# **khung doi**

**TRƯỜNG SỸ QUAN KTQS**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**MÔN: LẬP TRÌNH ỨNG DỤNG QUẢN LÝ**

**ĐỀ TÀI : XÂY DỰNG PHẦN MỀM**

**QUẢN LÝ CỬA HÀNG BÁN SÁCH**

Giảng viên hướng dẫn: Lê Thị Mộng Thanh

Sinh viên thực hiện: Tô Trung Hiếu

Bùi Văn An

Bùi Văn Lâm

Lớp: 23TDH28031

**Tháng 12, năm 2024**

# MỤC LỤC

[MỤC LỤC 1](#_Toc383)

[LỜI NÓI ĐẦU 2](#_Toc14291)

[I. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 3](#_Toc9290)

[1. Đặt vấn đề 3](#_Toc21502)

[- Phương pháp nghiên cứu: 4](#_Toc15173)

[- Đối tượng nghiên cứu: 4](#_Toc24887)

[2. Phân tích hệ thống 4](#_Toc24382)

[II. CÀI ĐẶT CƠ SỞ DỮ LIỆU 6](#_Toc19463)

[1. Mô hình liên kết các bảng trong thiết kế hệ thống 6](#_Toc28094)

[2. Mô tả thuộc tính của bảng 6](#_Toc14932)

[IV. CÁC PROCEDUCE ĐƯỢC TẠO TRONG CƠ SỞ DỮ LIỆU SQL SEVER. 7](#_Toc16784)

[1. Proceduce liên quan đến sách 7](#_Toc7758)

[1.1 Proceduce thêm sách 7](#_Toc17520)

[1.2. Proceduce cập nhật sách 8](#_Toc8676)

[2. Proceduce liên quan đến loại sách 8](#_Toc19241)

[2.1. Proceduce thêm loại sách 8](#_Toc14057)

[2.2. Proceduce cập nhật loại sách 8](#_Toc30636)

[3. Proceduce liên quan đến hóa đơn 9](#_Toc14667)

[3.1. Proceduce thêm hóa đơn 9](#_Toc30954)

[3.2 Proceduce cập nhật hóa đơn 9](#_Toc15366)

[4. Proceduce liên quan đến chi tiết hóa đơn 9](#_Toc32376)

[4.1. Proceduce thêm chi tiết hóa đơn 9](#_Toc7871)

[4.2. Proceduce cập nhật chi tiết hóa đơn 10](#_Toc9792)

[5. Proceduce liên quan đến phiếu nhâp 10](#_Toc14636)

[5.1 Proceduce thêm phiếu nhập 10](#_Toc7373)

[5.2 Proceduce cập nhập nhật phiếu nhập 11](#_Toc4780)

[6. Proceduce liên quan đến chi tiết phiếu nhập 11](#_Toc11513)

[6.1. Proceduce thêm chi tiết phiếu nhập 11](#_Toc4036)

[6.2. Proceduce cập nhật chi tiết phiếu nhâp. 11](#_Toc8707)

[IV. XÂY DỰNG GIAO DIỆN PHẦN MỀM TRÊN MICROSOFT VISUAL STUDIO 12](#_Toc3159)

[1. Form đăng nhập 12](#_Toc14166)

[2. Form quản lý 12](#_Toc20695)

[III. XÂY DỤNG CÁC TÍNH NĂNG CỦA PHẦN MỀM 16](#_Toc27529)

[4.1. Kết nối giữa SQL và VISUAL 16](#_Toc449)

[4.2. Các chức năng của ứng dụng 18](#_Toc30817)

[4.2.1. Tính năng thêm 18](#_Toc25625)

[4.2.2. Tính năng sửa 21](#_Toc32729)

[4.2.3 Tính năng xóa 22](#_Toc26318)

[4.2.4. Tính năng tìm kiếm 24](#_Toc16000)

[4.2.5. Tính năng bán hàng và nhập hàng. 25](#_Toc8231)

[4.2.6. Xuất file hóa đơn/phiếu nhập 30](#_Toc21931)

# LỜI NÓI ĐẦU

Trong thời đại ngày nay, công nghệ thông tin phát triển không ngừng, không chỉ trên thế giới mà ngay cả tại Việt Nam, ngành công nghệ thông tin cũng đang đóng vai trò quan trọng và là một phần không thể thiếu trong cuộc sống thường nhật. Việc ứng dụng công nghệ thông tin vào quản lý các hoạt động kinh doanh đã trở thành xu hướng tất yếu, giúp giảm tải khối lượng công việc, tối ưu hóa quy trình quản lý, và nâng cao hiệu quả cũng như chất lượng công việc.

Trong lĩnh vực kinh doanh sách, việc quản lý hiệu quả các khía cạnh như danh mục sách, thể loại, hóa đơn, và phiếu nhập đóng vai trò quan trọng trong việc đảm bảo sự vận hành trơn tru, minh bạch, và hiệu quả. Một hệ thống quản lý hiện đại không chỉ giúp người dùng dễ dàng kiểm soát số lượng sách, phân loại thể loại một cách rõ ràng, mà còn hỗ trợ quản lý thông tin liên quan đến giao dịch bán hàng và nhập kho, từ đó giảm thiểu sai sót và nâng cao trải nghiệm khách hàng.

Đề tài “Quản lý bán sách” được xây dựng với mục tiêu thiết kế và phát triển một hệ thống quản lý toàn diện, bao gồm các chức năng như lưu trữ thông tin sách, phân loại sách theo thể loại, quản lý hóa đơn bán hàng và các phiếu nhập sách. Hệ thống không chỉ đảm bảo sự chính xác, hiệu quả trong quản lý mà còn cung cấp công cụ phân tích dữ liệu hữu ích, hỗ trợ cho việc ra quyết định kinh doanh.

Với sự phát triển và ứng dụng của hệ thống này, chúng tôi kỳ vọng sẽ mang lại giải pháp tối ưu cho các doanh nghiệp kinh doanh sách, góp phần thúc đẩy sự phát triển của ngành sách và văn hóa đọc trong cộng đồng.

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

1. **Đặt vấn đề**

Trong bối cảnh xã hội hiện đại, nhu cầu đọc sách và tìm kiếm tri thức ngày càng được chú trọng, kéo theo sự phát triển không ngừng của ngành kinh doanh sách. Tuy nhiên, cùng với sự mở rộng quy mô kinh doanh, các cửa hàng và doanh nghiệp phân phối sách đang đối mặt với nhiều khó khăn trong việc quản lý thông tin một cách hiệu quả. Các vấn đề như theo dõi danh mục sách, phân loại thể loại, quản lý hóa đơn bán hàng và kiểm soát phiếu nhập kho thường gây ra sự nhầm lẫn, tốn thời gian và dễ dẫn đến sai sót.

Hệ thống quản lý truyền thống với việc ghi chép thủ công hay sử dụng các phần mềm rời rạc không còn đáp ứng được nhu cầu thực tế của ngành. Những hạn chế này không chỉ ảnh hưởng đến hiệu suất làm việc mà còn làm giảm khả năng phục vụ khách hàng và gây cản trở trong việc ra quyết định kinh doanh.

Trước những thách thức này, việc xây dựng một hệ thống quản lý sách, thể loại, hóa đơn và phiếu nhập tự động hóa, dễ sử dụng và tích hợp là giải pháp cấp thiết. Hệ thống này không chỉ giúp tối ưu hóa các quy trình quản lý mà còn mang lại sự chính xác, tiết kiệm thời gian và nâng cao hiệu quả hoạt động kinh doanh.

Với mục tiêu giải quyết những vấn đề trên, đề tài “Quản lý bán sách” ra đời, tập trung vào việc thiết kế và triển khai một hệ thống quản lý toàn diện, đáp ứng nhu cầu thực tế của ngành kinh doanh sách và góp phần nâng cao chất lượng quản lý doanh nghiệp.

- **Phương pháp nghiên cứu:**

* Tìm hiểu bài toán khảo sát, phân tích, tổng hợp.
* Mô tả yêu cầu chức năng của hệ thống.
* Phân tích thiết kế mô hình CSDL, code nội dung chương trình
* Xây dựng phần mềm dựa trên CSDL đã tạo bằng Winform trên nền tảng ngôn ngữ lập trình C#.

