**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**



ĐỒ ÁN TỐT NGHIỆP

Hệ thống website Chợ đồ cũ

**TRỊNH VĂN TÀI**

[tai.tv133408@sis.hust.edu.vn](mailto:tai.tv133408@sis.hust.edu.vn)

**Ngành Công nghệ Thông tin và Truyền thông**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | TS.Nguyễn Hữu Đức  Chữ ký của GVHD |
| **Bộ môn:** | Hệ thống Thông tin |
| **Viện:** | Công nghệ Thông tin và Truyền thông |
| **HÀ NỘI, 12/2019** | |

**ĐỀ TÀI TỐT NGHIỆP**

Giáo viên hướng dẫn

Ký và ghi rõ họ tên

**Lời cảm ơn**

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 em.

**Tóm tắt nội dung đồ án**

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, 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.

Sinh viên thực hiện

Ký và ghi rõ họ tên

Trịnh Văn Tài

MỤC LỤC

[CHƯƠNG 1. GIỚI THIỆU 11](#_Toc28312756)

[1.1 Giới thiệu bài toán 11](#_Toc28312757)

[1.2 Phạm vi hệ thống 11](#_Toc28312758)

[1.3 Cách tiếp cận 11](#_Toc28312759)

[1.4 Giới thiệu cấu trúc đồ án 12](#_Toc28312760)

[CHƯƠNG 2. CÁC NỀN TẢNG CÔNG NGHỆ 13](#_Toc28312761)

[2.1 Giới thiệu về Spring Boot 13](#_Toc28312764)

[2.2 Giới thiệu về Angular 7 14](#_Toc28312765)

[2.3 Bootstrap 15](#_Toc28312769)

[2.4 MySQL 16](#_Toc28312770)

[CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG 17](#_Toc28312771)

[3.1 Các đối tác 17](#_Toc28312772)

[3.2 Các ca sử dụng 18](#_Toc28312773)

[3.3 Đặc tả ca sử dụng 19](#_Toc28312777)

[3.3.1 Đăng nhập 19](#_Toc28312778)

[3.3.2 Đăng kí tài khoản 20](#_Toc28312779)

[3.3.3 Đổi lại mật khẩu 22](#_Toc28312780)

[3.3.4 Tìm kiếm 24](#_Toc28312781)

[3.3.5 Đăng tin 26](#_Toc28312782)

[3.3.6 Bình luận 28](#_Toc28312783)

[3.3.7 Nhắn tin 30](#_Toc28312784)

[3.3.8 Theo dõi tài khoản 31](#_Toc28312785)

[3.3.9 Nạp thẻ cào 33](#_Toc28312786)

[3.3.10 Đẩy tin đăng 35](#_Toc28312787)

[3.3.11 Ẩn tin 36](#_Toc28312788)

[3.3.12 Phản ánh tin đăng 38](#_Toc28312789)

[3.3.13 Theo dõi thống kê lượt truy cập tin đăng 40](#_Toc28312790)

[3.3.14 Phê duyệt tin đăng 42](#_Toc28312791)

[3.3.15 Phê duyệt phản ánh 44](#_Toc28312792)

[3.3.16 Thống kê số lượng tin đăng 46](#_Toc28312793)

[CHƯƠNG 4. THIẾT KẾ HỆ THỐNG 49](#_Toc28312794)

[4.1 Kiến trúc hệ thống 49](#_Toc28312795)

[4.1.1 Kiến trúc triển khai 49](#_Toc28312796)

[4.1.2 Kiến trúc logic 50](#_Toc28312797)

[4.1.3 Kiến trúc mã nguồn 51](#_Toc28312798)

[4.1.4 Các kĩ thuật 51](#_Toc28312799)

[4.2 Thiết kế ca sử dụng 56](#_Toc28312800)

[4.2.1 Đăng nhập 57](#_Toc28312801)

[4.2.2 Đăng kí 59](#_Toc28312802)

[4.2.3 Đăng tin 62](#_Toc28312803)

[4.2.4 Tìm kiếm 66](#_Toc28312804)

[4.2.5 Bình luận 69](#_Toc28312805)

[4.2.6 Nhắn tin 71](#_Toc28312806)

[4.2.7 Theo dõi tài khoản 75](#_Toc28312807)

[4.2.8 Nạp thẻ cào 77](#_Toc28312808)

[4.2.9 Đẩy tin đăng 79](#_Toc28312809)

[4.2.10 Thống kê số lượng tin đăng 81](#_Toc28312810)

[4.3 Thiết kế cơ sở dữ liệu 85](#_Toc28312811)

[4.3.1 Sơ đồ quan hệ 85](#_Toc28312812)

[4.3.2 Mô tả cơ sở dữ liệu 86](#_Toc28312813)

[4.4 Thiết kế giao diện 101](#_Toc28312814)

[4.4.1 Giao diện Đăng nhập 102](#_Toc28312815)

[4.4.2 Giao diện Đăng kí 102](#_Toc28312816)

[4.4.3 Giao diện Đăng tin 103](#_Toc28312817)

[4.4.4 Giao diện Tìm kiếm 104](#_Toc28312818)

[4.4.5 Giao diện Bình luận 104](#_Toc28312819)

[4.4.6 Giao diện Nhắn tin 105](#_Toc28312820)

[4.4.7 Giao diện Trang cá nhân 106](#_Toc28312821)

[4.4.8 Giao diện Nạp thẻ cào 106](#_Toc28312822)

[4.4.9 Giao diện Đẩy tin 107](#_Toc28312823)

[4.4.10 Giao diện Phản ánh tin đăng 107](#_Toc28312824)

[4.4.11 Giao diện Thống kê lượt truy cập tin đăng 108](#_Toc28312825)

[4.4.12 Giao diện Phê duyệt tin đăng 108](#_Toc28312826)

[4.4.13 Giao diện Thống kê số lượng tin đăng 109](#_Toc28312827)

[CHƯƠNG 5. XÂY DỰNG VÀ THỬ NGHIỆM 111](#_Toc28312828)

[5.1 Môi trường xây dựng 111](#_Toc28312829)

[5.1.1 Cài đặt môi trường Java 111](#_Toc28312830)

[5.1.2 Cài đặt môi trường Angular 111](#_Toc28312831)

[5.1.3 Cài đặt môi trường MySQL 112](#_Toc28312832)

[5.1.4 Cài đặt hệ thống 112](#_Toc28312833)

[5.2 Các kết quả thử nghiệm 113](#_Toc28312834)

[5.2.1 Đăng nhập 114](#_Toc28312835)

[5.2.2 Tìm kiếm 115](#_Toc28312836)

[5.2.3 Nhắn tin 116](#_Toc28312837)

[5.2.4 Đăng tin 117](#_Toc28312838)

[5.2.5 Quản lí tin đăng 119](#_Toc28312839)

[5.2.6 Phê duyệt tin đăng 120](#_Toc28312840)

[5.2.7 Phê duyệt phản ánh 121](#_Toc28312841)

[CHƯƠNG 6. KẾT LUẬN 123](#_Toc28312842)

[TÀI LIỆU THAM KHẢO 124](#_Toc28312843)

**DANH MỤC HÌNH VẼ**

[*Hình 1: Kiến trúc Spring* 13](#_Toc28312935)

[*Hình 2: Tương tác giữa các thành phần trong Angular 7* 14](#_Toc28312936)

[*Hình 3: Các ca sử dụng* 18](#_Toc28312937)

[*Hình 4: Kiến trúc triển khai của hệ thống* 49](#_Toc28312938)

[*Hình 5: Kiến trúc logic của ứng dụng* 50](#_Toc28312939)

[*Hình 6: Luồng xử lí Auth0* 53](#_Toc28312940)

[*Hình 7: Quan hệ giữa các bảng cơ sở dữ liệu* 85](#_Toc28312941)

[*Hình 8: Quan hệ giữa bảng Product với Category và Location* 87](#_Toc28312942)

[*Hình 9: Quan hệ giữa bảng Category và Information* 89](#_Toc28312943)

[*Hình 10: Quan hệ giữa Product và Information* 90](#_Toc28312944)

[*Hình 11: Quan hệ giữa Product và Location* 91](#_Toc28312945)

[*Hình 12: Quan hệ giữa Product và Image* 92](#_Toc28312946)

[*Hình 13: Quan hệ giữa Comment với Product và User* 93](#_Toc28312947)

[*Hình 14: Quan hệ giữa Chat với User và Product* 94](#_Toc28312948)

[*Hình 15: Quan hệ giữa User và Follow\_user* 95](#_Toc28312949)

[*Hình 16: Quan hệ giữa Follow\_Product với Product và User* 96](#_Toc28312950)

[*Hình 17: Quan hệ giữa Notify với các bảng khác* 97](#_Toc28312951)

[*Hình 18: Quan hệ giữa Search và User* 99](#_Toc28312952)

[*Hình 19: Quan hệ giữa Visits và Product* 100](#_Toc28312953)

[*Hình 20: Minh họa giao diện Đăng nhập* 102](#_Toc28312954)

[*Hình 21: Minh họa giao diện Đăng kí tài khoản* 102](#_Toc28312955)

[*Hình 22: Minh họa giao diện Đăng tin* 103](#_Toc28312956)

[*Hình 23: Minh họa giao diện Tìm kiếm* 104](#_Toc28312957)

[*Hình 24: Minh họa giao diện Bình luận* 104](#_Toc28312958)

[*Hình 25: Minh họa giao diện Nhắn tin* 105](#_Toc28312959)

[*Hình 26: Minh họa giao diện Trang cá nhân* 106](#_Toc28312960)

[*Hình 27: Minh họa giao diện Nạp thẻ cào* 106](#_Toc28312961)

[*Hình 28: Minh họa giao diện Đẩy tin đăng* 107](#_Toc28312962)

[*Hình 29: Minh họa giao diện Phản ánh tin đăng* 107](#_Toc28312963)

[*Hình 30: Minh họa giao diện Thống kê lượt truy cập của tin đăng* 108](#_Toc28312964)

[*Hình 31: Minh họa giao diện Phê duyệt tin đăng* 108](#_Toc28312965)

[*Hình 32: Minh họa giao diện Soạn Email* 109](#_Toc28312966)

[*Hình 33: Minh họa giao diện Thống kê số lượng tin đăng* 109](#_Toc28312967)

[*Hình 34: Giao diện Đăng nhập* 114](#_Toc28312968)

[*Hình 35: Giao diện Trang chủ* 114](#_Toc28312969)

[*Hình 36: Giao diện Tìm kiếm* 115](#_Toc28312970)

[*Hình 37: Giao diện modal Chọn địa điểm* 115](#_Toc28312971)

[*Hình 38: Giao diện modal Chọn danh mục* 116](#_Toc28312972)

[*Hình 39: Giao diện Chi tiết tin đăng* 116](#_Toc28312973)

[*Hình 40: Giao diện Nhắn tin* 117](#_Toc28312974)

[*Hình 41: Giao diện Chọn địa điểm để đăng tin* 117](#_Toc28312975)

[*Hình 42: Giao diện Chọn thể loại để đăng tin* 118](#_Toc28312976)

[*Hình 43: Giao diện Thông tin chung để đăng tin* 118](#_Toc28312977)

[*Hình 44: Giao diện Thông tin chi tiết lúc đăng tin* 119](#_Toc28312978)

[*Hình 45: Giao diện Quản lí tin đăng* 119](#_Toc28312979)

[*Hình 46: Giao diện Thống kê tin đăng* 119](#_Toc28312980)

[*Hình 47: Giao diện Quản trị tin đăng* 120](#_Toc28312981)

[*Hình 48: Giao diện Phê duyệt tin đăng* 121](#_Toc28312982)

[*Hình 49: Giao diện Phê duyệt phản ánh* 121](#_Toc28312983)

[*Hình 50: Giao diện Danh sách phản ánh của tin đăng* 122](#_Toc28312984)

**DANH MỤC HÌNH VẼ**

[*Biểu đồ 1: Biểu đồ trình tự Đăng nhập* 57](#_Toc28312985)

[*Biểu đồ 2: Biểu đồ lớp Đăng nhập* 57](#_Toc28312986)

[*Biểu đồ 3: Biểu đồ trình tự Đăng kí* 59](#_Toc28312987)

[*Biểu đồ 4: Biểu đồ lớp Đăng kí* 60](#_Toc28312988)

[*Biểu đồ 5: Biểu đồ trình tự Đăng tin* 62](#_Toc28312989)

[*Biểu đồ 6: Biểu đồ lớp Đăng tin* 63](#_Toc28312990)

[*Biểu đồ 7: Biểu đồ trình tự Tìm kiếm* 66](#_Toc28312991)

[*Biểu đồ 8: Biểu đố lớp Tìm kiếm* 67](#_Toc28312992)

[*Biểu đồ 9: Biểu đồ trình tự Bình luận* 69](#_Toc28312993)

[*Biểu đồ 10: Biểu đồ lớp Bình luận* 69](#_Toc28312994)

[*Biểu đồ 11: Biểu đồ trình tự Nhắn tin* 71](#_Toc28312995)

[*Biểu đồ 12: Biểu đồ lớp Nhắn tin* 72](#_Toc28312996)

[*Biểu đồ 13: Biểu đồ trình tự Theo dõi tài khoản* 75](#_Toc28312997)

[*Biểu đồ 14: Biểu đồ lớp Theo dõi tài khoản* 75](#_Toc28312998)

[*Biểu đồ 15: Biểu đồ trình tự Nạp thẻ cào* 77](#_Toc28312999)

[*Biểu đồ 16: Biểu đồ lớp Nạp thẻ cào* 77](#_Toc28313000)

[*Biểu đồ 17:Biểu đồ trình tự Đẩy tin* 79](#_Toc28313001)

[*Biểu đồ 18: Biểu đồ lớp Đẩy tin* 79](#_Toc28313002)

[*Biểu đồ 19: Biểu đồ trình tự Thống kê số lượng tin đăng* 81](#_Toc28313003)

[*Biểu đồ 20: Biểu đồ lớp Thống kê số lượng tin đăng* 82](#_Toc28313004)

**DANH MỤC BẢNG**

[*Bảng 4.1 Mô tả lớp User-ca sử dụng Đăng nhập* 58](#_Toc28313014)

[*Bảng 4.2 Mô tả lớp UserRepository-ca sử dụng Đăng nhập* 58](#_Toc28313015)

[*Bảng 4.3: Mô tả lớp UserService-ca sử dụng Đăng nhập* 58](#_Toc28313016)

[*Bảng 4.4: Mô tả lớp LoginController-ca sử dụng Đăng nhập* 58](#_Toc28313017)

[*Bảng 4.5: Mô tả lớp User-ca sử dụng Đăng kí* 60](#_Toc28313018)

[*Bảng 4.6: Mô tả lớp UserRepository-ca sử dụng Đăng kí* 61](#_Toc28313019)

[*Bảng 4.7: Mô tả lớp UserService-ca sử dụng Đăng nhập* 61](#_Toc28313020)

[*Bảng 4.8: Mô tả lớp MailService-ca sử dụng Đăng kí* 61](#_Toc28313021)

[*Bảng 4.9: Mô tả lớp RegisterController-ca sử dụng Đăng kí* 62](#_Toc28313022)

[*Bảng 4.10: Mô tả lớp Category-ca sử dụng Đăng tin* 63](#_Toc28313023)

[*Bảng 4.11: Mô tả lớp Location-ca sử dụng Đăng tin* 64](#_Toc28313024)

[*Bảng 4.12: Mô tả lớp Product-ca sử dụng Đăng tin* 64](#_Toc28313025)

[*Bảng 4.13: Mô tả lớp CategoryRepository-ca sử dụng Đăng tin* 65](#_Toc28313026)

[*Bảng 4.14: Mô tả lớp LocationRepository-ca sử dụng Đăng tin* 65](#_Toc28313027)

[*Bảng 4.15: Mô tả lớp UserService-ca sử dụng Đăng tin* 65](#_Toc28313028)

[*Bảng 4.16: Mô tả lớp ProductService-ca sử dụng Đăng tin* 65](#_Toc28313029)

[*Bảng 4.17: Mô tả lớp ProductController-ca sử dụng Đăng tin* 66](#_Toc28313030)

[*Bảng 4.18: Mô tả lớp CategoryService-ca sử dụng Tìm kiếm* 67](#_Toc28313031)

[*Bảng 4.19: Mô tả lớp LocationService-ca sử dụng Tìm kiếm* 67](#_Toc28313032)

[*Bảng 4.20: Mô tả lớp ProductService-ca sử dụng Tìm kiếm* 68](#_Toc28313033)

[*Bảng 4.21: Mô tả lớp ProductController-ca sử dụng Tìm kiếm* 68](#_Toc28313034)

[*Bảng 4.22: Mô tả lớp Comment-ca sử dụng Bình luận* 70](#_Toc28313035)

[*Bảng 4.23: Mô tả lớp CommentRepository-ca sử dụng Bình luận* 70](#_Toc28313036)

[*Bảng 4.24: Mô tả lớp ProductService-ca sử dụng Bình luận* 70](#_Toc28313037)

[*Bảng 4.25: Mô tả lớp CommentController-ca sử dụng Bình luận* 71](#_Toc28313038)

[*Bảng 4.26: Mô tả đối tượng Chat-ca sử dụng Nhắn tin* 73](#_Toc28313039)

[*Bảng 4.27: Mô tả đối tượng Dialog-ca sử dụng Nhắn tin* 73](#_Toc28313040)

[*Bảng 4.28: Mô tả lớp ChatRepository-ca sử dụng Nhắn tin* 73](#_Toc28313041)

[*Bảng 4.29: Mô tả lớp DialogRepository-ca sử dụng Nhắn tin* 74](#_Toc28313042)

[*Bảng 4.30: Mô tả lớp ChatController-ca sử dụng Nhắn tin* 74](#_Toc28313043)

[*Bảng 4.31: Mô tả lớp FollowUser-ca sử dụng Theo dõi tài khoản* 76](#_Toc28313044)

[*Bảng 4.32: Mô tả lớp FollowUserRepository-ca sử dụng Theo dõi tài khoản* 76](#_Toc28313045)

[*Bảng 4.33: Mô tả lớp UserController-ca sử dụng Theo dõi tài khoản* 76](#_Toc28313046)

[*Bảng 4.34: Mô tả lớp User-ca sử dụng Nạp thẻ cào* 78](#_Toc28313047)

[*Bảng 4.35: Mô tả lớp UserRepository- ca sử dụng Nạp thẻ cào* 78](#_Toc28313048)

[*Bảng 4.36: Mô tả lớp UserService- ca sử dụng Nạp thẻ cào* 78](#_Toc28313049)

[*Bảng 4.37: Mô tả lớp UserController-ca sử dụng Nạp thẻ cào* 78](#_Toc28313050)

[*Bảng 4.38: Mô tả lớp PushProduct-ca sử dụng Đẩy tin* 80](#_Toc28313051)

[*Bảng 4.39: Mô tả lớp PushProductRepository-ca sử dụng Đẩy tin* 80](#_Toc28313052)

[*Bảng 4.40: Mô tả lớp ProductController- ca sử dụng Đẩy tin* 80](#_Toc28313053)

[*Bảng 4.41: Mô tả lớp ProductRepository-ca sử dụng Thống kê số lượng tin đăng* 82](#_Toc28313054)

[*Bảng 4.42: Mô tả lớp ProductService-ca sử dụng Thống kê số lượng tin đăng* 83](#_Toc28313055)

[*Bảng 4.43: Mô tả lớp AdminController- ca sử dụng Thống kê số lượng tin đăng* 84](#_Toc28313056)

[*Bảng 4.44: Bảng cơ sở dữ liệu User* 86](#_Toc28313057)

[*Bảng 4.45: Bảng cơ sở dữ liệu Product* 88](#_Toc28313058)

[*Bảng 4.46: Bảng cơ cở dữ liệu Category* 89](#_Toc28313059)

[*Bảng 4.47: Bảng cơ sở dữ liệu Information* 90](#_Toc28313060)

[*Bảng 4.48: Bảng cơ sở dữ liệu Information\_value* 91](#_Toc28313061)

[*Bảng 4.49: Bảng cơ sở dữ liệu Location* 91](#_Toc28313062)

[*Bảng 4.50: Bảng cơ sở dữ liệu Image* 92](#_Toc28313063)

[*Bảng 4.51: Bảng cơ sở dữ liệu Comment* 93](#_Toc28313064)

[*Bảng 4.52: Bảng cơ sở dữ liệu Chat* 94](#_Toc28313065)

[*Bảng 4.53: Bảng cơ sở dữ liệu Follow\_User* 95](#_Toc28313066)

[*Bảng 4.54: Bảng cơ sở dữ liệu Follow\_Product* 96](#_Toc28313067)

[*Bảng 4.55: Bảng cơ sở dữ liệu Notify* 98](#_Toc28313068)

[*Bảng 4.56: Bảng cơ sở dữ liệu Search* 99](#_Toc28313069)

[*Bảng 4.57: Bảng cơ sở dữ liệu Visits* 100](#_Toc28313070)

# GIỚI THIỆU

Trong chương này em xin giới thiệu qua về đề tài và hướng tiếp cận để giải quyết bài toán.

## 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. Cung cấp cho người Việt Nam một nền tảng trực tuyến với cách sử dụng 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 framework Java Spring để triển khai 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

Mục này 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, 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 khó khăn và đưa ra hướng phát triển trong tương lai.

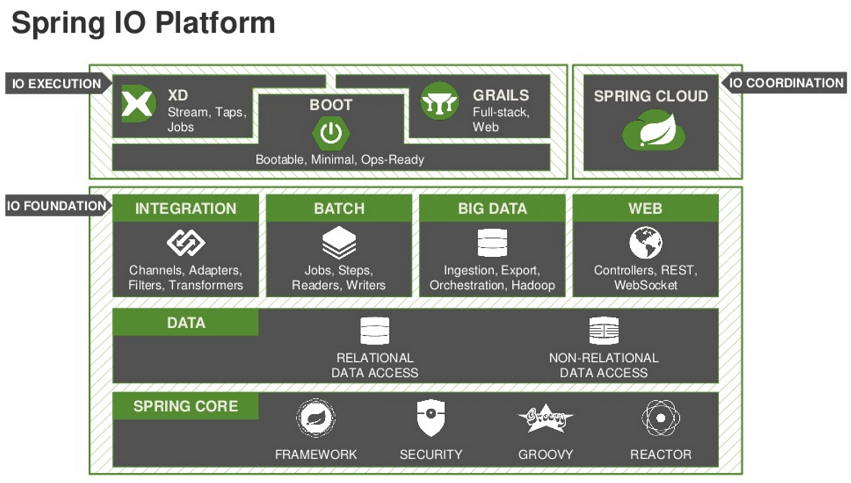
Phần cuối cùng là mục tài liệu tham khảo .

# CÁC NỀN TẢNG CÔNG NGHỆ



Chương này sẽ giới thiệu một số nền tảng công nghệ được sử dụng để triển khai đồ án.

## Giới thiệu về Spring Boot

****

*Hình 1: Kiến trúc Spring*

Spring Bootlà 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

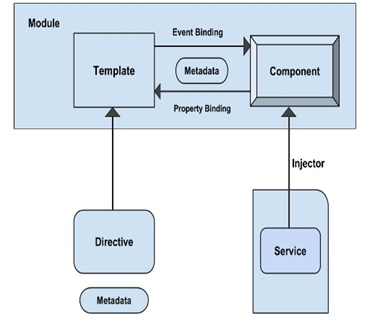
1. Dễ dàng phát triển các ứng dụng dựa trên Spring với java hoặc Groovy.
2. Giảm thiểu thời gian phát triển và tăng năng suất.
3. Tránh việc viết các file cấu hình xml.
4. Dễ dàng tương tác ứng dụng Spring Boot với hệ sinh thái của Spring như Spring JDBC, Spring ORM, Spring Data, Spring Security,…
5. 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.

Đọc tài liệu số [1], để hiểu thêm về Spring Boot.

## 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. Xem thêm tại tài liệu số [2] .

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. Xem thêm tại tài liệu số [3].

## 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. Xem thêm tại tài liệu số [4].

# PHÂN TÍCH HỆ THỐNG

Chương này trình bày về các yêu cầu của hệ thống Chợ đồ cũ bao gồm các chức năng chính-phụ, nghiệp vụ cần phải xử lí đảm bảo tính chính xác đối với yêu cầu của bài toán.

