

**2025.5**

**KHOA HỌC MÁY TÍNH**

**DƯƠNG QUỐC HƯNG**

**11212455**

**ĐẠI HỌC KINH TẾ QUỐC DÂN TRƯỜNG CÔNG NGHỆ**



**KHÓA LUẬN TỐT NGHIỆP**

Xây dựng Website bán đồng hồ

**Tên sinh viên : Dương Quốc Hưng Giảng viên hướng dẫn : ThS. Trần Thị Mỹ Diệp**

**Hà Nội, 5/2025**

**ĐẠI HỌC KINH TẾ QUỐC DÂN TRƯỜNG CÔNG NGHỆ**





**KHÓA LUẬN TỐT NGHIỆP**

Xây dựng hệ thống bán đồng hồ trực tuyến

**Tên sinh viên : Dương Quốc Hưng**

**Mã sinh viên 11212455**

**Lớp : Khoa học máy tính 63**

**Ngành : Khoa học máy tính**

**Khoa : Công nghệ thông tin Giảng viên hướng dẫn : ThS. Trần Thị Mỹ Diệp**

*(Chữ ký GVHD)*

**Hà Nội, 2/2025**

## MỤC LỤC

[MỤC LỤC 3](#_bookmark0)

[DANH MỤC CÁC TỪ VIẾT TẮT 6](#_bookmark1)

[DANH MỤC BẢNG 7](#_bookmark2)

[DANH MỤC HÌNH ẢNH 8](#_bookmark3)

[LỜI CAM ĐOAN 9](#_bookmark4)

[LỜI CẢM ƠN 10](#_bookmark5)

[MỞ ĐẦU 11](#_bookmark6)

[CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI 12](#_bookmark7)

* 1. [Đặt vấn đề 12](#_bookmark8)
  2. [Mục tiêu và phạm vi đề tài 13](#_bookmark9)
  3. [Quy trình nghiệp vụ 13](#_bookmark10)
  4. [Các yêu cầu của hệ thống 16](#_bookmark11)
     1. [Yêu cầu về chức năng 16](#_bookmark12)
     2. [Yêu cầu phi chức năng 16](#_bookmark13)
  5. [Công nghệ sử dụng 17](#_bookmark14)
     1. [JavaScript 17](#_bookmark15)
     2. [TypeScript 18](#_bookmark16)
     3. [ReactJS 18](#_bookmark17)
     4. [NestJS 19](#_bookmark18)
     5. [MongoDB 20](#_bookmark19)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 22](#_bookmark20)

* 1. [Phương pháp phân tích hệ thống 22](#_bookmark21)
  2. [Phân tích các chức năng của hệ thống 25](#_bookmark22)
     1. [Biểu đồ Usecase tổng quát 25](#_bookmark23)
     2. [Các biểu đồ use case phân rã 26](#_bookmark25)
     3. [Đặc tả use case 29](#_bookmark32)
  3. [Phân tích hành vi hệ thống 37](#_bookmark42)
     1. [Use case Đăng nhập 37](#_bookmark43)
     2. [Use case Xem thông tin sản phẩm 38](#_bookmark45)
     3. [Use case Đặt hàng 39](#_bookmark47)
     4. [Use case Xem đơn hàng 40](#_bookmark49)
     5. [Use case Quản lý giỏ hàng 41](#_bookmark51)
     6. [Use case Hủy đơn hàng 42](#_bookmark53)
     7. [Use case Quản lý đơn hàng 43](#_bookmark55)
     8. [Use case Cập nhật thông tin 44](#_bookmark57)
     9. [Use case Cập nhật sản phẩm 45](#_bookmark59)
  4. [Phân tích cấu trúc hệ thống 46](#_bookmark61)
     1. [Mô tả các lớp 46](#_bookmark62)
     2. [Mối quan hệ giữa các lớp 47](#_bookmark63)
     3. [Biểu đồ lớp 49](#_bookmark64)
  5. [Thiết kế cơ sở dữ liệu 50](#_bookmark66)
     1. [Mô hình hóa dữ liệu 50](#_bookmark67)
     2. [Thiết kế collection 50](#_bookmark69)
     3. [Quan hệ giữa các collection 53](#_bookmark70)
  6. [Thiết kế mô hình gợi ý 54](#_bookmark71)
     1. [Khái niệm 54](#_bookmark72)
     2. [Hướng tiếp cận 54](#_bookmark73)
     3. [Xây dựng Content-based Filtering RS (Recommender System) 56](#_bookmark75)

[CHƯƠNG 3. GIỚI THIỆU VỀ WEBSITE 60](#_bookmark78)

* 1. [Giới thiệu tính năng bên phía khách hàng 60](#_bookmark79)
     1. [Trang chủ 60](#_bookmark80)
     2. [Đăng nhập 61](#_bookmark82)
     3. [Sản phẩm 62](#_bookmark84)
     4. [Chi tiết sản phẩm 62](#_bookmark86)
     5. [Giỏ hàng 63](#_bookmark88)
     6. [Đặt hàng 64](#_bookmark90)
     7. [Thông báo đặt hàng thành công 65](#_bookmark92)
  2. [Giới thiệu chức năng bên phía quản trị viên 66](#_bookmark94)
     1. [Quản lý đơn hàng 66](#_bookmark95)
     2. [Quản lý doanh thu 67](#_bookmark97)
  3. [Kết quả thực nghiệm mô hình gợi ý trên website 68](#_bookmark99)
     1. [Thực nghiệm 68](#_bookmark100)
     2. [Tổng kết 70](#_bookmark103)

[KẾT LUẬN 71](#_bookmark104)

[TÀI LIỆU THAM KHẢO 73](#_bookmark105)

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Ý nghĩa** |
| CSDL | Cơ sở dữ liệu |
| SQL | Structured Query Language |
| API | Application programming interface |
| JSX | Javascrip XML |

# DANH MỤC BẢNG

[Bảng 2.1 Đặc tả usecase Đăng nhập tài khoản 29](#_bookmark33)

[Bảng 2.2 Đặc tả usecase xem thông tin sản phẩm 30](#_bookmark34)

[Bảng 2.3 Đặc tả usecase cập nhật thông tin 30](#_bookmark35)

[Bảng 2.4 Đặc tả usecase xem đơn hàng 31](#_bookmark36)

[Bảng 2.5 Đặc tả usecase quản lý giỏ hàng 32](#_bookmark37)

[Bảng 2.6 Đặc tả usecase cập nhật đơn hàng 33](#_bookmark38)

[Bảng 2.7 Đặc tả usecase đặt hàng 34](#_bookmark39)

[Bảng 2.8 Đặc tả usecase quản lý đơn hàng 35](#_bookmark40)

[Bảng 2.9 Đặc tả usecase cập nhật sản phẩm 36](#_bookmark41)

[Bảng 3.1 Kết quả thực nghiệm mô hình gợi ý 68](#_bookmark101)

[Bảng 3.2 Kết quả tính toán thực nghiệm 68](#_bookmark102)

# DANH MỤC HÌNH ẢNH

[Hình 2.1 Biểu đồ use case tổng quát 25](#_bookmark24)

[Hình 2.2 Phân rã chức năng quản lý giỏ hàng 26](#_bookmark26)

[Hình 2.3. Phân rã chức năng cập nhật đơn hàng 26](#_bookmark27)

[Hình 2.4. Phân rã chức năng quản lý đơn hàng 27](#_bookmark28)

[Hình 2.5. Phân rã chức năng cập nhật danh mục 27](#_bookmark29)

[Hình 2.6. Phân rã chức năng cập nhật người dùng 28](#_bookmark30)

[Hình 2.7. Chức năng cập nhật sản phẩm 28](#_bookmark31)

[Hình 2.8. Biểu đồ trình tự use case Đăng nhập 37](#_bookmark44)

[Hình 2.9. Biểu đồ trình tự use case Xem thông tin sản phẩm 38](#_bookmark46)

[Hình 2.10. Biểu đồ trình tự use case Đặt hàng 39](#_bookmark48)

[Hình 2.11. Biểu đồ trình tự use case Xem đơn hàng 40](#_bookmark50)

[Hình 2.12. Biểu đồ trình tự Quản lý giỏ hàng 41](#_bookmark52)

[Hình 2.13 Biểu đồ trình tự use case Hủy đơn hàng 42](#_bookmark54)

[Hình 2.14 Biểu đồ trình tự use case Quản lý đơn hàng 43](#_bookmark56)

[Hình 2.15 Biểu đồ trình tự use case Cập nhật thông tin 44](#_bookmark58)

[Hình 2.16 Biểu đồ trình tự usecase Cập nhật sản phẩm 45](#_bookmark60)

[Hình 2.17 Biểu đồ lớp 49](#_bookmark65)

[Hình 2.18 Các bảng trong CSDL 50](#_bookmark68)

[Hình 2.19. Mô hình các hệ thống gợi ý phổ biến 55](#_bookmark74)

[Hình 2.20. Ví dụ về điểm tương đồng trong thực tế 56](#_bookmark76)

[Hình 2.21. Ví dụ về điểm tương đồng của 2 sản phẩm trong database 57](#_bookmark77)

[Hình 3.1 Giao diện trang chủ website 60](#_bookmark81)

[Hình 3.2 Giao diện đăng nhập 61](#_bookmark83)

[Hình 3.3 Giao diện danh sách sản phẩm 62](#_bookmark85)

[Hình 3.4 Giao diện chi tiết sản phẩm 62](#_bookmark87)

[Hình 3.5 Giao diện giỏ hàng 63](#_bookmark89)

[Hình 3.6 Giao diện đặt hàng 64](#_bookmark91)

[Hình 3.7 Giao diện đặt hàng thành công 65](#_bookmark93)

[Hình 3.8 Giao diện quản lý đơn hàng 66](#_bookmark96)

[Hình 3.9 Giao diện quản lý doanh thu 67](#_bookmark98)

# LỜI CAM ĐOAN

Em xin cam đoan rằng toàn bộ nội dung của báo cáo chuyên đề thực tập với đề tài “Xây dựng hệ thống bán đồng hồ trực tuyến” là kết quả nghiên cứu và làm việc của các nhân em dưới sự hướng dẫn của cô Trần Thị Mỹ Diệp.

Tất cả các tài liệu tham khảo được sử dụng trong báo cáo đã được trích dẫn rõ ràng, đúng quy định. Những số liệu, bảng biểu, và hình ảnh trong báo cáo đều có nguồn gốc rõ ràng, được em tạo ra hoặc được em tự tổng hợp từ quá trình thực hiện báo cáo chuyên đề thực tập.

Em xin chịu hoàn toàn trách nhiệm về tính trung thực và chính xác của các nội dung đã trình bày. Em cam kết rằng báo cáo này không sao chép từ bất kỳ báo cáo hoặc nghiên cứu nào khác, đồng thời chưa từng được sử dụng để nộp cho bất kỳ chương trình đào tạo nào trước đây.

Nếu có bất kỳ sự không trung thực hoặc sai lệch nào được phát hiện, em xin hoàn toàn chịu trách nhiệm và chấp nhận mọi hình thức xử lý theo quy định của nhà trường.

Trân trọng!

Hà Nội, ngày 18/05/2025

# LỜI CẢM ƠN

Để hoàn thành khóa luận tốt nghiệp này, em xin gửi lời tri ân chân thành đến các thầy cô giảng viên thuộc Khoa Công nghệ Thông tin – Trường Công nghệ, Đại học Kinh tế Quốc dân. Nhờ sự tận tụy trong giảng dạy và hướng dẫn của thầy cô mà em có cơ hội học tập, nghiên cứu và từng bước hoàn thiện bản thân trong suốt những năm tháng tại trường.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến cô Trần Thị Mỹ Diệp, người đã tận tình hướng dẫn, chỉ bảo và hỗ trợ em trong suốt hành trình thực hiện khóa luận này. Tấm lòng nhiệt huyết, sự tận tâm cùng những lời khuyên quý giá của cô đã trở thành nguồn động lực lớn lao để em vượt qua những khó khăn và hoàn thành công việc một cách tốt nhất.

Mặc dù đã cố gắng và nỗ lực vào việc nghiên cứu, nhưng với những hạn chế về kiến thức và kinh nghiệm, bài khóa luận của em khó tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp chân thành từ thầy cô và các bạn để hoàn thiện và tiếp tục phát triển dự án này trong tương lai.

Em xin chân thành cảm ơn và gửi lời kính chúc sức khỏe, hạnh phúc đến các thầy cô.

# MỞ ĐẦU

Chúng ta đang sống trong thế kỷ 21 với sự bùng nổ của công nghệ, thương mại điện tử đã phát triển mạnh mẽ và không ngừng mở rộng. Việc mua sắm bán lẻ trực tuyến đã trở nên phổ biến và trở thành thói quen của nhiều người vì nó cho phép người tiêu dùng tiếp cận sản phẩm và mua sắm từ các thiết bị di động có kết nối internet chỉ bằng vài thao tác đơn giản, không bị giới hạn địa lý hay thời gian như trước kia.

Trong bối cảnh đó, ngành kinh doanh bán lẻ đồng hồ cũng cho thấy sự chuyển mình mạnh mẽ từ kinh doanh truyền thống sang thương mại điện tử. Việc mua bán đồng hồ trực tuyến không chỉ mang lại sự tiện lợi, tiết kiệm thời gian, chi phí của người mua mà còn mang đến cho các nhà bán hàng cơ hội mở rộng thị trường, tiết kiệm chi phí, tối ưu hóa quy trình kinh doanh và tăng khả năng cạnh tranh với các nhà bán hàng khác. Bên cạnh nhưng cơ hội, điều này cũng mang lại không ít những thách thức đối với các nhà bán hàng trong vấn đề quản lý kho, tiếp cận khách hàng, chăm sóc và duy trì mối quan hệ với khách hàng. Việc này đòi hỏi phải có một hệ thống quản lý hiệu quả để theo dõi, thống kê mọi hoạt động của cửa hàng một cách nhanh chóng, chính xác và thuận tiện.

Nhận thấy những thách thức cần giải pháp, em đã quyết định lựa chọn đề tài “Xây dựng hệ thống bán đồng hồ trực tuyến”. Mục tiêu của đề tài là xây dựng một hệ thống đơn giản, trực quan, dễ sử dụng có khả năng mạnh mẽ trong việc hỗ trợ quản lý nhằm đáp ứng nhu cầu của cả nhà bán hàng và người mua. Từ đó mang đến cho khách hàng một trải nghiệm mua hàng tiện lợi, an toàn, nhanh chóng, bên cạnh đó cũng giúp nhà bán hàng tối ưu hóa việc quản lý vận hành cửa hàng, giảm thiểu sai sót trong quy trình bán hàng, quản lý kho bãi, giúp nâng cao hiệu quả trong kinh doanh.

Nội dung báo cáo sẽ gồm: Chương 1: Tổng quan đề tài

Chương 2: Phân tích và thiết kế hệ thống Chương 3: Giới thiệu về website

# CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

### Đặt vấn đề

Đồng hồ từ lâu đã vượt xa vai trò của một thiết bị đo lường thời gian. Với nhiều người, nó là biểu tượng của sự tinh tế, phong cách và thậm chí là cá tính riêng. Giữa cuộc sống hiện đại đầy bận rộn, khi công nghệ số bùng nổ và thay đổi cách chúng ta mua sắm, những chiếc đồng hồ không còn chỉ xuất hiện trong các cửa hàng truyền thống. Giờ đây, chúng đã có một "không gian" trực tuyến, nơi người mua và người bán dễ dàng kết nối với nhau chỉ bằng vài cú nhấp chuột.

Thị trường đồng hồ trực tuyến không chỉ mở ra cơ hội cho những người đam mê sưu tập đồng hồ, mà còn giúp các cửa hàng nhỏ lẻ hoặc các nhà phân phối độc lập tìm thấy khách hàng của mình. Tuy nhiên, không phải ai cũng dễ dàng thích nghi với "sân chơi" này. Việc quản lý một cửa hàng trực tuyến đòi hỏi không chỉ là khả năng kinh doanh mà còn yêu cầu những giải pháp công nghệ hiệu quả. Quản lý kho hàng, theo dõi đơn đặt hàng, xử lý thanh toán hay thống kê doanh thu – tất cả những điều này có thể trở thành một bài toán đầy thách thức nếu thiếu đi một hệ thống quản lý hiện đại.

Các giải pháp truyền thống như ghi chép tay hoặc sử dụng bảng tính Excel giờ đây khó có thể đáp ứng nhu cầu của một môi trường trực tuyến đang thay đổi từng ngày. Một hệ thống quản lý tối ưu, được thiết kế đặc biệt cho lĩnh vực kinh doanh đồng hồ trực tuyến, không chỉ là công cụ cần thiết mà còn là "chìa khóa" để các cửa hàng vận hành hiệu quả hơn, giảm thiểu sai sót và tối ưu hóa quy trình.

Từ nhu cầu thực tế và tiềm năng phát triển của thị trường, ý tưởng xây dựng hệ thống quản lý bán đồng hồ trực tuyến đã ra đời. Mục tiêu của hệ thống này không chỉ là hỗ trợ các cửa hàng quản lý sản phẩm, khách hàng, doanh thu mà còn tạo ra một không gian thương mại số chuyên nghiệp, dễ sử dụng và đáp ứng đầy đủ nhu cầu cả từ phía người bán lẫn người mua. Đây không chỉ là một dự án công nghệ mà còn là một bước tiến thúc đẩy sự phát triển của ngành kinh doanh đồng hồ trong thời đại số, nơi mà từng giây phút đều quý giá và đáng được tôn vinh.

### Mục tiêu và phạm vi đề tài

Mục tiêu của đề tài là phát triển một hệ thống bán đồng hồ trực tuyến hiệu quả, dễ sử dụng và đáp ứng đầy đủ các chức năng cơ bản như tìm kiếm đồng hồ, giỏ hàng, thanh toán trực tuyến. Hệ thống phải đảm bảo mang đến cho khách hàng một trải nghiệm mua hàng tiện lợi và an toàn.

