**TRƯỜNG ĐẠI HỌC QUY NHƠN**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN LẬP TRÌNH TRÊN WINDOWS FORM**

**ĐỀ TÀI: THIẾT KẾ ỨNG DỤNG QUẢN LÝ SINH VIÊN TRÊN MÔI TRƯỜNG VISUAL STUDIO**

**GVHD:**

**SVTH:**

**LỚP: 19CCT**

**QUY NHƠN, THÁNG 6 NĂM 2023**

**LỜI CẢM ƠN**

Lời đầu tiên, em xin được gửi lời cảm ơn chân thành đến quý thầy cô trong Khoa Công nghệ Thông tin của Trường Đại học Quy Nhơn đã tận tình dạy bảo cho em nhiều kiến thức bổ ích trong suốt thời gian học tập tại trường cũng như đã tạo cho em thực hiện báo cáo này.

Em xin cám ơn thầy đã tận tình chỉ bảo, hỗ trợ cho em củng cố kiến thức cũng như hoàn thiện các kỹ năng cần thiết trong học tập trong quá trình thực tập.

Mặc dù vấp phải nhiều khó khăn khi thực hiện đề tài. Thế nhưng, nhờ có thầy đã giúp cho em hoàn thành bài báo cáo đúng như mong muốn.

Một lần nữa, em xin được gửi lời cảm ơn sâu sắc đến quý nhà trường, các thầy cô Khoa Công nghệ Thông tin, thầy và các bạn trong lớp đã giúp đỡ em.

Trân trọng.

**NHẬN XÉT CỦA GIÁO VIÊN**

Đánh giá về Báo cáo thực tập tốt nghiệp: ……../**10** điểm

**MỤC LỤC**

[**CHƯƠNG I: GIỚI THIỆU** 6](#_Toc137370522)

[**CHƯƠNG II: THIẾT KẾ CƠ SỞ DỮ LIỆU** 6](#_Toc137370523)

[**CHƯƠNG III: THIẾT KẾ GIAO DIỆN** 10](#_Toc137370524)

[**CHƯƠNG IV: KẾT LUẬN** 18](#_Toc137370525)

# **CHƯƠNG I: GIỚI THIỆU**

Hiện nay, các trường đại học và cao đẳng đều có nhu cầu quản lý thông tin sinh viên để giúp cho việc lưu trữ được chính xá, không bị thất thoát hồ sơ cũng như dễ dàng tra cứu các thông tin cần thiết để phục vụ các nhu cầu cụ thể trong trường.

Do đó, thiết kế một phần mềm quản lý sinh viên là việc làm cần thiết để giúp cho người quản lý, Ban giám hiệu cũng nhà các Khoa dễ dàng quản lý, cập nhật thông tin sinh viên đảm bảo đầy đủ và tính chính xác nhất.

Xuất phát từ những nhu cầu trên, hôm nay, em xin trình bày về ứng dụng quản lý Windows Form trên môi trường Visual Studio kết hợp cơ sở dữ liệu SQL Server.

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

Trong phạm vi phần mềm, ta sẽ dùng SQL Server để lưu trữ dữ liệu và dùng câu lệnh C# trong môi trường Visual Stuido để truy xuất cơ sở dữ liệu.

Đầu tiên, ta cần tạo cơ sở dữ liệu mang tên QUANLYSINHVIEN

create database QuanLySinhVien

go

Sau đó dùng cơ sở dữ liệu bằng cách:

use QuanLySinhVien

go

Sau khi đã có CSDL ta bắt đầu tạo các bảng trong CSDL

1. Thiết kế bảng Khoa

Trong trường sẽ phân ra nhiều Khoa khác nhau, vì vậy, đầu tiên ta cần tạo bảng Khoa để lưu trữ dữ liệu từng Khoa bao gồm Mã Khoa và Tên Khoa. Cú pháp tạo bảng như sau:

create table Khoa

(

MaKhoa varchar(10) not null primary key,

TenKhoa nvarchar(100)

)

Trong đó, mỗi Khoa sẽ có một Mã duy nhất để quản lý không bị trùng với các Khoa khác nên sẽ đặt **MaKhoa làm khóa chính** và khi tạo dữ liệu không được để trống (not null).

