**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 8](#_Toc25931555)

[1.1 Giới thiệu bài toán 8](#_Toc25931556)

[1.2 Phạm vi hệ thống 9](#_Toc25931557)

[1.3 Cách tiếp cận 9](#_Toc25931558)

[1.4 Giới thiệu cấu trúc đồ án 9](#_Toc25931559)

[CHƯƠNG 2: CÁC NỀN TẢNG CÔNG NGHỆ 9](#_Toc25931560)

[2.1 Giới thiệu về Spring Boot 9](#_Toc25931564)

[2.2 Giới thiệu về Angular 7 11](#_Toc25931565)

[CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG 13](#_Toc25931569)

[3.1 Các đối tác 13](#_Toc25931571)

[3.2 Các ca sử dụng 14](#_Toc25931572)

[3.3 Đặc tả ca sử dụng 15](#_Toc25931576)

[3.3.1 Đăng nhập 15](#_Toc25931577)

[3.3.2 Đăng kí tài khoản 17](#_Toc25931578)

[3.3.3 Đổi lại mật khẩu 20](#_Toc25931579)

[3.3.4 Tìm kiếm 23](#_Toc25931580)

[3.3.5 Đăng tin 25](#_Toc25931581)

[3.3.6 Bình luận 28](#_Toc25931582)

[3.3.7 Nhắn tin 30](#_Toc25931583)

[3.3.8 Theo dõi tài khoản 33](#_Toc25931584)

[3.3.9 Nạp thẻ cào 35](#_Toc25931585)

[3.3.10 Đẩy tin đăng 38](#_Toc25931586)

[3.3.11 Ẩn tin 41](#_Toc25931587)

[3.3.12 Phản ánh tin đăng 42](#_Toc25931588)

[3.3.13 Theo dõi thống kê 45](#_Toc25931589)

[3.3.14 Kiểm duyệt tin đăng 47](#_Toc25931590)

[3.3.15 Kiểm duyệt phản ánh 50](#_Toc25931591)

[3.3.16 Thống kê số lượng tin đăng 52](#_Toc25931592)

[CHƯƠNG 4: THIẾT KẾ HỆ THỐNG 56](#_Toc25931593)

[4.1 Kiến trúc hệ thống 56](#_Toc25931595)

[4.1.1 Kiến trúc triển khai 56](#_Toc25931596)

[4.1.2 Kiến trúc logic 57](#_Toc25931597)

[4.1.3 Kiến trúc mã nguồn 58](#_Toc25931598)

[4.1.4 Các kĩ thuật 59](#_Toc25931599)

[4.2 Thiết kế ca sử dụng 63](#_Toc25931600)

[4.2.1 Đăng tin 63](#_Toc25931601)

[4.2.2 Tìm kiếm 72](#_Toc25931602)

[4.2.3 Bình luận 76](#_Toc25931603)

[4.2.4 Nhắn tin 80](#_Toc25931604)

[4.2.5 Theo dõi tài khoản 85](#_Toc25931605)

[4.2.6 Nạp thẻ cào 88](#_Toc25931606)

[4.2.7 Đẩy tin đăng 93](#_Toc25931607)

[4.2.8 Phản ánh tin đăng 96](#_Toc25931608)

[4.2.9 Theo dõi thống kê tin đăng 96](#_Toc25931609)

[4.2.10 Kiểm duyệt tin đăng 96](#_Toc25931610)

[4.2.11 Kiểm duyệt thông tin phản ánh 96](#_Toc25931611)

[4.2.12 Thống kê số lượng tin đăng 96](#_Toc25931612)

[4.3 Thiết kế cơ sở dữ liệu 97](#_Toc25931613)

[4.3.1 Sơ đồ quan hệ 98](#_Toc25931614)

[4.3.2 Mô tả cơ sở dữ liệu 98](#_Toc25931615)

[4.4 Thiết kế giao diện 113](#_Toc25931616)

[4.4.1 Giao diện Đăng nhập 114](#_Toc25931617)

[4.4.2 Giao diện Đăng kí 116](#_Toc25931618)

[4.4.3 Giao diện Trang chủ 117](#_Toc25931619)

[4.4.4 Giao diện Tìm kiếm 119](#_Toc25931620)

[4.4.5 Giao diện Chi tiết tin đăng 123](#_Toc25931621)

[4.4.6 Giao diện Trang cá nhân 126](#_Toc25931622)

[4.4.7 Giao diện Đăng tin 128](#_Toc25931623)

[4.4.8 Giao diện Nhắn tin 132](#_Toc25931624)

[4.4.9 Giao diện Quản lí tin đăng 133](#_Toc25931625)

[4.4.10 Giao diện Đẩy tin 136](#_Toc25931626)

[4.4.11 Giao diện Quản trị tin đăng 138](#_Toc25931627)

[4.4.12 Giao diện Quản trị người dùng 139](#_Toc25931628)

[4.4.13 Giao diện Phê duyệt tin đăng 140](#_Toc25931629)

[4.4.14 Giao diện Phê duyệt phản ánh 142](#_Toc25931630)

[CHƯƠNG 5: XÂY DỰNG VÀ THỬ NGHIỆM 144](#_Toc25931631)

[5.1 Môi trường xây dựng và thử nghiệm 144](#_Toc25931633)

[5.1.1 Cài đặt môi trường Java 144](#_Toc25931634)

[5.1.2 Cài đặt môi trường Angular 145](#_Toc25931635)

[5.1.3 Cài đặt môi trường MySQL 145](#_Toc25931636)

[5.2 Các kết quả thử nghiệm 146](#_Toc25931637)

[CHƯƠNG 6: KẾT LUẬN 147](#_Toc25931638)

[6.1 Đã làm 147](#_Toc25931640)

[6.2 Mở rộng 147](#_Toc25931641)

[TÀI LIỆU THAM KHẢO 148](#_Toc25931642)

[*Hình 1: Kiến trúc Spring* 9](#_Toc25616996)

[*Hình 2: Tương tác giữa các thành phần trong Angular 7* 11](#_Toc25616997)

[*Hình 3: Minh họa giao diện đăng nhập* 16](#_Toc25616998)

[*Hình 4: Minh họa giao diện đăng kí tài khoản* 19](#_Toc25616999)

[*Hình 5: Minh họa giao diện đổi lại mật khẩu* 22](#_Toc25617000)

[*Hình 6: Minh họa giao diện tìm kiếm* 24](#_Toc25617001)

[*Hình 7: Minh họa giao diện đăng tin* 26](#_Toc25617002)

[*Hình 8: Minh họa giao diện bình luận* 29](#_Toc25617003)

[*Hình 9: Minh họa giao diện chát* 32](#_Toc25617004)

[*Hình 10: Minh họa trang cá nhân người dùng* 34](#_Toc25617005)

[*Hình 11: Minh họa giao diện nạp thẻ cào* 37](#_Toc25617006)

[*Hình 12: Minh họa giao diện đẩy tin đăng* 39](#_Toc25617007)

[*Hình 13: Minh họa giao diện phản ánh tin đăng* 44](#_Toc25617008)

[*Hình 14: Minh họa giao diện thống kê lượt truy cập của tin đăng* 46](#_Toc25617009)

[*Hình 15: Minh họa giao diện kiểm duyệt tin đăng* 49](#_Toc25617010)

[*Hình 16: Minh họa giao diện thống kê số lượng tin đăng* 54](#_Toc25617011)

[*Hình 17: Kiến trúc triển khai của hệ thống* 55](#_Toc25617012)

[*Hình 18: Kiến trúc logic của ứng dụng* 57](#_Toc25617013)

[*Hình 19: Luồng xử lí Auth0* 59](#_Toc25617014)

[*Hình 20: Quan hệ giữa các bảng cơ sở dữ liệu* 74](#_Toc25617015)

[*Hình 21: Tài liệu PDF cơ sở dữ liệu* 75](#_Toc25617016)

[*Hình 22: Quan hệ giữa bảng Product với Category và Location* 76](#_Toc25617017)

[*Hình 23: Quan hệ giữa bảng Category và Information* 78](#_Toc25617018)

[*Hình 24: Quan hệ giữa Product và Information* 80](#_Toc25617019)

[*Hình 25: Quan hệ giữa Product và Location* 81](#_Toc25617020)

[*Hình 26: Quan hệ giữa Product và Image* 82](#_Toc25617021)

[*Hình 27: Quan hệ giữa Comment với Product và User* 84](#_Toc25617022)

[*Hình 28: Quan hệ giữa Chat với User và Product* 85](#_Toc25617023)

[*Hình 29: Quan hệ giữa User và Follow\_user* 86](#_Toc25617024)

[*Hình 30: Quan hệ giữa Follow\_Product với Product và User* 88](#_Toc25617025)

[*Hình 31: Giao diện Đăng nhập* 90](#_Toc25617026)

[*Hình 32: Giao diện Đăng kí* 92](#_Toc25617027)

[*Hình 33: Giao diện Trang chủ* 93](#_Toc25617028)

[*Hình 34: Giao diện Tìm kiếm* 95](#_Toc25617029)

[*Hình 35: Giao diện modal Chọn địa điểm* 96](#_Toc25617030)

[*Hình 36: Giao diện modal Chọn danh mục* 96](#_Toc25617031)

[*Hình 37: Giao diện Chi tiết tin đăng* 99](#_Toc25617032)

[*Hình 38: Giao diện modal Gửi phản hồi* 99](#_Toc25617033)

[*Hình 39: Giao diện Trang cá nhân* 102](#_Toc25617034)

[*Hình 40: Giao diện Chọn địa điểm để đăng tin* 104](#_Toc25617035)

[*Hình 41: Giao diện Chọn thể loại để đăng tin* 104](#_Toc25617036)

[*Hình 42: Giao diện Thông tin chung để đăng tin* 105](#_Toc25617037)

[*Hình 43: Giao diện Thông tin chi tiết lúc đăng tin* 105](#_Toc25617038)

[*Hình 44: Giao diện Nhắn tin* 108](#_Toc25617039)

[*Hình 45: Giao diện Quản lí tin đăng* 109](#_Toc25617040)

[*Hình 46: Giao diện Thống kê tin đăng* 110](#_Toc25617041)

[*Hình 47: Giao diện Đẩy tin* 112](#_Toc25617042)

[*Hình 48: Giao diện Quản trị tin đăng* 114](#_Toc25617043)

[*Hình 49: Giao diện Quản trị người dùng* 115](#_Toc25617044)

[*Hình 50: Giao diện Phê duyệt tin đăng* 116](#_Toc25617045)

[*Hình 51: Giao diện Soạn Email* 116](#_Toc25617046)

[*Hình 52: Giao diện Phê duyệt phản ánh* 118](#_Toc25617047)

[*Hình 53: Giao diện Danh sách phản ánh của tin đăng* 118](#_Toc25617048)

[*Biểu đồ 1: Biểu đồ trình tự Đăng tin* 65](#_Toc25931699)

[*Biểu đồ 2: Biểu đồ lớp Đăng tin* 66](#_Toc25931700)

[*Biểu đồ 3: Biểu đồ trình tự Tìm kiếm* 73](#_Toc25931701)

[*Biểu đồ 4: Biểu đố lớp Tìm kiếm* 74](#_Toc25931702)

[*Biểu đồ 5: Biểu đồ trình tự Bình luận* 77](#_Toc25931703)

[*Biểu đồ 6: Biểu đồ lớp Bình luận* 78](#_Toc25931704)

[*Biểu đồ 7: Biểu đồ trình tự Nhắn tin* 81](#_Toc25931705)

[*Biểu đồ 8: Biểu đồ lớp Nhắn tin* 82](#_Toc25931706)

[*Biểu đồ 9: Biểu đồ trình tự Theo dõi tài khoản* 86](#_Toc25931707)

[*Biểu đồ 10: Biểu đồ lớp Theo dõi tài khoản* 87](#_Toc25931708)

[*Biểu đồ 11: Biểu đồ trình tự Nạp thẻ cào* 90](#_Toc25931709)

[*Biểu đồ 12: Biểu đồ lớp Nạp thẻ cào* 91](#_Toc25931710)

[*Biểu đồ 13:Biểu đồ trình tự Đẩy tin* 94](#_Toc25931711)

[*Biểu đồ 14: Biểu đồ lớp Đẩy tin* 95](#_Toc25931712)

[*Bảng 1: Bảng mô tả lớp User-ca sử dụng Đăng tin* 67](#_Toc25931713)

[*Bảng 2: Bảng mô tả lớp Category-ca sử dụng Đăng tin* 67](#_Toc25931714)

[*Bảng 3: Bảng mô tả lớp Location-ca sử dụng Đăng tin* 68](#_Toc25931715)

[*Bảng 4: Bảng mô tả lớp Product-ca sử dụng Đăng tin* 69](#_Toc25931716)

[*Bảng 5: Bảng mô tả lớp CategoryService-ca sử dụng Đăng tin* 70](#_Toc25931717)

[*Bảng 6: Bảng mô tả lớp LocationService-ca sử dụng Đăng tin* 70](#_Toc25931718)

[*Bảng 7: Bảng mô tả lớp UserService-ca sử dụng Đăng tin* 71](#_Toc25931719)

[*Bảng 8: Bảng mô tả lớp ProductService-ca sử dụng Đăng tin* 71](#_Toc25931720)

[*Bảng 9: Bảng mô tả lớp ProductController-ca sử dụng Đăng tin* 72](#_Toc25931721)

[*Bảng 10: Bảng mô tả lớp CategoryService-ca sử dụng Tìm kiếm* 75](#_Toc25931722)

[*Bảng 11: Bảng mô tả lớp LocationService-ca sử dụng Tìm kiếm* 75](#_Toc25931723)

[*Bảng 12: Bảng mô tả lớp ProductService-ca sử dụng Tìm kiếm* 76](#_Toc25931724)

[*Bảng 13: Bảng mô tả lớp ProductController-ca sử dụng Tìm kiếm* 76](#_Toc25931725)

[*Bảng 14: Bảng mô tả lớp Comment-ca sử dụng Bình luận* 79](#_Toc25931726)

[*Bảng 15: Bảng mô tả lớp CommentRepository-ca sử dụng Bình luận* 79](#_Toc25931727)

[*Bảng 16: Bảng mô tả lớp ProductService-ca sử dụng Bình luận* 80](#_Toc25931728)

[*Bảng 17: Bảng mô tả lớp CommenttController-ca sử dụng Bình luận* 80](#_Toc25931729)

[*Bảng 18: Bảng mô tả đối tượng Chat-ca sử dụng Nhắn tin* 83](#_Toc25931730)

[*Bảng 19: Bảng mô tả đối tượng Dialog-ca sử dụng Nhắn tin* 84](#_Toc25931731)

[*Bảng 20: Bảng mô tả lớp ChatRepository-ca sử dụng Nhắn tin* 84](#_Toc25931732)

[*Bảng 21: Bảng mô tả lớp DialogRepository-ca sử dụng Nhắn tin* 84](#_Toc25931733)

[*Bảng 22: Bảng mô tả lớp ChatController-ca sử dụng Nhắn tin* 85](#_Toc25931734)

[*Bảng 23: Bảng mô tả lớp FollowUser-ca sử dụng Theo dõi tài khoản* 88](#_Toc25931735)

[*Bảng 24: Bảng mô tả lớp FollowUserRepository-ca sử dụng Theo dõi tài khoản* 89](#_Toc25931736)

[*Bảng 25: Bảng mô tả lớp UserController-ca sử dụng Theo dõi tài khoản* 89](#_Toc25931737)

[*Bảng 26: Bảng mô tả lớp User-ca sử dụng Nạp thẻ cào* 91](#_Toc25931738)

[*Bảng 27: Bảng mô tả lớp UserRepository- ca sử dụng Nạp thẻ cào* 92](#_Toc25931739)

[*Bảng 28: Bảng mô tả lớp UserService- ca sử dụng Nạp thẻ cào* 92](#_Toc25931740)

[*Bảng 29: Bảng mô tả lớp UserController-ca sử dụng Nạp thẻ cào* 93](#_Toc25931741)

[*Bảng 30: Bảng mô tả lớp PushProduct-ca sử dụng Đẩy tin* 96](#_Toc25931742)

[*Bảng 31: Bảng mô tả lớp PushProductRepository-ca sử dụng Đẩy tin* 96](#_Toc25931743)

[*Bảng 32: Bảng mô tả lớp ProductController- ca sử dụng Đẩy tin* 96](#_Toc25931744)

[*Bảng 33: Bảng cơ sở dữ liệu User* 100](#_Toc25931745)

[*Bảng 34: Bảng cơ sở dữ liệu Product* 102](#_Toc25931746)

[*Bảng 35: Bảng cơ cở dữ liệu Category* 104](#_Toc25931747)

[*Bảng 36: Bảng cơ sở dữ liệu Information* 104](#_Toc25931748)

[*Bảng 37: Bảng cơ sở dữ liệu Information\_value* 105](#_Toc25931749)

[*Bảng 38: Bảng cơ sở dữ liệu Location* 107](#_Toc25931750)

[*Bảng 39: Bảng cơ sở dữ liệu Image* 108](#_Toc25931751)

[*Bảng 40: Bảng cơ sở dữ liệu Comment* 109](#_Toc25931752)

[*Bảng 41: Bảng cơ sở dữ liệu Chat* 111](#_Toc25931753)

[*Bảng 42: Bảng cơ sở dữ liệu Follow\_User* 112](#_Toc25931754)

[*Bảng 43: Bảng cơ sở dữ liệu Follow\_Product* 114](#_Toc25931755)

[*Bảng 44: Bảng mô tả giao diện Đăng nhập* 116](#_Toc25931756)

[*Bảng 45: Bảng mô tả giao diện Đăng kí* 118](#_Toc25931757)

[*Bảng 46: Bảng mô tả giao diện Trang chủ* 119](#_Toc25931758)

[*Bảng 47: Bảng mô tả giao diện Tìm kiếm* 123](#_Toc25931759)

[*Bảng 48: Bảng mô tả giao diện Chi tiết tin đăng* 126](#_Toc25931760)

[*Bảng 49: Bảng mô tả giao diện Trang cá nhân* 128](#_Toc25931761)

[*Bảng 50: Bảng mô tả giao diện Đăng tin* 133](#_Toc25931762)

[*Bảng 51: Bảng mô tả giao diện Nhắn tin* 134](#_Toc25931763)

[*Bảng 52: Bảng mô tả giao diện Quản lí tin đăng* 137](#_Toc25931764)

[*Bảng 53: Bảng mô tả giao diện Đẩy tin* 138](#_Toc25931765)

[*Bảng 54: Bảng mô tả giao diện Phê duyệt tin đăng* 143](#_Toc25931766)

[*Bảng 55: Bảng mô tả giao diện Phê duyệt phản ánh* 145](#_Toc25931767)

[*Bảng 56: Bảng mô tả các chức năng đã hoàn thành* 148](#_Toc25931768)

# 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

Spring là một framework java mãnh mẽ và phổ biến nhất hiện nay dành cho doanh nghiệp. Nó giúp rút ngắn thời gian lập trình và test, giảm sự rườm rà trong code nâng cao hiệu năng của lập trình viên.. Hệ sinh thái Spring hỗ trợ mọi layer từ frontend, backend, persistence, third-paty, hỗ trợ mọi kiến trúc, ngoài ra cú pháp của Java rõ ràng, dễ hiểu và dễ đọc, khái niệm OOP trong Java là chuẩn mực, kế thừa và interface khiến cho hệ thống dễ dàng mở rộng hơn. Chính vì spring giúp tăng cường quá trình lập trình và hỗ trợ các lập trình viên phát triển ứng dụng một cách nhanh chóng, cộng đồng sử dụng đông đảo cùng tài liệu học tập chất lượng, nên em chọn framwork Java Spring để phát triển backend

Angular là một Frameworks Javascript giúp chúng ta xây dựng ứng dụng Web đầy đủ tính năng từ phía Client với ưu điểm là hiệu năng cao ,cấu trúc rõ ràng, dễ dàng phát triển và mở rộng. Angular cũng có một cộng đồng sử dụng rất lớn, đồng thời được phát triển bởi gã khổng lồ Google, khiến Angular không ngừng trưởng thành với các phiên bản luôn được cập nhật. Với việc Angular đang sở hữu một cộng đồng sử dụng lớn và có rất nhiều tài liệu hỗ trợ tìm hiểu Angular, nên việc nắm bắt Angular khá dễ dàng, vì vậy em chọn framework Angular để phát triển ứng dụng phía client.

## Giới thiệu cấu trúc đồ án

Trong phần này em sẽ trình bày ngắn gọn về nội dung chính của từng phần trong luận văn đồ án nhằm giúp cho người đọc dễ dàng theo dõi.

Chương 1: Giới thiệu tổng quan về đề tài. Trong chương này còn trình bày về các vấn đề cụ thể cần phải giải quyết và hướng tiếp cận

Chương 2: Trình bày về cơ sở lý thuyết được sử dụng trong đề tài bao gồm các nội dung về ngôn ngữ lập trình Spring Boot framework, Angular, Bootstrap, MySQL.

- Trong phần Spring Boot framework, em sẽ giới thiệu sơ lược về framework

này và một số khái niệm liên quan

- Phần Agular sẽ trình bày tổng quan về framework này và những đặc trưng

nổi bật của nó.

- Phần tiếp theo trình bày về MySQL, và ưu điểm của hệ quản trị CSDL này.

- Phần cuối cùng sẽ giới thiệu về Bootstrap và tính năng responsive mà Bootstrap hỗ trợ trong việc xây dựng giao diện người dùng cho ứng dụng web.

Chương 3: Chương này trình bày phần phân tích hệ thống, đặc tả một số ca sử dụng quan trọng trong hệ thống

Chương 4:Trình bày về thiết kế hệ thống

Chương 5: Hướng dẫn triển khai ứng dụng

Chương 6: Những kết quả đạt được, thảo luận những vấn đề kho khăn và đưa ra

hướng phát triển trong tương lai.

Phần cuối cùng là tài liệu tham khảo

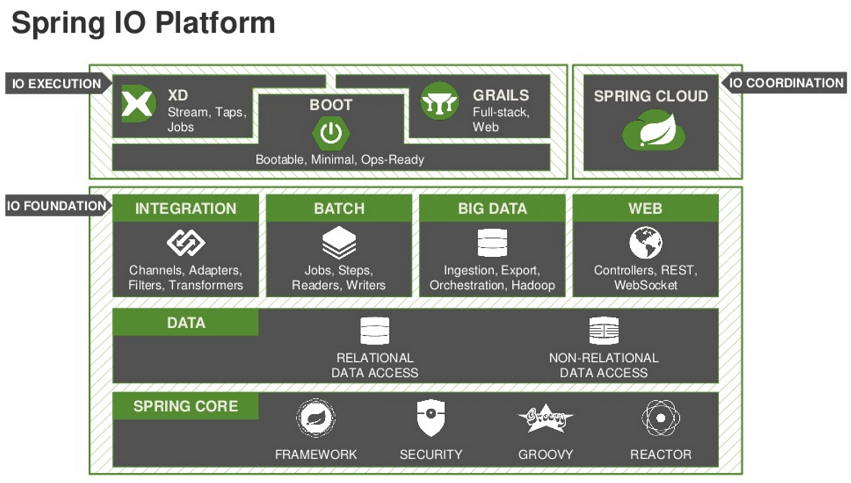
# CHƯƠNG 2: CÁC NỀN TẢNG CÔNG NGHỆ







## 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

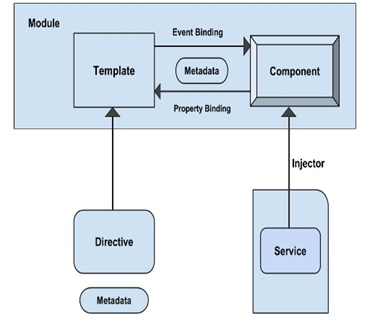
Dưới đây là các lợi ích của Spring Boot

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.

## Bootstrap

Bootstrap là một nền tảng (framework) miễn phí, mã nguồn mở, dựa trên HTML, CSS & Javascript, nó được tạo ra để xây dựng các giao diện Website tương thích với tất cả các thiết bị có kích thước màn hình khác nhau.

Bootstrap bao gồm những thứ cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Nó cũng có nhiều Component, Javascript hỗ trợ cho việc thiết kế Reponsive của Website trở nên dễ dàng, thuận tiện và nhanh chóng hơn. Boostrap định nghĩa sẵn các class CSS giúp người thiết kế giao diện website tiết kiệm rất nhiều thời gian.

Với Bootstrap, việc phát triển giao diện website để phù hợp với đa thiết bị trở nên dễ dàng hơn bao giờ hết. Đây là xu hướng phát triển giao diện website đang rất được ưu chuộng hiện nay. Bootstrap cung cấp tính năng responsive và mobile first, nghĩa là làm cho trang web có thể tự co giãn để tương thích với mọi thiết bị khác nhau, từ điện thoại di động đến máy tính bảng, máy tính xách tay, máy tính để bàn,... Một khía cạnh khác là responsive web design làm cho trang web cung cấp đượctrải nghiệm tuyệt vời cho người dùng trên nhiều thiết bị, kích thước màn hình khác nhau. Đây là một đặc điểm vô cùng quan trọng vì chúng ta không thể lường trước được các thiết bị, kích thước màn hình sẽ sử dụng để truy cập vào trang web. Một trang có thể hoạt động tốt bất kì thiết bị nào sẽ cung cấp một trải nghiệm người dùng tốt hơn là một trang được thiết kế cho một loại thiết bị và kích thước màn hình cụ thể.