Đề tài tập trung vào việc nghiên cứu và phát triển một giải pháp công nghệ giúp người dùng và nhà bán hàng dễ dàng tìm thấy nhau, giúp người dùng tìm được chiếc đồng hồ phù hợp và nhà bán hàng dễ dàng tiếp cận người dùng. Trang web cung cấp các chức năng chính như:

* Kênh kết nối giữa người bán và người tiêu dùng, cung cấp trải nghiệm mua sắm trực tuyến tiện lợi, hiện đại và nhanh chóng.
* Giao diện thân thiện với người dùng, giúp người dùng dễ dàng tìm kiếm và mua hàng
* Tích hợp tính năng thanh toán trực tuyến qua ZaloPay, cho phép người mua thực hiện giao dịch nhanh chóng, an toàn qua mã QR, đồng thời hỗ trợ các hình thức thanh toán khác như COD.
* Trang Admin cho phép quản trị viên theo dõi đơn hàng, quản lý sản phẩm và xem báo cáo kinh doanh.

### Quy trình nghiệp vụ

Quy trình nghiệp vụ của trang web bán đồng hồ sẽ bao gồm những bước sau:

### Đăng nhập tài khoản

Người dùng truy cập vào trang đăng nhập và nhập thông tin tài khoản bao gồm tên đăng nhập và mật khẩu. Hệ thống sẽ kiểm tra thông tin đăng nhập. Nếu thông tin không chính xác, hệ thống sẽ hiển thị thông báo lỗi “Tên tài khoản hoặc mật khẩu không đúng”. Nếu thông tin hợp lệ, người dùng sẽ được đăng nhập vào hệ thống chuyển đến trang chủ, đồng thời có thể truy cập các chức năng dành cho người dùng như giỏ hàng, lịch sử mua hàng.

### Khách hàng xem sản phẩm

Khi người dùng truy cập vào website, hệ thống sẽ hiển thị giao diện trang chủ với các danh mục sản phẩm hoặc gợi ý nổi bật. Người dùng có thể sử dụng thanh tìm kiếm để tìm loại sản phẩm mà họ quan tâm. Sau khi nhập từ khóa tìm kiếm, hệ thống sẽ trả về danh sách các sản phẩm phù hợp kèm theo thông tin cơ bản như tên sản phẩm, hình ảnh, giá và đánh giá. Người dùng có thể nhấp vào từng sản phẩm để xem chi tiết, bao gồm mô tả sản phẩm, thông số kỹ thuật, hình ảnh mở rộng, tùy chọn kích thước hoặc màu sắc. Ngoài ra, hệ thống cũng có thể gợi ý các sản phẩm liên quan để tăng trải nghiệm mua sắm. Sau khi xem xét, người dùng có thể quay lại danh sách sản phẩm, thêm sản phẩm vào giỏ hàng hoặc tiếp tục duyệt các sản phẩm khác.

### Mua hàng (Thêm sản phẩm vào giỏ hàng)

Người dùng duyệt các sản phẩm trên trang chủ hoặc thông qua tìm kiếm. Khi nhấn vào một sản phẩm, hệ thống chuyển đến trang chi tiết sản phẩm, nơi người dùng có thể xem hình ảnh, mô tả, giá và chọn số lượng. Người dùng nhấn nút “Thêm vào giỏ hàng”, hệ thống sẽ cập nhật giỏ hàng tương ứng và hiển thị thông báo “Sản phẩm đã được thêm vào giỏ hàng thành công”. Người dùng có thể tiếp tục mua hàng hoặc vào giỏ để tiến hành thanh toán.

### Thanh toán

Người dùng truy cập trang giỏ hàng để xem danh sách sản phẩm đã chọn. Tại đây, họ có thể điều chỉnh số lượng hoặc xóa sản phẩm. Khi nhấn nút “Thanh toán”, người dùng được chuyển đến trang nhập thông tin giao hàng. Sau khi nhập đầy đủ thông tin (họ tên, số điện thoại, địa chỉ), người dùng chọn phương thức thanh toán. Nếu chọn ZaloPay, hệ thống sẽ tạo mã QR và hiển thị để người dùng quét thanh toán qua ứng dụng ZaloPay.

* + Nếu thanh toán thành công: hệ thống cập nhật đơn hàng sang trạng thái “Đã thanh toán” và chuyển hướng đến trang xác nhận.
  + Nếu thanh toán thất bại hoặc bị hủy: hệ thống hiển thị thông báo lỗi và cho phép người dùng thử lại.

### Theo dõi đơn hàng

Đối với người dùng cá nhân, sau khi đăng nhập vào tài khoản trên website, họ có thể truy cập mục “Đơn hàng của tôi” trong trang cá nhân. Tại đây, hệ thống sẽ hiển thị danh sách các đơn hàng mà người dùng đã đặt, bao gồm thông tin như mã đơn hàng, ngày đặt, trạng thái xử lý (đang xử lý, đã giao, đã hủy, v.v.) và tổng giá trị đơn hàng. Người dùng có thể nhấp vào từng đơn để xem chi tiết, bao gồm danh sách sản phẩm, địa chỉ giao hàng, hình thức thanh toán và lịch sử cập nhật trạng thái.

Đối với quản trị viên (admin), sau khi đăng nhập vào hệ thống quản trị của website, họ có quyền truy cập vào mục “Quản lý đơn hàng”. Tại đây, admin có thể xem toàn bộ danh sách đơn hàng từ tất cả người dùng, lọc và tìm kiếm theo nhiều tiêu chí như mã đơn hàng, trạng thái, khách hàng, ngày đặt, hoặc hình thức thanh toán. Mỗi đơn hàng khi được chọn sẽ hiển thị chi tiết tương tự như phía người dùng, đồng thời cho phép admin cập nhật trạng thái đơn hàng (duyệt đơn, xác nhận thanh toán, chuyển hàng, hoàn tất, hủy đơn) và xử lý các yêu cầu liên quan như hoàn tiền hoặc hỗ trợ khách hàng. Quy trình này giúp admin giám sát hiệu quả hoạt động bán hàng và đảm bảo đơn hàng được xử lý đúng tiến độ.

### Admin cập nhật sản phẩm

Sau khi đăng nhập vào hệ thống quản trị, admin truy cập vào mục “Quản lý sản phẩm”. Tại đây, admin có thể lựa chọn cập nhật một sản phẩm hiện có bằng cách tìm kiếm theo tên, mã sản phẩm hoặc danh mục. Khi chọn sản phẩm cần chỉnh sửa, hệ thống sẽ hiển thị giao diện chỉnh sửa thông tin chi tiết. Sau khi hoàn tất chỉnh sửa, admin nhấn nút “Lưu” để hệ thống cập nhật các thông tin mới vào cơ sở dữ liệu và hiển thị ra giao diện người dùng.

### Các yêu cầu của hệ thống

Để đảm bảo hệ thống quản lý bán đồng hồ trực tuyến hoạt động ổn định, hiệu quả và đáp ứng tốt nhu cầu của cả người quản trị lẫn người dùng cuối, hệ thống cần được xây dựng dựa trên các yêu cầu cụ thể, bao gồm yêu cầu chức năng và phi chức năng. Các yêu cầu này đóng vai trò định hướng cho quá trình phân tích, thiết kế và triển khai hệ thống. Cụ thể:

### Yêu cầu về chức năng

Người quản trị:

* + - * Đăng nhập
      * Quản lý người dùng
      * Quản lý sản phẩm
      * Quản lý danh mục
      * Quản lý mã giảm giá
      * Quản lý đơn hàng Người dùng:
      * Đăng nhập, đăng ký
      * Xem sản phẩm
      * Đặt hàng
      * Quản lý đơn hàng
      * Quản lý giỏ hàng
      * Đánh giá sản phẩm
      * Cập nhật thông tin
      * Xem danh mục

### Yêu cầu phi chức năng

Giao diện:

* + - * Giao diện đơn giản, trực quan và dễ sử dụng

Bảo mật:

* + - * Thực hiện bảo mật bằng cách mã hóa các thông tin nhạy cảm của người dùng
      * Phân quyền truy cập, chỉ cho phép người dùng truy cập những chức năng được cho phép
      * Kiểm định những dữ liệu người dùng cung cấp Hiệu suất:
      * Tốc độ xử lý nhanh chóng, thực hiện loading khi lấy hoặc ghi dữ liệu
      * Hoạt động hiệu quả 24/24

### Công nghệ sử dụng

### JavaScript

JavaScript là một ngôn ngữ lập trình phổ biến trong lĩnh vực phát triển web. Ban đầu, nó được tạo ra để thêm các chức năng tương tác và động vào các trang web. Tuy nhiên, với sự phát triển của cộng đồng phát triển phần mềm, JavaScript đã trở thành một ngôn ngữ lập trình đa năng được sử dụng cho nhiều mục đích khác nhau, bao gồm phát triển ứng dụng web, ứng dụng di động, game, và thậm chí là phần mềm server-side.

Dưới đây là một số điểm nổi bật của JavaScript:

* + - * Động (Dynamic): JavaScript là một ngôn ngữ lập trình động, cho phép thay đổi và tương tác với các thành phần của trang web một cách linh hoạt.
      * Client-Side Scripting: JavaScript thường được sử dụng để thêm các chức năng tương tác và động vào các trang web, bao gồm xử lý sự kiện người dùng, thao tác với DOM (Document Object Model), và gửi và nhận dữ liệu từ máy chủ.
      * Cú pháp dễ hiểu: Cú pháp của JavaScript tương đối đơn giản và dễ hiểu, cho phép người mới bắt đầu dễ dàng học và sử dụng.
      * Cộng đồng lớn và hỗ trợ mạnh mẽ: JavaScript có một cộng đồng phát triển lớn mạnh, với rất nhiều thư viện và framework được phát triển để hỗ trợ các mục đích khác nhau.

JavaScript là một phần không thể thiếu trong việc phát triển các ứng dụng web hiện đại, và với sự phát triển của Node.js, nó cũng đã mở ra khả năng phát triển ứng dụng server-side và ứng dụng di động.

### TypeScript

TypeScript là một ngôn ngữ lập trình được phát triển và duy trì bởi Microsoft, lần đầu tiên ra mắt vào năm 2012. Được xem như một "superset" của JavaScript, TypeScript không chỉ giữ nguyên toàn bộ các tính năng của JavaScript mà còn mở rộng thêm nhiều tính năng mạnh mẽ, đặc biệt là hệ thống kiểu tĩnh (static typing). Trong JavaScript, kiểu dữ liệu của biến chỉ được xác định khi chạy chương trình, dẫn đến việc khó phát hiện lỗi trong quá trình phát triển, đặc biệt là đối với các dự án lớn. TypeScript giải quyết vấn đề này bằng cách cho phép nhà phát triển xác định kiểu dữ liệu của các biến, hàm, và đối tượng ngay khi viết mã, giúp phát hiện lỗi sớm hơn, cải thiện tính ổn định và bảo mật của ứng dụng.

Ngoài ra, TypeScript còn hỗ trợ các tính năng lập trình hướng đối tượng như classes, interfaces, và inheritance, giúp tổ chức mã nguồn một cách rõ ràng và dễ hiểu hơn. TypeScript cũng tương thích với các phiên bản JavaScript mới nhất, cho phép sử dụng các tính năng hiện đại ngay cả trong các môi trường không hỗ trợ trực tiếp. Khi biên dịch, mã TypeScript sẽ được chuyển đổi thành mã JavaScript thuần, đảm bảo có thể chạy trên bất kỳ trình duyệt hoặc môi trường Node.js nào. Tóm lại, TypeScript đã trở thành một công cụ quan trọng cho các nhà phát triển, đặc biệt trong việc xây dựng các ứng dụng web phức tạp và quy mô lớn.

### ReactJS

ReactJS, thường được gọi đơn giản là React, là một thư viện JavaScript mã nguồn mở do Facebook phát triển và phát hành lần đầu vào năm 2013. React được

thiết kế đặc biệt để xây dựng giao diện người dùng, tập trung vào việc tạo ra các thành phần (components) có thể tái sử dụng. Mỗi component trong React đại diện cho một phần của giao diện người dùng và có thể quản lý trạng thái riêng của mình. Điều này cho phép các nhà phát triển xây dựng các ứng dụng phức tạp nhưng vẫn dễ dàng quản lý và mở rộng.

Một trong những tính năng nổi bật của React là Virtual DOM. Thay vì thao tác trực tiếp với DOM thực (Document Object Model) của trình duyệt, React tạo ra một bản sao ảo của DOM. Khi trạng thái của một component thay đổi, React so sánh DOM ảo với DOM thực, và chỉ cập nhật những phần cần thiết, giúp tăng hiệu suất của ứng dụng một cách đáng kể.

JSX, viết tắt của JavaScript XML, là một cú pháp mở rộng cho phép viết mã HTML trực tiếp trong JavaScript. JSX làm cho mã React trở nên dễ đọc và trực quan hơn, cho phép các nhà phát triển kết hợp mã HTML và JavaScript một cách liền mạch. Ngoài ra, React cũng hỗ trợ mạnh mẽ việc quản lý state và props, giúp dễ dàng kiểm soát luồng dữ liệu trong ứng dụng. React đã trở thành một công cụ không thể thiếu cho việc phát triển ứng dụng web hiện đại, được sử dụng rộng rãi bởi các công ty công nghệ lớn và cộng đồng lập trình viên trên toàn thế giới.

### NestJS

NestJS là một framework Node.js tiến bộ và linh hoạt, được phát triển bởi Kamil Myśliwiec, lần đầu ra mắt vào năm 2017. NestJS được xây dựng trên nền tảng của TypeScript, và lấy cảm hứng từ các framework phổ biến như Angular, với mục tiêu giúp các nhà phát triển xây dựng các ứng dụng server-side mạnh mẽ, dễ bảo trì và mở rộng.

NestJS sử dụng kiến trúc module, cho phép chia ứng dụng thành các phần nhỏ, dễ quản lý và tái sử dụng. Mỗi module trong NestJS chứa các thành phần như controllers, services, và providers, giúp phân tách logic ứng dụng một cách rõ ràng và

có tổ chức. Điều này giúp việc phát triển, bảo trì, và mở rộng ứng dụng trở nên đơn giản và hiệu quả hơn.

NestJS cũng tích hợp sẵn với các công nghệ hiện đại như GraphQL, WebSockets, và Microservices, cho phép xây dựng các ứng dụng có khả năng mở rộng cao và hỗ trợ nhiều loại giao thức truyền thông khác nhau. Dependency Injection là một trong những tính năng quan trọng của NestJS, giúp quản lý các phụ thuộc trong ứng dụng một cách hiệu quả, làm tăng tính linh hoạt và khả năng mở rộng của mã nguồn.

Ngoài ra, NestJS cũng cung cấp hỗ trợ mạnh mẽ cho việc viết unit test và integration test, giúp đảm bảo chất lượng mã nguồn trong quá trình phát triển. Nhờ vào sự kết hợp giữa TypeScript và các kiến trúc lập trình hiện đại, NestJS đã nhanh chóng trở thành một framework phổ biến và được tin dùng trong việc phát triển các ứng dụng backend quy mô lớn.

### MongoDB

MongoDB là một cơ sở dữ liệu NoSQL mã nguồn mở, được phát triển bởi công ty MongoDB Inc. và phát hành lần đầu vào năm 2009. Khác với các cơ sở dữ liệu quan hệ truyền thống như MySQL hoặc PostgreSQL, MongoDB sử dụng mô hình lưu trữ dữ liệu dưới dạng tài liệu (document-based), với mỗi tài liệu được lưu trữ dưới dạng JSON (hoặc BSON - Binary JSON).

MongoDB không yêu cầu cấu trúc bảng cố định, cho phép lưu trữ dữ liệu một cách linh hoạt và dễ dàng thay đổi cấu trúc dữ liệu mà không cần thay đổi toàn bộ cơ sở dữ liệu. Điều này rất hữu ích trong các ứng dụng cần xử lý khối lượng dữ liệu lớn và phức tạp, hoặc trong các hệ thống đòi hỏi khả năng mở rộng nhanh chóng. Ngoài ra, MongoDB hỗ trợ tính năng sharding, giúp phân tán dữ liệu qua nhiều máy chủ, tăng cường khả năng chịu tải và mở rộng của hệ thống.

Một trong những điểm mạnh của MongoDB là khả năng tích hợp tự nhiên với các ngôn ngữ lập trình hiện đại như JavaScript, Python, và Go, cùng với sự hỗ trợ

mạnh mẽ từ cộng đồng. MongoDB cung cấp một API truy vấn linh hoạt và mạnh mẽ, cho phép thực hiện các thao tác phức tạp trên dữ liệu một cách dễ dàng. Cơ sở dữ liệu này thường được sử dụng trong các ứng dụng web hiện đại, đặc biệt là những ứng dụng yêu cầu tính linh hoạt cao trong việc quản lý và xử lý dữ liệu, như các nền tảng thương mại điện tử, mạng xã hội, và các ứng dụng phân tích dữ liệu.

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### Phương pháp phân tích hệ thống

Để thực hiện xây dựng hệ thống quản lý bán đồng hồ trực tuyến với mô hình bài toán đã đề xuất, đề tài sử dụng **phương pháp phân tích hướng đối tượng (Object-Oriented Analysis – OOA)**. Đây là một phương pháp phổ biến trong phát triển phần mềm hiện đại, tập trung vào việc phân tích hệ thống thông qua việc xác định các **đối tượng**, **lớp**, **thuộc tính**, **hành vi**, và **mối quan hệ** giữa chúng trong bối cảnh nghiệp vụ thực tế. Phân tích hướng đối tượng cho phép xây dựng một mô hình hệ thống rõ ràng, dễ bảo trì và phù hợp với tư duy thiết kế phần mềm theo hướng đối tượng.

Kết quả của quá trình phân tích được thể hiện thông qua các **sơ đồ và bảng biểu** sử dụng **ngôn ngữ mô hình hóa UML (Unified Modeling Language)** – một công cụ tiêu chuẩn trong phân tích và thiết kế hệ thống hướng đối tượng. UML được phát triển nhằm hỗ trợ việc mô hình hóa, minh họa, thiết kế và tạo tài liệu cho các hệ thống phần mềm. Ngôn ngữ này cung cấp cách thức trực quan để diễn đạt các khía cạnh quan trọng của hệ thống, bao gồm quy trình nghiệp vụ, cấu trúc dữ liệu và hành vi của các đối tượng.

