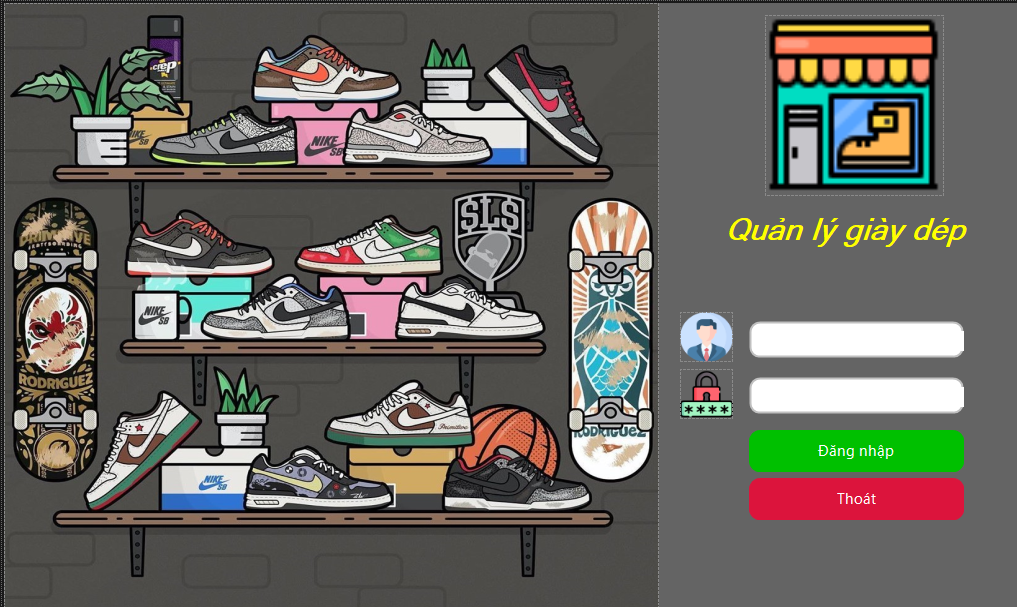
**Tên đề tài: Quản lý shop giày dép**

Các chức năng đã làm

1. **Đăng nhập**

* **Form đăng nhập**



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

{

string connectionString = "Data Source=LAPTOP-QPREIOKV;Initial Catalog=ShoesSalesManagement;Integrated Security=True";

SqlConnection conn = new SqlConnection(connectionString);

SqlCommand cmd = conn.CreateCommand();

try

{

cmd.CommandText = $"select \* from NguoiDung Where TaiKhoan=@account and MatKhau =@pass";

cmd.Parameters.Add("@account", SqlDbType.NVarChar, 1000);

cmd.Parameters.Add("@pass", SqlDbType.NVarChar, 1000);

cmd.Parameters["@account"].Value = tbUser.Text;

cmd.Parameters["@pass"].Value = tbPass.Text;

conn.Open();

SqlDataReader reader = cmd.ExecuteReader();

if (reader.Read() == true)

{

//MessageBox.Show("Đăng nhập thành công");

//Home home = new Home();

//home.ShowDialog();

if (tbUser.Text.ToLower() == "trungnguyen")

{

MessageBox.Show("Đăng nhập thành công với quyền admin");

Home home = new Home();

home.PhanQuyenAdmin();

home.ShowDialog();

}

else

{

MessageBox.Show("Đăng nhập thành công với quyền nhân viên");

Home home = new Home();

home.PhanQuyenNhanVien();

home.ShowDialog();

}

}

else

{

lblNotication.Text = "Sai tên đăng nhập hoặc mật khẩu";

}

conn.Close();

}

catch (Exception)