## MySQL

MySQL là một Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL). Là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet.

# CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG

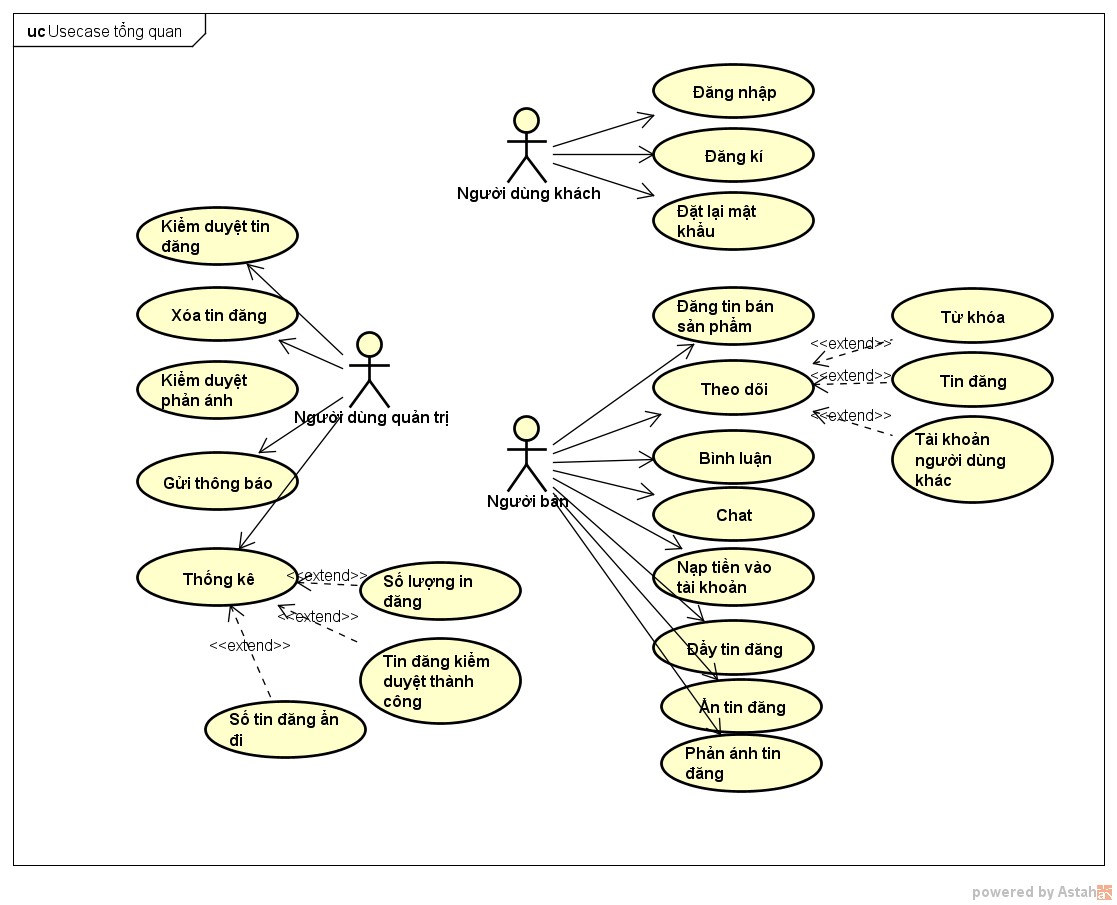


## Các đối tác

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 |

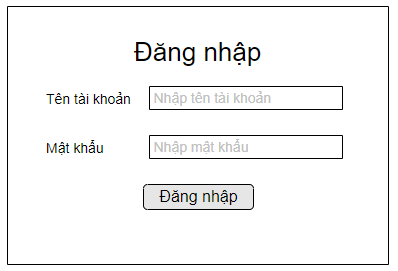
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

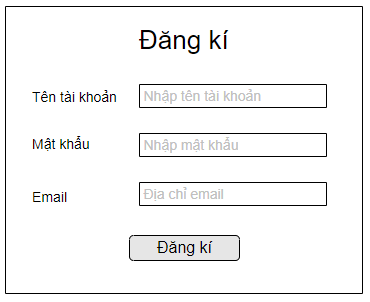
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

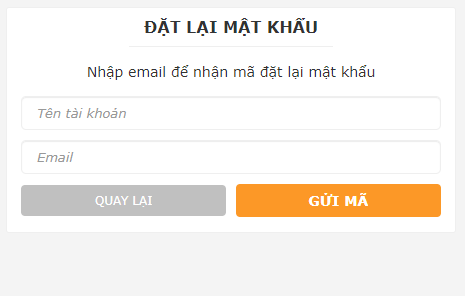
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

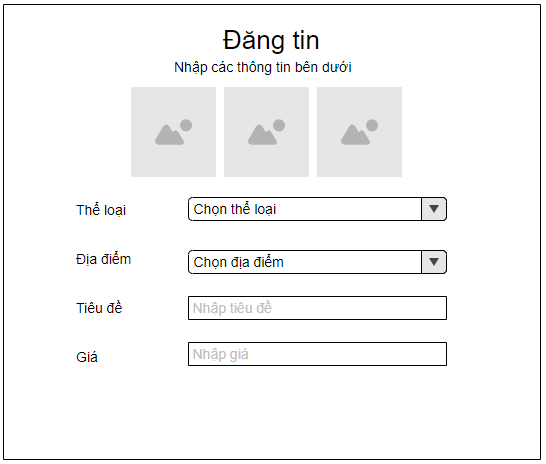
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

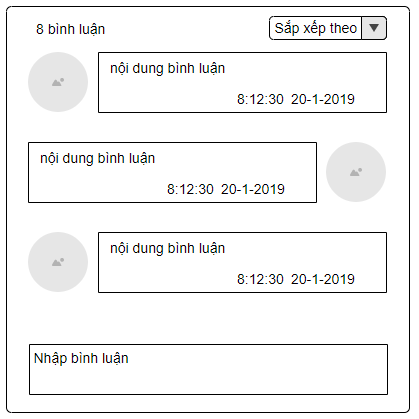
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

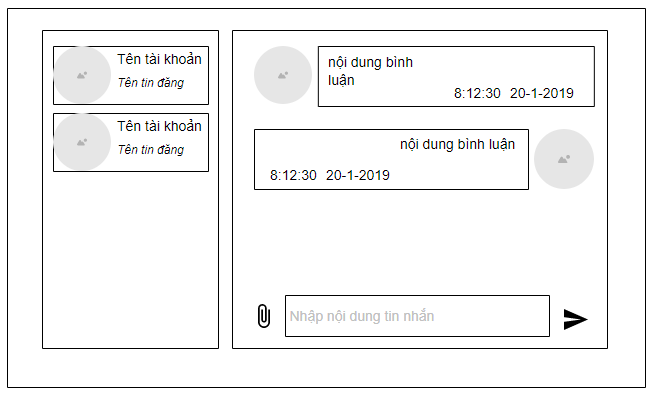
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

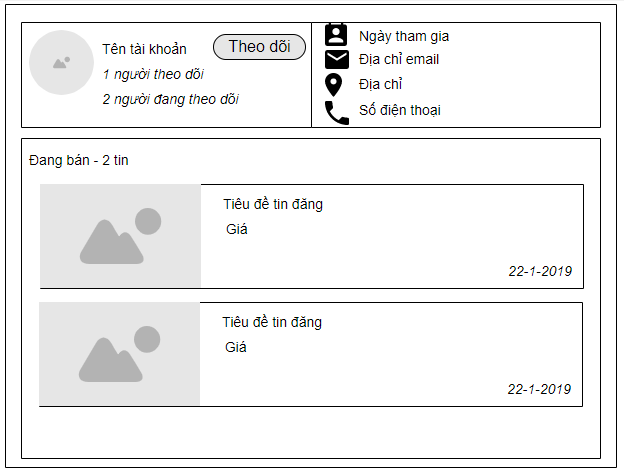
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

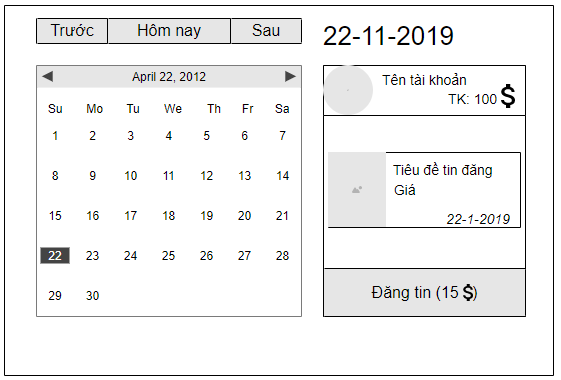
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Đẩy tin đăng | | | |
| 1 | Click button “Đẩy tin” |  |  |
|  |  | 2 | Hiển thị giao diện 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 |

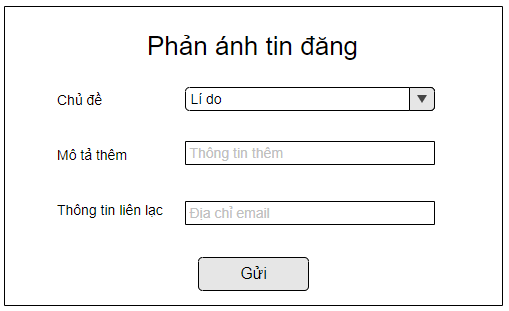
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **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) |

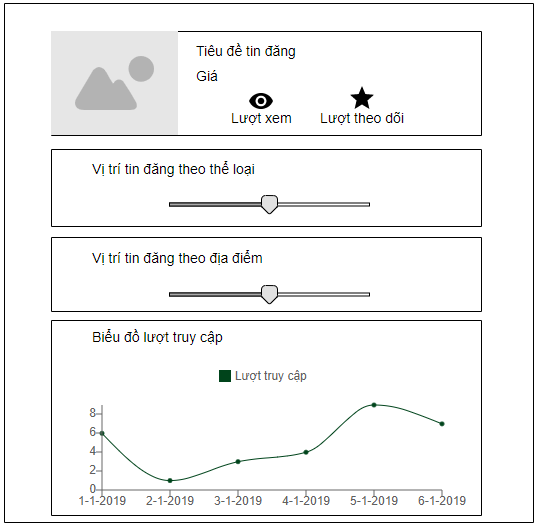
#### Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **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,tin đăng bị phản ánh, tin đăng ẩn 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 ,bị từ chối, tin đăng bị phản ánh và tin đăng ẩn 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 ,bị từ chối, tin đăng bị phản ánh và tin đăng ẩn 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



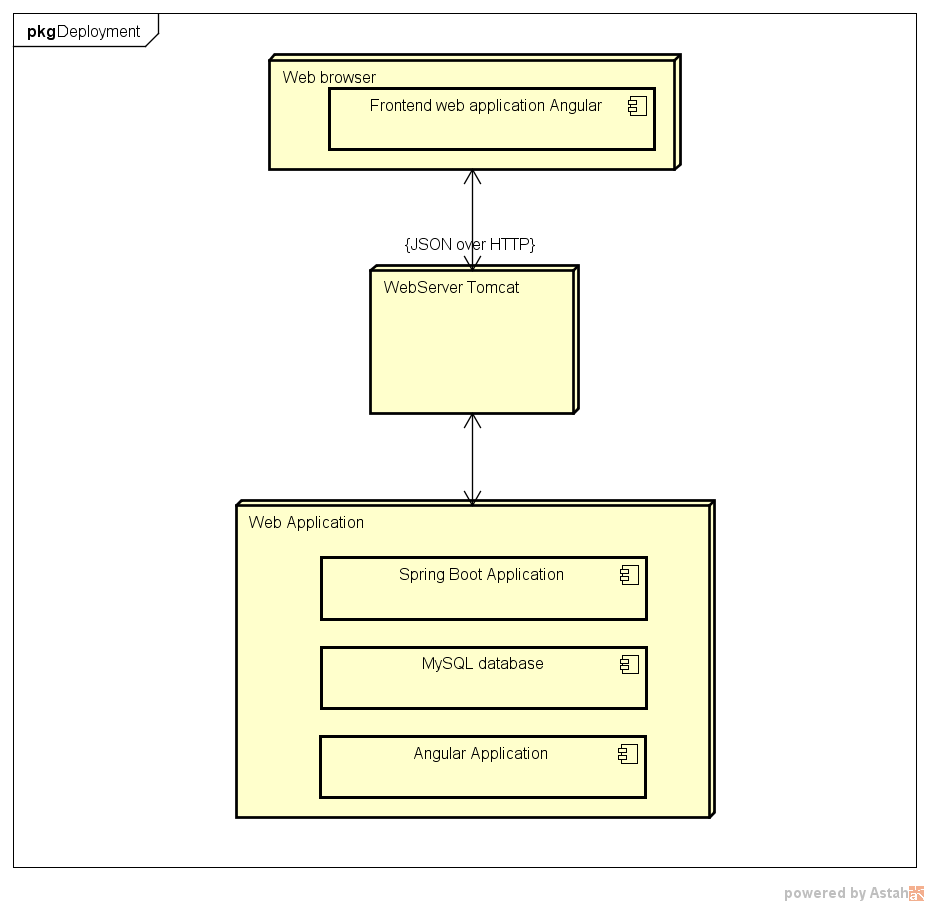
*Hình 16: Minh họa giao diện thống kê số lượng tin đăng*

# CHƯƠNG 4: THIẾT KẾ HỆ THỐNG



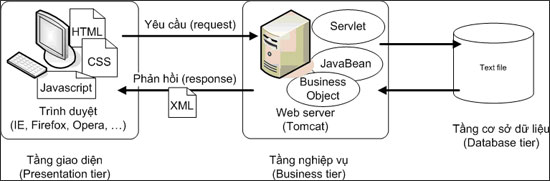
## Kiến trúc hệ thống

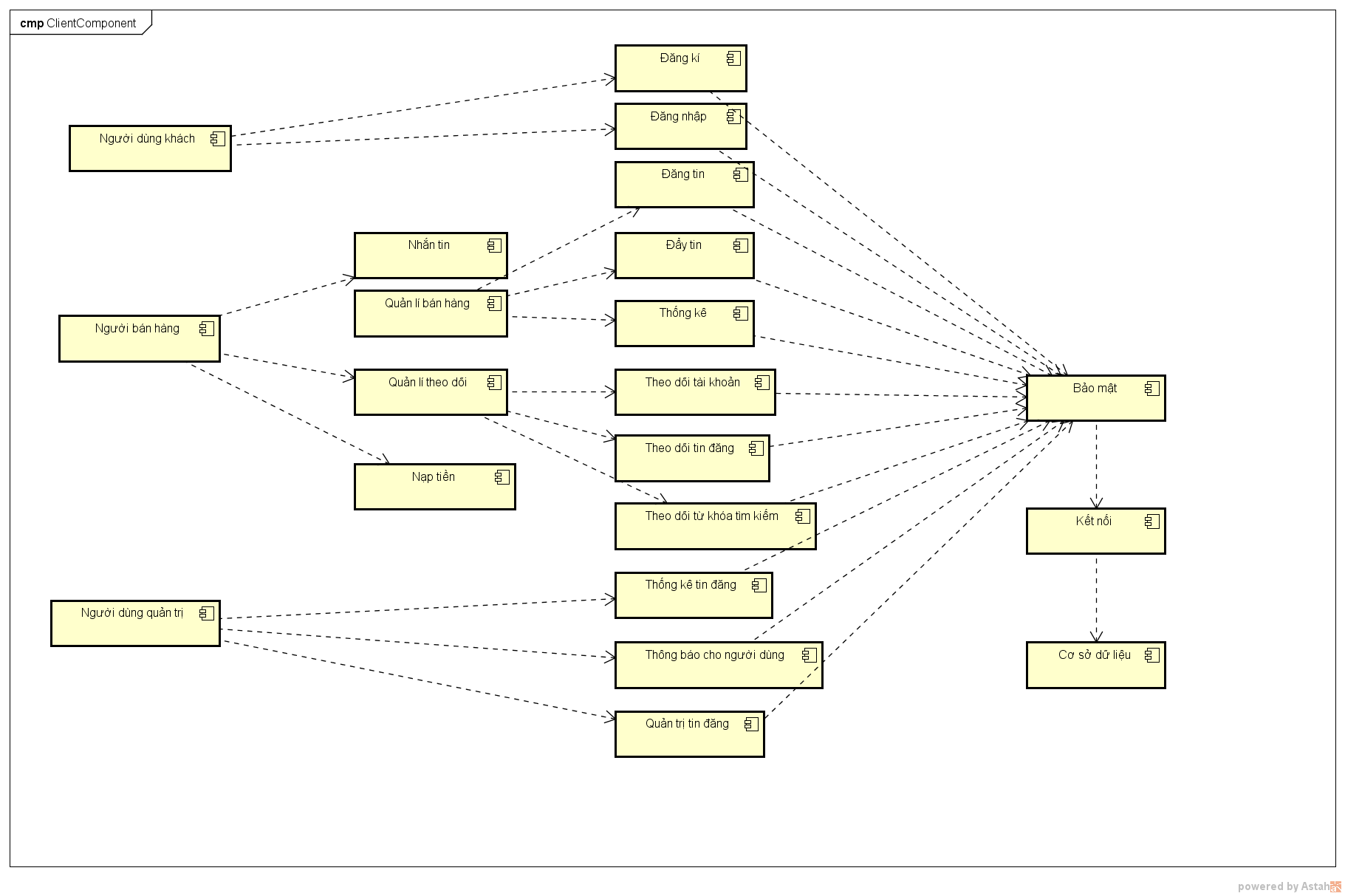
### Kiến trúc triển khai



*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

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

#### Restful WebService

REST viết tắt của Representational State Transfer đã được chọn sử dụng rộng rãi thay cho Web service dựa trên SOAP và WSDL.Nó là một kiểu kiến trúc thường sử dụng để thiết kế webservice,các webservice nó có thể sử dụng bởi nhiều các loại client khác nhau.Ý tưởng chính là sử dụng HTTP đơn giản để gọi thay cho việc sử dụng các cơ chế phức tạp như là CORBAR,RPC hoặc là SOAP.Trong thiết kế cơ bản của rest thì resouces được thực thi bằng tập các động từ thông dụng.

* Để tạo một resouces: thì sử dụng HTTP POST
* Để lấy một resouces: thì sử dung HTTP GET
* Để update một resouces: thì sử dụng HTTP PUT
* Để xóa một resouces: thì sử dụng HTTP DELETE

Khi tạo ra các RESTful API thì phải tuân theo các luật phía trên,thường thì REST dựa trên web service trả lại JSON hoặc XML ,mặc dù nó nó có thể trả các kiểu dữ liệu khác.Client có thể chỉ định loại resource mong muốn,và server sẽ trả lại resource đó bằng cách chỉ rõ Content-Type của resource.

* Spring Boot sử một anotation là @RestController, nó là sự kết hợp của @Controller và @ResponseBody.
* Nếu paramter của một phương thức được đánh dấu với @RequestBody, Spring sẽ bind các tham số của HTTP request tới các parameter đó,Spring sẽ sử dụng HTTP message converters để convert request body tới domain object,dựa vào ACCEPT or Content-Type trong header của request.
* Nếu một phương thức được đánh dấu với @ResponseBody,spring sẽ gắn kết quả vào HTTP response, Spring sẽ sử dụng HTTP Message converters để convert giá tri về HTTP response ,dựa vào ACCEPT or Content-Type trong header của request.
* ResponseEntity: nó đại diện cho toàn bộ HTTP response, người lập trình có thể kiểm soát bất kì bên trong nó, có thể chỉ rõ status code,header, và body.
* @PathVariable, @RestController, @RequestBody, @RequestParam và ResponseEntity là tất cả những gì cần thiết để implement một REST API trong Spring.

#### Auth0 JWT

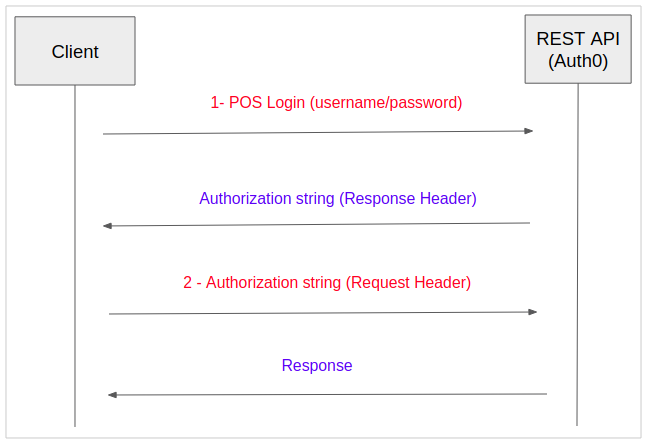
Giả sử rằng có một RESTful API được viết trên Spring Boot, các Client (ứng dụng khác) có thể gọi đến RESTful API đó và nhận kết quả về.

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.



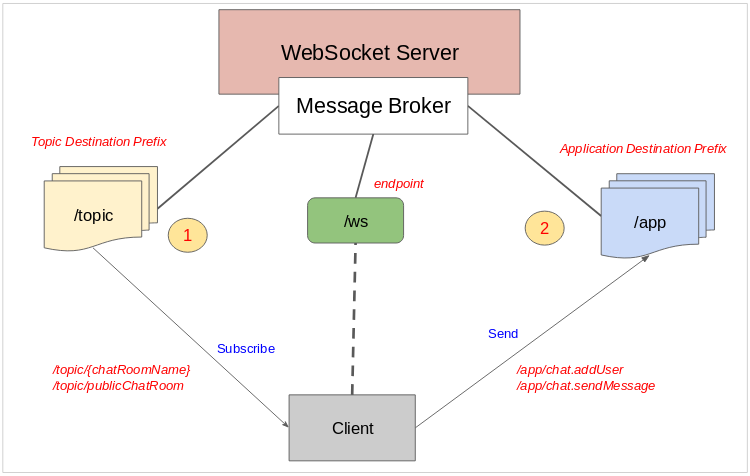
*Hình 19: Luồng xử lí Auth0*

#### Websocket

* 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.
* STOMP: Streaming Text Oriented Messaging Protocol hay còn gọi là 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.
* 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. 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). Hình dưới đây mô tả cấu trúc của MessageBroker



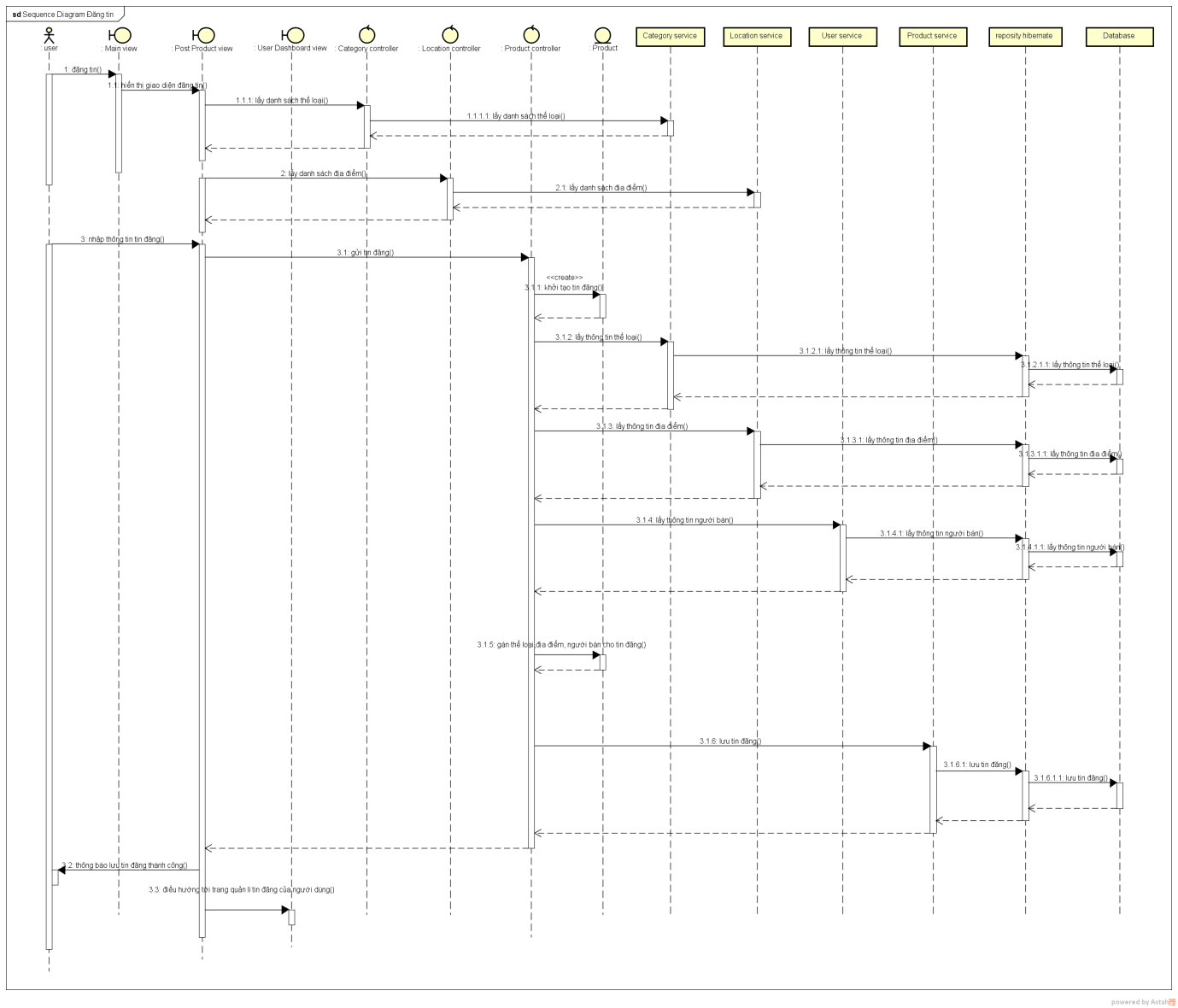
* + Điểm đến (1) là các chủ đề (topic) mà client có thể "đăng ký theo dõi" (subscribe), khi một chủ đề có tin nhắn, các tin nhắn sẽ được gửi đến cho những client đã đăng ký chủ đề này.
  + Điểm đến (2) là các nơi mà client có thể gửi tin nhắn tới WebSocket Server.
* SockJS: là một thư viện JavaScript.Không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

Controller là tập hợp các RESTful API, Controller sẽ gọi đến các Service hoặc các Hibernate Reposity để thực thi nghiệp vụ. Sau khi hoàn thành, kết quả trả về được chuyển sang định dạng JSON và gắn kết quả vào HTTP ressponse, RESTful API sẽ trả về dữ liệu kiểu ResponseEntity< >.

