**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT**

**---****---**

A blue and white logo

Description automatically generated

ĐỒ ÁN MÔN HỌC

CÔNG NGHỆ XML

ĐỀ TÀI: QUẢN LÝ BÁN HÀNG THỜI TRANG

Giáo viên hướng dẫn: Thầy Ngô Lê Quân

Nhóm 08

Sinh viên thực hiện:

Bùi Xuân Văn 22115053122147

Trần Đình Việt 22115053122149

Nguyễn Văn Hải 22115053122208

***Đà Nẵng, 15 tháng 12 năm 2024***

# LỜI MỞ ĐẦU

Chúng ta đang sống trong một kỷ nguyên mà mọi thành tựu khoa học và công nghệ đều xuất hiện một cách hết sức mau lẹ và cũng được đổi mới một cách cực kì nhanh chóng. Ngành công nghệ thông tin cũng ngày càng chứng tỏ được vị trí của mình bằng sự xuất hiện trong rất nhiều các lĩnh vực, mọi nơi, mọi chốn của đời sống. Có thể nói, nền kinh tế toàn cầu đang chịu ảnh hưởng sâu sắc từ công nghệ thông tin mà trong đó có không ít vai trò của các ngành công nghệ phần mềm. Các phần mềm làm ra được ứng dụng phục vụ cho công viêc, hoạt động của các công ty, tổ chức như : các chương trình quản lý nhân sự, quản lý kho, quản lý bến bãi, quản lý hệ thống bán hàng… ngày càng nhiều .Và chúng đều thể hiện được tầm quan trọng của mình đối với tổ chức.

Tuy nhiên, để làm ra được một phần mềm hoàn hảo và đáp ứng được mọi nhu cầu của người dùng thì không phải lài dễ. Nó cần sự hỗ trợ rất lớn của các chuyên ngành trong lĩnh vực công nghệ thông tin. Trong đó, Công nghệ XML là một công nghệ có đóng góp tương đối lớn trong việc hỗ trợ hoàn thành các sản phầm đó. Vì vậy, mà nhóm chúng em quyết định áp dụng công nghệ XML vào trong việc xây dựng chương trình “Quản lý bán hàng thời trang” trong đề tài bài tập lớn được giao. Qua đây chúng em xin bày tỏ lòng cảm ơn sâu sắc tới thầy Ngô Lê Quân, người đã nhiệt tình chỉ bảo cho chúng em trong quá trình thực hiện đề tài. Chúng em xin chân thành cảm ơn!

Nhóm sinh viên thực hiện!

# MỤC LỤC

**[LỜI MỞ ĐẦU 1](#_heading=h.gjdgxs)**

**[MỤC LỤC 2](#_heading=h.30j0zll)**

**[Chương 1: Giới thiệu 3](#_heading=h.3znysh7)**

**[I. Ích Lợi Khi Sử Dụng XML 3](#_heading=h.2et92p0)**

**[II. Mục tiêu 3](#_heading=h.2s8eyo1)**

**[III. Phạm vi đề tài, yêu cầu 3](#_heading=h.17dp8vu)**

**[Chương 2: Nội dung 4](#_heading=h.3rdcrjn)**

**[I. Khảo Sát Nghiệp Vụ 4](#_heading=h.26in1rg)**

**[1. Khảo sát và đánh giá hiện trạng 4](#_heading=h.lnxbz9)**

**[2. Mô hình hoạt động nghiệp vụ của quản lý bán](#_heading=h.35nkun2) hàng thời trang 4**

**[II. Phân tích thiết kế hệ thống 6](#_heading=h.1ksv4uv)**

**[1. Mô tả vấn đề 6](#_heading=h.44sinio)**

**[2. Đặc tả yêu cầu 6](#_heading=h.2jxsxqh)**

**[III. Thiết kế cơ sở dữ liệu 8](#_heading=h.z337ya)**

**[1. Mô hình phân cấp chức năng 8](#_heading=h.3j2qqm3)**

**[2. Mô hình luồng dữ liệu mức khung cảnh 8](#_heading=h.1y810tw)**

**[3. Mô hình dữ liệu mức đỉnh 9](#_heading=h.4i7ojhp)**

**[4. Xác định các thực thể 9](#_heading=h.2xcytpi)**

**[5. Mô hình dữ liệu mức quan hệ 11](#_heading=h.1ci93xb)**

**[6. Các bảng dữ liệu 11](#_heading=h.3whwml4)**

**[IV. Thiết kế và cài đặt chương trình 13](#_heading=h.2bn6wsx)**

**[1. Các module chính 13](#_heading=h.qsh70q)**

**[2. Giao diện chương trình 22](#_heading=h.3as4poj)**

**[Chương 3: Kết luận 26](#_heading=h.1pxezwc)**

# Chương 1: Giới thiệu

## Ích Lợi Khi Sử Dụng XML

* XML có thể tách rời dữ liệu,sử dụng XML dữ liệu sẽ được lưu trữ trong những tập tin XML riêng biệt.
* XML có thể mô tả thông tin của những đối tượng phức tạp mà CSDL quan hệ không thể giải quyết được
* XML dùng để chia sẻ dữ liệu với những tập tin văn bản dễ hiểu
* XML có thể dùng để chuyển đổi dữ liệu giữa các hệ thống không tương thích
* XML dùng để lưu trữ dữ liệu làm cho dữ liệu của chúng ta hữu ích hơn

## Mục tiêu

* + Tìm hiểu đề tài, khảo sát thực trạng nhu cầu và hoạt động quản lý bán hàng thời trang của thị trường trong thời điểm hiện tại.
  + Thiết kế phần mềm quản lý bán đĩa với mục tiêu hướng người sử dụng và hiệu quả: dễ sử dụng, quản lý tốt, giao diện bắt mắt, thao tác nhanh, dễ dàng….

## Phạm vi đề tài, yêu cầu

Phần mềm được xây dựng trên phạm vi môn học “ Công Nghệ XML”

Với editor là Visual Studio 2010 và cơ sở dữ liệu dạng file XML.

+ Yêu cầu:

* Cơ sở dữ liệu
* XSD (lược đồ)
* XML
* XSLT
* Form DataBase > xml
* Thêm
* Sửa xóa
* Hiển thị trình duyệt
* Cập nhật dữ liệu database
* Kiểm tra hợp lệ

# Chương 2: Nội dung

1. **Khảo Sát Nghiệp Vụ**
2. **Khảo sát và đánh giá hiện trạng**

Mặt hàng thời trang là một trong những sản phẩm phổ biến, đáp ứng nhu cầu đa dạng của khách hàng từ nhiều tầng lớp xã hội và độ tuổi khác nhau. Các mặt hàng thời trang bao gồm: quần áo, giày dép, túi xách, phụ kiện...

Vì vậy, bài toán quản lý bán hàng thời trang là sự kết hợp giữa việc quản lý danh mục sản phẩm với quản lý xuất nhập hàng hóa. Do hệ thống sử dụng cho các đơn vị kinh doanh từ nhỏ đến vừa, và mặt hàng kinh doanh là các sản phẩm thời trang đa dạng, bài toán cần tập trung vào việc quản lý danh mục sản phẩm và hệ thống bán hàng, không đề cập đến vấn đề của đơn vị cung cấp và khách hàng trong chương trình.

Bài toán tập chung vào các vấn đề sau:

* Quản lý nhân viên trong cửa hàng
* Quản lý các hóa đơn bán hàng
* Quản lý các mặt hàng thời trang
* Quản lý tài khoản người dùng phần mềm

Yêu cầu chương trình phần mềm:

* Phần mềm phải dễ sử dụng, dễ cài đặt, tạo điều kiện thuận lợi cho người dùng trong việc thao tác trên phần mềm khi sử dụng.
* Phần mềm phải giao tiếp dễ dang với người sử dụng, giao diện thân thiện và thích ứng với các phần cứng của máy tính.
* Phần mềm có phân quyền cho nhân viên và người quản lý trong hệ thống để quản lý có thể toàn quyền kiểm soát.

Yêu cầu dữ liệu:

* Dữ liệu phải được thiết kế chính xác, không được thiếu cho việc quản lý.
* Xử lý được các chức năng trong hệ thống.

1. **Mô hình hoạt động nghiệp vụ của quản lý bán đĩa phim**

* Cơ sở bán hàng thời trang cần thực hiện phân loại các sản phẩm theo danh mục như quần áo, giày dép, túi xách, phụ kiện... Cập nhật danh sách các danh mục sản phẩm vào máy tính, phân vùng các sản phẩm và trưng bày tại nơi mà khách hàng dễ tìm kiếm để thuận lợi cho việc tra cứu và mua sắm.
* Người kinh doanh phải thường xuyên thiết kế và trưng bày các bảng mục lục danh sách các đĩa, nội dung các đĩa gồm những gì, trợ giúp cho khách hàng đến tra cứu và tìm kiếm đĩa phim muốn mua.
* Người kinh doanh thực hiện các công việc bán hàng, ghi chép sổ sách và nhật ký bán hàng. Hỗ trợ khách hàng trong công việc tra cứu thông tin về đĩa.
* Trên cơ sở dữ liệu sổ sách và thực tế đề ra danh sách các đĩa phim cần phải bổ sung và nhập vào
* Kiểm tra định kỳ danh sách các đĩa về số lượng, số lượng bán được, số lượng nhập và số lương tồn đọng.
* Qua đó, nhận thấy việc sắp xếp, phân loại sản phẩm thời trang khá phức tạp, nhiều loại dễ gây nhầm lẫn cho người kinh doanh. Việc quản lý bán hàng cũng khó xử lý, khó tra cứu. Vì vậy, việc tạo ra phần mềm Quản lý bán hàng thời trang đặt ra cần đáp ứng được yêu cầu nghiệp vụ trên.

1. **Phân tích thiết kế hệ thống**
2. **Mô tả vấn đề**

**-** Với phần mềm phục vụ cho công việc quản lý bán đĩa phim, có thể nhận thấy:

+ Mục đích của hệ thống: tin học hóa việc về dịch vụ bán hàng thời trang cũng như quản lý của cửa hàng.

+ Các chức năng chính:

* Quản lý nhân viên của cửa hàng.
* Quản lý sản phẩm thời .
* Quản lý tài khoản người dùng

+ Đối tượng sử dụng và vai trò của mỗi đối tượng:

* Chủ cửa hàng: quản lý nhân viên, nhận báo cáo từ nhân viên.
* Nhân viên: Quản lý hóa đơn và việc bán đĩa

1. **Đặc tả yêu cầu**

* Quản lý nhân viên:

Có thể thực hiện thêm nhân viên mới vào danh sách, sửa đổi khi có những biến đổi xảy ra và xóa bỏ nhân viên khi bị sa thải. Các thông tin về nhân viên gồm có: Mã nhân viên, Tên nhân viên, Điện thoại liên hệ và các mô tả khác.

* Quản lý sản phẩm thời trang:

Sản phẩm sẽ được thêm vào khi cửa hàng nhập hàng mới về. Sản phẩm sẽ được xóa đi mỗi khi có sản phẩm được bán. Khi nhân viên muốn tìm kiếm một sản phẩm theo tên hoặc mã sản phẩm, sản phẩm sẽ được cập nhật số lượng và sẽ được sửa mỗi khi có sai sót.

Thông tin về sản phẩm bao gồm: Mã sản phẩm, tên sản phẩm, loại sản phẩm, giá thành, kích thước, số lượng tồn kho, hình ảnh.

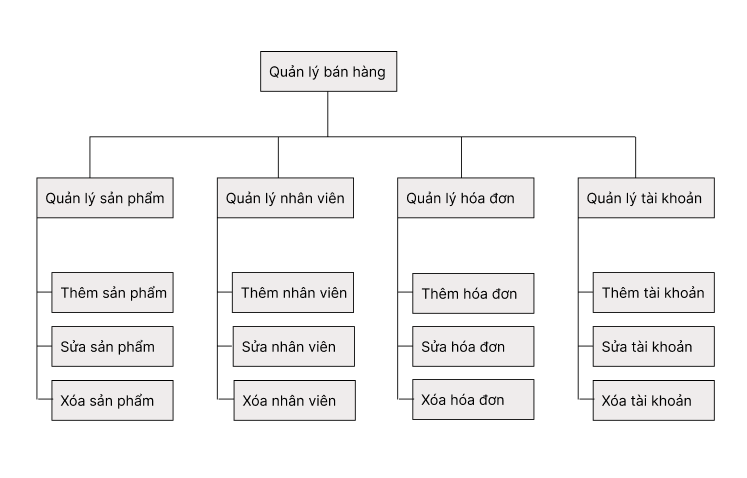
* Quản lý hóa đơn:

Mỗi khách hàng đến mua đĩa sẽ được nhận một hóa đơn. Trong đó bao gồm: mã hóa đơn, mã sản phẩm, số lượng mua, mã nhân viên, ngày lập.

* Quản lý tài khoản:

Mỗi nhân viên sử dụng phần mềm sẽ được cung cấp một tài khoản. Tài khoản bao gồm: Tài khoản, mật khẩu và quyền hạn.

