|  |
| --- |
|  |
| **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| BÁO CÁO ĐỒ ÁN TỐT NGHIỆP  NGÀNH CÔNG NGHỆ THÔNG TIN  ĐỀ TÀI: XÂY DỰNG WEBSITE BÁN GIÀY CHO CỬA HÀNG I-SPORT  BẰNG ASP.NET CORE |
|  |
|  |
|  |
|  |
| **CBHD:Th.S An Văn Minh** |
| **Sinh viên: Nguyễn Văn Long** |
| **Mã số sinh viên: 2020602847** |
|  |
|  |
|  |
| **Hà Nội – Năm 2024** |

**LỜI CẢM ƠN**

Trong suốt quá trình học tập tại trường Đại học Công Nghiệp Hà Nội, em đã nhận được rất nhiều sự quan tâm và giúp đỡ từ các thầy cô giáo trong trường. Em xin cảm ơn ban lãnh đạo nhà trường Đại học Công Nghiệp Hà Nội và tất cả các thầy cô giáo trong trường vì đã cho em những bài học, trải nghiệm khi học tại trường.

Em xin gửi lời cảm ơn đặc biệt đến các thầy cô trong khoa Công Nghệ Thông Tin. Các thầy cô đã trực tiếp giúp đỡ em tiếp thu được rất nhiều kiến thức bổ ích trong quá trình học tập tại trường. Các thầy cô trong khoa chính là những người đã đặt nền móng cho những kiến thức hiện giờ của em, giúp em có thể đem những kiến thức này áp dụng ra ngoài cuộc sống. Và em cũng cảm ơn những người bạn ở trường đã đồng hành cùng em trong 4 năm đại học. Em cũng xin gửi lời cảm ơn đến bố mẹ đã cho em có như ngày hôm nay. Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến thầy An Văn Minh, người đã trực tiếp hướng dẫn em trong đợt làm đồ án này. Thầy đã tận tình hướng dẫn, giúp đỡ em rất nhiều để có thể hoàn thiện được đề tài “***Xây dựng website bán giày cho cửa hàng i-Sport bằng Asp.net Core***”. Một lần nữa, em xin cảm ơn thầy.

Do điều kiện thời gian có hạn và kinh nghiệm làm báo cóa chưa nhiều nên em không thể tránh khỏi những thiếu sót trong lúc trình bày cũng như làm báo cáo, kính mong nhận được những ý kiến đóng góp từ quý thầy cô.

Em xin chân thành cảm ơn!

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH iii](#_Toc167429421)

[LỜI NÓI ĐẦU v](#_Toc167429422)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc167429423)

[1.1. Tên đề tài 1](#_Toc167429424)

[1.2. Mục đích của đề tài 1](#_Toc167429425)

[1.3. Lý do chọn đề tài 1](#_Toc167429426)

[1.4. Cơ sở lý thuyết của đề tài 2](#_Toc167429427)

[1.5. Cơ sở thực tiễn 3](#_Toc167429428)

[1.6. Phương pháp nghiên cứu để xây dựng và phát triển đề tài 4](#_Toc167429429)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG 6](#_Toc167429430)

[2.1. Cơ sở lý thuyết 6](#_Toc167429431)

[2.1.1. Tổng quan về quy trình phát triển phần mềm 6](#_Toc167429432)

[2.1.2. Tổng quan về cơ sở dữ liệu 7](#_Toc167429433)

[2.1.3. Tổng quan về phân tích và thiết kế hệ thống 9](#_Toc167429434)

[2.2. Công nghệ sử dụng 10](#_Toc167429435)

[2.2.1. Ngôn ngữ UML 10](#_Toc167429436)

[2.2.2. Ngôn ngữ lập trình C# 12](#_Toc167429437)

[2.2.3. Asp.Net Core 13](#_Toc167429438)

[2.2.4. Angular 14](#_Toc167429439)

[2.2.5. Hệ quản trị cơ sở dữ liệu SQL Server 16](#_Toc167429440)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 18](#_Toc167429441)

[3.1. Khảo sát hệ thống 18](#_Toc167429442)

[3.2. Phân tích hệ thống 19](#_Toc167429443)

[3.2.1. Vai trò của người dùng 19](#_Toc167429444)

[3.2.2. Yêu cầu chức năng 19](#_Toc167429445)

[3.2.3. Yêu cầu phi chức năng 20](#_Toc167429446)

[3.3. Biểu đồ use case tổng quát 20](#_Toc167429447)

[3.4. Đặc tả use case 20](#_Toc167429448)

[3.4.1. Use case đăng nhập 20](#_Toc167429449)

[3.4.2. Use case đăng ký 22](#_Toc167429450)

[3.4.3. Use case tìm kiếm 24](#_Toc167429451)

[3.4.4. Use case xem chi tiết sản phẩm 27](#_Toc167429452)

[3.4.6. Use case thêm giỏ hàng 29](#_Toc167429453)

[3.4.7. Use case xem giỏ hàng 31](#_Toc167429454)

[3.4.8. Use case đặt hàng 33](#_Toc167429455)

[3.4.9. Use case quản lý sản phẩm 35](#_Toc167429456)

[3.4.10. Use case quản lý thuộc tính sản phẩm 39](#_Toc167429457)

[3.4.11. Use case quản lý người dùng 43](#_Toc167429458)

[3.4.12. Use case quản lý danh mục 46](#_Toc167429459)

[3.4.13. Use case quản lý quyền 48](#_Toc167429460)

[3.5. Mô tả cơ sở dữ liệu 51](#_Toc167429461)

[3.6. Mô hình thực thể liên kết 58](#_Toc167429462)

[CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG 59](#_Toc167429463)

[4.1. Công cụ sử dụng 59](#_Toc167429464)

[4.1.1. Microsoft Visual Studio 59](#_Toc167429465)

[4.1.2. StarUML 59](#_Toc167429466)

[4.1.3. Sql Server Management Studio 60](#_Toc167429467)

[4.2. Giao diện trang web 61](#_Toc167429468)

[KẾT LUẬN 71](#_Toc167429469)

[TÀI LIỆU THAM KHẢO 73](#_Toc167429470)

**DANH MỤC HÌNH ẢNH**

Hình 3.1. Biểu đồ use case tổng quát

Hình 3.2. Biểu đồ use case đăng nhập

Hình 3.3. Biểu đồ trình tự đăng nhập

Hình 3.4. Biểu đồ use case đăng ký

Hình 3.5. Biểu đồ trình tự đăng ký

Hình 3.6. Biểu đồ use case tìm kiếm

Hình 3.7. Biểu đồ trình tự tìm kiếm

Hình 3.8. Biểu đồ use case xem chi tiết sản phẩm

Hình 3.9. Biểu đồ trình tự xem chi tiết sản phẩm

Hình 3.10. Biểu đồ trình tự xem sản phẩm theo loại

Hình 3.11. Biểu đồ use case thêm giỏ hàng

Hình 3.12. Biểu đồ trình tự thêm giỏ hàng

Hình 3.13. Biểu đồ use case xem giỏ hàng

Hình 3.14. Biểu đồ trình tự xem giỏ hàng

Hình 3.15. Biểu đồ use case đặt hàng

Hình 3.16. Biểu đồ trình tự đặt hàng

Hình 3.17. Biểu đồ use case quản lý sản phẩm

Hình 3.18. Biểu đồ trình tự use case quản lý sản phẩm

Hình 3.19. Biểu đồ use case quản lý thuộc tính sản phẩm

Hình 3.20. Biểu đồ trình tự quản lý thuộc tính sản phẩm

Hình 3.21. Biểu đồ use case quản lý người dùng

Hình 3.22. Biểu đồ trình tự quản lý người dùng

Hình 3.23. Biểu đồ use case quản lý danh mục

Hình 3.24. Biểu đồ trình tự quản lý danh mục

Hình 3.25. Biểu đồ use case quản lý quyền

Hình 3.26. Biểu đồ tuần tự quản lý quyền

Hình 3.27 Mô hình thực thể liên kết

Hình 4.1. Trang chủ web bán giày

Hình 4.2. Giao diện đăng nhập

Hình 4.3. Giao diện đăng ký

Hình 4.4. Giao diện đổi mật khẩu

Hình 4.5. Giao diện đổi thông tin cá nhân

Hình 4.6. Giao diện xem chi tiết sản phẩm

Hình 4.7. Giao diện thêm vào giỏ hàng

Hình 4.8. Giao diện đặt hàng

Hình 4.9. Giao diện xem đơn hàng

Hình 4.10. Giao diện chatbot

Hình 4.11. Giao diện quản lý người dùng

Hình 4.12. Giao diện quản lý đơn hàng

Hình 4.13. Giao diện quản lý sản phẩm

Hình 4.14. Giao diện quản lý nhà sản xuất

Hình 4.15. Giao diện quản lý loại sản phẩm

Hình 4.16. Giao diện quản lý thuộc tính sản phẩm

Hình 4.17. Giao diện thống kê

# LỜI NÓI ĐẦU

Với thời đại công nghệ phát triển như hiện nay, nhu cầu mua hàng online của người tiêu dùng ngày càng tăng. Trong đó có nhu cầu về mua giày, khách hàng và người bán có thể mua bán trực tuyến rất có lợi về nhiều mặt như thời gian, khoảng cách, dễ dàng so sánh nhiều loại sản phẩm, …

Sau khi tìm hiểu và dưới sự hướng dẫn của thầy An Văn Minh, em đã phân tích, thiết kế và tiến hành xây dựng ***Trang Web bán giày Isport***. Để giúp việc mua bán được triển khai một cách dễ dàng, ta cần một ứng dụng website có nhiều sản phẩm, dễ dàng sửa dụng, giao diện thân thiện với người dùng. Trang web gồm các chức năng cơ bản như xem sản phầm, chi tiết sản phẩm, tìm kiêm sản phẩm, thêm và xem giỏ hàng, đặt hàng.v.v. Người quản trị có thể quản lý người dùng, quản lý tài khoản, quản lý sản phẩm,.v.v..

Tuy nhiên vì thời gian có hạn và bản thân còn thiếu nhiều kinh nghiệm thực tiễn nên nội dung của báo cáo không tránh khỏi những thiếu sót, em rất mong nhận sự góp ý, chỉ bảo thêm của quý thầy cô hướng dẫn và thầy cô để báo cáo này được hoàn thiện hơn**.**

**Báo cáo gồm 4 chương:**

Chương 1: Tổng quan về đề tài

Chương 2: Cơ sở lý thuyết và công nghệ sử dụng

Chương 3: Phân tích và thiết kế hệ thống

Chương 4: Xây dựng ứng dụng

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

## 1.1. Tên đề tài

**“*Xây dựng Website bán giày cho cửa hàng i-Sport bằng Asp.Net Core”***

Đây là một website chuyên cung cấp các mẫu giày thể thao giúp người dùng có thể mua bán trực tuyến online một cách dễ dàng.

## 1.2. Mục đích của đề tài

1. Kiểm chứng kiến thức: Đồ án tốt nghiệp là cơ hội để áp dụng tất cả kiến thức đã học trong quá trình đào tạo. Xây dựng một trang web bán hàng sẽ cần phải áp dụng nhiều khía cạnh của công nghệ Asp.net Core.

2. Thể hiện kỹ năng: Bằng việc hoàn thành đồ án này, bạn có thể thể hiện khả năng phân tích, thiết kế và phát triển phần mềm web của mình trước các giảng viên, cán bộ khoa và đồng nghiệp.

3. Đồ án tốt nghiệp thành công sẽ là một thành tựu quan trọng trong quá trình học tập và cũng là một phần quan trọng trong danh mục các dự án của em khi ra trường và tìm kiếm việc làm.

4. Thực hành quy trình phát triển phần mềm: Quá trình phát triển đồ án sẽ giúp em hiểu rõ hơn về quy trình thiết kế, phát triển, kiểm thử và triển khai phần mềm trong một dự án thực tế.

5. Tìm hiểu công nghệ mới: Asp.net Core là một công nghệ phát triển web tiên tiến, việc sử dụng nó trong đồ án tốt nghiệp sẽ giúp bạn tiếp cận và làm quen với những công nghệ mới.

6. Tiếp thu năng giải quyết vấn đề: Xây dựng một trang web bán hàng yêu cầu bạn phải suy nghĩ logic, sắp xếp dữ liệu và giải quyết các vấn đề thực tế, từ đó rèn luyện khả năng tư duy logic và phân tích vấn đề.

## 1.3. Lý do chọn đề tài

Với mục tiêu chính là cung cấp cho bản thân em một cơ hội thực hành và áp dụng kiến thức đã học trong quá trình đào tạo. Cụ thể, các lý do chọn đề tài bao gồm:

1. Thực hành và áp dụng kiến thức về lập trình web và phát triển ứng dụng ASP.NET.

2. Nâng cao kỹ năng kỹ thuật liên quan đến phát triển website, bao gồm cả phát triển giao diện người dùng và phát triển cơ sở dữ liệu.

3. Hiểu biết thực tiễn về quy trình phát triển phần mềm từ việc xác định yêu cầu, thiết kế, triển khai và kiểm thử.

4. Tạo ra một sản phẩm thực tế là website bán đồ uống có khả năng thực sự được sử dụng trong thực tế.

5. Tạo cơ hội nghề nghiệp cho em trong tương lai, bằng cách xây dựng một dự án có thể trưng dụng khi tìm kiếm việc làm hoặc khi ứng dụng vào môi trường làm việc thực tế.

Ngoài ra đề tài này còn là cơ hội quan trọng để em thể hiện và phát triển kỹ năng của mình trong lĩnh vực phát triển website.

**1.4. Cơ sở lý thuyết của đề tài**

**ASP.NET Core và C#:**

- ASP.NET Core: Là một framework phát triển ứng dụng web mạnh mẽ của Microsoft, dựa trên nền tảng .NET Core. Nó cung cấp một mô hình lập trình theo mô hình MVC (Model-View-Controller) để xây dựng các ứng dụng web linh hoạt và dễ bảo trì. So với ASP.NET truyền thống, ASP.NET Core có hiệu suất cao hơn, khả năng đa nền tảng và tích hợp sâu với các công nghệ mới như Docker và Kubernetes.

- Ngôn ngữ lập trình C#: Là ngôn ngữ chính được sử dụng trong việc phát triển các ứng dụng trên nền tảng ASP.NET Core. C# là một ngôn ngữ lập trình hiệu quả, dễ học và mạnh mẽ, với cú pháp tương tự như C++ và hỗ trợ nhiều tính năng hiện đại như generics, lập trình hướng đối tượng và async/await.

Kiến thức về phát triển web:

- HTML, CSS và JavaScript: Là các ngôn ngữ lập trình cơ bản và không thể thiếu trong phát triển giao diện người dùng của các trang web trên nền tảng ASP.NET Core (MVC). HTML định nghĩa cấu trúc của trang web, CSS điều chỉnh kiểu dáng và trang trí, và JavaScript thêm tính năng tương tác động cho trang web.

**Quy trình phát triền phần mềm:**

**-** Xác định rõ yêu cầu của trang web bán hàng, tìm hiểu về sản phẩm/dịch vụ, đối tượng khách hàng, chức năng cần có, giao diện người dùng, v.v.

- Xây dựng wireframes và thiết kế giao diện người dùng dựa trên yêu cầu thu thập được. Đảm bảo trang web thân thiện với người dùng, dễ sử dụng và hấp dẫn.

- Sử dụng HTML, CSS, JavaScript và framework Angular để xây dựng giao diện người dùng thân thiện và tương tác.

- Xây dựng hệ thống quản trị để quản lý dữ liệu, xử lý logic kinh doanh, xác thực người dùng, thanh toán,v.v..

**Cơ sở dữ liệu:**

- Thiết kế và triển khai cơ sở dữ liệu để lưu trữ thông tin sản phẩm, đơn hàng, thông tin khách hàng, v.v. Sử dụng hệ quản trị cơ sở dữ liệu phù hợp như MySQL

**Phân tích và thiết kế hệ thống:**

- Phân tích và thiết kế hệ thống cho một trang web bán hàng rất quan trọng để đảm bảo hệ thống hoạt động hiệu quả, an toàn và thu hút người dùng.

**Bảo mật và xác thực:**

- Đảm bảo an toàn và bảo mật cho thông tin người dùng và dữ liệu bằng cách sử dụng các biện pháp bảo mật như xác thực và ủy quyền.

- Mã hóa dữ liệu quan trọng trước khi lưu trữ vào cơ sở dữ liệu để đảm bảo tính bảo mật của nó.

Thông hiểu về các khái niệm và nguyên lý này sẽ giúp xây dựng ứng dụng web ASP.NET Core một cách hiệu quả và linh hoạt.

**1.5. Cơ sở thực tiễn**

**-** Người dùng có thể dễ dàng tìm kiếm và đặt hàng sản phẩm mà mình muốn mua trực tuyến một cách nhanh chóng và thuận tiện.

- Người dùng có thể chọn sản phẩm theo nhu cầu của mình bao gồm chọn mã sản phẩm, nhà sản xuất, size, màu..

- Người dùng mua hàng có thể theo dõi đơn hàng đã đặt hoặc có thể hủy đơn hàng đã dặt một cách nhanh chóng.

- Người dùng quản trị có thể quản lý người dùng, sản phẩm, đơn hàng và thống kê doanh thu.

- Thông tin khách hàng được bảo vệ bằng các công nghệ bảo mật hiện đại, đảm bảo tính riêng tư và an toàn.

- Các thông tin về sản phẩm sẽ được quản trị viên liên tục cập nhật và kiểm tra định kỳ để đảm bảo tính chính xác và độ tin cậy cho khách hàng.

**1.6. Phương pháp nghiên cứu để xây dựng và phát triển đề tài**

- Nghiên cứu thị trường và phân tích yêu cầu:

+ Tiến hành khảo sát thị trường để hiểu về nhu cầu và xu hướng mua sắm nem đặc sản trực tuyến.

+ Phân tích yêu cầu chức năng và phi chức năng của trang web bán giày, xác định các tính năng quan trọng như quản lý sản phẩm, đơn hàng, khách hàng và tích hợp thanh toán.

- Nghiên cứu công nghệ:

+ Nghiên cứu về Angular và Web API ASP.NET Core, bao gồm các tính năng, ưu điểm, hạn chế và cách thức triển khai.

+ So sánh và lựa chọn công nghệ phù hợp cho dự án, dựa trên yêu cầu chức năng và mục tiêu đề ra.

- Thiết kế kiến trúc hệ thống:

+ Xác định kiến trúc tổng quan của hệ thống, bao gồm cấu trúc giao diện khách hàng và quản trị.

+ Thiết kế cấu trúc dữ liệu và quan hệ giữa các thành phần của hệ thống.

- Triển khai và phát triển phần mềm:

+ Phát triển giao diện người dùng sử dụng Angular, bao gồm các trang chính như trang chủ, trang sản phẩm, giỏ hàng và trang thanh toán.

+ Web API ASP.NET Core để cung cấp dữ liệu cho giao diện người dùng, bao gồm các endpoint cho việc quản lý sản phẩm, đơn hàng, khách hàng và tích hợp thanh toán.

- Tích hợp và kiểm thử:

+ Tích hợp các thành phần của hệ thống và kiểm tra tính tương thích và ổn định.

+ Tiến hành kiểm thử chức năng và hiệu suất cho cả phần giao diện khách hàng và giao diện quản trị của trang web.

- Triển khai và đánh giá:

+ Triển khai trang web trên môi trường sản phẩm thực tế và đảm bảo sự ổn định và khả dụng của hệ thống.

+ Đánh giá hiệu suất của hệ thống dựa trên các tiêu chí như tốc độ tải trang, thời gian phản hồi và độ tin cậy của các tính năng.

- Tổng kết và đánh giá kết quả:

+ Tổng kết quá trình phát triển dự án và đánh giá mức độ đạt được các mục tiêu đã đề ra.

+ Đề xuất các cải tiến và hướng phát triển tương lai cho hệ thống.

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG**

**2.1. Cơ sở lý thuyết**

**2.1.1. Tổng quan về quy trình phát triển phần mềm**

1. Phân tích

- Xác định các tính năng cần thiết như quản lý sản phẩm, giỏ hàng, thanh toán, quản lý đơn hàng, quản lý người dùng, và tích hợp các phương thức thanh toán.

Phân tích yêu cầu.

- Tạo tài liệu SRS (Software Requirements Specification) để mô tả chi tiết các yêu cầu chức năng và phi chức năng của hệ thống.

Sử dụng các kỹ thuật mô hình hóa như Use Case Diagrams, Sequence Diagrams để mô tả các yêu cầu.

2. Thiết kế hệ thống

- Thiết kế kiến trúc

+ Xác định kiến trúc 3 tầng: Presentation Layer (Angular), Business Logic Layer (ASP.NET Core), và Data Access Layer (Entity Framework Core).

- Thiết kế API RESTful để giao tiếp giữa front-end và back-end.

- Thiết kế cơ sở dữ liệu

+ Thiết kế lược đồ cơ sở dữ liệu sử dụng Entity Framework Core.

+ Xác định các bảng cần thiết: Users, Products, Orders, OrderDetails, Categories, Promotions, v.v.

- Thiết kế giao diện người dùng

+ Tạo các wireframe và prototype cho các trang chính: Home, Product Listing, Product Detail, Cart, Checkout, User Account.

+ Đảm bảo giao diện người dùng thân thiện và tương thích với các thiết bị di động.

3. Phát triển hệ thống

- Phát triển front-end với Angular

+ Cài đặt Angular CLI và tạo project Angular mới.

+ Xây dựng các component, service, và module cho các tính năng khác nhau.

+ Sử dụng Angular Router để điều hướng giữa các trang.

+ Tích hợp các thư viện UI như Angular Material hoặc Bootstrap.

- Phát triển back-end với ASP.NET Core

+ Tạo project ASP.NET Core và cấu hình các middleware cần thiết.

+ Xây dựng các API controller để xử lý các yêu cầu từ front-end.

+ Sử dụng Entity Framework Core để kết nối và tương tác với cơ sở dữ liệu.

+ Cấu hình Identity để quản lý xác thực và phân quyền người dùng.

**2.1.2. Tổng quan về cơ sở dữ liệu**

1. Khái niệm cơ bản

- Cơ sở dữ liệu (Database): Là tập hợp các dữ liệu có cấu trúc được tổ chức và lưu trữ để dễ dàng truy cập, quản lý và cập nhật.

- Hệ quản trị cơ sở dữ liệu (Database Management System - DBMS): Là phần mềm cho phép người dùng định nghĩa, tạo, duy trì và kiểm soát truy cập đến cơ sở dữ liệu. Ví dụ: MySQL, PostgreSQL, SQL Server, MongoDB.

2. Các loại cơ sở dữ liệu

- Cơ sở dữ liệu quan hệ (Relational Database): Dữ liệu được tổ chức dưới dạng bảng, và các bảng này có quan hệ với nhau. Sử dụng SQL (Structured Query Language) để truy vấn và thao tác dữ liệu. Ví dụ: MySQL, PostgreSQL, Oracle.

