**TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

**KHOA CÔNG NGHỆ THÔNG TIN**

---🙘🙚---



**BÁO CÁO MÔN HỌC: KIỂM THỬ PHẦN MỀM**

**ĐỀ TÀI: XÂY DỰNG WEB QUẢN LÝ NHÀ TRỌ**

**Giảng viên hướng dẫn: Th.S. Võ Phương Bình**

**Sinh viên thực hiện:** 2113005 – Lê Hà Hiếu Nghĩa

2116977 – Trần Hữu Bằng

2115277 – Nguyễn Trung Tín

2111813 – Hoàng Hải Đăng

**ĐÀ LẠT, 5/2024**

# MỤC LỤC

[MỤC LỤC 2](#_Toc167913930)

[LỜI CẢM ƠN 2](#_Toc167913931)

[DANH MỤC HÌNH 3](#_Toc167913932)

[DANH MỤC BẢNG 3](#_Toc167913933)

[CHƯƠNG 1: LÝ THUYẾT VỀ KIỂM THỬ PHẦN MỀM 4](#_Toc167913934)

[1.1. Một số khái niệm 4](#_Toc167913935)

[1.2. Các cấp độ kiểm thử 5](#_Toc167913936)

[1.3. Các phương pháp kiểm thử 5](#_Toc167913937)

[1.4. Các công cụ hỗ trợ KTPM 6](#_Toc167913938)

[CHƯƠNG 2: MÔ TẢ KIẾN TRÚC HỆ THỐNG 8](#_Toc167913939)

[2.1. Tóm tắt hệ thống 8](#_Toc167913940)

[2.2. Sơ đồ ngữ cảnh của hệ thống 8](#_Toc167913941)

[2.3. Xác định các tác nhân 8](#_Toc167913942)

[2.4. Sơ đồ Use Case 10](#_Toc167913943)

[2.5. Danh sách Use Cases 10](#_Toc167913944)

[2.5.1. Đăng nhập 10](#_Toc167913945)

[2.5.2. Đăng thông tin 11](#_Toc167913946)

[2.5.3. Tìm kiếm thông tin 12](#_Toc167913947)

[2.5.4. Cập nhật 12](#_Toc167913948)

[2.6. Các yêu cầu và ràng buộc phi chức năng 13](#_Toc167913949)

[2.7. Công nghệ và công cụ sử dụng 14](#_Toc167913950)

[2.8. Domain Name class 14](#_Toc167913951)

[2.8.1. Liên kết giữa các thuộc tính 14](#_Toc167913952)

[2.8.2. Chi tiết các bảng 17](#_Toc167913953)

[2.8.3. Hợp đồng hoạt động 20](#_Toc167913954)

[2.8.4. Sơ đồ hoạt động (Activity Diagram) 20](#_Toc167913955)

[CHƯƠNG 3: KẾT QUẢ ĐẠT ĐƯỢC 22](#_Toc167913956)

[3.1. Giao diện 22](#_Toc167913957)

[3.2. Các test scenario 25](#_Toc167913958)

[3.3. Kết quả kiểm thử thủ công và tự động 27](#_Toc167913959)

[3.4. Mã nguồn 27](#_Toc167913960)

[3.4.1. Client 27](#_Toc167913961)

[3.4.2. Server 56](#_Toc167913962)

[TÀI LIỆU THAM KHẢO 70](#_Toc167913963)

# LỜI CẢM ƠN

Lời đầu tiên chúng em xin chân thành cảm ơn giảng viên hướng dẫn, ThS.Võ Phương Bình – người hướng dẫn chúng em tận tình trong thời gian học tập và làm bài. Những sự chỉ dẫn là hướng đi chính xác để đề tài của nhóm em triển khai và hoàn thiện đáp ứng những tiêu chí đã được đưa ra.

Dưới đây là kết quả của quá trình tìm hiểu và nghiên cứu mà nhóm em đã đạt được trong thời gian vừa qua. Trong quá trình thực hiện, do khả năng còn hạn chế và chưa có được nhiều kinh nghiệm thực tế nên chúng em không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự thông cảm và những góp ý chỉ bảo tận tình của quý thầy, cô.

Trong khuôn khổ của bài tiểu luận, do thời gian và kinh nghiệm thực tế còn hạn chế nên có những phần thực hiện chưa được tốt, chúng em rất mong nhận được sự góp ý của thầy và các bạn.

Chúng em xin chân thành cảm ơn !

# DANH MỤC HÌNH

[Hình 1: Các cấp độ kiểm thử trong quy trình phát triển phần mềm 5](#_Toc167913964)

[Hình 2: Một số công cụ phổ biến hỗ trợ việc KTPM 6](#_Toc167913965)

[Hình 3: Mô tả ngữ cảnh của hệ thống 8](#_Toc167913966)

[Hình 4: Sơ đồ Use Case tổng quát 10](#_Toc167913967)

[Hình 5: Sơ đồ tuần tự cho UC1 11](#_Toc167913968)

[Hình 6: Sơ đồ tuần tự cho UC3 12](#_Toc167913969)

[Hình 7: Sơ đồ tuần tự cho UC4 13](#_Toc167913970)

[Hình 8: Jest Framework Testing 14](#_Toc167913971)

[Hình 9: Biểu đồ lớp cho HomePageController 15](#_Toc167913972)

[Hình 10: Biểu đồ lớp cho userController 16](#_Toc167913973)

[Hình 11: Sơ đồ hoạt động chung cho toàn hệ thống 21](#_Toc167913974)

[Hình 12: Sơ đồ hoạt động cho thao tác đăng ký 21](#_Toc167913975)

[Hình 13: Giao diện trang chủ 22](#_Toc167913976)

[Hình 14: Công cụ tìm kiếm theo ba yếu tố chính 23](#_Toc167913977)

[Hình 15: Các bài đăng nổi bật 24](#_Toc167913978)

[Hình 16: Phần Footer của web 25](#_Toc167913979)

# DANH MỤC BẢNG

[Bảng 1: So sánh hai phương pháp kiểm thử tự động và thủ công 6](#_Toc167913980)

[Bảng 2: So sánh ưu / nhược điểm của phương pháp kiểm thử Hộp đen 7](#_Toc167913981)

[Bảng 3:Tóm tắt tính năng hệ thống 8](#_Toc167913982)

[Bảng 4: Use case đăng nhập 11](#_Toc167913983)

[Bảng 5: Use case đăng thông tin 11](#_Toc167913984)

[Bảng 6: Use case tìm kiếm thông tin 12](#_Toc167913985)

[Bảng 7: Use case cập nhật (thêm xóa sửa) 13](#_Toc167913986)

[Bảng 8: Bảng các ràng buộc cho trang web 14](#_Toc167913987)

[Bảng 9: Mô tả mối quan hệ giữa các thuộc tính 14](#_Toc167913988)

[Bảng 10: Bảng Contract Operation của web 20](#_Toc167913989)

[Bảng 11: Các Scenario chung cho toàn bộ trang web 25](#_Toc167913990)

[Bảng 12: Điều kiện tiên quyết với thao tác kiểm thử giao diện 25](#_Toc167913991)

[Bảng 13: Điều kiện tiên quyết với thao tác đăng nhập 26](#_Toc167913992)

[Bảng 14: Điều kiện tiên quyết với thao tác đăng ký 26](#_Toc167913993)

[Bảng 15: Điều kiện tiên quyết cho thao tác lấy lại mật khẩu 27](#_Toc167913994)

# CHƯƠNG 1: LÝ THUYẾT VỀ KIỂM THỬ PHẦN MỀM

## Một số khái niệm

**Khái niệm phần mềm**

Phần mềm thường được sử dụng để hướng dẫn, cho phép người dùng có thể tương tác với các phần cứng của máy tính hoặc thực hiện nhiều tác vụ có liên quan tới công việc, hoạt động của máy tính. Nó bao gồm một tập hợp những câu lệnh được lập trình từ một hay nhiều ngôn ngữ lập trình khác nhau theo một trật tự nhất định. Điều này nhằm giúp máy tính hiểu được và thực hiện các chức năng hay một vấn đề nào đó.

Chỉ cần một sự cố nhỏ trong phần mềm đều dẫn đến những tác động nghiêm trọng như thiệt hại về tiền bạc, thời gian và danh tiếng đối với các công ty, doanh nghiệp. Năm 1980, lỗi phần mềm trong mã điều khiển máy xạ trị Therac-25 là nguyên nhân trực tiếp gây một số ca tử vong và thiệt hại 370 triệu USD [1] [2]. Một điểm khác cần xem xét đến là chất lượng của phần mềm ở Mỹ từ giai đoạn 2020 trở về sau thì tồi tệ hơn [3].

**Tại sao phải thực hiện KTPM?**

**Khái niệm KTPM**

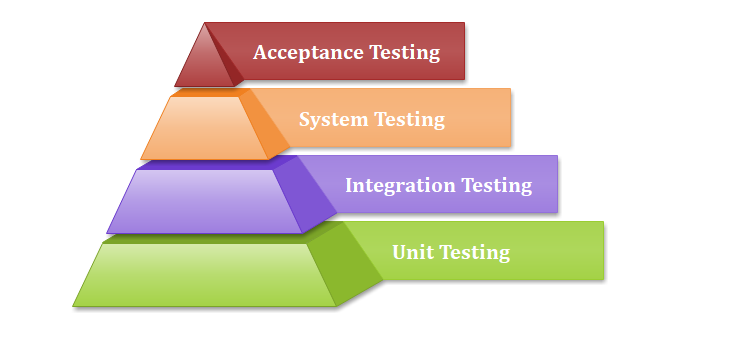
Glenford J.Myers (nhà khoa học máy tính và doanh nhân người Mỹ) định nghĩa kiểm thử phần mềm là “ *quá trình thực thi một chương trình hoặc một phần của chương trình với mục đích tìm ra lỗi* [4]”.

Nó đảm bảo sản phẩm phải đáp ứng chính xác theo các yêu cầu của khách hàng đề ra. Ngoài ra, nó còn tạo điều kiện cho người dùng tận dụng tối đa tư duy sáng tạo để bạn có thể phát hiện ra những điểm mà người khác chưa nhìn thấy.

Phần mềm có nhiều khả năng xảy ra lỗi vì chúng được thiết kế thủ công. Những lỗi này có thể xuất hiện trong giai đoạn mã hóa phần mềm và trong suốt vòng đời phát triển. Trong vòng đời phát triển, lỗi có thể xảy ra ở giai đoạn đầu khi các yêu cầu được hiểu, viết hoặc thiết kế. Ngoài ra, các tài liệu được xác định rõ ràng, khi được giao cho các nhóm khác nhau, thường có thể dẫn đến những cách hiểu khác nhau nếu các nhóm không được đào tạo để hiểu các tài liệu yêu cầu.

## Các cấp độ kiểm thử

Thông qua các mức độ kiểm thử phần mềm sẽ giúp bạn có thể đánh giá chức năng của ứng dụng phần mềm có đáp ứng được những yêu cầu đã chỉ định hay không. Đồng thời thông qua các mức độ kiểm thử phần mềm cũng giúp bạn tìm và tiến hành sửa lỗi nhằm đảm bảo sản phẩm tạo ra có chất lượng tốt nhất.



Hình : Các cấp độ kiểm thử trong quy trình phát triển phần mềm

## Các phương pháp kiểm thử

Kiểm thử phần mềm thường được tiến hành trong ba giai đoạn: tạo dựng, thực hiện và đánh giá các trường hợp kiểm thử [5]. Các kỹ thuật chính được sử dụng là kiểm thử thủ công và tự động.

**Kiểm thử thủ công (Manual testing)**

Đây là việc kiểm thử phần mềm được thực hiện thủ công bởi QA/tester. Nó được thực hiện để tìm ra bug trong quá trình phát triển phần mềm. Trong manual testing, Tester phải kiểm tra tất cả các tính năng cần thiết của ứng dụng. Trong quá trình này, Tester phải execute test, lập báo cáo mà không cần sự trợ giúp của bất kì công cụ tự động nào.

Quy trình Testint manual:

1. Hiểu rõ các yêu cầu
2. Viết các test case

Công cụ kiểm thử:

* + Quản lý kiểm tra: HP-ALM hoặc testlink: có thể tự viết ra các kịch bản kiểm thử, các test case và test scripting, có thêt liên kết yêu cầu của dự án. Nó còn đảm bảo các báo cáo thử nghiệm được dễ dàng và hiệu quả.
  + Quản lý lỗi: Bugzilla, Mantis: đây là cần thiết để nâng cao và quản lý các lỗi. Đôi khi bạn phải tạo các báo cáo lỗi như đã nêu ở bản phát hành(từ trước)

**Kiểm thử tự động**

Kiểm thử phần mềm tự động là thực hiện kiểm thử phần mềm bằng một chương trình đặc biệt với rất ít hoặc không có sự tương tác của con người, giúp cho người thực hiện việc kiểm thử phần mềm (tester) không phải lặp đi lặp lại các bước nhàm chán.

**Điểm khác biệt giữa hai phương pháp thủ công và tự động**

|  |  |  |
| --- | --- | --- |
| Yếu tố | Kiểm thử thủ công | Kiểm thử tự động |
| Thời gian xử lý | Kiểm thử thủ công tốn nhiều thời gian và tốn nhân lực. | Tốn nhiều thời gian và tốn nhân lực. |
| Độ tin cậy | Là phương pháp đáng tin cậy vì nó được thực hiện bằng các công cụ và tập lệnh. | Không chính xác vì có thể xảy ra lỗi do con người. |
| Hiệu quả | Không hiệu quả về mặt chi phí đối với hồi quy khối lượng thấp | Không hiệu quả về chi phí cho hồi quy khối lượng lớn. |

Bảng 1: So sánh hai phương pháp kiểm thử tự động và thủ công

## Các công cụ hỗ trợ KTPM

Đó là công cụ dùng để kiểm tra xem sản phẩm phần mềm trước khi đưa vào sử dụng có phù hợp với các yêu cầu mong đợi hay không và nó có khiếm khuyết hay lỗi nào không. Việc lựa chọn công cụ kiểm thử phần mềm phù hợp với tính chất của dự án sẽ giúp bạn tiết kiệm thời gian, công sức, đảm bảo kết quả nhất quán và năng suất kiểm nghiệm cao. Bên cạnh đó, thông tin về kiểm tra cũng được trình bày một cách rõ ràng, dễ hiểu.



