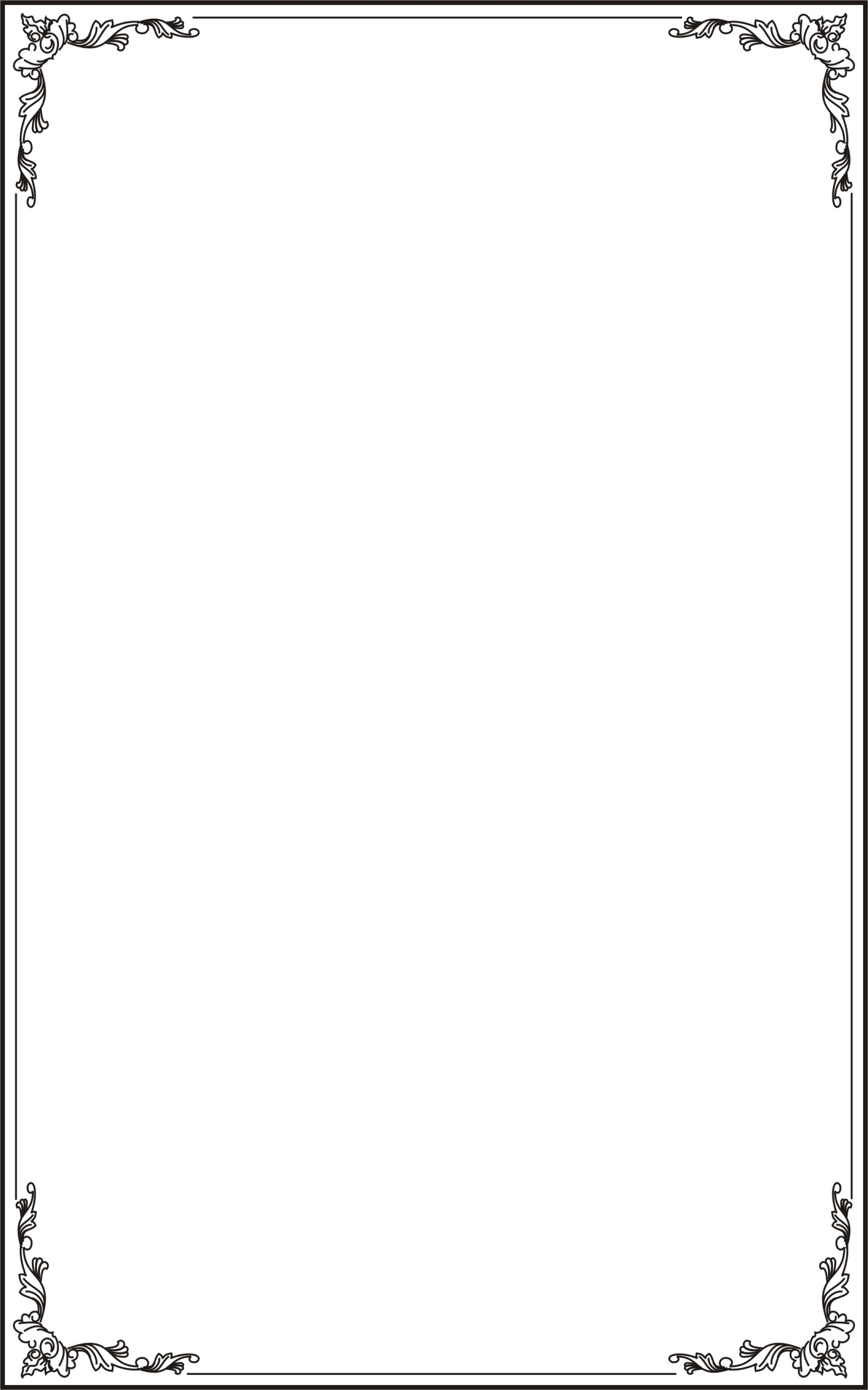
|  |
| --- |
| **DỰ ÁN 1** |
| NGÀNH CÔNG NGHỆ THÔNG TIN (ỨNG DỤNG PHẦN MỀM) |



**CAO ĐẲNG THỰC HÀNH FPT POLYTECHNIC**

**ĐÀ NẴNG**

|  |
| --- |
| ĐÀ NẴNG 2019 |



|  |
| --- |
| GIẢNG VIÊN HƯỚNG DẪN: NGUYỄN NGỌC ANH |
| SINH VIÊN THỰC HIỆN |
| 1. Nguyễn Viết Thắng (PD03196), Trưởng nhóm 2. Nguyễn Tấn Châu (PD03169) 3. Nguyễn Đan Trường (PD03192) |

MỤC LỤC

[1 Phân tích 2](#_Toc520132211)

[1.1 Hiện trạng 2](#_Toc520132212)

[1.2 Yêu cầu hệ thống 2](#_Toc520132213)

[1.3 Use case 2](#_Toc520132214)

[2 Thiết kế 2](#_Toc520132215)

[2.1 Mô hình triển khai 2](#_Toc520132216)

[2.2 Thiết kế CSDL 2](#_Toc520132217)

[2.2.1 Sơ đồ quan hệ thực thể 2](#_Toc520132218)

[2.2.2 Thiết kế chi tiết các thực thể 2](#_Toc520132219)

[2.3 Thiết kế giao diện 2](#_Toc520132220)

