**TRƯỜNG ĐẠI HỌC AN GIANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

**Chuyên đề Dotnet (TIE501)**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ CỬA HÀNG XE MÁY  
VỚI VISUAL STUDIO, SQL SERVER MANAGEMENT STUDIO 21**

**Giảng viên hướng dẫn: Ths. Nguyễn Ngọc Minh**

**Sinh viên thực hiện:**

**DTH235778 - Huỳnh Minh Thuận - Lớp DH24TH3\_NhómTH2\_TổTH2**

**DTH235742 - Tôn Anh Quân - Lớp DH24TH2\_NhómTH2\_TổTH2**

**AN GIANG, 10-2025**

**MỤC LỤC**

[I. Giới thiệu chung 2](#_Toc214820982)

**[1. Đặt vấn đề](#_Toc214820983)** [2](#_Toc214820983)

**[2. Mục tiêu đề tài](#_Toc214820984)** [2](#_Toc214820984)

**[3. Công cụ sử dụng](#_Toc214820985)** [2](#_Toc214820985)

**[4. Phạm vi đề tài](#_Toc214820986)** [2](#_Toc214820986)

[II.Phân tích thiết kế hệ thống 3](#_Toc214820987)

[III.Giao diện chương trình 3](#_Toc214820988)

**[A) Thiết kế giao diện](#_Toc214820989)** [3](#_Toc214820989)

**[B) Cách xây dựng chương trình](#_Toc214820990)** [8](#_Toc214820990)

**[1)Chuẩn bị CSDL SQL SERVER MANAGEMENT STUDIO 21](#_Toc214820991)** [8](#_Toc214820991)

**[2)Viết code chương trình](#_Toc214820992)** [10](#_Toc214820992)

[IV. Kết luận 35](#_Toc214820993)

[V.Tài liệu tham khảo 35](#_Toc214820994)

# I. Giới thiệu chung

## **1. Đặt vấn đề**

Cửa hàng kinh doanh xe máy cần một hệ thống quản lý hiệu quả để theo dõi sản phẩm, khách hàng, nhân viên và tồn kho. Ứng dụng được xây dựng nhằm giảm thiểu sai sót và tăng năng suất quản lý.

## **2. Mục tiêu đề tài**

* Phân tích và xây dựng một ứng dụng trong lập trình chương trình quản lý cửa hàng xe máy.
* Vận dụng kỹ thuật lập trình ứng dụng trên Windows để giải quyết mọi vấn đề trong việc quản lý cửa hang bán xe máy một cách thuận tiện, tổng quát và tối ưu cho việc kinh doanh.
* Tổ chức làm việc nhóm để thảo luận và giải quyết các vấn đề liên quan đến lập trình ứng dụng trên Windows

**Hình thức:**

* Báo cáo + Demo chương trình

## **3. Công cụ sử dụng**

-Dotnet

-SQL server management studio 21

## **4. Phạm vi đề tài**

**- Quản lý Sản phẩm Xe máy :**

-Thêm, xóa, sửa thông tin chi tiết của từng mẫu Xe.Tìm kiếm và lọc sản phẩm theo nhiều tiêu chí (Mã sản phẩm, Hãng, Kích thước, v.v.).

**- Quản lý Tồn kho :**

-Theo dõi số lượng nhập, xuất, tồn kho hiện tại của từng mẫu Xe.

-Cập nhật tình trạng tồn kho khi có giao dịch mua bán hoặc nhập hàng mới.

-Cảnh báo tồn kho thấp (nếu có thể).

**Quản lý Khách hàng:**

-Lưu trữ thông tin khách hang.Quản lý lịch sử mua hàng của từng khách hàng.

**Quản lý Giao dịch (Bán hàng):**

-Lập hóa đơn bán hàng, ghi nhận thông tin sản phẩm, số lượng, giá bán, chiết khấu (nếu có) và tổng tiền.Lưu trữ thông tin giao dịch vào cơ sở dữ liệu.

# 

# II.Phân tích thiết kế hệ thống

Cơ sở dữ liệu được thiết kế với 6 bảng chính, liên kết với nhau qua các khóa ngoại (Foreign Key) để đảm bảo tính toàn vẹn dữ liệu:

* **FormKhachHang:** Lưu thông tin khách hàng ( Họ tên, Địa chỉ, Số điện thoại).
* **FormXeMay:** Lưu thông tin chi tiết về các loại Xe máy (Hãng xe, Tên xe, Màu xe, Số lượng, Giá xe).
* **FormTonKho:** Theo dõi số lượng tồn kho của từng loại Xe theo ngày cập nhật (Liên kết với FormXeMay).
* **FormHoaDon:** Lưu thông tin hóa đơn (Khách hàng, Ngày lập, Xe máy, Tổng tiền, Số lượng) liên kết với **Xe Máy** và **Khách hàng**

