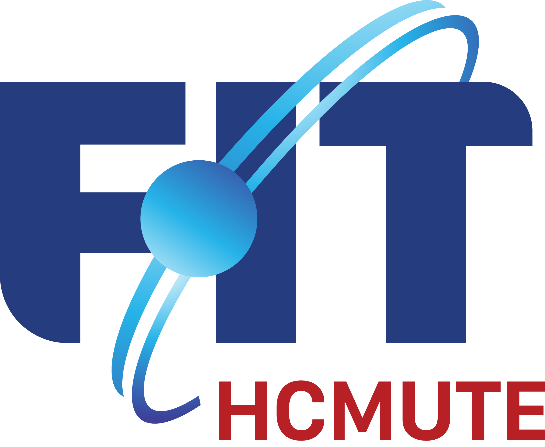


**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-----🙞🙞🙞🙞🙞-----**



**ĐỀ TÀI CUỐI KỲ MÔN HỆ QUẢN TRỊ CSDL**

**TÊN ĐỀ TÀI: XÂY DỰNG HỆ THỐNG QUẢN LÝ GIẢNG VIÊN**

**GVHD:** TS. Nguyễn Thành Sơn

**Lớp HP:** DBMS330284\_01

**SV thực hiện:** 23110244 – Đoàn Quang Khôi - Nhóm 5

**Học kỳ:** 1

**Năm học:** 2025 - 2026

*Thành phố Hồ Chí Minh, tháng 9, năm 2025*

**SINH VIÊN THỰC HIỆN**

HỌC KÌ 1 NĂM HỌC 2025 - 2026

**Nhóm 5**

**Đề tài:** Quản lý giảng viên

|  |  |
| --- | --- |
| **Họ tên sinh viên** | **Mã số sinh viên** |
| Đoàn Quang Khôi | 23110244 |

**NHẬN XÉT CỦA GV**

*Tp. Hồ Chí Minh, ngày …., tháng..., năm 2025*

**Giảng viên chấm điểm**

**Mục Lục**

[**CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG QUẢN LÝ** 4](#_Toc210085050)

[**1. Đặc tả đề tài** 4](#_Toc210085051)

[**1.1. Mô tả bài toán** 4](#_Toc210085052)

[**1.2 Mô tả dữ liệu** 4](#_Toc210085053)

[**1.3 Mô tả chức năng** 4](#_Toc210085054)

[**1.4 Mô tả giao diện** 5](#_Toc210085055)

[**CHƯƠNG 2: THIẾT KẾ CƠ SỞ DỮ LIỆU** 6](#_Toc210085056)

[**1.** **Thiết kế cơ sở dữ liệu mức quan niệm** 6](#_Toc210085057)

[**2.** **Thiết kế cơ sở dữ liệu mức logic** 6](#_Toc210085058)

[**3.** **Các ràng buộc** 7](#_Toc210085059)

[**4.** **Cài đặt CSDL và các ràng buộc** 11](#_Toc210085060)

[**CHƯƠNG 3. THIẾT KẾ CÁC CHỨC NĂNG** 22](#_Toc210085061)

[**1.** **Kết nối cơ sở dữ liệu** 22](#_Toc210085062)

[**2.** **Các chức năng của quản lý (Admin)** 23](#_Toc210085063)

[**3.** **Các chức năng của trưởng khoa** 24](#_Toc210085064)

[**4.** **Các chức năng của giảng viên** 36](#_Toc210085065)

[**CHƯƠNG 4. TẠO USER VÀ PHÂN QUYỀN** 37](#_Toc210085066)

[**1.** **Tạo role và phân quyền** 37](#_Toc210085067)

[**2.** **Đồng bộ hóa CRUD bảng TAIKHOAN với Login/User của SQL Server** 37](#_Toc210085068)

[**CHƯƠNG 5. THIẾT KẾ GIAO DIỆN HỆ THỐNG** 40](#_Toc210085069)

[**1.** **Kiến trúc hệ thống** 40](#_Toc210085070)

[**a.** **Cấu trúc thư mục hệ thống** 40](#_Toc210085071)

[**b.** **Mô hình dữ liệu (2 layer)** 40](#_Toc210085072)

[**2.** **Giao diện và chức năng của chương trình** 41](#_Toc210085073)

# **CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG QUẢN LÝ**

## **1. Đặc tả đề tài**

### **1.1. Mô tả bài toán**

Đề tài hướng tới việc xây dựng một hệ thống quản lý cán bộ giảng viên trong trường đại học. Bài toán thực tế đặt ra là mỗi giảng viên, trưởng khoa và admin đều cần một công cụ hỗ trợ quản lý dữ liệu phức tạp về nhân sự, môn học, lớp học phần, phân công giảng dạy và lương bổng. Các hoạt động hiện tại thường được thực hiện rời rạc bằng giấy tờ hoặc file Excel, gây khó khăn trong tra cứu, thống kê và đảm bảo tính chính xác. Hệ thống được thiết kế phải giải quyết bài toán phân quyền truy cập (admin có toàn quyền, trưởng khoa chỉ quản lý khoa mình, giảng viên chỉ xem thông tin liên quan đến bản thân), đồng thời bảo đảm dữ liệu được lưu trữ tập trung, toàn vẹn và dễ dàng mở rộng.

### **1.2 Mô tả dữ liệu**

Dữ liệu hệ thống được thiết kế trên SQL Server, bao gồm nhiều thực thể quan trọng. Các bảng chính là Role, Users, TaiKhoan để quản lý tài khoản và phân quyền; Khoa, Nganh, ChuongTrinh để tổ chức bộ máy khoa ngành; CanBo lưu thông tin cán bộ giảng viên; MonHoc, LopHocPhan mô tả các môn học và lớp học phần; PhanCongGiangDay quản lý việc phân công giảng dạy; và BangLuong, BacLuong, CauHinhLuong quản lý hệ thống lương. Các ràng buộc quan hệ và trigger đảm bảo tính nhất quán, ví dụ: một khoa chỉ có một trưởng khoa, không thể xóa môn học đã được phân công, hoặc khi nhập điểm sinh viên thì tự động tính trạng thái đạt/không đạt. Ngoài ra, dữ liệu mẫu đã được khởi tạo để minh họa cho nhiều khoa khác nhau, giảng viên cụ thể và các lớp học phần.

### **1.3 Mô tả chức năng**

Hệ thống cung cấp đầy đủ các chức năng CRUD (thêm, sửa, xóa, tra cứu) cho từng nhóm dữ liệu. Người dùng có thể đăng nhập bằng tài khoản SQL login, hệ thống tự xác định vai trò và khoa để áp dụng phân quyền. Admin có thể quản lý toàn bộ giảng viên, khoa, ngành, môn học, lớp học phần, phân công giảng dạy và tính lương. Trưởng khoa có quyền quản lý giảng viên, môn học, lớp học phần và phân công trong khoa của mình, đồng thời chỉ được xem bảng lương chứ không được chỉnh sửa. Giảng viên chỉ có thể xem hồ sơ cá nhân, lịch phân công, môn học và bảng lương của bản thân. Ngoài ra, hệ thống còn có các chức năng đặc thù như tính lương bằng stored procedure kèm transaction, trigger tự động cập nhật số lượng sinh viên trong lớp, function kiểm tra năm học hợp lệ.

### **1.4 Mô tả giao diện**

Phần giao diện được xây dựng bằng WinForms trên C#, với thiết kế trực quan và chia thành nhiều form chuyên biệt: **frm\_DangNhap** để đăng nhập, **frm\_QuanLyCanBo** làm form chính điều hướng, cùng các form chức năng như frm\_QuanLyGiangVien, frm\_QuanLyNganh, frm\_QuanLyMonHoc, frm\_QuanLyLopHocPhan, frm\_PhanCongGiangDay, frm\_QuanLyLuong và frm\_HoSoCanBo. Các form này được thiết kế có bảng dữ liệu (DataGridView) để hiển thị thông tin, các combobox để chọn khoa/ngành/môn/lớp, các textbox và nút bấm để nhập liệu và thao tác. Giao diện cũng phân quyền trực tiếp: ví dụ khi đăng nhập với role Giảng viên thì các nút thêm/sửa/xóa sẽ bị khóa, chỉ cho phép xem thông tin. Sự kết hợp này giúp giao diện vừa thân thiện, dễ dùng, vừa thể hiện rõ tính logic của phân quyền trong quản lý.

# **CHƯƠNG 2: THIẾT KẾ CƠ SỞ DỮ LIỆU**

## **Thiết kế cơ sở dữ liệu mức quan niệm**

1. **Thiết kế cơ sở dữ liệu mức logic**

Lược đồ quan hệ

* Role(RoleId, RoleName, Description)
* Users(UserId, FullName, Email, Phone, RoleId, IsActive)
* TaiKhoan(AccountId, UserId, Username, Password, IsLocked)
* Khoa(MaKhoa, TenKhoa)
* Nganh(MaNganh, TenNganh, MaKhoa)
* ChuongTrinh(MaCT, TenCT, MaNganh, HinhThuc, HeDaoTao)
* ChucVu(MaChucVu, TenChucVu)
* TrinhDo(MaTrinhDo, TenTrinhDo)
* CanBo(MaCB, HoTen, NgaySinh, GioiTinh, Email, Phone, MaKhoa, MaChucVu, MaTrinhDo, UserId, MaBacLuong, PhuCap)
* MonHoc(MonHocId, MaMon, TenMon, SoTiet, SoTinChi, MaKhoaPhuTrach)
* ChuongTrinh\_MonHoc(MaCT, MaMon, BatBuoc)
* LopHocPhan(MaLopHocPhan, TenLopHocPhan, MaKhoa, MaNganh, MaCT, SoLuongSV, MaMon)
* PhanCongGiangDay(MaCB, TenMon, MaMon, MaLopHocPhan, SoTiet, SoTuan, HocKy, NamHoc, MaNganh)
* BacLuong(MaBacLuong, HeSoLuong)
* BangLuong(MaBangLuong, MaCB, Thang, Nam, Thuong, KhauTru)
* CauHinhLuong(MucLuongCoSo)