[2.3.1 Sơ đồ tổ chức giao diện 2](#_Toc520132221)

[2.3.2 Thiết kế giao diện cho các chức năng nghiệp vụ 2](#_Toc520132222)

[3 Thực hiện viết mã 2](#_Toc520132223)

[3.1 Viết mã tạo CSDL 2](#_Toc520132224)

[3.1.1 Tạo CSDL 2](#_Toc520132225)

[3.1.2 SQL truy vấn và thao tác 2](#_Toc520132226)

[3.1.3 Các thủ tục lưu tổng hợp thống kê 2](#_Toc520132227)

[3.2 Lập trình JDBC 3](#_Toc520132228)

[3.2.1 Lớp hỗ trợ 3](#_Toc520132229)

[3.2.2 Model class - Các lớp mô tả dữ liệu 3](#_Toc520132230)

[3.2.3 DAO Class - Các lớp truy xuất dữ liệu 3](#_Toc520132231)

[3.3 Viết mã cho ứng dụng 3](#_Toc520132232)

[3.3.1 Xử lý Form X 3](#_Toc520132233)

[3.3.2 Xử lý Form Y 3](#_Toc520132234)

[4 Kiểm thử 3](#_Toc520132235)

[4.1 Kiểm thử form X 3](#_Toc520132236)

[4.2 Kiểm thử form Y 3](#_Toc520132237)

[5 Đóng gói và triển khai 3](#_Toc520132238)

[5.1 Hướng dẫn chuyển đổi jar thành exe 3](#_Toc520132239)

[5.2 Hướng dẫn cài đặt triển khai 3](#_Toc520132240)

[5.3 Hướng dẫn sử dụng phần mềm 3](#_Toc520132241)

# Phân tích

## Hiện trạng

Hiện nay, quản lý điểm là một công việc hết sức quan trọng đối với các trường học. Công việc đó hiện đang còn làm rất thủ công tại một số trường và chính vì thế nó mang lại hiệu quả không cao. Thực tế, hiện nay trường Đại học, Cao đẳng vẫn đang dùng hệ thống quản lý điểm trên Microsoft Excel. Công việc hằng ngày bao gồm:

* Nhập điểm cho sinh viên, sửa chữa thông tin về điểm
* In bảng điểm, in danh sách sinh viên đỗ, trượt, đạt học bổng…
* Lưu trữ thông tin các bảng điểm của Sinh viên…

Công việc quản lý còn hết sức thủ công và đòi hỏi nhiều kỹ năng của người quản lý.

Ví dụ: Hằng ngày, khi người quản lý nhập đểm cho sinh viên, tính toán, in danh sách theo yêu cầu của nhà trường: những sinh viên đỗ trượt đạt học bổng…. thời gian nhập thông tin mất nhiều thời gian, việc theo dõi thống kê, tổng hợp dễ bị nhầm lẫn, khó đảm bảo độ tin cậy…

## Yêu cầu hệ thống

* Hệ thống sử dụng hệ quản trị cơ sở dữ liệu đủ lớn để đáp ứng số lượng sinh viên ngày càng tăng.
* Máy chủ có khả năng tính toán nhanh, chính xác, lưu trữ lâu dài, bảo mật.
* Hệ thống mạng đáp ứng khả năng truy cập lớn.
* Đưa ra tổng kết, xếp loại sinh viên qua hệ thống, tự động.
* Thông tin có tính đồng bộ, phân quyền quản lý chặt chẽ.
* Yêu cầu về bảo mật

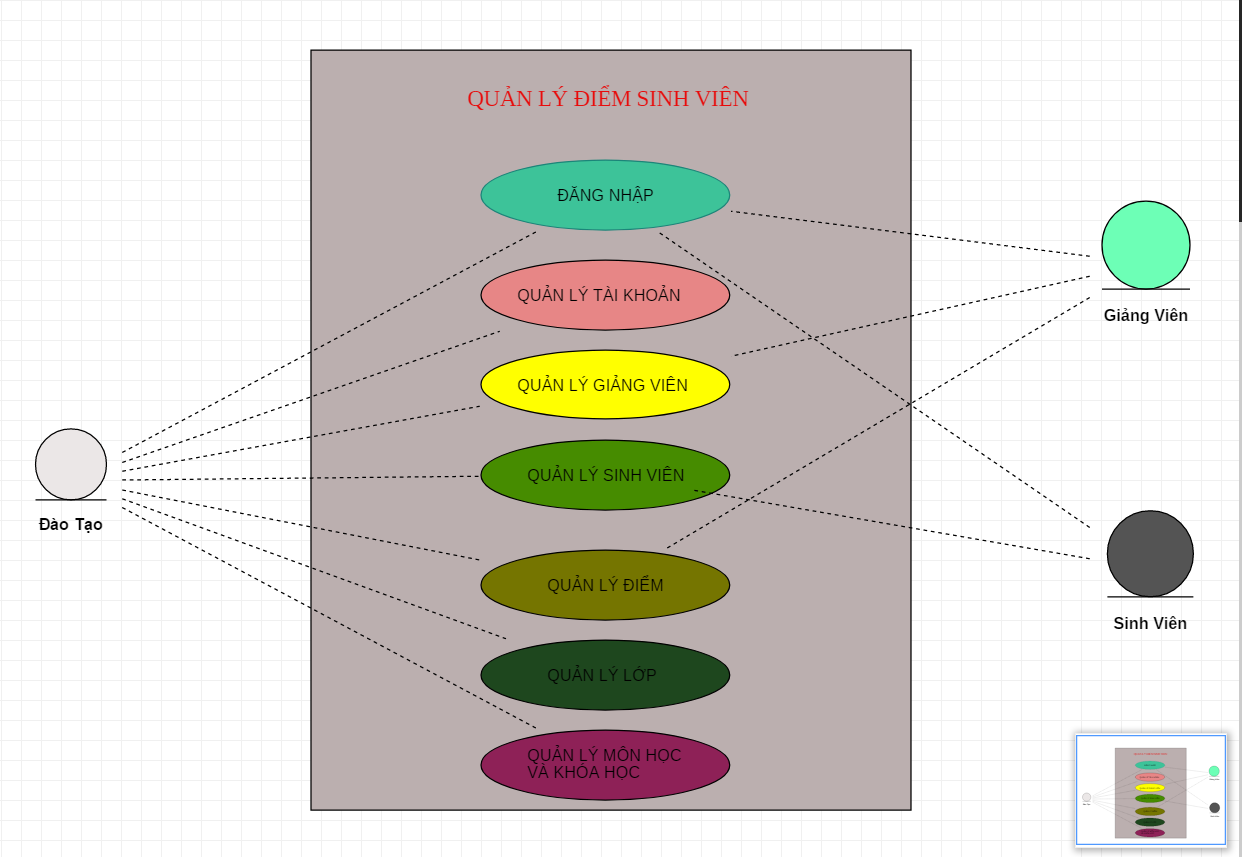
o Tất cả mọi thành viên phải đăng nhập mới sử dụng được phần mềm

