

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC ĐẠI NAM



ĐỒ ÁN TỐT NGHIỆP

SỬ DỤNG CÔNG NGHỆ VUEJS XÂY DỰNG WEBSITE KINH DOANH CHO HỆ THỐNG CỬA HÀNG LAPTOP

SINH VIÊN THỰC HIỆN : ĐẬU TRUNG ĐỨC

MÃ SINH VIÊN : 1451020061

KHOA : CÔNG NGHỆ THÔNG TIN

HÀ NỘI - 2024

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



ĐẠI TRUNG ĐỨC

**SỬ DỤNG CÔNG NGHỆ VUEJS XÂY
DỰNG WEBSITE KINH DOANH CHO HỆ
THỐNG CỬA HÀNG LAPTOP**

**CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN
MÃ SỐ : 74.80.201**

NGƯỜI HƯỚNG DẪN: ThS. NGUYỄN HÀ THANH

LỜI CAM ĐOAN

Em tên là Đậu Trung Đức cam đoan rằng báo cáo thực tập về đề tài "**sử dụng công nghệ Vuejs xây dựng website kinh doanh cho hệ thống cửa hàng laptop**" là sản phẩm của bản thân đã hoàn thành.

Em cam đoan rằng trong quá trình thực hiện báo cáo này, em đã tuân thủ các quy định về nghiên cứu khoa học và trích dẫn tài liệu. Mọi thông tin, số liệu và kết quả được trình bày trong báo cáo đều được thu thập và xử lý một cách trung thực và đáng tin cậy.

Em xin chịu trách nhiệm về tính chính xác và độ tin cậy của báo cáo này. Báo cáo được trình bày dưới dạng một tài liệu tham khảo cho mục đích học tập và nghiên cứu.

Tp.Hà Nội, ngày 24 tháng 05 năm 2024

Người thực hiện

LỜI CẢM ƠN

Trước hết, em xin bày tỏ lòng biết ơn đối với thầy **Nguyễn Hà Thanh** giảng viên hướng dẫn. Thầy đã hướng dẫn và động viên em trong suốt quá trình làm đề bài này.

Em xin chân thành cảm ơn tất cả các thầy, cô giáo trong Khoa Công nghệ thông tin Trường Đại học Đại Nam, chân thành cảm ơn các thầy, cô tham gia giảng dạy và truyền đạt những kiến thức quý báu trong suốt thời gian nhóm em học tại trường.

Do thời gian thực hiện có hạn, kiến thức còn nhiều hạn chế nên bài tập này chắc chắn không tránh khỏi những thiếu sót nhất định. Em rất mong nhận được ý kiến đóng góp của thầy để em có thêm kinh nghiệm và tiếp tục hoàn thiện bài của mình hơn.

Em xin chân thành cảm ơn!

[illegible]

MỤC LỤC

CHƯƠNG 1 : KHÁI QUÁT ĐỀ TÀI.....	1
1.1. Lý do chọn đề tài	1
1.2. Phạm vi đề tài	2
1.3. Phương pháp nghiên cứu	2
CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT.....	4
2.1 Mô hình tổng quan.....	4
2.1.1 Lý do nên sử dụng mô hình MVC (Model-View-Controller)	4
2.2 Ngôn ngữ lập trình C# và môi trường phát triển frontend	6
2.2.1 Ngôn ngữ lập trình C# là gì	6
2.2.2 Đặc điểm của C#	7
2.2.3 Môi trường phát triển Visual studio	8
2.2.4 Framework ASP.NET Core	10
2.3 Giới thiệu về Microsoft SQL Server Management Studio.....	12
2.3.1 Khái niệm.....	12
2.3.2 Các chức năng chính	13
2.4 Ngôn ngữ lập trình VueJS và môi trường phát triển Frontend	15
2.4.1 Ngôn ngữ lập trình VueJS là gì	15
2.4.2 Môi trường phát triển Visual studio code	16
2.4.3 Framework Vuejs	17
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	19
3.1 Phân tích yêu cầu hệ thống.....	19

3.1.1 Mô tả bài toán:	19
3.1.2 Yêu cầu hệ thống:	19
3.1.3 Các chức năng chính:	20
3.2 <i>Thiết kế hệ thống</i>	21
3.2.1 Xác định Usecase và actor	21
3.2.2 Biểu đồ ca tổng quát	23
3.3. <i>Đặc tả usecase</i>	23
3.3.1. Use case đăng nhập.....	23
3.3.2. Use case đăng ký	24
3.3.3 Use case đăng xuất	25
3.3.4. Use case xem sản phẩm	25
3.3.5 Use case mua sản phẩm.....	26
3.3.6 Use case tìm kiếm sản phẩm	27
3.3.7 Use case Thêm sản phẩm	27
3.3.8 Use case Sửa sản phẩm.....	28
3.3.9 Use case Xóa sản phẩm.....	28
3.3.10 Use case Thêm danh mục	29
3.3.11 Use case sửa danh mục	30
3.3.12 Use case Xóa danh mục	30
3.3.13 Use case Xóa đơn hàng	31
3.3.14 Use case Thêm tài khoản khách hàng	31
3.3.15 Use case sửa tài khoản khách hàng.....	32
3.3.16 Use case Xóa tài khoản khách hàng	33
3.4. <i>Biểu đồ hoạt động</i>	33

3.5. Biểu đồ tuần tự.....	35
CHƯƠNG 4. THIẾT KẾ CƠ SỞ DỮ LIỆU VÀ KẾT QUẢ ĐẠT ĐƯỢC.....	48
4.1 Thiết kế cơ sở dữ liệu.....	48
4.1.1 Danh sách các bảng	48
4.1.2 Mô hình quan hệ thực thể ERD.....	52
4.2 Kết quả đạt được	<i>Error! Bookmark not defined.</i>
4.3 Giao diện trang web	54
4.3.1 Trang chủ.....	54
4.3.2 Footer	54
4.3.3 Trang đăng nhập.....	55
4.3.4 Trang đăng ký	55
4.3.5 Trang thanh toán	56
4.3.6 Trang đơn hàng của tôi	57
4.3.7 Trang chi tiết sản phẩm	57
4.3.8 Giao diện trang chủ của admin	58
4.3.9 Trang loại sản phẩm.....	58
4.3.10 Trang sản phẩm	59
4.3.11 Trang đơn hàng.....	59
KẾT LUẬN.....	60
TÀI LIỆU THAM KHẢO.....	62

DANH MỤC HÌNH ẢNH

Hình 2.1: Mô hình tổng quan	4
Hình 2.2: Ngôn ngữ lập trình C#	6
Hình 2.3: Visual Studio Code	8
Hình 2.4: ASP.NET Core.....	10
Hình 2.5: Microsoft SQL Server Management Studio	12
Hình 2.6: Ngôn ngữ lập trình VueJS	15
Hình 3.1: Sơ đồ usecase tổng quát	23
Hình 3.2: Biểu đồ hoạt động đăng nhập.....	34
Hình 3.3: Biểu đồ hoạt động đăng xuất.....	35
Hình 3.4: Biểu đồ tuần tự Đăng nhập	36
Hình 3.5: Biểu đồ tuần tự Đăng kí.....	37
Hình 3.6: Biểu đồ tuần tự Đăng xuất.....	38
Hình 3.7: Biểu đồ tuần tự Xem chi tiết sản phẩm.....	39
Hình 3.8: Biểu đồ tuần tự Đánh giá sản phẩm	40
Hình 3.9: Biểu đồ tuần tự Mua sản phẩm.....	41
Hình 3.10: Biểu đồ tuần tự Thanh toán	42
Hình 3.11: Biểu đồ tuần tự Tìm kiếm sản phẩm	43
Hình 3.12: Biểu đồ tuần tự Thêm sản phẩm.....	44
Hình 3.13: Biểu đồ tuần tự Xóa sản phẩm.....	45
Hình 3.14: Biểu đồ tuần tự sửa sản phẩm.....	46
Hình 4.1: Mô hình quan hệ thực thể ERD	52
Hình 4.2: Các thành phần mã nguồn backend	53
Hình 4.3: Các phương thức trong API (Swagger)	54

Hình 4.4: Giao diện trang chủ.....	54
Hình 4.5: Giao diện footer	54
Hình 4.6: Giao diện đăng nhập	55
Hình 4.7: Giao diện đăng ký.....	56
Hình 4.8: Giao diện thanh toán.....	56
Hình 4.9: Giao diện đơn hàng của tôi.....	57
Hình 4.10: Giao diện chi tiết sản phẩm.....	57
Hình 4.11: Giao diện trang chủ của admin.....	58
Hình 4.12: Giao diện trang loại sản phẩm	58
Hình 4.13: Giao diện trang sản phẩm.....	59
Hình 4.14: Giao diện trang đơn hàng	59

DANH MỤC BẢNG BIỂU

Bảng 3.1: Bảng miêu tả chức năng admin	20
Bảng 3.2: Bảng miêu tả chức năng user	21
Bảng 3.3: Xác định Usecase và actor	23
Bảng 3.4: Use case đăng nhập	24
Bảng 3.5: Use case đăng ký	25
Bảng 3.6: Use case đăng xuất	25
Bảng 3.7: Use case xem sản phẩm.....	26
Bảng 3.8: Use case mua sản phẩm	27
Bảng 3.9: Use case tìm kiếm sản phẩm	27
Bảng 3.10: Use case Thêm sản phẩm.....	28
Bảng 3.11: Use case Sửa sản phẩm	28
Bảng 3.13: Use case Thêm danh mục	30
Bảng 3.15: Use case Xóa danh mục	31
Bảng 3.16: Use case Xóa đơn hàng	31
Bảng 3.17: Use case Thêm tài khoản khách hàng	32
Bảng 3.18: Use case sửa tài khoản khách hàng	33
Bảng 3.19: Use case Xóa tài khoản khách hàng	33
Bảng 4.1: Dữ Liệu Users	48
Bảng 4.2: Dữ liệu Sản phẩm.....	49
Bảng 4.3: Dữ liệu OrderProduct.....	50
Bảng 4.4: Dữ liệu OrderDetail	51
Bảng 4.5: Dữ liệu Category	52

CHƯƠNG 1 : KHÁI QUÁT ĐỀ TÀI

1.1. Lý do chọn đề tài

Việc chọn đề tài xây dựng website cho cửa hàng Laptop là một quyết định mang lại nhiều lợi ích và tiềm năng phát triển rất lớn. Đầu tiên và quan trọng nhất, ngành công nghiệp bán lẻ laptop đang ngày càng phát triển mạnh mẽ, với nhu cầu sử dụng laptop tăng cao từ cả cá nhân, doanh nghiệp và tổ chức. Mỗi ngày, có hàng ngàn người dùng tìm kiếm và mua sắm các sản phẩm công nghệ, trong đó laptop chiếm một phần quan trọng. Việc có một website chuyên nghiệp và tiện lợi sẽ giúp cửa hàng Laptop thu hút được nhiều khách hàng hơn, từ đó tăng doanh số bán hàng và lợi nhuận.

Thứ hai, môi trường mua sắm trực tuyến đang trở thành xu hướng phổ biến, đặc biệt là trong bối cảnh dịch bệnh và sự tiện lợi của việc mua sắm từ xa. Một website cửa hàng Laptop chất lượng không chỉ là một phần của chiến lược kinh doanh hiện đại mà còn là cơ hội để cửa hàng tiếp cận được một lượng lớn khách hàng tiềm năng, bất kể họ đến từ đâu.

Thứ ba, một website cửa hàng Laptop mang lại cho khách hàng trải nghiệm mua sắm trực tuyến dễ dàng và thoải mái hơn. Họ có thể duyệt qua các sản phẩm, so sánh giá cả, đặt hàng và thanh toán mà không cần phải rời khỏi nhà. Điều này giúp tiết kiệm thời gian và công sức cho khách hàng, từ đó tạo ra sự hài lòng và lòng trung thành với cửa hàng.

Cuối cùng, một website cửa hàng Laptop cũng mở ra cơ hội cho việc tương tác với khách hàng thông qua các kênh truyền thông xã hội, email marketing và các chương trình khuyến mãi. Điều này giúp cửa hàng tạo dựng một mối quan hệ chặt chẽ và nắm bắt được nhu cầu của khách hàng, từ đó điều chỉnh chiến lược kinh doanh một cách linh hoạt và hiệu quả.

Tóm lại, việc xây dựng một website cho cửa hàng Laptop không chỉ là một cách để mở rộng thị trường và tăng doanh số bán hàng mà còn là một bước tiến quan trọng trong việc thích ứng với xu hướng mua sắm hiện đại và tạo ra một trải nghiệm mua sắm tốt nhất cho khách hàng.

1.2. Phạm vi đề tài

Phạm vi của dự án này được xác định để tập trung vào việc phát triển một ứng dụng web bán Laptop một cách đáng tin cậy và hiệu quả.

- Phát triển một giao diện người dùng thân thiện và dễ sử dụng, bao gồm các tính năng nhập liệu, xem/xử lý tồn kho, tạo và quản lý đơn hàng.
- Xây dựng cơ sở dữ liệu để lưu trữ thông tin về vật liệu, tồn kho và các đơn hàng, đảm bảo tính nhất quán và bảo mật dữ liệu.
- Tối ưu hóa quá trình quản lý thông qua việc cung cấp các chức năng thống kê và phân tích dữ liệu, giúp người dùng đưa ra các quyết định thông minh và dự đoán nhu cầu vật liệu
- Triển khai và kiểm thử ứng dụng trên một số ứng dụng một số trường thực tế để đảm bảo tính ổn định và hiệu suất của hệ thống trước khi đưa vào sử dụng thực tế.

Với phạm vi nghiên cứu được xác định rõ ràng, dự án này hy vọng sẽ đạt được những kết quả tích cực và mang lại giá trị thực tế cho người sử dụng.

1.3. Phương pháp nghiên cứu

Dựa trên những kết quả tìm hiểu được để xây dựng một website có áp dụng tất cả những nội dung đã tìm hiểu.

Các công cụ hỗ trợ:

- Microsoft Word: dùng để viết tài liệu báo cáo.
- Visual Studio Code: dùng để code front-end.
- Visual Studio: dùng để code back-end.
- SQL Server Management Studio: dùng để lưu trữ cơ sở dữ liệu.

Trong phần này, sẽ mô tả các phương pháp nghiên cứu được sử dụng để thực hiện đề tài “

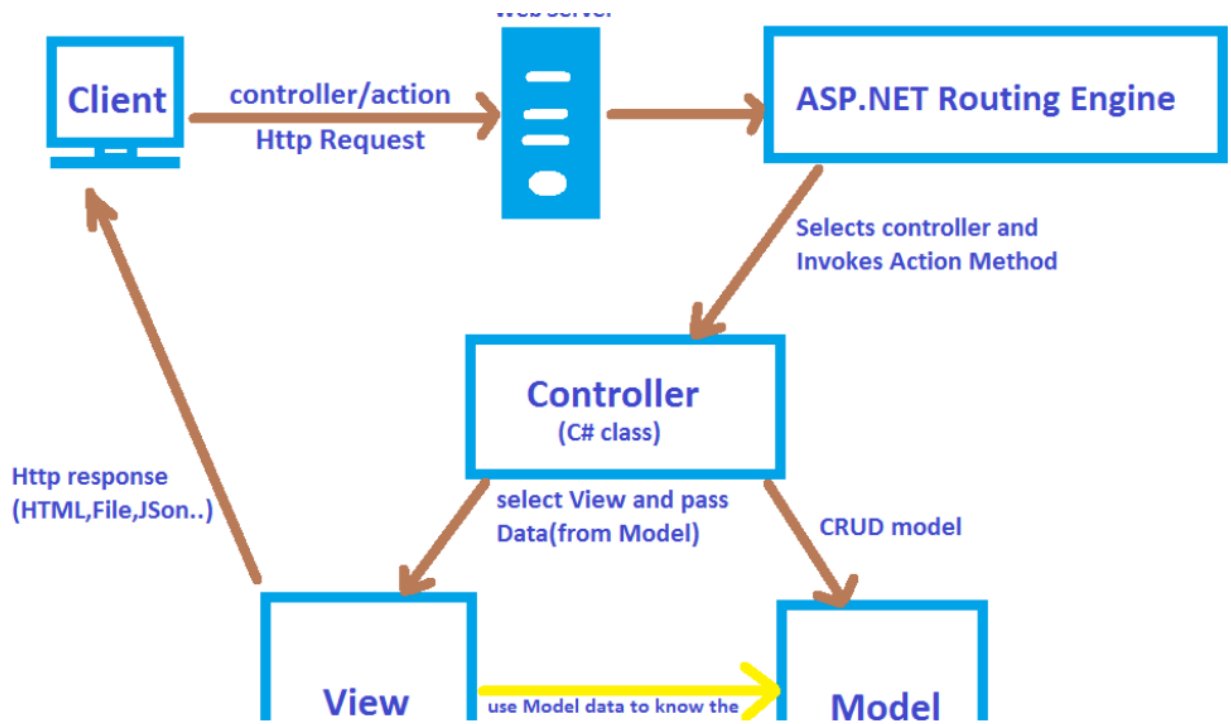
Sử dụng công nghệ Vuejs xây dựng website kinh doanh cho hệ thống cửa hàng Laptop”.

Phương pháp nghiên cứu này sẽ bao gồm các bước sau:

- Phân tích tài liệu: Tiến hành phân tích các tài liệu liên quan như sách, báo cáo, và các tài liệu nghiên cứu trước đây để hiểu rõ hơn về tình hình hiện tại của ngành buôn bán Laptop và các phương pháp quản lý đã được áp dụng. Điều này sẽ giúp xây dựng nền tảng cho việc tiếp cận vấn đề.
- Thu thập dữ liệu thực tế: Tiến hành thu thập dữ liệu thực tế từ các doanh nghiệp mua bán Laptop thông qua cuộc phỏng vấn, khảo sát và quan sát trực tiếp. Dữ liệu này sẽ giúp hiểu rõ hơn về thực tế hoạt động của các cửa hàng và các vấn đề mà họ đang đối mặt.
- Phương pháp nghiên cứu định lượng và định tính: Sử dụng phương pháp định lượng sẽ giúp đo lường và số hóa các yếu tố như hiệu suất kinh doanh và mức độ hài lòng của khách hàng, trong khi phương pháp định tính sẽ giúp hiểu sâu hơn về các quan điểm, ý kiến và trải nghiệm cá nhân.
- Phân tích dữ liệu: Dữ liệu thu thập được sẽ được phân tích một cách kỹ lưỡng để trả lời câu hỏi nghiên cứu và đạt được mục tiêu đề ra. Sử dụng các phương pháp phân tích thống kê và các công cụ phân tích dữ liệu để phát hiện ra mối liên hệ và xu hướng quan trọng trong dữ liệu.
- Đánh giá kết quả và rút ra kết luận: Cuối cùng, đánh giá kết quả của nghiên cứu và rút ra những kết luận và nhận định quan trọng. Xem xét mức độ đạt được của mục tiêu nghiên cứu, nhận dạng các hạn chế và yếu điểm, và đề xuất các hướng phát triển và ứng dụng tiếp theo.

CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

2.1 Mô hình tổng quan



Hình 2.1: Mô hình tổng quan

Mô hình MVC, viết tắt của Model-View-Controller, là một kiến trúc phần mềm phổ biến được sử dụng rộng rãi trong phát triển ứng dụng. Được thiết kế để tách biệt logic xử lý dữ liệu (Model), hiển thị giao diện người dùng (View), và điều phối cả hai (Controller), mô hình này mang lại nhiều lợi ích trong việc quản lý mã nguồn và cải thiện khả năng bảo trì của hệ thống.

Khi sử dụng mô hình MVC, những người phát triển cần phải sở hữu một số kỹ năng quan trọng để tận dụng hết tiềm năng của kiến trúc này. Không chỉ là việc hiểu rõ về cách các thành phần Model, View, và Controller tương tác, mà còn là khả năng áp dụng chúng linh hoạt để tạo ra các ứng dụng mạnh mẽ và linh hoạt.

2.1.1 Lý do nên sử dụng mô hình MVC (Model-View-Controller)

Mô hình MVC, hay Model-View-Controller, là một kiến trúc phần mềm được sử dụng để tổ chức và quản lý mã nguồn trong quá trình phát triển ứng dụng. Kiến trúc này tách biệt ba thành phần chính: Model, View, và Controller, giúp giảm độ phức tạp của hệ thống và làm cho mã nguồn dễ quản lý hơn.

