TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Đặng Huỳnh Trọng Tín - 51702196**

**Phạm Thành Đạt - 51702221**

**TÊN ĐỀ TÀI**

XÂY DỰNG WEBSITE BẤT ĐỘNG SẢN SỬ DỤNG SPRING BOOT VÀ MICROSERVICE

**DỰ ÁN CÔNG NGHỆ THÔNG TIN 2**

**TÊN CHUYÊN NGÀNH**

Kỹ thuật phần mềm

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Đặng Huỳnh Trọng Tín - 51702196**

**Phạm Thành Đạt - 51702221**

**TÊN ĐỀ TÀI**

XÂY DỰNG WEBSITE BẤT ĐỘNG SẢN SỬ DỤNG SPRING BOOT VÀ MICROSERVICE

**DỰ ÁN CÔNG NGHỆ THÔNG TIN 2**

**TÊN CHUYÊN NGÀNH**

Kỹ thuật phần mềm

Người hướng dẫn

**ThS. Võ Văn Thành**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

**LỜI CẢM ƠN**

Chúng tôi xin chân thành cảm ơn …………………………………… …………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

*TP. Hồ Chí Minh, ngày ... tháng … năm 20..*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

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

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

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

*TP. Hồ Chí Minh, ngày … tháng … năm 20..*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

**TÊN ĐỀ TÀI**

**TÓM TẮT**

(Time New Romans – 13)

**TITLE**

**ABSTRACT**

(Time New Romans – 13)

**MỤC LỤC**

[DANH MỤC HÌNH VẼ x](#_Toc143692291)

[DANH MỤC BẢNG BIỂU xi](#_Toc143692292)

[DANH MỤC CÁC CHỮ VIẾT TẮT xii](#_Toc143692293)

[CHƯƠNG 1. MỞ ĐẦU 1](#_Toc143692294)

[1.1 Lý do chọn đề tài 1](#_Toc143692295)

[1.2 Mục tiêu thực hiện đề tài 2](#_Toc143692296)

[1.3 Đối tượng và phạm vi nghiên cứu 2](#_Toc143692297)

[1.3.1 Đối tượng nghiên cứu 2](#_Toc143692298)

[1.3.2 Phạm vi chức năng 2](#_Toc143692299)

[1.3.3 Phạm vi dữ liệu 3](#_Toc143692300)

[1.4 Phương pháp nghiên cứu 3](#_Toc143692301)

[1.5 Ý nghĩa nghiên cứu 3](#_Toc143692302)

[CHƯƠNG 2. TỔNG QUAN 4](#_Toc143692303)

[2.1 Khảo sát thực trạng các trang website bất động sản hiện nay 4](#_Toc143692304)

[2.2 Giải pháp 4](#_Toc143692305)

[2.3 Quy trình lập kế hoạch thực hiện dự án 4](#_Toc143692306)

[2.4 Cấu trúc kế hoạch thực hiện dự án 5](#_Toc143692307)

[2.5 Công nghệ thực hiện 5](#_Toc143692308)

[2.6 Chức năng thực hiện 5](#_Toc143692309)

[2.7 Yêu cầu hệ thống 6](#_Toc143692310)

[2.8 Kết quả của hệ thống 6](#_Toc143692311)

[CHƯƠNG 3. CƠ SỞ LÝ THUYẾT 6](#_Toc143692312)

[3.1 Front-end 6](#_Toc143692313)

[3.1.1 Reatjs 7](#_Toc143692314)

[3.2 Back-end 8](#_Toc143692315)

[3.2.1 Spring-boot 8](#_Toc143692316)

[3.2.2 Kiến trúc Microservice 8](#_Toc143692317)

[3.3 Cơ sở dữ liệu 9](#_Toc143692318)

[3.3.1 MySQL 9](#_Toc143692319)

[3.3.2 MongoDB 10](#_Toc143692320)

[CHƯƠNG 4. THIẾT KẾ HỆ THỐNG 10](#_Toc143692321)

[4.1 Tổng quan về hệ thống 10](#_Toc143692322)

[4.2 Sơ đồ chức năng nghiệp vụ 10](#_Toc143692323)

[4.3 Sơ đồ Usecase 11](#_Toc143692324)

[4.3.1 Xác định các Usecase 11](#_Toc143692325)

[4.3.2 Biểu đồ Usecase tổng quát 12](#_Toc143692326)

[4.4 Đặc tả các Usecase 12](#_Toc143692327)

[4.4.1 Usecase đăng nhập 12](#_Toc143692328)

[4.4.2 Usecase đăng ký 15](#_Toc143692329)

[4.4.3 Usecase tạo tin đăng 18](#_Toc143692330)

[4.4.4 Usecase cập nhật tin đăng 21](#_Toc143692331)

[4.4.5 Usecase xóa tin đăng 24](#_Toc143692332)

[4.4.6 Usecase xem tin đăng 27](#_Toc143692333)

[4.4.7 Usecase cập nhật thông tin tài khoản 30](#_Toc143692334)

[4.4.8 Usecase xóa tài khoản 33](#_Toc143692335)

[4.4.9 Usecase quản lý các tin đăng 36](#_Toc143692336)

[4.5 Thiết kế cơ sở dữ liệu 36](#_Toc143692337)

[4.5.1 Bảng Listing 36](#_Toc143692338)

[4.5.2 Bảng Listingcategories 37](#_Toc143692339)

[4.5.3 Bảng Listingtypes 38](#_Toc143692340)

[4.5.4 Bảng Account 38](#_Toc143692341)

[4.5.5 Bảng Roles 39](#_Toc143692342)

[4.5.6 Bảng Account\_roles 39](#_Toc143692343)

[4.5.7 Bảng Payment 39](#_Toc143692344)

[4.5.8 Bảng PaymentHistory 40](#_Toc143692345)

[4.5.9 Bảng Bank 40](#_Toc143692346)

[4.5.10 Bảng image 40](#_Toc143692347)

[CHƯƠNG 5. THỰC NGHIỆM 41](#_Toc143692348)

[5.1 Dữ liệu thực nghiệm 41](#_Toc143692349)

[5.2 Cài đặt thực nghiệm 41](#_Toc143692350)

[CHƯƠNG 6. KẾT LUẬN 42](#_Toc143692351)

[6.1 Kết luận 42](#_Toc143692352)

[6.2 Hướng phát triển 42](#_Toc143692353)

[TÀI LIỆU THAM KHẢO 43](#_Toc143692354)

# DANH MỤC HÌNH VẼ

[Hình 2.1: Scaled Dot-Product Attention 3](#_Toc142677545)

# DANH MỤC BẢNG BIỂU

[Bảng 4.1: Thống kê kiểu thực thể trong tập VLSP 2016 4](#_Toc142677565)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| BERT | Bidirectional Encoder Representations from Transformers |
| GEC | Grammatical Error Correction |
| MLM | Masked Language Model |
| NLP | Natural Language Processing |
| NSP | Next Sentence Prediction |

# MỞ ĐẦU

## Lý do chọn đề tài

Ngày nay, trong thời đại số, với sự phát triển của công nghệ, đặc biệt là công nghệ thông tin đã tạo ra rất nhiều những cơ hội được tiếp cận với những dịch vụ tiện ích cho con người, cùng với đó là một nền công nghiệp tri thức, thương mai điện tử.

