**ĐẠI HỌC ĐÀ NẴNG   
TRƯỜNG ĐẠI HỌC KINH TẾ**

**KHOA THỐNG KÊ – TIN HỌC**



**BÁO CÁO THỰC TẬP NHẬN THỨC**

**QUẢN LÝ HỌC VIÊN**

**Giáo viên hướng dẫn: Nguyễn Thành Thủy**

**Sinh viên:**

1. Huỳnh Ngọc Trâm

Lớp: 44k21.2

1. Nguyễn Phi Hoàng

Lớp:44K14

*Đà Nẵng, 29 tháng 8 năm 2020*

# **LỜI CAM ĐOAN**

Chúng em xin cam đoan báo cáo HỆ THỐNG QUẢN LÝ HỌC VIÊN là kết quả nghiên cứu độc lập dưới sự hướng dẫn của giáo viên hướng dẫn:ThS.Nguyễn Thành Thủy. Không có sự sao chép nào của người khác.

Nội dung và báo cáo thực tập nhận thức là kết quả mà chúng em đã nghiên cứu và tìm hiểu trong quá trình học tập tại nhà cũng như học hỏi từ các video tham khảo mà thầy cô đã đưa ra. Các dữ liệu, kết quả trình bày trong báo cáo là hoàn toàn trung thực.

Chúng em xin chịu hoàn toàn trách nhiệm, kỷ luật của bộ môn và nhà trường đề ra nếu như có vấn đề xảy ra.

MỤC LỤC

[**LỜI CAM ĐOAN** 2](#_Toc49874308)

[**BẢNG PHÂN CHIA CÔNG VIỆC** 7](#_Toc49874309)

[**MÔ TẢ BÀI TOÁN** 9](#_Toc49874310)

[1.1 Tóm tắt bài toán: 9](#_Toc49874311)

[1.1.1 Tóm tắt hoạt động của hệ thống: 9](#_Toc49874312)

[1.1.2 Các thông tin sử dụng trong hệ thống 9](#_Toc49874313)

[1.2 Dự định của hệ thống 9](#_Toc49874314)

[1.2.1 Tên hệ thống 9](#_Toc49874315)

[1.3 Mô tả các yêu cầu của hệ thống 10](#_Toc49874316)

[1.3.1 Tổng quan về hệ thống 10](#_Toc49874317)

[1.3.2 Nhận định về cơ cấu quản lý của hệ thống 10](#_Toc49874318)

[1.3.3 Đặc tả cơ cấu quản lý của hệ thống 10](#_Toc49874319)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU 12](#_Toc49874320)

[2.1 Phân tích hệ thống 12](#_Toc49874321)

[2.1.1 Sơ đồ USECASE hệ thống: 12](#_Toc49874322)

[2.1.2 Mô tả chi tiết 13](#_Toc49874323)

[2.1.3 Sơ đồ phân rã chức năng 15](#_Toc49874324)

[2.1.4 Sơ đồ hoạt động (Activity) 17](#_Toc49874325)

[2.1.5 Sơ đồ Class: 19](#_Toc49874326)

[2.2 Thiết kế mô hình cơ sở dữ liệu: 19](#_Toc49874327)

[2.2.1 Các thực thể có trong hệ thống 19](#_Toc49874328)

[2.2.2 Thuộc tính của các thực thể và chuẩn hoá chúng 19](#_Toc49874329)

[2.2.3 Mô hình thực thể liên kết (ER) 20](#_Toc49874330)

[2.2.4 Lược đồ quan hệ 20](#_Toc49874331)

[2.2.5 Thiết kế chi tiết dữ liệu 20](#_Toc49874332)

[2.2.6 Sơ đồ dữ liệu quan hệ 21](#_Toc49874333)

[CHƯƠNG 3: THIẾT KẾ VÀ PHÁT TRIỂN ỨNG DỤNG 22](#_Toc49874334)

[3.1 Form Đăng nhập: 22](#_Toc49874335)

[3.1.1 Cấu hình form 22](#_Toc49874336)

[3.1.2 Xử lý sự kiện trong Form Đăng nhập 22](#_Toc49874337)

[3.2 Form MDI 22](#_Toc49874338)

[3.2.1 Cấu hình Form 22](#_Toc49874339)

[3.2.2 Xử lý sự kiện Form MDI 22](#_Toc49874340)

[3.3 From Add học viên 22](#_Toc49874341)

[3.3.1 Cấu hình Form 22](#_Toc49874342)

[3.3.2 Xử lý xự kiện Form Add 22](#_Toc49874343)

[3.4 Form Update học viên 23](#_Toc49874344)

[3.4.1 Cấu hình Form 23](#_Toc49874345)

[3.4.2 Xử lý sự kiện Form Update 23](#_Toc49874346)

[3.5 Form Search học viên 23](#_Toc49874347)

[3.5.1 Cấu hình Form 23](#_Toc49874348)

[3.5.2 Xử lý sự kiện Form Search 23](#_Toc49874349)

[3.6 Form Thống kê học viên 23](#_Toc49874350)

[3.6.1 Cấu hình Form 23](#_Toc49874351)

[3.6.2 Xử lý sự kiện Form Thống kê 23](#_Toc49874352)

[KẾT LUẬN 24](#_Toc49874353)

[4.1 Kết quả đạt được 24](#_Toc49874354)

[4.2 Những ưu điểm của hệ thống 24](#_Toc49874355)

[4.3 Những nhược điểm của hệ thống 24](#_Toc49874356)

[TÀI LIỆU THAM KHẢO 25](#_Toc49874357)

[PHỤ LỤC 26](#_Toc49874358)

**DANH MỤC HÌNH ẢNH**

[Hình 1: Sơ đồ Usecase 10](#_Toc49873604)

[Hình 2: Sơ đồ phân rã chức năng 14](#_Toc49873605)

[Hình 3: Sơ đồ Activity Đăng nhập 15](#_Toc49873606)

[Hình 4: Sơ đồ Activity Thêm học viên 15](#_Toc49873607)

[Hình 5: Sơ đồ Activity Cập nhật học viên 15](#_Toc49873608)

[Hình 6: Sơ đồ Activity Tìm kiếm học viên 16](#_Toc49873609)

[Hình 7: Sơ đồ Activity Xóa học viên 16](#_Toc49873610)

[Hình 8: Sơ đồ Activity Đăng xuất 16](#_Toc49873611)

[Hình 9: Sơ đồ Class 17](#_Toc49873612)

[Hình 10: Mô hình thực thể liên kết 18](#_Toc49873613)

[Hình 11: Sơ đồ dữ liệu quan hệ 19](#_Toc49873614)

**DANH MỤC BẢNG BIỂU**

[Bảng 1: Bảng HOCVIEN 19](#_Toc49873684)

[Bảng 2: Bảng DMLOP 19](#_Toc49873685)

# **BẢNG PHÂN CHIA CÔNG VIỆC**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên học viên | Nhiệm vụ | Ngày  bắt đầu | Ngày  kết thúc | Kết quả |
| 1 | Huỳnh Ngọc Trâm  Nguyễn Phi Hoàng | Xem lại tất cả video đã làm trong khóa 1 và 2 | 24/8/2020 | 15h  26/82020 | Hoàn thành |
| 2 | Nguyễn Phi Hoàng | Tạo cơ sở dữ liệu và nhập dữ liệu. | 15h 26/8/2020 | 17h  28/8/2020 | Hoàn thành |
| 3 | Huỳnh Ngọc Trâm | Thiết lập các ràng buộc cho bảng, kiểu dữ liệu. | 15h  26/8/2020 | 17h  26/8/2020 | Hoàn thành |
| 4 | Nguyễn Phi Hoàng | Tạo các Proc để viết chương trình | 15h  26/8/2020 | 17h  26/8/2020 | Hoàn thành |
| 5 | Huỳnh Ngọc Trâm | Làm báo cáo dựa trên dữ liệu đã thiết kế | 17h  26/8/2020 | 19h  26/8/2020 | Hoàn thành |
| 6 | Nguyễn Phi Hoàng | Tạo Form đăng nhập | 17h  27/8/2020 | 17h15  27/8/2020 | Hoàn thành |
| 7 | Huỳnh Ngọc Trâm | Viết code đăng nhập và chỉnh sửa. | 17h15  27/8/2020 | 18h  27/8/2020 | Hoàn thành |
| 8 | Huỳnh Ngọc Trâm | Thiết kế form MDI | 19h  27/8/2020 | 19h15  27/8/2020 | Hoàn thành |
| 9 | Nguyễn Phi Hoàng | Viết code liên kết với các form khác. | 19h15  27/8/2020 | 20h  27/8/2020 | Hoàn thành |
| 10 | Huỳnh Ngọc Trâm | Thiết kế Form, viết code phần Update và Search. | 9h  28/8/2020 | 15h  28/8/2020 | Hoàn thành |
| 11 | Nguyễn Phi Hoàng | Thiết kế Form, viết code phần Add và Delete. | 9h  28/8/2020 | 15h  28/8/2020 | Hoàn thành |
| 12 | Huỳnh Ngọc Trâm  Nguyễn Phi Hoàng | Xem lại tất cả các Form và góp ý cho nhau. | 16h  28/8/2020 | 20h  28/8/2020 | Hoàn thành |
| 13 | Huỳnh Ngọc Trâm  Nguyễn Phi Hoàng | Chỉnh sửa Form mình đảm nhận, thêm các trường hợp cho code được hợp lý. | 9h  29/8/2020 | 17h  30/8/2020 | Hoàn thành |
| 14 | Huỳnh Ngọc Trâm | Chỉnh lại bố cục giao diện và form chữ, hiệu ứng. | 19h  30/8/2020 | 20h30  30/8/2020 | Hoàn thành |
| 15 | Nguyễn Phi Hoàng | Tạo code cho form Thống kê | 9h  31/8/2020 | 15h  31/8/2020 | Hoàn thành |
| 17 | Nguyễn Phi Hoàng | Tạo code cho Đăng xuất | 15h  31/9/2020 | 19h  31/9/2020 | Hoàn thành |
| 16 | Huỳnh Ngọc Trâm | Thêm hình ảnh, màu sắc cho tất cả các form. | 9h  1/9/2020 | 21h  1/9/2020 | Hoàn thành |
| 18 | Huỳnh Ngọc Trâm | Làm báo cáo tổng kết tất cả. | 9h 2/9/2020 | 9h 4/9/2020 | Hoàn thành |
| 19 | Nguyễn Phi Hoàng | Quay video Tổng kết. | 9h  2/9/2020 | 9h  4/9/2020 | Hoàn thành |
| 20 | Huỳnh Ngọc Trâm  Nguyễn Phi Hoàng | Xem lại bản Word và video cũng như chạy lại tất cả chương trình. | 10h  4/9/2020 | 10h  5/9/2020 | Hoàn thành |

# **MÔ TẢ BÀI TOÁN**

## Tóm tắt bài toán:

### 1.1.1 Tóm tắt hoạt động của hệ thống:

* Yêu cầu làm một hệ thống Quản lý học viên, dực trên những yêu cầu giáo viên hướng dẫn đã đưa ra.
* Mô tả môi trường: bộ phận CNTT sẽ thực hiện quản lý thông tin của học viên theo từng lớp khác nhau, mỗi lớp bao gồm một danh sách học viên trực thuộc và mỗi sinh viên sẽ có thông tin về học phí và tên.
* Việc quản lý thông tin sinh viên như sau: sẽ lấy mã học viên làm khóa chính để quản lý thông tin liên quan đến tên, học phí, lớp học.
* Các chức năng chính của hệ thống:
* Xem (hiển thị)
* Thêm mới
* Chỉnh sửa
* Tìm kiếm
* Xóa
* Lưu thông tin
* Tổng hợp/Thống kê

### 1.1.2 Các thông tin sử dụng trong hệ thống

* Các thông tin chi tiết về học viên, lớp và học phí của từng học viên.