- **Đối tượng nghiên cứu:**

* Thông tin về các sách, thể loại sách của cừa hàng.
* Thông tin về khách hàng, nhà cung cấp.
* Công việc thực hiện các công đoạn.
* Xây dựng mô phòng cơ sở dữ liệu dựa trên phương pháp nghiên cứu.
* Xây dựng phần mềm bằng Microsoft Visual Studio.

1. **Phân tích hệ thống**

Hệ thống quản lý cửa hàng sách được chia làm 6 khối chức năng mỗi khối lần lượt như sau:

* Khối sản phẩm

Thể hiện mặt hàng được bán trong cửa hàng là sách. Gồm các thông tin của sản phẩm như mã sách, tên sách, tên loại sách, tác giả, số lượng, giá bán.

* Khối thể loại

Thể hiện các thể loại sách được bán trong cửa hàng. Gồm các thông tin như mã thể loại, tên thể loại.

* Khối hóa đơn

Quản lý hóa đơn bán lẻ của cửa hàng bao gồm thông tin mã hóa đơn, ngày lập hóa đơn, tên khách hàng, số điện thoại khách hang.

* Khối chi tiết hóa đơn

Quản lý chỉ tiết hóa đơn thông kế số lượng sách có trong hóa đơn từ đó tính ra được giá tiền cho mỗi hóa đơn.

* Khối phiếu nhập

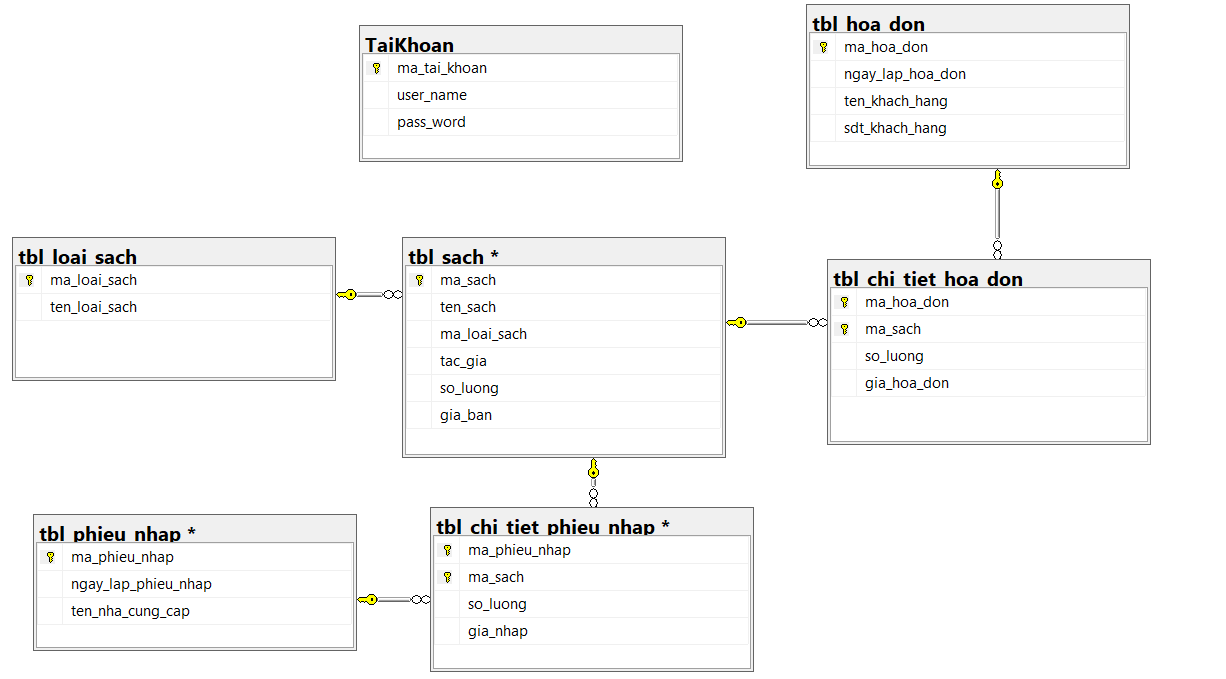
Quản lý chức năng nhập hàng của cửa hàng. Nhập hàng lưu trữ thông tin mã phiếu nhập, ngày lập phiếu nhập, tên nhà cung cấp.

* Khối chi tiết phiếu nhập

Quản lý chỉ tiết phiếu nhập thông kế mã phiếu nhập, mã sách, số lượng, giá nhập

# CÀI ĐẶT CƠ SỞ DỮ LIỆU

## Mô hình liên kết các bảng trong thiết kế hệ thống



1. **Mô tả thuộc tính của bảng**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu**  **dữ liệu** | **Độ rộng** | **Mô tả** |
| **TaiKhoan**  **(Tài khoản)** | ma\_tai\_khoan | Int |  | Mã tài khoản |
| user\_name | Nvarchar | 100 | Tên tài khoản |
| pass\_word | Nvarchar | 100 | Mật khẩu |
| **tbl\_sach**  **(Sách)** | ma\_sach | Int |  | Má sách |
| ten\_sach | Nvarchar | 100 | Tên sách |
| ma\_loai\_sach | Int |  | Mã loại sách |
| tac\_gia | Nvarchar | 100 | Tác giả |
| so\_luong | Int | 10 | Số lượng |
| gia\_ban | Decimal | (18,2) | Gía bán |
| **tbl\_loaisach**  **(Loại sách)** | ma\_loai\_sach | Int |  | Mã loại sách |
| ten\_loai\_sach | Nvarchar | 100 | Tên loại sách |
| **tbl\_hoa\_don**  **(Hóa đơn)** | ma\_hoa\_don | Int |  | Mã hóa đơn |
| ngay\_lap\_hoa\_don | Date | 10 | Ngày lập hóa đơn |
| ten\_khach\_hang | Nvarchar | 100 | Tên khách hàng |
| sdt\_khach\_hang | Nvarchar | 50 | Số điện thoại khách hàng |
| **tbl\_chi\_tiet\_**  **hoa\_don**  **(Chi tiết hóa đơn)** | ma\_hoa\_don | Int |  | Mã hóa đơn |
| ma\_sach | Int |  | Mã sách |
| so\_luong | Int |  | Số lượng |
| gia\_hoa\_don | Float |  | Gía hơn đơn |
| **tbl\_phieu\_nhap**  **(Phiếu nhập)** | ma\_phieu\_nhap | Int |  | Mã phiếu nhập |
| ngay\_lap\_  phieu\_nhap | Date |  | Ngày lập phiếu nhập |
| ten\_nha\_cunng\_cap | Nvarchar | 100 | Tên nhà cung cấp |
| **tbl\_chi\_tiet**  **\_phieu\_nhap** | ma\_phieu\_nhap | Int |  | Mã phiếu nhập |
| ma\_sach | Int |  | Mã Sách |
| so\_luong | Int |  | Số lượng |
| gia\_nhap | Float |  | Gía nhập |

# IV. CÁC PROCEDUCE ĐƯỢC TẠO TRONG CƠ SỞ DỮ LIỆU SQL SEVER.

1. **Proceduce liên quan đến sách**

# 1.1 Proceduce thêm sách

CREATE PROC proc\_them\_sach

@tenSach NVARCHAR(256), @maLoaiSach int, @tacGia NVARCHAR(256), @soLuong INT, @giaBan FLOAT

AS

BEGIN

INSERT INTO tbl\_sach(ten\_sach, ma\_loai\_sach, tac\_gia, so\_luong, gia\_ban)

VALUES(@tenSach, @maLoaiSach, @tacGia, @soLuong, @giaBan)

END

**1.2. Proceduce cập nhật sách**

CREATE PROC proc\_cap\_nhat\_sach

@maSach INT, @tenSach NVARCHAR(256), @maLoaiSach int, @tacGia NVARCHAR(256), @soLuong int, @giaBan float

AS

BEGIN

UPDATE tbl\_sach

SET ten\_sach = @tenSach, ma\_loai\_sach = @maLoaiSach, tac\_gia = @tacGia, so\_luong = @soLuong, gia\_ban = @giaBan

