

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

--- 📖 ---



ĐỒ ÁN TỐT NGHIỆP

XÂY DỰNG ỨNG DỤNG BÁN SẢN PHẨM TÁI CHẾ TRÊN NỀN TẢNG ASP.NET API VÀ VUEJS

SINH VIÊN THỰC HIỆN : NGUYỄN THẾ ANH

MÃ SINH VIÊN : 1451020009

KHOA : CÔNG NGHỆ THÔNG TIN

HÀ NỘI - 2024

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



NGUYỄN THẾ ANH

**XÂY DỰNG ỨNG DỤNG BÁN SẢN PHẨM TÁI CHẾ
TRÊN NỀN TẢNG ASP.NET API VÀ VUEJS**

CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN

MÃ SỐ : 74.80.201

NGƯỜI HƯỚNG DẪN : Ths. Nguyễn Đình Tuệ

HÀ NỘI - 2024

LỜI CAM ĐOAN

Em xin cam đoan rằng toàn bộ nội dung của báo cáo thực tập tốt nghiệp này là kết quả của công sức nghiên cứu và làm việc của chính em dưới sự hướng dẫn của Thầy Nguyễn Đình Tuệ. Mọi thông tin, dữ liệu, và kết quả được trình bày trong báo cáo đều là trung thực và chưa từng được công bố trước đây ở bất kỳ nơi nào khác. Em không sao chép hoặc tham khảo ý kiến của bất kỳ công trình nào khác mà không được ghi rõ nguồn gốc.

Thứ Hai, ngày 25 tháng 05 năm 2024

Sinh viên

LỜI CẢM ƠN

Em xin bày tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Đình Tuệ - giảng viên hướng dẫn của em, đã dành thời gian và tâm huyết để hướng dẫn và hỗ trợ trong suốt quá trình làm đồ án. Sự kiên nhẫn, tận tình của thầy không chỉ giúp em hoàn thành nhiệm vụ một cách hiệu quả mà còn giúp em học hỏi được nhiều điều mới mẻ và quý báu.

Do bản thân còn những hạn chế nhất định về chuyên môn và thời gian nên sẽ khó tránh khỏi những sai sót trong quá trình thực hiện. Em mong nhận được sự góp ý của các thầy/cô trong khoa để bài báo cáo của em được hoàn thiện tốt hơn.

Một lần nữa, em xin chân thành cảm ơn!

LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, việc tận dụng Internet để thúc đẩy việc tiêu thụ sản phẩm thủ công và tái chế không chỉ giúp tăng cường năng lực cạnh tranh của doanh nghiệp mà còn góp phần tích cực vào việc bảo vệ môi trường và thúc đẩy phong cách sống bền vững. Việc xây dựng và phát triển một trang web chuyên nghiệp, hiệu quả về mặt kinh doanh và môi trường, đồng thời đáp ứng được nhu cầu đa dạng của người tiêu dùng là một thách thức đáng kể đối với các doanh nghiệp và nhà phát triển.

Trong bối cảnh đó, đề tài đồ án tốt nghiệp "Xây dựng ứng dụng bán sản phẩm tái chế trên nền tảng ASP.NET Web API và VueJS" mà em đưa ra là một nỗ lực nhằm thực hiện các mục tiêu kinh doanh và môi trường nêu trên. Báo cáo này sẽ trình bày chi tiết quá trình nghiên cứu, phân tích, thiết kế, triển khai và đánh giá hiệu quả của dự án, đồng thời đề xuất những phương hướng phát triển tiếp theo.

Em hy vọng rằng báo cáo này không chỉ là kết quả của sự cố gắng, nỗ lực và kiên trì của em mà còn là một tài liệu tham khảo quý giá cho cộng đồng doanh nghiệp, nhà nghiên cứu và những người quan tâm đến lĩnh vực thương mại điện tử và bảo vệ môi trường.

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

DANH MỤC VIẾT TẮT

STT	Ký hiệu chữ viết tắt	Chữ viết đầy đủ
1	AI	Artificial Intelligence
2	BA	Business Analyst
3	IDE	Integrated Development Environment
4	UI/UX	User Interface/User Experience

DANH MỤC HÌNH ẢNH

Biểu đồ 1 Biểu đồ tổng quát trang quản trị	15
Biểu đồ 2 Biểu đồ phân rã của use case Xem sản phẩm.....	16
Biểu đồ 3 Biểu đồ phân rã của use case Quản lý giỏ hàng	16
Biểu đồ 4 Biểu đồ phân rã của use case Quản lý đơn hàng	17
Biểu đồ 5 Biểu đồ phân rã của use case Quản lý đơn hàng	17
Biểu đồ 6 Biểu đồ phân rã của use case Quản lý giá bán	18
Biểu đồ 7 Biểu đồ phân rã của use case Quản lý sản phẩm.....	18
Biểu đồ 8 Biểu đồ phân rã của use case Quản lý nhà phân phối	19
Biểu đồ 9 Biểu đồ phân rã của use case Quản lý nhân viên	19
Biểu đồ 10 Biểu đồ phân rã của use case Quản lý khách hàng.....	20
Biểu đồ 11 Biểu đồ phân rã của use case Quản lý hóa đơn nhập	20
Hình 1:Trang design giao diện người dùng	34
Hình 2:Trang design giao diện Admin.....	35
Hình 3:Trang chủ(1) giao diện User	37
Hình 4:Trang chủ(2) giao diện User	38
Hình 5:Trang chủ (3) giao diện User	39
Hình 6:Trang giỏ hàng giao diện User.....	40
Hình 7:Trang thông tin giao hàng giao diện User.....	40
Hình 8:Trang thông báo đặt hàng thành công giao diện User	41
Hình 9:Trang đăng nhập giao diện Admin.....	42
Hình 10:Trang đăng ký giao diện Admin	43
Hình 11: Trang quản lý sản phẩm(1) giao diện Admin	44
Hình 12: Trang quản lý sản phẩm(2) giao diện Admin	44
Hình 13: Trang quản lý sản phẩm(3) giao diện Admin	45

DANH MỤC BẢNG BIỂU

STT	Mô Tả
Bảng 3.1	Bảng sản phẩm
Bảng 3.2	Bảng khách hàng
Bảng 3.3	Bảng loại sản phẩm
Bảng 3.4	Bảng hóa đơn
Bảng 3.5	Bảng trạng thái
Bảng 3.6	Bảng Account
Bảng 3.7	Bảng chi tiết hóa đơn
Bảng 3.8	Bảng giỏ hàng
Bảng 3.9	Bảng kho

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU KHÁI QUÁT VỀ ĐỀ TÀI	1
1.1 Lý do chọn đề tài.....	1
1.2 Mục đích nghiên cứu:	1
1.3 Phạm vi nghiên cứu:	1
1.4 Phương pháp nghiên cứu:	2
CHƯƠNG 2: TỔNG QUAN VỀ ASP.NET CORE WEB API VÀ VUEJS	5
2.1. Công nghệ sử dụng.....	5
2.1.1. ASP.NET Core	5
2.1.2. Vue.js.....	6
2.1.3. UML - Unified Modeling Language	9
2.2. Công cụ sử dụng.....	10
2.2.1. Visual Paradigm	10
2.2.2. Figma.....	11
2.2.3. Visual Studio và VS Code	12
CHƯƠNG 3: XÂY DỰNG VÀ THIẾT KẾ HỆ THỐNG.....	13
3.1. Mô tả bài toán.....	13
3.2. Thiết kế hệ thống.....	15
3.3. Thiết kế cơ sở dữ liệu	28
3.3.1. Bảng SanPham	28
3.3.2. Bảng KháchHang	28
3.3.3. Bảng LoaiSanPham.....	29
3.3.4. Bảng HoaDon.....	29
3.3.5. Bảng TrangThai.....	30

3.3.6. Bảng Account.....	31
3.3.7. Bảng Chitiethoadon.....	31
3.3.8. Bảng GioHang.....	32
3.3.9. Bảng Kho.....	33
3.4. UI/UX Design	34
3.5. Giao diện và chức năng.....	36
3.5.1. Trang giao diện người dùng	36
3.5.2. Trang đăng nhập, đăng ký.....	41
3.5.3. Trang giao diện Admin	43
CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN	46
4.1. Kết quả đạt được	46
4.2. Hạn chế.....	46
4.3. Hướng phát triển	47
KẾT LUẬN	49
TÀI LIỆU THAM KHẢO.....	50

CHƯƠNG 1: GIỚI THIỆU KHÁI QUÁT VỀ ĐỀ TÀI

1.1 Lý do chọn đề tài

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, việc tận dụng Internet để thúc đẩy việc tiêu thụ sản phẩm thủ công và tái chế không chỉ giúp tăng cường năng lực cạnh tranh của doanh nghiệp mà còn góp phần tích cực vào việc bảo vệ môi trường và thúc đẩy phong cách sống bền vững. Việc xây dựng và phát triển một trang web chuyên nghiệp, hiệu quả về mặt kinh doanh và môi trường, đồng thời đáp ứng được nhu cầu đa dạng của người tiêu dùng là một thách thức đáng kể đối với các doanh nghiệp và nhà phát triển.

Trong bối cảnh đó, đề tài đồ án tốt nghiệp " Xây dựng ứng dụng bán sản phẩm tái chế trên nền tảng ASP.NET CORE Web API và VueJS" mà tôi nghiên cứu sau đây sẽ là một nỗ lực nhằm thực hiện các mục tiêu kinh doanh và môi trường nêu trên. Báo cáo này sẽ trình bày chi tiết quá trình nghiên cứu, phân tích, thiết kế, triển khai và đánh giá hiệu quả của dự án, đồng thời đề xuất những phương hướng phát triển tiếp theo.

Tôi hy vọng rằng báo cáo này không chỉ là kết quả của sự cố gắng, nỗ lực và kiến thức của tôi mà còn là một tài liệu tham khảo quý giá cho cộng đồng doanh nghiệp, nhà nghiên cứu và những người quan tâm đến lĩnh vực thương mại điện tử và bảo vệ môi trường.

1.2 Mục đích nghiên cứu:

Đề tài “Xây dựng ứng dụng bán sản phẩm tái chế trên nền tảng ASP.NET CORE Web API và VueJS” tập trung phát triển một ứng dụng web bán sản phẩm tái chế dựa trên nền tảng ASP.NET CORE Web API và VueJS để cung cấp cho người dùng một trải nghiệm mua sắm trực tuyến tốt nhất, tạo ra một không gian cho phép người dùng có thể tìm kiếm những sản phẩm mình cần và là một môi trường thú vị cho cộng đồng người yêu thích sản phẩm tái chế và bảo vệ môi trường .

1.3 Phạm vi nghiên cứu:

Phạm vi nghiên cứu của đề tài này sẽ bao gồm những việc như xác định các tính năng cụ thể và chức năng mà ứng dụng của tôi sẽ cung cấp cho người dùng. Với việc sử dụng backend là ASP.NET CORE Web API và frontend là VueJS, cũng như quản lý dữ liệu và tương tác người dùng. Tôi sẽ tập trung vào nhiều thể loại sản phẩm sẽ hỗ trợ trên trang web này ví dụ : Các loại sản phẩm làm từ giấy, chai nhựa,...

Xác định đối tượng người dùng của ứng dụng : độ tuổi, giới tính, nghề nghiệp, sở thích ,...

1.4 Phương pháp nghiên cứu:

Để hoàn thành đề tài này một cách hoàn thiện, tốt nhất thì phải có phương pháp nghiên cứu thật rõ ràng, cụ thể:

- Nghiên cứu thư mục: Tìm hiểu về các công nghệ ASP.NET Core Web API và VueJS, cũng như các tiêu chuẩn và quy trình phát triển ứng dụng web.
- Một số công cụ hỗ trợ trong quá trình thực hiện: Visual Paradigm, SQL Server, Figma,... Các công cụ này có thể hỗ trợ trong quá trình phát triển và kiểm thử ứng dụng bán sản phẩm tái chế của tôi, từ việc thiết kế giao diện đến phân tích dữ liệu và kiểm thử API.
- Phát triển nguyên mẫu (Prototype Development): Xây dựng một phiên bản nguyên mẫu của ứng dụng để thử nghiệm các tính năng cụ thể và thu thập phản hồi từ người dùng. Phiên bản này thường không hoàn chỉnh nhưng đủ để thử nghiệm các tính năng cơ bản và thu thập thông tin phản hồi từ người dùng để cải thiện và điều chỉnh trước khi triển khai phiên bản chính thức hoặc sản phẩm cuối cùng.
- Nghiên cứu người dùng (User Research): Tiến hành phỏng vấn hoặc không có thể điều tra, tìm hiểu qua nhiều nguồn mạng về người dùng để hiểu rõ nhu cầu và mong muốn của họ đối với ứng dụng bán sản phẩm này.
- Phân tích dữ liệu: Sử dụng các công cụ phân tích dữ liệu để hiểu về cách mà người dùng tương tác với ứng dụng và đánh giá hiệu suất của nó.

Bằng cách này, tôi có thể xác định một phạm vi nghiên cứu rõ ràng và thiết lập các phương pháp phù hợp để tiến hành nghiên cứu của mình một cách hiệu quả.

1.5 Bố cục luận văn

Chương 1: Khái quát đề tài

1.1 Lý do chọn đề tài

1.2 Mục tiêu nghiên cứu

1.3 Phạm vi nghiên cứu

1.4 Phương pháp nghiên cứu

Chương 2: Tổng quan về ASP.NET CORE WEB API và VueJS

2.1 Tổng quan về ASP.NET CORE WEB API

2.1.1. Khái niệm

2.1.2. Thành phần của ASP.NET CORE WEB API

2.1.3. Các loại ASP.NET CORE WEB API phổ biến

2.1.4. Ưu điểm và thách thức khi sử dụng ASP.NET CORE WEB API

2.2 Tổng quan về VueJS

2.2.1. Khái niệm

2.2.2. Ưu và nhược điểm của Vue.js

Chương 3: đặc tả yêu cầu và thiết kế hệ thống

3.1. Đặc tả yêu cầu

3.1.1. Yêu cầu chức năng

3.1.2. Các yêu cầu phi chức năng

3.2. Thiết kế hệ thống

3.2.1. Kiến trúc hệ thống

3.2.2. Cơ sở dữ liệu

3.2.2.1. Xác định thực thể quan hệ

3.2.2.2. Biểu đồ liên kết ERD

3.2.2.3. Biểu đồ Diagram Database

3.3. Giao diện người dùng và chức năng

3.3.1 Một số giao diện chính

3.3.2 Một vài chức năng nổi bật

Chương 4: Kết quả đạt được và hướng phát triển

4.1 Kết quả đạt được

4.2 Hạn chế

4.3 Hướng phát triển

Kết Luận

- Kết quả đạt được:
 - ✓ Cở sở lý thuyết
 - ✓ Cơ sở thực hành
- Kỹ năng đạt được:
 - ***Đánh giá***
 - Những điểm phù hợp:
 - Những điểm chưa phù hợp:
 - Đề xuất giải pháp:

Tài liệu tham khảo

CHƯƠNG 2: TỔNG QUAN VỀ ASP.NET CORE WEB API VÀ VUEJS

2.1. Công nghệ sử dụng

2.1.1. ASP.NET Core

ASP.NET Core là một framework phát triển ứng dụng web mã nguồn mở, được phát triển bởi Microsoft. Nó là một phiên bản tiếp theo của ASP.NET, được thiết kế để chạy trên nền tảng đa nền tảng, bao gồm Windows, macOS và Linux. ASP.NET Core cung cấp một cách tiếp cận hiện đại và linh hoạt để xây dựng các ứng dụng web hiệu suất cao, đa nhiệm vụ và có khả năng mở rộng.

