BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC THĂNG LONG**

Logo

Description automatically generated

**CHUYÊN ĐỀ TỐT NGHIỆP**

**XÂY DỰNG PHẦN MỀM TÌM KIẾM VÀ QUẢN LÝ PHÒNG TRỌ, CĂN HỘ, NHÀ CHO THUÊ**

**GIÁO VIÊN HƯỚNG DẪN: THS. ĐINH THỊ THÚY**

**SINH VIÊN THỰC HIỆN: A40534 - ĐẶNG THẾ THUYÊN**

**A40310 - ĐÀO THỊ THU THẢO**

**HÀ NỘI – 2024**

**MỤC LỤC**

[CHƯƠNG 1. Cơ sở lý thuyết 1](#_Toc171055880)

[1.1. Công nghệ sử dụng 1](#_Toc171055881)

[1.1.1. Javascript 1](#_Toc171055882)

[1.1.2. Reactjs 2](#_Toc171055883)

[1.1.3. Nodejs 6](#_Toc171055884)

[1.1.4. Redux 7](#_Toc171055885)

[1.1.5. Hệ quản trị cơ sở dữ liệu 10](#_Toc171055886)

[1.1.6. Scraping data (Cào dữ liệu) 11](#_Toc171055887)

[CHƯƠNG 2. Phân tích và thiết kế chương trình 14](#_Toc171055888)

[2.1. Kiến trúc hệ thống 14](#_Toc171055889)

[2.1.1. Sơ đồ kiến trúc 14](#_Toc171055890)

[2.1.2. Mô tả kiến trúc 14](#_Toc171055891)

[2.2. Yêu cầu chức năng 15](#_Toc171055892)

[2.3. Usecase 15](#_Toc171055893)

[2.3.1. Các tác nhân hệ thống 15](#_Toc171055894)

[2.3.2. Mô tả các Usecase 15](#_Toc171055895)

[2.3.3. Biểu đồ Usecase 17](#_Toc171055896)

[2.4. Đặc tả yêu cầu 17](#_Toc171055897)

[CHƯƠNG 3. Thiết kế hệ thống 41](#_Toc171055898)

[3.1. Thiết kế cơ sở dữ liệu 41](#_Toc171055899)

[3.1.1. Lược đồ cơ sở dữ liệu 41](#_Toc171055900)

[3.1.2. Danh sách các bảng 41](#_Toc171055901)

[3.1.3. Chi tiết các bảng 42](#_Toc171055902)

[3.2. Thiết kế API 47](#_Toc171055903)

[3.2.1. Thiết kế chi tiết API 48](#_Toc171055904)

[3.3. Thiết kế một số chức năng chính 51](#_Toc171055905)

[3.3.1. Chức năng Đăng nhập 51](#_Toc171055906)

[3.3.2. Chức năng Đăng ký 52](#_Toc171055907)

[3.3.3. Chức năng Quản lý bài viết 52](#_Toc171055908)

[3.3.4. Chức năng Quản lý liên hệ 53](#_Toc171055909)

[CHƯƠNG 4. Cài đặt hệ thống và Thử nghiệm 55](#_Toc171055910)

[4.1. Môi trường cài đặt 55](#_Toc171055911)

[4.2. Cấu trúc thư mục mã nguồn 56](#_Toc171055912)

[4.2.1. Server 56](#_Toc171055913)

[4.2.2. Client 57](#_Toc171055914)

[4.3. Thử nghiệm 58](#_Toc171055915)

[4.4. Kiểm thử 60](#_Toc171055916)

[4.4.1. UC giao diện chung của màn người dùng 61](#_Toc171055917)

[4.4.2. UC giao diện chung của màn Admin 62](#_Toc171055918)

[4.4.3. Kiểm thử UC dashboard 62](#_Toc171055919)

[CHƯƠNG 5. Kết luận 64](#_Toc171055920)

[5.1. Kết quả đạt được 64](#_Toc171055921)

[5.2. Hạn chế 64](#_Toc171055922)

[5.3. Hướng phát triển 64](#_Toc171055923)

**DANH MỤC MINH HỌA**

[Hình 11. Sơ đồ lớp phân tích Đăng nhập 51](#_Toc171040361)

[Hình 12. Sơ đồ trình tự Đăng nhập 51](#_Toc171040362)

[Hình 13. Sơ đồ lớp phân tích Đăng ký 52](#_Toc171040363)

[Hình 14. Sơ đồ lớp phân tích Đăng nhập 52](#_Toc171040364)

[Hình 15. Sơ đồ lớp phân tích Quản lý bài viết 52](#_Toc171040365)

[Hình 16. Sơ đồ trình tự quản lý bài viết 53](#_Toc171040366)

[Hình 17. Sơ đồ lớp phân tích Quản lý liên hệ 53](#_Toc171040367)

[Hình 18. Sơ đồ trình tự quản lý liên hệ 54](#_Toc171040368)

[Hình 19. Cấu hình thư mục server 56](#_Toc171040369)

[Hình 10. Cấu hình thư mục client 57](#_Toc171040370)

[Bảng 3.1. Bảng mô tả tên bảng dữ liệu 42](#_Toc171040393)

[Bảng 3.2. Bảng Users 42](#_Toc171040394)

[Bảng 3.3. Bảng Areas 43](#_Toc171040395)

[Bảng 3.4. Attributes 43](#_Toc171040396)

[Bảng 3.5. Prices 43](#_Toc171040397)

[Bảng 3.6. Bảng Posts 44](#_Toc171040398)

[Bảng 3.7. Bảng Provinces 44](#_Toc171040399)

[Bảng 3.8. Bảng Categories 45](#_Toc171040400)

[Bảng 3.9. Overviews 45](#_Toc171040401)

[Bảng 3.10. Images 45](#_Toc171040402)

[Bảng 3.11. Labels 46](#_Toc171040403)

Bảng 3.12. Contacts…………………………………………………………………………...46

Bảng 3.13. Admins……………………………………………………………………………46

**LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin được gửi lời cảm ơn chân thành nhất tới toàn thể các thầy cô giáo trong khoa Công Nghệ Thông Tin cũng như các thầy cô giảng dạy trong trường Đại học Thăng Long đã truyền đạt những kiến thức quý báu và bổ ích nhất cho chúng em trong những năm học vừa qua.

Đặc biệt, chúng em xin gửi lời cảm ơn chân thành và sâu sắc tới cô – ThS.Đinh Thị Thúy, giảng viên khoa Công Nghệ Thông Tin, trường Đại học Thăng Long. Thời gian qua nhờ vào sự hướng dẫn tận tình và sự động viên, giúp đỡ ân cần từ cô mà chúng em mới có thành quả là xây dựng và hoàn thiện được Chuyên đề tốt nghiệp như ngày hôm nay.

Bên cạnh đó, chúng em cũng xin chân thành gửi lời cảm ơn đến tất cả những người bạn cùng khoá, những người anh, người chị khoá trên đã nhiệt tình giúp đỡ và chia sẻ cho chúng em những kinh nghiệm và kiến thức đáng quý trong suốt quá trình chúng em xây dựng dự án của mình.

Vì kiến thức còn nhiều hạn chế, trong quá trình triển khai và thực hiện Chuyên đề tốt nghiệp này chúng em không tránh khỏi có những sai sót không mong muốn, chính vì vậy chúng em rất mong nhận được những ý kiến đóng góp từ các thầy cô cũng như các bạn trong khoa để hoàn thiện dự án này cũng như bổ sung những kiến thức còn thiếu sót.

Cuối cùng chúng em xin kính gửi lời chúc tới toàn thể các thầy cô trong khoa Công Nghệ Thông Tin cũng như các thầy cô đang giảng dạy trong trường Đại học Thăng Long luôn luôn mạnh khỏe và tràn đầy nhiệt huyết để truyền đạt kiến thức cho các thế hệ mai sau.

Hà Nội, tháng 07 năm 2024

Sinh viên thực hiện

Đặng Thế Thuyên

Đào Thị Thu Thảo

**LỜI CAM ĐOAN**

Tôi xin cam đoan đề tài “Xây dựng phần mềm tìm kiếm và quản lý phòng trọ, căn hộ, nhà cho thuê” trong chuyên đề tốt nghiệp của nhóm tôi được tiến hành một cách công khai và minh bạch, dựa trên sự cố gắng và nỗ lực của thành viên trong nhóm cũng như sự giúp đỡ, hướng dẫn tận tình của giảng viên hướng dẫn cô Định Thị Thúy.

Các số liệu nghiên cứu nêu trong đồ án đảm bảo tính trung thực, các nội dung trích dẫn đều ghi đầy đủ thông tin nguồn, tuân thủ quy định của Luật Bản quyền. Nếu phát hiện có sự sao chép, nhóm tôi xin chịu hoàn toàn trách nhiệm và kỷ luật từ phía nhà trường.

SINH VIÊN THỰC HIỆN

ĐẶNG THẾ THUYÊN

ĐÀO THỊ THU THẢO

**LỜI MỞ ĐẦU**

Ngày nay khi xã hội ngày càng phát triển, đời sống người dân ngày càng nâng cao thì nhu cầu tìm chỗ ở an toàn, sạch đẹp cũng rất quan trọng. Nhu cầu về nhà ở luôn là một yếu tố cơ bản trong đời sống hàng ngày của mọi người. Với sự gia tăng dân số đô thị, nhiều người lao động, sinh viên và gia đình trẻ luôn có nhu cầu tìm kiếm nhà ở hoặc phòng trọ phù hợp với điều kiện và hoàn cảnh của họ. Thị trường phòng trọ rất đa dạng, từ các căn hộ chung cư mini, phòng trọ cho sinh viên, đến những căn hộ dịch vụ cao cấp. Tuy nhiên, quá trình tìm kiếm phòng trọ hay nhà cho thuê hiện nay còn gặp nhiều khó khăn và bất tiện, do thiếu thông tin chính xác, minh bạch và cập nhật.

Với sự phát triển mạnh mẽ của công nghệ thông tin và Internet, việc ứng dụng các giải pháp công nghệ vào quản lý và tìm kiếm nhà cho thuê trở nên ngày càng khả thi và hiệu quả. Một nền tảng trực tuyến có thể giúp kết nối người thuê và chủ nhà một cách nhanh chóng và tiện lợi, giảm thiểu thời gian và công sức của cả hai bên. Việc cung cấp một nền tảng trực tuyến để tìm kiếm và thuê nhà ở phản ánh sự đa dạng và phong phú của dịch vụ này.

Trong đó không thể không kể đến nhu cầu tìm phòng trọ của các bạn sinh viên, người đi làm, hộ gia đình tại các thành phố lớn. Với nhu cầu tìm phòng trọ nhanh chóng, tiết kiệm thời gian và lựa chọn dễ dàng đối với những người đi thuê trọ. Một website tìm kiếm, quản lý phòng trọ, căn hộ và nhà cho thuê cung cấp sự tiện lợi và linh hoạt cho cả người tìm phòng và chủ nhà. Người thuê có thể dễ dàng tìm kiếm và đặt phòng từ bất kỳ đâu, mà không cần phải đi qua quá trình tìm kiếm truyền thống mất thời gian. Bài toán phần mềm " Xây dựng phần mềm tìm kiếm và quản lý phòng trọ, căn hộ, nhà cho thuê " được thiết kế để giúp quản lý và tối ưu hóa quá trình cho thuê phòng trọ, căn hộ, hoặc nhà cho thuê. Đây là một giải pháp hiện đại và thuận tiện, hỗ trợ cả chủ nhà và người thuê phòng trong việc quản lý thông tin, giao dịch và tương tác trực tuyến.

Chính vì vậy mà chúng em đã có ý tưởng xây dựng một hệ thống tìm kiếm, quản lý phòng trọ nhằm giải quyết bài toán các vấn đề bất cập trong quá trình tìm kiếm và quản lý nhà cho thuê, đề tài này được chúng em hướng tới mục tiêu đóng góp tích cực cho xã hội, nâng cao chất lượng cuộc sống và thúc đẩy phát triển bền vững. Việc tạo ra một nền tảng công nghệ hiệu quả không chỉ hỗ trợ người dân trong việc tìm kiếm nhà ở mà còn thúc đẩy sự minh bạch, công bằng và hiệu quả trong thị trường bất động sản.

Đối với người thuê: Phần mềm không chỉ giúp bạn tìm kiếm nơi ở mà còn tạo ra cơ hội tương tác trực tuyến với chủ nhà. Có thể dễ dàng tìm kiếm và so sánh các lựa chọn phòng trọ, căn hộ, nhà cho thuê dựa trên các tiêu chí như vị trí, giá cả, diện tích. Điều này giúp người thuê đưa ra quyết định thông minh và hợp lý hơn.

Đối với chủ trọ: Phần mềm có thể giúp chủ trọ quản lý danh sách phòng trọ, căn hộ, nhà cho thuê của mình một cách hiệu quả, tiếp cận được nhiều khách hàng tiềm năng hơn và giảm thiểu thời gian trống phòng. Điều này giúp chủ trọ có thể quản lý được số lượng phòng còn hay đã thuê một cách hiệu quả.

Với những lý do trên, đề tài " Xây dựng phần mềm tìm kiếm và quản lý phòng trọ, căn hộ, nhà cho thuê " được lựa chọn với kỳ vọng sẽ mang lại những giá trị thiết thực và góp phần cải thiện chất lượng dịch vụ trong lĩnh vực bất động sản.

Trong tài liệu này nội dung chuyên đề gồm có 5 chương sau:

**Chương 1. Cơ sở lý thuyết**

Giới thiệu cơ sở lý thuyết được sử dụng trong dự án.

**Chương 2. Phân tích và thiết kế chương trình**

Đưa ra kiến trúc tổng thể của hệ thống kèm theo mô tả từng nền tảng và tất cả các công nghệ đã được nghiên cứu và áp dụng cho mỗi nền tảng.

