**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



ĐỒ ÁN TỐT NGHIỆP

*ĐỀ TÀI*

HỆ THỐNG THƯƠNG MẠI ĐIỆN TỬ CHỢ ĐỒ CŨ

Giảng viên hướng dẫn: TS.Nguyễn Hữu Đức

Sinh viên thực hiện: Trịnh Văn Tài

***Hà Nội 8/10/2019***

**LỜI NÓI ĐẦU**

Ngày nay, các sản phẩm công nghệ thông tin ngày càng đóng vai trò quan trọng trong mọi lĩnh vực của đời sống. Trong số đó, nổi bật lên là các hệ thống thương mại điện tử, trở thành một xu hướng mới trong việc đáp ứng các nhu cầu mua bán. Nhận thấy được xu hướng này, chúng em đã có ý tưởng xây dựng một hệ thống thương mại điện tử “Chợ đồ cũ”, tạo ra một phương tiện kết nối giữa mọi người có nhu cầu về mua bán. Hệ thống được xây dựng và triển khai sẽ đáp ứng nhu cầu của mọi người về một nguồn thông tin mở, đáng tin cậy, do chính mọi người đóng góp nên

Thông qua việc thực hiện đồ án này, em đã được rèn luyện các kỹ năng tổng thể trong việc xây dựng một hệ thống phần mềm, từ việc lên ý tưởng, phân tích thiết kế hệ thống, cho đến lập trình, kiểm thử, viết tài liệu ... Đồng thời, chúng em còn có cơ hội nâng cao kỹ năng lập kế hoạch, một trong số những kỹ năng mềm quan trọng khi hội nhập nghề nghiệp.

Em xin chân thành cảm ơn TS.Nguyễn Hữu Đức đã nhiệt tình hướng dẫn, giám sát em trong suốt quá trình làm đồ án.

Trong quá trình thực hiện đồ án, không thể tránh khỏi những thiếu sót nhất định do thời gian và một số hạn chế về chuyên môn. Vì vậy, em mong nhận được sự góp ý từ thầy và hội đồng để tiếp tục hoàn thiện hơn nữa sản phẩm của chúng em.

*Hà Nội ngày 8 tháng 10 năm 2019*

**Sinh viên thực hiện**

Trịnh Văn Tài

MỤC LỤC

[CHƯƠNG 1: GIỚI THIỆU 7](#_Toc25413627)

[1.1 Giới thiệu bài toán 7](#_Toc25413628)

[1.2 Phạm vi hệ thống 7](#_Toc25413629)

[1.3 Cách tiếp cận 7](#_Toc25413630)

[1.4 Giới thiệu cấu trúc đồ án 7](#_Toc25413631)

[CHƯƠNG 2: CÁC NỀN TẢNG CÔNG NGHỆ 8](#_Toc25413632)

[2.1 Giới thiệu về Spring Boot 8](#_Toc25413636)

[2.2 Giới thiệu về Angular 7 9](#_Toc25413637)

[CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG 11](#_Toc25413641)

[3.1 Các đối tác 11](#_Toc25413643)

[3.2 Các ca sử dụng 12](#_Toc25413644)

[3.3 Đặc tả ca sử dụng 13](#_Toc25413648)

[3.3.1 Đăng nhập 13](#_Toc25413649)

[3.3.2 Đăng kí tài khoản 15](#_Toc25413650)

[3.3.3 Đổi lại mật khẩu 18](#_Toc25413651)

[3.3.4 Tìm kiếm 20](#_Toc25413652)

[3.3.5 Đăng tin 22](#_Toc25413653)

[3.3.6 Bình luận 25](#_Toc25413654)

[3.3.7 Nhắn tin 28](#_Toc25413655)

[3.3.8 Theo dõi tài khoản 30](#_Toc25413656)

[3.3.9 Nạp thẻ cào 33](#_Toc25413657)

[3.3.10 Đẩy tin đăng 36](#_Toc25413658)

[3.3.11 Ẩn tin 39](#_Toc25413659)

[3.3.12 Phản ánh tin đăng 40](#_Toc25413660)

[3.3.13 Theo dõi thống kê 43](#_Toc25413661)

[3.3.14 Kiểm duyệt tin đăng 45](#_Toc25413662)

[3.3.15 Kiểm duyệt phản ánh 48](#_Toc25413663)

[3.3.16 Thống kê số lượng tin đăng 50](#_Toc25413664)

[CHƯƠNG 4: THIẾT KẾ HỆ THỐNG 53](#_Toc25413665)

[4.1 Kiến trúc hệ thống 54](#_Toc25413667)

[4.1.1 Kiến trúc triển khai 54](#_Toc25413668)

[4.1.2 Kiến trúc logic 55](#_Toc25413669)

[4.1.3 Kiến trúc mã nguồn 56](#_Toc25413670)

[4.1.4 Các kĩ thuật 56](#_Toc25413671)

[4.2 Thiết kế ca sử dụng 59](#_Toc25413672)

[4.2.1 Đăng tin 59](#_Toc25413673)

[4.2.2 Bình luận 62](#_Toc25413674)

[4.2.3 Nhắn tin 64](#_Toc25413675)

[4.2.4 Theo dõi tài khoản 66](#_Toc25413676)

[4.2.5 Nạp thẻ cào 68](#_Toc25413677)

[4.2.6 Đẩy tin đăng 70](#_Toc25413678)

[4.2.7 Phản ánh tin đăng 72](#_Toc25413679)

[4.2.8 Theo dõi thống kê tin đăng 72](#_Toc25413680)

[4.2.9 Kiểm duyệt tin đăng 72](#_Toc25413681)

[4.2.10 Kiểm duyệt thông tin phản ánh 72](#_Toc25413682)

[4.2.11 Thống kê số lượng tin đăng 72](#_Toc25413683)

[4.3 Thiết kế cơ sở dữ liệu 73](#_Toc25413684)

[4.3.1 *Sơ đồ quan hệ* 73](#_Toc25413685)

[4.3.2 *Mô tả cơ sở dữ liệu* 74](#_Toc25413686)

[4.4 Thiết kế giao diện 80](#_Toc25413687)

[CHƯƠNG 5: XÂY DỰNG VÀ THỬ NGHIỆM 81](#_Toc25413688)

[5.1 Môi trường xây dựng và thử nghiệm 81](#_Toc25413690)

[5.2 Các kết quả thử nghiệm 81](#_Toc25413691)

[CHƯƠNG 6: KẾT LUẬN 81](#_Toc25413692)

[6.1 Đã làm 81](#_Toc25413694)

[6.2 Mở rộng 81](#_Toc25413695)

[TÀI LIỆU THAM KHẢO 81](#_Toc25413696)

[*Hình 1: Kiến trúc Spring* 8](#_Toc25419835)

[*Hình 2: Tương tác giữa các thành phần trong Angular 7* 10](#_Toc25419836)

[*Hình 3: Minh họa giao diện đăng nhập* 15](#_Toc25419837)

[*Hình 4: Minh họa giao diện đăng kí tài khoản* 18](#_Toc25419838)

[*Hình 5: Minh họa giao diện đổi lại mật khẩu* 21](#_Toc25419839)

[*Hình 6: Minh họa giao diện tìm kiếm* 23](#_Toc25419840)

[*Hình 7: Minh họa giao diện đăng tin* 25](#_Toc25419841)

[*Hình 8: Minh họa giao diện bình luận* 28](#_Toc25419842)

[*Hình 9: Minh họa giao diện chát* 31](#_Toc25419843)

[*Hình 10: Minh họa trang cá nhân người dùng* 33](#_Toc25419844)

[*Hình 11: Minh họa giao diện nạp thẻ cào* 36](#_Toc25419845)

[*Hình 12: Minh họa giao diện đẩy tin đăng* 38](#_Toc25419846)

[*Hình 13: Minh họa giao diện phản ánh tin đăng* 43](#_Toc25419847)

[*Hình 14: Minh họa giao diện thống kê lượt truy cập của tin đăng* 45](#_Toc25419848)

[*Hình 15: Minh họa giao diện kiểm duyệt tin đăng* 48](#_Toc25419849)

[*Hình 16: Minh họa giao diện thống kê số lượng tin đăng* 53](#_Toc25419850)

[*Hình 17: Kiến trúc triển khai của hệ thống* 54](#_Toc25419851)

[*Hình 18: Kiến trúc logic của ứng dụng* 56](#_Toc25419852)

[*Hình 19: Luồng xử lí Auth0* 58](#_Toc25419853)

[*Hình 20: Quan hệ giữa các bảng cơ sở dữ liệu* 73](#_Toc25419854)

[*Hình 21: Tài liệu PDF cơ sở dữ liệu* 74](#_Toc25419855)

[*Hình 22: Quan hệ giữa bảng Product với Category và Location* 75](#_Toc25419856)

[*Hình 23: Quan hệ giữa bảng Category và Information* 77](#_Toc25419857)

[*Hình 24: Quan hệ giữa Product và Information* 79](#_Toc25419858)

[*Hình 25: Quan hệ giữa Product và Location* 80](#_Toc25419859)

[*Hình 26: Quan hệ giữa Product và Image* 81](#_Toc25419860)

[*Hình 27: Quan hệ giữa Comment với Product và User* 83](#_Toc25419861)

[*Hình 28: Quan hệ giữa Chat với User và Product* 84](#_Toc25419862)

[*Biểu đồ 1: Biểu đồ trình tự Đăng tin* 58](#_Toc25409097)

[*Biểu đồ 2: Biểu đồ lớp Đăng tin* 59](#_Toc25409098)

[*Biểu đồ 3: Biểu đồ trình tự Bình luận* 60](#_Toc25409099)

[*Biểu đồ 4: Biểu đồ lớp Bình luận* 61](#_Toc25409100)

[*Biểu đồ 5: Biểu đồ trình tự Nhắn tin* 62](#_Toc25409101)

[*Biểu đồ 6: Biểu đồ lớp Nhắn tin* 63](#_Toc25409102)

[*Biểu đồ 7: Biểu đồ trình tự Theo dõi tài khoản* 64](#_Toc25409103)

[*Biểu đồ 8: Biểu đồ lớp Theo dõi tài khoản* 65](#_Toc25409104)

[*Biểu đồ 9: Biểu đồ trình tự Nạp thẻ cào* 66](#_Toc25409105)

[*Biểu đồ 10: Biểu đồ lớp Nạp thẻ cào* 67](#_Toc25409106)

[*Biểu đồ 11:Biểu đồ trình tự Đẩy tin* 68](#_Toc25409107)

[*Biểu đồ 12: Biểu đồ lớp Đẩy tin* 69](#_Toc25409108)

[*Bảng 1: Bảng cơ sở dữ liệu User* 74](#_Toc25419809)

[*Bảng 2: Bảng cơ sở dữ liệu Product* 76](#_Toc25419810)

[*Bảng 3: Bảng cơ cở dữ liệu Category* 78](#_Toc25419811)

[*Bảng 4: Bảng cơ sở dữ liệu Information* 78](#_Toc25419812)

[*Bảng 5: Bảng cơ sở dữ liệu Information\_value* 80](#_Toc25419813)

[*Bảng 6: Bảng cơ sở dữ liệu Location* 81](#_Toc25419814)

[*Bảng 7: Bảng cơ sở dữ liệu Image* 82](#_Toc25419815)

[*Bảng 8: Bảng cơ sở dữ liệu Comment* 83](#_Toc25419816)

[*Bảng 9: Bảng cơ sở dữ liệu Chat* 85](#_Toc25419817)

# CHƯƠNG 1: GIỚI THIỆU

## Giới thiệu bài toán

Mục đích của trang web là cho phép người mua và người bán kết nối và giao dịch an toàn, dễ dàng trong một môi trường tiện lợi và rõ ràng. Mục đích của trang là cung cấp cho người Việt Nam một nền tảng trực tuyến để sử dụng một cách đơn giản, tiện lợi, không rắc rối phức tạp. Không cần phải đăng ký và ai cũng có thể tìm kiếm và đăng các thông tin liên quan về các danh mục sản phẩm cần bán khác nhau, từ bất động sản, đồ điện tử đến xe ô tô và còn nhiều hơn nữa.

Mỗi người trong số chúng ta đều có những sản phẩm có thể bán được. Bên cạnh việc giữ sản phẩm không cần đến ở nhà, bất kỳ ai cũng có thể kiếm thêm tiền bằng cách bán nó cho người khác thông qua thị trường mua bán trực tuyến này.

## Phạm vi hệ thống

Về mặt nội dung và hình thức, hệ thống cần xây dựng hoạt động dưới hình thức của một hệ thống thương mại điện tử, nội dung hướng đến lĩnh vực ẩm thực, phục vụ các hoạt động ăn uống, giải trí, du lịch. Về mặt chức năng, hệ thống vừa phải có các chức năng cơ bản như đăng bài, comment, ... vừa phải có thêm các chức năng “thông minh” như tìm kiếm, . Về mặt quản lý, hệ thống phân cấp thành nhiều mức người dùng như Guest (khách vãng lai chưa có tài khoản), User (người dùng đã đăng ký tài khoản) và đội ngũ quản lý gồm các Admin. Về mặt hoạt động, hệ thống phải đáp ứng được các yêu cầu về an toàn thông tin, khả năng chịu tải, hoạt động ổn định trên môi trường mạng ...

## Cách tiếp cận

Back-end : Spring Boot 3

Front-end : Angular 7

## Giới thiệu cấu trúc đồ án

Tài liệu mô tả một cách tổng quan trạng thái hiện tại của hệ thống về mục đích phát triển hệ thống và những vấn đề gặp phải. Tài liệu sẽ gồm các mục chính sau:

* Yêu cầu kỹ thuật
* Mô tả thiết kế phần mềm
* Xây dựng và thử nghiệm
* Kết luận

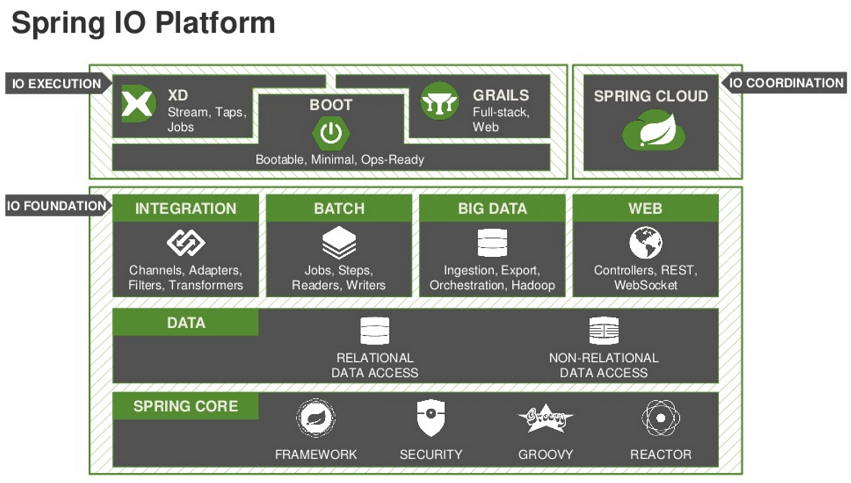
# CHƯƠNG 2: CÁC NỀN TẢNG CÔNG NGHỆ







## Giới thiệu về Spring Boot

****

*Hình 1: Kiến trúc Spring*

**Spring Boot**  là một Project nằm trên tầng IO Execution(Tầng thực thi) của Spring IO Framework. Spring Boot làm cho Spring dễ dàng hơn trong việc thiết lập và phát triển ứng dụng. Với Spring Boot các cấu hình của Spring 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 triển khai lên các Web Server. Spring Boot có thể giải thích một cách đơn giản bởi hình minh họa dưới đây.