Sự bùng nổ của Internet đã tạo ra một bước ngoặc về phát triển thương mại điện tử không chỉ ở các nước trên thế giới mà là ở cả Việt Nam. Chúng tạo ra vô vàn cơ hội cũng như thử thách mang lại cho các doanh nghiệp, đơn vị kinh doanh. Nói cách khác thương mại điện tử giúp thu hẹp khoảng cách giữa các doanh nghiệp.

Nhắc tới thương mại điện tử là người ta nghĩ ngay tới Web site, nghĩ tới kết nối mọi người lại với nhau ở bất kỳ nơi nào trên thế giới.Chứng tỏ Web site có vị trí vô cùng quan trọng trong việc phát triển thương mại điện tử.

Một vấn đề đang tồn tại đang tồn tại ở Việt Nam là vấn đề dân số. Đặc biệt là các thành phố lớn như Thành phố Hồ Chí Minh. Đây là một thành phố đông dân với mật độ dân số dày đặc nhất cả nước, nhu cầu nhà đất cũng vì thế mà trở nên phức tạp. Trên thực tế, rất nhiều người sinh sống ở Hồ Chí Minh là người dân ngoại tỉnh, ngoại quốc, họ có nhu cầu tìm thuê nhà, phòng trọ, chung cư, nhà nguyên căn để sinh hoạt và làm việc nhưng lại gặp khó khăn bởi sự thiếu hụt và nhiễu loạn thông tin về nhà đất. Bên cạnh đó những người chủ bất động sản có nhu cầu cho thuê cũng phải rất vất vả để quảng cáo tới khách hàng, nhiều người vẫn còn sử dụng biện pháp thủ công đó là dán tờ rơi lên bờ tường, cột điện,..rất lãng phí và gây mất mỹ quan đô thị. Trước thực tế đó, cấp thiết phải thiết kế một hệ thống tìm kiếm và cho thuê bất động sản trực tuyến, với tiêu chí thân thiện, dễ tiếp cận nhưng cũng đầy đủ các chức năng cần thiết để đáp ứng nhu cầu nhà đất đang ngày một tăng cao.

Quan trọng hơn, khách có nhu cầu giao dịch có thể liên hệ trực tiếp với người bán, không phải thông qua trung gian. Như vậy, người mua sẽ tránh được phải mất phí cho “cò” nhà đất.

Ý tưởng xây dựng website sàn giao dịch bất động sản ra đời với mong muốn trở thành sàn giao dịch bất động sản có thể đáp ứng được hoàn toàn các tiêu chí đó.

## Mục tiêu thực hiện đề tài

Thiết kế một website không chỉ thực hiện tốt các dịch vụ kinh doanh bất động sản, khai thác thị trường tiềm năng, mà sẵn sàng tư vấn cho khách hàng về quy hoạch, thủ tục pháp lý, chính sách tài chính trong lĩnh vực bất động sản. Trong đó :

- Website chứa đầy đủ hình ảnh và thông tin về các loại giao dịch bất động sản.

- Website có đầy đủ các chức năng cần thiết nhằm giới thiệu sản phẩm, tư vấn khách hàng, giao dịch mua bán giữa các nhà đầu tư.

- Các tác vụ phải linh hoạt và thuận tiện cho khách hàng: bố trí site để các chức năng được sử dụng một cách dễ dàng.

- Đảm bảo truy cập nhanh chóng, an toàn và chính xác.

- Hệ thống admin phải dễ dàng cho người quản trị: cập nhật , thêm, sửa, xoá thông tin.

## Đối tượng và phạm vi nghiên cứu

### Đối tượng nghiên cứu

Website đăng tin bất động sản nhằm quảng cáo và quản lý các bất động sản, làm trung gian, cầu nối giữa bên bán/cho thuê và bên mua/bên thuê thực hiện giao dịch giữa 2 bên.

### Phạm vi chức năng

* Website xây dựng với nhiều chức năng hợp lý, dễ dàng sử dụng.
* Dễ nâng cấp, bảo trì.

- Giới thiệu các giao dịch mua bán ,cho thuê, thông tin về từng bất động sản, giá thành.

- Xây dựng các chức năng tìm kiếm, thống kê.

- Chức năng cho phép khách hàng thực hiện giao dịch liên hệ trực tuyến giữa bên mua và bên bán.

- Chức năng liên hệ với người quản trị , tư vấn khách hàng.

### Phạm vi dữ liệu

Nguồn dữ liệu dùng cho hệ thống có thể được cập nhật từ nhiều nguồn khác

nhau, tuy nhiên trong điều kiện có thể và phạm vi của đề tài thì phạm vi dữ liệu của

website Giao dịch bất động sản được chủ yếu lấy từ các nguồn sau:

- Tin đăng từ các nhà giao dịch bất động sản.

- Tin đăng từ ban quản trị hệ thống.

- Thông tin từ các website khác.

## Phương pháp nghiên cứu

- Tiến hành thu thập và phân tích những thông tin, tài liệu liên quan đến đề tài để hình thành nên những ý tưởng tổng quan.

- Khảo sát thị trường bất động sản.

- Xác định các yêu cầu nhằm phân tích thiết kế hệ thống chương trình cho phù hợp.

- Triển khai chương trình và đánh giá kết quả

## Ý nghĩa nghiên cứu

* Giúp nhóm hiểu rỏ quy trình mua bán bất động sản.
* Giúp cho khách hàng có thể tìm kiếm bất động sản một cách nhanh chóng và dễ dàng.

# TỔNG QUAN

## Khảo sát thực trạng các trang website bất động sản hiện nay

Trong bối cảnh hiện nay, xu hướng chuyển đổi số trong nhiều lĩnh vực và sự thay đổi trong hành vi tìm kiếm thông tin của khách hàng đã thúc đẩy nhiều website bất động sản xuất hiện nhưng còn dàn trải và tự phát, không dựa trên những quy chuẩn về chất lượng website hay hiểu rõ về tâm lý của người sử dụng nên chưa thể phát huy được tối đa các lợi thế mà website bất động sản mang lại

## Giải pháp

Với xu thế thông tin toàn cầu, phần mềm quản lý đang là mô hình phát triển chung cho các hệ thống hiện nay. Dự án xây dựng phần mềm quản lý bất động sản online được xây dựng và phát triển nhằm mục đích tìm kiếm các bất động sản một cách nhanh nhất và dễ dàng nhất.

Đây là một website chứa một lượng lớn thông tin về các bất động sản mà hiện tại chủ nhà không có nhu cầu sử dụng và muốn cho thuê. Người truy cập vào website có thể sử dụng chức năng tìm kiếm theo khu vực như ở thành phố, quân huyện, theo giá tiền thuê cũng như theo diện tích, số phòng mong muốn…

Website là cầu nối giữa người bán/người cho thuê và người mua/người thuê thực hiện các chức năng quảng bá, giao dịch giữa 2 bên đem lại những giá trị thiết thực cho cả đơn vị chủ quản và các khách hàng

## Quy trình lập kế hoạch thực hiện dự án

- Thiết lập các ràng buộc dự án: thời gian, nhân lực.

- Đánh giá bước đầu về các “tham số” của dự án: quy mô, độ phức tạp, nguồn lực.

