**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**Môn: LẬP TRÌNH TRỰC QUAN**

**ĐỀ TÀI: NỀN TẢNG BÁN HÀNG TÍCH HỢP:**

**ỨNG DỤNG MUA SẮM DÀNH CHO KHÁCH HÀNG**

**VÀ CÔNG CỤ QUẢN LÝ HIỆU QUẢ CHO NHÀ QUẢN TRỊ**

Giảng viên hướng dẫn: ThS. LƯU TOÀN ĐỊNH

Sinh viên thực hiện: PHẠM THỊ NGỌC OANH – 6351071055

ĐỖ VĂN THÀNH ĐƯỢC – 6351071021

TRẦN PHƯƠNG ANH – 6351071002

HÀ VĂN DŨNG – 6351071011

Lớp: CQ.63.CNTT

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 CUỐI KỲ**

**Môn: LẬP TRÌNH TRỰC QUAN**

**ĐỀ TÀI: NỀN TẢNG BÁN HÀNG TÍCH HỢP:**

**ỨNG DỤNG MUA SẮM DÀNH CHO KHÁCH HÀNG**

**VÀ CÔNG CỤ QUẢN LÝ HIỆU QUẢ CHO NHÀ QUẢN TRỊ**

Giảng viên hướng dẫn: ThS. LƯU TOÀN ĐỊNH

Sinh viên thực hiện: PHẠM THỊ NGỌC OANH – 6351071055

ĐỖ VĂN THÀNH ĐƯỢC – 6351071021

TRẦN PHƯƠNG ANH – 6351071002

HÀ VĂN DŨNG – 6351071011

Lớp: CQ.63.CNTT

Tp. Hồ Chí Minh, năm 2025

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI **CỘNG HÒA XÃ HỘI CHỦ NGHIÃ VIỆT NAM**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH** Độc lập – Tự do – Hạnh phúc

**NHIỆM VỤ BÁO CÁO ĐỒ ÁN MÔN HỌC**

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

-------\*\*\*-------

**Mã sinh viên:** 6351071055 **Họ tên SV:** PHẠM THỊ NGỌC OANH

**Khóa:** 63 **Lớp:** CÔNG NGHỆ THÔNG TIN

**Mã sinh viên:** 6351071021 **Họ tên SV:** ĐỖ VĂN THÀNH ĐƯỢC

**Khóa:** 63 **Lớp:** CÔNG NGHỆ THÔNG TIN

**Mã sinh viên:** 6351071002 **Họ tên SV:** TRẦN PHƯƠNG ANH

**Khóa:** 63 **Lớp:** CÔNG NGHỆ THÔNG TIN

**Mã sinh viên:** 6351071011 **Họ tên SV:** HÀ VĂN DŨNG

**Khóa:** 63 **Lớp:** CÔNG NGHỆ THÔNG TIN

1. **Mục đích, yêu cầu**

**1.1 Mục đích:**

* Tìm hiểu cách thức một ứng dụng window hoàn chỉnh hoạt động.
* Tăng cường hiểu biết, vận dụng linh hoạt các kiến thức đã học về .NET.
* Củng cố các kiến thức về lập trình desktop.
* Nắm bắt được cách tạo ra một ứng dụng hoàn chỉnh.
  1. **Yêu cầu:**

- Yêu cầu công cụ

+ Sử dụng ngôn ngữ lập trình C#.

+ Sử dụng công cụ Visual Studio 2019, Microsoft SQL Server.

+ Sử dụng thêm Guna Framework hỗ trợ thêm làm đẹp giao diện.

- Yêu cầu chức năng

+ Công nghệ: .NET Winform và .NET Core Web API

+ Hệ thống có 2 trang chính: quản trị viên và người dùng cuối.

+ Hệ thống cho phép quản trị viên quản lý cửa hàng online và người dụng có thể thao tác mua hàng một cách tiện lợi.

- Yêu cầu phi chức năng

+ Giao diện: Trực quan, dễ nhìn, dễ thao tác với hệ thống.

- Yêu cầu về tổ chức mã nguồn

+Tổ chức mã nguồn gọn gàng, sạch sẽ, logic chặt chẽ.

**2. Nội dung và phạm vi đề tài**

**a, Nội dung:**

- Tổng quan về công nghệ đang sử dụng.

- Phân tích và thiết kế phần mềm.

- Lập trình xây dựng phần mềm.

- Kiểm thử và chạy thực nghiệm.

- Kết quả thu được:

+ Xây dựng giao diện cho người quản trị và người dùng.

+ Hoàn thành hầu hết các chức năng:

**b, Phạm vi đề tài:** Thực hiện việc xây dựng một phần mềm quản lý cơ sở dữ liệu ở mức một cửa hàng bán online với số lượng đầu mục sản phẩm không quá lớn.

**3. Công nghệ, công cụ và ngôn ngữ lập trình**

* Ngôn ngữ lập trình: C# .NET
* Công nghệ: WindowsForms Application, SQL Server
* Công cụ: Visual Studio, Microsoft SQL Server Management Studio.

**4. Các kết quả chính dự kiến và đạt được ứng dụng**

* Quyển báo cáo.

- Hoàn thiện phần mềm với các chức năng cơ bản nêu trên.

**5. Giảng viên và cán bộ hướng dẫn**

**Họ tên: ThS. LƯU TOÀN ĐỊNH**

**Đơn vị công tác:** Bộ môn Công nghệ Thông tin – Trường Đại học Giao thông

Vận tải phân hiệu tại Thành phố Hồ Chí Minh.

**Điện thoại:**  **Email:**

|  |  |
| --- | --- |
| **Ngày 06 tháng 01 năm 2025**  **Trưởng BM Công nghệ Thông tin** | **Đã giao nhiệm vụ BCMH**  **Giảng viên hướng dẫn** |
| **ThS. Trần Phong Nhã** | **ThS. Lưu Toàn Định** |

Đã nhận nhiệm vụ BCMH

Sinh viên: PHẠM THỊ NGỌC OANH Ký tên:

Điện thoại: Email: [6351071055@st.utc2.edu.vn](mailto:6351071055@st.utc2.edu.vn)

Sinh viên: ĐỖ VĂN THÀNH ĐƯỢC Ký tên:

Điện thoại: Email: [6351071021@st.utc2.edu.vn](mailto:6351071021@st.utc2.edu.vn)

Sinh viên: TRẦN PHƯƠNG ANH Ký tên:

Điện thoại: Email: [6351071002@st.utc2.edu.vn](mailto:6351071002@st.utc2.edu.vn)

Sinh viên: HÀ VĂN DŨNG Ký tên:

Điện thoại: Email: [6351071011@st.utc2.edu.vn](mailto:6351071011@st.utc2.edu.vn)

# 

# LỜI CẢM ƠN

Để hoàn thành đề tài báo cáo cuối kỳ này, trước hết chúng em xin gửi đến quý thầy, cô Bộ môn Công nghệ thông tin – Phân hiệu Trường Đại học Giao thông vận tải tại Thành phố Hồ Chí Minh lời cảm ơn chân thành vì đã truyền đạt cho chúng em những kiến thức không chỉ từ sách vở, mà còn là những kinh nghiệm quý giá từ cuộc sống trong khoảng thời gian học tập tại trường. Đặc biệt chúng em xin gửi đến thầy Lưu Toàn Định lời cảm ơn sâu sắc nhất vì thầy đã tận tình hướng dẫn, chỉ bảo chúng em trong suốt quá trình thực hiện đề tài. Tuy đề tài không được lớn nhưng nếu không có sự hướng dẫn chỉ bảo tận tình của thầy thì bài tập này khó có thể hoàn thành được.

Vì thời gian làm đồ án cuối kỳ không nhiều cũng như hiểu biết của nhóm còn hạn chế, chúng em cũng đã nỗ lực hết sức để hoàn thành bài báo cáo một cách tốt nhất, nhưng chắc chắn vẫn sẽ có những thiếu sót không thể tránh khỏi. Chúng em kính mong nhận được sự thông cảm và những ý kiến đóng góp chân thành từ quý thầy cô.

Sau cùng, em xin kính chúc quý Thầy Cô trong Bộ môn Công nghệ thông tin luôn mạnh khoẻ, hạnh phúc và thành công hơn nữa trong công việc cũng như trong cuộc sống. Chúng em xin chân thành cảm ơn

***Thành phố Hồ Chí Minh, ngày 06 tháng 01 năm 2025***

**Sinh viên thực hiện**

**Phạm Thị Ngọc Oanh**

**Đỗ Văn Thành Được**

**Trần Phương Anh**

**Hà Văn Dũng**

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

|  |
| --- |
| ***Tp. Hồ Chí Minh, ngày … tháng … năm 2025***  **Giảng viên hướng dẫn**  **ThS. Lưu Toàn Định** |

**BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ và tên** | **Các công việc thực hiện** | **Tỉ lệ khối**  **lượng công việc** | **Ký tên** |
| **Trần Phương Anh** | - Tham gia vào quá trình phân tích và thiết kế dự án.  - Thiết kế giao diện và chức năng của các form admin. | 25% |  |
| **Phạm Thị Ngọc Oanh** | - Tham gia vào quá trình phân tích và thiết kế dự án.  - Thiết kế giao diện và chức năng của các form customer. | 25% |  |
| **Đỗ Văn Thành Được** | - Tham gia vào quá trình phân tích và thiết kế dự án.  - Triển khai api vào dự án winfrom.  - Tham gia thiết kế giao diện và chức năng của form customer. | 25% |  |
| **Hà Văn Dũng** | - Tham gia vào quá trình phân tích và thiết kế dự án.  - Viết api cho dự án.  - Tham gia thiết kế giao diện của các form. | 25% |  |

**MỤC LỤC**

[LỜI CẢM ƠN i](#_Toc187363663)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN ii](#_Toc187363664)

[LỜI MỞ ĐẦU vi](#_Toc187363665)

[DANH MỤC CHỮ VIẾT TẮT vii](#_Toc187363666)

[DANH MỤC HÌNH ẢNH viii](#_Toc187363667)

[CHƯƠNG 1: TỔNG QUAN 1](#_Toc187363668)

[1.1. Đặt vấn đề 1](#_Toc187363669)

[1.2. Về quản lý ứng dụng mua sắm 1](#_Toc187363670)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 3](#_Toc187363671)

[2.1. Tổng quan về .NET Winform 3](#_Toc187363672)

[2.2. Tổng quan về .NET Core API 5](#_Toc187363673)

[2.3 Lập trình API 7](#_Toc187363674)

[2.4. Tổng quan về Guna UI 2 9](#_Toc187363675)

[2.5. Tổng quan về SQL Server 10](#_Toc187363676)

[CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG 12](#_Toc187363677)

[3.1 Mô tả bài toán 12](#_Toc187363678)

[3.2. Chức năng hệ thống 13](#_Toc187363679)

[CHƯƠNG 4: THIẾT KẾ HỆ THỐNG 17](#_Toc187363680)

[4.1 Thiết kế cơ sở dữ liệu 17](#_Toc187363681)

[4.2 Thiết kế API 37](#_Toc187363682)

[4.3. Thiết kế giao diện chính 44](#_Toc187363683)

[KẾT LUẬN VÀ KIẾN NGHỊ 59](#_Toc187363684)

[TÀI LIỆU THAM KHẢO 60](#_Toc187363685)

# 

# LỜI MỞ ĐẦU

Trong bối cảnh công nghệ số phát triển mạnh mẽ, các ứng dụng mua sắm trực tuyến ngày càng trở thành một phần không thể thiếu trong đời sống hàng ngày. Người dùng không chỉ mong muốn các ứng dụng này mang lại sự tiện lợi mà còn phải đáp ứng nhu cầu cá nhân hóa, tốc độ xử lý nhanh và tính bảo mật cao.

Hệ thống được thiết kế nhằm đáp ứng nhu cầu của cả người bán và người mua, tạo ra một nền tảng giao dịch hiệu quả, tiện lợi và an toàn. Hy vọng rằng, với hệ thống được xây dựng, người dùng có thể trải nghiệm một nền tảng mua sắm hiện đại, đồng thời mở ra nhiều cơ hội phát triển cho các doanh nghiệp bán hàng trực tuyến.

Em xin phép chọn đề tài xây dựng và thiết kế phần mềm quản lý ứng dụng mua sắm cho đồ án môn học. Vì kinh phí cũng như thời gian hạn hẹp nên các chức năng của phần mềm chưa được được hoàn thiện một cách đầy đủ nhất so với các điều kiện thực tế. Em hi vọng nếu có điều kiện đầy đủ, các chức năng cũng như giao diện có thể đầy đủ hơn và có thể áp dụng vào các yêu cầu thực tế.

Em xin chân thành cảm ơn thầy Lưu Toàn Định đã truyền đạt cho em rất nhiều kiến thức về môn học và sự hướng dẫn, tư vấn nhiệt tình của các cô để chúng em có thể hoàn thành đồ án một cách tốt nhất.