# 

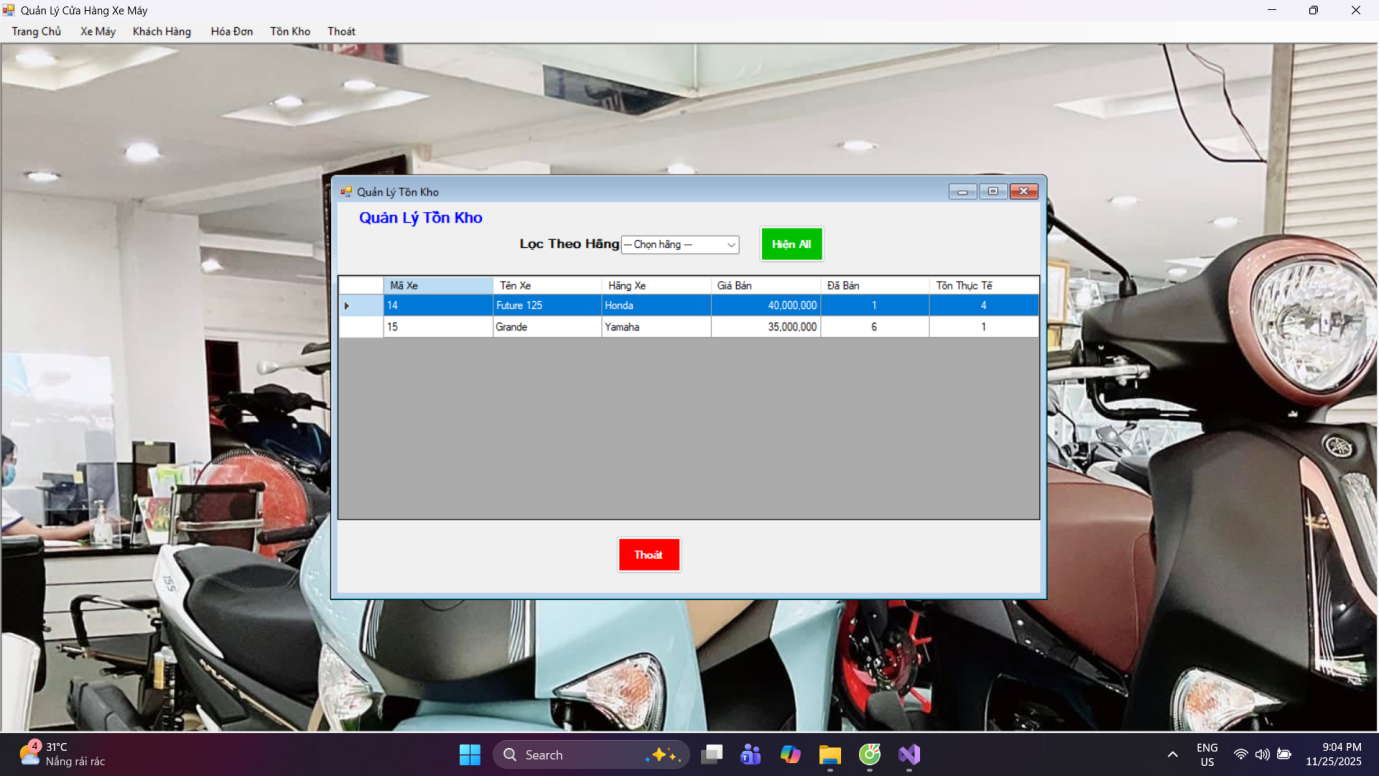
# III.Giao diện chương trình

## **A) Thiết kế giao diện**

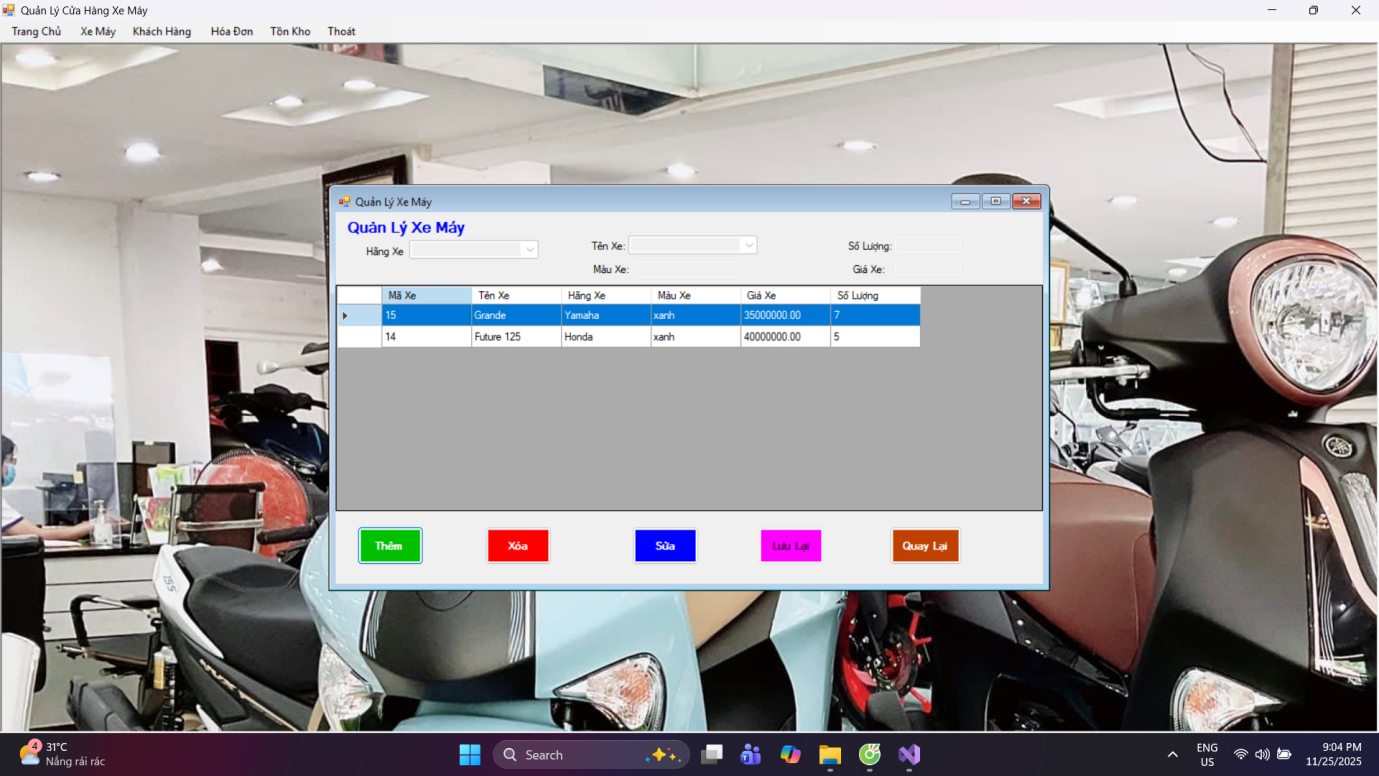
-Thêm các chức năng tiện ích như: Thêm, xóa, sửa,bỏ qua, lưu, Thoát