- Xác định các mốc thời gian thực hiện dự án và sản phẩm thu được ứng với mỗi mốc thời gian.

- Trong khi dự án chưa hoàn thành hoặc chưa bị hủy bỏ thì thực hiện theo kế hoạch sau:

1. Lập lịch thực hiện dự án trên Jira.

2. Thực hiện các hoạch động theo lịch trình.

3. Theo dõi sự tiến triển của dự án và so sánh với lịch trình.

4. Lên lịch thực hiện cho các chức năng mới.

## Cấu trúc kế hoạch thực hiện dự án

- Tổ chức dự án.

- Phân thích các rủi ro.

- Yêu cầu về tài nguyên phần cứng, phần mềm.

- Phân công công việc.

- Lập lịch dự án.

- Kiểm soát tiến độ và báo cáo.

## Chức năng thực hiện

Hệ thống sẽ có 2 phần User và Admin bao gồm các chức năng cơ bản như:

* **Quyền User:**
* Chức năng thêm, xóa, sửa xem các bài đăng, tin đăng về nhà đất.
* Chức năng ẩn tin đã đăng.
* Chức năng xem các gói tài chính.
* Chức năng cập nhật thông tin cá nhân.
* Chức năng thay đổi mật khẩu.
* Chức năng tìm kiếm tin đăng.
* **Quyền Admin:**
* Chức năng thống kê doanh thu
* Chức năng quản lý tất cả các tin đăng.
* Chức năng quản lý tất cả các tài khoản người dung.
* Chức năng cập nhật thông tin cá nhân.
* Chức năng thay đổi mật khẩu.
* Chức năng phê duyệt tin đăng.

### Yêu cầu hệ thống

#### Về giao diện

* Giao diện người dùng
  + Phải có một giao diện thân thiện dễ sử dụng.
  + Bố cục trang web phải rõ ràng, hợp lý.
  + Có form đăng nhập/đăng ký để trở thành thành viên.
  + Giới thiệu được tin đăng của web site đang có.
  + Các tin đăng ưu tiên phải hiển thị lên đầu tiên và được làm nổi bật bởi các banner.
  + Chức năng đăng tin phải được nổi bật, đặt ở vị trí thuận lợi dễ nhìn thấy.
  + Mỗi loại tin đăng cần hiển thị vừa đủ các thông tin quan trọng như tiêu đề, địa chỉ, giá. Phải có trang xem chi tiết tin đăng.
  + Trang chi tiết phải hiển thị được đầy đủ thông tin, chữ rõ ràng.
  + Hiển thị đầy đủ các danh mục tìm kiếm.
* Giao diện người quản trị.
* Đơn giản, dễ quản lý.
* Danh sách tài khoản và tin đăng hiển thị rõ ràng từng dòng và có các chức năng xóa, chỉnh sửa.
* Hiển thị doanh thu theo dạng biểu đồ.

#### Về nghiệp vụ

* Người xây dựng nên hệ thống phải am hiểu về thương mại điện tử, cụ thể là trang web bất động sản để xây dựng một trang web bất động sản tốt nhất đáp ứng nhu cầu thị trường.
* Người xây dựng nên hệ thống đặc biệt phải hiểu rõ các thông tin cần thiết cho một bài đăng: Có thể đưa ra đựơc một cái nhìn tổng quát, sâu rộng về bài đăng như giá cả, diện tích, vị trí.

#### Về bảo mật

* Hệ thống phân quyền sử dụng. Người Admin có toàn quyền giữ bảo mật cho web site bằng mật khẩu riêng. Các đối tượng khác được phân quyền theo các mức truy cập khác nhau.

#### Về bảo trì

* Thiết kế và xây dựng hệ thống đáp ứng đầy đủ các chức năng bằng cách áp dụng các tiêu chuẩn coding dễ bảo trì và nâng cấp sau này.

### Lựa chọn giải pháp và công nghệ

#### Front-end

* Sử dụng Framework Reatjs để thiết kế giao diện đẹp mắt và dễ dàng đồng thời tối ưu tốc độ tải trang.
* Sử dụng bootstrap để có thể thiết kế giao diện trang web nhanh hơn.

#### Back-end

* Do định hướng sẽ xây dựng một website bất động sản lớn và sẽ có nhiều tính năng nên sẽ áp dụng kiến trúc Microservice vào đề tài này cùng với ngôn ngữ lập trình Java Spring boot.

#### Cơ sở dữ liệu

* MySQL: Để lưu trữ tin đăng lớn và các tài nguyên khác của hệ thống ngoài ra nhờ tính bảo mật của cơ sở dữ liệu này đảm bảo độ an toàn cho dữ liệu của hệ thống.
* MongoDB: Tốc độ truy xuất nhanh nên thích hợp dùng để lưu trữ lượng hình ảnh khổng lồ của hệ thống.

## Kết quả mong muốn đạt được

- Hệ thống sẽ cung cấp dịch vụ trung gian giữa người bán nhà, thuê nhà và người cần mua,thuê nhà. Mỗi ngày sẽ có hơn 5000 bài đăng mới bán nhà, bán đất, căn hộ , dự án chung cư, dự án biệt thự, đất nền v.v…. Đa dạng về địa điểm, diện tích và mức giá. Do đó, để giúp bạn tìm kiếm căn nhà phù hợp thật nhanh chóng và thuận tiện, web site của chúng tôi đáp các yêu cầu trên.

- Ngoài ra, web site còn mang đến cho bạn những thông tin hữu ích từ các chuyên gia bất động sản, tài chính và pháp lý v.v... Từ đó, làm cho việc mua bán của bạn trở nên dễ dàng hơn.

# CƠ SỞ LÝ THUYẾT

## Front-end

### Reatjs

React là một thư viện JavaScript với mã nguồn mở hiệu quả, dễ khai báo và linh hoạt để xây dựng giao diện người dùng đơn giản, nhanh chóng và có thể mở rộng của các ứng dụng web. Reactjs được tạo ra bởi một nhân viên của Facebook là Jordan Walke với mục đích là cải thiện quá trình phát triển UI, giúp người lập trình UI có thể thực hiện một cách nhanh chóng và ít xảy ra lỗi hơn so với trước đây khi sử dụng Javascript thuần để phát triển UI. Để giải quyết 2 vấn đề trên Jordan Walke đã đưa ra 2 khái niệm quang trọng là JSX và Virtual DOM nhằm mang đến khả năng tái sử dụng code (Reusable Code).

#### JSX

Trước khi nói về JSX chúng ta sẽ nói một chút để hiểu về cấu tạo cơ của một website đó là nhứng HTML documents - thứ mà trình duyệt sẽ đọc để hiển thị nội dung trên các thiết bị của bạn.

A diagram of a document

Description automatically generated

Trình duyệt sẽ đọc những HTML documents và sinh ra Document Object Model – đó là một cấu trúc dạng tree giúp định hình cho việc hiển thị website như thế nào. Và bằng cách sử dụng Javascript, lập trình viên có thể thay đổi cấu trúc của cây DOM – tạo tính linh động (dynamic) cho website.

JSX hay còn gọi là Javascript extension là một React extension giúp chúng ta viết các khối dạng HTML bằng javascript và đặt chúng trong DOM một cách để dàng mà không cần bất kỳ phương thức nào như createElement hay appendChild().