### Đăng tin

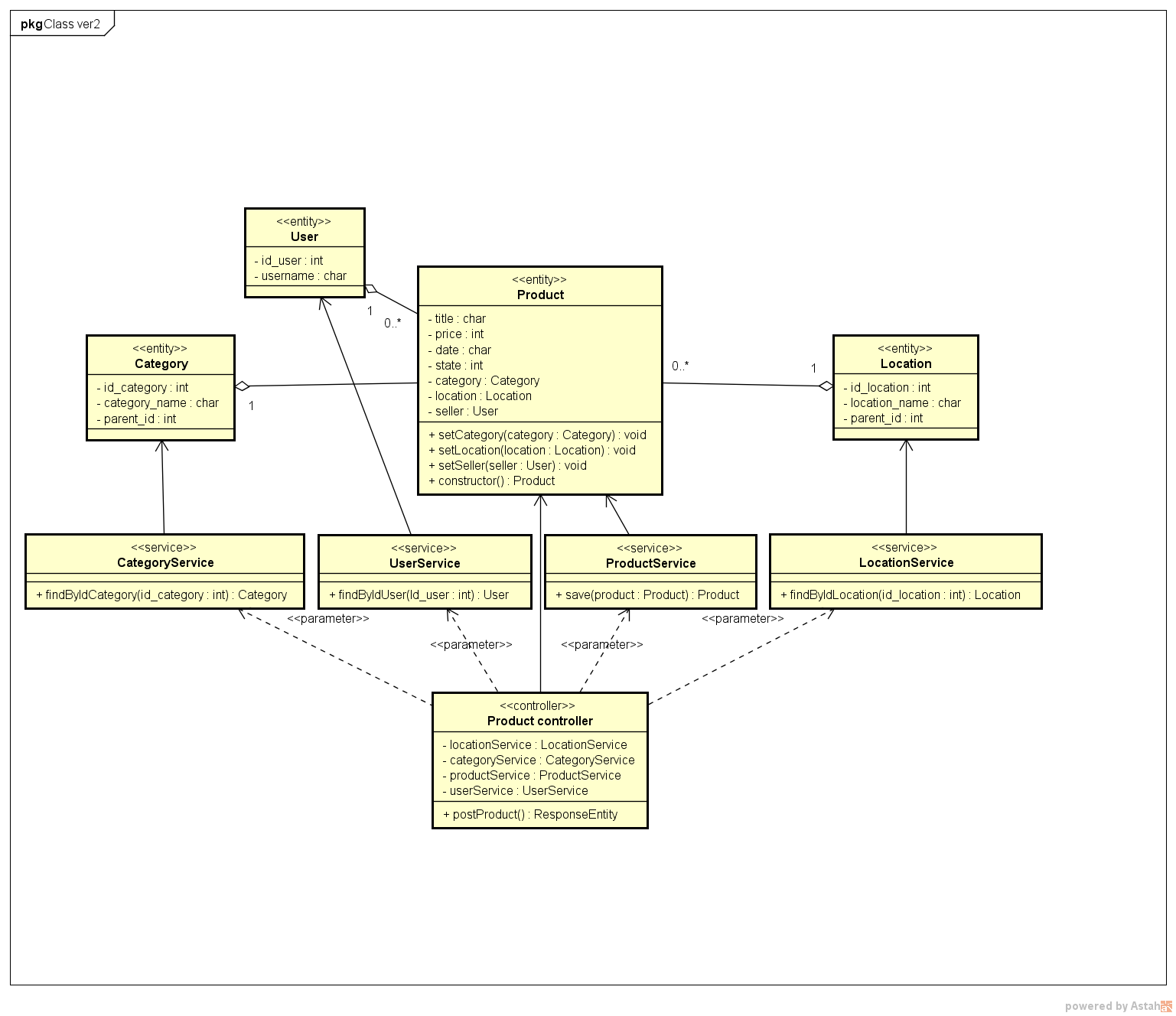
#### Biểu đồ trình tự



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

#### Biểu đồ lớp

##### Biểu đồ



*Biểu đồ 2: Biểu đồ lớp Đăng tin*

##### Mô tả lớp

|  |  |
| --- | --- |
| **User**  Đối tượng lưu thông tin tài khoản người dùng gồm 2 thuộc tính chính sau: | |
| **Thuộc tính** | **Mô tả** |
| id\_user | Khóa chính, mỗi User sẽ có một khóa chính riêng biệt,khóa chính sẽ đại diện cho User đó |
| username | Tên tài khoản dùng để đăng nhập |

*Bảng 1: Bảng mô tả lớp User-ca sử dụng Đăng tin*

|  |  |
| --- | --- |
| **Category**  Đối tượng lưu thông tin thể loại của một tin đăng, gồm 3 trường chính sau: | |
| **Thuộc tính** | **Mô tả** |
| id\_category | Khóa chính |
| category\_name | Tên thể loại |
| parent\_id | Mỗi thể loại sẽ thuộc một thể loại tổng quát hơn, parent\_id sẽ lưu id\_category của thể loại tổng quát đó |

*Bảng 2: Bảng mô tả lớp Category-ca sử dụng Đăng tin*

|  |  |
| --- | --- |
| **Location**  Đối tượng lưu thông tin vị trí của tin đăng, gồm 3 trường chính sau: | |
| **Thuộc tính** | **Mô tả** |
| id\_location | Khóa chính |
| location\_name | tên vị trí, là tên Thành Phố/Tỉnh hoặc Quận/Huyện |
| parent\_id | Nếu vị trí đó là Quận/Huyện, parent\_id sẽ lưu giá trị id\_location của vị trí có tên là Thành Phố/Tỉnh |

*Bảng 3: Bảng mô tả lớp Location-ca sử dụng Đăng tin*

|  |  |
| --- | --- |
| **Product**  Đối tượng lưu thông tin của tin đăng,gồm 3 trường chính sau: | |
| **Thuộc tính** | **Mô tả** |
| id\_product | Khóa chính |
| title | Tên sản phẩm muốn đăng |
| price | Giá bán |
| date | Thời gian đăng |
| state | Tình trạng hiện tại của sản phẩm |
| category | Thể loại của tin đăng |
| location | Vị trí giao hàng |
| seller | Người bán |
| **Phương thức** | **Mô tả** |
| setCategory(Category c) | Nhận đầu vào là đối tượng Category,phương thức cập nhật thông tin thể loại của tin đăng |
| setLocation(Location l) | Nhận đầu vào là đối tượng Location,phương thức cập nhật thông tin vị trí giao hàng |
| SetSeller(User u) | Nhận đầu vào là đối tượng User,phương thức cập nhật thông tin người bán |

*Bảng 4: Bảng mô tả lớp Product-ca sử dụng Đăng tin*

|  |  |
| --- | --- |
| **CategoryService**  Lớp service xử lí các tác vụ liên quan đến đối tượng Category | |
| **Hàm** | **Mô tả** |
| findByIdcategory(int id) | Trả về một đối tượng Category có giá trị *id\_category* bằng tham số đầu vào *id* |

*Bảng 5: Bảng mô tả lớp CategoryService-ca sử dụng Đăng tin*

|  |  |
| --- | --- |
| **LocationService**  Lớp service xử lí các tác vụ liên quan đến đối tượng Location | |
| **Hàm** | **Mô tả** |
| findByIdlocation(int id) | Trả về một đối tượng Location có giá trị *id\_location* bằng tham số đầu vào *id* |

*Bảng 6: Bảng mô tả lớp LocationService-ca sử dụng Đăng tin*

|  |  |
| --- | --- |
| **UserService**  Lớp service xử lí các tác vụ liên quan đến đối tượng User | |
| **Hàm** | **Mô tả** |
| findByIduser(int id) | Trả về một đối tượng User có giá trị *id\_user* bằng tham số đầu vào *id* |

*Bảng 7: Bảng mô tả lớp UserService-ca sử dụng Đăng tin*

|  |  |
| --- | --- |
| **ProductService**  Lớp service xử lí các tác vụ liên quan đến đối tượng Product(tin đăng) | |
| **Hàm** | **Mô tả** |
| save(Product product) | Lưu đối tượng Product vào DATABASE và trả về là một đối tượng Product được lưu thành công |

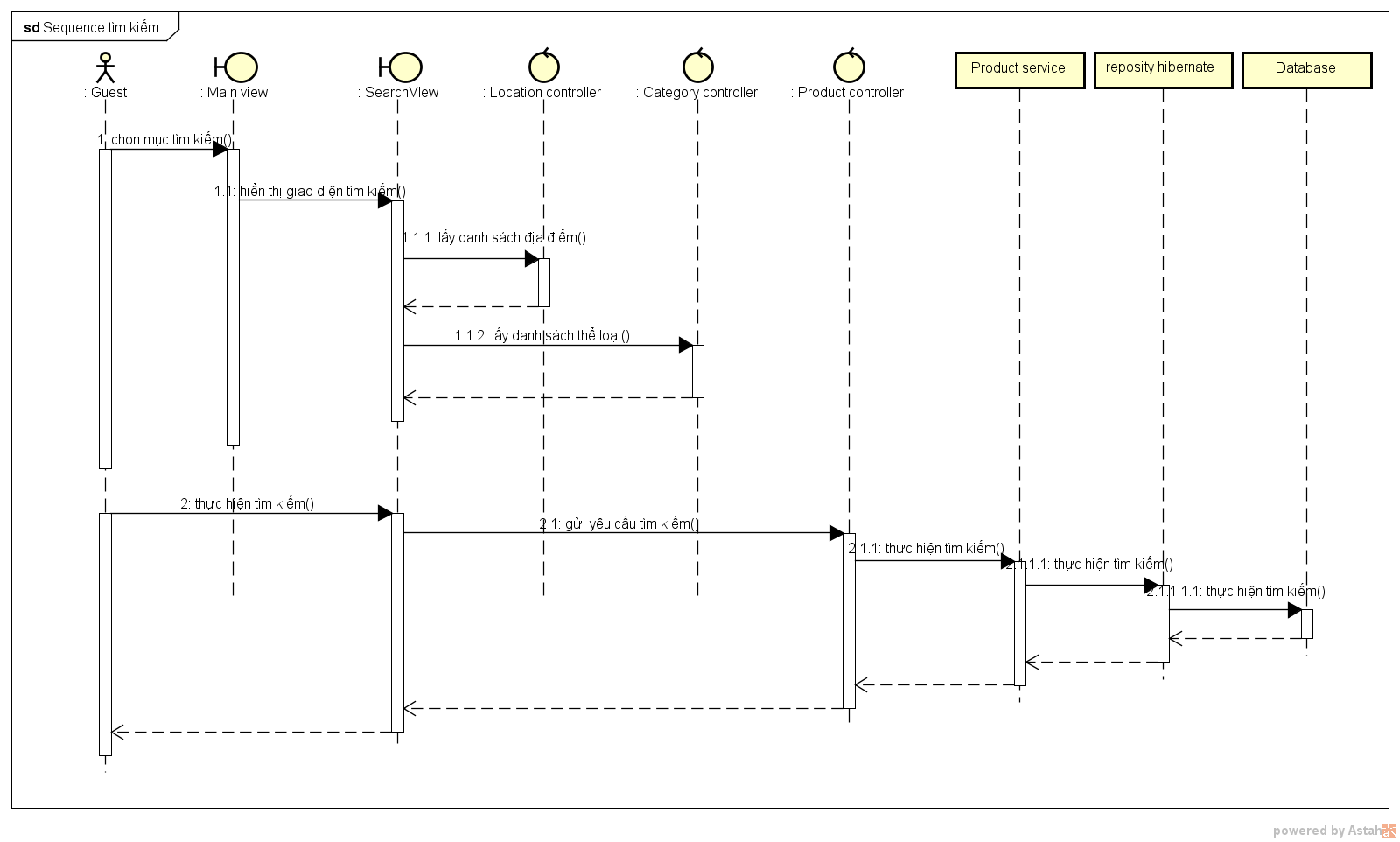
*Bảng 8: Bảng mô tả lớp ProductService-ca sử dụng Đăng tin*

|  |  |
| --- | --- |
| **ProductController**  Lớp Controller xử lí các yêu cầu từ người dùng liên quan đến đối tượng Product | |
| **Hàm** | **Mô tả** |
| postProduct(String param) | Hàm xử lí yêu cầu lưu tin đăng từ người dùng, nhận đầu vào là xâu JSON |

*Bảng 9: Bảng mô tả lớp ProductController-ca sử dụng Đăng tin*

### Tìm kiếm

#### Biểu đồ trình tự



*Biểu đồ 3: Biểu đồ trình tự Tìm kiếm*

#### Biểu đồ lớp

##### Biểu đồ



*Biểu đồ 4: Biểu đố lớp Tìm kiếm*

##### Mô tả lớp

|  |  |
| --- | --- |
| **CategoryService**  Lớp service xử lí các tác vụ liên quan đến đối tượng Category | |
| **Hàm** | **Mô tả** |
| findAllCategory() | Trả về danh sách đối tượng Category |

*Bảng 10: Bảng mô tả lớp CategoryService-ca sử dụng Tìm kiếm*

|  |  |
| --- | --- |
| **LocationService**  Lớp service xử lí các tác vụ liên quan đến đối tượng Location | |
| **Hàm** | **Mô tả** |
| findAllLocation() | Trả về danh sách đối tượng Location |

*Bảng 11: Bảng mô tả lớp LocationService-ca sử dụng Tìm kiếm*

|  |  |
| --- | --- |
| **ProductService**  Lớp service xử lí các tác vụ liên quan đến đối tượng Product(tin đăng) | |
| **Hàm** | **Mô tả** |
| filterSearch(String param) | Nhận đầu vào là xâu JSON( lưu nội dung tìm kiếm của người dùng) , trả về danh sách Product |

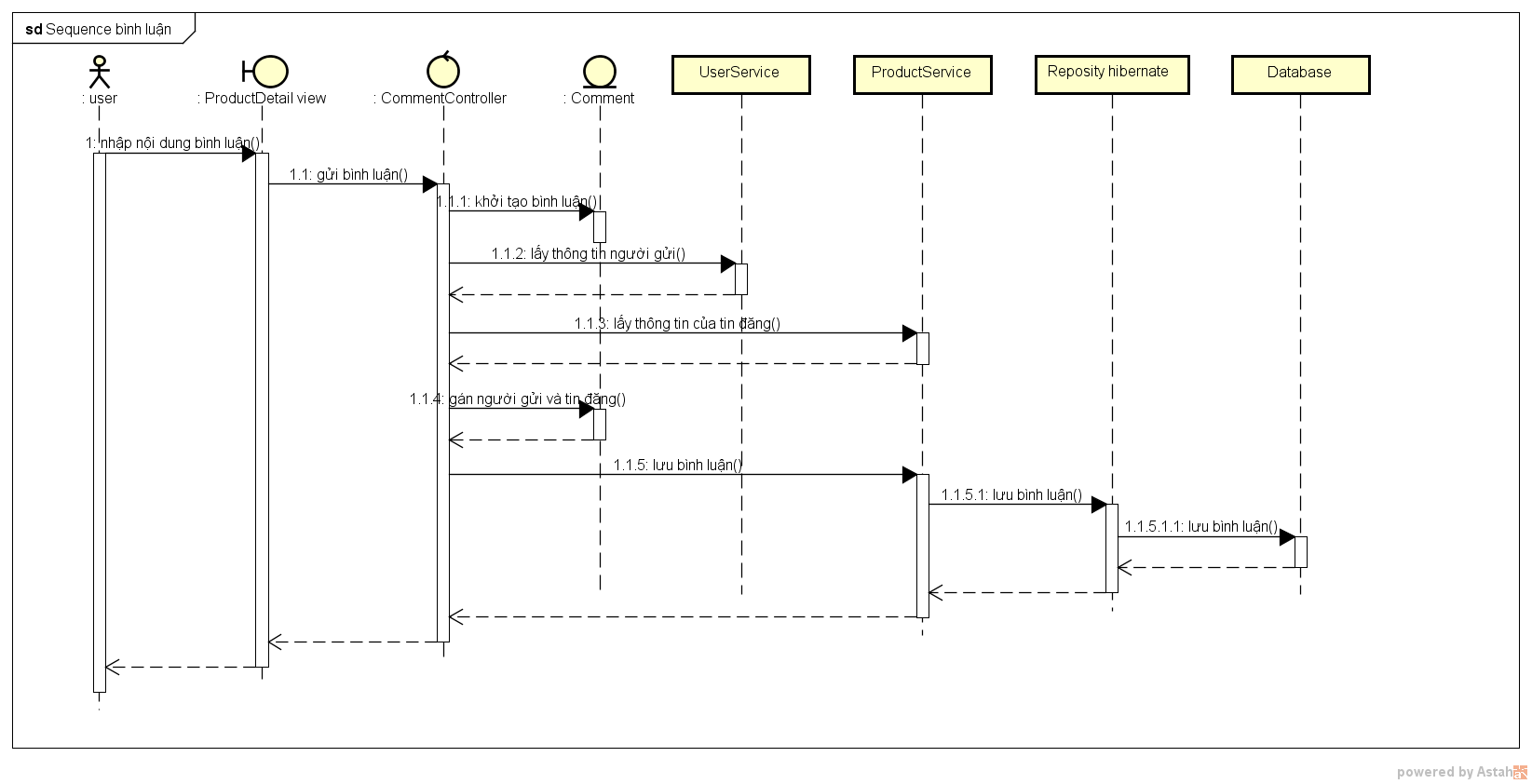
*Bảng 12: Bảng mô tả lớp ProductService-ca sử dụng Tìm kiếm*

|  |  |
| --- | --- |
| **ProductController**  Lớp Controller xử lí các yêu cầu từ người dùng liên quan đến đối tượng Product | |
| **Hàm** | **Mô tả** |
| filterSearch(String param) | Gọi phương thức filterSearch của ProductService, chuyển kết quả về dạng xâu và trả lại kết quả cho người dùng |

*Bảng 13: Bảng mô tả lớp ProductController-ca sử dụng Tìm kiếm*

### Bình luận

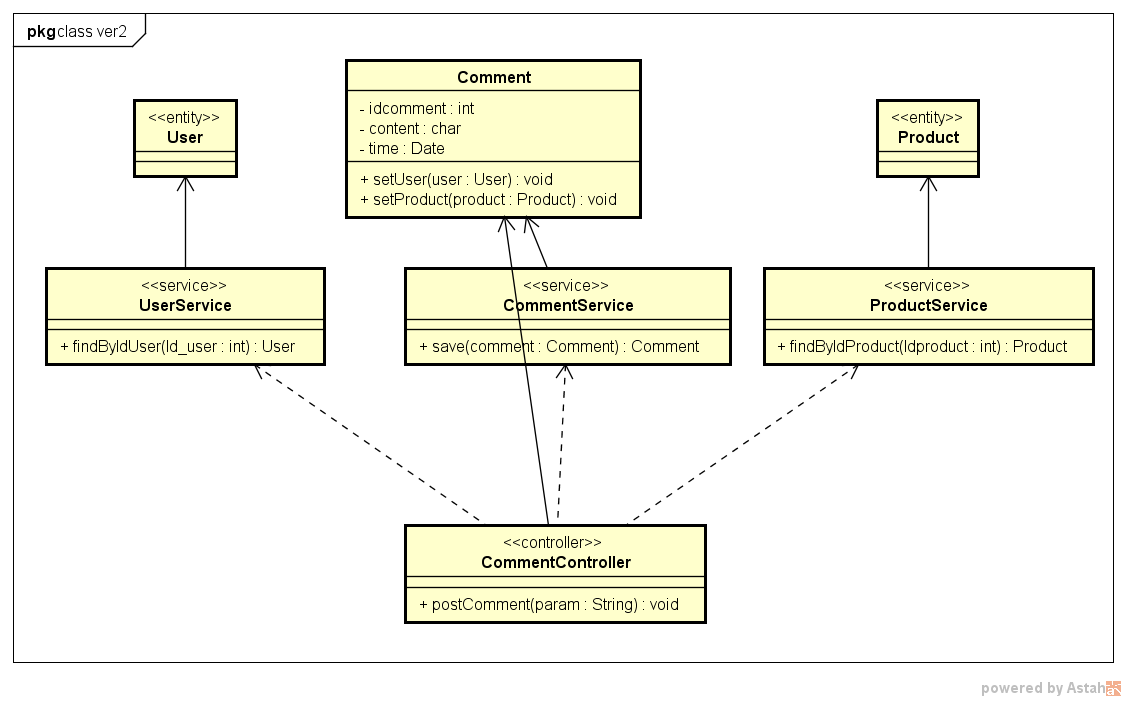
#### *Biểu đồ trình tự*



*Biểu đồ 5: Biểu đồ trình tự Bình luận*

#### Biểu đồ lớp

##### Biểu đồ



*Biểu đồ 6: Biểu đồ lớp Bình luận*

##### Mô tả lớp

|  |  |
| --- | --- |
| **Comment**  Đối tượng lưu thông tin bình luận,gồm các trường chính sau: | |
| **Thuộc tính** | **Mô tả** |
| idcomment | Khóa chính |
| content | Nội dung bình luận |
| time | Thời gian bình luận |
| from | Đối tượng User, thông tin người gửi |
| product | Đối tượng Product, thông tin tin đăng |
| **Phương thức** | **Mô tả** |
| setFrom(User u) | Nhận đầu vào là đối tượng User,phương thức cập nhật thông tin người gửi của bình luận |
| setProduct(Product p) | Nhận đầu vào là đối tượng Product,phương thức cập nhật thông tin tin đăng của bình luận |

*Bảng 14: Bảng mô tả lớp Comment-ca sử dụng Bình luận*

|  |  |
| --- | --- |
| **CommentRepository**  Interface hibernate xử lí các tác vụ liên quan đến đối tượng Comment | |
| **Hàm** | **Mô tả** |
| save(Comment c) | Lưu đối tượng bình luận vào database |

*Bảng 15: Bảng mô tả lớp CommentRepository-ca sử dụng Bình luận*

|  |  |
| --- | --- |
| **ProductService**  Lớp service xử lí các tác vụ liên quan đến đối tượng Product(tin đăng) | |
| **Hàm** | **Mô tả** |
| findByIdProduct(id int) | Trả về đối tượng Product có giá trị trường *idproduct* bằng đầu vào *id* |