Đặc điểm chính của ASP.NET Core bao gồm:

- Đa nền tảng (Cross-platform): ASP.NET Core có thể chạy trên nhiều hệ điều hành khác nhau, bao gồm Windows, macOS và Linux. Điều này mang lại sự linh hoạt cho việc triển khai và quản lý ứng dụng trên các môi trường khác nhau mà không bị ràng buộc bởi một nền tảng cụ thể nào.
- Hiệu suất cao: ASP.NET Core được tối ưu hóa để cung cấp hiệu suất tốt nhất, đặc biệt là trong các ứng dụng web yêu cầu xử lý lượng truy cập lớn và đồng thời từ nhiều nguồn khác nhau. Nó cung cấp các công cụ và kỹ thuật tối ưu hóa để tăng cường khả năng đáp ứng và tốc độ xử lý của ứng dụng.
- Modular và linh hoạt: ASP.NET Core được thiết kế theo mô hình modular, cho phép người phát triển chọn lựa các thành phần cần thiết cho ứng dụng của họ mà không cần phải đưa vào tất cả các thành phần của framework. Điều này giúp giảm kích thước của ứng dụng và tối ưu hóa việc sử dụng tài nguyên.
- Tích hợp tốt với các công nghệ khác: ASP.NET Core tích hợp tốt với các công nghệ và công cụ khác trong hệ sinh thái phát triển phần mềm của Microsoft như Entity Framework Core (ORM), SignalR (thời gian thực), và Identity (quản lý xác thực và phân quyền).

- Hỗ trợ phát triển ứng dụng web hiện đại: ASP.NET Core hỗ trợ các mô hình phát triển hiện đại như giao diện người dùng đơn trang (Single Page Applications - SPA), Web API cho việc xây dựng các dịch vụ web RESTful, và Razor Pages cho việc xây dựng các trang web truyền thống.
- Tích hợp với công cụ phát triển: ASP.NET Core tích hợp tốt với Visual Studio và Visual Studio Code, hai công cụ phát triển phổ biến của Microsoft, giúp người phát triển dễ dàng tạo, debug và triển khai ứng dụng.

ASP.NET Core đã trở thành một trong những lựa chọn phổ biến cho việc phát triển ứng dụng web hiện đại và đa nền tảng, nhờ vào tính linh hoạt, hiệu suất và khả năng tích hợp tốt với các công nghệ và công cụ khác.

2.1.2. Vue.js

2.1.2.1. Giới thiệu chung

Vue.js là một framework JavaScript được sử dụng để xây dựng giao diện người dùng và các ứng dụng web tương tác. Nó được thiết kế để tối ưu hóa việc xây dựng các thành phần giao diện và quản lý trạng thái của ứng dụng một cách dễ dàng và hiệu quả.

Vue.js cung cấp một cách tiếp cận linh hoạt và dễ hiểu để phát triển ứng dụng web. Với cú pháp gần gũi và tài liệu phong phú, người phát triển có thể nhanh chóng bắt đầu và xây dựng các ứng dụng web phức tạp. Vue.js cũng tích hợp tốt với các thư viện và framework khác, cho phép người phát triển lựa chọn các công cụ phù hợp nhất với dự án của mình.

Một trong những điểm mạnh của Vue.js là khả năng linh hoạt trong việc tích hợp vào các dự án có sẵn mà không cần phải thay đổi cấu trúc hoặc mã nguồn hiện tại của dự án. Vue.js cũng có cộng đồng lớn và tích hợp nhiều công cụ hỗ trợ như Vue Router (định tuyến), Vuex (quản lý trạng thái) và Vue CLI (công cụ dòng lệnh) giúp việc phát triển ứng dụng trở nên thuận lợi hơn.

2.1.2.2. Vuetify

Vuetify là một thư viện UI (User Interface) cho Vue.js, được xây dựng dựa trên nguyên tắc thiết kế Material Design của Google. Nó cung cấp một bộ công cụ UI đồ sộ và đa dạng, giúp người phát triển tạo ra các giao diện người dùng (UI) đẹp mắt và dễ sử dụng cho ứng dụng web và di động. Với Vuetify, việc xây dựng các thành phần UI như các nút, menu, bảng dữ liệu, biểu đồ và nhiều thành phần khác trở nên đơn giản hơn và nhanh chóng hơn.

Một trong những điểm mạnh của Vuetify là tích hợp sẵn các thành phần UI theo chuẩn Material Design, giúp người phát triển tạo ra các ứng dụng có giao diện đồng nhất và chuyên nghiệp. Ngoài ra, Vuetify cũng hỗ trợ tích hợp linh hoạt với các thư viện và framework khác của Vue.js, như Vuex cho quản lý trạng thái và Vue Router cho định tuyến, giúp tăng cường khả năng mở rộng và tái sử dụng trong quá trình phát triển ứng dụng.

Vuetify cũng được đánh giá cao về tài liệu hướng dẫn và cộng đồng hỗ trợ sôi nổi, điều này giúp người phát triển dễ dàng tìm kiếm thông tin, hướng dẫn và hỗ trợ khi gặp vấn đề trong quá trình sử dụng. Đồng thời, Vuetify cũng luôn cập nhật và phát triển mới để đáp ứng các yêu cầu và xu hướng mới trong việc xây dựng giao diện người dùng hiện đại và tương tác.

2.1.2.3. Vuex

Vuex là một thư viện quản lý trạng thái cho ứng dụng Vue.js. Nó giúp quản lý trạng thái của ứng dụng một cách hiệu quả và dễ dàng, đặc biệt là khi ứng dụng có quy mô lớn và phức tạp. Vuex cung cấp một cách tiếp cận chuẩn để lưu trữ và quản lý trạng thái của ứng dụng, giúp việc truy cập và cập nhật dữ liệu trở nên thuận tiện và đồng bộ.

Vuex hoạt động dựa trên mô hình kiến trúc Flux, với các thành phần chính như state (trạng thái), mutations (điều chỉnh), actions (hành động), và getters (bộ lọc).

- + State: Là nơi lưu trữ toàn bộ trạng thái của ứng dụng.

- + **Mutations:** Là các hàm được sử dụng để thay đổi trạng thái trong Vuex. Chúng chỉ có thể được thực hiện đồng bộ.
- + **Actions:** Là các hàm được sử dụng để thực hiện các tác vụ bất đồng bộ, như gửi yêu cầu HTTP hoặc thực hiện các tác vụ phức tạp trước khi thay đổi trạng thái.
- + **Getters:** Là các hàm được sử dụng để lấy dữ liệu từ trạng thái trong Vuex một cách động.

Vuex giúp tránh tình trạng rối loạn trạng thái (state) và tạo ra một luồng dữ liệu đơn điệu trong ứng dụng Vue.js. Điều này giúp việc theo dõi và debug ứng dụng trở nên dễ dàng hơn, đồng thời cũng tạo điều kiện thuận lợi để quản lý trạng thái toàn cục của ứng dụng một cách hiệu quả và có tổ chức.

2.1.2.4. Axios

Axios là một thư viện JavaScript dùng để thực hiện các yêu cầu HTTP từ phía client. Nó được thiết kế để làm việc trên cả môi trường Node.js và trình duyệt, cung cấp một cách dễ dàng để thực hiện các hoạt động như gọi API, gửi và nhận dữ liệu từ server.

Dưới đây là một số điểm nổi bật của Axios:

- **Tính Năng Chính của Axios Promise-based:** Axios sử dụng Promises, giúp việc quản lý các yêu cầu bất đồng bộ trở nên dễ dàng hơn. Điều này giúp bạn sử dụng cú pháp `async/await` để làm cho mã dễ đọc và duy trì.
- **Interceptors:** Axios cung cấp các Interceptors cho phép bạn can thiệp vào các yêu cầu hoặc phản hồi trước khi chúng được xử lý. Điều này rất hữu ích để thêm các headers xác thực, xử lý lỗi toàn cục, hoặc thay đổi dữ liệu trước khi gửi.
- **Tự động chuyển đổi dữ liệu:** Axios tự động chuyển đổi dữ liệu JSON khi gửi và nhận, giúp giảm bớt công việc xử lý dữ liệu thủ công.

- Hỗ trợ cho các yêu cầu đồng bộ: Bạn có thể gửi các yêu cầu HTTP đồng bộ như GET, POST, PUT, DELETE, v.v. Hỗ trợ cho timeouts: Bạn có thể dễ dàng đặt thời gian chờ cho các yêu cầu HTTP.
- Hỗ trợ XSRF: Axios có hỗ trợ tích hợp cho chống tấn công Cross-Site Request Forgery (XSRF).

2.1.3. UML - Unified Modeling Language

UML (Unified Modeling Language) là một ngôn ngữ đồ họa được sử dụng để mô hình hóa và thiết kế các hệ thống phần mềm. UML cung cấp một cách tiêu chuẩn để biểu diễn cấu trúc và hành vi của một hệ thống phần mềm, giúp các nhà phát triển và nhà quản lý dễ dàng hiểu và trao đổi ý tưởng với nhau.

UML bao gồm nhiều loại biểu đồ, mỗi loại biểu đồ được sử dụng để mô hình hóa một khía cạnh cụ thể của hệ thống. Các loại biểu đồ phổ biến trong UML bao gồm:

- Biểu đồ lớp (Class Diagram): Biểu diễn các lớp, các thuộc tính và phương thức của lớp, cũng như các mối quan hệ giữa các lớp.
- Biểu đồ use case (Use Case Diagram): Biểu diễn các chức năng của hệ thống và cách các thành phần tương tác với nhau trong các kịch bản sử dụng.
- Biểu đồ sequence (Sequence Diagram): Biểu diễn các thông điệp được trao đổi giữa các đối tượng trong các tình huống cụ thể.
- Biểu đồ hoạt động (Activity Diagram): Biểu diễn các hoạt động và luồng điều khiển trong hệ thống.
- Biểu đồ trạng thái (State Diagram): Biểu diễn các trạng thái của một đối tượng và các sự kiện dẫn đến việc chuyển đổi giữa các trạng thái đó.

UML là một công cụ quan trọng trong quy trình phát triển phần mềm, giúp cải thiện hiểu biết và truyền đạt thông tin giữa các thành viên trong nhóm phát triển.

2.2. Công cụ sử dụng

2.2.1. Visual Paradigm

Visual Paradigm là một công cụ mô hình hóa và thiết kế phần mềm mạnh mẽ và linh hoạt, hỗ trợ nhiều loại biểu đồ UML và các phương pháp phát triển phần mềm khác nhau. Được phát triển bởi công ty Visual Paradigm International, Visual Paradigm cung cấp một loạt các tính năng và công cụ cho việc phát triển phần mềm từ khâu phân tích yêu cầu, thiết kế, triển khai cho đến kiểm thử và quản lý dự án.

Một số tính năng chính của Visual Paradigm bao gồm:

- Hỗ trợ nhiều loại biểu đồ: Visual Paradigm hỗ trợ các loại biểu đồ UML như biểu đồ lớp, biểu đồ use case, biểu đồ hoạt động, biểu đồ sequence, biểu đồ trạng thái, cũng như các loại biểu đồ khác như biểu đồ ERD (Entity-Relationship Diagram), biểu đồ công cụ (Tool Overview Diagram), và biểu đồ BPMN (Business Process Model and Notation).

- Mô hình hóa và thiết kế đa dạng: Visual Paradigm cho phép mô hình hóa và thiết kế phần mềm theo nhiều phương pháp và quy trình khác nhau, bao gồm Agile, Waterfall, và SCRUM.

- Tích hợp công cụ quản lý dự án: Visual Paradigm tích hợp các công cụ quản lý dự án như Gantt Chart, Resource Management và Project Dashboard để hỗ trợ quản lý dự án phần mềm hiệu quả.

- Hỗ trợ đa nền tảng: Visual Paradigm hỗ trợ nhiều nền tảng như Windows, macOS và Linux, cho phép người dùng làm việc trên môi trường ưa thích của họ.

- Tính tương tác cao: Visual Paradigm cung cấp các công cụ để tạo ra các mô hình tương tác động, giúp người dùng hiểu rõ hơn về cách hệ thống hoạt động trong thực tế.

Visual Paradigm là một công cụ mô hình hóa và thiết kế phần mềm đa năng và mạnh mẽ, phù hợp cho các nhà phát triển phần mềm và nhóm làm việc trong môi trường phát triển phần mềm chuyên nghiệp.

2.2.2. Figma

Figma là một công cụ thiết kế UI/UX trực tuyến hàng đầu, mang đến cho người dùng khả năng tạo ra giao diện trang web, ứng dụng di động và nhiều sản phẩm khác một cách dễ dàng. Với Figma, người dùng được trang bị một loạt các công cụ vẽ mạnh mẽ, các thành phần tái sử dụng và phần tử giao diện sẵn có, giúp người dùng thể hiện ý tưởng và tạo ra các thiết kế chất lượng cao một cách nhanh chóng giúp tăng hiệu suất và tiết kiệm thời gian trong quá trình thiết kế.

Ưu điểm:

- Làm việc đa nền tảng: Figma cho phép người dùng truy cập và làm việc trên Windows, macOS và trình duyệt web, mang lại sự linh hoạt và tiện lợi cho người dùng.

- Hợp tác dễ dàng: Tính năng hợp tác của Figma cho phép người dùng chia sẻ dự án và nhận phản hồi trực tiếp từ đồng nghiệp, tạo điều kiện cho việc làm việc nhóm hiệu quả.

- Cập nhật và đồng bộ tự động: Figma cung cấp cập nhật và đồng bộ tự động, giúp người dùng luôn làm việc trên phiên bản mới nhất của dự án và tiết kiệm thời gian trong việc đồng bộ hóa.

Nhược điểm:

- Xử lý dự án lớn và phức tạp không hiệu quả: Hiệu suất của Figma có thể giảm đi khi làm việc trên các dự án lớn và phức tạp, gây trễ trong việc phản hồi và xử lý.

Đáp ứng chậm hơn so với các công cụ desktop: Một số người dùng có thể gặp phản ứng chậm hơn so với các công cụ desktop khác khi làm việc trên Figma, phụ thuộc vào tốc độ kết nối internet và phần cứng máy tính.

- Giới hạn của phiên bản miễn phí: Phiên bản miễn phí của Figma có giới hạn về số lượng dự án, lưu trữ và tính năng, người dùng cần nâng cấp lên phiên bản trả phí để sử dụng đầy đủ tính năng của Figma.

- Không thể làm việc ngoại tuyến: Do tính chất đám mây, Figma không cho phép làm việc ngoại tuyến và yêu cầu kết nối internet để truy cập và làm việc trên nó.

2.2.3. Visual Studio và VS Code

Visual Studio là một môi trường phát triển tích hợp (IDE) được phát triển bởi Microsoft. Visual Studio cung cấp một loạt các công cụ và tính năng phong phú để phát triển ứng dụng cho nhiều nền tảng khác nhau, bao gồm Windows, macOS, iOS, Android, và web. Visual Studio hỗ trợ nhiều ngôn ngữ lập trình như C#, C++, JavaScript, Python và nhiều ngôn ngữ khác.

Visual Studio có một giao diện đồ họa mạnh mẽ và cung cấp các công cụ để quản lý mã nguồn, xây dựng, debug, và triển khai ứng dụng. Nó cũng tích hợp các công cụ hỗ trợ cho việc làm việc theo nhóm như Git và Azure DevOps.

Visual Studio Code (hay VS Code) là một trình soạn thảo mã nguồn mở, nhẹ và mạnh mẽ được phát triển bởi Microsoft. Visual Studio Code được thiết kế để làm việc với nhiều ngôn ngữ lập trình và cung cấp một loạt các tính năng như highlight syntax, IntelliSense (auto-completion), debugging, và các extension để mở rộng khả năng của trình soạn thảo.