**Chương 3. Thiết kế hệ thống**

Đưa ra các sơ đồ kiến trúc CSDL, cấu trúc các bảng dữ liệu, API sử dụng.

**Chương 4. Cài đặt hệ thống và thử nghiệm**

Chỉ ra môi trường cài đặt hệ thống, hệ thống thư mục mã nguồn, những công cụ được sử dụng trong quá trình xây dựng hệ thống và kết quả

**Chương 5. Kết luận**

Đưa ra kết luận sau khi hoàn thành dự án, những hạn chế gặp phải và hướng phát triển cho tương lai.

# Cơ sở lý thuyết

Cũng chính bởi mục đích phục vụ cho việc tìm phòng trọ mà hệ thống đòi hỏi phải có sự tiện lợi và dễ dàng sử dụng vì vậy chúng em đã quyết định phát triển hệ thống trên nền tảng website và đây cũng chính là nền tảng lập trình mà em đã quyết định theo đuổi trong những năm sắp tới.

Hệ thống sử dụng webservies và cơ sở dữ liệu SQL Sever để làm nơi lưu trữ dữ liệu và giao tiếp: truyền, nhận thông tin tới ứng dụng. Hệ thống sử dụng các công cụ để thực hiện đề tài này như: Visual Studio 2013 với công nghệ Reactjs, Nodejs, MySQL.

## Công nghệ sử dụng

### Javascript

JavaScript là một ngôn ngữ lập trình đa nền tảng, ngôn ngữ lập trình kịch bản, hướng đối tượng. JavaScript là một ngôn ngữ nhỏ và nhẹ. Khi nằm bên trong một môi trường. JavaScript có thể kết nối tới các object của môi trường đó và cung cấp các cách quản lý chúng.

JavaScript chứa các thư viện tiêu chuẩn cho các object, ví dụ như: Array, Date, và Math, và các yếu tố cốt lõi của ngôn ngữ lập trình như: toán tử, cấu trúc điều khiển, và câu lệnh. JavaScript có thể được mở rộng cho nhiều mục đích bằng việc bổ sung thêm các object; ví dụ:

* *Client-side JavaScript* - JavaScript phía máy khách, JavaScript được mở rộng bằng cách cung cấp các object để quản lý trình duyệt và Document Object Model (DOM) của nó. Ví dụ, phần mở rộng phía máy khách cho phép một ứng dụng tác động tới các yếu tố trên một trang HTML và phản hồi giống các tác động của người dùng như click chuột, nhập form, và chuyển trang.
* *Server-side JavaScript* - JavaScript phía Server, JavaScript được mở rộng bằng cách cung cấp thêm các đối tượng cần thiết để để chạy JavaScript trên máy chủ. Ví dụ, phần mở rộng phía server này cho phép ứng dụng kết nối với cơ sở dữ liệu (database), cung cấp thông tin một cách liên tục từ một yêu cầu tới phần khác của ứng dụng, hoặc thực hiện thao tác với các tập tin trên máy chủ.

Đối với dự án " Xây dựng phần mềm tìm kiếm và quản lý phòng trọ, căn hộ, nhà cho thuê " này, Javascript được sử dụng để xây dựng Client-side.

Trong dự án “Xây dựng phần mềm tìm kiếm và quản lý phòng trọ, căn hộ, nhà cho thuê”, Javascript được sử dụng kết hợp tiêu chuẩn ES6.Hiện nay các JS Framework như NodeJS, Angular2, React Native ... đều sử dụng ES6. Vì vậy để sử dụng được các JS Framework trên thì việc sử dụng thành thạo ES6 là điều cần thiết

### Reactjs

1. Giới thiệu

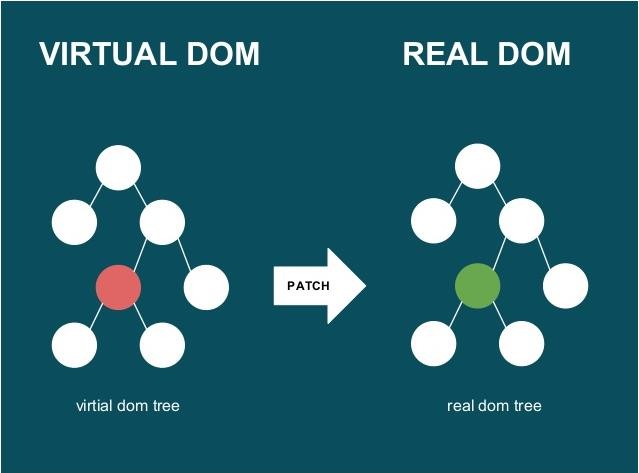
React.js là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Nếu như AngularJS là một Framework cho phép nhúng code javasscript trong code html thông qua các attribute như ng-model, ng- repeat...thì với react là một library cho phép nhúng code html trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS.Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn. 3 đặc điểm nổi bật của React:

* Just the UI: Chỉ sử dụng React để xây dựng giao diện người dùng, hiểu đơn giản thì trong mô hình MVC, React được coi như là phần V. Vì vậy nếu bạn muốn xây dựng một trang web hoàn chỉnh thì không thể dùng mỗi React được, mà nó cần thêm phần M và C nữa.
* Virtual DOM: công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng.
* Data Flow: React sử dụng luồng dữ liệu 1 chiều giúp kiểm soát dữ liệu dễ dàng hơn.

1. Cấu trúc của ReactJs

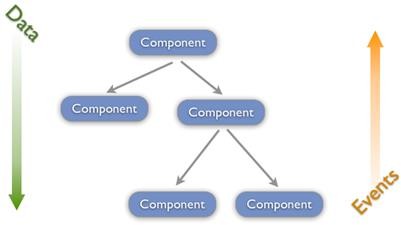
**Virtual DOM**

Công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. ReactJS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này. Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu cho việc re-render DOM tree thật.



*Hình 1. Virtual DOM trong ReactJs*

React sử dụng cơ chế one-way data binding – luồng dữ liệu 1 chiều. Dữ liệu được truyền từ parent đến child thông qua props. Luồng dữ liệu đơn giản giúp chúng ta dễ dàng kiểm soát cũng như sửa lỗi.



*Hình 2. Luồng dữ liệu trong ReactJs*

Với đặc điểm ở trên, React dùng để xây dựng các ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện. Ví dụ như Facebook: trên Newsfeed cùng lúc sẽ có các status khác nhau và mỗi status lại có số like, share, comment liên tục thay đổi. Khi đó React sẽ rất hữu ích để sử dụng.

* JSX: JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript.
* Đặc điểm: Faster: Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javacsript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript.
* Safer: an toàn hơn. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt.
* Easier: Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng.

Component: React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, trang web được xây dựng bằng cách sử dụng những thành phần (component) nhỏ. Một component có thể được tái sử dụng ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác. Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component. Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo.

Vòng đời của 1 component

* Lần lượt các hành động sau để khởi tạo Component
* Khởi tạo Class (đã thừa kế từ class Component của React)
* Khởi tạo giá trị mặc định cho Props (defaultProps)
* Khởi tạo giá trị mặc định cho State (trong hàm constuctor)
* Gọi hàm componentWillMount(): Thực hiện một số tác vụ, hàm này chỉ thực hiện 1 lần duy nhất.
* Gọi hàm render(): thực hiện việc render
* Gọi hàm componentDidMount(): Thực hiện một số tác vụ, hàm này chỉ thực hiện 1 lần duy nhất, hàm này được gọi để thông báo component đã tồn tại trên DOM, từ đó các thao tác trên DOM sẽ có thể thực hiện bình thường đối với component này
* Khi State thay đổi
* Cập nhật giá trị cho state
* Gọi hàm shouldComponentUpdate(): Hàm này thực hiện khi state và props thay đổi. Hàm này sẽ trả về kết quả true/false, sử dụng đến hàm này để xử lý xem có cần update component không
* Gọi hàm componentWillUpdate(): Hàm này thực hiện dựa vào kết quả của hàm trên (shouldComponentUpdate. Nếu hàm trên trả về false, thì React sẽ không gọi hàm này
* Gọi hàm render()
* Gọi hàm componentDidUpdate(): Hàm này thực hiện sau khi component được render lại, từ kết quả của componentWillUpdate
* Khi Props thay đổi
* Cập nhật giá trị cho props
* Gọi hàm componentWillReceiveProps(): Hàm này thực hiện liên tục mỗi khi props thay đổi
* Gọi hàm shouldComponentUpdate()
* Gọi hàm componentWillUpdate() – với điều kiện hàm trên return true
* Gọi hàm render()
* Gọi hàm componetDidUpdate()
* Khu unmount component
* Gọi hàm componentWillUnmount(): Hàm này thực hiện một lần duy nhất, khi component unmount. Hàm này hữu dụng khi cần xoá các timer không còn sử dụng
* Props và State
* Props: giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến.
* State: thể hiện trạng thái của ứng dụng, khi state thay đồi thì component đồng thời render lại để cập nhật UI.

### Nodejs

1. Giới thiệu

Node.js không phải là một ngôn ngữ lập trình, mà là một nền tảng phát triển ứng dụng dựa trên JavaScript. JavaScript là ngôn ngữ lập trình phổ biến được sử dụng phổ biến trong việc phát triển ứng dụng web. Node.js cho phép chạy mã JavaScript trên máy chủ, không chỉ trên trình duyệt như truyền thống. Node.js cung cấp các thư viện và framework như Express.js, Sails.js, Meteor.js để phát triển các ứng dụng web đơn trang.

NodeJS tương đối gọn nhẹ, tính hiệu quả cao và là một công cụ hoàn hảo dành cho mọi ứng dụng chuyên sâu về dữ liệu dựa theo khoảng thời gian thực khi chạy trên những thiết bị phân tán. NodeJs thường xuyên được dùng cho mục đích xây dựng một số ứng dụng như:  AdServer, Websocket server, Fast File Upload Client, RESTful API, Cloud Services, Any Real-time Data Application.

1. Nguyên lý hoạt động

Node.js là một nền tảng phát triển phía server được xây dựng trên nền tảng JavaScript. Khi một yêu cầu mạng đến từ một client, Node.js sẽ xử lý yêu cầu đó bằng cách thực hiện các bước như sau:

* Node.js tạo một event loop để theo dõi các yêu cầu mạng đến và đi.
* Khi một yêu cầu mạng đến, Node.js sẽ tạo một worker thread (luồng làm việc) để xử lý yêu cầu đó.
* Trong worker thread, Node.js sẽ thực hiện các tác vụ xử lý yêu cầu, chẳng hạn như đọc và ghi vào cơ sở dữ liệu, đọc và ghi file, tương tác với API, ...
* Khi worker thread đã hoàn thành các tác vụ, Node.js sẽ trả về kết quả cho client qua mạng.
* Nếu có yêu cầu mạng mới đến, Node.js sẽ tạo một worker thread mới để xử lý yêu cầu đó.

Các yêu cầu mạng đến và đi trong Node.js được xử lý bằng cách sử dụng các hàm callback, Promise, async/await để đảm bảo tính phi đồng bộ và tăng hiệu suất của ứng dụng. Các yêu cầu mạng được xử lý một cách độc lập, giúp tránh tình trạng "blocking" (chặn) trong quá trình xử lý yêu cầu. Node.js cũng có thể hoạt động với các module và thư viện khác để hỗ trợ cho việc phát triển ứng dụng web.

### Redux

1. Giới thiệu

Redux là một thư viện Javascript giúp tạo ra thành một lớp quản lý trạng thái của ứng dụng. Được dựa trên nền tảng tư tưởng của kiến trúc Flux do Facebook giới thiệu, do vậy Redux thường là bộ đôi kết hợp hoàn hảo với React (React Js và React Native).

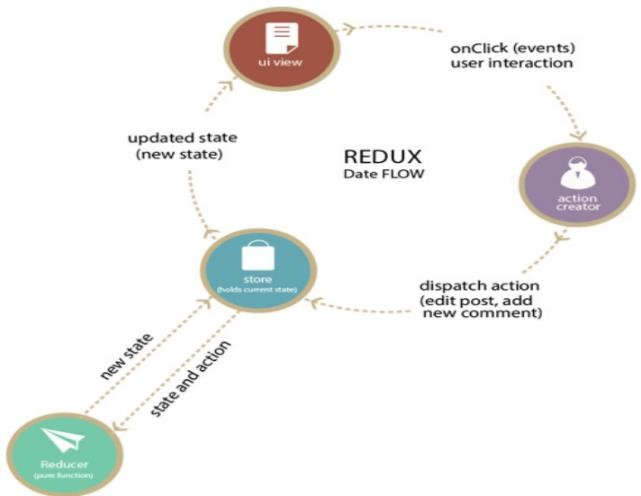
1. Nguyên lý hoạt động

Redux hoạt động dựa trên 3 nguyên lý:

* Nguyên lý đầu tiên của Redux là tất cả mọi thứ mà thay đổi trong ứng dụng, bao gồm dữ liệu và trạng thái giao diên – UI state, được lưu trong đối tượng gọi là state hoặc state tree. Trong quá trình hoạt động của ứng dụng, bị phụ thuộc khá nhiều yếu tố: dữ liệu từ máy chủ ban đầu, thao tác của người dùng (nhập dữ liệu, click menu, button...), dữ liệu cập nhật từ máy chủ, dữ liệu được tính toán trong ứng dụng (Ví dụ: tính toán số dư tài khoản dựa trên biến động của tỉ giá)... những yếu tố này còn gọi là nguồn dữ liệu. Những nguồn dữ liệu này đến từ những nơi khác nhau, bất kỳ khi nào, khiến cho ứng dụng của chúng ta rất khó kiểm soát, chúng tác động đến những thành phần đơn lẻ, hoặc nhiều thành phần trên ứng dụng, hoặc hiệu ứng dây chuyền. Chính vì sự phức tạp đó, là vấn đề mà Redux muốn giải quyết, tất cả các nguồn dữ liệu cần phải được quản lý và tạo thành một nguồn duy nhất, tin cậy.
* Nguyên lý thứ 2: State chỉ được phép đọc, Cách duy nhất để thay đổi State của ứng dụng là phát một Action (là 1 object mô tả những gì xảy ra). Trạng thái của ứng dụng không được phép thay đổi “trực tiếp”, trạng thái cũng chỉ là một đối tượng mà thôi, nên việc thay đổi là được. Tuy nhiên, với Redux hay Flux thì trạng thái chỉ thay đổi khi và chỉ khi có một sự kiện xảy ra, giống như ra trận thì chỉ được phép nghe lời từ chỉ huy, mọi tin tức tình báo đều được gởi tới chỉ huy, nếu không có lệnh từ chỉ huy thì tất cả không được phép manh động.
* Nguyên lý thứ 3: sử dụng pure function nhận tham số là state trước đó và action và trả về state tiếp theo. Hàm này gọi là reducer, (pure function là hàm trả về giá trị phụ thuộc duy nhất vào giá trị của tham số, pure function có tập các tham số thì giá trị trả về cũng là một tập tương tự như thế). Việc thay đổi trạng thái của ứng dụng, được thực hiện thông qua các hàm thuần tuý. Đưa vào giá trị sự kiện, trạng thái hiện tại và hàm trả về trạng thái tiếp theo. Dù ứng dụng có thể rất lớn, nhưng các hàm reducer này thì chỉ cần nhỏ gọn thay đổi trên từng lá của cây trạng thái, và chúng hoàn toàn có thể kết hợp với nhau tạo thành chuỗi sự kiện. Ví dụ: người click vào menu (một sự kiện

=> thay đổi trạng thái), sau đó router cũng cần thay đổi để phù hợp với ngữ cảnh.