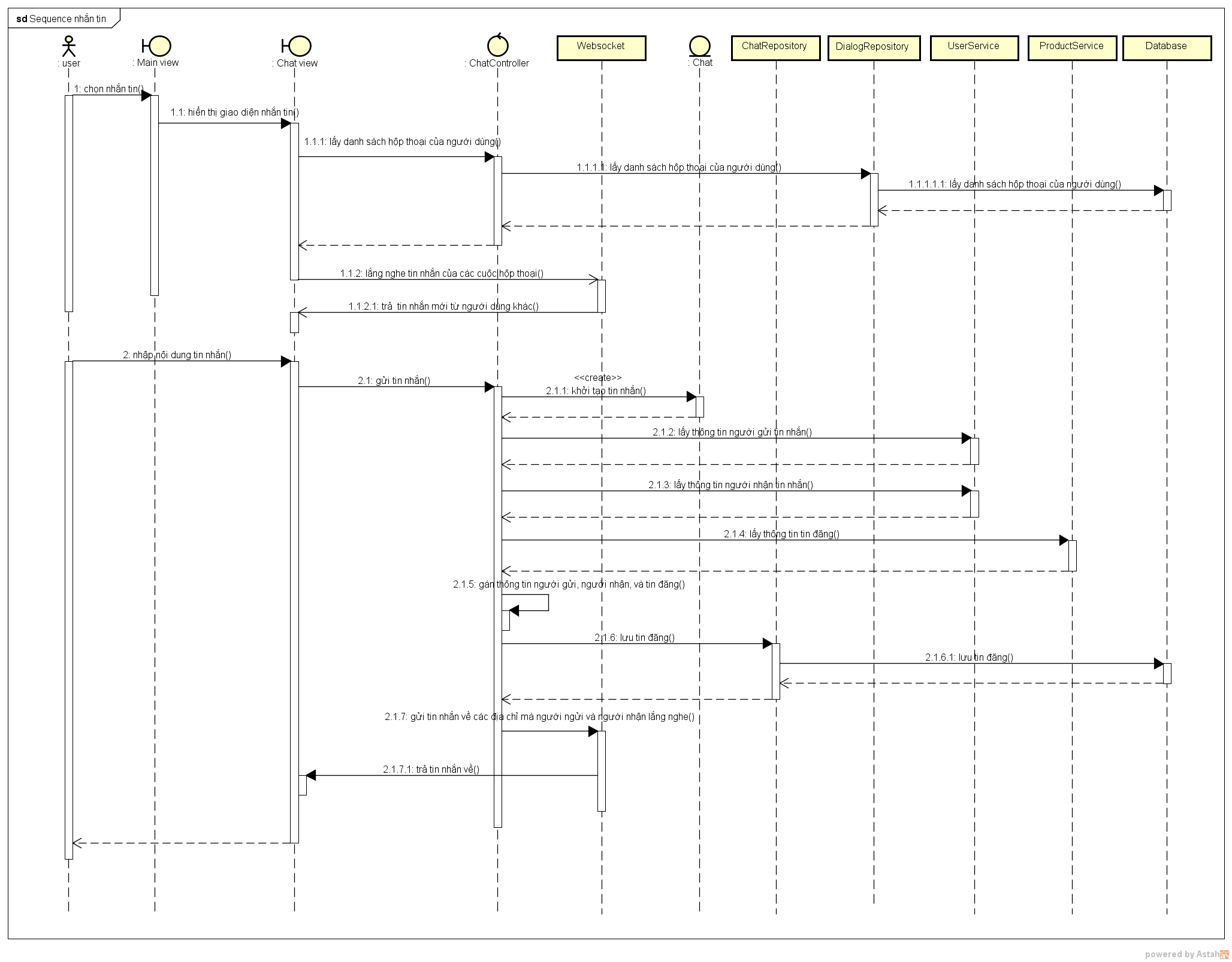
*Bảng 16: Bảng mô tả lớp ProductService-ca sử dụng Bình luận*

|  |  |
| --- | --- |
| **CommentController**  Lớp Controller xử lí các yêu cầu từ người dùng liên quan đến đối tượng Comment | |
| **Hàm** | **Mô tả** |
| postComment(String param) | Phương thức xử lí yêu cầu gửi comment của người dùng |

*Bảng 17: Bảng mô tả lớp CommenttController-ca sử dụng Bình luận*

### Nhắn tin

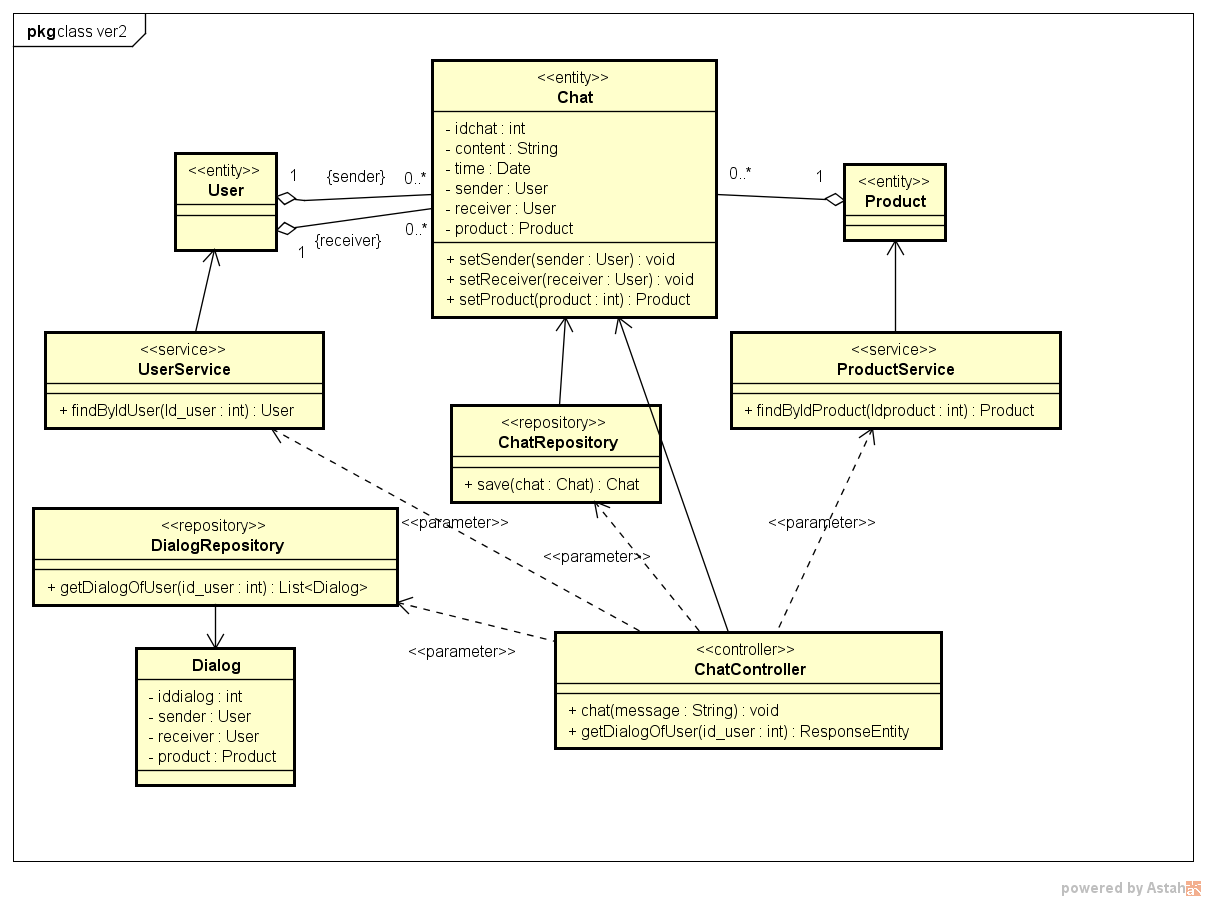
#### Biểu đồ trình tự



*Biểu đồ 7: Biểu đồ trình tự Nhắn tin*

#### Biểu đồ lớp

##### Biểu đồ



*Biểu đồ 8: Biểu đồ lớp Nhắn tin*

##### Mô tả lớp

|  |  |
| --- | --- |
| **Chat**  Đối tượng lưu nội dung tin nhắn,gồm các trường chính sau: | |
| **Thuộc tính** | **Mô tả** |
| idchat | Khóa chính |
| content | Nội dung tin nhắn |
| time | Thời gian bình luận |
| sender | Đối tượng User, thông tin người gửi |
| receiver | Đối tượng User, thông tin người nhận |
| product | Đối tượng Product, thông tin tin đăng |
| **Phương thức** | **Mô tả** |
| setSender(User u) | Nhận đầu vào là đối tượng User,phương thức cập nhật thông tin người gửi tin nhắn |
| setReceiver(User u) | Nhận đầu vào là đối tượng User,phương thức cập nhật thông tin người nhận tin nhắn |
| setProduct(Product p) | Nhận đầu vào là đối tượng Product,phương thức cập nhật thông tin tin đăng của bình luận |

*Bảng 18: Bảng mô tả đối tượng Chat-ca sử dụng Nhắn tin*

|  |  |
| --- | --- |
| **Dialog**  Đối tượng lưu hộp thoại,gồm các trường chính sau: | |
| **Thuộc tính** | **Mô tả** |
| iddialog | Khóa chính |
| sender | Đối tượng User, thông tin người gửi |
| receiver | Đối tượng User, thông tin người nhận |
| product | Đối tượng Product, thông tin tin đăng |

*Bảng 19: Bảng mô tả đối tượng Dialog-ca sử dụng Nhắn tin*

|  |  |
| --- | --- |
| **ChatRepository**  Interface hibernate xử lí các tác vụ liên quan đến đối tượng Chat | |
| **Hàm** | **Mô tả** |
| save(Chat c) | Lưu đối tượng tin nhắn vào database |

*Bảng 20: Bảng mô tả lớp ChatRepository-ca sử dụng Nhắn tin*

|  |  |
| --- | --- |
| **DialogRepository**  Interface hibernate xử lí các tác vụ liên quan đến đối tượng Dialog | |
| **Hàm** | **Mô tả** |
| getDialogOfUser(id\_user int) | Trả về danh sách đối tượng Dialog có giá trị trường *sender* hoặc *receiver* bằng đầu vào *id\_user* |

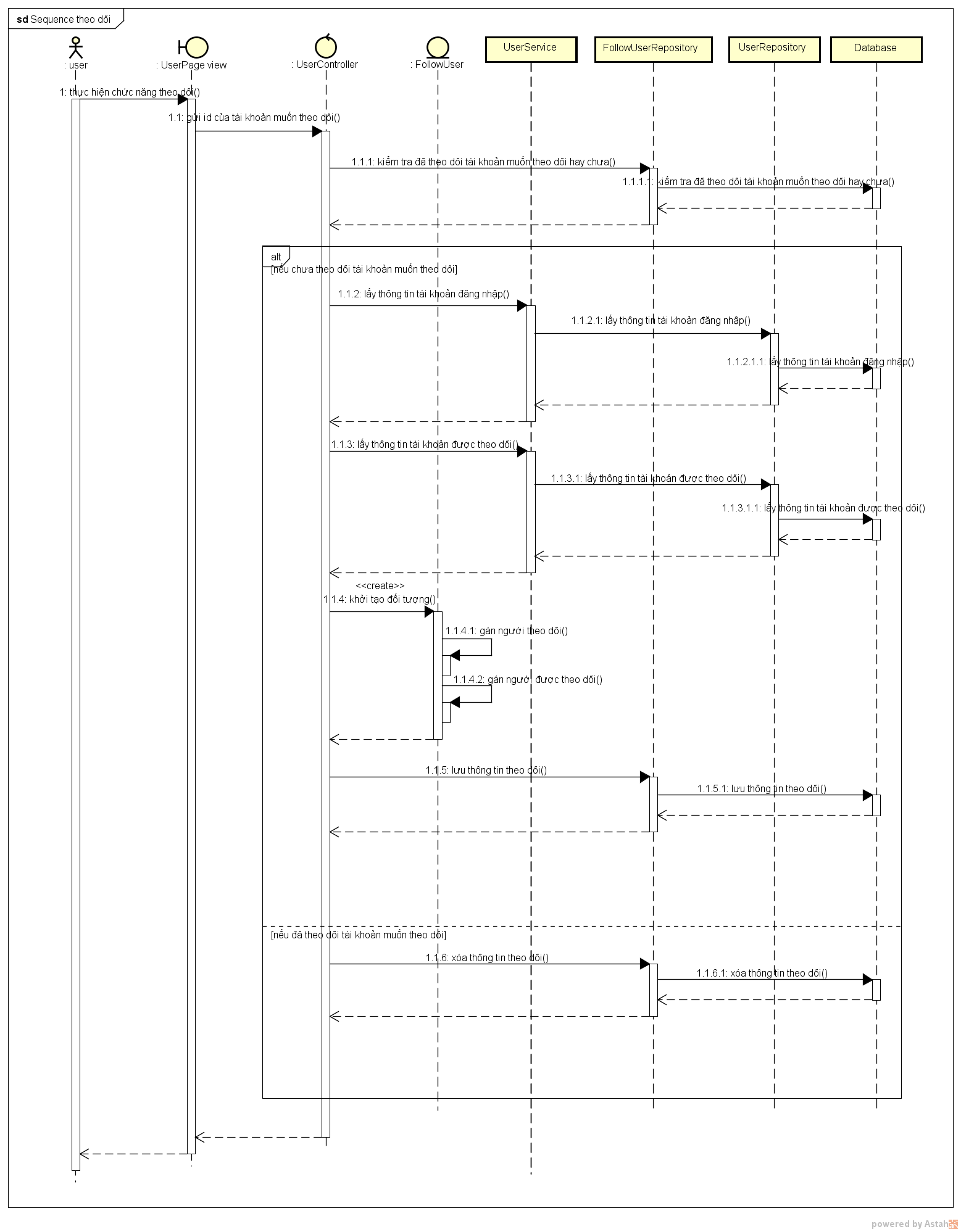
*Bảng 21: Bảng mô tả lớp DialogRepository-ca sử dụng Nhắn tin*

|  |  |
| --- | --- |
| **ChatController**  Lớp Controller xử lí các yêu cầu từ người dùng liên quan đến đối tượng Chat | |
| **Hàm** | **Mô tả** |
| chat(String message) | Phương thức lắng nghe yêu cầu gửi tin nhắn dựa trên websocket, sau đó gửi về các địa chỉ mà người gửi và người nhận đăng kí |
| getDialogOfUser(id\_user int) | Phương thức xử lí yêu cầu lấy danh sách hộp thoại của một User, nhận đầu vào là *id\_user* |

*Bảng 22: Bảng mô tả lớp ChatController-ca sử dụng Nhắn tin*

### Theo dõi tài khoản

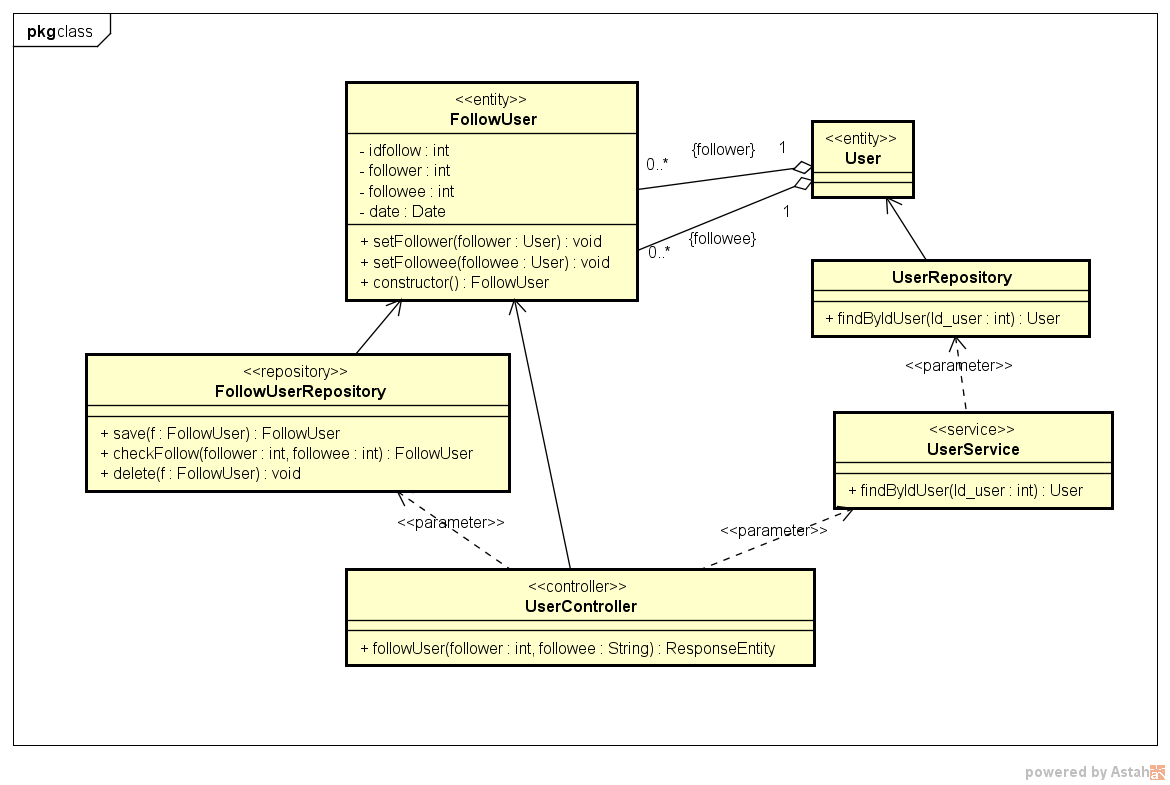
#### *Biểu đồ trình tự*



*Biểu đồ 9: Biểu đồ trình tự Theo dõi tài khoản*

#### Biểu đồ lớp

##### Biểu đồ



*Biểu đồ 10: Biểu đồ lớp Theo dõi tài khoản*

##### Mô tả lớp

|  |  |
| --- | --- |
| **FollowUser**  Đối tượng lưu thông tin theo dõi,gồm các trường chính sau: | |
| **Thuộc tính** | **Mô tả** |
| idfollow | Khóa chính |
| follower | Tài khoản theo dõi |
| followee | Tài khoản được theo dõi |
| date | Thời gian |
| **Phương thức** | **Mô tả** |
| setFollower(User u) | Nhận đầu vào là đối tượng User,phương thức cập nhật thông tin tài khoản theo dõi |
| setFollowee(Product p) | Nhận đầu vào là đối tượng User,phương thức cập nhật thông tin tài khoản được theo dõi |

*Bảng 23: Bảng mô tả lớp FollowUser-ca sử dụng Theo dõi tài khoản*

|  |  |
| --- | --- |
| **FollowUserRepository**  Interface hibernate xử lí các tác vụ liên quan đến đối tượng FollowUser | |
| **Hàm** | **Mô tả** |
| save(FollowUser c) | Lưu đối tượng FollowUser vào database |
| checkFollow(int follower, int followee) | Kiểm tra tài khoản có giá trị *iduser* là *follower* đã theo dõi tài khoản có giá trị *iduser* là *followee* hay chưa, nếu chưa trả về giá trị *null*, nếu đã theo dõi thì trả về một đối tượng FollowUser |
| delete(FollowUser f) | Xóa một đối tượng FollowUser |

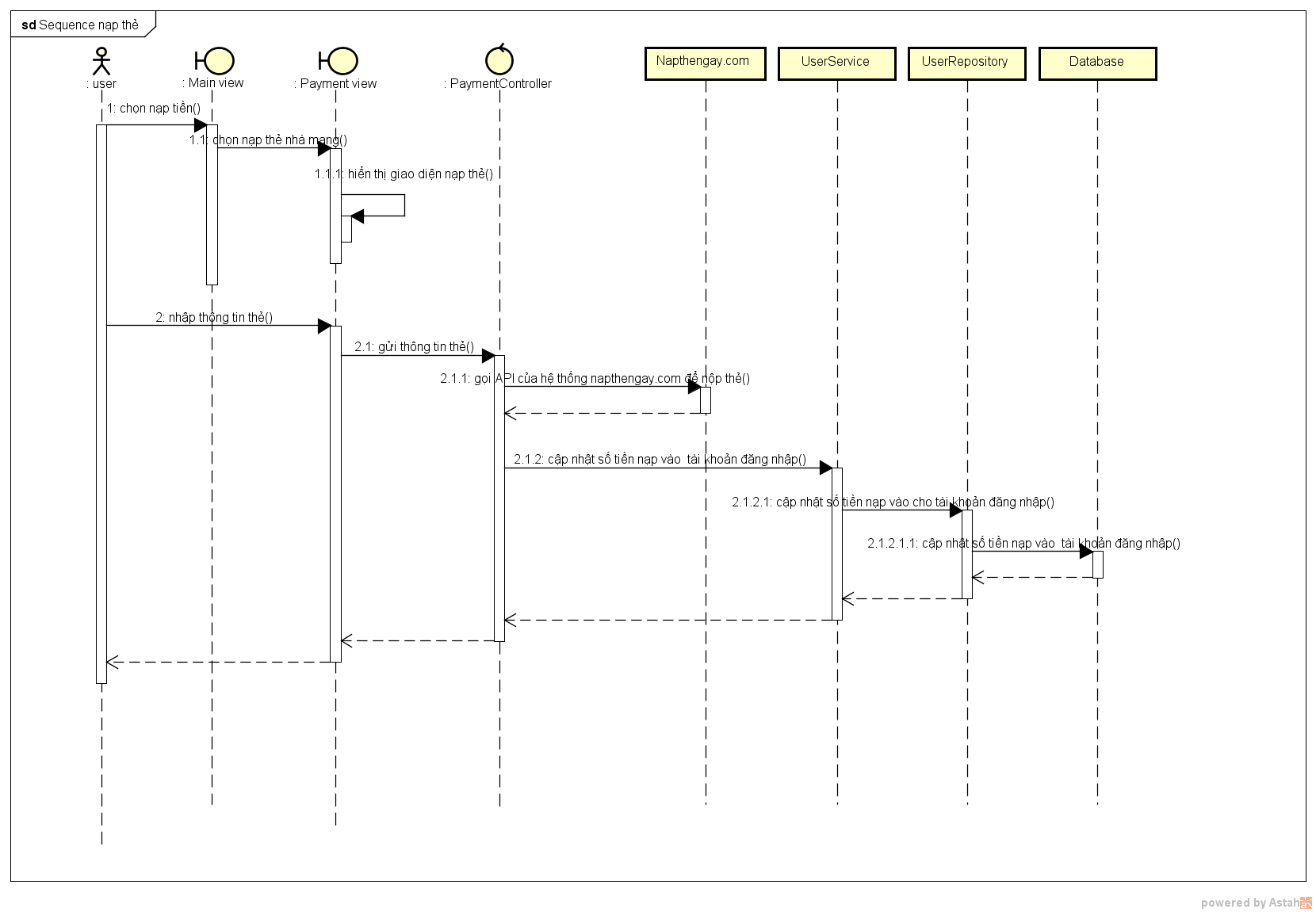
*Bảng 24: Bảng mô tả lớp FollowUserRepository-ca sử dụng Theo dõi tài khoản*

|  |  |
| --- | --- |
| **UserController**  Lớp Controller xử lí các yêu cầu từ người dùng liên quan đến đối tượng User | |
| **Hàm** | **Mô tả** |
| followUser(int follower,String followee) | Phương thức POST xử lí yêu cầu theo dõi người dùng, nhận đầu vào là một *@PathVariable int follower*( giá trị iduser của tài khoản thực hiện chức năng theo dõi )và một *@RequestParam String follower* ( giá trị iduser của tài khoản muốn theo dõi) |

*Bảng 25: Bảng mô tả lớp UserController-ca sử dụng Theo dõi tài khoản*

### Nạp thẻ cào

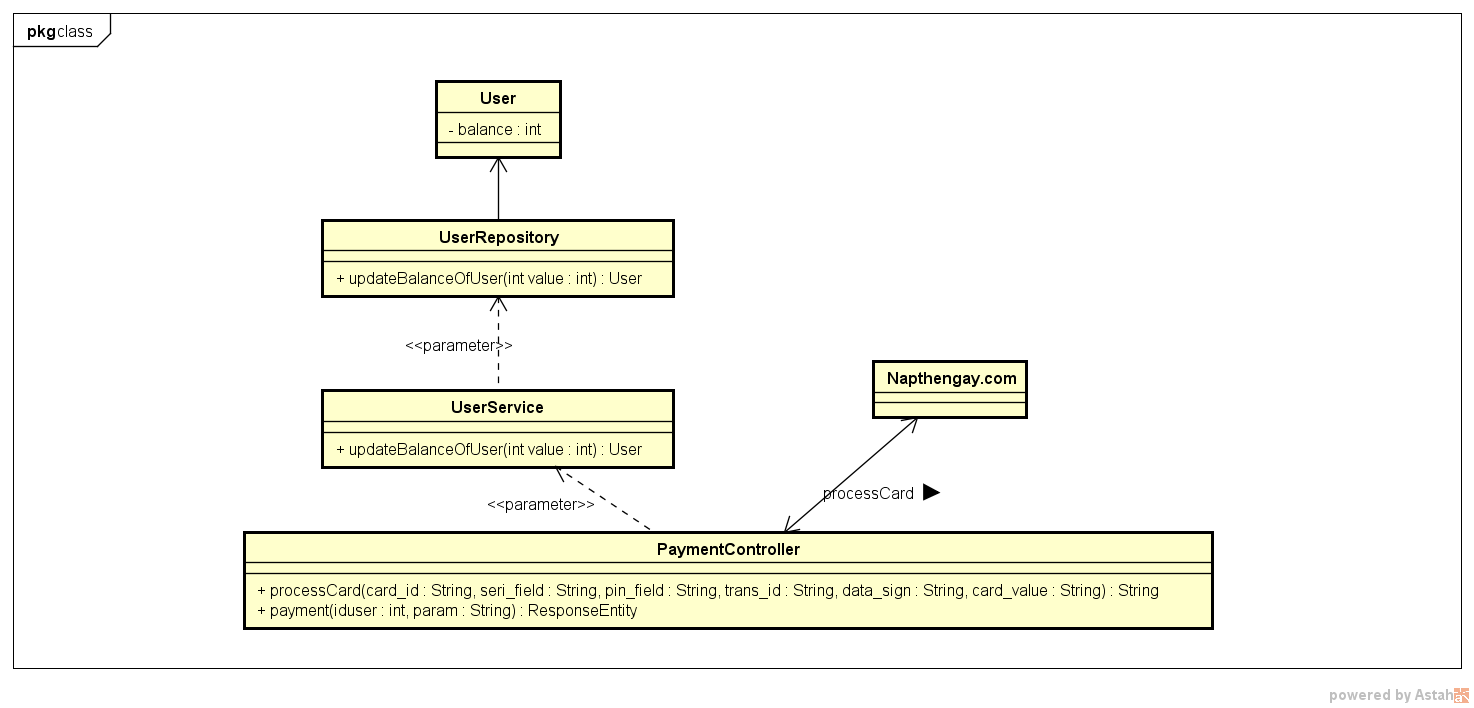
#### *Biểu đồ trình tự*



*Biểu đồ 11: Biểu đồ trình tự Nạp thẻ cào*

#### Biểu đồ lớp

##### Biểu đồ



*Biểu đồ 12: Biểu đồ lớp Nạp thẻ cào*

##### Mô tả lớp

|  |  |
| --- | --- |
| **User**  Đối tượng lưu thông tin tài khoản người dùng | |
| **Thuộc tính** | **Mô tả** |
| balance | Số tiền đã nạp vào |