WHERE ma\_sach = @maSach

END

## Proceduce liên quan đến loại sách

* 1. **Proceduce thêm loại sách**

CREATE PROC proc\_them\_loai\_sach

@ten\_loai\_sach NVARCHAR(256)

AS

BEGIN

INSERT INTO tbl\_loai\_sach(ten\_loai\_sach) VALUES(@ten\_loai\_sach)

END

* 1. **Proceduce cập nhật loại sách**

CREATE proc proc\_cap\_nhat\_loai\_sach

@maLoaiSach int, @tenLoaiSach nvarchar(256)

as

begin

update tbl\_loai\_sach

SET ten\_loai\_sach = @tenLoaiSach

WHERE ma\_loai\_sach = @maLoaiSach

END

## Proceduce liên quan đến hóa đơn

**3.1. Proceduce thêm hóa đơn**

CREATE PROC proc\_them\_hoa\_don

@ngayLapHoaDon DATE, @tenKhachHang NVARCHAR(256), @sdtKhachHang NVARCHAR(15)

AS

BEGIN

INSERT INTO tbl\_hoa\_don(ngay\_lap\_hoa\_don, ten\_khach\_hang, sdt\_khach\_hang)

VALUES(@ngayLapHoaDon, @tenKhachHang, @sdtKhachHang)

END

# 3.2 Proceduce cập nhật hóa đơn

CREATE PROC proc\_cap\_nhat\_hoa\_don

@maHoaDon INT, @ngayLapHoaDon DATE, @tenKhachHang NVARCHAR(256), @sdtKhachHang NVARCHAR(15)

AS

BEGIN

UPDATE tbl\_hoa\_don

SET ngay\_lap\_hoa\_don = @ngayLapHoaDon,

ten\_khach\_hang = @tenKhachHang,

sdt\_khach\_hang = @sdtKhachHang

WHERE ma\_hoa\_don = @maHoaDon

END

## Proceduce liên quan đến chi tiết hóa đơn

**4.1. Proceduce thêm chi tiết hóa đơn**

CREATE PROC proc\_them\_chi\_tiet\_hoa\_don

@maHoaDon INT, @maSach INT, @soLuong INT, @giaHoaDon FLOAT

AS

BEGIN

INSERT INTO tbl\_chi\_tiet\_hoa\_don(ma\_hoa\_don, ma\_sach, so\_luong, gia\_hoa\_don)

VALUES(@maHoaDon, @maSach, @soLuong, @giaHoaDon)

END

**4.2. Proceduce cập nhật chi tiết hóa đơn**

CREATE PROC proc\_cap\_nhat\_chi\_tiet\_hoa\_don

@maHoaDon INT, @maSach INT, @soLuong INT, @giaHoaDon FLOAT

AS

BEGIN

UPDATE tbl\_chi\_tiet\_hoa\_don

SET so\_luong = @soLuong

WHERE ma\_hoa\_don = @maHoaDon AND ma\_sach = @maSach

END

1. **Proceduce liên quan đến phiếu nhâp**
   1. **Proceduce thêm phiếu nhập**

CREATE PROC proc\_them\_phieu\_nhap

@ngayLapPhieuNhap DATE, @tenNhaCungCap NVARCHAR(256)

AS

BEGIN

INSERT INTO tbl\_phieu\_nhap(ngay\_lap\_phieu\_nhap, ten\_nha\_cung\_cap)

VALUES(@ngayLapPhieuNhap, @tenNhaCungCap)

END

* 1. **Proceduce cập nhập nhật phiếu nhập**

CREATE PROC proc\_cap\_nhat\_phieu\_nhap

@maPhieuNhap INT, @ngayLapPhieuNhap DATE, @tenNhaCungCap NVARCHAR(256)

AS

BEGIN

UPDATE tbl\_phieu\_nhap

SET ngay\_lap\_phieu\_nhap = @ngayLapPhieuNhap,

ten\_nha\_cung\_cap = @tenNhaCungCap

WHERE ma\_phieu\_nhap = @maPhieuNhap

END

1. **Proceduce liên quan đến chi tiết phiếu nhập**

**6.1. Proceduce thêm chi tiết phiếu nhập**

CREATE PROC proc\_them\_chi\_tiet\_phieu\_nhap

@maPhieuNhap INT, @maSach INT, @soLuong INT, @giaNhap float

AS

BEGIN

INSERT INTO tbl\_chi\_tiet\_phieu\_nhap(ma\_phieu\_nhap, ma\_sach, so\_luong, gia\_nhap)

VALUES(@maPhieuNhap, @maSach, @soLuong, @giaNhap)

END

**6.2. Proceduce cập nhật chi tiết phiếu nhâp.**

CREATE PROC proc\_cap\_nhat\_chi\_tiet\_phieu\_nhap

@maPhieuNhap INT, @maSach INT, @soLuong INT, @giaNhap float

AS

BEGIN

UPDATE tbl\_chi\_tiet\_phieu\_nhap

SET so\_luong = @soLuong

WHERE ma\_phieu\_nhap = @maPhieuNhap AND ma\_sach = @maSach

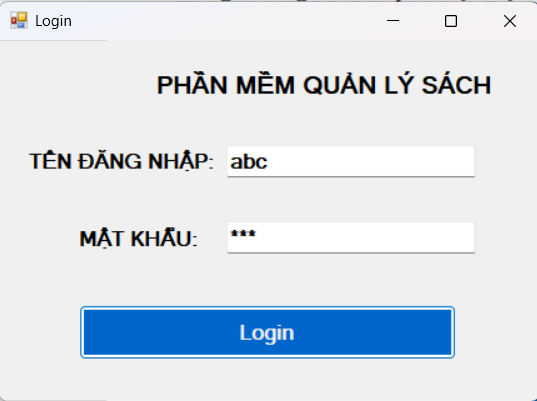
END

# IV. XÂY DỰNG GIAO DIỆN PHẦN MỀM TRÊN MICROSOFT VISUAL STUDIO

Phần mềm gòm các gồm các form: form đăng nhập, form quản lý, form chi tiết phiếu nhập, form chi tiết hóa đơn

1. **Form đăng nhập**

Có chức năng chính là nhập thông tin về tài khoản và mật khẩu, so sánh với CSDL và chọn quyền truy cập để chuyển hướng đến tab control thích hợp.



1. **Form quản lý**

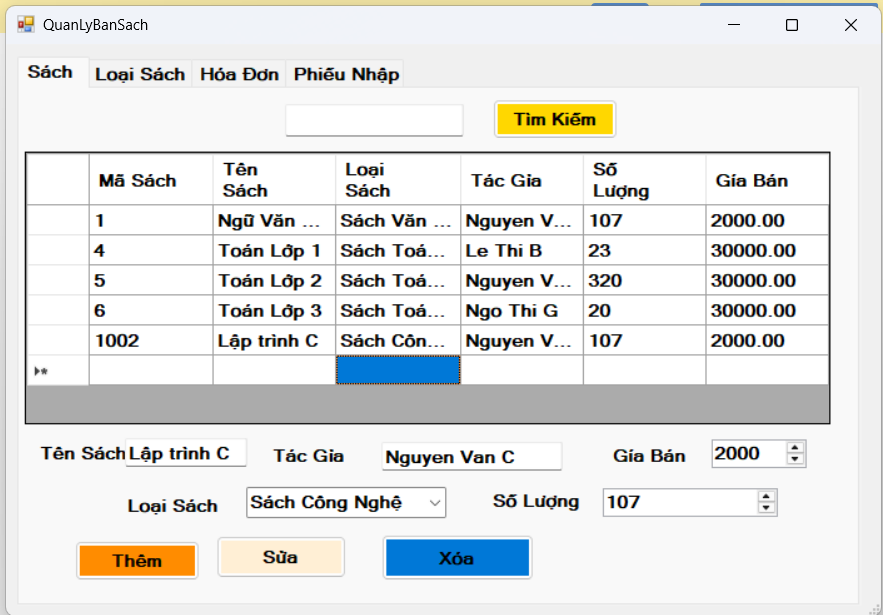
Là form làm việc cho người có chức vụ quản lý. Có các tab như: sách, loại sách, hóa đơn, phiếu nhập.