Hình : Một số công cụ phổ biến hỗ trợ việc KTPM

**Selenium** là một trong những công cụ kiểm thử phần mềm phổ biến nhất. Nó được thiết kế đặc biệt để hỗ trợ kiểm tra tự động hóa các khía cạnh chức năng của các ứng dụng trên web, đa dạng các nền tảng và trình duyệt.

**Applitools** là một công cụ kiểm tra tự động, tự động xác nhận giao diện và trải nghiệm người dùng của các ứng dụng và trang web. Nó được thiết kế để dễ dàng tích hợp với các bài kiểm tra hiện có thay vì yêu cầu tạo một bài kiểm tra mới.

**Postman** hiện là một trong những công cụ phổ biến nhất được sử dụng trong kiểm thử API. Nó bắt đầu vào năm 2012 như một dự án phụ của Abhinav Asthana để đơn giản hóa quy trình làm việc API trong kiểm thử và phát triển.

Ưu và nhược điểm của kiểm thử Hộp đen theo tác giả Isha Sunita Sangwan [6]:

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| Số lượng trường hợp kiểm thử được giảm để đạt được kiểm thử hợp lý | Các trường hợp kiểm thử khó thiết kế nếu không có thông số kỹ thuật rõ ràng. |
| Các trường hợp kiểm thử có thể hiển thị sự hiện diện hoặc vắng mặt của các lớp lỗi | Chỉ một số lượng nhỏ đầu vào có thể thực sự có thể được kiểm tra. |
| Hiệu quả hơn trên các đơn vị mã lớn hơn so với kiểm tra hộp rõ ràng. | Một số phần của backend hoàn toàn không được thử nghiệm. |
| Lập trình viên và tester đều độc lập với nhau. | Cơ hội có các đường dẫn không xác định trong quá trình thử nghiệm này |

Bảng : So sánh ưu / nhược điểm của phương pháp kiểm thử Hộp đen

# CHƯƠNG 2: MÔ TẢ KIẾN TRÚC HỆ THỐNG

## Tóm tắt hệ thống

Hệ thống nhà trọ Đà Lạt là một trang web tìm nhà trọ, các căn hộ hoặc nhà nguyên căn nhằm đáp ứng cho các sinh viên cũng như những người lao động từ những tỉnh thành khác trên cả nước khi đến để làm việc và học tập. Trang web giúp người dùng tiết kiệm được thời gian cũng như kinh phí sau khi sử dụng trang web.

Mô hình thác nước sẽ được sử dụng trong quá trình phát triển trang web.

|  |  |
| --- | --- |
| **ID** | **Mô tả tính năng** |
| **FEA-01** | **Đăng nhập** |
| **FEA-02** | **Đăng ký** |
| **FEA-03** | **Đăng thông tin liên quan** |
| **FEA-04** | **Tìm kiếm thông tin** |
| **FEA-05** | **Cập nhật thông tin của SinhVien, do ChuTro thực hiện** |

Bảng :Tóm tắt tính năng hệ thống

## Sơ đồ ngữ cảnh của hệ thống

A diagram of a diagram

Description automatically generated

Hình : Mô tả ngữ cảnh của hệ thống

## Xác định các tác nhân

Dựa trên tóm tắt về hệ thống, ta có thể xác định được các tác nhân chính của hệ thống như sau:

* + SINHVIEN
  + CHUTRO

Đặc điểm nhân khẩu học:

• Sinh viên

• Công nhân, người lao động

• Người mới chuyển đến thành phố

• Người cần thuê nhà ở tạm thời (đi du lịch hoặc công tác ngắn hạn)

• Mục đích thuê trọ: làm thuê xa nhà

Nhu cầu và mong muốn:

• Nhận được thông tin cần thiết liên quan: giá thuê hằng tháng, khoảng cách, vị trí

## Sơ đồ Use Case

**A diagram of a network

Description automatically generated**

Hình : Sơ đồ Use Case tổng quát

## Danh sách Use Cases

### Đăng nhập

|  |  |
| --- | --- |
| **Use Case UC1: Đăng nhập** | |
| **Phạm vi truy cập** | Tất cả các actor |
| **Luồng sự kiện chính** | 1. Hệ thống hiển thị màn hình đăng nhập 2. Người dùng nhập username, password 3. Nếu thành công, hệ thống thể hiện việc đăng nhập đã hoàn tất 4. Kết thúc use case |
| **Luồng sự kiện phụ** | **1a** Mật khẩu không hợp lệ (khi nhập sai username hoặc password   * Bắt nhập lại thông tin, đi kèm đó là thông báo nhập sai và yêu cầu nhập lại * Quay lại bước 2 trong luồng sự kiện chính   **1b.** Quên mật khẩu: khi người dùng chọn chức năng quên mật khẩu trên màn đăng nhập   * Hiển thị màn hình |

Bảng : Use case đăng nhập



Hình : Sơ đồ tuần tự cho UC1

Trong trường hợp quên mật khẩu, người dùng có thể nhấn vào thao tác quên mật khẩu và làm lại password mới.

### Đăng thông tin

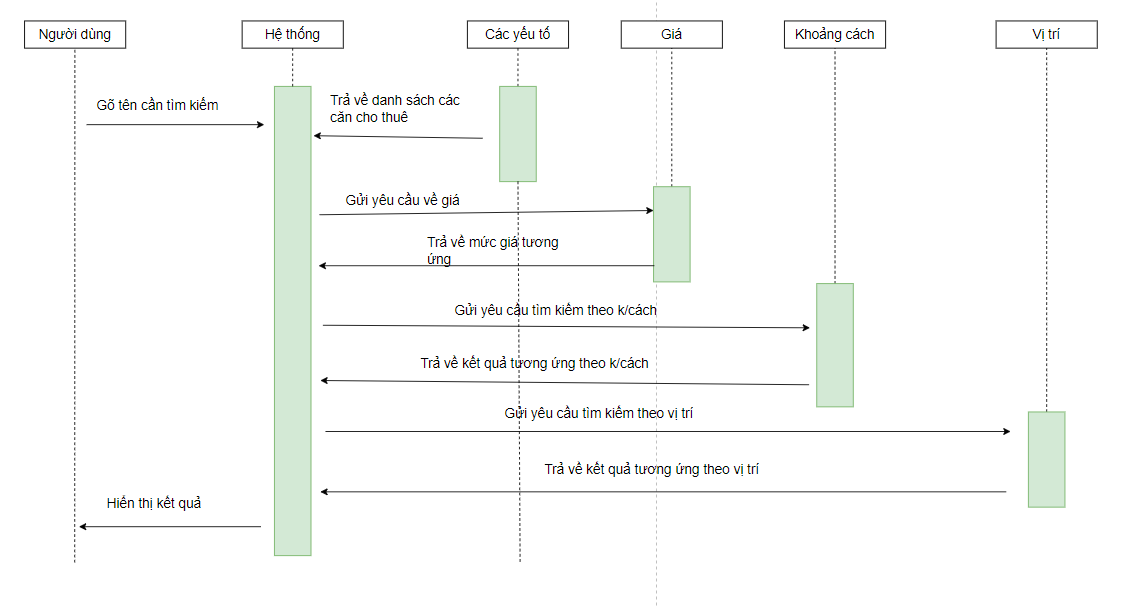
|  |  |
| --- | --- |
| **Use Case UC2: Đăng thông tin** | |
| **Phạm vi truy cập** | ChuTro |
| **Luồng sự kiện chính** | 1a: Tạo mới   1. Thực hiện việc đăng nhập 2. Đưa ra form chứa các textbox liên quan, người chủ chỉ cần nhập thông tin liên quan vào 3. Nhấn xác nhận 4. Gửi thông báo thành công, thông tin mới đó được hiển thị lên form 5. Kết thúc thao tác (có thể load lại trang) |
| **Luồng sự kiện phụ** | * Nếu chưa đăng nhập thì phải thực hiện lại bước **1** trong luồng sự kiện chính |

Bảng : Use case đăng thông tin

### Tìm kiếm thông tin

|  |  |
| --- | --- |
| **Use Case UC3: Tìm kiếm thông tin** | |
| **Phạm vi truy cập** | SinhVien |
| **Luồng sự kiện chính** | 1a:   1. Truy cập trang web 2. Thực hiện tìm trên trang chính với ba yếu tố chính: khoảng cách, giá, vị trí 3. Dựa trên thông tin được để lại, có thể thực hiện liên lạc với chủ sở hữu 4. Kết thúc thao tác |
| **Luồng sự kiện phụ** | * Có thể không cần đăng nhập, thực hiện thao tác ngay trên trang chính |

Bảng : Use case tìm kiếm thông tin

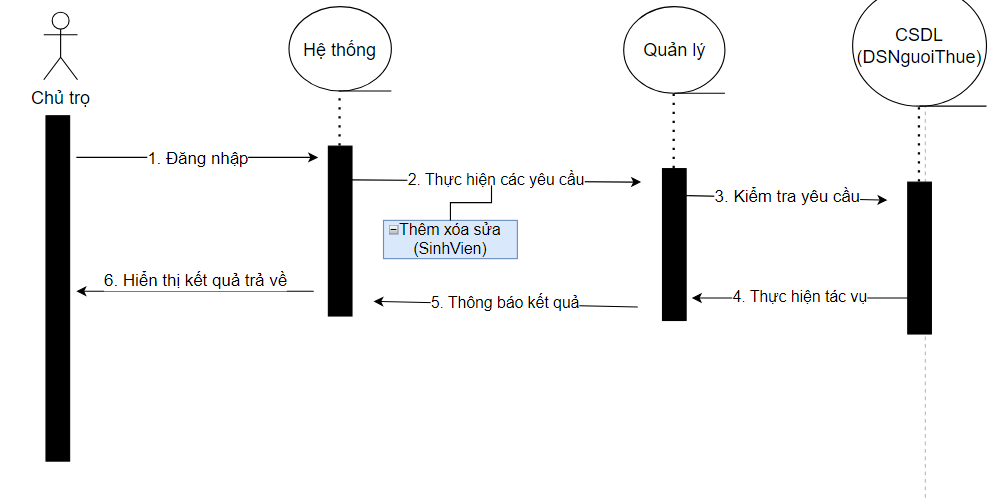
**

Hình : Sơ đồ tuần tự cho UC3

### Cập nhật

|  |  |
| --- | --- |
| **Use Case UC4: Cập nhật( Thêm xóa, sửa)** | |
| **Phạm vi truy cập** | ChuTro |
| **Điều kiện kiên quyết** | * Phải hoàn thành việc đăng nhập vào web |
| **Luồng sự kiện chính** | 1a: Chức năng thêm   1. Hệ thống hiển thị màn hình quản lý (điền thông tin của người mới đến thuê) 2. ChuTro nhập thông tin liên quan 3. Nếu thành công, hiển thị thêm thông tin thành công, cập nhật trong db 4. Kết thúc thao tác   1b: Chức năng chỉnh sửa hoặc xóa   1. Hệ thống hiển thị màn hình quản lý 2. Nhấn vào nút edit hoặc delete ở góc bên phải 3. Hiển thị thông tin để user có thể thực hiện thao tác 4. Hệ thống gửi form xác nhận có muốn thay đổi hay không? 5. Nếu thành công, hiển thị thông tin đã chỉnh sửa hoặc xóa thành công 6. Kết thúc thao tác |

Bảng : Use case cập nhật (thêm xóa sửa)

**

Hình : Sơ đồ tuần tự cho UC4

## Các yêu cầu và ràng buộc phi chức năng

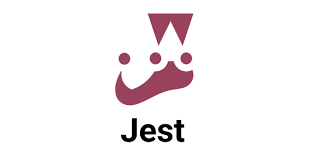
|  |  |  |
| --- | --- | --- |
| **ID** | **Loại** | **Mô tả** |
| NFA-01 | Hiệu suất | Truy cập trang web nhanh |
| NFA-02 | Khả năng sử dụng | Các đối tượng dùng web |
| NFA-03 | Độ tin cậy | Thông tin chính xác |

Bảng : Bảng các ràng buộc cho trang web

## Công nghệ và công cụ sử dụng

* Dự án được viết bằng ngôn ngữ ReactJS, sử dụng CSDL MongoDB
* Môi trường phát triển: Visual Studio Code
* Các design pattern sẽ được áp dụng trong dự án: Rest API, Expressjs và Back-end MVC

**Giới thiệu về công cụ kiểm thử Jest Framework**



Hình : Jest Framework Testing

Jest là một testing framework, được xây dựng bởi Facebook và cộng đồng React rất đón nhận. Có một quan niệm sai lầm đó là Jest được thiết kế đặc thù để làm việc với React, tuy nhiên, theo tài liệu của Jest, “Dù Jest có thể được xem là test runner đặc thù cho React nhưng thực tế là một nền tảng testing phổ cập, với khả năng thích nghi cho bất kể thư viện hoặc JavaScript framework nào. Jest hoàn toàn dễ sử dụng để test bất kỳ mã JavaScript nào”.