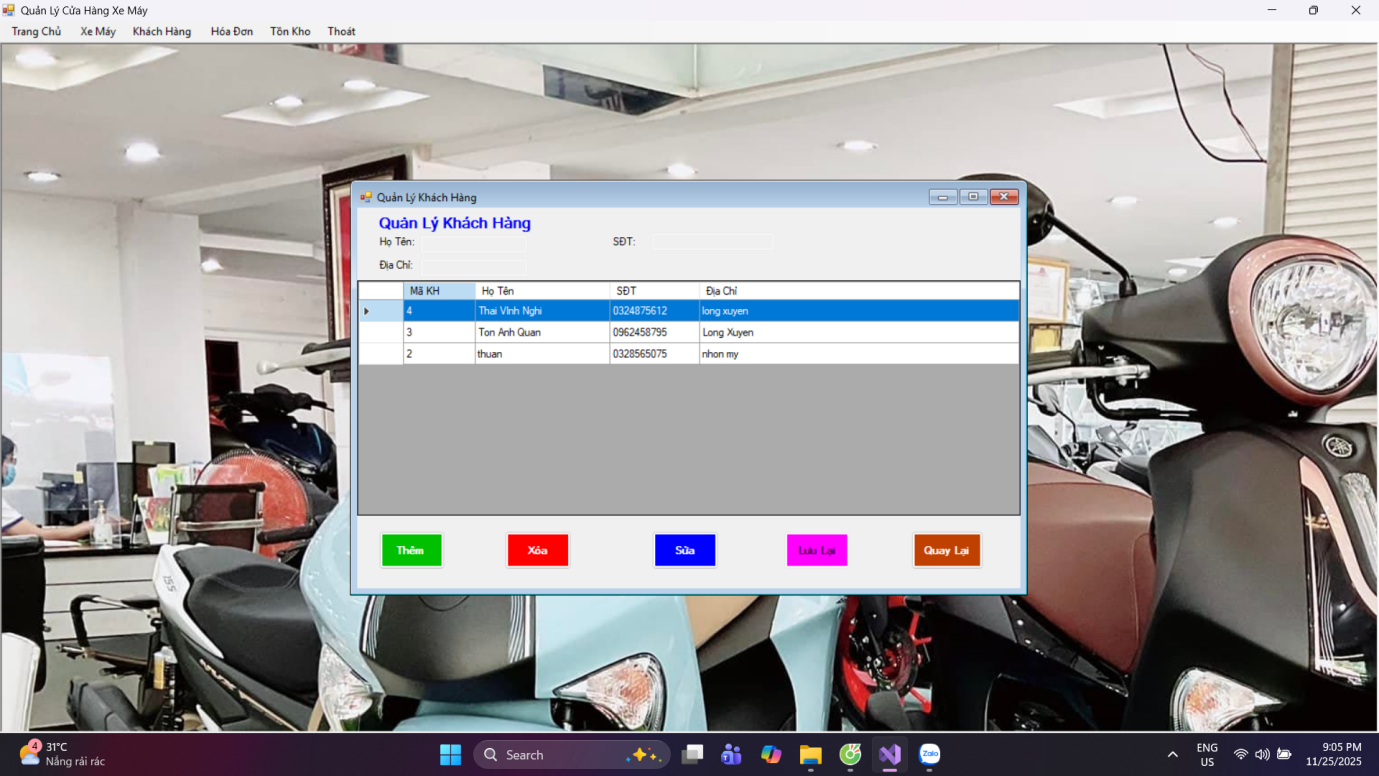
*Ảnh 1 : Giao diện của FormMain*

****

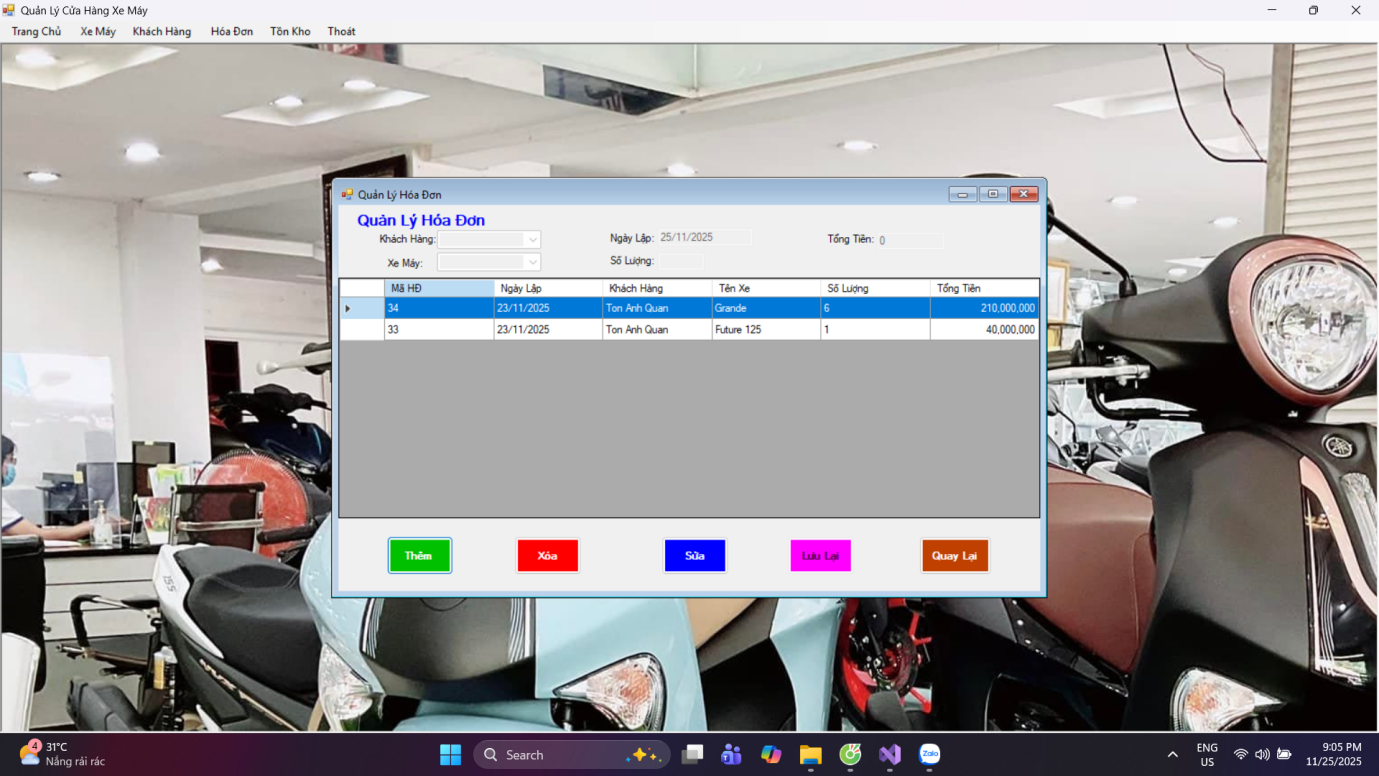
*Ảnh 2 : Giao diện của FormTonKho*

****

*Ảnh 3 : Giao diện của FormXeMay*

****

*Ảnh 4 : Giao diện của FormKhachHang*

****

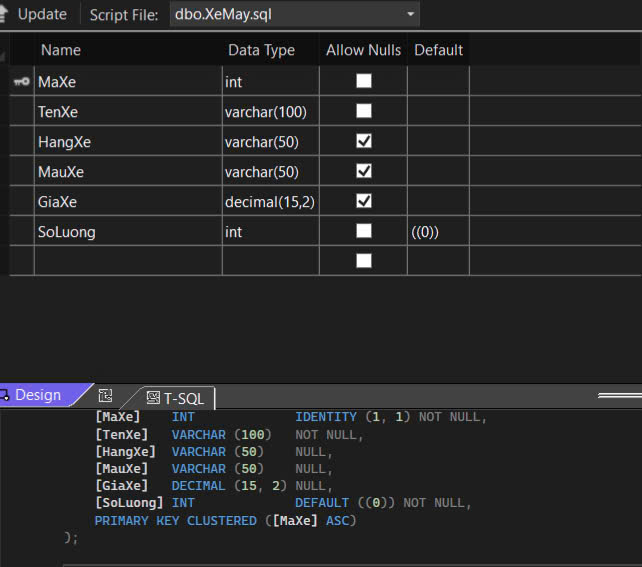
*Ảnh 5 : Giao diện của FormHoaDon*

**B) Cách xây dựng chương trình**

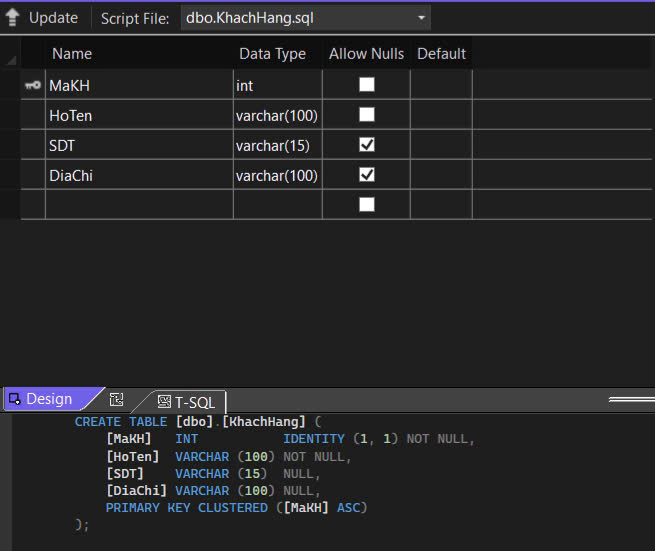
Tạo cơ sở dử liệu để sử dụng gồm tab chính như sau: FormXeMAy, FormTonKho, FormHoaDon, FormKhachHang.

### **Chuẩn bị CSDL SQL SERVER MANAGEMENT STUDIO 21\**

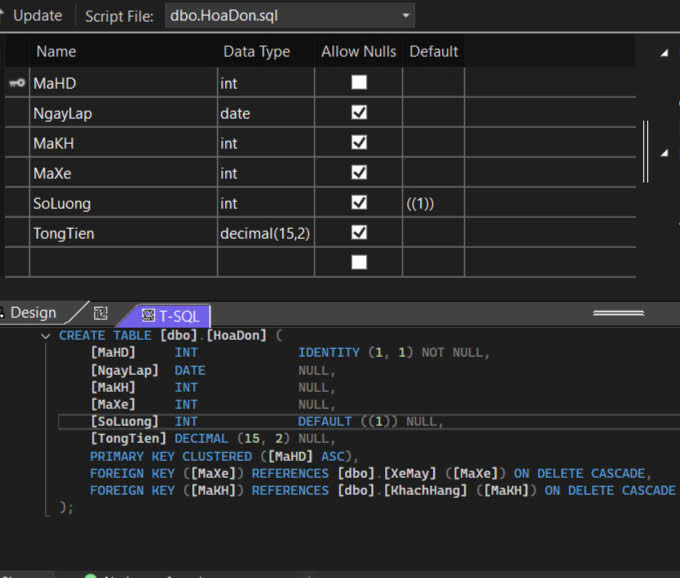
**-FormXeMay**



**- FormKhachHang :**

****

**-FormHoaDon:**

****

### **2)Viết code chương trình**

**a) Formmain:**

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;

using QuanLyCuaHangXeMay.Class;

namespace QuanLyCuaHangXeMay

{

public partial class FrmMain : Form

{

public FrmMain()

{

InitializeComponent();

}

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

{

Functions.Connect(); // Kết nối CSDL

// 1. Mở full màn hình

this.WindowState = FormWindowState.Maximized;

this.IsMdiContainer = true; // Đặt làm cha

}

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

{

FrmXeMay f = new FrmXeMay();

f.MdiParent = this; // Khai báo đây là con của Form Main

f.StartPosition = FormStartPosition.CenterScreen; // Hiện giữa màn hình

f.Show();

}

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

{

FrmKhachHang f = new FrmKhachHang();

f.MdiParent = this;

f.StartPosition = FormStartPosition.CenterScreen;

f.Show();

}

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

{

FrmHoaDon f = new FrmHoaDon();

f.MdiParent = this;

f.StartPosition = FormStartPosition.CenterScreen;

f.Show();

}

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

{

FrmTonKho f = new FrmTonKho();

f.MdiParent = this;

f.StartPosition = FormStartPosition.CenterScreen;

f.Show();

}

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

