**Ảnh có chứa quảng trường

Mô tả được tạo tự động**

**HÀ NỘI - 2023**

**Giảng viên hướng dẫn :** ThS**.** Bùi Thu Hải

**Nhóm sinh viên thực hiện :**

|  |  |  |
| --- | --- | --- |
| **Mã sinh viên** | **Họ & Tên** | **Lớp** |
| 21103101122 | Lê Văn Thông | DHTI15A18HN |
| 20103100244 | Mai Tạ An Giang | DHTI14A5HN |
| 21103101150 | Trần Thị Hoài Thu | DHTI15A18HN |
| 21103100459 | Lê Quốc Huy | DHTI15A8HN |

***Đề tài :***

**XÂY DỰNG PHẦN MỀM QUẢN LÝ SHOP GIÀY STORESHOES**

**BÁO CÁO HỌC PHẦN ĐỒ ÁN 1**

|  |  |
| --- | --- |
| **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC KINH TẾ**  **KỸ THUẬT CÔNG NGHIỆP** | **KHOA CÔNG NGHỆ THÔNG TIN** |

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC KINH TẾ - KỸ THUẬT CÔNG NGHIỆP**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙧 🕮 🙥**

**LỜI CẢM ƠN**

Trong bối cảnh hiện nay, việc mua hàng trực tuyến đang trở thành xu hướng chủ đạo. Tuy nhiên, không phải ai cũng có thể dễ dàng tiếp cận và tìm hiểu thông tin sản phẩm mình muốn một cách dễ dàng và nhanh chóng, đặc biệt là đối với mặt hàng thời trang như giày thể thao. Chính vì thế, chúng em đã chọn thực hiện đề tài “Quản lý shop giày Store Shoe” để nghiên cứu và đưa ra giải pháp quản lý hiệu quả cho cửa hàng giày Store Shoe.

Trong bài báo cáo này, chúng em sẽ đề cập đến những kiến thức cũng như kinh nghiệm để quản lý một cửa hàng giày Store Shoe hiệu quả. Chúng em cũng sẽ đưa ra những khái niệm cơ bản cũng như các yêu cầu cần thiết để quản lý cửa hàng, ngoài ra còn có những công nghệ, giải pháp phần mềm để hoàn thiện cửa hàng một cách tối ưu và hiệu quả nhất.

Chúng em hi vọng rằng bài báo cáo của chúng em có thể giúp cho mọi người nắm bắt được kiến thức cơ bản, cũng như có một cái nhìn khách quan về việc quản lý cửa hàng giày Store Shoe . Từ đó, mọi người có thể đánh giá và tiếp thu thêm nhiều kiến thức để tìm kiếm, lựa chọn một cửa hàng phù hợp với bản thân mình.

Với sự hướng dẫn cũng như chỉ dạy nhiệt tình của giảng viên Bùi Thu Hải, chúng em đã hoàn thành bài báo cáo đồ án này một cách nhanh chóng và hiệu quả nhất. Do chưa có nhiều kinh nghiệm nên bài báo cáo của chúng em sẽ không tránh khỏi những thiếu sót trong quá trình thực hiện. Chúng em rất mong sẽ nhận được sự thông cảm cũng như những lời nhận xét từ giảng viên, để chúng em có thể tiến bộ và hoàn thiện hơn trong những lần sắp tới.

*Chúng em xin chân thành cảm ơn!*

***Nhóm sinh viên***

Lê Văn Thông

Mai Tạ An Giang

Trần Thị Hoài Thu

Lê Quốc Huy

**LỜI MỞ ĐẦU**

Trong thế giới ngày nay, giày thể thao không chỉ đơn thuần là một phụ kiện thời trang mà còn là một biểu tượng của phong cách sống, sự năng động và sức khỏe. Sự phổ biến của giày thể thao đã tạo ra một thị trường lớn và đa dạng, với nhiều loại giày khác nhau phục vụ cho nhiều mục đích khác nhau từ chơi thể thao, đi chơi, đến làm việc.

Tuy nhiên, với sự phát triển không ngừng của công nghệ và sự cạnh tranh gay gắt trong thị trường, việc quản lý một cửa hàng giày đòi hỏi sự linh hoạt và hiệu quả hơn bao giờ hết. Điều này đặt ra nhu cầu cho một hệ thống quản lý hiệu quả, có thể giúp cửa hàng theo dõi tồn kho, quản lý đơn hàng, và cung cấp dịch vụ tốt nhất cho khách hàng.

Đồ án của chúng tôi, “Quản lý shop Store Shoes”, nhằm mục đích xây dựng một hệ thống quản lý hiệu quả cho cửa hàng giày thể thao. Hệ thống này sẽ giúp cửa hàng theo dõi tồn kho, quản lý đơn hàng, và cung cấp dịch vụ tốt nhất cho khách hàng. Chúng tôi sẽ tập trung vào việc phân tích và thiết kế hệ thống, đồng thời cung cấp các giải pháp để đảm bảo rằng hệ thống hoạt động một cách hiệu quả và linh hoạt.

Chúng tôi hy vọng rằng, thông qua việc nghiên cứu và phát triển hệ thống này, chúng tôi sẽ cung cấp một giải pháp quản lý hiệu quả, giúp cửa hàng giày thể thao hoạt động một cách suôn sẻ và phát triển bền vững trong tương lai. Chúng tôi rất mong nhận được sự hỗ trợ và góp ý từ thầy cô và các bạn đồng nghiệp để đồ án của chúng tôi có thể hoàn thiện hơn. Xin chân thành cảm ơn!

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 1](#_Toc154836444)

[1.1 Mục đích chọn đề tài 1](#_Toc154836445)

[1.2 Mục tiêu nghiên cứu 1](#_Toc154836446)

[1.3 Đối tượng và phạm vi nghiên cứu 2](#_Toc154836447)

[1.4 Phương pháp nghiên cứu 2](#_Toc154836448)

[1.4.1 Phân tích và dánh giá 2](#_Toc154836449)

[1.4.2 Khảo sát và thu thập dữ liệu 3](#_Toc154836450)

[1.4.3 Phương pháp thực nghiệm 4](#_Toc154836451)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 5](#_Toc154836452)

[2.1 Giới thiệu về ngôn ngữ C# và thư viện viện lớp đồ họa mã nguồn mở Windowns form 5](#_Toc154836453)

[2.1.1 Giới thiệu về ngôn ngữ C# và công nghệ .NET 5](#_Toc154836454)

[2.1.2 Thư viện viện lớp đồ họa mã nguồn mở Windowns form 9](#_Toc154836455)

[2.1.3 Cài đặt môi trường 10](#_Toc154836456)

[2.1.3 Giới thiệu về thư viện Guna UI2 13](#_Toc154836457)

[2.2 Thiết kế giao diện và trải nghiệm người dùng UI/UX 14](#_Toc154836458)

[2.3 Microsoft SQL Server 16](#_Toc154836459)

[2.3.1 Giới thiệu về Microsoft SQL Server 16](#_Toc154836460)

[2.3.2 Các thành phần của SQL Server 17](#_Toc154836461)

[2.4 ADO.NET 18](#_Toc154836462)

[2.4.1 Giới thiệu về ADO.NET 18](#_Toc154836463)

[2.4.2 Kiến trúc ADO.NET 20](#_Toc154836464)

[2.4.3 Tạo kết nối với DbConnection 21](#_Toc154836465)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 23](#_Toc154836466)

[3.1 Mô tả tổng quan hệ thống 23](#_Toc154836467)

[3.1.1 Nhiệm vụ cơ bản của hệ thống 23](#_Toc154836468)

[3.1.2 Mục tiêu của hệ thống 23](#_Toc154836469)

[3.2 Phân tích luồng dữ liệu và thiết kế 24](#_Toc154836470)

[3.2.1 Sơ đồ phân rã chức năng 24](#_Toc154836471)

[3.2.2 Sơ đồ luồng dữ liệu (Data Flow Diagram – DFD) 26](#_Toc154836472)

[CHƯƠNG 4: THIẾT KẾ HỆ THỐNG 33](#_Toc154836473)

[4.1 Mô hình thực thể liên hệ( mô hình ER) 33](#_Toc154836474)

[4.1.1 Vai trò của mô hình ER trong quá trình thiết kế cơ sở dữ liệu 33](#_Toc154836475)

[4.1.3 Xây dựng mô hình ER 35](#_Toc154836476)

[4.2 Thiết kế cơ sở dữ liệu 35](#_Toc154836477)

[4.2.1 Xây dựng mô hình liên kết thực thể ER 35](#_Toc154836478)

[4.3 Mô hình quan hệ 43](#_Toc154836479)

[4.4 Thiết kế cơ sở dữ liệu 44](#_Toc154836480)

[4.4.1 Thiết kế các bảng trong cơ sở dữ liệu 44](#_Toc154836481)

[4.4.2 Mô hình Diagram 48](#_Toc154836482)

[4.5 Thiết kế giao diện 49](#_Toc154836483)

[4.5.1 Giao diện màn hình đăng nhập 49](#_Toc154836484)

[4.5.2 Giao diện chính của chương trình 50](#_Toc154836485)

[4.5.3 Giao diện màn hình tất cả đơn hàng 51](#_Toc154836486)

[4.5.4 Giao diện màn hình bán hàng 52](#_Toc154836487)

[4.5.5 Giao diện màn hình nhập hàng 53](#_Toc154836488)

[4.5.6 Giao diện màn hình sản phẩm 55](#_Toc154836489)

[4.5.7 Giao diện màn hình quản lý nhà cung cấp 57](#_Toc154836490)

[4.5.8 Giao diện màn hình quản lý nhân viên 59](#_Toc154836491)

[4.5.9 Giao diện màn hình tài khoản và mật khẩu 60](#_Toc154836492)

[CHƯƠNG 5: KẾT QUẢ THỰC NGHIỆM 62](#_Toc154836493)

[5.1 Giao diện màn hình đăng nhập 62](#_Toc154836494)

[5.2 Giao diện chính của chương trình 63](#_Toc154836495)

[5.2.1 Giao diện thanh điều hướng 63](#_Toc154836496)

[5.2.2 Giao diện khi chọn điều hướng sản phẩm 64](#_Toc154836497)

[5.2.3 Giao diện khi chọn điều hướng đơn hàng 65](#_Toc154836498)

[5.2.4 Giao diện khi chọn điều hướng Nhà cung cấp 68](#_Toc154836499)

[5.2.5 Giao diện khi chọn điều hướng nhân viên 69](#_Toc154836500)

[5.2.6 Giao diện khi chọn điều hướng cấu hình 70](#_Toc154836501)

[KẾT LUẬN VÀ KIẾN NGHỊ 71](#_Toc154836502)

[TÀI LIỆU THAM KHẢO 72](#_Toc154836503)

**DANH MỤC HÌNH ẢNH**

[Hình 2.1: Ngôn ngữ lập trình C# 5](#_Toc154836624)

[Hình 2.2: Tiện ích của C# 6](#_Toc154836625)

[Hình 2.3 Cài đặt phần mềm Víual Studio 10](#_Toc154836626)

[Hình 2.4 Giao diện khi file cài đặt phầm mềm Víual Studio đang chạy 11](#_Toc154836627)

[Hình 2.5 Giao diện các phiên bản khác nhau của Visual Studio 11](#_Toc154836628)

[Hình 2.6 Giao diện chọn các Workloads để cài đặt Visual Studio 12](#_Toc154836629)

[Hình 2.7 Giao diện phần mềm Visual Studio 2019 12](#_Toc154836630)

[Hình 2.8 Guana Framework 13](#_Toc154836631)

[Hình 2.9 Thiết kế UI/UX 15](#_Toc154836632)

[Hình 2.10 ADO.NET của .NET framework 19](#_Toc154836633)

[Hình 2.11Kiến trúc ADO.NET 20](#_Toc154836634)

[Hình 3.1 Sơ đồ phân rã chức năng 25](#_Toc154836635)

[Hình 3.2 Sơ đồ luồng dữ liệu mức ngữ cảnh 27](#_Toc154836636)

[Hình 3.3 Sơ đồ dữ liệu mức đỉnh 28](#_Toc154836637)

[Hình 3.4. Sơ đồ mức dưới đỉnh của chức năng Quản Lý Nhập Hàng 29](#_Toc154836638)

[Hình 3.5 Sơ đồ mức dưới đỉnh của chức năng Quản Lý Bán Hàng 30](#_Toc154836639)

[Hình 3.6 Sơ đồ mức dưới đỉnh của chức năng Quản Lý Nhân Viên 31](#_Toc154836640)

[Hình 3.7 Sơ đồ mức dưới đỉnh của chức năng Báo Cáo Thống Kê 32](#_Toc154836641)

[Hình 4.1 : Sơ đồ ER 42](#_Toc154836642)

[Hình 4.2 Mối quan hệ giữa các bảng trong cơ sở dữ liệu 48](#_Toc154836643)

[Hình 4.3: Giao diện đăng nhập 49](#_Toc154836644)

[Hình 4.4 Giao diện chương trình 50](#_Toc154836645)

[Hình 4.5 Giao diện tất cả đơn hàng 51](#_Toc154836646)

[Hình 4.6 Giao diện màn hình bán hàng 52](#_Toc154836647)

[Hình 4.7 Giao diện màn hình nhập hàng 53](#_Toc154836648)

[Hình 4.8 Giao diện màn hình sản phẩm 55](#_Toc154836649)

[Hình 4.9 Giao diện quản lý nhà cung cấp 57](#_Toc154836650)

[Hình 4.10 Giao diện quản lý nhân viên 59](#_Toc154836651)

[Hình 4.11 Giao diện quản lý tài khoản 60](#_Toc154836652)

[Hình 5.1 Giao diện màn hình đăng nhập 62](#_Toc154836653)

[Hình 5.2 Giao diện thanh điều hướng 63](#_Toc154836654)

[Hình 5.3 Giao diện điều hướng sản phẩm 64](#_Toc154836655)

[Hình 5.4 Giao diện điều hướng tất cả đơn hàng 65](#_Toc154836656)

[Hình 5.5 Giao diện điều hướng bán hàng 66](#_Toc154836657)

[Hình 5.6 Giao diện điều hướng nhập hàng 66](#_Toc154836658)

[Hình 5.7 Giao diện điều hướng Nhà cung cấp 68](#_Toc154836659)

[Hình 5.8 Giao diện điều hướng nhân viên 69](#_Toc154836660)

[Hình 5.9 Giao diện điều hướng tài khoản 70](#_Toc154836661)

**DANH MỤC BẢNG BIỂU**

[Bảng 2-1 Phân tích điểm mạnh và điểm yếu của ngôn ngữ C# 16](#_Toc154836922)

[Bảng 2-2 Phân tích vai trò của UI/UX trong thiết kế phần mềm 24](#_Toc154836923)

[Bảng 2-3 Một số câu lệnh SQL cơ bản 25](#_Toc154836924)

[Bảng 2-4 Các key trong chuỗi kết nối SQL Server 29](#_Toc154836925)

[Bảng 2-5 Mở và đóng kết nối đến cơ sở dữ liêu SQL Server cơ bản 30](#_Toc154836926)

[Bảng 3-1 : Đặc điểm và mục đích của mô hình phân rã chức năng 33](#_Toc154836927)

[Bảng 4-1. Bảng TaiKhoan 52](#_Toc154836928)

[Bảng 4-2. Bảng NhanVien 52](#_Toc154836929)

[Bảng 4-3. Bảng NhaCungCap 53](#_Toc154836930)

[Bảng 4-4. Bảng LoaiGiay 53](#_Toc154836931)

[Bảng 4-5. Bảng SanPhamGiay 53](#_Toc154836932)

[Bảng 4-6. Bảng SizeGiay 54](#_Toc154836933)

[Bảng 4-7 Bảng HoaDonMua 54](#_Toc154836934)

[Bảng 4-8. Bảng ChiTietHoaDonMua 54](#_Toc154836935)

[Bảng 4-9. Bảng HoaDonBan 55](#_Toc154836936)

[Bảng 4-10. Bảng ChiTietHoaDonBan 55](#_Toc154836937)

[Bảng 4-11: Điều kiện xử lí giao diện đăng nhập 58](#_Toc154836938)

[Bảng 4-12: Điều kiện xử lí giao diện tất cả dơn hàng 59](#_Toc154836939)

[Bảng 4-13: Điều kiện xử lí giao diện bán hàng 61](#_Toc154836940)

[Bảng 4-14: Điều kiện xử lí giao diện màn hình nhập hàng 62](#_Toc154836941)

[Bảng 4-15: Điều kiện xử lí giao diện màn hình sản phẩm 64](#_Toc154836942)

[Bảng 4-16: Điều kiện xử lí giao diện quản lý nhà cung cấp 65](#_Toc154836943)

[Bảng 4-17: Điều kiện xử lí giao diện quản lý Nhân viên 67](#_Toc154836944)

# 

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

## 1.1 Mục đích chọn đề tài

Trong thời đại công nghệ số hiện nay, việc kinh doanh giày dép trên mạng là một xu hướng không thể bỏ qua. Theo thống kê của Hiệp hội Thương mại Điện tử Việt Nam, doanh thu bán hàng trực tuyến của ngành giày dép đạt 1,2 tỷ USD vào năm 2022, chiếm 12% tổng doanh thu của ngành. Điều này cho thấy nhu cầu mua sắm giày dép online của khách hàng ngày càng cao, đặc biệt là trong bối cảnh dịch bệnh Covid-19 vẫn chưa được kiểm soát hoàn toàn

Tuy nhiên, để kinh doanh giày dép thành công, các cửa hàng cần có một hệ thống quản lý chuyên nghiệp và hiệu quả. Đó là lý do tôi chọn đề tài quản lý shop giày StoreShoes để nghiên cứu và phát triển. Đề tài này nhằm xây dựng một hệ thống quản lý giày dép cho các cửa hàng, với các chức năng như quản lý kho hàng, doanh thu, lãi lỗ, nhân viên, khuyến mãi, hóa đơn, phiếu nhập, thống kê, ... Hệ thống này sẽ giúp các cửa hàng tiết kiệm thời gian, chi phí, nguồn nhân lực và tăng cường sự hài lòng của khách hàng. Ngoài ra, hệ thống này cũng có thiết kế đẹp mắt, thân thiện với người dùng và tương thích với nhiều thiết bị

Với những lợi ích trên, tôi tin rằng đề tài quản lý shop giày StoreShoes là một đề tài có ý nghĩa thực tiễn và hữu ích cho các cửa hàng kinh doanh giày dép trên thị trường hiện nay.

## 1.2 Mục tiêu nghiên cứu

Mục tiêu chính của đề tài là xây dựng một phần mềm quản lý shop giày StoreShoes, với các chức năng như quản lý kho hàng, doanh thu, lãi lỗ, nhân viên, khách hàng, khuyến mãi, hóa đơn, phiếu nhập, thống kê, ... Phần mềm này sẽ giúp các cửa hàng tiết kiệm thời gian, chi phí, nguồn nhân lực và tăng cường sự hài lòng của khách hàng. Ngoài ra, phần mềm này cũng có giao diện đẹp mắt, thân thiện với người dùng và tương thích với nhiều thiết bị. Đề tài cũng nghiên cứu về nhu cầu thị trường, nhu cầu khách hàng, cạnh tranh và chiến lược kinh doanh của các cửa hàng kinh doanh giày dép tại Việt Nam cụ thể bao gồm:

1. Phù hợp với điều kiện thực tiễn, khả thi
2. cần đảm bảo 5 tiêu chuẩn: “SMART”: S (Specific) : Cụ thể và rõ ràng. M (Measurable) : Có thể đo lường được. A (Achievable) : Khả thi. R (Reasonable) : Hợp lý. T (Timely) – Có thời gian quy định cụ thể
3. Mục tiêu nghiên cứu cần thể hiện được chiến lượng, kế hoạch nghiên cứu và tư duy logic của đề tài nghiên cứu

## 1.3 Đối tượng và phạm vi nghiên cứu

**Đối tượng nghiên cứu**: Phần mềm quản lý shop giày STOTESHOES

**Phạm vi nghiên cứu của đề tài**:

* Tìm hiểu về lập trình trên C# Winform tạo giao diện quét mã barcode và hiển thị thông tin đợn hàng.
* Tìm hiểu về hệ quản trị cơ sở dữ liệu SQL SERVER
* Tìm hiểu về UI (User Interface) là giao diện người dùng và UX (User Experience) là trải nghiệm người dùng
* Tìm hiểu về công nghệ truy cập dữ liệu từ Microsoft.NET Framework ADO.NET
* Tìm hiểu về  bộ công cụ thiết kế giao diện người dùng và trải nghiệm người dùng cho các nhà phát triển ứng dụng máy tính để bàn, báo cáo, trực quan hóa dữ liệu GunaFramework

## 1.4 Phương pháp nghiên cứu

### 1.4.1 Phân tích và dánh giá

**Tầm quan trọng của việc sử dụng phần mềm quản lý shop giày:**Giày dép là mặt hàng thời trang cực kì đa dạng, từ kiểu dáng, thương hiệu, nguyên liệu và xuất xứ. Để đưa hoạt động kinh doanh trở nên quy củ và chuyên nghiệp hơn thì việc quản lý bán hàng khoa học là vô cùng cần thiết.