1. Cấu trúc

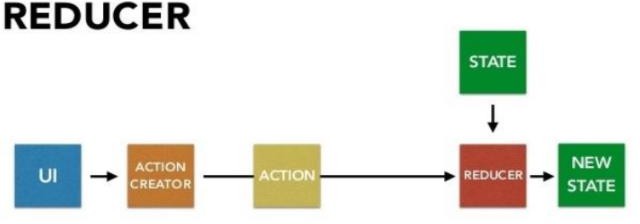


*Hình 3. Cấu trúc Redux*

Tất cả State được lưu trong store, được tạo ra bởi Redux.createStore, nó kết nối cả 3 nguyên lý của Redux. Nó lưu trạng thái hiện tại của ứng dụng, cho phép gửi đi một action. Khi tạo ra nó, phải xác định reducer để biết được rằng state được thay đổi như thế nào cùng với các action.

* getState() method: trả về state hiện tại của store .
* dispatch() method: cho phép gửi đi một action để thay đổi state của ứng dụng, có thể cập nhật UI của ứng dụng tương ứng với state hiện tại.
* subscribe() method: store the dõi state của ứng dụng.

1. Luồng dữ liệu trong Redux



*Hình 4. Luồng dữ liệu trong Redux*

Luồng dữ liệu trong Redux

* UI gọi tới store.dispatch (action)
* Reducer store gọi reducer nhận state hiện tại và action vừa được gửi đi
* Root reducer có thể kết hợp nhiều reducer trong state tree để đưa ra kết quả - state tiếp theo của ứng dụng. Sử dụng combineReducers().
* Store lưu toàn bộ state được trả về bởi root reducer.

1. Kết hợp Redux với React

Redux có thể được dùng với các thư viện Angular, Ember, jQuery... trong đó rất thích hợp với React, vì React và Redux cho phép quản lý UI giống như hàm của state. Redux cập nhật state tương ứng với action.

* Async Actions: action không đồng bộ, khi gọi tới API không đồng bộ quan tâm đến 2 thời điểm: thời điểm bắt đầu gọi API, thời điểm nhận trả lời (timeout) dẫn đến có 3 loại action:
* Action thông báo tới reducer rằng request bắt đầu: reducer xử lý action này bằng cách chuyển cờ isFetching của state. Do đó UI biết được thời gian để thay đổi
* Action thông báo reducer rằng request kết thúc thành công: reducer kết hợp dữ liệu mới với state hiện tại mà nó đang quản lý, chuyển cờ isFetching. UI hiện thị dữ liệu sau khi lấy được
* Action thông báo với reducer rằng request thất bại: chuyển cờ isFetching, một số reducer có thông báo chứa lỗi, UI có thể hiện thị các lỗi đó.
* Async Flow: không đồng bộ được thực hiện nhờ middleware: redux-thunk, redux promise cho phép gửi đi nhiều hơn một là action, có thể là hàm hoặc promise. Middleware được đặt giữa gửi đi action và reducer, sử dụng để ghi log, báo lỗi, nhận API không đồng bộ, routing...
* Sử dụng cùng react-router: routing trong ứng dụng Redux. Redux là tài nguyên cho dữ liệu, react-router tài nguyên cho URL Redux có 3 ứng dụng quan trọng nhất:
* Quản lý trạng thái: như một bản replay có thể undo/redo trạng thái của ứng dụng.
* Tăng tốc phát triển: với webpack đã có Hot Module Replacement, khi kết hợp với Redux, tạo thành sự kết hợp ăn ý, có thể viết code và debug rất dễ dàng.

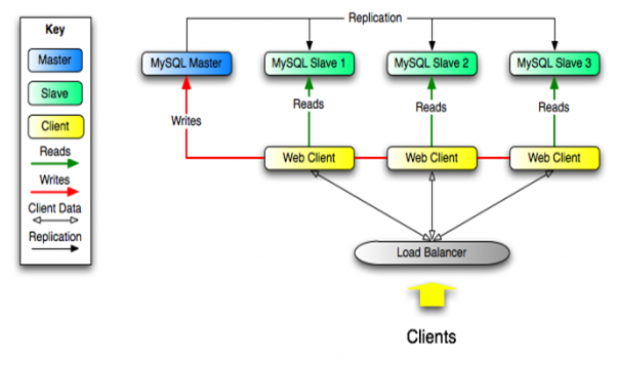
Ứng dụng offline: tất cả các thao tác của người dùng được lưu vào một cây trạng thái, khi có kết nối Internet, cây này sẽ được đồng bộ lên server bởi một loạt các sự kiện.

### Hệ quản trị cơ sở dữ liệu

1. Giới thiệu

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS) hoạt động theo mô hình client- server. RDBMS là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối liên hệ giữa chúng.

1. Hoạt động



Hình 5. Sơ đồ hoạt động

Máy chủ MySQL có sẵn một chương trình riêng biệt được sử dụng trong một môi trường riêng biệt mạng client-server, có thể nhúng vào những ứng dụng riêng biệt MySQL. Cách vận hành chính trong môi trường MySQL như sau:

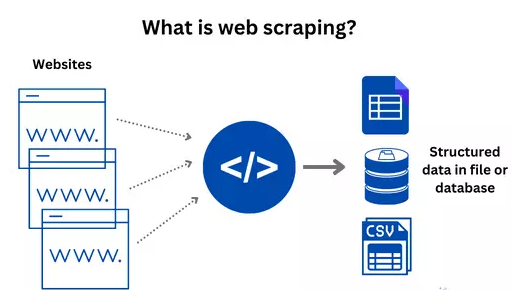
* MySQL tạo ra bảng để lưu trữ dữ liệu, định nghĩa sự liên quan giữa các bảng đó.
* Client sẽ gửi yêu cầu SQL bằng một lệnh đặc biệt trên MySQL.
* Ứng dụng trên server sẽ phản hồi thông tin và trà về kết quả trên máy client.

### Scraping data (Cào dữ liệu)

1. Giới thiệu

Scraping data (còn được gọi là web scraping) là quá trình lấy dữ liệu từ một trang web cụ thể bằng cách trích xuất thông tin từ các phần tử HTML của trang đó. Scraping thường được sử dụng khi bạn chỉ quan tâm đến dữ liệu từ một số trang web cụ thể và muốn lấy thông tin từ các thành phần nhất định trên trang web đó.

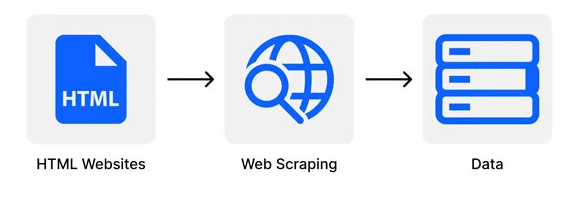
Web scraper hoạt động bằng cách quét mã HTML của trang web và lấy ra phần thông tin hữu ích ẩn trong đó. Tuy nhiên không nên đánh đồng web scraping với một hình thức khai thác dữ liệu khác là web crawling, khi web crawling thường trừu tượng hơn và lấy tất cả thông tin từ website, trong khi web scraping thì nhắm đến những tập dữ liệu cụ thể.



Hình 6. Scraping data

1. Hoạt động

* Đầu tiên, người ta cần cung cấp URL website mục tiêu cho các web scraper. Sau đó, scraper load toàn bộ code HTML hoặc trong vài trường hợp cả JS và CSS.
* Người dùng sau đó lựa chọn những dữ liệu cụ thể mà họ muốn như giá cả, kích thước của hàng hóa hay tiêu đề và nội dung các bài báo để scraper duyệt qua tất cả các trang liên quan và lấy dữ liệu. Các thông tin này có thể được cấu hình trước nếu nguồn dữ liệu là tĩnh, nhưng thường phải chọn thủ công cho mỗi website vì cấu trúc của chúng là không giống nhau.
* Cuối cùng, web scraper sẽ xuất ra tất cả dữ liệu đã thu thập được thành định dạng hữu ích hơn cho người dùng. Hầu hết các web scraper sẽ xuất dữ liệu sang bảng tính CSV hoặc Excel, trong khi các scraper nâng cao hơn sẽ hỗ trợ các định dạng khác như JSON, có thể được sử dụng cho API.

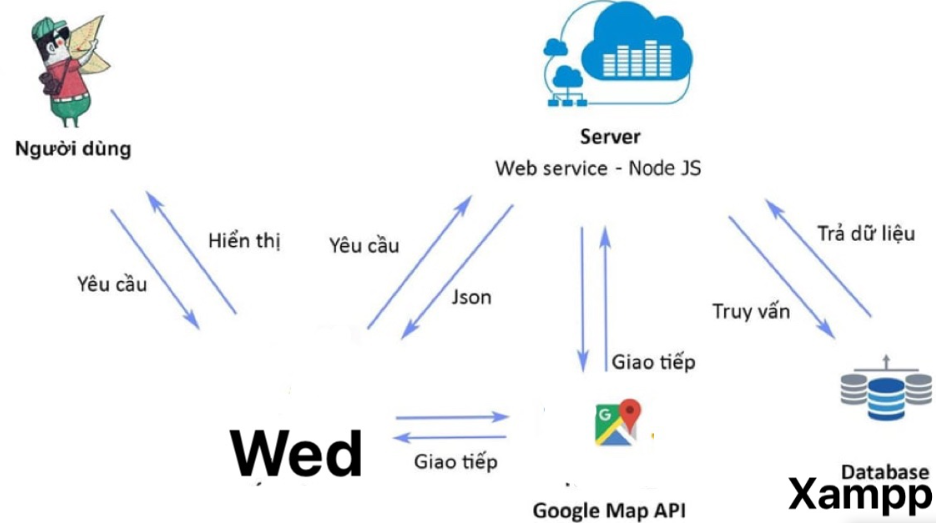


Hình 7. Sơ đồ hoạt động

# Phân tích và thiết kế chương trình

## Kiến trúc hệ thống

### Sơ đồ kiến trúc



Hình 8. Sơ đồ kiến trúc hệ thống

### Mô tả kiến trúc

Hệ thống được thiết kế với 2 phần chính, bao gồm: client-side và server-side. Cụ thể như sau:

* Client-side: Hệ thống website (Reactjs)
* Người dùng tương tác với hệ thống.
* Hệ thống trao đổi dữ liệu với server thông qua phương thức HTTP.
* Giao tiếp với một số API của các nền tảng:
* Google Map API: phục vụ việc tìm kiểm địa điểm và hiển thị bản đồ.
* Server-side: Từ API trả về dưới dạng Json
* Server trao đổi dữ liệu với ứng dụng website thông qua phương thức HTTP, server sẽ trả các dữ liệu cho client dưới dạng Json.
* Server sử dụng NodeJS kết nối tới cơ sở dữ liệu Xampp.
* Giao tiếp với một số API của các nền tảng:

## Yêu cầu chức năng

Tìm kiếm phòng trọ: Người dùng có thể tìm kiếm phòng trọ dựa trên các tiêu chí như vị trí, giá cả, diện tích, tiện ích, loại hình phòng, và các yêu cầu khác.

Xem chi tiết phòng trọ: Người dùng có thể xem thông tin chi tiết về mỗi phòng trọ bao gồm hình ảnh, mô tả, các tiện ích, quy định sử dụng, và thông tin liên lạc của chủ nhà.

Liên hệ và tương tác với bên cho thuê: Người dùng có thể liên hệ và tương tác trực tiếp với chủ nhà thông qua zalo hoặc số điện thoại.

Xem đánh giá và nhận xét: Người dùng có thể xem đánh giá và nhận xét từ những người đã ở trước đó để có cái nhìn đánh giá về chất lượng và dịch vụ của phòng trọ.

Đăng tin phòng trọ: Chủ nhà có thể đăng thông tin về phòng trọ của mình, bao gồm mô tả chi tiết, hình ảnh, giá cả, quy định sử dụng, và thông tin liên lạc.