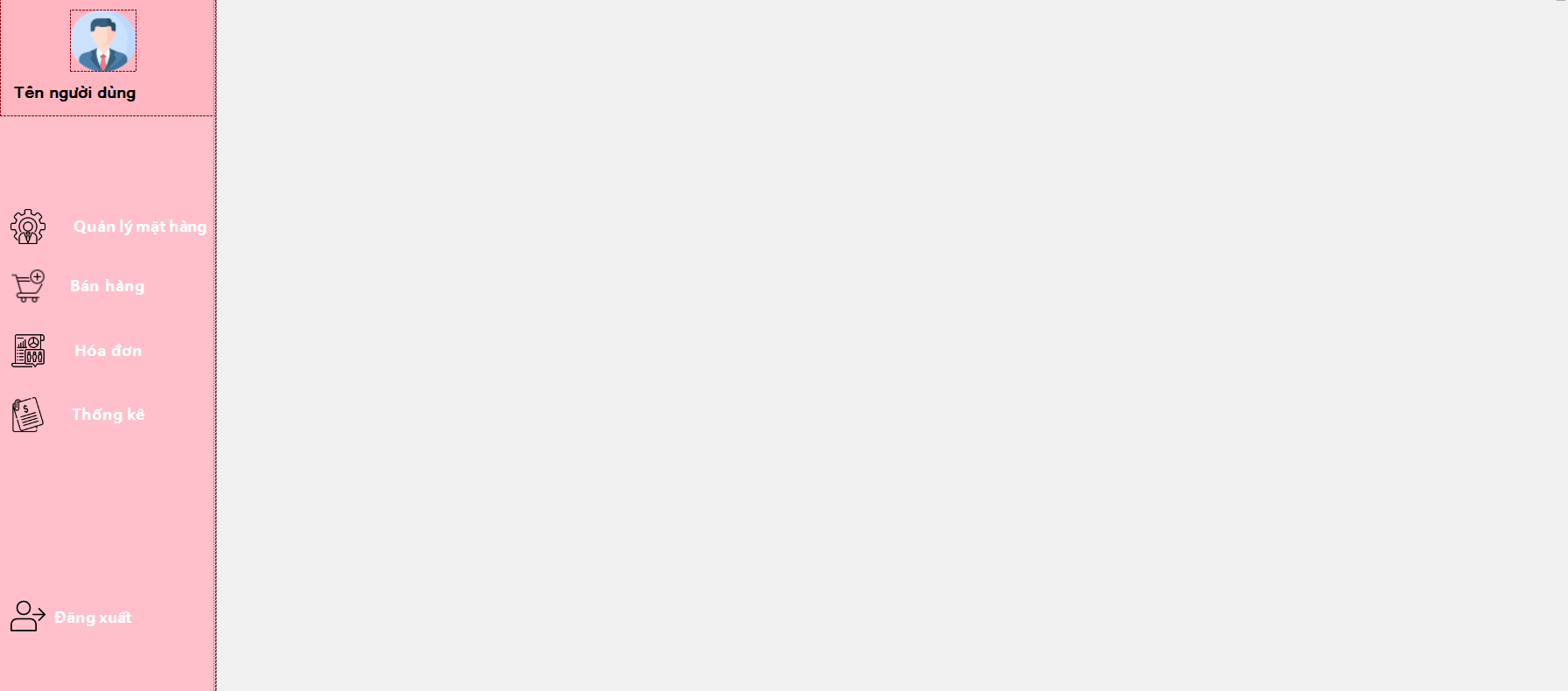
{

MessageBox.Show("Lỗi kết nối");

}

}

* **Form main**



public void PhanQuyenAdmin()

{

FullControl();

}

public void PhanQuyenNhanVien()

{

LimitConTrol();

}

public void FullControl()

{

btnBills.Enabled = true;

btnManagement.Enabled = true;

btnSale.Enabled = true;

btnStatistics.Enabled = true;

lbUserName.Text = "Nguyễn Trung Nguyên";

}

public void LimitConTrol()

{

btnBills.Enabled = true;

btnSale.Enabled = true;

btnManagement.Enabled = false;

btnStatistics.Enabled = false;

lbUserName.Text = "Nguyễn Thành Luân";

}

* **Đăng xuất**

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

{

DialogResult dialogResult = MessageBox.Show("Bạn có muốn đăng xuất không?", "Thông báo", MessageBoxButtons.OKCancel);

if (dialogResult == DialogResult.OK)