1. **Các ràng buộc**

|  |  |  |
| --- | --- | --- |
| **STT** | **BẢNG** | **RÀNG BUỘC** |
| **1** | **ROLE** | IDENTITY PRIMARY KEY RoleId INT  NOT NULL UNIQUE RoleName NVARCHAR(50)  NULL Description NVARCHAR(255) |
| **2** | **USERS** | IDENTITY PRIMARY KEY UserId INT  NOT NULL FullName NVARCHAR(150)  NULL Email NVARCHAR(150)  NULL Phone NVARCHAR(20)  NOT NULL RoleId INT  NOT NULL DEFAULT 1 IsActive BIT  RoleId tham chiếu đến bảng ROLE |
| **3** | **TaiKhoan** | IDENTITY PRIMARY KEY AccountId INT  NOT NULL UNIQUE UserId INT,  NOT NULL UNIQUE Username NVARCHAR(100),  NOT NULL Password NVARCHAR(255)  NULL DEFAULT 0 IsLocked BIT NOT,  UserId tham chiếu tới Users |
| **4** | **Khoa** | NOT NULL PRIMARY KEY MaKhoa NVARCHAR(20)  NOT NULL TenKhoa NVARCHAR(200) |
| **5** | **Nganh** | MaNganh NVARCHAR(20) NOT NULL PRIMARY KEY,  TenNganh NVARCHAR(200) NOT NULL,  MaKhoa NVARCHAR(20) NOT NULL, MaKhoa tham chiếu tới Khoa |
| **6** | **ChuongTrinh** | NOT NULL PRIMARY KEY MaCT NVARCHAR(20),  NOT NULL TenCT NVARCHAR(200),  NOT NULL MaNganh NVARCHAR(20),  NOT NULL HinhThuc NVARCHAR(20),  NOT NULL CHECK (HeDaoTao IN (N'Đại trà', N'CLC')) HeDaoTao NVARCHAR(20)  MaNganh tham chiếu tới bảng Ngành |
| **7** | **ChucVu** | NOT NULL PRIMARY KEY MaChucVu NVARCHAR(20),  NOT NULL TenChucVu NVARCHAR(100) |
| **8** | **TrinhDo** | NOT NULL PRIMARY KEY MaTrinhDo NVARCHAR(5),  NOT NULL UNIQUE TenTrinhDo NVARCHAR(100) |
| **9** | **CanBo** | NOT NULL UNIQUE MaCB NVARCHAR(20) NOT NULL UNIQUE,  NOT NULL HoTen NVARCHAR(150),  NULL NgaySinh DATE,  NULL CHECK (GioiTinh IN ('M','F') GioiTinh CHAR(1)),  NULL Email NVARCHAR(150),  NULL Phone NVARCHAR(20),  NULL MaKhoa NVARCHAR(20),  NULL MaChucVu NVARCHAR(20),  NULL MaTrinhDo NVARCHAR(5),  NULL UserID int  MaKhoa tham chiếu đến bảng Khoa  MaChucVu tham chiếu đến bảng ChucVu  MaTrinhDo tham chiếu đến bảng TrinhDo  UserID tham chiếu đến bảng Users |
| **10** | **MonHoc** | MonHocId INT IDENTITY PRIMARY KEY,  MaMon NVARCHAR(20) NOT NULL UNIQUE,  TenMon NVARCHAR(200) NOT NULL DEFAULT N'Chưa đặt tên',  SoTiet INT NOT NULL CHECK (SoTiet >= 0),  SoTinChi INT NOT NULL CHECK (SoTinChi >= 0),  MaKhoaPhuTrach NVARCHAR(20) NOT NULL |
| **11** | **ChuongTrinh\_MonHoc** | MaCT NVARCHAR(20) NOT NULL,  BatBuoc BIT NOT NULL DEFAULT 1,  MaMon NVARCHAR(20) NOT NULL,  MaCT tham chiếu đến bảng ChuongTrinh  MaMon tham chiếu đến bảng MonHoc  Tràng buộc UNIQUE (MaCT, MaMon) |
| **12** | **LopHocPhan** | MaLopHocPhan NVARCHAR(20) NOT NULL PRIMARY KEY,  TenLopHocPhan NVARCHAR(200) NOT NULL,  MaKhoa NVARCHAR(20) NULL,  MaNganh NVARCHAR(20) NULL,  MaCT NVARCHAR(20) NULL,  SoLuongSV INT NOT NULL DEFAULT 0,  MaMon NVARCHAR(20) NOT NULL,  MaKhoa tham chiếu đến bảng Khoa  MaNganh tham chiếu đến bảng Nganh  MaCT tham chiếu đến bảng ChuongTrinh  MaMon tham chiếu đến bảng MonHoc |
| **13** | **PhanCongGiangDay** | MaCB NVARCHAR(20) NOT NULL,  TenMon NVARCHAR(200) NOT NULL,  MaMon NVARCHAR(20) NOT NULL,  MaLopHocPhan NVARCHAR(20) NOT NULL,  SoTiet INT NOT NULL CHECK (SoTiet >= 0),  SoTuan INT NOT NULL DEFAULT 15 CHECK (SoTuan > 0),  HocKy INT NOT NULL,  NamHoc NVARCHAR(20) NOT NULL,  MaNganh NVARCHAR(20) NOT NULL  MaCB tham chiếu đến bảng CanBo  MaMon tham chiếu đến bảng MonHoc  MaLopHocPhan tham chiếu đến bảng LopHocPhan  MaNganh tham chiếu đến bảng Nganh |
| **14** | **BacLuong** | MaBacLuong NVARCHAR(20) PRIMARY KEY,  HeSoLuong DECIMAL(4,2) |
| **15** | **BangLuong** | MaCB NVARCHAR(20) NOT NULL,  Thang INT,  Nam INT,  Thuong DECIMAL(18,2) DEFAULT 0,  KhauTru DECIMAL(18,2) DEFAULT 0,  MaCB tham chiếu đến bảng CanBo  Ràng buộc UNIQUE (MaCB, Thang, Nam) |
| **16** | **CauHinhLuong** | MucLuongCoSo DECIMAL(18,2) |

1. **Cài đặt CSDL và các ràng buộc**

**Bảng Role**

CREATE TABLE Role (

RoleId INT IDENTITY PRIMARY KEY,

RoleName NVARCHAR(50) NOT NULL UNIQUE,

Description NVARCHAR(255) NULL

);

**Bảng Users**

CREATE TABLE Users (

UserId INT IDENTITY PRIMARY KEY,

FullName NVARCHAR(150) NOT NULL,

Email NVARCHAR(150) NULL,

Phone NVARCHAR(20) NULL,

RoleId INT NOT NULL,

IsActive BIT NOT NULL DEFAULT 1,

CONSTRAINT FK\_Users\_Role FOREIGN KEY (RoleId) REFERENCES Role(RoleId)

ON DELETE NO ACTION ON UPDATE CASCADE

);

**Bảng TaiKhoan**

CREATE TABLE TaiKhoan (

AccountId INT IDENTITY PRIMARY KEY,

UserId INT NOT NULL UNIQUE,

Username NVARCHAR(100) NOT NULL UNIQUE,

Password NVARCHAR(255) NOT NULL,

IsLocked BIT NOT NULL DEFAULT 0,

CONSTRAINT FK\_TaiKhoan\_Users FOREIGN KEY (UserId) REFERENCES Users(UserId) ON DELETE CASCADE ON UPDATE CASCADE

);

**Bảng Khoa**

CREATE TABLE Khoa (

MaKhoa NVARCHAR(20) NOT NULL PRIMARY KEY,

TenKhoa NVARCHAR(200) NOT NULL

);

**Bảng Nganh**

CREATE TABLE Nganh (

MaNganh NVARCHAR(20) NOT NULL PRIMARY KEY,

TenNganh NVARCHAR(200) NOT NULL,

MaKhoa NVARCHAR(20) NOT NULL,

CONSTRAINT FK\_Nganh\_Khoa FOREIGN KEY (MaKhoa) REFERENCES Khoa(MaKhoa) ON DELETE NO ACTION ON UPDATE CASCADE

);

**Bảng ChuongTrinh**

CREATE TABLE ChuongTrinh (

MaCT NVARCHAR(20) NOT NULL PRIMARY KEY,

TenCT NVARCHAR(200) NOT NULL,

MaNganh NVARCHAR(20) NOT NULL,

HinhThuc NVARCHAR(20) NOT NULL,

HeDaoTao NVARCHAR(20) NOT NULL CHECK (HeDaoTao IN (N'Đại trà', N'CLC')),

CONSTRAINT FK\_ChuongTrinh\_Nganh FOREIGN KEY (MaNganh) REFERENCES Nganh(MaNganh)

ON DELETE NO ACTION ON UPDATE CASCADE

);

**Bảng ChucVu**

CREATE TABLE ChucVu (

MaChucVu NVARCHAR(20) NOT NULL PRIMARY KEY,

TenChucVu NVARCHAR(100) NOT NULL

);

**Bảng TrinhDo**

CREATE TABLE TrinhDo (

MaTrinhDo NVARCHAR(5) NOT NULL PRIMARY KEY,

TenTrinhDo NVARCHAR(100) NOT NULL UNIQUE

);

**Bảng CanBo**

CREATE TABLE CanBo (

MaCB NVARCHAR(20) NOT NULL UNIQUE,

HoTen NVARCHAR(150) NOT NULL,

NgaySinh DATE NULL,

GioiTinh CHAR(1) NULL CHECK (GioiTinh IN ('M','F')),

Email NVARCHAR(150) NULL,

Phone NVARCHAR(20) NULL,

MaKhoa NVARCHAR(20) NULL, -- tham chiếu Khoa.MaKhoa

MaChucVu NVARCHAR(20) NULL, -- tham chiếu ChucVu.MaChucVu

MaTrinhDo NVARCHAR(5) NULL,

CONSTRAINT FK\_CanBo\_MaKhoa FOREIGN KEY (MaKhoa) REFERENCES Khoa(MaKhoa) ON DELETE SET NULL ON UPDATE CASCADE,

CONSTRAINT FK\_CanBo\_MaChucVu FOREIGN KEY (MaChucVu) REFERENCES ChucVu(MaChucVu) ON DELETE SET NULL ON UPDATE CASCADE,

CONSTRAINT FK\_CanBo\_TrinhDo FOREIGN KEY (MaTrinhDo) REFERENCES TrinhDo(MaTrinhDo) ON DELETE SET NULL ON UPDATE CASCADE

);

GO

IF COL\_LENGTH('dbo.CanBo','UserId') IS NULL

BEGIN

ALTER TABLE dbo.CanBo ADD UserId INT NULL;

END

GO

-- đảm bảo mỗi User chỉ gắn cho 1 cán bộ duy nhất

IF NOT EXISTS (SELECT 1 FROM sys.indexes

WHERE name = 'UX\_CanBo\_UserId' AND object\_id = OBJECT\_ID('dbo.CanBo'))

BEGIN

CREATE UNIQUE INDEX UX\_CanBo\_UserId

ON dbo.CanBo(UserId)

WHERE UserId IS NOT NULL; -- cho phép nhiều NULL nhưng không cho trùng UserId

END

GO

-- liên kết với Users

IF NOT EXISTS (SELECT 1 FROM sys.foreign\_keys WHERE name = 'FK\_CanBo\_Users')

BEGIN

ALTER TABLE dbo.CanBo

ADD CONSTRAINT FK\_CanBo\_Users

FOREIGN KEY(UserId) REFERENCES dbo.Users(UserId)

ON DELETE SET NULL ON UPDATE CASCADE;

END

GO

ALTER TABLE CanBo

ADD MaBacLuong NVARCHAR(20) FOREIGN KEY REFERENCES BacLuong(MaBacLuong),

PhuCap DECIMAL(18,2) DEFAULT 0;

-- Đảm bảo 1 trưởng khoa duy nhất: unique filtered index trên (MaKhoa) khi MaChucVu = 'TK';

CREATE UNIQUE INDEX UX\_CanBo\_OneHeadPerKhoa ON CanBo(MaKhoa)

WHERE MaChucVu = 'TK';

GO

**Bảng MonHoc**

CREATE TABLE MonHoc (

MonHocId INT IDENTITY PRIMARY KEY,

MaMon NVARCHAR(20) NOT NULL UNIQUE,

TenMon NVARCHAR(200) NOT NULL DEFAULT N'Chưa đặt tên',

SoTiet INT NOT NULL CHECK (SoTiet >= 0),

SoTinChi INT NOT NULL CHECK (SoTinChi >= 0),

MaKhoaPhuTrach NVARCHAR(20) NOT NULL

);

**Bảng ChuongTrinh\_MonHoc**

CREATE TABLE ChuongTrinh\_MonHoc (

MaCT NVARCHAR(20) NOT NULL,

BatBuoc BIT NOT NULL DEFAULT 1,

MaMon NVARCHAR(20) NOT NULL,

CONSTRAINT FK\_CTMH\_CT FOREIGN KEY (MaCT) REFERENCES ChuongTrinh(MaCT)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FK\_CTMH\_MH FOREIGN KEY (MaMon) REFERENCES MonHoc(MaMon)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT UQ\_CTMH UNIQUE (MaCT, MaMon)

);

**Bảng LopHocPhan**

CREATE TABLE LopHocPhan (

MaLopHocPhan NVARCHAR(20) NOT NULL PRIMARY KEY,

TenLopHocPhan NVARCHAR(200) NOT NULL,

MaKhoa NVARCHAR(20) NULL,

MaNganh NVARCHAR(20) NULL,

MaCT NVARCHAR(20) NULL,

SoLuongSV INT NOT NULL DEFAULT 0,

MaMon NVARCHAR(20) NOT NULL,

CONSTRAINT FK\_Lop\_MaKhoa FOREIGN KEY (MaKhoa) REFERENCES Khoa(MaKhoa) ON DELETE SET NULL ON UPDATE CASCADE,

CONSTRAINT FK\_Lop\_MaNganh FOREIGN KEY (MaNganh) REFERENCES Nganh(MaNganh) ON DELETE SET NULL,

CONSTRAINT FK\_Lop\_MaCT FOREIGN KEY (MaCT) REFERENCES ChuongTrinh(MaCT) ON DELETE SET NULL,

CONSTRAINT FK\_LopHocPhan\_MonHoc FOREIGN KEY (MaMon) REFERENCES MonHoc(MaMon)

);

**Bảng PhanCongGiangDay**

CREATE TABLE PhanCongGiangDay (

MaCB NVARCHAR(20) NOT NULL,

TenMon NVARCHAR(200) NOT NULL,

MaMon NVARCHAR(20) NOT NULL,

MaLopHocPhan NVARCHAR(20) NOT NULL,

SoTiet INT NOT NULL CHECK (SoTiet >= 0),

SoTuan INT NOT NULL DEFAULT 15 CHECK (SoTuan > 0),

HocKy INT NOT NULL, -- HK1

NamHoc NVARCHAR(20) NOT NULL, -- 2024 - 2025

CONSTRAINT FK\_PCGD\_MaCB FOREIGN KEY (MaCB) REFERENCES CanBo(MaCB) ON DELETE NO ACTION ON UPDATE CASCADE,

CONSTRAINT FK\_PCGD\_MaMon FOREIGN KEY (MaMon) REFERENCES MonHoc(MaMon) ON DELETE NO ACTION ON UPDATE CASCADE,

CONSTRAINT FK\_PCGD\_MaLopHocPhan FOREIGN KEY (MaLopHocPhan) REFERENCES LopHocPhan(MaLopHocPhan) ON DELETE NO ACTION,