## Usecase

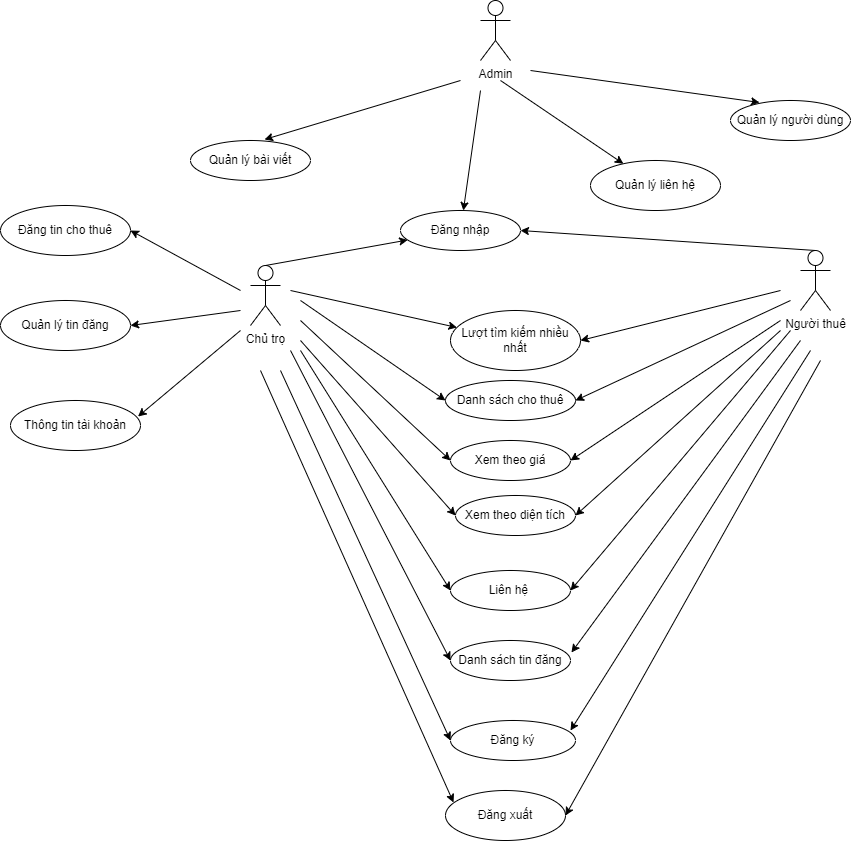
### Các tác nhân hệ thống

* Admin: Quản trị hệ thống.
* Chủ trọ: Người sử dụng hệ thống đăng tải các bài cho thuê.
* Người thuê: Người sử dụng hệ thống có nhu cầu xem thông tin phòng thuê

### Mô tả các Usecase

1. **UC#01 – Đăng nhập:** Hệ thống cho phép người dùng đăng nhập thông qua tên đăng nhập và mật khẩu để sử dụng các chức năng của hệ thống.
2. **UC#02 – Đăng ký:** Hệ thống yêu cầu người dùng là chủ trọ phải đăng ký tài khoản để có thể thực hiện các chức năng do hệ thống cung cấp. Do vậy, việc đầu tiên cần làm là tạo 1 tài khoản để sử dụng.
3. **UC#03 – Quản lý bài viết:** Hệ thống cho phép tác nhân Admin có thể quản lý các bài người dùng đăng lên.
4. **UC#04 – Quản lý liên hệ:** Hệ thống cho phép tác nhân Admin có thể quản lý các liên hệ phản hồi của người dùng gửi đến.
5. **UC#05 – Quản lý người dùng:** Hệ thống cho phép tác nhân Admin có thể quản lý thông tin người dùng đã truy cập hệ thống.
6. **UC#06 – Thông tin tài khoản:** Hệ thống cho phép người dùng có thể xem thông tin tài khoản.
7. **UC#07 – Quản lý tin đăng:** Hệ thống cho phép người dùng có thể quản lý các tin đăng.
8. **UC#08 – Đăng tin cho thuê:** Hệ thống cho phép người dùng có thể đăng tin bài viết thuê phòng.
9. **UC#09 – Lượt tìm kiếm nhiều nhất:** Hệ thống cho phép tác nhân người thuê xem thông tin các bài viết theo danh sách cho thuê như: Cho thuê căn hộ, cho thuê phòng trọ, nhà cho thuê.
10. **UC#10 – Danh sách cho thuê:** Hệ thống cho phép tác nhân người thuê xem được các bài đăng tin theo danh sách cho thuê.
11. **UC#11 – Xem theo giá:** Hệ thống cho phép tác nhân có thể xem các phòng lọc theo mức giá mong muốn.
12. **UC#12 – Liên hệ:** Hệ thống cho phép tác nhân liên hệ gửi ý kiến cho admin.
13. **UC#13 – Danh sách tin đăng:** Hệ thống cho phép tác nhân có thể xem theo danh sách tin đăng theo mặc định và theo tin đăng mới.
14. **UC#14 – Xem theo diện tích:** Hệ thống cho phép tác nhân có thể xem các phòng lọc theo diện tích mong muốn.
15. **UC#15 – Đăng xuất:** Hệ thống cho phép người dùng có thể đăng xuất khỏi hệ thống.

### Biểu đồ Usecase



Hình 9. Sơ đồ Usecase

## Đặc tả yêu cầu

1. **UC#01. Đăng nhập**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#01** | | **Đăng nhập** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng có thể đăng đăng nhập tài khoản để sử dụng các chức năng của hệ thống | |
| **Tác nhân** | | Chủ trọ, người thuê, admin | |
| **Tiền điều kiện** | | Người dùng phải có tài khoản trong hệ thống. | |
| **Hậu**  **điều kiện** | **Thành**  **công** | * Nếu đăng nhập vào trang admin: Hệ thống sẽ điều hướng về trang chủ admin * Nếu đăng nhập vào trang chủ: Hệ thống sẽ điều hướng về trang chủ. | |
| **Lỗi** | Đăng nhập tài khoản không thành công. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| *Luồng chính*  Chức năng này bắt đầu khi người dùng muốn sử dụng với vai trò là chủ trọ muốn đăng tải các bài viết về phòng/căn hộ cho thuê.   1. Người dùng truy cập vào hệ thống. 2. Hệ thống điều hướng đến trang đăng nhập. 3. Người dùng nhập thông tin đăng nhập. 4. Hệ thống kiểm tra thông tin đăng nhập. 5. Đăng nhập thành công, hệ thống chuyển hướng đến trang chủ. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| *Nhập thiếu thông tin đăng nhập*  Khi người dùng ấn đăng nhập, nếu nhập thiếu thông tin đăng nhập, hệ thống sẽ hiển thị thông báo “Bạn không được bỏ trống trường này” tại các ô trường nhập thông tin mà người dùng chưa nhập, người dùng cần nhập đầy đủ thông tin để có thể đăng nhập vào hệ thống. | | | |
| *Sai mật khẩu*  Khi hệ thống kiểm tra thông tin nhập vào của người dùng thấy mật khẩu không trùng khớp. Hệ thống sẽ đưa ra thông báo “ Mật khẩu phải có tối thiểu 6 ký tự” yêu cầu người dùng nhập chính xác mật khẩu tương ứng. | | | |
| **Giao diện minh họa** | | | |
|  | | | |

1. **UC#02. Đăng ký**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#02** | | **Đăng ký** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng có thể đăng ký tài khoản để sử dụng các chức  năng của hệ thống | |
| **Tác nhân** | | Chủ trọ, người thuê | |
| **Tiền điều kiện** | | Người dùng chọn chức năng Đăng ký. | |
| **Hậu**  **điều kiện** | **Thành**  **công** | Điều hướng về trang chủ. | |
| **Lỗi** | Đăng ký tài khoản không thành công. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| *Luồng chính:*  Chức năng này bắt đầu khi người dùng muốn đăng ký một tài khoản trên hệ thống thông qua số điện thoại   * Sau khi truy cập vào hệ thống, người dùng chọn chức năng Đăng ký tài khoản bằng cách ấn nút Đăng ký * Hệ thống sẽ hiển thị giao diện Đăng ký tài khoản yêu cầu nhập những thông tin:   + Họ tên   + Số điện thoại   + Mật khẩu * Người dùng nhập đầy đủ thông tin yêu cầu, sau đó nhấn nút “Đăng ký” để hoàn thành thao tác Đăng ký tài khoản | | | |
| **Luồng sự kiện phát sinh** | | | |
| *Nhập thiếu thông tin tài khoản*  Khi người dùng ấn đăng ký, nếu nhập thiếu thông tin tài khoản, ứng dụng sẽ hiển thị thông báo “Bạn không được bỏ trống trường này” tại các ô trường nhập thông tin mà người dùng chưa nhập, người dùng cần nhập đầy đủ thông tin để có thể đăng ký tài khoản. | | | |
| **Giao diện minh họa** | | | |
|  | | | |
|  | | | |

1. **UC#03. Quản lý bài viết**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#03** | | **Quản lý bài viết** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Chức năng này dùng để quản lý tổng hợp các bài viết của tất cả các chủ trọ đã đăng lên | |
| **Tác nhân** | | Admin | |
| **Tiền điều kiện** | | Tác nhân Admin chọn chức năng Quản lý bài viết | |
| **Hậu**  **điều kiện** | **Thành**  **công** | Hiển thị thành công tổng số bài đăng, và xóa bài đăng thành công | |
| **Lỗi** | Không hiển thị thành công tổng số bài đăng, và không xóa bài đăng được. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| *Luồng chính*  Chức năng này cho phép người dùng quản lý bài viết như:   * Tổng số các bài đăng * Tìm kiếm theo Hashtag Code * Tìm kiếm theo User ID * Xóa bài đăng. * Các trường trong danh sách bài đăng gồm:   + Id  + Mã  + Tên bài đăng  + Mô tả  + Giá  + Action   * + Tại màn hình trang chủ admin tác nhân lựa chọn “Quản lý bài viết”   + Người dùng có thể lựa chọn “xóa” để bài viết, luồng con “Xóa bài viết” được thực hiện.   + Người dùng có thể lựa chọn “Tìm kiếm” để tìm kiếm bài viết theo Hashtag Code, luồng con “Tìm kiếm theo Hastag Code” được thực hiện.   + Người dùng có thể lựa chọn “Tìm kiếm” để tìm kiếm bài viết theo User ID, luồng con “Tìm kiếm theo User ID” được thực hiện. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| *Không thể quản lý bài viết do lỗi hệ thống.*  Khi người dùng thực hiện quản lý bài viết, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, ứng dụng sẽ hiển thị thông báo cho người dùng biết và gợi ý cách giải quyết sự cố nếu có. | | | |
| **Giao diện minh họa** | | | |
|  | | | |

1. **UC#04. Thông tin cá nhân**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#04** | | **Thông tin tài khoản** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng có thể sửa thông tin cá nhân của mình | |
| **Tác nhân** | | Chủ trọ | |
| **Tiền điều kiện** | | Người dùng là chủ trọ chọn chức năng “Thông tin tài khoản”. | |
| **Hậu**  **điều kiện** | **Thành**  **công** | Cập nhật thành công thông tin cá nhân. | |
| **Lỗi** | Cập nhật không thành công thông tin cá nhân. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng này cho phép người dùng là chủ trọ có thể chỉnh sửa các thông tin cá nhân của mình như: Số điện thoại, Tên hiển thị, Ảnh đại diện   1. Người dùng chọn chức năng “Chỉnh sửa thông tin cá nhân” 2. Hệ thống hiển thị form chỉnh sửa thông tin gồm các thông tin:  * Mã thành viên * Số điện thoại * Tên hiển thị * Đường link Facebook * Zalo * Ảnh đại diện  1. Người dùng điền thông tin cần chỉnh sửa vào form 2. Click cập nhật   Hệ thống sẽ lưu lại thông tin vừa cập nhật | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **Luồng A: Huỷ xác nhận cập nhật** | | | |
| - Thông tin của user sẽ không bị thay đổi | | | |
| **Giao diện minh họa** | | | |
|  | | | |

1. **UC#05. Quản lý tin đăng**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#05** | | **Quản lý tin đăng** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng là chủ trọ có thể sửa, xóa các bài phòng đã đăng, lọc các bài trong tin đăng theo giá tiền | |
| **Tác nhân** | | Chủ trọ | |
| **Tiền điều kiện** | | Người dùng chọn chức năng Quản lý tin đăng. | |
| **Hậu**  **điều kiện** | **Thành**  **công** | Sửa, xóa các bài đã đăng, lọc các bài trong tin đăng theo giá tiền thành công. | |
| **Lỗi** | Sửa, xóa các bài đã đăng, lọc các bài trong tin đăng theo giá tiền không thành công. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| *Luồng chính*   1. Người dùng chọn chức năng “Quản lý tin đăng” 2. Hệ thống hiển thị form quản lý tin đăng gồm các thông tin:  * Lọc theo Giá * Mã tin * Ảnh đại diện * Tiêu đề * Giá * Ngày bắt đầu * Trạng thái * Tùy chọn  1. Người dùng có thể chọn “Sửa” để chỉnh sửa tin đăng 2. Người dùng có thể chọn “Xóa” để xóa tin đăng 3. Người dùng chọn “Giá” để lọc các bài viết còn   *Luồng con*   1. Sửa  * Người dùng chọn tin muốn sửa * Hệ thống hiển thì thông tin chi tiết tin đăng * Người dùng sửa lại thông tin trong bài viết * Hệ thống cập nhật lại thông tin  1. Xóa  * Người dùng chọn tin muốn xóa * Hệ thống thực hiện xóa tin và cập nhật lại danh sách  1. Lọc theo giá  * Người dùng chọn lọc theo khoảng giá * Hệ thống thực hiện lọc tin theo khoảng giá | | | |
| **Luồng sự kiện phát sinh / kịch bản phát sinh** | | | |
| **Luồng A: Người dùng thêm thiếu thông tin form** | | | |
| * Hệ thống hiển thị thông báo lỗi * Trong luồng con “Sửa”tất cả các thông tin đều bắt buộc phải điền nếu người dùng điền thiếu, hệ thống sẽ báo yêu cầu nhập đủ thông tin vào form. Người dùng nhập đủ thông tin rồi mới thực hiện thao tác tiếp theo. | | | |
| **Giao diện minh họa** | | | |
|  | | | |