Model (M - Model): Đây là thành phần chịu trách nhiệm cho xử lý dữ liệu và logic kinh doanh của ứng dụng. Model là nơi lưu trữ thông tin, thực hiện các thao tác cập nhật và truy vấn dữ liệu, mà không quan tâm đến cách dữ liệu được hiển thị hoặc tương tác với người dùng.

View (V - View): View là thành phần hiển thị giao diện người dùng và đảm nhận trách nhiệm hiển thị thông tin từ Model. View không có logic kinh doanh và chỉ chịu trách nhiệm về việc hiển thị dữ liệu một cách đẹp mắt và dễ hiểu cho người dùng.

Controller (C - Controller): Controller là thành phần điều phối và xử lý sự kiện từ người dùng. Nó nhận lệnh từ người dùng thông qua View, sau đó cập nhật Model dựa trên những thay đổi này và điều hướng hiển thị trở lại View. Controller giữ vai trò quan trọng trong việc duy trì sự đồng bộ giữa Model và View.

Mô hình MVC giúp tách biệt logic ứng dụng thành các phần nhỏ, dễ quản lý và bảo trì. Nó cũng thúc đẩy tái sử dụng mã nguồn, khiến cho quá trình phát triển và bảo trì ứng dụng trở nên hiệu quả hơn.

2.2 Ngôn ngữ lập trình C# và môi trường phát triển frontend

2.2.1 Ngôn ngữ lập trình C# là gì



Hình 2.2: Ngôn ngữ lập trình C#

C# (C Sharp, đọc là "xi-sáp") là một ngôn ngữ lập trình hướng đối tượng đa năng, mạnh mẽ được phát triển bởi Microsoft, C# là phần khởi đầu cho kế hoạch .NET của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là C#, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên C++ và Java. C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java.

C# được thiết kế chủ yếu bởi Anders Hejlsberg kiến trúc sư phần mềm nổi tiếng với các sản phẩm Turbo Pascal, Delphi, J++, WFC. Phiên bản gần đây nhất là 9.0, được phát hành vào năm 2020 cùng với Visual Studio 2019 phiên bản 16.8.

Ngôn ngữ C# là một ngôn ngữ lập trình đơn giản, hiện đại, hướng đến nhiều mục đích sử dụng, và là một ngôn ngữ lập trình hướng đối tượng.

Ngôn ngữ C# đáp ứng các nguyên tắc của ngành kỹ thuật phần mềm như kiểm tra chặt chẽ kiểu dữ liệu, kiểm tra giới hạn mảng, phát hiện các trường hợp sử dụng các

biến chưa có dữ liệu, và tự động thu gom rác. Tính mạnh mẽ, sự bền bỉ, và năng suất của việc lập trình là rất quan trọng đối với ngôn ngữ này.

Ngôn ngữ C# được thiết kế để phù hợp với việc viết các ứng dụng cho cả hai hệ thống: hosted và nhúng, từ các phần mềm quy mô lớn, đến các phần mềm chỉ có các chức năng đơn giản.

2.2.2 Đặc điểm của C#

a) C# là ngôn ngữ đơn giản

C# loại bỏ một vài sự phức tạp và rối rắm của những ngôn ngữ như Java và c++, bao gồm việc loại bỏ những macro, những template, đa kế thừa, và lớp cơ sở ảo (virtual base class).

Ngôn ngữ C# đơn giản vì nó dựa trên nền tảng C và C++. Nếu chúng ta thân thiện với C và C++ hoặc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn.

b) C# là ngôn ngữ hiện đại

Điều gì làm cho một ngôn ngữ hiện đại? Những đặc tính như là xử lý ngoại lệ, thu gom bộ nhớ tự động, những kiểu dữ liệu mở rộng, và bảo mật mã nguồn là những đặc tính được mong đợi trong một ngôn ngữ hiện đại. C# chứa tất cả những đặc tính trên. Nếu là người mới học lập trình có thể chúng ta sẽ cảm thấy những đặc tính trên phức tạp và khó hiểu. Tuy nhiên, cũng đừng lo lắng chúng ta sẽ dần dần được tìm hiểu những đặc tính qua các nội dung khoá học này.

c) C# là một ngôn ngữ lập trình thuần hướng đối tượng

Lập trình hướng đối tượng (OOP: Object-oriented programming) là một phương pháp lập trình có 4 tính chất. Đó là tính trừu tượng (abstraction), tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính kế thừa (inheritance). C# hỗ trợ cho chúng ta tất cả những đặc tính trên.

d) C# là một ngôn ngữ ít từ khóa

C# là ngôn ngữ sử dụng giới hạn những từ khóa. Phần lớn các từ khóa được sử dụng để mô tả thông tin. Chúng ta có thể nghĩ rằng một ngôn ngữ có nhiều từ khóa thì sẽ mạnh hơn. Điều này không phải sự thật, ít nhất là trong trường hợp ngôn ngữ C#, chúng ta có thể tìm thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

2.2.3 Môi trường phát triển Visual studio



Hình 2.3: Visual Studio Code

Visual Studio được hiểu là một hệ thống bao gồm tất cả những gì có liên quan đến phát triển ứng dụng như trình chỉnh sửa mã, thiết kế, gỡ lỗi, viết code hay chỉnh sửa thiết kế, ứng dụng một cách dễ dàng và nhanh chóng. Có thể nói đây là một phần mềm hỗ trợ đắc lực đối với dân lập trình trong việc lập trình website. Visual Studio cho phép người dùng có khả năng thiết kế và trải nghiệm giao diện như khi phát triển ứng dụng.

Visual Studio được thiết lập vào năm 1997 và thuộc quyền sở hữu bởi Microsoft, ông lớn này đã cho ra mắt hai phiên bản của Visual Studio là Professional và Enterprise.

Ban đầu phần mềm này có tên là Project Boston, nhưng sau đó được đổi tên lại vì được tích hợp thêm các công cụ phát triển và đóng gói sản phẩm duy nhất.

Trải qua quãng thời gian dài với nhiều phiên bản khác nhau, Visual Studio được chính thức phát hành vào tháng 7 năm 2015.

Visual Studio có một số tính năng rất độc đáo như:

- Hỗ trợ nhiều ngôn ngữ lập trình: Tính năng này giúp phát hiện bất kỳ lỗi hoặc tham chiếu ngôn ngữ chéo (cross-language reference) nào một cách dễ dàng.
- Intelli-Sense: Là một tính năng giúp phát hiện có bất kỳ đoạn code nào bị bỏ sót hay không, tự động thực thi cú pháp biến (variable syntaxes) và khai báo biến (variable declarations). Ví dụ: Nếu một biến nào đó đang được sử dụng trong chương trình và người dùng quên khai báo, intellisense sẽ khai báo biến đó cho người dùng.
- Hỗ trợ đa nền tảng: Visual Studio hoạt động trên cả 3 nền tảng Windows, Linux, Mac.
- Repository: Visual Studio được kết nối với Git hoặc có thể được kết nối với bất kỳ Repository nào khác.
- Code editor: Visual Studio có class Code editor tốt nhất hỗ trợ nhiều chức năng. Nó cho phép bookmark trong code để kết hợp Quick Navigation. Visual Studio cũng có chức năng Incremental Search, Regex Search, Multi-item Clipboard và Task-list.
- Web-Support: Các ứng dụng web có thể được xây dựng và hỗ trợ trong Visual Studio.
- Hỗ trợ Terminal: Visual Studio hỗ trợ Terminal hoặc Console tích hợp giúp người dùng không cần chuyển đổi giữa hai màn hình.
- Thiết kế đa dạng: Visual Studio cung cấp một số visual designer để trợ giúp trong việc phát triển các ứng dụng:

- WPF Designer: tạo giao diện người dùng cho Windows Presentation Foundation.
- Windows Forms Designer: tạo các ứng dụng GUI bằng Windows Forms.
- Class designer: Class designer cho phép chỉnh sửa các class bao gồm các thành viên và quyền truy cập của chúng bằng cách sử dụng mô hình UML.
- Web designer: Visual Studio cũng hỗ trợ một trình soạn thảo và thiết kế trang web, cho phép tạo các trang web bằng cách kéo và thả các widget.
- Mapping Designer: Mapping Designer được LINQ to SQL sử dụng để thiết kế mapping giữa các lược đồ thông tin và từ đó các class sẽ đóng gói dữ liệu.
- Data Designer: Data Designer được sử dụng để chỉnh sửa các lược đồ thông tin, cũng như các bảng được viết, khóa chính và khóa ngoại và các ràng buộc (constraint).

2.2.4 Framework ASP.NET Core



Hình 2.4: ASP.NET Core

ASP.NET Core 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 cụ mà được triển khai trên đám mây (cloud) hoặc chạy on-premise.

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

Một số lợi ích khi sử dụng ASP.NET Core cho việc xây dựng backend:

- Mã nguồn mở: ASP.NET Core là mã nguồn mở, cho phép bạn tự do sử dụng và sửa đổi.
- Đa nền tảng: ASP.NET Core có thể chạy trên Windows, macOS và Linux.
- Hiệu suất cao: ASP.NET Core được thiết kế để có hiệu suất cao, giúp bạn xây dựng các ứng dụng web có thể xử lý lượng truy cập lớn.
- Có thể mở rộng: ASP.NET Core được thiết kế để có thể mở rộng, giúp bạn dễ dàng phát triển các ứng dụng web lớn và phức tạp.
- Hỗ trợ nhiều ngôn ngữ lập trình: ASP.NET Core hỗ trợ nhiều ngôn ngữ lập trình khác nhau, bao gồm C#, F# và Visual Basic .NET.
- Cộng đồng lớn: ASP.NET Core có một cộng đồng lớn các nhà phát triển có thể hỗ trợ bạn trong quá trình phát triển.

ASP.NET Core có thể được sử dụng để xây dựng backend cho nhiều loại ứng dụng web khác nhau, bao gồm:

- Trang web: ASP.NET Core có thể được sử dụng để xây dựng backend cho các trang web tĩnh và động.
- API: ASP.NET Core có thể được sử dụng để xây dựng backend cho các API RESTful.

- Ứng dụng web một trang: ASP.NET Core có thể được sử dụng để xây dựng backend cho các ứng dụng web một trang (SPA) sử dụng JavaScript frameworks như AngularJS, ReactJS hoặc VueJS.
- Ứng dụng web di động: ASP.NET Core có thể được sử dụng để xây dựng backend cho các ứng dụng web di động bằng cách sử dụng Xamarin.

2.3 Giới thiệu về Microsoft SQL Server Management Studio



Hình 2.5: Microsoft SQL Server Management Studio

2.3.1 Khái niệm

Microsoft SQL Server Management Studio là phần mềm cho hệ điều hành Windows, thuộc nhóm phần mềm Software được phát triển bởi Microsoft. Phiên bản mới nhất của Microsoft SQL Server Management Studio là Version 18.11.1 (cập nhật 14/11/2023).

Microsoft SQL Server Management Studio là phần mềm cho phép người sử dụng, thường là nhà quản trị mạng, quản lý tất cả các khía cạnh của máy chủ SQL. Phiên bản ban đầu của phần mềm là với phiên bản năm 2005 của SQL Server. Người dùng có thể cấu hình tất cả các thành phần của máy chủ thông qua kịch bản và các công cụ đồ họa

được cung cấp với phần mềm. Microsoft SQL Server Management Studio cung cấp một môi trường tích hợp để thực hiện tất cả các cấu hình cần thiết và nhiệm vụ quản lý để duy trì cơ sở dữ liệu. Cách bố trí của môi trường có thể được tùy chỉnh, nơi người dùng có thể ẩn các thành phần của như Object Explorer thông qua Ẩn tùy chọn Auto. Thông thường, các thành phần ẩn được giảm thiểu, và lơ lửng chuột lên tên thành phần sẽ mở ra các thành phần ẩn một lần nữa. Thành phần tương tự như vậy có thể được undocked từ vị trí ban đầu của nó và chuyển đến một địa điểm khác trên bố trí. chương trình bàn phím tương tự như vậy có thể được cấu hình. Điều này có thể được thực hiện thông qua tab Options trong Tools tùy chọn. Trong trường hợp người dùng muốn quay trở lại các thiết lập ban đầu của cách bố trí môi trường, điều này có thể được thực hiện thông qua các tùy chọn Window Layout Reset, mà có thể được truy cập từ tab Window.

2.3.2 Các chức năng chính

SQL Server Management Studio là một công cụ miễn phí được cung cấp bởi Microsoft để quản lý và cấu hình các phiên bản SQL Server. Nó cung cấp một giao diện đồ họa trực quan giúp người dùng dễ dàng thực hiện các tác vụ quản trị mà không cần phải viết mã T-SQL.

SQL Server Management Studio có một số chức năng chính:

a) Quản lý cơ sở dữ liệu:

- Tạo, chỉnh sửa và xóa cơ sở dữ liệu: cho phép bạn tạo cơ sở dữ liệu mới cho trang webtoon của bạn, chỉnh sửa cấu trúc của nó và xóa nó khi cần thiết.

- Sao lưu và phục hồi cơ sở dữ liệu: cung cấp các công cụ để sao lưu cơ sở dữ liệu của bạn, đảm bảo dữ liệu được bảo vệ trong trường hợp xảy ra sự cố.

- Quản lý người dùng và quyền truy cập: cho phép bạn tạo người dùng mới và cấp quyền truy cập vào cơ sở dữ liệu cho họ, đảm bảo chỉ những người được ủy quyền mới có thể truy cập dữ liệu.

- Giám sát hiệu suất cơ sở dữ liệu: cung cấp các công cụ để theo dõi hiệu suất của cơ sở dữ liệu, giúp bạn xác định và giải quyết các vấn đề hiệu suất tiềm ẩn.

b) Phát triển ứng dụng:

- Viết và thực thi truy vấn T-SQL: cung cấp một môi trường để viết và thực thi các truy vấn T-SQL, cho phép bạn truy xuất và thao tác dữ liệu trong cơ sở dữ liệu.

- Cài đặt và quản lý stored procedures: cho phép bạn cài đặt và quản lý stored procedures, giúp bạn tự động hóa các tác vụ truy vấn và thao tác dữ liệu.

- Gỡ lỗi stored procedures: cung cấp các công cụ để gỡ lỗi stored procedures, giúp bạn xác định và sửa lỗi trong code.

- Phân tích dữ liệu: cung cấp các công cụ để phân tích dữ liệu trong cơ sở dữ liệu, giúp bạn thu thập thông tin chi tiết về hiệu suất trang webtoon của bạn.

- Báo cáo: cung cấp các công cụ để tạo báo cáo dựa trên dữ liệu trong cơ sở dữ liệu, giúp bạn dễ dàng chia sẻ thông tin chi tiết với các bên liên quan.

c) Tích hợp:

- Hỗ trợ nhiều phiên bản SQL Server: hỗ trợ nhiều phiên bản SQL Server, cho phép bạn sử dụng nó với bất kỳ phiên bản nào bạn đang sử dụng.

- Tích hợp với Visual Studio: tích hợp với Visual Studio, cho phép bạn dễ dàng phát triển các ứng dụng webtoon sử dụng SQL Server.

- Hỗ trợ PowerShell: hỗ trợ PowerShell, cho phép bạn tự động hóa các tác vụ quản trị cơ sở dữ liệu.

d) Dễ sử dụng:

- Giao diện đồ họa trực quan: cung cấp một giao diện đồ họa trực quan, giúp bạn dễ dàng sử dụng các chức năng của nó.

- Trợ giúp ngữ cảnh: cung cấp trợ giúp ngữ cảnh, giúp bạn dễ dàng tìm hiểu cách sử dụng các chức năng của nó.

- Nhiều tùy chọn tùy chỉnh: cung cấp nhiều tùy chọn tùy chỉnh, cho phép bạn điều chỉnh giao diện và chức năng của nó theo nhu cầu của bạn.

Ngoài ra, SQL Server Management Studio còn cung cấp một số tính năng nâng cao như:

- Quản lý các phiên bản SQL Server trên Azure

- Triển khai các gói Integration Services
- Quản lý các phiên bản SQL Server Analysis Services

SQL Server Management Studio là một công cụ mạnh mẽ và linh hoạt cho phép người dùng quản lý và cấu hình SQL Server một cách hiệu quả. Nó là một công cụ cần thiết cho bất kỳ ai làm việc với SQL Server. (Lưu ý: SQL Server Management Studio chỉ chạy trên hệ điều hành Windows.

2.4 Ngôn ngữ lập trình VueJS và môi trường phát triển Frontend



Hình 2.6: Ngôn ngữ lập trình VueJS

2.4.1 Ngôn ngữ lập trình VueJS là gì

Vue.js là một framework JavaScript được sử dụng chủ yếu để xây dựng các ứng dụng web đơn trang (Single Page Applications - SPAs). Nó giúp phát triển giao diện người dùng linh hoạt và dễ dàng quản lý các thành phần của ứng dụng.

Một số điểm chính về ngôn ngữ lập trình Vue.js:

- JavaScript-Based: Vue.js được xây dựng dựa trên ngôn ngữ JavaScript, nó sử dụng cú pháp gần giống với HTML để định nghĩa các thành phần và tương tác với DOM.

- Khả năng Tương tác: Vue.js cho phép bạn tương tác với DOM thông qua các biểu thức (expressions), directives (chỉ thị), và các sự kiện (events) để thay đổi trạng thái của ứng dụng.
- Cấu trúc Component: Vue.js thúc đẩy việc chia ứng dụng thành các thành phần riêng lẻ, giúp quản lý mã nguồn dễ dàng hơn. Mỗi thành phần có thể chứa template, logic, và CSS của nó.
- Reactivity (Phản ứng): Vue.js sử dụng hệ thống phản ứng để tự động cập nhật giao diện người dùng khi dữ liệu thay đổi. Điều này giúp giảm độ phức tạp của việc theo dõi và cập nhật DOM thủ công.
- Directive: Vue.js cung cấp các chỉ thị (directives) như v-bind, v-if, v-for, v-on, giúp bạn quản lý DOM một cách hiệu quả và linh hoạt.
- Vue CLI: Để tạo và quản lý dự án Vue.js, bạn có thể sử dụng Vue CLI, một công cụ dòng lệnh giúp tự động hóa quy trình phát triển.

2.4.2 Môi trường phát triển Visual studio code

Visual Studio Code (VSCode) là một môi trường phát triển tích hợp (Integrated Development Environment - IDE) nhỏ gọn, mạnh mẽ và miễn phí, được phát triển bởi Microsoft. Được thiết kế để hỗ trợ nhiều ngôn ngữ lập trình và framework khác nhau, VSCode đã trở thành một trong những trình biên soạn mã nguồn phổ biến nhất trong cộng đồng phát triển phần mềm.

VSCode nổi bật với giao diện người dùng đơn giản và trực quan, cung cấp một loạt các tính năng hữu ích như kiểm soát phiên bản tích hợp, gợi ý mã (code completion), và đánh giá mã (linting). Nó hỗ trợ nhiều ngôn ngữ lập trình như JavaScript, TypeScript, Python, Java, C#, và nhiều ngôn ngữ khác thông qua các extension mà cộng đồng phát triển đóng góp.

Một trong những điểm mạnh đặc biệt của VSCode là khả năng mở rộng linh hoạt thông qua hệ thống extension. Người dùng có thể tùy chỉnh và mở rộng chức năng của VSCode bằng cách cài đặt các extension từ Marketplace, giúp tối ưu hóa trải nghiệm lập trình theo nhu cầu cụ thể của họ.

Với khả năng tích hợp sâu sắc với nền tảng Git, khả năng gỡ lỗi tốt, và tương thích đa nền tảng (Windows, macOS, Linux), Visual Studio Code là một công cụ lý tưởng cho các nhà phát triển ứng dụng di động, web, và các dự án phần mềm đa dạng.

2.4.3 Framework Vuejs

Gọi tắt là Vue (phát âm là /vju:/, giống như view trong tiếng Anh), Vue.js là một framework linh động (nguyên bản tiếng Anh: progressive – tiệm tiến) dùng để xây dựng giao diện người dùng (user interfaces). Khác với các framework nguyên khối (monolithic), Vue được thiết kế từ đầu theo hướng cho phép và khuyến khích việc phát triển ứng dụng theo từng bước.