CONSTRAINT UQ\_PCGD UNIQUE (MaCB, MaMon, MaLopHocPhan, NamHoc, HocKy)

);

ALTER TABLE PhanCongGiangDay

ALTER COLUMN MaNganh NVARCHAR(20) NOT NULL;

ALTER TABLE PhanCongGiangDay

ADD CONSTRAINT FK\_PhanCong\_Nganh FOREIGN KEY (MaNganh)

REFERENCES Nganh(MaNganh);

**Bảng BacLuong**

CREATE TABLE BacLuong (

MaBacLuong NVARCHAR(20) PRIMARY KEY,

HeSoLuong DECIMAL(4,2)

);

**Bảng BangLuong**

CREATE TABLE BangLuong (

MaCB NVARCHAR(20) NOT NULL,

Thang INT,

Nam INT,

Thuong DECIMAL(18,2) DEFAULT 0,

KhauTru DECIMAL(18,2) DEFAULT 0,

CONSTRAINT FK\_BangLuong\_CanBo FOREIGN KEY (MaCB) REFERENCES CanBo(MaCB) ON DELETE CASCADE,

CONSTRAINT UQ\_BangLuong UNIQUE (MaCB, Thang, Nam)

);

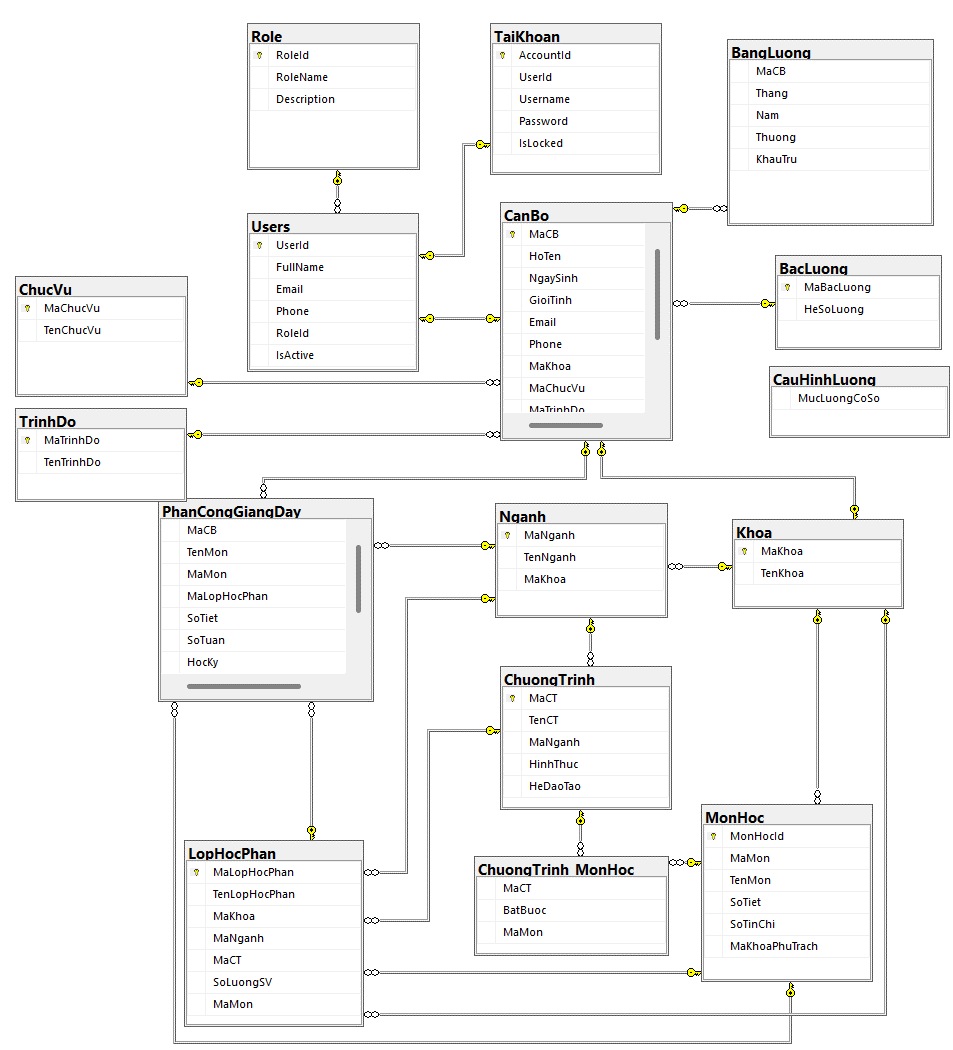
**Bảng CauHinhLuong**

CREATE TABLE CauHinhLuong (

MucLuongCoSo DECIMAL(18,2)

);

1. **Lược đồ quan hệ sau khi cài đặt SQL**



1. **Các View**

View Lương

-- View: Lương giảng viên chi tiết

CREATE OR ALTER VIEW Vw\_CanBo\_Luong

AS

SELECT

bl.MaCB,

cb.HoTen,

bl.Thang,

bl.Nam,

cb.PhuCap,

bl.Thuong,

bl.KhauTru,

b.HeSoLuong,

ch.MucLuongCoSo,

(ch.MucLuongCoSo \* b.HeSoLuong + cb.PhuCap + bl.Thuong - bl.KhauTru) AS TongLuong

FROM BangLuong bl

JOIN CanBo cb ON bl.MaCB = cb.MaCB

JOIN BacLuong b ON cb.MaBacLuong = b.MaBacLuong

CROSS APPLY (SELECT TOP 1 MucLuongCoSo FROM CauHinhLuong) ch;

SELECT \* FROM Vw\_CanBo\_Luong;

-- View: can bo khoa

CREATE OR ALTER VIEW dbo.Vw\_CanBo\_TrongKhoaCuaToi

AS

SELECT cb.MaCB,

cb.HoTen,

cb.MaKhoa,

cb.MaChucVu,

cb.MaTrinhDo,

cb.Email,

cb.Phone,

cb.NgaySinh,

cb.GioiTinh

FROM dbo.CanBo cb

CROSS JOIN dbo.RTO\_CurrentPrincipal() cp

WHERE cp.RoleName IN (N'TruongKhoa', N'GiangVien')

AND cb.MaKhoa = cp.MaKhoa;

-- View: Môn của Khoa

CREATE OR ALTER VIEW dbo.Vw\_MonHoc\_TrongKhoaCuaToi

AS

SELECT mh.MaMon,

mh.TenMon,

mh.MaKhoaPhuTrach

FROM dbo.MonHoc mh

CROSS JOIN dbo.RTO\_CurrentPrincipal() cp

WHERE cp.RoleName IN (N'TruongKhoa', N'GiangVien')

AND mh.MaKhoaPhuTrach = cp.MaKhoa;

GO

-- View: lớp học phần trong khoa

CREATE OR ALTER VIEW dbo.Vw\_LopHP\_TrongKhoaCuaToi

AS

SELECT lhp.MaLopHocPhan,

lhp.MaMon,

lhp.TenLopHocPhan,

lhp.SoLuongSV,

mh.MaKhoaPhuTrach

FROM dbo.LopHocPhan lhp

JOIN dbo.MonHoc mh ON lhp.MaMon = mh.MaMon

CROSS JOIN dbo.RTO\_CurrentPrincipal() cp

WHERE cp.RoleName IN (N'TruongKhoa', N'GiangVien')

AND mh.MaKhoaPhuTrach = cp.MaKhoa;

GO

-- View: Xem phân công của từng giảng viên

CREATE OR ALTER VIEW dbo.Vw\_PhanCong\_CuaToi

AS

SELECT

pc.MaCB,

pc.MaMon,

mh.TenMon,

pc.MaLopHocPhan,

lhp.TenLopHocPhan,

pc.SoTiet,

pc.SoTuan,

pc.HocKy,

pc.NamHoc,

pc.MaNganh

FROM dbo.PhanCongGiangDay pc

JOIN dbo.MonHoc mh ON pc.MaMon = mh.MaMon

JOIN dbo.LopHocPhan lhp ON pc.MaLopHocPhan = lhp.MaLopHocPhan

CROSS JOIN dbo.RTO\_CurrentPrincipal() cp

WHERE pc.MaCB = cp.MaCB;

GO

-- View: Ngành trong khoa

CREATE OR ALTER VIEW dbo.Vw\_Nganh\_TrongKhoaCuaToi

AS

SELECT ng.MaNganh,

ng.TenNganh,

ng.MaKhoa

FROM dbo.Nganh ng

CROSS JOIN dbo.RTO\_CurrentPrincipal() cp

WHERE cp.RoleName IN (N'TruongKhoa', N'GiangVien')

AND ng.MaKhoa = cp.MaKhoa;

GO

-- View: Khoa

CREATE OR ALTER VIEW dbo.Vw\_Khoa\_CuaToi

AS

SELECT DISTINCT k.MaKhoa, k.TenKhoa

FROM dbo.Khoa k

JOIN dbo.CanBo cb ON cb.MaKhoa = k.MaKhoa

CROSS JOIN dbo.RTO\_CurrentPrincipal() cp

WHERE cp.RoleName IN (N'TruongKhoa', N'GiangVien')

AND cb.MaCB = cp.MaCB;

GO

-- View: Lương

CREATE OR ALTER VIEW Vw\_Luong\_CuaToi

AS

SELECT v.\*

FROM Vw\_CanBo\_Luong v

CROSS JOIN RTO\_CurrentPrincipal() cp

WHERE v.MaCB = cp.MaCB;

SELECT \* FROM Vw\_Luong\_CuaToi;

# **CHƯƠNG 3. THIẾT KẾ CÁC CHỨC NĂNG**

1. **Kết nối cơ sở dữ liệu**

Hệ thống sử dụng lớp DataLayer để quản lý kết nối với SQL Server. Chuỗi kết nối mặc định được khai báo với chế độ Integrated Security.

private static string connectionString = "Data Source=.;Initial Catalog=QLCanBoGiangVien;Integrated Security=True;";

public DataAccess(string connStr)

{

connectionString = connStr;

conn = new SqlConnection(connectionString);

cmd = conn.CreateCommand();

}

public static void SetConnection(string username, string password)

{

connectionString = $"Data Source=.;Initial Catalog=QLCanBoGiangVien;" +

$"User ID={username};Password={password};" +

"Encrypt=True;TrustServerCertificate=True;";

}

Ngoài ra còn có các hàm dùng để thực thi

// ====== Dùng cho Stored Procedure ======

public DataTable ExecuteQuery(string spName, params SqlParameter[] parameters)

{

using (SqlConnection conn = new SqlConnection(connectionString))

{

SqlCommand cmd = new SqlCommand(spName, conn);

cmd.CommandType = CommandType.StoredProcedure;

if (parameters != null) cmd.Parameters.AddRange(parameters);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

}

// ====== Dùng cho View hoặc SELECT text ======

public DataTable ExecuteQueryText(string query, params SqlParameter[] parameters)

{

using (SqlConnection conn = new SqlConnection(connectionString))

{

SqlCommand cmd = new SqlCommand(query, conn);

cmd.CommandType = CommandType.Text;

if (parameters != null) cmd.Parameters.AddRange(parameters);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

}

// ====== Dùng cho Insert / Update / Delete ======

public int ExecuteNonQuery(string spName, params SqlParameter[] parameters)

{

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

SqlCommand cmd = new SqlCommand(spName, conn);

cmd.CommandType = CommandType.StoredProcedure;

if (parameters != null) cmd.Parameters.AddRange(parameters);

return cmd.ExecuteNonQuery();

}

}

// ====== Dùng cho Scalar (trả về 1 giá trị duy nhất) ======

public object ExecuteScalar(string query, params SqlParameter[] parameters)

{

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

SqlCommand cmd = new SqlCommand(query, conn);

cmd.CommandType = CommandType.Text;

if (parameters != null) cmd.Parameters.AddRange(parameters);

return cmd.ExecuteScalar();

}

}

1. **Các chức năng của quản lý (Admin)**

Có toàn quyền xử lý dữ liệu

-- ADMIN: toàn quyền schema dbo

GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::dbo TO r\_Admin;

