|  |  |  |
| --- | --- | --- |
| HỌC VIỆN NÔNG NGHIỆP VIỆT NAM  **KHOA CÔNG NGHỆ THÔNG TIN** |  | CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM  **Độc lập - Tự do - Hạnh phúc** |



**KHÓA LUẬN TỐT NGHIỆP**

**ĐỀ TÀI:**

***“Tìm hiểu mô hình kiến trúc Microservice và xây dựng ứng dụng”***

**Giảng viên hướng dẫn chính: Ths. Trần Trung Hiếu**

**Họ và Tên sinh viên: Đào Văn Tâm**

**Mã sinh viên: 601329**

**Lớp: K60THA**

**Hà Nội 2021**

**MỤC LỤC**

[**Phần 1: Mở đầu** 6](#_Toc63079107)

[**1.1** **Đặt vấn đề** 6](#_Toc63079108)

[**1.2** **Mục đích và yêu cầu** 8](#_Toc63079109)

[**1.2.1** **Mục đích** 8](#_Toc63079110)

[**1.2.2** **Yêu cầu** 8](#_Toc63079111)

[**Phần 2: Tổng quan tình hình nghiên cứu trong và ngoài nước** 10](#_Toc63079112)

[**2.1 Tình hình nghiên cứu trong nước** 10](#_Toc63079113)

[**2.2 Tình hình nghiên cứu ngoài nước** 10](#_Toc63079114)

[**2.3 Nêu tên đề tài, tính thời sự và tầm quan trọng của đề tài** 10](#_Toc63079115)

[***2.3.1. Các tính năng của mô hình kiến trúc Microservice*** 11](#_Toc63079116)

[***2.3.2. Đánh giá*** 11](#_Toc63079117)

[**Phần 3: Nội dung và phương pháp nghiên cứu** 15](#_Toc63079118)

[**3.1 Địa điểm và thời gian nghiên cứu** 15](#_Toc63079119)

[**3.2 Nội dung nghiên cứu** 15](#_Toc63079120)

[**3.2.1 HTML** 15](#_Toc63079121)

[**3.2.2 CSS** 17](#_Toc63079122)

[**3.2.3 Javascript** 18](#_Toc63079123)

[**3.2.4 Bootstrap** 18](#_Toc63079124)

[**3.2.5 Java** 18](#_Toc63079125)

[**3.2.6 Spring Boot** 20](#_Toc63079126)

[**3.2.7 Centralize-config (Spring Cloud Config Server)** 21](#_Toc63079127)

[**3.2.8 Eureka server** 24](#_Toc63079128)

[**3.2.9 API Gateway** 25](#_Toc63079129)

[**3.2.10 Zuul** 26](#_Toc63079130)

[**3.2.11** **Load-balancing** 27](#_Toc63079131)

[**3.2.12 Ribbon** 27](#_Toc63079132)

[**3.2.13 PostgreSQL** 29](#_Toc63079133)

[**3.2.14 Mongodb** 31](#_Toc63079134)

[**3.3 Phương pháp nghiên cứu** 32](#_Toc63079135)

[**Phần 4: Xây dựng chương trình** 33](#_Toc63079136)

[**4.1. Phát biểu bài toán** 33](#_Toc63079137)

[**4.2. Đặc tả yêu cầu phần mềm** 33](#_Toc63079138)

[***4.2.1. Người sử dụng hệ thống*** 33](#_Toc63079139)

[***4.2.2. Đặc tả yêu cầu chức năng*** 34](#_Toc63079140)

[**4.3.** **Thiết kế hệ thống** 63](#_Toc63079141)

[***4.3.1.*** ***Biểu đồ Use case*** 63](#_Toc63079142)

[***4.3.2. Sơ đồ trình tự*** 73](#_Toc63079143)

[***4.3.3. Sơ đồ lớp*** 92](#_Toc63079144)

[***4.3.4. Mô hình thực thể*** 98](#_Toc63079145)

[***4.3.5. Thiết kế dữ liệu*** 101](#_Toc63079146)

[***4.3.6. Thiết kế giao diện*** 111](#_Toc63079147)

[**4.4. Cài đặt và thử nghiệm** 117](#_Toc63079148)

[**4.4.1. Môi trường cài đặt và lập trình** 117](#_Toc63079149)

[**4.4.2. Thực thi ứng dụng** 117](#_Toc63079150)

[**4.4.3. Kiểm thử** 119](#_Toc63079151)

[**Phần 5: Kết luận và đề nghị** 121](#_Toc63079152)

[**5.1. Kết luận** 121](#_Toc63079153)

[**5.2. Đề nghị** 122](#_Toc63079154)

[**Phần 6: Tài liệu tham khảo** 123](#_Toc63079155)

**DANH MỤC HÌNH ẢNH**

[Hình 1. 1: Mô hình Monolithic 7](#_Toc63079156)

[Hình 2. 1: So sánh khác nhau giữa mô hình kiến trúc Monolithic và Microservices 12](file:///C:\Users\Laptop88\Desktop\khoa-luan-tot-nghiep.docx#_Toc63079157)

[Hình 2. 2: Cấu hình Dockerfile 14](file:///C:\Users\Laptop88\Desktop\khoa-luan-tot-nghiep.docx#_Toc63079158)

[Hình 3. 1: Quản lý cấu hình các service 22](#_Toc63079159)

[Hình 3. 2: Cấu hình centralize-configuration 22](#_Toc63079160)

[Hình 3. 3: Import eureka-server từ thư viện netflix 24](#_Toc63079161)

[Hình 3. 4: Khai báo @EnableEurekaServer để đánh dấu đây là eureka-servre 24](#_Toc63079162)

[Hình 3. 5: Giao diện Eureka-server khi chạy lên và sẽ quản lý các service 25](#_Toc63079163)

[Hình 3. 6: Cấu hình API Gateway với các service 26](#_Toc63079164)

[Hình 3. 7: Hình minh họa hoạt động của thuật toán Roud Robin 28](#_Toc63079165)

[Hình 3. 8: Cấu hình cho thuật toán Roud Robin 29](#_Toc63079166)

[Hình 3. 9: Cấu hình cho thuật toán WeightedResponseTimeRule 29](#_Toc63079167)

[Hình 4. 1: Use case tổng quát 63](#_Toc63079168)

[Hình 4. 2: Use case chi tiết 64](#_Toc63079169)

[Hình 4. 3: Use case đăng ký 66](#_Toc63079170)

[Hình 4. 4: Use case tìm bài đăng 67](#_Toc63079171)

[Hình 4. 5: Use case quản lý bài đăng 68](#_Toc63079172)

[Hình 4. 6: Use case quản lý bài viết 69](#_Toc63079173)

[Hình 4. 7: Use case quản lý phòng 70](#_Toc63079174)

[Hình 4. 8: Use case quản lý loại phòng 71](#_Toc63079175)

[Hình 4. 9: Use case quản lý ảnh bìa 72](#_Toc63079176)

[Hình 4. 10: Sơ đồ trình tự cho chức năng đăng nhập (chủ nhà) 73](#_Toc63079177)

[Hình 4. 11: Sơ đồ trình tự cho chức năng đăng nhập (Quản trị viên) 74](#_Toc63079178)

[Hình 4. 12: Sơ đồ trình tự cho chức năng đăng nhập (Sale) 75](#_Toc63079179)

[Hình 4. 13: Sơ đồ trình tự cho chức năng đăng xuất (Chủ nhà) 76](#_Toc63079180)

[Hình 4. 14: Sơ đồ trình tự cho chức năng đăng xuất (Admin) 77](#_Toc63079181)

[Hình 4. 15: Sơ đồ trình tự cho chức năng đăng xuất (Sale) 78](#_Toc63079182)

[Hình 4. 16: Sơ đồ trình tự cho chức năng đăng ký 79](#_Toc63079183)

[Hình 4. 17: Sơ đồ trình tự cho chức năng tìm bài đăng 80](#_Toc63079184)

[Hình 4. 18: Sơ đồ trình tự cho chức năng thêm bài đăng 81](#_Toc63079185)

[Hình 4. 19: Sơ đồ trình tự cho chức năng sửa bài đăng 82](#_Toc63079186)

[Hình 4. 20: Sơ đồ trình tự cho chức năng xóa bài đăng 83](#_Toc63079187)

[Hình 4. 21: Sơ đồ trình tự cho chức năng thêm loại phòng 83](#_Toc63079188)

[Hình 4. 22: Sơ đồ trình tự cho chức năng sửa loại phòng 84](#_Toc63079189)

[Hình 4. 23: Sơ đồ trình tự cho chức năng xóa loại phòng 85](#_Toc63079190)

[Hình 4. 24: Sơ đồ trình tự cho chức năng thêm bài viết (Admin) 86](#_Toc63079191)

[Hình 4. 25: Sơ đồ trình tự cho chức năng thêm bài viết (Sale) 87](#_Toc63079192)

[Hình 4. 26: Sơ đồ trình tự cho chức năng sửa bài viết (Admin) 88](#_Toc63079193)

[Hình 4. 27: Sơ đồ trình tự cho chức năng sửa bài viết (Sale) 89](#_Toc63079194)

[Hình 4. 28: Sơ đồ trình tự cho chức năng xóa bài viết (Admin) 90](#_Toc63079195)

[Hình 4. 29: Sơ đồ trình tự cho chức năng xóa bài viết (Sale) 90](#_Toc63079196)

[Hình 4. 30: Sơ đồ trình tự cho chức năng thêm ảnh bìa 91](#_Toc63079197)

[Hình 4. 31: Sơ đồ tổng quát các package của ManagerService 92](#_Toc63079198)

[Hình 4. 32: Sơ đồ tổng quát các package của HouseRentalService 93](#_Toc63079199)

[Hình 4. 33: Sơ đồ tổng quát các package của AccountService 93](#_Toc63079200)

[Hình 4. 34: Sơ đồ tổng quát các package của AuthService 94](#_Toc63079201)

[Hình 4. 35: Sơ đồ tổng quát các package của FileService 94](#_Toc63079202)

[Hình 4. 36: Sơ đồ tổng quát các package của InforService 95](#_Toc63079203)

[Hình 4. 37: Sơ đồ tổng quát các package của NotificationService 95](#_Toc63079204)

[Hình 4. 38: Sơ đồ tổng quát các package của StorageService 96](#_Toc63079205)

[Hình 4. 39: Sơ đồ lớp các chức năng liên quan đến phòng 96](#_Toc63079206)

[Hình 4. 40: Sơ đồ lớp các chức năng liên quan đến bài viết 97](#_Toc63079207)

[Hình 4. 41: Sơ đồ lớp các chức năng liên quan đến loại phòng 97](#_Toc63079208)

[Hình 4. 42: Sơ đồ lớp các chức năng liên quan đến tài khoản 98](#_Toc63079209)

[Hình 4. 43: Mô hình thực thể ứng với database house\_rental 99](#_Toc63079210)

[Hình 4. 44: Mô hình thực thể ứng với database auth\_service 100](#_Toc63079211)

[Hình 4. 45: Mô hình thực thể ứng với database infor\_service 100](#_Toc63079212)

[Hình 4. 46: Mô hình thực thể ứng với database file\_service 101](#_Toc63079213)

[Hình 4. 47: Giao diện trang chủ 111](#_Toc63079214)

[Hình 4. 48: Giao diện trang chủ (hiển thị các loại phòng) 111](#_Toc63079215)

[Hình 4. 49: Giao diện trang chủ (hiển thị các bài đăng) 112](#_Toc63079216)

[Hình 4. 50: Giao diện đăng nhập 113](#_Toc63079217)

[Hình 4. 51: Giao diện đăng ký 113](#_Toc63079218)

[Hình 4. 52: Giao diện quản lý bài đăng của chủ nhà 114](#_Toc63079219)

[Hình 4. 53: Giao diện quản lý thông tin người dùng 114](#_Toc63079220)

[Hình 4. 54: Giao diện quản lý phòng 115](#_Toc63079221)

[Hình 4. 55: Giao diện quản lý loại phòng 115](#_Toc63079222)

[Hình 4. 56: Giao diện quản lý bài viết 116](#_Toc63079223)

[Hình 4. 57: Giao diện quản lý ảnh bìa 116](#_Toc63079224)

[Hình 4. 58: Giao diện quản lý khách hàng của trang quản trị sale 117](#_Toc63079225)

[Hình 4. 59: Cấu hình Dockerfile của Centralize-configuration 118](#_Toc63079226)

[Hình 4. 60: Cấu hình Dockerfile của API Gateway 118](#_Toc63079227)

[Hình 4. 61: Cấu hình Dockerfile của Management-service 119](#_Toc63079228)

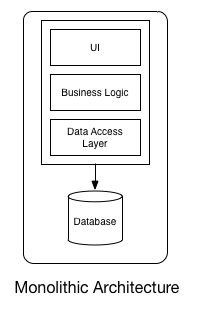
[Hình 4. 62: Cấu hình của file docker-compose.yml 119](#_Toc63079229)

# **Phần 1: Mở đầu**

* 1. **Đặt vấn đề**

Trong thời đại công nghệ hiện nay, có thể thấy công nghệ thông tin đang giữ một vai trò quan trọng đối với sự phát triển của loài người nói chung và sự phát triển kinh tế, chính trị, xã hội của một đất nước nói riêng. Với những bước tiến như vũ bão ở những tập kỷ cuối thế kỷ XX, công nghệ thông tin đã tạo nên một diện mạo mới – kỉ nguyên công nghệ thông tin. Nghành công nghệ thông tin rất đa dạng được chia thành nhiều lĩnh vực, trong đó phát triển ứng dụng web là một trong những linh vực có vai trò khá quan trọng và được con người hướng tới nhiều nhất. Từ những website tĩnh đơn sơ đến các website sinh động đẹp mắt như hiện nay, công nghệ phát triển ứng dụng web đã có những bước tiến vượt bậc. Cho đến hiện tại, lĩnh vực này đã vô cùng quen thuộc trong đời sống con người và sẽ là nghành nghề thu hút một nguồn nhân lực vô cùng lớn trong tương lai.

Trong quá khứ cũng như đến thời điểm hiện tại có 2 loại mô hình phổ biến hiện nay mà các doanh nghiệp cũng như các công ty trong và ngoài nước đã và đang phát triển đó là Monolithic và Microservice. Nhiều năm trước cũng như thời điểm hiện tại dường như mô hình Monolithic vẫn được đang áp dụng, tiếp cận dễ dàng.



Hình 1. 1: Mô hình Monolithic

Mô hình Monolothic hay còn gọi là mô hình kiến trúc một khối. Mô hình này sẽ bao gồm tầng giao diện người dùng (UI), tầng xử lý logic nghiệp vụ (Business Logic), tầng truy vấn dữ liệu (Data Access Layer), và cuối cùng là cơ sở dữ liệu (database). Tầng giao diện người dùng sẽ có nhiệm vụ giao tiếp với người dùng khi thao tác. Tầng xử lý logic nghiệp vụ sẽ đáp ứng yêu cầu thao tác dữ liệu từ phía giao diện người dùng và cũng là nơi kiểm tra các ràng buộc, tính toàn vẹn, xử lý yêu cầu nghiệp vụ. Tầng truy vấn dữ liệu sẽ có chức năng giao tiếp với hệ quản trị cơ sở dữ liệu như thực hiện đến lưu trữ và truy vẫn dữ liệu (tìm kiếm, thêm, sửa, xóa). Tất cả mọi quá trình phát triển code đều nằm trên một project và đều thông qua 3 tầng trên và dùng chung một hệ quản trị cơ sở dữ liệu.

Ưu điểm:  
- Dễ phát triển với hệ thống nhỏ.

- Dễ dàng test do toàn bộ project được đóng chung.

- Dễ phát triển do các công nghệ đều thống nhất ở tất cả các tầng của kiến trúc.

Nhược điểm:

Tuy cách tiếp cận kiến trúc một khối này dễ dàng nhưng lại bộc lộ nhiều khuyết điểm về lâu dài. Khi ứng dụng có số lượng người dùng tăng lên, yêu cầu thêm các tính năng mới, dữ liệu tăng lên, yêu cầu giao tiếp với các hệ thống bên ngoài khác nhiều hơn. Điều đó dẫn đến ứng dụng sẽ lớn hơn và việc bảo trì cũng như mở rộng sẽ khó khăn hơn. Một vấn đề khác nữa đó là do các module đều chạy trong một quá trình và hợp nhất với nhau, một lỗi trong bất kỳ module nào xảy ra cũng có khả năng làm lỗi cả một hệ thống.

Cách khắc phục:

Vậy nên để khắc phục những nhược điểm tồn tại trên mô hình kiến trúc Monolithic thì mô hình kiến trúc Microservice đã ra đời. Kiến trúc Microservice là một cách tiếp cận về kiến trúc, trong đó một ứng dụng duy nhất được tạo bởi nhiều thành phần (hay các dịch vụ) nhỏ hơn có thể triển khai độc lập. Kiến trúc Microservice hiện đang ngày càng phổ biến, ít nhất là đối với các nhà điều hành, những người đứng đầu dự án cũng như các nhà phát triển. Chính vì vậy, các lập trình viên càng có thêm cơ hội để tìm hiểu một cách dễ dàng và nhanh chóng hơn.

* 1. **Mục đích và yêu cầu**
     1. **Mục đích**
* Nghiên cứu mô hình kiến trúc Microservice và các công cụ, thư viện hỗ trợ như: Java, Spring Boot, Zuul, Eureka, Ribbon
* Ứng dụng mô hình kiến trúc Microservice để xây dựng ứng dụng đăng tin thuê nhà
  + 1. **Yêu cầu**
* Biết cách sử dụng ngôn ngữ lập trình Java, Spring framework và các công cụ hỗ trợ để thiết kế ứng dụng
* Xây dựng thành công ứng dụng đăng tin thuê nhà
* Yêu cầu chức năng:

+) Quản trị viên: Quản lý các bài đăng, quản lý các thành viên chủ nhà, quản lý loại phòng, quản lý phòng, quản lý bài viết, quản lý banner.

+) Sale: Quản lý thông tin khách hàng gửi đến

+) Chủ nhà: Đăng bài viết, sửa bài viết, xóa bài viết

+) Khách hàng: Tìm kiếm, xem thông tin bài đăng

**Phần 2: Tổng quan tình hình nghiên cứu trong và ngoài nước**

## **2.1 Tình hình nghiên cứu trong nước**

Ở Việt Nam, mặc dù công nghệ thông tin phát triển và muộn hơn so với nhiều nước trên thế giới nhưng việc nắm bắt công nghệ của Việt Nam lại không hề thua kém. Các công ty, doanh nghiệp Việt Nam đang học hỏi và áp dụng theo các công nghệ mới mà các nước phát triển áp dụng. Trong đó mô hình kiến trúc Microservice đang là một trong các mô hình được các doanh nghiệp áp dụng để triển khai hệ thống. Hướng tới những bước chuyển đổi công nghệ tại thị trường trong nước, nhiều doanh nghiệp hướng tới chia nhỏ hệ thống nguyên khối thành kiến trúc Microservice, nhằm cung cấp khả năng mở rộng lớn hơn và thời gian nhanh hơn, đảm bảo khả năng tái sử dụng cao. Do đó, kiến trúc Microservice là sự lựa chọn ưu việt cho các dự án phức tạp.

## **2.2 Tình hình nghiên cứu ngoài nước**

Hiện nay trên thế giới có nhiều tập đoàn như Amazon, eBay, Netflix đã giải quyết vấn đề ứng dụng bằng kiến trúc Microservice (nhiều dịch vụ nhỏ). Ý tưởng là chia nhỏ ứng dụng lớn ra thành các dịch vụ nhỏ để kết nối với nhau.

## **2.3 Nêu tên đề tài, tính thời sự và tầm quan trọng của đề tài**

Trong lĩnh vực công nghệ thông tin hiện nay, việc triển khai một hệ thống đang dần trở nên dễ dàng hơn nhờ vào sự ra đời cũng như phát triển của các công nghệ mới. Nhưng làm sao để có thể vừa triển khai dễ dàng vừa tối ưu hóa thì đó là điều vô cùng cần thiết. Với các doanh nghiệp họ luôn yêu cầu sản phẩm, ứng dụng hay hệ thống của họ phải đòi hỏi những điều đó.

Hiện nay các doanh nghiệp Việt nam đã và đang dần thay đổi mô hình kiến trúc trong sản phẩm hay hệ thống của họ. Nhằm mục đích bắt kịp xu thế mới cũng như ngày một nâng cấp hệ thống tốt hơn khắc phục được các vấn đề còn tồn đọng trước kia. Vì vậy mô hình kiến trúc Microservice sẽ giải quyết được các vấn đề trên.

Kiến trúc Microservice là kiểu mô hình kiến trúc tập hợp gồm nhiều ứng dụng nhỏ. Tất cả các dịch vụ được phân tách dựa trên các miền và chức năng của chúng và được phân bổ theo từng service.

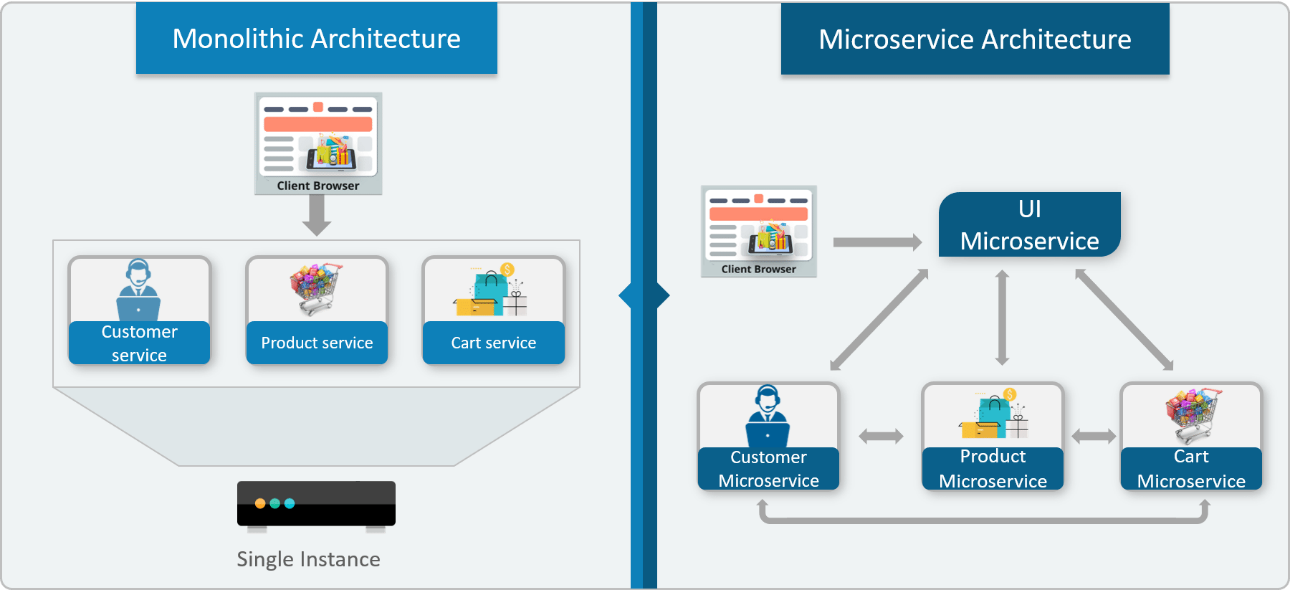
Mô hình kiến trúc Microservice sẽ có bộ cân bằng tải và thực thi riêng để thực thi các chức năng của service tương ứng.

Tất cả các service được giao tiếp thông qua API Gateway.

### ***2.3.1. Các tính năng của mô hình kiến trúc Microservice***

* Mỗi service trong hệ thống sẽ được tách biệt. Vì vậy, ứng dụng có thể dễ dàng được xây dựng, thay đổi và mở rộng quy mô
* Microservices được coi là các thành phần độc lập có thể dễ dàng thay thế và nâng cấp.
* Mỗi một thành phần trong kiến trúc microservice chỉ tập trung vào một nhiệm vụ duy nhất.
* Các lập trình viên hay các nhóm có thể làm việc độc lập với nhau trong quá trình phát triển.
* Không có mẫu chuẩn hóa hoặc bất kỳ mẫu công nghệ nào. Được tự do lựa chọn các công cụ hữu ích tốt nhất để có thể giải quyết vấn đề.

***2.3.2. Đánh giá***



Hình 2. 1: So sánh khác nhau giữa mô hình kiến trúc Monolithic và Microservices

Từ hình minh họa có thể thấy:

* Mô hình truyền thống sẽ bao gồm tất cả từ việc xây dựng hệ thống, phân chia các module, cấu hình và dùng chung cơ sở dữ liệu. Việc này khi muốn mở rộng hệ thống sẽ rất khó khăn và khi trong hệ thống có một lỗi.
* Hệ thống sẽ không hoạt động được.
* Với mô hình Microservice, mỗi tính năng sẽ được phân bổ cho một service khác nhau, xử lý dữ liệu riêng, có thể dùng ngôn ngữ khác nhau và thực hiện chức năng khác nhau, độc lập

Ưu điểm của Microservice:

* Phát triển độc lập: Tất cả các dịch vụ nhỏ có thể được phát triển dễ dàng dựa trên chức năng riêng của chúng.
* Triển khai độc lập: có thể triển khai trong riêng lẻ trên bất kỳ ứng dụng nào.
* Ngay khi một dịch vụ của ứng dụng không hoạt động, hệ thống vẫn tiếp tục hoạt động.
* Các ngôn ngữ và công nghệ khác nhau có thể sử dụng để xây dựng các service khác nhau của cùng một ứng dụng.
* Giảm thiểu sự gia tăng phức tạp khi gặp hệ thống lớn.
* Dễ nâng cấp, mở rộng.

Nhược điểm của kiến trúc Microservice:

* Microservice là mô hình kiến trúc chia nhỏ các service, nhưng khi chia nhỏ nhiều quá cũng dẫn đến khó kiếm soát.
* Việc quản lí nhiều database, và transaction giữa các service trong một hệ thống phân tán cũng là một khó khăn không nhỏ. Hay khi thực hiện test một service, cũng cần test các service có liên quan.
* Triển khai microservice cũng sẽ phức tạp hơn so với ứng dụng kiến trúc một khối.

Cách khắc phục nhược điểm của kiến trúc Microservice

* Docker chính là giải pháp để deploy microservice.Một microservice có thể được đóng gói trong docker image và phân lập thành một docker container. Bằng cách này thì ứng dụng có thể có môi trường độc lập với máy chủ.
* Thay vì sử dụng nhiều máy ảo, Docker container chia sẻ nhân hệ điều hành cho các host. Tiến trình từ container sẽ xuất hiện trong bảng tiến trình (process) của hệ điều hành mà Docker đang chạy.
* Để sử dụng Docker với microservice, bạn cần tạo Docker image qua một file gọi là Dockerfile.



Hình 2. 2: Cấu hình Dockerfile

Một số ví dụ hệ thống đang áp dụng mô hình kiến trúc Microservice

* Hệ thống ngân hàng trực tuyến TCB của ngân hàng Techcombank.
* Hệ thống ngân hàng trực tuyến VPBank.
* Hệ thống giao hàng tiết kiệm.
* Hệ thống mua sắm trực tuyến Tiki.

# **Phần 3: Nội dung và phương pháp nghiên cứu**

# **3.1 Địa điểm và thời gian nghiên cứu**

- Đề tài được thực hiện tại bộ môn Công nghệ phần mềm, trực thuộc Khoa Công nghệ thông tin

- Địa điểm thực tập tại: Công ty TNHH Rabiloo

- Thời gian thực hiện: Từ ngày 21/09/2020 đến ngày 29/11/2020

## **3.2 Nội dung nghiên cứu**

Nghiên cứu lý thuyết và các công cụ, công nghệ và thư viện hỗ trợ như: HTML, CSS, Javascript, Bootstrap, Java, Spring Boot, Zuul, Eureka, Ribbon, MySQL Server, SQL Server

### **3.2.1 HTML**

a) HTML là gì ?

- HTML là chữ viết tắt của cụm từ HyperText Markup Language, có nghĩa là “ngôn ngữ đánh dấu siêu văn bản”.

- HTML dùng để mô tả cấu trúc, các thuộc tính liên quan đến cách hiển thị của 1 đoạn văn bản nào đấy ra trình duyệt.

- Các phần tử HTML là các khối xây dựng nên trang web.

- Các phần tử HTML được biểu diễn bằng thẻ.

- Trình duyệt không hiển thị các thẻ HTML, nhưng sử dụng chúng để hiển thị nội dung của trang.

b) Bố cục cơ bản của HTML trên một trang web

- Thông thường, bố cụ của một trang web được chia làm 2 phần chính trong cặp thẻ <html></html>

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>This is a Heading</h1>

<p>This is a paragraph.</p>

</body>

</html>

Trong đó:

<!DOCTYPE html>: Dùng để xác định tài liệu này thuộc dạng html cho trình duyệt biết.

<html> </html>: Là phần tử gốc, nằm ngoài cùng mỗi trang web, có nhiệm vụ bao toàn bộ nội dung của trang web

<head> </head>: Phần mở đầu. Có nhiệm vụ khai báo tất cả thông tin của trang web

<title> </title>: Thẻ tiêu đề: Dùng để khai báo tiêu đề cho mỗi trang web. Tiêu đề này sẽ hiển thị ở thanh tab trên cùng của trình duyệt.

<body> </body>: Phần nội dung. Chứa tất cả nội dung sẽ hiển thị ra trên trình duyệt.

<h1> </h1> và <p> </p> : Các thẻ định dạng văn bản

c) Thẻ HTML

- Là các phần tử được bao quanh bởi các cặp ngoặc nhọn:

<tagname> Nội dung ở đây</tagname>

- Các thẻ trong HTML thường theo cặp

Ví dụ: <p> </p>

- Đầu tiên là thẻ bắt đầu (opening tag). Thẻ tiếp theo là thẻ kết thúc (closing tag)

- Thẻ kết thúc được viết giống như thẻ bắt đầu, chỉ khác thêm dấu gạch chéo (/) được chèn trước tên thẻ.

d) Trình duyệt web

- Có nhiều trình duyệt web hiện đang được sử dụng phổ biến trên như Chrome, IE, Firefox, Safari,… Mục đích của chúng là đọc các tài liệu HTML và hiển thị nội dung của chúng.

- Trình duyệt không hiển thị các thẻ HTML nhưng sử dụng nó để xác định cách hiển thị nội dung tài liệu.

e) Thẻ <!DOCTYPE>

- Thẻ khai báo <!DOCTYPE> dùng để xác định định dạng của tài liệu. Và giúp các trình duyệt hiển thị các trang web một cách chính xác.

- Thẻ <!DOCTYPE> chỉ xuất hiện một lần, ở đầu trang trước bất kỳ thẻ HTML nào.

- Thẻ không phân biệt chữ hoa và chữ thường. Nên <!DOCTYPE> hay <!doctype> đều giống nhau.

- Khai báo cho định dạng html là:  <!DOCTYPE html>.

### **3.2.2 CSS**

- Cascading Style Sheets, thường được viết tắt là CSS, là một ngôn ngữ định kiểu được sử dụng để định dạng – trình bày một tài liệu HTML hoặc XML (bao gồm các ngôn ngữ XML khác nhau như SVG hoặc XHTML). CSS mô tả quy cách các phần tử được cấu trúc phải được hiển thị trên màn hình, trên giấy, trong lời nói hoặc trên các phương tiện khác.