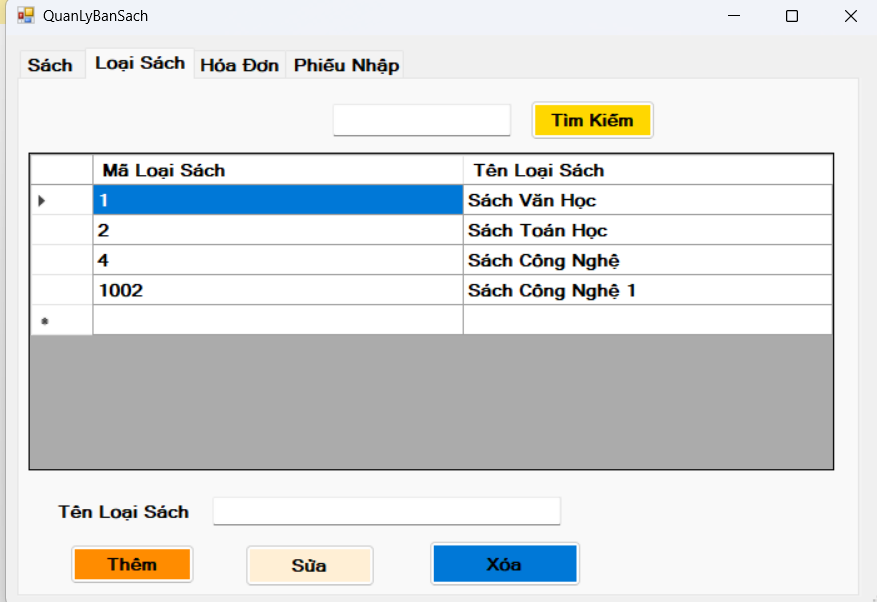
Giao diện form có các panel và label để thể hiện tiêu đề của hệ thống, thông tin học viên thực hiện.

Các tab có giao diện cơ bản giống nhau, thực hiện các chức năng thêm, sửa, xóa đối tượng, tìm kiếm.

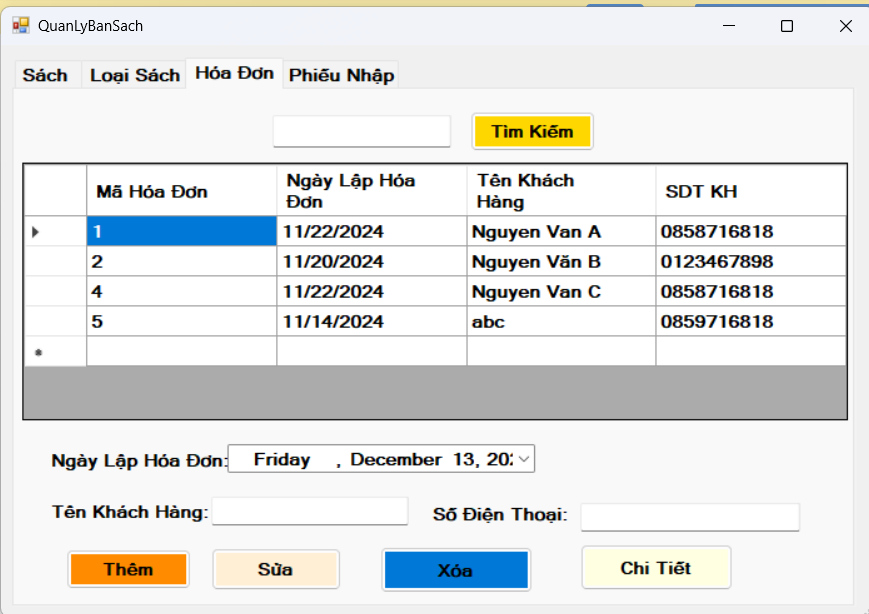
Datagridview tại vị trí trung tâm sẽ hiển thị nội dung của thuộc tính được lấy từ SQL.

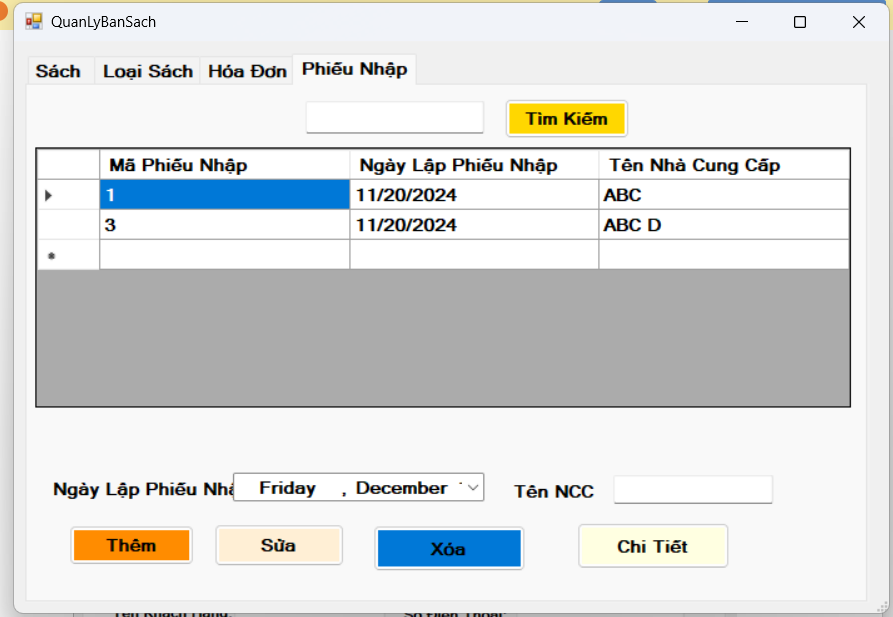
Tab sách và loại sách có mục đích chính là theo dõi tình trạng sách, loại sách và sau đó có thể thêm, xóa, sửa và tìm kiếm các thông tin cần thiết.



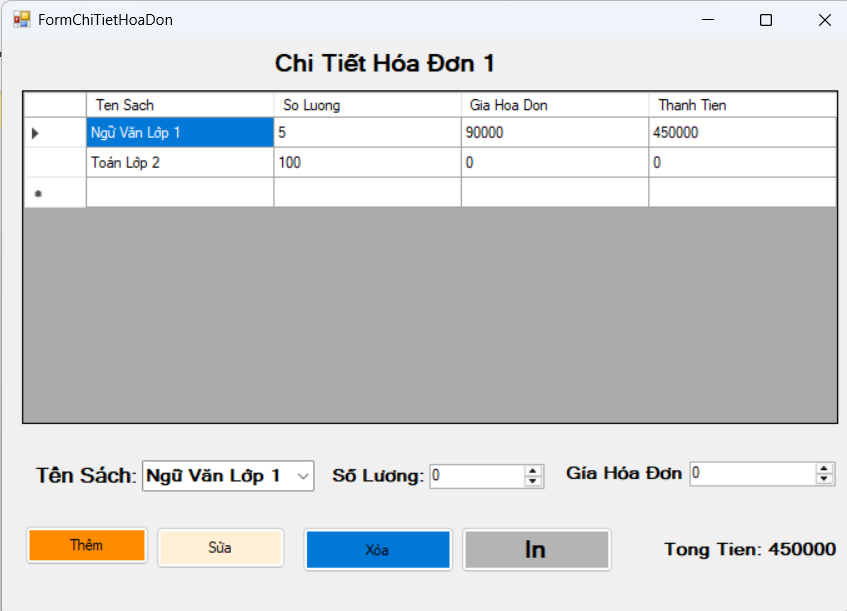


Tab hóa đơn và phiếu nhập có giao diện giống nhau với mục địch chính là theo dõi và tra xuất hóa đơn theo thời giam nhất định. Có button tìm kiếm, thêm, sửa, xóa, chi tiết hóa đơn, chi phiếu nhập. Datagridview sẽ hiển thị nội dung tìm kiếm được lấy từ SQL.