*Bảng 26: Bảng mô tả lớp User-ca sử dụng Nạp thẻ cào*

|  |  |
| --- | --- |
| **UserRepository**  Interface hibernate xử lí các tác vụ liên quan đến đối tượng User | |
| **Hàm** | **Mô tả** |
| updateBalanceOfUser(int balance) | Cập nhật giá trị balance |

*Bảng 27: Bảng mô tả lớp UserRepository- ca sử dụng Nạp thẻ cào*

|  |  |
| --- | --- |
| **UserRepoService**  Lớp service xử lí các tác vụ liên quan đến đối tượng User | |
| **Hàm** | **Mô tả** |
| updateBalanceOfUser(int balance) | Cập nhật giá trị balance |

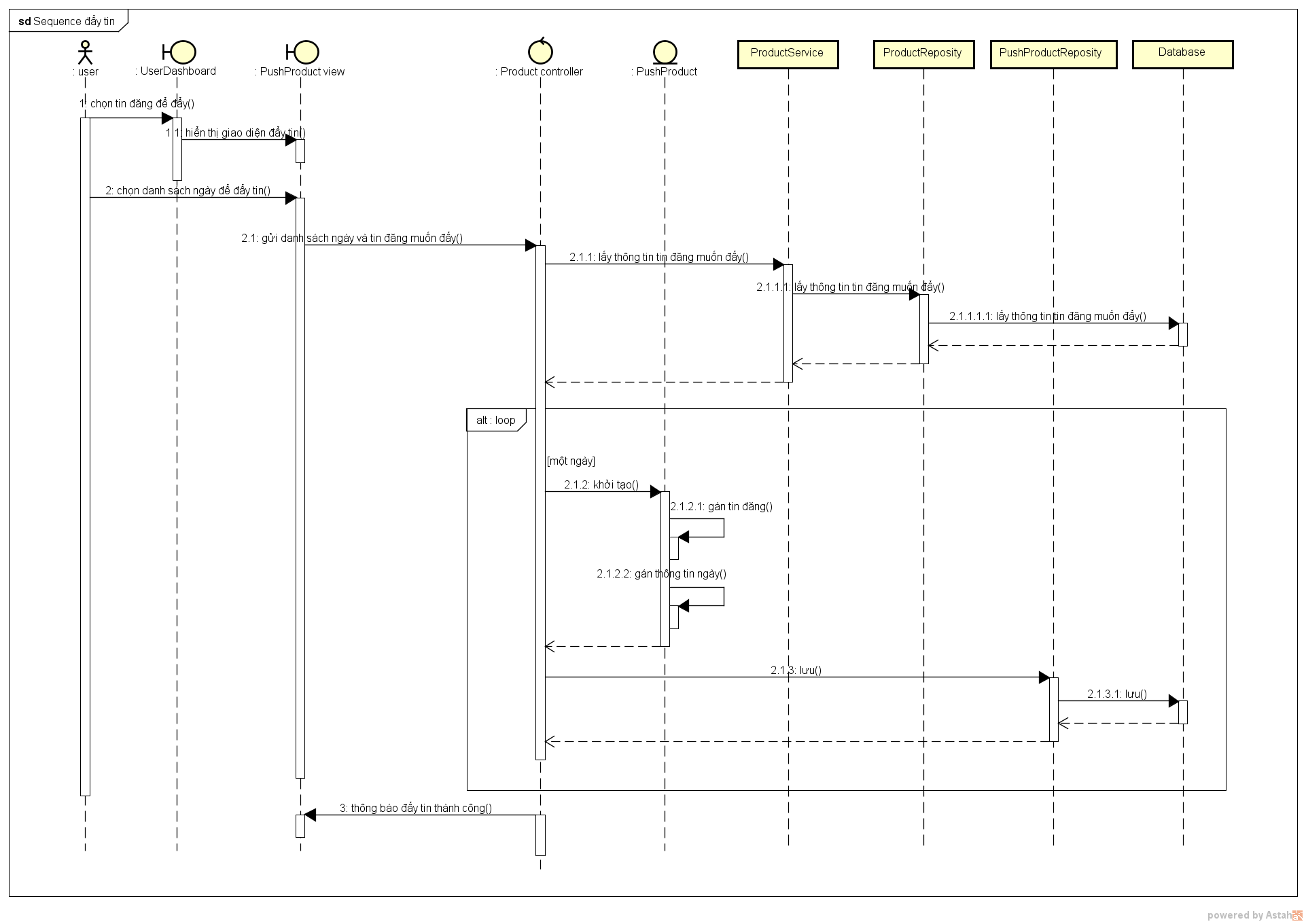
*Bảng 28: Bảng mô tả lớp UserService- ca sử dụng Nạp thẻ cào*

|  |  |
| --- | --- |
| **UserController**  Lớp Controller xử lí các yêu cầu từ người dùng liên quan đến đối tượng User | |
| **Hàm** | **Mô tả** |
| processCard(String card\_id , String seri\_field, String pin\_field, String trans\_id, String data\_sign, String card\_value) | Phương thức gọi đến API nạp thẻ cào của thệ thống Napthengay.com |
| payment(int iduser, String param ) | Phương thức POST xử lí yêu cầu nạp thẻ cào của người dùng, nhận đầu vào là một *@PathVariable int iduser* (iduser của tài khoản thực hiện yêu cầu) và một *@RequesrParam String param*(chuỗi JSON lưu thông tin thẻ cào) |

*Bảng 29: Bảng mô tả lớp UserController-ca sử dụng Nạp thẻ cào*

### Đẩy tin đăng

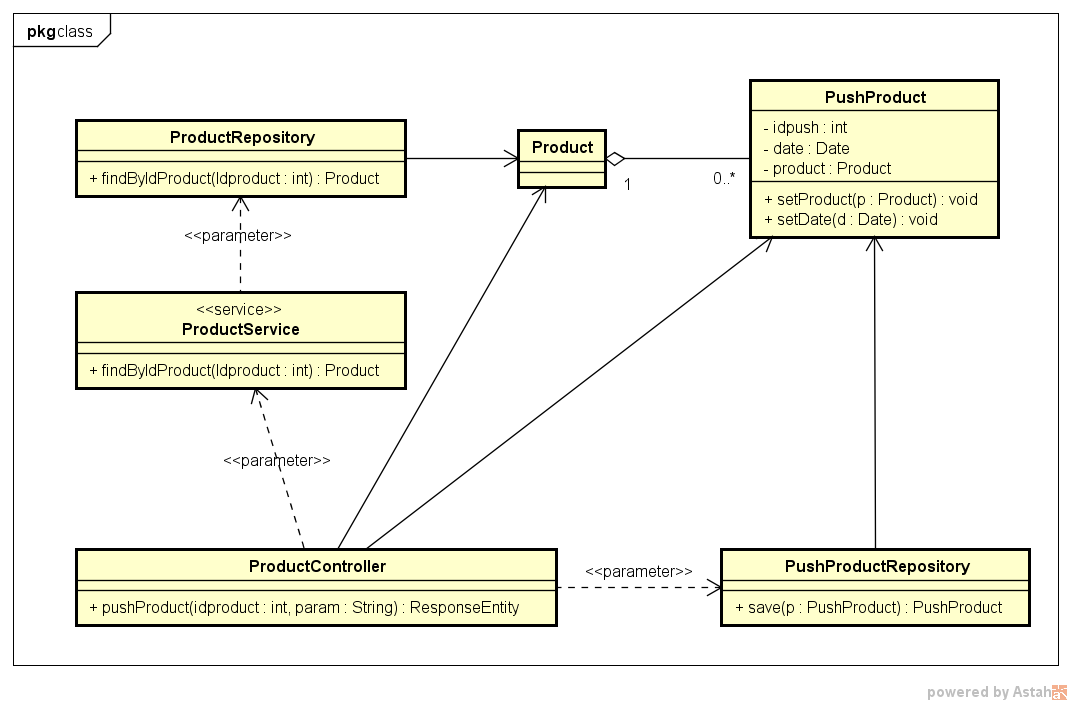
#### *Biểu đồ trình tự*



*Biểu đồ 13:Biểu đồ trình tự Đẩy tin*

#### Biểu đồ lớp

##### Biểu đồ



*Biểu đồ 14: Biểu đồ lớp Đẩy tin*

##### Mô tả lớp

|  |  |
| --- | --- |
| **PushProduct**  Đối tượng lưu thông tin tài khoản người dùng | |
| **Thuộc tính** | **Mô tả** |
| idpush | Khóa chính |
| date | Ngày |
| product | Đối tượng tin đăng |

*Bảng 30: Bảng mô tả lớp PushProduct-ca sử dụng Đẩy tin*

|  |  |
| --- | --- |
| **PushProductRepository**  Interface hibernate xử lí các tác vụ liên quan đến đối tượng PushProduct | |
| **Hàm** | **Mô tả** |
| save(PushProduct  p) | Lưu đối tượng PushProduct vào database |

*Bảng 31: Bảng mô tả lớp PushProductRepository-ca sử dụng Đẩy tin*

|  |  |
| --- | --- |
| **ProductController**  Lớp Controller xử lí các yêu cầu từ người dùng liên quan đến đối tượng Product | |
| **Hàm** | **Mô tả** |
| pushProduct(int idproduct, String param ) | Phương thức POST xử lí yêu cầu nạp thẻ cào của người dùng, nhận đầu vào là một *@PathVariable int idproduct* (iduser của tin đăng muốn đẩy) và một *@RequesrParam String param*(chuỗi JSON lưu danh sách thông tin thời gian để đẩy tin đăng) |

*Bảng 32: Bảng mô tả lớp ProductController- ca sử dụng Đẩy tin*

### Phản ánh tin đăng

### Theo dõi thống kê tin đăng

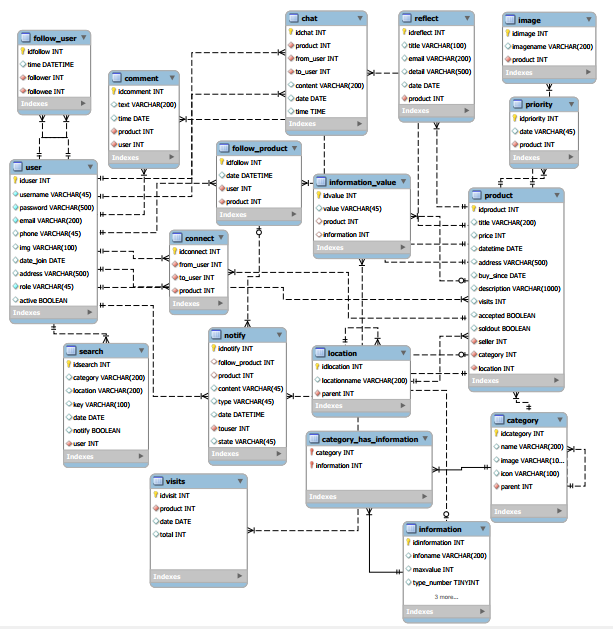
### Kiểm duyệt tin đăng

### 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

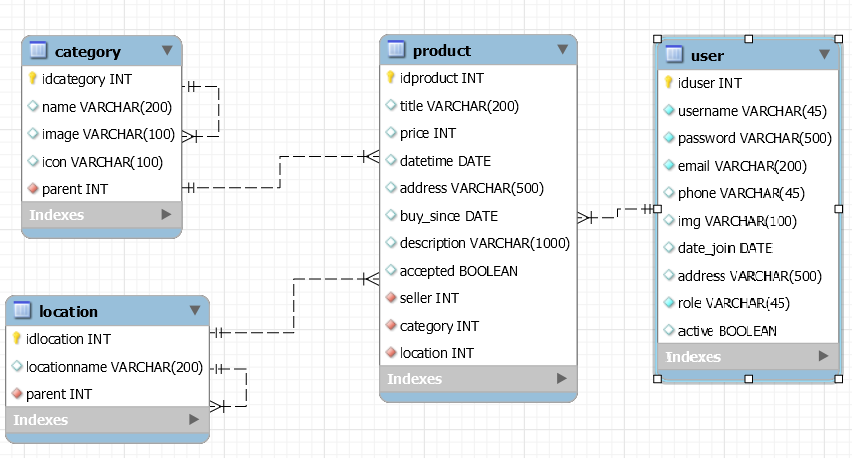
Bảng User lưu trữ thông tin tài khoản của người dùng trong đó có 2 trường là “username” và “password” dùng để đăng nhập vào hệ thống.

|  |  |  |
| --- | --- | --- |
| **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 33: Bảng cơ sở dữ liệu User*

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

Bảng Product lưu trữ thông tin của tin đăng, gồm một số trường thuộc tính bắt buộc như tiêu đề tin đăng (title), giá muốn bán (price), người bán (seller), thể loại (category), địa điểm muốn giao hàng (location).

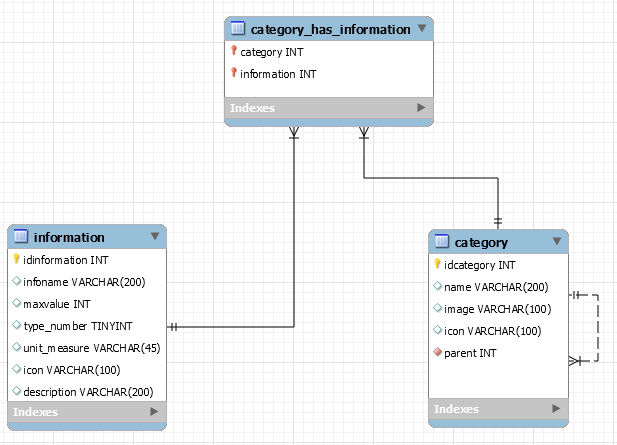


*Hình 22: Quan hệ giữa bảng Product với Category và Location*

|  |  |  |
| --- | --- | --- |
| **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 34: 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 35: Bảng cơ cở dữ liệu Category*

#### Bảng cơ sở dữ liệu Information

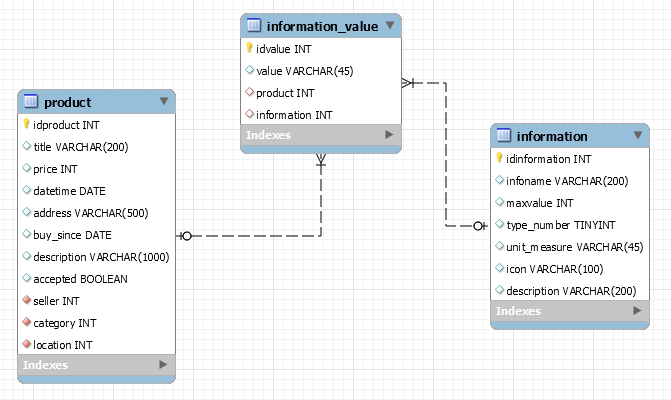
Mỗi thể loại (Category) sẽ có nhiều thuộc tính khác nhau và có thể được cập nhật trong tương lai, không thể biểu diễn hết tất cả thuộc tính trong bảng Category , vì vậy bảng Information sẽ mô tả thuộc tính của Category, quan hệ giữa cả hai là nhiều nhiều và được thể hiện qua bảng Category\_has\_Information

|  |  |  |
| --- | --- | --- |
| **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 36: Bảng cơ sở dữ liệu Information*

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

Bảng Information\_value lưu trữ giá trị thuộc tính của tin đăng (Product), một tin đăng sẽ thuộc một thể loại 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*

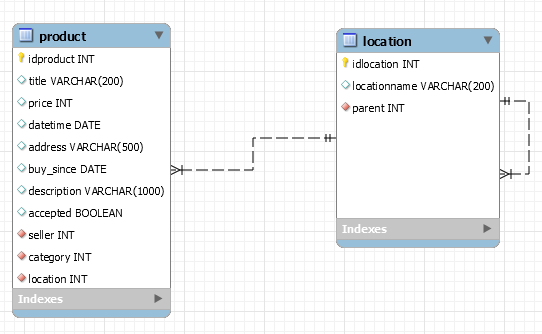
|  |  |  |
| --- | --- | --- |
| **Information\_value** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idvalue | int | Khóa chính |
| value | Varchar(500) | Giá trị thuộc tính |
| product | Integer | Khóa ngoài liên kết tới bảng Product |
| information | Integer | Khóa ngoài liên kết tới bảng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

*Bảng 37: 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 Thành Phố/Tỉnh nên bảng Location có thêm trường thuộc tính “parent”.

Vì vậy “parent” là khóa ngoài liên kết tới bảng Location, biểu diễn quan hệ giữa Quận/Huyện và Thành Phố/Tỉnh



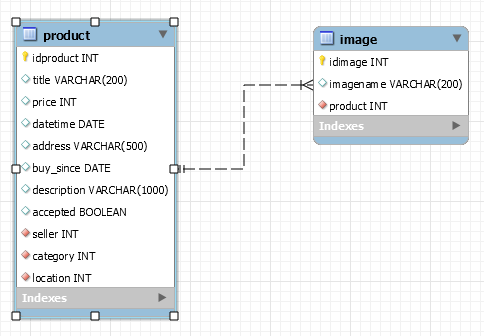
*Hình 25: Quan hệ giữa Product và Location*

|  |  |  |
| --- | --- | --- |
| **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 38: Bảng cơ sở dữ liệu Location*

#### Bảng cơ sở dữ liệu Image

Bảng Image lưu trữ thông tin ảnh của tin đăng (Product). Một tin đăng có thể có nhiều ảnh, nên liên kết giữ Product và Image là một-nhiều



*Hình 26: Quan hệ giữa Product và Image*

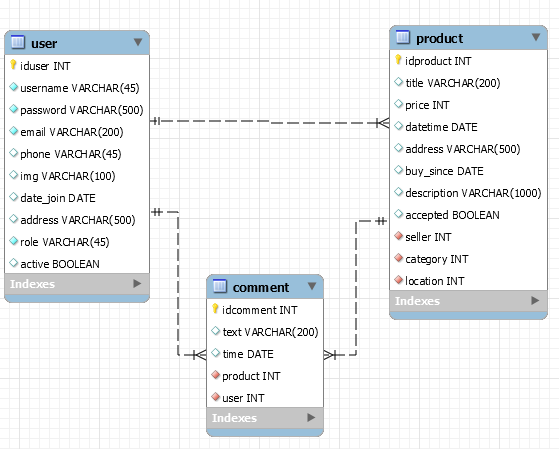
|  |  |  |
| --- | --- | --- |
| **Image** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idimage | Integer | Khóa chính |
| imagename | Varchar(200) | Tên ảnh |
| product | Int | Khóa ngoài liên kết tới bảng Product |

*Bảng 39: Bảng cơ sở dữ liệu Image*

#### Bảng cơ sở dữ liệu Comment

Bảng Comment lưu trữ thông tin bình luận của người dùng (User) về tin đăng (Product) gồm 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*

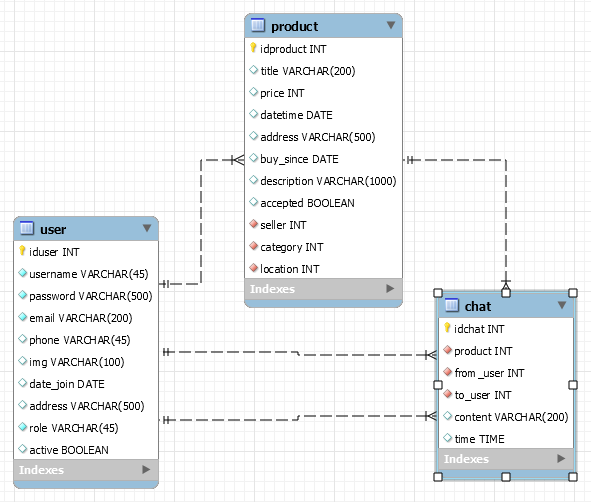
|  |  |  |
| --- | --- | --- |
| **Comment** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idcomment | Integer | Khóa chính |
| text | Varchar(500) | Nội dung bình luận |
| time | DATETIME | Thời gian |
| product | Int | Khóa ngoài liên kết tới bảng Product |
| user | Int | Khóa ngoài liên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

*Bảng 40: Bảng cơ sở dữ liệu Comment*

#### Bảng cơ sở dữ liệu Chat

Bảng Chat lưu trữ nội dung tin nhắn liên quan đến một tin đăng giữa hai người dùng( là người gửi “from\_user” và người nhận “to\_user”), gồm một số trường thuộc tính như nội dung tin nhắn ( content) , thời gian (time), tin đăng (product), người gửi (from\_user), người nhận (to\_user).