GRANT EXECUTE ON SCHEMA::dbo TO r\_Admin;

1. **Các chức năng của trưởng khoa**
2. Quyền thêm/xóa/sửa/ngành

CREATE OR ALTER PROCEDURE HasP\_InsertNganh

@MaNganh NVARCHAR(20),

@TenNganh NVARCHAR(200),

@MaKhoa NVARCHAR(20)

AS

BEGIN

INSERT INTO Nganh(MaNganh, TenNganh, MaKhoa)

VALUES(@MaNganh, @TenNganh, @MaKhoa);

END;

GO

CREATE OR ALTER PROCEDURE HasP\_UpdateNganh

@MaNganh NVARCHAR(20),

@TenNganh NVARCHAR(200),

@MaKhoa NVARCHAR(20)

AS

BEGIN

UPDATE Nganh

SET TenNganh=@TenNganh, MaKhoa=@MaKhoa

WHERE MaNganh=@MaNganh;

END;

GO

CREATE OR ALTER PROCEDURE HasP\_DeleteNganh

@MaNganh NVARCHAR(20)

AS

BEGIN

DELETE FROM Nganh WHERE MaNganh=@MaNganh;

END;

GO

Xử lý C#:

private void btn\_Them\_Click(object sender, EventArgs e)

{

db.ExecuteNonQuery("HasP\_InsertNganh",

new SqlParameter("@MaNganh", txt\_MaNganh.Text),

new SqlParameter("@TenNganh", txt\_TenNganh.Text),

new SqlParameter("@MaKhoa", cb\_Khoa.SelectedValue) // lấy mã khoa từ combobox

);

LoadNganh(cb\_Khoa.SelectedValue.ToString());

txt\_TenNganh.Clear();

txt\_MaNganh.Clear();

}

private void btn\_Sua\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(\_maNganhSelected)) return;

db.ExecuteNonQuery("HasP\_UpdateNganh",

new SqlParameter("@MaNganh", txt\_MaNganh.Text),

new SqlParameter("@TenNganh", txt\_TenNganh.Text),

new SqlParameter("@MaKhoa", cb\_Khoa.SelectedValue)

);

LoadNganh(cb\_Khoa.SelectedValue.ToString());

txt\_TenNganh.Clear();

txt\_MaNganh.Clear();

}

private void btn\_Xoa\_Click(object sender, EventArgs e)

{

db.ExecuteNonQuery("HasP\_DeleteNganh",

new SqlParameter("@MaNganh", txt\_MaNganh.Text)

);

LoadNganh(cb\_Khoa.SelectedValue.ToString());

txt\_TenNganh.Clear();

txt\_MaNganh.Clear();

}

1. Thêm/xóa/sửa Cán bộ

CREATE OR ALTER PROCEDURE dbo.HasP\_InsertCanBo

@MaCB NVARCHAR(20),

@HoTen NVARCHAR(200),

@NgaySinh DATE,

@GioiTinh NVARCHAR(5),

@Email NVARCHAR(100),

@Phone NVARCHAR(20),

@MaKhoa NVARCHAR(20),

@MaChucVu NVARCHAR(20),

@MaTrinhDo NVARCHAR(20)

AS

BEGIN

-- Kiểm tra nhập thiếu

IF @MaCB IS NULL OR LTRIM(RTRIM(@MaCB)) = ''

RAISERROR(N'Bạn chưa nhập Mã cán bộ.', 16, 1); RETURN;

IF @HoTen IS NULL OR LTRIM(RTRIM(@HoTen)) = ''

RAISERROR(N'Bạn chưa nhập Họ tên.', 16, 1); RETURN;

IF @MaKhoa IS NULL OR LTRIM(RTRIM(@MaKhoa)) = ''

RAISERROR(N'Bạn chưa nhập Mã khoa.', 16, 1); RETURN;

IF @MaChucVu IS NULL OR LTRIM(RTRIM(@MaChucVu)) = ''

RAISERROR(N'Bạn chưa nhập Mã chức vụ.', 16, 1); RETURN;

IF @MaTrinhDo IS NULL OR LTRIM(RTRIM(@MaTrinhDo)) = ''

RAISERROR(N'Bạn chưa nhập Mã trình độ.', 16, 1); RETURN;

DECLARE @role NVARCHAR(50), @mk NVARCHAR(20);

SELECT TOP(1) @role = RoleName, @mk = MaKhoa

FROM dbo.RTO\_CurrentPrincipal();

IF @role <> N'Admin' AND @mk <> @MaKhoa

RAISERROR(N'Không thêm giảng viên ngoài khoa của bạn.', 16, 1);

ELSE

INSERT INTO dbo.CanBo(MaCB, HoTen, NgaySinh, GioiTinh, Email, Phone, MaKhoa, MaChucVu, MaTrinhDo)

VALUES(@MaCB, @HoTen, @NgaySinh, @GioiTinh, @Email, @Phone, @MaKhoa, @MaChucVu, @MaTrinhDo);

END

GO

CREATE OR ALTER PROCEDURE dbo.HasP\_UpdateCanBo

@MaCB NVARCHAR(20),

@HoTen NVARCHAR(200),

@NgaySinh DATE,

@GioiTinh NVARCHAR(5),

@Email NVARCHAR(100),

@Phone NVARCHAR(20),

@MaKhoa NVARCHAR(20),

@MaChucVu NVARCHAR(20),

@MaTrinhDo NVARCHAR(20)

AS

BEGIN

-- Kiểm tra nhập thiếu

IF @MaCB IS NULL OR LTRIM(RTRIM(@MaCB)) = ''

RAISERROR(N'Bạn chưa nhập Mã cán bộ.', 16, 1); RETURN;

IF @HoTen IS NULL OR LTRIM(RTRIM(@HoTen)) = ''

RAISERROR(N'Bạn chưa nhập Họ tên.', 16, 1); RETURN;

IF @MaKhoa IS NULL OR LTRIM(RTRIM(@MaKhoa)) = ''

RAISERROR(N'Bạn chưa nhập Mã khoa.', 16, 1); RETURN;

IF @MaChucVu IS NULL OR LTRIM(RTRIM(@MaChucVu)) = ''

RAISERROR(N'Bạn chưa nhập Mã chức vụ.', 16, 1); RETURN;

IF @MaTrinhDo IS NULL OR LTRIM(RTRIM(@MaTrinhDo)) = ''

RAISERROR(N'Bạn chưa nhập Mã trình độ.', 16, 1); RETURN;

DECLARE @role NVARCHAR(50), @mk NVARCHAR(20), @target NVARCHAR(20);

SELECT TOP(1) @role = RoleName, @mk = MaKhoa

FROM dbo.RTO\_CurrentPrincipal();

SELECT @target = cb.MaKhoa FROM dbo.CanBo cb WHERE cb.MaCB = @MaCB;

IF @target IS NULL

RAISERROR(N'Cán bộ không tồn tại.', 16, 1);

ELSE IF @role <> N'Admin' AND @target <> @mk

RAISERROR(N'Không sửa giảng viên ngoài khoa của bạn.', 16, 1);

ELSE

UPDATE dbo.CanBo

SET HoTen = @HoTen,

NgaySinh = @NgaySinh,

GioiTinh = @GioiTinh,

Email = @Email,

Phone = @Phone,

MaKhoa = @MaKhoa,

MaChucVu = @MaChucVu,

MaTrinhDo = @MaTrinhDo

WHERE MaCB = @MaCB;

END

GO

CREATE OR ALTER PROCEDURE dbo.HasP\_DeleteCanBo

@MaCB NVARCHAR(20)

AS

BEGIN

DECLARE @role NVARCHAR(50), @mk NVARCHAR(20), @target NVARCHAR(20);

SELECT TOP(1) @role = RoleName, @mk = MaKhoa

FROM dbo.RTO\_CurrentPrincipal();

SELECT @target = cb.MaKhoa FROM dbo.CanBo cb WHERE cb.MaCB = @MaCB;

IF @target IS NULL

RAISERROR(N'Cán bộ không tồn tại.', 16, 1);

ELSE IF @role <> N'Admin' AND @target <> @mk

RAISERROR(N'Không xóa giảng viên ngoài khoa của bạn.', 16, 1);

ELSE

DELETE FROM dbo.CanBo WHERE MaCB = @MaCB;

END

GO

Xử lý C#:

private void btn\_Them\_Click(object sender, EventArgs e)

{

db.ExecuteNonQuery("HasP\_InsertCanBo",

new SqlParameter("@MaCB", txt\_MaCB.Text),

new SqlParameter("@HoTen", txt\_HoTen.Text),

new SqlParameter("@NgaySinh", dt\_NgaySinh.Value),

new SqlParameter("@GioiTinh", cb\_GioiTinh.Text),

new SqlParameter("@Email", txt\_Email.Text),

new SqlParameter("@Phone", txt\_Phone.Text),

new SqlParameter("@MaKhoa", txt\_MaKhoa.Text),

new SqlParameter("@MaChucVu", txt\_MaChucVu.Text),

new SqlParameter("@MaTrinhDo", txt\_MaTrinhDo.Text));

LoadGiangVien();

}

private void btn\_Sua\_Click(object sender, EventArgs e)

{

db.ExecuteNonQuery("HasP\_UpdateCanBo",

new SqlParameter("@MaCB", txt\_MaCB.Text),

new SqlParameter("@HoTen", txt\_HoTen.Text),

new SqlParameter("@NgaySinh", dt\_NgaySinh.Value),

new SqlParameter("@GioiTinh", cb\_GioiTinh.Text),

new SqlParameter("@Email", txt\_Email.Text),

new SqlParameter("@Phone", txt\_Phone.Text),

new SqlParameter("@MaKhoa", txt\_MaKhoa.Text),

new SqlParameter("@MaChucVu", txt\_MaChucVu.Text),

new SqlParameter("@MaTrinhDo", txt\_MaTrinhDo.Text));

LoadGiangVien();

}

private void btn\_Xoa\_Click(object sender, EventArgs e)

{

db.ExecuteNonQuery("HasP\_DeleteCanBo",

new SqlParameter("@MaCB", txt\_MaCB.Text)

);

LoadGiangVien();

}

1. Thêm/Xóa/Sửa Môn học

CREATE OR ALTER PROCEDURE dbo.HasP\_InsertCanBo

@MaCB NVARCHAR(20),

@HoTen NVARCHAR(200),

@NgaySinh DATE,

@GioiTinh NVARCHAR(5),

@Email NVARCHAR(100),

@Phone NVARCHAR(20),

@MaKhoa NVARCHAR(20),

@MaChucVu NVARCHAR(20),

@MaTrinhDo NVARCHAR(20)

AS

BEGIN

-- Kiểm tra nhập thiếu

IF @MaCB IS NULL OR LTRIM(RTRIM(@MaCB)) = ''

RAISERROR(N'Bạn chưa nhập Mã cán bộ.', 16, 1); RETURN;

IF @HoTen IS NULL OR LTRIM(RTRIM(@HoTen)) = ''

RAISERROR(N'Bạn chưa nhập Họ tên.', 16, 1); RETURN;

IF @MaKhoa IS NULL OR LTRIM(RTRIM(@MaKhoa)) = ''

RAISERROR(N'Bạn chưa nhập Mã khoa.', 16, 1); RETURN;

IF @MaChucVu IS NULL OR LTRIM(RTRIM(@MaChucVu)) = ''

RAISERROR(N'Bạn chưa nhập Mã chức vụ.', 16, 1); RETURN;

IF @MaTrinhDo IS NULL OR LTRIM(RTRIM(@MaTrinhDo)) = ''

RAISERROR(N'Bạn chưa nhập Mã trình độ.', 16, 1); RETURN;

DECLARE @role NVARCHAR(50), @mk NVARCHAR(20);

SELECT TOP(1) @role = RoleName, @mk = MaKhoa

FROM dbo.RTO\_CurrentPrincipal();

IF @role <> N'Admin' AND @mk <> @MaKhoa

RAISERROR(N'Không thêm giảng viên ngoài khoa của bạn.', 16, 1);

ELSE

INSERT INTO dbo.CanBo(MaCB, HoTen, NgaySinh, GioiTinh, Email, Phone, MaKhoa, MaChucVu, MaTrinhDo)

VALUES(@MaCB, @HoTen, @NgaySinh, @GioiTinh, @Email, @Phone, @MaKhoa, @MaChucVu, @MaTrinhDo);

