đăng bài mà điền thiếu thông tin thì hệ thống sẽ yêu cầu bổ sung hoặc kết thúc.
* Tại bước 3 trong luồng cơ bản, khi người dùng bấm nút hủy use case kết thúc.
* Tại bước 4 trong luồng cơ bản, khi người dùng ấn nút Cancel hệ thống sẽ thông báo Payment Failed và use case kết thúc
* Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

#### Use case Nhắn tin

* Đặc tả use case:

Use case cho phép người dùng có thể liên hệ với người có nhu cầu cho thuê.

* Luồng sự kiện:
* Luồng cơ bản:
* Use case này bắt đầu khi user kích vào nút “Nhắn tin cho người đăng”. Hệ thống hiển thị màn hình tin nhắn.
* Người dùng nhập câu hỏi muốn hỏi cho người đăng bài. Sau đo click vào nút “gửi”. Hệ thống sẽ tạo một bản ghi chứa nội dung tin nhắn trong bảng MESSAGE và hiển thị nội dung tin nhắn.
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

#### Use case Thanh toán trực tuyến

* Đặc tả use case:

Use case cho phép user thanh toán trực tuyến bài đăng qua vnpay.

* Luồng sự kiện:
* Luồng cơ bản:
* Use case này bắt đầu khi User click vào phần đăng tin. Hệ thống hiển thị form để user nhập vào bài đăng.
* User sẽ điền đầy đủ thông tin về bài đăng và ấn vào nút đăng tin. Hệ thống sẽ chuyển tiếp sang phần thanh toán.
* User chọn phương thức thanh toán và ấn vào nút thanh toán.
* Hệ thống hiển thị bảng phương thức thanh toán cho người dùng chọn.User chọn phương thức thanh toán bằng thẻ rồi nhập thông tin thanh toán và ấn vào nút thanh toán
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại 1 User nhập thiếu thông tin bài đăng use case kết thúc.
* Tại 2 User bấm hủy và chuyển về màn hình trang chủ use case kết thúc
* Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

#### Use case Xem lịch sử bài đăng

* Đặc tả use case:

Use case này cho phép người dùng xem những bài đăng đã tạo.

* Luồng sự kiện:
* Luồng cơ bản:
* Use case này bắt đầu khi người dùng vào nút “Xem lịch sử bài đăng” trong màn hình góc trên bên tay phải.
* Hệ thống hiển thị trang gồm những bài đăng mà người dùng đã đăng.
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bước 1, Nếu user chưa có bài đăng nào thì hệ thống sẽ hiển thị thông báo “Bài đã đăng trống” use case kết thúc.
* Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.
* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có

* Điểm mở rộng:

Không có.

#### Use case Quản lý user

* Đặc tả use case:

Use case này cho phép quản trị viên xóa những người dùng .

* Luồng sự kiện:
* Luồng cơ bản:
* Use case này bắt đầu khi người dùng vào nút “người dùng” trong màn hình góc trên bên tay trái.
* Hệ thống hiển thị trang gồm những người dùng. Quản trị viên click vào nút “xóa” để xóa người dùng.
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bước 2, Nếu quản trị viên xóa người có vai trò là quản trị viên thì hệ thống sẽ báo lỗi và use case kết thúc.
* Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.
* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có

* Điểm mở rộng:

Không có.

#### Use case Đổi quyền user

* Đặc tả use case:

Use case này cho phép quản trị viên đổi quyền của những người dùng .

* Luồng sự kiện:
* Luồng cơ bản:
* Use case này bắt đầu khi người dùng vào nút “người dùng” trong màn hình góc trên bên tay trái.
* Hệ thống hiển thị trang gồm những người dùng. Quản trị viên click vào nút “đổi quyền người dùng” để thay đổi quyền của người dùng.
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.
* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có

* Điểm mở rộng:

Không có.

#### Use case Quản lý bài đăng

* Đặc tả use case:

Use case này cho phép quản trị viên xem được những bài đã đăng.

* Luồng sự kiện:
* Luồng cơ bản:
* Use case này bắt đầu khi người dùng vào nút “bài đăng” trong màn hình góc trên bên tay trái.
* Hệ thống hiển thị trang gồm những bài đã đăng. Quản trị viên click vào bài đăng để xem thông tin chi tiết về bài đăng.
* Use case kết thúc.
* Luồng rẽ nhánh:
* Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.
* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

* Hậu điều kiện:

Không có

* Điểm mở rộng:

Không có.