**Thách thức quản lý cửa hàng giày:**

* **Quản lý hàng hóa**: Việc quản lý hàng hóa trong cửa hàng giày có thể trở nên phức tạp do sự đa dạng về mẫu mã, kích cỡ và màu sắc.
* **Quản lý nhà cung cấp:** Việc tìm kiếm, đánh giá và duy trì mối quan hệ với các nhà cung cấp là một thách thức không nhỏ.
* **Quản lý hóa đơn nhập hàng:** Việc theo dõi và kiểm soát hóa đơn nhập hàng đòi hỏi sự tỉ mỉ và chính xác.
* **Quản lý hóa đơn bán hàng:** Việc quản lý hóa đơn bán hàng cần sự minh bạch và chính xác để đảm bảo rằng tất cả các giao dịch đều được ghi nhận đúng.
* **Quản lý nhân sự:** Việc tuyển dụng, đào tạo và giữ chân nhân viên là một thách thức lớn.

**Giải pháp quản lý cửa hàng giày**

* **Sử dụng phần mềm quản lý:** Phần mềm quản lý cửa hàng giày có thể giúp cửa hàng tự động hóa nhiều quy trình, từ quản lý hàng hóa, quản lý nhân sự đến quản lý khách hàng.
* **Đào tạo nhân viên:** Việc đào tạo nhân viên về kỹ năng bán hàng và kỹ năng phục vụ khách hàng là rất quan trọng.
* **Chăm sóc khách hàng:** Cửa hàng cần xây dựng chính sách chăm sóc khách hàng tốt để thu hút và giữ chân khách hàng.

### 1.4.2 Khảo sát và thu thập dữ liệu

Trong quá trình nghiên cứu và phát triển phần mềm quản lý cửa hàng giày, việc khảo sát và thu thập dữ liệu là một bước quan trọng. Dưới đây là một số phương pháp được sử dụng:

**Khảo sát trực tiếp**: Tiến hành khảo sát trực tiếp tại các cửa hàng giày để hiểu rõ hơn về quy trình hoạt động, nhu cầu và thách thức mà họ đang đối mặt. Các phương pháp khảo sát có thể bao gồm phỏng vấn, quan sát, hoặc sử dụng biểu mẫu khảo sát.

**Thu thập dữ liệu từ nguồn thứ cấp**: Tìm kiếm thông tin từ các nguồn thứ cấp như sách, báo cáo nghiên cứu, bài báo, hoặc dữ liệu trực tuyến để hiểu rõ hơn về thị trường giày dép và xu hướng công nghệ trong ngành này.

**Sử dụng công cụ số hóa**: Công cụ số hóa như Google Forms, SurveyMonkey, hoặc Typeform có thể giúp bạn thu thập dữ liệu một cách hiệu quả và tổ chức dữ liệu một cách dễ dàng.

**Phân tích dữ liệu**: Sau khi thu thập dữ liệu, bạn cần phân tích dữ liệu để rút ra những thông tin hữu ích. Các phương pháp phân tích dữ liệu có thể bao gồm phân tích thống kê, phân tích xu hướng, hoặc phân tích so sánh.

### 1.4.3 Phương pháp thực nghiệm

Phương pháp thực nghiệm là một phần quan trọng của quá trình nghiên cứu, giúp kiểm tra và xác nhận hiệu quả của phần mềm quản lý cửa hàng giày. Dưới đây là một số bước thực hiện:

1. **Thiết kế thí nghiệm**: Trước hết, bạn cần xác định mục tiêu của thí nghiệm. Mục tiêu có thể là kiểm tra tính năng cụ thể của phần mềm, đánh giá hiệu suất của nó trong một tình huống cụ thể, hoặc kiểm tra khả năng tương thích với các hệ thống khác. Bạn cũng cần xác định các biến sẽ được sử dụng trong thí nghiệm, cũng như cách bạn sẽ đo lường và ghi lại kết quả.
2. **Chuẩn bị dữ liệu**: Bước tiếp theo là tạo ra hoặc thu thập dữ liệu thực tế để sử dụng trong thí nghiệm. Điều này có thể bao gồm dữ liệu về sản phẩm, thông tin khách hàng, hoặc dữ liệu về các giao dịch bán hàng. Bạn cần đảm bảo rằng dữ liệu này đại diện cho các tình huống thực tế mà phần mềm sẽ gặp phải khi được triển khai.
3. **Thực hiện thí nghiệm**: Sử dụng phần mềm trong một môi trường kiểm soát để thực hiện các tác vụ cụ thể và thu thập kết quả. Điều này có thể bao gồm việc thực hiện các tác vụ thông qua giao diện người dùng, hoặc việc sử dụng các công cụ tự động hóa để mô phỏng hoạt động của người dùng thực sự.
4. **Phân tích kết quả**: Sử dụng các phương pháp thống kê để phân tích kết quả thí nghiệm và rút ra kết luận. Điều này có thể bao gồm việc so sánh hiệu suất của phần mềm với các tiêu chuẩn công nghiệp, hoặc việc kiểm tra xem liệu phần mềm có đáp ứng được các yêu cầu về chức năng và hiệu suất hay không.
5. **Đánh giá và cải tiến**: Dựa trên kết quả thí nghiệm, đánh giá hiệu quả của phần mềm và xác định các cải tiến cần thiết. Điều này có thể bao gồm việc tinh chỉnh các thuật toán, cải thiện giao diện người dùng, hoặc thay đổi cách phần mềm tương tác với các hệ thống khác.

# CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

## 2.1 Giới thiệu về ngôn ngữ C# và thư viện viện lớp đồ họa mã nguồn mở Windowns form

### 2.1.1 Giới thiệu về ngôn ngữ C# và công nghệ .NET



Hình 2.1: Ngôn ngữ lập trình C#

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

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

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

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

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



Hình 2.2: Tiện ích của C#

**Công nghệ .NET**

.NET là một nền tảng nguồn mở, miễn phí và được Microsoft phát triển để xây dựng, cũng như có thể chạy nhiều loại ứng dụng khác nhau, bao gồm ứng dụng web, phần mềm máy tính để bàn, ứng dụng dành cho thiết bị di động, game…

Khi làm việc cùng .NET, nhà phát triển sẽ được cung cấp một bộ công cụ, thư viện và ngôn ngữ phong phú, giúp việc xây dựng các ứng dụng cho các nền tảng khác nhau trở nên dễ dàng và nhanh chóng hơn.

**Nền tảng .NET bao gồm hai thành phần chính, là:**

.NET Framework: Là một nền tảng trưởng thành đã xuất hiện trong nhiều năm, chủ yếu được sử dụng để xây dựng các ứng dụng máy tính để bàn Windows và ứng dụng web phía Server.

.NET Core: Là một Framework nguồn mở và đa nền tảng có thể được sử dụng để xây dựng các ứng dụng cho Windows, Linux và macOS. Nó cũng được tối ưu hóa cho các ứng dụng dựa trên đám mây.

Nền tảng .NET bao gồm một số ngôn ngữ lập trình, nổi bật nhất là C#, F# và Visual Basic, trong tổng số những ngôn ngữ khác. Ngoài ra, nó cũng bao gồm nhiều công cụ phát triển như Visual Studio, Visual Studio Code và Xamarin Studio, được sử dụng rộng rãi bởi các nhà phát triển trên toàn thế giới.

Một trong những lợi ích chính mà .NET mang lại đó là về mặt hiệu suất và khả năng mở rộng của nó. Các ứng dụng .NET có thể xử lý lưu lượng truy cập cao và các hoạt động phức tạp mà không ảnh hưởng đến tốc độ hoặc độ ổn định. Thêm vào đó, .NET còn cung cấp các tính năng bảo mật nâng cao, làm cho nó trở thành một nền tảng an toàn trong việc xây dựng các ứng dụng xử lý dữ liệu nhạy cảm.

**Mối quan hệ giữa ngôn ngữ C# và công nghệ .NET**

C# là ngôn ngữ lập trình được sử dụng trong nền tảng .NET để xây dựng các ứng dụng. Mặc dù, C# có thể được sử dụng để phát triển các ứng dụng bên ngoài nền tảng .NET, nhưng nó chủ yếu liên quan đến phát triển .NET.

C# là thành phần chính trong nền tảng .NET và là ngôn ngữ chính được sử dụng để phát triển các ứng dụng trên nền tảng này. Nền tảng .NET cung cấp một Framework gồm các thư viện và công cụ hỗ trợ, cho phép các nhà phát triển C# xây dựng các ứng dụng hiệu quả hơn, với hiệu năng và bảo mật tốt hơn.

Mã C# được biên dịch thành một ngôn ngữ trung gian gọi là Common Intermediate Language (CIL), được thực thi bởi môi trường thời gian chạy .NET. Môi trường thời gian chạy cung cấp một số dịch vụ chính như quản lý bộ nhớ, bảo mật và thực thi mã, những dịch vụ này rất quan trọng khi xây dựng các ứng dụng mạnh mẽ và đáng tin cậy.

Các nhà phát triển C# thường sử dụng các công cụ phát triển như Visual Studio hoặc Visual Studio Code (là một phần của nền tảng .NET), để viết, gỡ lỗi và kiểm tra mã của họ. Họ cũng có thể sử dụng nhiều thư viện và Framework do .NET cung cấp để xây dựng các loại ứng dụng khác nhau, chẳng hạn như ứng dụng web, phần mềm máy tính để bàn, ứng dụng dành cho thiết bị di động và game.

Nói đơn giản hơn, C# và .NET có quan hệ mật thiết với nhau, trong đó C# là ngôn ngữ chính được sử dụng để phát triển các ứng dụng trên nền tảng .NET. Cùng với nhau, chúng cung cấp một môi trường mạnh mẽ và linh hoạt hỗ trợ xây dựng hàng loạt các ứng dụng có khả năng mở rộng, an toàn và đáng tin cậy.

Sự khác nhau giữa C# và công nghệ .NET

Mặc dù C# và .NET là hai thực thể có liên quan đến nhau, nhưng lại riêng biệt trong quá trình phát triển phần mềm. Nếu như C# là ngôn ngữ lập trình, thì .NET lại là Framework phần mềm cung cấp thư viện và công cụ để xây dựng ứng dụng.

Cụ thể sự khác nhau giữa ngôn ngữ C# và công nghệ .NET được thể hiện qua những điểm sau:

C# là một ngôn ngữ lập trình hướng đối tượng hiện đại được phát triển bởi Microsoft. Nó được thiết kế đơn giản, hiệu quả và dễ sử dụng. C# còn được sử dụng để viết mã cho .NET, nhưng nó cũng có thể được sử dụng bên ngoài .NET Framework, chẳng hạn như trên Linux và macOS, với sự trợ giúp của các công cụ nguồn mở như .NET Core.

Mặt khác, .NET lại là một Framework phần mềm do Microsoft phát triển để xây dựng các ứng dụng dựa trên Windows. Nó bao gồm một loạt các thư viện và công cụ mà các nhà phát triển có thể sử dụng để xây dựng các ứng dụng dành cho máy tính để bàn, ứng dụng web và ứng dụng dành cho mobile. Ngoài ra, .NET Framework cũng cung cấp một môi trường thời gian chạy bao gồm Common Language Runtime (CLR), cho phép các ứng dụng chạy trên bất kỳ máy nào đã cài đặt .NET framework.

Mặc dù C# là ngôn ngữ phổ biến được sử dụng để phát triển các ứng dụng trên .NET Framework, nhưng cũng có thể sử dụng các ngôn ngữ khác, chẳng hạn như F# và Visual Basic.

Chưa dừng lại, .NET cũng bao gồm các thành phần khác, chẳng hạn như Windows Presentation Foundation (WPF), Windows Forms và ASP.NET, được sử dụng để xây dựng các loại ứng dụng khác nhau.

Bảng 2-1 Phân tích điểm mạnh và điểm yếu của ngôn ngữ C#

|  |  |
| --- | --- |
| **Điểm mạnh** | **Điểm yếu** |
| **Dễ học**: C# có nhiều tính năng giúp nó tương đối dễ học so với một số ngôn ngữ lập trình khác.  **Phát triển ứng dụng và trò chơi**: C# đặc biệt tốt trong việc phát triển ứng dụng máy tính để bàn Windows và trò chơi. Nó cũng có thể được sử dụng để phát triển ứng dụng và đã trở nên phổ biến hơn trong phát triển di động.  **Sức mạnh và linh hoạt**: C# là một ngôn ngữ lập trình mạnh mẽ, linh hoạt và được hỗ trợ tốt, dẫn đến việc nó trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới trong một khoảng thời gian tương đối ngắn.. | **Phức tạp:** Mặc dù C# có nhiều tính năng giúp nó dễ học, nhưng đây vẫn là một ngôn ngữ lập trình phức tạp và việc thành thạo nó có thể mất nhiều thời gian và công sức hơn so với một số ngôn ngữ đơn giản hơn như Python.  **Yêu cầu .NET Framework:** C# yêu cầu .NET Framework trên Windows để hoạt động. |

### 2.1.2 Thư viện viện lớp đồ họa mã nguồn mở Windowns form

WinForms là một phần của Microsoft .NET Framework, và nó cung cấp một mô hình lập trình để phát triển ứng dụng Windows dựa trên giao diện đồ họa người dùng (GUI). WinForms (Windows Forms) đã trở thành một công nghệ quan trọng cho việc phát triển ứng dụng Windows trong môi trường .NET. Dưới đây là một giới thiệu cơ bản về WinForms:

**Khái Niệm Cơ Bản:**

WinForms là một bộ công cụ lập trình (framework) cho việc xây dựng ứng dụng Windows dựa trên Windows Presentation Foundation (WPF).

Nó cung cấp một loạt các điều khiển (controls) như nút, hộp văn bản, danh sách, bảng, và các điều khiển khác để xây dựng giao diện người dùng.

**Ngôn Ngữ Lập Trình:**

WinForms thường được sử dụng với các ngôn ngữ lập trình như C# hoặc VB.NET.

Các ứng dụng WinForms có thể được xây dựng bằng cách sử dụng Visual Studio, môi trường phát triển tích hợp (IDE) mạnh mẽ của Microsoft.

**Thiết Kế Giao Diện Người Dùng**:

WinForms cho phép bạn thiết kế giao diện người dùng bằng cách kéo và thả các điều khiển từ hộp công cụ vào cửa sổ thiết kế.

Bạn có thể định dạng và tùy chỉnh giao diện người dùng dễ dàng thông qua giao diện người dùng của Visual Studio.

**Sự Kiện và Xử Lý Sự Kiện:**

WinForms hỗ trợ mô hình lập trình sự kiện (event-driven programming), nơi bạn có thể liên kết các xử lý sự kiện với các hành động người dùng.

Bạn có thể đáp ứng với các sự kiện như nhấp chuột, phím được nhấn, thay đổi trạng thái điều khiển, và nhiều sự kiện khác.

**Binding Dữ Liệu:**

WinForms hỗ trợ data binding, cho phép bạn liên kết các điều khiển trực tiếp với dữ liệu.

Điều này giúp giảm thời gian và công sức khi quản lý dữ liệu và đồng bộ hóa giữa giao diện người dùng và dữ liệu.

**Hỗ Trợ Đa Nhiệm:**

WinForms hỗ trợ các chức năng đa nhiệm thông qua việc sử dụng luồng (threads) và các tiến trình (processes).

Điều này cho phép ứng dụng xử lý nhiều công việc cùng một lúc mà không làm đóng băng giao diện người dùng.

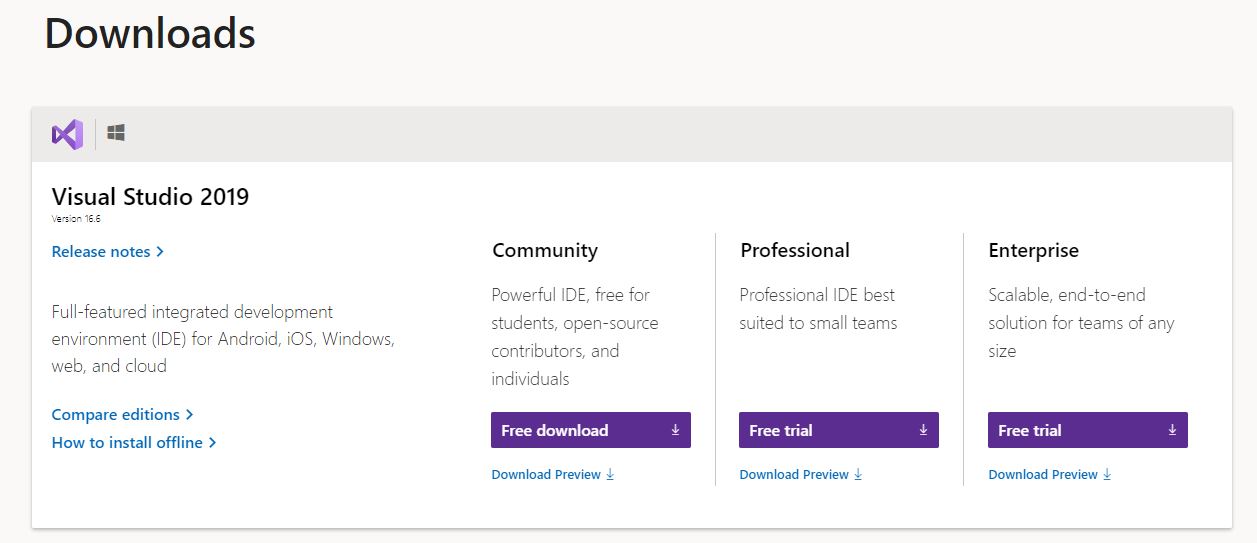
Lập Trình Dễ Dàng và Nhanh Chóng:

WinForms được đánh giá cao về sự dễ học và triển khai, làm cho nó là một lựa chọn phổ biến cho việc phát triển các ứng dụng Windows nhanh chóng.

WinForms tiếp tục được sử dụng rộng rãi trong các dự án phần mềm Windows truyền thống và vẫn là một lựa chọn mạnh mẽ cho những người phát triển ứng dụng .NET dựa trên Windows.

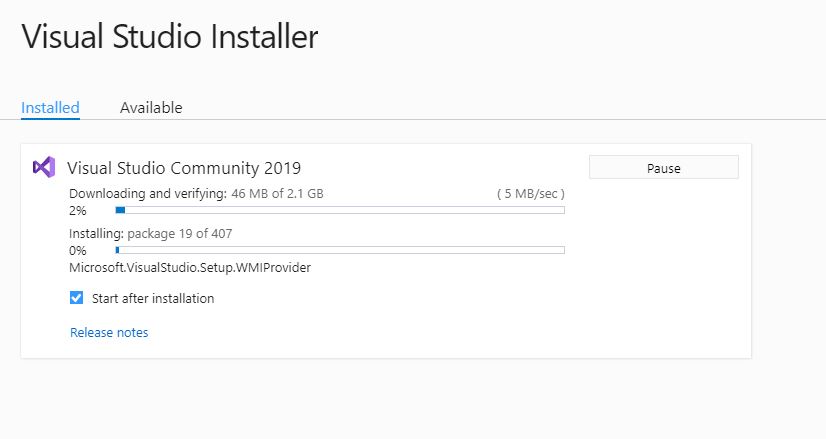
### 2.1.3 Cài đặt môi trường

**Bước 1:** Tải phần mềm visualstudio bản Community nhé, vì nó miễn phí, trong quá trình học tập thì bản này là dư dùng cho mình rồi.