XML

Configuration

Embedded

HTTP Servers

(Tomcat, Jetty,..)

Spring

Framework

Spring

Boot

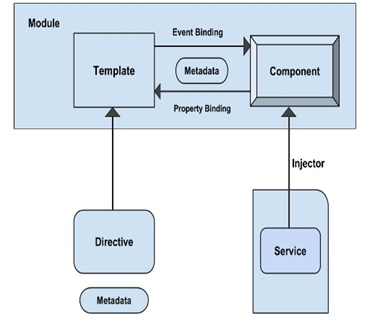
Dưới đây là các lợi ích của Spring Boot

1. Dễ dàng phát triển các ứng dụng dựa trên Spring với java hoặc Groovy
2. Giảm thiểu thời gian phát triển và tăng năng suất
3. Tránh việc viết các file cấu hình xml
4. Dễ dàng tương tác ứng dụng Spring Boot với hệ sinh thái của Spring như Spring JDBC, Spring ORM, Spring Data, Spring Security,…
5. Cung cấp các server nhúng (Embedded HTTP servers) như là Tomcat, Jetty,… để phát triển và tét các ứng dụng web nhanh chóng và dễ dàng
6. Cung cấp công cụ CLI( Command Line Interface) để phát triển và test các ứng dụng Spring Boot từ các dòng lệnh rất dễ dàng và nhanh chóng
7. Cung cấp nhiều plugin để làm việc với các cơ sở dữ liệu nhúng( embedded database)và các cơ sở dữ liệu lưu trữ trên bộ nhớ (in-memory Databases) một cách dễ dàng.
8. Cung cấp nhiều Plugin để phát triển và test các ứng dụng Spring Boot nhanh chóng, sử dụng các công cụ build như Maven và Gradle

## Giới thiệu về Angular 7



Tương tác giữa các thành phần trong Angular 7 được mô tả bằng hình vẽ sau:



*Hình 2: Tương tác giữa các thành phần trong Angular 7*

Các thành phần trực thuộc Module bao gồm :

1. Router : Sử dụng để điều hướng giữa các Component
2. Component : Các thành phần trực thuộc các Module con hoặc trực thuộc trực tiếp các AppModule của ứng dụng.
3. Template : Phần giao diện được sử dụng để tương tác với người dùng. Template sẽ chứa các Control để thực hiện tương tác và gắn kết với CSDL thông qua cơ chế Databinding.
4. Service : Sử dụng để gọi các Service ở phía Server, giao tiếp và truyền dữ liệu thông qua JSON

Một ứng dụng Angula 7 thường đi kèm với ngôn ngữ TypeScript (Ngôn ngữ cho việc xây dựng ứng dụng Angular7 ) .Typescript được phát triển để viết mã Javascript mang phong cách tính hướng đối tượng (Object-oriented). Bản thân Javacript nguyên thủy được thiết kế rất đơn giản và đi theo hướng lập trình chức năng (Functional programming) nên sẽ không có tính năng kế thừa, đa hình

Typescript mang một số tính năng cho Javascript như: kế thừa, kiểu dữ liệu, hàm khởi tạo,… Có thể nói, Typescript làm cho javascript mang phong cách rất "Java". Về mặt bản chất nhiều cú pháp của Typescript không được trình duyệt hiểu, vì vậy nó không thể chạy trực tiếp trên trình duyệt, vì vậy cần 1 bước đó là biên dịch (compile) mã Typescript thành Javascript thông thường. Vai trò của Typescript như là một ngôn ngữ trung gian, nó thường được biết tới như là Language specification (Đặc tả ngôn ngữ).

Angular đi kèm với một dependency là Rxjs. RxJS là một thư viện cho chương trình soạn bất đồng bộ và dựa trên sự kiện sử dụng trình tự có thể quan sát (**observable**). Các khái niệm cơ bản trong RxJS quản lý sự kiện không đồng bộ là:

1. **Observable**: đại diện cho khái niệm về một tập hợp các giá trị hoặc các sự kiện trong tương lai. Khi các giá trị hoặc sự kiện phát sinh trong tương lai, Observable sẽ điều phối nó đến Observer.
2. **Observer**: là một tập hợp các callbacks tương ứng cho việc lắng nghe các giá trị (**next**, **error**, hay **complete**) được gửi đến bởi Observable.
3. **Subscription**: là kết quả có được sau khi thực hiện một Observable, nó thường dùng cho việc hủy việc tiếp tục xử lý.

# CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG

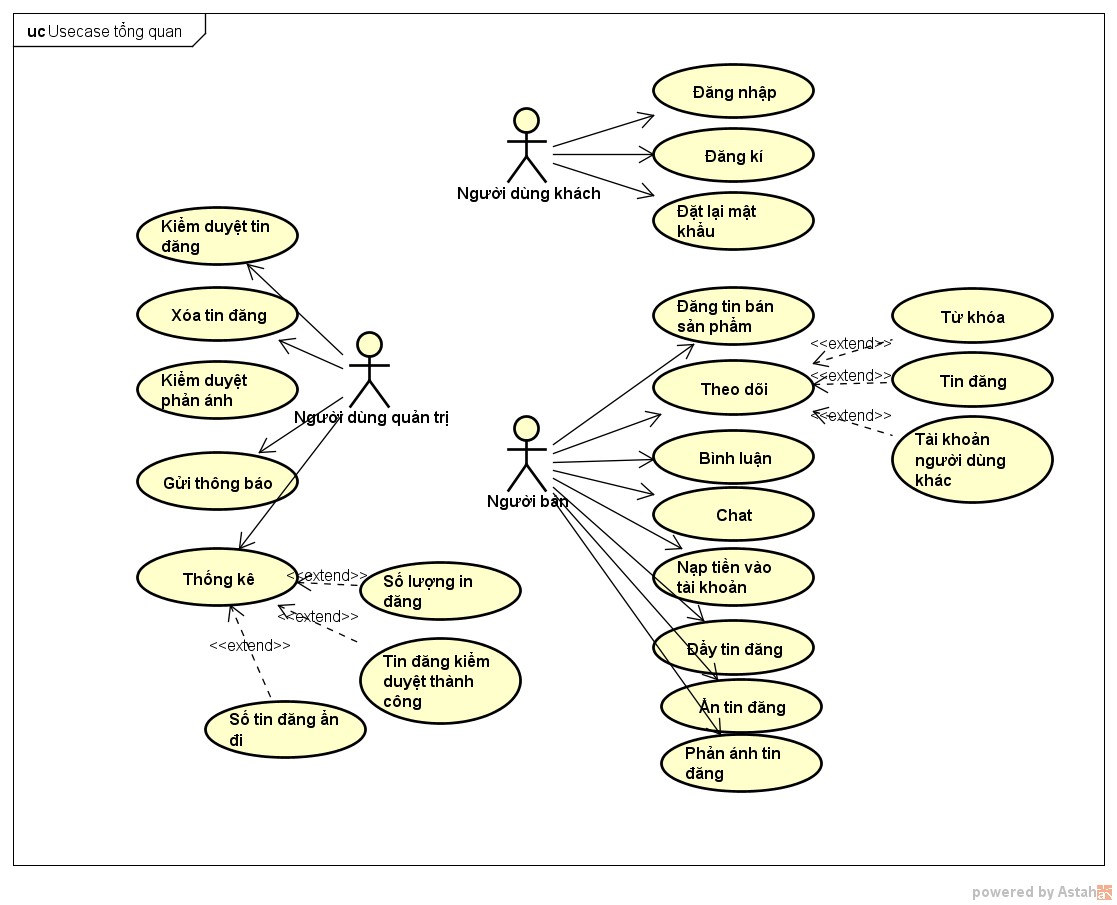


## Các đối tác

Hệ thống gồm các mức người dùng:

* **Guest** (người dùng khách,người dùng chưa có tài khoản): có thể thực hiện một số thao tác như xem danh sách tin đăng, tìm kiếm tin đăng,đăng kí tài khoản,đăng nhập hệ thống
* **User** (người dùng đã đăng ký tài khoản) : có thể thực hiện một số thao tác như đăng tin bán hàng,theo dõi người dùng khác, theo dõi tin đăng , nhắn tin , phản ánh tin đăng…
* **Admin**(đội ngũ quản lý giám sát hệ thống): có thể thực hiện một số thao tác như thống kê số lượng tin đăng , gửi thông báo…

## Các ca sử dụng





## Đặc tả ca sử dụng

### Đăng nhập

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Đăng nhập |
| **Tác nhân** | Guest |
| **Mục đích sử dụng** | Người dùng khách đăng nhập vào hệ thống |
| **Sự kiện kích hoạt** | Nhấp chọn button “Đăng nhập” |
| **Điều kiện tiên quyết** | Người dùng vào hệ thống |
| **Kết quả đầu ra** | Hệ thống trả về chuỗi jwt sau khi đăng nhập thành công |

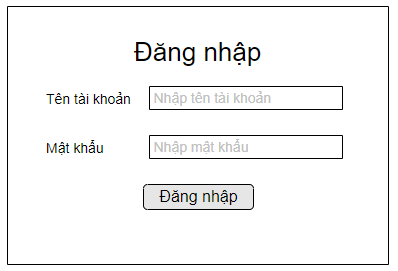
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Đăng nhập tài khoản | | | |
| 1 | Click button “Đăng nhập” |  |  |
|  |  | 2 | Hiển thị giao diện đăng nhập |
| 3 | Nhập tên tài khoản và mật khẩu |  |  |
|  |  | 4 | Kiểm tra thông tin đăng nhập |
|  |  | 5 | Sinh chuỗi JWT và trả về máy khách |
| **Kịch bản phụ: Tại bước 4 thông tin tên tài khoản không hợp lệ** | | | |
|  |  | 4.1 | Hệ thống yêu cầu đăng nhập lại |
| **Xử lí ngoại lệ : Tại bước 3 thông tin đăng nhập bị thiếu** | | | |
|  |  | 4.2 | Báo lỗi, yêu cầu đăng nhập lại |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Mã** | **Mô tả** |
| --- | --- | --- |
| 5 | BR**\_**1 | * + - * **Sinh chuỗi JWT:**   Chuỗi JWT chứa thông tin tài khoản và quyền truy cập của tài khoản |

#### View



*Hình 3: Minh họa giao diện đăng nhập*

### Đăng kí tài khoản

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Đăng kí tài khoản |
| **Tác nhân** | Guest |
| **Mục đích sử dụng** | Người dùng khách đăng kí tài khoản |
| **Sự kiện kích hoạt** | Nhấp chọn button “Đăng kí” |
| **Điều kiện tiên quyết** | Người dùng vào hệ thống |
| **Kết quả đầu ra** | Hệ thống lưu thông tin tài khoản,thông báo email kích hoạt tài khoản |

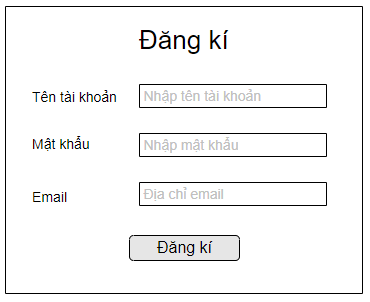
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Đăng kí tài khoản | | | |
| 1 | Click button “Đăng kí” |  |  |
|  |  | 2 | Hiển thị giao diện đăng kí |
| 3 | Nhập thông tin đăng kí và gửi |  |  |
|  |  | 4 | Kiểm tra thông tin đăng kí |
|  |  | 5 | * Lưu thông tin đăng kí * Gửi email kích hoạt tài khoản về email người dùng đã đăng kí |
| **Kịch bản phụ: Tại bước 3 người dùng thoát chức năng hoặc tải lại trang** | | | |
|  |  | 4.1 | Hệ thống yêu cầu xác nhận muốn tiếp tục từ người dùng |
| 5.1 | Chọn “OK” |  |  |
|  |  | 6.1 | Hệ thống thoát khỏi giao diện đăng tin hoặc tải lại trang |
| **Xử lí ngoại lệ 1: Tại bước 4 thông tin tên tài khoản bị trùng hoặc tài khoản email không đúng** | | | |
|  |  | 4.2 | * Hệ thống yêu cầu nhập lại thông tin * Không lưu thông tin đăng kí |
| **Xử lí ngoại lệ 2: Tại bước 3 người dùng nhập thiếu thông tin bắt buộc( gồm tên người dùng, mật khẩu , mail)** | | | |
|  |  | 4.1 | Báo lỗi ,yêu cầu nhập lại |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Mã** | **Mô tả** |
| --- | --- | --- |
| 4 | BR**\_**2 | * + - * **Lưu thông tin đăng kí:**   Lưu thông tin tài khoản với trạng thái chờ active |
| 5 | BR\_3 | * + - * **Gửi email kích hoạt:**       * Là một chuỗi JWT sinh ra từ thông tin tên tài khoản và mật khẩu của người dùng đăng kí |

#### View



*Hình 4: Minh họa giao diện đăng kí tài khoản*

### Đổi lại mật khẩu

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Đổi mật khẩu |
| **Tác nhân** | Guest |
| **Mục đích sử dụng** | Người dùng khách đổi lại mật khẩu của tài khoản đã đăng kí |
| **Sự kiện kích hoạt** | Nhấp chọn button “Đổi lại mật khẩu” |
| **Điều kiện tiên quyết** | Người dùng truy nhập hệ thống và vào màn hình đăng nhập |
| **Kết quả đầu ra** | Hệ thống thay đổi mật khẩu của tài khoản người dùng |

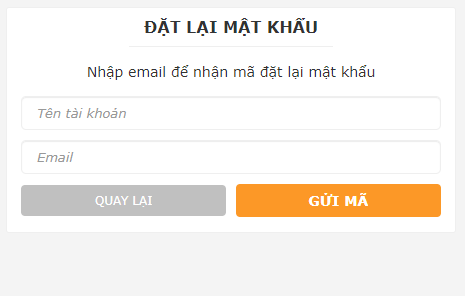
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Đổi mật khẩu | | | |
| 1 | Click button “Đổi mật khẩu” |  |  |
|  |  | 2 | Hiển thị giao diện nhập email |
| 3 | Gửi thông tin email |  |  |
|  |  | 4 | Gửi đường dẫn thay đổi mật khẩu về tài khoản email của người dùng |
| 5 | Người dùng vào đường dẫn thay đổi mật khẩu, nhập mật khẩu mới gửi về hệ thống |  |  |
|  |  | 6 | Lưu thông tin mật khẩu mới vào tài khoản của người dùng |
| **Kịch bản phụ: Tại bước 3 người dùng thoát chức năng hoặc tải lại trang** | | | |
|  |  | 4.1 | Hệ thống yêu cầu xác nhận muốn tiếp tục từ người dùng |
| 5.1 | Chọn “OK” |  |  |
|  |  | 6.1 | Hệ thống thoát khỏi giao diện đổi mật khẩu |
| **Xử lí ngoại lệ : Tại bước 4, gửi email bị lỗi** | | | |
|  |  | 4.1 | Báo lỗi, yêu cầu người dùng nhập lại thông tin email |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Mã** | **Mô tả** |
| --- | --- | --- |
| 4 | BR\_4 | * + - * **Gửi email thay đổi mật khẩu:**       * Là một chuỗi JWT sinh ra từ tên tài khoản người dùng và tên tài khoản email |