# DANH MỤC CHỮ VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mô tả** | **Ý nghĩa** | **Ghi chú** |
| 1 | GUI | Graphical User Interface |  |
| 2 | .NET | .NET Framework |  |
| 3 | API | Application Programming Interface |  |
| 4 | CRUD | Create, Read, Update, Delete |  |
| 5 | SQL | Structured Query Language |  |
| 6 | MSSQL | Microsoft SQL Server |  |
| 7 | XML | eXtensible Markup Language |  |
| 8 | ETL | Extract, Transform, Load |  |
| 9 | MDX | |  | | --- | |  |  |  | | --- | | MultiDimensional Expressions | |  |

# DANH MỤC HÌNH ẢNH

[*Hình 4.1 Sơ đồ quan hệ 33*](#_Toc187358897)

[*Hình 4.2 Mô hình BFD 35*](#_Toc187358898)

[*Hình 4.3 Mô hình DFD mức ngữ cảnh 36*](#_Toc187358899)

[*Hình 4.4 Mô hình DFD mức 0 36*](#_Toc187358900)

[*Hình 4.5 Mô tả chi tiết endpoint 37*](#_Toc187358901)

[*Hình 4.6 Các phương thức của Account 38*](#_Toc187358902)

[*Hình 4.7 Các phương thức của Cart 39*](#_Toc187358903)

[*Hình 4.8 Các phương thức của Categories 39*](#_Toc187358904)

[*Hình 4.9 Các phương thức của Colors 39*](#_Toc187358905)

[*Hình 4.10 Các phương thức của Discounts 40*](#_Toc187358906)

[*Hình 4.11 Các phương thức của Feedback 40*](#_Toc187358907)

[*Hình 4.12 Các phương thức của Images 40*](#_Toc187358908)

[*Hình 4.13 Các phương thức của Order 41*](#_Toc187358909)

[*Hình 4.14 Các phương thức của Payment 41*](#_Toc187358910)

[*Hình 4.15 Các phương thức của Permission 41*](#_Toc187358911)

[*Hình 4.16 Các phương thức của Product 42*](#_Toc187358912)

[*Hình 4.17 Các phương thức của Role 42*](#_Toc187358913)

[*Hình 4.18 Các phương thức của Sizes 42*](#_Toc187358914)

[*Hình 4.19 Các phương thức của User 43*](#_Toc187358915)

[*Hình 4.20 Các phương thức của Voucher 43*](#_Toc187358916)

[*Hình 4.21 Giao diện đăng nhập 44*](#_Toc187358917)

[*Hình 4.22 Giao diện đăng ký 45*](#_Toc187358918)

[*Hình 4.23 Giao diện quên mật khẩu 46*](#_Toc187358919)

[*Hình 4.24 Giao diện đổi mật khẩu 46*](#_Toc187358920)

[*Hình 4.25 Giao diện trang chủ admin 47*](#_Toc187358921)

[*Hình 4.26 Giao diện theo dõi danh sách sản phẩm 48*](#_Toc187358922)

[*Hình 4.27 Giao diện thêm sản phẩm 48*](#_Toc187358923)

[*Hình 4.28 Giao diện chi tiết đơn hàng 49*](#_Toc187358924)

[*Hình 4.29 Giao diện chi tiết khách hàng 49*](#_Toc187358925)

[*Hình 4.30 Giao diện xác thực 2 bước 50*](#_Toc187358926)

[*Hình 4.31 Giao diện hóa đơn khách hàng 50*](#_Toc187358927)

[*Hình 4.32 Gửi yêu cầu in riêng 51*](#_Toc187358928)

[*Hình 4.33 Cài đặt tài khoản 52*](#_Toc187358929)

[*Hình 4.34 Trang giới thiệu 53*](#_Toc187358930)

[*Hình 4.35 Kho voucher của shop 53*](#_Toc187358931)

[*Hình 4.36 Chi tiết sản phẩm 54*](#_Toc187358932)

[*Hình 4.37 Giỏ hàng 55*](#_Toc187358933)

[*Hình 4.38 Xem sản phẩm theo danh mục 56*](#_Toc187358934)

[*Hình 4.39 Giao diện chọn voucher 57*](#_Toc187358935)

[*Hình 4.40 Giao diện trang đặt hàng 58*](#_Toc187358936)

# TỔNG QUAN

## 1.1. Đặt vấn đề

Sự phát triển vượt bậc của công nghệ thông tin đã và đang làm thay đổi toàn diện cách con người sinh sống và làm việc. Các ngành nghề từ kinh tế, y tế, giáo dục đến công nghiệp đều đang tận dụng công nghệ để nâng cao hiệu suất, độ chính xác và trải nghiệm người dùng. Trong lĩnh vực thương mại, công nghệ không chỉ tạo ra những công cụ mạnh mẽ để quản lý mà còn định hình lại cách thức mua sắm của con người, đặc biệt là qua các nền tảng trực tuyến.

Mua sắm trực tuyến đã trở thành xu hướng toàn cầu nhờ sự tiện lợi, nhanh chóng và đa dạng lựa chọn. Người tiêu dùng ngày càng ưa chuộng việc chọn mua sản phẩm từ xa, tiết kiệm thời gian và nhận hàng tận nơi. Tuy nhiên, đi kèm với sự phát triển này là nhu cầu ngày càng cao về việc quản lý các ứng dụng mua sắm một cách hiệu quả, cả từ góc độ khách hàng lẫn người vận hành.

Đối với khách hàng, các ứng dụng mua sắm cần phải đảm bảo giao diện thân thiện, tính năng phong phú và trải nghiệm mượt mà. Người dùng mong muốn dễ dàng tìm kiếm sản phẩm, theo dõi đơn hàng, nhận thông báo về ưu đãi và thanh toán thuận tiện. Mặt khác, quản trị viên của các ứng dụng cần công cụ quản lý mạnh mẽ để xử lý kho hàng, đơn hàng, khách hàng và các chiến dịch khuyến mãi.

Hiện nay, nhiều hệ thống quản lý ứng dụng mua sắm đang gặp phải những vấn đề về hiệu quả vận hành, giao diện phức tạp, hoặc thiếu các tính năng phân tích, thống kê. Vì vậy, việc xây dựng một hệ thống quản lý ứng dụng mua sắm toàn diện, đáp ứng được nhu cầu của cả khách hàng và quản trị viên, là điều vô cùng cần thiết để nâng cao hiệu quả và thúc đẩy trải nghiệm mua sắm.

## 1.2. Về quản lý ứng dụng mua sắm

Một ứng dụng mua sắm điển hình cần tích hợp nhiều chức năng đáp ứng cả người dùng cuối (khách hàng) và nhà cung cấp (quản trị viên). Đối với khách hàng, các chức năng phổ biến bao gồm tìm kiếm và xem sản phẩm, thêm sản phẩm vào giỏ hàng, thanh toán, theo dõi đơn hàng và quản lý thông tin tài khoản cá nhân. Những tính năng này không chỉ giúp cải thiện trải nghiệm mua sắm mà còn tạo sự thuận tiện trong việc đánh giá, ghi nhận lịch sử giao dịch và nhận các thông báo ưu đãi.

Từ phía quản trị viên, hệ thống phải cho phép quản lý sản phẩm, danh mục hàng hóa, đơn hàng và các hoạt động khuyến mãi. Đồng thời, khả năng thống kê và báo cáo doanh thu, theo dõi hiệu suất bán hàng và đánh giá phản hồi từ khách hàng cũng là những yếu tố quan trọng để hỗ trợ quá trình ra quyết định.

# CƠ SỞ LÝ THUYẾT

## 2.1. Tổng quan về .NET Winform

**2.1.1. Khái niệm**

**WinForms (Windows Forms)** là một phần của .NET Framework và .NET. Đây là nền tảng phát triển ứng dụng giao diện người dùng (GUI) cho Windows. WinForms sử dụng mô hình lập trình hướng sự kiện, cho phép tạo các ứng dụng desktop với giao diện thân thiện, dễ sử dụng. Với những thành phần chính:

* **Form**: Là cửa sổ chính hoặc cửa sổ con trong ứng dụng.
* **Control**: Các thành phần giao diện như Button, Label, TextBox, ComboBox, ListBox, DataGridView, v.v.
* **Event**: Các hành động người dùng hoặc hệ thống (như nhấn nút, nhập liệu, đóng form) kích hoạt một sự kiện để thực thi code.

**2.1.2. Một số câu lệnh dùng nhiều**

1. Tạo và hiển thị form

|  |
| --- |
| Form myForm = new Form();  myForm.Text = "Hello WinForms!";  myForm.Show(); |

1. Gán tiêu đề và kích thước cho form

|  |
| --- |
| this.Text = "Ứng dụng WinForms";  this.Size = new Size(800, 600); |

1. Thêm Control vào Form

Button btnClickMe = new Button();

btnClickMe.Text = "Click Me!";

btnClickMe.Location = new Point(50, 50);

btnClickMe.Click += BtnClickMe\_Click;

this.Controls.Add(btnClickMe);

1. Sự kiện (Event)

private void BtnClickMe\_Click(object sender, EventArgs e)

{

    MessageBox.Show("Bạn đã nhấn nút!");

}

1. Hiển thị danh sách trong ListBox

ListBox listBox = new ListBox();

listBox.Items.Add("Item 1");

listBox.Items.Add("Item 2");

this.Controls.Add(listBox);

1. Lấy dữ liệu từ TextBox

TextBox txtInput = new TextBox();

this.Controls.Add(txtInput);

Button btnSubmit = new Button();

btnSubmit.Text = "Submit";

btnSubmit.Click += (sender, e) =>

{

    string userInput = txtInput.Text;

    MessageBox.Show($"Bạn đã nhập: {userInput}");

};

this.Controls.Add(btnSubmit);

1. Kết nối và hiển thị dữ liệu trong DataGridView

DataGridView dataGridView = new DataGridView();

var data = new List<dynamic>

{

    new { ID = 1, Name = "How to become Millionaire", Price = 100 },

    new { ID = 2, Name = "YOLO", Price = 200 }

};

dataGridView.DataSource = data;

this.Controls.Add(dataGridView);

1. Đóng Form

this.Close();

## 2.2. Tổng quan về .NET Core API

**2.2.1. Khái niệm**

**.NET Core API** là một phần của nền tảng .NET Core, được thiết kế để phát triển các dịch vụ RESTful APIs hiệu quả, nhanh chóng và đa nền tảng.

1. .NET Core API có thể chạy trên Windows, macOS, và Linux, giúp dễ dàng triển khai và vận hành trên nhiều môi trường khác nhau.
2. .NET Core được tối ưu hóa cho hiệu suất cao, thích hợp cho các ứng dụng web với lượng truy cập lớn.
3. Sử dụng kiến trúc dựa trên các gói (packages), chỉ thêm các thành phần cần thiết, giúp giảm kích thước và tăng tính linh hoạt.
4. Hỗ trợ pipeline linh hoạt thông qua middleware, giúp dễ dàng tùy chỉnh quy trình xử lý yêu cầu.

**2.2.2. Cấu trúc của .NET Core API**

1. Controllers: Xử lý các yêu cầu HTTP từ client và trả về phản hồi

[ApiController]

[Route("api/[controller]")]

public class WeatherController : ControllerBase

{

    [HttpGet]

    public IActionResult GetWeather()

    {

        return Ok(new { Temperature = 25, Condition = "Sunny" });

    }

}

1. Models: Đại diện cho dữ liệu của ứng dụng, thường được sử dụng để gửi và nhận thông tin giữa client và server.

public class WeatherForecast

{

    public DateTime Date { get; set; }

    public int TemperatureC { get; set; }

    public string Summary { get; set; }

}

1. Services: Xử lý logic nghiệp vụ, tách biệt khỏi controller.

public interface IWeatherService

{

    IEnumerable<WeatherForecast> GetForecasts();

}

public class WeatherService : IWeatherService

{

    public IEnumerable<WeatherForecast> GetForecasts()

    {

        return new List<WeatherForecast>

        {

            new WeatherForecast { Date = DateTime.Now, TemperatureC = 25, Summary = "Sunny" },

            new WeatherForecast { Date = DateTime.Now.AddDays(1), TemperatureC = 22, Summary = "Cloudy" }

        };

    }

}

1. Middleware: Quản lý pineline xử lý yêu cầu HTTP (logging, authentication,..)

public class LoggingMiddleware

{

    private readonly RequestDelegate \_next;

    public LoggingMiddleware(RequestDelegate next)

    {

        \_next = next;

    }

    public async Task Invoke(HttpContext context)

    {

        Console.WriteLine($"Request: {context.Request.Method} {context.Request.Path}");

        await \_next(context);

    }

}

## 2.3 Lập trình API

Lập trình API là một phần quan trọng trong việc xây dựng các ứng dụng hiện đại. Với .NET Core API, nó cung cấp một nền tảng mạnh mẽ và linh hoạt để tạo ra các API nhẹ, hiệu quả, và dễ bảo trì. Có thể nói API như là cầu nối giữa các hệ thống phần mềm, cho phép chúng giao tiếp và trao đổi dữ liệu với nhau.

1. Kiến trúc cơ bản

WinForm (Client):

* Gửi các yêu cầu (request) tới API.
* Nhận và xử lý dữ liệu trả về từ API (response).

API (Server):

* Xử lý logic nghiệp vụ và truy xuất dữ liệu.
* Gửi dữ liệu trả về cho WinForm.

1. Các bước triển khai

Tạo API:

* Sử dụng .NET Core Web API để xây dựng API backend.
* Đảm bảo các endpoint cung cấp đầy đủ chức năng mà ứng dụng WinForm cần (CRUD, xử lý logic nghiệp vụ).

Kết nối từ WinForm đến API:

* Dùng HttpClient để gửi và nhận dữ liệu từ API.

using System.Net.Http;

using System.Text.Json;

public async Task GetDataFromApi()

{

    using (HttpClient client = new HttpClient())

    {

        client.BaseAddress = new Uri("http://localhost:5000/api/");

        HttpResponseMessage response = await client.GetAsync("products");

        if (response.IsSuccessStatusCode)

        {

            var data = await response.Content.ReadAsStringAsync();

            var products = JsonSerializer.Deserialize<List<Product>>(data);

            // Xử lý dữ liệu

        }

    }

}

Xử lý dữ liệu và giao diện:

* Kết nối dữ liệu từ API tới các điều khiển trên WinForm như DataGridView, ComboBox.
* Cập nhật giao diện sau khi nhận dữ liệu.

## 2.4. Tổng quan về Guna UI 2

**Guna UI 2** là một bộ thư viện giao diện người dùng (UI) dành cho ứng dụng WinForms, được thiết kế để nâng cấp giao diện và trải nghiệm người dùng của các ứng dụng desktop. Thư viện này cung cấp nhiều thành phần giao diện hiện đại và tùy chỉnh cao, phù hợp với các tiêu chuẩn thiết kế hiện nay.