Khi phát triển lớp giao diện (view layer), người dùng chỉ cần dùng thư viện lõi (core library) của Vue, vốn rất dễ học và tích hợp với các thư viện hoặc dự án có sẵn. Cùng lúc đó, nếu kết hợp với những kỹ thuật hiện đại như SFC (single file components) và các thư viện hỗ trợ, Vue cũng đáp ứng được dễ dàng nhu cầu xây dựng những ứng dụng một trang (SPA - Single-Page Applications) với độ phức tạp cao hơn nhiều.

Điểm nổi bật của Vue.js:

- Dễ học: Vue.js có cú pháp trực quan, dễ hiểu, giúp bạn dễ dàng bắt đầu và tạo ra các ứng dụng web nhanh chóng.
- Linh hoạt: Vue.js có thể được sử dụng để xây dựng đa dạng các ứng dụng web, từ các trang web tĩnh đơn giản đến các ứng dụng web một trang (SPA) phức tạp.
- Hiệu suất cao: Vue.js được thiết kế để tối ưu hóa hiệu suất, giúp bạn tạo ra các ứng dụng web mượt mà và nhanh chóng.
- Cộng đồng lớn: Vue.js sở hữu cộng đồng phát triển rộng lớn với nhiều nguồn tài liệu, hướng dẫn và thư viện hỗ trợ phong phú.

Ứng dụng của Vue.js:

- Xây dựng trang web tĩnh: Vue.js giúp thêm các hiệu ứng tương tác cho trang web tĩnh, nâng cao trải nghiệm người dùng.

- Xây dựng ứng web một trang (SPA): Vue.js là lựa chọn lý tưởng cho SPA với khả năng chuyển tiếp mượt mà giữa các trang.
- Xây dựng ứng web thời gian thực: Vue.js có thể được sử dụng để xây dựng các ứng web thời gian thực như chatbots, bảng điều khiển,...

Lợi ích khi sử dụng Vue.js:

- Dễ dàng học và sử dụng: Vue.js phù hợp cho cả người mới bắt đầu và nhà phát triển JavaScript có kinh nghiệm.
- Linh hoạt và mở rộng: Vue.js dễ dàng tích hợp với các thư viện và framework khác.
- Hiệu suất cao: Vue.js giúp tối ưu hóa hiệu suất ứng dụng web.

Cộng đồng lớn và hỗ trợ: Cộng đồng Vue.js luôn sẵn sàng hỗ trợ giải đáp thắc mắc và chia sẻ kinh nghiệm.

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1 Phân tích yêu cầu hệ thống

3.1.1 Mô tả bài toán:

Hệ thống phần mềm cho trang web bán Laptop được thiết kế để cung cấp một trải nghiệm mua sắm trực tuyến tốt nhất cho người dùng. Với giao diện người dùng thân thiện và dễ sử dụng, người dùng có thể dễ dàng tìm kiếm, lựa chọn và mua các sản phẩm Laptop một cách thuận tiện. Quản lý sản phẩm linh hoạt cho phép quản trị viên quản lý các sản phẩm một cách hiệu quả, trong khi tính năng giỏ hàng và thanh toán an toàn đảm bảo việc mua sắm diễn ra một cách trơn tru và đáng tin cậy. Đồng thời, việc tích hợp các tính năng đăng ký thành viên, đăng nhập, tìm kiếm và lọc sản phẩm, cùng với khả năng đánh giá và phản hồi sản phẩm, tạo điều kiện thuận lợi cho người dùng tương tác và chia sẻ ý kiến. Tính năng quản trị tiện lợi và tối ưu hóa hiệu suất và bảo mật đảm bảo rằng cả người dùng và quản trị viên đều có trải nghiệm mua sắm và quản lý trực tuyến an toàn và mượt mà.

3.1.2 Yêu cầu hệ thống:

- Giao diện người dùng (UI/UX):
- Quản lý tài khoản: Người dùng có thể Đăng Nhập, Đăng Xuất, với admin thì có thể Tạo tài khoản, Sửa tài khoản, Xóa tài khoản và Phân quyền
- Quản lý sản phẩm: Cho phép quản trị viên thêm, sửa đổi và xóa sản phẩm một cách dễ dàng.
- Giỏ hàng và thanh toán: Cho phép người dùng thêm sản phẩm vào giỏ hàng và thanh toán một cách thuận tiện.
- Tìm kiếm và lọc sản phẩm: Hỗ trợ công cụ tìm kiếm và lọc sản phẩm dựa trên các tiêu chí như loại tranh, giá cả, và nghệ sĩ.
- Phản hồi và đánh giá sản phẩm: Cho phép người dùng đánh giá và viết phản hồi về sản phẩm sau khi mua hàng. Hiển thị các đánh giá và phản hồi để người dùng có thể tham khảo trước khi mua sản phẩm.
- Quản trị hệ thống: Giao diện quản trị dễ sử dụng cho quản trị viên để quản lý sản phẩm, đơn hàng, và người dùng.

- Tối ưu hiệu suất và bảo mật:

3.1.3 Các chức năng chính:

STT	Chức năng	Mô tả chức năng
1	Quản lý sản phẩm	Quản lý các sản phẩm trên trang web, bao gồm thêm, sửa đổi và xóa sản phẩm, phân loại vào các danh mục.
2	Quản lý đơn hàng	Xem danh sách đơn hàng, xác nhận đơn hàng, cập nhật trạng thái đơn hàng.
3	Quản lý người dùng	Quản lý thông tin người dùng, bao gồm xem, thêm, sửa đổi và xóa người dùng.
4	Xem báo cáo	Xem báo cáo về doanh số bán hàng, hoạt động của trang web.

Bảng 3.1: Bảng miêu tả chức năng admin

STT	Chức năng	Mô tả chức năng
1	Đăng nhập và đăng xuất	Đăng nhập vào tài khoản của mình để thực hiện các chức năng quản trị.
2	Thêm sản phẩm vào giỏ hàng	Chọn sản phẩm và thêm vào giỏ hàng để chuẩn bị thanh toán.
3	Thanh toán	Thực hiện thanh toán cho các sản phẩm trong giỏ hàng.
4	Đánh giá và phản hồi sản phẩm	Đánh giá và viết phản hồi về sản phẩm sau khi mua hàng.
5	Tìm kiếm và lọc sản phẩm	Tìm kiếm và lọc sản phẩm dựa trên các tiêu chí như loại Laptop, giá cả, và cấu hình.

6	Đăng ký thành viên	Đăng ký tài khoản thành viên để có thể sử dụng các tính năng đặc biệt và nhận ưu đãi.
---	--------------------	---------------------------------------------------------------------------------------

Bảng 3.2: Bảng miêu tả chức năng user

3.2 Thiết kế hệ thống

3.2.1 Xác định Usecase và actor

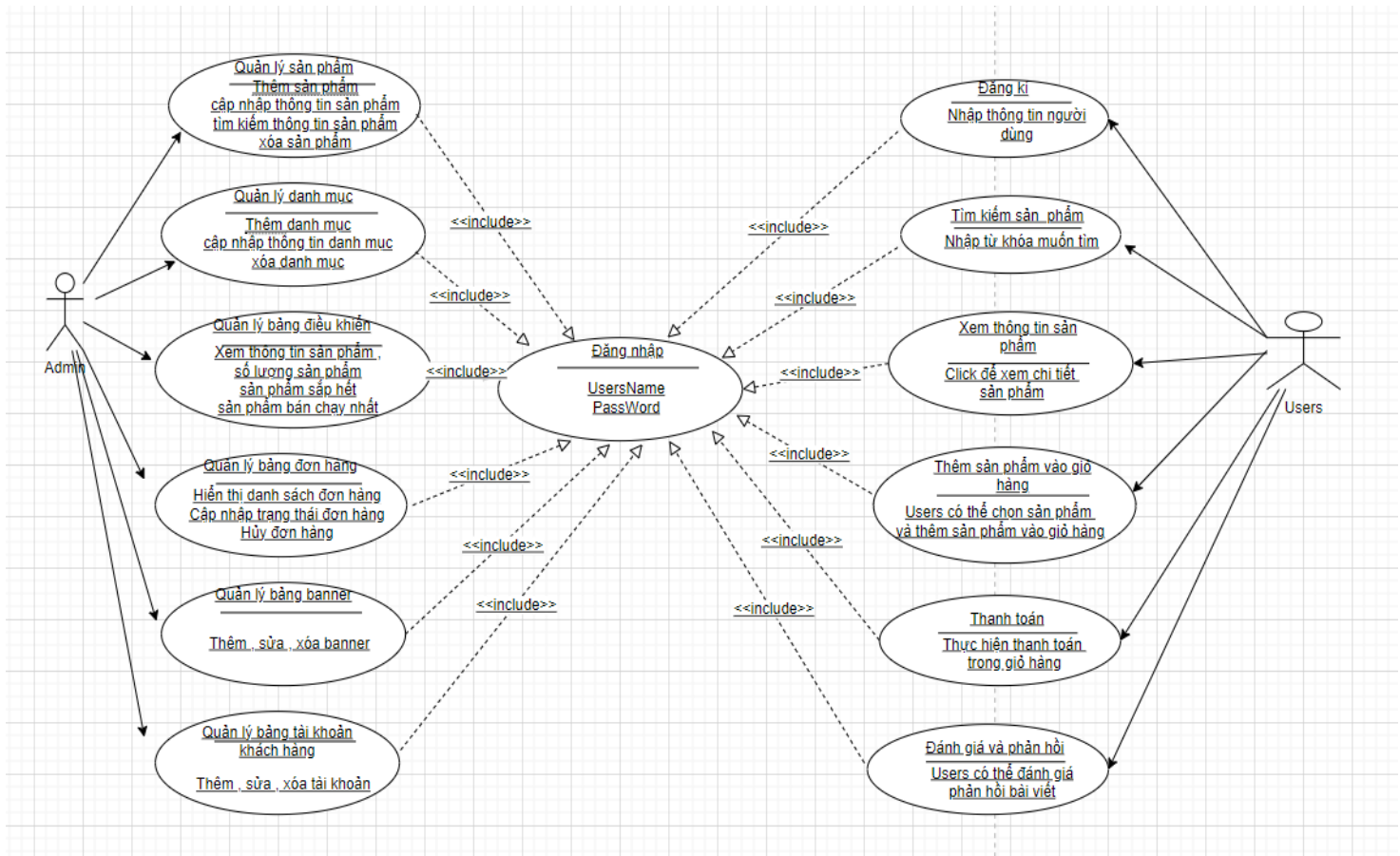
STT	Actor	Usecase
1	Admin	<p>Đăng nhập và đăng xuất:</p> <ul style="list-style-type: none"> - Admin có thể đăng nhập vào hệ thống để thực hiện các chức năng tương ứng. <p>Quản lý sản phẩm:</p> <ul style="list-style-type: none"> - Admin có thể thêm, sửa đổi, xóa sản phẩm và phân loại chúng vào các danh mục khác nhau. <p>Quản lý đơn hàng:</p> <ul style="list-style-type: none"> - Admin có thể xem danh sách đơn hàng, xác nhận đơn hàng và cập nhật trạng thái của chúng. <p>Quản lý người dùng:</p> <ul style="list-style-type: none"> - Admin có thể quản lý thông tin của người dùng, bao gồm xem, thêm, sửa đổi và xóa người dùng. <p>Xem báo cáo:</p>

		<ul style="list-style-type: none"> - Admin có thể xem báo cáo về doanh số bán hàng và hoạt động của trang web.
2	Users	<p>Đăng nhập và đăng xuất:</p> <ul style="list-style-type: none"> - Users có thể đăng nhập vào hệ thống để thực hiện các chức năng tương ứng. <p>Thêm sản phẩm vào giỏ hàng:</p> <ul style="list-style-type: none"> - Users có thể chọn sản phẩm và thêm vào giỏ hàng để chuẩn bị thanh toán. <p>Thanh toán:</p> <ul style="list-style-type: none"> - Users có thể thực hiện thanh toán cho các sản phẩm trong giỏ hàng. <p>Users có thể đánh giá và viết phản hồi về sản phẩm sau khi mua hàng.</p> <p>Xem lịch sử mua hàng:</p> <ul style="list-style-type: none"> - Users có thể xem lịch sử các đơn hàng đã mua trước đó. <p>Tìm kiếm và lọc sản phẩm:</p> <ul style="list-style-type: none"> - Users có thể tìm kiếm và lọc sản phẩm dựa trên các tiêu chí như loại Laptop, giá cả, và cấu hình.

		<p>Đăng ký thành viên:</p> <ul style="list-style-type: none"> - Users có thể đăng ký tài khoản thành viên để có thể sử dụng các tính năng đặc biệt và nhận ưu đãi.
--	--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Bảng 3.3: Xác định Usecase và actor

3.2.2 Biểu đồ ca tổng quát



Hình 3.1: Sơ đồ usecase tổng quát

Sơ đồ 2 là sơ đồ tổng quát bao gồm các chức năng như sau: Quản lý sản phẩm, đơn hàng, tài khoản, danh mục, kết quả doanh số. Users, Xem thông tin, tìm kiếm... Tất cả các chức năng này đều yêu cầu phải đăng nhập trước.

3.3. Đặc tả usecase

3.3.1. Use case đăng nhập

Tên Use case	Đăng nhập
--------------	-----------

Tác nhân	Admin, khách hàng
Mô tả	Mô tả hoạt động đăng nhập vào hệ thống của actor.
Điều kiện tiên quyết	Hệ thống đã được khởi động sẵn sàng, actor đã được đăng ký tài khoản trước đó
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Actor khởi động giao diện đăng nhập 2. Hệ thống hiển thị và yêu cầu nhập thông tin 3. Actor nhập thông tin cần thiết, chọn nút đăng nhập 4. Hệ thống kiểm tra thông tin
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Actor nhập sai thông tin - Hệ thống báo lỗi và yêu cầu nhập lại - Actor lặp lại bước 3
Điều kiện cuối	Không có

Bảng 3.4: Use case đăng nhập

3.3.2. Use case đăng ký

Tên Use case	Đăng ký
Tác nhân	Khách Hàng
Mô tả	Mô tả hoạt động đăng ký vào hệ thống của actor.
Điều kiện tiên quyết	Hệ thống đã được khởi động sẵn sàng, actor chưa có tài khoản trước đó
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Actor khởi động giao diện đăng ký 2. Hệ thống hiển thị và yêu cầu nhập thông tin 3. Actor nhập thông tin cần thiết, chọn nút đăng ký 4. Hệ thống kiểm tra thông tin và thông báo đăng ký thành công
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Actor nhập sai thông tin <p>Hệ thống báo lỗi và yêu cầu nhập lại</p>

	\ Actor lặp lại bước 3 Hệ thống báo lỗi và yêu cầu nhập lại \ Actor lặp lại bước 3 Hệ thống báo lỗi và yêu cầu nhập lại \ Actor lặp lại bước 3 - Hệ thống báo lỗi và yêu cầu nhập lại - Actor lặp lại bước 3
Điều kiện cuối	Dữ liệu đăng ký được lưu lại để đăng nhập

Bảng 3.5: Use case đăng ký

3.3.3 Use case đăng xuất

Tên Use case	Đăng xuất
Tác nhân	Khách Hàng, Admin
Mô tả	Mô tả hoạt động đăng xuất ra hệ thống của actor
Điều kiện tiên quyết	Hệ thống đã được khởi động sẵn sàng, actor đã đăng nhập vào hệ thống trước đó
Dòng sự kiện chính	1. Actor chọn nút đăng xuất 2. Hệ thống đăng xuất actor ra khỏi hệ thống
Dòng sự kiện phụ	Không có
Điều kiện cuối	Không có

Bảng 3.6: Use case đăng xuất

3.3.4. Use case xem sản phẩm

Tên Use case	Xem sản phẩm
Tác nhân	Khách Hàng
Mô tả	Mô tả hoạt động xem sản phẩm của actor

Điều kiện tiên quyết	Hệ thống đã được khởi động sẵn sàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Actor xem và lựa chọn sản phẩm. 2. Actor nhấn vào hình ảnh miêu tả sản phẩm. 3. Actor được chuyển đến xem chi tiết sản phẩm.
Dòng sự kiện phụ	Không có
Điều kiện cuối	Không có

Bảng 3.7: Use case xem sản phẩm

3.3.5 Use case mua sản phẩm

Tên Use case	Mua sản phẩm
Tác nhân	Khách Hàng
Mô tả	Mô tả hoạt động xem sản phẩm của actor
Điều kiện tiên quyết	Hệ thống đã được khởi động sẵn sàng, actor đã đăng nhập vào hệ thống trước đó
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Actor nhấn vào nút mua sản phẩm. 2. Sản phẩm được chuyển vào giỏ hàng. 3. Actor nhấn vào nút thanh toán. 4. Hệ thống yêu cầu actor nhập thông tin giao hàng. 5. Actor nhập thông tin giao hàng và nhấn nút Thanh toán. 6. Hệ thống kiểm tra thông tin và chuyển đến trang thanh toán. 7. Actor chọn phương thức thanh toán. 8. Hệ thống thông báo đã thanh toán cho actor.
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Actor nhập sai thông tin thông tin đơn hàng - Hệ thống báo lỗi và yêu cầu nhập lại

	- Actor lặp lại bước 3
Điều kiện cuối	Hệ thống lưu lại đơn hàng vào hệ thống

Bảng 3.8: Use case mua sản phẩm

3.3.6 Use case tìm kiếm sản phẩm

Tên Use case	Tìm kiếm sản phẩm
Tác nhân	Khách Hàng
Mô tả	Mô tả hoạt động xem sản phẩm của actor
Điều kiện tiên quyết	Hệ thống đã được khởi động sẵn sàng.
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Actor nhập tên cần tìm kiếm vào ô tìm kiếm 2. Actor nhấn vào nút tìm kiếm 3. Hệ thống kiểm tra thông tin và hiển thị kết quả tìm kiếm
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Actor nhập sai thông tin tìm kiếm - Hệ thống hiện thông báo không tìm thấy - Actor lặp lại bước 3
Điều kiện cuối	Không

Bảng 3.9: Use case tìm kiếm sản phẩm

3.3.7 Use case Thêm sản phẩm

Tên Use case	Thêm sản phẩm
Tác nhân	Admin
Mô tả	Use case này mô tả chức năng quản lý sản phẩm của actor Admin
Điều kiện cần	Actor Admin cần đăng nhập được vào hệ thống, chọn chức năng quản lý sản phẩm, hệ thống sẵn sàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Admin yêu cầu chức năng thêm sản phẩm. 2. Hệ thống hiển thị và yêu cầu nhập thông tin 3. Admin nhập thông tin, yêu cầu ghi nhận.

	4. Hệ thống ghi nhận, thông báo đã thêm thành công
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Hệ thống báo lỗi không nhập được. - Admin lặp lại bước 3
Điều kiện cuối	Nếu actor chọn lưu thì thông tin về sản phẩm được ghi nhận vào hệ thống

Bảng 3.10: Use case Thêm sản phẩm

3.3.8 Use case Sửa sản phẩm

Tên Use case	Sửa sản phẩm
Tác nhân	Admin
Mô tả	Use case này mô tả chức năng sửa thông tin 1 sản phẩm của Admin
Điều kiện cần	Actor Admin cần đăng nhập được vào hệ thống, chọn chức năng quản lý sản phẩm, hệ thống sẵn sàng.
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Actor Admin chọn sản phẩm cần sửa thông tin. 2. Hệ thống hiển thị thông tin sản phẩm được chọn. 3. Admin cập nhật lại thông tin, yêu cầu ghi nhận. 4. Hệ thống ghi nhận, thông báo và hiển thị danh sách
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Hệ thống báo lỗi không nhập được. - Admin lặp lại bước 3
Điều kiện cuối	Nếu actor Admin chọn lưu thì thông tin về sản phẩm được cập nhật lại vào hệ thống

Bảng 3.11: Use case Sửa sản phẩm

3.3.9 Use case Xóa sản phẩm

Tên Use case	Xóa sản phẩm
Tác nhân	Admin
Mô tả	Use case này mô tả chức năng xóa 1 sản phẩm của actor Admin

Điều kiện cần	Actor Admin cần đăng nhập được vào hệ thống, chọn chức năng quản lý sản phẩm, hệ thống sẵn sàng.
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Actor Admin chọn sản phẩm cần xóa. 2. Hệ thống hỏi xác nhận xóa sản phẩm. 3. Actor Admin xác nhận. 4. Hệ thống xóa thương hiệu được chọn, thông báo và hiển thị lại danh sách.
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Actor Admin chọn hủy thao tác - Hệ thống hủy việc xóa và liệt kê lại danh sách sản phẩm.
Điều kiện cuối	Sản phẩm được xóa thành công nếu actor xác nhận xóa

Bảng 3.12: Use case Xóa sản phẩm

3.3.10 Use case Thêm danh mục

Tên Use case	Thêm danh mục
Tác nhân	Admin
Mô tả	Use case này mô tả chức năng quản lý danh mục của actor Admin.
Điều kiện cần	Actor Admin cần đăng nhập được vào hệ thống, chọn chức năng Quản lý danh mục, hệ thống sẵn sàng.
Dòng sự kiện chính	<p>Thêm thương hiệu</p> <ol style="list-style-type: none"> 1. Admin yêu cầu chức năng thêm danh mục. 2. Hệ thống hiển thị và yêu cầu nhập thông tin. 3. Admin nhập thông tin, yêu cầu ghi nhận. 4. Hệ thống ghi nhận, thông báo đã thêm thành công
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Hệ thống báo lỗi không nhập được. - Admin lặp lại bước 3.