UML được khởi xướng vào năm 1981 bởi Rational Software, dưới sự lãnh đạo của Paul Levy và Mike Devlin. Đến năm 2005, UML đã được Tổ chức Tiêu chuẩn hóa Quốc tế (ISO) và Ủy ban Kỹ thuật Điện Quốc tế (IEC) công nhận như một tiêu chuẩn toàn cầu với mã hiệu ISO/IEC 19501.

Ngay từ khi ra đời, UML đã được thiết kế để đáp ứng các nhu cầu cốt lõi trong mô hình hóa hệ thống phần mềm. Nó giúp người dùng:

* Phát triển các mô hình với nhiều ý nghĩa và mục đích khác nhau.
* Mở rộng và phát triển chuyên môn thông qua việc mở rộng các khái niệm nền tảng.
* Thúc đẩy các công cụ phát triển phần mềm hướng đối tượng.
* Hỗ trợ các mô hình nâng cao như framework, pattern và component. Các chức năng chính của UML bao gồm:
* Biểu diễn các khái niệm, quy trình, hoặc chức năng của hệ thống một cách trực quan.
* Định nghĩa cấu trúc hệ thống, các thành phần có thể tái sử dụng và cách chúng tương tác.
* Thay thế các ngôn ngữ mô hình hóa cũ như Booch OOAD, OMT, OOSE, nhằm mang lại một cách tiếp cận thống nhất.

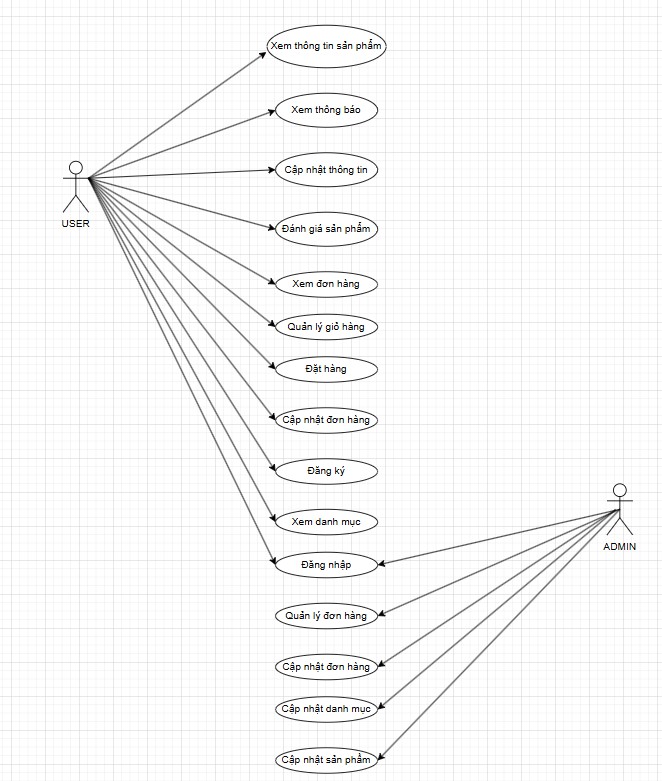
UML cung cấp nhiều loại biểu đồ, mỗi loại phục vụ một mục đích mô hình hóa riêng biệt:

1. **Use Case Diagram**: Minh họa các chức năng của hệ thống từ góc nhìn người dùng.
2. **Class Diagram**: Thể hiện cấu trúc hệ thống thông qua các lớp, thuộc tính, phương thức và mối quan hệ.
3. **Sequence Diagram**: Trình bày thứ tự tương tác giữa các đối tượng trong một kịch bản cụ thể.
4. **State Diagram**: Diễn giải các trạng thái mà một đối tượng có thể trải qua và các điều kiện chuyển trạng thái.
5. **Activity Diagram**: Mô tả luồng công việc hoặc các bước trong một quy trình nghiệp vụ.
6. **Component Diagram**: Thể hiện cấu trúc các thành phần phần mềm và cách chúng liên kết với nhau.
7. **Deployment Diagram**: Mô phỏng cách hệ thống phần mềm được triển khai trên phần cứng thực tế.
8. **Package Diagram**: Quản lý và tổ chức các nhóm lớp hoặc thành phần phần mềm.

Với khả năng mô hình hóa trực quan, linh hoạt và mạnh mẽ, UML không chỉ là công cụ hỗ trợ phân tích mà còn đóng vai trò quan trọng trong toàn bộ quá trình phát triển phần mềm hiện đại. Các nhà phát triển, kiến trúc sư hệ thống và nhóm dự án có thể sử dụng UML để tối ưu hóa quy trình làm việc, nâng cao khả năng hiểu hệ thống và đảm bảo chất lượng phần mềm.

### Phân tích các chức năng của hệ thống

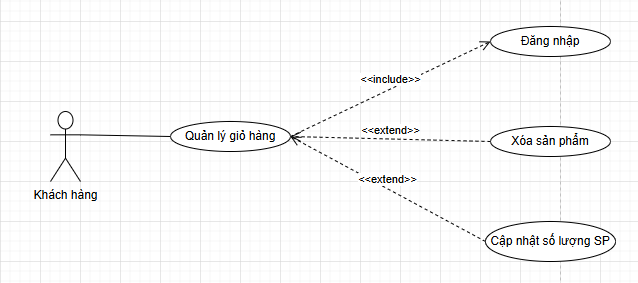
### Biểu đồ Usecase tổng quát

****

*Hình 2.1 Biểu đồ use case tổng quát*

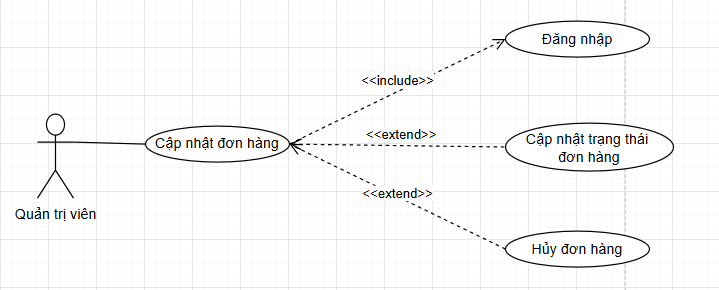
### Các biểu đồ use case phân rã

#### Chức năng quản lý giỏ hàng



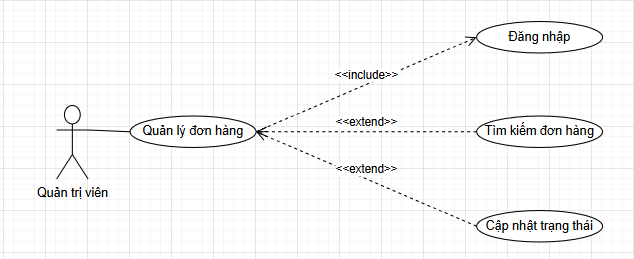
*Hình 2.2 Phân rã chức năng quản lý giỏ hàng*

#### Chức năng cập nhật đơn hàng



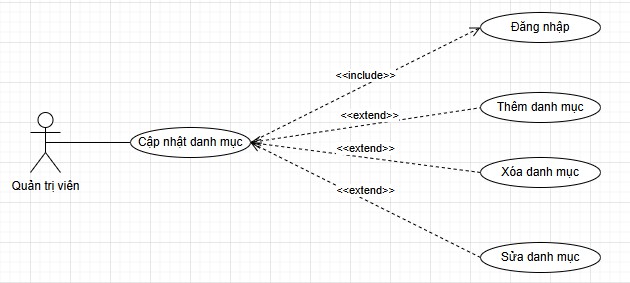
*Hình 2.3. Phân rã chức năng cập nhật đơn hàng*

#### Chức năng quản lý đơn hàng



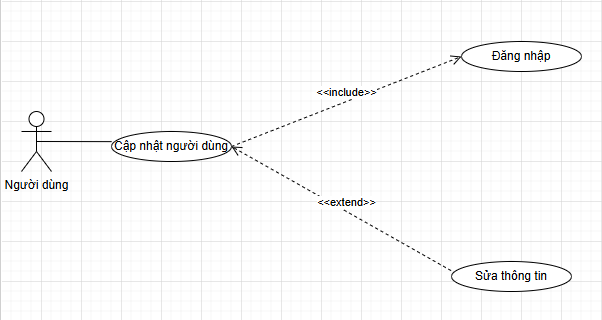
*Hình 2.4. Phân rã chức năng quản lý đơn hàng*

#### Chức năng cập nhật danh mục



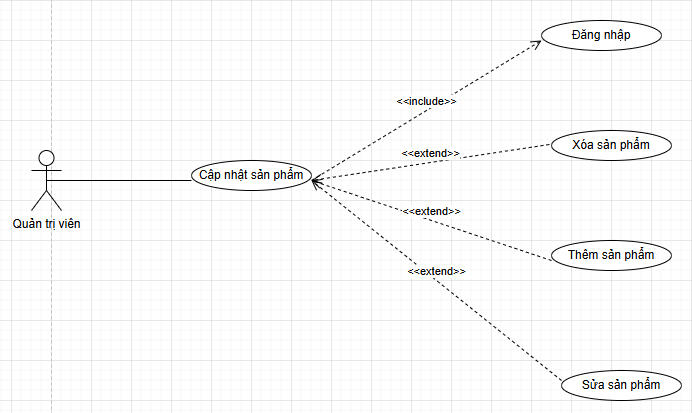
*Hình 2.5. Phân rã chức năng cập nhật danh mục*

#### Chức năng cập nhật người dùng



*Hình 2.6. Phân rã chức năng cập nhật người dùng*

#### Chức năng cập nhật sản phẩm



*Hình 2.7. Chức năng cập nhật sản phẩm*

### Đặc tả use case

* + - 1. Đặc tả use case Đăng nhập tài khoản

*Bảng 2.1 Đặc tả usecase Đăng nhập tài khoản*

|  |  |
| --- | --- |
| **Tên use case** | Đăng nhập tài khoản |
| **Tác nhân** | Người dùng, quản trị viên |
| **Mô tả** | Use case này cho phép người dùng (người dùng & quản trị) đăng nhập vào hệ thống. |
| **Luồng cơ bản** | 1. Hệ thống hiển thị màn hình yêu cầu nhập tên đăng nhập và mật khẩu 2. Người dùng nhập tên đăng nhập và mật khẩu sau đó chọn nút Đăng nhập 3. Hệ thống kiểm tra thông tin đăng nhập và quyền truy cập sau đó hiển thị màn hình trang chủ hệ thống |
| **Luồng rẽ nhánh** | -Tại bước 3 nếu người dùng nhập sai tên hoặc mật khẩu thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại  - Tại bước 2 nếu hệ thống không thể kết nối với cơ sở dữ liệu thì sẽ hiển thị thông báo lỗi |
| **Tiền điều kiện** | Không có |
| **Hậu điều kiện** | Không có |
| **Điểm mở rộng** | Không có |

* + - 1. Đặc tả use case xem thông tin sản phẩm

*Bảng 2.2 Đặc tả usecase xem thông tin sản phẩm*

|  |  |
| --- | --- |
| **Tên use case** | Xem thông tin tài sản phầm |
| **Tác nhân** | Người dùng |
| **Mô tả** | Use case này cho phép người dùng xem thông tin chi tiết của sản phẩm. |
| **Luồng cơ bản** | 1. Khi người dùng chọn vào 1 sản phẩm trên trang chủ, hệ thống sẽ chuyển hướng sang màn hình thông tin chi tiết của  màn hình đó bao gồm tên sản phẩm, hình ảnh, giá tiền, mô tả |
| **Luồng rẽ nhánh** | -Tại tất cả các bước nếu hệ thống không thể kết nối đưuọc với  cơ sở dữ liệu thì sẽ hiển thị một thông báo lỗi |
| **Tiền điều kiện** | Không có |
| **Hậu điều kiện** | Không có |
| **Điểm mở rộng** | Không có |

* + - 1. Đặc tả use case cập nhật thông tin

*Bảng 2.3 Đặc tả usecase cập nhật thông tin*

|  |  |
| --- | --- |
| **Tên use case** | Cập nhật thông tin |
| **Tác nhân** | Người dùng |
| **Mô tả** | Use case này cho phép người dùng xem và cập nhật thông tin  cá nhân. |
| **Luồng cơ bản** | 1. Người dùng chọn vào nút thông tin tài khoản trên thanh tiêu  đề hệ thống. Hệ thống sẽ chuyển sang màn hình thông tin tài khoản và hiển thị lên những thông tin gồm ảnh đại diện, tên |

|  |  |
| --- | --- |
|  | đăng nhập, họ tên, email, số điện thoại, ngày sinh, giới tính tại bảng users  2. Khi người dùng nhập mới các thông tin tài khoản và chọn  nút Lưu thay đổi, hệ thống sẽ cập nhật thông tin thay đổi của người dùng lên cơ sở dữ liệu |
| **Luồng rẽ nhánh** | -Tại bước 2, nếu người dùng nhập email và số điện thoại trùng với 1 người dùng khác trên hệ thống thì hệ thống sẽ báo lỗi và  yêu cầu nhập lại |
| **Tiền điều kiện** | Người dùng phải đăng nhập tài khoản |
| **Hậu điều kiện** | Nếu use case kết thúc thành công thì thông tin người dùng sẽ  được cập nhật |
| **Điểm mở rộng** | Không có |

* + - 1. Đặc tả use case xem đơn hàng

*Bảng 2.4 Đặc tả usecase xem đơn hàng*

|  |  |
| --- | --- |
| **Tên use case** | Xem đơn hàng |
| **Tác nhân** | Người dùng |
| **Mô tả** | Use case này cho phép người dùng xem thông tin đơn hàng |
| **Luồng cơ bản** | 1. Khi người dùng click vào nút đơn hàng tại icon account của người dùng phần trên cùng bên phải của màn hình hệ thống, hệ thống sẽ lấy ra thông tin của các đơn hàng gồm tên sản phẩm, số lượng tổng tiền, trạng thái, ảnh sản phẩm. 2. Khi người dùng click vào một sản phẩm, hệ thống chuyển sang màn hình thông tin chi tiết của sản phẩm đó |
| **Luồng rẽ nhánh** | -Tại tất cả các bước nếu hệ thống không thể kết nối đưuọc với  cơ sở dữ liệu thì sẽ hiển thị một thông báo lỗi |

|  |  |
| --- | --- |
| **Tiền điều kiện** | Người dùng phải đăng nhập tài khoản |
| **Hậu điều kiện** | Không có |
| **Điểm mở rộng** | Không có |

* + - 1. Đặc tả use case quản lý giỏ hàng

*Bảng 2.5 Đặc tả usecase quản lý giỏ hàng*

|  |  |
| --- | --- |
| **Tên use case** | Quản lý giỏ hàng |
| **Tác nhân** | Người dùng |
| **Mô tả** | Use case này cho phép người dùng quản lý giỏ hàng của mình. |
| **Luồng cơ bản** | 1. Khi người dùng chọn vào icon hình giỏ hàng tại phần đầu của màn hình giao diện, hệ thống sẽ chuyển sang màn hình thông tin giỏ hàng của người dùng gồm các sản phẩm, tổng tiền, số lượng 2. Khi người dùng xóa một sản phẩm khỏi giỏ hàng hệ thống sẽ cập nhật lại giỏ hàng 3. Khi người dùng thay đổi số lượng sản phẩm tại giỏ hàng hệ thống sẽ cập nhật lại số lượng của sản phẩm đó |
| **Luồng rẽ nhánh** | -Tại bước 3 luồng cơ bản, người dùng không thể nhập số lượng sản phẩm vượt quá số lượng có trong bảng products, nếu người dùng nhập quá hệ thống sẽ hiển thị số lượng lớn nhất hiện có  - Tại tất cả các bước nếu hệ thống không thể kết nối với cơ sở dữ liệu thì sẽ hiển thị một thông báo lỗi |
| **Tiền điều kiện** | Người dùng phải đăng nhập tài khoản |

|  |  |
| --- | --- |
| **Hậu điều kiện** | Nếu use case kết thúc thành công giỏ hàng sẽ được cập nhật |
| **Điểm mở rộng** | Không có |

* + - 1. Đặc tả use case cập nhật đơn hàng

*Bảng 2.6 Đặc tả usecase cập nhật đơn hàng*

|  |  |
| --- | --- |
| **Tên use case** | Cập nhật đơn hàng |
| **Tác nhân** | Người dùng |
| **Mô tả** | Use case này cho phép người dùng cập nhật đơn hàng |
| **Luồng cơ bản** | 1. Khi người dùng click vào nút đơn hàng trên menu tại tên tài khoản phần đầu màn hình. Hệ thống sẽ chuyển sang màn hình các đơn hàng của người dùng, người dùng click vào một sản phẩm. Hệ thống sẽ chuyển giao diện sang màn hình chi tiết đơn hàng đó, cho phép người dùng hủy đơn hàng 2. Khi người dùng click vào nút hủy đơn hàng. Hệ thống sẽ cho người dùng chọn xác nhận hủy đơn hàng. 3. Nếu người dùng click vào nút xác nhận, hệ thống sẽ thực hiện cập nhật lại trạng thái của đơn hàng là đã hủy. 4. Nếu người dùng hủy click vào nút hủy, hệ thống sẽ hủy bỏ thao tác hủy đơn hàng |
| **Luồng rẽ nhánh** | -Tại tất cả các bước nếu hệ thống không thể kết nối được với  cơ sở dữ liệu thì sẽ hiển thị một thông báo lỗi |
| **Tiền điều kiện** | Người dùng phải đăng nhập tài khoản |
| **Hậu điều kiện** | Không có |
| **Điểm mở rộng** | Không có |

* + - 1. Đặc tả use case đặt hàng

*Bảng 2.7 Đặc tả usecase đặt hàng*

|  |  |
| --- | --- |
| **Tên use case** | Đặt hàng |
| **Tác nhân** | Người dùng |
| **Mô tả** | Use case này cho phép người dùng đặt hàng |
| **Luồng cơ bản** | 1. Khi người dùng click vào nút đặt hàng tại giao diện giỏ hàng, hệ thống sẽ chuyển sang màn hình thanh toán gồm những thông tin sản phẩm, thông tin địa chỉ nhận hàng, số lượng, giá tiền, phương thức thanh toán. 2. Khi người dùng chọn vào nút đặt hàng tại màn hình thanh toán, hệ thống sẽ tạo đơn hàng cho người dùng |
| **Luồng rẽ nhánh** | -Tại bước 2 của luồng cơ bản, nếu người dùng không thể thực hiện thành công với phương thức thanh toán bằng ví điện tử, hệ thống sẽ báo lỗi và yêu cầu người dùng thực hiện lại  - Tại tất cả các bước trong luồng cơ bản nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị một thông báo lỗi |
| **Tiền điều kiện** | Người dùng phải đăng nhập tài khoản |
| **Hậu điều kiện** | Nếu use case kết thúc thành công hệ thống sẽ tạo mới một đơn  hàng |
| **Điểm mở rộng** | Không có |