Cơ sở dữ liệu không quan hệ (NoSQL Database): Dữ liệu không được tổ chức dưới dạng bảng. Thường được sử dụng cho các ứng dụng yêu cầu xử lý dữ liệu lớn và có tính linh hoạt cao. Ví dụ: MongoDB, Cassandra, Redis.

3. Thiết kế cơ sở dữ liệu

- Mô hình thực thể - quan hệ (Entity-Relationship Model - ER Model): Sử dụng các thực thể (entities) và mối quan hệ (relationships) giữa chúng để thiết kế cơ sở dữ liệu.

- Chuẩn hóa cơ sở dữ liệu (Database Normalization): Quá trình tổ chức dữ liệu trong cơ sở dữ liệu để giảm thiểu sự dư thừa và đảm bảo tính toàn vẹn của dữ liệu. Bao gồm các dạng chuẩn (Normal Forms) như 1NF, 2NF, 3NF.

4. Các thành phần của cơ sở dữ liệu

- Bảng (Table): Là cấu trúc chính trong cơ sở dữ liệu quan hệ, chứa các hàng (rows) và cột (columns).

- Hàng (Row): Một bản ghi (record) trong bảng, chứa dữ liệu về một thực thể cụ thể.

- Cột (Column): Đại diện cho một thuộc tính của thực thể.

- Khóa chính (Primary Key): Một cột hoặc tập hợp các cột mà giá trị là duy nhất cho mỗi hàng trong bảng.

- Khóa ngoại (Foreign Key): Một cột hoặc tập hợp các cột trong một bảng, tham chiếu đến khóa chính của một bảng khác.

5. Truy vấn cơ sở dữ liệu

- SQL (Structured Query Language): Ngôn ngữ chuẩn để truy cập và thao tác với cơ sở dữ liệu quan hệ.

- SELECT: Truy vấn dữ liệu từ một hoặc nhiều bảng.

- INSERT: Thêm dữ liệu mới vào bảng.

- UPDATE: Cập nhật dữ liệu hiện có trong bảng.

- DELETE: Xóa dữ liệu khỏi bảng.

6. Quản lý cơ sở dữ liệu

- Sao lưu và khôi phục (Backup and Restore): Đảm bảo an toàn dữ liệu bằng cách sao lưu định kỳ và khả năng khôi phục khi cần.

- Bảo mật cơ sở dữ liệu (Database Security): Bảo vệ dữ liệu khỏi truy cập trái phép thông qua các kỹ thuật như mã hóa, xác thực người dùng và phân quyền truy cập.

- Tối ưu hóa cơ sở dữ liệu (Database Optimization): Cải thiện hiệu suất truy vấn và xử lý dữ liệu thông qua các kỹ thuật như lập chỉ mục (indexing), phân mảnh (partitioning), và tối ưu hóa câu lệnh SQL.

7. Tích hợp cơ sở dữ liệu với ứng dụng

- ORM (Object-Relational Mapping): Kỹ thuật cho phép chuyển đổi giữa các đối tượng trong ngôn ngữ lập trình và các bảng trong cơ sở dữ liệu. Ví dụ: Entity Framework (C#), Hibernate (Java), SQLAlchemy (Python).

- Kết nối cơ sở dữ liệu: Sử dụng các thư viện và framework để kết nối ứng dụng với cơ sở dữ liệu. Ví dụ: JDBC (Java), ADO.NET (C#), psycopg2 (Python).

**2.1.3. Tổng quan về phân tích và thiết kế hệ thống**

1. Các loại yêu cầu

- Yêu cầu chức năng (Functional Requirements):

+ Mô tả các chức năng mà hệ thống cần thực hiện.

- Ví dụ: Hệ thống phải cho phép người dùng đăng ký tài khoản, đăng nhập, tìm kiếm sản phẩm, thêm sản phẩm vào giỏ hàng.

- Yêu cầu phi chức năng (Non-Functional Requirements):

+ Mô tả các đặc tính của hệ thống không liên quan trực tiếp đến chức năng, như hiệu suất, bảo mật, khả năng mở rộng, tính dễ sử dụng.

- Yêu cầu hệ thống (System Requirements):

+ Mô tả các yêu cầu liên quan đến phần cứng và phần mềm cần thiết để hệ thống hoạt động.

2. Quá trình phân tích yêu cầu

- Phân tích yêu cầu (Requirements Analysis)

+ Mô hình hóa yêu cầu (Requirements Modeling): Sử dụng các kỹ thuật và công cụ để biểu diễn yêu cầu một cách rõ ràng và dễ hiểu, như Use Case Diagrams, Activity Diagrams, Class Diagrams.

- Phân tích tính khả thi (Feasibility Analysis): Đánh giá tính khả thi về kỹ thuật, tài chính và thời gian của các yêu cầu.

- Đặc tả yêu cầu (Requirements Specification)

- Tài liệu SRS (Software Requirements Specification): Soạn thảo tài liệu mô tả chi tiết tất cả các yêu cầu của hệ thống.

- Use Case Specification: Mô tả chi tiết các trường hợp sử dụng (use cases), bao gồm các kịch bản chính và phụ, điều kiện tiên quyết, và kết quả mong đợi.

3. Kỹ thuật mô hình hóa yêu cầu

- Use Case Diagrams

+ Biểu diễn các chức năng chính của hệ thống và các tác nhân (actors) tương tác với hệ thống.

+ Giúp xác định và tổ chức các yêu cầu chức năng.

- Activity Diagrams

+ Biểu diễn luồng công việc hoặc các hoạt động trong hệ thống.

+ Giúp hiểu rõ quy trình làm việc và các bước thực hiện chức năng.

- Sequence Diagrams

+ Mô tả thứ tự các tương tác giữa các đối tượng trong hệ thống theo thời gian.

+ Giúp hiểu rõ cách thức các thành phần trong hệ thống tương tác với nhau.

- Class Diagrams

+ Biểu diễn cấu trúc tĩnh của hệ thống thông qua các lớp và mối quan hệ giữa chúng.

Giúp xác định các thực thể và mối quan hệ trong hệ thống.

**2.2. Công nghệ sử dụng**

**2.2.1. Ngôn ngữ UML**

UML là viết tắt của “Unified Modeling Language” - là ngôn ngữ mô hình hóa thống nhất, dùng để đặc tả, xây dựng và tư liệu hóa phần mềm hướng đối tượng.

UML dùng để:

* Trực quan hoá: Đưa ra các mô hình trực quan về hệ thống để chúng ta có thể hiểu được.
* Đặc tả: UML cho phép xây dựng các mô hình đúng đắn và đầy đủ để mô tả hệ thống.
* Các mô hình UML có thể liên kết trực tiếp với nhiều ngôn ngữ lập trình Java, C++, VB, ...
* Lập tài liệu: UML cho phép tạo các tài liệu như tài liệu về các yêu cầu của người dùng, kiến trúc hệ thống, thiết kế hệ thống, kiểm thử, kế hoạch dự án,...

UML có thể được sử dụng làm công cụ giao tiếp giữa người dùng, nhà phân tích, nhà thiết kế và nhà phát triển phần mềm.

Các phần tử của UML:

View (quan sát): Theo các phương diện khác nhau của hệ thống cần phân tích, thiết kế. Dựa vào các quan sát để thiết lập kiến trúc cho hệ thống cần phát triển, ta cần sử dụng các loại biểu đồ để mô tả hệ thống.

***Có 9 loại biểu đồ:***

* Use Case Diagram (biểu đồ casử dụng): Mô tả sự tương tác giữa các tác nhân và hệ thống thông qua các ca sử dụng.
* Class Diagram (biểu đồ lớp): Biểu đồ lớp chỉ ra sự tồn tại giữa các lớp và mối quan hệ giữa chúng trong thiết kế logic trong hệ thống. Khung nhìn tĩnh của hệ thống chủ yếu hỗ trợ các chức năng của hệ thống.
* Object Diagram (biểu đồ đối tượng): Biểu đồ đối tượng đại diện cho một thể hiện của biểu đồ lớp, cho nên các khái niệm cơ bản là giống biểu đồ lớp. Biểu đồ đối tượng được sử dụng để hiện thị một tập hợp các đối tượng và các mối quan hệ của chúng như một thể hiện.
* Sequence Diagram (biểu đồ trình tự): Là biểu đồ để xác định và chỉ rõ vai trò của đối tượng tham gia vào luồng sự kiện của use case.
* Collaboration Diagram (biểu đồ cộng tác): Tương tự như biểu đồ trình tự nhưng nhấn mạnh vào sự tương tác của các đối tượng trên cơ sở cộng tác với nhau bằng cách trao đổi các thông điệp để thực hiện các yêu cầu theo ngữ cảnh công việc.
* State Diagram (biểu đồ trạng thái): Thể hiện chu kỳ hoạt động của các đối tượng, của các hệ thống con và của cả hệ thống.
* Activity Diagram (biểu đồ hành động): Được sử dụng để mô tả các hoạt động và các hành động được thực hiện trong một use case.
* Component Diagram (biểu đồ thành phần): Chỉ ra cấu trúc vật lý của các thành phần trong hệ thống.
* Deployment Diagram (biểu đồ triển khai): Chỉ ra cách bố trí vật lý các thành phần theo kiến trúc được thiết kế của hệ thống.

Relationship (quan hệ):

* Realization (hiện thực hoá):Là quan hệ ngữ nghĩa giữa giao diện và lớp.
* Dependency (phụ thuộc):Là quan hệ ngữ nghĩa giữa hai phần tử, trong đó sự thay đổi của một tử sẽ tác động đến ngữ nghĩa của phần tử phụ thuộc.
* Generalization (kế thừa): Là quan hệ mô tả sự khái quát hoá mà trong đó một số đối tượng cụ thể (của lớp con) sẽ được kế thừa các thuộc tính, các phương thức của các đối tượng tổng quát (lớp cơ sở).

Association (kết hợp): Là quan hệ cấu trúc xác định mối liên kết giữa các lớp đối tượng.

**2.2.2. Ngôn ngữ lập trình C#**

- C# (hay còn gọi là C Sharp, là ngôn ngữ lập trình hướng đối tượng, hiện đại do Microsoft phát triển như một phần của nền tảng .NET. Nó được giới thiệu lần đầu tiên vào năm 2000 và kể từ đó đã trở thành một trong những ngôn ngữ lập trình được sử dụng rộng rãi nhất trong ngành.

- C# có thiết kế đơn giản, mạnh mẽ và linh hoạt. Sở hữu một cú pháp tương tự như các ngôn ngữ lập trình phổ biến, bao gồm Java và C++, giúp các nhà phát triển đã quen thuộc với các ngôn ngữ này dễ làm quen.

- Một trong những tính năng chính của C# là an toàn kiểu của nó, giúp ngăn ngừa các lỗi lập trình có thể xảy ra với các ngôn ngữ khác. Không những vậy, nó còn hỗ trợ quản lý bộ nhớ tự động thông qua trình thu gom rác, giúp dễ dàng viết mã đáng tin cậy mà không làm rò rỉ bộ nhớ hay gây ra bất kỳ sự cố nào.

- Nhờ những điều đó, nên C# được sử dụng rộng rãi trong việc phát triển các ứng dụng Windows dành cho máy tính để bàn, game, ứng dụng web và các ứng dụng dành cho thiết bị di động. Ngoài ra, nó còn được sử dụng trong phát triển phần mềm doanh nghiệp, với nhiều công ty lớn sử dụng C# để xây dựng các ứng dụng kinh doanh của họ.

- Với tổng quan những điều trên, thì chúng ta dễ dàng nhận thấy C# là một ngôn ngữ mạnh mẽ và linh hoạt, phù hợp với nhiều nhiệm vụ lập trình. Cho dù bạn mới bắt đầu làm quen với lập trình hay là một nhà phát triển có kinh nghiệm, thì C# luôn là một ngôn ngữ tuyệt vời để học và sử dụng khi muốn xây dựng các ứng dụng mạnh mẽ, đáng tin cậy và có thể mở rộng.

**2.2.3. Asp.Net Core**

**- ASP.NET Core là gì ?**

+ Nó là một open-source mới và framework đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile.

Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET Framework. Nó được thiết kế để cung cấp và tối ưu development framework cho những dụng cái mà được triển khai trên đám mây (clound) hoặc chạy on-promise.

Nó bao gồm các thành phần theo hướng module nhằm tối thiểu tài nguyên và chi phí phát triển, như vậy bạn giữ lại được sự mềm giẻo trong việc xây dựng giải pháp của bạn. Bạn có thể phát triển và chạy những ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.

Đồng thời nó đã trở thành một mã nguồn mở. Đây là một thay đổi rất lớn và theo mình là quan trọng nhất của ASP.NET Core. Điều mà trước đây khó có một lập trình viên nào có thể nghĩ đến. Có lẽ đó cũng là một xu thế mà các ngôn ngữ lập trình hiện nay đang hướng tới.

**- ASP.NET Core ra đời trong hoàn cảnh nào?**

**+** Bản phát hành đầu tiên của ASP.NET đã xuất hiện cách đây 15 năm trước, nó là một phần của .NET Framework. Từ đó, hàng triệu lập trình viên đã sử dụng nó để xây dựng những ứng dụng web tuyệt vời, và trên những năm đó Microsoft đã phát triển thêm nhiều tính năng mới.

ASP.NET Core có một số thay đổi kiến trúc lớn, đó là kết quả của việc học hỏi rất nhiều từ các framework module hóa khác. ASP.NET Core không còn dựa trên System.Web.dll nữa. Nó được dựa trên một tập hợp các gói, các module hay cũng được gọi là các Nuget packages.

**- Với ASP.NET Core:**

+ Hợp nhất việc xây dựng web UI và web APIs.

+ Tích hợp những client-side frameworks hiện đại và những luồng phát triển.

+ Hệ thống cấu hình dựa trên môi trường đám mây thật sự.

+ Dependency injection được xây dựng sẵn.

+ HTTP request được tối ưu nhẹ hơn.

+ Có thể host trên IIS hoặc self-host trong process của riêng bạn.

+ Được xây dựng trên .NET Core, hỗ trợ thực sự app versioning.

+ Chuyển các thực thể, thành phần, module như những NuGet packages.

+ Những công cụ mới để đơn giản hóa quá trình phát triển web hiện đại.

+ Xây dựng và chạy đa nền tảng (Windows, Mac và Linux).

+ Mã nguồn mở và tập trung vào cộng đồng.

**- Web UI và web API sửng dụng ASP.NET Core MVC**

+ Bạn có thể tạo ra các ứng dụng web có thể testing theo mô hình MVC (Model-View-Controller).

+ Bạn có thể xây dựng HTTP services cái mà hỗ trợ nhiều định dạng và có đầy đủ hỗ trợ cho nội dung của dữ liệu trả về.

+ Razor cung cấp một ngôn ngữ hiệu quả để tạo Views.

+ Tag Helper cho phép code server side tham gia vào việc tạo và render phần tử HTML.

+ Model Binding tự động ánh xạ dữ liệu từ HTTP request tới tham số của method action.

+ Model Validation tự động thực hiện validate client và server.

### **2.2.4. Angular**

**- Angular là gì?**

Angular là một front-end frameworks mã nguồn mở hay còn được gọi là open source được phát triển bởi Google vào năm 2009 nhằm tạo ra các ứng dụng web hiện đại. Nó sử dụng ngôn ngữ lập trình TypeScript dựa trên JavaScript để loại bỏ mã không cần thiết và đảm bảo các ứng dụng nhẹ hơn và nhanh hơn.

Angular được xem là frameworks [**front - end**](https://aptech.fpt.edu.vn/lap-trinh-front-end.html) mạnh mẽ nhất được các lập trình viên cắt HTML ưa thích sử dụng. Nó là một trong những khung JavaScript được tìm kiếm nhiều nhất hiện nay và nhằm mục đích làm cho việc phát triển front-end trở nên đơn giản và dễ tiếp cận hơn nhiều.

Bên cạnh đó, Angular còn giúp xây dựng các ứng dụng trang đơn (SPA – Single Page Application) tương tác thông qua các tính năng hấp dẫn bao gồm tạo khuôn mẫu, mô-đun hóa, ràng buộc hai chiều, xử lý RESTful API và xử lý AJAX.

Các nhà thiết kế website có thể sử dụng HTML làm ngôn ngữ hình mẫu và thậm chí mở rộng cú pháp HTML để dễ dàng truyền tải các thành phần của ứng dụng. Hơn nữa, sử dụng Angular, bạn cũng sẽ không cần phải dựa vào thư viện của bên thứ ba để xây dựng các ứng dụng động.

Angular sử dụng HTML để xác định giao diện người dùng của ứng dụng. So với JavaScript, HTML là một ngôn ngữ ít phức tạp hơn. Nó cũng là một ngôn ngữ khai báo và trực quan với các lệnh như: ng-app, ng-model, ng-repeat và điều khiển biểu mẫu.

Với sự trợ giúp của Angular, bạn không cần phải đầu tư thời gian vào các luồng chương trình và quyết định xem nội dung nào sẽ tải trước. Angular sẽ xử lý những việc bạn yêu cầu. Các front end có code rất thân thiện với HTML bởi dễ dàng binding data lên nền tảng này.

Angular là một nền tảng giao diện người dùng tuyệt vời không chỉ cho phép phát triển các ứng dụng hiện đại mà còn cho phép tạo các hình ảnh động cao cấp để nâng cao trải nghiệm người dùng. [**API**](https://aptech.fpt.edu.vn/api-la-gi.html) của Angular trực quan đến mức các nhà phát triển có thể tạo ra các đoạn video phức tạp cũng như hoạt ảnh với mã thấp.

Ngoài ra, với các khung kiểm tra đơn vị thông minh như Jasmine và Karma, bạn có thể sửa mã code bị hỏng của mình bất cứ lúc nào. Angular có hơn 11 mô-đun kiểm tra tích hợp để đảm bảo mã không có lỗi.

**2.2.5. Hệ quản trị cơ sở dữ liệu SQL Server**

**2.2.5.1. SQL Server là gì ?**

SQL Server (hay Microsoft SQL Server) là một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft.

SQL Server cung cấp cho người dùng các công cụ và tính năng để quản lý, lưu trữ, xử lý các truy vấn dữ liệu, kiểm soát truy cập, xử lý giao dịch và hỗ trợ tích hợp dữ liệu từ nhiều nguồn khác nhau.

Ngoài ra, SQL Server cũng cung cấp các công cụ để tạo báo cáo, phân tích và quản lý cơ sở dữ liệu trực quan thông qua giao diện người dùng hoặc các script lệnh SQL.

SQL Server được xây dựng dựa trên SQL, một ngôn ngữ lập trình tiêu chuẩn để tương tác với cơ sở dữ liệu quan hệ. SQL Server được liên kết với Transact-SQL hoặc T-SQL, triển khai SQL của Microsoft có bổ sung một tập hợp các cấu trúc lập trình độc quyền.

SQL Server hoạt động độc quyền trên môi trường Windows trong hơn 20 năm. Vào năm 2016, Microsoft đã cung cấp SQL Server trên Linux. SQL Server 2017 ra mắt vào tháng 10 năm 2016 chạy được trên cả Windows và Linux.

SQL Server thường đi kèm với việc thực hiện riêng các ngôn ngữ SQL, T-SQL, ... Cụ thể như sau :

T-SQL là một trong những loại ngôn ngữ thuộc quyền sở hữu của Microsoft và được gọi với cái tên Transact-SQL. Nó thường cung cấp thêm rất nhiều cho các khả năng khai báo biến, thủ tục lưu trữ và xử lý ngoại lệ, ...

SQL Server Management Studio là một loại công cụ giao diện chính cho máy chủ cơ sở của chính dữ liệu SQL, thông thường thì nó hỗ trợ cho cả môi trường 64 bit và 32 bit.

**2.2.5.2. Vì sao sử dụng SQL Server**

Đáng tin cậy và ổn định: SQL Server là một hệ thống quản lý cơ sở dữ liệu (DBMS) rất phổ biến và đáng tin cậy, được phát triển bởi Microsoft. Nó cung cấp hiệu suất ổn định và có khả năng xử lý tải cao.

An ninh dữ liệu: SQL Server cung cấp các tính năng bảo mật mạnh mẽ như xác thực và ủy quyền, mã hóa dữ liệu, kiểm tra ràng buộc, giúp bảo vệ dữ liệu của bạn khỏi các mối đe dọa an ninh.

Hiệu suất cao: SQL Server được tối ưu hóa để đạt hiệu suất cao trên các hệ thống phần cứng khác nhau, giúp xử lý dữ liệu nhanh chóng và hiệu quả.

Dễ quản lý: SQL Server đi kèm với các công cụ quản lý cơ sở dữ liệu mạnh mẽ như SQL Server Management Studio (SSMS), giúp quản lý và duy trì cơ sở dữ liệu một cách dễ dàng. Tích hợp tốt với .NET Framework: SQL Server tương thích tốt với .NET Framework và cung cấp các thư viện và công cụ để phát triển ứng dụng .NET kết nối và tương tác với cơ sở dữ liệu SQL Server.

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 3.1. Khảo sát hệ thống

1. Xác định các tính năng cần thiết

- Giao diện người dùng (UI)

+ Trang chủ (Home Page): Thiết kế hấp dẫn, dễ dàng điều hướng, hiển thị các sản phẩm nổi bật và khuyến mãi.

+ Trang danh mục sản phẩm (Product Listing Page): Hiển thị các sản phẩm theo danh mục, bộ lọc tìm kiếm (theo giá, thương hiệu, kích cỡ, màu sắc), sắp xếp sản phẩm (theo giá, mới nhất, bán chạy nhất).

+ Trang chi tiết sản phẩm (Product Detail Page): Hiển thị chi tiết sản phẩm (hình ảnh, mô tả, giá cả, đánh giá, kích cỡ, màu sắc), tùy chọn thêm vào giỏ hàng.

- Chức năng mua sắm

+ Giỏ hàng (Shopping Cart): Quản lý các sản phẩm đã chọn, cập nhật số lượng, tính tổng giá trị đơn hàng.

+ Thanh toán (Checkout): Hỗ trợ các phương thức thanh toán (thẻ tín dụng, PayPal, chuyển khoản ngân hàng), nhập thông tin vận chuyển, mã giảm giá.

+ Quản lý đơn hàng (Order Management): Theo dõi trạng thái đơn hàng, lịch sử mua hàng, hủy đơn hàng.

- Chức năng người dùng

+ Đăng ký/Đăng nhập (Sign Up/Login): Tạo tài khoản, đăng nhập, đăng xuất.

+ Quản lý tài khoản (Account Management): Cập nhật thông tin cá nhân, quản lý địa chỉ giao hàng, xem lịch sử đơn hàng.

2. Thiết kế và phát triển hệ thống

- Giao diện người dùng để mua hàng

+ Responsive Design: Đảm bảo trang web hiển thị tốt trên thiết bị.

+ Tối ưu hóa SEO: Tối ưu hóa nội dung và cấu trúc trang web để cải thiện thứ hạng trên các công cụ tìm kiếm.