## Phân tích use case

### Use case Đăng nhập

#### Biểu đồ trình tự



Hình 2.3: Biểu đồ trình tự chức năng đăng nhập

#### Biểu đồ lớp



Hình 2.4: Biểu đồ lớp chức năng đăng nhập

### Use case Đăng ký

#### Biểu đồ trình tự



Hình 2.5: Biểu đồ trình tự chức năng đăng ký

#### Biểu đồ lớp



Hình 2.6: Biểu đồ lớp chức năng đăng nhập

### Use case Tìm nhà trọ

#### Biểu đồ trình tự



Hình 2.7: Biểu đồ trình tự chức năng tìm kiếm nhà trọ

#### Biểu đồ lớp



Hình 2.8: Biểu đồ lớp chức năng tìm kiếm nhà trọ

### Use case Xem thông tin bài đăng

#### Biểu đồ trình tự



Hình 2.9: Biểu đồ trình tự chức năng xem thông tin bài đăng

#### Biểu đồ lớp



Hình 2.10: Biều đồ lớp chức năng xem thông tin bài đăng

### Use case Sửa thông tin cá nhân

#### Biểu đồ trình tự



Hình 2.11: Biểu đồ trình tự của chức năng sửa thông tin cá nhân

#### Biểu đồ lớp



Hình 2.12: Biểu đồ lớp chức năng sửa thông tin cá nhân

### Use case Đăng bài

#### Biểu đồ trình tự



Hình 2.13: Biểu đồ trình tự chức năng đăng bài

