# TRƯỜNG ĐẠI HỌC AN GIANG

# KHOA CÔNG NGHỆ THÔNG TIN



# BÁO CÁO ĐỒ ÁN

# MÔN: Lập trình .Net (TIE501)

# ĐỒ ÁN QUẢN LÝ HỌC SINH

# Nhóm báo cáo đồ án: Nhóm 14

# Sinh viên thực hiện:

# Nguyễn Phúc Tường - DTH225758

# Nguyễn Hoàng Thi - DTH25770

# An Giang 12/2024

**MỤC LỤC**

[Phần I: Mô tả hệ thống 1](#_Toc185442509)

[1. Giới thiệu chung về hệ thống 1](#_Toc185442510)

[2. Các chức năng chính của hệ thống 1](#_Toc185442511)

[2.1 Quản lý học sinh 1](#_Toc185442512)

[2.2 Quản lý năm học của học sinh 1](#_Toc185442513)

[2.3 Quản lý lớp học 1](#_Toc185442514)

[2.4 Phân tổ giáo viên 1](#_Toc185442515)

[2.5 Danh sách giáo viên 1](#_Toc185442516)

[2.6 Phân tích bảng điểm học sinh 2](#_Toc185442517)

[2.7 Xét hạnh kiểm học sinh qua từng học kỳ 2](#_Toc185442518)

[2.8 Quản lý học phí, BHYT 2](#_Toc185442519)

[Phần II: Thiết kế cở sở dữ liệu 2](#_Toc185442520)

[1. Tạo Database 2](#_Toc185442521)

[2. Tạo Table 3](#_Toc185442522)

[3. Tạo Khóa ngoại 4](#_Toc185442523)

[4. Quan hệ giữa các bảng (Relationship) 5](#_Toc185442524)

[Phần III: Giao diện hệ thống 6](#_Toc185442525)

[1. Login 6](#_Toc185442526)

[2. Admin 7](#_Toc185442527)

[3. Quản lý Năm học 8](#_Toc185442528)

[4. Quản lý Lớp học 9](#_Toc185442529)

[5. Quản lý Tổ giáo viên 10](#_Toc185442530)

[6. Quản lý Giáo viên 11](#_Toc185442531)

[7. Quản lý Học sinh 12](#_Toc185442532)

[8. Quản lý Thu ngân 14](#_Toc185442533)

[9. Quản lý Tài khoản 15](#_Toc185442534)

[10. Thông tin Giáo viên 16](#_Toc185442535)

[11. Hệ thống quản lý (Giáo viên) 17](#_Toc185442536)

[12. Quản lý học sinh (Giáo viên) 18](#_Toc185442537)

[13. Điểm số 19](#_Toc185442538)

[14. Đánh giá 21](#_Toc185442539)

[Phần IV: Data Transfer Object (DTO) 21](#_Toc185442540)

[Phần V: Data Access Layer – Lớp dữ liệu 22](#_Toc185442541)

[1. Connnection 22](#_Toc185442542)

[2. NamHocDAL 23](#_Toc185442543)

[3. LopHocDAL 24](#_Toc185442544)

[4. TaiKhoanDAL 27](#_Toc185442545)

[5. GiaoVienDAL 28](#_Toc185442546)

[6. HocSinhDAL 30](#_Toc185442547)

[7. BangDiemDAL 34](#_Toc185442548)

[8. HanhKiemDAL 37](#_Toc185442549)

[9. ToGVDAL 38](#_Toc185442550)

[10. ThuNganDAL 40](#_Toc185442551)

[Phần VI: Business Logic Layer - Lớp nghiệp vụ 42](#_Toc185442552)

[1. NamHocBLL 42](#_Toc185442553)

[2. LopHocBLL 43](#_Toc185442554)

[3. TaiKhoanBLL 44](#_Toc185442555)

[4. GiaoVienBLL 45](#_Toc185442556)

[5. HocSinhBLL 45](#_Toc185442557)

[6. BangDiemBLL 46](#_Toc185442558)

[7. HanhKiemBLL 48](#_Toc185442559)

[8. ToGVBLL 48](#_Toc185442560)

[9. ThuNganBLL 49](#_Toc185442561)

[Phần VII: Presentation Layer – Lớp giao diện 50](#_Toc185442562)

[1. frmLogin 50](#_Toc185442563)

[2. frmAdmin 51](#_Toc185442564)

[3. frmQuanLyNH 52](#_Toc185442565)

[4. frmQuanLyLH 53](#_Toc185442566)

[5. frmQuanLyToGV 55](#_Toc185442567)

[6. frmQuanLyGV 57](#_Toc185442568)

[7. frmQuanLyHS 60](#_Toc185442569)

[8. frmQuanLyTN 63](#_Toc185442570)

[9. frmQuanLyTK 67](#_Toc185442571)

[10. frmThongTinGV 68](#_Toc185442572)

[11. frmStudentManagement 69](#_Toc185442573)

[12. frmQuanLy 70](#_Toc185442574)

[13. frmDiemSo 71](#_Toc185442575)

[14. frmDanhGiaHanhKiem 75](#_Toc185442576)

[Tài liệu tham khảo 78](#_Toc185442577)

# Phần I: Mô tả hệ thống

## 1. Giới thiệu chung về hệ thống

Hệ thống quản lý học sinh được xây dựng nhằm giúp nhà trường quản lý một cách hiệu quả các thông tin liên quan đến học sinh, giáo viên và các công tác quản lý hành chính trong môi trường giáo dục. Hệ thống hỗ trợ các chức năng quản lý học sinh, lớp học, năm học, giáo viên, bảng điểm, hạnh kiểm, học phí, và bảo hiểm y tế của học sinh. Mỗi chức năng được thiết kế đơn giản và dễ sử dụng, giúp tiết kiệm thời gian và công sức trong việc quản lý.

Hệ thống ứng dụng được viết theo mô hình 3 lớp và có sử dụng DataSet với mô hình Disconnected nhằm dễ dàng quản lý xây dựng hệ thống và thao tác dễ dàng với cơ sở dữ liệu.

## 2. Các chức năng chính của hệ thống

### 2.1 Quản lý học sinh

* Hệ thống cho phép quản lý thông tin cá nhân của học sinh, bao gồm tên, ngày sinh, giới tính, địa chỉ, số điện thoại, và các thông tin liên quan đến gia đình.
* Hỗ trợ tìm kiếm và lọc học sinh theo các tiêu chí tên, lớp học, hoặc năm học.

### 2.2 Quản lý năm học của học sinh

* + Hệ thống ghi nhận các năm học mà học sinh đã tham gia, bao gồm thông tin về lớp học, môn học.

### 2.3 Quản lý lớp học

* + Hệ thống cho phép quản lý thông tin về các lớp học, bao gồm tên lớp, số lượng học sinh, giáo viên chủ nhiệm và lịch học.
  + Hỗ trợ phân công lớp học cho học sinh và theo dõi sự thay đổi của lớp học trong suốt năm học.
  + Tạo danh sách lớp học và cho phép tìm kiếm lớp học dựa trên các tiêu chí như tên lớp, giáo viên chủ nhiệm, hay mã học sinh.

### 2.4 Phân tổ giáo viên

* + Hệ thống hỗ trợ phân công giảng dạy cho giáo viên theo mã tổ , môn giảng dạy
  + Hệ thống lưu trữ các thông tin của tổ giáo viên, bao gồm tên tổ, mã tổ và trưởng tổ của từng môn học.

### 2.5 Danh sách giáo viên

* + Quản lý đầy đủ thông tin của giáo viên, bao gồm các dữ liệu cá nhân và mã giáo viên, thuộc tổ nào.
  + Hỗ trợ tra cứu và tìm kiếm thông tin giáo viên theo các tiêu chí như tên, môn học, hoặc lớp giảng dạy.

### 2.6 Phân tích bảng điểm học sinh

* + Hệ thống cho phép nhập và lưu trữ điểm của học sinh theo từng môn học, học kỳ và năm học.
  + Tự động tính toán điểm tổng kết của học sinh và phân loại kết quả học tập (loại giỏi, khá, trung bình, yếu).
  + Hỗ trợ phân tích kết quả học tập của học sinh, bao gồm báo cáo chi tiết về điểm số từng môn học.

### 2.7 Xét hạnh kiểm học sinh qua từng học kỳ

* + Hệ thống cho phép nhập và quản lý thông tin về hạnh kiểm của học sinh qua từng học kỳ.
  + Cung cấp công cụ để giáo viên đánh giá và ghi nhận các hành vi của học sinh trong suốt học kỳ.
  + Báo cáo kết quả xét hạnh kiểm theo các tiêu chí: hạnh kiểm tốt, khá, trung bình, hoặc yếu.

### 2.8 Quản lý học phí, BHYT

* + Quản lý thông tin về học phí của học sinh, bao gồm các khoản phí phải đóng, BHYTT và thời gian đóng học phí, BHYT
  + Hệ thống hỗ trợ tính toán và theo dõi các khoản học phí đã thu và còn thiếu.

# Phần II: Thiết kế cơ sở dữ liệu

## 1. Tạo Database

CREATE DATABASE QLHS

ON

(

NAME = QLHS\_DATA,

FILENAME = 'D:\QLHS.MDF',

SIZE = 8,

MAXSIZE = UNLIMITED,

FILEGROWTH = 2

)

LOG ON

(

NAME = QLHS\_LOG,

FILENAME = 'D:\QLHS.LDF',

SIZE = 8,

MAXSIZE = UNLIMITED,

FILEGROWTH = 2

)

## 2. Tạo Table

**2.1. LOGIN\_TABLE**

CREATE TABLE LOGIN\_TABLE

(

USERNAME VARCHAR(15),

PASSWORD VARCHAR(15),

QUYEN NVARCHAR(5),

CONSTRAINT PK\_LOGIN\_TABLE PRIMARY KEY(USERNAME)

)

**2.2. HOCSINH**

CREATE TABLE HOCSINH

(

MAHS CHAR(5),

HOTEN NVARCHAR(255) NOT NULL,

NGAYSINH DATE,

DIACHI NVARCHAR(255),

SODT VARCHAR(10),

MALOP CHAR(5) NOT NULL,

GIOITINH NVARCHAR(3),

USERNAME VARCHAR(15)

CONSTRAINT PK\_HOCSINH PRIMARY KEY(MAHS)

)

**2.3 NAMHOC**

CREATE TABLE NAMHOC

(

MANH CHAR(5),

MAHK TINYINT,

CONSTRAINT PK\_NAMHOC PRIMARY KEY(MANH, MAHK)

)

**2.4 LOPHOC**

CREATE TABLE LOPHOC

(

MALOP CHAR(5),

TENLOP NVARCHAR(15) NOT NULL,

SISO TINYINT CHECK(SISO >= 0 AND SISO <=50),

GVCN CHAR(5) NOT NULL,

MANH CHAR(5),

MAHK TINYINT,

CONSTRAINT PK\_LOPHOC PRIMARY KEY(MALOP)

)

**2.5 TOGV**

CREATE TABLE TOGV

(

MATOGV CHAR(5),

TRUONGTOGV CHAR(5),

TENTOGV NVARCHAR(255),

CONSTRAINT PK\_TOGV PRIMARY KEY(MATOGV)

)

**2.6 GIAOVIEN**

CREATE TABLE GIAOVIEN

(

MAGV CHAR(5),

HOTEN NVARCHAR(255) NOT NULL,

NGAYSINH DATE,

QUEQUAN NVARCHAR(255),

SODT VARCHAR(10),

EMAIL VARCHAR(255) CHECK(EMAIL LIKE '%@%' AND EMAIL NOT LIKE '% %') UNIQUE,

GIOITINH NVARCHAR(3),

MONGD NVARCHAR(30),

MATOGV CHAR(5),

USERNAME VARCHAR(15)

CONSTRAINT PK\_GIAOVIEN PRIMARY KEY(MAGV)

)

**2.7 BANGDIEM**

CREATE TABLE BANGDIEM

(

MAHS CHAR(5),

MANH CHAR(5),

MAHK TINYINT,

TOAN FLOAT,

NGUVAN FLOAT,

NGOAINGU FLOAT,

VATLY FLOAT,

HOAHOC FLOAT,

SINHHOC FLOAT,

GDCD FLOAT,

DIALY FLOAT,

LICHSU FLOAT,

DTB FLOAT,

CONSTRAINT PK\_BANGDIEM PRIMARY KEY(MAHS, MAHK, MANH)

)

**2.8 HANHKIEM**

CREATE TABLE HANHKIEM

(

MAHS CHAR(5),

MANH CHAR(5),

MAHK TINYINT,

LOAIHK NVARCHAR(30),

CONSTRAINT PK\_HANHKIEM PRIMARY KEY(MAHS, MANH, MAHK)

)

**2.9 THUNGAN**

CREATE TABLE HANHKIEM

(

MAHS CHAR(5),

MANH CHAR(5),

MAHK TINYINT,

LOAIHK NVARCHAR(30),

CONSTRAINT PK\_HANHKIEM PRIMARY KEY(MAHS, MANH, MAHK)

)

## 3. Tạo Khóa ngoại

ALTER TABLE HOCSINH

ADD CONSTRAINT FK\_HOCSINH\_LOPHOC FOREIGN KEY (MALOP) REFERENCES LOPHOC(MALOP),

CONSTRAINT FK\_HOCSINH\_LOGIN\_TABLE FOREIGN KEY(USERNAME) REFERENCES LOGIN\_TABLE(USERNAME)

ALTER TABLE TOGV

ADD CONSTRAINT FK\_TOGV\_GIAOVIEN FOREIGN KEY (TRUONGTOGV) REFERENCES GIAOVIEN(MAGV)

ALTER TABLE GIAOVIEN

ADD CONSTRAINT FK\_GIAOVIEN\_TOGV FOREIGN KEY (MATOGV) REFERENCES TOGV(MATOGV),

CONSTRAINT FK\_GIAOVIEN\_LOGIN\_TABLE FOREIGN KEY(USERNAME) REFERENCES LOGIN\_TABLE(USERNAME)

ALTER TABLE LOPHOC

ADD CONSTRAINT FK\_LOPHOC\_NAMHOC FOREIGN KEY(MANH, MAHK) REFERENCES NAMHOC(MANH, MAHK),

CONSTRAINT FK\_LOPHOC\_GIAOVIEN FOREIGN KEY(GVCN) REFERENCES GIAOVIEN(MAGV)

ALTER TABLE BANGDIEM

ADD CONSTRAINT FK\_BANGDIEM\_HOCSINH FOREIGN KEY(MAHS) REFERENCES HOCSINH(MAHS),

CONSTRAINT FK\_BANGDIEM\_NAMHOC FOREIGN KEY(MANH, MAHK) REFERENCES NAMHOC(MANH, MAHK)

ALTER TABLE THUNGAN

ADD CONSTRAINT FK\_THUNGAN\_HOCSINH FOREIGN KEY(MAHS) REFERENCES HOCSINH(MAHS),

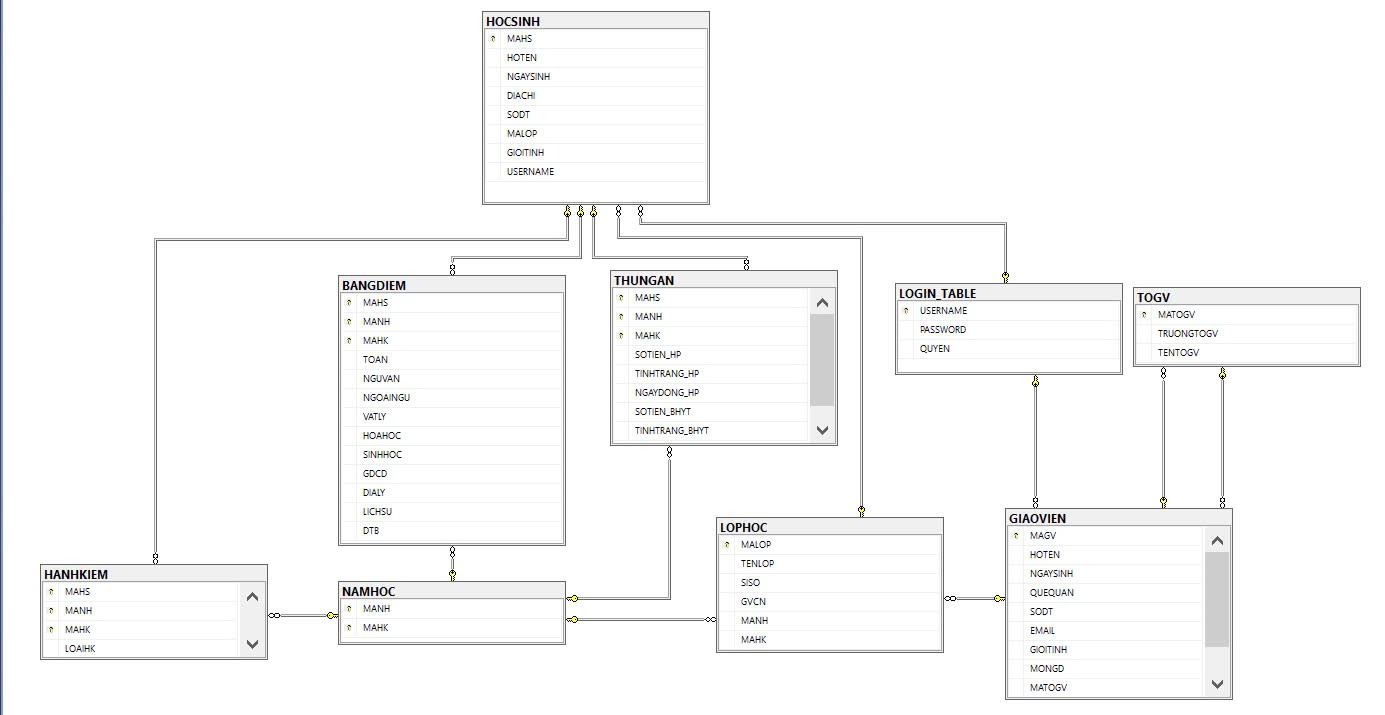
CONSTRAINT FK\_THUNGAN\_NAMHOC FOREIGN KEY(MANH, MAHK) REFERENCES NAMHOC(MANH, MAHK)

ALTER TABLE HANHKIEM

ADD CONSTRAINT FK\_HANHKIEM\_HOCSINH FOREIGN KEY(MAHS) REFERENCES HOCSINH(MAHS),

CONSTRAINT FK\_HANHKIEM\_NAMHOC FOREIGN KEY(MANH, MAHK) REFERENCES NAMHOC(MANH, MAHK)

## 4. Quan hệ giữa các bảng (Relationship)



# Phần III: Giao diện hệ thống

## 1. Login

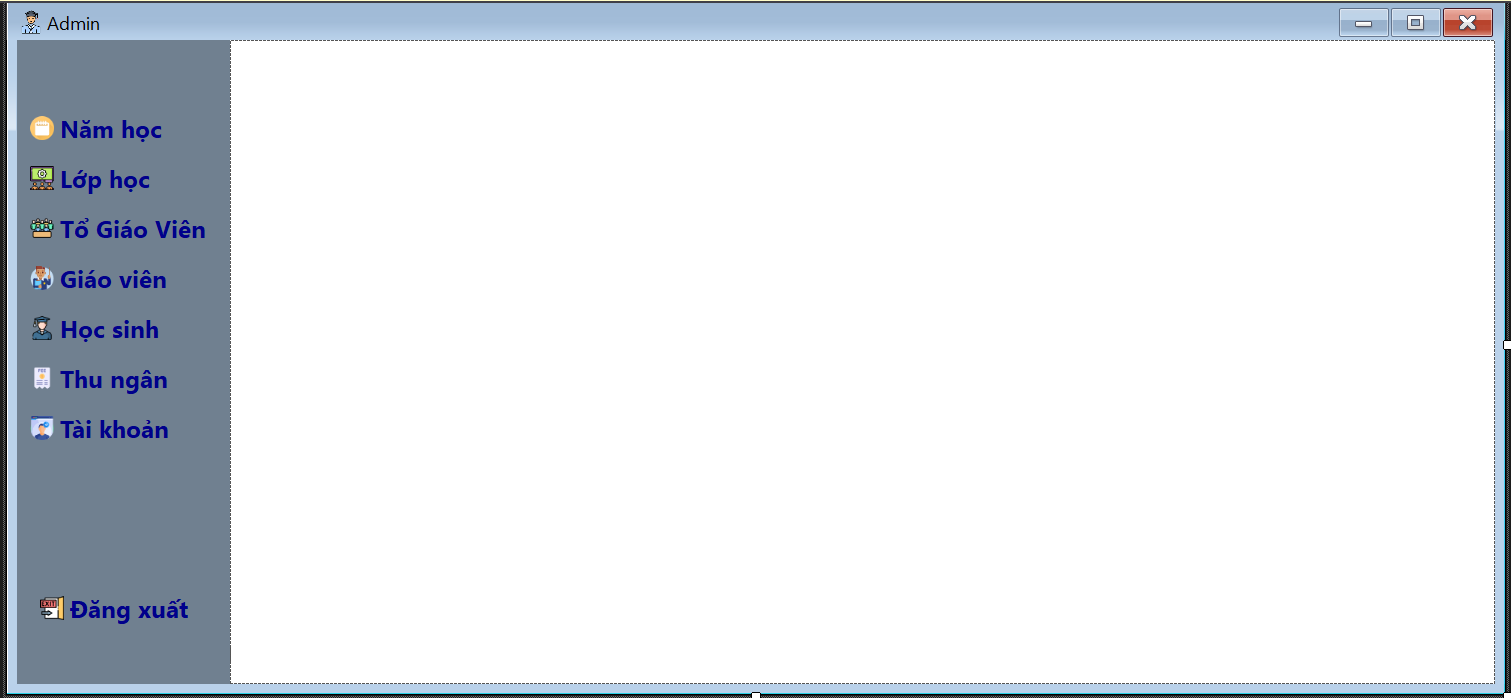
- Tên form: frmLogin



- Các thành phần:

|  |  |
| --- | --- |
| Name | Text |
| lblTenHeThong | Hệ thống quản lý học sinh |
| picLogo |  |
| lblUsername | Tên tài khoản |
| lblPassword | Mật khẩu |
| txtUsername |  |
| txtPassword |  |
| btnLogin | Đăng nhập |
| btnExit | Thoát |