## 1.2 Dự định của hệ thống

* Đây là một phần mềm cho việc tự động hóa công tác quản lý học viên trong nhà trường hoặc trung tâm dạy học.
* Xây dựng hệ thống với chức năng chính là chức năng quản trị. Người quản lý phải đăng nhập vào hệ thống để sử dụng chức năng quản trị, có quyền thêm, sửa, xóa, tìm kiếm, thống kê các thông tin liên quan đến học viên.

### 1.2.1 Tên hệ thống

* Hệ thống quản lý học viên.

1.2.2 Môi trường triển khai

* Triển khai trong môi trường: Visual Studio 2013.
* Hệ quản trị CSDL SQL Server.
* Ngôn ngữ lập trình C# Winform.

## 1.3 Mô tả các yêu cầu của hệ thống

### 1.3.1 Tổng quan về hệ thống

* Mỗi học viên sẽ được gắn cho một mã học viên. Mã học viên này không thay đổi trong suốt quá trình học tập và quản lý dữ liệu.
* Thông tin về tên và học phí của học viên.

### 1.3.2 Nhận định về cơ cấu quản lý của hệ thống

* Chức năng quản lý chính: quản lý thông tin học viên (thêm, sửa, xóa, tìm kiếm thông tin về học phí).
* Quản lý lớp (tìm kiếm thông tin lớp sinh hoạt).
* Học phí học viên (thêm, sửa, xóa, thông tin học phí).
* Thống kê sinh viên và học phí theo lớp.

### 1.3.3 Đặc tả cơ cấu quản lý của hệ thống

* Hệ thống là sự tích hợp các chức năng liên quan đến quản lý học viên, có khả
* năng thao tác với các chức năng: thêm, sửa, xóa, sao lưu các dữ liệu khi thay đổi
* Thông tin được bảo toàn trọn vẹn các một cách tối ưu nhất.
* Để hiểu hơn về hoạt động của hệ thống, ta đi tìm hiểu cụ thể về cơ cấu quản lý của hệ thống.

**Chức năng quản lý chính: quản lý học viên**

* Mục đích: giúp chúng ta có thể thao tác và thống kê thông tin của học viên một cách dễ dàng, trực quan hơn.
* Tóm lược: Quản trị viên sẽ đăng nhập vào hệ thống Quản lý học viên bằng tài khoản và mật khẩu của mình. Hệ thống kiểm tra thấy mật khẩu là đúng, sau đó chọn vào sau đó chọn các chức năng thêm sửa tìm cập nhật xóa, sau đó có thể thao tác trên form có chức năng đó.
* Đầu vào:
* Mã học viên
* Họ và tên học viên
* Học phí
* Mã lớp
* Đầu ra:
* Danh sách học viên.
* Thông tin cần biết về học viên.
* Sự cố xảy ra :

Người quản lý đăng nhập mật khẩu vào không đúng. Người quản lý phải nhập lại tài khoản và mật khẩu hoặc kết thúc sử dụng.

**Quản lý lớp**

* Mục đích: quản lý lớp cũng rất quan trọng, do trong trường có rất nhiều sinh viên trực thuộc những lớp khác nhau. Hệ thống sẽ giúp theo dõi quá trình nộp học phí của học viên trong lớp, quản lý tự động hoá mà không mất nhiều thời gian.
* Tóm lược: Mỗi lớp sinh hoạt có nhiều học viên. Khi đăng nhập thành công hệ thống cho phép tìm kiếm các thông tin liên quan đến lớp.
* Đầu vào:

Tên lớp

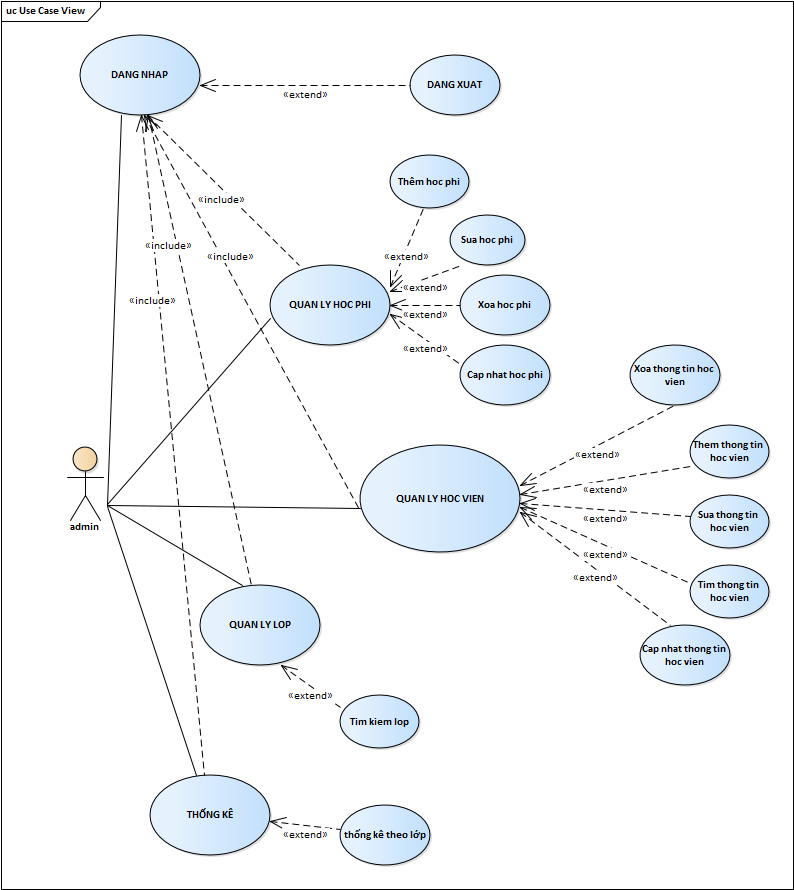
* Đầu ra:

Danh sách các lớp

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU

2.1 Phân tích hệ thống

### 2.1.1 Sơ đồ USECASE hệ thống:



Hình 1: Sơ đồ Usecase

### 2.1.2 Mô tả chi tiết

|  |  |
| --- | --- |
| Usecase | Nội dung |
| Tên usecase | Đăng nhập |
| Mô tả | Cho phép người quản lý đăng nhập vào hệ thống. |
| Actor | Admin |
| Điều kiện kích hoạt | Có tài khoản và mật khẩu |
| Nhập | Tên: admin  Mật khẩu: admin |
| Xuất | * Nếu nhập sai hiển thị thông báo tài khoản hoặc mật khẩu không đúng và yêu cầu nhập lại. * Đúng thì chuyển qua form MDI |

|  |  |
| --- | --- |
| Usecase | Nội dung |
| Tên usecase | Quản lý học viên |
| Mô tả | Cho phép truy cập chức năng quản lý sinh viên và thực hiện các thao tác thêm, sửa, xóa, cập nhật, tìm kiếm. |
| Actor | Admin |
| Điều kiện kích hoạt | Nhập đúng tên và mật khẩu |
| Nhập | Người quản lý đăng nhập thành công |
| Xuất | Thực hiện các chức năng muốn chọn |

|  |  |
| --- | --- |
| Usecase | Nội dung |
| Tên usecase | Quản lý lớp |
| Mô tả | Cho phép truy cập chức năng quản lý lớp và thực hiện tìm kiếm lớp. |
| Actor | Admin |
| Điều kiện kích hoạt | Có tài khoản và mật khẩu |
| Nhập | Người quản lý đăng nhập thành công |
| Xuất | Thực hiện xem các lớp và tìm kiếm lớp |

|  |  |
| --- | --- |
| Usecase | Nội dung |
| Tên usecase | Quản lý học phí |
| Mô tả | Cho phép truy cập chức năng quản lý học phí và thực hiện thêm, sửa, xóa, cập nhật học phí. |
| Actor | Admin |
| Điều kiện kích hoạt | Có tài khoản và mật khẩu |
| Nhập | Người quản lý đăng nhập thành công |
| Xuất | Thực hiện các thao tác tùy chọn |

|  |  |
| --- | --- |
| Usecase | Nội dung |
| Tên usecase | Thống kê |
| Mô tả | Hệ thống hiện lên thông tin học phí theo lớp |
| Actor | Admin |
| Điều kiện kích hoạt | Có tài khoản và mật khẩu |
| Nhập | Người quản lý đăng nhập thành công |
| Xuất | Hệ thống hiện thống kê theo lớp |

2.1.3 Sơ đồ phân rã chức năng

Hệ thống quản lý sinh viên chỉ bao gồm 5 chức năng chính: thêm, xóa, cập nhật, tìm kiếm, thống kê.

- Biểu đồ sẽ tiến hành theo một cấu trúc dạng nhánh phân mức từ trên xuống:

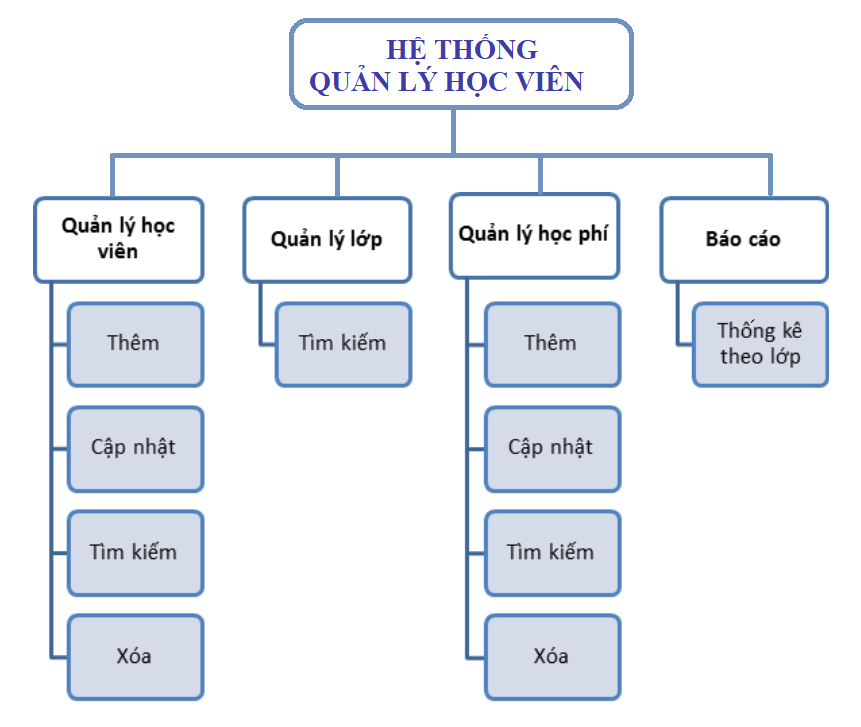
Chức năng 1: thêm thông tin sinh viên, lớp sinh hoạt, học phí.