Mặc dù Visual Studio Code không phải là một IDE đầy đủ như Visual Studio, nhưng nó rất linh hoạt và hỗ trợ nhiều ngôn ngữ lập trình và framework khác nhau. Visual Studio Code cũng có cộng đồng lớn và sôi động, với hàng ngàn extension để giúp tăng cường tính năng và sức mạnh của trình soạn thảo này.

CHƯƠNG 3: XÂY DỰNG VÀ THIẾT KẾ HỆ THỐNG

3.1. Mô tả bài toán

Chương trình phần mềm quản lý cửa hàng bán sản phẩm tái chế cho phép quản lý và theo dõi số lượng sản phẩm còn lại trong kho, trạng thái sử dụng, giá bán và nhà cung cấp. Ngoài ra, phần mềm còn cho phép đặt hàng mới từ nhà cung cấp khi số lượng sản phẩm còn ít, giúp đảm bảo kho luôn đầy đủ hàng hóa để phục vụ khách hàng.

Chương trình phần mềm quản lý cửa hàng bán sản phẩm tái chế có quản lý và theo dõi hóa đơn, kiểm tra tình trạng đơn hàng, và quản lý đơn hàng tồn kho. Điều này giúp người quản lý đưa ra các quyết định kinh doanh phù hợp, cải thiện hiệu quả kinh doanh và phục vụ khách hàng tốt hơn. Phần mềm cũng có thể gửi thông báo nhắc nhở về các đơn hàng chưa hoàn tất hoặc cần được xử lý gấp.

Chương trình phần mềm quản lý cửa hàng bán sản phẩm tái chế sẽ có quản lý và lưu trữ thông tin khách hàng, bao gồm tên, địa chỉ, số điện thoại, email, và hình ảnh khách hàng. Điều này nhằm phục vụ tốt hơn cho các chương trình khuyến mãi và chăm sóc khách hàng. Phần mềm có thể gửi email marketing tự động, thông báo các chương trình khuyến mãi và sự kiện đặc biệt đến khách hàng.

Chương trình phần mềm quản lý cửa hàng bán sản phẩm tái chế cho phép quản lý lấy dữ liệu để phân tích số lượng sản phẩm bán ra mỗi ngày, mỗi tuần và mỗi tháng. Từ đó, giúp đưa ra các quyết định kinh doanh phù hợp và hiệu quả. Phần mềm cũng có thể tạo ra các báo cáo chi tiết về xu hướng mua sắm của khách hàng, giúp tối ưu hóa chiến lược kinh doanh.

Chương trình phần mềm quản lý cửa hàng bán sản phẩm tái chế cho phép quản lý tài khoản đăng nhập của nhân viên và kiểm soát các hành động trên phần mềm để đảm bảo an toàn và bảo mật thông tin của cửa hàng. Phần mềm có thể theo dõi hoạt động của từng nhân viên, từ đó đánh giá hiệu suất làm việc và đưa ra các biện pháp khuyến khích phù hợp.

Chương trình phần mềm quản lý cửa hàng bán sản phẩm tái chế còn cho phép tạo và quản lý các chương trình khuyến mãi, ưu đãi đặc biệt. Phần mềm có thể theo dõi hiệu quả của các chương trình này, từ đó điều chỉnh để tối ưu hóa lợi ích kinh doanh.

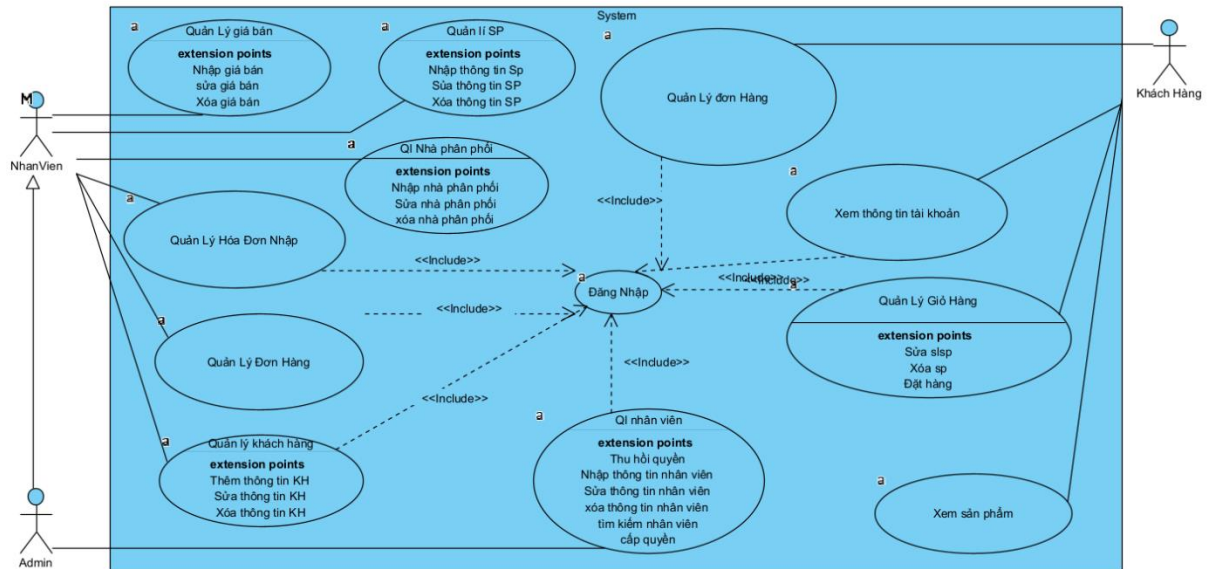
Chương trình phần mềm quản lý cửa hàng bán sản phẩm tái chế hỗ trợ quản lý các dịch vụ hậu mãi và bảo hành sản phẩm. Thông tin về bảo hành, các yêu cầu sửa chữa hoặc đổi trả sản phẩm được lưu trữ và theo dõi một cách hiệu quả, giúp nâng cao sự hài lòng của khách hàng.

Chương trình phần mềm quản lý cửa hàng bán sản phẩm tái chế có khả năng tích hợp với các hệ thống khác như hệ thống quản lý kho, hệ thống kế toán, và các công thanh toán trực tuyến. Điều này giúp tạo ra một quy trình kinh doanh liền mạch và hiệu quả hơn.

3.2. Thiết kế hệ thống

Các yêu cầu chức năng Trang quản trị

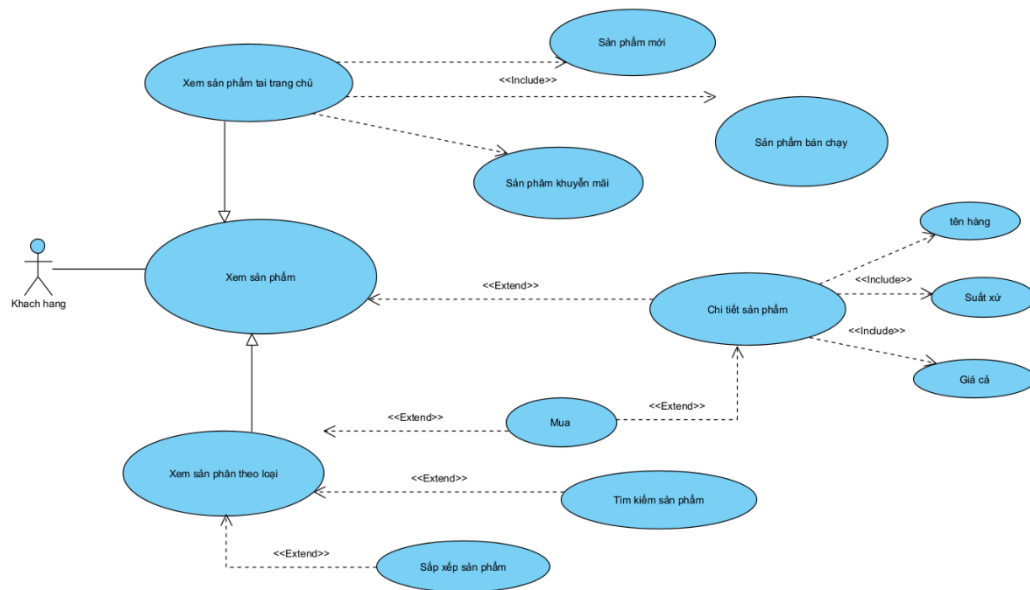
- Biểu đồ tổng quát



Biểu đồ 1 Biểu đồ tổng quát trang quản trị

Biểu đồ tổng quát trang quản trị gồm các chức năng chính như sau: Các chức năng ở phía người quản lý và các nhân viên gồm Quản lý nhân viên, Quản lý nhà phân phối, Quản lý khách hàng, Quản lý giá bán, quản lý sản phẩm, quản lý đơn hàng. Tất cả các chức năng này đều yêu cầu phải đăng nhập. Các chức năng phía khách hàng như mua hàng, đặt hàng, xem sản phẩm, quản lý giỏ hàng, xem thông tin tài khoản.

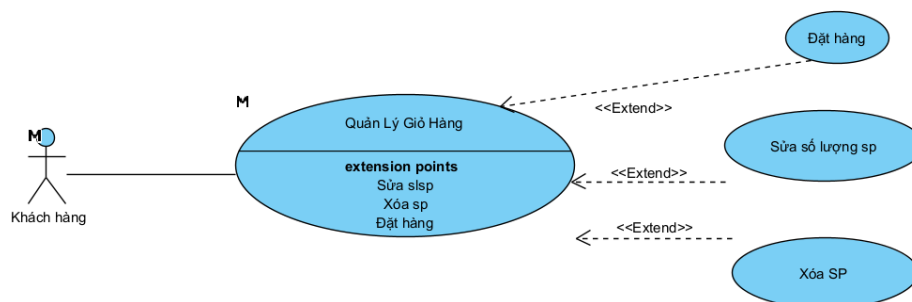
- Biểu đồ phân rã của use case Xem sản phẩm



Biểu đồ 2 Biểu đồ phân rã của use case Xem sản phẩm

Biểu đồ 2 thể hiện biểu đồ phân rã của use case Xem sản phẩm, gồm các use case phân rã là: Xem tại trang chủ, xem theo loại, xem chi tiết,...

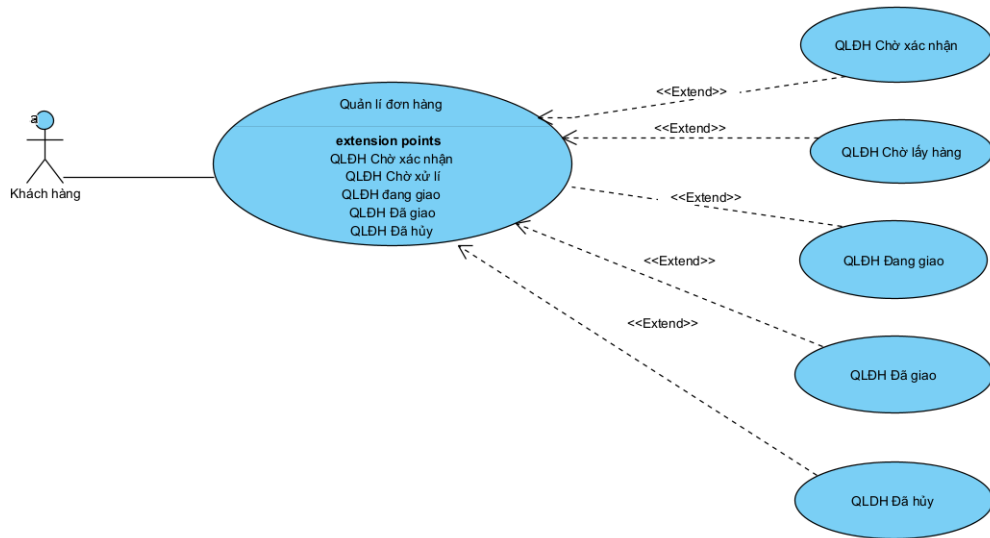
- Biểu đồ phân rã của use case Giỏ hàng



Biểu đồ 3 Biểu đồ phân rã của use case Quản lý giỏ hàng

Biểu đồ 3 thể hiện biểu đồ phân rã của use case Giỏ hàng, gồm các use case phân rã là: Đặt hàng, sửa số lượng sản phẩm, xóa sản phẩm.

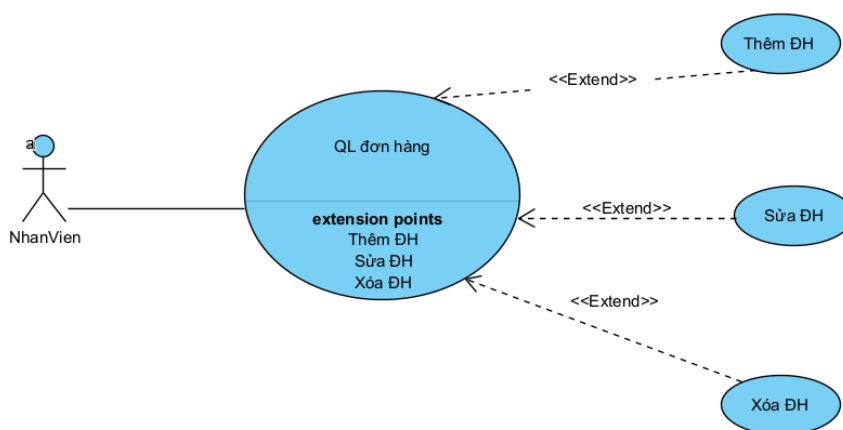
- Biểu đồ phân rã của use case Quản lý đơn hàng



Biểu đồ 4 Biểu đồ phân rã của use case Quản lý đơn hàng

Biểu đồ 4 thể hiện biểu đồ phân rã của use case Quản lý đơn hàng, gồm các use case phân rã là: Chờ xác nhận, Chờ lấy hàng, Đang giao, Đã giao, Đã hủy.

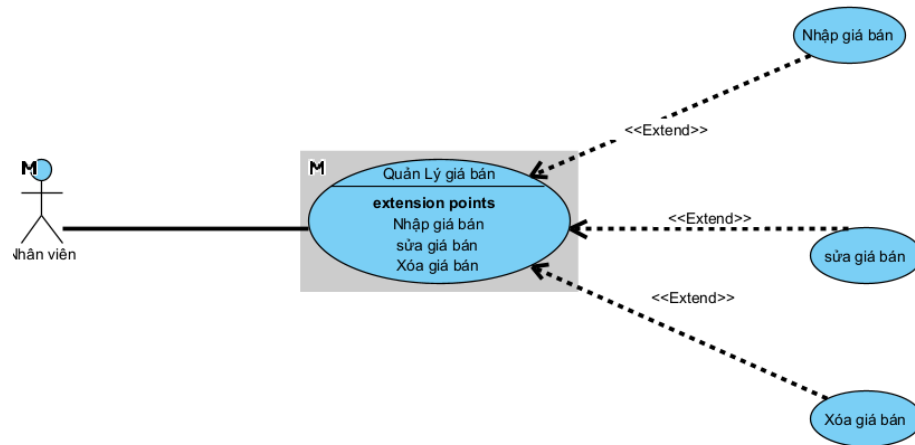
- Biểu đồ phân rã của use case Quản lý đơn hàng



Biểu đồ 5 Biểu đồ phân rã của use case Quản lý đơn hàng

Biểu đồ 5 thể hiện biểu đồ phân rã của use case Quản lý đơn hàng, gồm các use case phân rã là: Thêm, Sửa, Xóa.