1. **UC#06. Đăng tin cho thuê**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#06** | | **Đăng tin cho thuê** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng có thể đăng ký tài khoản để sử dụng các chức  năng của hệ thống | |
| **Tác nhân** | | Chủ trọ | |
| **Tiền điều kiện** | | Người dùng chọn chức năng Đăng tin cho thuê. | |
| **Hậu**  **điều kiện** | **Thành**  **công** | Điều hướng về trang đăng tin cho thuê. | |
| **Lỗi** | Không đăng tin cho thuê được. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| *Luồng chính*   1. Người dùng chọn chức năng “Đăng tin cho thuê”   Hệ thống hiển thị form Đăng tin cho thuê gồm các thông tin:   * Địa chỉ cho thuê: Người dùng thêm địa chỉ cho thuê * Thông tin mô tả: Mô tả loại hình thuê * Thông tin liên hệ: Người dùng để lại thông tin liên hệ * Đối tượng cho thuê: Người dùng lựa chọn đối tượng cho thuê * Hình ảnh: Thêm ảnh phòng   *Luồng con*   1. Địa chỉ cho thuê  * Hệ thống hiển thị form Địa chỉ cho thuê với các thông tin như * Tỉnh/Thành phố * Quận/Huyện * Người dùng thực hiện lựa chọn theo tỉnh/Thành phố, Quận/Huyện  1. Thông tin mô tả  * Hệ thống hiển thị thông tin mô tả gồm các thông tin như * Loại chuyên mục * Tiêu đề * Mô tả * Người dùng thực hiện lựa chọn loại chuyên mục theo loại hình cho thuê, ghi thông tin nội dung tiêu đề và mô tả  1. Thông tin liên hệ  * Hệ thống hiển thị thông tin liên hệ gồm các thông tin như: * Tên * Điện thoại * Giá cho thuê * Diện tích * Người dùng thực hiện ghi thông tin cho giá thuê phòng và diện tích  1. Đối tượng cho thuê  * Người dùng thực hiện lựa chọn đối tượng cho thuê  1. Hình ảnh  * Người dùng thực hiện thêm ảnh phòng cho thuê | | | |
| **Luồng sự kiện phát sinh / kịch bản phát sinh** | | | |
| **Luồng A: Người dùng không lựa chọn địa chỉ cho thuê** | | | |
| * Hệ thống sẽ hiển thị thông báo “ Bạn không được bỏ trống trường này” * Người dùng bắt buộc lựa chọn theo tỉnh/Thành phố, Quận/Huyện rồi mới thực hiện được các thao tác tiếp theo | | | |
| **Luồng A1: Người dùng không điền thông tin mô tả** | | | |
| * Hệ thống sẽ hiển thị thông báo “ Bạn không được bỏ trống trường này” * Người dùng bắt điền thông tin mô tả rồi mới thực hiện được các thao tác tiếp theo | | | |
| **Luồng A2: Người dùng không lựa chọn địa chỉ cho thuê** | | | |
| * Hệ thống sẽ hiển thị thông báo “ Bạn không được bỏ trống trường này” * Người dùng bắt buộc lựa chọn theo tỉnh/Thành phố, Quận/Huyện rồi mới thực hiện được các thao tác tiếp theo | | | |
| **Giao diện minh họa** | | | |
|  | | | |
|  | | | |

1. **UC#07. Quản lý người dùng**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#07** | | **Quản lý người dùng** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép các tác nhân admin quản lý được danh sách người dùng | |
| **Tác nhân** | | Admin | |
| **Tiền điều kiện** | | Người dùng chọn chức năng “Quản lý người dùng” . | |
| **Hậu**  **điều kiện** | **Thành**  **công** | Xem được danh sách người dùng. | |
| **Lỗi** | Không xem được danh sách người dùng | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| * Hệ thống hiển thị danh sách người dùng hệ thống sẽ bao gồm các thông tin như:   + ID  + Tên  + SĐT  + Xóa   * Hệ thống cho xem theo danh sách người dùng và có thể xóa người dùng | | | |
| **Luồng sự kiện kịch bản phát sinh** | | | |
| *Không thể quản lý người dùng do lỗi hệ thống.*   * Khi người dùng thực hiện quản lý người dùng, nếu xảy ra lỗi từ hệ thống hoặc lỗi kết nối tới hệ thống, ứng dụng sẽ hiển thị thông báo cho người dùng biết và gợi ý cách giải quyết sự cố nếu có. | | | |
| **Giao diện minh họa** | | | |
|  | | | |

1. **UC#08. Danh sách tin đăng**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#08** | | **Danh sách tin đăng** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng có thể xem theo tin “Mặc định”, “Mới nhất” | |
| **Tác nhân** | | Người thuê, chủ trọ | |
| **Tiền điều kiện** | | Người dùng chọn xem danh sách tin đăng | |
| **Hậu**  **điều kiện** | **Thành**  **công** | Xem được tin đăng mặc định, tin mới đăng, bài viết chi tiết | |
| **Lỗi** | Không xem được các bài mới đăng trong 3-5 ngày | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| *Luồng chính*   1. Hệ thống hiển thị bài viết phòng cho thuê   *Luồng con*   1. Mặc định  * Hệ thống hiển thị các bài đăng mặc định * Người dùng chọn xem một bài viết chi tiết * Hệ thống sẽ link sang một trang xem chi tiết bài viết người dùng đã chọn và hiển thị các thông tin như:   + Tiêu đề bài viết  + Chuyên mục  + Giá tiền  + Diện tích  + Thông tin mô tả  + Đặc điểm tin đăng  + Thông tin liên hệ   1. Mới nhất  * Hệ thống hiển thị các bài đăng mới đăng gần nhất, sẽ được sắp xếp bài đăng gần nhất từ trên xuống dưới * Người dùng chọn xem một bài viết chi tiết * Hệ thống sẽ link sang một trang xem chi tiết bài viết người dùng đã chọn và hiển thị các thông tin như:   + Tiêu đề bài viết  + Chuyên mục  + Giá tiền  + Diện tích  + Thông tin mô tả  + Đặc điểm tin đăng  + Thông tin liên hệ | | | |
| **Luồng sự kiện kịch bản phát sinh** | | | |
| * Không xem được bài viết | | | |
| **Giao diện minh họa** | | | |
|  | | | |
|  | | | |

1. **UC#09. Xem theo giá**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#09** | | **Xem theo giá** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng có thể xem theo giá tiền mong muốn thuê | |
| **Tác nhân** | | Người thuê, Chủ trọ | |
| **Tiền điều kiện** | | Người dùng chọn chức năng Xem theo giá. | |
| **Hậu**  **điều kiện** | **Thành**  **công** | Điều hướng về trang Xem theo khoảng giá lựa chọn | |
| **Lỗi** | Không xem được theo giá tiền | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| * Người dùng vào trang chủ * Lựa chọn theo khoảng giá tiền sẽ bao gồm các khoảng giá   + Dưới 1 triệu  + Từ 2-3 triệu  + Từ 5-7 triệu  + Từ 10-15 triệu  + Từ 1-2 triệu  + Từ 3-5 triệu  + Từ 7-10 triệu  + Trên 15 triệu   * Hệ thống hiển thị theo danh mục đã chọn | | | |
| **Luồng sự kiện kịch bản phát sinh** | | | |
| Không hiển thị đúng khoảng giá lựa chọn | | | |
| **Giao diện minh họa** | | | |
|  | | | |

1. **UC#10. Liên hệ**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#10** | | **Liên hệ** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng có thể liên hệ với admin gửi lại phản hồi cho admin | |
| **Tác nhân** | | Người thuê, chủ trọ | |
| **Tiền điều kiện** | | Người dùng chọn chức năng Liên hệ. | |
| **Hậu**  **điều kiện** | **Thành**  **công** | Gửi ý kiến cho Admin thành công. | |
| **Lỗi** | Lỗi gửi ý kiến cho Admin thành công. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| * Người dùng chọn “Liên hệ” * Trong trang “Liên hệ” sẽ có các trường như:   + Họ và tên của bạn  + Số điện thoại  + Nội dung mô tả   * Chọn “Gửi ngay” sẽ hoàn thành ý kiến gửi. | | | |
| **Luồng sự kiện kịch bản phát sinh** | | | |
| Không gửi được ý kiến, lỗi gửi. | | | |
| **Giao diện minh họa** | | | |
|  | | | |
|  | | | |

1. **UC#11. Lượt tìm kiếm nhiều nhất**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#11** | | **Lượt tìm kiếm nhiều nhất** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng có thể chọn xem được các bài viết có lượt tìm kiếm nhiều nhất được chia theo 3 khu:Hà Nội, Hồ Chí Minh, Đà Nẵng | |
| **Tác nhân** | | Người thuê, Chủ trọ | |
| **Tiền điều kiện** | | Người dùng chọn chức năng Lượt tìm kiếm nhiều nhất. | |
| **Hậu**  **điều kiện** | **Thành**  **công** | Xem được các bài viết có lượt tìm kiếm nhiều nhất theo đúng khu vực. | |
| **Lỗi** | Không xem được các bài viết có lượt tìm kiếm nhiều nhất theo đúng khu vực. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| * Người dùng vào trang chủ * Lựa chọn xem lượt tìm kiếm nhiều nhất có bao gồm:   + Phòng trọ Hồ Chí Minh  + Phòng trọ Hà Nội  + Phòng trọ Đà Nẵng  Hệ thống hiển thị theo danh mục đã chọn | | | |
| **Luồng sự kiện kịch bản phát sinh** | | | |
| Không hiển thị đúng lượt tìm kiếm nhiều nhất | | | |
| **Giao diện minh họa** | | | |
|  | | | |
|  | | | |

1. **UC#12. Xem theo diện tích**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#12** | | **Xem theo diện tích** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng có thể xem theo diện tích cần thuê | |
| **Tác nhân** | | Người thuê, Chủ trọ | |
| **Tiền điều kiện** | | Người dùng chọn chức năng Xem theo diện tích. | |
| **Hậu**  **điều kiện** | **Thành**  **công** | Xem được bài viết theo đúng diện tích lựa chọn. | |
| **Lỗi** | Không xem được bài viết theo đúng diện tích lựa chọn. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| * Người dùng vào trang chủ * Lựa chọn xem theo diện tích sẽ bao gồm các kích thước có diện tích như:   + Dưới 20m  + Từ 30-50m  + Từ 70-90m  + Từ 20-30m  + Từ 50-70m  + Trên 90m  Hệ thống hiển thị theo danh mục đã chọn | | | |
| **Luồng sự kiện kịch bản phát sinh** | | | |
| Không hiển thị đúng diện tích lựa chọn | | | |
| **Giao diện minh họa** | | | |
|  | | | |
|  | | | |

1. **UC#13. Danh sách cho thuê**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#13** | | **Danh sách cho thuê** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng có thể xem theo danh sách cho thuê được chia làm ba loại như : Cho thuê căn hộ, Cho thuê phòng trọ, Nhà cho thuê | |
| **Tác nhân** | | Người thuê, Chủ trọ | |
| **Tiền điều kiện** | | Người dùng chọn chức năng Danh sách cho thuê. | |
| **Hậu**  **điều kiện** | **Thành**  **công** | Xem được đúng theo danh sách cho thuê. | |
| **Lỗi** | Không xem được đúng theo danh sách cho thuê. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| * Người dùng vào trang chủ * Lựa chọn theo danh sách cho thuê sẽ bao gồm các loại hình thuê như:   + Cho thuê căn hộ  + Cho thuê phòng trọ  + Nhà cho thuê  Hệ thống hiển thị theo danh mục đã chọn | | | |
| **Luồng sự kiện kịch bản phát sinh** | | | |
| Không hiển thị đúng loại hình thuê lựa chọn | | | |
| **Giao diện minh họa** | | | |
|  | | | |
|  | | | |

1. **UC#14. Quản lý liên hệ**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#13** | | **Quản lý liên hệ** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép Admin có thể Quản lý liên hệ các ý kiến gửi về và có thể xóa liên hệ đấy | |
| **Tác nhân** | | Admin | |
| **Tiền điều kiện** | | Người dùng chọn chức năng Quản lý liên hệ. | |
| **Hậu**  **điều kiện** | **Thành**  **công** | Truy cập được vào màn hình Quản lý liên hệ . | |
| **Lỗi** | Không truy cập được vào màn hình Quản lý liên hệ. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng này cho phép người dùng quản liên hệ như:   * Tổng số các ý kiến * Xóa bài ý kiến. * Các trường trong danh sách bài đăng gồm:   + Id  + Name  + Phone  + Contact  + Action   * + Tại màn hình trang chủ admin tác nhân lựa chọn “Quản lý liên hệ”   + Người dùng có thể lựa chọn “xóa” để bài viết, luồng con “Xóa ý kiến” được thực hiện. | | | |
| **Luồng sự kiện kịch bản phát sinh** | | | |
| Không hiển thị được danh sách ý kiến phản hồi và xóa danh sách phản hồi. | | | |
| **Giao diện minh họa** | | | |
|  | | | |
|  | | | |

1. **UC#15. Đăng xuất**

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#14** | | **Đăng xuất** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Cho phép người dùng có thể đăng xuất tài khoản. | |
| **Tác nhân** | | Admin, chủ trọ, người thuê | |
| **Tiền điều kiện** | | Người dùng chọn chức năng Đăng xuất. | |
| **Hậu**  **điều kiện** | **Thành**  **công** | Hệ thống điều hướng về giao diện đăng nhập. | |
| **Lỗi** | Trạng thái hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| 1. Người dùng truy cập vào hệ thống. 2. Người dùng chọn nút “Đăng xuất”.   Tài khoản người dùng được đăng xuất khỏi hệ thống. Hệ thống hiển thị giao diện đăng nhập. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **Luồng A: Đăng xuất thất bại** | | | |
| - Trạng thái hệ thống không thay đổi | | | |
| **Giao diện minh họa** | | | |
| C:\Users\Thuyen\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\75f6754a-4add-4e7a-811e-2ac226785d69.jpeg  C:\Users\Thuyen\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\25cc44c5-5f84-4d5f-afbc-fc297764979f (1).jpeg | | | |