o Trưởng phòng đào tạo được phép thực hiện tất cả các chức năng

o Giảng viên và sinh viên không được phép xóa dữ liệu và cũng không được xem thông tin về doanh thu

## Use case

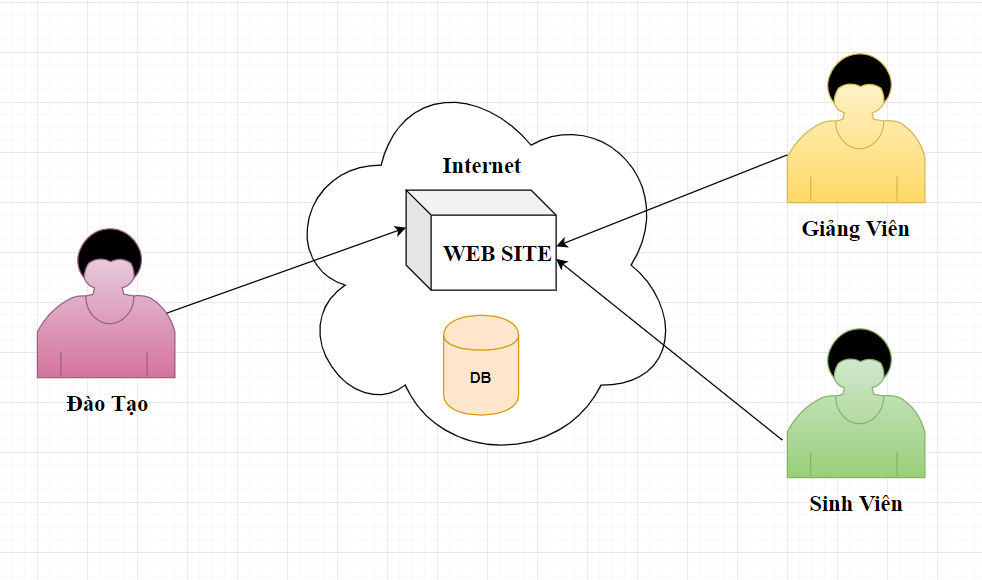
Use case là sơ đồ tổng quan về mặt chức năng và phân vai trò người sử dụng. Dựa vào yêu cầu hệ thống của khách hàng, chúng ta có thể phác thảo sơ đồ use case như sau.



Sơ đồ use case

# Thiết kế

## Mô hình triển khai



Mô hình triển khai

YÊU CẦU HỆ THỐNG

* Hệ điều hành: Windows 8.1+
* Web server: Apache hỗ trợ PHP 5+
* Hệ quản trị CSDL: MySQL 5+
* Kiến trúc mạng: Internet/Intranet

