

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**  
**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**



**BÁO CÁO ĐỒ ÁN MÔN HỌC LẬP TRÌNH TRỰC QUAN**  
**TÊN ĐỀ TÀI**  
**QUẢN LÝ CHUỖI CỬA HÀNG THỜI TRANG**

**Giảng viên hướng dẫn:** ThS. LƯU TOÀN ĐỊNH

**Sinh viên thực hiện** MSSV

Nguyễn Nguyễn Huy 6351071030

Nguyễn Minh Nhật 6351071052

TP. Hồ Chí Minh, năm 2025

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**  
**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**  
**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN MÔN HỌC LẬP TRÌNH TRỰC QUAN**  
**TÊN ĐỀ TÀI**  
**QUẢN LÝ CHUỖI CỬA HÀNG THỜI TRANG**

**Giảng viên hướng dẫn:** ThS. LƯU TOÀN ĐỊNH

**Sinh viên thực hiện** MSSV

Nguyễn Nguyễn Huy 6351071030

Nguyễn Minh Nhật 6351071052

TP. Hồ Chí Minh, năm 2025

## LỜI CẢM ƠN

Qua thời gian học tập và rèn luyện tại trường Trường Đại học Giao thông Vận tải phân hiệu tại TP HCM, đến nay chúng em đã được trang bị những kỹ năng, kiến thức để có thể hoàn thành được bài tập lớn do giảng viên giao.

Cảm ơn tập thể các thầy cô giáo Bộ môn Công Nghệ Thông Tin và các thầy cô thỉnh giảng đã giảng dạy, luôn quan tâm và không ngần ngại dành thời gian để chỉ bày và giải đáp những thắc mắc của chúng em trong những tiết học và cả những lúc ngoài giờ.

Và cảm ơn thầy ThS. Lưu Toàn Định đã luôn quan tâm nhiệt tình hướng dẫn, giúp đỡ chúng em trong quá trình triển khai và thực hiện bài tập lớn. Thầy cũng luôn nhắc nhở, giúp đỡ mỗi khi chúng em gặp khó khăn, nhờ vậy mà chúng em đã hoàn thành bài tập lớn của nhóm mình đúng thời hạn được giao. Nếu không có sự hướng dẫn nhiệt tình của thầy thì có lẽ chúng em đã khó có thể thực hiện được bài tập đúng theo mong muốn của mình.

Nhóm chúng em đã bỏ ra nhiều thời gian để tìm hiểu và trang bị thêm kiến thức nhằm phục vụ cho việc thực hiện ý tưởng, nhưng chắc chắn rằng nhóm chúng em sẽ không thể tránh khỏi những sai sót không đáng có vì kiến thức còn hạn chế. Chúng em hi vọng rằng sẽ nhận được những lời góp ý quý báu của thầy để chúng em có thể hoàn thiện ý tưởng của nhóm một cách tốt nhất có thể.

## This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

**ThS. Lưu Toàn Định**

## BẢN PHÂN CHIA CÔNG VIỆC

Thành viên	Khối lượng công việc	Các phần được phân công	Tự đánh giá	Ký tên
Nguyễn Nguyên Huy MSSV: 6351071030 (Leader)	40%	Chịu trách nhiệm về thiết kế giao diện người dùng với vai trò shop, admin và xử lý logic trong phần admin. Cùng với thành viên thiết kế database.	8/10	
Nguyễn Minh Nhật MSSV: 6351071052 (Member)	60%	Chịu trách nhiệm về thiết kế giao diện người dùng với vai trò user và xử lý logic trong phần shop và admin. Cùng với leader thiết kế cơ sở dữ liệu và tích hợp nâng cao	9/10	

(\*) Trên đây là bảng phân chia công việc, trong quá trình làm bài, mọi thành viên đều hỗ trợ cho nhau để bài tập lớn được hoàn thiện nhất có thể.

# MỤC LỤC

MỤC LỤC .....	vi
DANH MỤC HÌNH ẢNH .....	viii
DANH MỤC VIẾT TẮT .....	ix
CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI .....	1
1.1. Lý do chọn đề tài.....	1
1.2. Mục tiêu, nội dung, phương pháp nghiên cứu đề tài .....	1
1.2.1. Mục tiêu nghiên cứu.....	1
1.2.2. Nội dung nghiên cứu .....	2
1.2.3. Phương pháp nghiên cứu đề tài.....	2
1.2. Đối tượng và phạm vi nghiên cứu .....	3
1.2.1 Đối tượng nghiên cứu.....	3
1.2.2 Phạm vi nghiên cứu.....	4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	5
2.1. Lý thuyết công nghệ WPF .....	5
2.2. Tổng quan về Guna UI 2 .....	10
2.3. Tổng quan về SQL Server .....	11
2.4. Một số lý thuyết công nghệ khác .....	12
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG .....	17
3.1. Mô tả bài toán .....	17
3.1.1. Quản lý sản phẩm.....	17
3.1.2. Quản lý khách hàng.....	17
3.1.3. Quản lý đơn hàng .....	18
3.1.4. Đánh giá và phản hồi sản phẩm .....	18
3.1.5. Giao diện thân thiện và hiệu quả.....	18
3.2. Biểu đồ phân rã chức năng (BFD) .....	19
3.3. Biểu đồ luồng dữ liệu (DFD).....	20
3.3.1. Biểu đồ mức ngữ cảnh.....	20
3.3.2. Mức đỉnh (mức 0) .....	21
3.3.3. Mức dưới đỉnh (mức 1) .....	21
3.4. Mô hình thực thể - liên kết ER (Entity – Relationship Model).....	23
3.4.1. Sơ đồ ERD .....	23
3.4.2. Các thực thể và thuộc tính .....	26
CHƯƠNG 4: GIAO DIỆN CHƯƠNG TRÌNH .....	35
4.1. Giao diện User .....	35
4.2. Giao diện Shop.....	38

4.3. Giao diện Admin.....	45
KẾT LUẬN VÀ KIẾN NGHỊ.....	49
1. Kết luận .....	49
2. Hạn chế.....	49
3. Kiến nghị .....	49
TÀI LIỆU THAM KHẢO.....	51

## DANH MỤC HÌNH ẢNH

Hình 2.1 Mô tả các cách để tải và hiển thị/phân tích giao diện XAML.....	7
Hình 2.2 Kiến trúc WPF .....	8
Hình 2.3 Code mẫu tổng quan XAML .....	10
Hình 2.4 Guna Framework.....	11
Hình 2.5 SQL Server .....	12
Hình 2.6 Vị trí của Entity Framework trong ứng dụng .....	13
Hình 2.7 LINQ .NET .....	14
Hình 2.8 Newtonsoft.Json.....	15
Hình 2.9 Git and GitHub.....	16
Hình 3.1 Biểu đồ BFD .....	19
Hình 3.2 Biểu đồ mức ngữ cảnh.....	20
Hình 3.3 Biểu đồ DFD mức 0 .....	21
Hình 3.5 Biểu đồ DFD mức 1_Quản lý sản phẩm .....	21
Hình 3.6 Biểu đồ DFD mức 1_Quản lý đơn hàng.....	22
Hình 3.7 Biểu đồ DFD mức 1_Quản lý người dùng .....	22
Hình 3.8 Biểu đồ DFD mức 1_Thống kê, báo cáo.....	23
Hình 3.9 Biểu đồ DFD mức 1_Cài đặt ứng dụng.....	23
Hình 3.10 Sơ đồ ERD .....	25
Hình 4.1 Giao diện Login .....	35
Hình 4.2 Giao diện Sign up.....	35
Hình 4.3 Giao diện chính.....	36
Hình 4.4 Thông tin User .....	36
Hình 4. 5 Giao diện tất cả sản phẩm.....	37
Hình 4.6 Giao diện đơn đặt hàng.....	37
Hình 4.7 Giao diện trang chủ .....	38
Hình 4.8 Giao diện thông tin Shop.....	38
Hình 4.9 Giao diện Shop Bag .....	39
Hình 4. 10 Giao diện thanh toán.....	39
Hình 4.11 Giao diện giỏ hàng .....	40
Hình 4.12 Giao diện trạng thái đơn hàng.....	40
Hình 4.13 Giao diện hiển thị sản phẩm .....	41
Hình 4.14 Giao diện đơn hàng .....	41
Hình 4.15 Giao diện quản lý thông tin sản phẩm.....	42
Hình 4. 16 Giao diện thêm sản phẩm .....	42
Hình 4.17 Giao diện sửa sản phẩm.....	43
Hình 4.18 Giao diện Promotation.....	43
Hình 4.19 Giao diện thông tin Promo.....	44
Hình 4.20 Giao diện Rating .....	44
Hình 4.21 Giao diện thống kê .....	45
Hình 4.22 Giao diện thông tin User.....	45
Hình 4.23 Giao diện thông tin Shop.....	46
Hình 4.24 Giao diện thông tin sản phẩm .....	46
Hình 4.25 Giao diện thông tin quảng cáo .....	47
Hình 4.26 Giao diện thông tin loại sản phẩm .....	47
Hình 4.27 Giao diện thông tin thương hiệu .....	48
Hình 4.28 Giao diện thông tin Admin .....	48



## DANH MỤC VIẾT TẮT

STT	Viết tắt	Diễn giải	Ý nghĩa
1	WPF	Windows Presentation Foundation	là framework của Microsoft để xây dựng ứng dụng giao diện người dùng (UI) trên Windows, sử dụng XAML và công nghệ đồ họa hiện đại. Nó giúp tách biệt thiết kế giao diện và logic ứng dụng.
2	XAML	Extensible Application Markup Language	là một ngôn ngữ đánh dấu mở rộng, được sử dụng để mô tả và tạo ra các giao diện người dùng trong các ứng dụng Windows.

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

## 1.1. Lý do chọn đề tài

*Nhu cầu thực tế:* Trong bối cảnh ngành thời trang ngày càng phát triển, việc quản lý kho hàng, sản phẩm, và hóa đơn trở thành một thách thức lớn đối với các cửa hàng quần áo và giày dép. Đề tài này giải quyết được nhu cầu thực tế của các doanh nghiệp trong việc tự động hóa và tối ưu hóa quy trình kinh doanh, đồng thời nâng cao hiệu quả quản lý.

*Ứng dụng công nghệ trong thương mại bán lẻ:* Thương mại bán lẻ hiện đại yêu cầu sự hỗ trợ mạnh mẽ từ các công cụ công nghệ. Việc phát triển một hệ thống quản lý thông minh không chỉ cải thiện năng suất mà còn giúp các cửa hàng thời trang bắt kịp xu hướng số hóa và cạnh tranh trên thị trường.

*Phát triển kỹ năng chuyên môn:* Đề tài này là cơ hội để áp dụng các kiến thức về lập trình, cơ sở dữ liệu, và phân tích hệ thống. Đồng thời, nó giúp phát triển kỹ năng thiết kế giao diện người dùng, xây dựng hệ thống quản lý và xử lý dữ liệu lớn.

*Khả năng phân tích và hỗ trợ ra quyết định:* Hệ thống không chỉ dừng lại ở việc quản lý kho hàng mà còn cung cấp các công cụ phân tích như doanh số bán hàng, hiệu quả của chương trình khuyến mãi, và xu hướng mua sắm của khách hàng. Điều này giúp chủ cửa hàng đưa ra các quyết định kinh doanh chiến lược.

*Ý nghĩa xã hội và kinh tế:* Hệ thống giúp các doanh nghiệp vừa và nhỏ cải thiện khả năng quản lý và cạnh tranh, qua đó đóng góp vào sự phát triển của nền kinh tế. Ngoài ra, nó tạo điều kiện để khách hàng nhận được sản phẩm và dịch vụ tốt hơn.

## 1.2. Mục tiêu, nội dung, phương pháp nghiên cứu đề tài

### 1.2.1. Mục tiêu nghiên cứu

*Xây dựng hệ thống quản lý hiệu quả:* Phát triển một hệ thống quản lý hiện đại, chuyên nghiệp cho cửa hàng thời trang (bao gồm quần áo và giày dép), giúp tự động hóa các quy trình từ quản lý kho hàng, sản phẩm đến xử lý hóa đơn và đơn hàng.

*Tăng cường khả năng kiểm soát hàng hóa:* Đảm bảo việc theo dõi chính xác số lượng tồn kho, thông tin sản phẩm (kích cỡ, màu sắc, giá cả) và giảm thiểu tình trạng thất thoát hàng hóa.

*Hỗ trợ quyết định kinh doanh:* Tích hợp công cụ báo cáo và phân tích dữ liệu, giúp quản lý cửa hàng theo dõi doanh số, hiệu quả của chương trình khuyến mãi, và xu hướng mua sắm của khách hàng để đưa ra các chiến lược kinh doanh phù hợp.

*Nâng cao trải nghiệm người dùng:* Tạo ra giao diện thân thiện, dễ sử dụng, hỗ trợ các tính năng như tìm kiếm, lọc sản phẩm theo danh mục, quản lý giỏ hàng, và kiểm tra lịch sử mua hàng.

*Thúc đẩy chuyển đổi số trong ngành bán lẻ:* Hỗ trợ cửa hàng thời trang hiện đại hóa quy trình kinh doanh, giảm phụ thuộc vào các phương pháp quản lý thủ công, góp phần thúc đẩy sự chuyên nghiệp hóa trong ngành bán lẻ.

### 1.2.2. Nội dung nghiên cứu

Dự án **WPFCommerceApp** tập trung vào nghiên cứu và phát triển một ứng dụng thương mại điện tử dựa trên nền tảng WPF (Windows Presentation Foundation). Mục tiêu chính của dự án là xây dựng một hệ thống quản lý hoàn chỉnh cho các hoạt động kinh doanh trực tuyến, từ quản lý sản phẩm, đơn hàng, đến người dùng và các chức năng thống kê. Đây là một giải pháp hướng tới việc nâng cao trải nghiệm người dùng thông qua giao diện hiện đại, trực quan, được hỗ trợ bởi Material Design.

Nội dung nghiên cứu của dự án bao gồm ba khía cạnh chính: giao diện người dùng (UI), kiến trúc hệ thống, và tích hợp các dịch vụ bên ngoài. Trên khía cạnh UI, dự án khai thác sức mạnh của WPF để tạo ra giao diện động, dễ tùy chỉnh, đồng thời áp dụng các nguyên tắc thiết kế hiện đại nhằm tăng tính thẩm mỹ và hiệu quả sử dụng. Về kiến trúc hệ thống, dự án được thiết kế dựa trên mô hình MVVM (Model-View-ViewModel), đảm bảo tính linh hoạt, dễ mở rộng và bảo trì. Điều này cho phép phân tách rõ ràng giữa giao diện người dùng và logic nghiệp vụ, giúp mã nguồn trở nên dễ quản lý hơn. Ngoài ra, dự án cũng tập trung nghiên cứu việc tích hợp các dịch vụ bên ngoài, chẳng hạn như các cổng thanh toán hoặc API của bên thứ ba, để mở rộng chức năng và đáp ứng tốt hơn nhu cầu thực tế.

Dự án không chỉ phục vụ mục đích học thuật mà còn có tiềm năng ứng dụng thực tiễn cao, cung cấp một nền tảng hiệu quả để các doanh nghiệp nhỏ và vừa triển khai hoạt động thương mại điện tử.

### 1.2.3. Phương pháp nghiên cứu đề tài

Dự án **WPFCommerceApp** áp dụng phương pháp nghiên cứu kết hợp giữa lý thuyết và thực nghiệm nhằm đảm bảo tính chính xác và khả năng ứng dụng cao. Phương pháp này được triển khai qua các bước cụ thể như sau:

*Nghiên cứu lý thuyết:* Đầu tiên, nhóm nghiên cứu tập trung thu thập và phân tích tài liệu liên quan đến các công nghệ và mô hình phát triển phần mềm. Điều này bao gồm việc nghiên cứu về WPF (Windows Presentation Foundation) và mô hình MVVM (Model-View-ViewModel), các nguyên tắc thiết kế giao diện người dùng hiện đại như Material Design, cũng như các phương pháp tổ chức mã nguồn theo hướng module hóa. Đồng thời, các khái niệm về quản lý cơ sở dữ liệu, tích hợp API, và bảo mật trong ứng dụng thương mại điện tử cũng được xem xét kỹ lưỡng.

*Phân tích yêu cầu:* Nhóm nghiên cứu tiến hành thu thập và phân tích các yêu cầu hệ thống từ người dùng cuối, bao gồm cả khách hàng và quản trị viên. Các yêu cầu được chuyển đổi thành các chức năng cụ thể, như quản lý sản phẩm, đơn hàng, người dùng, và thống kê. Bên cạnh đó, các yếu tố phi chức năng như hiệu năng, bảo mật, và trải nghiệm người dùng cũng được xem xét.

*Thiết kế hệ thống:* Dựa trên yêu cầu thu thập được, nhóm thiết kế mô hình hệ thống với trọng tâm là kiến trúc MVVM. Các thành phần chính như Model (quản lý dữ liệu), View (giao diện người dùng), và ViewModel (xử lý logic) được xác định rõ ràng. Bên cạnh đó, biểu đồ ngữ cảnh, biểu đồ phân rã chức năng (BFD), và sơ đồ luồng dữ liệu (DFD) được sử dụng để minh họa cấu trúc và luồng thông tin của hệ thống.

*Xây dựng và kiểm thử:* Dự án được triển khai trên nền tảng WPF với việc sử dụng C# và XAML. Các thư viện hỗ trợ như Material Design Toolkit và FontAwesome được tích hợp để cải thiện giao diện và chức năng. Trong quá trình phát triển, từng module được kiểm thử độc lập bằng Unit Test nhằm đảm bảo hoạt động chính xác và tránh lỗi. Các phương pháp kiểm thử hộp đen và kiểm thử hộp trắng cũng được áp dụng để kiểm tra toàn diện.

*Đánh giá và cải tiến:* Sau khi hoàn thiện, dự án được chạy thử nghiệm với các tình huống thực tế để thu thập phản hồi từ người dùng. Từ đó, nhóm thực hiện các cải tiến nhằm tối ưu hóa hiệu năng và giao diện, cũng như khắc phục bất kỳ vấn đề nào phát sinh.

*Tích hợp và tài liệu hóa:* Cuối cùng, toàn bộ các thành phần của hệ thống được tích hợp và tài liệu hóa một cách chi tiết để hỗ trợ bảo trì và mở rộng trong tương lai.

## **1.2. Đối tượng và phạm vi nghiên cứu**

### **1.2.1 Đối tượng nghiên cứu**

Đối tượng nghiên cứu của dự án WPF EcommerceApp là hệ thống quản lý ứng dụng thương mại điện tử dựa trên nền tảng WPF (Windows Presentation Foundation), với mục tiêu phát triển một giải pháp toàn diện cho các doanh nghiệp nhỏ và vừa trong lĩnh vực bán hàng trực tuyến. Hệ thống này nhằm cải thiện và tối ưu hóa các quy trình kinh doanh qua môi trường số, bao gồm các chức năng quản lý sản phẩm, đơn hàng, người dùng và thống kê. Đặc biệt, hệ thống được thiết kế để phục vụ nhu cầu của người quản lý, người bán và người mua trong một môi trường giao dịch trực tuyến, với giao diện người dùng hiện đại và dễ sử dụng.

Một yếu tố quan trọng trong đối tượng nghiên cứu là cách thức xây dựng và tối ưu hóa trải nghiệm người dùng (UI), đặc biệt là việc sử dụng WPF để tạo ra các giao diện có tính tương tác cao và dễ tùy chỉnh. Các thành phần giao diện này phải đáp ứng tiêu chí không chỉ về thẩm mỹ mà còn về hiệu quả sử dụng, giúp người dùng dễ dàng thao tác và tìm kiếm thông

tin. Cùng với đó, dự án cũng nghiên cứu và áp dụng các nguyên tắc thiết kế hiện đại, chẳng hạn như Material Design, để đảm bảo tính trực quan và thuận tiện cho người sử dụng.

Bên cạnh đó, đối tượng nghiên cứu còn bao gồm việc xây dựng hệ thống kiến trúc vững chắc, sử dụng mô hình MVVM (Model-View-ViewModel) để phân tách rõ ràng giữa giao diện người dùng và logic nghiệp vụ. Điều này không chỉ giúp mã nguồn dễ bảo trì và mở rộng mà còn hỗ trợ việc tích hợp với các dịch vụ bên ngoài, chẳng hạn như các cổng thanh toán, API bên thứ ba, nhằm nâng cao tính năng và tính linh hoạt của hệ thống. Nhờ vào việc nghiên cứu và triển khai các giải pháp này, dự án có thể cung cấp một công cụ hữu ích giúp các doanh nghiệp nhỏ và vừa cải thiện hiệu quả kinh doanh, từ đó thúc đẩy sự phát triển của các hoạt động thương mại điện tử trong bối cảnh hiện nay.

### **1.2.2 Phạm vi nghiên cứu**

Phạm vi nghiên cứu của dự án WPFCommerceApp tập trung vào việc xây dựng một hệ thống ứng dụng thương mại điện tử dựa trên nền tảng WPF, từ đó phát triển một giải pháp quản lý toàn diện cho các hoạt động thương mại điện tử. Nghiên cứu này bao gồm ba khía cạnh chính: giao diện người dùng (UI), kiến trúc hệ thống và tích hợp các dịch vụ bên ngoài. Cụ thể, dự án sẽ nghiên cứu và phát triển các chức năng cơ bản của một ứng dụng thương mại điện tử như quản lý sản phẩm, đơn hàng, người dùng, và các thông kê liên quan. Hệ thống này không chỉ phục vụ mục đích học thuật mà còn có khả năng áp dụng thực tế trong các doanh nghiệp nhỏ và vừa, mang đến một giải pháp thực tiễn cho các nhu cầu thương mại điện tử hiện nay.

Về mặt giao diện người dùng, phạm vi nghiên cứu sẽ đi sâu vào việc ứng dụng WPF và các nguyên tắc thiết kế hiện đại như Material Design để tạo ra các giao diện động và dễ tùy chỉnh, từ đó nâng cao trải nghiệm người dùng. Đồng thời, nghiên cứu cũng sẽ đề cập đến việc xây dựng một hệ thống kiến trúc sử dụng mô hình MVVM, giúp phân tách rõ ràng giữa các tầng của ứng dụng, đảm bảo tính dễ bảo trì, dễ mở rộng và linh hoạt trong việc phát triển các tính năng mới.

