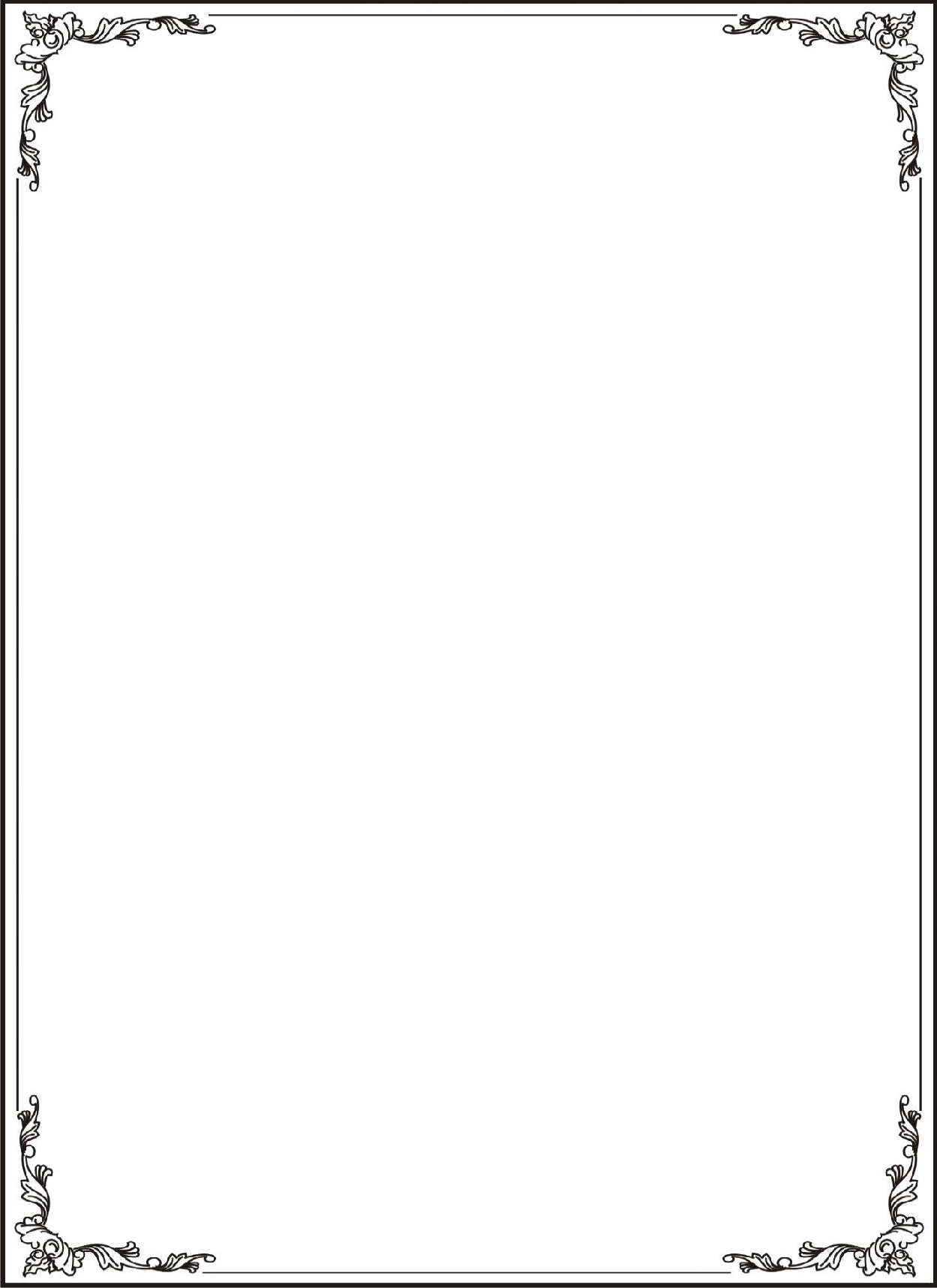
+

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

**KHOA CÔNG NGHỆ THÔNG TIN 1**



**ĐỀ TÀI:**

**XÂY DỰNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ THEO KIẾN TRÚC MICROSERVICE VÀ TÍCH HỢP CHATBOT TƯ VẤN KHÁCH HÀNG**

**Giảng viên hướng dẫn : TS. ĐÀO NGỌC PHONG**

**Sinh viên thực hiện : NGUYỄN MẠNH TUÂN**

**Lớp : D19CNPM2**

**Mã sinh viên : B19DCCN603**

**Hệ : Đại học**

**HÀ NỘI-NĂM 2023**

**ĐỒ ÁN**

**TỐT NGHIỆP ĐẠI HỌC**

# **LỜI CẢM ƠN**

Lời đầu tiên em xin được bày tỏ sự biết ơn sâu sắc đến giảng viên hướng dẫn TS. Đào Ngọc Phong đã tận tình giúp đỡ, hướng dẫn, chỉ bảo em trong suốt quá trình tìm hiểu, nghiên cứu và hoàn thành đồ án tốt nghiệp.

Em xin gửi lời cảm ơn sâu sắc tới thầy cô trong Học Viện Công Nghệ Bưu Chính Viễn Thông nói chung và trong khoa Công nghệ thông tin 1 nói riêng đã luôn nhiệt huyết, tận tình trong từng bài giảng và tạo điều kiện thuận lợi nhất cho em trong thời gian học tập tại trường.

Đối với em, đồ án tốt nghiệp là sự tổng hợp những kiến thức được các thầy cô truyền đạt trong quá trình học tập tại Học viện Công nghệ Bưu chính Viễn thông. Trong quá trình thực hiện không thể tránh khỏi thiếu sót vì thời gian có hạn và trình độ hiểu biết còn hạn chế. Em rất mong có được sự đóng góp ý kiến của thầy cô để đồ án trở nên hoàn thiện hơn.

Em xin chân thành cảm ơn!

***Hà Nội, Tháng 12/2023***

Sinh viên

NGUYỀN MẠNH TUÂN

**NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM**

**(Của Giảng viên hướng dẫn)**

……………………………………………………………………………………..…………………………………………………………………………………………..……………………………………………………………………………………..…………………………………………………………………………………………..……………………………………………………………………………………..…………………………………………………………………………………………..……………………………………………………………………………………..…………………………………………………………………………………………..……………………………………………………………………………………..…………………………………………………………………………………………..……………………………………………………………………………………..…………………………………………………………………………………………..……………………………………………………………………………………..…………………………………………………………………………………………..……………………………………………………………………………………..…………………………………………………………………………………………..……………………………………………………………………………………..…………………………………………………………………………………………..……………………………………………………………………………………..…………………………………………………………………………………………..……………………………………………………………………………………..…………………………………………………………………………………………..

**Điểm:** ………………………… (**bằng chữ:** ……………………………)

**Đồng ý/Không đồng ý** cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

……………, ngày tháng năm 2023

**CÁN BỘ-GIẢNG VIÊN HƯỚNG DẪN**

**NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM**

**(Của Giảng viên phản biện)**

………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..………..…..…………………………………………………………………………..

**Điểm:** ………………………… (**bằng chữ:** ……………………………)

**Đồng ý/Không đồng ý** cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

……………, ngày tháng năm 2023

**CÁN BỘ-GIẢNG VIÊN PHẢN BIỆN**

**MỤC LỤC**

[LỜI CẢM ƠN 2](#_Toc154499177)

[MỤC LỤC 5](#_Toc154499178)

[DANH MỤC HÌNH 7](#_Toc154499179)

[DANH MỤC BẢNG 9](#_Toc154499180)

[DANH MỤC TỪ VIẾT TẮT 10](#_Toc154499181)

[MỞ ĐẦU 11](#_Toc154499182)

[CHƯƠNG 1 : Khảo sát thực trạng và xác định yêu cầu hệ thống 13](#_Toc154499183)

[1.1. Khảo sát thực trạng 13](#_Toc154499184)

[1.2. Yêu cầu hệ thống 14](#_Toc154499185)

[1.3. Mục tiêu và chức năng chính của hệ thống 15](#_Toc154499186)

[1.4. Phương hướng giải quyết vấn đề của hệ thống 15](#_Toc154499187)

[1.5. Công nghệ sử dụng 16](#_Toc154499188)

[1.5.1. Front-End 16](#_Toc154499189)

[1.5.2 Back-End 21](#_Toc154499190)

[1.5.3. Database 24](#_Toc154499191)

[1.5.4. Một số dịch vụ bên thứ 3 25](#_Toc154499192)

[1.6. Kết luận 27](#_Toc154499193)

[CHƯƠNG 2 : Kiến trúc Microservice và hệ thống gợi ý 28](#_Toc154499194)

[2.1. Hệ thống gợi ý 28](#_Toc154499195)

[2.1.1. Giới thiệu chung 28](#_Toc154499196)

[2.1.2. Ý tưởng thuật toán 29](#_Toc154499197)

[2.1.3. Nguyên lý cài đặt thuật toán 30](#_Toc154499198)

[2.2. Kiến trúc microservice 35](#_Toc154499199)

[2.2.1. Một số kiến trúc phần mềm truyền thống 35](#_Toc154499200)

[2.2.2. Sơ lược về kiến trúc microservice 39](#_Toc154499201)

[2.2.3. Nguyên tắc thiết kế kiến trúc Microservice 42](#_Toc154499202)

[3.3. Kết luận 43](#_Toc154499203)

[CHƯƠNG 3 : Phân tích và thiết kế hệ thống 44](#_Toc154499204)

[3.1. Xác định và mô tả các tác nhân: 44](#_Toc154499205)

[3.2. Xây dựng bảng thuật ngữ 44](#_Toc154499206)

[3.3. Xác định và mô tả các ca sử dụng 46](#_Toc154499207)

[3.3.1. Danh sách các usecase cho người dùng 46](#_Toc154499208)

[3.3.2. Danh sách các usecase cho người quản trị 48](#_Toc154499209)

[3.4. Biểu đồ usecase 48](#_Toc154499210)

[3.4.1. Biểu đồ usecase tổng quát hệ thống 48](#_Toc154499211)

[3.4.2. Phân rã một số usecase chi tiết 49](#_Toc154499212)

[3.5. Xây dựng kịch bản 60](#_Toc154499213)

[3.5.1. Các kịch bản ứng với các chức năng của người dùng 60](#_Toc154499214)

[3.5.2. Các kịch bản ứng với các chức năng của quản trị hệ thống 72](#_Toc154499215)

[3.6. Xây dựng biểu đồ lớp phân tích 84](#_Toc154499216)

[3.6.1. Trích các lớp thực thể hệ thống 84](#_Toc154499217)

[3.6.2. Xác định quan hệ giữa các lớp 85](#_Toc154499218)

[3.6.3. Biểu đồ lớp phân tích 86](#_Toc154499219)

[3.7. Thiết kế kiến trúc microservie 87](#_Toc154499220)

[3.7.1. Tổng quan 87](#_Toc154499221)

[3.7.2. Các thành phần hệ thống 87](#_Toc154499222)

[3.8. Thiết kế các biểu đồ tuần tự hệ thống 88](#_Toc154499223)

[3.8.1. Biểu đồ tuần tự các chức năng của người dùng 88](#_Toc154499224)

[3.8.2. Biểu đồ tuần tự các chức năng của quản lí 97](#_Toc154499225)

[3.9. Thiết kế cơ sở dữ liệu 102](#_Toc154499226)

[3.9.1. Lược đồ cơ sở dữ liệu hệ thống 102](#_Toc154499227)

[3.10. Kết luận 103](#_Toc154499228)

[CHƯƠNG 4 : Cài đặt hệ thống 104](#_Toc154499229)

[4.1. Công cụ hỗ trợ 104](#_Toc154499230)

[4.1.1. Một số công cụ 104](#_Toc154499231)

[4.2. Chạy ứng dụng dưới local 104](#_Toc154499232)

[4.3. Deployment 104](#_Toc154499233)

[4.4. Kết quả cài đặt 105](#_Toc154499234)

[4.4.1. Kết quả áp dụng thuật toán gợi ý 105](#_Toc154499235)

[4.4.2. Giao diện phía người dùng 107](#_Toc154499236)

[4.4.3. Giao diện trang quản trị 110](#_Toc154499237)

[4.5. Kết luận 112](#_Toc154499238)

[KẾT LUẬN 113](#_Toc154499239)

[DANH MỤC TÀI LIỆU THAM KHẢO 114](#_Toc154499240)

# **DANH MỤC HÌNH**

[Hình 1.1 Cơ chế hoạt động của JavaScript trên trang web 19](#_Toc154499288)

[Hình 2.1 So sánh collaborative và contentbased filtering 29](#_Toc154499289)

[Hình 2.2 Ma trận người dùng – rating tương ứng 30](#_Toc154499290)

[Hình 2.3 Ma trận người dùng – rating tương ứng 31](#_Toc154499291)

[Hình 2.4 Chuẩn hóa ma trận 32](#_Toc154499292)

[Hình 2.5 Ma trận sau khi chuẩn hóa 32](#_Toc154499293)

[Hình 2.6 Ma trận độ tương tự giữa các người dùng 33](#_Toc154499294)

[Hình 2.7 Ma trận sau khi đự đoán đánh giá 34](#_Toc154499295)

[Hình 2.8 Ma trận đánh giá cuối cùng sau khi chuẩn hóa lại 35](#_Toc154499296)

[Hình 2.9 Kiến trúc Monolithic 36](#_Toc154499297)

[Hình 2.10 Mô hình hoạt động hệ thống SOA 38](#_Toc154499298)

[Hình 2.11 Mô hình kiến trúc Microservice 40](#_Toc154499299)

[Hình 3.1 Usecase tổng quát hệ thống 49](#_Toc154499300)

[Hình 3.2 Phân rã usecase Đăng kí 49](#_Toc154499301)

[Hình 3.3 Phân rã usecase Đăng nhập 50](#_Toc154499302)

[Hình 3.4 Phân rã usecase Quên mật khẩu 50](#_Toc154499303)

[Hình 3.5 Phân rã usecase Thay đổi thông tin cá nhân 51](#_Toc154499304)

[Hình 3.6 Phân rã usecase Tìm kiếm hàng điện tử 52](#_Toc154499305)

[Hình 3.7 Phân rã usecase Xem Danh sách hàng điện tử 52](#_Toc154499306)

[Hình 3.8 Phân rã usecase Lọc sản phẩm 53](#_Toc154499307)

[Hình 3.9 Phân rã usecase Sắp xếp sản phẩm 53](#_Toc154499308)

[Hình 3.10 Phân rã usecase Đánh giá cho hàng điện tử 54](#_Toc154499309)

[Hình 3.11 Phân rã usecase Gợi ý hàng điện tử 54](#_Toc154499310)

[Hình 3.12 Phân rã usecase Thêm vào giỏ hàng 55](#_Toc154499311)

[Hình 3.13 Phân rã usecasae Đặt hàng 55](#_Toc154499312)

[Hình 3.14 Phân rã usecase Xem lịch sử giao dịch 56](#_Toc154499313)

[Hình 3.15 Phân rã usecase Xem Danh sách hàng điện tử yêu thích 56](#_Toc154499314)

[Hình 3.16 Phân rã usecase So sánh sản phẩm 57](#_Toc154499315)

[Hình 3.17 Phân rã usecase Quản lý hàng điện tử 57](#_Toc154499316)

[Hình 3.18 Phân rã usecase Quản lý thương hiệu 58](#_Toc154499317)

[Hình 3.19 Phân rã usecase Quản lý thể loại 58](#_Toc154499318)

[Hình 3.20 Phân rã usecase Quản lý bộ sưu tập 59](#_Toc154499319)

[Hình 3.21 Phân rã usecase Quản lý đơn hàng 59](#_Toc154499320)

[Hình 3.22 Phân rã usecase Xem thống kê 60](#_Toc154499321)

[Hình 3.23 Biểu đồ lớp phân tích 86](#_Toc154499322)

[Hình 3.24 Lược đồ phân rã microservice 87](#_Toc154499323)

[Hình 3.25 Biểu đồ tuấn tự chức năng Đăng ký 89](#_Toc154499324)

[HÌnh 3.26 Biểu đồ tuần tự chức năng đăng nhập 90](#_Toc154499325)

[Hình 3.27 Biểu đồ tuần tự chức năng quên mật khẩu 91](#_Toc154499326)

[Hình 3.28 Biểu đồ tuần tự chức năng xem thông tin cá nhân 91](#_Toc154499327)

[Hình 3.29 Biểu đồ tuần tự chức năng thay đổi thông tin cá nhân 92](#_Toc154499328)

[Hình 3.30 Biểu đồ tuần tự chức năng Đánh giá 93](#_Toc154499329)

[Hình 3.31 Biểu đồ tuần tự chức năng Gợi ý tài khoản 94](#_Toc154499330)

[Hình 3.32 Biểu đồ tuần tự chức năng Tìm kiếm sản phẩm 94](#_Toc154499331)

[Hình 3.33 Biểu đồ tuần tự chức năng Thêm vào giỏ hàng 95](#_Toc154499332)

[Hình 3.34 Biểu đồ tuần tự chức năng Đặt hàng 96](#_Toc154499333)

[Hình 3.35 Biểu đồ tuần tự chức năng Xem những tài khoản yêu thích 97](#_Toc154499334)

[Hình 3.36 Biểu đồ tuần tự chức năng Xem những tài khoản 97](#_Toc154499335)

[Hình 3.37 Biểu đồ tuần tự chức năng thêm tài khoản mới 98](#_Toc154499336)

[Hình 3.38 Biểu đồ tuần tự chức năng sửa tài khoản 99](#_Toc154499337)

[Hình 3.39 Biểu đồ tuần tự chức năng xóa tài khoản 100](#_Toc154499338)

[Hình 3.40 Biểu đồ tuần tự chức năng xóa người dùng 101](#_Toc154499339)

[Hình 3.41 Biểu đồ tuần tự chức năng thống kê đơn hàng 102](#_Toc154499340)

[Hình 3.42 Biểu đồ tuần tự chức năng thống kê doanh thu 102](#_Toc154499341)

[Hình 3.43 Lược đồ cơ sở dữ liệu hệ thống 103](#_Toc154499342)

[Hình 4.1 Triển khai front-end trên Vercel 105](#_Toc154499343)

[Hình 4.2 Gợi ý sản phẩm trên trang chủ 106](#_Toc154499344)

[Hình 4.3 Gợi ý sản phẩm trong trang Chi tiết sản phẩm 106](#_Toc154499345)

[Hình 4.4 Giao diện trang chủ hệ thống 108](#_Toc154499346)

[Hình 4.5 Giao diện chi tiết sản phẩm 108](#_Toc154499347)

[Hình 4.6 Giao diện bình luận 109](#_Toc154499348)

[Hình 4.7 Giao diện giỏ hàng 109](#_Toc154499349)

[Hình 4.8 Thanh toán qua VNPay 110](#_Toc154499350)

[Hình 4.9 Giao diện thành toán thành công 110](#_Toc154499351)

[Hình 4.10 Giao diện quản lý tài khoản 111](#_Toc154499352)

[Hình 4.11 Giao diện thêm tài khoản 111](#_Toc154499353)

[Hình 4.12 Giao diện thống kế đơn hàng 112](#_Toc154499354)

# **DANH MỤC BẢNG**

[Bảng 3.1 Xác định các tác nhân 45](#_Toc154499355)

[Bảng 3.2 Các thuật ngữ dùng trong hệ thống 45](#_Toc154499356)

[Bảng 3.3 Danh sách usecase của người dùng 47](#_Toc154499357)

[Bảng 3.4 Danh sách điện tử usecase người quản trị 49](#_Toc154499358)

[Bảng 3.5 Kịch bản cho chức năng đăng ký 61](#_Toc154499359)

[Bảng 3.6 Kịch bản cho chức năng đăng nhập 62](#_Toc154499360)

[Bảng 3.7 Kịch bản cho chức năng quên mật khẩu 62](#_Toc154499361)

[Bảng 3.8 Kịch bản cho chức năng đổi mật khẩu 63](#_Toc154499362)

[Bảng 3.9 Kịch bản cho chức năng đánh giá 64](#_Toc154499363)

[Bảng 3.10 Kịch bản cho chức năng tìm kiếm sản phẩm theo tên 65](#_Toc154499364)

[Bảng 3.11 Kịch bản cho chức năng xem thông tin cá nhân 65](#_Toc154499365)

[Bảng 3.12 Kịch bản cho chức năng thay đổi thông tin cá nhân 66](#_Toc154499366)

[Bảng 3.13 Kịch bản cho chức năng gợi ý hàng điện tử 67](#_Toc154499367)

[Bảng 3.14 Kịch bản cho chức năng thêm vào giỏ hàng 67](#_Toc154499368)

[Bảng 3.15 Kịch bản cho chức năng đặt hàng và thanh toán 68](#_Toc154499369)

[Bảng 3.16 Kịch bản cho chức năng xem lịch sử giao dịch 69](#_Toc154499370)

[Bảng 3.17 Kịch bản cho chức năng xem những hàng điện tử đã thích 69](#_Toc154499371)

[Bảng 3.18 Kịch bản cho chức năng xem danh sách sản phẩm 70](#_Toc154499372)

[Bảng 3.19 Kịch bản cho chức năng lọc sản phẩm 71](#_Toc154499373)

[Bảng 3.20 Kịch bản cho chức năng sắp xếp sản phẩm 71](#_Toc154499374)

[Bảng 3.21 Kịch bản cho chức năng so sánh sản phẩm 72](#_Toc154499375)

[Bảng 3.22 Kịch bản cho chức năng xem danh sách hàng điện tử 73](#_Toc154499376)

[Bảng 3.23 Kịch bản cho chức năng thêm hàng điện tử mới 74](#_Toc154499377)

[Bảng 3.24 Kịch bản cho chức năng sửa hàng điện tử mới 75](#_Toc154499378)

[Bảng 3.25 Kịch bản cho chức năng xóa hàng điện tử 75](#_Toc154499379)

[Bảng 3.26 Kịch bản cho chức năng xem Danh sách điện tử Thể loại 76](#_Toc154499380)

[Bảng 3.27 Kịch bản cho chức năng thêm Thể loại mới 77](#_Toc154499381)

[Bảng 3.28 Kịch bản cho chức năng sửa Thể loại 77](#_Toc154499382)

[Bảng 3.29 Kịch bản cho chức năng xóa Thể loại 78](#_Toc154499383)

[Bảng 3.30 Kịch bản cho chức năng xem danh sách Bộ sưu tập 79](#_Toc154499384)

[Bảng 3.31 Kịch bản cho chức năng thêm Bộ sưu tập mới 79](#_Toc154499385)

[Bảng 3.32 Kịch bản cho chức năng sửa Bộ sưu tập mới 80](#_Toc154499386)

[Bảng 3.33 Kịch bản cho chức năng xóa Bộ sưu tập 81](#_Toc154499387)

[Bảng 3.34 Kịch bản cho chức năng xem danh tài khoản tài khoản 81](#_Toc154499388)

[Bảng 3.35 Kịch bản cho chức năng thêm tài khoản mới 82](#_Toc154499389)

[Bảng 3.36 Kịch bản cho chức năng sửa tài khoản mới 82](#_Toc154499390)

[Bảng 3.37 Kịch bản cho chức năng xóa tài khoản 83](#_Toc154499391)

[Bảng 3.38 Kịch bản cho chức năng thống kê đơn hàng 84](#_Toc154499392)

[Bảng 3.39 Kịch bản cho chức năng thống kê doanh thu 85](#_Toc154499393)

[Bảng 3.40 Các thành phần trong hệ thống microserivce 88](#_Toc154499394)

# **DANH MỤC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt** | **Ý nghĩa** |
| 1 | HTML | HyperText Markup Language |
| 2 | CSS | Cascading Style Sheet |
| 3 | XML | eXtensible Markup Language |
| 4 | NBCF | Neighborhood-based Collaborative Filtering |
| 5 | DRF | Django Rest Framework |
| 6 | CSDL | Cơ sơ dữ liệu |
| 7 | IP | Internet Protocol |

# **MỞ ĐẦU**

Trong thời đại hiện nay, công nghệ thông tin đang phát triển rất mạnh mẽ vì hầu hết người dùng đều sử dụng mạng internet với nhiều mục đích như: tìm kiếm thông tin, mua bán hàng trực tuyến, khai thác dữ liệu,… Trong đó, hình thức kinh doanh online đang phát triển rất mạnh mẽ. Chính vì thế, việc xây dựng hệ thống website thương mại điện tử trên mạng là một nhu cầu cần thiết giúp các độc giả dễ dàng tìm kiếm những sản phẩm mà họ cần mua.

Lợi ích đầu tiên khi sở hữu một website thương mại điện tử đó chính là tiếp cận khách hàng tiềm năng. Dù khách hàng ở bất cứ đâu trên thế giới, chỉ cần có mạng internet là có thể tìm thấy website bán hàng của, biết đến sản phẩm của . Việc sở hữu một website thương mại điện tử giúp cho cửa hàng của tiết kiệm được nhiều chi phí trong quảng bá thương hiệu. Vì không cần phải đầu tư một số tiền lớn để thuê mặt bằng làm cửa hàng, không cần phải thuê nhân viên phục vụ mà chỉ cần đầu tư một khoản tiền nhỏ để thiết kế website bán hàng trực tuyến.

Cùng với đó, trong những năm gần đây, việc áp dụng kiến trúc microservices để xây dựng các ứng dụng doanh nghiệp ngày càng phổ biến. Sự ra đời của microservices đem lại nhiều lợi ích như làm giảm độ phức tạp trong việc xây dựng các ứng dụng lớn cho doanh nghiệp, tăng tính mở rộng cũng như nâng cao khả năng bảo trì hệ thống. Tại Việt Nam, nhiều công ty như Tiki, Viettel đều đã áp dụng mô hình microservices để xây dựng các giải pháp phần mềm phục vụ bài toán doanh nghiệp.