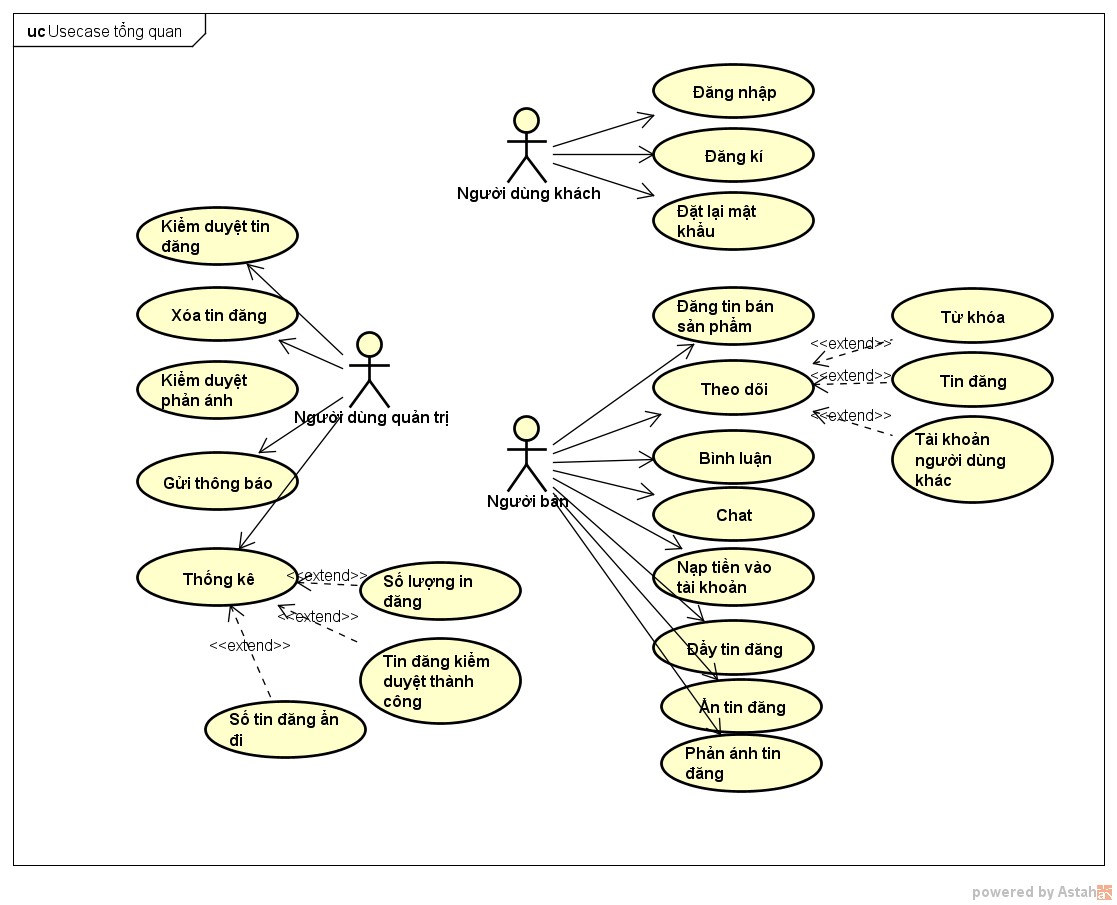
## Các đối tác

Hệ thống gồm các mức người dùng sau đây:

* **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, phản ánh tin đăng…
* **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 , …
* **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

*Hình* ***3*** dưới đây mô tả tổng quan về các ca sử dụng trong hệ thống. Chi tiết hơn về các ca sử dụng sẽ được mô tả ở mục 3.3



*Hình 3: Các ca sử dụng*



## Đặc tả ca sử dụng

Sau đây là đặc tả chi tiết các ca sử dụng của hệ thống, mỗi ca sử dụng sẽ gồm mô tả chung, mô tả kịch bản và mô tả một số nghiệp vụ thuộc mỗi ca sử dụng.

### Đăng nhập

1. Mô tả chung

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

1. Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Đăng nhập tài khoản | | | |
| 1 | Nhấp chọn “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 |

1. Mô tả nghiệp vụ

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

### Đăng kí tài khoản

1. Mô tả chung

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

1. Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Đăng kí tài khoản | | | |
| 1 | Nhấp chọn “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 đã đăng kí** | | | |
|  |  | 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 |

1. 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 |

### Đổi lại mật khẩu

1. Mô tả chung

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

1. Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Đổi mật khẩu | | | |
| 1 | Nhấp chọn “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 |

1. 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 |

### Tìm kiếm

1. Mô tả chung

|  |  |
| --- | --- |
| **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 được phân trang |

1. Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Tìm kiếm | | | |
| 1 | Nhấp chọn “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** | | | |
|  |  |  |  |

1. Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 4 | BR\_5 | * + - * **Tìm kiếm tin đăng:**   Sử dụng Specification, Pagination và Sorting trong Spring Data JPA để viết câu truy vấn tìm kiếm |

### Đăng tin

1. Mô tả chung

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Đăng tin |
| **Tác nhân** | User |
| **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 |

1. Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Đăng tin | | | |
| 1 | Nhấp chọn “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 |

1. Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 4 | BR\_6 | * + - * **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 |

### Bình luận

1. Mô tả chung

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Bình luận |
| **Tác nhân** | User |
| **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, cập nhật giao diện bình luận |

1. 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 |
|  |  | 5 | * Gửi lại nội dung bình luận về máy khách * Cập nhật giao diện bình luận |
| **Kịch bản phụ: Không có** | | | |
|  |  |  |  |
| **Xử lí ngoại lệ: Không có** | | | |
|  |  |  |  |

1. Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 4 | BR**\_**7 | * + - * **Lưu bình luận:**   Bình luận được lưu gồm các thông tin bắt buộc như thời gian, thông tin tài khoản bình luận,nội dung,… |
| 5 | BR\_8 | * + - * **Gửi bình luận:**       * Dùng cơ chế websocket để gửi bình luận |

### Nhắn tin

1. Mô tả chung

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

1. Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Chat | | | |
| 1 | Nhấp chọn “Trò chuyện” |  |  |
|  |  | 2 | Hiển thị giao diện nhắn tin |
| 3 | Gửi nội dung tin nhắn |  |  |
|  |  | 4 | * Hệ thống lưu tin nhắn |
|  |  | 5 | * Gửi tin nhắn đến tài khoản nhận * Cập nhật giao diện nhắn tín |
| **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** | | | |
|  |  |  |  |

1. Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 4 | BR\_9 | * + - * **Lưu tin nhắn:**   Tin nhắn được lưu gồm các thông tin bắt buộc như thời gian, người gửi,người nhận, nội dung,tin đăng… |
| 5 | BR\_10 | * + - * **Gửi tin nhắn đến tài khoản nhận:**       * Dùng cơ chế websocket để gửi tin nhắn |

### Theo dõi tài khoản

1. Mô tả chung

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Theo dõi tài khoản |
| **Tác nhân** | User |
| **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 mới lên hệ thống |
| **Sự kiện kích hoạt** | Nhấp chọn button “Theo dõi” |
| **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 theo dõi, thông báo đã theo dõi thành công |

1. Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Theo dõi tài khoản | | | |
| 1 | Nhấp chọn “Theo dõi” |  |  |
|  |  | 2 | Hệ thống lưu thông tin theo dõi, thông báo đã theo dõi thành công |
| **Kịch bản phụ: Không có** | | | |
|  |  |  |  |
| **Xử lí ngoại lệ: Không có** | | | |
|  |  |  |  |

1. Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 4 | BR\_11 | * + - * **Lưu thông tin tài khoản theo dõi:**   Thông tin theo dõi được lưu gồm người theo dõi, người được theo dõi |

### Nạp thẻ cào

1. Mô tả chung

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Nạp thẻ cào |
| **Tác nhân** | User |
| **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 nạp tiền |
| **Kết quả đầu ra** | Hệ thống cộng dồn số tiền trong thẻ cào vào tài khoản của người dùng, lưu lịch sử nạp tiền |

1. 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 |

1. Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 2 | BR**\_**12 | * + - * **Gọi API trang napthengay.com:**   Các tham số đầu vào bao gồm:   |  |  | | --- | --- | | 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 | |
| 3 | BR\_13 | * + - * **Lưu lịch sử nạp tiền:**       * Thông tin bao gồm thời gian, tài khoản người dùng và số tiền nạp vào, mã thẻ |

### Đẩy tin đăng

1. Mô tả chung

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Đẩy tin đăng |
| **Tác nhân** | User |
| **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 |

1. Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Đẩy tin đăng | | | |
| 1 | Nhấp chọn “Đẩy tin” một tin đăng bất kì |  |  |
|  |  | 2 | Hiển thị giao diện đẩy tin |
| 3 | Chọn ngày để đẩy tin |  |  |
|  |  | 4 | Hệ thống hiển thị số tiền cần thanh toán để đẩy tin |
| 5 | Đẩy tin |  |  |
|  |  | 6 | Hệ thống trừ số tiền của tài khoản người dùng, tin đăng được lưu vào bảng ưu tiên. |
| **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** | | | |
|  |  |  |  |

1. Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 6 | BR\_14 | * + - * **Lưu tin đăng vào bảng ưu tiên:**   Bản ghi bao gồm tin đăng, ngày đẩy tin |

### Ẩn tin

1. Mô tả chung

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Ẩn tin đăng |
| **Tác nhân** | User |
| **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 thực hiện tìm kiếm |

1. Kịch bản hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự** | **Tác nhân** | **Thứ tự** | **System** |
| **Kịch bản chính:** Ẩn tin | | | |
| 1 | Nhấp chọn “Ẩn tin” một tin đăng bất kì |  |  |
|  |  | 2 | Hỏi người dùng có muốn tiếp tục thao tác |
| 3 | Chọn Ok |  |  |
|  |  | 4 | Cập nhật trạng thái của tin đăng về “đã ẩn” |
| **Kịch bản phụ: Không có** | | | |
|  |  |  |  |
| **Xử lí ngoại lệ:Không có** | | | |
|  |  |  |  |

1. Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 4 | BR**\_**15 | * + - * **Ẩn tin đăng:**   Thay đổi trạng thái của tin đăng từ “chấp thuận” sang “đã ẩn”, tin đăng đã bán sẽ không xuất hiện trong kết quả tìm kiếm từ hệ thống |

### Phản ánh tin đăng

1. Mô tả chung

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Phản ánh tin đăng |
| **Tác nhân** | Guest |
| **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 |

1. 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** | | | |
|  |  |  |  |

1. Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 4 | BR**\_**16 | * + - * **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 |

### Theo dõi thống kê lượt truy cập tin đăng

1. Mô tả chung

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Theo dõi thống kê lượt truy cập tin đăng |
| **Tác nhân** | User |
| **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) |

1. 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 trong tháng được chọn * Hiển thị số liệu theo biểu đồ đường |
| **Kịch bản phụ: Không có** | | | |
|  |  |  |  |
| **Xử lí ngoại lệ:Không có** | | | |
|  |  |  |  |

1. Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 6 | BR\_17 | * + - * **Thu thập lượng truy cập:**   Lấy thông tin số lượng truy cập của các ngày thuộc tháng được chọn |

### Phê duyệt tin đăng

1. Mô tả chung

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Phê duyệt tin đăng |
| **Tác nhân** | admin |
| **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 |

1. 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** | | | |
|  |  |  |  |

1. Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 6 | BR**\_**18 | * + - * **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 |

### Phê duyệt phản ánh

1. Mô tả chung

|  |  |
| --- | --- |
| **Tên ca sử dụng** | Phê duyệt phản ánh |
| **Tác nhân** | Admin |
| **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 |

1. 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 chi tiết của tin đăng 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” |  |  |
|  |  | 6.1 | Quay lại màn hình phản ánh của tin đăng |
| **Xử lí ngoại lệ:Không có** | | | |
|  |  |  |  |

1. Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 6 | BR**\_**19 | * + - * **Cập nhật trạng thái tin đăng:**   Tin đăng bị từ chối có trạng thái là 2 |

### Thống kê số lượng tin đăng

1. Mô tả chung

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

1. 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 thuộc năm được 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** | | | |
|  |  |  |  |

1. Mô tả nghiệp vụ

| **Thứ tự** | **BR Code** | **Mô tả** |
| --- | --- | --- |
| 2 | BR\_20 | * + - * **Thu thập số lượng tin đăng theo năm:**   Nhóm các tin đăng có thời gian đăng cùng năm lại với nhau |
| 4 | BR\_21 | * + - * **Thu thập số lượng tin đăng theo tháng:**   Nhóm các tin đăng có thời gian đăng cùng tháng thuộc năm người dùng chọn |
| 6 | BR\_22 | * + - * **Thu thập số lượng tin đăng theo ngày:**   Nhóm các tin đăng có thời gian đăng thuộc tháng và năm người dùng chọn |

# THIẾT KẾ HỆ THỐNG

Sau đây là phần thiết kế hệ thống website Chợ đồ cũ, gồm 4 phần:

* Phần 1 : Giới thiệu về kiến trúc hệ thống mà em sẽ triển khai.
* Phần 2 : Mô tả thiết kế các ca sử dụng chính trong hệ thống.
* Phần 3 : Mô tả thiết kế cơ sở dữ liệu.
* Phần 4 : Mô tả thiết kế giao diện.

## Kiến trúc hệ thống

Mở đầu là phần kiến trúc của hệ thống , bao gồm kiến trúc triển khai, kiến trúc logic của ứng dụng , kiến trúc mã nguồn và một số kĩ thuật được sử dụng để xây dựng hệ thống.

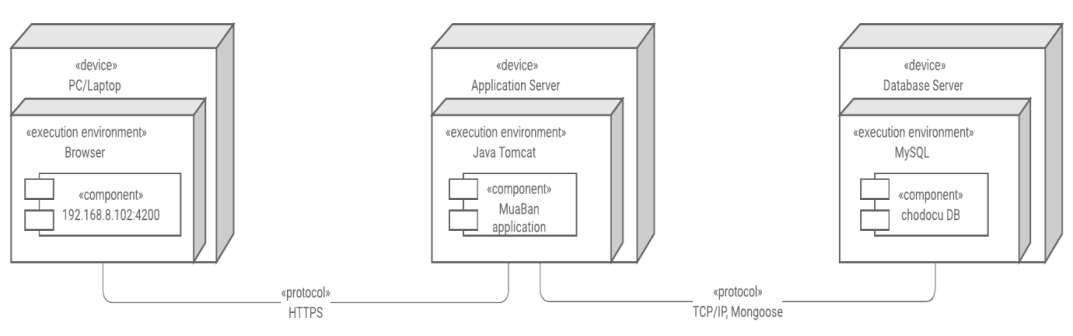
### Kiến trúc triển khai

Vì mục tiêu xây dựng một hệ thống phục vụ số lượng người sử dụng vừa phải, tiết kiệm chi phí nên em lựa chọn triển khai ứng dụng trên hai server gồm:

* Application Server : dùng để triển khai chức năng nghiệp vụ xử lí các yêu cầu từ phía người dùng.
* Database Server: dùng để triển khai database của ứng dụng.

Ứng dụng website Chợ đồ cũ sẽ triển khai trên server môi trường Java Tomcat, nó lắng nghe những yêu cầu từ phía người dùng (client), ví dụ như từ một trình duyệt và xử lý những yêu cầu đó, sau đó trả kết quả có định dạng là chuỗi JSON. Trình duyệt sẽ hiển thị những kết quả này lên một cách tốt nhất đến client. Ứng dụng kết nối tới chodocu database sử dụng môi trường MySQL, thực hiện các truy vấn đến cơ sở dữ liệu dựa vào mã nguồn ứng dụng

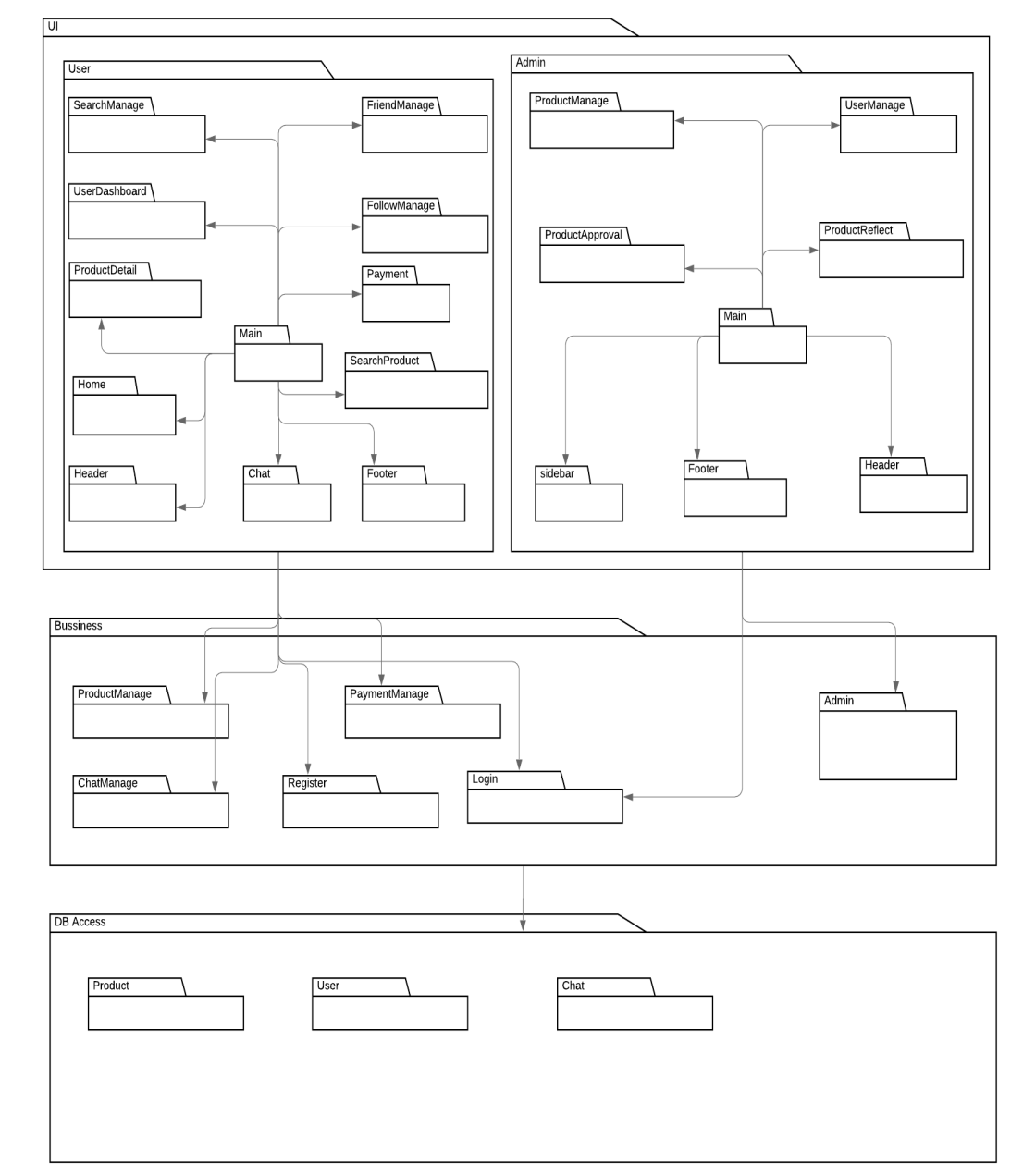
Người dùng sẽ kết nối tới ứng dụng thông qua trình duyệt bằng địa chỉ 192.168.8.102:4200. *Hình* ***4*** dưới đây đưa ra cái nhìn tổng quan về kiến trúc triển khai của hệ thống.



*Hình 4: Kiến trúc triển khai của hệ thống*

### Kiến trúc logic

*Hình* ***5*** dưới đây mô tả tổng quan về kiến trúc logic của hệ thống, gồm 3 phần chính là UI , Bessiness , DB Access và một số thành phần chính của mỗi phần.



*Hình 5: Kiến trúc logic của ứng dụng*

### Kiến trúc mã nguồn

Mã nguồn của hệ thống gồm 2 phần

* Phần Backend dựa trên nền tảng Spring Boot 2.0 sẽ có một số gói chính như sau
  + Config: chứa các file cấu hình bảo mật Spring Security và cấu hình Websocket.
  + Constant : chứa các file lưu giá trị một số biến được sử dụng trong hệ thống và các URL thao tác với resource.
  + Controller : chứa các Restful API, xử lí yêu cầu người dùng.
  + Model : chứa các lớp đối tượng của hệ thống, mỗi đối tượng sẽ tương ứng với một bảng trong database.
  + Repository: chứa các file interface được implement từ JpaRepository, các file có nhiệm vụ xử lý truy vấn với database
  + Service: chứa các file interface mô tả tác vụ của hệ thống
  + ServiceImp: chứa các file cài đặt xử lí nghiệp vụ
* Phần Frontend gồm 2 chương trình Angular Application:
  + Chương trình dành cho nhân viên quản trị
  + Chương rình dành cho người dùng bình thường

### Các kĩ thuật

1. 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, xem thêm tại tài liệu số [5] để hiểu thêm về RxJS

* 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

1. 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. Đọc tài liệu số [6] và số [7] để hiểu thêm cách tạo một RESTful web services trong Spring Boot.

* 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.

1. 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.

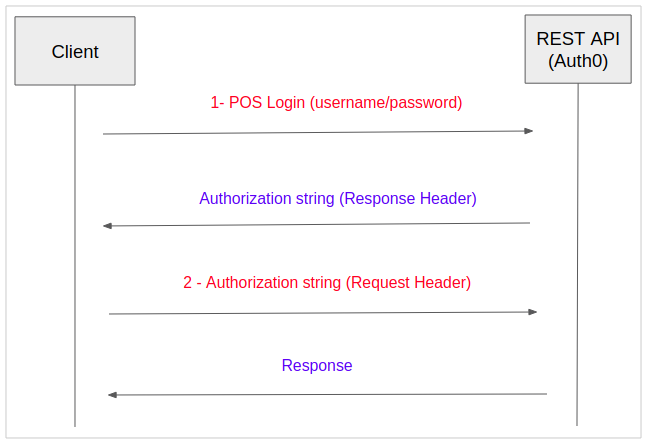
Tuy nhiên không phải tất cả các các RESTful API đều có thể công khai, vì tính nhạy cảm của nó, vì vậy cần phải bảo mật chúng. Có một vài kỹ thuật để bảo mật RESTful API như :

* 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.

Đọc tài liệu số [8] để hiểu thêm cách triển khai bảo mật một RESTful API trong Spring Boot



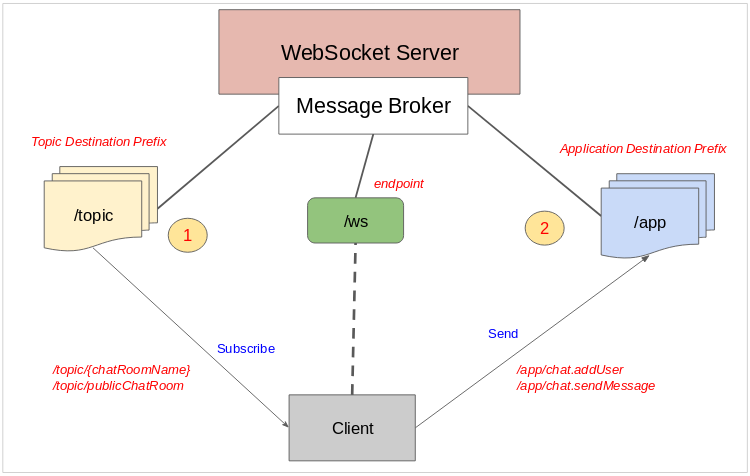
*Hình 6: Luồng xử lí Auth0*

1. 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.

Đọc thêm tài liệu số [9] để hiểu thêm WebSocket trong Spring Boot.