#### View



*Hình 5: Minh họa giao diện đổi lại mật khẩu*

### Tìm kiếm

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Tìm kiếm |
| **Tác nhân** | Guest |
| **Mục đích sử dụng** | Tìm kiếm tin đăng theo thể loại ,vị trí bán và nội dung người dùng nhập, lọc kết quả tìm kiếm theo ngày đăng và giá bán,sắp xếp kết quả tìm kiếm theo giá và thời gian đăng, phân trang kết quả tìm kiếm |
| **Sự kiện kích hoạt** | * Click “Tìm kiếm” trên thanh điều hướng * Click chọn thể loại ở trang chủ |
| **Điều kiện tiên quyết** | Người dùng vào hệ thống |
| **Kết quả đầu ra** | Trả về kết quả tìm kiếm |

#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Tìm kiếm | | | |
| 1 | Click “Tìm kiếm” |  |  |
|  |  | 2 | Hiển thị giao diện tìm kiếm |
| 3 | Nhập nội dung tìm kiếm,chọn thể loại ,chọn địa điểm,… |  |  |
|  |  | 4 | * Lấy ra danh sách tin đăng trang thứ nhất và tổng số trang tương ứng với nội dung tìm kiếm. * Cập nhật giao diện tìm kiếm |
| **Kịch bản phụ: Không có** | | | |
|  |  |  |  |
| **Xử lí ngoại lệ: Không có** | | | |
|  |  |  |  |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
|  |  |  |

#### View



*Hình 6: Minh họa giao diện tìm kiếm*

### Đăng tin

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Đăng tin |
| **Tác nhân** | Người bán |
| **Mục đích sử dụng** | Đăng tin rao bán lên hệ thống, một tin đăng phải thuộc một danh mục, địa điểm bán và người bán.  Người bán phải cung cấp các thông tin như tiêu đề , giá sản phẩm. tình trạng sản phẩm …. |
| **Sự kiện kích hoạt** | Nhấp chọn button “Đăng tin” |
| **Điều kiện tiên quyết** | Người dùng đã đăng nhập thành công vào hệ thống |
| **Kết quả đầu ra** | Hệ thống lưu tin đăng, điều hướng đến danh sách tin đăng chờ phê duyệt của người dùng |

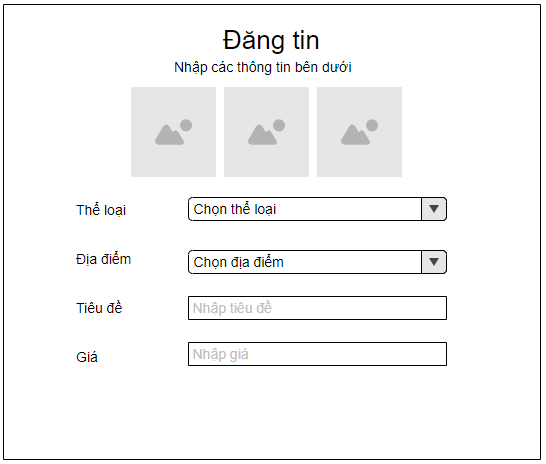
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Đăng tin | | | |
| 1 | Click button “Đăng tin” |  |  |
|  |  | 2 | Hiển thị giao diện đăng tin |
| 3 | Nhập thông tin sản phẩm,gửi về hệ thống |  |  |
|  |  | 4 | * Lưu tin đăng * Điều hướng đến danh sách tin đăng chờ kiểm duyệt của người dùng |
| **Kịch bản phụ: Tại bước 3 người dùng thoát chức năng hoặc tải lại trang** | | | |
|  |  | 4.1 | Hệ thống yêu cầu xác nhận muốn tiếp tục từ người dùng |
| 5.1 | Chọn “OK” |  |  |
|  |  | 6.1 | Hệ thống thoát khỏi giao diện đăng tin hoặc tải lại trang |
| **Xử lí ngoại lệ: Tại bước 3 người dùng nhập thiếu thông tin bắt buộc** | | | |
|  |  | 4.2 | Hệ thống yêu cầu nhập đủ thông tin,không gửi tin đăng về hệ thống |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 4 | BR\_5 | * + - * **Lưu tin đăng:**   Tin đăng được lưu với trạng thái chờ kiểm duyệt, gồm các thông tin bắt buộc như ngày đăng, người đăng,tên sản phẩm, giá,thể loại và địa điểm giao dịch |

#### View



*Hình 7: Minh họa giao diện đăng tin*

### Bình luận

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Bình luận |
| **Tác nhân** | Người bán |
| **Mục đích sử dụng** | Gửi bình luận về 1 bài đăng |
| **Sự kiện kích hoạt** | Chọn mục bình luận trong màn hình chi tiết của một bài đăng |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập thành công vào hệ thống * Ở màn hình chi tiết của tin đăng |
| **Kết quả đầu ra** | Hệ thống lưu bình luận |

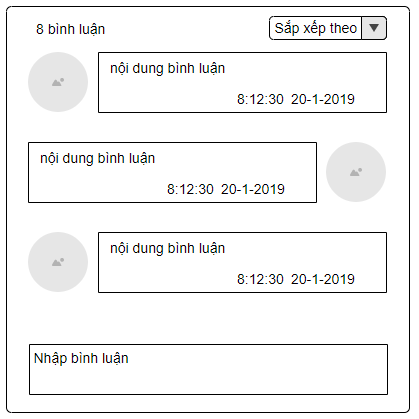
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Bình luận | | | |
| 1 | Chọn xem chi tiết 1 tin đăng |  |  |
|  |  | 2 | Hiển thị giao diện thông tin chi tiết của 1 tin đăng |
| 3 | Gửi bình luận |  |  |
|  |  | 4 | * Hệ thống lưu bình luận * Cập nhật giao diện bình luận |
| **Kịch bản phụ: Không có** | | | |
|  |  |  |  |
| **Xử lí ngoại lệ: Không có** | | | |
|  |  |  |  |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 4 | BR**\_**5 | * + - * **Lưu bình luận:**   Bình luận được lưu gồm các thông tin bắt buộc như thời gian, thông tin tài khoản bình luận,nội dung,… |

#### View



*Hình 8: Minh họa giao diện bình luận*

### Nhắn tin

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Nhắn tin |
| **Tác nhân** | Người bán |
| **Mục đích sử dụng** | Nhắn tin với một người dùng đăng nhập khác |
| **Sự kiện kích hoạt** | Nhấp chọn button “Nhắn tin với người bán” |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập thành công vào hệ thống * Đang ở giao diện chi tiết của sản phẩm |
| **Kết quả đầu ra** | Hệ thống lưu tin nhắn, gửi tin nhắn đến tài khoản đích |

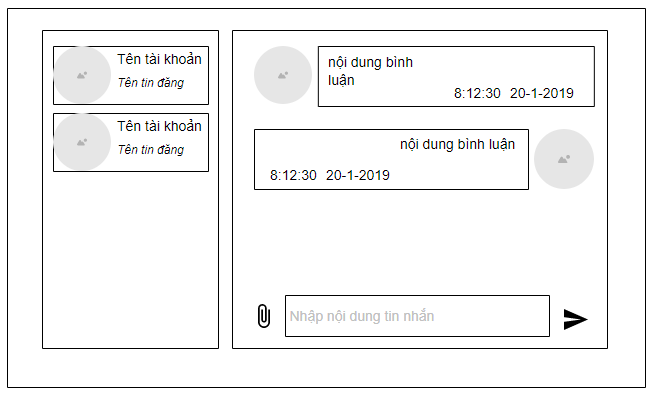
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Chat | | | |
| 1 | Click button “Nhắn tin với người bán” |  |  |
|  |  | 2 | Hiển thị giao diện nhắn tin |
| 3 | Gửi nội dung tin nhắn |  |  |
|  |  | 4 | * Hệ thống lưu tin nhắn * Gửi đến tài khoản mà người dùng đang nhắn tin |
| **Kịch bản phụ: Tại bước 3 người dùng thoát chức năng hoặc tải lại trang** | | | |
|  |  | 4.1 | Hệ thống yêu cầu xác nhận muốn tiếp tục từ người dùng |
| 5.1 | Chọn “OK” |  |  |
|  |  | 6.1 | Hệ thống thoát khỏi giao diện đăng tin hoặc tải lại trang |
| **Xử lí ngoại lệ: Không có** | | | |
|  |  |  |  |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 4 | BR\_6 | * + - * **Lưu dòng chat:**   Dòng chat được lưu gồm các thông tin bắt buộc như thời gian, người gửi,người nhận, nội dung,tin đăng…  Dùng cơ chế websocket để gửi |

#### View



*Hình 9: Minh họa giao diện chát*

### Theo dõi tài khoản

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Theo dõi tài khoản |
| **Tác nhân** | Người bán |
| **Mục đích sử dụng** | Theo dõi tài khoản của người dùng khác , hệ thống sẽ thông báo cho người dùng khi tài khoản được theo dõi đăng tin lên hệ thống |
| **Sự kiện kích hoạt** | Nhấp chọn button “Theo dõi tài khoản” |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập thành công vào hệ thống * Đang ở giao diện trang cá nhân của một người dùng khác |
| **Kết quả đầu ra** | Hệ thống lưu thông tin tài khoản mà người dùng theo dõi, thông báo đã theo dõi thành công |

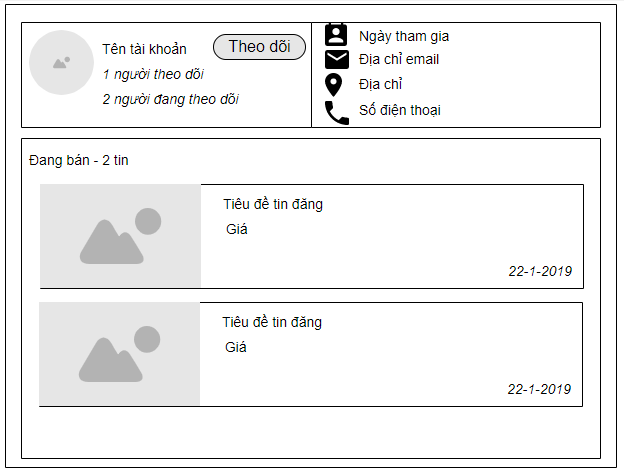
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Theo dõi tài khoản | | | |
| 1 | Click button “Theo dõi tài khoản” |  |  |
|  |  | 2 | Hệ thống lưu thông tin tài khoản mà người dùng theo dõi, thông báo đã theo dõi thành công |
|  |  | 3 | Lấy thông tin chi tiết của tài khoản được theo dõi và trả về máy khách |
| **Kịch bản phụ: Không có** | | | |
|  |  |  |  |
| **Xử lí ngoại lệ: Không có** | | | |
|  |  |  |  |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 4 | BR\_7 | * + - * **Lưu thông tin tài khoản theo dõi:**   Thông tin tài khoản được theo gõi sẽ lưu gồm người theo dõi, người được theo dõi |

#### View



*Hình 10: Minh họa trang cá nhân người dùng*

### Nạp thẻ cào

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Nạp thẻ cào |
| **Tác nhân** | Người bán |
| **Mục đích sử dụng** | Người dùng nạp tiền vào tài khoản hệ thống để phục vụ chức năng đẩy tin |
| **Sự kiện kích hoạt** | Nhấp chọn button “Nạp tiền” |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập thành công vào hệ thống * Đang ở giao diện quản lí thanh toán |
| **Kết quả đầu ra** | Hệ thống cộng dồn số tiền trong thẻ cào vào tài khoản của người dùng, lưu lịch sử nạp tiền |

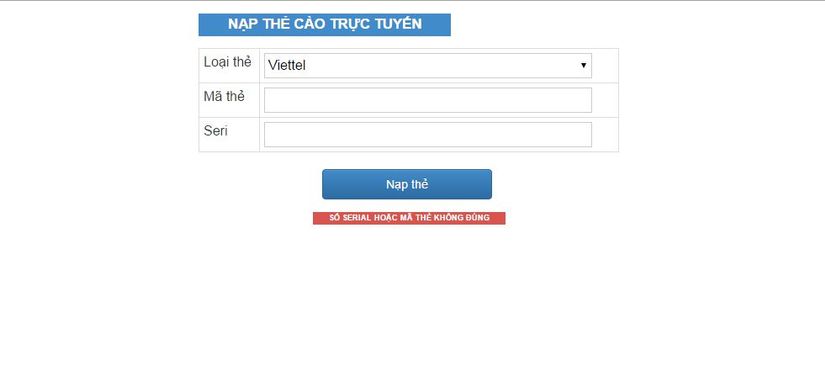
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Nạp thẻ cào | | | |
| 1 | Nhập thông tin thẻ, và gửi về hệ thống |  |  |
|  |  | 2 | Hệ thống gọi đến api của trang [https://napthengay.com](https://napthengay.com/) để lấy số tiền của thẻ cào |
|  |  | 4 | Cộng dồn số tiền trong thẻ cào vào tài khoản người dùng, lưu lịch sử nạp tiền |
|  |  | 5 | Thông báo lại cho người dùng |
| **Kịch bản phụ: Không có** | | | |
|  |  |  |  |
| **Xử lí ngoại lệ: Trang** [**https://napthengay.com**](https://napthengay.com/) **báo lỗi** | | | |
|  |  | 2.1 | Thông báo lỗi , yêu cầu người dùng thực hiện lại chức năng |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 2 | BR**\_**8 | * + - * **Gọi API trang najpthengay.com:**   Các tham số đầu vào bao gồm:   |  |  | | --- | --- | | Tên tham số | Mô tả | | merchant\_id | Mã MerchantID đăng ký trên Napthengay | | api\_email | Email đăng ký tài khoản trên Napthengay.com | | trans\_id | Mã TransactionId xác định giao dịch bên khách hàng | | card\_id | Loại thẻ cào là 1 trong số loại (VINA, MOBI, VIETTEL, VTC, GATE) | | card\_value | Mệnh giá thẻ cào cần nạp (Chọn chính xác,sai mệnh giá sẽ không được nhận tiền) | | pin\_field | Mã Pin thẻ cào dưới lớp bạc | | seri\_field | Mã Seri thẻ | | algo\_mode | Thuật toán mã hóa tạo data\_sign (md5 hoặc hmac) | | data\_sign | Trường mô tả tính toàn vẹn dữ liệu gửi lên (Mô tả bên dưới) | |
| 3 | BR\_9 | * + - * **Lưu lịch sử nạp tiền:**       * Thông tin bao gồm thời gian, tài khoản người dùng và số tiền nạp vào, mã thẻ |

#### View



*Hình 11: Minh họa giao diện nạp thẻ cào*

### Đẩy tin đăng

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Đẩy tin đăng |
| **Tác nhân** | Người bán |
| **Mục đích sử dụng** | Đẩy tin đăng để tin đăng được xuất hiện tại mục sản phẩm ưu tiên, người dùng có thể chọn số lần để tin đăng được đẩy |
| **Sự kiện kích hoạt** | Nhấp chọn button “Đẩy tin” |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập thành công vào hệ thống * Đang ở giao diện quản lí tin đăng |
| **Kết quả đầu ra** | Tin đăng được đẩy sẽ xuất hiện ở mục ưu tiên |