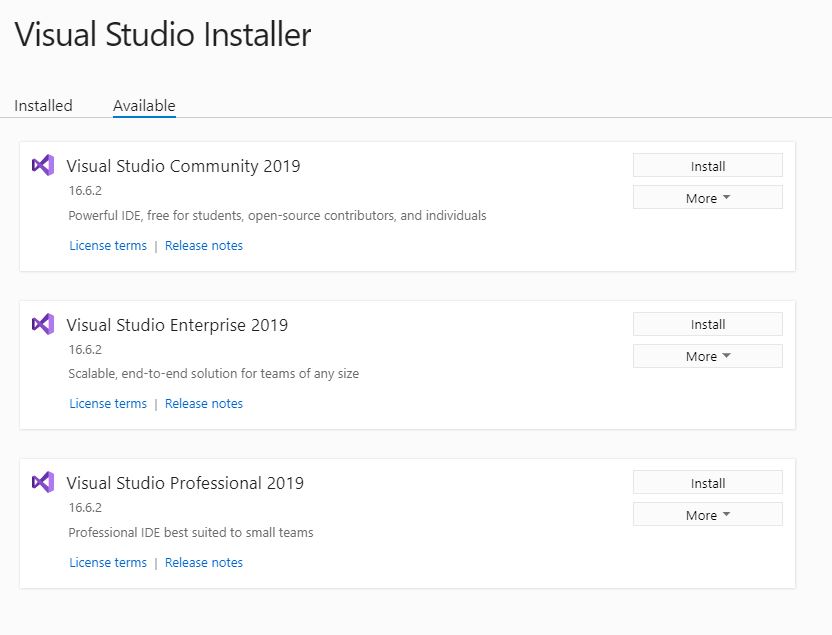


Hình 2.3 Cài đặt phần mềm Víual Studio

**Bước 2**: Cài đặt Sau khi tải về bạn nhấp double click vào file cài đặt và cho nó chạy.

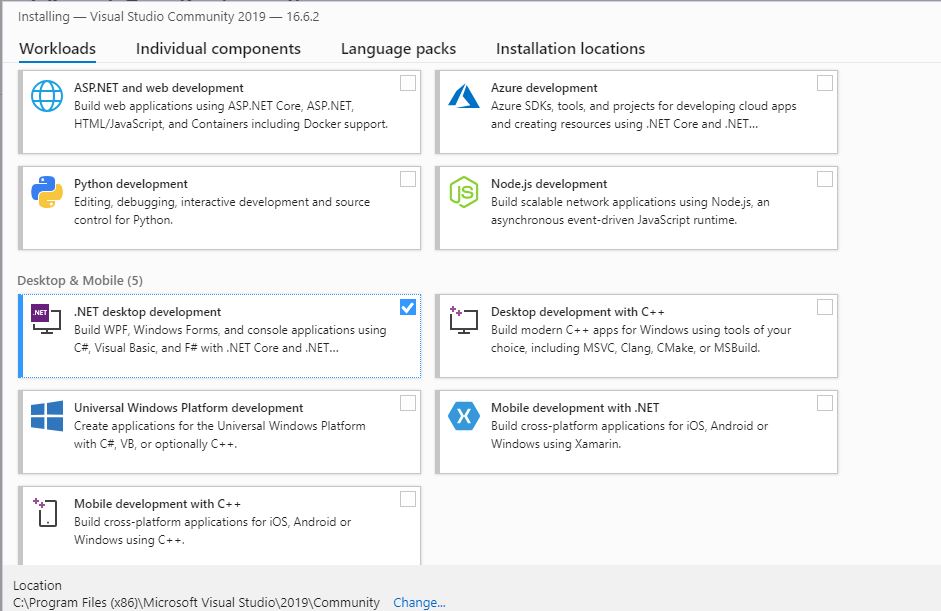


Hình 2.4 Giao diện khi file cài đặt phầm mềm Víual Studio đang chạy

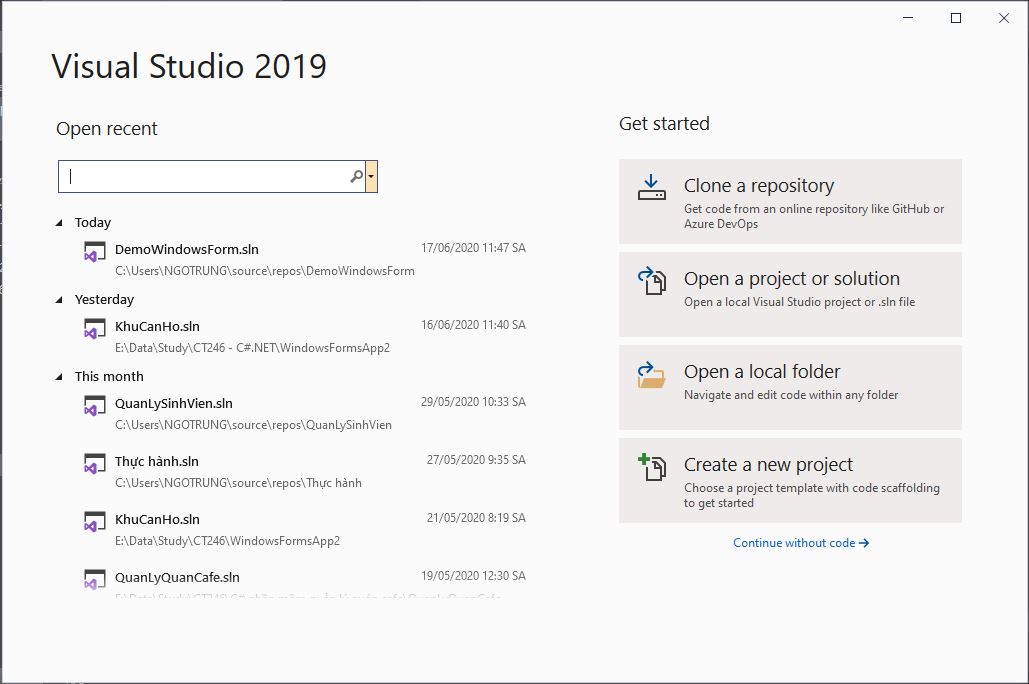
**Bước 3**: Vẫn chọn bản Community nhé. Rồi nhấn Next.

Hình 2.5 Giao diện các phiên bản khác nhau của Visual Studio

**Bước 4**: Bạn chọn vào “**.**NET Desktop Development”. Nếu bạn lập trình phần nào khác thì cứ chọn vào để nó cài đặt luôn. Lưu ý là khá nặng đấy.



Hình 2.6 Giao diện chọn các Workloads để cài đặt Visual Studio

Đến bước này thì bạn đã xong rồi chỉ cần đợi nó cài đặt thôi.  Khi cài đặt xong sẽ hiển thị như hình bên dưới (ở đây nếu bạn chưa có mở Project nào thì phần Open recent nó sẽ trống nhé). 

Hình 2.7 Giao diện phần mềm Visual Studio 2019

### 2.1.3 Giới thiệu về thư viện Guna UI2

“Guna.UI.dll” là một thư viện liên kết động liên quan đến Khung Guna.UI, được phát triển bởi Guna Technology. Khung Guna.UI được biết đến với việc cung cấp cho các nhà phát triển một bộ công cụ và điều khiển để nâng cao vẻ ngoài và chức năng của các ứng dụng Windows Forms của họ. Khung công việc này về cơ bản làm cho việc thiết kế giao diện người dùng hấp dẫn về mặt hình ảnh trở nên dễ dàng hơn mà không cần phải bắt đầu từ đầu.

Với một loạt các điều khiển và thành phần được thiết kế trước, Khung Guna.UI cho phép phát triển nhanh chóng trong khi đảm bảo sản phẩm cuối cùng có vẻ ngoài hiện đại và tinh xảo. Các điều khiển này có thể bao gồm từ nút nâng cao, hộp văn bản, và thanh trượt đến các yếu tố phức tạp hơn như bảng thông báo và hộp thoại tùy chỉnh.

Kết hợp “Guna.UI.dll” vào một dự án cung cấp truy cập trực tiếp vào các điều khiển và thành phần này. Nếu bạn thấy DLL này trên hệ thống của mình, nó gợi ý rằng một ứng dụng phần mềm mà bạn đang sử dụng đã được xây dựng bằng Khung Guna.UI, và DLL này đóng một vai trò quan trọng trong việc hiển thị và quản lý các yếu tố giao diện người dùng tùy chỉnh của ứng dụng đó. Nó nhấn mạnh xu hướng trong phát triển phần mềm nơi mà thẩm mỹ và trải nghiệm người dùng được coi là quan trọng, và các khung công việc như Guna.UI giúp đạt được điều đó mà không tạo ra gánh nặng đáng kể cho quá trình phát triển.



Hình 2.8 Guana Framework

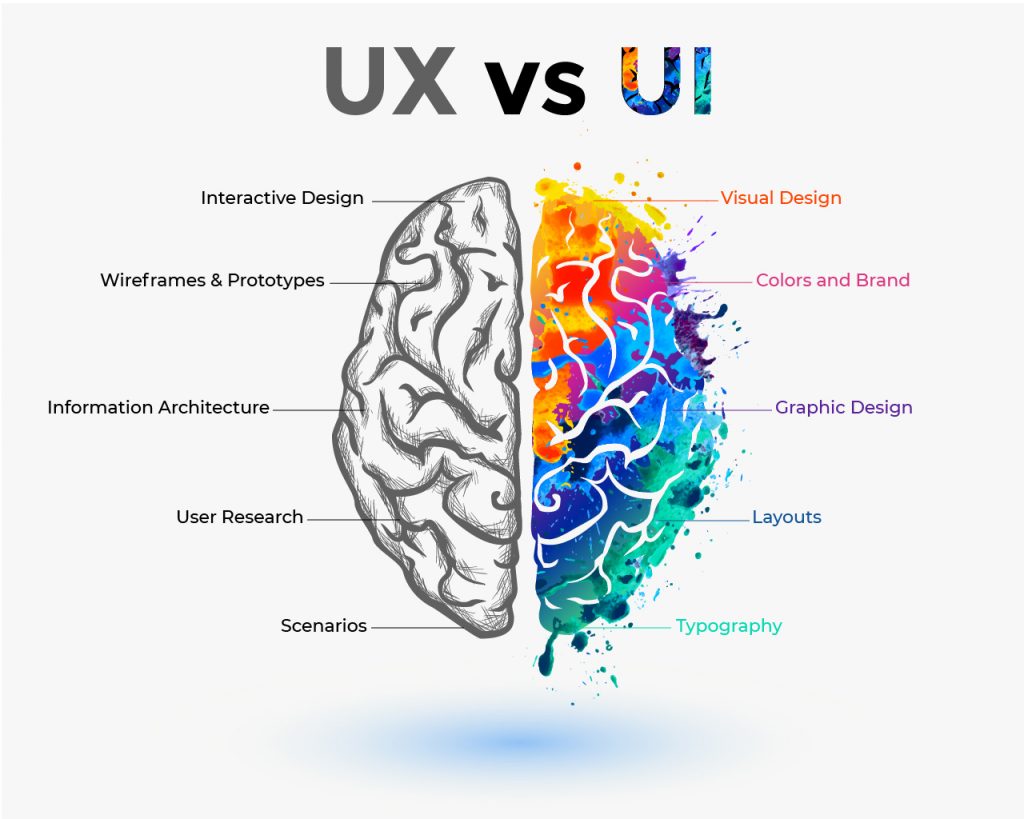
Guna UI2 là một công cụ mạnh mẽ cho việc phát triển ứng dụng Windows Forms, giúp tạo ra các ứng dụng với giao diện người dùng chuyên nghiệp và thân thiện. Tuy nhiên, như bất kỳ công cụ nào khác, việc sử dụng nó hiệu quả đòi hỏi sự hiểu biết về cách làm việc của nó và cách tận dụng tối đa các tính năng mà nó cung cấp.Sau đây là những điểm nổi bật của framework này:

1. **Đa dạng các điều khiển**: Guna UI2 cung cấp một loạt các điều khiển UI như nút, hộp kiểm, thanh trượt, hộp văn bản, biểu đồ, và nhiều hơn nữa. Điều này giúp tạo ra các ứng dụng với giao diện người dùng phong phú và tương tác.
2. **Tùy chỉnh linh hoạt**: Các điều khiển trong Guna UI2 có thể được tùy chỉnh một cách linh hoạt, cho phép bạn thay đổi màu sắc, kích thước, hình dạng, và nhiều thuộc tính khác.
3. **Hỗ trợ .NET Framework và .NET Core**: Guna UI2 hỗ trợ .NET Framework v4.0 trở lên, .NETCoreApp 3.1 trở lên và .NET 6, .NET 7. Điều này giúp nó tương thích với nhiều phiên bản của .NET.
4. **Dễ dàng tích hợp**: Bạn có thể dễ dàng thêm Guna UI2 vào dự án của mình thông qua NuGet Package Manager. Sau khi đã cài đặt, bạn có thể sử dụng các điều khiển DLL của Guna.UI để xây dựng giao diện người dùng.
5. **Tăng năng suất phát triển**: Guna UI2 giúp giảm thời gian phát triển bằng cách cung cấp các điều khiển sẵn có, giúp bạn tập trung vào logic ứng dụng hơn là việc xây dựng giao diện từ đầu.

## 2.2 Thiết kế giao diện và trải nghiệm người dùng UI/UX

**UI (User Interface):** UI liên quan đến giao diện của sản phẩm số mà người dùng tương tác. Mục tiêu chính của UI là thu hút sự chú ý của người dùng bằng cách làm cho giao diện người dùng trở nên hấp dẫn và thẩm mỹ1. UI đóng vai trò quan trọng trong việc tạo ra trải nghiệm người dùng tích cực, tăng sự tương tác và làm cho thương hiệu trở nên dễ nhận biết1.

**UX (User Experience):** UX liên quan đến trải nghiệm tổng thể mà người dùng có được khi sử dụng sản phẩm. UX tập trung vào việc tạo ra các sản phẩm dễ sử dụng và dễ điều hướng, cải thiện sự hài lòng của người dùng. UX giúp tăng sự tương tác, cải thiện hình ảnh thương hiệu và tạo ra lợi thế cạnh tranh.



Hình 2.9 Thiết kế UI/UX

**Nguyên tắc thiết kế UI/UX:**

* Đơn giản (Simplicity): Thiết kế nên dễ sử dụng và điều hướng, đòi hỏi nỗ lực tối thiểu.
* Thiết kế tập trung vào người dùng (User-centered Design): UI nên được thiết kế theo cách tập trung vào nhu cầu và sở thích của người dùng.
* Tính nhất quán (Consistency): Thiết kế nên nhất quán về cách hoạt động và hình thức trên tất cả các trang/màn hình và sản phẩm trong cùng một nhãn hiệu hoặc gia đình sản phẩm.
* Hệ thống phân cấp (Hierarchy): Hệ thống phân cấp định hình cách người dùng điều hướng sản phẩm.

Để tạo ra một sản phẩm số thành công, cả UI và UX đều quan trọng. Một giao diện người dùng đẹp mắt (UI tốt) sẽ thu hút người dùng, trong khi một trải nghiệm người dùng tốt (UX tốt) sẽ giữ chân họ.

Bảng 2-2 Phân tích vai trò của UI/UX trong thiết kế phần mềm

|  |  |
| --- | --- |
| **UI (User Interface)** | **UX(User Experience)** |
| **Trực quan:** Thiết kế UI tốt có thể được nắm bắt nhanh chóng và dễ dàng bởi người dùng, mà không quá phức tạp hoặc dài dòng.  **Quen thuộc**: Thiết kế UI tốt làm cho mọi thứ trở nên ‘quen thuộc’. Khi bạn quen thuộc với một thứ gì đó, bạn biết cách nó hoạt động và biết phải mong đợi gì.  **Phản hồi:** UI cần phải phản hồi lại hành động của người dùng.  Nhất quán và rõ ràng: Nhất quán và rõ ràng trong thiết kế UI giúp người dùng dễ dàng tìm hiểu và sử dụng sản phẩm.  **Thiết kế tối giản:** Thiết kế tối giản giúp tập trung vào những yếu tố quan trọng nhất, giúp người dùng dễ dàng tương tác.  **Bao gồm mọi người:** Thiết kế UI tốt cần phải dễ sử dụng và hiểu được cho mọi người. | **Tương tác**: Thiết kế UX giúp người dùng có thể hoàn thành công việc của họ một cách dễ dàng bằng cách sử dụng các thành phần tương tác của hệ thống.  **Nghiên cứu người dùng**: Hiểu rõ người dùng là một phần quan trọng của thiết kế UX. Nó giúp thiết kế viên tạo ra các sản phẩm phù hợp với nhu cầu và mong đợi của người dùng.  **Kiến trúc thông tin**: Quá trình sắp xếp và tổ chức thông tin sao cho người dùng có thể dễ dàng tìm kiếm và sử dụng.  **Nguyên mẫu:** Nguyên mẫu giúp thiết kế viên trực quan hóa ý tưởng và kiểm tra chúng trước khi triển khai. |

## 2.3 Microsoft SQL Server

### 2.3.1 Giới thiệu về Microsoft SQL Server

**SQL** là viết tắt của từ **Structured Query Language**, đây là một dạng ngôn ngữ truy vấn có cấu trúc. SQL là tập hợp các câu lệnh để tương tác với cơ sở dữ liệu. Trong thực tế, SQL là một ngôn ngữ truy vấn tiêu chuẩn, được dùng trong hầu hết các hệ quản trị cơ sở dữ liệu như:SQL Server, MySQL, Oracle, MS Access,…

SQL được sử dụng để điều khiển tất cả các chức năng mà một hệ quản trị cơ sở dữ liệu cung cấp cho người dùng bao gồm:

* **Định nghĩa dữ liệu:** SQL cung cấp khả năng định nghĩa các cơ sở dữ liệu, các cấu trúc lưu trữ và tổ chức dữ liệu cũng như mối quan hệ giữa các thành phần dữ liệu.
* **Truy xuất và thao tác dữ liệu**: Với SQL, người dùng có thể dễ dàng thực hiện các thao tác truy xuất, bổ sung, cập nhật và loại bỏ dữ liệu trong các cơ sở dữ liệu.
* **Điều khiển truy cập**: SQL có thể được sử dụng để cấp phát và kiểm soát các thao tác của người sử dụng trên dữ liệu, đảm bảo sự an toàn cho cơ sở dữ liệu
* **Đảm bảo toàn vẹn dữ liệu:** SQL định nghĩa các ràng buộc toàn vẹn trong cơ sở dữ liệu nhờ đó đảm bảo tính hợp lệ và chính xác của dữ liệu trước các thao tác cập nhật cũng như các lỗi của hệ thống.

Bảng 2-3 Một số câu lệnh SQL cơ bản

|  |  |  |
| --- | --- | --- |
|  | **Câu lệnh** | **Chức năng** |
| **Thao tác dữ liệu** | SELECT | Truy xuất dữ liệu |
| INSERT | Bổ sung dữ liệu |
| UPDATE | Cập nhật dữ liệu |
| DELETE | Xoá dữ liệu |
| TRUNCATE | Xóa toàn bộ dữ liệu trong bảng |
| **Định nghĩa dữ liệu** | CREATE TABLE | Tạo bảng |
| DROP TABLE | Xóa bảng |
| ALTER TABLE | Sửa đổi bảng |

### 2.3.2 Các thành phần của SQL Server

**SQL Server** được cấu thành từ 7 thành phần khác nhau, mỗi thành phần mang một chức năng riêng. Tất cả tạo nên một hệ thống SQL Server hoàn chỉnh, đáp ứng chức năng phân tích và lưu trữ dữ liệu.

1. Database Engine: Có chức năng lưu trữ dữ liệu dưới dạng table và support, có thể tự điều chỉnh được.
2. Integration Services: Là tập hợp các đối tượng lập trình, các công cụ đồ họa, nhằm thực hiện các thao tác sao chép, di chuyển, chuyển đổi dữ liệu.
3. Analysis Services: Là dịch vụ phân tích dữ liệu bằng kỹ thuật khai thác hình khối đa chiều và dữ liệu có sẵn.
4. Notification Services: Là nền tảng phát triển và triển khai các ứng dụng soạn và gửi thông báo. Ngoài ra, dịch vụ này cũng cung cấp tính năng gửi thông báo đến hàng ngàn người đăng ký sử dụng trên các thiết bị khác nhau.
5. Reporting Services: Là một công cụ có chức năng tạo, quản lý và triển khai các báo cáo cho Server và Client. Đây cũng là nền tảng đẩy xây dựng ứng dụng báo cáo.
6. Full Text Search Services: Đây là thành phần đặc biệt trong việc truy vấn và đánh giá các chỉ mục dữ liệu văn bản được lưu trữ trong database.
7. Service Broker: Là môi trường lập trình tạo ra các ứng dụng.

## 2.4 ADO.NET

### 2.4.1 Giới thiệu về ADO.NET

ADO.NET là một công nghệ truy cập dữ liệu từ Microsoft .NET Framework, cung cấp giao tiếp giữa các hệ thống quan hệ và phi quan hệ thông qua một tập hợp các thành phần chung. ADO.NET bao gồm các lớp quản lý cho phép các ứng dụng .NET kết nối với các nguồn dữ liệu như Microsoft SQL Server, Microsoft Access, Oracle, XML, v.v., thực thi các lệnh và quản lý dữ liệu ngắt kết nối.

ADO.NET tách biệt quá trình truy cập dữ liệu khỏi quá trình thao tác dữ liệu thành các thành phần riêng biệt, có thể được sử dụng riêng biệt hoặc cùng nhau. ADO.NET bao gồm các nhà cung cấp dữ liệu .NET Framework để kết nối với cơ sở dữ liệu, thực thi các lệnh và truy xuất kết quả. Kết quả đó được xử lý trực tiếp, đặt trong một đối tượng DataSet ADO.NET để được tiếp xúc với người dùng theo cách tự do, kết hợp với dữ liệu từ nhiều nguồn, hoặc chuyển giữa các tầng.