1. **Thiết kế cơ sở dữ liệu**
2. **Mô hình phân cấp chức năng**

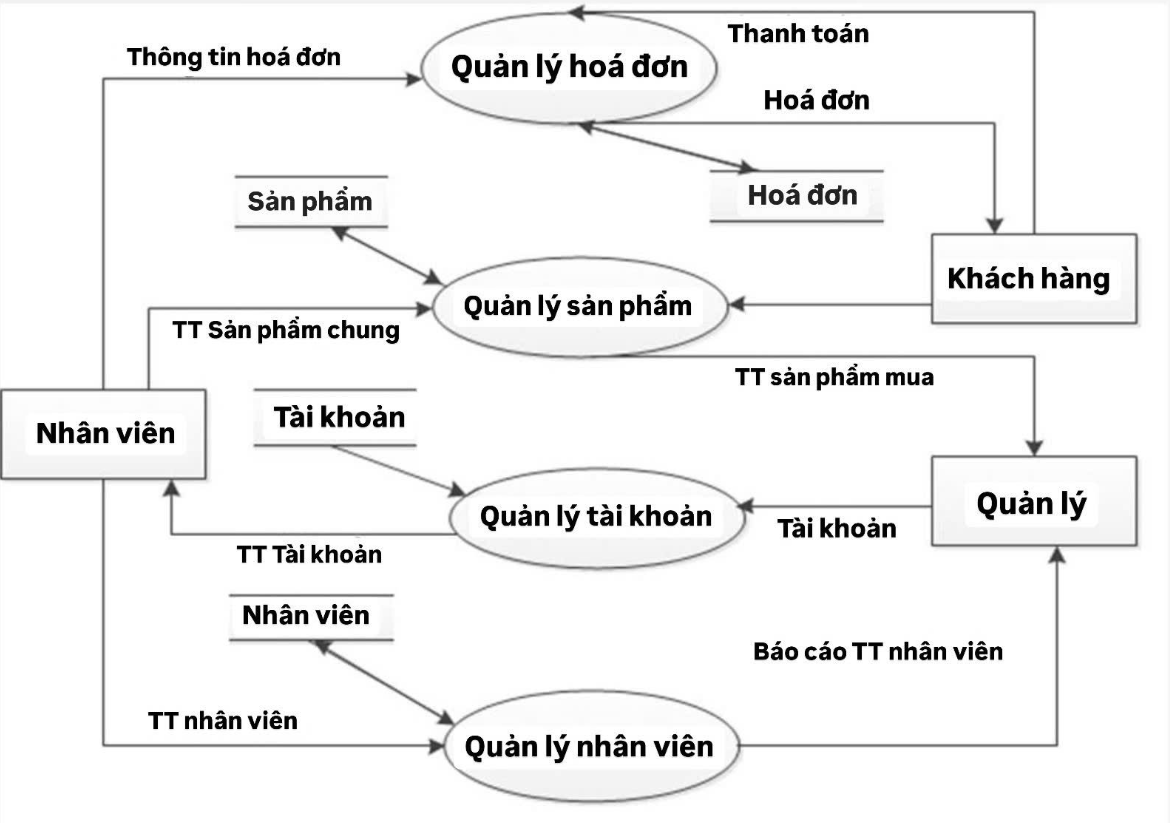


1. **Mô hình luồng dữ liệu mức khung cảnh**

A diagram with text and arrows

Description automatically generated with medium confidence

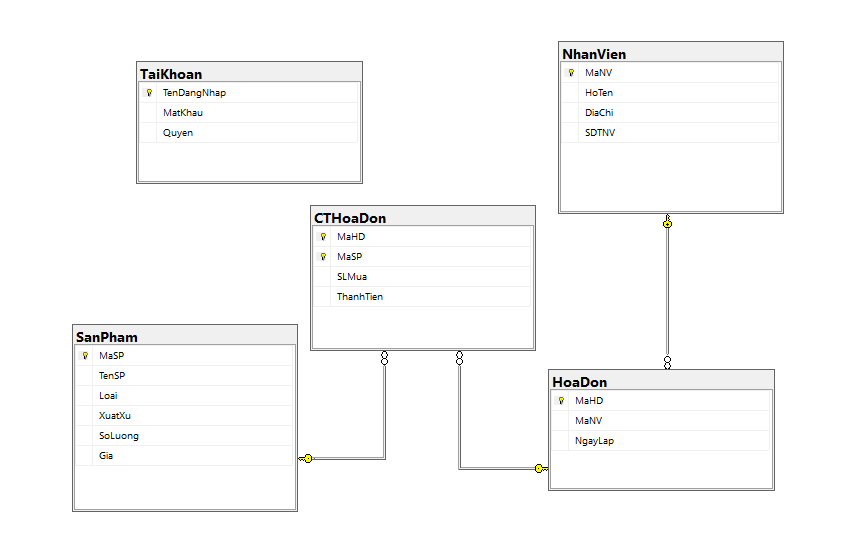
1. **Mô hình dữ liệu mức đỉnh**



1. **Xác định các thực thể**

|  |  |
| --- | --- |
| Tên thực thể | Thuộc tính |
| Nhân viên | * Mã nhân viên * Họ tên * Địa chỉ * Số điện thoại |
| Tài khoản | * Tên tài khoản * Mật khẩu * Quyền hạn |
| Sản phẩm | * Mã sản phẩm * Tên sản phẩm * Phân loại sản phẩm * Xuất xứ * Số lượng * Giá |
| Hóa đơn | * Mã hóa đơn * Mã nhân viên * Ngày lập |
| Chi tiết hóa đơn | * Mã hóa đơn * Mã đĩa * Số lượng mua * Thành tiền |

1. **Mô hình dữ liệu mức quan hệ**



1. **Các bảng dữ liệu**

Bảng tài khoản:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu(Độ rộng) | Ràng buộc | Khóa chính/  Khóa Phụ  (PK/FK) | Mô tả |
| 1 | TaiKhoan | Nvchar(50) | Not null | PK | Tài khoản |
| 2 | matkhau | Nvchar(50) | Null |  | Mật khẩu |
| 3 | Quyen | Int | Null |  | Quyền hạn |

Bảng nhân viên:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu(Độ rộng) | Ràng buộc | Khóa chính/  Khóa Phụ  (PK/FK) | Mô tả |
| 1 | MaNV | NVChar(5) | Not null | PK | Mã nhân viên |
| 2 | HoTen | NVChar(100) | Not null |  | Tên nhân viên |
| 3 | DiaChi | NVChar(150) | Not null |  | Địa chỉ |
| 4 | SDTNV | NVChar(11) | Not null |  | SDT nhân viên |

Bảng sản phẩm:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu(Độ rộng) | Ràng buộc | Khóa chính/  Khóa Phụ  (PK/FK) | Mô tả |
| 1 | MaSP | NCHAR(10) | Not null | PK | Mã sản phẩm (quần áo) |
| 2 | TenSP | NVARCHAR(100) | Null |  | Tên sản phẩm (tên quần áo) |
| 3 | Loai | NVARCHAR(50) | Null |  | Phân loại sp |
| 4 | XuatXu | NVARCHAR(50) | Null |  | Xuất xứ |
| 5 | SoLuong | INT | Null |  | Số lượng |
| 6 | Gia | DECIMAL(18, 0) | Null |  | Giá |

Bảng hóa đơn:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu(Độ rộng) | Ràng buộc | Khóa chính/  Khóa Phụ  (PK/FK) | Mô tả |
| 1 | MaHD | NChar(10) | Not null | PK | Mã hóa đơn |
| 2 | NgayLap | DATE | Null |  | Ngày lập |
| 3 | MaNV | NVChar(5) | Not null | FK | Mã nhân viên |

Bảng chi tiết hóa đơn:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu(Độ rộng) | Ràng buộc | Khóa chính/  Khóa Phụ  (PK/FK) | Mô tả |
| 1 | MaHD | NChar(10) | Not null | PK | Mã hóa đơn |
| 2 | MaSP | NCHAR(10) | Not null | PK | Mã sản phẩm (quần áo) |
| 3 | SLMual | INT | Not null |  | Số lượng mua |
| 4 | ThanhTien | DECIMAL(18, 0) | Not null |  | Thành tiền |

1. **Thiết kế và cài đặt chương trình**
2. **Các module chính**

Các Class( Gồm phương thức thêm, sửa, xóa , tìm kiếm, kiểm tra và hàm tạo XML, Load lên DataGridview)

1. **Form đăng nhập**

using System;

using System.Linq;

using System.Xml.Linq;

using System.Windows.Forms;

namespace ShopQuanAo

{

public partial class DangNhap : Form

{

public DangNhap()

{

InitializeComponent();

}

private void guna2Button1\_Click(object sender, EventArgs e)

{

// Lấy giá trị từ các textbox

string tenDN = txttaikhoan.Text;

string matKhau = txtMatKhau.Text;

// Đường dẫn đến file XML chứa thông tin tài khoản

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\TaiKhoan.xml";

try

{

// Load file XML

XDocument doc = XDocument.Load(xmlFilePath);

// Tìm kiếm tài khoản khớp tên đăng nhập và mật khẩu trong XML

var user = doc.Descendants("TaiKhoan")

.FirstOrDefault(u =>

u.Element("TenDangNhap") != null && u.Element("MatKhau") != null &&

u.Element("TenDangNhap").Value == tenDN &&

u.Element("MatKhau").Value == matKhau);

// Kiểm tra nếu tìm thấy thông tin đăng nhập

if (user != null)

{

// Kiểm tra quyền của tài khoản

string quyen = user.Element("Quyen")?.Value;

// Nếu không phải Admin

if (quyen != "Admin")

{

MessageBox.Show("Bạn chỉ có thể đăng nhập với tài khoản có quyền Admin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

MessageBox.Show("Đăng nhập thành công với quyền Admin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

// Tạo và hiển thị Form Main

Main main = new Main();

main.Show();

// Ẩn Form hiện tại

this.Hide();

}

}

else

{

MessageBox.Show("Sai tên đăng nhập hoặc mật khẩu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi đọc file XML: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void guna2Button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void DangNhap\_Load(object sender, EventArgs e)

{

}

private void guna2HtmlLabel1\_Click(object sender, EventArgs e)

{

}

}

}

1. **Form main (trang chính)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ShopQuanAo

{

public partial class Main : Form

{

public Main()

{

InitializeComponent();

}

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

{

this.Close();

// Tạo instance của Form QLNV

QLNV qlnvForm = new QLNV();

// Hiển thị Form QLNV

qlnvForm.Show();

}

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

{

this.Hide();

// Tạo instance của Form QLNV

QLSP qlnvForm = new QLSP();

// Hiển thị Form QLNV

qlnvForm.Show();

}

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

{

this.Close();

// Tạo instance của Form QLTK

QLTK TKForm = new QLTK();

// Hiển thị Form QLTK

TKForm.Show();

}

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

{

this.Close();

DangNhap dn = new DangNhap();

dn.Show();

}

private void Main\_Load(object sender, EventArgs e)

{

}

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

{

this.Close();

// Tạo instance của Form QLSP

QLHD qlhdForm = new QLHD();

// Hiển thị Form QLSP

qlhdForm.Show();

}

}

}

1. **Form quản lý đơn hàng**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Web.UI.WebControls;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

using System.Xml.Linq;

using System.Xml;

using System.Diagnostics;

using System.IO;

namespace ShopQuanAo

{

public partial class QLHD : Form

{

public QLHD()

{

InitializeComponent();

this.StartPosition = FormStartPosition.CenterScreen;

}

private void QLHD\_Load(object sender, EventArgs e)

{

LoadNhanVien();

// Đường dẫn tới file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\CTHoaDon.xml";

try

{

// Kiểm tra xem file XML có tồn tại không

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML

DataSet dataSet = new DataSet();

dataSet.ReadXml(xmlFilePath);

// Kiểm tra xem có bảng dữ liệu không

if (dataSet.Tables.Count > 0)

{

// Gắn dữ liệu vào DataGridView

dataGridView1.DataSource = dataSet.Tables[0];

}

else

{

MessageBox.Show("Không tìm thấy bảng dữ liệu trong file XML!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (System.IO.IOException ioEx)

{

MessageBox.Show("Lỗi khi đọc file XML: " + ioEx.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (Exception ex)

{

MessageBox.Show("Đã xảy ra lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void dataGridView1\_SelectionChanged(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

// Lấy dữ liệu từ dòng đã chọn

var row = dataGridView1.SelectedRows[0];

txtMahd.Text = row.Cells["MaSP"].Value.ToString();

txtMasp.Text = row.Cells["TenSP"].Value.ToString();

txtslmua.Text = row.Cells["Gia"].Value.ToString();

}

}

private void UpdateSqlFromXml()

{

// Đường dẫn đến file XML CTHoaDon

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\CTHoaDon.xml";

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

try

{

if (System.IO.File.Exists(xmlFilePath))

{

XDocument doc = XDocument.Load(xmlFilePath);

foreach (XElement cthdElement in doc.Descendants("CTHoaDon"))

{

string maHD = cthdElement.Element("MaHD")?.Value;

string maSP = cthdElement.Element("MaSP")?.Value;

string slMua = cthdElement.Element("SLMua")?.Value;

string thanhTien = cthdElement.Element("ThanhTien")?.Value;

if (!string.IsNullOrEmpty(maHD) && !string.IsNullOrEmpty(maSP) &&

!string.IsNullOrEmpty(slMua) && !string.IsNullOrEmpty(thanhTien))

{

if (int.TryParse(slMua, out int slMuaInt) && decimal.TryParse(thanhTien, out decimal thanhTienDec))

{

string sqlQuery = @"

IF EXISTS (SELECT 1 FROM CTHoaDon WHERE MaHD = @MaHD AND MaSP = @MaSP)