## Thiết kế ca sử dụng

Sau đây là phần thiết kết một số ca sử dụng quan trọng trọng hệ thống, mỗi ca sử dụng sẽ được mô tả qua hai biểu đồ sau:

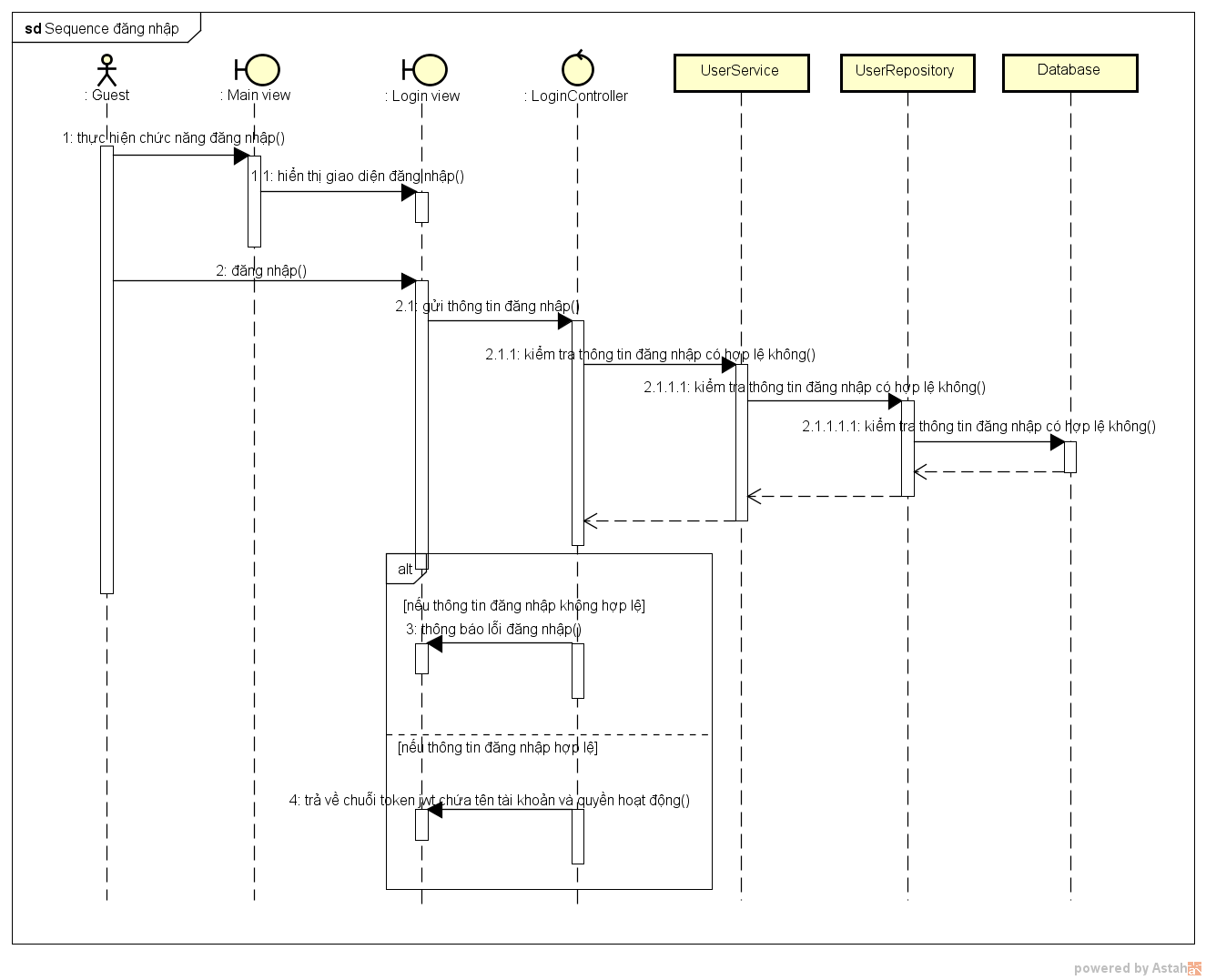
* Biểu đồ trình tự (Sequence Diagram): là bản vẽ mô tả sự tương tác theo thời gian của các đối tượng để tạo nên các chức năng của hệ thống.
* Biểu đồ lớp (Class Diagram): là bản vẽ mô tả cấu trúc của hệ thống bằng cách hiển thị các lớp của hệ thống, thuộc tính, hoạt động của chúng và mối quan hệ giữa các đối tượng.

Trong biểu đồ lớp sẽ có một số lớp như sau:

* 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< >.
* Entity: lớp đối mô tả các đối tượng trong hệ thống.
* Service: lớp xử lí nghiệp vụ của hệ thống.
* Repository: lớp thực hiện các truy vấn đến database.
* Boundary: lớp mô tả giao hiện của hệ thống.

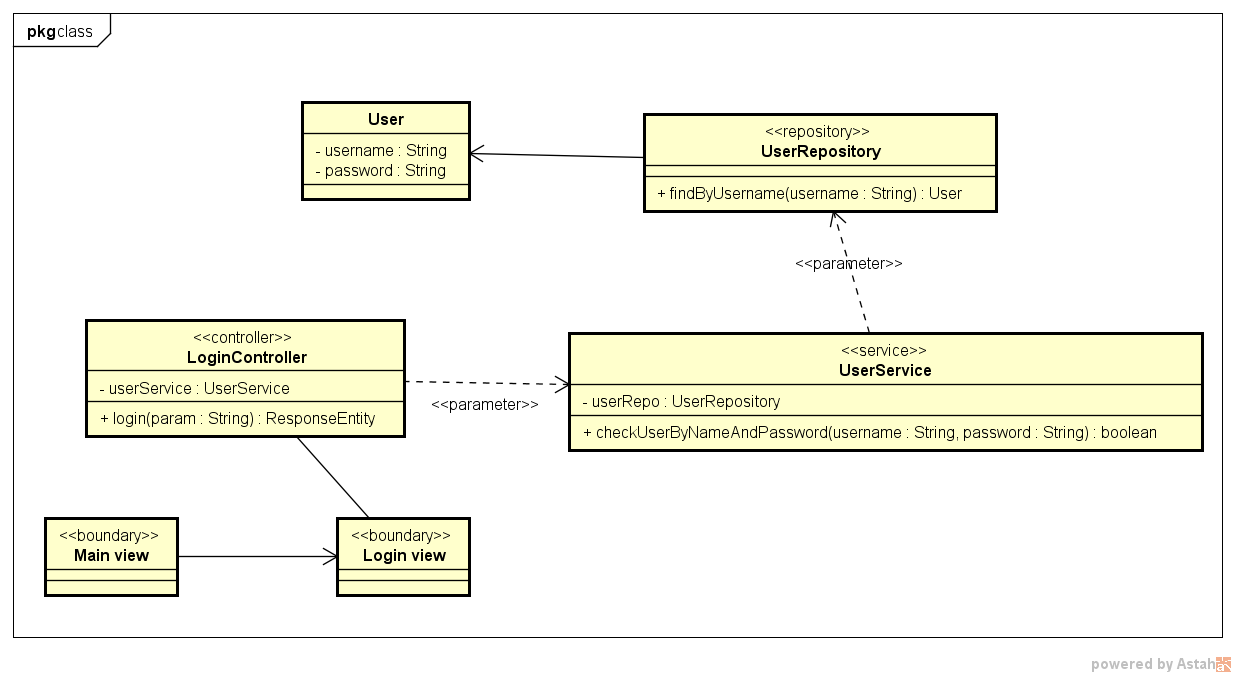
### Đăng nhập

1. Biểu đồ trình tự



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

1. Biểu đồ lớp



*Biểu đồ 2: Biểu đồ lớp Đăng nhập*

*Bảng 4.1 Mô tả lớp User-ca sử dụng Đăng nhập*

|  |  |
| --- | --- |
| **User**  Đối tượng lưu thông tin tài khoản người dùng | |
| **Thuộc tính** | **Mô tả** |
| username | Tên tài khoản dùng để đăng nhập |
| password | Mật khẩu tài khoản dùng để đăng nhập |

*Bảng 4.2 Mô tả lớp UserRepository-ca sử dụng Đăng nhập*

|  |  |
| --- | --- |
| **UserRepository**  Interface kế thừa từ JpaRepository thực hiện các câu truy vấn liên quan đến đối tượng User | |
| **Hàm** | **Mô tả** |
| findByUsername(String username) | Trả về một đối tượng User có giá trị trường *username* bằng tham số đầu vào *username* |

*Bảng 4.3: Mô tả lớp UserService-ca sử dụng Đăng nhập*

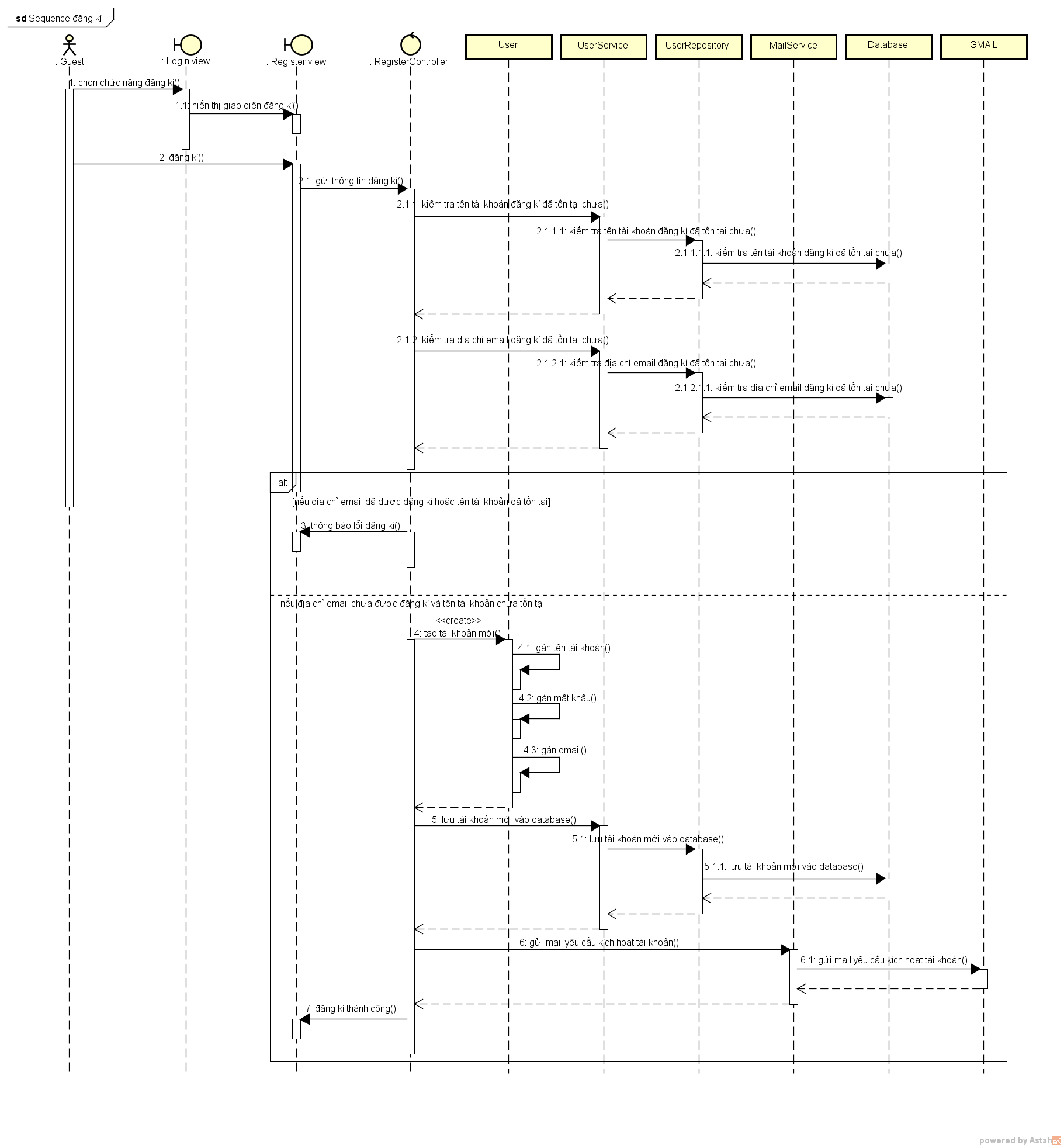
|  |  |
| --- | --- |
| **UserService**  Lớp service xử lí các tác vụ liên quan đến đối tượng User | |
| **Hàm** | **Mô tả** |
| checkUserByNameAndPassword(String username, String password) | Nếu tồn tại một tài khoản có tên tài khoản là *username* và mật khẩu là *password* thì trả về giá trị true, ngược lại là false |

*Bảng 4.4: Mô tả lớp LoginController-ca sử dụng Đăng nhập*

|  |  |
| --- | --- |
| **LoginController**  *Lớp Controller xử lí yêu cầu đăng nhập* | |
| ***Hàm*** | ***Mô tả*** |
| *login(String param)* | Hàm xử lí yêu cầu đăng nhập từ người dùng, nhận đầu vào là một *@RequestParam String param*(chuỗi JSON lưu thông tin đăng nhập) |

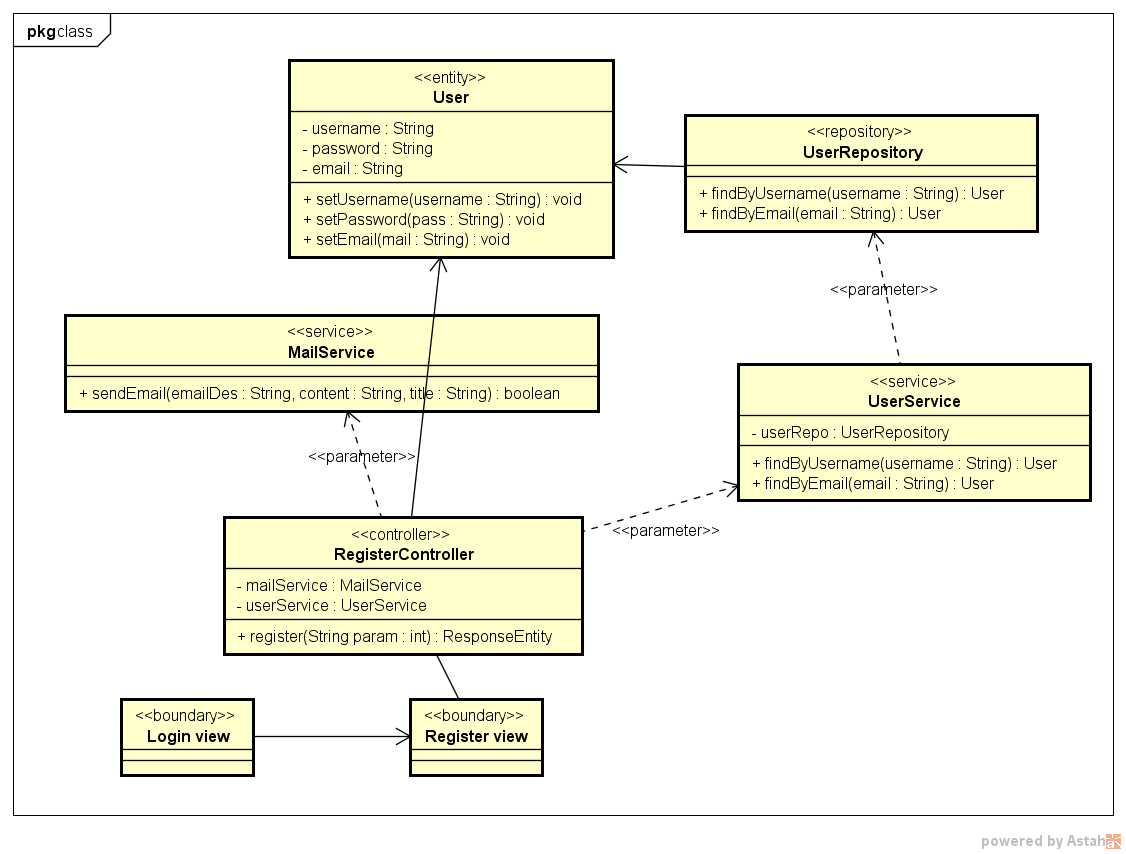
### Đăng kí

1. Biểu đồ trình tự



*Biểu đồ 3: Biểu đồ trình tự Đăng kí*

1. Biểu đồ lớp



*Biểu đồ 4: Biểu đồ lớp Đăng kí*

*Bảng 4.5: Mô tả lớp User-ca sử dụng Đăng kí*

|  |  |
| --- | --- |
| **User**  Đối tượng lưu thông tin tài khoản người dùng | |
| **Thuộc tính** | **Mô tả** |
| username | Tên tài khoản dùng để đăng nhập |
| password | Mật khẩu tài khoản dùng để đăng nhập |
| email | Địa chỉ Email |

*Bảng 4.6: Mô tả lớp UserRepository-ca sử dụng Đăng kí*

|  |  |
| --- | --- |
| **UserRepository**  Interface kế thừa từ JpaRepository thực hiện các câu truy vấn liên quan đến đối tượng User | |
| **Hàm** | **Mô tả** |
| findByUsername(String username) | Trả về một đối tượng User có giá trị trường *username* bằng tham số đầu vào *username* |
| findByEmail(String email) | Trả về một đối tượng User có giá trị trường *email* bằng tham số đầu vào *email* |
| save(User user) | Lưu một đối tượng User vào database |

*Bảng 4.7: Mô tả lớp UserService-ca sử dụng Đăng nhập*

|  |  |
| --- | --- |
| **UserService**  Lớp service xử lí các tác vụ liên quan đến đối tượng User | |
| **Hàm** | **Mô tả** |
| findByUsername(String username) | Trả về một đối tượng User có giá trị trường *username* bằng tham số đầu vào *username* |
| findByEmail(String email) | Trả về một đối tượng User có giá trị trường *email* bằng tham số đầu vào *email* |
| save(User user) | Lưu một đối tượng User vào database |

*Bảng 4.8: Mô tả lớp MailService-ca sử dụng Đăng kí*

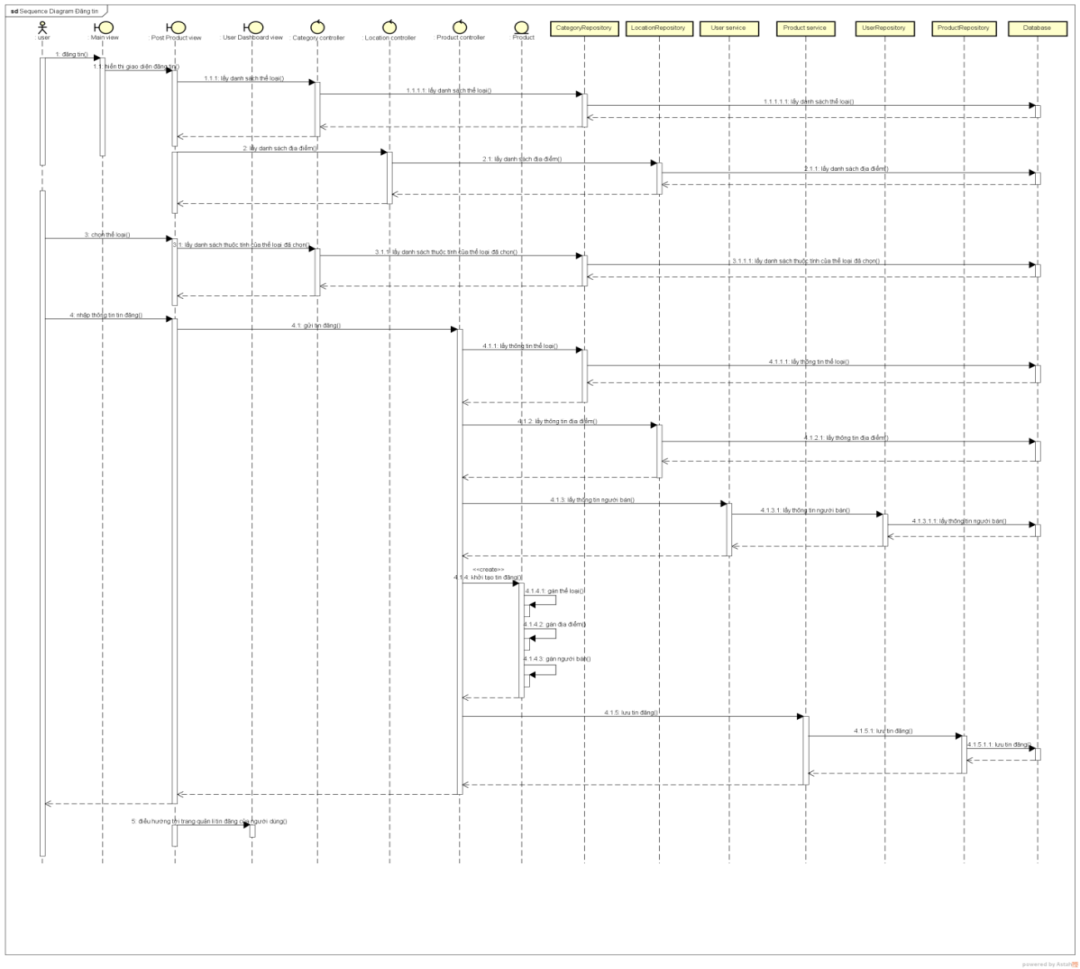
|  |  |
| --- | --- |
| **MailService**  Lớp service xử lí các tác vụ gửi email | |
| **Hàm** | **Mô tả** |
| sendMail(String destination, String title, String content) | Thực hiện chức năng gửi email có nội dung *content*, với tiêu đề là *title* đến địa chỉ *destination* |

*Bảng 4.9: Mô tả lớp RegisterController-ca sử dụng Đăng kí*

|  |  |
| --- | --- |
| **RegisterController**  *Lớp Controller xử lí yêu cầu đăng kí* | |
| ***Hàm*** | ***Mô tả*** |
| *register(String param)* | Hàm xử lí yêu cầu đăng kí từ người dùng,nhận đầu vào là một *@RequestParam String param*(chuỗi JSON lưu thông tin đăng kí) |

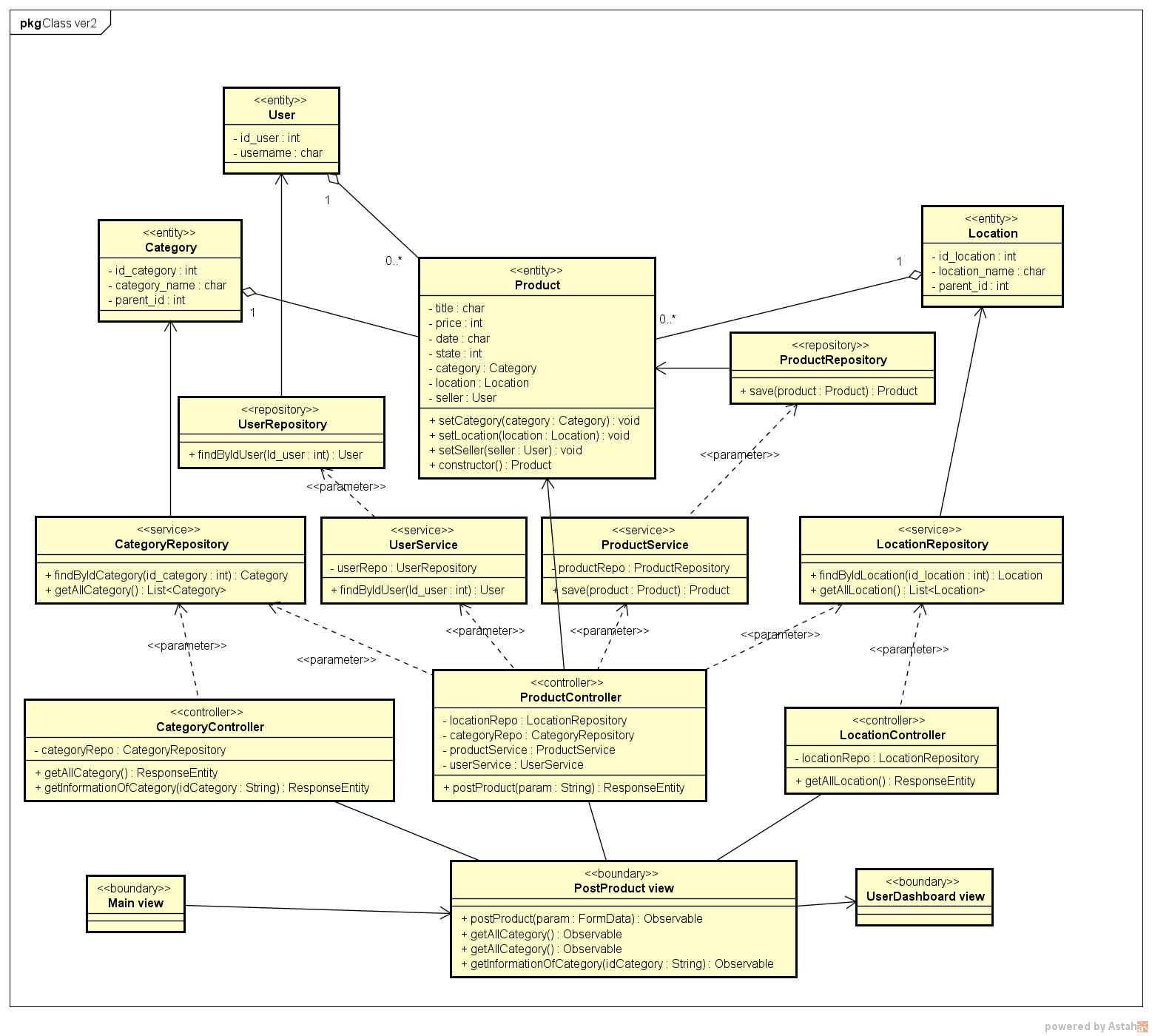
### Đăng tin

1. Biểu đồ trình tự



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

1. Biểu đồ lớp



*Biểu đồ 6: Biểu đồ lớp Đăng tin*

*Bảng 4.10: Mô tả lớp Category-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 |