Điều kiện cuối	Nếu actor chọn lưu thì thông tin về danh mục được ghi nhận vào hệ thống
----------------	-------------------------------------------------------------------------

Bảng 3.13: Use case Thêm danh mục

3.3.11 Use case sửa danh mục

Tên Use case	Sửa danh mục
Tác nhân	Admin
Mô tả	Use case này mô tả chức năng sửa thông tin 1 danh mục của Admin
Điều kiện cần	Actor Admin cần đăng nhập được vào hệ thống, chọn chức năng Quản lý danh mục, hệ thống sẵn sàng.
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Actor Admin chọn danh mục cần sửa thông tin. 2. Hệ thống hiển thị thông tin danh mục được chọn. 3. Admin cập nhật lại thông tin, yêu cầu ghi nhận. 4. Hệ thống ghi nhận, thông báo và hiển thị danh sách
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Hệ thống báo lỗi không nhập được. - Admin lặp lại bước 3.
Điều kiện cuối	Nếu actor Admin chọn lưu thì thông tin về danh mục được cập nhật lại vào hệ thống

Bảng 3.14: Use case sửa danh mục.

3.3.12 Use case Xóa danh mục

Tên Use case	Xóa danh mục
Tác nhân	Admin
Mô tả	Use case này mô tả chức năng xóa 1 danh mục của actor Admin
Điều kiện cần	Actor Admin cần đăng nhập được vào hệ thống, chọn chức năng quản lý danh mục, hệ thống sẵn sàng

Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Actor Admin chọn danh mục cần xóa. 2. Hệ thống hỏi xác nhận xóa danh mục. 3. Actor Admin xác nhận. 4. Hệ thống xóa danh mục được chọn, thông báo và hiển thị lại danh sách
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Actor Admin chọn hủy thao tác - Hệ thống hủy việc xóa và liệt kê lại danh sách danh mục
Điều kiện cuối	Các hãg được xóa thành công nếu actor xác nhận xóa

Bảng 3.15: Use case Xóa danh mục

3.3.13 Use case Xóa đơn hàng

Tên Use case	Xóa đơn hàng
Tác nhân	Admin
Mô tả	Use case này mô tả chức năng xóa 1 đơn hàng của actor Admin
Điều kiện cần	Actor Admin cần đăng nhập được vào hệ thống, chọn chức năng Quản lý đơn hàng, hệ thống sẵn sàng.
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Actor Admin chọn đơn hàng cần xóa. 2. Hệ thống hỏi xác nhận xóa đơn hàng. 3. Actor Admin xác nhận. 4. Hệ thống xóa đơn hàng được chọn, thông báo và hiển thị lại danh sách
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Actor Admin chọn hủy thao tác - Hệ thống hủy việc xóa và liệt kê lại danh sách đơn hàng
Điều kiện cuối	Đơn hàng được xóa thành công nếu actor xác nhận xóa

Bảng 3.16: Use case Xóa đơn hàng

3.3.14 Use case Thêm tài khoản khách hàng

Tên Use case	Thêm tài khoản khách hàng
--------------	---------------------------

Tác nhân	Admin
Mô tả	Use case này mô tả chức năng quản lý tài khoản khách hàng của actor Admin và Nhân viên
Điều kiện cần	Actor Admin cần đăng nhập được vào hệ thống, chọn chức năng Quản lý tài khoản khách hàng, hệ thống sẵn sàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Admin yêu cầu chức năng thêm tài khoản khách hàng. 2. Hệ thống hiển thị và yêu cầu nhập thông tin. 3. Admin nhập thông tin, yêu cầu ghi nhận. 4. Hệ thống ghi nhận, thông báo đã thêm thành
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Hệ thống báo lỗi không nhập được - Admin lặp lại bước 3
Điều kiện cuối	Nếu actor chọn lưu thì thông tin về tài khoản khách hàng được ghi nhận vào hệ thống.

Bảng 3.17: Use case Thêm tài khoản khách hàng

3.3.15 Use case sửa tài khoản khách hàng

Tên Use case	Sửa tài khoản khách hàng
Tác nhân	Admin
Mô tả	Use case này mô tả chức năng sửa thông tin 1 tài khoản khách hàng của Admin
Điều kiện cần	Actor Admin và Nhân viên cần đăng nhập được vào hệ thống, chọn chức năng Quản lý tài khoản khách hàng, hệ thống sẵn sàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Actor Admin hoặc Nhân viên chọn tài khoản khách hàng cần sửa thông tin. 2. Hệ thống hiển thị thông tin tài khoản khách hàng được chọn. 3. Admin hoặc Nhân viên cập nhật lại thông tin, yêu cầu ghi nhận.

	4. Hệ thống ghi nhận, thông báo và hiển thị danh sách
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Hệ thống báo lỗi không nhập được - Admin lặp lại bước 3
Điều kiện cuối	Nếu actor Admin chọn lưu thì thông tin về tài khoản khách hàng được cập nhật lại vào hệ thống

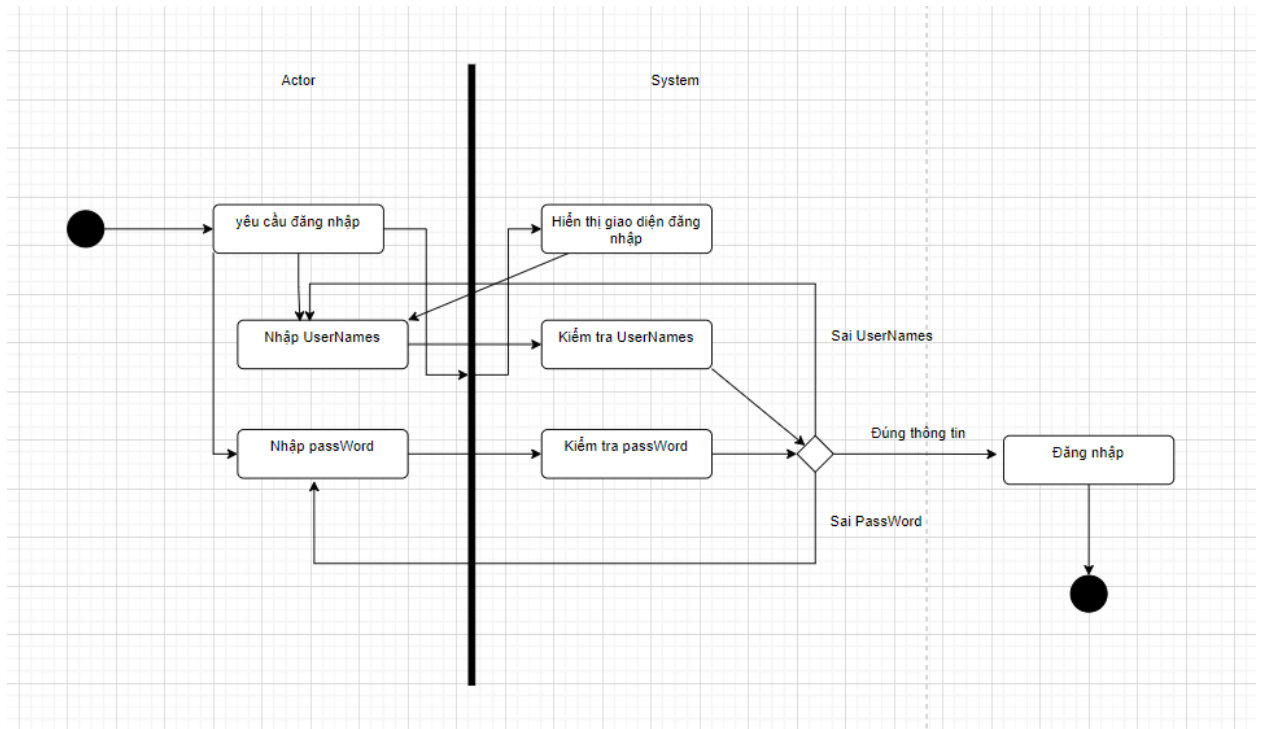
Bảng 3.18: Use case sửa tài khoản khách hàng

3.3.16 Use case Xóa tài khoản khách hàng

Tên Use case	Xóa tài khoản khách hàng
Tác nhân	Admin
Mô tả	Use case này mô tả chức năng xóa 1 tài khoản khách hàng của actor Admin
Điều kiện cần	Actor Admin và Nhân viên cần đăng nhập được vào hệ thống, chọn chức năng quản lý tài khoản khách hàng, hệ thống sẵn sàng
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Actor Admin hoặc Nhân viên chọn tài khoản khách hàng cần xóa. 2. Hệ thống hỏi xác nhận xóa tài khoản khách hàng. 3. Actor Admin hoặc Nhân viên xác nhận. 4. Hệ thống xóa tài khoản khách hàng được chọn, thông báo và hiển thị lại danh sách.
Dòng sự kiện phụ	<ul style="list-style-type: none"> - Actor Admin chọn hủy thao tác - Hệ thống hủy việc xóa và liệt kê lại danh sách tài khoản khách hàng.
Điều kiện cuối	Khách hàng được xóa thành công nếu actor xác nhận xóa

Bảng 3.19: Use case Xóa tài khoản khách hàng

3.4. Biểu đồ hoạt động



Hình 3.2: Biểu đồ hoạt động đăng nhập

Trong biểu đồ này:

Bước 1: Khách hàng nhấn vào nút "Đăng nhập" trên trang web.

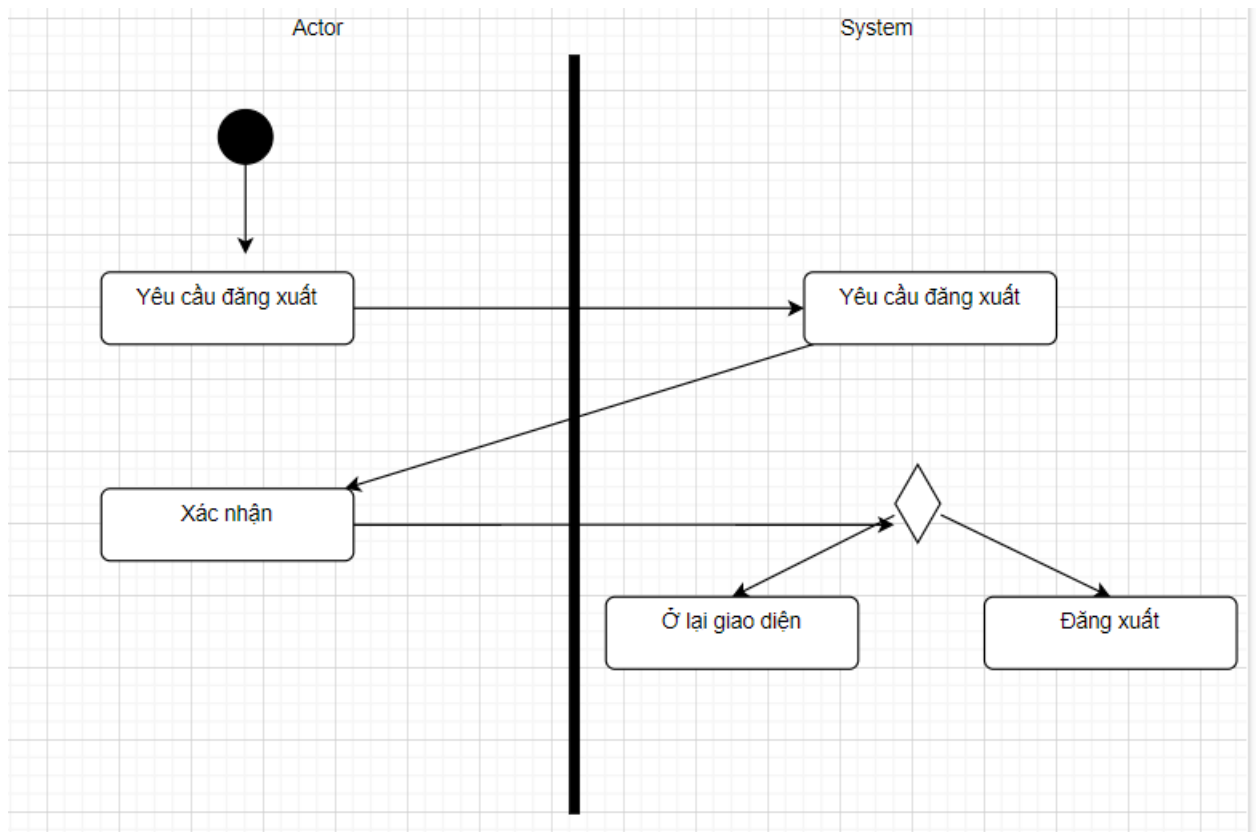
Bước 2: Hệ thống hiển thị giao diện đăng nhập.

Bước 3: Khách hàng nhập tên đăng nhập và mật khẩu vào giao diện đăng nhập.

Bước 4: Thông tin đăng nhập được gửi từ trang web lên hệ thống.

Bước 5: Hệ thống xác thực thông tin đăng nhập và kiểm tra tính hợp lệ của nó.

Bước 6: Nếu đăng nhập thành công, hệ thống cho phép khách hàng truy cập vào tài khoản.



Hình 3.3: Biểu đồ hoạt động đăng xuất

Trong biểu đồ này:

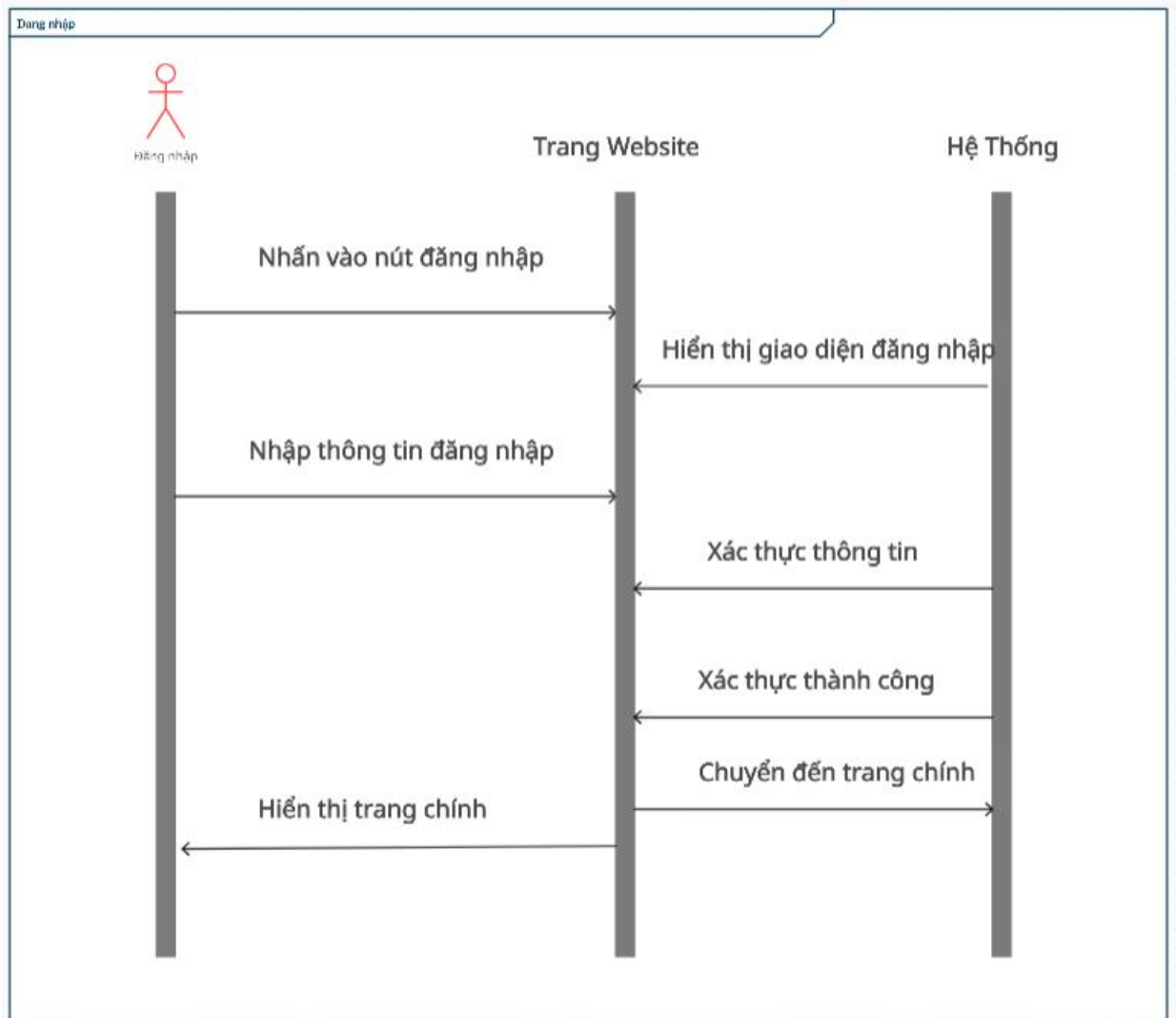
Bước 1: Người dùng nhấn vào nút "Đăng xuất" trên trang web.

Bước 2: Hệ thống nhận yêu cầu đăng xuất từ người dùng và xác nhận.

Sau khi xác nhận, hệ thống thực hiện xử lý yêu cầu đăng xuất và người dùng sẽ được đăng xuất khỏi trang web.