UPDATE CTHoaDon SET SLMua = @SLMua, ThanhTien = @ThanhTien

WHERE MaHD = @MaHD AND MaSP = @MaSP

ELSE

INSERT INTO CTHoaDon (MaHD, MaSP, SLMua, ThanhTien)

VALUES (@MaHD, @MaSP, @SLMua, @ThanhTien)";

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

using (SqlCommand cmd = new SqlCommand(sqlQuery, conn))

{

cmd.Parameters.AddWithValue("@MaHD", maHD);

cmd.Parameters.AddWithValue("@MaSP", maSP);

cmd.Parameters.AddWithValue("@SLMua", slMuaInt);

cmd.Parameters.AddWithValue("@ThanhTien", thanhTienDec);

cmd.ExecuteNonQuery();

}

}

}

else

{

MessageBox.Show($"Dữ liệu không hợp lệ cho CTHoaDon với mã hóa đơn {maHD} và mã sản phẩm {maSP}. Số lượng hoặc thành tiền không đúng.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

MessageBox.Show("Dữ liệu đã được cập nhật thành công vào cơ sở dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("File XML không tồn tại!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi cập nhật dữ liệu vào cơ sở dữ liệu SQL Server: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void groupBox2\_Enter(object sender, EventArgs e)

{

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void guna2DataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

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

{

this.Close();

DangNhap dn = new DangNhap();

dn.Show();

}

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

{

this.Close();

// Tạo instance của Form QLTK

QLTK TKForm = new QLTK();

// Hiển thị Form QLTK

TKForm.Show();

}

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

{

}

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

{

this.Close();

// Tạo instance của Form QLSP

QLSP qlspForm = new QLSP();

// Hiển thị Form QLSP

qlspForm.Show();

}

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

{

this.Close();

// Tạo instance của Form QLSP

QLNV qlnvForm = new QLNV();

// Hiển thị Form QLSP

qlnvForm.Show();

}

private void menuStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void QLHD\_Load\_1(object sender, EventArgs e)

{

}

private decimal CalculateThanhTien(string maSP, int slMua)

{

decimal giaSP = 0;

// Truy vấn giá sản phẩm từ cơ sở dữ liệu

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

string query = "SELECT Gia FROM SanPham WHERE MaSP = @MaSP"; // Sửa lại tên cột Gia

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

using (SqlCommand cmd = new SqlCommand(query, conn))

{

cmd.Parameters.AddWithValue("@MaSP", maSP);

var result = cmd.ExecuteScalar();

if (result != null)

{

decimal.TryParse(result.ToString(), out giaSP);

}

}

}

// Nếu không tìm thấy giá, trả về 0

return giaSP \* slMua;

}

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

{

string maHD = txtMahd.Text.Trim();

string maSP = txtMasp.Text.Trim();

string slMua = txtslmua.Text.Trim();

string maNV = comboBox1.SelectedValue?.ToString();

if (string.IsNullOrEmpty(maHD) || string.IsNullOrEmpty(maSP) || string.IsNullOrEmpty(slMua) || string.IsNullOrEmpty(maNV))

{

MessageBox.Show("Vui lòng điền đầy đủ các trường!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (!int.TryParse(slMua, out int slMuaInt) || slMuaInt <= 0)

{

MessageBox.Show("Số lượng mua phải là số nguyên dương!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

try

{

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

// Kiểm tra nếu cả MaHD và MaSP đã tồn tại trong bảng CTHoaDon

string checkBothQuery = "SELECT COUNT(\*) FROM CTHoaDon WHERE MaHD = @MaHD AND MaSP = @MaSP";

using (SqlCommand cmdCheckBoth = new SqlCommand(checkBothQuery, conn))

{

cmdCheckBoth.Parameters.AddWithValue("@MaHD", maHD);

cmdCheckBoth.Parameters.AddWithValue("@MaSP", maSP);

int countBoth = (int)cmdCheckBoth.ExecuteScalar();

if (countBoth > 0)

{

MessageBox.Show("Cặp mã hóa đơn và mã sản phẩm này đã tồn tại trong hệ thống!", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

}

// Thêm dữ liệu vào bảng Hóa Đơn (Cập nhật MaHD, MaNV, Ngày lập)

string insertHoaDonQuery = "INSERT INTO HoaDon (MaHD, MaNV, NgayLap) VALUES (@MaHD, @MaNV, @NgayLap)";

using (SqlCommand cmdInsertHoaDon = new SqlCommand(insertHoaDonQuery, conn))

{

cmdInsertHoaDon.Parameters.AddWithValue("@MaHD", maHD);

cmdInsertHoaDon.Parameters.AddWithValue("@MaNV", maNV);

cmdInsertHoaDon.Parameters.AddWithValue("@NgayLap", DateTime.Now); // Ngày lập là ngày hiện tại

cmdInsertHoaDon.ExecuteNonQuery();

}

// Thêm dữ liệu vào bảng CTHoaDon (chi tiết hóa đơn)

string insertCTHoaDonQuery = "INSERT INTO CTHoaDon (MaHD, MaSP, SLMua, ThanhTien) VALUES (@MaHD, @MaSP, @SLMua, @ThanhTien)";

using (SqlCommand cmdInsertCTHoaDon = new SqlCommand(insertCTHoaDonQuery, conn))

{

cmdInsertCTHoaDon.Parameters.AddWithValue("@MaHD", maHD);

cmdInsertCTHoaDon.Parameters.AddWithValue("@MaSP", maSP);

cmdInsertCTHoaDon.Parameters.AddWithValue("@SLMua", slMuaInt);

cmdInsertCTHoaDon.Parameters.AddWithValue("@ThanhTien", CalculateThanhTien(maSP, slMuaInt)); // Hàm tính thành tiền

cmdInsertCTHoaDon.ExecuteNonQuery();

}

// Lưu vào file HoaDon.xml

SaveCTHoaDonToXml(maHD, maSP, slMuaInt, CalculateThanhTien(maSP, slMuaInt));

MessageBox.Show("Dữ liệu đã được thêm vào cơ sở dữ liệu và HoaDon.xml thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

SaveHoaDonToXml(maHD, maNV, DateTime.Now);

// Thông báo thành công

MessageBox.Show("Dữ liệu đã được thêm vào cơ sở dữ liệu và HoaDon.xml thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

// Load lại dữ liệu hóa đơn từ bảng Hóa Đơn

LoadHoaDon(); // Gọi lại phương thức để load lại dữ liệu từ bảng HoaDon

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi thêm dữ liệu vào cơ sở dữ liệu: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

// Lấy mã sản phẩm cần xóa từ textbox

string maHD = txtMahd.Text.Trim();

string maSP = txtMasp.Text.Trim(); // Thêm trường mã sản phẩm (nếu cần)

// Đường dẫn đến file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\CTHoaDon.xml";

// Kết nối với cơ sở dữ liệu SQL Server

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

try

{

// Kiểm tra nếu file XML đã tồn tại

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML hiện tại

XDocument doc = XDocument.Load(xmlFilePath);

// Tìm phần tử CTHoaDon với MaHD và MaSP tương ứng để xóa

XElement cthoadonToDelete = doc.Descendants("CTHoaDon")

.FirstOrDefault(x => x.Element("MaHD")?.Value == maHD && x.Element("MaSP")?.Value == maSP);

if (cthoadonToDelete != null)

{

// Xóa phần tử CTHoaDon khỏi XML

cthoadonToDelete.Remove();

// Lưu lại file XML sau khi xóa

doc.Save(xmlFilePath);

// Cập nhật SQL Server - xóa dữ liệu khỏi bảng CTHoaDon

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string deleteQuery = "DELETE FROM CTHoaDon WHERE MaHD = @MaHD AND MaSP = @MaSP";

using (SqlCommand command = new SqlCommand(deleteQuery, connection))

{

command.Parameters.AddWithValue("@MaHD", maHD);

command.Parameters.AddWithValue("@MaSP", maSP);

command.ExecuteNonQuery();

}

}

MessageBox.Show("Xóa hóa đơn thành công từ file XML và cơ sở dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

QLHD\_Load(sender, e); // Cập nhật lại danh sách hóa đơn trên giao diện

}

else

{

MessageBox.Show("Không tìm thấy hóa đơn với mã " + maHD + " và mã sản phẩm " + maSP, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi xóa dữ liệu trong file XML hoặc cơ sở dữ liệu: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

string maHD = txtMahd.Text.Trim(); // Lấy mã hóa đơn từ textbox

string maSP = txtMasp.Text.Trim(); // Lấy mã sản phẩm từ textbox

string slMua = txtslmua.Text.Trim(); // Lấy số lượng mua từ textbox

if (string.IsNullOrEmpty(maHD) || string.IsNullOrEmpty(maSP) || string.IsNullOrEmpty(slMua))

{

MessageBox.Show("Vui lòng điền đầy đủ các trường để sửa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Nếu có trường trống, dừng lại

}

// Kiểm tra số lượng mua (phải là số nguyên và lớn hơn 0)

if (!int.TryParse(slMua, out int slMuaInt) || slMuaInt <= 0)

{

MessageBox.Show("Số lượng mua phải là số nguyên dương!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Đường dẫn đến file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\CTHoaDon.xml";

try

{

// Kiểm tra nếu file XML đã tồn tại

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML

XDocument doc = XDocument.Load(xmlFilePath);

// Tìm phần tử hóa đơn và sản phẩm tương ứng trong file XML

XElement itemToEdit = doc.Descendants("CTHoaDon")

.FirstOrDefault(x => x.Element("MaHD")?.Value == maHD && x.Element("MaSP")?.Value == maSP);

if (itemToEdit != null)

{

// Cập nhật số lượng mua mới

itemToEdit.Element("SLMua")?.SetValue(slMuaInt);

// Tính lại thành tiền nếu cần

itemToEdit.Element("ThanhTien")?.SetValue(CalculateThanhTien(maSP, slMuaInt));

// Lưu lại file XML sau khi sửa

doc.Save(xmlFilePath);

MessageBox.Show("Hóa đơn đã được sửa thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

// Cập nhật lại UI hoặc gọi hàm load dữ liệu nếu cần

QLHD\_Load(sender, e); // Ví dụ gọi lại hàm load dữ liệu hóa đơn

// Cập nhật cơ sở dữ liệu SQL Server từ XML

UpdateSqlFromXml(); // Giả sử bạn đã có hàm này để cập nhật SQL từ XML

}

else

{

MessageBox.Show("Không tìm thấy hóa đơn hoặc sản phẩm này!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi sửa hóa đơn: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void txtslmua\_TextChanged(object sender, EventArgs e)

{

}

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

{

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

try

{

// Câu lệnh SQL để lấy dữ liệu từ bảng CTHoaDon (hoặc bảng bạn cần load)

string sqlQuery = "SELECT MaHD, MaSP, SLMua, ThanhTien FROM CTHoaDon";

// Tạo kết nối đến cơ sở dữ liệu

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

// Tạo câu lệnh SQL

using (SqlCommand cmd = new SqlCommand(sqlQuery, conn))

{

// Tạo DataTable để lưu trữ dữ liệu

DataTable dt = new DataTable();

// Sử dụng SqlDataAdapter để điền dữ liệu vào DataTable

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt); // Điền dữ liệu vào DataTable

}

// Gán DataTable vào DataGridView để hiển thị

dataGridView1.DataSource = dt;

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi tải dữ liệu: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

// Lấy nội dung trong textBoxSearch và loại bỏ khoảng trắng thừa

string searchQuery = textBoxSearch.Text.Trim(); // Giữ nguyên nội dung tìm kiếm

// Kiểm tra nếu người dùng đã nhập từ khóa tìm kiếm

if (!string.IsNullOrEmpty(searchQuery))

{

try

{

// Câu lệnh SQL chỉ truy vấn bảng CTHoaDon và tìm kiếm theo MaHD hoặc MaSP

string sqlQuery = "SELECT c.MaHD, c.MaSP, c.SLMua, c.ThanhTien " +

"FROM CTHoaDon c " +

"WHERE c.MaHD = @searchQuery OR c.MaSP = @searchQuery";

// Tạo kết nối tới cơ sở dữ liệu SQL Server

using (SqlConnection conn = new SqlConnection(connectionString))

{

// Mở kết nối

conn.Open();

// Tạo câu lệnh SQL với tham số

using (SqlCommand cmd = new SqlCommand(sqlQuery, conn))

{

// Thêm tham số tìm kiếm

cmd.Parameters.AddWithValue("@searchQuery", searchQuery);

// Tạo DataTable để lưu kết quả

DataTable dt = new DataTable();

// Tạo DataAdapter để điền dữ liệu vào DataTable

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt); // Điền dữ liệu vào DataTable

}

// Kiểm tra nếu không có dữ liệu trả về

if (dt.Rows.Count == 0)

{

MessageBox.Show("Không tìm thấy hóa đơn hoặc sản phẩm với từ khóa: " + searchQuery, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

// Gán DataTable vào DataGridView

dataGridView1.DataSource = dt;

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi tìm kiếm: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

MessageBox.Show("Vui lòng nhập mã hóa đơn hoặc mã sản phẩm để tìm kiếm.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void button1\_Click(object sender, EventArgs e)

{

textBoxSearch.Clear();

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void LoadNhanVien()

{

string connectionString = "Data Source=.;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

string query = "SELECT MaNV, HoTen FROM NhanVien";

SqlDataAdapter da = new SqlDataAdapter(query, conn);

DataTable dt = new DataTable();

da.Fill(dt);