*Bảng 4.11: Mô tả lớp Location-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 4.12: Mô tả lớp Product-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 4.13: Mô tả lớp CategoryRepository-ca sử dụng Đăng tin*

|  |  |
| --- | --- |
| **CategoryRepository**  Interface kế thừa từ JpaRepository thực hiện các câu truy vấn 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 4.14: Mô tả lớp LocationRepository-ca sử dụng Đăng tin*

|  |  |
| --- | --- |
| **LocationRepository**  Interface kế thừa từ JpaRepository thực hiện các câu truy vấn 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 4.15: Mô tả lớp UserService-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 4.16: Mô tả lớp ProductService-ca sử dụng Đăng tin*

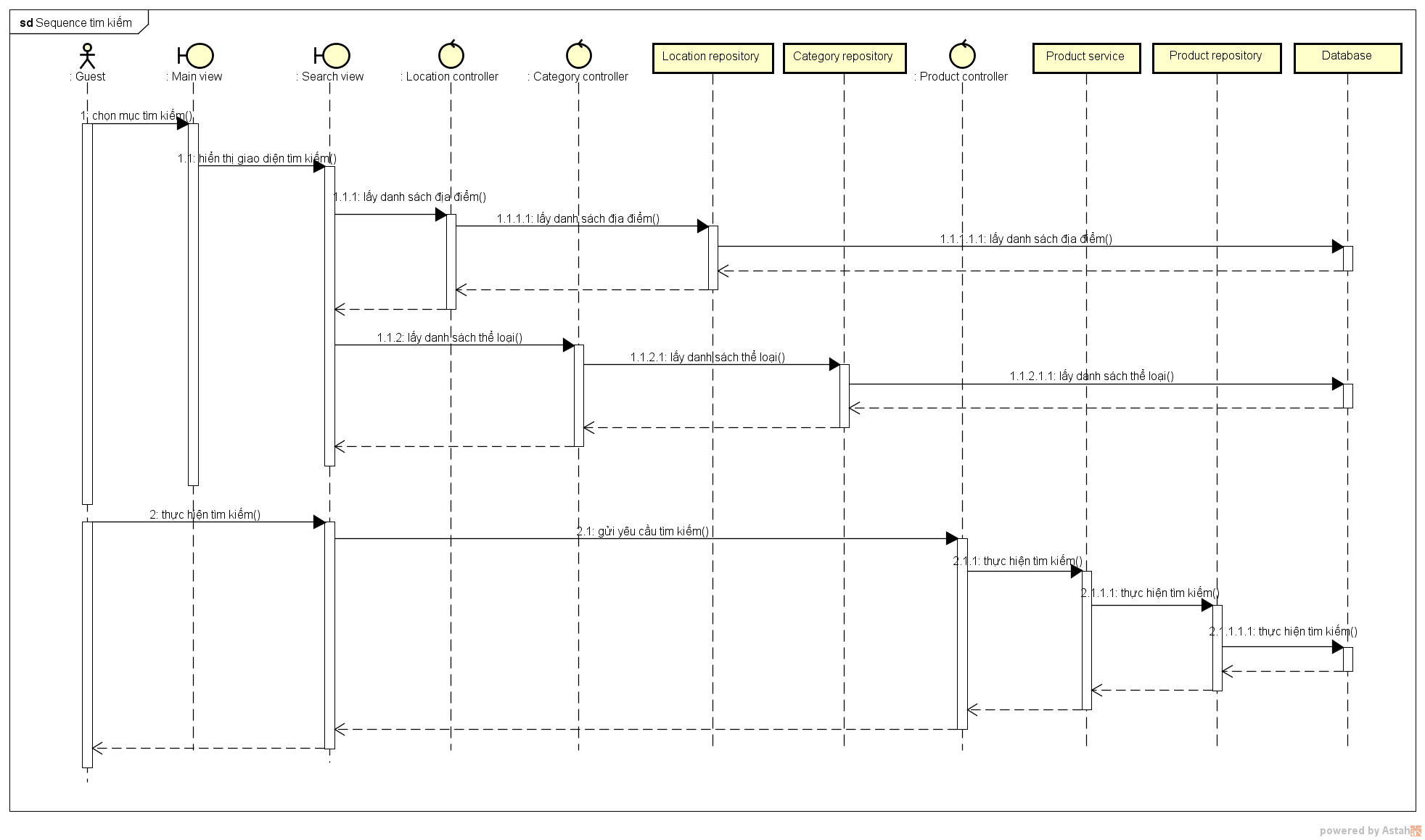
|  |  |
| --- | --- |
| **ProductService**  Lớp service xử lí các tác vụ liên quan đến đối tượng Product(tin đăng) | |
| **Hàm** | **Mô tả** |
| save(Product product) | Lưu đối tượng Product vào database và trả về là một đối tượng Product được lưu thành công |

*Bảng 4.17: Mô tả lớp ProductController-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 một *@RequestParam String param*(chuỗi JSON lưu thông tin của tin đăng) |

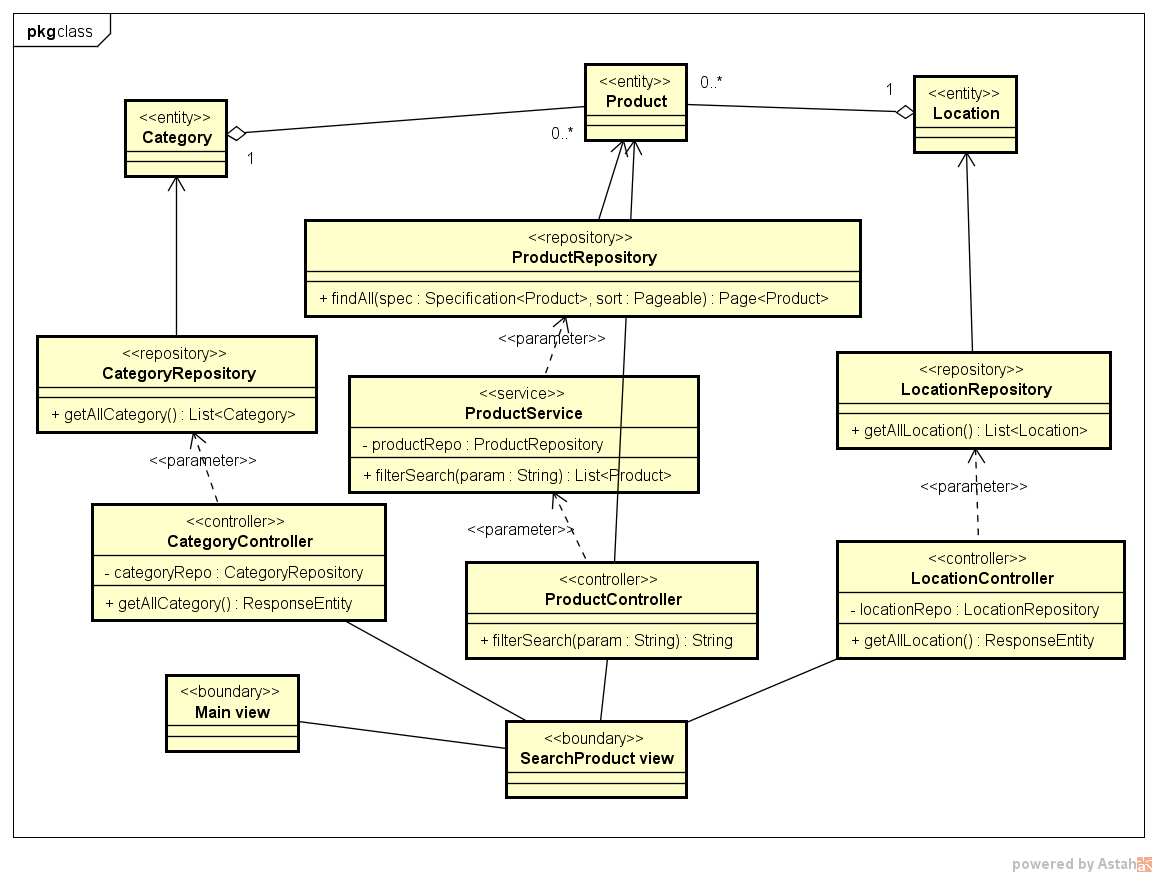
### Tìm kiếm

1. Biểu đồ trình tự



*Biểu đồ 7: Biểu đồ trình tự Tìm kiếm*

1. Biểu đồ lớp



*Biểu đồ 8: Biểu đố lớp Tìm kiếm*

*Bảng 4.18: Mô tả lớp CategoryService-ca sử dụng Tìm kiếm*

|  |  |
| --- | --- |
| **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 4.19: Mô tả lớp LocationService-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 4.20: Mô tả lớp ProductService-ca sử dụng Tìm kiếm*

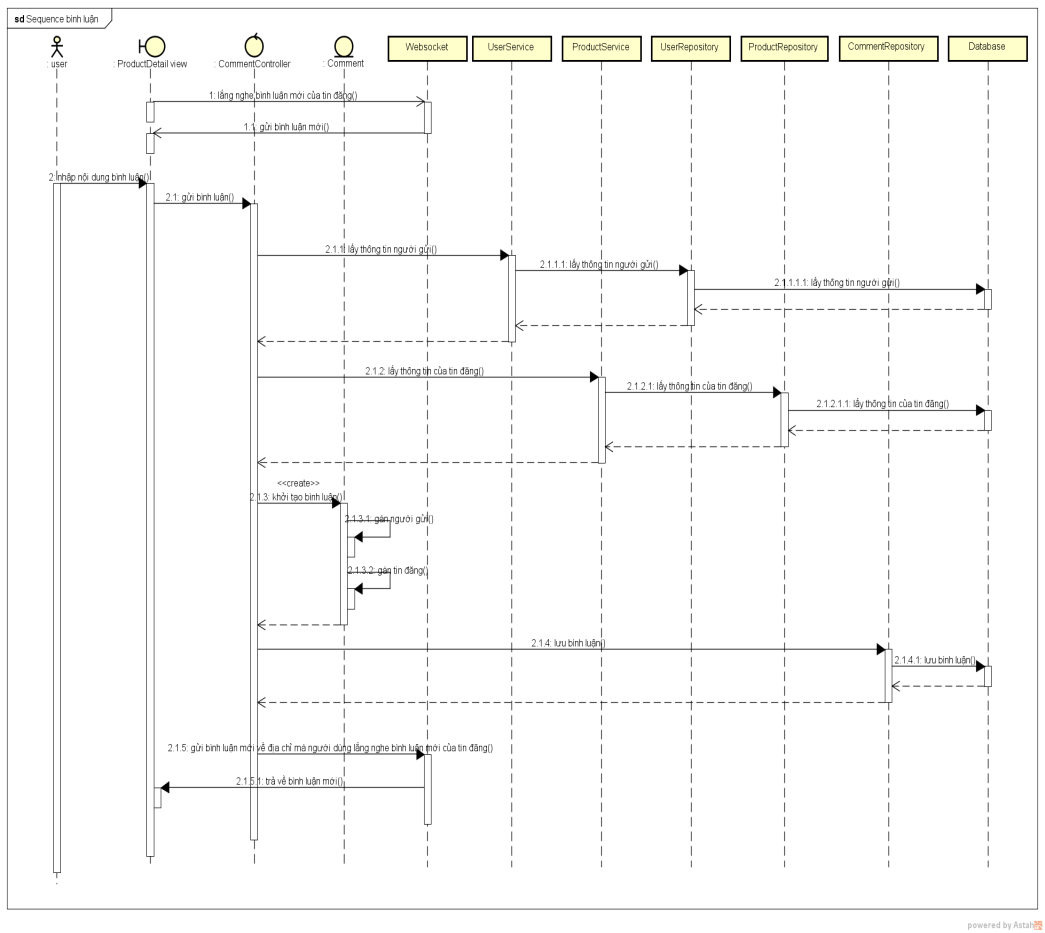
|  |  |
| --- | --- |
| **ProductService**  Lớp service xử lí các tác vụ liên quan đến đối tượng Product(tin đăng) | |
| **Hàm** | **Mô tả** |
| filterSearch(String param) | Nhận đầu vào là xâu JSON( lưu nội dung tìm kiếm của người dùng) , trả về danh sách Product |

*Bảng 4.21: Mô tả lớp ProductController-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, nhận đầu vào là một *@RequestParam String param*(chuỗi JSON lưu yêu cầu tìm kiếm của người dùng) |

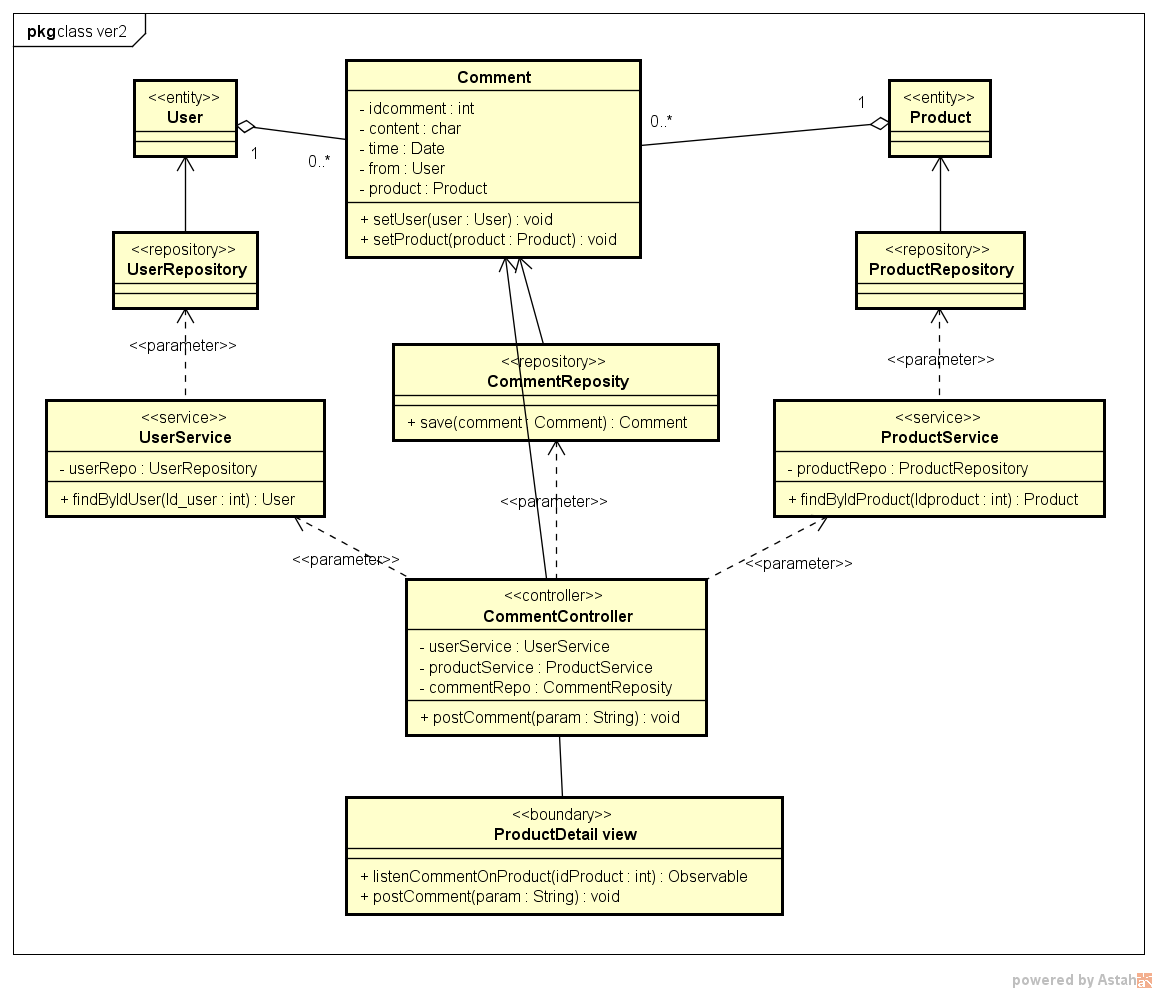
### Bình luận

1. Biểu đồ trình tự



*Biểu đồ 9: Biểu đồ trình tự Bình luận*

1. Biểu đồ lớp



*Biểu đồ 10: Biểu đồ lớp Bình luận*

*Bảng 4.22: Mô tả lớp Comment-ca sử dụng Bình luận*

|  |  |
| --- | --- |
| **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 4.23: Mô tả lớp CommentRepository-ca sử dụng Bình luận*

|  |  |
| --- | --- |
| **CommentRepository**  Interface kế thừa từ JpaRepository thực hiện các câu truy vấn liên quan đến đối tượng Comment | |
| **Hàm** | **Mô tả** |
| save(Comment c) | Lưu đối tượng bình luận vào database |

*Bảng 4.24: Mô tả lớp ProductService-ca sử dụng Bình luận*

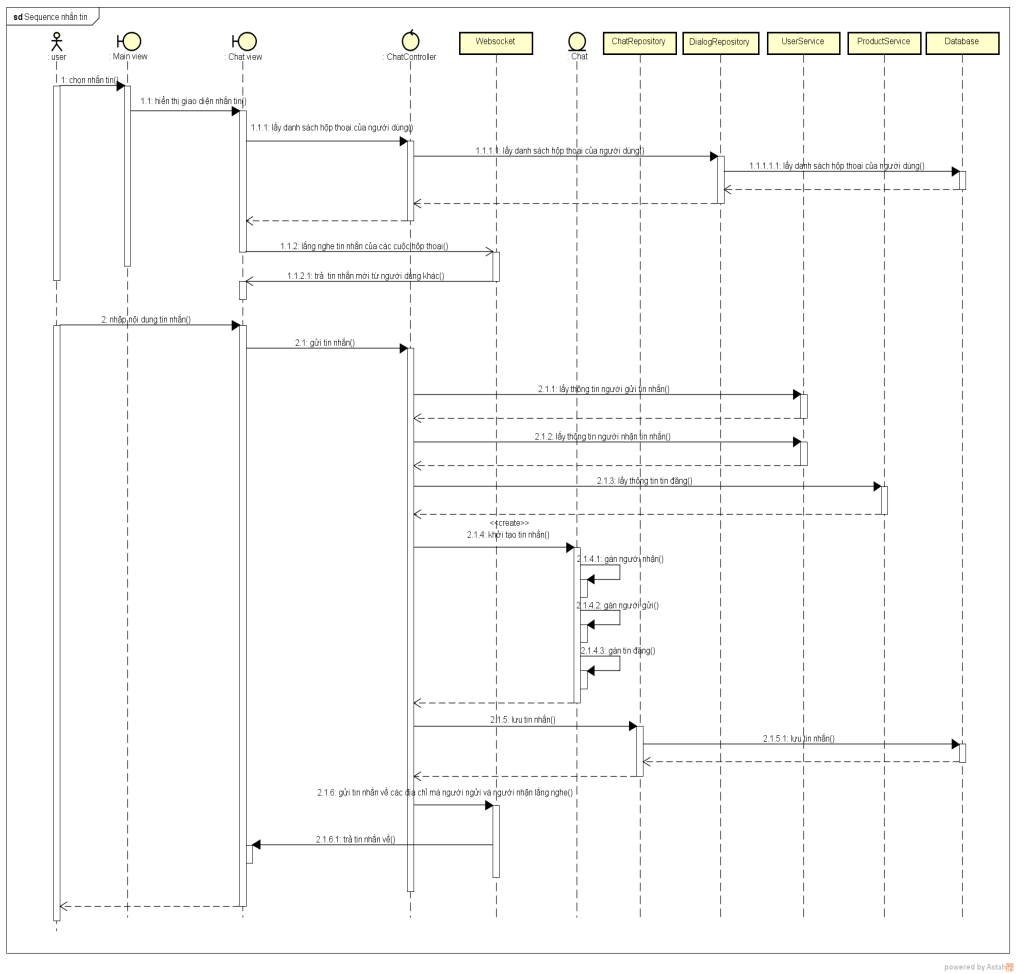
|  |  |
| --- | --- |
| **ProductService**  Lớp service xử lí các tác vụ liên quan đến đối tượng Product(tin đăng) | |
| **Hàm** | **Mô tả** |
| findByIdProduct(id int) | Trả về đối tượng Product có giá trị trường *idproduct* bằng đầu vào *id* |

*Bảng 4.25: Mô tả lớp CommentController-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, nhận đầu vào là một *@RequestParam String param*(chuỗi JSON lưu nội dung bình luận) |