**2.4.1 Cài đặt Guna UI 2**

Sử dụng **NuGet Package Manager** trong Visual Studio:

1. Mở Visual Studio.
2. Chọn **Tools** > **NuGet Package Manager** > **Manage NuGet Packages for Solution**.
3. Tìm kiếm **Guna.UI2.WinForms** và nhấn **Install**.

**2.4.2 Đa dạng Control**

|  |  |
| --- | --- |
| **Control** | **Mô tả** |
| gG  Guna2 Button | Nút bấm với thiết kế hiện đại, có các hiệu ứng hover, click và bóng đổ. |
| Guna2TextBox | TextBox hỗ trợ placeholder và biểu tượng (icon) bên trong. |
| Guna2ComboBox | ComboBox tùy chỉnh giao diện, hỗ trợ dropdown hiện đại. |
| Guna2ProgressBar | ProgressBar với màu sắc gradient và hiệu ứng mượt mà. |
| Guna2DataGridView | DataGridView cải tiến với giao diện đẹp, hỗ trợ hiển thị dữ liệu tốt hơn. |
| Guna2ShadowPanel | Panel với hiệu ứng bóng đổ, giúp phân vùng nội dung rõ ràng hơn. |
| Guna2CircleButton | Nút bấm dạng tròn, phù hợp cho các ứng dụng yêu cầu thiết kế sáng tạo. |
| Guna2ToggleSwitch | Nút chuyển trạng thái (on/off) hiện đại, thay cho CheckBox truyền thống. |

## 2.5. 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, Intergration 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).

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.

# PHÂN TÍCH HỆ THỐNG

## 3.1 Mô tả bài toán

**3.1.1. Quy trình mua sắm và quản lý đơn hàng**

Khách hàng bắt đầu bằng việc đăng ký hoặc đăng nhập vào tài khoản cá nhân. Sau khi đăng nhập, họ có thể cập nhật thông tin cá nhân, địa chỉ giao hàng và phương thức thanh toán. Tiếp theo, khách hàng duyệt qua các sản phẩm, tìm kiếm những sản phẩm ưng ý và xem đánh giá từ những người mua trước để đưa ra quyết định lựa chọn.

Khi đã chọn được sản phẩm, khách hàng thêm vào giỏ hàng với các thông tin chi tiết như tên sản phẩm, số lượng, kích cỡ, màu sắc và giá tiền. Sau đó, họ sẽ kiểm tra lại giỏ hàng để đảm bảo các sản phẩm đã chọn đúng, chỉnh sửa số lượng hoặc loại bỏ sản phẩm không cần thiết.

Sau khi chắc chắn về lựa chọn của mình, khách hàng tiến hành thanh toán bằng phương thức phù hợp như thanh toán khi nhận hàng (COD). Khi quá trình thanh toán hoàn tất, hệ thống ghi nhận đơn hàng và gửi thông báo xác nhận cho khách hàng.

Sau khi thanh toán, khách hàng có thể theo dõi trạng thái đơn hàng qua ứng dụng, từ khi đơn hàng được đặt, đang xử lý, vận chuyển cho đến khi giao hàng thành công. Khi nhận hàng, khách hàng kiểm tra sản phẩm và có thể gửi đánh giá về sản phẩm để giúp những người mua sau tham khảo.

Mọi giao dịch mua hàng của khách hàng được lưu trữ trong lịch sử mua hàng, giúp họ dễ dàng xem lại các đơn hàng đã đặt và theo dõi các sản phẩm đã mua trong quá khứ. Đồng thời, khách hàng sẽ nhận được thông báo về các chương trình khuyến mãi, cập nhật trạng thái đơn hàng hoặc những thông tin quan trọng khác.

**3.1.2. Quy trình quản trị hệ thống**

Về phía quản trị viên, họ sẽ quản lý các sản phẩm và danh mục sản phẩm, thêm mới, chỉnh sửa hoặc xóa sản phẩm, đảm bảo rằng các sản phẩm luôn được cập nhật chính xác và dễ dàng tìm kiếm. Quản trị viên cũng có thể tạo, chỉnh sửa hoặc xóa các danh mục sản phẩm để giúp khách hàng duyệt sản phẩm hiệu quả hơn.

Bên cạnh đó, quản trị viên theo dõi danh sách đơn hàng, bao gồm thông tin khách hàng, sản phẩm và phương thức thanh toán. Các trạng thái của đơn hàng cũng sẽ được cập nhật thường xuyên để thông báo cho khách hàng biết tình trạng của đơn hàng. Thông qua hệ thống, quản trị viên cũng dễ dàng theo dõi và kiểm duyệt các đánh giá sản phẩm từ khách hàng, phản hồi hoặc xóa các đánh giá không phù hợp.

Quản trị viên cũng có thể tạo các mã khuyến mãi, theo dõi các chương trình khuyến mãi để tối ưu hóa chiến lược bán hàng. Các báo cáo doanh thu, số lượng đơn hàng, tỷ lệ hủy và các chỉ số quan trọng khác được tạo ra và xuất dưới dạng Excel để giúp quản lý hiệu quả các hoạt động của ứng dụng.

Quy trình này không chỉ giúp tối ưu hóa trải nghiệm mua sắm của khách hàng mà còn giúp quản trị viên dễ dàng quản lý và kiểm soát các hoạt động bán hàng, từ đó đảm bảo hiệu quả vận hành của ứng dụng.

## 3.2. Chức năng hệ thống

|  |  |
| --- | --- |
| **Vai trò** | **Chức năng** |
| Khách hàng | **- Chức năng tìm kiếm:** Hệ thống sẽ gợi ý các sản phẩm phù hợp với từ khóa mà khách hàng nhập vào. Chức năng tìm kiếm này giúp tối ưu hóa trải nghiệm mua sắm, giúp khách hàng tiết kiệm thời gian và dễ dàng tìm thấy sản phẩm mà họ đang tìm kiếm.  **- Thêm giỏ hàng:** Khách hàng có thể thêm các sản phẩm vào giỏ hàng của mình. Mỗi sản phẩm được thêm vào giỏ hàng sẽ bao gồm các thông tin chi tiết như tên sản phẩm, hình ảnh, số lượng, kích cỡ, màu sắc và giá tiền. Việc này giúp khách hàng dễ dàng quản lý các sản phẩm đã chọn và chuẩn bị cho quá trình thanh toán. Khách hàng có thể xem lại các sản phẩm đã thêm vào giỏ hàng, điều chỉnh số lượng hoặc xóa sản phẩm nếu cần thiết trước khi tiến hành đặt hàng.  **- Đặt hàng:** Sau khi khách hàng hoàn tất việc chọn sản phẩm và kiểm tra giỏ hàng, hệ thống sẽ gửi thông tin chi tiết của đơn hàng đã đặt tới email của khách hàng để họ có thể kiểm tra lại mọi thông tin liên quan.  **- Quản lý tài khoản**: Khách hàng có thể đăng ký hoặc đăng nhập vào tài khoản cá nhân của mình để truy cập vào các tính năng của hệ thống. Sau khi đăng nhập thành công, khách hàng có thể chỉnh sửa và cập nhật thông tin tài khoản cá nhân như tên, email, số điện thoại, và địa chỉ giao hàng  **- Theo dõi đơn hàng**: Khách hàng có thể theo dõi trạng thái hiện tại của đơn hàng mình đã đặt, bao gồm các thông tin về việc đơn hàng đã được đặt, đang trong quá trình vận chuyển hay đã được giao thành công.  **- Lịch sử mua hàng**: Hệ thống hiển thị danh sách các sản phẩm mà khách hàng đã mua trong quá khứ, giúp họ dễ dàng theo dõi và tìm lại các đơn hàng trước đó. Khách hàng có thể xem lại các sản phẩm đã mua, kiểm tra các chi tiết về giá cả, tình trạng giao hàng, và thời gian mua.  - **Xem chi tiết sản phẩm:** Mỗi sản phẩm sẽ có một trang chi tiết hiển thị đầy đủ các thông tin cần thiết để khách hàng đưa ra quyết định mua sắm. Các thông tin bao gồm tên sản phẩm, mô tả chi tiết về tính năng, giá sản phẩm trước và sau khi áp dụng các ưu đãi, cũng như các lựa chọn về kích cỡ và màu sắc của sản phẩm.  **- Gửi yêu cầu:** Khách hàng có thể gửi yêu cầu hỗ trợ đến cho quản trị viên nếu muốn mua những sản phẩm đặt riêng mà hệ thống không có sẵn trong danh mục sản phẩm hiện tại. |
| Quản trị viên | **1. Quản lý sản phẩm**  **- Thêm mới sản phẩm**: Nhập thông tin chi tiết (tên, mô tả, giá, hình ảnh, thuộc tính như kích thước, màu sắc, số lượng).  **- Chỉnh sửa sản phẩm**: Cập nhật mô tả, giá, hình ảnh, hoặc thuộc tính sản phẩm.  **- Xóa sản phẩm**: Xóa các sản phẩm không còn kinh doanh, kèm theo tùy chọn lưu trữ để dễ dàng khôi phục.  - **Hiện danh sách sản phẩm:** Hiển thị danh sách sản phẩm và lọc theo danh mục với giá tiền.  **2. Quản lý danh mục sản phẩm**  **- Thêm mới danh mục**: Tạo danh mục mới hoặc danh mục con để phân loại sản phẩm dễ dàng.  **- Chỉnh sửa danh mục**: Thay đổi tên, trạng thái hoặc mô tả danh mục.  **- Xóa danh mục**: Xóa danh mục khi không còn sử dụng.  **3. Quản lý đơn hàng**  **- Xem danh sách đơn hàng**: Hiển thị danh sách đầy đủ đơn hàng, bao gồm thông tin về khách hàng, sản phẩm, phương thức thanh toán.  **- Cập nhật trạng thái đơn hàng**: Chuyển đổi trạng thái đơn hàng (đang xử lý, đã giao hàng, đã hủy) và thông báo cho khách hàng.  **- In hóa đơn và nhãn vận chuyển**: Tạo và in hóa đơn bán hàng cũng như nhãn vận chuyển trực tiếp từ hệ thống.  **4. Quản lý khách hàng**  **- Xem thông tin khách hàng**: Xem chi tiết hồ sơ khách hàng (tên, email, địa chỉ, phương thức thanh toán).  **- Quản lý lịch sử mua hàng**: Theo dõi lịch sử đơn hàng và tương tác của khách hàng trên website.  **- Gửi email thông báo**: Gửi thông báo qua email cho khách hàng về khuyến mãi, trạng thái đơn hàng, và các chương trình đặc biệt.  **5. Quản lý mã khuyến mãi**  **- Tạo mã khuyến mãi**: Admin có thể tạo mã giảm giá dựa trên phần trăm hoặc giá trị cụ thể.  **- Chỉnh sửa mã khuyến mãi**: Cập nhật thông tin mã khuyến mãi, thời hạn sử dụng, và giới hạn số lần sử dụng.  **- Xóa mã khuyến mãi**: Xóa các mã khuyến mãi hết hạn hoặc không còn sử dụng.  **6. Thống kê và báo cáo**  **- Tạo báo cáo doanh thu**: Tạo báo cáo chi tiết về doanh thu hàng ngày, hàng tháng, hàng năm  **- Thống kê đơn hàng**: Xem số lượng đơn hàng, tỷ lệ hủy đơn, và tỷ lệ thành công.  **- Xuất báo cáo**: Xuất dữ liệu báo cáo dưới dạng Excel để chia sẻ hoặc sử dụng nội bộ.  **8. Quản lý đánh giá và nhận xét**  **- Xem đánh giá chi tiết**: Admin có thể xem các đánh giá và nhận xét của khách hàng về sản phẩm. |