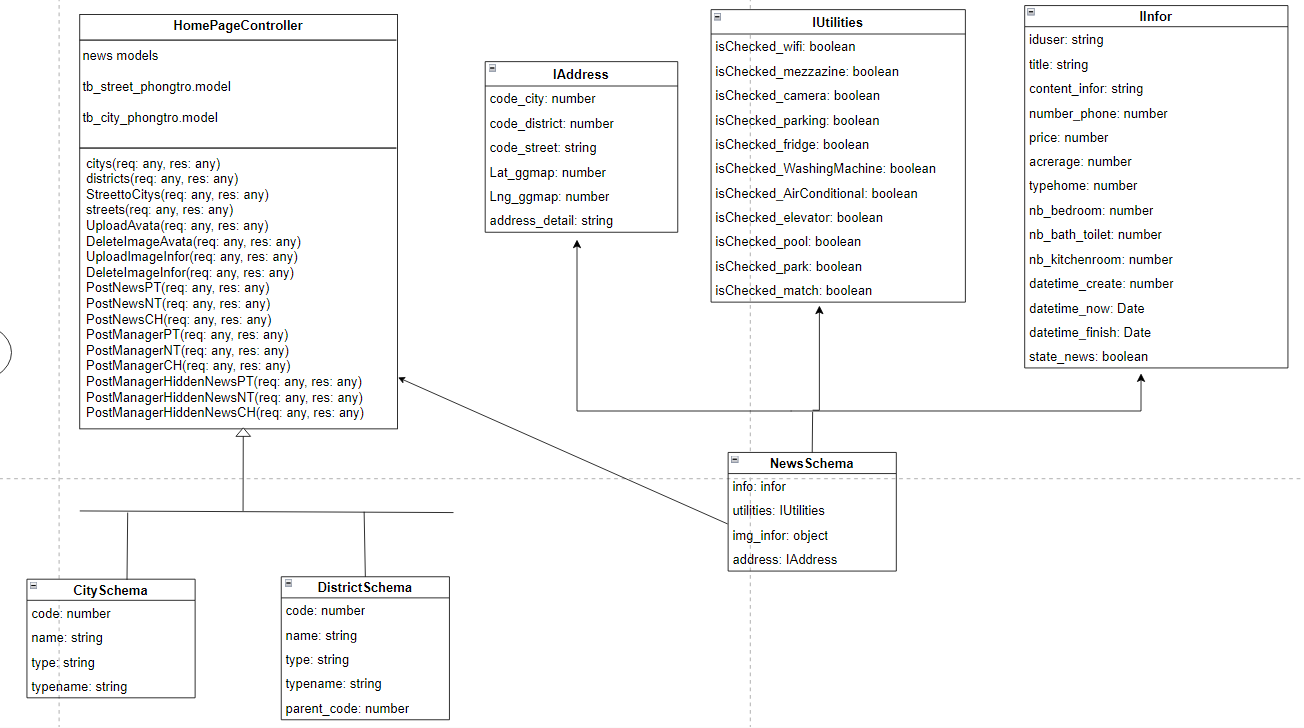
## Domain Name class

### Liên kết giữa các thuộc tính

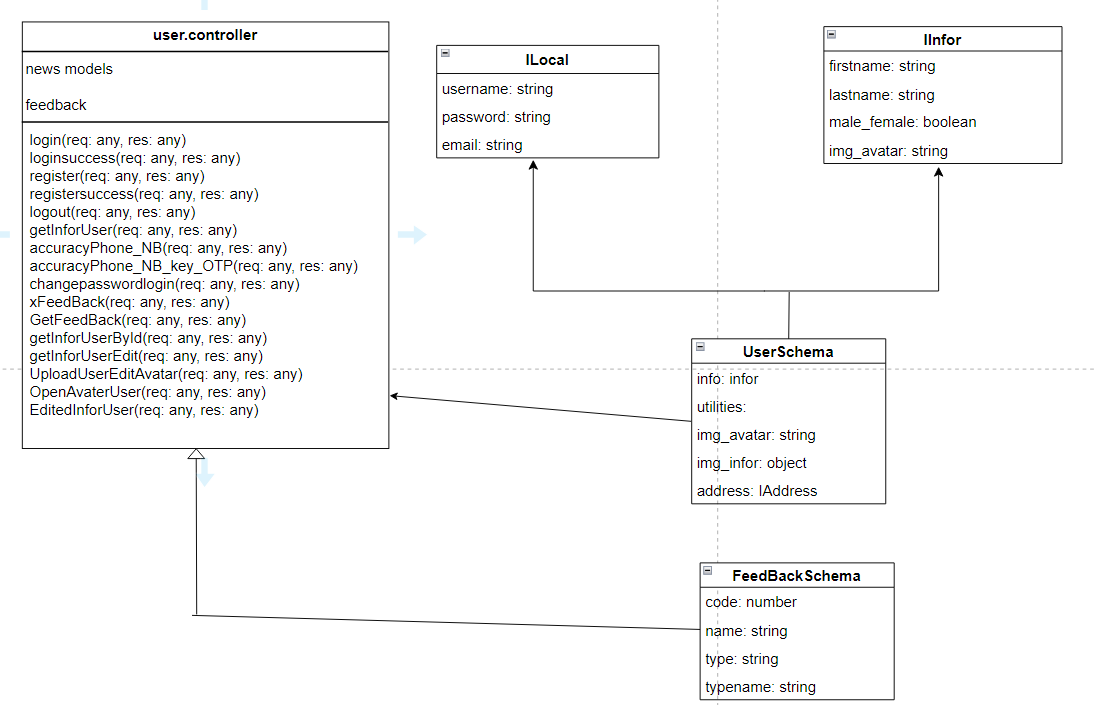
|  |
| --- |
| **Mô tả các quan hệ** |
| Một **SinhVien** có thể đến thuê tại một phòng trong một khoảng thời gian nhất định. |
| Một **chủ trọ** có thể cho nhiều **SinhVien** thuê tại một khu trọ |
| Một **chủ trọ** có thể sở hữu nhiều khu trọ (nếu có thể) |
| Một **Tro** có thể được nhiều người sử dụng trong nhiều lần, và chỉ được sở hữu bởi duy nhất một **ChuTro** |

Bảng : Mô tả mối quan hệ giữa các thuộc tính

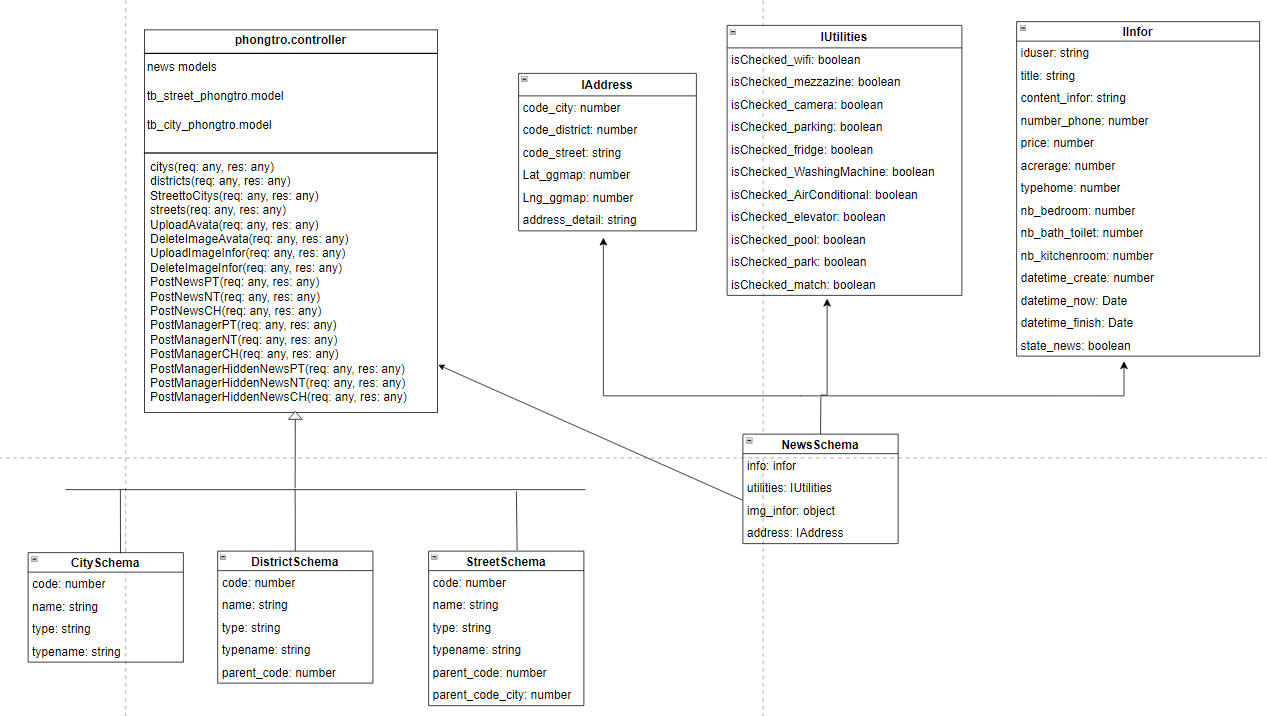
**Biểu đồ lớp**



Hình : Biểu đồ lớp cho HomePageController



Hình : Biểu đồ lớp cho userController



Biểu đồ cho lớp phongtro.controller

### Chi tiết các bảng

**Phương thức dành cho HomePage.controller**

* news models
* tb\_street\_phongtro.model
* tb\_city\_phongtro.model
* citys(req: any, res: any)
* districts(req: any, res: any)
* StreettoCitys(req: any, res: any)
* streets(req: any, res: any)
* UploadAvata(req: any, res: any)
* DeleteImageAvata(req: any, res: any)
* UploadImageInfor(req: any, res: any)
* DeleteImageInfor(req: any, res: any)
* PostNewsPT(req: any, res: any)
* PostNewsNT(req: any, res: any)
* PostNewsCH(req: any, res: any)
* PostManagerPT(req: any, res: any)
* PostManagerNT(req: any, res: any)
* PostManagerCH(req: any, res: any)
* PostManagerHiddenNewsCH(req: any, res: any)
* PostManagerHiddenNewsPT(req: any, res: any)
* PostManagerHiddenNewsNT(req: any, res: any)

**Phương thức dành cho phongtro.controller**

* news models
* tb\_street\_phongtro.model
* tb\_city\_phongtro.model
* citys(req: any, res: any)
* districts(req: any, res: any)
* StreetCitys(req: any, res: any)
* streets(req: any, res: any)
* UploadAvata(req: any, res: any)
* DeleteImageAvata(req: any, res: any)
* UploadImageInfor(req: any, res: any)
* DeleteImageInfor(req: any, res: any)
* PostNewsPT(req: any, res: any)
* PostNewsNT(req: any, res: any)
* PostNewsCH(req: any, res: any)
* PostManagerPT(req: any, res: any)
* PostManagerNT(req: any, res: any)
* PostManagerCH(req: any, res: any)
* PostManagerHiddenNewsPT(req: any, res: any)
* PostManagerHiddenNewsNT(req: any, res: any)
* PostManagerHiddenNewsCH(req: any, res: any)

**Các phương thức dành cho user.controller**

* news models
* feedback
* login(req: any, res: any)
* loginsuccess(req: any, res: any)
* register(req: any, res: any)
* registersuccess(req: any, res: any)
* logout(req: any, res: any)
* getInforUser(req: any, res: any)
* accuracyPhone\_NB(req: any, res: any)
* accuracyPhone\_NB\_key\_OTP(req: any, res: any)
* changepasswordlogin(req: any, res: any)
* xFeedBack(req: any, res: any)
* GetFeedBack(req: any, res: any)
* getInforUserById(req: any, res: any)
* getInforUserEdit(req: any, res: any)
* UploadUserEditAvatar(req: any, res: any)
* OpenAvaterUser(req: any, res: any)
* EditedInforUser(req: any, res: any)

**Bảng Infor: dành cho lớp userController**

* firstname: string
* lastname: string
* male\_female: boolean (Giới tính)
* img\_avatar: string (Ảnh đại diện)

**Bảng ILocal**

* username: string
* password: string
* email: string

**Bảng IAddress**

* code\_city: number
* code\_district: number
* Lat\_ggmap: number
* Lng\_ggmap: number
* address\_detail: string (chi tiết về địa chỉ)

**Bảng IUtilities**

* isChecked\_wifi: boolean
* isChecked\_mezzazine: boolean
* isChecked\_camera: boolean
* isChecked\_fridge: boolean
* isChecked\_WashingMachine: boolean
* isChecked\_AirConditional: boolean
* isChecked\_pool: boolean
* isChecked\_match: boolean

**Bảng IInfor: dành cho lớp phongtro.controller**

* iduser: string
* title: string (nội dung liên quan trong gói thông tin)
* content\_infor: string
* price: number (Giá phòng)
* nb\_bedroom: number
* nb\_bath\_toilet: number
* nb\_kitchenroom: number
* typehome: number (loại hình cho thuê: thuê nguyên căn, phòng trọ)
* datetime\_create: number
* datetime\_now: Date
* datetime\_finish: Date

**Bảng NewsSchema**

* code: number
* name: string
* type: string
* typename: string

**Bảng FeedBackSchema**

* info: Infor
* utilities: IUtilities
* img\_infor: object (hình ảnh chi tiết trong các news)
* address: IAddress

**Bảng CitySchema**

* code: number
* name: string: (tên thành phố hoặc tỉnh thành)
* type:string
* typename: string

**Bảng StreetSchema**

* Code: number
* name: string (tên đường)
* type:string
* typename: string

**Bảng DistrictSchema**

* Code: number
* name: string: Địa chỉ cụ thể
* type:string
* typename: string
* parent\_code: number

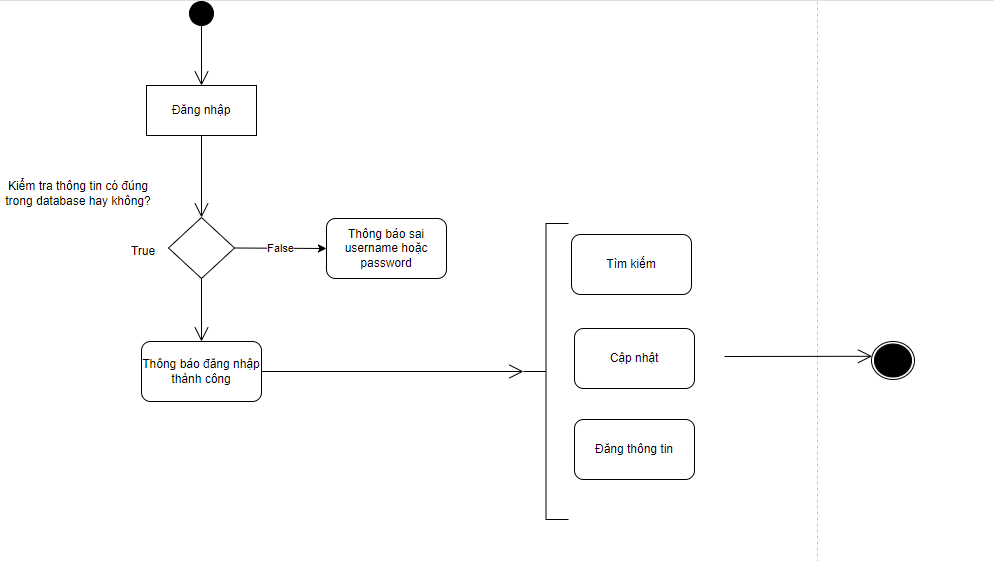
### Hợp đồng hoạt động

|  |  |
| --- | --- |
| **Contract Operation ID** | **Mô tả** |
| CO-01 | Đăng nhập |
| CO-02 | Đăng ký |
| CO-03 | Hiển thị thông tin |
| CO-04 | Cập nhật( thêm, xóa, sửa) |
| CO-05 | Đăng xuất |

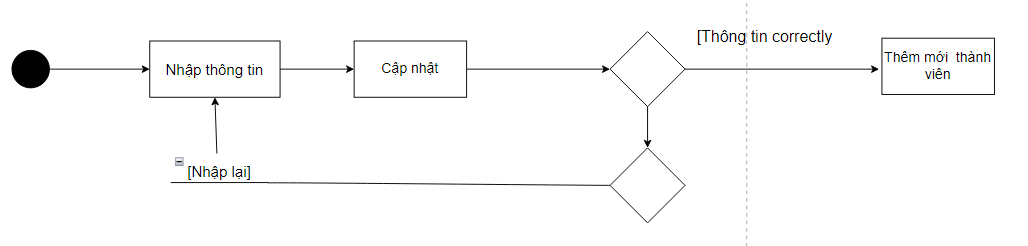
Bảng : Bảng Contract Operation của web

### Sơ đồ hoạt động (Activity Diagram)

Đây là bản vẽ tập trung vào mô tả các hoạt động, luồng xử lý bên trong hệ thống. Nó có thể được sử dụng để mô tả các qui trình nghiệp vụ trong hệ thống, các luồng của một chức năng hoặc các hoạt động của một đối tượng.