Ngoài ra, phạm vi nghiên cứu cũng bao gồm việc tích hợp các dịch vụ bên ngoài vào hệ thống, chẳng hạn như các cổng thanh toán trực tuyến, dịch vụ vận chuyển, hoặc API của các bên thứ ba. Việc tích hợp này sẽ giúp hệ thống mở rộng các tính năng hỗ trợ hoạt động kinh doanh trực tuyến, đáp ứng yêu cầu thực tế của người dùng và nâng cao tính cạnh tranh của ứng dụng trên thị trường. Cuối cùng, dự án sẽ nghiên cứu các vấn đề bảo mật và tối ưu hiệu suất của hệ thống để đảm bảo tính ổn định và bảo mật cao trong quá trình sử dụng.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1. Lý thuyết công nghệ WPF

“Cùng với Windows 1.0 vào năm 1985, Microsoft đã giới thiệu Graphics Device Interface (GDI) và hệ thống con USER để xây dựng giao diện người dùng đồ họa (GUI) dựa trên Windows. Vào năm 1990, OpenGL đã được giới thiệu để tạo đồ họa 2D và 3D trên các hệ thống Windows và không phải Windows. Vào năm 1995, Microsoft đã trình bày một công nghệ khác gọi là DirectX để tạo đồ họa 2D/3D hiệu suất cao. Sau đó, GDI+ được giới thiệu để thêm hỗ trợ trộn alpha và cộ gradient trên cơ sở GDI hiện có.

Vào năm 2002, Microsoft đã giới thiệu .NET Framework. Cùng với đó, Windows Forms cũng được giới thiệu để xây dựng giao diện người dùng (UI) cho Windows sử dụng ngôn ngữ C# và Visual Basic. Nó được xây dựng trên cơ sở GDI+, và do đó, nó vẫn còn những hạn chế của GDI+ và hệ thống con USER.

Qua nhiều năm, Microsoft đã quyết định đưa ra một công nghệ mới để xây dựng giao diện người dùng giàu tính năng cho các ứng dụng dựa trên Windows, không chỉ giúp các nhà phát triển và thiết kế thoát khỏi những hạn chế của GDI/GDI+ và hệ thống con USER, mà còn giúp họ cải thiện năng suất khi xây dựng các ứng dụng dựa trên máy tính để bàn.

Vào tháng 11 năm 2006, cùng với .NET 3.0, Windows Presentation Foundation (WPF) đã được giới thiệu để cung cấp cho các nhà phát triển một mô hình lập trình thống nhất để xây dựng các ứng dụng máy tính để bàn động, dữ liệu dẫn cho Windows. Nó đi kèm với một tập hợp rộng các tính năng để tạo một hệ thống con đồ họa để hiển thị giao diện người dùng giàu tính năng sử dụng các điều khiển, bố cục, đồ họa, tài nguyên và nhiều hơn nữa, xem xét ứng dụng và bảo mật dữ liệu. Khi nó được vận chuyển lần đầu tiên như một phần của .NET Framework 3.0, phiên bản đầu tiên được gọi là WPF 3.0.

WPF là một khung khổ độc lập với độ phân giải sử dụng một động cơ kết xuất dựa trên vector sử dụng một ngôn ngữ dựa trên XML gọi là XAML (được phát âm là Zammel), để tạo ra những trải nghiệm người dùng hiện đại cung cấp một mô hình khai báo cho lập trình ứng dụng. Sử dụng điều này, bạn có thể dễ dàng tùy chỉnh các điều khiển và thêm vỏ bọc vào đó để có được một đại diện tốt hơn về giao diện người dùng của ứng dụng.

Vì WPF khác so với Windows Forms cổ điển, vì nó sử dụng XAML, ràng buộc dữ liệu, mẫu, phong cách, hoạt ảnh, tài liệu và nhiều hơn nữa, ban đầu nó nhận được ít sự chú ý. Tuy nhiên, sau đó nó bắt đầu thu hút được rất nhiều sự quan tâm và phổ biến. Nhiều phiên bản cập nhật đã được phát hành để thêm nhiều chức năng hơn vào đó để làm cho nó mạnh mẽ và mạnh mẽ.

Trong cuốn sách này, chúng tôi sẽ bao gồm một tập hợp các công thức sẽ chỉ cho bạn cách thực hiện các nhiệm vụ chung bằng cách sử dụng WPF. Bắt đầu với các nguyên tắc cơ bản của WPF, chúng tôi sẽ bao gồm các điều khiển tiêu chuẩn, bố cục, bảng điều khiển, ràng

buộc dữ liệu, điều khiển tùy chỉnh, điều khiển người dùng, phong cách, mẫu, kích hoạt và hoạt ảnh và sau đó chuyển sang việc sử dụng tài nguyên, mẫu MVVM, dịch vụ WCF, gỡ lỗi, luồng và khả năng tương tác của WPF để đảm bảo bạn hiểu rõ nền tảng.

Các ví dụ được đưa ra trong cuốn sách này đơn giản, dễ hiểu và cung cấp cho bạn những gì bạn cần để học và làm chủ kỹ năng cần thiết để xây dựng các ứng dụng máy tính để bàn sử dụng WPF. Khi bạn đến cuối cuốn sách này, bạn sẽ đủ trình độ với kiến thức sâu rộng về mỗi chương mà nó bao gồm. Mặc dù cuốn sách này đã bao gồm hầu hết các chủ đề quan trọng, nhưng vẫn sẽ có một số chủ đề mà không có cuốn sách nào có thể bao quát hoàn toàn. Bạn chắc chắn sẽ thích đọc cuốn sách này vì có rất nhiều bước đồ họa và văn bản để giúp bạn tự tin khi làm việc với Windows Presentation Foundation.” - Trích *Windows Presentation Foundation Development Cookbook* author Kunal Chowdhury, Production reference: 12/10/218

Windows Presentation Foundation (WPF) là một khung khổ lập trình mạnh mẽ được Microsoft giới thiệu vào năm 2006, cùng với .NET Framework 3.0. WPF được thiết kế để giúp các nhà phát triển và thiết kế tạo ra các ứng dụng máy tính để bàn có giao diện người dùng giàu tính năng và hiện đại, vượt qua những hạn chế của các công nghệ trước đó như GDI và GDI+.

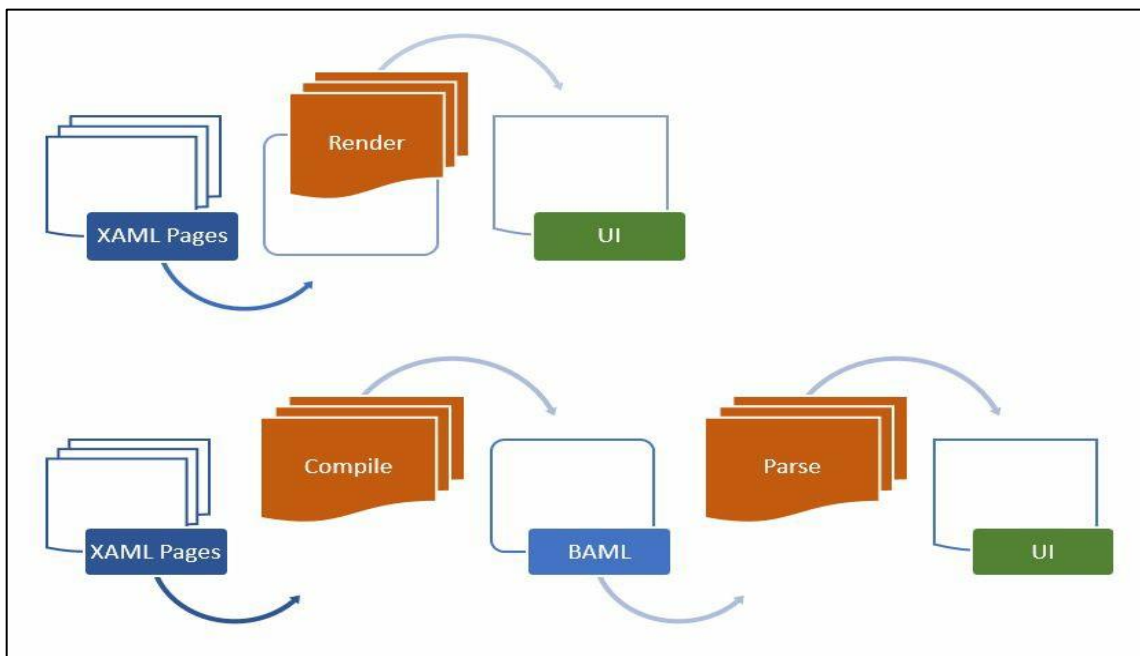
Một trong những điểm nổi bật của WPF là việc sử dụng ngôn ngữ XML gọi là XAML (eXtensible Application Markup Language). XAML cho phép tạo ra các giao diện người dùng một cách khai báo, giúp tách biệt rõ ràng giữa logic ứng dụng và thiết kế giao diện. Điều này không chỉ làm cho quá trình phát triển trở nên dễ dàng hơn mà còn cho phép các nhà thiết kế và nhà phát triển làm việc độc lập hiệu quả. Các nhà thiết kế có thể tập trung vào việc tạo ra giao diện người dùng đẹp mắt và chức năng mà không cần phải biết chi tiết về mã code, trong khi các nhà phát triển có thể tập trung vào logic ứng dụng mà không cần phải lo lắng về thiết kế.

WPF cung cấp một tập hợp rộng các tính năng để tạo ra các giao diện người dùng động và dữ liệu dẫn. Nó bao gồm các điều khiển, bố cục, đồ họa, tài nguyên và nhiều hơn nữa. Các điều khiển trong WPF có thể được tùy chỉnh dễ dàng bằng cách sử dụng phong cách, mẫu và kích hoạt, giúp tạo ra một trải nghiệm người dùng nhất quán và hấp dẫn. Ví dụ, bạn có thể tạo ra các nút, hộp văn bản, danh sách và nhiều loại điều khiển khác chỉ bằng cách viết một vài dòng mã XAML.

Ràng buộc dữ liệu là một tính năng quan trọng khác của WPF. Nó cho phép cập nhật tự động giữa giao diện người dùng và mô hình dữ liệu, giúp giảm thiểu số lượng mã code cần thiết để quản lý dữ liệu. Điều này không chỉ làm cho ứng dụng trở nên linh hoạt hơn mà còn giúp giảm thiểu lỗi do cập nhật thủ công.

WPF cũng hỗ trợ hoạt ảnh và hiệu ứng hình ảnh phức tạp, giúp nâng cao tính tương tác và thẩm mỹ của ứng dụng. Bạn có thể tạo ra các hiệu ứng chuyển tiếp mượt mà, hoạt ảnh phức tạp và nhiều hiệu ứng khác chỉ bằng cách sử dụng XAML hoặc mã code C# hoặc VB.NET.

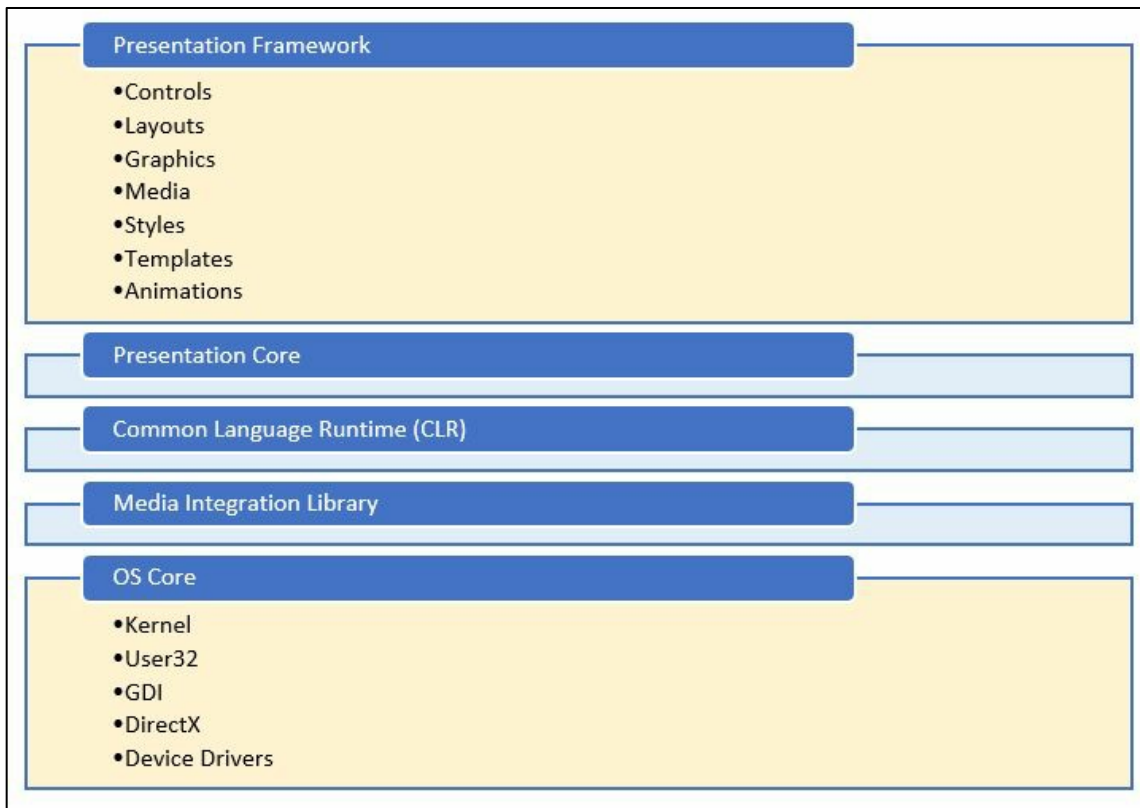
Ngoài ra, WPF tích hợp tốt với các công nghệ khác trong .NET Framework, như WCF (Windows Communication Foundation) cho dịch vụ web và MVVM (Model-View-ViewModel) cho kiến trúc ứng dụng. Kiến trúc MVVM giúp tách biệt rõ ràng giữa mô hình dữ liệu, logic ứng dụng và giao diện người dùng, làm cho ứng dụng trở nên sạch sẽ và dễ bảo trì hơn.



Hình 2.1 Mô tả các cách để tải và hiển thị/phân tích giao diện XAML

“WPF sử dụng một kiến trúc phân lớp bao gồm các API quản lý, không quản lý và lõi, được chia thành năm lớp khác nhau, bao gồm: **Presentation Framework**, **Presentation Core**, **Common Language Runtime**, **Media Integration Library**, và **OS Core**. Mô hình lập trình được cung cấp thông qua mã quản lý.





Hình 2.2 Kiến trúc WPF

Khung giao diện trình bày (**Presentation Framework**), là một phần của tệp **presentationframework.dll**, cung cấp các thành phần cơ bản cần thiết (chẳng hạn như các điều khiển, bố cục, đồ họa, phương tiện, kiểu dáng, mẫu, hoạt ảnh, và nhiều hơn nữa) để bắt đầu xây dựng giao diện người dùng (UI) cho các ứng dụng WPF của bạn. Đây là một phần của lớp quản lý.

Lớp **Presentation Core**, là một phần của tệp **presentationcore.dll**, cung cấp lớp bao bọc xung quanh **Media Integration Library (MIL)**. Nó cung cấp các giao diện công khai để truy cập vào **MIL Core** và **Hệ thống Visual** nhằm phát triển **Cây Visual (Visual Tree)**. Lớp này chứa các thành phần hình ảnh và các hướng dẫn kết xuất để xây dựng ứng dụng Windows bằng các công cụ **XAML**. Đây cũng là một phần của mã quản lý.

**Common Language Runtime (CLR)**, thường được gọi là CLR và là một phần của lớp quản lý, cung cấp nhiều tính năng giúp xây dựng các ứng dụng mạnh mẽ, bao gồm **Hệ thống kiểu chung (CTS)**, xử lý lỗi, quản lý bộ nhớ, và nhiều hơn nữa.

**Media Integration Library (MIL)**, nằm trong tệp **milcore.dll**, thuộc lớp không quản lý, được sử dụng để hiển thị tất cả đồ họa thông qua công cụ **DirectX**. MIL cung cấp hỗ trợ cơ bản cho các bề mặt 2D và 3D, cho phép truy cập các thành phần không quản lý để tích

hợp chặt chẽ với DirectX. Nó cũng cho phép cải thiện hiệu suất khi kết xuất các hướng dẫn từ **Hệ thống Visual (Visual System)** đến **Common Language Runtime (CLR)**.

Ngay sau MIL là lớp **OS Core**, cung cấp quyền truy cập vào các API cấp thấp để xử lý các thành phần cốt lõi của hệ điều hành. Lớp này bao gồm **Kernel**, **User32**, **DirectX**, **GDI**, và các trình điều khiển thiết bị (device drivers).” - Trích *Windows Presentation Foundation Development Cookbook* author Kunal Chowdhury, Production reference: 12/10/218

Windows Presentation Foundation (WPF) là một nền tảng được Microsoft phát triển nhằm hỗ trợ việc xây dựng các ứng dụng giao diện người dùng (UI) một cách linh hoạt, hiện đại và tỷ chỉnh. WPF sử dụng kiến trúc phân lớp, bao gồm các API được quản lý, không quản lý và các thành phần lõi. Kiến trúc này được chia thành năm lớp chính: Presentation Framework, Presentation Core, Common Language Runtime (CLR), Media Integration Library (MIL), và OS Core.

**Presentation Framework:** Lớp Presentation Framework, nằm trong tệp **presentationframework.dll**, cung cấp các thành phần cơ bản để bắt đầu xây dựng giao diện người dùng cho các ứng dụng WPF. Các thành phần này bao gồm các điều khiển, bố cục, đồ họa, định dạng, mẫu, hoạt động, và nhiều tính năng khác. Presentation Framework là phần của lớp quản lý (managed layer) trong WPF.

**Presentation Core:** Presentation Core, thuộc tệp **presentationcore.dll**, cung cấp một lớp bao bọc cho **Media Integration Library (MIL)**. Nó cung cấp các giao diện công khai để truy cập MIL Core và Hệ thống Visual, cho phép phát triển **Cây Visual (Visual Tree)**. Lớp này chứa các phần tử hình ảnh và hướng dẫn kết xuất, giúp xây dựng các ứng dụng Windows bằng các công cụ XAML. Giống như Presentation Framework, Presentation Core cũng thuộc lớp quản lý.

**Common Language Runtime (CLR):** CLR, hay **Common Language Runtime**, là một phần quan trọng trong lớp quản lý của WPF. Nó cung cấp nhiều tính năng hỗ trợ việc xây dựng các ứng dụng mạnh mẽ. Những tính năng này bao gồm **Hệ thống kiểu chung (CTS)**, quản lý bộ nhớ, xử lý lỗi và nhiều chức năng khác.

**Media Integration Library (MIL):** MIL, nằm trong tệp **milcore.dll**, thuộc lớp không quản lý. MIL được sử dụng để hiển thị tất cả đồ họa được kết xuất thông qua **DirectX**. Nó cung cấp hỗ trợ cơ bản cho các bề mặt 2D và 3D, đồng thời cho phép truy cập các thành phần không quản lý nhằm tích hợp chặt chẽ với DirectX. Nhờ MIL, WPF có thể tăng cường hiệu suất khi kết xuất các hướng dẫn từ Hệ thống Visual đến CLR.

**OS Core:** Ngay sau MIL là lớp **OS Core**, cung cấp quyền truy cập đến các API cấp thấp để xử lý các thành phần lõi của hệ điều hành. Lớp này bao gồm **Kernel**, **User32**, **DirectX**, **GDI**, và các trình điều khiển thiết bị. Nó cung cấp nền tảng cho các lớp cao hơn trong kiến trúc WPF.

Mặc dù không bắt buộc phải sử dụng XAML để tạo giao diện người dùng (UI), nhưng XAML đã được công nhận rộng rãi là một lựa chọn thông minh để tạo toàn bộ giao diện ứng dụng, vì nó làm cho việc tạo UI trở nên dễ dàng hơn. Bạn cũng có thể tạo UI bằng cách viết mã C# hoặc VB.NET, nhưng điều đó khiến công việc trở nên khó khăn hơn và khó bảo trì hơn. Ngoài ra, việc này cũng gây khó khăn cho các nhà thiết kế khi làm việc độc lập.

Thiết kế giao diện ứng dụng bằng XAML dễ dàng như việc viết một nút XML với một vài thuộc tính tùy chọn. Các thuộc tính được sử dụng để thiết lập kiểu dáng, hành vi và các thuộc tính bổ sung. Để tạo một nút bấm đơn giản trong giao diện, bạn chỉ cần viết `<Button />` trong tệp XAML của mình. Tương tự, bạn có thể viết `<TextBox />` để tạo một hộp nhập liệu cho người dùng.

Ngoài ra, bạn có thể thêm nhiều chi tiết hơn cho các điều khiển. Ví dụ, để thêm nhãn (label) vào một nút bấm, bạn sử dụng thuộc tính **Content**, và để đặt kích thước, bạn sử dụng các thuộc tính **Height** và **Width**, như trong đoạn mã sau:

```
<Button Content="Click Here" />
<Button Height="36" Width="120" />
```

Hình 2.3 Code mẫu tổng quan XAML

## 2.2. Tổng quan về Guna UI 2

**Guna UI 2** là một framework giao diện người dùng (UI Framework) được thiết kế để phát triển các ứng dụng .NET WinForms với giao diện hiện đại, đẹp mắt và chuyên nghiệp. Đây là phiên bản nâng cấp từ Guna UI Framework, với nhiều cải tiến về tính năng, hiệu suất và khả năng tùy chỉnh, giúp lập trình viên xây dựng các ứng dụng WinForms nhanh chóng và hiệu quả hơn. Một trong những điểm nổi bật nhất của Guna UI 2 là việc áp dụng nguyên tắc thiết kế Material Design, mang lại giao diện trực quan và nhất quán, đồng thời giúp ứng dụng có vẻ ngoài hiện đại và phù hợp với xu hướng công nghệ hiện nay. Điều này đặc biệt hữu ích khi xây dựng các ứng dụng dành cho doanh nghiệp hoặc khách hàng yêu cầu giao diện cao cấp, thân thiện với người dùng.