{

Functions.Disconnect();

Application.Exit();

}

}

}

**b)FormKhachHang:**

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;

using System.Data.SqlClient;

using QuanLyCuaHangXeMay.Class;

namespace QuanLyCuaHangXeMay

{

public partial class FrmKhachHang : Form

{

DataTable tblKH;

private bool themMoi = false;

private string maKHDangChon = "";

public FrmKhachHang()

{

InitializeComponent();

}

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

{

Functions.Connect(); // Kết nối CSDL

LoadDataGridView(); // Tải dữ liệu lên lưới

ResetValues(); // Xóa trắng các ô nhập

SetControlStatus(false);

}

private void LoadDataGridView()

{

string sql = "SELECT MaKH, HoTen, SDT, DiaChi FROM KhachHang ORDER BY MaKH DESC";

tblKH = Functions.GetDataToTable(sql);

dgvKhachHang.DataSource = tblKH;

// Cấu hình giao diện lưới

dgvKhachHang.AllowUserToAddRows = false;

dgvKhachHang.EditMode = DataGridViewEditMode.EditProgrammatically;

// Cài đặt chế độ chọn dòng

dgvKhachHang.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dgvKhachHang.MultiSelect = false;

// Đặt tên cột hiển thị

if (dgvKhachHang.Columns.Count >= 4)

{

dgvKhachHang.Columns[0].HeaderText = "Mã KH";

dgvKhachHang.Columns[1].HeaderText = "Họ Tên";

dgvKhachHang.Columns[2].HeaderText = "SĐT";

dgvKhachHang.Columns[3].HeaderText = "Địa Chỉ";

dgvKhachHang.Columns[0].Width = 80;

dgvKhachHang.Columns[1].Width = 150;

dgvKhachHang.Columns[2].Width = 100;

dgvKhachHang.Columns[3].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

}

}

private void ResetValues()

{

maKHDangChon = "";

txtHoTen.Text = "";

txtSdt.Text = "";

txtDiaChi.Text = "";

}

private void SetControlStatus(bool enabled)

{

// Bật/Tắt các ô nhập liệu

txtHoTen.Enabled = enabled;

txtSdt.Enabled = enabled;

txtDiaChi.Enabled = enabled;

// Bật/Tắt các nút chức năng

btnThem.Enabled = !enabled;

btnSua.Enabled = !enabled;

btnXoa.Enabled = !enabled;