+ Hiệu suất và tốc độ: Tối ưu hóa hiệu suất trang web để giảm thời gian tải trang.

- Quản trị

+ Quản lý cơ sở dữ liệu: Thiết kế cơ sở dữ liệu để lưu trữ thông tin sản phẩm, đơn hàng, người dùng.

+ Xử lý giao dịch: Đảm bảo tính toàn vẹn và bảo mật của các giao dịch mua hàng.

+ API: Xây dựng các API để hỗ trợ giao tiếp giữa front-end và back-end.

## 3.2. Phân tích hệ thống

### **3.2.1. Vai trò của người dùng**

* Hệ thống chia làm 2 nhóm người dùng chính:
* Người mua hàng: những người đăng ký tài khoản để sử dụng với mục đích để mua sản phẩm trên trang web.
* Người quản trị: là người quản lý hệ thống và được truy cập vào các chức năng quản lý người dùng, quản lý web.

**3.2.2. Yêu cầu chức năng**

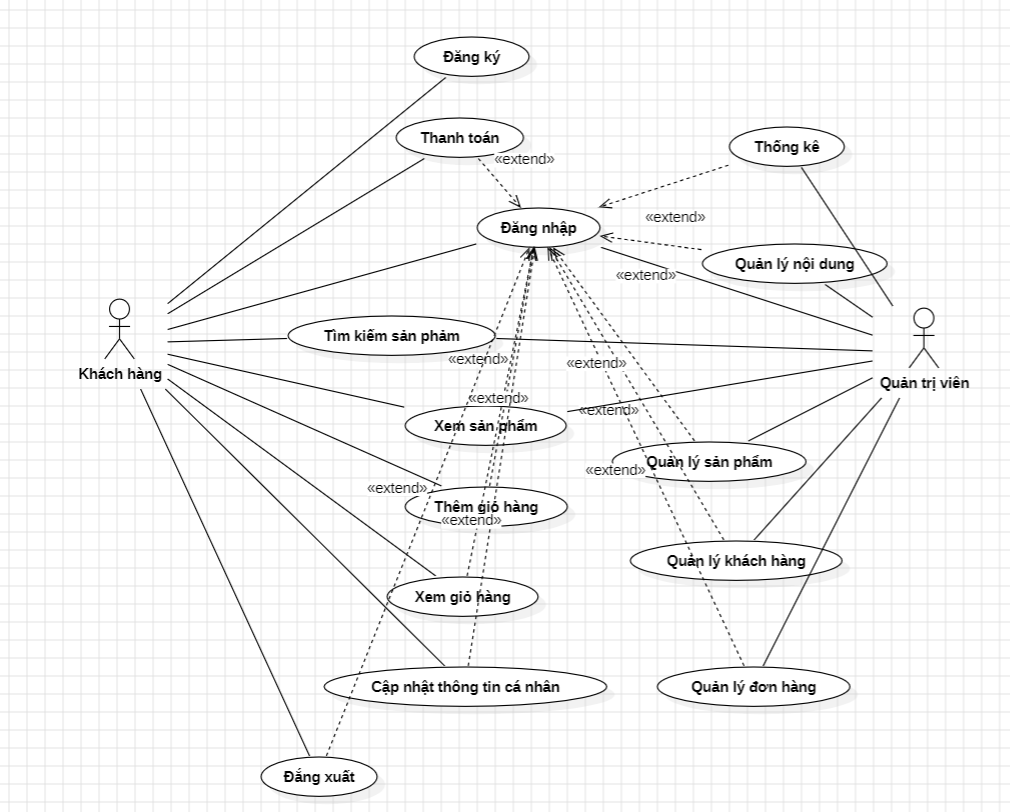
- Người mua hàng:

* + Đăng nhập
  + Đăng ký
  + Đăng xuất
  + Tìm kiếm
  + Xem chi tiết sản phẩm
  + Thêm vào giỏ hàng
  + Xem giỏ hàng
  + Thanh toán
* Quản trị viên:
  + Đăng nhập
  + Đăng xuất
  + Tìm kiếm
  + Quản lý người dùng
  + Quản lý sản phẩm
  + Quản lý thuộc tính sản phẩm
  + Quản lý danh mục
  + Quản lý nhà sản xuât
  + Quản lý quyền
  + Quản lý đơn hàng
  + Thống kê doanh thu

### **3.2.3. Yêu cầu phi chức năng**

* Giao diện trang web được thiết kế dễ sử dụng và có tính thẩm mỹ cao.
* Hiệu năng: tốc độ, khả năng và độ tin cậy của hệ thống.
* Website bảo mật phân quyền để thực hiện các chức năng của hệ thống.

## 3.3. Biểu đồ use case tổng quát

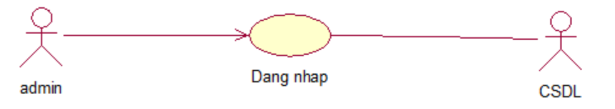
****

*Hình 3.1. Biểu đồ use case tổng quát*

## 3.4. Đặc tả use case

### **3.4.1. Use case đăng nhập**

a) Biểu đồ use case:

******

*Hình 3.2. Biểu đồ use case đăng nhập*

b) Đặc tả use case:

Use case này cho phép người dùng đăng nhập để xác định quyền truy cập vào hệ thống.

* Luồng sự kiện:
  + Luồng cơ bản:

1. Use case này bắt đầu khi người dùng truy cập vào hệ thống.
2. Hệ thống yêu cầu người dùng nhập các thông tin bao gồm: tên đăng nhập và mật khẩu.
3. Người dùng nhập các thông tin: tên đăng nhập và mật khẩu và bấm nút “Đăng nhập”.
4. Hệ thống kiểm tra tên đăng nhập và mật khẩu đã nhập và chuyển hướng sang trang chủ.
5. Use case kết thúc.
   * Luồng rẽ nhánh:
6. Sai tên đăng nhập hoặc mật khẩu: tại bước 3 trong luồng cơ bản nếu người dùng nhập tên đăng nhập hoặc mật khẩu sai, hệ thống hiển thị thông báo lỗi. Người dùng có thể nhập lại hoặc bỏ qua thao tác, khi đó use case kết thúc.
7. Đăng ký: tại bước 3 trong luồng cơ bản, nếu người dùng bấm nút “Đăng ký” use case kết thúc.
8. Quên mật khẩu: tại bước 3 trong luồng cơ bản, nếu người dùng bấm nút “Quên mật khẩu” use case kết thúc.
9. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

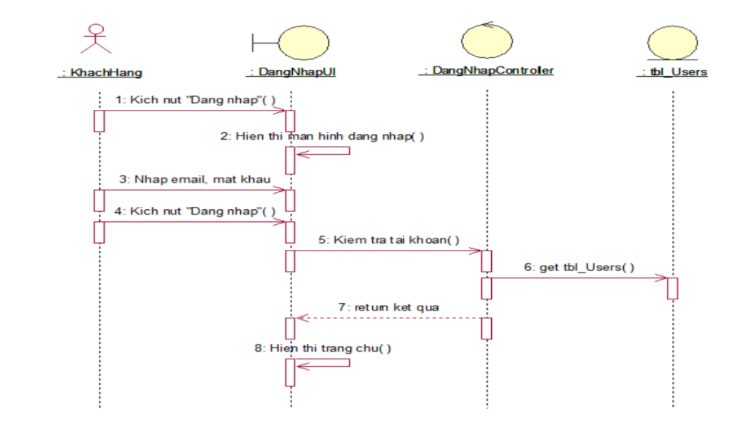
* Hậu điều kiện:

Nếu use case thành công, người dùng sẽ đăng nhập được vào hệ thống. Nếu không trạng thái của hệ thống không thay đổi.

* Điểm mở rộng:

Không có.

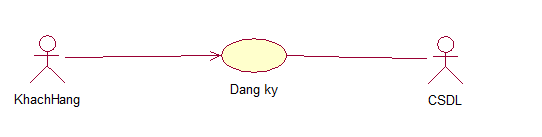
1. Biểu đồ trình tự



*Hình 3.3. Biểu đồ trình tự đăng nhập*

### **3.4.2. Use case đăng ký**

a) Biểu đồ use case:



*Hình 3.4. Biểu đồ use case đăng ký*

b) Đặc tả use case:

Use case cho phép khách đăng ký tài khoản trên hệ thống.

* Luồng sự kiện:
  + Luồng cơ bản:

1) Use case này bắt đầu khi khách hàng click vào link đăng ký trên màn hình đăng nhập. Hệ thống sẽ hiển thị màn hình đăng ký.

2) Khách hàng nhập thông tin trên form đăng ký gồm Email, Password, ConfirmPassword rồi kích nút đăng ký. Hệ thống thêm một bản ghi mới vào trong bảng USER trong cơ sở dữ liệu.

3) Use case kết thúc.

* + Luồng rẽ nhánh:

1) Tại bước 2 trong luồng cơ bản, khi nhập thông tin không thoả mãn, hệ thống thông báo lỗi. Quay lại bước 2.

2) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Không có.

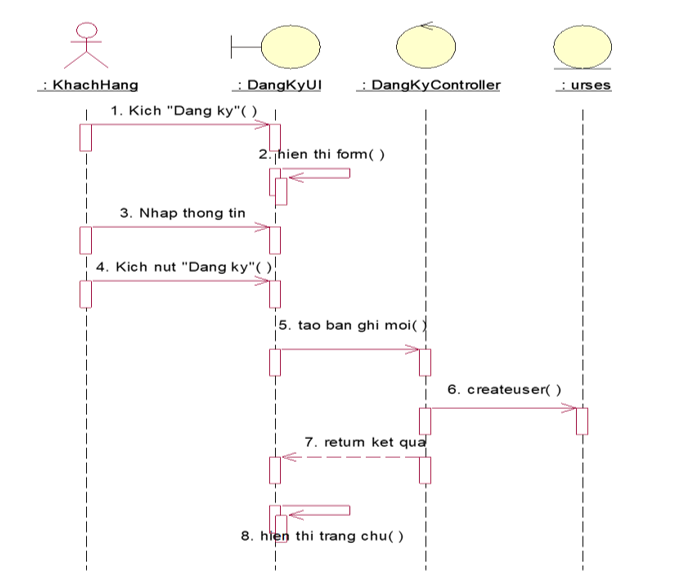
* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

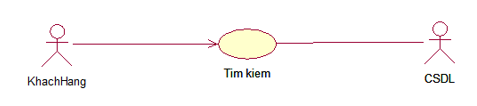
c) Biểu đồ trình tự:



*Hình 3.5. Biểu đồ trình tự đăng ký*

### **3.4.3. Use case tìm kiếm**

a) Biểu đồ use case:



*Hình 3.6. Biểu đồ use case tìm kiếm*

b) Đặc tả use case:

Use case cho phép người dùng tìm kiếm dữ liệu.

* Luồng sự kiện:
  + Luồng cơ bản:

1) Use case này bắt đầu khi người dùng nhập từ khóa tìm kiếm vào ô tìm kiếm. Hệ thống sẽ lọc dữ liệu phù hợp với từ khóa và hiển thị lên màn hình.

2) Use case kết thúc.

* + Luồng rẽ nhánh:

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Đã đăng nhập vào hệ thống.

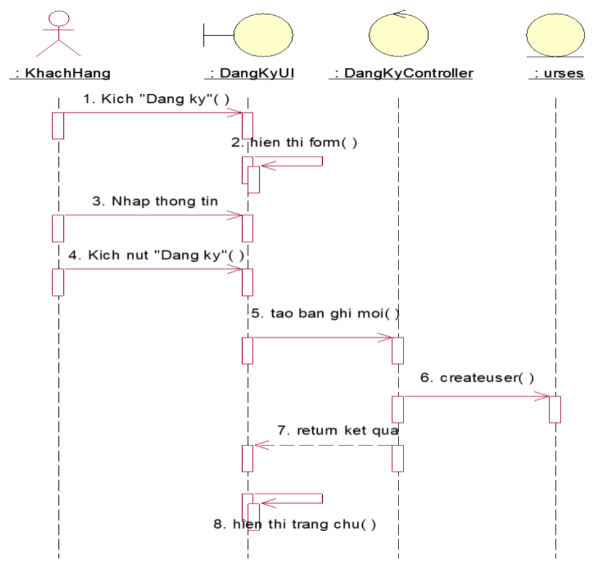
* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

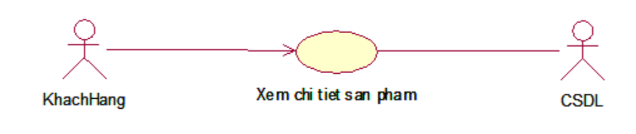
1. Biểu đồ trình tự:



*Hình 3.7. Biểu đồ trình tự tìm kiếm*

### **3.4.4. Use case xem chi tiết sản phẩm**

a) Biểu đồ use case



*Hình 3.8. Biểu đồ use case xem chi tiết sản phẩm*

b) Đặc tả use case:

Use case này cho phép khách hàng xem chi tiết sản phẩm.

- Luồng sự kiện

+ Luồng cơ bản:

1) Use case bắt đầu khi khách hàng kích vào ảnh minh họa của sản phẩm. Hệ thống sẽ lấy từ bảng product các thông tin chi tiết của sản phẩm bao gồm: Tên sản phẩm, Hình ảnh, Mô tả, Kiểu sản phẩm, mã SKU, Mô tả cho SEO

Use case kết thúc

- Luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Đã đăng nhập vào hệ thống.

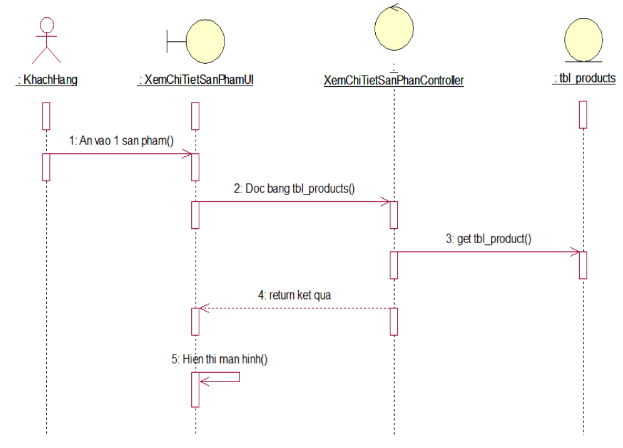
* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

c) Biểu đồ trình tự:



*Hình 3.9. Biểu đồ trình tự xem chi tiết sản phẩm*

**3.5.5. Use case xem sản phẩm theo danh mục**

a) Tên use case:

Xem sản phẩm theo danh mục.

b) Đặc tả use case:

- Luồng sự kiện:

+ Luồng cơ bản:

1) Use case này bắt đầu khi khách hàng kích vào danh mục sản phẩm. Hệ thống sẽ lấy thông tin về các danh mục từ bảng CATEGORIES và hiển thị lên màn hình.

2) Khách hàng kích vào tên một danh mục sản phẩm. Hệ thống lấy thông tin về danh sách các sản phẩm thuộc danh mục đó bao gồm tên, ảnh, mã code sản phẩm đi từ bảng PROPDUCTS và hiển thị lên màn hình.

Use case kết thúc.

+ Luồng rẽ nhánh:

1) Tại bước 2 trong luồng cơ bản, nếu loại tour khách hàng lựa chọn không có tour nào, hệ thống sẽ không hiển thị thông tin và use case kết thúc.

2) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Đã đăng nhập vào hệ thống.

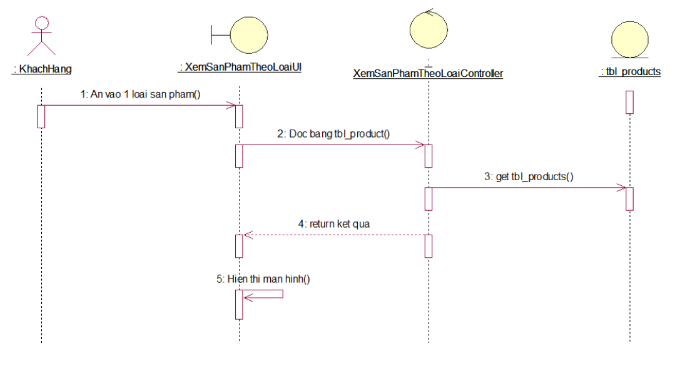
* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

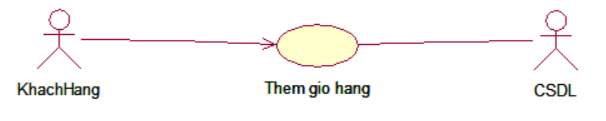
c) Biểu đồ trình tự :



*Hình 3.10. Biểu đồ trình tự xem sản phẩm theo loại*

**3.4.6. Use case thêm giỏ hàng**

a) Biểu đồ use case:

****

*Hình 3.11. Biểu đồ use case thêm giỏ hàng*

b) Đặc tả use case:

Use case này cho phép thêm sản phẩm muốn mua vào giỏ hàng.

- Luồng sự kiện:

+ Luồng cơ bản:

1) Use case bắt đầu khi người dùng kích vào “Thêm vào giỏ hàng” trên mỗi sản phẩm. Hệ thống hiển thị danh sách các sản phẩm trong giỏ hàng bao gồm tên sản phẩm, số lượng, giá, tổng tiền và xóa sản phẩm (nếu muốn xóa).

Use case kết thúc.

+ Luồng rẽ nhánh:

1) Tại bước 1 luồng cơ bản, nếu khách hàng muốn tăng giảm số lượng sản phẩm có thể kích nút “+” và “-” trên cột số lượng của mỗi sản phẩm. Hệ thống sẽ hiển thị số lượng tương ứng sau khi thay đổi.

2) Nếu khách hàng kích vào “Thêm sản phẩm”. Hệ thống sẽ trở về trang chủ.

3) Tại bước 1 luồng cơ bản, nếu khách hàng muốn xóa sản phẩm ra khỏi giỏ hàng, có thể kích nút “Xóa” trên cột xóa của mỗi sản phẩm. Hệ thống cập nhật và hiển thị danh sách sản phẩm trong giỏ hàng sau khi xóa

4) Nếu khách hàng muốn xóa các sản phẩm ra khỏi giỏ hàng có thể kích “Xóa tất cả”. Hệ thống sẽ hiển thị thông báo “Bạn có 0 sản phẩm trong giỏ hàng”.

5) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Đã đăng nhập vào hệ thống.

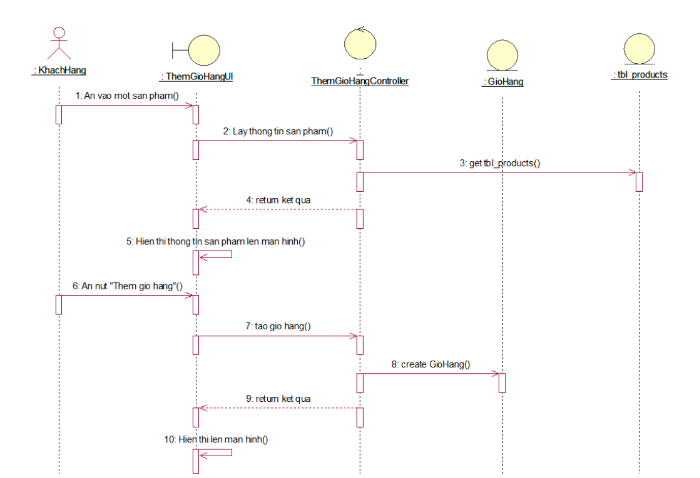
* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

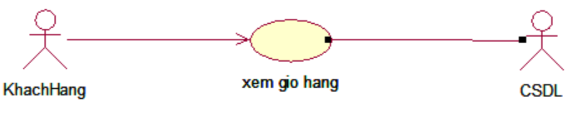
c) Biểu đồ trình tự :



*Hình 3.12. Biểu đồ trình tự thêm giỏ hàng*

### **3.4.7. Use case xem giỏ hàng**

a) Biểu đồ use case:



*Hình 3.13. Biểu đồ use case xem giỏ hàng*

b) Đặc tả use case:

Use case này cho phép khách hàng xem giỏ hàng của mình.

- Luồng sự kiện:

+ Luồng cơ bản:

1) Use case này bắt đầu khi khách hàng kích vào biểu tượng giỏ hàng trên thanh menu. Hệ thống sẽ lấy thông tin về tên sản phẩm, giá bán, số lượng trong cơ sở dữ liệu và hiển thị lên màn hình.

Use case kết thúc.

+ Luồng rẽ nhánh:

1) Nếu giỏ hàng của khách hàng không có sản phẩm nào, hệ thống sẽ không hiển thị thông tin và use case kết thúc.

2) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Đã đăng nhập vào hệ thống.

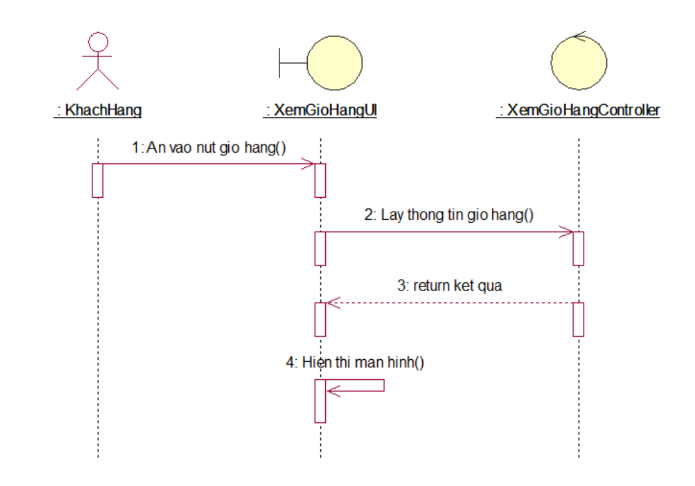
* Hậu điều kiện:

Không có.

* Điểm mở rộng:

Không có.

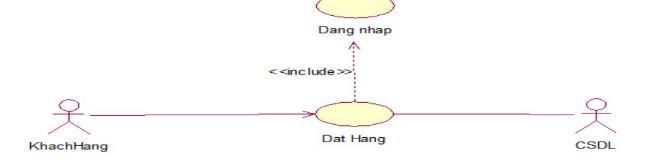
c) Biểu đồ trình tự :



*Hình 3.14. Biểu đồ trình tự xem giỏ hàng*

### **3.4.8. Use case đặt hàng**

a) Biểu đồ use case:



*Hình 3.15. Biểu đồ use case đặt hàng*

b) Đặc tả use case:

Use case này cho phép khách hàng đặt hàng.

- Luồng sự kiện:

+ Luồng cơ bản:

1) Use case bắt đầu khi người dùng kích vào biểu tượng giỏ hàng trên thanh menu. Hệ thống hiển thị hiển thị danh sách các sản phẩm trong giỏ hàng bao gồm tên sản phẩm, số lượng, giá, tổng tiền và xóa sản phẩm (nếu muốn xóa).