Chức năng 2: cập nhật thông tin sinh viên, lớp sinh hoạt, học phí.

Chức năng 3: tìm kiếm thông tin sinh viên, lớp sinh hoạt.

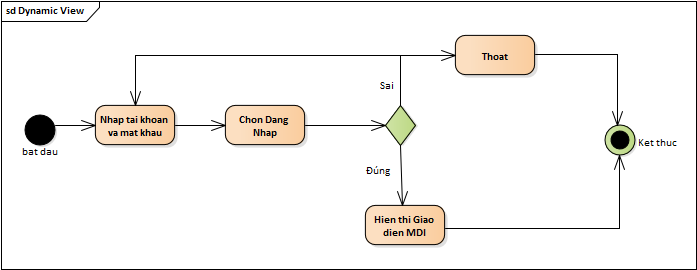
Chức năng 4: xóa thông tin sinh viên, học phí.

Chức năng 5: báo cáo thống kê học phí theo lớp.



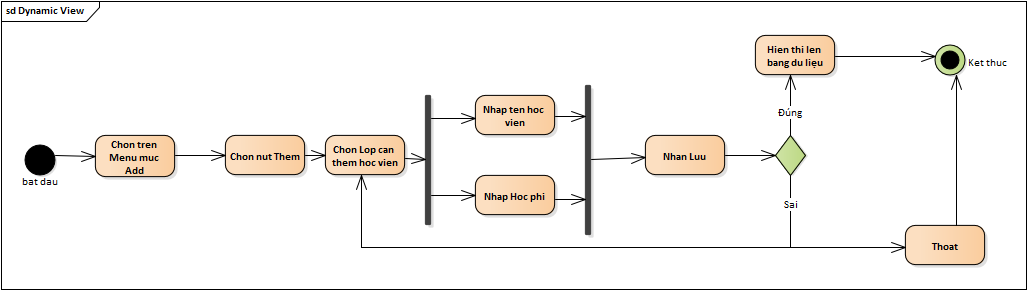
Hình 2: Sơ đồ phân rã chức năng

* + 1. Sơ đồ hoạt động (Activity)
* Hoạt động đăng nhập:



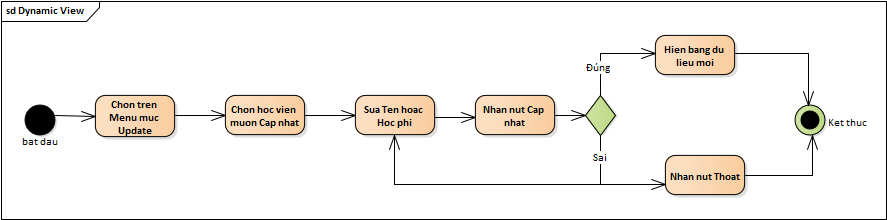
Hình 3: Sơ đồ Activity Đăng nhập

* Hoạt động Thêm học viên:



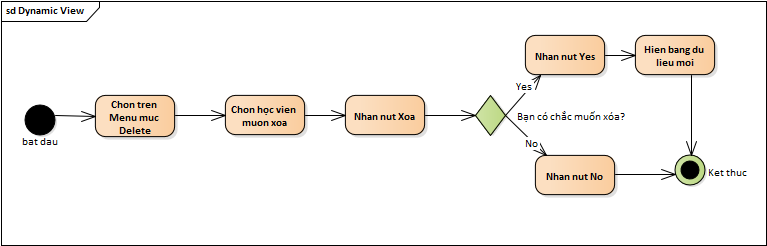
Hình 4: Sơ đồ Activity Thêm học viên

* Hoạt động Cập nhật học viên:



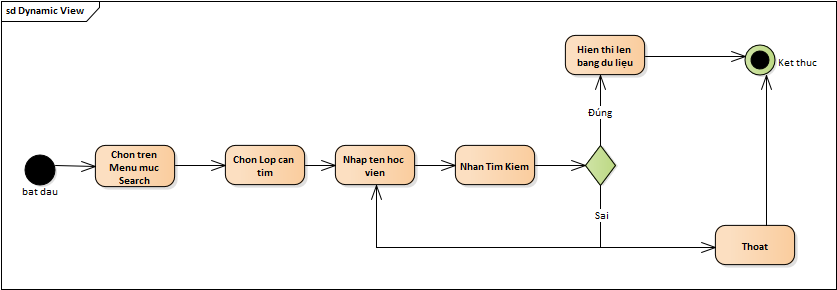
Hình 5: Sơ đồ Activity Cập nhật học viên

* Hoạt động Tìm kiếm học viên:



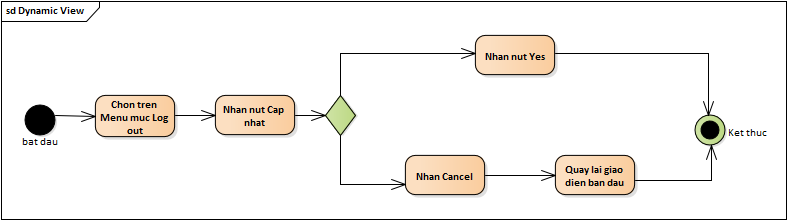
Hình 6: Sơ đồ Activity Tìm kiếm học viên

* Hoạt động Xóa học viên:



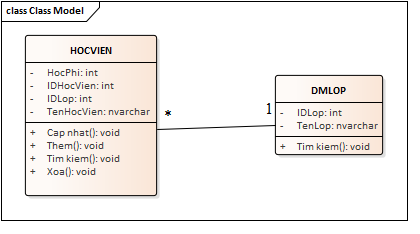
Hình 7: Sơ đồ Activity Xóa học viên

* Hoạt động Đăng xuất:



Hình 8: Sơ đồ Activity Đăng xuất

* + 1. Sơ đồ Class:



Hình 9: Sơ đồ Class

* 1. Thiết kế mô hình cơ sở dữ liệu:

2.2.1 Các thực thể có trong hệ thống

- Học Viên (HOCVIEN)

- Lớp (DMLOP)

- Tài khoản (TAIKHOAN)

2.2.2 Thuộc tính của các thực thể và chuẩn hoá chúng

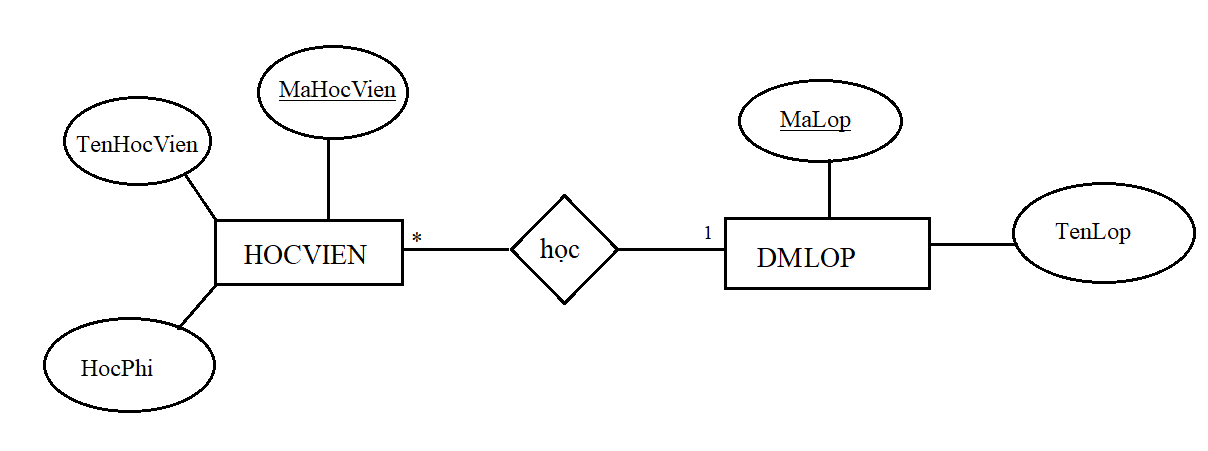
- HOCVIEN (MaHocVien, TenHocVien, HocPhi).

- LOPSINHHOAT (MaLop, TenLop).

- Các thuộc tính được chọn làm KHOÁ CHÍNH là các thuộc tính được gạch chân.

2.2.3 Mô hình thực thể liên kết (ER)

* Thông tin về lớp sinh hoạt: mã lớp là duy nhất, tên lớp.
* Mỗi sinh viên được xếp vào một lớp. Thông tin về sinh viên: mã sinh viên, học phí, mã lớp. Một lớp sẽ có nhiều sinh viên.



Hình 10: Mô hình thực thể liên kết

2.2.4 Lược đồ quan hệ

* HOCVIEN (IDHocVien, TenHocVien, HocPhi, IDLop)
* IDLop là khoá ngoại của HOCVIEN nối tới bảng DMLOP.
* DMLOP (IDLop, TenLop)

2.2.5 Thiết kế chi tiết dữ liệu

HOCVIEN

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu | Độ rộng | Ràng buộc |
| IDHocVien | Int | Auto number | Primary key |
| TenHocVien | Nvarchar | 50 | Not null |
| HocPhi | Int |  | Not null |
| IDLop | Int |  | Not null |

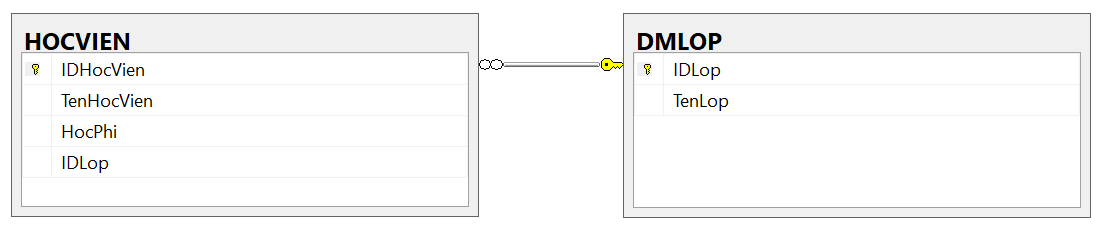
Bảng 1: Bảng HOCVIEN

DMLOP

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Kiểu | Độ rộng | Ràng buộc |
| IDLop | Int | Auto number | Primary key |
| TenLop | Nvarchar | 20 | Not null |