{

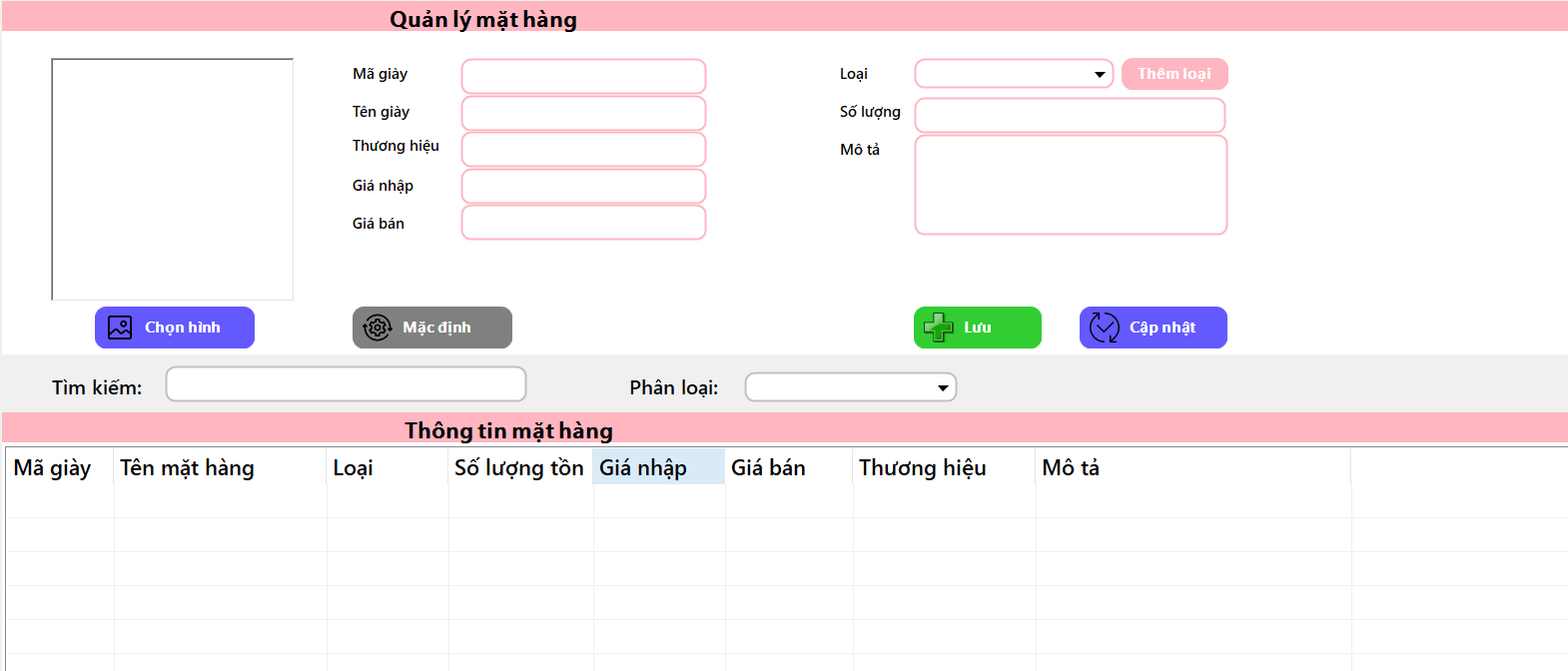
Form1 form1 = new Form1();

form1.ShowDialog();

}

}

1. **Quản lý mặt hàng**



* **Load mặt hàng**

**+ Tầng Data Access**

* **Lớp Ultilities:**

public static string MatHang\_GetAll = "MatHang\_GetAll";

public static string LoaiHang\_GetAll = "LoaiHang\_GetAll";

public static string Loai\_GetIDCha = "LoaiHang\_GetIDCha";

public static string Loai\_GetIDCon = "LoaiHang\_GetIDCon";

* **Lớp mặt hàng**

public class MatHang

{

public int ID { get; set; }

public string TenMatHang { get; set; }

public int ID\_Loai { get; set; }

public int SoLuong { get; set; }

public int GiaNhap { get; set; }

public int GiaBan { get; set; }

public string ThuongHieu { get; set; }

public byte[] HinhAnh { get; set; }

public string MoTa { get; set; }

}

* **Lớp mặt hàng DA**

public class MatHangDA

{

public List<MatHang> GetAll()

{

SqlConnection sqlConnection = new SqlConnection(Ultilities.ConnectionString);

sqlConnection.Open();

SqlCommand cmd = sqlConnection.CreateCommand();

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = Ultilities.MatHang\_GetAll;

SqlDataReader reader = cmd.ExecuteReader();

List<MatHang> listMH = new List<MatHang>();

while (reader.Read())

{

MatHang matHang = new MatHang();

matHang.ID = Convert.ToInt32(reader["ID"]);

matHang.TenMatHang = reader["TenMatHang"].ToString();

matHang.ID\_Loai = Convert.ToInt32(reader["ID\_Loai"]);

matHang.SoLuong = Convert.ToInt32(reader["SoLuongTon"]);

matHang.GiaNhap = Convert.ToInt32(reader["GiaNhap"]);

matHang.GiaBan = Convert.ToInt32(reader["GiaBan"]);

matHang.ThuongHieu = reader["ThuongHieu"].ToString();

matHang.MoTa = reader["MoTa"].ToString();

if (reader["HinhAnh"] is DBNull)

{

matHang.HinhAnh = null;

}

else

{

matHang.HinhAnh = (byte[])reader["HinhAnh"];

}

listMH.Add(matHang);

}

sqlConnection.Close();

return listMH;

}

* **Lớp Business Logic**

public class MatHangBL

{

MatHangDA matHangDA = new MatHangDA();

public List<MatHang> GetAll()

{

return matHangDA.GetAll();

}

public class LoaiHangBL

{

LoaiHangDA loaiHangDA = new LoaiHangDA();

public List<LoaiHang> GetAll()