Ngoài ra, để giúp tăng trải nghiệm mua sắm của khách hàng thì hệ thống chatbot và gợi ý khách hàng (recomment system) cũng được đưa vào để hỗ trợ khách hàng có thể dễ dàng lựa chọn các sản phẩm một cách dễ dàng nhất

Chính vì thế em đã thực hiện đề tài *“Xây dựng website thương mại điện tử theo kiến trúc microservice và tích hợp chatbot tư vấn khách hàng”* nhằm vận dụng các kiến thức đã học được để áp dụng vào giải quyết các vấn đề về mua bán hàng của hệ thống thương mại điện tử

**Mục đích đề tài**

Tìm hiểu cách thức hoạt động của thương mại điện tử cụ thể là trang web bán sản phẩm điện tử trực tuyến. Hệ thống phải cung cấp nhiều chức năng phục vụ việc mua sắm của khách hàng như tìm kiếm sản phẩm, đặt hàng và Thanh toán online… Đảm bảo người quản trị có thể quản lý web dễ dàng. Giao diện hệ thống thân thiện, dễ dử dụng, hiệu năng tốt không bị giật lag.

Tìm hiểu nguyên lí hoạt động của hệ thống sử dụng kiến trúc microservice, tìm hiểu các kỹ thuật để phát triển theo hướng kiến trúc này sử dụng công nghệ trên nền tảng Python như Django .

**Phạm vi đề tài**

Đề tài thực hiện xây dựng hệ thống bán hàng điện tử trực tuyến của một cửa hàng nhằm phục vụ cho khách hàng trong nước. Tuy nhiên dữ liệu lưu trữ vừa và nhỏ, hệ thống chạy trên máy cục bộ. Trong tương lai sẽ phát triển thêm không chỉ một cửa hàng mà nhiều cửa hàng, dữ liệu lưu trữ lớn hơn và chạy trên máy chủ.

**Công nghệ sử dụng**

Hệ thống sử dụng ngôn ngữ HTML, CSS, JavaScriptcùng với thư viện thiết kế giao diện tốt nhất hiện nay ReactJS và công cụ hỗ trợ framework Django Rest Framework(DRF) được xây dựng trên nền tảng Python.

**Nội dung chính của đề tài**

Nội dung đồ án gồm 4 chương sau:

**Chương 1:** Khảo sát thực trạng và xác định yêu cầu hệ thống

**Chương 2:** Kiến trúc Microservice và thuật toán gợi ý

**Chương 3:** Phân tích và thiết kế hệ thống

**Chương 4:** Cài đặt hệ thống

# **CHAT BOT đâu???**

# **: Khảo sát thực trạng và xác định yêu cầu hệ thống**

* 1. **Khảo sát thực trạng**

Đối tượng khảo sát là một số trang web bán hàng điện tử trực tuyến và thanh toán trực tuyến như CellphoneS, HoangHaMobile, thegiodidong,… Qua quá trình khảo sát thu được một số kết quả sau:

Mỗi hệ thống bán hàng trực tuyến thường gồm 2 phần chính:

* Phần thứ nhất là phần giao dịch với khách hàng trên Internet.
* Phần thứ hai dành cho nhân viên cửa hàng quản trị viên hệ thống.

Mỗi phần có các chức năng con khác nhau phục vụ cho mục đích kinh doanh và quản lý của cửa hàng.

**Hoạt động mua hàng của khách hàng:**

Khách hàng truy cập vào trang web trên trình duyệt. Trình duyệt sẽ hiển thị các sản phẩm và dịch vụ mà cửa hàng cung ứng. Khách hàng có thể duyệt sản phẩm theo thể loại hoặc có thể sử dụng chức năng tìm kiếm để tìm kiếm sản phẩm.

Mỗi khách hàng được cung cấp một giỏ hàng tương ứng. Khi khách hàng thêm những mặt hàng vào giỏ hàng của mình, một trang chi tiết giỏ hàng hiện ra, trong đó có những sản phẩm, số lượng sản phẩm mà khách hàng chọn mua, khách hàng có thể cập nhật số lượng hay xóa sản phẩm muốn mua hoặc tiếp tục mua hàng và thanh toán.

Muốn đặt được hàng thì khách hàng phải đăng nhập hệ thống. Nếu khách hàng chưa có tài khoản thì hệ thống sẽ chuyển đến trang đăng ký. Sau đó khách hàng sẽ cung cấp các thông tin về địa chỉ nhận hàng cũng như phương thức thanh toán với cửa hàng. Nếu cửa hàng mà có liên kết với các ngân hàng hoặc các ví điện tử để xây dựng một hệ thống thanh toán trực tuyến thì hệ thống đang hoạt động sẽ chuyển đến trang thanh toán để người dùng cung cấp các thông tin về tài khoản của họ. Ngân hàng hoặc ví điện tử sẽ dựa vào những thông tin đó để thực hiện các nghiệp vụ thanh toán. Sau đó hệ thống sẽ gửi thông báo tới người dùng kết quả thanh toán.

Sau khi đã cung cấp đầy đủ thông tin, khách hàng có thể xem lại, chỉnh sửa đơn hàng rồi bấm đặt hàng. Hệ thống sẽ phản hồi thông báo cho khách hàng biết là đơn hàng đã đặt thành công và cửa hàng sẽ gửi lại cho khách hàng về biên đặt hàng hoặc việc giao hàng và quá trình mua hàng sẽ kết thúc. Ngoài ra khách hàng còn có thể thực hiện các công việc như: Đánh giá nhận xét sản phẩm, nhắn tin với cửa hàng, thay đổi thông tin tài khoản…

Các thông tin cần hiển thị trên trang chủ của ứng dụng: Danh mục thể loại hàng điện tử, hàng điện tử mới phát hành, hàng điện tử bán chạy, một vài chức năng như tìm kiếm, đăng ký, đăng nhập, xem giỏ hàng, lịch sử đơn hàng…

Các thông tin về tài khoản của khách hàng bao gồm: số điện thoại, địa chỉ email, mật khẩu truy nhập, tên khách hàng.

Mỗi hàng điện tử có các thông tin sau: tên hàng điện tử, ảnh minh họa, thể loại, giá, thương hiệu, kích thước, mô tả, màu sắc.

Giỏ hàng lưu các thông tin về sản phẩm khách hàng chọn mua: tên hàng điện tử, số lượng đặt mua, đơn giá và tổng trị giá các sản phẩm trong giỏ hàng. Ngoài ra gồm các chức năng thêm, xóa, cập nhật sản phẩm trong giỏ hàng.

Đơn đặt hàng gồm các thông tin về khách hàng: tên khách hàng, số điện thoại, email, địa chỉ giao hàng, phương thức vận chuyển, phương thức thanh toán. Các thông tin về sản phẩm đặt mua: tên sản phẩm, số lượng, giá bán và tổng giá trị đơn hàng.

Về phương thức thanh toán thì hệ thống hỗ trợ các phương thức thanh toán: trực tiếp, chuyển khoản qua ngân hàng hoặc ví điện tử.

Về phương thức vận chuyển hệ thống cung cấp các hình thức vận chuyển và phí vận chuyển.

**Hoạt động quản lý hệ thống của quản trị viên:**

Để truy cập vào chức năng quản trị hệ thống, quản trị viên cần có tài khoản để đăng nhập vào hệ thống. Sau khi đăng nhập thành công quản trị viên có thể thực hiện các chức năng sau:

Quản lý hàng điện tử: Thêm, sửa và xóa thông tin của mỗi hàng điện tử.

Quản lý người dùng: sửa hoặc xóa thông tin người dùng trong hệ thống.

Quản lý đơn hàng: xử lý các đơn đặt hàng của khách hàng, giao hàng hoặc hủy đơn hàng.

Quản lý các phương thức thanh toán: thêm, sửa hoặc xóa thông tin phương thức thanh toán.

Quản lý các phương thức vận chuyển: thêm, sửa hoặc xóa thông tin phương thức vận chuyển.

Quản lý tài chính: sử dụng các thông tin về đơn đặt hàng mà khách hàng đã đặt để có thể thống kê tài chính theo từng tháng, từng quý.

* 1. **Yêu cầu hệ thống**

Sau khi khảo sát các cửa hàng bán hàng điện tử online, xác định được yêu cầu của hệ thống như sau: khách hàng có thể tìm kiếm, mua bất cứ sản phẩm nào trong trạng thái còn hàng. Đơn hàng, thông tin khách hàng phải được bảo mật. Khách hàng có thể thanh toán khi nhận sản phẩm hoặc thanh toán bằng thẻ ngân hàng, ví điện tử. Quản trị viên có thể quản lý thông tin khách hàng, cập nhật tình trạng đơn hàng trên hệ thống và gửi thông tin đến email cá nhân của khách hàng. Giao diện người dùng, giao diện quản lý thân thiện, đẹp mắt, hiệu năng tốt.

Các đối tượng sử dụng hệ thống:

* Quản trị viên (Admin): là người có quyền cao nhất trong hệ thống, có thể thêm, sửa, xóa các thông tin quan trọng liên quan đến hệ thống (quản lý về sản phẩm, đơn hàng, người dùng,..).
* Người dùng (User): là thành phần thao tác chính với hệ thống, có các chức năng đăng ký, quản lý tài khoản, tìm kiếm, đặt hàng và thanh toán sản phẩm.
  1. **Mục tiêu và chức năng chính của hệ thống**
     1. **Mục tiêu của hệ thống**

Cung cấp cho khách hàng cơ hội mua hàng hóa và sử dụng dịch vụ với sự trợ giúp của hệ thống thanh toán điện tử mà không cần rời khỏi nhà.

Cung cấp các thông tin hoặc thông báo cho khách hàng về sản phẩm, về đơn đặt hàng.

Tạo ra một nơi giao tiếp hiệu quả giữa nhân viên cửa hàng với khách hàng.

Tăng cơ sở khách hàng hiện có, tiếp cận khách hàng mới và tăng chuyển đổi thành khách hàng tiềm năng.

* + 1. **Chức năng của hệ thống**

Trang web cần có các chức năng cơ bản của một trang thương mại điện tử:

Người dùng có thể đăng ký và đăng nhập vào hệ thống để quản lý các thông tin như: tài khoản, giỏ hàng, đặt hàng, thanh toán, đánh giá và nhận xét sản phẩm. Ngoài ra, người dùng có thể tìm kiếm sản phẩm theo các tiêu chí khác nhau (tên sản phẩm, giá tiền, thể loại,…)và tra cứu lịch sử đơn hàng hoặc hủy đơn hàng nếu hàng chưa được vận chuyển. Khi người dùng xác nhận thanh toán hoặc người quản trị xác nhận vận chuyển đơn hàng, người dùng có thể tự động nhận được tin nhắn về chi tiết đơn hàng đã đặt hoặc trạng thái đơn hàng.

Ngoài việc có thể thực hiện các chức năng như người dùng, quản trị viên có thêm các chức năng về quản lý cửa hàng như: Quản lý sản phẩm (thêm, sửa, xóa sản phẩm), đơn hàng (giao đơn hàng, xóa đơn hàng), người dùng (sửa, xóa người dùng), quảng cáo (thêm, sửa, xóa ảnh bìa). Thêm nữa, quản trị viên có thể xem thống kê cửa hàng thể hiện tổng số người dùng, tổng thu nhập tháng, số đơn đặt hàng trong tháng, số lượng sản phẩm theo thể loại...

* 1. **Phương hướng giải quyết vấn đề của hệ thống**

Xây dựng website bán hàng điện tử trực tuyến với các chức năng của người dùng như: đăng ký, đăng nhập, tìm kiếm hàng điện tử, bán hàng điện tử, đánh giá hàng điện tử. Và các chức năng của người quản trị như: quản lý hàng điện tử, hàng điện tử, thống kê lượt xem, doanh thu,…

Hệ thống áp dụng mô hình microservice, gồm nhiều dịch vụ khác nhau mỗi dịch vụ thực hiện một nghiệp vụ xác định

Ngoài ra, hệ thống sẽ áp dụng hệ thống gợi ý sử dụng thuật toán Neighborhood based Collaborative Filtering để đưa ra gợi ý cho người sử dụng.

* 1. **Công nghệ sử dụng**
     1. **Front-End**

Giao diện của hệ thống (front-end) được xây dựng dựa trên các yếu tố chính là HTML, CSS và JavaScript nhưng được viết dưới sự hỗ trợ của một thư viện mạnh mẽ xây dựng UI mạnh mẽ là ReactJS để xây dựng lên một trang web. Cụ thể sẽ được trình bày ở phần dưới

* + - 1. **HTML**

**HTML** **(HyperText Markup Language)** là ngôn ngữ đánh dầu tiêu chuẩn để xây dựng trang web [1]. Đây chính là xương sống của mọi website, tạo nên nền móng cho bất kỳ một trang web nào trên Internet. Các trình duyệt web sẽ nhận các HTML từ Server hoặc từ Local(HTML Document).

Mỗi tài liệu HTML (HTML Document) bao gồm một loạt các phần tử sử dụng để bao bọc hoặc bao bọc các phần khác nhau của nội dung nhằm làm cho nội dung đó xuất hiện theo một cách nhất định hoặc hoạt động theo một cách nhất định. Các thẻ kèm theo có thể tạo siêu liên kết hoặc hình ảnh đến một nơi khác, có thể in nghiêng các từ, có thể làm cho phông chữ lớn hơn hoặc nhỏ hơn, v.v. Ví dụ:

**<h1>My First Heading</h1>**

**<p>My first paragraph.</p>**

Phần tử có thể bao gồm các thuộc tính, ví dụ như sau:

**<p class=”editor”>My cat is very grumpy</p>**

Thuộc tính chứa thông tin bổ sung về thành phần mà không muốn xuất hiện trong nội dung thực tế. Ở đây, lớp là tên thuộc tính và editor-note là giá trị thuộc tính. Thuộc tính lớp cho phép cung cấp cho phần tử một mã định danh không duy nhất có thể được sử dụng để nhắm mục tiêu nó (và bất kỳ phần tử nào khác có cùng giá trị lớp) với thông tin kiểu và những thứ khác. Một số thuộc tính không có giá trị, chẳng hạn như thuộc tính bắt buộc.

Hiện tại HTML6 - phiên bản mới nhất của HTML với nhiều công nghệ giúp cho việc phát triển trang web trở nên nhanh và đạt hiệu quả hơn.

* + - 1. **CSS**

**CSS(Cascading Style Sheets):** là ngôn ngữ để tạo kiểu cho tài liệu HTML(HTML Document) [2]. CSS mô tả cách hiển thị các phần tử HTML. Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc…

CSS được phát triển bởi **W3C** ([World Wide Web Consortium](https://www.w3.org/)) vào năm 1996, vì HTML không được thiết kế để gắn tag để giúp định dạng trang web.

Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.

Một đoạn CSS bao gồm các phần như sau:

Bộ chọn{

Thuộc tính: giá trị;

Thuộc tính: giá trị;

…

}

Trong đó:

* **Bộ chọn (Selector):**là mẫu để chọn phần tử HTML muốn định nghĩa phong cách. Các selector được áp dụng cho các trường hợp sau:
  + Tất cả phần tử theo một dạng cụ thể nào đó, ví dụ phần tử tiêu đề h1.
  + Thuộc tính id và class của các phần tử.
  + Các phần tử dựa vào mối liên quan với các phần tử khác trong cây phân cấp tài liệu.
* **Khai báo (Declaration)**: Khối khai báo chứa một hoặc nhiều khai báo, phân tách với nhau bằng các dấu chấm phẩy. Mỗi khai báo gồm tên và giá trị đặc tính CSS, phân tách bằng dấu phẩy. Khai báo CSS luôn kết thúc bằng dấu chấm phẩy, khối khai báo nằm trong các dấu ngoặc móc. Trong ví dụ dưới đây, các phần tử <p> sẽ được căn giữa, chữ màu đỏ.

p { color: red; text-align: center; }

* **Thuộc tính (Properties):** Những cách có thể tạo kiểu cho một phần tử HTML. (Với trường hợp này thì color được xem là một trong những  thuộc tính của phần tử p). Chính vì vậy, với CSS chỉ cần lựa chọn thuộc tính muốn tác động nhất trong bộ quy tắc của mình.
* **Giá trị thuộc tính:** Ở bên phải của thuộc tính sau dấu hai chấm(:), chúng ta sẽ sở hữu giá trị thuộc tính mà việc lựa chọn trong số đó sẽ xuất hiện nhiều lần để có thể cho một thuộc tính cụ thể nào đó.
  + - 1. **JavaScript**

**JavaScript** là một ngôn ngữ lập trình bậc cao, hỗ trợ lập trình hướng đối tượng. JavaScript ra đời năm 1995, là một ngôn ngữ lập trình phía máy khách (client-side) phổ biến và mạnh mẽ, được sử dụng chủ yếu để xây dựng tương tác giữa các thành phần trong một trang web với nhau, đảm nhiệm xử lý các sự kiện và thao tác phía người dùng.

**Cơ chế hoạt động của JavaScript trên 1 trang web:**

Khi web browser tải một web page, HTML parser bắt đầu phân tích cú pháp HTML code và tạo DOM. Bất cứ khi nào trình phân tích cú pháp gặp một chỉ thị CSS hoặc JavaScript (inline hoặc externally loaded), nó sẽ được chuyển giao cho CSS parser hoặc JavaScript engine theo yêu cầu.

JavaScript engine tải các tệp JavaScript bên ngoài và inline code, nhưng không chạy mã ngay lập tức, mà đợi HTML và CSS phân tích hoàn tất. Khi điều này được thực hiện, JavaScript được thực thi theo thứ tự chúng được tìm thấy trên trang web: các biến và hàm được xác định, các lời gọi hàm được thực thi, trình xử lý sự kiện được kích hoạt,... Các hoạt động này dẫn đến việc DOM được cập nhật bởi JavaScript và được hiển thị ngay lập tức bởi trình duyệt.

A diagram of a software development process

Description automatically generated

Hình 1.1 Cơ chế hoạt động của JavaScript trên trang web

**Các điểm nổi bất của JavaScript:**

**Tính tương tác trên môi trường web:** JavaScript cho phép tương tác với các phần tử HTML và CSS trên trang web. Bằng cách sử dụng JavaScript, có thể thay đổi nội dung của trang, thêm hoặc xóa các phần tử, thực hiện các hiệu ứng và kiểm soát hành vi của trang.

**Đa nền tảng**: JavaScript không chỉ chạy trên trình duyệt web, mà còn có thể chạy trên nhiều nền tảng khác nhau như máy chủ (server-side) thông qua Node.js, ứng dụng di động thông qua các framework như React Native và Ionic, và ứng dụng desktop thông qua Electron.

**Đa mục đích:** JavaScript không chỉ giới hạn trong việc tạo ra các ứng dụng web. Nó cũng có thể được sử dụng để phát triển ứng dụng máy chủ, ứng dụng di động, trò chơi, phần mềm desktop, và nhiều lĩnh vực khác.

**Cú pháp dễ học:** JavaScript có cú pháp dễ học và dễ hiểu, với nhiều tài liệu và nguồn học trực tuyến có sẵn. Nếu đã quen thuộc với HTML và CSS, việc học JavaScript sẽ trở nên dễ dàng hơn.

**Hỗ trợ rộng rãi:** JavaScript có một cộng đồng lớn và nhiều thư viện, framework hỗ trợ phong phú để giúp phát triển ứng dụng một cách nhanh chóng và hiệu quả. Một số framework phổ biến như React, Angular và Vue.js đang được sử dụng rộng rãi trong cộng đồng phát triển web.

**Tính tương thích cao:** JavaScript được hỗ trợ trên hầu hết các trình duyệt web hiện đại, bao gồm Chrome, Firefox, Safari, và Edge. Điều này đảm bảo rằng ứng dụng JavaScript của có thể chạy trên nhiều nền tảng khác nhau mà không cần phải viết lại mã nguồn.

* + - 1. **ReactJS**

**ReactJS** là một thư viện JavaScript có tính hiệu quả và linh hoạt để xây dựng giao diện người dùng (UI) cho các ứng dụng web đơn trang (Single Page Applications - SPAs) và ứng dụng di động.. ReactJS giúp phân chia các UI phức tạp thành các phần tử nhỏ (được gọi là component). Nó ra đời năm 2013 bởi Jordan Walke, một kỹ sư phần mềm tại Facebook. ReactJS ban đầu được phát triển và duy trì bởi Facebook và sau đó được doanh nghiệp lớn như Netflix, WhatsApp, Ebay,… tin chọn.

Dưới đây là một số đặc điểm và khái niệm quan trọng trong ReactJS:

* **Components (Các thành phần):** ReactJS xây dựng giao diện người dùng dựa trên các thành phần. Mỗi thành phần đại diện cho một phần của giao diện và có thể được xây dựng lại sử dụng lại hoặc kết hợp để tạo thành giao diện phức tạp hơn. Các thành phần trong ReactJS được xây dựng bằng cách sử dụng JavaScript và JSX (JavaScript XML), một cú pháp tương tự HTML.
* **Virtual DOM:** ReactJS sử dụng một cấu trúc dữ liệu gọi là Virtual DOM (DOM ảo) để tăng hiệu suất. Virtual DOM là một bản sao của DOM thực tế, nơi ReactJS thực hiện các thay đổi và cập nhật chỉ khi cần thiết. Sử dụng Virtual DOM giúp giảm thiểu số lượng các thay đổi trên DOM thực tế, giúp cải thiện hiệu suất và tốc độ của ứng dụng.
* **Unidirectional Data Flow (Luồng dữ liệu một chiều):** ReactJS sử dụng mô hình luồng dữ liệu một chiều, nghĩa là dữ liệu di chuyển từ thành phần cha xuống thành phần con. Điều này giúp duy trì sự nhất quán dữ liệu và dễ dàng theo dõi các thay đổi. ReactJS cũng hỗ trợ một quy tắc gọi là "props" để truyền dữ liệu từ thành phần cha xuống thành phần con.
* **JSX (JavaScript XML):** JSX là một phần mở rộng cú pháp của JavaScript, cho phép viết mã HTML tương tự trong JavaScript. JSX giúp tạo ra mã nguồn dễ đọc và dễ hiểu hơn khi xây dựng giao diện người dùng trong ReactJS.
* **Cập nhật giao diện tự động:** ReactJS sử dụng một thuật toán gọi là "reconciliation" để xác định các thay đổi trong giao diện và cập nhật chỉ những phần cần thiết. Điều này giúp giảm tải cho trình duyệt và cải thiện hiệu suất của ứng dụng.
* **Cộng đồng lớn và hỗ trợ mạnh mẽ:** ReactJS có một cộng đồng lớn và năng động, cung cấp nhiều tài liệu, thư viện và công cụ hỗ trợ phong phú. Nhiều framework và thư viện phổ biến khác như React Router, Redux, và Material-UI cũng được phát triển dựa trên ReactJS.

Bên cạnh ReactJS còn là những thư viện khác hỗ trợ cho việc xây dựng một giao diện đẹp và hiệu năng cao như:

* **Boostrap 5:** là một thư viện chưa các các component (thành phần) đã được xây dựng sẵn. Giúp giảm thời gian coding, cũng như đem lại cho trang web giao diện đẹp mắt hơn.
* **Axios:** là một HTTP client được phát triển trên đối tượng JavaScript, axios có thể sử dụng trong các ứng dụng front-end. Sử dụng Axios dễ dàng gửi đi các request HTTP bất đồng bộ hay các cơ sở dữ liệu để thực hiện các chức năng thêm, sửa, đọc, xóa.

### **1.5.2 Back-End**

Phía server của website sẽ được xử lý bởi ngôn ngữ Python và Django Rest Framework, củ thể sẽ được trình bày ở phần dưới

**1.5.2.1. Python**

**Python** là một ngôn ngữ lập trình mạnh mẽ và dễ học được phát triển vào năm 1991 bởi Guido van Rossum. Nó đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới và được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau.

**Ứng dụng của Python**

Python thường được sử dụng để phát triển trang web và phần mềm, tự động hóa tác vụ, phân tích dữ liệu và trực quan hóa dữ liệu. Vì tương đối dễ học, Python đã được nhiều người không phải là lập trình viên như kế toán và nhà khoa học áp dụng cho nhiều công việc hàng ngày, chẳng hạn như tổ chức tài chính.

**Phân tích dữ liệu và học máy**

Python đã trở thành một yếu tố chính trong khoa học dữ liệu, cho phép các nhà phân tích dữ liệu và các chuyên gia khác sử dụng ngôn ngữ này để thực hiện các phép tính thống kê phức tạp, tạo trực quan hóa dữ liệu, xây dựng thuật toán [học máy](https://glints.com/vn/blog/hoc-machine-learning/" \t "_blank), thao tác và phân tích dữ liệu cũng như hoàn thành các nhiệm vụ khác liên quan đến dữ liệu.

Python có thể xây dựng nhiều dạng trực quan hóa dữ liệu khác nhau, chẳng hạn như biểu đồ đường và thanh, biểu đồ hình tròn, biểu đồ 3D. Python cũng có một số thư viện cho phép các lập trình viên viết chương trình để phân tích dữ liệu và học máy nhanh hơn và hiệu quả hơn, như TensorFlow và Keras.

**Phát triển web**

Python thường được sử dụng để phát triển back-end của trang web hoặc ứng dụng—những phần mà người dùng không nhìn thấy. Vai trò của Python trong phát triển web có thể bao gồm gửi dữ liệu đến và đi từ máy chủ, xử lý dữ liệu và giao tiếp với cơ sở dữ liệu, định tuyến URL và đảm bảo tính bảo mật. Python cung cấp một số khuôn khổ để phát triển web. Những cái thường được sử dụng bao gồm Django và Flask.

Một số công việc phát triển web sử dụng Python bao gồm kỹ sư phụ trợ, nhà phát triển Python, kỹ sư phần mềm và kỹ sư DevOps.