// Gán dữ liệu vào ComboBox1

comboBox1.DataSource = dt;

comboBox1.DisplayMember = "HoTen"; // Hiển thị tên nhân viên

comboBox1.ValueMember = "MaNV"; // Lấy mã nhân viên

}

}

private void LoadHoaDon()

{

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

try

{

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

string query = "SELECT MaHD, MaNV, NgayLap FROM HoaDon"; // Truy vấn từ bảng HoaDon

SqlDataAdapter da = new SqlDataAdapter(query, conn);

DataTable dt = new DataTable();

da.Fill(dt);

//// Gán dữ liệu vào DataGridView (giả sử bạn có DataGridView tên dgvHoaDon)

//dgvHoaDon.DataSource = dt;

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi tải dữ liệu hóa đơn: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void SaveHoaDonToXml(string maHD, string maNV, DateTime ngayLap)

{

string filePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\HoaDon.xml"; // Đường dẫn file HoaDon.xml

// Load file XML

XmlDocument xmlDoc = new XmlDocument();

xmlDoc.Load(filePath);

// Tạo phần tử mới cho hóa đơn

XmlElement hoaDonElement = xmlDoc.CreateElement("HoaDon");

XmlElement maHDElement = xmlDoc.CreateElement("MaHD");

maHDElement.InnerText = maHD;

hoaDonElement.AppendChild(maHDElement);

XmlElement maNVElement = xmlDoc.CreateElement("MaNV");

maNVElement.InnerText = maNV;

hoaDonElement.AppendChild(maNVElement);

XmlElement ngayLapElement = xmlDoc.CreateElement("NgayLap");

ngayLapElement.InnerText = ngayLap.ToString("yyyy-MM-dd HH:mm:ss");

hoaDonElement.AppendChild(ngayLapElement);

// Thêm phần tử mới vào trong root

xmlDoc.DocumentElement.AppendChild(hoaDonElement);

// Lưu lại file XML

xmlDoc.Save(filePath);

}

private void SaveCTHoaDonToXml(string maHD, string maSP, int slMua, decimal thanhTien)

{

string filePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\CTHoaDon.xml"; // Đường dẫn file CTHoaDons.xml

// Nếu file CTHoaDons.xml không tồn tại, tạo mới

// Load file XML

XmlDocument xmlDoc = new XmlDocument();

xmlDoc.Load(filePath);

// Tạo phần tử mới cho chi tiết hóa đơn

XmlElement chiTietElement = xmlDoc.CreateElement("CTHoaDon");

XmlElement maHDElement = xmlDoc.CreateElement("MaHD");

maHDElement.InnerText = maHD;

chiTietElement.AppendChild(maHDElement);

XmlElement maSPEl = xmlDoc.CreateElement("MaSP");

maSPEl.InnerText = maSP;

chiTietElement.AppendChild(maSPEl);

XmlElement slMuaElement = xmlDoc.CreateElement("SLMua");

slMuaElement.InnerText = slMua.ToString();

chiTietElement.AppendChild(slMuaElement);

XmlElement thanhTienElement = xmlDoc.CreateElement("ThanhTien");

thanhTienElement.InnerText = thanhTien.ToString("F2");

chiTietElement.AppendChild(thanhTienElement);

// Thêm phần tử mới vào trong root

xmlDoc.DocumentElement.AppendChild(chiTietElement);

// Lưu lại file XML

xmlDoc.Save(filePath);

}

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

{

string filePath = "D:\\nam3-HK124\\Cong-nghe-XML-124CNX01\\BaiTapLon\\finalBTL\\ShopQuanAo\\ShopQuanAo\\ShopQuanAo\\bin\\Debug\\Data\\CTHoaDon.xml"; // Đường dẫn đến file XML

if (File.Exists(filePath))

{

// Đọc nội dung XML

string xmlContent = File.ReadAllText(filePath);

// Chuyển đổi XML thành HTML với kiểu dáng bảng

string htmlContent = ConvertXmlToHtml(xmlContent);

// Lưu nội dung HTML vào file

string htmlFilePath = Path.Combine(Application.StartupPath, "HoaDon.html");

File.WriteAllText(htmlFilePath, htmlContent);

// Mở file HTML trong trình duyệt mặc định

System.Diagnostics.Process.Start(htmlFilePath);

}

else

{

MessageBox.Show("Không tìm thấy file XML!");

}

}

private string ConvertXmlToHtml(string xmlContent)

{

// Tạo HTML cơ bản với CSS

StringBuilder html = new StringBuilder();

html.AppendLine("<html><head><title>Hóa Đơn</title>");

html.AppendLine("<style>");

html.AppendLine("body { font-family: Arial, sans-serif; margin: 20px; background-color: #f4f4f9; }");

html.AppendLine("h1 { text-align: center; color: #333; }");

html.AppendLine("table { width: 100%; border-collapse: collapse; margin-top: 20px; }");

html.AppendLine("th, td { padding: 8px; text-align: left; border: 1px solid #ddd; }");

html.AppendLine("th { background-color: #4CAF50; color: white; }");

html.AppendLine("tr:nth-child(even) { background-color: #f2f2f2; }");

html.AppendLine("tr:hover { background-color: #ddd; }");

html.AppendLine("</style>");

html.AppendLine("</head><body>");

html.AppendLine("<h1>Danh Sách Hóa Đơn</h1>");

html.AppendLine("<table><tr><th>Mã Hóa Đơn</th><th>Mã Sản Phẩm</th><th>Số Lượng</th><th>Thành Tiền</th></tr>");

// Dùng XmlDocument để phân tích cú pháp XML

XmlDocument xmlDoc = new XmlDocument();

xmlDoc.LoadXml(xmlContent);

// Duyệt qua các phần tử <CTHoaDon> trong XML

XmlNodeList nodes = xmlDoc.SelectNodes("//CTHoaDon");

foreach (XmlNode node in nodes)

{

string maHD = node["MaHD"]?.InnerText ?? "";

string maSP = node["MaSP"]?.InnerText ?? "";

string slMua = node["SLMua"]?.InnerText ?? "";

string thanhTien = node["ThanhTien"]?.InnerText ?? "";

// Thêm dòng dữ liệu vào bảng HTML

html.AppendLine($"<tr><td>{maHD}</td><td>{maSP}</td><td>{slMua}</td><td>{thanhTien}</td></tr>");

}

html.AppendLine("</table></body></html>");

return html.ToString();

}

}

}

1. **Form quản lý nhân viên**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Xml.Linq;

namespace ShopQuanAo

{

public partial class QLNV : Form

{

public QLNV()

{

InitializeComponent();

}

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

{

this.Close();

// Tạo instance của Form QLSP

QLHD qlhdForm = new QLHD();

// Hiển thị Form QLSP

qlhdForm.Show();

}

private void QLNV\_Load(object sender, EventArgs e)

{

// Đường dẫn tới file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\NhanVien.xml";

try

{

// Kiểm tra xem file XML có tồn tại không

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML

DataSet dataSet = new DataSet();

dataSet.ReadXml(xmlFilePath);

// Gắn dữ liệu vào DataGridView

if (dataSet.Tables.Count > 0)

{

dataGridView1.DataSource = dataSet.Tables[0];

}

else

{

MessageBox.Show("Không tìm thấy bảng dữ liệu trong file XML!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi đọc file XML: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

// Kiểm tra nếu click vào một ô hợp lệ (không phải header)

if (e.RowIndex >= 0)

{

try

{

// Lấy dữ liệu từ hàng được click

DataGridViewRow row = dataGridView1.Rows[e.RowIndex];

// Gán dữ liệu vào các TextBox nếu cột tồn tại và không null

if (dataGridView1.Columns.Contains("MaNV") && dataGridView1.Columns.Contains("HoTen"))

{

txtMaNv.Text = row.Cells["MaNV"].Value?.ToString() ?? "";

txtTenNv.Text = row.Cells["HoTen"].Value?.ToString() ?? "";

txtDC.Text = row.Cells["DiaChi"].Value?.ToString() ?? "";

txtSDTNV.Text = row.Cells["SDTNV"].Value?.ToString() ?? "";

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi lấy dữ liệu từ dòng: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void guna2Button1\_Click(object sender, EventArgs e)

{

// Lấy dữ liệu từ các TextBox

string maNV = txtMaNv.Text.Trim();

string hoTen = txtTenNv.Text.Trim();

string diaChi = txtDC.Text.Trim();

string sdtNV = txtSDTNV.Text.Trim();

// Kiểm tra dữ liệu nhập vào không để trống

if (string.IsNullOrEmpty(maNV) || string.IsNullOrEmpty(hoTen) || string.IsNullOrEmpty(diaChi) || string.IsNullOrEmpty(sdtNV))

{

MessageBox.Show("Vui lòng nhập đầy đủ thông tin nhân viên!", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Dừng lại nếu có thông tin bị thiếu

}

// Kiểm tra số điện thoại hợp lệ

if (!System.Text.RegularExpressions.Regex.IsMatch(sdtNV, @"^\d+$"))

{

MessageBox.Show("Số điện thoại chỉ được chứa các chữ số!", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Dừng lại nếu số điện thoại không hợp lệ

}

// Đường dẫn file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\NhanVien.xml";

try

{

// Kiểm tra nếu file XML đã tồn tại

XDocument doc;

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML hiện tại

doc = XDocument.Load(xmlFilePath);

// Kiểm tra mã nhân viên đã tồn tại

XElement existingEmployee = doc.Descendants("NhanVien")

.FirstOrDefault(x => x.Element("MaNV")?.Value == maNV);

if (existingEmployee != null)

{

MessageBox.Show("Mã nhân viên đã tồn tại! Vui lòng nhập mã khác.", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Ngừng thêm dữ liệu

}

}

else

{

// Tạo mới file XML nếu chưa tồn tại

doc = new XDocument(new XElement("NhanViens"));

}

// Thêm dữ liệu mới vào file XML

XElement nhanViensElement = doc.Element("NhanViens");

if (nhanViensElement != null)

{

XElement newEmployee = new XElement("NhanVien",

new XElement("MaNV", maNV),

new XElement("HoTen", hoTen),

new XElement("DiaChi", diaChi),

new XElement("SDTNV", sdtNV)

);

nhanViensElement.Add(newEmployee);

}

// Lưu file XML

doc.Save(xmlFilePath);

MessageBox.Show("Thêm dữ liệu vào file XML thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

// Tải lại dữ liệu và cập nhật SQL Server

QLNV\_Load(sender, e);

UpdateSqlFromXml();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi thêm dữ liệu vào file XML hoặc cập nhật SQL Server: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void txtMaNv\_TextChanged(object sender, EventArgs e)

{

}

private void txtTenNv\_TextChanged(object sender, EventArgs e)

{

}

private void txtDC\_TextChanged(object sender, EventArgs e)

{

}

private void txtSDTNV\_TextChanged(object sender, EventArgs e)

{

}

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

{

// Lấy thông tin từ các textbox

string maNV = txtMaNv.Text;

string hoTen = txtTenNv.Text;

string diaChi = txtDC.Text;

string sdtNV = txtSDTNV.Text;

// Đường dẫn đến file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\NhanVien.xml";

try

{

// Kiểm tra nếu file XML đã tồn tại

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML hiện tại

XDocument doc = XDocument.Load(xmlFilePath);

// Tìm phần tử NhanVien với MaNV tương ứng để sửa

XElement nhanVienToEdit = doc.Descendants("NhanVien")

.FirstOrDefault(x => x.Element("MaNV")?.Value == maNV);

if (nhanVienToEdit != null)

{

// Sửa dữ liệu của NhanVien

nhanVienToEdit.SetElementValue("HoTen", hoTen);

nhanVienToEdit.SetElementValue("DiaChi", diaChi);

nhanVienToEdit.SetElementValue("SDTNV", sdtNV);

// Lưu file XML sau khi sửa

doc.Save(xmlFilePath);

MessageBox.Show("Sửa dữ liệu trong file XML thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

QLNV\_Load(sender, e);

// Gọi hàm UpdateSqlFromXml để cập nhật SQL Server từ file XML

UpdateSqlFromXml();

}

else

{

MessageBox.Show("Không tìm thấy nhân viên với mã " + maNV, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi sửa dữ liệu trong file XML hoặc cập nhật SQL Server: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void UpdateSqlFromXml()

{

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\NhanVien.xml";

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

try

{

if (System.IO.File.Exists(xmlFilePath))

{

// Load dữ liệu từ file XML

XDocument doc = XDocument.Load(xmlFilePath);

XElement nhanViensElement = doc.Element("NhanViens");

if (nhanViensElement != null)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

foreach (XElement employee in nhanViensElement.Elements("NhanVien"))

{

string maNV = employee.Element("MaNV")?.Value;

string hoTen = employee.Element("HoTen")?.Value;

string diaChi = employee.Element("DiaChi")?.Value;

string sdtNV = employee.Element("SDTNV")?.Value;

// Câu lệnh SQL để thêm hoặc cập nhật dữ liệu

string query = "IF EXISTS (SELECT 1 FROM NhanVien WHERE MaNV = @MaNV) " +

"UPDATE NhanVien SET HoTen = @HoTen, DiaChi = @DiaChi, SDTNV = @SDTNV " +

"WHERE MaNV = @MaNV " +

"ELSE " +

"INSERT INTO NhanVien (MaNV, HoTen, DiaChi, SDTNV) " +

"VALUES (@MaNV, @HoTen, @DiaChi, @SDTNV)";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@MaNV", maNV);

command.Parameters.AddWithValue("@HoTen", hoTen);

command.Parameters.AddWithValue("@DiaChi", diaChi);

command.Parameters.AddWithValue("@SDTNV", sdtNV);

command.ExecuteNonQuery();