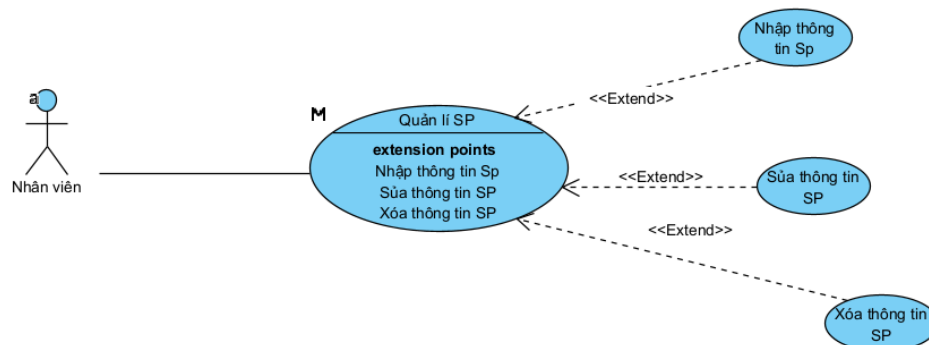
- Biểu đồ phân rã của use case Quản lý giá bán



Biểu đồ 6 Biểu đồ phân rã của use case Quản lý giá bán

Biểu đồ 6 thể hiện biểu đồ phân rã của use case Quản lý giá bán, gồm các use case phân rã là: Nhập giá bán, Sửa giá bán, Xóa giá bán.

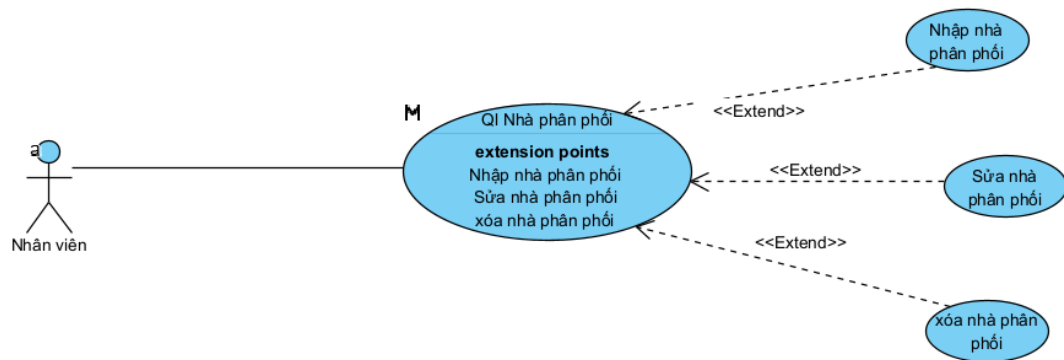
- Biểu đồ phân rã của use case Quản lý sản phẩm



Biểu đồ 7 Biểu đồ phân rã của use case Quản lý sản phẩm

Biểu đồ 7 thể hiện biểu đồ phân rã của use case Quản lý sản phẩm gồm các use case phân rã là: Nhập thông tin sp, Sửa thông tin sp, Xóa thông tin sp.

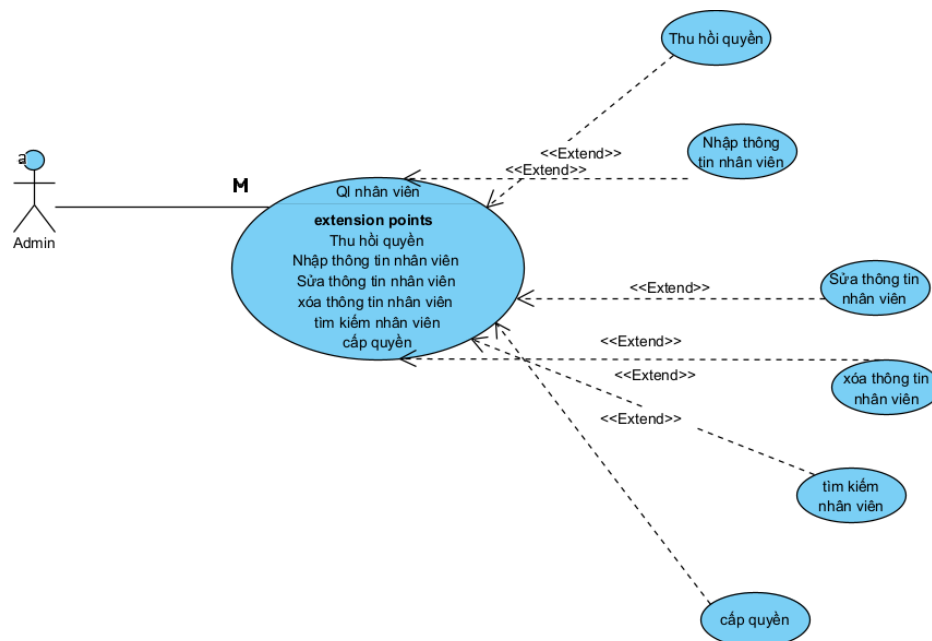
- Biểu đồ phân rã của use case Quản lý nhà phân phối



Biểu đồ 8 Biểu đồ phân rã của use case Quản lý nhà phân phối

Biểu đồ 8 thể hiện biểu đồ phân rã của use case Quản lý nhà phân phối gồm các use case phân rã là: Nhập thông tin nhà phân phối, Sửa thông tin, Xóa thông tin.

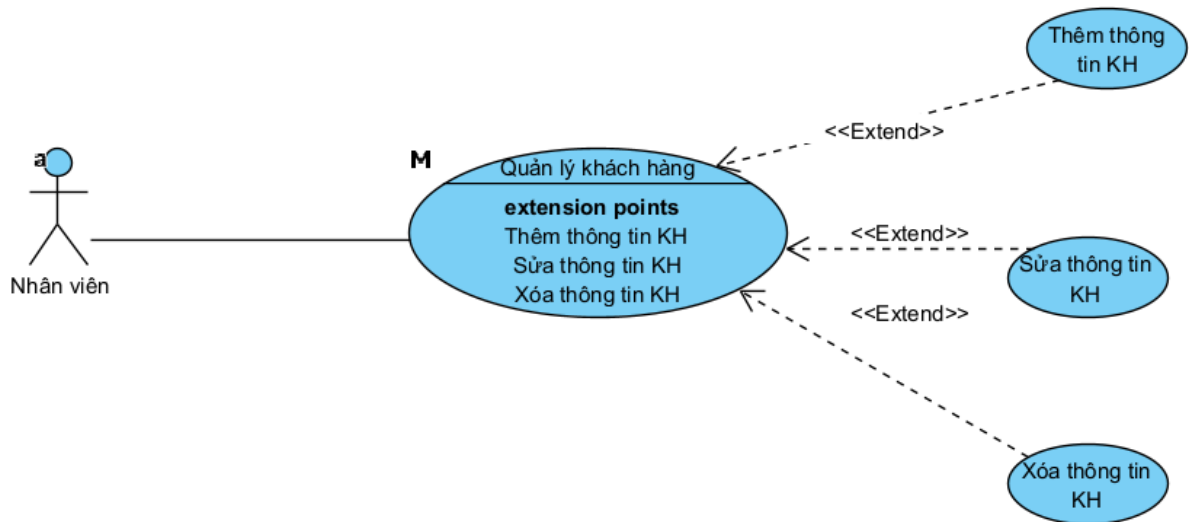
- Biểu đồ phân rã của use case Quản lý nhân viên



Biểu đồ 9 Biểu đồ phân rã của use case Quản lý nhân viên

Biểu đồ 9 thể hiện biểu đồ phân rã của use case Quản lý nhân viên, gồm các use case phân rã là: Thu hồi quyền, Nhập thông tin nhân viên, Sửa thông tin, Xóa thông tin, Tìm kiếm nhân viên, Cấp quyền.

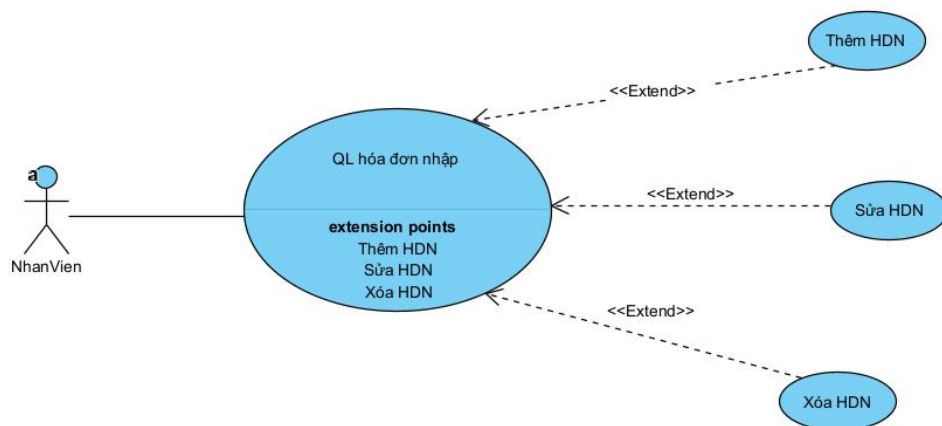
- Biểu đồ phân rã của use case Quản lý khách hàng



Biểu đồ 10 Biểu đồ phân rã của use case Quản lý khách hàng

Biểu đồ 10 thể hiện biểu đồ phân rã của use case Quản lý khách hàng, gồm các use case phân rã là: Thêm thông tin khách hàng, Sửa thông tin, Xóa thông tin.

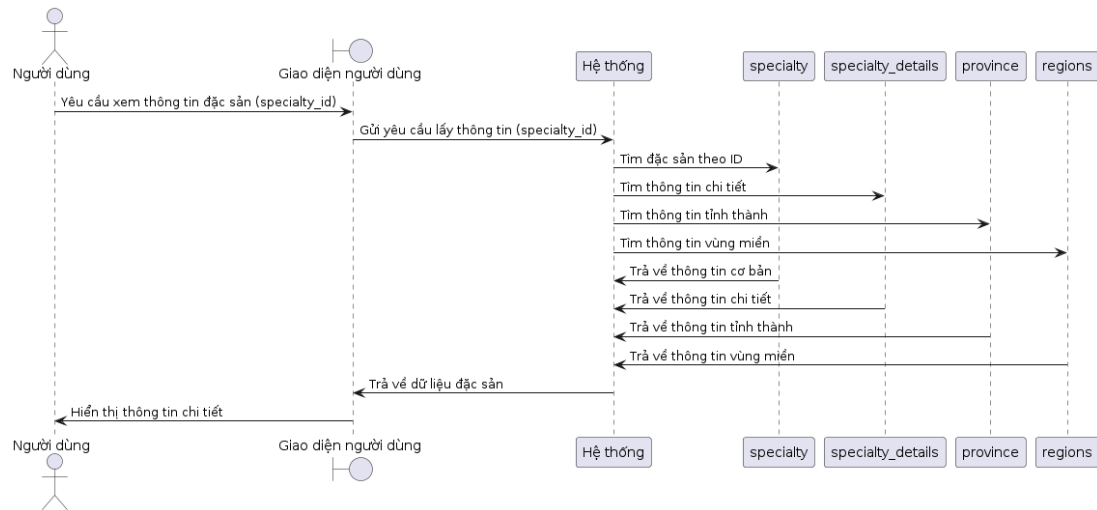
- Biểu đồ phân rã của use case Quản lý hóa đơn nhập



Biểu đồ 11 Biểu đồ phân rã của use case Quản lý hóa đơn nhập

Biểu đồ 11 thể hiện biểu đồ phân rã của use case Quản lý hóa đơn nhập, gồm các use case phân rã là: Thêm hóa đơn nhập, Sửa hóa đơn nhập, Xóa hóa đơn nhập.

- Biểu đồ tuần tự (Sequence Diagram)



Biểu đồ 12: Biểu đồ tuần tự (Sequence Diagram)

Người dùng yêu cầu xem thông tin sản phẩm: Người dùng tương tác với giao diện người dùng (UI) để chọn một sản phẩm cụ thể mà họ muốn xem thông tin.

UI gửi yêu cầu lấy thông tin: Giao diện người dùng gửi yêu cầu đến hệ thống, cung cấp ID của đặc sản (specialtyId) mà người dùng đã chọn.

Hệ thống tìm sản phẩm theo ID: Hệ thống nhận yêu cầu và tìm kiếm thông tin về sản phẩm tương ứng trong cơ sở dữ liệu.

Hệ thống trả về thông tin: Hệ thống trả về các thông tin đã tìm được cho giao diện người dùng.

UI hiển thị thông tin chi tiết: Giao diện người dùng nhận dữ liệu từ hệ thống và hiển thị thông tin chi tiết của sản phẩm, bao gồm cả thông tin về loại sản phẩm và sản phẩm liên quan, cho người dùng xem.

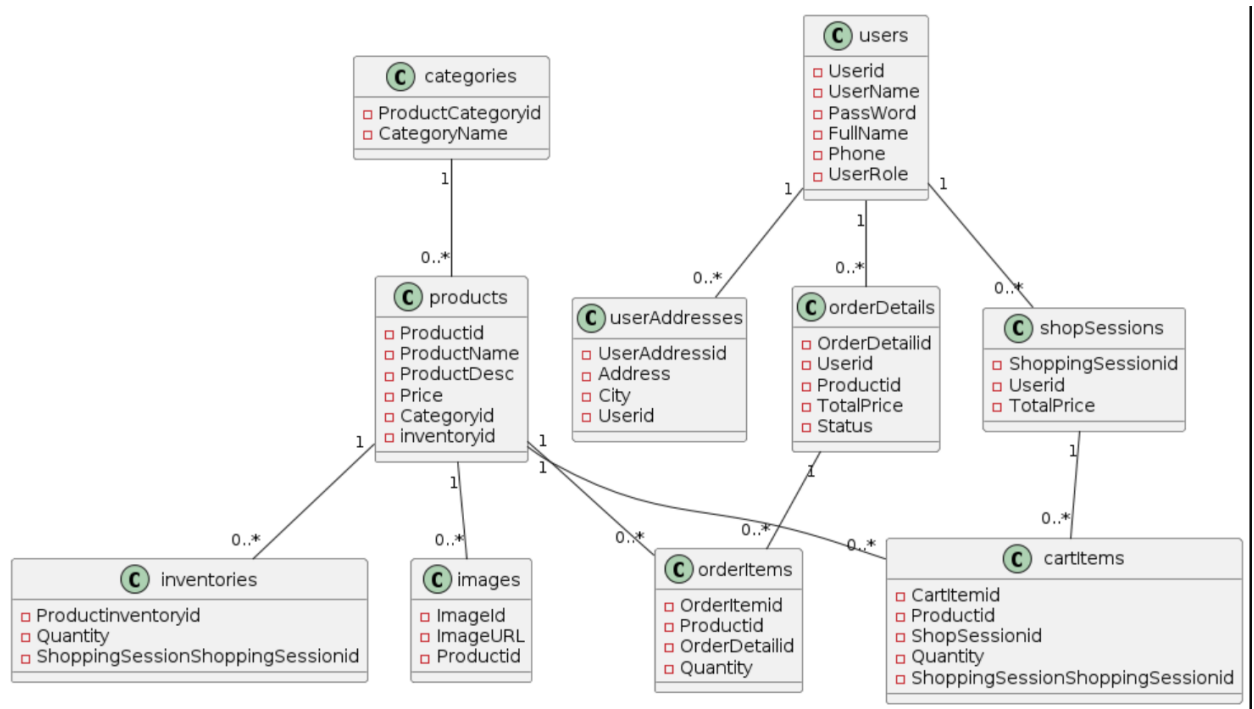
Các thành phần trong biểu đồ:

Người dùng: Người sử dụng ứng dụng.

Giao diện người dùng: Phần giao diện mà người dùng tương tác để xem thông tin sản phẩm.

Hệ thống: Phần xử lý logic và truy xuất dữ liệu của ứng dụng.