******

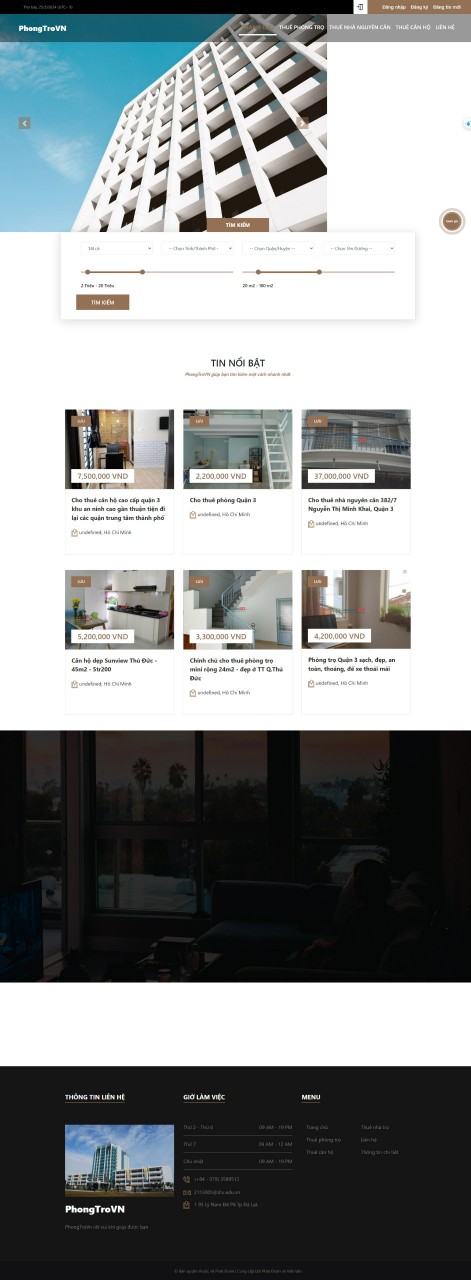
Hình : Sơ đồ hoạt động chung cho toàn hệ thống

******

Hình : Sơ đồ hoạt động cho thao tác đăng ký

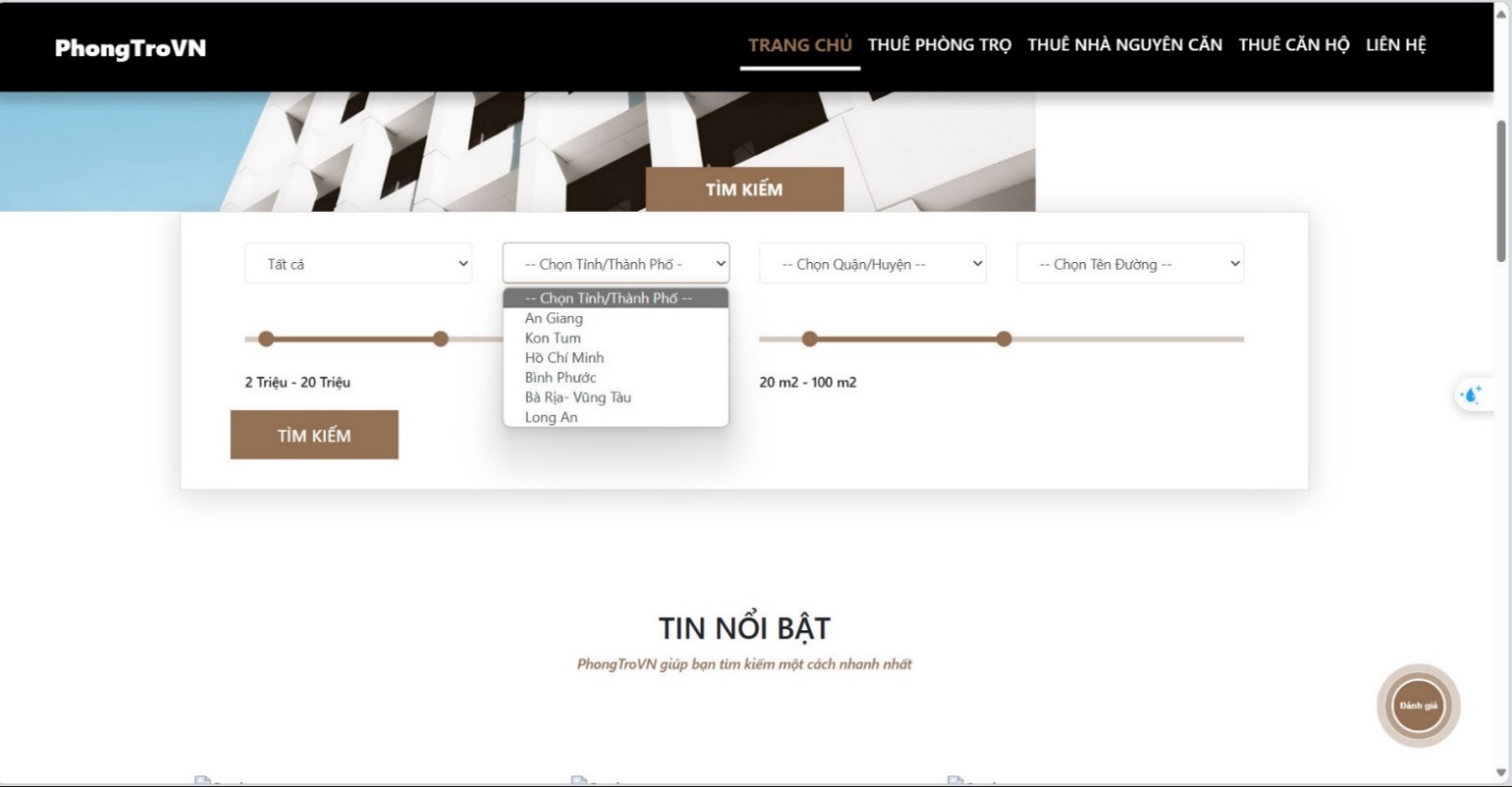
# CHƯƠNG 3: KẾT QUẢ ĐẠT ĐƯỢC

## Giao diện



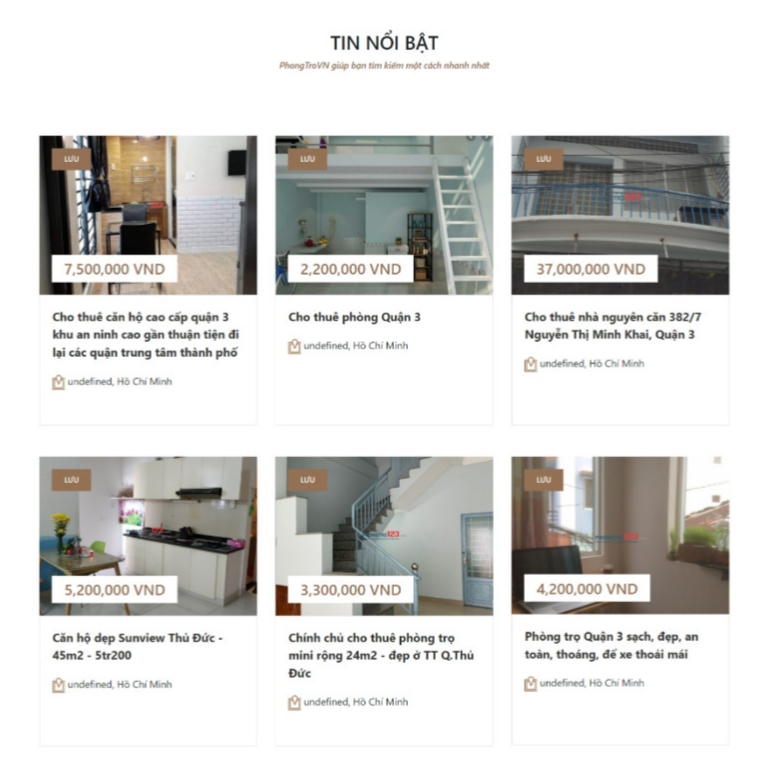
Hình : Giao diện trang chủ

Bao gồm các thanh điều hướng giúp người dùng chuyển sang các hình thức (phòng trọ, nhà nguyên căn, căn hộ). Phía header của trang web sẽ có ba button thực hiện việc đăng nhập đăng ký, truy cập đến các task khác.



Hình : Công cụ tìm kiếm theo ba yếu tố chính

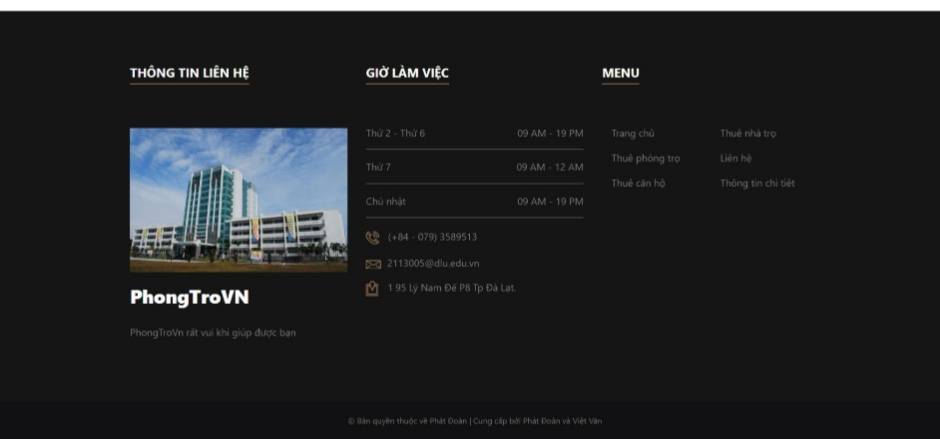
Đây là nơi cho phép tìm kiếm thông tin ngay ở UC2. Bao gồm việc tìm kiếm trên ba yếu tố chính:vị trí, giá, khoảng cách.



Hình : Các bài đăng nổi bật

Tác vụ tin nổi bật và tin phòng trọ cho phép người dùng có thể thấy ngay những thông tin liên quan, giúp việc tra cứu trở nên dễ dàng hơn.

Thanh Footer của trang web



Hình : Phần Footer của web

## Các test scenario

|  |  |  |
| --- | --- | --- |
| **Test Scenario** | **Requirement** | **Test Scenario Description** |
| ID001 | S1 | Kiểm thử giao diện |
| ID002 | S2 | Kiểm thử đăng nhập |
| ID003 | S3 | Kiểm thử chức năng đăng ký |
| ID004 | S4 | Kiểm thử lấy lại mật khẩu |
| ID005 | S5 | Kiểm thử chức năng quản lý các trọ |
| ID006 | S6 | Kiểm thử chức năng cập nhật (thêm, xóa sửa) |

Bảng : Các Scenario chung cho toàn bộ trang web

Test case 1: Kiểm thử giao diện

|  |  |  |
| --- | --- | --- |
| **S #** | **Prerequisites:** | **Các tiêu chí** |
| 1 | Access to Chrome Browser | font chữ: 20px in đậm cho các tiêu đề nổi bật |
| 2 | Navigate to some form in web | Màn hình 1920x1080 |
| 3 | Check size of button, image, font-size, font-weight |  |

Bảng : Điều kiện tiên quyết với thao tác kiểm thử giao diện

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step#** | **Step Details** | **Expected Results** | **Actual Results** | **Pass / Fail / Not executed / Suspended** |
|  |  |  |  |  |  |
| 1 | Access to website | Page should open | As Expected | Pass |  |
| 2 | Changing screen size | website changes into smaller screen | As Expected | Not executed |  |
| 3 | View on smartphone | website changes into type web app | As Expected | Suspended |  |

Test case 2: Kiểm thử đăng nhập

|  |  |
| --- | --- |
| **S #** | **Prerequisites:** |
| 1 | Truy cập thành công trên Google Chrome |
| 2 | Chuyển đến form đăng nhập |
| 3 | Username và mật khẩu phải trùng khớp |

Bảng : Điều kiện tiên quyết với thao tác đăng nhập

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step#** | **Step Details** | **Expected Results** | **Actual Results** | **Pass / Fail / Not executed / Suspended** |
|  |  |  |  |  |  |
| 1 | Nhập username và mật khẩu | Thông tin được đưa vào form | As Expected | Pass |  |
| 2 | Nhấn nút đăng nhập | Người dùng đăng nhập thành công | As Expected | Pass |  |
| 3 | In thông báo | In thông báo đăng nhập thành công, có thể tiếp cận các chức năng khác | As Expected | Pass |  |