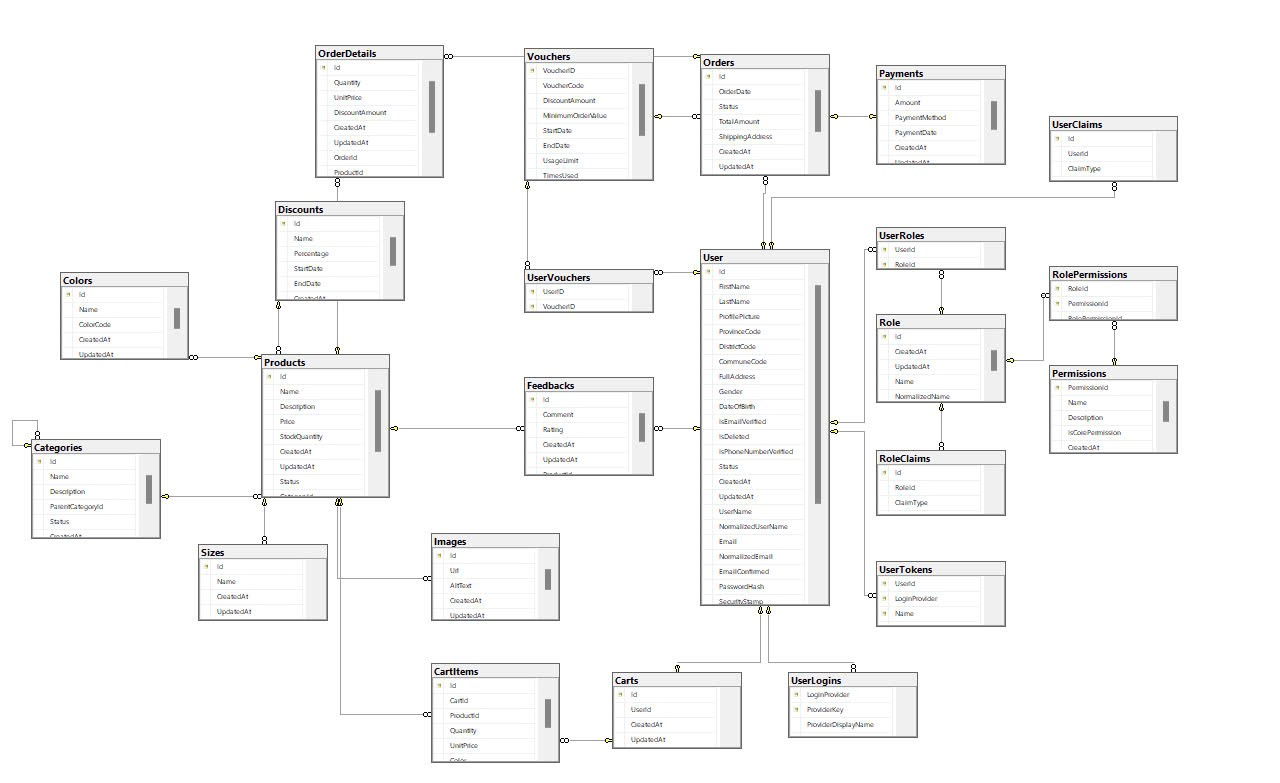
# THIẾT KẾ HỆ THỐNG

## 4.1 Thiết kế cơ sở dữ liệu

**4.1.1. Các thực thể và thuộc tính.**

|  |  |
| --- | --- |
| Entity | Vai trò & Nhiệm vụ |
| 1. User | Vai trò: Đại diện cho người dùng của hệ thống.  Nhiệm vụ: Lưu thông tin về người dùng như tên, email, mật khẩu, và các thông tin cá nhân khác.  Thuộc tính:  + UserID (PK): nvarchar(50)  + UserName: nvarchar(25)  + Email: nvarchar(50)  + PhoneNumber: nvarchar(50)  + PasswordHash: nvarchar(150)  + IsActive: bool  + CreatedDate: Datetime  + LastLoginDate: Datetime |
| 1. Role | Vai trò: Định nghĩa các vai trò khác nhau trong hệ thống (ví dụ: Admin, CEO, Manager, Shipper, Customer Service Staff,....).  Nhiệm vụ: Quản lý quyền hạn của từng vai trò, xác định những gì người dùng có thể thực hiện.  Thuộc tính:  + RoleID (PK): nvarchar(50)  + RoleName: nvarchar(50)  + Description: nvarchar(250), null |
| 1. UserRoles (Liên kế n-n giữa User và Role) | Vai trò: Liên kết nhiều người dùng với nhiều vai trò.  Nhiệm vụ: Quản lý mối quan hệ giữa người dùng và vai trò của họ.  Thuộc tính:  + UserID (FK): nvarchar(50)  + RoleID (FK): nvarchar(50)  + AssignedDate: Datetime |
| 1. UserClaims | Vai trò: Chứa các thông tin bổ sung (claims) của người dùng, thường là các quyền hạn hoặc thông tin liên quan đến bảo mật.  Nhiệm vụ: Xác thực quyền truy cập và lưu trữ các thông tin tùy chỉnh của người dùng.  Thuộc tính:  + UserClaimID (PK): nvarchar(50)  + UserID (FK): nvarchar(50)  + ClaimType: nvarchar(50)  + ClaimValue: nvarchar(250) |
| 1. UserTokens | Vai trò: Quản lý các token bảo mật cho người dùng, thường dùng cho việc khôi phục mật khẩu hoặc xác thực hai bước.  Nhiệm vụ: Lưu trữ token tạm thời để xác minh danh tính hoặc khôi phục tài khoản.  Thuộc tính:  + UserTokenID (PK): nvarchar(50)  + UserID (FK): nvarchar(50)  + Token: nvarchar(150)  + TokenType: nvarchar(50)  + CreatedDate: DateTime  + ExpirationDate: DateTime |
| 1. UserLogins | Vai trò: Quản lý thông tin đăng nhập của người dùng, đặc biệt khi đăng nhập qua dịch vụ thứ ba (OAuth).  Nhiệm vụ: Lưu trữ thông tin về nhà cung cấp dịch vụ đăng nhập của người dùng (Google, Facebook, v.v.).  Thuộc tính:  + UserLoginID (PK): nvarchar(50)  + UserID (FK): nvarchar(50)  + LoginProvider: nvarchar(50)  + ProviderKey: nvarchar(150)  + ProviderDisplayName: nvarchar(150) |
| 1. Permission | Vai trò: Định nghĩa các quyền cụ thể trong hệ thống.  Nhiệm vụ: Quản lý các quyền người dùng có thể có trong hệ thống dựa trên vai trò của họ.  Thuộc tính:  + PermissionID (PK): nvarchar(50)  + RoleID (FK): nvarchar(50)  + PermissionName: nvarchar(150)  + Description: nvarchar(250) |
| 1. Order | Vai trò: Đại diện cho đơn hàng của khách hàng.  Nhiệm vụ: Quản lý chi tiết đơn hàng như ngày đặt hàng, trạng thái đơn hàng, và thông tin khách hàng.  Thuộc tính:  + OrderID (PK): nvarchar(50)  + UserID (FK): nvarchar(50)  + CustomerID (FK): nvarchar(50)  + CartID (FK): nvarchar(50)  + PaymentID (FK): nvarchar(50)  + OrderDate: DateTime  + TotalAmount: decimal(18, 2)  + Status: enum   * Pending - Đơn hàng mới được tạo và chờ xác nhận thanh toán (nếu có) * Processing - Đơn hàng đang được xử lý, chuẩn bị hàng và sắp xếp vận chuyển. * Delivery - Đơn hàng đang trong quá trình giao hàng, dự kiến đến tay khách hàng trong ngày. * Delivered - Đơn hàng đã được giao thành công đến khách hàng. * Completed - Đơn hàng đã hoàn tất sau khi xác nhận việc nhận hàng thành công và không có yêu cầu trả hàng. * Canceled - Đơn hàng bị hủy do yêu cầu từ khách hàng hoặc do vấn đề phát sinh (hết hàng, lỗi thanh toán). * Returned - Đơn hàng đã được trả lại từ phía khách hàng và đang trong quá trình xử lý hoàn tiền hoặc đổi hàng. * Refunded - Đơn hàng đã được hoàn tiền cho khách hàng sau khi xử lý yêu cầu trả hàng hoặc hủy đơn.   + ShippingAddress: nvarchar(250) |
| 1. OrderItem | Vai trò: Lưu trữ thông tin về các sản phẩm trong đơn hàng.  Nhiệm vụ: Quản lý sản phẩm, số lượng và giá tiền của từng mục trong đơn hàng.  Thuộc tính:  + OrderItemID (PK): nvarchar(50)  + OrderID (FK): nvarchar(50)  + ProductID (FK): nvarchar(50)  + Quantity: int  + UnitPrice: decimal(18, 2)  + Discount: double |
| 10. Payment | Vai trò: Đại diện phương thức thanh toán cho một đơn hàng.  Nhiệm vụ: Lưu trữ thông tin thanh toán bao gồm phương thức thanh toán, số tiền, và trạng thái thanh toán.  Thuộc tính:  + PaymentID (PK): nvarchar(50)  + PaymentDestination (FK): nvarchar(50)  + CustomerID (FK): nvarchar(50)  + RefID: nvarchar(50)  + RequiredAmount: decimal(18, 2)  + PaymentContent: nvarchar(350)  + ExpireDate: datetime  + PaymentCurrency: nvarchar(25)  + PaymentLanguage: nvarchar(25)  + PaidAmount: decimal(18, 2)  + PaymentLastMessage: nvarchar(250)  + PaymentStatus: enum   * Pending - Thanh toán đang chờ xử lý hoặc xác nhận từ khách hàng (ví dụ: chờ khách hàng hoàn tất thanh toán online). * Authorized - Thanh toán đã được phê duyệt nhưng chưa thực sự rút tiền từ tài khoản khách hàng (trạng thái này thường xảy ra khi thanh toán qua thẻ). * Paid/Completed - Thanh toán đã hoàn tất, số tiền đã được chuyển thành công từ khách hàng sang hệ thống của bạn. * Failed - Thanh toán thất bại do lỗi kỹ thuật, từ chối giao dịch, hoặc do thông tin thanh toán không hợp lệ. * Canceled - Thanh toán bị hủy bỏ, có thể do khách hàng yêu cầu hoặc vì giao dịch không hợp lệ. * Refunded - Thanh toán đã được hoàn tiền lại cho khách hàng (thường áp dụng trong trường hợp trả hàng hoặc hủy đơn hàng). * Partially Refunded - Một phần thanh toán đã được hoàn lại cho khách hàng (thường dùng khi hoàn tiền chỉ cho một phần của đơn hàng). * Disputed - Thanh toán đang trong trạng thái tranh chấp, thường là do khách hàng khiếu nại về giao dịch hoặc yêu cầu bồi hoàn.   + PaymentDate: datetime |
| 11. PaymentSignture | Vai trò: Chứa thông tin chữ ký xác thực giao dịch thanh toán.  Nhiệm vụ: Đảm bảo tính bảo mật và hợp lệ của giao dịch.  Thuộc tính:  + PaymentSignID (PK): nvarchar(50)  + PaymentID (FK): nvarchar(50)  + SignValue: nvarchar(250)  + SignAlgo: nvarchar(250)  + SignDate: datetime  + SignOwn: nvarchar(50) |
| 12. PaymentTransaction | Vai trò: Quản lý chi tiết các giao dịch thanh toán.  Nhiệm vụ: Lưu trữ các giao dịch liên quan đến thanh toán của đơn hàng.  Thuộc tính:  + PaymentTranID (PK): nvarchar(50)  + PaymentID (FK): nvarchar(50)  + TranDate: datetime  + TranAmount: decimal(18, 2)  + TranStatus: enum   * Pending - Giao dịch đang chờ xử lý, có thể là đang chờ xác nhận từ khách hàng hoặc chờ hệ thống thanh toán phản hồi. * Completed/Success - Giao dịch đã hoàn thành thành công và số tiền đã được chuyển đổi. * Failed - Giao dịch thất bại do lỗi kỹ thuật, từ chối thanh toán, hoặc thông tin giao dịch không chính xác. * Canceled - Giao dịch đã bị hủy bởi khách hàng hoặc hệ thống do không tuân thủ quy trình. * Refunded - Giao dịch đã hoàn tất và số tiền đã được hoàn lại cho khách hàng. * Partially Refunded - Một phần giao dịch đã được hoàn tiền, có thể do khách hàng chỉ yêu cầu hoàn lại một phần. * Authorized - Giao dịch đã được phê duyệt nhưng chưa rút tiền (thường gặp trong thanh toán thẻ tín dụng khi số tiền được giữ nhưng chưa chuyển đổi). * Chargeback - Giao dịch bị hoàn lại tiền do khiếu nại từ khách hàng hoặc ngân hàng, yêu cầu hoàn tiền từ phía đơn vị phát hành thẻ.   + TranMessage: nvarchar(250)  + TranPayload: text |
| 13. PaymentDestination | Vai trò: Đại diện cho nơi thanh toán được chuyển tới (ví dụ: tài khoản ngân hàng, ví điện tử).  Nhiệm vụ: Lưu trữ thông tin đích của thanh toán.  Thuộc tính:  + PaymentDesID (PK): nvarchar(50)  + IsActive: bool  + DesLogo: text  + DesShortName: nvarchar(50)  + DesSortIndex: tinyint  + DesName: nvarchar(50) |
| 14. PaymentNotification | Vai trò: Gửi thông báo về tình trạng thanh toán.  Nhiệm vụ: Quản lý việc gửi và theo dõi thông báo thanh toán thành công hoặc thất bại.  Thuộc tính:  + PaymentNotiID (PK): nvarchar(50)  + CustomerID (FK): nvarchar(50)  + PaymentID (FK): nvarchar(50)  + PayNotiAmount: decimal(18, 2)  + PayNotiContent: nvarchar(250)  + PayNotiResDate: datetime  + PayNotiDate: datetime  + PayNotiStatus: enum   * Pending - Thông báo thanh toán đang chờ xử lý hoặc chưa được xác nhận. * Received - Thông báo thanh toán đã được nhận từ hệ thống thanh toán (thí dụ: từ ngân hàng hoặc cổng thanh toán). * Processed - Thông báo đã được xử lý thành công và cập nhật trạng thái thanh toán trong hệ thống. * Failed - Xử lý thông báo thanh toán thất bại do lỗi kỹ thuật, dữ liệu không hợp lệ, hoặc không thể xác thực. * Canceled - Thông báo thanh toán đã bị hủy bỏ, có thể do yêu cầu từ người dùng hoặc do lỗi trong quá trình gửi thông báo. * Refunded - Thông báo về việc hoàn tiền đã được xử lý và xác nhận. * Disputed - Thông báo liên quan đến một giao dịch bị tranh chấp, thường liên quan đến việc khách hàng khiếu nại.   + PayNotiRefID: nvarchar(50)  + PayNotiSignature: nvarchar(50)  + PayNotiMessage: nvarchar(350) |
| 15. ReturnRequest | Vai trò: Quản lý yêu cầu trả hàng từ khách hàng.  Nhiệm vụ: Lưu trữ các yêu cầu trả hàng và trạng thái xử lý.  Thuộc tính:  + ReturnRequestID (PK): nvarchar(50)  + OrderID (FK): nvarchar(50)  + ProductID (FK): nvarchar(50)  + RequestDate: datetime  + ReturnReason: nvarchar(500)  + ReturnStatus: enum   * Pending - Yêu cầu trả hàng đã được tạo nhưng chưa được xử lý. * Approved - Yêu cầu trả hàng đã được phê duyệt và có thể tiếp tục đến bước tiếp theo (thường là gửi hàng trả lại). * Rejected - Yêu cầu trả hàng không được chấp nhận, có thể do lý do không hợp lệ hoặc không đủ điều kiện. * Returned - Hàng đã được gửi trả lại và đang chờ xử lý. * Refunded - Yêu cầu hoàn tiền đã được xử lý, và số tiền đã được hoàn trả cho khách hàng. * Partially Refunded - Một phần của yêu cầu trả hàng đã được hoàn tiền. * Completed - Tất cả các quy trình liên quan đến yêu cầu trả hàng đã hoàn tất, bao gồm cả hoàn tiền và nhận hàng. * Cancelled - Yêu cầu trả hàng đã bị hủy bỏ, có thể do yêu cầu từ khách hàng. |
| 16. Customer | Vai trò: Đại diện cho khách hàng trong hệ thống.  Nhiệm vụ: Lưu trữ thông tin chi tiết về khách hàng như tên, địa chỉ, email.  Thuộc tính:  + CustomerID (PK): nvarchar(50)  + CartID (FK): nvarchar(50)  + FullName: nvarchar(100)  + Email: nvarchar(100)  + PhoneNumber: nvarchar(50)  + ShippingAddress: nvarchar(350)  + CreatedDate: datetime |
| 17. UserCustomers (Liên kết n-n giữa User và Customer) | Vai trò: Quản lý mối quan hệ giữa người dùng và khách hàng.  Nhiệm vụ: Lưu trữ thông tin liên quan đến việc một người dùng quản lý nhiều khách hàng (Ví dụ nhân viên chăm sóc khách hàng sẽ quản lý nhiều khách hàng, một khách hàng sẽ được chăm sóc bởi nhiều nhân viên)  Thuộc tính:  + UserID (FK): nvarchar(50)  + CustomerID (FK): nvarchar(50) |
| 18. Cart | Vai trò: Đại diện cho giỏ hàng của khách hàng.  Nhiệm vụ: Lưu trữ sản phẩm mà khách hàng đã thêm vào giỏ hàng nhưng chưa thanh toán.  Thuộc tính:  + CartID (PK): nvarchar(50)  + CreatedDate: datetime  + Status: enum   * Active - Giỏ hàng đang hoạt động, có thể có sản phẩm và khách hàng có thể tiếp tục thêm hoặc xóa sản phẩm. * Saved - Giỏ hàng đã được lưu lại (có thể là cho người dùng đã đăng nhập), và sẽ được giữ lại cho lần truy cập sau. * Abandoned - Giỏ hàng không còn hoạt động trong một khoảng thời gian dài mà không có hành động từ người dùng (thường là sau khi khách hàng rời trang mà không thanh toán). * CheckOut - Giỏ hàng đã được thanh toán, và quy trình đặt hàng đã được thực hiện. * Empty - Giỏ hàng không có sản phẩm nào. Trạng thái này có thể được dùng để xác định khi nào người dùng bắt đầu thêm sản phẩm vào giỏ. * Cleared - Giỏ hàng đã bị xóa hoặc tất cả sản phẩm đã bị loại bỏ. |
| 19. CartItem | Vai trò: Quản lý các sản phẩm trong giỏ hàng.  Nhiệm vụ: Lưu trữ chi tiết sản phẩm, số lượng và giá cho từng mục trong giỏ hàng.  Thuộc tính:  + CartItemID (PK): nvarchar(50)  + CartID (FK): nvarchar(50)  + ProductID (FK): nvarchar(50)  + Quantity: int  + UnitPrice: decimal(18, 2) |
| 20. Product | Vai trò: Đại diện cho sản phẩm trong cửa hàng.  Nhiệm vụ: Quản lý thông tin sản phẩm như tên, mô tả, giá, hình ảnh.  Thuộc tính:  + ProductID (PK): nvarchar(50)  + CategoryID (FK): nvarchar(50)  + UserID (FK): nvarchar(50)  + ProductName: nvarchar(150)  + Description: text  + Price: decimal(18, 2)  + StockQuantity: int  + CreatedDate: datetime |
| 21. ReturnProduct | Thuộc tính:  + ProductID (FK): nvarchar(50)  + ReturnRequestID (FK): nvarchar(50) |
| 21. DiscountCode | Vai trò: Chứa mã giảm giá mà khách hàng có thể sử dụng.  Nhiệm vụ: Quản lý thông tin mã giảm giá, phần trăm giảm giá, và ngày hết hạn.  Thuộc tính:  + DiscountCodeID (PK): nvarchar(50)  + Code: nvarchar(250)  + DiscountPercentage: decimal(18, 2)  + StartDate: datetime  + EndDate: datetime  + IsActive: bool |
| 22. DiscountProducts (Liên kết n-n giữa DiscountCode và Product) | Vai trò: Quản lý mối quan hệ giữa mã giảm giá và sản phẩm áp dụng.  Nhiệm vụ: Lưu trữ thông tin về những sản phẩm nào có thể áp dụng mã giảm giá.  Thuộc tính:  + DiscountCodeID (FK): nvarchar(50)  + ProductID (FK): nvarchar(50) |
| 23. Category | Vai trò: Đại diện cho danh mục sản phẩm.  Nhiệm vụ: Quản lý thông tin danh mục và các sản phẩm thuộc danh mục đó.  Thuộc tính:  + CategoryID (PK): nvarchar(50)  + CategoryName: nvarchar(250)  + Description: nvarchar(500)  + ParentCategoryID (nullable, FK) |
| 24. ReviewComment | Vai trò: Quản lý các bình luận, đánh giá sản phẩm.  Nhiệm vụ: Lưu trữ nội dung đánh giá, xếp hạng và thông tin người dùng đã đánh giá.  Thuộc tính:  + ReviewCommentID (PK): nvarchar(50)  + ProductID (FK): nvarchar(50)  + CustomerID (FK): nvarchar(50)  + Comment: text  + Rating: double  + CreatedDate: datetime |
| 25. Warehouse | Vai trò: Đại diện cho kho chứa hàng (kho quần, kho áo, kho mũ,....) tương ứng với mỗi kho là một nhà sản xuất sản phẩm cho kho đó  Nhiệm vụ: Lưu trữ thông tin về các kho và sản phẩm có sẵn trong kho.  Thuộc tính:  + WarehouseID (PK): nvarchar(50)  + ManufacturerID (FK): nvarchar(50)  + WarehouseName: nvarchar(250)  + Location: nvarchar(50)  + Capacity: int |
| 26. WarehouseProducts (Liên kết n-n giữa Warehouse và Product) | Vai trò: Quản lý mối quan hệ giữa kho và sản phẩm.  Nhiệm vụ: Lưu trữ thông tin về số lượng sản phẩm có trong từng kho.  Thuộc tính:  + WarehouseID (FK): nvarchar(50)  + ProductID (FK): nvarchar(50)  + StockQuantity: int |
| 27. Manufacturer | Vai trò: Đại diện cho nhà sản xuất sản phẩm.  Nhiệm vụ: Lưu trữ thông tin về nhà sản xuất như tên, địa chỉ, liên hệ.  Thuộc tính:  + ManufacturerID (PK): nvarchar(50)  + ManufacturerName: nvarchar(250)  + ContactInfo: nvarchar(10)  + Address: nvarchar(250) |
| 28. InventoryTransaction | Vai trò: Quản lý giao dịch nhập/xuất kho.  Nhiệm vụ: Lưu trữ thông tin về các giao dịch liên quan đến thay đổi số lượng hàng tồn kho.  Thuộc tính:  + InventoryTransactionID (PK): nvarchar(50)  + WarehouseID (FK): nvarchar(50)  + TransactionDate: datetime  + QuantityChanged: int  + TransactionType (nhập/xuất kho) |
| 29. ChatSystem | Vai trò: Hệ thống trò chuyện giữa khách hàng và người quản trị/nhân viên.  Nhiệm vụ: Lưu trữ thông tin về hệ thống chat (VD: Zalo, Message, Chat Realtime,...)  Thuộc tính:  + ChatSystemID (PK): nvarchar(50)  + ChatMessageID (FK): nvarchar(50) |
| 30. ChatMessage | Vai trò: Đại diện cho tin nhắn trong hệ thống trò chuyện.  Nhiệm vụ: Lưu trữ nội dung tin nhắn, thời gian gửi và người gửi/nhận.  Thuộc tính:  + ChatMessageID (PK): nvarchar(50)  + UserID (FK): nvarchar(50)  + CustomerID (FK): nvarchar(50)  + MessageContent: text  + SentDate: datetime |