- Biểu đồ Class Diagram



Biểu đồ 13: Biểu đồ lớp (Class Diagram)

Biểu đồ mô tả hệ thống quản lý bán hàng trực tuyến. Nó bao gồm các lớp (class) đại diện cho các thực thể chính trong hệ thống như người dùng, sản phẩm, đơn hàng, giỏ hàng, v.v. Các mối quan hệ giữa các lớp này mô tả cách chúng tương tác với nhau.

Các lớp chính:

users (Người dùng): Lưu trữ thông tin về người dùng như Userid, Username, Password, Fullname, Phone, UserRole. Có mối quan hệ 1-nhiều với userAddresses (địa chỉ người dùng), orderDetails (chi tiết đơn hàng) và shopSessions (phiên mua sắm).

products (Sản phẩm): Chứa thông tin về sản phẩm như Productid, ProductName, ProductDesc, Price, Categoryid (loại sản phẩm), inventoryid (tồn kho). Có mối quan hệ nhiều-1 với categories (loại sản phẩm) và inventories (tồn kho).

kho). Có mối quan hệ 1-nhiều với images (hình ảnh sản phẩm), orderItems (mặt hàng trong đơn hàng) và cartItems (mặt hàng trong giỏ hàng).

orderDetails (Chi tiết đơn hàng): Lưu trữ thông tin về đơn hàng như OrderDetailid, Userid, Productid, TotalPrice, Status. Có mối quan hệ 1-nhiều với orderItems (mặt hàng trong đơn hàng).

shopSessions (Phiên mua sắm): Theo dõi các phiên mua sắm của người dùng và chứa thông tin như ShoppingSessionid, Userid, TotalPrice. Có mối quan hệ 1-nhiều với cartItems (mặt hàng trong giỏ hàng).

Các lớp khác:

categories (Loại sản phẩm): Chứa thông tin về các loại sản phẩm như ProductCategoryid, CategoryName.

userAddresses (Địa chỉ người dùng): Lưu trữ địa chỉ của người dùng.

inventories (Tồn kho): Theo dõi số lượng sản phẩm trong kho.

images (Hình ảnh sản phẩm): Lưu trữ hình ảnh liên quan đến sản phẩm.

orderItems (Mặt hàng trong đơn hàng): Chi tiết về các sản phẩm trong một đơn hàng.

cartItems (Mặt hàng trong giỏ hàng): Chi tiết về các sản phẩm trong giỏ hàng của người dùng.

Phân tích mối quan hệ: Mối quan hệ 1-nhiều giữa users và userAddresses, orderDetails, shopSessions: Một người dùng có thể có nhiều địa chỉ, nhiều đơn hàng và nhiều phiên mua sắm.

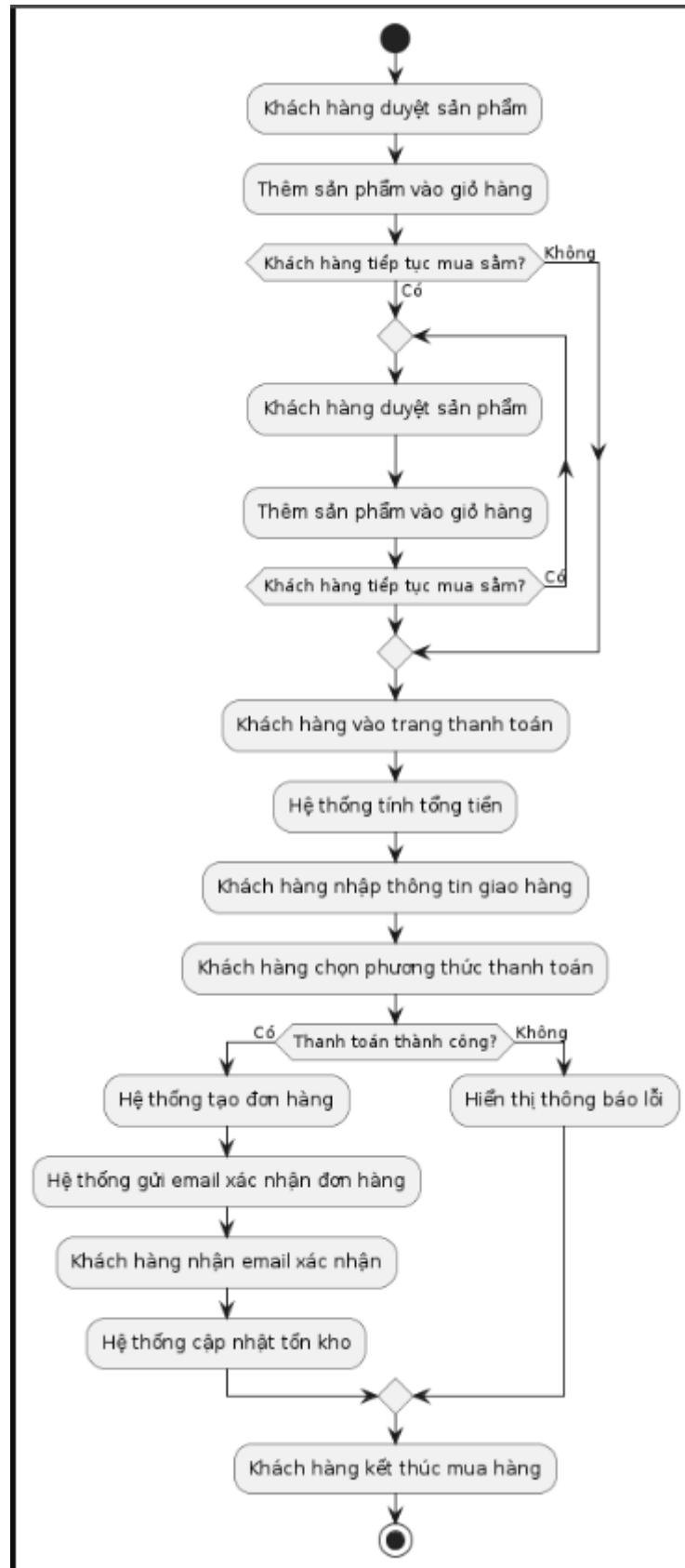
Mối quan hệ nhiều-1 giữa products và categories, inventories: Một sản phẩm thuộc về một loại sản phẩm và có một mục trong bảng tồn kho.

Mối quan hệ 1-nhiều giữa products và images, orderItems, cartItems: Một sản phẩm có thể có nhiều hình ảnh, xuất hiện trong nhiều đơn hàng và giỏ hàng khác nhau.

Mối quan hệ 1-nhiều giữa orderDetails và orderItems: Một đơn hàng có thể chứa nhiều mặt hàng.

Mối quan hệ 1-nhiều giữa shopSessions và cartItems: Một phiên mua sắm có thể chứa nhiều mặt hàng trong giỏ hàng.

- Biểu đồ Activity Diagram



Biểu đồ 14: Biểu đồ hoạt động (Activity Diagram)

Biểu đồ này mô tả quy trình mua hàng trực tuyến của khách hàng, từ lúc bắt đầu duyệt sản phẩm đến khi kết thúc giao dịch.

Khách hàng duyệt sản phẩm: Khách hàng xem các sản phẩm trên website/ứng dụng.

Thêm sản phẩm vào giỏ hàng: Nếu khách hàng thích sản phẩm, họ sẽ thêm vào giỏ hàng.

Khách hàng vào trang thanh toán: Khách hàng bắt đầu quá trình thanh toán.

Hệ thống tính tổng tiền: Hệ thống tự động tính tổng giá trị đơn hàng.

Khách hàng nhập thông tin giao hàng: Khách hàng cung cấp thông tin địa chỉ, số điện thoại để nhận hàng.

Khách hàng chọn phương thức thanh toán: Khách hàng lựa chọn hình thức thanh toán (ví dụ: thẻ tín dụng, chuyển khoản, ví điện tử...).

Hệ thống tạo đơn hàng: Tạo một đơn hàng mới trong hệ thống với các thông tin sản phẩm, khách hàng, địa chỉ giao hàng, phương thức thanh toán.

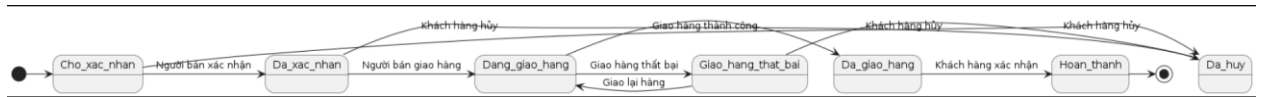
Hệ thống gửi email xác nhận đơn hàng: Gửi email thông báo cho khách hàng về việc đơn hàng đã được tạo thành công, kèm theo các thông tin chi tiết về đơn hàng.

Khách hàng nhận email xác nhận: Khách hàng nhận được email và có thể kiểm tra lại thông tin đơn hàng.

Hệ thống cập nhật tồn kho: Sau khi đơn hàng được tạo, hệ thống sẽ tự động trừ số lượng sản phẩm đã mua khỏi kho.

Khách hàng kết thúc mua hàng: Quy trình mua hàng hoàn tất.

- Biểu đồ State Diagram



Biểu đồ 15: Biểu đồ trạng thái (State Diagram)

Biểu đồ minh họa vòng đời của một đơn hàng trong hệ thống bán hàng trực tuyến. Các trạng thái và chuyển đổi trạng thái được thể hiện rõ ràng:

- Trạng thái:

Chờ xác nhận: Đơn hàng mới được tạo, đang chờ người bán xác nhận. **Đã xác nhận:** Đơn hàng đã được người bán xác nhận và đang chờ xử lý tiếp theo.

Đang giao hàng: Đơn hàng đang được vận chuyển đến khách hàng.

Đã giao hàng: Đơn hàng đã được giao đến khách hàng, chờ khách hàng xác nhận.

Hoàn thành: Khách hàng đã xác nhận nhận hàng thành công, đơn hàng hoàn tất.

Đã hủy: Đơn hàng đã bị hủy bởi khách hàng hoặc người bán.

Giao hàng thất bại: Đơn hàng không thể giao thành công.

- Chuyển đổi trạng thái:

Người bán xác nhận: Chuyển từ "Chờ xác nhận" sang "Đã xác nhận".

Người bán giao hàng: Chuyển từ "Đã xác nhận" sang "Đang giao hàng".

Giao hàng thành công: Chuyển từ "Đang giao hàng" sang "Đã giao hàng".

Khách hàng xác nhận: Chuyển từ "Đã giao hàng" sang "Hoàn thành".

Khách hàng hủy: Chuyển từ "Chờ xác nhận", "Đã xác nhận", "Đang giao hàng" hoặc "Giao hàng thất bại" sang "Đã hủy".

Giao lại hàng: Chuyển từ "Giao hàng thất bại" sang "Đang giao hàng".

3.3. Thiết kế cơ sở dữ liệu

3.3.1. Bảng SanPham

Bảng 3.1. Bảng sản phẩm

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	Sanpham_Id	Int	Id sản phẩm (Primary Key)
2	Tensp	Varchar(255)	Tên sản phẩm
3	Loaisp_Id	Int	Id loại sản phẩm
4	Chatlieu	Varchar(255)	Chất liệu sản phẩm
5	Soluongton	Varchar(255)	Số lượng sản phẩm
6	Image	Varchar(255)	Ảnh sản phẩm
7	Mota	Varchar(255)	Mô tả thông tin sản phẩm
8	Giasp	Varchar(255)	Giá của sản phẩm

Bảng Sản Phẩm chứa các thông tin về Sanpham_Id là khóa chính được sinh ra ngẫu nhiên trong quá trình thêm mới ngoài ra còn có tên sản phẩm, các thông tin liên quan đến sản phẩm đó.

3.3.2. Bảng KháchHàng

Bảng 3.2. Bảng khách hàng

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	Khachhang_id	int	Id khách hàng (Primary Key)
2	TenKhachHang	Varchar(255)	Tên khách hàng

3	DiaChi	Varchar(255)	Địa chỉ của khách hàng
4	Phone	int	Số điện thoại
5	Email	Varchar(255)	Email của khách hàng

Bảng KháchHang chứa các thông tin về Id khách hàng là khóa chính được sinh ra ngẫu nhiên trong quá trình thêm mới. Ngoài ra còn có tên của khách hàng, một số thông tin cơ bản của khách như địa chỉ, số điện thoại ,...

3.3.3. Bảng *LoaiSanPham*

Bảng 3.3. Bảng loại sản phẩm

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	Loaisp_id	int(36)	Id loại sản phẩm (Primary Key)
2	TenLoaiSanPham	Varchar(255)	Tên loại sản phẩm

Bảng LoaiSanPham chứa các thông tin về Id loại sản phẩm là khóa chính được sinh ra trong quá trình thêm mới. Bảng còn chứa các thông tin như là tên của Loại sản phẩm.

3.3.4. Bảng *HoaDon*

Bảng 3.4. Bảng Hóa đơn

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	HoaDon_id	Int	Id hóa đơn (Primary Key)

2	Ngaydathang	Datetime	Thời gian khách đặt hàng
3	Ngaygiaohang	Datetime	Thời gian bắt đầu giao hàng
4	Diachigiaohang	Varchar(255)	Địa chỉ khách nhận hàng
5	Khachhang_id	Int	Id khách hàng sản (Foreign Key)
6	Chitietdonhang_id	Int	Id chi tiết đơn hàng sản (Foreign Key)
7	Trangthai_id	Int	Id đặc sản (Foreign Key)

Bảng hóa đơn chứa thông tin về: Id hóa đơn là khóa chính được sinh ra ngẫu nhiên trong quá trình thêm mới, thời gian giao nhận sản phẩm, địa chỉ giao hàng, Khachhang_id, Chitietdonhang_id, Trangthai_id là khóa phụ liên kết tới các bảng khách hàng, trạng thái, chi tiết hóa đơn.

3.3.5. Bảng *TrangThai*

Bảng 3.5. Bảng trạng thái

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	TrangThai_id	Int	Id trạng thái (Primary Key)
2	TenTrangThai	Varchar(255)	Trang thái của đơn hàng

Bảng TrangThai chứa các thông tin là Id trạng thái được sinh ra ngẫu nhiên và trạng thái của đơn hàng.

3.3.6. Bảng Account

Bảng 3.6. Bảng Account

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	account_id	Varchar(36)	Id tài khoản (Primary Key)
2	Username	Varchar(255)	Tên đăng nhập
3	Password	Varchar(255)	Mật khẩu

Bảng Account chứa các thông tin: Id tài khoản là khóa chính được sinh ra ngẫu nhiên trong quá trình tạo mới tài khoản, username và password là 2 trường giúp Admin có thể lấy đó làm cái đăng nhập vào ứng dụng để quản lý sản phẩm, cũng như đơn hàng.

3.3.7. Bảng Chitiethoadon

Bảng 3.7. Bảng chi tiết hóa đơn

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	Chitiethoadon_id	Int	Id chi tiết hóa đơn (Primary Key)
2	HoaDon_id	Int	Id hóa đơn (Foreign Key)
3	Sanpham_id	Int	Id sản phẩm (Foreign Key)
4	SoLuong	Int	Số lượng sản phẩm

5	DonGia	Varchar(255)	Giá của từng sản phẩm
6	ThanhTien	Varchar(255)	Tổng tiền của hóa đơn

Bảng Chitiethoadon chứa các thông tin: Id chi tiết hóa đơn là khóa chính được sinh ra ngẫu nhiên. Các trường Số lượng, đơn giá, thành tiền là các trường lưu trữ thông tin của hóa đơn. HoaDon_id, Sanpham_id là khóa phụ liên kết tới bảng Chitiethoadon.