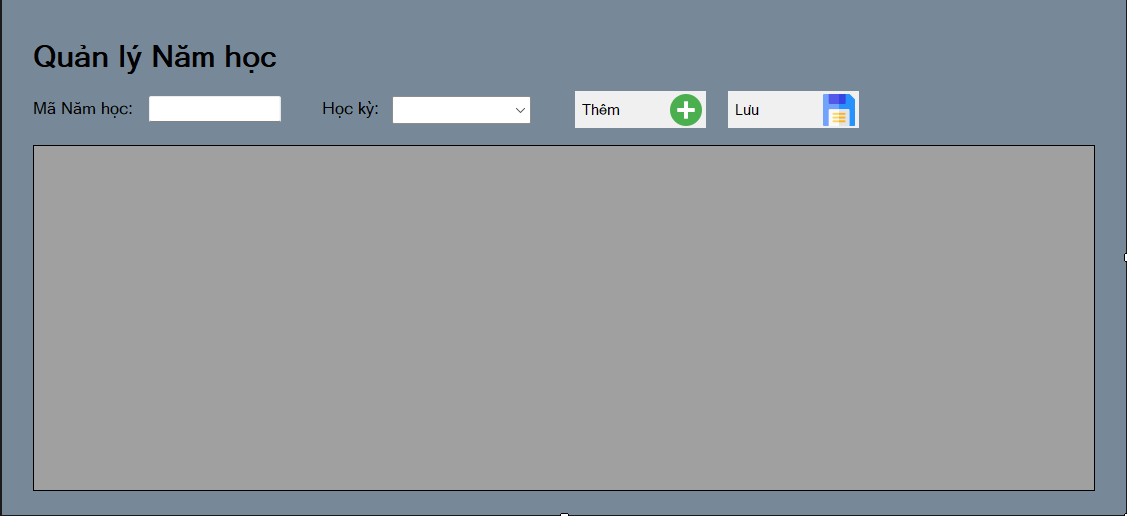
## 2. Admin

- Tên form: frmAdmin

- Các thành phần:

|  |  |
| --- | --- |
| Name | Text |
| stMnuItemQLNH | Năm học |
| tsMnuItemQLLH | Lớp học |
| tsMnuItemToGV | Tổ Giáo Viên |
| tsMnuItemQLGV | Giáo viên |
| tsMnuItemQLHS | Học sinh |
| tsMnuItemQLTN | Thu ngân |
| tsMnuItemQLTK | Tài khoản |
| tsMnuItemDangXuat | Đăng xuất |

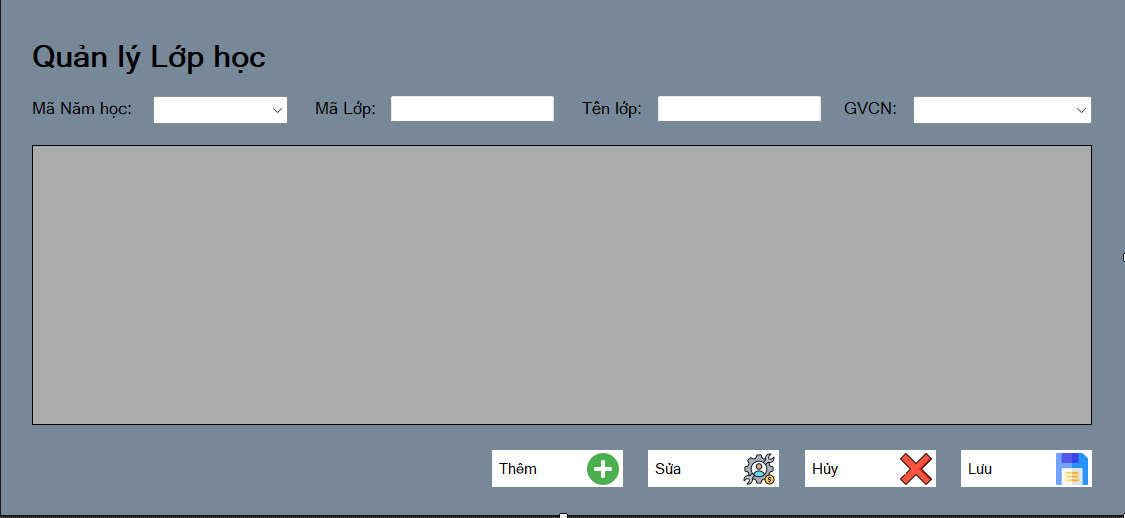
## 3. Quản lý Năm học

- Tên form: frmQuanLyNH

- Các thành phần

|  |  |
| --- | --- |
| Name | Text |
| lblQuanLyNamHoc | Quản lý Năm học |
| lblMaNH | Mã Năm học |
| txtNamHoc |  |
| lblHocKy | Học kỳ |
| cbHocKy |  |
| btnThemMoi | Thêm |
| btnLuu | Lưu |

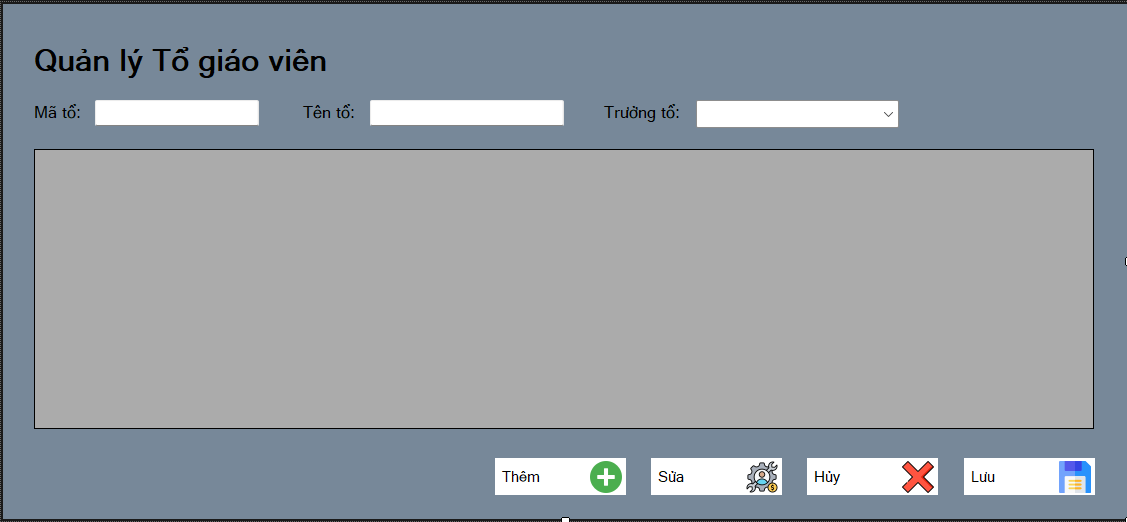
## 4. Quản lý Lớp học

- Tên form: frmQuanLyLH

- Các thành phần

|  |  |
| --- | --- |
| Name | Text |
| lblQuanLyLopHoc | Quản lý Lớp học |
| lblMaNH | Mã năm học |
| cbMaNH |  |
| lblMaLop | Mã lớp |
| txtMaLop |  |
| lblTenLop | Tên lớp |
| txtTenLop |  |
| lblGVCN | GVCN |
| cbGVCN |  |
| btnThem | Thêm |
| btnSua | Sửa |
| btnHuy | Hủy |
| btnLuu | Lưu |

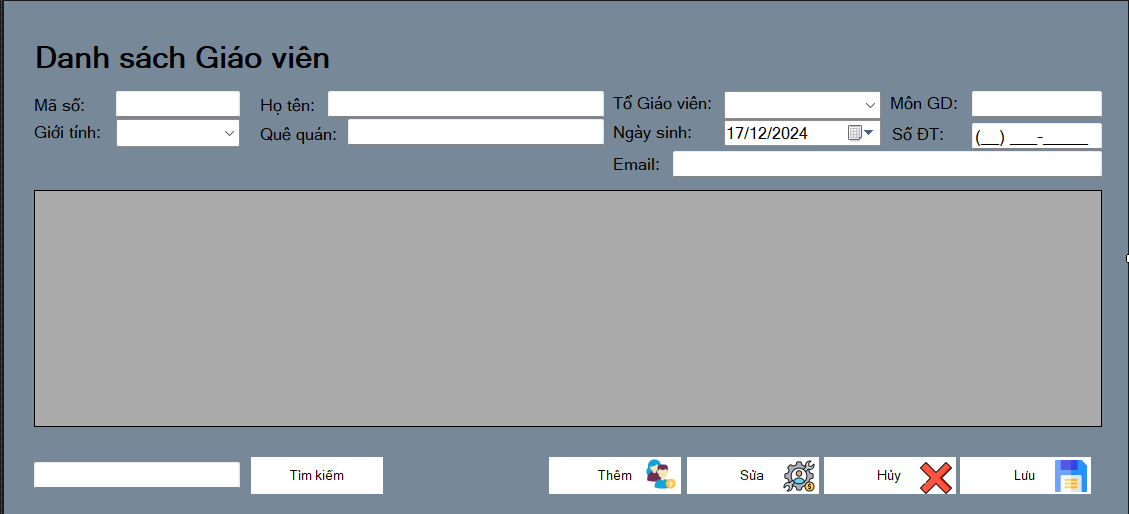
## 5. Quản lý Tổ giáo viên

- Tên form: frmQuanLyToGV

- Các thành phần

|  |  |
| --- | --- |
| Name | Text |
| lblPCToGV | Quản lý Tổ giáo viên |
| lblMaTo | Mã tổ |
| txtMaToGV |  |
| lblTenTo | Tên tổ |
| txtTenTo |  |
| lblTruongTo | Trưởng tổ |
| cbTruongTo |  |
| btnThem | Thêm |
| btnSua | Sửa |
| btnHuy | Hủy |
| btnLuu | Lưu |

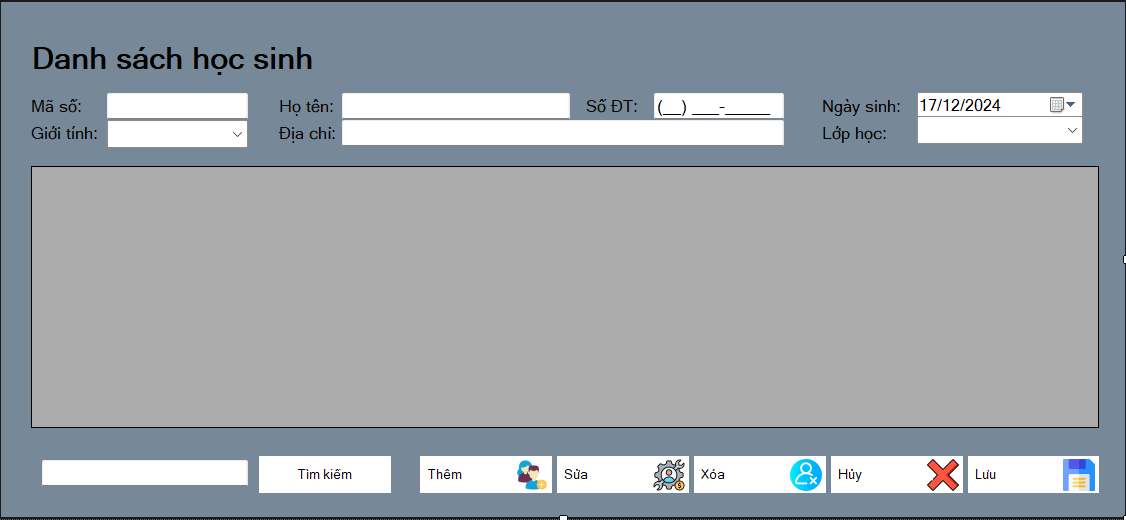
## 6. Quản lý Giáo viên

- Tên form: frmQuanLyGV

- Các thành phần:

|  |  |
| --- | --- |
| Name | Text |
| lblDanhSachGV | Danh sách Giáo viên |
| lblMaSo | Mã số |
| txtMaSo |  |
| lblHoTen | Họ tên |
| txtHoTen |  |
| lblToGV | Tổ Giáo viên |
| txtToGV |  |
| lblMonGD | Môn GD |
| txtMonGD |  |
| lblGioiTinh | Giới tính |
| cbGioiTinh |  |
| lblQueQuan | Quê quán |
| txtQueQuan |  |
| txtNgaySinh | Ngày sinh |
| dtpNgaySinh |  |
| lblSoDT | Số ĐT |
| mtxtSoDT |  |
| lblEmail | Email |
| txtEmail |  |
| btnTimKiem | Tìm Kiếm |
| txtTimKiem |  |
| btnThemMoi | Thêm |
| btnSua | Sửa |
| btnHuy | Hủy |
| btnLuu | Lưu |

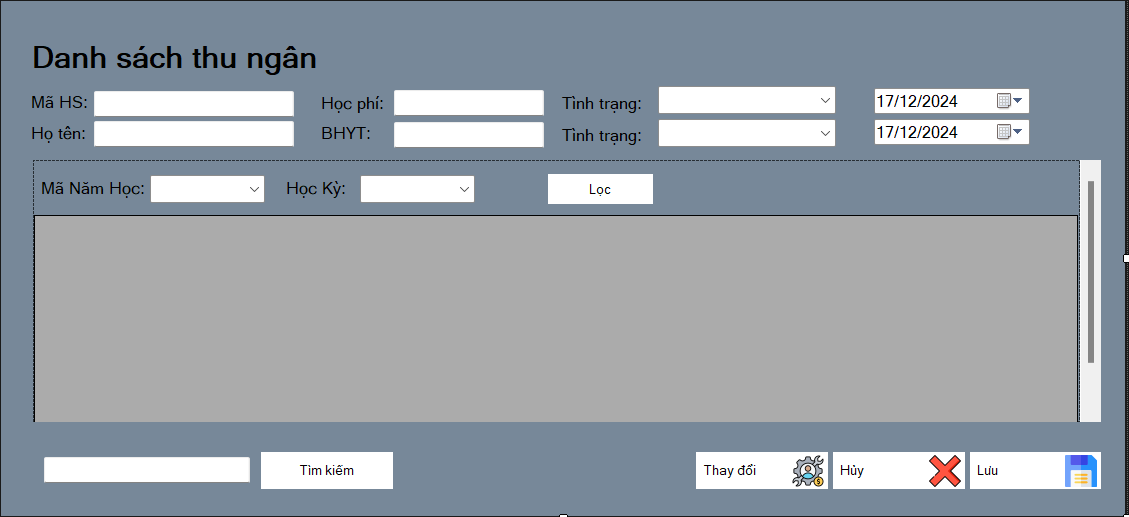
## 7. Quản lý Học sinh

- Tên form: frmQuanLyHS

- Các thành phần

|  |  |
| --- | --- |
| Name | Text |
| lblThongTin | Danh sách học sinh |
| lblMaSo | Mã số |
| txtMaSo |  |
| lblHoTen | Họ tên |
| txtHoTen |  |
| lblSoDT | Số ĐT |
| txtSoDT |  |
| txtNgaySinh | Ngày sinh |
| dtpNgaySinh |  |
| lblGioiTinh | Giới tính |
| cbGioiTinh |  |
| lblDiaChi | Địa chỉ |
| txtDiaChi |  |
| lblMaLop | Lớp học |
| cbMaLop |  |
| btnTimKiem | Tìm kiếm |
| txtTimKiem |  |
| btnThemMoi | Thêm |
| btnSua | Sửa |
| btnXoa | Xóa |
| btnHuy | Hủy |
| btnLuu | Lưu |

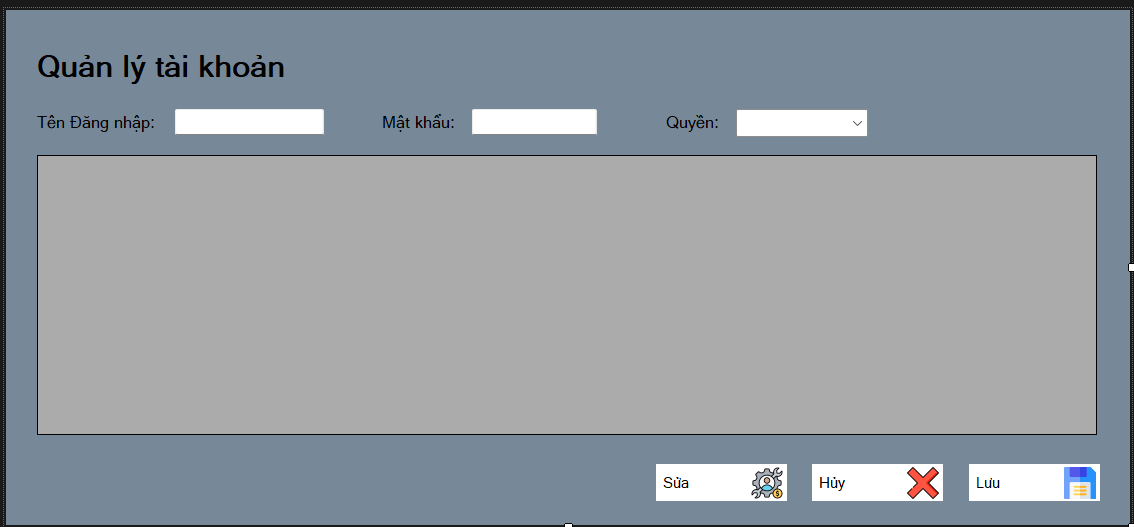
## 8. Quản lý Thu ngân

- Tên form: frmQuanLyTN

- Các thành phần

|  |  |
| --- | --- |
| Name | Text |
| lblThuNgan | Danh sách thu ngân |
| lblMaSo | Mã HS |
| txtMaSo |  |
| lblHocPhi | Học phí |
| txtHocPhi |  |
| lblHoTen | Họ tên |
| txtHoTen |  |
| lblBHYT | BHYT |
| txtBHYT |  |
| lblTTHP | Tình trạng |
| cbTTHP |  |
| lblTTBHYT | Tình trạng |
| cbTTBHYT |  |
| dtpNgayDongHP |  |
| dtpNgayDongBHYT |  |
| lblMaNH | Mã Năm Học |
| cbMaNH |  |
| lblHocKy | Học Kỳ |
| cbHocKy |  |
| btnLoc | Lọc |
| btnTimKiem | Tìm Kiếm |
| txtTimKiem |  |
| btnThayDoi | Thay đổi |
| btnHuy | Hủy |
| btnLuu | Lưu |

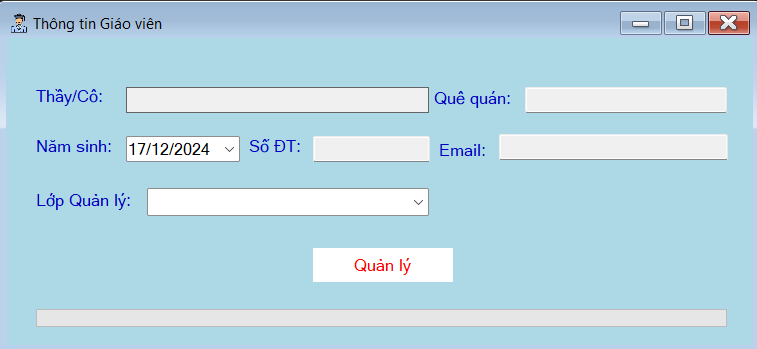
## 9. Quản lý Tài khoản

- Tên form: frmQuanLyTK

- Các thành phần

|  |  |
| --- | --- |
| Name | Text |
| lblQLTK | Quản lý tài khoản |
| lblTenDN | Tên Đăng nhập |
| txtTenDN |  |
| lblMatKhau | Mật khẩu |
| txtMatKhau |  |
| lblQuyen | Quyền |
| cbQuyen |  |
| btnSua | Sửa |
| btnHuy | Hủy |
| btnLuu | Lưu |

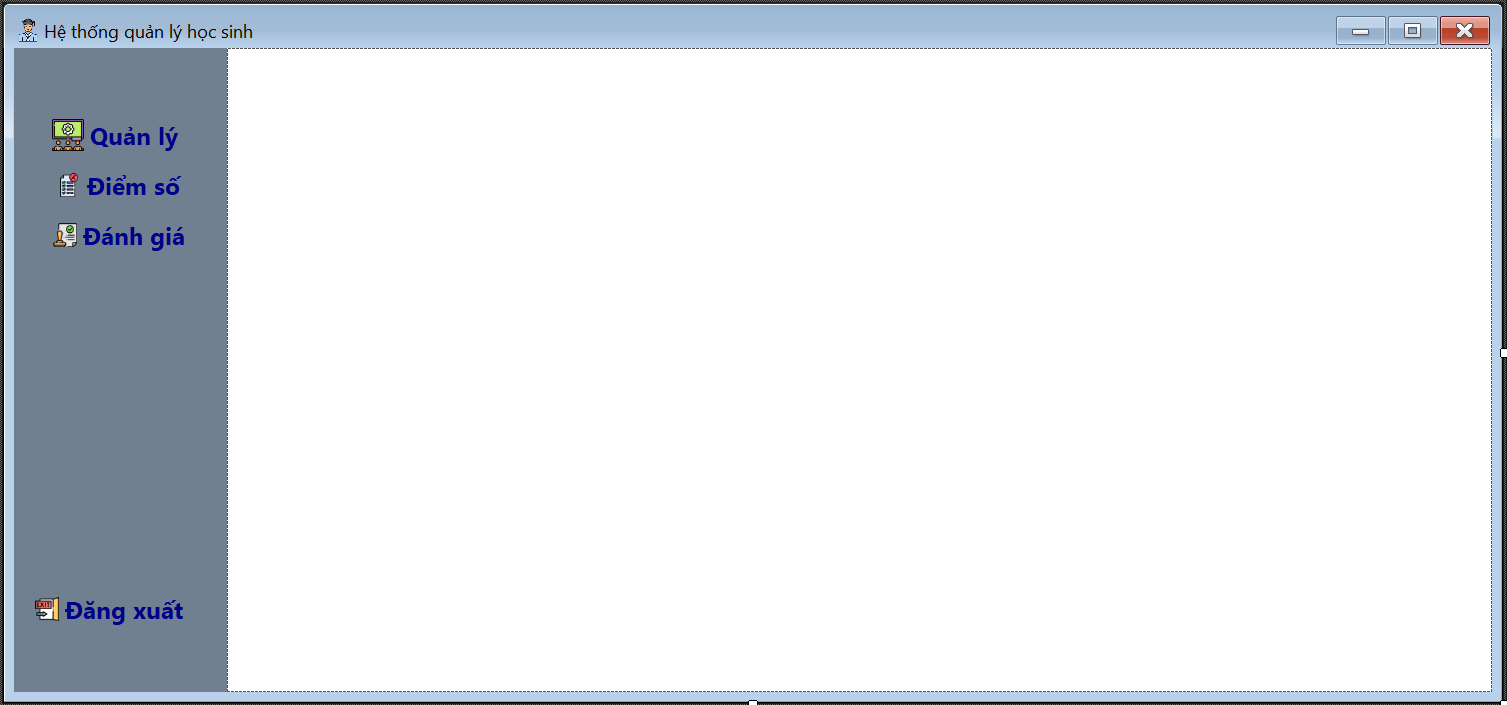
## 10. Thông tin Giáo viên

- Tên form: frmThongTinGV

- Các thành phần

|  |  |
| --- | --- |
| Name | Text |
| lblHoTen | Thầy/Cô |
| txtHoTen |  |
| lblQueQuan | Quê quán |
| txtQueQuan |  |
| lblNamSinh | Năm sinh |
| dtpNamSinh |  |
| lblSoDT | Số ĐT |
| txtSoDT |  |
| lblEmail | Email |
| txtEmail |  |
| lblLopQL | Lớp Quản Lý |
| cbMaLop |  |
| btnQuanLy | Quản lý |
| pbLoading |  |