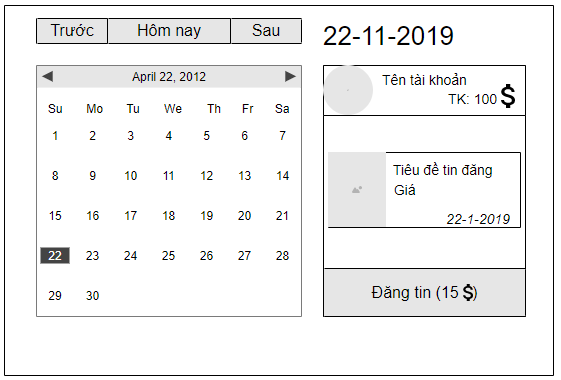
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Đẩy tin đăng | | | |
| 1 | Click button “Đẩy tin” |  |  |
|  |  | 2 | Hiển thị giao diện tiện ích đẩy tin |
| 3 | Chọn ngày để đẩy tin |  |  |
|  |  | 4 | Hệ thống thông báo số tiền cần thanh toán để đẩy tin |
| 5 | Thanh toán |  |  |
|  |  | 6 | Hệ thống trừ số tiền của tài khoản người dùng |
| **Kịch bản phụ: Số tiền trong tài khoản người dùng ít hơn số tiền để thanh toán** | | | |
|  |  | 6.1 | Thông báo lỗi |
| **Xử lí ngoại lệ:Không có** | | | |
|  |  |  |  |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
|  |  |  |

#### View



*Hình 12: Minh họa giao diện đẩy tin đăng*

### Ẩn tin

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Ẩn tin đăng |
| **Tác nhân** | Người bán |
| **Mục đích sử dụng** | Ẩn tin đăng để tin đăng không xuất hiện trong danh dách các sản phẩm đang bán |
| **Sự kiện kích hoạt** | Nhấp chọn button “Ẩn tin” |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập thành công vào hệ thống * Đang ở giao diện quản lí tin đăng |
| **Kết quả đầu ra** | Tin đăng sẽ không xuất hiện khi người dùng khác vào hệ thống |

#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Ẩn tin | | | |
| 1 | Click button “Ẩn tin” |  |  |
|  |  | 2 | Hỏi người dùng có muốn tiếp tục thao tác |
| 3 | Chọn Ok |  |  |
|  |  | 4 | Cập nhật trạng thái tin đăng |
| **Kịch bản phụ: Không có** | | | |
|  |  |  |  |
| **Xử lí ngoại lệ:Không có** | | | |
|  |  |  |  |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 4 | BR**\_**10 | * + - * **Ẩn tin đăng:**   Thay đổi trạng thái của tin đăng từ “đang bán” sang “đã bán”, tin đăng đã bán sẽ không xuất hiện trong kết quả tìm kiếm từ hệ thống |

#### View

### Phản ánh tin đăng

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Phản ánh tin đăng |
| **Tác nhân** | Người bán |
| **Mục đích sử dụng** | Người dùng phản ánh tin đăng không hợp lệ |
| **Sự kiện kích hoạt** | Nhấp chọn button “Phản ánh tin đăng” |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập thành công vào hệ thống * Đang ở giao diện thông tin chi tiết của tin đăng hoặc giao diện tìm kiếm |
| **Kết quả đầu ra** | Thông tin phản ánh sẽ được lưu lại |

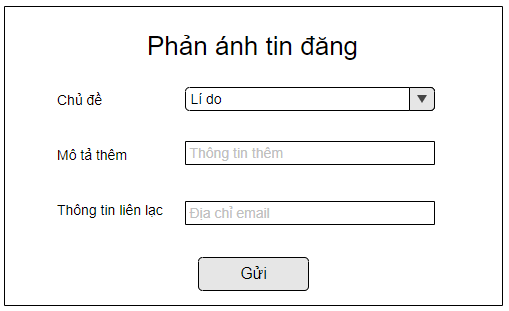
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Phản ánh tin đăng | | | |
| 1 | Click button “Phản ánh” |  |  |
|  |  | 2 | Hiển thị giao diện tiện phản ánh |
| 3 | Chọn lí do phản ánh(chọn 1 trong 3 lí do sau : sản phẩm trùng lặp,hàng đã bán, không liên lạc được, thông tin không hợp lệ) và gửi vê hệ thống |  |  |
|  |  | 4 | Lưu lại thông tin phản ánh |
| **Kịch bản phụ: Người dùng tải lại trang hoặc chuyển sang tác vụ khác** | | | |
| 3.1 | Tải lại trang hoặc chuyển tác vụ |  |  |
|  |  | 4.1 | Xác nhận người dùng có muốn tiếp tục không |
| 5.1 | Chọn “Có” |  |  |
|  |  | 6.1 | Tải lại trang hoặc chuyển tác vụ khác |
| **Xử lí ngoại lệ:Không có** | | | |
|  |  |  |  |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 4 | BR**\_**11 | * + - * **Lưu thông tin phản ánh:**   Thông tin phản ánh được lưu bao gồm mã tin đăng, thông tin người dùng , thời gian và lí do phản ánh |

#### View



*Hình 13: Minh họa giao diện phản ánh tin đăng*

### Theo dõi thống kê

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Theo dõi thống kê |
| **Tác nhân** | Người bán |
| **Mục đích sử dụng** | Xem các thông số thống kê bao gồm số người theo dõi,vị trí trang của tin đăng, lượt truy cập hiển thị thành biểu đồ(biểu đồ đường mô tả lượt truy cập các ngày trong tháng) của một tin đăng đã được chấp thuận |
| **Sự kiện kích hoạt** | Nhấp chọn button “Thống kê” |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập thành công vào hệ thống * Đang ở giao diện quản lí tin đăng |
| **Kết quả đầu ra** | Hiển thị các thông số thống kê bao gồm số người theo dõi,vị trí trang của tin đăng, lượt truy cập hiển thị thành biểu đồ(biểu đồ đường mô tả lượt truy cập các ngày trong tháng) |

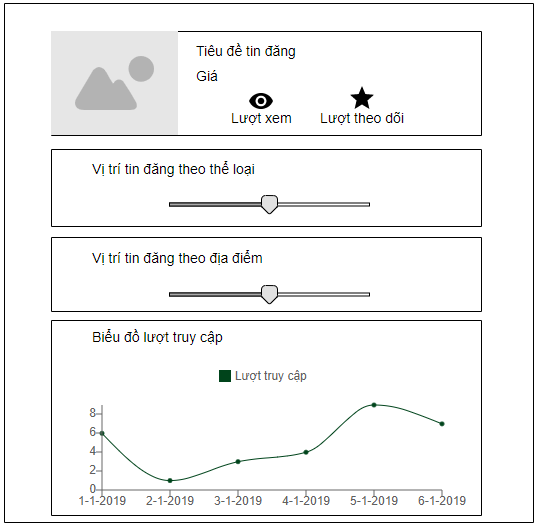
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Thống kê | | | |
| 1 | Click button “Thống kê” |  |  |
|  |  | 2 | Hiển thị số người đang theo dõi và vị trí trang hiện tại của tin đăng |
| 3 | Chọn tháng |  |  |
|  |  | 4 | * Thu thập số lượng truy cập của tin đăng của các ngày trong tháng đã được chọn * Hiển thị số liệu theo biểu đồ đường |
| **Kịch bản phụ: Không có** | | | |
|  |  |  |  |
| **Xử lí ngoại lệ:Không có** | | | |
|  |  |  |  |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
|  |  |  |

#### View



*Hình 14: Minh họa giao diện thống kê lượt truy cập của tin đăng*

### Kiểm duyệt tin đăng

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Kiểm duyệt tin đăng |
| **Tác nhân** | Người dùng quản trị |
| **Mục đích sử dụng** | Kiểm duyệt tính chính xác thông tin trong tin đăng của người dùng |
| **Sự kiện kích hoạt** | Chọn tin đăng thuộc danh sách tin đăng chờ kiểm duyệt |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập thành công vào hệ thống với tài khoản có quyền quản trị * Người dùng đang ở giao diện “Quản trị tin đăng” |
| **Kết quả đầu ra** | Cập nhật trạng thái của tin đăng,thông báo email đến người dùng |

#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Kiểm duyệt tin đăng | | | |
| 1 | Chọn tin đăng thuộc danh sách tin đăng đang chờ kiểm duyệt |  |  |
|  |  | 2 | Hiển thị giao diện kiểm duyệt và thông tin tin đăng |
| 3 | Chọn “Chấp nhận” hoặc “Từ chối” |  |  |
|  |  | 4 | Xác nhận người dùng muốn tiếp tục không |
| 5 | Chọn “Có” |  |  |
|  |  | 6 | * Cập nhật trạng thái tin đăng * Gửi email đến tài khoản sở hữu tin đăng |
|  |  | 7 | Thông báo thao tác thành công |
| **Kịch bản phụ: Tại bước 5, Admin hủy thao tác** | | | |
| 5.1 | Chọn “Hủy” |  |  |
|  |  | 6.1 | Quay lại giao diện kiểm duyệt tin đăng |
| **Xử lí ngoại lệ:Không có** | | | |
|  |  |  |  |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 8 | BR**\_**13 | * + - * **Cập nhật trạng thái tin đăng:** * Nếu tin đăng được chấp nhận,trạng thái tin đăng sẽ là 1 * Nếu tin đăng không được chấp nhận, trạng thái tin đăng sẽ là 2 |

#### View



*Hình 15: Minh họa giao diện kiểm duyệt tin đăng*

### Kiểm duyệt phản ánh

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Kiểm duyệt phản ánh |
| **Tác nhân** | Người quản trị |
| **Mục đích sử dụng** | Kiểm duyệt tính lại tính chính xác của tin đăng bị phản ánh bởi người dùng |
| **Sự kiện kích hoạt** | Chọn tin đăng thuộc danh sách tin đăng bị phản ánh |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập thành công vào hệ thống với tài khoản có quyền quản trị * Người dùng đang ở giao diện “Quản trị tin đăng” |
| **Kết quả đầu ra** | Cập nhật trạng thái của tin đăng,gửi email về người dùng |

#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Kiểm duyệt thông tin phản ánh | | | |
| 1 | Chọn tin đăng |  |  |
|  |  | 2 | Hiển thị giao diện phản ánh,bao gồm thông tin và danh sách phản ánh của tin đăng |
| 3 | Chọn “Từ chối” |  |  |
|  |  | 4 | Hệ thống yêu cầu người dùng có muốn tiếp tục không |
| 5 | Chọn “Có” |  |  |
|  |  | 6 | Cập nhật trạng thái tin đăng thành “Bị từ chối” |
| **Kịch bản phụ: Tại bước 5, Admin hủy thao tác** | | | |
| 5.1 | Chọn “Hủy” |  |  |
|  |  | 8.1 | Quay lại màn hình phản ánh của tin đăng |
| **Kịch bản phụ: Không có** | | | |
|  |  |  |  |
|  |  |  |  |
| **Xử lí ngoại lệ:Không có** | | | |
|  |  |  |  |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
|  |  |  |

#### View

### Thống kê số lượng tin đăng

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Thống kê số lượng tin đăng |
| **Tác nhân** | Người dùng quản trị |
| **Mục đích sử dụng** | Thống kê số lượng tin đăng được chấp nhận,tin đăng đang chờ phê duyệt,tin đăng bị từ chối theo các các ngày trong tháng , các tháng trong năm và theo từng năm |
| **Sự kiện kích hoạt** | Nhấp chọn button “Tin đăng” tại thanh điều hướng (sidebar) |
| **Điều kiện tiên quyết** | Người dùng đã đăng nhập thành công vào hệ thống với tài khoản có quyền quản trị |
| **Kết quả đầu ra** | Hiển thị thông tin thống kê theo biểu đồ đường |

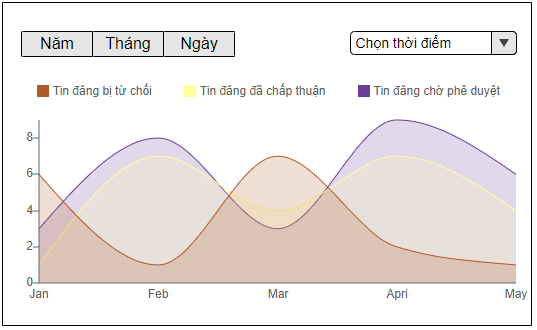
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Thống kê | | | |
| 1 | Nhấp chọn button “Tin đăng” |  |  |
|  |  | 2 | * Thu thập số lượng tin đăng được chấp thuận,chờ phê duyệt và bị từ chối theo từng năm * Hiển thị số liệu thống kê theo biểu đồ đường |
| 3 | * Chọn thống kê theo tháng * Chọn năm muốn thống kê |  |  |
|  |  | 4 | * Thu thập số lượng tin đăng được chấp thuận,chờ phê duyệt và bị từ chối trong 12 tháng của năm người dùng chọn * Hiển thị số liệu thống kê theo biểu đồ đường |
| 5 | * Chọn thống kê theo ngày * Chọn tháng muốn thống kê |  |  |
|  |  | 6 | * Thu thập số lượng tin đăng được chấp thuận,chờ phê duyệt và bị từ chối theo từng ngày của tháng người dùng chọn * Hiển thị số liệu thống kê theo biểu đồ đường |
| **Kịch bản phụ: Không có** | | | |
|  |  |  |  |
| **Xử lí ngoại lệ:Không có** | | | |
|  |  |  |  |

#### Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
|  |  |  |

#### View



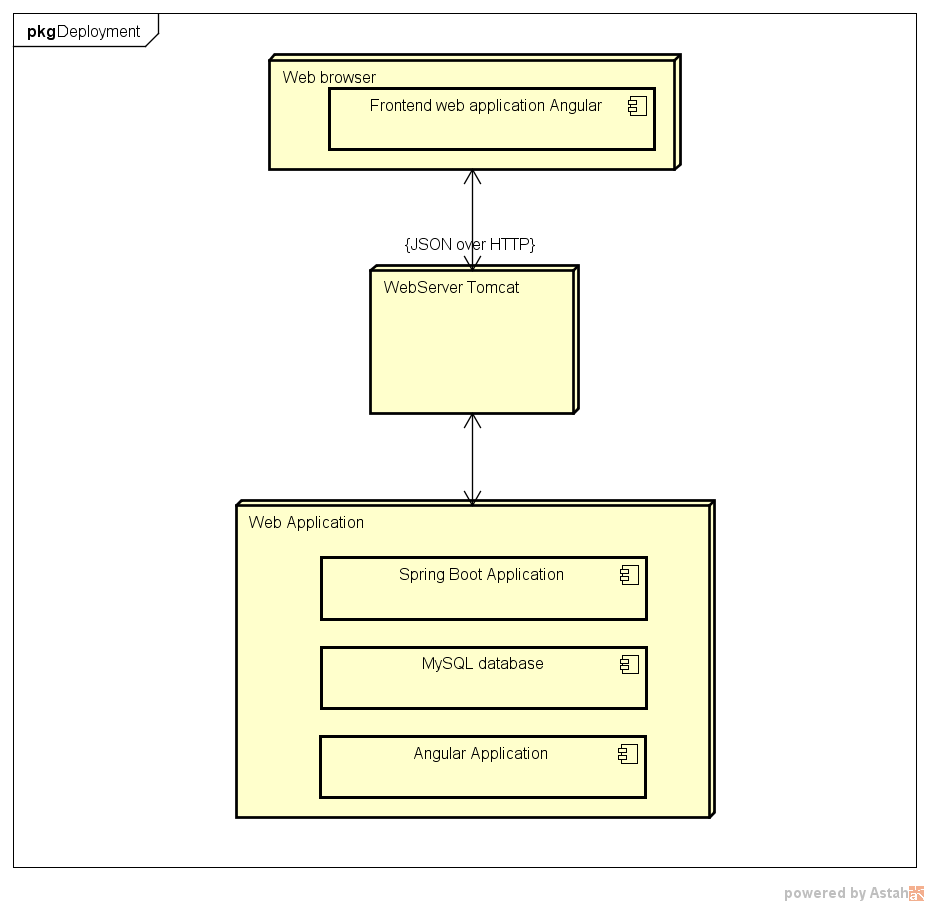
*Hình 16: Minh họa giao diện thống kê số lượng tin đăng*