### **3.2.3 Javascript**

a) Khái niệm: JavaScrip là ngôn ngữ lập trình HTML và Web. JavaScript là một ngôn ngữ kịch bản dựa trên đối tượng nhằm phát triển các ứng dụng Internet chạy trên phía client và phía server. Javascript rất hữu ích trong việc xây dựng các hệ thống HTML có thể tương tác với người dùng.

b) Tại sao phải dùng Javascript ?

- Javascript là một trong 3 ngôn ngữ được tất cả nhà phát triển web sử dụng

- Ưu điểm của Javascript:

+) Sự tương tác Server ít hơn.

+) Phản hồi thông tin ngay lập tức tới khách truy cập.

+) Khả năng tương tác với người dùng tăng lên.

+) Giao diện phong phú hơn.

- Nhược điểm của Javascript:

+) Client-side JavaScript không cho phép đọc hoặc ghi các tập tin vì lý do bảo mật

+) Không được sử dụng cho các ứng dụng mạng, vì không hỗ trợ.

+) Không hỗ trợ xử lý đa luồng hay đa xử lý

### **3.2.4 Bootstrap**

- Bootstrap là một framework cho phép thiết kế web động nhanh hơn và dễ dàng hơn. Bootstrap là bao gồm các HTML templates, CSS templates và Javascript tạo ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm các plugin Javascript trong nó. Giúp cho việc thiết kế web động của bạn dễ dàng hơn và nhanh chóng hơn.

### **3.2.5 Java**

a) Tổng quan về ngôn ngữ Java

- Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.

- Java được tạo ra với tiêu chí “Viết một lần, thực thi khắp nơi”. Bởi Java là một ngôn ngữ lập trình có khả năng biên dịch và khả năng thông dịch. Chương trình nguồn viết bằng ngôn ngữ Java có đuôi (\*.java) đầu tiên được biên dịch thành tập tin có đuôi (\*.class) và sau đó sẽ được trình thông dịch thông dịch thành mã máy. Các Java class file có thể được dùng ở bất kỳ flatform nào

- Ưu điểm: Phương pháp này giúp các đoạn mã viết bằng Java có thể chạy được trên nhiều nền tảng khác nhau. Với điều kiện máy ảo Java (JVM) có hỗ trợ chạy trên nền tảng này

- Nhược điểm: Quá trình chạy các đoạn mã Java chậm hơn các ngôn ngữ biên dịch khác

b) Các dạng công nghệ Java

- Desktop applications – J2SE

+) Phiên bản chuẩn – Java 2 Standard Edition. J2SE hỗ trợ viết các ứng dụng đơn, ứng dụng client-server

+) Java Applications: ứng dụng Java thông thường trên desktop

+) Java Applets: ứng dụng nhúng hoạt động trong trình duyệt web

- Server applications – J2EE

+) Nền tảng Java 2, phiên bản doanh nghiệp – Java 2 Enterprise Edition. Hỗ trợ phát triển các ứng dụng thương mại

+) Chạy trên máy chủ lớn với sức mạnh xử lý và dung lượng bộ nhớ lớn, hỗ trợ gắn liền với servlet, jsp và XML.

- Mobile application – J2ME

+) Phiên bản thu nhỏ - Java 2 Micro Edition

+) Hỗ trợ viết các ứng dụng trên các thiết bị di động, không dây, thiết bị nhúng,…

c) Các dạng ứng dụng của Java

Dùng Java ta có thể viết các dạng ứng dụng sau:

* Ứng dụng độc lập (console application)
* Ứng dụng Applets
* Ứng dụng giao diện (GUI application)
* Ứng dụng Web (Servlet và Jsp)
* Ứng dụng nhúng (embedded applcation)
* Ứng dụng cơ sở dữ liệu (database application)
* Games

### **3.2.6 Spring Boot**

a) Spring Boot là gì ?

- Spring Boot là một module của Spring Framework, cung cấp tính năng RAD (Rapid Application Development) – Phát triển ứng dụng nhanh.

- Spring Boot được dùng để tạo các ứng dụng độc lập dựa trên Spring

- Với Spring Boot các cấu hình được giảm thiểu tối đa. Spring Boot hỗ trợ các bộ chứa nhúng (embedded containers) điều này cho phép các ứng dụng web có thể chạy độc lập mà không cần phải khai lên các Web Server.

b) Ưu điểm của Spring Boot

- Có các tính năng của Spring Framework

- Tạo ứng dụng độc lập, có thể chạy java -jar (cho cả java web)

- Nhúng trực tiếp các ứng dụng server (Tomcat, Jetty,…) do đó không cần phải triển khai file WAR

- Cấu hình ít, tự động cấu hình bất kì khi nào có thể -> giảm thời gian viết code, tăng năng suất

- Cung cấp nhiều plugin

- Chuẩn cho mô hình kiến trúc Microservice (Cloud support, giảm việc setup, config, các thư viện hỗ trợ,…)

### **3.2.7 Centralize-config (Spring Cloud Config Server)**

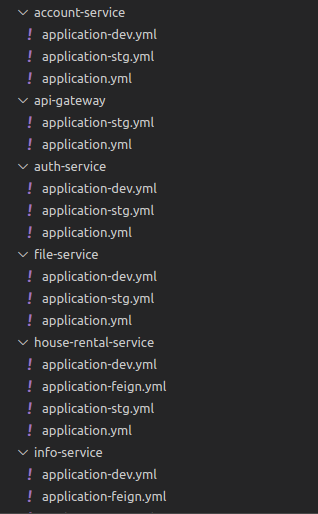
a) Khái niệm

- Spring cloud config server được hỗ trợ dưới dạng máy chủ và máy khách là các service trong hệ thống, các thuộc tính cấu hình được đưa lên máy chủ sẽ được ánh xạ đến các service con thông qua 2 đối tượng là Spring environment và PropertySource.

- Những thuộc tính có trong file config đều giống như trong các ứng dụng spring thông thường.

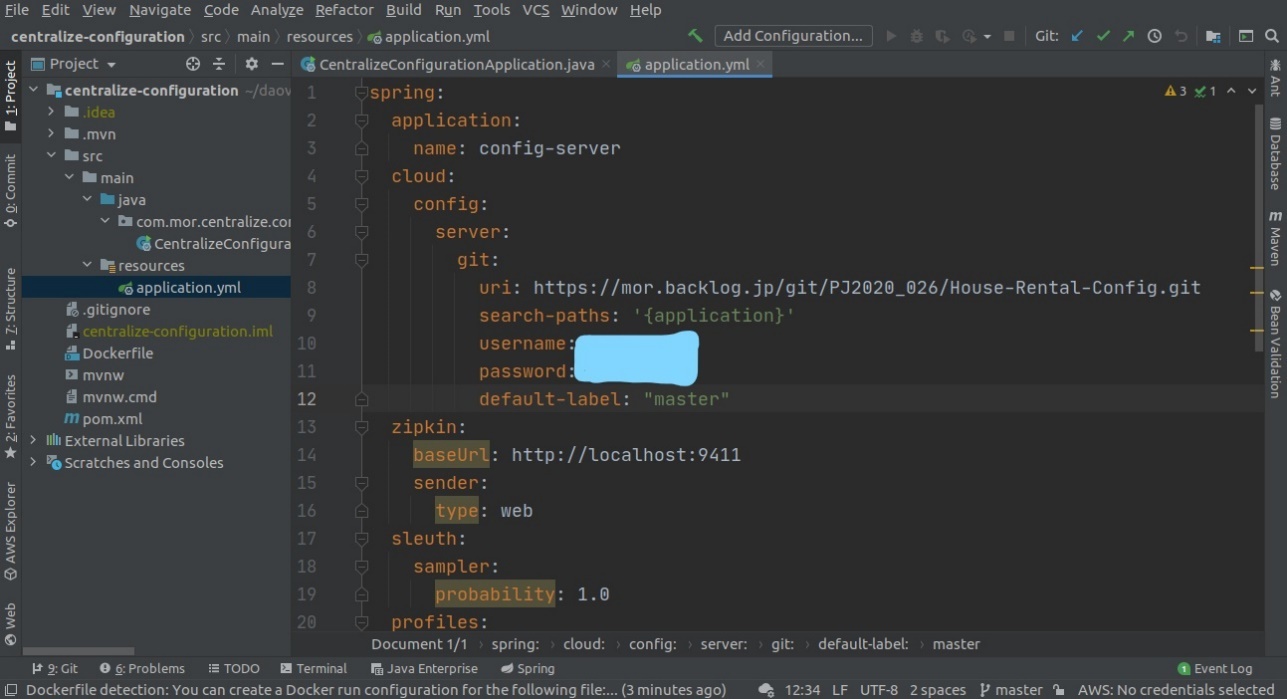
b) Cách thức hoạt động

- File House-rental-config: sẽ quản lý configuration thông qua git repository và sẽ cấu hình config cho mỗi service riêng và được đặt trong một thư mục riêng và environment trong một service sẽ phân biệt bằng file name theo mẫu “application-enviroment”.

****

Hình 3. 1: Quản lý cấu hình các service

**-** Trong Centralize-configuration(config-server): trong file application.yml cần phải cấu hình để config-server sử dụng file-system(House-rental-config) để quản lý và chỉ cho config-server quét ra các config file tương ứng với mỗi service.



Hình 3. 2: Cấu hình centralize-configuration

**-** Các service(Spring Cloud Config Client) cần kết nối tới config-server để lấy các configuration tương ứng với service đó thông qua cấu hình trong file bootstrap.yml (bao gồm các thông tin: địa chỉ của config-server, tên.

c) Ưu điểm

- Trong các ứng dụng thông thường, khi thay đổi config của project, ta cần build và deploy lại, với Spring cloud config khi thay đổi config, ứng dụng thông qua Spring cloud bus sẽ được cập nhật đến các service.

- Một service có thể được deploy multi node, và ở các máy khác nhau, nhờ Spring Cloud Config Server các node sẽ chạy một cách thống nhất về cài đặt, khi có thay đổi config sẽ không gặp khó khăn.

- Vì chứa tất cả các config của các môi trường, nên các service con khi được đưa lên các môi trường khác nhau sẽ đơn giản hơn.

d) Mục đích

- Quản lý configuration, cho phép truy xuất configuration của các service.

- Các service sẽ tự động truy xuất configuration của mình ngay khi khởi động.

- Sử dụng file configuration được lưu trên git để quản lý các configuration và chỉ chỗ cho Server quét ra các configuration file tương ứng với mỗi service.

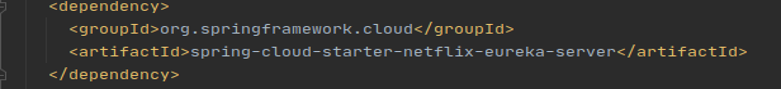
e) Tại sao cần centralize-config trong hệ thống microservice

- Trong hệ thống microservice, với mỗi service sẽ có các thông tin cấu hình như database, logging, … Thông thường những thông tin cấu hình này sẽ lưu trong file application.properties. Trong hệ thống microservice, với mỗi service sẽ start rất nhiều instance, vì vậy khi cần sửa đổi những thông tin cần cấu hình sẽ phải sửa tất cả ở các service instance mà đã start lên và đồng thời restart lại tất cả instance để apply lại những thay đổi đó.

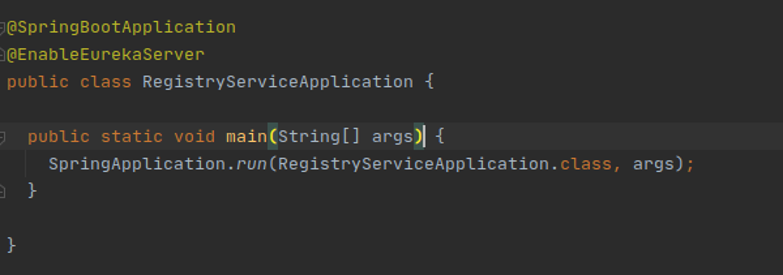
### **3.2.8 Eureka server**

a) Khái niệm

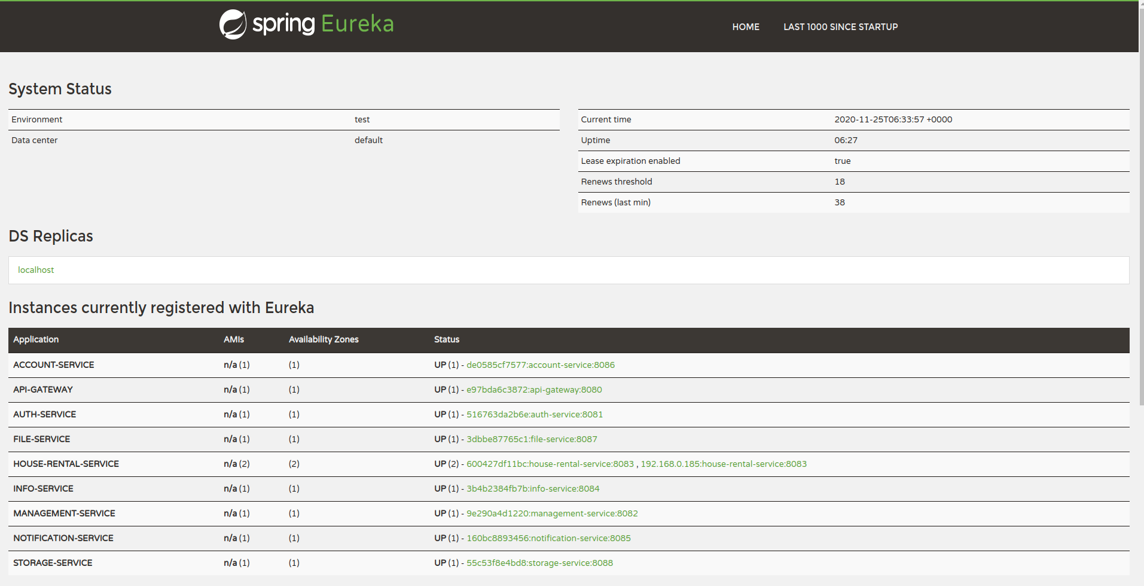
- Eureka cho phép các service trong hệ thống tự đăng kí với nó, và sau khoảng thời gian mỗi 5p các service sẽ tự ping lên eureka server để báo mình còn sống, thao tác đấy được gọi là health check

****

Hình 3. 3: Import eureka-server từ thư viện netflix

****

Hình 3. 4: Khai báo @EnableEurekaServer để đánh dấu đây là eureka-servre

****

Hình 3. 5: Giao diện Eureka-server khi chạy lên và sẽ quản lý các service

### **3.2.9 API Gateway**

a) Khái niệm:

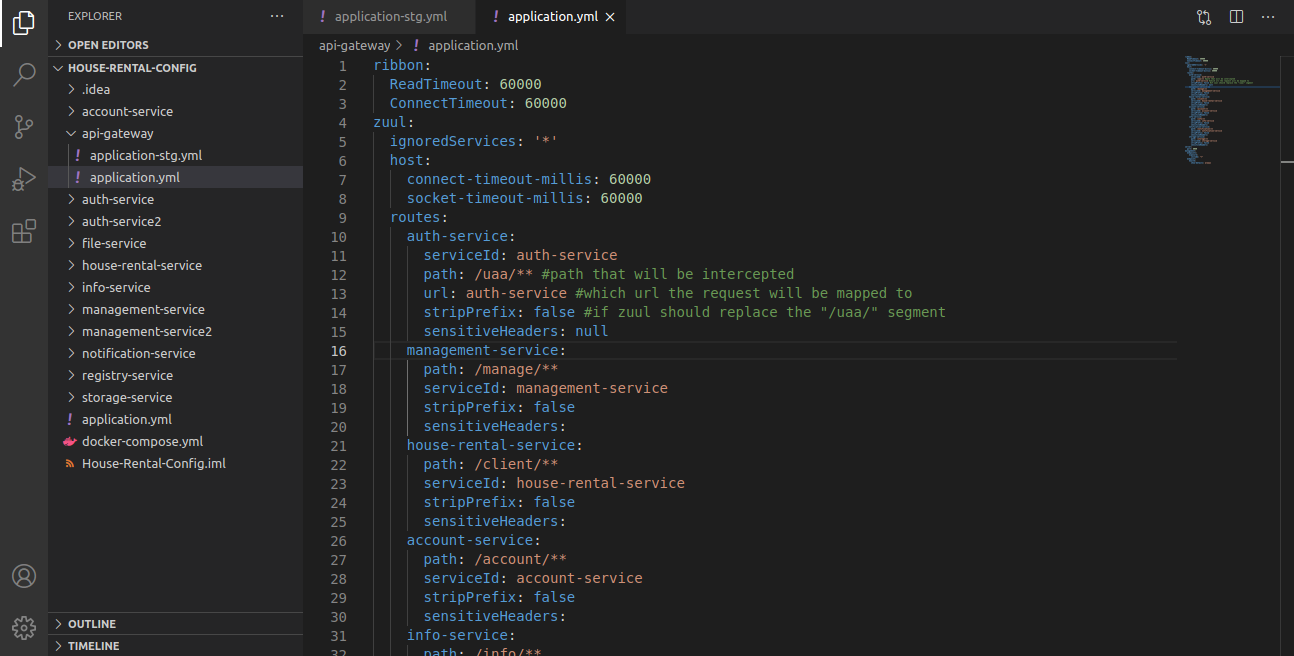
- API Gateway đóng vai trò cân bằng tải và router các request. Trong nó sử dụng ribbon để cân bằng tải và zuul để router.

- là cổng trung gian nhận các request từ phía client và điều hướng đến các service cụ thể.

b) Tác dụng:

- Để cho font-end giao tiếp với một địa chỉ duy nhất mà không cần quan tâm đến các con service con chạy trên địa chỉ nào vì đã có zuul router cho hệ thống rồi.

- API Gateway sẽ nhận vào tất cả các request từ phía client, xử lý bằng cách gọi nhiều microservice và tổng hợp lại rồi đưa ra kết quả đường dẫn tốt nhất bằng thuật toán.

****

Hình 3. 6: Cấu hình API Gateway với các service

- Spring Cloud Zuul theo mặc định hiển thị tất cả các dịch vụ được đăng ký trong máy chủ Eureka. Nếu bạn muốn bỏ qua việc bổ sung tự động mọi dịch vụ, bạn phải đặt thuộc tính zuul.ignored-services: ‘\*’. Zuul sẽ tìm danh sách các dịch vụ từ Eureka và tự động liên kết chúng.

- Nhưng khi chỉ rõ đường path của các service nó sẽ chỉ giao tiếp với service đó và cùng với serviced.

### **3.2.10 Zuul**

- Zuul hoạt động như một cổng API. Nó nhận tất cả các yêu cầu đến từ giao diện người dùng và sau đó ủy quyền các yêu cầu cho các dịch vụ nhỏ bên trong. Vì vậy, phải tạo một microservice hoàn toàn mới hỗ trợ Zuul và dịch vụ này nằm trên tất cả các microservice khác. Nó hoạt động như một dịch vụ Edge hoặc dịch vụ hướng tới khách hàng. API dịch vụ của nó phải được hiển thị với máy khách / giao diện người dùng. Máy khách gọi dịch vụ này như một proxy cho một dịch vụ nhỏ bên trong, sau đó dịch vụ này ủy quyền yêu cầu cho dịch vụ thích hợp.

### **3.2.11** **Load-balancing**

a) Khái niệm:

- Load Balancing dùng để phân phối khối lượng công việc trên nhiều tài nguyên như: server, hay ổ đĩa,…

- Load Balancing nhằm mục đích cải thiện thời gian phản hồi của ứng dụng, tránh làm quá tải một tài nguyên, tối ưu hóa việc sử dụng tài nguyên và tối đa hóa thông lượng.

b) Phân loại:

- Có 2 kiểu cân bằng tải:

+) Cân bằng tải phía máy chủ (Server-side Load Balancing): Các instance của các service được triển khai trên nhiều server và sau đó bộ cân bằng tải sẽ được đặt trước chúng. Tất cả request khi đến sẽ được bộ cân bằng tải quyết định 1 request nào đó nhờ vào một số thuật toán.

+) Cân bằng tải phía Client (Client-side Load Balancing): Các service phía client sẽ nắm giữ danh sách các máy chủ và sẽ quyết định máy chủ nào sẽ đươc yêu cầu đến nhờ vào thuật toán của cân bằng tải

### **3.2.12 Ribbon**

a) Khái niệm:

- Ở đây ta sẽ dùng ribbon hỗ trợ từ thư viện Netflix

- Ribbon cung cấp bộ cân bằng tải phần mềm để giao tiếp với cụm máy chủ. Bộ cân bằng tải cung cấp các chức năng cơ bản sau:

+) Cung cấp tên DNS hoặc IP công cộng của từng máy chủ cho ứng dụng khách giao tiếp.

+) Xoay vòng danh sách các máy chủ theo logic nhất định.

b) Một số thuật toán cân bằng tải:

- Roud Robin:

+) Round-robin load balancing là thuật toán cân bằng tải được triển khai rộng rãi nhất. Cân bằng tải sẽ điều hướng các request kết nối giữa các máy chủ và khách theo thứ tự nhận được các request.

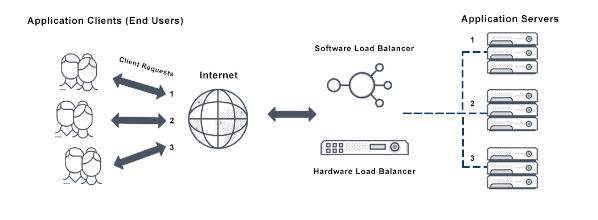
Ví dụ: Có 3 máy chủ: Server A, Server B, Server C

Yêu cầu đầu tiên được gửi đến Server A.

Yêu cầu thứ hai được gửi đến Server B.

Yêu cầu thứ ba được gửi đến Server C.

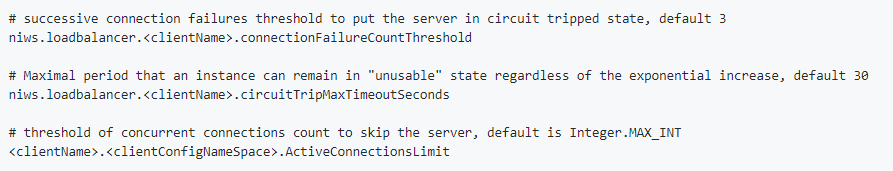
Bộ cân bằng tải tiếp tục chuyển các yêu cầu đến server dựa theo thứ thự này. Điều này đảm bảo máy chủ được phân phối đồng đều để xử lý lưu lượng truy cập cao.



Hình 3. 7: Hình minh họa hoạt động của thuật toán Roud Robin

+) Để bật các thuộc tính sau cần được đặt thông qua Archaius

ConfigurationManager:



Hình 3. 8: Cấu hình cho thuật toán Roud Robin

**-** WeightedResponseTimeRule

**+)** Với thuật toán này mỗi máy chủ được cung cấp một trọng số theo thời gian phản hồi trung bình của nó. Thời gian phản hồi càng lâu thì trọng số càng giảm. Quy tắc chọn ngẫu nhiên một máy chủ trong đó khả năng được xác định bởi trọng số của máy chủ.

**+)** Để bật WeightedResponseTimeRule, hãy đặt nó bằng bộ cân bằng tải qua API hoặc đặt thuộc tính sau:



Hình 3. 9: Cấu hình cho thuật toán WeightedResponseTimeRule

### **3.2.13 PostgreSQL**

- PostgreSQL là một hệ thống cơ sở dữ liệu quan hệ đối tượng mã nguồn mở mạnh mẽ, sử dụng và mở rộng ngôn ngữ SQL kết hợp với nhiều tính năng giúp lưu trữ và chia tỷ lệ một cách an toàn khối lượng công việc dữ liệu phức tạp nhất. Nguồn gốc của PostgreSQL có từ năm 1986 như một phần của dự án POSTGRES tại Đại học California ở Berkeley và đã có hơn 30 năm phát triển tích cực trên nền tảng cốt lõi.

- PostgreSQL được thiết kế để chạy trên các nền tảng tương tự UNIX. Tuy nhiên, PostgreSQL sau đó cũng được điều chỉnh linh động để có thể chạy được trên nhiều nền tảng khác nhau như Mac OS X, Solaris và Windows

- PostgreSQL sở hữu một hệ tính năng đa dạng giúp hỗ trợ các nhà phát triển xây dựng app, các nhà quản trị bảo vệ toàn vẹn dữ liệu, và tạo ra một môi trường chịu lỗi fault-tolerant giúp bạn quản lý dữ liệu bất kể tập dữ liệu lớn hay nhỏ. Bên cạnh hệ thống nguồn mở và miễn phí, PostgreSQL cũng có khả năng mở rộng tuyệt vời. Ví dụ, bạn có thể định nghĩa các kiểu dữ liệu riêng của bạn, xây dựng các hàm tùy chỉnh, hay viết mã từ các ngôn ngữ lập trình khác nhau mà không cần biên dịch lại cơ sở dữ liệu!

- PostgreSQL tuân theo tiêu chuẩn SQL nhưng không mâu thuẫn với các tính năng truyền thống hay có thể dẫn đến các quyết định kiến trúc gây hại. Nhiều tính năng theo tiêu chuẩn SQL được hỗ trợ, tuy nhiên đôi khi có thể có cú pháp hoặc hàm hơi khác một chút.

- Ưu điểm của PostgreSQL:

+) PostgreSQL là hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, miễn phí và mạnh mẽ

+) PostgreSQL được hỗ trợ bởi một cộng đồng nhiệt tình và đầy kinh nghiệm.

+) Bất kể các tính năng tiên tiến thì PostgreSQL đều được hỗ trợ bởi các công cụ tuyệt vời của bên thứ 3 trong việc quản lý cơ sơ dữ liệu

+) Nó có thể mở rộng thiết lập PostgreSQL với các lưu trữ thủ tục, giống như một tiến trình RDBMS.

+) PostgreSQL không chỉ là một hệ thống cơ sở dữ liệu quản trị, mà nó còn được hóa dữ liệu đối tượng.

- Nhược điểm của PostgreSQL:

+) Hiệu suất: Đối với các toán tử đơn giản thì PostgreSQL thực hiện kém hiệu quả hơn so với các hệ quản trị cơ sở dữ liệu quan hệ khác như MySQL. Nhưng đối với toán tử phức tạp thì PostgreSQL thực hiện tốt hơn rất nhiều.

+) Sự phổ biến: Do chưa có sự phổ biến nên ảnh hưởng đến người dùng nhận được sự hỗ trợ kịp thời.

+) Hosting: Do các yếu tố được đề cập bên trên nên nó khó được hỗ trợ bởi các host.

### **3.2.14 Mongodb**

- MongoDB là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL(\*) hàng đầu, được hàng triệu người sử dụng. MongoDB được viết bằng C++.

- Ngoài ra, MongoDB là một cơ sở dữ liệu đa nền tảng, hoạt động trên các khái niệm Collection và Document, nó cung cấp hiệu suất cao, tính khả dụng cao và khả năng mở rộng dễ dàng.

- NoSQL là 1 dạng CSDL mã nguồn mở không sử dụng Transact-SQL để truy vấn thông tin. NoSQL viết tắt bởi: None-Relational SQL, hay có nơi thường gọi là Not-Only SQL. CSDL này được phát triển trên Javascript Framework với kiểu dữ liệu JSON. (Cú pháp của JSON là “key:value”) NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu xót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS về tốc độ, tính năng, khả năng mở rộng, memory cache,...

- Ưu điểm của Mongodb:

+) Ít schema hơn: Vì schema được sinh ra là để nhóm các đối tượng vào 1 cụm, dễ quản lý. Ví dụ như tạo 1 schema tên là Students chẳng hạn thì chỉ có những gì liên quan đến student thì mới được cho vào schema này. Trong khi đó trong mongodb thì chỉ 1 collection ta có thể chứa nhiều document khác nhau . Với mỗi document thì số trường, nội dung, kích thước lại có thể khác nhau.

+) Cấu trúc của một đối tượng rõ ràng.

+) Không có các Join phức tạp.

+) Khả năng mở rộng cực lớn: việc mở rộng dữ liệu mà không phải lo đến các vấn đề như khóa ngoại, khóa chính, kiểm tra ràng buộc, ... MongoDB cho phép thực hiện replication và sharding nên việc mở rộng cũng thuận lợi hơn.

- Nhược điểm của Mongodb:

+) Dữ liệu được caching, lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.

+) Như đã giới thiệu ở trên, mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

## **3.3 Phương pháp nghiên cứu**

- Tham khảo giáo trình, tài liệu liên quan đến nội dung đề tài

- Tìm hiểu, nghiên cứu lý thuyết về các kỹ thuật thiết kế và xây dựng website

- Phân tích, thiết kế đối tượng và cơ sở dữ liệu quan hệ

- Xây dựng chương trình theo mô hình tiến hóa và kiểm thử

- Thu thập ý kiến người sử dụng để bảo trì và cập nhật

# **Phần 4: Xây dựng chương trình**

## **4.1. Phát biểu bài toán**

Hiện nay hình thức cho thuê nhà vẫn đang phổ biến trong đời sống nhằm đáp ứng nhu cầu cho người dân. Thế nhưng quy trình để tìm và thuê nhà mà thỏa mãn về nhu cầu cũng như đáp ứng tiêu chí của người thuê sẽ gặp một số khó khăn nhất định như:

- Không biết được giá cả, chi phí thuê như nào, cao hay thấp và giá đó so với mặt bằng chung chênh ra sao.

- Không có ảnh chi tiết để người thuê biết rõ thông tin hơn.

- Khó khăn trong việc tìm những nhà có địa điểm thuận tiện cho việc đi lại.

- Khó khăn trong việc cần thông tin liên hệ với chủ.

Qua những bất cập trên em đã quyết định thực hiện đề tài “Xây dựng ứng dụng đăng tin thuê nhà”. Nhằm giúp mọi người hay bất kỳ ai có thể nắm bắt thông tin rõ hơn, khách hàng có thể truy cập ứng dụng để xem thông tin, giá cả phù hợp với nhu cầu. Dưới sự hướng dẫn của Ths. Trần Trung Hiếu, em mong muốn sẽ làm ra một ứng dụng trực tuyến để tối ưu nhu cầu cho khách hàng.

## **4.2. Đặc tả yêu cầu phần mềm**

### ***4.2.1. Người sử dụng hệ thống***

|  |  |
| --- | --- |
| Người sử dụng | Mô tả |
| Quản trị viên | Quản lý: thêm, sửa, xóa phòng; Quản lý: thêm, sửa, xóa bài viết; Quản lý: thêm, sửa, xóa loại phòng. |
| Sale | Quản lý thông tin khách hàng đăng ký, Quản lý: thêm, sửa, xóa phòng; Quản lý: thêm, sửa, xóa loại phòng. |
| Chủ nhà | Đăng bài, sửa bài, xóa bài |
| Khách hàng | Tìm kiếm nhà |

### ***4.2.2. Đặc tả yêu cầu chức năng***

*4.2.2.1.**SRS001: MODULE Login*

a.Mô tả nghiệp vụ:

- Chức năng đăng nhập cho phép quản trị viên đăng nhập vào trang dành cho quản trị viên, với chủ nhà cho phép đăng nhập vào trang dành cho chủ nhà.

b. Dòng sự kiện chính.