3.5. Biểu đồ tuần tự

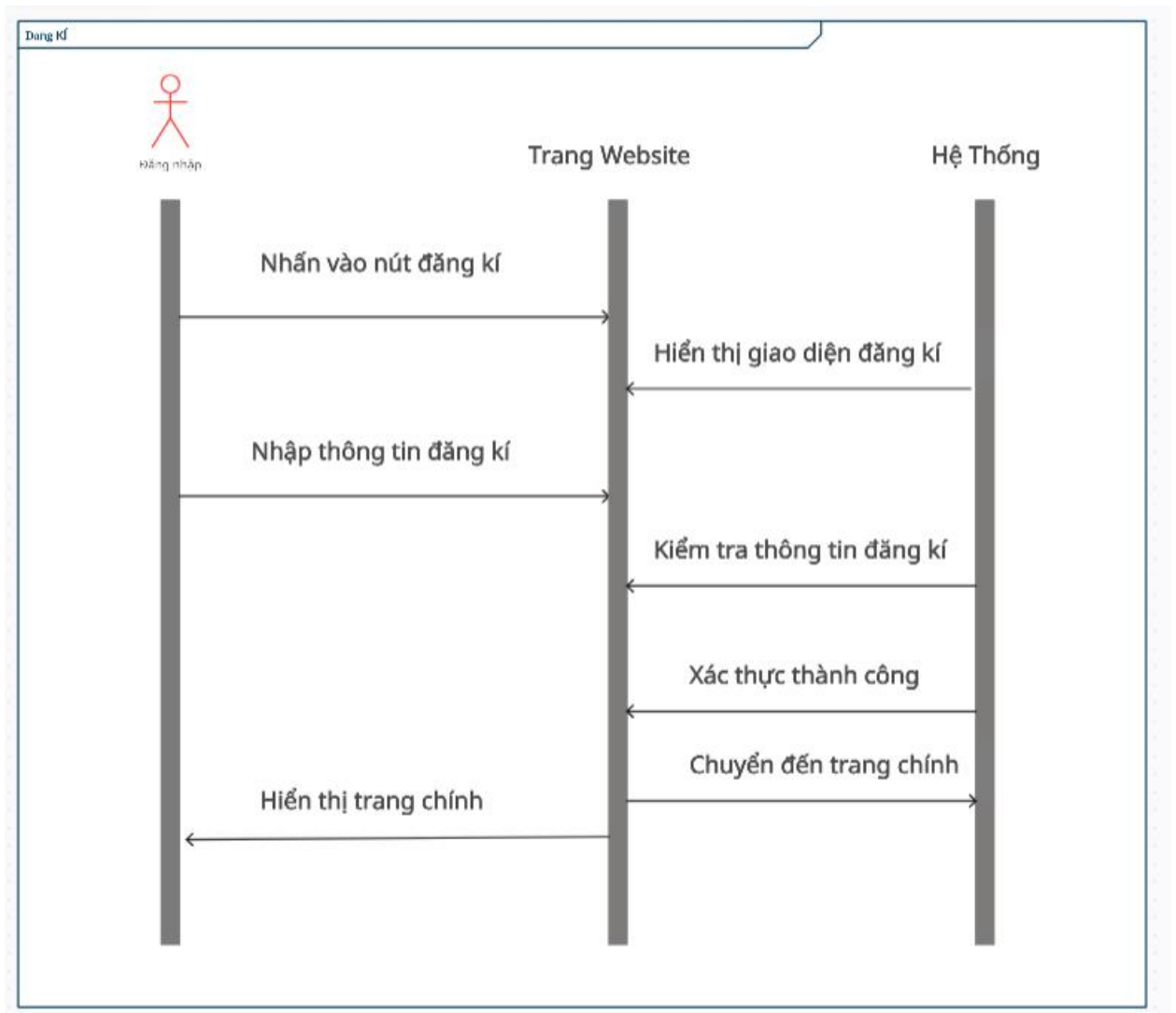
a) Users:



Hình 3.4: Biểu đồ tuần tự Đăng nhập

Trong biểu đồ này:

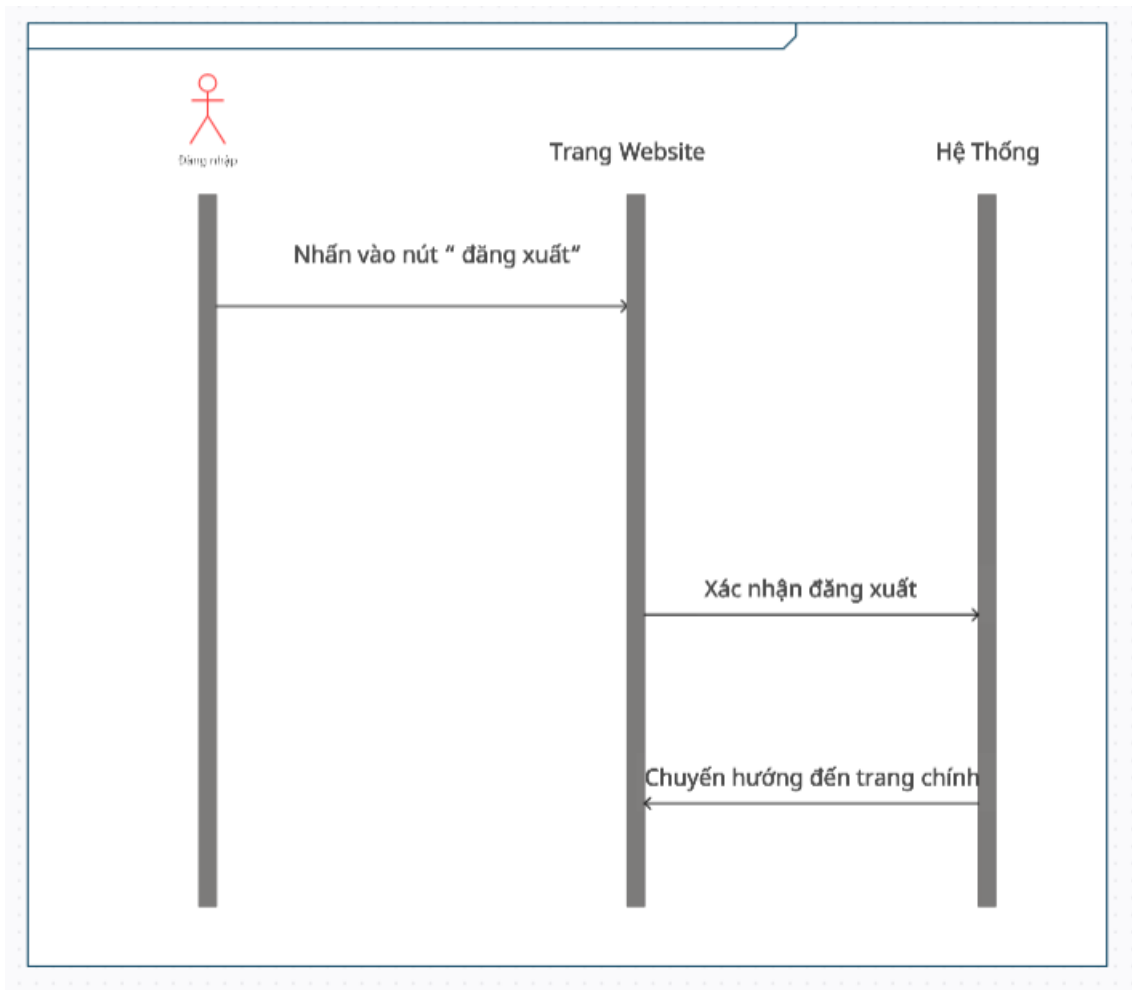
- Quá trình đăng nhập bắt đầu khi người dùng nhấn vào nút "Đăng nhập" trên trang web.
- Trang web hiển thị giao diện đăng nhập và người dùng nhập thông tin đăng nhập.
- Thông tin đăng nhập được gửi đến hệ thống để xác thực.
- Nếu xác thực thành công, hệ thống chuyển hướng người dùng đến trang chính của trang web.



Hình 3.5: Biểu đồ tuần tự Đăng kí

Trong biểu đồ này:

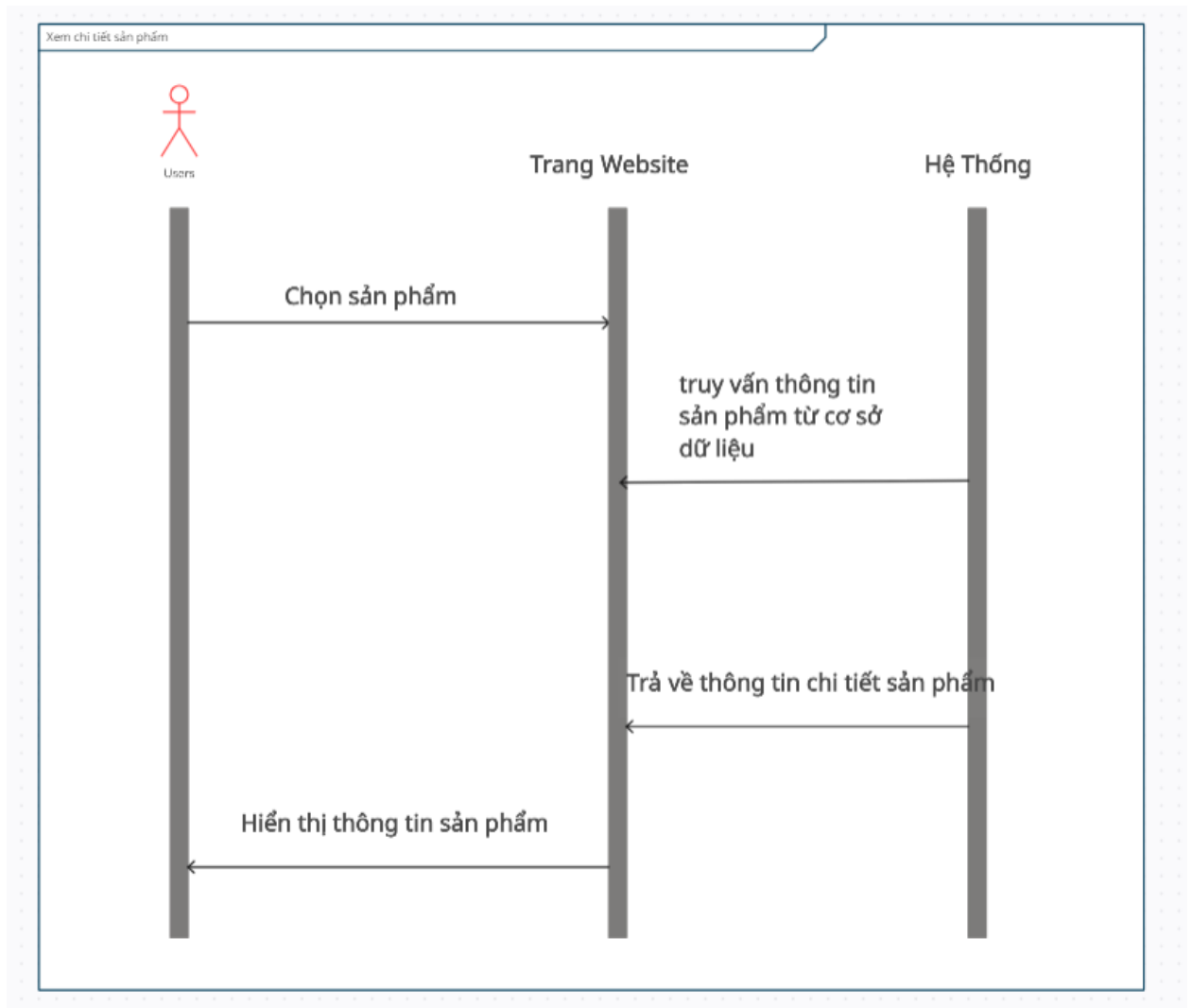
- Quá trình đăng ký thành viên bắt đầu khi người dùng nhấn vào nút "Đăng ký" trên trang web.
- Trang web hiển thị giao diện đăng ký và người dùng nhập thông tin đăng ký.
- Thông tin đăng ký được gửi đến hệ thống để kiểm tra.
- Nếu thông tin hợp lệ, hệ thống xác thực thành công và chuyển hướng người dùng đến trang chính của trang web.



Hình 3.6: Biểu đồ tuần tự Đăng xuất

Trong biểu đồ này:

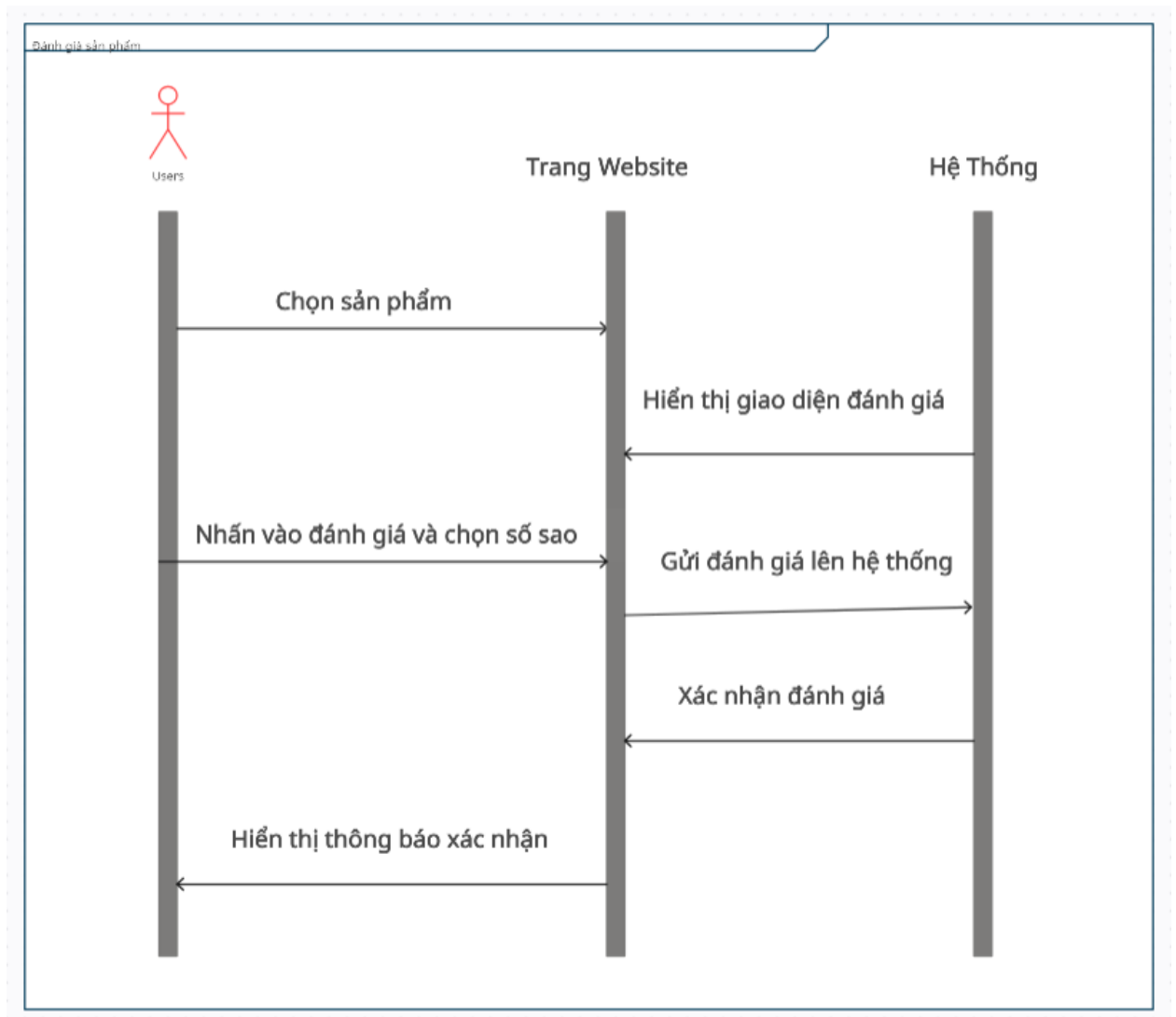
- Quá trình đăng xuất bắt đầu khi người dùng nhấn vào nút "Đăng xuất" trên trang web.
- Trang web xác nhận đăng xuất và chuyển hướng người dùng đến trang đăng nhập.
- Không có thông tin giao tiếp nào được thực hiện với hệ thống trong quá trình đăng xuất, vì quá trình này chỉ liên quan đến giao diện người dùng.



Hình 3.7: Biểu đồ tuần tự Xem chi tiết sản phẩm

Trong biểu đồ này:

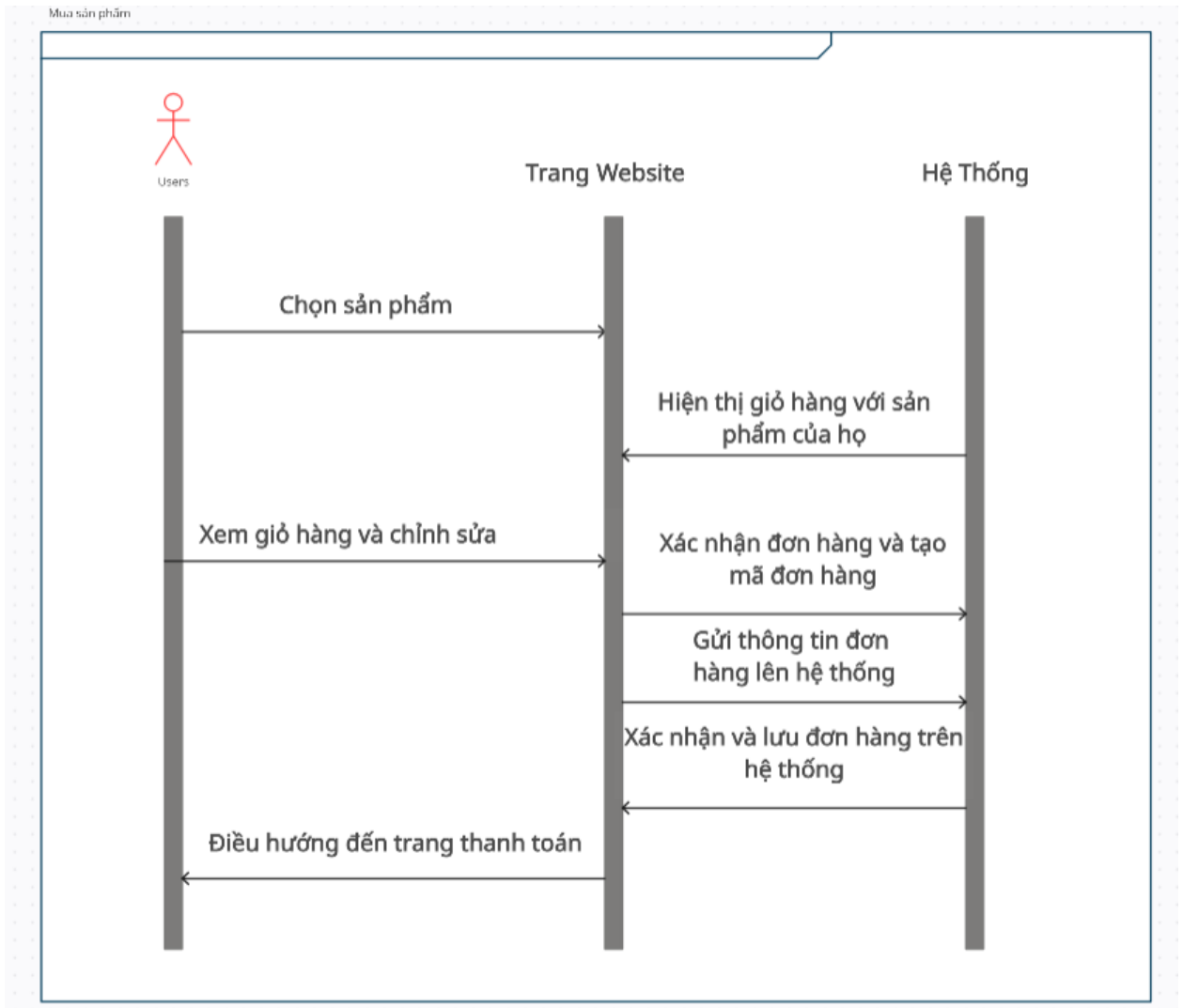
- Quá trình xem sản phẩm bắt đầu khi người dùng chọn một sản phẩm trên trang web.
- Trang web gửi yêu cầu truy vấn thông tin sản phẩm đến hệ thống.
- Hệ thống truy vấn thông tin chi tiết của sản phẩm từ cơ sở dữ liệu.
- Hệ thống trả về thông tin chi tiết của sản phẩm cho trang web.
- Trang web hiển thị thông tin chi tiết của sản phẩm cho người dùng.



Hình 3.8: Biểu đồ tuần tự Đánh giá sản phẩm

Trong biểu đồ này:

- Quá trình đánh giá sản phẩm bắt đầu khi người dùng chọn một sản phẩm để đánh giá trên trang web.
- Trang web hiển thị giao diện đánh giá và nhập số sao cho người dùng.
- Người dùng nhập đánh giá và chọn số sao của họ vào trang web.
- Thông tin đánh giá được gửi từ trang web lên hệ thống.
- Hệ thống xác nhận nhận được đánh giá từ người dùng và cập nhật vào cơ sở dữ liệu.
- Sau đó, trang web hiển thị thông báo xác nhận cho người dùng.