Đối tượng DataSet cũng có thể được sử dụng độc lập với nhà cung cấp dữ liệu .NET Framework để quản lý dữ liệu cục bộ cho ứng dụng hoặc nguồn từ XML. Các lớp ADO.NET được tìm thấy trong System.Data.dll, và được tích hợp với các lớp XML được tìm thấy trong System.Xml.dll.

ADO.NET cung cấp chức năng cho các nhà phát triển viết mã quản lý tương tự như chức năng được cung cấp cho các nhà phát triển đối tượng thành phần gốc (COM) bởi ActiveX Data Objects (ADO). Chúng tôi khuyến nghị bạn sử dụng ADO.NET, không phải ADO, để truy cập dữ liệu trong các ứng dụng .NET của bạn3. ADO.NET cung cấp phương pháp truy cập dữ liệu trực tiếp nhất trong .NET Framework.

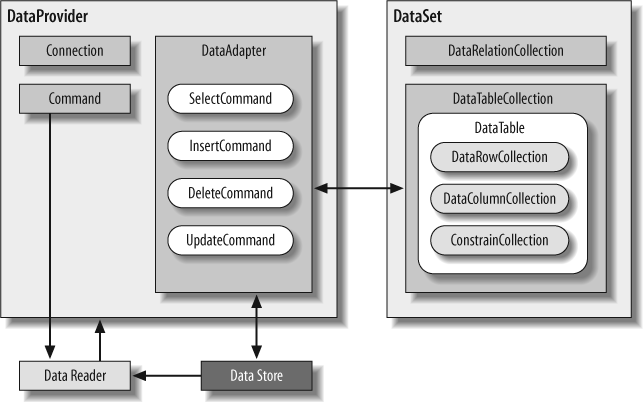
Đối với một trừu tượng cấp cao hơn cho phép các ứng dụng làm việc chống lại mô hình khái niệm thay vì mô hình lưu trữ cơ bản, hãy xem ADO.NET Entity Framework.  
 Mặc định trong .NET sẽ cung cấp sẵn thư viện để kết nối tới Sql Server–SqlClient (System.Data), còn nếu muốn sử dụng các cơ sở dữ liệu khác thì ta tải về từ Nuget



Hình 2.10 ADO.NET của .NET framework

### 2.4.2 Kiến trúc ADO.NET

ADO.NET được chia thành 2 thành phần chính là Data Provider và Data Set như hình vẽ dưới đây:



Hình 2.11Kiến trúc ADO.NET

**Data Provider:**

Gồm các thư viện cung cấp cách thức kết nối tới CSDL và thao tác tới dữ liệu. Mặc định .Net cung cấp sẵn cho chúng ta thư viện để kết nối tới Sql Server – SqlClient (System.Data). Để sử dụng các loại cơ sở dữ liệu khác chúng ta có thể tải về từ Nuget.

Một số từ khóa bạn có thể sử dụng để tìm kiếm như: mysql provider (provider cho Mysql); sqlite provider (provider cho SqlLite); …

**Data Set:**

Gồm các thư viện để quản lý dữ liệu như DataTable (đại diện cho các bảng có trong CSDL); DataColumn (đại diện cho các cột có trong bảng); DataRow (đại diện cho dữ liệu); khóa chính; Constraint (đại diện cho khóa ngoại của bảng); DataRelation (đại diện cho mỗi quan hệ giữa các bảng)…

### 2.4.3 Tạo kết nối với DbConnection

**DbConnection** là lớp biểu diễn sự kết nối tới máy chủ CSDL. Từ lớp này các thư viện triển khai cụ thể cho từng loại CSDL như SQL Server có lớp triển khai (kế thừa) DbConnection là SqlConnection.

SqlConnection sử dụng với các bước cơ bản sau:

* Tạo đối tượng SqlConnection từ một chuỗi kết nối tới SQL Server
* Mở kết nối với phương thức Open()
* Thực hiện các truy vấn bằng cách sử dụng các lớp như SqlComand, SqlDataAdapter, SqlDataReader ...
* Không còn sử dụng đến kết nối nữa thì cần đóng lại bằng phương thức Close()

Bảng 2-4 Các key trong chuỗi kết nối SQL Server

|  |  |
| --- | --- |
| **Key** | **Mô tả** |
| Connect Timeout | Số giây cố gắng kết nối trước khi phát sinh lỗi (mặc định 15s) |
| Timeout |
| Connection Timeout |
| Data Source | Key này dùng để gán địa chỉ mạng (tên máy hoặc IP hoặc domain) của SQL Server, hoặc là tên của hiện hành đang chạy của SQL Server. Nếu muốn gán cả cổng thì sẽ thêm vào phía sau địa chỉ này ,port |
| Server |
| Address |
| Addr |
| Initial Catalog | Tên của Database |
| Database |
| Password | Password để kết nối |
| PWD |
| User ID | Tài khoản (account) dùng để đăng nhập |
| UID |

Bảng 2-5 Mở và đóng kết nối đến cơ sở dữ liêu SQL Server cơ bản

|  |
| --- |
| String connectionString = "Data Source=myServerName;Initial Catalog =databaseName;User ID=myUser;Password=myPassword";  SqlConnection sqlConnection = new SqlConnection();  sqlConnection.ConnectionString = connectionString;  sqlConnection.Open();  // thực hiện cá tác vụ truy vấn CSDL (CRUD - Insert, Select, Update, Delete)  sqlConnection.Close(); |

**Giải thích chi tiết từng bước:**

1. Khởi tạo chuỗi kết nối: Chuỗi kết nối connectionString chứa thông tin cần thiết để kết nối đến cơ sở dữ liệu, bao gồm tên máy chủ (Data Source), tên cơ sở dữ liệu (Initial Catalog), tên người dùng (User ID) và mật khẩu (Password).
2. Khởi tạo đối tượng SqlConnection: SqlConnection là một lớp trong .NET Framework được sử dụng để quản lý kết nối đến cơ sở dữ liệu SQL Server.
3. Gán chuỗi kết nối: sqlConnection.ConnectionString = connectionString; gán chuỗi kết nối đã khởi tạo vào đối tượng sqlConnection.
4. Mở kết nối: sqlConnection.Open(); mở kết nối đến cơ sở dữ liệu.
5. Thực hiện các truy vấn: Tại đây, bạn có thể thực hiện các truy vấn CSDL như CRUD (Insert, Select, Update, Delete).
6. Đóng kết nối: sqlConnection.Close(); đóng kết nối sau khi đã hoàn thành các truy vấn.

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 3.1 Mô tả tổng quan hệ thống

### 3.1.1 Nhiệm vụ cơ bản của hệ thống

**Kiểm soát sản phẩm:** Hệ thống giúp người quản lý cửa hàng kiểm soát một khối lượng lớn các sản phẩm giày thể thao. Điều này bao gồm việc theo dõi số lượng tồn kho, cập nhật thông tin sản phẩm mới, và xác định khi nào cần đặt hàng thêm. Hệ thống cũng giúp người quản lý tìm kiếm thông tin sản phẩm một cách nhanh chóng và dễ dàng, giúp tiết kiệm thời gian và công sức.

**Quản lý cửa hàng tồn kho:** Theo dõi số lượng tồn kho giày theo từng mẫu, kích thước, màu sắc và loại giày. Tự động cập nhật số lượng tồn kho khi có mua bán.

**Tra cứu và thống kê:** Hệ thống cho phép cửa hàng tra cứu số lượng các mặt hàng nhập vào và bán ra. Từ đó, hệ thống có thể tự động tính toán và lên hóa đơn cho khách hàng, giúp tiết kiệm thời gian và giảm thiểu khả năng mắc lỗi. Ngoài ra, hệ thống cũng giúp thống kê báo cáo doanh thu của cửa hàng, giúp người quản lý nắm bắt được tình hình kinh doanh và đưa ra quyết định kịp thời.

**Quản lý thanh toán**: Hệ thống giúp nhận tiền thanh toán từ khách hàng và thanh toán tiền hàng cho nhà cung cấp. Hệ thống cũng giúp kiểm soát các giao dịch tài chính, giúp người quản lý nắm bắt được tình hình tài chính của cửa hàng.

**Quản lý nhân sự:** Hệ thống cũng giúp quản lý thông tin nhân viên và lương. Điều này bao gồm cập nhật và truy cập thông tin nhân viên, thông tin liên lạc

Từ đó hệ thống cửa hàng giày giúp tối ưu hóa quy trình kinh doanh, cải thiện trải nghiệm của người dùng và hỗ trợ quyết định chiến lược trên dữ liệu và thông tin cụ thể.

### 3.1.2 Mục tiêu của hệ thống

**Hệ thống được thiết kế với mục tiêu chính sau:**

1. Hiệu quả: Hệ thống cần phải tối ưu hóa quy trình làm việc, giảm thiểu thời gian và công sức cần thiết để hoàn thành các nhiệm vụ. Điều này đòi hỏi hệ thống phải được thiết kế một cách thông minh, sử dụng các thuật toán và công nghệ hiện đại để xử lý dữ liệu và thực hiện các tác vụ một cách nhanh chóng và chính xác.
2. Độ tin cậy: Hệ thống cần phải hoạt động một cách ổn định và không gặp lỗi. Điều này đòi hỏi hệ thống phải được thiết kế và kiểm thử kỹ lưỡng để đảm bảo rằng nó có thể hoạt động liên tục mà không gặp phải các sự cố hoặc lỗi nghiêm trọng.
3. Dễ sử dụng: Giao diện của hệ thống cần phải thân thiện với người dùng, dễ hiểu và dễ sử dụng. Điều này đòi hỏi hệ thống phải được thiết kế với sự chú trọng vào trải nghiệm người dùng, với các menu và lựa chọn rõ ràng, hướng dẫn sử dụng dễ hiểu, và phản hồi nhanh chóng từ hệ thống.
4. Bảo mật: Hệ thống cần phải bảo vệ thông tin của người dùng và tổ chức khỏi các mối đe dọa và vi phạm bảo mật. Điều này đòi hỏi hệ thống phải được thiết kế với các biện pháp bảo mật mạnh mẽ, bao gồm mã hóa dữ liệu, xác thực người dùng, và quản lý quyền truy cập.
5. Tính mở rộng: Hệ thống cần phải được thiết kế để có thể mở rộng và thích ứng với các yêu cầu thay đổi trong tương lai. Điều này đòi hỏi hệ thống phải được xây dựng trên một nền tảng linh hoạt, cho phép thêm các chức năng mới hoặc mở rộng các chức năng hiện tại mà không gây ra các vấn đề về hiệu suất hoặc ổn định.

## 3.2 Phân tích luồng dữ liệu và thiết kế

### 3.2.1 Sơ đồ phân rã chức năng

Mô hình phân rã chức năng (BFD – Business Function Diagram) là công cụ biểu diễn việc phân rã có thứ bậc đơn giản các công việc cần thực hiện. Mỗi công việc được chia ra làm các công việc con, số mức chia ra phụ thuộc kích cỡ và độ phức tạp của hệ thống.

Chức năng là công việc mà tổ chức cần làm và được phân theo nhiều mức từ tổng hợp đến chi tiết. Hình được biểu diễn cho chức năng là hình chữ nhật.

Quy ước đặt tên chức năng: “Động từ + Bổ ngữ”

Bảng 3-1 : Đặc điểm và mục đích của mô hình phân rã chức năng

| **Đặc điểm** | **Mục đích** |
| --- | --- |
| Cung cấp cách nhìn khái quát về chức năng  BFD có cấu trúc đơn giản và dễ hiểu, giúp người dùng dễ dàng xây dựng và sử dụng Gần gũi với sơ đồ tổ chức  Không đưa ra được mối liên quan về thông tin giữa các chức năng. | Xác định phạm vi của hệ thống cần phân tích  Cho phép mô tả khái quát dần các chức năng của tổ chức một cách trực tiếp, khách quan, phát hiện được chức năng thiếu hoặc trùng lặp  Tạo điều kiện thuận lợi khi hợp tác giữa nhà thiết kế và người sử dụng trong qua trình phát triển hệ thống. |

Hình 3.1 Sơ đồ phân rã chức năng

### 3.2.2 Sơ đồ luồng dữ liệu (Data Flow Diagram – DFD)

#### 3.2.2.1 Đặc điểm sơ đồ DFD

Sơ đồ DFD là sơ đồ mô tả sự dịch chuyển thông tin từ chức năng ( tiến trình, bộ phận ) này sang chức năng ( tiến trình chức năng bộ phận ) khác. Thể hiện sự luân chuyển thông tin giữa các bộ phận trong hệ thống

**Các thành phần trong sơ đồ DFD:**

* Tiến trình (Process) : Là một qui trình xử lý biến đổi thông tin. Bao gồm các hoạt động được thể hiện theo một trình tự nhất định nhằm nhận thông tin đầu vào, biến đổi, xử lý, để tạo ra thông tin mới phục vụ cho hoạt động của hệ thống
  + Quy tắc đặt tên: Tên tiến trình = Động từ + Bổ ngữ

|  |
| --- |
| Ký pháp : |

* Luồng dữ liệu (Data Flow): Là dòng dữ liệu biểu diễn thông tin vào hoặc ra từ một vị trí này đến một vị trí khác trong hệ thống
  + Quy tắc đặt tên: Tên là sự tổng hợp các phần tử dữ liệu lưu chuyển trong luồng. (Đôi khi tên trùng với tên của tài liệu vật lý mang thông tin)

|  |
| --- |
| Ký Pháp: |

* Kho dữ liêu (Data Store): Là nơi lưu trữ những dữ liệu được sinh ra nhưng không được xử lý ngay tại thời điểm sinh ra
  + Quy tắc đặt tên: Tên kho = Danh từ + tính từ

|  |
| --- |
| Ký pháp: |

* Tác nhân (Actor): Người, đối tượng, tổ chức có quan hệ trao đổi thông tin với hệ thống
  + Quy tắc đặt tên: Sử dụng danh từ để đặt tên cho tác nhân

|  |
| --- |
| Ký Pháp: |

#### 3.2.2.2 Xây dựng sơ đồ ngữ cảnh(Context diagram)

Sơ đồ ngữ cảnh: đây là sơ đồ mức cao nhất. Nó cho ra một cái nhìn tổng quát về hệ thống trong môi trường nó đang tồn tại. Ở mức này, sơ đồ ngữ cảnh chỉ có một tiến trình duy nhất, các tác nhân và các luồng dữ liệu (không có kho dữ liệu).

|  |
| --- |
|  |

Hình 3.2 Sơ đồ luồng dữ liệu mức ngữ cảnh

#### 3.2.2.3 Xây dựng sơ đồ mức đỉnh(Top-Level Diagram)

Sơ đồ mức đỉnh là sơ đồ phân rã từ sơ đồ ngữ cảnh. Với mục đích mô tả hệ thống chi tiết hơn, sơ đồ mức 0 được phân rã từ sơ đồ ngữ cảnh với các tiến trình được trình bày chính là các mục chức năng chính của hệ thống.

|  |
| --- |
|  |

Hình 3.3 Sơ đồ dữ liệu mức đỉnh

#### 3.2.2.4 Xây dựng mức dưới đỉnh(Lower-Level Diagram)

Sơ đồ mức dưới đỉnh là sơ đồ được phân rã từ sơ đồ mức đỉnh. Mỗi sơ đồ phân rã mức dưới định chính là sự chi tiết hóa một tiến trình mức trước. Quá trình phân rã sẽ dừng khi đạt được sơ đồ luồng dữ liệu sơ cấp (khi một tiến trình là một tính toán hay thao tác dữ liệu đơn giản, khi mỗi luồng dữ liệu không cần chia nhỏ hơn nữa.

Trong sơ đồ sau chức năng Quản lý nhập hàng bao gồm các bước sau:

Đặt hàng: nhân viên đặt hàng từ nhà cung cấp , nhà cung cấp sẽ tiếp nhận nhận đơn hàng

Nhận hàng: nhà cung cấp giao hàng , nhân viên sẽ nhận kiểm tra sản phẩm . Khi đúng loại hàng đã đặt, nhân viên sẽ nhận hàng và cập nhật sản phẩm cùng với thông tin hóa đơn vào kho

Thanh toán cho nhà cung cấp: nhà cung cấp nhận tiền đơn hàng

|  |
| --- |
|  |

Hình 3.4. Sơ đồ mức dưới đỉnh của chức năng Quản Lý Nhập Hàng

Sơ đồ mức dưới đỉnh "quản lý bán hàng này" mô tả các bước trong quá trình bán hàng, bao gồm các bước sau:

* Tiếp nhận đơn hàng: nhân viên nhận được thông tin đặt hàng
* Kiểm tra hàng hóa: Nhân viên kiểm tra hàng hóa và nhận lại thông tin phản hồi
* Xử lý đơn hàng: Nhân viên xác nhận thông tin của các sản phẩm trong đơn hàng và duyệt đơn cho khách
* Thanh toán: hóa đơn được gửi cho khách hàng và khách thanh toán hóa đơn

|  |
| --- |
|  |

Hình 3.5 Sơ đồ mức dưới đỉnh của chức năng Quản Lý Bán Hàng

Có thể thấy rằng chức năng Tra cứu bao gồm các bước sau:

1. Tra cứu hàng hoá: Nhân viên yêu cầu tra cứu thông tin hàng hoá ( số lượng, màu sắc, giá cả, loại hàng, tồn kho……)
2. Tra cứu nhân viên: Tra cứu thông tin nhân viên trong cửa hàng. Thông tin nhân viên chính thức, nhân viên thực tập, nhân viên kho, nhân viên cửa hàng, chức vụ từng nhân viên, thời gian làm việc của từng nhân viên…..
3. Tra cứu nhà cung cấp: Thông tin của nhà cung cấp hàng hoá cho cửa hàng ( tên, địa chỉ, số điện thoại…), giá cả của hàng hoá, chất lượng hàng hoá.

|  |
| --- |
|  |

Hình 3.6 Sơ đồ mức dưới đỉnh của chức năng Quản Lý Nhân Viên

Trong sơ đồ chức năng Báo Cáo Thống Kê dưới đây được thực hiện như sau:

1. Thu thập dữ liệu: dữ liệu được gửi từ nhân viên , được chia ra thành 3 kho dữ liệu là “hóa đơn nhập” , “hóa đơn xuất” và “danh sách công việc”
2. Xử lý dữ liệu: dữ liệu của 3 kho sẽ được gộp lại để xử lý
3. Phân tích dữ liệu: tổng hợp dữ liệu đưa về xử lý và sắp xếp lại để đưa ra kết quả
4. Trình bày dữ liệu: đưa ra kết quả tổng quan và chính xác nhất
5. Trình bày kết quả: in ra bản báo cáo cuối cùng của tháng và nộp lên người quản lý

|  |
| --- |
|  |

Hình 3.7 Sơ đồ mức dưới đỉnh của chức năng Báo Cáo Thống Kê

# CHƯƠNG 4: THIẾT KẾ HỆ THỐNG

## 4.1 Mô hình thực thể liên hệ( mô hình ER)

### 4.1.1 Vai trò của mô hình ER trong quá trình thiết kế cơ sở dữ liệu

* Mô hình thực thể-liên hệ, hay mô hình ER (Entity-Relationship model), là một mô hình dữ liệu sử dụng để mô tả cách dữ liệu được tổ chức và quan hệ giữa các thực thể trong cơ sở dữ liệu.
* Mô hình ER được phát triển để giúp thiết kế cơ sở dữ liệu một cách hiệu quả và đơn giản bằng cách sử dụng biểu đồ để biểu diễn thông tin về thực thể (entities), liên hệ (relationships) và thuộc tính (attributes).4.1.2 Các thành phần cơ bản của mô hình ER

Mô hình ER sử dụng ba thành phần chính:

* Thực thể (Entity): Đại diện cho các đối tượng hoặc thực thể trong hệ thống, như người, đồ vật, sự kiện hoặc khái niệm. Mỗi thực thể được biểu diễn bằng một hình hộp chữ nhật trong biểu đồ ER.
* Liên hệ (Relationship): Mô tả mối quan hệ giữa các thực thể. Liên hệ biểu diễn các tương tác, kết nối hoặc tương tác giữa các thực thể khác nhau. Liên hệ được biểu diễn bằng các đường nối giữa các thực thể.
* Thuộc tính (Attribute): Là các đặc điểm hoặc thông tin cụ thể về mỗi thực thể. Thuộc tính được liên kết với các thực thể và được biểu diễn bằng các đường nối từ thực thể đến thuộc tính.
* Thuộc tính bao gồm các loại như sau:
* Thuộc tính đơn – không thể tách nhỏ ra được
* Thuộc tính phức hợp – có thể tách ra thành các thành phần nhỏ hơn
* Các loại giá trị của thuộc tính:
* Đơn trị: các thuộc tính có giá trị duy nhất cho một thực thể.
* Đa trị: các thuộc tính có một tập giá trị cho cùng một thực thể.
* Suy diễn được

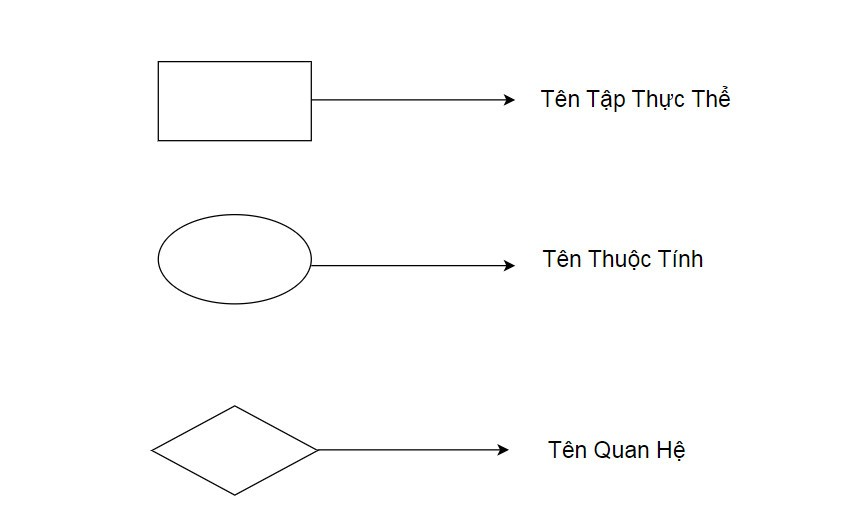
Mỗi thực thể đều được phân biệt bởi thuộc tính khóa.