{

return loaiHangDA.GetAll();

}

* **Load loại**

private void LoadType()

{

LoaiHangBL loaiHangBL = new LoaiHangBL();

dsLoai = loaiHangBL.GetAll();

cbbType.DataSource = dsLoai;

cbbType.ValueMember = "ID";

cbbType.DisplayMember = "TenLoai";

}

**+ Tầng giao diện**

private void LoadMatHang()

{

MatHangBL matHangBL = new MatHangBL();

dsMatHang = matHangBL.GetAll();

int count = 1;

lvMatHang.Items.Clear();

foreach (var giay in dsMatHang)

{

ListViewItem item = lvMatHang.Items.Add(count.ToString());

item.SubItems.Add(giay.TenMatHang);

item.SubItems.Add(giay.ID\_Loai.ToString());

item.SubItems.Add(giay.SoLuong.ToString());

item.SubItems.Add(giay.GiaNhap.ToString());

item.SubItems.Add(giay.GiaBan.ToString());

item.SubItems.Add(giay.ThuongHieu);

item.SubItems.Add(giay.MoTa);

count++;

}

}

* **Thêm mặt hàng**

**+ Tầng giao diện**

* **Lớp Ultinities**

public static string MatHang\_InsertUpdateDelete = "MatHang\_InsertUpdateDelete";

* **Lớp mặt hàng DA**

public int Insert\_Update\_Delete(MatHang matHang, int action)

{

SqlConnection conn = new SqlConnection(Ultilities.ConnectionString);

conn.Open();

SqlCommand cmd = conn.CreateCommand();

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = Ultilities.MatHang\_InsertUpdateDelete;

SqlParameter parameter = new SqlParameter("@ID", SqlDbType.Int);

parameter.Direction = ParameterDirection.InputOutput;

cmd.Parameters.Add(parameter).Value = matHang.ID;

cmd.Parameters.Add("@TenMatHang", SqlDbType.NVarChar, 1000).Value = matHang.TenMatHang;

cmd.Parameters.Add("@ID\_Loai", SqlDbType.Int).Value = matHang.ID\_Loai;

cmd.Parameters.Add("@SoLuongTon", SqlDbType.Int).Value = matHang.SoLuong;

cmd.Parameters.Add("@GiaNhap", SqlDbType.Int).Value = matHang.GiaNhap;

cmd.Parameters.Add("@GiaBan", SqlDbType.Int).Value = matHang.GiaBan;

cmd.Parameters.Add("@ThuongHieu", SqlDbType.NVarChar, 1000).Value = matHang.ThuongHieu;

cmd.Parameters.Add("@HinhAnh", SqlDbType.Image).Value = matHang.HinhAnh;

cmd.Parameters.Add("@MoTa", SqlDbType.NVarChar, 3000).Value = matHang.MoTa;

cmd.Parameters.Add("@Action", SqlDbType.Int).Value = action;

int result = cmd.ExecuteNonQuery();

if (result > 0)

return (int)cmd.Parameters["@ID"].Value;

return 0;

}

**+ Tầng Business Logic**

public int Insert(MatHang matHang)

{

return matHangDA.Insert\_Update\_Delete(matHang, 0);

}

public int Update(MatHang matHang)

{

return matHangDA.Insert\_Update\_Delete(matHang, 1);

}

public int Delete(MatHang matHang)

{

return matHangDA.Insert\_Update\_Delete(matHang, 2);

}

**+ Tầng giao diện**

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

{

btnSave.Enabled = false;

for (int i = 0; i < lvMatHang.Items.Count; i++)

{

if (lvMatHang.Items[i].Selected)

{

mhCurrent = dsMatHang[i];

tbID.Text = mhCurrent.ID.ToString();

tbName.Text = mhCurrent.TenMatHang;

tbQuantity.Text = mhCurrent.SoLuong.ToString();

tbThuongHieu.Text = mhCurrent.ThuongHieu;

tbGiaNhap.Text = mhCurrent.GiaNhap.ToString();

tbGiaBan.Text = mhCurrent.GiaBan.ToString();

cbbType.SelectedValue = mhCurrent.ID\_Loai;

tbNotes.Text = mhCurrent.MoTa;

if (mhCurrent.HinhAnh == null)

{

pbPicture.Image = null;

}

else

{

MemoryStream ms = new MemoryStream(mhCurrent.HinhAnh);

pbPicture.Image = Image.FromStream(ms);

}

}

}

}

public int InsertMatHang()

{

MatHang matHang = new MatHang();

matHang.ID = 0;

if (tbName.Text == "" || tbQuantity.Text == "" || tbThuongHieu.Text == "" || tbGiaNhap.Text == "" ||

tbGiaBan.Text == "" || cbbType.Text == "")