3.3.8. Bảng GioHang

Bảng 3.8. Bảng giỏ hàng

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	GioHang_id	Int	Id giỏ hàng (Primary Key)
2	Sanpham_id	Int	Id sản phẩm (Foreign Key)
3	Khachhang_id	Int	Id khách hàng sản (Foreign Key)
4	ThanhTien	Varchar(255)	Tổng tiền của sản phẩm trong giỏ hàng

Bảng GioHang chứa các trường thông tin: Id giỏ hàng là khóa chính được sinh ra ngẫu nhiên trong quá trình thêm mới giỏ hàng. Sanpham_id, KhachHang_id là khóa phụ liên kết tới các bảng Sản phẩm, khách hàng, Thành tiền sẽ cho biết số tiền phải trả khi thanh toán những sản phẩm trong giỏ hàng.

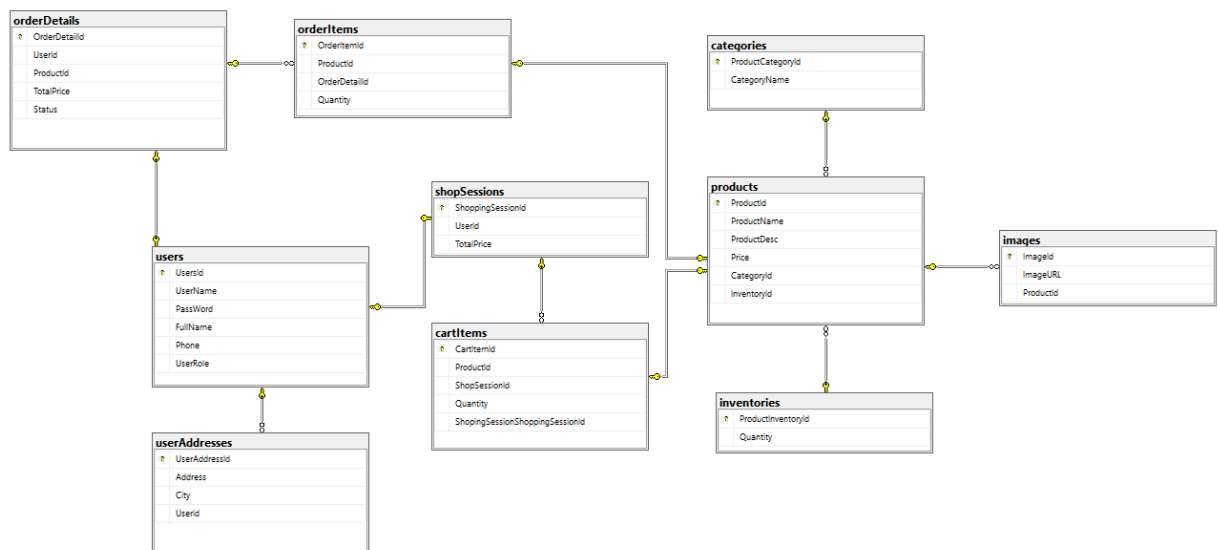
3.3.9. Bảng Kho

Bảng 3.9. Bảng kho

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	Int	Id kho (Foreign Key)
2	SoLuong	int	Số lượng sản phẩm tồn kho

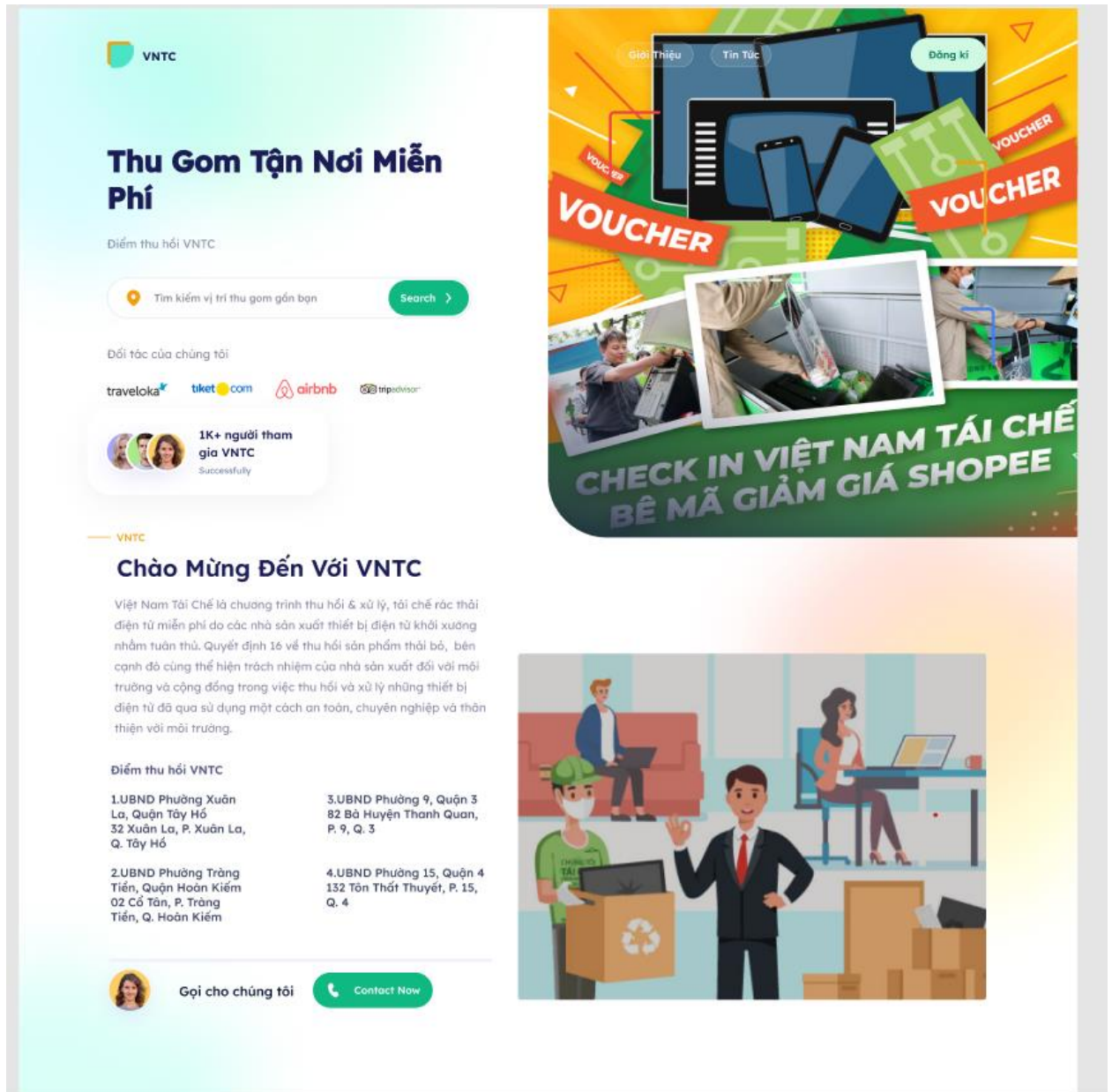
Bảng kho là Id kho là khóa chính được sinh ra ngẫu nhiên. SoLuong là số lượng sản phẩm còn trong kho hàng.

- Biểu đồ Database Diagram

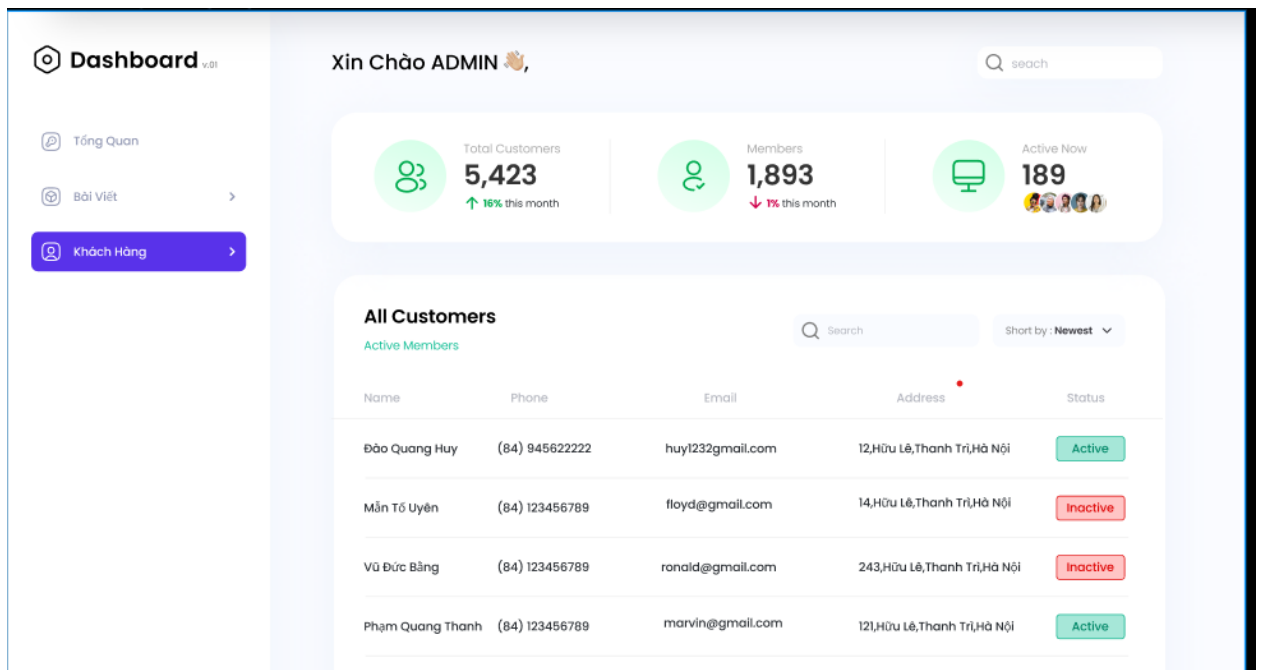


Biểu đồ 16: Biểu đồ tổng quát của cơ sở dữ liệu

3.4. UI/UX Design



Hình 1: Trang design giao diện người dùng



Hình 2: Trang design giao diện Admin

Quản lý hệ thống:

Đây là chức năng cốt lõi, chịu trách nhiệm quản lý tổng thể ứng dụng. Các chức năng con bao gồm:

Quản lý tài khoản: Quản lý thông tin người dùng, đăng ký, đăng nhập.

Quản lý sản phẩm: Thêm, sửa, xóa, cập nhật thông tin về sản phẩm.

Quản lý loại sản phẩm: Thêm, sửa, xóa, cập nhật thông tin về loại sản phẩm.

Quản lý giỏ hàng: Thêm, sửa, xóa, cập nhật thông tin về giỏ hàng.

Quản lý hóa đơn: Thêm, sửa, xóa, cập nhật thông tin về hóa đơn.

Quản lý người dùng:

Chức năng này tập trung vào việc quản lý thông tin và hoạt động của người dùng trên ứng dụng. Các chức năng con bao gồm:

Xem thông tin cá nhân: Admin có thể xem và chỉnh sửa thông tin cá nhân của mình, khách hàng.

Xem danh sách sản phẩm: Người dùng có thể xem danh sách các sản phẩm đã lưu vào mục giỏ hàng.

Quản lý sản phẩm:

Chức năng này tập trung vào việc quản lý thông tin về sản phẩm trên web. Các chức năng con bao gồm:

Xem danh sách sản phẩm: Người dùng có thể xem danh sách tất cả các đặc sản có trên web.

Xem chi tiết sản phẩm: Người dùng có thể xem thông tin chi tiết về một đặc sản cụ thể (nguồn gốc, hình ảnh, thành phần,...).

Tìm kiếm sản phẩm: Người dùng có thể tìm kiếm sản phẩm theo tên.

Quản lý loại sản phẩm: Chức năng này tập trung vào việc quản lý thông tin về loại sản phẩm trên web.

Các chức năng con bao gồm: Xem danh sách loại sản phẩm: Người dùng có thể xem danh sách tất cả các loại sản phẩm có trên web.

Xem danh sách sản phẩm theo loại: Người dùng có thể xem danh sách các sản phẩm thuộc một loại sản phẩm cụ thể.

Quản lý giỏ hàng: Chức năng này tập trung vào việc quản lý thông tin về giỏ hàng trên web.

Các chức năng con bao gồm: Xem danh sách sản phẩm trong giỏ hàng: Người dùng có thể xem danh sách tất cả các sản phẩm trong giỏ hàng có trên web.

3.5. Giao diện và chức năng

3.5.1. Trang giao diện người dùng

Trang chủ là trang mặc định khi người dùng truy cập vào ứng dụng. Các chức năng và nội dung của trang chủ có thể bao gồm:

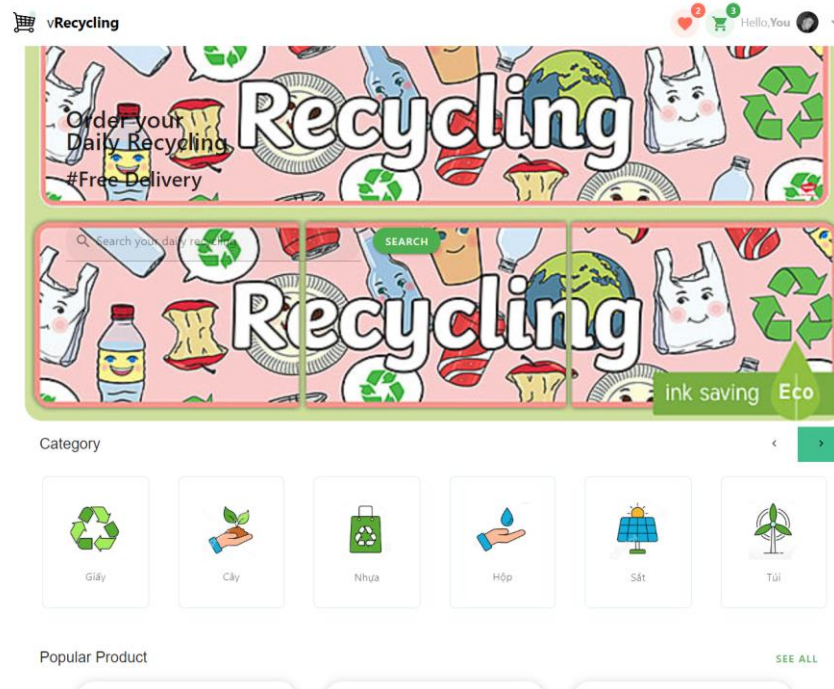
Cung cấp chức năng tìm kiếm để người dùng có thể tìm kiếm sản phẩm theo tên.

Hiển thị các sản phẩm nổi bật hoặc mới nhất.