Bảng 2: Bảng DMLOP

2.2.6 Sơ đồ dữ liệu quan hệ

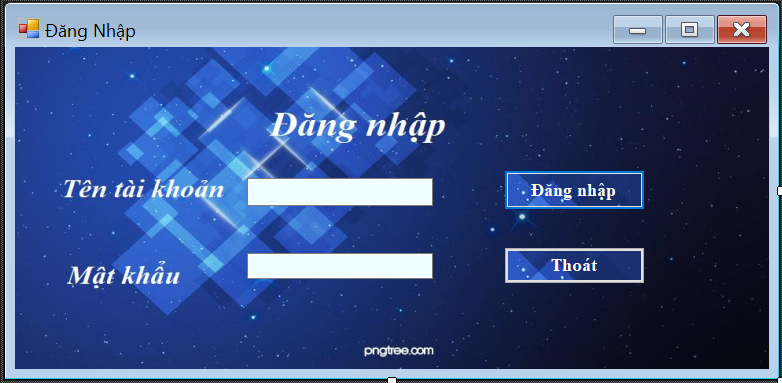


Hình 11: Sơ đồ dữ liệu quan hệ

CHƯƠNG 3: THIẾT KẾ VÀ PHÁT TRIỂN ỨNG DỤNG

3.1 Form Đăng nhập:

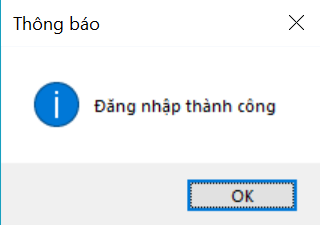
3.1.1 Cấu hình form



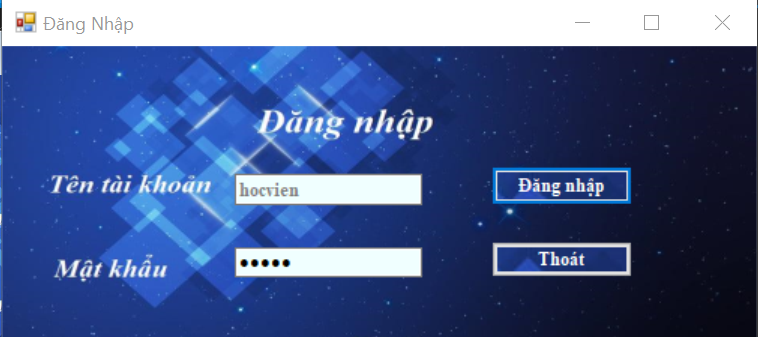
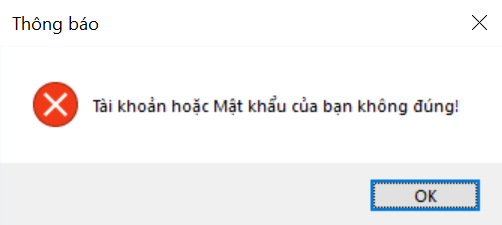
Hình 12: Form Đăng nhập

3.1.2 Xử lý sự kiện trong Form Đăng nhập

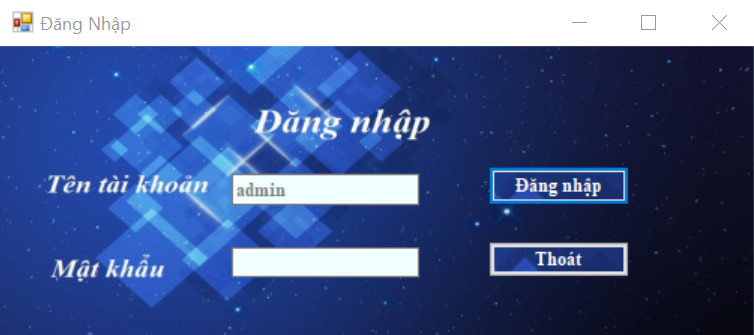
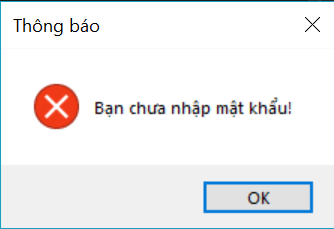
* Khi nhà quản lý muốn vào hệ thống thì cần đăng nhập. Sẽ cấp cho nhà quản lý một tài khoản, khi đăng nhập sai hoặc chưa nhập thì sẽ thông báo chưa nhập hoặc nhập sai. Người quản lý sẽ nhập lại hoặc thoát. Nếu nhập đúng tài khoản thì thông báo “Đăng nhập thành công” và chuyển đến form MDI.
* Đăng nhập đúng:

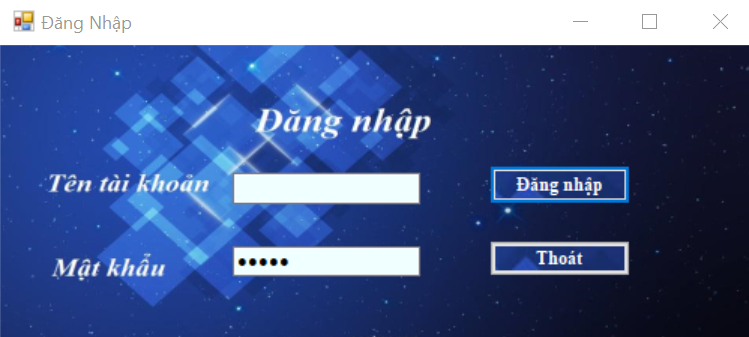
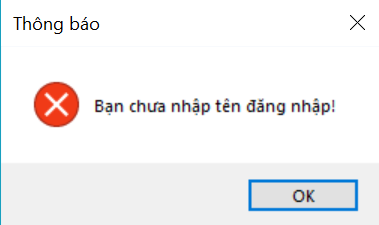
 

* Đăng nhập sai:

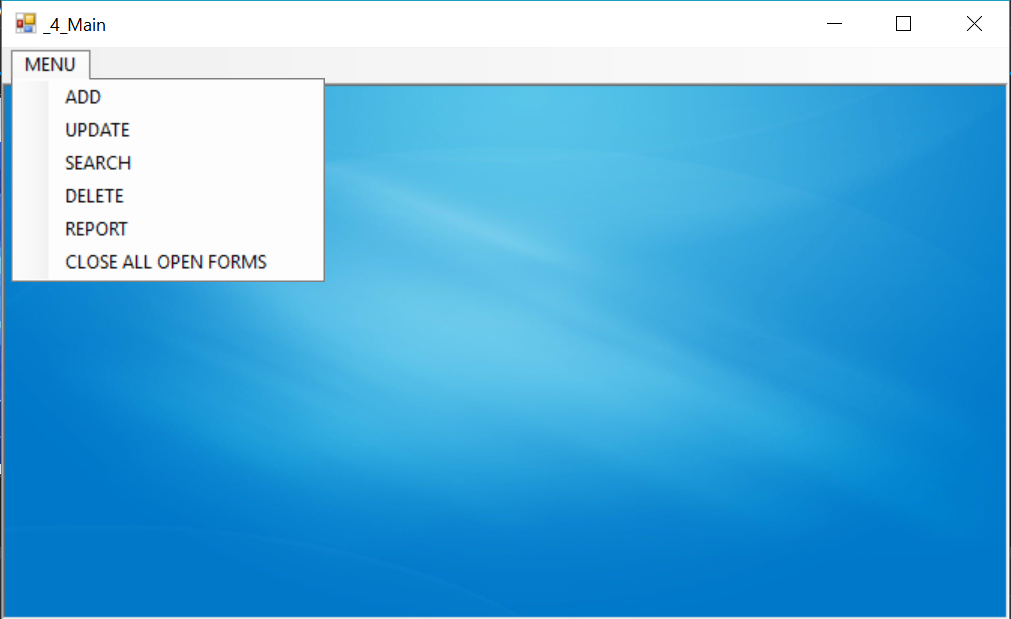
* Đăng nhập thiếu tên hoặc mật khẩu:

3.2 Form MDI

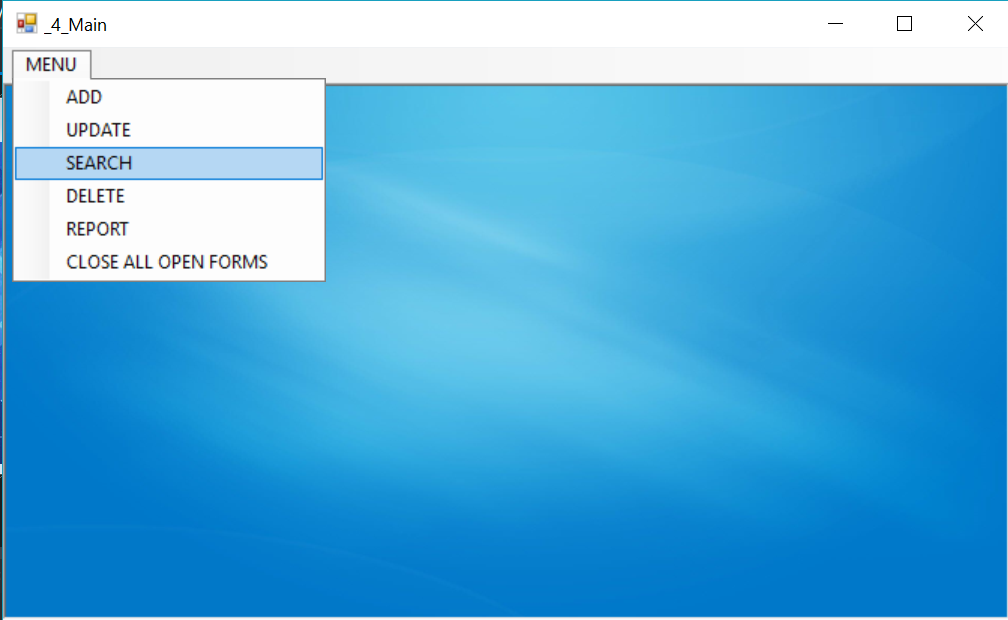
3.2.1 Cấu hình Form



Hình 13: Form MDI

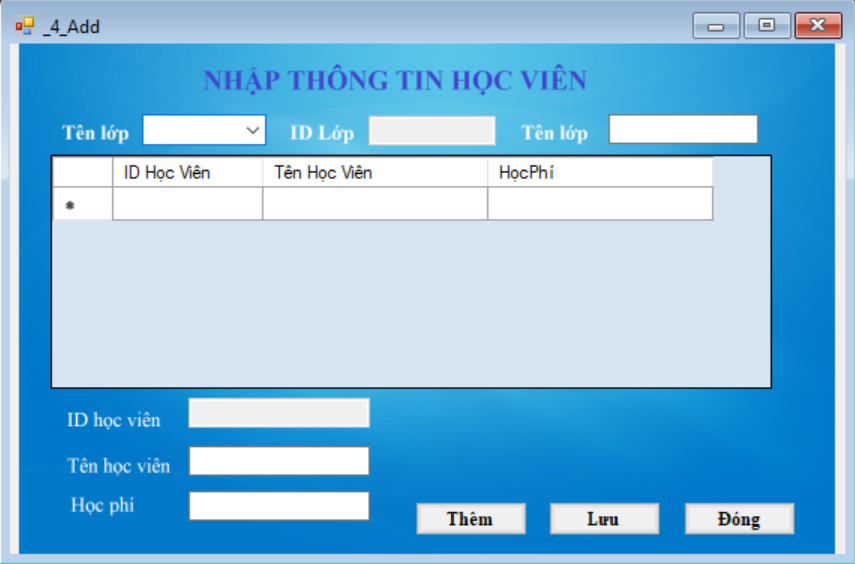
3.2.2 Xử lý sự kiện Form MDI

* Khi Người quản lý Đăng nhập vào hệ thống, sẽ chọn mục mà mình muốn thao tác từ Menu sẽ hiện ra form muốn thao tác.