**Tự động hoá và phát triển phần mềm**

Quá trình viết code được sử dụng để xây dựng các quy trình tự động này được gọi là viết script. Tự động hóa có thể được sử dụng để kiểm tra lỗi trên nhiều tệp, chuyển đổi tệp, thực hiện phép toán đơn giản và loại bỏ các bản sao trong dữ liệu.

Python thậm chí có thể được sử dụng bởi những người mới bắt đầu để tự động hóa các tác vụ đơn giản trên máy tính—chẳng hạn như đổi tên tệp, tìm và tải xuống nội dung trực tuyến hoặc gửi email hoặc văn bản theo khoảng thời gian mong muốn. Trong phát triển phần mềm. Với Python, các nhà phát triển phần mềm có thể tự động kiểm tra các sản phẩm hoặc tính năng mới. Một số công cụ Python được sử dụng để kiểm thử phần mềm bao gồm Green và Requestium.

**1.5.2.2. Django và Django Rest Framework**

**Django:** Django là một framework phát triển ứng dụng web mạnh mẽ và linh hoạt được viết bằng Python. Nó được phát triển bởi Django Software Foundation và có mục tiêu giúp nhà phát triển xây dựng các ứng dụng web nhanh chóng, hiệu quả và bảo mật.

Dưới đây là một số điểm nổi bật và khái niệm quan trọng trong Django:

* **Mô hình phát triển MVC:** Django tuân theo mô hình phát triển MVC, trong đó "Model" đại diện cho dữ liệu và quản lý truy xuất cơ sở dữ liệu, "View" đại diện cho giao diện người dùng và xử lý logic hiển thị, và "Controller" đại diện cho các thành phần xử lý yêu cầu và điều hướng. Sự phân chia rõ ràng này giúp quản lý mã nguồn dễ dàng hơn và tách biệt logic ứng dụng.
* **ORM (Object-Relational Mapping):** Django đi kèm với một ORM mạnh mẽ gọi là Django ORM, cho phép tương tác với cơ sở dữ liệu một cách dễ dàng và không phụ thuộc vào cú pháp SQL. ORM chuyển đổi các đối tượng trong Python thành bảng và hàng trong cơ sở dữ liệu, giúp tiết kiệm thời gian và công sức trong việc xử lý cơ sở dữ liệu.
* **Hệ thống xử lý URL:** Django cung cấp một hệ thống xử lý URL mạnh mẽ, cho phép định nghĩa các quy tắc URL và liên kết chúng với các hàm xử lý tương ứng. Điều này giúp dễ dàng xử lý các yêu cầu HTTP đến ứng dụng và điều hướng chúng đến các view tương ứng.
* **Quản lý phiên:** Django hỗ trợ quản lý phiên (session) một cách dễ dàng. Bằng cách sử dụng các phiên, có thể lưu trữ và truy xuất dữ liệu liên quan đến phiên làm việc của người dùng, như thông tin đăng nhập, giỏ hàng, và trạng thái đăng nhập.
* **Bảo mật:** Django có các tính năng tích hợp để giúp bảo vệ ứng dụng khỏi các lỗ hổng bảo mật phổ biến. Nó cung cấp xác thực người dùng, phân quyền, bảo vệ chống tấn công CSRF (Cross-Site Request Forgery), và các công cụ khác để giảm thiểu các rủi ro bảo mật.
* **Cộng đồng lớn và mở rộng:** Django có một cộng đồng rộng lớn, với nhiều tài liệu, thư viện và mô-đun bổ sung. Cộng đồng này đóng góp vào việc phát triển và cải tiến Django, cung cấp các tài liệu hướng dẫn, hỗ trợ và giải đáp thắc mắc.

**Django Rest Framework (DRF):** một framework mở rộng của Django, được sử dụng để phát triển các API (Application Programming Interface) RESTful trong Django. Nó cung cấp các công cụ và lớp trừu tượng để giúp xây dựng các API mạnh mẽ, dễ bảo trì và dễ mở rộng.

Dưới đây là một số điểm nổi bật và khái niệm quan trọng trong Django Rest Framework:

* **Serializer:** DRF cung cấp class Serializer để giúp chuyển đổi dữ liệu giữa các đối tượng Python và định dạng dữ liệu như JSON hoặc XML. Serializer cho phép định nghĩa các trường, quan hệ và kiểu dữ liệu để xác định cách dữ liệu sẽ được truyền đi và nhận về qua API.
* **Viewsets và Routers:** DRF hỗ trợ Viewsets và Routers để giúp xây dựng các API một cách dễ dàng. Viewsets là một cách tiện lợi để tổ chức các hành động CRUD (Create, Retrieve, Update, Delete) cho một tập hợp các đối tượng dữ liệu. Routers giúp tự động tạo ra các URL và liên kết chúng với các Viewsets.
* **Authentication và Permissions:** DRF cung cấp hỗ trợ cho xác thực người dùng và phân quyền. Bằng cách sử dụng các lớp Authentication và Permissions có sẵn, bạn có thể xác thực người dùng thông qua token, session hoặc OAuth, và xác định quyền truy cập vào các tài nguyên API.
* **Throttling:** DRF cung cấp chế độ giới hạn tốc độ (throttling) để kiểm soát tần suất yêu cầu từ các khách hàng. Điều này giúp bảo vệ API khỏi các cuộc tấn công DDoS và sử dụng tài nguyên hợp lý.
* **Pagination:** DRF cung cấp các lớp Pagination để giúp phân trang kết quả trả về từ API. Bằng cách sử dụng Pagination, bạn có thể giới hạn số lượng đối tượng trả về trong mỗi trang và cung cấp các liên kết dẫn đến các trang tiếp theo.
* **Định dạng dữ liệu:** DRF hỗ trợ nhiều định dạng dữ liệu phổ biến như JSON, XML, YAML và cả HTML. Điều này cho phép bạn dễ dàng tương tác với API từ các ứng dụng web, di động hoặc các công cụ khác.

Django Rest Framework là một công cụ mạnh mẽ để xây dựng các API RESTful trong Django. Nó cung cấp các tính năng quan trọng như Serializer, Viewsets, Authentication, Permissions và nhiều hơn nữa để giúp bạn xây dựng các API linh hoạt và dễ sử dụng. Với DRF, bạn có thể xây dựng các ứng dụng web hoặc di động phụ thuộc vào API một cách dễ dàng và hiệu quả.

### **1.5.3. Database**

Để xây dựng một trang web hoàn thiện, không thể không kể đến dữ liệu. Hệ thống sử dụng MySQL để lưu trữ dữ liệu:

**MySQL:** là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến. Nó được phát triển bởi Oracle Corporation và được sử dụng rộng rãi trên các ứng dụng web và hệ thống phân tán trên toàn thế giới. MySQL cho phép lưu trữ và quản lý dữ liệu trong các bảng có mối quan hệ với nhau. Nó hỗ trợ ngôn ngữ truy vấn SQL (Structured Query Language) để tương tác với cơ sở dữ liệu và thực hiện các thao tác như truy vấn, chèn, cập nhật và xóa dữ liệu.

**Đặc điểm MySQL:**

**Quản lý cơ sở dữ liệu quan hệ:** MySQL được thiết kế để quản lý và lưu trữ dữ liệu trong cơ sở dữ liệu quan hệ. Nó sử dụng ngôn ngữ truy vấn SQL (Structured Query Language) để tương tác với cơ sở dữ liệu và thực hiện các thao tác như truy vấn, chèn, cập nhật và xóa dữ liệu.

**Tính nhất quán và bảo mật:** MySQL cung cấp các tính năng nhất quán và bảo mật để đảm bảo tính toàn vẹn và an ninh của dữ liệu. Nó hỗ trợ giao thức ACID (Atomicity, Consistency, Isolation, Durability) để đảm bảo rằng các giao dịch được thực hiện một cách đáng tin cậy và an toàn.

**Đa nền tảng:** MySQL có sẵn trên nhiều nền tảng hệ điều hành như Windows, macOS và Linux. Điều này cho phép bạn triển khai cơ sở dữ liệu MySQL trên nền tảng phù hợp với yêu cầu của bạn.

**Tích hợp và mở rộng:** MySQL tích hợp tốt với các ngôn ngữ lập trình phổ biến như PHP, Java, Python và .NET, cho phép xây dựng ứng dụng web hoặc di động sử dụng MySQL làm cơ sở dữ liệu. Ngoài ra, MySQL có khả năng mở rộng linh hoạt, cho phép bạn xử lý tải lớn và mở rộng hệ thống dễ dàng.

**Tối ưu hiệu suất:** MySQL cung cấp các công cụ và cấu hình để tối ưu hiệu suất cơ sở dữ liệu. Nó hỗ trợ các chỉ mục, tối ưu câu truy vấn và có khả năng xử lý tải cao, giúp đảm bảo rằng ứng dụng của bạn chạy mượt mà và đáp ứng nhanh chóng.

**Cộng đồng và hỗ trợ:** MySQL có một cộng đồng sôi nổi và rộng lớn, với nhiều tài liệu, diễn đàn và nguồn tài nguyên trực tuyến.

### **1.5.4. Một số dịch vụ bên thứ 3**

Bên cạnh các công nghệ đã nêu ở trên, trang web còn tích hợp thêm các dịch vụ từ bên thứ 3 giúp lưu trữ và xử lí các nghiệp vụ. Cụ thể là:

**1.5.4.1. Fibase Storage**

**Fibase Storage** là dịch vụ lưu trữ đám mây của Google Firebase, giúp lưu trữ và quản lý các tệp tin, file trong ứng dụng di động hoặc web của mình. Trong đồ án này em sử dụng dịch vụ của Firebase Storage để lưu trữ ảnh trong hệ thống

**Đặc điểm Firebase Storege**

**Lưu trữ tệp tin đa phương tiện:** Firebase Storage cho phép lưu trữ các tệp tin như hình ảnh, video, âm thanh và tài liệu. Có thể tải lên và tải xuống các tệp tin này từ ứng dụng của mình.

**Tự động mở rộng và tin cậy:** Firebase Storage sử dụng cơ sở hạ tầng đám mây của Google, đảm bảo khả năng mở rộng tự động và tính sẵn sàng cao. Dịch vụ được quản lý và duy trì bởi Google, giúp chúng ta không phải lo lắng về việc quản lý cơ sở hạ tầng.

**Tích hợp dễ dàng với các dịch vụ khác của Firebase:** Firebase Storage tích hợp tốt với các dịch vụ khác của Firebase như Firebase Authentication và Firebase Realtime Database hoặc Firestore. Ta có thể sử dụng xác thực người dùng để kiểm soát quyền truy cập vào các tệp tin và lưu trữ dữ liệu tệp tin trong cơ sở dữ liệu Firebase.

**Quản lý metadata**: Firebase Storage cho phép lưu trữ và quản lý các thông tin metadata liên quan đến các tệp tin. Ta có thể tùy chỉnh các thuộc tính như tên tệp tin, loại tệp tin, kích thước, ngày tạo và các thuộc tính tuỳ chỉnh khác.

**Giới hạn kích thước và loại tệp tin:** Cóthể giới hạn kích thước và loại tệp tin được tải lên vào Firebase Storage. Điều này giúp bạn kiểm soát tài nguyên và bảo vệ hệ thống khỏi việc tải lên các tệp tin không hợp lệ hoặc quá lớn.

**1.5.4.2. VN Pay**

**VN Pay:** VNPay là một công ty cung cấp dịch vụ thanh toán điện tử tại Việt Nam. API thanh toán của VNPay cho phép các doanh nghiệp tích hợp các tính năng thanh toán vào ứng dụng hoặc trang web của họ. Ở đây em tích hợp API thanh toán của VNPay để thực hiện thanh toán các đơn hàng

**Đặc điểm VN Pay:**

**Tích hợp API:** VNPay cung cấp API thanh toán để các doanh nghiệp có thể tích hợp chức năng thanh toán vào ứng dụng hoặc trang web của mình. API này cho phép các giao dịch thanh toán trực tuyến thông qua các phương thức như thẻ ngân hàng, ví điện tử, hoặc chuyển khoản ngân hàng.

**Phương thức thanh toán đa dạng:** VNPay hỗ trợ nhiều phương thức thanh toán, bao gồm thanh toán qua thẻ ngân hàng, QR code, ví điện tử và chuyển khoản ngân hàng. Điều này cho phép người dùng có nhiều lựa chọn khi thực hiện thanh toán trực tuyến.

**Bảo mật và an toàn:** VNPay đảm bảo tính bảo mật và an toàn cho các giao dịch thanh toán. API thanh toán của VNPay sử dụng các tiêu chuẩn bảo mật cao như mã hóa dữ liệu, xác thực hai lớp và chứng chỉ SSL để đảm bảo rằng thông tin thanh toán được bảo vệ và không bị xâm phạm.

**Tích hợp dễ dàng:** VNPay cung cấp tài liệu hướng dẫn chi tiết và các ví dụ mã nguồn để giúp các doanh nghiệp tích hợp API thanh toán vào ứng dụng hoặc trang web của mình một cách dễ dàng. Các tài liệu này cung cấp các hướng dẫn về cách tạo yêu cầu thanh toán, xử lý phản hồi và kiểm tra trạng thái thanh toán.

**Quản lý giao dịch:** VNPay API cung cấp các chức năng quản lý giao dịch cho doanh nghiệp. Điều này bao gồm khả năng truy xuất lịch sử giao dịch, kiểm tra trạng thái thanh toán và xử lý hoàn trả.

**1.5.4.3. Brevo**

**Brevo:** là một dịch vụ gửi email giao dịch (transactional email) được cung cấp bởi một công ty có cùng tên. Dịch vụ gửi email giao dịch là một phương pháp để gửi các email tự động và cá nhân hóa cho người dùng dựa trên sự tương tác của họ với hệ thống hoặc ứng dụng. Trong đồ án này, em áp dụng dịch vụ gửi mail này cho việc gửi mã xác thực OTP

**Đặc điểm Brevo:**

**Gửi email tự động:** Brevo cho phép tự động gửi email dựa trên các sự kiện hoặc hành động cụ thể trong hệ thống. Ví dụ, khi người dùng đăng ký tài khoản, đặt hàng thành công, hoặc yêu cầu đặt lại mật khẩu, Brevo sẽ tự động gửi email thông báo cho họ.

**Thiết kế email linh hoạt:** Brevo cung cấp các công cụ để tạo và tùy chỉnh giao diện email. Ta có thể tạo các mẫu email chuyên nghiệp và cá nhân hóa chúng theo nhu cầu của mình. Brevo hỗ trợ cả HTML và các trình chỉnh sửa giao diện người dùng trực quan để ta có thể tạo nội dung email dễ dàng.

**Theo dõi và phân tích:** Brevo cung cấp các công cụ phân tích và theo dõi hiệu quả của các email giao dịch. Ta có thể xem số lượng email đã gửi, tỷ lệ mở email, tỷ lệ nhấp vào liên kết, và các thông tin khác để đánh giá hiệu suất và tối ưu hóa chiến dịch gửi email của mình.

**Tích hợp dễ dàng:** Brevo cung cấp API và các thư viện để tích hợp dịch vụ gửi email giao dịch vào ứng dụng một cách dễ dàng. Ta có thể gửi email thông qua Brevo từ bất kỳ ngôn ngữ lập trình nào và tích hợp với các hệ thống hiện có.

**Bảo mật và tin cậy:** Brevo cam kết đảm bảo an ninh và bảo mật cho các email giao dịch của bạn. Dịch vụ này sử dụng các biện pháp bảo mật để bảo vệ dữ liệu và ngăn chặn thư rác hoặc email giả mạo.

* 1. **Kết luận**

Trong chương 1, đồ án đã khảo sát nghiệp vụ và thu thập yêu cầu từ thực tế được rút ra từ nhiều trang bán hàng điện tử trực tuyến phổ biến tại Việt Nam. Từ những dữ liệu đó, chương này đưa ra mục tiêu và các chức năng chính của trang web. Đồng thời, nghiên cứu công nghệ phù hợp và giới thiệu chi tiết các ngôn ngữ cũng như thư viện, công cụ thiết kế cơ sở dữ liệu.

Phần giao diện xây dựng dựa trên các yếu tố chính là HTML, CSS và JavaScript được viết dưới sự hỗ trợ của thư viện ReactJS. Bên cạnh giao diện là phía server, hệ thống sử dụng DRF giúp xây dựng các ứng dụng mạng nhanh chóng và có thể mở rộng được một cách dễ dàng, là công cụ hoàn hảo cho các ứng dụng chuyên sâu về dữ liệu theo thời gian thực. Công cụ thiết kế cơ sở dữ liệu được lựa chọn đó là MySQL, bởi nó có cấu trúc bảng rất rõ ràng, dễ đọc và dễ hiểu. Bên cạnh đó là giới thiệu về một số dịch vụ bên thứ 3 mà đồ án sẽ sử dụng

Chương 2 sẽ đi vào giới thiệu về kiến trúc Microservice và hệ thống gợi ý sản phẩm sự dụng trong đồ án.

# **: Kiến trúc Microservice và hệ thống gợi ý**

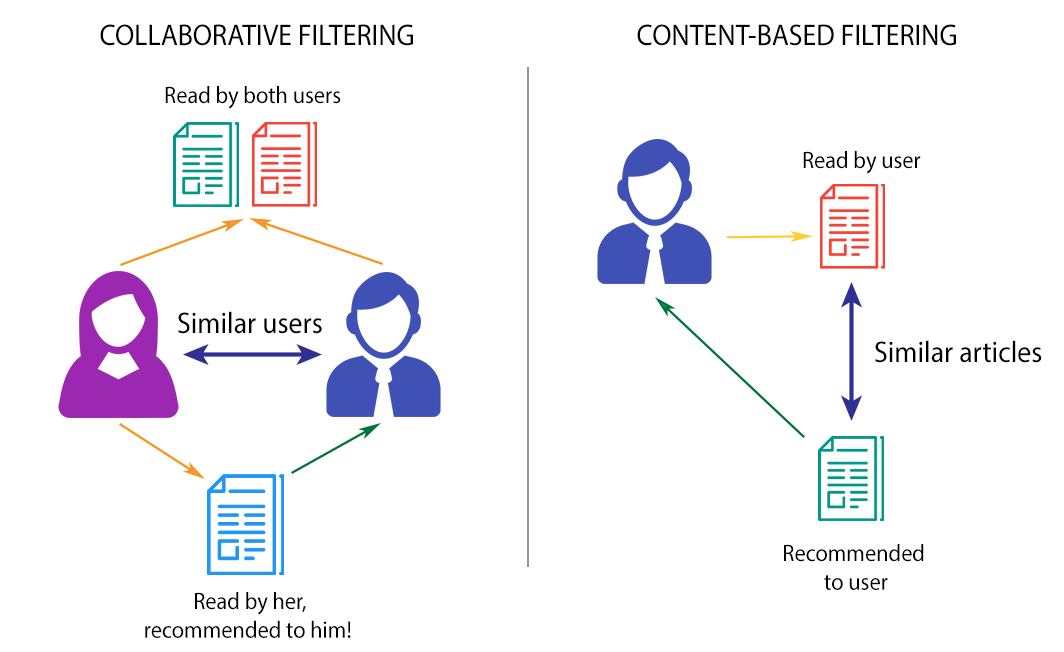
* 1. **Hệ thống gợi ý** 
     1. **Giới thiệu chung**

Hệ gợi ý là các công cụ phần mềm cung cấp các đề xuất sản phẩm có thể được quan tâm tới người dùng. Các đề xuất này có thể giúp ích cho quá trình đưa ra quyết định lựa chọn sản phẩm của người dùng, như lựa chọn địa điểm du lịch muốn đi, mặt hàng để mua, bài báo hay một bản nhạc, một hàng điện tử*…* nhằm gợi ý các mục thông tin có thể được quan tâm bởi người dùng. Hệ thống gợi ý sẽ đưa ra các gợi ý dựa trên quá trình thu thập, xử lý và phân tích dữ liệu từ người dùng.

Ngày nay, internet đang phát triển mạnh mẽ, vì thế lượng thông tin có sẵn về các sản phẩm (hàng điện tử, hàng điện tử, địa điểm du lịch, nhạc…) vô cùng lớn (tình trạng quá tải thông tin. Vì vậy hệ gợi ý chủ yếu nhắm tới các đối tượng người dùng chưa đủ năng lực hoặc kinh nghiệm cần thiết để ước lượng và tìm kiếm những sản phẩm tiềm năng cho nhu cầu của mình trong vô số các hạng mục tràn ngập trên. Một hệ gợi ý hàng điện tử sẽ đưa ra cho bạn những hàng điện tử mà nó cho rằng bạn muốn xem, hoặc hàng điện tử mà bạn có thể quan tâm và muốn xem, ta có một ví dụ rất nổi tiếng đó là Tiktok, Youtube, họ có một hệ thống gợi ý cực kỳ chính xác.

**Các kĩ thuật trong hệ thống gợi ý:**

* **Gợi ý dựa trên nội dung (Content-based):** là nhóm phương pháp thực hiện việc gợi ý dựa vào hồ sơ (profiles) của người dùng hoặc dựa vào các nội dung/thuộc tính của những sản phẩm tương tự như những sản phẩm mà người dùng đã chọn trong quá khứ. Ví dụ: một người dùng thường xem hàng điện tử thể loại điện thoại, vậy thì gợi ý một hàng điện tử về điện thoại. Ví dụ hàng điện tử “Iphone 14 Promax”*.*
* **Lọc cộng tác (Collaborative filtering):** là nhóm phương pháp gợi ý dựa trên sự tương quan giữa các người dùng hoặc các sản phẩm với nhau. Hay có thể hiểu rằng ở nhóm này một item được gợi ý tới một người dùng dựa trên những người dùng có hành vi tương tự. Ví dụ: người dùng Tuân, Minh, Dũng đều thích hàng điện tử “Iphone 14”. Hệ thống biết rằng Minh, Dũng cũng thích hàng điện tử “Iphone 15” nhưng chưa có thông tin về việc liệu Tuân có thích không. Dựa trên thông tin của những người dùng tương tự là Minh và Dũng, hệ thống dự đoán rằng Tuân cũng thích hàng điện tử “Iphone 15” và gợi ý hàng điện tử này cho Tuân.



Hình 2.1 So sánh collaborative và contentbased filtering

* + 1. **Ý tưởng thuật toán**
       1. **Thuật toán Collaborative Filtering**

Hệ thống gợi ý (Recommended system) ban đầu được bắt nguồn từ các sự việc đơn giản: người dùng thường dựa trên các gợi ý của người khác để đưa ra các quyết định bình thường hàng ngày, ví dụ họ sẽ thường tham khảo ý kiến của người khác nếu muốn mua một sản phẩm hay xem một hàng điện tử. Người dùng thường có xu hướng chọn những sản phẩm mà được đánh giá tốt từ mọi người.

Để mô tả hành vi đó, hệ gợi ý thường sử dụng các thuật toán để đưa ra các gợi ý tạo ra bởi một nhóm người dùng cho một người dùng khác đang có nhu cầu. Những gợi ý này đưa ra sản phẩm mà những người dùng khác có cùng sở thích đã chọn lựa, cách tiếp cận này gọi là lọc tương đồng hay tiếng anh là Collabolative filtering, cơ sở của nó là coi những người có sở thích giống nhau (do chọn lựa những sản phẩm tương đồng trong quá khứ) sẽ đưa ra dự đoán những lựa chọn trong tương lai cũng như nhau.

Các gợi ý được cung cấp cho người dùng dưới dạng đơn giản như một Danh sách điện tử đánh giá các sản phẩm, để có thể tạo ra được Danh sách điện tử đó, các hệ gợi ý sẽ dự đoán các sản phẩm thích hợp nhất với người dùng dựa trên sở thích, để làm được điều này, các hệ gợi ý thu thập thông tin liên quan tới người dùng: sở thích, các lần đánh giá về các sản phẩm, hoặc được suy luận từ các hành vi của người dùng đối với hệ thống. Kết quả gợi ý là các sản phẩm mà những người dùng tương tự đã thích, các sản phẩm này có thể hoàn toàn mới đối với người dùng được gợi ý. Phương pháp tiếp cận này được gọi là kỹ thuật lọc cộng tác.

* + - 1. **Thuật toán Neighborhood-based Collaborative Filtering**

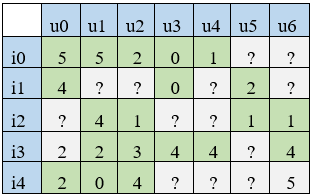
Phương pháp láng giềng hay còn gọi là kỹ thuật dựa bộ nhớ (Memory based). Ý tưởng cơ bản của NBCF là xác định mức độ quan tâm của một người dùng tới một sản phẩm dựa trên các người dùng khác có hành vi “gần giống” với người dùng hiện tại. Mức độ gần giống (khoảng cách) giữa các người dùng có thể được xác định thông qua mức độ quan tâm của các người dùng khác tới các sản phẩm mà hệ thống đã biết

Thuật toán gồm các kỹ thuật dựa người dùng (user based) tức dựa vào dữ liệu quá khứ của người dùng với các người dùng khác tương tự, và kỹ thuật dựa sản phẩm (item based) dựa trên dữ liệu quá khứ của các sản phẩm tương tự.

* + 1. **Nguyên lý cài đặt thuật toán**
       1. **Hàm tương tự - Similarity functions**

*2.2.1.1.1 Rating matrix là gì?*

Rating matrix là ma trận người dùng sản phẩm bao gồm cột là user, hàng là sản phẩm, trong đó các phần tử ở vị trí matrix[i,j] là số sao mà user[j] đã đánh đánh giá cho sản phẩm item[i]. Ví dụ trong trường hợp này, hàng ngang là các user từ u0 đến u6, sản phẩm coi như các hàng điện tử từ i0 đên i4.