Test case 3: Kiểm thử đăng ký

|  |  |  |
| --- | --- | --- |
| **S #** | **Prerequisites** | **Các tiêu chí** |
| 1 | Access to Chrome Browser | Họ tên: Nguyễn Văn A |
| 2 | Valid information | Địa chỉ: 37 Trần Quốc Toản, phường 2, Quận Tân Bình, TP.HCM |
| 3 | Password must contain number, string and special character | Số điện thoại: 0927376893 |
|  |  | email: zomb9998@gmail.com |
|  |  | password: 12fn736@#5&$$# |

Bảng : Điều kiện tiên quyết với thao tác đăng ký

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step#** | **Step Details** | **Expected Results** | **Actual Results** | **Pass / Fail / Not executed / Suspended** |
|  |  |  |  |  |  |
| 1 | Navigate to Login form | Page should open | As Expected | Pass |  |
| 2 | Enter full name | Information has been entered | As Expected | Pass |  |
| 3 | Enter address | Information has been entered | As Expected | Pass |  |
| 4 | Enter phone number | Information has been entered | As Expected | Pass |  |
| 5 | Navigate to Login form | Information has been entered | As Expected | Pass  Fail => if Prerequisites 3 is wrong |  |
| 6 | Enter full name | Customer can logged in | As Expected | Pass |  |

Test case 4: Chức năng lấy lại mật khẩu

|  |  |  |
| --- | --- | --- |
| **S #** | **Prerequisites** | **Các tiêu chí'** |
| 1 | Click on Forgot Password => navigate to scence change password | Họ tên: Nguyễn Văn A |
| 2 | Verify that the Current Password, New Password, and Confirm Password fields are displayed on the screen ? | Số điện thoại: 0927376893 |
|  | Vertify that user UI of currently password, new password and verfify\_password is correctly based on database? | Địa chỉ: 37 Trần Quốc Toản, phường 2, Quận Tân Bình, TP.HCM |
|  |  | email: zomb9998@gmail.com |
|  |  | Mật khẩu cũ: 12fn736@#5&$$# |
|  |  | Mật khẩu mới: cocaimatkhaucungquen |

Bảng : Điều kiện tiên quyết cho thao tác lấy lại mật khẩu

## Kết quả kiểm thử thủ công và tự động

## Mã nguồn

Đường link dự án: <https://github.com/LeHaHieuNghia/QL_NhaTro_VN.git>

### Client

#### App.jsx

import React, { Component } from 'react';

import { BrowserRouter as Router} from "react-router-dom";

import './App.css';

import Header from './component/header/header';

import RouterWeb from './router';

import Newshome from './component/newshome/newshome';

import Footer from './component/footer/footer';

import ChangePassword from './component/newshome/change\_password/change\_password';

class App extends Component {

constructor(props) {

super(props);

this.state={

nextpage:false,

stateFiterandslide\_img:false,

statefilter:false,

NewsFilter:[],

NameCityFilter:[],

NameDistrictsFilter:[],

clickFindNews:false,

typenews\_menu:'',

}

}

clickMovedOnUsertoApp=()=>{

this.setState({

nextpage:!this.state.nextpage

});

}

clickPostNewstoApp=(nextpage)=>{

if(nextpage) this.setState({

nextpage:!this.state.nextpage

});

}

ClickGoHome=()=>{

this.setState({

nextpage:!this.state.nextpage,

stateFiterandslide\_img:false

})

}

NewsDetailtoApp=()=>{

this.setState({

stateFiterandslide\_img:true

});

}

StateFiterandslide=()=>{

this.setState({

stateFiterandslide\_img:false

});

}

// Trạng tháng của filter

StateFiterTyhomeNews=()=>{

this.setState({

statefilter:true,

NewsFilter:[],

clickFindNews:false,

typenews\_menu:'',

NameCityFilter:[],

NameDistrictsFilter:[]

});

}

StateFiterTyhomeNews\_F=(value)=>{

this.setState({

statefilter:false,

typenews\_menu:value

});

}

GetNewsFilter= (NewsFilter,NameCityFilter,NameDistrictsFilter,clickFindNews)=>{

this.setState({

NewsFilter:NewsFilter,

clickFindNews:clickFindNews,

NameCityFilter:NameCityFilter,

NameDistrictsFilter:NameDistrictsFilter

})

}

ListNewsReset=()=>{

this.setState({

NewsFilter:[],

clickFindNews:false,

NameCityFilter:[],

NameDistrictsFilter:[]

});

}

render() {

return (

<div className="App">

<Router>

{

!this.state.nextpage?

<div>

<Header clickPostNewstoApp={this.clickPostNewstoApp}

clickMovedOnUsertoApp={this.clickMovedOnUsertoApp}

stateFiterandslide\_imgApp={this.state.stateFiterandslide\_img} // (True/Faile) Hidden Fitter and slideShow Image

StateFiterandslide\_FuncApp={this.StateFiterandslide} // Show Fitter and slideShow Image

StateFiterTyhomeNews\_TF={this.state.statefilter} // (True/Faile) Hidden Fitter typenews

GetNewsFiltertoApp={this.GetNewsFilter} // get News Filter (Lấy tin từ tìm kiếm)

GetTypeNewstoApp={this.state.typenews\_menu} // Get value type news filter

/>

<RouterWeb NewsDetailtoApp={this.NewsDetailtoApp}

stateFiterandslide\_imgApp={this.state.stateFiterandslide\_img} // (True/Faile) Hidden Fitter and slideShow Image

StateFiterTyhomeNewstoApp={this.StateFiterTyhomeNews} // Show Fitter and slideShow Image

StateFiterTyhomeNews\_FtoApp={this.StateFiterTyhomeNews\_F}

StateNextPage={this.state.nextpage} //Giử trạng thái next page Home sang HomeNews

GetNameCityFiltertoApp={this.state.NameCityFilter} // Truyền Array chứa tên thành phố khi tìm kiếm

GetNameDistrictsFiltertoApp={this.state.NameDistrictsFilter} // Truyền Array chứa tên quận huyện khi tìm kiếm

NewsFiltertoApp={this.state.NewsFilter} // News Filter

clickFindNewstoApp={this.state.clickFindNews} //(True/Faile) Click button Find News

ListNewsResettoApp={this.ListNewsReset}

/>

<ChangePassword/>

<Footer/>

</div>

:

<Newshome ClickGoHome={this.ClickGoHome}

StateNextPage={this.state.nextpage}/> // (True/Faile) next page go manager

}

</Router>

</div>

);

}

}

export default App;

#### user.model.js

import mongoose, { Document, Schema, Model, model } from "mongoose"

interface IUser extends Document {

infor: {

firstname: string

lastname: string

male\_female: {

male: boolean

female: boolean

}

img\_avatar: string

}

local: {

username: string

password: string

email: string

}

role: string

number\_phone: number

}

const UserSchema = new Schema(

{

infor: {

firstname: {

type: String,

required: true,

},

lastname: {

type: String,

required: true,

},

male\_female: {

male: {

type: Boolean,

default: true,

},

female: {

type: Boolean,

default: false,

},

},

img\_avatar: {

type: String,

default: "",

},

},

local: {

username: {

type: String,

required: true,

},

password: {

type: String,

required: true,

},

email: {

type: String,

required: true,

},

},

role: {

type: String,

default: "MEMBER",

},

number\_phone: {

type: Number,

default: "",

},

},

{

versionKey: false,

},

)

export default model("User", UserSchema)

#### Index.ts

import express from 'express';

import mongoose from 'mongoose';

import passport from 'passport';

import bodyParser from 'body-parser';

import session from 'express-session';

import userRouter from './routers/user.router';

import homeRouter from './routers/phongtro.router';

import homePageRouter from './routers/home.router';

import flash from 'connect-flash';

const app = express();

/\* Khai báo để sử dụng kịch bản passport \*/

import './config/passport.config';

app.use(session({

name: 'login',

secret : 'secured\_key',

resave : false,

saveUninitialized : false,

cookie:{

maxAge:1000\*60\*15

}

}))

app.use([

bodyParser.json(),

bodyParser.urlencoded({

extended: true,

})

]);

app.use(flash());

app.use(passport.initialize());

app.use(passport.session());

app.use(express.static("public"));

app.use('/nguoi-dung',userRouter);

app.use('/phong-tro',homeRouter);

app.use('/trang-chu',homePageRouter);

app.listen(8000, async ()=>{

await mongoose.connect('mongodb://localhost:51734', {

dbName: "Database\_PhongTroVN"

});

console.log("Listen port 8000");

});

#### googlemap.js

import React, { Component } from 'react';

import GoogleMapReact from 'google-map-react';

import icon\_ggmap from './image\_ggmap/icon\_ggmap.png';

const AnyReactComponent = ({ text }) => <div>{text}</div>;

class GoogleMap extends Component {

constructor(props) {

super(props);

this.state={

center:{

lat:10.8365648,

lng:106.7560183

},

zoom:16

}

}

async componentDidMount(){

if (navigator.geolocation) {

await navigator.geolocation.getCurrentPosition(this.getLocationForBrowser);

} else {

alert("Geolocation is not supported by this browser.");

}

}

// getLocation=()=>{

// if (navigator.geolocation) {

// navigator.geolocation.getCurrentPosition(this.getLocationForBrowser);

// } else {

// alert("Geolocation is not supported by this browser.");

// }

// }

getLocationForBrowser=(position)=>{

this.setState({

center:{

lat:position.coords.latitude,

lng:position.coords.longitude

}

})

this.props.getLocationtoNewsNews(this.state.center)

}

handleApiLoaded=(map, maps)=>{

new maps.Marker({

position: this.state.center,

map,

title: 'Home Your!',

icon:{

url: icon\_ggmap,

scaledSize:

{

width:40,height:40

}

}

});

}

render() {

return (

<div style={{ height: '500px', width: '100%' }}>

<GoogleMapReact

bootstrapURLKeys={{key:'AIzaSyDLhm8DHP3A6kMCIsiwQWUU-pX5hSbyaQo'}}

defaultCenter={this.state.center}

defaultZoom={this.state.zoom}

yesIWantToUseGoogleMapApiInternals

onGoogleApiLoaded={({ map, maps }) => this.handleApiLoaded(map, maps)}

>

</GoogleMapReact>

<AnyReactComponent

lat={this.state.center.lat}

lng={this.state.center.lng}

/>

</div>

);

}

}

export default GoogleMap;

#### Change\_password.js

import React, { Component } from 'react';

import axios from 'axios';

import img\_icon\_password from './icon\_change\_password/lock.png';

class ChangePassword extends Component {

    constructor(props) {

        super(props);

        this.state={

            passwordold:'',

            password:'',

            verifypassword:'',

            message:'',

        }

    }

    handleChangeField=()=>{

        this.setState({

           password:this.refs.password.value,

           passwordold:this.refs.passwordold.value,

           verifypassword:this.refs.verifypassword.value,

       })

    }

    SubmitChangePassword=async()=>{

        let {passwordold,password,verifypassword}=this.state;

        if(!passwordold || !password || !verifypassword){

            this.setState({

                message:"Vui lòng nhập đầy đủ thông tin"

            })

        }else{

            if(password===verifypassword)

           {

            await axios.post('/nguoi-dung/doi-mat-khau',{

                passwordold,password

             },{headers: {'Accept': 'application/json'}})

            .then(res => {

                this.setState({

                    message:res.data.message,

                    result:res.data.result,

                });

                 if(res.data.result){

                    const close = document.getElementById("IDChangePassword");

                    setTimeout(()=>{close.click()},1000\*1);

                 }

            })

            .catch( (error) => console.log(error));

           }else{

            this.setState({

                message:"Nhập lại mật khẩu không khớp",

            })

        }

        }

    }

    ClickCloseChangePassword=()=>{

        this.setState({

            passwordold:'',

            password:'',

            verifypassword:'',

            message:''

        })

    }

    render() {

        var KTL=true;

        if(this.state.message.length>0){

            KTL=false;

        }

        return (

            <div className="modal fade" id="modalChangePassword" tabIndex={-1} role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">

                <div className="modal-dialog" role="document">

            <div className="modal-content">

            <div className="modal-header text-center">

                <h4 className="modal-title w-100 font-weight-bold">Đổi mật khẩu</h4>

                <button type="button" className="close"  data-dismiss="modal" aria-label="Close">

                <span aria-hidden="true" id="IDChangePassword" onClick={this.ClickCloseChangePassword}>×</span>

                </button>

            </div>

            <div className="modal-body mx-3">

            {!KTL &&

                      <p  className="error\_message-login">{ this.state.message}

                      </p>

            }

                <div className="md-form mb-5 row">

                    <div className="col-md-2 col-sm-2 col-xs-3 icon\_username">

                        <img src={img\_icon\_password} alt="icon"/>

                    </div>

                    <div className="col-md-10 col-sm-8 col-xs-9 inputusername">

                            <input type="password"

                            className="form-control "

                            placeholder="Mật khẩu cũ"

                            ref="passwordold"

                            onChange={this.handleChangeField}

                            value={this.state.passwordold}

                           />

                    </div>

                </div>

                <div className="md-form mb-5 row">

                    <div className="col-md-2 col-sm-2 col-xs-3 icon\_password">

                            <img src={img\_icon\_password} alt="icon"/>

                    </div>

                    <div className="col-md-10 col-sm-8 col-xs-9 inputpassword">

                        <input type="password"

                        className="form-control "

                        placeholder="Mật khẩu mới"

                        ref="password"

                        onChange={this.handleChangeField}

                        value={this.state.password}

                        />

                    </div>

                </div>

                <div className="md-form mb-5 row">

                    <div className="col-md-2 col-sm-2 col-xs-3 icon\_password">

                        <img src={img\_icon\_password} alt="icon"/>

                    </div>

                    <div className="col-md-10 col-sm-8 col-xs-9 inputpassword">

                        <input type="password"

                        className="form-control "

                        placeholder="Nhập lại mật khẩu mới"

                        ref="verifypassword"

                        onChange={this.handleChangeField}

                        value={this.state.verifypassword}

                        />

                    </div>

                </div>

            </div>

                <div className="modal-footer d-flex justify-content-center bntdangnhap">

                <input type="button" className="btn btn-default" value="Lưu"  onClick={this.SubmitChangePassword}/>

                </div>

            </div>

    </div>

            </div>

        );

    }

}

export default ChangePassword;