- Đối với quản trị viên:

+) Người dùng đăng nhập với link: admin.vitahousing.com bao gồm thông tin “Tên người dùng” và “Mật khẩu”.

+) Tiếp theo nhấn nút “Đăng nhập”, hệ thống sẽ hiển thị trang quản trị cho quản trị viên.

- Đối với Sale:

+) Người dùng đăng nhập với link: admin.vitahousing.com bao gồm thông tin “Tên người dùng” và “Mật khẩu”.

+) Tiếp theo nhấn nút “Đăng nhập”, hệ thống sẽ hiển thị trang quản trị cho nhân viên sale.

- Đối với chủ nhà:

+) Người dùng truy cập vào website đăng tin.

+) Nếu người dùng muốn đăng nhập vào hệ thống website thì nhấn “Đăng nhập”, sau đó hệ thống sẽ hiện thị trang đăng nhập cho người dùng điền thông tin cá nhân để đăng nhập vào website.

+) Người dùng sẽ điền thông tin vào trang đăng nhập gồm “Tên người dùng” và “Mật khẩu”.

+) Tiếp theo nhấn nút “Đăng nhập”, nếu thông tin “Tên người dùng” và “Mật khẩu” người dùng điền đúng, hệ thống sẽ hiện thị trang quản trị dành cho chủ nhà.

c. Dòng sự kiện rẽ nhánh.

- Đối với quản trị viên:

+) Người dùng đăng nhập với link: admin.vitahousing.com bao gồm thông tin “Tên người dùng” và “Mật khẩu”.

+) Nếu người dùng không nhập hai trường “Tên người dùng” và “Mật khẩu”, hệ thống sẽ báo “Vui lòng nhập tên người dùng!” và “Vui lòng nhập mật khẩu!” tại 2 trường tương ứng.

+) Nếu người dùng nhập “Tên người dùng” nhưng không điền mật khẩu, hệ thống sẽ thông báo “Vui lòng nhập mật khẩu!”.

+) Nếu người dùng nhập “Mật khẩu” nhưng không điền “Tên người dùng”, hệ thống sẽ thông báo “Vui lòng nhập tên người dùng!”.

+) Nếu người dùng nhập sai thông tin một trong hai trường “Tên người dùng”, “Mật khẩu” hoặc sai cả hai trường thì hệ thống sẽ thông báo “invalid\_grant!”.

- Đối với Sale:

+) Người dùng đăng nhập với link: admin.vitahousing.com bao gồm thông tin “Tên người dùng” và “Mật khẩu”.

+) Nếu người dùng không nhập hai trường “Tên người dùng” và “Mật khẩu”, hệ thống sẽ báo “Vui lòng nhập tên người dùng!” và “Vui lòng nhập mật khẩu!” tại 2 trường tương ứng.

+) Nếu người dùng nhập “Tên người dùng” nhưng không điền mật khẩu, hệ thống sẽ thông báo “Vui lòng nhập mật khẩu!”.

+) Nếu người dùng nhập “Mật khẩu” nhưng không điền “Tên người dùng”, hệ thống sẽ thông báo “Vui lòng nhập tên người dùng!”.

+) Nếu người dùng nhập sai thông tin một trong hai trường “Tên người dùng”, “Mật khẩu” hoặc sai cả hai trường thì hệ thống sẽ thông báo “invalid\_grant!”.

- Đối với chủ nhà:

+) Nếu bỏ trống thông tin “Tên người dùng” và “Mật khẩu” thì sẽ hiện thông báo “ Vui lòng nhập tên người dùng!” và “Vui lòng nhập mật khẩu!” cho người dùng thấy để đăng nhập lại.

+) Nếu người dùng điền “Tên người dùng” nhưng bỏ trống “Mật khẩu” thì hệ thống sẽ hiện thông báo “Vui lòng nhập mật khẩu!” cho người dùng biết và đăng nhập lại vào form đăng nhập.

+) Nếu người dùng điền “Mật khẩu” nhưng bỏ trống “Tên người dùng” thì hệ thống sẽ hiện thông báo “Vui lòng nhập tên người dùng!” cho người dùng biết và đăng nhập lại vào form đăng nhập.

+) Nếu người dùng đăng nhập sai thông tin “Tên người dùng” và “Mật khẩu” thì hệ thống sẽ thông báo “Bad credentinals”.

d. Điều kiện trước.

- Đối với quản trị viên:

+) Quản trị viên cần đăng nhập vào link admin.vitahousing, rồi sau đó click vào “Đăng nhập”.

- Đối với Sale:

+) Sale cần đăng nhập vào link admin.vitahousing, rồi sau đó click vào “Đăng nhập”.

- Đối với chủ nhà:

+) Chủ nhà sẽ phải truy cập vào website, rồi sau đó click vào “Đăng nhập”.

e. Dòng sự kiện sau.

- Đối với quản trị viên:

+) Sau khi điền đủ thông tin chính xác thì sẽ chuyển hướng sang trang dành cho quản trị viên.

- Đối với sale:

+) Sau khi điền đủ thông tin chính xác thì sẽ chuyển hướng sang trang dành cho sale.

- Đối với chủ cho thuê:

+) Sau khi điền đủ thông tin chính xác thì sẽ chuyển hướng sang trang dành cho chủ nhà.

f. Tác nhân: Quản trị viên, Chủ nhà, Sale.

*4.2.2.2. SRS002: MODULE Logout*

a.Mô tả nghiệp vụ.

- Chức năng đăng xuất cho phép người dùng thoát ra để kết thúc phiên làm việc hoặc không muốn đăng nhập vào hệ thống để truy cập thông tin.

b. Dòng sự kiện chính.

- Đối với quản trị viên:

+) Người dùng sẽ click vào biểu tượng icon bên phải phía trên màn hình và ấn nút “Đăng xuất”. Hệ thống sẽ đưa người dùng quay lại trang đăng nhập dành cho quản trị viên.

- Đối với sale:

+) Người dùng sẽ click vào biểu tượng icon bên phải phía trên màn hình và ấn nút “Đăng xuất”. Hệ thống sẽ đưa người dùng quay lại trang đăng nhập dành cho sale.

- Đối với chủ nhà:

+) Người dùng sẽ click vào biểu tượng icon đại diện góc bên phải trên cùng màn hình sau đó sẽ xuất hiện nút “Đăng xuất”, tiếp theo người dùng sẽ nhấn vào nút “Đăng xuất” và hệ thống sẽ đưa người dùng về trang chủ.

c. Dòng sự kiện rẽ nhánh.

d. Điều kiện trước.

- Đối với quản trị viên:

+) Người dùng đã đăng nhập thành công vào trang quản trị dành cho quản trị viên.

- Đối với sale:

+) Người dùng đã đăng nhập thành công vào trang quản trị dành cho sale.

- Đối với chủ nhà:

+) Người dùng đã đăng nhập thành công vào trang dành cho chủ nhà.

e. Dòng sự kiện sau.

- Đối với quản trị viên:

+) Sau khi click vào nút “Đăng xuất” hệ thống sẽ đưa người dùng về trang đăng nhập dành cho quản trị viên.

- Đối với sale:

+) Sau khi click vào nút “Đăng xuất” hệ thống sẽ đưa người dùng về trang đăng nhập dành cho sale.

- Đối với chủ nhà:

+) Sau khi click vào nút “Đăng xuất” hệ thống sẽ đưa người dùng về trang chủ.

f. Tác nhân: Quản trị viên, sale, chủ nhà.

*4.2.2.3. SRS003 MODULE Đăng ký*

a.Mô tả nghiệp tả nghiệp vụ:

- Chức năng đăng ký cho phép người dùng đăng ký thông tin của mình để đăng nhập vào website đăng bài.

b. Dòng sự kiện chính.

- Đối với admin:

+) Người dùng sẽ đăng nhập vào trang chủ website.

+) Khi người dùng muốn đăng ký tài khoản cá nhân để đăng nhập vào website, chỉ cần nhấn nút “Đăng ký” xuất hiện ở trên cùng bên phải màn hình.

+) Sau đó hệ thống sẽ hiển thị trang đăng ký.

+) Tiếp theo người dùng sẽ điền thông tin cá nhân của mình vào form đăng ký. Cuối cùng sau khi điền đủ thông tin người dùng sẽ nhấn nút “Đăng ký” và hệ thống sẽ hiện thông báo cho người dùng “Đăng ký thành công”.

+) Như vậy người dùng đã đăng ký tài khoản thành công với danh nghĩa mặc định là chủ nhà.

+) Tiếp theo Admin sẽ cấp lại quyền cho người đó từ quyền chủ nhà thành quyền admin.

- Đối với sale:

+) Người dùng sẽ đăng nhập vào trang chủ website.

+) Khi người dùng muốn đăng ký tài khoản cá nhân để đăng nhập vào website, chỉ cần nhấn nút “Đăng ký” xuất hiện ở trên cùng bên phải màn hình.

+) Sau đó hệ thống sẽ hiển thị trang đăng ký.

+) Tiếp theo người dùng sẽ điền thông tin cá nhân của mình vào form đăng ký. Cuối cùng sau khi điền đủ thông tin người dùng sẽ nhấn nút “Đăng ký” và hệ thống sẽ hiện thông báo cho người dùng “Đăng ký thành công”.

+) Như vậy người dùng đã đăng ký tài khoản thành công với danh nghĩa mặc định là chủ nhà.

+) Tiếp theo Admin sẽ cấp lại quyền cho người đó từ quyền chủ nhà thành quyền sale.

- Đối với chủ nhà:

+) Người dùng sẽ đăng nhập vào website

+) Khi người dùng muốn đăng ký tài khoản cá nhân để đăng nhập vào website, chỉ cần nhấn nút “Đăng ký” xuất hiện ở trên cùng bên phải màn hình.

+) Sau đó hệ thống sẽ hiển thị trang đăng ký.

+) Tiếp theo người dùng sẽ điền thông tin cá nhân của người dùng vào form đăng ký. Cuối cùng sau khi điền đủ thông tin người dùng sẽ nhấn nút “Đăng ký” và hệ thống sẽ hiện thông báo cho người dùng “Đăng ký thành công”.

+) Như vậy người dùng đã đăng ký tài khoản thành công.

c. Dòng sự kiện rẽ nhánh.

- Đối với quản trị viên:

+) Khi người dùng điền các thông tin vào form đăng ký, nếu bỏ trống một trong các trường như: Email, tên người dùng, mật khẩu, xác nhận mật khẩu, số điện thoại, thẻ căn cước/CMTND thì hệ thống sẽ thông báo các lỗi tương ứng với từng field mà người dùng bỏ trống.

Ví dụ:

Trường hợp 1: Người dùng bỏ trống field tên Email, hệ thống sẽ thông báo “ Vui lòng nhập email!”.

Trường hợp 2: Người dùng bỏ trống field tên Tên người dùng, hệ thống sẽ thông báo “ Vui lòng nhập tên người dùng!”.

Trường hợp 3: Người dùng bỏ trống field tên mật khẩu, hệ thống sẽ thông báo “ Vui lòng nhập mật khẩu!” và đồng thời cũng thông báo tại field xác nhận mật khẩu “Hai mật khẩu không khớp!”.

Trường hợp 4: Người dùng bỏ trống field tên Xác nhận mật khẩu, hệ thống sẽ thông báo “ Xác nhận lại mật khẩu!”. Nếu điền sai mật khẩu hệ thống sẽ báo “Hai mật khẩu không khớp!”.

Trường hợp 5: Người dùng bỏ trống field tên Số điện thoại, hệ thống sẽ thông báo “ Vui lòng nhập số điện thoại!”.

Trường hợp 6: Người dùng nhập sai định dạng Số điện thoại, hệ thống sẽ thông báo “ Vui lòng nhập đúng định dạng số điện thoại!”.

Trường hợp 7: : Người dùng bỏ trống field tên Thẻ căn cước/CMTND, hệ thống sẽ thông báo “ Vui lòng nhập số thẻ căn cước/CMTND!”.

- Đối với sale:

+) Khi người dùng điền các thông tin vào form đăng ký, nếu bỏ trống một trong các trường như: Email, tên người dùng, mật khẩu, xác nhận mật khẩu, số điện thoại, thẻ căn cước/CMTND thì hệ thống sẽ thông báo các lỗi tương ứng với từng field mà người dùng bỏ trống.

Ví dụ:

Trường hợp 1: Người dùng bỏ trống field tên Email, hệ thống sẽ thông báo “ Vui lòng nhập email!”.

Trường hợp 2: Người dùng bỏ trống field tên Tên người dùng, hệ thống sẽ thông báo “ Vui lòng nhập tên người dùng!”.

Trường hợp 3: Người dùng bỏ trống field tên mật khẩu, hệ thống sẽ thông báo “ Vui lòng nhập mật khẩu!” và đồng thời cũng thông báo tại field xác nhận mật khẩu “Hai mật khẩu không khớp!”.

Trường hợp 4: Người dùng bỏ trống field tên Xác nhận mật khẩu, hệ thống sẽ thông báo “ Xác nhận lại mật khẩu!”. Nếu điền sai mật khẩu hệ thống sẽ báo “Hai mật khẩu không khớp!”.

Trường hợp 5: Người dùng bỏ trống field tên Số điện thoại, hệ thống sẽ thông báo “ Vui lòng nhập số điện thoại!”.

Trường hợp 6: Người dùng nhập sai định dạng Số điện thoại, hệ thống sẽ thông báo “ Vui lòng nhập đúng định dạng số điện thoại!”.

Trường hợp 7: : Người dùng bỏ trống field tên Thẻ căn cước/CMTND, hệ thống sẽ thông báo “ Vui lòng nhập số thẻ căn cước/CMTND!”.

- Đối với chủ nhà:

+) Khi người dùng điền các thông tin vào form đăng ký, nếu bỏ trống một trong các trường như: Email, tên người dùng, mật khẩu, xác nhận mật khẩu, số điện thoại, thẻ căn cước/CMTND thì hệ thống sẽ thông báo các lỗi tương ứng với từng field mà người dùng bỏ trống.

Ví dụ:

Trường hợp 1: Người dùng bỏ trống field tên Email, hệ thống sẽ thông báo “ Vui lòng nhập email!”.

Trường hợp 2: Người dùng bỏ trống field tên Tên người dùng, hệ thống sẽ thông báo “ Vui lòng nhập tên người dùng!”.

Trường hợp 3: Người dùng bỏ trống field tên mật khẩu, hệ thống sẽ thông báo “ Vui lòng nhập mật khẩu!” và đồng thời cũng thông báo tại field xác nhận mật khẩu “Hai mật khẩu không khớp!”.

Trường hợp 4: Người dùng bỏ trống field tên Xác nhận mật khẩu, hệ thống sẽ thông báo “ Xác nhận lại mật khẩu!”. Nếu điền sai mật khẩu hệ thống sẽ báo “Hai mật khẩu không khớp!”.

Trường hợp 5: Người dùng bỏ trống field tên Số điện thoại, hệ thống sẽ thông báo “ Vui lòng nhập số điện thoại!”.

Trường hợp 6: Người dùng nhập sai định dạng Số điện thoại, hệ thống sẽ thông báo “ Vui lòng nhập đúng định dạng số điện thoại!”.

Trường hợp 7: : Người dùng bỏ trống field tên Thẻ căn cước/CMTND, hệ thống sẽ thông báo “ Vui lòng nhập số thẻ căn cước/CMTND!”.

d. Điều kiện trước.

- Đối với quản trị viên:

+) Người dùng phải truy cập vào website, rồi sau đó ấn nút “Đăng ký”.

- Đối với sale:

+) Người dùng phải truy cập vào website, rồi sau đó ấn nút “Đăng ký”.

- Đối với chủ nhà:

+) Người dùng phải truy cập vào website, rồi sau đó ấn nút “Đăng ký”.

e. Dòng sự kiện sau.

- Đối với quản trị viên:

+) Sau khi đăng ký thành công, hệ thống sẽ thông báo “Đăng ký thành công!” và quay về trang đăng nhập.

+) Admin sẽ cấp lại thành quyền quản trị viên.

- Đối với sale:

+) Sau khi đăng ký thành công, hệ thống sẽ thông báo “Đăng ký thành công!” và quay về trang đăng nhập.

+) Admin sẽ cấp lại thành quyền sale.

- Đối với chủ nhà:

+) Sau khi chủ nhà đăng ký thành công, hệ thống sẽ hiện thông báo “Đăng ký thành công !” và quay về trang đăng nhập.

f. Tác nhân: Quản trị viên, Sale, Chủ nhà

*4.2.2.4. SRS004: MODULE Tìm bài đăng*

a.Mô tả nghiệp vụ:

- Chức năng tìm bài đăng cho phép người dùng tìm các căn nhà, căn hộ thông qua các bài đăng.

b. Dòng sự kiện chính.

- Người dùng sẽ truy cập vào website.

- Sau đó người dùng sẽ tìm kiếm theo “Tỉnh/thành phố”, “Quận/huyện”, “Phường/xã”, “Mức giá”, “Diện tích”, “Kiểu chỗ ở” hoặc có thể nhập từ khóa muốn tìm vào ô search rồi ấn “Tìm kiếm”.

- Tiếp theo hệ thống sẽ hiển thị các bài đăng, cho thuê các căn hộ, homestay, chung cư mà người dùng mong muốn.

c. Dòng sự kiện rẽ nhánh.

d. Điều kiện trước.

- Người dùng cần vào trang website để có thể tìm kiếm.

e. Dòng sự kiện sau.

- Người dùng sẽ truy cập vào website.

- Sau khi người dùng đã điền thông tin muốn tìm kiếm, rồi ấn “Tìm kiếm”. Hệ thống sẽ hiện ra các bài đăng cho thuê các căn hộ, homestay, chung cư,…

f. Tác nhân: Khách hàng

*4.2.2.5. SRS005: MODULE Quản lý bài đăng.*

a.Mô tả nghiệp vụ:

- Chức năng quản lý bài đăng cho phép chủ nhà quản lý các bài đăng đã post bao gồm: thêm, sửa, xóa bài đăng.

b. Dòng sự kiện chính.

- Đối với thêm:

+) Người dùng cần đăng nhập vào hệ thống dành cho chủ nhà

+) Tiếp đó người dùng chọn mục “Chủ nhà”, hệ thống sẽ hiện ra các bài đã đăng tin.

+) Người dùng chọn mục “Đăng phòng mới”, hệ thống sẽ hiện ra form đăng bài cho người dùng.

+) Người dùng sẽ điền các thông tin cần đăng bao gồm: Tiêu đề, Loại phòng, Giá tiền, Diện tích, Số phòng ngủ, Mô tả, Ảnh chi tiết, Địa chỉ.

+) Tiếp theo người dùng ấn nút “Đăng bài” để thêm mới bài đăng.

- Đối với sửa:

+) Người dùng cần đăng nhập vào hệ thống dành cho chủ nhà.

+) Tiếp đó người dùng chọn mục “Chủ nhà”, hệ thống sẽ hiện ra các bài đã đăng tin.

+) Mục “Danh sách phòng đã đăng” sẽ hiện ra các bài chủ nhà đã đăng.

+) Người dùng click “Chỉnh sửa”, hệ thống sẽ hiện trang “thông tin phòng” bao gồm các thông tin cũ. Sau đó người dùng sẽ cập nhật các trường nào muốn thay đổi và ấn nút “Cập nhật”.

+) Hệ thống sẽ cập nhật lại thông tin mới của bài đăng và thông báo “Cập nhật thông tin thành công!” và quay lại mục “Danh sách phòng đã đăng”.

- Đối với xóa:

+) Người dùng cần đăng nhập vào hệ thống dành cho chủ nhà.

+) Người dùng click vào nút “Xóa”, hệ thống sẽ hiện thông báo “Bạn có thực sự muốn xóa phòng này?”. Người dùng click “OK” để xóa bài đăng và hệ thống báo “Xóa phòng thành công!”.

c. Dòng sự kiện rẽ nhánh.

- Đối với thêm:

+) Người dùng cần đăng nhập vào hệ thống dành cho chủ nhà

+) Tiếp đó người dùng chọn mục “Chủ nhà”, hệ thống sẽ hiện ra các bài đã đăng tin.

+) Người dùng chọn mục “Đăng phòng mới”, hệ thống sẽ hiện ra form đăng bài cho người dùng.

+) Người dùng sẽ điền các thông tin cần đăng bao gồm: Tiêu đề, Loại phòng, Giá tiền, Diện tích, Số phòng ngủ, Mô tả, Ảnh chi tiết, Địa chỉ.

+) Nếu người dùng bỏ trống không nhập các field trong form thì các thông báo tại các field sẽ hiện cho người dùng biết.

+) Nếu người dùng bỏ trống “Tiêu đề”, hệ thống sẽ báo “Vui lòng nhập tiêu đề bài đăng!”.

+) Nếu người dùng bỏ trống “Loại phòng”, hệ thống sẽ báo “Vui lòng lựa chọn loại phòng!”.

+) Nếu người dùng bỏ trống “Giá tiền”, hệ thống sẽ báo “Vui lòng nhập giá tiền!”.

+) Nếu người dùng bỏ trống “Diện tích”, hệ thống sẽ báo “Vui lòng nhập diện tích!”.

+) Nếu người dùng bỏ trống “Số phòng ngủ”, hệ thống sẽ báo “Vui lòng nhập số phòng ngủ!”.

+) Nếu người dùng bỏ trống “Mô tả”, hệ thống sẽ báo “Vui lòng nhập mô tả bài đăng!”.

+) Nếu người dùng không chọn “Ảnh chi tiết”, hệ thống sẽ báo “Vui lòng chọn ảnh!”.

+) Nếu người dùng không chọn “Địa chỉ”, hệ thống sẽ báo “Vui lòng chọn Tỉnh/thành phố!”, “Vui lòng chọn Quận/huyện!”, “Vui lòng chọn Phường/xã!”.

- Đối với sửa:

+) Người dùng cần đăng nhập vào hệ thống dành cho chủ nhà.

+) Tiếp đó người dùng chọn mục “Chủ nhà”, hệ thống sẽ hiện ra các bài đã đăng tin.

+) Mục “Danh sách phòng đã đăng” sẽ hiện ra các bài chủ nhà đã đăng.

+) Người dùng click nút “Chỉnh sửa”, hệ thống sẽ hiện trang “thông tin phòng” bao gồm các thông tin cũ. Sau đó người dùng sẽ cập nhật các trường nào muốn thay đổi và ấn nút “Cập nhật”.

+) Nếu người dùng xóa các thông tin cũ tại các trường thì hệ thống sẽ thông báo để người dùng không được bỏ trống.

- Đối với xóa:

+) Người dùng cần đăng nhập vào hệ thống dành cho chủ nhà.

+) Người dùng click vào nút “Xóa”, hệ thống sẽ hiện thông báo “Bạn có thực sự muốn xóa phòng này?”.

+) Nếu người dùng click chọn “Hủy”, hệ thống sẽ quay lại danh sách các bài đã đăng.

d. Điều kiện trước.

- Người dùng cần đăng nhập vào trang dành cho chủ nhà.

e. Dòng sự kiện sau.

- Sau khi người dùng thêm, sửa, xóa bài đăng, hệ thống sẽ quay lại trang “Danh sách phòng đã đăng”.

f. Tác nhân: chủ nhà

*4.2.2.6. SRS005: MODULE Quản lý bài viết.*

a.Mô tả nghiệp vụ:

- Chức năng quản lý bài viết cho phép quản trị viên quản lý các bài viết về vấn đề thuê, mua chung cư

b. Dòng sự kiện chính.

- *Đối với quản trị viên*:

- Với thêm bài viết:

+) Người dùng cần đăng nhập vào trang dành cho quản trị viên.

+) Sau đó chọn mục “Quản lý bài viết”, hệ thống sẽ hiện ra danh sách các bài viết đã đăng.

+) Tiếp theo ấn nút “Thêm” màu xanh trên cùng bên phải màn hình, hệ thống sẽ chuyển tới form “Thêm mới bài đăng”. Tại đây người dùng cần nhập các thông tin như: Tiêu đề, tag, ảnh đại diện và nội dung.

+) Tiếp theo ấn nút “Thêm” để thêm mới bài đăng vào danh sách và hệ thống sẽ thông báo “Đăng bài thành công!”

- Với sửa bài viết:

+) Người dùng cần đăng nhập vào trang dành cho quản trị viên.

+) Sau đó chọn mục “Quản lý bài viết”, hệ thống sẽ hiện ra danh sách các bài viết đã đăng.

+) Tiếp theo click nút “Chỉnh sửa”, hệ thống sẽ hiện form “Cập nhật thông tin bài đăng”. Người dùng sẽ cập nhật lại thông tin các trường muốn thay đổi và ấn nút “Cập nhật”.

+) Sau đó hệ thống sẽ báo “Cập nhật thành công!” và quay lại danh sách bài đăng.

- Với xóa bài viết:

+) Người dùng cần đăng nhập vào trang dành cho quản trị viên.

+) Sau đó chọn mục “Quản lý bài viết”, hệ thống sẽ hiện ra danh sách các bài viết đã đăng.

+) Tiếp theo người dùng ấn nút “Xóa”, hệ thống sẽ thông báo “Xóa bài này?”. Người dùng click chọn OK để xác nhận xóa bài viết và hệ thống sẽ báo “Xóa bài thành công!” và quay về danh sách bài đăng.

- *Đối với sale*:

- Với thêm bài viết:

+) Người dùng cần đăng nhập vào trang dành quản trị dành cho sale.

+) Sau đó chọn mục “Quản lý bài viết”, hệ thống sẽ hiện ra danh sách bài đăng.

+) Tiếp theo ấn nút “Thêm” màu xanh trên cùng bên phải màn hình, hệ thống sẽ chuyển tới form “Thêm mới bài đăng”. Tại đây người dùng cần nhập các thông tin như: Tiêu đề, tag, ảnh đại diện và nội dung.

+) Tiếp theo ấn nút “Thêm” để thêm mới bài đăng vào danh sách và hệ thống sẽ thông báo “Đăng bài thành công!”

- Với sửa bài viết:

+) Người dùng cần đăng nhập vào trang quản trị dành cho sale.

+) Sau đó chọn mục “Quản lý bài viết”, hệ thống sẽ hiện ra danh sách bài đăng.

+) Tiếp theo click nút “Chỉnh sửa”, hệ thống sẽ hiện form “Cập nhật thông tin bài đăng”. Người dùng sẽ cập nhật lại thông tin các trường muốn thay đổi và ấn nút “Cập nhật”.

+) Sau đó hệ thống sẽ báo “Cập nhật thành công!” và quay lại danh sách bài đăng.

- Với xóa bài viết:

+) Người dùng cần đăng nhập vào trang quản trị dành cho sale.

+) Sau đó chọn mục “Quản lý bài viết”, hệ thống sẽ hiện ra danh sách các bài viết đã đăng.

+) Tiếp theo người dùng ấn nút “Xóa”, hệ thống sẽ thông báo “Xóa bài này?”. Người dùng click chọn OK để xác nhận xóa bài viết và hệ thống sẽ báo “Xóa bài thành công!” và quay về danh sách bài đăng.

c. Dòng sự kiện rẽ nhánh.

- *Đối với quản trị viên:*

- Với thêm bài viết:

+) Người dùng cần đăng nhập vào trang dành cho quản trị viên.

+) Sau đó chọn mục “Quản lý bài viết”, hệ thống sẽ hiện ra danh sách các bài viết đã đăng.

+) Tiếp theo ấn nút “Thêm” màu xanh trên cùng bên phải màn hình, hệ thống sẽ chuyển tới form “Thêm mới bài đăng”. Tại đây người dùng cần nhập các thông tin như: Tiêu đề, tag, ảnh đại diện và nội dung.

+) Nếu người dùng bỏ trống không nhập tất cả, hệ thống sẽ thông báo lỗi theo từng field.

+) Nếu người dùng bỏ trống tiêu đề, hệ thống sẽ thông báo “Vui lòng nhập tiêu đề!”.

+) Nếu người dùng không chọn ảnh đại diện, hệ thống sẽ thông báo “Vui lòng chọn ảnh!”.

+) Nếu người dùng bỏ trống nội dung, hệ thống sẽ thông báo “Vui lòng nhập nội dung!”.

- Với sửa bài viết:

+) Người dùng cần đăng nhập vào trang dành cho quản trị viên.

+) Sau đó chọn mục “Quản lý bài viết”, hệ thống sẽ hiện ra danh sách các bài viết đã đăng.

+) Tiếp theo click nút “Chỉnh sửa”, hệ thống sẽ hiện form “Cập nhật thông tin bài đăng”. Người dùng sẽ cập nhật lại thông tin các trường muốn thay đổi và ấn nút “Cập nhật”.

+) Nếu người dùng xóa các thông tin cũ của bài viết, hệ thống sẽ báo lỗi tương ứng với các field và yêu cầu người dùng phải điền đầy đủ thông tin.

+) Nếu người dùng xóa tiêu đề cũ mà không điền tiêu đề mới, hệ thống sẽ hiển thị thông báo “Vui lòng nhập tiêu đề!”.

+) Nếu người dùng bỏ ảnh đại diện cũ và không thay ảnh mới, hệ thống sẽ báo “Vui lòng chọn ảnh!”.

+) Nếu người dùng xóa nội dung cũ và không nhập nội dung mới hệ thống sẽ báo “Vui lòng nhập nội dung!”.

- *Đối với sale:*

- Với thêm bài viết:

+) Người dùng cần đăng nhập vào trang quản trị dành cho sale.

+) Sau đó chọn mục “Quản lý bài viết”, hệ thống sẽ hiện ra danh sách các bài viết đã đăng.

+) Tiếp theo ấn nút “Thêm” màu xanh trên cùng bên phải màn hình, hệ thống sẽ chuyển tới form “Thêm mới bài đăng”. Tại đây người dùng cần nhập các thông tin như: Tiêu đề, tag, ảnh đại diện và nội dung.

+) Nếu người dùng bỏ trống không nhập tất cả, hệ thống sẽ thông báo lỗi theo từng field.

+) Nếu người dùng bỏ trống tiêu đề, hệ thống sẽ thông báo “Vui lòng nhập tiêu đề!”.

+) Nếu người dùng không chọn ảnh đại diện, hệ thống sẽ thông báo “Vui lòng chọn ảnh!”.

+) Nếu người dùng bỏ trống nội dung, hệ thống sẽ thông báo “Vui lòng nhập nội dung!”.

- Với sửa bài viết:

+) Người dùng cần đăng nhập vào trang quản trị dành cho sale.

+) Sau đó chọn mục “Quản lý bài viết”, hệ thống sẽ hiện ra danh sách các bài viết đã đăng.

+) Tiếp theo click nút “Chỉnh sửa”, hệ thống sẽ hiện form “Cập nhật thông tin bài đăng”. Người dùng sẽ cập nhật lại thông tin các trường muốn thay đổi và ấn nút “Cập nhật”.

+) Nếu người dùng xóa các thông tin cũ của bài viết, hệ thống sẽ báo lỗi tương ứng với các field và yêu cầu người dùng phải điền đầy đủ thông tin.