#### Biểu đồ lớp



Hình 2.14: Biểu đồ lớp chức năng đăng bài

### Use case Nhắn tin

#### Biểu đồ trình tự



Hình 2.15: Biểu đồ trình tự chức năng nhắn tin

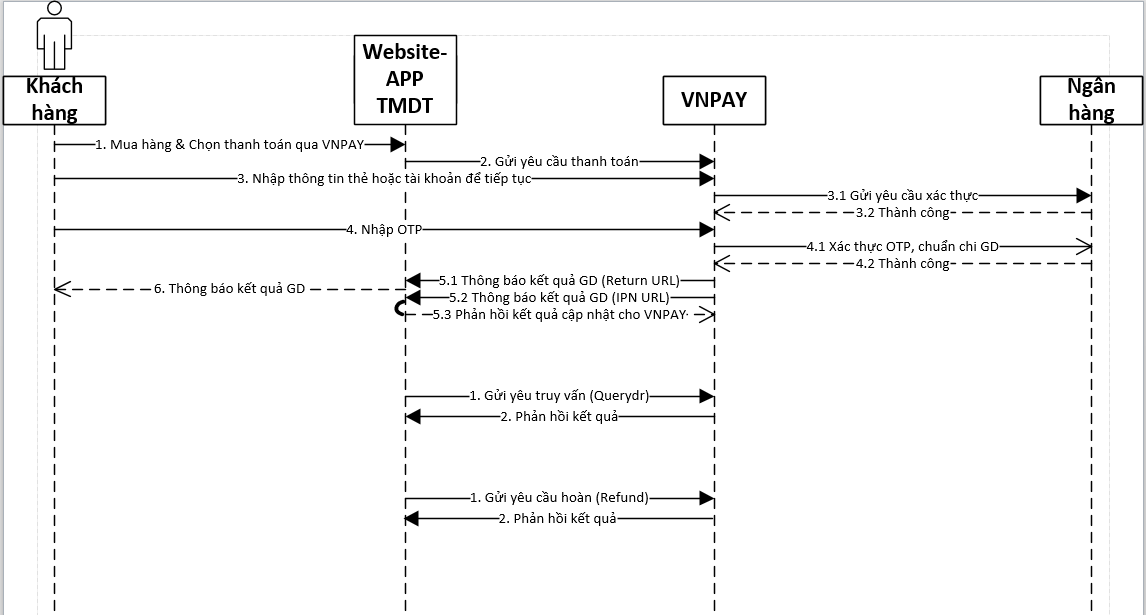
#### Biểu đồ lớp



Hình 2.16: Biểu đồ lớp chức năng nhắn tin

### Use case Thanh toán trực truyến

#### Biểu đồ trình tự



Hình 2.17: Biểu đồ trình tự chức năng thanh toán trực tuyến

### Use case Xem lịch sử bài đăng

#### Biểu đồ trình tự



Hình 2.18: Biểu đồ trình tự chức năng xem lịch sử bài đăng

#### Biểu đồ lớp



Hình 2.19: Biểu đồ lớp chức năng xem lịch sử bài đăng

### Use case Quản lý user

#### Biểu đồ trình tự



Hình 2.20: Biểu đồ trình tự chức năng quản lý user

#### Biểu đồ lớp



Hình 2.21: Biểu đồ lớp chức năng quản lý user

### Use case Đổi quyền user

#### Biểu đồ trình tự



Hình 2.22: Biểu đồ trình tự chức năng đổi quyền user

#### Biều đồ lớp



Hình 2.23: Biểu đồ lớp chức năng đổi quyền user

### Use case Quản lý bài đăng

#### Biểu đồ trình tự



Hình 2.24: Biểu đồ trình tự chức năng quản lý bài đăng