Ta định dạng thuộc tính trong bảng tương ứng với trường dữ liệu như phù hợp với nhu cầu như varchar (kiểu chữ), nvarchar (kiểu chữ có dấu).

1. Thiết kế bảng SinhVien

Ta cần có một bảng dữ liệu để lưu trữ thông tin sinh viên bao gồm Mã sinh viên, Họ tên, Giới tính, số điện thoại, điểm GPA, Mã Khoa.

Trong đó, mỗi sinh viên sẽ có **01 Mã** duy nhất để không bị trùng với các sinh viên khác cùng Khoa hoặc khác Khoa. Do đó, ta sẽ đặt **Mã Sinh Viên là Khóa Chính.**

Cú pháp tạo bảng SINHVIEN như sau:

create table SinhVien

(

MaSinhVien varchar(10) not null primary key,

HoTen nvarchar(100),

GioiTinh nvarchar(10),

NgaySinh varchar(11),

SoDienThoai varchar(11),

GPA float,

MaKhoa varchar(10)

Constraint fk\_sinhvien\_makhoa foreign key(MaKhoa) references Khoa(MaKhoa)

)

Trong đó, ta định dạng thuộc tính trong bảng tương ứng với trường dữ liệu như phù hợp với nhu cầu như varchar (kiểu chữ), nvarchar (kiểu chữ có dấu), float (số thập phân).

Do ta quản lý thông tin Sinh viên dựa trên MaSinhVien nên sẽ đặt MaSinhVien là khóa chính và không được để trống (not null).

Mỗi sinh viên chỉ đang học tại một Khoa duy nhất nên ta cần tham chiếu đến **bảng KHOA** tương ứng. Do đó, ta sẽ **tạo khóa ngoại MaKhoa** của bảng sinh viên **tham chiếu đến MaKhoa trong bảng Khoa.**

1. Thiết kế bảng TaiKhoanNguoiDung

Mục đích bảng **Tài Khoản Người Dùng** để sinh viên có thể đăng nhập vào phần mềm để kiểm tra thông tin lại thông tin đã chính xác chưa, tra cứu điểm thi, quản lý mượn sách thư viện…

Trong đó, mỗi sinh viên sẽ có **01 Mã Người Dùng** duy nhất để không thể đăng nhập vào tài khoản của sinh viên khác. Do đó, ta sẽ đặt **TaiKhoan là khóa chính và not null.**

Cú pháp tạo bảng TaiKhoanNguoiDung như sau:

create table TaiKhoanNguoiDung

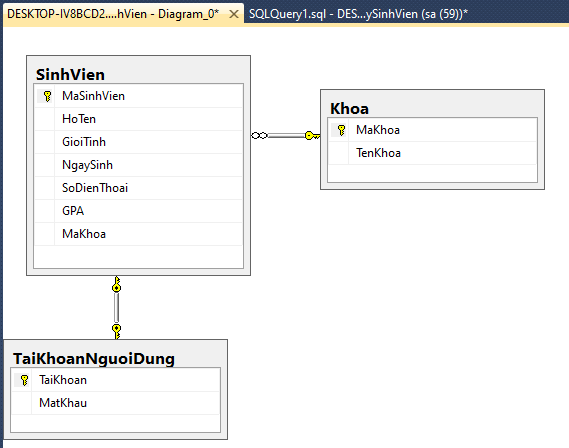
(

TaiKhoan varchar(10) not null primary key,

MatKhau varchar(100)

)

Sau khi tạo 3 bảng, ta sẽ có biểu đồ như sau:



Ta tiến hành thêm dữ liệu thử bằng câu lệnh như sau:

insert into Khoa values

('CNTT', N'Công nghệ thông tin'),

('TCNH', N'Tài chính ngân hàng'),

('QTKD', N'Quản trị kinh doanh')

insert into SinhVien values

('SV001', N'Nguyễn Văn A', 'Nam', '12/12/2000', '0909000000', 8.3, 'CNTT'),

('SV002', N'Nguyễn Thị B', N'Nữ', '08/04/2001', '0909000111', 6.6, 'TCNH'),

('SV003', N'Nguyễn Văn C', 'Nam', '31/03/1999', '0909000222', 5.8, 'QTKD')

insert into TaiKhoanNguoiDung values

('SV001', '12345678'),

('SV002', 'nguyenthib'),

('SV003', '31031999')

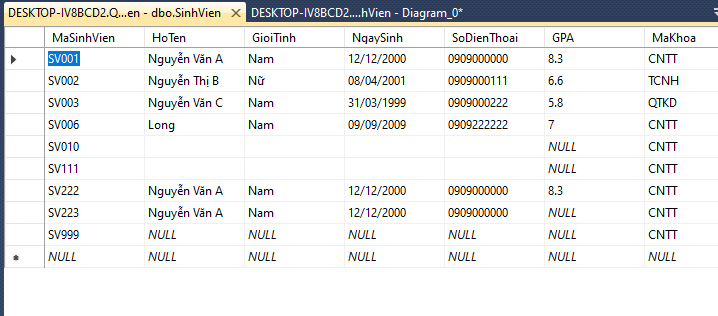
Ta sử dụng câu lệnh để kiểm tra dữ liệu đã được thêm vào chưa

select \* from Khoa

select \* from SinhVien

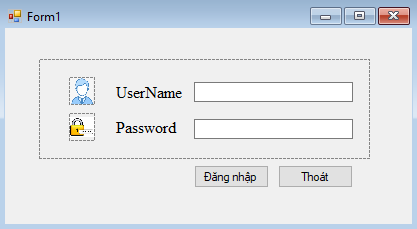
select \* from TaiKhoanNguoiDung

Hoặc có thể mở từng bảng để xem trực quan hơn



# **CHƯƠNG III: THIẾT KẾ GIAO DIỆN**

1. Màn hình đăng nhập



Các thành phần trong màn hình

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu | Mô tả |
| UserName | Textbox | Người dùng nhập username |
| Password | Textbox | Người dùng nhập password |
| Đăng nhập | Button | Bấm nút “đăng nhập” sau khi điền đủ thông tin |
| Thoát | Button | Bấm nút “thoát” khi không sử dụng phần mềm nữa |

Câu lệnh trong Class Form1:

//Tạo chuỗi kết nối đến SQL Server

string Connectionstring = "Data Source=DESKTOP-IV8BCD2;Initial Catalog=QUANLYSINHVIEN;Integrated Security=True";

//Tạo biến kết nối tên là conn mặc định null (trước khi tạo phải khai báo thư viện using System.Data.SqlClient để kết nối đến SQL)

SqlConnection conn = null;

//Tạo lớp trung gian để lấy dữ liệu mặc định null

SqlDataAdapter da = null;

//Tạo bảng để lấy dữ liệu từ SQL Server mặc định null

DataTable dt = null;

Câu lệnh cho nút đăng nhập

//Mở cổng kết nối cơ sở dữ liệu từ câu truy vấn Connectionstring đã tạo bên trên

conn = new SqlConnection(Connectionstring);

//Tạo câu truy vấn tên "da" với kết nối là "conn" đến CSDL SQL

da = new SqlDataAdapter("select Count(\*) from TaiKhoanNguoiDung where TaiKhoan='" + UserName.Text + "' and MatKhau='" + PassWord.Text + "'", conn);

//Tạo bảng

dt = new DataTable();

//Lệnh Fill để đổ dữ liệu từ câu truy lên bảng đã khai báo là dt

da.Fill(dt);

//Hàm xử lý theo yêu cầu: Nếu số hàng trong câu truy vấn SQL là 1 có nghĩa là câu truy vấn đúng TaiKhoan và MatKhau bên CSDL

if(dt.Rows[0][0].ToString() == "1")