#### Visual DOM

Virtual DOM là một DOM ảo do Reactjs tạo nên nó là bản copy của DOM thật trên trang website đó. Reactjs sẽ sử dụng Virtual DOM để tìm kiếm và cập nhật DOM đúng phần DOM thật ở trang web mà không phải load lại toàn bộ cây DOM. Điều này giúp cải thiện performance của website lên rất nhiều và đặc biết có lợi đối với các trang web lớn cần tối ưu hiệu năng.

### Ưu điểm của Reactjs

* **Tốc độ:** Về cơ bản, React cho phép các nhà phát triển sử dụng các phần riêng lẻ của ứng dụng của họ ở cả phía máy khách và phía máy chủ, điều này cuối cùng giúp tăng tốc độ của quá trình phát triển. Nói một cách dễ hiểu, các nhà phát triển khác nhau có thể viết các phần riêng lẻ và tất cả các thay đổi được thực hiện sẽ không gây ra tính logic của ứng dụng.
* **Linh hoạt:** So với các frontend framework khác, mã React dễ bảo trì hơn và linh hoạt hơn do cấu trúc mô-đun của nó. Do đó, sự linh hoạt này giúp tiết kiệm rất nhiều thời gian và chi phí cho cho việc phát triển.
* **Hiệu suất:** React JS được thiết kế để cung cấp hiệu suất cao. Cốt lõi của khung cung cấp chương trình DOM ảo và kết xuất phía máy chủ, giúp các ứng dụng phức tạp chạy cực nhanh.
* **Nhiều thư viện hỗ trợ:** Do có nhiều thư viện hỗ trợ cho Reactjs nên sẽ giảm việc code nhiều mà vẫn tạo ra được một giao diện ưng ý.

## Back-end

### Spring-boot

Spring boot là một framework mã nguồn mở nằm trong hệ sinh thái Spring Framework. Có thể nó coi nó như một công cụ làm cho quá trình phát triển một ứng dụng web hay một ứng dụng microservice trở nên dễ dàng và nhanh chóng hơn với những khả năng mà nó mang lại:

* Giảm thiểu các bước cấu hình phức tạp ở bằng XML của Spring Framework.
* Dễ hiểu và dễ phát triển một ứng dụng Spring.
* Giảm thời gian phát triển cho lập trình viên.
* Tương tác và hỗ trợ tốt cho việc phát triển một hệ thống microservice.

### Kiến trúc Microservice

Kiến trúc microservice hay còn hiểu là kiến trúc của một hệ thống phân tán. Trên thực tế không có một định nghĩa cụ thể nào cho một mô hình microservice, ngược lại, mô hình microservice có rất nhiều biến thể phụ thuộc vào tính chất của sản phẩm, dịch vụ hay mô hình kinh doanh. Microservice có thể hiểu một cách đại khái là một kỹ thuật phát triển phần mềm, một biến thể của thuộc kiến trúc hướng dịch vụ (SOA), được dùng để xây dựng một hệ thống hay ứng dụng có tính chất phân tán, độc lập về dữ liệu , logic xử lý và cả mã nguồn.

A diagram of a software process

Description automatically generated

Vì đặc tính của hệ thống phân tán nên microservice sẽ mang lại nhiều lợi ích như:

+ Dễ quảng lý các service.

+ Dễ vận hành tổ chức (do mỗi team sẽ đảm nhiệm 1 service, mỗi service sẽ đại diện cho một chức năng).

+ Dễ thay thế và mở rộng.

+ Khả năng chịu lỗi cao (1 service bị lỗi các service khác vẫn hoạt động bình thường).

+ Đem lại khả năng dự phòng rui rỏ cho hệ thống (phiên bản mới của service bị lỗi có thể dễ dàng thay thế bằng các service cũ dự phòng).

### Rabbit MQ

## Cơ sở dữ liệu

### MySQL

MySQL là một hệ thống quảng trị cơ sở dữ liệu mã nguồn mở (RDBMS – Relational Database Management System) hoạt động theo mô hình client-server và được tích hợp sẳn apache, PHP. Là một hệ quảng trị cơ sở dữ liệu phổ biến giúp quảng trị các dữ liệu có cấu trúc (SQL – Structured Query Language) trong cơ sở dữ liệu (DB - Database) là nơi lưu trữ, truy xuất dữ liệu có cấu trúc dạng table-based.

### MongoDB

MongoDB là một dạng Document Database (database hướng tài liệu) , một kiểu NoSQL Database. Khác với RDBMS, Mongo DB lưu dữ liệu dạng JSON nên có thể linh hoạt lưu trữ dữ liệu theo các kích cỡ khác nhau. Tương tự như định nghĩa table, row, column ở RDBMS ở Mongo DB cũng có các collection, document, field nhưng với tính “Nosql” dữ liệu ở MongoDB không bị ràng buộc bởi các khóa chính, khóa ngoại... và một khuôn khổ nhất định giúp tiết kiệm thời gian cho việc kiểm tra sự thỏa mãn về cấu trúc. Ngoài ra MongoDB còn hỗ trợ cached dữ liệu lên RAM để lượt truy vấn sau diễn ra nhanh hơn mà không cần đọc từ ổ cứng đem đến tốc độ truy vấn nhanh hơn nhiều so với hệ quảng trị cơ sở dữ liệu quang hệ RDBMS.

# THIẾT KẾ HỆ THỐNG

## Cấu trúc của hệ thống

Hệ thống Website Bất động sản sẽ áp dụng các kiến trúc Microservice, chia các chức năng thành các service riêng hoạt động độc lập. Bao gồm các service chính sau:

* *Auth service:* có chức năng liên quan đến xử lý đăng nhập, đăng xuất, phân quyền.
* *Listing service:* có chức năng liên quan đến đăng tin, quản lý tin đăng của hệ thống.
* *Image service:* có chức năng liên quan đến quản lý hình ảnh của hệ thống
* *Payment service:* có chức năng liên quan đến xử lý thanh toán.
* *Finance service:* có chức năng liên quan đến xử lý các gói vay ngân hàng.
* *Crawl service:* Dùng để crawl dữ liệu thực tế.

Mỗi service sẽ có cơ sở dữ liệu riêng phục vụ cho việc lưu trữ và truy xuất theo từng chức năng của service.

*Auth service:*

* Bảng Roles: Dùng để lưu các quyền đăng nhập có trong hệ thống
* Bảng Account: Dùng để lưu thông tin tài khoản của hệ thống
* Bảng Account\_roles: Dùng để lưu các phân quyền của các tài khoản có trong hệ thống

*Listing service:*

* Bảng Listing: Dùng để lưu thông tin các tin đăng của thành viên trong hệ thống.
* Bảng Listingcategories: Dùng để lưu thông tin các loại nhà đất có trong hệ thống.
* Bảng Listingtype: Dùng để lưu thông tin các tin đăng có trong hệ thống.