Guna UI 2 cung cấp hơn 30 thành phần giao diện (controls) phong phú như Button, TextBox, ComboBox, ProgressBar, Chart, DataGridView,... Tất cả đều được thiết kế với tính năng nâng cao, cho phép tích hợp nhanh các hiệu ứng như hiệu ứng hover, bo góc, gradient màu, hiệu ứng chuyển đổi mượt mà và nhiều hoạt ảnh hấp dẫn khác. Các thành phần này không chỉ giúp cải thiện giao diện mà còn tối ưu hóa trải nghiệm người dùng, nhờ sự hỗ trợ mạnh mẽ trong việc tùy chỉnh giao diện. Các lập trình viên có thể điều chỉnh mọi yếu tố về màu sắc, kích thước, kiểu dáng và hiệu ứng động của các thành phần trực tiếp trong mã hoặc thông qua giao diện kéo thả tích hợp trong Visual Studio. Điều này

không chỉ tăng tốc độ phát triển mà còn giảm bớt khối lượng công việc cho các nhóm thiết kế giao diện.



Hình 2.4 Guna Framework

Một ưu điểm quan trọng khác của Guna UI 2 là khả năng tối ưu hóa hiệu suất. Framework này được thiết kế để sử dụng tài nguyên hệ thống một cách hiệu quả, đảm bảo các ứng dụng hoạt động mượt mà ngay cả khi sử dụng nhiều hiệu ứng đồ họa phức tạp. Đối với các ứng dụng cần xử lý nhiều tác vụ hoặc chạy trên hệ thống hạn chế tài nguyên, Guna UI 2 vẫn đáp ứng tốt nhờ cơ chế tối ưu hóa và giảm tải cho bộ xử lý. Ngoài ra, framework còn hỗ trợ các tính năng như thiết kế giao diện đa nền tảng, giúp ứng dụng dễ dàng tương thích với nhiều độ phân giải màn hình khác nhau.

Với sự phát triển mạnh mẽ của ứng dụng desktop hiện đại, Guna UI 2 trở thành một lựa chọn lý tưởng cho các nhà phát triển muốn tạo ra các phần mềm WinForms có tính thẩm mỹ cao mà không cần quá nhiều công sức. Framework này không chỉ mang lại sự linh hoạt trong thiết kế mà còn giúp tiết kiệm thời gian và chi phí phát triển. Bên cạnh đó, cộng đồng người dùng đông đảo và tài liệu hướng dẫn chi tiết của Guna UI 2 là những yếu tố hỗ trợ đắc lực cho lập trình viên, giúp họ dễ dàng tiếp cận và triển khai các tính năng của framework vào dự án thực tế. Nhìn chung, Guna UI 2 là một công cụ mạnh mẽ và toàn diện, giúp các ứng dụng WinForms không chỉ đáp ứng mà còn vượt xa mong đợi về giao diện và trải nghiệm người dùng.

### 2.3. Tổng quan về SQL Server

Hệ quản trị cơ sở dữ liệu Microsoft SQL Server (MSSQL) là một trong những hệ quản trị cơ sở dữ liệu thông dụng nhất hiện nay. Đây là hệ quản trị cơ sở dữ liệu thường được sử dụng trong các hệ thống trung bình, với ưu điểm có các công cụ quản lý mạnh mẽ giúp cho việc quản lý và bảo trì hệ thống dễ dàng, hỗ trợ nhiều phương pháp lưu trữ, phân vùng và đánh chỉ mục phục vụ cho việc tối ưu hóa hiệu năng.

SQL Server luôn được Microsoft cải tiến để nâng cao hiệu năng, tính sẵn sàng của hệ thống, khả năng mở rộng và bảo mật, cung cấp nhiều công cụ cho người phát triển ứng dụng được tích hợp với bộ Visual Studio do Microsoft cung cấp. SQL Server có 4 dịch vụ lớn là Database Engine, Integration Service, Reporting service và Analysis Services. DataBase

Engine: được phát triển để thực thi tốt hơn với việc hỗ trợ cả dữ liệu có cấu trúc và dữ liệu phi cấu trúc (XML).



Hình 2.5 SQL Server

Khả năng sẵn sàng của hệ thống được nâng cao, hỗ trợ các chức năng: Database mirroring (cơ sở dữ liệu gương), failover clustering, snapshots và khôi phục dữ liệu nhanh. Việc quản lý chỉ mục được thực hiện song song với việc hoạt động của hệ thống.

Người dùng có thể thêm chỉ mục, xây dựng lại chỉ mục hay xóa một chỉ mục đi trong khi hệ thống vẫn được sử dụng. Chức năng phân vùng dữ liệu được hỗ trợ: Người dùng có thể phân vùng các bảng và chỉ mục cũng như quản lý phân vùng dữ liệu một cách dễ dàng. Việc hỗ trợ phân vùng dữ liệu giúp nâng cao hiệu năng hoạt động của hệ thống.

Dịch vụ đồng bộ hóa dữ liệu được mở rộng với việc hỗ trợ mô hình đồng bộ hóa ngang hàng. Đây là dịch vụ giúp đồng bộ hóa dữ liệu giữa các máy chủ dữ liệu, giúp mở rộng khả năng của hệ thống.

Dịch vụ tích hợp (Integration Service) thiết kế lại cho phép người dùng tích hợp dữ liệu và phân tích dữ liệu từ nhiều nguồn khác nhau. Hỗ trợ việc quản lý chất lượng dữ liệu và làm sạch dữ liệu, một công việc quan trọng trong tiến trình ETL.

Dịch vụ phân tích dữ liệu (Analysis Service): cung cấp khung nhìn tích hợp và thống nhất về dữ liệu cho người dùng, hỗ trợ việc phân tích dữ liệu.

Công cụ khai phá dữ liệu (Data mining) được tích hợp hỗ trợ nhiều thuật toán khai phá dữ liệu, hỗ trợ cho việc phân tích, khai phá dữ liệu và xây dựng các hệ thống hỗ trợ ra quyết định cho người quản lý.

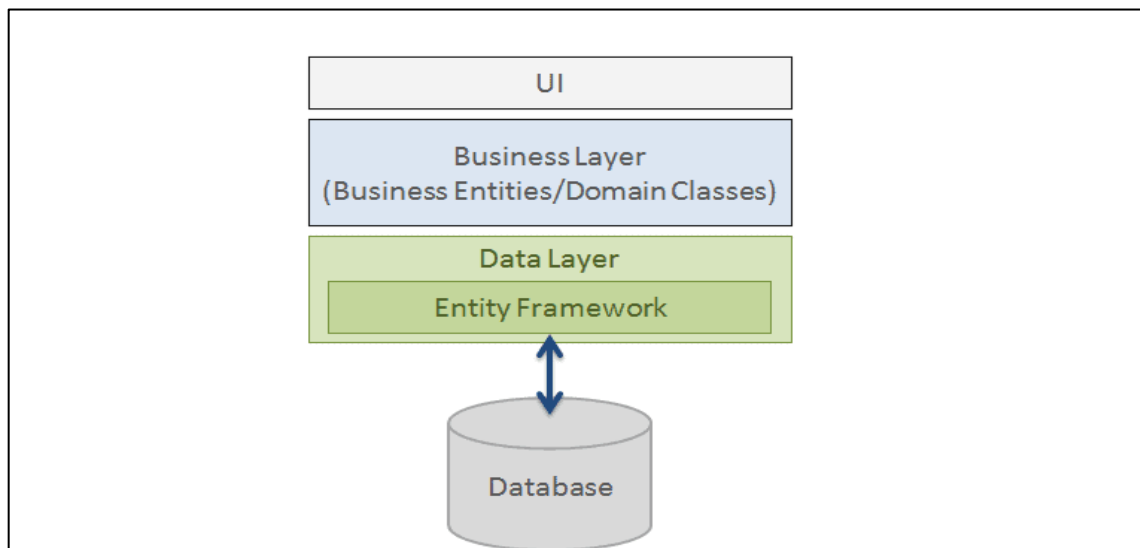
Dịch vụ xây dựng quản lý báo cáo (Reporting Service) dựa trên nền tảng quản trị doanh nghiệp thông minh và được quản lý qua dịch vụ web. Báo cáo có thể được xây dựng dễ dàng với ngôn ngữ truy vấn MDX. Thông qua các công cụ trên Business Intelligent, người dùng dễ dàng truy cập báo cáo và trích xuất ra nhiều định dạng khác nhau thông qua trình duyệt web.

## 2.4. Một số lý thuyết công nghệ khác

*Entity Framework (EF)* là một API Object-Relational Mapping (ORM) được Microsoft cung cấp trong .NET Framework, cho phép các nhà phát triển tương tác với cơ sở dữ liệu một cách dễ dàng và trực quan hơn. EF tự động chuyển đổi các bảng trong cơ sở dữ liệu thành các lớp đối tượng trong ứng dụng, giúp nhà phát triển thao tác với các bản ghi dưới dạng đối tượng thay vì viết truy vấn SQL truyền thống.

EF cung cấp ba cách tiếp cận chính để quản lý dữ liệu: **Database First**, **Model First**, và **Code First**. Database First cho phép tạo lớp tự động dựa trên các bảng của cơ sở dữ liệu đã tồn tại. Model First cho phép tạo các lớp dữ liệu bằng cách thiết kế các mô hình trong designer tool. Code First, một cách tiếp cận hiện đại, cho phép nhà phát triển tạo các lớp đối tượng trong code và tự động sinh ra cơ sở dữ liệu.

Trong dự án này, Entity Framework được sử dụng để thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên cơ sở dữ liệu. EF hỗ trợ nhiều lợi ích như tự động xác thực, quản lý dữ liệu bằng LINQ, và tăng cường bảo mật bằng cách loại bỏ nguy cơ SQL Injection. Tuy nhiên, EF không phải lúc nào cũng hiệu quả cao đối với các truy vấn phức tạp hoặc các tác vụ tùy chỉnh, nên SQL Client được sử dụng bổ sung trong dự án.



Hình 2.6 Vị trí của Entity Framework trong ứng dụng

*SQL Client* là một API chuẩn trong .NET Framework, cho phép kết nối và tương tác với cơ sở dữ liệu quan hệ như SQL Server hoặc SQLite. Nó cung cấp các lớp và phương pháp như **SqlConnection**, **SqlCommand**, và **SqlDataReader** để thực thi các truy vấn SQL một cách hiệu quả.

Trong SQL Client, **SqlConnection** được sử dụng để thiết lập kết nối tới cơ sở dữ liệu. **SqlCommand** cho phép nhà phát triển thực thi các truy vấn SQL hoặc các stored procedures. **SqlDataReader** cung cấp cách truy xuất dữ liệu theo dòng, giúp tối ưu bộ nhớ.

Trong dự án này, SQL Client được sử dụng để xử lý các tác vụ đòi hỏi hiệu suất cao hoặc những tình huống đặc thù không thể thực hiện hiệu quả bằng EF. SQL Client mang lại tính linh hoạt và tốc độ cao nhất khi cần thao tác trực tiếp với SQL.

*LINQ* là một tính năng mạnh mẽ trong .NET, cho phép nhà phát triển thực hiện các truy vấn dữ liệu một cách dễ dàng và ngắn gọn hơn, trực tiếp trên các tập hợp dữ liệu trong bộ nhớ hoặc trên cơ sở dữ liệu. LINQ tích hợp ngôn ngữ truy vấn trực tiếp vào ngôn ngữ lập trình (C#, VB.NET), giúp đồng nhất các cú pháp và giảm thiểu lỗi lập trình.

LINQ hỗ trợ nhiều nhà cung cấp dữ liệu, chẳng hạn như LINQ to SQL, LINQ to Entities (cho Entity Framework), LINQ to XML, LINQ to Objects, và nhiều hơn nữa. Trong LINQ, các truy vấn dữ liệu được viết bằng cú pháp giống SQL nhưng thân thiện hơn với nhà phát triển.



Hình 2.7 LINQ .NET

*Newtonsoft.Json* là một thư viện phổ biến trong hệ sinh thái .NET, được sử dụng để xử lý dữ liệu JSON một cách linh hoạt và hiệu quả. JSON (JavaScript Object Notation) là định dạng dữ liệu nhẹ, dễ đọc và dễ ghi, thường được sử dụng để trao đổi dữ liệu giữa các ứng dụng và dịch vụ. Thư viện *Newtonsoft.Json* cung cấp các phương thức để chuyển đổi dữ liệu giữa các đối tượng .NET và chuỗi JSON, giúp dễ dàng lưu trữ hoặc chia sẻ dữ liệu theo định dạng này. Trong dự án WPF của bạn, *Newtonsoft.Json* được sử dụng để đọc và ghi dữ liệu JSON, đặc biệt trong việc lưu trữ cấu hình hệ thống hoặc xuất dữ liệu dưới dạng tệp JSON để sao lưu. Việc sử dụng thư viện này giúp giảm thiểu việc phải thao tác thủ công với chuỗi JSON, đồng thời đảm bảo tính linh hoạt trong việc xử lý và chuyển đổi dữ liệu.



liệu. Các tính năng như khả năng tự động hóa việc ánh xạ giữa đối tượng .NET và JSON, cùng với các tùy chọn cấu hình linh hoạt, giúp thư viện này trở thành một công cụ mạnh mẽ trong việc quản lý dữ liệu JSON trong ứng dụng.



Hình 2.8 Newtonsoft.Json

*LiveCharts* là một thư viện mạnh mẽ dành cho việc vẽ biểu đồ trong .NET, giúp hiển thị dữ liệu dưới dạng đồ họa trực quan. Thư viện này hỗ trợ nhiều loại biểu đồ khác nhau như biểu đồ cột, biểu đồ tròn, biểu đồ đường và các biểu đồ tương tác khác, giúp người dùng dễ dàng phân tích và trình bày dữ liệu. Trong dự án của bạn, *LiveCharts* được sử dụng để hiển thị các biểu đồ liên quan đến doanh số bán hàng, tình trạng kho hàng, và xu hướng mua sắm của khách hàng. Các biểu đồ này không chỉ giúp người dùng theo dõi các chỉ số kinh doanh một cách trực quan mà còn hỗ trợ việc ra quyết định bằng cách phân tích những xu hướng cụ thể từ dữ liệu. *LiveCharts* cung cấp các tính năng như cập nhật thời gian thực, khả năng tùy chỉnh biểu đồ và các hiệu ứng hoạt hình, giúp các biểu đồ không chỉ chính xác mà còn sinh động và dễ hiểu. Sự tích hợp của thư viện này vào dự án giúp tăng cường tính trực quan của các báo cáo và thống kê, tạo trải nghiệm người dùng tốt hơn.

*Printing API* là một công cụ hữu ích hỗ trợ các ứng dụng in ấn hóa đơn, báo cáo và các tài liệu khác trực tiếp từ phần mềm. Thư viện này cung cấp các lớp và phương thức giúp ứng dụng dễ dàng giao tiếp với các máy in, thực hiện các tác vụ như in văn bản, hình ảnh, hoặc đồ họa lên giấy. Trong dự án WPF của bạn, *Printing API* được tích hợp để hỗ trợ chức năng in hóa đơn hoặc báo cáo trực tiếp từ giao diện ứng dụng. Người dùng có thể tùy chọn



xem trước nội dung trước khi in, điều này đảm bảo rằng các thông tin được in ra chính xác và theo đúng yêu cầu. Việc tích hợp API in giúp tiết kiệm thời gian và chi phí cho doanh nghiệp, đồng thời tăng cường tính chuyên nghiệp của ứng dụng khi có thể cung cấp các báo cáo in ấn ngay lập tức cho người dùng mà không cần phải qua các phần mềm bên ngoài.

*App.config* là một tập tin cấu hình quan trọng trong các ứng dụng .NET, giúp lưu trữ các thông tin cấu hình cần thiết cho ứng dụng. Các thông tin này có thể bao gồm chuỗi kết nối cơ sở dữ liệu, cài đặt hệ thống, tham số mặc định, và các giá trị cấu hình khác mà ứng dụng cần để hoạt động đúng đắn. Trong dự án WPF của bạn, *App.config* được sử dụng để lưu trữ các cài đặt hệ thống quan trọng, như chuỗi kết nối đến cơ sở dữ liệu, cũng như các tham số cấu hình khác phục vụ cho ứng dụng. Bên cạnh đó, dữ liệu người dùng hoặc các cài đặt tạm thời có thể được lưu trữ trong Local Storage để hỗ trợ trải nghiệm người dùng mượt mà, đặc biệt khi hệ thống hoạt động offline. Việc sử dụng *App.config* giúp tách biệt mã nguồn và cấu hình, làm cho ứng dụng dễ dàng duy trì và nâng cấp mà không cần phải thay đổi mã nguồn trực tiếp.

*Git* là một hệ thống quản lý phiên bản phân tán, giúp theo dõi các thay đổi của mã nguồn trong suốt quá trình phát triển phần mềm. *Git* cho phép các lập trình viên làm việc độc lập, tạo các nhánh (branch) để phát triển tính năng mới, và sau đó gộp lại các thay đổi khi cần thiết mà không làm gián đoạn công việc của người khác. *GitHub* là dịch vụ lưu trữ mã nguồn trực tuyến, sử dụng *Git* để quản lý phiên bản mã nguồn và tạo điều kiện cho việc hợp tác giữa các lập trình viên. Trong dự án của bạn, *Git* và *GitHub* được sử dụng để theo dõi và quản lý các thay đổi trong mã nguồn, cho phép các lập trình viên làm việc đồng thời trên các tính năng khác nhau mà không gây xung đột. *GitHub* cung cấp các công cụ như pull request, issue tracking, và tích hợp CI/CD để hỗ trợ việc phát triển phần mềm hiệu quả. Việc sử dụng *Git* và *GitHub* giúp đảm bảo mã nguồn được quản lý tốt, dễ dàng kiểm tra lại các phiên bản trước đó và hợp tác một cách suôn sẻ giữa các thành viên trong đội ngũ phát triển.



Hình 2.9 Git and GitHub

## CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 3.1. Mô tả bài toán

Hệ thống **WPFCommerceApp** được xây dựng nhằm giải quyết bài toán quản lý bán hàng trực tuyến trên nền tảng ứng dụng WPF (Windows Presentation Foundation). Đây là một hệ thống thương mại điện tử tập trung vào việc cung cấp giao diện người dùng thân thiện để quản lý sản phẩm, khách hàng, đơn hàng và các hoạt động kinh doanh khác. Bài toán được chia thành các nhóm chức năng cụ thể như sau:

#### 3.1.1. Quản lý sản phẩm

Quản lý sản phẩm là một trong những chức năng cốt lõi của hệ thống, hỗ trợ doanh nghiệp dễ dàng kiểm soát thông tin sản phẩm trên nền tảng. Các yêu cầu chính của bài toán bao gồm:

- **Thêm mới sản phẩm:** Người quản trị có thể nhập thông tin về sản phẩm mới, bao gồm:
  - **Tên sản phẩm:** Tên gọi chính của sản phẩm.
  - **Thương hiệu:** Thông tin về thương hiệu sản phẩm.
  - **Giới tính:** Phân loại đối tượng sử dụng (Nam, Nữ, Unisex).
  - **Giá bán:** Giá gốc của sản phẩm.
  - **Giảm giá:** Tỷ lệ phần trăm hoặc số tiền giảm giá áp dụng cho sản phẩm.
  - **Tình trạng:** Đánh dấu sản phẩm là mới (IsNew) hoặc bán chạy (IsHot).
  - **Hình ảnh:** Ảnh đại diện minh họa sản phẩm để hiển thị trực quan trên giao diện.
- **Chỉnh sửa sản phẩm:** Cho phép cập nhật thông tin sản phẩm khi có thay đổi như giá cả, trạng thái khuyến mãi hoặc hình ảnh.
- **Xóa sản phẩm:** Hệ thống hỗ trợ xóa sản phẩm không còn kinh doanh, đồng thời đảm bảo dữ liệu liên quan (ví dụ: đơn hàng, đánh giá) không bị ảnh hưởng.
- **Hiển thị danh sách sản phẩm:** Cung cấp danh sách sản phẩm kèm thông tin chi tiết để người dùng hoặc khách hàng dễ dàng tìm kiếm và lựa chọn.

#### 3.1.2. Quản lý khách hàng

Để duy trì mối quan hệ khách hàng, hệ thống cần đảm bảo các chức năng sau:

- **Lưu trữ thông tin khách hàng:** Bao gồm tên, email, số điện thoại, địa chỉ giao hàng và lịch sử mua sắm.
- **Theo dõi hành vi mua sắm:** Ghi nhận thông tin về các sản phẩm đã xem, các đơn hàng đã đặt để cá nhân hóa đề xuất sản phẩm.
- **Quản lý tài khoản:** Hỗ trợ khách hàng đăng ký, đăng nhập, và cập nhật thông tin cá nhân.

### 3.1.3. Quản lý đơn hàng

Hệ thống cho phép doanh nghiệp theo dõi và xử lý đơn hàng một cách linh hoạt. Các tính năng chính gồm:

- **Tạo đơn hàng:** Khách hàng có thể thêm sản phẩm vào giỏ hàng và tiến hành đặt hàng.
- **Theo dõi trạng thái đơn hàng:** Hệ thống quản lý trạng thái đơn hàng theo các giai đoạn:
  - Đang xử lý (Processing)
  - Đã giao hàng (Delivered)
  - Hủy đơn (Canceled)
- **Quản lý thanh toán:** Hệ thống hỗ trợ ghi nhận thanh toán qua các phương thức như tiền mặt, chuyển khoản hoặc ví điện tử.
- **Quản lý giao hàng:** Theo dõi thông tin vận chuyển, đảm bảo đơn hàng được giao đúng thời gian và địa chỉ.