* + - 1. Đặc tả use case quản lý đơn hàng

*Bảng 2.8 Đặc tả usecase quản lý đơn hàng*

|  |  |
| --- | --- |
| **Tên use case** | Quản lý đơn hàng |
| **Tác nhân** | Quản trị viên |
| **Mô tả** | Use case này cho phép quản trị có thể theo dõi và cập nhật  trạng thái của đơn hàng. |
| **Luồng cơ bản** | 1. Khi quản trị viên chọn nút Quản lý đơn hàng trên thanh menu, hệ thống sẽ hiển thị danh sách các đơn hàng. Người quản trị có thể theo dõi trạng thái và thông tin của các đơn hàng. 2. Khi quản trị viên chọn 1 đơn hàng, hệ thống sẽ hiển thị các thông tin của đơn hàng đó và cho phép quản trị viên cập nhật trạng thái của đơn hàng đó. Nếu quản trị viện chọn 1 trạng thái cho đơn hàng và bấm nút Lưu hệ thống sẽ cập nhật lại trạng   thái đơn hàng. |
| **Luồng rẽ nhánh** | - Tại tất cả các bước trong luồng cơ bản nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị một thông báo  lỗi |
| **Tiền điều kiện** | Quản trị viên phải đăng nhập tài khoản với quyền quản trị |
| **Hậu điều kiện** | Nếu use case kết thúc thành công hệ thống cập nhật trạng thái  đơn hàng |
| **Điểm mở rộng** | Không có |

* + - 1. Đặc tả use case cập nhật sản phẩm

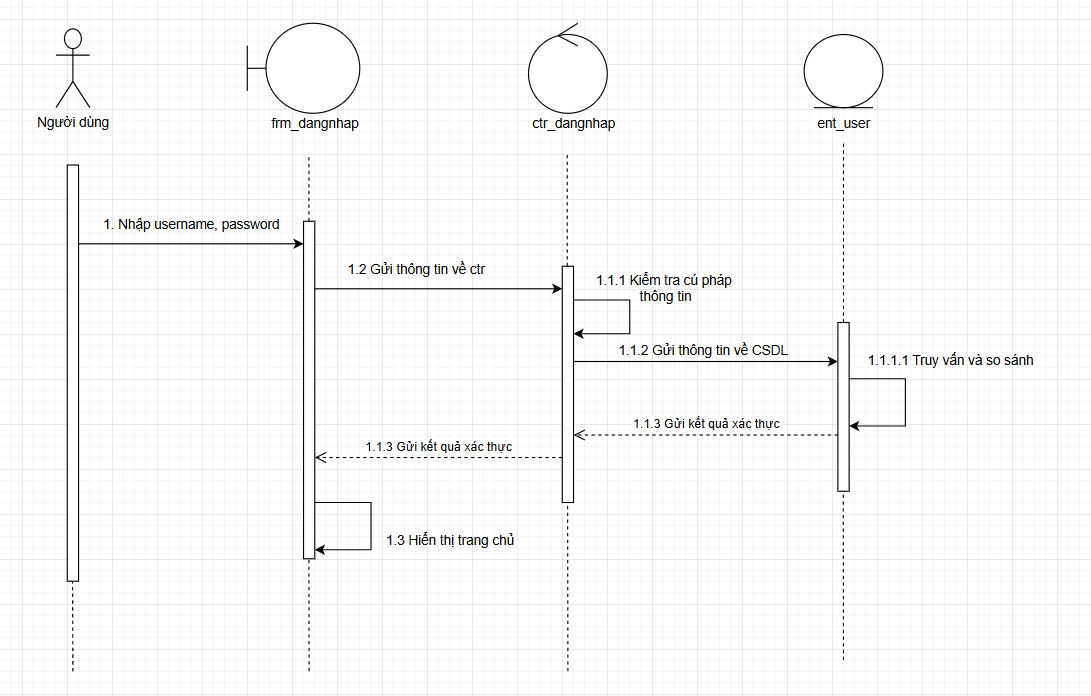
*Bảng 2.9 Đặc tả usecase cập nhật sản phẩm*

|  |  |
| --- | --- |
| **Tên use case** | Cập nhật sản phẩm |
| **Tác nhân** | Quản trị viên |
| **Mô tả** | Use case này cho phép quản trị có thể thêm, sửa, xóa các sản  phẩm. |
| **Luồng cơ bản** | 1. Khi người dùng chọn vào nút Sản phẩm trên thanh menu hệ thống quản trị, hệ thống sẽ hiển thị danh sách các sản phẩm.  2 Thêm sản phẩm: Khi quản trị viên nhập thông tin chi tiết về sản phẩm gồm: danh mục, tiêu đề, giá tiền, giảm giá, mô tả, phân loại vào 1 form bên dưới danh sách và chọn nút Thêm. Hệ thống sẽ kiểm tra dữ liệu nhập vào và thêm sản phẩm mới vào danh sách   1. Sửa sản phẩm: Khi quản trị viên chọn vào nút Sửa trên một dòng thông tin về sản phẩm, hệ thống sẽ hiển thị các trường cho phép quản trị viên sửa thông tin của sản phẩm. Sau khi quản trị viên thay đổi thông tin của sản phẩm và chọn nút Lưu, hệ thống sẽ kiểm tra thông tin thay đổi và cập nhật sản phẩm đó. 2. Xóa sản phẩm: Khi người quản trị chọn nút Xóa trên 1 dòng   thông tin sản phẩm. Hệ thống sẽ xóa sản phẩm khỏi danh sách và cập nhật lại danh sách sản phẩm |
| **Luồng rẽ nhánh** | -Tại bước 2 hoặc bước 3 trong luồng cơ bản. Khi thông tin về sản phẩm nhập không đủ thì hệ thống sẽ thông báo lỗi và yêu cầu nhập bổ sung. Người quản trị nhập bổ sung để tiếp tục hoặc  chọn nút Bỏ qua và use case kết thúc. |

|  |  |
| --- | --- |
|  | -Trong quá trình thực hiện use case nếu hệ thống không thể kết nối được với cơ sở dữ liệu thì sẽ hiển thị một thông báo lỗi và  use case kết thúc. |
| **Tiền điều kiện** | Quản trị viên phải đăng nhập tài khoản với quyền quản trị |
| **Hậu điều kiện** | Nếu use case kết thúc thành công thì thông tin về sản phẩm sẽ  được cập nhật |
| **Điểm mở rộng** | Không có |

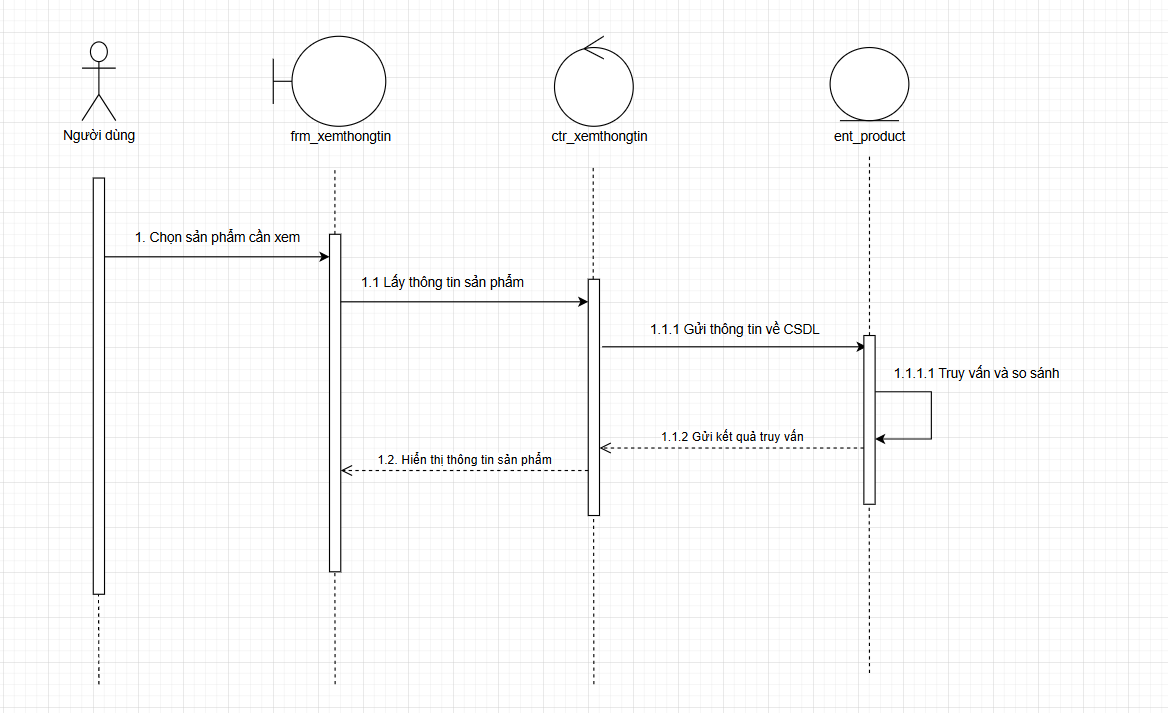
### Phân tích hành vi hệ thống

* + 1. **Use case Đăng nhập Biểu đồ trình tự**



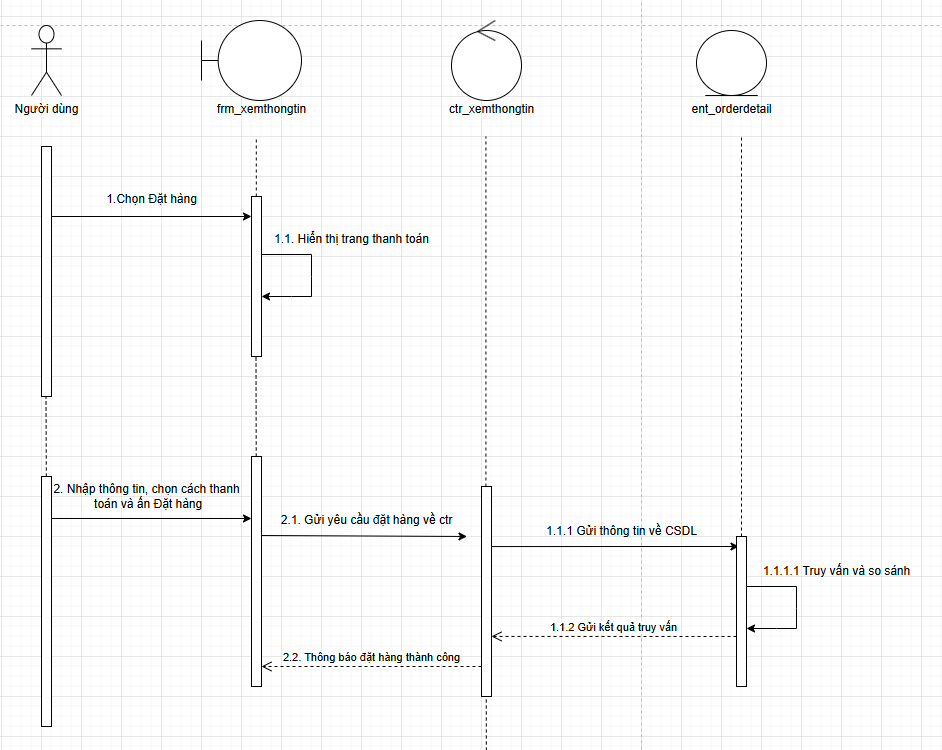
*Hình 2.8. Biểu đồ trình tự use case Đăng nhập*

### Use case Xem thông tin sản phẩm Biểu đồ trình tự



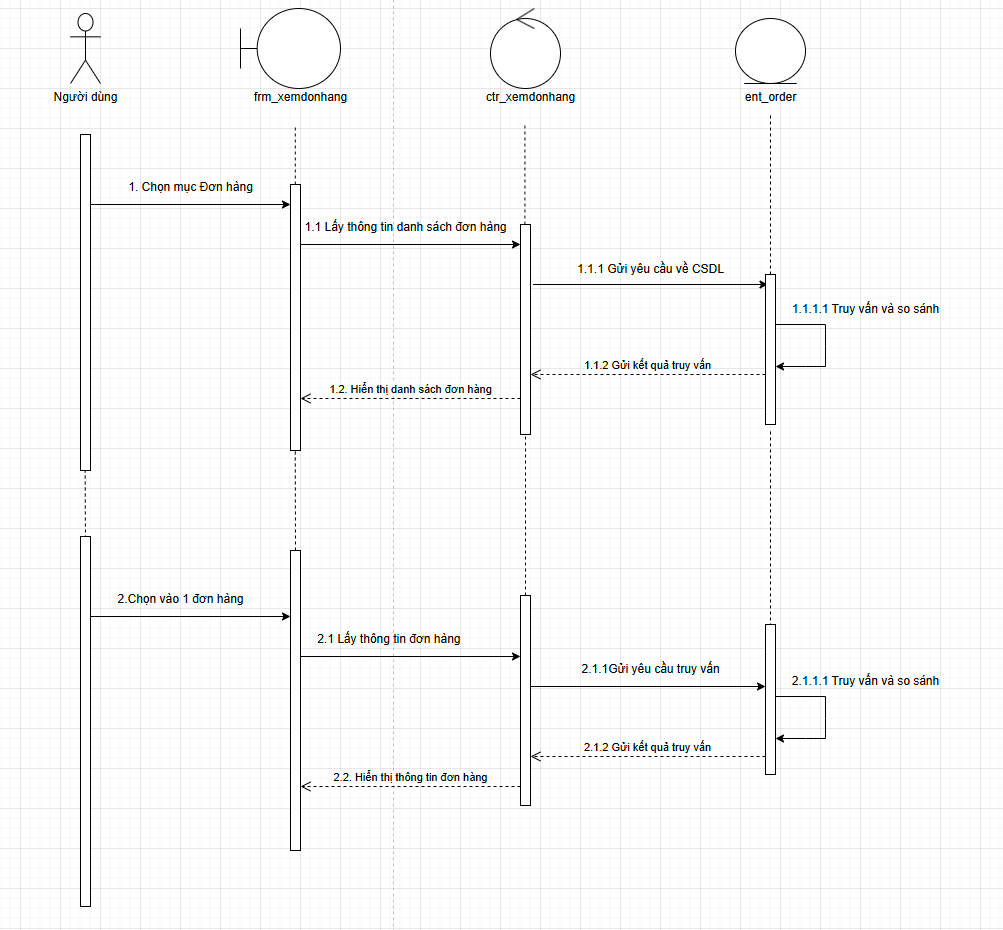
*Hình 2.9. Biểu đồ trình tự use case Xem thông tin sản phẩm*

### Use case Đặt hàng Biểu đồ trình tự



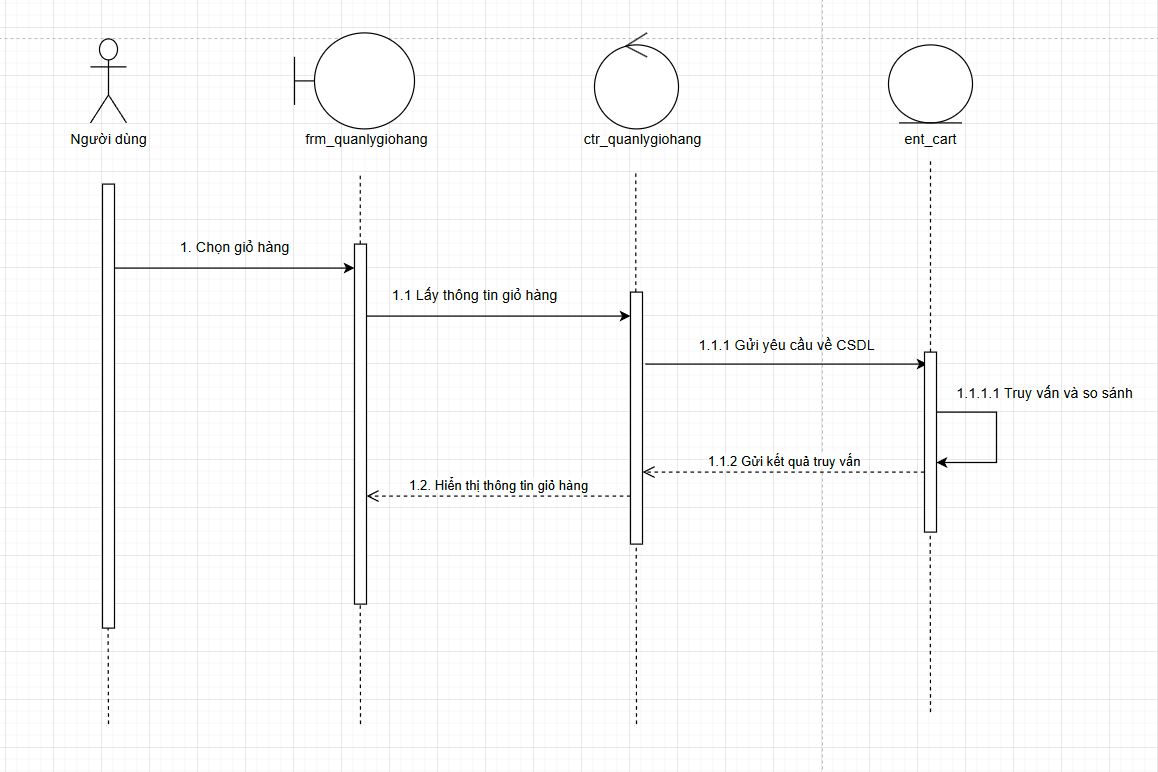
*Hình 2.10. Biểu đồ trình tự use case Đặt hàng*