2) Người dùng kích vào “Đặt hàng”. Hệ thống hiển thị một biểu mẫu thông tin khách hàng bao gồm họ và tên, email, địa chỉ nhận hàng, số điện thoại, một bảng danh sách sản phẩm muốn mua bao gồm tên sản phẩm, số lượng, giá, tổng tiền và một hóa đơn giá lên màn hình.

3) Người dùng nhập thông tin vào biểu mẫu và kích vào “Gửi đơn hàng”. Hệ thống sẽ hiển thị lên thông báo “Bạn đã đặt đơn hàng thành công”.

Use case kết thúc.

+ Luồng rẽ nhánh:

1) Tại bước 3 luồng cơ bản, nếu khách hàng kích vào “Quay lại giỏ hàng”. Hệ thống sẽ hiển thị danh sách giỏ hàng.

2) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use kết thúc.

* Các yêu cầu đặc biệt:

Không có.

* Tiền điều kiện:

Đã đăng nhập vào hệ thống.

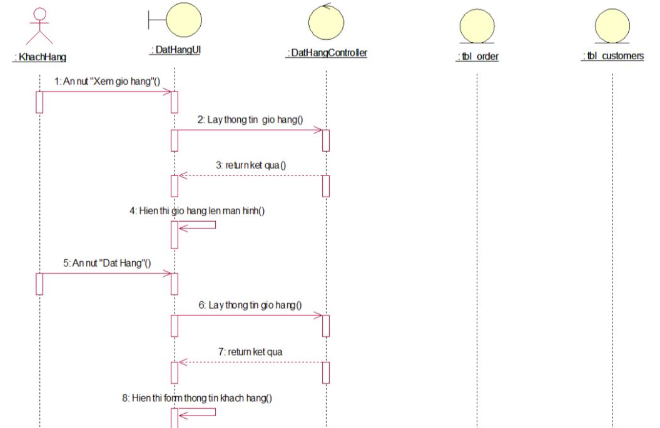
* Hậu điều kiện:

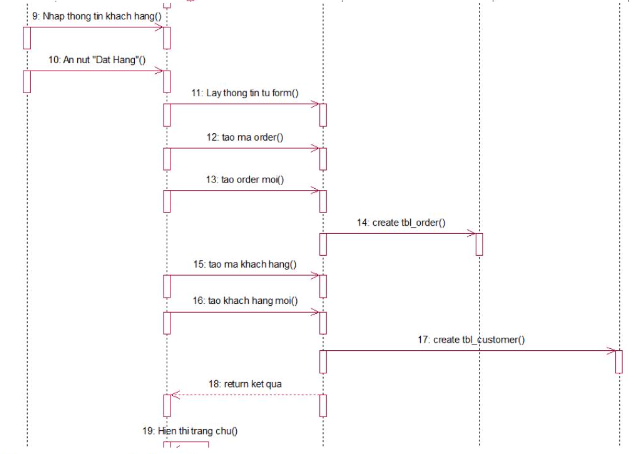
Không có.

* Điểm mở rộng:

Không có.

c) Biểu đồ trình tự :

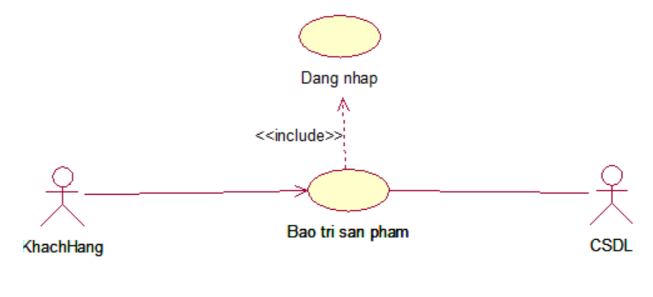




*Hình 3.16. Biểu đồ trình tự đặt hàng.*

**3.4.9. Use case quản lý sản phẩm**

a) Biểu đồ use case:



*Hình 3.17. Biểu đồ use case quản lý sản phẩm*

b) Đặc tả use case:

Use case này cho phép admin xem, thêm, sửa, xóa các sản phẩm.

- Luồng sự kiện:

+ Luồng cơ bản:

1) Use case này bắt đầu khi admin kích vào “Quản lý sản phẩm” trên menu quản trị. Hệ thống sẽ lấy thông tin của sản phẩm gồm: ảnh, tên sản phẩm, loại sản phẩm, kiểu sản phẩm, mã code sản phảm từ bảng PRODUCTS trong cơ sở dữ liệu và hiển thị danh sách các sản phẩm lên màn hình.

2) Xem sản phẩm

Khi admin kích vào nút “Xem” trên một dòng sản phẩm. Hệ thống sẽ lấy thông tin chi tiết về sản phẩm gồm: tên sản phẩm, tên nhà sản xuất, mã code sản phẩm, slug, kiểu sản phẩm, mã SKU, thứ tự sắp xếp, danh mục sản phẩm, Mô tả cho SEO, Mô tả sản phẩm, hình ảnh từ bảng PRODUCTS trong cơ sở dữ liệu và hiển thị chi tiết sản phẩm lên màn hình.

3) Thêm sản phẩm

Admin kích vào nút “Thêm sản phẩm” trên cửa sổ danh sách các sản phẩm. Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho sản phẩm gồm: tên sản phẩm, tên nhà sản xuất, mã code sản phẩm, slug, kiểu sản phẩm, mã SKU, thứ tự sắp xếp, danh mục sản

phẩm, Mô tả cho SEO, Mô tả sản phẩm, hình ảnh, tự sinh một mã sản phẩm mới.

Admin kích vào nút “Thêm”, hệ thống sẽ tạo thêm một bản ghi mới trong bảng PRODUCTS và hiển thị danh sách các sản phẩm sau khi đã được cập nhật.

4) Sửa sản phẩm

Admin kích vào nút “Sửa” trên một dòng sản phẩm. Hệ thống lấy thông tin cũ của sản phẩm được chọn gồm: tên sản phẩm, tên nhà sản xuất, mã code sản phẩm, slug, kiểu sản phẩm, mã SKU, thứ tự sắp xếp, danh mục sản phẩm, Mô tả cho SEO, Mô tả sản phẩm, hình ảnh từ bảng PRODUCTS và hiển thị lên màn hình.

Admin nhập thông tin mới cho sản phẩm và kích nút “Sửa”. Hệ thống sẽ sửa thông tin của sản phẩm được chọn và hiển thị lại danh sách các sản phẩm sau khi đã cập nhật.

5) Xóa sản phẩm

Admin kích vào nút “Xóa” trên một dòng sản phẩm. Hệ thống sẽ hiển thị thông báo xác nhận xóa, admin ấn nút “OK”. Hệ thống sẽ xóa bản ghi về sản phẩm đó trong bảng PRODUCTS và hiển thị lên màn hình danh sách các sản phẩm sau khi đã cập nhật.

Use case kết thúc.

+ Luồng rẽ nhánh:

1) Tại bước 4 hoặc 5 trong luồng cơ bản, nếu admin nhập thông tin không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại, admin có thể nhập lại hoặc hủy bỏ để kết thúc.

2) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use kết thúc.

* Các yêu cầu đặc biệt:

Use case này chỉ cho phép admin thực hiện.

- Tiền điều kiện:

Use case này chỉ cho phép admin thực hiện.

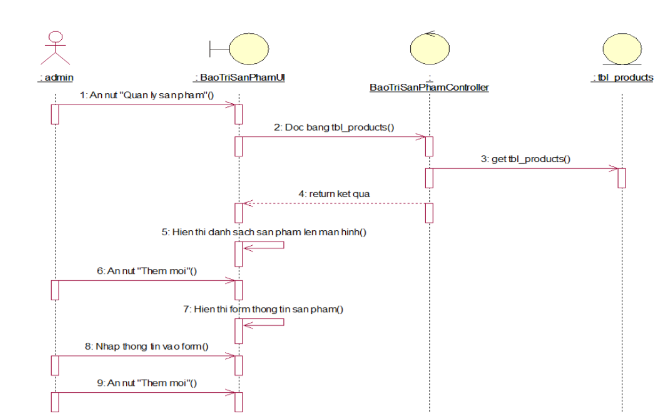
* Hậu điều kiện:

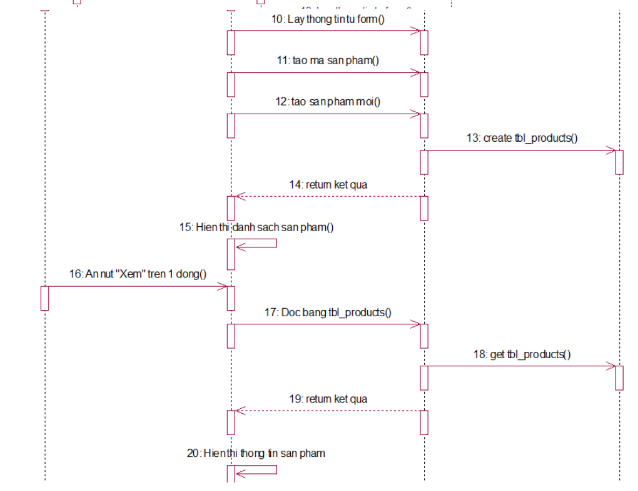
Không có.

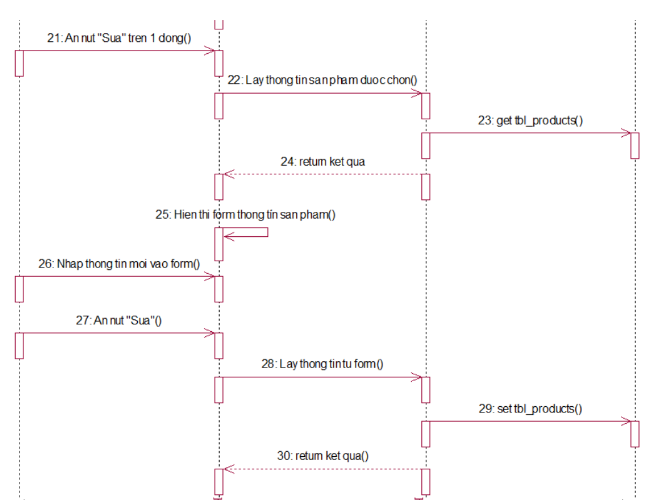
* Điểm mở rộng:

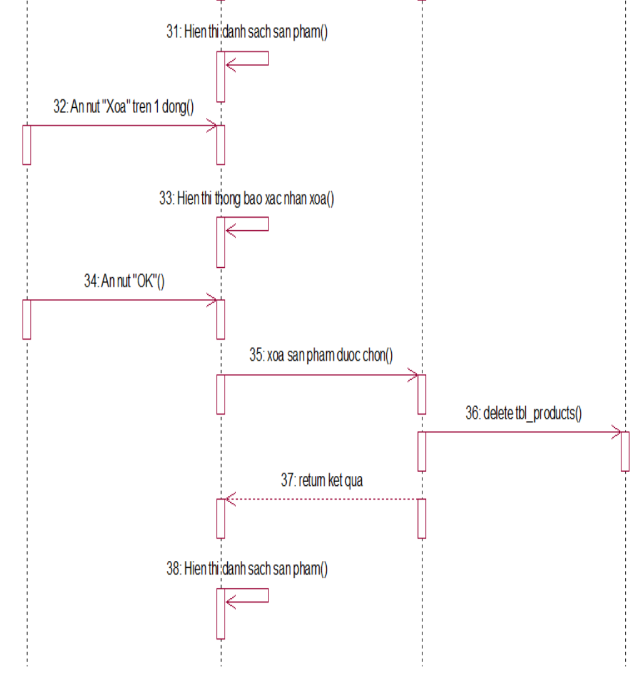
Không có.

c) Biểu đồ trình tự :





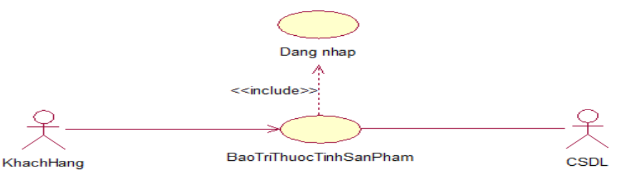




*Hình 3.18. Biểu đồ trình tự quản lý sản phẩm*

**3.4.10. Use case quản lý thuộc tính sản phẩm**

a) Biểu đồ use case:



*Hình 3.19. Biểu đồ use case quản lý thuộc tính sản phẩm*

b) Đặc tả use case:

Use case này cho phép admin xem, thêm, sửa, xóa các thuộc tính sản phẩm.

- Luồng sự kiện:

+ Luồng cơ bản:

1) Use case này bắt đầu khi admin kích vào “Quản lý thuộc tính sản phẩm” trên menu quản trị. Hệ thống sẽ lấy thông tin của thuộc tính sản phẩm gồm: code thuộc tính, kiểu thuộc tính, nhãn thuộc tính, thứ tự, ghi chú từ bảng PRODUCTATTRIBUTE trong cơ sở dữ liệu và hiển thị danh sách các sản phẩm lên màn hình.

2) Xem sản phẩm

Khi admin kích vào nút “Xem” trên một dòng sản phẩm. Hệ thống sẽ lấy thông tin chi tiết về sản phẩm gồm: code thuộc tính, kiểu thuộc tính, nhãn thuộc tính, thứ tự, ghi chú từ bảng PRODUCTATTRIBUTE trong cơ sở dữ liệu và hiển thị chi tiết sản phẩm lên màn hình.

3) Thêm sản phẩm

Admin kích vào nút “Thêm sản phẩm” trên cửa sổ danh sách các sản phẩm. Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho sản phẩm gồm: code thuộc tính, kiểu thuộc tính, nhãn thuộc tính, thứ tự, ghi chú, tự sinh một mã thuộc tính sản phẩm mới.

Admin kích vào nút “Thêm”, hệ thống sẽ tạo thêm một bản ghi mới trong bảng PRODUCTATTRIBUTE và hiển thị danh sách các thuộc tính sản phẩm sau khi đã được cập nhật.

4) Sửa sản phẩm

Admin kích vào nút “Sửa” trên một dòng thuộc tính sản phẩm. Hệ thống lấy thông tin cũ của thuộc tính sản phẩm được chọn gồm: code thuộc tính, kiểu thuộc tính, nhãn thuộc tính, thứ tự, ghi chú từ bảng PRODUCTATTRIBUTE và hiển thị lên màn hình.

Admin nhập thông tin mới cho thuộc tính sản phẩm và kích nút “Sửa”. Hệ thống sẽ sửa thông tin của thuộc tính sản phẩm được chọn và hiển thị lại danh sách các thuộc tính sản phẩm sau khi đã cập nhật.

5) Xóa sản phẩm

Admin kích vào nút “Xóa” trên một dòng thuộc tính sản phẩm. Hệ thống sẽ hiển thị thông báo xác nhận xóa, admin ấn nút “OK”. Hệ thống sẽ xóa bản ghi về thuộc tính sản phẩm đó trong bảng PRODUCTATTRIBUTE và hiển thị lên màn hình danh sách các thuộc tính sản phẩm sau khi đã cập nhật. Use case kết thúc.

Use case kết thúc.

- Luồng rẽ nhánh:

1) Tại bước 4 hoặc 5 trong luồng cơ bản, nếu admin nhập thông tin không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại, admin có thể nhập lại hoặc hủy bỏ để kết thúc.

2) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use kết thúc.

* Các yêu cầu đặc biệt:

Use case này chỉ cho phép admin thực hiện.

- Tiền điều kiện:

Người dùng cần đăng nhập với vai trò admin trước khi có thể thực hiện use case.

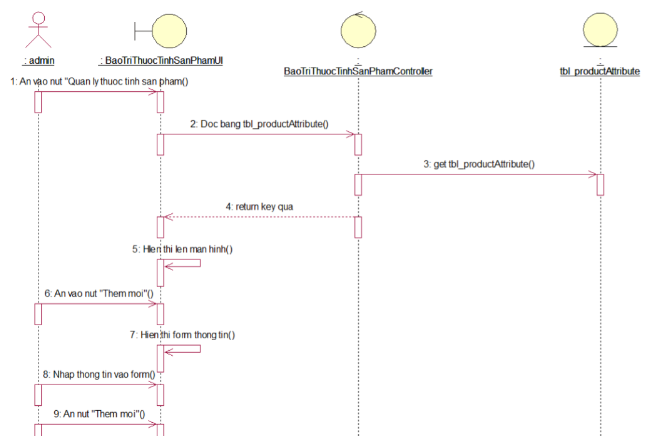
* Hậu điều kiện:

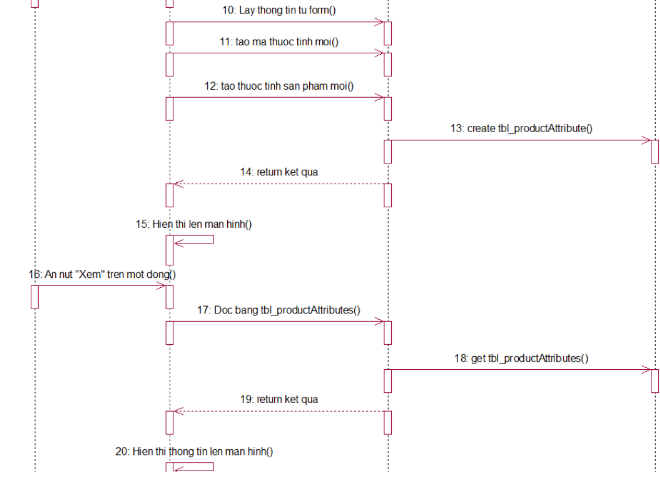
Không có.

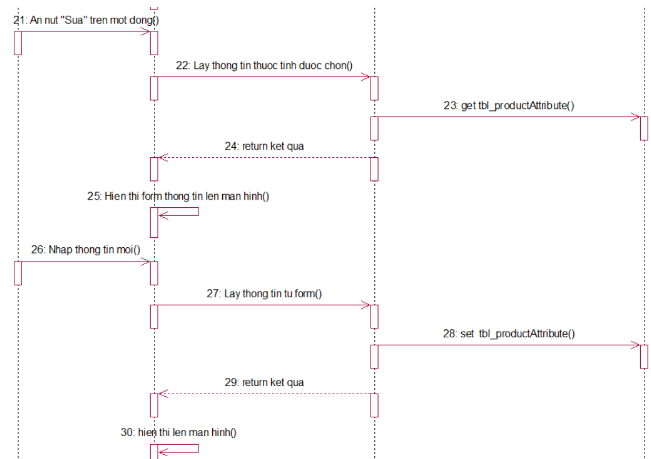
* Điểm mở rộng:

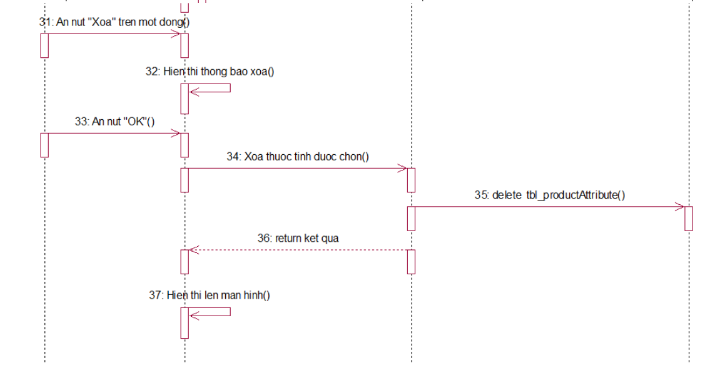
Không có.

c) Biểu đồ trình tự :





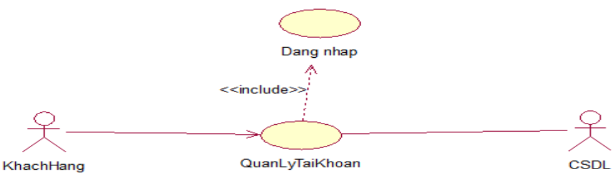




*Hình 3.20. Biểu đồ trình tự quản lý thuộc tính sản phẩm*

### **3.4.11. Use case quản lý người dùng**

a) Biểu đồ use case:



*Hình 3.21. Biểu đồ use case quản lý người dùng*

b) Đặc tả use case:

Use case này cho phép admin sửa, xóa tài khoản đã được đăng ký.

- Luồng sự kiện:

+ Luồng cơ bản:

1) Use case này bắt đầu khi admin kích vào “Quản lý tài khoản” trên menu quản trị, hệ thống sẽ hiển danh sách tài khoản bao gồm mã tài khoản, tên tài khoản, email, kiểu tài khoản lên màn hình.

2) Sửa tài khoản

Admin kích vào nút “Sửa” trên một dòng tài khoản. Hệ thống lấy thông tin cũ của tài khoản được chọn gồm tên tài khoản, email, kiểu tài khoản và hiển thị lên màn hình.

Admin nhập thông tin mới cho tài khoản và kích nút “Sửa”. Hệ thống sẽ sửa thông tin tài khoản được chọn và hiển thị lại danh sách các tài khoản sau khi đã cập nhật.

3) Xóa tài khoản

Admin kích vào nút “Xóa” trên một dòng tài khoản. Hệ thống sẽ hiển thị thông báo xác nhận xóa, admin ấn nút “OK”. Hệ thống sẽ xóa bản ghi về tài khoản đó trong bảng tài khoản và hiển thị lên màn hình danh sách các tài khoản sau khi đã cập nhật.

Use case kết thúc.

+ Luồng rẽ nhánh:

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use kết thúc.

* Các yêu cầu đặc biệt:

Đăng nhập với vai trò là người quản lý

- Tiền điều kiện:

Đã đăng nhập vào hệ thống.

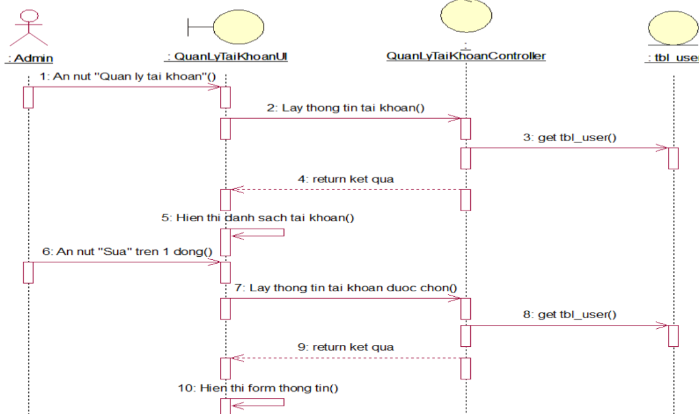
* Hậu điều kiện:

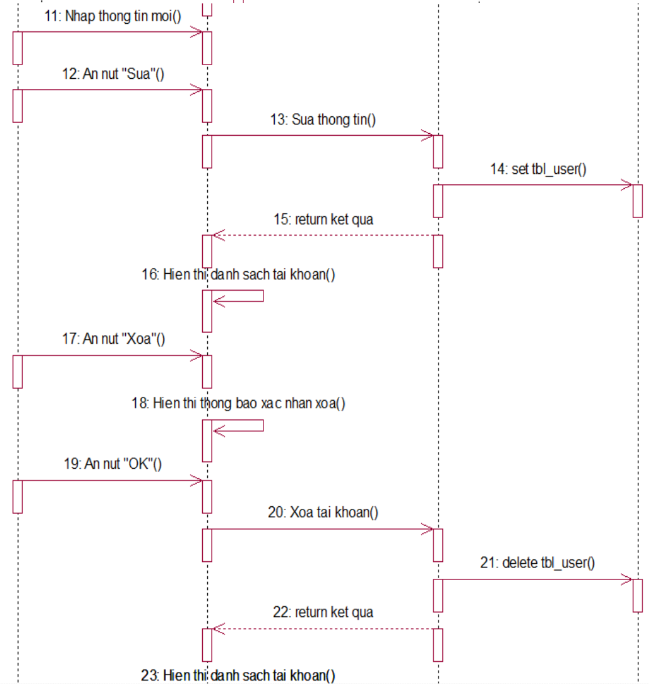
Không có.