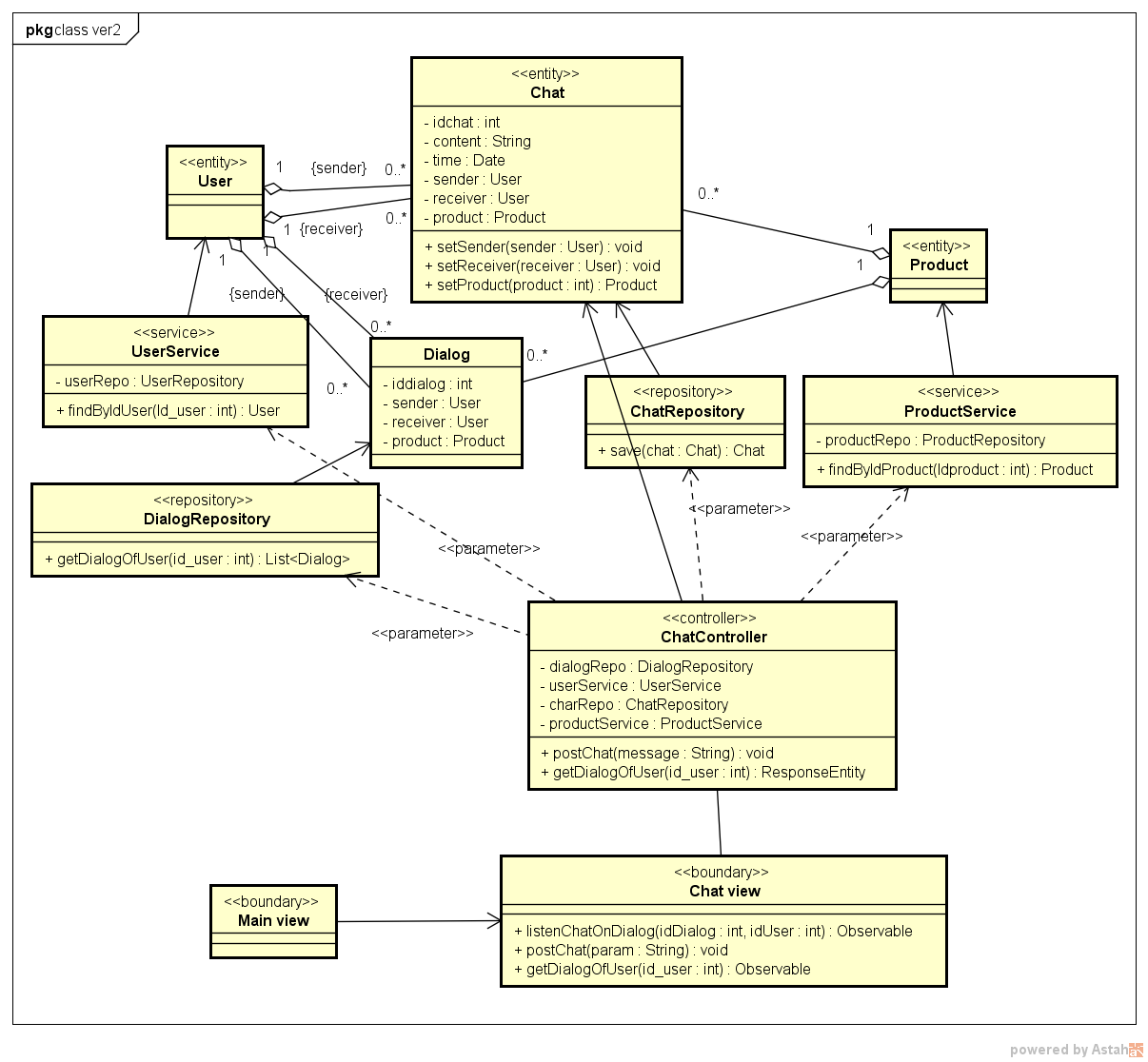
### Nhắn tin

1. Biểu đồ trình tự



*Biểu đồ 11: Biểu đồ trình tự Nhắn tin*

1. Biểu đồ lớp



*Biểu đồ 12: Biểu đồ lớp Nhắn tin*

*Bảng 4.26: Mô tả đối tượng Chat-ca sử dụng Nhắn tin*

|  |  |
| --- | --- |
| **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 4.27: Mô tả đối tượng Dialog-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 4.28: Mô tả lớp ChatRepository-ca sử dụng Nhắn tin*

|  |  |
| --- | --- |
| **ChatRepository**  Interface kế thừa từ JpaRepository thực hiện các câu truy vấn liên quan đến đối tượng Chat | |
| **Hàm** | **Mô tả** |
| save(Chat c) | Lưu đối tượng tin nhắn vào database |

*Bảng 4.29: Mô tả lớp DialogRepository-ca sử dụng Nhắn tin*

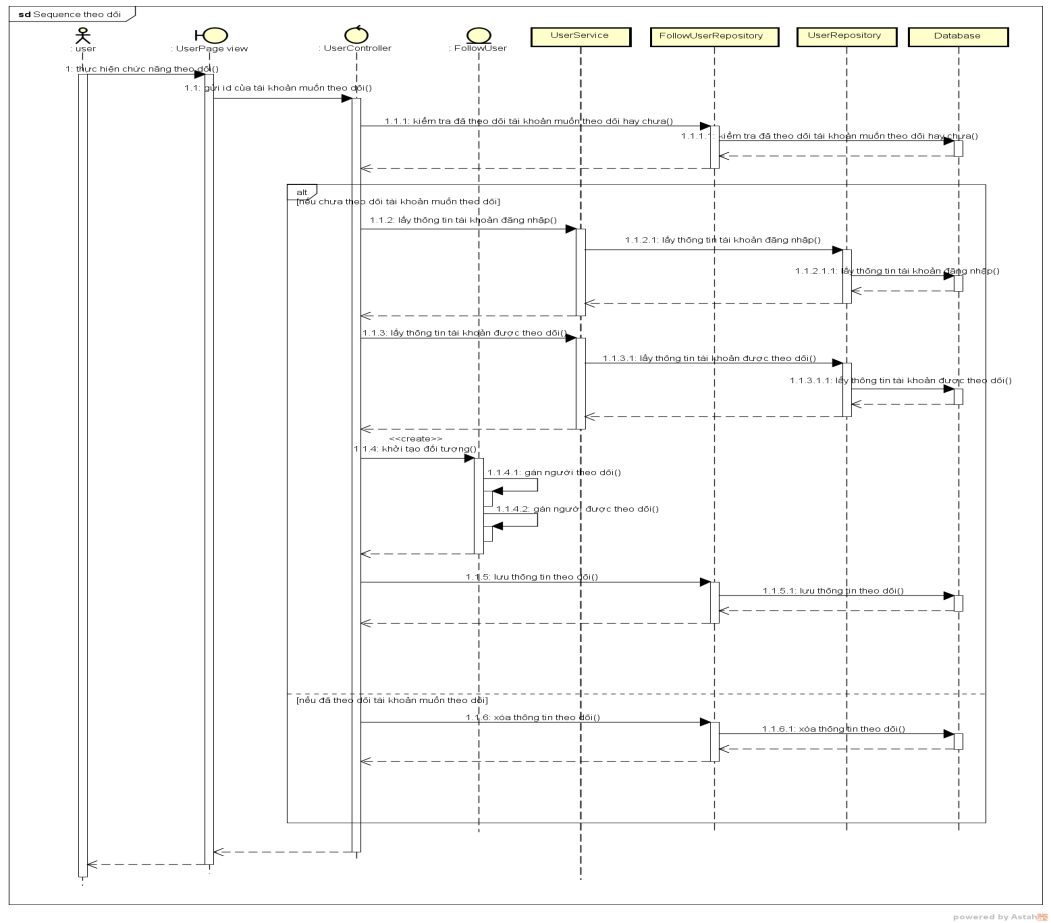
|  |  |
| --- | --- |
| **DialogRepository**  Interface kế thừa từ JpaRepository thực hiện các câu truy vấn liên quan đến đối tượng Dialog | |
| **Hàm** | **Mô tả** |
| getDialogOfUser(id\_user int) | Trả về danh sách đối tượng Dialog có giá trị trường *sender* hoặc *receiver* bằng đầu vào *id\_user* |

*Bảng 4.30: Mô tả lớp ChatController-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 một *@PathVariable int id\_user*(khóa chính) |

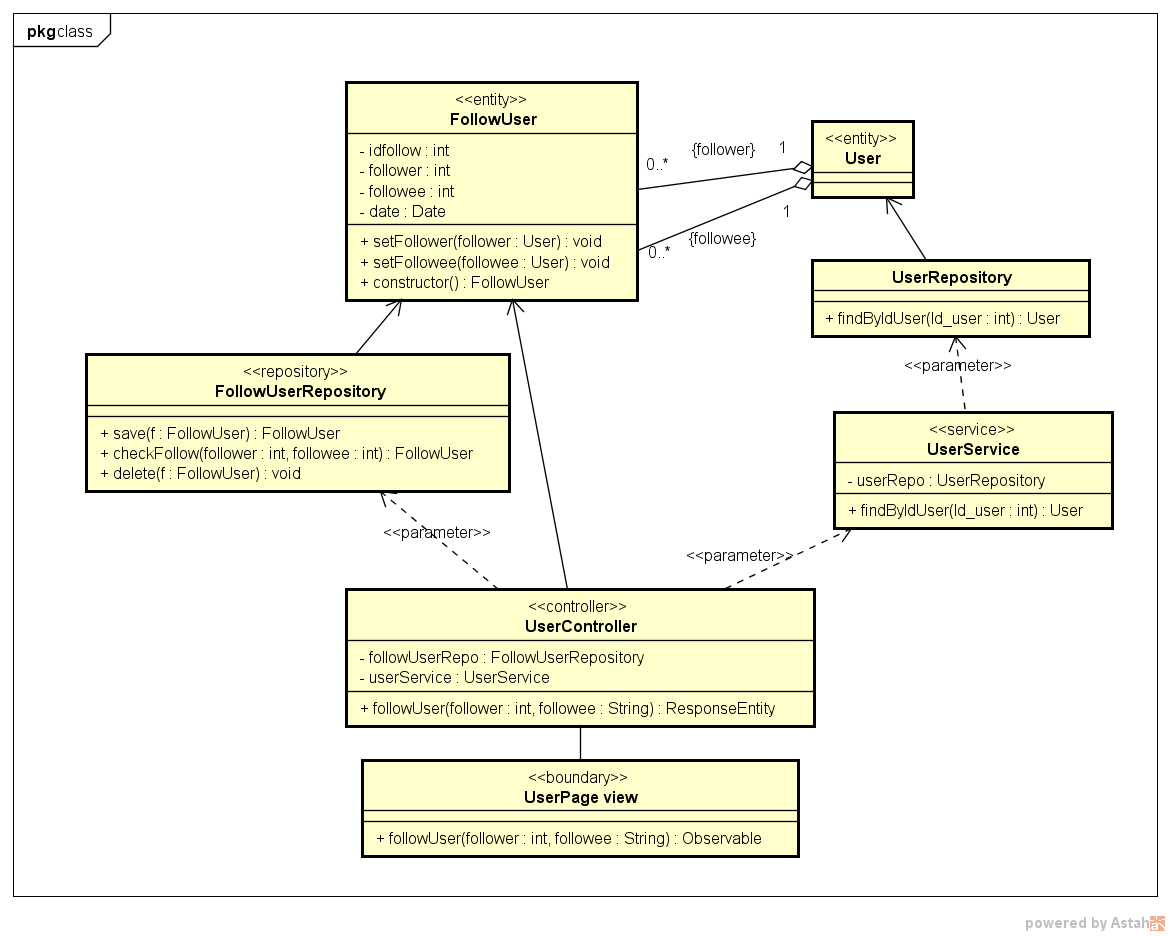
### Theo dõi tài khoản

1. Biểu đồ trình tự



*Biểu đồ 13: Biểu đồ trình tự Theo dõi tài khoản*

1. Biểu đồ lớp



*Biểu đồ 14: Biểu đồ lớp Theo dõi tài khoản*

*Bảng 4.31: Mô tả lớp FollowUser-ca sử dụng Theo dõi tài khoản*

|  |  |
| --- | --- |
| **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 4.32: Mô tả lớp FollowUserRepository-ca sử dụng Theo dõi tài khoản*

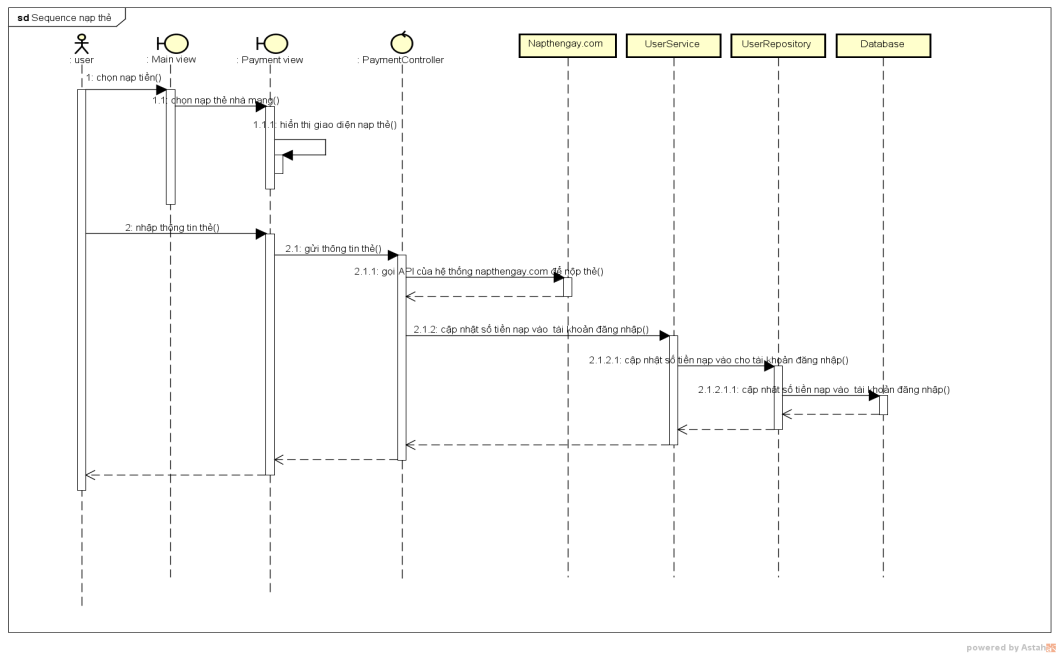
|  |  |
| --- | --- |
| **FollowUserRepository**  Interface kế thừa từ JpaRepository thực hiện các câu truy vấn liên quan đến đối tượng FollowUser | |
| **Hàm** | **Mô tả** |
| save(FollowUser c) | Lưu đối tượng FollowUser vào database |
| checkFollow(int follower, int followee) | Kiểm tra tài khoản 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 4.33: Mô tả lớp UserController-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) |

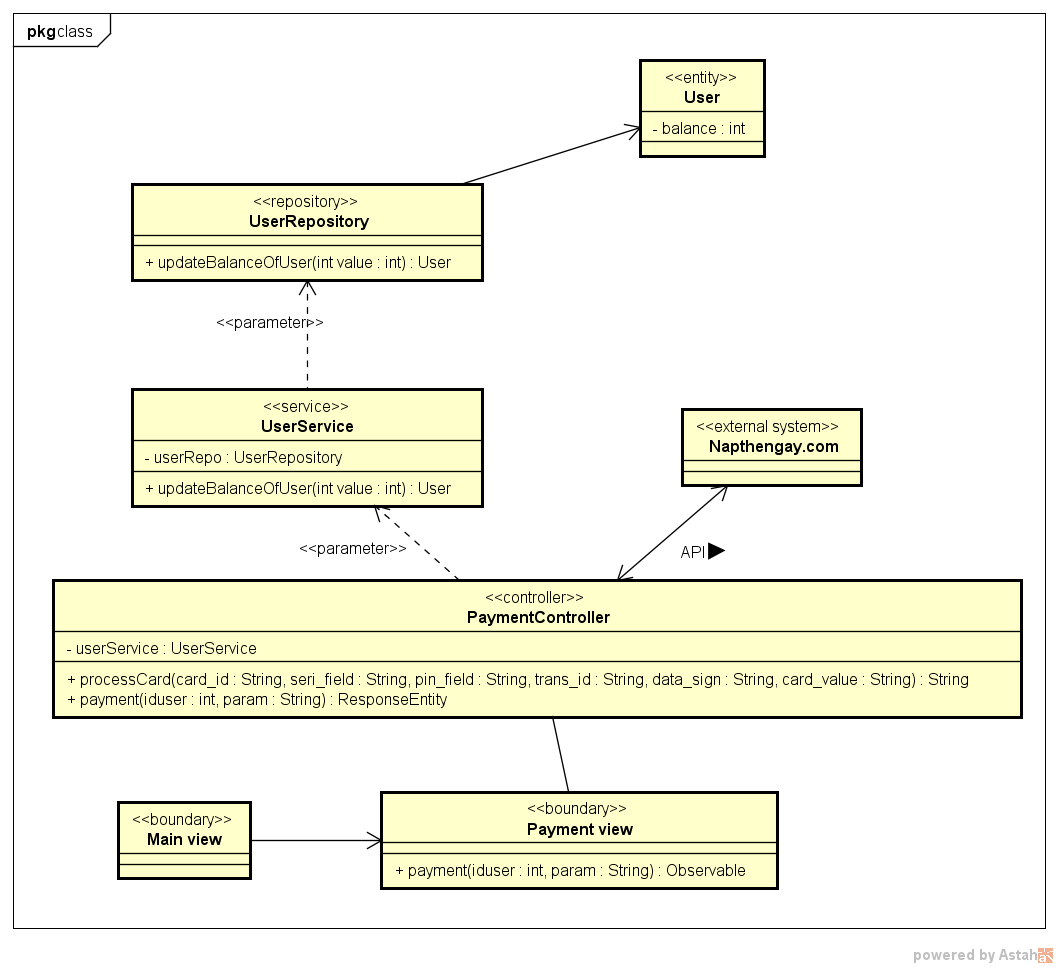
### Nạp thẻ cào

1. Biểu đồ trình tự



*Biểu đồ 15: Biểu đồ trình tự Nạp thẻ cào*

1. Biểu đồ lớp



*Biểu đồ 16: Biểu đồ lớp Nạp thẻ cào*

*Bảng 4.34: Mô tả lớp User-ca sử dụng Nạp thẻ cào*

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

*Bảng 4.35: Mô tả lớp UserRepository- ca sử dụng Nạp thẻ cào*

|  |  |
| --- | --- |
| **UserRepository**  Interface kế thừa từ JpaRepository thực hiện các câu truy vấn liên quan đến đối tượng User | |
| **Hàm** | **Mô tả** |
| updateBalanceOfUser(int balance) | Cập nhật giá trị balance |

*Bảng 4.36: Mô tả lớp UserService- ca sử dụng Nạp thẻ cào*

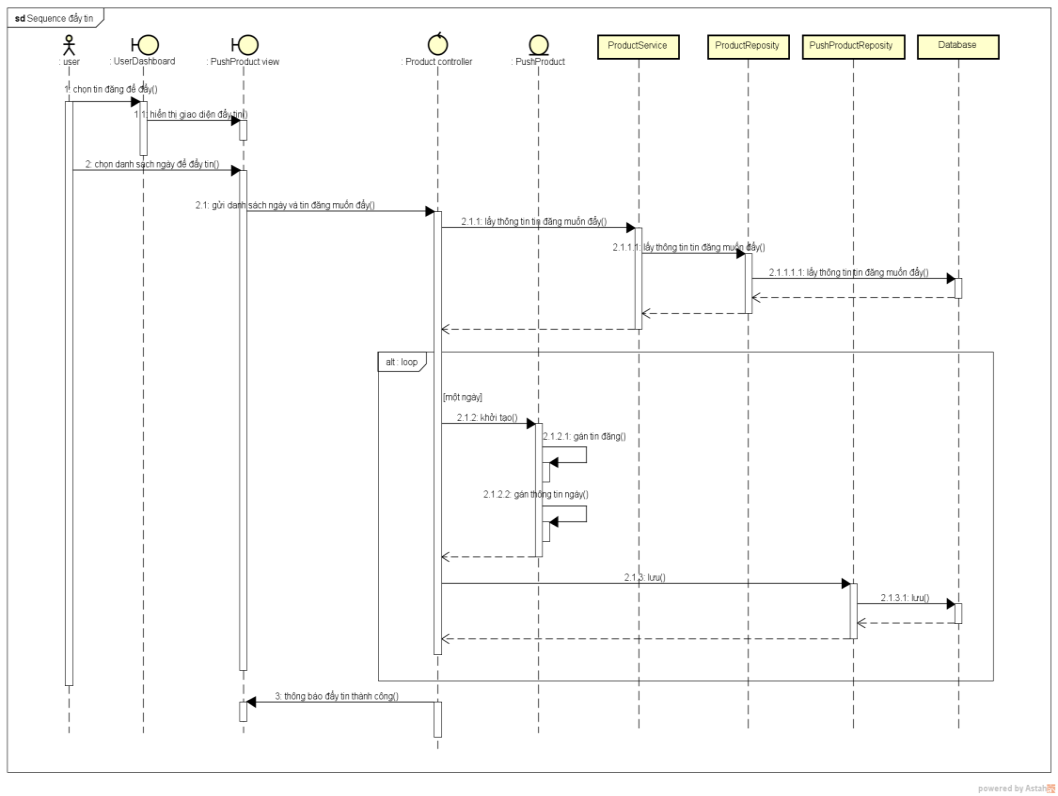
|  |  |
| --- | --- |
| **UserService**  Lớp service xử lí các tác vụ liên quan đến đối tượng User | |
| **Hàm** | **Mô tả** |
| updateBalanceOfUser(int balance) | Cập nhật giá trị balance |

*Bảng 4.37: Mô tả lớp UserController-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) |

### Đẩy tin đăng

1. Biểu đồ trình tự



*Biểu đồ 17:Biểu đồ trình tự Đẩy tin*

1. Biểu đồ lớp



*Biểu đồ 18: Biểu đồ lớp Đẩy tin*

*Bảng 4.38: Mô tả lớp PushProduct-ca sử dụng Đẩy tin*

|  |  |
| --- | --- |
| **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 4.39: Mô tả lớp PushProductRepository-ca sử dụng Đẩy tin*

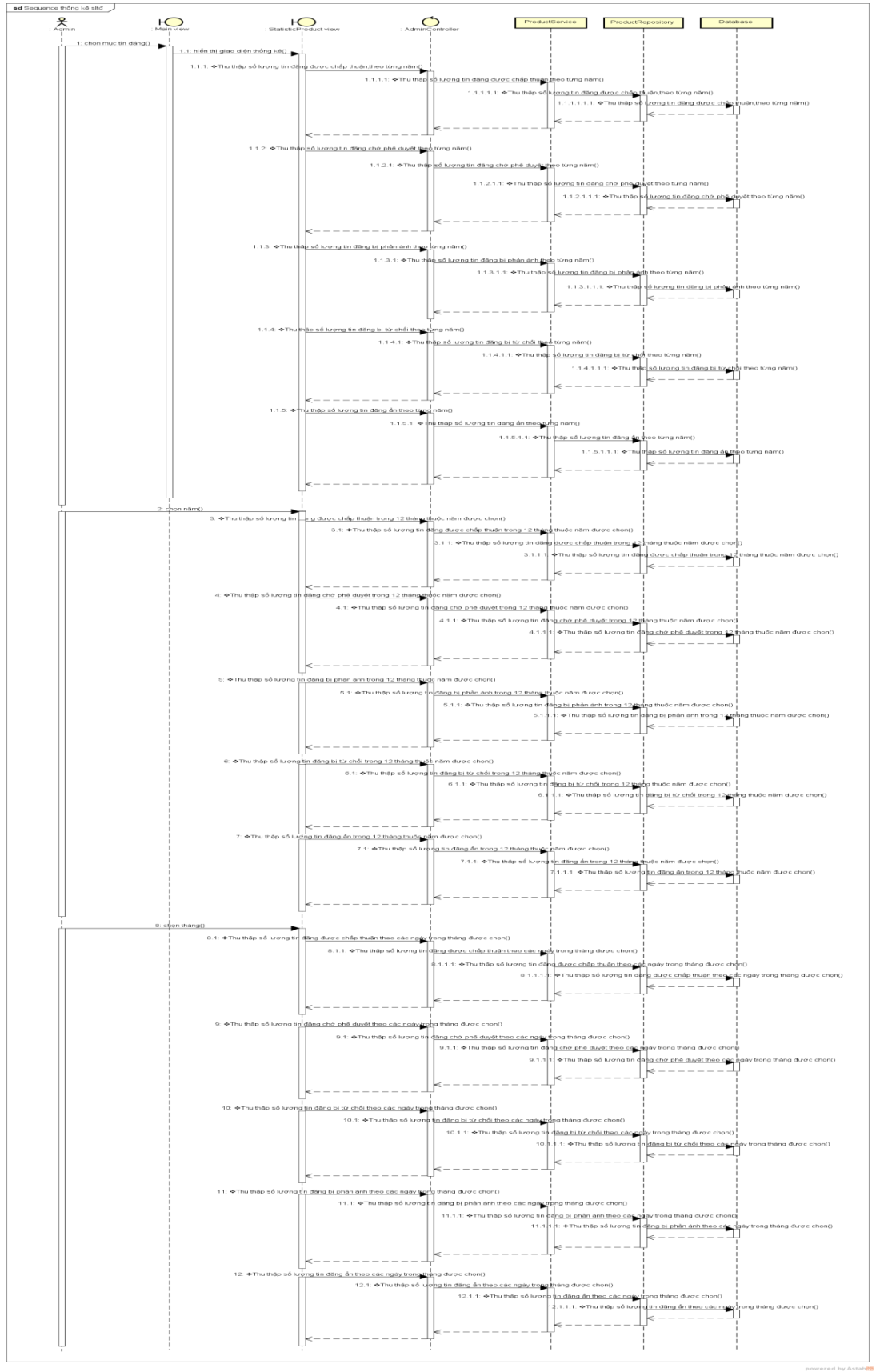
|  |  |
| --- | --- |
| **PushProductRepository**  Interface kế thừa từ JpaRepository thực hiện các câu truy vấn liên quan đến đối tượng PushProduct | |
| **Hàm** | **Mô tả** |
| save(PushProduct  p) | Lưu đối tượng PushProduct vào database |