Hình 2.2 Ma trận người dùng – rating tương ứng

Nhìn vào bảng dữ liệu trên ta thấy u0 và u1 đều thích i0 và đánh giá cho i0 là 5 sao. Mà u0 lại đánh giá khá thích i1 với số sao là 4 nên có thể dự đoán u1 cũng quan tâm đến i1.

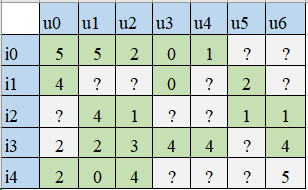
* + - 1. *Độ tương tự Cosine*

Mức độ tương tự giữa người dùng ui và uj kí hiệu là sim(ui,uj) được tính dựa vào tập các hàng điện tử mà 2 người đã đánh giá. Trong đồ án này, em sử dụng khoảng cách cosine hay độ tương đồng cosine (cosine similarity). Độ tương tự cosine là độ đo khá phổ biến để tính khoảng cách của 2 véc tơ có cùng độ dài, gọi 2 véc tơ đó lần lượt là A,B. [3]

Công thức của độ tương đồng cosine được tính như sau:

* + - 1. **Các bước thực hiện**
* Bước 1: Xây dựng ma trận đánh giá của hệ thống:[4]

Dựa vào thiết kế CSDL xây dựng được ma trận đánh giá



Hình 2.3 Ma trận người dùng – rating tương ứng

* Bước 2: Trung bình hóa dữ liệu hay chuẩn hóa dữ liệu

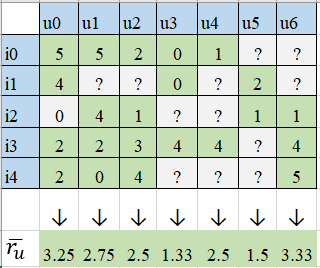
Một vấn đề là ta nên thay dấu ? bằng giá trị nào vì trong hệ thống luôn tồn tại những người khó tính và dễ tính. Thay bằng giá trị nào thì cũng không phù hợp với mọi người. Ví dụ người khó tính là đánh giá 3 sao với họ là thích, còn người dễ tính thì là 4,5 sao mới là thích. Vậy nên ta cần chuẩn hóa lại dữ liệu để có thể thay thế dấu ? với giá trị phù hợp với mọi người.

Theo ma trận ban đầu trên, những hàng điện tử chưa được đánh giá bởi người dùng người dùng nào đó thì đặt bằng 0 tức là những dấu ? sẽ được thay thế bằng 0. Như vậy, ta thấy đánh giá này còn kém hơn đánh giá kém nhất là 1 (với mức đánh giá từ 1 đến 5 với mỗi hàng điện tử).

Để xử lý chỗ này, ta sẽ làm thao tác gọi là trung bình hóa hay chuẩn hóa ma trận. Ta quy ước các dấu ? sẽ được thay thế bởi trung bình các đánh giá của 1 người dùng. Còn lại với mỗi đánh giá của người dùng sẽ lấy giá trị gốc trừ đi giá trị trung bình đánh giá. Giá trị trung bình đánh giá của người dùng được tính bằng công thức:

Trong đó: là giá trị đánh giá trung bình của người dùng u

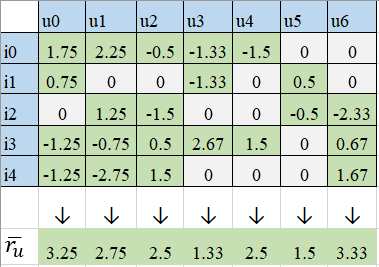
là Danh sách điện tử những hàng điện tửđã đánh giá của người dùng i, là đánh giá của người dùng i cho hàng điện tử thứ k trong Danh sách hàng điện tử đã đánh giá. Ví dụ với u0 ta có



Hình 2.4 Chuẩn hóa ma trận

Hàng cuối cùng là giá trị trung bình đánh giá của mỗi người dùng.  
Sau khi tính giá trị trung bình đánh giá của mỗi người dùng. Ta sẽ thay dấu ? là 0, còn các giá trị khác sẽ được tính bằng cách trừ đi giá trị đánh giá trung bình vừa tính

Ví dụ với u0,i0 ta có: 5 - 3.25 = 1.75



Hình 2.5 Ma trận sau khi chuẩn hóa

Việc trừ đi trung bình đánh giá khiến mỗi cột có giá trị âm và dương. Giá trị dương tương ứng với user thích hàng điện tử đó, giá trị âm tương tứng user không thích item đó, giá trị 0 tương ứng với việc chưa xác định được user có thích hàng điện tử đó hay không.

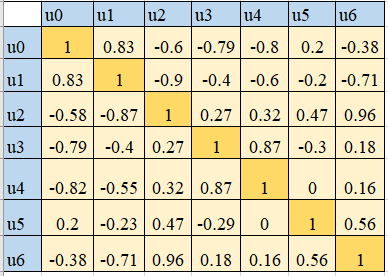
* Bước 3: Tính độ tương đồng giữa các người dùng.

Sau khi chuẩn hóa xong dữ liệu, ta thực hiện tính độ tương đồng giữa các người dùng bằng độ tương đồng cosine.

Ví dụ giữa u0 (1.75, 0.75, 0, -1.25, -1.25) và u1( 2.25, 0, 1.25, -0.75, -2.75). Ta có độ tương tự Cosine hay khoảng cách là:

similarity =

Tương tự với những người dùng khác ta được bảng sau:



Hình 2.6 Ma trận độ tương tự giữa các người dùng

* Bước 4: Dự đoán đánh giá người dùng A với hàng điện tử X với K người dùng đã chọn.

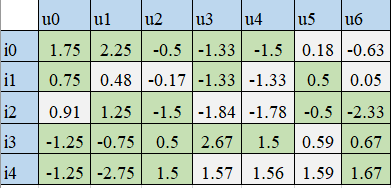
Ta chỉ quan tâm đến những người dùng đã đánh giá hàng điện tử X. Từ khoảng của A đến những người dùng đã đánh giá hàng điện tử X, ta chọn được K người dùng có khoảng cách đến người dùng A là nhỏ nhỏ nhất tức là chọn K người dùng tương tự với A nhất mà đã đánh giá hàng điện tử X. Công thức phổ biến được sử dụng để dự đoán đánh giả của người dùng A cho hàng điện tử X là:

Trong đó:  
 đánh giá của người dùng A với hàng điện tử X  
 đánh giá của người dùng uj với hàng điện tử X  
: Tập hợp K người dùng có khoảng cách đến X là nhỏ nhất hay độ tương tự lớn nhất đã đánh giá hàng điện tử A.  
: độ tương tự cosine giữa người dùng A và uj.

* Bước 5: Cộng lại giá trị vừa tính với trung bình đánh giá vừa tính ở trên là ra dự đoán đánh giá của người dùng A với hàng điện tử X.  
  Ví dụ: Với người dùng u0 chưa đánh giá hàng điện tử i2. Ta thực hiện theo các bước của thuật toán để dự đoán đánh giá của người dùng u0 với hàng điện tử i2. Với k=2:
* Các user đã đánh giá hàng điện tử i2: u1, u2, u5, u6
* Chọn ra 2 user với độ tương tự lớn nhất theo ma trận độ tương tự: sim(u0,u1)= 0.83, sim(u0,u2)= -0.58, sim(u0,u5)= 0.2, sim(u0,u6)= -0.38 => Chọn u1 và u5
* Giá trị đánh giá với hàng điện tử i2 tương ứng với u1, u5 là 1.25 và -0.5
* Tính toán theo công thức:

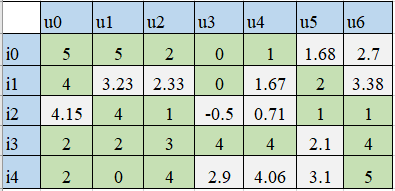
evaluate = = 091

Sau đó, sẽ cộng lại giá trị vừa tính với giá trị trung bình là ra dự đoán đánh giá của u0 với hàng điện tử i2 là: 0.91 + 3.25 = 4.18 , làm tròn là 4 sao.  
Có thể làm tương tự với những người dùng còn lại để dự đoán đánh giá cho tất cả người dùng, ta được ma trận sau:



Hình 2.7 Ma trận sau khi đự đoán đánh giá

Sau đó cộng các giá trị này với đánh giá trung bình là ra đánh giá dự đoán:



Hình 2.8 Ma trận đánh giá cuối cùng sau khi chuẩn hóa lại

* Bước 6: Dựa vào ma trận đánh giá sẽ đưa ra Danh sách điện tử các sản phẩm có rating cao (thể hiện độ yêu thích của người dùng với sản phẩm), ở đây em sẽ lấy có sản phẩm có độ rating >= 3.
  1. **Kiến trúc microservice**
     1. **Một số kiến trúc phần mềm truyền thống**
        1. **Kiến trúc nguyên khổi**

Trong công nghệ phần mềm, thuật ngữ monolithic được sử dụng để mô tả cho một loại hình kiến trúc mà ở đó các thành phần của hệ thống được xây dựng và nằm trong một khối duy nhất không thể chia tách. Vì đặc điểm này, kiến trúc monolithic còn được gọi là kiến trúc nguyên khối hay kiến trúc một khối.

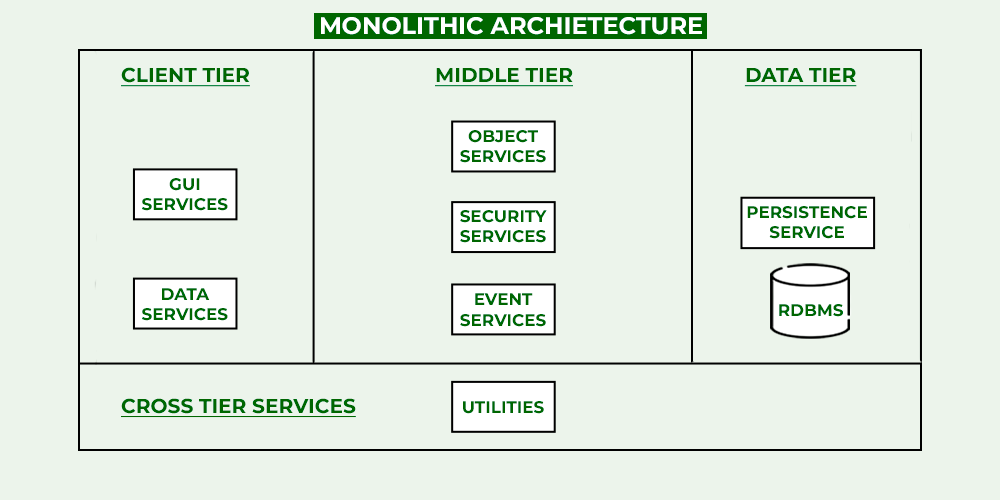
Thông thường, một ứng dụng theo mô hình kiến trúc một khối sẽ được phân tách thành các tầng hoặc lớp sau:

• Tầng giao diện người dùng: Đây là tầng quen thuộc với người dùng, nó có thể là một trang web hoặc một ứng dụng. Tầng này sẽ tiếp nhận dữ liệu từ phía người dùng thông qua form giao diện, tương tác với máy chủ (server) và trả lại kết quả cho người dùng

• Tầng dịch vụ hay còn gọi là Tầng giữa (Middle Tier): Tầng này thực hiện nhiệm vụ xử lí các logic nghiệp vụ, nhận dữ liệu từ Client, xử lí chúng, tầng này cũng tương tác với tầng truy cập dữ liệu để thực hiện các giao tác như truy vấn hoặc lưu trữ dữ liệu

• Tầng dữ liệu: Nó bao gồm cơ chế lưu giữ dữ liệu (DB) và liên lạc với các ứng dụng khác. Nó bao gồm cơ sở dữ liệu, hàng đợi tin nhắn, v.v.

Cả ba thành phần này được xây dựng, đóng gói và triển khai trong một khối duy nhất.



Hình 2.9 Kiến trúc Monolithic

Kiến trúc nguyên khối cho ứng dụng web Kiến trúc nguyên khối được sử dụng phổ biến cho hầu hết các ứng dụng ở mức vừa và nhỏ. Đây được xem là một giải pháp truyền thống từ trước tới nay để xây dựng ứng dụng bởi lẽ kiến trúc này đem lại một số ưu điểm nhất định như mô tả bên dưới.

**Ưu điểm của kiến trúc nguyên khối**:

* **Dễ dàng phát triển:** về mặt cấu trúc, chương trình được chia tách thành các gói dịch vụ riêng biệt hay còn gọi là các module. Các module này sẽ được tổ chức thành một khối duy nhất trong cùng một cấu trúc mã nguồn. Về mặt kỹ thuật, kiến trúc nguyên khối cho phép sử dụng thống nhất các công nghệ ở các tầng. Vì thế, việc phát triển ứng dụng theo cách này được xem là đơn giản, dễ dàng và tốn ít thời gian.
* **Dễ dàng triển khai và vận hành:** các module của chương trình được đóng gói và cài đặt thành một khối duy nhất, vì vậy việc triển khai và vận hành ứng dụng sẽ đơn giản.
* **Dễ kiểm thử:** Việc các module nằm trên môt khối duy nhất giúp cho việc kiểm thử từng phần cũng như kiểm thử tích hợp vô cùng dễ dàng

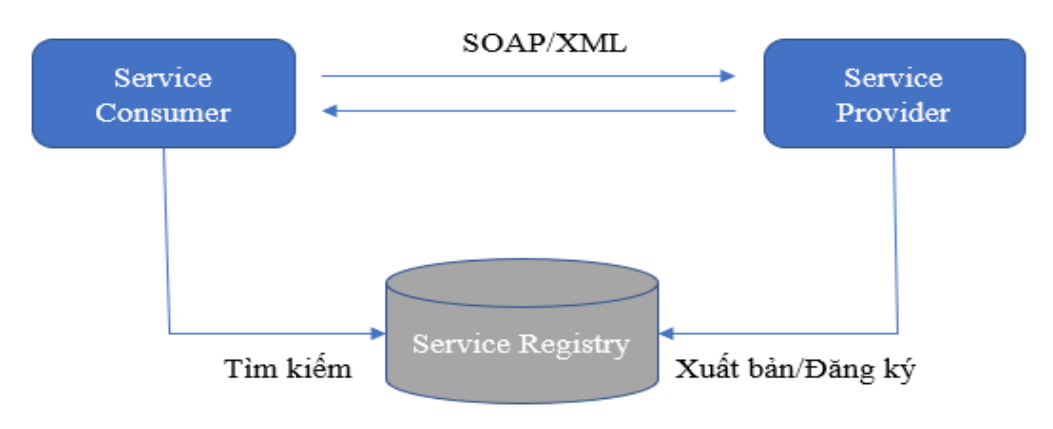
**Nhược điểm của kiến trúc nguyên khối**:

* **Khó đáp ứng khả năng thay đổi:** trong kiến trúc nguyên khối, các thành phần gắn kết với nhau một cách chặt chẽ và được tổ chức thành một khối duy nhất, không thể tách rời. Chính điều này sẽ gây khó khăn cho việc thay đổi (thêm hoặc bớt) các tính năng về nghiệp vụ, cũng như sự nâng cấp các yêu cầu về mặt công nghệ.
* **Khả năng chịu lỗi thấp:** vì các thành phần liên kết chặt chẽ với nhau, nên mỗi khi có sự thay đổi ở một module này sẽ kéo theo sự thay đổi ở một module khác. Chính điều này làm giảm đi khả năng chịu lỗi của hệ thống.
* **Khó khăn trong việc bảo trì:** theo thời gian, kích cỡ chương trình gia tăng để áp ứng các yêu cầu thay đổi về nghiệp vụ. Việc bảo trì, tìm và sửa lỗi trong một khối mã nguồn lớn sẽ là một thách thức không nhỏ cho các nhà phát triển.
* **Giới hạn về mặt công nghệ:** ứng dụng theo hướng nguyên khối gây khó khăn cho việc cập nhật, nâng cấp các công nghệ.
  + - 1. **Kiến trúc hướng dịch vụ (SOA)**

Kiến trúc hướng dịch vụ (SOA) là một phương pháp phát triển phần mềm sử dụng các thành phần của phần mềm được gọi là dịch vụ để tạo ra các ứng dụng dành cho doanh nghiệp. Mỗi dịch vụ cung cấp một tính năng doanh nghiệp, đồng thời các dịch vụ cũng có thể giao tiếp với nhau giữa nhiều nền tảng và ngôn ngữ thống qua các giao diện lập trình ứng dụng (API). Nhà phát triển tận dụng SOA để tái sử dụng các dịch vụ trong nhiều hệ thống khác nhau hoặc kết hợp một số dịch vụ độc lập để thực hiện các tác vụ phức tạp.

Mô hình cơ bản của một hệ thống SOA bao gồm ba thành phần:

* Thành phần sử dụng dịch vụ (service consumer): thành phần này có thể là một chương trình ứng dụng hoặc là một dịch vụ khác có yêu cầu sử dụng dịch vụ. Nó sẽ thực hiện việc xác định các dịch vụ trong kho đăng ký dịch vụ và thực thi nhiệm vụ bằng cách gửi một yêu cầu tới phía cung cấp dịch vụ.
* Thành phần cung cấp dịch vụ (service provider): nhiệm vụ chính của nó là chấp nhận và thực thi các yêu cầu được gửi tới từ phía sử dụng dịch vụ. Các dịch vụ do phía cung cấp dịch vụ xuất bản sẽ được đăng ký tại kho lưu trữ dịch vụ để cho phía sử dụng dịch vụ có thể truy cập được.
* Thành phần đăng ký dịch vụ (service registry): đây là nơi lưu trữ các dịch vụ để phục vụ cho bên sử dụng dịch vụ



Hình 2.10 Mô hình hoạt động hệ thống SOA

Trong mô hình hoạt động ở trên, việc trao đổi dữ liệu giữa ba bên được thực hiện thông qua một chuẩn về giao thức (SOAP) và một quy ước về chuẩn dữ liệu (XML). Bên cạnh SOAP/XML, hiện nay các chuẩn về REST và JSON được sử dụng phổ biến hơn nhờ các tính ưu việt của nó như sự đơn giản trong cấu trúc cũng như việc bóc tách dữ liệu dễ dàng

**Ưu điểm của kiến trúc hướng dịch vụ**:

* **Rút ngắn thời gian đưa ra thị trường:** Nhà phát triển tái sử dụng các dịch vụ trên những quy trình kinh doanh khác nhau để tiết kiệm thời gian và chi phí. Họ có thể hợp dịch các ứng dụng nhanh hơn bằng SOA so với việc lập trình và thực hiện tích hợp từ đầu.
* **Bảo trì hiệu quả** Các dịch vụ nhỏ dễ tạo dựng, cập nhật và khắc phục lỗi hơn những đoạn mã lớn trong ứng dụng đơn khối. Việc sửa đổi bất kỳ dịch vụ nào trong SOA cũng không làm ảnh hưởng đến chức năng tổng thể của quy trình kinh doanh.
* **Khả năng thích ứng cao**: SOA dễ thích ứng hơn với những cải tiến về công nghệ. có thể hiện đại hóa các ứng dụng của mình một cách hiệu quả và tiết kiệm. Ví dụ: các tổ chức chăm sóc sức khỏe có thể sử dụng chức năng của các hệ thống hồ sơ y tế điện tử cũ hơn vào những ứng dụng mới trên nền tảng đám mây.

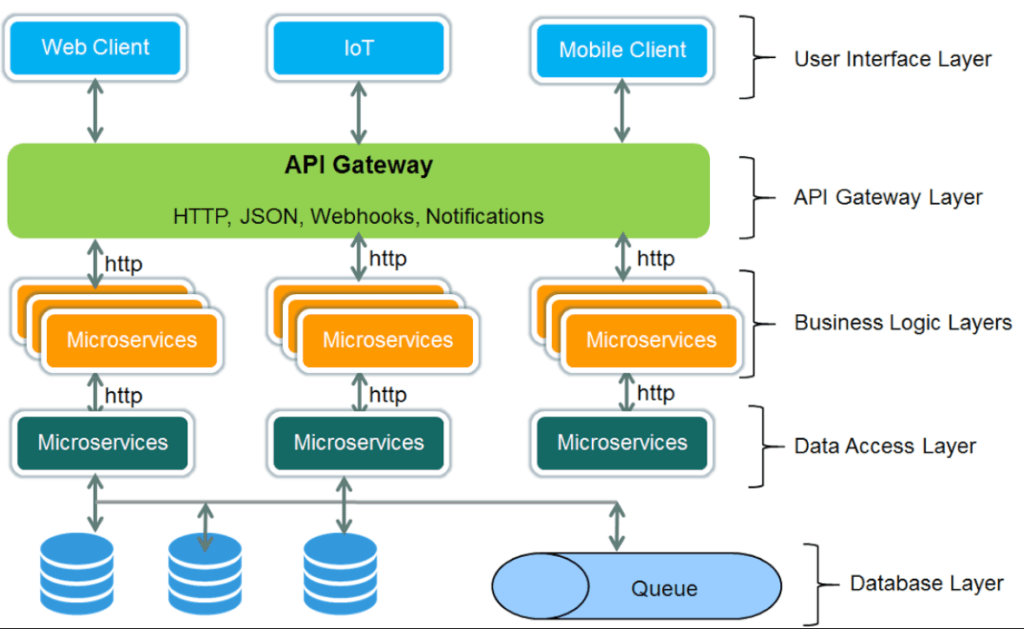
**Nhược điểm của kiến trúc hướng dịch vụ**:

* **Phức tạp trong việc quản lý các dịch vụ:** Hệ thống SOA bao gồm nhiều dịch vụ hoạt động trao đổi và tương tác với nhau bằng cách gửi và nhận các thông điệp. Khi số lượng các dịch vụ tăng lên đi kèm với một lượng lớn các thông điệp được trao đổi thì việc đảm bảo sự hoạt động thông suốt của hệ thống là một thử thách không nhỏ.
* **Chi phí cao cho quá trình phát triển:** Bản thân kiến trúc SOA đã bao gồm sự phức tạp trong việc tổ chức các thành phần, các tầng/lớp trong kiến trúc. Chính vì thế, việc phát triển và triển khai hệ thống SOA sẽ đòi hỏi một chi phí cao hơn nhiều cho nguồn lực cũng như công nghệ (phần mềm, phần cứng) so với việc phát triển các hệ thống thông thường.
  + 1. **Sơ lược về kiến trúc microservice**

Microservices cung cấp cho các nhà phát triển một hướng tiếp cận mới trong việc xây dựng phần mềm theo hướng dịch vụ bằng cách tổ chức các thành phần phần mềm thành những module nhỏ. Các module này được coi là những ứng dụng riêng rẽ, được triển khai độc lập và giao tiếp với nhau thông qua các API. Vì những đặc tính này, có nhiều quan điểm cho rằng microservices thực chất là một bước tiến hóa tiếp theo của SOA

Có nhiều định nghĩa khác nhau về microservices tùy theo ngữ cảnh và hướng tiếp cận. Theo tác giả James Lewis và Martin Fowler, microservices được hiểu như sau: “*Microservices là một hướng tiếp cận trong việc phát triển một ứng dụng nguyên khối thành một tập các dịch vụ nhỏ, mỗi một dịch vụ này chạy trong một tiến trình riêng của nó và giao tiếp với nhau thông qua cơ chế mềm dẻo – thường là qua HTTP, giao diện lập trình ứng dụng (API).”* [5]

Như vậy, bản chất của microservices chính là SOA và nó mang các đặc điểm của SOA. Cách tiếp cận theo hướng microservices tập trung vào việc chia nhỏ một ứng dụng lớn thành nhiều dịch vụ nhỏ, và các dịch vụ này có thể kết nối, giao tiếp được với nhau. Mỗi một dịch vụ được xem như một chương trình ứng dụng độc lập, thực hiện một chức năng riêng biệt trong miền nghiệp vụ mà nó đảm nhiệm.



Hình 2.11 Mô hình kiến trúc Microservice

**Đặc điểm của kiến trúc Microservice:**

Khi mô tả về microservices, người ta đề cập tới một số tính chất quan trọng sau:

**Tính đơn nhiệm:** Về mặt thiết kế, mỗi một dịch vụ trong hệ thống microservices chỉ tập trung vào giải quyết và hoàn thành một tác vụ trong miền nghiệp vụ của bài toán mà nó đảm nhiệm. Đặc điểm này vì thế được gọi là sự đơn nhiệm. Nhìn ở khía cạnh khác, các dịch vụ trong hệ thống được chia thành nhiều dịch vụ nhỏ. Khái niệm “nhỏ” về mặt chức năng, nghiệp vụ ở đây cũng chính là yếu tố dẫn đến việc triển khai về mặt mã nguồn (lập trình). Mỗi một dịch vụ sẽ sử dụng một cấu trúc mã nguồn nhỏ, và nó có thể dễ dàng được phát triển cũng như triển khai trong một thời gian ngắn.

**Tính đóng gói:** Mỗi một microservice sẽ chỉ nên thực hiện một nhiệm vụ riêng biệt, và nó sẽ đóng gói toàn bộ dữ liệu cũng như chi tiết về mặt cài đặt trong từng dịch vụ. Một cách lý tưởng, giới chuyên gia cho rằng mỗi một microservice chỉ nên sử dụng một cơ sở dữ liệu riêng cho nó, và các dữ liệu này được giữ ở mức riêng tư. Đặc điểm của sự đóng gói này đảm bảo sự phụ thuộc lỏng lẻo giữa các dịch vụ trong toàn hệ thống.

**Khả năng tự trị:** Khả năng này đảm bảo cho sự độc lập trong quá trình triển khai và phát triển hệ thống. Điều đó có nghĩa là mỗi một microservice có thể được xây dựng độc lập bởi một đội ngũ phát triển, và được triển khai cài đặt trên từng máy riêng biệt.