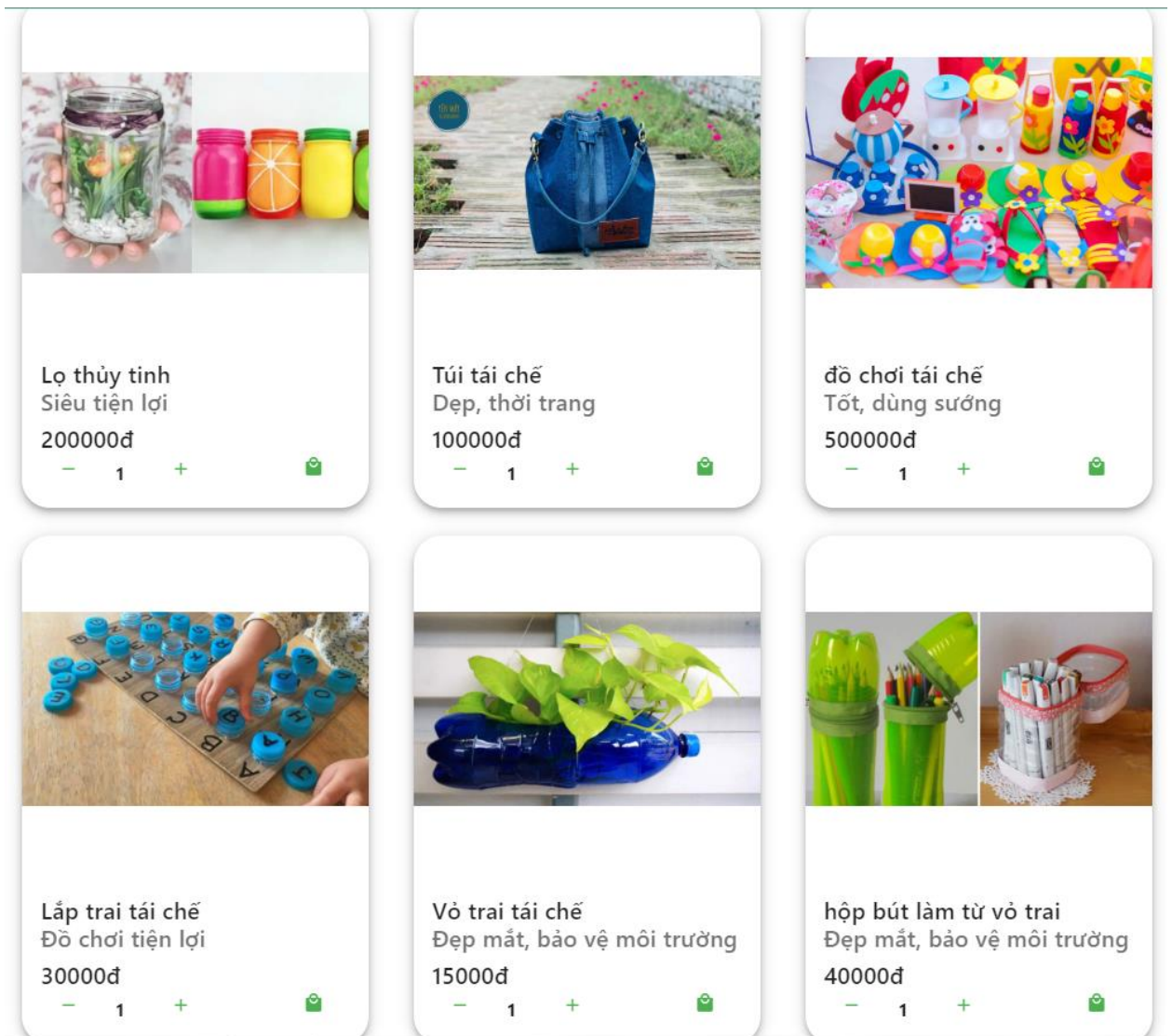
Cung cấp liên kết đến các thể loại sản phẩm tái chế khác nhau.

Hiển thị danh sách các sản phẩm tái chế theo thể loại.

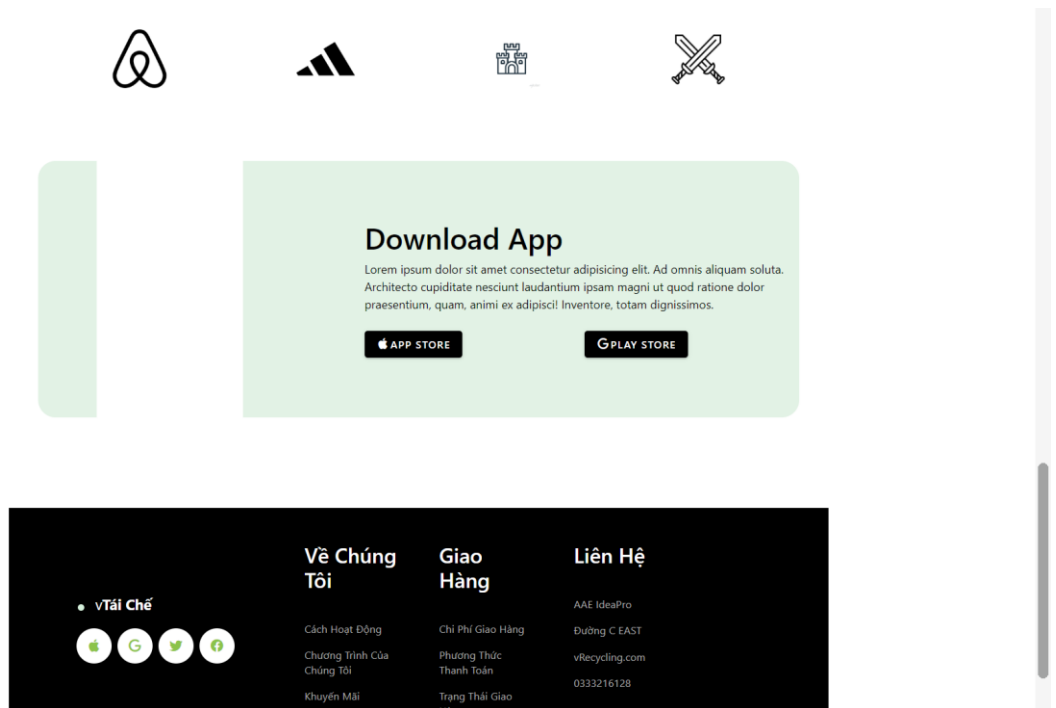
Hiển thị chi tiết sản phẩm tái chế khi người dùng chọn một sản phẩm tái chế cụ thể.



Hình 3: Trang chủ(1) giao diện User



Hình 4: Trang chủ(2) giao diện User

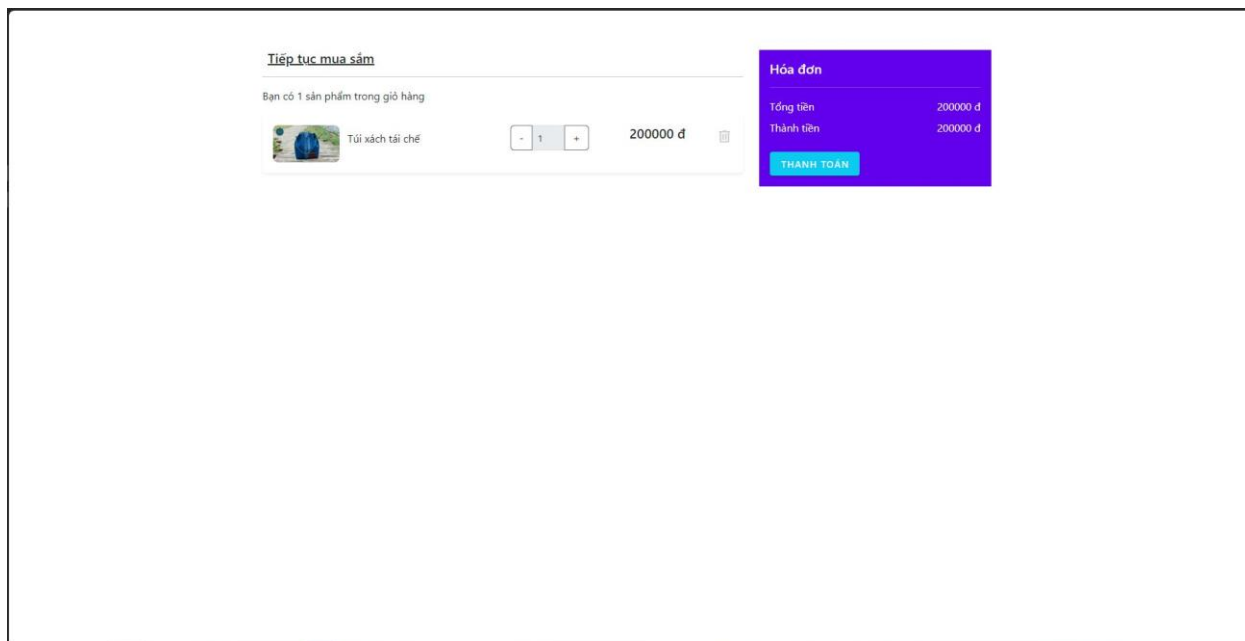


Hình 5: Trang chủ (3) giao diện User

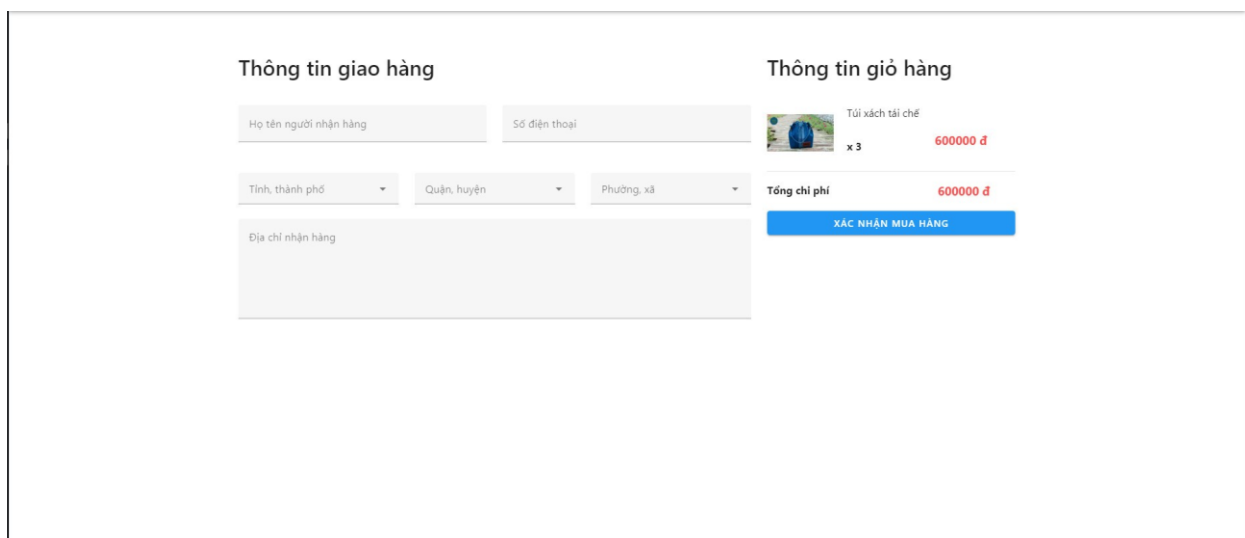
Khi khách hàng muốn mua 1 sản phẩm tái chế bất kì, khách hàng sẽ phải cho số lượng và nhấn vào giỏ hàng để bên dưới sản phẩm để chuyển sản phẩm đến trang giỏ hàng.

Khi khách hàng đã chọn được những sản phẩm mua, các sản phẩm này sẽ được cập nhật trong giỏ hàng.

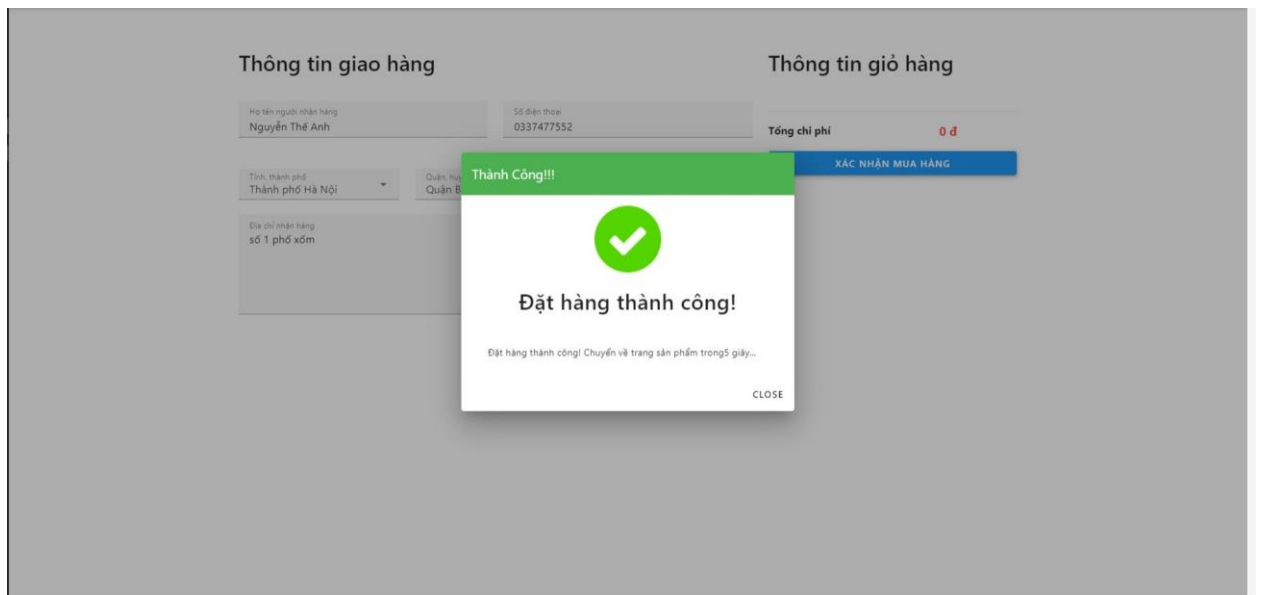
Khi khách hàng nhấn thanh toán sẽ được chuyển đến trang nhập thông tin giao hàng, ở đó khách hàng phải điền đủ thông tin và sau đó nhấn xác nhận mua hàng sẽ có thông báo xác nhận khách hàng đã đặt hàng thành công.



Hình 6: Trang giỏ hàng giao diện User



Hình 7: Trang thông tin giao hàng giao diện User



Hình 8: Trang thông báo đặt hàng thành công giao diện User

3.5.2. Trang đăng nhập, đăng ký

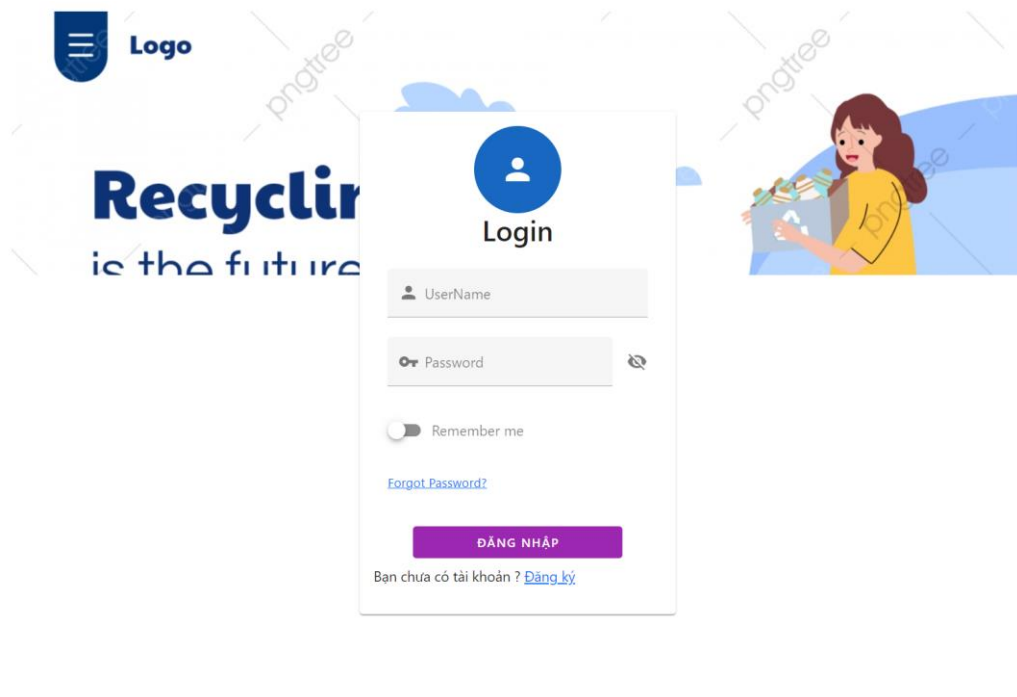
Trang đăng nhập và đăng ký không chỉ là nơi để người dùng nhập thông tin tài khoản, mà còn là nơi tạo và quản lý token cho việc xác thực. Dưới đây là mô tả về cách token được sử dụng trong hai trang này:

- Đăng nhập

Xác thực thông qua JWT (JSON Web Token): Sau khi người dùng nhập thông tin đăng nhập và gửi yêu cầu, máy chủ sẽ xác thực thông tin và tạo một JWT chứa thông tin xác thực như ID người dùng và thời gian hết hạn.