*Image service:*

* Bảng Image: Dùng để lưu hình ảnh của các tin đăng trong hệ thống

*Payment service:*

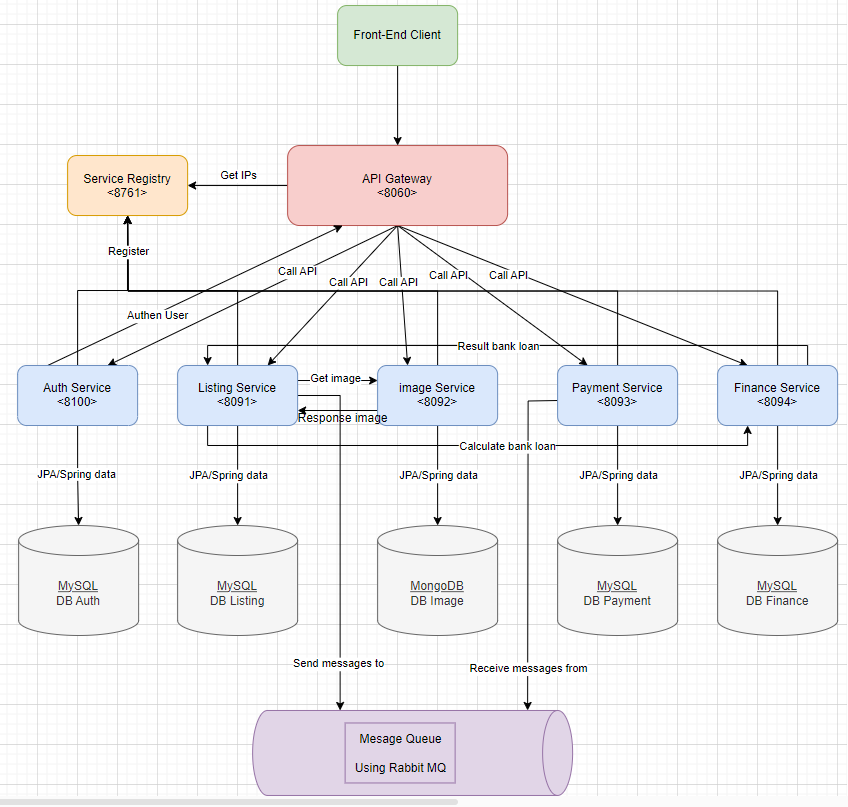
* Bảng Payment: Dùng để lưu thanh toán của người dùng
* Bảng PaymentHistory: Dùng để lưu lịch sử giao dịch

*Finance service:*

* Bảng Bank: Dùng để lưu thông tin khoản vay của các ngân hàng

Ngoài ra, hệ thống sẽ có một API Gateway để quản lý các domain của service. API gateway sẽ nhận các request từ client và điều hướng tới service tương ứng với chức năng mà người dùng yêu cầu.

## Sơ đồ cấu trúc hệ thống



## Sơ đồ chức năng nghiệp vụ

Với yêu cầu nghiệp vụ và thông tin đặt ra đối với hệ thống ta xây dựng được sơ đồ chức năng của hệ thống.

A diagram of a company

Description automatically generated

## Sơ đồ Usecase

### Xác định các Usecase

Các tác nhân của hệ thống bao gồm:

* **Admin:** Là người quản trị của hệ thống, có các quyền và chức năng như: quản lý bài đăng, quản trị thành viên, phê duyệt tin đăng....
* **Member:** Là thành viên của hệ thống có chức năng: Đăng kí, đăng nhập, đăng xuất, tìm kiếm, xem, sửa thông tin cá nhân, xem thông tin chi tiết về bất động sản, đăng tin bất động sản, quản lý tin đăng của mình.
* **Customer:** Là người dùng vãng lai có chức năng: Đăng kí, tìm kiếm, xem thông tin bài đăng, xem chi tiết bài đăng.

### Biểu đồ Usecase tổng quát

A diagram of a person with text

Description automatically generated with medium confidence

## Đặc tả các Usecase

### Usecase đăng nhập

#### Đặc tả luồng sự kiện đăng nhập

|  |  |
| --- | --- |
| **Use case** | Đăng nhập |
| **Tác nhân** | Admin, Member |
| **Mục đích** | Đăng nhập hệ thống |
| **Mô tả chung** | Người dùng khi muốn đăng nhập vào hệ thống |
| **Luồng sự kiện chính** | 1. Người dùng chọn chức năng đăng nhập.* |
| 2. Hệ thống hiển thị giao diện đăng nhập. |
| 3. Người dùng nhập thông tin đăng nhập. |
| 4. Gửi thông tin đã nhập đến hệ thống |
| 5. Kiểm tra thông tin đăng nhập, nếu đúng thì cho phép truy cập, sai thì thông báo yêu cầu nhập lại |
| **Luồng thay thế** | Không có |
| **Các yêu cầu cụ thể** | Tên đăng nhập và mật khẩu khớp với thông tin đã đăng ký. |
| **Điều kiện trước** | Nguời dùng đã đăng ký thành viên trong hệ thống |
| **Điều kiện sau** | Thông báo đăng nhập thành công/ thất bại |
| **Các yêu cầu mở rộng** | Use case đăng xuất xuất hiện sau khi đăng nhập.  Hiển thị các các chức năng tương ứng quyền truy cập người dùng. |

#### Biểu đồ use case đăng nhập

A diagram of a diagram

Description automatically generated

#### Biểu đồ tuần tự đăng nhập

A diagram of a computer

Description automatically generated

#### Biểu hoạt động đăng nhập

A diagram of a computer

Description automatically generated

### Usecase đăng ký

#### Đặc tả luồng sự kiện đăng ký

|  |  |
| --- | --- |
| **Use case** | Đăng kí |
| **Tác nhân** | Customer |
| **Mục đích** | Đăng ký tài khoản để quản lý người dùng |
| **Mô tả chung** | Người dùng có nhu cầu đăng kí tải khoản trên hệ thống để sử dụng nhiều chức năng hơn |
| **Luồng sự kiện chính** | 1. Người dùng chọn chức năng đăng ký. |
| 2. Hệ thống hiển thị giao diện đăng ký. |
| 3. Người dùng nhập thông tin đăng ký và gửi đến hệ thống. |
| 4. Hệ thống sẽ kiểm tra thông tin người dùng đã nhập. |
| **Luồng thay thế** | 4.1. Kiểm tra username đã tồn tại trong hệ thống chưa. Nếu tồn tại thông báo lỗi. |
| 4.2. Kiểm tra email đã tồn tại trong hệ thống chưa. Nếu tồn tại thông báo lỗi. |
| 4.3. Kiểm tra password có đủ ký tự yêu cầu. Nếu mật khẩu không thỏa yêu cầu thông báo lỗi. |
| **Các yêu cầu cụ thể** | Định dạng email phải đúng, email phải tồn tại và chính xác. |
| **Điều kiện trước** | Nguời dùng lần đầu đăng ký tài khoản |
| **Điều kiện sau** | Thông báo đăng ký thành công/ thất bại |
| **Các yêu cầu mở rộng** | Không có. |