}

}

}

MessageBox.Show("Cập nhật SQL Server từ file XML thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("File XML không chứa dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi cập nhật SQL Server từ file XML: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

// Lấy mã nhân viên cần xóa từ textbox

string maNV = txtMaNv.Text;

// Đường dẫn đến file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\NhanVien.xml";

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

try

{

// Kiểm tra nếu file XML đã tồn tại

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML hiện tại

XDocument doc = XDocument.Load(xmlFilePath);

// Tìm phần tử NhanVien với MaNV tương ứng để xóa

XElement nhanVienToDelete = doc.Descendants("NhanVien")

.FirstOrDefault(x => x.Element("MaNV")?.Value == maNV);

if (nhanVienToDelete != null)

{

// Xóa phần tử NhanVien khỏi XML

nhanVienToDelete.Remove();

// Lưu file XML sau khi xóa

doc.Save(xmlFilePath);

// Cập nhật SQL Server - xóa nhân viên khỏi cơ sở dữ liệu

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string deleteQuery = "DELETE FROM NhanVien WHERE MaNV = @MaNV";

using (SqlCommand command = new SqlCommand(deleteQuery, connection))

{

command.Parameters.AddWithValue("@MaNV", maNV);

command.ExecuteNonQuery();

}

}

MessageBox.Show("Xóa dữ liệu trong file XML và cơ sở dữ liệu thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

QLNV\_Load(sender, e);

}

else

{

MessageBox.Show("Không tìm thấy nhân viên với mã " + maNV, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi xóa dữ liệu trong file XML hoặc cơ sở dữ liệu: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

// Đường dẫn tới file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\NhanVien.xml";

try

{

// Kiểm tra xem file XML có tồn tại không

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML

DataSet dataSet = new DataSet();

dataSet.ReadXml(xmlFilePath);

// Gắn dữ liệu vào DataGridView

if (dataSet.Tables.Count > 0)

{

dataGridView1.DataSource = dataSet.Tables[0];

}

else

{

MessageBox.Show("Không tìm thấy bảng dữ liệu trong file XML!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi đọc file XML: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

this.Close();

DangNhap dn = new DangNhap();

dn.Show();

}

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

{

this.Close();

// Tạo instance của Form QLSP

QLSP qlspForm = new QLSP();

// Hiển thị Form QLSP

qlspForm.Show();

}

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

{

this.Close();

// Tạo instance của Form QLTK

QLTK TKForm = new QLTK();

// Hiển thị Form QLTK

TKForm.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

textBoxSearch.Clear();

}

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

{

// Đường dẫn kết nối đến cơ sở dữ liệu SQL Server

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

// Lấy nội dung trong textBoxSearch và loại bỏ khoảng trắng thừa

string searchQuery = textBoxSearch.Text.Trim().ToLower(); // Chuyển sang chữ thường để tìm kiếm không phân biệt hoa-thường

// Kiểm tra nếu người dùng đã nhập từ khóa tìm kiếm

if (!string.IsNullOrEmpty(searchQuery))

{

try

{

// Câu lệnh SQL để tìm kiếm nhân viên có tên chứa từ khóa tìm kiếm

string sqlQuery = "SELECT MaNV, HoTen, DiaChi, SDTNV FROM NhanVien " +

"WHERE LOWER(HoTen) LIKE @searchQuery COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS"; // Tìm kiếm không phân biệt hoa-thường

// Tạo kết nối tới cơ sở dữ liệu SQL Server

using (SqlConnection conn = new SqlConnection(connectionString))

{

// Mở kết nối

conn.Open();

// Tạo câu lệnh SQL với tham số

using (SqlCommand cmd = new SqlCommand(sqlQuery, conn))

{

// Thêm tham số tìm kiếm, sử dụng '%' để tìm kiếm phần tử chứa từ khóa (cả trước và sau từ khóa)

cmd.Parameters.AddWithValue("@searchQuery", "%" + searchQuery + "%");

// Tạo DataTable để lưu kết quả

DataTable dt = new DataTable();

// Tạo DataAdapter để điền dữ liệu vào DataTable

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt); // Điền dữ liệu vào DataTable

}

// Kiểm tra nếu không có dữ liệu trả về

if (dt.Rows.Count == 0)

{

MessageBox.Show("Không tìm thấy nhân viên nào với từ khóa: " + searchQuery, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

// Gán DataTable vào DataGridView

dataGridView1.DataSource = dt;

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi tìm kiếm: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

MessageBox.Show("Vui lòng nhập tên nhân viên để tìm kiếm.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

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

{

}

}

}

1. **Form quản lý sản phẩm**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Xml.Linq;

namespace ShopQuanAo

{

public partial class QLSP : Form

{

public QLSP()

{

InitializeComponent();

}

private void QLSP\_Load(object sender, EventArgs e)

{

// Đường dẫn tới file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\SanPham.xml";

try

{

// Kiểm tra xem file XML có tồn tại không

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML

DataSet dataSet = new DataSet();

dataSet.ReadXml(xmlFilePath);

// Kiểm tra xem có bảng dữ liệu không

if (dataSet.Tables.Count > 0)

{

// Gắn dữ liệu vào DataGridView

dataGridView1.DataSource = dataSet.Tables[0];

}

else

{

MessageBox.Show("Không tìm thấy bảng dữ liệu trong file XML!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (System.IO.IOException ioEx)

{

MessageBox.Show("Lỗi khi đọc file XML: " + ioEx.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (Exception ex)

{

MessageBox.Show("Đã xảy ra lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (e.RowIndex >= 0) // Chỉ xử lý khi click vào một hàng hợp lệ

{

try

{

DataGridViewRow row = dataGridView1.Rows[e.RowIndex];

// Gán dữ liệu vào các TextBox

textBoxMaSP.Text = row.Cells["MaSP"].Value?.ToString() ?? "";

textBoxTenSP.Text = row.Cells["TenSP"].Value?.ToString() ?? "";

textBoxGiaSP.Text = row.Cells["Gia"].Value?.ToString() ?? "";

textBoxLoaiSP.Text = row.Cells["Loai"].Value?.ToString() ?? "";

textBoxXuatXuSP.Text = row.Cells["XuatXu"].Value?.ToString() ?? "";

textBoxSLSP.Text = row.Cells["SoLuong"].Value?.ToString() ?? "";

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi lấy dữ liệu từ dòng: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

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

{

this.Close();

// Tạo instance của Form QLNV

QLNV qlnvForm = new QLNV();

// Hiển thị Form QLNV

qlnvForm.Show();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void txtTenNv\_TextChanged(object sender, EventArgs e)

{

}

private void label6\_Click(object sender, EventArgs e)

{

}

private void UpdateSqlFromXml()

{

// Đường dẫn đến file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\SanPham.xml";

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

try

{

// Kiểm tra sự tồn tại của file XML

if (System.IO.File.Exists(xmlFilePath))

{

// Tải dữ liệu từ file XML

XDocument doc = XDocument.Load(xmlFilePath);

// Lặp qua tất cả các sản phẩm trong XML và thực hiện thao tác chèn hoặc cập nhật vào SQL Server

foreach (XElement productElement in doc.Descendants("SanPham"))

{

// Lấy dữ liệu từ từng phần tử sản phẩm trong XML

string maSP = productElement.Element("MaSP")?.Value;

string tenSP = productElement.Element("TenSP")?.Value;

string giaSP = productElement.Element("Gia")?.Value;

string loaiSP = productElement.Element("Loai")?.Value;

string xuatXu = productElement.Element("XuatXu")?.Value;

string soLuong = productElement.Element("SoLuong")?.Value;

// Kiểm tra dữ liệu sản phẩm hợp lệ

if (!string.IsNullOrEmpty(maSP) && !string.IsNullOrEmpty(tenSP) && !string.IsNullOrEmpty(giaSP) &&

!string.IsNullOrEmpty(loaiSP) && !string.IsNullOrEmpty(xuatXu) && !string.IsNullOrEmpty(soLuong))

{

// Kiểm tra tính hợp lệ của giá và số lượng

if (decimal.TryParse(giaSP, out decimal gia) && int.TryParse(soLuong, out int soLuongInt))

{

// Câu lệnh SQL để chèn hoặc cập nhật dữ liệu vào bảng SanPham

string sqlQuery = "IF EXISTS (SELECT 1 FROM SanPham WHERE MaSP = @MaSP) " +

"UPDATE SanPham SET TenSP = @TenSP, Gia = @Gia, Loai = @LoaiSP, XuatXu = @XuatXu, SoLuong = @SoLuong " +

"WHERE MaSP = @MaSP " +

"ELSE " +

"INSERT INTO SanPham (MaSP, TenSP, Gia, Loai, XuatXu, SoLuong) " +

"VALUES (@MaSP, @TenSP, @Gia, @LoaiSP, @XuatXu, @SoLuong)";

// Tạo kết nối đến SQL Server và thực thi câu lệnh SQL

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

using (SqlCommand cmd = new SqlCommand(sqlQuery, conn))

{

cmd.Parameters.AddWithValue("@MaSP", maSP);

cmd.Parameters.AddWithValue("@TenSP", tenSP);

cmd.Parameters.AddWithValue("@Gia", gia);

cmd.Parameters.AddWithValue("@LoaiSP", loaiSP);

cmd.Parameters.AddWithValue("@XuatXu", xuatXu);

cmd.Parameters.AddWithValue("@SoLuong", soLuongInt);

// Thực thi câu lệnh SQL

cmd.ExecuteNonQuery();

}

}

}

else

{

MessageBox.Show($"Dữ liệu không hợp lệ cho sản phẩm {maSP}. Giá hoặc số lượng không đúng.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

MessageBox.Show("Dữ liệu đã được cập nhật thành công vào cơ sở dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("File XML không tồn tại!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi cập nhật dữ liệu vào cơ sở dữ liệu SQL Server: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

// Lấy dữ liệu từ các textbox

string maSP = textBoxMaSP.Text.Trim();

string tenSP = textBoxTenSP.Text.Trim();

string giaSP = textBoxGiaSP.Text.Trim();

string loaiSP = textBoxLoaiSP.Text.Trim();

string xuatXu = textBoxXuatXuSP.Text.Trim();

string soLuong = textBoxSLSP.Text.Trim();

// Kiểm tra các trường có bị bỏ trống không

if (string.IsNullOrEmpty(maSP) || string.IsNullOrEmpty(tenSP) || string.IsNullOrEmpty(giaSP) ||

string.IsNullOrEmpty(loaiSP) || string.IsNullOrEmpty(xuatXu) || string.IsNullOrEmpty(soLuong))

{

MessageBox.Show("Vui lòng điền đầy đủ tất cả các trường!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Dừng lại nếu có trường trống

}

// Kiểm tra giá sản phẩm (phải là số và lớn hơn 0)

if (!decimal.TryParse(giaSP, out decimal gia) || gia <= 0)

{

MessageBox.Show("Giá sản phẩm phải là số dương!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Kiểm tra số lượng sản phẩm (phải là số nguyên và lớn hơn hoặc bằng 0)

if (!int.TryParse(soLuong, out int soLuongSP) || soLuongSP < 0)

{

MessageBox.Show("Số lượng sản phẩm phải là số nguyên và không âm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Đường dẫn file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\SanPham.xml";

try

{

// Kiểm tra nếu file XML đã tồn tại

XDocument doc;

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML hiện tại

doc = XDocument.Load(xmlFilePath);

// Kiểm tra mã sản phẩm đã tồn tại

XElement existingProduct = doc.Descendants("SanPham")

.FirstOrDefault(x => x.Element("MaSP")?.Value == maSP);

if (existingProduct != null)

{

MessageBox.Show("Mã sản phẩm đã tồn tại! Vui lòng nhập mã khác.", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Ngừng thêm dữ liệu

}

}

else

{

// Tạo mới file XML nếu chưa tồn tại

doc = new XDocument(new XElement("SanPhams"));

}

// Thêm dữ liệu mới vào file XML

XElement sanPhamsElement = doc.Element("SanPhams");

if (sanPhamsElement != null)

{

XElement newProduct = new XElement("SanPham",

new XElement("MaSP", maSP),

new XElement("TenSP", tenSP),

new XElement("Gia", giaSP),

new XElement("Loai", loaiSP),

new XElement("XuatXu", xuatXu),

new XElement("SoLuong", soLuong)

);

sanPhamsElement.Add(newProduct);

}

// Lưu file XML

doc.Save(xmlFilePath);

MessageBox.Show("Thêm dữ liệu vào file XML thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

// Gọi lại hàm load lại dữ liệu hoặc cập nhật UI nếu cần

QLSP\_Load(sender, e); // Ví dụ nếu có hàm load lại dữ liệu sản phẩm

// Gọi hàm UpdateSqlFromXml để cập nhật SQL Server từ file XML

UpdateSqlFromXml();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi thêm dữ liệu vào file XML hoặc cập nhật SQL Server: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

// Lấy dữ liệu từ các textbox

string maSP = textBoxMaSP.Text.Trim();

string tenSP = textBoxTenSP.Text.Trim();

string giaSP = textBoxGiaSP.Text.Trim();

string loaiSP = textBoxLoaiSP.Text.Trim();

string xuatXu = textBoxXuatXuSP.Text.Trim();

string soLuong = textBoxSLSP.Text.Trim();