{

MessageBox.Show("Yêu cầu nhập đầy đủ thông tin");

}

else

{

int giaban = 0;

try

{

giaban = int.Parse(tbGiaBan.Text);

}

catch

{

giaban = 0;

}

int gianhap = 0;

try

{

gianhap = int.Parse(tbGiaNhap.Text);

}

catch

{

gianhap = 0;

}

int soluong = 0;

try

{

soluong = int.Parse(tbQuantity.Text);

}

catch

{

soluong = 0;

}

matHang.TenMatHang = tbName.Text;

matHang.ID\_Loai = int.Parse(cbbType.SelectedValue.ToString());

matHang.SoLuong = soluong;

matHang.GiaNhap = gianhap;

matHang.GiaBan = giaban;

matHang.ThuongHieu = tbThuongHieu.Text;

matHang.MoTa = tbNotes.Text;

if (matHang.HinhAnh == null)

{

FileStream file = new FileStream(imgLocation, FileMode.Open, FileAccess.Read);

BinaryReader binary = new BinaryReader(file);

img = binary.ReadBytes((int)file.Length);

matHang.HinhAnh = img;

}

MatHangBL matHangBL = new MatHangBL();

return matHangBL.Insert(matHang);

}

return -1;

}

* **Cập nhật mặt hàng**

**+ Tầng giao diện**

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

{

int update = UpdateMatHang();

if (update > 0)

{

MessageBox.Show("Cập nhật dữ liệu thành công");

LoadMatHang();

}

else

{

MessageBox.Show("Không có gì để cập nhật");

}

}

public int UpdateMatHang()

{

MatHang matHang = mhCurrent;

if (tbName.Text == "" || tbQuantity.Text == "" || tbThuongHieu.Text == "" || tbGiaNhap.Text == "" ||

tbGiaBan.Text == "" || cbbType.Text == "")

{

MessageBox.Show("Yêu cầu nhập đẩy đủ thông tin");

}

else

{

int giaban = 0;

try

{

giaban = int.Parse(tbGiaBan.Text);

}

catch

{

giaban = 0;

}

int gianhap = 0;

try

{

gianhap = int.Parse(tbGiaNhap.Text);

}

catch

{

gianhap = 0;

}

int soluong = 0;

try

{

soluong = int.Parse(tbQuantity.Text);

}

catch

{

soluong = 0;

}

matHang.TenMatHang = tbName.Text;

matHang.ID\_Loai = int.Parse(cbbType.SelectedValue.ToString());

matHang.SoLuong = soluong;

matHang.GiaNhap = gianhap;

matHang.GiaBan = giaban;

matHang.ThuongHieu = tbThuongHieu.Text;

matHang.MoTa = tbNotes.Text;

if (matHang.HinhAnh != null)

{

matHang.HinhAnh = mhCurrent.HinhAnh;

}

else

{

FileStream file = new FileStream(imgLocation, FileMode.Open, FileAccess.Read);

BinaryReader binary = new BinaryReader(file);

img = binary.ReadBytes((int)file.Length);

matHang.HinhAnh = img;

}

MatHangBL matHangBL = new MatHangBL();

return matHangBL.Update(matHang);

}

return -1;

}

* **Xóa mặt hàng**

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

{

if (MessageBox.Show("Bạn có chắc muốn xóa đồ chơi này không?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)

{

MatHangBL matHangBL = new MatHangBL();

if (matHangBL.Delete(mhCurrent) > 0)

{

MessageBox.Show("Xóa thành công", "Thông báo");

LoadMatHang();

}

else

{

MessageBox.Show("Vui lòng thử lại", "Lỗi");

}

}

}

* **Nút mặc định**

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

{

btnSave.Enabled = true;

tbID.Text = "";

tbName.Text = "";

tbQuantity.Text = "";

tbThuongHieu.Text = "";

tbGiaNhap.Text = "";

tbGiaBan.Text = "";

tbNotes.Text = "";

cbbType.SelectedIndex = -1;

pbPicture.Hide();

}

* **Nút chọn hình**

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

{

OpenFileDialog dlg = new OpenFileDialog();

dlg.Title = "Open File Image";

dlg.FileName = "Hãy chọn File";

dlg.Filter = "Image Files (JPEG, GIF, BMP, etc.)|"

+ ".jpg;.jpeg;\*.gif;\*.bmp;"

+ ".tif;.tiff;\*.png|"

+ "JPEG files (.jpg;.jpeg)|\*.jpg;\*.jpeg|"

+ "GIF files (.gif)|.gif|"

+ "BMP files (.bmp)|.bmp|"

+ "TIFF files (.tif;.tiff)|\*.tif;\*.tiff|"

+ "PNG files (.png)|.png|"