**4.1.2. Xây dựng mô hình thực thể liên kết**



Hình 4.1 Sơ đồ quan hệ

**4.1.3. Chuyển từ mô hình thực thể liên kết sang mô hình quan hệ**

+ Users (UserID, FirstName, LastName, ProfilePicture, ProviderCode, ReferralCode, FullAddress, DateOfBirth, Gender, PhoneNumber, IsPhoneNumberVerified, Status, CreatedAt, UpdatedAt, Email, PasswordHash, NormalizedUserName, NormalizedEmail, SecurityStamp)

+ Roles (RoleID, Name, NormalizedName, CreatedAt, UpdatedAt)

+ UserRoles (UserID, RoleID)

+ Permissions (PermissionID, Name, Description, IsCorePermission, CreatedAt)

+ RolePermissions (RoleID, PermissionID)

+ UserClaims (ClaimID, UserID, ClaimType)

+ RoleClaims (ClaimID, RoleID)

+ Orders (OrderID, *UserID*, OrderDate, Status, TotalAmount, ShippingAddress, PaymentMethod)

+ OrderDetails (DetailID, *OrderID*, *ProductID*, Quantity, UnitPrice, DiscountAmount)

+ Products (ProductID, Name, Description, Price, StockQuantity, Status, CreatedAt, UpdatedAt, *CategoryID*, *ColorID*, *SizeID*)

+ Categories (CategoryID, Name, Description, ParentCategoryID)

+ Colors (ColorID, ColorCode, CreatedAt, UpdatedAt)

+ Sizes (SizeID, Name, CreatedAt, UpdatedAt)

+ Vouchers (VoucherID, Code, DiscountAmount, MinimumOrderValue, StartDate, EndDate, UsageLimit)

+ UserVouchers (UserID, VoucherID)

+ Feedbacks (FeedbackID, *ProductID*, *UserID*, Rating, Comment, CreatedAt)

+ CartItems (CartItemID, *CartID*, *ProductID*, Quantity, UnitPrice)

+ Carts (CartID, *UserID*, CreatedAt, UpdatedAt)

+ Payments (PaymentID, *OrderID*, Amount, PaymentMethod, CreatedAt)

**4.1.4. Chuẩn hoá**

Tất cả các bảng đều:

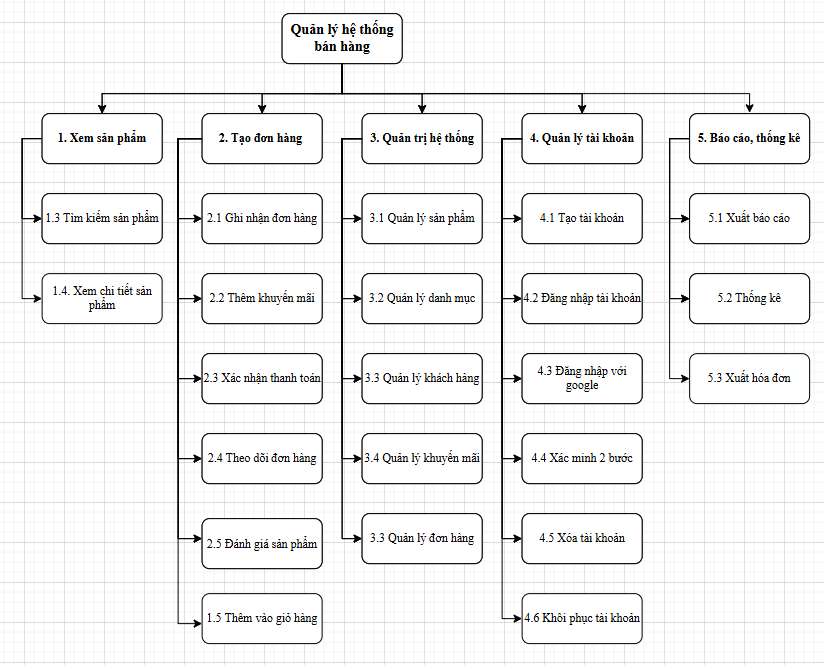
 Đạt chuẩn 1NF.

 Đạt chuẩn 2NF.