**Tính đàn hồi:** Đặc tính này đảm bảo trường hợp nếu một module này bị trục trặc thì sự hỏng hóc đó là cô lập với các thành phần khác, và các dịch vụ khác vẫn có thể hoạt động được mà ít chịu ảnh hưởng. Sự đàn hồi là yếu tố làm tăng khả năng chịu lỗi cho hệ thống.

**Sự mềm dẻo trong khả năng mở rộng:** Kiến trúc microservices đáp ứng linh hoạt cho việc mở rộng hệ thống. Mỗi khi có các yêu cầu bổ sung các tính năng mới về chức năng nghiệp vụ hoặc cần sự nâng cấp về mặt công nghệ, quá trình cập nhật này có thể được tiến hành độc lập cho từng module phần mềm mà không làm ảnh hưởng tới các thành phần khác.

**Ưu điểm của kiến trúc Microservice:**

**Tính mô-đun cao:** Mô hình microservice cho phép phân tách ứng dụng thành các microservice nhỏ, mỗi microservice chỉ chịu trách nhiệm cho một chức năng cụ thể. Điều này giúp tăng tính mô-đun của hệ thống và giảm độ phức tạp toàn cầu của ứng dụng.

**Độc lập và mở rộng linh hoạt:** Với kiến trúc microservice, mỗi microservice có thể được phát triển, triển khai và mở rộng độc lập. Điều này cho phép các nhóm phát triển làm việc độc lập với nhau và mở rộng chỉ các thành phần cần thiết, giúp tăng khả năng mở rộng của hệ thống.

**Sử dụng công nghệ đa dạng:** Mô hình microservice không ràng buộc việc sử dụng một ngôn ngữ lập trình hay công nghệ cụ thể nào. Mỗi microservice có thể được xây dựng bằng ngôn ngữ và công nghệ phù hợp với nhiệm vụ của nó. Điều này giúp tận dụng tốt các công nghệ mới và phù hợp với yêu cầu cụ thể của từng microservice.

**Dễ dàng bảo trì và cập nhật:** Với mô hình microservice, việc bảo trì và cập nhật trở nên dễ dàng hơn. có thể tập trung vào việc chỉnh sửa, nâng cấp hoặc sửa lỗi cho một microservice cụ thể mà không ảnh hưởng đến toàn bộ hệ thống. Điều này giúp giảm thời gian và rủi ro khi triển khai các thay đổi.

**Nhược điểm của kiến trúc Microservice:**

**Độ phức tạp lớn:** Với kiến trúc microservice, hệ thống sẽ được chia thành nhiều microservice nhỏ. Điều này dẫn đến sự phức tạp trong việc quản lý giao tiếp và theo dõi các microservice.

**Quản lý và triển khai phức tạp:** Với nhiều microservice độc lập, việc quản lý và triển khai trở nên phức tạp hơn. Mỗi microservice yêu cầu quy trình triển khai và quản lý riêng biệt, và việc đảm bảo tính nhất quán giữa các phiên bản microservice cũng là một thách thức.

**Tăng độ phức tạp của kiểm thử và gỡ lỗi:** Với sự phân tách chức năng và dữ liệu, việc kiểm thử và gỡ lỗi trong môi trường microservice trở nên phức tạp hơn. Điều này đòi hỏi các bộ kiểm thử phải xử lý các vấn đề như sự phụ thuộc giữa các microservice, giao tiếp giữa chúng và các vấn đề liên quan đến dữ liệu.

**Tăng chi phí phát triển và vận hành:** Triển khai kiến trúc microservice đòi hỏi sự đầu tư lớn vào việc phát triển và vận hành. Cần có nhóm phát triển và quản lý chuyên biệt để xây dựng, triển khai và duy trì các microservice. Điều này có thể dẫn đến tăng chi phí và đòi hỏi sự đầu tư lâu dài.

**Khó khăn trong việc quản lý sự nhất quán dữ liệu:** Với mỗi microservice sở hữu cơ sở dữ liệu riêng của mình, việc duy trì sự nhất quán dữ liệu trở nên phức tạp hơn. Điều này yêu cầu sự quản lý cẩn thận và các biện pháp đồng bộ hóa dữ liệu giữa các microservice.

**Độ trễ và hiệu suất:** Sự phân tách chức năng và giao tiếp thông qua mạng có thể gây ra độ trễ và ảnh hưởng đến hiệu suất của hệ thống. Việc đảm bảo sự ổn định và tối ưu hóa hiệu suất trở thành một thách thức trong kiến trúc microservice.

* + 1. **Nguyên tắc thiết kế kiến trúc Microservice**

Khi thiết kế kiến trúc microservice, có một số nguyên tắc quan trọng cần được tuân thủ để đảm bảo hiệu quả và thành công của hệ thống. Dưới đây là một số nguyên tắc thiết kế kiến trúc microservice:

* **Phân tách chức năng:** Chia ứng dụng thành các microservice nhỏ độc lập, mỗi microservice chỉ đảm nhận một chức năng cụ thể. Điều này giúp giảm độ phức tạp của mỗi microservice và tăng tính mô-đun của hệ thống.
* **Giao tiếp thông qua giao thức nhẹ:** Sử dụng các giao thức nhẹ như HTTP hoặc message queue để giao tiếp giữa các microservice. Điều này giúp giảm sự phụ thuộc và tăng tính linh hoạt của hệ thống.
* **Quản lý độc lập:** Mỗi microservice nên có khả năng triển khai, cập nhật và mở rộng độc lập. Điều này cho phép các nhóm phát triển làm việc độc lập và giảm sự ràng buộc giữa các microservice.
* **Quản lý dữ liệu độc lập:** Mỗi microservice nên có cơ sở dữ liệu riêng của nó. Điều này giúp đảm bảo tính độc lập và cô lập của dữ liệu, tránh sự phụ thuộc và tác động tiêu cực giữa các microservice.
* **Tách biệt trách nhiệm**: Mỗi microservice nên có trách nhiệm rõ ràng và giới hạn. Điều này giúp tăng tính rõ ràng và dễ bảo trì của hệ thống.
* **Quản lý lỗi và khả năng khôi phục:** Xây dựng cơ chế quản lý lỗi và khôi phục cho mỗi microservice. Điều này giúp hệ thống chịu được lỗi và có khả năng tự phục hồi khi xảy ra sự cố.
* **Giám sát và theo dõi:** Sử dụng các công cụ và kỹ thuật giám sát để theo dõi hoạt động của các microservice và sự tương tác giữa chúng. Điều này giúp phát hiện và giải quyết sự cố một cách nhanh chóng và hiệu quả.
* **Chỉ triển khai những gì cần thiết:** Triển khai và mở rộng chỉ những microservice cần thiết để đáp ứng yêu cầu hiện tại. Điều này giúp tối ưu hóa tài nguyên và đảm bảo tính linh hoạt của hệ thống.

Tuy nhiên, nguyên tắc thiết kế kiến trúc microservice có thể thay đổi tùy thuộc vào yêu cầu cụ thể của dự án. Nên xem xét và điều chỉnh nguyên tắc thiết kế để phù hợp với môi trường và yêu cầu của dự án cụ thể.

## **3.3. Kết luận**

Trong chương này, chúng ta đã tìm hiểu về hệ thống gợi ý, đi sâu hơn về phương pháp Neighborhood-based Collaborative Filtering trong hệ gợi ý. Ta đã tìm hiểu ý tưởng thuật toán, nguyên lý và chi tiết các bước cài đặt thuật toán. Từ đó có thể áp dụng gợi ý sản phẩm cho người dùng. Để hoạt động ổn định và chính xác thì thuật toán cần khá nhiều dữ liệu lịch sử đánh giá của người dùng.

Chúng ta cũng tìm hiểu sâu hơn về các mô hình phát triển phần mềm phổ biến là mô hình nguyên khối, môi hình SOA, và đặt biệt là mô hình microservice. Ta cũng nắm được đặc điểm, các ưu nhược điểm và một số nguyên tắc khi triển khai mô hình Microservice vào các dự án thực tế.

Trong chương tiếp theo, ta sẽ trình bày về phân tích và thiết kế hệ thống thương mại điện tử.

# **: Phân tích và thiết kế hệ thống**

* 1. **Xác định và mô tả các tác nhân:**

Bảng 3.1 Xác định các tác nhân

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên Actor | Mô tả | Hệ thống tương tác |
| 1 | Admin (Nhân viên quản lý) | Người quản trị hệ thống sẽ quản lý những thông tin về hàng điện tử, bộ sưu hàng điện tử(collection), thể loại, gói dịch vụ của hệ thống, tài khoản, bài viết, các giao dịch phát sinh của hệ thống, thống kê doanh thu lượt mua của hệ thống trong một khoảng thời gian. | Ứng dụng trên web |
| 2 | Member (Thành viên hệ thống như người người mua hàng) | Người dùng truy web dành cho thành viên hệ thống, có thể sử dụng những chức năng dành cho người dùng như: Đăng ký, đăng nhập, xem Danh sách hàng điện tử, tìm kiếm hàng điện tử, yêu thích, đánh giá, bình luận hàng điện tử,… | Ứng dụng trên web |

* 1. **Xây dựng bảng thuật ngữ**
* Mục đích: Nhằm làm sáng tỏ các thuật ngữ được sử dụng cho một miền nào đó để người đọc hiểu trong quá trình phát triển phần mềm. [6]
* Một số thuật ngữ được dùng trong hệ thống:

Bảng 3.2 Các thuật ngữ dùng trong hệ thống

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tiếng Anh | Tiếng Việt | Giải thích nội dung |
| 1 | User | Người dùng | Người dùng truy web dành cho thành viên hệ thống, có thể sử dụng những chức năng dành cho người dùng như: Đăng ký, đăng nhập, xem Danh sách hàng điện tử, tìm kiếm hàng điện tử, yêu thích, đánh giá, bình luận hàng điện tử,… |
| 2 | Admin | Nhân viên quản lý | Người quản trị hệ thống sẽ quản lý những thông tin về hàng điện tử, bộ sưu hàng điện tử, thể loại, gói dịch vụ của hệ thống, tài khoản, các giao dịch phát sinh của hệ thống, thống kê doanh thu và thống kê lượt xem của hệ thống trong một khoảng thời gian. |
| 3 | Comment | Bình luận | Chứa bình luận của người dùng theo thời gian |
| 4 | Favourite | Mục yêu thích | Gồm các hàng điện tử được người dùng thêm vào Danh sách điện tử yêu thích |
| 5 | Product | Hàng điện tử | Gồm các thông tin về hàng điện tử như tiêu đề, màu sắc, kích thước, ngày phát hành,… |
| 6 | Blog | Bài viết | Là các bài viết đánh giá về một sản phẩm |
| 7 | Category | Thể loại | Dùng để phân loại các hàng điện tử thuộc thể loại như: Điện thoại, Tivi, Máy ảnh,… |
| 8 | Brand | Thương hiệu | Gồm các thông tin về thương hiệu như tên, hình ảnh,… |
| 9 | Collection | Bộ sưu tập | Là tên 1 nhóm sản phẩm chung 1 đặc điểm về thời gian ra mắt, thương hiệu, …Ví dụ: Iphone 15 Series |
| 10 | Rating | Đánh giá | Chứa thông tin đánh giá sao của người dùng |
| 11 | Cart | Giỏ hàng | Chứa thông tin sản phẩm và số lượng sản phẩm mà người dùng có ý định mua |
| 12 | Payment | Thanh toán | Gồm phương thức sẽ thanh toán cho 1 đơn hàng, có thể là online hoặc offline |

* 1. **Xác định và mô tả các ca sử dụng**
     1. **Danh sách các usecase cho người dùng**

Bảng 3.3 Danh sách usecase của người dùng

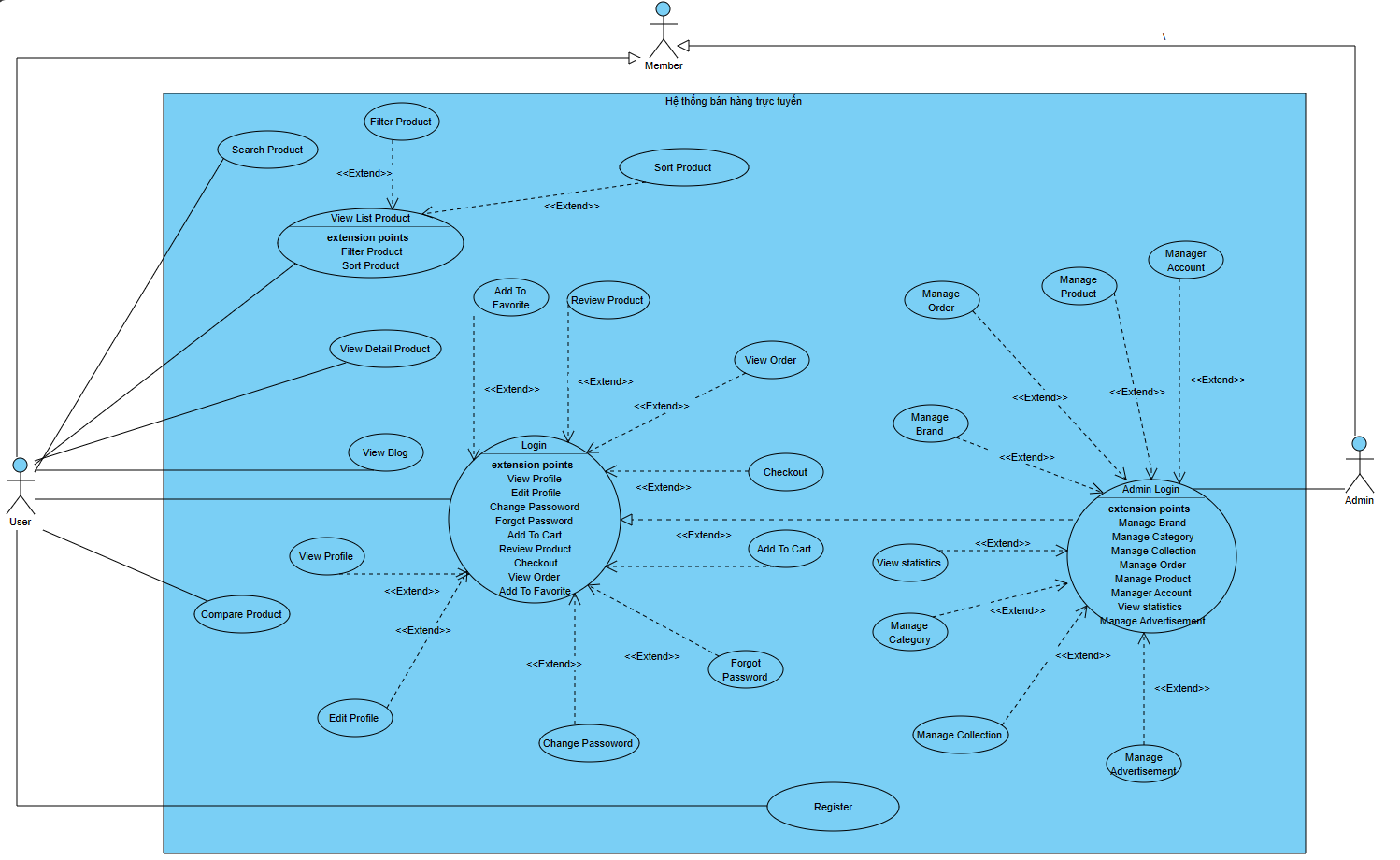
|  |  |  |
| --- | --- | --- |
| STT | Tên usecase | Mô tả usecase |
| 1 | Register (Đăng ký) | Người dùng đăng ký tài khoản trên hệ thống |
| 2 | Login (Đăng nhập) | Người dùng đăng nhập vào hệ thống |
| 3 | Forgot password (Quên mật khẩu) | Người dùng quên mật khẩu của tài khoản |
| 4 | Change password (Đổi mật khẩu) | Người dùng đổi mật khẩu trên hệ thống |
| 5 | View profile (Xem thông tin cá nhân) | Người dùng xem thông tin cá nhân của mình |
| 5 | Change profile (Thay đổi thông tin cá nhân) | Người dùng thay đổi các thông tin cá nhân |
| 6 | Search product (Tìm kiếm hàng điện tử) | Tìm kiếm hàng điện tử theo tên |
| 7 | View list product (Xem Danh sách hàng điện tử) | Người dùng xem Danh sách điện tử các hàng điện tử có trên hệ thống |
| 8 | Filter product (Lọc sản phẩm) | Người dùng mong muốn xem Danh sách điện tử sản phẩm theo các tiêu chí về giá cả, thương hiệu, thể loại |
| 9 | Sort product | Người dùng mong muốn xem Danh sách điện tử sản phẩm được sắp xếp từ cao đến thấp hoặc từ thấp về cao theo tiêu chí về độ mới sản phẩm, giá cả, tên. |
| 10 | Review product (Đánh giá cho hàng điện tử) | Người dùng bình luận và đánh giá sao cho hàng điện tử thể hiện sự hài lòng |
| 11 | Recommend product (Gợi ý hàng điện tử) | Người dùng được hệ thống gợi ý các hàng điện tử tương tự |
| 12 | Add to Cart (Thêm vào giỏ hàng) | Người dùng thực hiện thêm sản phẩm vào giỏ hàng để sau đó có thể mua hoặc xem lại sản phẩm |
| 13 | Order  (Đơn hàng) | Người dùng thực hiện mua hàng gồm xem Danh sách điện tử đơn hàng, tổng giá, chọn địa chỉ vận chuyển, chọn hình thức thanh toán, chọn ngân hàng thanh toán… |
| 14 | View history transaction (Xem lịch sử giao dịch) | Người dùng có thể xem lại các đơn hàng đã đặt, trạng thái đơn hàng,… |
| 15 | View favorite product (Xem Danh sách hàng điện tử yêu thích) | Người dùng xem những hàng điện tử nằm trong Danh sách điện tử yêu thích |
| 16 | Compare product  (So sánh sản phẩm) | Người dùng có thể chọn các sản phẩm để so sánh về các thông số kĩ thuật, giá, từ đó đưa ra lựa chọn hợp lí |

* + 1. **Danh sách các usecase cho người quản trị**

Bảng 3.4 Danh sách điện tử usecase người quản trị

|  |  |  |
| --- | --- | --- |
| 1 | Manage product (Quản lý hàng điện tử) | Quản trị xem Danh sách hàng điện tử, thêm một hàng điện tử mới, sửa hàng điện tử và xóa trên hệ thống  Thêm màu sắc, thêm kích thước, số lương hàng cho từng loại sản phẩm (product variant) |
| 2 | Manage brand (Quản lý tập thương hiệu) | Người quản trị có thể xem, thêm, sửa, xóa thương hiệu |
| 3 | Manage account (Quản lý tài khoản) | Người quản trị có thể xem Danh sách điện tử các tài khoản trên hệ thống |
| 4 | Manage category (Quản lý thể loại) | Người quản trị có thể thêm, sửa, xóa thể loại trên hệ thống |
| 5 | Manage collection (Quản lý thể loại bộ sưu tập sản phẩm) | Người quản trị có thể thêm, sửa, xóa bộ sưu tập sản phẩm trên hệ thống |
| 6 | Manage Advertisement | Người quản trị có thể thêm, sửa, xóa quảng cáo sản phẩm trên hệ thống |
| 7 | Manage order (Quản lý đơn hàng) | Người quản trị có thể xem tất cả đơn hàng, chỉnh sửa trạng thái đơn hàng |
| 8 | View Statistic (xem thống kế) | Người quản trị có thể xem doanh thu theo một khoảng thời gian, xem biểu đồ doanh thu theo từng tháng của của cửa hàng |

* 1. **Biểu đồ usecase**
     1. **Biểu đồ usecase tổng quát hệ thống**



Hình 3.1 Usecase tổng quát hệ thống

* + 1. **Phân rã một số usecase chi tiết**
* Phân rã usecase Đăng kí:

A blue oval with black text

Description automatically generated

Hình 3.2 Phân rã usecase Đăng kí

* Phân rã usecase Đăng nhập:

A blue and black line with a blue oval

Description automatically generated

Hình 3.3 Phân rã usecase Đăng nhập

* Phân rã usecase Quên mật khẩu:

A blue oval with black text

Description automatically generated

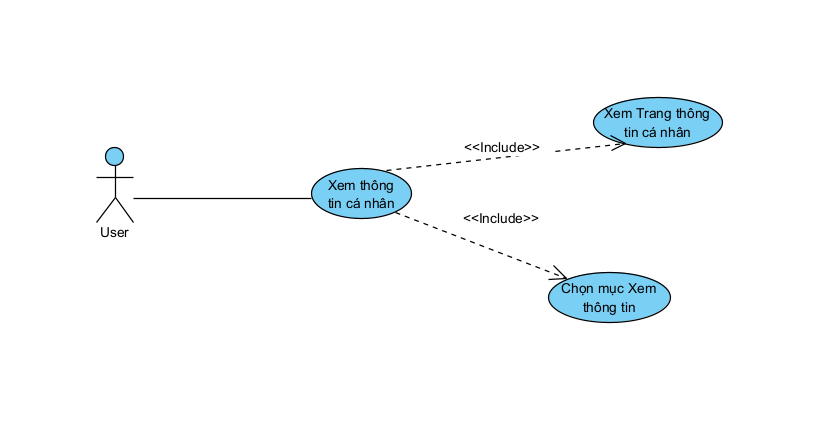
Hình 3.4 Phân rã usecase Quên mật khẩu

* Phân rã usecase Thay đổi mật khẩu:



Hình 3.4 Phân rã usecase Thay đổi mật khẩu

* Phân rã usecase Xem thông tin cá nhân:



Hình 3.5 Phân rã usecase Xem thông tin cá nhân

* Phân rã usecase Thay đổi thông tin cá nhân:

A diagram of a diagram

Description automatically generated

Hình 3.5 Phân rã usecase Thay đổi thông tin cá nhân

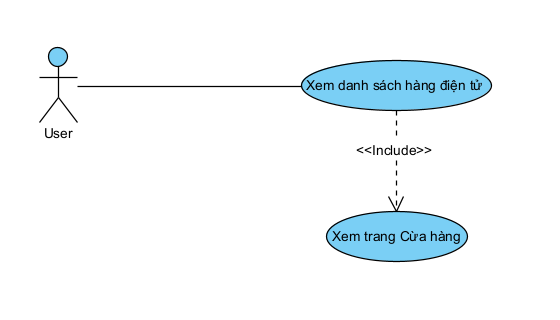
* Phân rã usecase Tìm kiếm hàng điện tử:

A diagram of a diagram

Description automatically generated

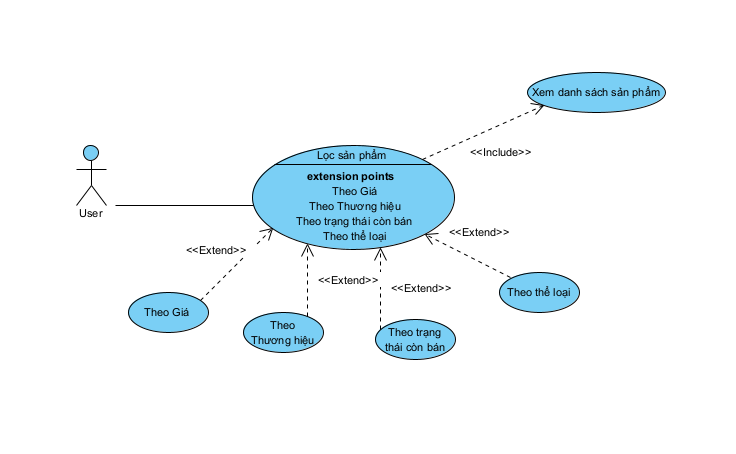
Hình 3.6 Phân rã usecase Tìm kiếm hàng điện tử

* Phân rã usecase Xem Danh sách hàng điện tử:



Hình 3.7 Phân rã usecase Xem Danh sách hàng điện tử

* Phân rã usecase Lọc sản phẩm



Hình 3.8 Phân rã usecase Lọc sản phẩm

* Phân rã usecase Sắp xếp sản phẩm

A diagram of a diagram

Description automatically generated

Hình 3.9 Phân rã usecase Sắp xếp sản phẩm

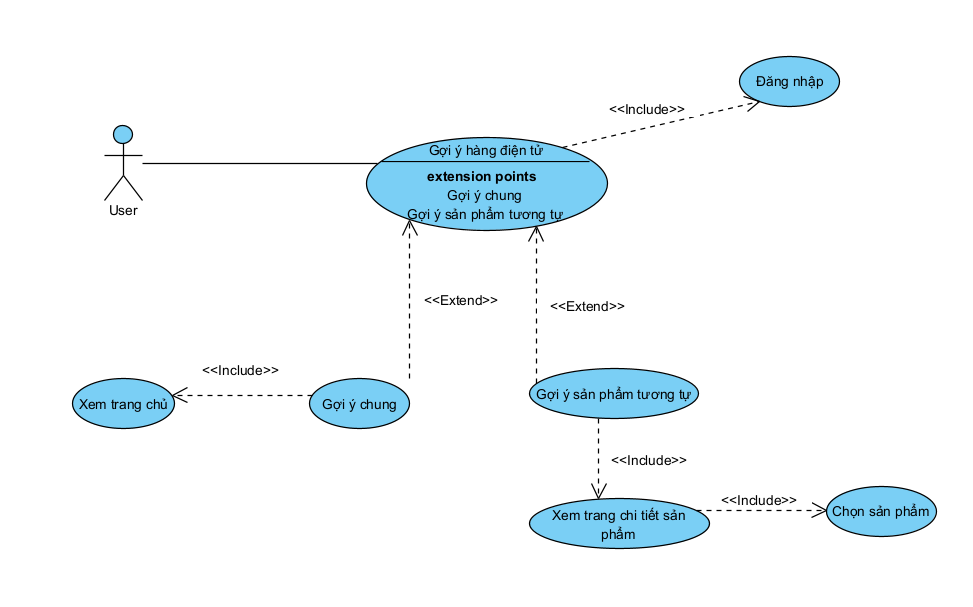
* Phân rã usecase Đánh giá cho hàng điện tử