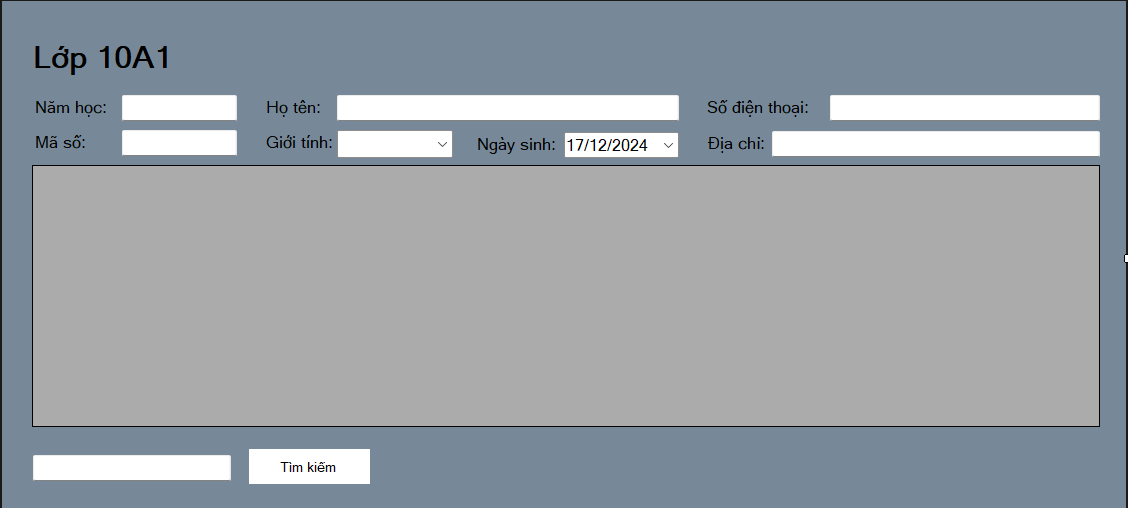
## 11. Hệ thống quản lý (Giáo viên)

- Tên form: frmStudentManagement

- Các thành phần

|  |  |
| --- | --- |
| Name | Text |
| tsMnItemQuanLy | Quản lý |
| tsMnItemDiemSo | Điểm số |
| tsMnItemDanhGia | Đánh giá |
| tsMnuItemDangXuat | Đăng xuất |

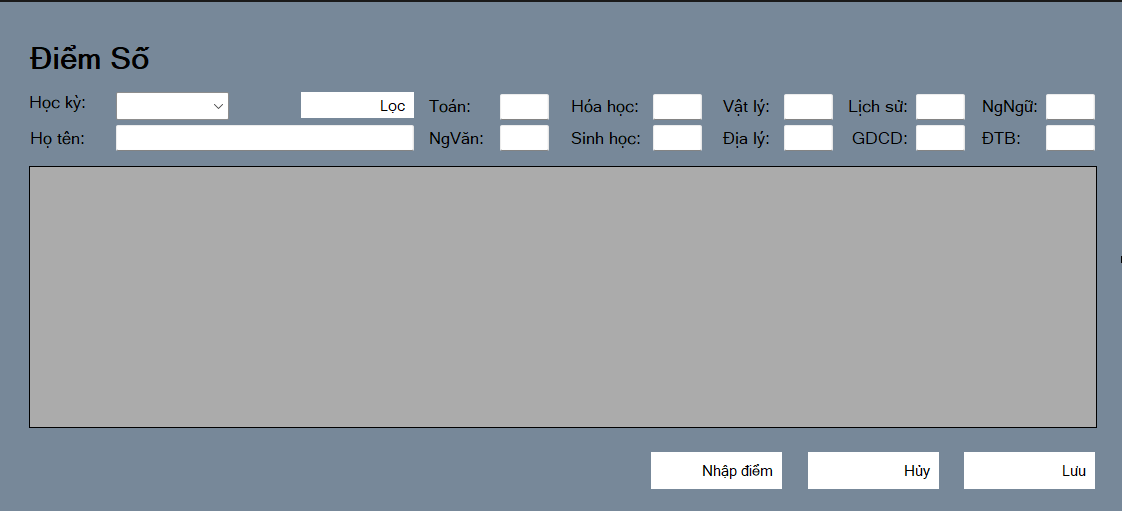
## 12. Quản lý học sinh (Giáo viên)

- Tên form: frmQuanLy

- Các thành phần

|  |  |
| --- | --- |
| Name | Text |
| lblLop | Lớp 10A1 |
| lblNamHoc | Năm học |
| txtNamHoc |  |
| lblMaSo | Mã số |
| txtMaso |  |
| lblHoTen | Họ tên |
| txtHoTen |  |
| lblGioiTinh | Giới tính |
| cbGioiTinh |  |
| txtNgaySinh | Ngày sinh |
| dtpNgaySinh |  |
| lblSoDT | Số điện thoại |
| txtSoDT |  |
| lblDiaChi | Địa chỉ |
| txtDiaChi |  |
| txtTimKiem |  |
| btnTim | Tìm kiếm |

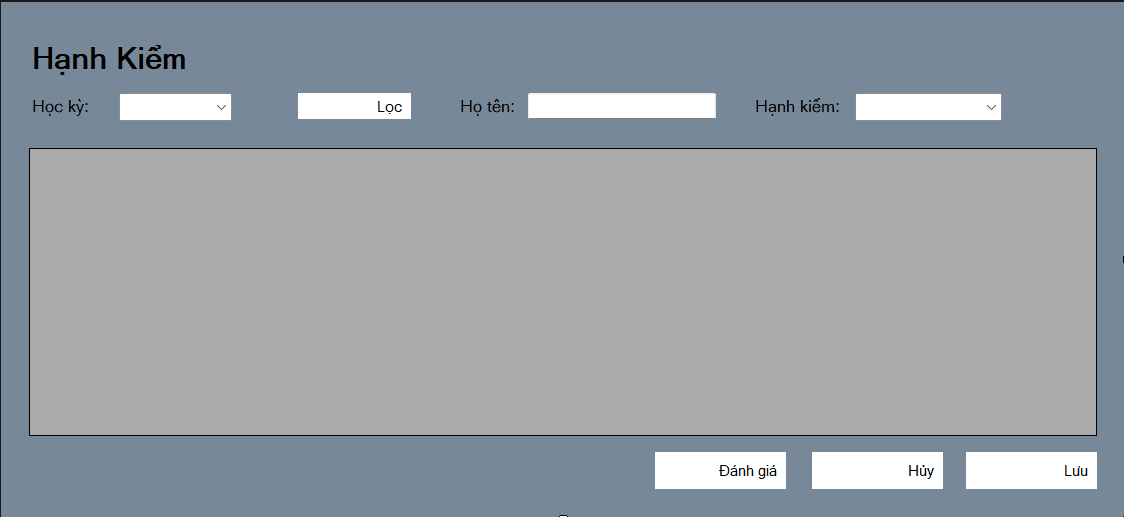
## 13. Điểm số

- Tên form: frmDiemSo

- Các thành phần

|  |  |
| --- | --- |
| Name | Text |
| lblQLDS | Điểm Số |
| lblHocKy | Học kỳ |
| lblHoTen | Họ tên |
| cbHocKy |  |
| txtHoTen |  |
| btnLoc | Lọc |
| lblToan | Toán |
| lblVan | NgVăn |
| lblHoa | Hóa học |
| lblSinh | Sinh học |
| lblLy | Vật lý |
| lblDia | Địa lý |
| lblSu | Lịch sử |
| lblGDCD | GDCD |
| lblNgNgu | NgNgữ |
| lblDTB | ĐTB |
| txtToan | txt |
| txtVan |  |
| txtHoa |  |
| txtSinh |  |
| txtLy |  |
| txtDia |  |
| txtSu |  |
| txtGDCD |  |
| txtNgNgu |  |
| txtDTB |  |
| btnNhapDiem | Nhập điểm |
| btnHuy | Hủy |
| btnLuu | Lưu |

## 14. Đánh giá

- Tên form: frmDanhGiaHanhKiem

- Các thành phần

|  |  |
| --- | --- |
| Name | Text |
| lblDanhGiaHK | Hạnh Kiểm |
| lblHocKy | Học kỳ |
| btnLoc | Lọc |
| lblHoTen | Họ tên |
| lblLoaiHK | Hạnh kiểm |
| btnDanhGia | Đánh giá |
| btnHuy | Hủy |
| btnLuu | Lưu |
| cbHocKy |  |
| txtHoTen |  |
| cbLoaiHK |  |

# Phần IV: Data Transfer Object (DTO)

Là lớp dùng để truyền dữ liệu giữa các lớp hoặc các tầng trong ứng dụng, thường chỉ chứa các thuộc tính (properties) và không chứa logic xử lý nghiệp vụ. Giúp đóng góp dữ liệu và truyền dữ liệu rõ ràng, an toàn giữa các lớp.

Mô hình biểu diễn các lớp trong DTO:



# Phần V: Data Access Layer – Lớp dữ liệu

Lớp dữ liệu dùng để truy xuất dữ liệu, đọc và ghi dữ liệu từ cơ sở dữ liệu vào ứng dụng.

Lớp này chứa các phương thức kết nối, truy vấn SQL và thao tác với cơ sở dữ liệu. Kết quả sẽ được trả về lớp nghiệp vụ (BLL)

## 1. Connnection

public class Connection

{

// Các field

private SqlConnection \_conn;

private DataSet \_dataSet;

// Các properties

public SqlConnection Conn { get => \_conn; set => \_conn = value; }

public DataSet DataSet { get => \_dataSet; set => \_dataSet = value; }

// Constructor

public Connection()

{

\_conn = new SqlConnection(@"Data Source=DESKTOP-Q9F3DFL\HOANGTHI;Initial Catalog=QLHS;User ID=sa;Pwd=130804;TrustServerCertificate=True");

\_dataSet = new DataSet();

}

// Execute Stored Procedure

public DataTable SelectQuery(string procName, SqlParameter[] para)

{

DataTable dt = new DataTable();

try

{

SqlCommand cmd = new SqlCommand(procName, Conn);

if(para != null)

cmd.Parameters.AddRange(para);

cmd.CommandType = CommandType.StoredProcedure;

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(cmd);

Conn.Open();

sqlDataAdapter.Fill(dt);

}

catch (Exception ex)

{

throw ex;

}

finally

{

if (Conn.State == ConnectionState.Open)

Conn.Close();

}

return dt;

}

public DataSet GetDataSet(SqlDataAdapter adapter, string tblName)

{

adapter.Fill(DataSet, tblName);

return DataSet;

}

public bool CheckCondition(DataTable table, string condition)

{

DataRow[] rows = table.Select(condition);

return rows.Length < 1 ? true : false; // Không tồn tại giá trị

}

public void Save(SqlDataAdapter dataAdapter, string tblName)

{

dataAdapter.Update(DataSet, tblName);

}

}

## 2. NamHocDAL

public class NamHocDAL : Connection

{

private SqlDataAdapter \_da;

public NamHocDAL()

{

string strSQL = "SELECT \* FROM NAMHOC";

\_da = new SqlDataAdapter(strSQL, Conn);

SqlCommandBuilder sqlCommandBuilder = new SqlCommandBuilder(\_da);

}

public DataSet GetDataSet()

{

return GetDataSet(\_da, "tblNAMHOC");

}

public DataTable GetDataMaNH()

{

return SelectQuery("sp\_Select\_NamHoc\_ByMaHK", null);

}

public string Insert(NamHoc namHoc)

{

if (this.CheckPrimary(namHoc)==false)

return "Mã năm học và Học kỳ đã tồn tại";

DataRow row = DataSet.Tables["tblNAMHOC"].NewRow();

row["MANH"] = namHoc.MaNH;

row["MAHK"] = namHoc.MaHK;

DataSet.Tables["tblNAMHOC"].Rows.Add(row);

return "Thêm Thành Công";

}

public bool CheckPrimary(NamHoc namHoc)

{

DataTable dt = DataSet.Tables["tblNAMHOC"];

string condition = $"MaNH = '{namHoc.MaNH}' AND MaHK = '{namHoc.MaHK}'";

return CheckCondition(dt, condition);

}

public void Save()

{

Save(\_da, "tblNAMHOC");

}

}

## 3. LopHocDAL

public class LopHocDAL : Connection

{

private SqlDataAdapter \_da;

public LopHocDAL()

{

string strSQL = "SELECT lh.MANH, lh.MALOP, lh.TENLOP, lh.SISO, lh.GVCN, lh.MAHK, gv.HOTEN FROM LOPHOC lh, GIAOVIEN gv WHERE lh.GVCN = gv.MAGV";

\_da = new SqlDataAdapter(strSQL, Conn);

// Manually specify the commands

\_da.InsertCommand = new SqlCommand("INSERT INTO LOPHOC (MANH, MALOP, TENLOP, SISO, GVCN, MAHK) VALUES (@MANH, @MALOP, @TENLOP, @SISO, @GVCN, @MAHK)", Conn);

\_da.UpdateCommand = new SqlCommand("UPDATE LOPHOC SET TENLOP = @TENLOP, SISO = @SISO, GVCN = @GVCN, MAHK = @MAHK WHERE MALOP = @MALOP", Conn);

\_da.DeleteCommand = new SqlCommand("DELETE FROM LOPHOC WHERE MALOP = @MALOP", Conn);

// Add parameters for InsertCommand

\_da.InsertCommand.Parameters.Add("@MANH", SqlDbType.VarChar, 50, "MANH");

\_da.InsertCommand.Parameters.Add("@MALOP", SqlDbType.VarChar, 50, "MALOP");

\_da.InsertCommand.Parameters.Add("@TENLOP", SqlDbType.NVarChar, 50, "TENLOP");

\_da.InsertCommand.Parameters.Add("@SISO", SqlDbType.Int, 4, "SISO");

\_da.InsertCommand.Parameters.Add("@GVCN", SqlDbType.VarChar, 50, "GVCN");

\_da.InsertCommand.Parameters.Add("@MAHK", SqlDbType.VarChar, 50, "MAHK");

// Add parameters for UpdateCommand

\_da.UpdateCommand.Parameters.Add("@TENLOP", SqlDbType.NVarChar, 50, "TENLOP");

\_da.UpdateCommand.Parameters.Add("@SISO", SqlDbType.Int, 4, "SISO");

\_da.UpdateCommand.Parameters.Add("@GVCN", SqlDbType.VarChar, 50, "GVCN");

\_da.UpdateCommand.Parameters.Add("@MAHK", SqlDbType.VarChar, 50, "MAHK");

\_da.UpdateCommand.Parameters.Add("@MALOP", SqlDbType.VarChar, 50, "MALOP");

// Add parameters for DeleteCommand

\_da.DeleteCommand.Parameters.Add("@MALOP", SqlDbType.VarChar, 50, "MALOP");

}

public DataSet GetDataSet()

{

return GetDataSet(\_da, "tblLOPHOC");

}

// Get data by MaLop

public DataTable GetTableByMaLop(string MaLop)

{

DataTable dt = new DataTable();

using (SqlDataAdapter da = new SqlDataAdapter($"SELECT \* FROM LOPHOC WHERE MALOP = '{MaLop}'", Conn))

{

da.Fill(dt);

return dt;

}

}

// Get data by GVCN

public DataTable GetTableByGVCN(string GVCN)

{

DataTable dt = new DataTable();

using(SqlDataAdapter da = new SqlDataAdapter($"SELECT \* FROM LOPHOC WHERE GVCN = '{GVCN}'", Conn))

{

da.Fill(dt);

return dt;

}

}

public bool CheckPrimary(LopHoc lopHoc)

{

DataTable dt = DataSet.Tables["tblLOPHOC"];

string condition = $"MALOP = '{lopHoc.MaLop}'";

return CheckCondition(dt, condition);

}

public bool CheckTenLop(LopHoc lopHoc)

{

DataTable dt = DataSet.Tables["tblLOPHOC"];

string condition = $"TENLOP = '{lopHoc.TenLop}' AND MANH = '{lopHoc.MaNH}'";

return CheckCondition(dt, condition);

}

public bool CheckGVCN(LopHoc lopHoc)

{

DataTable dt = DataSet.Tables["tblLOPHOC"];

string condition = $"GVCN = '{lopHoc.Gvcn}' AND MANH = '{lopHoc.MaNH}'";

return CheckCondition(dt, condition);

}

public string Insert(LopHoc lopHoc)

{

if (CheckPrimary(lopHoc) == false)

return "Mã lớp đã tồn tại";

if (CheckTenLop(lopHoc) == false)

return "Tên lớp đã tồn tại trong cùng học kỳ";

if (CheckGVCN(lopHoc) == false)

return "GVCN đã tồn tại trong cùng học kỳ";

DataRow row = DataSet.Tables["tblLOPHOC"].NewRow();

row["MALOP"] = lopHoc.MaLop;

row["TENLOP"] = lopHoc.TenLop;

row["GVCN"] = lopHoc.Gvcn;

row["SISO"] = lopHoc.SiSo;

row["MANH"] = lopHoc.MaNH;

row["MAHK"] = lopHoc.MaHK;

using (SqlDataAdapter daGV = new SqlDataAdapter("SELECT \* FROM GIAOVIEN", Conn))

{

daGV.Fill(DataSet, "tblGIAOVIEN");

DataRow findGV = DataSet.Tables["tblGIAOVIEN"].Select($"MAGV = '{lopHoc.Gvcn}'")[0];

if (findGV != null)

row["HOTEN"] = findGV["HOTEN"];

}

DataSet.Tables["tblLOPHOC"].Rows.Add(row);

return "Thêm Thành công";

}

public string Update(LopHoc lopHoc)

{

if (CheckTenLop(lopHoc) == false)

return "Tên lớp đã tồn tại trong cùng học kỳ";

//if (CheckGVCN(lopHoc) == false)

// return "GVCN đã tồn tại trong cùng học kỳ";

DataRow row = DataSet.Tables["tblLOPHOC"].Select($"MALOP = '{lopHoc.MaLop}'")[0];

row["TENLOP"] = lopHoc.TenLop;

row["GVCN"] = lopHoc.Gvcn;

using (SqlDataAdapter daGV = new SqlDataAdapter("SELECT \* FROM GIAOVIEN", Conn))

{

daGV.Fill(DataSet, "tblGIAOVIEN");

DataRow findGV = DataSet.Tables["tblGIAOVIEN"].Select($"MAGV = '{lopHoc.Gvcn}'")[0];

if (findGV != null)

row["HOTEN"] = findGV["HOTEN"];

}

return "Sửa thành công";

}

public void Cancel()

{

DataSet.Tables["tblLOPHOC"].RejectChanges();

}

public void Save()

{

Save(\_da, "tblLOPHOC");

}

}

## 4. TaiKhoanDAL

public class TaiKhoanDAL : Connection

{

private SqlDataAdapter \_da;

public TaiKhoanDAL()

{

string strSQL = "SELECT \* FROM LOGIN\_TABLE";

\_da = new SqlDataAdapter(strSQL, Conn);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(\_da);

}

public DataSet GetDataSet()

{

return GetDataSet(\_da, "tblLOGIN\_TABLE");

}

public string Update(TaiKhoan tk)

{

DataRow row = DataSet.Tables["tblLOGIN\_TABLE"].Select($"USERNAME = '{tk.Taikhoan}'")[0];

row["PASSWORD"] = tk.Matkhau;

return "Sửa thành công";

}

public void Cancel()

{

DataSet.Tables["tblLOGIN\_TABLE"].RejectChanges();

}

public void Save()

{

Save(\_da, "tblLOGIN\_TABLE");

}

public string CheckLogin(TaiKhoan tk)

{

string user = null;

SqlDataReader dr;

try

{

SqlParameter[] para = new SqlParameter[]

{

new SqlParameter("@tenTK", tk.Taikhoan),

new SqlParameter("@matKhau", tk.Matkhau)

};

SqlCommand cmd = new SqlCommand("sp\_checkLogin\_LOGIN\_TABLE", Conn);

cmd.Parameters.AddRange(para);

cmd.CommandType = CommandType.StoredProcedure;

Conn.Open();

dr = cmd.ExecuteReader();

if (dr.HasRows)

while (dr.Read())

{

user = dr.GetString(0) + " " + dr.GetString(2);

break;

}

else

user = "Tài khoản không hợp lệ";

dr.Close();

}

catch (Exception )

{

user = "Lỗi kết nối";

}

finally

{

if(Conn.State == ConnectionState.Open)

Conn.Close();

}

return user;

}

}

## 5. GiaoVienDAL

public class GiaoVienDAL : Connection

{

private SqlDataAdapter \_da;

private SqlDataAdapter \_daTk;

public GiaoVienDAL()

{

string strSQL = "SELECT \* FROM GIAOVIEN";

string strSQLTK = "SELECT \* FROM LOGIN\_TABLE";

\_da = new SqlDataAdapter(strSQL, Conn);

\_daTk = new SqlDataAdapter(strSQLTK, Conn);

\_daTk.Fill(DataSet, "tblLOGINTABLE");

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(\_da);

SqlCommandBuilder commandBuilderTK = new SqlCommandBuilder(\_daTk);

}

public DataSet GetDataSet()

{

return GetDataSet(\_da, "tblGIAOVIEN");

}

public DataTable GetTableByMaGV(string MaGV)

{

DataTable dtGV = new DataTable();

string strSQL = $"SELECT \* FROM GIAOVIEN WHERE MAGV = '{MaGV}'";

using(SqlDataAdapter da = new SqlDataAdapter(strSQL, Conn))

{

da.Fill(dtGV);

return dtGV;

}

}

public bool CheckPrimary(GiaoVien giaoVien)

{

DataTable dt = DataSet.Tables["tblGIAOVIEN"];

string conditon = $"MAGV = '{giaoVien.MaGV}'";

return CheckCondition(dt, conditon);

}

public bool CheckEmail(GiaoVien giaoVien)

{

DataTable dt = DataSet.Tables["tblGIAOVIEN"];

string conditon = $"EMAIL = '{giaoVien.Email}'";

return CheckCondition(dt, conditon);

}

public string Insert(GiaoVien giaoVien)

{

if (CheckPrimary(giaoVien) == false)

return "Mã số đã tồn tại";

if (CheckEmail(giaoVien) == false)

return "Email đã tồn tại";

DataRow row = DataSet.Tables["tblGIAOVIEN"].NewRow();

row["MAGV"] = giaoVien.MaGV;

row["HOTEN"] = giaoVien.TenGV;

row["NGAYSINH"] = giaoVien.NgaySinh;

row["QUEQUAN"] = giaoVien.QueQuan;

row["EMAIL"] = giaoVien.Email;

row["SODT"] = giaoVien.SoDT;

row["GIOITINH"] = giaoVien.GioiTinh;

row["MATOGV"] = giaoVien.MaToGV;

row["USERNAME"] = giaoVien.Taikhoan;

row["MONGD"] = giaoVien.MonGD;

DataRow rowTK = DataSet.Tables["tblLOGINTABLE"].NewRow();

rowTK["USERNAME"] = giaoVien.Taikhoan;

rowTK["PASSWORD"] = giaoVien.Matkhau;

rowTK["QUYEN"] = "GV";

DataSet.Tables["tblLOGINTABLE"].Rows.Add(rowTK);

DataSet.Tables["tblGIAOVIEN"].Rows.Add(row);

return "Thêm Thành công";

}

public string Update(GiaoVien giaoVien)

{

DataRow row = DataSet.Tables["tblGIAOVIEN"].Select($"MAGV = '{giaoVien.MaGV}'")[0];

row["HOTEN"] = giaoVien.TenGV;

row["NGAYSINH"] = giaoVien.NgaySinh;

row["QUEQUAN"] = giaoVien.QueQuan;

row["EMAIL"] = giaoVien.Email;

row["SODT"] = giaoVien.SoDT;

row["GIOITINH"] = giaoVien.GioiTinh;

row["MATOGV"] = giaoVien.MaToGV;

row["USERNAME"] = giaoVien.Taikhoan;

row["MONGD"] = giaoVien.MonGD;

return "Sửa Thành công";

}

public void Cancel()

{

DataSet.Tables["tblGIAOVIEN"].RejectChanges();

}

public void Save()

{

Save(\_daTk, "tblLOGINTABLE");

Save(\_da, "tblGIAOVIEN");

}

}

## 6. HocSinhDAL

public class HocSinhDAL : Connection

{

private SqlDataAdapter \_da;

private SqlDataAdapter \_daTk;

private SqlDataAdapter \_daLopHoc;

private SqlDataAdapter \_daThuNgan;

private SqlDataAdapter \_daBangDiem;

private SqlDataAdapter \_daHanhKiem;

public HocSinhDAL()

{

string strSQL = "SELECT \* FROM HOCSINH";

\_da = new SqlDataAdapter(strSQL, Conn);

string strSQLTK = "SELECT \* FROM LOGIN\_TABLE";

\_daTk = new SqlDataAdapter(strSQLTK, Conn);

\_daTk.Fill(DataSet, "tblLOGINTABLE");

string strSQLLopHoc = "SELECT \* FROM LOPHOC";

\_daLopHoc = new SqlDataAdapter(strSQLLopHoc, Conn);

\_daLopHoc.Fill(DataSet, "tblLOPHOC");

string strSQLThuNgan = "SELECT \* FROM THUNGAN";

\_daThuNgan = new SqlDataAdapter(strSQLThuNgan, Conn);

\_daThuNgan.Fill(DataSet, "tblTHUNGAN");

string strSQLBangDiem = "SELECT \* FROM BANGDIEM";

\_daBangDiem = new SqlDataAdapter(strSQLBangDiem, Conn);

\_daBangDiem.Fill(DataSet, "tblBANGDIEM");

string strSQLHanhKiem = "SELECT \* FROM HANHKIEM";

\_daHanhKiem = new SqlDataAdapter(strSQLHanhKiem, Conn);

\_daHanhKiem.Fill(DataSet, "tblHANHKIEM");

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(\_da);

SqlCommandBuilder commandBuilderTK = new SqlCommandBuilder(\_daTk);

SqlCommandBuilder sqlCommandBuilderLopHoc = new SqlCommandBuilder(\_daLopHoc);

SqlCommandBuilder sqlCommandBuilderThuNgan = new SqlCommandBuilder(\_daThuNgan);

SqlCommandBuilder sqlCommandBuilderBangDiem = new SqlCommandBuilder(\_daBangDiem);

SqlCommandBuilder sqlCommandBuilderHanhKiem = new SqlCommandBuilder(\_daHanhKiem);

}

public DataSet GetDataSet()

{

return GetDataSet(\_da, "tblHOCSINH");

}

public DataSet GetDataSetByMaLop(string Malop)

{

SqlDataAdapter \_daHS\_LopHoc = new SqlDataAdapter($"SELECT \* FROM HOCSINH WHERE MALOP = '{Malop}'", Conn);

return GetDataSet(\_daHS\_LopHoc, "tblHOCSINH\_ByMaLop");

}

public bool CheckPrimary(HocSinh hs)

{

DataTable dt = DataSet.Tables["tblHOCSINH"];

string condition = $"MAHS = '{hs.MaHS}'";

return CheckCondition(dt, condition);

}

public string Insert(HocSinh hs)

{

if (this.CheckPrimary(hs) == false)

return "Mã học sinh đã tồn tại";

DataRow dr = DataSet.Tables["tblHOCSINH"].NewRow();

dr["MAHS"] = hs.MaHS;

dr["HOTEN"] = hs.HoTen;

dr["NGAYSINH"] = hs.NgaySinh;

dr["GIOITINH"] = hs.GioiTinh;

dr["DIACHI"] = hs.DiaChi;

dr["SODT"] = hs.SoDT;

dr["MALOP"] = hs.MaLop;

dr["GIOITINH"] = hs.GioiTinh;

dr["USERNAME"] = hs.Taikhoan;

DataSet.Tables["tblHOCSINH"].Rows.Add(dr);

// Cập nhật sĩ số Lớp học

DataRow rowsLop = DataSet.Tables["tblLOPHOC"].Select($"MALOP = '{hs.MaLop}'")[0];

string MaNH = rowsLop["MANH"].ToString();

rowsLop["SISO"] = int.Parse(rowsLop["SISO"].ToString()) + 1;

// Tự động tạo bảng Thu ngân

// Học Kì 1

DataRow rowTNHK1 = DataSet.Tables["tblTHUNGAN"].NewRow();

rowTNHK1["MANH"] = MaNH;

rowTNHK1["MAHK"] = "1";

rowTNHK1["MAHS"] = hs.MaHS;

rowTNHK1["SOTIEN\_HP"] = 0;

rowTNHK1["TINHTRANG\_HP"] = "Chưa hoàn thành";

rowTNHK1["NGAYDONG\_HP"] = DBNull.Value;

rowTNHK1["SOTIEN\_BHYT"] = 0;

rowTNHK1["TINHTRANG\_BHYT"] = "Chưa hoàn thành";

rowTNHK1["NGAYDONG\_BHYT"] = DBNull.Value; ;

DataSet.Tables["tblTHUNGAN"].Rows.Add(rowTNHK1);

// Học Kỳ 2

DataRow rowTNHK2 = DataSet.Tables["tblTHUNGAN"].NewRow();

rowTNHK2["MANH"] = MaNH;

rowTNHK2["MAHK"] = "2";

rowTNHK2["MAHS"] = hs.MaHS;

rowTNHK2["SOTIEN\_HP"] = 0;

rowTNHK2["TINHTRANG\_HP"] = "Chưa hoàn thành";

rowTNHK2["NGAYDONG\_HP"] = DBNull.Value;

rowTNHK2["SOTIEN\_BHYT"] = 0;

rowTNHK2["TINHTRANG\_BHYT"] = "Chưa hoàn thành";

rowTNHK2["NGAYDONG\_BHYT"] = DBNull.Value;

DataSet.Tables["tblTHUNGAN"].Rows.Add(rowTNHK2);

// Tự động tạo bảng Bảng điểm

// Học kì 1

DataRow rowBDHK1 = DataSet.Tables["tblBANGDIEM"].NewRow();

rowBDHK1["MANH"] = MaNH;

rowBDHK1["MAHK"] = "1";

rowBDHK1["MAHS"] = hs.MaHS;

rowBDHK1["TOAN"] = 0;

rowBDHK1["NGUVAN"] = 0;

rowBDHK1["NGOAINGU"] = 0;

rowBDHK1["VATLY"] = 0;

rowBDHK1["HOAHOC"] = 0;

rowBDHK1["SINHHOC"] = 0;

rowBDHK1["LICHSU"] = 0;

rowBDHK1["DIALY"] = 0;

rowBDHK1["GDCD"] = 0;

rowBDHK1["DTB"] = 0;

DataSet.Tables["tblBANGDIEM"].Rows.Add(rowBDHK1);

// Học kì 2

DataRow rowBDHK2 = DataSet.Tables["tblBANGDIEM"].NewRow();

rowBDHK2["MANH"] = MaNH;

rowBDHK2["MAHK"] = "2";

rowBDHK2["MAHS"] = hs.MaHS;

rowBDHK2["TOAN"] = 0;

rowBDHK2["NGUVAN"] = 0;

rowBDHK2["NGOAINGU"] = 0;

rowBDHK2["VATLY"] = 0;

rowBDHK2["HOAHOC"] = 0;

rowBDHK2["SINHHOC"] = 0;

rowBDHK2["LICHSU"] = 0;

rowBDHK2["DIALY"] = 0;

rowBDHK2["GDCD"] = 0;

rowBDHK2["DTB"] = 0;

DataSet.Tables["tblBANGDIEM"].Rows.Add(rowBDHK2);

// Tự động tạo bảng Hạnh kiểm

// Học kì 1

DataRow rowHKHK1 = DataSet.Tables["tblHANHKIEM"].NewRow();

rowHKHK1["MANH"] = MaNH;

rowHKHK1["MAHK"] = "1";

rowHKHK1["MAHS"] = hs.MaHS;

rowHKHK1["LOAIHK"] = "";

DataSet.Tables["tblHANHKIEM"].Rows.Add(rowHKHK1);

// Học kì 2

DataRow rowHKHK2 = DataSet.Tables["tblHANHKIEM"].NewRow();

rowHKHK2["MANH"] = MaNH;

rowHKHK2["MAHK"] = "2";

rowHKHK2["MAHS"] = hs.MaHS;

rowHKHK2["LOAIHK"] = "";

DataSet.Tables["tblHANHKIEM"].Rows.Add(rowHKHK2);

// Tự động tạo Tài khoản

DataRow rowTK = DataSet.Tables["tblLOGINTABLE"].NewRow();

rowTK["USERNAME"] = hs.Taikhoan;

rowTK["PASSWORD"] = hs.Matkhau;

rowTK["QUYEN"] = "HS";

DataSet.Tables["tblLOGINTABLE"].Rows.Add(rowTK);

return "Thêm thành công";

}

public string Update(HocSinh hs)

{

DataRow dr = DataSet.Tables["tblHOCSINH"].Select($"MAHS = '{hs.MaHS}'")[0];

string MaLop\_Old = dr["MALOP"].ToString(); // Malop hiện tại

DataRow rowNH\_Old = DataSet.Tables["tblLOPHOC"].Select($"MALOP = '{MaLop\_Old}'")[0];

string MaNH\_Old = rowNH\_Old["MANH"].ToString(); // MaNH hiện tại

DataRow rowNH\_New = DataSet.Tables["tblLOPHOC"].Select($"MALOP = '{hs.MaLop}'")[0];

string MaNH\_New = rowNH\_New["MANH"].ToString(); // MaNH mới

// Cập nhật khi chuyển lớp trong cùng năm học

if (MaNH\_Old != MaNH\_New)

return "Không thể chuyển lớp khác năm học";

else

{

// Cập nhật thông tin bình thường

dr["HOTEN"] = hs.HoTen;

dr["NGAYSINH"] = hs.NgaySinh;

dr["GIOITINH"] = hs.GioiTinh;

dr["DIACHI"] = hs.DiaChi;

dr["SODT"] = hs.SoDT;

if (MaLop\_Old != hs.MaLop)

{

// Cập nhật sĩ số lớp cũ

DataRow rowOldClass = DataSet.Tables["tblLOPHOC"].Select($"MALOP = '{MaLop\_Old}'")[0];

rowOldClass["SISO"] = int.Parse(rowOldClass["SISO"].ToString()) - 1;

// Cập nhật sĩ số lớp mới

DataRow rowNewClass = DataSet.Tables["tblLOPHOC"].Select($"MALOP = '{hs.MaLop}'")[0];

rowNewClass["SISO"] = int.Parse(rowNewClass["SISO"].ToString()) + 1;

}

dr["MALOP"] = hs.MaLop; // MaLop mới

}

return "Sửa thành công";

}

public string Delete(string MaHS)

{

DataRow dr = DataSet.Tables["tblHOCSINH"].Select($"MAHS = '{MaHS}'")[0];

string MaLop = dr["MALOP"].ToString();

// Get Mã năm học

string MaNH = DataSet.Tables["tblLOPHOC"].Select($"MALOP = '{MaLop}'")[0]["MANH"].ToString();

// Xóa học sinh trong bảng thu ngân

DataRow[] rowsTN = DataSet.Tables["tblTHUNGAN"].Select($"MAHS = '{MaHS}' AND MANH = '{MaNH}' AND MAHK IN (1,2)");

foreach (DataRow rowTN in rowsTN)

rowTN.Delete();

// Xóa học sinh trong bảng bảng điểm

DataRow[] rowsBD = DataSet.Tables["tblBANGDIEM"].Select($"MAHS = '{MaHS}' AND MANH = '{MaNH}' AND MAHK IN (1,2)");

foreach (DataRow rowBD in rowsBD)

rowBD.Delete();

// Xóa học sinh trong bảng hạnh kiểm

DataRow[] rowsHK = DataSet.Tables["tblHANHKIEM"].Select($"MAHS = '{MaHS}' AND MANH = '{MaNH}' AND MAHK IN (1,2)");

foreach(DataRow rowHK in rowsHK)

rowHK.Delete();

// Xóa học sinh trong bảng tài khoản

DataRow rowTK = DataSet.Tables["tblLOGINTABLE"].Select($"USERNAME = '{MaHS}'")[0];

rowTK.Delete();

// Chỉnh sửa số lương học sinh trong lớp

DataRow rowLop = DataSet.Tables["tblLOPHOC"].Select($"MALOP = '{MaLop}'")[0];

rowLop["SISO"] = int.Parse(rowLop["SISO"].ToString()) - 1;

dr.Delete();

Save(\_daLopHoc, "tblLOPHOC");

Save(\_daThuNgan, "tblTHUNGAN");

Save(\_daHanhKiem, "tblHANHKIEM");

Save(\_daBangDiem, "tblBANGDIEM");

Save(\_da, "tblHOCSINH");

Save(\_daTk, "tblLOGINTABLE");

return "Xóa thành công";

}

public void Cancel()

{

DataSet.Tables["tblHOCSINH"].RejectChanges();

}

public void Save()

{

Save(\_daLopHoc, "tblLOPHOC");

Save(\_daTk, "tblLOGINTABLE");

Save(\_da, "tblHOCSINH");

Save(\_daThuNgan, "tblTHUNGAN");

Save(\_daBangDiem, "tblBANGDIEM");

Save(\_daHanhKiem, "tblHANHKIEM");

}

}

## 7. BangDiemDAL

public class BangDiemDAL : Connection

{

private SqlDataAdapter \_da;

private LopHoc \_lopHoc;

public LopHoc LopHoc { get => \_lopHoc; set => \_lopHoc = value; }

public BangDiemDAL(LopHoc lopHoc)

{

this.LopHoc = lopHoc;

string strSQL = $"SELECT bd.MAHS, hs.HOTEN, bd.MAHK, bd.TOAN, bd.NGUVAN, bd.NGOAINGU, bd.VATLY, bd.HOAHOC, bd.SINHHOC, bd.DIALY, bd.LICHSU, bd.GDCD, bd.DTB FROM BANGDIEM bd JOIN HOCSINH hs ON bd.MAHS = hs.MAHS WHERE bd.MANH = '{lopHoc.MaNH}' AND hs.MALOP = '{lopHoc.MaLop}'";

\_da = new SqlDataAdapter(strSQL, Conn);

// Define the InsertCommand

\_da.InsertCommand = new SqlCommand(

"INSERT INTO BANGDIEM (MAHS, MAHK, MANH, TOAN, NGUVAN, NGOAINGU, VATLY, HOAHOC, SINHHOC, DIALY, LICHSU, GDCD, DTB) " +

"VALUES (@MAHS, @MAHK, @MANH, @TOAN, @NGUVAN, @NGOAINGU, @VATLY, @HOAHOC, @SINHHOC, @DIALY, @LICHSU, @GDCD, @DTB)", Conn);

\_da.InsertCommand.Parameters.Add("@MAHS", SqlDbType.VarChar, 50, "MAHS");

\_da.InsertCommand.Parameters.Add("@MAHK", SqlDbType.VarChar, 50, "MAHK");

\_da.InsertCommand.Parameters.Add("@MANH", SqlDbType.VarChar, 50, "MANH");

\_da.InsertCommand.Parameters.Add("@TOAN", SqlDbType.Decimal, 10, "TOAN");

\_da.InsertCommand.Parameters.Add("@NGUVAN", SqlDbType.Decimal, 10, "NGUVAN");

\_da.InsertCommand.Parameters.Add("@NGOAINGU", SqlDbType.Decimal, 10, "NGOAINGU");

\_da.InsertCommand.Parameters.Add("@VATLY", SqlDbType.Decimal, 10, "VATLY");

\_da.InsertCommand.Parameters.Add("@HOAHOC", SqlDbType.Decimal, 10, "HOAHOC");

\_da.InsertCommand.Parameters.Add("@SINHHOC", SqlDbType.Decimal, 10, "SINHHOC");

\_da.InsertCommand.Parameters.Add("@DIALY", SqlDbType.Decimal, 10, "DIALY");

\_da.InsertCommand.Parameters.Add("@LICHSU", SqlDbType.Decimal, 10, "LICHSU");

\_da.InsertCommand.Parameters.Add("@GDCD", SqlDbType.Decimal, 10, "GDCD");

\_da.InsertCommand.Parameters.Add("@DTB", SqlDbType.Decimal, 10, "DTB");

// Define the UpdateCommand

\_da.UpdateCommand = new SqlCommand(

"UPDATE BANGDIEM SET TOAN = @TOAN, NGUVAN = @NGUVAN, NGOAINGU = @NGOAINGU, VATLY = @VATLY, HOAHOC = @HOAHOC, SINHHOC = @SINHHOC, DIALY = @DIALY, LICHSU = @LICHSU, GDCD = @GDCD, DTB = @DTB " +

"WHERE MAHS = @MAHS AND MAHK = @MAHK AND MANH = @MANH", Conn);

\_da.UpdateCommand.Parameters.Add("@TOAN", SqlDbType.Decimal, 10, "TOAN");

\_da.UpdateCommand.Parameters.Add("@NGUVAN", SqlDbType.Decimal, 10, "NGUVAN");

\_da.UpdateCommand.Parameters.Add("@NGOAINGU", SqlDbType.Decimal, 10, "NGOAINGU");

\_da.UpdateCommand.Parameters.Add("@VATLY", SqlDbType.Decimal, 10, "VATLY");

\_da.UpdateCommand.Parameters.Add("@HOAHOC", SqlDbType.Decimal, 0, "HOAHOC");

\_da.UpdateCommand.Parameters.Add("@SINHHOC", SqlDbType.Decimal, 10, "SINHHOC");

\_da.UpdateCommand.Parameters.Add("@DIALY", SqlDbType.Decimal, 10, "DIALY");

\_da.UpdateCommand.Parameters.Add("@LICHSU", SqlDbType.Decimal, 10, "LICHSU");

\_da.UpdateCommand.Parameters.Add("@GDCD", SqlDbType.Decimal, 10, "GDCD");

\_da.UpdateCommand.Parameters.Add("@DTB", SqlDbType.Decimal , 10, "DTB");

\_da.UpdateCommand.Parameters.Add("@MAHS", SqlDbType.VarChar, 50, "MAHS");

\_da.UpdateCommand.Parameters.Add("@MAHK", SqlDbType.VarChar, 50, "MAHK");

\_da.UpdateCommand.Parameters.Add("@MANH", SqlDbType.VarChar, 50, "MANH");

// Define the DeleteCommand

\_da.DeleteCommand = new SqlCommand(

"DELETE FROM BANGDIEM WHERE MAHS = @MAHS AND MAHK = @MAHK AND MANH = @MANH", Conn);

\_da.DeleteCommand.Parameters.Add("@MAHS", SqlDbType.VarChar, 50, "MAHS");

\_da.DeleteCommand.Parameters.Add("@MAHK", SqlDbType.VarChar, 50, "MAHK");

\_da.DeleteCommand.Parameters.Add("@MANH", SqlDbType.VarChar, 50, "MANH");

}

public DataSet GetDataSet()

{

return GetDataSet(\_da, "tblBANGDIEM");

}

public DataSet GetDataSet(string maHK)

{

string strSQL = $"SELECT bd.MANH, bd.MAHS, hs.HOTEN, bd.MAHK, bd.TOAN, bd.NGUVAN, bd.NGOAINGU, bd.VATLY, bd.HOAHOC, bd.SINHHOC, bd.DIALY, bd.LICHSU, bd.GDCD, bd.DTB FROM BANGDIEM bd JOIN HOCSINH hs ON bd.MAHS = hs.MAHS WHERE bd.MANH = '{this.LopHoc.MaNH}' AND hs.MALOP = '{this.LopHoc.MaLop}' AND bd.MAHK = '{maHK}'";

\_da.SelectCommand = new SqlCommand(strSQL, Conn);

return GetDataSet(\_da, "tblBANGDIEM\_MaHK");

}

public string Update(BangDiem bd)

{

if(!DataSet.Tables.Contains("tblBANGDIEM\_MaHK"))

return "Hãy chọn học kì để nhập điểm";

DataRow row = DataSet.Tables["tblBANGDIEM\_MaHK"].Select($"MAHS = '{bd.MaHS}' AND MAHK = '{bd.MaHK}' AND MANH = '{bd.MaNH}'")[0];

row["TOAN"] = Math.Round(decimal.Parse(bd.Toan), 2);

row["NGUVAN"] = Math.Round(decimal.Parse(bd.NgVan), 2);

row["NGOAINGU"] = Math.Round(decimal.Parse(bd.NgNgu), 2);

row["VATLY"] = Math.Round(decimal.Parse(bd.VatLy), 2);

row["HOAHOC"] = Math.Round(decimal.Parse(bd.HoaHoc), 2);

row["SINHHOC"] = Math.Round(decimal.Parse(bd.SinhHoc), 2);

row["DIALY"] = Math.Round(decimal.Parse(bd.DiaLy), 2);

row["LICHSU"] = Math.Round(decimal.Parse(bd.LichSu), 2);

row["GDCD"] = Math.Round(decimal.Parse(bd.Gdcd), 2);

bd.TinhDTB();

row["DTB"] = Math.Round(decimal.Parse(bd.Dtb), 2);

return "Nhập điểm thành công";

}

public void Cancel()

{

if (DataSet.Tables.Contains("tblBANGDIEM"))

DataSet.Tables["tblBANGDIEM"].RejectChanges();

if (DataSet.Tables.Contains("tblBANGDIEM\_MaHK"))

DataSet.Tables["tblBANGDIEM\_MaHK"].RejectChanges();

}

public void Save()

{

if(DataSet.Tables.Contains("tblBANGDIEM"))

Save(\_da, "tblBANGDIEM");

if (DataSet.Tables.Contains("tblBANGDIEM\_MaHK"))

Save(\_da, "tblBANGDIEM\_MaHK");

}

}

## 8. HanhKiemDAL

public class HanhKiemDAL : Connection

{

private SqlDataAdapter \_da;

private LopHoc \_lopHoc;

public LopHoc LopHoc { get => \_lopHoc; set => \_lopHoc = value; }

public HanhKiemDAL(LopHoc lopHoc)

{

this.LopHoc = lopHoc;

string strSQL = $"SELECT hk.MAHK, hs.MAHS, hs.HOTEN, hk.LOAIHK FROM HANHKIEM hk JOIN HOCSINH hs ON hk.MAHS = hs.MAHS WHERE hk.MANH = '{this.LopHoc.MaNH}' AND hs.MALOP = '{this.LopHoc.MaLop}'";

\_da = new SqlDataAdapter(strSQL, Conn);

// InsertCommand

\_da.InsertCommand = new SqlCommand("INSERT INTO HANHKIEM (MAHK, MANH, MAHS, LOAIHK) VALUES (@MAHK, @MANH, @MAHS, @LOAIHK)", Conn);

\_da.InsertCommand.Parameters.Add("@MAHK", SqlDbType.VarChar, 50, "MAHK");

\_da.InsertCommand.Parameters.Add("@MANH", SqlDbType.VarChar, 50, "MANH");

\_da.InsertCommand.Parameters.Add("@MAHS", SqlDbType.VarChar, 50, "MAHS");

\_da.InsertCommand.Parameters.Add("@LOAIHK", SqlDbType.NVarChar, 50, "LOAIHK");

// UpdateCommand

\_da.UpdateCommand = new SqlCommand("UPDATE HANHKIEM SET LOAIHK = @LOAIHK WHERE MAHK = @MAHK AND MANH = @MANH AND MAHS = @MAHS", Conn);

\_da.UpdateCommand.Parameters.Add("@LOAIHK", SqlDbType.NVarChar, 50, "LOAIHK");

\_da.UpdateCommand.Parameters.Add("@MAHK", SqlDbType.VarChar, 50, "MAHK");

\_da.UpdateCommand.Parameters.Add("@MANH", SqlDbType.VarChar, 50, "MANH");

\_da.UpdateCommand.Parameters.Add("@MAHS", SqlDbType.VarChar, 50, "MAHS");

// DeleteCommand

\_da.DeleteCommand = new SqlCommand("DELETE FROM HANHKIEM WHERE MAHK = @MAHK AND MANH = @MANH AND MAHS = @MAHS", Conn);

\_da.DeleteCommand.Parameters.Add("@MAHK", SqlDbType.VarChar, 50, "MAHK");

\_da.DeleteCommand.Parameters.Add("@MANH", SqlDbType.VarChar, 50, "MANH");

\_da.DeleteCommand.Parameters.Add("@MAHS", SqlDbType.VarChar, 50, "MAHS");

}

public DataSet GetDataSet()

{

return GetDataSet(\_da, "tblHANHKIEM");

}

public DataSet GetDataSet(string maHK)

{

string strSQL = $"SELECT hk.MAHK, hk.MANH, hs.MAHS, hs.HOTEN, hk.LOAIHK FROM HANHKIEM hk JOIN HOCSINH hs ON hk.MAHS = hs.MAHS WHERE hk.MANH = '{this.LopHoc.MaNH}' AND hs.MALOP = '{this.LopHoc.MaLop}' AND hk.MAHK = '{maHK}'";

\_da.SelectCommand = new SqlCommand(strSQL, Conn);

return GetDataSet(\_da, "tblHANHKIEM\_MaHK");

}

public string Update(HanhKiem hk)

{

if(!DataSet.Tables.Contains("tblHANHKIEM\_MaHK"))

return "Hãy chọn học kì để đánh giá";

DataRow dr = DataSet.Tables["tblHANHKIEM\_MaHK"].Select($"MAHK = '{hk.MaHK}' AND MANH = '{hk.MaNH}' AND MAHS = '{hk.MaHS}'")[0];

dr["LOAIHK"] = hk.LoaiHK;

return "Đánh giá thành công";

}

public void Cancel()

{

if(DataSet.Tables.Contains("tblHANHKIEM"))

DataSet.Tables["tblHANHKIEM"].RejectChanges();

if (DataSet.Tables.Contains("tblHANHKIEM\_MaHK"))

DataSet.Tables["tblHANHKIEM\_MaHK"].RejectChanges();

}

public void Save()

{

if(DataSet.Tables.Contains("tblHANHKIEM"))

Save(\_da, "tblHANHKIEM");

if (DataSet.Tables.Contains("tblHANHKIEM\_MaHK"))

Save(\_da, "tblHANHKIEM\_MaHK");

}

}

## 9. ToGVDAL

public class ToGVDAL : Connection

{

private SqlDataAdapter \_da;

public ToGVDAL()

{

string strSQL = "SELECT toGV.\*, gv.HOTEN FROM TOGV toGV, GIAOVIEN gv WHERE toGV.TRUONGTOGV = gv.MAGV";

\_da = new SqlDataAdapter(strSQL, Conn);

// InsertCommand

\_da.InsertCommand = new SqlCommand("INSERT INTO TOGV (MATOGV, TRUONGTOGV, TENTOGV) VALUES (@MATOGV, @TRUONGTOGV, @TENTOGV)", Conn);

\_da.InsertCommand.Parameters.Add("@MATOGV", SqlDbType.Char, 5, "MATOGV");

\_da.InsertCommand.Parameters.Add("@TRUONGTOGV", SqlDbType.VarChar, 50, "TRUONGTOGV");

\_da.InsertCommand.Parameters.Add("@TENTOGV", SqlDbType.NVarChar, 50, "TENTOGV");

// UpdateCommand

\_da.UpdateCommand = new SqlCommand("UPDATE TOGV SET TRUONGTOGV = @TRUONGTOGV, TENTOGV = @TENTOGV WHERE MATOGV = @MATOGV", Conn);

\_da.UpdateCommand.Parameters.Add("@MATOGV", SqlDbType.Char, 5, "MATOGV");

\_da.UpdateCommand.Parameters.Add("@TRUONGTOGV", SqlDbType.VarChar, 50, "TRUONGTOGV");

\_da.UpdateCommand.Parameters.Add("@TENTOGV", SqlDbType.NVarChar, 50, "TENTOGV");

// DeleteCommand

\_da.DeleteCommand = new SqlCommand("DELETE FROM TOGV WHERE MATOGV = @MATOGV", Conn);

\_da.DeleteCommand.Parameters.Add("@MATOGV", SqlDbType.VarChar, 50, "MATOGV");

}

public DataSet GetDataSet()

{

return GetDataSet(\_da, "tblTOGV");

}

public bool CheckPrimary(ToGV toGV)

{

DataTable dt = DataSet.Tables["tblTOGV"];

string condition = $"MATOGV = '{toGV.MaToGV}'";

return CheckCondition(dt, condition);

}

public bool CheckTruongToGV(ToGV toGV)

{

DataTable dt = DataSet.Tables["tblTOGV"];

string condition = $"TRUONGTOGV = '{toGV.TruongToGV}'";

return CheckCondition(dt, condition);

}

public string Insert(ToGV toGV)

{

if (CheckPrimary(toGV) == false)

return "Mã tổ Đã tồn tại";

if (CheckTruongToGV(toGV) == false)

return "Giáo viên này đã là Trưởng tổ";

DataRow row = DataSet.Tables["tblTOGV"].NewRow();

row["MATOGV"] = toGV.MaToGV;

row["TRUONGTOGV"] = toGV.TruongToGV;

row["TENTOGV"] = toGV.TenToGV;

using (GetDataSet(new SqlDataAdapter("SELECT \* FROM GIAOVIEN", Conn), "tblGIAOVIEN"))

{

row["HOTEN"] = DataSet.Tables["tblGIAOVIEN"].Select($"MAGV = '{toGV.TruongToGV}'")[0]["HOTEN"];

}

DataSet.Tables["tblTOGV"].Rows.Add(row);

return "Thêm thành công";

}

public string Update(ToGV toGV)

{

DataRow row = DataSet.Tables["tblTOGV"].Select($"MATOGV = '{toGV.MaToGV}'")[0];

row["MATOGV"] = toGV.MaToGV;

row["TRUONGTOGV"] = toGV.TruongToGV;

row["TENTOGV"] = toGV.TenToGV;

using (GetDataSet(new SqlDataAdapter("SELECT \* FROM GIAOVIEN", Conn), "tblGIAOVIEN"))

{

row["HOTEN"] = DataSet.Tables["tblGIAOVIEN"].Select($"MAGV = '{toGV.TruongToGV}'")[0]["HOTEN"];

}

return "Sửa Thành công";

}

public void Cancel()

{

DataSet.Tables["tblTOGV"].RejectChanges();

}

public void Save()

{

Save(\_da, "tblTOGV");

}

}

## 10. ThuNganDAL

public class ThuNganDAL : Connection

{

private SqlDataAdapter \_da;

public ThuNganDAL()

{

string strSQL = "SELECT tn.MANH, tn.MAHK, tn.MAHS, hs.HOTEN, tn.SOTIEN\_HP, tn.TINHTRANG\_HP, tn.NGAYDONG\_HP, tn.SOTIEN\_BHYT, tn.TINHTRANG\_BHYT, tn.NGAYDONG\_BHYT FROM THUNGAN tn JOIN HOCSINH hs ON tn.MAHS = hs.MAHS";

\_da = new SqlDataAdapter(strSQL, Conn);

// InsertCommand

\_da.InsertCommand = new SqlCommand("INSERT INTO THUNGAN (MANH, MAHK, MAHS, SOTIEN\_HP, TINHTRANG\_HP, NGAYDONG\_HP, SOTIEN\_BHYT, TINHTRANG\_BHYT, NGAYDONG\_BHYT) VALUES (@MANH, @MAHK, @MAHS, @SOTIEN\_HP, @TINHTRANG\_HP, @NGAYDONG\_HP, @SOTIEN\_BHYT, @TINHTRANG\_BHYT, @NGAYDONG\_BHYT)", Conn);

\_da.InsertCommand.Parameters.Add("@MANH", SqlDbType.Char, 5, "MANH");

\_da.InsertCommand.Parameters.Add("@MAHK", SqlDbType.Char, 5, "MAHK");

\_da.InsertCommand.Parameters.Add("@MAHS", SqlDbType.Char, 5, "MAHS");

\_da.InsertCommand.Parameters.Add("@SOTIEN\_HP", SqlDbType.Decimal, 18, "SOTIEN\_HP");

\_da.InsertCommand.Parameters.Add("@TINHTRANG\_HP", SqlDbType.NVarChar, 50, "TINHTRANG\_HP");

\_da.InsertCommand.Parameters.Add("@NGAYDONG\_HP", SqlDbType.DateTime, 8, "NGAYDONG\_HP");

\_da.InsertCommand.Parameters.Add("@SOTIEN\_BHYT", SqlDbType.Decimal, 18, "SOTIEN\_BHYT");

\_da.InsertCommand.Parameters.Add("@TINHTRANG\_BHYT", SqlDbType.NVarChar, 50, "TINHTRANG\_BHYT");

\_da.InsertCommand.Parameters.Add("@NGAYDONG\_BHYT", SqlDbType.DateTime, 8, "NGAYDONG\_BHYT");

// UpdateCommand

\_da.UpdateCommand = new SqlCommand("UPDATE THUNGAN SET SOTIEN\_HP = @SOTIEN\_HP, TINHTRANG\_HP = @TINHTRANG\_HP, NGAYDONG\_HP = @NGAYDONG\_HP, SOTIEN\_BHYT = @SOTIEN\_BHYT, TINHTRANG\_BHYT = @TINHTRANG\_BHYT, NGAYDONG\_BHYT = @NGAYDONG\_BHYT WHERE MANH = @MANH AND MAHK = @MAHK AND MAHS = @MAHS", Conn);

\_da.UpdateCommand.Parameters.Add("@SOTIEN\_HP", SqlDbType.Decimal, 18, "SOTIEN\_HP");

\_da.UpdateCommand.Parameters.Add("@TINHTRANG\_HP", SqlDbType.NVarChar, 50, "TINHTRANG\_HP");

\_da.UpdateCommand.Parameters.Add("@NGAYDONG\_HP", SqlDbType.DateTime, 8, "NGAYDONG\_HP");

\_da.UpdateCommand.Parameters.Add("@SOTIEN\_BHYT", SqlDbType.Decimal, 18, "SOTIEN\_BHYT");

\_da.UpdateCommand.Parameters.Add("@TINHTRANG\_BHYT", SqlDbType.NVarChar, 50, "TINHTRANG\_BHYT");

\_da.UpdateCommand.Parameters.Add("@NGAYDONG\_BHYT", SqlDbType.DateTime, 8, "NGAYDONG\_BHYT");

\_da.UpdateCommand.Parameters.Add("@MANH", SqlDbType.Char, 5, "MANH");

\_da.UpdateCommand.Parameters.Add("@MAHK", SqlDbType.Char, 5, "MAHK");

\_da.UpdateCommand.Parameters.Add("@MAHS", SqlDbType.Char, 5, "MAHS");

// DeleteCommand

\_da.DeleteCommand = new SqlCommand("DELETE FROM THUNGAN WHERE MANH = @MANH AND MAHK = @MAHK AND MAHS = @MAHS", Conn);

\_da.DeleteCommand.Parameters.Add("@MANH", SqlDbType.Char, 5, "MANH");

\_da.DeleteCommand.Parameters.Add("@MAHK", SqlDbType.Char, 5, "MAHK");

\_da.DeleteCommand.Parameters.Add("@MAHS", SqlDbType.Char, 5, "MAHS");

}

public DataSet GetDataSet()

{

return GetDataSet(\_da, "tblTHUNGAN");

}

public DataSet GetDataSetByNamHoc\_HocKy(NamHoc nh)

{

string strSQL = $"SELECT tn.MANH, tn.MAHK, tn.MAHS, hs.HOTEN, tn.SOTIEN\_HP, tn.TINHTRANG\_HP, tn.NGAYDONG\_HP, tn.SOTIEN\_BHYT, tn.TINHTRANG\_BHYT, tn.NGAYDONG\_BHYT FROM THUNGAN tn JOIN HOCSINH hs ON tn.MAHS = hs.MAHS WHERE tn.MANH = '{nh.MaNH}' AND tn.MAHK = '{nh.MaHK}'";

\_da.SelectCommand = new SqlCommand(strSQL, Conn);

return GetDataSet(\_da, "tblTHUNGAN\_NH");

}

public string Update(ThuNgan tn)

{

if(DataSet.Tables.Contains("tblTHUNGAN\_NH"))

{

DataRow drTN\_NH = DataSet.Tables["tblTHUNGAN\_NH"].Select($"MANH = '{tn.MaNH}' AND MAHK = '{tn.MaHK}' AND MAHS = '{tn.MaHS}'")[0];

drTN\_NH["SOTIEN\_HP"] = tn.TienHP;

drTN\_NH["TINHTRANG\_HP"] = tn.TinhTrangHP;

drTN\_NH["NGAYDONG\_HP"] = tn.NgayDongHP;

drTN\_NH["SOTIEN\_BHYT"] = tn.TienBHYT;

drTN\_NH["TINHTRANG\_BHYT"] = tn.TinTrangBHYT;

drTN\_NH["NGAYDONG\_BHYT"] = tn.NgayDongBHYT;

}

DataRow dr = DataSet.Tables["tblTHUNGAN"].Select($"MANH = '{tn.MaNH}' AND MAHK = '{tn.MaHK}' AND MAHS = '{tn.MaHS}'")[0];

dr["SOTIEN\_HP"] = tn.TienHP;

dr["TINHTRANG\_HP"] = tn.TinhTrangHP;

dr["NGAYDONG\_HP"] = tn.NgayDongHP;

dr["SOTIEN\_BHYT"] = tn.TienBHYT;

dr["TINHTRANG\_BHYT"] = tn.TinTrangBHYT;

dr["NGAYDONG\_BHYT"] = tn.NgayDongBHYT;

return "Cập nhật thành công";

}

public void Cancel()

{

DataSet.Tables["tblTHUNGAN"].RejectChanges();

}

public void Save()

{

Save(\_da, "tblTHUNGAN");

if (DataSet.Tables.Contains("tblTHUNGAN\_NH"))

{

Save(\_da, "tblTHUNGAN\_NH");

DataSet.Tables["tblTHUNGAN\_NH"].Dispose();

}

}

}

# Phần VI: Business Logic Layer - Lớp nghiệp vụ

Lớp nghiệp vụ dùng để xử lý các nghiệp vụ của ứng dụng, thực hiện các quy tắc logic lên dữ liệu.

Lớp này chứa các phương thức xử lý logic, tính toán dữ liệu. BLL là lớp giao tiếp giữa lớp giao diện (Presentation Layer) và lớp dữ liệu (DAL).

## 1. NamHocBLL

public class NamHocBLL

{

NamHocDAL namHocDAL = new NamHocDAL();

public DataTable GetData()

{

return namHocDAL.GetDataSet().Tables["tblNAMHOC"];

}

public DataTable GetDataMaNH()

{

return namHocDAL.GetDataMaNH();

}

public bool CheckInput(NamHoc namHoc)

{

if(namHoc.MaNH==""||namHoc.MaNH.Length!=5||namHoc.MaHK == null)

return false;

return true;

}

public string Insert(NamHoc namHoc)

{

if (this.CheckInput(namHoc) == false)

return "Nhập thông tin không hợp lệ";

return namHocDAL.Insert(namHoc);

}

public void Save()

{

namHocDAL.Save();

}

}

## 2. LopHocBLL

public class LopHocBLL

{

LopHocDAL lopHocDAL = new LopHocDAL();

public DataTable GetData()

{

return lopHocDAL.GetDataSet().Tables["tblLOPHOC"];

}

// get classes by GVCV

public DataTable GetDataByGVCV(string GVCN)

{

return lopHocDAL.GetTableByGVCN(GVCN);

}

// get info by MaLop

public DataTable GetDataByMaLop(string MaLop)

{

return lopHocDAL.GetTableByMaLop(MaLop);

}

public bool CheckInput(LopHoc lopHoc)

{

if (lopHoc.MaLop == "" || lopHoc.MaLop.Length != 5 || lopHoc.TenLop == "" || lopHoc.MaNH == null || lopHoc.Gvcn == null)

return false;

return true;

}

public string Insert(LopHoc lopHoc)

{

if (this.CheckInput(lopHoc) == false)

return "Nhập thông tin Lớp học không hợp lệ";

return lopHocDAL.Insert(lopHoc);

}

public string Update(LopHoc lopHoc)

{

if (this.CheckInput(lopHoc) == false)

return "Nhập thông tin Lớp học không hợp lệ";

return lopHocDAL.Update(lopHoc);

}

public void Cancel()

{

lopHocDAL.Cancel();

}

public void Save()

{

lopHocDAL.Save();

}

}

## 3. TaiKhoanBLL

public class TaiKhoanBLL

{

TaiKhoanDAL taiKhoanDAL = new TaiKhoanDAL();

public TaiKhoanBLL() { }

public DataTable GetData()

{

return taiKhoanDAL.GetDataSet().Tables["tblLOGIN\_TABLE"];

}

public bool CheckInput(TaiKhoan tk)

{

if(tk.Matkhau == "" || tk.Matkhau.Length >15)

return false;

return true;

}

public bool CheckQuyenADMIN(TaiKhoan tk)

{

if (tk.Quyen == "AD")

return false;

return true;

}

public string Update(TaiKhoan tk)

{

if (this.CheckQuyenADMIN(tk) == false)

return "Không thể sửa quyền ADMIN";

if (this.CheckInput(tk) == false)

return "Mật khẩu không hợp lệ";

return taiKhoanDAL.Update(tk);

}

public void Destroy()

{

taiKhoanDAL.Cancel();

}

public void Save()

{

taiKhoanDAL.Save();

}

public string CheckLogin(TaiKhoan tk)

{

if(tk.Taikhoan == "" || tk.Matkhau == "")

return "Tài khoản không hợp lệ";

string user = taiKhoanDAL.CheckLogin(tk);

return user;

}

}

## 4. GiaoVienBLL

public class GiaoVienBLL

{

GiaoVienDAL giaoVienDAL = new GiaoVienDAL();

public DataTable GetData()

{

return giaoVienDAL.GetDataSet().Tables["tblGIAOVIEN"];

}

public DataTable GetData(string MaGV)

{

return giaoVienDAL.GetTableByMaGV(MaGV);

}

public bool CheckInput(GiaoVien giaoVien)

{

if (giaoVien.MaGV == "" || giaoVien.MaGV.Length != 5 || giaoVien.TenGV == "" || giaoVien.SoDT == "" || giaoVien.SoDT.Length != 10 || giaoVien.MonGD == "" || giaoVien.Email == "" || giaoVien.QueQuan == "" || giaoVien.GioiTinh == null)

return false;

return true;

}

public string Insert(GiaoVien giaoVien)

{

if(this.CheckInput(giaoVien) == false)

return "Nhập thông tin Giáo viên không hợp lệ";

return giaoVienDAL.Insert(giaoVien);

}

public string Update(GiaoVien giaoVien)

{

if (this.CheckInput(giaoVien) == false)

return "Nhập thông tin Giáo viên không hợp lệ";

return giaoVienDAL.Update(giaoVien);

}

public void Cancel()

{

giaoVienDAL.Cancel();

}

public void Save()

{

giaoVienDAL.Save();

}

}

## 5. HocSinhBLL

public class HocSinhBLL

{

HocSinhDAL hocSinhDAL = new HocSinhDAL();

public HocSinhBLL() { }

public DataTable GetData()

{

return hocSinhDAL.GetDataSet().Tables["tblHOCSINH"];

}

public DataTable GetDataByMaLop(string MaLop)

{

return hocSinhDAL.GetDataSetByMaLop(MaLop).Tables["tblHOCSINH\_ByMaLop"];

}

public bool CheckInput(HocSinh hs)

{

if (hs.MaHS == "" || hs.MaHS.Length != 5 || hs.HoTen == "" || hs.GioiTinh == null || hs.DiaChi == "" || hs.SoDT == "" || hs.SoDT.Length != 10 || hs.MaLop == null)

return false;

return true;

}

public string Insert(HocSinh hs)

{

if (this.CheckInput(hs) == false)

return "Nhập thông tin không hợp lệ";

return hocSinhDAL.Insert(hs);

}

public string Update(HocSinh hs)

{

if(this.CheckInput(hs) == false)

return "Nhập thông tin không hợp lệ";

return hocSinhDAL.Update(hs);

}

public string Delete(string MaHS)

{

return hocSinhDAL.Delete(MaHS);

}

public void Cancel()

{

hocSinhDAL.Cancel();

}

public void Save()

{

hocSinhDAL.Save();

}

}

## 6. BangDiemBLL

public class BangDiemBLL

{

BangDiemDAL bangDiemDAL;

public DataTable GetDataByMaLop(LopHoc lopHoc)

{

bangDiemDAL = new BangDiemDAL(lopHoc);

return bangDiemDAL.GetDataSet().Tables["tblBANGDIEM"];

}

// fillter by HK

public DataTable GetDataByMaLop(LopHoc lopHoc, string maHK)

{

bangDiemDAL = new BangDiemDAL(lopHoc);

if(maHK == "Cả năm")

{

if (bangDiemDAL.DataSet.Tables.Contains("tblBANGDIEM"))

bangDiemDAL.DataSet.Tables["tblBANGDIEM"].Clear();

return bangDiemDAL.GetDataSet().Tables["tblBANGDIEM"];

}

if (bangDiemDAL.DataSet.Tables.Contains("tblBANGDIEM\_MaHK"))

bangDiemDAL.DataSet.Tables["tblBANGDIEM\_MaHK"].Clear();

return bangDiemDAL.GetDataSet(maHK).Tables["tblBANGDIEM\_MaHK"];

}

public bool CheckInput(BangDiem bd)

{

if (bd.Toan == "" || bd.NgVan == "" || bd.NgNgu == "" || bd.VatLy == "" || bd.HoaHoc == "" || bd.SinhHoc == "" || bd.DiaLy == "" || bd.LichSu == "" || bd.Gdcd == "")

return false;

if (double.Parse(bd.Toan) < 0 || double.Parse(bd.Toan) > 10 ||

double.Parse(bd.NgVan) < 0 || double.Parse(bd.NgVan) > 10 ||

double.Parse(bd.NgNgu) < 0 || double.Parse(bd.NgNgu) > 10 ||

double.Parse(bd.NgVan) < 0 || double.Parse(bd.NgVan) > 10 ||

double.Parse(bd.VatLy) < 0 || double.Parse(bd.VatLy) > 10 ||

double.Parse(bd.HoaHoc) < 0 || double.Parse(bd.HoaHoc) > 10 ||

double.Parse(bd.SinhHoc) < 0 ||double.Parse(bd.SinhHoc) > 10 ||

double.Parse(bd.DiaLy) < 0 || double.Parse(bd.DiaLy) > 10 ||

double.Parse(bd.LichSu) < 0 || double.Parse(bd.LichSu) > 10 ||

double.Parse(bd.NgVan) < 0 || double.Parse(bd.NgVan) > 10 )

return false;

return true;

}

public string Update(BangDiem bd)

{

if(bd.MaHK == "Cả năm")

return "Hãy chọn học kì để nhập điểm";

if (this.CheckInput(bd) == false)

return "Nhập điểm không hợp lệ";

return bangDiemDAL.Update(bd);

}

public void Save()

{

bangDiemDAL.Save();

}

public void Cancel()

{

bangDiemDAL.Cancel();

}

}

## 7. HanhKiemBLL

public class HanhKiemBLL

{

HanhKiemDAL hanhKiemDAL;

public HanhKiemBLL() { }

public DataTable GetDataByMaLop(LopHoc lopHoc)

{

hanhKiemDAL = new HanhKiemDAL(lopHoc);

return hanhKiemDAL.GetDataSet().Tables["tblHANHKIEM"];

}

public DataTable GetDataByMaLop(LopHoc lopHoc, string maHK)

{

hanhKiemDAL = new HanhKiemDAL(lopHoc);

if(maHK == "Cả năm")

{

if (hanhKiemDAL.DataSet.Tables.Contains("tblHANHKIEM"))

hanhKiemDAL.DataSet.Tables["tblHANHKIEM"].Clear();

return hanhKiemDAL.GetDataSet().Tables["tblHANHKIEM"];

}

if (hanhKiemDAL.DataSet.Tables.Contains("tblHANHKIEM\_MaHK"))

hanhKiemDAL.DataSet.Tables["tblHANHKIEM\_MaHK"].Clear();

return hanhKiemDAL.GetDataSet(maHK).Tables["tblHANHKIEM\_MaHK"];

}

public string Update(HanhKiem hanhKiem)

{

if(hanhKiem.MaHK == "Cả năm")

return "Hãy chọn học kì để đánh giá";

if(hanhKiem.LoaiHK == null)

return "Hãy chọn loại hạnh kiểm";

return hanhKiemDAL.Update(hanhKiem);

}

public void Cancel()

{

hanhKiemDAL.Cancel();

}

public void Save()

{

hanhKiemDAL.Save();

}

}

## 8. ToGVBLL

public class ToGVBLL

{

ToGVDAL ToGVDAL = new ToGVDAL();

public ToGVBLL() { }

public DataTable GetData()

{

return ToGVDAL.GetDataSet().Tables["tblTOGV"];

}

public bool CheckInput(ToGV toGV)

{

if(toGV.TenToGV == ""||toGV.MaToGV ==""||toGV.MaToGV.Length!= 5)

return false;

return true;

}

public string Insert(ToGV toGV)

{

if (this.CheckInput(toGV) == false)

return "Nhập thông tin Tổ Giáo viên không chính xác";

return ToGVDAL.Insert(toGV);

}

public string Update(ToGV toGV)

{

return ToGVDAL.Update(toGV);

}

public void Cancel()

{

ToGVDAL.Cancel();

}

public void Save()

{

ToGVDAL.Save();

}

}

## 9. ThuNganBLL

public class ThuNganBLL

{

ThuNganDAL thuNganDAL = new ThuNganDAL();

public ThuNganBLL() { }

public DataTable GetData()

{

return thuNganDAL.GetDataSet().Tables["tblTHUNGAN"];

}

public DataTable GetDataByNamHoc\_HocKy(NamHoc namHoc)

{

if (thuNganDAL.DataSet.Tables.Contains("tblTHUNGAN\_NH"))

thuNganDAL.DataSet.Tables["tblTHUNGAN\_NH"].Clear();

return thuNganDAL.GetDataSetByNamHoc\_HocKy(namHoc).Tables["tblTHUNGAN\_NH"];

}

public string Update(ThuNgan tn)

{

return thuNganDAL.Update(tn);

}

public void Cancel()

{

thuNganDAL.Cancel();

}

public void Save()

{

thuNganDAL.Save();

}

}

# Phần VII: Presentation Layer – Lớp giao diện

## 1. frmLogin

public partial class frmLogin : Form

{

TaiKhoan taiKhoan = new TaiKhoan();

TaiKhoanBLL taiKhoanBLL = new TaiKhoanBLL();

public frmLogin()

{

InitializeComponent();

}

private void frmLogin\_FormClosing(object sender, FormClosingEventArgs e)

{

if (DialogResult.OK == MessageBox.Show("Bạn muốn thoát hệ thống", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Question))

e.Cancel = false;

else

e.Cancel = true;

}

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

{

taiKhoan.Taikhoan = this.txtUsername.Text;

taiKhoan.Matkhau = this.txtPassword.Text;

this.txtUsername.Text = string.Empty;

this.txtPassword.Text = string.Empty;

this.txtUsername.Focus();

string user = taiKhoanBLL.CheckLogin(taiKhoan);

if (user == "Lỗi kết nối")

{

MessageBox.Show("Lỗi kết nối cơ sở dữ liệu", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (user == "Tài khoản không hợp lệ")

{

MessageBox.Show("Tài khoản không hợp lệ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string[] DangNhap = user.Split(' ');

if (DangNhap[1] == "AD")

{

this.Hide();

frmAdmin frmAdmin = new frmAdmin();

frmAdmin.ShowDialog();

this.Show();

}

else if (DangNhap[1] == "GV")

{

this.Hide();

frmThongTinGV frmThongTin = new frmThongTinGV(DangNhap[0]);

frmThongTin.ShowDialog();

this.Show();

}

}

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

{

this.Close();

}

}

## 2. frmAdmin

public partial class frmAdmin : Form

{

private Form \_currentForm = null;

public frmAdmin()

{

InitializeComponent();

this.SwitchFrom(new frmQuanLyNH());

}

public void SwitchFrom(Form childForm)

{

if (this.\_currentForm != null)

this.\_currentForm.Close();

// Đặt thuộc tính cho childForm

childForm.TopLevel = false;

childForm.Dock = DockStyle.Fill;

// Thêm childForm vào pnContainer

pnContainer.Controls.Clear();

pnContainer.Controls.Add(childForm);

childForm.Show();

this.\_currentForm = childForm;

}

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

{

this.SwitchFrom(new frmQuanLyHS());

}

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

{

this.SwitchFrom(new frmQuanLyNH());

}

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

{

this.SwitchFrom(new frmQuanLyTK());

}

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

{

this.SwitchFrom(new frmQuanLyLH());

}

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

{

this.SwitchFrom(new frmQuanLyToGV());

}

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

{

this.SwitchFrom(new frmQuanLyGV());

}

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

{

this.SwitchFrom(new frmQuanLyTN());

}

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

{

this.Close();

}

}

## 3. frmQuanLyNH

public partial class frmQuanLyNH : Form

{

NamHocBLL namHocBLL = new NamHocBLL();

LopHocBLL lopHocBLL = new LopHocBLL();

public frmQuanLyNH()

{

InitializeComponent();

// this.cbHocKy

string[] hk = new string[2] { "1", "2" };

this.cbHocKy.Items.AddRange(hk);

// Danh sách Năm học

DataTable dtNH = namHocBLL.GetData();

this.dgvDanhSachNH.DataSource = dtNH;

this.dgvDanhSachNH.Columns["MaNH"].HeaderText = "Năm học";

this.dgvDanhSachNH.Columns["MaHK"].HeaderText = "Học kỳ";

this.txtNamHoc.Focus();

this.cbHocKy.SelectedIndex = -1;

}

public void InsertNH()

{

NamHoc namHoc = new NamHoc();

namHoc.MaNH = this.txtNamHoc.Text.ToUpper();

namHoc.MaHK = this.cbHocKy.SelectedItem?.ToString();

string resultNH = namHocBLL.Insert(namHoc);

MessageBox.Show(resultNH, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.txtNamHoc.Text = string.Empty;

this.cbHocKy.SelectedIndex = -1;

}

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

{

if (flagNH == true)

{

this.InsertNH();

flagNH = false;

}

}

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

{

if (DialogResult.Yes == MessageBox.Show("Bạn có muốn lưu?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question))

{

namHocBLL.Save();

this.dgvDanhSachNH.ClearSelection();

}

}

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

{

if (this.dgvDanhSachNH.SelectedRows.Count > 0)

{

this.txtNamHoc.ReadOnly = true;

this.txtNamHoc.BackColor = Color.LightGray;

DataGridViewRow row = this.dgvDanhSachNH.SelectedRows[0];

this.txtNamHoc.Text = row.Cells[0].Value.ToString();

if (row.Cells[1].Value.ToString() == "1")

this.cbHocKy.SelectedIndex = 0;

else

this.cbHocKy.SelectedIndex = 1;

}

}

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

{

this.txtNamHoc.ReadOnly = false;

this.dgvDanhSachNH.ClearSelection();

this.txtNamHoc.BackColor = Color.White;

}

}

## 4. frmQuanLyLH

public partial class frmQuanLyLH : Form

{

NamHocBLL namHocBLL = new NamHocBLL();

LopHocBLL lopHocBLL = new LopHocBLL();

GiaoVienBLL giaoVienBLL = new GiaoVienBLL();

public frmQuanLyLH()

{

InitializeComponent();

// this.dgvLopHoc

DataTable dtLop = lopHocBLL.GetData();

this.dgvLopHoc.DataSource = dtLop;

this.dgvLopHoc.Columns["MALOP"].HeaderText = "Mã Lớp";

this.dgvLopHoc.Columns["TENLOP"].HeaderText = "Tên Lớp";

this.dgvLopHoc.Columns["SISO"].HeaderText = "Sĩ Số";

this.dgvLopHoc.Columns["GVCN"].HeaderText = "GVCN";

this.dgvLopHoc.Columns["MANH"].HeaderText = "Mã Năm học";

this.dgvLopHoc.Columns["HOTEN"].HeaderText = "Tên GVCN";

this.dgvLopHoc.Columns["MAHK"].Visible = false;

// this.cbGVCN

DataTable dtGVCN = giaoVienBLL.GetData();

this.cbGVCN.DataSource = dtGVCN;

this.cbGVCN.DisplayMember = "HOTEN";

this.cbGVCN.ValueMember = "MAGV";

// this.cbMaNH

this.cbMaNH.DataSource = namHocBLL.GetDataMaNH();

this.cbMaNH.DisplayMember = "MaNH";

this.cbMaNH.ValueMember = "MaNH";

this.cbMaNH.Focus();

this.cbMaNH.SelectedIndex = -1;

this.cbGVCN.SelectedIndex = -1;

}

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

{

if (this.dgvLopHoc.SelectedRows.Count > 0)

{

this.txtMaLop.ReadOnly = true;

this.txtMaLop.BackColor = Color.LightGray;

DataGridViewRow row = this.dgvLopHoc.SelectedRows[0];

this.txtMaLop.Text = row.Cells["MALOP"].Value.ToString();

this.txtTenLop.Text = row.Cells["TENLOP"].Value.ToString();

this.cbMaNH.SelectedValue = row.Cells["MANH"].Value.ToString();

this.cbGVCN.SelectedValue = row.Cells["GVCN"].Value.ToString();

}

}

private void InsertLopHoc()

{

LopHoc lopHoc = new LopHoc();

lopHoc.MaLop = this.txtMaLop.Text.ToUpper();

lopHoc.TenLop = this.txtTenLop.Text;

lopHoc.Gvcn = this.cbGVCN.SelectedValue?.ToString();

lopHoc.SiSo = "0";

lopHoc.MaNH = this.cbMaNH.SelectedValue?.ToString();

lopHoc.MaHK = "1"; // Default

string resultLopHoc = lopHocBLL.Insert(lopHoc);

MessageBox.Show(resultLopHoc, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.txtMaLop.Text = string.Empty;

this.txtTenLop.Text = string.Empty;

this.cbMaNH.SelectedIndex = -1;

this.cbGVCN.SelectedIndex = -1;

}

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

{

InsertLopHoc();

}

private void UpdateLopHoc()

{

LopHoc lopHoc = new LopHoc();

lopHoc.MaLop = this.txtMaLop.Text.ToUpper();

lopHoc.TenLop = this.txtTenLop.Text;

lopHoc.Gvcn = this.cbGVCN.SelectedValue?.ToString();

lopHoc.MaNH = this.cbMaNH.SelectedValue?.ToString();

string resultLopHoc = lopHocBLL.Update(lopHoc);

MessageBox.Show(resultLopHoc, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.txtMaLop.Text = string.Empty;

this.txtTenLop.Text = string.Empty;

this.cbMaNH.SelectedIndex = -1;

this.cbGVCN.SelectedIndex = -1;

}

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

{

if (this.dgvLopHoc.SelectedRows.Count > 0)

UpdateLopHoc();

}

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

{

if (DialogResult.Yes == MessageBox.Show("Bạn có muốn lưu thay đổi?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question))

{

lopHocBLL.Save();

this.dgvLopHoc.ClearSelection();

}

}

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

{

lopHocBLL.Cancel();

this.dgvLopHoc.ClearSelection();

this.txtMaLop.ReadOnly = false;

this.txtMaLop.BackColor = Color.White;

this.txtMaLop.Focus();

this.txtMaLop.Text = string.Empty;

this.txtTenLop.Text = string.Empty;

this.cbMaNH.SelectedIndex = -1;

this.cbGVCN.SelectedIndex = -1;

}

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

{

this.dgvLopHoc.ClearSelection();

this.txtMaLop.ReadOnly = false;

this.txtMaLop.BackColor = Color.White;

}

}

## 5. frmQuanLyToGV

public partial class frmQuanLyToGV : Form

{

ToGVBLL toGVBLL = new ToGVBLL();

GiaoVienBLL giaoVienBLL = new GiaoVienBLL();

DataTable dtGV = new DataTable();

public frmQuanLyToGV()

{

InitializeComponent();

// dgvDanhSachToGV

this.dgvDanhSachToGV.DataSource = toGVBLL.GetData();

this.dgvDanhSachToGV.Columns["MaToGv"].HeaderText = "Mã tổ";

this.dgvDanhSachToGV.Columns["TenToGv"].HeaderText = "Tên tổ";

this.dgvDanhSachToGV.Columns["TruongToGV"].Visible = false;

this.dgvDanhSachToGV.Columns["HOTEN"].HeaderText = "Tên Trưởng tổ";

// cbTruongTo

dtGV = giaoVienBLL.GetData();

this.cbTruongTo.DataSource = dtGV;

this.cbTruongTo.DisplayMember = "HOTEN";

this.cbTruongTo.ValueMember = "MAGV";

this.cbTruongTo.SelectedIndex = -1;

}

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

{

if (this.dgvDanhSachToGV.SelectedRows.Count > 0)

{

this.txtMaToGV.ReadOnly = true;

this.txtMaToGV.BackColor = Color.LightGray;

DataGridViewRow row = this.dgvDanhSachToGV.SelectedRows[0];

this.txtMaToGV.Text = row.Cells["MATOGV"].Value.ToString();

this.txtTenTo.Text = row.Cells["TENTOGV"].Value.ToString();

DataRow[] findrow = dtGV.Select($"MAGV = '{row.Cells["TRUONGTOGV"].Value.ToString()}'");

if (findrow.Length > 0)

this.cbTruongTo.SelectedValue = findrow[0]["MAGV"];

}

}

public void InsertToGV()

{

ToGV toGV = new ToGV();

toGV.MaToGV = this.txtMaToGV.Text.ToUpper();

toGV.TenToGV = this.txtTenTo.Text;

toGV.TruongToGV = this.cbTruongTo.SelectedValue?.ToString();

string resultToGV = toGVBLL.Insert(toGV);

MessageBox.Show(resultToGV, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.txtMaToGV.Text = string.Empty;

this.txtTenTo.Text = string.Empty;

this.cbTruongTo.SelectedIndex = -1;

}

public void Update\_ToGV()

{

ToGV toGV = new ToGV();

toGV.MaToGV = this.txtMaToGV.Text;

toGV.TenToGV = this.txtTenTo.Text;

toGV.TruongToGV = this.cbTruongTo.SelectedValue?.ToString();

string resultUpadte = toGVBLL.Update(toGV);

MessageBox.Show(resultUpadte, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

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

{

this.InsertToGV();

}

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

{

if(this.dgvDanhSachToGV.SelectedRows.Count > 0)

{

this.Update\_ToGV();

}

}

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

{

toGVBLL.Cancel();

this.txtMaToGV.ReadOnly = false;

this.txtMaToGV.BackColor = Color.White;

this.dgvDanhSachToGV.ClearSelection();

this.txtMaToGV.Focus();

this.txtMaToGV.Text = string.Empty;

this.txtTenTo.Text = string.Empty;

this.cbTruongTo.SelectedIndex = -1;

}

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

{

if (DialogResult.Yes == MessageBox.Show("Bạn có muốn lưu thay đổi", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question))

{

toGVBLL.Save();

this.txtMaToGV.ReadOnly = false;

this.txtMaToGV.BackColor = Color.White;

this.dgvDanhSachToGV.ClearSelection();

this.txtMaToGV.Focus();

this.txtMaToGV.Text = string.Empty;

this.txtTenTo.Text = string.Empty;

this.cbTruongTo.SelectedIndex = -1;

}

}

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

{

this.txtMaToGV.ReadOnly = false;

this.txtMaToGV.BackColor = Color.White;

this.dgvDanhSachToGV.ClearSelection();

}

}

## 6. frmQuanLyGV

public partial class frmQuanLyGV : Form

{

ToGVBLL toGVBLL = new ToGVBLL();

GiaoVienBLL giaoVienBLL = new GiaoVienBLL();

DataTable dtGV;

DataTable dtToGV;

public frmQuanLyGV()

{

InitializeComponent();

dtGV = giaoVienBLL.GetData();

dtToGV = toGVBLL.GetData();

// cbToGV

this.cbToGV.DataSource = dtToGV;

this.cbToGV.DisplayMember = "TENTOGV";

this.cbToGV.ValueMember = "MATOGV";

this.cbToGV.SelectedIndex = -1;

// dgvDanhSach

this.dgvDanhSach.DataSource = dtGV;

this.dgvDanhSach.Columns["MAGV"].HeaderText = "Mã GV";

this.dgvDanhSach.Columns["HOTEN"].HeaderText = "Họ tên";

this.dgvDanhSach.Columns["NGAYSINH"].HeaderText = "Ngày sinh";

this.dgvDanhSach.Columns["QUEQUAN"].HeaderText = "Quê quán";

this.dgvDanhSach.Columns["SODT"].HeaderText = "Số ĐT";

this.dgvDanhSach.Columns["GIOITINH"].HeaderText = "Giới tính";

this.dgvDanhSach.Columns["MONGD"].HeaderText = "Môn GD";

this.dgvDanhSach.Columns["MATOGV"].HeaderText = "Mã tổ";

this.dgvDanhSach.Columns["USERNAME"].HeaderText = "Tài khoản";

this.cbGioiTinh.SelectedIndex = -1;

}

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

{

if (this.dgvDanhSach.SelectedRows.Count > 0)

{

this.txtMaSo.ReadOnly = true;

this.txtMaSo.BackColor = Color.LightGray;

DataGridViewRow row = this.dgvDanhSach.SelectedRows[0];

this.txtMaSo.Text = row.Cells["MAGV"].Value.ToString();

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

this.txtQueQuan.Text = row.Cells["QUEQUAN"].Value.ToString();

this.mtxtSoDT.Text = row.Cells["SODT"].Value.ToString();

this.txtEmail.Text = row.Cells["EMAIL"].Value.ToString();

this.txtMonGD.Text = row.Cells["MONGD"].Value.ToString();

this.dtpNgaySinh.Text = row.Cells["NGAYSINH"].Value.ToString();

if (row.Cells["GIOITINH"].Value.ToString() == "Nam")

cbGioiTinh.SelectedIndex = 0;

else

cbGioiTinh.SelectedIndex = 1;

DataRow[] findrow = dtToGV.Select($"MATOGV = '{row.Cells["MATOGV"].Value.ToString()}'");

if (findrow.Length > 0)

this.cbToGV.SelectedValue = findrow[0]["MATOGV"];

}

}

public void InsertGV()

{

GiaoVien gv = new GiaoVien();

gv.MaGV = this.txtMaSo.Text.ToUpper();

gv.TenGV = this.txtHoTen.Text;

gv.QueQuan = this.txtQueQuan.Text;

gv.SoDT = this.mtxtSoDT.Text;

gv.MaToGV = this.cbToGV.SelectedValue?.ToString();

gv.NgaySinh = this.dtpNgaySinh.Value.ToShortDateString();

gv.GioiTinh = this.cbGioiTinh.SelectedItem?.ToString();

gv.Email = this.txtEmail.Text;

gv.MonGD = this.txtMonGD.Text;

gv.Taikhoan = gv.MaGV;

gv.Matkhau = gv.MaGV.ToLower();

string resultGV = giaoVienBLL.Insert(gv); // Insert GiaoVien

MessageBox.Show(resultGV, "Thông báo",MessageBoxButtons.OK, MessageBoxIcon.Information);

this.txtMaSo.Text = string.Empty;

this.txtHoTen.Text = string.Empty;

this.txtQueQuan.Text = string.Empty;

this.mtxtSoDT.Text = string.Empty;

this.cbToGV.SelectedIndex = -1;

this.cbGioiTinh.SelectedIndex = -1;

this.txtEmail.Text = string.Empty;

this.txtMonGD.Text = string.Empty;

}

public void Update\_GV()

{

GiaoVien gv = new GiaoVien();

gv.MaGV = this.txtMaSo.Text;

gv.TenGV = this.txtHoTen.Text;

gv.QueQuan = this.txtQueQuan.Text;

gv.SoDT = this.mtxtSoDT.Text;

gv.MaToGV = this.cbToGV.SelectedValue?.ToString();

gv.NgaySinh = this.dtpNgaySinh.Value.ToShortDateString();

gv.GioiTinh = this.cbGioiTinh.SelectedItem?.ToString();

gv.Email = this.txtEmail.Text;

gv.MonGD = this.txtMonGD.Text;

gv.Taikhoan = gv.MaGV;

string resultUpadte = giaoVienBLL.Update(gv);

MessageBox.Show(resultUpadte, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

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

{

this.InsertGV();

}

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

{

if (this.dgvDanhSach.SelectedRows.Count > 0)

{

this.Update\_GV();

}

}

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

{

giaoVienBLL.Cancel();

this.dgvDanhSach.ClearSelection();

this.txtMaSo.ReadOnly = false;

this.txtMaSo.BackColor = Color.White;

this.txtMaSo.Focus();

this.txtMaSo.Text = string.Empty;

this.txtHoTen.Text = string.Empty;

this.txtQueQuan.Text = string.Empty;

this.mtxtSoDT.Text = string.Empty;

this.cbToGV.SelectedIndex = -1;

this.cbGioiTinh.SelectedIndex = -1;

this.txtEmail.Text = string.Empty;

this.txtMonGD.Text = string.Empty;

}

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

{

if (DialogResult.Yes == MessageBox.Show("Bạn có muốn lưu?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question))

{

giaoVienBLL.Save();

this.dgvDanhSach.ClearSelection();

this.txtMaSo.ReadOnly = false;

this.txtMaSo.BackColor = Color.White;

this.txtMaSo.Text = string.Empty;

this.txtHoTen.Text = string.Empty;

this.txtQueQuan.Text = string.Empty;

this.mtxtSoDT.Text = string.Empty;

this.cbToGV.SelectedIndex = -1;

this.cbGioiTinh.SelectedIndex = -1;

this.txtEmail.Text = string.Empty;

this.txtMonGD.Text = string.Empty;

}

}

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

{

if(this.txtTimKiem.Text != "")

{

DataRow[] findrow = dtGV.Select($"HOTEN = '{this.txtTimKiem.Text}'");

if (findrow.Length > 0)

{

this.dgvDanhSach.ClearSelection();

this.dgvDanhSach.Rows[dtGV.Rows.IndexOf(findrow[0])].Selected = true;

this.dgvDanhSach\_CellContentClick(null, null);

this.txtTimKiem.Clear();

}

else

{

MessageBox.Show("Không tìm thấy", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.txtTimKiem.Clear();

}

}

}

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

{

this.dgvDanhSach.ClearSelection();

this.txtMaSo.ReadOnly = false;

this.txtMaSo.BackColor = Color.White;

}

}

## 7. frmQuanLyHS

public partial class frmQuanLyHS : Form

{

HocSinhBLL hocSinhBLL = new HocSinhBLL();

LopHocBLL lopHocBLL = new LopHocBLL();

DataTable dtHS = new DataTable();

DataTable dtLH = new DataTable();

public frmQuanLyHS()

{

InitializeComponent();

// Load data

dtHS = hocSinhBLL.GetData();

this.dgvDanhSach.DataSource = dtHS;

this.dgvDanhSach.Columns["MAHS"].HeaderText = "Mã học sinh";

this.dgvDanhSach.Columns["HOTEN"].HeaderText = "Họ tên";

this.dgvDanhSach.Columns["NGAYSINH"].HeaderText = "Ngày sinh";

this.dgvDanhSach.Columns["DIACHI"].HeaderText = "Địa chỉ";

this.dgvDanhSach.Columns["SODT"].HeaderText = "Số điện thoại";

this.dgvDanhSach.Columns["GIOITINH"].HeaderText = "Giới tính";

this.dgvDanhSach.Columns["MALOP"].HeaderText = "Mã lớp";

this.dgvDanhSach.Columns["USERNAME"].HeaderText = "Tài khoản";

// Set up combobox

dtLH = lopHocBLL.GetData();

dtLH.Columns.Add("MALOP\_MANH", typeof(string), "TENLOP + ' - ' + MANH");

this.cbMaLop.DataSource = dtLH;

this.cbMaLop.DisplayMember = "MALOP\_MANH"; // "TENLOP

this.cbMaLop.ValueMember = "MALOP";

this.cbMaLop.SelectedIndex = -1;

this.cbGioiTinh.SelectedIndex = -1;

}

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

{

if(this.dgvDanhSach.SelectedRows.Count>0)

{

this.txtMaSo.ReadOnly = true;

this.txtMaSo.BackColor = Color.LightGray;

this.txtMaSo.Text = this.dgvDanhSach.SelectedRows[0].Cells["MAHS"].Value.ToString();

this.txtHoTen.Text = this.dgvDanhSach.SelectedRows[0].Cells["HOTEN"].Value.ToString();

this.dtpNgaySinh.Text = this.dgvDanhSach.SelectedRows[0].Cells["NGAYSINH"].Value.ToString();

this.txtDiaChi.Text = this.dgvDanhSach.SelectedRows[0].Cells["DIACHI"].Value.ToString();

this.mtxtSoDT.Text = this.dgvDanhSach.SelectedRows[0].Cells["SODT"].Value.ToString();

this.cbMaLop.SelectedValue = this.dgvDanhSach.SelectedRows[0].Cells["MALOP"].Value.ToString();

this.cbGioiTinh.SelectedItem = this.dgvDanhSach.SelectedRows[0].Cells["GIOITINH"].Value.ToString();

}

}

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

{

HocSinh hs = new HocSinh();

hs.MaHS = this.txtMaSo.Text.ToUpper();

hs.HoTen = this.txtHoTen.Text;

hs.DiaChi = this.txtDiaChi.Text;

hs.SoDT = this.mtxtSoDT.Text;

hs.NgaySinh = this.dtpNgaySinh.Value.ToShortDateString();

hs.MaLop = this.cbMaLop.SelectedValue?.ToString();

hs.GioiTinh = this.cbGioiTinh.SelectedItem?.ToString();

hs.Taikhoan = this.txtMaSo.Text.ToUpper();

hs.Matkhau = this.txtMaSo.Text.ToLower();

string result = hocSinhBLL.Insert(hs);

MessageBox.Show(result, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.txtMaSo.Focus();

this.txtMaSo.Text = string.Empty;

this.txtHoTen.Text = string.Empty;

this.txtDiaChi.Text = string.Empty;

this.mtxtSoDT.Text = string.Empty;

this.cbMaLop.SelectedIndex = -1;

this.cbGioiTinh.SelectedIndex = -1;

}

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

{

if(this.dgvDanhSach.SelectedRows.Count >0)

{

HocSinh hs = new HocSinh();

hs.MaHS = this.txtMaSo.Text.ToUpper();

hs.HoTen = this.txtHoTen.Text;

hs.DiaChi = this.txtDiaChi.Text;

hs.SoDT = this.mtxtSoDT.Text;

hs.NgaySinh = this.dtpNgaySinh.Value.ToShortDateString();

hs.MaLop = this.cbMaLop.SelectedValue?.ToString();

hs.GioiTinh = this.cbGioiTinh.SelectedItem?.ToString();

hs.Taikhoan = this.txtMaSo.Text.ToUpper();

hs.Matkhau = this.txtMaSo.Text.ToLower();

string result = hocSinhBLL.Update(hs);

MessageBox.Show(result, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

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

{

if(this.dgvDanhSach.SelectedRows.Count > 0)

{

if (DialogResult.Yes == MessageBox.Show("Bạn chắc có muốn xóa.\n Việc xóa không thể hoàn tác lại", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Warning))

{

string result = hocSinhBLL.Delete(this.txtMaSo.Text);

this.txtMaSo.Text = string.Empty;

this.txtHoTen.Text = string.Empty;

this.txtDiaChi.Text = string.Empty;

this.mtxtSoDT.Text = string.Empty;

this.cbMaLop.SelectedIndex = -1;

this.cbGioiTinh.SelectedIndex = -1;

}

}

}

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

{

hocSinhBLL.Cancel();

this.txtMaSo.Focus();

this.txtMaSo.ReadOnly = false;

this.txtMaSo.BackColor = Color.White;

this.dgvDanhSach.ClearSelection();

this.txtMaSo.Text = string.Empty;

this.txtHoTen.Text = string.Empty;

this.txtDiaChi.Text = string.Empty;

this.mtxtSoDT.Text = string.Empty;

this.cbMaLop.SelectedIndex = -1;

this.cbGioiTinh.SelectedIndex = -1;

}

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

{

if (DialogResult.Yes == MessageBox.Show("Bạn có muốn lưu?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question))

{

this.txtMaSo.Focus();

this.txtMaSo.ReadOnly = false;

this.txtMaSo.BackColor = Color.White;

this.dgvDanhSach.ClearSelection();

hocSinhBLL.Save();

this.txtMaSo.Text = string.Empty;

this.txtHoTen.Text = string.Empty;

this.txtDiaChi.Text = string.Empty;

this.mtxtSoDT.Text = string.Empty;

this.cbMaLop.SelectedIndex = -1;

this.cbGioiTinh.SelectedIndex = -1;

}

}

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

{

this.txtMaSo.ReadOnly = false;

this.txtMaSo.BackColor = Color.White;

this.dgvDanhSach.ClearSelection();

}

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

{

if (this.txtTimKiem.Text != "")

{

DataRow[] findRows = dtHS.Select($"HOTEN = '{this.txtTimKiem.Text}'");

if (findRows.Length > 0)

{

DataRow findRow = findRows[0];

this.dgvDanhSach.ClearSelection();

this.dgvDanhSach.Rows[dtHS.Rows.IndexOf(findRow)].Selected = true;

this.dgvDanhSach\_CellContentClick(null, null);

this.txtTimKiem.Clear();

}

else

{

MessageBox.Show("Không tìm thấy học sinh", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.txtTimKiem.Clear();

}

}

}

}

## 8. frmQuanLyTN

public partial class frmQuanLyTN : Form

{

ThuNganBLL thuNganBLL = new ThuNganBLL();

NamHocBLL namHocBLL = new NamHocBLL();

DataTable dtTN;

DataTable dtNH;

public frmQuanLyTN()

{

InitializeComponent();

// Load cbNamHoc

dtNH = namHocBLL.GetDataMaNH();

this.cbMaNH.DataSource = dtNH;

this.cbMaNH.DisplayMember = "MaNH";

this.cbMaNH.ValueMember = "MaNH";

this.cbMaNH.SelectedIndex = -1;

this.cbHocKy.SelectedIndex = -1;

// Load Data

dtTN = thuNganBLL.GetData();

this.dgvDanhSachTN.DataSource = dtTN;

this.dgvDanhSachTN.Columns["MANH"].HeaderText = "Năm học";

this.dgvDanhSachTN.Columns["MAHK"].HeaderText = "Học kỳ";

this.dgvDanhSachTN.Columns["MAHS"].HeaderText = "Mã HS";

this.dgvDanhSachTN.Columns["HOTEN"].HeaderText = "Họ tên";

this.dgvDanhSachTN.Columns["SOTIEN\_HP"].HeaderText = "Học phí";

this.dgvDanhSachTN.Columns["TINHTRANG\_HP"].HeaderText = "Tình trạng";

this.dgvDanhSachTN.Columns["NGAYDONG\_HP"].HeaderText = "Ngày đóng";

this.dgvDanhSachTN.Columns["SOTIEN\_BHYT"].HeaderText = "BHYT";

this.dgvDanhSachTN.Columns["TINHTRANG\_BHYT"].HeaderText = "Tình trạng";

this.dgvDanhSachTN.Columns["NGAYDONG\_BHYT"].HeaderText = "Ngày đóng";

this.txtMaSo.Enabled = false;

this.txtMaSo.BackColor = Color.LightGray;

this.txtHoTen.Enabled = false;

this.txtHoTen.BackColor = Color.LightGray;

}

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

{

if(this.dgvDanhSachTN.SelectedRows.Count > 0)

{

this.txtMaSo.Text = this.dgvDanhSachTN.SelectedRows[0].Cells["MAHS"].Value.ToString();

this.txtHoTen.Text = this.dgvDanhSachTN.SelectedRows[0].Cells["HOTEN"].Value.ToString();

this.txtHocPhi.Text = this.dgvDanhSachTN.SelectedRows[0].Cells["SOTIEN\_HP"].Value.ToString();

this.txtBHYT.Text = this.dgvDanhSachTN.SelectedRows[0].Cells["SOTIEN\_BHYT"].Value.ToString();

this.cbTTHP.SelectedItem = this.dgvDanhSachTN.SelectedRows[0].Cells["TINHTRANG\_HP"].Value.ToString();

this.cbTTBHYT.SelectedItem = this.dgvDanhSachTN.SelectedRows[0].Cells["TINHTRANG\_BHYT"].Value.ToString();

this.dtpNgayDongHP.Text = this.dgvDanhSachTN.SelectedRows[0].Cells["NGAYDONG\_HP"].Value.ToString();

this.dtpNgayDongBHYT.Text = this.dgvDanhSachTN.SelectedRows[0].Cells["NGAYDONG\_BHYT"].Value.ToString();

}

}

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

{

if(this.dgvDanhSachTN.SelectedRows.Count > 0)

{

ThuNgan tn = new ThuNgan();

tn.MaNH = this.dgvDanhSachTN.SelectedRows[0].Cells["MANH"].Value.ToString();

tn.MaHK = this.dgvDanhSachTN.SelectedRows[0].Cells["MAHK"].Value.ToString();

tn.MaHS = this.txtMaSo.Text;

tn.TienHP = float.Parse(this.txtHocPhi.Text);

tn.TinhTrangHP = this.cbTTHP.SelectedItem?.ToString();

tn.NgayDongHP = this.dtpNgayDongHP.Value.ToShortDateString();

tn.TienBHYT = float.Parse(this.txtBHYT.Text);

tn.TinTrangBHYT = this.cbTTBHYT.SelectedItem?.ToString();

tn.NgayDongBHYT = this.dtpNgayDongBHYT.Value.ToShortDateString();

string result = thuNganBLL.Update(tn);

MessageBox.Show(result, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

// Refresh Data

this.txtMaSo.Text = string.Empty;

this.txtHoTen.Text = string.Empty;

this.txtHocPhi.Text = string.Empty;

this.txtBHYT.Text = string.Empty;

this.cbTTHP.SelectedIndex = -1;

this.cbTTBHYT.SelectedIndex = -1;

}

}

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

{

thuNganBLL.Cancel();

this.dgvDanhSachTN.ClearSelection();

this.txtMaSo.Enabled = true;

this.txtMaSo.BackColor = Color.White;

this.txtHoTen.Enabled = true;

this.txtHoTen.BackColor = Color.White;

this.txtMaSo.Text = string.Empty;

this.txtHoTen.Text = string.Empty;

this.txtHocPhi.Text = string.Empty;

this.txtBHYT.Text = string.Empty;

this.cbTTHP.SelectedIndex = -1;

this.cbTTBHYT.SelectedIndex = -1;

}

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

{

if (DialogResult.Yes == MessageBox.Show("Bạn có muốn lưu?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question))

{

thuNganBLL.Save();

this.dgvDanhSachTN.ClearSelection();

this.txtMaSo.Enabled = true;

this.txtMaSo.BackColor = Color.White;

this.txtHoTen.Enabled = true;

this.txtHoTen.BackColor = Color.White;

this.txtMaSo.Text = string.Empty;

this.txtHoTen.Text = string.Empty;

this.txtHocPhi.Text = string.Empty;

this.txtBHYT.Text = string.Empty;

this.cbTTHP.SelectedIndex = -1;

this.cbTTBHYT.SelectedIndex = -1;

}

}

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

{

this.dgvDanhSachTN.ClearSelection();

this.txtMaSo.Enabled = true;

this.txtMaSo.BackColor = Color.White;

this.txtHoTen.Enabled = true;

this.txtHoTen.BackColor = Color.White;

this.txtMaSo.Text = string.Empty;

this.txtHoTen.Text = string.Empty;

this.txtHocPhi.Text = string.Empty;

this.txtBHYT.Text = string.Empty;

this.cbTTHP.SelectedIndex = -1;

this.cbTTBHYT.SelectedIndex = -1;

}

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

{

NamHoc namHoc = new NamHoc();

namHoc.MaNH = this.cbMaNH.SelectedValue?.ToString();

namHoc.MaHK = this.cbHocKy.SelectedItem?.ToString();

if(namHoc.MaNH == null || namHoc.MaHK == null)

{

MessageBox.Show("Vui lòng chọn năm học và học kỳ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

else

{

if(this.dgvDanhSachTN.Rows.Count > 0)

{

this.dgvDanhSachTN.DataSource = null;

this.dgvDanhSachTN.Rows.Clear();

}

dtTN = thuNganBLL.GetDataByNamHoc\_HocKy(namHoc);

this.dgvDanhSachTN.DataSource = dtTN;

this.dgvDanhSachTN.Columns["MANH"].HeaderText = "Năm học";

this.dgvDanhSachTN.Columns["MAHK"].HeaderText = "Học kỳ";

this.dgvDanhSachTN.Columns["MAHS"].HeaderText = "Mã HS";

this.dgvDanhSachTN.Columns["HOTEN"].HeaderText = "Họ tên";

this.dgvDanhSachTN.Columns["SOTIEN\_HP"].HeaderText = "Học phí";

this.dgvDanhSachTN.Columns["TINHTRANG\_HP"].HeaderText = "Tình trạng";

this.dgvDanhSachTN.Columns["NGAYDONG\_HP"].HeaderText = "Ngày đóng";

this.dgvDanhSachTN.Columns["SOTIEN\_BHYT"].HeaderText = "BHYT";

this.dgvDanhSachTN.Columns["TINHTRANG\_BHYT"].HeaderText = "Tình trạng";

this.dgvDanhSachTN.Columns["NGAYDONG\_BHYT"].HeaderText = "Ngày đóng";