3.3 From Add học viên

3.3.1 Cấu hình Form



Hình 14: Form Add

3.3.2 Xử lý xự kiện Form Add

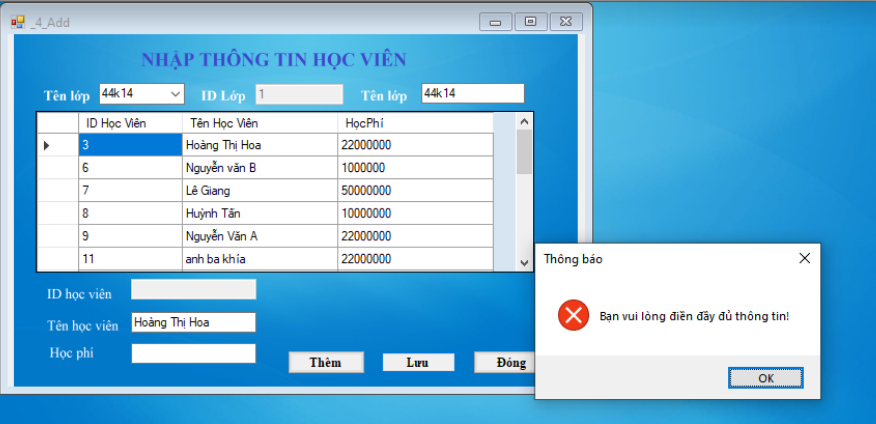
* Khi đã mở form Add, xuất hiện các textbox hiển thị Mã học viên, Tên học viên, Lớp, Học phí.



* Đầu tiên, bấm vào combobox **Lớp** để chọn lớp, sau đó chọn nút **Thêm** rồi **nhập Tên và Học phí** vào nhấn nút **Lưu** để lưu kết quả, trên DataGridView sẽ hiện lên thông tin học viên đã thêm.

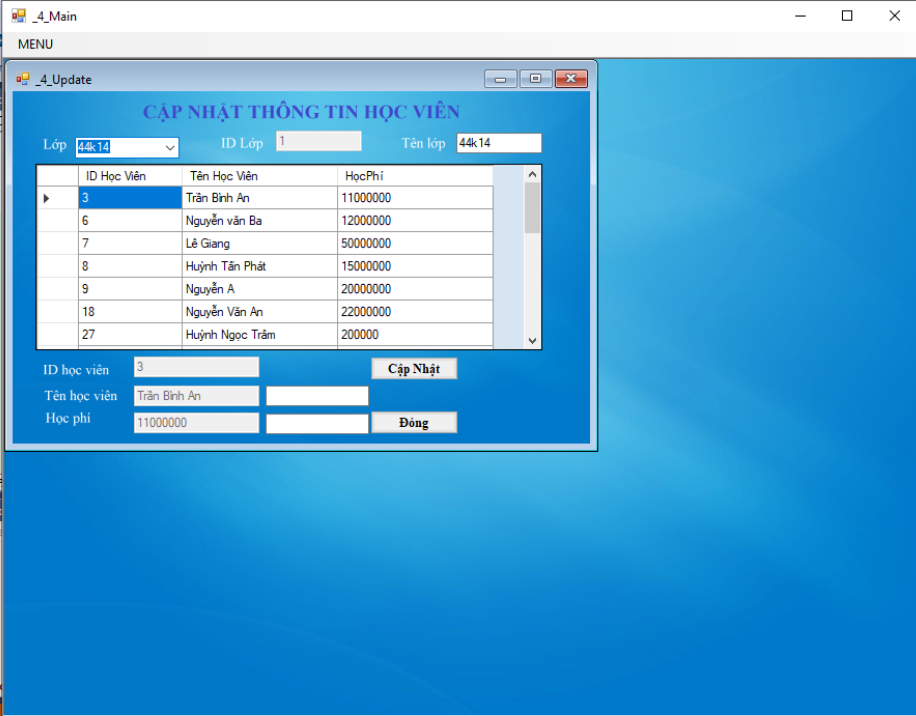
* Nếu **chưa** nhập đầu đủ Tên và Học phí thì sẽ xuất hiện thông báo “Bạn vui lòng điền đầy đủ thông tin”



* Nếu nhập thông tin hợp lệ sẽ có thông báo “Thêm học viên thành công”

3.4 Form Update học viên

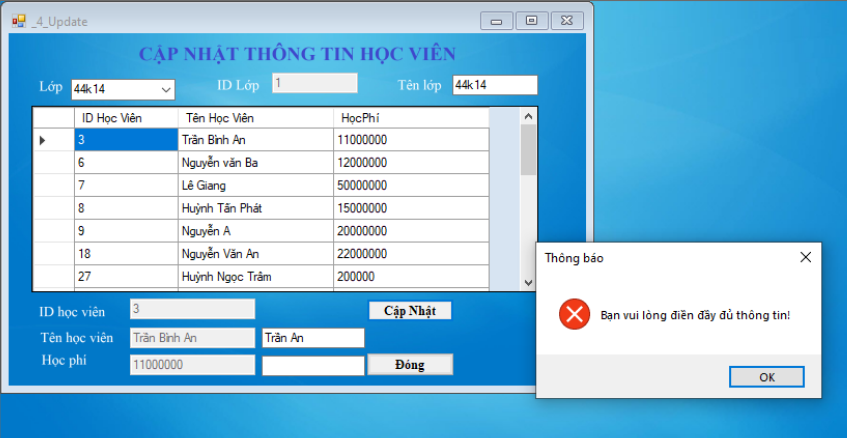
3.4.1 Cấu hình Form

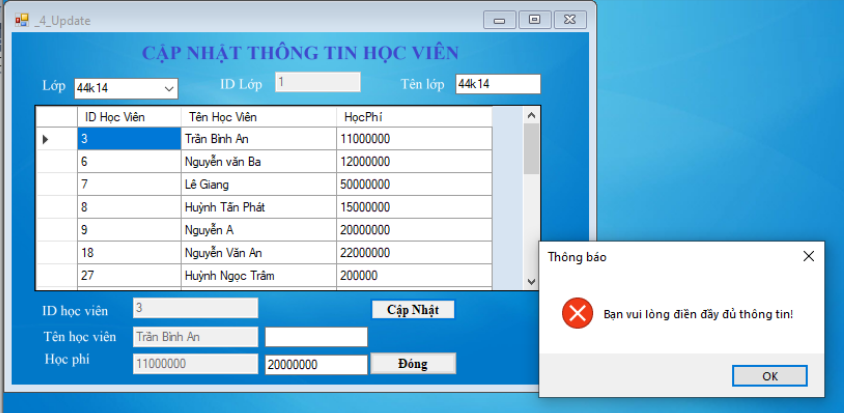


Hình 15: Form Update

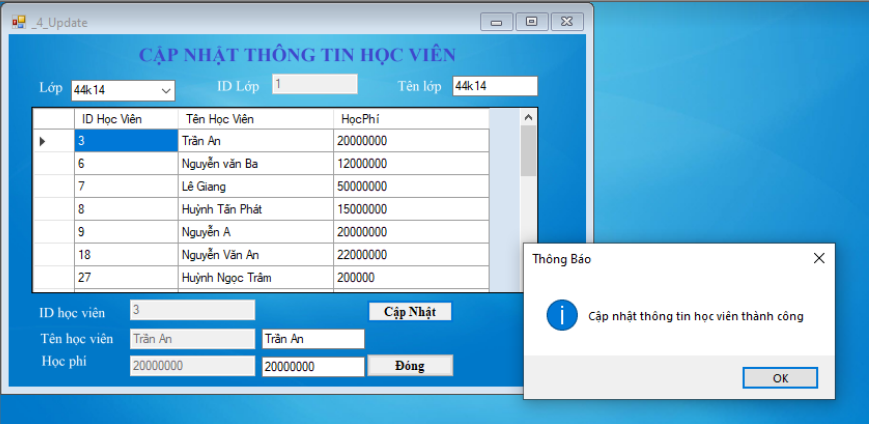
3.4.2 Xử lý sự kiện Form Update

* Khi đã mở form Update, xuất hiện các textbox hiển thị Mã học viên, Tên học viên, Lớp, Học phí.
* Đầu tiên, bấm vào combobox **Lớp** để chọn lớp, sau đó **sửa Tên và Học phí** vào nhấn nút **Cập nhật** để lưu kết quả, trên DataGridView sẽ hiện lên thông tin học viên đã cập nhật.
* Nếu **chưa** nhập đầu đủ Tên và Học phí thì sẽ xuất hiện thông báo “Vui lòng điền đầy đủ thông tin”





* Nếu nhập thông tin hợp lệ sẽ có thông báo “Cập nhật học viên thành công”