### 3.1.4. Đánh giá và phản hồi sản phẩm

Để cải thiện chất lượng sản phẩm và dịch vụ, hệ thống tích hợp chức năng đánh giá từ khách hàng:

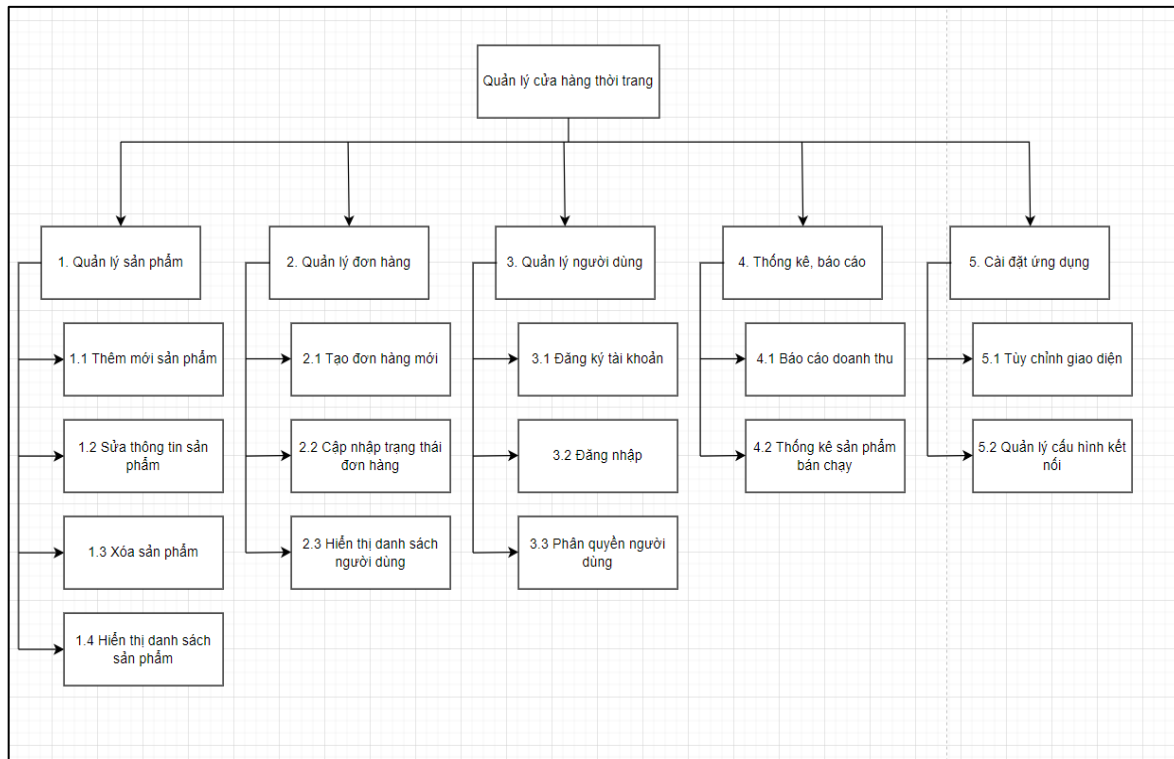
- **Điểm đánh giá (RatePoint):** Khách hàng có thể để lại điểm số (ví dụ: 1 đến 5 sao) cho mỗi sản phẩm đã mua.
- **Nhận xét và phản hồi:** Hỗ trợ khách hàng chia sẻ ý kiến cá nhân về sản phẩm, từ đó doanh nghiệp có thể cải tiến dịch vụ.
- **Tính toán điểm trung bình:** Hệ thống tự động tính toán điểm đánh giá trung bình cho từng sản phẩm, giúp người dùng có cái nhìn tổng quan về chất lượng sản phẩm.

### 3.1.5. Giao diện thân thiện và hiệu quả

Hệ thống được xây dựng trên nền tảng **WPF (Windows Presentation Foundation)** với mục tiêu:

- **Tính trực quan:** Giao diện được thiết kế đơn giản, dễ sử dụng cho cả quản trị viên và khách hàng.
- **Tốc độ xử lý cao:** Đảm bảo phản hồi nhanh khi người dùng thao tác.
- **Tính linh hoạt:** Hệ thống có thể mở rộng hoặc thay đổi chức năng theo nhu cầu kinh doanh trong tương lai.

### 3.2. Biểu đồ phân rã chức năng (BFD)



Hình 3.1 Biểu đồ BFD

Biểu đồ phân rã chức năng (BFD - Breakdown Function Diagram) của dự án WPFCommerceApp là một biểu diễn trực quan mô tả cấu trúc phân cấp các chức năng chính của ứng dụng. Ở cấp cao nhất, ứng dụng được chia thành năm nhóm chức năng chính, bao gồm: **Quản lý sản phẩm**, **Quản lý đơn hàng**, **Quản lý người dùng**, **Thống kê và báo cáo**, và **Cài đặt ứng dụng**.

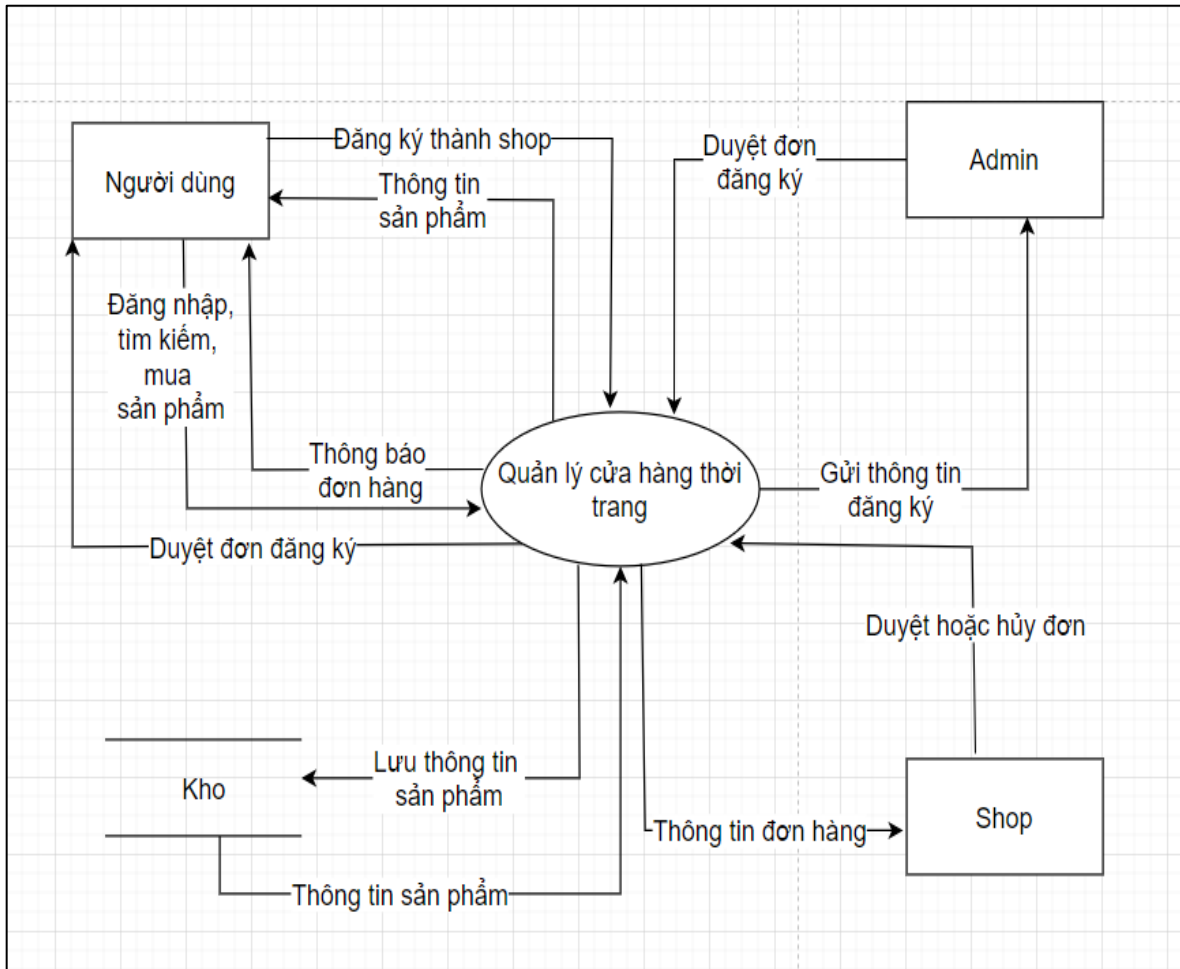
Mỗi nhóm chức năng lại được phân rã chi tiết hơn. Ví dụ, **Quản lý sản phẩm** bao gồm các chức năng như thêm, sửa, xóa và hiển thị danh sách sản phẩm. **Quản lý đơn hàng** cung cấp các tính năng tạo đơn hàng mới, cập nhật trạng thái đơn hàng và hiển thị danh sách đơn hàng. Tương tự, **Quản lý người dùng** tập trung vào các chức năng liên quan đến đăng ký, đăng nhập và phân quyền người dùng.

Ngoài ra, nhóm chức năng **Thống kê và báo cáo** hỗ trợ tạo báo cáo doanh thu và thống kê các sản phẩm bán chạy, giúp doanh nghiệp đưa ra các quyết định kinh doanh hiệu quả. **Cài đặt ứng dụng** cho phép người dùng tùy chỉnh giao diện và quản lý cấu hình kết nối.

Biểu đồ BFD không chỉ giúp tổ chức các chức năng của dự án một cách logic mà còn hỗ trợ quá trình phát triển và bảo trì ứng dụng hiệu quả hơn.

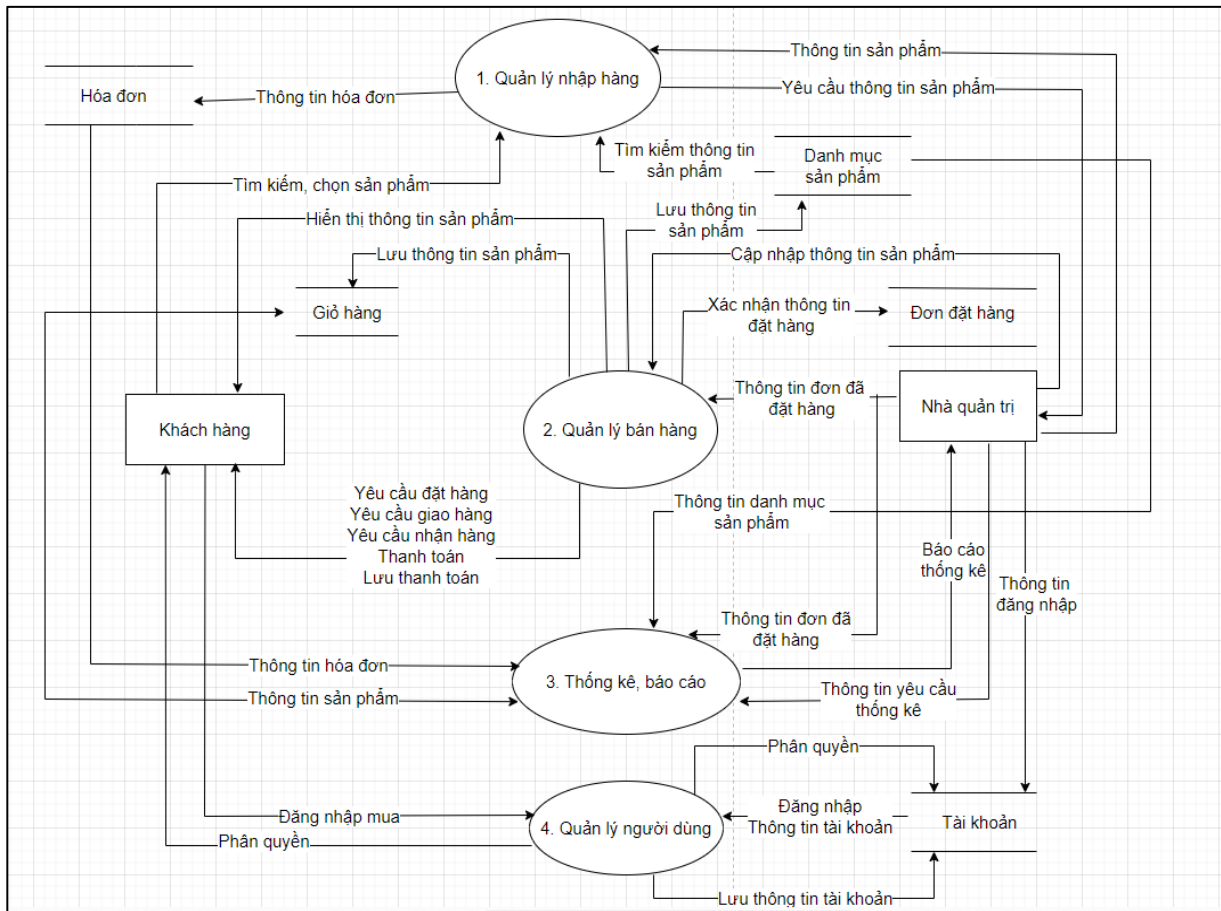
### 3.3. Biểu đồ luồng dữ liệu (DFD)

#### 3.3.1. Biểu đồ mức ngữ cảnh



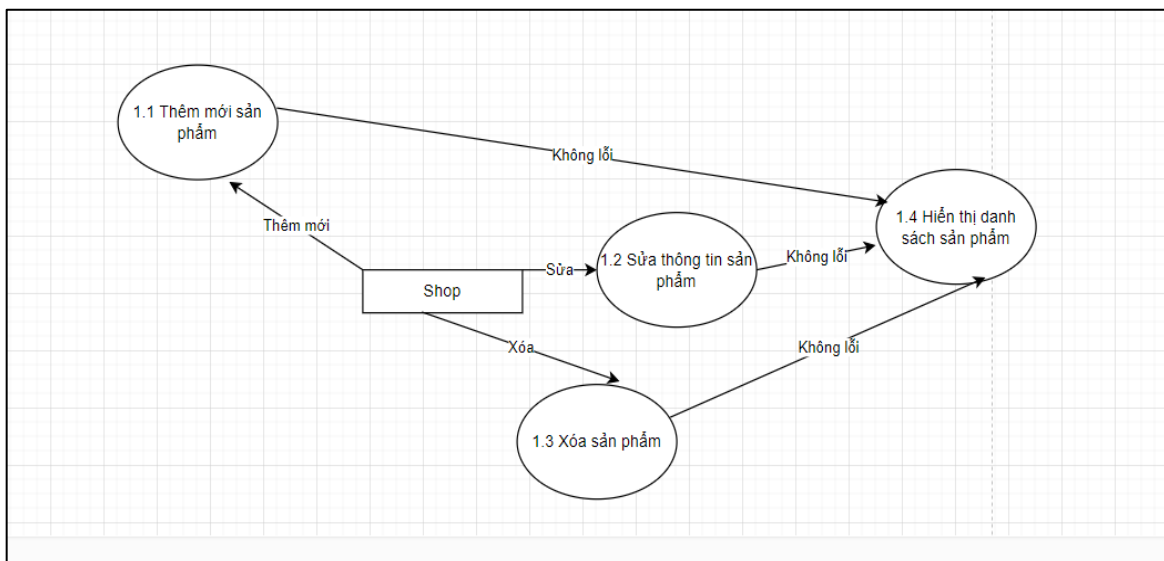
Hình 3.2 Biểu đồ mức ngữ cảnh

### 3.3.2. Mức đỉnh (mức 0)

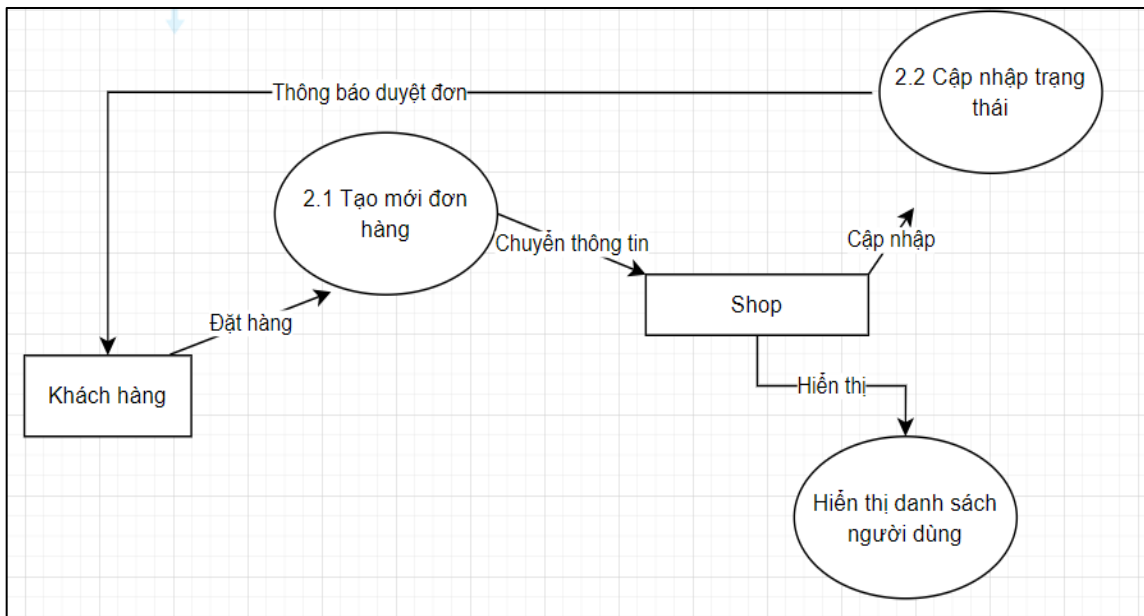


Hình 3.3 Biểu đồ DFD mức 0

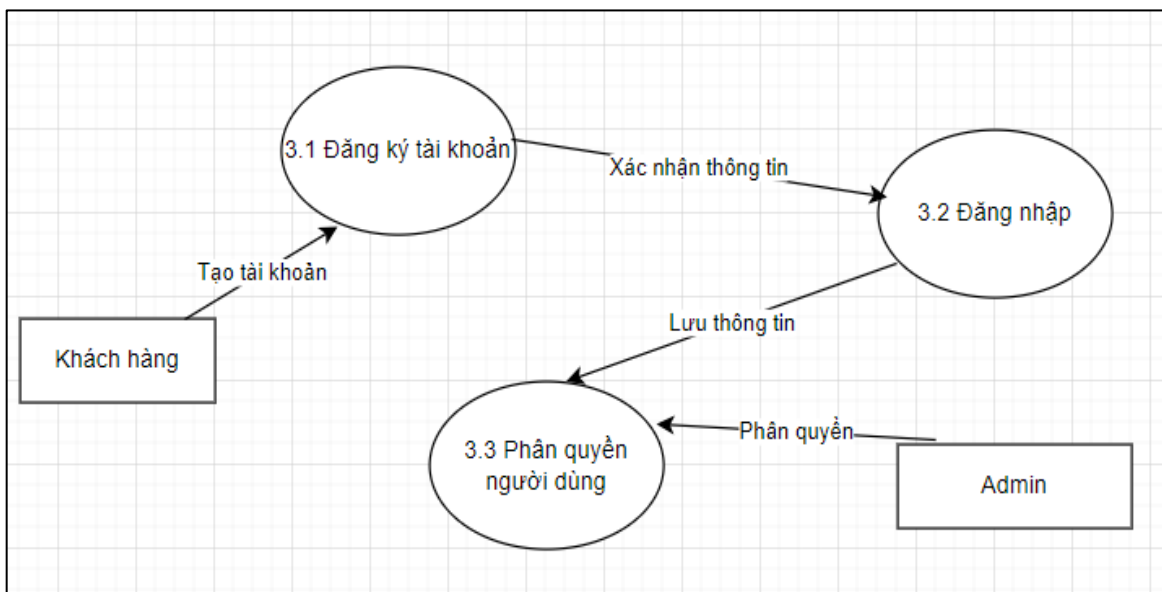
### 3.3.3. Mức dưới đỉnh (mức 1)



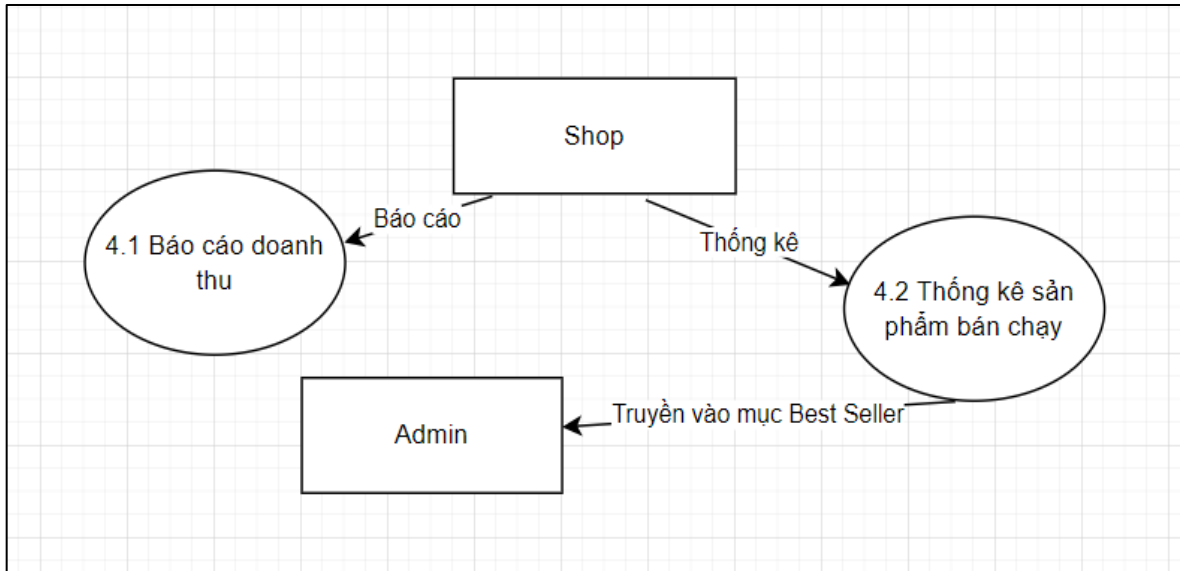
Hình 3.4 Biểu đồ DFD mức 1\_Quản lý sản phẩm



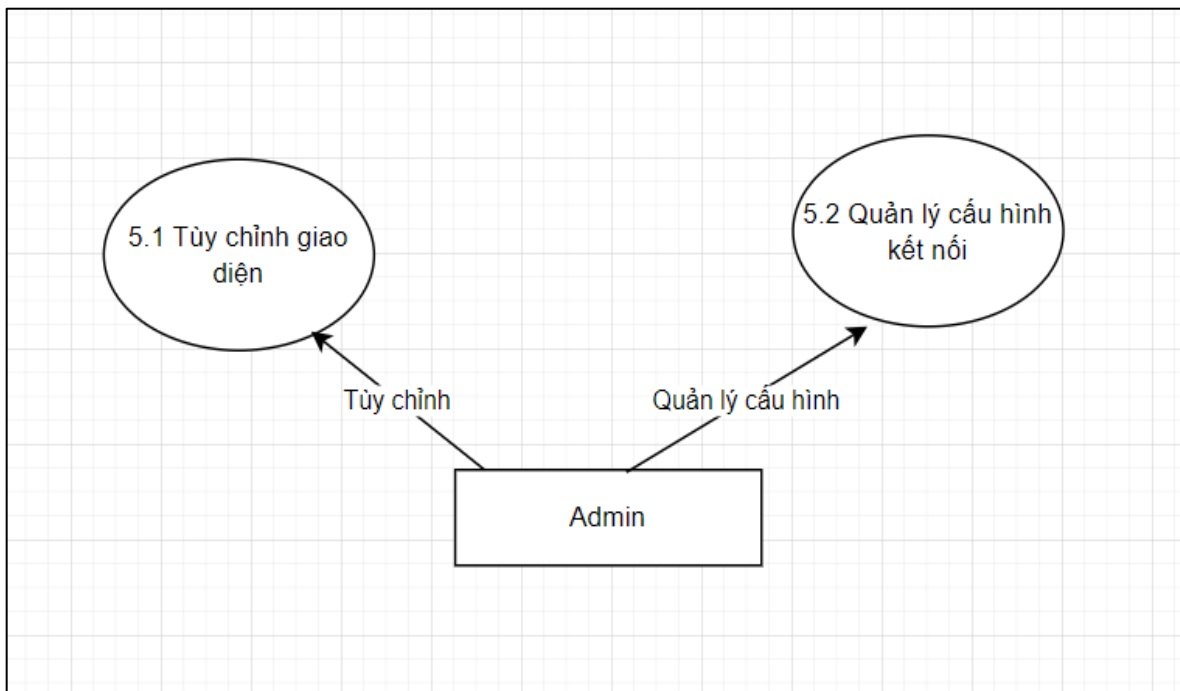
Hình 3.5 Biểu đồ DFD mức 1\_Quản lý đơn hàng



Hình 3.6 Biểu đồ DFD mức 1\_Quản lý người dùng



Hình 3.7 Biểu đồ DFD mức 1\_ Thống kê, báo cáo



Hình 3.8 Biểu đồ DFD mức 1\_ Cài đặt ứng dụng

### 3.4. Mô hình thực thể - liên kết ER (Entity – Relationship Model)

#### 3.4.1. Sơ đồ ERD

Sơ đồ ERD (Entity-Relationship Diagram) của chương trình WPFCommerceApp mô tả các thực thể chính trong hệ thống và mối quan hệ giữa chúng. Các thực thể này bao gồm



"Product" (Sản phẩm), "Order" (Đơn hàng), "User" (Người dùng) và "OrderDetail" (Chi tiết đơn hàng), mỗi thực thể đại diện cho một đối tượng quan trọng trong hệ thống quản lý thương mại điện tử.