#### Biểu đồ use case đăng ký

A diagram of a diagram

Description automatically generated

#### Biểu đồ tuần tự đăng ký

A diagram of a diagram

Description automatically generated

#### Biểu hoạt động đăng ký

A diagram of a diagram

Description automatically generated

### Usecase tạo tin đăng

#### Đặc tả luồng sự kiện tạo tin đăng

|  |  |
| --- | --- |
| **Use case** | Tạo tin đăng |
| **Tác nhân** | Member |
| **Mục đích** | Người dùng cập nhật thông tin bài đăng |
| **Luồng sự kiện chính** | 1. Người dùng chọn đăng tin |
| 2. Hệ thống hiển thị form đăng tin |
| 3. Người dùng nhập thông tin và gửi |
| 4. Hệ thống hệ thống kiểm tra thông tin người dùng nhập |
| 5. Hệ thống lưu tin đăng |
| **Luồng thay thế** | Không có |
| **Các yêu cầu cụ thể** | Người dùng phải nhập đầy đủ thông tin |
| **Điều kiện trước** | Người dùng đã đăng nhập vào hệ thống. |
|
| **Điều kiện sau** | Thông báo thành công |
| **Các yêu cầu mở rộng** | * Có thể chỉnh sửa tin đăng * Có thể xóa tin đăng |

#### Biểu đồ use case tạo tin đăng

A diagram of a person

Description automatically generated

#### Biểu đồ tuần tự tạo tin đăng

A diagram of a diagram

Description automatically generated

#### Biểu hoạt động tạo tin đăng

A diagram of a diagram

Description automatically generated

### Usecase cập nhật tin đăng

#### Đặc tả luồng sự cập nhật tin đăng

|  |  |
| --- | --- |
| **Use case** | Cập nhật tin đăng |
| **Tác nhân** | Admin, Member |
| **Mục đích** | Người dùng cập nhật thông tin bài đăng |
| **Luồng sự kiện chính** | 1. Người dùng quản lý tin đăng |
| 2. Hệ thống danh sách tin đăng |
| 3. Chọn chức năng chỉnh sửa |
| 4. Hệ thống hiển thị ra trang chỉnh sửa |
| 5. Người dùng nhập thông tin cần chỉnh sửa và xác nhận |
| 6. Hệ thống kiểm tra và lưu tin đăng |
| **Luồng thay thế** | Không có |
| **Các yêu cầu cụ thể** | Không có |
| **Điều kiện trước** | Người dùng đã đăng nhập vào hệ thống. |
|
| **Điều kiện sau** | Thông báo cập nhật thành công |
| **Các yêu cầu mở rộng** | Không có |

#### Biểu đồ use case cập nhật tin đăng

A diagram with text and words

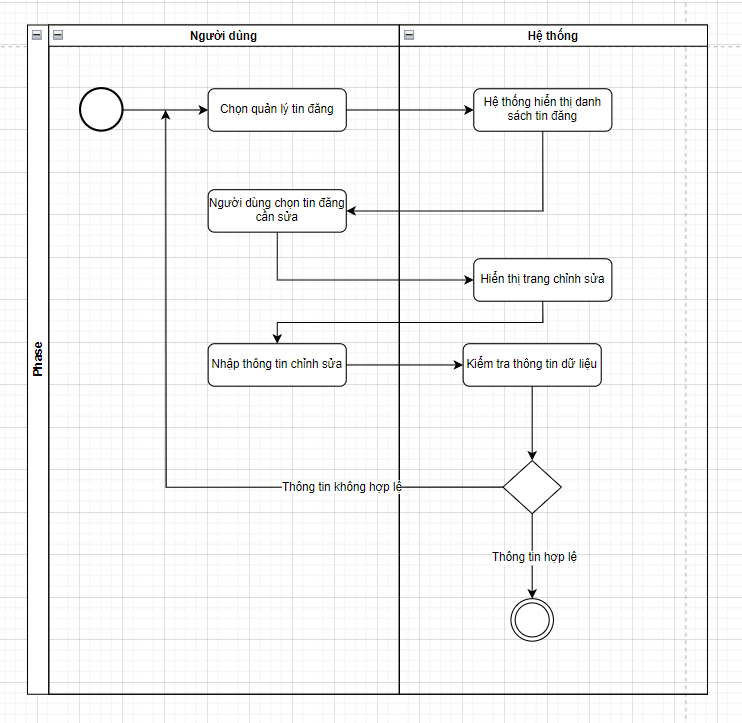
Description automatically generated

#### Biểu đồ tuần tự cập nhật tin đăng

A diagram of a diagram

Description automatically generated

#### Biểu hoạt động cập nhật tin đăng



### Usecase xóa tin đăng

#### Đặc tả luồng xóa tin đăng

|  |  |
| --- | --- |
| **Use case** | Xóa tin đăng |
| **Tác nhân** | Admin, Member |
| **Mục đích** | Người dùng muốn xóa tài khoản đã đăng ký |
| **Luồng sự kiện chính** | 1. Người dùng chọn tài khoản cần xóa |
| 2. Hệ thống xóa tài khoản ra khỏi database |
| **Luồng thay thế** | Không có |
| **Các yêu cầu cụ thể** | Không có |
| **Điều kiện trước** | Người dùng đã đăng nhập vào hệ thống. |
|
| **Điều kiện sau** | Thông báo xóa thành công |
| **Các yêu cầu mở rộng** | Không có |

#### Biểu đồ use case xóa tin đăng

A diagram of a diagram

Description automatically generated

#### Biểu đồ tuần tự xóa tin đăng

A diagram of a diagram

Description automatically generated

#### Biểu hoạt động xóa tin đăng

A diagram of a diagram

Description automatically generated

### Usecase xem tin đăng

#### Đặc tả luồng xem tin đăng

|  |  |
| --- | --- |
| **Use case** | Xem chi tiết tin đăng |
| **Tác nhân** | Admin, Member, Customer |
| **Mục đích** | Người dùng muốn xem chi tiết tin đăng |
| **Luồng sự kiện chính** | 1. Người dùng chọn thông tin cá nhân cần cần xem |
| 2. Hệ thống hiển thị chi tiết tin đăng |
| **Luồng thay thế** | Không có |
| **Các yêu cầu cụ thể** | Không có |
| **Điều kiện trước** | Không có |
|
| **Điều kiện sau** | * Hiển thị gợi ý bài đăng cùng khu vực * Hiển thị gợi ý môi giới cùng khu vực * Cho phép sử dụng chức năng tính toán khoản vay |
| **Các yêu cầu mở rộng** | Không có |

#### Biểu đồ use case xem tin đăng

A diagram of a company

Description automatically generated

#### Biểu đồ tuần tự xem tin đăng

A diagram of a diagram

Description automatically generated

#### Biểu hoạt động xem tin đăng

A diagram with a diagram and a diagram

Description automatically generated with medium confidence

### Usecase cập nhật thông tin tài khoản

#### Đặc tả luồng sự cập nhật thông tin tài khoản

|  |  |
| --- | --- |
| **Use case** | Cập nhật thông tin tài khoản |
| **Tác nhân** | Admin, Member |
| **Mục đích** | Người dùng muốn xem và thay đổi thông tin đã đăng ký |
| **Luồng sự kiện chính** | 1. Người dùng chọn thông tin cá nhân cần chỉnh sửa |
| 2. Hệ thống hiển thị giao diện thông tin cá nhân |
| 3. Chọn chức năng chỉnh sửa |
| 4. Hệ thống hiển thị ra trang chỉnh sửa |
| 5. Người dùng nhập thông tin cần chỉnh sửa và gửi |
| 6. Hệ thống kiểm tra và lưu thông tin người dùng |
| **Luồng thay thế** | Không có |
| **Các yêu cầu cụ thể** | Email chỉnh sửa không được trùng trong hệ thống |
| **Điều kiện trước** | Người dùng đã đăng nhập vào hệ thống. |
|
| **Điều kiện sau** | Thông báo cập nhật thành công |
| **Các yêu cầu mở rộng** | Không có |