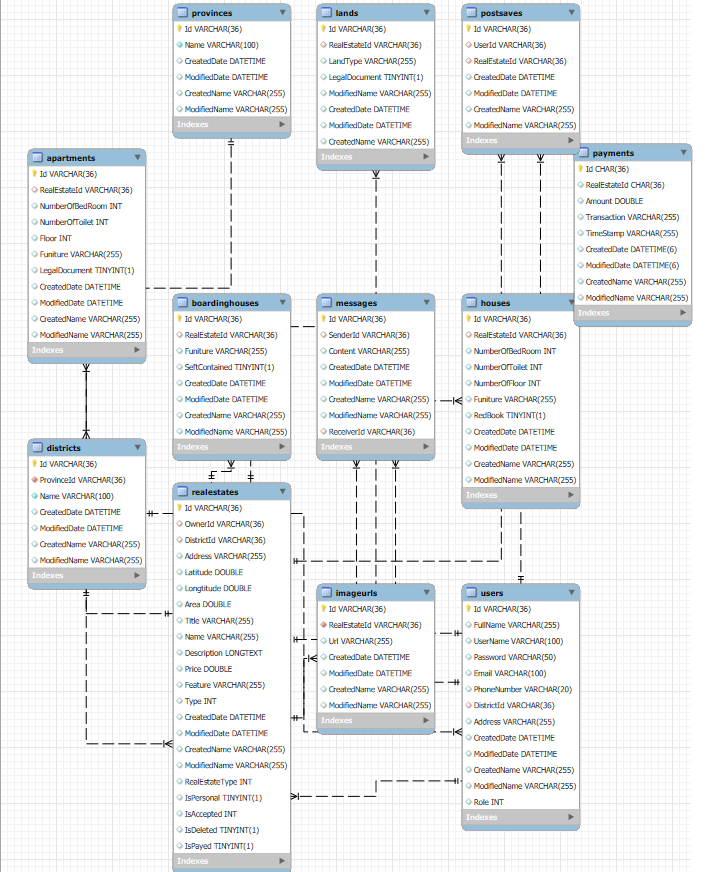
#### Biểu đồ lớp



Hình 2.25: Biểu đồ lớp chức năng quản lý bài đăng

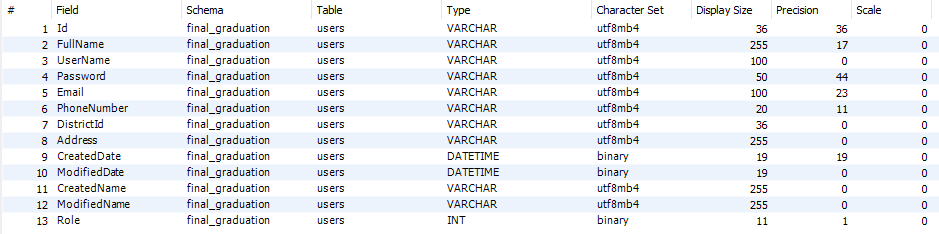
## Thiết kế cơ sở dữ liệu

### Biểu đồ mô tả cấu trúc cơ sở dữ liệu

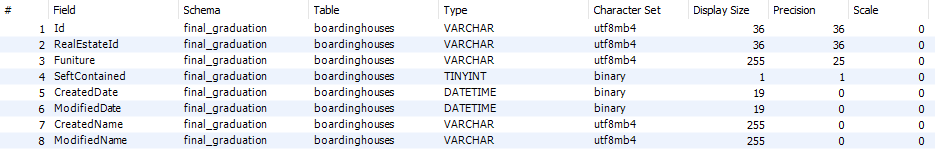


Hình 2.26: Biểu đồ miêu tả cấu trúc cơ sở dữ liệu

### Mô tả cơ sở dữ liệu

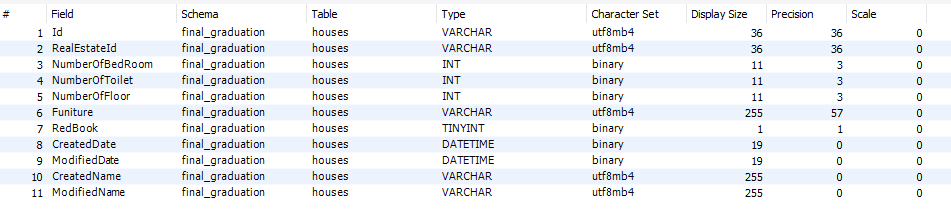


Hình 2.27: Bảng user

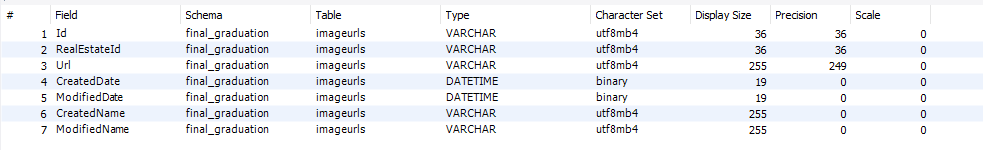


Hình 2.28: Bảng boardinghouses

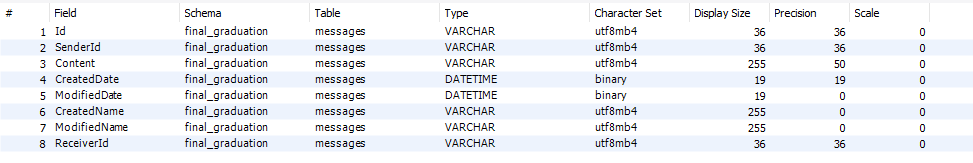
Hình 2.29: Bảng districts



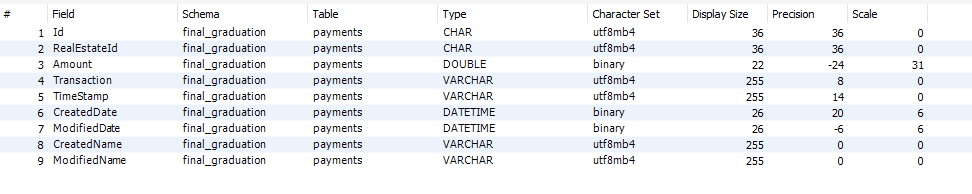
Hình 2.30: Bảng houses



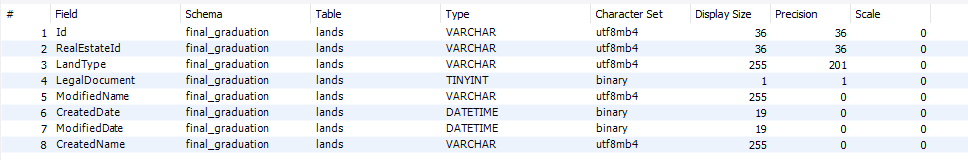
Hình 2.31: Bảng imageurls



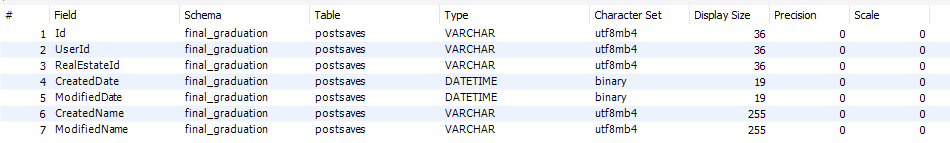
Hình 2.32: Bảng messages



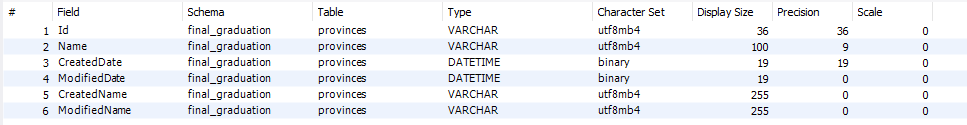
Hình 2.33: Bảng payments



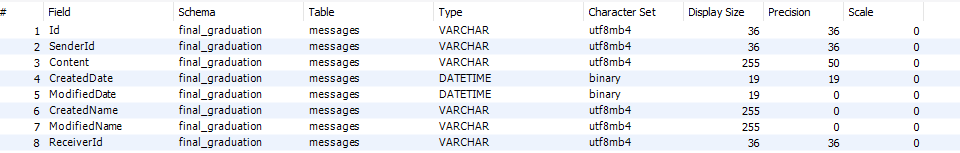
Hình 2.34: Bảng lands

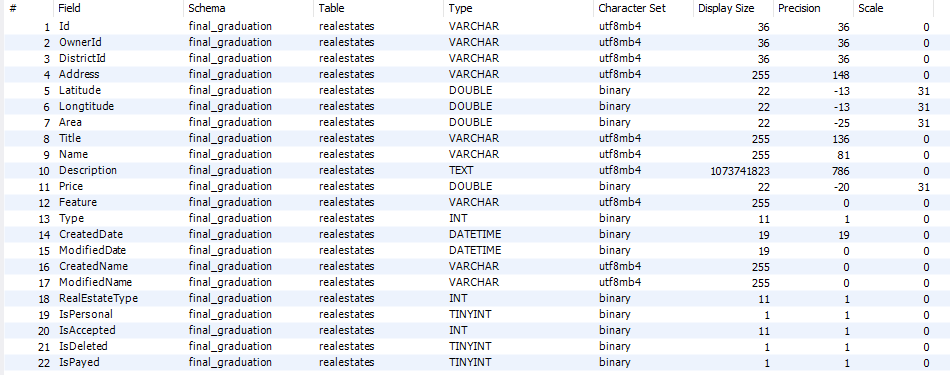


Hình 2.35: Bảng postsaves