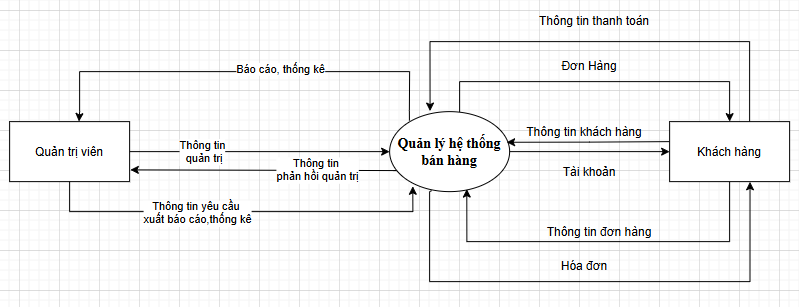
 Đạt chuẩn 3NF

**4.1.5. Mô hình phân tích hệ thống**

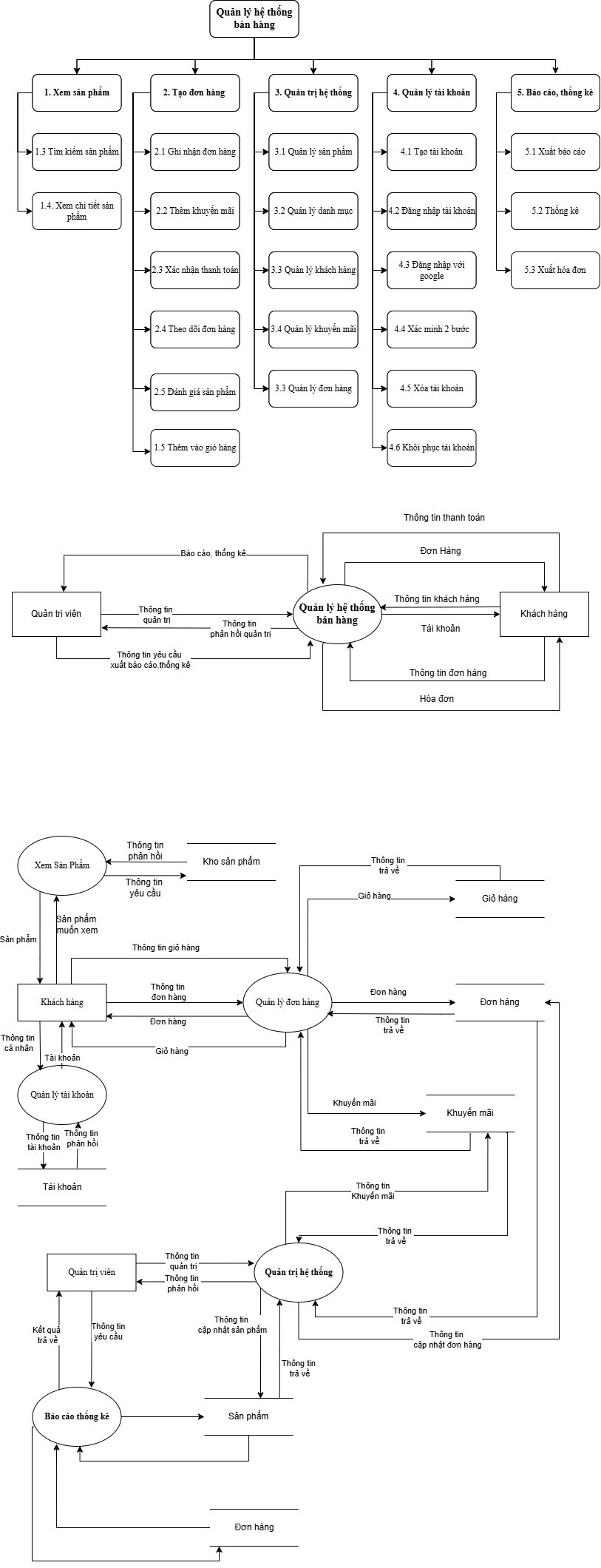
Dưới đây là các mô hình phân tích của hệ thống được biểu diễn bằng mô hình BFD, DFD mức ngữ cảnh và DFD mức 0.



Hình 4.2 Mô hình BFD



Hình 4.3 Mô hình DFD mức ngữ cảnh



Hình 4.4 Mô hình DFD mức 0

## 4.2 Thiết kế API

**4.2.1. Mô tả chi tiết endpoint**

Dưới đây là cách sử dụng API **POST /api/Categories** để tạo một danh mục mới. API này không yêu cầu tham số truyền qua URL (No parameters) và dữ liệu gửi lên server phải ở định dạng application/json. Dữ liệu bao gồm tên danh mục, trạng thái danh mục ("active" hoặc "inactive"), và ID danh mục cha (nếu danh mục này là danh mục con). Endpoint sử dụng là /api/Categories, và nút "Try it out" cho phép kiểm tra API trực tiếp từ giao diện. Khi yêu cầu được gửi thành công, server sẽ trả về mã trạng thái 200 OK. Hình cũng giải thích chi tiết các thành phần chính của request và response trong API.

A screenshot of a computer

Description automatically generated

Hình 4.5 Mô tả chi tiết endpoint

**4.2.2. Toàn bộ APIs trong dự án**

Dự án tích hợp APIs với hơn 100 phương thức, bao gồm các đối tượng như Account, Cart, Categories, Colors, Discounts, Feedback, Images, Order, Payment, Permission, Product, Role, Sizes, User, và Voucher. Để đảm bảo hiệu quả và tính bảo mật, dự án áp dụng các giải pháp sau:

- **Xác thực và phân quyền:** Tích hợp OAuth 2.0 hoặc JWT (JSON Web Token) nhằm đảm bảo các phương thức API chỉ được truy cập bởi các người dùng hoặc hệ thống được cấp phép.

- **Tài liệu API tự động:** Sử dụng công cụ Swagger/OpenAPI để tạo tài liệu API chi tiết, giúp tăng cường khả năng kiểm tra, sử dụng, và tích hợp giữa các đội phát triển.

- **Tối ưu hóa dữ liệu trả về:** Hỗ trợ phân trang (pagination) và sắp xếp (sorting) trên các API trả về danh sách lớn, đảm bảo hiệu suất cao và trải nghiệm người dùng mượt mà.

A screenshot of a computer

Description automatically generated

Hình 4.6 Các phương thức của Account

A group of rectangular colored lines

Description automatically generated with medium confidence

Hình 4.7 Các phương thức của Cart

A screenshot of a computer

Description automatically generated

Hình 4.8 Các phương thức của Categories

A screenshot of a computer

Description automatically generated

Hình 4.9 Các phương thức của Colors

A screenshot of a computer

Description automatically generated

Hình 4.10 Các phương thức của Discounts

A screenshot of a chat

Description automatically generated

Hình 4.11 Các phương thức của Feedback

A screenshot of a chat

Description automatically generated

Hình 4.12 Các phương thức của Images

A screenshot of a computer

Description automatically generated

Hình 4.13 Các phương thức của Order

A screenshot of a computer

Description automatically generated

Hình 4.14 Các phương thức của Payment

A screenshot of a computer

Description automatically generated

Hình 4.15 Các phương thức của Permission

A screenshot of a computer

Description automatically generated

Hình 4.16 Các phương thức của Product

A screenshot of a computer

Description automatically generated

Hình 4.17 Các phương thức của Role

A group of rectangular colored rectangles

Description automatically generated

Hình 4.18 Các phương thức của Sizes

A screenshot of a computer

Description automatically generated

Hình 4.19 Các phương thức của User

A screenshot of a computer

Description automatically generated

Hình 4.20 Các phương thức của Voucher

## 4.3. Thiết kế giao diện chính

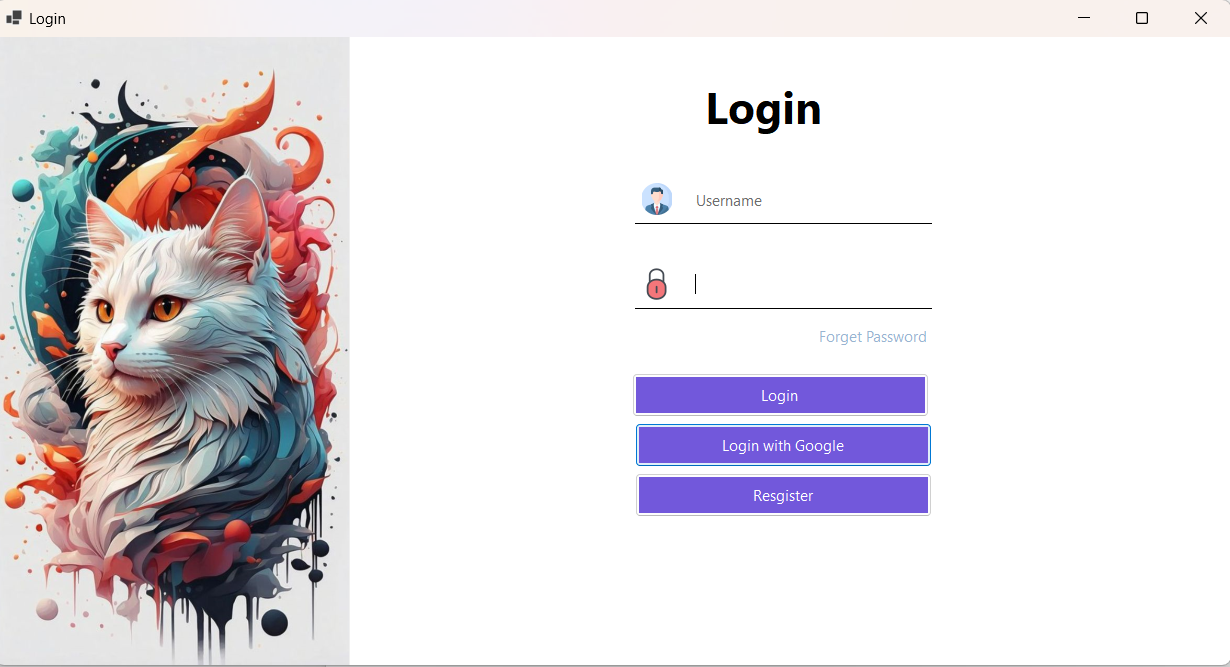
**4.3.1. Giao diện đăng nhập**

Trang này cung cấp hai phương thức đăng nhập tiện lợi cho người dùng:

1. Tên đăng nhập và mật khẩu: Người dùng có thể sử dụng thông tin đã tạo trong quá trình đăng ký để truy cập tài khoản của mình.

2. Đăng nhập bằng Google: Tích hợp với Google giúp người dùng nhanh chóng đăng nhập mà không cần tạo tài khoản mới, đồng thời đảm bảo tính bảo mật và thuận tiện.

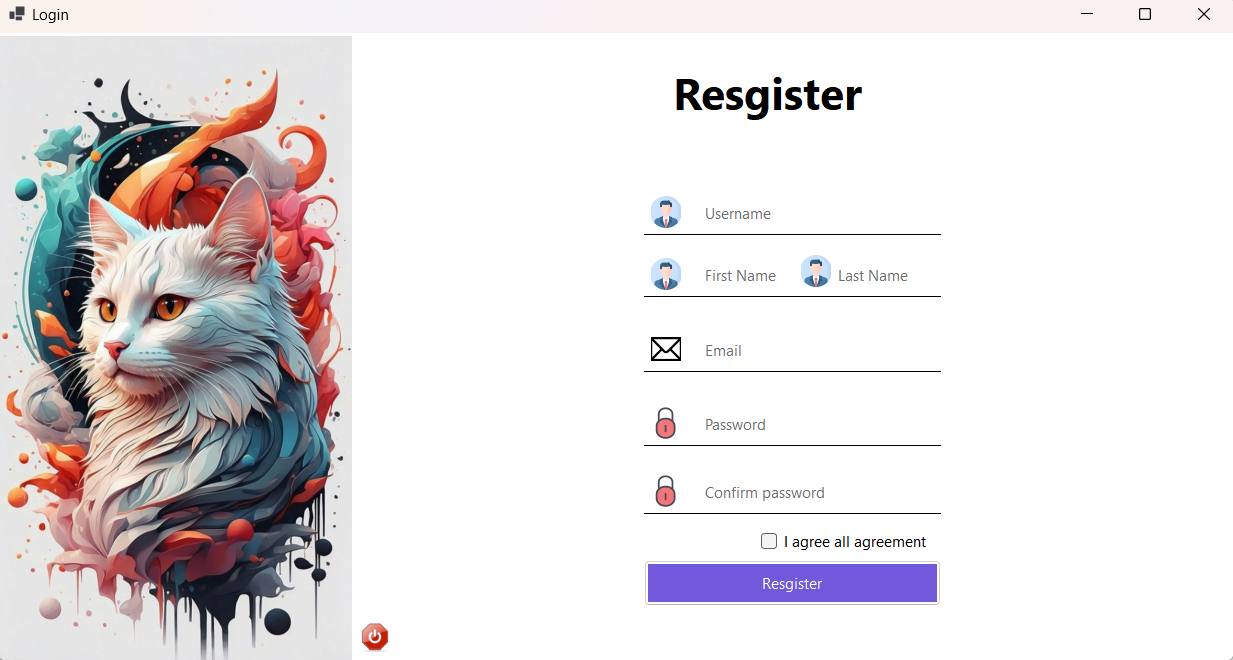
Cả hai phương thức đều được thiết kế để mang lại trải nghiệm đăng nhập đơn giản và an toàn.



Hình 4.21 Giao diện đăng nhập

**4.3.2. Giao diện đăng ký**

Người dùng có thể dễ dàng đăng ký tài khoản trực tiếp trên trang dành riêng cho Khách hàng. Tuy nhiên, tính năng này không áp dụng cho Quản trị viên, vì tài khoản quản trị sẽ được tạo và cung cấp riêng biệt nhằm đảm bảo an toàn và quyền kiểm soát hệ thống.



Hình 4.22 Giao diện đăng ký

**4.3.3. Giao diện quên mật khẩu**

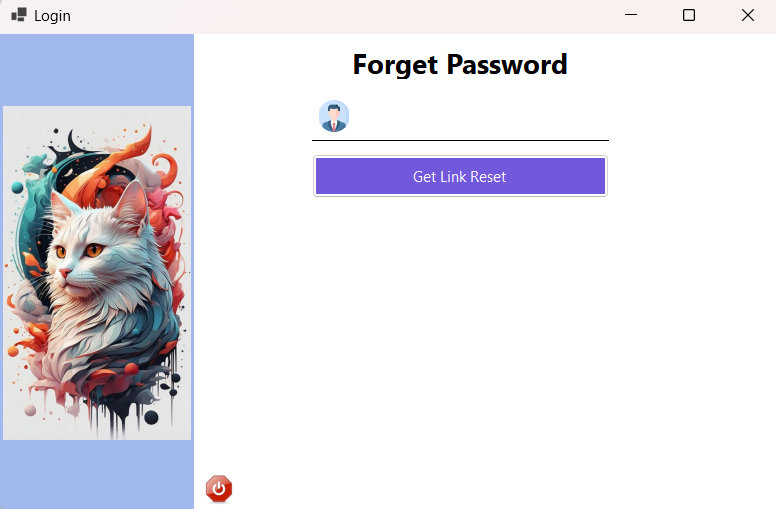
Giao diện "Quên mật khẩu" được thiết kế để hỗ trợ người dùng khôi phục mật khẩu một cách nhanh chóng và an toàn. Tính năng này hoạt động theo các bước sau:

1. Nhập ‘tên đăng nhập’: Người dùng cung cấp ‘tên đăng nhập’ đã dùng để đăng ký tài khoản.

2. Gửi mã xác nhận: Hệ thống sẽ gửi mã xác nhận hoặc liên kết khôi phục mật khẩu đến email đã được sử dụng để đăng ký tài khoản đó.