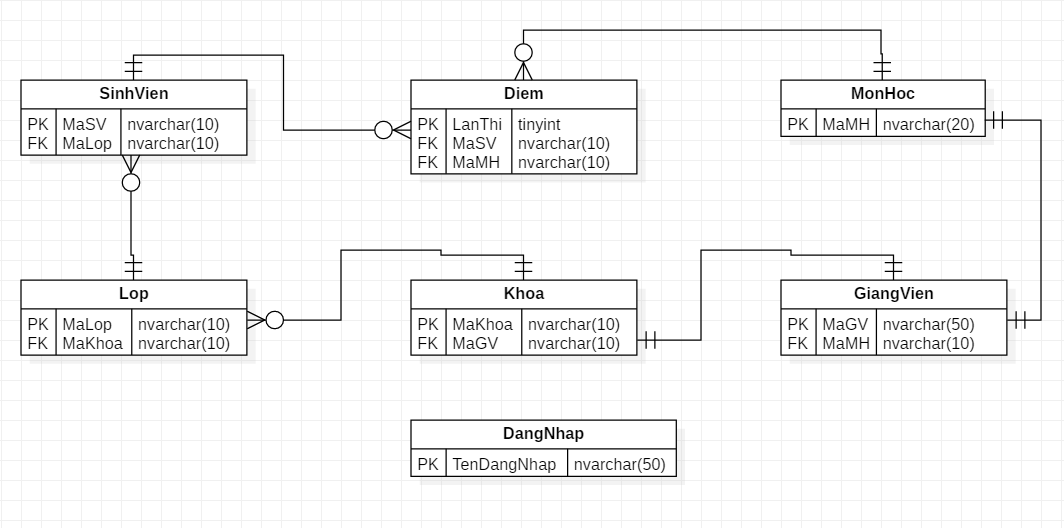
## Thiết kế CSDL

### Sơ đồ quan hệ thực thể

Khảo sát yêu cầu hệ thống quản lý đào tạo của khách hàng, chúng ta dễ dàng tìm thấy các thực thể tồn tại trong hệ thống gồm:

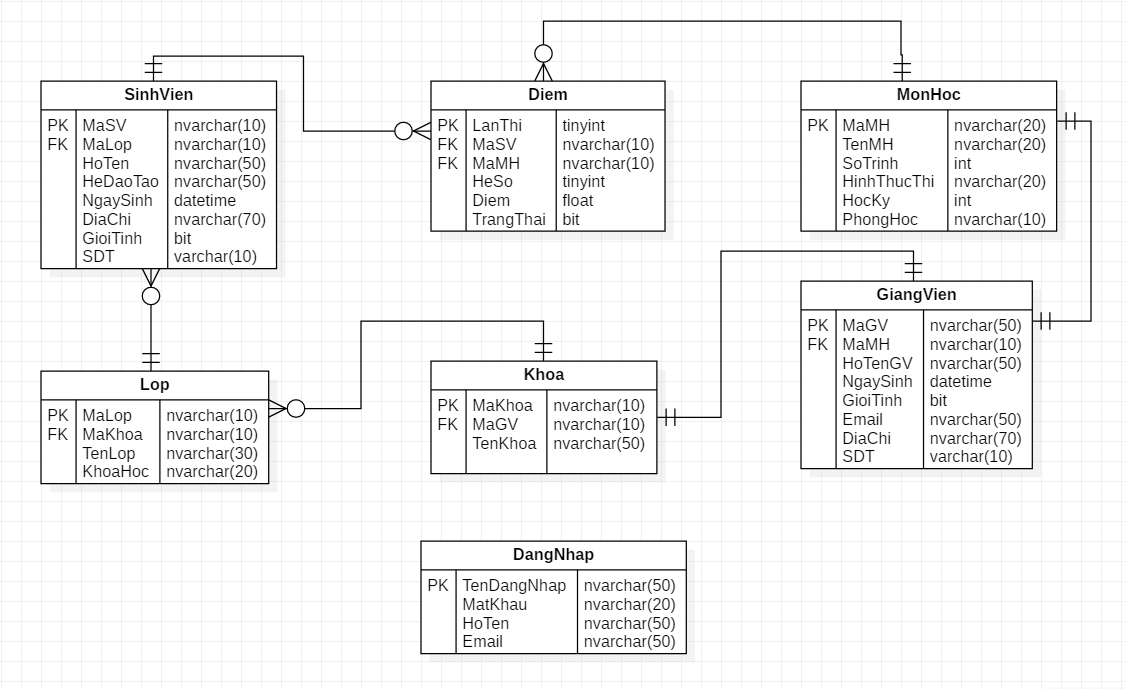
* Khoa: mô tả thông tin khóa học
* GiangVien: mô tả thông tin giảng viên
* Diem: mô tả thông tin điểm của sinh viên
* SinhVien: mô tả thông tin sinh viên
* LopHoc: mô tả thông tin lớp học
* DaoTao: mô tả thông tin nhân viên phòng đào tạo

Phân tích mối quan hệ giữa các thực thể, chúng ta có sơ đồ quan thệ thực thể như sau:



ERD Diagram level 1

Nghiên cứu các thông tin cần thiết mỗi thực thể trong hệ thống, chúng ta đưa ra cấu trúc thông tin mỗi thực thể như sau:



ERD Diagram level 2

### Thiết kế chi tiết các thực thể

#### Môn Học

Bảng quản lý thông tin môn học với cấu trúc sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaMH | VARCHAR(10) | PK, NOT NULL | Mã môn học |
| TenMH | VARCHAR(20) | NOT NULL | Tên môn học |
| SoTrinh | INT |  | Số trình |
| HinhThucThi | VARCHAR(20) |  | Hình thức thi |
| HocKy | INT |  | Học Kỳ |
| PhongHoc | VARCHAR(10) |  | Phòng học |

#### Giảng Viên

Bảng GiangVien quản lý thông tin các giảng viên với cấu trúc sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaGV | VARCHAR(10) | PK, NOT NULL | Mã giảng viên |
| HoTenGV | VARCHAR(70) | NOT NULL | Họ Tên giảng viên |
| MaMH | VARCHAR(10) | FK, NOT NULL | Mã môn học |
| NgaySinh | datetime |  | Ngày sinh |
| GioiTinh | bit |  | Giới tính |
| Email | VARCHAR(50) |  | Email |
| DiaChi | VARCHAR(70) |  | Địa chỉ |
| SĐT | VARCHAR(10) |  | Số điện thoại |