**Mối quan hệ giữa các tập thực thể**

* Quan hệ là sự liên kết giữa hai hay nhiều tập thực thể
* Tập quan hệ là tập hợp các mối quan hệ giống nhau

**Lược đồ ER**

* Là đồ thị biểu diễn các tập thực thể, thuộc tính và mối quan hệ.
* Các ký hiệu trong lược đồ ER:
* Ký hiệu đỉnh:



* Ký hiệu cung: là đường nối giữa tập thực thể và thuộc tính, mối quan hệ và tập thực thể.

**Các kiểu liên kết trong lược đồ ER**

* Liên kết 1 - 1
* Liên kết 1 - n
* Liên kết n - n

**Thuộc tính khóa**

* Mỗi tập thực thể phải có một khóa
* Một khóa có thể có một hay nhiều thuộc tính
* Có thể có nhiều khóa trong một tập thực thể, ta sẽ chọn một khóa làm khóa chính cho tập thực thể đó.

### 4.1.3 Xây dựng mô hình ER

**Quy tắc:**

* Chính xác
* Tránh trùng lặp
* Dễ hiểu
* Chọn đúng mối quan hệ
* Chọn đúng kiểu thuộc tính

**Các bước xây dựng:**

* Liệt kê, chọn lọc thông tin
* Xác định tập thực thể
* Xác định mối quan hệ
* Xác định thuộc tính và gắn thuộc tính cho tập thực thể và mối quan hệ
* Quyết định thuộc tính khóa
* Vẽ biểu đồ mô hình thực thể ER
* Chuẩn hóa biểu đồ

## 4.2 Thiết kế cơ sở dữ liệu

### 4.2.1 Xây dựng mô hình liên kết thực thể ER

**Từ phát biểu bài toán ta xác định các danh từ là thực thể của hệ thống như sau:**

**Nhập hàng:**

* Nhà cung cấp gửi thông tin mặt hàng và thông tin nhà cung cấp. Thông tin mặt hàng bao gồm: tên mặt hàng , số lượng, giá, chất lượng, thông tin cơ bản, nguồn gốc xuất xứ…
* Thông tin nhà cung cấp được lưu lại: Tên nhà cung cấp , địa chỉ ,số điện thoại, ghi chú ( nếu có ).
* Nhân viên có chức vụ nhập hàng dựa vào tình hình xu thế hiện nay cũng như yêu câu đơn đặt hàng từ khách hàng để lên đơn sản phẩm cần đặt. Sau đó thống kê gửi lên quản lí yêu cầu duyệt đơn, nếu yêu cầu đực chấp thuận thì nhân viên sẽ liên lạc với nhà cung cấp để đặt hàng. Nhà cung cấp sẽ xem xét đơn đặt, nếu như bên nhà cung cấp đủ yêu cầu từ nhân viên thì cả 2 bên sẽ cùng thảo luận ngày nhập hàng và số lượng nhập hàng
* Đến ngày giao hàng như đã hẹn thì nhà cung cấp sẽ giao đúng và đủ đơn cho bên cửa hàng kèm theo tờ hóa đơn ghi đầy đủ thông tin cơ bản cần thiết để lưu lại trên hệ thống giao dịch của 2 bên.
* Sau khi nhận hàng từ nhà cung cấp, đơn hàng sẽ được vận chuyển vào trong kho
* Nếu hàng bị lỗi, nhân viên sẽ yêu cầu đổi hàng hoặc cũng có thể là hoàn trả hàng cho nhà cung cấp

**Bán hàng:**

* Khách hàng lựa chọn mặt hàng theo sở thích cá nhân hoặc xu hướng thị trường, sau đó kiểm tra chất lượng sản phẩm và giá sản phẩm
* Nếu khách hàng quyết định mua thì sẽ ra quầy thanh toán để hoàn thành giao dịch với bên cửa hàng
* Nhân viên sẽ tiếp nhận sản phẩm và gửi hóa đơn thanh toán cho khách hàng
* Nhân viên có thể nhận đơn theo yêu cầu của khách trong trường hợp bên cửa hàng có thể đáp ứng nhu cầu sau khi trao đổi với khách ( khách cọc trước 1 nửa số tiên khi lên đơn đặt theo yêu cầu )

**Tra Cứu**

* Tra cứu hàng hóa: Thông tin các mặt hàng có trong kho hoặc tại cửa hàng. Nhân viên kiểm tra số lượng hàng, các loại mã hàng, màu sản phẩm, giá bán sản phẩm… Từ đó thu thập lại thống kê xem tình trạng hàng tồn trong kho xem có cần nhập thêm hàng hay không, có đủ số lượng hàng mà khách cần hay không và kiểm soát được lợi nhận thu chi của cửa hàng.
* Tra cứu nhân viên: Thông tin nhân viên trong cửa hàng bao gồm nhân viên tại cửa hàng, nhân viên tại kho hay nhân viên thực tập. Từ đó ta kiểm soát được thời gian làm của nhân viên, chức vụ của nhân viên cũng như tính được bảng lương thưởng cho nhân viên trong cửa hàng
* Tra cứu nhà cung cấp: Thông tin nhà cung cấp được nhân viên tìm và lưu lại trên hệ thống gồm: Tên, số điện thoại, địa chỉ…. Nhân viên sẽ kiểm tra chất lượng mặt hàng bên nhà cung cấp có đảm bảo yêu cầu hay không thì mới có thể hợp tác làm việc cùng nhau. Trong trường hợp nhà cung cấp không đạt đủ yêu cầu hay trong quá trình hợp tác có xảy ra vấn đề thì bên cửa hàng có quyền dừng làm việc cùng bên phía nhà cung cấp và tìm 1 bên khác để hợp tác cùng

**Báo cáo – thông kê:**

* Cuối tháng nhân viên cửa hàng sẽ phải làm bản báo cáo thống kê chi tiết từng mục trong tháng
* Tổng kết hóa đơn mua, hóa đơn bán, lương thưởng nhân viên
* Các mặt hàng đã bán, mặt hàng nhập về và các mặt hàng còn tồn kho trong cửa hàng
* Báo cáo nhân viên ( tuyển thêm, giảm bớt….)
* Các đơn hàng theo yêu cầu của khách
* Lên kế hoạch các chiến dịch sale hay chương trình quà tặng tri ân khách hàng trong tháng tới

Từ trên đây ta xác định được các thực thể và liên kết sau:

1. Thực thể

* TAIKHOAN(MaTK, TenTK, MatKhau, PhanQuyen)
* NHANVIEN(MaNV, MaTK, TenNV, ChucVu, DiaChi, Email, SDT, Luong)
* NHACUNGCAP(MaNCC, TenNCC, DiaChi, SDT)
* SANPHAMGIAY(MaSPG, MaLG, MaNCC, TenGiay, Img, GiaMua, GiaBan, TonKho)
* SIZEGIAY(MaSize, MaSPG, KichThuoc, SoLuong)
* LOAIGIAY(MaLG, Ten)
* HOADONMUA(MaHDM, MaNV, NgayMua, TongTien)
* HOADONBAN(MaHDB, MaNV, NgayBan, TongTien)

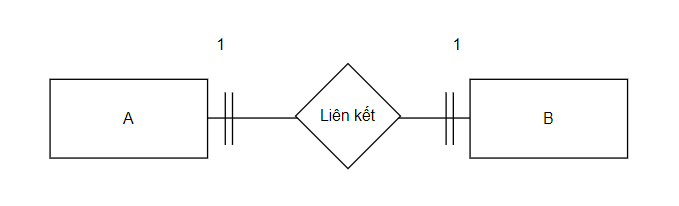
1. Xác định liên kết

* TAIKHOAN – “Thuộc” – NHANVIEN
* NHANVIEN – “Tạo” - HOADONBAN
* NHACUNGCAP – “Nhận” – HOADONMUA
* NHACUNGCAP - “Bán” – SANPHAMGIAY – “Từ” – LOAIGIAY
* SIZEGIAY – “ Thuộc” – SANPHAMGIAY
* CHITIETHOADONMUA – “Thuộc” – HOADONMUA
* CHITIETHOADONBAN – “Thuộc” – HOADONBAN

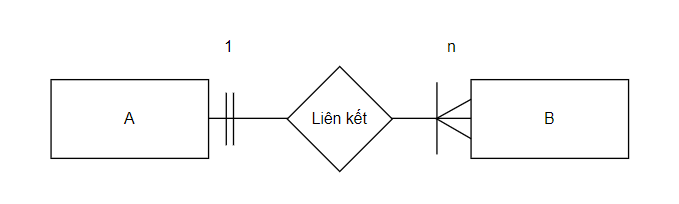
C. Các kiểu liên kết:

\*quy ước:

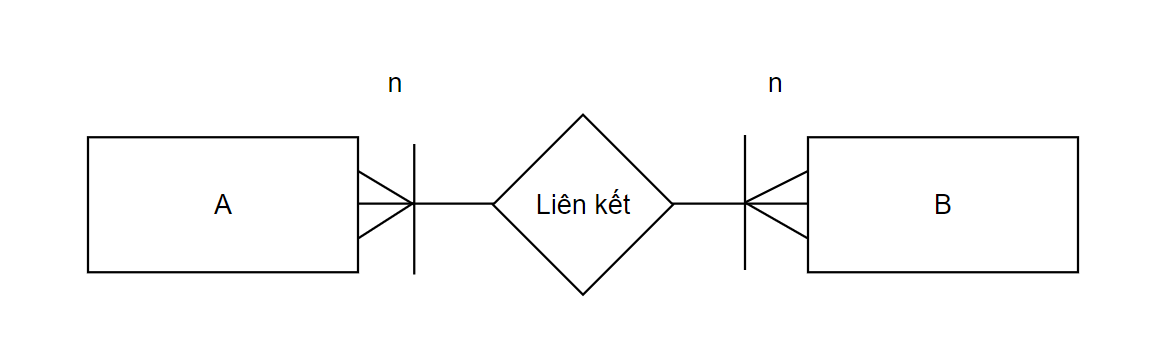
-mối quan hệ 1-1:



-mối quan hệ 1-n:



-mối quan hệ n-n:



**TAIKHOAN – “Thuộc” – NHANVIEN**

|  |
| --- |
|  |

**NHANVIEN – “Tạo” – HOADONBAN**

|  |
| --- |
|  |

**NHACUNGCAP – “Nhận” – HOADONMUA**

|  |
| --- |
|  |

**NHACUNGCAP - “Bán” – SANPHAMGIAY – “Thuộc” – LOAIGIAY**

|  |
| --- |
|  |

**SIZEGIAY – “ Thuộc” – SANPHAMGIAY**

|  |
| --- |
|  |

**CHITIETHOADONMUA – “Thuộc” – HOADONMUA**

|  |
| --- |
|  |

**CHITIETHOADONBAN – “Thuộc” – HOADONBAN**

|  |
| --- |
|  |

|  |
| --- |
|  |

Hình 4.1 : Sơ đồ ER

## 4.3 Mô hình quan hệ

**4.3.1 Mô hình quan hệ là gì ?**

* Mô hình quan hệ biểu diễn cơ sở dữ liệu dưới dạng một tập hợp các quan hệ (bảng giá trị).
* Mỗi bảng giá trị có các cột và hàng được gọi lần lượt là thuộc tính (attributes) và bộ giá trị (tuples).
* Mỗi bộ giá trị (tuple) kí hiệu một thực thể hoặc mối quan hệ trong thế giới thực. Tên của quan hệ và tên của các thuộc tính sẽ góp phần giải thích ý nghĩa của từng bộ.

*Gồm các đặc điểm chính như sau :*

Cơ sở dữ liệu là một tập hợp các quan hệ có liên quan (bảng giá trị).

* Mỗi quan hệ có một tên gọi riêng cho biết loại tuple (bộ dữ liệu) mà quan hệ có.
* Mỗi quan hệ có một tập hợp các thuộc tính (tên cột) đại diện cho các tính chất hoặc các đặc trưng của từng thực thể.
* Một bộ – tuple (hàng) biểu diễn một thực thể với các các giá trị tương ứng với từng thuộc tính.
* Mỗi cột trong bảng còn được gọi là một trường (field)

**Các ràng buộc của mô hình quan hệ**.

Ràng buộc chính là những hạn chế được chỉ định cho các giá trị dữ liệu trong cơ sở dữ liệu quan hệ. Có thể kể đến các ràng buộc chính như sau:

* Ràng buộc dựa trên mô hình vốn có: Ví dụ, một quan hệ trong cơ sở dữ liệu không được có các bộ giá trị trùng lặp, tuy nhiên, không có bất cứ ràng buộc nào trong thứ tự của các bộ giá trị và thuộc tính.
* Ràng buộc dựa trên lược đồ: Các ràng buộc được chỉ định trong khi xác định lược đồ của cơ sở dữ liệu sử dụng DDL là các ràng buộc dựa trên lược đồ. Chúng được phân loại cụ thể thành ràng buộc miền, ràng buộc khóa, ràng buộc tính toàn vẹn thực thể, ràng buộc toàn vẹn tham chiếu và ràng buộc trên giá trị rỗng
* Application-based Constraints (Ràng buộc dựa trên ứng dụng): Các ràng buộc không thể áp dụng trong khi xác định lược đồ cơ sở dữ liệu sẽ được thể hiện trong các chương trình ứng dụng.

## 4.4 Thiết kế cơ sở dữ liệu

### 4.4.1 Thiết kế các bảng trong cơ sở dữ liệu

Bảng 4-1. Bảng TaiKhoan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Giá trị** | **Ràng buộc** | **Ý nghĩa** |
| 1 | MaTK | NVARCHAR | PRIMARY KEY | Mã tài khoản |
| 2 | TenTK | NVARCHAR | NOT NULL | Tên tài khoản |
| 3 | MatKhau | NVARCHAR | NOT NULL | Mật khẩu |
| 4 | PhanQuyen | NVARCHAR | NOT NULL | Phân quyền |

Bảng 4-2. Bảng NhanVien

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Giá trị** | **Ràng buộc** | **Ý nghĩa** |
| 1 | MaNV | NVARCHAR | PRIMARY KEY | Mã nhân viên |
| 2 | MaTK | NVARCHAR | FOREIGN KEY | Mã tài khoản |
| 3 | TenNV | NVARCHAR | NOT NULL | Tên nhân viên |
| 4 | DiaChi | NVARCHAR | NOT NULL | Địa chỉ |
| 5 | SDT | NVARCHAR | UNIQUE | Số điện thoại |
| 6 | ChucVu | NVARCHAR | NOT NULL | Chức vụ |
| 7 | Luong | MVARCHAR |  | Lương |

Bảng 4-3. Bảng NhaCungCap

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Giá trị** | **Ràng buộc** | **Ý nghĩa** |
| 1 | MaNCC | NVARCHAR | PRIMARY KEY | Mã nhà cung cấp |
| 2 | TenNCC | NVARCHAR | NOT NULL | Tên nhà cung cấp |
| 3 | DiaChi | NVARCHAR | NOT NULL | Địa chỉ |
| 4 | SDT | NVARCHAR | UNIQUE | Số điện thoại |

Bảng 4-4. Bảng LoaiGiay

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Giá trị** | **Ràng buộc** | **Ý nghĩa** |
| 1 | MaLG | NVARCHAR | PRIMARY KEY | Mã loại giày |
| 2 | KichThuoc | NVARCHAR | NOT NULL | Kích thước loại giày |
| 3 | SoLuong | INT | NOT NULL | Số lượng |

Bảng 4-5. Bảng SanPhamGiay

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Giá trị** | **Ràng buộc** | **Ý nghĩa** |
| 1 | MaSPG | NVARCHAR | PRIMARY KEY | Mã sản phẩm giày |
| 2 | MaLG | NVARCHAR | FOREIGN KEY | Mã loại giày |
| 3 | MaNCC | NVARCHAR | FOREIGN KEY | Mã nhà cung cấp |
| 4 | TenGiay | NVARCHAR | NOT NULL | Tên giày |
| 5 | Img | NVARCHAR | NOT NULL | Đường dẫn ảnh |
| 6 | GiaMua | NVARCHAR |  | Giá mua |
| 7 | GiaBan | NVARCHAR |  | Giá Bán |
| 8 | TonKho | INT |  | Tồn kho |

Bảng 4-6. Bảng SizeGiay

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Giá trị** | **Ràng buộc** | **Ý nghĩa** |
| 1 | MaSize | NVARCHAR | PRIMARY KEY | Mã size giày |
| 2 | MaSPG | NVARCHAR | FOREIGN KEY | Mã sản phẩm giày |
| 3 | KichThuoc | INT |  | Kích thước |
| 4 | SoLuong | INT |  | Số lượng |

Bảng 4-7 Bảng HoaDonMua

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Giá trị** | **Ràng buộc** | **Ý nghĩa** |
| 1 | MaHDM | NVARCHAR | PRIMARY KEY | Mã hóa đơn mua |
| 2 | MaNV | NVARCHAR | FOREIGN KEY | Mã nhân viên |
| 3 | NgayMua | DATETIME | DEFAULT  GETDATE() | Ngày mua |
| 4 | TongTien | NVARCHAR |  | Tổng tiền |

Bảng 4-8. Bảng ChiTietHoaDonMua

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Giá trị** | **Ràng buộc** | **Ý nghĩa** |
| 1 | MaHDM | NVARCHAR | FOREIGN KEY,  PRIMARY KEY | Mã hóa đơn mua |
| 2 | MaSPG | NVARCHAR | FOREIGN KEY,  PRIMARY KEY | Mã sản phẩm giày |
| 3 | SoLuong | INT |  | Số lượng |

Bảng 4-9. Bảng HoaDonBan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Giá trị** | **Ràng buộc** | **Ý nghĩa** |
| 1 | MaHDB | NVARCHAR | PRIMARY KEY | Mã hóa đơn bán |
| 2 | MaNV | NVARCHAR | FOREIGN KEY | Mã Nhân viên |
| 3 | NgayBan | DATETIME | DEFAULE  GETDATE() | Ngày Bán hàng |
| 4 | TongTien | NVARCHAR |  | Tổng tiền |

Bảng 4-10. Bảng ChiTietHoaDonBan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Giá trị** | **Ràng buộc** | **Ý nghĩa** |
| 1 | MaHDB | NVARCHAR | FOREIGN KEY,  PRIMARY KEY | Mã hóa đơn bán |
| 2 | MaSPG | NVARCHAR | FOREIGN KEY,  PRIMARY KEY | Mã sản phẩm giày |
| 3 | SoLuong | INT |  | Số lượng |

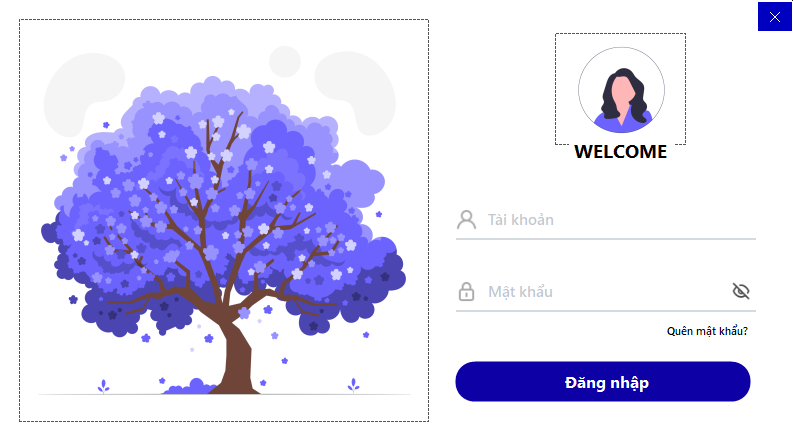
### 4.4.2 Mô hình Diagram



Hình 4.2 Mối quan hệ giữa các bảng trong cơ sở dữ liệu

## 4.5 Thiết kế giao diện

### 4.5.1 Giao diện màn hình đăng nhập



Hình 4.3: Giao diện đăng nhập

Màn hình đăng nhập được thiết kế đơn giản và thân thiện với người dùng. Khi bạn mở ứng dụng, màn hình đăng nhập sẽ xuất hiện đầu tiên.