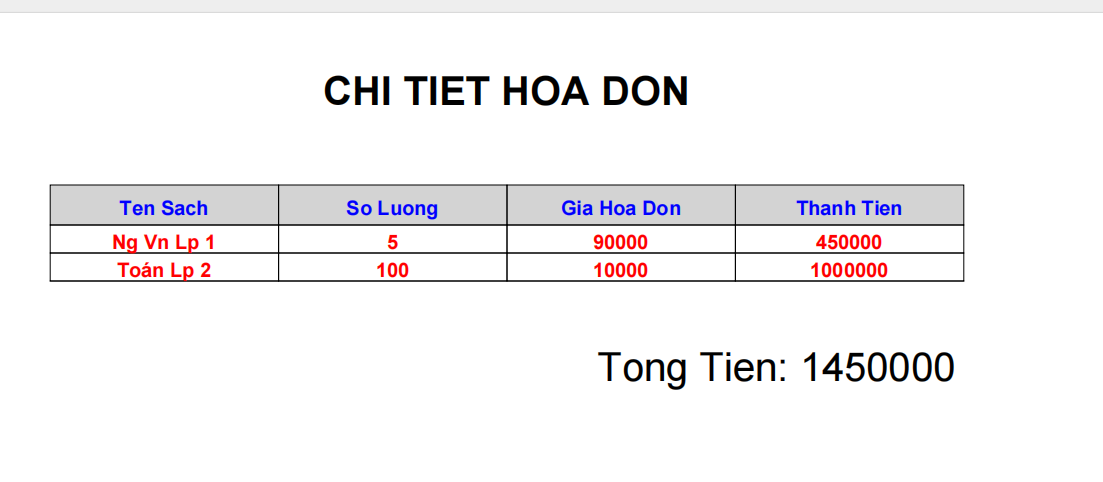




Người dùng có thể click và các button ‘Chi Tiết’ để chuyển hướng đến chi tiết hóa đơn hoặc phiếu nhập tương ứng để xem số lượng , giá bán và tổng tiền của hơn đơn hoặc phiếu nhập từ đó có thể điều chỉnh thêm, xóa ,sửa, tìm kiếm cho từng chi tiết hóa đơn, phiếu nhập.



Sau khi điều chỉnh đủ các chi tiết hóa đơn, phiếu nhập cần thiết thì có thể xuất hóa đơn, phiếu nhập



# XÂY DỤNG CÁC TÍNH NĂNG CỦA PHẦN MỀM

**4.1**. **Kết nối giữa SQL và VISUAL**

Muốn liên kết đầu tiên cần thêm thư viện của SQL vào visual

using System.Data.SqlClient;

Tạo một string chứa đường dẫn liên kết đên sql

private string connectionString = @"Data Source=DESKTOP;Connect Timeout=30;Encrypt=True;Initial Catalog=QLBS;TrustServerCertificate=True";

Các hàm cung cấp sau được sử dụng để tương tác với cơ sở dữ liệu SQL.

Mục đích của hàm execQuery là thực hiện một truy vấn SQL và trả về dữ liệu dưới dạng một bảng (DataTable). Thông thường cho các truy vấn lấy dữ liệu (chẳng hạn như SELECT).

public DataTable execQuery(string query)

{

DataTable dt = new DataTable();

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

SqlCommand cmd = new SqlCommand(query, conn);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

adapter.Fill(dt);

conn.Close();

}

return dt;

}

Mục đích của hàm execNonQuery là thực hiện một truy vấn SQL không trả về kết quả (ví dụ: câu lệnh thay đổi dữ liệu hoặc cấu trúc cơ sở dữ liệu).

Hàm này thích hợp cho các truy vấn như INSERT, UPDATE, DELETE, hoặc các câu lệnh DDL như CREATE TABLE. Phương thức này trả về số lượng dòng bị ảnh hưởng bởi câu lệnh (ví dụ: số lượng bản ghi bị thay đổi).

public int execNonQuery(string query)

{

int data = 0;

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

SqlCommand cmd = new SqlCommand(query, conn);

data = cmd.ExecuteNonQuery();

conn.Close();

}

return data;

}

Mục đích của hàm execScaler là thực hiện một truy vấn SQL và trả về một giá trị đơn (scaler)

Hàm này được sử dụng khi bạn muốn lấy một giá trị duy nhất từ cơ sở dữ liệu, ví dụ: số lượng bản ghi, giá trị tối đa hoặc trung bình. Phương thức này trả về một đối tượng, thường là kiểu int, string, decimal, hoặc null nếu không có giá trị nào được tìm thấy.

Thường dùng cho các truy vấn cần trả về một giá trị đơn (chẳng hạn như SELECT COUNT(\*) hoặc SELECT MAX(Price)).

public object execScaler(string query)

{

object data = 0;

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

SqlCommand cmd = new SqlCommand(query, conn);

data = cmd.ExecuteScalar();

conn.Close();

}

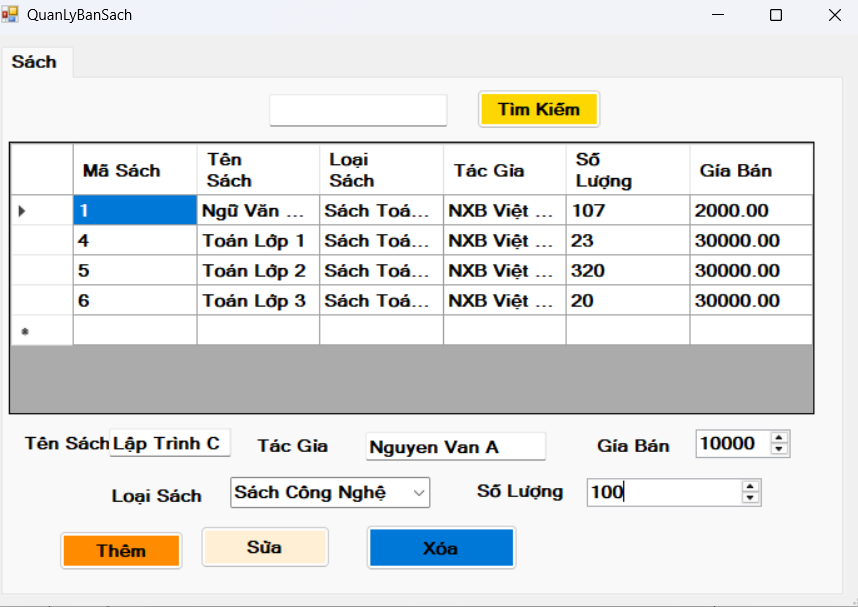
return data;

}

**4.2. Các chức năng của ứng dụng**

Các tính năng thêm sách loại sách, hóa, đơn, phiếu nhập, … về thuật toán giống nhau chỉ khác cấu trúc lệnh thực hiện trên sql.

**4.2.1. Tính năng thêm**



Người dùng nhập tên sách, tác giá, số lượng, chọn sách thuộc thể loại nào và nhấn nút “Thêm” để thêm một cuốn sách mới vào cơ sở dữ liệu sách.

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

{

if (string.IsNullOrWhiteSpace(txtSachTenSach.Text))

{

MessageBox.Show("Tên sách không được để trống!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (string.IsNullOrWhiteSpace(txtSachTacGia.Text))

{

MessageBox.Show("Tác giả không được để trống!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (numSachSoLuong.Value <= 0)

{

MessageBox.Show("Số lượng sách phải lớn hơn 0!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (numSachGiaBan.Value <= 0)

{

MessageBox.Show("Giá bán phải lớn hơn 0!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

StringBuilder query = new StringBuilder("EXEC proc\_them\_sach");

query.Append(" @tenSach = N'" + txtSachTenSach.Text + "'");

query.Append(",@maLoaiSach = " + maSachLoaiSach);

query.Append(",@tacGia = N'" + txtSachTacGia.Text + "'");

query.Append(",@soLuong = " + numSachSoLuong.Value);

query.Append(",@giaBan = " + numSachGiaBan.Value);

int result = dataProvider.execNonQuery(query.ToString());

if (result > 0)

{

MessageBox.Show("Thêm Sách thành công!", "Thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);

loaddgSach();

}

else

{

MessageBox.Show("Thêm sách không thành công!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

Sau khi thêm 1 loại sách mới thì load datagridview để hiện thị danh sách trong cơ sở dữ liệu lại.

private void loaddgSach()

{

DataTable dt = new DataTable();