+ "All files (.)|\*.\*";

dlg.InitialDirectory = Environment.CurrentDirectory;

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

{

imgLocation = dlg.FileName.ToString();

pbPicture.ImageLocation = imgLocation;

}

}

* **Click vào listview load lên control**

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

{

btnSave.Enabled = false;

for (int i = 0; i < lvMatHang.Items.Count; i++)

{

if (lvMatHang.Items[i].Selected)

{

mhCurrent = dsMatHang[i];

tbID.Text = mhCurrent.ID.ToString();

tbName.Text = mhCurrent.TenMatHang;

tbQuantity.Text = mhCurrent.SoLuong.ToString();

tbThuongHieu.Text = mhCurrent.ThuongHieu;

tbGiaNhap.Text = mhCurrent.GiaNhap.ToString();

tbGiaBan.Text = mhCurrent.GiaBan.ToString();

cbbType.SelectedValue = mhCurrent.ID\_Loai;

tbNotes.Text = mhCurrent.MoTa;

if (mhCurrent.HinhAnh == null)

{

pbPicture.Image = null;

}

else

{

MemoryStream ms = new MemoryStream(mhCurrent.HinhAnh);

pbPicture.Image = Image.FromStream(ms);

}

}

}

}

* **Chức năng phân loại**

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

{

if (cbbPhanLoai.SelectedIndex == -1)

LoadMatHang();

else

{

int ID = 0;

Int32.TryParse(cbbPhanLoai.SelectedValue.ToString(), out ID);

if (ID == 0)

{

return;

}

else

{

var matHangs = dsMatHang.FindAll(x => x.ID\_Loai == ID).ToList();

LoadMatHang(matHangs);

}

}

}

private void LoadMatHang(List<MatHang> matHangs = null)

{

if (dsMatHang == null)

{

matHangs = dsMatHang;

}

int count = 1;

lvMatHang.Items.Clear();

foreach (var giay in matHangs)

{

ListViewItem item = lvMatHang.Items.Add(count.ToString());

item.SubItems.Add(giay.TenMatHang);

item.SubItems.Add(giay.ID\_Loai.ToString());

item.SubItems.Add(giay.SoLuong.ToString());

item.SubItems.Add(giay.GiaNhap.ToString());

item.SubItems.Add(giay.GiaBan.ToString());

item.SubItems.Add(giay.ThuongHieu);

item.SubItems.Add(giay.MoTa);

count++;

}

}

* **Tìm kiếm**

**+ Tầng Business Logic**

public List<MatHang> Search(string key)

{

List<MatHang> list = GetAll();

List<MatHang> result = new List<MatHang>();

foreach (var item in list)

{

if (item.GiaBan.ToString().Contains(key)

|| item.ID.ToString().Contains(key)

|| item.TenMatHang.Contains(key)

|| item.SoLuong.ToString().Contains(key)

|| item.ThuongHieu.Contains(key)

|| item.GiaBan.ToString().Contains(key))

result.Add(item);

}

return result;

}

**+ Tầng giao diện**

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

{

List<MatHang> dsMatHang = new List<MatHang>();

lvMatHang.Items.Clear();

MatHangBL matHangBL = new MatHangBL();

dsMatHang = matHangBL.Search(tbSearch.Text);

foreach (MatHang mathang in dsMatHang)

{

ListViewItem item = new ListViewItem(mathang.ID.ToString());

item.SubItems.Add(mathang.TenMatHang);

item.SubItems.Add(mathang.ID\_Loai.ToString());

item.SubItems.Add(mathang.SoLuong.ToString());

item.SubItems.Add(mathang.GiaNhap.ToString());

item.SubItems.Add(mathang.GiaBan.ToString());

item.SubItems.Add(mathang.ThuongHieu);

lvMatHang.Items.Add(item);

}

}

* **Form loại**

**+ Load loại lên listview**

public void LoadTypeToLV()

{

LoaiHangBL loaiHangBL = new LoaiHangBL();

dsLoaiHang = loaiHangBL.GetAll();

int count = 1;

lvType.Items.Clear();

foreach (var loai in dsLoaiHang)

{

ListViewItem item = lvType.Items.Add(count.ToString());

item.SubItems.Add(loai.TenLoai);

item.SubItems.Add(loai.ID\_Cha.ToString());

item.SubItems.Add(loai.MoTa);

count++;

}

}

**+ Load loại trên combobox**

private void LoadType()

{

LoaiHangBL loaiHangBL = new LoaiHangBL();

dsLoaiHang = loaiHangBL.GetAll();

cbbIDCha.DataSource = dsLoaiHang;

cbbIDCha.ValueMember = "ID";

cbbIDCha.DisplayMember = "ID\_Cha";