### Use case Xem đơn hàng Biểu đồ trình tự



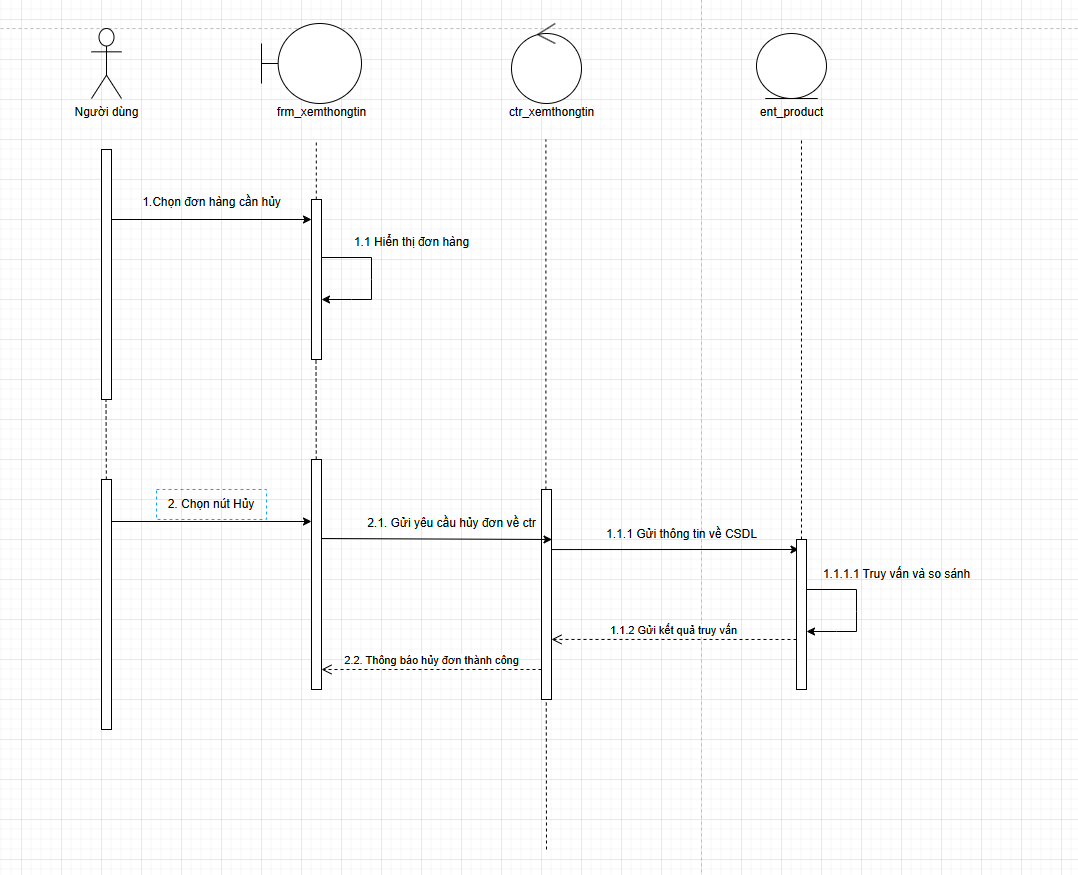
*Hình 2.11. Biểu đồ trình tự use case Xem đơn hàng*

### Use case Quản lý giỏ hàng Biểu đồ trình tự



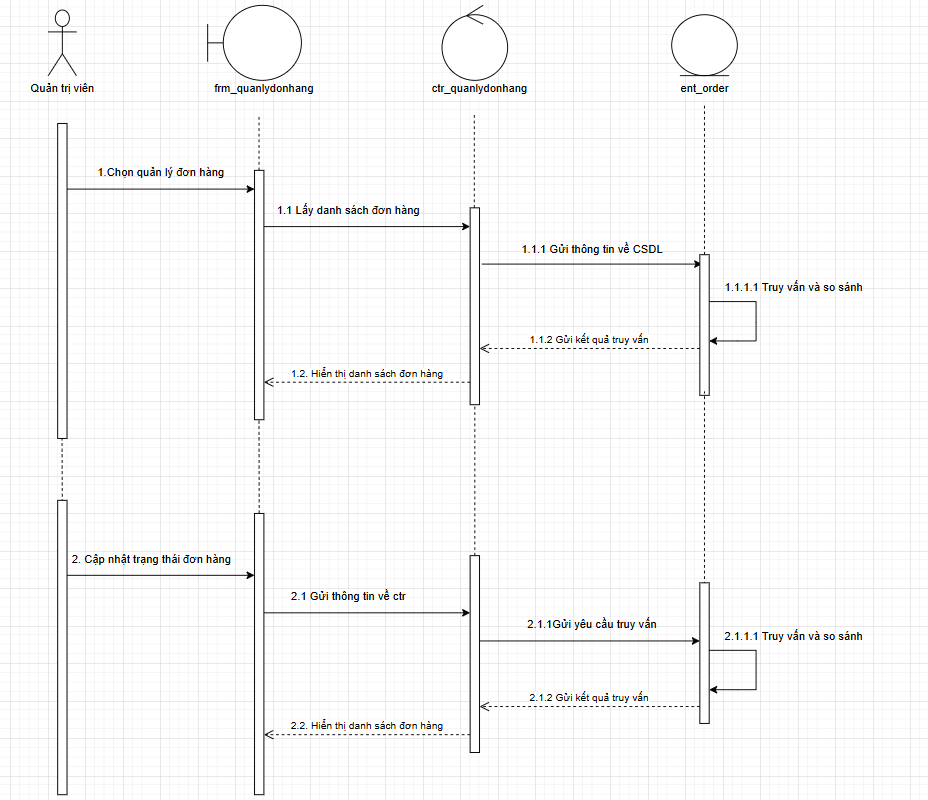
*Hình 2.12. Biểu đồ trình tự Quản lý giỏ hàng*

### Use case Hủy đơn hàng Biểu đồ trình tự



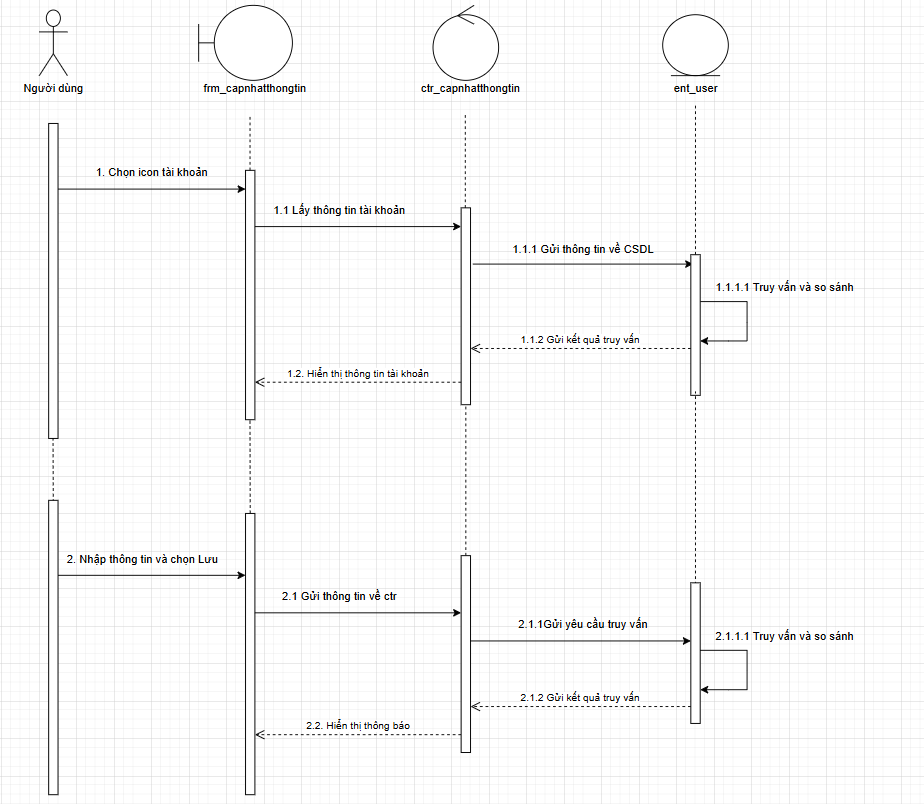
*Hình 2.13 Biểu đồ trình tự use case Hủy đơn hàng*

### Use case Quản lý đơn hàng Biểu đồ trình tự



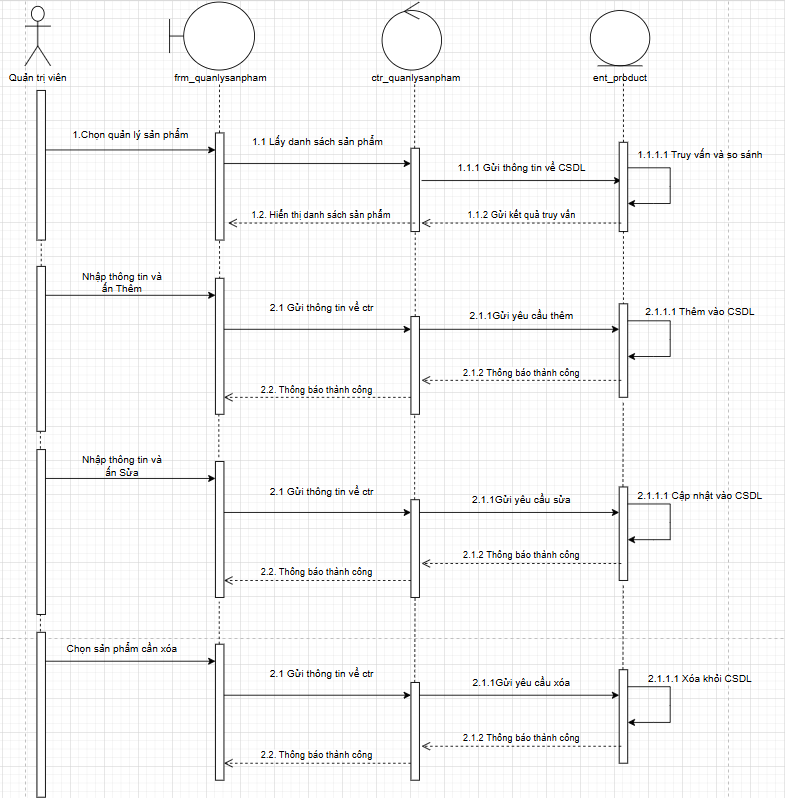
*Hình 2.14 Biểu đồ trình tự use case Quản lý đơn hàng*

### Use case Cập nhật thông tin Biểu đồ trình tự



*Hình 2.15 Biểu đồ trình tự use case Cập nhật thông tin*

### Use case Cập nhật sản phẩm Biểu đồ trình tự



*Hình 2.16 Biểu đồ trình tự usecase Cập nhật sản phẩm*

### Phân tích cấu trúc hệ thống

### Mô tả các lớp

1. **User (Người dùng)**

Lớp User đại diện cho tất cả người dùng trong hệ thống, bao gồm cả khách hàng và quản trị viên. Người dùng có thể thực hiện các hành động như đặt hàng, viết đánh giá sản phẩm và quản lý giỏ hàng. Vai trò (role) của người dùng sẽ xác định quyền truy cập và chức năng họ có thể sử dụng trên hệ thống.

### Product (Sản phẩm)

Lớp Product mô tả các sản phẩm được bán trên website, cụ thể là các mẫu đồng hồ. Mỗi sản phẩm có thể được phân loại theo kiểu dáng, giới tính, và chất liệu. Sản phẩm có thể được đánh giá bởi người dùng và được thêm vào đơn hàng hoặc giỏ hàng.

### Menu (Menu điều hướng)

Lớp Menu đại diện cho các mục điều hướng chính trong giao diện người

dùng.

### Category (Danh mục)

Lớp Category dùng để phân loại sản phẩm theo nhóm cụ thể hơn, chẳng hạn như “Đồng hồ dây da”, “Đồng hồ thể thao”.

### Type (Loại sản phẩm)

Lớp Type cung cấp một mức phân loại sâu hơn bên trong danh mục. Mỗi loại sản phẩm đại diện cho một nhóm cụ thể dựa trên kiểu dáng hoặc chức năng, ví dụ như “Automatic”, “Quartz”, hoặc “Chronograph”.

### Order (Đơn hàng)

Lớp Order lưu thông tin về các giao dịch mua hàng do người dùng thực hiện.

Một đơn hàng chứa nhiều sản phẩm, có trạng thái xử lý (như đang xử lý, đã giao, hủy), và liên kết đến người dùng tạo đơn.

### OrderItem (Mục đơn hàng)

Lớp OrderItem đại diện cho từng sản phẩm cụ thể nằm trong một đơn hàng.

Nó giúp ghi lại thông tin chi tiết như sản phẩm nào được mua, số lượng bao nhiêu và đơn giá tại thời điểm mua hàng

### Cart (Giỏ hàng)

Lớp Cart là nơi lưu tạm thời các sản phẩm mà người dùng muốn mua. Mỗi người dùng chỉ có một giỏ hàng tại một thời điểm. Giỏ hàng hỗ trợ thêm, sửa, xóa sản phẩm trước khi tiến hành đặt hàng.

### CartItem (Mục trong giỏ hàng)

Lớp CartItem đại diện cho một dòng sản phẩm cụ thể trong giỏ hàng của người dùng. Nó cho biết người dùng đã thêm sản phẩm nào và với số lượng bao nhiêu.

### Review (Đánh giá)

Lớp Review lưu trữ các đánh giá mà người dùng đã viết cho sản phẩm. Mỗi đánh giá bao gồm xếp hạng sao và bình luận. Các đánh giá giúp cung cấp phản hồi và tăng độ tin cậy cho sản phẩm đối với khách hàng khác.

### Mối quan hệ giữa các lớp

Trong hệ thống, mỗi người dùng (**User**) có thể thực hiện nhiều hoạt động, bao gồm đặt hàng, đánh giá sản phẩm, và sử dụng giỏ hàng. Cụ thể, một người dùng có thể tạo ra nhiều đơn hàng (**Order**) khác nhau theo thời gian.

Đồng thời, người dùng cũng có thể viết nhiều đánh giá (**Review**) cho các sản phẩm mà họ đã mua. Mỗi người dùng được gắn liền với một giỏ hàng (**Cart**) duy nhất, trong đó chứa các sản phẩm mà họ đang quan tâm.

Một đơn hàng (**Order**) được tạo ra bởi một người dùng sẽ bao gồm nhiều mục đơn hàng (**OrderItem**), tương ứng với từng sản phẩm cụ thể được đặt. Mỗi mục đơn hàng này liên kết tới một sản phẩm (**Product**) cụ thể, đi kèm số lượng và đơn giá tại thời điểm đặt hàng.

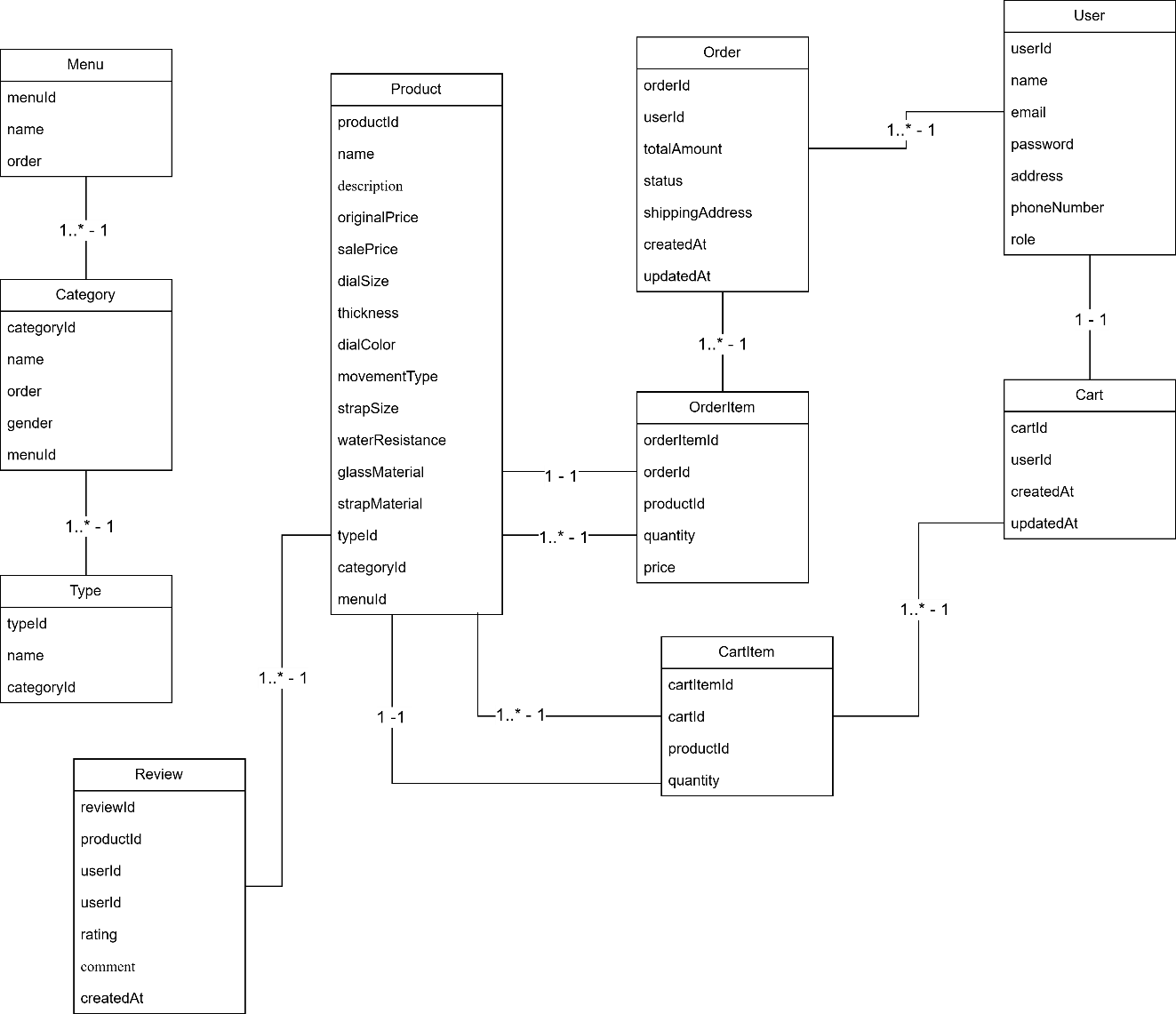
Tương tự, giỏ hàng (**Cart**) của người dùng cũng chứa nhiều mục trong giỏ hàng (**CartItem**). Mỗi mục giỏ hàng đại diện cho một sản phẩm người dùng thêm vào, và cũng liên kết đến đối tượng sản phẩm cụ thể.