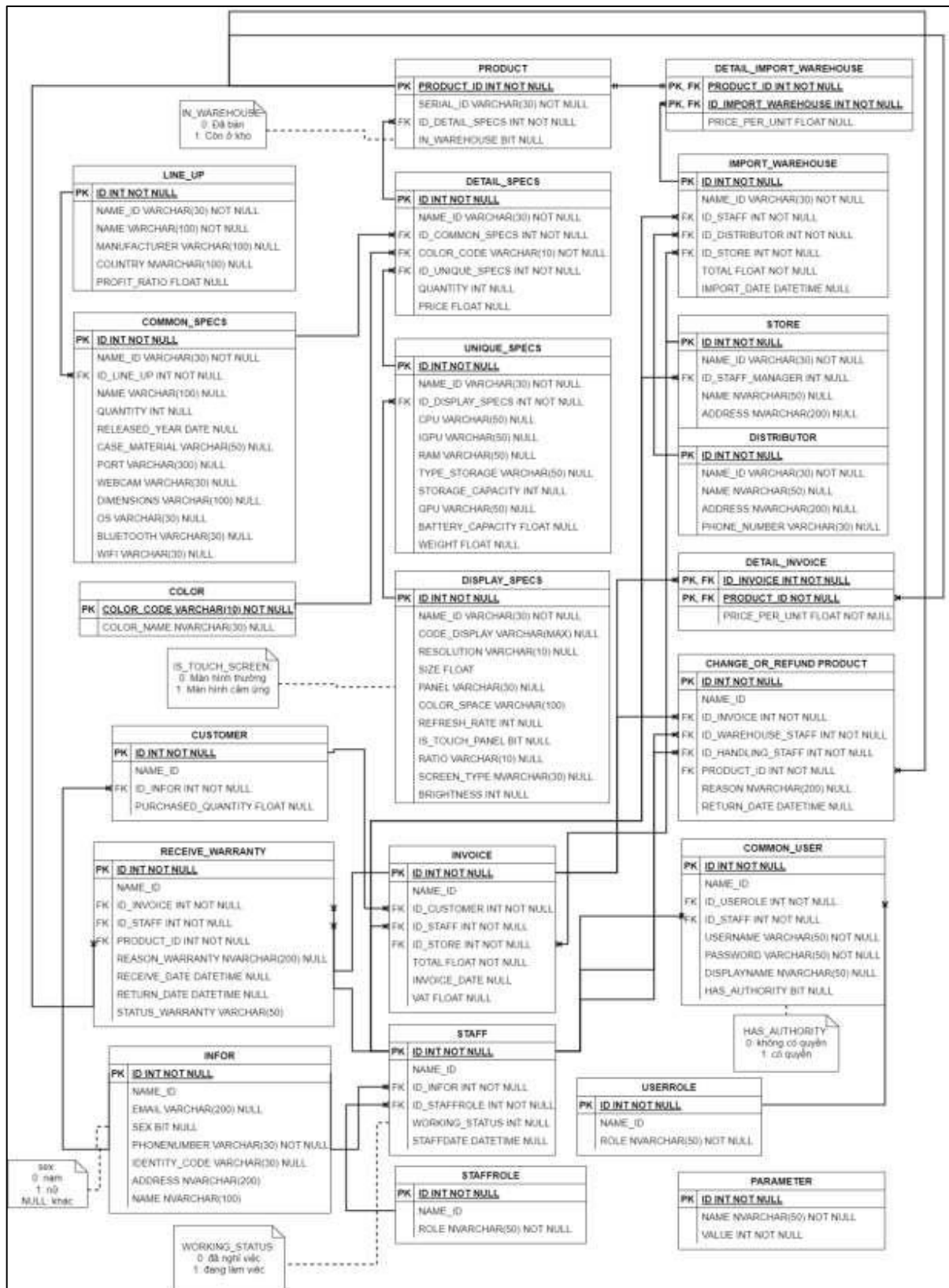
Thực thể "Product" chứa thông tin về các sản phẩm mà cửa hàng kinh doanh, bao gồm các thuộc tính như mã sản phẩm, tên, thương hiệu, giá, giảm giá, các thuộc tính đánh dấu sản phẩm hot hoặc mới, và hình ảnh của sản phẩm. Đây là thực thể cốt lõi trong hệ thống vì nó liên quan trực tiếp đến hoạt động bán hàng.

Thực thể "Order" lưu trữ các thông tin về đơn hàng mà người dùng thực hiện, bao gồm mã đơn hàng, ngày đặt hàng, tổng giá trị đơn hàng, mã người dùng (liên kết với thực thể "User") và trạng thái của đơn hàng (chưa thanh toán, đã thanh toán, đã giao hàng...). Mỗi quan hệ giữa "Order" và "User" là quan hệ một-nhiều, bởi mỗi người dùng có thể có nhiều đơn hàng, nhưng mỗi đơn hàng chỉ thuộc về một người dùng.

Thực thể "User" quản lý thông tin người dùng của hệ thống, bao gồm các thuộc tính như mã người dùng, tên đăng nhập, mật khẩu, email và vai trò (Khách hàng hoặc Quản trị viên). Mỗi quan hệ giữa "User" và "Order" là một-nhiều, vì một người dùng có thể thực hiện nhiều đơn hàng.

Thực thể "OrderDetail" mô tả các sản phẩm có trong một đơn hàng, liên kết "Order" và "Product" thông qua các thuộc tính như mã chi tiết đơn hàng, số lượng sản phẩm trong đơn hàng và giá của sản phẩm tại thời điểm mua. Mỗi quan hệ giữa "Order" và "OrderDetail" là một-nhiều, vì một đơn hàng có thể chứa nhiều sản phẩm, trong khi mỗi quan hệ giữa "OrderDetail" và "Product" là nhiều-một, vì mỗi chi tiết đơn hàng chỉ thuộc về một sản phẩm.

Thông qua sơ đồ ERD này, ta có thể thấy được cấu trúc dữ liệu của hệ thống thương mại điện tử, giúp hiểu rõ cách các thực thể liên kết với nhau để hỗ trợ các chức năng như quản lý sản phẩm, đơn hàng, người dùng và các thống kê liên quan trong quá trình giao dịch và quản lý.



Hình 3.9 Sơ đồ ERD

### 3.4.2. Các thực thể và thuộc tính

**Bảng UserLogin:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/ghi chú
1	IdUser	varchar(30)	primary key	ID người dùng, khóa chính
2	CreatedDate	datetime	Không	Ngày tạo tài khoản
3	Password	varchar(100)	Không	Mật khẩu của người dùng
4	Username	nvarchar(100)	Unique	Tên đăng nhập (duy nhất)
5	Salt	varchar(100)	Không	Chuỗi Salt bảo mật mật khẩu
6	Provider	int	Không	Loại tài khoản: 0 - Email/Password 1 - Googl0065 -1 - Admin

**Bảng MUser:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	Id	varchar(30)	khóa chính primary key	ID người dùng
2	Role	varchar(5)	not null default	Vai trò: Admin, User, Shop. Mặc định là User
3	Name	nvarchar(200)	not null default	Tên đầy đủ của người dùng
4	PhoneNumber	varchar(15)	not null default	Số điện thoại người dùng
5	Email	nvarchar(200)	not null default	Email người dùng
6	Gender	bit	not null default	Giới tính: 0 – Nam 1 - Nữ
7	DOB	smalldatetime		Ngày sinh
8	Description	nvarchar(1000)	not null default	Mô tả người dùng

9	StatusUser	varchar(10)		Trạng thái tài khoản: Banned hoặc NotBanned
10	StatusShop	varchar(10)		Trạng thái cửa hàng: NotExist, Banned, NotBanned
11	SourceImageAva	nvarchar(200)		Đường dẫn ảnh đại diện
12	SourceImageBackground	nvarchar(200)		Đường dẫn ảnh nền
13	DefaultAddress	varchar(14)		Địa chỉ mặc định của người dùng

### Bảng Product:

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	Id	varchar(8)	khóa chính primary key	ID sản phẩm, khóa chính
2	Name	nvarchar(200)	not null default	Tên sản phẩm
3	IdCategory	varchar(8)	not null	ID danh mục sản phẩm
4	IdBrand	varchar(8)	not null	ID thương hiệu sản phẩm
5	IdShop	varchar(30)	not null	ID cửa hàng cung cấp sản phẩm
6	Price	float(53)	not null default 0	Giá sản phẩm
7	Sale	int	not null default 0	Phần trăm giảm giá
8	InStock	int	not null default 0	Số lượng tồn kho
9	Sold	int	not null default 0	Số lượng đã bán
10	IsOneSize	bit	not null default 0	Sản phẩm chỉ có một size
11	IsHadSizeS	bit	not null default 0	Có size S
12	IsHadSizeM	bit	not null default 0	Có size M

13	IsHadSizeL	bit	not null default 0	Có size L
14	IsHadSizeXL	bit	not null default 0	Có size XL
15	IsHadSizeXXL	bit	not null default 0	Có size XXL
16	Color	nvarchar(200)	not null default ''	Màu sắc sản phẩm
17	Description	nvarchar(1000)	not null default ''	Mô tả chi tiết sản phẩm
18	DateOfSale	smalldatetime		Ngày mở bán
19	BanLevel	int		Mức độ cấm bán: 0 - Không cấm

**Bảng FavouriteProduct:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	IdUser	varchar(30)	not null	ID người dùng
2	IdProduct	varchar(8)	not null	ID sản phẩm
3	primary key	(IdUser, IdProduct)		Khóa chính kết hợp giữa người dùng và sản phẩm

**Bảng ImageProduct:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	IdProduct	varchar(8)	not null	ID sản phẩm
2	Source	varchar(200)	not null	Đường dẫn hình ảnh sản phẩm
3	primary key	(IdProduct, Source)		Khóa chính kết hợp giữa sản phẩm và hình ảnh

**Bảng Cart:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	IdUser	varchar(30)	not null	ID người dùng

2	IdProduct	varchar(8)		ID sản phẩm
3	Amount	int	default 1	Số lượng sản phẩm
4	Size	varchar(7)		Kích cỡ: S, M, L, XL, XXL, OneSize
5	primary key	(IdUser, IdProduct, Size)		Khóa chính kết hợp giữa người dùng, sản phẩm, kích cỡ

**Bảng Category:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	Id	varchar(8)	primary key	ID danh mục, khóa chính
2	Name	nvarchar(200)	not null unique)	Tên danh mục (duy nhất)
3	Status	varchar(10)		Trạng thái danh mục: Banned hoặc NotBanned

**Bảng Brand:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	Id	varchar(8)	primary key	ID thương hiệu, khóa chính
2	Name	nvarchar(200)	not null unique	Tên thương hiệu (duy nhất)
3	Status	varchar(10)		Trạng thái thương hiệu: Banned hoặc NotBanned

**Bảng Advertisement:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	Id	varchar(8)	primary key	ID quảng cáo, khóa chính
2	Image	nvarchar(200)	not null	Đường dẫn hình ảnh quảng cáo

**Bảng AdInUser:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	Id	varchar(8)	Primary key	ID quảng cáo đang sử dụng
2	Position	int	check (Position in (1, 2, 3)) unique	Vị trí hiển thị quảng cáo: 1, 2 hoặc 3
3	primary key	(Id, Position)		Khóa chính kết hợp giữa ID và vị trí quảng cáo

**Bảng Notification:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	Id	varchar(8)	primary key	ID thông báo, khóa chính
2	IdReceiver	varchar(30)	not null	ID người nhận thông báo
3	IdSender	varchar(30)	not null	ID người gửi thông báo
4	Date	smalldatetime	not null	Ngày gửi thông báo
5	Content	nvarchar(1000)	not null default "	Nội dung thông báo
6	HaveSeen	bit		Trạng thái đã xem:  0 – Chưa xem  1 - Đã xem

**Bảng MOrder:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	Id	varchar(8)	primary key	ID đơn hàng, khóa chính
2	IdCustomer	varchar(30)	not null	ID khách hàng
3	IdShop	varchar(30)	not null	ID cửa hàng
4	ShipTotal	float(53)	default 0	Chi phí vận chuyển
5	DateBegin	smalldatetime		Ngày bắt đầu đặt hàng
6	DateEnd	smalldatetime		Ngày hoàn thành đơn hàng
7	OrderTotal	float(53)	not null default 0	Tổng giá trị đơn hàng
8	Discounted	float(53)		Giá trị đã giảm
9	Promo	varchar(8)		Mã khuyến mãi áp dụng
10	ShippingMethod	int		Phương thức vận chuyển
11	ShippingSpeedMethod	int		Phương thức tốc độ giao hàng

12	AddressIndex	varchar(14)		Địa chỉ giao hàng
13	Status	varchar(20)		Trạng thái đơn hàng: Processing, Delivering, Delivered, Cancelled, Completed

**Bảng OrderInfo:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	IdOrder	varchar(8)	not null	ID đơn hàng
2	IdProduct	varchar(8)	not null	ID sản phẩm
3	IdRating	varchar(8)		ID đánh giá sản phẩm
4	ImageProduct	varchar(200)		Hình ảnh sản phẩm
5	Size	varchar(10)		Kích thước sản phẩm
6	Count	int	not null default 0	Số lượng sản phẩm
7	TotalPrice	float(53)	not null default 0	Tổng giá trị sản phẩm
8	primary key	(IdOrder, IdProduct, Size)		Khóa chính kết hợp giữa ID đơn hàng, sản phẩm, kích thước

**Bảng Rating:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	Id	varchar(8)	primary key	ID đánh giá, khóa chính
2	DateRating	smalldatetime		Ngày đánh giá
3	Rating	int		Số sao đánh giá
4	Comment	nvarchar(500)		Bình luận đánh giá

**Bảng RatingInfo:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	IdUser	varchar(30)	not null i	ID người dùng phản hồi
2	IdRating	varchar(8)	not null	ID đánh giá được phản hồi
3	Comment	nvarchar(500)		Nội dung phản hồi
4	DateReply	datetime	not null	Ngày phản hồi



5	primary key	(IdUser, IdRating, DateReply)	Khóa chính kết hợp giữa người dùng, đánh giá, ngày phản hồi
---	-------------	-------------------------------	---

**Bảng BrandRequest:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	Id	varchar(8)	primary key	ID yêu cầu thương hiệu, khóa chính
2	IdShop	varchar(30)		ID cửa hàng gửi yêu cầu
3	Name	nvarchar(200)	not null unique	Tên thương hiệu yêu cầu
4	Reason	nvarchar(1000)	default " not null	Lý do yêu cầu

**Bảng CategoryRequest:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	Id	varchar(8)	primary key	ID yêu cầu danh mục, khóa chính
2	IdShop	varchar(30)		ID cửa hàng gửi yêu cầu
3	Name	nvarchar(200)	not null unique	Tên danh mục yêu cầu
4	Reason	nvarchar(1000)	default " not null	Lý do yêu cầu

**Bảng ShopRequest:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	Id	varchar(8)	primary key	ID yêu cầu cửa hàng, khóa chính
2	IdUser	varchar(30)		ID người dùng gửi yêu cầu
3	Description	nvarchar(1000)	default "	Mô tả yêu cầu

**Bảng Promo:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	Id	varchar(8)	primary key	ID khuyến mãi, khóa chính
2	IdShop	varchar(30)	not null	ID cửa hàng cung cấp khuyến mãi
3	Code	nvarchar(15)	not null	Mã khuyến mãi

4	Name	nvarchar(150)		Tên khuyến mãi
5	Description	nvarchar(200)		Mô tả khuyến mãi
6	DateBegin	smalldatetime		Ngày bắt đầu khuyến mãi
7	DateEnd	smalldatetime		Ngày kết thúc khuyến mãi
8	Amount	int		Số lượng khuyến mãi
9	AmountUsed	int		Số lượng đã sử dụng
10	MaxSale	float(53)		Giá trị giảm tối đa
11	MinCost	float(53)		Giá trị đơn hàng tối thiểu
12	CustomerType	int		Loại khách hàng áp dụng
13	Sale	float(53)		Tỷ lệ giảm giá
14	Status	int		Trạng thái khuyến mãi

**Bảng PromoDetail:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	IdProduct	varchar(8)		ID sản phẩm áp dụng khuyến mãi
2	IdPromo	varchar(8)		ID khuyến mãi
3	primary key	(IdProduct, IdPromo)		Khóa chính kết hợp giữa sản phẩm và khuyến mãi

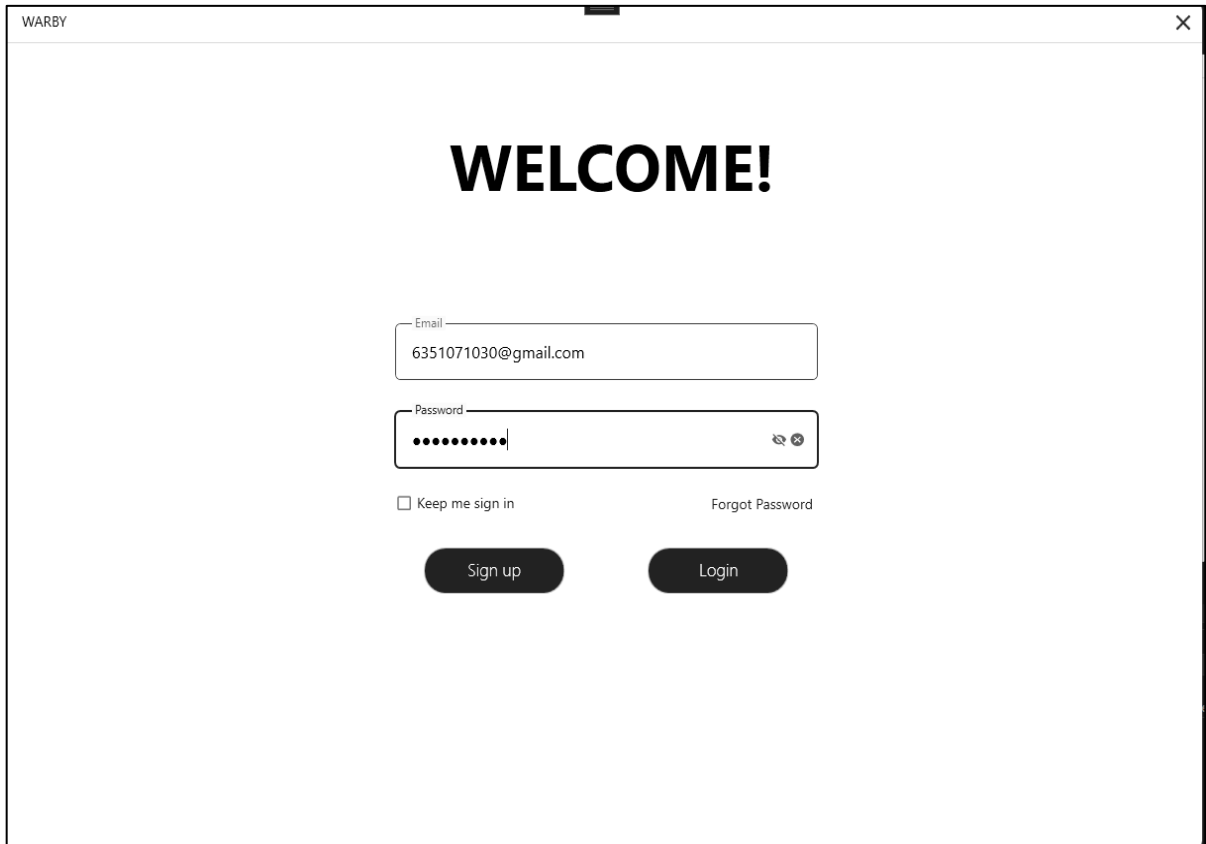
**Bảng Address:**

Thứ tự	Tên thuộc tính	Kiểu	Ràng buộc	Ý nghĩa/Ghi chú
1	Id	varchar(14)	primary key	ID địa chỉ, khóa chính
2	IdUser	varchar(30)		ID người dùng sở hữu địa chỉ
3	City	nvarchar(200)		Tên thành phố
4	District	nvarchar(200)		Tên quận/huyện
5	Ward	nvarchar(200)		Tên phường/xã
6	Street	nvarchar(500)		Địa chỉ chi tiết
7	FullAddress	nvarchar(1200)		Địa chỉ đầy đủ dạng chuỗi