END

GO

CREATE OR ALTER PROCEDURE dbo.HasP\_UpdateCanBo

@MaCB NVARCHAR(20),

@HoTen NVARCHAR(200),

@NgaySinh DATE,

@GioiTinh NVARCHAR(5),

@Email NVARCHAR(100),

@Phone NVARCHAR(20),

@MaKhoa NVARCHAR(20),

@MaChucVu NVARCHAR(20),

@MaTrinhDo NVARCHAR(20)

AS

BEGIN

-- Kiểm tra nhập thiếu

IF @MaCB IS NULL OR LTRIM(RTRIM(@MaCB)) = ''

RAISERROR(N'Bạn chưa nhập Mã cán bộ.', 16, 1); RETURN;

IF @HoTen IS NULL OR LTRIM(RTRIM(@HoTen)) = ''

RAISERROR(N'Bạn chưa nhập Họ tên.', 16, 1); RETURN;

IF @MaKhoa IS NULL OR LTRIM(RTRIM(@MaKhoa)) = ''

RAISERROR(N'Bạn chưa nhập Mã khoa.', 16, 1); RETURN;

IF @MaChucVu IS NULL OR LTRIM(RTRIM(@MaChucVu)) = ''

RAISERROR(N'Bạn chưa nhập Mã chức vụ.', 16, 1); RETURN;

IF @MaTrinhDo IS NULL OR LTRIM(RTRIM(@MaTrinhDo)) = ''

RAISERROR(N'Bạn chưa nhập Mã trình độ.', 16, 1); RETURN;

DECLARE @role NVARCHAR(50), @mk NVARCHAR(20), @target NVARCHAR(20);

SELECT TOP(1) @role = RoleName, @mk = MaKhoa

FROM dbo.RTO\_CurrentPrincipal();

SELECT @target = cb.MaKhoa FROM dbo.CanBo cb WHERE cb.MaCB = @MaCB;

IF @target IS NULL

RAISERROR(N'Cán bộ không tồn tại.', 16, 1);

ELSE IF @role <> N'Admin' AND @target <> @mk

RAISERROR(N'Không sửa giảng viên ngoài khoa của bạn.', 16, 1);

ELSE

UPDATE dbo.CanBo

SET HoTen = @HoTen,

NgaySinh = @NgaySinh,

GioiTinh = @GioiTinh,

Email = @Email,

Phone = @Phone,

MaKhoa = @MaKhoa,

MaChucVu = @MaChucVu,

MaTrinhDo = @MaTrinhDo

WHERE MaCB = @MaCB;

END

CREATE OR ALTER PROCEDURE dbo.HasP\_DeleteCanBo

@MaCB NVARCHAR(20)

AS

BEGIN

DECLARE @role NVARCHAR(50), @mk NVARCHAR(20), @target NVARCHAR(20);

SELECT TOP(1) @role = RoleName, @mk = MaKhoa

FROM dbo.RTO\_CurrentPrincipal();

SELECT @target = cb.MaKhoa FROM dbo.CanBo cb WHERE cb.MaCB = @MaCB;

IF @target IS NULL

RAISERROR(N'Cán bộ không tồn tại.', 16, 1);

ELSE IF @role <> N'Admin' AND @target <> @mk

RAISERROR(N'Không xóa giảng viên ngoài khoa của bạn.', 16, 1);

ELSE

DELETE FROM dbo.CanBo WHERE MaCB = @MaCB;

END

Xử lý c#:

private void btn\_Them\_Click(object sender, EventArgs e)

{

db.ExecuteNonQuery("HasP\_InsertMonHoc",

new SqlParameter("@MaMon", txt\_MaMon.Text),

new SqlParameter("@TenMon", txt\_TenMon.Text),

new SqlParameter("@SoTiet", int.Parse(txt\_SoTiet.Text)),

new SqlParameter("@SoTinChi", int.Parse(txt\_SoTinChi.Text)),

new SqlParameter("@MaCT", \_maCT),

new SqlParameter("@BatBuoc", chk\_BatBuoc.Checked),

new SqlParameter("@MaKhoaPhuTrach", cb\_TenKhoa.SelectedValue));

LoadMon(\_maCT);

Clear\_Inputs();

}

private void btn\_Sua\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(\_maMonSelected)) return;

db.ExecuteNonQuery("HasP\_UpdateMonHoc",

new SqlParameter("@MaMon", txt\_MaMon.Text),

new SqlParameter("@TenMon", txt\_TenMon.Text),

new SqlParameter("@SoTiet", int.Parse(txt\_SoTiet.Text)),

new SqlParameter("@SoTinChi", int.Parse(txt\_SoTinChi.Text)),

new SqlParameter("@MaCT", \_maCT),

new SqlParameter("@BatBuoc", chk\_BatBuoc.Checked),

new SqlParameter("@MaKhoaPhuTrach", cb\_TenKhoa.SelectedValue));

LoadMon(\_maCT);

Clear\_Inputs();

}

private void btn\_Xoa\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(\_maMonSelected)) return;

db.ExecuteNonQuery("HasP\_DeleteMonHoc",

new SqlParameter("@MaMon", \_maMonSelected),

new SqlParameter("@MaCT", \_maCT));

LoadMon(\_maCT);

Clear\_Inputs();

}

1. Thêm/ xóa /sửa lớp học phần

CREATE OR ALTER PROCEDURE dbo.HasP\_InsertLopHocPhan

@MaMon NVARCHAR(20),

@SoThuTu INT,

@SoLuongSV INT

AS

BEGIN

DECLARE @role NVARCHAR(50), @mk NVARCHAR(20),

@khoaMon NVARCHAR(20), @tenMon NVARCHAR(200),

@MaLop NVARCHAR(32), @TenLop NVARCHAR(250);

SELECT TOP(1) @role=RoleName, @mk=MaKhoa FROM dbo.RTO\_CurrentPrincipal();

SELECT @khoaMon = MaKhoaPhuTrach, @tenMon = TenMon

FROM dbo.MonHoc WHERE MaMon = @MaMon;

IF @khoaMon IS NULL

RAISERROR(N'Môn học không tồn tại.',16,1);

ELSE IF @role <> N'Admin' AND @khoaMon <> @mk

RAISERROR(N'Không thêm lớp ngoài khoa của bạn.',16,1);

ELSE

BEGIN

-- Mã lớp dạng MaMon\_XX (01, 02, 03, ...)

SET @MaLop = @MaMon + N'\_' + RIGHT('00' + CAST(@SoThuTu AS NVARCHAR(2)),2);

-- Tên lớp = tên môn (không thêm đuôi nhóm)

SET @TenLop = @tenMon;

IF EXISTS (SELECT 1 FROM dbo.LopHocPhan WHERE MaLopHocPhan = @MaLop)

RAISERROR(N'Mã lớp đã tồn tại.',16,1);

ELSE

INSERT dbo.LopHocPhan(MaLopHocPhan, TenLopHocPhan, MaMon, SoLuongSV)

VALUES(@MaLop, @TenLop, @MaMon, @SoLuongSV);

END

END

GO

CREATE OR ALTER PROCEDURE dbo.HasP\_UpdateLopHocPhan

@OldMaLopHocPhan NVARCHAR(20),

@SoThuTu INT,

@SoLuongSV INT

AS

BEGIN

DECLARE @role NVARCHAR(50), @mk NVARCHAR(20),

@targetKhoa NVARCHAR(20),

@MaMon NVARCHAR(20), @TenMon NVARCHAR(200),

@NewMaLop NVARCHAR(32), @NewTenLop NVARCHAR(250);

SELECT TOP(1) @role=RoleName, @mk=MaKhoa

FROM dbo.RTO\_CurrentPrincipal();

-- Lấy thông tin lớp cũ

SELECT lhp.MaMon, mh.TenMon, mh.MaKhoaPhuTrach

INTO #tmp

FROM dbo.LopHocPhan lhp

JOIN dbo.MonHoc mh ON lhp.MaMon = mh.MaMon

WHERE lhp.MaLopHocPhan = @OldMaLopHocPhan;

IF NOT EXISTS (SELECT 1 FROM #tmp)

BEGIN

RAISERROR(N'Lớp học phần không tồn tại.',16,1);

RETURN;

END

SELECT @MaMon=MaMon, @TenMon=TenMon, @targetKhoa=MaKhoaPhuTrach FROM #tmp;

DROP TABLE #tmp;

IF @role <> N'Admin' AND @targetKhoa <> @mk

RAISERROR(N'Không sửa lớp ngoài khoa của bạn.',16,1);

ELSE

BEGIN

-- Sinh mã lớp mới theo số thứ tự

SET @NewMaLop = @MaMon + N'\_' + RIGHT('00' + CAST(@SoThuTu AS NVARCHAR(2)),2);

-- Tên lớp = tên môn (không đổi)

SET @NewTenLop = @TenMon;

-- Nếu mã mới trùng lớp khác thì báo lỗi

IF EXISTS (SELECT 1 FROM dbo.LopHocPhan WHERE MaLopHocPhan=@NewMaLop AND MaLopHocPhan<>@OldMaLopHocPhan)

RAISERROR(N'Mã lớp mới đã tồn tại.',16,1);

ELSE

UPDATE dbo.LopHocPhan

SET MaLopHocPhan = @NewMaLop,

TenLopHocPhan = @NewTenLop,

SoLuongSV = @SoLuongSV

WHERE MaLopHocPhan = @OldMaLopHocPhan;

END

END

GO

CREATE OR ALTER PROCEDURE dbo.HasP\_DeleteLopHocPhan

@MaLopHocPhan NVARCHAR(20)

AS

BEGIN

DECLARE @role NVARCHAR(50), @mk NVARCHAR(20), @target NVARCHAR(20);

SELECT TOP(1) @role=RoleName, @mk=MaKhoa FROM dbo.RTO\_CurrentPrincipal();

SELECT @target = mh.MaKhoaPhuTrach

FROM dbo.LopHocPhan lhp

JOIN dbo.MonHoc mh ON lhp.MaMon = mh.MaMon

WHERE lhp.MaLopHocPhan = @MaLopHocPhan;

IF @target IS NULL

RAISERROR(N'Lớp học phần không tồn tại.',16,1);

ELSE IF @role <> N'Admin' AND @target <> @mk

RAISERROR(N'Không xóa lớp ngoài khoa của bạn.',16,1);

ELSE

DELETE FROM dbo.LopHocPhan WHERE MaLopHocPhan = @MaLopHocPhan;

END

Go

Xử lý C#:

private void btn\_Them\_Click(object sender, EventArgs e)

{

if (cb\_TenMon.SelectedValue == null || !int.TryParse(txt\_MaSoLop.Text, out var soThuTu) || !int.TryParse(txt\_SoLuongSV.Text, out var soLuong))

{

MessageBox.Show("Vui lòng chọn môn và nhập Số thứ tự / Số lượng SV hợp lệ!");

return;

}

try

{

db.ExecuteNonQuery("HasP\_InsertLopHocPhan",

new SqlParameter("@MaMon", (string)cb\_TenMon.SelectedValue),

new SqlParameter("@SoThuTu", soThuTu),

new SqlParameter("@SoLuongSV", soLuong)

);

LoadLopHocPhanByMon((string)cb\_TenMon.SelectedValue);

txt\_MaSoLop.Clear();

txt\_SoLuongSV.Clear();

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btn\_Sua\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(\_maLopSelected) || string.IsNullOrEmpty(txt\_MaSoLop.Text) || string.IsNullOrEmpty(txt\_SoLuongSV.Text))

{

MessageBox.Show("Vui lòng chọn lớp học phần để sửa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

try

{

db.ExecuteNonQuery("HasP\_UpdateLopHocPhan",

new SqlParameter("@OldMaLopHocPhan", \_maLopSelected),

new SqlParameter("@SoThuTu", int.Parse(txt\_MaSoLop.Text)),

new SqlParameter("@SoLuongSV", int.Parse(txt\_SoLuongSV.Text))

);

LoadLopHocPhanByMon(txt\_MaMon.Text);

txt\_MaSoLop.Clear();

txt\_SoLuongSV.Clear();

MessageBox.Show("Cập nhật lớp học phần thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show("Lỗi cập nhật lớp học phần: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btn\_Xoa\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(\_maLopSelected))

{

MessageBox.Show("Vui lòng chọn lớp học phần để xóa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

try

{

db.ExecuteNonQuery("HasP\_DeleteLopHocPhan",

new SqlParameter("@MaLopHocPhan", \_maLopSelected)

);

LoadLopHocPhanByMon(txt\_MaMon.Text);

txt\_MaSoLop.Clear();

txt\_SoLuongSV.Clear();

MessageBox.Show("Xóa lớp học phần thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show("Lỗi xóa lớp học phần: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