// Kiểm tra các trường có bị bỏ trống không

if (string.IsNullOrEmpty(maSP) || string.IsNullOrEmpty(tenSP) || string.IsNullOrEmpty(giaSP) ||

string.IsNullOrEmpty(loaiSP) || string.IsNullOrEmpty(xuatXu) || string.IsNullOrEmpty(soLuong))

{

MessageBox.Show("Vui lòng điền đầy đủ tất cả các trường!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Dừng lại nếu có trường trống

}

// Đường dẫn đến file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\SanPham.xml";

try

{

// Kiểm tra nếu file XML đã tồn tại

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML hiện tại

XDocument doc = XDocument.Load(xmlFilePath);

// Tìm phần tử Sản Phẩm với MaSP tương ứng để sửa

XElement sanPhamToEdit = doc.Descendants("SanPham")

.FirstOrDefault(x => x.Element("MaSP")?.Value == maSP);

if (sanPhamToEdit != null)

{

// Sửa dữ liệu của Sản Phẩm

sanPhamToEdit.SetElementValue("TenSP", tenSP);

sanPhamToEdit.SetElementValue("Gia", giaSP);

sanPhamToEdit.SetElementValue("Loai", loaiSP);

sanPhamToEdit.SetElementValue("XuatXu", xuatXu);

sanPhamToEdit.SetElementValue("SoLuong", soLuong);

// Lưu file XML sau khi sửa

doc.Save(xmlFilePath);

MessageBox.Show("Sửa dữ liệu trong file XML thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

QLSP\_Load(sender, e);

// Gọi hàm UpdateSqlFromXml để cập nhật SQL Server từ file XML

UpdateSqlFromXml();

}

else

{

MessageBox.Show("Không tìm thấy sản phẩm với mã " + maSP, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi sửa dữ liệu trong file XML hoặc cập nhật SQL Server: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

// Lấy mã sản phẩm cần xóa từ textbox

string maSP = textBoxMaSP.Text.Trim();

// Đường dẫn đến file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\SanPham.xml";

// Kết nối với cơ sở dữ liệu SQL Server

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

try

{

// Kiểm tra nếu file XML đã tồn tại

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML hiện tại

XDocument doc = XDocument.Load(xmlFilePath);

// Tìm phần tử sản phẩm với MaSP tương ứng để xóa

XElement sanPhamToDelete = doc.Descendants("SanPham")

.FirstOrDefault(x => x.Element("MaSP")?.Value == maSP);

if (sanPhamToDelete != null)

{

// Xóa phần tử sản phẩm khỏi XML

sanPhamToDelete.Remove();

// Lưu lại file XML sau khi xóa

doc.Save(xmlFilePath);

// Cập nhật SQL Server - xóa sản phẩm khỏi cơ sở dữ liệu

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string deleteQuery = "DELETE FROM SanPham WHERE MaSP = @MaSP";

using (SqlCommand command = new SqlCommand(deleteQuery, connection))

{

command.Parameters.AddWithValue("@MaSP", maSP);

command.ExecuteNonQuery();

}

}

MessageBox.Show("Xóa sản phẩm thành công từ file XML và cơ sở dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

QLSP\_Load(sender, e); // Cập nhật lại danh sách sản phẩm trên giao diện

}

else

{

MessageBox.Show("Không tìm thấy sản phẩm với mã " + maSP, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi xóa dữ liệu trong file XML hoặc cơ sở dữ liệu: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

// Đường dẫn tới file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\SanPham.xml";

try

{

// Kiểm tra xem file XML có tồn tại không

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML

DataSet dataSet = new DataSet();

dataSet.ReadXml(xmlFilePath);

// Gắn dữ liệu vào DataGridView

if (dataSet.Tables.Count > 0)

{

dataGridView1.DataSource = dataSet.Tables[0];

}

else

{

MessageBox.Show("Không tìm thấy bảng dữ liệu trong file XML!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi đọc file XML: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

this.Close();

DangNhap dn = new DangNhap();

dn.Show();

}

private void label7\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

textBoxSearch.Clear();

}

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

{

// Đường dẫn kết nối đến cơ sở dữ liệu SQL Server

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

// Lấy nội dung trong textBoxSearch và loại bỏ khoảng trắng thừa

string searchQuery = textBoxSearch.Text.Trim().ToLower(); // Chuyển sang chữ thường để tìm kiếm không phân biệt hoa-thường

// Kiểm tra nếu người dùng đã nhập từ khóa tìm kiếm

if (!string.IsNullOrEmpty(searchQuery))

{

try

{

// Câu lệnh SQL để tìm kiếm sản phẩm có tên chứa từ khóa tìm kiếm

string sqlQuery = "SELECT MaSP, TenSP, Loai, XuatXu, SoLuong, Gia FROM SanPham " +

"WHERE LOWER(TenSP) LIKE @searchQuery COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS"; // Tìm kiếm không phân biệt hoa-thường

// Tạo kết nối tới cơ sở dữ liệu SQL Server

using (SqlConnection conn = new SqlConnection(connectionString))

{

// Mở kết nối

conn.Open();

// Tạo câu lệnh SQL với tham số

using (SqlCommand cmd = new SqlCommand(sqlQuery, conn))

{

// Thêm tham số tìm kiếm, sử dụng '%' để tìm kiếm phần tử chứa từ khóa (cả trước và sau từ khóa)

cmd.Parameters.AddWithValue("@searchQuery", "%" + searchQuery + "%");

// Tạo DataTable để lưu kết quả

DataTable dt = new DataTable();

// Tạo DataAdapter để điền dữ liệu vào DataTable

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt); // Điền dữ liệu vào DataTable

}

// Kiểm tra nếu không có dữ liệu trả về

if (dt.Rows.Count == 0)

{

MessageBox.Show("Không tìm thấy sản phẩm nào với từ khóa: " + searchQuery, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

// Gán DataTable vào DataGridView

dataGridView1.DataSource = dt;

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi tìm kiếm: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

MessageBox.Show("Vui lòng nhập tên sản phẩm để tìm kiếm.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

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

{

}

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

{

this.Close();

// Tạo instance của Form QLTK

QLTK TKForm = new QLTK();

// Hiển thị Form QLTK

TKForm.Show();

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

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

{

this.Close();

// Tạo instance của Form QLSP

QLHD qlhdForm = new QLHD();

// Hiển thị Form QLSP

qlhdForm.Show();

}

}

}

1. **Form quản lý tài khoản**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Xml.Linq;

namespace ShopQuanAo

{

public partial class QLTK : Form

{

public QLTK()

{

InitializeComponent();

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (e.RowIndex >= 0) // Chỉ xử lý khi click vào một hàng hợp lệ

{

try

{

DataGridViewRow row = dataGridView1.Rows[e.RowIndex];

// Kiểm tra xem các cột có tồn tại trước khi gán dữ liệu

if (dataGridView1.Columns.Contains("TenDangNhap"))

textBoxTK.Text = row.Cells["TenDangNhap"].Value?.ToString() ?? ""; // Tên đăng nhập

if (dataGridView1.Columns.Contains("MatKhau"))

textBoxMK.Text = row.Cells["MatKhau"].Value?.ToString() ?? ""; // Mật khẩu

if (dataGridView1.Columns.Contains("Quyen"))

comboBox1.SelectedItem = row.Cells["Quyen"].Value?.ToString() ?? ""; // Quyền

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi lấy dữ liệu từ dòng: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

textBoxSearch.Clear();

}

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

{

// Đường dẫn kết nối đến cơ sở dữ liệu SQL Server

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

// Lấy nội dung trong textBoxSearch và loại bỏ khoảng trắng thừa

string searchQuery = textBoxSearch.Text.Trim().ToLower(); // Chuyển sang chữ thường để tìm kiếm không phân biệt hoa-thường

// Kiểm tra nếu người dùng đã nhập từ khóa tìm kiếm

if (!string.IsNullOrEmpty(searchQuery))

{

try

{

// Câu lệnh SQL để tìm kiếm sản phẩm có tên chứa từ khóa tìm kiếm

string sqlQuery = "SELECT TenDangNhap, MatKhau, Quyen FROM TaiKhoan " +

"WHERE LOWER(TenDangNhap) LIKE @searchQuery COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS"; // Tìm kiếm không phân biệt hoa-thường

// Tạo kết nối tới cơ sở dữ liệu SQL Server

using (SqlConnection conn = new SqlConnection(connectionString))

{

// Mở kết nối

conn.Open();

// Tạo câu lệnh SQL với tham số

using (SqlCommand cmd = new SqlCommand(sqlQuery, conn))

{

// Thêm tham số tìm kiếm, sử dụng '%' để tìm kiếm phần tử chứa từ khóa (cả trước và sau từ khóa)

cmd.Parameters.AddWithValue("@searchQuery", "%" + searchQuery + "%");

// Tạo DataTable để lưu kết quả

DataTable dt = new DataTable();

// Tạo DataAdapter để điền dữ liệu vào DataTable

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

da.Fill(dt); // Điền dữ liệu vào DataTable

}

// Kiểm tra nếu không có dữ liệu trả về

if (dt.Rows.Count == 0)

{

MessageBox.Show("Không tìm thấy sản phẩm nào với từ khóa: " + searchQuery, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

// Gán DataTable vào DataGridView

dataGridView1.DataSource = dt;

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi tìm kiếm: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

MessageBox.Show("Vui lòng nhập tên đăng nhập để tìm kiếm.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

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

{

this.Close();

// Tạo instance của Form QLNV

QLNV qlnvForm = new QLNV();

// Hiển thị Form QLNV

qlnvForm.Show();

}

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

{

this.Close();

DangNhap dn = new DangNhap();

dn.Show();

}

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

{

this.Close();

// Tạo instance của Form QLSP

QLSP qlspForm = new QLSP();

// Hiển thị Form QLSP

qlspForm.Show();

}

private void QLTK\_Load(object sender, EventArgs e)

{

// Đường dẫn tới file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\TaiKhoan.xml";

try

{

// Kiểm tra xem file XML có tồn tại không

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML

DataSet dataSet = new DataSet();

dataSet.ReadXml(xmlFilePath);

// Kiểm tra xem có bảng dữ liệu không

if (dataSet.Tables.Count > 0)

{

// Đảm bảo bảng có các cột đúng theo cấu trúc

DataTable table = dataSet.Tables[0];

if (table.Columns.Contains("TenDangNhap") &&

table.Columns.Contains("MatKhau") &&

table.Columns.Contains("Quyen"))

{

// Gắn dữ liệu vào DataGridView

dataGridView1.DataSource = table;

}

else

{

MessageBox.Show("File XML không khớp với cấu trúc dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("Không tìm thấy bảng dữ liệu trong file XML!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (System.IO.IOException ioEx)

{

MessageBox.Show("Lỗi khi đọc file XML: " + ioEx.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (Exception ex)

{

MessageBox.Show("Đã xảy ra lỗi: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

// Cập nhật combobox Quyền

comboBox1.Items.Clear();

comboBox1.Items.Add(""); // Mặc định để trống

comboBox1.Items.Add("Admin");

comboBox1.Items.Add("User");

comboBox1.SelectedIndex = 0; // Mặc định chọn giá trị trống

}

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

{

QLTK\_Load(sender, e);

}

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

{

// Lấy tên tài khoản cần xóa từ textbox

string taiKhoan = textBoxTK.Text.Trim();

// Đường dẫn đến file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\TaiKhoan.xml";

// Kết nối với cơ sở dữ liệu SQL Server

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

try

{

// Kiểm tra nếu file XML đã tồn tại

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML hiện tại

XDocument doc = XDocument.Load(xmlFilePath);

// Tìm phần tử tài khoản với TenDangNhap tương ứng để xóa

XElement taiKhoanToDelete = doc.Descendants("TaiKhoan")

.FirstOrDefault(x => x.Element("TenDangNhap")?.Value == taiKhoan);

if (taiKhoanToDelete != null)

{

// Kiểm tra quyền của tài khoản cần xóa

string quyen = taiKhoanToDelete.Element("Quyen")?.Value;

// Kiểm tra nếu tài khoản cần xóa là Admin, đảm bảo còn ít nhất 1 tài khoản Admin

if (quyen == "Admin")

{

// Kiểm tra số lượng tài khoản Admin còn lại trong XML

int adminCount = doc.Descendants("TaiKhoan")

.Count(x => x.Element("Quyen")?.Value == "Admin");

// Nếu chỉ còn 1 tài khoản Admin, không cho phép xóa

if (adminCount <= 1)

{

MessageBox.Show("Không thể xóa tài khoản Admin vì hệ thống yêu cầu ít nhất 1 tài khoản Admin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

}

// Xóa phần tử tài khoản khỏi XML

taiKhoanToDelete.Remove();

// Lưu lại file XML sau khi xóa

doc.Save(xmlFilePath);

// Cập nhật SQL Server - xóa tài khoản khỏi cơ sở dữ liệu

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string deleteQuery = "DELETE FROM TaiKhoan WHERE TenDangNhap = @TaiKhoan";

using (SqlCommand command = new SqlCommand(deleteQuery, connection))

{

command.Parameters.AddWithValue("@TaiKhoan", taiKhoan);

command.ExecuteNonQuery();

}

}

MessageBox.Show("Xóa tài khoản thành công từ file XML và cơ sở dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