#### Khóa

Bảng Khoa quản lý thông tin các khóa với cấu trúc sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaKhoa | VARCHAR(10) | PK, NOT NULL | Mã khóa |
| TenKhoa | VARCHAR(50) | NOT NULL | Tên khóa |
| SDT | VARCHAR(10) |  |  |
| MaGV | VARCHAR(10) | FK, NOT NULL | Mã giảng viên |

#### Sinh Viên

Bảng SinhVien quản lý thông tin các sinh viên với cấu trúc sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaSV | VARCHAR(10) | PK, NOT NULL | Mã sinh viên |
| HoTen | VARCHAR(50) | NOT NULL | Họ Tên sinh viên |
| MaLop | VARCHAR(10) | FK, NOT NULL | Mã lớp |
| HeDaoTao | VARCHAR(50) | NOT NULL | Hệ đào tạo |
| NgaySinh | datetime |  | Ngày sinh |
|  |  |  |  |
| DiaChi | VARCHAR(100) |  | Địa chỉ |
| SĐT | VARCHAR(10) |  | Số điện thoại |

#### Lớp

Bảng Lop quản lý thông tin các lớp với cấu trúc sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaLop | VARCHAR(10) | PK, NOT NULL | Mã lớp |
| TenLop | VARCHAR(50) | NOT NULL | Tên lớp |
| MaKhoa | VARCHAR(10) | FK, NOT NULL | Mã khóa |
| KhoaHoc | VARCHAR(20) |  | Khóa học |

#### Điểm

Bảng Diem quản lý thông tin các điểm với cấu trúc sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| LanThi | tinyint | PK, NOT NULL | Lần thi |
| MaSV | VARCHAR(10) | FK1, NOT NULL | Mã sinh viên |
| MaMH | VARCHAR(10) | Fk2, NOT NULL | Mã môn học |
| HeSo | tinyint |  | Hệ số |
| Diem | float |  | Điểm |
| TrangThai | bit |  | Trạng thái |

#### Đăng nhập

Bảng DangNhap quản lý thông tin đăng nhập với cấu trúc sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| TenDangNhap | VARCHAR(50) | PK, NOT NULL | Tên đăng nhập |
| MatKhau | VARCHAR(20) | NOT NULL | Mật khẩu |
| HoTen | VARCHAR(50) |  | Họ tên |
| Email | VARCHAR(50) | NOT NULL | email |

## Thiết kế giao diện

### Sơ đồ tổ chức giao diện

Giao diện chính là một cửa sổ chứa menu chính và toolbar. Thông qua đó để đi đến các giao diện thành viên để thực hiện các chức năng trong hệ thống.

Theo yêu cầu thì mọi nhân viên phải đăng nhập trước khi sử dụng ứng dụng nên form đăng nhập xuất hiện trước để yêu cầu đăng nhập.

Ngoài ra mỗi ứng dụng trong thời gian khởi động cần có một màn hình chào cùng với thanh tiến trình để người có cảm giác ứng dụng đang khởi động.

### Thiết kế giao diện cho các chức năng nghiệp vụ

Trong phần thiết kế giao diện các bạn nên sử dụng gói hình ảnh được định dạng đúng tên, kí tự để tránh bị lỗi. Các bạn có thể sử dụng nhiều hình ảnh phù hợp với các bạn

Lưu ý quan trọng:

* Các bạn phải đặt tên đúng như hướng dẫn vì các tên này ảnh hưởng đến việc viết mã sau này. Nếu các bạn đặt tên khác thì phải chỉnh mã sau này theo tên mà các bạn đã đặt.
* Trong lập trình giao diện người ta thường qui ước tiếp đầu ngữ của tên các thành phần giao diện như sau

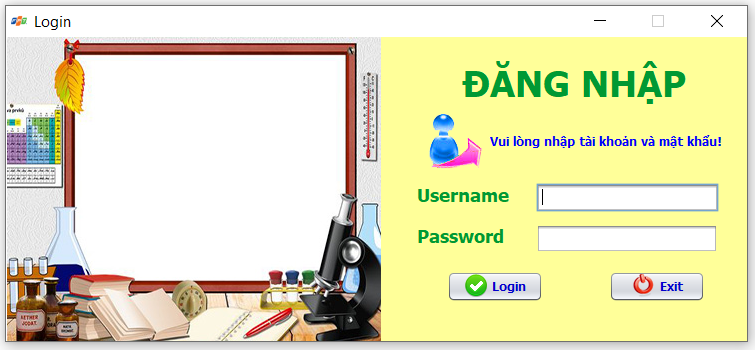
#### Màn hình chào



Các thành phần giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | Chao | JFrame | undecorated: true  Layout:Border  Layout preferredSize: [514, 350] |
| 2 | lblLogo | JLabel | Icon:logosmall.png  Layout.Derection: LEADING |
| 3 | progressBar | JProgressBar | Layout.Derection:  last stringPainted:true |

#### Đăng nhập



Các thành phần giao diện:

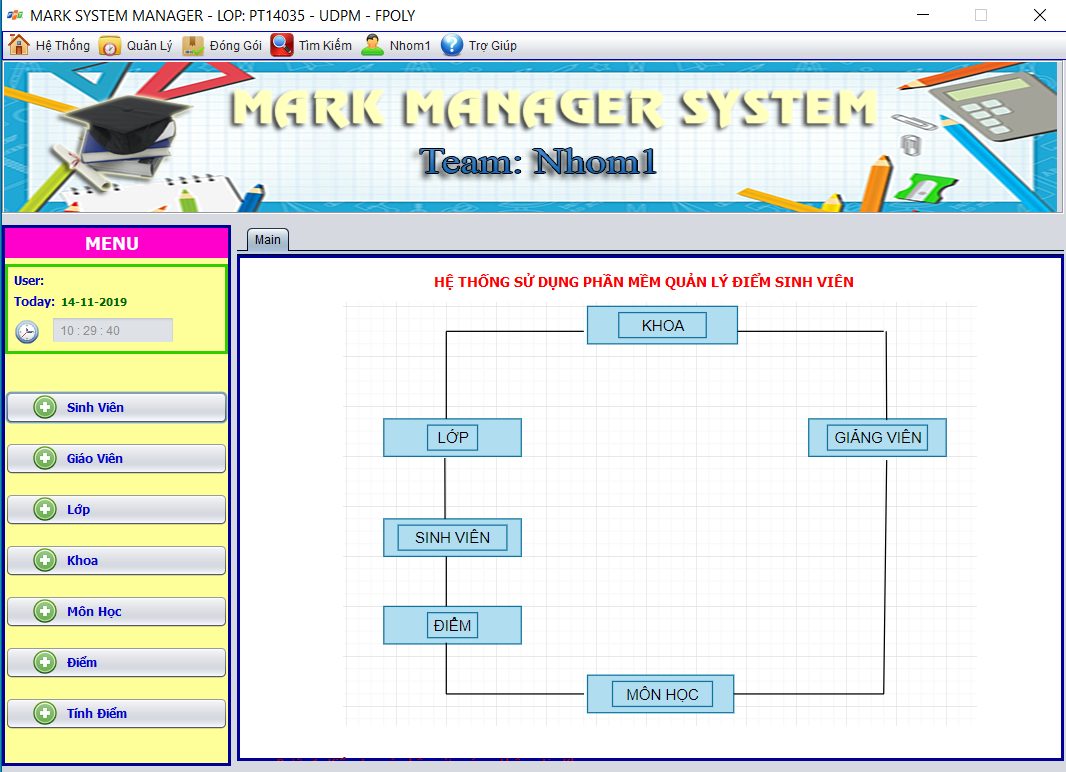
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | Login | JFrame | title:Login  Layout:BorderLayout defaultCloseOperation:DO\_NOTHING |
| 2 | lblTitel | JLabel | Text: ĐĂNG NHẬP  Font:Tohama 36 Bold  Foreground: [0,153,51] |
| 3 | lblTitel2 | JLabel | Text: Vui lòng đăng nhập tài khoản và mật khẩu  Font:Tohama 12 Bold  Foreground: [0,0,255]  Icon: Login\_iconSmall.png |
| 4 | lblImage | JLable | Icon: bialogin.png  Layout.Direction:before |
| 5 | lblusername | JLable | Text: Username  Font: Tahoma 18 Bold  Foreground: [0,153,51] |
| 6 | txtusername | JTextField |  |
| 7 | lblpass | JLable | Text: Password  Font: Tahoma 18 Bold  Foreground: [0,153,51] |
| 8 | txtusername | JPassword |  |
| 9 | btnlogin | JButton | Text:Login  Icon: Login.png |
| 10 | btnexit | JButton | Text:Exit  Icon: Logoff.png |

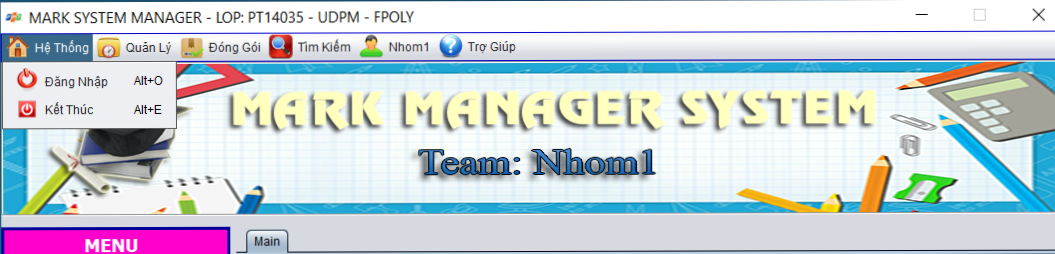
#### Giới thiệu sản phẩm (About)