#### sNew\_news,js

import React, { Component } from 'react';

import { Redirect } from 'react-router-dom'

import axios from 'axios';

import img\_icon\_location from '../../home/image\_icon/location.png'

import img\_icon\_price from './news\_image/price.png'

import img\_icon\_space from './news\_image/space.png'

import img\_icon\_phone\_call from './news\_image/phone-call.png'

import './news\_new.css'

import UploadImage from './upload\_image\_news/upload\_image';

import SelectOption from './select\_option\_NT/select\_option\_NT';

import GoogleMap from './google\_map/google\_map';

import Utilities from './utilities/utilities';

class Newsnew extends Component {

constructor(props) {

super(props);

this.state={

//###########\_\_ Get Children Select address\_\_########

city:'',

district:'',

street:'',

//###########\_\_ Get data Select address (to database)\_\_########

citys:[],

districts:[],

streets:[],

//###########\_\_ Get value input address\_\_########

inputmap:'',

open\_selectoption\_NT\_CH:false,

result\_postnews:false,

//################\_\_ Save News \_\_###############

code\_city:'' ,

code\_dictrict:'',

code\_street:'',

number\_home:'',

news\_title:'',

code\_type\_news:'',

news\_content\_infor:'',

price:'',

acreage:'',

url\_Image:'',

url\_Images\_Infor:[],

message\_postnews:'',

price\_format:'',

//################\_\_ Optiion Select Home \_\_########

nb\_bedroom:'',

nb\_bath\_toilet:'',

nb\_kitchenroom:'',

//################\_\_Utilities \_\_########

utilities:'',

//################\_\_Type Home (Thể loại đăng tin) \_\_########

typehome:1,

urltypenews:'',

Lat\_ggmap:'',

Lng\_ggmap:'',

}

}

componentDidMount(){

axios.post('/phong-tro/dang-tin-moi/chon-tinhTP')

.then(res=>this.setState({

citys:res.data.citys

}))

.catch( (error) => console.log(error));

}

submitClickCity= async (e)=>{ // Get value and children submit

var city="";

let parent\_code = e.target.value;

if(e.target.value!== "0") {city=e.target[e.target.selectedIndex].text}

this.setState({

code\_city:parent\_code,

city:city,

inputmap:this.state.number\_home+" "+this.state.street+this.state.district+city

})

await axios.post('/phong-tro/dang-tin-moi/chon-QH',{

parent\_code

},{headers: {'Accept': 'application/json'}})

.then(res => {

this.setState({

districts:res.data.districts

});

})

.catch((error) => console.log(error));

// Select list streets to code city (Lấy mã tỉnh chon ra danh sach duong)

await axios.get(`/phong-tro/dang-tin-moi/danh-sach-duong/${parent\_code}`)

.then(res => {

this.setState({

streets:res.data.streets

});

if(res.data.streets){

this.setState({

number\_home:'',

district:'',

street:'',

inputmap:city

})

}

})

.catch((error) => console.log(error));

if(parent\_code==="0"){

this.setState({

streets:[],

code\_street:'',

inputmap:'',

number\_home:''

});

}

}

sumitClickDictrict= async (e)=>{

var district="";

let parent\_code = e.target.value;

let parent\_code\_city=this.state.code\_city;

if(e.target.value!== 0) {district=e.target[e.target.selectedIndex].text+", "}

this.setState({

district:district,

code\_dictrict:parent\_code,

inputmap:this.state.number\_home+" "+this.state.street+district+this.state.city

})

await axios.post('/phong-tro/dang-tin-moi/chon-Duong',{

parent\_code,

parent\_code\_city

},{headers: {'Accept': 'application/json'}})

.then(res => {

this.setState({

streets:res.data.streets

});

})

.catch( (error) => console.log(error));

}

HandlerInput=()=>{

}

sumitClickStreet= (e)=>{

var street="";

if(e.target.value!== "0") {street=e.target[e.target.selectedIndex].text+", "}

this.setState({

street:street,

code\_street:e.target.value,

inputmap:this.state.number\_home+" "+street+this.state.district+this.state.city

})

}

HandlerInputNumberHome=()=>{

let number\_home=this.refs.number\_home.value;

this.setState({

number\_home:number\_home,

inputmap:this.refs.number\_home.value+" "+this.state.street+this.state.district+this.state.city

})

}

SelectTypeHome=(e)=>{

if(e.target.value==="1"){

this.setState({

typehome: 1,

open\_selectoption\_NT\_CH:false

})

}else if(e.target.value==="2"){

this.setState({

typehome:2,

open\_selectoption\_NT\_CH:true

})

}else{

this.setState({

typehome:3,

open\_selectoption\_NT\_CH:true

})

}

}

getSelectSelectOption=(nb\_bedroom,nb\_bath\_toilet,nb\_kitchenroom)=>{

this.setState({

nb\_bedroom:nb\_bedroom,

nb\_bath\_toilet:nb\_bath\_toilet,

nb\_kitchenroom:nb\_kitchenroom

})

}

getUrlImage\_News=(Url\_Image,NameImages\_Infor)=>{

var Url\_Images\_Infor = [];

NameImages\_Infor.forEach(item => {

Url\_Images\_Infor.push("http://localhost:3001/phong-tro/open\_image/nameimage="+item);

});

this.setState({

url\_Image:Url\_Image,

url\_Images\_Infor:Url\_Images\_Infor

})

}

formatNumber=(num)=> {

return num.toString().replace(/(\d)(?=(\d{3})+(?!\d))/g, '$1,')

}

handleChangeField=()=>{

this.setState({

news\_title:this.refs.title.value,

news\_content\_infor:this.refs.content\_infor.value,

price:this.refs.price.value,

price\_format:this.formatNumber(this.refs.price.value),

acreage:this.refs.acreage.value

})

}

sumitPostNews=async()=>{

let date = new Date();

let date\_format = date.toLocaleDateString();

let date\_n = new Date();

let date\_fn = new Date(date\_n.setMonth(date\_n.getMonth()+1));

let date\_finish\_format=date\_fn.toLocaleDateString();

let title=this.state.news\_title;

let content\_infor=this.state.news\_content\_infor;

let number\_phone=this.props.GetPhone\_Number;

let price=this.state.price;

let acreage=this.state.acreage;

let img\_avatar=this.state.url\_Image;

let img\_infor=this.state.url\_Images\_Infor;

let code\_city=this.state.code\_city;

let code\_dictrict=this.state.code\_dictrict;

let code\_street=this.state.code\_street;

let address\_detail =this.state.inputmap;

let typehome=this.state.typehome;

let date\_now=date\_format;

let date\_finish=date\_finish\_format;

let utilities= this.state.utilities

let Lat\_ggmap=this.state.Lat\_ggmap;

let Lng\_ggmap=this.state.Lng\_ggmap;

if(this.state.typehome===1){

await axios.post('/phong-tro/dang-tin-moi/phong-tro',{

title,content\_infor,number\_phone,price,acreage,img\_avatar,img\_infor,

code\_city,code\_dictrict,code\_street,Lat\_ggmap,Lng\_ggmap,typehome,

utilities,date\_now,date\_finish,address\_detail

},{headers: {'Accept': 'application/json'}})

.then(res => {

if(res.data.result)

{

this.setState({

result\_postnews:true,

message\_postnews:res.data.message,

urltypenews:'/nguoi-dung/quan-ly-tin-dang/phong-tro'

});

}else{

this.setState({

result\_postnews:false,

message\_postnews:res.data.message

});

}

})

.catch( (error) => console.log(error));

}else if(this.state.typehome===2){

let nb\_bedroom=this.state.nb\_bedroom;

let nb\_bath\_toilet=this.state.nb\_bath\_toilet;

let nb\_kitchenroom=this.state.nb\_kitchenroom;

axios.post('/phong-tro/dang-tin-moi/nha-tro',{

title,content\_infor,number\_phone,price,acreage,img\_avatar,img\_infor,

code\_city,code\_dictrict,code\_street,Lat\_ggmap,Lng\_ggmap,nb\_bedroom,nb\_bath\_toilet,

nb\_kitchenroom,utilities,date\_now,date\_finish,typehome,address\_detail

},{headers: {'Accept': 'application/json'}})

.then(res => {

if(res.data.result)

{

this.setState({

result\_postnews:true,

message\_postnews:res.data.message,

urltypenews:'/nguoi-dung/quan-ly-tin-dang/nha-tro'

});

}else{

this.setState({

result\_postnews:false,

message\_postnews:res.data.message

});

}

})

.catch( (error) => console.log(error));

}

else{

let nb\_bedroom=this.state.nb\_bedroom;

let nb\_bath\_toilet=this.state.nb\_bath\_toilet;

let nb\_kitchenroom=this.state.nb\_kitchenroom;

axios.post('/phong-tro/dang-tin-moi/can-ho',{

title,content\_infor,number\_phone,price,acreage,img\_avatar,img\_infor,

code\_city,code\_dictrict,code\_street,Lat\_ggmap,Lng\_ggmap,nb\_bedroom,nb\_bath\_toilet,

nb\_kitchenroom,utilities,date\_now,date\_finish,typehome,address\_detail

},{headers: {'Accept': 'application/json'}})

.then(res => {

if(res.data.result)

{

this.setState({

result\_postnews:true,

message\_postnews:res.data.message,

urltypenews:'/nguoi-dung/quan-ly-tin-dang/can-ho'

});

}else{

this.setState({

result\_postnews:false,

message\_postnews:res.data.message

});

}

})

.catch( (error) => console.log(error));

}

}

// Check value utilities (Lấy giá trị của các tiện ích)

getValueUtilities=(valueisCheck)=>{

console.log(valueisCheck.isChecked\_wifi)

this.setState({

utilities:valueisCheck

})

}

getLocation=(value)=>{

this.setState({

Lat\_ggmap:value.lat,

Lng\_ggmap:value.lng,

})

}

render() {

var KTL=true;

if(this.state.message\_postnews){

KTL=false;

}

if(this.state.result\_postnews) return <Redirect to={this.state.urltypenews}/>

return (

<div className="container-fluid">

<div className="row alert\_messager">

{!KTL && <div className="alert alert-danger">{this.state.message\_postnews}</div>}

</div>

<div className="row">

<div className="col-md-12 col-sm-12 col-xs-12 tieudepage\_mg">

<h2 className="tieudepage\_mg-h2">Đăng tin mới</h2>

<p>Thông tin càng chính xác giúp cho người thuê một cách tốt nhất</p>

</div>

</div>

<div className="ggmappage wow fadeInUp" data-wow-delay="0.1s">

<div className="row">

<div className="col-md-12 col-sm-12 col-xs-12 ggmappage\_mg">

<h2 className="ggmappage\_mg-h2" >Địa chỉ cho thuê</h2>

</div>

</div>

<div className="row">

<div className="col-md-3 col-sm-3 col-xs-12">

<div className="form-group">

<select className="form-control nice-select wide select\_item" name="Haha" onChange={e=>this.submitClickCity(e)}>

<option value='0' name="0">-- Chọn Tỉnh/Thành Phố --</option>

{

this.state.citys.map((item,index)=>

<option key={index} value={item.code} name={item.name}>{item.name}</option>)

}

</select>

</div>

</div>

<div className="col-md-3 col-sm-3 col-xs-12">

<div className="form-group">

<select className="form-control nice-select wide select\_item" onChange={e=>this.sumitClickDictrict(e)}>

<option value='0'>-- Chọn Quận/Huyện --</option>

{

this.state.districts.map((item,index)=>

<option key={index} value={item.code} typename={item.typename}>{item.typename}</option>

)

}

</select>

</div>

</div>

<div className="col-md-3 col-sm-3 col-xs-12">

<div className="form-group">

<select className="form-control nice-select wide select\_item" onChange={e=>this.sumitClickStreet(e)}>

<option value='0'>-- Chọn Tên Đường --</option>

{

this.state.streets.map((item,index)=>

<option key={index} value={item.code}>{item.typename}</option>

)

}

</select>

</div>

</div>

<div className="col-md-3 col-sm-3 col-xs-12">

<input className="input\_so\_nha"

placeholder="Số Nhà"

value={this.state.number\_home}

ref="number\_home"

onChange={this.HandlerInputNumberHome}

/>

</div>

</div>

<div className="row">

<div className="col-md-1 col-sm-1 col-xs-2 exact\_icon\_location">

<img src={img\_icon\_location} alt="icon location"/>

</div>

<div className="col-md-11 col-sm-11 col-xs-10 exact\_input\_location">

<input className="exact\_input\_location-input"

placeholder="Địa chỉ chính xác"

value={this.state.inputmap}

ref="inputmap"

onChange={this.HandlerInput}

disabled/>

</div>

</div>

<div className="row">

<div className="col-md-12 col-sm-12 col-xs-12 google\_map">

<GoogleMap getLocationtoNewsNews={this.getLocation}/>

</div>

</div>

</div>

<div className="row info\_news wow fadeInUp" data-wow-delay="0.1s">

<div className="col-md-12 col-sm-12 col-xs-12 info\_news\_div">

<h2 className="info\_news\_div-h2">Thông tin mô tả</h2>

</div>

<div className="col-md-8 col-sm-8 col-xs-12">

<input className="infor\_news\_input\_td"

ref="title"

value={this.state.news\_title}

onChange={this.handleChangeField}

placeholder="Tiêu đề"/>

</div>

<div className="col-md-4 col-sm-4 col-xs-12">

<div className="col-md-10 col-sm-10 col-xs-12">

<select className="form-control nice-select wide select\_item" onChange={e=>this.SelectTypeHome(e)}>

<option value="1">Thuê phòng trọ</option>

<option value="2">Thuê nhà nguyên căn</option>

<option value="3">Thuê căn hộ</option>

</select>

</div>

</div>

<div className="row content\_news">

<div className="col-md-8 col-sm-8 col-xs-12 content\_news">

<textarea

ref="content\_infor"

value={this.state.news\_content\_infor}

onChange={this.handleChangeField}

placeholder="Mô tả thông tin nhà trọ"/>

</div>

<div className="col-md-4 col-sm-4 col-xs-12 content-warning">

<div className="alert alert-warning" role="alert" >

<div className="row titel-warning">

<h4>Lưu ý khi đăng tin</h4>

</div>

<div className="row body-warning">

<ul>

<li>Nội dung phải viết bằng tiếng Việt có dấu</li>

<li>Tiêu đề tin không dài quá 100 kí tự</li>

<li>Các bạn nên điền đầy đủ thông tin vào các mục để tin đăng có hiệu quả hơn.</li>