// Nút Lưu và Quay Lại sáng khi đang nhập liệu

btnLuuLai.Enabled = enabled;

btnQuayLai.Enabled = true;

}

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

{

if (e.RowIndex == -1 || dgvKhachHang.CurrentRow == null) return;

if (btnThem.Enabled == false && btnSua.Enabled == false && btnLuuLai.Enabled == true) return;

try

{

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

maKHDangChon = row.Cells["MaKH"].Value.ToString();

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

txtSdt.Text = row.Cells["SDT"].Value.ToString();

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

btnSua.Enabled = true;

btnXoa.Enabled = true;

}

catch (Exception ex)

{

MessageBox.Show("Lỗi chọn dòng: " + ex.Message);

}

}

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

{

themMoi = true;

ResetValues();

SetControlStatus(true);

txtHoTen.Focus();

}

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

{

if (string.IsNullOrEmpty(maKHDangChon))

{

MessageBox.Show("Bạn chưa chọn khách hàng để xóa!");

return;

}

if (MessageBox.Show("Bạn có chắc muốn xóa khách hàng mã " + maKHDangChon + " không?",

"Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

string sql = $"DELETE FROM KhachHang WHERE MaKH = {maKHDangChon}";

Functions.RunSQL(sql);

LoadDataGridView();

ResetValues();

}

}

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

{

if (string.IsNullOrEmpty(maKHDangChon))

{

MessageBox.Show("Bạn chưa chọn khách hàng nào để sửa!");

return;

}

themMoi = false;

SetControlStatus(true);

txtHoTen.Focus();

}

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

{

if (string.IsNullOrWhiteSpace(txtHoTen.Text)) { MessageBox.Show("Bạn phải nhập tên khách hàng."); txtHoTen.Focus(); return; }

if (string.IsNullOrWhiteSpace(txtSdt.Text)) { MessageBox.Show("Bạn phải nhập số điện thoại."); txtSdt.Focus(); return; }

string sql;

if (themMoi)

{

sql = "INSERT INTO KhachHang (HoTen, SDT, DiaChi) " +

$"VALUES (N'{txtHoTen.Text}', '{txtSdt.Text}', N'{txtDiaChi.Text}')";

}

else

{

sql = "UPDATE KhachHang SET " +

$"HoTen = N'{txtHoTen.Text}', SDT = '{txtSdt.Text}', DiaChi = N'{txtDiaChi.Text}' " +

$"WHERE MaKH = {maKHDangChon}";

}

Functions.RunSQL(sql);

LoadDataGridView();

ResetValues();

SetControlStatus(false);

themMoi = false;

MessageBox.Show("Đã lưu thành công!");

}

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

{

if (btnLuuLai.Enabled == true)

{

ResetValues();

SetControlStatus(false);

themMoi = false;

}

else

{

this.Close();

}

}

}

}

**c)FormXeMay:**

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;

using System.Data.SqlClient;

using QuanLyCuaHangXeMay.Class;

namespace QuanLyCuaHangXeMay

{

public partial class FrmXeMay : Form

{

DataTable tblXM;

private bool themMoi = false;

private string maXeDangChon = "";

public FrmXeMay()

{

InitializeComponent();

}

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

{

Functions.Connect();

LoadCombos();

LoadDataGridView();

ResetValues();

SetControlStatus(false);

}

private void LoadDataGridView()

{

string sql = "SELECT MaXe, TenXe, HangXe, MauXe, GiaXe, SoLuong FROM XeMay ORDER BY MaXe DESC";

tblXM = Functions.GetDataToTable(sql);

dgvXeMay.DataSource = tblXM;

dgvXeMay.AllowUserToAddRows = false;

dgvXeMay.EditMode = DataGridViewEditMode.EditProgrammatically;

dgvXeMay.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dgvXeMay.MultiSelect = false;

if (dgvXeMay.Columns.Count > 5)

{

dgvXeMay.Columns[0].HeaderText = "Mã Xe";

dgvXeMay.Columns[1].HeaderText = "Tên Xe";

dgvXeMay.Columns[2].HeaderText = "Hãng Xe";

dgvXeMay.Columns[3].HeaderText = "Màu Xe";

dgvXeMay.Columns[4].HeaderText = "Giá Xe";

dgvXeMay.Columns[5].HeaderText = "Số Lượng";

}

}

private void LoadCombos()

{

cboHangXe.DataSource = null;

cboHangXe.Items.Clear();

cboHangXe.Items.Add("Honda");

cboHangXe.Items.Add("Yamaha");

cboHangXe.Items.Add("Suzuki");

cboHangXe.SelectedIndex = -1;

cboTenXe.DataSource = null;

cboTenXe.Items.Clear();

}

private void SetControlStatus(bool enabled)

{

cboHangXe.Enabled = enabled;

cboTenXe.Enabled = enabled;

txtMauXe.Enabled = enabled;

txtGiaXe.Enabled = enabled;

txtSoLuong.Enabled = enabled;

btnThem.Enabled = !enabled;

btnXoa.Enabled = !enabled;

btnSua.Enabled = !enabled;

btnLuuLai.Enabled = enabled;

btnQuayLai.Enabled = true;

}

private void ResetValues()

{

maXeDangChon = "";

cboHangXe.Text = "";

cboTenXe.Text = "";

txtMauXe.Text = "";

txtGiaXe.Text = "";

txtSoLuong.Text = "";

}

private void cboHangXe\_SelectedIndexChanged(object sender, EventArgs e)

{

cboTenXe.DataSource = null;

cboTenXe.Items.Clear();

cboTenXe.Text = "";

string hangChon = cboHangXe.Text.Trim();

if (hangChon == "Honda")

{

string[] xeHonda = { "Vision", "Air Blade 125/160", "Lead 125", "SH Mode", "SH 125/150i", "Winner X", "Wave Alpha", "Future 125" };

cboTenXe.Items.AddRange(xeHonda);

}

else if (hangChon == "Yamaha")

{

string[] xeYamaha = { "Exciter 155 VVA", "Grande", "Janus", "NVX 155", "Sirius", "Jupiter Finn", "Latte" };

cboTenXe.Items.AddRange(xeYamaha);

}

else if (hangChon == "Suzuki")

{

string[] xeSuzuki = { "Raider R150", "Satria F150", "Burgman Street", "Impulse 125", "GSX-R150" };

cboTenXe.Items.AddRange(xeSuzuki);

}

}

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

{

if (btnThem.Enabled == false && btnSua.Enabled == false && btnLuuLai.Enabled == true) return;

if (dgvXeMay.CurrentRow == null || dgvXeMay.CurrentRow.Index == -1) { ResetValues(); return; }

try

{

if (dgvXeMay.CurrentRow.Cells["MaXe"].Value != null)