# CHƯƠNG 4: THIẾT KẾ HỆ THỐNG



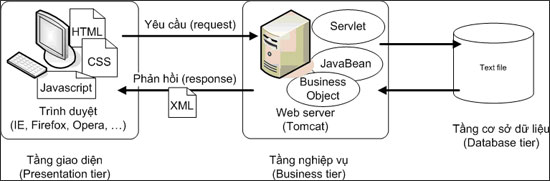
## Kiến trúc hệ thống

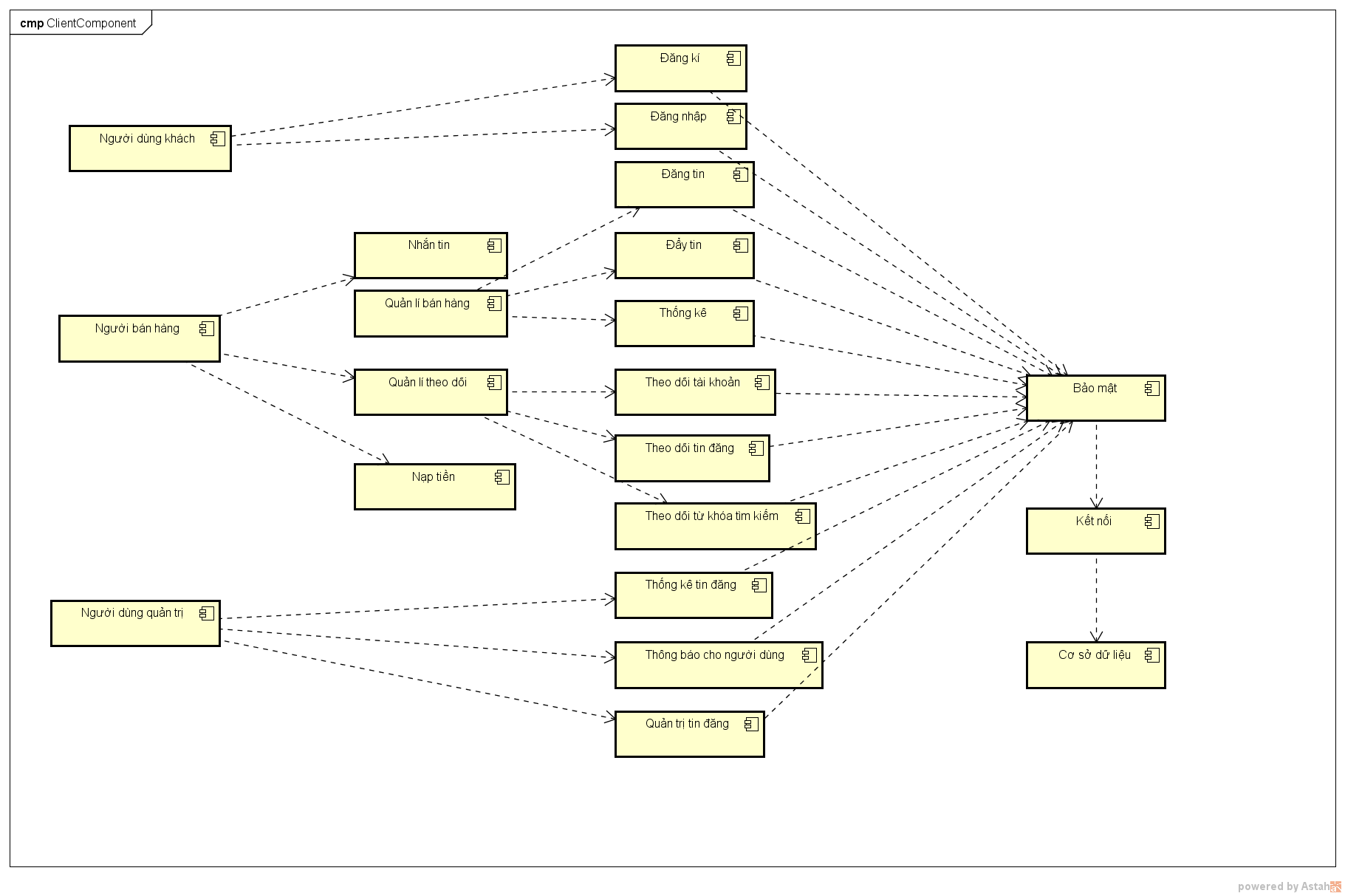
### Kiến trúc triển khai



*Hình 17: Kiến trúc triển khai của hệ thống*

### Kiến trúc logic





*Hình 18: Kiến trúc logic của ứng dụng*

### Kiến trúc mã nguồn

### Các kĩ thuật

#### *RxJS*

Angular Observable sử dụng như một giao diện để xử lý nhiều hoạt động không đồng bộ phổ biến như gửi dữ liệu quan sát được từ thành phần con đến thành phần cha, xử lý các yêu cầu và phản hồi AJAX hoặc HTTP, lắng nghe và phản hồi đầu vào của người dùng trong Angular Router và Forms. RxJS (Phần mở rộng phản ứng cho JavaScript) là một thư viện để xử lí bất đồng bộ bằng cách sử dụng **observable** dễ dàng soạn thảo mã không đồng bộ.Sau đây là một vài kĩ thuật sử dụng RxJS trọng một ứng dụng Angular

1. Ứng dụng RxJS để lắng nghe sự kiện
2. Ứng dụng RxJS trong việc gọi API của HttpClient
3. Trao đổi dữ liệu giữa các thành phần( component)
4. Xử lí dữ liệu của trong Router
5. Ứng dụng RxJS để xử lí Form

#### *Auth0 JWT*

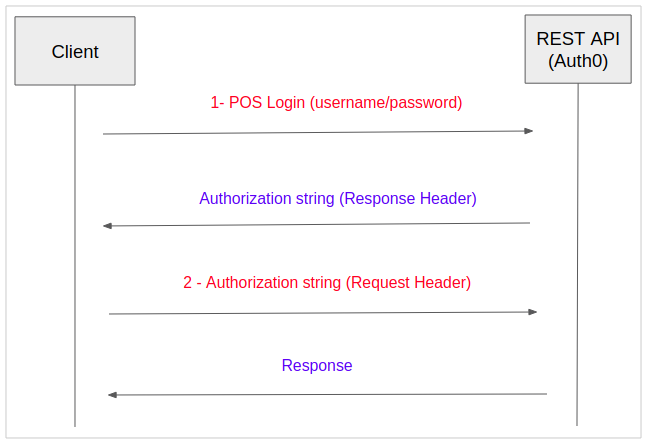
Giả sử rằng có một RESTful API được viết trên Spring Boot, các Client (ứng dụng khác) có thể gọi đến RESTful API đó và nhận kết quả về.

Tuy nhiên không phải tất cả các các RESTful API đều có thể công khai, vì tính nhạy cảm của nó, vì vậy cần phải bảo mật chúng. Có một vài kỹ thuật để bảo mật RESTful API như :

* Bảo mật RESTful API với Basic Authentication (Xác thực cơ bản)
* Bảo mật RESTful API với JWT (JSON Web Token)

Với các REST API được bảo mật với Auth0, sẽ trải qua các bước cơ bản sau:

* Bước 1: Client phải gửi một request (yêu cầu) đăng nhập chứa username/password, và nhận được phản hồi là một "Authorization String" (Chuỗi ủy quyền) đính kèm trên Response Header.
* Bước 2: Sau khi có "Authorization String", đính kèm nó trên Request Header để gọi đến REST API.



*Hình 19: Luồng xử lí Auth0*

#### *Websocket*

1. WebSocket là gì?

Websocket là một giao thức truyền thông (communication protocol), nó giúp thành lập một kênh liên lạc 2 chiều (two-way communication channel) giữa client và server. Một giao thức truyền thông quen thuộc đó là HTTP

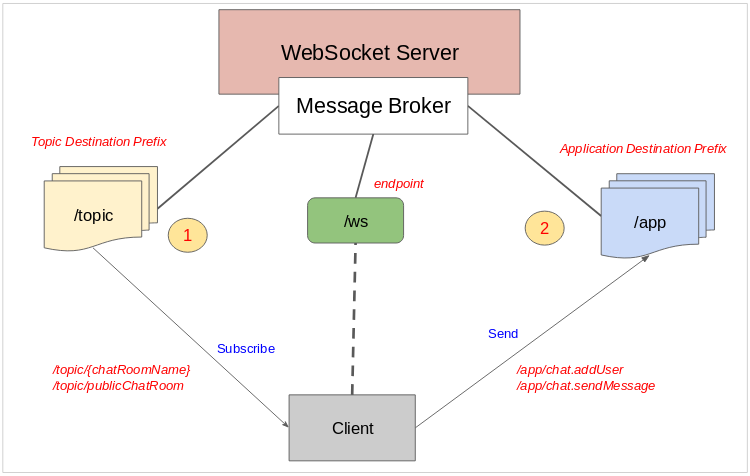
* HTTP (Hypertext Transfer Protocol): Là một giao thức request-response (Yêu cầu - Đáp ứng). Client (Trình duyệt) muốn một cái gì đó, nó gửi yêu cầu tới Server, và Server đáp ứng yêu cầu đó. HTTP là một giao thức truyền thông một chiều, mục đích ở đây là để giải quyết "Làm thế nào để tạo một yêu cầu tại client, và làm thế nào để đáp ứng yêu cầu của client".
* WebSocket: Không phải là giao thức request-response (Yêu cầu - Đáp ứng), nơi mà chỉ Client có mới thể gửi yêu cầu tới Server. Một khi kết nối với giao thức WebSocket được thiết lập, client & server có thể gửi dữ liệu tới cho nhau, cho tới khi kết nối ở tầng dưới là TCP được đóng lại. WebSocket về cơ bản rất giống với khái niệm TCP Socket, sự khác biệt là WebSocket được tạo ra để sử dụng cho các ứng dụng Web.

1. STOMP

STOMP (Streaming Text Oriented Messaging Protocol): (Giao thức luồng văn bản theo hướng tin nhắn) là một giao thức truyền thông, một nhánh của WebSocket. Khi client và server liên lạc với nhau theo giao thức này chúng sẽ chỉ gửi cho nhau các dữ liệu dạng tin nhắn văn bản.

1. Một vài khái niệm liên quan tới WebSocket trong Spring

* MessageBroker: MessageBroker là một chương trình trung gian, nó tiếp nhận các tin nhắn được gửi đến trước khi phân phát tới các địa chỉ cần thiết. Vì vậy cần nói với Spring bật (enable) chương trình này cho nó hoạt động. Hình dưới đây mô tả cấu trúc của MessageBroker:



MessageBroker phơi bầy ra một endpoint (Điểm cuối) để client có thể liên lạc và hình thành một kết nối. Để liên lạc client sử dụng thư viện SockJS để làm việc này.Đồng thời MessageBroker cũng phơi bày ra 2 loại điểm đến (1) và (2)

* Điểm đến (1) là các chủ đề (topic) mà client có thể "đăng ký theo dõi" (subscribe), khi một chủ đề có tin nhắn, các tin nhắn sẽ được gửi đến cho những client đã đăng ký chủ đề này.
* Điểm đến (2) là các nơi mà client có thể gửi tin nhắn tới WebSocket Server.

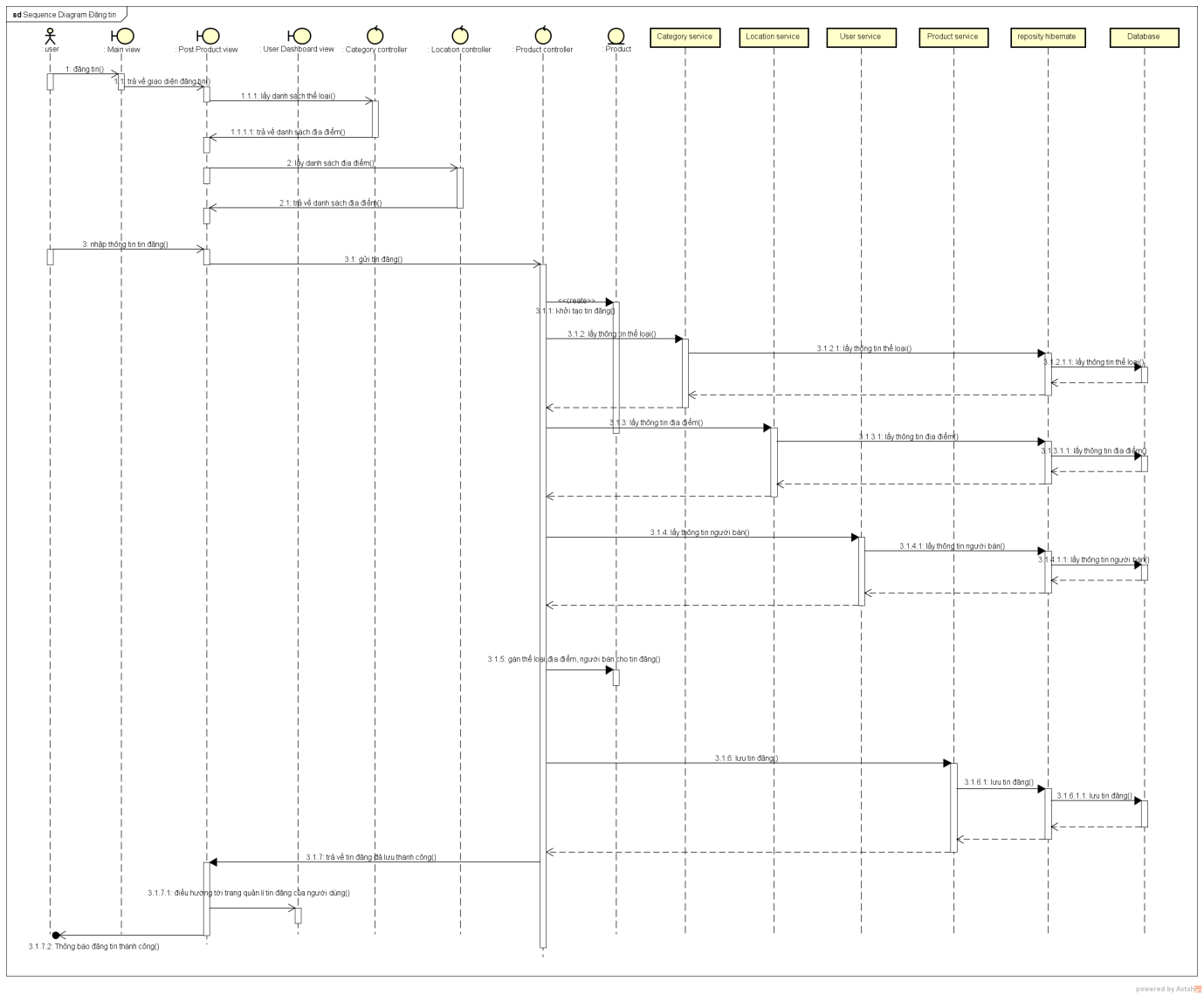
1. SockJS

SockJS là một thư viện JavaScript.Không phải tất cả các trình duyệt đều hỗ trợ giao thức WebSocket. Vì vậy SockJS là một tùy chọn dự phòng , nó sẽ được kích hoạt cho các trình duyệt không hỗ trợ WebSocket.

