TRƯỜNG ĐẠI HỌC AN GIANG

**KHOA CÔNG NGHỆ THÔNG TIN**

Ảnh có chứa văn bản, biểu tượng, Nhãn hiệu, Phông chữ

Mô tả được tạo tự động

**ĐỒ ÁN MÔN HỌC LẬP TRÌNH .NET**

**HỆ THỐNG QUẢN LÝ KÝ TÚC XÁ**

Tổ thực hành 2\_Nhóm 11

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **Sinh viên thực hiện:** |
| Ths.Phạm Hữu Dũng | Trần Khả Ái – DTH215808 |
|  | Trần Minh Quang - DTH216109 |
|  |  |

An Giang, 12/2023

* **Nội dung thực hiện, tỷ lệ tham gia của các thành viên:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MSSV** | **Họ tên sinh viên** | **Lớp** | **Nội dung thực hiện** | **Tỷ lệ tham gia** |
| DTH215808 | Trần Khả Ái | DH22TH3 | Word, PPT, CSDL, Code, Giao Diện | 100% |
| DTH216109 | Trần Minh Quang | DH22TH3 | Word, PPT, Code, Giao Diện | 100% |

***Mục lục***

***[LỜI MỞ ĐẦU 3](#_Toc152972334)***

***[I. PHÂN TÍCH TỔNG QUAN. 4](#_Toc152972335)***

***[1. Giới thiệu tổng quan hệ thống. 4](#_Toc152972336)***

***[2. Yêu cầu hệ thống. 4](#_Toc152972337)***

***[a) Quản lý thông tin sinh viên. 4](#_Toc152972338)***

***[b) Quản lý thông tin nhân viên. 4](#_Toc152972339)***

***[c) Quản lý thông tin thân nhân. 5](#_Toc152972340)***

***[d) Quản lý thông tin điện nước. 5](#_Toc152972341)***

***[e) Quản lý thông tin hóa đơn. 5](#_Toc152972342)***

***[f) Quản lý thông tin hợp đồng. 5](#_Toc152972343)***

***[g) Quản lý phòng. 6](#_Toc152972344)***

***[II. THIẾT KẾ VÀ CÀI ĐẶT CHƯƠNG TRÌNH. 6](#_Toc152972345)***

***[1. Xây dựng mô hình quan hệ. 6](#_Toc152972346)***

***[2. Thiết kế dữ liệu. 7](#_Toc152972347)***

***[III. SƠ ĐỒ QUAN HỆ THỰC THỂ. 9](#_Toc152972348)***

***[IV. CÀI ĐẶT CHƯƠNG TRÌNH 10](#_Toc152972349)***

# LỜI MỞ ĐẦU

Thời đại ngày nay, con người đang đặt sự chú trọng lớn vào việc học tập và nâng cao kiến thức của mình. Đồng thời, sự phát triển mạnh mẽ trong các lĩnh vực kinh tế và xã hội đã tạo ra nhu cầu ngày càng cao cho ứng dụng công nghệ thông tin, một lực lượng không ngừng biến đổi.

Với sự phức tạp và phát triển của hệ thống, các phương thức quản lý cổ điển truyền thống trở nên cồng kềnh và khó có thể đáp ứng được yêu cầu ngày càng cao của hệ thống. Để xử lý thông tin một cách nhanh chóng, chính xác và hiệu quả, không thể thiếu sự hỗ trợ của công nghệ thông tin. Đây là yếu tố quyết định, đặc biệt là trong bối cảnh công nghệ thông tin ngày càng trở thành trụ cột chính của nền kinh tế tri thức.

Hiện nay, đa phần sinh viên đến từ các tỉnh lẻ, nhu cầu về chỗ ở trở thành một vấn đề cấp bách. Sinh viên thường chọn ở ký túc xá vì đó là lựa chọn an toàn, giá phòng hợp lý và có mô hình học tập sinh hoạt tốt. Tuy nhiên, nhiều ký túc xá của các trường đại học áp dụng mô hình quản lý không hợp lý, đặc biệt là khi số lượng sinh viên đăng ký vượt quá khả năng quản lý.

Việc này dẫn đến khó khăn trong quản lý thông tin, ghi chép thủ công và nguy cơ sai sót cao. Điều này không chỉ ảnh hưởng đến chất lượng quản lý mà còn có thể dẫn đến thất thoát tài chính. Vì thế, biết được tầm quan trọng của ứng dụng công nghệ thông tin trong quản lý, chúng tôi đã đặt ra ý định tìm hiểu ngôn ngữ lập trình C# và hệ quản trị cơ sở dữ liệu MySQL. Điều này nhằm mục đích xây dựng một phần mềm quản lý ký túc xá hiện đại, đáp ứng đầy đủ yêu cầu và phù hợp với môi trường học thuật.

# **PHÂN TÍCH TỔNG QUAN.**

## Giới thiệu tổng quan hệ thống.

* Ở thời điểm mà nhu cầu về chỗ ở đối với sinh viên và cộng đồng đại học ngày càng tăng, vai trò của phần mềm quản lý ký túc xá trở nên vô cùng quan trọng. Trong không khí sôi động của môi trường đại học hiện đại, không thể phủ nhận sự cần thiết của sự hỗ trợ từ các giải pháp phần mềm quản lý ký túc xá.
* Một hệ thống quản lý ký túc xá thông minh đem lại sự thuận tiện và hiệu quả cho cả quản lý và cư dân. Với tính linh hoạt và đa dạng, phần mềm giúp quản lý tốt thông tin về phòng ở, quản lý tình trạng phòng trống, và theo dõi lịch sử chuyển phòng, tất cả chỉ qua một giao diện duy nhất.
* Sự tự động hóa được tích hợp giúp giảm thiểu thủ tục rườm rà, từ quá trình đăng ký, thanh toán cho đến quản lý các yêu cầu bảo trì. Thời gian và công sức được tiết kiệm, giúp tăng cường hiệu suất quản lý ký túc xá.
* Thông qua việc phân tích dữ liệu và theo dõi các xu hướng về sử dụng phòng ở, hệ thống cung cấp cái nhìn rõ ràng về hoạt động của ký túc xá. Điều này giúp quản lý xây dựng chiến lược hiệu quả hơn, tối ưu hóa nguồn lực và nâng cao trải nghiệm sống cho cư dân.
* Không chỉ giúp quản lý mọi khía cạnh của ký túc xá, hệ thống quản lý ký túc xá còn tạo điều kiện thuận lợi cho tương tác giữa quản lý và cư dân. Việc quản lý cơ sở dữ liệu cư dân, theo dõi tiến trình đăng ký và giao nhận phòng, cùng việc tối ưu hóa chiến lược tiếp thị dựa trên phản hồi từ cư dân giúp tăng cường mối quan hệ và nâng cao sự hài lòng của cư dân đối với ký túc xá.

## Yêu cầu hệ thống.

### Quản lý thông tin sinh viên.

* + - * Trong thời gian sinh viên tạm trú ở ký túc xá, bộ phận quản lý sinh viên sẽ tiếp nhận và lưu trữ toàn bộ thông tin về sinh viên, bao gồm: mã sinh viên, họ tên, ngày sinh, giới tính, lớp - khóa, số điện thoại, thông tin người thân để liên lạc…

### Quản lý thông tin nhân viên.

* + - * Chức năng quản lý nhân viên của hệ thống này cũng rất quan trọng để duy trì sự tổ chức và sắp xếp công việc trong doanh nghiệp. Hết sức tiện ích để có một công cụ giúp tích hợp dữ liệu về thông tin cá nhân của từng nhân viên - bao gồm mã nhân viên, tên nhân viên, thông tin liên hệ và lương của từng nhân viên - vào một nền tảng duy nhất. Điều này cho phép các người quản lý dễ dàng theo dõi các chỉ số hiệu suất và tổ chức công việc theo phù hợp.

### Quản lý thông tin thân nhân.

* + - * Khi sinh viên đăng kí nội trú trong ktx thì cần cung cấp sđt liên lạc cũng như địa chỉ của thân nhân. Ban quản lí cũng cần phải quản lí đối tượng này phòng khi có các trường hợp khẩn cấp xảy ra (sinh viên gặp tai nạn, mất tích, đau ốm,…) hay khi người thân của sinh viên có nhu cầu thăm nom và lưu lại ktx qua đêm.

### Quản lý thông tin điện nước.

* + - * Hàng tháng, bộ phận quản lý lập hóa đơn điện nước cho từng phòng, kiểm tra công tơ điện và nước, và tính toán thành tiền. Thông tin chi tiết bao gồm mã điện nước, chỉ số đầu – cuối của nước, chỉ số đầu – cuối của điện, ngày bắt đầu, ngày kết thúc. Khi có thay đổi giá điện nước, bộ phận quản lý sửa đổi thông tin để duy trì độ chính xác của hóa đơn và dữ liệu tiêu thụ.

### Quản lý thông tin hóa đơn.

* + - * Trong quá trình quản lý bán hàng, khi nhận được đơn đặt hàng từ phía khách hàng mua trực tiếp hoặc gián tiếp, nhân viên sẽ tiếp nhận và xử lý đơn hàng này. Sau khi xác nhận đơn hàng với khách hàng thành công và thanh toán hoàn tất, nhân viên sẽ thực hiện xuất hóa đơn cho khách hàng.
      * Thông tin trên hóa đơn bao gồm mã hóa đơn, mã hợp đồng, mã điện nước, tổng tiền điện, tổng tiền nước và tháng năm thanh toán. Những thông tin này không chỉ là minh chứng cho quá trình giao dịch mà còn là nguồn dữ liệu quan trọng để thực hiện các công đoạn thống kê và báo cáo trong hệ thống quản lý của công ty. Điều này giúp tạo ra một cơ sở dữ liệu đáng tin cậy, cung cấp thông tin cần thiết để đưa ra quyết định và theo dõi hiệu suất bán hàng của công ty.

### Quản lý thông tin hợp đồng.

* + - * Khi có nhu cầu tạm trú trong kí túc xá, đặc biệt là vào đầu các năm học, sinh viên cần thực hiện đăng kí với bộ phận quản lý kí túc xá, được chủ trì bởi trung tâm hỗ trợ sinh viên. Bộ phận quản lý sẽ tiến hành xem xét và kiểm tra hồ sơ của sinh viên. Nếu hợp đồng đáp ứng đủ yêu cầu, sinh viên sẽ được tiếp nhận và thực hiện thủ tục đăng kí để vào ở trong kí túc xá.
      * Mỗi sinh viên sẽ ký một hợp đồng có thời hạn kéo dài 9 tháng, tương đương với một học kỳ. Khi hợp đồng này hết hạn, sinh viên có hai lựa chọn: thực hiện thủ tục đăng kí hợp đồng mới hoặc gia hạn nếu muốn tiếp tục tạm trú trong kí túc xá. Điều này giúp quy định rõ ràng thời gian ở và quản lý việc tái đàm phán hợp đồng cho những sinh viên muốn tiếp tục ở lại.

### Quản lý phòng.

* + - * Ban quản lý sẽ thường xuyên cập nhật thông tin về mỗi phòng ở mỗi nhà kí túc xá, bao gồm số phòng, tầng, tình trạng phòng, số lượng sinh viên, và các thông tin khác liên quan. Điều này nhằm đảm bảo rằng hệ thống luôn phản ánh đúng và chi tiết về tình hình ở ký túc xá.
      * Trong trường hợp có sự cố như cháy nổ, hoặc hỏng hóc trang thiết bị và vật tư trong ký túc xá, sinh viên có trách nhiệm báo ngay với ban quản lý. Ngược lại, khi ký túc xá thực hiện bổ sung trang thiết bị mới, ban quản lý sẽ thông báo cho trưởng phòng của từng phòng để họ có thể đến nhận tại nơi qui định hoặc yêu cầu người chuyển lên tận phòng. Điều này giúp quản lý hiệu quả các vấn đề về trang thiết bị và đảm bảo rằng mọi phòng đều được trang bị đầy đủ và an toàn.

# THIẾT KẾ VÀ CÀI ĐẶT CHƯƠNG TRÌNH.

## Xây dựng mô hình quan hệ.

* + - * + Sinh Viên ( **MaSV**, MaNguoiThan, HoTen, GioiTinh, NgaySinh, Email, DiaChi, SoDienThoai, MaPhong )

*( Mỗi sinh viên phải có một mã người thân, họ tên người thân, giới tính, ngày sinh, email, địa chỉ, số điện thoại, mã phòng ).*

* + - * + Nhân viên ( **Manv**, HoTen, GioiTinh, QueQuan, Email, ChucVu, SDT )

*( Mỗi nhân viên phải có một mã nhân viên, họ tên, quê quán, email, chức vụ, số điện thoại ).*

* + - * + Người thân ( **MaNguoiThan**, HoTenNguoiThan, GioiTinh, DiaChi, SoDienThoai, Email, NgaySinh, MaSV )

*( Mỗi sinh viên phải có một mã người thân, họ tên người thân, giới tính, địa chỉ, số điện thoại, email, mã sinh viên ).*

* + - * + Điện nước ( **MaDienNuoc,** ChiSoDau\_Dien, ChiSoDau\_Nuoc, ChiSoCuoi\_Dien, ChiSoCuoi\_Nuoc, NgayBatDau, NgayKetThuc )

*( Mỗi hóa đơn điện nước phải có một mã điện nước, chỉ số đầu của điện, chỉ số đầu của nước, chỉ số cuối của điện, chr số cuối của nước, ngày bắt đầu, ngày kết thúc ).*

* + - * + Hoá đơn ( **MaHD**, MaHopDong, MaDienNuoc, TongTienDien, TongTienNuoc, ThangNamThanhToan )

*( Mỗi hoá đơn phải có mã hoá đơn, mã hợp đồng, mã điện nước, tổng tiền điện, tổng tiền nước, tháng năm thanh toán ).*

* + - * + Hợp đồng ( **MaHopDong**, MaNV, MaSV, MaPhong, NgayNhanPhong, NgayTraPhong )

*( Mỗi hợp đồng phải có mã hợp đồng, mã nhân viên, mã sinh viên, mã phòng, ngày nhận phòng, ngày trả phòng ).*

* + - * + Phòng ( **MaPhong**, Tang, SucChua, LoaiPhong, TrangThaiPhong )

*( Mỗi phòng phải có mã phòng, tầng, sức chứa, loại phòng, trạng thái phòng).*

## Thiết kế dữ liệu.

* **Bảng Sinh Viên.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Khoá** | **Ghi chú** |
| MaSV | CHAR(9) | PRIMARY KEY | Mã sinh viên |
| MaNguoiThan | INT |  | Mã người thân |
| HoTen | NVARCHAR(50) |  | Họ tên |
| GioiTinh | NVARCHAR(3) |  | Giới tính |
| NgaySinh | DATE |  | Ngày sinh |
| Email | VARCHAR(100) |  | Email |
| DiaChi | NVARCHAR(100) |  | Địa chỉ |
| SoDienThoai | VARCHAR(12) |  | Số điện thoại |
| MaPhong | CHAR(4) |  | Mã phòng |

* **Bảng Nhân Viên.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Khoá** | **Ghi chú** |
| MaNV | CHAR(5) | PRIMARY KEY | Mã nhân viên |
| HoTen | NVARCHAR(50) |  | Họ tên |
| GioiTinh | NVARCHAR(3) |  | Giới tính |
| QueQuan | NVARCHAR(50) |  | Quê quán |
| Email | VARCHAR(100) |  | Email |
| ChucVu | NVARCHAR(50) |  | Chức vụ |
| SDT | CHAR(11) |  | Số điện thoại |

* **Bảng Người Thân.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Khoá** | **Ghi chú** |
| MaNguoiThan | INT | PRIMARY KEY | Mã người thân |
| HoTenNguoiThan | NVARCHAR(50) |  | Họ tên người thân |
| GioiTinh | NVARCHAR(3) |  | Giới tính |
| DiaChi | NVARCHAR(100) |  | Địa chỉ |
| SoDienThoai | CHAR(11) |  |  |
| Email | VARCHAR(100) |  |  |
| NgaySinh | DATE |  |  |
| MaSV | CHAR(9) |  |  |

* **Bảng Điện Nước.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Khoá** | **Ghi chú** |
| MaDienNuoc | INT | PRIMARY KEY | Mã điện nước |
| ChiSoDau\_Dien | INT |  | Chỉ số đầu của điện |
| ChiSoDau\_Nuoc | INT |  | Chỉ số đầu của nước |
| ChiSoCuoi\_Dien | INT |  | Chỉ số cuối của điện |
| ChiSoCuoi\_Nuoc | INT |  | Chỉ số cuối của nước |
| NgayBatDau | DATE |  | Ngày bắt đầu |
| NgayKetThuc | DATE |  | Ngày kết thúc |

* **Bảng Hóa Đơn.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Khóa** | **Ghi chú** |
| MaHD | INT | PRIMARY KEY | Mã hoá đơn |
| MaHopDong | CHAR(6) | PRIMARY KEY | Mã hợp đồng |
| MaDienNuoc | INT |  | Mã điện nước |
| TongTienDien | DECIMAL |  | Tổng tiền điện |
| TongTienNuoc | DECIMAL |  | Tổng tiền nước |
| ThangNamThanhToan | DATE |  | Tháng năm thanh toán |

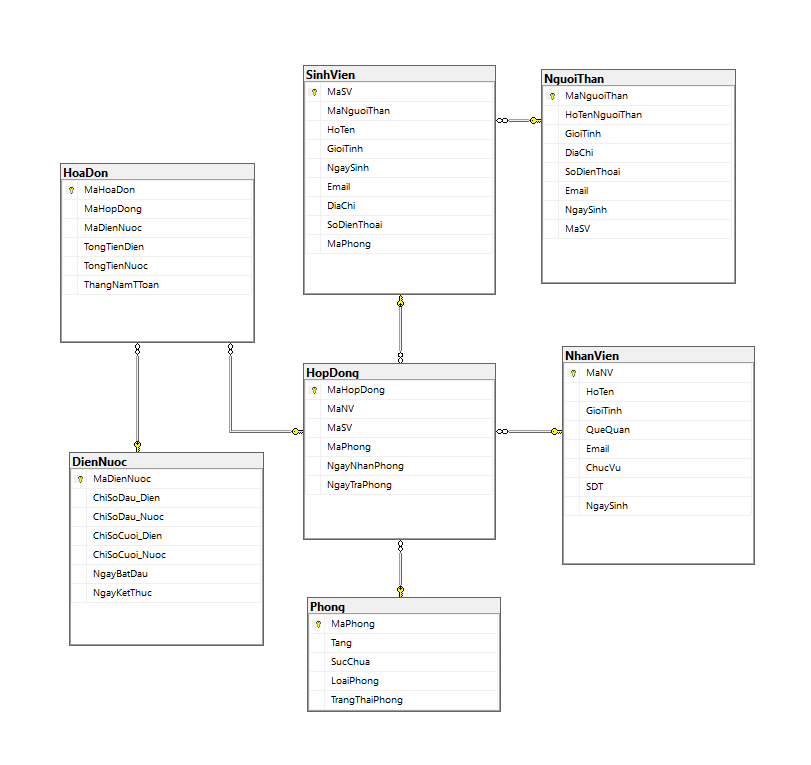
* **Bảng Hợp Đồng.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Khóa** | **Ghi chú** |
| MaHopDong | CHAR(6) | PRIMARY KEY | Mã hợp đồng |
| MaNV | CHAR(5) |  | Mã nhân viên |
| MaSV | CHAR(9) |  | Mã sinh viên |
| MaPhong | CHAR(4) |  | Mã phòng |
| NgayNhanPhong | DATE |  | Ngày nhận phòng |
| NgayTraPhong | DATE |  | Ngày trả phòng |

* **Bảng Phòng.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Khóa** | **Ghi chú** |
| MaPhong | CHAR(4) | PRIMARY KEY | Mã phòng |
| Tang | INT |  | Tầng |
| SucChua | INT |  | Sức chứa |
| LoaiPhong | NVARCHAR |  | Loại phòng |
| TrangThaiPhong | NVARCHAR |  | Trạng thái phòng |

# SƠ ĐỒ QUAN HỆ THỰC THỂ.



# CÀI ĐẶT CHƯƠNG TRÌNH

1. **From Đăng nhập hệ thống**

**– Giao diện:**



**– Code:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Do\_an

{

public partial class FrmTrangChu : Form

{

public FrmTrangChu()

{

InitializeComponent();

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

}

//nút sinh viên nhe

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

{

FrmSinhVien f1 = new FrmSinhVien();

this.Hide();

f1.ShowDialog();

this.Show();

}

//nút nhân viên nè

private void button2\_Click(object sender, EventArgs e)

{

FrmNhanVien f2 = new FrmNhanVien();

this.Hide();

f2.ShowDialog();

this.Show();

}

//nút phòng

private void button3\_Click(object sender, EventArgs e)

{

FrmPhongcs f3 = new FrmPhongcs();

this.Hide();

f3.ShowDialog();

this.Show();

}

//nút người thân nhe ba

private void button7\_Click(object sender, EventArgs e)

{

FrmNguoiThan f4 = new FrmNguoiThan();

this.Hide();

f4.ShowDialog();

this.Show();

}

//nút điện nước nè bà nội

private void button4\_Click(object sender, EventArgs e)

{

FrmDienNuoc f5 = new FrmDienNuoc();

this.Hide();

f5.ShowDialog();

this.Show();

}

//nút hoá đơn đây

private void button5\_Click(object sender, EventArgs e)

{

FrmHoáĐơn f6 = new FrmHoáĐơn();

this.Hide();

f6.ShowDialog();

this.Show();

}

private void button6\_Click(object sender, EventArgs e)

{

FrmHopDong f7 = new FrmHopDong();

this.Hide();

f7.ShowDialog();

this.Show();

}

private void button8\_Click(object sender, EventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Bạn muốn thoát ứng dụng không ?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Stop);

if (dialogResult == DialogResult.Yes)

{

FrmĐăng\_Nhập f8 = new FrmĐăng\_Nhập();

f8.Show();

this.Close();

}

else

{

//Ở lại trang chủ

}

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

}

}

}

1. **Form Trang chủ**

**– Giao diện:**



**– Code**:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Do\_an

{

public partial class FrmĐăng\_Nhập : Form

{

public FrmĐăng\_Nhập()

{

InitializeComponent();

}

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

{

if(txtTaiKhoan.Text == "" || txtMatKhau.Text == "")

{

MessageBox.Show("Nhập thông tin đầy đủ nào !!", "Nào nào nào ", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (txtTaiKhoan.Text != "User1234" || txtMatKhau.Text != "123456789")

{

MessageBox.Show("Tài khoản hoặc mật khẩu không chính xác", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (txtTaiKhoan.Text == "User1234" && txtMatKhau.Text == "123456789")

{

FrmTrangChu f\_TrangChu = new FrmTrangChu();

this.Hide();

f\_TrangChu.ShowDialog();

this.Show();

}

}

private void FrmĐăng\_Nhập\_Load(object sender, EventArgs e)

{

}

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

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

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

{

MessageBox.Show("Bạn muốn thoát ứng dụng không ?", "Khoan dừng lạii", MessageBoxButtons.OK);

Application.Exit();

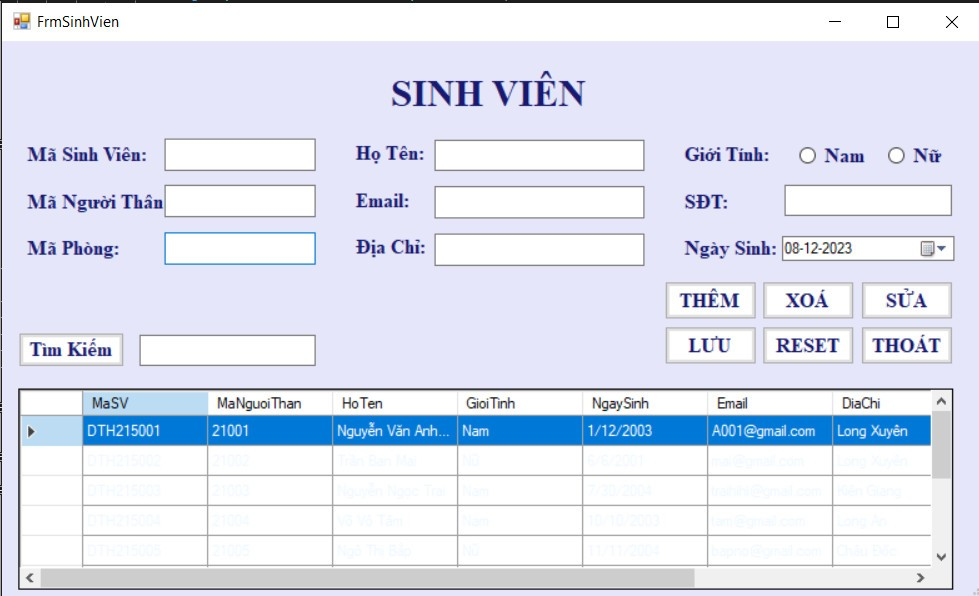
}

}

}

1. **Form Sinh viên**

**– Giao diện:**



**– Code:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Do\_an

{

public partial class FrmSinhVien : Form

{

string ChuoiKN = "Data Source=LAPTOP-1170Q5GN\\MSSQLSERVER01;Initial Catalog=QLKTX;Integrated Security=True";

SqlConnection conn;

public FrmSinhVien()

{

InitializeComponent();

}

private void OpenSQL()

{

conn = new SqlConnection(ChuoiKN);

conn.Open();

}

private void txtMaSV\_HD\_TextChanged(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

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

{

OpenSQL();

LoaddataShow();

dateTimePicker\_SV.Format = DateTimePickerFormat.Custom;

dateTimePicker\_SV.CustomFormat = "dd-MM-yyyy";

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void label9\_Click(object sender, EventArgs e)

{

}

private void LoaddataShow()

{

string str\_Open = @"select \* from SinhVien";

SqlDataAdapter adapter = new SqlDataAdapter(str\_Open, conn);

DataTable dt = new DataTable();

adapter.Fill(dt);

dgvThongTinSinhVien.DataSource = dt;

}

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

{

try

{

string str = @"insert into SinhVien values (@MaSV, @MaNguoiThan, @HoTen, @GioiTinh, @NgaySinh, @Email, @DiaChi, @SoienThoai, @MaPhong)";

if (txtMaSV\_SV.Text == "" || txtMaNT\_SV.Text == "" || txtMaPhong\_SV.Text == "" ||

txtHoten\_SV.Text == "" || txtDiaChi\_SV.Text == "" || txtEmail\_SV.Text == "" ||

txtSDT\_SV.Text == "" || dateTimePicker\_SV.Text == "" || (radNam\_SV.Checked == false && radNu\_SV.Checked == false))

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

else

{

using (SqlConnection connection = new SqlConnection(ChuoiKN))

{

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

{

connection.Open();

command.Parameters.AddWithValue("@MaSV", txtMaSV\_SV.Text);

command.Parameters.AddWithValue("@MaNguoiThan", txtMaNT\_SV.Text);

command.Parameters.AddWithValue("@HoTen", txtHoten\_SV.Text);

if (radNam\_SV.Checked)

{

command.Parameters.AddWithValue("@GioiTinh", "Nam");

}

else if (radNu\_SV.Checked)

{

command.Parameters.AddWithValue("@GioiTinh", "Nữ");

}

else

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

DateTime ngayDuocChon = dateTimePicker\_SV.Value;

string ngaysinh = ngayDuocChon.ToString("yyyy-MM-dd");

command.Parameters.AddWithValue("@Email", txtEmail\_SV.Text);

command.Parameters.AddWithValue("@DiaChi", txtDiaChi\_SV.Text);

command.Parameters.AddWithValue("@NgaySinh", ngaysinh);

command.Parameters.AddWithValue("@SoienThoai", txtSDT\_SV.Text);

command.Parameters.AddWithValue("@MaPhong", txtMaPhong\_SV.Text);

command.ExecuteNonQuery();

}

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

MessageBox.Show("Thêm thông tin sinh viên thành công");

LoaddataShow();

}

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

{

/\*

foreach (DataGridViewRow i in dgvThongTinSinhVien.SelectedRows)

{

if (!i.IsNewRow)

{

dgvThongTinSinhVien.Rows.RemoveAt(dgvThongTinSinhVien.SelectedRows[0].Index);

}

else

{

MessageBox.Show("Chưa chọn dữ liệu để xóa", "Thông báo", MessageBoxButtons.OK);

}

}\*/

string str\_MaSV = dgvThongTinSinhVien.SelectedRows[0].Cells["MaSV"].Value.ToString();

string str = @"delete from SinhVien where MaSV = @MaSV";

using (SqlCommand command = new SqlCommand(str, conn))

{

command.Parameters.AddWithValue("@MaSV", txtMaSV\_SV.Text);

try

{

command.ExecuteNonQuery();

MessageBox.Show("Đã xóa sinh viên!", "Thông báo", MessageBoxButtons.OK);

}

catch (Exception ex)

{

MessageBox.Show("Xóa lỗi", "Thông báo", MessageBoxButtons.OK);

}

}

LoaddataShow();

}

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

{

string str = "update SinhVien set MaNguoiThan = @MaNguoiThan, HoTen = @HoTen, GioiTinh = @GioiTinh, Email = @Email, DiaChi = @DiaChi, SoDienThoai = @SDT, MaPhong = @MaPhong where MaSV = @MaSV";

using (SqlConnection connection = new SqlConnection(ChuoiKN))

{

connection.Open();

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

{

command.Parameters.AddWithValue("@MaSV", txtMaSV\_SV.Text);

command.Parameters.AddWithValue("@HoTen", txtHoten\_SV.Text);

command.Parameters.AddWithValue("@MaNguoiThan", txtMaNT\_SV.Text);

//Chuyển giá trị của datime picker kiểu datetime sang định dạng năm-tháng-ngày để lưu vào csdl

command.Parameters.AddWithValue("@NgaySinh", dateTimePicker\_SV.Value.ToString("yyyy-MM-dd"));

command.Parameters.Add("@DiaChi", SqlDbType.NVarChar, 100).Value = txtDiaChi\_SV.Text;

command.Parameters.AddWithValue("@SDT", txtSDT\_SV.Text);

command.Parameters.AddWithValue("@MaPhong", txtMaPhong\_SV.Text);

command.Parameters.AddWithValue("@Email", txtEmail\_SV.Text);

string gioiTinh = radNam\_SV.Checked ? "Nam" : "Nữ";

command.Parameters.Add("@GioiTinh", SqlDbType.NVarChar, 3).Value = gioiTinh;

try

{

command.ExecuteNonQuery();

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

//MessageBox.Show("Lỗi", "Thông báo", MessageBoxButtons.OK);

}

MessageBox.Show("Sửa thông tin sinh viên thành công");

LoaddataShow();

}

}

}

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

{

try

{

DataGridViewRow dataGridViewRow = dgvThongTinSinhVien.SelectedRows[0];

txtMaSV\_SV.Text = dataGridViewRow.Cells["MaSV"].Value.ToString();

txtHoten\_SV.Text = dataGridViewRow.Cells["HoTen"].Value.ToString();

txtMaNT\_SV.Text = dataGridViewRow.Cells["MaNguoiThan"].Value.ToString();

txtMaPhong\_SV.Text = dataGridViewRow.Cells["MaPhong"].Value.ToString();

txtSDT\_SV.Text = dataGridViewRow.Cells["SoDienThoai"].Value.ToString();

dateTimePicker\_SV.Text = dataGridViewRow.Cells["NgaySinh"].Value.ToString();

txtEmail\_SV.Text = dataGridViewRow.Cells["Email"].Value.ToString();

radNu\_SV.Text = dataGridViewRow.Cells["GioiTinh"].Value.ToString();

radNam\_SV.Text = dataGridViewRow.Cells["GioiTinh"].Value.ToString();

}

catch

{

MessageBox.Show("Hãy chọn dữ liệu", "THông báo", MessageBoxButtons.OK);

}

}

private void btnRS\_SV\_Click(object sender, EventArgs e)

{

txtMaSV\_SV.Clear();

txtHoten\_SV.Clear();

txtMaNT\_SV.Clear();

txtMaPhong\_SV.Clear();

txtSDT\_SV.Clear();

txtEmail\_SV.Clear();

txtDiaChi\_SV.Clear();

radNam\_SV.Checked = false;

radNu\_SV.Checked = false;

DateTime dt = new DateTime(2000, 1, 1);

dateTimePicker\_SV.Value = dt;

}

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

{

FrmTrangChu f = new FrmTrangChu();

f.Show();

this.Close();

}

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

{

}

private void btFind\_SV\_Click(object sender, EventArgs e)

{

try

{

string str\_Tim = @"select \* from SinhVien where MaSV like '" + txtTKiem\_SV.Text + "%'";

SqlDataAdapter dataAdapter = new SqlDataAdapter(str\_Tim, conn);

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

dgvThongTinSinhVien.DataSource = dataTable;

}

catch { MessageBox.Show("Hãy nhập mã số nhân viên cần tìm", "Thông báo", MessageBoxButtons.OK); }

}

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

{

txtMaSV\_SV.Text= dgvThongTinSinhVien.SelectedRows[0].Cells[0].Value.ToString();

txtMaNT\_SV.Text = dgvThongTinSinhVien.SelectedRows[0].Cells[1].Value.ToString();

txtMaPhong\_SV.Text = dgvThongTinSinhVien.SelectedRows[0].Cells[8].Value.ToString();

txtHoten\_SV.Text = dgvThongTinSinhVien.SelectedRows[0].Cells[2].Value.ToString();

txtEmail\_SV.Text = dgvThongTinSinhVien.SelectedRows[0].Cells[5].Value.ToString();

txtDiaChi\_SV.Text = dgvThongTinSinhVien.SelectedRows[0].Cells[6].Value.ToString();

txtSDT\_SV.Text = dgvThongTinSinhVien.SelectedRows[0].Cells[7].Value.ToString();

string gioiTinh = dgvThongTinSinhVien.SelectedRows[0].Cells[3].Value.ToString();

if(gioiTinh.Equals("Nam"))

radNam\_SV.Checked = true;

else radNu\_SV.Checked = true;

dateTimePicker\_SV.Text = dgvThongTinSinhVien.SelectedRows[0].Cells[4].Value.ToString();

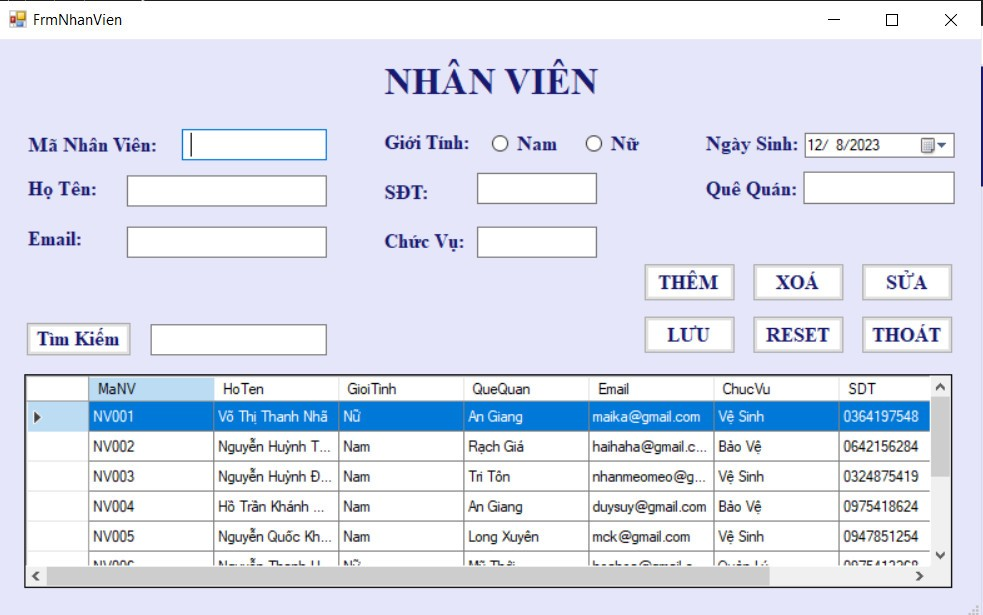
}

}

}

1. **From Nhân viên:**

**– Giao diện:**



**– Code:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Do\_an

{

public partial class FrmNhanVien : Form

{

string ChuoiKN = "Data Source=LAPTOP-1170Q5GN\\MSSQLSERVER01;Initial Catalog=QLKTX;Integrated Security=True";

SqlConnection conn;

string maNV\_Currency;

public FrmNhanVien()

{

InitializeComponent();

}

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

{

OpenSQL();

LoaddataShow();

}

private void OpenSQL()

{

conn = new SqlConnection(ChuoiKN);

conn.Open();

}

private void LoaddataShow()

{

string str\_Open = @"select \* from NhanVien";

SqlDataAdapter adapter = new SqlDataAdapter(str\_Open, conn);

DataTable dt = new DataTable();

adapter.Fill(dt);

dgDS\_NV.DataSource = dt;

}

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

{

}

private void btnLuu\_NT\_Click(object sender, EventArgs e)

{

}

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

{

FrmTrangChu f = new FrmTrangChu();

f.Show();

this.Close();

}

private void btnHuy\_NT\_Click(object sender, EventArgs e)

{

txtMaNV\_NV.Clear();

txtHoten\_NV.Clear();

txtQueQuan\_NV.Clear();

txtEmail\_NV.Clear();

txtSDT\_NV.Clear();

txtChucVu\_NV.Clear();

//radNam\_NV.Checked = false;

//radNu\_NV.Checked = false;

DateTime dt = new DateTime(2000, 1, 1);

dtpNgaySinh.Value = dt;

}

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

{

try

{

string str = @"insert into NhanVien values (@MaNV, @HoTen, @GioiTinh, @QueQuan, @Email, @ChucVu, @SDT, @NgaySinh)";

if (txtMaNV\_NV.Text == "" || txtHoten\_NV.Text == "" || txtQueQuan\_NV.Text == "" ||

txtEmail\_NV.Text == "" || txtSDT\_NV.Text == "" ||

txtChucVu\_NV.Text == "" || (radNam\_NV.Checked == false && radNu\_NV.Checked == false))

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

else

{

using (SqlConnection connection = new SqlConnection(ChuoiKN))

{

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

{

command.Parameters.AddWithValue("@MaNV", txtMaNV\_NV.Text);

command.Parameters.AddWithValue("@HoTen", txtHoten\_NV.Text);

command.Parameters.AddWithValue("@QueQuan", txtQueQuan\_NV.Text);

command.Parameters.AddWithValue("@Email", txtEmail\_NV.Text);

command.Parameters.AddWithValue("@SDT", txtSDT\_NV.Text);

command.Parameters.AddWithValue("@NgaySinh", dtpNgaySinh.Value.ToString("yyyy-MM-dd"));

command.Parameters.AddWithValue("@ChucVu", txtChucVu\_NV.Text);

if (radNam\_NV.Checked)

{

command.Parameters.AddWithValue("@GioiTinh", radNam\_NV.Text);

}

else if (radNu\_NV.Checked)

{

command.Parameters.AddWithValue("@GioiTinh", radNu\_NV.Text);

}

else

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

connection.Open();

command.ExecuteNonQuery();

connection.Close();

}

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

MessageBox.Show("Thêm thông tin nhân viên thành công");

LoaddataShow();

}

private void label4\_Click(object sender, EventArgs e)

{

}

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

{

foreach (DataGridViewRow i in dgDS\_NV.SelectedRows)

{

if (!i.IsNewRow)

{

string str\_MaNV = dgDS\_NV.SelectedRows[0].Cells["MaNV"].Value.ToString();

string str = @"delete from SinhVien where MaNV = @MaNV";

using (SqlCommand command = new SqlCommand(str, conn))

{

command.Parameters.AddWithValue("@MaNV", txtMaNV\_NV.Text);

try

{

command.ExecuteNonQuery();

}

catch

{

MessageBox.Show("Xóa lỗi", "Thông báo", MessageBoxButtons.OK);

}

}

dgDS\_NV.Rows.RemoveAt(dgDS\_NV.SelectedRows[0].Index);

}

else

{

MessageBox.Show("Chưa chọn dữ liệu để xóa", "Thông báo", MessageBoxButtons.OK);

}

}

}

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

{

string str = @"UPDATE NhanVien SET MaNV=@MaNV, HoTen=@HoTen, GioiTinh=@GioiTinh, QueQuan=@QueQuan, Email=@Email, ChucVu=@ChucVu, SDT=@SDT, NgaySinh=@NgaySinh where MaNV = @MaNV";

using (SqlConnection connection = new SqlConnection(ChuoiKN))

{

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

{

command.Parameters.AddWithValue("@MaNV", txtMaNV\_NV.Text);

command.Parameters.AddWithValue("@HoTen", txtHoten\_NV.Text);

command.Parameters.AddWithValue("@QueQuan", txtQueQuan\_NV.Text);

command.Parameters.AddWithValue("@Email", txtEmail\_NV.Text);

command.Parameters.AddWithValue("@SDT", txtSDT\_NV.Text);

command.Parameters.AddWithValue("@NgaySinh", dtpNgaySinh.Text);

command.Parameters.AddWithValue("@ChucVu", txtChucVu\_NV.Text);

if (radNam\_NV.Checked)

{

command.Parameters.AddWithValue("@GioiTinh", radNam\_NV.Text);

}

else if (radNu\_NV.Checked)

{

command.Parameters.AddWithValue("@GioiTinh", radNu\_NV.Text);

}

else

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

connection.Open();

try

{

command.ExecuteNonQuery();

}

catch

{

MessageBox.Show("Lỗi", "Thông báo", MessageBoxButtons.OK);

}

connection.Close();

LoaddataShow();

}

}

}

private void btFind\_NV\_Click(object sender, EventArgs e)

{

try

{

string str\_Tim = @"select \* from NhanVien where MaNV like '" + txtTKiem\_NV.Text + "%'";

SqlDataAdapter dataAdapter = new SqlDataAdapter(str\_Tim, conn);

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

dgDS\_NV.DataSource = dataTable;

}

catch { MessageBox.Show("Hãy nhập mã số nhân viên cần tìm", "Thông báo", MessageBoxButtons.OK); }

}

private void dgDS\_NV\_Click(object sender, EventArgs e)

{

maNV\_Currency = txtMaNV\_NV.Text;

txtMaNV\_NV.Text = dgDS\_NV.SelectedRows[0].Cells[0].Value.ToString();

txtHoten\_NV.Text = dgDS\_NV.SelectedRows[0].Cells[1].Value.ToString();

string gioiTinh = dgDS\_NV.SelectedRows[0].Cells[2].Value.ToString();

if (gioiTinh.Equals("Nam"))

radNam\_NV.Checked = true;

else radNu\_NV.Checked = true;

txtQueQuan\_NV.Text = dgDS\_NV.SelectedRows[0].Cells[3].Value.ToString();

txtEmail\_NV.Text = dgDS\_NV.SelectedRows[0].Cells[4].Value.ToString();

txtChucVu\_NV.Text = dgDS\_NV.SelectedRows[0].Cells[5].Value.ToString();

txtSDT\_NV.Text = dgDS\_NV.SelectedRows[0].Cells[6].Value.ToString();

}

private void radNam\_NV\_CheckedChanged(object sender, EventArgs e)

{

}

}

}

1. **Form Phòng**

**– Giao diện**



**– Code:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Do\_an

{

public partial class FrmPhongcs : Form

{

string ChuoiKN = "Data Source=LAPTOP-1170Q5GN\\MSSQLSERVER01;Initial Catalog=QLKTX;Integrated Security=True";

SqlConnection conn;

private void OpenSQL()

{

conn = new SqlConnection(ChuoiKN);

conn.Open();

}

public FrmPhongcs()

{

InitializeComponent();

}

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

{

OpenSQL();

LoaddataShow();

}

private void LoaddataShow()

{

string str\_Open = @"select \* from Phong";

SqlDataAdapter adapter = new SqlDataAdapter(str\_Open, conn);

DataTable dt = new DataTable();

adapter.Fill(dt);

dgDS\_Phong.DataSource = dt;

}

private void txtDiaChi\_SV\_TextChanged(object sender, EventArgs e)

{

}

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

{

FrmTrangChu f = new FrmTrangChu();

f.Show();

this.Close();

}

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

{

try

{

string str = @"insert into Phong values (@MaPhong,@Tang,@SucChua, @LoaiPhong, @TrangThaiPhong)";

if (txtMaPhong\_Phong.Text == "" || txtTang\_Phong.Text == "" || (radioChuaday.Checked == false && radioDay.Checked == false) ||

(radMLanh\_Phong.Checked == false && radMQuat\_Phong.Checked == false) || (radSC4\_Phong.Checked == false && radSC6\_Phong.Checked == false))

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

else

{

using (SqlConnection connection = new SqlConnection(ChuoiKN))

{

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

{

connection.Open();

command.Parameters.AddWithValue("@MaPhong", txtMaPhong\_Phong.Text);

command.Parameters.AddWithValue("@Tang", txtTang\_Phong.Text);

if (radioChuaday.Checked)

{

command.Parameters.AddWithValue("@TrangThaiPhong", radioChuaday.Text);

}

else if(radioDay.Checked)

{

command.Parameters.AddWithValue("@TrangThaiPhong", radioDay.Text);

}

else

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

if (radMLanh\_Phong.Checked)

{

command.Parameters.AddWithValue("@LoaiPhong", radMLanh\_Phong.Text);

}

else if (radMQuat\_Phong.Checked)

{

command.Parameters.AddWithValue("@LoaiPhong", radMQuat\_Phong.Text);

}

else

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

if (radSC4\_Phong.Checked)

{

command.Parameters.AddWithValue("@SucChua", int.Parse(radSC4\_Phong.Text));

}

else if (radSC6\_Phong.Checked)

{

command.Parameters.AddWithValue("@SucChua", int.Parse(radSC6\_Phong.Text));

}

else

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

command.ExecuteNonQuery();

}

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

LoaddataShow();

}

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

{

foreach (DataGridViewRow i in dgDS\_Phong.SelectedRows)

{

if (!i.IsNewRow)

{

string str\_MaPhong = dgDS\_Phong.SelectedRows[0].Cells["Maphong"].Value.ToString();

string str = @"delete from Phong where MaPhong = @MaPhong";

using (SqlCommand command = new SqlCommand(str, conn))

{

command.Parameters.AddWithValue("@Maphong", txtMaPhong\_Phong.Text);

try

{

command.ExecuteNonQuery();

}

catch

{

MessageBox.Show("Xóa lỗi", "Thông báo", MessageBoxButtons.OK);

}

}

dgDS\_Phong.Rows.RemoveAt(dgDS\_Phong.SelectedRows[0].Index);

}

else

{

MessageBox.Show("Chưa chọn dữ liệu để xóa", "Thông báo", MessageBoxButtons.OK);

}

}

}

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

{

string str = string.Format(@"UPDATE Phong SET Tang=@Tang, TrangThaiPhong=@TrangThaiPhong, LoaiPhong=@LoaiPhong, SucChua=@SucChua where MaPhong='{0}'", txtMaPhong\_Phong.Text);

using (SqlConnection connection = new SqlConnection(ChuoiKN))

{

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

{

connection.Open();

command.Parameters.AddWithValue("@Tang", txtTang\_Phong.Text);

if (radioChuaday.Checked)

{

command.Parameters.AddWithValue("@TrangThaiPhong", radioChuaday.Text);

}

else if (radioDay.Checked)

{

command.Parameters.AddWithValue("@TrangThaiPhong", radioDay.Text);

}

else

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

if (radMLanh\_Phong.Checked)

{

command.Parameters.AddWithValue("@LoaiPhong", radMLanh\_Phong.Text);

}

else if (radMQuat\_Phong.Checked)

{

command.Parameters.AddWithValue("@LoaiPhong", radMQuat\_Phong.Text);

}

else

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

if (radSC4\_Phong.Checked)

{

command.Parameters.AddWithValue("@SucChua", int.Parse(radSC4\_Phong.Text));

}

else if (radSC6\_Phong.Checked)

{

command.Parameters.AddWithValue("@SucChua", int.Parse(radSC6\_Phong.Text));

}

else

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

//command.ExecuteNonQuery();

try

{

command.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

LoaddataShow();

}

}

}

private void btnHuy\_Phong\_Click(object sender, EventArgs e)

{

txtMaPhong\_Phong.Clear();

txtTang\_Phong.Clear();

radMLanh\_Phong.Enabled= false;

radMQuat\_Phong.Enabled = false;

radioChuaday.Enabled = false;

radioDay.Enabled = false;

radSC4\_Phong.Enabled = false;

radSC6\_Phong.Enabled = false;

}

private void btFind\_Phong\_Click(object sender, EventArgs e)

{

try

{

string str\_Tim = @"select \* from Phong where MaPhong like '" + txtTKiem\_Phong.Text + "%'";

SqlDataAdapter dataAdapter = new SqlDataAdapter(str\_Tim, conn);

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

dgDS\_Phong.DataSource = dataTable;

}

catch { MessageBox.Show("Hãy nhập mã số nhân viên cần tìm", "Thông báo", MessageBoxButtons.OK); }

}

private void dgDS\_Phong\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void dgDS\_Phong\_Click(object sender, EventArgs e)

{

txtMaPhong\_Phong.Text = dgDS\_Phong.SelectedRows[0].Cells[0].Value.ToString();

txtTang\_Phong.Text = dgDS\_Phong.SelectedRows[0].Cells[1].Value.ToString();

string sucChua = dgDS\_Phong.SelectedRows[0].Cells[2].Value.ToString();

if (sucChua.Equals("4"))

radSC4\_Phong.Checked = true;

else radSC6\_Phong.Checked = true;

string loaiPhong = dgDS\_Phong.SelectedRows[0].Cells[3].Value.ToString();

if(loaiPhong.Equals("Quạt"))

radMQuat\_Phong.Checked = true;

else radMLanh\_Phong.Checked = true;

string trangThaiPhong = dgDS\_Phong.SelectedRows[0].Cells[4].Value.ToString();

if (trangThaiPhong.Equals("Đầy"))

radioDay.Checked = true;

else

radioChuaday.Checked = true;

}

}

}

1. **Form Người thân**

**– Giao diện:**



**– Code:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Do\_an

{

public partial class FrmNguoiThan : Form

{

string ChuoiKN = "Data Source=LAPTOP-1170Q5GN\\MSSQLSERVER01;Initial Catalog=QLKTX;Integrated Security=True";

SqlConnection conn;

public FrmNguoiThan()

{

InitializeComponent();

}

private void OpenSQL()

{

conn = new SqlConnection(ChuoiKN);

conn.Open();

}

private void Loaddatashow()

{

string str\_Open = @"select \* from NguoiThan";

SqlDataAdapter adapter = new SqlDataAdapter(str\_Open, conn);

DataTable dt = new DataTable();

adapter.Fill(dt);

dgDS\_NguoiThan.DataSource = dt;

}

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

{

dtpNgaySinh.Format = DateTimePickerFormat.Custom;

dtpNgaySinh.CustomFormat = "dd-MM-yyyy";

OpenSQL();

Loaddatashow();

}

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

{

FrmTrangChu f = new FrmTrangChu();

f.Show();

this.Close();

}

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

{

string str = @"insert into NguoiThan values (@MaNguoiThan, @HoTenNguoiThan, @GioiTinh, @DiaChi, @SoDienThoai, @Email, @NgaySinh, @MaSV)";

if (txtMaSV\_NT.Text == "" || txtMaNT\_NT.Text == "" || txtDiaChi\_NT.Text == "" ||

txtHoten\_NT.Text == "" || txtEmail\_NT.Text == "" ||

txtSDT\_NT.Text == "" || (radNam\_NT.Checked == false && radNu\_NT.Checked == false))

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

else

{

using (SqlConnection connection = new SqlConnection(ChuoiKN))

{

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

{

command.Parameters.AddWithValue("@MaSV", txtMaSV\_NT.Text);

command.Parameters.AddWithValue("@HoTenNguoiThan", txtHoten\_NT.Text);

command.Parameters.AddWithValue("@MaNguoiThan", txtMaNT\_NT.Text);

command.Parameters.AddWithValue("@NgaySinh", dtpNgaySinh.Value.ToString("yyyy-MM-dd"));

command.Parameters.AddWithValue("@SoDienThoai", txtSDT\_NT.Text);

command.Parameters.AddWithValue("@DiaChi", txtDiaChi\_NT.Text);

command.Parameters.AddWithValue("@Email", txtEmail\_NT.Text);

if (radNam\_NT.Checked)

{

command.Parameters.AddWithValue("@GioiTinh", radNam\_NT.Text);

}

else if (radNu\_NT.Checked)

{

command.Parameters.AddWithValue("@GioiTinh", radNu\_NT.Text);

}

else

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

connection.Open();

try

{

command.ExecuteNonQuery();

MessageBox.Show("Thêm thành công", "Thông báo", MessageBoxButtons.OK);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

Loaddatashow();

}

}

}

}

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

{

string str = @"UPDATE NguoiThan SET MaNguoiThan=@MaNguoiThan, HoTenNguoiThan=@HoTenNguoiThan, MaSV=@MaSV, NgaySinh=@NgaySinh, SoDienThoai=@SoienThoai, DiaChi=@DiaChi, Email=@Email, GioiTinh=@GioiTinh where MaNguoiThan=@MaNguoiThan";

using (SqlConnection connection = new SqlConnection(ChuoiKN))

{

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

{

command.Parameters.AddWithValue("@MaSV", txtMaSV\_NT.Text);

command.Parameters.AddWithValue("@HoTenNguoiThan", txtHoten\_NT.Text);

command.Parameters.AddWithValue("@MaNguoiThan", txtMaNT\_NT.Text);

command.Parameters.AddWithValue("@NgaySinh", dtpNgaySinh.Value.ToString("yyyy-MM-dd"));

command.Parameters.AddWithValue("@SoienThoai", txtSDT\_NT.Text);

command.Parameters.AddWithValue("@DiaChi", txtDiaChi\_NT.Text);

command.Parameters.AddWithValue("@Email", txtEmail\_NT.Text);

if (radNam\_NT.Checked)

{

command.Parameters.AddWithValue("@GioiTinh", radNam\_NT.Text);

}

else if (radNu\_NT.Checked)

{

command.Parameters.AddWithValue("@GioiTinh", radNu\_NT.Text);

}

else

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

connection.Open();

try

{

command.ExecuteNonQuery();

MessageBox.Show("Sửa thành công", "Thông báo", MessageBoxButtons.OK);

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

Loaddatashow();

}

}

}

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

{

foreach (DataGridViewRow i in dgDS\_NguoiThan.SelectedRows)

{

if (!i.IsNewRow)

{

string str\_MaNT = dgDS\_NguoiThan.SelectedRows[0].Cells["MaNT"].Value.ToString();

string str = @"delete from NguoiThan where MaNT = @MaNT";

using (SqlCommand command = new SqlCommand(str, conn))

{

command.Parameters.AddWithValue("@MaNT", txtMaNT\_NT.Text);

try

{

command.ExecuteNonQuery();

}

catch

{

MessageBox.Show("Xóa lỗi", "Thông báo", MessageBoxButtons.OK);

}

}

dgDS\_NguoiThan.Rows.RemoveAt(dgDS\_NguoiThan.SelectedRows[0].Index);

}

else

{

MessageBox.Show("Chưa chọn dữ liệu để xóa", "Thông báo", MessageBoxButtons.OK);

}

}

}

private void btnHuy\_NT\_Click(object sender, EventArgs e)

{

txtMaSV\_NT.Clear();

txtHoten\_NT.Clear();

txtMaNT\_NT.Clear();

txtDiaChi\_NT.Clear();

txtSDT\_NT.Clear();

//txtNgaySinh\_NT.Clear();

txtEmail\_NT.Clear();

radNam\_NT.Enabled = false;

radNu\_NT.Enabled = false;

}

private void btFind\_NT\_Click(object sender, EventArgs e)

{

try

{

string str\_Tim = @"select \* from NguoiThan where MaNguoiThan like '" + txtTKiem\_NT.Text + "%'";

SqlDataAdapter dataAdapter = new SqlDataAdapter(str\_Tim, conn);

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

dgDS\_NguoiThan.DataSource = dataTable;

}

catch { MessageBox.Show("Hãy nhập mã số nhân viên cần tìm", "Thông báo", MessageBoxButtons.OK); }

}

private void dgDS\_NguoiThan\_Click(object sender, EventArgs e)

{

txtMaSV\_NT.Text = dgDS\_NguoiThan.SelectedRows[0].Cells[7].Value.ToString();

txtMaNT\_NT.Text = dgDS\_NguoiThan.SelectedRows[0].Cells[0].Value.ToString();

//txtMaPhong.Text = dgvThongTinSinhVien.SelectedRows[0].Cells[8].Value.ToString();

txtHoten\_NT.Text = dgDS\_NguoiThan.SelectedRows[0].Cells[1].Value.ToString();

txtEmail\_NT.Text = dgDS\_NguoiThan.SelectedRows[0].Cells[5].Value.ToString();

txtDiaChi\_NT.Text = dgDS\_NguoiThan.SelectedRows[0].Cells[3].Value.ToString();

txtSDT\_NT.Text = dgDS\_NguoiThan.SelectedRows[0].Cells[4].Value.ToString();

string gioiTinh = dgDS\_NguoiThan.SelectedRows[0].Cells[2].Value.ToString();

if (gioiTinh.Equals("Nam"))

radNam\_NT.Checked = true;

else radNu\_NT.Checked = true;

dtpNgaySinh.Value = (DateTime)dgDS\_NguoiThan.SelectedRows[0].Cells[6].Value;

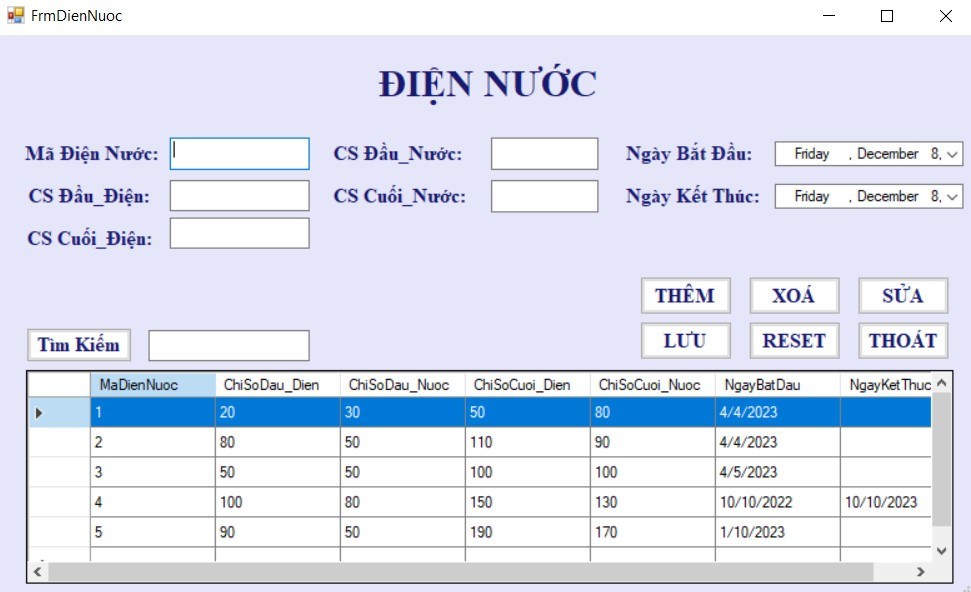
}

}

}

1. **Form Điện nước**

**– Giao diện:**



– Code:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Do\_an

{

public partial class FrmDienNuoc : Form

{

public FrmDienNuoc()

{

InitializeComponent();

}

string ChuoiKN = "Data Source=LAPTOP-1170Q5GN\\MSSQLSERVER01;Initial Catalog=QLKTX;Integrated Security=True";

SqlConnection conn;

private void OpenSQL()

{

conn = new SqlConnection(ChuoiKN);

conn.Open();

}

private void LoaddataShow()

{

string str\_Open = @"select \* from DienNuoc";

SqlDataAdapter adapter = new SqlDataAdapter(str\_Open, conn);

DataTable dt = new DataTable();

adapter.Fill(dt);

dgDS\_DN.DataSource = dt;

}

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

{

OpenSQL();

LoaddataShow();

}

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

{

FrmTrangChu f = new FrmTrangChu();

f.Show();

this.Close();

}

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

{

string str = @"UPDATE DienNuoc SET MaDienNuoc=@MaDienNuoc,

ChiSoDau\_Dien=@ChiSoDau\_Dien, ChiSoDau\_Nuoc=@ChiSoDau\_Nuoc,

ChiSoCuoi\_Dien=@ChiSoCuoi\_Dien, ChiSoCuoi\_Nuoc=@ChiSoCuoi\_Nuoc,

NgayBatDau=@NgayBatDau, NgayKetThuc=@NgayKetThuc where MaDienNuoc = @MaDienNuoc ";

using (SqlConnection connection = new SqlConnection(ChuoiKN))

{

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

{

command.Parameters.AddWithValue("@MaDienNuoc", txtMaDienNuoc\_DN.Text);

command.Parameters.AddWithValue("@ChiSoDau\_Dien", txtCSD\_Dien.Text);

command.Parameters.AddWithValue("@ChiSoDau\_Nuoc", txtCSD\_Nuoc.Text);

command.Parameters.AddWithValue("@ChiSoCuoi\_Dien", txtCSC\_Dien.Text);

command.Parameters.AddWithValue("@ChiSoCuoi\_Nuoc", txtCSC\_Nuoc.Text);

command.Parameters.AddWithValue("@NgayBatDau", dtp\_NBT\_DN.Value.ToString("yyyy-MM-dd"));// trở về ngày mặc định

command.Parameters.AddWithValue("@NgayKetThuc", dtp\_NKT\_DN.Value.ToString("yyyy-MM-dd"));// trở về ngày mặc định

connection.Open();

try

{

MessageBox.Show("Sửa thành công", "Thông báo", MessageBoxButtons.OK);

command.ExecuteNonQuery();

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

LoaddataShow();

}

}

}

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

{

foreach (DataGridViewRow i in dgDS\_DN.SelectedRows)

{

if (!i.IsNewRow)

{

string str\_MaDienNuoc = dgDS\_DN.SelectedRows[0].Cells["MaDienNuoc"].Value.ToString();

string str = @"delete from DienNuoc where MaDienNuoc = @MaDienNuoc";

using (SqlCommand command = new SqlCommand(str, conn))

{

command.Parameters.AddWithValue("@MaDienNuoc", txtMaDienNuoc\_DN.Text);

try

{

command.ExecuteNonQuery();

}

catch

{

MessageBox.Show("Xóa lỗi", "Thông báo", MessageBoxButtons.OK);

}

}

dgDS\_DN.Rows.RemoveAt(dgDS\_DN.SelectedRows[0].Index);

}

else

{

MessageBox.Show("Chưa chọn dữ liệu để xóa", "Thông báo", MessageBoxButtons.OK);

}

}

}

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

{

string str = @"insert into DienNuoc values (@MaDienNuoc, @ChiSoDau\_Dien, @ChiSoDau\_Nuoc, @ChiSoCuoi\_Dien, @ChiSoCuoi\_Nuoc, @NgayBatDau, @NgayKetThuc )";

if (txtMaDienNuoc\_DN.Text == "" || txtCSD\_Dien.Text == "" ||

txtCSD\_Nuoc.Text == "" || txtCSC\_Dien.Text == "" || txtCSC\_Nuoc.Text == "" ||

dtp\_NBT\_DN.Text == "" || dtp\_NKT\_DN.Text == "")

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

else

{

using (SqlConnection connection = new SqlConnection(ChuoiKN))

{

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

{

command.Parameters.AddWithValue("@MaDienNuoc", txtMaDienNuoc\_DN.Text);

command.Parameters.AddWithValue("@ChiSoDau\_Dien", int.Parse(txtCSD\_Dien.Text));

command.Parameters.AddWithValue("@ChiSoDau\_Nuoc", int.Parse(txtCSD\_Nuoc.Text));

command.Parameters.AddWithValue("@ChiSoCuoi\_Dien",int.Parse(txtCSC\_Dien.Text));

command.Parameters.AddWithValue("@ChiSoCuoi\_Nuoc",int.Parse(txtCSC\_Nuoc.Text));

command.Parameters.AddWithValue("@NgayBatDau", dtp\_NBT\_DN.Value.ToString("yyyy-MM-dd"));// trở về ngày mặc định

command.Parameters.AddWithValue("@NgayKetThuc", dtp\_NKT\_DN.Value.ToString("yyyy-MM-dd"));// trở về ngày mặc định

connection.Open();

try

{

command.ExecuteNonQuery();

MessageBox.Show("Thêm thành công", "Thông báo", MessageBoxButtons.OK);

}

catch(Exception ex )

{

MessageBox.Show(ex.Message);

}

LoaddataShow();

}

}

}

}

private void btnHuy\_DN\_Click(object sender, EventArgs e)

{

txtMaDienNuoc\_DN.Clear();

txtCSD\_Dien.Clear();

txtCSD\_Nuoc.Clear();

txtCSC\_Dien.Clear();

txtCSC\_Nuoc.Clear();

DateTime dt1 = new DateTime(2000,1,1);

dtp\_NBT\_DN.Value = dt1;

DateTime dt2 = new DateTime(2000, 1, 1);

dtp\_NKT\_DN.Value = dt2;

}

private void dgDS\_DN\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void btFind\_DN\_Click(object sender, EventArgs e)

{

try

{

string str\_Tim = @"select \* from DienNuoc where MaDienNuoc like '" + txtTKiem\_DN.Text + "%'";

SqlDataAdapter dataAdapter = new SqlDataAdapter(str\_Tim, conn);

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

dgDS\_DN.DataSource = dataTable;

}

catch { MessageBox.Show("Hãy nhập mã số nhân viên cần tìm", "Thông báo", MessageBoxButtons.OK); }

}

private void txtCSD\_Nuoc\_TextChanged(object sender, EventArgs e)

{

}

private void dgDS\_DN\_Click(object sender, EventArgs e)

{

txtMaDienNuoc\_DN.Text = dgDS\_DN.SelectedRows[0].Cells[0].Value.ToString();

txtCSD\_Dien.Text = dgDS\_DN.SelectedRows[0].Cells[1].Value.ToString();

txtCSD\_Nuoc.Text = dgDS\_DN.SelectedRows[0].Cells[2].Value.ToString();

txtCSC\_Dien.Text = dgDS\_DN.SelectedRows[0].Cells[3].Value.ToString();

txtCSC\_Nuoc.Text = dgDS\_DN.SelectedRows[0].Cells[4].Value.ToString();

dtp\_NBT\_DN.Text = dgDS\_DN.SelectedRows[0].Cells[5].Value.ToString();

dtp\_NKT\_DN.Text = dgDS\_DN.SelectedRows[0].Cells[6].Value.ToString();

}

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

{

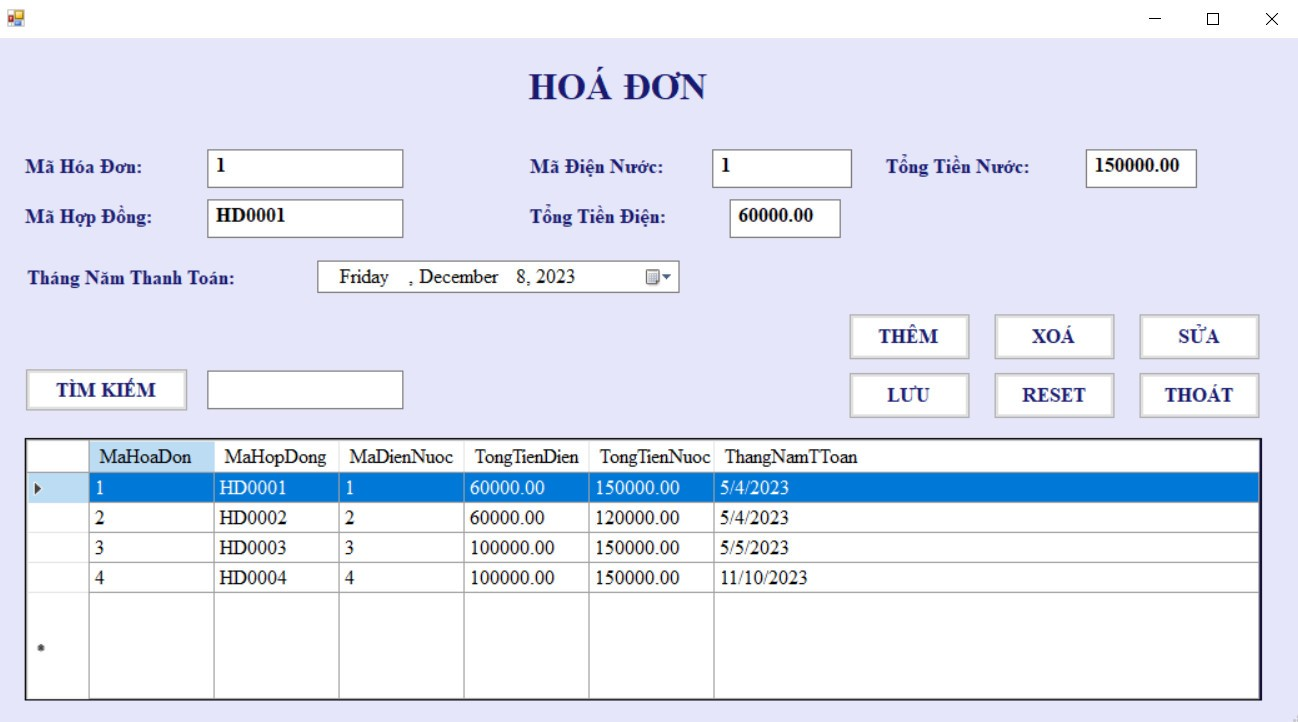
}

}

}

1. **From Hóa đơn**

**– Giao diện:**



**– Code:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Do\_an

{

public partial class FrmHoáĐơn : Form

{

public int ChiSoDau\_Nuoc, ChiSoDau\_Dien, ChiSoCuoi\_Dien, ChiSoCuoi\_Nuoc;

public FrmHoáĐơn()

{

InitializeComponent();

}

string ChuoiKN = "Data Source=LAPTOP-1170Q5GN\\MSSQLSERVER01;Initial Catalog=QLKTX;Integrated Security=True";

SqlConnection conn;

private void OpenSQL()

{

conn = new SqlConnection(ChuoiKN);

conn.Open();

}

private void LoaddataShow()

{

string str\_Open = @"select \* from HoaDon";

SqlDataAdapter adapter = new SqlDataAdapter(str\_Open, conn);

DataTable dt = new DataTable();

adapter.Fill(dt);

dgDS\_HoaDon.DataSource = dt;

}

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

{

}

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

{

OpenSQL();

LoaddataShow();

}

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

{

FrmTrangChu f = new FrmTrangChu();

f.Show();

this.Close();

}

//Tổng tiền nước

private void txtMaPhong\_HD\_TextChanged(object sender, EventArgs e)

{

}

// Tổng tiền điện

private void txtTTDien\_HD\_TextChanged(object sender, EventArgs e)

{

}

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

{

{

string str = @"insert into HoaDon values (@MaHoaDon, @MaHopDong,@MaDienNuoc, @TongTienDien, @TongTienNuoc, @ThangNamTToan)";

if (txtMaHD\_HD.Text == "" || txtTTN\_HD.Text == "" || txtMaHopDong\_HD.Text == "" ||

txtTTDien\_HD.Text == "" || dtp\_TNTToan\_HD.Text == ""|| txtMaDNuoc\_HD.Text=="")

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

else

{

using (SqlConnection connection = new SqlConnection(ChuoiKN))

{

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

{

command.Parameters.AddWithValue("@MaHoaDon", txtMaHD\_HD.Text);

command.Parameters.AddWithValue("@MaPhong", txtTTN\_HD.Text);

command.Parameters.AddWithValue("@MaHopDong", txtMaHopDong\_HD.Text);

command.Parameters.AddWithValue("@TongTienDien", txtTTDien\_HD.Text);

command.Parameters.AddWithValue("@TongTienNuoc", txtTTN\_HD.Text);

command.Parameters.AddWithValue("@ThangNamThanhToan", dtp\_TNTToan\_HD.Text);

connection.Open();

try

{

command.ExecuteNonQuery();

MessageBox.Show("Thêm thành công", "Thông báo", MessageBoxButtons.OK);

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

LoaddataShow();

}

}

}

}

}

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

{

string str = @"UPDATE HoaDon SET MaHoaDon=@MaHoaDon, @MaHopDong,@MaDienNuoc, @TongTienDien, @TongTienNuoc, @ThangNamTToan where MaHoaDon=@MaHoaDon";

using (SqlConnection connection = new SqlConnection(ChuoiKN))

{

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

{

command.Parameters.AddWithValue("@MaHoaDon", txtMaHD\_HD.Text);

command.Parameters.AddWithValue("@MaHopDong", txtTTN\_HD.Text);

command.Parameters.AddWithValue("@TongTienDien", txtTTDien\_HD.Text);

command.Parameters.AddWithValue("@TongTienNuoc", txtTTN\_HD.Text);

command.Parameters.AddWithValue("@ThangNamThanhToan", dtp\_TNTToan\_HD.Text);

connection.Open();

try

{

command.ExecuteNonQuery();

}

catch

{

MessageBox.Show("Lỗi", "Thông báo", MessageBoxButtons.OK);

}

LoaddataShow();

}

}

}

private void dgDS\_HoaDon\_Click(object sender, EventArgs e)

{

txtMaHD\_HD.Text = dgDS\_HoaDon.SelectedRows[0].Cells[0].Value.ToString();

txtMaHopDong\_HD.Text = dgDS\_HoaDon.SelectedRows[0].Cells[1].Value.ToString();

txtMaDNuoc\_HD.Text = dgDS\_HoaDon.SelectedRows[0].Cells[2].Value.ToString();

txtTTDien\_HD.Text = dgDS\_HoaDon.SelectedRows[0].Cells[3].Value.ToString();

txtTTN\_HD.Text = dgDS\_HoaDon.SelectedRows[0].Cells[4].Value.ToString();

}

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

{

foreach (DataGridViewRow i in dgDS\_HoaDon.SelectedRows)

{

if (!i.IsNewRow)

{

string str\_MaHoaDon = dgDS\_HoaDon.SelectedRows[0].Cells["MaHoaDon"].Value.ToString();

string str = @"delete from HoaDon where MaHoaDon= @MaHoaDon";

using (SqlCommand command = new SqlCommand(str, conn))

{

command.Parameters.AddWithValue("@MaHoaDon", txtMaHD\_HD.Text);

try

{

command.ExecuteNonQuery();

}

catch

{

MessageBox.Show("Xóa lỗi", "Thông báo", MessageBoxButtons.OK);

}

}

dgDS\_HoaDon.Rows.RemoveAt(dgDS\_HoaDon.SelectedRows[0].Index);

}

else

{

MessageBox.Show("Chưa chọn dữ liệu để xóa", "Thông báo", MessageBoxButtons.OK);

}

}

}

private void btnHuy\_HD\_Click(object sender, EventArgs e)

{

txtMaDNuoc\_HD.Clear();

txtMaHD\_HD.Clear();

txtMaHopDong\_HD.Clear();

txtTTN\_HD.Clear();

txtTKiemHD\_HD.Clear();

txtTTDien\_HD.Clear();

}

private void btFind\_HDon\_Click(object sender, EventArgs e)

{

try

{

string str\_Tim = @"select \* from HoaDon where MaHoaDon like '" + txtTKiemHD\_HD.Text + "%'";

SqlDataAdapter dataAdapter = new SqlDataAdapter(str\_Tim, conn);

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

dgDS\_HoaDon.DataSource = dataTable;

}

catch { MessageBox.Show("Hãy nhập mã số nhân viên cần tìm", "Thông báo", MessageBoxButtons.OK); }

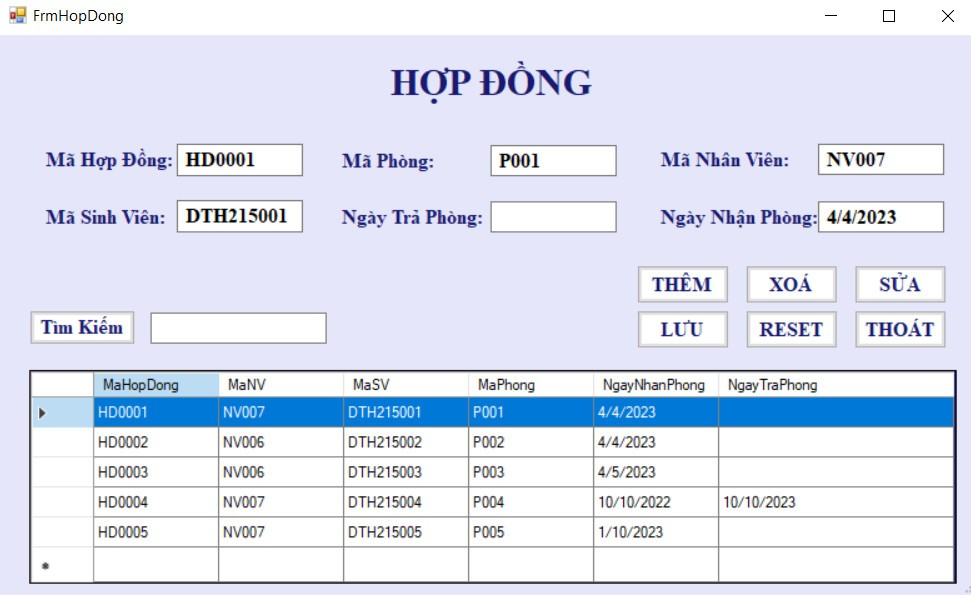
}

}

}

1. **Form Hợp đồng**

**– Giao diện:**



**– Code:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Do\_an

{

public partial class FrmHopDong : Form

{

string ChuoiKN = "Data Source=LAPTOP-1170Q5GN\\MSSQLSERVER01;Initial Catalog=QLKTX;Integrated Security=True";

SqlConnection conn;

public FrmHopDong()

{

InitializeComponent();

}

private void OpenSQL()

{

conn = new SqlConnection(ChuoiKN);

conn.Open();

}

private void LoaddataShow()

{

string str\_Open = @"select \* from HopDong";

SqlDataAdapter adapter = new SqlDataAdapter(str\_Open, conn);

DataTable dt = new DataTable();

adapter.Fill(dt);

dgDS\_HD.DataSource = dt;

}

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

{

OpenSQL();

LoaddataShow();

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void dateTimePicker2\_ValueChanged(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

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

{

FrmTrangChu f = new FrmTrangChu();

f.Show();

this.Close();

}

private void btnHuy\_HDong\_Click(object sender, EventArgs e)

{

txtMaHopDong\_HDong.Clear();

txtMaNV\_HDong.Clear();

txtMaPhong\_HDong.Clear();

txtNgayNPhong\_HDong.Clear();

txtNgayTPhong\_HDong.Clear();

txtSV\_HDong.Clear();

txtTKiem\_HDong.Clear();

}

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

{

foreach (DataGridViewRow i in dgDS\_HD.SelectedRows)

{

if (!i.IsNewRow)

{

string str\_MaHopDong = dgDS\_HD.SelectedRows[0].Cells["MaHopDong"].Value.ToString();

string str = @"delete from HopDong where MaHopDong= @MaHopDong";

using (SqlCommand command = new SqlCommand(str, conn))

{

command.Parameters.AddWithValue("@MaHopDong", str\_MaHopDong);

try

{

command.ExecuteNonQuery();

}

catch

{

MessageBox.Show("Xóa lỗi", "Thông báo", MessageBoxButtons.OK);

}

}

dgDS\_HD.Rows.RemoveAt(dgDS\_HD.SelectedRows[0].Index);

}

else

{

MessageBox.Show("Chưa chọn dữ liệu để xóa", "Thông báo", MessageBoxButtons.OK);

}

}

}

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

{

string str = @"UPDATE HopDong SET MaHopDong=@MaHopDong, MaSV=@MaSV, MaNV=@MaNV, NgayNhanPhong=@NgayNhanPhong, NgayTraPhong=@NgayTraPhong, MaPhong=@MaPhong where MaHopDong=@MaHopDong";

using (SqlConnection connection = new SqlConnection(ChuoiKN))

{

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

{

command.Parameters.AddWithValue("@MaHopDong", txtMaHopDong\_HDong.Text);

command.Parameters.AddWithValue("@MaNV", txtMaNV\_HDong.Text);

command.Parameters.AddWithValue("@MaSV", txtSV\_HDong.Text);

command.Parameters.AddWithValue("@NgayNhanPhong", txtNgayNPhong\_HDong.Text);

command.Parameters.AddWithValue("@NgayTraPhong", txtNgayTPhong\_HDong.Text);

command.Parameters.AddWithValue("@MaPhong", txtMaPhong\_HDong.Text);

connection.Open();

try

{

command.ExecuteNonQuery();

MessageBox.Show("Sửa thành công", "Thông báo", MessageBoxButtons.OK);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

LoaddataShow();

}

}

}

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

{

string str = @"insert into HopDong values (@MaHopDong, @MaNV, @MaSV, @MaPhong, @NgayNhanPhong, @NgayTraPhong)";

if (txtMaHopDong\_HDong.Text == "" || txtMaNV\_HDong.Text == "" || txtSV\_HDong.Text == "" ||

txtNgayNPhong\_HDong.Text == "" || txtNgayTPhong\_HDong.Text == "" || txtMaPhong\_HDong.Text == "")

{

MessageBox.Show("Không được bỏ trống", "Thông báo", MessageBoxButtons.OK);

}

else

{

using (SqlConnection connection = new SqlConnection(ChuoiKN))

{

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

{

command.Parameters.AddWithValue("@MaHopDong", txtMaHopDong\_HDong.Text);

command.Parameters.AddWithValue("@MaNV", txtMaNV\_HDong.Text);

command.Parameters.AddWithValue("@MaSV", txtSV\_HDong.Text);

command.Parameters.AddWithValue("@NgayNhanPhong", txtNgayNPhong\_HDong.Text);

command.Parameters.AddWithValue("@NgayTraPhong", txtNgayTPhong\_HDong.Text);

command.Parameters.AddWithValue("@MaPhong", txtMaPhong\_HDong.Text);

connection.Open();

try

{

command.ExecuteNonQuery();

MessageBox.Show("Thêm thành công", "Thông báo", MessageBoxButtons.OK);

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

LoaddataShow();

}

}

}

}

private void btFind\_HDong\_Click(object sender, EventArgs e)

{

try

{

string str\_Tim = @"select \* from HopDong where MaHopDong like '" + txtMaHopDong\_HDong.Text + "%'";

SqlDataAdapter dataAdapter = new SqlDataAdapter(str\_Tim, conn);

DataTable dataTable = new DataTable();

dataAdapter.Fill(dataTable);

dgDS\_HD.DataSource = dataTable;

}

catch { MessageBox.Show("Hãy nhập mã số nhân viên cần tìm", "Thông báo", MessageBoxButtons.OK); }

}

private void dgDS\_HD\_Click(object sender, EventArgs e)

{

txtMaHopDong\_HDong.Text = dgDS\_HD.SelectedRows[0].Cells[0].Value.ToString();

txtMaNV\_HDong.Text = dgDS\_HD.SelectedRows[0].Cells[1].Value.ToString();

txtSV\_HDong.Text = dgDS\_HD.SelectedRows[0].Cells[2].Value.ToString();

txtMaPhong\_HDong.Text = dgDS\_HD.SelectedRows[0].Cells[3].Value.ToString();

txtNgayNPhong\_HDong.Text = dgDS\_HD.SelectedRows[0].Cells[4].Value.ToString();

txtNgayTPhong\_HDong.Text = dgDS\_HD.SelectedRows[0].Cells[5].Value.ToString();

}

}

}