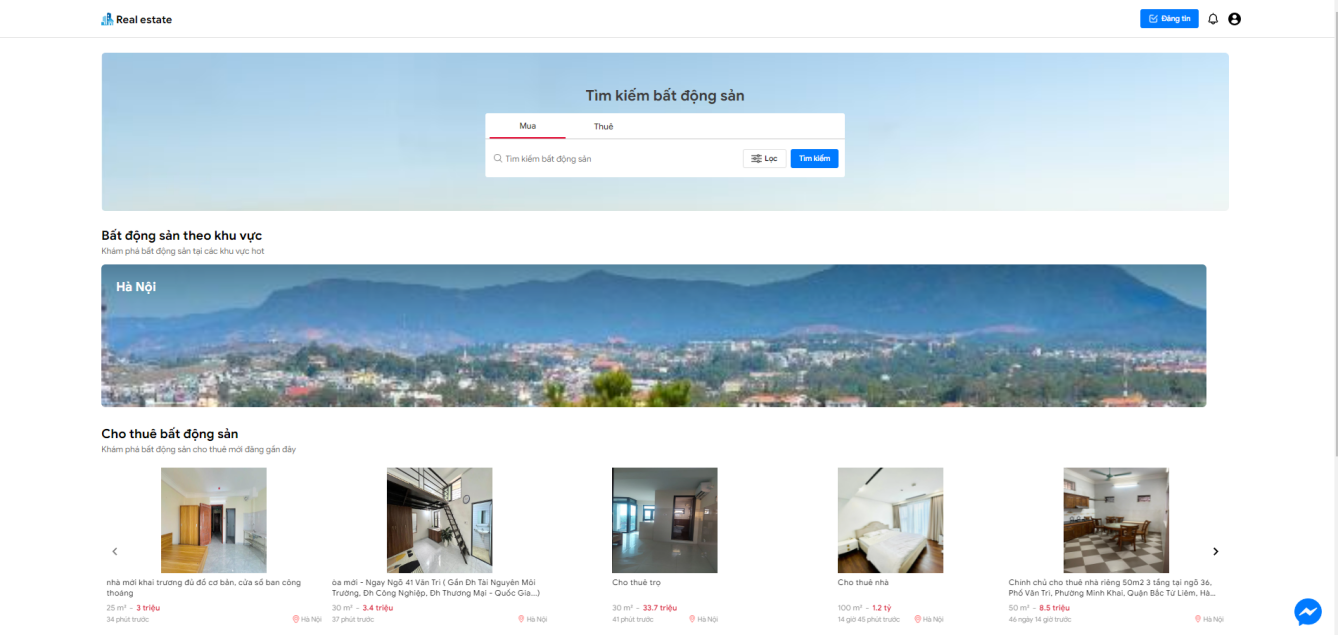
Hình 2.36: Bảng provinces

Hình 2:37*: Bảng message*

Hình 2.38: Bảng Realestates

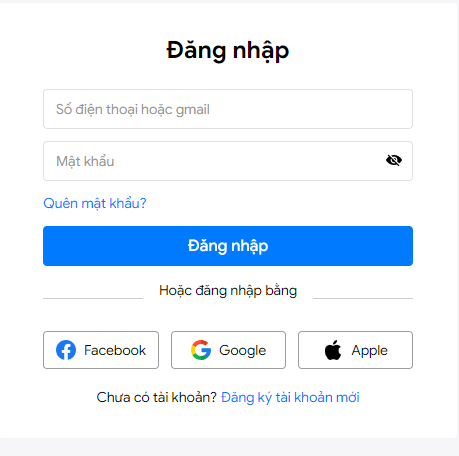
## Thiết kế giao diện

### Màn hình chính



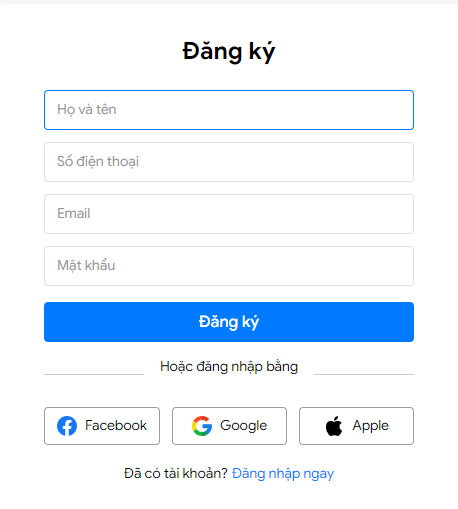
Hình 2.39: Màn hình chính

### Giao diện đăng nhập

**

Hình 2.40: Giao diện màn hình đăng nhập

### Giao diện đăng ký

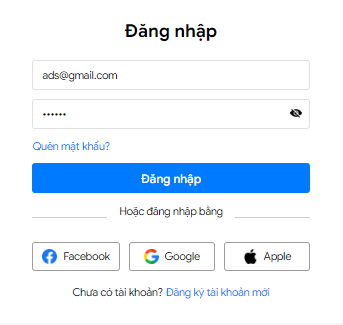
**

Hình 2.41: Giao diện màn hình đăng ký

# KẾT QUẢ, CÀI ĐẶT VÀ KIỂM THỬ

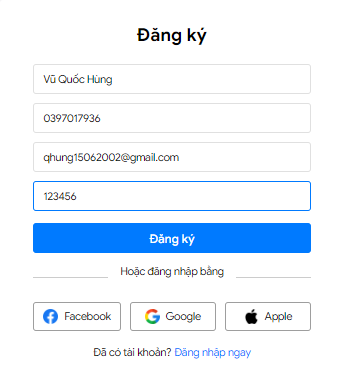
## Kết quả đạt được

### Use case đăng nhập



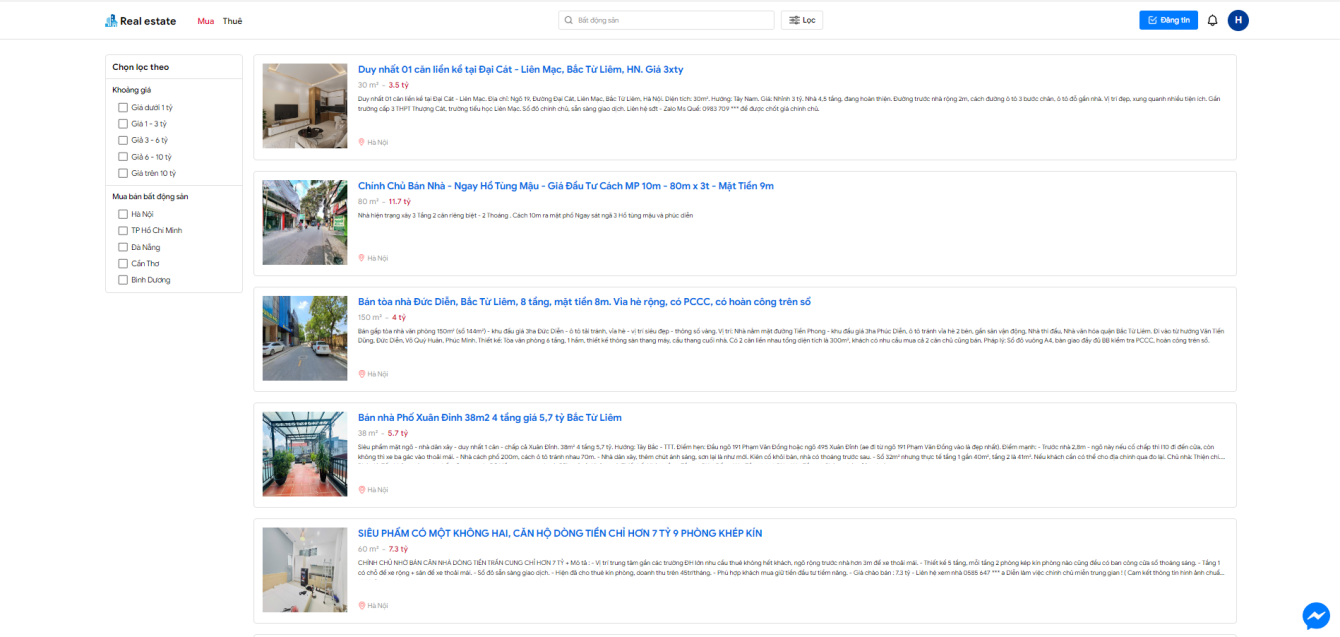
Hình 3.1: Màn hình chức năng đăng nhập

### Use case đăng ký



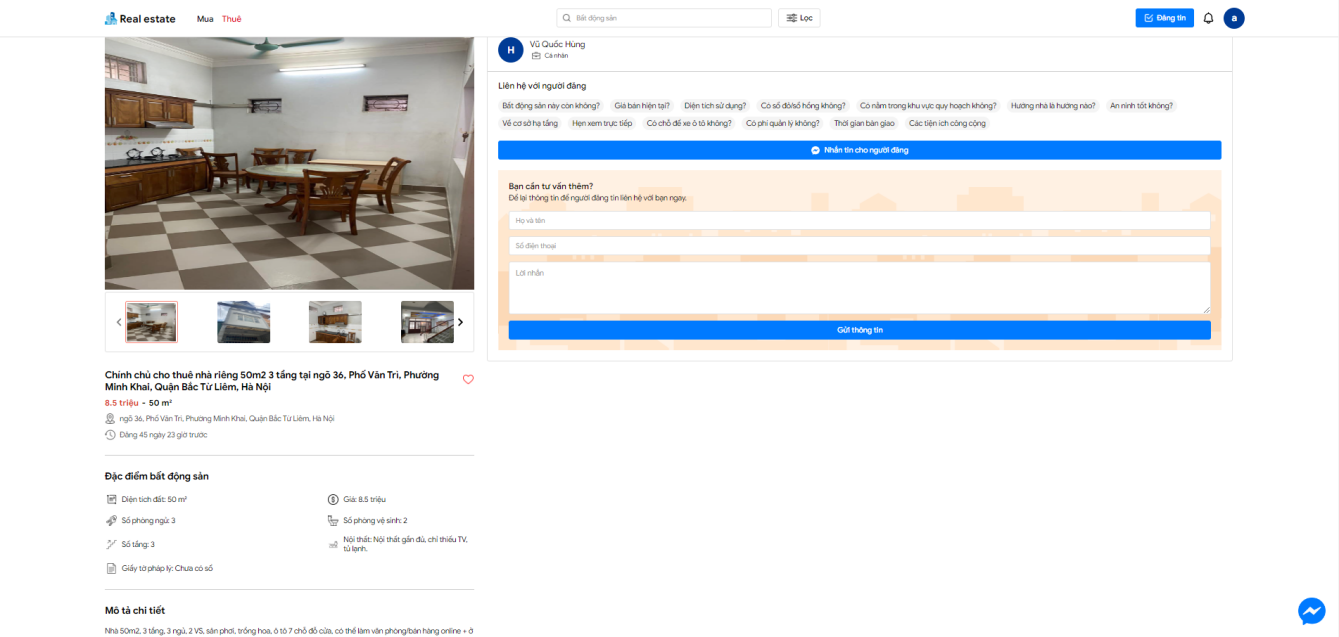
Hình 3.2: Màn hình chức năng đăng ký

### Use case tìm kiếm nhà trọ



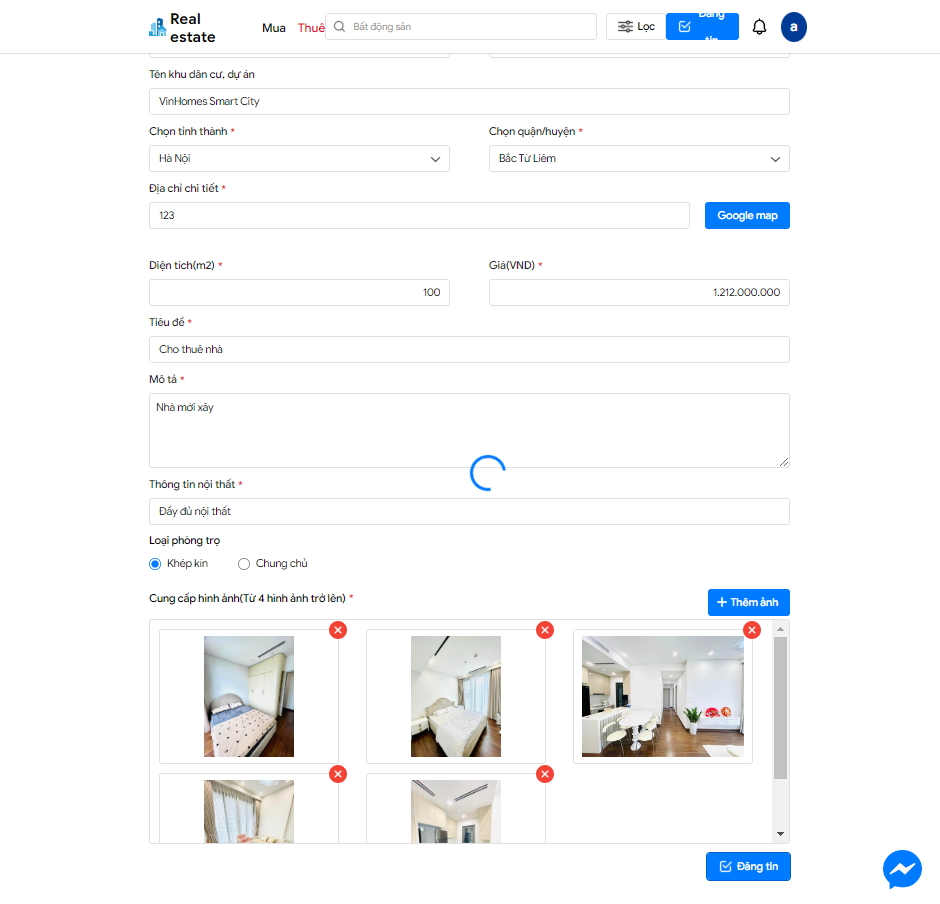
Hình 3.3: Màn hình tìm kiếm nhà trọ

### Use case xem thông tin nhà trọ



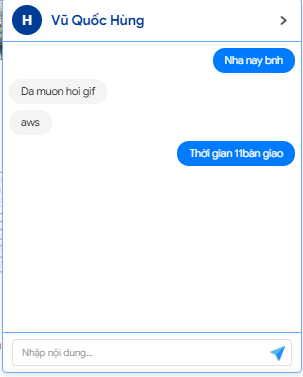
Hình 3.4: Màn hình xem thông tin chi tiết