Mỗi sản phẩm (**Product**) có thể nhận được nhiều đánh giá (**Review**) từ các người dùng khác nhau. Một sản phẩm cũng có thể xuất hiện trong nhiều đơn hàng và giỏ hàng, thông qua các đối tượng **OrderItem** và **CartItem**.

Về mặt phân loại, các sản phẩm được tổ chức theo cấu trúc phân cấp bắt đầu từ Menu. Mỗi menu có thể chứa nhiều danh mục (**Category**). Mỗi danh mục lại bao gồm nhiều loại sản phẩm (**Type**), và một loại sản phẩm sẽ có thể chứa nhiều sản phẩm cụ thể.

### Biểu đồ lớp

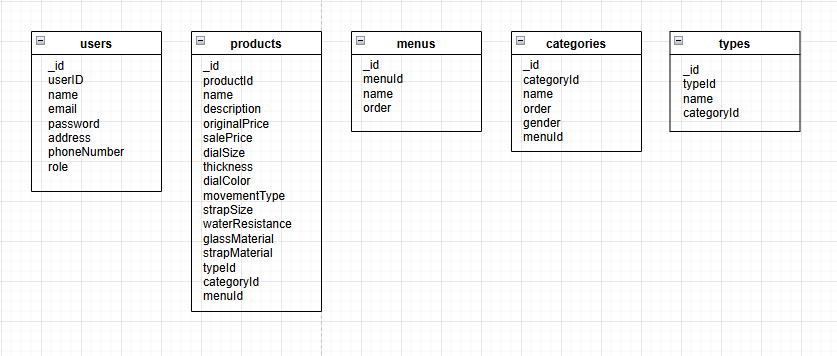
Từ các lớp đã phân tích ở trên ta có biểu đồ lớp như sau:



*Hình 2.17 Biểu đồ lớp*

### Thiết kế cơ sở dữ liệu

### Mô hình hóa dữ liệu

****

*Hình 2.18 Các bảng trong CSDL*

### Thiết kế collection

1. **User**
   * **userId:** ID duy nhất của người dùng (Primary Key).
   * **name:** Tên người dùng.
   * **email:** Địa chỉ email của người dùng.
   * **password:** Mật khẩu để đăng nhập.
   * **address:** Địa chỉ người dùng.
   * **phoneNumber:** Số điện thoại của người dùng.
   * **role:** Vai trò của người dùng (admin hoặc customer).

### Product

* + **productId:** ID duy nhất của sản phẩm (Primary Key).
  + **name:** Tên của sản phẩm.
  + **description:** Mô tả chi tiết về sản phẩm.
  + **originalPrice:** Giá gốc của sản phẩm.
  + **salePrice:** Giá khuyến mãi của sản phẩm.
  + **dialSize:** Kích thước mặt đồng hồ.
  + **thickness:** Độ dày của đồng hồ.
  + **dialColor:** Màu sắc mặt đồng hồ.
  + **movementType:** Loại bộ máy đồng hồ (ví dụ: Automatic).
  + **strapSize:** Kích thước dây đeo.
  + **waterResistance:** Khả năng chống nước của đồng hồ.
  + **glassMaterial:** Chất liệu mặt kính.
  + **strapMaterial:** Chất liệu dây đeo.
  + **typeId:** ID của loại sản phẩm (Foreign Key).
  + **categoryId:** ID của danh mục sản phẩm (Foreign Key).
  + **menuId:** ID của menu chứa sản phẩm (Foreign Key).

### Menu

* + **menuId:** ID duy nhất của menu (Primary Key).
  + **name:** Tên menu.
  + **order:** Thứ tự hiển thị của menu.

### Category

* + **categoryId:** ID duy nhất của danh mục (Primary Key).
  + **name:** Tên danh mục.
  + **order:** Thứ tự hiển thị của danh mục.
  + **gender:** Giới tính liên quan đến danh mục (ví dụ: Nam, Nữ).
  + **menuId:** ID của menu chứa danh mục (Foreign Key).

### Type

* + **typeId:** ID duy nhất của loại sản phẩm (Primary Key).
  + **name:** Tên loại sản phẩm.
  + **categoryId:** ID của danh mục chứa loại sản phẩm (Foreign Key).

### Order

* + **orderId:** ID duy nhất của đơn hàng (Primary Key).
  + **userId:** ID của người dùng đã đặt đơn hàng (Foreign Key).
  + **totalAmount:** Tổng số tiền của đơn hàng.
  + **status:** Trạng thái của đơn hàng (ví dụ: Đang xử lý, Đã giao hàng).
  + **shippingAddress:** Địa chỉ giao hàng.
  + **createdAt:** Ngày tạo đơn hàng.
  + **updatedAt:** Ngày cập nhật đơn hàng gần nhất.

### OrderItem

* + **orderItemId**: ID duy nhất của mục đơn hàng (Primary Key).
  + **orderId:** ID của đơn hàng (Foreign Key).
  + **productId:** ID của sản phẩm trong đơn hàng (Foreign Key).
  + **quantity:** Số lượng sản phẩm.
  + **price:** Giá của sản phẩm.

### Cart

* + **cartId:** ID duy nhất của giỏ hàng (Primary Key).
  + **userId:** ID của người dùng sở hữu giỏ hàng (Foreign Key).
  + **createdAt:** Ngày tạo giỏ hàng.
  + **updatedAt:** Ngày cập nhật giỏ hàng gần nhất**.**

### CartItem

* + **cartItemId:** ID duy nhất của mục trong giỏ hàng (Primary Key).
  + **cartId:** ID của giỏ hàng (Foreign Key).
  + **productId:** ID của sản phẩm trong giỏ hàng (Foreign Key).
  + **quantity:** Số lượng sản phẩm trong giỏ hàng.

### Review

* + **reviewId:** ID duy nhất của đánh giá (Primary Key).
  + **productId:** ID của sản phẩm được đánh giá (Foreign Key).
  + **userId:** ID của người dùng đã viết đánh giá (Foreign Key).
  + **rating:** Xếp hạng của sản phẩm (theo số sao, ví dụ: 1-5).
  + **comment:** Nhận xét của người dùng.
  + **createdAt:** Ngày tạo đánh giá.

### Quan hệ giữa các collection

* **User - Order**: Một User có thể có nhiều Order.
* **Order - OrderItem**: Một Order có thể chứa nhiều OrderItem.
* **OrderItem - Product**: Một OrderItem thuộc về một Product.
* **Product - Review**: Một Product có thể có nhiều Review.
* **User - Review**: Một User có thể viết nhiều Review.
* **Menu - Category**: Một Menu có thể chứa nhiều Category.
* **Category - Type**: Một Category có thể chứa nhiều Type.
* **Type - Product**: Một Type có thể chứa nhiều Product.
* **Cart - CartItem**: Một Cart có thể chứa nhiều CartItem.
* **CartItem - Product**: Một CartItem thuộc về một Product.

### Thiết kế mô hình gợi ý

### Khái niệm

Hệ thống gợi ý (Recommender System) là một trong những ứng dụng nổi bật của khoa học dữ liệu ngày nay, được sử dụng để dự đoán "điểm số" hoặc "sở thích" mà người dùng có thể dành cho một sản phẩm hoặc dịch vụ cụ thể. Hầu hết các công ty công nghệ lớn đều tích hợp hệ thống này dưới nhiều hình thức khác nhau:

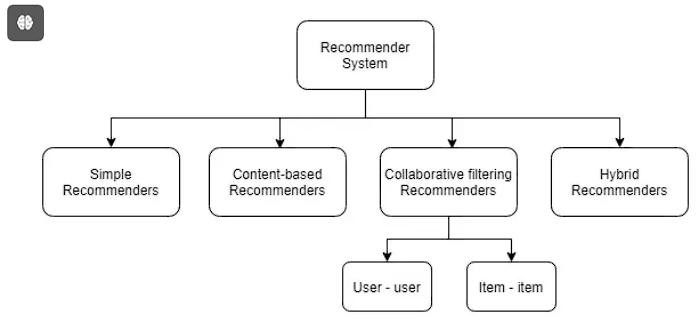
* **Amazon** dùng để đề xuất sản phẩm phù hợp cho khách hàng.
* **Netflix** giúp gợi ý các bộ phim dựa trên sở thích người dùng.
* **YouTube** tự động chọn video kế tiếp dựa trên lịch sử xem.
* **Facebook** đề xuất bạn bè, trang yêu thích hoặc tài khoản theo dõi.

### Hướng tiếp cận

Một hệ thống gợi ý sản phẩm thường được chia thành ba quá trình chính:

* **Tìm đặc trưng ảnh hưởng đến đánh giá người dùng**: Qua quá trình phân tích dữ liệu và khám phá các yếu tố liên quan, hệ thống sẽ xác định những đặc trưng quan trọng có ảnh hưởng đến quyết định của người dùng.
* **Áp dụng giải thuật lọc**: Dựa trên các đặc trưng đã xác định, hệ thống sẽ chọn giải thuật lọc phù hợp để đưa ra gợi ý chính xác nhất.
* **Huấn luyện mô hình**: Tiến hành huấn luyện hệ thống dựa trên dữ liệu để tối ưu hóa khả năng gợi ý.

Có bốn loại hệ thống gợi ý phổ biến:



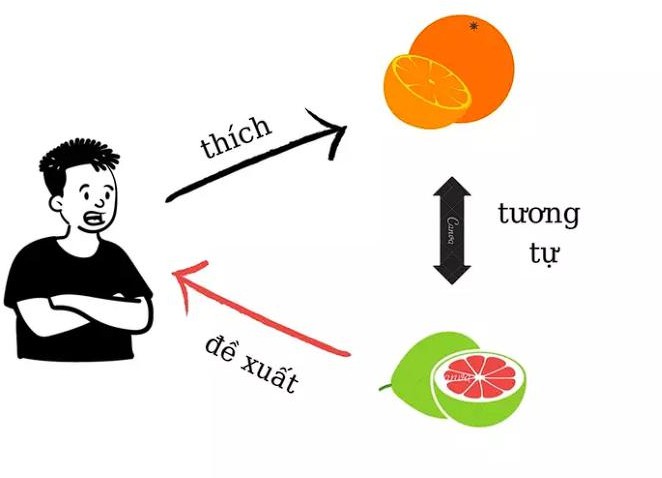
*Hình 2.19. Mô hình các hệ thống gợi ý phổ biến*

* **Simple Recommenders**: Cung cấp đề xuất dựa trên tính phổ biến của sản phẩm hoặc dịch vụ mà không phân biệt người dùng. Ví dụ: Bảng xếp hạng phim IMDB Top 250.
* **Content-based Recommenders**: Gợi ý các sản phẩm có thuộc tính tương tự dựa trên những gì người dùng đã chọn trước đó. Ví dụ: YouTube gợi ý video dựa trên lịch sử xem của người dùng.
* **Collaborative Filtering Recommenders**: Dự đoán sở thích của người dùng dựa trên hành vi của những người dùng khác có cùng sở thích. Đây là phương pháp phổ biến vì nó không cần dựa vào thông tin cụ thể của sản phẩm.
* **Hybrid Recommenders**: Kết hợp cả hai phương pháp lọc dựa trên nội dung và lọc cộng tác để đưa ra gợi ý linh hoạt hơn.

### Xây dựng Content-based Filtering RS (Recommender System)

* + - 1. Thuật toán

Đối với phương pháp content-based, hệ thống sẽ đánh giá các đặc tính của items được recommended. Nó sẽ gợi ý các item dựa trên hồ sơ (profiles) của người dùng hoặc dựa vào nội dung, thuộc tính (attributes) của những item tương tự như item mà người dùng đã chọn trong quá khứ. Ví dụ: một người rất thích ăn cam, vậy thì hệ thống gợi ý một loại trái cây tương tự với cam, ở đây là bưởi để đề xuất. Cách tiếp cận này yêu cầu việc sắp xếp các items vào từng nhóm hoặc đi tìm các đặc trưng của từng item.



*Hình 2.20. Ví dụ về điểm tương đồng trong thực tế*

### Áp dụng vào dự án:

****

*Hình 2.21. Ví dụ về điểm tương đồng của 2 sản phẩm trong database*

Khi người dùng xem sản phẩm Rise, hệ thống sẽ đề xuất sản phẩm 2 Mile, cùng kiểu với sản phẩm người dùng thích. Chính vì vậy, hệ thống chỉ cần biết người dùng xem sản phẩm nào chứ không cần dữ liệu về ratings, giúp nó hoạt động ngay cả khi người dùng không có thói quen đánh giá sản phẩm

* + - 1. Cơ chế hoạt động
         * Tạo ra các đặc điểm (features) mô tả sản phẩm.
         * Xác định các sản phẩm tương tự dựa trên các đặc điểm này.

*Bảng 2.10. Ưu và nhược điểm của gợi ý theo nội dung*

|  |  |
| --- | --- |
| **Lợi ích** | **Nhược điểm** |
| Không cần thông tin về hành vi người  dùng | Không thể gợi ý sản phẩm mà người  dùng chưa biết hoặc chưa từng xem |

|  |  |
| --- | --- |
| Gợi ý chính xác cho những người dùng  có sở thích cụ thể | Có thể dẫn đến việc gợi ý sản phẩm quá  giống nhau |

* + - 1. Phương pháp TF-IDF

Ma Trận TF-IDF (Term Frequency-Inverse Document Frequency) : là một kỹ thuật phổ biến trong xử lý ngôn ngữ tự nhiên (NLP) để đánh giá tầm quan trọng của từ trong một tài liệu so với toàn bộ tập tài liệu.

Công thức tính:

* Tern Frequency (TF): Đếm số lần từ xuất hiện trong tài liệu. Nó phản ánh sự phổ biến của từ trong tài liệu đó.

𝑇𝐹(𝑡, 𝑑) =

Số lần từ t xuất hiện trong tài liệu d Tổng số từ trong tài liệu d

Inverse Document Frequency (IDF): Đo lường tầm quan trọng của từ trong toàn bộ tập tài liệu. Từ xuất hiện ít trong các tài liệu khác thì có giá trị IDF cao hơn.

Tổng số tài liệu|𝐷|

𝐼𝐷𝐹(𝑡, 𝐷) = log (

)

Số tài liệu chứa từ t

* TF-IDF Score: Tính toán tầm quan trọng của từ trong tài liệu

TF-IDF(*t,d,D*) = TF(*t,d*) × IDF(*t,D*)

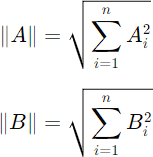
* + - 1. Tính góc tương đồng (Cosine Similaryty)

Phương pháp tính góc tương đồng Cosine là một phương pháp phổ biến để tính độ tương đồng giữa hai vector mà không phụ thuộc vào độ dài của hai vector. Do đó ta lựa chọn sử dụng phương pháp này kết hơp cùng các kỹ thuật trên để tìm độ tương đồng giữa 2 vector đại diện cho đặc trưng mới của mỗi sản phẩm giúp tìm ra được các sản phẩm có tính tương đồng nhau.

Công thức:

Trong đó:

A. B

𝐶𝑜𝑠𝑖𝑛𝑒 𝑆𝑖𝑚𝑖𝑙𝑎𝑟𝑖𝑡𝑦 = ‖𝐴‖‖𝐵‖

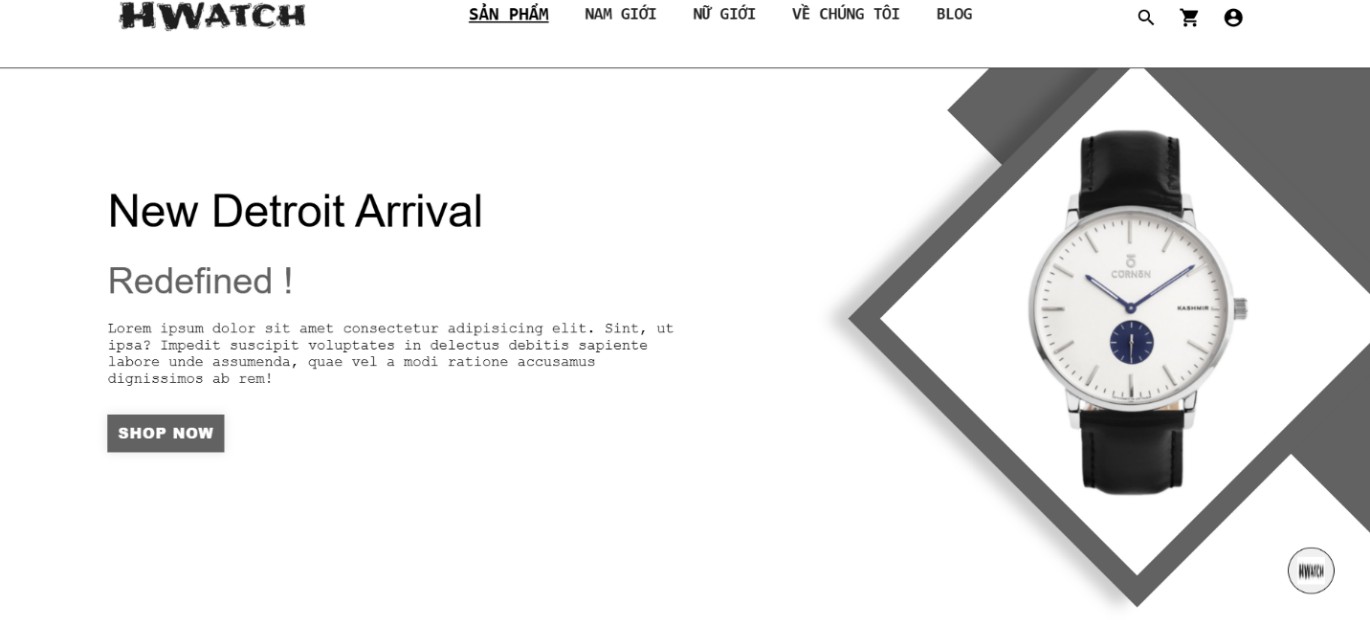


# CHƯƠNG 3. GIỚI THIỆU VỀ WEBSITE

### Giới thiệu tính năng bên phía khách hàng

### Trang chủ

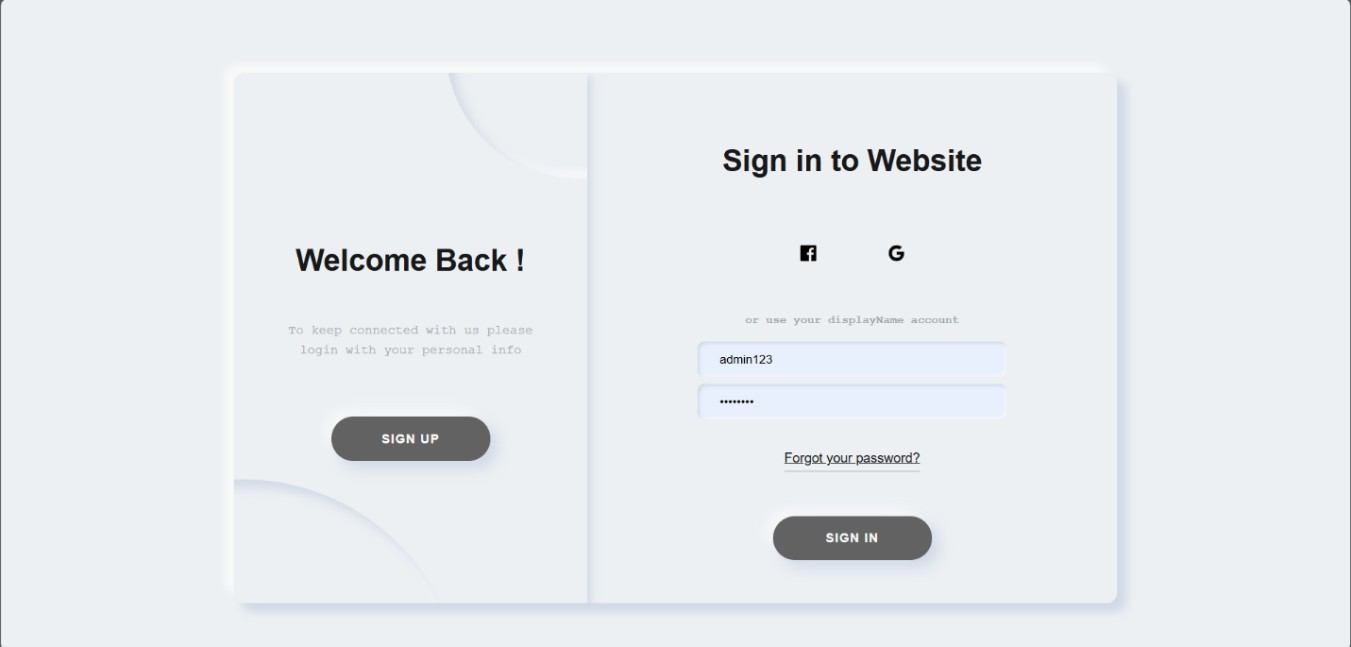
Trang chủ website hiển thị giới thiệu về website



*Hình 3.1 Giao diện trang chủ website*

### Đăng nhập

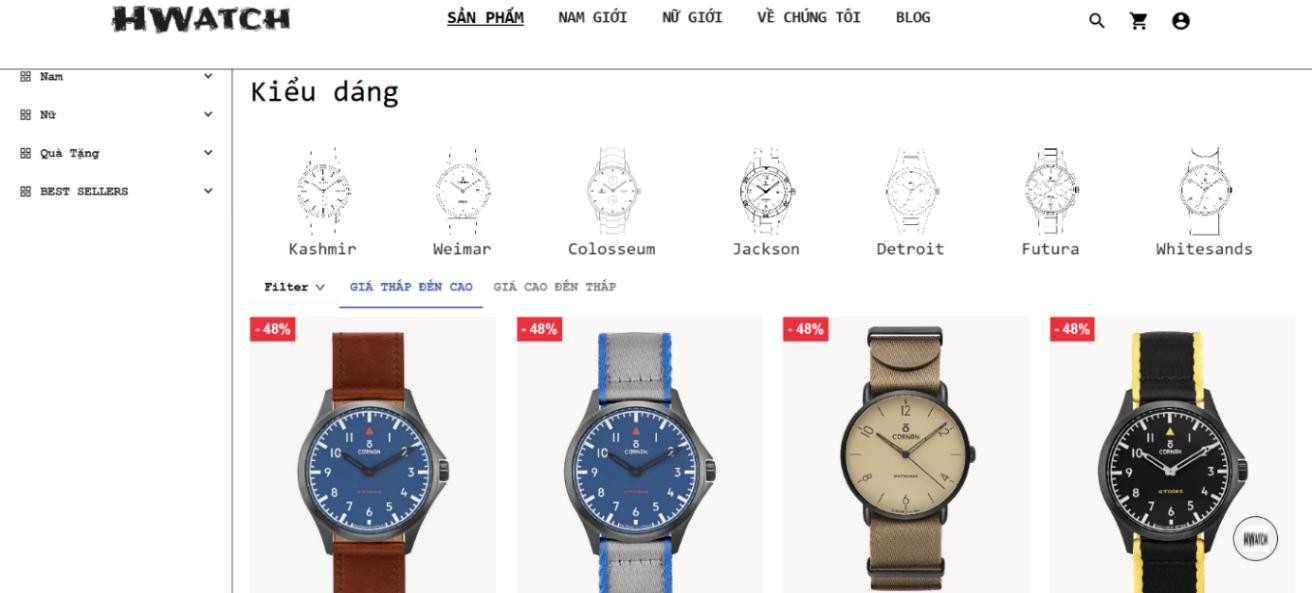
Khách hàng đã có tài khoản sẽ đăng nhập website bằng tên đăng nhập và mật khẩu đã đăng ký. Nếu người dùng chưa có tài khoản có thể chọn chuyển sang trang Đăng ký để đăng ký tài khoản.



*Hình 3.2 Giao diện đăng nhập*

### Sản phẩm

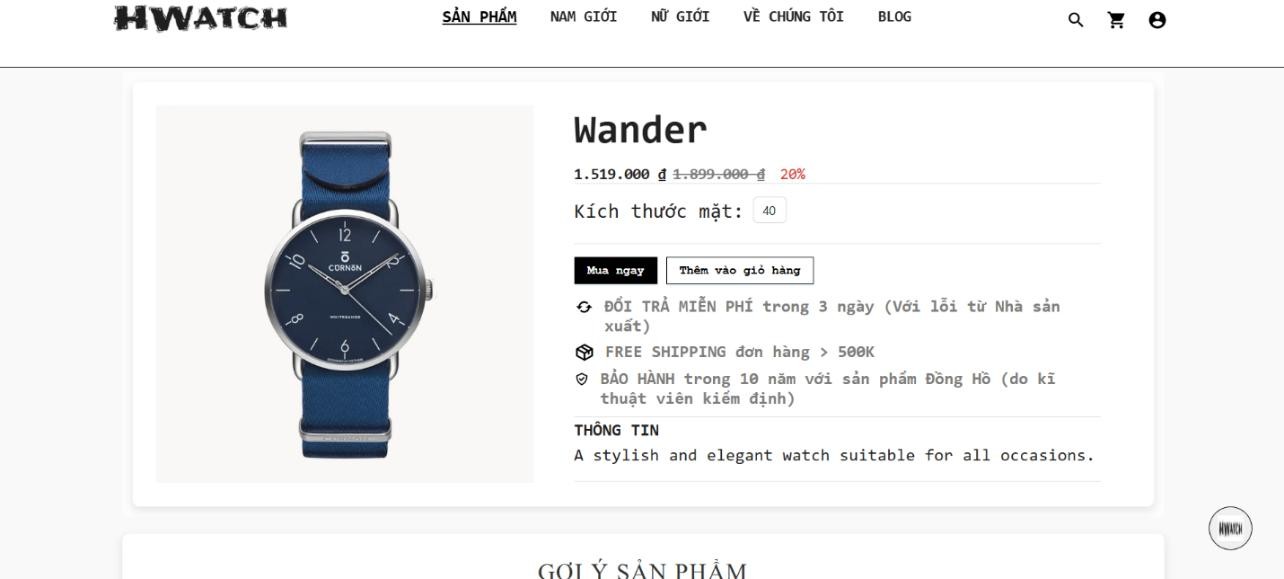
Trang sản phẩm hiển thị tất cả các sản phẩm để người dùng có thể tham khảo và lựa chọn



*Hình 3.3 Giao diện danh sách sản phẩm*

### Chi tiết sản phẩm

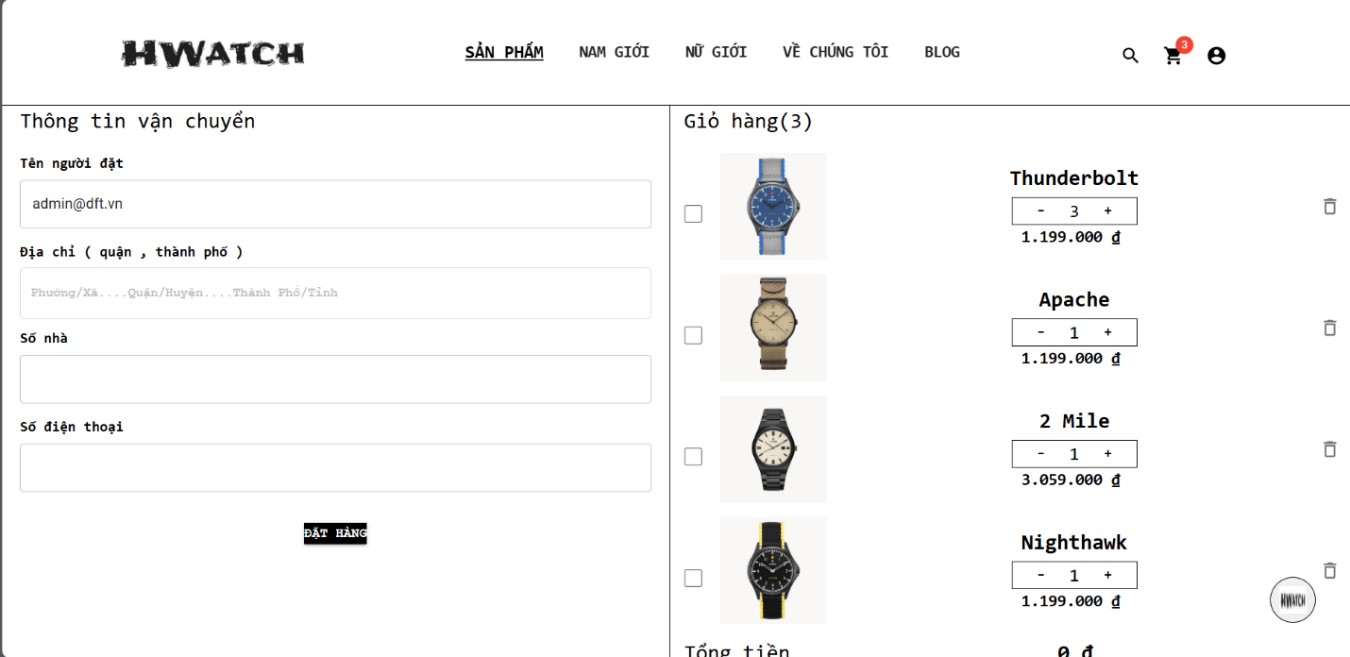
Trang chi tiết sản phẩm hiển thị chi tiết về giá tiền, mô tả, thông số sản phẩm giúp người dùng hiểu hơn và có lựa chọn sáng suốt



*Hình 3.4 Giao diện chi tiết sản phẩm*

### Giỏ hàng

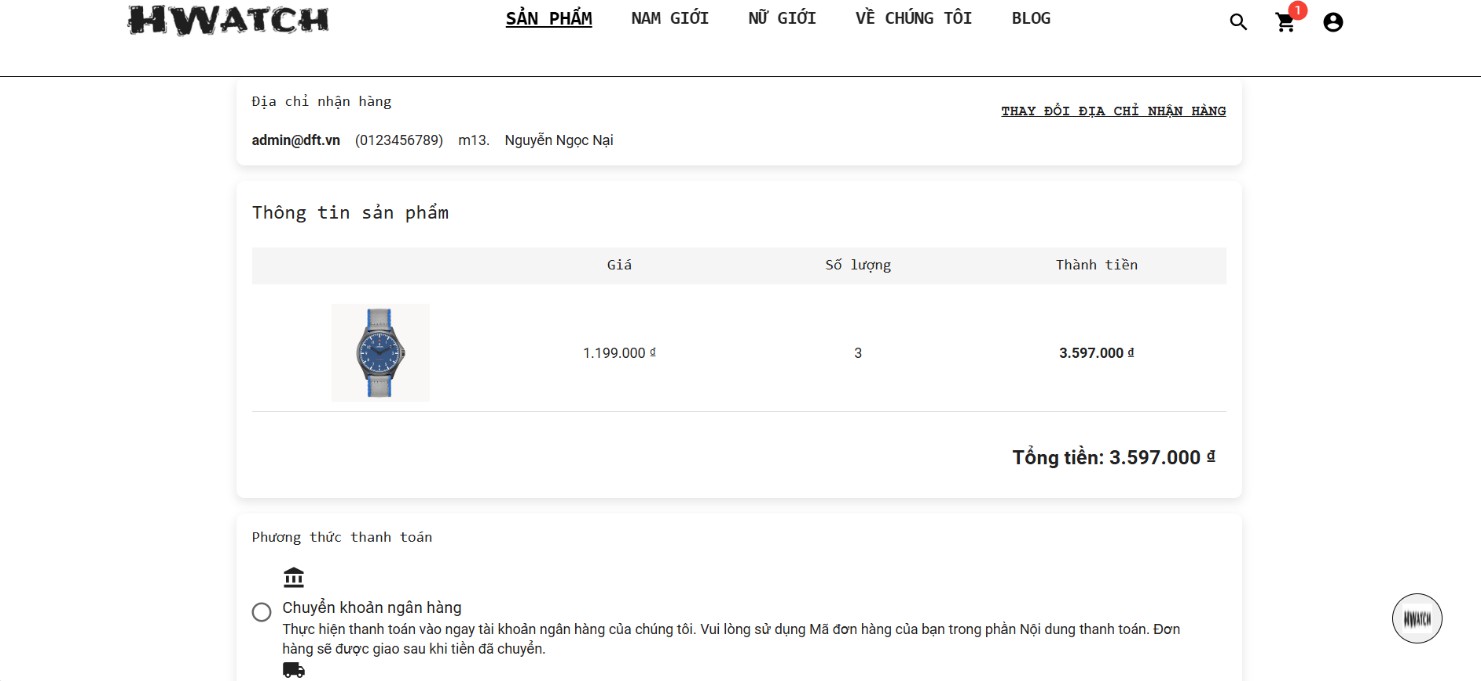
Trang giỏ hàng hiện thị các sản phẩm đã được người dùng thêm trước đó, tại đây người dùng cũng có thể điền đầy đủ các thông tin giao hàng và chuyển tới trang đặt hàng để đặt hàng



*Hình 3.5 Giao diện giỏ hàng*

### Đặt hàng

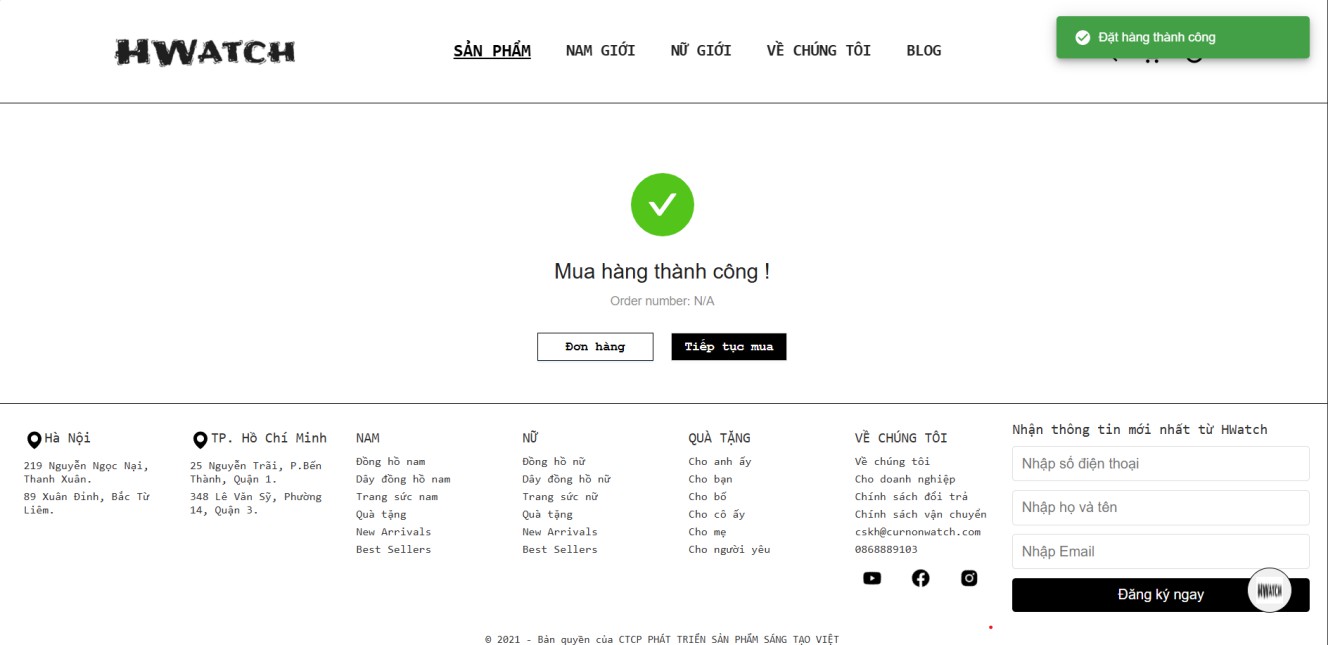
Trang đặt hàng hiển thị các sản phẩm đã đặt và các lựa chọn thanh toán cho người dùng