A diagram of a diagram

Description automatically generated with medium confidence

Hình 3.10 Phân rã usecase Đánh giá cho hàng điện tử

* Phân rã usecase Gợi ý hàng điện tử



Hình 3.11 Phân rã usecase Gợi ý hàng điện tử

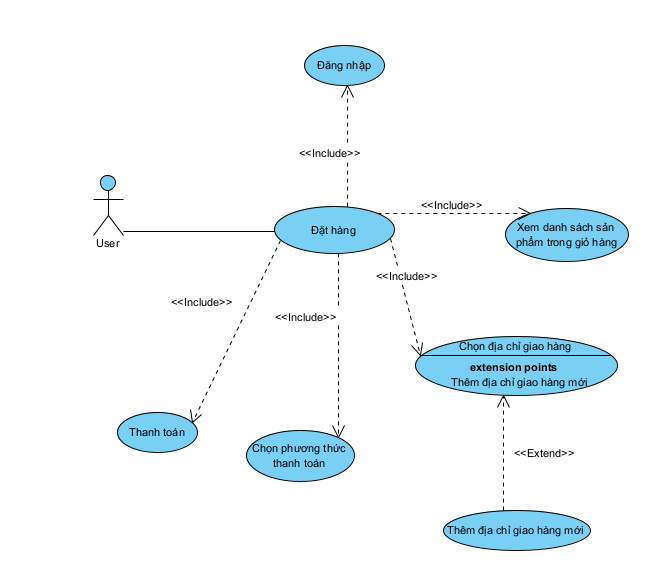
* Phân rã usecase Thêm vào giỏ hàng

A diagram of a company

Description automatically generated

Hình 3.12 Phân rã usecase Thêm vào giỏ hàng

* Phân rã usecase Đặt hàng



Hình 3.13 Phân rã usecasae Đặt hàng

* Phân rã usecase Xem lịch sử giao dịch

A diagram of a diagram

Description automatically generated

Hình 3.14 Phân rã usecase Xem lịch sử giao dịch

* Phân rã usecase Xem Danh sách hàng điện tử yêu thích

A diagram of a diagram

Description automatically generated

Hình 3.15 Phân rã usecase Xem Danh sách hàng điện tử yêu thích

* Phân rã usecase So sánh sản phẩm

A diagram of a diagram

Description automatically generated

Hình 3.16 Phân rã usecase So sánh sản phẩm

* Phân rã usecase Quản lý hàng điện tử

A diagram of a diagram

Description automatically generated

Hình 3.17 Phân rã usecase Quản lý hàng điện tử

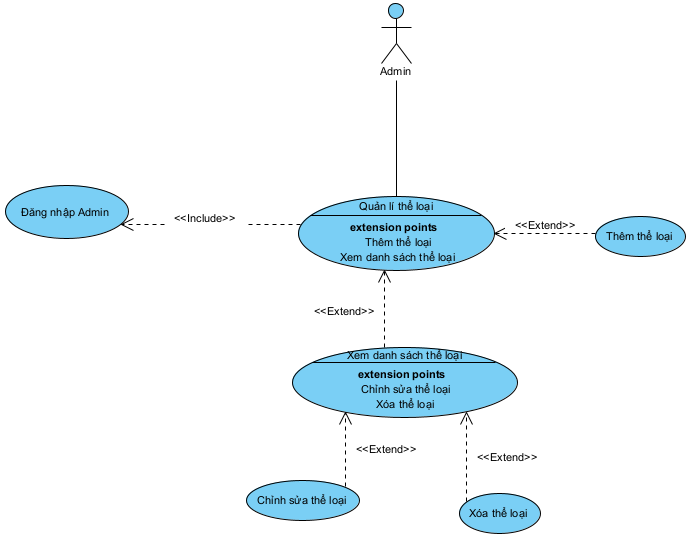
* Phân rã usecase Quản lý thương hiệu

A diagram of a diagram

Description automatically generated

Hình 3.18 Phân rã usecase Quản lý thương hiệu

* Phân rã usecase Quản lý thể loại



Hình 3.19 Phân rã usecase Quản lý thể loại

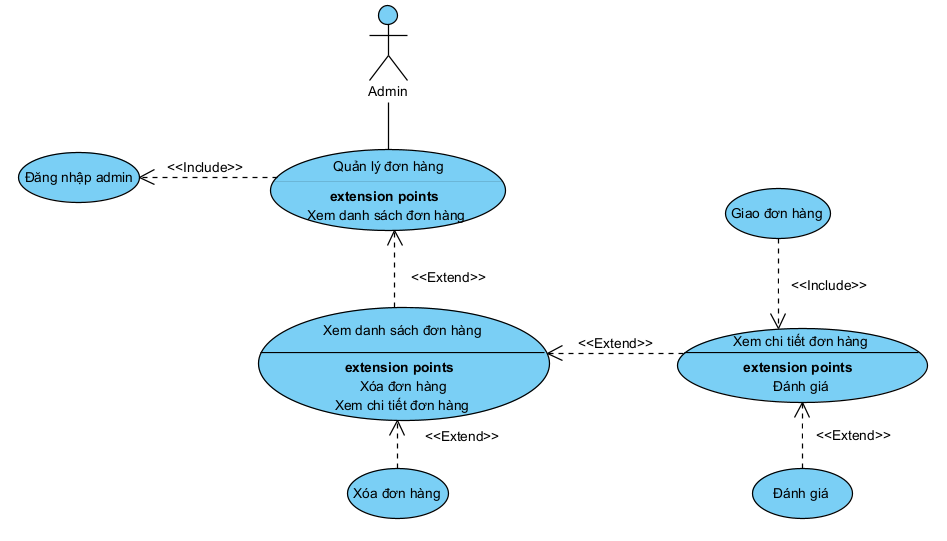
* Phân rã usecase Quản lý bộ sưu tập

A diagram of a diagram

Description automatically generated

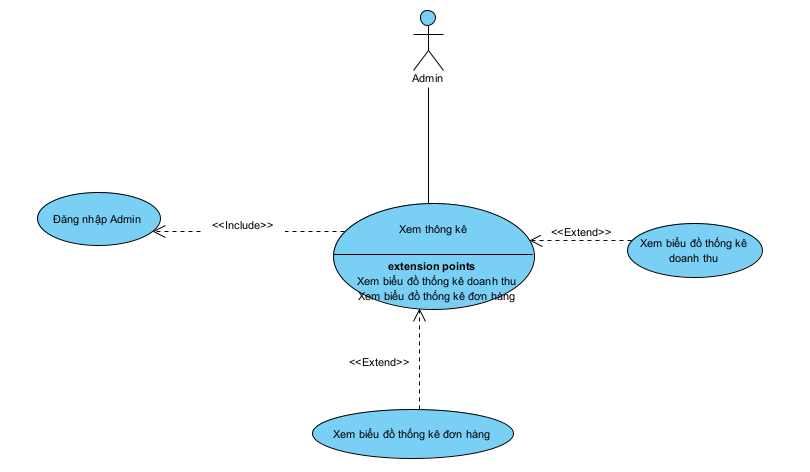
Hình 3.20 Phân rã usecase Quản lý bộ sưu tập

* Phân rã usecase Quản lý đơn hàng



Hình 3.21 Phân rã usecase Quản lý đơn hàng

* Phân rã usecase Xem thống kê



Hình 3.22 Phân rã usecase Xem thống kê

* 1. **Xây dựng kịch bản**
     1. **Các kịch bản ứng với các chức năng của người dùng**
* **Chức năng đăng ký**

Bảng 3.5 Kịch bản cho chức năng đăng ký

|  |  |
| --- | --- |
| Tên UseCase | Đăng ký |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng truy cập vào hệ thống |
| Hậu điều kiện | Đăng ký tài khoản thành công |
| **Chuỗi sự kiện chính:**   1. Người dùng chọn mục đăng nhập trong giao diện hệ thống. 2. Giao diện đăng nhập hiện ra, người dùng click vào nút “Đăng ký” 3. Sau khi click vào button đăng ký, giao diện đăng ký hiện ra với các trường  * Họ và tên * Email * Số điện thoại * Tên đăng nhập * Mật khẩu * Nhập lại mật khẩu  1. Người dùng nhập đầy đủ thông tin các trường 2. Sau khi nhập đầy đủ, người dùng click đăng ký 3. Hệ thống sẽ chuyển sang giao diện đăng nhập là đăng ký thành công | |
| **Ngoại lệ:**  5**.** Sau khi nhập đầy đủ, người dùng click đăng ký  5.1 Email không đúng định dạng  5.2 Tên đăng nhập đã tồn tại  5.3 Mật khẩu nhập lại không khớp | |

* **Chức năng đăng nhập**

Bảng 3.6 Kịch bản cho chức năng đăng nhập

|  |  |
| --- | --- |
| Tên UseCase | Đăng nhập |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đã có tài khoản |
| Hậu điều kiện | Đăng nhập thành công |
| **Chuỗi sự kiện chính:**  1. Người dùng chọn mục đăng nhập trong giao diện hệ thống.   1. Giao diện đăng nhập hiện ra với trường :  * Tên đăng nhập * Mật khẩu  1. Người dùng nhập thông tin 2. Sau khi nhập đầy đủ thông tin, người dùng click vào button đăng nhập 3. Hệ thống chuyển hướng sang trang chủ với thông tin cá nhân ở góc màn hình là đăng nhập thành công | |
| **Ngoại lệ:**  4**.** Sau khi nhập đầy đủ thông tin, người dùng click vào button đăng nhập  4.1. Thông tin tài khoản, mật khẩu chưa chính xác  4.2. Nhập thiếu 1 trong 2 trường thông tin | |

* **Chức năng quên mật khẩu**

Bảng 3.7 Kịch bản cho chức năng quên mật khẩu

|  |  |
| --- | --- |
| Tên UseCase | Quên mật khẩu |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đã có tài khoản và đang ở trang đăng nhập |
| Hậu điều kiện | Hệ thống thông báo đổi mật khẩu thành công và chuyển hướng sang trang đăng nhập |
| **Chuỗi sự kiện chính:**   1. Người dùng chọn quên mật khẩu. 2. Hệ thống chuyển hướng sang trang khôi phục mật khẩu. 3. Người dùng nhập địa chỉ email và chọn Gửi. 4. Hệ thống thông báo kiểm tra email và chuyển sang trang xác nhận OTP. 5. Người dùng kiểm tra email và nhận mã OTP để nhập mật khẩu mới. 6. Người dùng nhập mã OTP vào trang xác nhận 7. Hệ thống chuyển sang trang Đổi lại mật khẩu 8. Người dùng nhập mật khẩu mới 2 lần và chọn Xác nhận. 9. Hệ thống thông báo đổi mật khẩu thành công và chuyển hướng sang trang đăng nhập. | |
| **Ngoại lệ:**  3**.** Người dùng nhập địa chỉ email và chọn Gửi  3.1. Email người dùng nhập không chính xác, email khác với email người dùng đã đăng kí  6. Người dùng nhập mã OTP vào trang xác nhận  6.1. Người dùng nhập sai mã OTP | |

* **Chức năng đổi mật khẩu**

Bảng 3.8 Kịch bản cho chức năng đổi mật khẩu

|  |  |
| --- | --- |
| Tên UseCase | Đổi mật khẩu |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đã có tài khoản và đăng nhập thành công |
| Hậu điều kiện | Đổi mật khẩu thành công |
| **Chuỗi sự kiện chính:**   1. Sau khi đăng nhập, người dùng click vào tên mình trên góc phải màn hình 2. Hệ thống hiển thị thông tin cá người dùng. 3. Người dùng chọn mục “Đổi mật khẩu” ở phần sidebar của trang thông tin cá nhân 4. Hệ thống hiển thị ra form đổi mật khẩu của người dùng với các trường thông tin:  * Mật khẩu cũ * Mật khẩu mới * Nhập lại mật khẩu mới  1. Người dùng thực hiện nhập các trường thông tin 2. Sau đó click button lưu thông tin 3. Hệ thống thông báo đổi mật khẩu thành công | |
| **Ngoại lệ:**  6**.** Người dùng click button lưu thông tin  6.1. Mật khẩu cũ chưa chính xác  6.2. Mật khẩu nhập lại không trùng khớp | |

* **Chức năng đánh giá**

Bảng 3.9 Kịch bản cho chức năng đánh giá

|  |  |
| --- | --- |
| Tên UseCase | Đánh giá |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công |
| Hậu điều kiện | Đánh giá thành công |
| **Chuỗi sự kiện chính:**   1. Sau khi đăng nhập, người dùng click vào hàng điện tử muốn thực hiện đánh giá 2. Sau khi chọn, thông tin chi tiết của hàng điện tử được hiển thị. 3. Người dùng click vào button Xem bình luận ở phần mô tả sản phẩm. 4. Hệ thống hiển thị những bình luận đã có của hàng điện tử. 5. Người dùng click vào nút viết bình luận 6. Hệ thống hiện ra form gồm số sao và input để nhập nội dùng 7. Người dùng nhập nội dung bình luận và click chọn số sao muốn đánh giá. 8. Click vào button bình luận 9. Hệ thống hiển thị nội dung người dùng vừa bình luận với thời gian bình luận | |
| **Ngoại lệ:**  8. Click vào button bình luận  6.1 Nội dung bình luận trống  6.2. Chưa chọn số sao đánh giá  6.3 Hệ thống không hiển thị nội dung vừa bình luận | |

* **Chức năng tìm kiếm sản phẩm theo tên**

Bảng 3.10 Kịch bản cho chức năng tìm kiếm sản phẩm theo tên

|  |  |
| --- | --- |
| Tên UseCase | Tìm kiếm sản phẩm |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng truy cập vào hệ thống (không cần đăng nhập) |
| Hậu điều kiện | Hệ thống hiển thị kết quả tìm kiếm tương ứng với tên sản phẩm mà người dùng đã nhập |
| **Chuỗi sự kiện chính:**   1. Người dùng nhập tên sản phẩm muốn tìm vào ô text input tìm kiếm trên phần topbar của hệ thống. 2. Hệ thống sẽ hiển thị Danh sách điện tử sản phẩm ứng với tên người dùng vừa gõ 3. Người dùng click chọn sản phẩm muốn tìm trong danh sách hàng điện tử 4. Hệ thống chuyển đến trang chi tiết sản phẩm mà người dùng vừa gõ | |
| **Ngoại lệ:**  2. Hệ thống sẽ hiển thị Danh sách điện tử sản phẩm ứng với tên người dùng vừa gõ:  2.1. Không có sản phẩm nào trùng với input người dùng nhập vào hoặc Hệ thống chưa có sản phẩm nào. | |

* **Chức năng xem thông tin cá nhân**

Bảng 3.11 Kịch bản cho chức *năng* xem thông tin cá nhân

|  |  |
| --- | --- |
| Tên UseCase | Xem thông tin cá nhân |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công |
| Hậu điều kiện | Người dùng xem được các thông tin của cá nhân |
| **Chuỗi sự kiện chính:**   1. Sau khi đăng nhập, người dùng click vào tên mình trên góc phải màn hình 2. Hệ thống hiển thị thông tin cá người dùng gồm các thông tin:  * Họ và tên * Username * Email * Số điện thoại * Địa chỉ * Ngày sinh * Ngày tham gia hệ thống * Ảnh đại diện người dùng | |
| **Ngoại lệ:** | |

* **Chức năng thay đổi thông tin cá nhân**

Bảng 3.12 Kịch bản cho chức năng thay đổi thông tin cá nhân

|  |  |
| --- | --- |
| Tên UseCase | Sửa thông tin cá nhân |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công |
| Hậu điều kiện | Cập nhật thông tin thành công |
| **Chuỗi sự kiện chính:**   1. Sau khi đăng nhập, người dùng click vào tên mình trên góc phải màn hình 2. Hệ thống hiển thị thông tin cá người dùng gồm các thông tin:  * Họ và tên * Username * Email * Số điện thoại * Địa chỉ * Ngày sinh * Ngày tham gia hệ thống * Ảnh đại diện người dùng  1. Người dùng click vào button sửa thông tin cá nhân 2. Hệ thống hiện thị popup chứa thông tin người dùng 3. Người dùng sửa lần lượt các thông tin trên 4. Click button lưu thông tin 5. Hệ thống cập nhật thông tin thành công | |
| **Ngoại lệ:**  6.Click button lưu thông tin  6.1. Người dùng để trống thông tin  6.2. Người dùng giữ nguyên không sửa thông tin gì | |

* **Chức năng gợi ý hàng điện tử**

Bảng 3.13 Kịch bản cho chức năng gợi ý hàng điện tử

|  |  |
| --- | --- |
| Tên UseCase | Gợi ý hàng điện tử |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công |
| Hậu điều kiện | Người dùng xem được các sản phẩm gợi ý |
| **Chuỗi sự kiện chính:**  **Kịch bản 1:**   1. Sau khi đăng nhập, người dùng sẽ thấy các sản phẩm gợi ý trong phần Sản phẩm gợi ý của trang chủ   **Kịch bản 2:**  1.Sau khi đăng nhập, người dùng chọn xem thông tin 1 san phẩm   1. Hệ thống hiển thị thông tin sản phẩm đó 2. Người dùng scroll xuống dưới phần bình luận 3. Danh sách sản phẩm cùng loại sẽ được hiện thị để gợi ý cho người dùng | |
| **Ngoại lệ:** | |

* **Chức năng xem Thêm vào giỏ hàng**

Bảng 3.14 Kịch bản cho chức năng thêm vào giỏ hàng

|  |  |
| --- | --- |
| Tên Use Case | Thêm vào giỏ hàng |
| Tác nhân | Người dùng |
| Tiền điều kiện | Người dùng đã có tài khoản trên hệ thống và đã truy cập vào hệ thống. |
| Hậu điều kiện | Hệ thống cập nhật giỏ hàng của người dùng |
| **Chuỗi sự kiện chính:**  1. Chọn nút “Thêm vào giỏ hàng”   * TH1: Người dùng đang ở trang Home hoặc OurStore * TH2: Người dùng đang ở trang chi tiết sản phẩm   2. Hệ thống thêm sản phẩm vào giỏ hàng và cập nhật số lượng sản phẩm trong giỏ hàng trên màn hình. | |
| **Ngoại lệ:**   1. Chọn nút “Thêm vào giỏ hàng”    1. Sản phẩm đã bị hết hàng/ sản phẩm bị quá số lượng trong kho không thể cộng thêm sản phẩm. | |

* **Chức năng Đặt hàng và thanh toán**

Bảng 3.15 Kịch bản cho chức năng đặt hàng và thanh toán

|  |  |
| --- | --- |
| Tên Use Case | Đặt hàng và thanh toán |
| Tác nhân | Người dùng |
| Tiền điều kiện | Người dùng đang ở trang giỏ hàng |
| Hậu điều kiện | Hệ thống hiển thị thông báo Đơn hàng đã được thanh toán |
| **Chuỗi sự kiện chính:**  1. Người dùng tích chọn các sản phẩm muốn mua ở trang giỏ hàng chọn nút “Thanh toán”  2. Hệ thống chuyển đến trang “Thanh toán đơn hàng”  3. Người dùng nhập các thông tin liên quan đến địa chỉ giao hàng, phương thức thanh toán và click “Thanh toán bằng VNPay”  4. Hệ thống hiện thị trang thanh toán của VNPay  5. Người dùng nhập thông tin tài khoản thanh toán  6. Hệ thống hiện thị trang thanh toán thành công | |
| **Ngoại lệ:**  5. Người dùng nhập thông tin tài khoản thanh toán  5.1. Người dùng nhập thiếu/ sai định dạng các trường thông tin Tài khoản thanh toán  5.2. Người dùng không đủ số dư thanh toán | |

* **Chức năng xem lịch sử giao dịch**

Bảng 3.16 Kịch bản cho chức năng xem lịch sử giao dịch

|  |  |
| --- | --- |
| Tên UseCase | Xem lịch sử giao dịch |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công |
| Hậu điều kiện | Người dùng vào trang lịch sử giao dịch |
| **Chuỗi sự kiện chính:**   1. Sau khi đăng nhập, người dùng click vào tên mình trên góc phải màn hình 2. Hệ thống hiển thị thông tin cá người dùng. 3. Người dùng chọn mục lịch sử giao dịch ở phần sidebar của trang thông tin cá nhân 4. Hệ thống hiển thị ra lịch sử những lần giao dịch của người dùng với các trường thông tin:  * Mã đơn hàng * Tên sản phẩm * Hình ảnh * Số lượng * Giá | |
| **Ngoại lệ:** | |

* **Chức năng xem những hàng điện tử đã yêu thích**

Bảng 3.17 Kịch bản cho chức năng xem những hàng điện tử đã thích

|  |  |
| --- | --- |
| Tên UseCase | Xem những hàng điện tử đã yêu thích |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng đăng nhập thành công |
| Hậu điều kiện | Người dùng xem được các sản phẩm trong trang sản phẩm yêu thích |
| **Chuỗi sự kiện chính:**   1. Sau khi đăng nhập, người dùng click vào phần Danh sách sản phẩm yêu thích góc phải màn hình 2. Hệ thống hiển thị ra danh sách những hàng điện tử yêu thích của người dùng với các trường thông tin:  * Tiêu đề * Số sao của hàng điện tử * Hình ảnh * Giá cả | |
| **Ngoại lệ:**  1. Hệ thống hiển thị ra danh sách điện tử những hàng điện tử yêu thích của người dùng   * 1. Người dùng chưa yêu thích hàng điện tử nào.   2. Hệ thống hiển thị thông báo không tìm thấy kết quả. | |

* **Chức năng xem danh sách sản phẩm**

Bảng 3.18 Kịch bản cho chức năng xem danh sách sản phẩm

|  |  |
| --- | --- |
| Tên UseCase | Xem danh sách sản phẩm |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng truy cập vào hệ thống bán hàng |
| Hậu điều kiện | Người dùng xem được danh sách sản phẩm |
| **Chuỗi sự kiện chính:**   1. Sau khi truy cập vào hệ thống, người dùng click vào Cửa hàng 2. Hệ thống hiển thị ra danh sách tất cả sản phẩm có trên hệ thống, mỗi sản phẩm gồm:  * Tiêu đề * Số sao của hàng điện tử * Hình ảnh * Giá cả * Khuyến mại | |
| **Ngoại lệ:** | |

* **Chức năng lọc sản phẩm**

Bảng 3.19 Kịch bản cho chức năng lọc sản phẩm

|  |  |
| --- | --- |
| Tên UseCase | Lọc sản phẩm |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng truy cập vào hệ thống bán hàng |
| Hậu điều kiện | Người dùng xem được danh sách sản phẩm sau khi đã lọc |
| **Chuỗi sự kiện chính:**   1. Sau khi truy cập vào hệ thống, người dùng click vào Cửa hàng   2. Hệ thống hiển thị ra danh sách tất cả sản phẩm có trên hệ thống, mỗi sản phẩm gồm:   * Tiêu đề * Số sao của hàng điện tử * Hình ảnh * Giá cả * Khuyến mại   3. Người dụng chọn lọc sản phẩm theo các tiêu chí:   * Loại * Thương hiệu * Tình trạng(còn hàng hay hết hàng) * Giá   4. Hệ thống hiện thị các sản phẩm đáp ứng tiêu chí vừa lọc | |
| **Ngoại lệ:**  3. Người dụng chọn lọc sản phẩm theo các tiêu chí:  3.1. Không có sản phẩm nào đáp ứng tiêu chí người dùng vừa lọc | |

* **Chức năng sắp xếp sản phẩm**

Bảng 3.20 Kịch bản cho chức năng sắp xếp sản phẩm

|  |  |
| --- | --- |
| Tên UseCase | Sắp xếp sản phẩm |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng truy cập vào hệ thống bán hàng |
| Hậu điều kiện | Người dùng xem được danh sách sản phẩm sau khi đã sắp xếp |
| **Chuỗi sự kiện chính:**   1. Sau khi truy cập vào hệ thống, người dùng click vào Cửa hàng   2. Hệ thống hiển thị ra danh sách tất cả sản phẩm có trên hệ thống, mỗi sản phẩm gồm:   * Tiêu đề * Số sao của hàng điện tử * Hình ảnh * Giá cả * Khuyến mại   3. Người dụng chọn sắp xếp sản phẩm theo các tiêu chí:   * Tiêu đề * Độ mới * Giá   4. Hệ thống hiện thị các sản phẩm đáp ứng tiêu chí vừa sắp xếp | |
| **Ngoại lệ:** | |

* **Chức năng so sánh sản phẩm**

Bảng 3.21 Kịch bản cho chức năng so sánh sản phẩm

|  |  |
| --- | --- |
| Tên UseCase | So sánh sản phẩm |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Người dùng truy cập vào hệ thống bán hàng |
| Hậu điều kiện | Người dùng xem được danh sách sản phẩm đã chọn để so sánh |
| **Chuỗi sự kiện chính:**   1. Sau khi truy cập vào hệ thống, người dùng click vào So sánh sản phẩm   2. Hệ thống hiển thị ra danh sách tất cả sản phẩm trong danh sách so sánh trên hệ thống, mỗi sản phẩm gồm:   * Tiêu đề * Số sao của hàng điện tử * Hình ảnh * Giá cả * Khuyến mại | |
| **Ngoại lệ:**  2. Hệ thống hiển thị ra danh sách tất cả sản phẩm trong danh sách so sánh trên hệ thống  2.1. Nếu chưa có sản phẩm nó trong danh sách  2.2. Người dùng quay lại danh sách sản phẩm và chọn 1 sản phẩm phẩm muốn so sánh  2.3. Người dùng click vào button “Thêm vào danh sách so sánh” | |