3.5 Form Search học viên

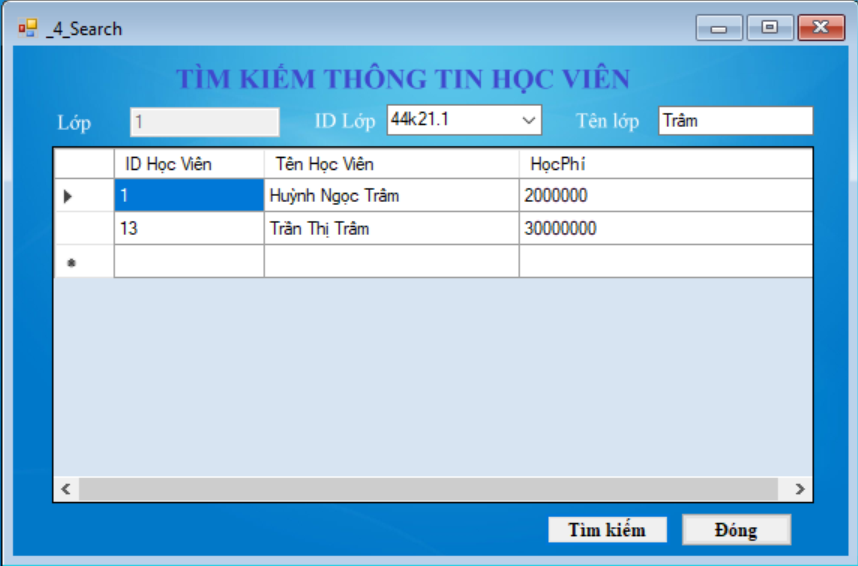
3.5.1 Cấu hình Form



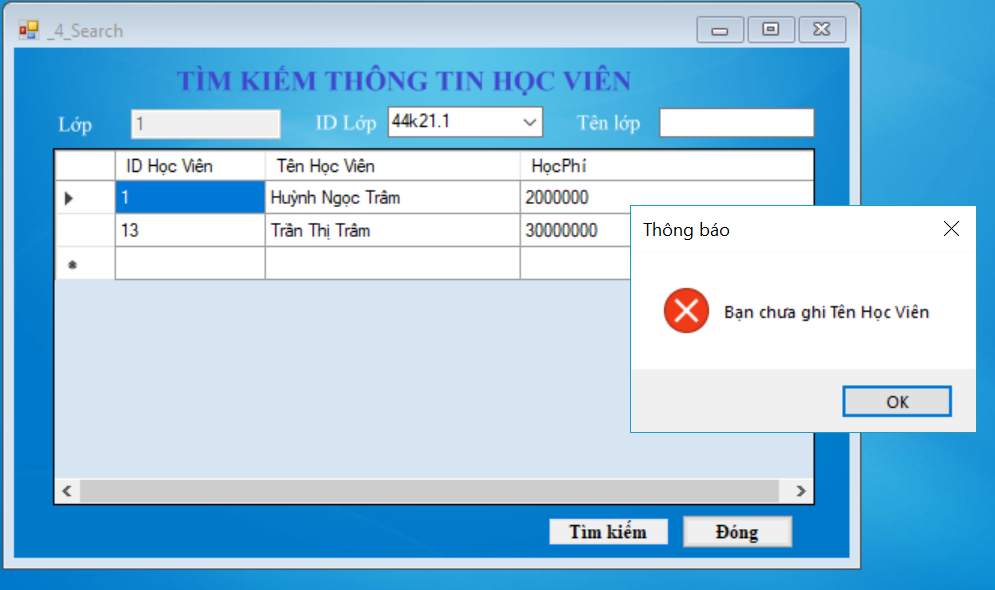
Hình 16: Form Search

3.5.2 Xử lý sự kiện Form Search

* Khi đã mở form Search, xuất hiện các textbox hiển thị Mã học viên, Tên học viên, và combobox Lớp
* Đầu tiên, bấm vào combobox **Lớp** để chọn lớp, sau đó **nhập Tên học viên** nhấn nút **Tìm kiếm**, trên DataGridView sẽ hiện lên thông tin học viên cần tìm.
* Nếu nhập **không đầy đủ** Tên học viên vẫn có thể tìm trong danh sách có tên tương tự.



* Nếu **không** nhập Tên thì sẽ xuất hiện thông báo “Bạn chưa ghi tên Học viên”.



3.6 Form Delete

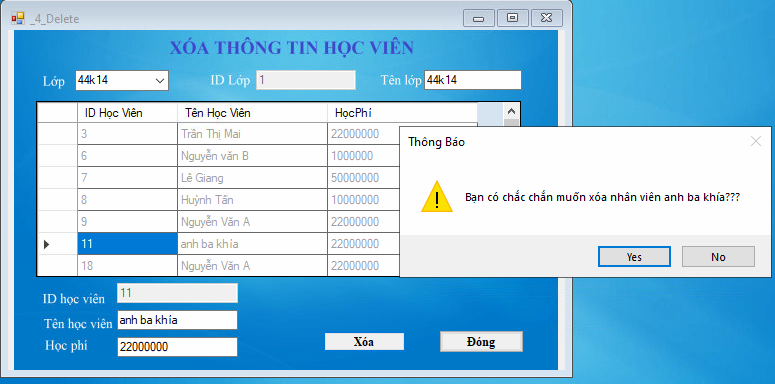
3.6.1 Cấu hình Form



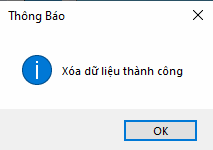
Hình 17: Form Delete

3.6.2 Xử lý sự kiện Form Delete

* Khi đã mở form Delete, xuất hiện các textbox hiển thị Mã học viên, Tên học viên, và combobox Lớp
* Đầu tiên, bấm vào combobox **Lớp** để chọn lớp, sau đó **nhấn vào Tên học viên** nhấn nút **Xóa**, hiện lên thông báo “Bạn có chắc muốn xóa?” và nhấn **yes**, trên DataGridView sẽ không hiện lên thông tin học viên.