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



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

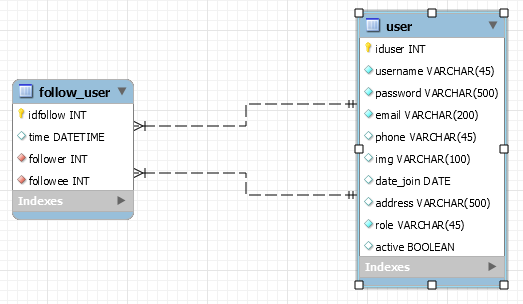
|  |  |  |
| --- | --- | --- |
| **Chat** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idcomment | Integer | Khóa chính |
| content | Varchar(500) | Nội dung tin nhắn |
| time | DATETIME | Thời gian |
| product | Int | Khóa ngoài liên kết tới bảng Product |
| from\_user | Int | Khóa ngoài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

*Bảng 41: Bảng cơ sở dữ liệu Chat*

#### Bảng cơ sở dữ liệu Follow\_User

Bảng Follow\_User lưu trữ thông tin theo dõi giữa người dùng với nhau, gồm một số thuộc tính như thời gian (time), người theo dõi (follower) và người được theo dõi (followee)

Liên kết giữa User và Follow\_User trên khóa ngoài “follower” là một-nhiều, liên kết giữa User và Follow\_User trên khóa ngoài “followee” là một-nhiều



*Hình 29: Quan hệ giữa User và Follow\_user*

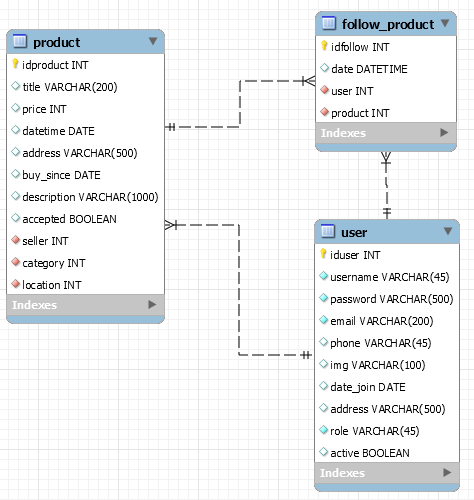
|  |  |  |
| --- | --- | --- |
| **Follow\_User** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idfollow | Integer | Khóa chính |
| time | DATETIME | Thời gian |
| follower | Int | Khóa ngoài liên kết tới bảng User, thông tin người theo dõi |
| followee | Int | Khóa ngoài liên kết tới bảng User, thông tin người được theo dõi |

*Bảng 42: Bảng cơ sở dữ liệu Follow\_User*

#### Bảng cơ sở dữ liệu Follow\_Product

Bảng Follow\_Product lưu trữ thông tin theo dõi tin đăng (Product) của người dùng (User)

Liên kết giữa User và Follow\_Product trên khóa ngoài “user” là một-nhiều, liên kết giữa User và Follow\_Product trên khóa ngoài “product” là một-nhiều



*Hình 30: Quan hệ giữa Follow\_Product với Product và User*

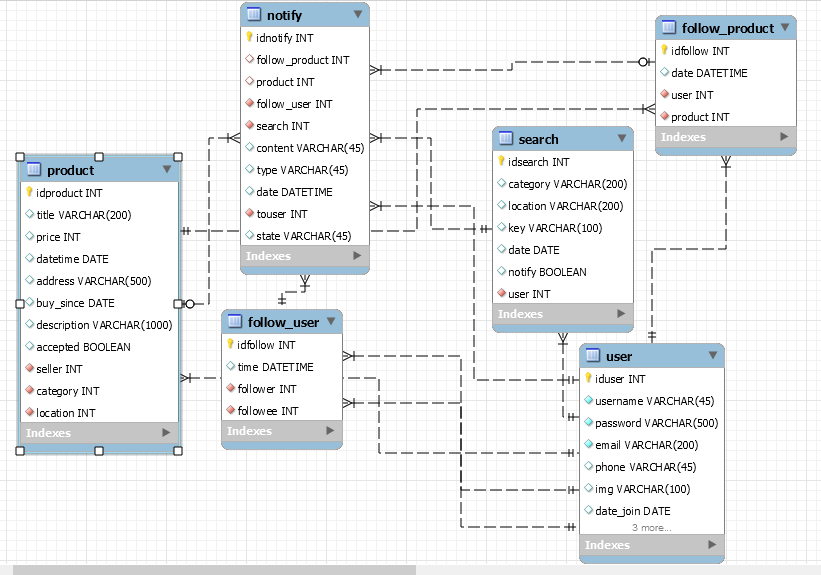
|  |  |  |
| --- | --- | --- |
| **Follow\_Product** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idfollow | Integer | Khóa chính |
| date | DATETIME | Thời gian |
| user | Int | Khóa ngoài liên kết tới bảng User, thông tin người theo dõi |
| product | Int | Khóa ngoài liên kết tới bảng product,  thông tin tin đăng được theo dõi |

*Bảng 43: Bảng cơ sở dữ liệu Follow\_Product*

#### Bảng cơ sở dữ liệu Notify

Bảng Notify lưu trữ nội dung thông báo đến người dùng

Liên kết giữa User và Notify trên khóa ngoài “touser” là một-nhiều, liên kết giữa Product và Notify trên khóa ngoài “product” là một-nhiều, liên kết giữa Follow\_Product và Notify trên khóa ngoài “follow\_product” là một-nhiều, liên kết giữa Follow\_User và Notify trên khóa ngoài “follow\_user” là một-nhiều, liên kết giữa Search và Notify trên khóa ngoài “search” là một-nhiều



*Hình 31: Quan hệ giữa Notify với các bảng khác*

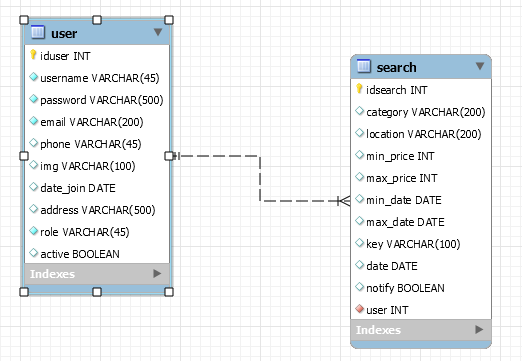
|  |  |  |
| --- | --- | --- |
| **Notify** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idnotify | Integer | Khóa chính |
| follow\_product | Integer | Là khóa ngoài liên kết tới bảng Follow\_Product |
| product | Integer | Khóa ngoài liên kết tới bảng product,  thông tin tin đăng được theo dõi |
| follow\_user | Integer | Là khóa ngoài liên kết tới bảng Follow\_User |
| search | Integer | Là khóa ngoài liên kết tới bảng Search |
| content | Varchar(100) | Nội dung thông báo |
| type | Integer | Kiểu thông báo |
| date | DATETIME | Thời gian theo ngày và giờ |
| touser | Integer | Là khóa ngoài liên kết tới bảng User, thông tin người nhận thông báo |
| state | Boolean | Trạng thái thông báo đã được người dùng đọc hay chưa |

*Bảng 44: Bảng cơ sở dữ liệu Notify*

#### Bảng cơ sở dữ liệu Search

Bảng Search lưu trữ nội dung tìm kiếm người dùng lưu

Liên kết giữa User và Search trên khóa ngoài “user” là một-nhiều



*Hình 32: Quan hệ giữa Search và User*

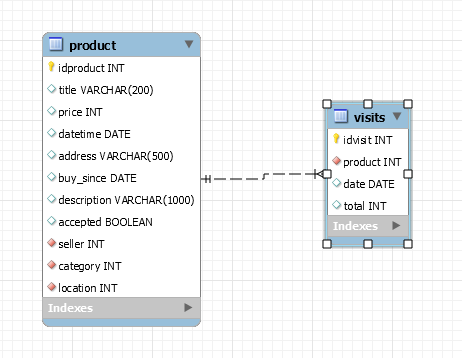
|  |  |  |
| --- | --- | --- |
| **Search** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idsearch | Integer | Khóa chính |
| category | Varchar(100) | Tên thể loại |
| location | Inte Varchar(100)ger | Tên vị trí |
| min\_price | Integer | Giới hạn trên của giá sản phẩm |
| max\_price | Integer | Giới hạn dưới của giá sản phẩm |
| min\_date | Varchar(100) | Giới hạn dưới của ngày đăng |
| max\_date | Integer | Giới hạn trên của ngày đăng |
| key | Varchar(100) | Nội dung tìm kiếm |
| date | DATETIME | Thời gian theo ngày và giờ |
| user | Integer | Là khóa ngoài liên kết tới bảng User |
| notify | Boolean | Nếu người dùng muốn nhận thông báo , trường *notify* có giá trị *true* |

*Bảng 45: Bảng cơ sở dữ liệu Search*

#### Bảng cơ sở dữ liệu Visits

Bảng Visit lưu trữ số liệu lượt truy cập của một tin đăng theo ngày

Liên kết giữa Product và Visít trên khóa ngoài “product” là một-nhiều



*Hình 33: Quan hệ giữa Visits và Product*

|  |  |  |
| --- | --- | --- |
| **Visits** | | |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idvisit | Integer | Khóa chính |
| product | Integer | Khóa ngoài liên kết tới bảng Product |
| date | Date | Thời gian ngày |
| total | Integer | Số lượt truy cập |

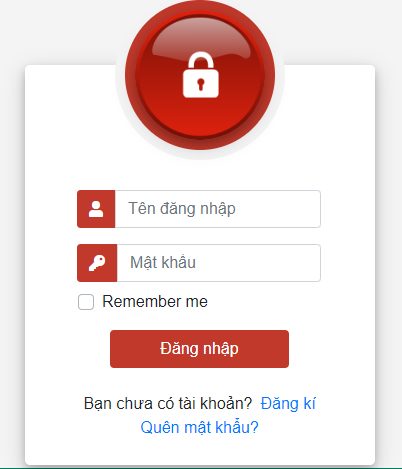
*Bảng 46: Bảng cơ sở dữ liệu Visits*

## Thiết kế giao diện

Trong lúc thiết kế giao diện,em đã dùng một số framwork sau đây:

* Bootstrap: là một framework cho phép thiết kế website reponsive nhanh hơn và dễ dàng hơn. Bootstrap là bao gồm các HTML templates, CSS templates và Javascript tao ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác.
* Angular Material: Materials cung cấp sẵn rất nhiều built-in module khi xây dựng ứng dụng Angular. Cung cấp nhiều tính năng như autocomplete, datepicker, slider, menu, grid và các toolbar và nhiều tiện ích khác giúp cho việc xây dựng website một cách nhanh chóng
* Ngoài ra ứng dụng của em sử dụng một số built-in module có sẵn trong Angular như :
* NgImageSliderModule: Một module thiết kế thanh trượt hình ảnh hỗ trợ reponsive. Cũng hỗ trợ các video youtube và mp4.
* NgxInfiniteScrollerModule : Một module hỗ trợ làm việc với thanh cuộn
* ChartsModule: Một module hỗ trợ hiển thị biểu đồ reactive, responsive của Angular dựa trên Chart.js

### Giao diện Đăng nhập

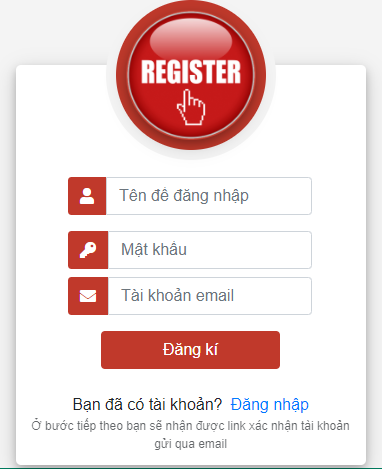


*Hình 34: Giao diện Đăng nhập*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả giao diện Đăng nhập** | | | | |
| **Nội dung hiển thị** | **Kiểu** | **Editable** | **Giá trị mặc định** | **Mô tả** |
| **Hiển thị mặc định** | | | | |
| Tên đăng nhập | Input text | true | N/A | Nhập tên tài khoản đăng nhập |
| Mật khẩu | Input password | true | N/A | Nhập mật khẩu tài khoản |
| Đăng nhập | Button | no | N/A | Người dùng click gửi thông tin đăng nhập |
| Đăng kí | Link | no | Đường dẫn đến trang đăng kí | Điều hướng tới trang đăng kí |
| Quên mật khẩu | Link | no | Đường dẫn đến trang thay đổi mật khẩu | Điều hướng tới trang thay đổi mật khẩu |

*Bảng 47: Bảng mô tả giao diện Đăng nhập*

### Giao diện Đăng kí



*Hình 35: Giao diện Đăng kí*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mô tả giao diện Đăng nhập** | | | | | |
| **Nội dung hiển thị** | **Kiểu** | | **Editable** | **Giá trị mặc định** | **Mô tả** |
| **Hiển thị mặc định** | | | | | |
| Tên để đăng nhập | | Input text | true | N/A | Nhập tên tài khoản |
| Mật khẩu | | Input password | true | N/A | Nhập mật khẩu |
| Tài khoản email | | Input text | true | N/A | Nhập thông tin tài khoản email |
| Đăng kí | | Button | no | N/A | Người dùng click gửi thông tin đăng kí |
| Đăng nhập | | Link | no | Đường dẫn đến trang đăng nhập | Điều hướng tới trang đăng nhập |

*Bảng 48: Bảng mô tả giao diện Đăng kí*

### Giao diện Trang chủ

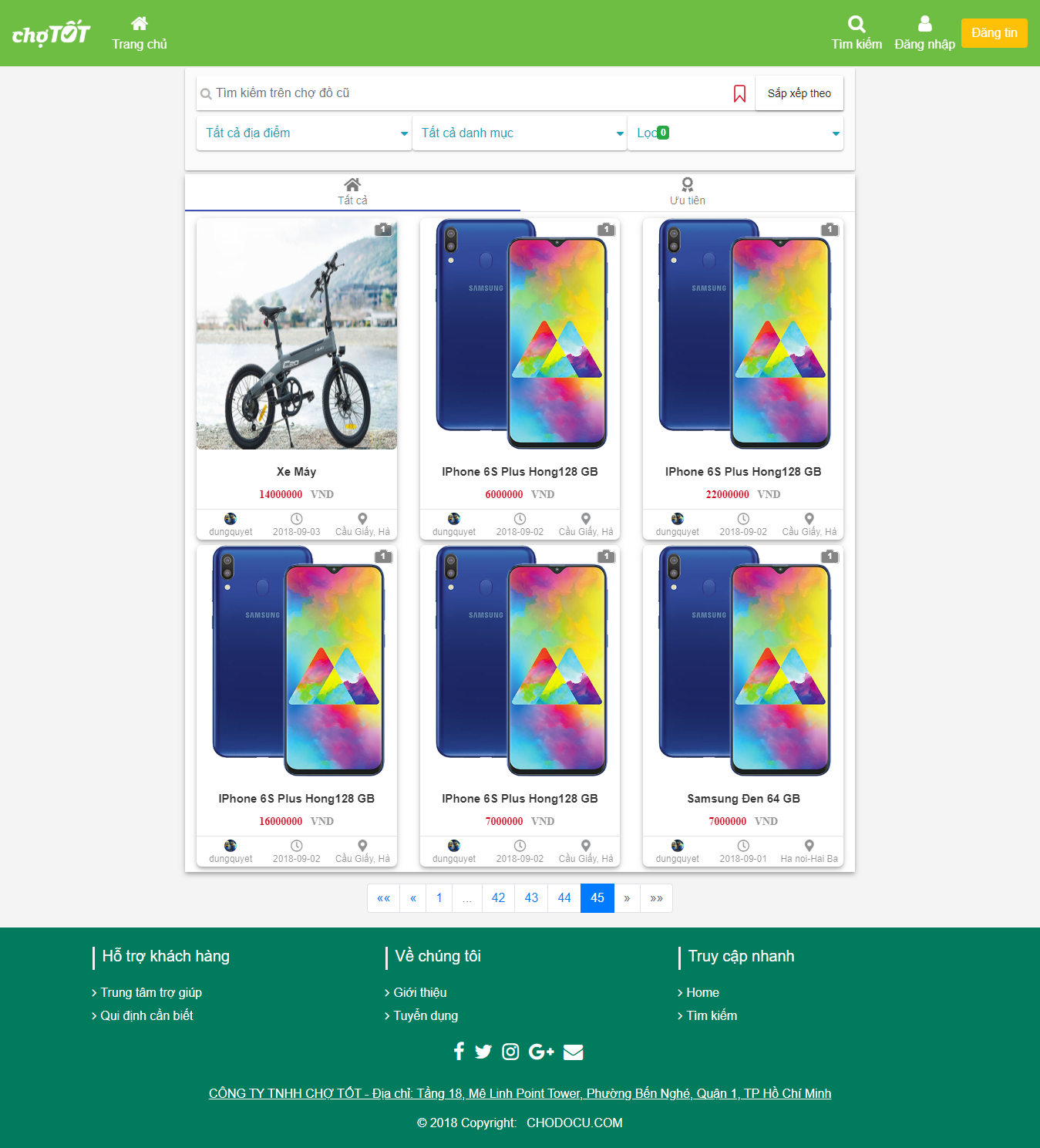


*Hình 36: Giao diện Trang chủ*

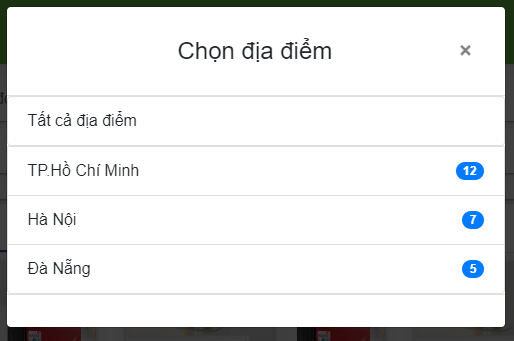
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mô tả giao diện Trang chủ** | | | | | |
| **Nội dung hiển thị** | **Kiểu** | | **Editable** | **Giá trị mặc định** | **Mô tả** |
| **Hiển thị mặc định** | | | | | |
| Trang chủ | | Link | no | Đường dẫn tới trang chủ | Điều hướng tới trang chủ |
| Tìm kiếm | | Link | no | Đường dẫn tới trang tìm kiếm | Điều hướng tới trang tìm kiếm |
| Đăng nhập | | Link | no | Đường dẫn tới trang đăng nhập | Điều hướng tới trang đăng nhập |
| Đăng tin | | Button | no | Đường dẫn tới trang đăng tin | Điều hướng tới trang đăng tin |

*Bảng 49: Bảng mô tả giao diện Trang chủ*

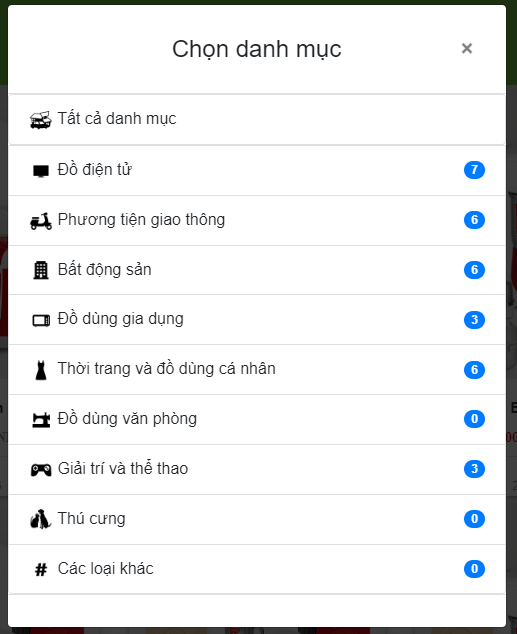
### Giao diện Tìm kiếm



*Hình 37: Giao diện Tìm kiếm*



*Hình 38: Giao diện modal Chọn địa điểm*

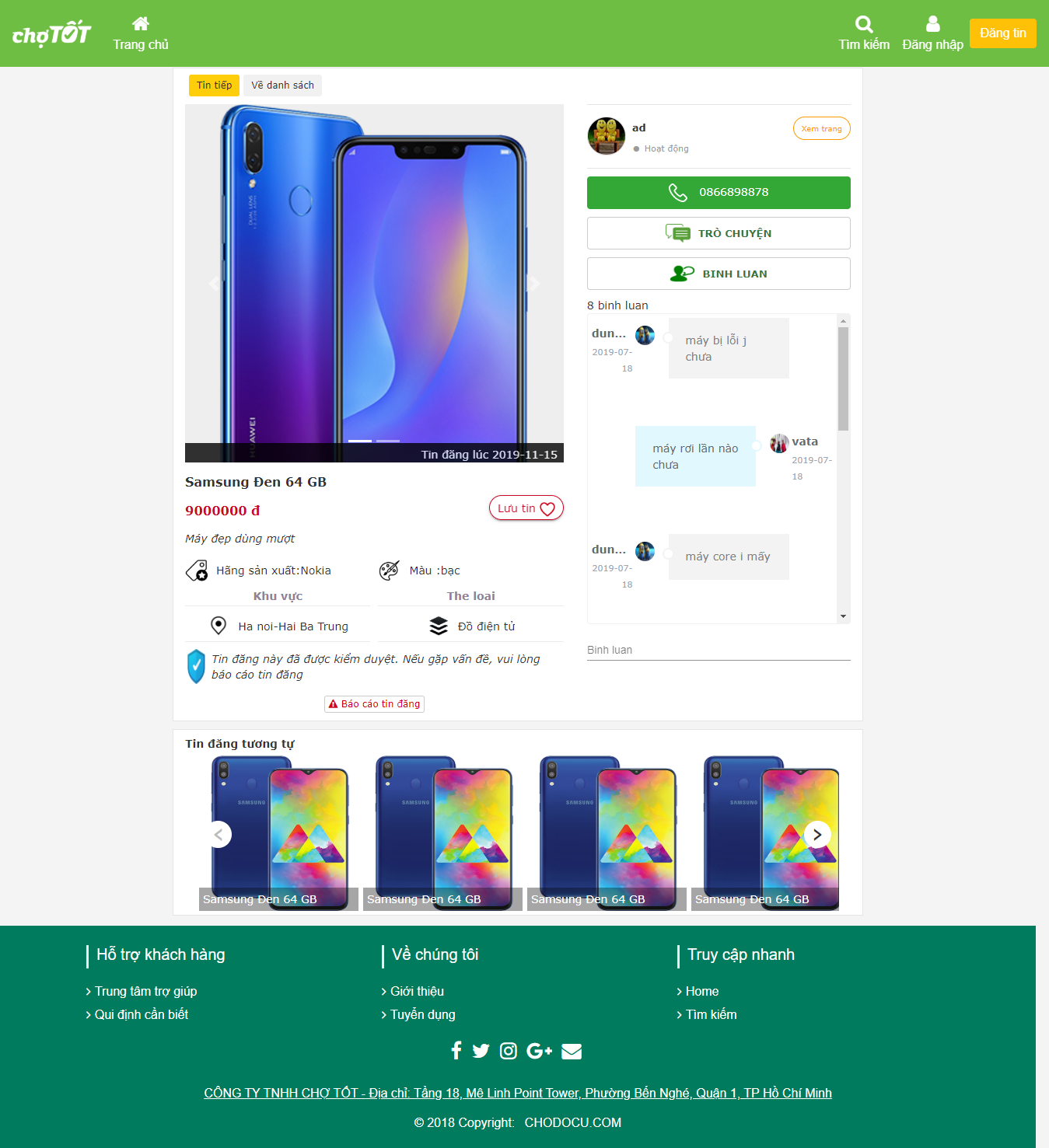


*Hình 39: Giao diện modal Chọn danh mục*

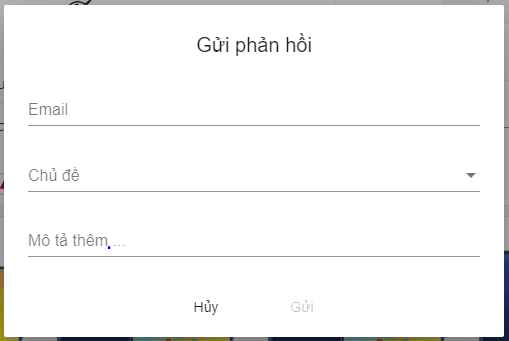
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mô tả giao diện Tìm kiếm** | | | | | |
| **Nội dung hiển thị** | **Kiểu** | | **Editable** | **Giá trị mặc định** | **Mô tả** |
| **Hiển thị mặc định** | | | | | |
| Tìm kiếm trên chợ đồ cũ | | Input text | yes | N/A | Nhập nội dung tìm kiếm |
|  | | Button | no | N/A | Thực hiện chức năng lưu tìm kiếm |
| Sắp xếp theo | | Mat\_menu (Angular Material) | no | N/A | Sắp xếp giảm dần,tăng dần theo giá hoặc thời gian đăng |
| Tất cả địa điểm | | Button | no | N/A | Click để hiển thị Modal chọn địa điểm |
| Tất cả danh mục | | Button | no | N/A | Click để hiển thị Modal chọn danh mục |
| Lọc | | Button | no | N/A | Click để hiển thị Modal lọc |
| Tất cả | | Mat-tab  (Angular Material) | no | N/A | Tab hiển thị tất cả tin đăng được chấp thuận |
| Ưu tiên | | Mat-tab  (Angular Material) | no | N/A | Tab hiển thị tất cả tin đăng ưu tiên |
|  | | Ngb-pagination  (thẻ phân trang trong Angular) | no | 1 | Hiển thị danh sách phân trang |
| **Modal “chọn địa điêm”**  Hiển thị danh sách Thành phố/Tỉnh và Quận/Huyện | | | | | |
|  | | Badge (Bootstrap) | no | N/A | Hiển thị tổng số Quận/Huyện của một Thành Phố/Tỉnh  Click vào để hiển thị danh sách Quận/Huyện |
| **Modal “chọn danh mục”**  Hiển thị danh sách thể loại | | | | | |
|  | | Badge (Bootstrap) | no | N/A | Hiển thị tổng số thể loại con của một thể loại  Click vào để hiển thị danh sách thể loại con |