Lưu trữ token: Token JWT được lưu trữ trên phía máy khách, thường được lưu trong cookie hoặc local storage để sử dụng cho các yêu cầu tiếp theo mà không cần người dùng nhập lại thông tin đăng nhập.

Xác thực và phân quyền: Mỗi lần người dùng gửi yêu cầu đến máy chủ, token sẽ được gửi kèm theo để xác thực. Máy chủ sẽ kiểm tra tính hợp lệ của token và dựa vào đó phân quyền truy cập cho người dùng.

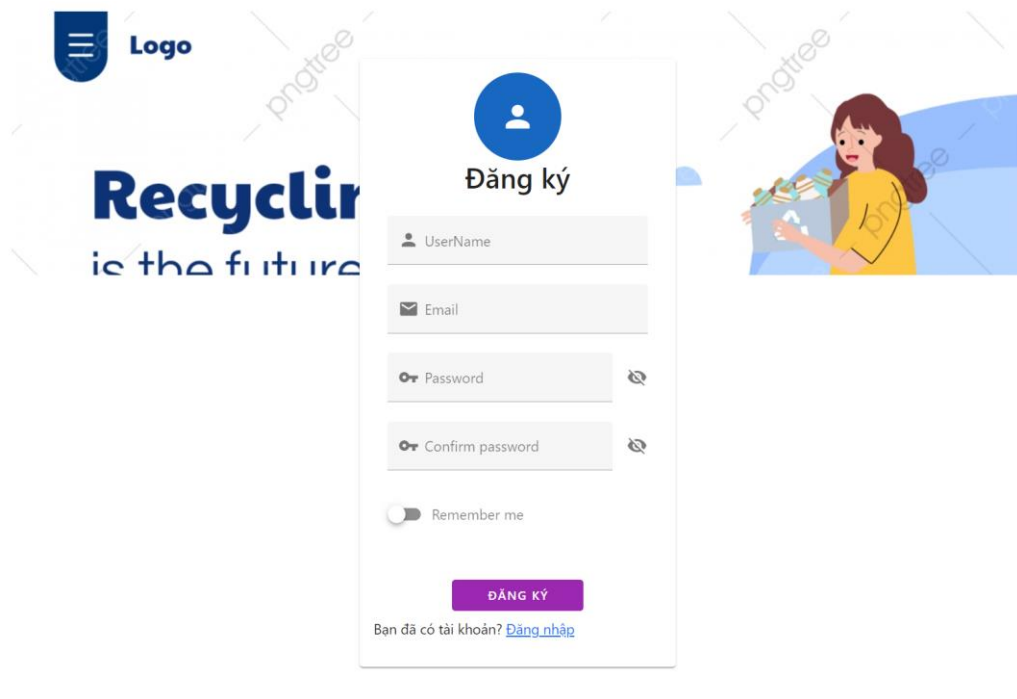


Hình 9: Trang đăng nhập giao diện Admin

- Đăng ký

Tạo token cho tài khoản mới: Sau khi người dùng nhập thông tin và gửi yêu cầu đăng ký, máy chủ sẽ tạo một token mới và gửi lại cho người dùng. Token này có thể được sử dụng ngay lập tức để đăng nhập mà không cần thực hiện lại quá trình đăng nhập.

Xác thực và gửi token: Máy chủ sẽ gửi token cho người dùng qua email hoặc trả về trong phản hồi sau khi đăng ký thành công. Người dùng sẽ sử dụng token này để đăng nhập lần đầu tiên và kích hoạt tài khoản.



Hình 10: Trang đăng ký giao diện Admin

3.5.3. Trang giao diện Admin

Trang quản lý là nơi quản lý các chức năng quản trị của ứng dụng. Các chức năng có thể bao gồm:

Quản lý tài khoản: Thêm, sửa, xóa người dùng, cũng như quản lý vai trò và quyền hạn của họ.










Quản lý sản phẩm: Hiển thị danh sách sản phẩm tái chế, cho phép quản lý và thao tác các sản phẩm tái chế như thêm, sửa, xóa.

Quản lý loại sản phẩm: Hiển thị danh sách các thể loại sản phẩm tái chế, cho phép thêm, sửa, xóa thể loại.










Quản lý giỏ hàng: Xem, Thêm, sửa, xóa các giỏ hàng .

Quản lý đơn hàng: Xem, Thêm, sửa, xóa các đơn hàng.












Quản lý kho: Xem, Thêm, sửa, xóa các số lượng sản phẩm trong kho.

Quản lý sản phẩm								
STT	Loại sản phẩm	Tên sản phẩm	Chất liệu	Số lượng	Hình ảnh	Giá	Chức năng	
40	Giấy	Hộp đựng bút	Giấy	4		30000		
41	Giấy	Giá sách	Giấy	5		25000		
42	Giấy	Đèn trần	Giấy	6		45000		

Hình 11: Trang quản lý sản phẩm(1) giao diện Admin

Quản lý sản phẩm								
STT	Loại sản phẩm	Tên sản phẩm	Chất liệu	Số lượng	Hình ảnh	Giá	Chức năng	
42	Giấy	Đèn trần	Giấy	6		45000		
43	Nhựa	Bình hoa	Chai nhựa	10		20000		
44	Nhựa	Chậu trồng cây	Chai nhựa	7		12000		

Hình 12: Trang quản lý sản phẩm(2) giao diện Admin

Quản lý sản phẩm							
				<div>+</div> <div>Q</div>			
STT	Loại sản phẩm	Tên sản phẩm	Chất liệu	Số lượng	Hình ảnh	Giá	Chức năng
46	Vải cũ	Balo	Vải cũ	5		70000	 
48	Vải cũ	Gối	Vải cũ	6		30000	 
49	Hộp	Tủ cá nhân	Hộp cát tông	4		45000	 
50	Hộp	Hộp đựng bút	Hộp	7		10000	

Hình 13: Trang quản lý sản phẩm(3) giao diện Admin

Tiểu kết:

Chương 3 đã trình bày chi tiết quá trình thiết kế và cài đặt chương trình. Các bảng cơ sở dữ liệu được thiết kế cẩn thận để lưu trữ hiệu quả thông tin quan trọng, đồng thời đảm bảo tính toàn vẹn và nhất quán dữ liệu thông qua các mối quan hệ rõ ràng.

Sơ đồ Use Case đã phác họa một cách trực quan các tương tác giữa người dùng và hệ thống, làm nổi bật các chức năng chính và cách thức người dùng sẽ tương tác với chương trình.

Giao diện người dùng được thiết kế với mục tiêu tối ưu hóa trải nghiệm người dùng, chú trọng vào tính dễ sử dụng, trực quan và thân thiện.

CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết quả đạt được

Qua quá trình thực hiện đề tài em đã đạt được một số kết quả:

- Thiết kế giao diện đơn giản, thân thiện, dễ sử dụng mang lại hiệu quả trải nghiệm tốt đối với người dùng, phù hợp với người dùng ở nhiều độ tuổi.
- Nhờ vào khả năng tối ưu hiệu năng của vueify, ứng dụng hoạt động mượt mà, thời gian phản hồi nhanh.
- Hệ thống API được xây dựng bằng ASP.net web API mang lại khả năng quản lý dữ liệu và hiệu quả bảo mật cao đồng thời tích hợp với cơ sở dữ liệu Sql Server và có thể mở rộng quy mô trong tương lai.
- Ứng dụng cung cấp cho người dùng một số chức năng như đặt hàng, tìm kiếm sản giúp người dùng dễ dàng tìm kiếm sản phẩm mong muốn, hiển thị các thông tin về sản phẩm mà khách hàng đang quan tâm, lưu trữ những sản phẩm yêu thích vào giỏ hàng giúp dễ dàng đặt mua sau này.

4.2. Hạn chế

Ngoài những thành tựu đã đạt được, em cũng gặp những hạn chế nhất định trong quá trình thực hiện đề tài:

- Ứng dụng gặp một số vấn đề về hiệu năng trên các thiết bị cũ và khi dữ liệu ứng dụng ngày càng tăng sẽ khiến việc xử lý ở những thiết bị này gặp vấn đề.
- Ứng dụng chưa được tích hợp với hệ thống chức năng phân quyền làm giảm khả năng mở rộng của các chức năng liên quan tới thương mại điện tử.
- Ứng dụng phụ thuộc vào Internet nên có thể sẽ gây ra trải nghiệm không tốt đối với người dùng ở những khu vực có kết nối mạng kém.
- Ứng dụng hiện chưa có tính năng đề xuất sản phẩm dựa trên sở thích và lịch sử tìm kiếm của người dùng.
- Chưa tích hợp các dịch vụ thanh toán như thanh toán ngân hàng nào,...

- Ứng dụng còn hạn chế hiển thị ngôn ngữ, chưa quốc tế hóa trong việc hỗ trợ đa ngôn ngữ.

- Thời gian thực hiện đề tài, khả năng của bản thân còn hạn chế dẫn tới giao diện chức năng vẫn còn sơ sài so với mục tiêu ban đầu đặt ra.

4.3. Hướng phát triển

Từ kết quả đạt được và hạn chế còn tồn đọng, đề tài của em cũng có những giải pháp và hướng phát triển trong tương lai:

- Tiếp tục tối ưu hiệu năng cả trên các thiết bị cấu hình thấp, các điện thoại khác nhau, máy tính bảng nhằm nâng cao khả năng tương thích của ứng dụng, đảm bảo trải nghiệm liền mạch với nhiều thiết bị di động khác nhau.

- Phát triển phiên bản dành cho hệ điều hành IOS.

- Triển khai các kỹ thuật tối ưu cơ sở dữ liệu như sharding, replication để xử lý dữ liệu một cách hiệu quả.

- Áp dụng các kỹ thuật machine learning để giúp cá nhân hóa trải nghiệm người dùng.

- Phát triển thêm các chức năng như quản lý thời gian giao hàng, theo dõi vận chuyển. Tích hợp hệ thống thanh toán như ví điện tử, thẻ tín dụng giúp người dùng có thể dễ dàng trong việc mua bán đặc sản.

- Phát triển việc hỗ trợ nhiều ngôn ngữ nhằm mở rộng đối tượng người dùng không những trong nước mà cả nước ngoài giúp họ dễ tiếp cận với ứng dụng và văn hóa, đặc sản vùng miền nơi họ đã đến hoặc dự định sẽ đến.

- Phát triển một số chức năng cho phép người dùng tùy chỉnh theo sở thích cá nhân như thay đổi chế độ sáng tối, thay đổi font chữ, kích thước hiển thị, thay đổi ngôn ngữ hiển thị,...

- Tích hợp Google Map vào ứng dụng giúp người dùng có thể dễ dàng định vị và khám phá các địa điểm của cửa hàng trên khắp Việt Nam.

- Thêm các chức năng mạng xã hội như tính năng đánh giá, bình luận, chia sẻ giúp tạo nên cộng đồng sôi nổi cũng như lan tỏa thông tin và thu hút người dùng mới một cách tự nhiên.

- Nghiên cứu và phát triển các tính năng như đề xuất sản phẩm dựa trên sở thích, lịch sử tìm kiếm của người dùng,...

- Phát triển phiên bản mobile của ứng dụng hướng tới tiếp cận cả những đối tượng không sử dụng hoặc không biết tới ứng dụng web này.

- Xây dựng chiến lược marketing để quảng bá, giới thiệu ứng dụng tới người dùng không những trong nước mà cả nước ngoài.

Tiểu kết:

Chương 4 đã tổng kết những kết quả đạt được của đề tài, bao gồm việc thiết kế giao diện thân thiện, tối ưu hiệu năng ứng dụng, xây dựng hệ thống API bảo mật và cung cấp các chức năng hữu ích cho người dùng.

Tuy nhiên, đề tài cũng gặp một số hạn chế như vấn đề hiệu năng trên thiết bị cũ, chưa tích hợp thanh toán, phụ thuộc vào Internet và chưa có phiên bản mobile.

Hướng phát triển trong tương lai tập trung vào việc cải thiện hiệu năng, phát triển phiên bản mobile, tối ưu cơ sở dữ liệu, cá nhân hóa trải nghiệm người dùng và bổ sung các tính năng như giỏ hàng, thanh toán, hỗ trợ đa ngôn ngữ.

Bên cạnh đó, việc tích hợp Google Maps, thêm các chức năng mạng xã hội, phát triển tính năng ngoại tuyến cũng là những hướng đi tiềm năng để phát triển.

Cuối cùng, việc xây dựng cộng đồng người dùng và quảng bá ứng dụng sẽ góp phần đưa ứng dụng đến gần hơn với người dùng trong và ngoài nước.

KẾT LUẬN

Trong quá trình thực hiện đề tài "Xây dựng website giới thiệu sản phẩm thủ công/tái chế", chúng tôi đã đạt được những kết quả đáng kể. Website đã được xây dựng với giao diện thân thiện, tính năng tương tác cao và nội dung phong phú. Đây không chỉ là một nền tảng giới thiệu sản phẩm mà còn là một kênh kết nối giữa người sản xuất và người tiêu dùng, góp phần thúc đẩy phong cách sống bền vững và bảo vệ môi trường.

Qua quá trình thực hiện dự án, chúng tôi nhận thấy sự quan trọng của việc phát triển các sản phẩm thủ công và tái chế, cũng như việc tạo ra các kênh tiếp cận và quảng cáo cho những sản phẩm này. Website đã giúp tạo ra cơ hội kinh doanh mới cho các doanh nghiệp và cá nhân sản xuất các sản phẩm này, đồng thời hỗ trợ giảm lượng rác thải và bảo vệ môi trường.

Tuy nhiên, để website có thể phát triển bền vững và hiệu quả, chúng tôi nhận thấy cần phải tiếp tục cải thiện và phát triển thêm các tính năng và nội dung, đồng thời chú trọng đến việc quảng cáo và tiếp thị để nâng cao nhận thức và sự chấp nhận của cộng đồng đối với sản phẩm thủ công và tái chế.

Cuối cùng, chúng tôi xin bày tỏ lòng biết ơn sâu sắc đến các giảng viên, cán bộ hướng dẫn và các đồng nghiệp đã hỗ trợ và động viên chúng tôi trong suốt quá trình thực hiện dự án. Đây là một trải nghiệm quý báu và đáng nhớ đối với chúng tôi trong quá trình học tập và phát triển nghề nghiệp của mình.

TÀI LIỆU THAM KHẢO

Tiếng Việt:

- [1] Thiết Kế Giao Diện Người Dùng (UI) của Luận Nguyễn
- [2] Phân Tích Kinh Doanh của Nguyễn Ngọc Dũng

Tiếng Anh:

- [1] The Business Analyst Handbook của James Cadle
- [2] Becoming a Business Analyst của IIBA

Danh mục các Website tham khảo:

- [1] <https://www.w3schools.com/>
- [2] <http://www.c-sharpcorner.com/>
- [3] [Tài liệu video trên kênh youtube: Kteam.](#)
- [4] <https://vuejs.org/>