*Hình 3.6 Giao diện đặt hàng*

### Thông báo đặt hàng thành công

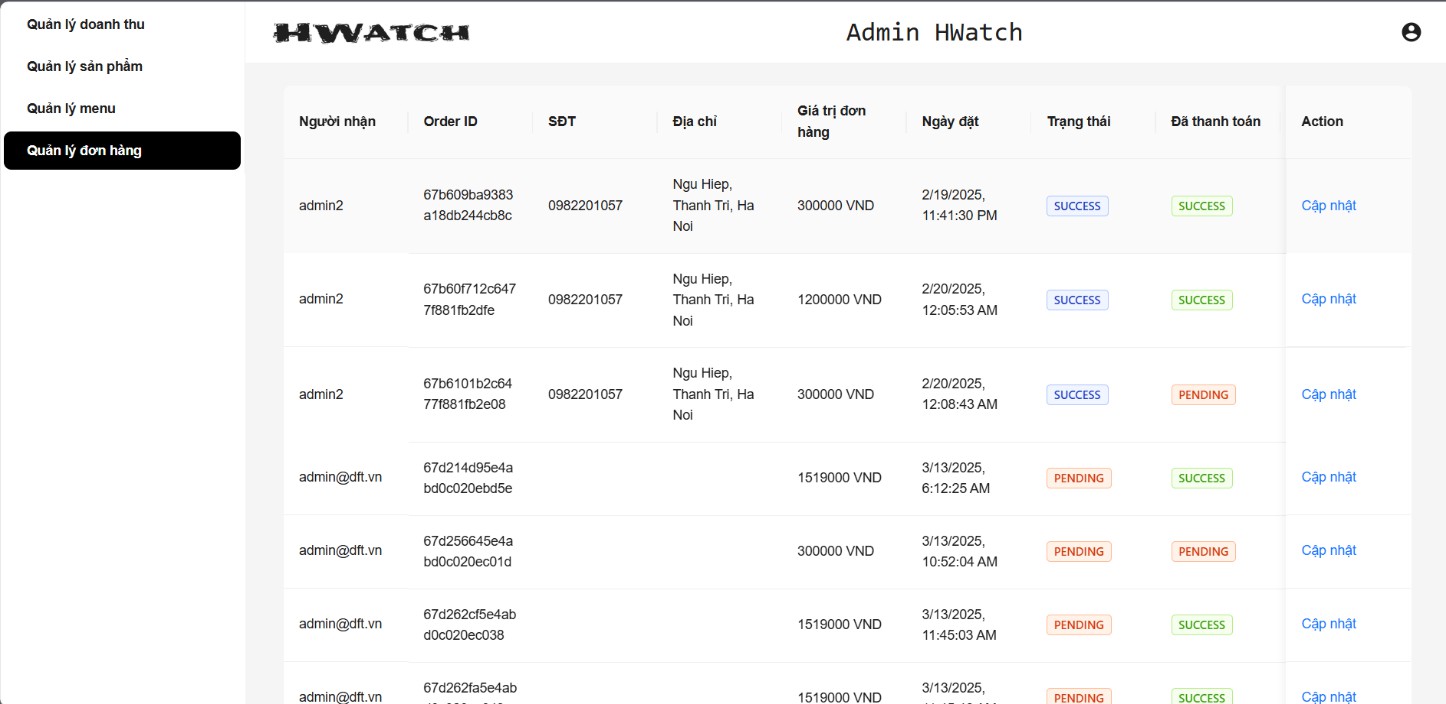
Sau khi đặt hàng thành công người dùng sẽ nhận được thông báo như dưới, người dùng có thể tiếp tục xem, mua hàng hoặc xem thông tin đơn hàng



*Hình 3.7 Giao diện đặt hàng thành công*

### Giới thiệu chức năng bên phía quản trị viên

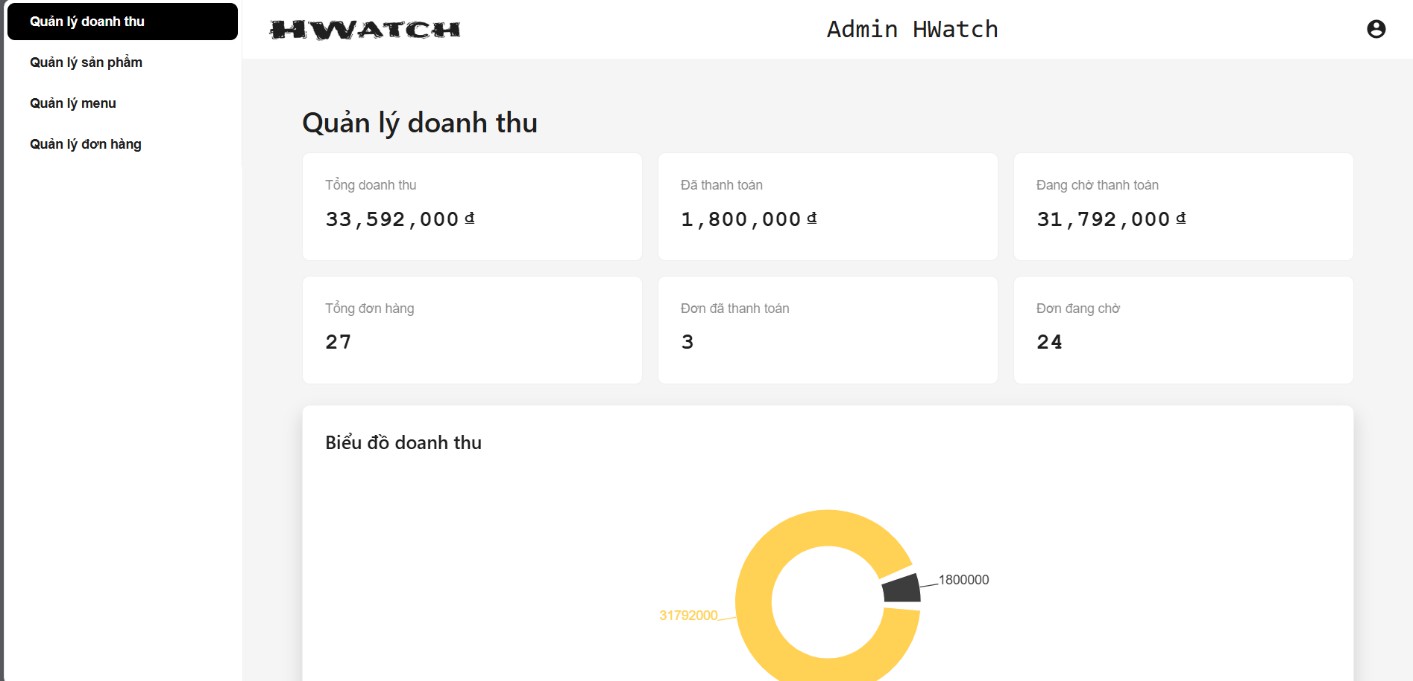
### Quản lý đơn hàng

Quản trị viên có thể xem các đơn hàng và chi tiết về đơn hàng đồng thời có thể cập nhật tình trạng đơn hàng từ đây

*Hình 3.8 Giao diện quản lý đơn hàng*

### Quản lý doanh thu

Quản trị viên được cung cấp báo cáo doanh thu chi tiết, doanh thu còn được biểu đồ hóa giúp quản trị viên có cái nhìn trực quan hơn



*Hình 3.9 Giao diện quản lý doanh thu*

### Kết quả thực nghiệm mô hình gợi ý trên website

### Thực nghiệm

*Bảng 3.1 Kết quả thực nghiệm mô hình gợi ý*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Product  ID | Description | Dial  Size | Dial  Color | Movement  Type | Strap  Material | Combined Text |
| P1 | “Elegant watch” | 40 | “Black” | “Automatic” | “Leather” | “Elegant watch 40 Black Automatic  Leather” |
| P2 | “Sporty design” | 42 | “Blue” | “Quartz” | “Rubber” | “Sporty design  42 Blue Quartz Rubber” |
| P3 | “Classic Style” | 40 | “Black” | “Automatic” | “Leather” | “Classic style 40 Black Automatic  Leather” |

Trong ví dụ này, chúng ta có 3 sản phẩm mẫu với ID lần lượt là P1, P2, P3. Với các thuộc tính “Description”, “Dial Size”, “Dial Color”, … Chúng ta sẽ tạo ra một feature mới với tên là “Combined Text” (ghép chuỗi các thuộc tính của mỗi sản phẩm). Sau đó ta sử dụng kỹ thuật TF-IDF để đánh giá mức độ quan trọng của mỗi từ trong feature mới so với toàn bộ tập dữ liệu. Ta được bẳng sau

*Bảng 3.2 Kết quả tính toán thực nghiệm*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Từ** | **Tài liệu** | **TF** | **IDF(log)** | **TF-IDF(TF x IDF)** |
| “Elegant” | 1 | 1/6 | log(3/1)≈ 0.477 | 0.079 |
| “watch” | 1 | 1/6 | log(3/1)≈ 0.477 | 0.079 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| “Sporty” | 2 | 1/6 | log(3/1)≈ 0.477 | 0.079 |
| “design” | 2 | 1/6 | log(3/1)≈ 0.477 | 0.079 |
| “Classic” | 3 | 1/6 | log(3/1)≈ 0.477 | 0.079 |
| “style” | 3 | 1/6 | log(3/1)≈ 0.477 | 0.079 |
| “40” | 1,3 | 1/6 | log(3/2)≈ 0.176 | 0.029(Tài liệu 1) |
|  | 3 | 1/6 | log(3/2)≈ 0.176 | 0.029(Tài liệu 3) |
| “42” | 2 | 1/6 | log(3/1)≈ 0.477 | 0.079 |
| “black” | 1,3 | 1/6 | log(3/2)≈ 0.176 | 0.029(Tài liệu 1) |
|  | 3 | 1/6 | log(3/2)≈ 0.176 | 0.029(Tài liệu 3) |
| “Automatic” | 1,3 | 1/6 | log(3/2)≈ 0.176 | 0.029(Tài liệu 1) |
|  | 3 | 1/6 | log(3/2)≈ 0.176 | 0.029(Tài liệu 3) |
| “Quartz” | 2 | 1/6 | log(3/1)≈ 0.477 | 0.079 |
| “Leather” | 1 | 1/6 | log(3/1)≈ 0.477 | 0.079 |
| “Rubber” | 2 | 1/6 | log(3/1)≈ 0.477 | 0.079 |

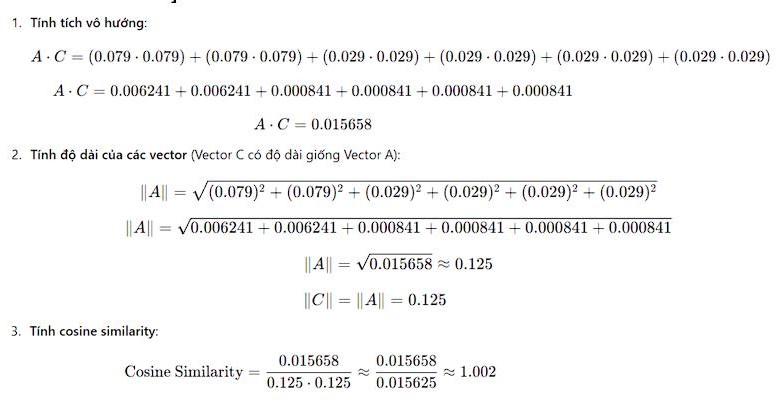
Sau khi có điểm đánh giá mức độ quan trọng của mỗi từ trong tập dữ liệu ta có thể suy ra vector đại diện cho 3 sản phẩm với ID P1, P2, P3 lần lượt là:

**Sản phẩm 1 (Vector A):** [ 0.079, 0.079, 0.029, 0.029, 0.029, 0.029]

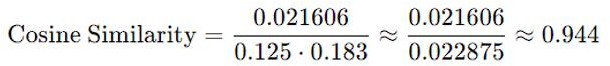
**Sản phẩm 2 (Vector B):** [ 0.079, 0.079, 0.079, 0.079, 0.079, 0.079]

**Sản phẩm 3 (Vector C):** [ 0.079, 0.079, 0.029, 0.029, 0.029, 0.029]

Qua 3 vector trên, ta có thể nhận thấy rõ sự tương đồng giữa 2 vector A và C. Chứng minh bằng góc tương đồng Cosine:



Tương tự, tính độ tương đồng giữa vector A và B:



**Kết luận:** Product 1 và Product 3 có độ tương tự cao hơn Product 1 và Product 2 nên khi người dùng chọn sản phẩm 1 hệ thống sẽ gợi ý người dùng sản phẩm 3

### Tổng kết

Từ các phép tính trong ma trận TF-IDF, hệ thống sẽ xác định mức độ tương đồng giữa các sản phẩm dựa trên các đặc trưng và từ đó đưa ra các gợi ý phù hợp với người dùng. Điều này giúp hệ thống có thể gợi ý sản phẩm ngay cả khi không có dữ liệu đánh giá trực tiếp từ người dùng.

# KẾT LUẬN

### Kết quả đạt được

Qua quá trình nghiên cứu và thực hiện, em đã hoàn thành khóa luận xây dựng website bán đồng hồ tích hợp thanh toán trực tuyến. Website đã cơ bản đáp ứng được các yêu cầu và chức năng chính của một nền tảng thương mại điện tử, bao gồm việc giới thiệu sản phẩm chi tiết, quản lý đơn hàng, giỏ hàng và thực hiện giao dịch mua bán một cách thuận tiện.

Một điểm nổi bật là hệ thống đã được tích hợp thanh toán trực tuyến thông qua cổng ZaloPay, giúp khách hàng dễ dàng hoàn tất đơn hàng với mức độ an toàn và bảo mật cao.

Giao diện của website được thiết kế thân thiện với người dùng, dễ sử dụng và tương thích tốt trên nhiều thiết bị.

### Hạn chế

Mặc dù đạt được những kết quả nhất định, website vẫn còn một số hạn chế cần cải thiện để nâng cao hiệu quả sử dụng và trải nghiệm người dùng. Hiện tại, hệ thống chưa có tính năng chatbot để hỗ trợ khách hàng trong việc tìm kiếm sản phẩm hoặc giải đáp thắc mắc nhanh chóng.

Ngoài ra, việc cập nhật các chương trình khuyến mãi, sự kiện ưu đãi còn thực hiện thủ công và chưa được đồng bộ tự động với hệ thống thông báo hoặc email marketing.

Hệ thống đăng nhập cũng chưa hỗ trợ liên kết nhanh với các nền tảng mạng xã hội như Google hoặc Facebook, khiến quá trình đăng ký và đăng nhập còn mất thời gian.

### Hướng phát triển

Trong tương lai, em đề xuất một số hướng phát triển cho website bao gồm: tích hợp chatbot để hỗ trợ khách hàng trực tuyến một cách nhanh chóng và hiệu quả, triển

khai hệ thống thông báo tự động qua email hoặc trình duyệt để cập nhật các chương trình ưu đãi, khuyến mãi mới.

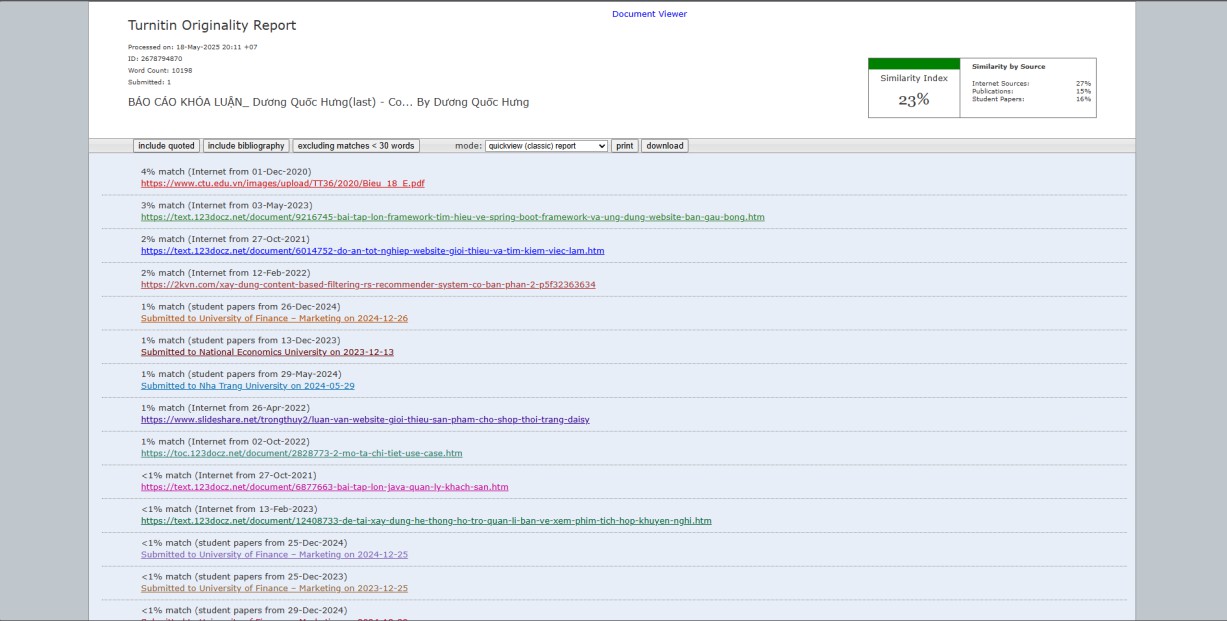
Đồng thời, mở rộng các phương thức đăng nhập thông qua mạng xã hội nhằm tăng sự tiện lợi và thu hút người dùng.

Bên cạnh đó, việc hợp tác với các đơn vị vận chuyển và dịch vụ thanh toán khác cũng sẽ góp phần nâng cao chất lượng phục vụ và mở rộng khả năng kinh doanh của nền tảng thương mại điện tử này.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | "W3schools," [Online]. Available: https://[www.w3schools.com/.](http://www.w3schools.com/) |
| [2] | Đại học Kinh tế Quốc dân, Slide bài giảng môn “Phân tích thiết kế hệ thống” |
| [3] | "MDN web docs - JavaScript Guide," [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide. |
| [4] | https://themeforest.net/ |
| [5] | https://cuuduongthancong.com/sjdt/co-so-du-lieu//dai-hoc-kinh-te-quoc-dan. |
| [6] | https://docs.staruml.io/ |

**KẾT QUẢ CHECK TURNITIN**

****