}

}

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

{

if(this.txtTimKiem.Text != "")

{

DataRow dr = dtTN.Select($"HOTEN = '{this.txtTimKiem.Text}'")[0];

if(dr != null)

{

this.txtTimKiem.Clear();

this.dgvDanhSachTN.ClearSelection();

this.dgvDanhSachTN\_CellContentClick(null, null);

this.dgvDanhSachTN.Rows[dtTN.Rows.IndexOf(dr)].Selected = true;

}

else

{

MessageBox.Show("Không tìm thấy học sinh", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.txtTimKiem.Clear();

}

}

}

}

## 9. frmQuanLyTK

public partial class frmQuanLyTK : Form

{

TaiKhoanBLL taiKhoanBLL = new TaiKhoanBLL();

DataTable dtTK = new DataTable();

public frmQuanLyTK()

{

InitializeComponent();

dtTK = taiKhoanBLL.GetData();

this.dgvDanhSach.DataSource = dtTK;

this.dgvDanhSach.Columns["USERNAME"].HeaderText = "Tên Đăng Nhập";

this.dgvDanhSach.Columns["PASSWORD"].HeaderText = "Mật khẩu";

this.dgvDanhSach.Columns["QUYEN"].HeaderText = "Quyền";

this.txtTenDN.Enabled = false;

this.txtTenDN.BackColor = Color.LightGray;

}

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

{

this.cbQuyen.Enabled = true;

this.dgvDanhSach.ClearSelection();

}

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

{

if (this.dgvDanhSach.SelectedRows.Count > 0)

{

this.cbQuyen.Enabled = false;

this.txtTenDN.Text = this.dgvDanhSach.SelectedRows[0].Cells["USERNAME"].Value.ToString();

this.txtMatKhau.Text = this.dgvDanhSach.SelectedRows[0].Cells["PASSWORD"].Value.ToString();

this.cbQuyen.SelectedItem = this.dgvDanhSach.SelectedRows[0].Cells["QUYEN"].Value.ToString();

}

}

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

{

taiKhoanBLL.Destroy();

this.dgvDanhSach.ClearSelection();

this.txtTenDN.Text = string.Empty;

this.txtMatKhau.Text = string.Empty;

this.cbQuyen.SelectedIndex = -1;

}

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

{

TaiKhoan tk = new TaiKhoan();

tk.Taikhoan = this.txtTenDN.Text;

tk.Matkhau = this.txtMatKhau.Text.ToLower();

tk.Quyen = this.cbQuyen.SelectedItem.ToString();

string result = taiKhoanBLL.Update(tk);

MessageBox.Show(result, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

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

{

if(DialogResult.Yes == MessageBox.Show("Bạn có muốn lưu thông tin không?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question))

{

taiKhoanBLL.Save();

this.dgvDanhSach.ClearSelection();

this.txtTenDN.Text = string.Empty;

this.txtMatKhau.Text = string.Empty;

this.cbQuyen.SelectedIndex = -1;

}

}

}

## 10. frmThongTinGV

public partial class frmThongTinGV : Form

{

private string \_maGV, \_maLop;

GiaoVienBLL giaoVienBLL = new GiaoVienBLL();

LopHocBLL lopHocBLL = new LopHocBLL();

DataTable dtGV = new DataTable();

DataTable dtLH = new DataTable();

public string MaGV { get => \_maGV; set => \_maGV = value; }

public string MaLop { get => \_maLop; set => \_maLop = value; }

public frmThongTinGV(string MaGV)

{

InitializeComponent();

this.MaGV = MaGV;

// Load thông tin giáo viên

dtGV = giaoVienBLL.GetData(MaGV);

this.txtHoTen.Text = dtGV.Rows[0]["HOTEN"].ToString();

this.txtQueQuan.Text = dtGV.Rows[0]["QUEQUAN"].ToString();

this.txtSoDT.Text = dtGV.Rows[0]["SODT"].ToString();

this.txtEmail.Text = dtGV.Rows[0]["EMAIL"].ToString();

this.dtpNamSinh.Text = dtGV.Rows[0]["NGAYSINH"].ToString();

// Load thông tin lớp học

dtLH = lopHocBLL.GetDataByGVCV(MaGV);

dtLH.Columns.Add("TenLop\_NamHoc", typeof(string), "TENLOP + ' - ' + MANH");

this.cbMaLop.DataSource = dtLH;

this.cbMaLop.DisplayMember = "TenLop\_NamHoc";

this.cbMaLop.ValueMember = "MALOP";

}

private async void btnQuanLy\_Click(object sender, EventArgs e)

{

for(int i=0; i<=150; i++)

{

pbLoading.Value = i;

await Task.Delay(50);

}

MaLop = this.cbMaLop.SelectedValue?.ToString();

if (MaLop == null)

{

MessageBox.Show("Giáo viên này chưa quản lý lớp");

return;

}

this.Hide();

frmStudentManagement frm = new frmStudentManagement(MaLop);

frm.ShowDialog();

pbLoading.Value = 0;

this.Show();

}

}

## 11. frmStudentManagement

public partial class frmStudentManagement : Form

{

private Form \_currentForm = null;

private string \_maLop;

public string MaLop { get => \_maLop; set => \_maLop = value; }

public frmStudentManagement(string maLop)

{

InitializeComponent();

MaLop = maLop;

this.SwitchFrom(new frmQuanLy(maLop));

}

public void SwitchFrom(Form childForm)

{

if (this.\_currentForm != null)

this.\_currentForm.Close();

// Đặt thuộc tính cho childForm

childForm.TopLevel = false;

childForm.Dock = DockStyle.Fill;

// Thêm childForm vào pnContainer

pnContainer.Controls.Clear();

pnContainer.Controls.Add(childForm);

childForm.Show();

this.\_currentForm = childForm;

}

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

{

this.Close();

}

public void tsMnItemDangXuat\_Click(object sender, EventArgs e)

{

this.Close();

}

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

{

this.SwitchFrom(new frmQuanLy(MaLop));

}

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

{

this.SwitchFrom(new frmDanhGiaHanhKiem(MaLop));

}

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

{

this.SwitchFrom(new frmDiemSo(MaLop));

}

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

{

this.Close();

}

}

## 12. frmQuanLy

public partial class frmQuanLy : Form

{

private string \_maLop;

HocSinhBLL hocSinhBLL = new HocSinhBLL();

LopHocBLL LopHocBLL = new LopHocBLL();

DataTable dtHS = new DataTable();

public string MaLop { get => \_maLop; set => \_maLop = value; }

public frmQuanLy(string maLop)

{

InitializeComponent();

MaLop = maLop;

// Load thông tin học sinh

dtHS = hocSinhBLL.GetDataByMaLop(MaLop);

this.dgvDanhSach.DataSource = dtHS;

this.dgvDanhSach.Columns["MAHS"].HeaderText = "Mã học sinh";

this.dgvDanhSach.Columns["HOTEN"].HeaderText = "Họ tên";

this.dgvDanhSach.Columns["NGAYSINH"].HeaderText = "Ngày sinh";

this.dgvDanhSach.Columns["DIACHI"].HeaderText = "Địa chỉ";

this.dgvDanhSach.Columns["SODT"].HeaderText = "Số điện thoại";

this.dgvDanhSach.Columns["GIOITINH"].HeaderText = "Giới tính";

this.dgvDanhSach.Columns["MALOP"].Visible = false;

this.dgvDanhSach.Columns["USERNAME"].Visible = false;

// Get tên lớp

DataTable lopHoc = LopHocBLL.GetDataByMaLop(MaLop);

this.lblLop.Text = lopHoc.Rows[0]["TENLOP"].ToString();

this.txtNamHoc.ReadOnly = true;

this.txtNamHoc.BackColor = Color.LightGray;

this.txtNamHoc.Text = lopHoc.Rows[0]["MANH"].ToString();

}

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

{

if(this.dgvDanhSach.SelectedRows.Count > 0)

{

this.txtMaso.Text = this.dgvDanhSach.SelectedRows[0].Cells["MAHS"].Value.ToString();

this.txtHoTen.Text = this.dgvDanhSach.SelectedRows[0].Cells["HOTEN"].Value.ToString();

this.dtpNgaySinh.Text = this.dgvDanhSach.SelectedRows[0].Cells["NGAYSINH"].Value.ToString();

this.txtDiaChi.Text = this.dgvDanhSach.SelectedRows[0].Cells["DIACHI"].Value.ToString();

this.txtSoDT.Text = this.dgvDanhSach.SelectedRows[0].Cells["SODT"].Value.ToString();

this.cbGioiTinh.Text = this.dgvDanhSach.SelectedRows[0].Cells["GIOITINH"].Value.ToString();

}

}

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

{

if(this.txtTimKiem.Text != "")

{

DataRow[] findRows = dtHS.Select($"HOTEN = '{this.txtTimKiem.Text}'");

if(findRows.Length > 0)

{

DataRow findRow = findRows[0];

this.dgvDanhSach.Rows[dtHS.Rows.IndexOf(findRow)].Selected = true;

this.dgvDanhSach\_CellContentClick(null, null);

this.dgvDanhSach.ClearSelection();

this.txtTimKiem.Clear();

}

else

{

MessageBox.Show("Không tìm thấy học sinh", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.txtTimKiem.Clear();

}

}

}

}

## 13. frmDiemSo

public partial class frmDiemSo : Form

{

private string \_maLop;

private LopHoc \_lopHoc;

BangDiemBLL bangDiemBLL = new BangDiemBLL();

LopHocBLL lopHocBLL = new LopHocBLL();

DataTable dtBD;

public string MaLop { get => \_maLop; set => \_maLop = value; }

public LopHoc LopHoc { get => \_lopHoc; set => \_lopHoc = value; }

public frmDiemSo(string maLop)

{

InitializeComponent();

MaLop = maLop;

// Load Danh sách Điểm học sinh theo lớp

// Get thông tin lớp học

DataRow rowLH = lopHocBLL.GetDataByMaLop(MaLop).Rows[0];

this.LopHoc = new LopHoc();

this.LopHoc.MaLop = rowLH["MALOP"].ToString();

this.LopHoc.MaNH = rowLH["MANH"].ToString();

// Load thông tin bảng điểm

dtBD = bangDiemBLL.GetDataByMaLop(this.LopHoc);

this.dgvDanhSach.DataSource = dtBD;

this.dgvDanhSach.Columns["MAHS"].HeaderText = "Mã số";

this.dgvDanhSach.Columns["HOTEN"].HeaderText = "Họ tên";

this.dgvDanhSach.Columns["HOTEN"].Width = 200;

this.dgvDanhSach.Columns["MAHK"].HeaderText = "Học kỳ";

this.dgvDanhSach.Columns["TOAN"].HeaderText = "Toán";

this.dgvDanhSach.Columns["NGUVAN"].HeaderText = "Ngữ văn";

this.dgvDanhSach.Columns["NGOAINGU"].HeaderText = "Ngoại ngữ";

this.dgvDanhSach.Columns["VATLY"].HeaderText = "Vật lý";

this.dgvDanhSach.Columns["HOAHOC"].HeaderText = "Hóa học";

this.dgvDanhSach.Columns["SINHHOC"].HeaderText = "Sinh học";

this.dgvDanhSach.Columns["DIALY"].HeaderText = "Địa lý";

this.dgvDanhSach.Columns["LICHSU"].HeaderText = "Lịch sử";

this.dgvDanhSach.Columns["GDCD"].HeaderText = "GDCD";

this.dgvDanhSach.Columns["DTB"].HeaderText = "ĐTB";

this.txtDTB.Enabled = false;

this.txtDTB.BackColor = Color.LightGray;

this.txtHoTen.Enabled = false;

this.txtHoTen.BackColor = Color.LightGray;

this.dgvDanhSach.ClearSelection();

this.cbHocKy.SelectedIndex = 0;

}

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

{

if (this.dgvDanhSach.SelectedRows.Count > 0)

{

this.txtToan.Focus();

this.cbHocKy.SelectedItem = this.dgvDanhSach.SelectedRows[0].Cells["MAHK"].Value.ToString();

this.txtHoTen.Text = this.dgvDanhSach.SelectedRows[0].Cells["HOTEN"].Value.ToString();

this.txtToan.Text = this.dgvDanhSach.SelectedRows[0].Cells["TOAN"].Value.ToString();

this.txtVan.Text = this.dgvDanhSach.SelectedRows[0].Cells["NGUVAN"].Value.ToString();

this.txtNgNgu.Text = this.dgvDanhSach.SelectedRows[0].Cells["NGOAINGU"].Value.ToString();

this.txtLy.Text = this.dgvDanhSach.SelectedRows[0].Cells["VATLY"].Value.ToString();

this.txtHoa.Text = this.dgvDanhSach.SelectedRows[0].Cells["HOAHOC"].Value.ToString();

this.txtSinh.Text = this.dgvDanhSach.SelectedRows[0].Cells["SINHHOC"].Value.ToString();

this.txtSu.Text = this.dgvDanhSach.SelectedRows[0].Cells["LICHSU"].Value.ToString();

this.txtDia.Text = this.dgvDanhSach.SelectedRows[0].Cells["DIALY"].Value.ToString();

this.txtGDCD.Text = this.dgvDanhSach.SelectedRows[0].Cells["GDCD"].Value.ToString();

this.txtDTB.Text = this.dgvDanhSach.SelectedRows[0].Cells["DTB"].Value.ToString();

}

}

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

{

if(this.cbHocKy.SelectedIndex != -1)

{

if(this.dgvDanhSach.Rows.Count > 0)

{

this.dgvDanhSach.DataSource = null;

this.dgvDanhSach.Rows.Clear();

}

dtBD = bangDiemBLL.GetDataByMaLop(this.LopHoc, this.cbHocKy.SelectedItem.ToString());

this.dgvDanhSach.DataSource = dtBD;

this.dgvDanhSach.Columns["MAHS"].HeaderText = "Mã số";

this.dgvDanhSach.Columns["HOTEN"].HeaderText = "Họ tên";

this.dgvDanhSach.Columns["HOTEN"].Width = 200;

this.dgvDanhSach.Columns["MAHK"].HeaderText = "Học kỳ";

this.dgvDanhSach.Columns["TOAN"].HeaderText = "Toán";

this.dgvDanhSach.Columns["NGUVAN"].HeaderText = "Ngữ văn";

this.dgvDanhSach.Columns["NGOAINGU"].HeaderText = "Ngoại ngữ";

this.dgvDanhSach.Columns["VATLY"].HeaderText = "Vật lý";

this.dgvDanhSach.Columns["HOAHOC"].HeaderText = "Hóa học";

this.dgvDanhSach.Columns["SINHHOC"].HeaderText = "Sinh học";

this.dgvDanhSach.Columns["DIALY"].HeaderText = "Địa lý";

this.dgvDanhSach.Columns["LICHSU"].HeaderText = "Lịch sử";

this.dgvDanhSach.Columns["GDCD"].HeaderText = "GDCD";

this.dgvDanhSach.Columns["DTB"].HeaderText = "ĐTB";

}

}

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

{

if (this.dgvDanhSach.SelectedRows.Count > 0)

{

this.dgvDanhSach.ClearSelection();

this.txtHoTen.Clear();

this.txtToan.Clear();

this.txtVan.Clear();

this.txtNgNgu.Clear();

this.txtLy.Clear();

this.txtHoa.Clear();

this.txtSinh.Clear();

this.txtSu.Clear();

this.txtDia.Clear();

this.txtGDCD.Clear();

this.txtDTB.Clear();

}

}

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

{

BangDiem bangDiem = new BangDiem();

bangDiem.MaNH = this.LopHoc.MaNH;

bangDiem.MaHK = this.cbHocKy.SelectedItem?.ToString();

bangDiem.MaHS = this.dgvDanhSach.SelectedRows[0].Cells["MAHS"].Value.ToString();

bangDiem.Toan = this.txtToan.Text;

bangDiem.NgVan = this.txtVan.Text;

bangDiem.NgNgu = this.txtNgNgu.Text;

bangDiem.VatLy = this.txtLy.Text;

bangDiem.HoaHoc = this.txtHoa.Text;

bangDiem.SinhHoc = this.txtSinh.Text;

bangDiem.DiaLy = this.txtDia.Text;

bangDiem.LichSu = this.txtSu.Text;

bangDiem.Gdcd = this.txtGDCD.Text;

string result = bangDiemBLL.Update(bangDiem);

MessageBox.Show(result, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.txtHoTen.Clear();

this.txtToan.Clear();

this.txtVan.Clear();

this.txtNgNgu.Clear();

this.txtLy.Clear();

this.txtHoa.Clear();

this.txtSinh.Clear();

this.txtSu.Clear();

this.txtDia.Clear();

this.txtGDCD.Clear();

this.txtDTB.Clear();

this.dgvDanhSach.ClearSelection();

}

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

{

bangDiemBLL.Cancel();

this.cbHocKy.SelectedIndex = 0;

this.txtHoTen.Clear();

this.txtToan.Clear();

this.txtVan.Clear();

this.txtNgNgu.Clear();

this.txtLy.Clear();

this.txtHoa.Clear();

this.txtSinh.Clear();

this.txtSu.Clear();

this.txtDia.Clear();

this.txtGDCD.Clear();

this.txtDTB.Clear();

this.dgvDanhSach.ClearSelection();

}

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

{

if (DialogResult.Yes == MessageBox.Show("Bạn có muốn lưu?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question))

{

bangDiemBLL.Save();

this.cbHocKy.SelectedIndex = 0;

this.txtHoTen.Clear();

this.txtToan.Clear();

this.txtVan.Clear();

this.txtNgNgu.Clear();

this.txtLy.Clear();

this.txtHoa.Clear();

this.txtSinh.Clear();

this.txtSu.Clear();

this.txtDia.Clear();

this.txtGDCD.Clear();

this.txtDTB.Clear();

this.dgvDanhSach.ClearSelection();

}

}

}

## 14. frmDanhGiaHanhKiem

public partial class frmDanhGiaHanhKiem : Form

{

private string \_maLop;

private LopHoc \_lopHoc;

HanhKiemBLL hanhKiemBLL = new HanhKiemBLL();

LopHocBLL lopHocBLL = new LopHocBLL();

DataTable dtHK;

public string MaLop { get => \_maLop; set => \_maLop = value; }

public LopHoc LopHoc { get => \_lopHoc; set => \_lopHoc = value; }

public frmDanhGiaHanhKiem(string maLop)

{

InitializeComponent();

this.TopLevel = false;

this.MaLop = maLop;

// Load Danh sách Điểm học sinh theo lớp

// Get thông tin lớp học

DataRow rowLH = lopHocBLL.GetDataByMaLop(MaLop).Rows[0];

this.LopHoc = new LopHoc();

this.LopHoc.MaLop = rowLH["MALOP"].ToString();

this.LopHoc.MaNH = rowLH["MANH"].ToString();

// Load data Hạnh kiểm

dtHK = hanhKiemBLL.GetDataByMaLop(this.LopHoc);

this.dgvDanhSach.DataSource = dtHK;

this.dgvDanhSach.Columns["MAHK"].HeaderText = "Học kỳ";

this.dgvDanhSach.Columns["MAHS"].HeaderText = "Mã số";

this.dgvDanhSach.Columns["HOTEN"].HeaderText = "Họ tên";

this.dgvDanhSach.Columns["LOAIHK"].HeaderText = "Loại hạnh kiểm";

this.txtHoTen.Enabled = true;

this.txtHoTen.BackColor = Color.LightGray;

this.cbHocKy.SelectedIndex = 0;

}

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

{

if (this.dgvDanhSach.SelectedRows.Count > 0)

{

this.cbHocKy.SelectedItem = this.dgvDanhSach.SelectedRows[0].Cells["MAHK"].Value.ToString();

this.txtHoTen.Text = this.dgvDanhSach.SelectedRows[0].Cells["HOTEN"].Value.ToString();

this.cbLoaiHK.SelectedItem = this.dgvDanhSach.SelectedRows[0].Cells["LOAIHK"].Value.ToString();

}

}

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

{

if(this.cbHocKy.SelectedIndex != -1)

{

if(this.dgvDanhSach.Rows.Count > 0)

{

this.dgvDanhSach.DataSource = null;

this.dgvDanhSach.Rows.Clear();

}

dtHK = hanhKiemBLL.GetDataByMaLop(this.LopHoc, this.cbHocKy.SelectedItem.ToString());

this.dgvDanhSach.DataSource = dtHK;

this.dgvDanhSach.Columns["MAHK"].HeaderText = "Học kỳ";

this.dgvDanhSach.Columns["MAHS"].HeaderText = "Mã số";

this.dgvDanhSach.Columns["HOTEN"].HeaderText = "Họ tên";

this.dgvDanhSach.Columns["LOAIHK"].HeaderText = "Loại hạnh kiểm";

this.txtHoTen.Clear();

this.cbLoaiHK.SelectedIndex = -1;

}

}

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

{

HanhKiem hk = new HanhKiem();

hk.MaNH = this.LopHoc.MaNH;

hk.MaHK = this.cbHocKy.SelectedItem?.ToString();

hk.MaHS = this.dgvDanhSach.SelectedRows[0].Cells["MAHS"].Value.ToString();

hk.LoaiHK = this.cbLoaiHK.SelectedItem?.ToString();

string result = hanhKiemBLL.Update(hk);

MessageBox.Show(result, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.txtHoTen.Clear();

this.cbLoaiHK.SelectedIndex = -1;

}

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

{

hanhKiemBLL.Cancel();

this.dgvDanhSach.ClearSelection();

this.cbHocKy.SelectedIndex = 0;

this.txtHoTen.Text = "";

this.cbLoaiHK.SelectedIndex = -1;

}

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

{

if (DialogResult.Yes == MessageBox.Show("Bạn có muốn lưu?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question))

{

hanhKiemBLL.Save();

this.dgvDanhSach.ClearSelection();

this.cbHocKy.SelectedIndex = 0;

this.txtHoTen.Text = "";

this.cbLoaiHK.SelectedIndex = -1;

}

}

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

{

if(this.dgvDanhSach.SelectedRows.Count > 0)

this.dgvDanhSach.ClearSelection();

this.cbHocKy.SelectedIndex = 0;

this.txtHoTen.Text = "";

this.cbLoaiHK.SelectedIndex = -1;

}

}

# Tài liệu tham khảo

- Cơ sở dữ liệu: <http://surl.li/zibahl>

- Xây dựng theo mô hình 3 lớp: <http://surl.li/mlafcv>