# Thiết kế hệ thống

## Thiết kế cơ sở dữ liệu

### Lược đồ cơ sở dữ liệu

A screenshot of a computer

Description automatically generated

Hình 10. Lược đồ cơ sở dữ liệu

### Danh sách các bảng

| **STT** | **Tên bảng** | **Mô tả** |
| --- | --- | --- |
| 1 | Users | Thông tin tài khoản người dùng |
| 2 | Areas | Diện tích phòng |
| 3 | Attributes | Lưu trữ thông tin chi tiết bài viết cho từng phòng thuê |
| 4 | Prices | Lưu trữ giá phòng |
| 5 | Posts | Lưu trữ thông tin bài viết |
| 6 | Provinces | Lưu khu vực tỉnh thành |
| 7 | Categories | Loại hình thuê |
| 8 | Overviews | Tổng quan bài viết |
| 9 | Images | Lưu trữ ảnh phòng |
| 10 | Labels | Chuyên mục |
| 11 | Contacts | Lưu ý kiến phản hồi gửi từ liên hệ |
| 12 | Admins | Thông tin người quản lý tải khoản, bài viết và ý kiển phản hồi |

Bảng 3.1. Bảng mô tả tên bảng dữ liệu

### Chi tiết các bảng

1. Users – Thông tin tài khoản

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Miêu tả** |
| --- | --- | --- | --- |
| 1 | Id | Varchar | Khóa chính |
| 2 | Name | Varchar | Tên |
| 3 | Password | Varchar | Mật khẩu |
| 4 | phone | Varchar | Số điện thoại |
| 5 | Zalo | Varchar | Zalo |
| 6 | fbUrl | Varchar | Facebook |
| 7 | Avatar | Longblob | Ảnh đại diện |
| 8 | Role | Int(11) | vai trò người dùng |
| 9 | createdAt | Datetime | Ngày tạo |
| 10 | updatedAt | datetime | Ngày cập nhật |

Bảng 3.2. Bảng Users

1. Areas – Diện tích phòng

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Miêu tả** |
| --- | --- | --- | --- |
| 1 | Id | Int | Khóa chính |
| 2 | Order | Int | Số lượng được đặt |
| 3 | Code | Varchar | Code |
| 4 | Value | Varchar | Giá trị diện tích trong khoảng |
| 5 | createdAt | Datetime | Ngày tạo |
| 6 | UpdatedAt | Datetime | Ngày cập nhật |

Bảng 3.3. Bảng Areas

1. Attributes – Lưu trữ thông tin chi tiết bài viết

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Miêu tả** |
| --- | --- | --- | --- |
| 1 | Id | Varchar | Khóa chính |
| 2 | Price | Varchar | Giá tiền |
| 3 | Acreage | Varchar | Diện tích phòng |
| 4 | Published | Varchar | Khoảng thời gian tạo bài |
| 5 | Hashtag | Varchar | Thứ tự bài |
| 6 | createdAt | Datetime | Ngày tạo |
| 7 | updatedAt | Datetime | Ngày cập nhật |

Bảng 3.4. Attributes

1. Prices – Lưu trữ giá phòng

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Miêu tả** |
| --- | --- | --- | --- |
| 1 | Id | Int | Khóa chính |
| 2 | Order | Int | Dùng sắp xếp |
| 3 | Code | Varchar | Code |
| 4 | Value | Varchar | Giá trị khoảng giá tiền |
| 5 | createdAt | Datetime | Ngày tạo |
| 6 | UpdatedAt | Datetime | Ngày cập nhật |

Bảng 3.5. Prices

1. Posts – Lưu trữ thông tin bài viết

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Miêu tả** |
| --- | --- | --- | --- |
| 1 | Id | Varchar | Khóa chính |
| 2 | Title | Varchar | Tiêu đề bài viết |
| 3 | Star | Varchar | Số phòng có |
| 4 | LabelCode | Varchar | Lưu trữ mã chuyên mục |
| 5 | Address | Varchar | Địa chỉ |
| 6 | Attributesld | Varchar | Thông tin bài viết |
| 7 | categoryCode | Varchar | Lưu trữ mã loại hình thuê |
| 8 | priceCode | Varchar | Mã tiền |
| 9 | provinceCode | Varchar | Mã tỉnh |
| 10 | areaCode | Varchar | Mã diện tích |
| 11 | Description | longtext | Sự miêu tả |
| 12 | Userld | Varchar | Người dùng |
| 13 | Overviewld | Varchar | Tổng quan |
| 14 | Imagesld | Varchar | Hình ảnh |
| 15 | priceNumber | Float | Số tiền |
| 16 | areaNumber | Float | Số diện tích |
| 17 | createdAt | Datetime | Ngày tạo |
| 18 | updatedAt | Datetime | Ngày cập nhật |

Bảng 3.6. Bảng Posts

1. Provinces – Lưu khu vực tỉnh thành

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Miêu tả** |
| --- | --- | --- | --- |
| 1 | Id | Int | Khóa chính |
| 2 | Code | Varchar | Code |
| 3 | Value | Varchar | Tỉnh thành |
| 4 | createdAt | Datetime | Ngày tạo |
| 5 | UpdatedAt | Datetime | Ngày cập nhật |

Bảng 3.7. Bảng Provinces

1. Categories – Loại hình thuê

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Miêu tả** |
| --- | --- | --- | --- |
| 1 | Id | Int | Khóa chính |
| 2 | Code | Varchar | Code |
| 3 | Value | Varchar | Giá trị |
| 4 | Header | Varchar | Header |
| 5 | Subheader | Varchar | subheader |
| 6 | CreatedAt | Datetime | Ngày tạo |
| 7 | UpdatedAt | Datetime | Ngày cập nhật |

Bảng 3.8. Bảng Categories

1. Overviews – Tổng quan bài viết

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Miêu tả** |
| --- | --- | --- | --- |
| 1 | Id | Varchar | Khóa chính |
| 2 | Code | Varchar | Code |
| 3 | Area | Varchar | Area |
| 4 | Type | Varchar | Kiểu thuê |
| 5 | Target | Varchar | Mục tiêu |
| 6 | bonus | Varchar | bonus |
| 7 | Created | Datetime | Ngày tạo |
| 8 | expired | Datetime | Ngày cập nhật |
| 9 | CreatedAt | Datetime | Ngày tạo |
| 10 | UpdatedAt | Datetime | Ngày cập nhật |

Bảng 3.9. Overviews

1. Images – Lưu ảnh phòng

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Miêu tả** |
| --- | --- | --- | --- |
| 1 | Id | Varchar | Khóa chính |
| 2 | image | Longtext | Ảnh |
| 3 | CreatedAt | Datetime | Ngày tạo |
| 4 | UpdatedAt | Datetime | Ngày cập nhật |

Bảng 3.10. Images

1. Labels – Chuyên mục

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Miêu tả** |
| --- | --- | --- | --- |
| 1 | Id | Varchar | Khóa chính |
| 2 | Code | Varchar | Code |
| 3 | value | Varchar | Value |
| 6 | CreatedAt | Datetime | Ngày tạo |
| 5 | UpdatedAt | Datetime | Ngày cập nhật |

Bảng 3.11. Labels

1. Contacts – Ý kiến phản hồi từ người dùng

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Miêu tả** |
| --- | --- | --- | --- |
| 1 | Id | Int(11) | Khóa chính |
| 2 | Name | Varchar | Tên |
| 3 | Phone | Int(11) | Số điện thoại |
| 4 | Content | Varchar | Nội dung |
| 5 | Role | Int(11) | Vai trò quản lý ý kiến |
| 6 | CreatedAt | Datetime | Ngày tạo |
| 5 | UpdatedAt | Datetime | Ngày cập nhật |

Bảng 3.12. Contacts

1. Admins – Quản lý tài khoản người dùng, bài viết và ý kiến phản hồi

| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Miêu tả** |
| --- | --- | --- | --- |
| 1 | Id | Int(11) | Khóa chính |
| 2 | Admin\_name | Varchar | Tên Admin đăng nhập |
| 3 | Admin\_password | Varchar | Mật khẩu Admin |
| 4 | Admin\_role | Int(11) | Vai trò quản lý Users |
| 5 | Contact\_role | Int(11) | Vai trò quản lý Contacts |

Bảng 3.13. Admins

## Thiết kế API

| **STT** | **URL** | **Phương thức** |
| --- | --- | --- |
| **Auth** | | |
| 1 | http://localhost:5000/api/v1/auth/register | POST |
| 2 | http://localhost:5000/api/v1/auth/login | POST |
| **Category** | | |
| 7 | http://localhost:5000/api/v1/category/all | GET |
|  |  |  |
| **Post** | | |
| 12 | http://localhost:5000/api/v1/post/all | GET |
| 13 | http://localhost:5000/api/v1/post/limit | GET |
| 14 | http://localhost:5000/api/v1/post/new\_post | GET |
| 15 | http://localhost:5000/api/v1/post/create\_new | POST |
| 16 | http://localhost:5000/api/v1/post/limit\_user | GET |
| 17 | http://localhost:5000/api/v1/post/update | PUT |
| 18 | http://localhost:5000/api/v1/post/delete | DELETE |
| **Insert** | | |
| 28 | http://localhost:5000/api/v1/insert/ | POST |
| **Price** | | |
|  | http://localhost:5000/api/v1/price/all | GET |
| Area | | |
|  | http://localhost:5000/api/v1/area/all | GET |
| Province | | |
|  | http://localhost:5000/api/v1/province/all | GET |
| Contact | | |
|  | http://localhost:5000/api/v1/contact/create-contact | POST |

### Thiết kế chi tiết API

#### API Authentication

| **URL** | **Mô tả chi tiết** |
| --- | --- |
| http://localhost:5000/api/v1/auth/register | - Mục đích: Giúp tạo tài khoản người dùng  - Tham số:      + Body: name, password, phone |
| http://localhost:5000/api/v1/auth/login | - Mục đích: Giúp người dùng đăng nhập vào tài khoản  - Tham số:      + Body: phone, password |

#### API Category

| **URL** | **Mô tả chi tiết** |
| --- | --- |
| http://localhost:5000/api/v1/category/all | - Mục đích: Giúp người dùng chọn danh mục cho thuê |

#### API Post

| **URL** | **Mô tả chi tiết** |
| --- | --- |
| http://localhost:5000/api/v1/post/all | - Mục đích: Lấy tất cả bài viết |
| http://localhost:5000/api/v1/post/limit | - Mục đích: Giúp người dùng lấy bài viết theo giới hạn mong muốn |
| http://localhost:5000/api/v1/post/new\_post | - Mục đích: Giúp người dùng lấy bài viết mới nhất |
| http://localhost:5000/api/v1/post/create\_new | - Mục đích: giúp người dung tạo bài viết mới |
| http://localhost:5000/api/v1/post/limit\_user | - Mục đích: Lấy tất cả  bài viết của người dùng cụ thể  - Tham số:      + Body: idUser |
| http://localhost:5000/api/v1/post/update | - Mục đích: giúp người dùng sửa bài viết của chính mình  - Tham số:      + Body: idUser |
| http://localhost:5000/api/v1/post/delete | - Mục đích: giúp người dùng xóa bài viết của chính mình  - Tham số:      + Body: idUser |

#### API Insert

|  |  |
| --- | --- |
| **URL** | **Mô tả chi tiết** |
| http://localhost:5000/api/v1/insert | - Mục đích: dùng để thêm các bài viết từ file data |

#### API Price

|  |  |
| --- | --- |
| **URL** | **Mô tả chi tiết** |
| http://localhost:5000/api/v1/price/all | - Mục đích: dùng để lấy giá trị các khoảng giá tiền |

#### API Area

|  |  |
| --- | --- |
| **URL** | **Mô tả chi tiết** |
| http://localhost:5000/api/v1/area/all | - Mục đích: dùng để lấy giá trị các khoảng diện tích |

1. Province

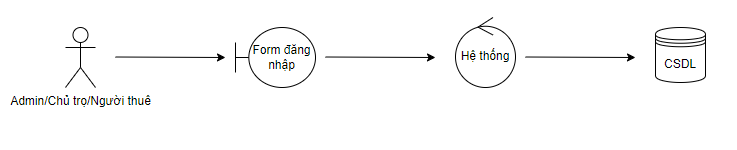
|  |  |
| --- | --- |
| **URL** | **Mô tả chi tiết** |
| http://localhost:5000/api/v1/insert | - Mục đích: dùng để thêm các bài viết từ file data |

#### API Contact

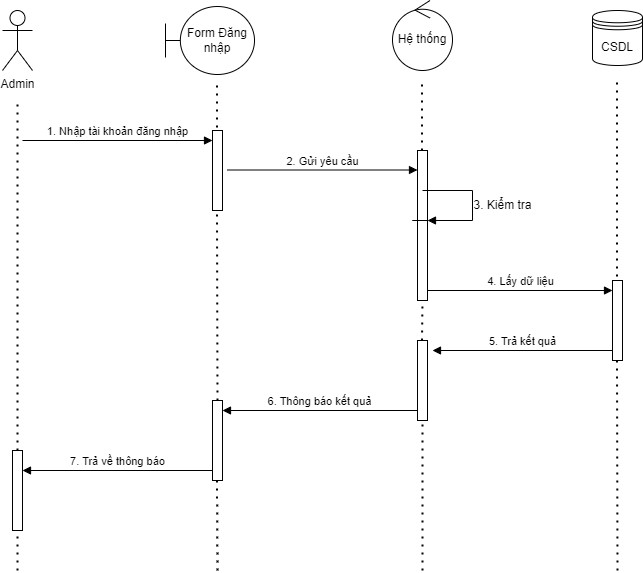
|  |  |
| --- | --- |
| **URL** | **Mô tả chi tiết** |
| http://localhost:5000/api/v1/contact/create\_contact | - Mục đích: Giúp người dùng gửi ý kiến phản hồi cho quản lý |