+) Nếu người dùng xóa tiêu đề cũ mà không điền tiêu đề mới, hệ thống sẽ hiển thị thông báo “Vui lòng nhập tiêu đề!”.

+) Nếu người dùng bỏ ảnh đại diện cũ và không thay ảnh mới, hệ thống sẽ báo “Vui lòng chọn ảnh!”.

+) Nếu người dùng xóa nội dung cũ và không nhập nội dung mới hệ thống sẽ báo “Vui lòng nhập nội dung!”.

d. Điều kiện trước.

- Quản trị viên sẽ đăng nhập vào trang quản trị dành cho quản trị viên.

- Sale sẽ đăng nhập vào trang quản trị dành cho sale

e. Dòng sự kiện sau.

- Sau khi thêm bài viết hệ thống sẽ quay lại trang danh sách bài đăng.

- Sau khi sửa bài viết hệ thống sẽ quay lại trtang danh sách bài đăng với thông tin đã được cập nhật.

- Sau khi xóa bài viết hệ thống sẽ quay lại trang danh sách bài đăng.

f. Tác nhân: Quản trị viên, sale

*4.2.2.6. SRS006: MODULE người dùng*

a.Mô tả nghiệp vụ:

- Chức năng người dùng cho phép quản trị viên xem danh sách người dùng mới đăng ký và sửa quyền cho người dùng.

b. Dòng sự kiện chính.

- Quản trị viên đăng nhập vào trang quản trị dành cho quản trị viên.

- Tiếp theo chọn mục “Người dùng”, hệ thống sẽ hiển thị danh sách chủ nhà chưa assign cho sale và danh sách sale.

c. Dòng sự kiện rẽ nhánh.

- Đối với danh sách chủ nhà chưa assign cho sale:

+) Đây là danh sách các người dùng mới đăng ký sẽ hiện tại đây.

+) Nếu quản trị muốn đổi quyền tên người dùng thì click vào sửa quyền chủ nhà đó gồm: quyền admin, quyền chủ nhà, quyền sale.

- Đối với danh sách sale:

+) Đây là danh sách người sale sẽ quản lý các chủ nhà khi đăng bài. Từ đây người sale sẽ kiểm tra được thông tin những khách hàng đăng ký nhận thông tin từ bài đăng kèm theo chủ nhà của bài đăng đó.

+) Quản trị sẽ sửa quyền cho những ai tên chủ nhà đăng ký tên là: sale là “Sale” bằng cách click vào “Sửa quyền”.

+) Sau đó tên người sale đó sẽ chuyển sang “danh sách sale”.

+) Tại mục “Assign chủ nhà”, quản trị viên sẽ click “Chọn chủ nhà” để thêm các chủ nhà được assign cho Sale đó.

d. Điều kiện trước.

- Quản trị viên cần đăng nhập vào trang quản trị thành công.

e. Dòng sự kiện sau.

f. Tác nhân: Quản trị viên

*4.2.2.7. SRS007: MODULE Quản lý phòng*

a.Mô tả nghiệp vụ:

- Chức năng quản lý phòng cho phép quản trị viên và Sale có thể thêm, sửa, xóa các phòng, căn hộ đang được đăng cho thuê.

b. Dòng sự kiện chính.

- *Đối với quản trị viên:*

- Quản trị viên đăng nhập vào trang quản trị.

- Sau đó chọn mục “Quản lý phòng”, hệ thống sẽ hiện ra danh sách phòng đang được đăng.

- Với thêm phòng:

+) Người dùng đăng click chọn “Thêm phòng” phía trên bên phải màn hình, hệ thống hiện ra form “Thêm phòng mới” cho người dùng nhập.

+) Người dùng sẽ điền các thông tin cần đăng bao gồm: Tiêu đề, Loại phòng, Giá tiền, Diện tích, Số phòng ngủ, Mô tả, Ảnh chi tiết, Địa chỉ.

+) Tiếp theo người dùng ấn nút “Thêm” để thêm mới và quay lại danh sách phòng

- Với sửa phòng:

+) Người dùng click “Chi tiết”, hệ thống sẽ hiện trang “Thông tin chi tiết” bao gồm các thông tin cũ. Sau đó người dùng sẽ cập nhật các trường nào muốn thay đổi và ấn nút “Cập nhật”.

+) Tiếp đó, hệ thống sẽ cập nhật lại thông tin mới của bài đăng và thông báo “Cập nhật thông tin thành công!” và quay lại danh sách phòng.

- Với xóa phòng:

+) Người dùng click nút “Xóa”, hệ thống sẽ hiện thông báo “Xóa phòng này?”. Tiếp theo ấn “Yes” để xác nhận xóa và hiện thông báo “Xóa phòng thành công!”. Sau khi xóa hệ thống quay về danh sách phòng.

- *Đối với Sale*:

- Sale đăng nhập vào trang quản trị dành cho Sale

- Sau đó chọn mục “Quản lý phòng”, hệ thống sẽ hiện ra danh sách phòng đang được đăng.

- Với thêm phòng:

+) Người dùng đăng click chọn “Thêm phòng” phía trên bên phải màn hình, hệ thống hiện ra form “Thêm phòng mới” cho người dùng nhập.

+) Người dùng sẽ điền các thông tin cần đăng bao gồm: Tiêu đề, Loại phòng, Giá tiền, Diện tích, Số phòng ngủ, Mô tả, Ảnh chi tiết, Địa chỉ.

+) Tiếp theo người dùng ấn nút “Thêm” để thêm mới và quay lại danh sách phòng

- Với sửa phòng:

+) Người dùng click “Chi tiết”, hệ thống sẽ hiện trang “Thông tin chi tiết” bao gồm các thông tin cũ. Sau đó người dùng sẽ cập nhật các trường nào muốn thay đổi và ấn nút “Cập nhật”.

+) Tiếp đó, hệ thống sẽ cập nhật lại thông tin mới của bài đăng và thông báo “Cập nhật thông tin thành công!” và quay lại danh sách phòng.

- Với xóa phòng:

+) Người dùng click nút “Xóa”, hệ thống sẽ hiện thông báo “Xóa phòng này?”. Tiếp theo ấn “Yes” để xác nhận xóa và hiện thông báo “Xóa phòng thành công!”. Sau khi xóa hệ thống quay về danh sách phòng.

c. Dòng sự kiện rẽ nhánh.

- *Đối với quản trị viên*:

- Quản trị viên đăng nhập vào trang quản trị.

- Sau đó chọn mục “Quản lý phòng”, hệ thống sẽ hiện ra danh sách phòng đang được đăng.

- Với thêm phòng:

+) Người dùng đăng click chọn “Thêm phòng” phía trên bên phải màn hình, hệ thống hiện ra form “Thêm phòng mới” cho người dùng nhập.

+) Người dùng sẽ điền các thông tin cần đăng bao gồm: Tiêu đề, Loại phòng, Giá tiền, Diện tích, Số phòng ngủ, Mô tả, Ảnh chi tiết, Địa chỉ.

+) Nếu người dùng bỏ trống không nhập các field trong form thì các thông báo tại các field sẽ hiện cho người dùng biết.

+) Nếu người dùng bỏ trống “Tiêu đề”, hệ thống sẽ báo “Vui lòng nhập tiêu đề bài đăng!”.

+) Nếu người dùng bỏ trống “Loại phòng”, hệ thống sẽ báo “Vui lòng lựa chọn loại phòng!”.

+) Nếu người dùng bỏ trống “Giá tiền”, hệ thống sẽ báo “Vui lòng nhập giá tiền!” và “Vui lòng nhập đơn vị!”.

+) Nếu người dùng bỏ trống “Diện tích”, hệ thống sẽ báo “Vui lòng nhập diện tích!”.

+) Nếu người dùng bỏ trống “Số phòng ngủ”, hệ thống sẽ báo “Vui lòng nhập số phòng ngủ!”.

+) Nếu người dùng bỏ trống “Mô tả”, hệ thống sẽ báo “Vui lòng nhập mô tả bài đăng!”.

+) Nếu người dùng không chọn “Ảnh chi tiết”, hệ thống sẽ báo “Vui lòng chọn ảnh!”.

+) Nếu người dùng không chọn “Địa chỉ”, hệ thống sẽ báo “Vui lòng chọn Tỉnh/thành phố!”, “Vui lòng chọn Quận/huyện!”, “Vui lòng chọn Phường/xã!”.

- Với sửa phòng:

+) Người dùng click “Chi tiết”, hệ thống sẽ hiện trang “Thông tin chi tiết” bao gồm các thông tin cũ. Sau đó người dùng sẽ cập nhật các trường nào muốn thay đổi và ấn nút “Cập nhật”.

+) Nếu người dùng xóa các thông tin cũ tại các trường mà không nhập thông tin mới thì hệ thống sẽ thông báo để người dùng không được bỏ trống tương ứng với các field.

+) Nếu xóa nội dung tiêu đề cũ và không nhập tiêu đề mới, hệ thống sẽ báo “Vui lòng nhập tiêu đề bài đăng!”.

+) Nếu xóa nội dung loại phòng cũ và không nhập loại phòng mới, hệ thống sẽ báo “Vui lòng lựa chọn loại phòng!”.

+) Nếu xóa nội dung giá tiền cũ và không nhập giá tiền mới, hệ thống sẽ báo “Vui lòng nhập giá tiền!”.

+) Nếu xóa nội dung diện tích cũ và không nhập diện tích mới, hệ thống sẽ báo “Vui lòng nhập diện tích!”.

+) Nếu xóa nội dung số phòng ngủ cũ và không nhập số phòng ngủ mới, hệ thống sẽ báo “Vui lòng nhập số phòng ngủ!”.

+) Nếu xóa nội dung mô tả cũ và không nhập mô tả mới, hệ thống sẽ báo “Vui lòng nhập mô tả bài đăng!”.

+) Nếu xóa ảnh chi tiết cũ và không thay bằng ảnh chi tiết mới, hệ thống sẽ báo “Vui lòng đăng ảnh chi tiết!”.

- Với xóa phòng:

+) Người dùng click chọn “Xóa”.

+) Hệ thống sẽ hiện thông báo “Xóa phòng này?”

+) Nếu người dùng chọn “No”, hệ thống vẫn ở lại trang danh sách phòng.

- *Đối với Sale*:

- Sale đăng nhập vào trang quản trị.

- Sau đó chọn mục “Quản lý phòng”, hệ thống sẽ hiện ra danh sách phòng đang được đăng.

- Với thêm phòng:

+) Người dùng đăng click chọn “Thêm phòng” phía trên bên phải màn hình, hệ thống hiện ra form “Thêm phòng mới” cho người dùng nhập.

+) Người dùng sẽ điền các thông tin cần đăng bao gồm: Tiêu đề, Loại phòng, Giá tiền, Diện tích, Số phòng ngủ, Mô tả, Ảnh chi tiết, Địa chỉ.

+) Nếu người dùng bỏ trống không nhập các field trong form thì các thông báo tại các field sẽ hiện cho người dùng biết.

+) Nếu người dùng bỏ trống “Tiêu đề”, hệ thống sẽ báo “Vui lòng nhập tiêu đề bài đăng!”.

+) Nếu người dùng bỏ trống “Loại phòng”, hệ thống sẽ báo “Vui lòng lựa chọn loại phòng!”.

+) Nếu người dùng bỏ trống “Giá tiền”, hệ thống sẽ báo “Vui lòng nhập giá tiền!” và “Vui lòng nhập đơn vị!”.

+) Nếu người dùng bỏ trống “Diện tích”, hệ thống sẽ báo “Vui lòng nhập diện tích!”.

+) Nếu người dùng bỏ trống “Số phòng ngủ”, hệ thống sẽ báo “Vui lòng nhập số phòng ngủ!”.

+) Nếu người dùng bỏ trống “Mô tả”, hệ thống sẽ báo “Vui lòng nhập mô tả bài đăng!”.

+) Nếu người dùng không chọn “Ảnh chi tiết”, hệ thống sẽ báo “Vui lòng chọn ảnh!”.

+) Nếu người dùng không chọn “Địa chỉ”, hệ thống sẽ báo “Vui lòng chọn Tỉnh/thành phố!”, “Vui lòng chọn Quận/huyện!”, “Vui lòng chọn Phường/xã!”.

- Với sửa phòng:

+) Người dùng click “Chi tiết”, hệ thống sẽ hiện trang “Thông tin chi tiết” bao gồm các thông tin cũ. Sau đó người dùng sẽ cập nhật các trường nào muốn thay đổi và ấn nút “Cập nhật”.

+) Nếu người dùng xóa các thông tin cũ tại các trường mà không nhập thông tin mới thì hệ thống sẽ thông báo để người dùng không được bỏ trống tương ứng với các field.

+) Nếu xóa nội dung tiêu đề cũ và không nhập tiêu đề mới, hệ thống sẽ báo “Vui lòng nhập tiêu đề bài đăng!”.

+) Nếu xóa nội dung loại phòng cũ và không nhập loại phòng mới, hệ thống sẽ báo “Vui lòng lựa chọn loại phòng!”.

+) Nếu xóa nội dung giá tiền cũ và không nhập giá tiền mới, hệ thống sẽ báo “Vui lòng nhập giá tiền!”.

+) Nếu xóa nội dung diện tích cũ và không nhập diện tích mới, hệ thống sẽ báo “Vui lòng nhập diện tích!”.

+) Nếu xóa nội dung số phòng ngủ cũ và không nhập số phòng ngủ mới, hệ thống sẽ báo “Vui lòng nhập số phòng ngủ!”.

+) Nếu xóa nội dung mô tả cũ và không nhập mô tả mới, hệ thống sẽ báo “Vui lòng nhập mô tả bài đăng!”.

+) Nếu xóa ảnh chi tiết cũ và không thay bằng ảnh chi tiết mới, hệ thống sẽ báo “Vui lòng đăng ảnh chi tiết!”.

- Với xóa phòng:

+) Người dùng click chọn “Xóa”.

+) Hệ thống sẽ hiện thông báo “Xóa phòng này?”

+) Nếu người dùng chọn “No”, hệ thống vẫn ở lại trang danh sách phòng.

d. Điều kiện trước.

- Quản trị viên cần phải đăng nhập vào trang quản trị dành cho quản trị.

- Sale cần phải đăng nhập vào trang quản trị dành cho Sale.

e. Dòng sự kiện sau.

- Sau khi thêm phòng, hệ thống sẽ quay lại danh sách phòng.

- Sau khi cập nhật phòng, hệ thống sẽ quay lại danh sách phòng với thông tin mới.

- Sau khi xóa phòng, hệ thống sẽ quay lại danh sách phòng.

f. Tác nhân: Quản trị viên, Sale

*4.2.2.8. SRS008: MODULE Quản lý loại phòng*

a.Mô tả nghiệp vụ:

- Chức năng quản lý loại phòng cho phép quản trị viên quản lý các loại phòng bao gồm: Thêm, sửa, xóa loại phòng.

b. Dòng sự kiện chính.

- Đối với thêm loại phòng

+) Hệ thống hiển thị danh sách các loại phòng.

+) Muốn thêm loại phòng, ấn nút “Thêm”. Sau đó hệ thống sẽ hiển thị trang “Thêm mới loại phòng” cho quản trị viên điền thông tin.

+) Tiếp theo quản trị viên điền đầy đủ các thông tin bao gồm: Loại phòng, mô tả, và upload ảnh.

+) Sau khi điền xong, ấn nút “Thêm” để thêm mới loại phòng vào website và hệ thống sẽ thông báo “Thêm mới loại phòng thành công!”

- Đối với sửa loại phòng

+) Hệ thống hiển thị danh sách các loại phòng.

+) Muốn sửa, ấn nút “Chỉnh sửa” màu xanh. Sau đó hệ thống sẽ hiển thị trang “Cập nhật thông tin loại phòng” cho quản trị viên sửa đổi.

+) Tiếp theo quản trị viên sẽ nhập các giá trị mới vào từng field để cập nhật dữ liệu mới.

+) Sau khi cập nhật các trường, ấn nút “Cập nhật” để lưu các thông tin mới và hệ thống sẽ thông báo “Cập nhật thành công!”.

- Đối với xóa loại phòng

+) Hệ thống hiển thị danh sách các loại phòng.

+) Muốn xóa loại phòng, ấn nút “Xóa” màu cam. Sau đó hệ thống sẽ thông báo cho quản trị viên “Xóa loại phòng này ?”.

+) Nếu người dùng đồng ý xóa thì chọn nút “OK” và hệ thống sẽ thông báo “Xóa loại phòng thành công !”.

+) Nếu người dùng không muốn xóa loại phòng thì chọn nút “Hủy” và hệ thống vẫn sẽ ở lại giao diện danh sách loại phòng.

c. Dòng sự kiện rẽ nhánh.

- Đối với thêm loại phòng:

+) Hệ thống hiển thị danh sách các loại phòng.

+) Muốn thêm loại phòng, ấn nút “Thêm”. Sau đó hệ thống sẽ hiển thị trang “Thêm mới loại phòng” cho quản trị viên điền thông tin.

+) Nếu người dùng bỏ trống field Loại phòng, mô tả, và không upload ảnh, hệ thống sẽ thông báo “Vui lòng nhập tên loại phòng!” tại field loại phòng, “Vui lòng nhập mô tả!” tại field mô tả, “Vui lòng đăng ảnh!” tại ô upload.

+) Nếu người dùng nhập tên loại phòng và mô tả nhưng không upload ảnh, hệ thống sẽ báo “Vui lòng đăng ảnh!”.

+) Nếu người dùng nhập tên loại phòng và upload ảnh nhưng không điền mô tả, hệ thống sẽ báo “Vui lòng nhập mô tả!”.

+) Nếu người dùng nhập mô tả và upload ảnh nhưng không điền loại phòng, hệ thống sẽ báo “Vui lòng nhập tên loại phòng!”.

- Đối với sửa loại phòng:

+) Hệ thống hiển thị danh sách các loại phòng.

+) Muốn sửa, ấn nút “Chỉnh sửa” màu xanh. Sau đó hệ thống sẽ hiển thị trang “Cập nhật thông tin loại phòng” cho quản trị viên sửa đổi.

+) Tiếp theo quản trị viên sẽ nhập các giá trị mới vào từng field để cập nhật dữ liệu mới.

+) Nếu người dùng xóa dữ liệu cũ và bỏ trống field loại phòng, hệ thống sẽ thông báo “Vui lòng nhập tên loại phòng!”.

+) Nếu người dùng xóa dữ liệu cũ và bỏ trống field mô tả, hệ thống sẽ thông báo “Vui lòng nhập mô tả!”.

+) Nếu người dùng xóa ảnh cũ và bỏ trống, hệ thống sẽ thông báo “Vui lòng đăng ảnh!”.

+) Nếu người dùng vẫn giữ dữ liệu cũ mà ấn cập nhật, hệ thống sẽ báo “No data has been change” và vẫn giữ các giá trị cũ.

d. Điều kiện trước.

- Quản trị viên sẽ đăng nhập thành công vào trang quản trị.

e. Dòng sự kiện sau.

- Sau khi thêm loại phòng thành công hệ thống sẽ cập nhật lại danh sách loại phòng mới nhất.

- Sau khi sửa loại phòng thành công hệ thống sẽ cập nhật lại danh sách loại phòng mới nhất.

- Sau khi xóa loại phòng thành công hệ thống sẽ cập nhật lại danh sách loại phòng mới nhất.

f. Tác nhân: Quản trị viên

*4.2.2.9. SRS009: MODULE Quản lý banner*

a.Mô tả nghiệp vụ:

- Chức năng quản lý banner cho phép quản trị viên thay đổi ảnh nền giao diện trang chủ.

b. Dòng sự kiện chính.

- Hệ thống hiển thị danh sách các ảnh nền.

- Quản trị viên sẽ click button trạng thái ở từng ảnh để có cho phép hiển thị ngoài trang chủ không.

- Đối với thêm:

+) Hệ thống hiển thị danh sách các ảnh nền

+) Quản trị viên muốn thêm một ảnh mới, ấn nút “Thêm”, tiếp đó hệ thống sẽ hiển thị form “Thêm mới banner” gồm tên và phần upload ảnh.

+) Sau khi quản trị viên điền đủ thông tin và chọn xong ảnh, ấn nút “Thêm” để thêm mới và hệ thống sẽ hiện thông báo “Thêm mới banner thành công!”.

c. Dòng sự kiện rẽ nhánh.

- Đối với thêm:

+) Hệ thống hiển thị danh sách các ảnh nền.

+) Nếu quản trị viên điền tên nhưng không upload ảnh, hệ thống sẽ thông báo “Vui lòng đăng ảnh!”.

d. Điều kiện trước.

- Quản trị viên đăng nhập thành công vào trang dành cho quản trị.

e. Dòng sự kiện sau.

- Sau khi click trạng thái đổi sang màu xanh thì ảnh đó đã được hiển thị, và hệ thông báo “Thay đổi trạng thái thành công!”.

- Nếu click trạng thái đổi sang màu xám thì ảnh đó không được hiển thị, và hệ thông báo “Thay đổi trạng thái thành công!”.

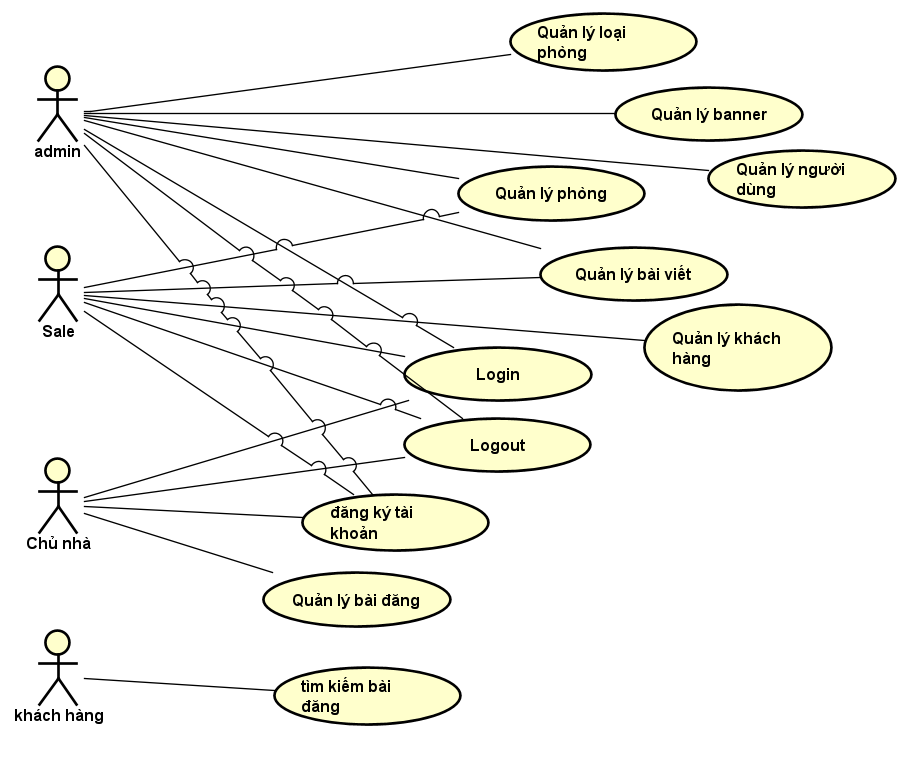
- Đối với thêm:

+) Sau khi quản trị viên điền đủ thông tin và chọn xong ảnh, ấn nút “Thêm” để thêm mới và hệ thống sẽ hiện thông báo “Thêm mới banner thành công!”.

f. Tác nhân: Quản trị viên

## **4.3.** **Thiết kế hệ thống**

### ***4.3.1.*** ***Biểu đồ Use case***



Hình 4. 1: Use case tổng quát

Diagram

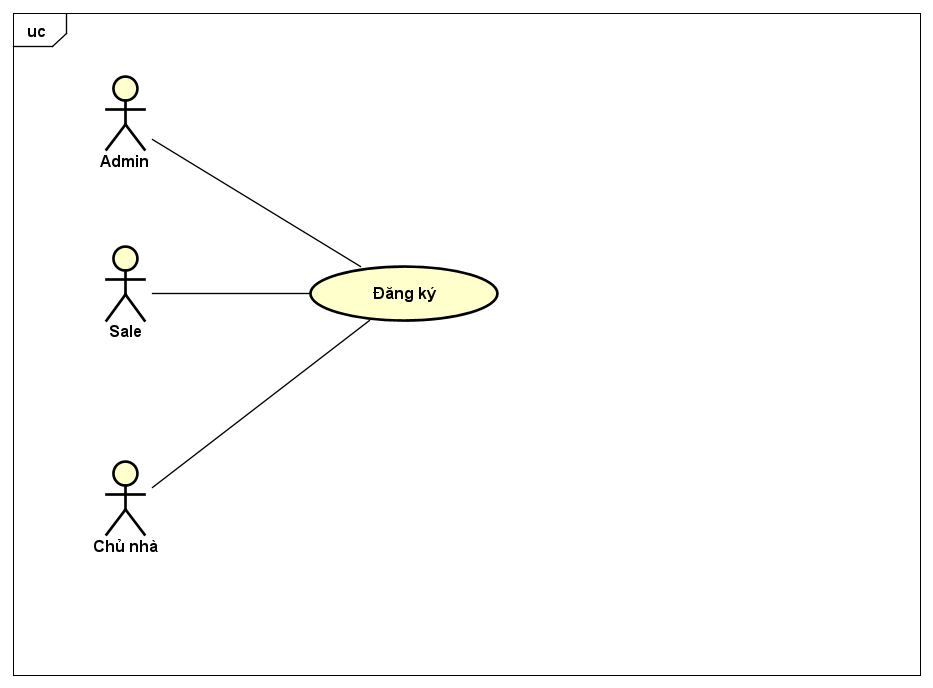
Description automatically generated

Hình 4. 2: Use case chi tiết

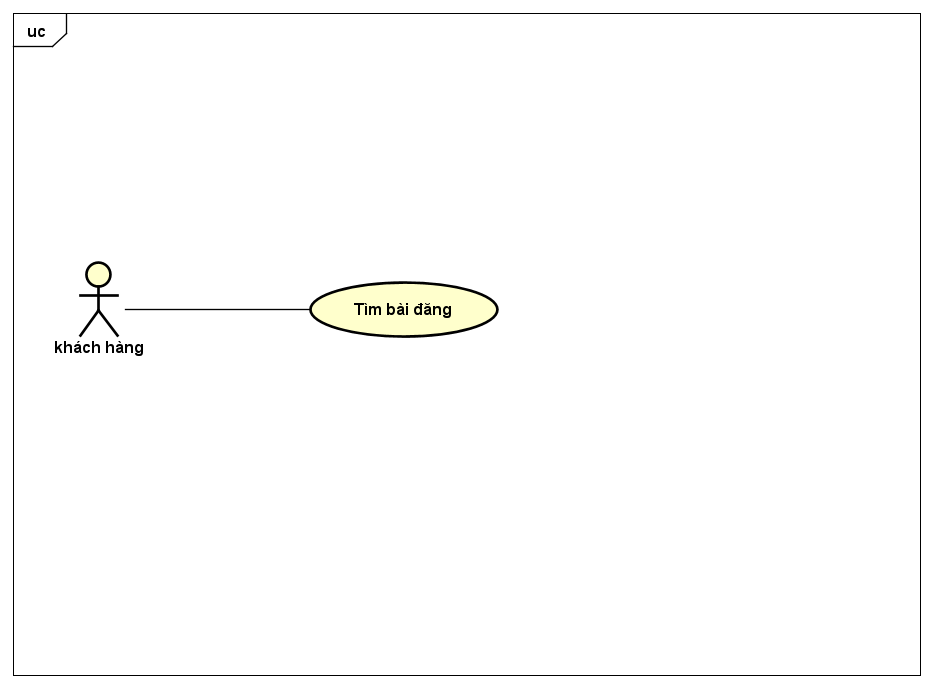
Diagram

Description automatically generated

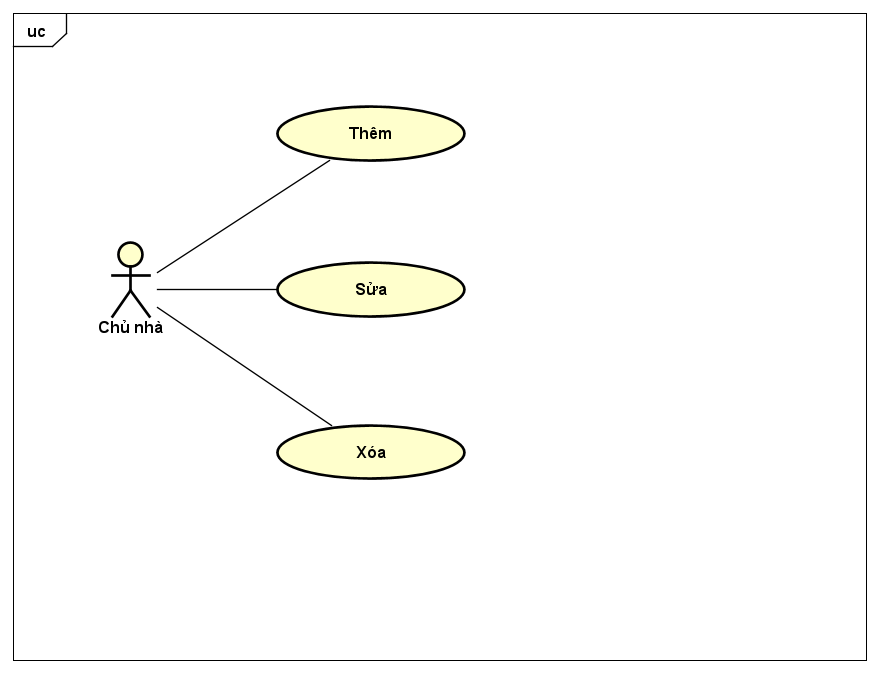
Hình 4. 3: Use case đăng nhập và đăng xuất