StringBuilder query = new StringBuilder("SELECT ma\_sach as [Mã Sách]");

query.Append(",ten\_sach as [Tên Sách]");

query = query.Append(", ten\_loai\_sach as [Loại Sách]");

query.Append(",tac\_gia as [Tác Gỉa]");

query.Append(",so\_luong as [Số Lượng]");

query.Append(",gia\_ban as [Gía Bán]");

query.Append(" FROM tbl\_sach, tbl\_loai\_sach");

query.Append(" WHERE tbl\_sach.ma\_loai\_sach = tbl\_loai\_sach.ma\_loai\_sach");

dt = dataProvider.execQuery(query.ToString());

dgSach.DataSource = dt;

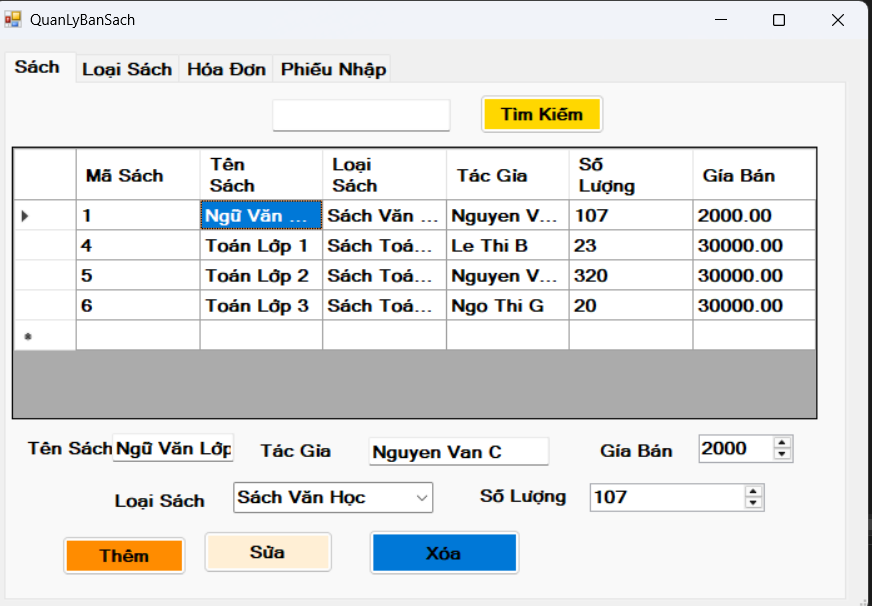
maSachSach = (int)dt.Rows[0][0];

}

**4.2.2. Tính năng sửa**

Người dùng click vào hàng cấn sửa trong datagridview để lấy dòng cần sửa

sau đó có thể điều chỉnh các thông tin cần thiết trong panel và nhấn nút “Sửa”.



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

{

if (string.IsNullOrWhiteSpace(txtSachTenSach.Text))

{

MessageBox.Show("Tên sách không được để trống!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (string.IsNullOrWhiteSpace(txtSachTacGia.Text))

{

MessageBox.Show("Tác giả không được để trống!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (numSachSoLuong.Value <= 0)

{

MessageBox.Show("Số lượng sách phải lớn hơn 0!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (numSachGiaBan.Value <= 0)

{

MessageBox.Show("Giá bán phải lớn hơn 0!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

StringBuilder query = new StringBuilder("EXEC proc\_cap\_nhat\_sach");

query = query.Append(" @maSach = " + maSachSach);

query.Append(",@tenSach = N'" + txtSachTenSach.Text + "'");

query.Append(",@maLoaiSach = " + maSachLoaiSach);

query.Append(",@tacGia = N'" + txtSachTacGia.Text + "'");

query.Append(",@soLuong = " + numSachSoLuong.Value);

query.Append(",@giaBan = " + numSachGiaBan.Value);

int result = dataProvider.execNonQuery(query.ToString());

if (result > 0)

{

MessageBox.Show("Cập nhật Sách thành công!", "Thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);

loaddgSach();

}

else

{

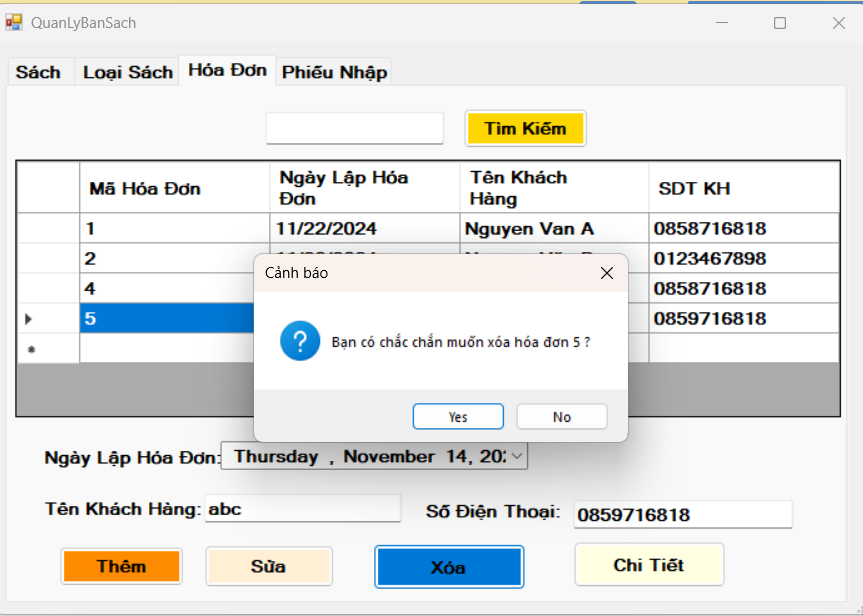
MessageBox.Show("Cập nhật Sách không thành công!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

**4.2.3 Tính năng xóa**

Tương tự tính năng sửa người dùng muốn xóa một thông tin trong các tab control bất kì vẫn click vào một hàng bất kì trong datagridview sau đó kiểm tra thông tin cần xóa rồi nhấn nút “Xóa” .



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

{

DialogResult check = MessageBox.Show("Bạn có chắc chắn muốn xóa hóa đơn " + maHoaDonHoaDon+ " ?", "Cảnh báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (check == DialogResult.Yes) {

string query = "DELETE FROM tbl\_hoa\_don WHERE ma\_hoa\_don = " + maHoaDonHoaDon;

int result = dataProvider.execNonQuery(query);

if (result > 0)

{

loadDgoaDon();

MessageBox.Show("Xóa Hóa đơn thành công!", "Thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("Xóa Hóa đơn không thành công!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

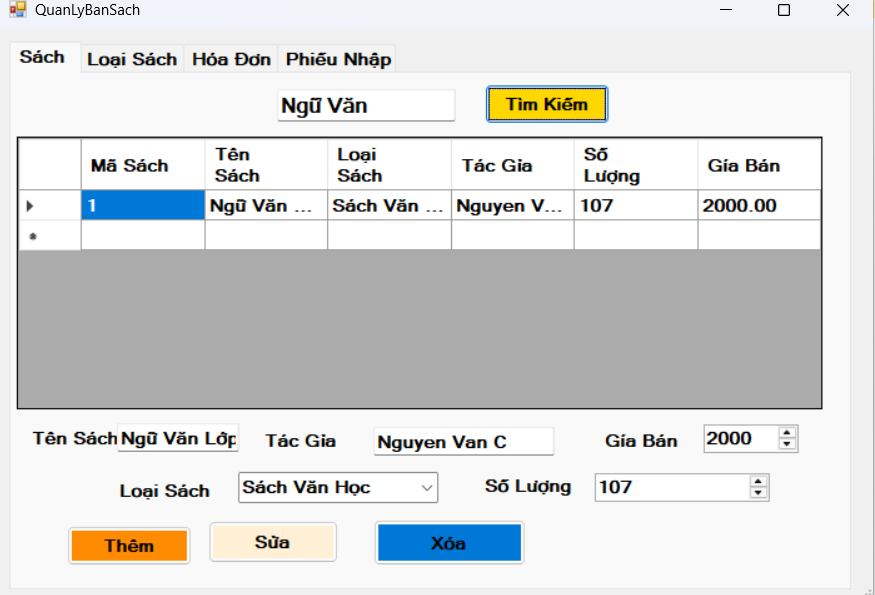
}

}

}

**4.2.4. Tính năng tìm kiếm**

Người dùng nhập thông tin về sách, loại sách, tên giá giả bất kì sau đó nhất nút “Tìm kiếm”. Hệ thống sẽ tự động tìm kiếm thông tin và nạp lại dữ liệu và đưa ra view cho người dùng.