1. Phân công giảng dạy cho giảng viên khoa

-- Thêm phân công

CREATE OR ALTER PROCEDURE dbo.HasP\_InsertPhanCong

@MaLopHocPhan NVARCHAR(20),

@MaCB NVARCHAR(20),

@MaMon NVARCHAR(20),

@SoTiet INT,

@SoTuan INT,

@HocKy INT,

@NamHoc NVARCHAR(20),

@MaNganh NVARCHAR(20)

AS

BEGIN

DECLARE @role NVARCHAR(50), @mk NVARCHAR(20), @target NVARCHAR(20), @TenMon NVARCHAR(200);

SELECT TOP(1) @role=RoleName,@mk=MaKhoa FROM dbo.RTO\_CurrentPrincipal();

SELECT @target=cb.MaKhoa FROM dbo.CanBo cb WHERE cb.MaCB=@MaCB;

SELECT @TenMon=TenMon FROM dbo.MonHoc WHERE MaMon=@MaMon;

IF @role<>N'Admin' AND @target<>@mk

RAISERROR(N'Không phân công giảng viên ngoài khoa.',16,1);

ELSE

INSERT INTO dbo.PhanCongGiangDay

(MaLopHocPhan, MaCB, MaNganh, MaMon, TenMon, SoTiet, SoTuan, HocKy, NamHoc)

VALUES(@MaLopHocPhan,@MaCB,@MaNganh,@MaMon,@TenMon,@SoTiet,@SoTuan,@HocKy,@NamHoc);

END

GO

CREATE OR ALTER PROCEDURE dbo.HasP\_UpdatePhanCong

@OldMaCB NVARCHAR(20),

@OldMaMon NVARCHAR(20),

@OldMaLopHocPhan NVARCHAR(20),

@OldHocKy INT,

@OldNamHoc NVARCHAR(9),

@NewMaMon NVARCHAR(20),

@NewMaLopHocPhan NVARCHAR(20),

@NewSoTiet INT,

@NewSoTuan INT,

@NewHocKy INT,

@NewNamHoc NVARCHAR(9),

@NewMaNganh NVARCHAR(20) -- thêm ngành

AS

BEGIN

DECLARE @role NVARCHAR(50), @mk NVARCHAR(20), @target NVARCHAR(20), @TenMon NVARCHAR(200);

SELECT TOP(1) @role=RoleName, @mk=MaKhoa FROM dbo.RTO\_CurrentPrincipal();

SELECT @target=cb.MaKhoa FROM dbo.CanBo cb WHERE cb.MaCB=@OldMaCB;

SELECT @TenMon=TenMon FROM dbo.MonHoc WHERE MaMon=@NewMaMon;

IF @target IS NULL

RAISERROR(N'Giảng viên mới không tồn tại.',16,1);

ELSE IF @role<>N'Admin' AND @target<>@mk

RAISERROR(N'Không phân công giảng viên ngoài khoa.',16,1);

ELSE

UPDATE dbo.PhanCongGiangDay

SET MaLopHocPhan=@NewMaLopHocPhan,

MaNganh=@NewMaNganh,

MaMon=@NewMaMon,

TenMon=@TenMon,

-- TenMon=(SELECT TenMon FROM MonHoc WHERE MaMon=@NewMaMon),

SoTiet=@NewSoTiet,

SoTuan=@NewSoTuan,

HocKy=@NewHocKy,

NamHoc=@NewNamHoc

WHERE MaLopHocPhan=@OldMaLopHocPhan

AND MaCB=@OldMaCB

AND HocKy=@OldHocKy

AND NamHoc=@OldNamHoc

--AND MaMon=@OldMaMon

;

END

GO

CREATE OR ALTER PROCEDURE dbo.HasP\_DeletePhanCong

@MaLopHocPhan NVARCHAR(20),

@MaCB NVARCHAR(20),

@HocKy INT,

@NamHoc NVARCHAR(20),

@MaMon NVARCHAR(20)

AS

BEGIN

DECLARE @role NVARCHAR(50), @mk NVARCHAR(20), @target NVARCHAR(20);

SELECT TOP(1) @role=RoleName, @mk=MaKhoa FROM dbo.RTO\_CurrentPrincipal();

SELECT @target=cb.MaKhoa

FROM dbo.CanBo cb

WHERE cb.MaCB=@MaCB;

IF @target IS NULL

RAISERROR(N'Giảng viên không tồn tại.',16,1);

ELSE IF @role<>N'Admin' AND @target<>@mk

RAISERROR(N'Không xóa phân công ngoài khoa.',16,1);

ELSE

DELETE FROM dbo.PhanCongGiangDay

WHERE MaLopHocPhan=@MaLopHocPhan AND MaCB=@MaCB

AND HocKy=@HocKy AND NamHoc=@NamHoc;

END

GO

1. **Các chức năng của giảng viên**

Giảng viên chỉ có quyền xem và không thể thao tác dữ liệu

-- Giảng viên: chỉ SELECT view an toàn

GRANT SELECT ON dbo.Vw\_CanBo\_TrongKhoaCuaToi TO r\_GiangVien;

GRANT SELECT ON dbo.Vw\_Nganh\_TrongKhoaCuaToi TO r\_GiangVien;

GRANT SELECT ON dbo.Vw\_MonHoc\_TrongKhoaCuaToi TO r\_GiangVien;

GRANT SELECT ON dbo.Vw\_LopHP\_TrongKhoaCuaToi TO r\_GiangVien;

GRANT SELECT ON dbo.Vw\_LopHP\_CuaToi TO r\_GiangVien;

GRANT SELECT ON dbo.Vw\_PhanCong\_CuaToi TO r\_GiangVien;

GRANT SELECT ON dbo.Vw\_CanBo\_Luong TO r\_GiangVien;

GRANT EXECUTE ON dbo.HasP\_GetCanBoByName TO r\_GiangVien;

GRANT SELECT ON dbo.Vw\_Khoa\_CuaToi TO r\_GiangVien;

GRANT EXECUTE ON dbo.HasP\_GetChuongTrinhByNganh TO r\_GiangVien;

GRANT EXECUTE ON dbo.HasP\_GetMonHocByCT TO r\_GiangVien;

GRANT EXECUTE ON dbo.HasP\_SearchMonHoc TO r\_GiangVien;

GRANT EXECUTE ON dbo.HasP\_GetLopHocPhanByMon TO r\_GiangVien;

GRANT EXECUTE ON dbo.HasP\_GetMonHocByNganh TO r\_GiangVien;

GRANT EXECUTE ON dbo.HasP\_GetNganhByKhoa TO r\_GiangVien;

GRANT EXECUTE ON dbo.HasP\_GetMonHocByNganh\_PC TO r\_GiangVien;

GRANT EXECUTE ON dbo.HasP\_GetSoTietByMon TO r\_GiangVien;

GRANT EXECUTE ON dbo.HasP\_GetLopChuaPhanCong TO r\_GiangVien;

GRANT SELECT ON dbo.Vw\_Luong\_CuaToi TO r\_GiangVien;

# **CHƯƠNG 4. TẠO USER VÀ PHÂN QUYỀN**

1. **Tạo role và phân quyền**

Phần mềm hoạt động với 3 role chính: Admin, Trưởng Khoa, Giảng Viên

\*=========================

1. PHÂN QUYỀN

=========================\*/

USE QLCanBoGiangVien;

GO

IF NOT EXISTS (SELECT 1 FROM sys.database\_principals WHERE name = N'r\_Admin')

CREATE ROLE r\_Admin;

IF NOT EXISTS (SELECT 1 FROM sys.database\_principals WHERE name = N'r\_TruongKhoa')

CREATE ROLE r\_TruongKhoa;

IF NOT EXISTS (SELECT 1 FROM sys.database\_principals WHERE name = N'r\_GiangVien')

CREATE ROLE r\_GiangVien;

GO

-- Hàm lấy danh tính hiện tại (role, MaCB, MaKhoa) Dùng USER\_NAME() để map sang TaiKhoan.Username

CREATE OR ALTER FUNCTION dbo.RTO\_CurrentPrincipal()

RETURNS TABLE

AS

RETURN

(

SELECT t.UserId,

u.FullName,

r.RoleName,

cb.MaCB,

cb.MaKhoa

FROM dbo.TaiKhoan t

JOIN dbo.Users u ON u.UserId = t.UserId

JOIN dbo.Role r ON r.RoleId = u.RoleId

LEFT JOIN dbo.CanBo cb ON cb.UserId = u.UserId

WHERE t.Username = USER\_NAME()

);

GO

1. **Đồng bộ hóa CRUD bảng TAIKHOAN với Login/User của SQL Server**

Tài khoản các user được gán với role để có quyền khi đăng nhập

/\* SAU KHI THÊM TÀI KHOẢN -> TỰ TẠO LOGIN/USER + ADD ROLE \*/

CREATE OR ALTER TRIGGER TRG\_TaiKhoan\_AI

ON dbo.TaiKhoan

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

DECLARE @u NVARCHAR(100), @p NVARCHAR(255), @role NVARCHAR(50), @sql NVARCHAR(MAX);

SELECT TOP(1)

@u = i.Username,

@p = i.Password,

@role = r.RoleName

FROM inserted i

JOIN dbo.Users u ON u.UserId = i.UserId

JOIN dbo.Role r ON r.RoleId = u.RoleId;

-- Tạo/đổi mật khẩu LOGIN

IF SUSER\_ID(@u) IS NULL

SET @sql = N'CREATE LOGIN ' + QUOTENAME(@u) +

N' WITH PASSWORD = ''' + REPLACE(@p,'''','''''') + N''', CHECK\_POLICY=OFF, CHECK\_EXPIRATION=OFF;';

ELSE

SET @sql = N'ALTER LOGIN ' + QUOTENAME(@u) +

N' WITH PASSWORD = ''' + REPLACE(@p,'''','''''') + N''', CHECK\_POLICY=OFF;';

EXEC(@sql);

-- Tạo USER trong database (nếu chưa có)

IF USER\_ID(@u) IS NULL

BEGIN

SET @sql = N'CREATE USER ' + QUOTENAME(@u) + N' FOR LOGIN ' + QUOTENAME(@u) + N';';

EXEC(@sql);

END

-- Add vào ROLE

IF @role = N'Admin'

SET @sql = N'ALTER ROLE r\_Admin ADD MEMBER ' + QUOTENAME(@u) + N';';

ELSE IF @role = N'TruongKhoa'

SET @sql = N'ALTER ROLE r\_TruongKhoa ADD MEMBER ' + QUOTENAME(@u) + N';';

ELSE IF @role = N'GiangVien'

SET @sql = N'ALTER ROLE r\_GiangVien ADD MEMBER ' + QUOTENAME(@u) + N';';

ELSE

SET @sql = NULL;

IF @sql IS NOT NULL EXEC(@sql);

END

GO

/\* XÓA TÀI KHOẢN -> DROP USER + LOGIN + GỠ ROLE \*/

CREATE OR ALTER TRIGGER TRG\_TaiKhoan\_AD

ON dbo.TaiKhoan

AFTER DELETE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @u NVARCHAR(100), @sql NVARCHAR(MAX);

SELECT TOP(1) @u = d.Username FROM deleted d;

-- Gỡ khỏi mọi role (thử từng role)

SET @sql = N'IF IS\_ROLEMEMBER(''r\_Admin'',''' + REPLACE(@u,'''','''''') + N''')=1 ALTER ROLE r\_Admin DROP MEMBER ' + QUOTENAME(@u) + N';';

EXEC(@sql);

SET @sql = N'IF IS\_ROLEMEMBER(''r\_TruongKhoa'',''' + REPLACE(@u,'''','''''') + N''')=1 ALTER ROLE r\_TruongKhoa DROP MEMBER ' + QUOTENAME(@u) + N';';

EXEC(@sql);

SET @sql = N'IF IS\_ROLEMEMBER(''r\_GiangVien'',''' + REPLACE(@u,'''','''''') + N''')=1 ALTER ROLE r\_GiangVien DROP MEMBER ' + QUOTENAME(@u) + N';';

EXEC(@sql);

-- Drop USER

IF USER\_ID(@u) IS NOT NULL

BEGIN

SET @sql = N'DROP USER ' + QUOTENAME(@u) + N';';

EXEC(@sql);

END

-- Drop LOGIN

IF SUSER\_ID(@u) IS NOT NULL

BEGIN

SET @sql = N'DROP LOGIN ' + QUOTENAME(@u) + N';';

EXEC(@sql);