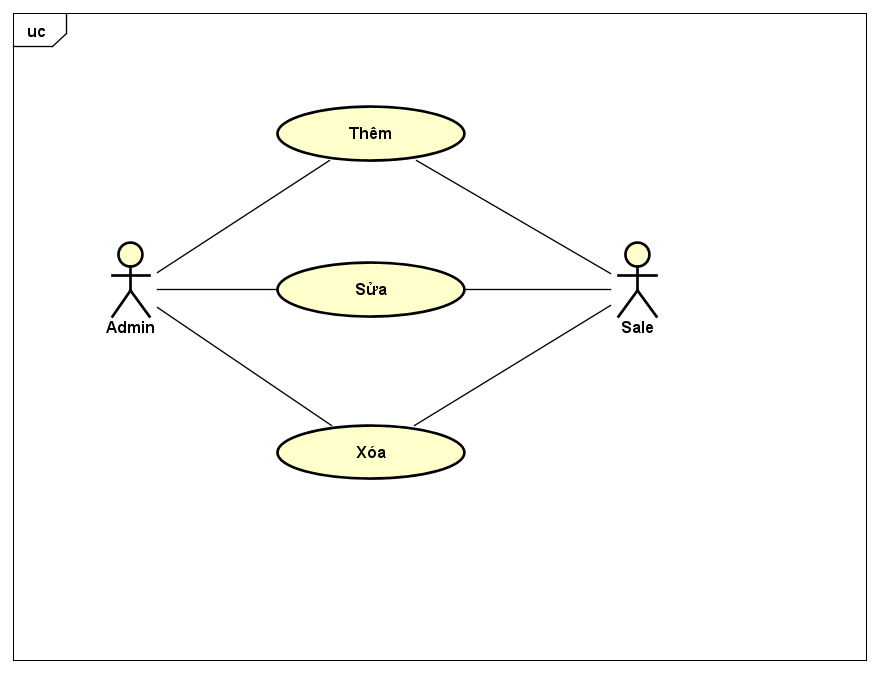
Hình 4. 3: Use case đăng ký



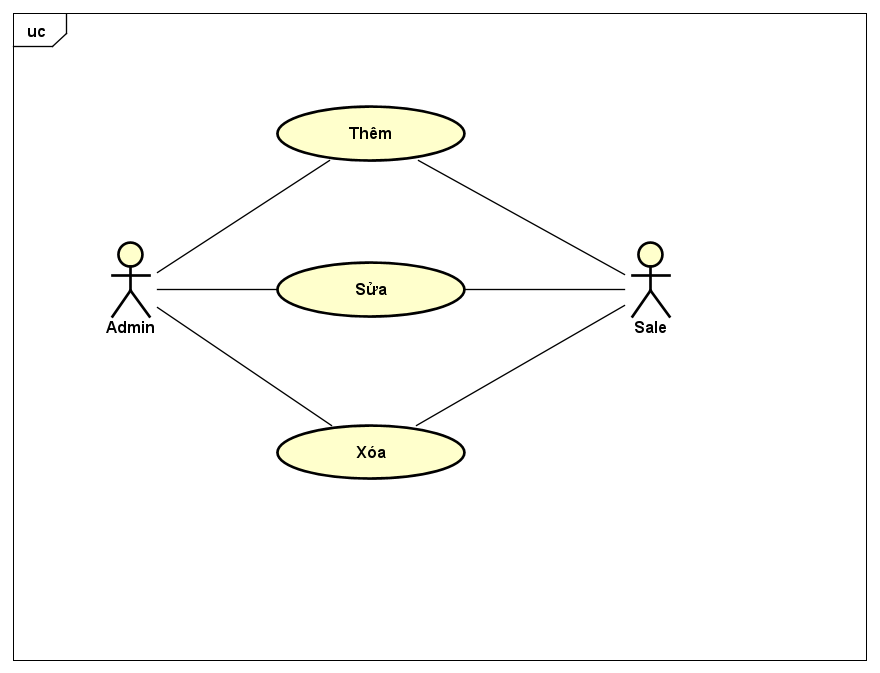
Hình 4. 4: Use case tìm bài đăng



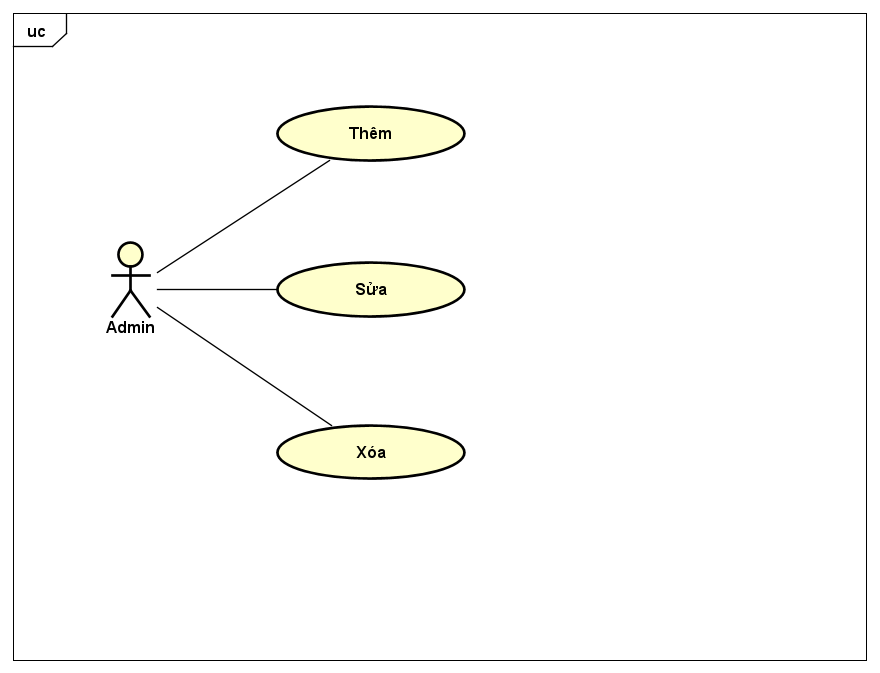
Hình 4. 5: Use case quản lý bài đăng



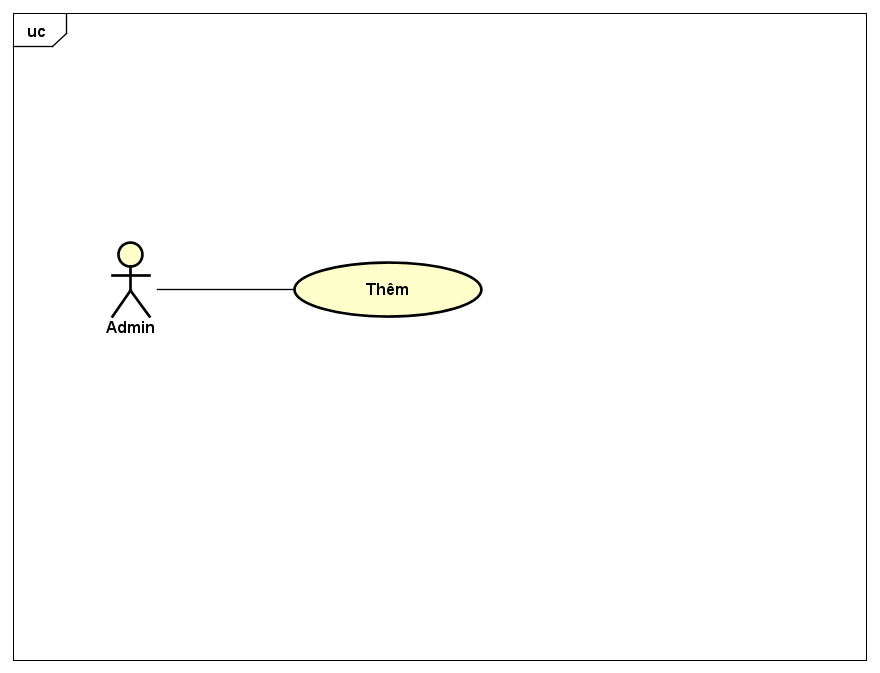
Hình 4. 6: Use case quản lý bài viết



Hình 4. 7: Use case quản lý phòng

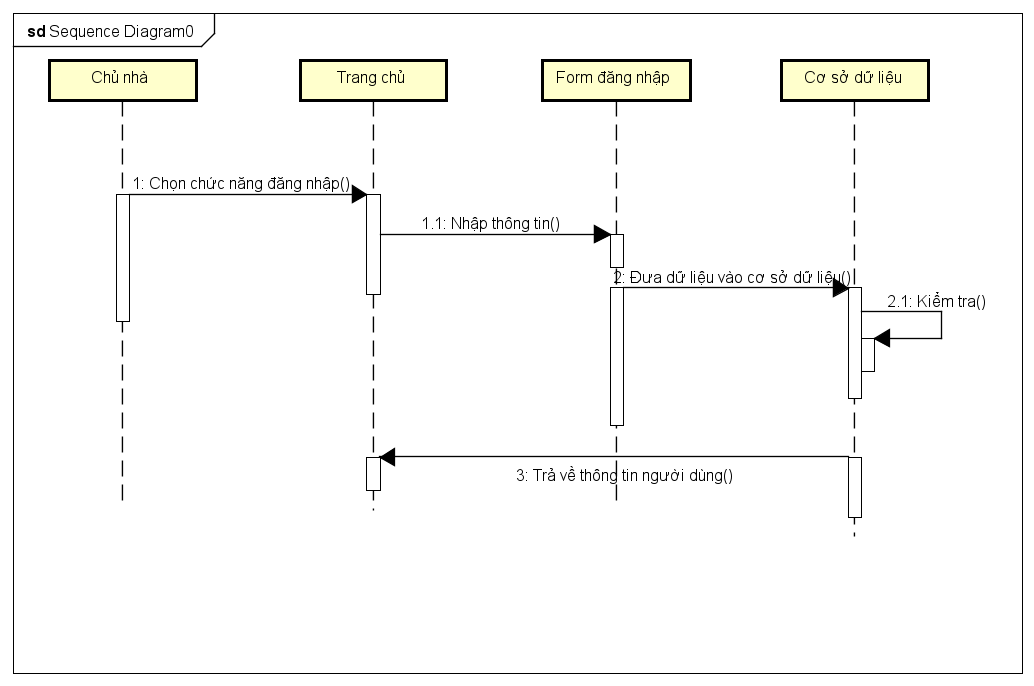


Hình 4. 8: Use case quản lý loại phòng

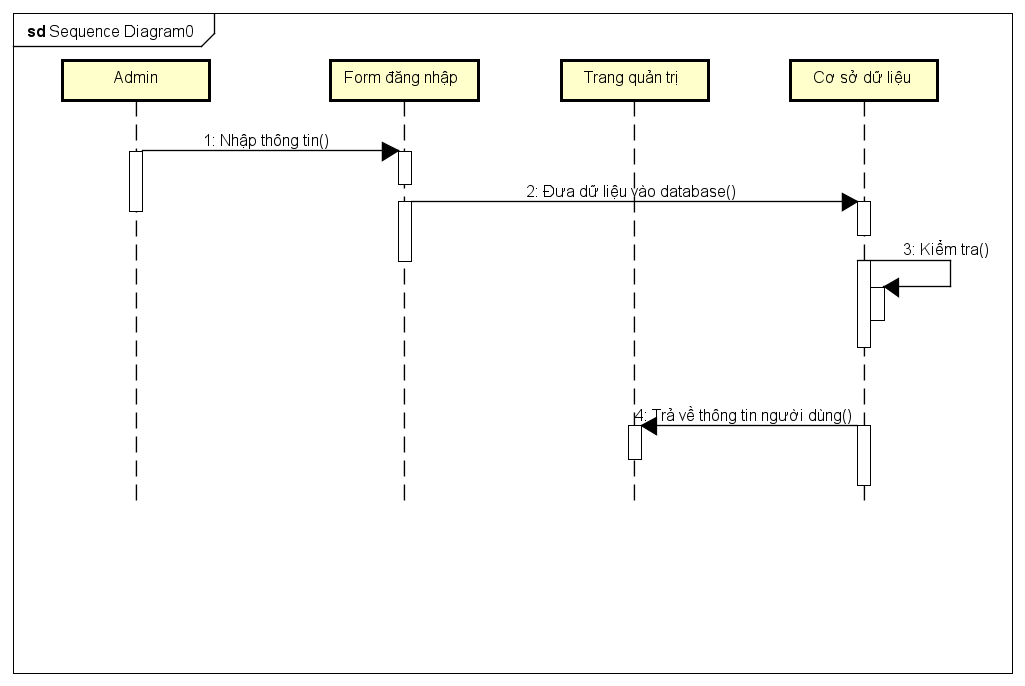


Hình 4. 9: Use case quản lý ảnh bìa

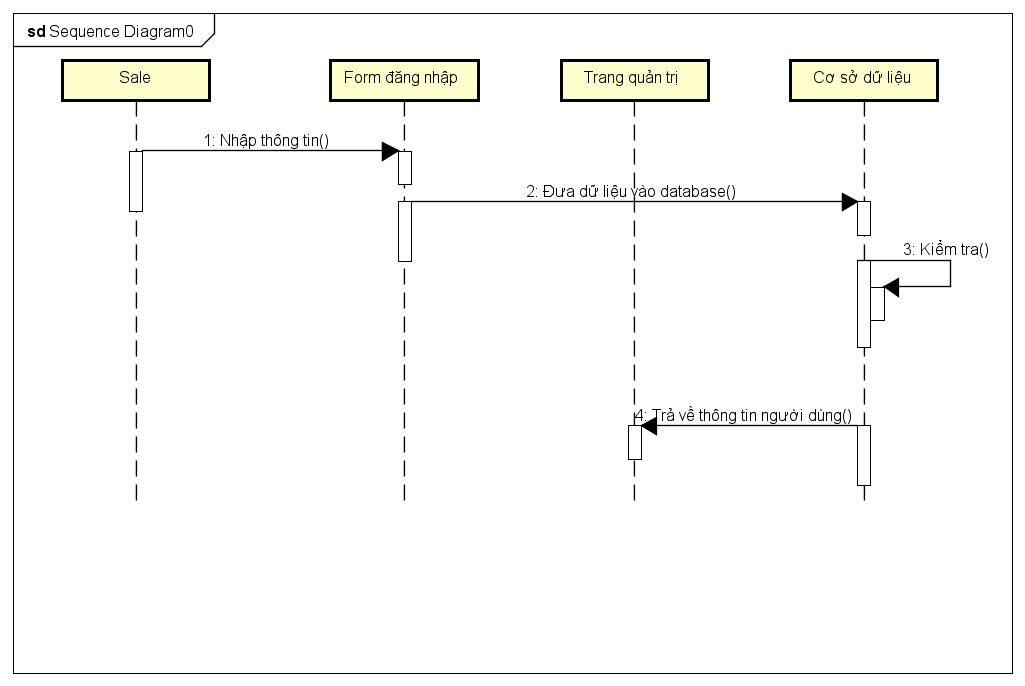
### ***4.3.2. Sơ đồ trình tự***



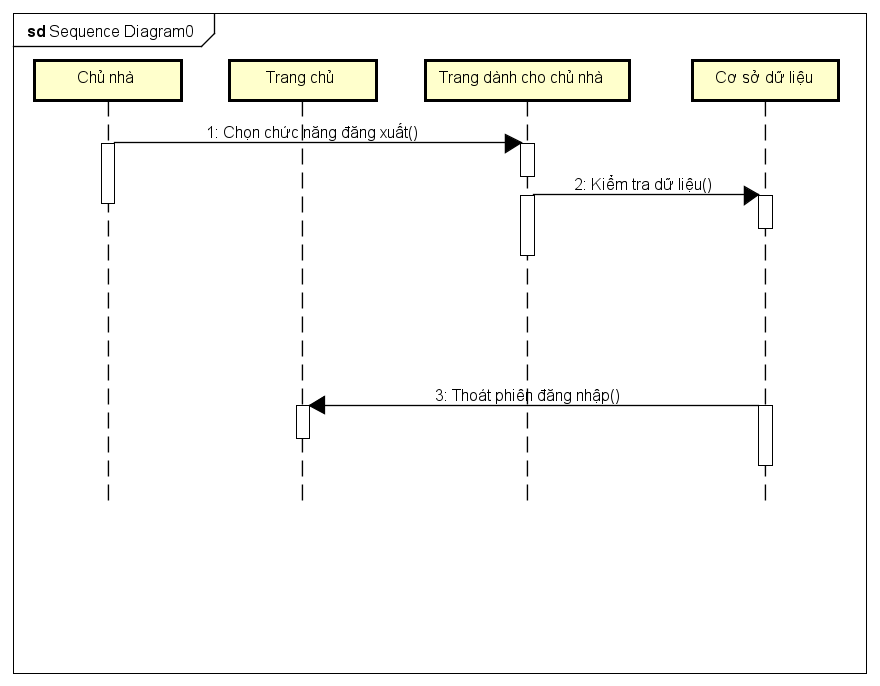
Hình 4. 10: Sơ đồ trình tự cho chức năng đăng nhập (chủ nhà)



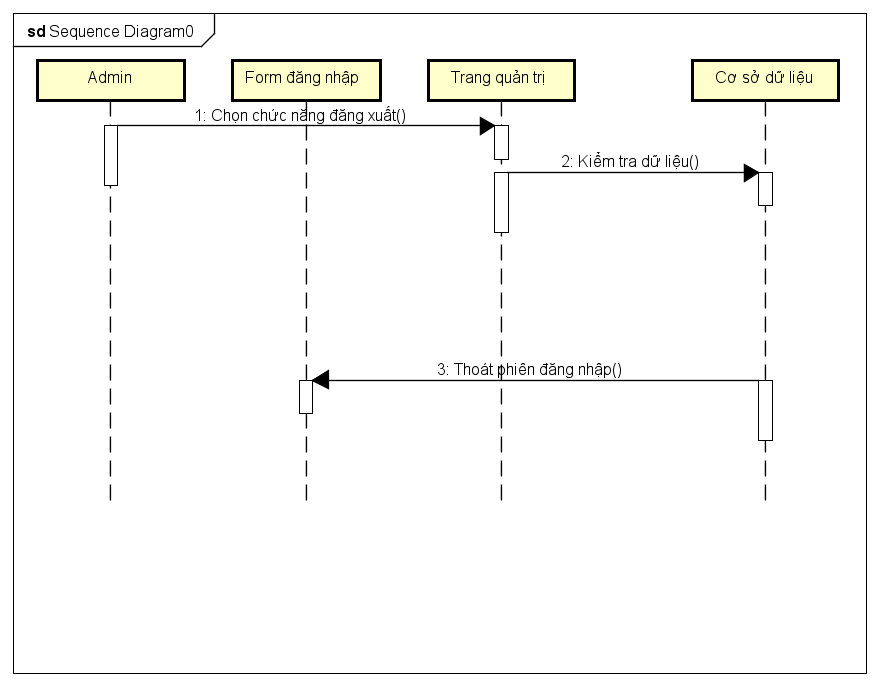
Hình 4. 11: Sơ đồ trình tự cho chức năng đăng nhập (Quản trị viên)



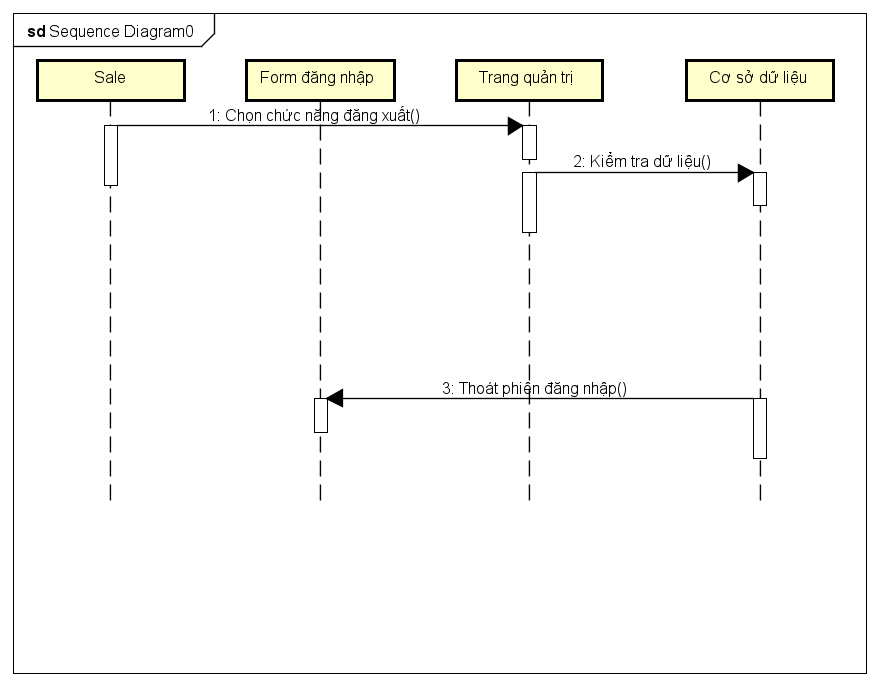
Hình 4. 12: Sơ đồ trình tự cho chức năng đăng nhập (Sale)



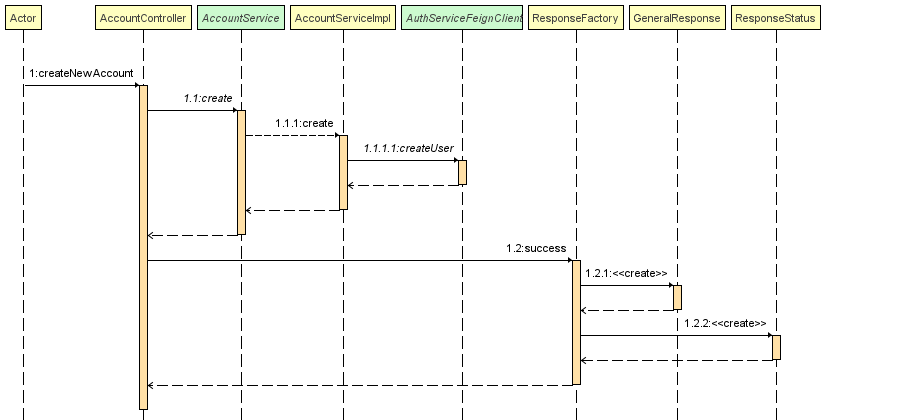
Hình 4. 13: Sơ đồ trình tự cho chức năng đăng xuất (Chủ nhà)



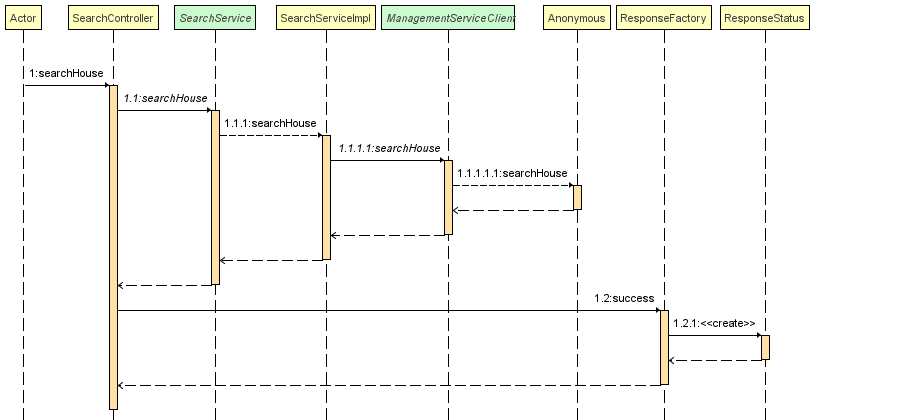
Hình 4. 14: Sơ đồ trình tự cho chức năng đăng xuất (Admin)



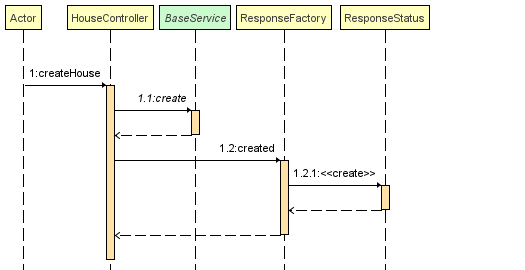
Hình 4. 15: Sơ đồ trình tự cho chức năng đăng xuất (Sale)



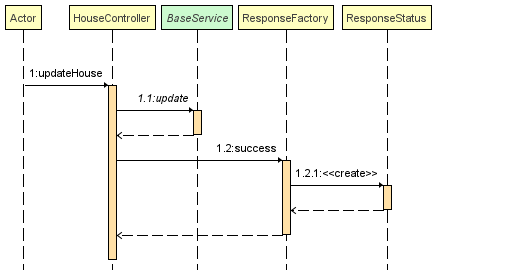
Hình 4. 16: Sơ đồ trình tự cho chức năng đăng ký



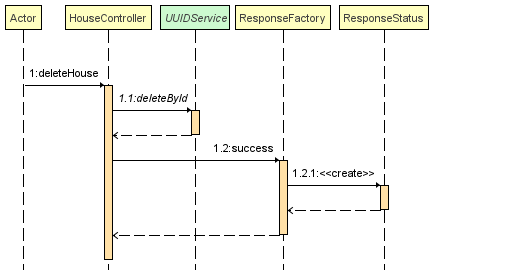
Hình 4. 17: Sơ đồ trình tự cho chức năng tìm bài đăng



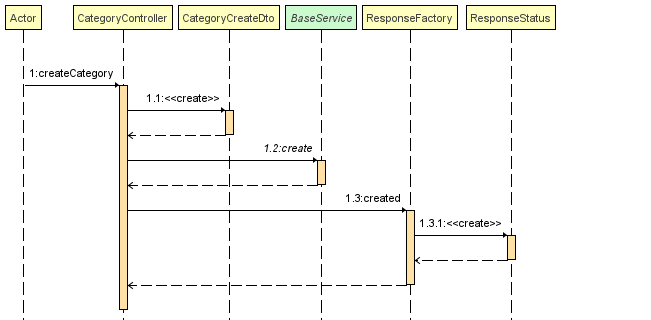
Hình 4. 18: Sơ đồ trình tự cho chức năng thêm bài đăng



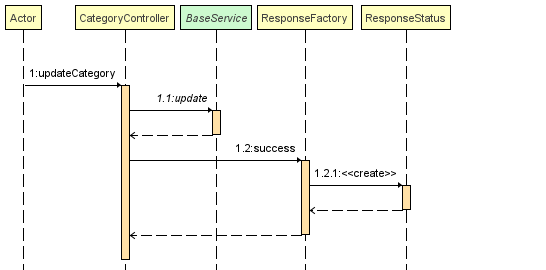
Hình 4. 19: Sơ đồ trình tự cho chức năng sửa bài đăng



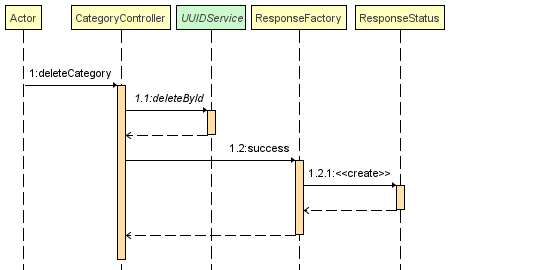
Hình 4. 20: Sơ đồ trình tự cho chức năng xóa bài đăng



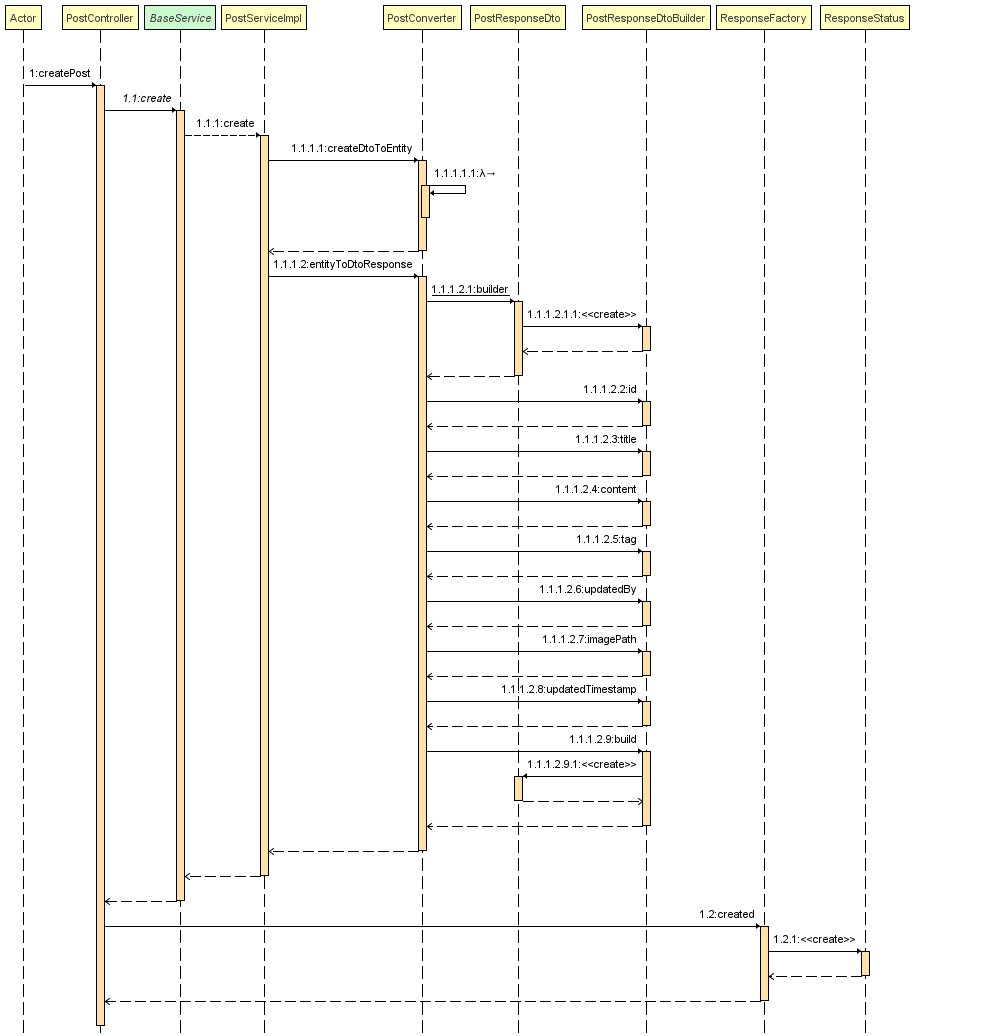
Hình 4. 21: Sơ đồ trình tự cho chức năng thêm loại phòng



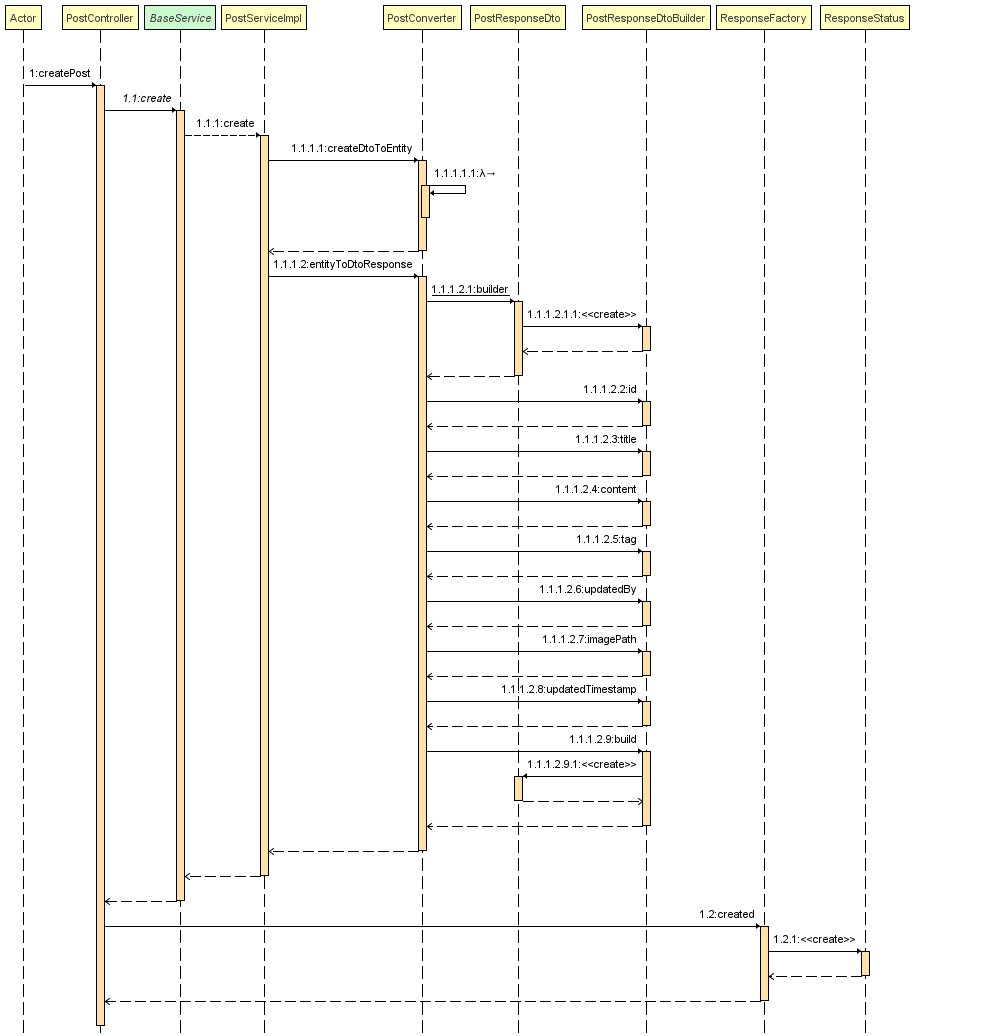
Hình 4. 22: Sơ đồ trình tự cho chức năng sửa loại phòng