## Thiết kế một số chức năng chính

### Chức năng Đăng nhập

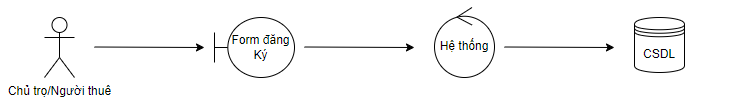


Hình 11. Sơ đồ lớp phân tích Đăng nhập

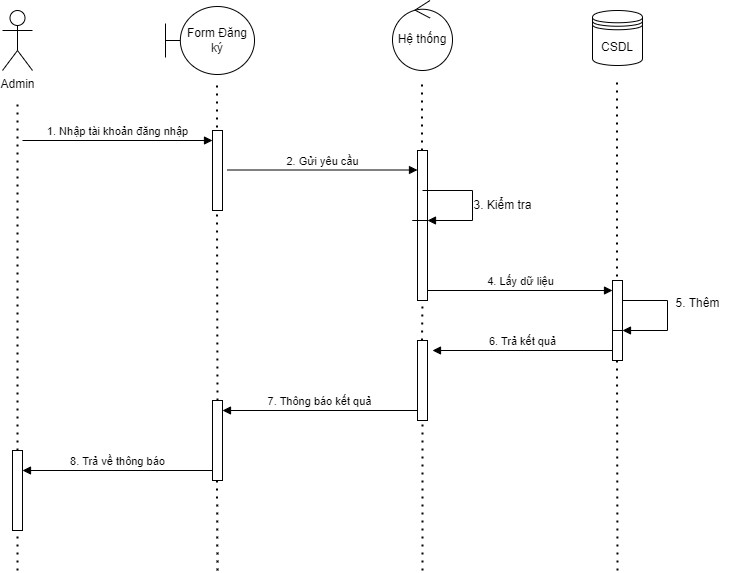


Hình 12. Sơ đồ trình tự Đăng nhập

### Chức năng Đăng ký

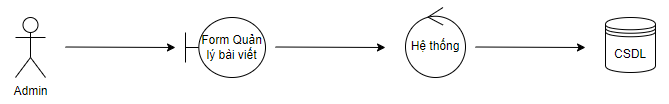


Hình 13. Sơ đồ lớp phân tích Đăng ký

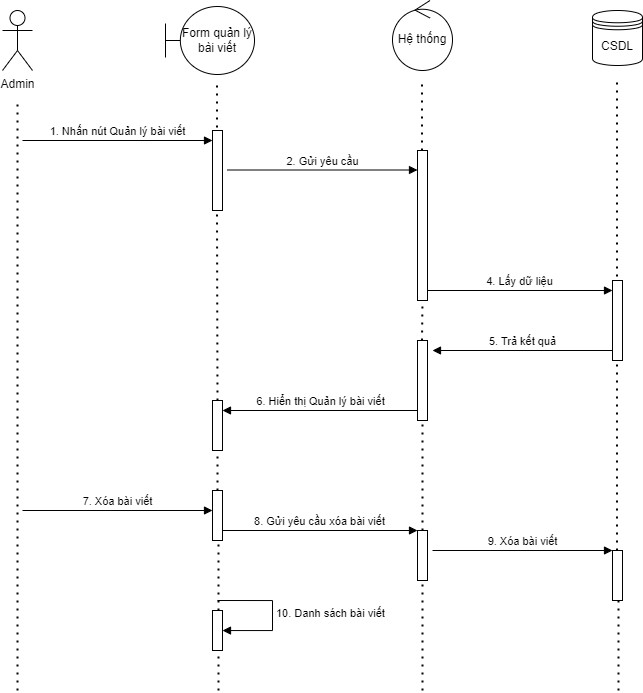


Hình 14. Sơ đồ lớp phân tích Đăng nhập

### Chức năng Quản lý bài viết

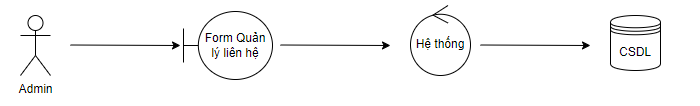


Hình 15. Sơ đồ lớp phân tích Quản lý bài viết

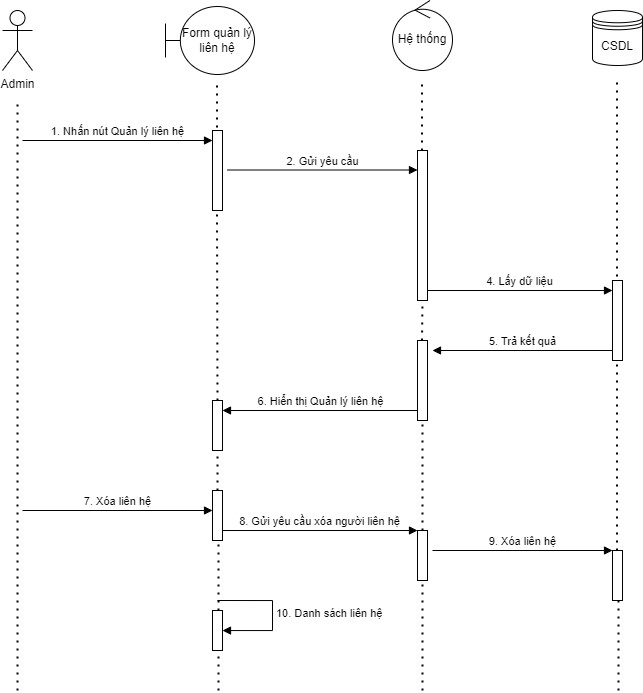


Hình 16. Sơ đồ trình tự quản lý bài viết

### Chức năng Quản lý liên hệ



Hình 17. Sơ đồ lớp phân tích Quản lý liên hệ



Hình 18. Sơ đồ trình tự quản lý liên hệ

# Cài đặt hệ thống và Thử nghiệm

## Môi trường cài đặt

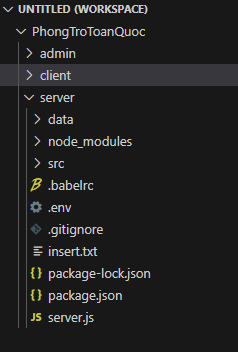
Trong quá trình cài đặt, các giải pháp công nghệ được ứng dụng trong hệ thống bao gồm:

|  |  |
| --- | --- |
| **Yêu cầu** | **Công nghệ** |
| Hệ điều hành | Windows 10 Pro |
| Môi trường server | NodeJS v6.2.2 |
| Môi trường ứng dụng | Reactjs |
| IDE (Môi trường phát triển) | Visual Studio Code |
| Hệ quản trị CSDL | MySQL, Xampp |
| VCS (Hệ thống quản lý phiên bản) | Git – Github |
| Công cụ xử lý văn bản | MS Office – Word 2010 |
| Công cụ quản lý tiến độ dự án | MS Office – Excel 2010 |
| Công cụ tạo báo cáo thuyết trình | MS Office – Power point 2010 |

*Bảng 4.1. Các môi trường cài đặt hệ thống*

## Cấu trúc thư mục mã nguồn

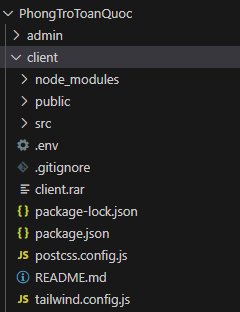
### Server



Hình 19. Cấu hình thư mục server

|  |  |
| --- | --- |
| **Folder/File** | **Mô tả** |
| node\_modules | Thư mục này được tạo tự động khi bạn cài đặt các mô-đun JavaScript của bên thứ ba. Nó chứa các mô-đun đã được biên dịch và sẵn sàng sử dụng. |
| public | Thư mục này chứa các thư viện JavaScript bên ngoài được sử dụng bởi ứng dụng. |
| src | Thư mục src là thư mục chứa toàn bộ code lúc viết, chỉnh sửa và phát triển code ở trong này |
| package-lock.json | File này dùng để cố định version khi chạy npm install |
| package.json | File cấu hình của npm, giúp cho npm hiểu nó cần phải cài đặt cái gì, thông tin về ứng dụng, phiên bản |
| data | Thư mục này chứa dữ liệu các file loại hình danh mục |
| .babelrc | File cấu hình Babel, dùng để thiết lập các plugin và preset cho quá trình chuyển đổi mã JavaScript. |
| .env | File chứa các biến môi trường chung cho toàn dự án |
| .gitignore | File này xác định các file và thư mục mà Git sẽ bỏ qua không theo dõi. |

### Client

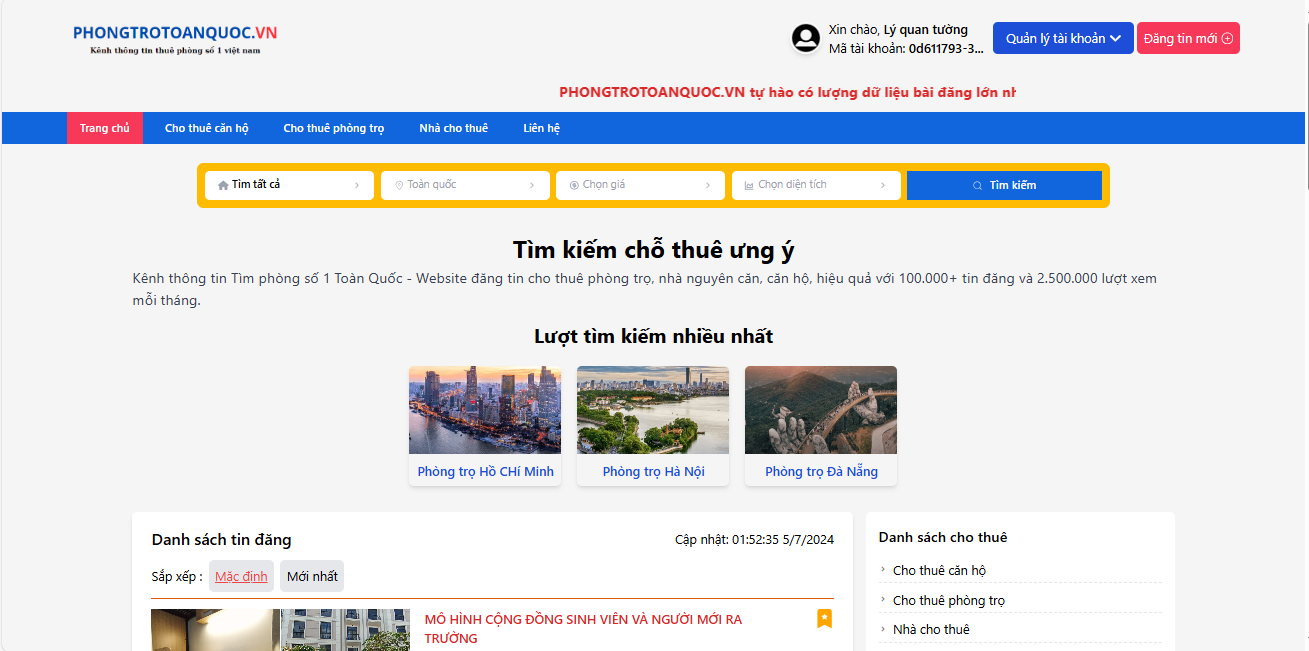


Hình 10. Cấu hình thư mục client

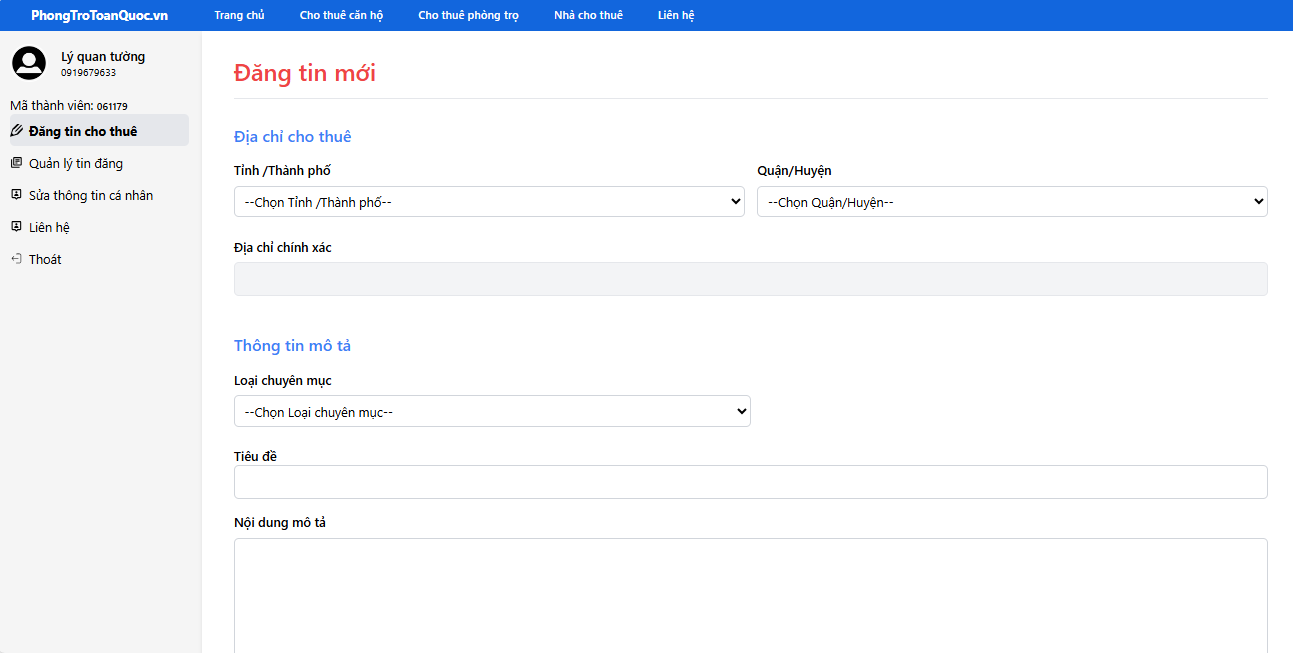
|  |  |
| --- | --- |
| **Folder/File** | **Mô tả** |
| node\_modules | Thư mục này được tạo tự động khi bạn cài đặt các mô-đun JavaScript của bên thứ ba. Nó chứa các mô-đun đã được biên dịch và sẵn sàng sử dụng. |
| public | Thư mục này chứa các thư viện JavaScript bên ngoài được sử dụng bởi ứng dụng. |
| src | Thư mục src là thư mục chứa toàn bộ code lúc viết, chỉnh sửa và phát triển code ở trong này |
| package-lock.json | File này dùng để cố định version khi chạy npm install |
| package.json | File cấu hình của npm, giúp cho npm hiểu nó cần phải cài đặt cái gì, thông tin về ứng dụng, phiên bản |
| .gitignore | File này xác định các file và thư mục mà Git sẽ bỏ qua không theo dõi. |