* + 1. **Các kịch bản ứng với các chức năng của quản trị hệ thống**
* **Chức năng quản lý hàng điện tử**

Bảng 3.22 Kịch bản cho chức năng xem danh sách hàng điện tử

|  |  |
| --- | --- |
| Tên UseCase | Quản lí hàng điện tử |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Xem danh sách hàng điện tử |
| **Chuỗi sự kiện chính:**   1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý hàng điện tử” trong phần menu hệ thống. 2. Danh sách hàng điện tử của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm hàng điện tử  * Tiêu đề * Số sao * Ngày tạo * Người tạo * Hình anh * Màu sắc * Kích thước | |
| **Ngoại lệ:** | |

* **Chức năng thêm hàng điện tử mới**

Bảng 3.23 Kịch bản cho chức năng thêm hàng điện tử mới

|  |  |
| --- | --- |
| Tên UseCase | Quản lí hàng điện tử |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Thêm thành công hàng điện tử |
| **Chuỗi sự kiện chính:**  1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý hàng điện tử” trong phần menu hệ thống.  2. Danh sách hàng điện tử của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm hàng điện tử gồm: Tiêu đề, Trạng thái, Số tập, Hình thức, Ngày phát hành, Số sao, Lượt xem, Ngày tạo, Người tạo, Tùy chọn   1. Sau đó click vào button thêm hàng điện tử 2. Sau khi click xong, giao diện diện lên với các trường nhập thông tin và button lưu, button hủy :  * Tiêu đề * Mô tả * Hình ảnh * Màu sắc * Kích thước * Bộ sưu tập * Thương hiệu * Thể loại hàng điện tử  1. Sau khi nhập đầy đủ thông tin, quản trị viên click button lưu 2. Hệ thống lưu thành công và chuyển hướng về trang danh sách hàng điện tử | |
| **Ngoại lệ:**  5.Sau khi nhập đầy đủ thông tin, quản trị viên click button lưu  5.1 Nhập thiếu trường thông tin | |

* **Chức năng sửa hàng điện tử**

Bảng 3.24 Kịch bản cho chức năng sửa hàng điện tử mới

|  |  |
| --- | --- |
| Tên UseCase | Quản lí hàng điện tử |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Sửa hàng điện tử thành công |
| **Chuỗi sự kiện chính:**  1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý hàng điện tử” trong phần menu hệ thống.  2. Danh sách hàng điện tử của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm hàng điện tử.  3. Sau đó click vào nút sửa trong mục tùy chọn của từng hàng điện tử   1. Sau khi click xong, giao diện sửa hiện lên với các trường thông tin của hàng điện tử có sẵn trong hệ thống và button lưu, button hủy :  * Tiêu đề * Mô tả * Hình ảnh * Màu sắc * Kích thước * Bộ sưu tập * Thương hiệu * Thể loại hàng điện tử  1. Sau sửa đầy đủ thông tin, quản trị viên click button lưu. 2. Hệ thống lưu thành công và chuyển hướng về trang Danh sách hàng điện tử | |
| **Ngoại lệ:**  5. Sau khi click xong, giao diện sửa hiện lên với các trường thông tin của hàng điện tử có sẵn trong hệ thống và button lưu, button hủy  5.1. Quản trị để trống các trường thông tin   * 1. Lỗi hệ thống | |

* **Chức năng xóa hàng điện tử**

Bảng 3.25 Kịch bản cho chức năng xóa hàng điện tử

|  |  |
| --- | --- |
| Tên UseCase | Quản lí hàng điện tử |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Thông tin hàng điện tử không hiển thị trong Danh sách hàng điện tử |
| **Chuỗi sự kiện chính:**  1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý hàng điện tử” trong phần menu hệ thống.  2. Danh sách hàng điện tử của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm hàng điện tử   * Tiêu đề * Số sao * Ngày tạo * Người tạo * Hình anh * Màu sắc * Kích thước   3. Sau đó click vào nút xóa trong mục tùy chọn của từng hàng điện tử   1. Hệ thống xóa thành công và chuyển hướng về trang Danh sách hàng điện tử | |
| **Ngoại lệ:** | |

* **Chức năng quản lý Thể loại**

Bảng 3.26 Kịch bản cho chức năng xem Danh sách điện tử Thể loại

|  |  |
| --- | --- |
| Tên UseCase | Quản lí Thể loại |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Xem danh Thể loại |
| **Chuỗi sự kiện chính:**   1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý Thể loại” trong phần menu hệ thống. 2. Danh sách điện tử Thể loại của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm tập Thể loại  * Tiêu đề * Hình ảnh * Ngày thêm | |
| **Ngoại lệ:** | |

* **Chức năng thêm Thể loại mới**

Bảng 3.27 Kịch bản cho chức năng thêm Thể loại mới

|  |  |
| --- | --- |
| Tên UseCase | Quản lí thể loại |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Thêm Thể loại mới thành công |
| **Chuỗi sự kiện chính:**  1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý Thể loại” trong phần menu hệ thống.  2. Danh sách điện tử Thể loại của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm tập Thể loại gồm: Tiêu đề, Hình ảnh, Ngày thêm   1. Sau đó click vào button thêm Thể loại 2. Sau khi click xong, giao diện diện lên với các trường nhập thông tin và button lưu, button hủy :  * Tiêu đề * Hình ảnh * Ngày thêm  1. Sau khi nhập đầy đủ thông tin, quản trị viên click button lưu 2. Hệ thống lưu thành công và chuyển hướng về trang Danh sách điện tử Thể loại | |
| **Ngoại lệ:**  5.Sau khi nhập đầy đủ thông tin, quản trị viên click button lưu  5.1 Nhập thiếu trường thông tin | |

* **Chức năng sửa Thể loại**

Bảng 3.28 Kịch bản cho chức năng sửa Thể loại

|  |  |
| --- | --- |
| Tên UseCase | Quản lí Thể loại |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Sửa Thể loại thành công |
| **Chuỗi sự kiện chính:**  1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý Thể loại” trong phần menu hệ thống.  2. Danh sách điện tử Thể loại của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm Thể loại.  3. Sau đó click vào nút sửa trong mục tùy chọn của từng Thể loại  4. Sau khi click xong, giao diện sửa hiện lên với các trường thông tin của Thể loại có sẵn trong hệ thống và button lưu, button hủy :   * Tiêu đề * Hình ảnh  1. Sau sửa đầy đủ thông tin, quản trị viên click button lưu. 2. Hệ thống lưu thành công và chuyển hướng về trang Danh sách điện tử Thể loại | |
| **Ngoại lệ:**  4. Sau khi click xong, giao diện sửa hiện lên với các trường thông tin của Thể loại có sẵn trong hệ thống và button lưu, button hủy  4.1. Quản trị để trống các trường thông tin   * 1. Lỗi hệ thống | |

* **Chức năng xóa Thể loại**

Bảng 3.29 Kịch bản cho chức năng xóa Thể loại

|  |  |
| --- | --- |
| Tên UseCase | Quản lí Thể loại |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Thông tin Thể loại không hiển thị trong danh sách Thể loại |
| **Chuỗi sự kiện chính:**  1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý Thể loại” trong phần menu hệ thống.  2. Danh sách điện tử Thể loại của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm Thể loại   * Tiêu đề * Hình ảnh * Ngày thêm   3. Sau đó click vào nút xóa trong mục tùy chọn của từng Thể loại  4. Hệ thống xóa thành công và chuyển hướng về trang Danh sách điện tử Thể loại | |
| **Ngoại lệ:** | |

* **Chức năng quản lý Bộ sưu tập**

Bảng 3.30 Kịch bản cho chức năng xem danh sách Bộ sưu tập

|  |  |
| --- | --- |
| Tên UseCase | Quản lí Bộ sưu tập |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Xem danh sách Bộ sưu tập |
| **Chuỗi sự kiện chính:**   1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý Bộ sưu tập” trong phần menu hệ thống. 2. Danh sách điện tử Bộ sưu tập của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm tập Bộ sưu tập  * Tiêu đề * Hình ảnh * Ngày thêm | |
| **Ngoại lệ:** | |

* **Chức năng thêm Bộ sưu tập mới**

Bảng 3.31 Kịch bản cho chức năng thêm Bộ sưu tập mới

|  |  |
| --- | --- |
| Tên UseCase | Quản lí Bộ sưu tập |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Thêm Bộ sưu tập mới thành công |
| **Chuỗi sự kiện chính:**  1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý Bộ sưu tập” trong phần menu hệ thống.  2. Danh sách điện tử Bộ sưu tập của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm tập Bộ sưu tập gồm: Tiêu đề, Hình ảnh, Ngày thêm   1. Sau đó click vào button thêm Bộ sưu tập 2. Sau khi click xong, giao diện diện lên với các trường nhập thông tin và button lưu, button hủy :  * Tiêu đề * Hình ảnh * Ngày thêm  1. Sau khi nhập đầy đủ thông tin, quản trị viên click button lưu 2. Hệ thống lưu thành công và chuyển hướng về trang danh sách Bộ sưu tập | |
| **Ngoại lệ:**  5.Sau khi nhập đầy đủ thông tin, quản trị viên click button lưu  5.1 Nhập thiếu trường thông tin | |

* **Chức năng sửa Bộ sưu tập**

Bảng 3.32 Kịch bản cho chức năng sửa Bộ sưu tập mới

|  |  |
| --- | --- |
| Tên UseCase | Quản lí Bộ sưu tập |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Sửa Bộ sưu tập thành công |
| **Chuỗi sự kiện chính:**  1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý Bộ sưu tập” trong phần menu hệ thống.  2. Danh sách điện tử Bộ sưu tập của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm Bộ sưu tập.  3. Sau đó click vào nút sửa trong mục tùy chọn của từng Bộ sưu tập  4. Sau khi click xong, giao diện sửa hiện lên với các trường thông tin của Bộ sưu tập có sẵn trong hệ thống và button lưu, button hủy :   * Tiêu đề * Hình ảnh  1. Sau sửa đầy đủ thông tin, quản trị viên click button lưu. 2. Hệ thống lưu thành công và chuyển hướng về trang danh sách Bộ sưu tập | |
| **Ngoại lệ:**  4. Sau khi click xong, giao diện sửa hiện lên với các trường thông tin của Bộ sưu tập có sẵn trong hệ thống và button lưu, button hủy  4.1. Quản trị để trống các trường thông tin   * 1. Lỗi hệ thống | |

* **Chức năng xóa Bộ sưu tập**

Bảng 3.33 Kịch bản cho chức năng xóa Bộ sưu tập

|  |  |
| --- | --- |
| Tên UseCase | Quản lí Bộ sưu tập |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Thông tin Bộ sưu tập không hiển thị trong danh sách Bộ sưu tập |
| **Chuỗi sự kiện chính:**  1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý Bộ sưu tập” trong phần menu hệ thống.  2. Danh sách điện tử Bộ sưu tập của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm Bộ sưu tập   * Tiêu đề * Hình ảnh * Ngày thêm   3. Sau đó click vào nút xóa trong mục tùy chọn của từng Bộ sưu tập  4. Hệ thống xóa thành công và chuyển hướng về trang danh sách Bộ sưu tập | |
| **Ngoại lệ:** | |

* **Chức năng quản lý tài khoản**

Bảng 3.34 Kịch bản cho chức năng xem danh tài khoản tài khoản

|  |  |
| --- | --- |
| Tên UseCase | Quản lí tài khoản |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Xem danh sách tài khoản |
| **Chuỗi sự kiện chính:**   1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý tài khoản” trong phần menu hệ thống. 2. Danh sách tài khoản của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm tập tài khoản  * Họ tên * Username * Số điện thoại * Địa chỉ * Email | |
| **Ngoại lệ:** | |

* **Chức năng thêm tài khoản mới**

Bảng 3.35 Kịch bản cho chức năng thêm tài khoản mới

|  |  |
| --- | --- |
| Tên UseCase | Quản lí tài khoản |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Thêm thành công tài khoản |
| **Chuỗi sự kiện chính:**  1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý tài khoản” trong phần menu hệ thống.  2. Danh sách tài khoản của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm tập tài khoản gồm: Họ tên, Username, Số điện thoại, Địa chỉ, Email và Tùy chọn   1. Sau đó click vào button thêm tài khoản 2. Sau khi click xong, giao diện diện lên với các trường nhập thông tin và button lưu, button hủy :  * Tên đầu * Tên cuối * Usename * Email * Số điện thoại * Ngày sinh * Địa chỉ  1. Sau khi nhập đầy đủ thông tin, quản trị viên click button lưu 2. Hệ thống lưu thành công và chuyển hướng về trang danh sách tài khoản | |
| **Ngoại lệ:**  5.Sau khi nhập đầy đủ thông tin, quản trị viên click button lưu  5.1 Nhập thiếu trường thông tin | |

* **Chức năng sửa tài khoản**

Bảng 3.36 Kịch bản cho chức năng sửa tài khoản mới

|  |  |
| --- | --- |
| Tên UseCase | Quản lí tài khoản |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Sửa tài khoản thành công |
| **Chuỗi sự kiện chính:**  1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý tài khoản” trong phần menu hệ thống.  2. Danh tài khoản tài khoản của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm tài khoản.  3. Sau đó click vào nút sửa trong mục tùy chọn của từng tài khoản   1. Sau khi click xong, giao diện sửa hiện lên với các trường thông tin của tài khoản có sẵn trong hệ thống và button lưu, button hủy :  * Tên đầu * Tên cuối * Usename * Email * Số điện thoại * Ngày sinh * Địa chỉ  1. Sau sửa đầy đủ thông tin, quản trị viên click button lưu. 2. Hệ thống lưu thành công và chuyển hướng về trang danh sách tài khoản | |
| **Ngoại lệ:**  5. Sau khi click xong, giao diện sửa hiện lên với các trường thông tin của tài khoản có sẵn trong hệ thống và button lưu, button hủy  5.1. Quản trị để trống các trường thông tin   * 1. Lỗi hệ thống | |

* **Chức năng xóa tài khoản**

Bảng 3.37 Kịch bản cho chức năng xóa tài khoản

|  |  |
| --- | --- |
| Tên UseCase | Quản lí tài khoản |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện | Thông tin tài khoản không hiển thị trong danh tài khoản tài khoản |
| **Chuỗi sự kiện chính:**  1. Sau khi đăng nhập, quản trị viên chọn mục “Quản lý tài khoản” trong phần menu hệ thống.  2. Danh sách tài khoản của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau và button thêm tài khoản   * Tên đầu * Tên cuối * Usename * Email * Số điện thoại * Ngày sinh * Địa chỉ   3. Sau đó click vào nút xóa trong mục tùy chọn của từng tài khoản   1. Hệ thống xóa thành công và chuyển hướng về trang danh sách tài khoản | |
| **Ngoại lệ:** | |

* **Chức năng thống kê đơn hàng**

Bảng 3.38 Kịch bản cho chức năng thống kê đơn hàng

|  |  |
| --- | --- |
| Tên UseCase | Thống kê đơn hàng |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện |  |
| **Chuỗi sự kiện chính:**   1. Sau khi đăng nhập, quản trị viên chọn mục “Thống kê đơn hàng” trong phần menu hệ thống. 2. Danh tài khoản tài khoản của hệ thống sẽ hiện ra dưới dạng bảng với các trường sau, mặc định thời gian bắt đầu thống kê là trong ngày hiện tại  * Tiêu đề * Trạng thái * Mã đơn hàng  1. Người dùng chọn ngày bắt đầu, ngày kết thúc muốn thống kê đơn hàng 2. Người dùng click button thống kê 3. Sau khi click xong, giao diện diện lên với các trường nhập thông tin và button lưu, button hủy :  * Tiêu đề * Trạng thái * Mã đơn hàng | |
| **Ngoại lệ:**  5. Quản trị viên chọn ngày bắt đầu lớn hơn ngày kết thúc  5.1 Hệ thống hiển thị doanh thu là 0 hết. | |

* **Chức năng thống kê doanh thu**

Bảng 3.39 Kịch bản cho chức năng thống kê doanh thu

|  |  |
| --- | --- |
| Tên UseCase | Thống kê doanh thu |
| Tác nhân chính | Quản trị hệ thống |
| Tiền điều kiện | Login thành công với quyền quản trị |
| Hậu điều kiện |  |
| **Chuỗi sự kiện chính:**   1. Sau khi đăng nhập, quản trị viên chọn mục “Thống kê doanh thu” trong phần menu hệ thống. 2. Danh tài khoản các gói dịch vụ của hệ thống sẽ hiện ra dưới dạng biểu đồ với các thông tin vê doanh thu theo tháng | |
| **Ngoại lệ:** | |

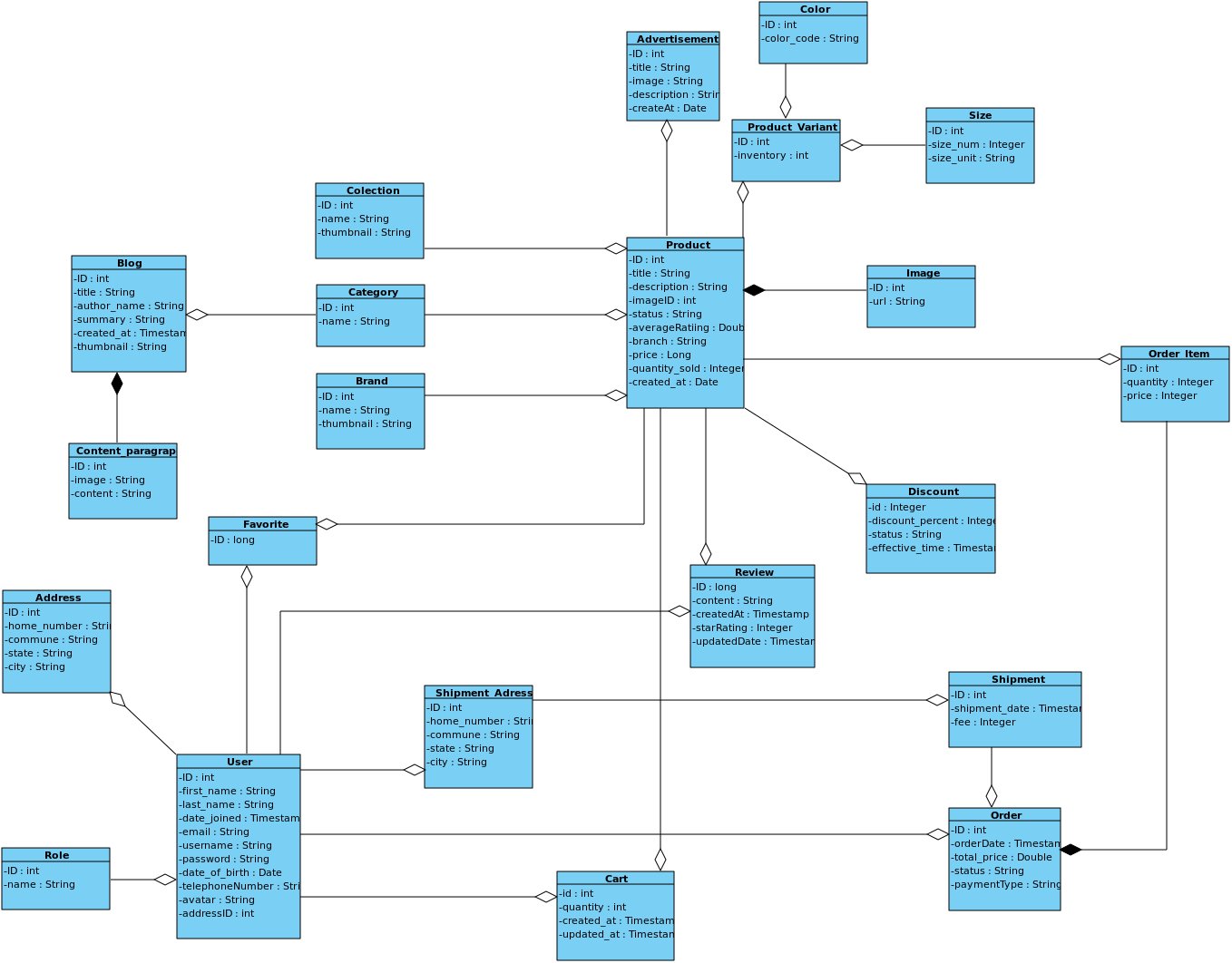
* 1. **Xây dựng biểu đồ lớp phân tích**
     1. **Trích các lớp thực thể hệ thống**

Các lớp thực thể của hệ thống bao gồm:

* Tài khoản: Product
* Người dùng: User
* Vai trò: Role
* Địa chỉ: Address
* Sản phẩm yêu thích: Favorite
* Bài viết: Blog
* Thể loại: Category
* Bộ sưu tập: Collection
* Thương hiệu: Brand
* Quảng cáo: Advertisement
* Màu sắc: Color
* Kích thước: Size
* Ảnh: Image
* Đơn hàng: Order
* Giảm giá: Discount
* Shipment: Đơn vận chuyển
* Giỏ hàng: Cart
* Đơn hàng: Order

Đề xuất các lớp bổ sung:

* Đoạn văn trong bài viết: Content\_paragraph
* Biến thể sản phẩm: Product\_Variant
* Phần tử đặt hàng: Order\_Item
* Địa chỉ vận chuyển: Shipment\_Address
  + 1. **Xác định quan hệ giữa các lớp**
* Mỗi người dùng chỉ có 1 vai trò, mỗi vai trò có nhiều người dùng.
* Mỗi người dùng có 1 địa chỉ, mỗi địa chỉ có nhiều người dùng.
* Mỗi Blog có nhiều đoạn văn, mỗi đoạn chỉ thuộc duy nhất 1 blog
* Mỗi Thể loại có nhiều Blog, mỗi Blog chỉ thuộc 1 thể loại
* Mỗi người dùng có nhiều Sản phẩm Yêu thích (Favarite), Mỗi sản phẩm yêu thích (Favorite) chỉ thuộc 1 người dùng
* Mỗi sản phẩm yêu thích tương ứng với 1 sản phẩm, mỗi sản phẩm có thể là nhiều sản phẩm yêu thích
* Mỗi sản phẩm thuộc 1 bộ sưu tập, mỗi bộ sưu tập có nhiều sản phẩm
* Mỗi sản phẩm thuộc 1 thể loại, mỗi thể loại có nhiều sản phẩm
* Mỗi sản phẩm thuộc 1 thương hiệu, mỗi thương hiệu có nhiều sản phẩm
* Mỗi sản phẩm thuộc 1 quảng cáo, mỗi quảng cáo cho 1 sản phẩm
* Mỗi sản phẩm có nhiều biến thể (Product\_Variant), mỗi biến thể chỉ thuộc một sản phẩm
* Mỗi biến thể gồm một màu và một kích thước, mỗi màu mỗi kích thước có thể thuộc nhiều biến thể
* Mỗi sản phẩm có nhiều ảnh, mỗi ảnh chỉ thuộc một sản phẩm
* Mỗi sản phẩm có 1 phần giảm giá, mỗi phần giảm giá thuộc 1 sản phẩm
* Mỗi Đánh giá áp dụng cho một sản phẩm và được thực hiện bởi 1 người dùng, Mỗi người dùng có thể có nhiều đánh giá, mỗi sản phẩm có thể có nhiều đánh giá.
* Mỗi giỏ hàng chỉ thuộc 1 người dùng, và chứa 1 sản phẩm, Mỗi người dùng có nhiều giỏ hàng, và mỗi sản phẩm thuộc nhiều giỏ hàng
* Mỗi Order Item chỉ chứa 1 sản phẩm, mỗi sản phẩm có thể thuộc nhiều Order Item
* Mỗi Đơn hàng (Order) chứa nhiều Order Item, Mỗi Order Item chỉ thuộc 1 đơn hàng
* Mỗi người dùng có nhiều Order, mỗi Order chỉ thuộc về một người dùng
* Mỗi người dùng có nhiều địa chỉ giao hàng, mỗi địa chỉ giao hàng chỉ thuộc một người dùng
* Mỗi đơn vận chuyển (Shipment) chỉ có một địa chỉ giao hàng, mỗi địa chỉ giao hàng có thể thuộc nhiều đơn vận chuyển
* Mỗi đơn hàng chứa một đơn vận chuyển, mỗi đơn vận chuyển thuộc một đơn hàng
  + 1. **Biểu đồ lớp phân tích**



Hình 3.23 Biểu đồ lớp phân tích

* 1. **Thiết kế kiến trúc microservie**
     1. **Tổng quan**

Hệ thống được xây dựng dựa trên kiến trúc Microservice gồm 7 service:

* User-service
* Product-service
* Review-service
* Cart-service
* Order-service
* Payment-service
* Shipment-service



Hình 3.24 Lược đồ phân rã microservice

* + 1. **Các thành phần hệ thống**

Bảng 3.40 Các thành phần trong hệ thống microserivce

|  |  |  |
| --- | --- | --- |
| STT | Tên | Mô tả |
| 1 | API Gateway | Xứ lí xác thực các yêu cầu từ client, thực hiện định tuyến các yêu cầu tới service tương ứng |
| 2 | User-service | Xử lí các nghiệp vụ liên quan đến tài khoản, thông tin người dùng, đăng nhập, đăng kí |
| 3 | Product-service | Xử lí các nghiệp vụ liên quan đến sản phẩm như tạo mới, hiển thị sản phẩm, thêm sản phẩm vào danh tài khoản yêu thích,… |
| 4 | Review-service | Xử lí các nghiệp vụ liên quan đến bình luận, đánh giá sản phẩm, tích hợp hệ thống gợi ý dựa trên đánh giá người dùng về sản phẩm |
| 5 | Cart-service | Xử lí các nghiệp vụ liên quan đến giỏ hàng như thêm giỏ hàng, hiển thị danh tài khoản giỏ hàng |
| 6 | Order-serivce | Xử lí nghiệp vụ liên quan đến đặt hàng như thêm đơn hàng, xem lịch sử giao dịch đơn hàng |
| 7 | Payment | Xử lí các nghiệp vụ liên quan đến thanh toán, tích hợp API VNPay để thực hiện thanh toán online |
| 8 | Shipment | Xử lí các nghiệp vụ liên quan đến tạo đơn vận chuyển, các tích hợp với API giao hàng nhau để tính số tiền ship của đơn hàng |