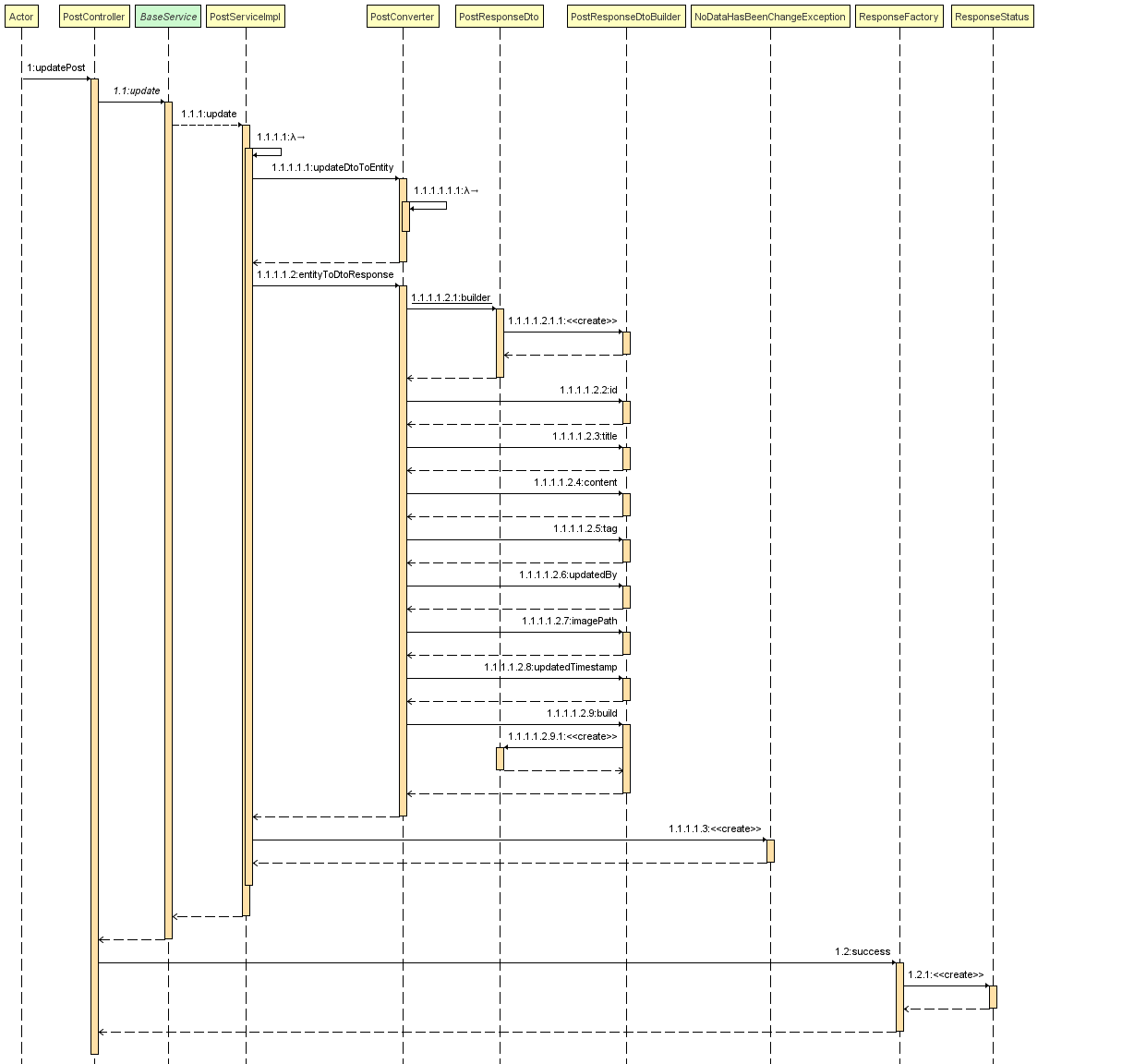
Hình 4. 23: Sơ đồ trình tự cho chức năng xóa loại phòng



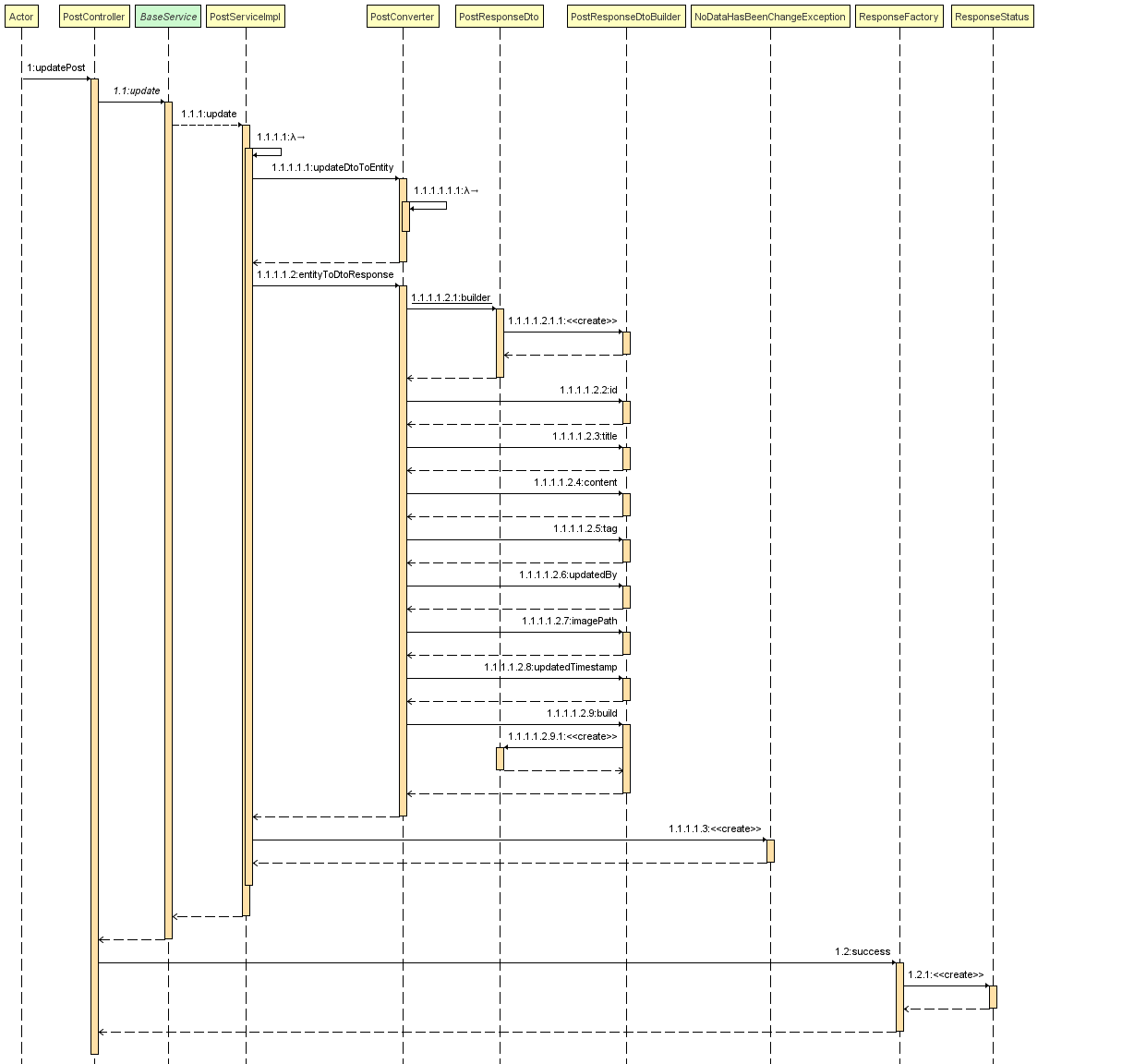
Hình 4. 24: Sơ đồ trình tự cho chức năng thêm bài viết (Admin)



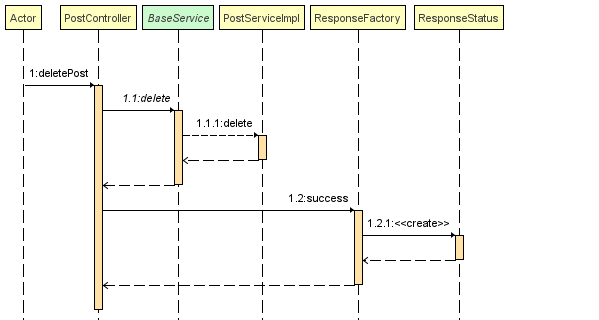
Hình 4. 25: Sơ đồ trình tự cho chức năng thêm bài viết (Sale)



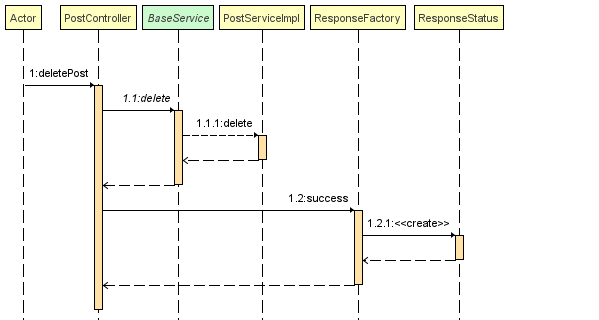
Hình 4. 26: Sơ đồ trình tự cho chức năng sửa bài viết (Admin)



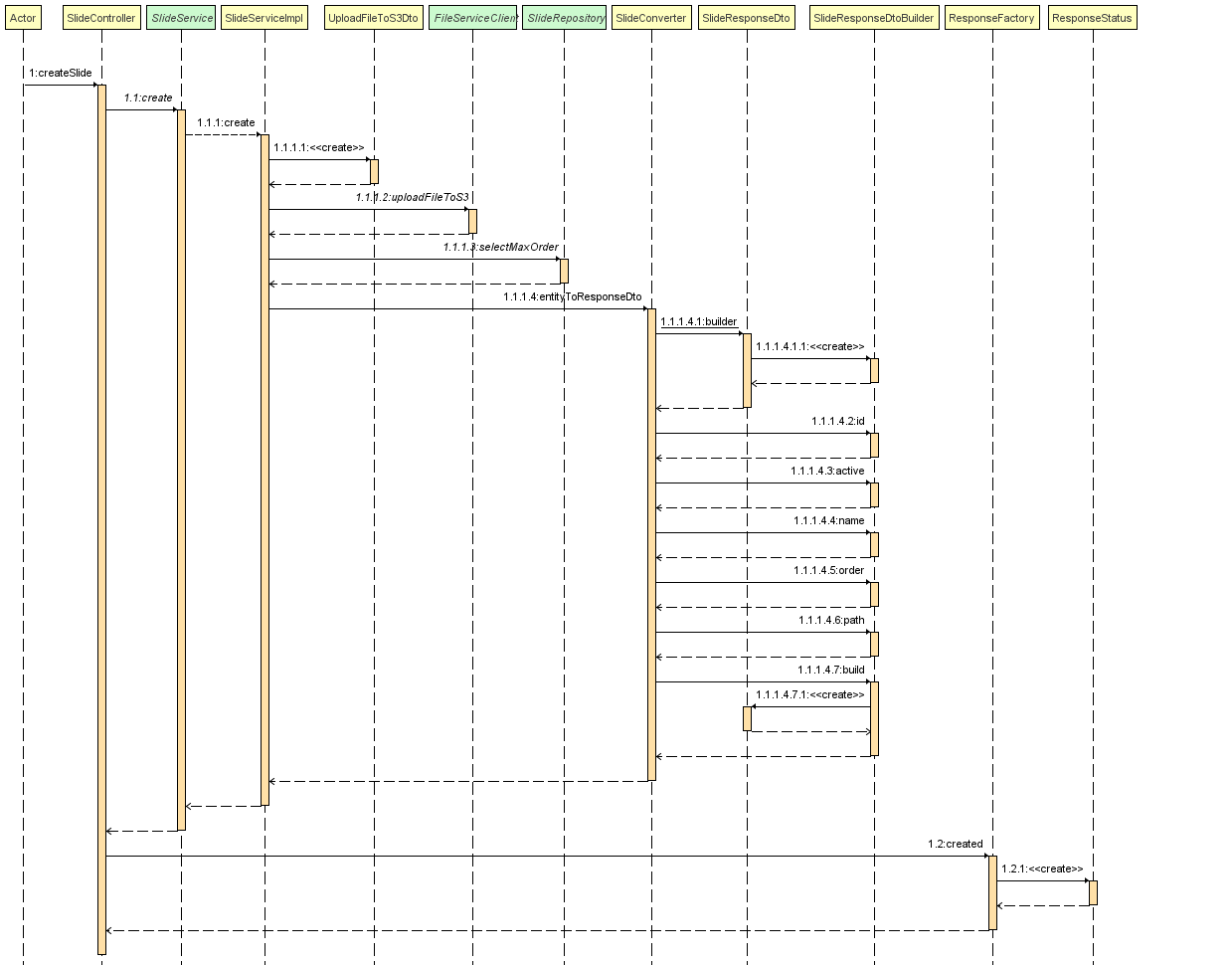
Hình 4. 27: Sơ đồ trình tự cho chức năng sửa bài viết (Sale)



Hình 4. 28: Sơ đồ trình tự cho chức năng xóa bài viết (Admin)

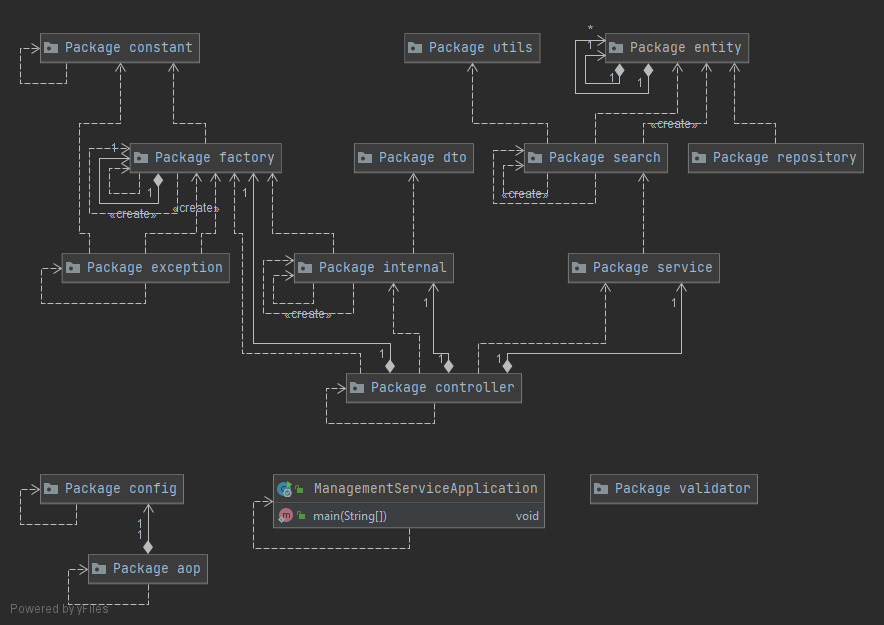


Hình 4. 29: Sơ đồ trình tự cho chức năng xóa bài viết (Sale)

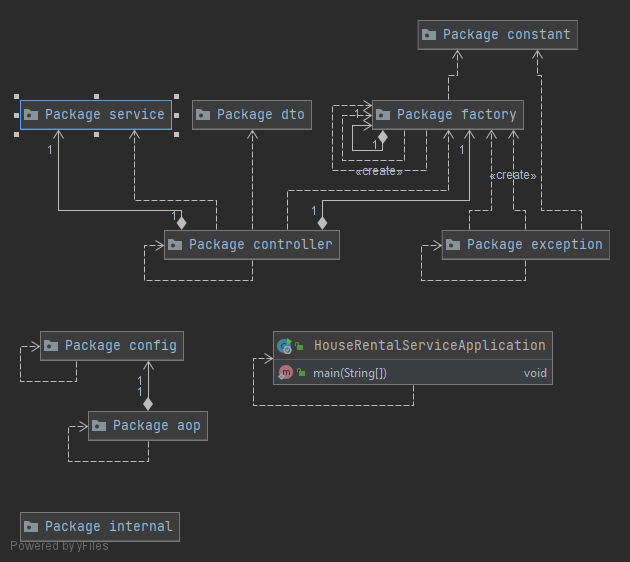


Hình 4. 30: Sơ đồ trình tự cho chức năng thêm ảnh bìa

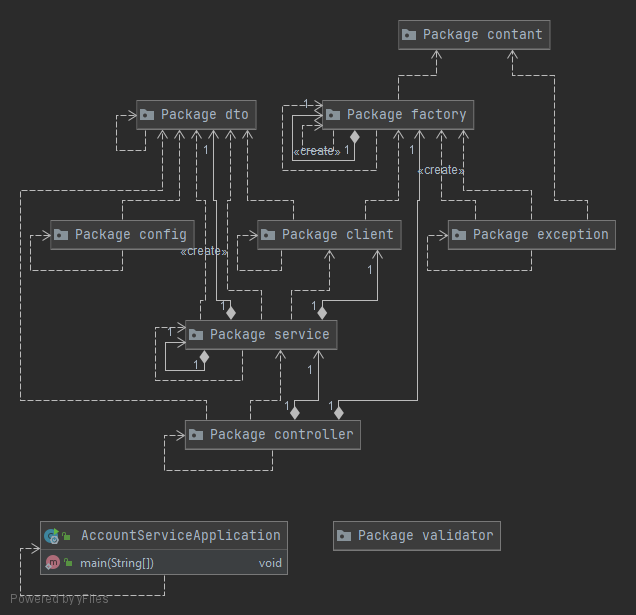
### ***4.3.3. Sơ đồ lớp***



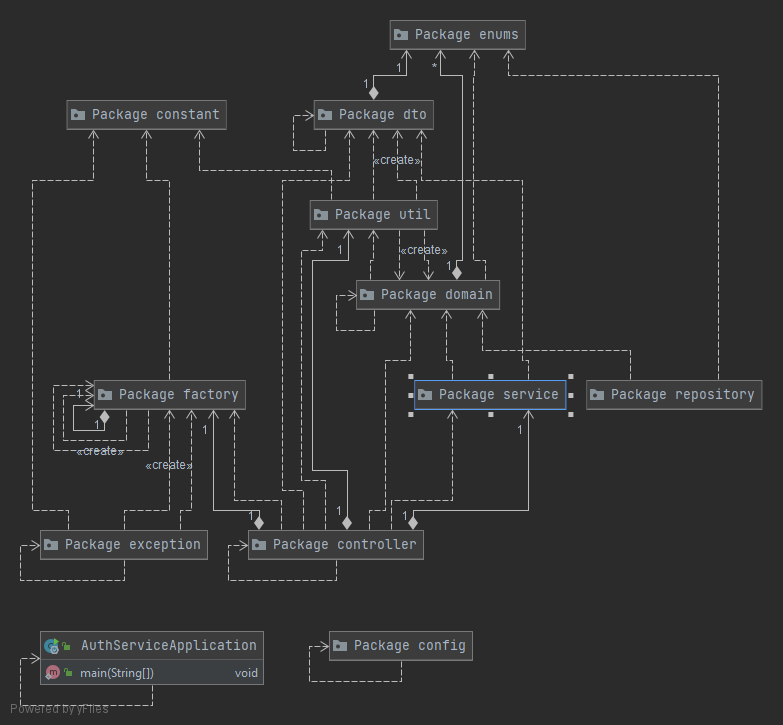
Hình 4. 31: Sơ đồ tổng quát các package của ManagerService



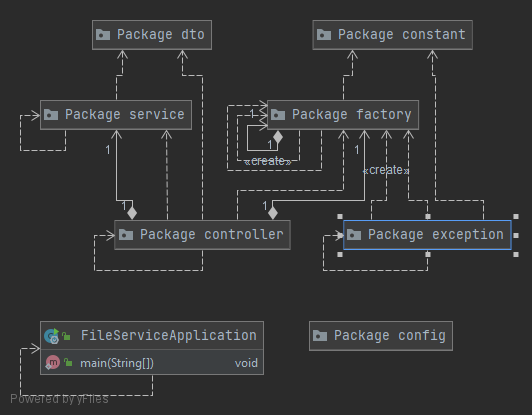
Hình 4. 32: Sơ đồ tổng quát các package của HouseRentalService



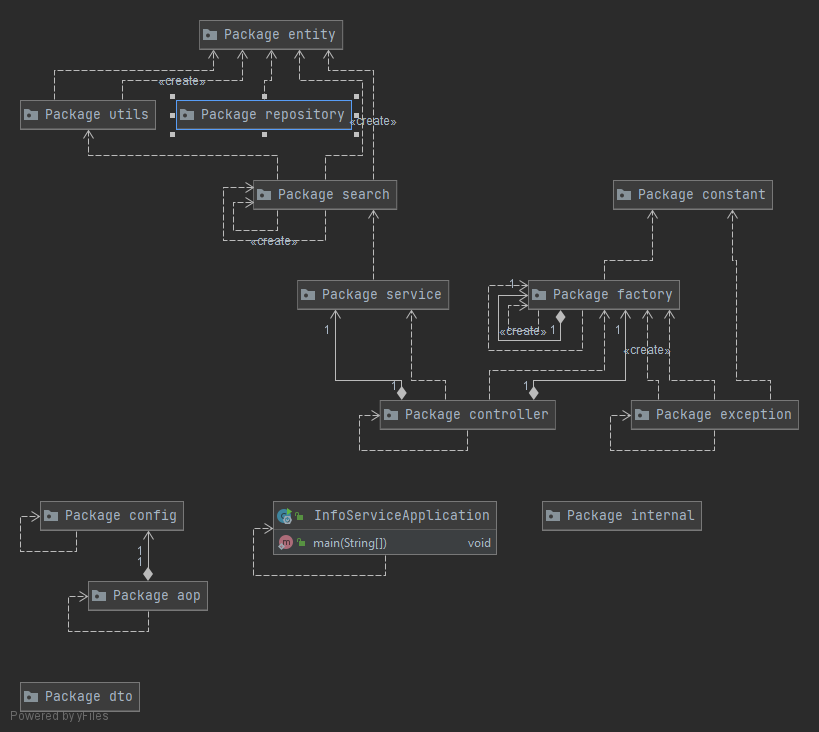
Hình 4. 33: Sơ đồ tổng quát các package của AccountService



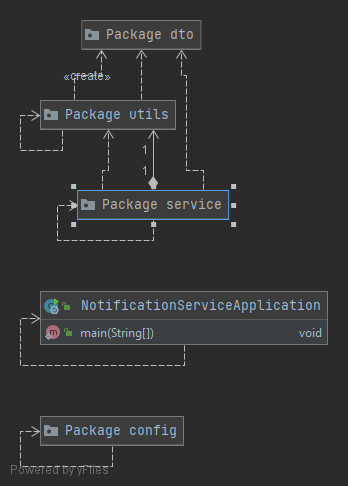
Hình 4. 34: Sơ đồ tổng quát các package của AuthService



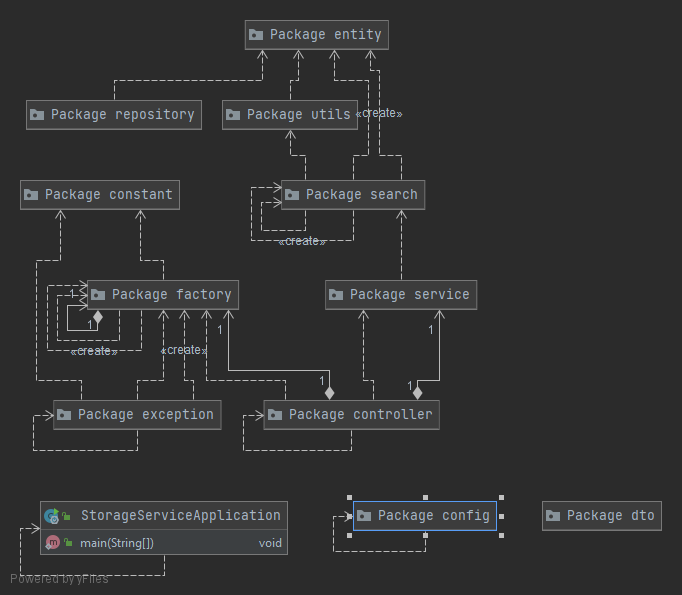
Hình 4. 35: Sơ đồ tổng quát các package của FileService



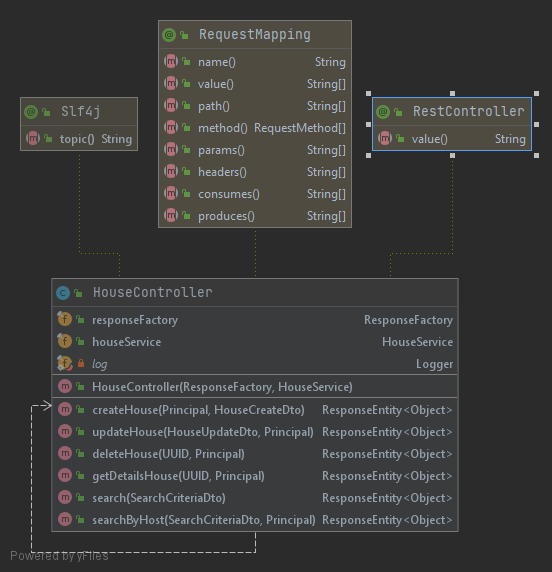
Hình 4. 36: Sơ đồ tổng quát các package của InforService



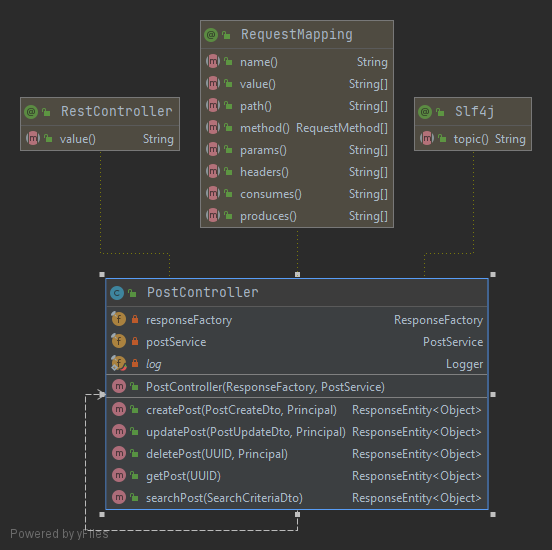
Hình 4. 37: Sơ đồ tổng quát các package của NotificationService



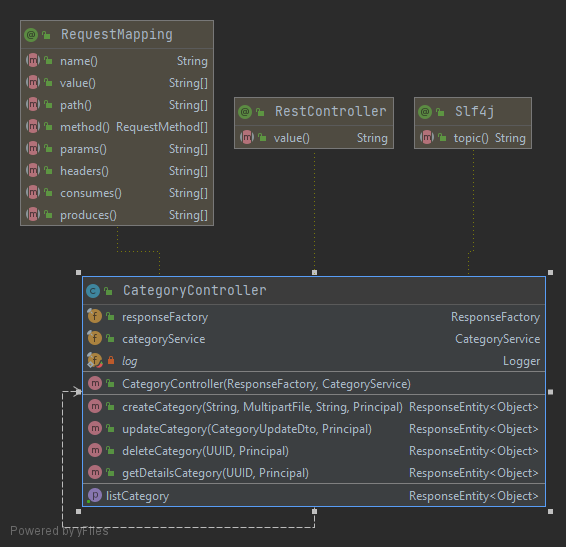
Hình 4. 38: Sơ đồ tổng quát các package của StorageService



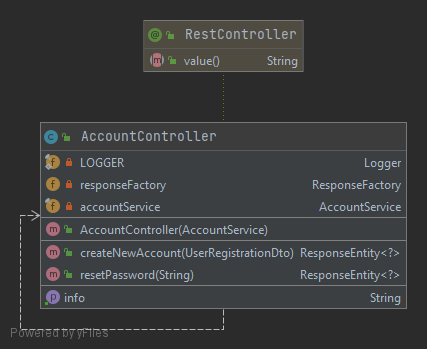
Hình 4. 39: Sơ đồ lớp các chức năng liên quan đến phòng