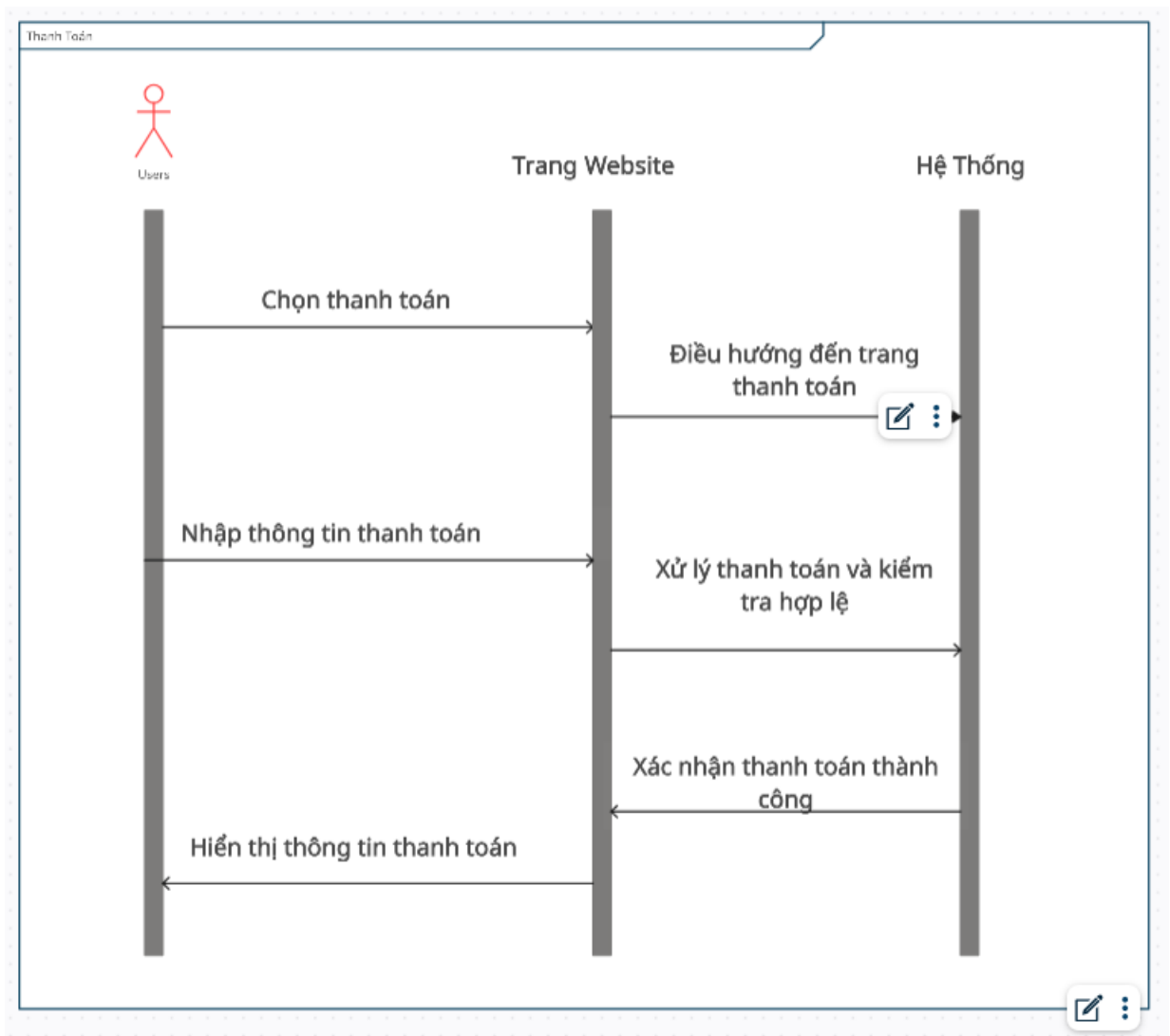


Hình 3.9: Biểu đồ tuần tự Mua sản phẩm

Trong biểu đồ này:

- Quá trình mua sản phẩm bắt đầu khi người dùng chọn một sản phẩm để mua trên trang web.
- Trang web hiển thị giỏ hàng với sản phẩm đã được chọn và cho phép người dùng xem và chỉnh sửa thông tin giỏ hàng.
- Người dùng xác nhận đơn hàng và trang web tạo mã đơn hàng.
- Thông tin đơn hàng được gửi từ trang web lên hệ thống để xác nhận và lưu vào cơ sở dữ liệu.
- Hệ thống xác nhận và lưu đơn hàng vào cơ sở dữ liệu và gửi thông báo xác nhận về trang thanh toán.

- Sau đó, trang web chuyển hướng người dùng đến trang thanh toán để hoàn tất quá trình mua hàng.

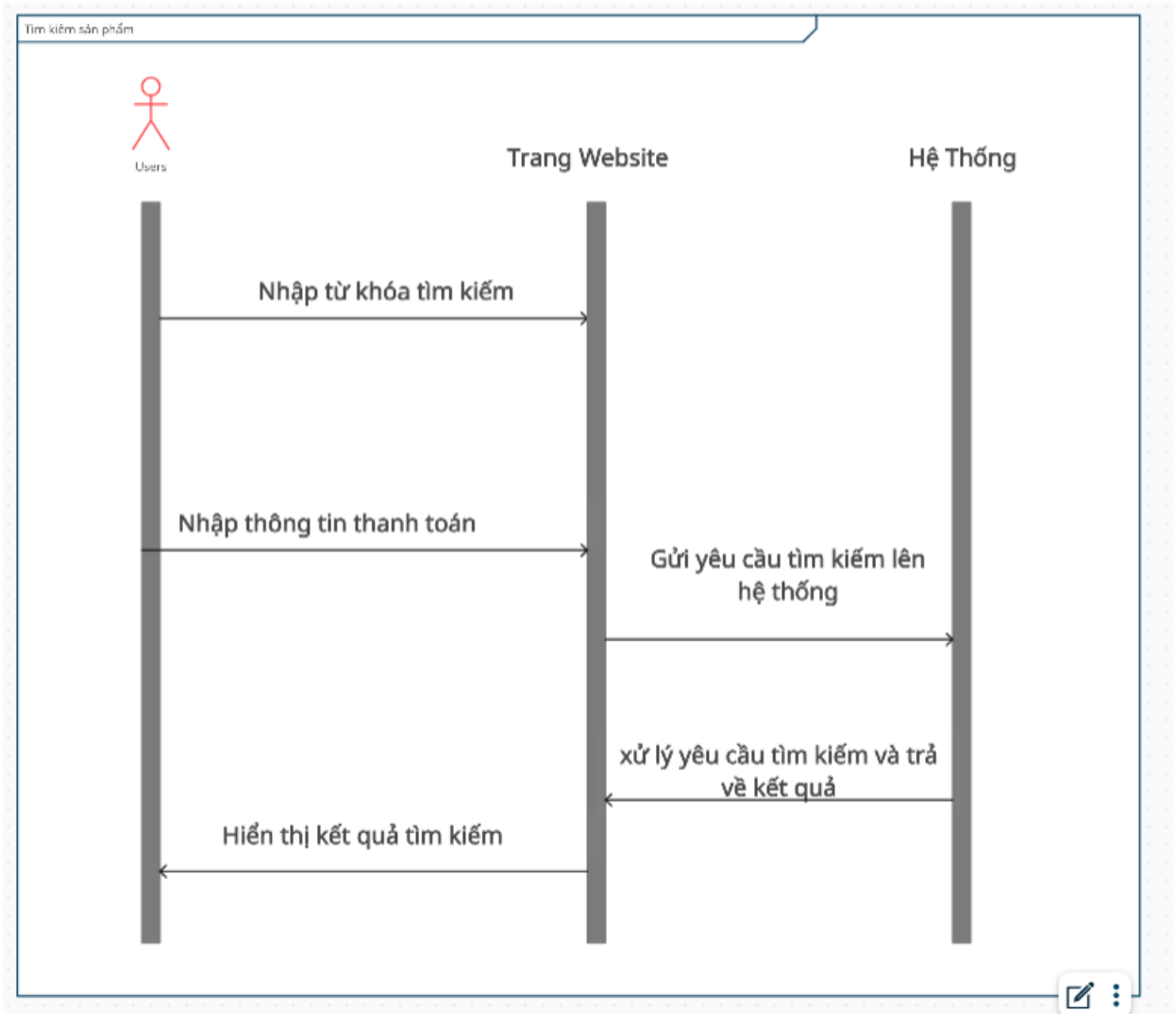


Hình 3.10: Biểu đồ tuần tự Thanh toán

Trong biểu đồ này:

- Quá trình thanh toán bắt đầu khi người dùng chọn phương thức thanh toán trên trang web.
- Trang web chuyển hướng người dùng đến trang thanh toán để nhập thông tin thanh toán.
- Người dùng nhập thông tin thanh toán và gửi thông tin đến trang web.

- Trang web gửi thông tin thanh toán đến hệ thống để xử lý và kiểm tra tính hợp lệ của thanh toán.
- Hệ thống xác nhận thanh toán thành công và gửi thông báo về trang web.
- Trang web hiển thị kết quả của quá trình thanh toán cho người dùng.

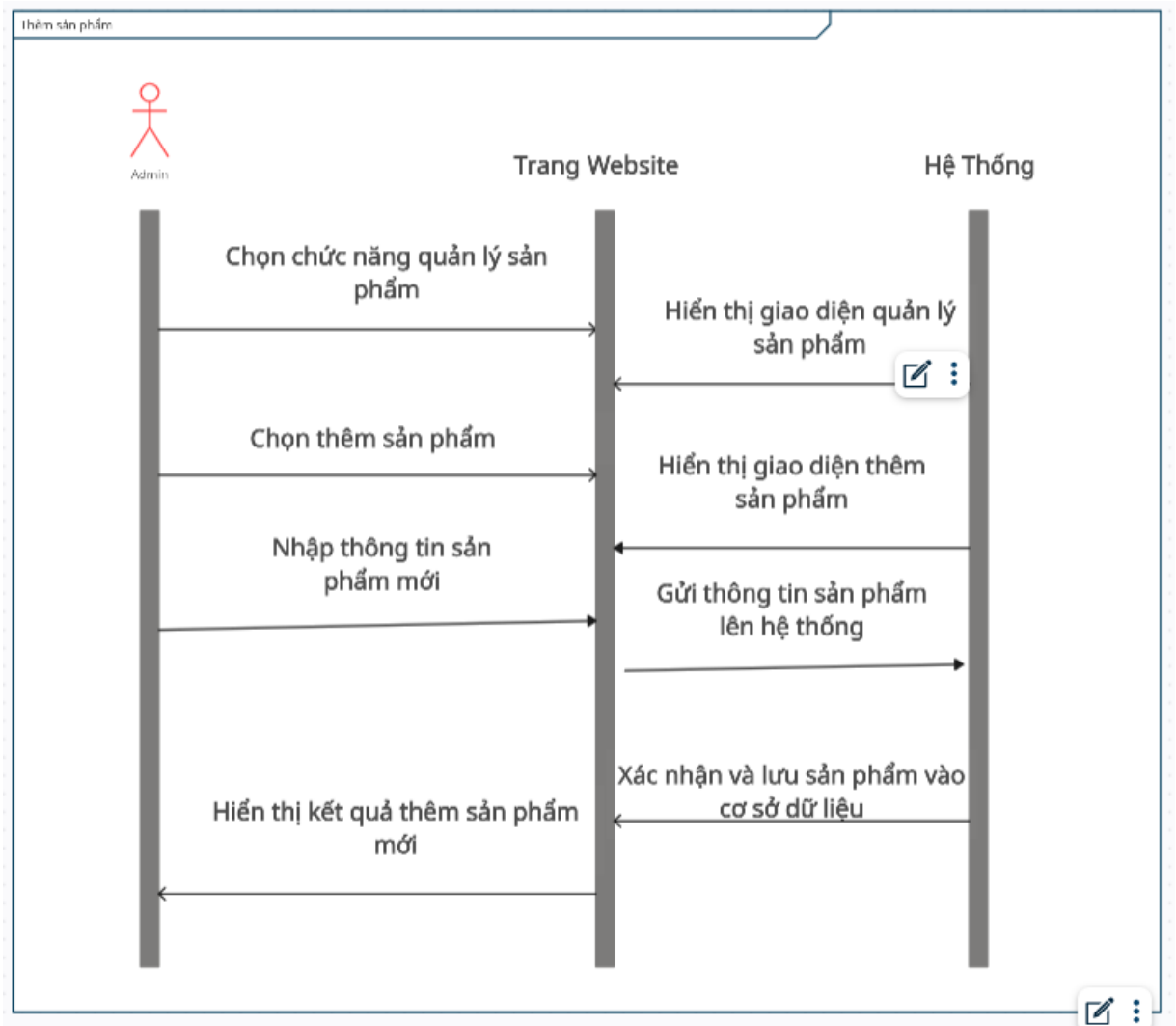


Hình 3.11: Biểu đồ tuần tự Tìm kiếm sản phẩm

Trong biểu đồ này:

- Quá trình tìm kiếm bắt đầu khi người dùng nhập từ khóa tìm kiếm vào trang web.
- Trang web gửi yêu cầu tìm kiếm với từ khóa đã nhập lên hệ thống.
- Hệ thống xử lý yêu cầu tìm kiếm và trả về kết quả tìm kiếm tương ứng.
- Trang web hiển thị kết quả tìm kiếm cho người dùng.

b) Admin:

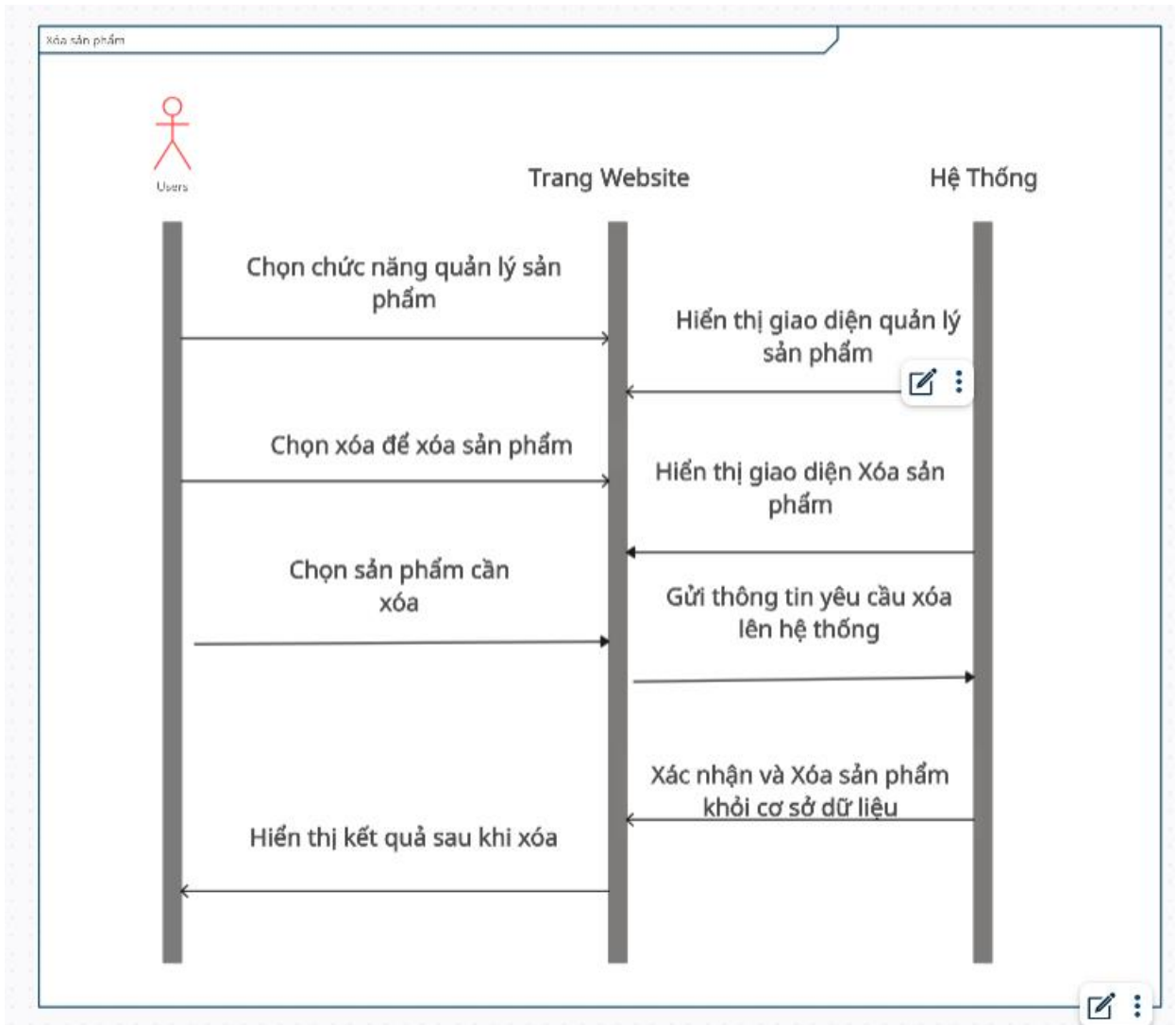


Hình 3.12: Biểu đồ tuần tự Thêm sản phẩm

Trong biểu đồ này:

- Quá trình thêm sản phẩm bắt đầu khi admin chọn chức năng quản lý sản phẩm trên trang web.
- Admin chọn thêm sản phẩm để thêm mới.
- Trang web hiển thị giao diện thêm sản phẩm.
- Admin nhập thông tin sản phẩm mới.
- Thông tin sản phẩm mới được gửi từ trang web lên hệ thống để xác nhận và lưu vào cơ sở dữ liệu.

- Hệ thống xác nhận và lưu sản phẩm vào cơ sở dữ liệu.
- Trang web hiển thị kết quả sau khi sản phẩm được thêm vào cơ sở dữ liệu.

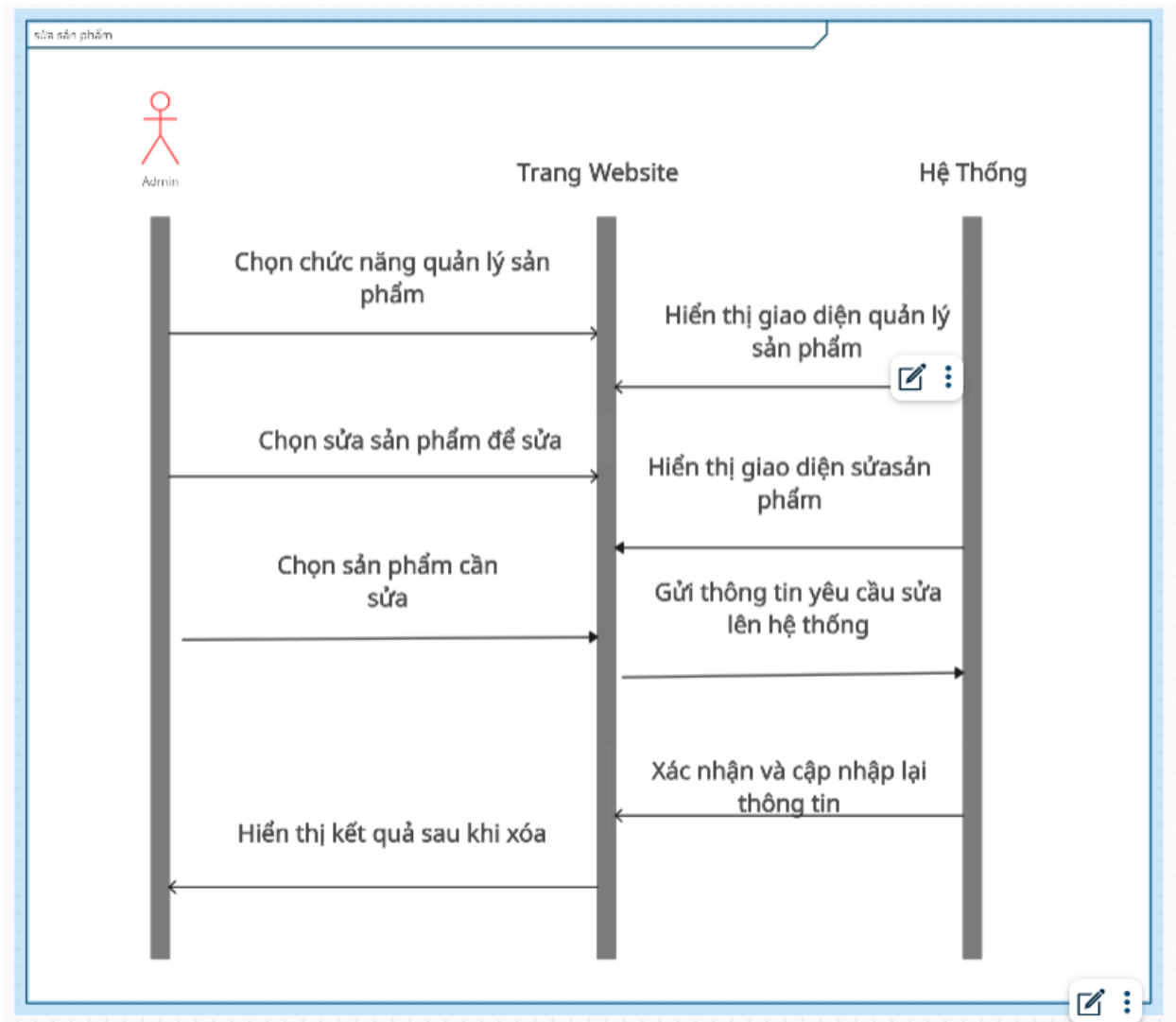


Hình 3.13: Biểu đồ tuần tự Xóa sản phẩm

Trong biểu đồ này:

- Quá trình xóa sản phẩm bắt đầu khi admin chọn chức năng quản lý sản phẩm trên trang web.
- Admin chọn xóa sản phẩm để thực hiện việc loại bỏ sản phẩm khỏi danh sách.
- Trang web hiển thị giao diện xóa sản phẩm và cho phép admin chọn sản phẩm cần xóa.