8	Type	int check(Type in (1, 2, 3))	Loại địa chỉ:  1 - Nhà riêng,  2 - Nơi làm việc,  3 - Khác
9	IsDefault	bit	Cờ đánh dấu địa chỉ mặc định

## CHƯƠNG 4: GIAO DIỆN CHƯƠNG TRÌNH

### 4.1. Giao diện User



WARBY

# WELCOME!

Email  
6351071030@gmail.com

Password  
••••••••

☐ Keep me sign in [Forgot Password](#)

[Sign up](#) [Login](#)

Hình 4.1 Giao diện Login



WARBY

# WELCOME!

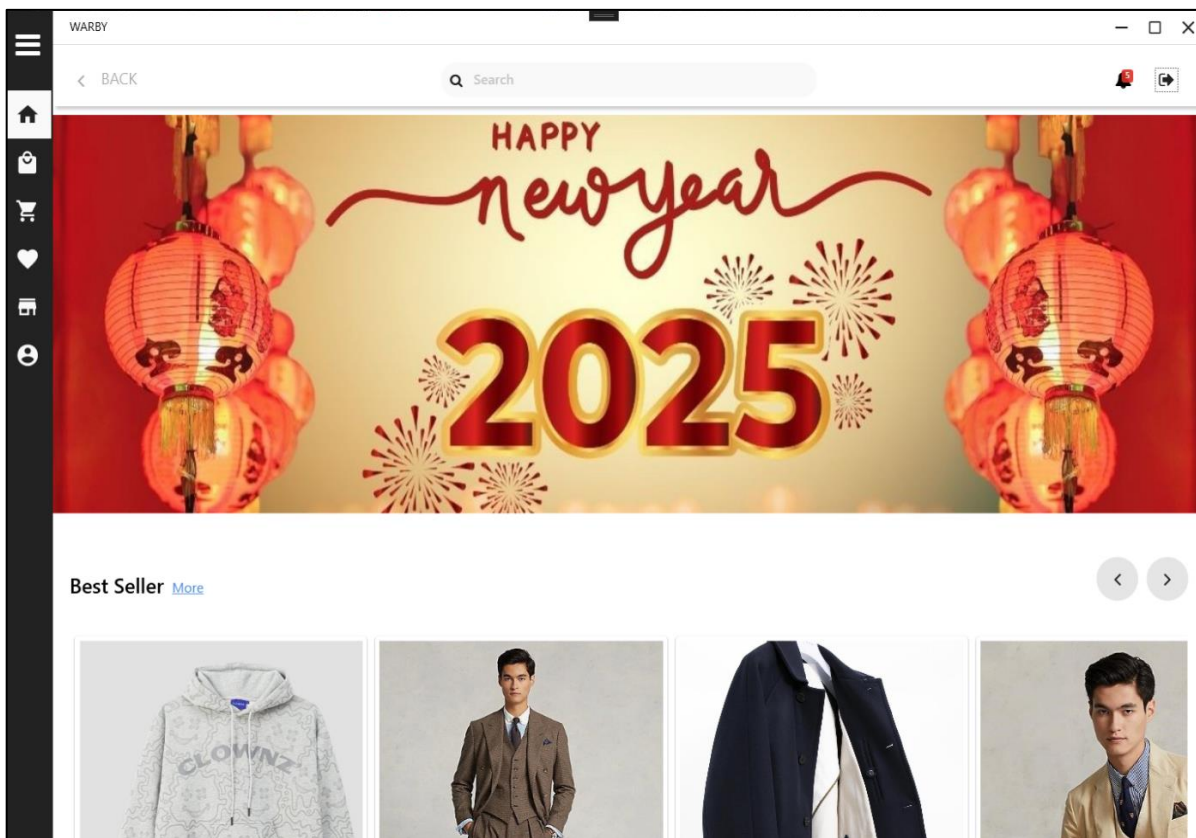
Email  
6351071030@gmail.com

Password  
••••••••

☐ Keep me sign in [Forgot Password](#)

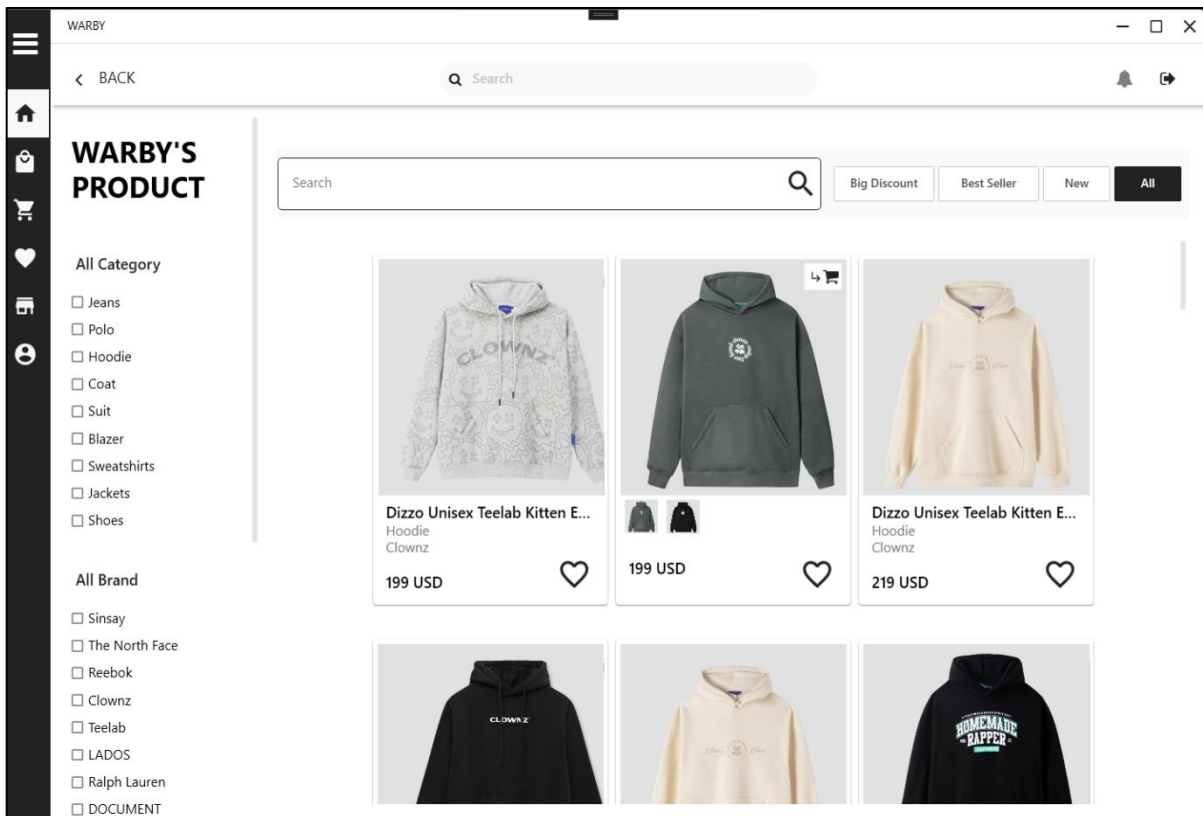
[Sign up](#) [Login](#)