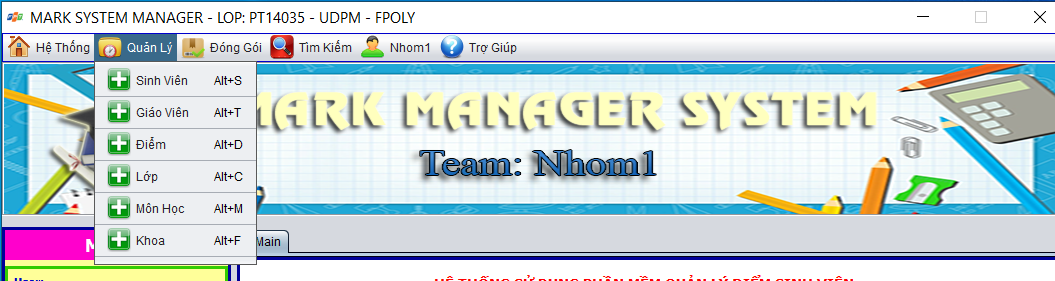


|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | AboutMe | JFrame |  |
| 2 | lblTitel | JLabel | Text: INFORMATION  Font: Tahoma 24 Bold  Foreground: [255,0,0] |
| 3 | lbl1 | JLable | Text: Product Name: Mark System Manager  Font: Tahoma 18 Bold  Foreground: [0,0,255] |
| 4 | lbl2 | JLable | Text: Design By Team: Nhom1  Font: Tahoma 18 Bold  Foreground: [0,0,204] |
| 5 | lbl3 | JLable | Text:ID:PD03196-PD03169-PD03192  Font: Tahoma 18 Bold  Foreground: [0,0,204] |
| 6 | lbl4 | JLable | Text: Class: PT14305  Font: Tahoma 18 Bold  Foreground: [0,0,204] |
| 7 | lblimage | JLable | Icon: about.png |
| 8 | btnok | JButton | Text: OK |

#### Giao diện chính:

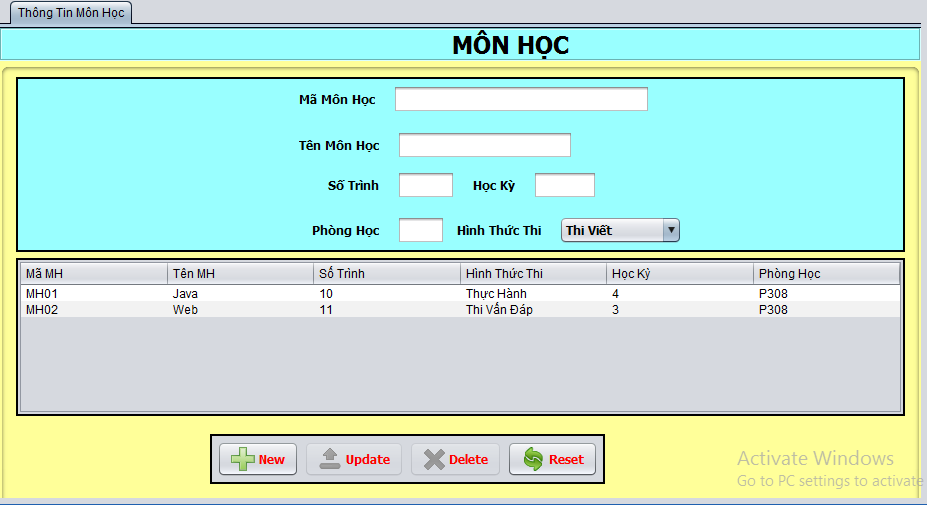






|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | Main | JFrame | Title: MARK SYSTEM MANAGER - LOP: PT14035 - UDPM - FPOLY  Layout:BorderLayout |
| 2 | mnuhethong | JMenu | Text:Hệ thống  Icon: Home.png |
| 3 | mnidangnhap | JMenuItem | Text:Đăng Nhập  Icon: Logoff.png  Accelerator: Alt+O |
| 4 | mniketthuc | JMenuItem | Text: Kết Thúc  Icon: Logout.png  Accelerator: Alt+E |
| 5 | mnuquanly | JMenu | Text:Quản Lý  Icon: Folder.png |
| 6 | mnisinhvien | JMenuItem | Text: Sinh Viên  Icon: Create.png  Accelerator: Alt+S |
| 7 | mnigiaovien | JMenuItem | Text: Giáo Viên  Icon: Create.png  Accelerator: Alt+T |
| 8 | mnidiem | JMenuItem | Text: Điểm  Icon: Create.png  Accelerator: Alt+D |
| 9 | mnilop | JMenuItem | Text: Lớp  Icon: Create.png  Accelerator: Alt+C |
| 10 | mnimonhoc | JMenuItem | Text: Môn Học  Icon: Create.png  Accelerator: Alt+M |
| 11 | mnikhoa | JMenuItem | Text: Khoa  Icon: Create.png  Accelerator: Alt+F |
| 12 | mnudonggoi | JMenu | Text: Đóng Gói  Icon: Package.png |
| 13 | mnutimkiem | JMenu | Text: Tìm Kiếm  Icon: Search-icon.png |
| 14 | mnunhom | JMenu | Text: Nhom1  Icon: changepass.png |
| 15 | mnutrogiup | JMenu | Text: Trợ Giúp  Icon: Help.png |
| 16 | lblbia | JLable | Icon: aaaa.png |
| 17 | pnlmenu | JPanel | Background: [255,0,204] |
| 18 | lblmenu | JLable | Text: MENU |
| 19 | pnlchucnang | JPanel | Background: [255,255,153] |
| 20 | lbluser | JLable | Text: user |
| 21 | lbluser1 | JLable |  |
| 22 | lbltoday | JLable | Text: Today |
| 23 | lbltoday1 | JLable |  |
| 24 | lblclock | JLable | Icon: Clock.png |
| 25 | txtclock | JTextField |  |
| 26 | btnsinhvien | JButton | Text: Sinh Viên  Icon: Create.png  Foreground: [0,0,204] |
| 27 | btngiaovien | JButton | Text: Giáo Viên  Icon: Create.png  Foreground: [0,0,204] |
| 28 | btnlop | JButton | Text: Lớp  Icon: Create.png  Foreground: [0,0,204] |
| 29 | btnkhoa | JButton | Text: Khoa  Icon: Create.png  Foreground: [0,0,204] |
| 30 | btnmonhoc | JButton | Text: Môn Học  Icon: Create.png  Foreground: [0,0,204] |
| 31 | btndiem | JButton | Text: Điểm  Icon: Create.png  Foreground: [0,0,204] |
| 32 | btntinhdiem | JButton | Text: Tính Điểm  Icon: Create.png  Foreground: [0,0,204] |
| 33 | tabmain | JTabbedPane |  |
| 34 | pnlmain | JPanel |  |
| 35 | lblmain | JLabel | Text: DỤNG PHẦN MỀM QUẢN LÝ ĐIỂM SINH VIÊN  Foreground: [255,0,0] |
| 36 | lblimagemain | JLabel | Icon: main\_1.png |