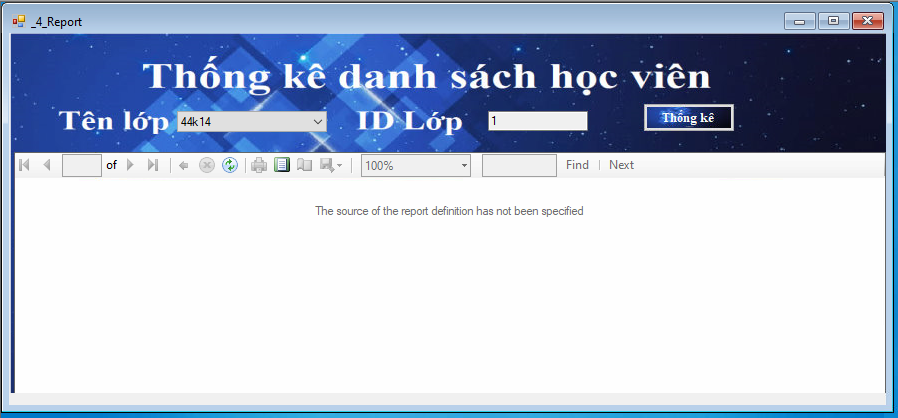


* Nếu nhấn “OK” sẽ hiện thông báo



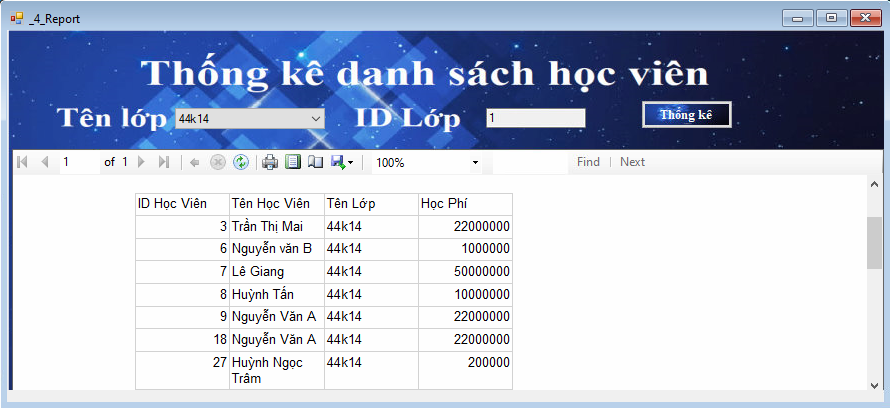
3.6 Form Thống kê học viên

3.6.1 Cấu hình Form

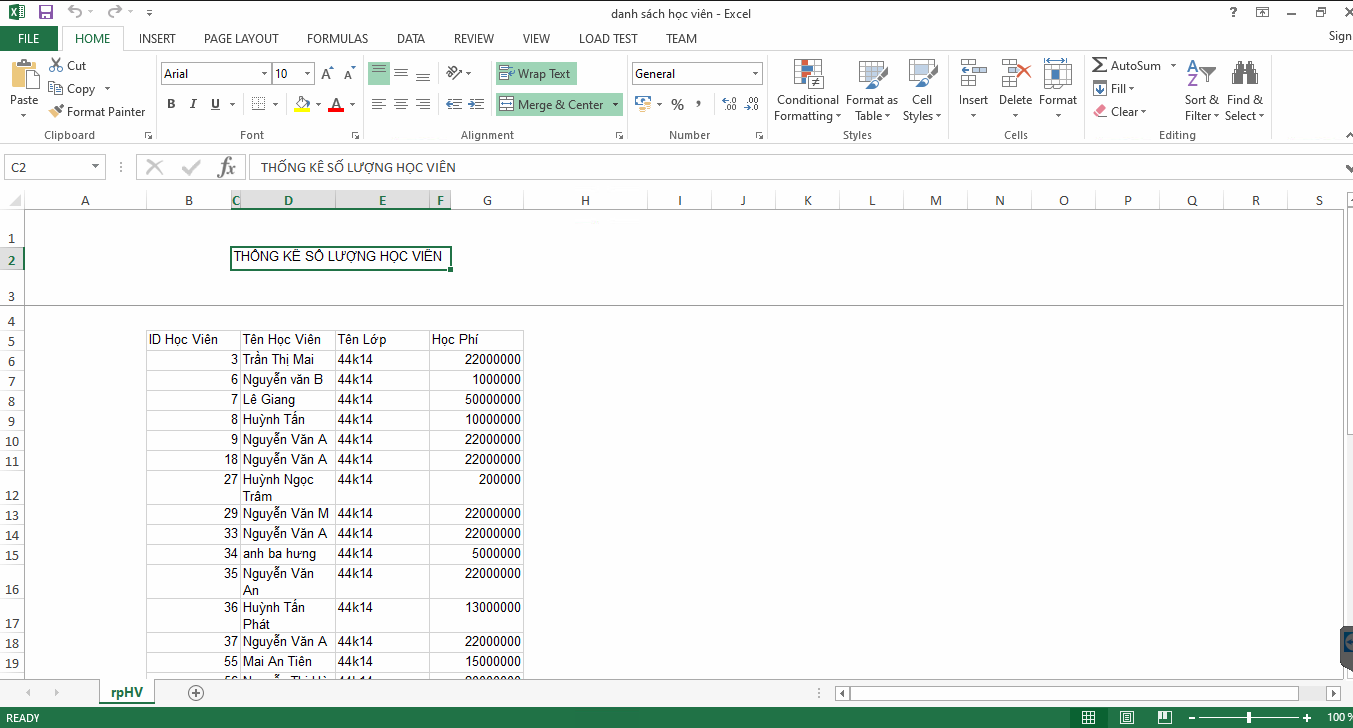


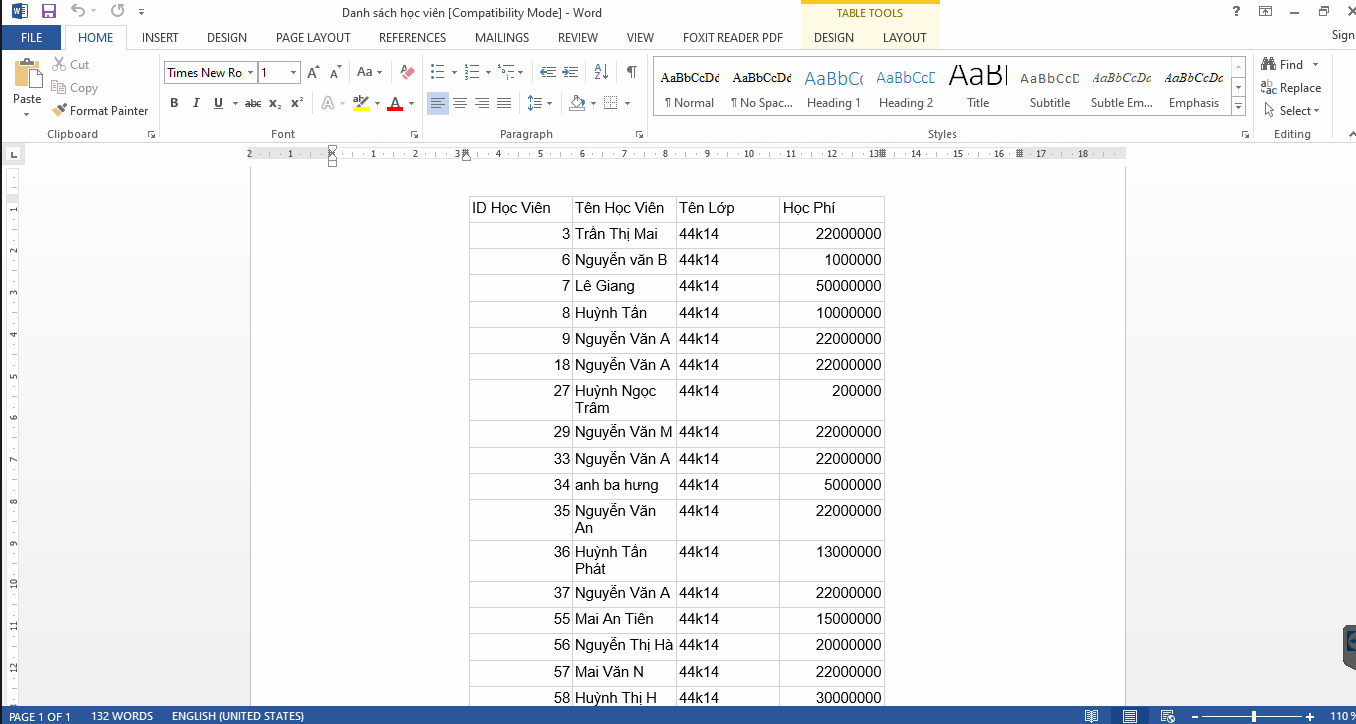
3.6.2 Xử lý sự kiện Form Thống kê

* Khi đã mở form Report, xuất hiện combobox Lớp.
* Đầu tiên, bấm vào combobox **Lớp** để chọn lớp, sau đó **nhấn Thống kê**, trên bảng sẽ hiện lên thông tin học viên được thống kê theo lớp.



* Có thể xuất file Word hoặc Excel:





KẾT LUẬN

4.1 Kết quả đạt được

* Phân tích và thiết kế được hệ thống quản lý sinh viên với dữ liệu và chức năng đơn giản.
* Vẽ được các sơ đồ Usecase, Activity, Class để mô tả hệ thống.
* Thiết kế và thực thi được cơ sở dữ liệu, bảng, liên kết các bảng và nhập dữ liệu tương ứng.
* Thiết kế được các StoredProcedure đơn giản trong SQL, hoàn thành đúng các yêu cầu giáo viên mong muốn.
* Thiết kế được các Form và thành lập được code để chạy thành công chương trình.
* Biết sửa các lỗi gặp phải khi chạy hệ thống, cũng như biết tối ưu hóa code để có thể logic hơn.
* Hoàn thiện được hệ thống quản lý sinh viên với dữ liệu và các chức năng đơn giản, thiết kế và bố cục hoàn chỉnh.
* Hoàn thiện báo cáo với đầy đủ các nội dung đã đề ra.
* Kỹ năng làm việc nhóm được nâng cao hơn và kĩ năng tự học hỏi cũng được trao dồi.

4.2 Những ưu điểm của hệ thống

* Hệ thống có thể thực hiện các chức năng: tìm kiếm, thêm, sửa, xoá, cập nhật và thống kê.
* Người quản lý có thể sử dụng hệ thống để thao tác bất kỳ chức năng nào.

4.3 Những nhược điểm của hệ thống

* Do thời gian thực tập và khả năng còn hạn hẹp, hệ thống quản lý sinh viên còn đơn giản và thiếu sót.

TÀI LIỆU THAM KHẢO

* Các video và bài giảng mà giáo viên hướng dẫn cung cấp.
* Tham khảo thêm các trang web trên mạng.

**PHỤ LỤC**

* + - 1. **Viết code cho Form Đăng nhập**
* Tạo điều kiện cho Form:

private void dangnhap()

{

if (Tendn.Text.Length == 0 && Mk.Text.Length == 0)

MessageBox.Show("Bạn chưa đăng nhập!", "Thông báo", MessageBoxButtons.OK,MessageBoxIcon.Error);

else

if (this.Tendn.Text.Length == 0)

MessageBox.Show("Bạn chưa nhập tên đăng nhập!","Thông báo", MessageBoxButtons.OK,MessageBoxIcon.Error);

else

if (this.Mk.Text.Length == 0)

MessageBox.Show("Bạn chưa nhập mật khẩu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

else

if (this.Tendn.Text == "admin" && this.Mk.Text == "admin")

MessageBox.Show("Đăng nhập thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

else

MessageBox.Show("Tài khoản hoặc Mật khẩu của bạn không đúng!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

* Tạo Tên và mật khẩu:

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

{

\_4\_Main fr = new \_4\_Main();

if (this.Tendn.Text == "admin" && this.Mk.Text == "admin")

{

fr.Show();

}

dangnhap();

}

* Thoát:

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

{

Application.Exit();

}

**2. Viết code cho Form MDI**

* Liên kết với các Form khác:

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

{

bool IsOpen = false;

foreach (Form f in Application.OpenForms)

{

if (f.Text == "\_4\_Add")

{

IsOpen = true;

f.Focus();

break;

}

}

if (IsOpen == false)

{

\_4\_Add f2 = new \_4\_Add();

f2.MdiParent = this;

f2.Show();

}

}

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

{

bool IsOpen = false;

foreach (Form f in Application.OpenForms)

{

if (f.Text == "\_4\_Update")

{

IsOpen = true;

f.Focus();

break;

}

}

if (IsOpen == false)

{

\_4\_Update f3 = new \_4\_Update();

f3.MdiParent = this;

f3.Show();

}

}

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

{

bool IsOpen = false;

foreach (Form f in Application.OpenForms)

{

if (f.Text == "\_4\_Search")

{

IsOpen = true;

f.Focus();

break;

}

}

if (IsOpen == false)

{

\_4\_Search f4 = new \_4\_Search();

f4.MdiParent = this;

f4.Show();

}

}

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

{

bool IsOpen = false;

foreach (Form f in Application.OpenForms)

{

if (f.Text == "\_4\_Delete")

{

IsOpen = true;

f.Focus();

break;

}

}

if (IsOpen == false)

{

\_4\_Delete f5 = new \_4\_Delete();

f5.MdiParent = this;

f5.Show();

}

}

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

{

bool IsOpen = false;

foreach (Form f in Application.OpenForms)

{

if (f.Text == "\_4\_REPORT")

{

IsOpen = true;

f.Focus();

break;

}

}

if (IsOpen == false)

{

\_4\_Report f7 = new \_4\_Report();

f7.MdiParent = this;

f7.Show();

}

}

* Tạo Form đóng tất cả các form không hoạt động:

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

{

foreach (Form frm in this.MdiChildren)

{

if (!frm.Focused)

{

frm.Visible = false;

frm.Dispose();

}

}

}

1. **Viết code cho Form Add**

* Kết nối CSDL:

public partial class \_4\_Add : Form

{

public \_4\_Add()

{

InitializeComponent();

}

SqlConnection conn = new SqlConnection(@"Data Source=PC\NGUYENPHIHOANG;Initial Catalog=QLHocVien;Integrated Security=True");

* Đưa CSDL lên Textbox và DataGripView

private void \_4\_Add\_Load(object sender, EventArgs e)

{

try

{

var cmd = new SqlCommand("DMLOP\_SELECTALL", conn);

cmd.CommandType = CommandType.StoredProcedure;

var dap = new SqlDataAdapter(cmd);

var table = new DataTable();

dap.Fill(table);

cbTenlop.DisplayMember = "TenLop";

cbTenlop.ValueMember = "IDLop";

cbTenlop.DataSource = table;

textIDlop.DataBindings.Clear();

textIDlop.DataBindings.Add("Text", cbTenlop.DataSource, "IDLop");

textTenlop.DataBindings.Clear();

textTenlop.DataBindings.Add("Text", cbTenlop.DataSource, "TenLop");

}

catch

{

MessageBox.Show("Load dữ liệu thất bại. Kiểm tra lại SQL Server trên máy tính của bạn", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

try

{

var cmd = new SqlCommand("HOCVIEN\_SELECTID", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@IDLop", SqlDbType.Int).Value = Convert.ToInt32(cbTenlop.SelectedValue);

var dap = new SqlDataAdapter(cmd);

var table = new DataTable();

dap.Fill(table);

dsHocvien.DataSource = table;

textIDhv.DataBindings.Clear();

textIDhv.DataBindings.Add("Text", dsHocvien.DataSource, "IDHocVien");

textTenhv.DataBindings.Clear();

textTenhv.DataBindings.Add("Text", dsHocvien.DataSource, "TenHocVien");

texthp.DataBindings.Clear();

texthp.DataBindings.Add("Text", dsHocvien.DataSource, "HocPhi");

}

catch

{

MessageBox.Show("Load dữ liệu thất bại. Kiểm tra lại SQL Server trên máy tính của bạn", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

* Tạo code cho nút Thêm

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

{

textIDhv.Text = "";

textTenhv.Text = "";

texthp.Text = "";

textTenhv.Focus();

addNV = true;

}

* Tạo code cho nút Lưu

Boolean addNV = false;

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

{

if (this.textTenhv.TextLength ==0 || this.texthp.TextLength== 0)

{

MessageBox.Show("Bạn vui lòng điền đầy đủ thông tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

try

{

conn.Open();

var cmd = new SqlCommand("HOCVIEN\_ADD", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@TenHocVien", SqlDbType.NVarChar).Value = textTenhv.Text;

cmd.Parameters.Add("@HocPhi", SqlDbType.Int).Value = texthp.Text;

cmd.Parameters.Add("@IDLop", SqlDbType.Int).Value = Convert.ToInt32(cbTenlop.SelectedValue);

cmd.ExecuteNonQuery();

conn.Close();

cbTenlop\_SelectedIndexChanged(sender, e);

MessageBox.Show("Thêm học viên thành công", "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

// tat che do them vao khi da them vao xong

addNV = false;

}

catch

{

MessageBox.Show("Thêm học viên thất bại!", "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

* Tạo code cho nút Thoát:

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

{

this.Close();

}

1. **Viết code cho Form Update**

* Kết nối CSDL

public partial class \_4\_Update : Form

{

public \_4\_Update()

{

InitializeComponent();