*Bảng 4.40: Mô tả lớp ProductController- 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) |

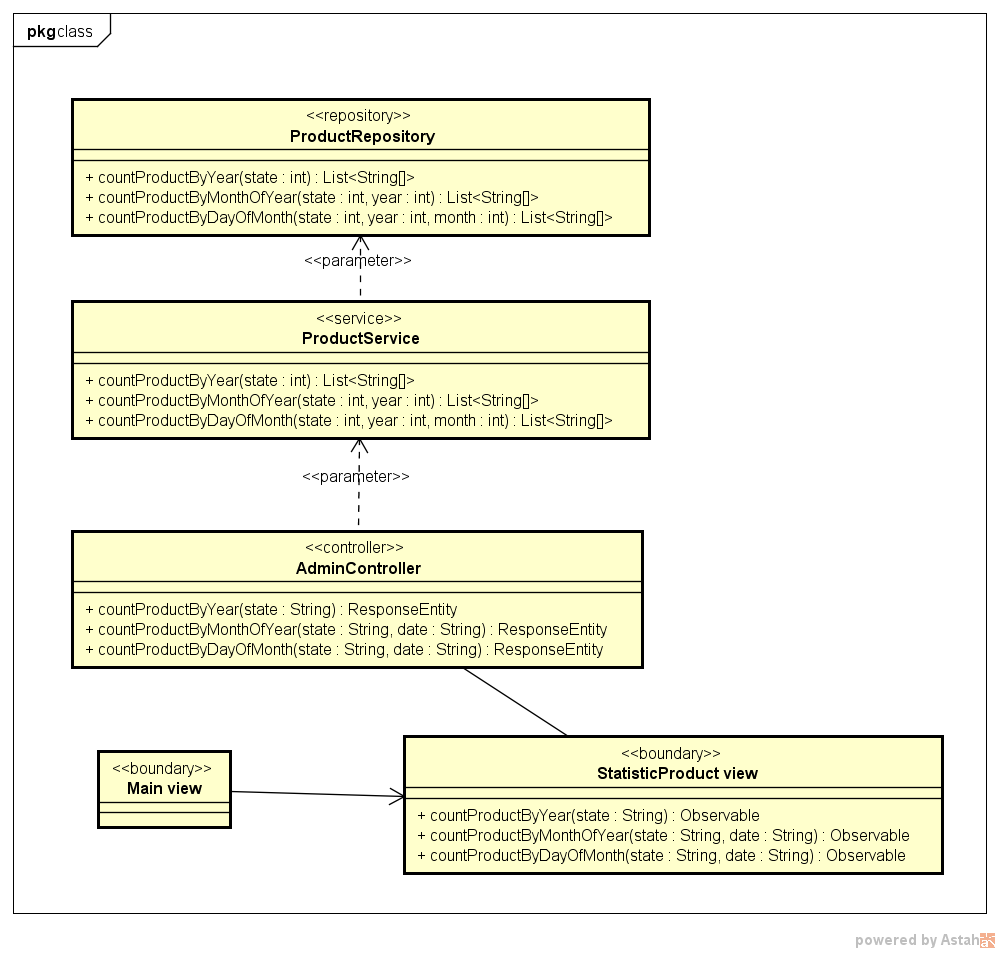
### Thống kê số lượng tin đăng

1. Biểu đồ trình tự



*Biểu đồ 19: Biểu đồ trình tự Thống kê số lượng tin đăng*

1. Biểu đồ lớp



*Biểu đồ 20: Biểu đồ lớp Thống kê số lượng tin đăng*

*Bảng 4.41: Mô tả lớp ProductRepository-ca sử dụng Thống kê số lượng tin đăng*

|  |  |
| --- | --- |
| **ProductRepository**  Interface kế thừa từ JpaRepository thực hiện các câu truy vấn liên quan đến đối tượng Product | |
| **Hàm** | **Mô tả** |
| countProductByYear(int state) | Đếm số lượng tin đăng trong các năm ( đầu vào là giá trị trạng thái của tin đăng) |
| countProductByMonthOfYear(int state, int year) | Đếm số lượng tin đăng trong 12 tháng của một năm bất kì ( đầu vào là giá trị trạng thái của tin đăng và năm) |
| countProductByDayOfMonth(int state , int year, int month) | Đếm số lượng tin đăng mỗi ngày của một tháng bất kì ( đầu vào là giá trị trạng thái của tin đăng, năm và tháng) |

*Bảng 4.42: Mô tả lớp ProductService-ca sử dụng Thống kê số lượng tin đăng*

|  |  |
| --- | --- |
| **ProductService**  Lớp service xử lí các tác vụ liên quan đến đối tượng Product | |
| **Hàm** | **Mô tả** |
| countProductByYear(int state) | Đếm số lượng tin đăng trong các năm , đầu vào là giá trị trạng thái của tin đăng.  -*state=0* là tin đăng đang chờ phê duyệt  -*state=1* là tin đăng đã được chấp thuận  -*state=2* là tin đăng bị từ chối  -*state=3* là tin đăng bị phản ánh  -*state=4* là tin đăng đã ẩn |
| countProductByMonthOfYear(int state, int year) | Đếm số lượng tin đăng trong 12 tháng của một năm bất kì ( đầu vào là giá trị trạng thái của tin đăng và năm) |
| countProductByDayOfMonth(int state , int year, int month) | Đếm số lượng tin đăng mỗi ngày của một tháng bất kì ( đầu vào là giá trị trạng thái của tin đăng, năm và tháng) |

*Bảng 4.43: Mô tả lớp AdminController- ca sử dụng Thống kê số lượng tin đăng*

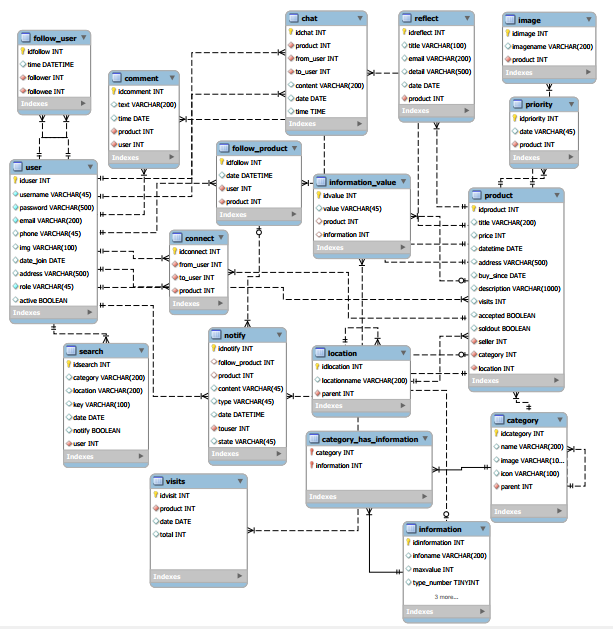
|  |  |
| --- | --- |
| **AdminController**  Lớp Controller xử lí các yêu cầu từ người dùng liên quan đến tác vụ dành cho Admin | |
| **Hàm** | **Mô tả** |
| countProductByYear(String state) | Phương thức GET đếm số lượng tin đăng trong các năm, nhận đầu vào là một *@RequesrParam String state*(trạng thái tin đăng) |
| countProductByMonthOfYear(String state, String date) | Đếm số lượng tin đăng trong 12 tháng của một năm bất kì , nhận đầu vào là một *@RequesrParam String state*(trạng thái tin đăng) và *@RequesrParam String date(* ngày*)* |
| countProductByDayOfMonth(String state, String date) | Đếm số lượng tin đăng mỗi ngày của một tháng bất kì ,nhận đầu vào là *@RequesrParam String state*(trạng thái tin đăng) và *@RequesrParam String date(* ngày*)* |

## Thiết kế cơ sở dữ liệu

Sau đây là tổng quan về cơ sở dữ liệu của hệ thống và mô tả chi tiết các bảng cơ sở dữ liệu.

### Sơ đồ quan hệ

Cơ sở dữ liệu của hệ thống được biểu diễn như hình 7 dưới đây.



*Hình 7: Quan hệ giữa các bảng cơ sở dữ liệu*

### Mô tả cơ sở dữ liệu

Dưới đây là mô tả chi tiết về một số bảng cơ sở dữ liệu quan trọng trong hệ thống.

1. Bảng cơ sở dữ liệu User

Bảng User lưu trữ thông tin tài khoản của người dùng

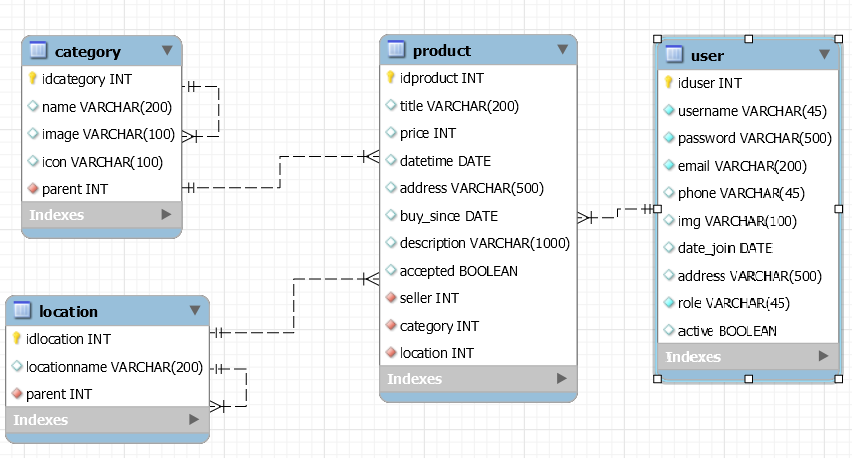
*Bảng 4.44: Bảng cơ sở dữ liệu User*

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

1. Bảng cơ sở dữ liệu Product (Tin đăng)

Bảng Product lưu trữ thông tin của tin đăng, một tin đăng sẽ thuộc một thể loại, một địa điểm và một người bán.

Liên kết giữa Category và Product là một-nhiều trên khóa ngoài “category”, liên kết giữa Location và Product là một-nhiều trên khóa ngoài “location”, liên kết giữa User và Product là một-nhiều trên khóa ngoài “seller”.



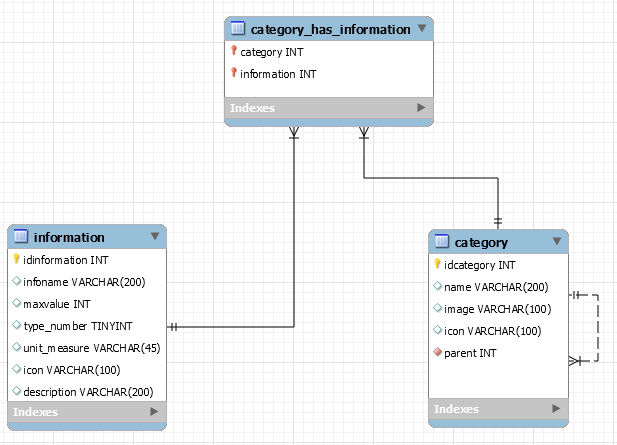
*Hình 8: Quan hệ giữa bảng Product với Category và Location*

*Bảng 4.45: Bảng cơ sở dữ liệu Product*

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

1. Bảng cơ sở dữ liệu Category

Bảng Category mô tả thể loại của tin đăng , một thể loại có thể trực thuộc thể loại khác,“parent” là khóa ngoài liên kết tới bảng Category.



*Hình 9: Quan hệ giữa bảng Category và Information*

*Bảng 4.46: Bảng cơ cở dữ liệu Category*

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

1. Bảng cơ sở dữ liệu Information

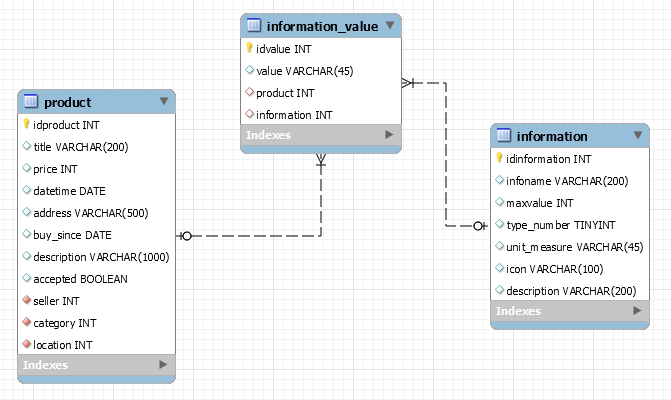
Mỗi thể loại (Category) sẽ có nhiều thuộc tính khác nhau và có thể được cập nhật trong tương lai, không thể biểu diễn hết tất cả thuộc tính trong bảng Category , vì vậy bảng Information sẽ mô tả thuộc tính của Category, quan hệ giữa cả hai là nhiều-nhiều và được thể hiện qua bảng Category\_has\_Information

*Bảng 4.47: Bảng cơ sở dữ liệu 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 |

1. Bảng cơ sở dữ liệu Information\_value

Bảng Information\_value lưu trữ giá trị thuộc tính của tin đăng (Product), một tin đăng sẽ có nhiều thuộc tính (Information), quan hệ giữa Product và Information là nhiều-nhiều được biểu diễn bởi bảng Information\_value.



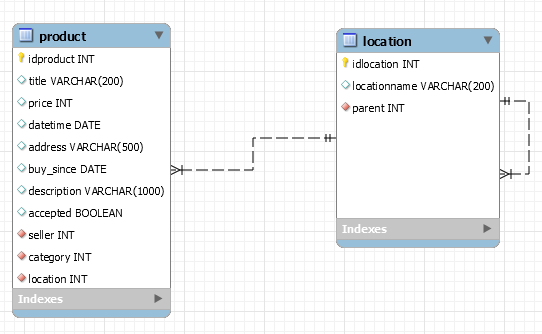
*Hình 10: Quan hệ giữa Product và Information*

*Bảng 4.48: Bảng cơ sở dữ liệu Information\_value*

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

1. 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



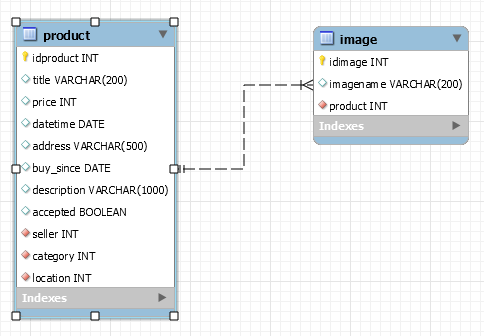
*Hình 11: Quan hệ giữa Product và Location*

*Bảng 4.49: Bảng cơ sở dữ liệu 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 |

1. 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 12: Quan hệ giữa Product và Image*

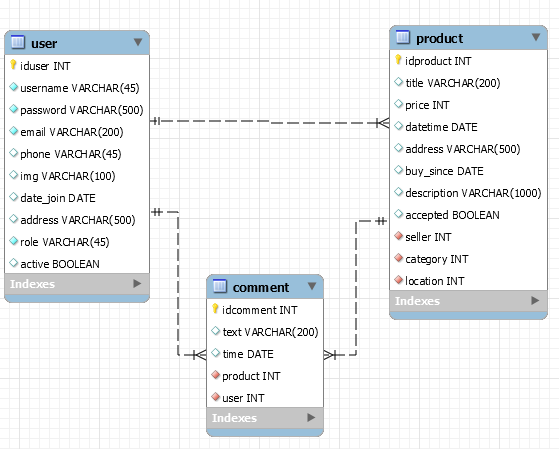
*Bảng 4.50: Bảng cơ sở dữ liệu Image*

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

1. Bảng cơ sở dữ liệu Comment

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

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



*Hình 13: Quan hệ giữa Comment với Product và User*

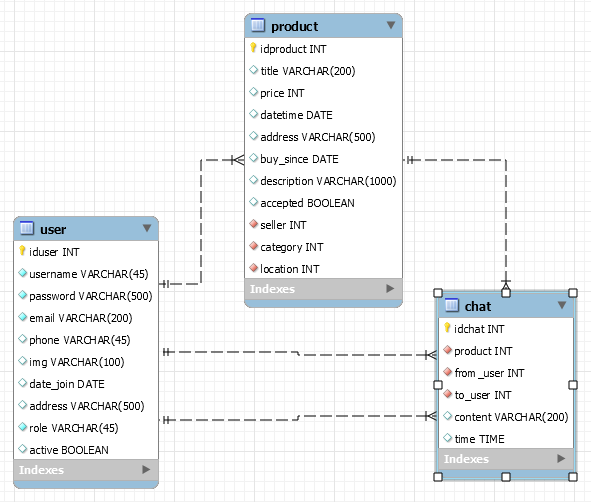
*Bảng 4.51: Bảng cơ sở dữ liệu Comment*

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

1. Bảng cơ sở dữ liệu Chat

Bảng Chat lưu trữ nội dung tin nhắn liên quan đến một tin đăng giữa hai người dùng.

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 14: Quan hệ giữa Chat với User và Product*

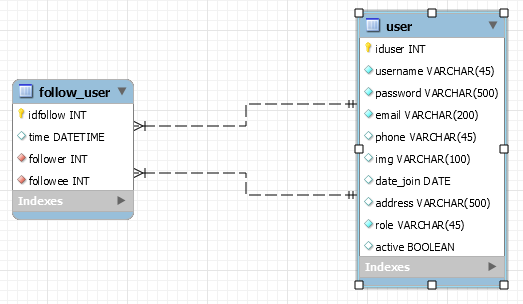
*Bảng 4.52: Bảng cơ sở dữ liệu Chat*

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

1. Bảng cơ sở dữ liệu Follow\_User

Bảng Follow\_User lưu trữ thông tin theo dõi giữa người dùng.

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 15: Quan hệ giữa User và Follow\_user*

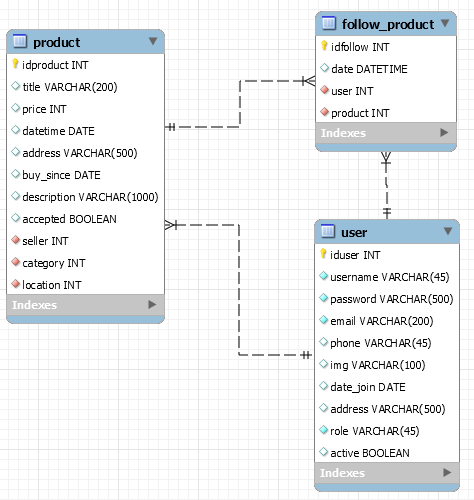
*Bảng 4.53: Bảng cơ sở dữ liệu Follow\_User*

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

1. 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 16: Quan hệ giữa Follow\_Product với Product và User*

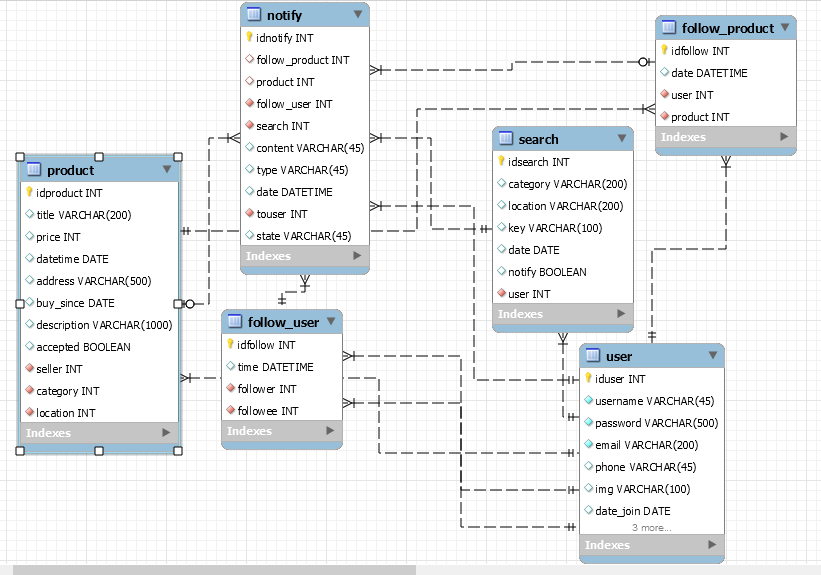
*Bảng 4.54: Bảng cơ sở dữ liệu Follow\_Product*

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

1. 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 17: Quan hệ giữa Notify với các bảng khác*

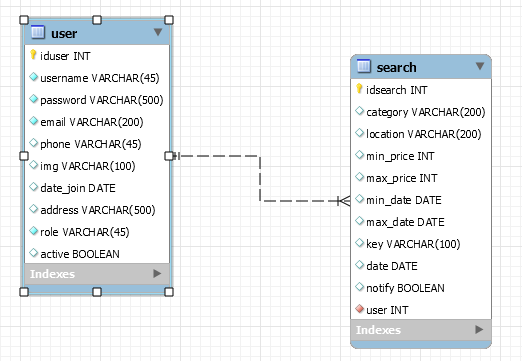
*Bảng 4.55: Bảng cơ sở dữ liệu Notify*

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

1. 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 18: Quan hệ giữa Search và User*

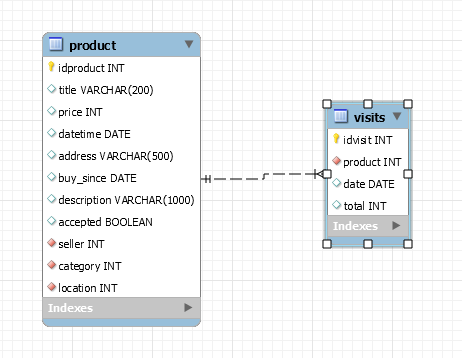
*Bảng 4.56: Bảng cơ sở dữ liệu Search*

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

1. 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 Visits trên khóa ngoài “product” là một-nhiều.



*Hình 19: Quan hệ giữa Visits và Product*

*Bảng 4.57: Bảng cơ sở dữ liệu Visits*

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

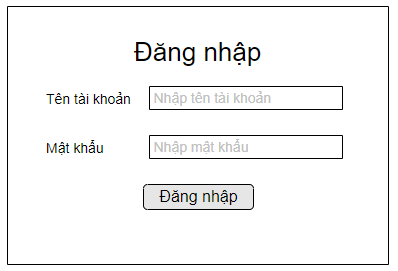
## Thiết kế giao diện

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. Đọc tài liệu số [10] để hiểu thêm về Angular Material.
* 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* ***20*** dưới đây minh họa giao diện Đăng nhập.



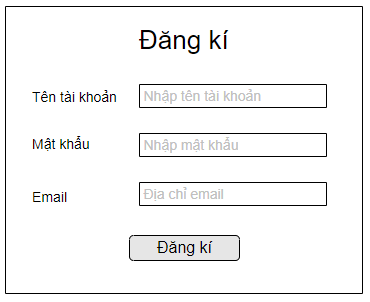
*Hình 20: Minh họa giao diện Đăng nhập*

Người dùng đăng nhập vào hệ thống bằng cách nhập Tên tài khoản và Mật khẩu, sau đó nhấn nút Đăng nhập.

Tên tài khoản không quá 45 kí tự, nếu để trống hoặc số kí tự vượt quá 45, thì hệ thống sẽ thông báo lỗi, tương tự mật khẩu không quá 500 kí tự, nếu để trống hoặc số kí tự vượt quá 500 thì hệ thống sẽ thông báo lỗi. Nếu thông tin đăng nhập không tồn tại , thì hệ thống thông báo lỗi và yêu cầu người dùng đăng nhập lại.

### Giao diện Đăng kí

*Hình* ***21*** dưới đây minh họa giao diện Đăng kí.