* 1. **Thiết kế các biểu đồ tuần tự hệ thống**
     1. **Biểu đồ tuần tự các chức năng của người dùng**

1. Biểu đồ tuần tự của usecase “Đăng ký”

A diagram of a diagram

Description automatically generated

Hình 3.25 Biểu đồ tuấn tự chức năng Đăng ký

1. Biểu đồ tuần tự của usecase “Đăng nhập”

A diagram of a diagram

Description automatically generated

HÌnh 3.26 Biểu đồ tuần tự chức năng đăng nhập

1. Biểu đồ tuần tự của usecase “Quên mật khẩu”

A diagram of a project

Description automatically generated

Hình 3.27 Biểu đồ tuần tự chức năng quên mật khẩu

1. Biểu đồ tuần tự của usecase “Xem thông tin cá nhân”

A diagram of a diagram

Description automatically generated

Hình 3.28 Biểu đồ tuần tự chức năng xem thông tin cá nhân

1. Biểu đồ tuần tự của usecase “Thay đổi thông tin cá nhân”

A diagram of a diagram

Description automatically generated

Hình 3.29 Biểu đồ tuần tự chức năng thay đổi thông tin cá nhân

1. Biểu đồ tuần tự của usecase “Đánh giá”

A diagram of a project

Description automatically generated

Hình 3.30 Biểu đồ tuần tự chức năng Đánh giá

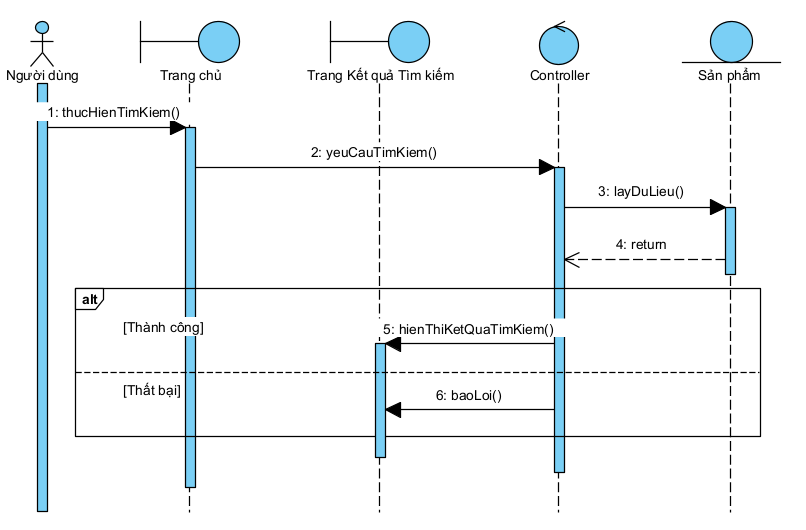
1. Biểu đồ tuần tự của usecase “Gợi ý tài khoản”

A diagram of a diagram

Description automatically generated

Hình 3.31 Biểu đồ tuần tự chức năng Gợi ý tài khoản

1. Biểu đồ tuần tự của usecase “Tìm kiếm sản phẩm”



Hình 3.32 Biểu đồ tuần tự chức năng Tìm kiếm sản phẩm

1. Biểu đồ tuần tự của usecase “Thêm vào giỏ hàng”

A diagram of a diagram

Description automatically generated

Hình 3.33 Biểu đồ tuần tự chức năng Thêm vào giỏ hàng

1. Biểu đồ tuần tự của usecase “Đặt hàng”

A diagram of a project

Description automatically generated

Hình 3.34 Biểu đồ tuần tự chức năng Đặt hàng

1. Biểu đồ tuần tự của usecase “Xem những tài khoản yêu thích”

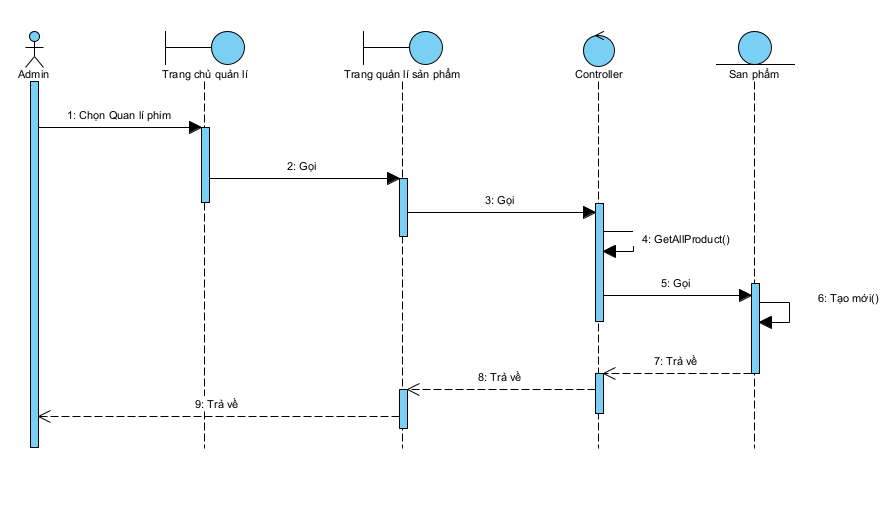
A diagram of a diagram

Description automatically generated

Hình 3.35 Biểu đồ tuần tự chức năng Xem những tài khoản yêu thích

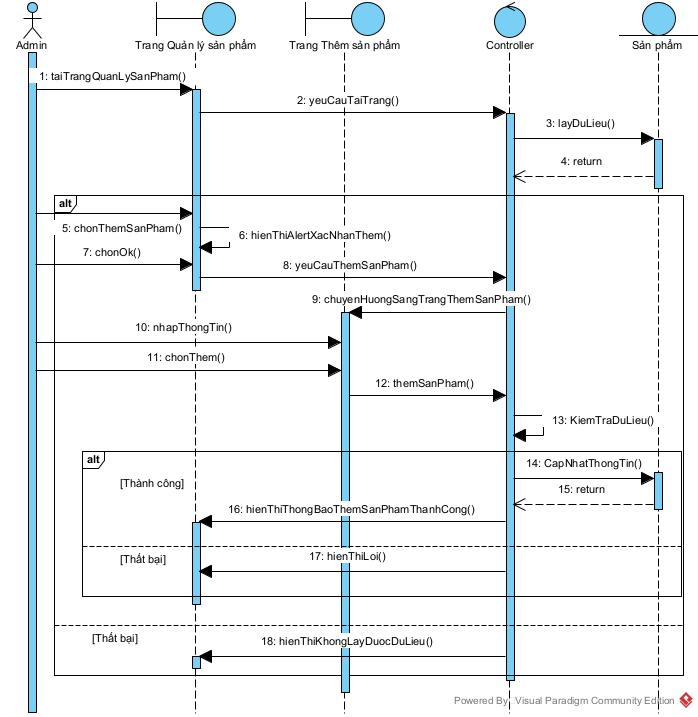
* + 1. **Biểu đồ tuần tự các chức năng của quản lí**

1. Biểu đồ tuần tự của usecase “Xem danh tài khoản tài khoản”



Hình 3.36 Biểu đồ tuần tự chức năng Xem những tài khoản

1. Biểu đồ tuần tự của usecase “Thêm tài khoản mới”



Hình 3.37 Biểu đồ tuần tự chức năng thêm tài khoản mới

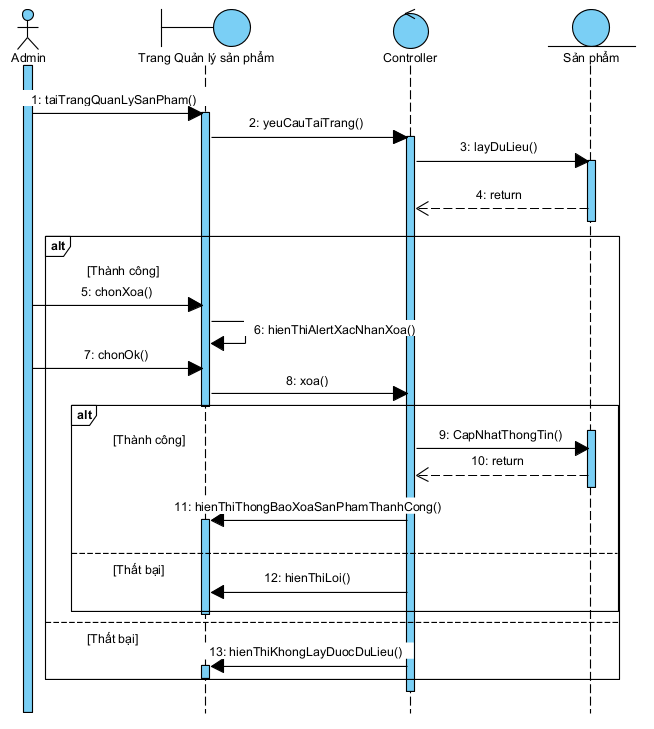
1. Biểu đồ tuần tự của usecase “Sửa tài khoản”

A diagram of a diagram

Description automatically generated

Hình 3.38 Biểu đồ tuần tự chức năng sửa tài khoản

1. Biểu đồ tuần tự của usecase “Xóa tài khoản”



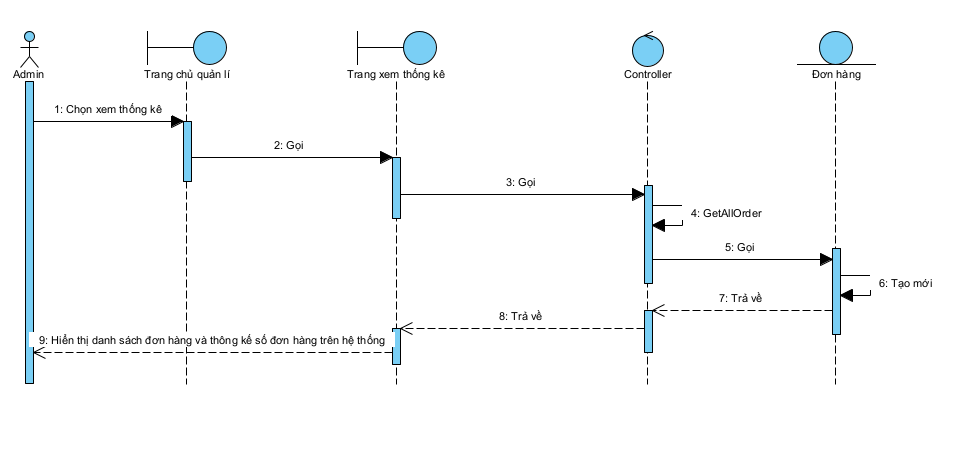
Hình 3.39 Biểu đồ tuần tự chức năng xóa tài khoản

1. Biểu đồ tuần tự của usecase “Xóa người dùng”



Hình 3.40 Biểu đồ tuần tự chức năng xóa người dùng

1. Biểu đồ tuần tự của usecase “Thống kê đơn hàng”



Hình 3.41 Biểu đồ tuần tự chức năng thống kê đơn hàng

1. Biểu đồ tuần tự của usecase “Thống kê doanh thu”

A diagram of a diagram

Description automatically generated

Hình 3.42 Biểu đồ tuần tự chức năng thống kê doanh thu

* 1. **Thiết kế cơ sở dữ liệu**
     1. **Lược đồ cơ sở dữ liệu hệ thống**

A diagram of a computer

Description automatically generated

Hình 3.43 Lược đồ cơ sở dữ liệu hệ thống

* 1. **Kết luận**

Trong chương này, ta đã phân tích hệ thống về các chức năng cùng biểu đồ ca sử dụng hay các biểu đồ usecase và kịch bản phân rã. Chương đã cung cấp biểu đồ thực thể của pha phân tích một cách đầy đủ rõ ràng.

Ta cũng đã thực hiện xong bước thiết kế hệ thống bán hàng trực tuyến. Gồm có biểu đồ tuần tự các chức năng, lược đồ cơ sở dữ liệu và mô tả lược đồ cơ sở dữ liệu để hiểu rõ hơn về các bảng và các trường của từng bảng.

# **: Cài đặt hệ thống**

* 1. **Công cụ hỗ trợ**
     1. **Một số công cụ**
* Visual Studio Code
* MySQL
  1. **Chạy ứng dụng dưới local**

Các service chạy dưới local lần lượt triển khai trên các port:

* User-service: 8001
* Product-serivce: 8002
* Review-service: 8003
* Cart-service: 8004
* Order-service: 8005
* Payment-service: 8006
* Shipment-service: 8007
* API Gateway: 8000

Sau khi triển khai service thì ta sẽ thực hiện triển khai phần web client trên các địa chỉ lần lượt là:

* User-website: <http://127.0.0.1:3000/>
* Admin-website: <http://127.0.0.1:3001/>
  1. **Deployment**

Phần website front-end sẽ được triển khai trên Vercel: Vercel là một nền tảng đám mây (cloud platform) được tạo ra để phục vụ việc phát triển và triển khai ứng dụng web nhanh chóng và dễ dàng. Nền tảng này cho phép xây dựng, triển khai và quản lý các ứng dụng web mà không cần quan tâm đến việc cấu hình hệ thống máy chủ.

A screenshot of a computer

Description automatically generated

Hình 4.1 Triển khai front-end trên Vercel

Sau khi triển khai xong ta có được phần User và Admin đã được triển khai trên các tên miền mặc định của Vercel

* 1. **Kết quả cài đặt**
     1. **Kết quả áp dụng thuật toán gợi ý**
        1. *Đối với người dùng mới*

Khi một người dùng mới đăng ký tài khoản xong và vào trang chủ hệ thống. Do người dùng chưa đánh giá bất cứ sản phẩm nào nên hệ thống sẽ đưa ra gợi ý dựa trên các sản phẩm được đánh giá số sao cao nhất.

A screenshot of a cell phone

Description automatically generated

Hình 4.2 Gợi ý sản phẩm trên trang chủ

Khi người dùng click vào một tài khoản để xem, hệ thống sẽ gợi ý chi tiết hơn về những tài khoản tương tự nhất với tài khoản người dùng vừa click. Ví dụ khi người dùng click chọn tài khoản “iPhone 15 Plus 128GB | Chính hãng VN/A”, hệ thống sẽ đưa những tài khoản tương tự nhất với tài khoản “iPhone 15 Plus 128GB | Chính hãng VN/A”

A screenshot of a phone

Description automatically generated

Hình 4.3 Gợi ý sản phẩm trong trang Chi tiết sản phẩm

* + - 1. *Đối với người dùng cũ*

Gợi ý được thực hiện tương tự người dùng mới, khi đó hệ thống sẽ đưa ra sản phẩm các thể khác với các sản phẩm được đánh sao cao nhất mà tùy thuộc vào sở thích và các lượt đánh giá của người dùng đó.

* + - 1. *Kết quả sau khi áp dụng gợi ý*
* Đối với người dùng:
* Tìm ra những tài khoản phù hợp nhất với mình: Hệ thống gợi ý theo đa số người dùng thích chúng, và được sắp xếp theo thứ tự đánh giá giảm dần
* Gợi ý một nhóm các tài khoản tương tự nhau: Người dùng vào chi tiết một tài khoản, hệ thống sẽ gợi ý nhóm những tài khoản tương tự với tài khoản vừa được click.
* Bày tỏ ý kiến cá nhân người dùng: Người dùng có thể đánh giá bất kỳ tài khoản nào thích hoặc không thích.
* Tác động đến những người dùng khác: Một số người dùng đánh giá tài khoản với mục đích tác động lên hệ thống gợi ý, do đó ảnh hưởng đến những người dùng khác
* Đối với nhà cung cấp tài khoản:
* Tăng số lượng tài khoản được tiếp cận đến người dùng
* Cải thiện sự hài lòng của người dùng: Khi người dùng tìm được tài khoản mong muốn từ hệ thống mà không cần thao tác tìm kiếm.
* Nâng cao sự trung thành của người dùng: Khi đã hài lòng về hệ thống thì người dùng sẽ coi như đây là web “ruột” của mình và sẽ là lựa chọn đầu tiên khi cần giải trí
* Hiểu biết tốt hơn về nhu cầu người dùng: Quản lý có thể xem số lượng đơn hàng, đánh giá trung bình của tài khoản, từ đó có thể tìm hiểu ra nhu cầu và cải thiện hơn về các tài khoản đưa lên.
  + 1. **Giao diện phía người dùng**

Giao diện trang chủ

A screenshot of a computer

Description automatically generated

Hình 4.4 Giao diện trang chủ hệ thống

Giao diện xem chi tiết sản phẩm

A screenshot of a cell phone

Description automatically generated

Hình 4.5 Giao diện chi tiết sản phẩm

Giao diện bình luận

A screenshot of a computer

Description automatically generated

Hình 4.6 Giao diện bình luận

Giao diện Giỏ hàng

A screenshot of a computer

Description automatically generated

Hình 4.7 Giao diện giỏ hàng

Giao diện Thanh toán qua VNPay:

A screenshot of a phone

Description automatically generated

Hình 4.8 Thanh toán qua VNPay

Giao diện thanh toán thành công:

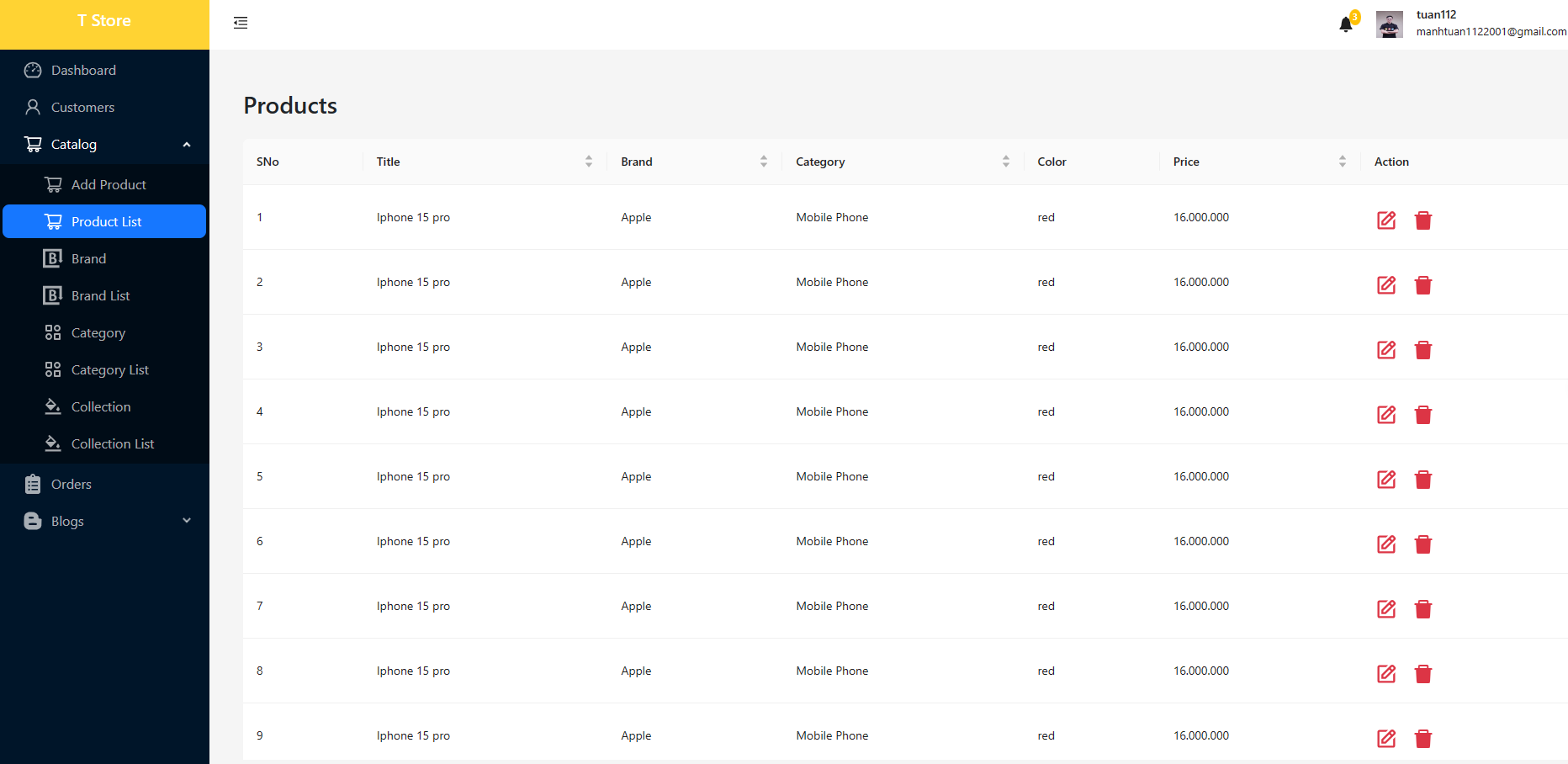
A screenshot of a computer

Description automatically generated

Hình 4.9 Giao diện thành toán thành công

* + 1. **Giao diện trang quản trị**

Giao diện quản lý tài khoản



Hình 4.10 Giao diện quản lý tài khoản

Giao diện thêm tài khoản

A screenshot of a computer

Description automatically generated

Hình 4.11 Giao diện thêm tài khoản

Giao diện thông kê đơn hàng

A screenshot of a computer

Description automatically generated

Hình 4.12 Giao diện thống kế đơn hàng

* 1. **Kết luận**

Ở chương này, đồ án đã hoàn thành các bước cài đặt môi trường triển khai hệ thống bán hàng trực tuyến và kiểm thử hệ thống.

Về phía server, tiến hành cài đặt môi trường Python, cài đặt công cụ quản trị cơ sở dữ liệu MySQL, kèm theo đó là các plugin phục vụ cho việc lập trình. Hệ thống được triển khai chạy trên trình duyệt Chrome, khi hoàn thành mỗi chức năng sẽ thực hiện kiểm thử, nếu phát hiện lỗi thì fix (chỉnh sửa) luôn cho đến khi hoàn thiện hệ thống đúng yêu cầu đặt ra.

Trong phần tiếp theo, đồ án sẽ đưa ra kết quả đạt được của hệ thống cũng như những hạn chế còn tồn đọng, đồng thời định hướng phát triển hệ thống trong tương lai.

# **KẾT LUẬN**

Được sự hướng dẫn của thầy, trong thời gian làm đồ án, em đã thực hiện được những nội dung sau:

**Kết quả đạt được**

* Website đã hoàn thành được những mục tiêu ban đầu đặt ra
* Hoàn thành được các chức năng cơ bản:
* Quản lý thông tin tài khoản, tập tài khoản, gói dịch vụ,…
* Thống kê đơn hàng, doanh thu
* Tích hợp nhiều phương thức Thanh toán online
* Tích hợp thuật toán gợi ý tài khoản cho người dùng
* Người dùng có thể tìm kiếm tài khoản, bán tài khoản, bình luận, đánh giá, yêu thích sản phẩm,..
* Người dùng có thể quản lý thông tin cá nhân
* Giao diện thân thiện với người dùng, cơ bản có đầy đủ các phần của website bán tài khoản trực tuyến
* Tích hợp được gợi ý sản phẩm cho người dùng, bao gồm cả người dùng mới

**Các hạn chế**

* Các chức năng đang còn ít và mới chỉ dừng lại ở mức xây dựng cơ bản
* Chức năng bảo mật thông tin người dùng chưa cao, còn gặp nhiều hạn chế
* Thuật toán gợi ý cơ bản, chưa đi sâu vào các kỹ thuật phức tạp
* Cách thiết kế giao diện, màu sắc website chưa chuyên nghiệp
* Chưa deploy các service lên môi trường production mà mới chỉ chạy ở dưới local
* Hệ thống còn nhỏ nên áp dụng microservice vẫn chưa được hiệu quả, database chưa đủ lớn lên khi phân tách gặp nhiều khó khăn do sự rành buộc giữa các bảng

**Định hướng phát triển**

* Triển khai các service lên môi trường production
* Tái cấu trúc lại các service để các service này chuyên sâu và độc lập hơn, ít phụ thuộc vào nhau nhất có thể
* Tích hợp thêm một số thuật toán gợi ý chính xác hơn trong gợi ý tài khoản
* Tối ưu hóa code và câu truy vấn dữ liệu để thời gian thực hiện được nhanh hơn
* Cải thiện chức năng bảo mật thông tin người dùng
* Học thêm về thiết kế website để giao diện, màu sắc hài hòa, bắt mắt.

# **DANH MỤC TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | "W3Schools," [Online]. Available: https://www.w3schools.com/html/html\_intro.asp. |
| [2] | "W3schools," [Online]. Available: https://www.w3schools.com/css/. |
| [3] | "WikipediA," [Online]. Available: https://vi.wikipedia.org/wiki/%C4%90%E1%BB%99\_t%C6%B0%C6%A1ng\_t%E1%BB%B1\_cosin. |
| [4] | "WikipediA," [Online]. Available: https://machinelearningcoban.com/2017/05/24/collaborativefiltering/. |
| [5] | "Microservices Guide," [Online]. Available: https://martinfowler.com/microservices/. |
| [6] | Đ.N.Phong, in *Slide môn Phân tích và Thiết kế Hệ thống Thông tin*, Học viện Công nghệ Bưu chính Viễn thông. |