## Thiết kế ca sử dụng

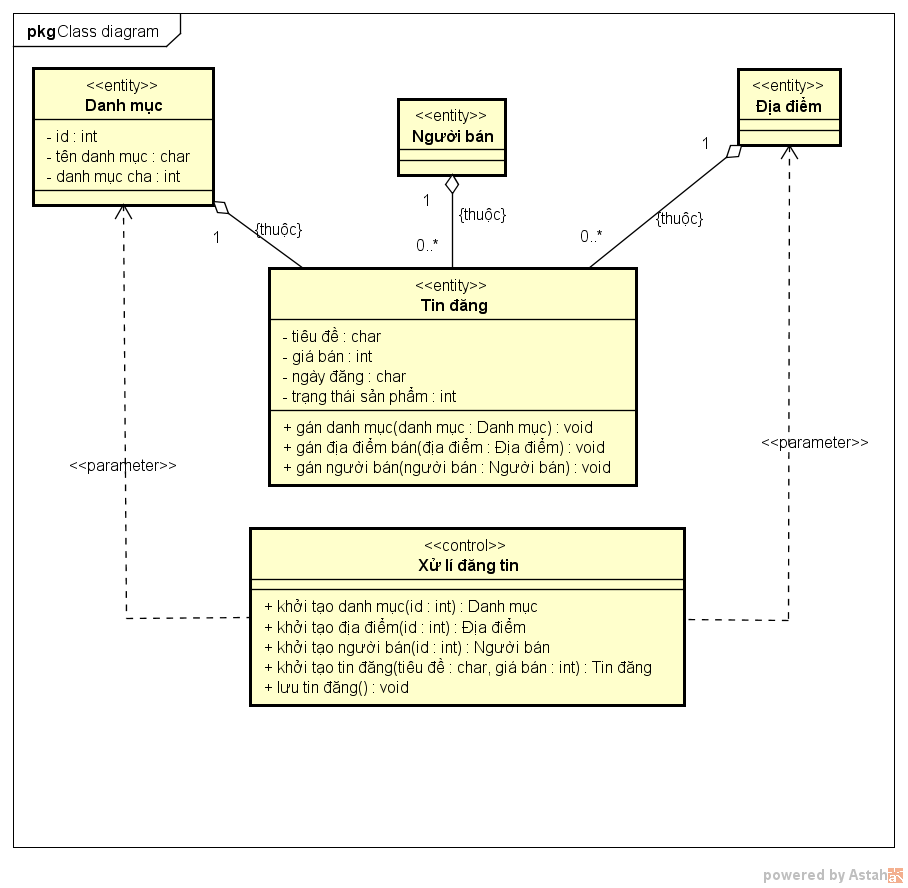
### Đăng tin

#### *Biểu đồ trình tự*



*Biểu đồ 1: Biểu đồ trình tự Đăng tin*

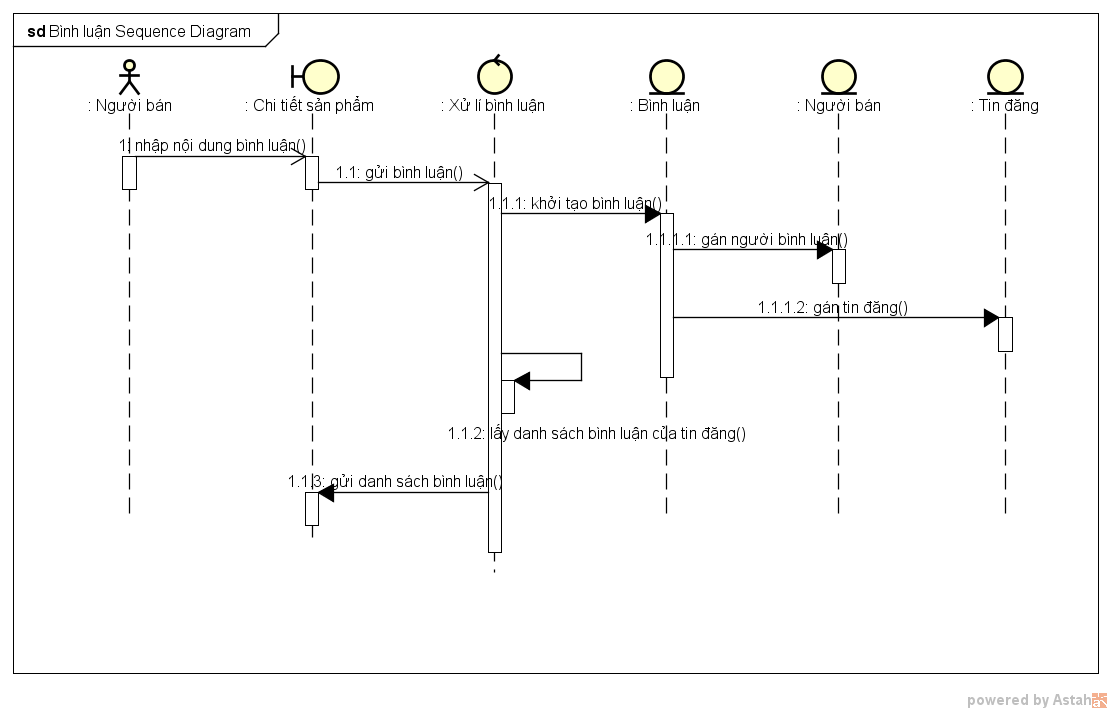
#### *Biểu đồ lớp*



*Biểu đồ 2: Biểu đồ lớp Đăng tin*

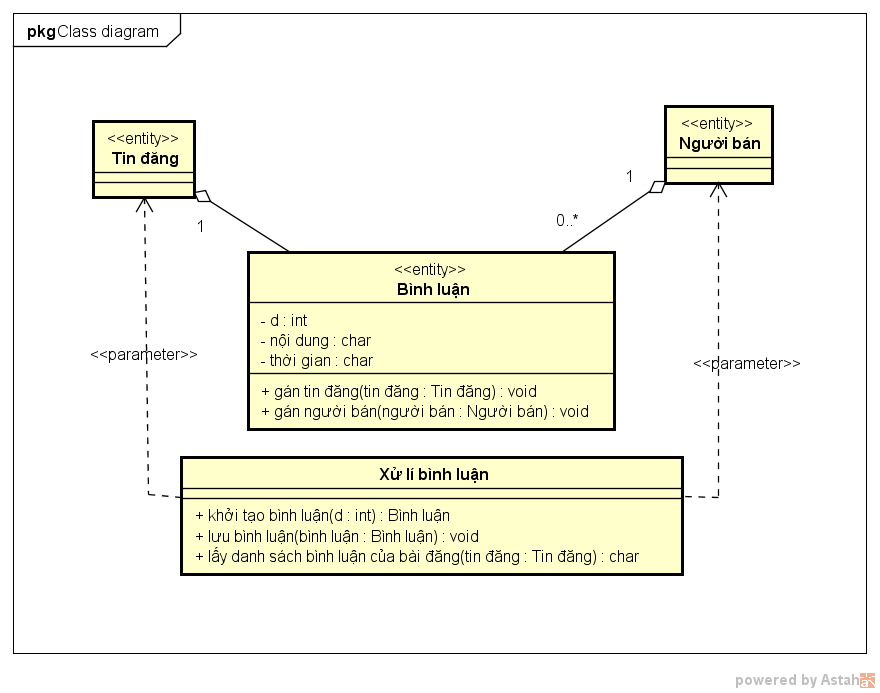
### Bình luận

#### *Biểu đồ trình tự*



*Biểu đồ 3: Biểu đồ trình tự Bình luận*

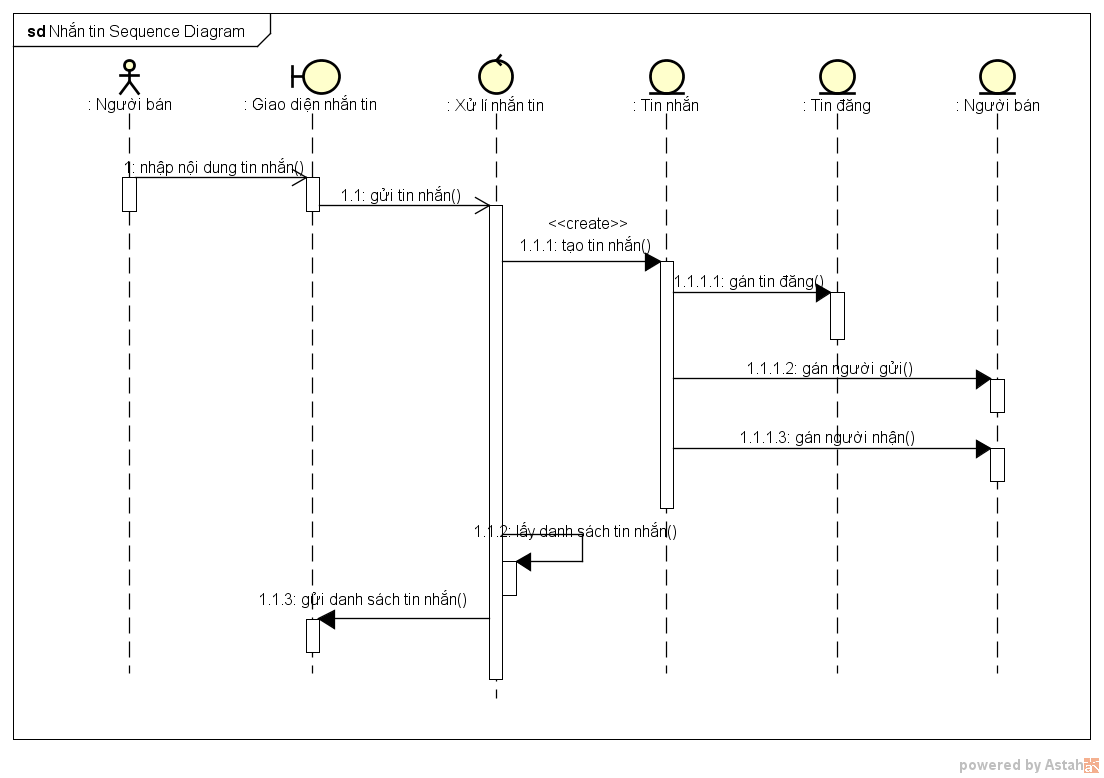
#### *Biểu đồ lớp*



*Biểu đồ 4: Biểu đồ lớp Bình luận*

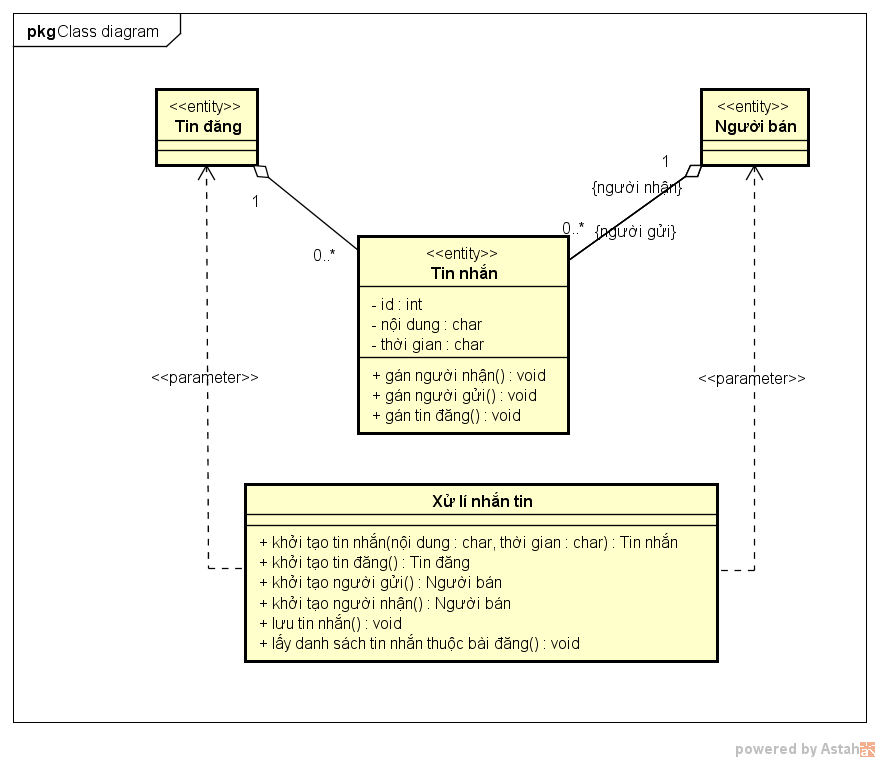
### Nhắn tin

#### *Biểu đồ trình tự*



*Biểu đồ 5: Biểu đồ trình tự Nhắn tin*

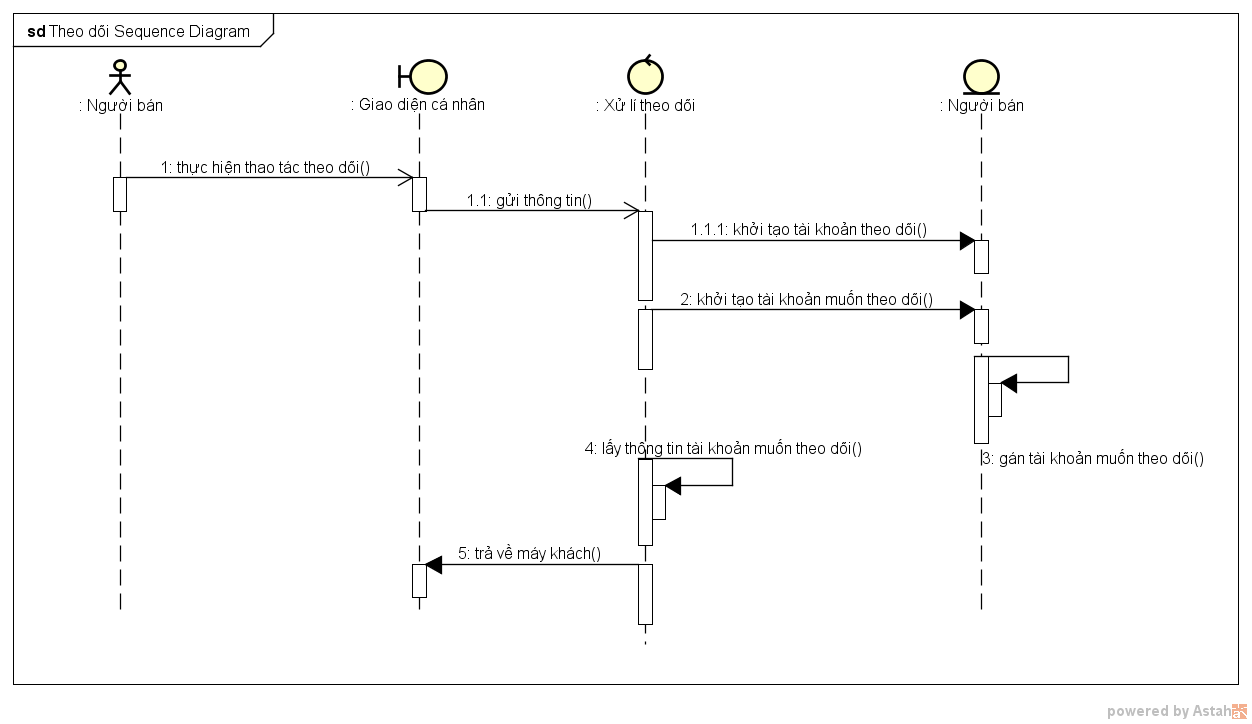
#### *Biểu đồ lớp*



*Biểu đồ 6: Biểu đồ lớp Nhắn tin*

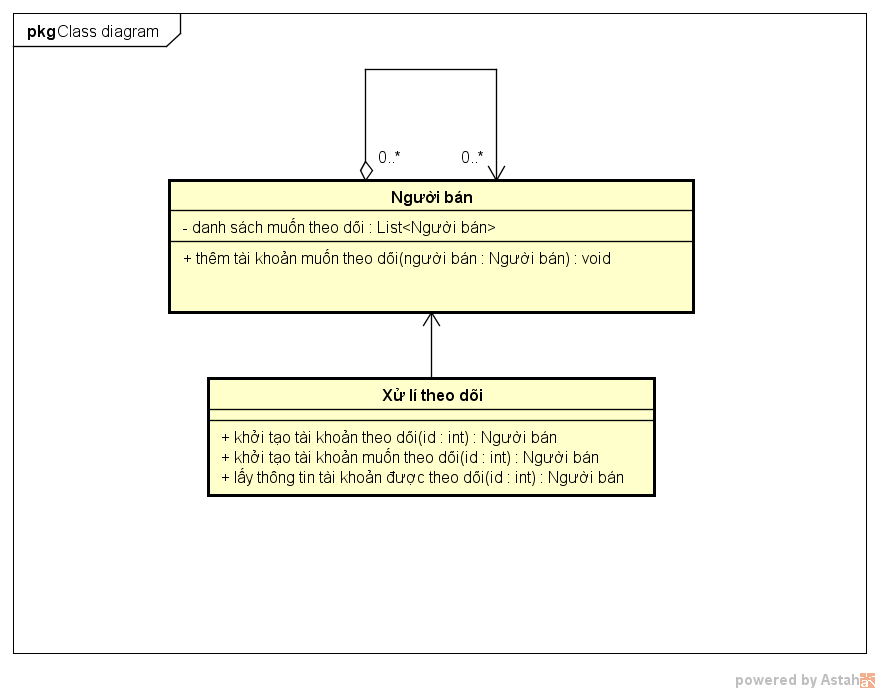
### Theo dõi tài khoản

#### *Biểu đồ trình tự*



*Biểu đồ 7: Biểu đồ trình tự Theo dõi tài khoản*

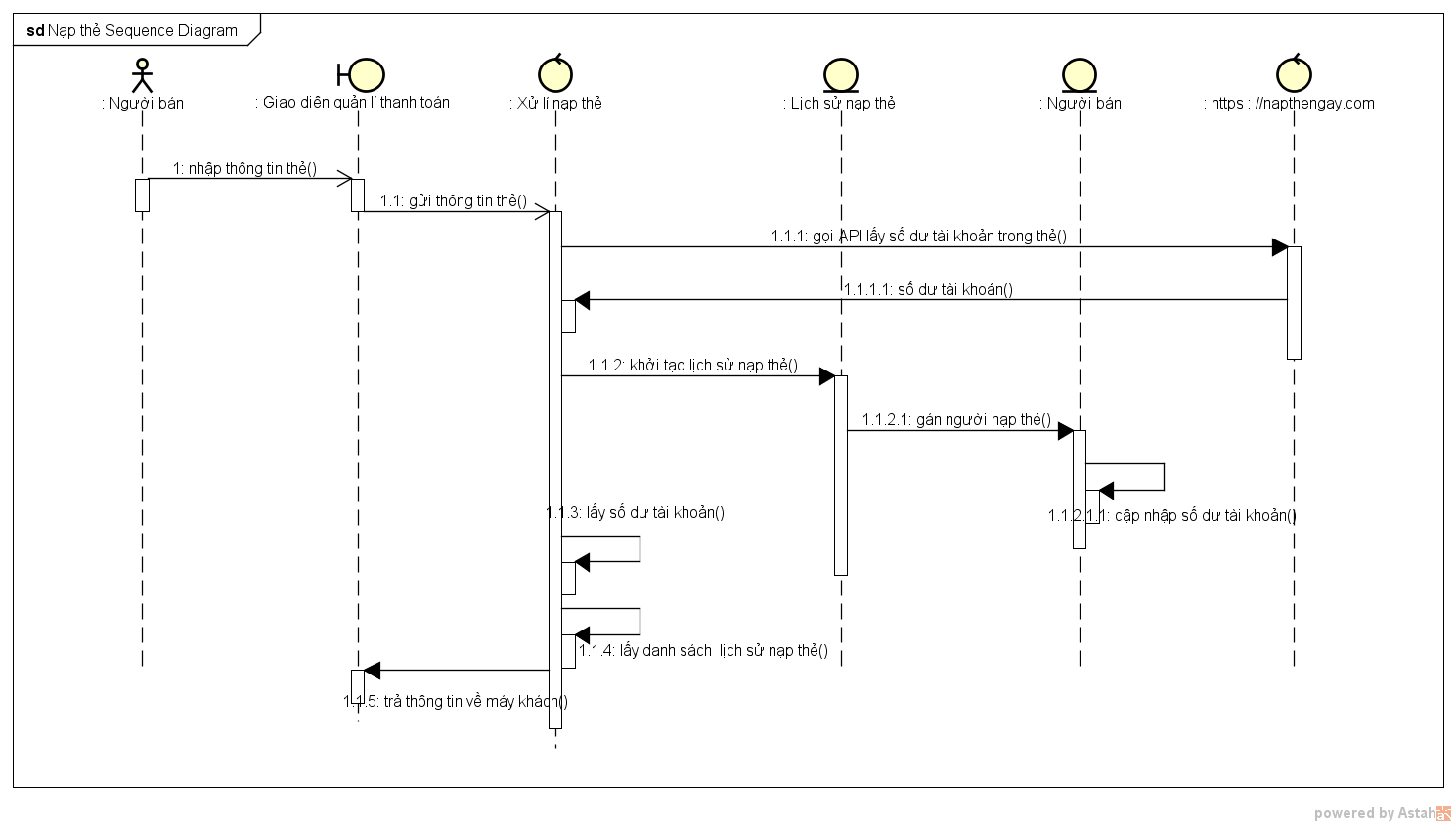
#### *Biểu đồ lớp*



*Biểu đồ 8: Biểu đồ lớp Theo dõi tài khoản*