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

{

string keyword = txtSearchSach.Text.Trim();

if (!string.IsNullOrEmpty(keyword))

{

StringBuilder query = new StringBuilder("SELECT ma\_sach as [Mã Sách]");

query.Append(",ten\_sach as [Tên Sách]");

query.Append(", ten\_loai\_sach as [Loại Sách]");

query.Append(",tac\_gia as [Tác Gỉa]");

query.Append(",so\_luong as [Số Lượng]");

query.Append(",gia\_ban as [Gía Bán]");

query.Append(" FROM tbl\_sach, tbl\_loai\_sach");

query.Append(" WHERE tbl\_sach.ma\_loai\_sach = tbl\_loai\_sach.ma\_loai\_sach");

query.Append(" AND (ten\_sach LIKE N'%" + keyword + "%'");

query.Append(" OR ten\_loai\_sach LIKE N'%" + keyword + "%'");

query.Append(" OR tac\_gia LIKE N'%" + keyword + "%')");

DataTable dt = dataProvider.execQuery(query.ToString());

dgSach.DataSource = dt;

if (dt.Rows.Count == 0)

{

MessageBox.Show("Không tìm thấy kết quả nào!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

else

{

MessageBox.Show("Vui lòng nhập từ khóa tìm kiếm!", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

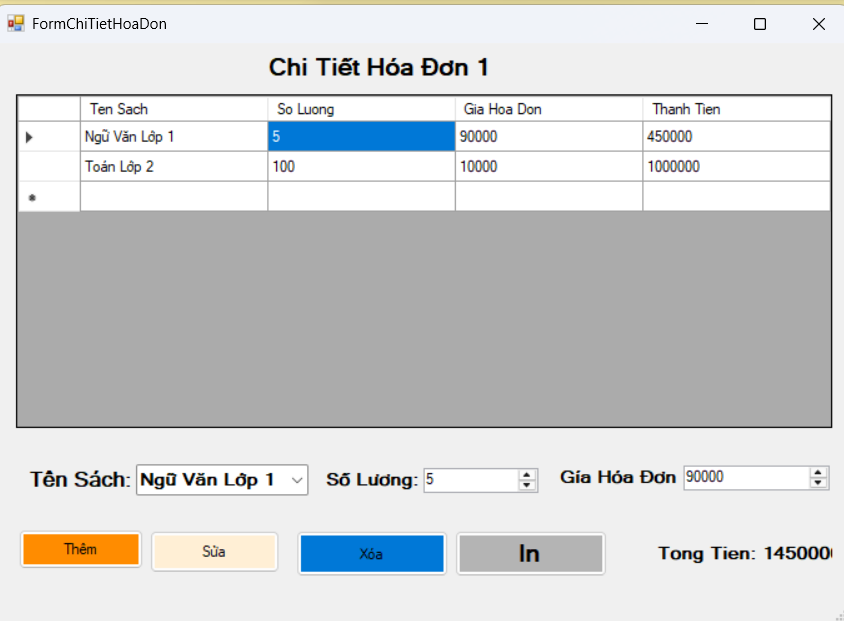
loaddgSach(); // Load lại toàn bộ dữ liệu nếu không nhập từ khóa

}

}

**4.2.5. Tính năng bán hàng và nhập hàng.**

Người dùng click vào hóa đơn hoặc nhập hàng bất kì trên datagridview và kiểm tra thông tin trong control panel sau đó nhấn nút “Chi Tiết” để xem thông tin chi tiết của hóa đơn hoặc nhập từ đó có thể điều chỉnh các thông tin hóa đơn cần thiết như thêm, xóa, sửa.



Thêm 1 chi tiết hóa đơn mới, người dùng nhập tên sách, số lượng, giá hóa đơn sau đó nhấn nút “Thêm”. Hệ thống sẽ sẽ tự động kiểm tra xem số lượng tồn kho có đủ cho người dùng mua và tự động cập nhật số lượng sách, thành tiền cho từng chi tiết hóa đơn và tổng tiền nếu số lượng người dùng chọn là phù hợp.

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

{

int soLuongTon = (int)dataProvider.execScaler("SELECT so\_luong FROM tbl\_sach WHERE ma\_sach = " + maSach);

// Kiểm tra số lượng tồn kho

if (numSachSoLuong.Value > soLuongTon)

{

MessageBox.Show("Không đủ số lượng sách trong kho!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

int dem = (int)dataProvider.execScaler("SELECT count(\*) FROM tbl\_chi\_tiet\_hoa\_don WHERE ma\_hoa\_don = " + maHoaDon + " AND ma\_sach = " + maSach);

if (dem == 0)

{

StringBuilder query = new StringBuilder("EXEC proc\_them\_chi\_tiet\_hoa\_don");

query.Append(" @maHoaDon = " + maHoaDon);

query.Append(",@maSach = " + maSach);

query.Append(",@soLuong = " + numSachSoLuong.Value);

query.Append(",@giaHoaDon = " + numSachGiaHoaDon.Value);

int result = dataProvider.execNonQuery(query.ToString());

if (result > 0)

{

dataProvider.execNonQuery("UPDATE tbl\_sach SET so\_luong = so\_luong - " + numSachSoLuong.Value + " WHERE ma\_sach = " + maSach);

MessageBox.Show("Thêm Sách vào phiếu hóa đơn thành công!", "Thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);

loadDgHoaDon();

loadTongTien();

}

else

{

MessageBox.Show("Thêm sách vào phiếu hóa đơn không thành công!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

dem = (int)dataProvider.execScaler("SELECT SUM(so\_luong) FROM tbl\_chi\_tiet\_hoa\_don WHERE ma\_hoa\_don = " + maHoaDon + " AND ma\_sach = " + maSach);

update(dem);

}

Người dùng có thể sửa một hóa đơn hoặc nhập hàng bằng cách click vào hàng bất kì cần sửa trong datagridview sau đó điều chỉnh thông tin cần cập nhật và nhất nút “Sửa”. Hệ thống sẽ tự động cập nhật tổng hóa đơn, thành tiền và cập số lượn tồn kho trong kho sách.

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

{

update(0);

}

private void update(int soLuong)

{

StringBuilder query = new StringBuilder("EXEC proc\_cap\_nhat\_chi\_tiet\_hoa\_don");

query.Append(" @maHoaDon = " + maHoaDon);

query.Append(",@maSach = " + maSach);

query.Append(",@soLuong = " + (numSachSoLuong.Value + soLuong));

query.Append(",@giaHoaDon = " + numSachGiaHoaDon.Value);

int result = dataProvider.execNonQuery(query.ToString());

if (result > 0)

{

if(soLuong <= numSachSoLuong.Value)

{

dataProvider.execNonQuery("UPDATE tbl\_sach SET so\_luong = so\_luong + " + (numSachSoLuong.Value - soLuong) + " WHERE ma\_sach = " + maSach);

} else

{

dataProvider.execNonQuery("UPDATE tbl\_sach SET so\_luong = so\_luong - " + (soLuong - numSachSoLuong.Value) + " WHERE ma\_sach = " + maSach);

}

MessageBox.Show("Cập nhật số lượng trong hóa đơn thành công!", "Thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);

loadDgHoaDon();

loadTongTien();