* Điểm mở rộng:

Không có.

c) Biểu đồ trình tự :

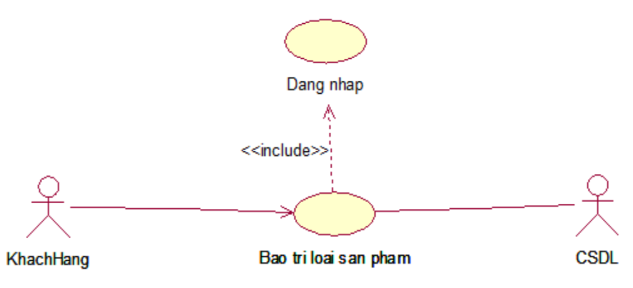




*Hình 3.22. Biểu đồ trình tự quản lý người dùng*

**3.4.12. Use case quản lý danh mục**

a) Biểu đồ use case:



*Hình 3.23. Biểu đồ use case quản lý danh mục*

b) Đặc tả use case:

Use case này cho phép admin xem, thêm, sửa, xóa các loại sản phẩm.

- Luồng sự kiện:

+ Luồng cơ bản:

1) Use case này bắt đầu khi admin kích vào “Quản lý danh mục” trên menu quản trị. Hệ thống sẽ lấy thông tin về tên của các loại sản phẩm từ bảng CATEGORIES và hiển thị lên màn hình.

2) Thêm danh mục

Admin kích vào nút “Thêm mới” trên cửa sổ danh sách các danh mục sản phẩm. Hệ thống hiển thị màn hình yêu cầu nhập thông tin cho danh mục sản phẩm gồm tên danh mục, tự sinh mã danh mục mới.

Admin kích nút “Thêm”, hệ thống sẽ tạo thêm một bản ghi mới trong bảng CATEGORIES và hiển thị danh sách các danh mục sản phẩm sau khi đã được cập nhật.

3) Sửa danh mục

Admin kích vào nút “Sửa” trên một dòng danh mục sản phẩm. Hệ thống lấy thông tin cũ của danh mục sản phẩm được chọn gồm: tên danh mục từ bảng CATEGORIES và hiển thị lên màn hình.

Admin nhập thông tin mới cho danh mục sản phẩm và kích nút “Sửa”. Hệ thống sẽ sửa thông tin của danh mục sản phẩm được chọn và hiển thị lại danh sách các danh mục sản phẩm sau khi đã cập nhật.

4) Xóa danh mục

Admin kích vào nút “Xóa” trên một dòng danh mục sản phẩm. Hệ thống sẽ hiển thị thông báo xác nhận xóa, admin ấn nút “OK”. Hệ thống sẽ xóa bản ghi về danh mục sản phẩm đó trong bảng CATEGORIES và hiển thị lên màn hình danh sách các danh mục sản phẩm sau khi đã cập nhật.

Use case kết thúc.

+ Luồng rẽ nhánh:

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use kết thúc.

* Các yêu cầu đặc biệt:

Use case này chỉ cho phép admin thực hiện.

- Tiền điều kiện:

Người dùng cần đăng nhập với vai trò admin trước khi có thể thực hiện use case.

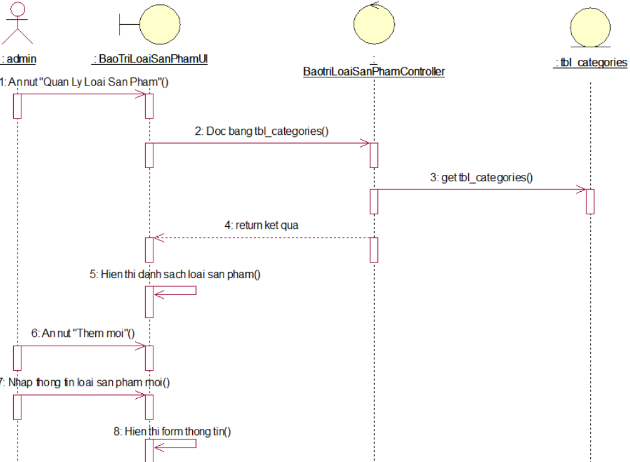
* Hậu điều kiện:

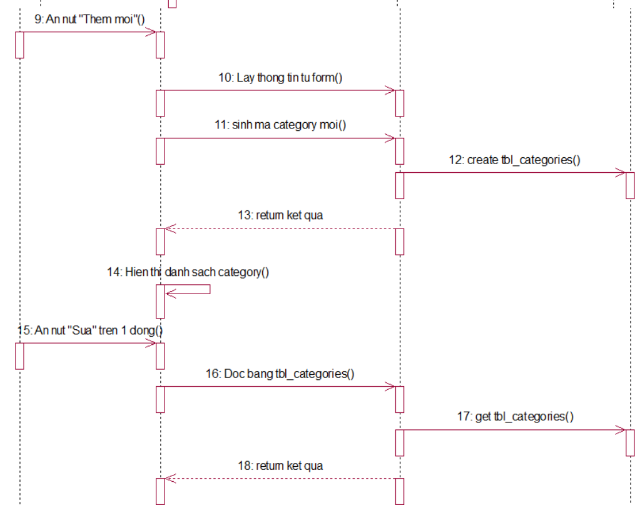
Không có.

* Điểm mở rộng:

Không có.

c) Biểu đồ tuần tự :

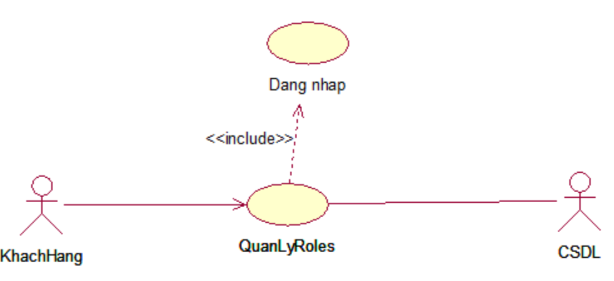




*Hình 3.24. Biểu đồ trình tự quản lý danh mục*

**3.4.13. Use case quản lý quyền**

a) Biểu đồ use case:



*Hình 3.25. Biểu đồ use case quản lý quyền*

b) Đặc tả use case:

Use case này cho phép admin xem, thêm, sửa, xóa quyền.

- Luồng sự kiện:

+ Luồng cơ bản:

1) Use case này bắt đầu khi admin kích vào “Quản lý quyền” trên menu quản trị, hệ thống sẽ hiển danh sách vai trò bao gồm mã quyền, tên quyền lên màn hình.

2) Thêm quyền

Admin kích vào “Thêm mới” phía trên bên phải của trang website. Hệ thống hiển thị một biểu mẫu gồm tên quyền lên màn hình.

Admin kích nút “Thêm”, hệ thống sẽ tạo thêm một bản ghi mới trong bảng ROLES và hiển thị danh sách quyền sau khi đã được cập nhật.

3) Sửa quyền

Admin kích vào nút “Sửa” trên một dòng quyền. Hệ thống lấy thông tin cũ quyền được chọn gồm: tên quyền hiển thị lên màn hình.

Admin nhập thông tin mới cho quyền và kích nút “Sửa”. Hệ thống sẽ sửa thông tin quyền được chọn và hiển thị lại danh sách các quyền sau khi đã cập nhật.

4) Xóa quyền

Admin kích vào nút “Xóa” trên một dòng quyền. Hệ thống sẽ hiển thị thông báo xác nhận xóa, admin ấn nút “OK”. Hệ thống sẽ xóa bản ghi về quyền đó trong bảng ROLES và hiển thị lên màn hình danh sách các quyền sau khi đã cập nhật.

Use case kết thúc.

- Luồng rẽ nhánh:

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use kết thúc.

* Các yêu cầu đặc biệt:

Use case này chỉ cho phép admin thực hiện.

- Tiền điều kiện:

Người dùng cần đăng nhập với vai trò admin trước khi có thể thực hiện use case.

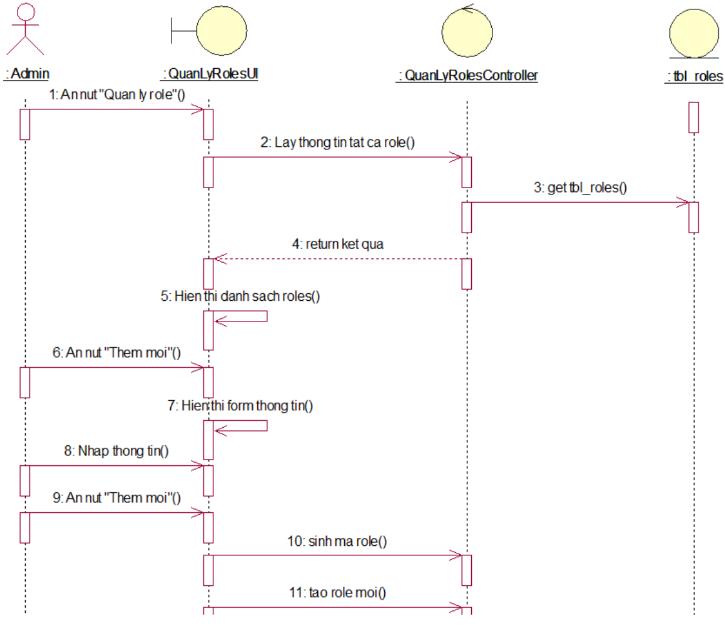
* Hậu điều kiện:

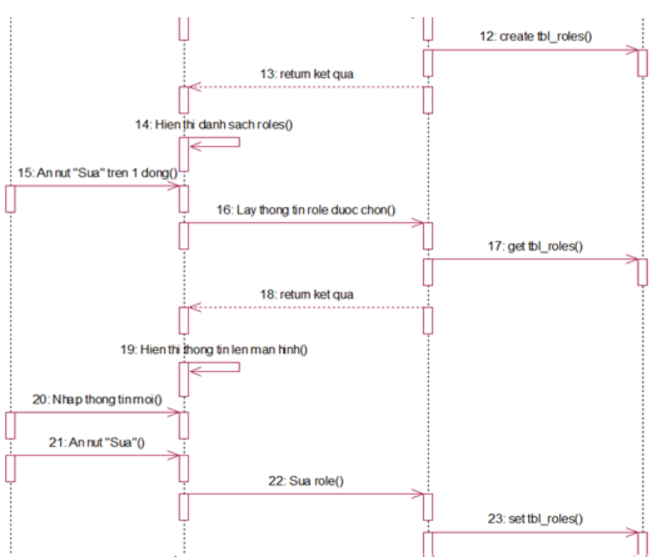
Không có.

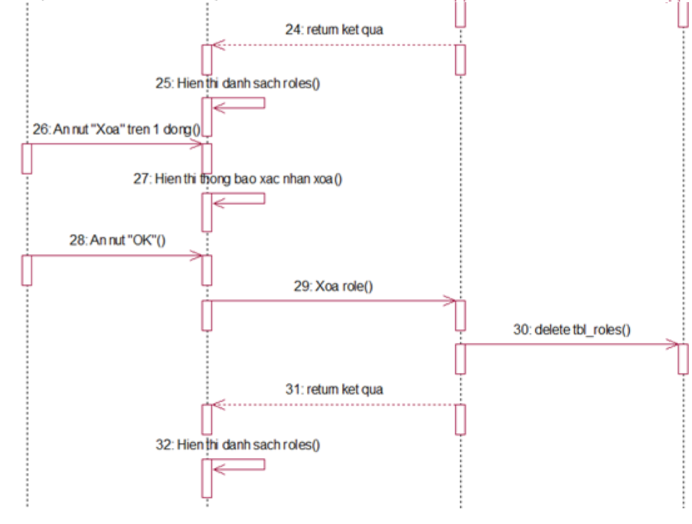
* Điểm mở rộng:

Không có.

c) Biểu đồ tuần tự :







*Hình 3.26. Biểu đồ trình tự quản lý quyền*

**3.5. Mô tả cơ sở dữ liệu**

*Bảng 3.5.1: ProductAttribute gồm thuộc tính sản phẩm*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã thuộc tính |
| Code | Varchar(50) | Code thuộc tính |
| Type | int | Kiểu thuộc tính |
| Label | Nvarchar(50) | Nhãn của thuộc tính |
| SortOrder | int | Thứ tự |
| Visibility | bit | Hiển thị |
| IsActive | bit | Kích hoạt |
| IsRequired | bit | Bắt buộc |
| IsUnique | bit | Duy Nhất |
| Note | Nvarchar(MAX) | Ghi chú |

*Bảng 3.5.2: ProductAttributeInt lưu các thuộc tính kiểu int*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| AttributeId | Guid() | Mã thuộc tính |
| ProductId | Guid() | Mã sản phẩm |
| Value | int | Giá trị |

*Bảng 3.5.3: ProductAttributeText lưu các thuộc tính kiểu text*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| AttributeId | Guid() | Mã thuộc tính |
| ProductId | Guid() | Mã sản phẩm |
| Value | Nvarchar(MAX) | Giá trị |

*Bảng 3.5.4: ProductAttributeDecimal lưu các thuộc tính kiểu decimal*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| AttributeId | Guid() | Mã thuộc tính |
| ProductId | Guid() | Mã sản phẩm |
| Value | Decimal(18,2) | Giá trị |

*Bảng 3.5.5: ProductAttributeDateTime lưu các thuộc tính kiểu datetimne*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| AttributeId | Guid() | Mã thuộc tính |
| ProductId | Guid() | Mã sản phẩm |
| Value | Nvarchar(MAX) | Giá trị |

\

*Bảng 3.5.6: ProductAttributeVarchar lưu các thuộc tính kiểu nvarchar*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| AttributeId | Guid() | Mã thuộc tính |
| ProductId | Guid() | Mã sản phẩm |
| Value | Nvarchar(500) | Giá trị |

*Bảng 3.5.7: Products lưu các sản phẩm:*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| ManufacturerId | Guid() | Mã nhà sản xuất |
| Name | Nvarchar(50) | Tên sản phẩm |
| Code | varchar(50) | Mã code của sản phẩm |
| Slug | varchar(50) | URL-friendly của tên sản phẩm |
| ProductType | int | Kiểu sản phẩm |
| SKU | Varchar(50) | Mã SKU |
| SortOrder | Int | Thứ tự sắp xếp |
| Visibility | bit | Hiển thị |
| IsActive | bit | Kích hoạt |
| CategoryId | Guid() | Id danh mục sản phẩm |
| SeoMetaDescription | Nvarchar(250) | Mô tả meta cho SEO |
| Description | Nvarchar(MAX) | Mô tả |
| ThumbnailPicture | Nvarchar(250) | Hình ảnh |

*Bảng 3.5.8: ProductCategories lưu các danh mục sản phẩm:*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| Name | Nvarchar(50) | Tên danh mục |
| Code | varchar(50) | Mã định danh danh mục |
| Slug | varchar(50) | URL-friendly của tên danh mục |
| SortOrder | Int | Thứ tự sắp xếp |
| CoverPicture | Nvarchar(250) | Hình Ảnh |
| Visibility | bit | Hiển thị |
| IsActive | bit | Kích hoạt |
| ParentId | Guid() | Id Sản phẩm |
| SeoMetaDescription | Nvarchar(250) | Mô tả meta cho SEO |

*Bảng 3.5.9: ProductReviews lưu các đánh giá của sản phẩm*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| ProductId | Guid() | Mã sản phẩm |
| ParentId | Guid() | Mã danh mục |
| Title | Nvarchar(250) | Tiêu đề đánh giá |
| Rating | float | Điểm đánh giá |
| PulishedDate | datetime | Ngày đánh giá |
| Content | Nvarchar(MAX) | Nội dung |
| OrderId | Guid() | Mã đơn hàng |

*Bảng 3.5.10: Manufacturers lưu các nhà sản xuất*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| Name | Nvarchar(50) | Tên nhà sản xuất |
| Code | Varchar(50) | Mã định danh |
| Slug | Varchar(50) | URL-friendly của tên nhà sản xuất |
| CoverPicture | Nvarchar(250) | Hình Ảnh |
| Visibility | bit | Hiển Thị |
| IsActive | bit | Kích Hoạt |
| Country | Nvarchar(MAX) | Địa chỉ |

*Bảng 3.5.11: Tags lưu các thẻ (tag)*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Nvarchar(50) | Mã |
| Name | Nvarchar(50) | Tên thẻ |

*Bảng 3.5.12: ProductTag*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| ProductId | Guid() | Mã sản phẩm |
| TagId | Nvarchar(450) | Mã của thẻ |

*Bảng 3.5.13: Orders lưu các đơn hàng:*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| Code | varchar(50) | Mã định danh đơn hàng |
| Status | int | Trạng Thái |
| PaymentMethod | int | Hình thức thanh toán |
| ShippingFee | int | Phí ship |
| Tax | float | Thuế |
| Total | float | Giá trị đơn hàng trước thuế |
| Subtotal | float | Giá trị đơn hàng sau thuế |
| Discount | float | Tổng giảm |
| GrandTotal | float | Tổng giá trị cuối cùng |
| CustomerName | Nvarchar(50) | Tên khách hàng đặt hàng |
| CustomerPhoneNumber | Nvarchar(50) | Số điện thoại khách hàng |
| CustomerAddress | Nvarchar(250) | Địa chỉ |
| CustomerUserId | Guid() | Mã Khách hàng |

*Bảng 3.5.14: OrderItems quản lí sản phẩm trong đơn hàng:*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| OrderId | Guid() | Mã đơn hàng |
| ProductId | Guid() | Mã sản phẩm |
| SKU | Varchar(50) | Mã SKU |
| Quantity | int | Số lượng |
| Price | float | Giá |

*Bảng 3.5.15: OrderTransactions quản lí giao dịch của đơn hàng:*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| Code | Varchar(50) | Mã code |
| OrderId | Guid() | Mã đơn hàng |
| UserId | Guid() | Mã khách hàng |
| TransactionType | int | Kiểu giao dịch |
| Note | Nvarchar(50) | Ghi chú |

*Bảng 3.5.16: Promotions quản lí các khuyến mãi:*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Nvarchar(50) | Mã |
| Name | Nvarchar(50) | Tên khuyến mãi |
| CouponCode | Varchar(50) | Mã giảm giá |
| RequireUseCouponCode | bit | Yêu cầu nhập mã khuyến mãi |
| ValidDate | datetime | Ngày bắt đầu |
| ExpriedDate | datetime | Ngày hết hạn |
| DiscountMount | float | Tỉ lệ giảm |
| DiscountUnit | int | Đơn vị của giảm giá |
| LimitedUsageTimes | bit | Giới hạn số lần khuyến mãi |
| MaximumDiscountAmount | bigint | Tiền giảm tối đa |
| IsActive | bit | Hoạt động |

*Bảng 3.5.17: PromotionProducts quản lí sản phẩm trong khuyến mãi:*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| ProductId | Guid() | Mã sản phẩm |
| PromotionId | Guid() | Mã khuyến mãi |

*Bảng 3.5.18: PromotionCategories quản lí danh mục trong khuyến mãi:*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| CategoryId | Guid() | Mã danh mục |
| PromotionId | Guid() | Mã khuyến mãi |

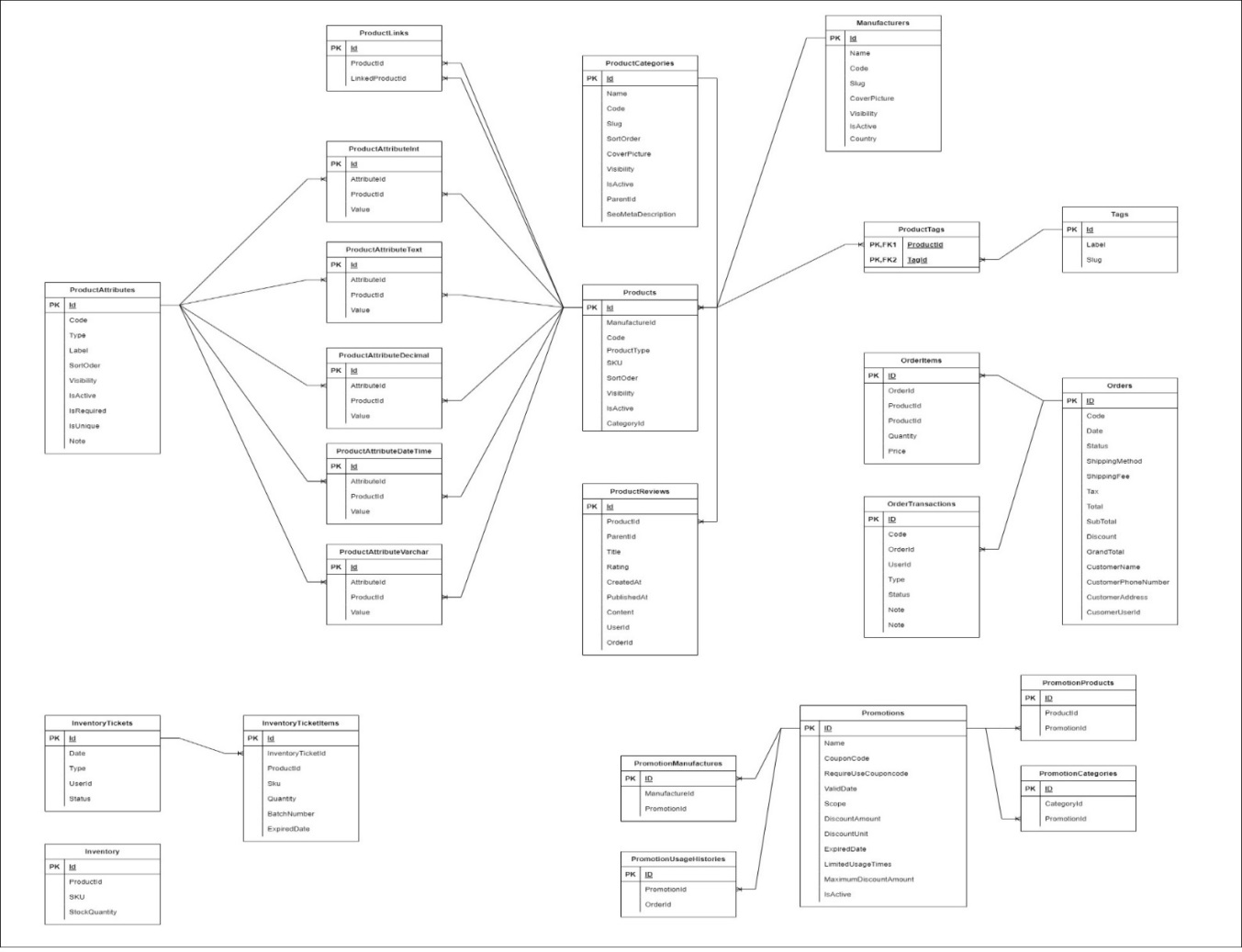
*Bảng 3.5.19: PromotionManufactures quản lí nhà sản xuất trong khuyến mãi:*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| ManufactureId | Guid() | Mã nhà sản xuất |
| PromotionId | Guid() | Mã khuyến mãi |

*Bảng 3.5.20: PromotionUsageHistories quản lí đơn hàng trong khuyến mãi*

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu** | **Mô Tả** |
| Id | Guid() | Mã |
| OrderId | Guid() | Mã đơn hàng |
| PromotionId | Guid() | Mã khuyến mãi |

## 3.6. Mô hình thực thể liên kết



*Hình 3.27. Mô hình thực thể liên kết*

**CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG**

**4.1. Công cụ sử dụng**

### **4.1.1. Microsoft Visual Studio**

* **Microsoft Visual Studio** là một trong những công cụ phát triển phần mềm phổ biến và mạnh mẽ nhất hiện nay. Được Microsoft giới thiệu lần đầu tiên vào năm 1997, đến nay Visual Studio đã trở thành lựa chọn hàng đầu của các nhà phát triển phần mềm chuyên nghiệp.
* **Microsoft Visual Studio** là môi trường phát triển tích hợp (IDE) được thiết kế dành cho giới lập trình viên và các nhà phát triển ứng dụng. Đây là công cụ hỗ trợ phát triển phần mềm mạnh mẽ của Microsoft, cho phép người dùng viết, dịch mã và gỡ lỗi các ứng dụng dựa trên nhiều ngôn ngữ lập trình khác nhau như C++, C#, Visual Basic, Python, JavaScript... Visual Studio bao gồm một trình biên tập mã nguồn, các công cụ gỡ lỗi và xây dựng ứng dụng đa nền tảng. Nó giúp tăng năng suất và hiệu quả công việc cho các lập trình viên.