*Bảng 50: Bảng mô tả giao diện Tìm kiếm*

### Giao diện Chi tiết tin đăng



*Hình 40: Giao diện Chi tiết tin đăng*

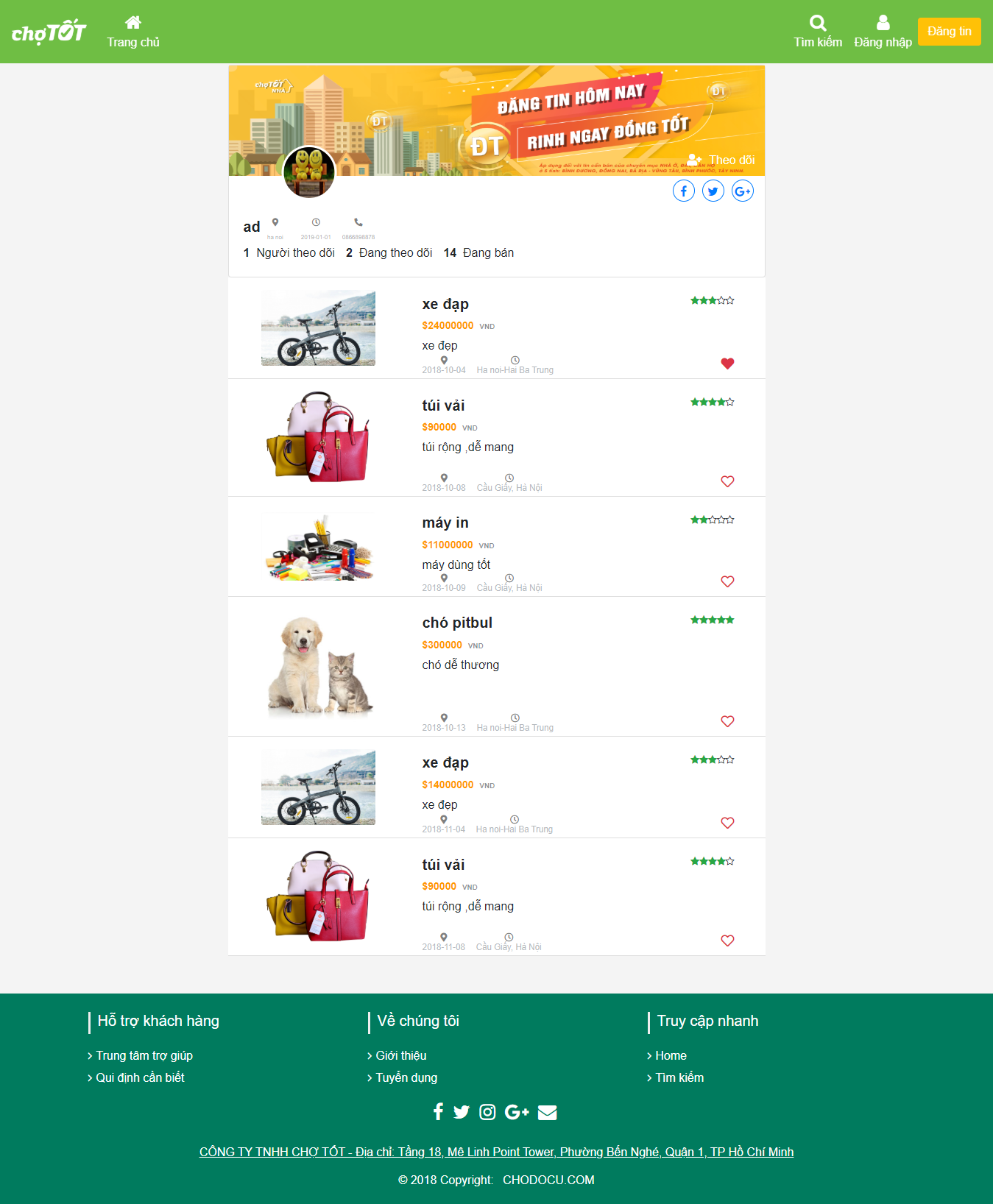


*Hình 41: Giao diện modal Gửi phản hồi*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mô tả giao diện Chi tiết tin đăng** | | | | | | | | |
| **Nội dung hiển thị** | **Kiểu** | | | **Editable** | | **Giá trị mặc định** | **Mô tả** | |
| **Hiển thị mặc định** | | | | | | | | |
| Tin tiếp | Button | | | no | | N/A | Click để hiển thị tin đăng tiếp theo thuộc danh sách tìm kiếm | |
| Về danh sách | Button | | | no | | N/A | Click vào để về giao diện trang cá nhân người dùng sở hữu tin đăng | |
| Xem trang | Button | | | no | | N/A | Sắp xếp giảm dần,tăng dần theo giá hoặc thời gian đăng | |
|  | Img | | | no | | N/A | Hình ảnh tài khoản sở hữu tin đăng | |
|  | Button | | | no | | N/A | Hiển thị số điện thoại liên lạc của người sở hữu tin đăng | |
| Trò chuyện | Button | | | no | | Đường dẫn trang nhắn tin | Click để đến giao diện nhắn tin | |
| Bình luận | Button | | | no | | N/A | Click để hiển thị hoặc ẩn nội dung bình luận | |
| Lưu tin | Button | | | no | | N/A | Click để theo dõi tin đăng | |
| Báo cáo tin đăng | Button | | | no | | N/A | Click để hiển thị modal gửi phản hồi tin đăng | |
| Tin đăng tương tự | ng-image-slider  (module thuộc Angular) | | | no | | N/A | Hiển thị danh sách tin đăng cùng thể loại với tin đăng tại màn hình “Chi tiết tin đăng) | |
| **Modal “gửi phản hồi”**  Hiển thị form để nhập nội dung phản hồi | | | | | | | | |
| Email | | Tài khoản email người dùng | no | | N/A | | | Nhập tài khoản email |
| Chủ đề | | Mat-select  (thẻ thuộc Angular Material) | yes | | N/A | | | Clickchojn lí do phản hồi tin đăng |
| Mô tả thêm | | Textarea | yes | | N/A | | | Mô tả thêm về tin đăng muốn phản hồi |
| Hủy | | Button | no | | N/A | | | Click để hủy thao tác |
| Gửi | | Button | no | | N/A | | | Click để gửi phản hồi về hệ thống |

*Bảng 51: Bảng mô tả giao diện Chi tiết tin đăng*

### Giao diện Trang cá nhân

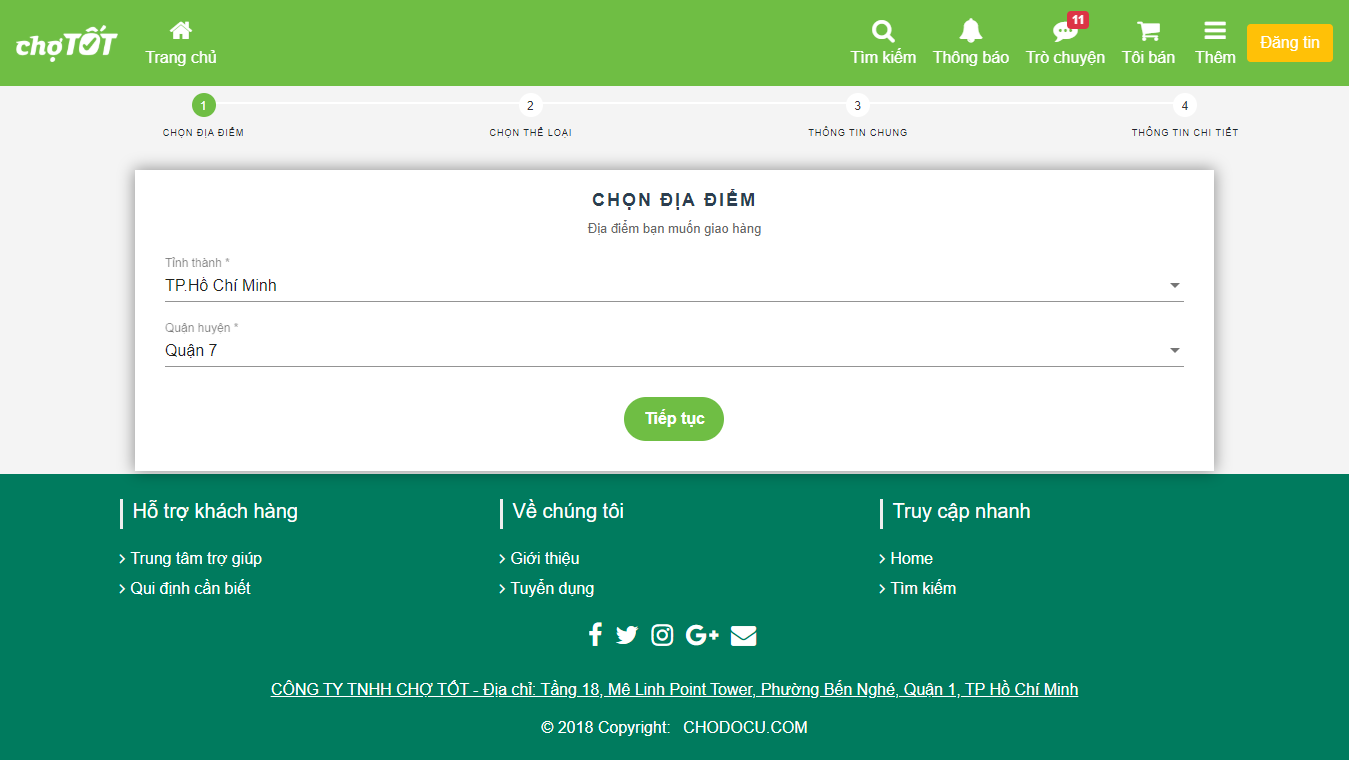


*Hình 42: Giao diện Trang cá nhân*

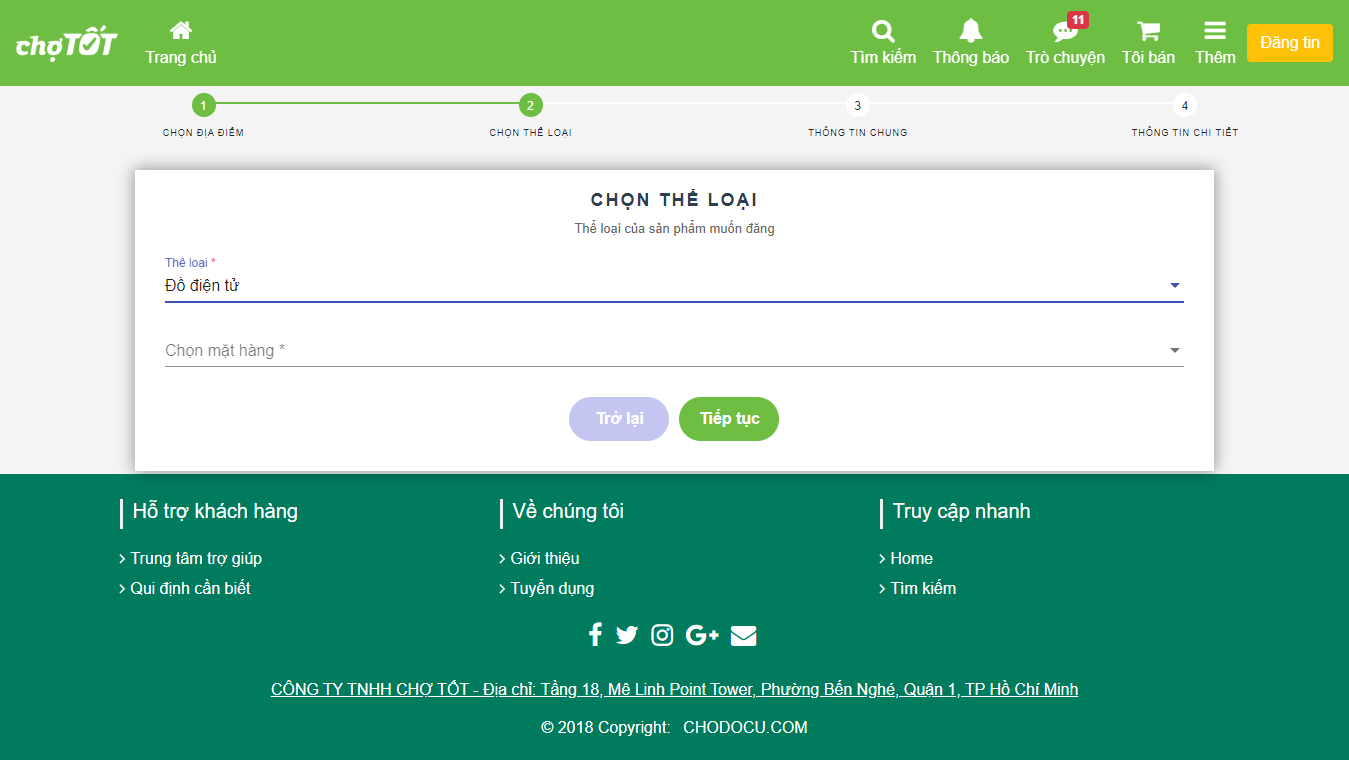
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả giao diện Trang cá nhân** | | | | |
| **Nội dung hiển thị** | **Kiểu** | **Editable** | **Giá trị mặc định** | **Mô tả** |
| **Hiển thị mặc định**  Danh sách tin đăng đã được chấp thuận của người dùng | | | | |
| Theo dõi | Button | no | N/A | Click để theo dõi tài khoản |
| Người theo dõi | Label | no | 0 | Số lượng người dùng theo dõi tài khoản |
| Đang theo dõi | Label | no | 0 | Số lượng người dùng mà tài khoản hiện tại đang theo dõi |
| Đang bán | Label | no | 0 | Số lượng tin đăng được chấp thuận |
|  | Icon | no | N/A | Click để theo dõi tin đăng |

*Bảng 52: Bảng mô tả giao diện Trang cá nhân*

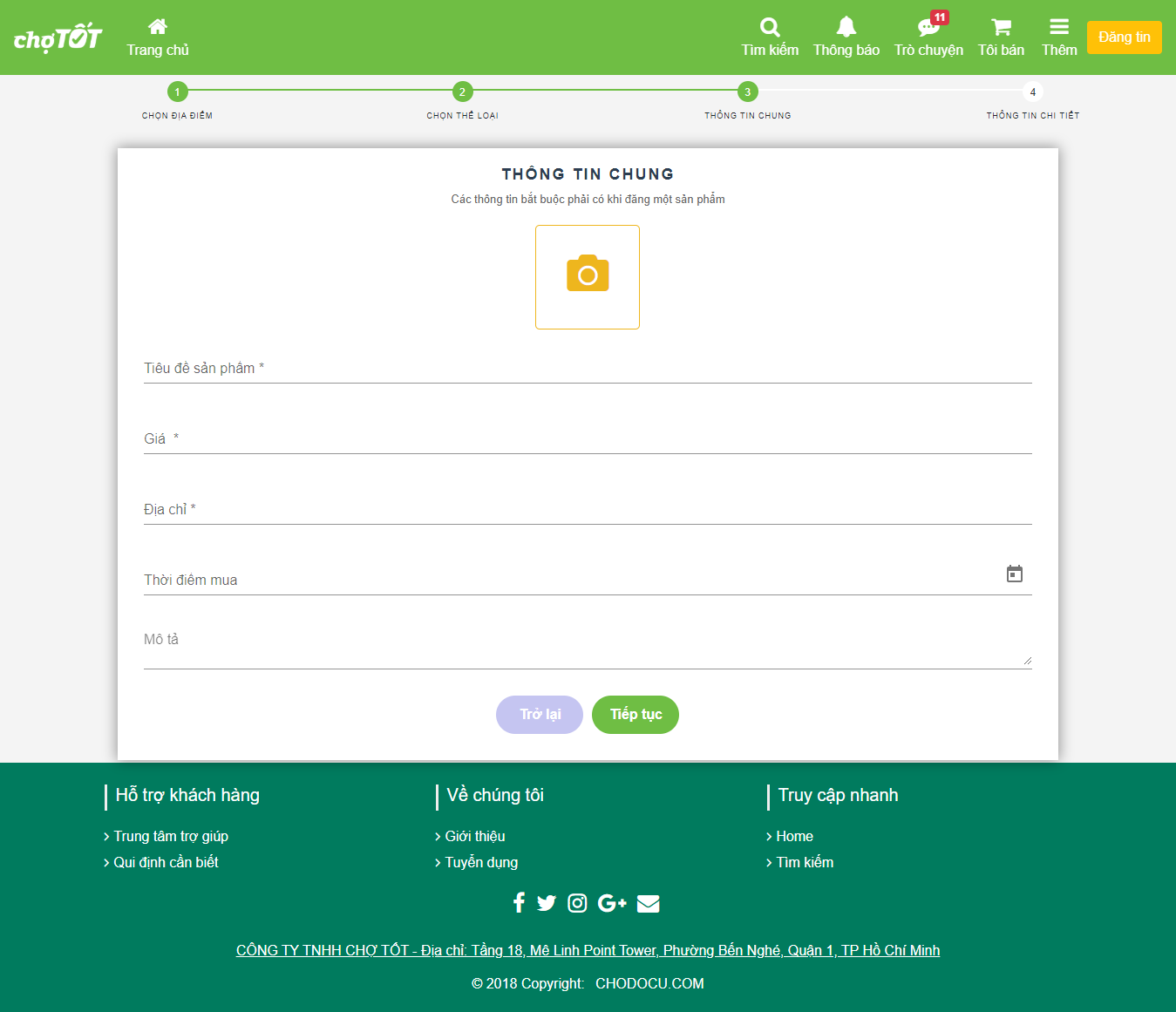
### Giao diện Đăng tin



*Hình 43: Giao diện Chọn địa điểm để đăng tin*



*Hình 44: Giao diện Chọn thể loại để đăng tin*



*Hình 45: Giao diện Thông tin chung để đăng tin*

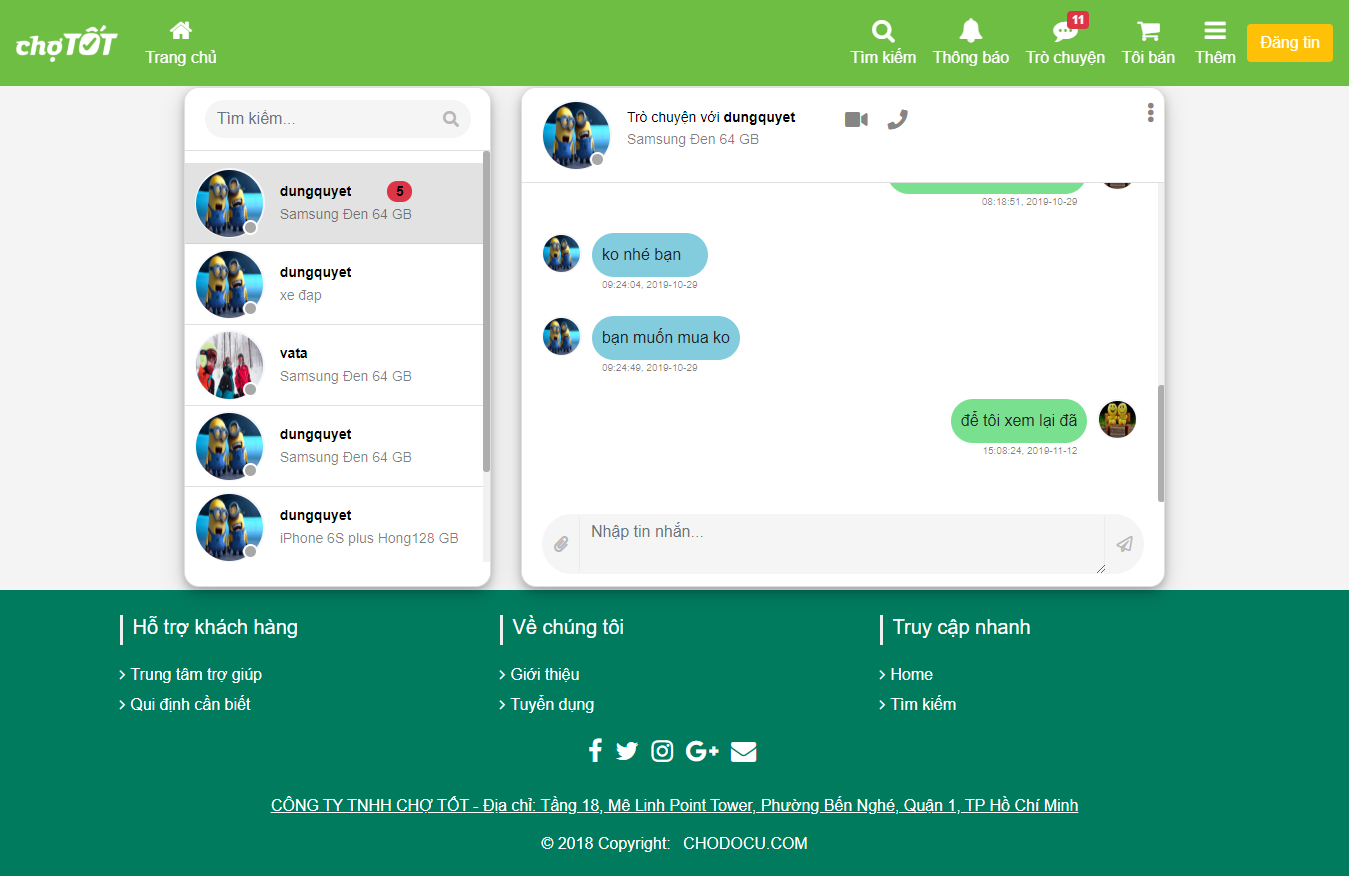


*Hình 46: Giao diện Thông tin chi tiết lúc đăng tin*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả giao diện Đăng tin** | | | | |
| **Nội dung hiển thị** | **Kiểu** | **Editable** | **Giá trị mặc định** | **Mô tả** |
| **Hiển thị mặc định**  Giao diện Chọn địa điểm | | | | |
| Tỉnh thành | Mat-select (Angular Material) | yes | N/A | Chọn Thành Phố/Tỉnh |
| Quận huyện | Mat-select (Angular Material) | yes | N/A | Chọn Quận/Huyện |
| Tiếp tục | Button | no | N/A | Click để chuyển sang giao diện “Chọn thể loại” |
| **Giao diện “Chọn thể loại”** | | | | |
| Chọn thể loại | Mat-select  (Angular Material) | no | N/A | Click để chọ thể loại của tin muốn đăng |
| Chọn mặt hàng | Mat-select  (Angular Material) | no | N/A | Click để chọn loại kiểu mặt hàng muốn đăng |
| Tiếp tục | Button | no | N/A | Click để chuyển sang giao diện “Thông tin chung” |
| Trở lại | Button | no | N/A | Click để trở lại giao diện “Chọn địa điểm” |
| **Giao diện “Thông tin chung”** | | | | |
|  | Icon | no | no | Click để tải ảnh lên |
| Tiêu đề sản phẩm | Input text | yes | N/A | Nhập tên sản phẩm muốn đăng |
| Giá | Input number | yes | N/A | Nhập giá sản phẩm |
| Địa chỉ | Input text |  |  | Nhập địa chỉ chi tiết muốn giao hàng |
| Thời điểm mua | Input date | yes | N/A | Nhập thời điểm mua sản phẩm |
| Mổ tả | Input text | yes | N/A | Mô tả thêm về sản phẩm |
| Tiếp tục | Button | no | N/A | Click để chuyển sang giao diện “Thông tin chi tiết” |
| Trở lại | Button | no | N/A | Click để trở lại giao diện “Chọn thể loại” |
| **Giao diện “Thông tin chi tiết”**  Mỗi thể loại hoặc danh mục sẽ có các thông tin chi tiết riêng | | | | |
| Tạo mới | Button | no | N/A | Click để lưu đăng tin lên hệ thống |
| Trở lại | Button | no | N/A | Click để trở lại giao diện “Thông tin chung” |

*Bảng 53: Bảng mô tả giao diện Đăng tin*

### Giao diện Nhắn tin

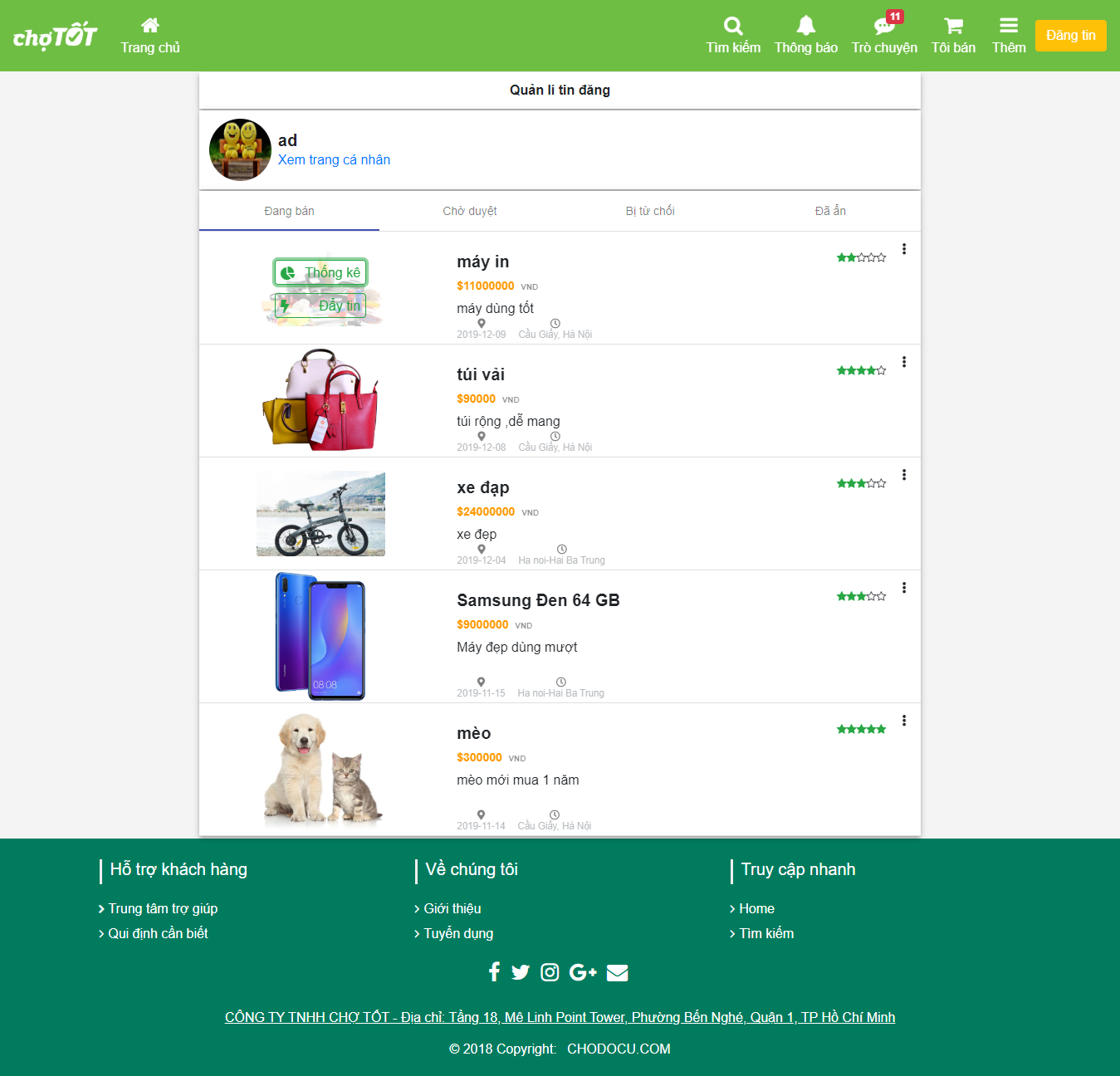


*Hình 47: Giao diện Nhắn tin*

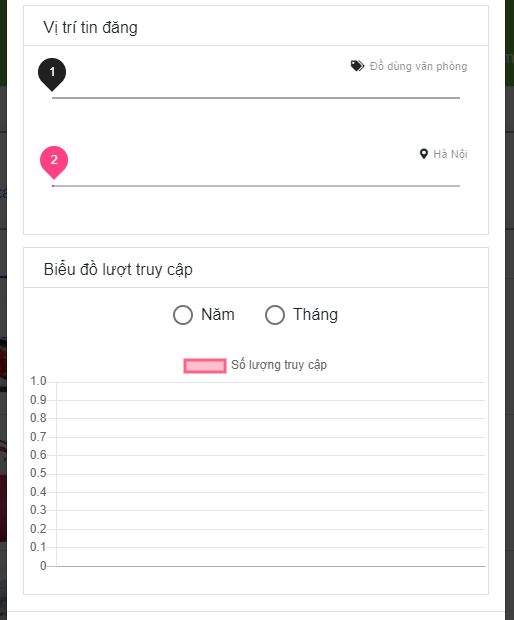
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả giao diện Nhắn tin**  Hiển thị danh sách cuộc trò chuyện và tin nhắn | | | | |
| **Nội dung hiển thị** | **Kiểu** | **Editable** | **Giá trị mặc định** | **Mô tả** |
| **Hiển thị mặc định** | | | | |
| Tìm kiếm | Input text | yes | N/A | Nhập từ khóa bất kì để tìm cuộc trò chuyện |
| Nhập tin nhắn | Input text | yes | N/A | Nhập nội dung tin nhắn |
|  | Icon | no | N/A | Click để gửi tin nhắn |