{

maXeDangChon = dgvXeMay.CurrentRow.Cells["MaXe"].Value.ToString();

cboTenXe.Text = dgvXeMay.CurrentRow.Cells["TenXe"].Value.ToString();

cboHangXe.Text = dgvXeMay.CurrentRow.Cells["HangXe"].Value.ToString();

txtMauXe.Text = dgvXeMay.CurrentRow.Cells["MauXe"].Value.ToString();

txtGiaXe.Text = dgvXeMay.CurrentRow.Cells["GiaXe"].Value.ToString();

txtSoLuong.Text = dgvXeMay.CurrentRow.Cells["SoLuong"].Value.ToString();

}

btnSua.Enabled = true;

btnXoa.Enabled = true;

}

catch (Exception ex) { MessageBox.Show("Lỗi khi chọn dòng: " + ex.Message); }

}

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

{

dgvXeMay\_Click(sender, e);

}

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

{

themMoi = true;

ResetValues();

SetControlStatus(true);

cboHangXe.Focus();

}

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

{

if (string.IsNullOrEmpty(maXeDangChon)) { MessageBox.Show("Bạn chưa chọn bản ghi nào để sửa."); return; }

themMoi = false;

SetControlStatus(true);

cboHangXe.Focus();

}

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

{

string sql;

if (string.IsNullOrWhiteSpace(cboHangXe.Text) || string.IsNullOrWhiteSpace(cboTenXe.Text) ||

string.IsNullOrWhiteSpace(txtMauXe.Text) || string.IsNullOrWhiteSpace(txtGiaXe.Text) ||

string.IsNullOrWhiteSpace(txtSoLuong.Text))

{

MessageBox.Show("Bạn phải nhập đầy đủ thông tin."); return;

}

if (themMoi)

{

sql = "INSERT INTO XeMay (TenXe, HangXe, MauXe, GiaXe, SoLuong) " +

$"VALUES (N'{cboTenXe.Text}', N'{cboHangXe.Text}', N'{txtMauXe.Text}', " +

$"{txtGiaXe.Text.Replace(",", ".")}, {txtSoLuong.Text})";

}

else

{

if (string.IsNullOrEmpty(maXeDangChon)) return;

sql = "UPDATE XeMay SET " +

$"TenXe = N'{cboTenXe.Text}', HangXe = N'{cboHangXe.Text}', " +

$"MauXe = N'{txtMauXe.Text}', GiaXe = {txtGiaXe.Text.Replace(",", ".")}, " +

$"SoLuong = {txtSoLuong.Text} WHERE MaXe = {maXeDangChon}";

}

Functions.RunSQL(sql);

LoadDataGridView();

ResetValues();

SetControlStatus(false);

themMoi = false;

MessageBox.Show("Đã lưu dữ liệu thành công!");

}

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

{

if (string.IsNullOrEmpty(maXeDangChon)) { MessageBox.Show("Bạn chưa chọn xe để xóa!"); return; }

if (MessageBox.Show("Bạn có chắc muốn xóa xe mã " + maXeDangChon + " không?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

string sql = $"DELETE FROM XeMay WHERE MaXe = {maXeDangChon}";

Functions.RunSQL(sql);

LoadDataGridView();

ResetValues();

}

}

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

{

this.Close();

}

private void dgvXeMay\_CellContentClick\_1(object sender, DataGridViewCellEventArgs e)

{

dgvXeMay\_Click(sender, e);

}

}

}

**d)FormTonKho:**

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;

using QuanLyCuaHangXeMay.Class;

namespace QuanLyCuaHangXeMay

{

public partial class FrmTonKho : Form

{

DataTable tblTonKho;

public FrmTonKho()

{

InitializeComponent();

}

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

{

Functions.Connect();

LoadComboBoxHang();

LoadDataGridView();

}

private void LoadComboBoxHang()

{

string sql = "SELECT DISTINCT HangXe FROM XeMay";

DataTable dt = Functions.GetDataToTable(sql);

cboHangXe.DataSource = dt;

cboHangXe.DisplayMember = "HangXe";

cboHangXe.ValueMember = "HangXe";

cboHangXe.SelectedIndex = -1;

cboHangXe.Text = "--- Chọn hãng ---";

}

private void LoadDataGridView()

{

// Logic: Lấy Số lượng gốc (x.SoLuong) TRỪ ĐI Tổng số lượng đã bán (SUM(h.SoLuong))

string sql = "SELECT x.MaXe, x.TenXe, x.HangXe, x.GiaXe, " +

"ISNULL(SUM(h.SoLuong), 0) AS DaBan, " +

"(x.SoLuong - ISNULL(SUM(h.SoLuong), 0)) AS TonKho " +

"FROM XeMay x " +

"LEFT JOIN HoaDon h ON x.MaXe = h.MaXe ";

// Xử lý lọc theo hãng

if (cboHangXe.SelectedIndex != -1 && !string.IsNullOrEmpty(cboHangXe.Text) && cboHangXe.Text != "--- Chọn hãng ---")

{

sql += $"WHERE x.HangXe = N'{cboHangXe.Text}' ";

}

sql += "GROUP BY x.MaXe, x.TenXe, x.HangXe, x.GiaXe, x.SoLuong " +

"ORDER BY TonKho DESC";

tblTonKho = Functions.GetDataToTable(sql);

dgvTonKho.DataSource = tblTonKho;

FormatLuoi();

// Tính tổng và tô màu cảnh báo

int tongTonKho = 0;

foreach (DataGridViewRow row in dgvTonKho.Rows)

{

if (row.Cells[5].Value != null)

{

int tonKho = 0;

if (int.TryParse(row.Cells[5].Value.ToString(), out tonKho))

{

tongTonKho += tonKho;

if (tonKho <= 0) // Hết hàng

{

row.DefaultCellStyle.BackColor = Color.Red;

row.DefaultCellStyle.ForeColor = Color.White;

}

else if (tonKho <= 2) // Sắp hết

{

row.DefaultCellStyle.BackColor = Color.Yellow;

row.DefaultCellStyle.ForeColor = Color.Black;

}

}

}

}

}

private void FormatLuoi()

{

dgvTonKho.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dgvTonKho.MultiSelect = false;

dgvTonKho.AllowUserToAddRows = false;

dgvTonKho.EditMode = DataGridViewEditMode.EditProgrammatically;

dgvTonKho.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

if (dgvTonKho.Columns.Count >= 6)

{

dgvTonKho.Columns[0].HeaderText = "Mã Xe";

dgvTonKho.Columns[1].HeaderText = "Tên Xe";

dgvTonKho.Columns[2].HeaderText = "Hãng Xe";

dgvTonKho.Columns[3].HeaderText = "Giá Bán";

dgvTonKho.Columns[4].HeaderText = "Đã Bán";

dgvTonKho.Columns[5].HeaderText = "Tồn Thực Tế";

dgvTonKho.Columns[3].DefaultCellStyle.Format = "N0";

dgvTonKho.Columns[3].DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleRight;