3. Nhập mã xác nhận: Người dùng điền mã xác nhận từ email vào giao diện hoặc nhấn vào liên kết được gửi.

4. Tạo mật khẩu mới: Sau khi xác thực, người dùng có thể tạo mật khẩu mới để thay thế mật khẩu cũ.



Hình 4.23 Giao diện quên mật khẩu



Hình 4.24 Giao diện đổi mật khẩu

**4.3.4. Giao diện phía ‘Quản trị viên’**

**Giao diện chính của quản trị viên** được thiết kế trực quan với các thành phần chính như sau:

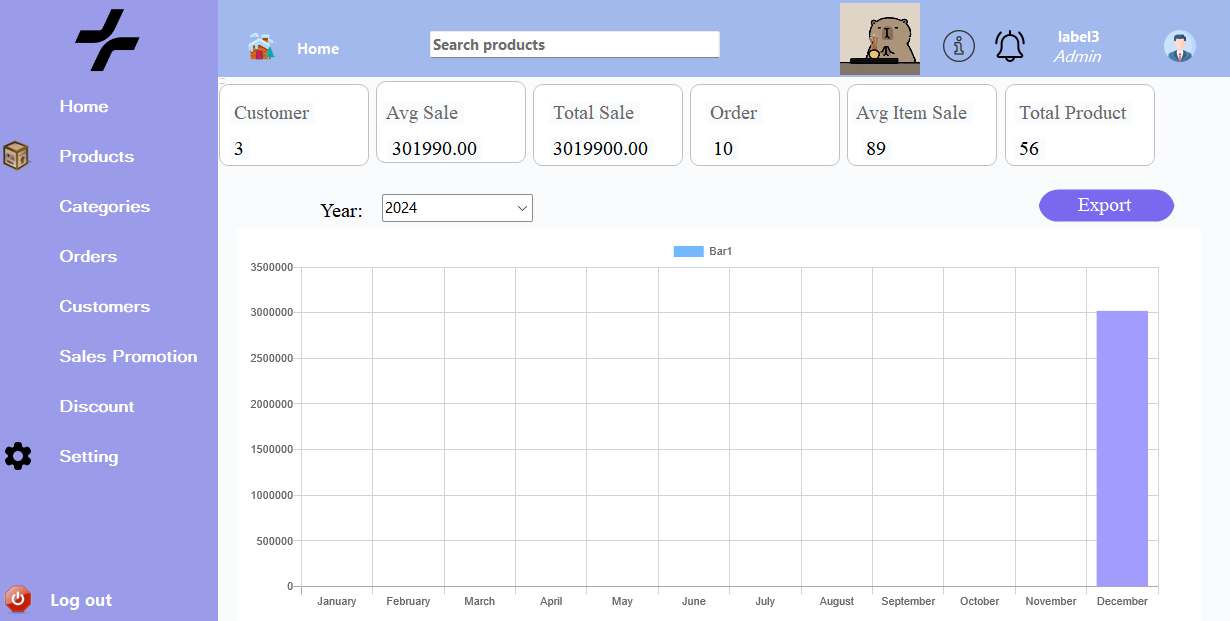
- Thanh điều hướng: Cung cấp các mục quản lý như Home, Products, Categories, Orders, Customers, Sales Promotion, Discount, Setting và nút Log out.

- Thanh công cụ trên cùng: Hỗ trợ tìm kiếm, hiển thị thông tin tài khoản, thông báo và cài đặt.

- Thông tin thống kê: Hiển thị số liệu quan trọng như tổng khách hàng, doanh thu trung bình, tổng sản phẩm, tổng đơn hàng, v.v.

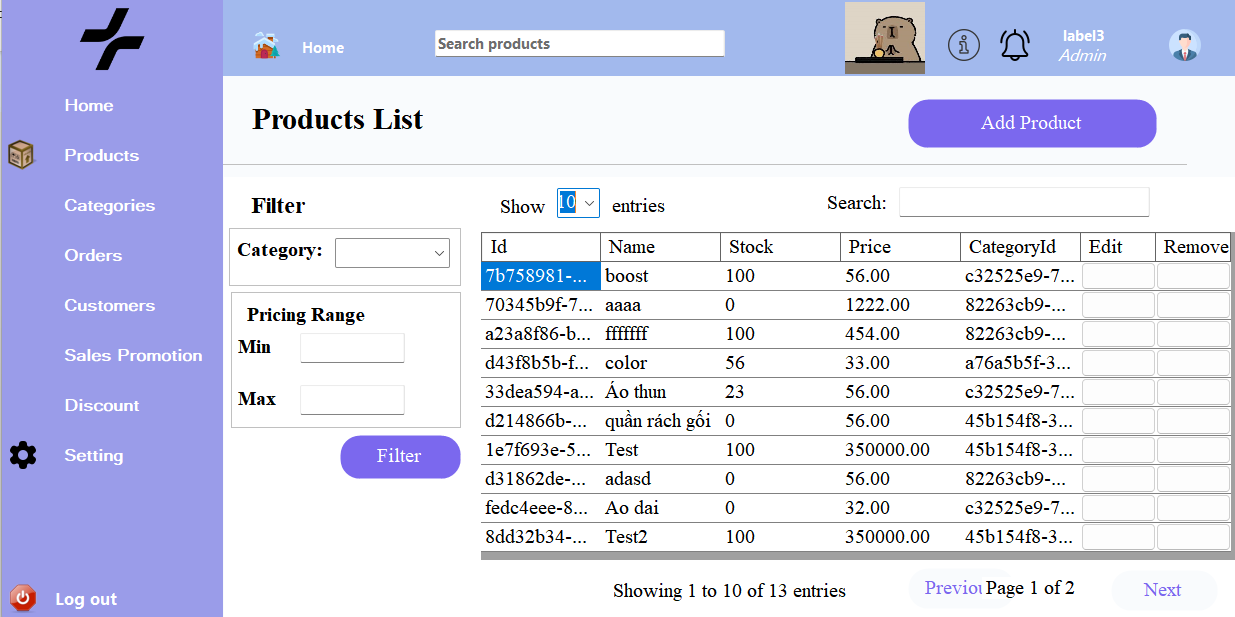
- Biểu đồ trực quan: Biểu đồ cột theo tháng, hỗ trợ chọn năm để phân tích số liệu.

- Nút Export: Cho phép xuất dữ liệu báo cáo.

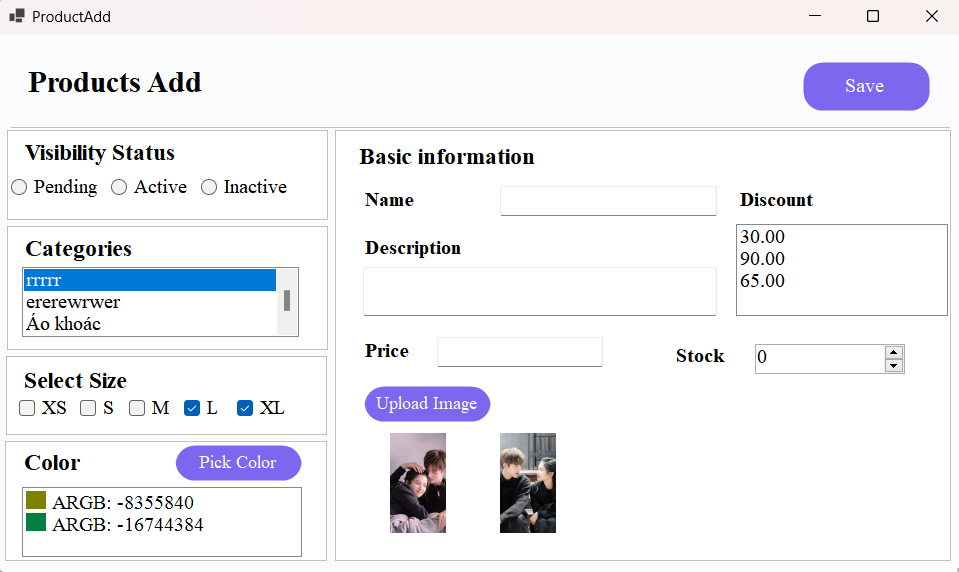


Hình 4.25 Giao diện trang chủ admin

Chỉ những tài khoản được cấp quyền admin mới có thể truy cập vào trang này.

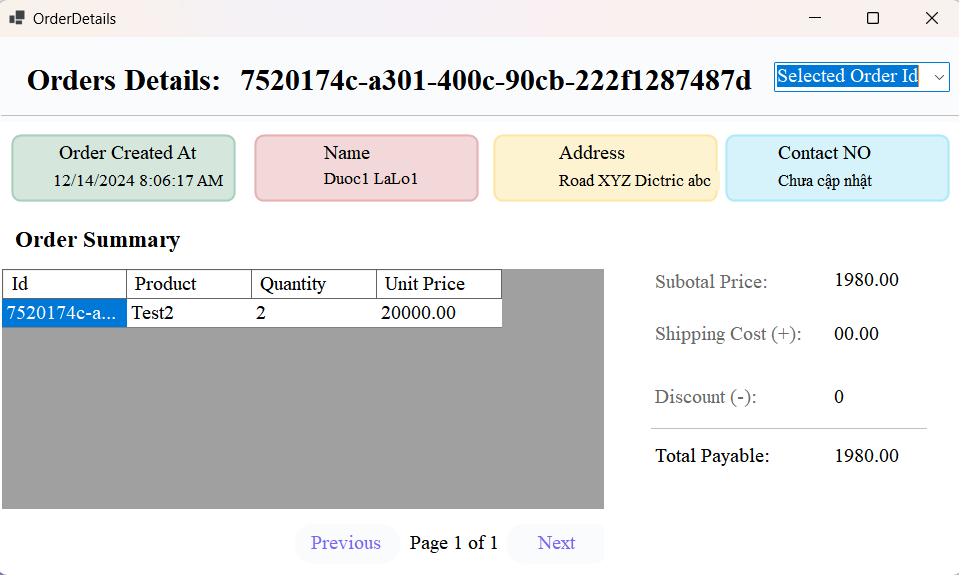
Đối với tài khoản có quyền 'admin', quản trị viên có thể dễ dàng theo dõi và quản lý danh sách sản phẩm hiện có, bao gồm các thông tin như số lượng tồn kho, giá cả, và các chi tiết khác. Quản trị viên cũng có thể lọc sản phẩm theo các danh mục (Categories) hoặc theo khoảng giá từ (Min) đến (Max). Bên cạnh đó, quản trị viên có thể thêm sản phẩm mới vào hệ thống bằng cách nhấp chuột vào nút 'Add Product', giao diện 'Thêm sản phẩm' sẽ hiển thị, cho phép quản trị viên nhập thông tin chi tiết của sản phẩm mới.

Hình 4.26 Giao diện theo dõi danh sách sản phẩm

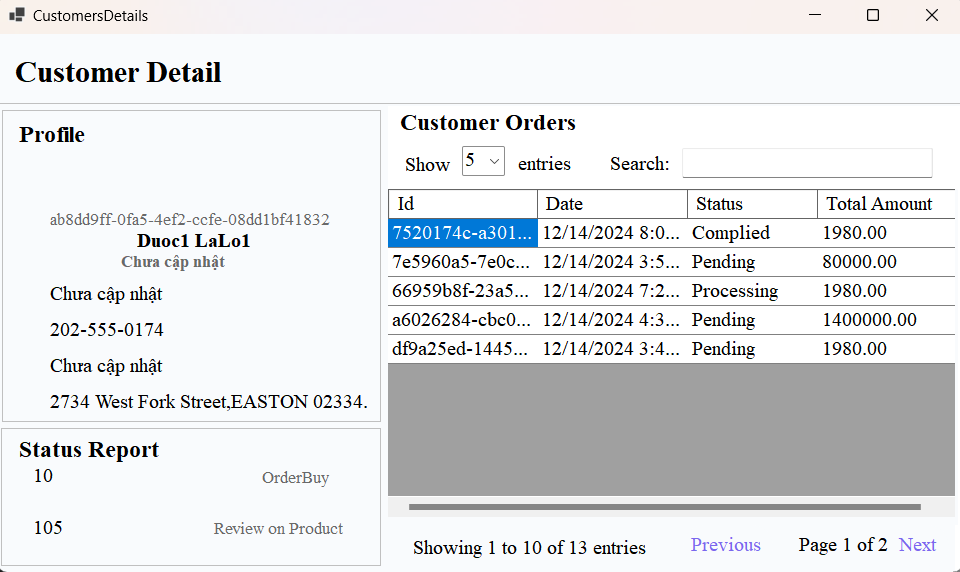


Hình 4.27 Giao diện thêm sản phẩm

Bên cạnh đó, quản trị viên còn có quyền xem chi tiết các đơn đặt hàng cùng thông tin của khách hàng, từ đó có thể theo dõi tình trạng đơn hàng và hỗ trợ khách hàng một cách hiệu quả.



Hình 4.28 Giao diện chi tiết đơn hàng



Hình 4.29 Giao diện chi tiết khách hàng

**4.3.5. Giao diện phía ‘Khách hàng’**

Để nâng cao tính bảo mật, những khách hàng đã kích hoạt tính năng xác thực hai bước sẽ được yêu cầu nhập mã code gửi đến email đã đăng ký mỗi khi thực hiện đăng nhập hoặc thay đổi mật khẩu.

**A screenshot of a computer error

Description automatically generated**

Hình 4.30 Giao diện xác thực 2 bước

Khách hàng cũng có thể dễ dàng theo dõi lịch sử các đơn hàng đã đặt của mình tại trang này.