Hình 4.2 Giao diện Sign up

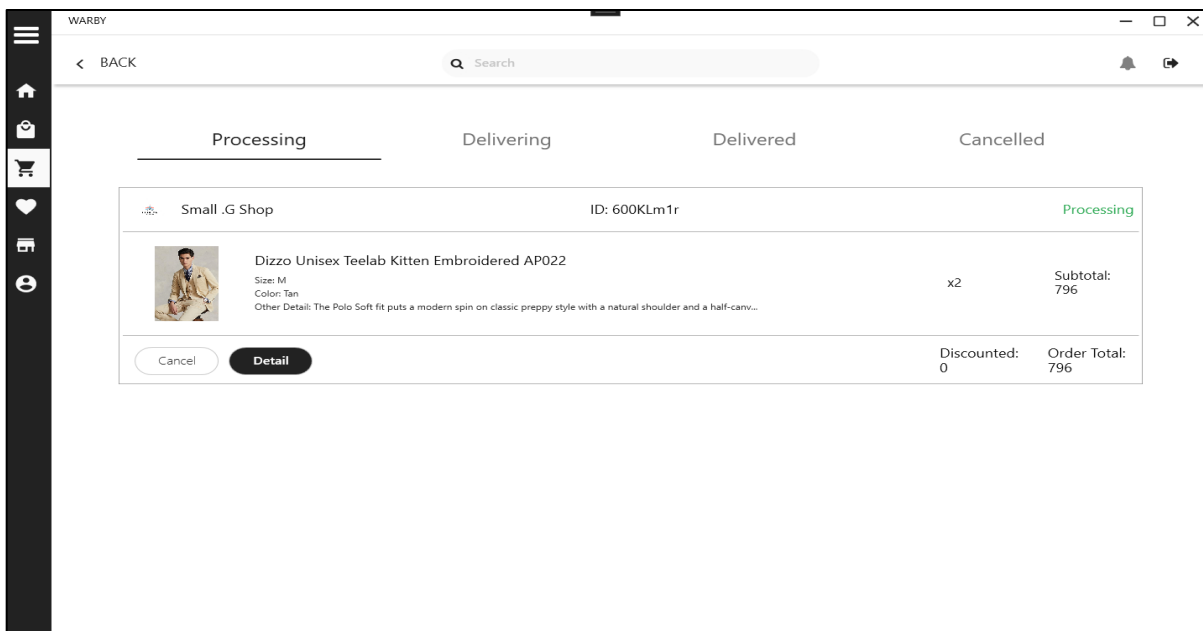


Hình 4.3 Giao diện chính

Hình 4.4 Thông tin User

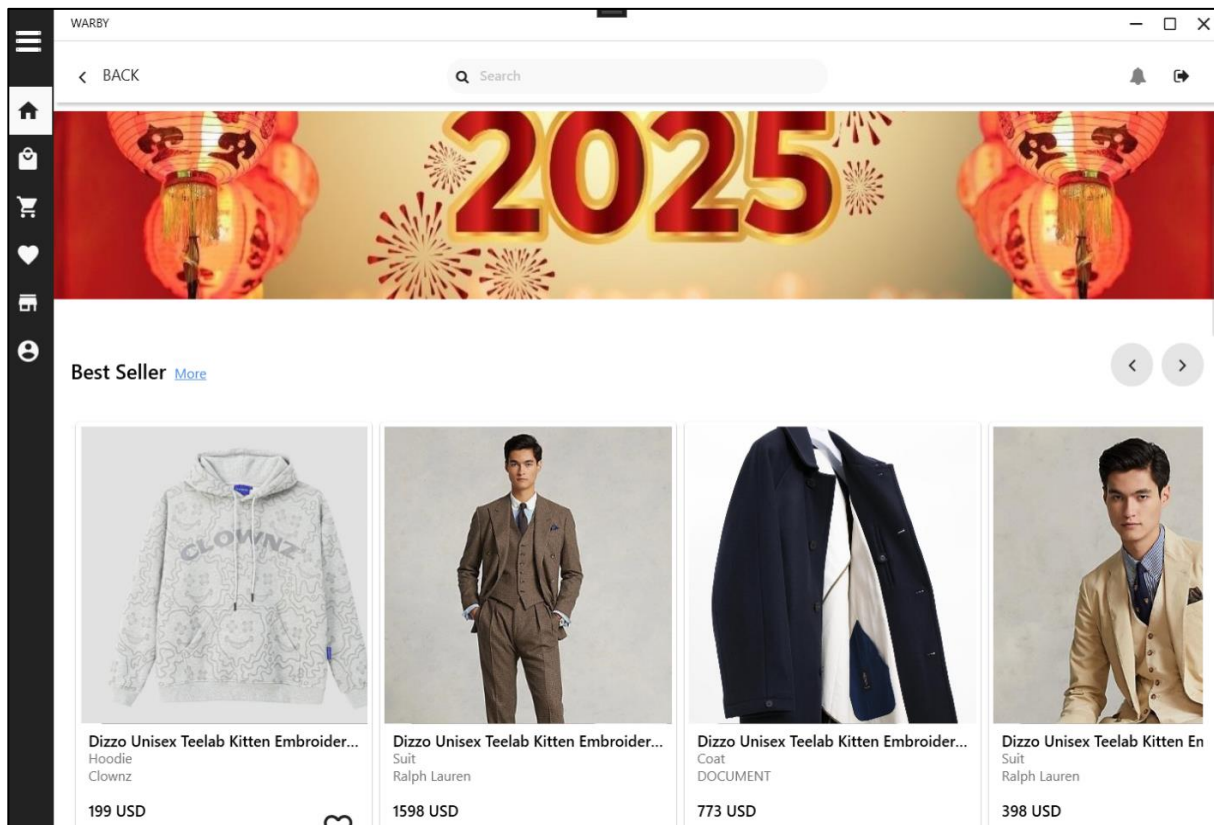


Hình 4. 5 Giao diện tất cả sản phẩm

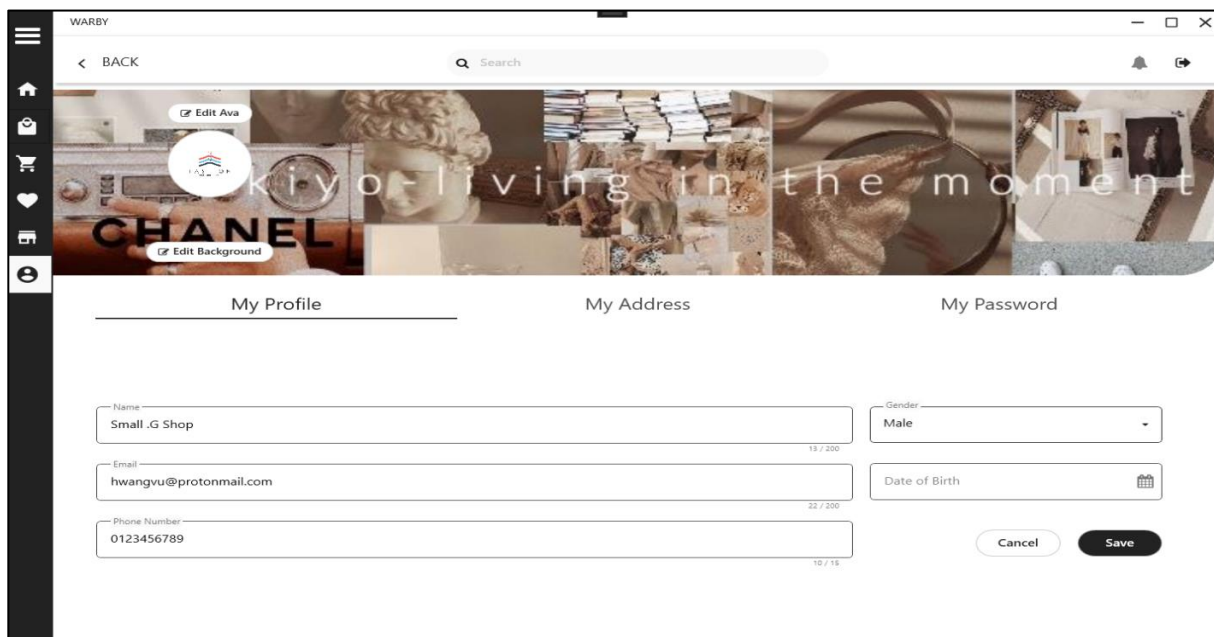


Hình 4.6 Giao diện đơn đặt hàng

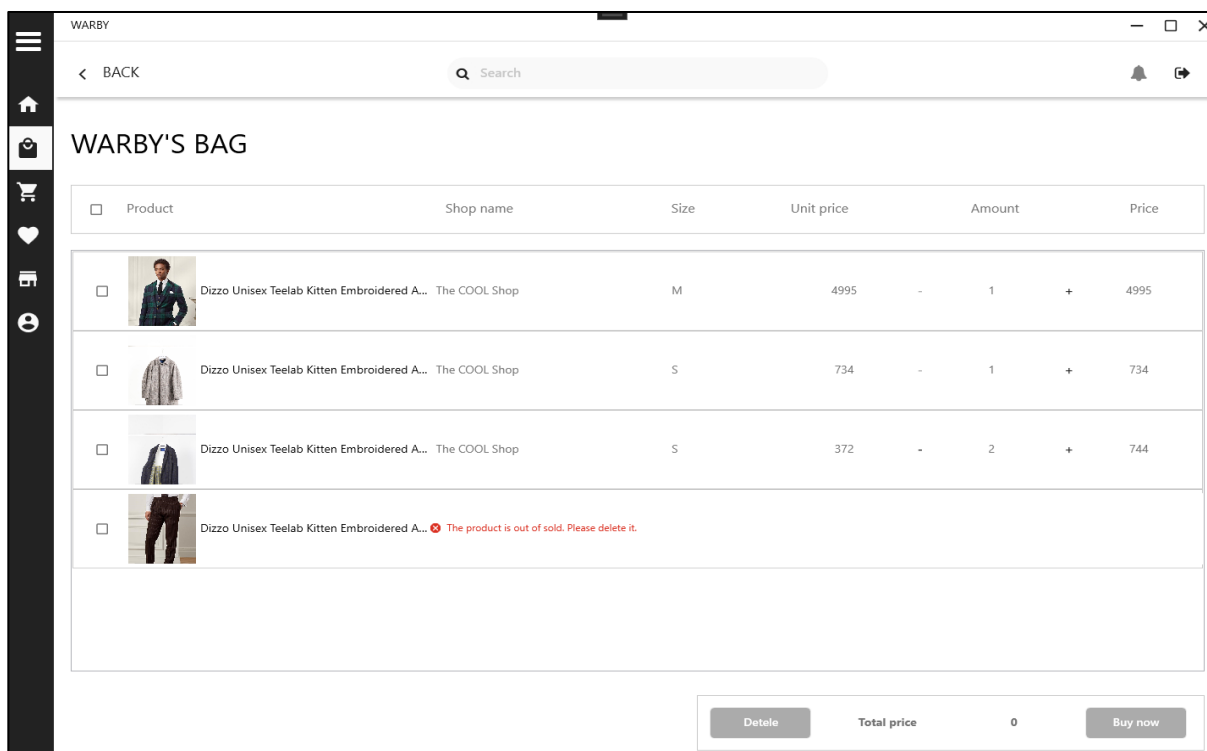
## 4.2. Giao diện Shop



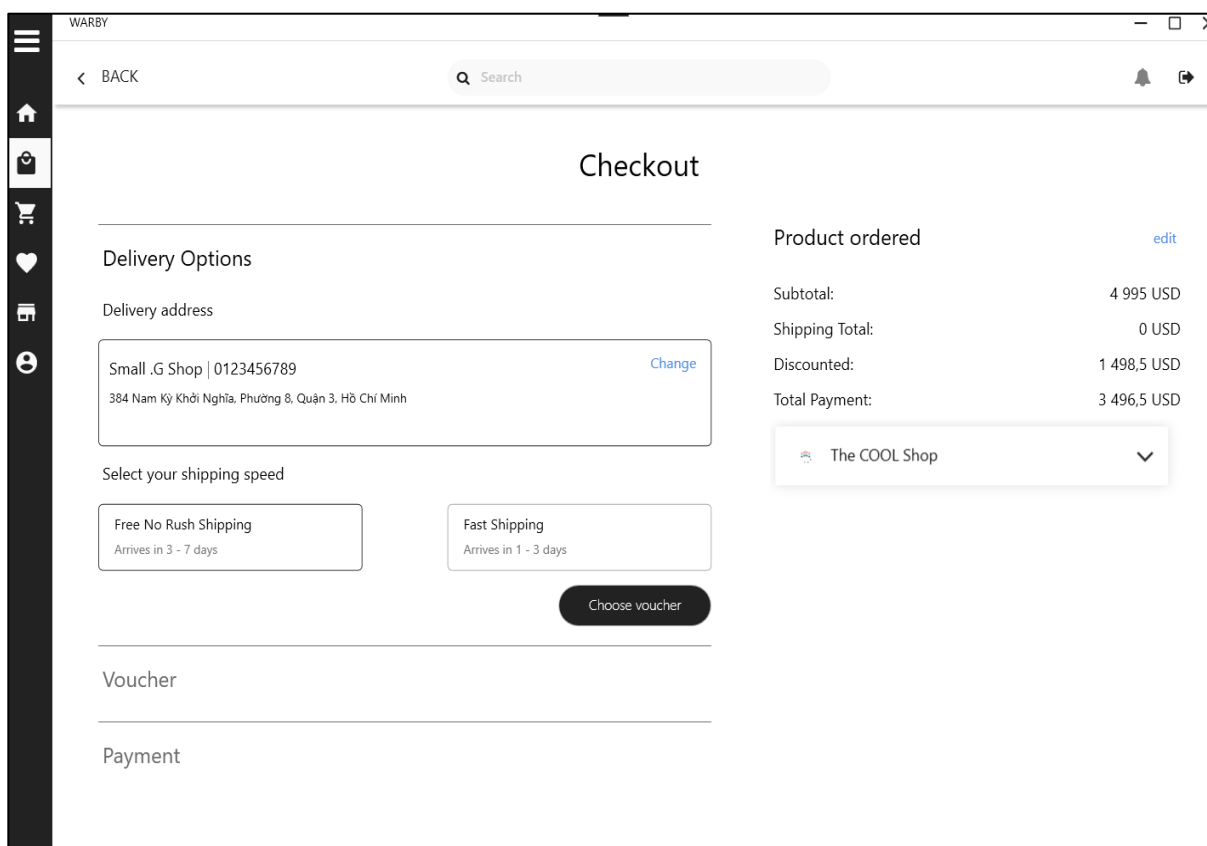
Hình 4.7 Giao diện trang chủ



Hình 4.8 Giao diện thông tin Shop

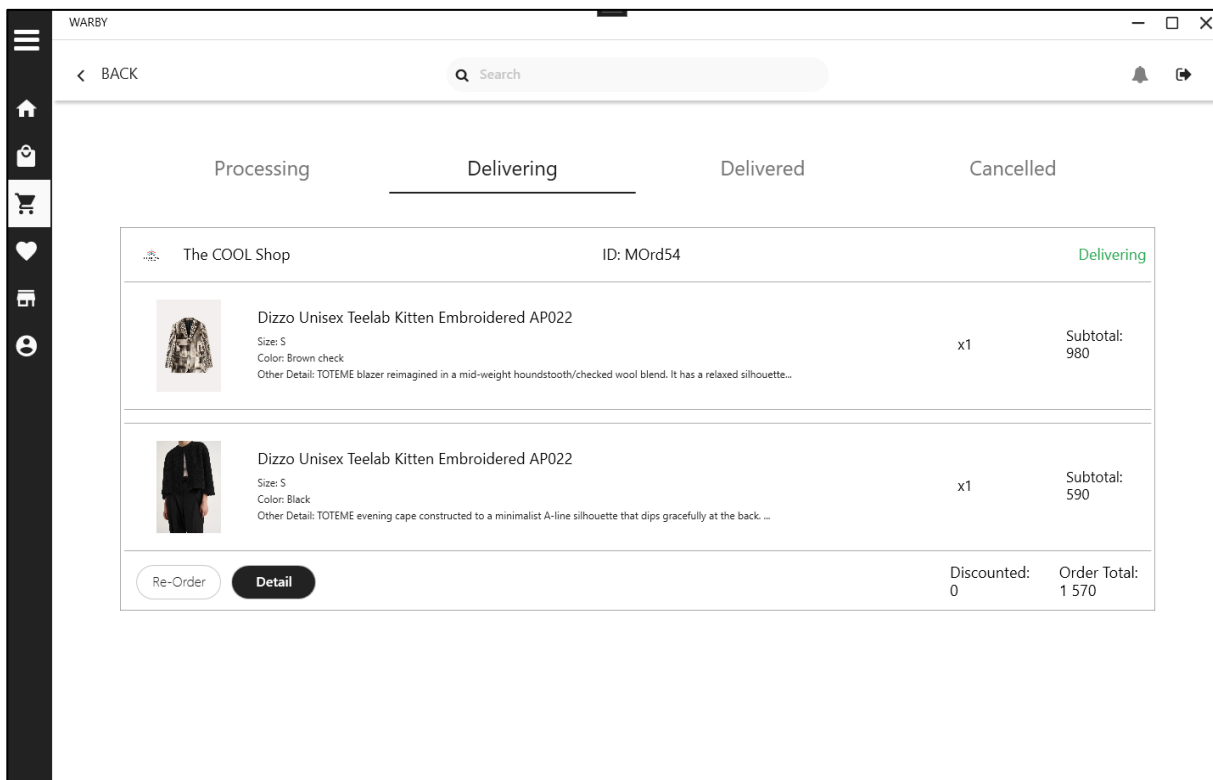


Hình 4.9 Giao diện Shop Bag

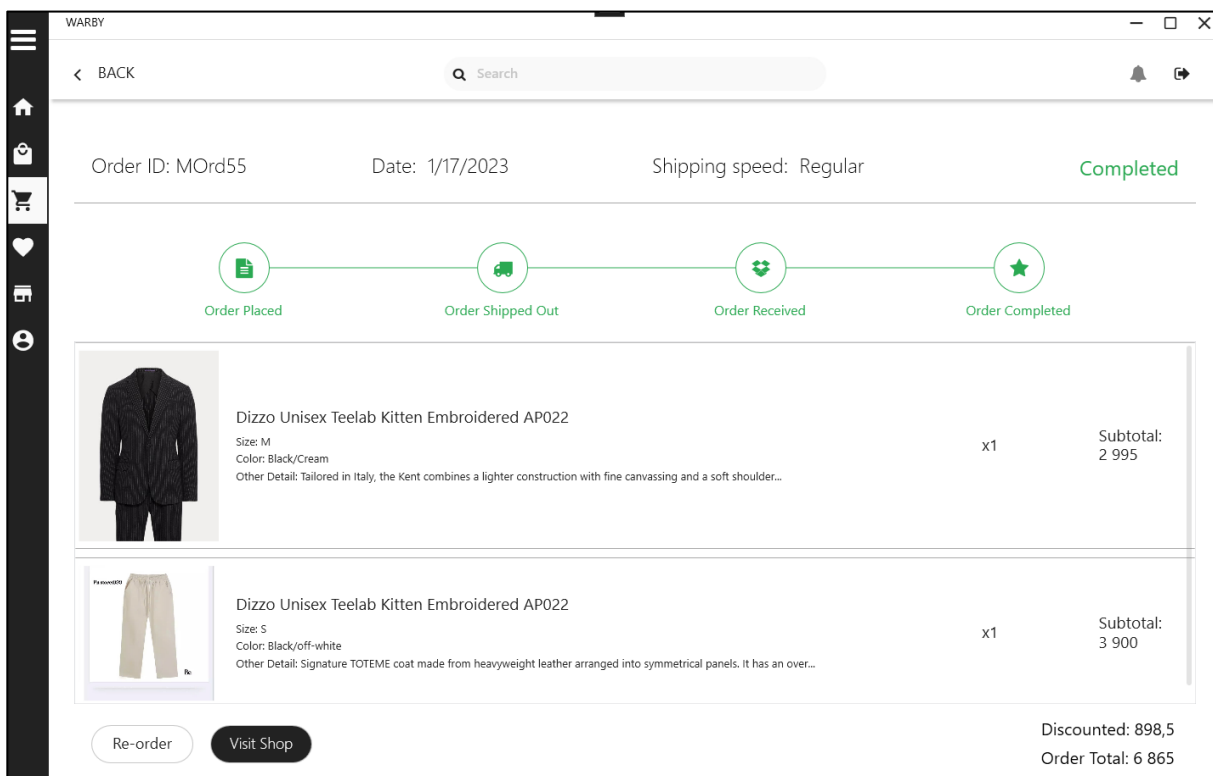


Hình 4. 10 Giao diện thanh toán

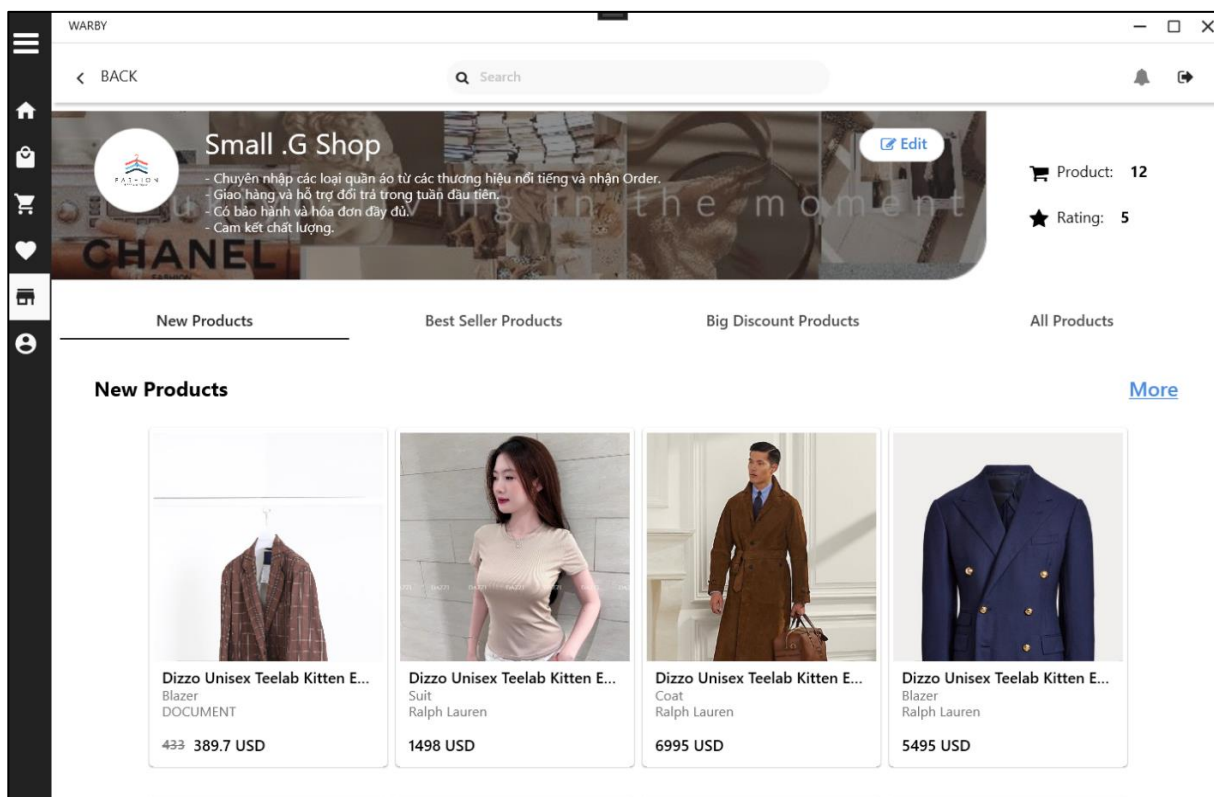




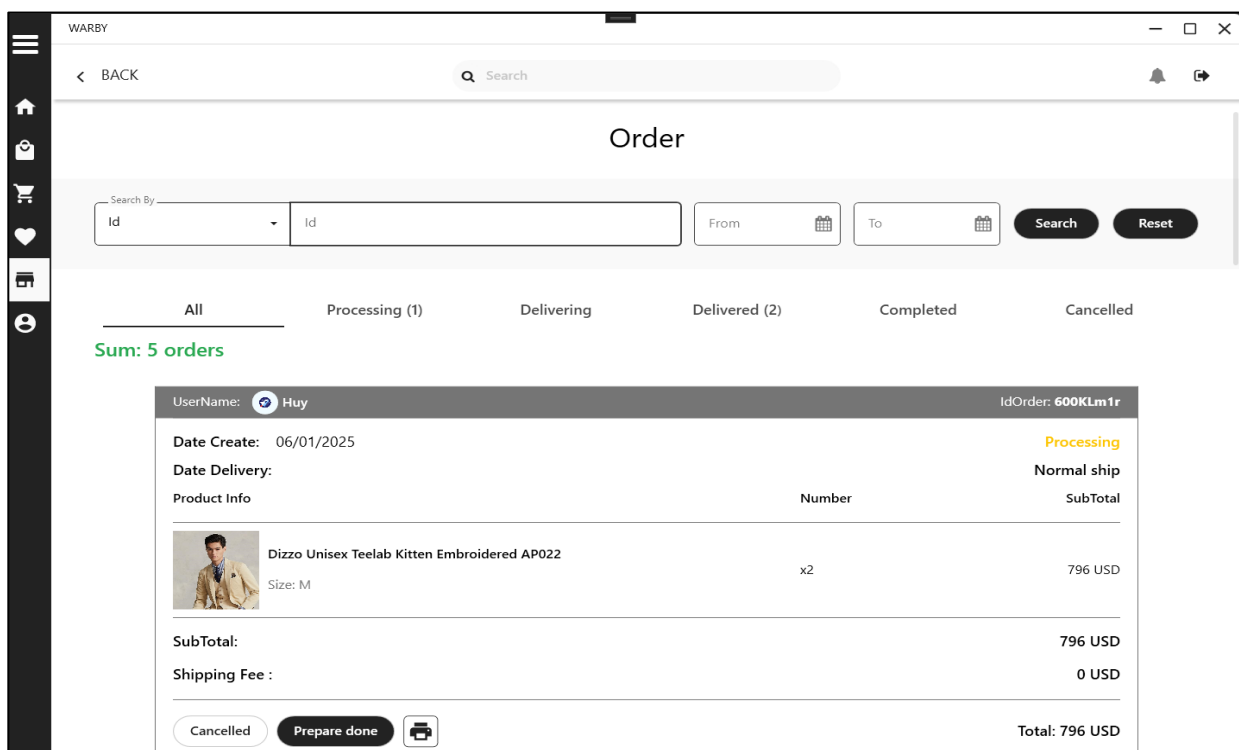
Hình 4.11 Giao diện giỏ hàng



Hình 4.12 Giao diện trạng thái đơn hàng



Hình 4.13 Giao diện hiển thị sản phẩm



Hình 4.14 Giao diện đơn hàng

WARBY

< BACK

Q Search

🔔

🔗

Product

+

Product Name

Price

Min

—

Max

Search

Category

Brand

Stock

Min

—

Max

Reset

All

On Sale

Out Of Stock

Banned

ID	Name	Category	Brand	Price	InStock	Sold	Status
prod54	Dizzo Unisex Teelab Kitten Embroidered AP022	Blazer	DOCUMENT	389,7 \$	123	0	On Sale
prod61	Dizzo Unisex Teelab Kitten Embroidered AP022	Suit	Ralph Lauren	1498 \$	131	0	On Sale
prod73	Dizzo Unisex Teelab Kitten Embroidered AP022	Coat	Ralph Lauren	6995 \$	9	0	On Sale
prod57	Dizzo Unisex Teelab Kitten Embroidered AP022	Blazer	Ralph Lauren	5495 \$	5	0	On Sale
prod66	Dizzo Unisex Teelab Kitten Embroidered AP022	Blazer	TOTEME	920 \$	3	0	On Sale
prod72	Dizzo Unisex Teelab Kitten Embroidered AP022	Coat	DOCUMENT	817 \$	9	0	On Sale
prod53	Dizzo Unisex Teelab Kitten Embroidered AP022	Coat	TOTEME	3900 \$	120	0	On Sale
prod60	Dizzo Unisex Teelab Kitten Embroidered AP022	Coat	TOTEME	1800 \$	35	0	On Sale
prod67	Dizzo Unisex Teelab Kitten	Coat	DOCUMENT	773 \$	135	1	On Sale

Hình 4.15 Giao diện quản lý thông tin sản phẩm

# Add new product

Product Name

0 / 200

Price

USD

Images

Change

Size ☐ Free Size

Size ▾

+ Add

Category ▾

+

In Stock

\*If you can't find the category, press button add new and we will update

Brand ▾

+

Sale %

\*If you can't find the brand, press button add new and we will update

Color

0 / 200

Description

0 / 1000




Add Product

Hình 4. 16 Giao diện thêm sản phẩm

# Dizzo Unisex Teelab Kitten Er

No rating yet

View product in the window

Change

Size

☐ Free Size

Price

Sale

In Stock

Sold

Category

Hình 4.17 Giao diện sửa sản phẩm

WARBY

BACK

Search

Shop Promo

Search By

Promo Code

Promo Code

From

To

Search

Reset

All




In process

Upcoming

Expired

Requesting

Deleted

Promo Code/Name	Target Customer	Sale	Num Pros	Used	Status	Action
 <b>NICEDAY</b> Have a nice day	New Customer	20 %	7	0 / +inf	Expired 1/11/2023 12:00:00 PM 3/31/2023 12:00:00 PM	
 <b>NEWYEAR2023</b> HAPPY NEW YEAR	New Customer	30 %	12	0 / +inf	Expired 1/15/2023 12:00:00 PM 3/31/2023 12:00:00 PM	
 <b>BIGBIG</b> ALMOST FREE	New Customer	99 %	12	0 / +inf	Expired 1/18/2023 12:00:00 PM 3/28/2023 12:00:00 PM	

Hình 4.18 Giao diện Promotion

WARBY

< BACK Search

### Promo Information

**Basic information promo**

Target customer: New customer All customers

Name 0 / 150 Code 0 / 15

From To

Description 0 / 200

**Set up promo**

Min Cost USD Sale %

Maximum reduction: Limited Unlimited

Hình 4.19 Giao diện thông tin Promo

WARBY

< BACK Search

### Rating

Product Name UserName Search

Category Brand From To Reset

**All** 5 Star 4 Star 3 Star 2 Star 1 Star

**The COOL Shop**

Dizzo Unisex Teelab Kitten Embroidered AP022

★★★★★

Awesome!!!  
I like it a lot.

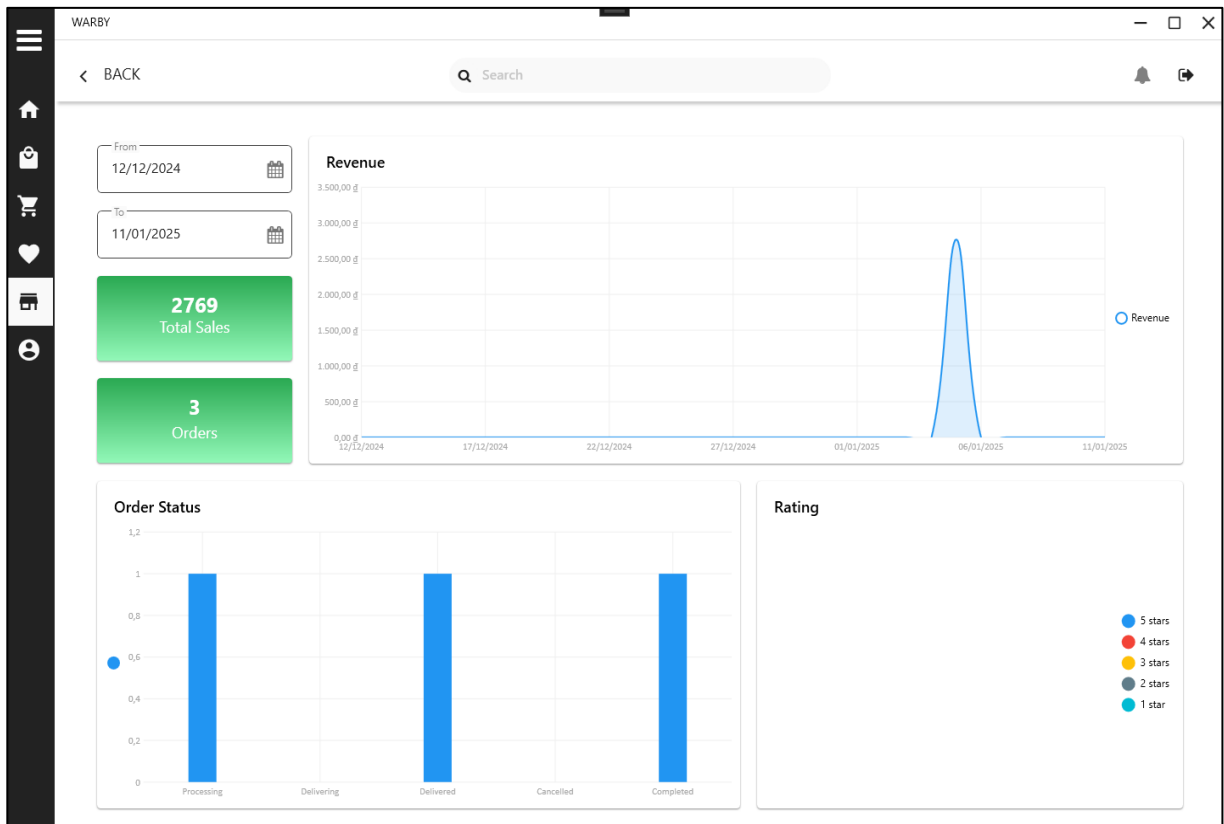
Size: M  
Date Rating: 17/01/2023

[Reply](#)

**Small .G Shop** 1/17/2023 9:07:34 AM

Thank you so much.

Hình 4.20 Giao diện Rating



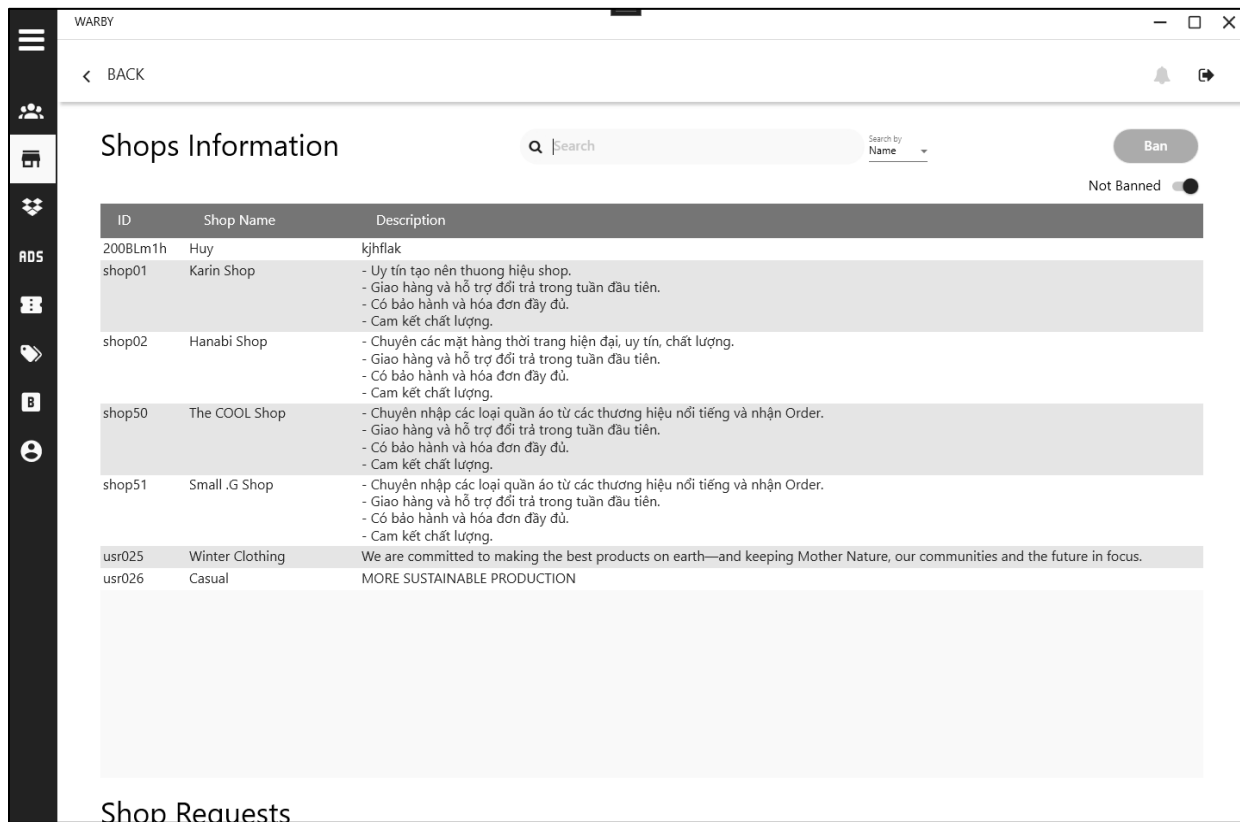
Hình 4.21 Giao diện thống kê

### 4.3. Giao diện Admin

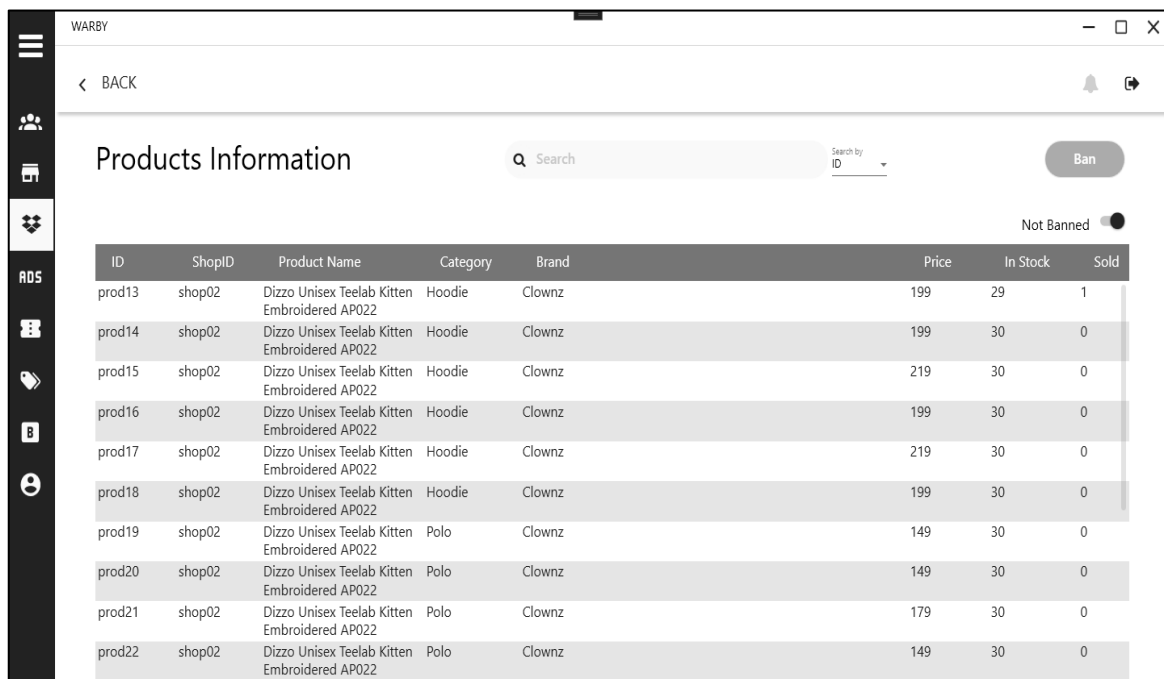
The screenshot displays the WARBY Admin interface. The top navigation bar includes a 'BACK' button and a search bar. Below this, there is a 'Users Information' section with a search bar and a 'Search by Name' dropdown. To the right of the search bar, there are 'Ban' and 'Add' buttons. Below the search bar, there is a 'Not Banned' toggle switch. The main content area is a table listing user information.