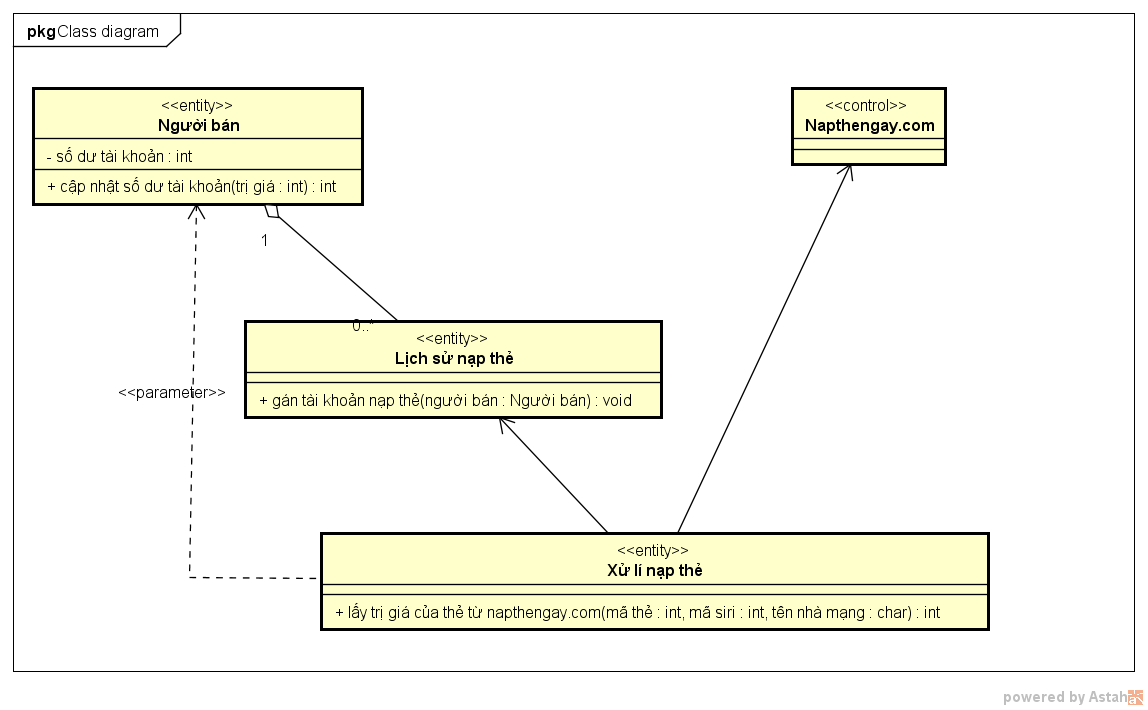
### Nạp thẻ cào

#### *Biểu đồ trình tự*



*Biểu đồ 9: Biểu đồ trình tự Nạp thẻ cào*

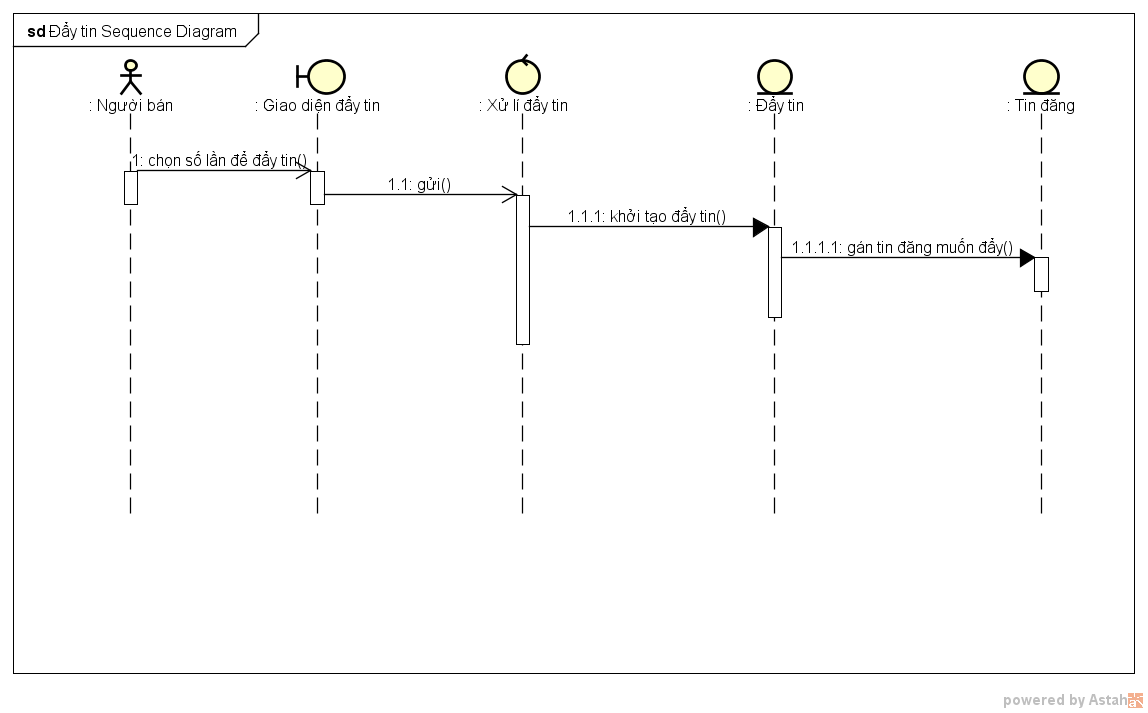
#### *Biểu đồ lớp*



*Biểu đồ 10: Biểu đồ lớp Nạp thẻ cào*

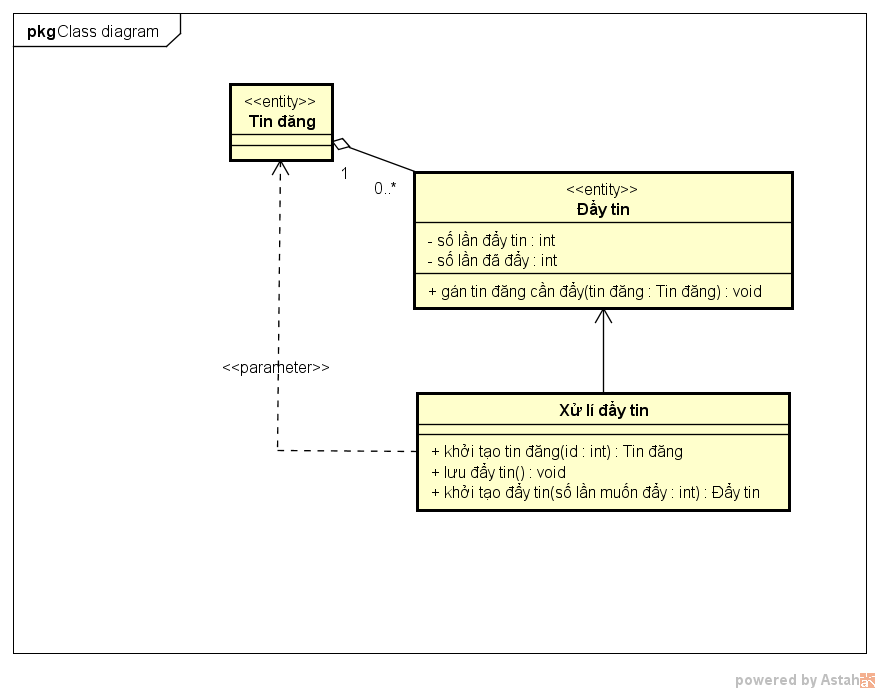
### Đẩy tin đăng

#### *Biểu đồ trình tự*



*Biểu đồ 11:Biểu đồ trình tự Đẩy tin*

#### *Biểu đồ lớp*



*Biểu đồ 12: Biểu đồ lớp Đẩy tin*

### Phản ánh tin đăng

### Theo dõi thống kê tin đăng

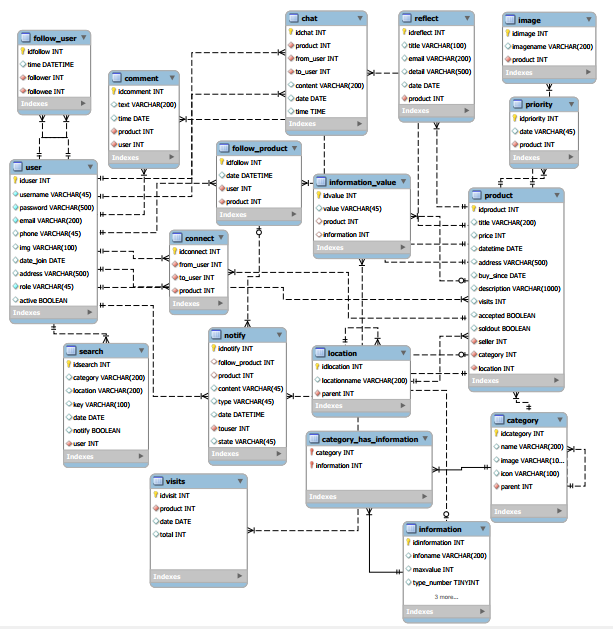
### Kiểm duyệt tin đăng

### Kiểm duyệt thông tin phản ánh

### Thống kê số lượng tin đăng

## Thiết kế cơ sở dữ liệu

### *Sơ đồ quan hệ*



*Hình 20: Quan hệ giữa các bảng cơ sở dữ liệu*



*Hình 21: Tài liệu PDF cơ sở dữ liệu*

### *Mô tả cơ sở dữ liệu*

#### *Bảng cơ sở dữ liệu User*

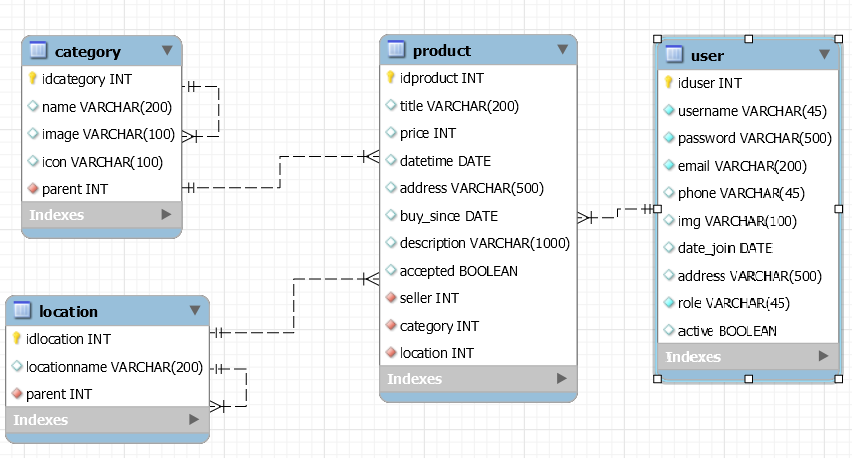
Bảng User chứa thông tin tài khoản của người dùng trong đó có 2 trường là “username” và “password” dùng để đăng nhập vào hệ thống.