Hình 4. 40: Sơ đồ lớp các chức năng liên quan đến bài viết

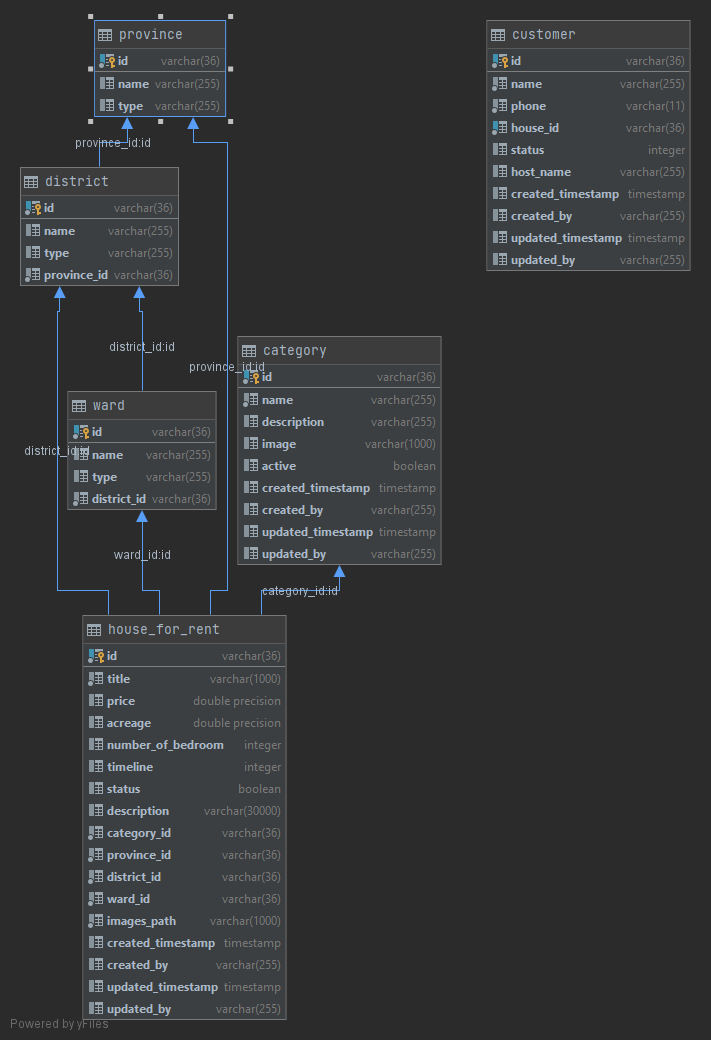


Hình 4. 41: Sơ đồ lớp các chức năng liên quan đến loại phòng

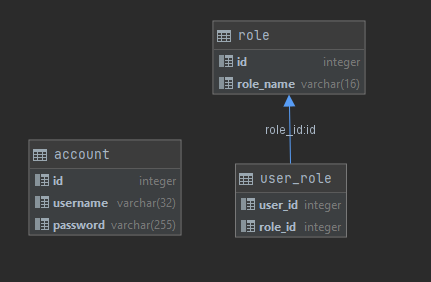


Hình 4. 42: Sơ đồ lớp các chức năng liên quan đến tài khoản

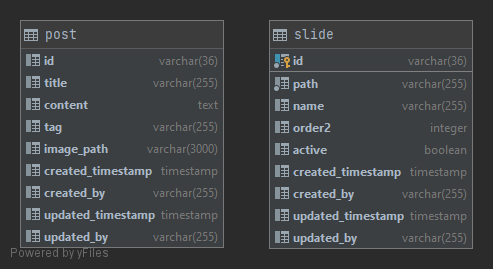
### ***4.3.4. Mô hình thực thể***



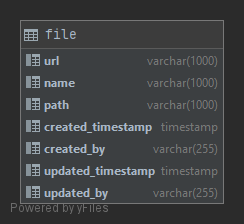
Hình 4. 43: Mô hình thực thể ứng với database house\_rental



Hình 4. 44: Mô hình thực thể ứng với database auth\_service



Hình 4. 45: Mô hình thực thể ứng với database infor\_service



Hình 4. 46: Mô hình thực thể ứng với database file\_service

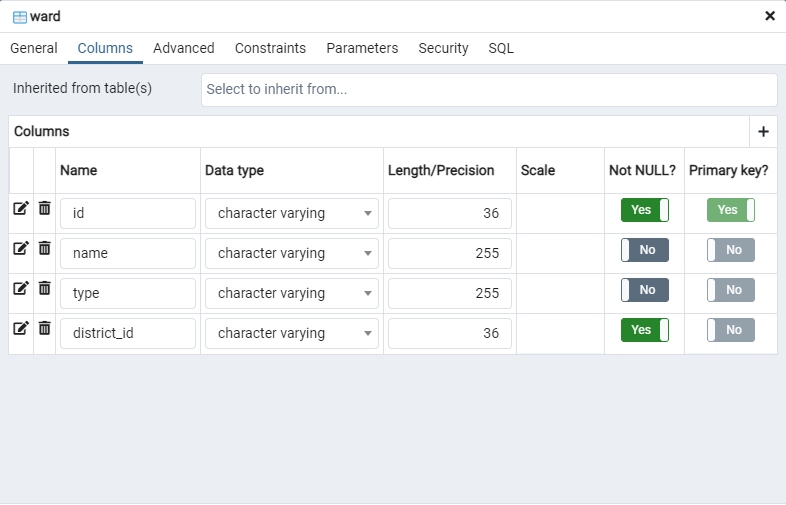
### ***4.3.5. Thiết kế dữ liệu***

a. Danh sách các bảng

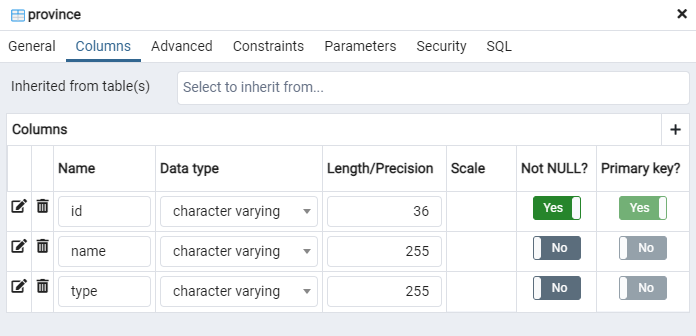
Bảng 4.1: Danh sách các bảng

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên bảng** | **Mô tả** |
| 1 | category | Lưu thông tin loại nhà |
| 2 | house\_for\_rent | Lưu thông tin bài đăng cho thuê nhà |
| 3 | customer | Lưu thông tin khách hàng |
| 4 | ward | Lưu thông tin Phường/xã |
| 5 | district | Lưu thông tin Quận/huyện |
| 6 | province | Lưu thông tin Thành phố/tỉnh |
| 7 | post | Lưu thông tin bài đăng tin tức |
| 8 | slide | Lưu thông tin banner trang chủ |
| 9 | file | Lưu thông tin các file upload |
| 10 | account | Lưu thông tin người dùng |
| 11 | role | Lưu thông tin quyền |
| 12 | user\_role | Lưu thông tin người dùng và quyền |

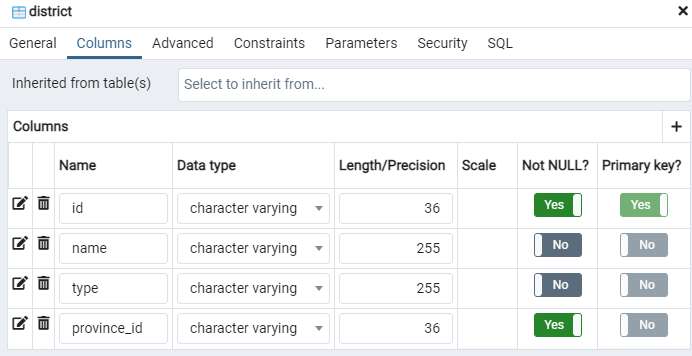
b. Các bảng dữ liệu



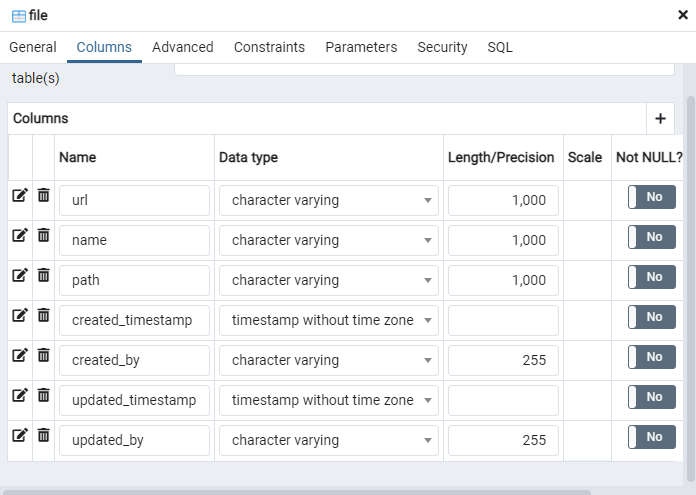
Bảng 4.2: Bảng lưu thông tin phường/xã

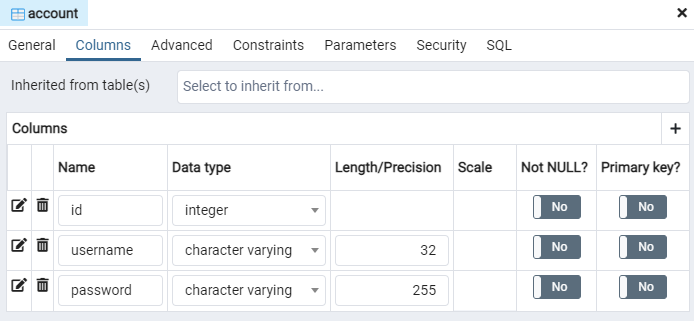


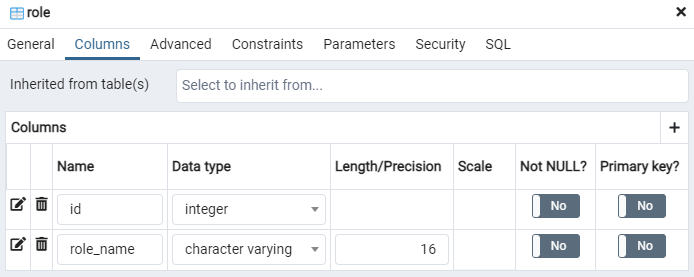
Bảng 4.3: Bảng lưu thông tin thành phố/tỉnh



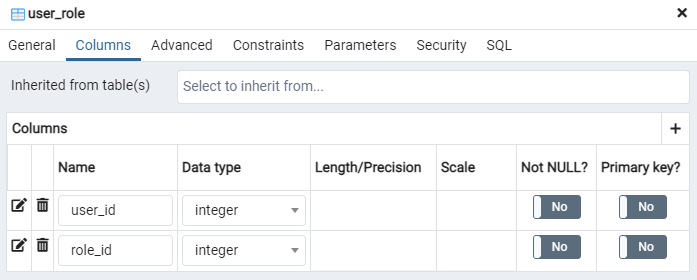
Bảng 4.4: Bảng lưu thông tin quận/huyện

  
Bảng 4.5: Bảng lưu thông tin các file upload

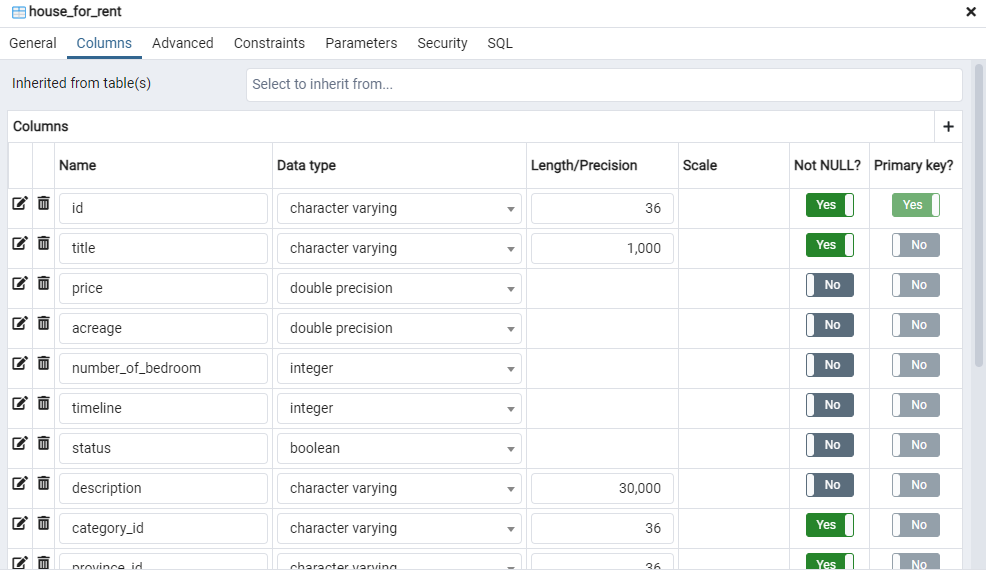
Bảng 4.6: Bảng lưu thông tin người dùng

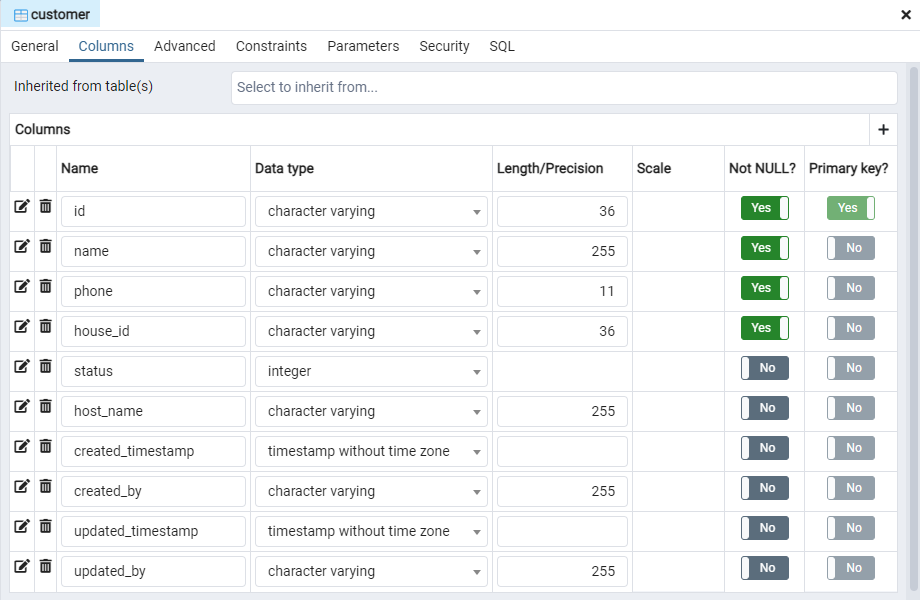


Bảng 4.7: Bảng lưu thông tin quyền của người dùng

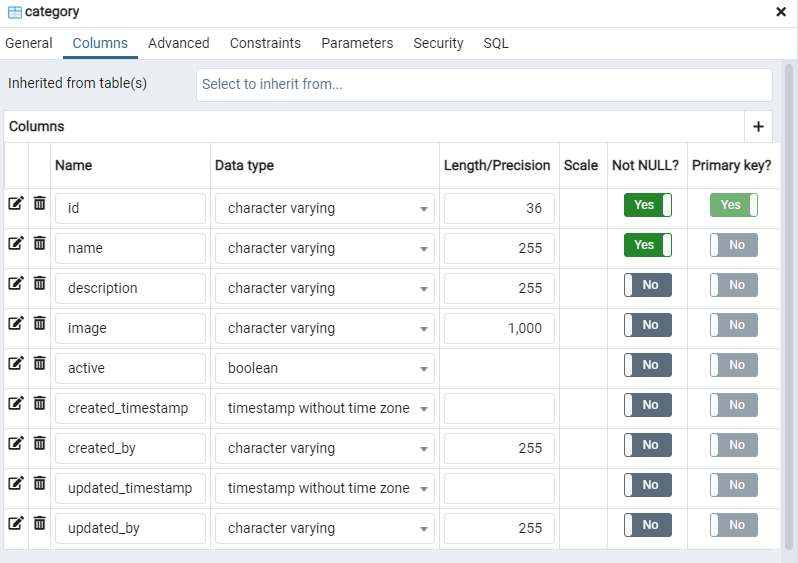


Bảng 4.8: Bảng lưu thông tin người dùng và quyền

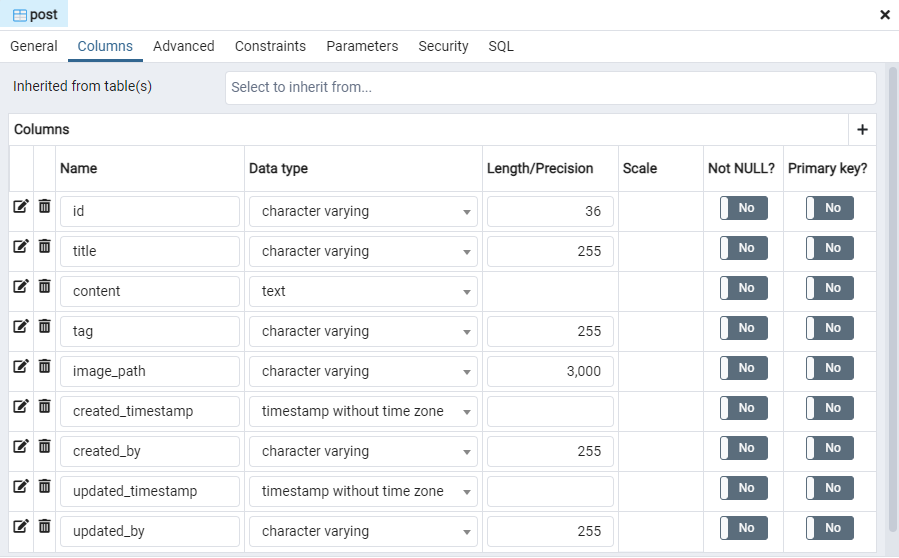
  
Bảng 4.9: Bảng lưu thông tin bài đăng cho thuê nhà



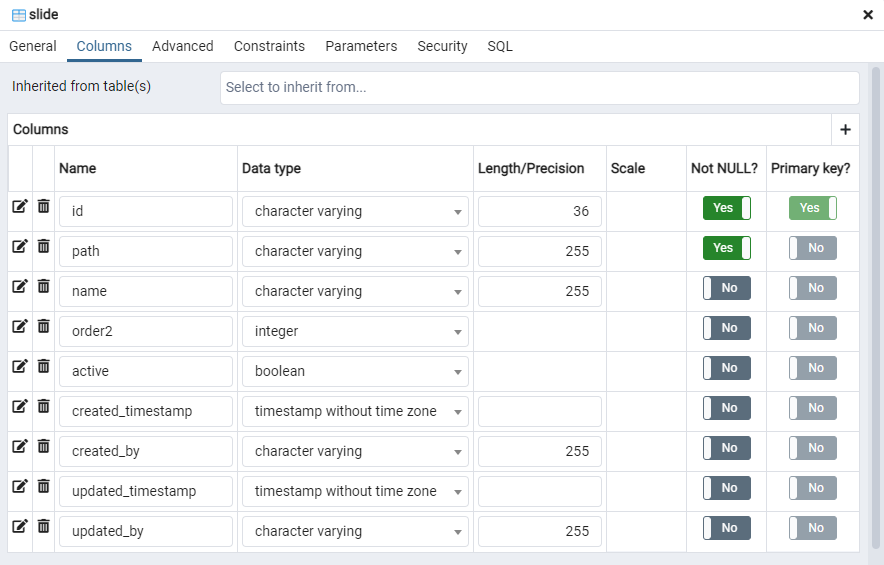
Bảng 5.0: Bảng lưu thông tin khách hàng



Bảng 5.1: Bảng lưu thông tin loại nhà



Bảng 5.2: Bảng lưu thông tin bài viết

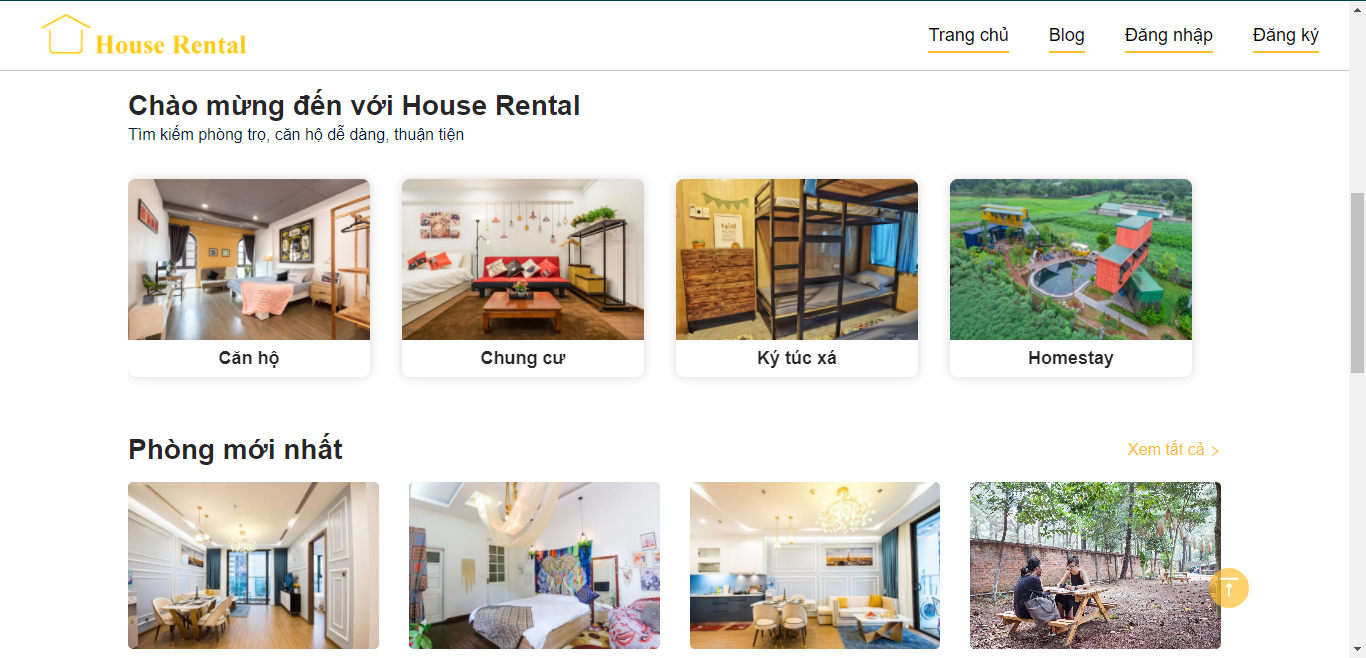


Bảng 5.3: Bảng lưu thông tin ảnh bìa trang chủ

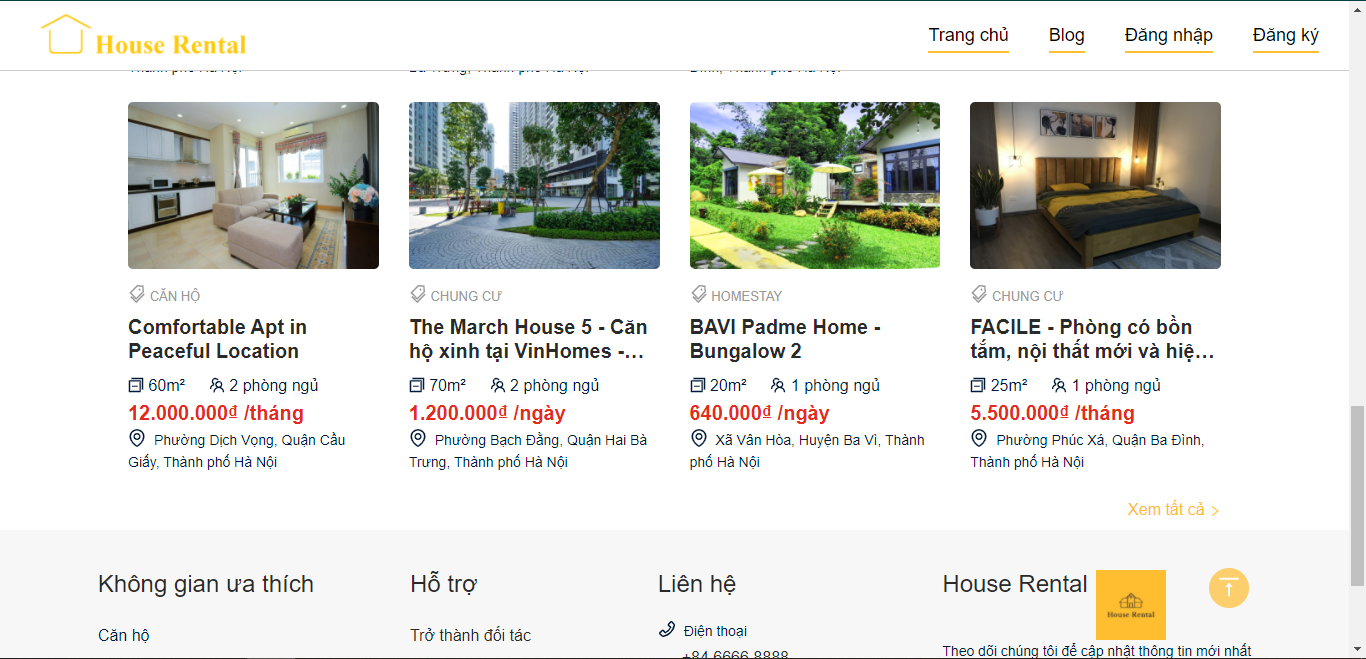
### ***4.3.6. Thiết kế giao diện***

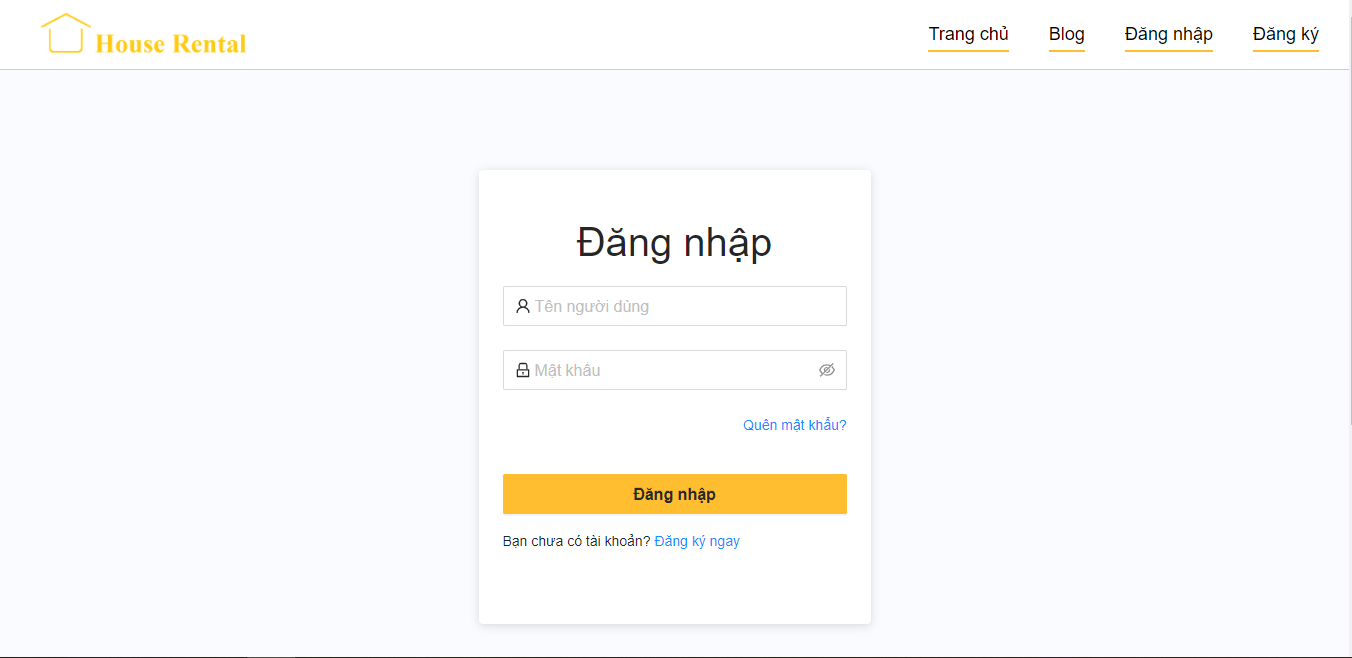


Hình 4. 47: Giao diện trang chủ

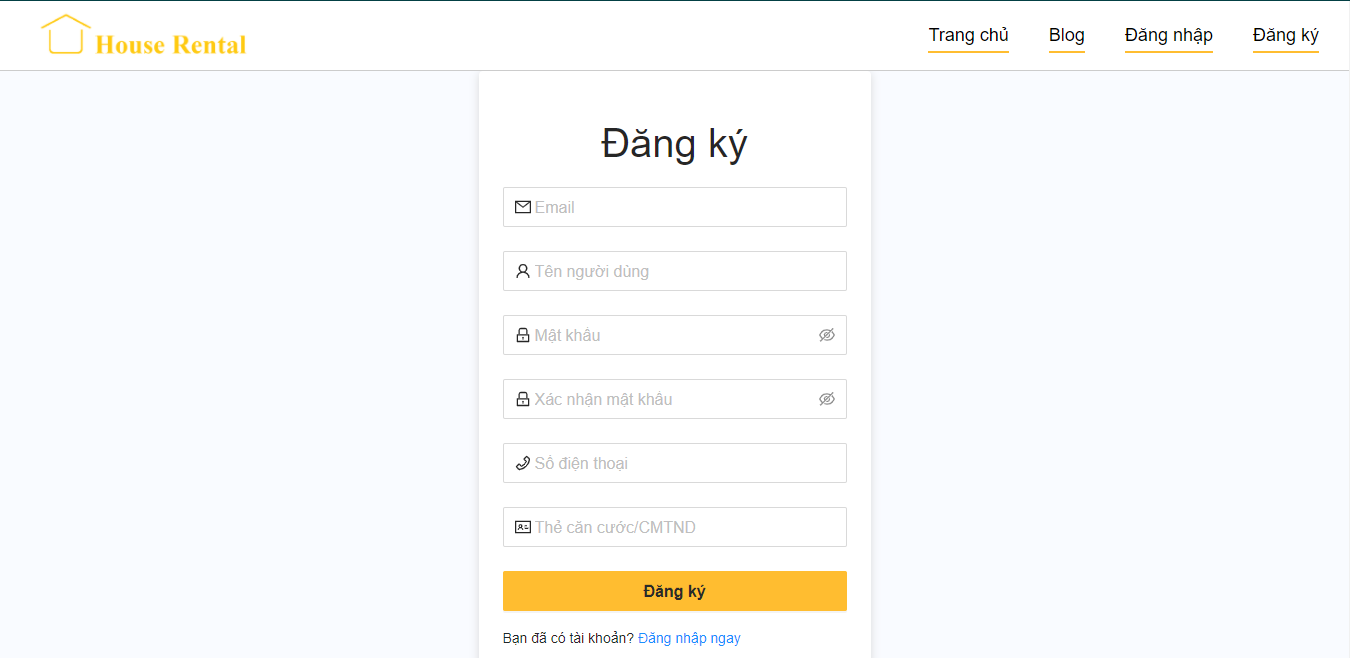


Hình 4. 48: Giao diện trang chủ (hiển thị các loại phòng)