END

END

GO

/\* CẬP NHẬT TÊN/MẬT KHẨU -> ĐỔI LOGIN/USER TƯƠNG ỨNG \*/

CREATE OR ALTER TRIGGER TRG\_TaiKhoan\_AU

ON dbo.TaiKhoan

AFTER UPDATE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @oldU NVARCHAR(100), @newU NVARCHAR(100), @newP NVARCHAR(255), @sql NVARCHAR(MAX);

SELECT TOP(1)

@oldU = d.Username, @newU = i.Username, @newP = i.Password

FROM inserted i CROSS JOIN deleted d;

IF UPDATE(Username) AND @oldU <> @newU

BEGIN

SET @sql = N'ALTER LOGIN ' + QUOTENAME(@oldU) + N' WITH NAME = ' + QUOTENAME(@newU) + N';';

EXEC(@sql);

IF USER\_ID(@oldU) IS NOT NULL

BEGIN

SET @sql = N'ALTER USER ' + QUOTENAME(@oldU) + N' WITH NAME = ' + QUOTENAME(@newU) + N';';

EXEC(@sql);

END

END

IF UPDATE(Password)

BEGIN

DECLARE @target NVARCHAR(100) = CASE WHEN UPDATE(Username) AND @oldU <> @newU THEN @newU ELSE @oldU END;

SET @sql = N'ALTER LOGIN ' + QUOTENAME(@target) + N' WITH PASSWORD = ''' + REPLACE(@newP,'''','''''') + N''', CHECK\_POLICY=OFF;';

EXEC(@sql);

END

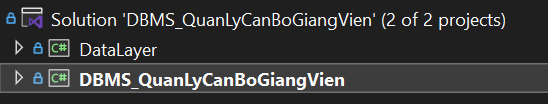
END

# **CHƯƠNG 5. THIẾT KẾ GIAO DIỆN HỆ THỐNG**

1. **Kiến trúc hệ thống**

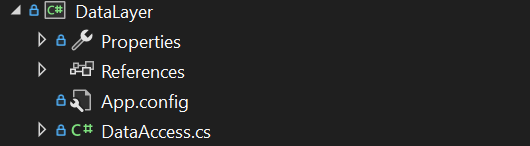
Môi trường cài đặt và công nghệ sử dụng

* Hệ cơ sở dữ liệu Microsoft SQL Server Management Studio 19.
* Chương trình Winform được viết bằng (.NET Framework) C#.
* Công cụ lập trình: Microsoft Visual Studio 2022.
  1. **Cấu trúc thư mục hệ thống**

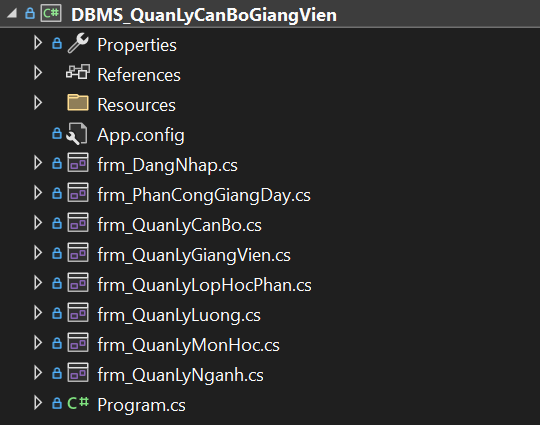


* 1. **Mô hình dữ liệu (2 layer)**

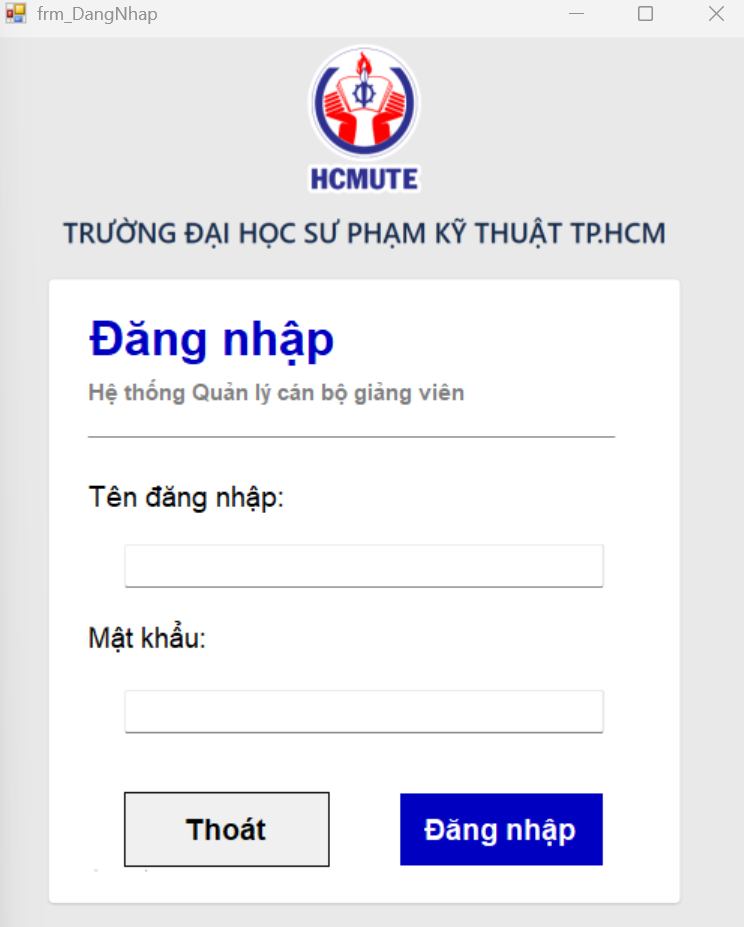
Tầng DataLayer



Tầng Application



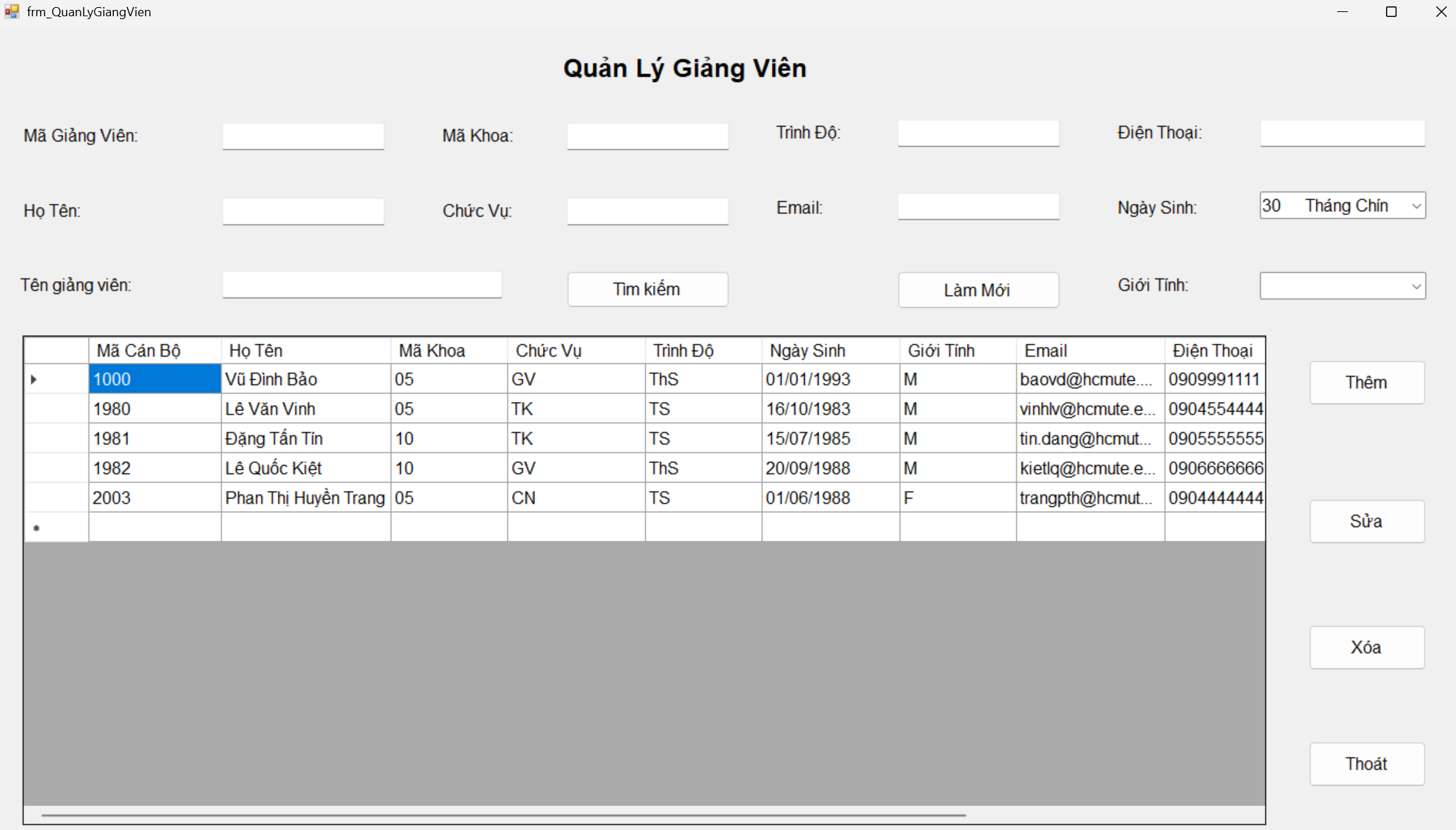
1. **Giao diện và chức năng của chương trình**
   1. Đăng nhập