}

}

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

{

this.Close();

}

private void cboLocTheoHang\_SelectedIndexChanged(object sender, EventArgs e)

{

LoadDataGridView();

}

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

{

cboHangXe.SelectedIndex = -1;

cboHangXe.Text = "--- Chọn hãng ---";

LoadDataGridView();

}

private void dgvTonKho\_CellContentClick(object sender, DataGridViewCellEventArgs e) { }

}

} }

}

**e)FormHoaDon:**

using QuanLyCuaHangXeMay.Class;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace QuanLyCuaHangXeMay

{

public partial class FrmHoaDon : Form

{

DataTable tblHD;

private bool themMoi = false;

private string maHDDangChon = "";

public FrmHoaDon()

{

InitializeComponent();

}

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

{

Functions.Connect();

LoadComboBoxKhachHang();

LoadComboBoxXeMay();

LoadDataGridView();

ResetValues();

SetControlStatus(false);

}

private void LoadDataGridView()

{

string sql = "SELECT a.MaHD, a.NgayLap, b.HoTen, c.TenXe, a.SoLuong, a.TongTien, a.MaKH, a.MaXe " +

"FROM HoaDon a " +

"JOIN KhachHang b ON a.MaKH = b.MaKH " +

"JOIN XeMay c ON a.MaXe = c.MaXe " +

"ORDER BY a.MaHD DESC";

tblHD = Functions.GetDataToTable(sql);

dgvHoaDon.DataSource = tblHD;

dgvHoaDon.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dgvHoaDon.MultiSelect = false;

dgvHoaDon.AllowUserToAddRows = false;

dgvHoaDon.EditMode = DataGridViewEditMode.EditProgrammatically;

dgvHoaDon.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

if (dgvHoaDon.Columns.Count >= 6)

{

dgvHoaDon.Columns[0].HeaderText = "Mã HĐ";

dgvHoaDon.Columns[1].HeaderText = "Ngày Lập";

dgvHoaDon.Columns[2].HeaderText = "Khách Hàng";

dgvHoaDon.Columns[3].HeaderText = "Tên Xe";

dgvHoaDon.Columns[4].HeaderText = "Số Lượng";

dgvHoaDon.Columns[5].HeaderText = "Tổng Tiền";

dgvHoaDon.Columns[6].Visible = false;

dgvHoaDon.Columns[7].Visible = false;

dgvHoaDon.Columns[5].DefaultCellStyle.Format = "N0";

dgvHoaDon.Columns[5].DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleRight;

dgvHoaDon.Columns[1].DefaultCellStyle.Format = "dd/MM/yyyy";

}

}

private void LoadComboBoxKhachHang()

{

string sql = "SELECT MaKH, HoTen FROM KhachHang";

DataTable dt = Functions.GetDataToTable(sql);

cboKhachHang.DataSource = dt;

cboKhachHang.DisplayMember = "HoTen";

cboKhachHang.ValueMember = "MaKH";

cboKhachHang.SelectedIndex = -1;

}

private void LoadComboBoxXeMay()

{

string sql = "SELECT MaXe, TenXe FROM XeMay";

DataTable dt = Functions.GetDataToTable(sql);

cboXeMay.DataSource = dt;

cboXeMay.DisplayMember = "TenXe";

cboXeMay.ValueMember = "MaXe";

cboXeMay.SelectedIndex = -1;

}

private void CapNhatTongTien()

{

if (cboXeMay.SelectedValue == null || string.IsNullOrEmpty(txtSoLuong.Text))

{

txtTongTien.Text = "0";

return;

}

try

{

int soLuong = 0;

if (!int.TryParse(txtSoLuong.Text, out soLuong)) return;

string maXe = cboXeMay.SelectedValue.ToString();

string sqlGia = "SELECT GiaXe FROM XeMay WHERE MaXe = " + maXe;

string strGia = Functions.GetFieldValues(sqlGia);

if (!string.IsNullOrEmpty(strGia))

{

double giaXe = Convert.ToDouble(strGia);

double tongTien = giaXe \* soLuong;

txtTongTien.Text = tongTien.ToString("N0");

}

}

catch { }

}

private void cboXeMay\_SelectedIndexChanged(object sender, EventArgs e)

{

CapNhatTongTien();

}

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

{

CapNhatTongTien();

}

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

{

if (e.RowIndex == -1 || dgvHoaDon.CurrentRow == null) return;

if (btnThem.Enabled == false && btnSua.Enabled == false && btnLuuLai.Enabled == true) return;

try

{

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

maHDDangChon = row.Cells[0].Value.ToString();

if (row.Cells[1].Value != DBNull.Value)

{

DateTime dt = Convert.ToDateTime(row.Cells[1].Value);

txtNgayLap.Text = dt.ToString("dd/MM/yyyy");

}

else txtNgayLap.Text = "";

cboKhachHang.SelectedValue = row.Cells[6].Value;

cboXeMay.SelectedValue = row.Cells[7].Value;

txtSoLuong.Text = row.Cells[4].Value.ToString();

txtTongTien.Text = row.Cells[5].Value.ToString();

btnSua.Enabled = true;

btnXoa.Enabled = true;

}

catch (Exception ex) { MessageBox.Show("Lỗi chọn dòng: " + ex.Message); }

}

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

{

themMoi = true;

ResetValues();

SetControlStatus(true);

txtNgayLap.Text = DateTime.Now.ToString("dd/MM/yyyy");

cboKhachHang.Focus();

}

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

{

if (string.IsNullOrEmpty(maHDDangChon)) { MessageBox.Show("Chưa chọn hóa đơn!"); return; }

themMoi = false;

SetControlStatus(true);

}

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

{

if (string.IsNullOrEmpty(maHDDangChon)) { MessageBox.Show("Chưa chọn hóa đơn!"); return; }

if (MessageBox.Show("Bạn có chắc chắn xóa hóa đơn này?", "Xác nhận", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

// Lưu ý: Khi xóa hóa đơn, đúng ra nên cộng lại số lượng vào kho (bài toán nâng cao)

string sql = $"DELETE FROM HoaDon WHERE MaHD = {maHDDangChon}";

Functions.RunSQL(sql);

LoadDataGridView();

ResetValues();

}

}

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

{

// 1. Kiểm tra dữ liệu cơ bản

if (cboKhachHang.SelectedIndex == -1) { MessageBox.Show("Chưa chọn Khách hàng"); return; }

if (cboXeMay.SelectedIndex == -1) { MessageBox.Show("Chưa chọn Xe máy"); return; }

if (string.IsNullOrEmpty(txtSoLuong.Text)) { MessageBox.Show("Chưa nhập số lượng"); return; }

if (string.IsNullOrWhiteSpace(txtNgayLap.Text)) { MessageBox.Show("Chưa nhập ngày lập"); return; }