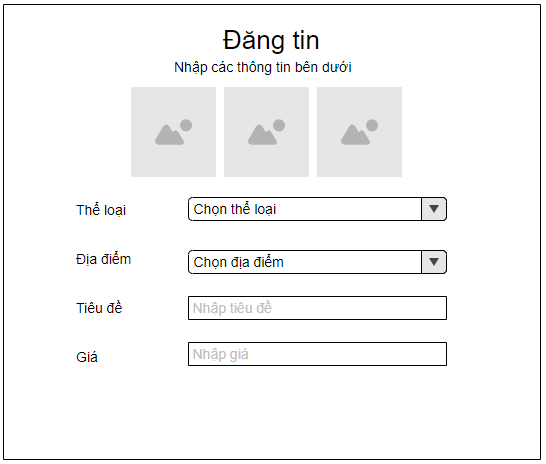
*Hình 21: Minh họa giao diện Đăng kí tài khoản*

Người dùng khách đăng kí tài khoản bằng cách nhập Tên tài khoản, Mật khẩu và địa chỉ Email, sau đó nhấn nút Đăng kí.

Tên tài khoản không quá 45 kí tự, nếu để trống hoặc số kí tự vượt quá 45, thì hệ thống sẽ thông báo lỗi, mật khẩu không quá 500 kí tự, nếu để trống hoặc số kí tự vượt quá 500 thì hệ thống sẽ thông báo lỗi, tương tự địa chỉ Email không được vượt quá 200 kí tự, nếu để trống hoặc số kí tự vượt quá 500 thì hệ thống sẽ thông báo lỗi. Nếu tên tài khoản đăng kí đã tồn tại, thì hệ thống thông báo lỗi và yêu cầu người dùng nhập lại thông tin.

### Giao diện Đăng tin

*Hình* ***22*** dưới đây minh họa giao diện Đăng tin



*Hình 22: Minh họa giao diện Đăng tin*

Người dùng nhập một số thông tin liên quan đến tin đăng như Ảnh sản phẩm, Thể loại tin đăng, Địa điểm giao hàng, Tiêu đề của tin đăng, Giá sản phẩm muốn bán…, sau đó nhấn nút Đăng tin.

Tiêu đề không được nhập quá 200 kí tự, nếu để trống hoặc nhập vượt quá 200 kí tự, hệ thống sẽ thông báo lỗi.

Giá sản phẩm chỉ chấp nhận định dạng kiểu số, nếu người dùng nhập nội dung không phải kiểu số, hệ thống sẽ thông báo lỗi, yêu cầu nhập đúng định dạng.

### Giao diện Tìm kiếm

*Hình* ***23*** dưới đây minh họa giao diện Tìm kiếm.

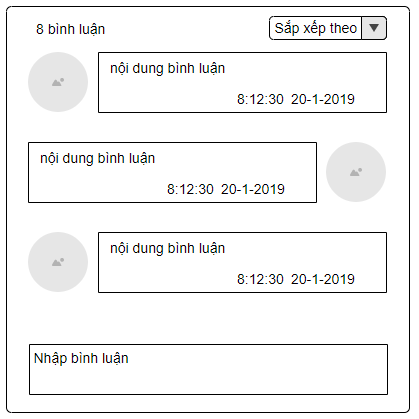


*Hình 23: Minh họa giao diện Tìm kiếm*

Người dùng có thể thực hiện tìm kiếm bằng cách nhập nội dung tìm kiếm, tìm kiếm theo Địa điểm, theo Thể loại, lọc kết quả tìm kiếm theo thời gian đăng , theo giá sản phẩm và sắp xếp kết quả tìm kiếm tăng dần hoặc giảm giần theo thời gian đăng hoặc giá sản phẩm. Kết quả tìm kiếm sẽ được phân phân trang, mỗi trang hiển thị tối đa 12 sản phẩm.

### Giao diện Bình luận

*Hình* ***24*** dưới đây minh họa giao diện Bình luận



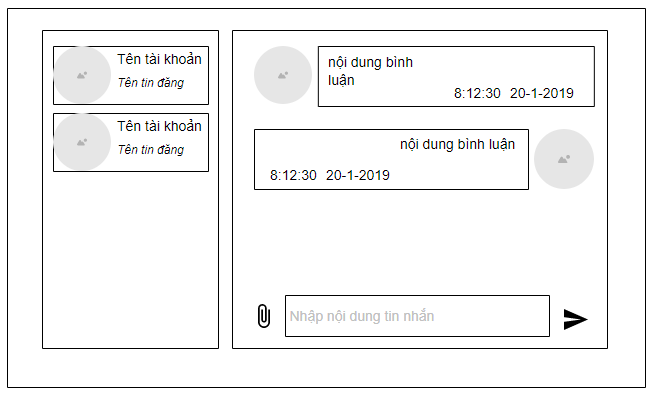
*Hình 24: Minh họa giao diện Bình luận*

Giao diện hiển thị danh sách bình luận mới nhất thuộc tin đăng, mỗi bình luận sẽ hiển thị nội dung bình luận, thời gian và ảnh tài khoản thực hiện bình luận đó, người dùng có thể cuộn chuột lên trên để tải thêm bình luận cũ hơn.

Để gửi bình luận, người dùng nhập nội dung và ấn phím Enter hoặc nhấp vào biểu tượng gửi.

### Giao diện Nhắn tin

*Hình* ***25*** dưới đây minh họa giao diện Nhắn tin



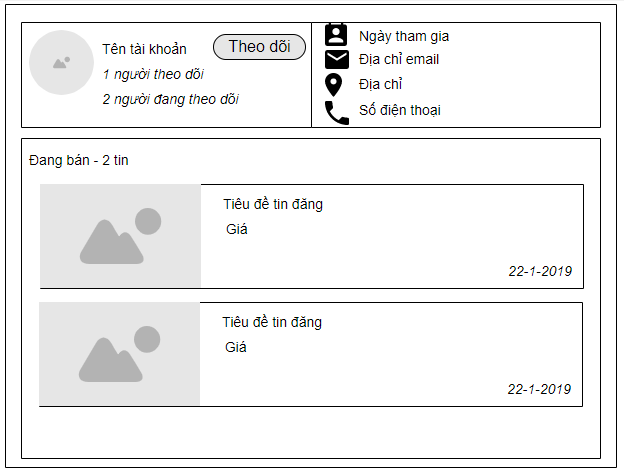
*Hình 25: Minh họa giao diện Nhắn tin*

Giao diện nhắn tin sẽ hiển thị danh sách các hộp thoại mà người dùng đã tương tác trong quá khứ. Khi người dùng nhấp chọn một hộp thoại, hệ thống sẽ hiển thị nội dung 8 tin nhắn mới nhất, mỗi tin nhắn sẽ hiển thị nội dung tin nhắn, thời gian và ảnh người nhắn, người dùng có thể cuộn chuột lên trên để tải thêm tin nhắn cũ hơn.

Để gửi tin nhắn, người dùng nhập nội dung tin nhắn và ấn phím Enter hoặc nhấp vào biểu tượng gửi.

### Giao diện Trang cá nhân

*Hình* ***26*** dưới đây minh họa giao diện Trang cá nhân



*Hình 26: Minh họa giao diện Trang cá nhân*

Khi vào trang cá nhân của một tài khoản khác, người dùng có thể thực hiện theo dõi tài khoản đó, hệ thống sẽ thông báo đến người dùng mỗi khi tài khoản được theo dõi đăng một tin đăng mới. Để thực hiện theo dõi một tài khoản, người dùng nhấp vào nút Theo dõi, hệ thống sẽ thông báo theo dõi thành công nếu thao tác không xảy ra lỗi.

### Giao diện Nạp thẻ cào

*Hình* ***27*** dưới đây minh họa giao diện Trang cá nhân

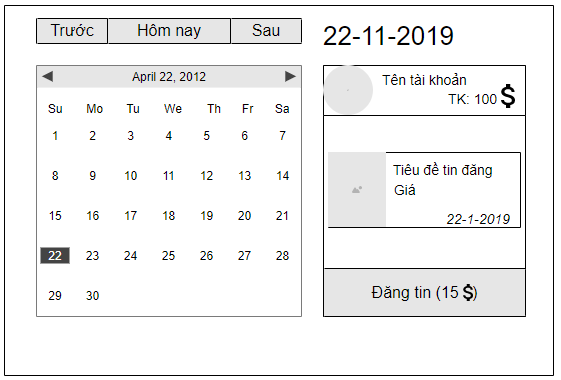


*Hình 27: Minh họa giao diện Nạp thẻ cào*

Người dùng nạp thẻ cào vào hệ thống bằng cách nhập chọn loại thẻ (Viettel, vinaphone hoặc mobifone), mã thẻ và số seri của thẻ, sau đó nhấp nút Nạp thẻ.

### Giao diện Đẩy tin

*Hình* ***28*** dưới đây minh họa giao diện Đẩy tin đăng

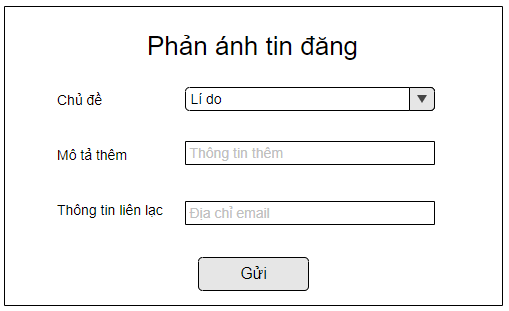


*Hình 28: Minh họa giao diện Đẩy tin đăng*

Giao diện đẩy tin sẽ hiển thị thông tin Tin đăng mà người dùng muốn đẩy thành sản phẩm ưu tiên và calendar. Để thực hiện đẩy tin, đầu tiên người dùng nhấp chọn các ngày để đẩy tin đăng, sau đó nhấp nút Đăng tin. Nếu người dùng chọn ngày trước thời điểm hiện tại, hệ thống sẽ thông báo lỗi. Nếu số tiền trong tài khoản ít hơn số tiền để đẩy tin đăng, hệ thống sẽ thông báo lỗi.

### Giao diện Phản ánh tin đăng

*Hình* ***29*** dưới đây minh họa giao diện Phản ánh tin đăng

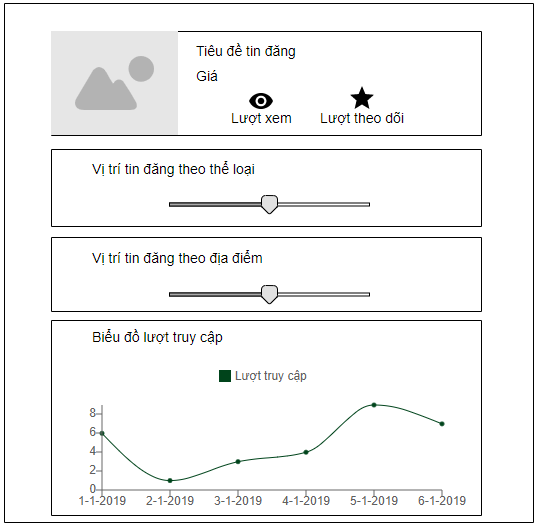


*Hình 29: Minh họa giao diện Phản ánh tin đăng*

Để thực hiện phản ánh một tin đăng, người dùng cần cung cấp các thông tin sau đây : lý do phản ánh, mô tả thêm và thông tin liên lạc, sau đó nhấn nút Gửi.

### Giao diện Thống kê lượt truy cập tin đăng

*Hình* ***30*** dưới đây minh họa giao diện Thống kê lượt truy cập tin đăng



*Hình 30: Minh họa giao diện Thống kê lượt truy cập của tin đăng*

Các thông tin thống kê về một tin đăng bao gồm tổng số lượt truy cập, lượt theo dõi, vị trí tin đăng hiện tại theo thể loại và địa điểm, biểu đồ lượt truy cập biểu diễn lượt xem của các ngày trong tháng hiện tại.

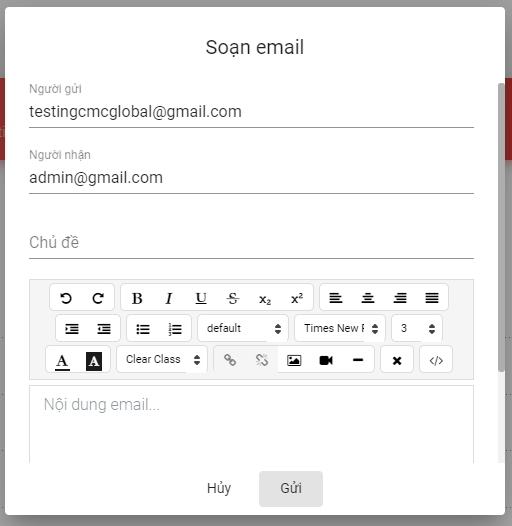
### Giao diện Phê duyệt tin đăng

*Hình* ***31*** dưới đây minh họa giao diện Phê duyệt tin đăng



*Hình 31: Minh họa giao diện Phê duyệt tin đăng*

Giao diện Phê duyệt tin đăng sẽ hiển thị thông tin của Tin đăng và thông tin của người bán. Để chấp thuận một tin đăng rao bán lên hệ thống, người quản trị nhấp nút Chấp thuận, ngược lại nhấp nút Từ chối. Người quản trị có thể thực hiện chức năng gửi Email cho người bán bằng cách nhấp nút Gửi thông báo, *Hình 32* dưới đây minh họa giao diện gửi mail:



*Hình 32:* *Minh họa giao diện Soạn Email*

Để gửi một Email, người quản trị chỉ cần nhập Nội dung email và Chủ đề của email, sau đó nhấp nút Gửi.

### Giao diện Thống kê số lượng tin đăng

*Hình* ***33*** dưới đây minh họa giao diện Thống kê số lượng tin đăng



*Hình 33: Minh họa giao diện Thống kê số lượng tin đăng*

Biểu đồ thống kê sẽ biểu diễn số lượng tin đăng bị từ chối, tin đăng đã chấp thuận, tin đăng chờ phê duyệt theo năm hoặc các tháng trong một năm hoặc các ngày trong một tháng.

# XÂY DỰNG VÀ THỬ NGHIỆM

Chương 5 trình bày về việc thiết lập môi trường để chạy ứng dụng website Chợ đồ cũ và các kết quả đạt được khi triển khai ứng dụng.

## Môi trường xây dựng

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

### Cài đặt hệ thống

Sau khi hoàn tất việc cài đặt môi trường, bước tiếp theo là tiến hành cài đặt hệ thống. Hệ thống gồm 2 chương trình riêng biệt là backend và frontend. Để thực hiện cài đặt, cần clone chương trình từ trang github với tài khoản xác thực là

*Username: trinhvantai*

*Password : trinhvantai2007*

* Cài đặt chương trình backend

B1:chạy câu lệnh sau để thực hiện clone chương trình backend: *git clone* [*https://github.com/trinhvantai/MuaBan.git*](https://github.com/trinhvantai/MuaBan.git)

B2: import chương trình vào eclipse và chạy

* Cài đặt chương trình frontend dành cho user

B1: chạy câu lệnh sau để thực hiện clone chương trình frontend dành cho user: *git clone* [*https://github.com/trinhvantai/docu.git*](https://github.com/trinhvantai/docu.git)

B2: vào thư mục code vừa tải về và chạy câu lệnh sau bằng Command Prompt: *npm i,* để cài đặt các thư viện cần thiết .

B3: chạy câu lệnh sau bằng Command Prompt *: ng serve –o* để start chương trình.

* Cài đặt chương trình frontend dành cho admin

B1: chạy câu lệnh sau để thực hiện clone chương trình frontend dành cho admin: *git clone* [*https://github.com/trinhvantai/admin.git*](https://github.com/trinhvantai/admin.git)

B2: vào thư mục code vừa tải về và chạy câu lệnh sau bằng Command Prompt: *npm i,* để cài đặt các thư viện cần thiết.

B3: chạy câu lệnh sau bằng Command Prompt *: ng serve -o --port 4201* để start chương trình.

B4: đăng nhập với tài khoản có tên đăng nhập là “ad” và mật khẩu là “ad”

* Kết nối database

B1: chạy câu lệnh sau để thực hiện clone database

*git clone* [*https://github.com/trinhvantai/database.git*](https://github.com/trinhvantai/database.git)

B2: tạo database có tên là chotot bằng workbench

B3: import file chotot.sql vào database vừa mới tạo

B4: thay đổi các trường thông tin sau đây thuộc fileapplication.properties của chương trình backend tương ứng với username và password khi thực hiện cài đặt môi trường MySQL.

*spring.datasource.username*

*spring.datasource.password*

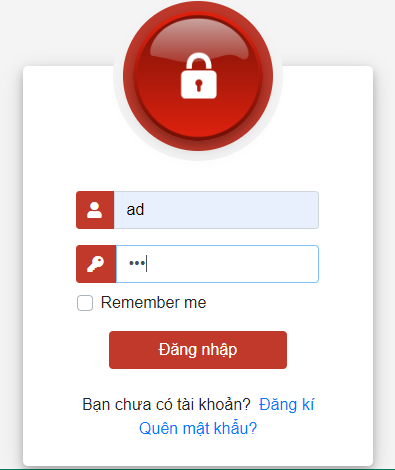
## Các kết quả thử nghiệm

Sau khi đã tiến hành phân tích và thiết kế hệ thống , dưới đây là một số kết quả chính được tiến hành với các môi trường đã lựa chọn

### Đăng nhập

B1: truy cập vào hệ thống thông qua đường dẫn [*http://localhost:4200/*](http://localhost:4200/)

B2: nhấp chọn nút Đăng nhập trên thanh điều hướng, kết quả sẽ hiển thị như *Hình* ***34*** dưới đây:



*Hình 34: Giao diện Đăng nhập*

Thực hiện đăng nhập với tài khoản có tên tài khoản là “ad” và mật khẩu là “ad”. Nếu thông tin đăng nhập không hợp lệ hoặc tài khoản chưa được kích hoạt, hệ thống sẽ thông báo lỗi,ngược lại kết quả sau khi đăng nhập thành công sẽ điều hướng tới trang chủ như *Hình* ***35*** dưới đây:



*Hình 35: Giao diện Trang chủ*

### Tìm kiếm

B1: truy cập vào hệ thống thông qua đường dẫn [*http://localhost:4200/*](http://localhost:4200/)

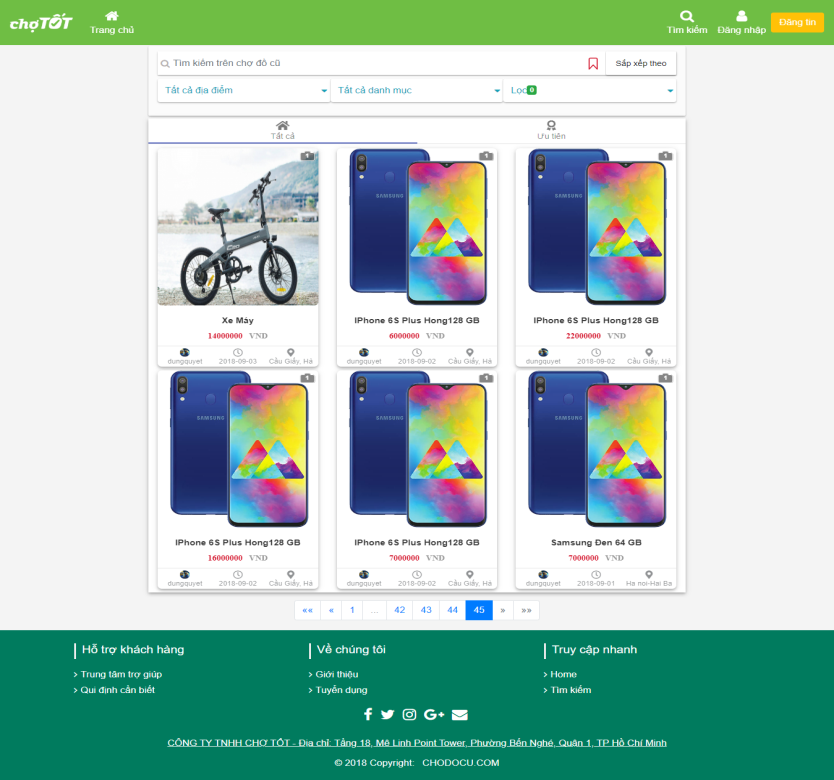
B2:Sau khi vào hệ thống, người dùng nhấp vào biểu tượng tìm kiếm, kết quả hiển thị là trang đầu tiên của danh sách sản phẩm đang rao bán trên hệ thống và danh sách các trang, như *Hình 36*.

B3: chọn địa điểm, *Hình* ***37***

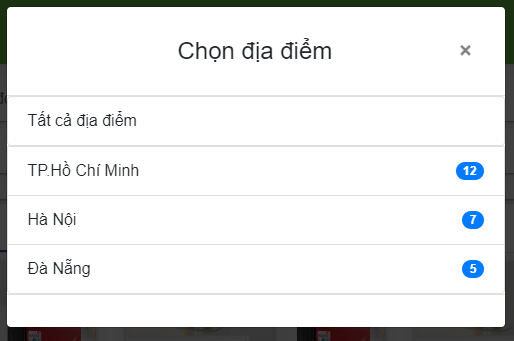
B4: chọn danh mục, *Hình* ***38***

B5: lọc kết quả tìm kiếm

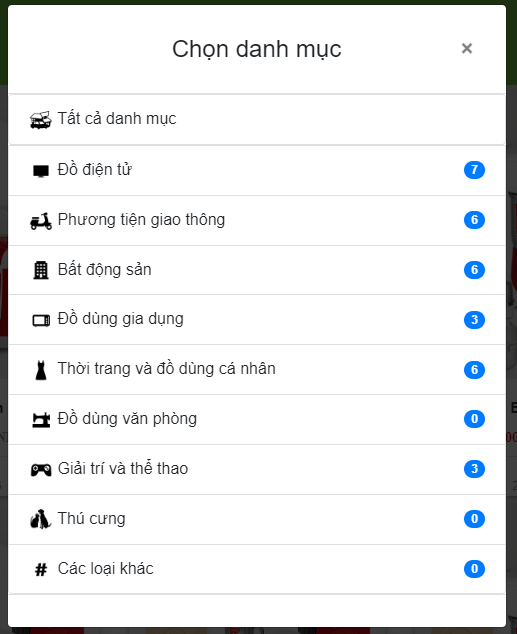
B6: sắp xếp kết quả tìm kiếm



*Hình 36: Giao diện Tìm kiếm*



*Hình 37: Giao diện modal Chọn địa điểm*



*Hình 38: Giao diện modal Chọn danh mục*

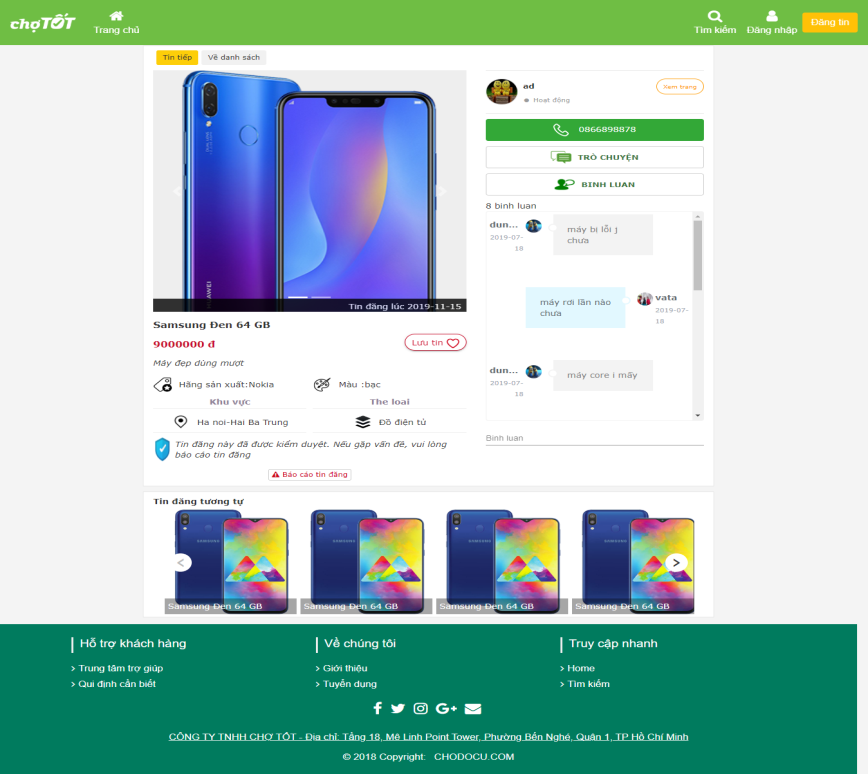
### Nhắn tin

Để vào giao diện nhắn tin, người dùng có thể nhấp chọn biểu tượng Trò chuyện hoặc từ giao diện Chi tiết tin đăng hoặc từ phần header của ứng dụng.