**Các tính năng nổi bật của Microsoft Visual Studio:**

* Hỗ trợ đa ngôn ngữ lập trình
* Hỗ trợ gỡ lỗi mạnh mẽ
* Tích hợp Git
* Tự động hoàn thành mã (IntelliSense)
* Hỗ trợ đa nền tảng

Với Visual Studio, lập trình viên có thể dễ dàng xây dựng các ứng dụng chạy trên nhiều nền tảng, tiết kiệm thời gian phát triển và chi phí. Đây là một tính năng rất hữu ích và mạnh mẽ của Visual Studio.

### **4.1.2. StarUML**

- StarUML: Là một công cụ mô hình hóa UML nguồn mở và thương mại, được sử dụng để tạo các biểu đồ UML cho việc thiết kế hệ thống phần mềm. StarUML hỗ trợ nhiều loại biểu đồ UML như biểu đồ lớp, biểu đồ trình tự, biểu đồ hoạt động, biểu đồ trường hợp sử dụng, và nhiều hơn nữa.

Phiên bản: Hiện nay, StarUML có phiên bản 3.x là phiên bản thương mại với nhiều tính năng nâng cao, và phiên bản 2.x là phiên bản miễn phí mã nguồn mở.

- Các tính năng chính của StarUML

Hỗ trợ đa biểu đồ: StarUML hỗ trợ hầu hết các loại biểu đồ UML tiêu chuẩn, bao gồm biểu đồ lớp (Class Diagram), biểu đồ trình tự (Sequence Diagram), biểu đồ hoạt động (Activity Diagram), biểu đồ trạng thái (State Diagram), và nhiều biểu đồ khác.

Tích hợp tốt: StarUML hỗ trợ tích hợp với các công cụ khác thông qua các plugin và mô-đun mở rộng.

Giao diện người dùng thân thiện: Cung cấp giao diện đồ họa dễ sử dụng, giúp việc tạo và chỉnh sửa các biểu đồ UML trở nên trực quan và hiệu quả.

Khả năng tùy chỉnh: Cho phép người dùng tùy chỉnh các yếu tố của biểu đồ để phù hợp với nhu cầu cụ thể của dự án.

### **4.1.3. Sql Server Management Studio**

Microsoft SQL Server, thường được viết tắt là **SQL Server**, là một phần mềm được phát triển bởi Microsoft để quản lý và tương tác với dữ liệu theo tiêu chuẩn của hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS). **SQL Server** được tối ưu hóa để xử lý các cơ sở dữ liệu lớn, có thể lên đến hàng terabyte.

* ***Đặc điểm nổi bật:***

1. *Hiệu Suất Đáng Kinh Ngạc:* **SQL Server** được xây dựng để đáp ứng với các yêu cầu của cơ sở dữ liệu lớn với hiệu suất đáng kinh ngạc.
2. *Đa Nhiệm và Tương Tác:* Cung cấp một loạt các công cụ từ giao diện đồ họa đến ngôn ngữ truy vấn SQL, giúp người dùng quản lý dữ liệu một cách hiệu quả.
3. *Tương Thích với Nền Tảng:* **SQL Server** có khả năng tích hợp với nhiều nền tảng phổ biến như ASP.NET, C#, và có thể hoạt động độc lập.

* ***Ngôn ngữ truy vấn và công cụ quản lý:***

**SQL Server** thường sử dụng ngôn ngữ truy vấn SQL cùng với một số tính năng mở rộng độc quyền của Microsoft, được gọi là Transact-SQL (T-SQL). Các thành phần chính bao gồm:

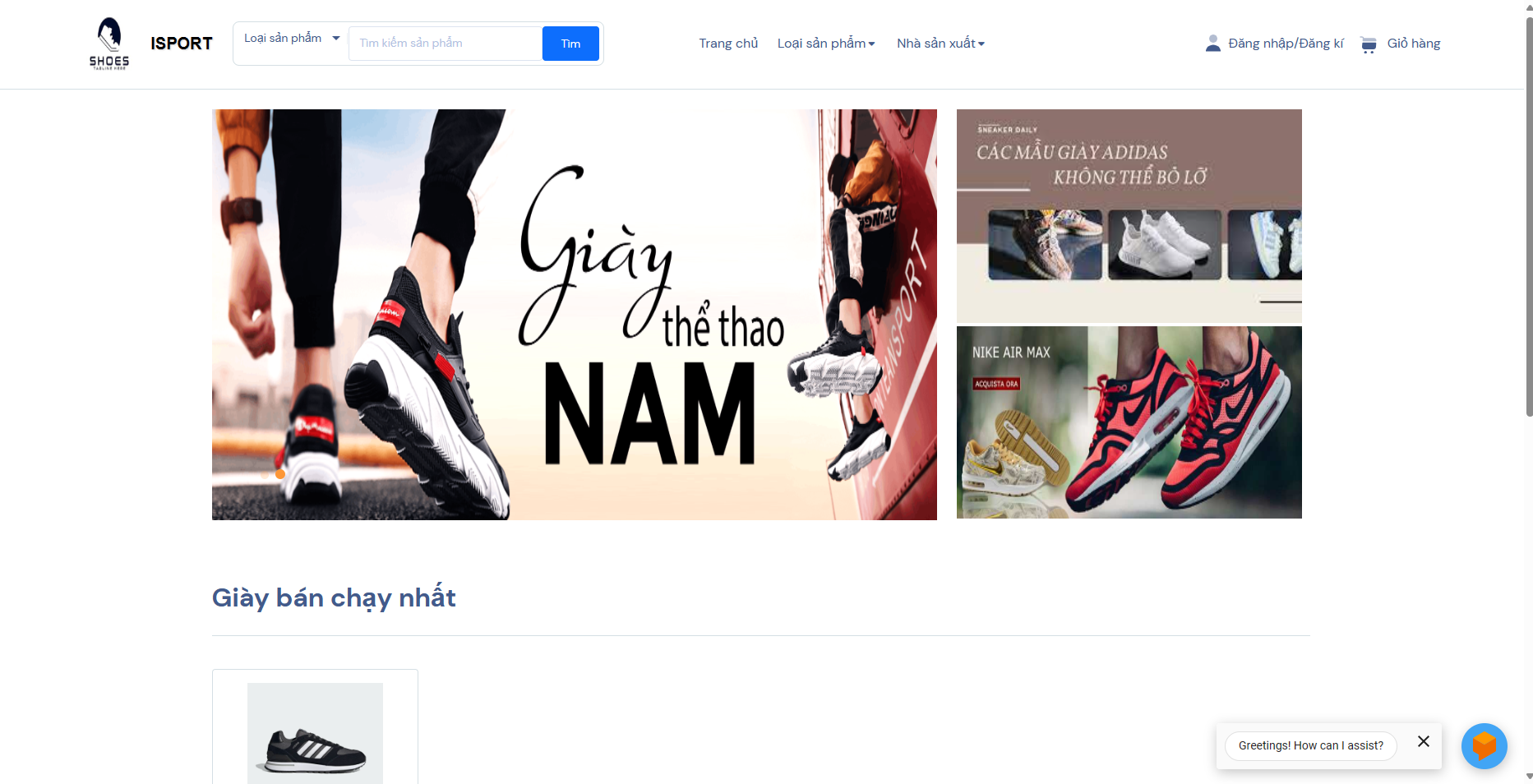
1. *SQL Server Management Studio (SSMS):* Một phần mềm cung cấp giao diện đồ họa và chức năng quản lý cho máy chủ cơ sở dữ liệu SQL.
2. *T-SQL:* Là một mở rộng của SQL, T-SQL mang lại những chức năng nâng cao như khai báo biến, thủ tục lưu trữ, xử lý ngoại lệ và nhiều tính năng mạnh mẽ khác.

**SQL Server** không chỉ là một hệ quản trị cơ sở dữ liệu mạnh mẽ mà còn là một hệ sinh thái đa nhiệm và linh hoạt, phục vụ nhu cầu đa dạng của người dùng.

**4.2. Giao diện trang web**

\*Giao diện trang chủ

- Khi truy cập vào website bán giày I-sport trang chủ sẽ hiện ra, trang chủ sẽ gồm các chức năng: Tìm kiếm, đăng nhập, đăng ký tài khoản. Và gồm các sản phẩm và banner.

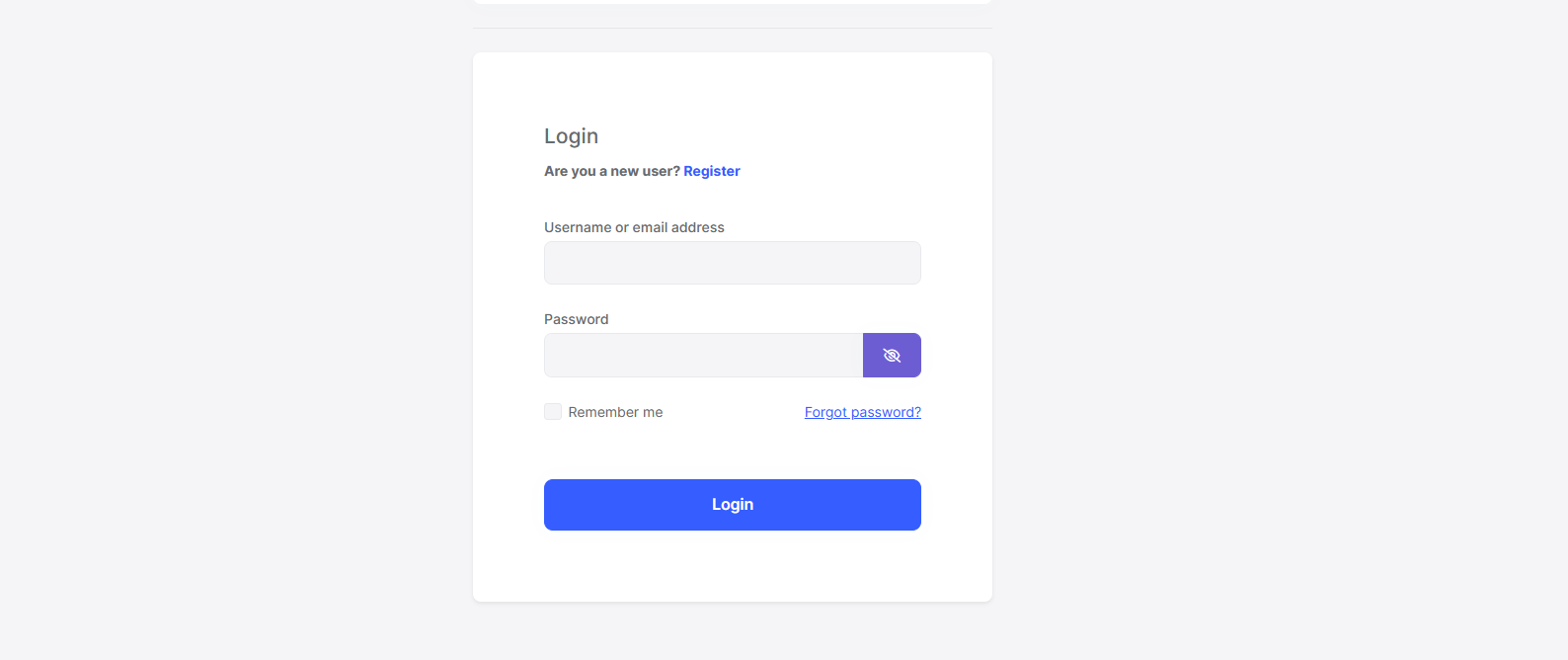
****

*Hình 4.1. Trang chủ web bán giày*

\*Giao diện đăng nhập

- Khi click vào chữ đăng nhập trên trang chủ màn hình sẽ hiện ra nơi nhập tài khoản mật khẩu nếu người dùng đã có tài khoản. Sau khi đăng nhập, khách hàng có thể truy cập và sử dụng tất cả các chức năng cũng như dịch vụ của trang web.

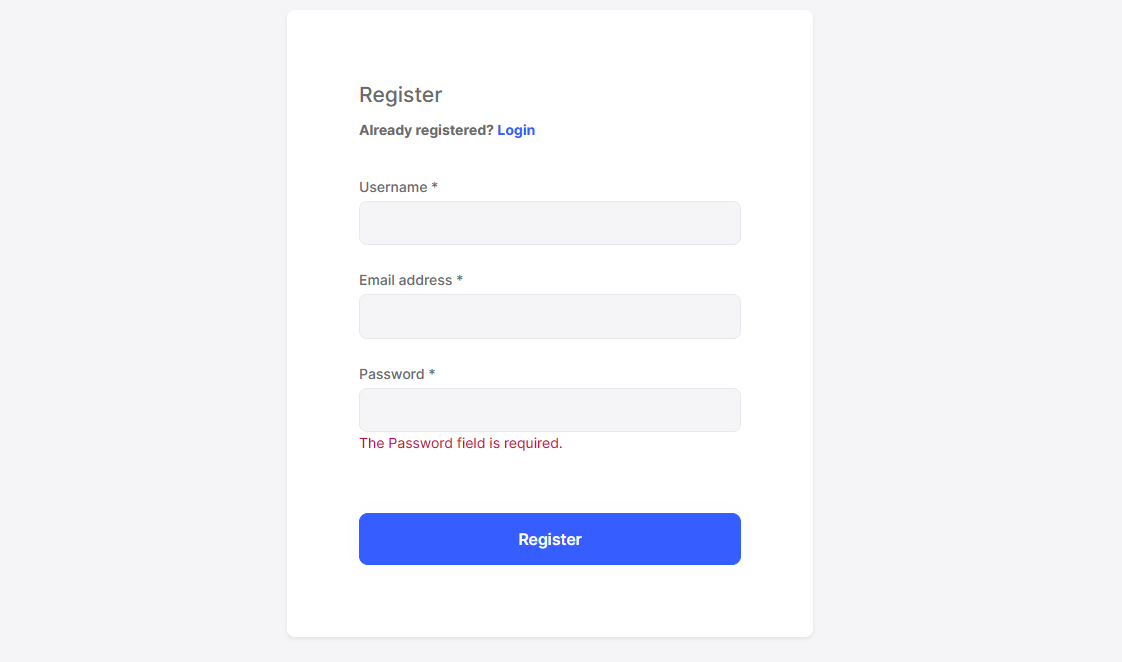
- Nếu người dùng quên mật khẩu có thể click vào forgot password để lấy lại mật khẩu.



*Hình 4.2. Giao diện đăng nhập*

\*Giao diện đăng ký tài khoàn

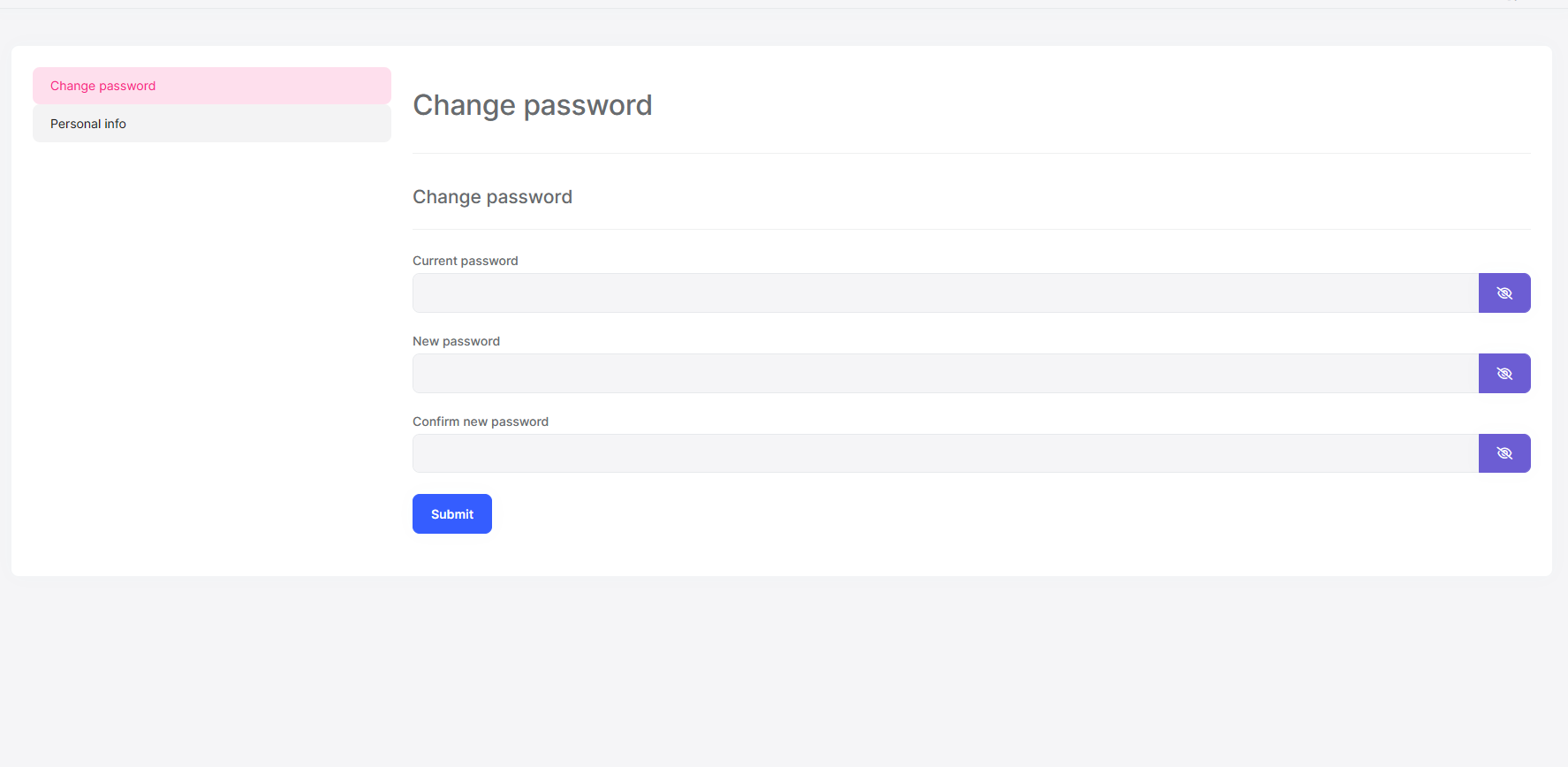
- Nếu người dùng chưa có tài khoản khi click vào register sẽ hiện ra trang đăng ký, ở đây người dùng có thể đăng ký sau đó tiến hành đăng nhập sau khi đăng ký. Nếu khách hàng không đăng ký tài khoản thì chỉ được xem các thông tin về sản phẩm, chi tết sản phẩm,...

**

*Hình 4.3. Giao diện đăng ký*

\*Giao diện đổi mật khẩu

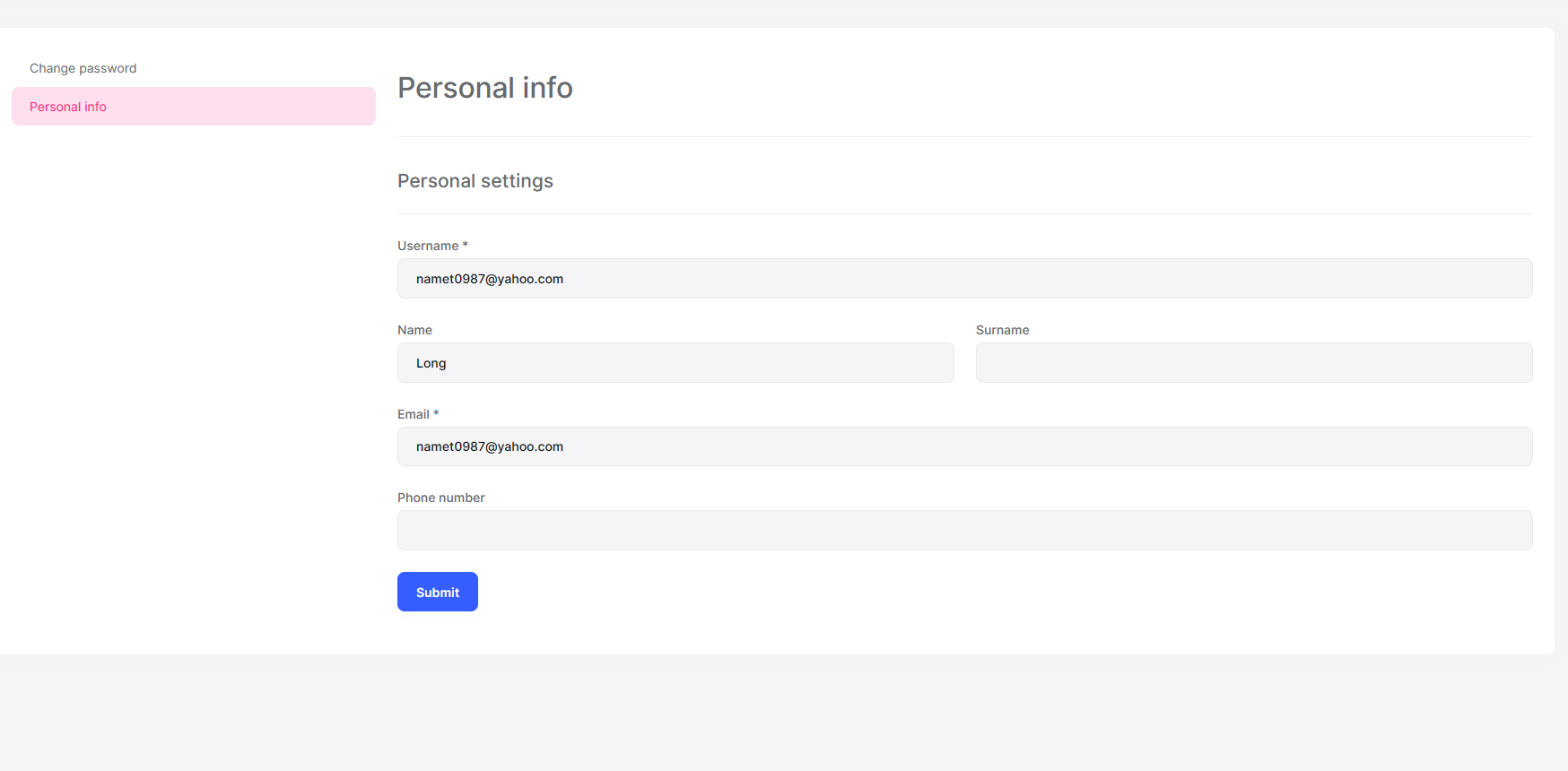
- Cho phép người dùng có nhu cầu muốn đổi mật khẩu.