### Use case đăng bài



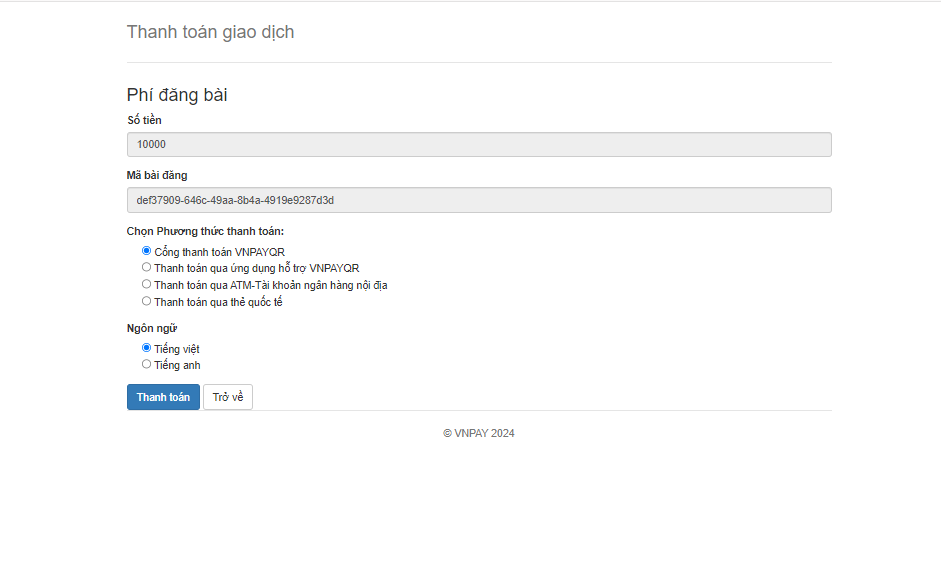
Hình 3.5: Màn hình đăng bài

### Use case nhắn tin

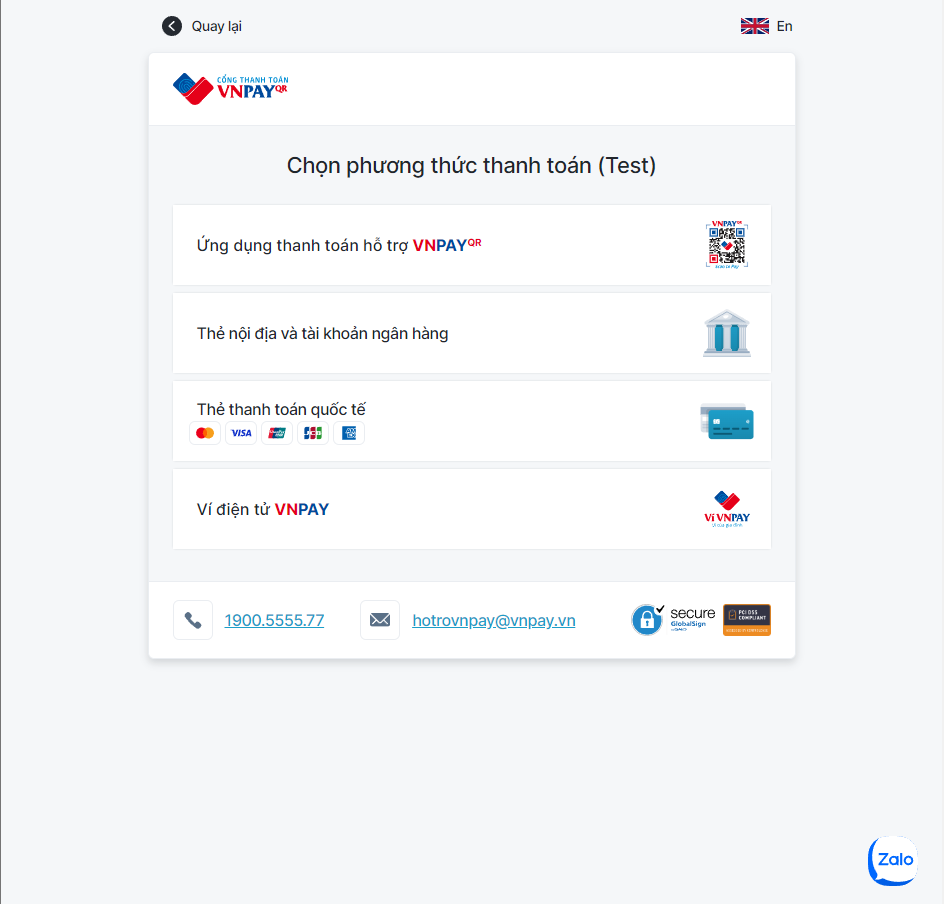


Hình 3.6: Màn hình nhắn tin

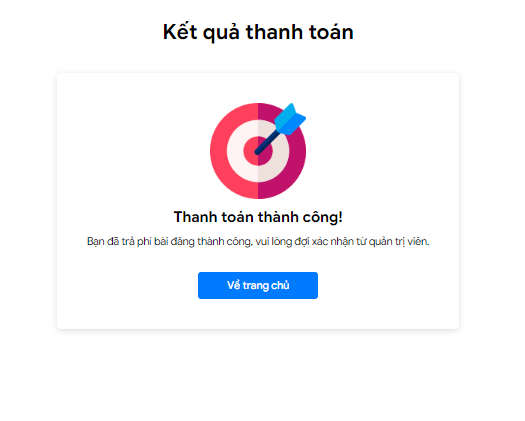
### Use case thanh toán trực tuyến



Hình 3.7: Màn hình thanh toán bước 1

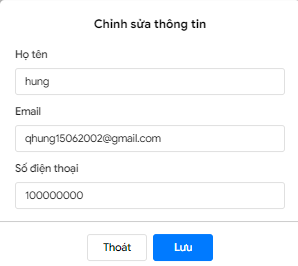


Hình 3.8: Màn hình thanh toán bước 2



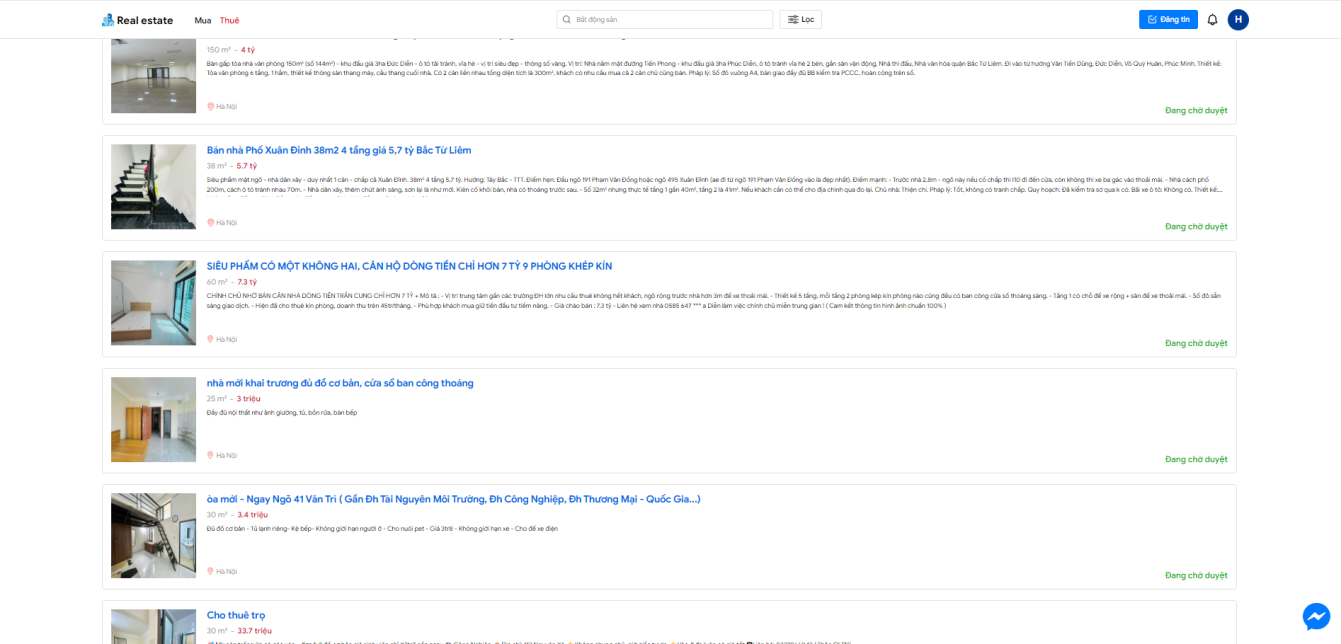
Hình 3.9: Màn hình thanh toán thành công

### Use case sửa thông tin cá nhân



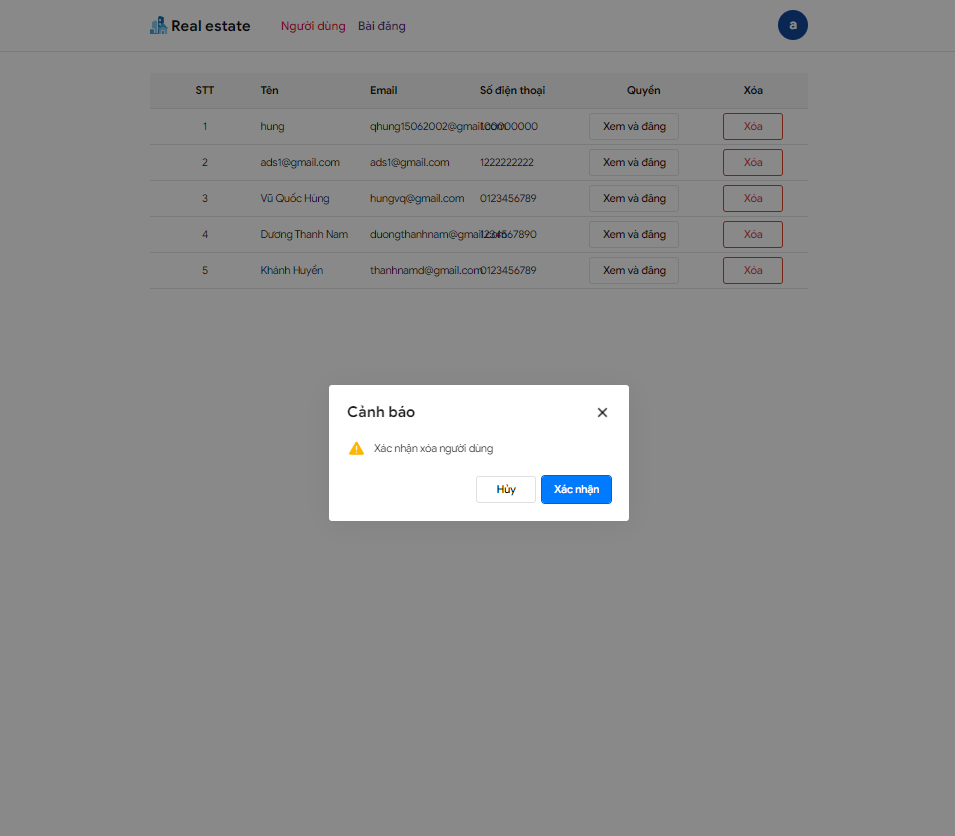
Hình 3.10: Màn hình sửa thông tin cá nhân