- Admin chọn sản phẩm cần xóa và gửi yêu cầu xóa sản phẩm lên hệ thống.
- Hệ thống xác nhận và thực hiện xóa sản phẩm khỏi cơ sở dữ liệu.
- Trang web hiển thị kết quả sau khi sản phẩm đã được xóa khỏi cơ sở dữ liệu.



Hình 3.14: Biểu đồ tuần tự sửa sản phẩm

Trong biểu đồ này:

- Quá trình sửa sản phẩm bắt đầu khi admin chọn chức năng quản lý sản phẩm trên trang web.
- Admin chọn sửa sản phẩm để thực hiện việc chỉnh sửa thông tin của sản phẩm.

- Trang web hiển thị giao diện sửa sản phẩm và cho phép admin chọn sản phẩm cần sửa.
- Admin chọn sản phẩm cần sửa và gửi yêu cầu sửa sản phẩm lên hệ thống.
- Hệ thống xác nhận và thực hiện cập nhật thông tin của sản phẩm trong cơ sở dữ liệu.
- Trang web hiển thị kết quả sau khi thông tin của sản phẩm đã được cập nhật.

CHƯƠNG 4. THIẾT KẾ CƠ SỞ DỮ LIỆU VÀ GIAO DIỆN WEBSITE

4.1 Thiết kế cơ sở dữ liệu

4.1.1 Danh sách các bảng

STT	Trường	Kiểu & Độ rộng	Mô tả
1	UserId	int	Khóa chính duy nhất, ID của người dùng
2	UserName	nvarchar(100)	Tên đăng nhập của người dùng
3	UserPassword	nvarchar(50)	Mật khẩu được mã hóa của người dùng
4	UserPhone	nvarchar(14)	Số điện thoại của người dùng
5	UserEmail	nvarchar(50)	Địa chỉ email của người dùng
6	UserAddress	nvarchar(200)	Địa chỉ của người dùng
7	UserRole	nvarchar(50)	Phân quyền người dùng

Bảng 4.1: Dữ Liệu Users

Trong bảng "Users" này:

- Trường "User_id" là một số nguyên duy nhất được sử dụng làm khóa chính để xác định mỗi người dùng.
- "Username" và "Email" là các trường để lưu trữ thông tin đăng nhập của người dùng.
- "Password" là một trường lưu trữ mật khẩu của người dùng, thường được mã hóa để bảo mật thông tin.
- " UserName ", " UserAddress " và " UserPhone " lưu trữ các thông tin cá nhân khác của người dùng.
- UserRole dùng để phân quyền người dùng khi đăng nhập
- UserPhone dùng để lưu trữ số điện thoại duy nhất của người dùng.

* Miêu tả: Bảng này dùng để lưu trữ và phân quyền cho người dùng giữa admin và user

STT	Trường	Kiểu & Độ rộng	Mô tả
1	ProductsId	int	Khóa chính duy nhất, ID của sản phẩm
2	ProductsName	nvarchar(200)	Tên của sản phẩm
3	Quatity	int	Số lượng tồn kho của sản phẩm
4	Price	DECIMAL(10,2)	Giá sản phẩm
5	ProductsDescription	nvarchar(MAX)	Mô tả chi tiết về sản phẩm
6	CategoryId	int	Khóa ngoại đến danh mục (category) mà sản phẩm thuộc về
7	ImagerUrl	varchar(MAX)	Đường dẫn hình ảnh
8	Promotion	int	Khuyến mại
9	Origin	nvarchar(100)	Nguồn gốc của sản phẩm
10	Warranty	nvarchar(20)	Thông tin bảo hành của sản phẩm

Bảng 4.2: Dữ liệu Sản phẩm

Trong bảng "Products" này:

- Trường "Product_id" là một số nguyên duy nhất được sử dụng làm khóa chính để xác định mỗi sản phẩm.
- " ProductsName " là trường để lưu trữ tên của sản phẩm.
- " ProductsDescription " là trường lưu trữ mô tả chi tiết về sản phẩm.
- "Price" là trường lưu trữ giá của sản phẩm. Định dạng DECIMAL(10,2) có nghĩa là giá có tối đa 10 chữ số và 2 chữ số sau dấu thập phân.
- " Quatity " là trường lưu trữ số lượng tồn kho của sản phẩm.
- " CategoryId " là một trường khóa ngoại tham chiếu đến bảng danh mục (category), xác định danh mục mà sản phẩm thuộc về.

- “ Promotion ” là trường lưu trữ giá khuyến mại của sản phẩm
- “ Origin ” là trường hiển thị nguồn gốc của sản phẩm
- “ Warranty ” là trường hiển thị thời gian bảo hành của sản phẩm

* Miêu tả: Bảng này dùng để hiển thị chi tiết thông tin sản phẩm như là: tên, giá, khuyến mại, nguồn gốc,...

STT	Trường	Kiểu & Độ rộng	Mô tả
1	OrderID	int	Khóa chính duy nhất, ID của chi tiết đơn hàng
2	UsersId	int	Khóa ngoại tham chiếu đến bảng đơn hàng
3	OrderDate	datetime	Ngày và giờ đặt hàng
4	UserPhone	nvarchar(14)	Số điện thoại của người đặt hàng
5	UserAddress	nvarchar(200)	Địa chỉ giao hàng

Bảng 4.3: Dữ liệu OrderProduct

Trong bảng "OrderProduct" này:

- Các trường " OrderID ", " UsersId ", " OrderDate ", " UserPhone ", " UserAddress" được thêm vào để lưu trữ thông tin liên quan đến người dùng và đơn hàng.
- "Order_id" là một trường khóa ngoại tham chiếu đến bảng đơn hàng, xác định đơn hàng mà chi tiết đơn hàng này thuộc về.
- "User_id" là một trường khóa ngoại tham chiếu đến bảng Users, xác định người dùng đặt hàng.
- "Order_date" lưu trữ ngày và giờ mà đơn hàng được đặt.
- "User_phone" và "User_address" lưu trữ thông tin liên hệ và địa chỉ giao hàng của người đặt hàng.

* Miêu tả: Bảng này dùng để hiển thị thông tin đặt hàng của người dùng bao gồm ngày đặt, địa chỉ, số điện thoại và lấy id của từng cái để phân biệt giữa các đơn hàng.

STT	Trường	Kiểu & Độ rộng	Mô tả
1	OrderDetailId	int	Khóa chính duy nhất, ID của chi tiết đơn hàng
2	OrderID	int	Khóa ngoại tham chiếu đến bảng đơn hàng
3	ProductsId	int	Khóa ngoại tham chiếu đến bảng sản phẩm
4	Quatity	int	Số lượng của sản phẩm trong chi tiết đơn hàng
5	Price	int	Giá của sản phẩm trong chi tiết đơn hàng
6	OrderStatus	nvarchar(30)	Trạng thái đơn hàng

Bảng 4.4: Dữ liệu OrderDetail

Trong bảng "OrderDetail" này:

- Trường " OrderDetailId " là một số nguyên duy nhất được sử dụng làm khóa chính để xác định mỗi chi tiết đơn hàng.
- " OrderID " là một trường khóa ngoại tham chiếu đến bảng đơn hàng, xác định đơn hàng mà chi tiết đơn hàng này thuộc về.
- " ProductsId " là một trường khóa ngoại tham chiếu đến bảng sản phẩm, xác định sản phẩm được đặt hàng trong chi tiết đơn hàng này.
- "Quantity" là trường lưu trữ số lượng của sản phẩm trong chi tiết đơn hàng.
- "Price" là trường lưu trữ giá của sản phẩm trong chi tiết đơn hàng.
- " OrderStatus " là trường hiển thị trạng thái đơn hàng.

* Miêu tả: Bảng này dùng để lấy dữ liệu từ bảng Order để tham chiếu về bảng Product để xác định xem còn hàng hay không và hiển thị ra số lượng, giá cả và khuyến mại.

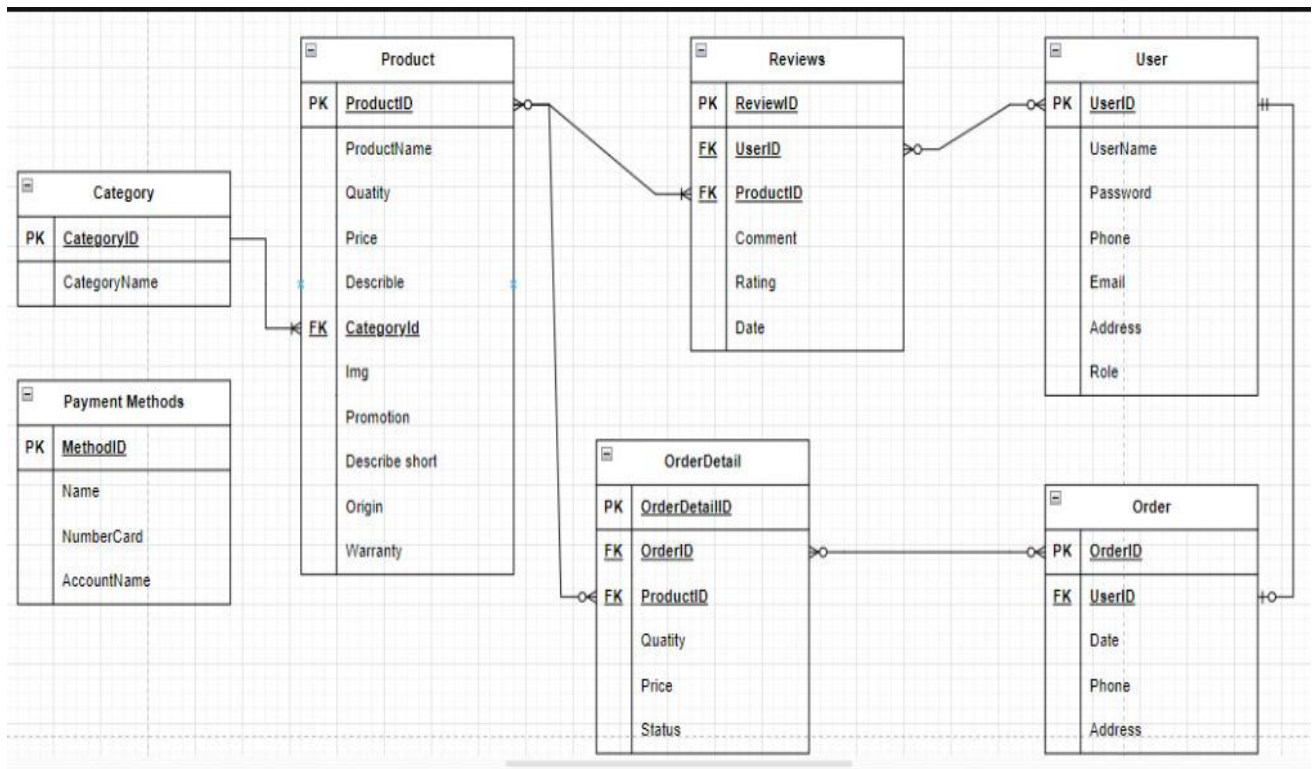
STT	Trường	Kiểu & Độ rộng	Mô tả
1	CategoryId	int	Khóa chính duy nhất, ID của danh mục
2	CategoryName	nvarchar(200)	Tên của danh mục

Bảng 4.5: Dữ liệu Category

Trong bảng "Category" này:

- Trường "CategoryId" là một số nguyên duy nhất được sử dụng làm khóa chính để xác định mỗi danh mục.
- "CategoryName" là trường để lưu trữ tên của danh mục.
- * Miêu tả: Bảng này dùng để lưu trữ các danh mục của sản phẩm.

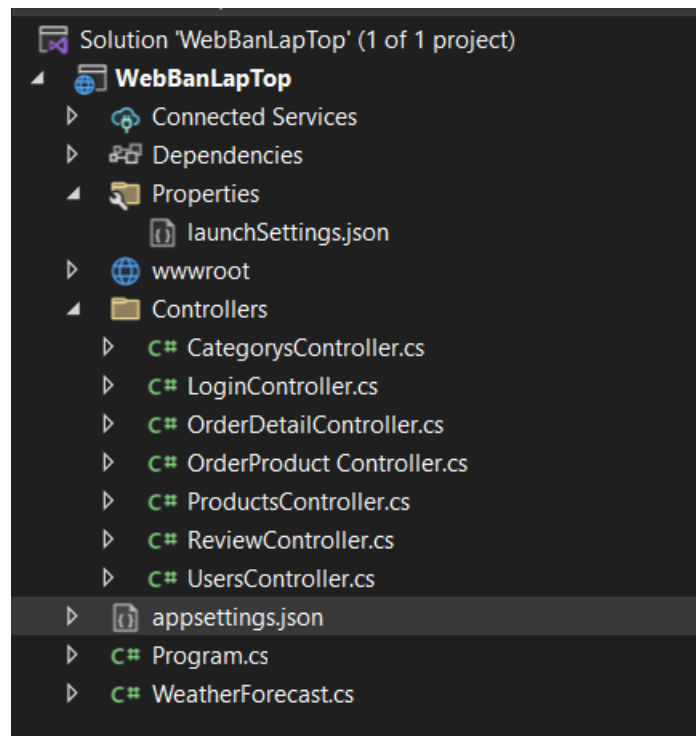
4.1.2 Mô hình quan hệ thực thể ERD



Hình 4.1: Mô hình quan hệ thực thể ERD

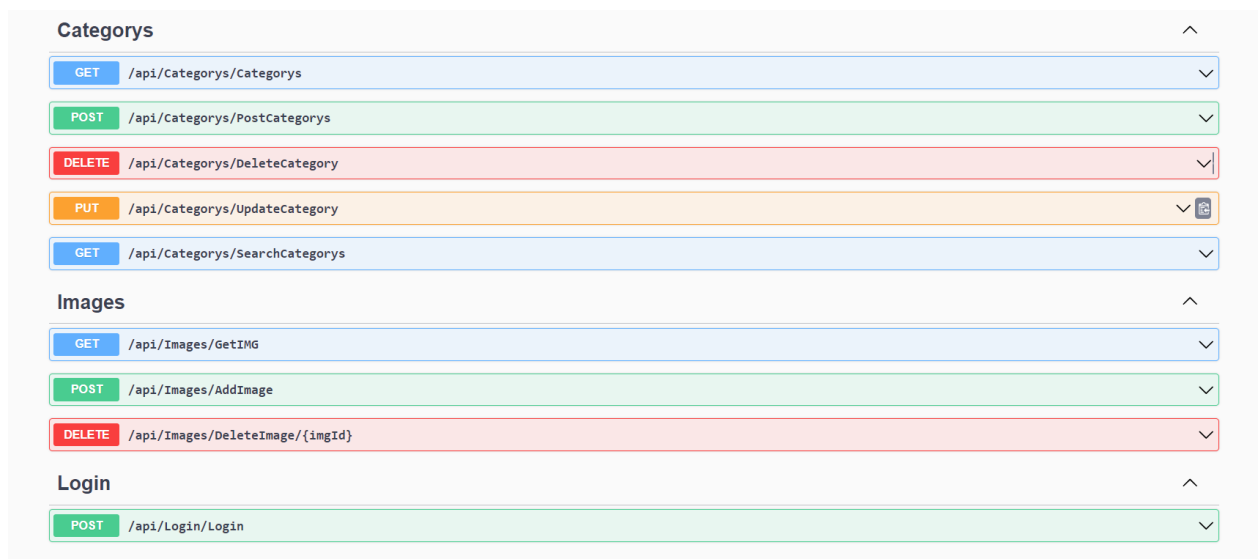
4.2 Giao diện website

a) Tổ chức mã nguồn Backend



Hình 4.2: Các thành phần mã nguồn backend

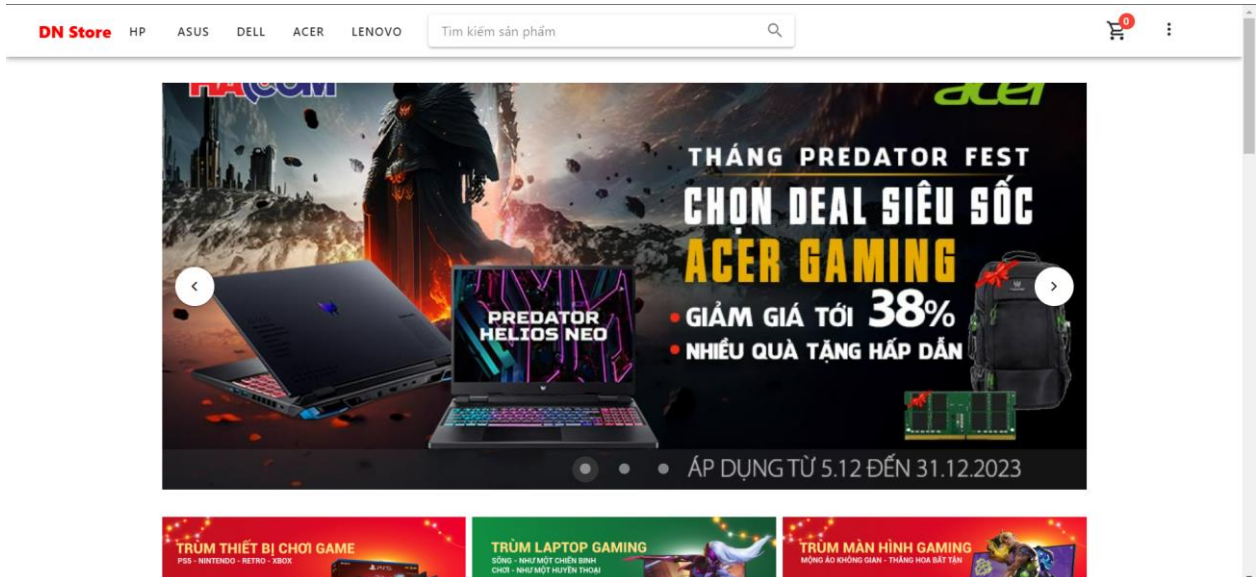
b) API Swagger



Hình 4.3: Các phương thức trong API (Swagger)