*Hình 4.4. Giao diện đổi mật khẩu*

*\**Giao diện cập nhật thông tin cá nhân

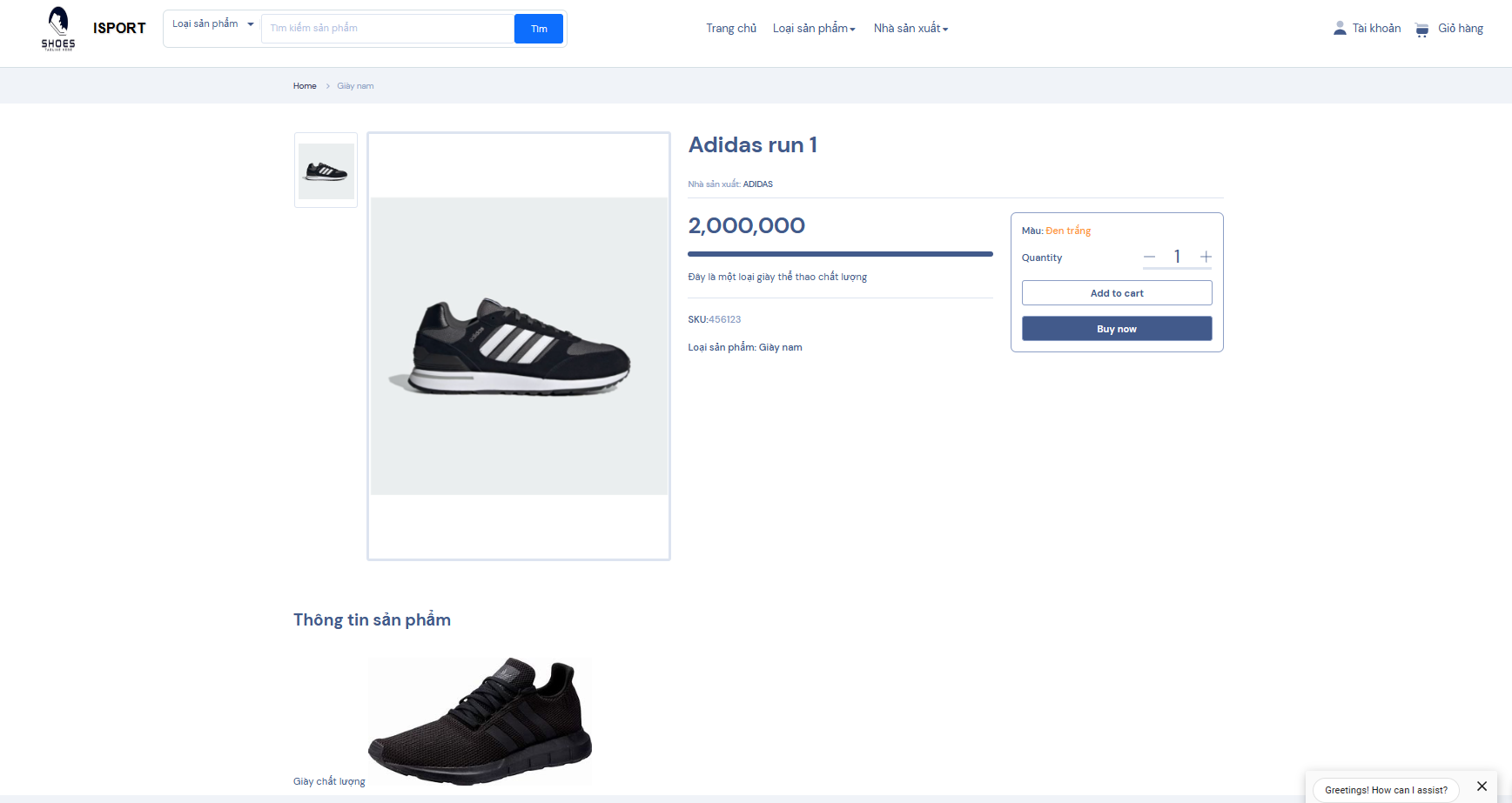
- Cho phép người dùng đổi các thông tin cá nhân như gmail, họ tên, số điện thoại.

**

*Hình 4.5. Giao diện cập nhật thông tin cá nhân*

\*Giao diện xem chi tiết sản phẩm

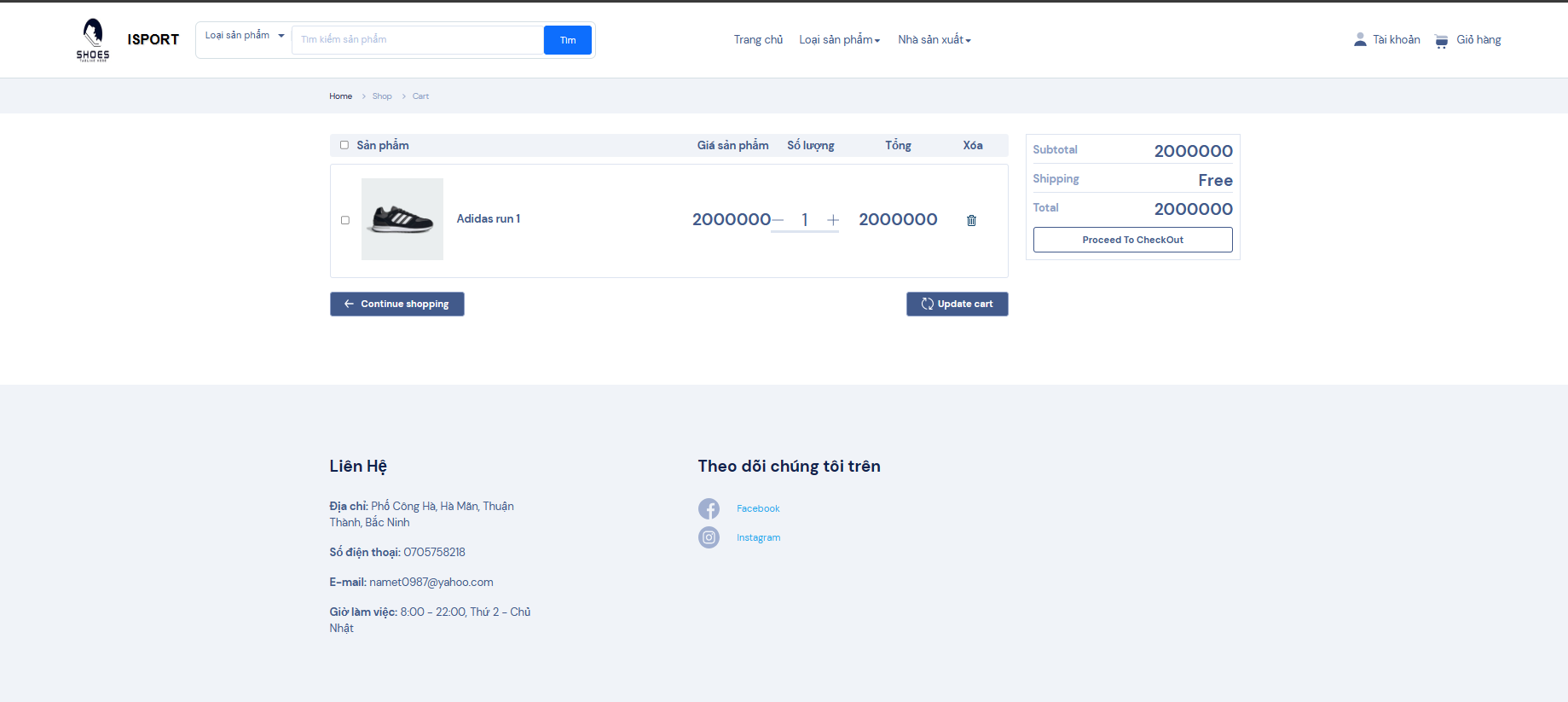
- Khi click vào một sản phẩm, sẽ hiện ra thông tin chi tiết về sản phẩm như: giá tiền, mô tả, chi tiết, chỉnh sửa số lượng mua theo ý muốn.

**

*Hình 4.6. Giao diện xem chi tiết sản phẩm*

\*Giao diện thêm vào giỏ hàng

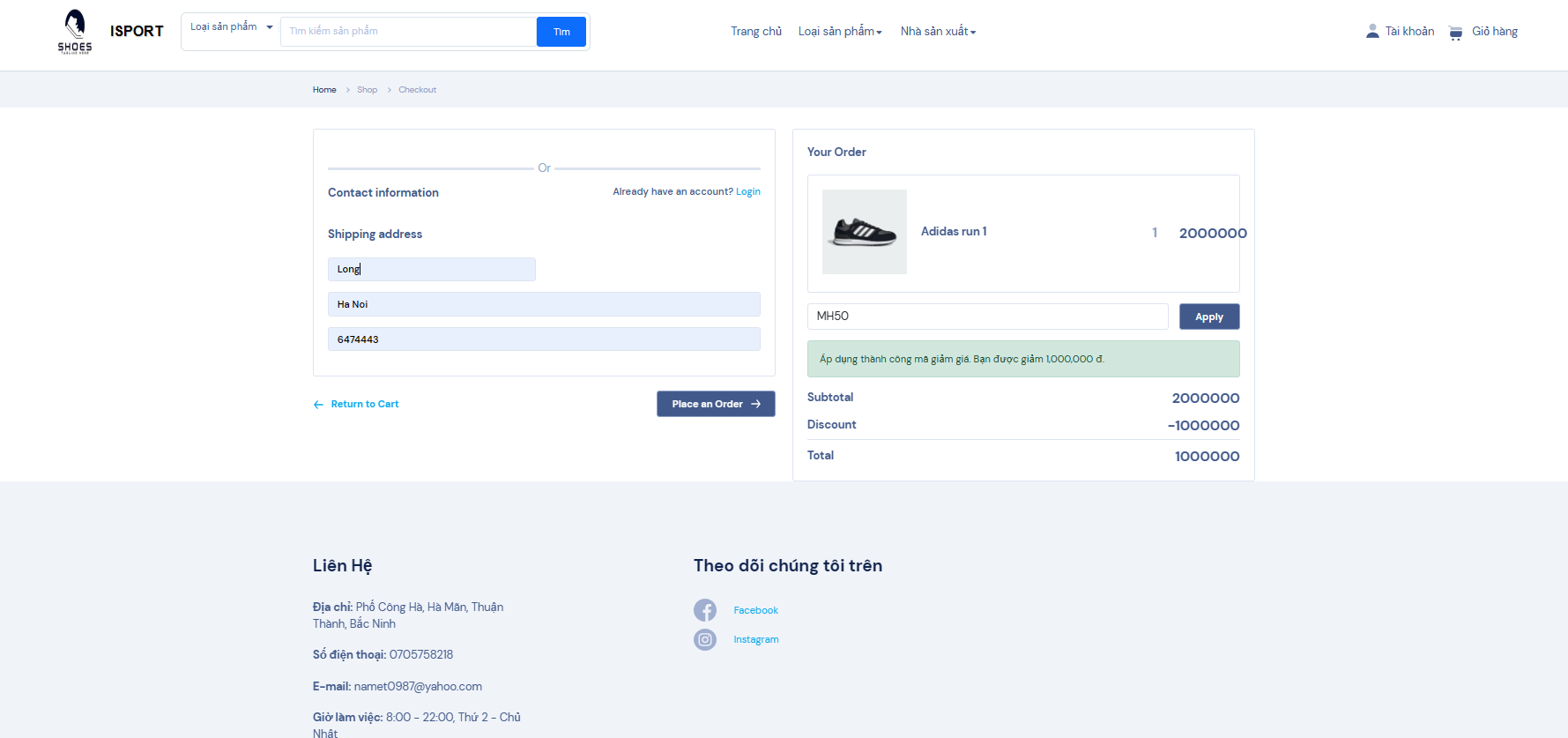
- Khi muốn mua sản phẩm khách hàng bấm vào buy now lúc đó sẽ hiện ra trang thêm vào giỏ hàng. Ở đây, khách hàng có thể chỉnh sửa số lượng mua, xem tổng thành tiền...

**

*Hình 4.7. Giao diện thêm vào giỏ hàng*

\*Giao diện đặt hàng

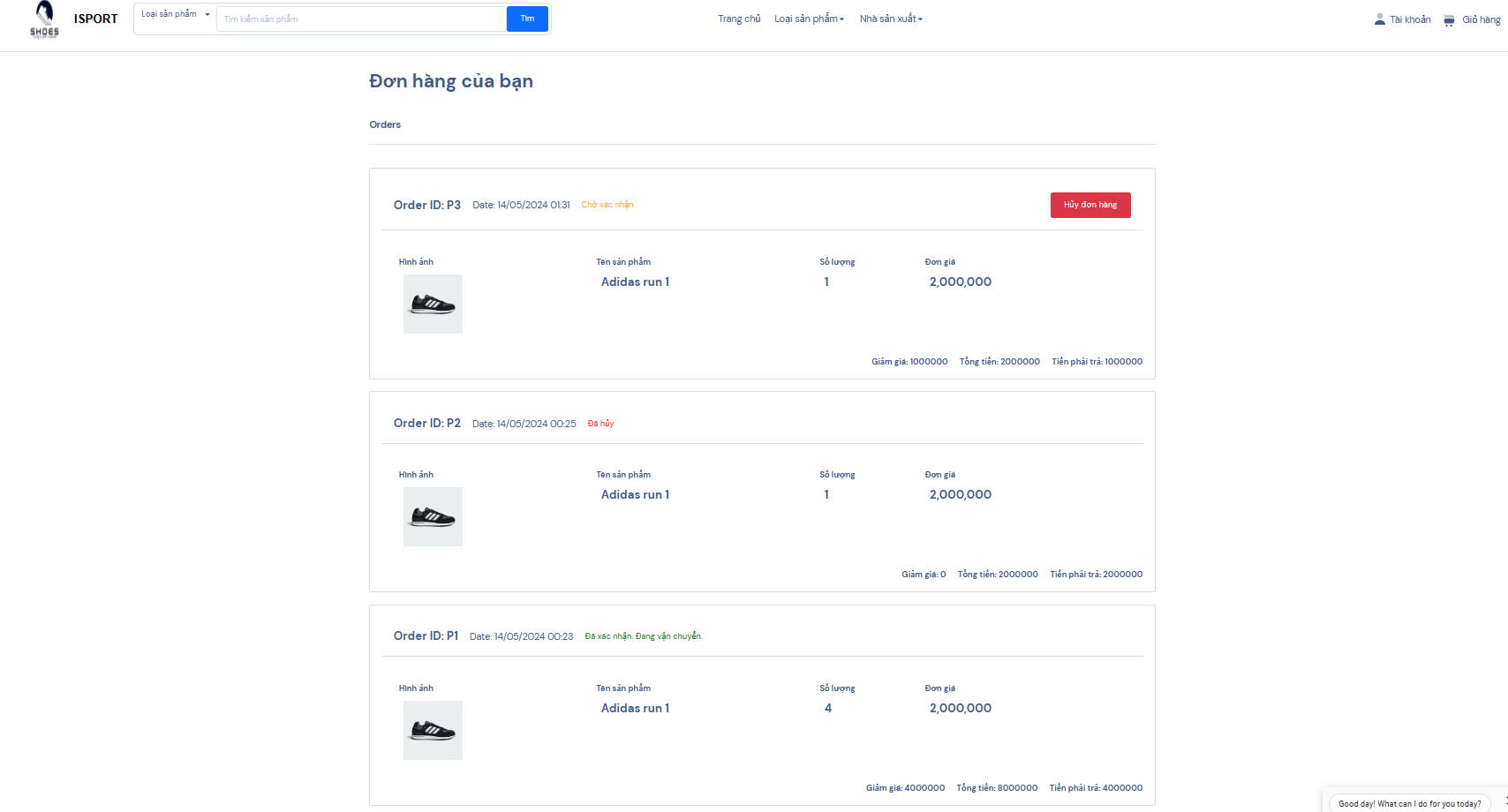
- Sau khi đã chọn mua hàng bên giỏ hàng khách hàng sẽ chuyển đến trang xác nhận đặt hàng để điện thông tin cá nhân người nhận như: Họ tên, số điện thoại, địa chỉ, để xác nhận mua hàng.

**

*Hình 4.8. Giao diện đặt hàng*

\*Giao diện xem đơn hàng

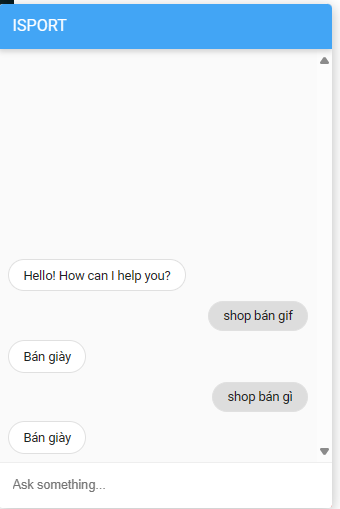
- Khi khách hàng muốn xem lại các đơn hàng đã đặt, khách hàng chỉ cần click vào mục xem đơn hàng. Ở đây sẽ hiện thị các sản phẩm, giá tiền, số lượng, tình trạng đơn hàng mà khách hàng đã đặt. Khách hàng có thể hủy đơn hàng tại đây.

**

*Hình 4.9. Giao diện xem đơn hàng*

*\**Giao diện chatbot

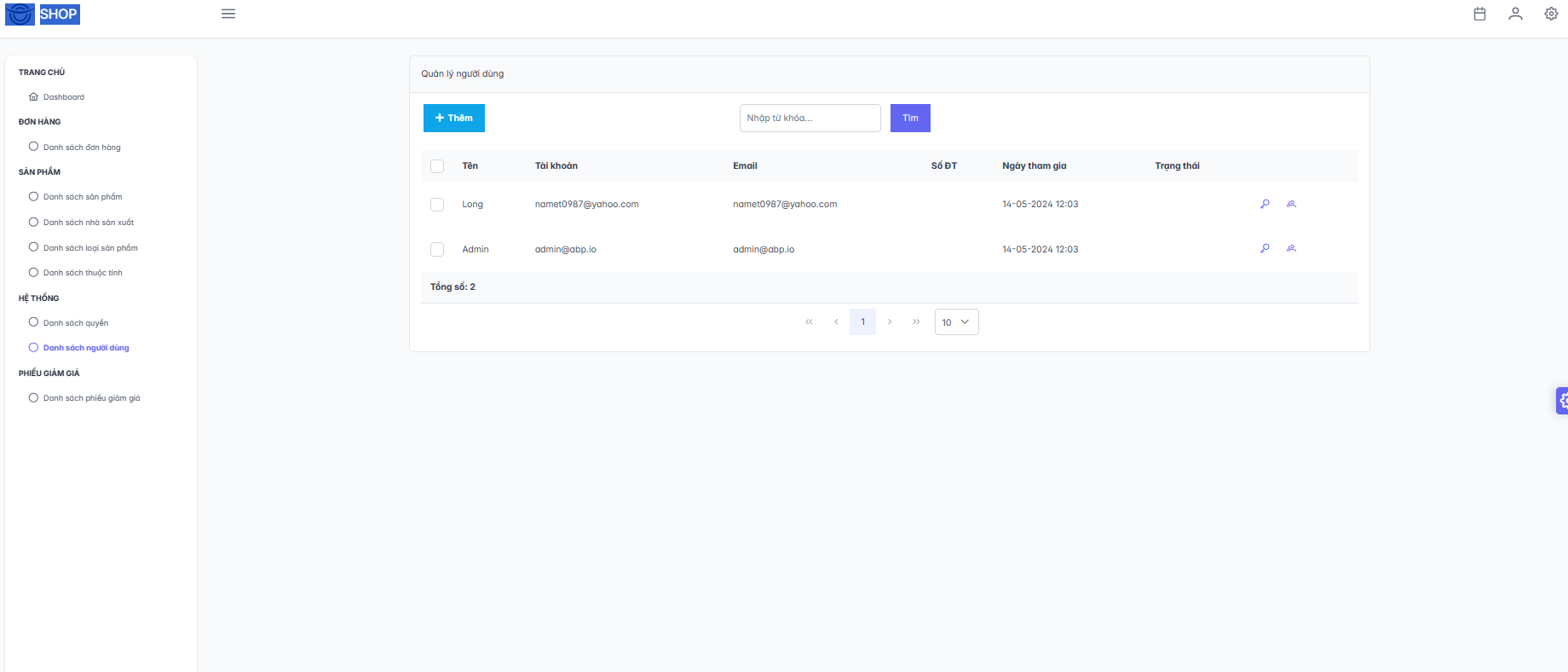
- Ở đây khách hàng có thể hỏi những câu hỏi cơ bản liên quan đến cửa hàng và sản phẩm của cưat hàng

**

*Hình 4.10. Giao diện chatbot*

\*Giao diện quản lý người dùng

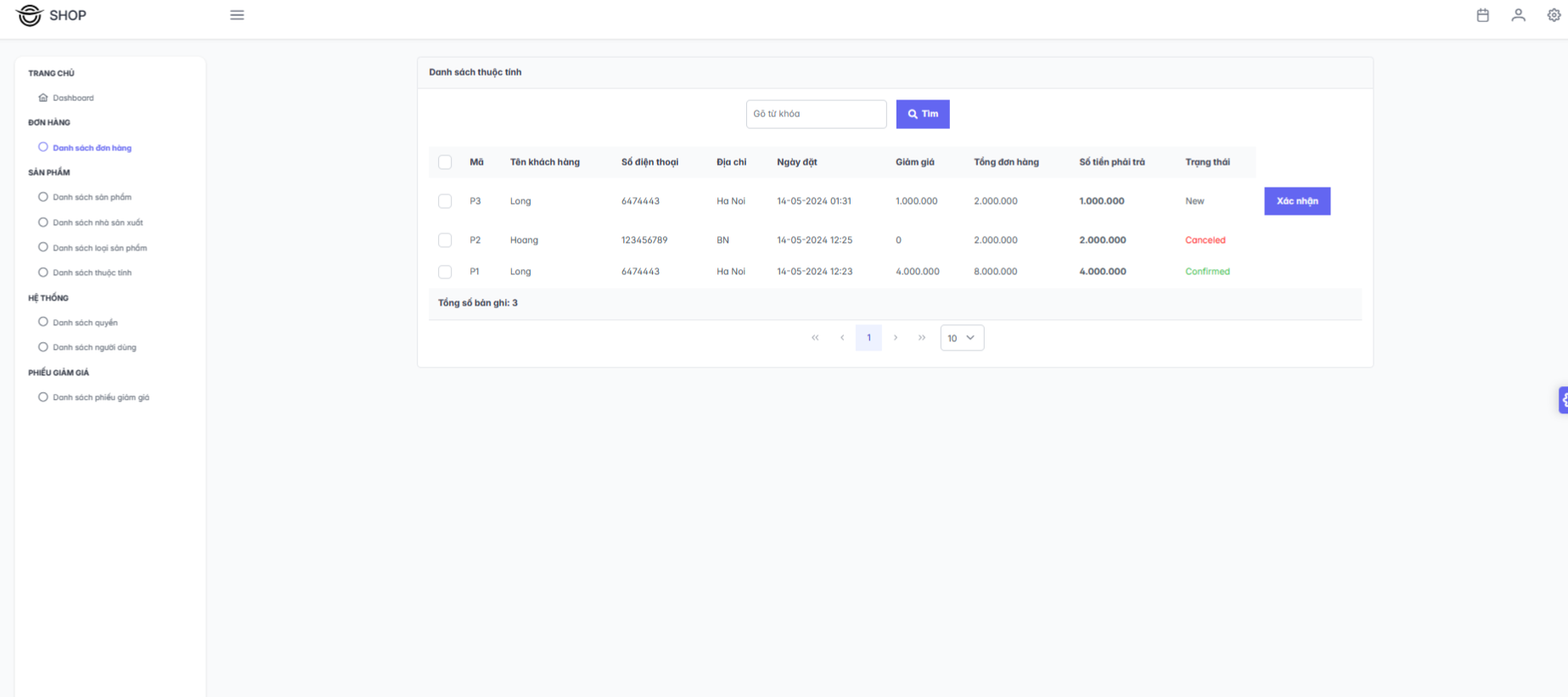
*-* Sau khi đã đăng nhập là thành viên quản trị của website thì quản trị có thể thấy được giao diện này. Quản trị có thể quản lý các tài khoản, mật khẩu, thông tin cá nhân của các khách hàng.