DateTime dtNgayLap;

string[] formats = { "dd/MM/yyyy", "d/M/yyyy", "dd-MM-yyyy", "d-M-yyyy" };

if (!DateTime.TryParseExact(txtNgayLap.Text, formats, CultureInfo.InvariantCulture, DateTimeStyles.None, out dtNgayLap))

{

MessageBox.Show("Ngày tháng không hợp lệ!", "Lỗi định dạng", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtNgayLap.Focus();

return;

}

string ngayLapSQL = dtNgayLap.ToString("yyyy-MM-dd");

string tongTien = txtTongTien.Text.Replace(".", "").Replace(",", "");

if (string.IsNullOrEmpty(tongTien)) tongTien = "0";

// --- 2. LOGIC KIỂM TRA TỒN KHO BẮT ĐẦU TỪ ĐÂY ---

// Lấy số lượng khách muốn mua

int soLuongMua = 0;

if (!int.TryParse(txtSoLuong.Text, out soLuongMua) || soLuongMua <= 0)

{

MessageBox.Show("Số lượng mua phải lớn hơn 0.");

return;

}

// Lấy mã xe đang chọn

string maXe = cboXeMay.SelectedValue.ToString();

// Truy vấn CSDL để lấy Tồn kho hiện tại của xe đó

string sqlKiemTraTon = "SELECT SoLuong FROM XeMay WHERE MaXe = " + maXe;

string sTon = Functions.GetFieldValues(sqlKiemTraTon); // Hàm này trả về chuỗi

int tonKhoHienTai = 0;

if (!string.IsNullOrEmpty(sTon)) tonKhoHienTai = Convert.ToInt32(sTon);

// Kiểm tra điều kiện

if (themMoi)

{

// Trường hợp Thêm Mới: Nếu mua nhiều hơn tồn -> Báo lỗi

if (soLuongMua > tonKhoHienTai)

{

MessageBox.Show($"Trong kho chỉ còn {tonKhoHienTai} chiếc. Không đủ để bán {soLuongMua} chiếc!\nVui lòng kiểm tra lại.",

"Hết hàng", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtSoLuong.Focus();

return; // Dừng lại, không lưu

}

}

else // Trường hợp Sửa

{

// Nếu đang sửa, ta phải tính toán phức tạp hơn chút:

// Tồn kho khả dụng = Tồn kho hiện tại + Số lượng của hóa đơn cũ (trả lại kho)

string sqlLaySoLuongCu = "SELECT SoLuong FROM HoaDon WHERE MaHD = " + maHDDangChon;

int soLuongCu = Convert.ToInt32(Functions.GetFieldValues(sqlLaySoLuongCu));

if (soLuongMua > (tonKhoHienTai + soLuongCu))

{

MessageBox.Show($"Kho không đủ hàng (Kể cả khi trả lại đơn cũ)! Tổng khả dụng: {tonKhoHienTai + soLuongCu}",

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtSoLuong.Focus();

return;

}

}

// --- KẾT THÚC KIỂM TRA TỒN KHO ---

// 3. Thực hiện Lưu

string sql;

if (themMoi)

{

sql = "INSERT INTO HoaDon (NgayLap, MaKH, MaXe, SoLuong, TongTien) " +

$"VALUES ('{ngayLapSQL}', {cboKhachHang.SelectedValue}, " +

$"{cboXeMay.SelectedValue}, {txtSoLuong.Text}, {tongTien})";

// MỞ RỘNG: Nếu bạn muốn tự động trừ kho khi bán, hãy thêm lệnh này:

// Functions.RunSQL($"UPDATE XeMay SET SoLuong = SoLuong - {soLuongMua} WHERE MaXe = {maXe}");

}

else // Sửa

{

if (string.IsNullOrEmpty(maHDDangChon)) return;

sql = "UPDATE HoaDon SET " +

$"NgayLap = '{ngayLapSQL}', " +

$"MaKH = {cboKhachHang.SelectedValue}, " +

$"MaXe = {cboXeMay.SelectedValue}, " +

$"SoLuong = {txtSoLuong.Text}, " +

$"TongTien = {tongTien} " +

$"WHERE MaHD = {maHDDangChon}";

}

Functions.RunSQL(sql);

LoadDataGridView();

ResetValues();

SetControlStatus(false);

themMoi = false;

MessageBox.Show("Đã lưu hóa đơn thành công!");

}

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

{

if (btnLuuLai.Enabled)

{

ResetValues();

SetControlStatus(false);

themMoi = false;

}

else this.Close();

}

private void ResetValues()

{

maHDDangChon = "";

cboKhachHang.SelectedIndex = -1;

cboXeMay.SelectedIndex = -1;

txtSoLuong.Text = "";

txtTongTien.Text = "0";

txtNgayLap.Text = DateTime.Now.ToString("dd/MM/yyyy");

}

private void SetControlStatus(bool enabled)

{

cboKhachHang.Enabled = enabled;

cboXeMay.Enabled = enabled;

txtSoLuong.Enabled = enabled;

txtNgayLap.Enabled = enabled;

txtTongTien.Enabled = false;

btnThem.Enabled = !enabled;

btnSua.Enabled = !enabled;

btnXoa.Enabled = !enabled;

btnLuuLai.Enabled = enabled;

btnQuayLai.Enabled = true;

}

}

}

# IV. Kết luận

Phần mềm Quản lý Cửa hàng Xe máy là một ứng dụng nhỏ nhưng thực tế, giúp sinh viên rèn luyện kỹ năng lập trình giao diện, thao tác cơ sở dữ liệu và xây dựng hệ thống quản lý hoàn chỉnh.

Sinh viên có thể mở rộng hệ thống này thành phần mềm bán hàng hoặc quản lý kho chuyên nghiệp trong tương lai.

# V.Tài liệu tham khảo

- Xây dựng chương trình quản lý bán hàng bằng C#

**Link : https://timoday.edu.vn/xay-dung-chuong-trinh-quan-ly-ban-hang-bang-c/**

- "Lập trình C# WinForm - Phần mềm Quản lý quán Cafe"

**Link: https://howkteam.vn/course/lap-trinh-c-winform-phan-mem-quan-ly-quan-cafe-28**

**- Laptrinhvb.net:** Kho tàng source code của sinh viên Việt Nam.

**Link: https://laptrinhvb.net/**

**- Viblo (Cộng đồng IT Việt Nam):** Nơi chia sẻ các bài viết về kỹ thuật xử lý cụ thể (ví dụ: kết nối SQL, mô hình 3 lớp).

**Link: https://viblo.asia/search?q=quản lý bán hàng C#**