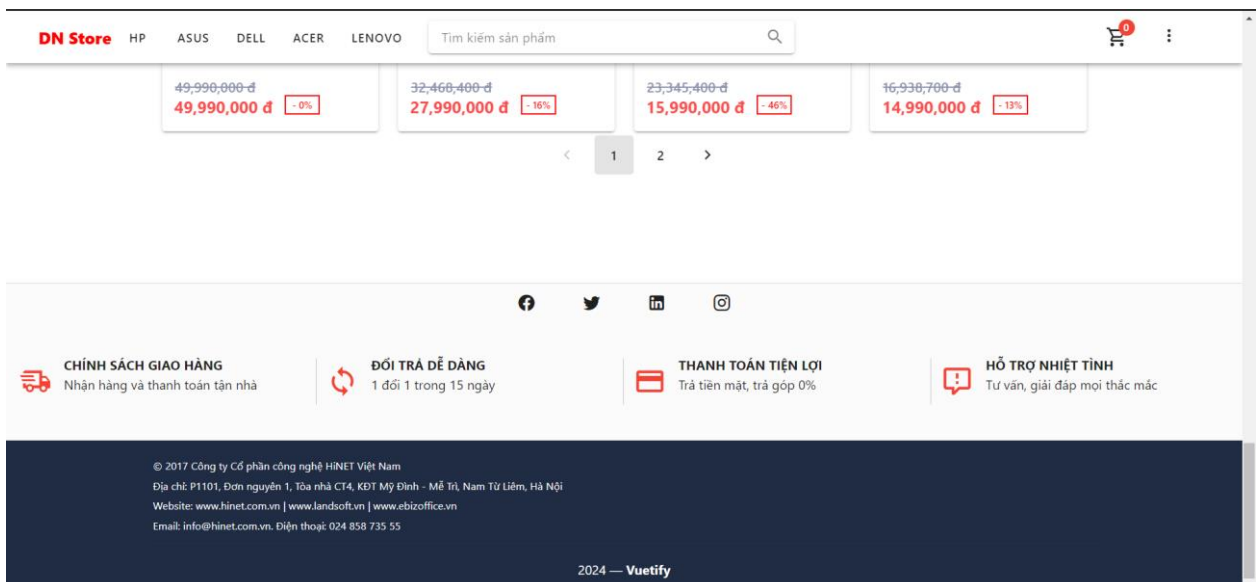
4.3 Giao diện trang web

4.3.1 Trang chủ




Hình 4.4: Giao diện trang chủ

4.3.2 Footer






Hình 4.5: Giao diện footer

4.3.3 Trang đăng nhập



Đăng nhập

 Tên đăng nhập

 Mật khẩu 

[Forgot login password?](#)

LOG IN

[Sign up now >](#)

Hình 4.6: Giao diện đăng nhập

4.3.4 Trang đăng ký



Đăng kí

ĐĂNG KÝ

[Đăng nhập ngay >](#)

Hình 4.7: Giao diện đăng ký

4.3.5 Trang thanh toán

Thông tin giao hàng

Họ tên người nhận hàng
SỐn

Số điện thoại
0123456789


Tỉnh, thành phố
Thành phố Hà Nội

Quận, huyện
Quận Long Biên

Phường, xã
Phường Long Biên

Địa chỉ nhận hàng
Số 1

Thông tin giỏ hàng

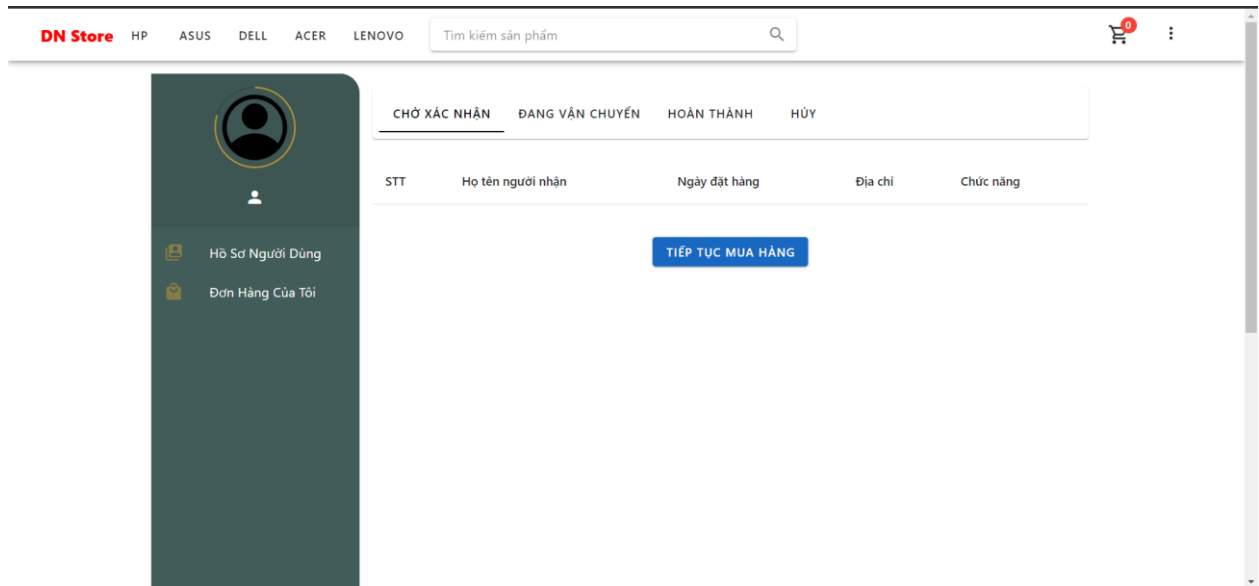
 áđ
x 1
120000 đ

Tổng chi phí **120000 đ**

XÁC NHẬN MUA HÀNG

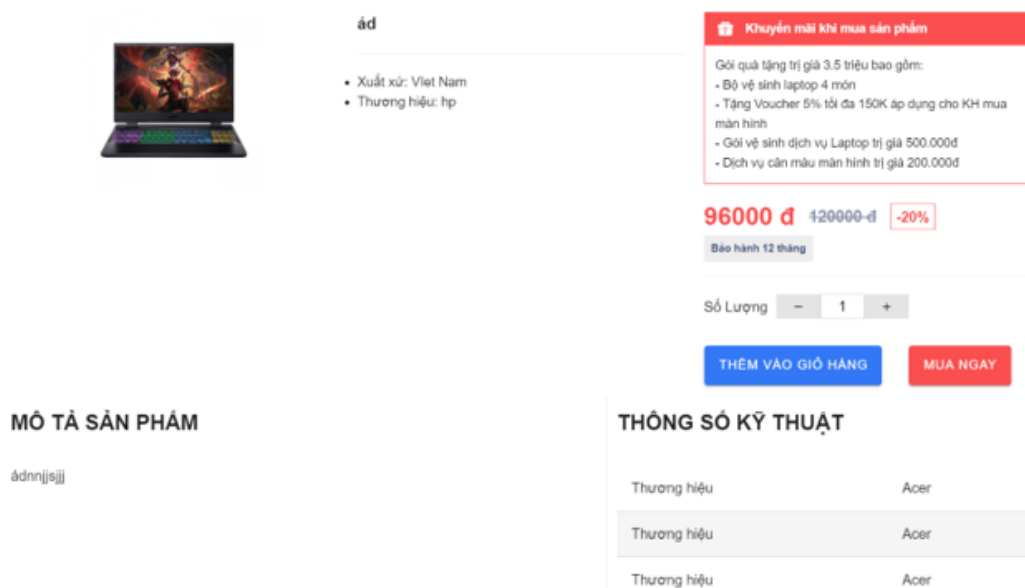
Hình 4.8: Giao diện thanh toán

4.3.6 Trang đơn hàng của tôi



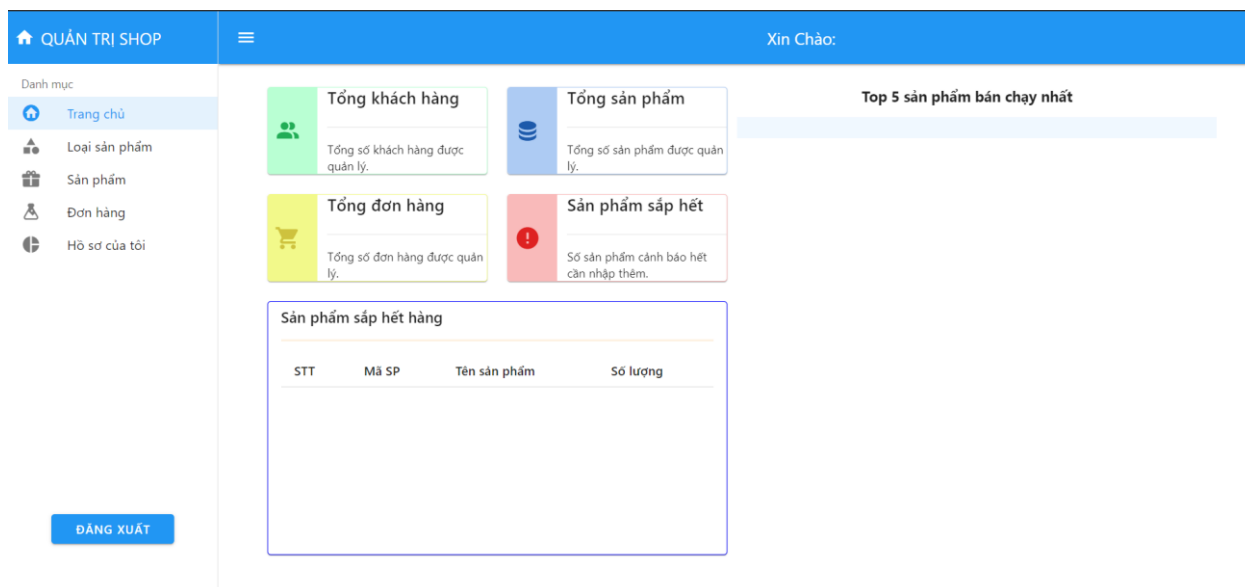
Hình 4.9: Giao diện đơn hàng của tôi

4.3.7 Trang chi tiết sản phẩm



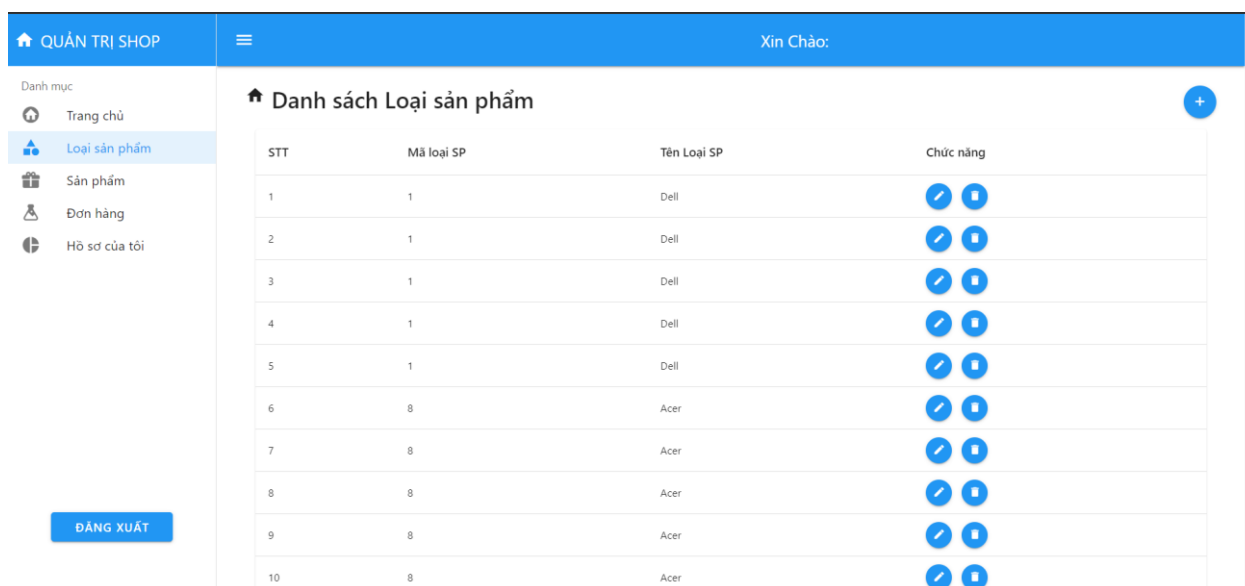
Hình 4.10: Giao diện chi tiết sản phẩm

4.3.8 Giao diện trang chủ của admin



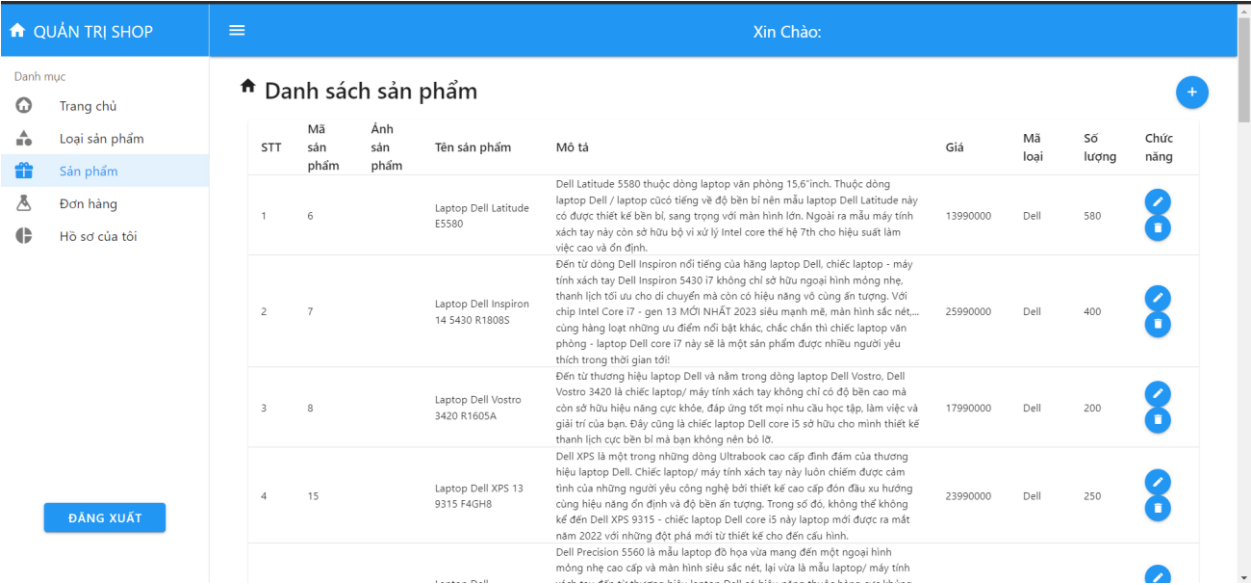
Hình 4.11: Giao diện trang chủ của admin

4.3.9 Trang loại sản phẩm



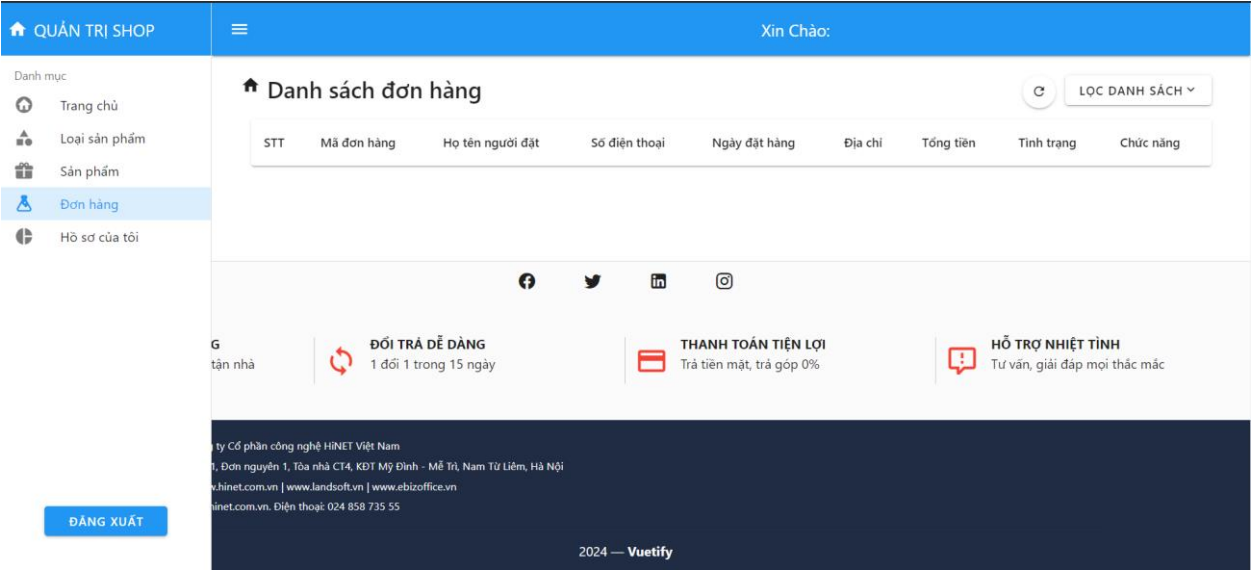
Hình 4.12: Giao diện trang loại sản phẩm

4.3.10 Trang sản phẩm



Hình 4.13: Giao diện trang sản phẩm

4.3.11 Trang đơn hàng



Hình 4.14: Giao diện trang đơn hàng

KẾT LUẬN

Xây dựng website bán các Laptop giúp cho việc mua bán online hoặc việc chọn lựa diễn ra một cách thuận tiện và nhanh chóng. Việc sử dụng công cụ máy tính hệ thống mạng máy tính giúp con người làm việc được nhanh chóng đồng thời lưu trữ dữ liệu được lâu dài. Việc xây dựng một website là không thể thiếu ở trong công ty hay là bất kỳ lĩnh vực khác. Ngày nay việc mua sắm online là vô cùng phát triển nhu cầu khách hàng sử dụng cao. Để có thể sử dụng quản lý và thống kê một cách hiệu quả thì người bán phải có một website của riêng cửa hàng. Việc xây dựng website để bán các mặt hàng Laptop vừa giúp việc mua hàng của khách hàng trở nên dễ dàng và đơn giản hơn thì người bán cũng có thể quản lý cửa hàng một cách đơn giản hơn mà vẫn có thể bao quát được hết cửa hàng. Có một website để bán hàng cũng sẽ dễ dàng hơn về mặt thời gian cho việc mua bán vì có thể giao dịch 24/24. Với khả năng phát triển cao internet thì người tiêu dùng có thể liên hệ trực tiếp được với cửa hàng để được chăm sóc khách hàng một cách nhanh nhất vào thời điểm đó.

***Kết quả đạt được:**

- Xây dựng thành công trang web giới thiệu và bán Laptop với giao diện thân thiện, dễ sử dụng và tính năng đa dạng.
- Sử dụng được ASP.Net Core và Web API cho phần backend, cùng với Vue.js và Bootstrap cho phần frontend, tạo ra trải nghiệm người dùng mượt mà và tương tác tốt.
- Kiểm thử và tối ưu hóa trang web để đảm bảo hoạt động ổn định và hiệu quả trên nhiều thiết bị và trình duyệt.

***Kết quả chưa đạt được:**

- Quản lý dữ liệu sản phẩm vẫn còn một số hạn chế, có thể cần cải thiện để tối ưu hóa quản lý và hiển thị sản phẩm.
- Hiệu suất và tương thích đa nền tảng của trang web còn có thể được cải thiện để đáp ứng tốt hơn nhu cầu của người dùng.

Định hướng phát triển:

- Xây dựng cơ sở hạ tầng, hoàn thiện website với đầy đủ nghiệp vụ hơn để có thể hoàn toàn đi vào sử dụng
- Phát triển thêm các giao diện quản lý, các chức năng như chat trực tiếp, báo cáo, thống kê số liệu, in hóa đơn.
- Tăng cường bảo mật, mã hóa dữ liệu đầu cuối
- Tối ưu hóa website, tăng hiệu suất phản hồi
- Đào tạo và hỗ trợ người dùng, bên cạnh đó cần phải duy trì và nâng cấp thường xuyên

TÀI LIỆU THAM KHẢO

- [1]. <https://developer.paypal.com/developer/applications/>
- [2]. <https://developers.facebook.com/docs/messenger-platform/overview>
- [3]. <https://www.techtarget.com/searchapparchitecture/definition/RESTful-API>
- [4]. <https://topdev.vn/blog/mo-hinh-3-lop-la-gi/>
- [5]. <https://topdev.vn/blog/tim-hieu-mo-hinh-MVC/>
- [6]. <https://netcore.vn/bai-viet/kien-thuc-entity-framework/>
- [7]. <https://vuetifyjs.com/en/>
- [8]. <https://vuetifyjs.com/en/resources/themes/>
- [9]. <https://www.tutorialspoint.com/vuejs/index.html>
- [10]. <https://www.vuemastery.com/pdf/Vue-Essentials-Cheat-Sheet.pdf>
- [11]. <https://laracasts.com/series/learn-vue-2-step-by-step>
- [12]. <https://kungfutech.edu.vn/posts/tai-lieu-giao-trinh-hoc-vuejs>
- [13]. Slide bài giảng của thầy Phạm Văn Tiệp
- [14]. Sách Vue.js – Up and Running
- [15]. Sách JavaScript & jQuery: Interactive Front-End Web Development