## Thử nghiệm

Sau đây là một số màn giao diện chính của hệ thống:



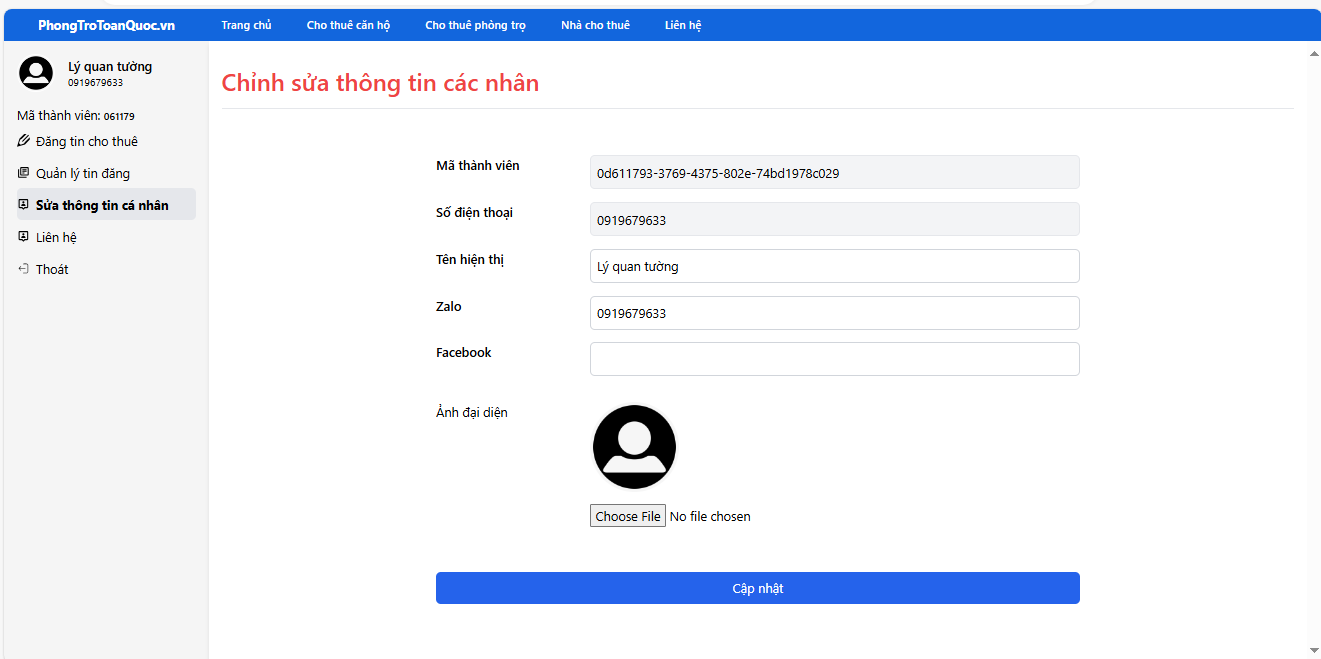
Hình 21. Giao diện trang chủ



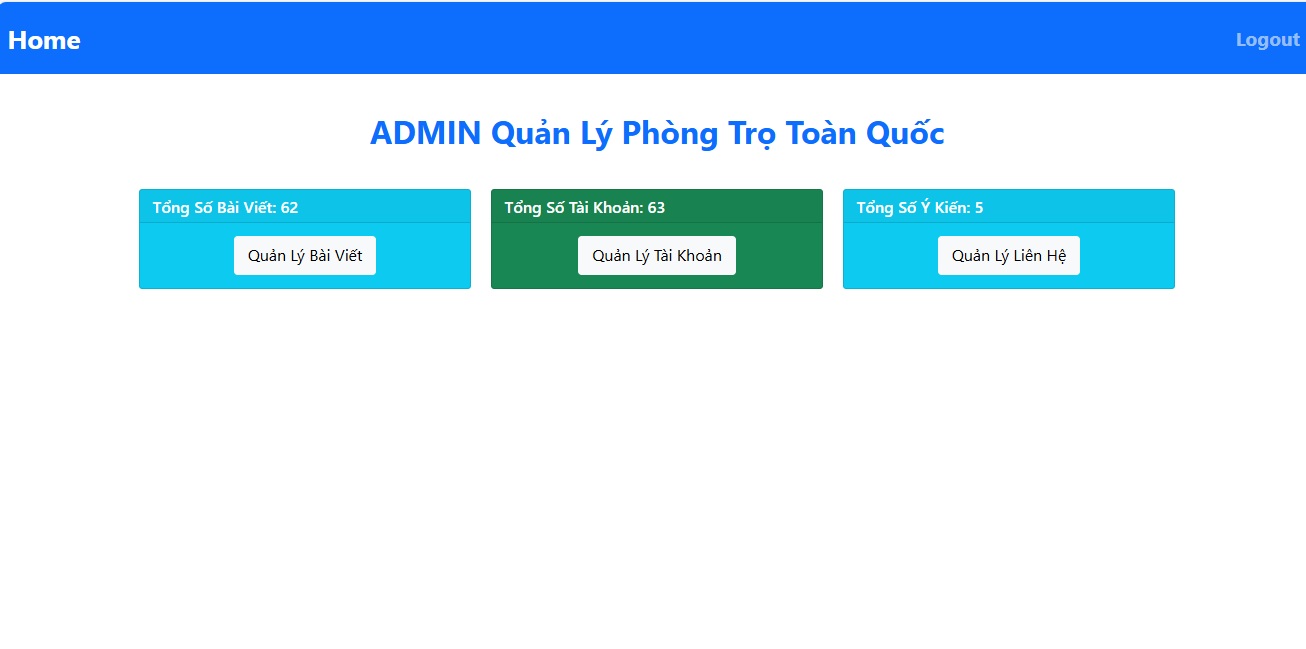
Hình 22. Giao diện đăng tin cho thuê

****

Hình 23. Giao diện quản lý tin đăng



Hình 24. Giao diện sửa thông tin cá nhân



Hình 25. Giao diện trang chủ trang Admin

## Kiểm thử

Trong phần này, trình bày kịch bản kiểm thử  và kết quả kiểm thử cho website.

### UC giao diện chung của màn người dùng

| **STT** | **Mục tiêu kiểm thử** | **Ý nghĩ** | **Pass** | **Fail** |
| --- | --- | --- | --- | --- |
| 1 | Kiểm tra giao diện chung | Kiểm tra giao diện chính của ứng dụng sau khi đăng nhập. Đảm bảo rằng các phần như số Lượt tìm kiếm nhiều nhất, Danh sách cho thuê, Xem theo giá, Xem theo diện tích được hiển thị chính xác.  Kiểm tra các nút hoặc các menu để truy cập các tính năng quan trọng như thêm Đăng tin cho thuê, Quản lý tin đăng, Thông tin tài khoản, Liên hệ. | 2 | 0 |
| 2 | Kiểm tra validate giao diện Thông tin tài khoản | Kiểm tra validate control cập nhật tài khoản cá nhân | 1 | 0 |
| 3 | Kiểm tra chức năng các button khác trong giao diện cập nhật tài khoản | Kiểm tra về các button, textbox có thực hiện đúng chức năng mong muốn của người sử dụng hay không | 1 | 0 |
| Tổng số TestCase Pass/Fail | | | 4 | 0 |

### UC giao diện chung của màn Admin

| **STT** | **Mục tiêu kiểm thử** | **Ý nghĩ** | **Pass** | **Fail** |
| --- | --- | --- | --- | --- |
| 1 | Kiểm tra giao diện chung | Kiểm tra giao diện chính của ứng dụng sau khi đăng nhập. Đảm bảo hiển thị được Tổng số bài viết, Tổng số người dùng, Tổng số liên hệ. | 1 | 0 |
| 2 | Kiểm tra validate giao diện Tổng số bài viết | Kiểm tra validate control Tổng số bài viết | 1 | 0 |
| 3 | Kiểm tra chức năng các button khác trong giao diện Tổng số bài viết | Kiểm tra về các button, textbox có thực hiện đúng chức năng mong muốn của người sử dụng hay không | 1 | 0 |
| Tổng số TestCase Pass/Fail | | | 3 | 0 |

### Kiểm thử UC dashboard

| **STT** | **Mục tiêu kiểm thử** | **Ý nghĩ** | **Pass** | **Fail** |
| --- | --- | --- | --- | --- |
| 1 | Kiểm tra giao diện chung | - Kiểm tra lại tổng thể giao diện chung của dashboard  - Màn hình ở trạng thái khởi tạo đã được hiển thị đúng hay chưa (font chữ, label, textbox, ...)  - Kiểm tra hiển thị của Xem theo giá  - Kiểm tra hiển thị của Xem theo diện tích | 3 | 0 |
| 2 | Kiểm tra Xem theo giá | - Kiểm tra hiển thị khi chọn xem theo giá có hiển thị chính xác khoảng giá lựa chọn không | 1 | 0 |
| 2 | Kiểm tra Xem diện tích | - Kiểm tra hiển thị khi chọn xem theo diện tích có hiển thị chính xác khoảng diện tích lựa chọn không | 1 | 0 |
| Tổng số TestCase Pass/Fail | | | 4 | 0 |

# Kết luận

## Kết quả đạt được

Sau khi thực hiện Chuyên đề tốt nghiệp, chúng em đã hoàn thành đề tài “Xây dựng phần mềm tìm kiếm và quản lý phòng trọ, căn hộ, nhà cho thuê” với đầy đủ các chức năng cơ bản như: Quản lý bài viết cho thuê, Tìm phòng,...

Sau khi hoàn thiện đề tài, chúng em đã học và tích lũy được nhiều kiến thức và kỹ năng trong quá trình phát triển phần mềm như:

* Phân tích yêu cầu nghiệp vụ
* Tìm hiểu và phân tích giải pháp phù hợp cho bài toán
* Thiết kế các mô hình đối tượng, các sơ đồ quan hệ
* Kỹ năng viết tài liệu
* Kỹ năng quản lý công việc.

Thêm vào đó chúng em cũng tích lũy được một số kinh nghiệm về ngôn ngữ lập trình Reactjs, Nodejs cũng như tìm hiểu một số ngôn ngữ khác như javascript, html, css, ... và các thư viện liên quan để tạo ra một sản phẩm web hoàn chỉnh. Đồng thời trau dồi được nhiều kiến thức cũng như kỹ năng lập trình, phát triển khả năng tìm hiểu và học hỏi, phục vụ cho việc phát triển sản phẩm website nói riêng hay bất kỳ loại hình ứng dụng nào.

## Hạn chế

Cuối dùng, do khối lượng công việc và khả năng có hạn về kiến thức cũng như thời gian nên diễn đàn chưa được hoàn thiện nhất và không tránh khỏi những sai sót không đáng có. Vì vậy chúng em rất mong nhận được những ý kiến góp ý của thầy cô trong hội đồng để sản phẩm được hoàn thiện hơn và được ứng dụng vào thực tế.

## Hướng phát triển

* **Mở rộng phạm vi hoạt động**: Hệ thống có thể được mở rộng để phục vụ nhiều khu vực và đối tượng người dùng hơn, không chỉ giới hạn ở các thành phố lớn mà còn ở các khu vực nông thôn và ngoại ô.
* **Cải thiện công nghệ**: Tiếp tục nghiên cứu và áp dụng các công nghệ mới như AI, Machine Learning và Big Data để nâng cao hiệu suất và khả năng đáp ứng của hệ thống.
* **Tăng cường bảo mật**: Đảm bảo an toàn thông tin và bảo mật cho người dùng, đặc biệt là trong các giao dịch tài chính và quản lý dữ liệu cá nhân.

**TÀI LIỆU THAM KHẢO**

1. CRM Việt: <https://crmviet.vn/>
2. Spring Framework: <https://docs.spring.io/spring/docs/4.3.12.RELEASE/spring-> [framework-reference/htmlsingle/](https://docs.spring.io/spring/docs/4.3.12.RELEASE/spring-framework-reference/htmlsingle/)
3. Spring REST: <http://www.allitebooks.com/spring-rest/>
4. AdvancedSpringData JPA: https://spring.io/blog/2011/04/26/advanced- springdata-jpa-specifications-and-querydsl/
5. ReactJs: [https://reactjs.org](https://reactjs.org/)
6. Javascript: <https://developer.mozilla.org/vi/docs/Web/JavaScript>