{

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

//Khởi tạo biến f để mở Form Manage

Manage f = new Manage();

//Ẩn form hiện hành

this.Hide();

//Mở form Manage;

f.ShowDialog();

//Mở lại form đăng nhập khi đóng form Manage

this.Show();

}

else

{

MessageBox.Show("Sai tên đăng nhập hoặc mật khẩu", "Thông báo", MessageBoxButtons.OK);

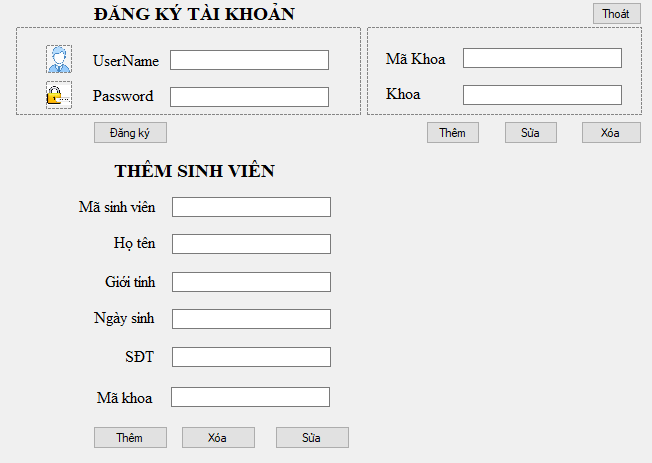
}

conn.Close();

Câu lệnh cho nút thoát

this.Close();

1. Màn hình Registration



Các thành phần trong màn hình

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu | Mô tả |
| UserName | Textbox | Người dùng nhập username để tạo tài khoản mới |
| Password | Textbox | Người dùng nhập password để tạo tài khoản mới |
| Đăng ký | Button | Người dùng bấm nút “đăng nhập” sau khi điền đủ thông tin |
| Thoát | Button | Bấm nút “thoát” cửa sổ hiện hành |
| Mã Khoa | Textbox | Người dùng nhập Mã Khoa để tạo Mã Khoa Mới |
| Tên Khoa | Textbox | Người dùng nhập tên Khoa mới |
| Thêm | Button | Người ấn nút tương ứng theo nhu cầu trong bảng Khoa |
| Xóa | Button |
| Sửa | Button |
| Mã sinh viên | Textbox | Người dùng nhập các thông tin trong bảng sinh viên để tạo mới |
| Họ tên | Textbox |
| Giới tính | Textbox |
| Ngày sinh | Textbox |
| SĐT | Textbox |
| Mã Khoa | Textbox |
| Thêm | Button | Người ấn nút tương ứng theo nhu cầu trong bảng SinhVien |
| Xóa | Button |
| Sửa | Button |

Câu lệnh trong Class Form Registration:

//Tạo chuỗi kết nối đến SQL Server

string Connectionstring = "Data Source=DESKTOP-IV8BCD2;Initial Catalog=QUANLYSINHVIEN;Integrated Security=True";

//Tạo biến kết nối tên là conn mặc định null (trước khi tạo phải khai báo thư viện using System.Data.SqlClient để kết nối đến SQL)

SqlConnection conn = null;

//Tạo lớp trung gian để lấy dữ liệu mặc định null

SqlDataAdapter da = null;

//Tạo bảng để lấy dữ liệu từ SQL Server mặc định null

DataTable dt = null;

Hàm khi nhấn nút “Thêm” Khoa:

if (txtMaKhoa1.Text == "" || txtTenKhoa.Text == "")

{

MessageBox.Show("Vui lòng đầy đủ thông tin trong bảng Khoa", "Thông báo");

}

else

{

try

{

conn.Open();

SqlCommand cmd = new SqlCommand("insert into Khoa values(N'" + txtMaKhoa1.Text + "', N'" + txtTenKhoa.Text + "')", conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm thông tin thành công", "Thông báo");

conn.Close();