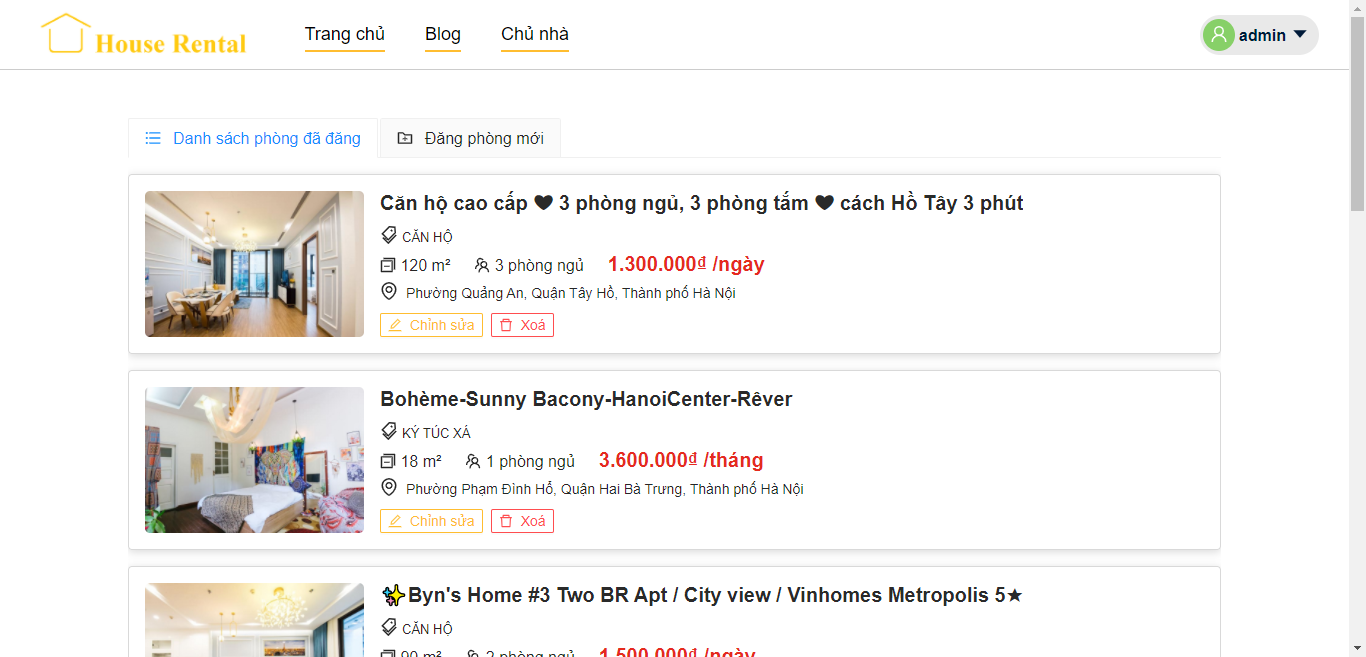


Hình 4. 49: Giao diện trang chủ (hiển thị các bài đăng)

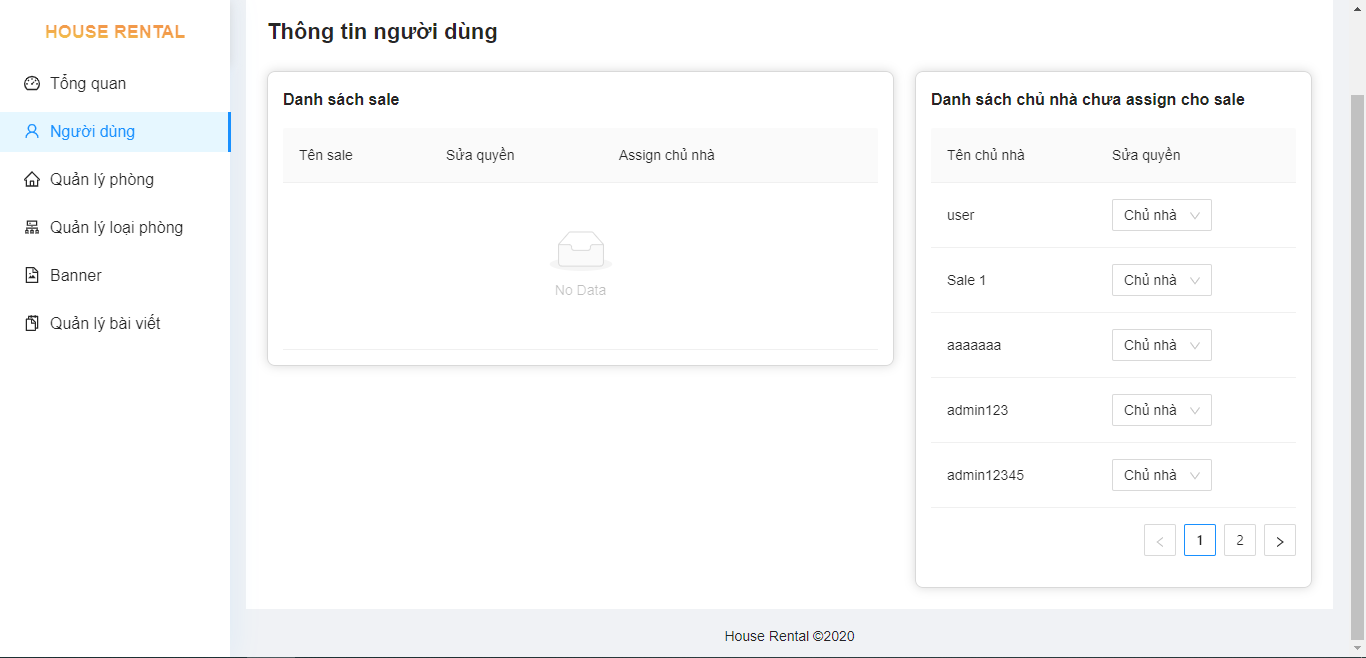
Hình 4. 50: Giao diện đăng nhập



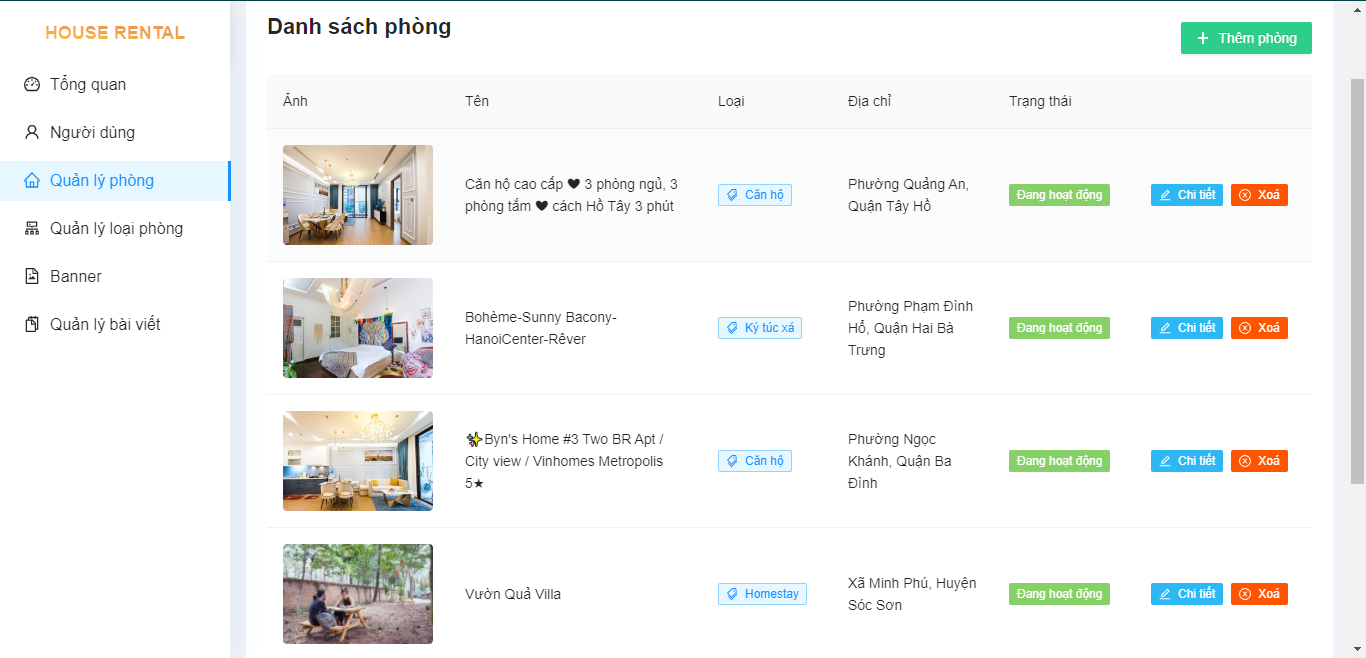
Hình 4. 51: Giao diện đăng ký



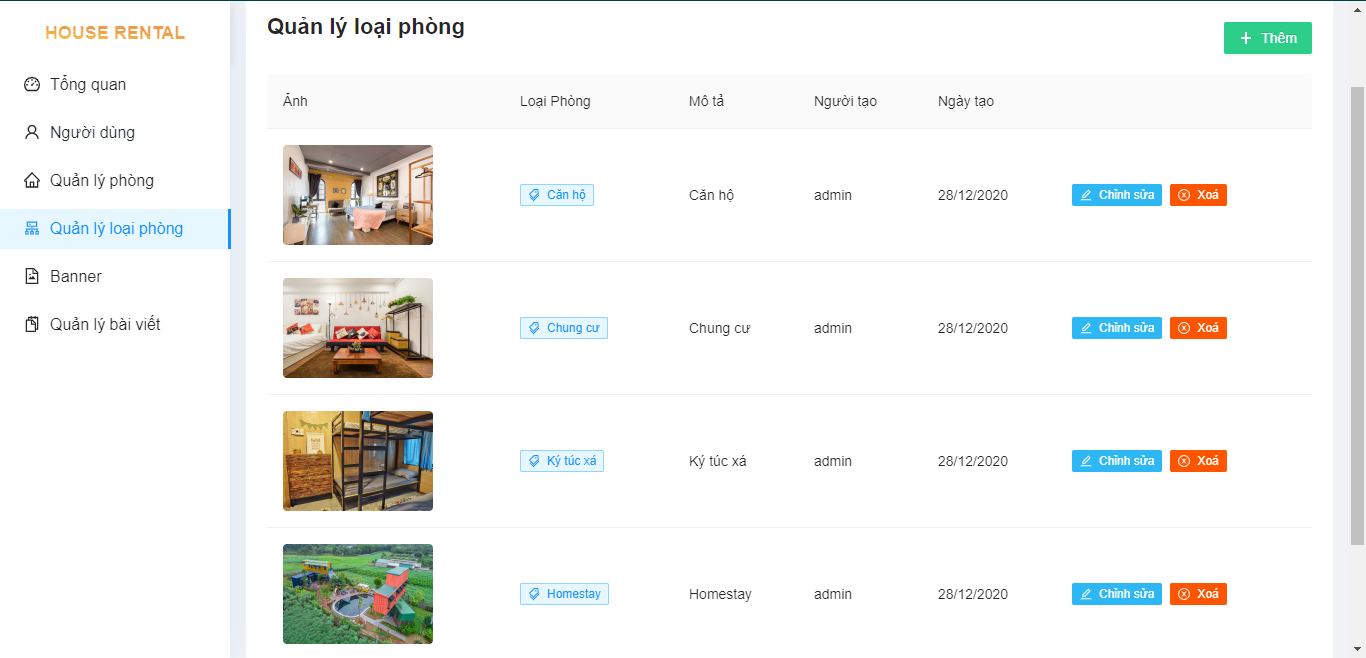
Hình 4. 52: Giao diện quản lý bài đăng của chủ nhà



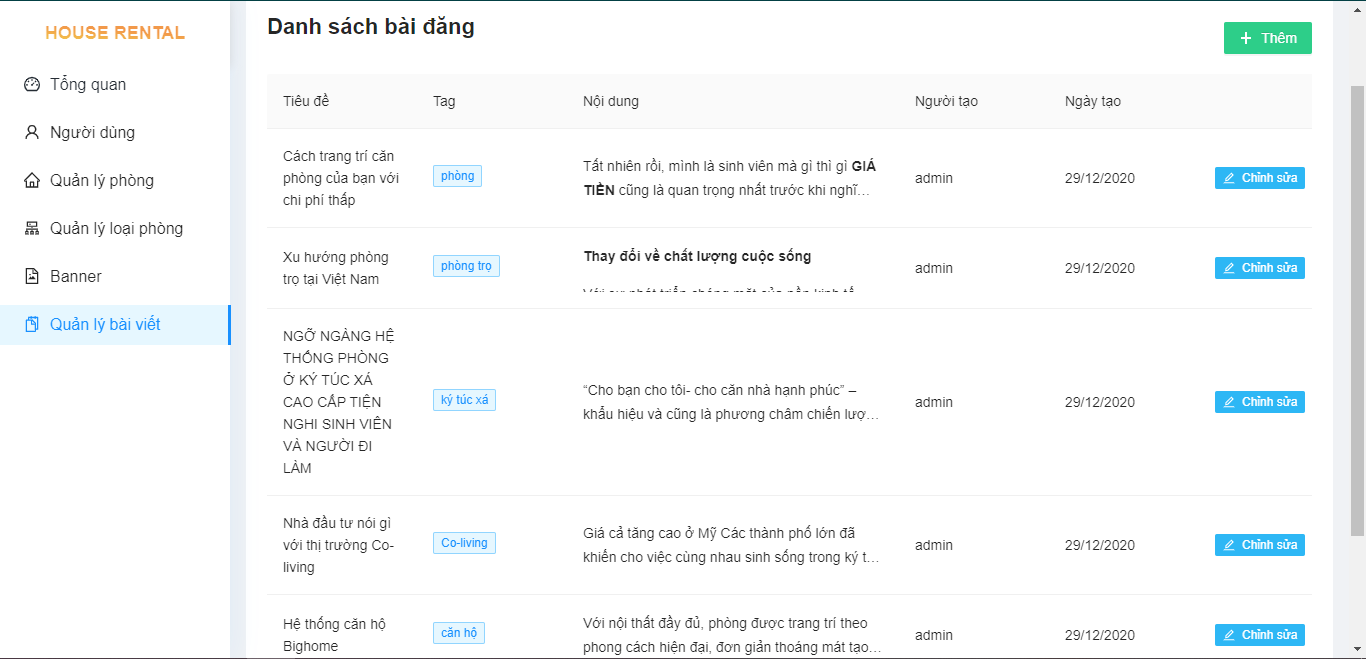
Hình 4. 53: Giao diện quản lý thông tin người dùng



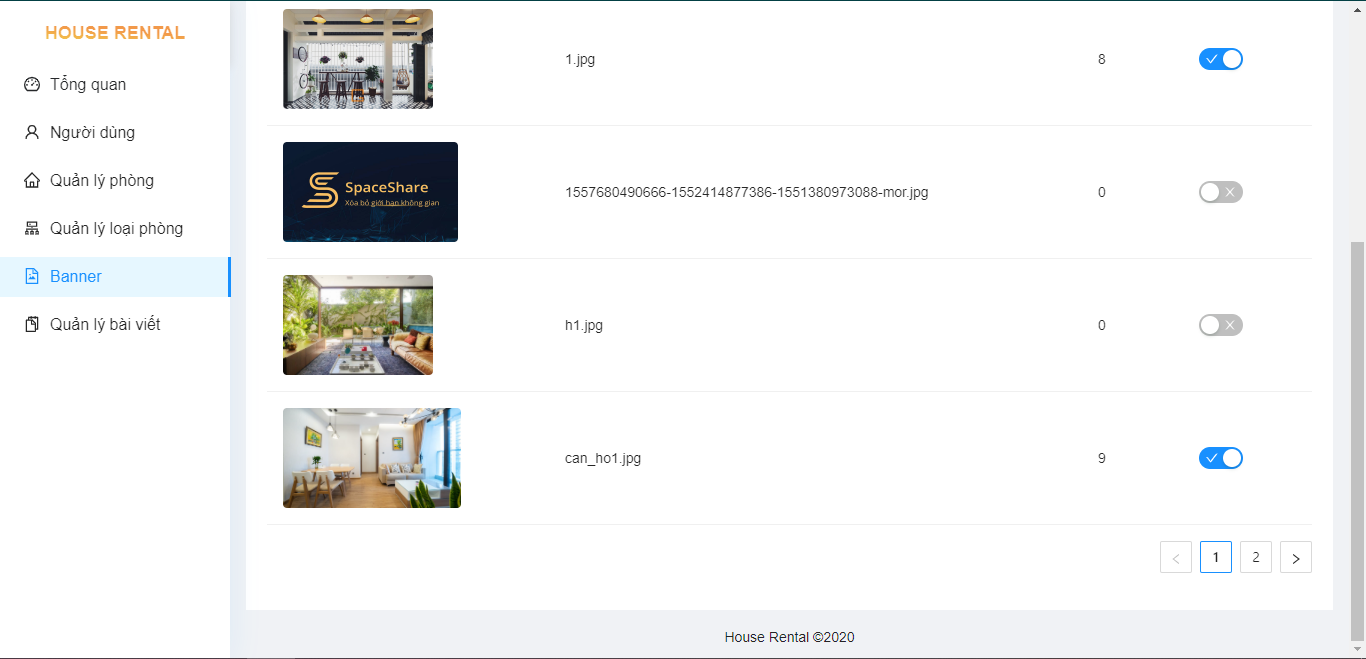
Hình 4. 54: Giao diện quản lý phòng



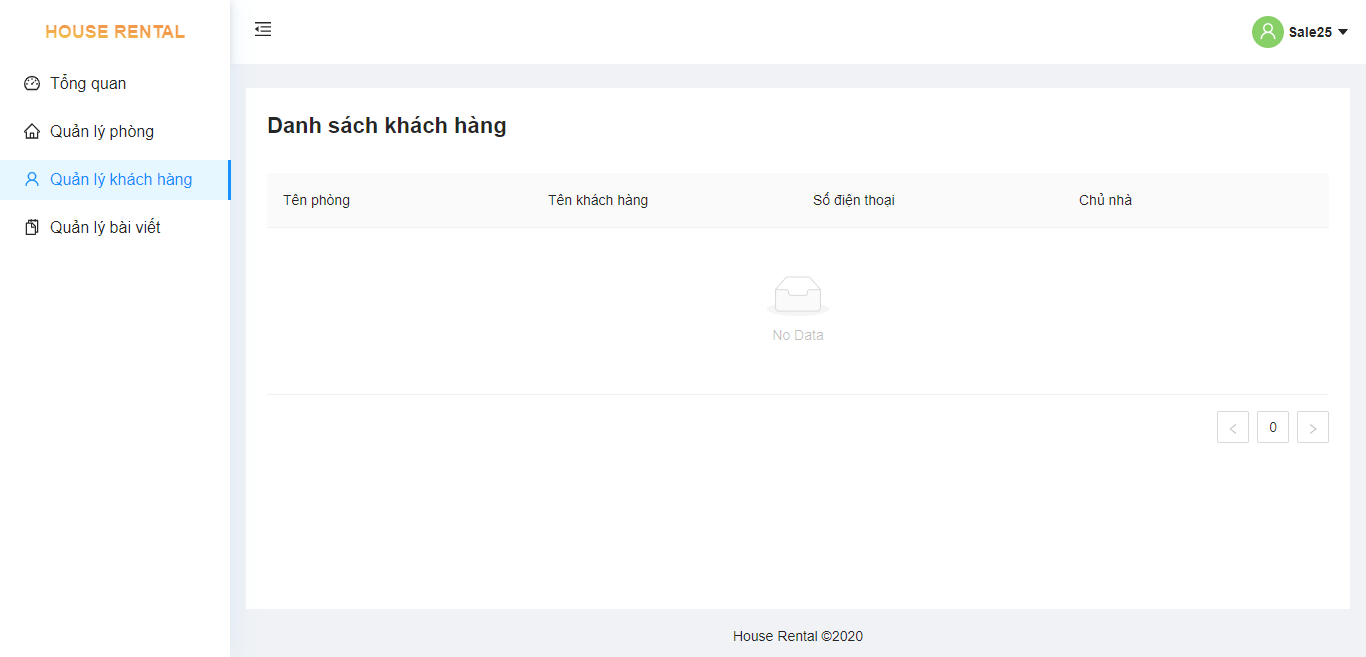
Hình 4. 55: Giao diện quản lý loại phòng



Hình 4. 56: Giao diện quản lý bài viết



Hình 4. 57: Giao diện quản lý ảnh bìa



Hình 4. 58: Giao diện quản lý khách hàng của trang quản trị sale

## **4.4. Cài đặt và thử nghiệm**

### **4.4.1. Môi trường cài đặt và lập trình**

- Ngôn ngữ lập trình: Java, Nodejs

- Framework: SpringBoot

- Hệ quản trị cơ sở dữ liệu: Postgresql, Mongodb

- Công cụ lập trình: Visual Studio Code, IntelliJ IDEA, Sublime Text 3

### **4.4.2. Thực thi ứng dụng**

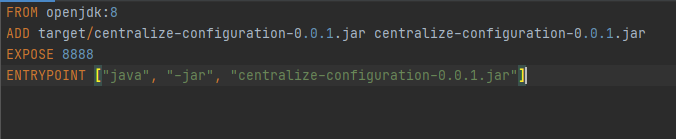
- Quy trình hoạt động của ứng dụng được thực thi bởi Docker. Trước tiên ta nói về Docker.

- Docker là nền tảng để phát triển và chạy các ứng dụng. Docker cho phép bạn tách các ứng dụng khỏi cơ sở hạ tầng để bạn có thể phát triển phần mềm một cách nhanh chóng.

- Về nền tảng Docker, Docker cung cấp khả năng đóng gói và chạy ứng dụng trong một môi trường cô lập được gọi là “container” đồng thời cho phép bạn chạy nhiều container trên một máy chủ nhất định.

- Ở đây ta thực thi hệ thống, ta sẽ thực thi từ Dockerfile của từng project. Dockerfile là một loại văn bản chứa tất cả các lệnh mà người dùng có thể gọi trên dòng lệnh để sử dụng image. Nó sẽ chứa tất cả các phụ thuộc mà hệ thống yêu cầu. Để chạy Dockerfile thì ta sẽ sử dụng Docker Compose.

- Đầu tiên ta cần chạy Dockerfile của Centralize-configuration



Hình 4. 59: Cấu hình Dockerfile của Centralize-configuration

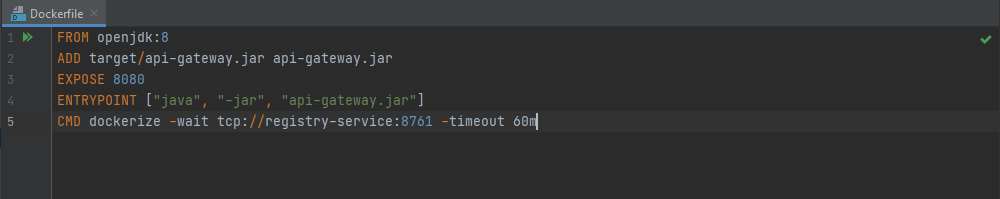
Với FROM là để xây dựng image bắt đầu.

ADD sẽ giải nén dưới dạng thư mục và sao chép nội dung của thư mục

EXPOSE thông báo cho Docker rằng vùng container sẽ lắng nghe trên các cổng được chỉ định trong thời gian chạy.

ENTRYPOINT sẽ cho phép bạn cấu hình một vùng chứa sẽ chạy dưới dạng tệp để thực thi

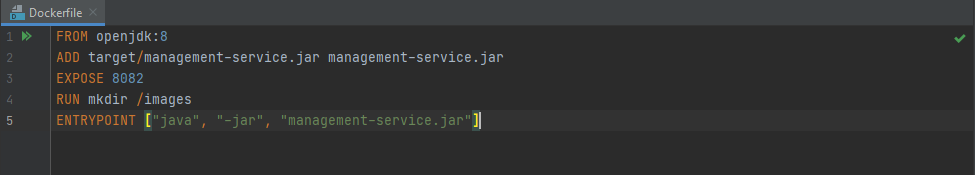
Tiếp đó đến Dockerfile của API Gateway



Hình 4. 60: Cấu hình Dockerfile của API Gateway

Ta có thêm từ khóa nữa là CMD: chỉ định service nào được chạy trong khoảng thời gian nhất định.

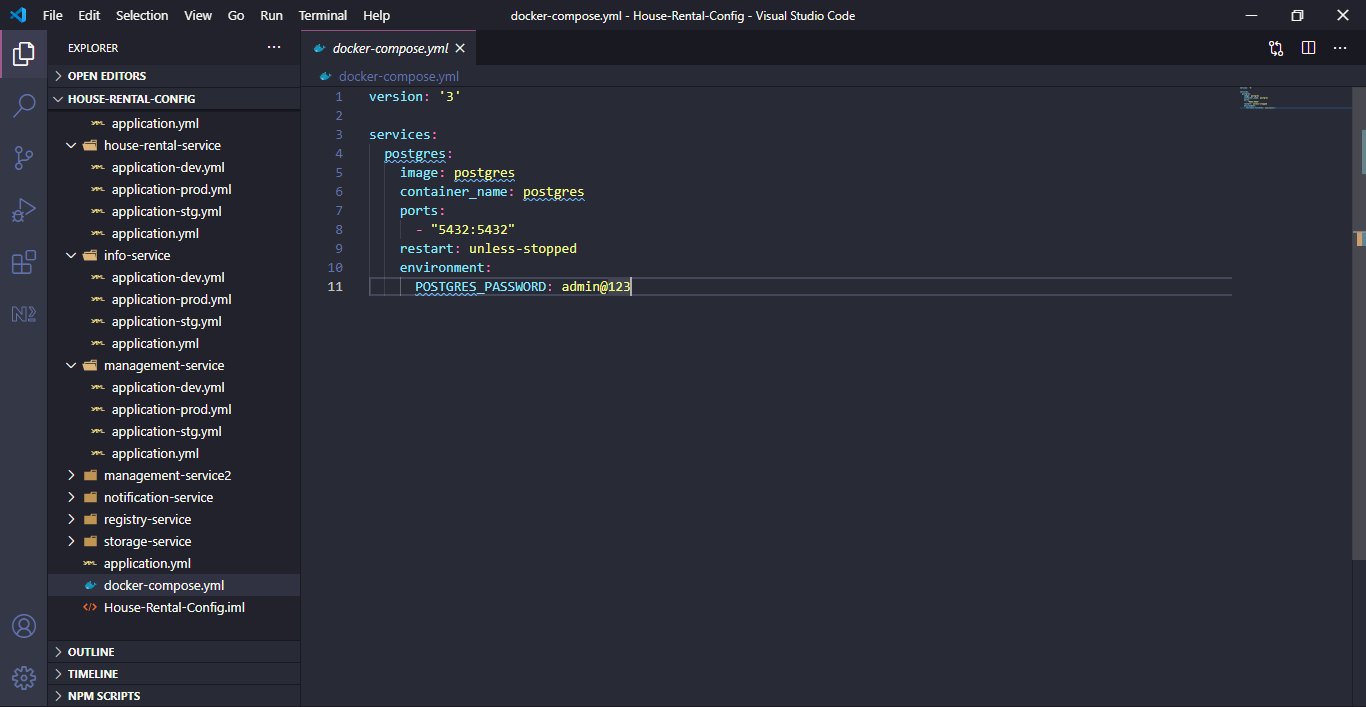
Tiếp theo đến Dockerfile của các service



Hình 4. 61: Cấu hình Dockerfile của Management-service

Ta có thêm lệnh RUN: để thực thi lệnh tạo thư mục images.

Tiếp theo ta chạy file docker-compose.yml



Hình 4. 62: Cấu hình của file docker-compose.yml

Sau đó ta chạy bằng cách nhập lệnh: docker-compose up

### **4.4.3. Kiểm thử**

Danh sách các chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Số thứ tự | Tên chức năng | Tình trạng | Ghi chú |
| 1 | Login | ĐẠT |  |
| 2 | Logout | ĐẠT |  |
| 3 | Đăng ký | ĐẠT |  |
| 4 | Tìm bài đăng | ĐẠT |  |
| 5 | Thêm loại phòng | ĐẠT |  |
| 6 | Sửa loại phòng | ĐẠT |  |
| 7 | Xóa loại phòng | ĐẠT |  |
| 8 | Thêm bài đăng | ĐẠT |  |
| 9 | Sửa bài đăng | ĐẠT |  |
| 10 | Xóa bài đăng | ĐẠT |  |
| 11 | Thêm banner | ĐẠT |  |
| 12 | Thêm bài viết | ĐẠT |  |
| 13 | Sửa bài viết | ĐẠT |  |
| 14 | Xóa bài viết | ĐẠT |  |

# **Phần 5: Kết luận và đề nghị**

## **5.1. Kết luận**

Những kết quả đạt được:

- Về kiến thức:

**+)** Tìm hiểu và nắm bắt được kiến thức về phát triển ứng dụng web.

**+)** Thiết kế hệ thống: thiết kế giao diện, thiết kế cơ sở dữ liệu, thiết kế thông tin…

**+)** Các phương pháp phân tích và thiết kế.

**+)** Tìm hiểu và nắm bắt được các phương thức quản lý đề cương môn học một cách tốt và hiệu quả nhất.

- Về kỹ năng:

**+)** Kỹ năng khảo sát thực tế, thu thập thông tin, xử lý dữ liệu phục vụ tìm hiểu và phát triển tài liệu lý thuyết và xây dựng ứng dụng.

**+)** Đọc tài liệu, tài liệu liên quan và tổng hợp lại.

**+)** Nắm chắc các cách thức làm việc với việc xây dựng chương trình từ bản phân tích thiết kế hệ thống và hướng đối tượng.

- Về tham khảo ý kiến:

**+)** Tiếp nhận ý kiến của giảng viên hướng dẫn, định hướng, công ty thực tập,…

**+)** Tham khảo ý kiến chuyên gia.

- Về ứng dụng:

Bước đầu đã xây dựng được một số chức năng cho website, đáp ứng được một số yêu cầu trong việc quản lý, tìm kiếm,…. Website đã hoàn thiện các chức năng cơ bản sau:

+) Quản lý bài viết

+) Quản lý bài đăng

+) Quản lý loại phòng

+) Quản lý banner

+) Quản lý thành viên

Phần tìm hiểu lý thuyết có thể sử dụng như một tài liệu tham khảo trong việc tìm hiểu về Web cùng các công cụ phát triển cho những người có nhu cầu. Phần ứng dụng hoàn thành khá đầy đủ các chức năng, giúp khách hàng và người quản trị, có thể dễ dàng tìm kiếm, sử dụng một cách hiệu quả, nhanh chóng, đơn giản, thuận tiện hơn và chính xác.

## **5.2. Đề nghị**

Để ứng dụng có thể thực sự áp dụng được phát triển hơn nữa, ứng dụng cần thêm một số chức năng.

# **Phần 6: Tài liệu tham khảo**

Trần Trung Hiếu: Slide bài giảng môn công nghệ phần mềm, Java  
Cuốn: Monolith to Microservices – Tác giả: Sam Newman

Cuốn: Microservices From Design to Deployment – Tác giả: Chris Richardson with Floyd Smith

Cuốn: Building Microservices – Tác giả: Sam Newman

W3schools, <https://www.w3schools.com/>

Microservice, <https://www.nginx.com/blog/introduction-to-microservices/> ,

<https://www.edureka.co/blog/what-is-microservices/> ,

<https://viblo.asia/p/gioi-thieu-ve-kien-truc-microservices-4P8566O35Y3>

<https://github.com/Netflix/ribbon/wiki/Working-with-load-balancers?fbclid=IwAR0A3e7TCR5zGnVcaHMSUBkLaL5TWf5S9ZY8EetHridh>

<https://www.edureka.co/blog/what-is-microservices/>

<https://microservices.io/>

<https://www.nginx.com/resources/glossary/round-robin-load-balancing/>

<https://avinetworks.com/glossary/round-robin-load-balancing/>

**GIẢNG VIÊN HƯỚNG DẪN SINH VIÊN**

(*Ký và ghi rõ họ tên) (Ký và ghi rõ họ tên)*