*Bảng 54: Bảng mô tả giao diện Nhắn tin*

### Giao diện Quản lí tin đăng



*Hình 48: Giao diện Quản lí tin đăng*



*Hình 49: Giao diện Thống kê tin đăng*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả giao diện Quản lí tin đăng** | | | | |
| **Nội dung hiển thị** | **Kiểu** | **Editable** | **Giá trị mặc định** | **Mô tả** |
| **Hiển thị mặc định**  Giao diện Chọn địa điểm | | | | |
| Xem trang cá nhân | Link | no | Đường dẫn tới trang cá nhân của tài khoản đang đăng nhập | Điều hướng tới trang cá nhân |
| Đang bán | Mat-tab (Angular Material) | no | N/A | Click để hiển thị tin đăng được chấp thuận |
| Chờ duyệt | Mat-tab (Angular Material) | no | N/A | Click để hiển thị tin đăng chờ phê duyệt |
| Bị từ chối | Mat-tab (Angular Material) | no | N/A | Click để hiển thị tin đăng bị từ chối |
| Đã ẩn | Mat-tab (Angular Material) | no | N/A | Click để hiển thị tin đăng đã ẩn |
|  | Icon | no | N/A | Click để hiển thị menu “Ẩn tin” và “Sửa tin” |
| Ẩn tin | Button | no | N/A | Click để ẩn tin đăng |
| Sửa tin | Button | no | Đường dẫn đến trang thay đổi tin đăng | Click để đến trang thay đổi tin đăng |
| Thống kê | Button | no | N/A | Click để xem thống kê về tin đăng |
| Đẩy tin | Button | no | Đường dẫn đến trang đẩy tin | Click để đến trang đẩy tin |

*Bảng 55: Bảng mô tả giao diện Quản lí tin đăng*

### Giao diện Đẩy tin

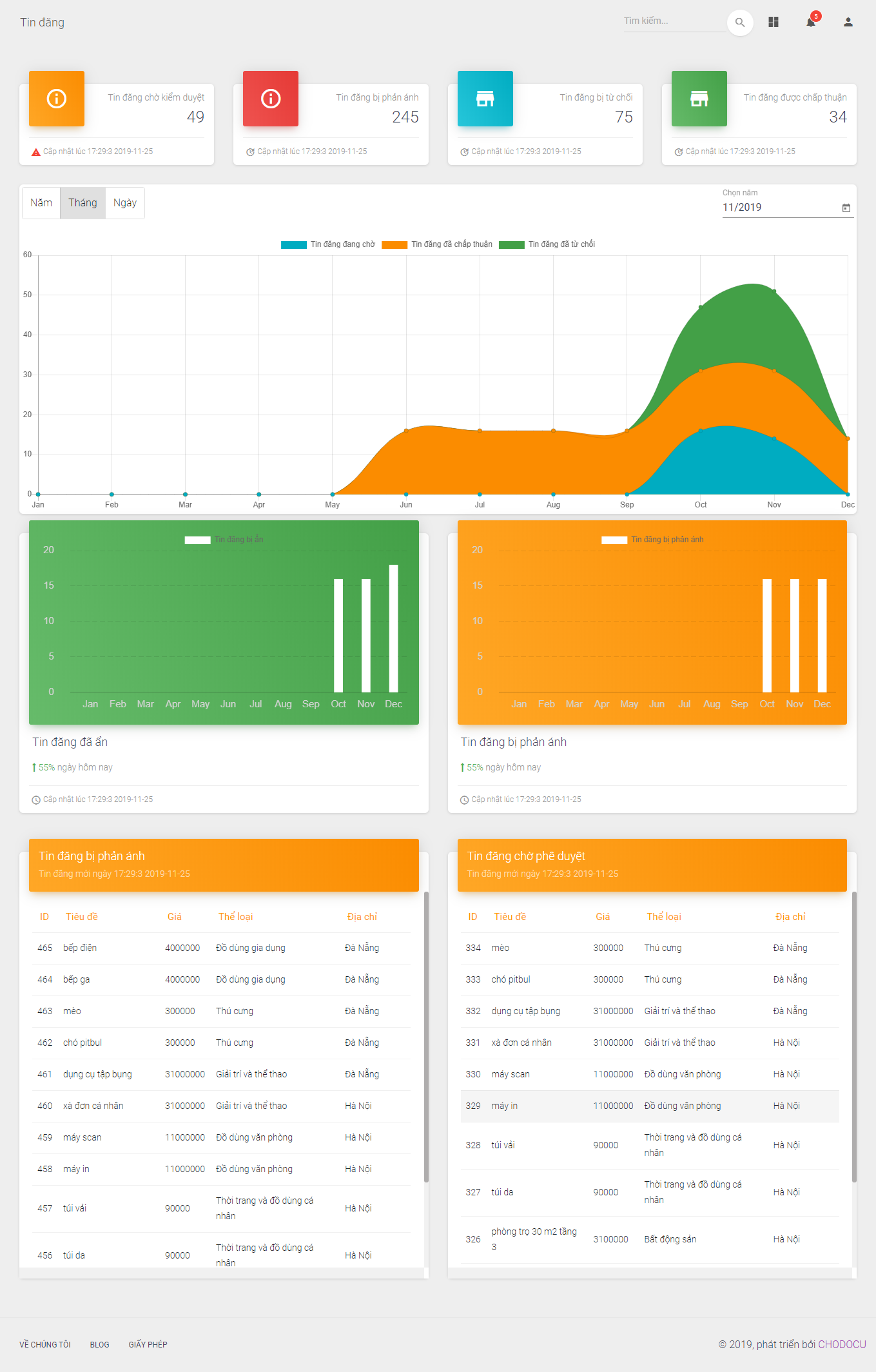


*Hình 50: Giao diện Đẩy tin*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả giao diện Đẩy tin** | | | | |
| **Nội dung hiển thị** | **Kiểu** | **Editable** | **Giá trị mặc định** | **Mô tả** |
| **Hiển thị mặc định** | | | | |
| Trước | Button | no | N/A | Click để chuyển đến tháng trước tháng hiện tại |
| Hôm nay | Button | no | N/A | Click để chuyển đến tháng hiện tại |
| Sau | Button | no | N/A | Click để chuyển đến tháng sau |
| Đẩy tin | Button | no | N/A | Click để thực hiện chức năng đẩy tin |
|  | 1. CalendarMonthView   (Angular  Calendar) | no | N/A | Click để chọn danh sách ngày muốn đẩy tin |

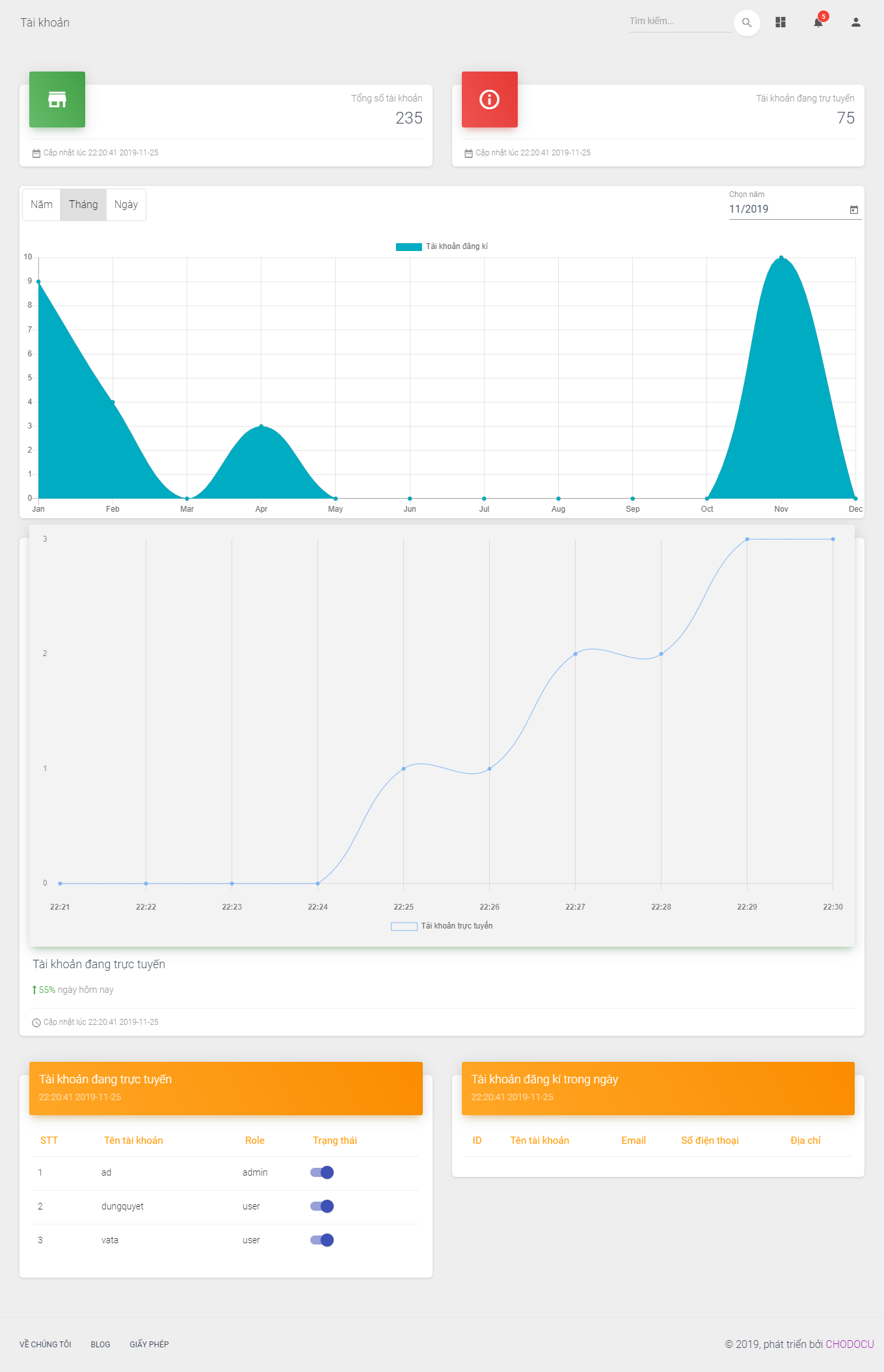
*Bảng 56: Bảng mô tả giao diện Đẩy tin*

### Giao diện Quản trị tin đăng



*Hình 51: Giao diện Quản trị tin đăng*

### Giao diện Quản trị người dùng

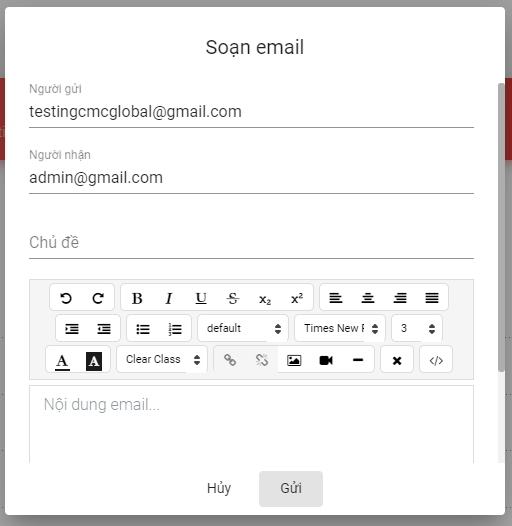


*Hình 52: Giao diện Quản trị người dùng*

### Giao diện Phê duyệt tin đăng



*Hình 53: Giao diện Phê duyệt tin đăng*

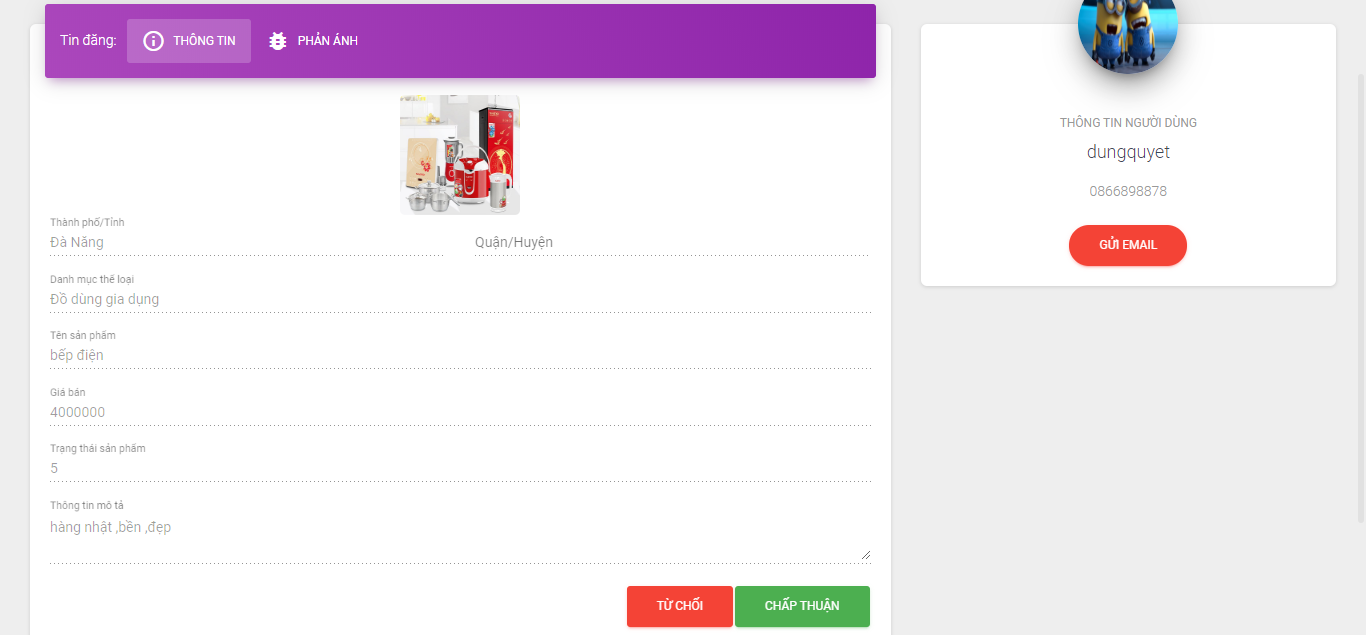


*Hình 54: Giao diện Soạn Email*

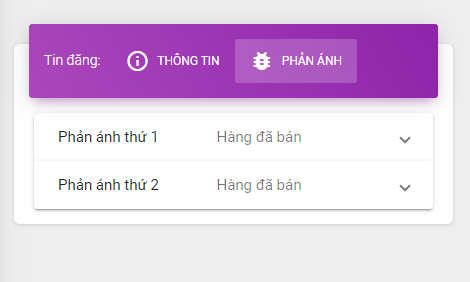
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả giao diện Phê duyệt tin đăng** | | | | |
| **Nội dung hiển thị** | **Kiểu** | **Editable** | **Giá trị mặc định** | **Mô tả** |
| **Hiển thị mặc định**  Thông tin của tin đăng | | | | |
| Từ chối | Button | no | N/A | Click để từ chối tin đăng |
| Chấp thuận | Button | no | N/A | Click để chấp thuận tin đăng |
| Gửi email | Button | no | N/A | Click để hiển thị giao diện Soạn Email |
| **Giao diện Soạn Email** | | | | |
| Người gửi | Input text | yes | N/A | Nhập địa chỉ email người gửi |
| Người nhận | Input text | yes | N/A | Nhập email người gửi |
| Chủ đề | Input text | yes | N/A | Nhập chủ đề |
| Nội dung Email | Angular Editor  (Angular module) | yes | N/A | Nhập nội dung email |
| Hủy | Button | no | N/A | Click hủy thao tái soạn email |
| Gửi | Button | no | N/A | Click để gửi email |

*Bảng 57: Bảng mô tả giao diện Phê duyệt tin đăng*

### Giao diện Phê duyệt phản ánh



*Hình 55: Giao diện Phê duyệt phản ánh*



*Hình 56: Giao diện Danh sách phản ánh của tin đăng*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả giao diện Phê duyệt phản ánh** | | | | |
| **Nội dung hiển thị** | **Kiểu** | **Editable** | **Giá trị mặc định** | **Mô tả** |
| **Hiển thị mặc định**  Thông tin của tin đăng | | | | |
| Thông tin | Button | no | N/A | Click để hiển thị danh sách phản ánh của tin đăng |
| Phản ánh | Button | no | N/A | Click để hiển thị danh sách phản ánh |
| Từ chối | Button | no | N/A | Click để từ chối tin đăng |
| Chấp thuận | Button | no | N/A | Click để chấp thuận tin đăng |
| Gửi email | Button | no | N/A | Click để hiển thị giao diện Soạn Email |
| **Giao diện Soạn Email** | | | | |
| Người gửi | Input text | yes | N/A | Nhập địa chỉ email người gửi |
| Người nhận | Input text | yes | N/A | Nhập email người gửi |
| Chủ đề | Input text | yes | N/A | Nhập chủ đề |
| Nội dung Email | Angular Editor  (Angular module) | yes | N/A | Nhập nội dung email |
| Hủy | Button | no | N/A | Click hủy thao tái soạn email |
| Gửi | Button | no | N/A | Click để gửi email |

*Bảng 58: Bảng mô tả giao diện Phê duyệt phản ánh*

# CHƯƠNG 5: XÂY DỰNG VÀ THỬ NGHIỆM



## Môi trường xây dựng và thử nghiệm

Ứng dụng “Chợ đồ cũ”được xây dựng dựa trên môi trường Java 8, Spring Boot 2 và Angular 7

### Cài đặt môi trường Java

* JDK(Java Deverlopment Kit) : là một bộ công cụ cung cấp môi trường phát triển ứng dụng viết bằng ngôn ngữ Java. Về cơ bản nó bao gồm:
  + - JRE (Java Runtime Environment): là một môi trường chạy ứng dụng Java.
    - Javac: Một chương trình để dịch mã mà bạn viết thành mã bytecode
    - Archive (jar): Là một chương trình nén các file thành một file duy nhất có đuôi jar. Thường dùng để đóng gói các file class.

Link tải JDK 8:

[*https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html*](https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)

* Spring Boot :

Spring Boot là một dự án phát triển bởi JAV (ngôn ngữ java) trong hệ sinh thái Spring framework. Nó giúp cho các lập trình đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển nghiệp vụ cho ứng dụng.

Để xây dựng một ứng dụng Spring Boot, cần cài đặt Spring Tool Suite (STS) trong Eclipse. String Tool Suite là một môi trường phát triển dựa trên Eclipse có được tùy biến cho phát triển ứng dụng Spring. Nó cung cấp một môi trường để thực thi, gỡ lỗi, chạy và triển khai các ứng dụng Spring, bao gồm tích hợp một số thành phần như Pivotal tc Server, Pivotal Cloud Foundry, Git, Maven, AspectJ để chạy một ứng dụng Spring Boot

### Cài đặt môi trường Angular

Đầu tiên muốn xây dựng một ứng dụng Angular, phải có một IDE như Visual Studio Code IDE hoặc JetBrains WebStorm để chạy ứng dụng Angular, sau đấy là cài đặt môi trường Angular

Link tải Visual Studio Code: [*https://code.visualstudio.com/*](https://code.visualstudio.com/)

Để cài đặt Angular , ta cần một số thứ như sau:

* NodeJS: Cung cấp các thư viện cần thiết để chạy dự án Angular

Link tải NodeJS : *https://nodejs.org/en/download/*

* npm (Node package manager - trình quản lý các package): Khi nodejs được cài đặt, npm cũng sẽ được cài đặt cùng với nó.
* Angular CLI: Sử dụng npm để cài đặt Angular CLI:

*npm install -g @angular/cli*

### Cài đặt môi trường MySQL

MySQL là hệ quản trị cơ sở dữ liệu quan hệ sử dụng ngôn ngữ truy vấn có cấu trúc (SQL). MySQL là có tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.Truy cập đường link để tải bản MySQL Community: [*https://dev.mysql.com/downloads/installer/*](https://dev.mysql.com/downloads/installer/)

Sau khi download và cài đặt sẽ có 2 phần quan trọng sau đây:

* MySQL Server: môi trường để thực hiện các câu lệnh MySQL
* MySQL Workbench (Công cụ trực quan để học và làm việc với MySQL)

Để cài MySQL Workbench yều cầu phải cài trước 2 thư viện mở rộng là:

* Microsoft .NET Framework 4 Client Profile

Link tải:

*https://www.microsoft.com/en-us/download/details.aspx?id=17113*

* Visual C++ Redistributable for Visual Studio 2013

Link tải:

*http://www.microsoft.com/en-us/download/details.aspx?id=40784*

## Các kết quả thử nghiệm

# CHƯƠNG 6: KẾT LUẬN



## Đã làm

Hệ thống hoạt động tốt trên điều kiện máy tính hoạt động bình thường, được cài đặt sẵn trình duyệt web: Chrome, Cốc cốc, Internet Explorer,…

Dưới đây là danh sách các chức năng hệ thống và trạng thái hiện tại:

|  |  |  |
| --- | --- | --- |
| **Chức năng** | **Mô tả** | **Trạng thái** |
| Đăng ký | Đăng ký thành viên để trở thành User Account | Hoàn thành |
| Đăng tin | Cho phép User đăng thông tin sản phẩm muốn bán lên hệ thống | Hoàn thành |
| Lưu tin,Bình luận | Cho phép User lưu và bình luận bài đăng | Hoàn thành |
| Xem thông tin chi tiết về bài đăng | Cho phép hiển thị chi tiết thông tin của bài đăng | Hoàn thành |
| Tìm kiếm tin đăng | Cho phép tìm kiếm tin đăng theo từ khóa hoặc theo tiêu chí như địa chỉ, thể loại | Hoàn thành |
| Kiểm duyệt, xóa bài đăng | Cho phép Kiểm duyệt và xóa bài của User | Hoàn thành |

*Bảng 59: Bảng mô tả các chức năng đã hoàn thành*

## Mở rộng

Sau khi đã hoàn thành gần như đầy đủ các chức năng theo bản thiết kế, nhóm mong muốn sẽ hoàn thiện, nâng cấp các chức năng gợi ý tìm kiếm, tìm kiếm full text search để cải thiện trải nghiệm người dùng, nâng cao hiệu năng cũng như giao diện của hệ thống, để đạt được một hệ thống nhanh, thông minh và đẹp cho người sử dụng.

# TÀI LIỆU THAM KHẢO

[1] <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/>

[2] <https://spring.io/guides/gs/spring-boot/>

[3] <https://spring.io/guides/gs/messaging-stomp-websocket/>

[4] <https://www.baeldung.com/spring-security-oauth-jwt>

[5] <https://angular.io/guide/quickstart>

[6] <https://angular-templates.io/tutorials/about/learn-angular-from-scratch-step-by-step>

[7] <https://material.angular.io/components/categories>

[8] <https://getbootstrap.com/docs/4.3/getting-started/introduction/>

[9] <https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp>