|  |
| --- |
| **TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**  **KHOA ĐIỆN TỬ**  **BỘ MÔN KỸ THUẬT MÁY TÍNH**  C:\Users\Windows 10 TIMT\Desktop\index.jpg  **BÀI TẬP LỚN**  **LẬP TRÌNH TRONG MÔI TRƯỜNG WINDOW**  Sinh viên : Nguyễn Tá Anh  Lớp : K54 KMT01  Mssv: K185520103229  Giảng viên hướng dẫn : Đỗ Duy Cốp  **Thái Nguyên –2023** |

|  |  |
| --- | --- |
| **TRƯỜNG ĐHKTCN** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **KHOA ĐIỆN TỬ** | ***Độc lập - Tự do - Hạnh phúc*** |

**BÀI TẬP LỚN**

**LẬP TRÌNH TRONG MÔI TRƯỜNG WINDOW**

Sinh viên: Nguyễn Tá Anh

MSSV: K185520103229

Lớp: K54KMT Ngành:Tin học công nghiệp

Giáo viên: Đỗ Duy Cốp

Tên đề tài: Xây dựng App quản lý ký túc xá.

|  |
| --- |
| **GIÁO VIÊN HƯỚNG DẪN** |
| *(Ký và ghi rõ họ tên)* |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

### Thái Nguyên, ngày….tháng…..năm 20....

## GIÁO VIÊN HƯỚNG DẪN

*(Ký ghi rõ họ tên)*

I.Giới thiệu đề bài

**Project 1: Xây dựng app quản lý Ký túc xá** **sv**

1. Xây dựng csdl phù hợp cho bài toán.

2. Có chức năng hiển thị toàn bộ danh sách sinh viên, danh sách phòng.

3. Có chức năng tìm kiếm thông minh (ví dụ nhập Nguyễn Huệ -> tìm thấy Trần Nguyễn Thị Huệ Chi).

4. Có chức năng thêm mới sinh viên vào bảng dữ liệu sinh viên.

5. Có chức năng sửa dữ liệu.

6. Có chức năng xóa dữ liệu.

7. Có chức năng in ấn danh sách đang hiển thị ra máy in. (thêm vào header logo tnut)

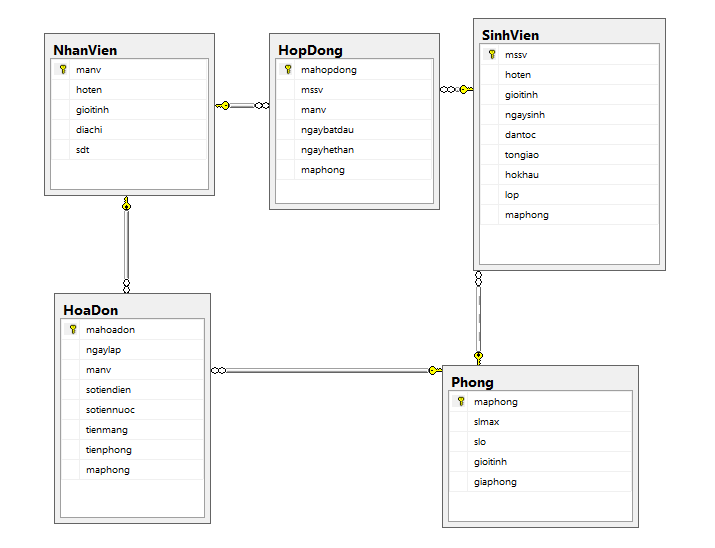
8. Có chức năng tìm kiếm phòng phù hợp cho sinh viên mới muốn vào ở ktx.

9. Có chứa năng in ra hợp đồng vào ở ký túc xá.

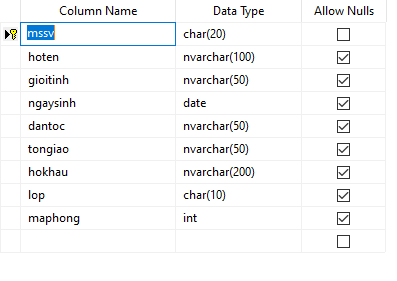
10. Có chức năng thống kê danh sách sinh viên đang ở theo từng phòng

II. Kết quả thiết kế CSDL phù hợp với đề tài.

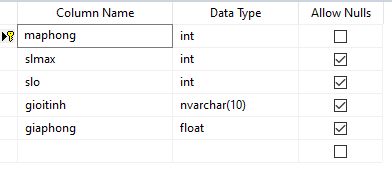
1. Bảng liên kết Diagram



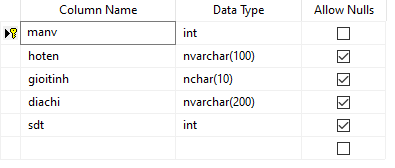
2. Bảng sinh viên



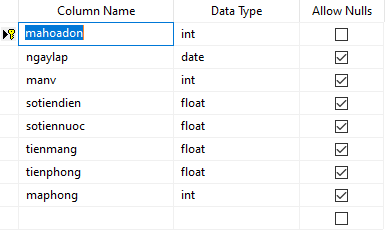
3. Bảng phòng



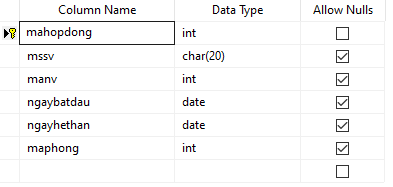
4.Bảng nhân viên



5. Bảng hóa đơn



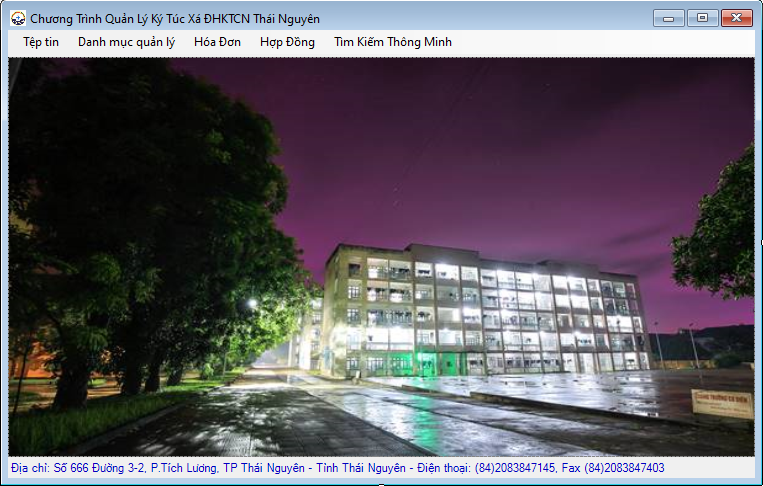
6. Bảng hợp đồng



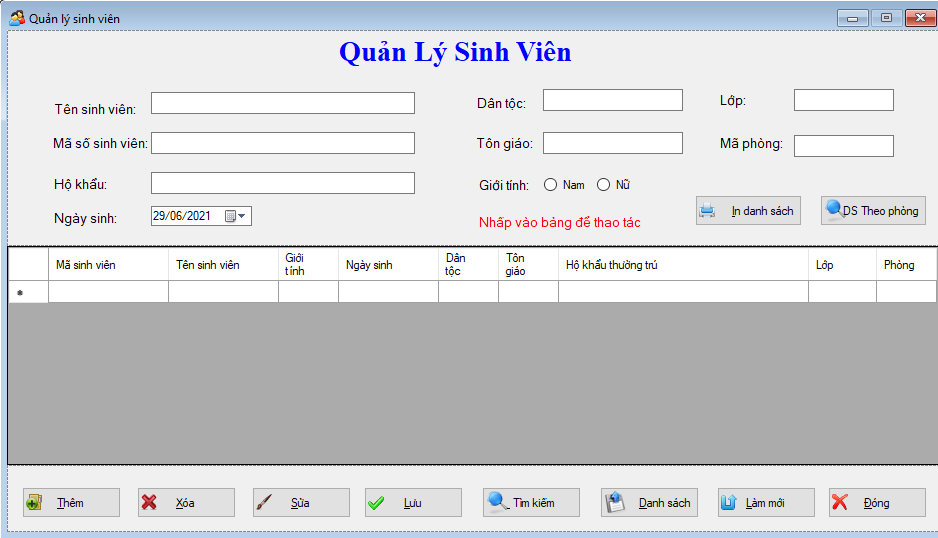
**III. Thiết kế giao diện, các hàm phục vụ cho các chức năng của chương trình.**

1. Các chức năng trong chương trình.

1.1 Menu



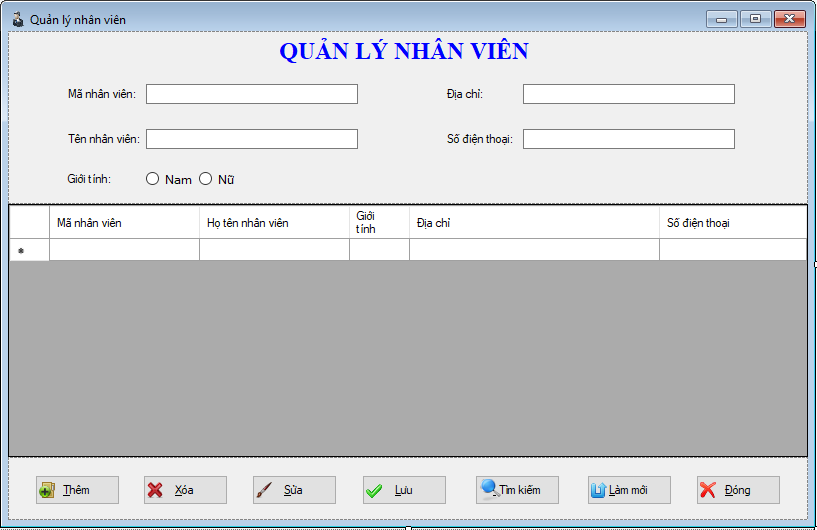
1.2 Quản lý sinh viên



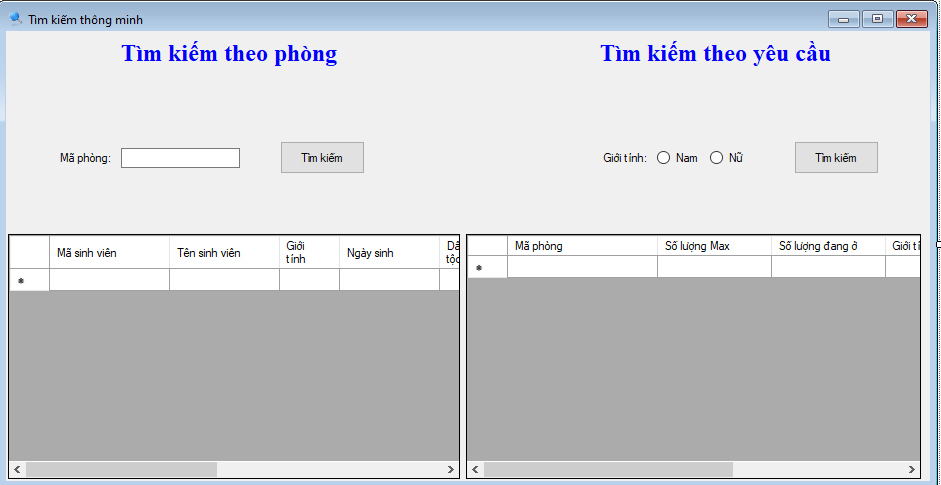
1.3 Quản lý phòng



1.4 Quản lý nhân viên



1.5 Tìm kiếm thông minh



1.6 Quản lý hợp đồng



1.7 Quản lý hóa đơn



2. Có chức năng hiển thị toàn bộ danh sách sinh viên, danh sách phòng.



3. Có chức năng tìm kiếm thông minh (ví dụ nhập Nguyễn Huệ -> tìm thấy Trần Nguyễn Thị Huệ Chi).



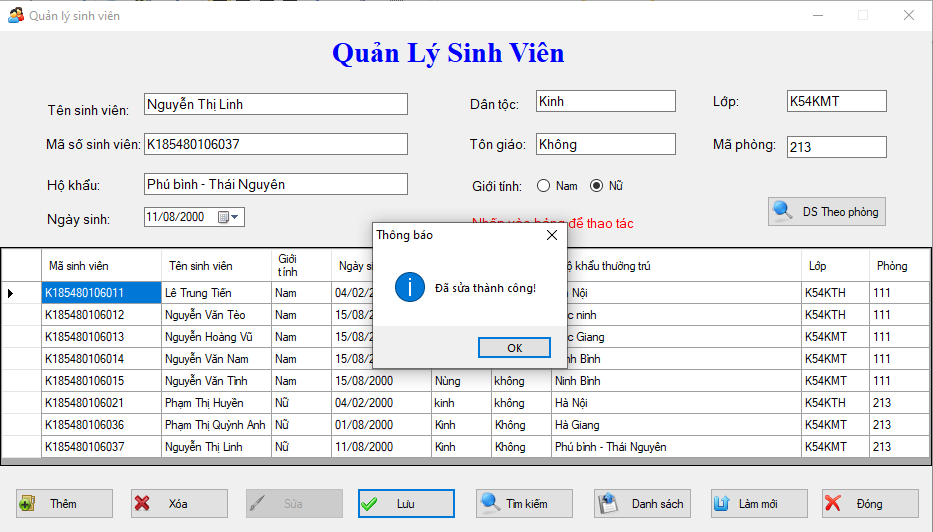
- Người dùng nhập trực tiếp vào textbox và tìm kiếm theo những textbox đó.



4. Có chức năng thêm mới sinh viên vào bảng dữ liệu sinh viên.



5. Có chức năng sửa dữ liệu.



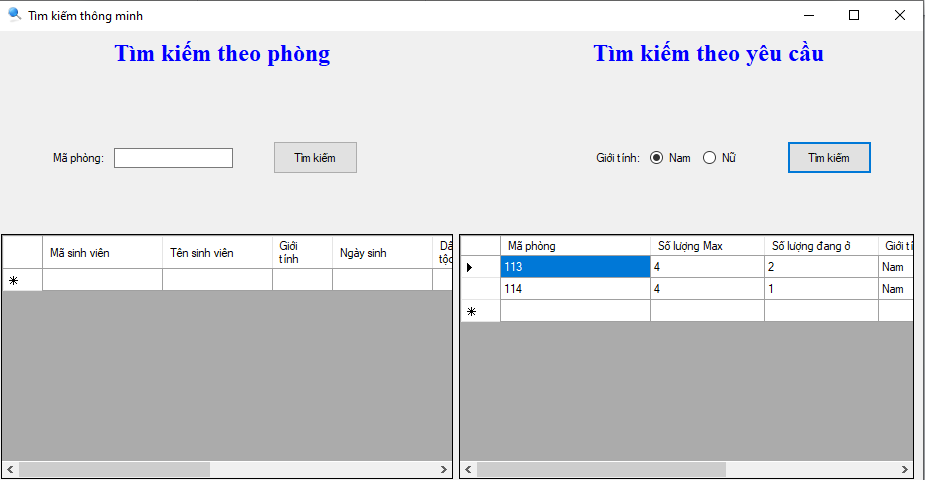
6. Có chức năng xóa dữ liệu.



7. Có chức năng in ấn danh sách đang hiển thị ra máy in. (thêm vào header logo tnut)



8. Có chức năng tìm kiếm phòng phù hợp cho sinh viên mới muốn vào ở ktx.



9. Có chứa năng in ra hợp đồng vào ở ký túc xá.

10. Có chức năng thống kê danh sách sinh viên đang ở theo từng phòng.

**IV.** **Mã nguồn của chương trình.**

1.1 Mã nguồn Function

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms; // khai báo thư viện để sử dụng MessageBox

namespace AppQuanLyBanHang.Class

{

class Function

{

// khai báo biến kết nối sql

public static SqlConnection con;

// tạo phương thức kết nối

public static void Connnect()

{

string sqlcon = @"Data Source=DESKTOP-7KJ5F4U\SQLEXPRESS;Initial Catalog=QuanLyKTX;Integrated Security=True";

con = new SqlConnection(sqlcon);

if (con.State != ConnectionState.Open)

{

con.Open();

}

else MessageBox.Show("Kết nối DataBase không thành công!");

}

// Phương thức ngắt kết nối

public static void Disconnect()

{

if(con.State == ConnectionState.Open)

{

con.Close();

con.Dispose(); // giải phóng tài nguyên

con = null;

}

}

// Phương thức thực thi câu lệnh lấy dữ liệu

public static DataTable GetDataToTable(String sql)

{

DataTable table = new DataTable();

SqlDataAdapter dap = new SqlDataAdapter(sql, con);

dap.Fill(table);

return table;

}

//Hàm thực hiện câu lệnh SQL

public static void RunSQL(string sql)

{

SqlCommand cmd; //Đối tượng thuộc lớp SqlCommand

cmd = new SqlCommand();

cmd.Connection = con; //Gán kết nối

cmd.CommandText = sql; //Gán lệnh SQL

try

{

cmd.ExecuteNonQuery(); //Thực hiện câu lệnh SQL

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

cmd.Dispose();//Giải phóng bộ nhớ

cmd = null;

}

//Hàm kiểm tra mã sinh viên đã tồn tại hay chưa

public static bool CheckKey(string sql)

{

SqlDataAdapter dap = new SqlDataAdapter(sql, con);

DataTable table = new DataTable();

dap.Fill(table);

if (table.Rows.Count > 0)

return true;

else return false;

}

}

}

1.2 Main

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using AppQuanLyBanHang.Class;

namespace AppQuanLyBanHang

{

public partial class frmMain : Form

{

public frmMain()

{

InitializeComponent();

}

//kết nối csdl khi load form main

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

{

Function.Connnect();

}

// Đóng kết nối csdl khi đóng form main

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

{

Function.Disconnect();

Application.Exit();

}

// Mở form quản lý sinh viên

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

{

frmQLSV qlsv = new frmQLSV();

qlsv.ShowDialog();

}

// Quản lý nhân viên

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

{

frmQLNV Nhanvien = new frmQLNV();

Nhanvien.ShowDialog();

}

// Quản lý phòng

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

{

frmQLPhong Phong = new frmQLPhong();

Phong.ShowDialog();

}

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

{

new frmHopDong().Show();

}

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

{

new frmSmart().Show();

}

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

{

new frmHoadon().Show();

}

}

}

1.3 Quản lý sinh viên

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient; //Sử dụng thư viện để làm việc với SQL server

using AppQuanLyBanHang.Class; //Sử dụng class Functions.cs

namespace AppQuanLyBanHang

{

public partial class frmQLSV : Form

{

DataTable tblQLSV; // Tạo bảng để hứng dữ liệu

String masv; // Lưu mã sinh viên khi Click vào DataGridView

public frmQLSV()

{

InitializeComponent();

}

// Hàm load dữ liệu lên bảng khi Load frm QLSV

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

{

LoadDataGridView();

btnXoa.Enabled = false;

btnSua.Enabled = false;

btnLuu.Enabled = false;

}

// Truyền dữ liệu từ tblQLSV sang datagridview

private void LoadDataGridView()

{

string sql = "SELECT \* FROM SinhVien";

tblQLSV = Function.GetDataToTable(sql); //Đọc dữ liệu từ bảng

dgvQLSV.DataSource = tblQLSV; //Nguồn dữ liệu

dgvQLSV.AllowUserToAddRows = false; //Không cho người dùng thêm dữ liệu trực tiếp

dgvQLSV.EditMode = DataGridViewEditMode.EditProgrammatically; //Không cho sửa dữ liệu trực tiếp

}

// xử lý sự kiện click vào dgv Băm nhỏ dữ liệu từ datagridview sang textbox

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

{

int i = dgvQLSV.CurrentRow.Index;

txtTenSV.Text = dgvQLSV.Rows[i].Cells[1].Value.ToString();

txtMaSV.Text = dgvQLSV.Rows[i].Cells[0].Value.ToString();

dtpSV.Value = (DateTime)dgvQLSV.Rows[i].Cells[3].Value;

txtHokhau.Text = dgvQLSV.Rows[i].Cells[6].Value.ToString();

txtDantoc.Text = dgvQLSV.Rows[i].Cells[4].Value.ToString();

txtTongiao.Text = dgvQLSV.Rows[i].Cells[5].Value.ToString();

txtLop.Text = dgvQLSV.Rows[i].Cells[7].Value.ToString();

txtphong.Text = dgvQLSV.Rows[i].Cells[8].Value.ToString();

// So sánh hai chuỗi không phân biệt chữ hoa chữ thường

if (String.Compare(dgvQLSV.Rows[i].Cells[2].Value.ToString(), "Nam", true) == 0)

{

rdonam.Checked = true;

}

else

{

rdonu.Checked = true;

}

// Chuyển mã sinh viên vào biến masv

masv = dgvQLSV.Rows[i].Cells[0].Value.ToString();

// Khi click vào một dòng thì Enable 2 nút xóa và sửa

btnSua.Enabled = true;

btnXoa.Enabled = true;

}

// Nút xóa sinh viên

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

{

String sql;

if(MessageBox.Show("Bạn có muốn xóa sinh viên này không?", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Information) == DialogResult.OK)

{

sql = "DELETE FROM SinhVien WHERE mssv = '" + txtMaSV.Text.Trim() + "'";

Function.RunSQL(sql);

LoadDataGridView();

ResetValue();

btnXoa.Enabled = false;

}

}

// Nút thêm mới một sinh viên

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

{

btnSua.Enabled = false;

btnXoa.Enabled = false;

btnLuu.Enabled = true;

txtTenSV.Focus();

}

// Hàm xóa dữ liệu trong txtbox

private void ResetValue()

{

txtMaSV.Text = "";

txtTenSV.Text = "";

txtHokhau.Text = "";

dtpSV.Value = DateTime.Now;

txtTongiao.Text = "";

txtDantoc.Text = "";

txtLop.Text = "";

txtphong.Text = "";

rdonam.Checked = false;

rdonu.Checked = false;

}

// Nút lưu sinh viên vào cơ sở dữ liệu

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

{

String sql;

String gt = "";

if (Check\_Value() == true) return;

sql = "SELECT mssv FROM SinhVien WHERE mssv = '" + txtMaSV.Text.Trim() + "'";

if (Function.CheckKey(sql))

{

MessageBox.Show("Mã sinh viên đã tồn tại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtMaSV.Focus();

return;

}

if (rdonam.Checked == true)

{

gt = "Nam";

}

else gt = "Nữ";

sql = "INSERT INTO SinhVien VALUES('" + txtMaSV.Text + "',N'" + txtTenSV.Text + "',N'" + gt

+ "','" + dtpSV.Value.ToString("yyyy/MM/dd") + "', N'" + txtDantoc.Text + "', N'" + txtTongiao.Text + "', N'"

+ txtHokhau.Text + "', N'" + txtLop.Text + "'," + txtphong.Text + ")";

Function.RunSQL(sql); // Chạy câu lệnh sql

LoadDataGridView(); // Làm mới dữ liệu

btnLuu.Enabled = false;

btnXoa.Enabled = false;

btnSua.Enabled = false;

}

// Sửa(UPDATE) dữ liệu

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

{

String sql;

String gt = "";

if (Check\_Value() == true) return;

// Check mã sinh viên

sql = "SELECT mssv FROM SinhVien WHERE mssv = '" + txtMaSV.Text.Trim() + "'";

if (String.Compare(txtMaSV.Text, masv, true) != 0 && Function.CheckKey(sql))

{

MessageBox.Show("Mã sinh viên đã tồn tại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtMaSV.Focus();

return;

}

// xách định giới tính

if (rdonam.Checked == true)

{

gt = "Nam";

}

else gt = "Nữ";

sql = "UPDATE SinhVien SET mssv = '" + txtMaSV.Text + "', hoten = N'" + txtTenSV.Text + "', gioitinh = N'" + gt + "', ngaysinh = N'"

+ dtpSV.Value.ToString("yyyy/MM/dd") + "', dantoc = N'" + txtDantoc.Text + "', tongiao = N'" + txtTongiao.Text + "', hokhau = N'"

+ txtHokhau.Text + "', lop = '" + txtLop.Text + "', maphong = " + txtphong.Text + " WHERE mssv = '" + masv + "'";

Function.RunSQL(sql);

LoadDataGridView();

btnSua.Enabled = false;

MessageBox.Show("Đã sửa thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

// Đóng form

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

{

this.Close();

}

// Nút làm mới

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

{

ResetValue();

btnXoa.Enabled = false;

btnLuu.Enabled = false;

btnSua.Enabled = false;

}

// Nút load danh sách

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

{

LoadDataGridView();

}

// Check xem đã điền đầy đủ giá trị trong txt hay chưa

private bool Check\_Value()

{

if (txtTenSV.Text.Trim().Length == 0) // Nếu chưa nhập gì trong txt

{

MessageBox.Show("Bạn chưa nhập tên sinh viên", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtTenSV.Focus();

return true;

}

if (txtMaSV.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập mã sinh viên", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaSV.Focus();

return true;

}

if (txtHokhau.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập hộ khẩu", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtHokhau.Focus();

return true;

}

if (txtDantoc.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập dân tộc", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaSV.Focus();

return true;

}

if (txtTongiao.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập tôn giáo", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtTongiao.Focus();

return true;

}

if (txtLop.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập lớp sinh viên", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtLop.Focus();

return true;

}

if (txtphong.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập phòng sinh viên", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtphong.Focus();

return true;

}

if (rdonam.Checked == false && rdonu.Checked == false)

{

MessageBox.Show("Bạn chưa chọn giới tính", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

return true;

}

return false;

}

// Nút tìm kiếm

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

{

string sql;

sql = "SELECT \* from SinhVien WHERE 1=1";

if (txtMaSV.Text != "")

sql += " AND mssv LIKE N'%" + txtMaSV.Text + "%'";

if (txtTenSV.Text != "")

sql += " AND hoten LIKE N'%" + txtTenSV.Text + "%'";

if (txtHokhau.Text != "")

sql += " AND hokhau LIKE N'%" + txtHokhau + "%'";

tblQLSV = Function.GetDataToTable(sql);

if (tblQLSV.Rows.Count == 0)

MessageBox.Show("Không có bản ghi thoả mãn điều kiện tìm kiếm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

else MessageBox.Show("Có " + tblQLSV.Rows.Count + " bản ghi thoả mãn điều kiện!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

dgvQLSV.DataSource = tblQLSV;

ResetValue();

}

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

{

new frmSmart().Show();

}

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

{

new Form1().Show();

}

}

}

1.4 Quản lý phòng

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

using AppQuanLyBanHang.Class;

namespace AppQuanLyBanHang

{

public partial class frmQLPhong : Form

{

DataTable tblPhong;

String mphong;

public frmQLPhong()

{

InitializeComponent();

}

// Những thao tác khi load form

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

{

LoadDataGridView();

btnXoa.Enabled = false;

btnSua.Enabled = false;

btnLuu.Enabled = false;

}

// Load data lên DataGridView

private void LoadDataGridView()

{

string sql = "SELECT \* FROM Phong";

tblPhong = Function.GetDataToTable(sql); // Hứng dữ liệu từ bảng trong Function

dgvPhong.DataSource = tblPhong; //Nguồn dữ liệu

dgvPhong.AllowUserToAddRows = false; //Không cho người dùng thêm dữ liệu trực tiếp

dgvPhong.EditMode = DataGridViewEditMode.EditProgrammatically; //Không cho sửa dữ liệu trực tiếp

}

// Băm nhỏ dữ liệu

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

{

int i = dgvPhong.CurrentRow.Index;

txtMaPhong.Text = dgvPhong.Rows[i].Cells[0].Value.ToString();

txtSoLuong.Text = dgvPhong.Rows[i].Cells[1].Value.ToString();

txtDango.Text = dgvPhong.Rows[i].Cells[2].Value.ToString();

txtGiaphong.Text = dgvPhong.Rows[i].Cells[4].Value.ToString();

// Chuyển mã sinh viên vào biến masv

mphong = dgvPhong.Rows[i].Cells[0].Value.ToString();

// Khi click vào một dòng thì Enable 2 nút xóa và sửa

btnSua.Enabled = true;

btnXoa.Enabled = true;

}

// Nút thêm

private void btnThem\_Click\_1(object sender, EventArgs e)

{

btnSua.Enabled = false;

btnXoa.Enabled = false;

btnLuu.Enabled = true;

txtMaPhong.Focus();

}

// Nút xóa

private void btnXoa\_Click\_1(object sender, EventArgs e)

{

String sql;

if(MessageBox.Show("Bạn có muốn xóa phòng này không?", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Information) == DialogResult.OK)

{

sql = "DELETE FROM Phong WHERE maphong = " + txtMaPhong.Text.Trim();

Function.RunSQL(sql);

LoadDataGridView();

ResetValue();

btnXoa.Enabled = false;

}

}

// Nút sửa (UPDATE)

private void btnSua\_Click\_1(object sender, EventArgs e)

{

String sql;

if (Check\_Value() == true) return;

// Check mã phòng

sql = "SELECT maphong FROM Phong WHERE maphong = '" + txtMaPhong.Text.Trim() + "'";

if (String.Compare(txtMaPhong.Text, mphong, true) != 0 && Function.CheckKey(sql))

{

MessageBox.Show("Mã phòng đã tồn tại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtMaPhong.Focus();

return;

}

sql = "UPDATE Phong SET maphong = " + txtMaPhong.Text + " , slmax = " + txtSoLuong.Text + ", slo = " + txtDango.Text + ", giaphong = " + txtGiaphong.Text

+ "WHERE maphong =" + mphong;

Function.RunSQL(sql);

LoadDataGridView();

btnSua.Enabled = false;

MessageBox.Show("Đã sửa thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

// Nút lưu

private void btnLuu\_Click\_1(object sender, EventArgs e)

{

String sql;

if (Check\_Value() == true) return;

sql = "SELECT maphong FROM Phong WHERE maphong = '" + txtMaPhong.Text.Trim() + "'";

if (Function.CheckKey(sql))

{

MessageBox.Show("Mã phòng đã tồn tại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtMaPhong.Focus();

return;

}

sql = "INSERT INTO Phong(maphong, slmax, slo, giaphong) VALUES(" + txtMaPhong.Text + "," + txtSoLuong.Text + "," + txtDango.Text + "," + txtGiaphong.Text + ")";

Function.RunSQL(sql); // Chạy câu lệnh sql

LoadDataGridView(); // Làm mới dữ liệu

btnLuu.Enabled = false;

btnXoa.Enabled = false;

btnSua.Enabled = false;

}

// Nút làm mới

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

{

ResetValue();

btnXoa.Enabled = false;

btnSua.Enabled = false;

btnLuu.Enabled = false;

}

// Nút đóng form

private void btnDong\_Click\_1(object sender, EventArgs e)

{

this.Close();

}

// Hàm ResetValue

private void ResetValue()

{

txtMaPhong.Text = "";

txtSoLuong.Text = "";

txtDango.Text = "";

txtGiaphong.Text = "";

btnXoa.Enabled = false;

btnSua.Enabled = false;

btnLuu.Enabled = false;

}

// Check xem đã điền đầy đủ giá trị trong txt hay chưa

private bool Check\_Value()

{

if (txtMaPhong.Text.Trim().Length == 0) // Nếu chưa nhập gì trong txt

{

MessageBox.Show("Bạn chưa nhập mã phòng", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaPhong.Focus();

return true;

}

if (txtSoLuong.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập số lượng max", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtSoLuong.Focus();

return true;

}

if (txtDango.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập số lượng đang ở", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtDango.Focus();

return true;

}

if (txtGiaphong.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập giá phòng", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtGiaphong.Focus();

return true;

}

return false;

}

// Nút tìm kiếm

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

{

string sql;

if ((txtGiaphong.Text == "") && (txtMaPhong.Text == "") && (txtSoLuong.Text == "") && (txtDango.Text == ""))

{

MessageBox.Show("Bạn hãy nhập điều kiện tìm kiếm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

sql = "SELECT \* from Phong WHERE 1=1";

if (txtMaPhong.Text != "")

sql += " AND maphong LIKE '%" + txtMaPhong.Text + "%'";

if (txtSoLuong.Text != "")

sql += " AND slmax LIKE '%" + txtSoLuong.Text + "%'";

if (txtGiaphong.Text != "")

sql += " AND giaphong LIKE '%" + txtGiaphong + "%'";

tblPhong = Function.GetDataToTable(sql);

if (tblPhong.Rows.Count == 0)

MessageBox.Show("Không có bản ghi thoả mãn điều kiện tìm kiếm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

else MessageBox.Show("Có " + tblPhong.Rows.Count + " bản ghi thoả mãn điều kiện!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

dgvPhong.DataSource = tblPhong;

ResetValue();

}

}

}

1.5 Quản lý nhân viên

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using AppQuanLyBanHang.Class;

using System.Data.SqlClient;

namespace AppQuanLyBanHang

{

public partial class frmQLNV : Form

{

private DataTable tblNV;

string msnv;

public frmQLNV()

{

InitializeComponent();

}

// Những thao tác khi load form

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

{

LoadDataGridView();

btnXoa.Enabled = false;

btnSua.Enabled = false;

btnLuu.Enabled = false;

}

// Load data lên DataGridView

private void LoadDataGridView()

{

string sql = "SELECT \* FROM NhanVien";

tblNV = Function.GetDataToTable(sql); //Đọc dữ liệu từ bảng

dgvNhanvien.DataSource = tblNV; //Nguồn dữ liệu

dgvNhanvien.AllowUserToAddRows = false; //Không cho người dùng thêm dữ liệu trực tiếp

dgvNhanvien.EditMode = DataGridViewEditMode.EditProgrammatically; //Không cho sửa dữ liệu trực tiếp

}

// Băm nhỏ dữ liệu

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

{

int i = dgvNhanvien.CurrentRow.Index;

txtManhanvien.Text = dgvNhanvien.Rows[i].Cells[0].Value.ToString();

txtTen.Text = dgvNhanvien.Rows[i].Cells[1].Value.ToString();

txtDiachi.Text = dgvNhanvien.Rows[i].Cells[3].Value.ToString();

txtDienthoai.Text = dgvNhanvien.Rows[i].Cells[4].Value.ToString();

if (String.Compare(dgvNhanvien.Rows[i].Cells[2].Value.ToString(), "Nam", true) == 0)

{

rdoNam.Checked = true;

}

else

{

rdoNu.Checked = true;

}

// Chuyển mã sinh viên vào biến masv

msnv = dgvNhanvien.Rows[i].Cells[0].Value.ToString();

// Khi click vào một dòng thì Enable 2 nút xóa và sửa

btnSua.Enabled = true;

btnXoa.Enabled = true;

}

// Nút thêm

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

{

btnSua.Enabled = false;

btnXoa.Enabled = false;

btnLuu.Enabled = true;

txtManhanvien.Focus();

}

// Nút xóa

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

{

String sql;

if(MessageBox.Show("Bạn có muốn xóa nhân viên này không?", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Information) == DialogResult.OK)

{

sql = "DELETE FROM NhanVien WHERE manv = " + txtManhanvien.Text.Trim();

Function.RunSQL(sql);

LoadDataGridView();

ResetValue();

btnXoa.Enabled = false;

}

}

// Nút sửa

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

{

String sql;

String gt = "";

if (Check\_Value() == true) return;

// Check mã nhân viên

sql = "SELECT manv FROM NhanVien WHERE manv = '" + txtManhanvien.Text.Trim() + "'";

if (String.Compare(txtManhanvien.Text, msnv, true) != 0 && Function.CheckKey(sql))

{

MessageBox.Show("Mã sinh viên đã tồn tại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtManhanvien.Focus();

return;

}

// xách định giới tính

if (rdoNam.Checked == true)

{

gt = "Nam";

}

else gt = "Nữ";

sql = "UPDATE NhanVien SET manv = " + txtManhanvien.Text + " , hoten = N'" + txtTen.Text + "', gioitinh = N'" + gt

+ "', diachi = N'" + txtDiachi.Text + "', sdt = " + txtDienthoai.Text + " WHERE manv = " + msnv;

Function.RunSQL(sql);

LoadDataGridView();

btnSua.Enabled = false;

MessageBox.Show("Đã sửa thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

// Nút lưu

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

{

String sql;

String gt = "";

if (Check\_Value() == true) return;

sql = "SELECT manv FROM NhanVien WHERE manv = '" + txtManhanvien.Text.Trim() + "'";

if (Function.CheckKey(sql))

{

MessageBox.Show("Mã nhân viên đã tồn tại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtManhanvien.Focus();

return;

}

if (rdoNam.Checked == true)

{

gt = "Nam";

}

else gt = "Nữ";

sql = "INSERT INTO NhanVien VALUES(" + txtManhanvien.Text + ",N'" + txtTen.Text + "',N'" + gt

+ "', N'" + txtDiachi.Text + "'," + txtDienthoai.Text + ")";

Function.RunSQL(sql); // Chạy câu lệnh sql

LoadDataGridView(); // Làm mới dữ liệu

btnLuu.Enabled = false;

btnXoa.Enabled = false;

btnSua.Enabled = false;

}

// Nút làm mới

private void btnLammoi\_Click\_1(object sender, EventArgs e)

{

ResetValue();

btnXoa.Enabled = false;

btnSua.Enabled = false;

btnLuu.Enabled = false;

}

// Nút đóng form

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

{

this.Close();

}

// Hàm ResetValue

private void ResetValue()

{

txtDiachi.Text = "";

txtDienthoai.Text = "";

txtManhanvien.Text = "";

txtTen.Text = "";

btnXoa.Enabled = false;

btnSua.Enabled = false;

btnLuu.Enabled = false;

rdoNam.Checked = false;

rdoNu.Checked = false;

}

// Check xem đã điền đầy đủ giá trị trong txt hay chưa

private bool Check\_Value()

{

if (txtManhanvien.Text.Trim().Length == 0) // Nếu chưa nhập gì trong txt

{

MessageBox.Show("Bạn chưa nhập mã sinh viên", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtManhanvien.Focus();

return true;

}

if (txtTen.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập tên nhân viên", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtTen.Focus();

return true;

}

if (txtDiachi.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập địa chỉ", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtDiachi.Focus();

return true;

}

if (txtDienthoai.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập điện thoại", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtDienthoai.Focus();

return true;

}

if (rdoNam.Checked == false && rdoNu.Checked == false)

{

MessageBox.Show("Bạn chưa chọn giới tính", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

return true;

}

return false;

}

// Nút xóa

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

{

string sql;

if ((txtDiachi.Text == "") && (txtDienthoai.Text == "") && (txtManhanvien.Text == "") && (txtTen.Text == "") && (rdoNam.Checked == false)

&& (rdoNu.Checked == false))

{

MessageBox.Show("Bạn hãy nhập điều kiện tìm kiếm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

sql = "SELECT \* from NhanVien WHERE 1=1";

if (txtTen.Text != "")

sql += " AND hoten LIKE N'%" + txtTen.Text + "%'";

if (txtManhanvien.Text != "")

sql += " AND manv LIKE '%" + txtManhanvien.Text + "%'";

if (txtDienthoai.Text != "")

sql += " AND sdt LIKE '%" + txtDienthoai.Text + "%'";

if (txtDiachi.Text != "")

sql += " AND diachi LIKE N'%" + txtDiachi.Text + "%'";

if(rdoNam.Checked == true)

sql += " AND gioitinh LIKE N'%Nam%'";

if (rdoNu.Checked == true)

sql += " AND gioitinh LIKE N'%Nữ%'";

tblNV = Function.GetDataToTable(sql);

if (tblNV.Rows.Count == 0)

MessageBox.Show("Không có bản ghi thoả mãn điều kiện tìm kiếm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

else

{

MessageBox.Show("Có " + tblNV.Rows.Count + " bản ghi thoả mãn điều kiện!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

dgvNhanvien.DataSource = tblNV;

}

}

}

}

1.6 Quản lý hóa đơn

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient; //Sử dụng thư viện để làm việc với SQL server

using AppQuanLyBanHang.Class; //Sử dụng class Functions.cs

namespace AppQuanLyBanHang

{

public partial class frmHoadon : Form

{

DataTable tblHoaDon;

public frmHoadon()

{

InitializeComponent();

}

// Load hóa đơn

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

{

LoadDataGridView();

btnXoa.Enabled = false;

btnLuu.Enabled = false;

btnIn.Enabled = false;

}

// Truyền dữ liệu từ tblQLSV sang datagridview

private void LoadDataGridView()

{

string sql = "SELECT \* FROM HoaDon";

tblHoaDon = Function.GetDataToTable(sql); //Đọc dữ liệu từ bảng

dgvhd.DataSource = tblHoaDon; //Nguồn dữ liệu

dgvhd.AllowUserToAddRows = false; //Không cho người dùng thêm dữ liệu trực tiếp

dgvhd.EditMode = DataGridViewEditMode.EditProgrammatically; //Không cho sửa dữ liệu trực tiếp

}

// xử lý sự kiện click vào dgv Băm nhỏ dữ liệu từ datagridview sang textbox

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

{

int i = dgvhd.CurrentRow.Index;

txtMaHoaDon.Text = dgvhd.Rows[i].Cells[0].Value.ToString();

txtMaphong.Text = dgvhd.Rows[i].Cells[7].Value.ToString();

txtManv.Text = dgvhd.Rows[i].Cells[2].Value.ToString();

txtTienDien.Text = dgvhd.Rows[i].Cells[3].Value.ToString();

txtTienNuoc.Text = dgvhd.Rows[i].Cells[4].Value.ToString();

txtTienMang.Text = dgvhd.Rows[i].Cells[5].Value.ToString();

txtTienPhong.Text = dgvhd.Rows[i].Cells[6].Value.ToString();

dtp1.Value = (DateTime)dgvhd.Rows[i].Cells[1].Value;

btnXoa.Enabled = true;

btnIn.Enabled = true;

}

// Hàm xóa dữ liệu trong txtbox

private void ResetValue()

{

txtMaHoaDon.Text = "";

txtManv.Text = "";

txtMaphong.Text = "";

dtp1.Value = DateTime.Now;

txtTienDien.Text = "";

txtTienMang.Text = "";

txtTienNuoc.Text = "";

txtTienPhong.Text = "";

}

// Check dữ liệu

private bool Check\_Value()

{

if (txtMaHoaDon.Text.Trim().Length == 0) // Nếu chưa nhập gì trong txt

{

MessageBox.Show("Bạn chưa nhập mã hóa đơn", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaHoaDon.Focus();

return true;

}

if (txtManv.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập mã nhân viên", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtManv.Focus();

return true;

}

if (txtMaphong.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập mã phòng", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaphong.Focus();

return true;

}

if (txtTienDien.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập tiền điện", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtTienDien.Focus();

return true;

}

if (txtTienMang.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập tiền mạng", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtTienMang.Focus();

return true;

}

if (txtTienNuoc.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập tiền nước", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtTienNuoc.Focus();

return true;

}

if (txtTienPhong.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập tiền phòng", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtTienPhong.Focus();

return true;

}

return false;

}

// Nút tạo hóa đơn

private void btnTaoHoaDon\_Click\_1(object sender, EventArgs e)

{

btnLuu.Enabled = true;

btnXoa.Enabled = false;

btnIn.Enabled = false;

txtMaHoaDon.Focus();

}

// Nút xóa hóa đơn

private void btnXoa\_Click\_1(object sender, EventArgs e)

{

String sql;

if (MessageBox.Show("Bạn có muốn xóa hóa đơn này không?", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Information) == DialogResult.OK)

{

sql = "DELETE FROM HoaDon WHERE mahoadon = '" + txtMaHoaDon.Text.Trim() + "'";

Function.RunSQL(sql);

LoadDataGridView();

ResetValue();

btnXoa.Enabled = false;

}

}

// Nút lưu hóa đơn

private void btnLuu\_Click\_1(object sender, EventArgs e)

{

String sql;

if (Check\_Value() == true) return;

sql = "SELECT mahoadon FROM HoaDon WHERE mahoadon = '" + txtMaHoaDon.Text.Trim() + "'";

if (Function.CheckKey(sql))

{

MessageBox.Show("Mã hóa đơn đã tồn tại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtMaHoaDon.Focus();

return;

}

sql = "INSERT INTO HoaDon VALUES(" + txtMaHoaDon.Text + ",'" + dtp1.Value.ToString("yyyy/MM/dd") + "',"

+ txtManv.Text + "," + txtTienDien.Text + "," + txtTienNuoc.Text + "," + txtTienMang.Text

+ "," + txtTienPhong.Text + "," + txtMaphong.Text + ")";

Function.RunSQL(sql); // Chạy câu lệnh sql

LoadDataGridView(); // Làm mới dữ liệu

btnLuu.Enabled = false;

btnXoa.Enabled = false;

}

// Nút làm mới

private void btnLammoi\_Click\_1(object sender, EventArgs e)

{

ResetValue();

btnXoa.Enabled = false;

btnLuu.Enabled = false;

btnIn.Enabled = false;

}

// Nút In

private void btnIn\_Click\_1(object sender, EventArgs e)

{

}

// Nút đóng

private void btnDong\_Click\_1(object sender, EventArgs e)

{

this.Close();

}

// Nút tìm kiếm

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

{

string sql;

sql = "SELECT \* from HoaDon WHERE 1=1";

if (txtMaHoaDon.Text != "")

sql += " AND mahoadon LIKE N'%" + txtMaHoaDon.Text + "%'";

if (txtMaphong.Text != "")

sql += " AND maphong LIKE N'%" + txtMaphong.Text + "%'";

if (txtManv.Text != "")

sql += " AND manv LIKE N'%" + txtManv + "%'";

tblHoaDon = Function.GetDataToTable(sql);

if (tblHoaDon.Rows.Count == 0)

MessageBox.Show("Không có bản ghi thoả mãn điều kiện tìm kiếm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

else MessageBox.Show("Có " + tblHoaDon.Rows.Count + " bản ghi thoả mãn điều kiện!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

dgvhd.DataSource = tblHoaDon;

ResetValue();

}

}

}

1.7 Quản lý hợp đồng

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient; //Sử dụng thư viện để làm việc với SQL server

using AppQuanLyBanHang.Class; //Sử dụng class Functions.cs

namespace AppQuanLyBanHang

{

public partial class frmHopDong : Form

{

DataTable tblHopDong;

public frmHopDong()

{

InitializeComponent();

}

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

{

LoadDataGridView();

btnXoa.Enabled = false;

btnIn.Enabled = false;

}

// Truyền dữ liệu từ tblQLSV sang datagridview

private void LoadDataGridView()

{

string sql = "SELECT \* FROM HopDong";

tblHopDong = Function.GetDataToTable(sql); //Đọc dữ liệu từ bảng

dgvhd.DataSource = tblHopDong; //Nguồn dữ liệu

dgvhd.AllowUserToAddRows = false; //Không cho người dùng thêm dữ liệu trực tiếp

dgvhd.EditMode = DataGridViewEditMode.EditProgrammatically; //Không cho sửa dữ liệu trực tiếp

}

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

{

int i = dgvhd.CurrentRow.Index;

txtMHD.Text = dgvhd.Rows[i].Cells[0].Value.ToString();

txtMaphong.Text = dgvhd.Rows[i].Cells[5].Value.ToString();

txtDmasv.Text = dgvhd.Rows[i].Cells[1].Value.ToString();

txtmanv.Text = dgvhd.Rows[i].Cells[2].Value.ToString();

dtpBd.Value = (DateTime)dgvhd.Rows[i].Cells[3].Value;

dtpBd.Value = (DateTime)dgvhd.Rows[i].Cells[4].Value;

btnXoa.Enabled = true;

btnIn.Enabled = true;

}

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

{

String sql;

if (MessageBox.Show("Bạn có muốn xóa hợp đồng này không?", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Information) == DialogResult.OK)

{

sql = "DELETE FROM HoaDon WHERE mahoadon = '" + txtMHD.Text.Trim() + "'";

Function.RunSQL(sql);

LoadDataGridView();

ResetValue();

btnXoa.Enabled = false;

}

}

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

{

String sql;

if (Check\_Value() == true) return;

// Check

sql = "SELECT mahopdong FROM HopDong WHERE mahopdong = '" + txtMHD.Text.Trim() + "'";

if (String.Compare(txtMHD.Text, mhd, true) != 0 && Function.CheckKey(sql))

{

MessageBox.Show("Mã hợp đã tồn tại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtMHD.Focus();

return;

}

sql = "UPDATE HopDong SET mahopdong = '" + txtMHD.Text + "', mssv = '" + txtDmasv.Text + "', manv = '" + txtmanv.Text

+ "', ngaybatdau = '" + dtpBd.Value.ToString("yyyy/MM/dd") + "', ngayhethan = '" + dtpkt.Value.ToString("yyyy/MM/dd")

+ "', maphong = " + txtMaphong.Text;

Function.RunSQL(sql);

LoadDataGridView();

btnSua.Enabled = false;

MessageBox.Show("Đã sửa thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information); }

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

{

String sql;

if (Check\_Value() == true) return;

sql = "SELECT mahopdong FROM HoaDon WHERE mahopdong = '" + txtMHD.Text.Trim() + "'";

if (Function.CheckKey(sql))

{

MessageBox.Show("Mã hợp đồng đã tồn tại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtMHD.Focus();

return;

}

sql = "INSERT INTO HopDong VALUES(" + txtMHD.Text + "," + txtDmasv.Text + "," + txtmanv.Text

+ ",'" + dtpBd.Value.ToString("yyyy/MM/dd") + "','" + dtpBd.Value.ToString("yyyy/MM/dd") + "',"

+ txtMaphong.Text + ")";

Function.RunSQL(sql); // Chạy câu lệnh sql

LoadDataGridView(); // Làm mới dữ liệu

btnXoa.Enabled = false;

}

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

{

string sql;

sql = "SELECT \* from HopDong WHERE 1=1";

if (txtMHD.Text != "")

sql += " AND mahopdong LIKE N'%" + txtMHD.Text + "%'";

if (txtMaphong.Text != "")

sql += " AND maphong LIKE N'%" + txtMaphong.Text + "%'";

if (txtDmasv.Text != "")

sql += " AND mssv LIKE N'%" + txtDmasv + "%'";

tblHopDong = Function.GetDataToTable(sql);

if (tblHopDong.Rows.Count == 0)

MessageBox.Show("Không có bản ghi thoả mãn điều kiện tìm kiếm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

else MessageBox.Show("Có " + tblHopDong.Rows.Count + " bản ghi thoả mãn điều kiện!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

dgvhd.DataSource = tblHopDong;

ResetValue();

}

// Nút in

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

{

}

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

{

ResetValue();

btnXoa.Enabled = false;

btnIn.Enabled = false;

}

private void ResetValue()

{

txtMHD.Text = "";

txtMaphong.Text = "";

txtmanv.Text = "";

dtpBd.Value = DateTime.Now;

dtpkt.Value = DateTime.Now;

txtDmasv.Text = "";

}

private bool Check\_Value()

{

if (txtMHD.Text.Trim().Length == 0) // Nếu chưa nhập gì trong txt

{

MessageBox.Show("Bạn chưa nhập mã hợp đồng", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMHD.Focus();

return true;

}

if (txtMaphong.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập mã phòng", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaphong.Focus();

return true;

}

if (txtmanv.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập mã nhân viên", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtmanv.Focus();

return true;

}

if (txtDmasv.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn chưa nhập mã sinh viên", "Thông Báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtDmasv.Focus();

return true;

}

return false;

}

}

}

1.8 Tìm kiếm thông minh

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient; //Sử dụng thư viện để làm việc với SQL server

using AppQuanLyBanHang.Class; //Sử dụng class Functions.cs

namespace AppQuanLyBanHang

{

public partial class frmSmart : Form

{

DataTable tblTKTM;

public frmSmart()

{

InitializeComponent();

}

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

{

string sql;

if(txtmaphong.Text == "")

{

MessageBox.Show("Bạn chưa nhập mã phòng!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtmaphong.Focus();

return;

} else

{

sql = "SELECT \* from SinhVien WHERE maphong = " + txtmaphong.Text;

}

tblTKTM = Function.GetDataToTable(sql);

if (tblTKTM.Rows.Count == 0)

MessageBox.Show("Không có bản ghi thoả mãn điều kiện tìm kiếm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

else MessageBox.Show("Có " + tblTKTM.Rows.Count + " bản ghi thoả mãn điều kiện!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

dgvQLSV.DataSource = tblTKTM;

}

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

{

string sql;

String gt;

if (rdoNam.Enabled == false && rdoNu.Enabled == false)

{

MessageBox.Show("Bạn chưa chọn giới tính!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

rdoNam.Focus();

return;

}

else

{

if (rdoNam.Checked == true)

{

gt = "Nam";

}

else gt = "Nữ";

sql = "SELECT \* from Phong WHERE gioitinh = N'" + gt + "'";

}

tblTKTM = Function.GetDataToTable(sql);

if (tblTKTM.Rows.Count == 0)

MessageBox.Show("Không có bản ghi thoả mãn điều kiện tìm kiếm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

else MessageBox.Show("Có " + tblTKTM.Rows.Count + " bản ghi thoả mãn điều kiện!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

dgvPhong.DataSource = tblTKTM;

}

}

}

1.9 From In sanh sách sinh viên

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using Microsoft.Reporting.WinForms;

namespace AppQuanLyBanHang

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

SqlConnection conn = new SqlConnection

(@"Data Source=DESKTOP-7KJ5F4U\SQLEXPRESS;Initial Catalog=QuanLyKTX;Integrated Security=True");

SqlDataAdapter dap = new SqlDataAdapter("select \* from SinhVien", conn);

DataSet ds = new DataSet();

dap.Fill(ds, "con");

this.reportViewer1.LocalReport.ReportEmbeddedResource =

"AppQuanLyKTX.Report1.rdlc";

ReportDataSource rds = new ReportDataSource();

rds.Name = "DataSet1";

rds.Value = ds.Tables[0];

reportViewer1.LocalReport.DataSources.Add(rds);

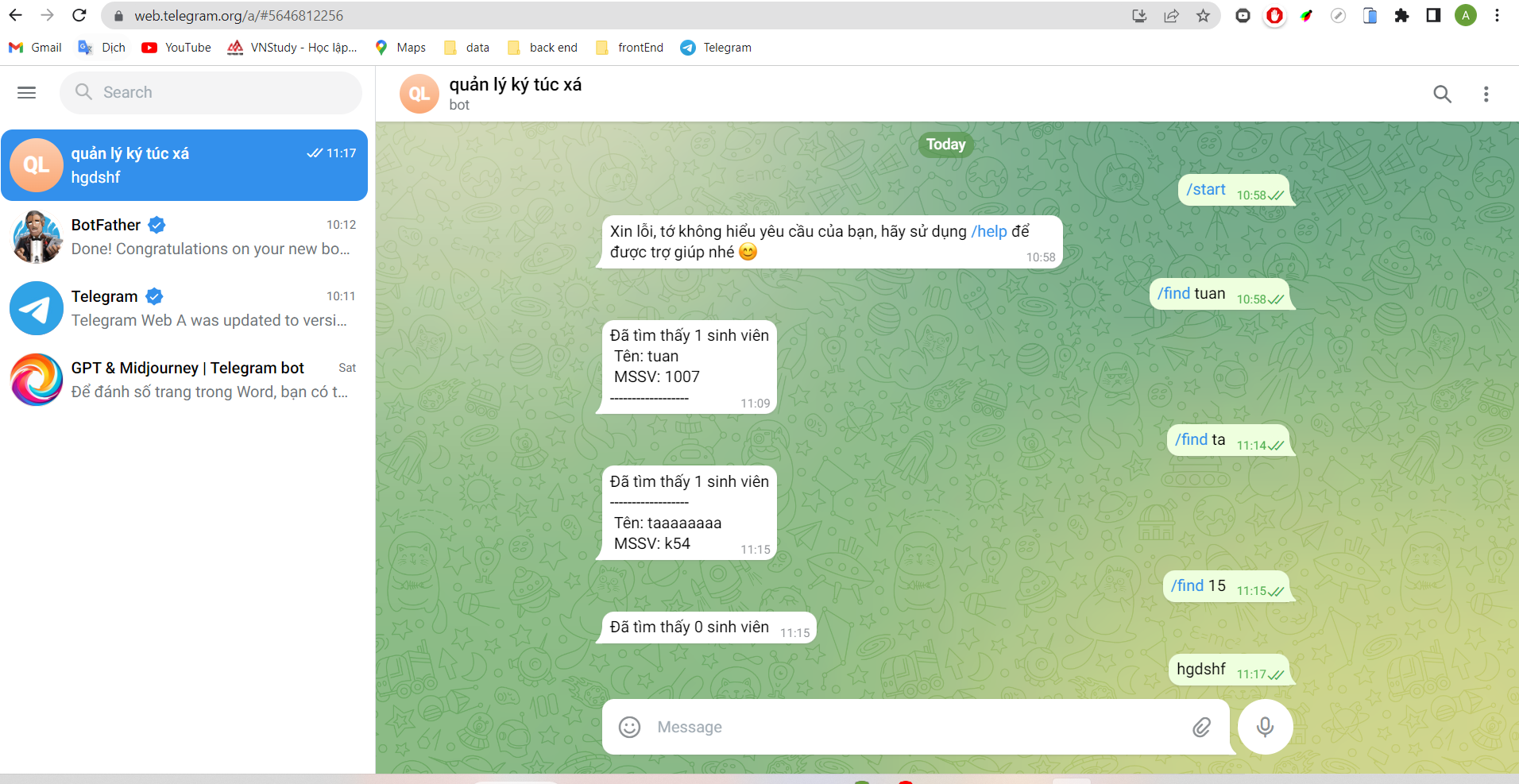
this.reportViewer1.RefreshReport();

}

}

}

2.0 : boot chat telegram



Code:

using CoffeeRestaurentTeleBot.Services;

using System;

using System.Data;

using System.IO;

using System.Security.Cryptography;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

using Telegram.Bot;

using Telegram.Bot.Exceptions;

using Telegram.Bot.Polling;

using Telegram.Bot.Types;

using Telegram.Bot.Types.Enums;

namespace TeleBot

{

public partial class TeleBot : Form

{

TelegramBotClient botClient;

int logCounter = 0;

void AddLog(string msg)

{

if (txtLog.InvokeRequired)

{

txtLog.BeginInvoke((MethodInvoker)delegate ()

{

AddLog(msg);

});

}

else

{

logCounter++;

if (logCounter > 100)

{

txtLog.Clear();

logCounter = 0;

}

txtLog.AppendText(msg + "\r\n");

}

}

public TeleBot()

{

InitializeComponent();

string token = "5646812256:AAEtg2-9puHUpJfvqjzB-qC65sIiuG5ZhyY";

//Console.WriteLine("my token=" + token);

botClient = new TelegramBotClient(token);

using CancellationTokenSource cts = new();

// StartReceiving does not block the caller thread. Receiving is done on the ThreadPool.

ReceiverOptions receiverOptions = new()

{

AllowedUpdates = Array.Empty<UpdateType>() // receive all update types except ChatMember related updates

};

botClient.StartReceiving(

updateHandler: HandleUpdateAsync, //hàm xử lý khi có người chát đến

pollingErrorHandler: HandlePollingErrorAsync,

receiverOptions: receiverOptions,

cancellationToken: cts.Token

);

Task<User> me = botClient.GetMeAsync();

AddLog($"Bot begin working: @{me.Result.Username}");

async Task HandleUpdateAsync(ITelegramBotClient botClient, Update update, CancellationToken cancellationToken)

{

// Only process Message updates: https://core.telegram.org/bots/api#message

bool ok = false;

Telegram.Bot.Types.Message? message = null;

if (update.Message != null)

{

message = update.Message;

ok = true;

}

if (update.EditedMessage != null)

{

message = update.EditedMessage;

ok = true;

}

if (!ok || message == null) return;

string? messageText = message.Text;

if (messageText == null) return;

var username = message.Chat.Username;

var chatId = message.Chat.Id;

AddLog($"{username}: {messageText}");

string reply = "";

if (messageText.StartsWith("/help"))

{

//SqlService sqlService = new SqlService();

reply = "Hướng dẫn sử dụng bot 🤖 của Coffee Restaurent 😊:\n/order: để được gọi và đặt đồ ăn 🍵🍹\n/menu: để lấy menu của quán nhé 📄\n- Cần hỗ trợ: 0852072574 ☎️";

}

else if (messageText.StartsWith("/dl "))

{

string path = messageText.Substring(4);

string fn = System.IO.Path.GetFileName(path);

reply = $"Đã gửi file {fn} qua telegram rồi, click vào mũi tên để lưu về máy";

await using Stream stream = System.IO.File.OpenRead(path);

Telegram.Bot.Types.Message fileMessage = await botClient.SendDocumentAsync(

chatId: chatId,

document: InputFile.FromStream(stream: stream, fileName: fn),

caption: $"Download file {path}");

}

else if (messageText.StartsWith("/find "))

{

string studentName = messageText.Substring(6);

using(SqlService sql = new SqlService())

{

string query = $@"SELECT mssv, hoten FROM SinhVien WHERE hoten like N'%{studentName}%' OR mssv like N'%{studentName}%'";

DataTable dt = sql.GetDataQuery(query);

reply = $"Đã tìm thấy {dt.Rows.Count} sinh viên\n";

foreach (DataRow row in dt.Rows)

{

reply += "------------------\n";

reply += " Tên: " + row["hoten"].ToString() + "\n";

reply += " MSSV: " + row["mssv"].ToString() + "\n";

}

}

}

else

{

reply = "Xin lỗi, tớ không hiểu yêu cầu của bạn, hãy sử dụng /help để được trợ giúp nhé 😊";

}

// Echo received message text

Telegram.Bot.Types.Message sentMessage = await botClient.SendTextMessageAsync(

chatId: chatId,

text: reply

);

}

Task HandlePollingErrorAsync(ITelegramBotClient botClient, Exception exception, CancellationToken cancellationToken)

{

var ErrorMessage = exception switch

{

ApiRequestException apiRequestException

=> $"Telegram API Error:\n[{apiRequestException.ErrorCode}]\n{apiRequestException.Message}",

\_ => exception.ToString()

};

AddLog(ErrorMessage);

return Task.CompletedTask;

}

}

private void Form1\_Load(object sender, EventArgs e)

{

}

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

{

}

}

}

**V. Kết luận:**

- Trong bài này em vẫn chưa hoàn thành xong phần in hợp đồng vì em không biết lấy riêng dữ liệu từng sinh viên để in hợp đồng.

- Trong bài cũng còn nhiều phần chưa hoàn thiện em sẽ cố gắng hoàn thiện nốt.