#### Biểu đồ use case cập nhật thông tin tài khoản

A diagram of a diagram

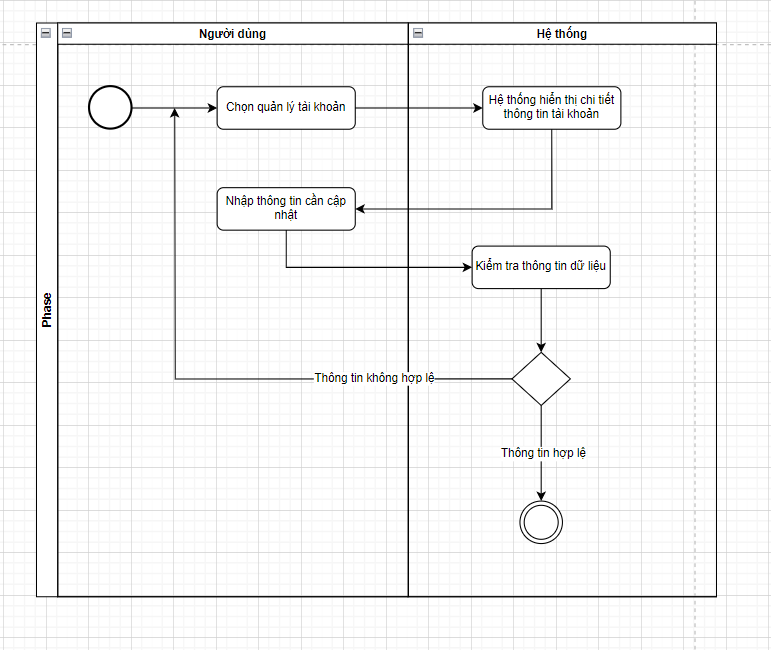
Description automatically generated

#### Biểu đồ tuần tự cập nhật thông tin tài khoản

A diagram of a diagram

Description automatically generated

#### Biểu hoạt động cập nhật thông tin tài khoản



### Usecase xóa tài khoản

#### Đặc tả luồng xóa tài khoản

|  |  |
| --- | --- |
| **Use case** | Xóa tài khoản |
| **Tác nhân** | Admin |
| **Mục đích** | Người dùng muốn xóa tài khoản đã đăng ký |
| **Luồng sự kiện chính** | 1. Người dùng chọn tài khoản cần xóa |
| 2. Hệ thống xóa tài khoản ra khỏi database |
| **Luồng thay thế** | Không có |
| **Các yêu cầu cụ thể** | Không có |
| **Điều kiện trước** | Người dùng đã đăng nhập vào hệ thống. |
|
| **Điều kiện sau** | Thông báo xóa thành công |
| **Các yêu cầu mở rộng** | Không có |

#### Biểu đồ use case xóa tài khoản

A diagram of a diagram

Description automatically generated

#### Biểu đồ tuần tự xóa tài khoản

A diagram of a diagram

Description automatically generated

#### Biểu hoạt động xóa tài khoản

A diagram of a diagram

Description automatically generated

### Usecase quản lý các tin đăng

## Thiết kế cơ sở dữ liệu

### Bảng Listing

- Dùng để lưu thông tin các tin đăng của thành viên trong hệ thống

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Chú thích** |
| id\_product | Int | Id tin đăng |
| name\_product | Varchar(1000) | Tên tin đăng |
| detail\_product | Varchar(500) | Chi tiết tin đăng |
| image\_product | Varchar(500) | Hình ảnh chính tin đăng |
| date\_create | DATETIME (20) | Ngày tạo |
| id\_productcate | INT | Loại nhà đất |
| price | DOUBLE | Giá |
| person\_modified | int | Người đăng |
| date\_modified | datetime | Ngày sửa |
| enable\_product | BOOLEAN | Tin đăng có hoạt động |
| address | VARCHAR(300) | Địa chỉ |
| area | int | Diện tích |
| floor\_space | int | Số tầng |
| owner\_project | VARCHAR(30) | Dự án |
| id\_producttype | int | Loại tin đăng |
| date\_expired | datetime | Ngày hết hạn |
| approve | boolean | Tin đăng có được duyệt |
| priority | boolean | Tin đăng ưu tiên |
| priority\_type | int | Thứ tự ưu tiên |

### Bảng Listingcategories

- Dùng để lưu thông tin các loại nhà đất có trong hệ thống

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Chú thích** |
| id\_productcate | Int | Id loại nhà đất |
| name\_productcate | Varchar(20) | Loại nhà đất |

### Bảng Listingtypes

- Dùng để lưu thông tin các tin đăng có trong hệ thống

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Chú thích** |
| id\_producttype | Int | Id loại tin đăng |
| name\_producttype | Varchar(20) | Loại tin đăng |

### Bảng Account

- Dùng để lưu thông tin tài khoản của hệ thống

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Chú thích** |
| id\_account | Int | Id tài khoản |
| username | Varchar(20) | Tên đăng nhập |
| password | Varchar(255) | Mật khẩu |
| first\_name | Varchar(20) | Họ |
| last\_name | Varchar(20) | Tên |
| email | Varchar(30) | Email |
| date\_of\_birth | Datetime | Ngày sinh |
| phone | Int | Điện thoại |
| address | Varchar(100) | Địa chỉ |
| avatar | Varchar(20) | Ảnh đại diện |
| create\_date | Datetime | Ngày tạo |
| last\_login | Datetime | Đăng nhập lần cuối |
| enable\_account | Boolean | Cho phép hoạt động |

### Bảng Roles

- Dùng để lưu các quyền đăng nhập có trong hệ thống

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Chú thích** |
| Roles\_id | Int | Id quyền |
| rolename | Varchar(100) | Tên quyền đăng nhập |

### Bảng Account\_roles

- Dùng để lưu các phân quyền của các tài khoản có trong hệ thống

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Chú thích** |
| Account\_id | Int | Id tài khoản |
| roles\_id | Int | Id quyền |

### Bảng Payment

* Dùng để lưu thanh toán của người dùng

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Chú thích** |
| id\_payment | Int | Id giao dịch thanh toán |
| name\_payment | Varchar(20) | Tên giao dịch thanh toán |
| id\_account | Int | Id tài khoản thanh toán |
| content\_payment | Varchar(300) | Nội dung giao dịch |
| amount | Double | Số tiền thanh toán |
| person\_modified | Int | Người chỉnh sửa giao dịch |
| date\_modified | Datetime | Ngày chỉnh sửa giao dịch |

### Bảng PaymentHistory

* Dùng để lưu lịch sử giao dịch

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Chú thích** |
| id\_paymenthistory | Int |  |
| id\_payment | Int |  |
| pre\_amount | Double |  |
| pay\_money | Double |  |
| aft\_amount | Double |  |
| person\_modified | Int |  |
| date\_modified | Datetime |  |

### Bảng Bank

* Dùng để lưu thông tin khoản vay của các ngân hàng

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Chú thích** |
| id\_bank | Int | Id ngân hàng |
| bank\_name | Varchar(100) | Tên ngân hàng |
| bank\_rate | Float | Lãi suất vay |
| maximum\_term | Int | Kỳ hạn vay tối đa |
| maximum\_rate | Int | Lãi suất vay tối đa |

### Bảng image

* Dùng để lưu hình ảnh của các tin đăng trong hệ thống

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Chú thích** |
| id | Int | Id ảnh |
| fileUri | String | Đường dẫn |
| groupid | Int | Id tin đăng |
| image | BinData | Dữ liệu ảnh |

# THỰC THI HỆ THỐNG

## Phần giao diện

## Phần hệ thống

### Auth Service

### Listing Service

Listing Service bao gồm các chức năng liên quan đến tin đăng. Được đăng ký với Eureka Server với port 8091. Bên dưới là cấu trúc thư mục của Listing Service.