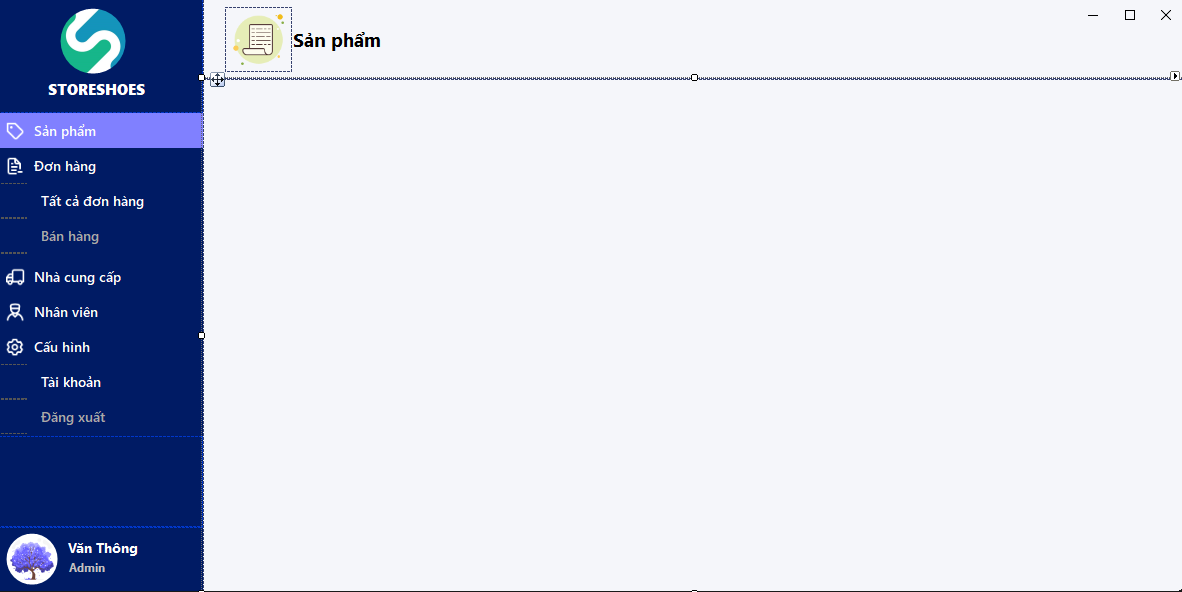
1. **Nhập tài khoản**: Trên cùng của màn hình là một ô dành cho bạn nhập tài khoản của mình
2. **Nhập mật khẩu**: Ngay dưới ô nhập tài khoản là ô để bạn nhập mật khẩu. Khi bạn nhập mật khẩu, các ký tự sẽ được thay thế bằng dấu sao (\*) để bảo mật thông tin của bạn.
3. **Hiển thị/Ẩn mật khẩu**: Bên cạnh ô mật khẩu có một biểu tượng mắt. Khi bạn nhấn vào biểu tượng này, mật khẩu bạn đã nhập sẽ được hiển thị, giúp bạn kiểm tra lại mật khẩu. Khi bạn nhấn lần nữa, mật khẩu sẽ trở lại dạng ẩn.

Sau khi nhập đầy đủ thông tin, bạn chỉ cần nhấn vào nút ‘Đăng nhập’ để tiếp tục. Hãy nhớ rằng, bạn cần phải nhập đúng tài khoản và mật khẩu đã đăng ký để có thể đăng nhập thành công

Bảng 4-11: Điều kiện xử lí giao diện đăng nhập

|  |  |  |
| --- | --- | --- |
| **STT** | **Điều kiện kích hoạt** | **Xử lí** |
| 1 | Sau khi ấn nút “ Đăng nhập “ | Form hiện tại sẽ bị ẩn đi và Form main sẽ suất hiện |
| 2 | Sau khi nhấn vào icon | mật khẩu bạn đã nhập sẽ được hiển thị, giúp bạn kiểm tra lại mật khẩu  và icon  sẽ được hiện thị thay thế icon vừa nhấn |
| 3 | Sau khi nhấn vào icon | mật khẩu sẽ được hiện thị dưới dạng “\*” và icon sẽ được hiện thị thay thế icon vừa nhấn |

### 4.5.2 Giao diện chính của chương trình

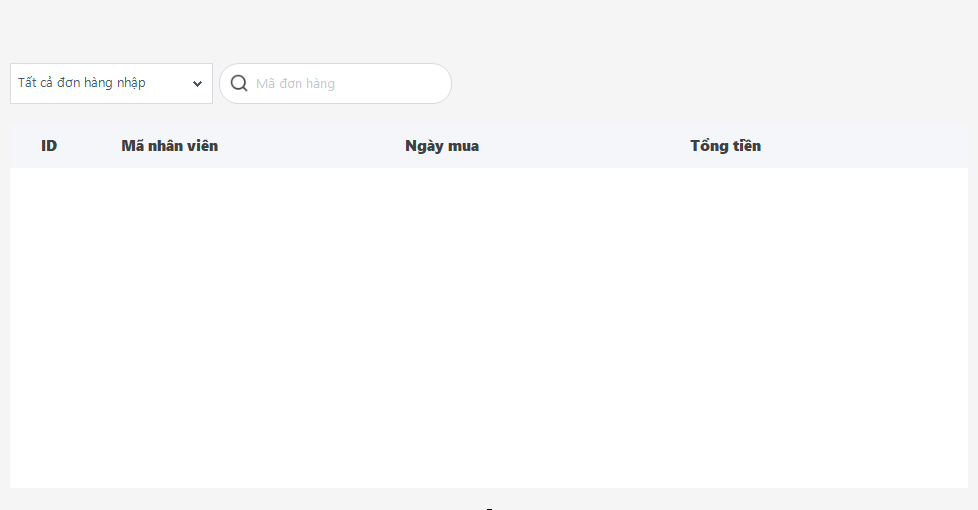


Hình 4.4 Giao diện chương trình

Ứng dụng “STORESHOES” có thiết kế giao diện người dùng đơn giản và tối giản. Logo và tên ứng dụng được đặt ở góc trên bên trái của cửa sổ. Bên dưới logo là thanh điều hướng màu tím đậm với các biểu tượng và nhãn bằng tiếng Việt, giúp người dùng sử dụng dụng tài khoản với quyền “Admin” dễ dàng truy cập vào các phần khác nhau của ứng dụng như “thống kê”, “đơn hàng”, “sản phẩm”, “nhân viên”, “nhà cung cấp”, “Nhân viên”, “Cấu hình”.Ở dưới cùng sẽ hiển thị tên với quyền truy cập của người dùng dựa trên tài khoản đã đăng kí.

Giao diện được thiết kế Bảng màu chủ yếu của ứng dụng bao gồm màu tím đậm, trắng và xám nhạt, tạo nên vẻ ngoài chuyên nghiệp. Tổng thể, giao diện người dùng của ứng dụng rất

### 4.5.3 Giao diện màn hình tất cả đơn hàng



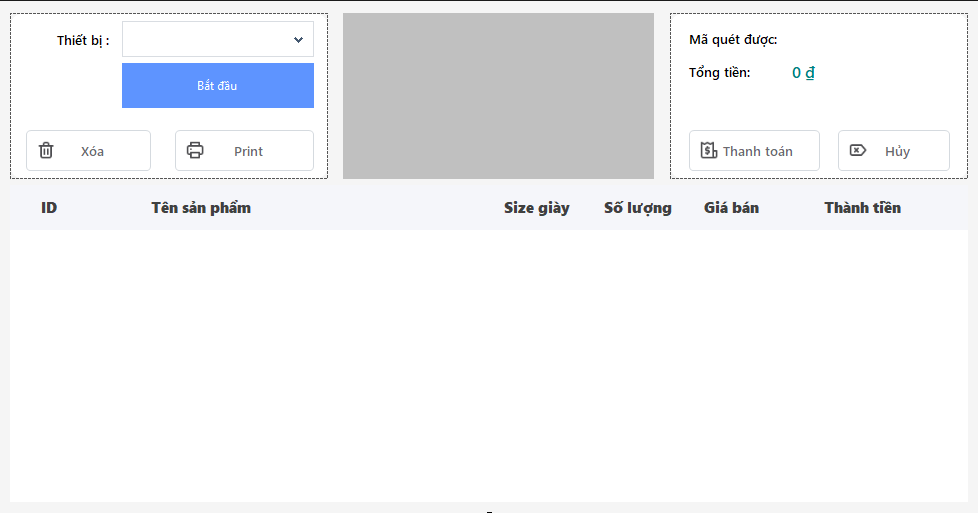
Hình 4.5 Giao diện tất cả đơn hàng

Đây là giao diện của một hệ thống quản lý đơn hàng, hiển thị bằng tiếng Việt. Có các chức năng tìm kiếm và lọc đơn hàng, cũng như thông tin chi tiết về từng đơn hàng bao gồm ID, mã nhân viên, ngày mua và tổng tiền.

Bảng 4-12: Điều kiện xử lí giao diện tất cả dơn hàng

|  |  |  |
| --- | --- | --- |
| **STT** | **Điều kiện kích hoạt** | **Xử lí** |
|  | Sau khi nhập dữ liệu vào thanh “Search” | DataGridView sẽ hiển thị dữ liệu dựa trên những gì người dùng đã tìm kiếm |
|  | Sau khi chọn nội dung trên comboBox | DataGridView sẽ hiển thị dữ liệu dựa trên những gì người dùng đã chọn trong comboBox |

### 4.5.4 Giao diện màn hình bán hàng



Hình 4.6 Giao diện màn hình bán hàng

Bên trái màn hình, có một khu vực để chọn “Thiết bị” với một menu dropdown, cho phép người dùng chọn thiết bị Camera kết nối với ứng dụng mong muốn. Phía dưới khu vực này là các biểu tượng cho phép chỉnh sửa, xóa và in, giúp người dùng dễ dàng thao tác với ứng dụng.

Bên phải màn hình, thông tin về “Mã quét được” và “Tổng tiền” được hiển thị rõ ràng, giúp người dùng dễ dàng theo dõi. Phía dưới đây là các nút “Thanh toán” và “Hủy”, cho phép người dùng hoàn tất hoặc hủy bỏ đơn hàng một cách nhanh chóng.

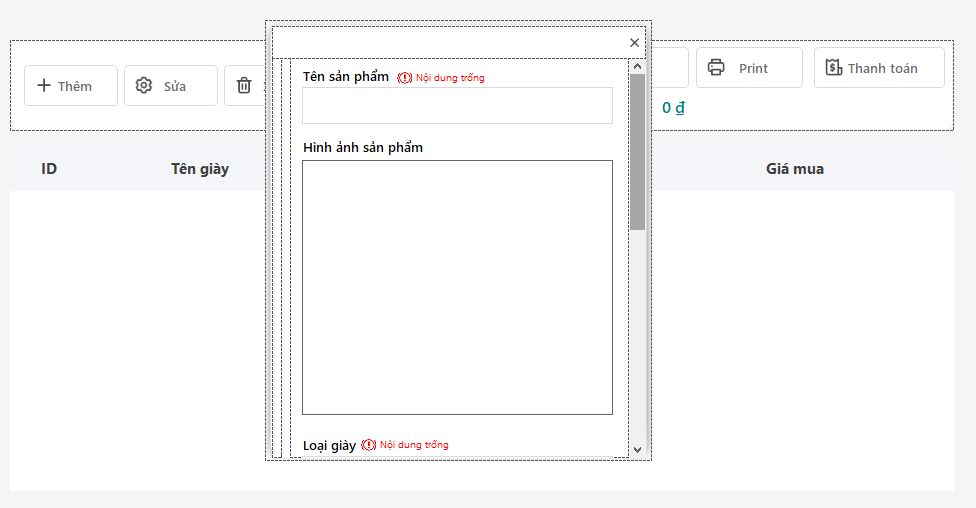
Cuối cùng, phía dưới cả hai khu vực này là một bảng trống với các cột “Tên sản phẩm”, “Giá bán”, “Số lượng”, “Giảm giá”, và “Thành tiền”. Bảng này có thể được sử dụng để nhập thông tin sản phẩm, giúp người dùng dễ dàng quản lý đơn hàng của mình.

Tổng thể, giao diện người dùng của ứng dụng được thiết kế một cách logic và dễ sử dụng, giúp người dùng quản lý đơn hàng/bán hàng một cách hiệu quả.

Bảng 4-13: Điều kiện xử lí giao diện bán hàng

|  |  |  |
| --- | --- | --- |
| **STT** | **Điều kiện kích hoạt** | **Xử lí** |
| 1 | Sau khi nhấn nút “Bắt đầu” | Ứng dụng sẽ kết nối với thiết bị camera mà người dùng đã chọn, và tiến hành quét mã mà người dùng đưa vào |
| 2 | Sau khi nhấn nút “Thanh toán” | Cập nhật lại tồn kho trên cơ sở dữ liệu. Và xóa toàn bộ nội dung trong DataGridView, reset lại mã quét được và tổng tiền |
| 3 | Sau khi nhấn nút “Hủy” | Xóa toàn bộ nội dung trong DataGridView, reset lại mã quét được và tổng tiền |
| 4 | Sau khi chọn 1 dòng trong DataGridView rồi nhấn nút “Xóa” | Xóa dữ liệu hàng đã Select, và các dữ liệu trong DataTable có liên quan |
| 5 | Sau khi nhấn nút “Print” | Sẽ in ra hóa đơn dựa trên các dữ liệu trong DataGridView |

### 4.5.5 Giao diện màn hình nhập hàng



Hình 4.7 Giao diện màn hình nhập hàng

Giao diện người dùng của ứng dụng “Đơn hàng / Nhập hàng” được thiết kế một cách rõ ràng và dễ sử dụng. Ở phía trên cùng, tiêu đề “Đơn hàng / Nhập hàng” cho thấy chức năng chính của ứng dụng.

Có ba nút chức năng “Thêm”, “Sửa”, và “Xóa” được sắp xếp theo thứ tự từ trái sang phải, giúp người dùng thêm, chỉnh sửa và xóa các mục. Bên phải cùng của giao diện, có hai nút “Print”, “Hủy” và “Thanh toán” cho phép người dùng in hóa đơn và thanh toán.

Phía dưới các nút chức năng, có một bảng trống với các tiêu đề cột là “ID”, “Tên giày”, “Số lượng”, và “Giá Mua”. Bảng này có thể được sử dụng để nhập thông tin sản phẩm, giúp người dùng dễ dàng quản lý đơn hàng của mình.

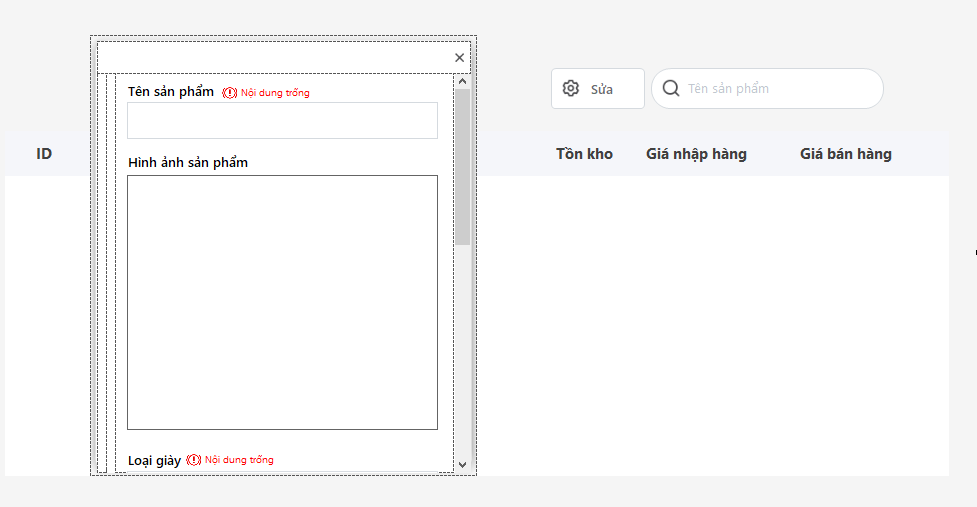
Cuối cùng, phía dưới bảng, thông tin tổng tiền được hiển thị rõ ràng, giúp người dùng dễ dàng theo dõi tổng số tiền cần thanh toán.

Tổng thể, giao diện người dùng của ứng dụng được thiết kế một cách logic và dễ sử dụng, giúp người dùng quản lý đơn hàng/nhập hàng một cách hiệu quả. Màu sắc trắng và đen tạo nên sự tương phản, giúp nội dung dễ đọc và thân thiện với mắt.

Bảng 4-14: Điều kiện xử lí giao diện màn hình nhập hàng

|  |  |  |
| --- | --- | --- |
| **STT** | **Điều kiện kích hoạt** | **Xử lí** |
| 1 | Sau khi ấn nút “ Thêm“ | Sẽ hiện lên một khung cho người dùng có thể điền thông tin của nhà cung cấp mới và vô hiệu hóa chắc năng của tất cả các control bên ngoài khung |
| 2 | Sau khi nhấn vào nút “xác nhận” | * Khi người dùng điền thiếu trường thông tin sẽ có một cảnh báo hiện ra yêu cầu người dùng điền đầy đủ thông tin   Khi người dung điền đầy đủ tất cả trường thông tin rồi thì khung hiện tại sẽ đóng lại và thêm hoặc cập nhật một nhà cung cấp mới tùy theo chức năng mà người dùng chọn |
| 3 | Sau khi nhấn vào nút “Hủy” | Thì khung hiện tại sẽ đóng lại và không thêm bất cứ thông tin gì vào danh sách |
| 4 | Sau khi chọn nhà cung cấp rồi nhấn nút “Sửa” | Sẽ hiện lên một khung và cho phép người dùng cập nhật lại thông tin của nhà cung cấp và vô hiệu hóa chắc năng của tất cả các control bên ngoài khung |
| 5 | Sau khi chọn nhà cung cấp rồi nhấn nút “Xóa” | Sẽ xóa một nhà cung cấp đã chọn trong danh sách |
| 6 | Sau khi click liên tục 2 lần vào khung của “Hình ảnh sản phẩm” | Sẽ mở File Explorer cho người dùng chọn ảnh  Nếu như người dùng chọn 1 file không phải ảnh hoặc không chọn file sẽ sẽ có thông báo hiện lên màn hình |
| 7 | Sau khi nhấn nút “Print” | Sẽ in ra hóa đơn dựa trên các dữ liệu trong DataGridView |

### 4.5.6 Giao diện màn hình sản phẩm



Hình 4.8 Giao diện màn hình sản phẩm

Giao diện người dùng của ứng dụng “Sản phẩm” được thiết kế một cách rõ ràng và dễ sử dụng. Ở phía trên cùng, tiêu đề “Sản phẩm” cho thấy chức năng chính của ứng dụng.

Có nút chức năng “sửa”,được sắp xếp theo thứ tự từ trái sang phải, giúp người dùng thêm, chỉnh sửa và xóa các mục. Bên phải cùng của giao diện, có một khung tìm kiếm với biểu tượng kính lúp và placeholder text “Tìm sản phẩm”, cho phép người dùng tìm kiếm sản phẩm một cách nhanh chóng.

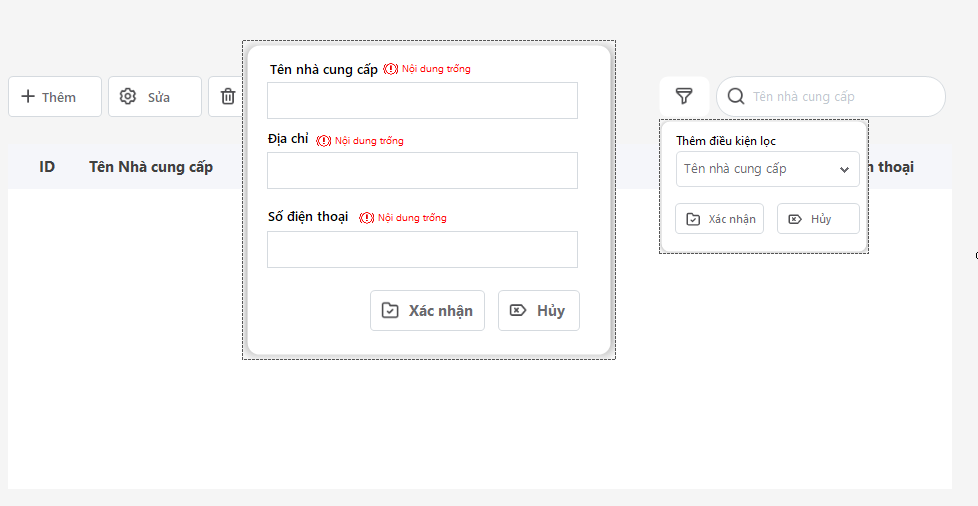
Phía dưới các nút chức năng và khung tìm kiếm, có một bảng trống với các tiêu đề cột là “ID”, “Tên Giày”, “Tồn kho”, “Giá mua”, và “Gia bán”. Bảng này có thể được sử dụng để nhập thông tin sản phẩm, giúp người dùng dễ dàng quản lý sản phẩm của mình.