<li> Để tăng độ tin cậy và tin rao được nhiều người quan tâm hơn, hãy sửa vị trí tin rao của bạn trên bản đồ bằng cách kéo icon tới đúng vị trí của tin rao.</li>

<li>Tin đăng có hình ảnh rõ ràng sẽ được xem và gọi gấp nhiều lần so với tin rao không có ảnh. Hãy đăng ảnh để được giao dịch nhanh chóng!</li>

</ul>

</div>

</div>

</div>

</div>

<div className="col-md-4 col-sm-4 col-xs-4 ">

<div className="row">

<div className="col-md-2 col-sm-2 col-xs-4 icon\_option">

<img src={img\_icon\_phone\_call} alt="icon phone"/>

</div>

<div className="col-md-10 col-sm-10 col-xs-8 input\_option">

<input className="content\_news\_ip"

placeholder="Số điện thoại"

value={"0"+ this.props.GetPhone\_Number}

disabled/>

</div>

</div>

</div>

<div className="col-md-4 col-sm-4 col-xs-4">

<div className="row">

<div className="col-md-2 col-sm-2 col-xs-2 icon\_option">

<img src={img\_icon\_price} alt="icon price"/>

</div>

<div className="col-md-6 col-sm-6 col-xs-6 input\_option">

<input className="content\_news\_ip"

ref="price"

value={this.state.price}

onChange={this.handleChangeField}

placeholder="Giá tiền"/>

</div>

{ this.state.price\_format &&

<div className="col-md-4 col-sm-6 col-xs-4 row price\_format">

<div className="row price\_format-label">

<label>Số tiền:</label>

</div>

<div className="row price\_format-span">

<span>{this.state.price\_format ? this.state.price\_format+ " VND" : " "}</span>

</div>

</div>

}

</div>

</div>

<div className="col-md-4 col-sm-4 col-xs-4">

<div className="row">

<div className="col-md-2 col-sm-2 col-xs-4 icon\_option">

<img src={img\_icon\_space} alt="icon space"/>

</div>

<div className="col-md-10 col-sm-10 col-xs-8 input\_option">

<input className="content\_news\_ip"

ref="acreage"

value={this.state.acreage}

onChange={this.handleChangeField}

placeholder="Diện tích"/>

<label className="input\_option-acreage" >m2</label>

</div>

</div>

</div>

{this.state.open\_selectoption\_NT\_CH && <SelectOption getSelectSelectOption={this.getSelectSelectOption}/> }

</div>

<Utilities typehome={this.state.typehome} open\_selectoption\_NT\_CH={this.state.open\_selectoption\_NT\_CH} getValueUtilities={this.getValueUtilities}/>

<UploadImage getUrlImage\_News={this.getUrlImage\_News} PostNewsResult={this.state.result\_postnews}/>

<div className="col-md-12 col-sm-12 col-xs-12">

<button className="btn\_PostNews" onClick={this.sumitPostNews}>Đăng tin</button>

</div>

</div>

);

}

}

export default Newsnew;

#### HomePage.Controller.js

var News =require('../models/News/news.model');

var Citys = require('../models/tb\_city.phongtro.model');

var Districts = require('../models/tb\_district.phongtro.model');

module.exports.News\_All= async (req,res)=>{

try{

await News.find({"infor.state\_news":1}).sort({"infor.date\_now":-1}).limit(6).exec(

(err, result)=> {

if(err) console.log(err);

res.json({

All\_News:result

})

})

}catch(err){

res.json({

result:false,

message:err

})

}

}

module.exports.News\_Detail= async (req,res)=>{

try{

await News.find({"\_id":req.params.id},(err,result)=>{

if(err) console.log(err);

res.json({

news:result

})

})

}catch(err){

res.json({

result:false,

message:err

})

}

}

module.exports.GetNameCity= async (req,res)=>{

try{

await Citys.findOne({"code":req.params.code\_city},(err, result)=> {

if(err) console.log(err);

res.json({

NameCity:result.name

})

})

}catch(err){

res.json({

result:false,

message:err

})

}

}

module.exports.GetNameDictrict= async (req,res)=>{

try{

await Districts.findOne({"code":req.params.code\_dictrict},(err, result)=> {

if(err) console.log(err);

res.json({

NameDistricts:result.typename

})

})

}catch(err){

res.json({

result:false,

message:err

})

}

}

module.exports.NewsNears= async (req,res)=>{

try{

await News.find({"address.code\_city":req.params.code\_city,"address.code\_dictrict":req.params.code\_dictrict},(err, result)=> {

if(err) console.log(err);

res.json({

NewsNears:result

})

})

}catch(err){

res.json({

result:false,

NewsNears:'',

message:err

})

}

}

module.exports.News\_RoomHome= async (req,res)=>{

try{

await News.find({"infor.state\_news":1,"infor.typehome":1}).sort({"infor.date\_now":-1}).limit(6).exec(

(err, result)=> {

if(err) console.log(err);

res.json({

NewsRoom:result

})

})

}catch(err){

res.json({

result:false,

message:err

})

}

}

module.exports.News\_HouseHome=async (req,res)=>{

try{

await News.find({"infor.state\_news":1,"infor.typehome":2}).sort({"infor.date\_now":-1}).limit(6).exec(

(err, result)=> {

if(err) console.log(err);

res.json({

HouseHome:result

})

})

}catch(err){

res.json({

result:false,

message:err

})

}

}

module.exports.News\_ApartmentHome=async (req,res)=>{

try{

await News.find({"infor.state\_news":1,"infor.typehome":3}).sort({"infor.date\_now":-1}).limit(6).exec(

(err, result)=> {

if(err) console.log(err);

res.json({

ApartmentHome:result

})

})

}catch(err){

res.json({

result:false,

message:err

})

}

}

module.exports.NewsFilter=async (req,res)=>{

//let{code\_city,code\_dictrict,code\_street,typehome,valuePrice,valueAcreage}

try{

let{PriceMin,PriceMax,AcreageMin,AcreageMax}=req.body;

await News.find({"infor.price":{ $gte: PriceMin, $lte: PriceMax},"infor.acreage":{ $gte: AcreageMin, $lte: AcreageMax},"infor.state\_news":1}).exec(

(err, result)=> {

if(err) console.log(err);

res.json({

NewsFilter:result

})

})

}catch(err){

console.log(err);

res.json({

result:false,

message:err

})

}

}

#### App.test.js

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';

it('renders without crashing', () => {

  const div = document.createElement('div');

  ReactDOM.render(<App />, div);

  ReactDOM.unmountComponentAtNode(div);

});

#### Header.js

import React, { Component } from 'react';

import './header.css'

import Datenow from './datenow/datenow';

import img\_icon\_login from '../header/image\_header/icon\_login.png'

import LoginRegister from '../login\_\_register/login\_\_register';

import Menu from '../menu/menu';

import SlideHeader from './slide\_header/slide\_header';

import Filter from './filter/filter';

class Header extends Component {

    render() {

        return (

            <div className="container-fluid">

                <div className="row header1">

                    <div className="col-md-6 col-sm-6 col-xs-6 header1\_info\_left" >

                        <Datenow/>

                    </div>

                    <div className="col-md-6 col-sm-6 col-xs-6 header1\_info\_right">

                        <LoginRegister clickPostNewstoApp={this.props.clickPostNewstoApp}  clickMovedOnUsertoApp={this.props.clickMovedOnUsertoApp}/>

                        <div className="header1\_info\_right\_\_div-img">

                             <img  src={img\_icon\_login} alt="icon\_login"/>

                        </div>

                    </div>

                </div>

                <div className="row header2">

                    <div className="row header2-slide-menu" id={!this.props.stateFiterandslide\_imgApp ? "header2-menu" :""}>

                       <Menu StateFiterandslide\_FuncApp={this.props.StateFiterandslide\_FuncApp}

                       stateFiterandslide\_imgApp={this.props.stateFiterandslide\_imgApp}/>

                    </div>

                 {!this.props.stateFiterandslide\_imgApp &&

                    <div className="row header\_slide">

                       <SlideHeader/>

                    </div>

                 }

                </div>

                {!this.props.stateFiterandslide\_imgApp &&

                <div className="row">

                    <div className="container form\_filter">

                        <Filter StateFiterTyhomeNews\_TF={this.props.StateFiterTyhomeNews\_TF}  // Hiện hoặc ẩn fitter

                        GetNewsFiltertoApp={this.props.GetNewsFiltertoApp}

                         GetTypeNewstoApp={this.props.GetTypeNewstoApp}                       // GetTypeNewstoApp

                         />

                    </div>

                </div>

                }

            </div>

        );

    }

}

export default Header;

#### menu.js

import React, { Component } from 'react';

import img\_icon\_menu from '../header/image\_header/icons\_menu.png'

import {NavLink,Link} from "react-router-dom";

class Menu extends Component {

constructor(props) {

super(props);

this.state={

scrolled:false,

}

}

componentDidMount(){

window.addEventListener('scroll',()=>{

const isTop = window.scrollY < 150;

if(isTop !==true) {

this.setState({

scrolled:true

})

}

else this.setState({scrolled:false})

});

}

render() {

return (

<div className={!this.props.stateFiterandslide\_imgApp ?

"col-md-12 col-sm-12 col-xs-12 header2-menu "

: "col-md-12 col-sm-12 col-xs-12 header2-menu\_detail\_news" }>

<nav className={!this.state.scrolled ? "navbar navbar-expand-lg header2-menu-nav" :"navbar navbar-expand-lg header2-menu-nav-fixed"} id="stickyHeader">

<Link className="navbar-brand" to="/">PhongTroVN</Link>

<button className="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span className="navbar-toggler-icon">

<img src={img\_icon\_menu} alt="icomn\_menu"></img>

</span>

</button>

<div className="collapse navbar-collapse" id="navbarNav">

<ul className="nav navbar-nav ml-auto">

<li className="nav-item active">

<NavLink className="nav-link header2-menu-nav-item" exact activeClassName="header2-menu-nav-item-active" to='/' onClick={this.props.StateFiterandslide\_FuncApp}>Trang chủ</NavLink>

</li>

<li className="nav-item">

<NavLink className="nav-link header2-menu-nav-item" to='/thue-phong-tro' activeClassName="header2-menu-nav-item-active" onClick={this.props.StateFiterandslide\_FuncApp}>Thuê Phòng trọ</NavLink>

</li>

<li className="nav-item">

<NavLink className="nav-link header2-menu-nav-item" activeClassName="header2-menu-nav-item-active" to='/thue-can-ho' onClick={this.props.StateFiterandslide\_FuncApp}>Thuê căn hộ</NavLink>

</li>

<li className="nav-item">

<NavLink className="nav-link header2-menu-nav-item" activeClassName="header2-menu-nav-item-active" to='/lien-he'>Liên hệ</NavLink>

</li>

</ul>

</div>

</nav>

</div>

);

}

}

export default Menu;

#### Slide.header.js

import React, { Component } from 'react';

import img\_slide1 from '../image\_header/hero1.jpg'

import img\_slide2 from '../image\_header/hero2.jpg'

import img\_slide3 from '../image\_header/hero3.jpg'

class SlideHeader extends Component {

    render() {

        return (

            <div id="demo" className="carousel slide" data-ride="carousel">

            {/\* Indicators \*/}

                                {/\* The slideshow \*/}

                            <div className="carousel-inner">

                            <div className="carousel-item active">

                                <img src={img\_slide1} alt="Los Angeles" className="carousel-item\_img"/>

                            </div>

                            <div className="carousel-item">

                                <img  src={img\_slide2} alt="Chicago" className="carousel-item\_img"/>

                            </div>

                            <div className="carousel-item">

                                <img  src={img\_slide3} alt="New York" className="carousel-item\_img"/>

                            </div>

                            </div>

                                {/\* Left and right controls \*/}

                            <a className="carousel-control-prev" href="#demo" data-slide="prev">

                                <div className="prev\_backgroups">

                                    <span className="carousel-control-prev-icon" />

                                </div>

                            </a>

                                <a className="carousel-control-next" href="#demo" data-slide="next">

                                <div className="next\_backgroups">

                                    <span className="carousel-control-next-icon"/>

                                </div>

                            </a>

            </div>

        );

    }

}

export default SlideHeader;

### Server

#### Home.router.js

var express = require("express")

var router = express.Router()

var controller= require('../controllers/HomePage.controller')

router.get("/tin-tong-hop",controller.News\_All);

router.get("/tin-phong-tro",controller.News\_RoomHome);

router.get("/tin-nha-tro",controller.News\_HouseHome);

router.get("/tin-can-ho",controller.News\_ApartmentHome);

router.get("/thong-tin-chi-tiet/:id",controller.News\_Detail);

router.get("/thong-tin-chi-tiet/city/:code\_city",controller.GetNameCity);

router.get("/thong-tin-chi-tiet/dictrict/:code\_dictrict",controller.GetNameDictrict);

router.get("/tin-tuc-gan-do/:code\_city/:code\_dictrict",controller.NewsNears);

router.post("/tim-kiem/gia-tien/dien-tich",controller.NewsFilter);

module.exports=router;

#### index.ts

import express from 'express';

import mongoose from 'mongoose';

import passport from 'passport';

import bodyParser from 'body-parser';

import session from 'express-session';

import userRouter from './routers/user.router';

import homeRouter from './routers/phongtro.router';

import homePageRouter from './routers/home.router';

import flash from 'connect-flash';

const app = express();

/\* Khai báo để sử dụng kịch bản passport \*/

import './config/passport.config';

app.use(session({

    name: 'login',

    secret : 'secured\_key',

    resave : false,

    saveUninitialized : false,

    cookie:{

      maxAge:1000\*60\*15

    }

}))

app.use([

  bodyParser.json(),

  bodyParser.urlencoded({

    extended: true,

  })

]);

app.use(flash());

app.use(passport.initialize());

app.use(passport.session());

app.use(express.static("public"));

app.use('/nguoi-dung',userRouter);

app.use('/phong-tro',homeRouter);

app.use('/trang-chu',homePageRouter);

app.listen(8000, async ()=>{

    await mongoose.connect('mongodb://localhost:51734', {

      dbName: "Database\_PhongTroVN"