**A screenshot of a computer

Description automatically generated**

Hình 4.31 Giao diện hóa đơn khách hàng

Ngoài ra, khách hàng còn có thể chỉnh sửa và cập nhật thông tin cá nhân của mình.

**A screenshot of a chat

Description automatically generated**

Hình 4.28 Thông tin khách hàng

Trong trường hợp khách hàng muốn đặt đơn hàng riêng, họ có thể gửi yêu cầu bằng cách điền đầy đủ thông tin vào phiếu yêu cầu.

**A screenshot of a computer

Description automatically generated**

Hình 4.32 Gửi yêu cầu in riêng

Giao diện ‘Setting’ cung cấp các tùy chọn quản lý tài khoản và cá nhân hóa trải nghiệm khách hàng. Các chức năng chính bao gồm:

1. Tài khoản:

- Kích hoạt xác minh hai bước để tăng cường bảo mật.

- Thay đổi mật khẩu và tùy chỉnh các thiết lập bảo mật.

2. Thông báo:

- Quản lý nhận thông báo qua email.

- Bật hoặc tắt thông báo trên các thiết bị được kết nối.

3. Giao diện:

- Lựa chọn ngôn ngữ hiển thị phù hợp với người dùng.

- Chuyển đổi giữa chế độ sáng và tối để tối ưu hóa trải nghiệm.

4. Nâng cao:

- Tùy chỉnh hiển thị trạng thái hoạt động với người khác.

- Xóa vĩnh viễn tài khoản cùng dữ liệu liên quan.

**A screenshot of a chat

Description automatically generated**

Hình 4.33 Cài đặt tài khoản

**A screenshot of a computer

Description automatically generated**

Hình 4.34 Trang giới thiệu

**A screenshot of a store

Description automatically generated**

Hình 4.35 Kho voucher của shop

Đây là giao diện chính hiển thị sau khi khách hàng đăng nhập thành công, với thanh điều hướng nằm ở phía bên trái và khu vực kho voucher được bố trí bên phải, mang lại sự tiện lợi trong việc truy cập các chức năng và ưu đãi.

Giao diện chi tiết sản phẩm được thiết kế để cung cấp đầy đủ thông tin và chức năng hỗ trợ quá trình mua sắm của người dùng. Các chức năng chính bao gồm:

- Hiển thị thông tin sản phẩm: Tên sản phẩm, giá bán, mức giảm giá, số lượng đã bán, và đánh giá từ khách hàng. Điều này giúp người dùng nắm rõ các đặc điểm quan trọng trước khi quyết định mua.

- Tùy chọn cá nhân hóa sản phẩm:

+ Lựa chọn màu sắc: Người dùng có thể chọn giữa các màu của sản phẩm

+ Lựa chọn kích thước: Các size sẽ giúp phù hợp với nhu cầu cá nhân.

+ Điều chỉnh số lượng: Tính năng tăng/giảm số lượng đáp ứng nhu cầu mua nhiều hay ít sản phẩm.

- Hành động mua sắm:

+ Thêm vào giỏ hàng: Cho phép người dùng lưu sản phẩm lại để mua sau hoặc tiếp tục lựa chọn thêm.

+ Mua ngay: Hỗ trợ quá trình thanh toán nhanh chóng, tiết kiệm thời gian cho người dùng.

- Hình ảnh minh họa sản phẩm: Hình ảnh sản phẩm được hiển thị chi tiết từ nhiều góc nhìn, giúp người dùng đánh giá sản phẩm trực quan.

**A screenshot of a website

Description automatically generated**

Hình 4.36 Chi tiết sản phẩm

Sau khi khách hàng nhấn nút "Thêm vào giỏ hàng", thông tin chi tiết về các sản phẩm đã chọn sẽ được hiển thị trong trang "Cart". Tại giao diện này, khách hàng có thể dễ dàng quản lý danh sách sản phẩm bằng cách thực hiện các thao tác sau:

1. Kiểm tra lại thông tin sản phẩm: Tên sản phẩm, phân loại và số lượng.

2. Lựa chọn sản phẩm để mua:

- Khách hàng có thể đánh dấu vào ô vuông ở đầu mỗi dòng sản phẩm để chọn một hoặc nhiều sản phẩm cùng lúc.

- Ngoài ra, nút "Chọn tất cả" được tích hợp ở cuối danh sách, giúp khách hàng nhanh chóng chọn toàn bộ sản phẩm trong giỏ hàng.

3. Thao tác xoá sản phảm: Khách hàng có thể nhấn vào nút "Xóa" bên cạnh từng sản phẩm để loại bỏ những mặt hàng không còn nhu cầu.

4. Tổng hợp thông tin thanh toán:

- Mục "Tổng thanh toán" tự động tính toán tổng số tiền dựa trên các sản phẩm đã chọn, giúp khách hàng theo dõi chi phí một cách minh bạch.

- Nếu không có sản phẩm nào được chọn, tổng thanh toán sẽ hiển thị giá trị là 0.

Sau khi hoàn tất việc lựa chọn, khách hàng có thể nhấn nút "Mua hàng" để chuyển sang bước thanh toán.

**A screenshot of a computer

Description automatically generated**

Hình 4.37 Giỏ hàng

Tại trang này, toàn bộ các sản phẩm sẽ được hiển thị với thiết kế tối ưu cho trải nghiệm người dùng. Mỗi lần hiển thị tối đa 6 sản phẩm, giúp giao diện gọn gàng và dễ quan sát. Khách hàng có thể sử dụng các nút điều hướng qua trái và qua phải để duyệt qua danh sách sản phẩm một cách linh hoạt.

Khi khách hàng nhấn vào một sản phẩm bất kỳ, thông tin chi tiết của sản phẩm sẽ được phóng to ở khu vực bên trái màn hình, giúp người dùng có cái nhìn rõ ràng hơn về sản phẩm, bao gồm hình ảnh, màu sắc, và các chi tiết quan trọng.

Ngoài ra, tính năng nổi bật là nút ‘View Detail’, cho phép khách hàng chuyển hướng trực tiếp đến trang chi tiết của sản phẩm để xem thông tin đầy đủ hơn. Bên cạnh đó, một thanh công cụ tìm kiếm cũng được tích hợp, cho phép khách hàng nhanh chóng tìm kiếm sản phẩm theo ý muốn, nâng cao trải nghiệm và tiết kiệm thời gian.**A screenshot of a website

Description automatically generated**

Hình 4.38 Xem sản phẩm theo danh mục

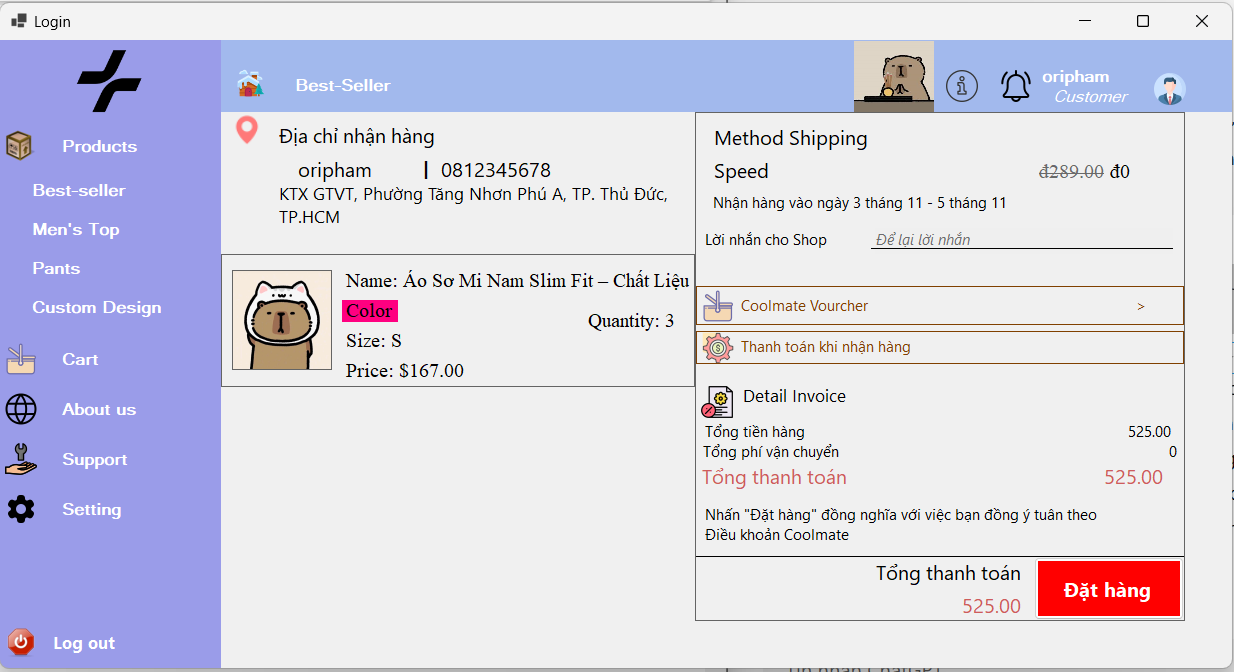
Tại trang mua hàng, hệ thống sẽ hiển thị thông tin chi tiết về sản phẩm đã chọn, bao gồm tên sản phẩm, số lượng, kích cỡ, màu sắc, giá tiền và các thông tin liên quan khác. Đồng thời, khách hàng cũng sẽ kiểm tra thông tin người nhận, bao gồm tên, địa chỉ giao hàng và số điện thoại. Nếu khách hàng chưa nhập đầy đủ thông tin hoặc có bất kỳ thay đổi nào cần thiết, hệ thống sẽ tự động điều hướng khách hàng đến trang ‘Chỉnh sửa thông tin’ để cập nhật hoặc bổ sung.

Ngoài ra, khách hàng có thể nhấn chọn mục ‘Voucher’ để áp dụng các mã giảm giá hoặc khuyến mãi có sẵn. Việc này giúp khách hàng tiết kiệm chi phí khi mua hàng. Nếu có bất kỳ ghi chú gì, khách hàng có thể nhập vào mục ‘Để lại lời nhắn’. Sau khi đã hoàn tất kiểm tra và điều chỉnh các thông tin, khách hàng có thể tiếp tục tiến hành thanh toán để hoàn tất đơn hàng.

Thông tin đơn hàng sẽ được gửi về email của khách hàng để kiểm tra và xác nhận lần nữa.



Hình 4.39 Giao diện chọn voucher



Hình 4.40 Giao diện trang đặt hàng

# KẾT LUẬN VÀ KIẾN NGHỊ

**Kết quả đạt được**

Trong suốt khoảng thời gian nghiên cứu và thực hiện đề tài, nhóm đã cố gắng làm hết khả năng của mình và vì thời gian làm có hạn nên kết quả đạt được vẫn còn nhiều hạn chế nhưng nhóm đã học hỏi được khá nhiều kiến thức mới mẻ cũng như chuyên sâu về thuật toán, lập trình... Và những việc chúng em đã đạt được như sau:

+ Giao diện cho phép người dùng thao tác mua sắm, theo dõi sản phẩm, đơn hàng một cách tiện ích, thuận tiện.

+ Giao diện admin cho phép quản lý cửa hàng online hiệu quả: quản lý sản phẩm, danh mục, đơn hàng, khách hàng,...

**Tồn tại**

Trong quá trình triển khai phần mềm, hiển nhiên không tránh khỏi những sai xót ảnh hưởng tới hiệu suất sử dụng của ứng dụng. Như thiết kế UI/UX chưa tối ưu. Và 1 vài lỗi nhỏ về logic khi chạy ứng dụng.

**Hướng phát triển**

Với những chức năng đã làm như trên, nhóm định hướng phát triển thêm tính năng:

+ Thanh toán qua các nền tảng khác.

+ Admin có khả năng phân quyền cho nhân viên

+ Đặt áo theo yêu cầu

Đi cùng với việc tối ưu hóa UX//UI.

# TÀI LIỆU THAM KHẢO

1. **Nguyễn Tất Bảo Thiện, Phạm Quang Hiển**, Lập Trình Windows Form Và Web Form Với C#, 2020
2. **Andrew Stellman & Jennifer Greene**, Head First C#: A Learner's Guide to Real-World Programming with C# and .NET Core, 4th Edition, 2019
3. **[Phạm Quang Hiển](https://newshop.vn/tac-gia/pham-quang-hien-17851.html)**[, Tự học lập trình bằng hình ảnh, 2023](https://newshop.vn/tac-gia/pham-quang-hien-17851.html)
4. **[Phạm Văn Ất](https://newshop.vn/tac-gia/pham-quang-hien-17851.html)**[, Lập trình Windows Forms với C#, 2020](https://newshop.vn/tac-gia/pham-quang-hien-17851.html)
5. **[Nguyễn Tấn Trần Minh Khang](https://newshop.vn/tac-gia/pham-quang-hien-17851.html)**[, Hướng dẫn học lập trình C# và ứng dụng, 2007](https://newshop.vn/tac-gia/pham-quang-hien-17851.html)
6. **[Chris Sells](https://newshop.vn/tac-gia/pham-quang-hien-17851.html)**[, Windows Forms Programming in C#, 2003](https://newshop.vn/tac-gia/pham-quang-hien-17851.html)
7. [https://learn.microsoft.com/en-us/dotnet/desktop/winforms/ "Microsoft Learn Challenge" - truy cập vào ngày 18/12/2024](https://newshop.vn/tac-gia/pham-quang-hien-17851.html)

[8].https://www.geeksforgeeks.org/introduction-to-c-sharp-windows-forms-applications

“Windows Forms C# Tutorial” – truy cập vào ngày 02/01/2025.

[9] https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-9.0&tabs=visual-studio

“.NET Core API” - truy cập vào ngày 01/01/2025.

[10]. <https://www.w3schools.com/cs/index.php> “C# Tutorial”– truy cập vào ngày 01/01/2025.