Tổng thể, giao diện người dùng của ứng dụng được thiết kế một cách logic và dễ sử dụng, giúp người dùng quản lý sản phẩm một cách hiệu quả. Màu sắc trắng và đen tạo nên sự tương phản, giúp nội dung dễ đọc và thân thiện với mắt.

Bảng 4-15: Điều kiện xử lí giao diện màn hình sản phẩm

|  |  |  |
| --- | --- | --- |
| **STT** | **Điều kiện kích hoạt** | **Xử lí** |
| 1 | Sau khi nhấn vào nút “xác nhận” | Khi người dùng điền thiếu trường thông tin sẽ có một cảnh báo hiện ra yêu cầu người dùng điền đầy đủ thông tin  Khi người dung điền đầy đủ tất cả trường thông tin rồi thì khung hiện tại sẽ đóng lại và thêm hoặc cập nhật một nhà cung cấp mới tùy theo chức năng mà người dùng chọn |
| 2 | Sau khi nhấn vào nút “Hủy” | Thì khung hiện tại sẽ đóng lại và không thêm bất cứ thông tin gì vào danh sách |
| 3 | Sau khi chọn nhà cung cấp rồi nhấn nút “Sửa” | Sẽ hiện lên một khung và cho phép người dùng cập nhật lại thông tin của nhà cung cấp và vô hiệu hóa chắc năng của tất cả các control bên ngoài khung |
| 4 | Sau khi chọn nhà cung cấp rồi nhấn nút “Xóa” | Sẽ xóa một nhà cung cấp đã chọn trong danh sách |
| 5 | Sau khi click liên tục 2 lần vào khung của “Hình ảnh sản phẩm” | Sẽ mở File Explorer cho người dùng chọn ảnh  Nếu như người dùng chọn 1 file không phải ảnh hoặc không chọn file sẽ sẽ có thông báo hiện lên màn hình |
| 6 | Sau khi nhấn vào nút | Sẽ hiện lên một khung cho phép người dùng chọn điều kiện lọc để ô tìm kiếm có thể giúp người dùng tìm kiếm theo điều kiện đã thêm vào |
| 7 | Nhập thông tin vào ô “search” | Danh sách sẽ hiện lên những nhà cung cấp có liên quan đến điều kiện lọc và thông tin người dùng nhập vào |

### 4.5.7 Giao diện màn hình quản lý nhà cung cấp



Hình 4.9 Giao diện quản lý nhà cung cấp

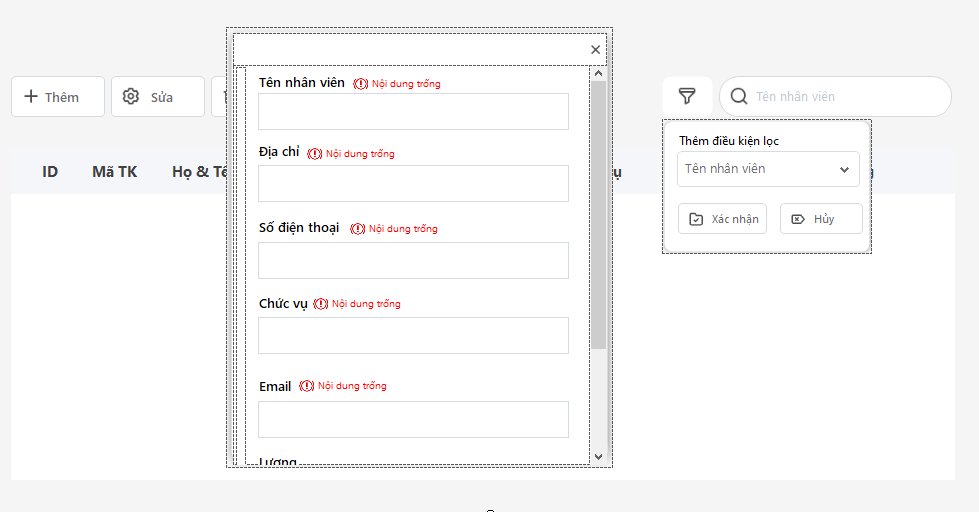
Màn hình quản lý nhà cung cấp được thiết kế để giúp bạn dễ dàng quản lý thông tin của các nhà cung cấp. Khi bạn truy cập vào màn hình này, bạn sẽ thấy một danh sách các nhà cung cấp hiện có. Mỗi dòng trong danh sách đại diện cho một nhà cung cấp, bao gồm tên, địa chỉ, số điện thoại và các thông tin khác.

Bảng 4-16: Điều kiện xử lí giao diện quản lý nhà cung cấp

|  |  |  |
| --- | --- | --- |
| **STT** | **Điều kiện kích hoạt** | **Xử lí** |
| 1 | Sau khi ấn nút “ Thêm“ | Sẽ hiện lên một khung cho người dùng có thể điền thông tin của nhà cung cấp mới và vô hiệu hóa chắc năng của tất cả các control bên ngoài khung |
| 2 | Sau khi nhấn vào nút “xác nhận” | * Khi người dùng điền thiếu trường thông tin sẽ có một cảnh báo hiện ra yêu cầu người dùng điền đầy đủ thông tin * Khi người dung điền đầy đủ tất cả trường thông tin rồi thì khung hiện tại sẽ đóng lại và thêm hoặc cập nhật một nhà cung cấp mới tùy theo chức năng mà người dùng chọn |
| 3 | Sau khi nhấn vào nút “Hủy” | Thì khung hiện tại sẽ đóng lại và không thêm bất cứ thông tin gì vào danh sách |
| 4 | Sau khi chọn nhà cung cấp rồi nhấn nút “Sửa” | Sẽ hiện lên một khung và cho phép người dùng cập nhật lại thông tin của nhà cung cấp và vô hiệu hóa chắc năng của tất cả các control bên ngoài khung |
| 5 | Sau khi chọn nhà cung cấp rồi nhấn nút “Xóa” | Sẽ xóa một nhà cung cấp đã chọn trong danh sách |
| 6 | Sau khi nhấn vào nút | Sẽ hiện lên một khung cho phép người dùng chọn điều kiện lọc để ô tìm kiếm có thể giúp người dùng tìm kiếm theo điều kiện đã thêm vào |
| 7 | Nhập thông tin vào ô “search” | Danh sách sẽ hiện lên những nhà cung cấp có liên quan đến điều kiện lọc và thông tin người dùng nhập vào |

### 

### 4.5.8 Giao diện màn hình quản lý nhân viên



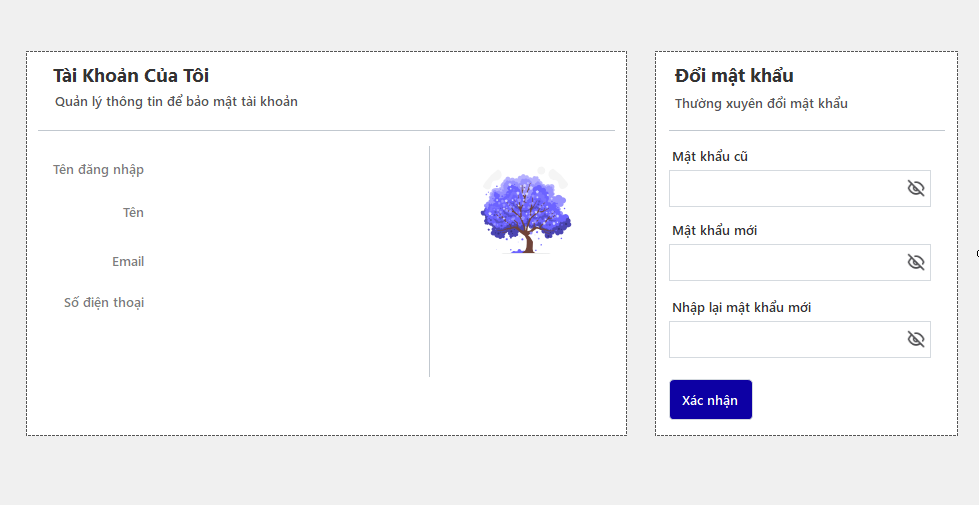
Hình 4.10 Giao diện quản lý nhân viên

Màn hình quản lý Nhân viên được thiết kế để giúp bạn quản lý thông tin của các nhân viên một cách dễ dàng. Khi bạn truy cập vào màn hình này, bạn sẽ thấy một danh sách các nhân viên hiện có với thông tin chi tiết như tên, chức vụ, số điện thoại và địa chỉ email.

Bảng 4-17: Điều kiện xử lí giao diện quản lý Nhân viên

|  |  |  |
| --- | --- | --- |
| **STT** | **Điều kiện kích hoạt** | **Xử lí** |
| 1 | Sau khi ấn nút “ Thêm“ | Sẽ hiện lên một khung cho người dùng có thể điền thông tin của nhà cung cấp mới và vô hiệu hóa chắc năng của tất cả các control bên ngoài khung |
| 2 | Sau khi nhấn vào nút “xác nhận” | * Khi người dùng điền thiếu trường thông tin sẽ có một cảnh báo hiện ra yêu cầu người dùng điền đầy đủ thông tin * Khi người dung điền đầy đủ tất cả trường thông tin rồi thì khung hiện tại sẽ đóng lại và thêm hoặc cập nhật một nhà cung cấp mới tùy theo chức năng mà người dùng chọn |
| 3 | Sau khi nhấn vào nút “Hủy” | Thì khung hiện tại sẽ đóng lại và không thêm bất cứ thông tin gì vào danh sách |
| 4 | Sau khi chọn nhà cung cấp rồi nhấn nút “Sửa” | Sẽ hiện lên một khung và cho phép người dùng cập nhật lại thông tin của nhà cung cấp và vô hiệu hóa chắc năng của tất cả các control bên ngoài khung |
| 5 | Sau khi chọn nhà cung cấp rồi nhấn nút “Xóa” | Sẽ xóa một nhà cung cấp đã chọn trong danh sách |
| 6 | Sau khi nhấn vào nút | Sẽ hiện lên một khung cho phép người dùng chọn điều kiện lọc để ô tìm kiếm có thể giúp người dùng tìm kiếm theo điều kiện đã thêm vào |
| 7 | Nhập thông tin vào ô “search” | Danh sách sẽ hiện lên những nhân viên có liên quan đến điều kiện lọc và thông tin người dùng nhập vào |

### 4.5.9 Giao diện màn hình tài khoản và mật khẩu



Hình 4.11 Giao diện quản lý tài khoản

Bên trái màn hình, có một khu vực “Tài Khoản Của Tôi” hiển thị thông tin cá nhân của người dùng như tên đăng nhập, tên, email và số điện thoại. Người dùng có thể chỉnh sửa thông tin này và lưu lại bằng nút “Lưu”. Phía dưới thông tin cá nhân, người dùng có thể thay đổi ảnh đại diện của mình bằng nút “Chọn ảnh”.

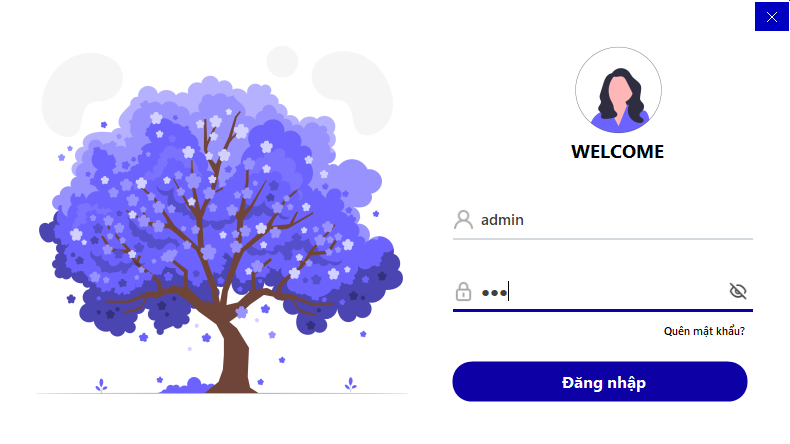
Bên phải màn hình, có một khu vực “Đổi Mật Khẩu” cho phép người dùng thay đổi mật khẩu của họ. Người dùng cần nhập mật khẩu hiện tại của họ, sau đó nhập và xác nhận mật khẩu mới. Sau khi hoàn tất, họ có thể nhấn nút “Xác nhận” để thay đổi mật khẩu.

Bảng 4-18: Điều kiện xử lí giao diện đăng nhập

|  |  |  |
| --- | --- | --- |
| **STT** | **Điều kiện kích hoạt** | **Xử lí** |
| 1 | Sau khi ấn nút “ Xác nhận“ | Kiểm tra các điều kiện đã rằng buộc ở trong code. Rồi xử lý đổi mật khẩu |
| 2 | Sau khi nhấn vào icon | mật khẩu bạn đã nhập sẽ được hiển thị, giúp bạn kiểm tra lại mật khẩu.và icon sẽ được hiện thị thay thế icon vừa nhấn |
| 3 | Sau khi nhấn vào icon | mật khẩu sẽ được hiện thị dưới dạng “\*” và icon sẽ được hiện thị thay thế icon vừa nhấn |

# CHƯƠNG 5: KẾT QUẢ THỰC NGHIỆM

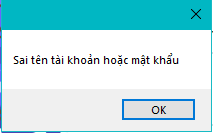
## 5.1 Giao diện màn hình đăng nhập



Hình 5.1 Giao diện màn hình đăng nhập

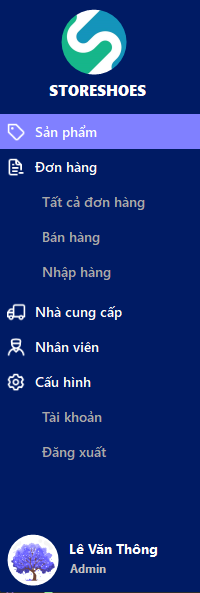
Khi mở phần mềm lần đầu tiên, màn hình đăng nhập sẽ xuất hiện. Màn hình này cho phép người dùng đăng nhập bằng tài khoản đã có. Người dùng cần nhập tên người dùng và mật khẩu vào các trường được cung cấp. Nếu người dùng quên mật khẩu, họ có thể nhấp vào liên kết “Quên mật khẩu?” để khôi phục lại. Sau khi điền thông tin, người dùng nhấn vào nút “Đăng nhập” để truy cập hệ thống. Lưu ý rằng không có tùy chọn đăng nhập bằng Facebook hoặc đăng ký tài khoản mới trên giao diện này. Hệ thống sẽ kiểm tra thông tin đăng nhập và nếu thành công, người dùng sẽ được đưa vào hệ thống.

Khi người dùng nhập sai tài khoản hoặc mật khẩu sẽ có thông báo lỗi hiện lên:



## 5.2 Giao diện chính của chương trình

### 5.2.1 Giao diện thanh điều hướng



Hình 5.2 Giao diện thanh điều hướng

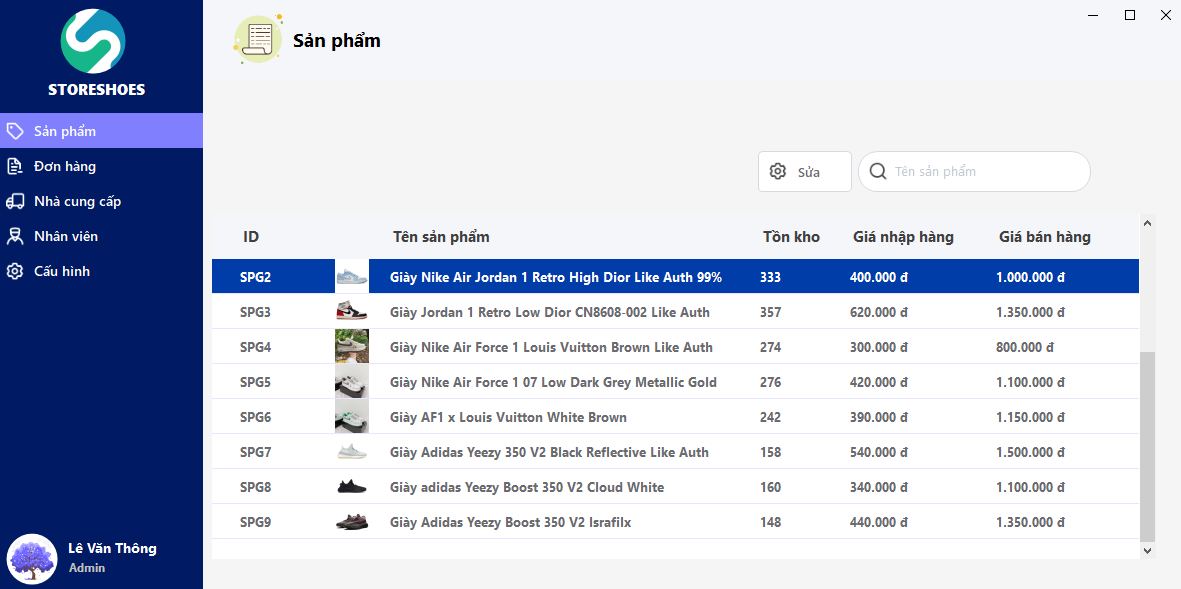
Thanh điều hướng của ứng dụng “STORESHOES” hoạt động như sau:

1. **Logo**: Ở đầu thanh điều hướng, có logo của ứng dụng, thiết kế theo hình vòng tròn và tên “STORESHOES” được viết bên dưới.
2. **Các mục điều hướng**: Dưới logo, có năm mục điều hướng được liệt kê theo thứ tự dọc, mỗi mục đi kèm với một biểu tượng:
   * “Sản phẩm” với biểu tượng túi mua sắm,
   * “Đơn hàng” với biểu tượng clipboard và có các tùy chọn phụ như “Tất cả đơn hàng”, “Bán hàng”, và “Nhập hàng”,
   * “Nhà cung cấp” với biểu tượng nhà máy,
   * “Nhân viên” với biểu tượng người,
   * “Cấu hình” với biểu tượng bánh răng và các tùy chọn phụ như “Tài khoản” và “Đăng xuất”.

Người dùng có thể nhấp vào các mục này để điều hướng đến các phần tương ứng của ứng dụng.

1. **Phần hồ sơ người dùng**: Ở cuối thanh điều hướng, có một phần hồ sơ người dùng hiển thị tên “Lê Văn Thông”, và quyền truy cập của tài khoản vào phần mềm là “Admin”.

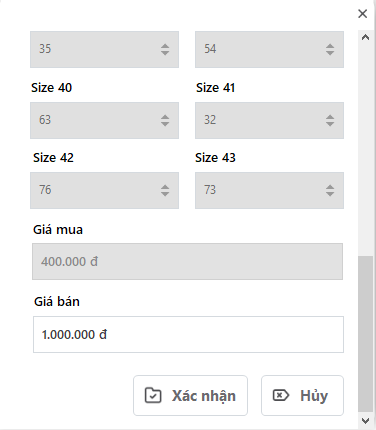
### 5.2.2 Giao diện khi chọn điều hướng sản phẩm



Hình 5.3 Giao diện điều hướng sản phẩm

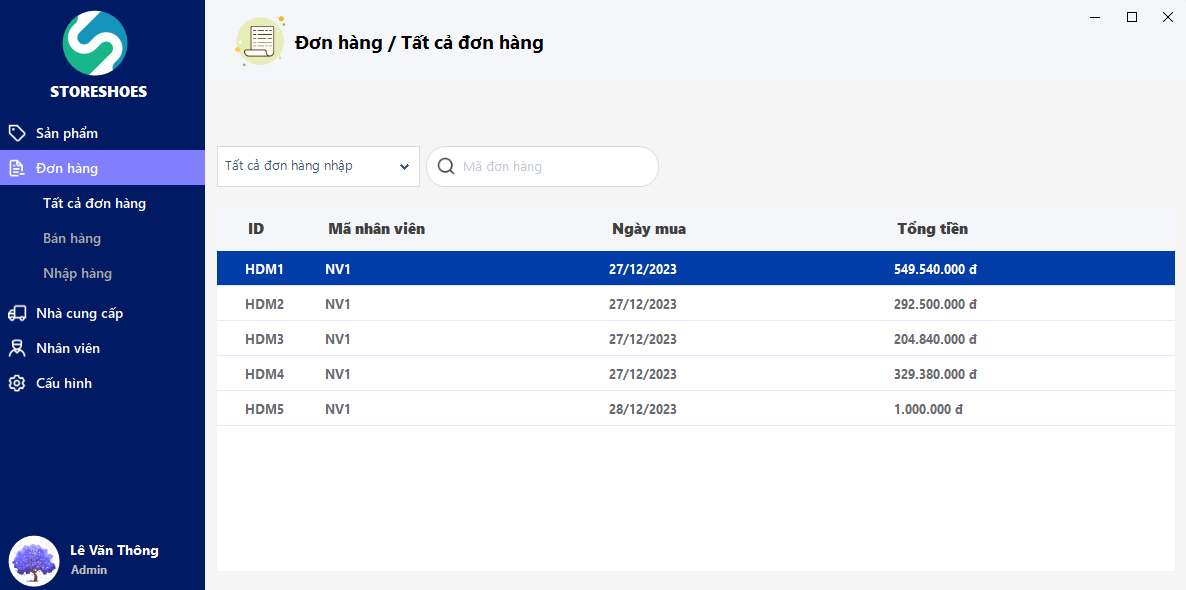
Đây là giao diện để hiển thị thông tin tất cả những đôi giày có trong kho của cửa hàng. Giúp người dùng có thể nắm bắt được những thông tin có bản “ Tên sản phẩm”, “Tồn kho”, “Giá nhập hàng”, “Giá bán hàng”.