ID	User Name	Email	Phone	Birth	Gender
200BLm1h	Huy	6351071030@gmail.com	0123456		Male
shop01	Karin Shop	shop01@gmail.com	0123456789		Male
shop02	Hanabi Shop	shop02@gmail.com	0123456789		Male
shop50	The COOL Shop	yauboi@protonmail.com	0123456789		Male
shop51	Small .G Shop	hwangvu@protonmail.com	0123456789		Male
usr025	Winter Clothing	6351071052@st.utc2.edu.vn	0135798642		Female
usr026	Casual	lethimcook@gmail.com	0796789123		Female

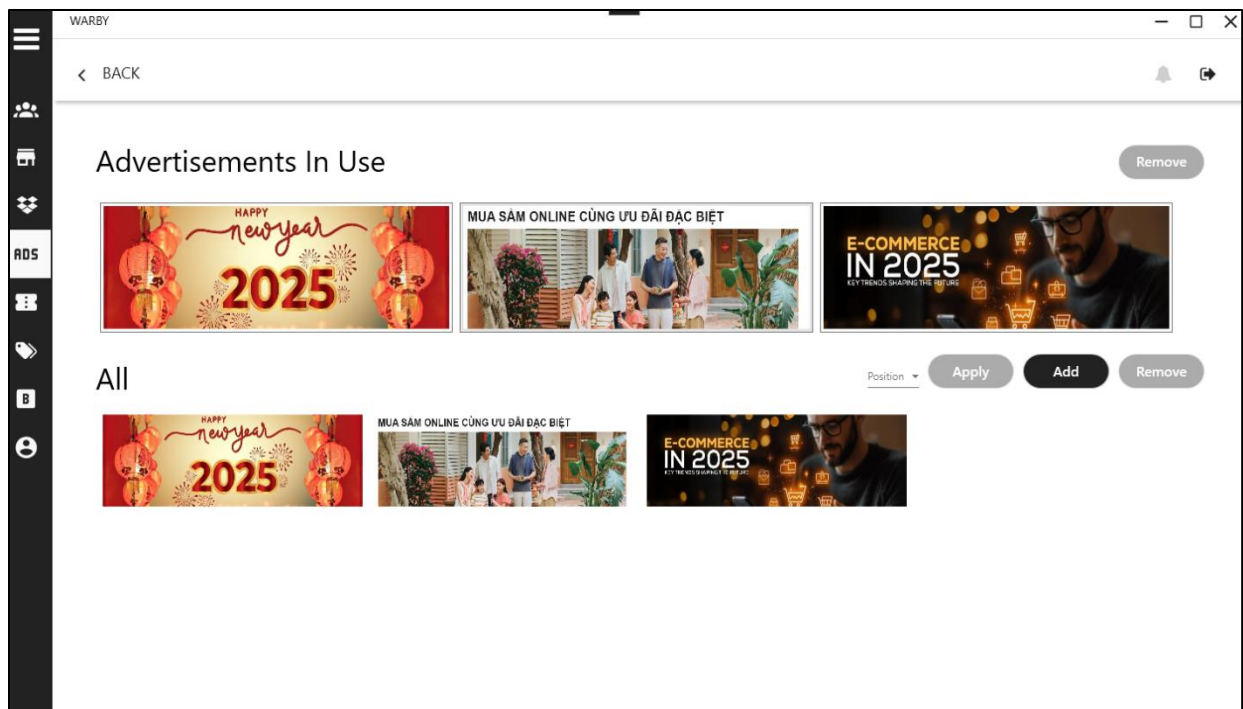
Hình 4.22 Giao diện thông tin User



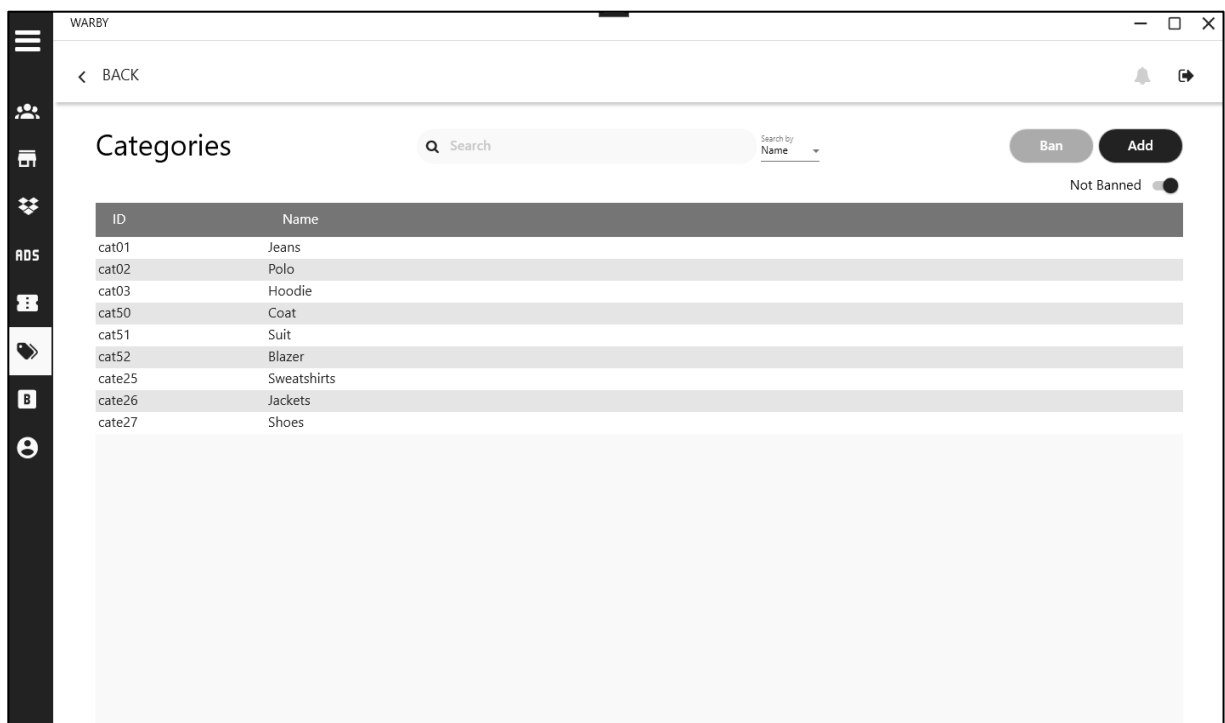
Hình 4.23 Giao diện thông tin Shop



Hình 4.24 Giao diện thông tin sản phẩm

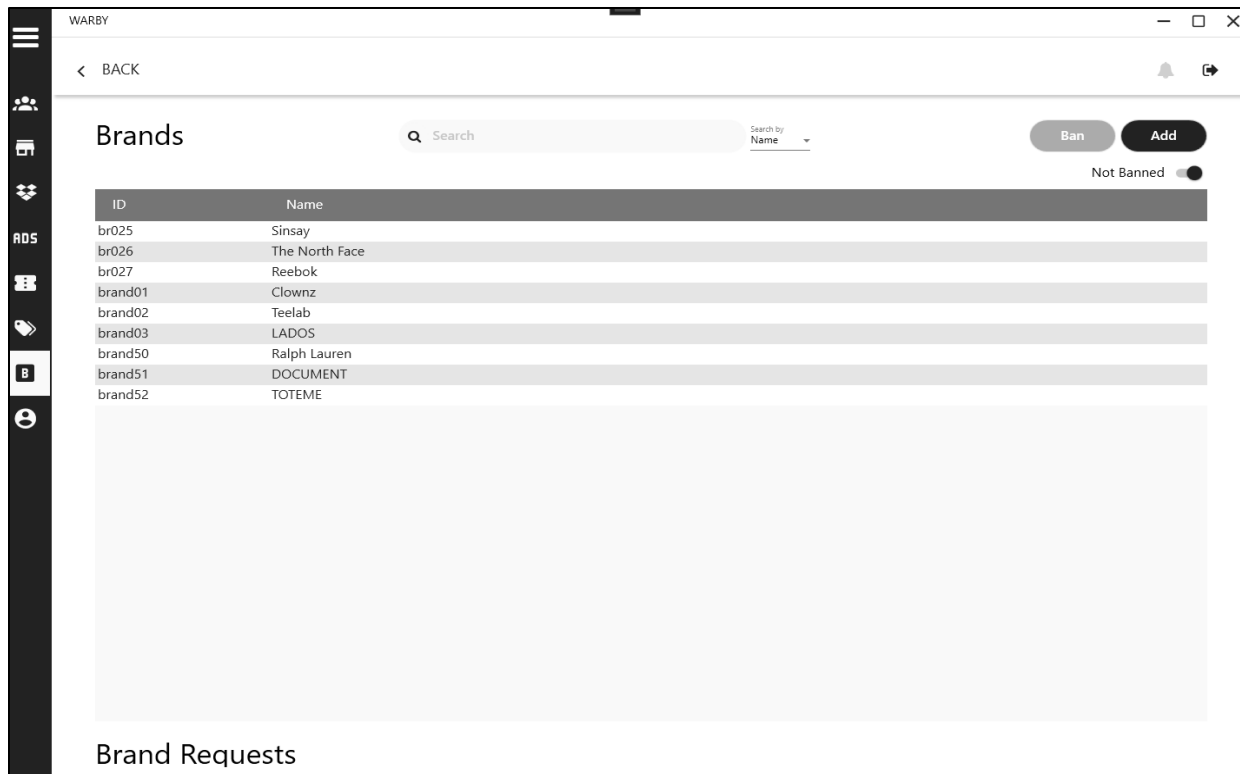


Hình 4.25 Giao diện thông tin quảng cáo

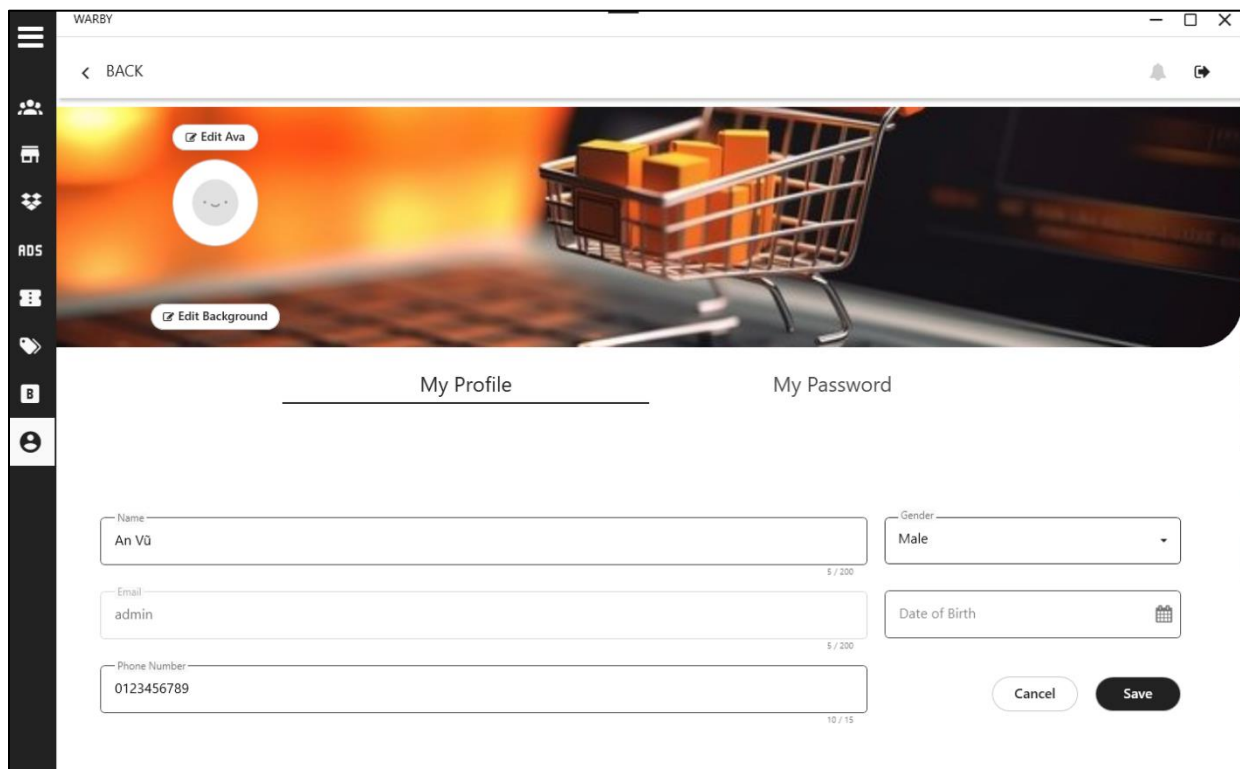


Hình 4.26 Giao diện thông tin loại sản phẩm





Hình 4.27 Giao diện thông tin thương hiệu



Hình 4.28 Giao diện thông tin Admin

# KẾT LUẬN VÀ KIẾN NGHỊ

## 1. Kết luận

Hệ thống Quản lý Cửa hàng Bán Quần Áo và Giày Dép là một giải pháp toàn diện được thiết kế nhằm nâng cao hiệu quả quản lý, tối ưu hóa quy trình kinh doanh, và mang lại sự tiện lợi cho cả nhân viên và khách hàng. Với việc tích hợp các tính năng hiện đại như quản lý sản phẩm, theo dõi đơn hàng, kiểm soát tồn kho, và phân tích dữ liệu, hệ thống không chỉ giúp tự động hóa các công việc thường nhật mà còn hỗ trợ ra quyết định một cách nhanh chóng và chính xác.

Nhờ sử dụng các công nghệ tiên tiến và thiết kế cơ sở dữ liệu khoa học, hệ thống đảm bảo khả năng xử lý dữ liệu chính xác, nhất quán và bảo mật. Đồng thời, giao diện người dùng được tối ưu hóa, dễ sử dụng ngay cả với những người không chuyên về công nghệ. Điều này giúp tăng cường trải nghiệm người dùng, giảm thiểu sai sót và tiết kiệm thời gian trong vận hành.

Hệ thống không chỉ dừng lại ở việc đáp ứng các yêu cầu cơ bản của một cửa hàng bán lẻ mà còn mở rộng với các tính năng vượt trội như quản lý khuyến mãi, đánh giá khách hàng, và báo cáo chi tiết. Đây là một công cụ mạnh mẽ, hỗ trợ doanh nghiệp nâng cao năng lực cạnh tranh, cải thiện dịch vụ khách hàng và gia tăng doanh thu.

Trong tương lai, hệ thống có tiềm năng được phát triển thêm để tích hợp với các nền tảng thương mại điện tử, quản lý giao hàng thông minh, và ứng dụng công nghệ AI trong phân tích dữ liệu. Điều này sẽ không chỉ mang lại giá trị cao hơn cho doanh nghiệp mà còn tạo ra những trải nghiệm mua sắm hiện đại, đáp ứng tốt hơn nhu cầu ngày càng cao của thị trường.

## 2. Hạn chế

Dự án WPF này có một số hạn chế. Đầu tiên, hiệu năng khởi động của ứng dụng có thể bị ảnh hưởng bởi việc thực hiện các tác vụ nền, chẳng hạn như khởi tạo dịch vụ DPIService và Internet trong quá trình khởi chạy. Điều này có thể dẫn đến thời gian tải chậm nếu không được tối ưu hóa. Thứ hai, ứng dụng phụ thuộc nhiều vào các thư viện bên thứ ba như Material Design và FontAwesome. Nếu các thư viện này ngừng hỗ trợ hoặc gặp lỗi, việc bảo trì và nâng cấp ứng dụng có thể trở nên khó khăn. Bên cạnh đó, dự án hiện chưa cung cấp đủ thông tin về các chức năng chính, chẳng hạn như quản lý sản phẩm, đơn hàng hoặc người dùng, khiến người dùng và nhà phát triển khó nắm bắt toàn bộ mục tiêu của ứng dụng. Cuối cùng, khả năng kiểm tra và gỡ lỗi có thể bị hạn chế nếu không có các lớp kiểm tra đơn vị hoặc giao diện riêng cho các dịch vụ.

## 3. Kiến nghị

*Tích Hợp Công Nghệ Web và Di Động:* Một hướng phát triển quan trọng là mở rộng hệ thống từ ứng dụng desktop sang ứng dụng web và di động. Bằng cách này, các quản lý và

nhân viên có thể truy cập hệ thống từ bất kỳ đâu và bất kỳ thiết bị nào, tăng tính tiện lợi và linh hoạt. Công nghệ web như React, Angular, hoặc Vue.js có thể được sử dụng cho giao diện người dùng, kết hợp với các dịch vụ back-end như Node.js hoặc Spring Boot. Đối với ứng dụng di động, Flutter hoặc React Native có thể là các lựa chọn tốt để xây dựng ứng dụng cho cả iOS và Android.

*Cải Thiện Trải Nghiệm Người Dùng (UX/UI):* Giao diện người dùng cần được thiết kế lại để trở nên thân thiện hơn, dễ sử dụng hơn. Việc cải thiện trải nghiệm người dùng giúp tăng hiệu suất làm việc của nhân viên và làm cho hệ thống trở nên hấp dẫn hơn đối với người dùng cuối. Sử dụng nguyên tắc thiết kế UX/UI hiện đại và tiến hành các thử nghiệm người dùng để liên tục cải thiện giao diện.

*Cải Thiện Hệ Thống Báo Cáo và Phân Tích Dữ Liệu:* Phát triển các tính năng báo cáo và phân tích dữ liệu nâng cao sẽ giúp doanh nghiệp có cái nhìn sâu sắc về hoạt động kinh doanh của mình. Các báo cáo có thể bao gồm doanh số bán hàng, hiệu quả của các chương trình giảm giá, tồn kho, và hiệu suất làm việc của nhân viên. Sử dụng các công cụ phân tích dữ liệu như Power BI hoặc Tableau có thể giúp tạo ra các báo cáo trực quan và dễ hiểu.

*Tăng Cường Bảo Mật Hệ Thống:* Vấn đề bảo mật luôn là một yếu tố quan trọng. Hệ thống cần được phát triển thêm các lớp bảo mật như xác thực hai yếu tố (2FA), mã hóa dữ liệu, và các cơ chế phát hiện và ngăn chặn xâm nhập. Điều này giúp bảo vệ dữ liệu nhạy cảm của doanh nghiệp và khách hàng khỏi các mối đe dọa bảo mật.

## TÀI LIỆU THAM KHẢO

- [1]. Andrew Troelsen & Philip Japikse, *Pro C# 10 and the .NET 6 Platform*, 1st edition, Apress, 2021.
- [2]. Pavan Podila & Kevin Hoffman, *WPF Control Development Unleashed: Building Advanced User Experiences*, 1st edition, Sams Publishing, 2009.
- [3]. Chris Andrade, Shawn Livermore, Mike Meyers, & Scott Van Vliet, *Professional WPF and C# Programming: Practical Software Development Using WPF and C#*, 1st edition, Wrox, 2010.
- [4]. Arlen Feldman & Maxx Daymon, *WPF in Action with Visual Studio 2008: Covers Visual Studio 2008 Service Pack 1 and .NET 3.5 Service Pack 1*, 1st edition, Manning Publications, 2009.
- [5]. Evangelos Petroutsos, *Mastering Windows Forms Development*, 1st edition, Sybex, 2005.
- [6]. Kunal Chowdhury, *Windows Presentation Foundation Development Cookbook*, 1st edition, Packt, 2018.
- [7]. *Programming WPF*, Chris Sells & Ian Griffiths, O'Reilly Media, 1st edition, 2007, truy cập tại [Amazon Learning](#).
- [8]. Microsoft Learn, “Windows Presentation Foundation Documentation,” <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/>, ngày truy cập 02 tháng 01 năm 2025.
- [9]. TutorialsPoint, “WPF Tutorial,” <https://www.tutorialspoint.com/wpf/index.htm>, ngày truy cập 02 tháng 01 năm 2025.
- [10]. Microsoft Learning, “WPF Documentation – .NET Desktop Guide,” <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/?view=netdesktop-9.0>, ngày truy cập 02 tháng 01 năm 2025.