|  |  |  |
| --- | --- | --- |
| **User** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| iduser | Int | Khóa chính bảng User |
| username | Varchar(45) | Tên tài khoản dùng để đăng nhập |
| password | Varchar(500) | Mật khẩu tài khoản dùng để đăng nhập |
| email | Varchar(200) | Địa chỉ email dùng để kích hoạt tài khoản đăng kí |
| img | Varchar(200) | Hình ảnh người dùng |
| date\_join | Date | Ngày đăng kí tài khoản |
| address | Varchar(500) | Địa chỉ người dùng |
| role | Varchar(45) | Quyền của tài khoản( hệ thống có hai quyền là “user” và “admin” |
| active | Boolean | Trạng thái của tài khoản(đã kích hoạt hay chưa) |

*Bảng 1: Bảng cơ sở dữ liệu User*

#### *Bảng cơ sở dữ liệu Product (Tin đăng)*

Mỗi tin đăng sẽ có các thông tin bắt buộc như tiêu đề tin đăng (title), giá muốn bán (price), người bán (seller), thể loại (category), địa điểm muốn giao hàng (location)

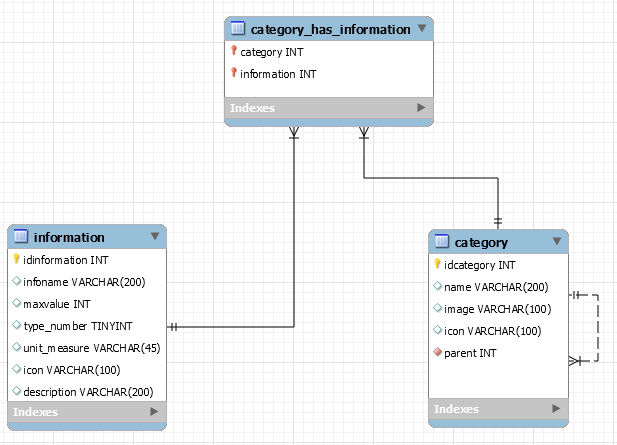


*Hình 22: Quan hệ giữa bảng Product với Category và Location*

|  |  |  |
| --- | --- | --- |
| **Product** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idproduct | Int | Khóa chính |
| title | Varchar(200) | Tiêu đề tin đăng |
| price | Int | Giá muốn bán |
| datetime | Date | Thời gian đăng |
| address | Varchar(500) | Địa chỉ chi tiết muốn giao hàng |
| buy\_since | Date | Thời điểm mua hàng |
| description | Varchar(1000) | Mô tả chi tiết sản phẩm |
| accepted | Boolean | Trạng thái tin đăng, gồm 5 trạng thái   * Trạng thái “chờ phê duyệt” ,giá trị “accepted” bằng 0, khi một tin đăng mới được đăng lên, chờ hệ thống kiểm duyệt * Trạng thái “chấp thuận” ,giá trị “accepted” bằng 1 , khi tin đăng đã được hệ thống chấp thuận * Trạng thái “bị từ chối” ,giá trị “accepted” bằng 2, khi tin đăng bị từ chối * Trạng thái “ẩn” ,giá trị “accepted” bằng 3,khi tin đăng được ẩn * Trạng thái “bị phản ánh” ,giá trị “accepted” bằng 4, khi tin đăng bị phản ánh bởi người dung khác |
| seller | Int | Thông tin người bán ,là khóa ngoài liên kết đến bảng User (liên kết nhiều một) |
| category | Int | Thông tin thể loại sản phẩm muốn bán ,là khóa ngoài liên kết đến bảng Category (liên kết nhiều-một) |
| location | Int | Thông tin địa điểm sản phẩm muốn giao hàng ,là khóa ngoài liên kết đến bảng Location (liên kết nhiều-một) |

*Bảng 2: Bảng cơ sở dữ liệu Product*

#### *Bảng cơ sở dữ liệu Category (Tin đăng)*



*Hình 23: Quan hệ giữa bảng Category và Information*

|  |  |  |
| --- | --- | --- |
| **Category (Thể loại)** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idcategory | Int | Khóa chính bảng Category |
| name | Varchar(200) | Tên thể loại |
| image | Varchar(100) | Ảnh minh họa |
| icon | Varchar(100) | Icon minh họa |
| parent | Int | Mỗi thể loại chứa những thể loại con hơn bên trong, là khóa ngoài liên kết đến chính nó ( liên kết nhiều-một), cho biết thể loại cha của thể loại hiện tại |

*Bảng 3: Bảng cơ cở dữ liệu Category*

#### *Bảng cơ sở dữ liệu Information*

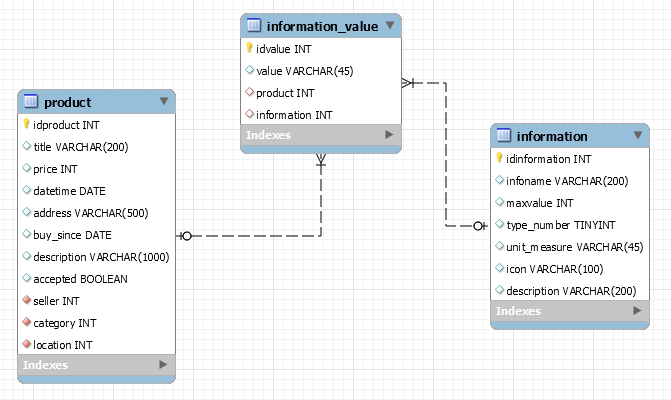
Mỗi thể loại (Category) sẽ có nhiều thuộc tính khác nhau và có thể được cập nhật trong tương lai, không thể biểu diễn hết tất cả thuộc tính trong bảng Category , vì vậy bảng Information sẽ biểu diễn thuộc tính của Category, quan hệ giữa cả hai là nhiều nhiều và được thể hiện qua bảng Category\_has\_Information

|  |  |  |
| --- | --- | --- |
| **Information** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idinformation | Int | Khóa chính |
| infoname | Varchar(200) | Tên thuộc tính |
| max\_value | Int | Giá trị max của thuộc tính nếu có |
| type | Int | Kiểu của thuộc tính, có ba giá trị   * Bằng 1 nếu thuộc tính có kiểu số * Bằng 2 nếu thuộc tính có kiểu xâu * Bằng 3 nếu thuộc tính có kiểu date |
| icon | Varchar(100) | Icon minh họa |
| description | Varchar(100) | Mô tả thuộc tính |

*Bảng 4: Bảng cơ sở dữ liệu Information*

#### *Bảng cơ sở dữ liệu Information\_value*

Biểu diễn giá trị thuộc tính của một tin đăng (Product), một tin đăng sẽ thuộc một thể loại cụ thể nào đó, vì vậy sẽ chứa thuộc tính (Information) của thể loại đó, quan hệ giữa Product và Information là nhiều nhiều được biểu diễn bởi bảng Information\_value.



*Hình 24: Quan hệ giữa Product và Information*

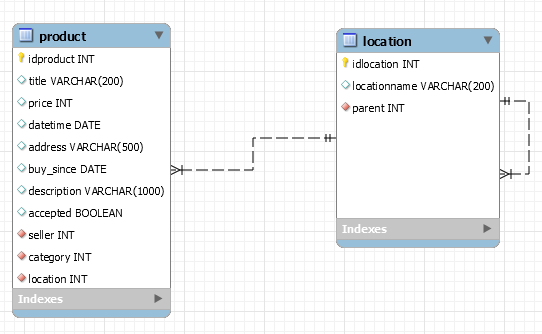
|  |  |  |
| --- | --- | --- |
| **Information\_value** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idvalue | int | Khóa chính |
| value | Varchar(500) | Giá trị thuộc tính |
| product | Integer | Khóa ngoài liên kết tới bảng Product |
| information | Integer | Khóa ngoài liên kết tới bảng Information |

*Bảng 5: Bảng cơ sở dữ liệu Information\_value*

#### *Bảng cơ sở dữ liệu Location*

Bảng Location mô tả địa điểm muốn giao hàng , chứa trường thuộc tính bắt buộc như tên địa điểm “locationname”, tên địa điểm là tên của Tỉnh/Thành Phố hoặc Quận/Huyện, vì các Quận/Huyện trực thuộc Tỉnh/Thành Phố nên bảng Location có thêm trường thuộc tính “parent”.

Vì vậy “parent” là khóa ngoài liên kết tới bảng Location, biểu diễn thông tin một địa điểm trực thuộc thuộc một địa điểm khác



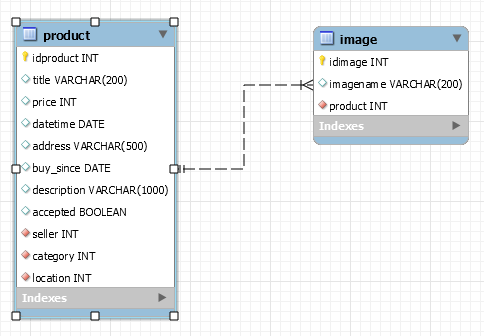
*Hình 25: Quan hệ giữa Product và Location*

|  |  |  |
| --- | --- | --- |
| **Location** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idlocation | Integer | Khóa chính |
| locationname | Varchar(200) | Tên địa điểm |
| parent | Int | Khóa ngoài liên kết tới bảng Location |

*Bảng 6: Bảng cơ sở dữ liệu Location*

#### Bảng cơ sở dữ liệu Image

Một tin đăng ( Product ) sẽ có nhiều ảnh



*Hình 26: Quan hệ giữa Product và Image*

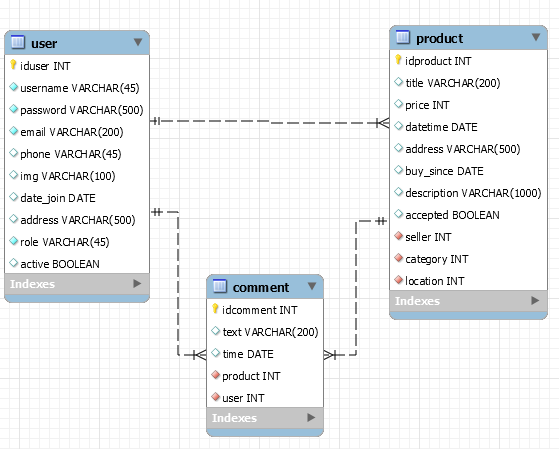
|  |  |  |
| --- | --- | --- |
| **Image** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idimage | Integer | Khóa chính |
| imagename | Varchar(200) | Tên ảnh |
| product | Int | Khóa ngoài liên kết tới bảng Product |

*Bảng 7: Bảng cơ sở dữ liệu Image*

#### Bảng cơ sở dữ liệu Comment

Một người dùng (User) có thể bình luận nhiều lần trên bất kì một tin đăng. Bảng cơ sở dữ liệu Comment lưu thông tin bình luận của một User trên một Product chứa các trường như nội dung bình luận (text) , thời gian (time).

Liên kết giữa User và Comment là một-nhiều, liên kết giữ Product và Comment là một-nhiều



*Hình 27: Quan hệ giữa Comment với Product và User*

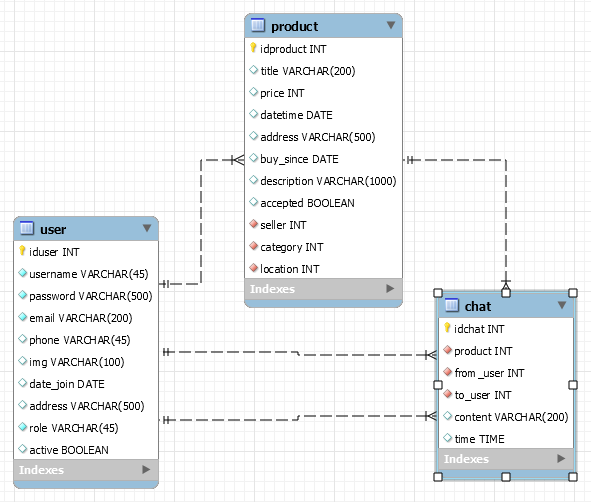
|  |  |  |
| --- | --- | --- |
| **Comment** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idcomment | Integer | Khóa chính |
| text | Varchar(500) | Nội dung bình luận |
| time | DATETIME | Thời gian |
| product | Int | Khóa ngoài liên kết tới bảng Product |
| user | Int | Khóa ngoài liên kết tới bảng User |

*Bảng 8: Bảng cơ sở dữ liệu Comment*

#### Bảng cơ sở dữ liệu Chat

Một người dùng (User) có thể nhắn tin với một người dùng (User) khác liên quan đến một tin đăng (Product)

Liên kết giữa Product và Chat là một-nhiều, một người dùng có thể nhắn tin liên quan đến nhiều tin đăng nên liên kết của User và Chat trên khóa ngoài “from\_user” là một-nhiều, tương tự một người dùng có thể nhận tin nhắn liên quan đến nhiều tin đăng nên liên kết của User và Chat trên khóa ngoài “to\_user” là một-nhiều



*Hình 28: Quan hệ giữa Chat với User và Product*

|  |  |  |
| --- | --- | --- |
| **Chat** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idcomment | Integer | Khóa chính |
| content | Varchar(500) | Nội dung tin nhắn |
| time | DATETIME | Thời gian |
| product | Int | Khóa ngoài liên kết tới bảng Product |
| from\_user | Int | Khóa ngoài liên kết tới bảng User, thông tin người gửi |
| to\_user | Int | Khóa ngoài liên kết tới bảng User, thông tin người nhận |

*Bảng 9: Bảng cơ sở dữ liệu Chat*

## Thiết kế giao diện

# CHƯƠNG 5: XÂY DỰNG VÀ THỬ NGHIỆM



## Môi trường xây dựng và thử nghiệm

## Các kết quả thử nghiệm

# CHƯƠNG 6: KẾT LUẬN



## Đã làm

## Mở rộng

# TÀI LIỆU THAM KHẢO