A screenshot of a computer

Description automatically generated

#### Chức năng đăng tin

* *Entity*

Tạo 1 entity cho Listing với các thuộc tính của một bất động sản. Entity này tương ứng với bảng listing.

A screenshot of a computer program

Description automatically generated

* *Controller*

A screen shot of a computer program

Description automatically generated

Sử dụng method POST với api <http://localhost:8091/realestates/create-listing>. Khi người dùng sử dụng chức năng đăng tin hệ thống sẽ request đến api và lấy dữ liệu người dùng nhập từ form bằng anotation @RequestBody. Dữ liệu này sẽ map với Entity Listing và gọi đến tầng Service để giao tiếp với database sau đó trả về chuỗi JSON của thông tin bất động sản đã lưu.

* *Service*

A screen shot of a computer code

Description automatically generated

Ở tầng Service sẽ nhận dữ liệu của listing từ Controller và gọi đến Respository của listing để lưu dữ liệu.

* *Respository*

A black background with text

Description automatically generated

Sử dụng hàm save của Spring JPA để ánh xạ dữ liệu của listing tới bảng listing và lưu dữ liệu vào database.

#### Chức năng quản lý tin đăng

Chức năng quản lý tin đăng bao gồm các chức năng nhỏ như: xem, sửa, xóa.

* Xem chi tiết tin đăng
* *Controller*

Sử dụng method GET với api [http://localhost:8091/realestates/get-listing/{id}](http://localhost:8091/realestates/get-listing/%7bid%7d)

A screen shot of a computer code

Description automatically generated

Bên phía client sẽ gọi đến api kèm theo id của tin đăng. Controller sẽ lấy id gửi kèm từ client gọi xuống Service để xử lý tiếp tục.

* *Service*

Sử dụng hàm findById của JPAResponsitory để tìm tin đăng theo id.

A screen shot of a computer program

Description automatically generated

* Chỉnh sửa tin đăng
* *Controller*

Sử dụng method PUT với api http://localhost:8091/realestates/update-listing/{id}

A screen shot of a computer code

Description automatically generated

Khi Client chỉnh sửa thông tin của một tin đăng và xác nhận, front end sẽ gửi về id tin đăng cần sửa và dữ liệu người dùng chỉnh sửa bởi annotation @RequestBody.

* *Service*

Tại Service sẽ nhận thông tin đầu vào là id và dữ liệu Listing. Đầu tiên sẽ tìm listing cần chỉnh sửa sử dụng hàm findById(). Sau đó sẽ gọi phương thức set của từng thuộc tính và gọi xuống tầng Responsitory để lưu lại thông tin đã chỉnh sửa.

A screen shot of a computer program

Description automatically generated

* Xóa tin đăng
* *Controller*

Cũng giống như chức năng xem tin đăng chức năng xóa tin đăng cũng nhận về id của tin đăng cần xóa khi client sử dụng chức năng này. Khác ở chỗ chức năng này sẽ dụng dụng method DELETE và có api http://localhost:8091/realestates/delete-listing/{id}

A screen shot of a computer program

Description automatically generated

* *Service*

A screen shot of a computer code

Description automatically generated

Tầng Service sẽ gọi hàm deleteById() của Respository để thực hiện xóa.

#### Chức năng đề xuất tin đăng

* *Controller*

Khi người dùng chọn vào một tin đăng bất kỳ để xem hệ thống ngay lập tức gửi về server địa chỉ tin đăng mà người dùng đang xem. Sau đó gọi đến Service để thực hiện tìm kiếm tin đăng.

A black background with colorful text

Description automatically generated

* *Service*

Ở Service sẽ gọi findListingByAddress() của Responsitory.

A screen shot of a computer code

Description automatically generated

* *Responsitory*

Ở tầng này sử dụng anotation @Query để thực hiện câu lệnh SQL thao tác với database và trả về danh sách tin đăng cùng khu vực.

A computer screen shot of code

Description automatically generated

#### Chức năng tìm kiếm

* Search theo từ khóa:
* *Controller*

Client nhập từ khóa và chọn tìm kiếm, từ khóa sẽ được đọc qua Param gửi về.

A screen shot of a computer program

Description automatically generated

* *Service*

A computer screen shot of a black background

Description automatically generated

Sau khi nhận được từ khóa từ controller gửi xuống, Service sẽ gọi hàm searchProductByKeyword() từ Respository.

* *Responsitory*

A screen shot of a computer code

Description automatically generated

Ở Responsitory sẻ dùng câu query để tìm listing theo từ khóa mà người dùng mong muốn. Kết quả trả về là danh sách tin đăng đã tìm kiếm được.

* Search theo điều kiện:
* *Dto*

Tạo một class trong Dto có tên ListingSearchRequest chứa các thuộc tính tương ứng với các điều kiện tìm kiếm.

A screenshot of a computer program

Description automatically generated

* *Controller*

A black background with colorful text

Description automatically generated with medium confidence

Các điều kiện tìm kiếm sẽ được gửi xuống server thông qua method POST.

* *Service*

A computer screen shot of a program

Description automatically generated

Ở Service các điều kiện tìm kiếm sẽ được map với dto và gọi hàm findListingByFilterParam đẩy xuống Responsity để xử lý tiếp.

* *Responsitory*

*A computer screen shot of a program

Description automatically generated*

Ở chức năng tìm kiếm này sẽ sử dụng JPA Criteria để build câu lệnh SQL giúp quản lý dễ dàng câu lệnh SQL với nhiều param và dễ thay đổi và nâng cấp sau này.

### Image Service

### Payment Service

### Finance Service

### Crawl Service

# TRIỂN KHAI HỆ THỐNG

## Trang đăng ký

A screenshot of a computer

Description automatically generated

## Trang đăng nhập

A screenshot of a computer

Description automatically generated

## Trang chủ

A screenshot of a web page

Description automatically generated

## Trang đăng tin

A screenshot of a computer

Description automatically generated

## Trang chi tiết tài khoản

A screenshot of a computer

Description automatically generated

## Trang chi tiết tin đăng

A screenshot of a computer

Description automatically generated

## Quản lý tin đăng

A screenshot of a computer

Description automatically generated

## Quản lý tài khoản thành viên

A screenshot of a computer

Description automatically generated

# KẾT LUẬN

## Kết luận

## Hướng phát triển

# TÀI LIỆU THAM KHẢO

Tiếng Việt

…

Tiếng Anh

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. *Neural Computation*, *9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762