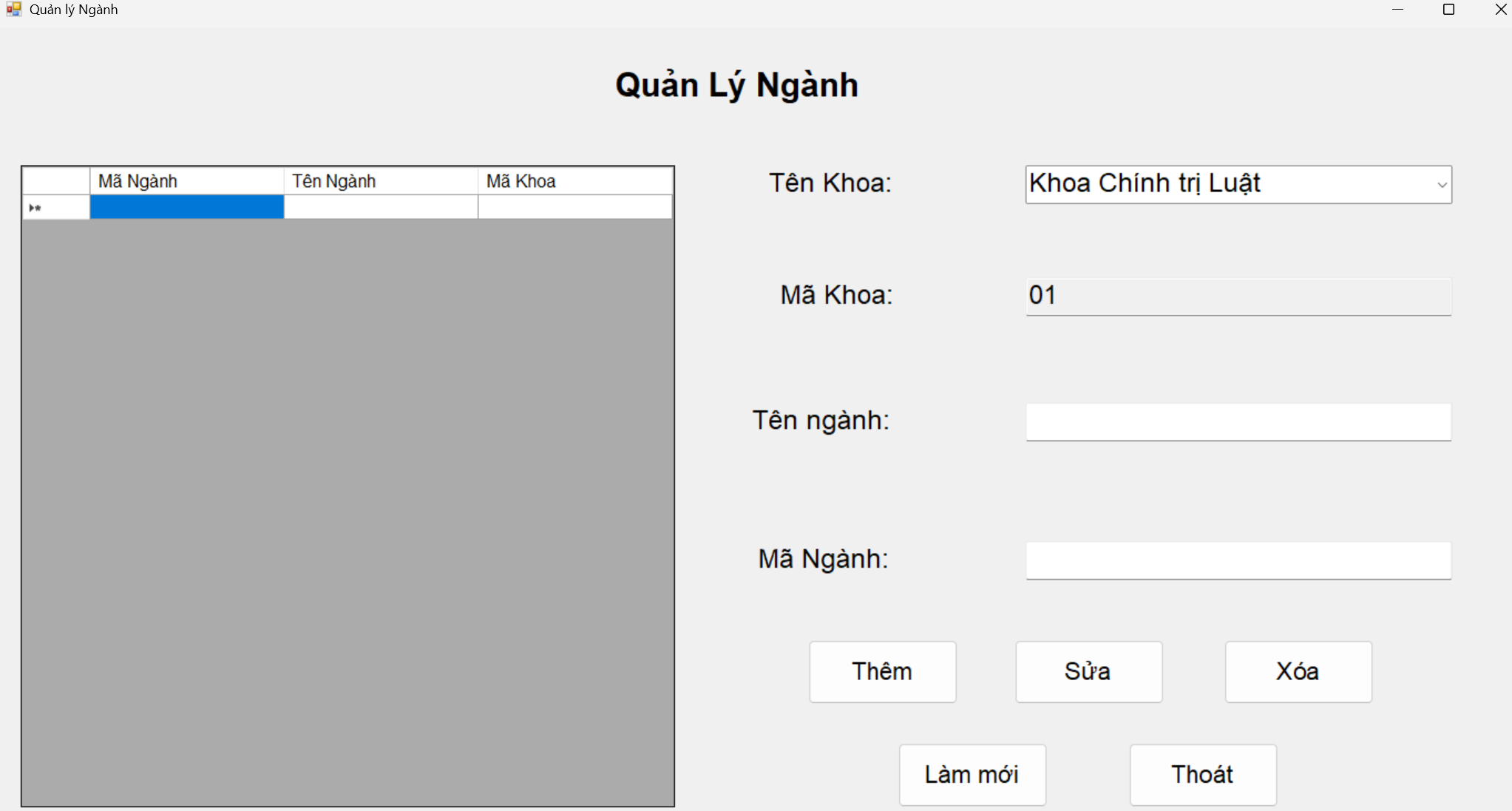
* 1. Giao diện chính



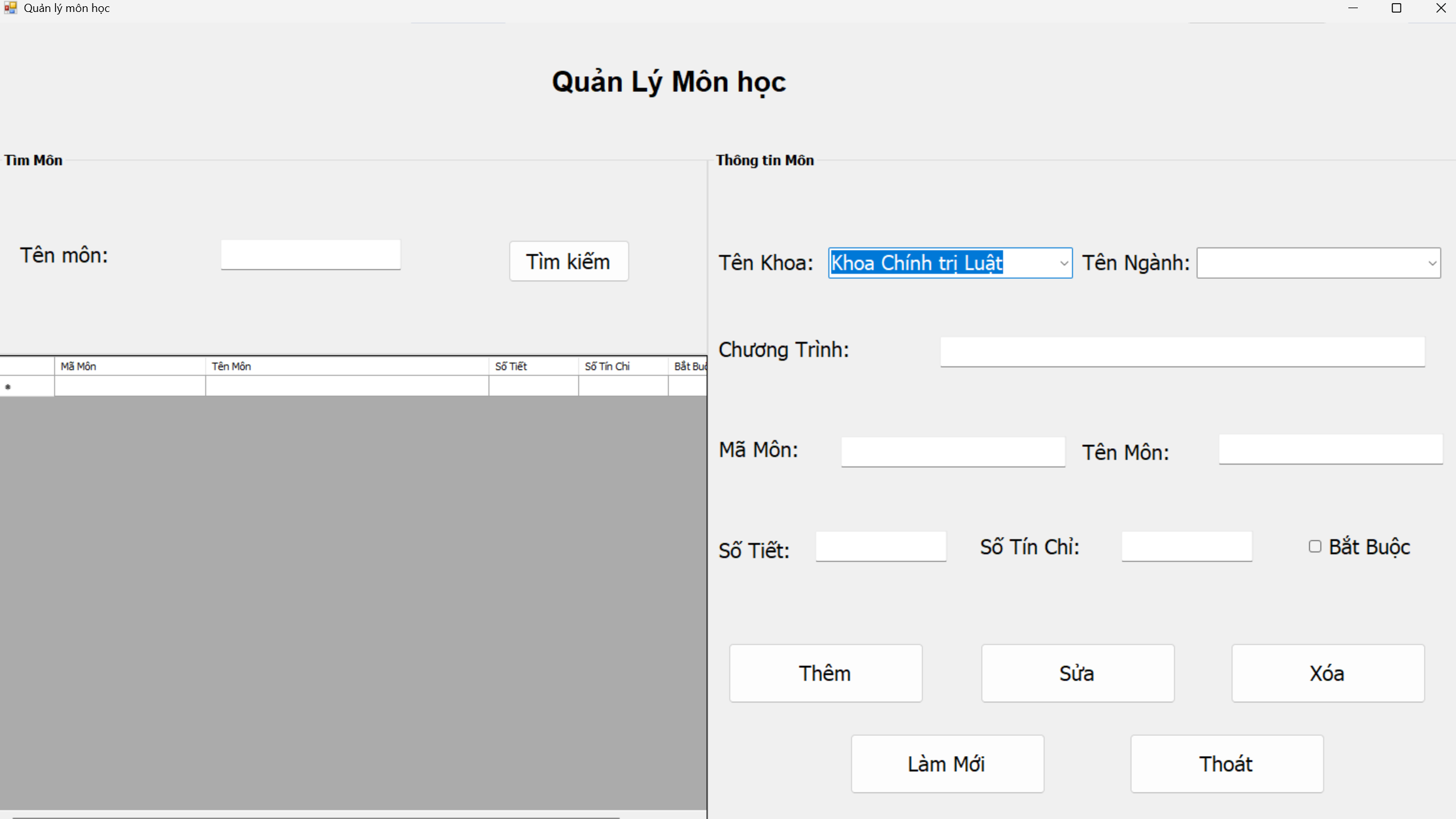
* 1. Giao diện quản lý Giảng Viên



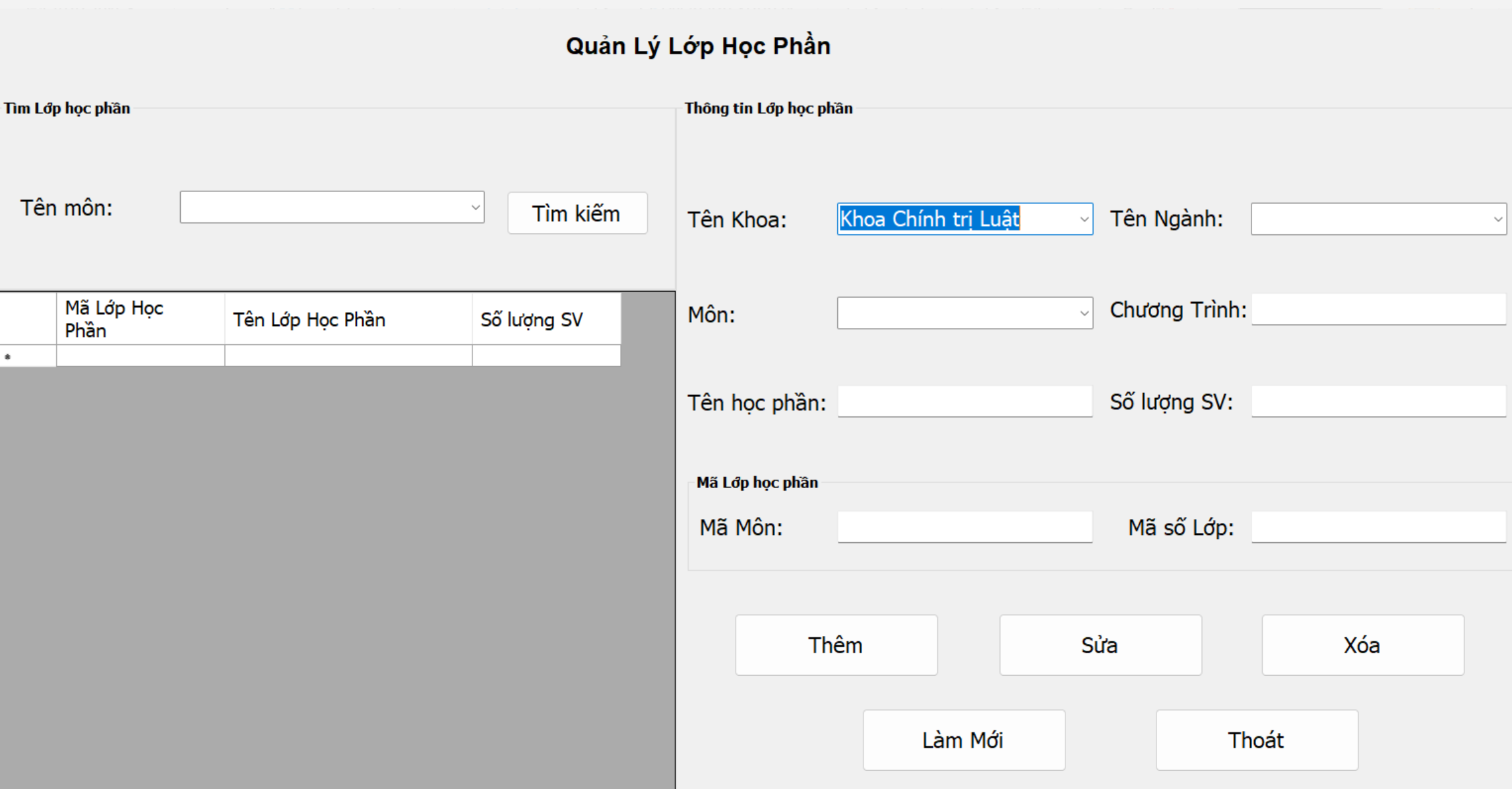
* 1. Giao diện quản lý Ngành



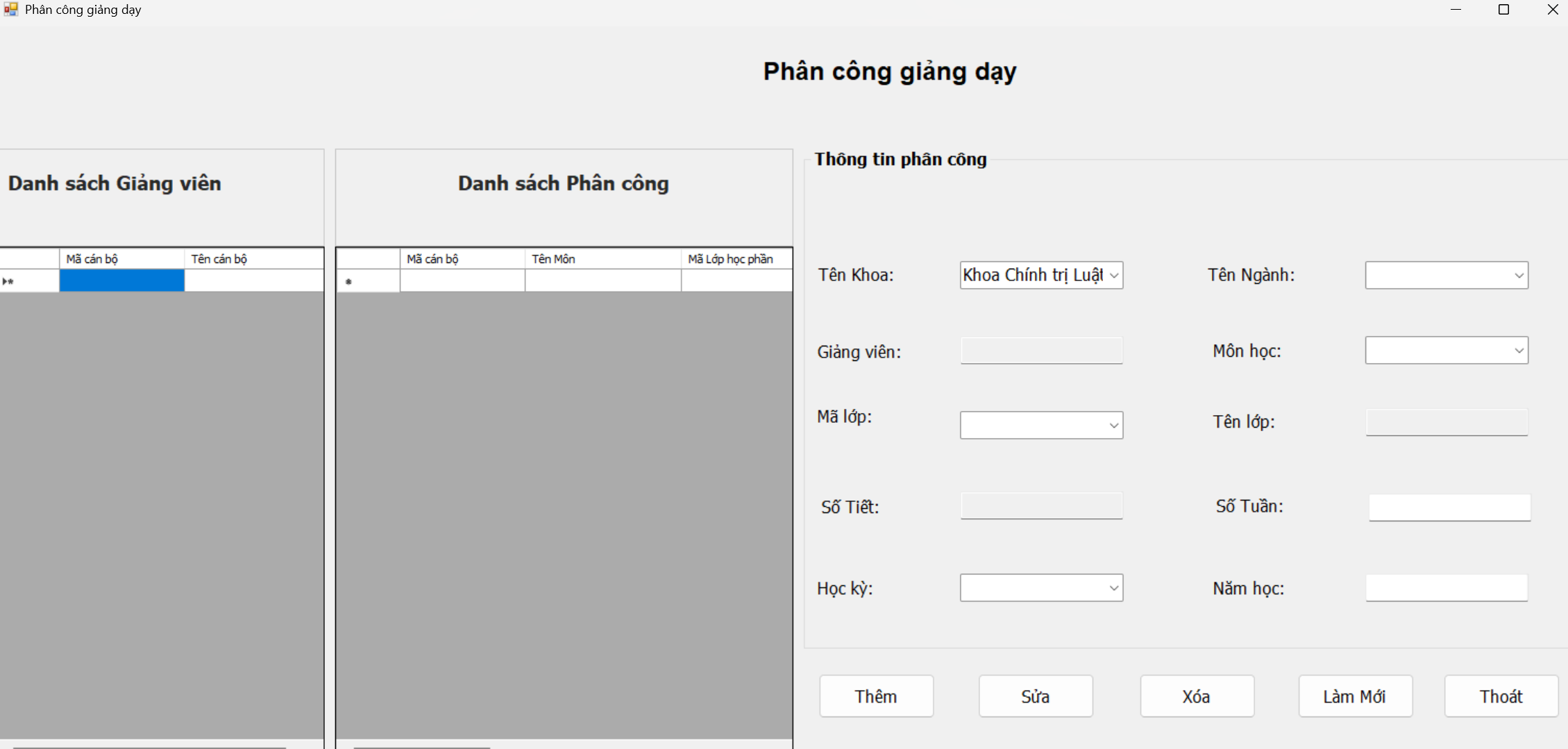
* 1. Giao diện quản lý Môn học



* 1. Giao diện quản lý Lớp học phần



* 1. Giao diện Phân công giảng dạy



* 1. Quản lý lương