    });

    console.log("Listen port 8000");

});

#### Phongtro.router.js

var express = require("express")

var router = express.Router()

var controller= require('../controllers/phongtro.controller')

//################\_\_Đăng tin mới\_Address\_\_\_####################

router.post("/dang-tin-moi/chon-tinhTP",controller.citys);

router.get("/dang-tin-moi/danh-sach-duong/:code\_city",controller.StreetstoCitys);

router.post("/dang-tin-moi/chon-QH",controller.districts);

router.post("/dang-tin-moi/chon-Duong",controller.streets);

//#########\_\_Đăng tin upload\_Image\_\_\_##############

router.post("/dang-tin-moi/upload\_image",controller.UploadAvarta);

router.get("/open\_image/nameimage=:imagename",controller.getImageAvarta);

router.post("/dang-tin-moi/xoa-anh-dai-dien",controller.DeleteImageAvarta);

router.post("/dang-tin-moi/up-load\_hinh-mo-ta",controller.UploadImageInfor);

router.post("/dang-tin-moi/xoa-anh-mo-ta",controller.DeleteImageInfor);

//#########\_\_Đăng tin mới\_Finish\_\_\_##############

router.post("/dang-tin-moi/phong-tro",controller.PostNewsPT);

router.post("/dang-tin-moi/nha-tro",controller.PostNewsNT);

router.post("/dang-tin-moi/can-ho",controller.PostNewsCH);

//################\_\_Phong\_Tro\_\_\_####################

router.get("/quan-ly-tin-dang/phong-tro",controller.PostManagerPT);

router.get("/quan-ly-tin-dang/nha-tro",controller.PostManagerNT);

router.get("/quan-ly-tin-dang/can-ho",controller.PostManagerCH);

router.post("/quan-ly-tin-dang/an-tin-tuc-phong-tro",controller.PostManagerHiddenNewsPT);

router.post("/quan-ly-tin-dang/an-tin-tuc-nha-tro",controller.PostManagerHiddenNewsNT);

router.post("/quan-ly-tin-dang/an-tin-tuc-can-ho",controller.PostManagerHiddenNewsCH);

module.exports=router;

#### user.rouer,js

import express from "express"

import passport from "passport"

import \* as controller from "../controllers/user.controller"

const router = express.Router()

//################\_\_Login\_\_\_####################

router.get("/dang-nhap", controller.login)

router.get("/dang-nhap-thanh-cong", controller.loginsuccess)

router.post(

"/dang-nhap",

passport.authenticate("local.login", {

successRedirect: "/nguoi-dung/dang-nhap-thanh-cong",

failureRedirect: "/nguoi-dung/dang-nhap",

failureFlash: true,

}),

)

//################\_\_Register\_\_\_####################

router.post(

"/dang-ky",

passport.authenticate("local.register", {

successRedirect: "/nguoi-dung/dang-ky-thanh-cong",

failureRedirect: "/nguoi-dung/dang-ky",

failureFlash: true,

}),

)

router.get("/dang-ky-thanh-cong", controller.registersuccess)

router.get("/dang-ky", controller.register)

router.get("/thong-tin", controller.getInforUser)

router.post("/xac-thuc/so-dien-thoai", controller.accuracyPhone\_NB)

router.post(

"/xac-thuc/so-dien-thoai/ma-OTP",

controller.accuracyPhone\_NB\_key\_OTP,

)

router.get("/dang-xuat", controller.logout)

router.post("/doi-mat-khau", controller.ChangePassword)

router.post("/danh-gia", controller.xFeedBack)

router.get("/danh-gia", controller.GetFeedBack)

router.get("/thong-tin/:id", controller.getInforUserById)

router.get("/chinh-sua-thong-tin", controller.getInforUserEdit)

router.post(

"/chinh-sua-thong-tin/anh-dai-dien",

controller.UploadUserEditAvatar,

)

router.get("/open\_image/nameimage=:nameimage", controller.OpenAvatarUser)

router.post("/chinh-sua-thong-tin-ca-nhan", controller.EditedInforUser)

export default router

#### HomePage.controller.ts

import { Request, Response } from "express"

import News from "../models/News/news.model"

import Citys from "../models/tb\_city.phongtro.model"

import Districts from "../models/tb\_district.phongtro.model"

export const News\_All = async (req: Request, res: Response) => {

try {

const result = await News.find({ "infor.state\_news": true })

.sort({ "infor.date\_now": -1 })

.limit(6)

.exec()

res.json({

All\_News: result,

})

} catch (err) {

res.json({

result: false,

message: err,

})

}

}

export const News\_Detail = async (req: Request, res: Response) => {

try {

const result = await News.find({ \_id: req.params.id })

res.json({

news: result,

})

} catch (err) {

res.json({

result: false,

message: err,

})

}

}

export const GetNameCity = async (req: Request, res: Response) => {

try {

const result = await Citys.findOne({ code: req.params.code\_city })

if (!result) throw new Error("not found")

res.json({

NameCity: result.name,

})

} catch (err) {

res.json({

result: false,

message: err,

})

}

}

export const GetNameDictrict = async (req: Request, res: Response) => {

try {

const result = await Districts.findOne({ code: req.params.code\_dictrict })

if (!result) throw new Error("not found")

res.json({

NameDistricts: result.typename,

})

} catch (err) {

res.json({

result: false,

message: err,

})

}

}

export const NewsNears = async (req: Request, res: Response) => {

try {

const result = await News.find({

"address.code\_city": req.params.code\_city,

"address.code\_dictrict": req.params.code\_dictrict,

})

res.json({

NewsNears: result,

})

} catch (err) {

res.json({

result: false,

NewsNears: "",

message: err,

})

}

}

export const News\_RoomHome = async (req: Request, res: Response) => {

try {

const result = await News.find({

"infor.state\_news": 1,

"infor.typehome": 1,

})

.sort({ "infor.date\_now": -1 })

.limit(6)

.exec()

res.json({

NewsRoom: result,

})

} catch (err) {

res.json({

result: false,

message: err,

})

}

}

export const News\_HouseHome = async (req: Request, res: Response) => {

try {

const result = await News.find({

"infor.state\_news": 1,

"infor.typehome": 2,

})

.sort({ "infor.date\_now": -1 })

.limit(6)

.exec()

res.json({

HouseHome: result,

})

} catch (err) {

res.json({

result: false,

message: err,

})

}

}

export const News\_ApartmentHome = async (req: Request, res: Response) => {

try {

const result = await News.find({

"infor.state\_news": 1,

"infor.typehome": 3,

})

.sort({ "infor.date\_now": -1 })

.limit(6)

.exec()

res.json({

ApartmentHome: result,

})

} catch (err) {

res.json({

result: false,

message: err,

})

}

}

export const NewsFilter = async (req: Request, res: Response) => {

try {

const { PriceMin, PriceMax, AcreageMin, AcreageMax } = req.body

const result = await News.find({

"infor.price": { $gte: PriceMin, $lte: PriceMax },

"infor.acreage": { $gte: AcreageMin, $lte: AcreageMax },

"infor.state\_news": 1,

}).exec()

res.json({

NewsFilter: result,

})

} catch (err) {

console.log(err)

res.json({

result: false,

message: err,

})

}

}

#### news.model.ts

import mongoose, { Document, Schema, Model, model } from "mongoose"

interface IUtilities {

isChecked\_wifi: boolean

isChecked\_mezzanine: boolean

isChecked\_camera: boolean

isChecked\_parking: boolean

isChecked\_fridge: boolean

isChecked\_WashingMachine: boolean

isChecked\_television: boolean

isChecked\_AirConditional: boolean

isChecked\_elevator: boolean

isChecked\_pool: boolean

isChecked\_park: boolean

isChecked\_mattress: boolean

}

interface IAddress {

code\_city: number

code\_dictrict: number

code\_street: number

Lat\_ggmap: number

Lng\_ggmap: number

address\_detail: string

}

interface IInfor {

iduser: string

title: string

content\_infor: string

number\_phone: number

price: number

acreage: number

typehome: number

nb\_bedroom: number

nb\_bath\_toilet: number

nb\_kitchenroom: number

datetime\_create: string

date\_now: Date

datetime\_finish: string

state\_news: boolean

}

interface INews extends Document {

infor: IInfor

utilities: IUtilities

img\_avatar: string

img\_infor: object

address: IAddress

}

const NewsSchema = new Schema(

{

infor: {

iduser: {

type: String,

required: true,

},

title: {

type: String,

required: true,

},

content\_infor: {

type: String,

required: true,

},

number\_phone: {

type: Number,

required: true,

},

price: {

type: Number,

required: true,

},

acreage: {

type: Number,

required: true,

},

typehome: {

type: Number,

required: true,

},

nb\_bedroom: {

type: Number,

default: 1,

},

nb\_bath\_toilet: {

type: Number,

default: 1,

},

nb\_kitchenroom: {

type: Number,

default: 1,

},

datetime\_create: {

type: String,

required: true,

},

date\_now: {

type: Date,

default: Date.now,

},

datetime\_finish: {

type: String,

required: true,

},

state\_news: {

type: Boolean,

default: true,

},

},

utilities: {

isChecked\_wifi: {

type: Boolean,

default: false,

},

isChecked\_mezzanine: {

type: Boolean,

default: false,

},

isChecked\_camera: {

type: Boolean,

default: false,

},

isChecked\_parking: {

type: Boolean,

default: false,

},

isChecked\_fridge: {

type: Boolean,

default: false,

},

isChecked\_WashingMachine: {

type: Boolean,

default: false,

},

isChecked\_television: {

type: Boolean,

default: false,

},

isChecked\_AirConditional: {

type: Boolean,

default: false,

},

isChecked\_elevator: {

type: Boolean,

default: false,

},

isChecked\_pool: {

type: Boolean,

default: false,

},

isChecked\_park: {

type: Boolean,

default: false,

},

isChecked\_mattress: {

type: Boolean,

default: false,

},

},

img\_avatar: {

type: String,

required: true,

},

img\_infor: {

type: Object,

required: true,

},

address: {

code\_city: {

type: Number,

required: true,

},

code\_dictrict: {

type: Number,

required: true,

},

code\_street: {

type: Number,

required: true,

},

Lat\_ggmap: {

type: Number,

required: true,

},

Lng\_ggmap: {

type: Number,

required: true,

},

address\_detail: {

type: String,

required: true,

},

},

},

{

versionKey: false,

},

)

export default model("News", NewsSchema)

#### tb\_city\_phongtro.ts

import mongoose, { Document, Schema, Model, model} from 'mongoose';

interface ICity extends Document {

code: number;

name: string;

type: string;

typename: string;

}

const CitySchema = new Schema(

{

code: {

type:Number,

required:true

},

name:{

type:String,

required:true

},

type:{

type:String,

required:true

},

typename:{

type:String,

required:true

}

},{

versionKey:false

}

);

export default model('citys', CitySchema);

#### tb\_street\_phongtro.ts

import mongoose, { Document, Schema, Model, model} from 'mongoose';

interface IStreet extends Document {

code: number;

name: string;

type: string;

typename: string;

parent\_code: number;

parent\_code\_city: number;

}

const StreetSchema = new Schema(

{

code: {

type:Number,

required:true

},

name:{

type:String,

required:true

},

type:{

type:String,

required:true

},

typename:{

type:String,

required:true

},

parent\_code:{

type:Number,

required:true

},

parent\_code\_city:{

type:Number,

required:true

}

},{

versionKey:false

}

);

export default model('streets', StreetSchema);

#### tb\_district\_phongtro.model.js

import mongoose, { Document, Schema, Model, model } from "mongoose"

interface IDistrict extends Document {

code: number

name: string

type: string

typename: string

parent\_code: number

}

const DistrictSchema = new Schema(

{

code: {

type: Number,

required: true,

},

name: {

type: String,

required: true,

},

type: {

type: String,

required: true,

},

typename: {

type: String,

required: true,

},

parent\_code: {

type: Number,

required: true,

},

},

{

versionKey: false,

},

)

export default model("district", DistrictSchema)

#### Comment.ts

import mongoose, { Document, Schema, Model, model } from "mongoose"

interface IComment extends Document {

idnews: string

idmb: string

contentcomment: string

timeComment: Date

}

const CommentSchema = new Schema(

{

idnews: {

type: String,

required: true,

},

idmb: {

type: String,

required: true,

},

contentcomment: {

type: String,

required: true,

},

timeComment: {

type: Date,

default: Date.now,

},

},

{

versionKey: false,

},

)

export default model("Comment", CommentSchema)

#### feekback.ts

import mongoose, { Document, Schema, Model, model} from 'mongoose';

interface IFeebBack extends Document {

iduser: string;

titelfeedback: string;

contentfeedback: string;

timefeedback: Date;

}

const FeebBackSchema = new Schema(

{

iduser:{

type:String,

required:true

},

titelfeedback:{

type:String,

required:true

},

contentfeedback:{

type:String,

required:true

},

timefeedback:{

type:Date,

default: Date.now

}

},{

versionKey:false

}

);

export default model('FeebBack', FeebBackSchema);

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Simplilearn, "Fundamentals of Software Testing: Concepts and Process," 21 7 2023. [Online]. Available: https://www.simplilearn.com/tutorials/devops-tutorial/fundamentals-of-software-testing. |
| [2] | P. H. Hoàng, "NHỮNG DÒNG CODE GIẾT NGƯỜI – ĐẠO ĐỨC VÀ CÁI TÂM CỦA DEVELOPER," 30 7 2016. [Online]. Available: https://toidicodedao.com/2016/12/15/dao-duc-developer/. |
| [3] | H. Krasner, "The cost of Poor Software Quality in the US: A 2022 Report," Spicewood, 2022. |
| [4] | Glenford J. Myers, Corey Sandler, Tom Badgett, The Art of Software Testing 3rd Edition, Wiley, 2023. |
| [5] | FATIH GURCAN, GONCA GOKCE MENEKSE DALVEREN, NERGIZ ERCIL CAGILTAY, "Evolution of Software Testing Strategies and Trends: Semantic Content Analysis of Software Research Corpus of the Last 40 Years," *NTNU Open,* vol. 10, 2022. |
| [6] | S. S. Isha, “Software Testing Techniques and Strategies,” tập 4, số 4, pp. 99-102, 4 2014. |