B1: truy cập vào hệ thống thông qua đường dẫn [*http://localhost:4200/*](http://localhost:4200/)

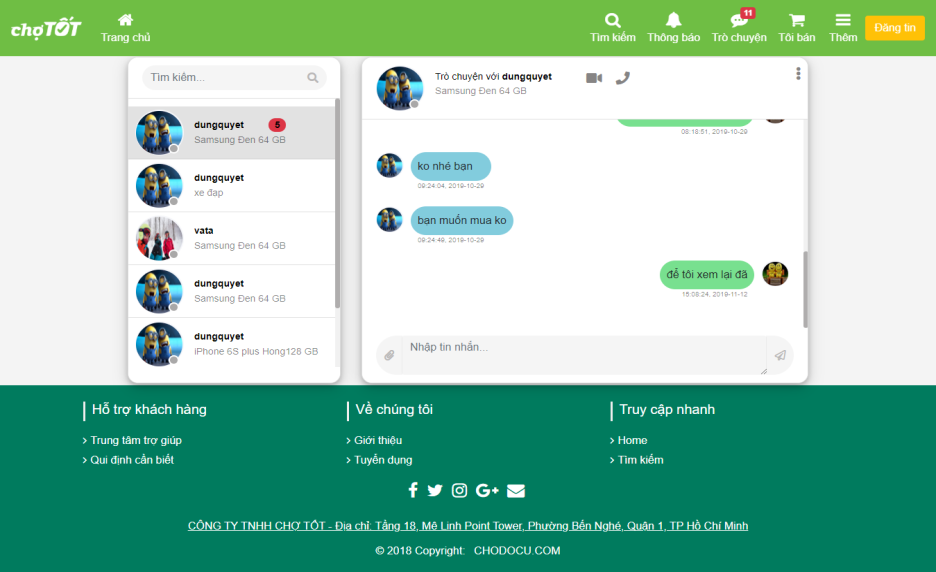
B2: vào trang tìm kiếm

B3: nhấp chọn một tin đăng bất kì, hệ thống sẽ hiển thị các thông tin chi tiết của một tin đăng. *Hình* ***39*** dưới đây là giao diện chi tiết của một tin đăng:



*Hình 39: Giao diện Chi tiết tin đăng*

B4: nhấp chọn “Trò chuyện”, hệ thống sẽ điều hướng tới trang nhắn tin như *Hình* ***40*** dưới đây, tại đây sẽ hiển thị các hộp thoại trò chuyện trong quá khứ và trạng thái của hộp thoại đó ( người dùng khác có đang sử dụng hệ thống hay không)



*Hình 40: Giao diện Nhắn tin*

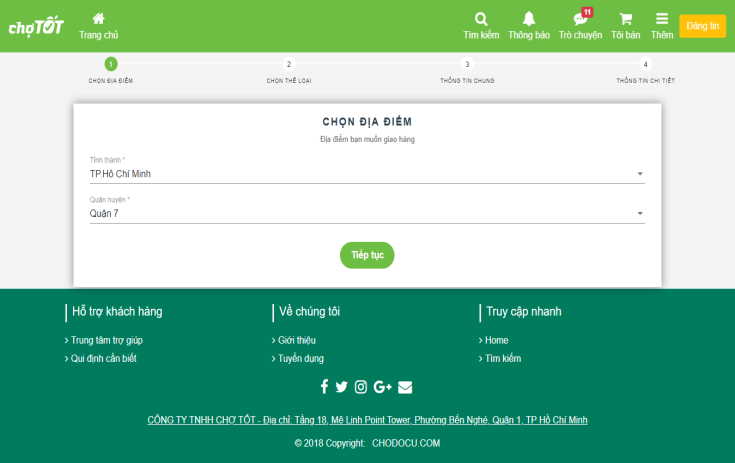
### Đăng tin

B1: truy cập vào hệ thống thông qua đường dẫn [*http://localhost:4200/*](http://localhost:4200/)

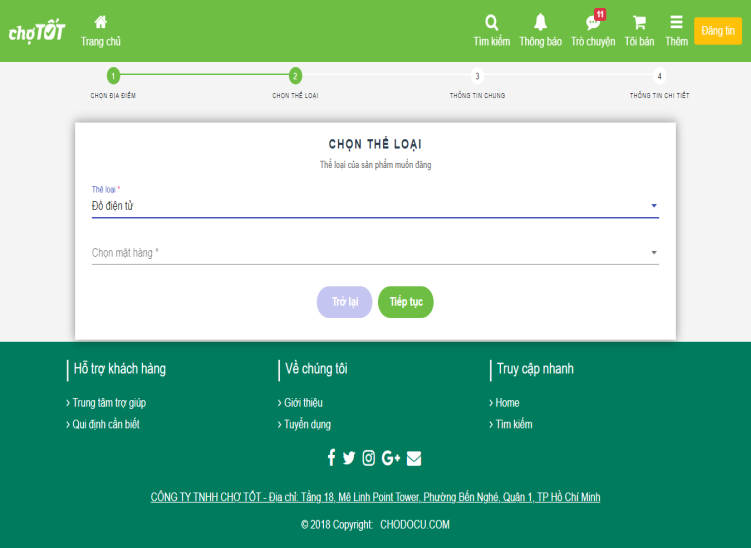
B2: người dùng nhấp chọn Đăng tin ,

B3: nếu người dùng chưa đăng nhập, hoặc thông tin đăng nhập đã hết hạn, hệ thống sẽ điều hướng tới trang đăng nhập, người dùng thực hiện đăng nhập, đăng nhập thành công hệ thống sẽ điều hướng tới trang Đăng tin. Để thực hiện đăng một tin đăng mới, người dùng cần nhập các thông tin bắt buộc sau đây :

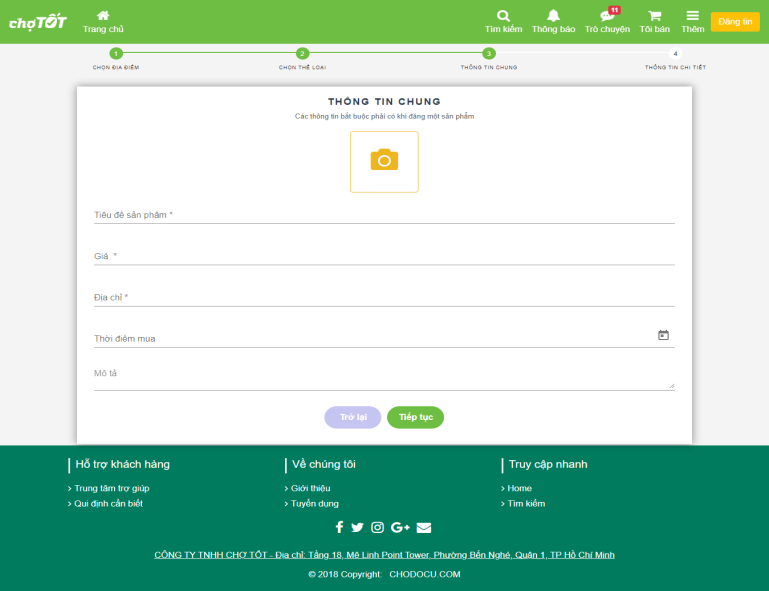
* + thông tin địa điểm, *Hình* ***41***
  + thông tin thể loại, *Hình* ***42***
  + thông tin chung, *Hình* ***43***



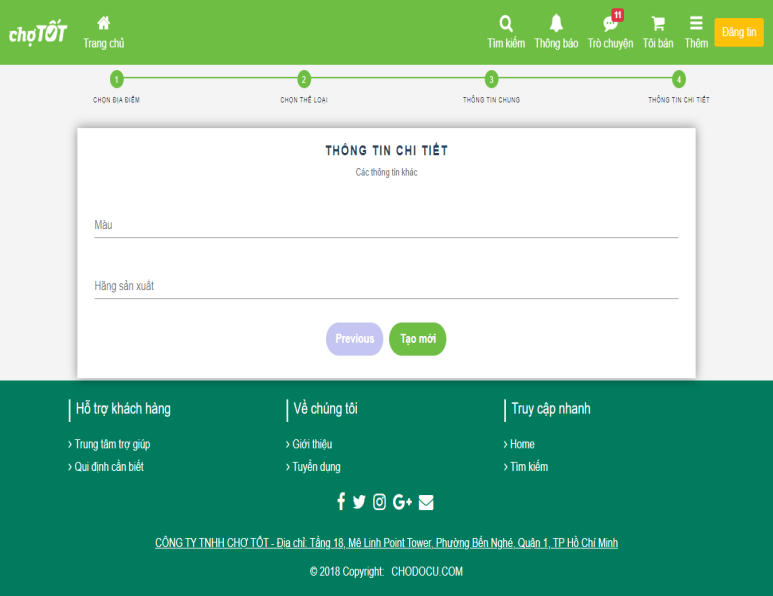
*Hình 41: Giao diện Chọn địa điểm để đăng tin*



*Hình 42: Giao diện Chọn thể loại để đăng tin*



*Hình 43: Giao diện Thông tin chung để đăng tin*



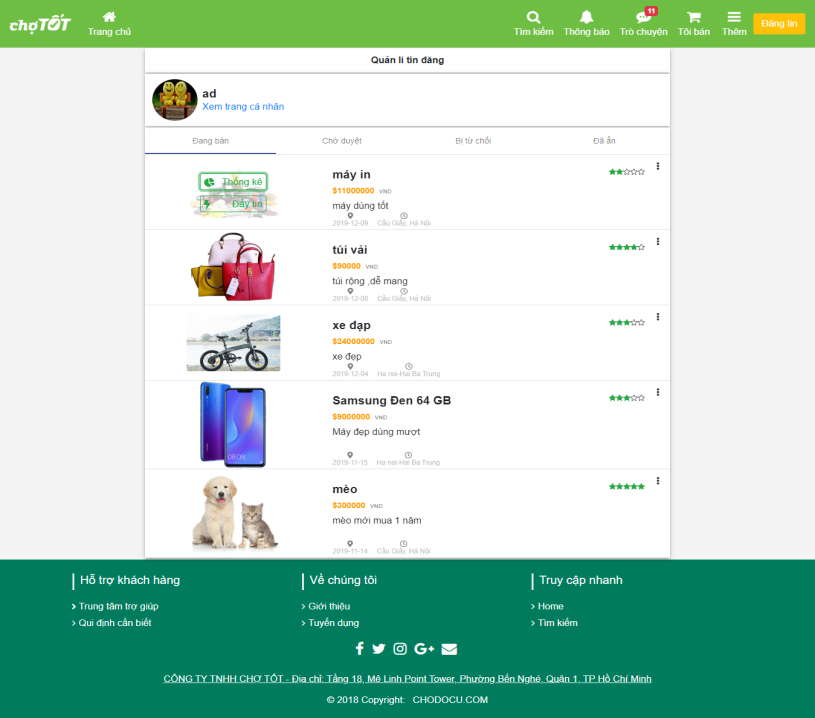
*Hình 44: Giao diện Thông tin chi tiết lúc đăng tin*

### Quản lí tin đăng

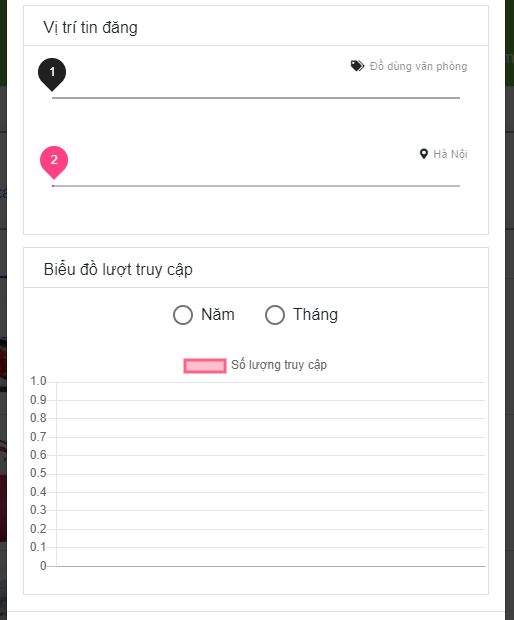
B1: truy cập vào hệ thống thông qua đường dẫn [*http://localhost:4200/*](http://localhost:4200/)

B2: đăng nhập

B3: nhấp chọn biểu tượng Tôi bán để vào trang Quản lí tin đăng như *Hình* ***45*** dưới đây, trang sẽ hiển thị danh sách các tin đăng đang bán, chờ duyệt, bị từ chối và đã ẩn. Trên danh sách đang bán, người dùng có thể thực hiện các thao tác Ẩn tin, Sửa tin Đẩy tin và xem Thống kê về tin đăng như *Hình* ***46***.



*Hình 45: Giao diện Quản lí tin đăng*

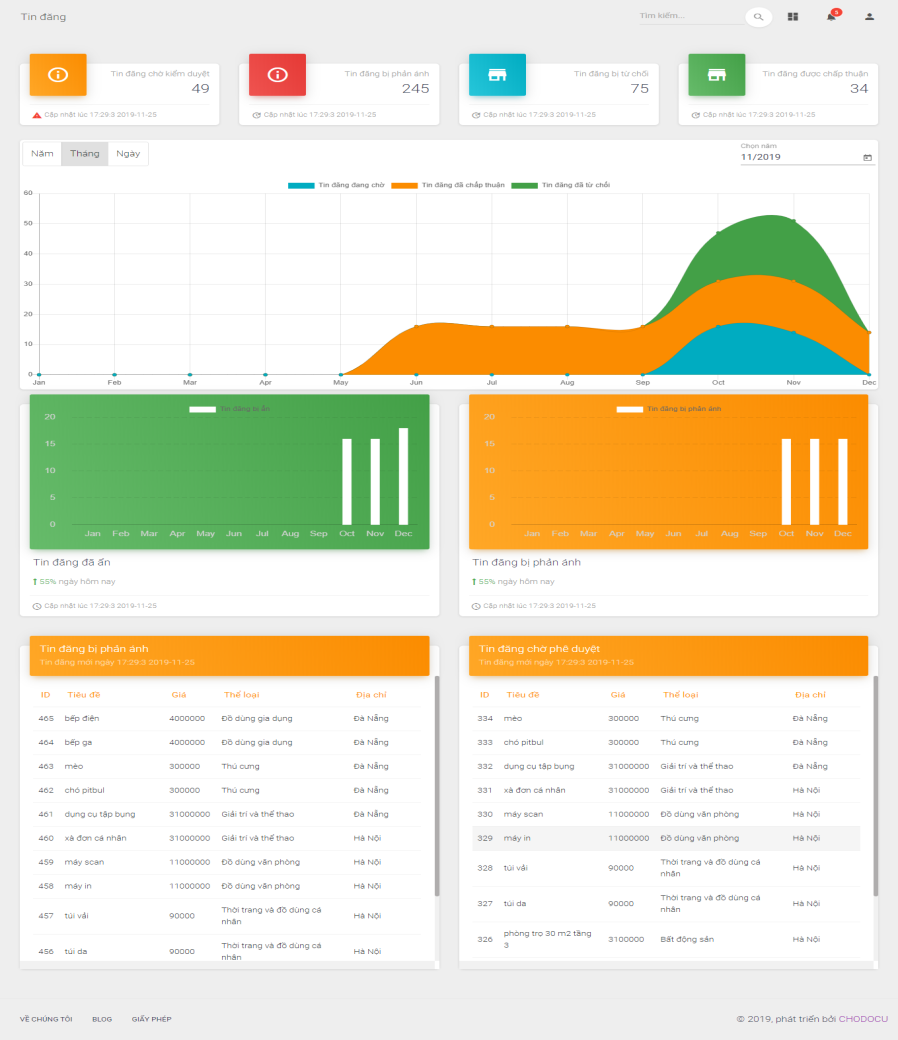


*Hình 46: Giao diện Thống kê tin đăng*

### Phê duyệt tin đăng

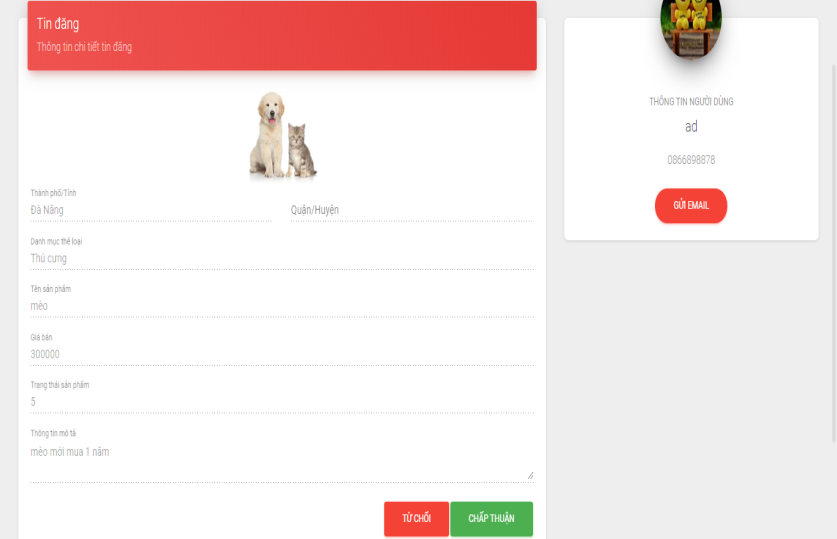
B1: truy cập vào hệ thống thông qua đường dẫn [*http://localhost:4201/*](http://localhost:4201/)

B2: đăng nhập với tài khoản có tên tài khoản là “admin” và mật khẩu là “admin”, sau đó nhấp chọn biểu tượng Tin đăng, hệ thống hiển thị danh sách các tin đăng mới cần được phê duyệt,kết quả hiển thị như *Hình* ***47*** dưới đây:



*Hình 47: Giao diện Quản trị tin đăng*

B3: nhấp chọn một tin đăng bất kì thuộc danh sách tin đăng cần phê duyệt, để đi đến trang Phê duyệt tin đăng như *Hình* ***48***, ở đây người quản trị có thể thực hiện thao tác Từ chối hoặc Chấp thuận và Gửi Email.



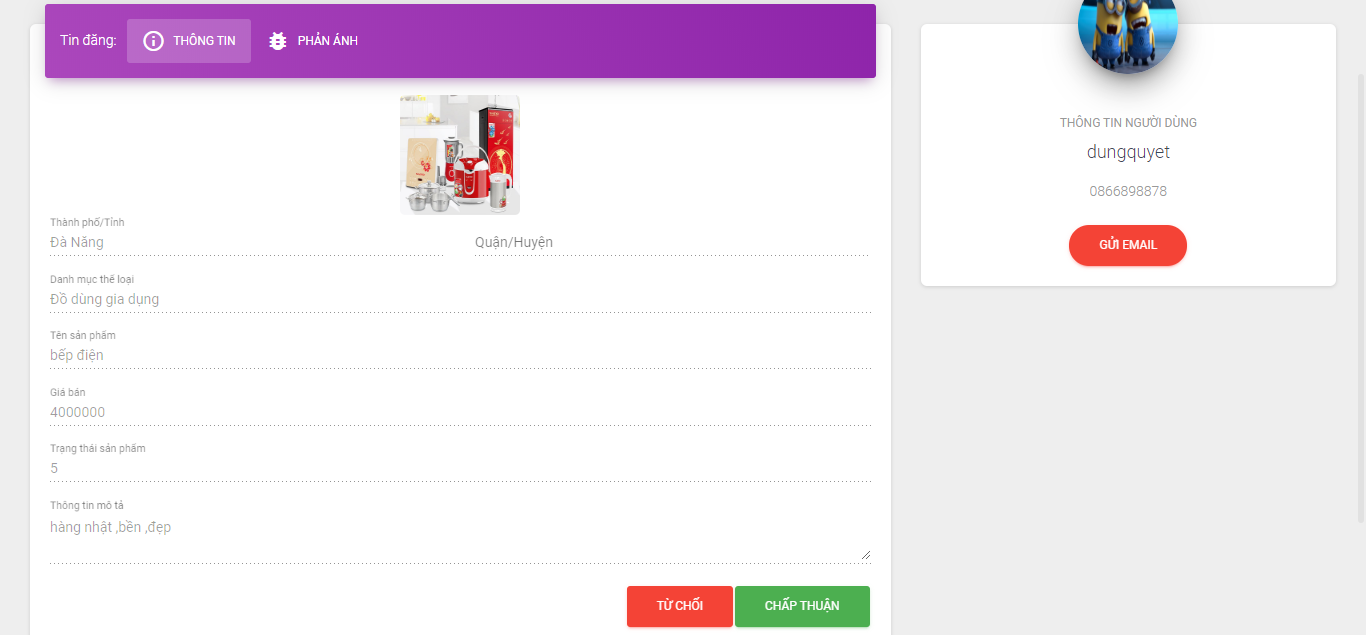
*Hình 48: Giao diện Phê duyệt tin đăng*

### Phê duyệt phản ánh

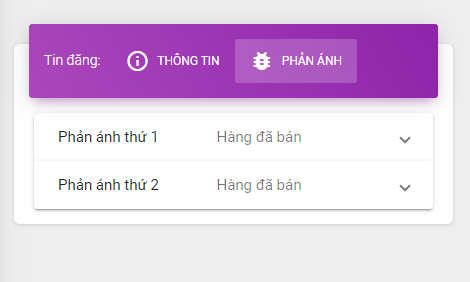
B1: truy cập vào hệ thống thông qua đường dẫn [*http://localhost:4201/*](http://localhost:4201/)

B2: đăng nhập với tài khoản có tên tài khoản là “admin” và mật khẩu là “admin”, sau đó nhấp chọn biểu tượng Tin đăng, hệ thống hiển thị danh sách các tin đăng bị phản ánh

B3: nhấp chọn một tin đăng bất kì thuộc danh sách tin đăng bị phản ánh để đi đến trang Phê duyệt phản ánh như *Hình* ***49*** dưới đây, tại đây người quản trị có thể thực hiện thao tác Từ chối hoặc Chấp thuận, Gửi Email, xem danh sách phản ánh của tin đăng bằng cách nhấp chọn Phản ánh



*Hình 49: Giao diện Phê duyệt phản ánh*



*Hình 50: Giao diện Danh sách phản ánh của tin đăng*

# KẾT LUẬN

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,cơ bản đã hoàn thành các yêu cầu đề ra của hệ thống.Quá trình phát triển phần mềm diễn ra từ thu thập yêu cầu, phân tích, thiết kế hệ thống đến triển khai hệ thống trên nền tảng Java Spring Boot và Angular 7.

Nhưng do hạn hẹp về thời gian nên hệ thống vẫn còn một vài yếu điểm***,*** các bước phân tích, thiết kế vẫn còn thiếu sót nên có thể chưa hoàn toàn bao quát hết các chức năng nghiệp vụ của hệ thống. Phần mềm được xây dựng vẫn còn một số chức năng chưa hoàn toàn đúng với yêu cầu đưa ra cần được khắc phục và xử lý để hoàn thiện hơn.Việc tìm kiếm chưa thực sự thân thiện với người dùng, cần cung cấp thêm 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. 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.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | https://spring.io/guides/gs/spring-boot/. |
| [2] | https://angular.io/guide/quickstart. |
| [3] | https://getbootstrap.com/docs/4.3/getting-started/introduction/. |
| [4] | https://www.tutorialspoint.com/mysql/index.htm. |
| [5] | https://www.learnrxjs.io/. |
| [6] | https://spring.io/guides/gs/rest-service/. |
| [7] | https://www.tutorialspoint.com/spring\_boot/spring\_boot\_building\_restful\_web\_services.htm. |
| [8] | https://auth0.com/blog/implementing-jwt-authentication-on-spring-boot/. |
| [9] | https://spring.io/guides/gs/messaging-stomp-websocket/. |
| [10] | [https://material.angular.io/components/categories. |