QLTK\_Load(sender, e); // Cập nhật lại danh sách tài khoản trên giao diện

}

else

{

MessageBox.Show("Không tìm thấy tài khoản với tên " + taiKhoan, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi xóa dữ liệu trong file XML hoặc cơ sở dữ liệu: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void UpdateSqlFromXml()

{

// Đường dẫn đến file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\TaiKhoan.xml";

string connectionString = @"Data Source=LAPTOP-V3R09SE1;Initial Catalog=ShopThoiTrang1;Integrated Security=True";

try

{

// Kiểm tra sự tồn tại của file XML

if (System.IO.File.Exists(xmlFilePath))

{

// Tải dữ liệu từ file XML

XDocument doc = XDocument.Load(xmlFilePath);

// Lặp qua tất cả các tài khoản trong XML và thực hiện thao tác chèn hoặc cập nhật vào SQL Server

foreach (XElement accountElement in doc.Descendants("TaiKhoan"))

{

// Lấy dữ liệu từ từng phần tử tài khoản trong XML

string tenDangNhap = accountElement.Element("TenDangNhap")?.Value;

string matKhau = accountElement.Element("MatKhau")?.Value;

string quyen = accountElement.Element("Quyen")?.Value;

// Kiểm tra dữ liệu tài khoản hợp lệ

if (!string.IsNullOrEmpty(tenDangNhap) && !string.IsNullOrEmpty(matKhau) && !string.IsNullOrEmpty(quyen))

{

// Câu lệnh SQL để chèn hoặc cập nhật dữ liệu vào bảng TaiKhoan

string sqlQuery = "IF EXISTS (SELECT 1 FROM TaiKhoan WHERE TenDangNhap = @TenDangNhap) " +

"UPDATE TaiKhoan SET MatKhau = @MatKhau, Quyen = @Quyen " +

"WHERE TenDangNhap = @TenDangNhap " +

"ELSE " +

"INSERT INTO TaiKhoan (TenDangNhap, MatKhau, Quyen) " +

"VALUES (@TenDangNhap, @MatKhau, @Quyen)";

// Tạo kết nối đến SQL Server và thực thi câu lệnh SQL

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

using (SqlCommand cmd = new SqlCommand(sqlQuery, conn))

{

cmd.Parameters.AddWithValue("@TenDangNhap", tenDangNhap);

cmd.Parameters.AddWithValue("@MatKhau", matKhau);

cmd.Parameters.AddWithValue("@Quyen", quyen);

// Thực thi câu lệnh SQL

cmd.ExecuteNonQuery();

}

}

}

else

{

MessageBox.Show($"Dữ liệu không hợp lệ cho tài khoản {tenDangNhap}. Vui lòng kiểm tra lại.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

MessageBox.Show("Dữ liệu tài khoản đã được cập nhật thành công vào cơ sở dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("File XML không tồn tại!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi cập nhật dữ liệu tài khoản vào cơ sở dữ liệu SQL Server: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

// Lấy dữ liệu từ các textbox và combobox

string tenDN = textBoxTK.Text.Trim();

string matKhau = textBoxMK.Text.Trim();

string quyen = comboBox1.Text.Trim();

// Kiểm tra các trường có bị bỏ trống không

if (string.IsNullOrEmpty(tenDN) || string.IsNullOrEmpty(matKhau) || string.IsNullOrEmpty(quyen))

{

MessageBox.Show("Vui lòng bổ sung đủ thông tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Dừng lại nếu có trường trống

}

// Đường dẫn file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\TaiKhoan.xml";

try

{

// Tạo biến tài liệu XML

XDocument doc;

// Kiểm tra nếu file XML đã tồn tại

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML hiện tại

doc = XDocument.Load(xmlFilePath);

// Kiểm tra nếu tên đăng nhập đã tồn tại

XElement existingAccount = doc.Descendants("TaiKhoan")

.FirstOrDefault(x => x.Element("TenDangNhap")?.Value == tenDN);

if (existingAccount != null)

{

MessageBox.Show("Tên đăng nhập đã tồn tại! Vui lòng nhập lại.", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Ngừng thêm dữ liệu

}

}

else

{

// Tạo mới file XML nếu chưa tồn tại

doc = new XDocument(new XElement("TaiKhoans"));

}

// Thêm tài khoản mới vào file XML

XElement newAccount = new XElement("TaiKhoan",

new XElement("TenDangNhap", tenDN),

new XElement("MatKhau", matKhau),

new XElement("Quyen", quyen)

);

doc.Element("TaiKhoans")?.Add(newAccount);

// Lưu file XML

doc.Save(xmlFilePath);

MessageBox.Show("Thêm tài khoản thành công vào file XML!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

// Gọi lại hàm load lại dữ liệu trên giao diện

btnLoad\_Click(sender, e);

// Gọi hàm cập nhật SQL Server từ file XML nếu cần

UpdateSqlFromXml();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi thêm tài khoản vào file XML: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

// Lấy dữ liệu từ các textbox

string tenDangNhap = textBoxTK.Text.Trim();

string matKhau = textBoxMK.Text.Trim();

string quyen = comboBox1.Text.Trim();

// Kiểm tra các trường có bị bỏ trống không

if (string.IsNullOrEmpty(tenDangNhap) || string.IsNullOrEmpty(matKhau) || string.IsNullOrEmpty(quyen))

{

MessageBox.Show("Vui lòng điền đầy đủ tất cả các trường!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Dừng lại nếu có trường trống

}

// Đường dẫn đến file XML

string xmlFilePath = @"D:\nam3-HK124\Cong-nghe-XML-124CNX01\BaiTapLon\finalBTL\ShopQuanAo\ShopQuanAo\ShopQuanAo\bin\Debug\Data\TaiKhoan.xml";

try

{

// Kiểm tra nếu file XML đã tồn tại

if (System.IO.File.Exists(xmlFilePath))

{

// Load file XML hiện tại

XDocument doc = XDocument.Load(xmlFilePath);

// Tìm phần tử Tài Khoản với TenDangNhap tương ứng để sửa

XElement taiKhoanToEdit = doc.Descendants("TaiKhoan")

.FirstOrDefault(x => x.Element("TenDangNhap")?.Value == tenDangNhap);

if (taiKhoanToEdit != null)

{

// Sửa dữ liệu của Tài Khoản

taiKhoanToEdit.SetElementValue("MatKhau", matKhau);

taiKhoanToEdit.SetElementValue("Quyen", quyen);

// Lưu file XML sau khi sửa

doc.Save(xmlFilePath);

MessageBox.Show("Sửa dữ liệu trong file XML thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

QLTK\_Load(sender, e);

// Gọi hàm UpdateSqlFromXmlForTaiKhoan để cập nhật SQL Server từ file XML

UpdateSqlFromXml();

}

else

{

MessageBox.Show("Không tìm thấy tài khoản với tên đăng nhập " + tenDangNhap, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("File XML không tồn tại!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi sửa dữ liệu trong file XML hoặc cập nhật SQL Server: " + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

this.Close();

// Tạo instance của Form QLSP

QLHD qlhdForm = new QLHD();

// Hiển thị Form QLSP

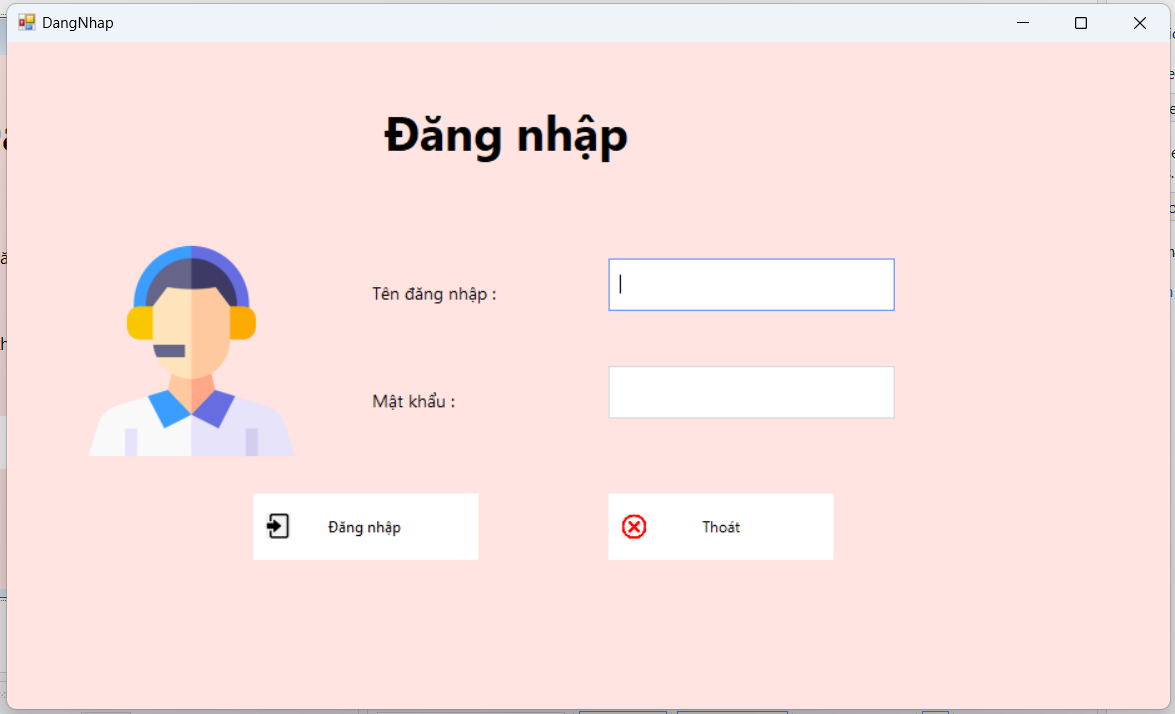
qlhdForm.Show();