### Use case xem lịch sử bài đăng

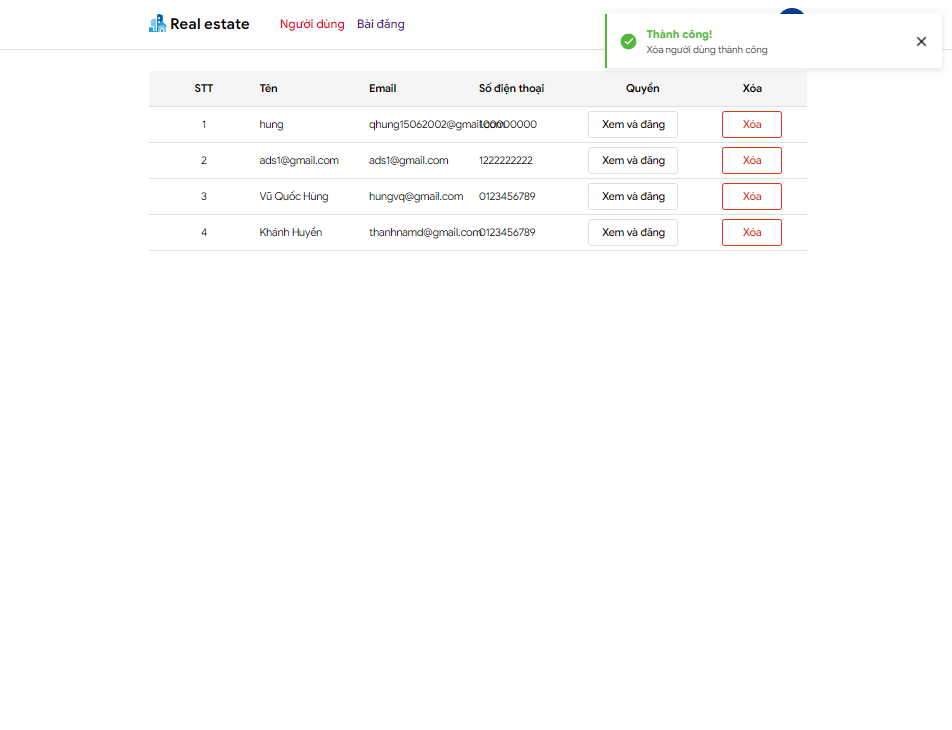


Hình 3.11: Màn hình xem lịch sử bài đăng

### Use case quản lý user

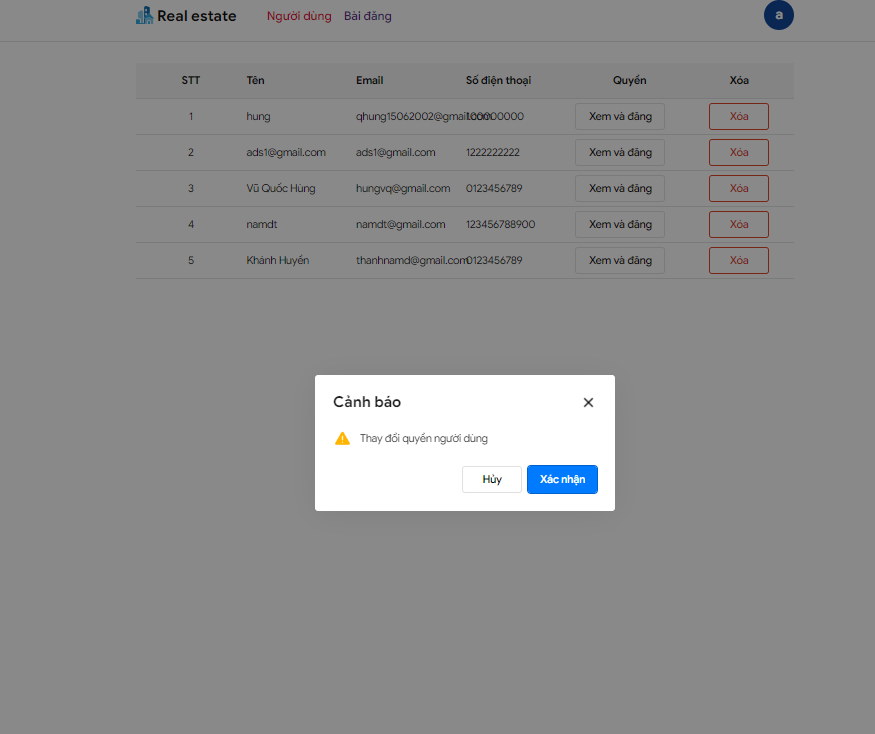


Hình 3.12: Màn hình trước khi xóa user

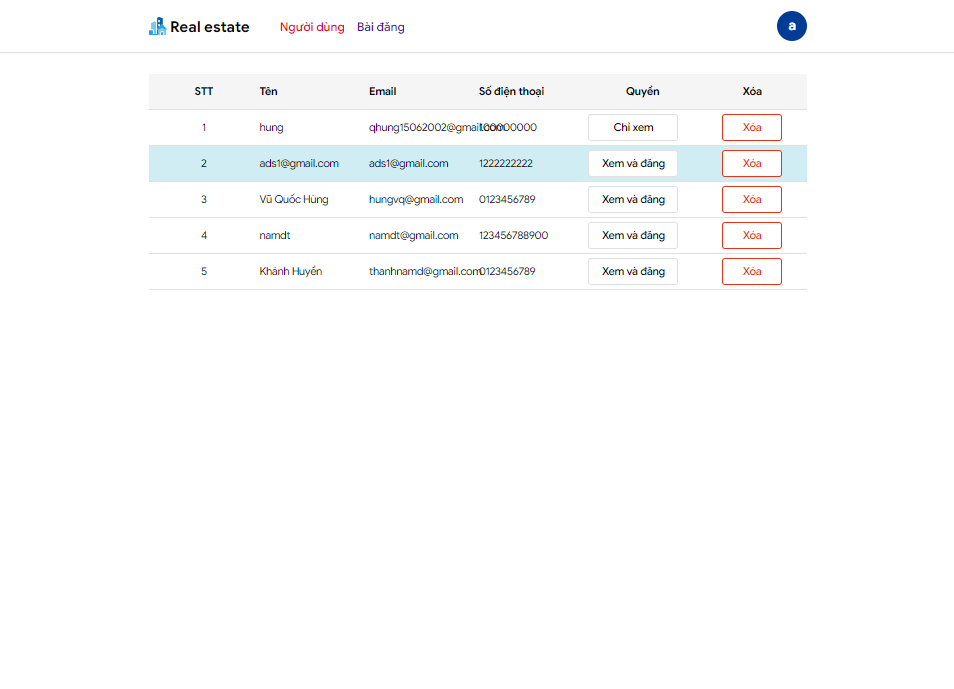


Hình 3.13: Màn hình sau khi xóa user

### Use case đổi quyền user

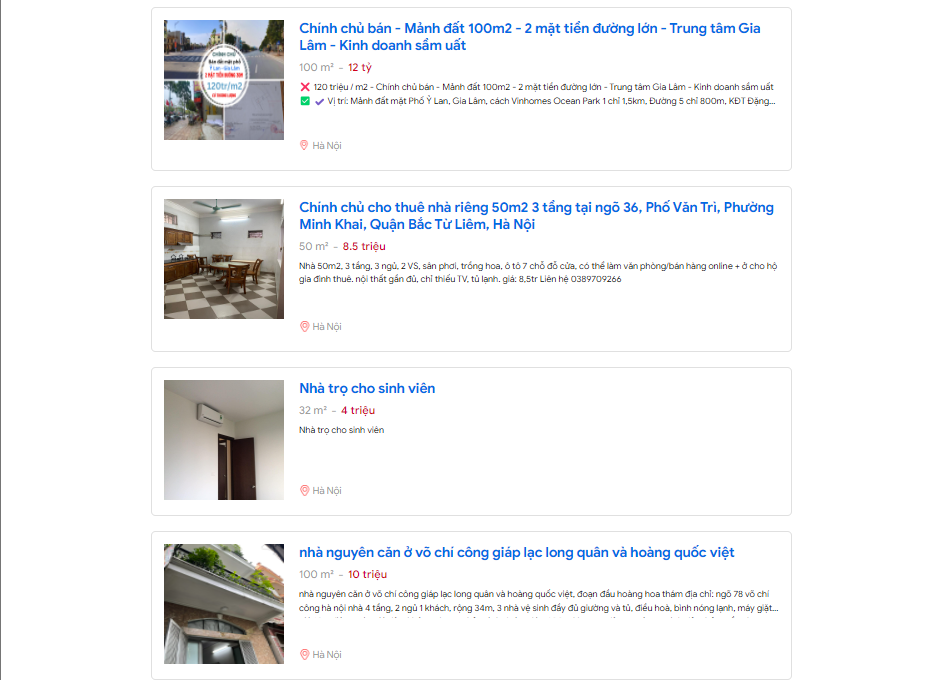


Hình 3.14: Màn hình trước khi đổi quyền user



Hình 3.15: Màn hình sau khi đổi quyền user

### Use case quản lý bài đăng



Hình 3.16: Màn hình quản lý bài đăng

## Cài đặt chương trình

### Cài đặt Visual Studio

Bước 1: Bạn vào trang [Download Visual Studio](https://visualstudio.microsoft.com/downloads/" \o "Download Visual Studio" \t "https://www.thegioididong.com/game-app/_blank). Ở đây mình chọn bản Community. Các bạn nhấn vào Free download để tải về nhé.

Bước 2: Bạn nhấn đúp chuột vào file vừa mới tải về, sau đó chọn Run để bắt đầu cài đặt Visual Studio.

Bước 3: Chọn Continue và bạn chờ vài phút để tải file cài đặt về.

Bước 4: Bạn chọn các mục, các ngôn ngữ lập trình mà bạn cần sử dụng. Ở đây mình chọn mục Desktop development with C++ và Python development.

Bước 5: Bạn chọn vị trí lưu file của Visual Studio, ở đây mình sẽ để mặc định. Sau đó bạn chọn  Install để bắt đầu quá trình tải và cài đặt.

Bước 6: Chờ Visual Studio tải và cài đặt các gói tài nguyên, quá trình này nhanh hay chậm tùy vào tốc độ tải của máy bạn. Bạn chọn Start after installation để khởi động Visual Studio sau khi tải xong nhé.

### Công cụ Visual Studio Code

Visual Studio Code (VS Code) là một trình biên tập mã nguồn mạnh mẽ và nhẹ được phát triển bởi Microsoft. Được xây dựng trên nền tảng Electron, VS Code hỗ trợ nhiều ngôn ngữ lập trình và cung cấp một loạt các tính năng và công cụ để giúp nhà phát triển viết, gỡ lỗi và quản lý mã nguồn.

Một số tính năng nổi bật của Visual Studio Code bao gồm:

* Giao diện người dùng đơn giản và hiện đại: VS Code có giao diện sạch sẽ và dễ sử dụng, với cấu trúc cây thư mục và thanh công cụ linh hoạt.
* Hỗ trợ nhiều ngôn ngữ lập trình: VS Code hỗ trợ nhiều ngôn ngữ lập trình phổ biến như JavaScript, TypeScript, C#, Python, HTML, CSS và nhiều ngôn ngữ khác. Nó cung cấp cú pháp sắc nét, kiểm tra lỗi và gợi ý thông minh trong quá trình viết mã.
* Gỡ lỗi và gợi ý mã: VS Code cung cấp tính năng gỡ lỗi mạnh mẽ cho nhiều ngôn ngữ và nền tảng. Nó cũng cung cấp gợi ý mã thông minh, tự động hoàn thành và kiểm tra lỗi trong quá trình viết mã.
* Hệ thống mở rộng và tích hợp công cụ: VS Code cho phép bạn mở rộng chức năng của nó thông qua các tiện ích mở rộng. Bạn có thể cài đặt các tiện ích mở rộng từ cộng đồng người dùng hoặc tạo ra của riêng bạn để tùy chỉnh trình biên tập theo nhu cầu của bạn. Ngoài ra, nó tích hợp tốt với các công cụ và dịch vụ phát triển khác như Git, Docker và các nền tảng đám mây.
* Hỗ trợ đa nền tảng: VS Code có sẵn trên Windows, macOS và Linux, cho phép bạn làm việc trên nhiều hệ điều hành khác nhau mà không mất tính tương thích.

