|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| Báo cáo Đồ án WPF  Đề tài: Quản lý bán sách |
| |  |  |  | | --- | --- | --- | | Trần Hồng Phong | 4/15/21 | Lập trình Window 3 | |

MỤC BÁO CÁO: BẢNG NHÂN VIÊN

# Dữ liệu các bảng có liên quan

Create database QL\_BANSACH

GO

USE QL\_BANSACH

GO

SET DATEFORMAT dmy

Create table dbo.[CT\_HOADON]

(

[SoHD] VARCHAR(5) NOT NULL,

[MaSach] VARCHAR(5) NOT NULL,

[SoLuong] Int check ([SoLuong]>0) NOT NULL,

Primary Key ([MaSach],[SoHD])

)

go

Create table dbo.[PHIEUNHAPSACH]

(

[SoPNS] VARCHAR(5) NOT NULL,

[MaNV] VARCHAR(5) NOT NULL,

[NgayNhap] Date NULL,

Primary Key ([SoPNS])

)

go

Create table dbo.[HOADON]

(

[SoHD] VARCHAR(5) NOT NULL,

[MaKhachHang] VARCHAR(5) NOT NULL,

[MaNV] VARCHAR(5) NOT NULL,

[NgayHoaDon] Date NULL,

[TrangThai] nvarchar(40) check ([TrangThai] IN(N'Đã thanh toán',N'Chưa thanh toán')) NOT NULL,

Primary Key ([SoHD])

)

go

Create table dbo.[NHAXUATBAN]

(

[MaNhaXuatBan] VARCHAR(5) NOT NULL,

[TenNhaXuatBan] nvarchar(70) NOT NULL,

[DiaChiNXB] nvarchar(70) NULL,

[DienThoai] nvarchar(70) NULL,

Primary Key ([MaNhaXuatBan])

)

go

Create table dbo.[SACH]

(

[MaSach] VARCHAR(5) NOT NULL,

[MaTacGia] VARCHAR(5) NOT NULL,

[MaTheLoai] VARCHAR(5) NOT NULL,

[MaNhaXuatBan] VARCHAR(5) NOT NULL,

[MaKM] VARCHAR(5) NOT NULL,

Primary Key ([MaSach])

)

go

Create table dbo.[THELOAI]

(

[MaTheLoai] VARCHAR(5) NOT NULL,

[TenTheLoai] nVarchar(100) NOT NULL,

Primary Key ([MaTheLoai])

)

go

Create table dbo.[NHANVIEN]

(

[MaNV] VARCHAR(5) NOT NULL,

[UserName] varchar(25) NOT NULL,

[MaChucVu] VARCHAR(5) NOT NULL,

[TenNV] nvarchar(100) NOT NULL,

[DiaChi] nvarchar(100) NOT NULL,

[SoDienThoai] int NOT NULL,

[CMND] varChar(11) NOT NULL,

[Phai] nvarchar(7) check ([Phai] IN(N'Nam',N'Nữ')) NOT NULL,

[NgaySinh] date NULL,

Primary Key ([MaNV])

)

go

Create table dbo.[TACGIA]

(

[MaTacGia] VARCHAR(5) NOT NULL,

[TenTacGia] nvarchar(70) NOT NULL,

[NgaySinh] date NULL,

[QueQuan] nvarchar(70) NULL,

[Phai] nvarchar(7) check ([Phai] IN(N'Nam',N'Nữ')) NOT NULL,

Primary Key ([MaTacGia])

)

go

Create table dbo.[ChucVu]

(

[MaChucVu] VARCHAR(5) NOT NULL,

[TenChucVu] nvarchar(50) NOT NULL,

Primary Key ([MaChucVu])

)

go

Create table dbo.[TAIKHOAN]

(

[UserName] varchar(25) NOT NULL,

[MaQuyen] VARCHAR(5) NOT NULL,

[Pass] varChar(20) NOT NULL,

Primary Key ([UserName])

)

go

Create table dbo.[KHACHHANG]

(

[MaKhachHang] VARCHAR(5) NOT NULL,

[TenKhachHang] NVARCHAR(70) NOT NULL,

[Phai] nvarchar(7) check ([Phai] IN(N'Nam',N'Nữ')) NOT NULL,

[DiaChi] NVARCHAR(70) NOT NULL,

[SDT] int NOT NULL,

[Email] VARCHAR(50) NULL,

[NgaySinh] date NULL,

Primary Key ([MaKhachHang])

)

go

Create table dbo.[PHANQUYEN]

(

[MaQuyen] VARCHAR(5) NOT NULL,

[TenQuyen] VARCHAR(15) NOT NULL,

[Mota] NVARCHAR(50) NULL,

Primary Key ([MaQuyen])

)

go

Create table dbo.[CHITIETPHIEUNHAP]

(

[SoPNS] VARCHAR(5) NOT NULL,

[MaSach] VARCHAR(5) NOT NULL,

[SoLuongNhap] int check ([SoLuongNhap]>0) NOT NULL,

[DonGiaNhap] DECIMAL NOT NULL,

Primary Key ([SoPNS],[MaSach])

)

go

Create table dbo.[KHUYENMAI]

(

[MaKM] VARCHAR(5) NOT NULL,

[TenChuongTrinh] NVARCHAR(70) NOT NULL,

[GiamGia] int check ([GiamGia]>=0 and 100>=[GiamGia]) NOT NULL,

Primary Key ([MaKM])

)

go

Create table dbo.[CHITIETSACH]

(

[MaSach] VARCHAR(5) NOT NULL,

[TenSach] NVARCHAR(70) NOT NULL,

[DonGiaBan] DECIMAL NOT NULL,

[SoLuongTon] int check ([SoLuongTon]>0) NOT NULL,

[LanTaiBan] int check ([LanTaiBan]>=1) NOT NULL,

[SoTrang] smallint check ([SoTrang]>=1 and 2000>=[SoTrang]) NOT NULL,

[LoaiBia] nvarchar(15) check ([LoaiBia] IN(N'Bìa cứng',N'Bìa Mềm')) NOT NULL,

[NgayXuatBan] date NULL,

Primary Key ([MaSach])

)

go

Alter table dbo.[CHITIETPHIEUNHAP] add foreign key([SoPNS]) references [PHIEUNHAPSACH] ([SoPNS]) ON UPDATE CASCADE ON DELETE CASCADE

go

Alter table dbo.[CT\_HOADON] add foreign key([SoHD]) references [HOADON] ([SoHD]) ON UPDATE CASCADE ON DELETE CASCADE

go

Alter table dbo.[SACH] add foreign key([MaNhaXuatBan]) references [NHAXUATBAN] ([MaNhaXuatBan]) ON UPDATE CASCADE ON DELETE CASCADE

go

Alter table dbo.[CT\_HOADON] add foreign key([MaSach]) references [SACH] ([MaSach]) ON UPDATE CASCADE ON DELETE CASCADE

go

Alter table dbo.[CHITIETPHIEUNHAP] add foreign key([MaSach]) references [SACH] ([MaSach]) ON UPDATE CASCADE ON DELETE CASCADE

go

Alter table dbo.[CHITIETSACH] add foreign key([MaSach]) references [SACH] ([MaSach]) ON UPDATE CASCADE ON DELETE CASCADE

go

Alter table dbo.[SACH] add foreign key([MaTheLoai]) references [THELOAI] ([MaTheLoai]) ON UPDATE CASCADE ON DELETE CASCADE

go

Alter table dbo.[PHIEUNHAPSACH] add foreign key([MaNV]) references [NHANVIEN] ([MaNV]) ON UPDATE CASCADE ON DELETE CASCADE

go

Alter table dbo.[HOADON] add foreign key([MaNV]) references [NHANVIEN] ([MaNV]) ON UPDATE CASCADE ON DELETE CASCADE

go

Alter table dbo.[SACH] add foreign key([MaTacGia]) references [TACGIA] ([MaTacGia]) ON UPDATE CASCADE ON DELETE CASCADE

go

Alter table dbo.[NHANVIEN] add foreign key([MaChucVu]) references [ChucVu] ([MaChucVu]) ON UPDATE CASCADE ON DELETE CASCADE

go

Alter table dbo.[NHANVIEN] add foreign key([UserName]) references [TAIKHOAN] ([UserName]) ON UPDATE CASCADE ON DELETE CASCADE

go

Alter table dbo.[HOADON] add foreign key([MaKhachHang]) references [KHACHHANG] ([MaKhachHang]) ON UPDATE CASCADE ON DELETE CASCADE

go

Alter table dbo.[TAIKHOAN] add foreign key([MaQuyen]) references [PHANQUYEN] ([MaQuyen]) ON UPDATE CASCADE ON DELETE CASCADE

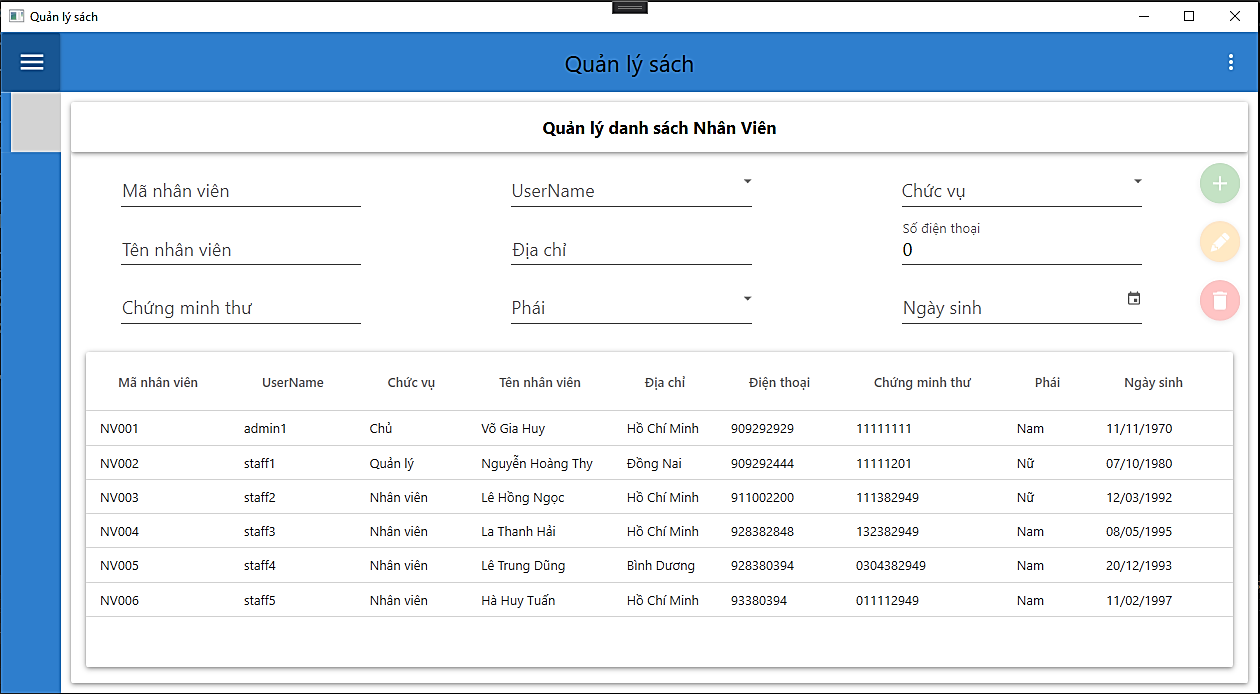
go

Alter table dbo.[SACH] add foreign key([MaKM]) references [KHUYENMAI] ([MaKM]) ON UPDATE CASCADE ON DELETE CASCADE

GO

# MÀN HÌNH THỂ HIỆN VÀ CHỨC NĂNG

1. ***Màn hình thể hiện (****NhanVienView.xaml****)***



1. ***Phần code chức năng*** *(NhanVienViewModel.cs)*

using System;

using System.Collections.ObjectModel;

using System.Linq;

using System.Windows;

using System.Windows.Input;

using QL\_SACH.Model;

namespace QL\_SACH.ViewModel

{

public class NhanVienViewModel : BaseViewModel

{

private ObservableCollection<NHANVIEN> \_ListNhanVien;

public ObservableCollection<NHANVIEN> ListNhanVien { get => \_ListNhanVien; set { \_ListNhanVien = value; OnPropertyChanged(); } }

private ObservableCollection<TAIKHOAN> \_TaiKhoan;

public ObservableCollection<TAIKHOAN> TaiKhoan { get => \_TaiKhoan; set { \_TaiKhoan = value; OnPropertyChanged(); } }

private ObservableCollection<ChucVu> \_ChucVu;

public ObservableCollection<ChucVu> ChucVu { get => \_ChucVu; set { \_ChucVu = value; OnPropertyChanged(); } }

private string \_MaNV;

public string MaNV { get => \_MaNV; set { \_MaNV = value; OnPropertyChanged(); } }

private string \_TenNV;

public string TenNV { get => \_TenNV; set { \_TenNV = value; OnPropertyChanged(); } }

private string \_DiaChi;

public string DiaChi { get => \_DiaChi; set { \_DiaChi = value; OnPropertyChanged(); } }

private int \_SoDienThoai;

public int SoDienThoai { get => \_SoDienThoai; set { \_SoDienThoai = value; OnPropertyChanged(); } }

private string \_CMND;

public string CMND { get => \_CMND; set { \_CMND = value; OnPropertyChanged(); } }

private string \_Phai;

public string Phai { get => \_Phai; set { \_Phai = value; OnPropertyChanged(); } }

private DateTime? \_NgaySinh;

public DateTime? NgaySinh { get => \_NgaySinh; set { \_NgaySinh = value; OnPropertyChanged(); } }

/// <summary>

/// Khai báo Command

/// </summary>

public ICommand AddCommand { get; set; }

public ICommand EditCommand { get; set; }

public ICommand DeleteCommand { get; set; }

/// <summary>

/// Thay đổi giá trị trong control khi chọn một đối tượng trong ListView

/// </summary>

private Model.NHANVIEN \_SelectedItem;

public Model.NHANVIEN SelectedItem

{

get => \_SelectedItem;

set

{

\_SelectedItem = value;

OnPropertyChanged();

if (SelectedItem != null)

{

MaNV = SelectedItem.MaNV;

SelectedTaiKhoan = SelectedItem.TAIKHOAN;

SelectedChucVu = SelectedItem.ChucVu;

TenNV = SelectedItem.TenNV;

DiaChi = SelectedItem.DiaChi;

SoDienThoai = SelectedItem.SoDienThoai;

CMND = SelectedItem.CMND;

Phai = SelectedItem.Phai;

NgaySinh = SelectedItem.NgaySinh;

}

}

}

private Model.TAIKHOAN \_SelectedTaiKhoan;

public Model.TAIKHOAN SelectedTaiKhoan

{

get => \_SelectedTaiKhoan;

set

{

\_SelectedTaiKhoan = value;

OnPropertyChanged();

}

}

private Model.ChucVu \_SelectedChucVu;

public Model.ChucVu SelectedChucVu

{

get => \_SelectedChucVu;

set

{

\_SelectedChucVu = value;

OnPropertyChanged();

}

}

/// <summary>

/// Reset dữ liệu trên các control về rỗng

/// </summary>

public void Reset()

{

MaNV = null;

SelectedTaiKhoan = null;

SelectedChucVu = null;

TenNV = null;

DiaChi = null;

SoDienThoai = 0;

CMND = null;

Phai = null;

NgaySinh = null;

SelectedItem = null;

}

/// <summary>

/// Các xử lý chính

/// </summary>

public NhanVienViewModel()

{

ListNhanVien = new ObservableCollection<NHANVIEN>(DataProvider.Ins.DB.NHANVIEN);

TaiKhoan = new ObservableCollection<TAIKHOAN>(DataProvider.Ins.DB.TAIKHOAN);

ChucVu = new ObservableCollection<ChucVu>(DataProvider.Ins.DB.ChucVu);

//Thêm nhân viên

AddCommand = new RelayCommand<object>((p) =>

{

if ((SelectedItem == null) && (SelectedTaiKhoan != null && SelectedChucVu != null && TenNV != null && DiaChi != null && SoDienThoai.ToString() != null && CMND != null && Phai != null && NgaySinh != null))

{

return true;

}

return false;

}, (p) =>

{

if (DataProvider.Ins.DB.NHANVIEN.Find(MaNV) == null)

{

var AddRecord = MessageBox.Show("Bạn có chắc chắn muốn Thêm Nhân viên " + TenNV + " không?", "Xác nhận", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (AddRecord == MessageBoxResult.Yes)

{

var nhanvien = new Model.NHANVIEN() { MaNV = MaNV, UserName = SelectedTaiKhoan.UserName, MaChucVu = SelectedChucVu.MaChucVu, TenNV = TenNV, DiaChi = DiaChi, SoDienThoai = SoDienThoai, CMND = CMND, Phai = Phai, NgaySinh = NgaySinh };

DataProvider.Ins.DB.NHANVIEN.Add(nhanvien);

DataProvider.Ins.DB.SaveChanges();

ListNhanVien.Add(nhanvien);

MessageBox.Show("Nhân viên " + TenNV + " đã thêm thành công", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Information);

Reset();

}

}

else

{

MessageBox.Show("Mã nhân viên " + MaNV + " đã tồn tại", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Warning);

}

});

//Sửa thông tin nhân viên

EditCommand = new RelayCommand<object>((p) =>

{

if ((SelectedTaiKhoan == null || SelectedChucVu == null || TenNV == null || DiaChi == null) || SoDienThoai.ToString() == null || CMND == null || Phai == null || NgaySinh == null)

return false;

var displayList = DataProvider.Ins.DB.NHANVIEN.Where(x => x.MaNV == SelectedItem.MaNV);

if (displayList != null && displayList.Count() != 0)

return true;

return false;

}, (p) =>

{

var FixRecord = MessageBox.Show("Bạn có chắc chắn muốn sửa Thông tin nhân viên " + SelectedItem.TenNV + " không?", "Xác nhận", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (FixRecord == MessageBoxResult.Yes)

{

var nhanvien = DataProvider.Ins.DB.NHANVIEN.Where(x => x.MaNV == SelectedItem.MaNV).SingleOrDefault();

nhanvien.UserName = SelectedTaiKhoan.UserName;

nhanvien.MaChucVu = SelectedChucVu.MaChucVu;

nhanvien.TenNV = TenNV;

nhanvien.DiaChi = DiaChi;

nhanvien.SoDienThoai = SoDienThoai;

nhanvien.CMND = CMND;

nhanvien.Phai = Phai;

nhanvien.NgaySinh = NgaySinh;

DataProvider.Ins.DB.SaveChanges();

//SelectedItem.MaNV = MaNV;

MessageBox.Show("Sửa thông tin Nhân viên " + TenNV + " đã sửa thành công", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Information);

Reset();

}

});

//Xóa nhân viên

DeleteCommand = new RelayCommand<object>((p) =>

{

if (SelectedItem == null)

return false;

var displayList = DataProvider.Ins.DB.NHANVIEN.Where(x => x.MaNV == SelectedItem.MaNV);

if (displayList != null && displayList.Count() != 0)

return true;

return false;

}, (p) =>

{

var DeleteRecord = MessageBox.Show("Bạn có chắc chắn muốn xóa Nhân viên " + SelectedItem.TenNV + " không?", "Xác nhận", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (DeleteRecord == MessageBoxResult.Yes)

{

var nhanvien = DataProvider.Ins.DB.NHANVIEN.Where(x => x.MaNV == SelectedItem.MaNV).FirstOrDefault();

DataProvider.Ins.DB.NHANVIEN.Remove(nhanvien);

DataProvider.Ins.DB.SaveChanges();

MessageBox.Show("Nhân viên " + TenNV + " đã xóa thành công", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Information);

ListNhanVien.Remove(nhanvien);

Reset();

}

});

}

}

}

1. ***Giải thích từng chức năng***
2. Chức năng thêm nhân viên

//Thêm nhân viên

AddCommand = new RelayCommand<object>((p) =>

{

if ((SelectedItem == null) && (SelectedTaiKhoan != null && SelectedChucVu != null && TenNV != null && DiaChi != null && SoDienThoai.ToString() != null && CMND != null && Phai != null && NgaySinh != null))

{

return true;

}

return false;

}, (p) =>

{

if (DataProvider.Ins.DB.NHANVIEN.Find(MaNV) == null)

{

var AddRecord = MessageBox.Show("Bạn có chắc chắn muốn Thêm Nhân viên " + TenNV + " không?", "Xác nhận", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (AddRecord == MessageBoxResult.Yes)

{

var nhanvien = new Model.NHANVIEN() { MaNV = MaNV, UserName = SelectedTaiKhoan.UserName, MaChucVu = SelectedChucVu.MaChucVu, TenNV = TenNV, DiaChi = DiaChi, SoDienThoai = SoDienThoai, CMND = CMND, Phai = Phai, NgaySinh = NgaySinh };

DataProvider.Ins.DB.NHANVIEN.Add(nhanvien);

DataProvider.Ins.DB.SaveChanges();

ListNhanVien.Add(nhanvien);

MessageBox.Show("Nhân viên " + TenNV + " đã thêm thành công", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Information);

Reset();

}

}

else

{

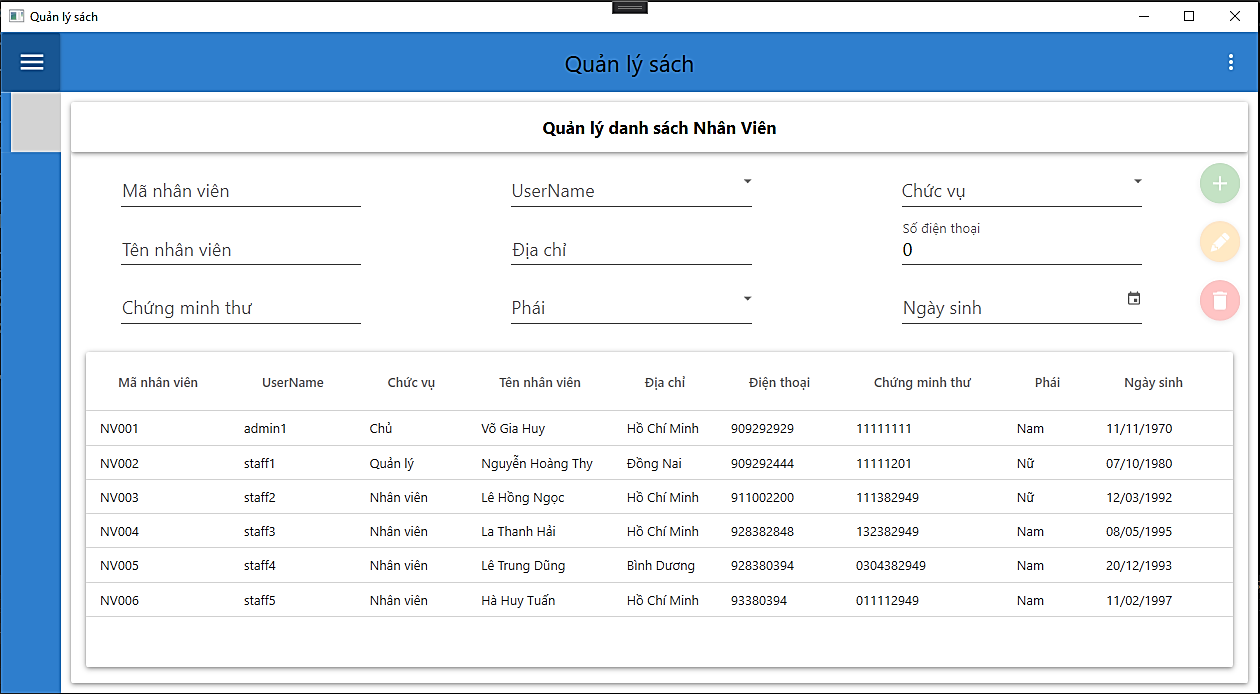
MessageBox.Show("Mã nhân viên " + MaNV + " đã tồn tại", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Warning);

}

});

* **Giải thích:**
* Lúc này nút thêm đang bị *tắt*

Nút thêm nhân viên



* Nút thêm nhân viên binding đến AddCommand trong lớp NhanVienViewModel.cs để thực hiện thức năng thêm nhân viên
* Button Thêm chỉ được IsEnabled khi dữ liệu trong tất cả các control *không bị rỗng* và không có nhân viên nào *đang* được chọn trong ListNhanVien

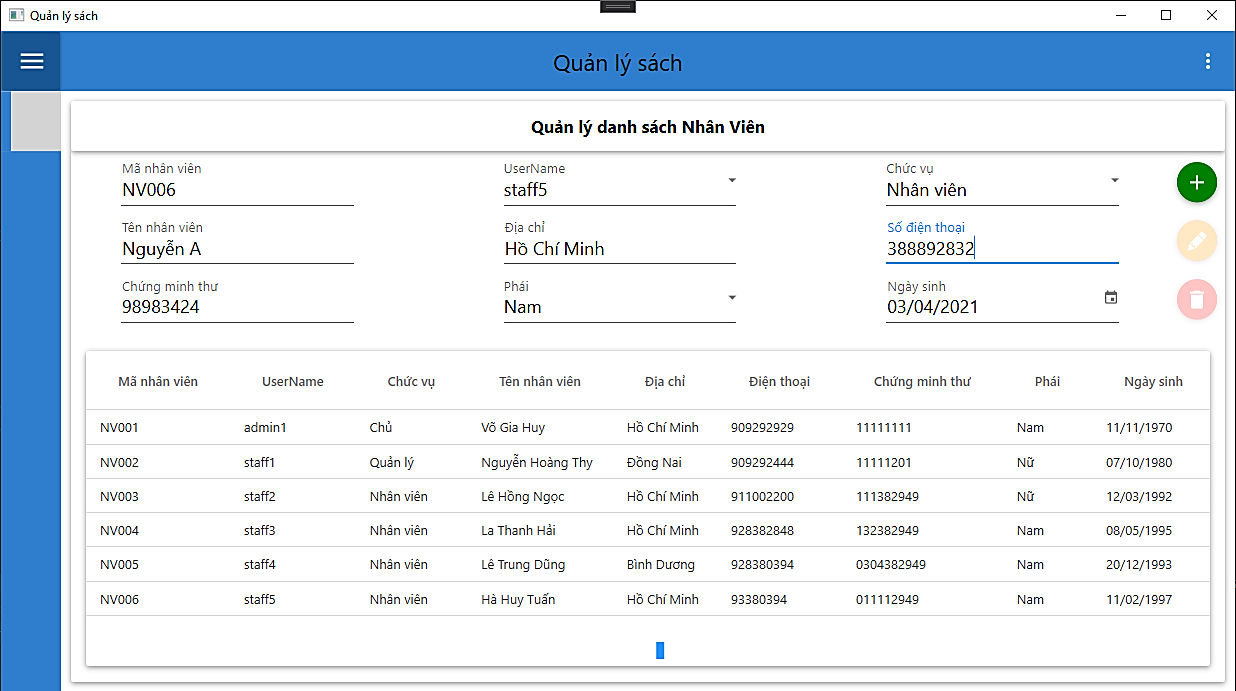
if ((SelectedItem == null) && (SelectedTaiKhoan != null && SelectedChucVu != null && TenNV != null && DiaChi != null && SoDienThoai.ToString() != null && CMND != null && Phai != null && NgaySinh != null))

{

return true;

}

return false;



Nút thêm đã được bật

* Lúc này nút thêm đã được bật khi dữ liệu trong các control đã được nhập vào đầy đủ và không có đối tượng nào đang được chọn trong ListNhanVien.
* Khí nhấn nút thêm, nút thêm sẽ binding đến AddCommand trong lớp NhanVienViewModel.cs để thực hiện thức năng thêm nhân viên.

if (DataProvider.Ins.DB.NHANVIEN.Find(MaNV) == null)

{

var AddRecord = MessageBox.Show("Bạn có chắc chắn muốn Thêm Nhân viên " + TenNV + " không?", "Xác nhận", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (AddRecord == MessageBoxResult.Yes)

{

var nhanvien = new Model.NHANVIEN() { MaNV = MaNV, UserName = SelectedTaiKhoan.UserName, MaChucVu = SelectedChucVu.MaChucVu, TenNV = TenNV, DiaChi = DiaChi, SoDienThoai = SoDienThoai, CMND = CMND, Phai = Phai, NgaySinh = NgaySinh };

DataProvider.Ins.DB.NHANVIEN.Add(nhanvien);

DataProvider.Ins.DB.SaveChanges();

ListNhanVien.Add(nhanvien);

MessageBox.Show("Nhân viên " + TenNV + " đã thêm thành công", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Information);

Reset();

}

}

else

{

MessageBox.Show("Mã nhân viên " + MaNV + " đã tồn tại", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Warning);

* Đầu tiên kiểm tra mã nhân viên đã tồn tại hay chưa, nếu đã tồn tại thì sẽ xuất hiện hộp thoại báo lỗi cho người dùng.

Graphical user interface, application

Description automatically generated

Báo lỗi trùng mã nhân viên

* Nếu mã nhân viên chưa tồn tại thì sẽ xuất hiện hộp thoại để người dùng chắc chắn muốn thêm nhân viên mới này.

Graphical user interface

Description automatically generated

Hỏi người dùng xác mình việc thêm nhân viên

* Khi người dùng bấm No thì nhân viên sẽ không được thêm vào, nếu bấm Yes thì việc thêm nhân viên sẽ được thực hiện.

if (AddRecord == MessageBoxResult.Yes)

{

var nhanvien = new Model.NHANVIEN() { MaNV = MaNV, UserName = SelectedTaiKhoan.UserName, MaChucVu = SelectedChucVu.MaChucVu, TenNV = TenNV, DiaChi = DiaChi, SoDienThoai = SoDienThoai, CMND = CMND, Phai = Phai, NgaySinh = NgaySinh };

DataProvider.Ins.DB.NHANVIEN.Add(nhanvien);

DataProvider.Ins.DB.SaveChanges();

ListNhanVien.Add(nhanvien);

MessageBox.Show("Nhân viên " + TenNV + " đã thêm thành công", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Information);

Reset();

}

* Đầu tiên tạo một object *nhanvien* kiểu NHANVIEN với dữ liệu lấy từ các control vừa nhập trên form.

var nhanvien = new Model.NHANVIEN() { MaNV = MaNV, UserName = SelectedTaiKhoan.UserName, MaChucVu = SelectedChucVu.MaChucVu, TenNV = TenNV, DiaChi = DiaChi, SoDienThoai = SoDienThoai, CMND = CMND, Phai = Phai, NgaySinh = NgaySinh };

* Sau đó thêm nhanvien vào cơ sở dữ liệu và lưu lại bằng phương thức Add() và SaveChanges().

DataProvider.Ins.DB.NHANVIEN.Add(nhanvien);

DataProvider.Ins.DB.SaveChanges();

* Xuất hiện hộp thoại thông báo cho người dùng biết đã thêm thành công.

MessageBox.Show("Nhân viên " + TenNV + " đã thêm thành công", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Information);

Graphical user interface

Description automatically generated

Thông báo người dùng thêm nhân viên thành công

* Sử dụng phương thức Reset() để làm trống dữ liệu vừa nhập trên form.

public void Reset()

{

MaNV = null;

SelectedTaiKhoan = null;

SelectedChucVu = null;

TenNV = null;

DiaChi = null;

SoDienThoai = 0;

CMND = null;

Phai = null;

NgaySinh = null;

SelectedItem = null;

}

* Hoàn thành chức năng thêm.

1. Chức năng sửa thông tin nhân viên

EditCommand = new RelayCommand<object>((p) =>

{

if (SelectedTaiKhoan == null || SelectedChucVu == null || TenNV == null || DiaChi == null || SoDienThoai.ToString() == null || CMND == null || Phai == null || NgaySinh == null || SelectedItem == null)

return false;

var displayList = DataProvider.Ins.DB.NHANVIEN.Where(x => x.MaNV == SelectedItem.MaNV);

if (displayList != null && displayList.Count() != 0)

return true;

return false;

}, (p) =>

{

var FixRecord = MessageBox.Show("Bạn có chắc chắn muốn sửa Thông tin nhân viên " + SelectedItem.TenNV + " không?", "Xác nhận", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (FixRecord == MessageBoxResult.Yes)

{

var nhanvien = DataProvider.Ins.DB.NHANVIEN.Where(x => x.MaNV == SelectedItem.MaNV).SingleOrDefault();

nhanvien.UserName = SelectedTaiKhoan.UserName;

nhanvien.MaChucVu = SelectedChucVu.MaChucVu;

nhanvien.TenNV = TenNV;

nhanvien.DiaChi = DiaChi;

nhanvien.SoDienThoai = SoDienThoai;

nhanvien.CMND = CMND;

nhanvien.Phai = Phai;

nhanvien.NgaySinh = NgaySinh;

DataProvider.Ins.DB.SaveChanges();

//SelectedItem.MaNV = MaNV;

MessageBox.Show("Sửa thông tin Nhân viên " + TenNV + " đã sửa thành công", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Information);

Reset();

}

});

* **Giải thích**
* Nút sửa đang bị tắt vì chưa chọn đối tượng nào để sửa trong ListNhanVien.

Graphical user interface, application

Description automatically generated

Nút sửa đang bị tắt

* Nút sửa thông tin nhân viên binding đến EditCommand trong lớp NhanVienViewModel.cs để thực hiện thức năng sửa thông tin nhân viên.
* Button Sửa chỉ được IsEnabled khi một đối tượng nhân viên được chọn trong ListNhanVien và dữ liệu trong các control là đầy đủ.

if (SelectedTaiKhoan == null || SelectedChucVu == null || TenNV == null || DiaChi == null || SoDienThoai.ToString() == null || CMND == null || Phai == null || NgaySinh == null || SelectedItem == null)

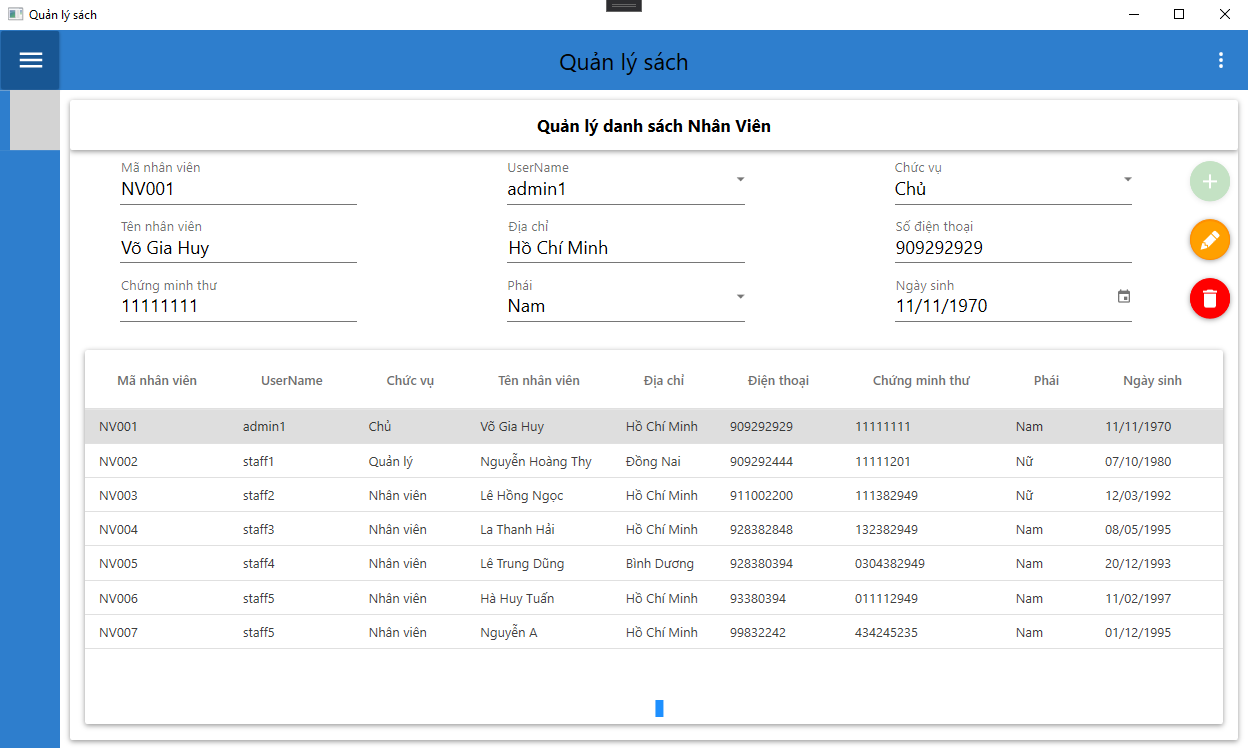
return false;

var displayList = DataProvider.Ins.DB.NHANVIEN.Where(x => x.MaNV == SelectedItem.MaNV);

if (displayList != null && displayList.Count() != 0)

return true;

return false;

****

Nút sửa được mở

Nhân viên được chọn

* Khi nút sửa được ấn, một hộp thoại sẽ xuất hiện để xác minh người nhập có thực sự muốn sửa thông tin nhân viên đang chọn hay không.

**Graphical user interface, application

Description automatically generated**

* Khi người dùng bấm No, việc sửa sẽ không được thực hiện, nếu người dùng bấm Yes, công việc sửa dữ liệu sẽ được tiến hành.

if (FixRecord == MessageBoxResult.Yes)

{

var nhanvien = DataProvider.Ins.DB.NHANVIEN.Where(x => x.MaNV == SelectedItem.MaNV).SingleOrDefault();

nhanvien.UserName = SelectedTaiKhoan.UserName;

nhanvien.MaChucVu = SelectedChucVu.MaChucVu;

nhanvien.TenNV = TenNV;

nhanvien.DiaChi = DiaChi;

nhanvien.SoDienThoai = SoDienThoai;

nhanvien.CMND = CMND;

nhanvien.Phai = Phai;

nhanvien.NgaySinh = NgaySinh;

DataProvider.Ins.DB.SaveChanges();

//SelectedItem.MaNV = MaNV;

MessageBox.Show("Sửa thông tin Nhân viên " + TenNV + " đã sửa thành công", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Information);

Reset();

}

* Đầu tiên lấy ra nhân viên có mã nhân viên tương ứng với mã nhân viên đang chọn trên control bằng phương thức Where() .

var nhanvien = DataProvider.Ins.DB.NHANVIEN.Where(x => x.MaNV == SelectedItem.MaNV).SingleOrDefault();

* Sau đó gán dữ liệu mới cho các thuộc tính của đối tượng vừa lấy từ các dữ liệu trên các control.

nhanvien.UserName = SelectedTaiKhoan.UserName;

nhanvien.MaChucVu = SelectedChucVu.MaChucVu;

nhanvien.TenNV = TenNV;

nhanvien.DiaChi = DiaChi;

nhanvien.SoDienThoai = SoDienThoai;

nhanvien.CMND = CMND;

nhanvien.Phai = Phai;

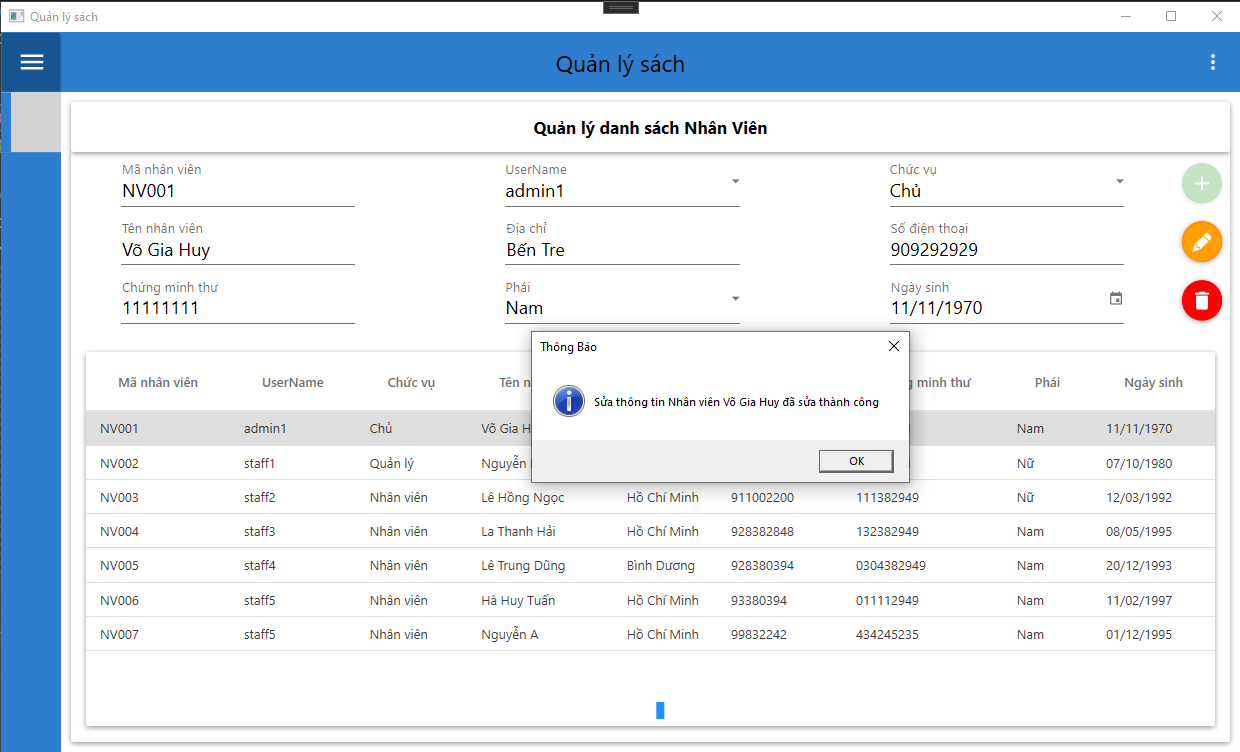
nhanvien.NgaySinh = NgaySinh;

* Sau đó lưu lại bằng phương thức SaveChanges() và thông báo cho người dùng biết đã sửa thông tin thành công.

DataProvider.Ins.DB.SaveChanges();

//SelectedItem.MaNV = MaNV;

MessageBox.Show("Sửa thông tin Nhân viên " + TenNV + " đã sửa thành công", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Information);



Thông báo người dùng đã sửa thành công

* Sử dụng phương thức Reset() để làm trống dữ liệu vừa nhập trên form.

public void Reset()

{

MaNV = null;

SelectedTaiKhoan = null;

SelectedChucVu = null;

TenNV = null;

DiaChi = null;

SoDienThoai = 0;

CMND = null;

Phai = null;

NgaySinh = null;

SelectedItem = null;

}

* Hoàn thành chức năng sửa thông tin nhân viên.

1. Chức năng xóa nhân viên.

DeleteCommand = new RelayCommand<object>((p) =>

{

if (SelectedItem == null)

return false;

var displayList = DataProvider.Ins.DB.NHANVIEN.Where(x => x.MaNV == SelectedItem.MaNV);

if (displayList != null && displayList.Count() != 0)

return true;

return false;

}, (p) =>

{

var DeleteRecord = MessageBox.Show("Bạn có chắc chắn muốn xóa Nhân viên " + SelectedItem.TenNV + " không?", "Xác nhận", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (DeleteRecord == MessageBoxResult.Yes)

{

var nhanvien = DataProvider.Ins.DB.NHANVIEN.Where(x => x.MaNV == SelectedItem.MaNV).FirstOrDefault();

DataProvider.Ins.DB.NHANVIEN.Remove(nhanvien);

DataProvider.Ins.DB.SaveChanges();

MessageBox.Show("Nhân viên " + TenNV + " đã xóa thành công", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Information);

ListNhanVien.Remove(nhanvien);

Reset();

}

});

* **Giải thích**
* Nút xóa đang bị *tắt* vì chưa chọn đối tượng nào để xóa trong ListNhanVien.

Graphical user interface, application

Description automatically generated

Nút xóa đang bị tắt

* Nút xóa nhân viên binding đến DeleteCommand trong lớp NhanVienViewModel.cs để thực hiện thức năng xóa nhân viên.
* Button Xóa chỉ được IsEnabled khi một đối tượng nhân viên được chọn trong ListNhanVien.

if (SelectedItem == null)

return false;

var displayList = DataProvider.Ins.DB.NHANVIEN.Where(x => x.MaNV == SelectedItem.MaNV);

if (displayList != null && displayList.Count() != 0)

return true;

return false;

Graphical user interface, application

Description automatically generated

Nhân viên đang chọn

Nút xóa được bật

* Khi nút xóa được ấn, một hộp thoại sẽ xuất hiện để xác minh người nhập có thực sự muốn xóa nhân viên đang chọn hay không.

**ssGraphical user interface

Description automatically generated**

Người dùng có thật sự muốn xóa hay không

* Khi người dùng bấm No việc xóa sẽ không được thực hiện, nếu bấm Yes việc xóa nhân viên sẽ bắt đầu.

if (DeleteRecord == MessageBoxResult.Yes)

{

var nhanvien = DataProvider.Ins.DB.NHANVIEN.Where(x => x.MaNV == SelectedItem.MaNV).FirstOrDefault();

DataProvider.Ins.DB.NHANVIEN.Remove(nhanvien);

DataProvider.Ins.DB.SaveChanges();

MessageBox.Show("Nhân viên " + TenNV + " đã xóa thành công", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Information);

ListNhanVien.Remove(nhanvien);

Reset();

}

* Đầu tiên sử dụng phương thức Where() để lấy ra nhân viên cần xóa với mã nhân viên tương ứng với mã nhân viên đang được chọn trong ListNhanVien.

var nhanvien = DataProvider.Ins.DB.NHANVIEN.Where(x => x.MaNV == SelectedItem.MaNV).FirstOrDefault();

* Sử dụng phương thức Remove() để tiến hành xóa nhân viên và phương thức SaveChanges() để lưu lại. Và xuất hiện hộp thoại để thông báo người dùng đã xóa thành công.
* MessageBox.Show("Nhân viên " + TenNV + " đã xóa thành công", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Information);
* ListNhanVien.Remove(nhanvien);

Graphical user interface, application

Description automatically generated

* Sử dụng phương thức Reset() để làm trống dữ liệu trên form.

public void Reset()

{

MaNV = null;

SelectedTaiKhoan = null;

SelectedChucVu = null;

TenNV = null;

DiaChi = null;

SoDienThoai = 0;

CMND = null;

Phai = null;

NgaySinh = null;

SelectedItem = null;

}

* Hoàn thành chức năng xóa nhân viên.