}

catch (SqlException)

{

MessageBox.Show("Có lỗi khi thêm thông tin", "Thông báo");

}

}

Hàm khi nhấn nút “Sửa” Khoa:

if (txtMaKhoa1.Text == "" || txtTenKhoa.Text == "")

{

MessageBox.Show("Vui lòng đầy đủ thông tin trong bảng Khoa", "Thông báo");

}

else

{

try

{

conn.Open();

SqlCommand cmd = new SqlCommand("update Khoa set TenKhoa=N'" + txtTenKhoa.Text + "' where MaKhoa='" + txtMaKhoa1.Text + "'", conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Sửa thông tin thành công", "Thông báo");

conn.Close();

}

catch (SqlException)

{

MessageBox.Show("Có lỗi khi sửa thông tin", "Thông báo");

}

}

Hàm khi nhấn nút “Xóa” Khoa:

if (txtMaKhoa1.Text == "")

{

MessageBox.Show("Vui lòng đầy đủ thông tin trong bảng Khoa", "Thông báo");

}

else

{

try

{

conn.Open();

SqlCommand cmd = new SqlCommand("delete from Khoa where MaKhoa='" + txtMaKhoa1.Text + "'", conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Xóa thông tin thành công", "Thông báo");

conn.Close();

}

catch (SqlException)

{

MessageBox.Show("Có lỗi khi xóa thông tin", "Thông báo");

}

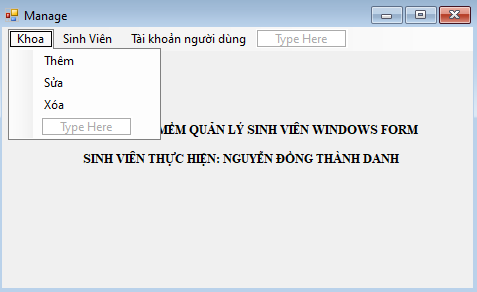
}

Ta thiết lập tương tự cho các nút thêm, xóa, sửa trong bảng Thêm Sinh Viên và Đăng Ký Người Dùng.

Lệnh khi bấm nút “thoát”

this.Close();

1. Màn hình Manage



Các thành phần trong màn hình

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu | Mô tả |
| Menutrip | Menu | Tạo các menu trên thanh menu |
| Thêm | MenuToolStrip | Tạo các sub menu thao tác theo nhu cầu |
| Xóa | MenuToolStrip | Tạo các sub menu thao tác theo nhu cầu |
| Sửa | MenuToolStrip | Tạo các sub menu thao tác theo nhu cầu |

Các lệnh tkhi click vào menu “Thêm” “Sửa” “Xóa”

//Khởi tạo biến f để mở form đăng ký (Registration)

Registration f = new Registration();

//Khi click vào nút Đăng ký sẽ ẩn form hiện hành

this.Hide();

//Mở form mới f vừa khai báo bên trên

f.ShowDialog();

//Mở lại form đăng nhập khi đóng form đăng ký

this.Show();

# **CHƯƠNG IV: KẾT LUẬN**

Ngôn ngữ C# và SQL có liên hệ mật thiết với nhau, ta có thể sử dụng ngôn ngữ C# trên môi trường Visual Studio vẫn có thể kết hợp sử dụng ngôn ngữ SQL Server để truy vấn đến CSDL bằng cách khai báo biến string và sử dụng các thư viện có sẵn trên Visual để kết nối đến SQL.

Khi thêm dữ liệu trong từng bảng phải đảm bảo tính toàn vẹn dữ liệu truy vấn đến CSDL đã thiết kế.

Ví dụ, ta không thể thêm sinh viên nếu khai báo thiếu Mã Khoa trong dữ liệu vì bảng CSDL SinhVien đã khai báo khóa ngoại truy vấn đến bảng Khoa.

Tương tự ta không thể thêm tài khoản đăng nhập khi chưa có Mã Sinh Viên hoặc mã sinh viên đó đã có tài khoản.