}

SqlConnection conn = new SqlConnection(@"Data Source=PC\NGUYENPHIHOANG;Initial Catalog=QLHocVien;Integrated Security=True");

* Đưa CSDL lên các Textbox và DataGripView

private void \_4\_Update\_Load(object sender, EventArgs e)

{

try

{

var cmd = new SqlCommand("DMLOP\_SELECTALL", conn);

cmd.CommandType = CommandType.StoredProcedure;

var dap = new SqlDataAdapter(cmd);

var table = new DataTable();

dap.Fill(table);

cbTenlop2.DisplayMember = "TenLop";

cbTenlop2.ValueMember = "IDLop";

cbTenlop2.DataSource = table;

textIDlop2.DataBindings.Clear();

textIDlop2.DataBindings.Add("Text", cbTenlop2.DataSource, "IDLop");

textTenlop2.DataBindings.Clear();

textTenlop2.DataBindings.Add("Text", cbTenlop2.DataSource, "TenLop");

}

catch

{

MessageBox.Show("Load dữ liệu thất bại. Kiểm tra lại SQL Server trên máy tính của bạn", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void cbTenlop2\_SelectedIndexChanged(object sender, EventArgs e)

{

try

{

var cmd = new SqlCommand("HOCVIEN\_SELECTID", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@IDLop", SqlDbType.Int).Value = Convert.ToInt32(cbTenlop2.SelectedValue);

var dap = new SqlDataAdapter(cmd);

var table = new DataTable();

dap.Fill(table);

dsHocvien2.DataSource = table;

textIDhv2.DataBindings.Clear();

textIDhv2.DataBindings.Add("Text", dsHocvien2.DataSource, "IDHocVien");

textTenhv2.DataBindings.Clear();

textTenhv2.DataBindings.Add("Text", dsHocvien2.DataSource, "TenHocVien");

texthp2.DataBindings.Clear();

texthp2.DataBindings.Add("Text", dsHocvien2.DataSource, "HocPhi");

}

catch

{

MessageBox.Show("Load dữ liệu thất bại. Kiểm tra lại SQL Server trên máy tính của bạn", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

* Tạo code cho nút Update

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

{

if (this.textTenhv3.TextLength != 0 && this.texthp3.TextLength != 0)

{

try

{

conn.Open();

var cmd = new SqlCommand("HOCVIEN\_UPDATE", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@TenHocVien", SqlDbType.NVarChar).Value = textTenhv3.Text;

cmd.Parameters.Add("@HocPhi", SqlDbType.Int).Value = Convert.ToInt32(texthp3.Text);

cmd.Parameters.Add("@IDHocVien", SqlDbType.Int).Value = Convert.ToInt32(textIDhv2.Text);

cmd.ExecuteNonQuery();

conn.Close();

cbTenlop2\_SelectedIndexChanged(sender, e);

MessageBox.Show("Cập nhật thông tin học viên thành công", "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch

{

MessageBox.Show("Cập nhật thông tin học viên thất bại!", "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

MessageBox.Show("Bạn vui lòng điền đầy đủ thông tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

* Tạo code cho nút Thoát

private void btnclose2\_Click(object sender, EventArgs e)

{

this.Close();

}

1. **Viết code cho Form Search**

* Kết nối CSDL

public partial class \_4\_Search : Form

{

public \_4\_Search()

{

InitializeComponent();

}

SqlConnection conn = new SqlConnection(@"Data Source=PC\NGUYENPHIHOANG;Initial Catalog=QLHocVien;Integrated Security=True");

* Đưa CSDL lên DataGripView, Combox, Textbox

private void \_4\_Search\_Load(object sender, EventArgs e)

{

try

{

var cmd = new SqlCommand("DMLOP\_SELECTALL", conn);

cmd.CommandType = CommandType.StoredProcedure;

var dap = new SqlDataAdapter(cmd);

var table = new DataTable();

dap.Fill(table);

cbTenlop4.DisplayMember = "TenLop";

cbTenlop4.ValueMember = "IDLop";

cbTenlop4.DataSource = table;

textIDlop4.DataBindings.Clear();

textIDlop4.DataBindings.Add("Text", cbTenlop4.DataSource, "IDLop");

}

catch

{

MessageBox.Show("Load dữ liệu thất bại. Kiểm tra lại SQL Server trên máy tính của bạn", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void cbTenlop4\_SelectedIndexChanged(object sender, EventArgs e)

{

try

{

var cmd = new SqlCommand("HOCVIEN\_SELECTID", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@IDLop", SqlDbType.Int).Value = Convert.ToInt32(cbTenlop4.SelectedValue);

var dap = new SqlDataAdapter(cmd);

var table = new DataTable();

dap.Fill(table);

dsHocvien4.DataSource = table;

}

catch

{

MessageBox.Show("Load dữ liệu thất bại. Kiểm tra lại SQL Server trên máy tính của bạn", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

* Tạo code nút Tìm kiếm

void timdata()

{

try

{

if (!textfind.Text.Equals(""))

{

var cmd = new SqlCommand();

cmd = conn.CreateCommand();

cmd.CommandText = "Select IDHocVien, TenHocVien, HocPhi from DMLOP inner join HOCVIEN on HOCVIEN.IDLop = DMLOP.IDLop where TenHocVien like N'%" + textfind.Text + "%' and TenLop=N'" + cbTenlop4.Text + "'";

var dap = new SqlDataAdapter(cmd);

var table = new DataTable();

dap.SelectCommand = cmd;

table.Clear();

dap.Fill(table);

dsHocvien4.DataSource = table;

}

else

{

MessageBox.Show("Bạn chưa ghi Tên Học Viên", "Thông báo",MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch

{

MessageBox.Show("Tìm kiếm bị lỗi!", "THÔNG BÁO", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

timdata();

}

* Tạo code cho nút Thoát

private void btnclose4\_Click(object sender, EventArgs e)

{

this.Close();

}

1. **Viết code cho Form Delete**

* Kết nối CSDL

public partial class \_4\_Delete : Form

{

public \_4\_Delete()

{

InitializeComponent();

}

SqlConnection conn = new SqlConnection(@"Data Source=PC\NGUYENPHIHOANG;Initial Catalog=QLHocVien;Integrated Security=True");

* Đưa dữ liệu lên Textbox, ComboBox, DataGripView

private void \_4\_Delete\_Load(object sender, EventArgs e)

{

try

{

var cmd = new SqlCommand("DMLOP\_SELECTALL", conn);

cmd.CommandType = CommandType.StoredProcedure;

var dap = new SqlDataAdapter(cmd);

var table = new DataTable();

dap.Fill(table);

cbTenlop3.DisplayMember = "TenLop";

cbTenlop3.ValueMember = "IDLop";

cbTenlop3.DataSource = table;

textIDlop3.DataBindings.Clear();

textIDlop3.DataBindings.Add("Text", cbTenlop3.DataSource, "IDLop");

textTenlop3.DataBindings.Clear();

textTenlop3.DataBindings.Add("Text", cbTenlop3.DataSource, "TenLop");

}

catch

{

MessageBox.Show("Load dữ liệu thất bại. Kiểm tra lại SQL Server trên máy tính của bạn", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void cbTenlop3\_SelectedIndexChanged(object sender, EventArgs e)

{

try

{

var cmd = new SqlCommand("HOCVIEN\_SELECTID", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@IDLop", SqlDbType.Int).Value = Convert.ToInt32(cbTenlop3.SelectedValue);

var dap = new SqlDataAdapter(cmd);

var table = new DataTable();

dap.Fill(table);

dsHocvien3.DataSource = table;

textIDhv3.DataBindings.Clear();

textIDhv3.DataBindings.Add("Text", dsHocvien3.DataSource, "IDHocVien");

textTenhv3.DataBindings.Clear();

textTenhv3.DataBindings.Add("Text", dsHocvien3.DataSource, "TenHocVien");

texthp3.DataBindings.Clear();

texthp3.DataBindings.Add("Text", dsHocvien3.DataSource, "HocPhi");

}

catch

{

MessageBox.Show("Load dữ liệu thất bại. Kiểm tra lại SQL Server trên máy tính của bạn", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

* Tạo code cho nút Xóa

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

{

DialogResult lenh = MessageBox.Show(" Bạn có chắc chắn muốn xóa nhân viên " + textTenhv3.Text + "???",

"Thông Báo", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (lenh == DialogResult.Yes)

{

try

{

conn.Open();

var cmd = new SqlCommand("HOCVIEN\_DELETE", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add("@IDHocVien", SqlDbType.Int).Value = Convert.ToInt32(textIDhv3.Text);

cmd.ExecuteNonQuery();

conn.Close();

cbTenlop3\_SelectedIndexChanged(sender, e);

MessageBox.Show("Xóa dữ liệu thành công", "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch

{

MessageBox.Show("Xóa dữ liệu thất bại", "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

* Tạo code cho nút Thoát

private void btnclose2\_Click(object sender, EventArgs e)

{

this.Close();

}

1. **Viết code cho Form Report**

* Liên kết CSDL

public partial class \_4\_Report : Form

{

public \_4\_Report()

{

InitializeComponent();

}

SqlConnection conn = new SqlConnection(@"Data Source=PC\NGUYENPHIHOANG;Initial Catalog=QLHocVien;Integrated Security=True");

* Đưa dữ liệu vào các ComboBox, TextBox

private void \_4\_Report\_Load(object sender, EventArgs e)

{

var cmd = new SqlCommand("Select \* From DMLOP", conn);

cmd.CommandType = CommandType.Text;

var dap = new SqlDataAdapter(cmd);

var table = new DataTable();

dap.Fill(table);

cbLop.DisplayMember = "TenLop";

cbLop.ValueMember = "IDLop";

cbLop.DataSource = table;

txtIDLop.DataBindings.Clear();

txtIDLop.DataBindings.Add("Text", cbLop.DataSource, "IDLop");

this.reportViewer1.RefreshReport();

}

* Viết code cho nút Thống kê

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

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = Properties.Settings.Default.QLHocVienConnectionString;

SqlCommand cmd = new SqlCommand();

cmd.CommandText = "TKHV\_Report";

cmd.CommandType = CommandType.StoredProcedure;

cmd.Connection = conn;

cmd.Parameters.Add(new SqlParameter("@TenLop", cbLop.Text));

DataSet ds = new DataSet();

SqlDataAdapter dap = new SqlDataAdapter(cmd);

dap.Fill(ds);

reportViewer1.ProcessingMode = ProcessingMode.Local;

reportViewer1.LocalReport.ReportPath = "rpHV.rdlc";

ReportDataSource rds = new ReportDataSource();

rds.Name = "dsReport";

rds.Value = ds.Tables[0];

reportViewer1.LocalReport.DataSources.Clear();

reportViewer1.LocalReport.DataSources.Add(rds);

reportViewer1.RefreshReport();

}

}

1. **Viết code cho Form Đăng Xuất**

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

{

if (MessageBox.Show(

"Bạn có chắc chắn muốn đăng xuất không?", "Đăng Xuất", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

Close();

System.Windows.Forms.Application.Exit();

}

}