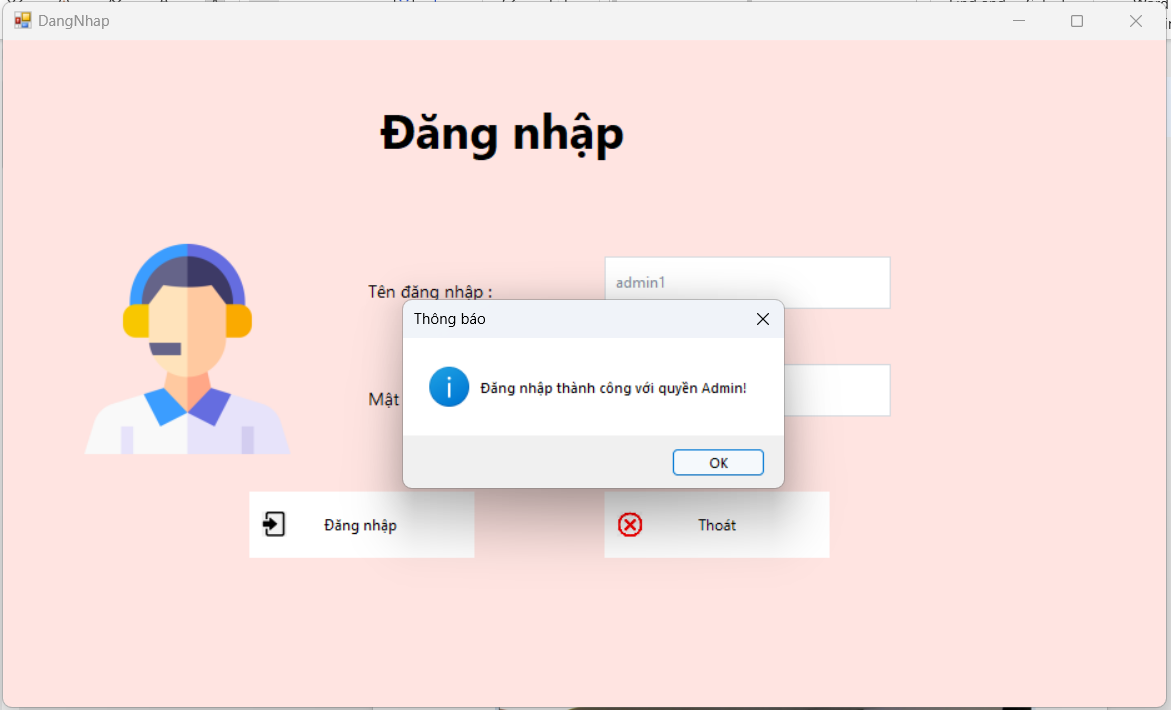
}

}

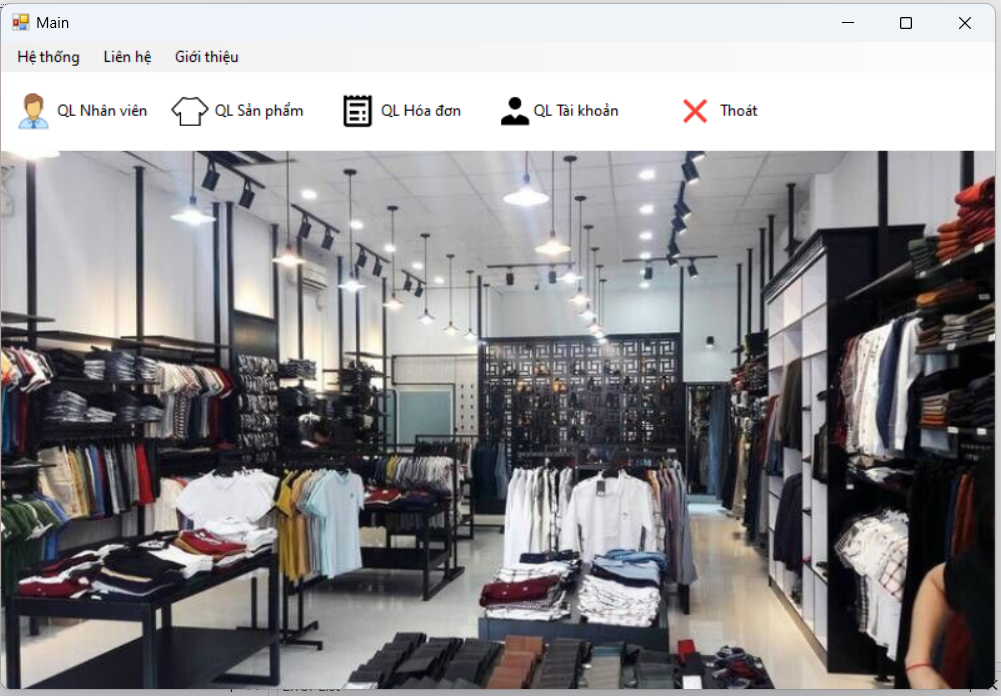
}

1. **Giao diện chương trình**

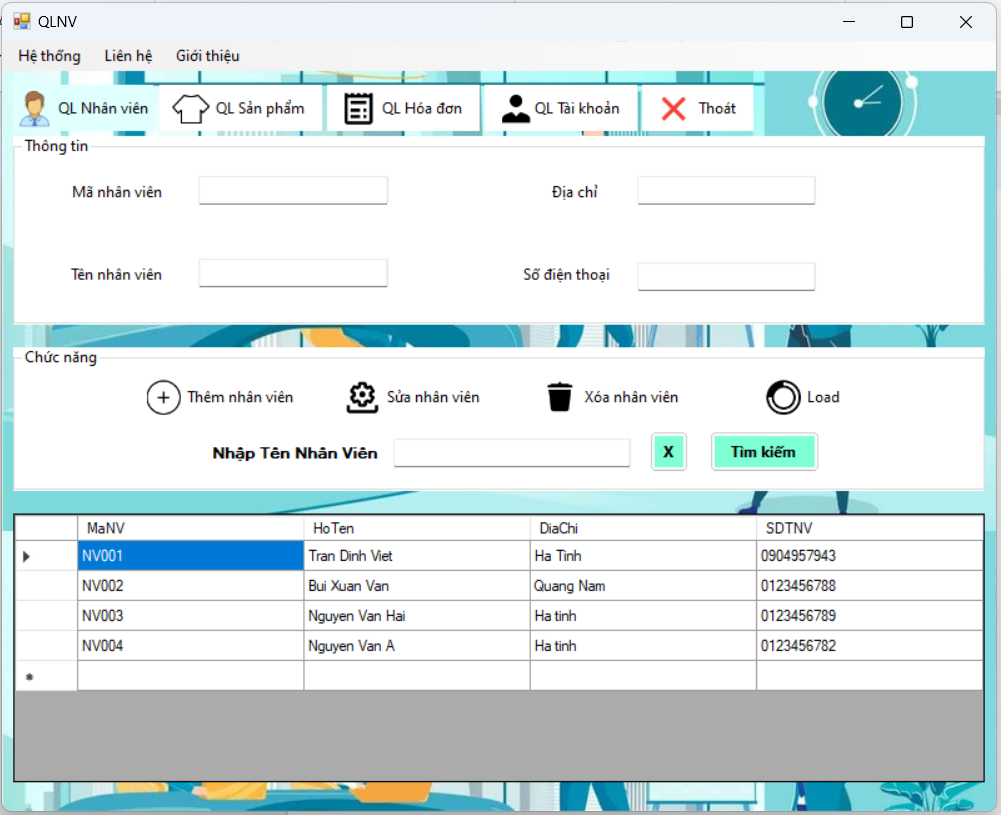




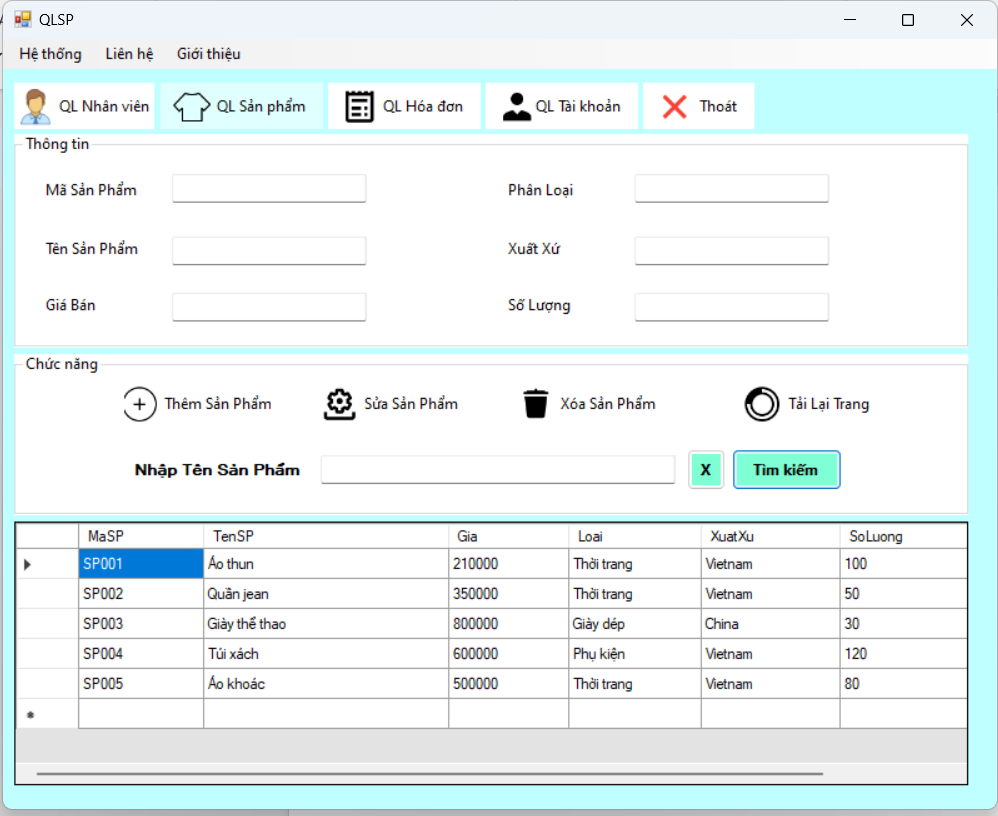
Form đăng nhập



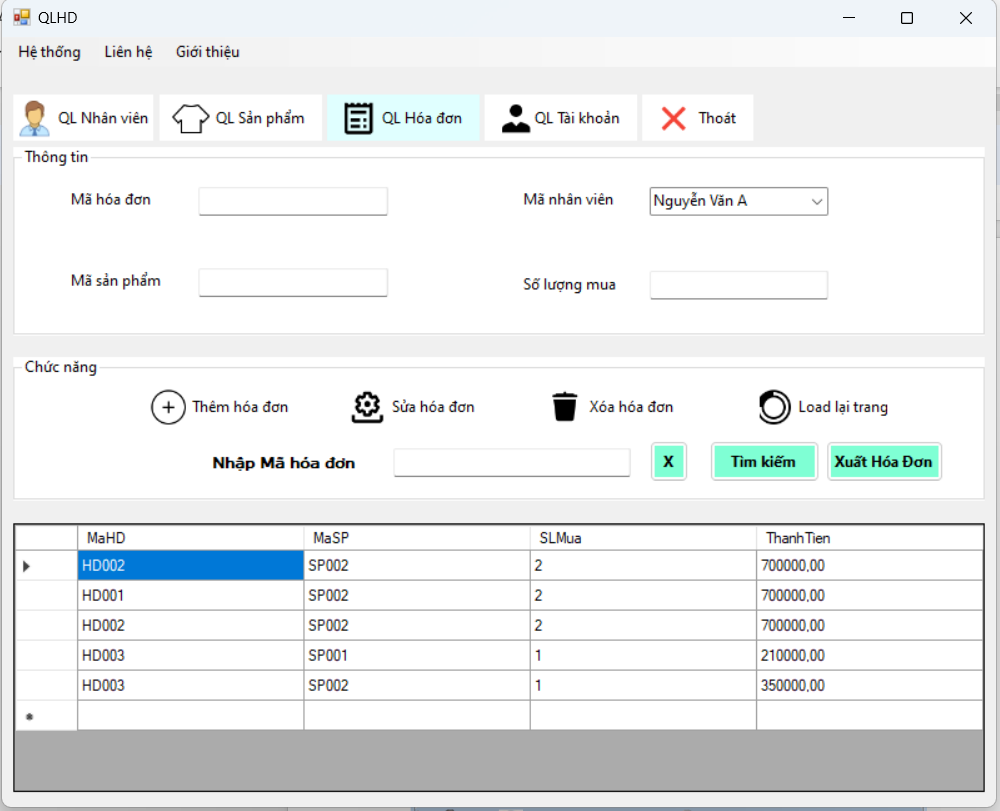
Form main



Form quản lý nhân viên



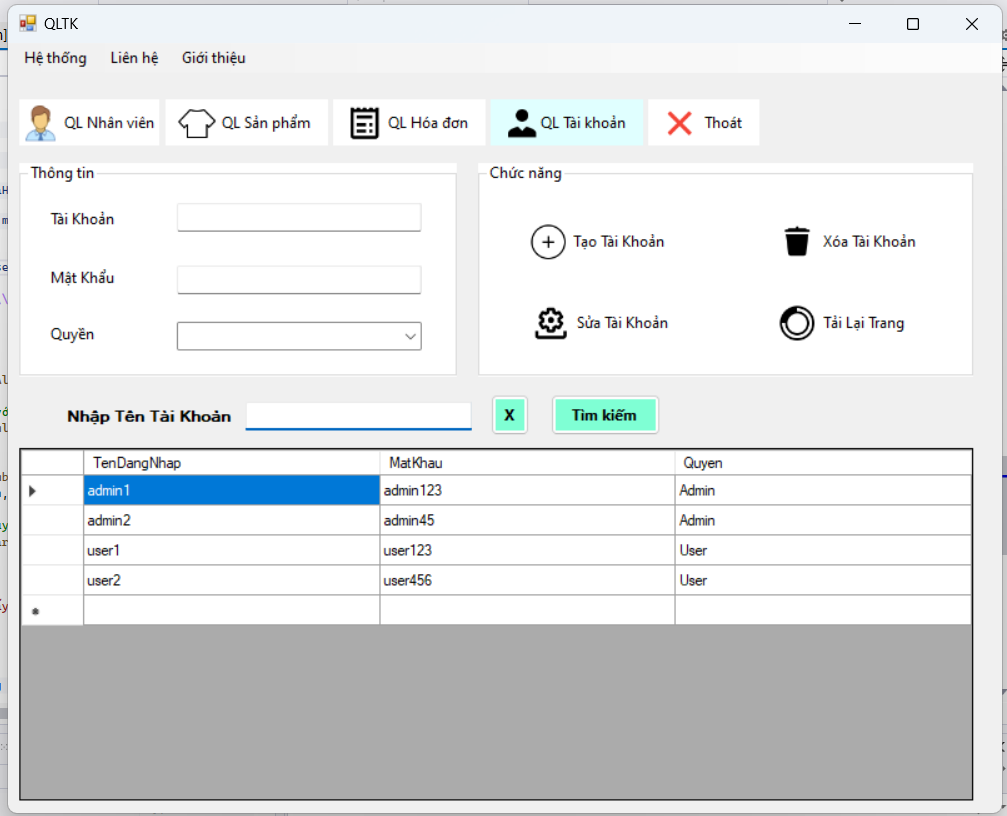
Form quản lý sản phẩm



Form quản lý hóa đơn



Form xuất hóa đơn lên trình duyệt



Form quản lý tài khoản

**Chương 3: Kết luận**

Đây là lần đầu tiên nhóm chúng em áp dụng cônghệ XML vào việc thiết kế, xây dựng phần mềm để quản lý việc bán quần áo cho các cửa hàng. Vận dụng kiến thức đã được học ở trường, kết hợp với những tài liệu tham khảo trên mạng , sách báo, và đặc biệt là sự hướng dẫn của thầy, nhóm thực hiện chúng em đã hoàn thành bài tập lớn. Qua bài báo chúng em muốn gửi lời cảm ơn giảng viên hướng dẫn Thầy Ngô Lê Quân , rất cảm ơn thầy đã nhiệt tình dạy bảo và hướng dẫn chúng em hoàn thành bài tập lớn này. Vì kiến thức còn hạn chế, điều kiện về thời gian hơi gấp nên bài báo cáo không tránh khỏi thiếu sót .Vì thế, nhóm chúng em rất mong nhận được chỉ dẫn, bổ sung, sửa chữa, những đóng góp và sự cảm thông sâu sắc của thầy, cô cũng như bạn bè để bài tập của chúng em được hoàn thiện hơn.Nhóm chúng em xin chân thành cảm ơn!

BẢNG CÔNG VIỆC

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành viên** | **Công việc thực hiện** |
| 1 | Nguyễn Văn Hải | * Form đăng nhập * Form trang chủ * Form quản lý nhân viên |
| 2 | Bùi Xuân Văn | * Form quản lý tài khoản * Form quản lý sản phẩm |
| 3 | Trần Đình Việt | * Form quản lý hóa đơn * Tích hợp tìm kiếm vào các form khác |