}

**+ Click vào listview load lên control**

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

{

btnAdd.Enabled = false;

for (int i = 0; i < lvType.Items.Count; i++)

{

if (lvType.Items[i].Selected)

{

typeCurrent = dsLoaiHang[i];

tbID.Text = typeCurrent.ID.ToString();

tbNameType.Text = typeCurrent.TenLoai;

cbbIDCha.Text = typeCurrent.ID\_Cha.ToString();

tbNotes.Text = typeCurrent.MoTa;

}

}

}

**+ Xóa loại**

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

{

if (MessageBox.Show("Bạn có chắc muốn xóa đồ chơi này không?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)

{

LoaiHangBL loaiHangBL = new LoaiHangBL();

if (loaiHangBL.Delete(typeCurrent) > 0)

{

MessageBox.Show("Xóa thành công", "Thông báo");

LoadTypeToLV();

}

else

{

MessageBox.Show("Vui lòng thử lại", "Lỗi");

}

}

}

**+ Thêm loại**

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

{

int Insert = InsertType();

if (Insert > 0)

{

MessageBox.Show("Thêm dữ liệu thành công");

LoadTypeToLV();

}

else

{

MessageBox.Show("Vui lòng kiểm tra lại");

}

}

public int InsertType()

{

LoaiHang loai = new LoaiHang();

loai.ID = 0;

if (tbNameType.Text == "")

{

MessageBox.Show("Yêu cầu nhập đầy đủ thông tin");

LoadTypeToLV();

}

else

{

loai.TenLoai = tbNameType.Text;

loai.ID\_Cha = int.Parse(cbbIDCha.SelectedValue.ToString());

loai.MoTa = tbNotes.Text;

LoaiHangBL loaiHangBL = new LoaiHangBL();

return loaiHangBL.Insert(loai);

}

return -1;

}

**+ Cập nhật loại**

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

{

btnAdd.Enabled = true;

int Update = UpdateType();

if (Update > 0)

{

MessageBox.Show("Cập nhật dữ liệu thành công");

LoadTypeToLV();

}

else

{

MessageBox.Show("Vui lòng kiểm tra lại");

}

}

public int UpdateType()

{

LoaiHang loai = typeCurrent;

if (tbNameType.Text == "")

{

MessageBox.Show("Yêu cầu nhập đầy đủ thông tin");

LoadTypeToLV();

}

else

{

loai.TenLoai = tbNameType.Text;

loai.ID\_Cha = int.Parse(cbbIDCha.SelectedValue.ToString());

loai.MoTa = tbNotes.Text;

LoaiHangBL loaiHangBL = new LoaiHangBL();

return loaiHangBL.Update(loai);

}

return -1;

}

* **Form bán hàng**

**+ Load mặt hàng (tầng giao diện)**

private void LoadMatHang()

{

MatHangBL matHangBL = new MatHangBL();

dsMatHang = matHangBL.GetAll();

int count = 1;

lvMatHang.Items.Clear();

foreach (var giay in dsMatHang)

{

ListViewItem item = lvMatHang.Items.Add(count.ToString());

item.SubItems.Add(giay.TenMatHang);

item.SubItems.Add(giay.ID\_Loai.ToString());

item.SubItems.Add(giay.GiaBan.ToString());

count++;

}

}

**+ Load người dùng (Tầng Data Access)**

* **Lớp người dùng**

public class NguoiDung

{

public int ID { get; set; }

public string TenNhanVien { get; set; }

public string SoDienThoai { get; set; }

public string DiaChi { get; set; }

public string TaiKhoan { get; set; }

public string MatKhau { get; set; }

public bool TrangThai { get; set; }

}

* **Lớp người dùng DA**

public List<NguoiDung> GetAll()

{

SqlConnection conn = new SqlConnection(Ultilities.ConnectionString);

SqlCommand cmd = conn.CreateCommand();

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = Ultilities.NguoiDung\_GetAll;

conn.Open();

SqlDataReader reader = cmd.ExecuteReader();

List<NguoiDung> listNguoiDung = new List<NguoiDung>();

while (reader.Read())

{

NguoiDung nguoiDung = new NguoiDung();

nguoiDung.ID = Convert.ToInt32(reader["ID"]);

nguoiDung.TenNhanVien = reader["TenNhanVien"].ToString();

nguoiDung.SoDienThoai = reader["SoDienThoai"].ToString();

nguoiDung.DiaChi = reader["DiaChi"].ToString();

nguoiDung.TaiKhoan = reader["TaiKhoan"].ToString();

nguoiDung.MatKhau = reader["MatKhau"].ToString();

nguoiDung.TrangThai = bool.Parse(reader["TrangThai"].ToString());

listNguoiDung.Add(nguoiDung);