**

*Hình 4.11. Giao diện quản lý người dùng*

\*Giao diện quản lý đơn hàng

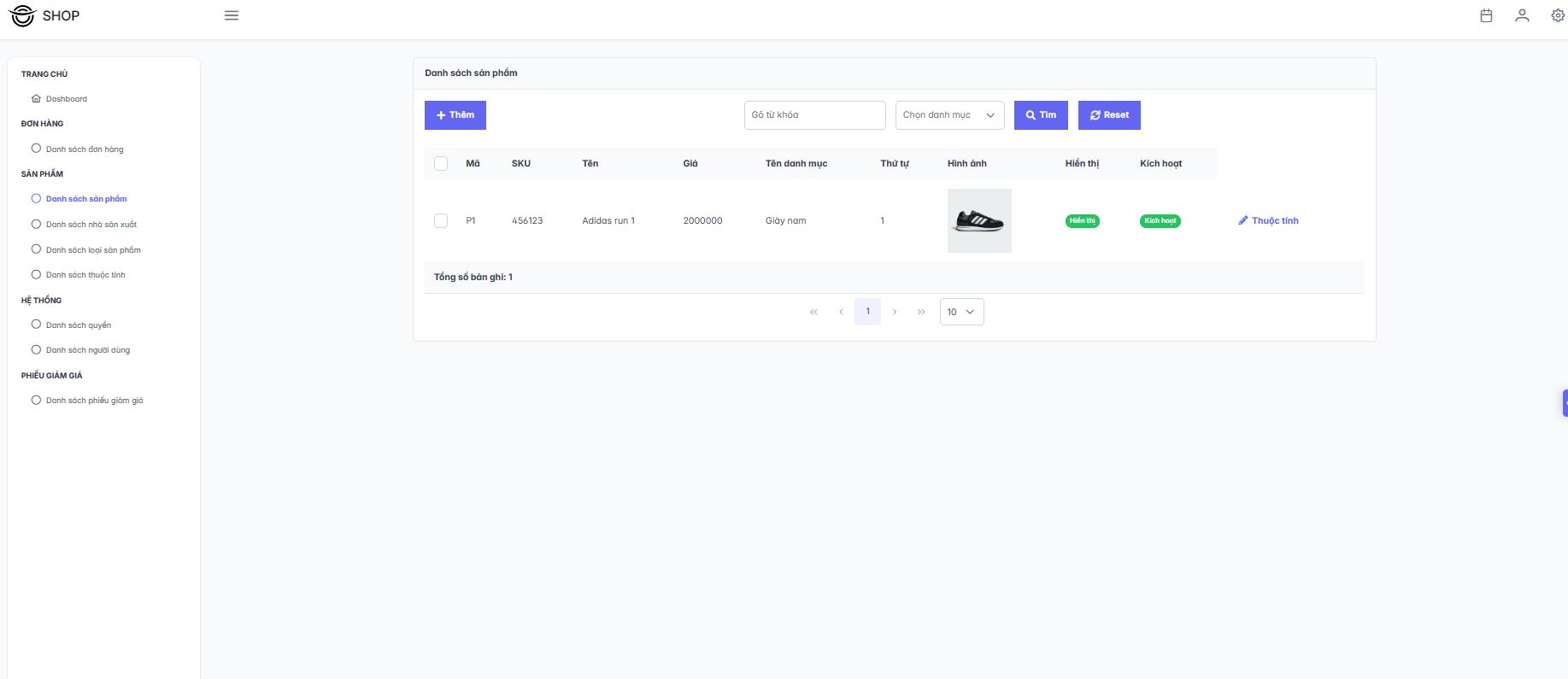
*-* Quản trị có thể lấy thông tin đơn hàng gồm mã đơn hàng ngày đặt hàng và tính trạng của đơn hàng. Có thể xác nhận đơn hàng hoặc xóa đơn hàng nếu muốn.

**

*Hình 4.12. Giao diện quản lý đơn hàng*

*\**Giao diện quản lý sản phẩm

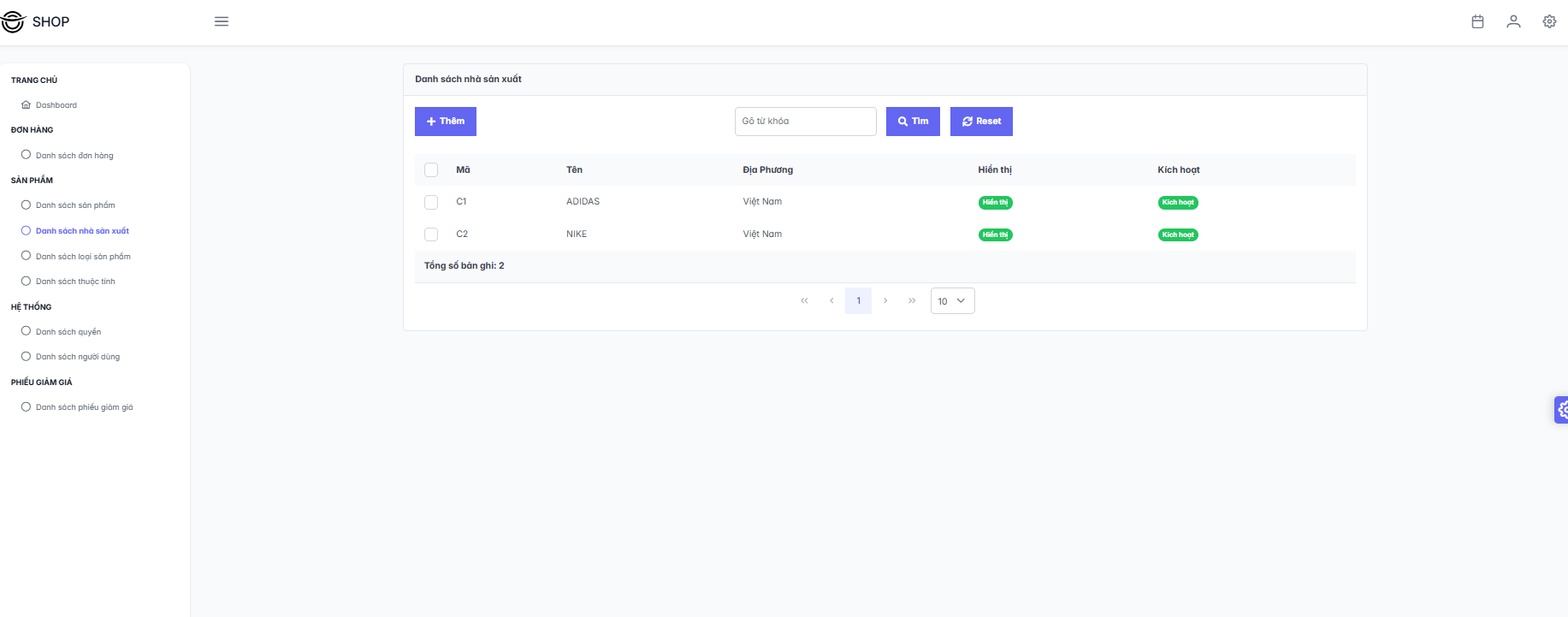
*-* Quản trị viên có thể thực hiện các thao tác: thêm, sửa, xóa, các sản phẩm. Có đầy đủ thông tin chi tiết của từng sản phẩm. Có thể chọn kích hoạt sản phẩm để sản phẩm hiện thị bên trang bán hàng của cửa hàng hoặc ẩn sản phẩm đó không cho nó xuất hiện trên trang bán hàng.

**

*Hình 4.13. Giao diện quản lý sản phẩm*

\*Giao diện quản lý nhà sản xuất

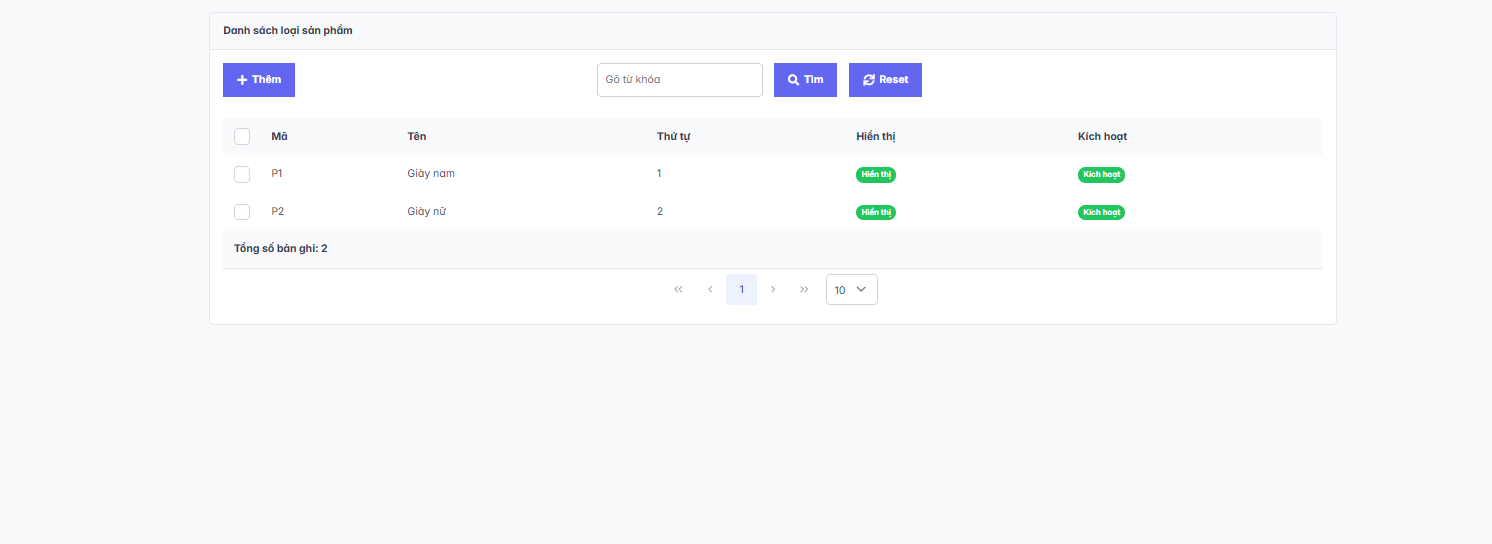
- Quản trị viên có thể thao tác: thêm, sửa, xóa, các nhà sản xuất.

**

*Hình 4.14. Giao diện quản lý nhà sản xuất*

\*Giao diện quản lý loại sản phẩm

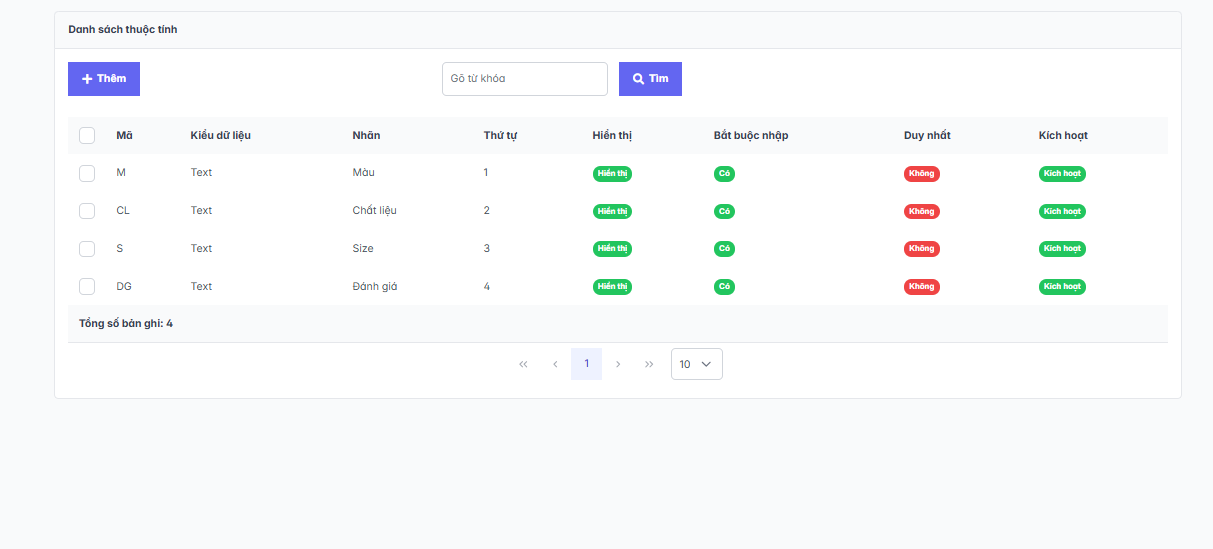
- Ở trang quản lý loại sản phẩm, quản trị viên có thể thao tác: thêm, sửa, xóa, các loại sản phẩm để thêm các sản phẩm tương ứng với mỗi loại sản phẩm.



*Hình 4.15. Giao diện quản lý loại sản phẩm*

\*Giao diện quản lý thuộc tính sản phẩm

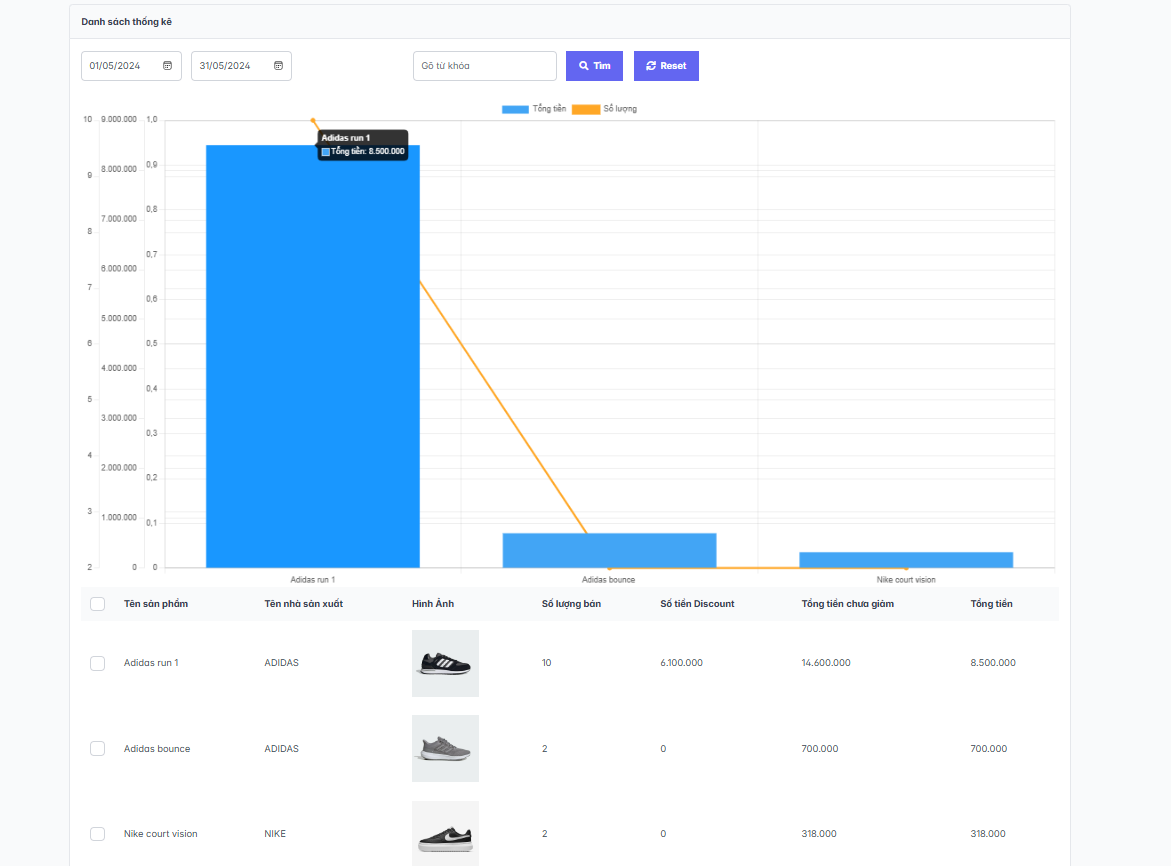
- Khi quản trị viên muốn thêm, sửa, xóa các thuộc tính của sản phẩm như: màu sắc, chất liệu, size, đánh giá... Quản trị viên có thể thao tác ở trang này



*Hình 4.16. Giao diện quản lý thuộc tính sản phẩm*

\*Giao diện thống kê

- Quản trị viên có thể chọn ngày để thống kê đơn hàng và tính doanh thu ở giao diện này.



*Hình 4.17. Giao diện thống kê*

# KẾT LUẬN

Đề tài **“*Xây dựng Website bán giày cho cửa hàng i-Sport bằng Asp.net Core*”** xuất phát từ những nhu cầu thực tế mà ngày nay các doanh nghiệp bán giày cần có. Những phương pháp buôn bán trực tiếp không phù hợp với một số người và thay vào đó là phương pháp mua bán trực tuyến giúp tiết kiệm thời gian và có nhiều thuận tiện.

Qua quá trình thực hiện đề tài, em đã tìm hiểu, tích lũy và học hỏi thêm được nhiều kinh nghiệm cũng như kiến thức công nghệ mới.

Đối với cá nhân:

- Củng cố được kiến thức nền vững chắc về CSDL, phân tích thiết kế hệ thống, các ngôn ngữ lập trình web.

- Tiếp thu được nhiều công nghệ mới, tối ưu hóa thời gian thực hiện dự án

- Được tham gia trực tiếp vào dự án thực tiễn, nâng cao kinh nghiệm làm việc trước khi ra trường.

- Củng cố thêm kiến thức về phân tích và thiết kế hệ thống.

- Nắm rõ quy trình phát triển một phần mềm.

Đối với sản phẩm:

- Cho phép người dùng có thể tìm và đặt hàng một cách nhanh chóng. Người quản lý cũng dễ dàng hơn trong việc quản lý cửa hàng.

- Xây dựng một hệ thống mua bán, quản lý cho cửa hàng bán giày, người quản trị có thể quản lý thông tin hồ sơ khách hàng, biết được doanh thu, quản lý đơn hàng hiệu quả nhanh chóng.

Yêu cầu chưa đạt được:

- Chưa có được chức năng như bình luận, đánh giá sản phẩm để khảo sát được mức độ hài lòng của khách hàng.

- Chưa tích hợp được thanh toán online để quá trình đặt hàng của khách hàng có thêm nhiều lựa chọn hơn.

- Chưa tích hợp chức năng đăng nhập bằng các tài khoản facebook, gmail.. Hướng nghiên cứu trong tương lai

- Tiếp tục cải thiện chương trình thêm các tính năng như bảo mật 2 lớp, thanh toán online, bình luận...

- Tối ưu cách viết code và cơ sở dữ liệu. Do còn thiếu kinh nghiệm thực tế cũng như thời gian hạn chế, ứng dụng của em vẫn còn nhiều hạn chế. Vì vậy, em mong rằng các quý thầy cô có thể đóng góp ý kiến để giúp em hoàn thiện ứng dụng tốt hơn trong tương lai.

# TÀI LIỆU THAM KHẢO

[1] Hoàng Quang Huy, Phùng Đức Hòa, Trịnh Bá Quý, Nhập môn công nghệ phần mềm, NXB Đại học Công nghiệp Hà Nội.

[2] Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Thị Kim Phượng, Giáo trình phân tích thiết kế hệ thống, NXB Giáo dục VN.

[3] Giáo trình thiết kế web, Trường đại học Công Nghiệp Hà Nội.

[4] Trang tài liệu UML <https://www.tutorialspoint.com/uml/index.htm>

[5] Trang tài liệu ASP.Net Core https://www.w3schools.com/asp/default.asp

[6] Trang tin tức về lập trình <https://viblo.asia>