Với sự kết hợp của những tính năng và khả năng mở rộng, Visual Studio Code đã trở thành một trình biên tập mã nguồn phổ biến và được ưa chuộng bởi các nhà phát triển trên toàn thế giới.

## Kiểm thử chương trình

### Kế hoạch kiểm thử

Những chức năng được kiểm thử:

* Đăng nhập: Người dùng đăng nhập vào hệ thống
* Đăng ký: Người dùng đăng ký tài khoản
* Tìm kiếm: Người dùng tìm kiếm nhà trọ theo nhu cầu
* Nhắn tin: Người dùng có thể nhắn tin cho người đăng bài
* Xem thông tin chi tiết: Người dùng có thể xem thông tin bài đăng chi tiết
* Sửa thông tin cá nhân: Người dùng cập nhật thông tin cá nhân
* Thanh toán trực tuyến: Người dùng trả phí cho bài đăng
* Xem lịch sử bài đăng: Người dùng có thể xem lịch sử những bài đã đăng

### Chức năng phía người dùng

Bảng 1: Kiểm thử chức năng phía người dùng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Chức năng** | **Các trường hợp thử nghiệm** | **Kết quả mong đợi** | **Kết quả thực tế** | **Kết quả thực tế** |
| 1 | Đăng nhập | Không nhập email và mật khẩu | Thông báo chưa nhập email và mât khẩu | Hiển thị thông báo chưa nhập email và mật khẩu | Pass |
| Nhập email không nhập mật khẩu | Thông báo chưa nhập mật khẩu | Hiển thị thông báo chưa nhập mật khẩu | Pass |
| Nhập mật khẩu không nhập email | Thông báo chưa nhập email tài khoản | Hiển thị thông báo chưa nhập email | Pass |
| Nhập sai thông tin tài khoản | Thông báo email hoặc mật khẩu không chính xác | Hiển thị thông báo email hoặc mật khẩu không chính xác | Pass |
| Nhập đúng tài khoản | Thông báo đăng nhập thành công và truy cập trang chủ hệ thống | Hiển thị thông báo đăng nhập thành công và truy cập trang chủ hệ thống | Pass |
| 2 | Đăng ký | Nhập tên email đã tồn tại | Thông báo đăng ký bằng email khác | Hiển thị thông báo đăng ký bằng email khác | Pass |
| Nhập mật khẩu không đúng yêu cầu | Thông báo mật khẩu bị sai | Hiển thị thông báo mật khẩu bị sai | Pass |
| Nhập xác nhận mật khẩu không khớp | Thông báo mật khẩu không trùng khớp | Hiển thị thông báo mật khẩu không trùng khớp | Pass |
| Nhập đầy đủ thông tin và chính xác | Thông báo đăng ký thành công và chuyển sang trang đăng nhập | Hiển thị thông báo đăng ký thành công và chuyển sang trang đăng nhập | Pass |
| 3 | Sửa thông tin cá nhân | Nhập số điện thoại ít hơn 10 số | Thông báo số điện thoại phải ít hơn 10 số | Thông báo số điện thoại phải ít hơn 10 số | Pass |
| Nhập đầy đủ thông tin và chính xác | Thông báo cập nhật thông tin thành công | Hiển thị thông báo cập nhật thông tin thành công | Pass |
| 4 | Xem chi tiết bài đăng | Hiển thị thông tin đầy đủ | Thông tin trên giao diện đã trùng khớp với dữ liệu tương ứng | Thông tin trên giao diện đã trùng khớp với dữ liệu tương ứng | Pass |
| 5 | Nhắn tin | Hiển thị đầy đủ thông tin | Thông tin trên giao diện đã trùng khớp với dữ liệu tương ứng | Thông tin trên giao diện đã trùng khớp với dữ liệu tương ứng | Pass |
| 6 | Thanh toán trực tuyến | Nhập đầy đủ thông tin thanh toán | Hiển thị thông báo thanh toán thành công và chuyển sang màn hình chính | Hiển thị thông báo thanh toán thành công và chuyển sang màn hình chính | Pass |
| 7 | Chỉnh sửa thông tin cá nhân | Nhập đầy đủ thông tin cá nhân cần sửa | Hiển thị thông báo cập nhật thông tin cá nhân thành công | Hiển thị thông báo cập nhật thông tin cá nhân thành công | Pass |

### Kết quả kiểm thử

* Tỉ lệ test case đạt: 100%
* Tỉ lệ test case thất bại: 0%
* Hệ thống chạy ổn định trên các trình duyệt web khác nhau như Firefox, Google Chrome, Microsoft Edge, CocCoc.

# KẾT LUẬN

Đề tài “Xây dựng website cung cấp dịch vụ cho thuê nhà trọ trên địa bàn Hà Nội” xuất phát từ những nhu cầu thực tế mà ngày nay trong ngành kinh doanh cần có. Qua quá trình thực hiện đề tài, em đã tìm hiểu, tích lũy và vận dụng được những kiến thức công nghệ được học từ trường, từ nơi làm việc và đã hoàn thành được những phần sau đây.

Kết quả đạt được

* Xây dựng thành công Website “Xây dựng website cung cấp dịch vụ cho thuê nhà trọ trên địa bàn Hà Nội”.
* Tìm hiểu được các công nghệ và phần mềm ứng dụng cho xây dựng Website.
* Nghiên cứu được quy trình xây dựng Website thương mại điện tử.

Việc xây dựng một website cung cấp dịch vụ cho thuê nhà trọ trên địa bàn Hà Nội là một giải pháp hiệu quả nhằm đáp ứng nhu cầu tìm kiếm chỗ ở của người dân và sinh viên tại một thành phố lớn với sự phát triển đô thị nhanh chóng. Website không chỉ giúp tối ưu hóa quá trình kết nối giữa chủ nhà và người thuê mà còn cung cấp một nền tảng minh bạch, an toàn và tiện lợi trong việc tìm kiếm, so sánh và lựa chọn chỗ ở.

Trong tương lai, việc mở rộng thêm các chức năng, cải tiến giao diện người dùng và liên tục cập nhật thông tin sẽ là những bước đi cần thiết để đảm bảo website luôn đáp ứng nhu cầu ngày càng cao của thị trường nhà trọ, góp phần vào việc nâng cao chất lượng cuộc sống và hỗ trợ quá trình phát triển đô thị của Hà Nội.

Em muốn bày tỏ lòng biết ơn chân thành nhất đến quý thầy cô trong khoa Công nghệ thông tin vì đã dành tâm huyết giảng dạy và truyền đạt kiến thức quý giá và cần thiết trong những năm học tại trường, đóng góp quan trọng cho sự hoàn thiện của đề tài đồ án tốt nghiệp này. Đặc biệt, em muốn gửi lời cảm ơn đặc biệt tới thầy TS. Nguyễn Văn Tỉnh đã hướng dẫn và hỗ trợ tận tâm, giúp em hoàn thành báo cáo đồ án tốt nghiệp cùng sản phẩm đi kèm một cách hoàn thiện nhất.

Do thiếu kinh nghiệm thực tế cũng như thời gian hạn chế, ứng dụng của em vẫn còn nhiều hạn chế. Vì vậy, em mong rằng các quý thầy cô có thể đóng góp ý kiến để giúp em hoàn thiện ứng dụng tốt hơn.

# TÀI LIỆU THAM KHẢO

[1] Hoàng Quang Huy, Phùng Đức Hòa, Trịnh Bá Quý, *Nhập môn công nghệ phần mềm*, NXB Đại học Công nghiệp Hà Nội.

[2] Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Thị Kim Phượng, *Giáo trình phân tích thiết kế hệ thống*, NXB Giáo dục VN.

[3] Vũ Thị Dương, Phùng Đức Hòa, Nguyễn Thị Hương Lan (2015), “Giáo trình Phân tích thiết kế hướng đối tượng”, NXB Khoa học và Kỹ thuật

[4] Trang tài liệu UML <https://www.tutorialspoint.com/uml/index.htm>

[5] Trang hỏi đáp về lập trình <https://stackoverflow.com/home/>