Khi người dùng chọn sản phẩm và click và nút sửa sẽ hiện lên một khung cho người dung sửa thông tin của sản phẩm.



### 5.2.3 Giao diện khi chọn điều hướng đơn hàng

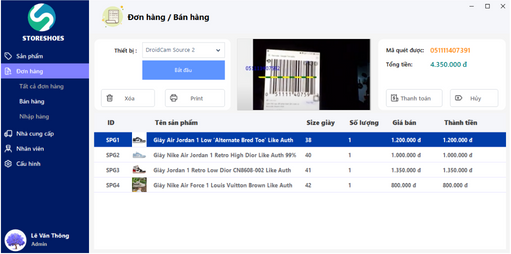
#### 5.2.3.1 Giao diện khi chọn điều hướng phụ “Tất cả đơn hàng”



Hình 5.4 Giao diện điều hướng tất cả đơn hàng

Đây là giao diện hiển thị các thông tin khi chọn option tương ứng“Tất cả đơn nhập hàng” và “Tất cả đơn bán hàng”. Giúp người dùng có thể nắm bắt được thông tin Nhân viên nào đã lập hóa đơn đó “Mã nhân viên”, “Ngày mua”, “Ngày bán”, “Tổng tiền”

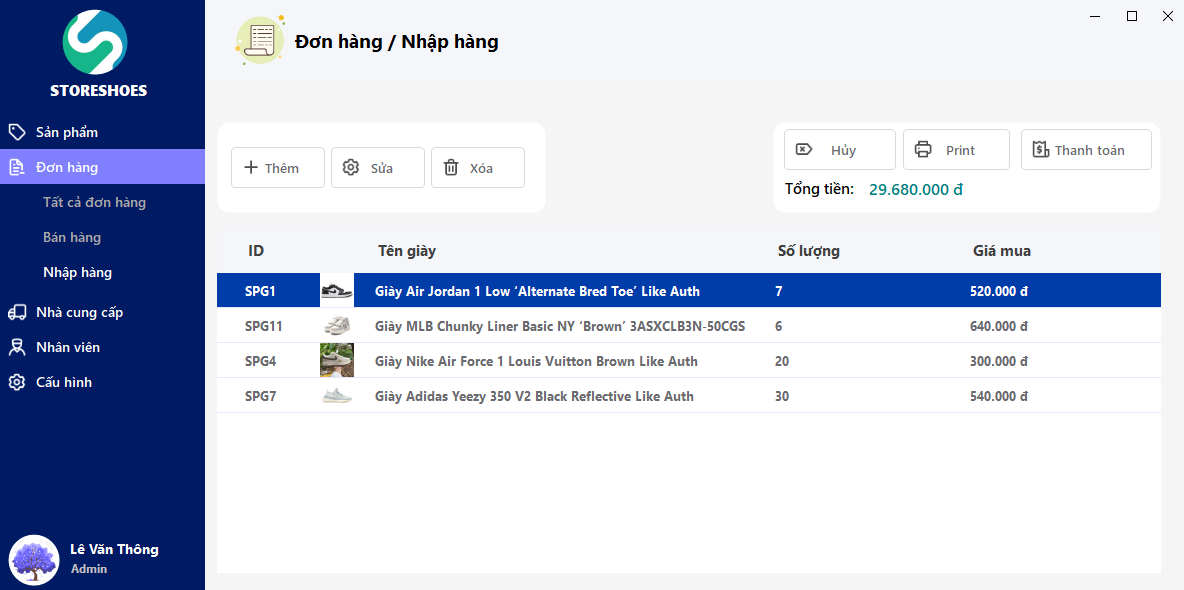
#### 5.2.3.2 Giao diện khi chọn điều hướng phụ “Bán hàng”



Hình 5.5 Giao diện điều hướng bán hàng

Đây là giao diện hiển thị thông tin về sản phẩm khi người dùng quét mã barcode trên một tấm thẻ gán ở đôi giày.Giúp người dùng có thể cập nhật tồn kho và cộng lại tất cả các chi phí liên quan đến đôi giày.

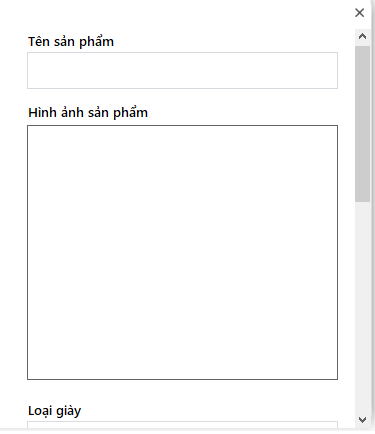
#### 5.2.3.3 Giao diện khi chọn điều hướng phụ “Nhập hàng”



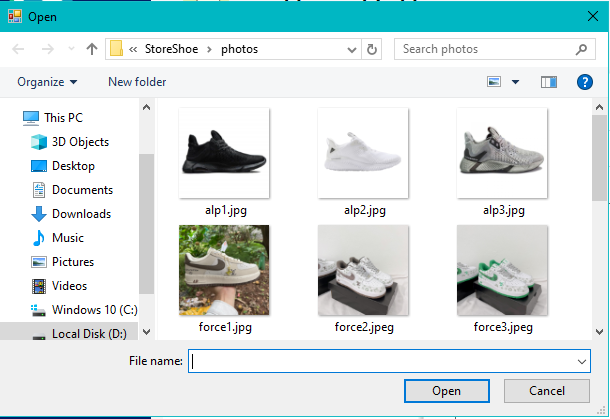
Hình 5.6 Giao diện điều hướng nhập hàng

Đây là giao diện hiển thị thông tin về sản phẩm đang được nhập vào cửa hàng.Giúp người dùng có thể cập nhật được sản phẩm mới vào cửa hàng, Nếu sản phẩm nhập nhập vào đã tồn tại trong cửa hàng thì sẽ được cập nhật lại tồn kho của sản phẩm đó

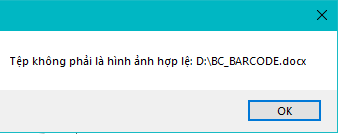
Sau đây là giao diện khi người dùng nhấn nút thêm để thêm thông tin về sản phầm được nhập:



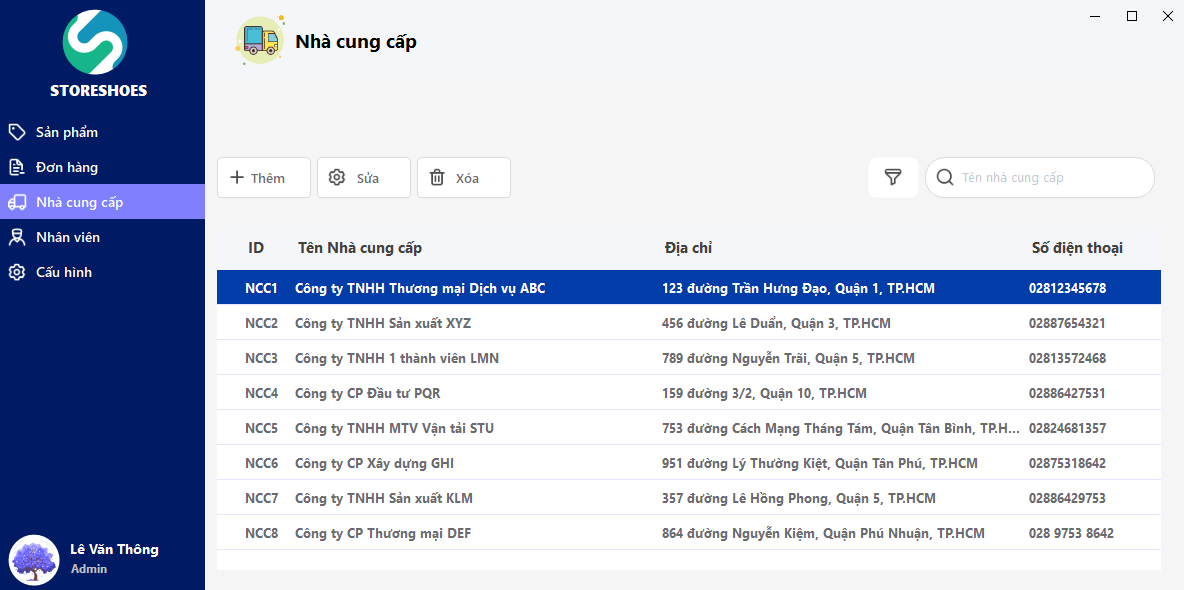
Khi ngươi dùng nhấn liên tục 2 lần vào ô “Hình ảnh sản phẩm” sẽ hiện lên 1 cửa sổ cho người dùng chọn file ảnh.:



Nếu người dụng chọn file không phải là định dạng ảnh, hoặc không chọn file sẽ hiện lên một thông báo lỗi:



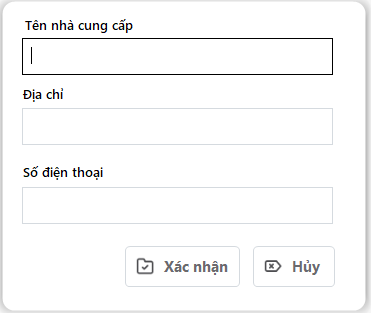
### 5.2.4 Giao diện khi chọn điều hướng Nhà cung cấp



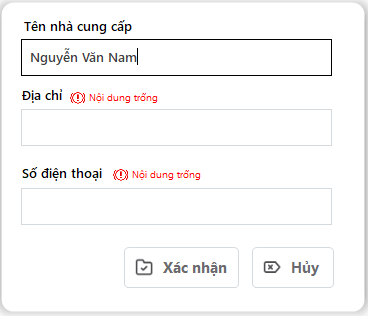
Hình 5.7 Giao diện điều hướng Nhà cung cấp

Đây là giao diện để hiển thị thông tin về các nhà cung cấp của cửa hàng. Giúp người dùng có thể nắm bắt được thông tin cơ bản như “Tên nhà cung cấp”, “Địa chỉ”, ‘Số điện thoại”

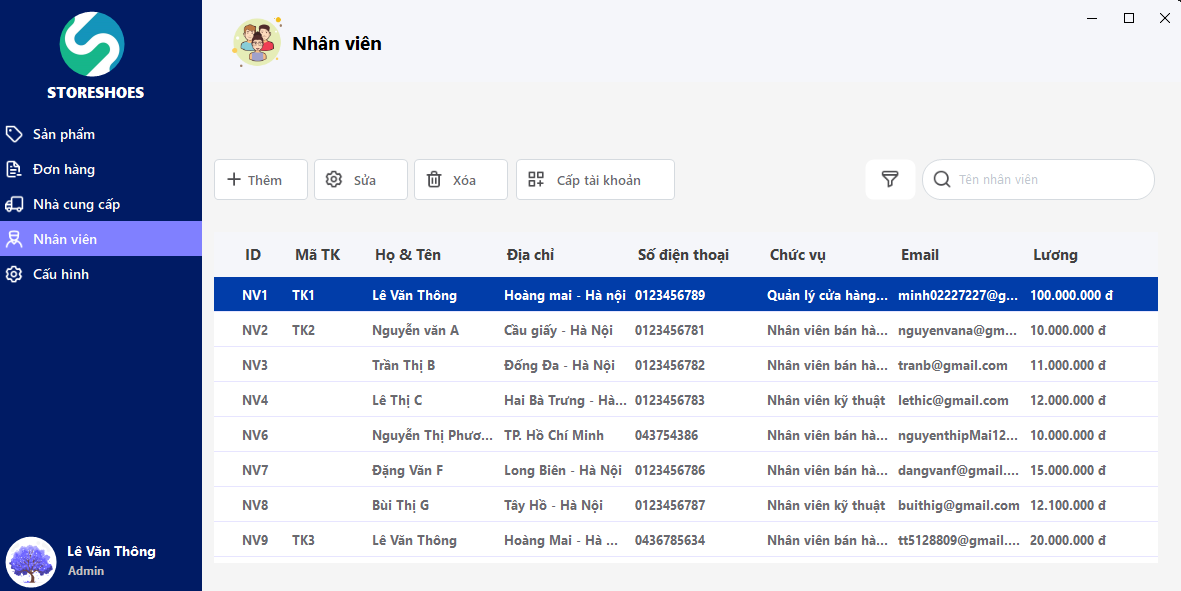
Sau đây là giao diện khi người dùng nhấn nút thêm để thêm thông tin về nhà cung cấp



Sau đây là giao diện khi người dùng nhập thiếu thông tin và ấn nút xác nhận, sẽ hiện lên các cảnh báo yêu cầu người dùng phải điền đầy đủ thông tin



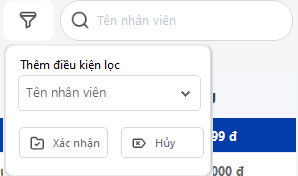
### 5.2.5 Giao diện khi chọn điều hướng nhân viên



Hình 5.8 Giao diện điều hướng nhân viên

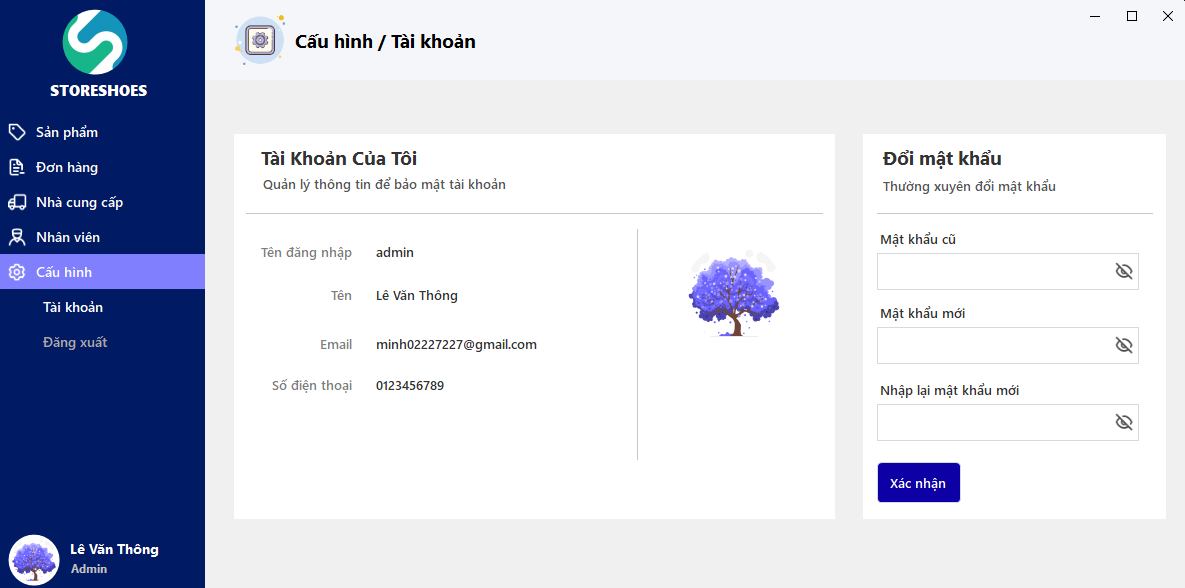
Đây là giao diện để hiển thị thông tin về các nhân viên. Giúp người dùng có thể nắm bắt được các thông tin cơ bản “Họ & Tên”, “Địa chỉ”, “Số điẹn thoại”, “Chức vụ”, “Email”, “Lương”

Sau đây là giao diện khi người dùng nhấn vào biểu tượng Fillter bên cạnh thanh tìm kiếm



### 5.2.6 Giao diện khi chọn điều hướng cấu hình

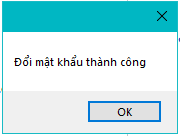
#### 5.2.6.1 Giao diện khi chọn điều hướng phụ Tài khoản



Hình 5.9 Giao diện điều hướng tài khoản

Giao diện này hiển thị thông tin cá nhân liên quan đến người dùng tài khoản và có thêm phần thay đổi mật khẩu để giúp người dùng có thể thay đổi mật khẩu theo sở thích.

Khi nhập đúng và điền đầy đủ tất cả các thông tin bên phần “Đổi mật khẩu” và nhấn nút xác nhận người dùng sẽ nhận được thông báo sau



# KẾT LUẬN VÀ KIẾN NGHỊ

Để xây dựng đồ án có đề tài trên thì cần khảo sát thực tế trên thị thị trường hiện nay, những nhu cầu theo xu hướng thiết thực của người tiêu dùng gồm những gì. Không những thế, việc khảo sát hệ thống cũng vô cùng quan trọng và cần thiết để tiến hành các bước tiếp theo trong dự án cần làm.

Đây là bài phân tích tổng hợp tất cả về xây dựng phần mềm quản lý shop giày Storeshoes của nhóm em trong quá trình thực hiện đồ án:

1. **Kết quả đã đạt được :**

Hoàn thành mục phân tích thiết kế hệ thống

Tạo dựng được chương trình cơ bản, dễ dử dụng và đáp ứng được các nhu cầu tối thiểu trong lĩnh vực kinh doanh

Quản lý chặt chẽ các vùng nhập dữ liệu được kiểm soát, tránh sai sót thông tin trong quá trình lập dữ liệu

Thiết kế giao diện đơn giản, đẹp mắt, tiện dụng và tối ưu hóa các nhu cầu cần thiết

1. **Kết quả chưa được đạt được như mong muốn**:

Do còn hạn chế về mặt kiến thức nên thiếu đi sự đa dạng về cách xây dựng đề tài, chưa đầy đủ các thông tin khảo sát so với thực tế

Chưa xây dựng được một chương trình đầy đủ tất cả các chức năng

Chưa phát triển được chức năng hỗ trợ khách hàng khi cần thiết

**3. Phương hướng phát triển:**

bổ sung thêm kiến thức để nâng cấp hệ thống kinh doang sao cho sát với thực tế nhất có thể

Xây dựng chương trình dễ tiếp cận với nhu cầu thị trường, sử dụng, dễ cài đặt và tốn ít dữ liệu bộ nhớ

-Cải thiện lại thiết bị truy cập cũng như tìm kiếm có thể thuận tiện, nhanh chóng và cải tiến hơn về mặt dữ liệu

Nâng cao các phương thức tiếp cận khách hàng và hỗ trợ khách hàng

# TÀI LIỆU THAM KHẢO

[1] Giới thiệu về lập trình winform C#. [Trực tuyến]. Địa chỉ: <https://codegym.vn/blog/tai-lieu-lap-trinh-c-winform/>

[2] SQL Server là gì? Hướng dẫn cài đặt SQL Server. [Trực tuyến]. Địa chỉ:

<https://vietnix.vn/sql-server-la-gi/>

[3] Giới thiệu ado.net và kết nối SQL Server với SqlConnection. [Trực tuyến]. Địa chỉ: <https://xuanthulab.net/ado-net-gioi-thieu-ado-net-va-ket-noi-sql-server-voi-sqlconnection.html>

[4] Sơ đồ luồng dữ liệu (Data Flow Diagram - DFD). [Trực tuyến]. Địa chỉ:

<https://vietnambiz.vn/so-do-luong-du-lieu-data-flow-diagram-dfd-la-gi-20190909111117996.htm>

[5] What is C# Programing & What Are Its Strengths?[Trực tuyến]. Địa chỉ:

<https://www.lateral.com.au/blog/august-2018/what-is-c-programing-what-are-its-strengths>