}

else

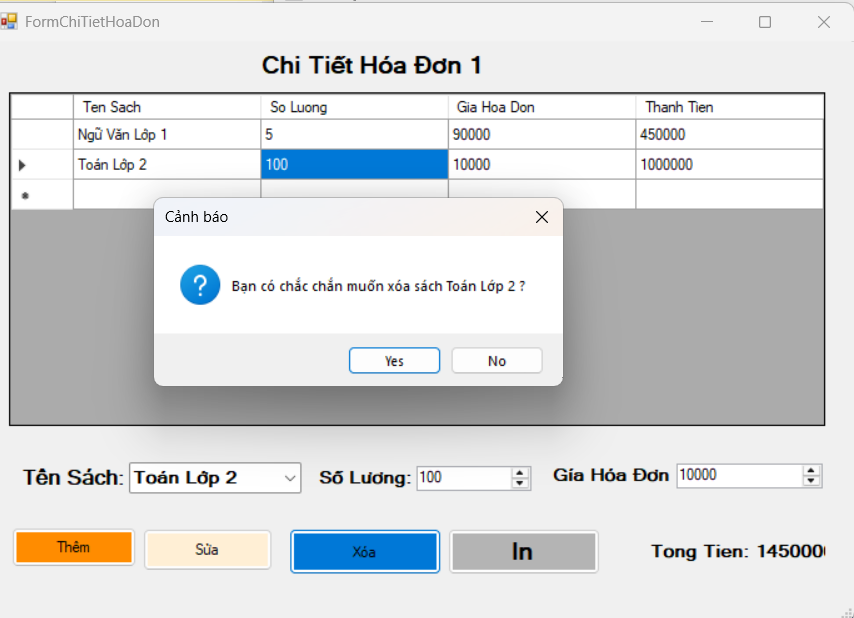
{

MessageBox.Show("Cập nhật số lượng trong hóa đơn không thành công!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

Nếu người dùng muốn xóa một chi tiết hóa đơn và phiếu nhập thì nhấn vào một hàng cần xóa trên datagridview sau đó kiểm tra thông tin cần xóa và nhấn nút “Xóa”, nhất “Yes” nếu đồng ý xóa chi tiết hóa đơn.



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

{

DialogResult check = MessageBox.Show("Bạn có chắc chắn muốn xóa sách " + tenSach + " ?", "Cảnh báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (check == DialogResult.Yes)

{

int soLuongTrongHoaDon = (int)dataProvider.execScaler("SELECT so\_luong FROM tbl\_chi\_tiet\_hoa\_don WHERE ma\_hoa\_don = " + maHoaDon + " AND ma\_sach = " + maSach);

string query = "DELETE FROM tbl\_chi\_tiet\_hoa\_don WHERE ma\_hoa\_don= " + maHoaDon + "AND ma\_sach = " + maSach;

int result = dataProvider.execNonQuery(query);

if (result > 0)

{

dataProvider.execNonQuery("UPDATE tbl\_sach SET so\_luong = so\_luong + " + soLuongTrongHoaDon + " WHERE ma\_sach = " + maSach);

MessageBox.Show("Xóa khỏi hóa đơn thành công!", "Thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);

loadDgHoaDon();

loadTongTien();

}

else

{

MessageBox.Show("Xóa khỏi hóa đơn không thành công!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

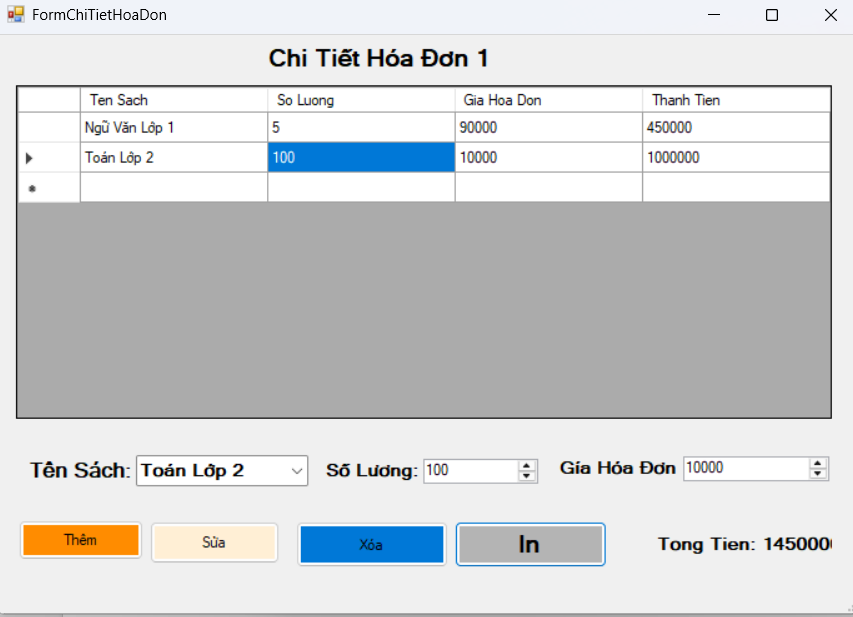
}

}

**4.2.6. Xuất file hóa đơn/phiếu nhập**

Sau khi đã kiểm tra đầy đủ thông tin của một hóa đơn/phiếu nhập, người dùng có thể in hóa đơn/phiếu nhập.

Trên giao diện người dùng nhấn nút “In’.



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

{

if (dgChiTietHD.Rows.Count > 0)

{

SaveFileDialog saveFileDialog = new SaveFileDialog();

saveFileDialog.Filter = "PDF (\*.pdf)|\*.pdf";

saveFileDialog.FileName = "HoaDon";

if (saveFileDialog.ShowDialog() == DialogResult.OK)

{

try

{

using (FileStream fileStream = new FileStream(saveFileDialog.FileName, FileMode.Create))

{

Document document = new Document(PageSize.A4, 8f, 16f, 16f, 8f);

PdfWriter.GetInstance(document, fileStream);

document.Open();

iTextSharp.text.Font font3 = new iTextSharp.text.Font(iTextSharp.text.Font.FontFamily.HELVETICA, 20f, iTextSharp.text.Font.BOLD);

//Add title to the PDF document

Paragraph title = new Paragraph("CHI TIET HOA DON", font3);

title.Alignment = Element.ALIGN\_CENTER;

//Thiet lap khoang cach le tren va le duoi

title.SpacingBefore = 20f;

title.SpacingAfter = 40f;

document.Add(title);

PdfPTable pTable = new PdfPTable(dgChiTietHD.Columns.Count);

pTable.DefaultCell.Padding = 2;

pTable.WidthPercentage = 80;

pTable.HorizontalAlignment = Element.ALIGN\_CENTER;

float[] columnWidths = new float[dgChiTietHD.Columns.Count];

float equalWidth = 100f / dgChiTietHD.Columns.Count; // Calculate equal width for each column

for (int i = 0; i < dgChiTietHD.Columns.Count; i++)

{

columnWidths[i] = equalWidth; // Set equal width for each column

}

pTable.SetWidths(columnWidths);

iTextSharp.text.Font font1 = new iTextSharp.text.Font(iTextSharp.text.Font.FontFamily.HELVETICA, 10, iTextSharp.text.Font.BOLD, BaseColor.BLUE);

// Add column headers to the PDF table

foreach (DataGridViewColumn col in dgChiTietHD.Columns)

{

PdfPCell pCell = new PdfPCell(new Phrase(col.HeaderText, font1));

pCell.BackgroundColor = new BaseColor(System.Drawing.Color.LightGray);

pCell.Padding = 5;

pCell.HorizontalAlignment = Element.ALIGN\_CENTER;

pTable.AddCell(pCell);

}

iTextSharp.text.Font font11 = new iTextSharp.text.Font(iTextSharp.text.Font.FontFamily.HELVETICA, 10f, iTextSharp.text.Font.BOLD, BaseColor.RED);

foreach (DataGridViewRow viewRow in dgChiTietHD.Rows)

{

foreach (DataGridViewCell cell in viewRow.Cells)

{

PdfPCell contentCell = new PdfPCell(new Phrase(cell.Value?.ToString(), font11));

contentCell.HorizontalAlignment = Element.ALIGN\_CENTER;

pTable.AddCell(contentCell);

}

}

document.Add(pTable);

//set Font

iTextSharp.text.Font font = new iTextSharp.text.Font(iTextSharp.text.Font.FontFamily.HELVETICA, 20f, iTextSharp.text.Font.BOLD);

font.SetStyle(iTextSharp.text.Font.NORMAL);

//Add title to the PDF document

Paragraph ending = new Paragraph(txtTongTien.Text + " ", font);

ending.Alignment = Element.ALIGN\_RIGHT;

//Thiet lap khoang cach le tren va le duoi

ending.SpacingBefore = 20f;

document.Add(ending);

document.Close();

}

MessageBox.Show("Data Printed Successfully", "Info");

}

catch (Exception ex)

{

MessageBox.Show("Error While Exporting Data: " + ex.Message, "Error");

}

}

}

else

{

MessageBox.Show("No Record Found", "Info");

}

}