}

return listNguoiDung;

}

* **Lớp Ultinities**

public static string NguoiDung\_GetAll = "NguoiDung\_GetAll";

* **Tầng giao diện**

private void LoadNguoiDung()

{

NguoiDungBL nguoiDungBL = new NguoiDungBL();

dsNguoiDung = nguoiDungBL.GetAll();

cbbNhanVien.DataSource = dsNguoiDung;

cbbNhanVien.DisplayMember = "TenNhanVien";

cbbNhanVien.ValueMember = "ID";

}

**+ Tìm kiếm**

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

{

List<MatHang> dsMatHang = new List<MatHang>();

lvMatHang.Items.Clear();

MatHangBL matHangBL = new MatHangBL();

dsMatHang = matHangBL.Search(tbSearch.Text);

foreach (MatHang mathang in dsMatHang)

{

ListViewItem item = new ListViewItem(mathang.ID.ToString());

item.SubItems.Add(mathang.TenMatHang);

item.SubItems.Add(mathang.ID\_Loai.ToString());

item.SubItems.Add(mathang.GiaBan.ToString());

lvMatHang.Items.Add(item);

}

}

* **Hóa đơn**

private void loadKhachHang()

{

HoaDonBL hoaDonBL = new HoaDonBL();

dsHoaDon = hoaDonBL.GetAll();

int count = 1;

lvThongTin.Items.Clear();

foreach (var item in dsHoaDon)

{

ListViewItem list = lvThongTin.Items.Add(count.ToString());

list.SubItems.Add(item.TenKhachHang);

list.SubItems.Add(item.SoDienThoai);

count++;

}

}

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

{

loadKhachHang();

}

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

{

List<HoaDon> dsHoaDon = new List<HoaDon>();

lvThongTin.Items.Clear();

HoaDonBL hoaDonBL = new HoaDonBL();

dsHoaDon = hoaDonBL.Search(tbSearch.Text);

foreach (HoaDon hoaDon in dsHoaDon)

{

ListViewItem item = new ListViewItem(hoaDon.ID.ToString());

item.SubItems.Add(hoaDon.TenKhachHang);

item.SubItems.Add(hoaDon.SoDienThoai);

lvThongTin.Items.Add(item);

}

}

private void lvThongTin\_DoubleClick(object sender, EventArgs e)

{

FormChiTietHoaDon formChiTietHoa = new FormChiTietHoaDon();

formChiTietHoa.ShowDialog();

string IDHoaDon = lvThongTin.SelectedItems[0].ToString();

formChiTietHoa.loadBillsDetails(IDHoaDon);

}

* Business Logic

public class HoaDonBL

{

HoaDonDA HoaDonDA = new HoaDonDA();

public List<HoaDon> GetAll()

{

return HoaDonDA.GetAll();

}

public List<HoaDon> Search(string key)

{

List<HoaDon> dsHoaDon = GetAll();

List<HoaDon> result = new List<HoaDon>();

foreach (var item in dsHoaDon)

{

if (item.TenKhachHang.ToString().Contains(key)

|| item.SoDienThoai.Contains(key))

result.Add(item);

}

return result;

}

}

* **Data Access**

public List<HoaDon> GetAll()

{

SqlConnection conn = new SqlConnection(Ultilities.ConnectionString);

SqlCommand cmd = conn.CreateCommand();

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = Ultilities.HoaDon\_GetAll;

conn.Open();

SqlDataReader reader = cmd.ExecuteReader();

List<HoaDon> dsHoaDon = new List<HoaDon>();

while (reader.Read())

{

HoaDon hoaDon = new HoaDon();

hoaDon.ID = Convert.ToInt32(reader["ID"]);

hoaDon.NgayBanHang = Convert.ToDateTime(reader["NgayBanHang"]);

hoaDon.TenKhachHang = reader["TenKhachHang"].ToString();

hoaDon.SoDienThoai = reader["SoDienThoai"].ToString();

hoaDon.ID\_NhanVien = Convert.ToInt32(reader["ID\_NhanVien"]);

hoaDon.KhuyenMai = Convert.ToInt32(reader["KhuyenMai"]);

dsHoaDon.Add(hoaDon);

}

conn.Close();

return dsHoaDon;

}

* Ultinities

public static string HoaDon\_GetAll = "HoaDon\_GetAll";

public static string ChiTietHoaDon\_GetAll = "ChiTietHoaDon\_GetAll";

* Lớp hóa đơn

public class HoaDon

{

public int ID { get; set; }

public DateTime NgayBanHang { get; set; }

public string TenKhachHang { get; set; }

public string SoDienThoai { get; set; }

public int ID\_NhanVien { get; set; }

public float KhuyenMai { get; set; }

}