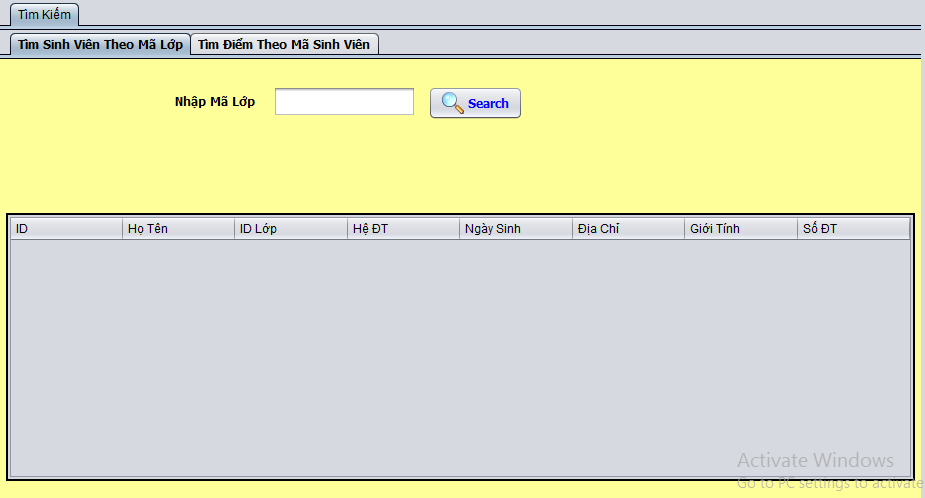
#### Môn Học

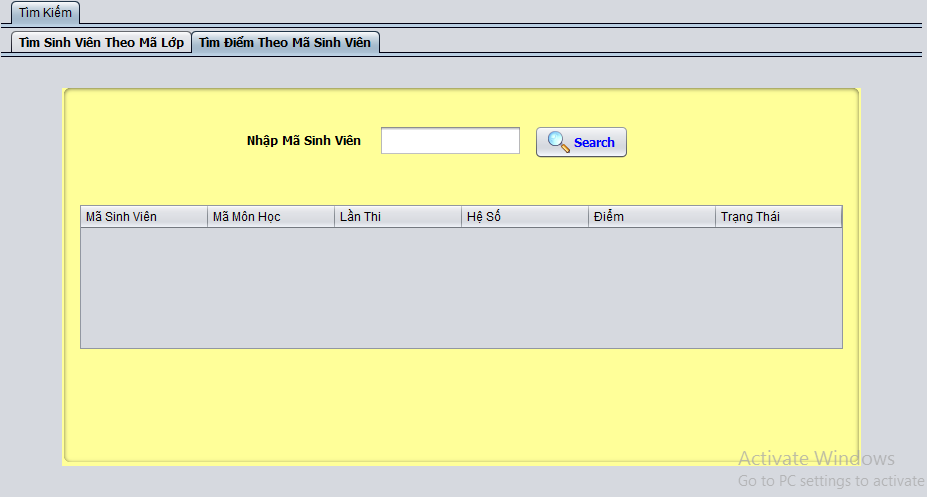


Các thành phần giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | MonHoc | JFrame | Title: MÔN HỌC |
| 2 | lblmamonhoc | JLabel | Text: Mã môn học |
| 3 | lbltenmonhoc | JLabel | Text: Tên môn học |
| 4 | lblsotrinh | JLabel | Text: Số trình |
| 5 | lblhocky | JLabel | Text: Học kỳ |
| 6 | lblphonghoc | JLabel | Text: Phòng học |
| 7 | lblhinhthucthi | JLabel | Text: Hình thức thi |
| 8 | txtmamonhoc | JTextField |  |
| 9 | txttenmonhoc | JTextField |  |
| 10 | txtsotrinh | JTextField |  |
| 11 | txthocky | JTextField |  |
| 12 | txtphongthi | JTextField |  |
| 13 | cbohinhthucthi | JCombobox |  |
| 14 | tblmonhoc | JTable | Model: như hình |
| 15 | btnnew | JButton | Text:New |
| 16 | btnupdate | JButton | Text:Update |
| 17 | btndelete | JButton | Text:Delete |
| 18 | btnreset | JButton | Text:Reset |

#### Tìm Kiếm

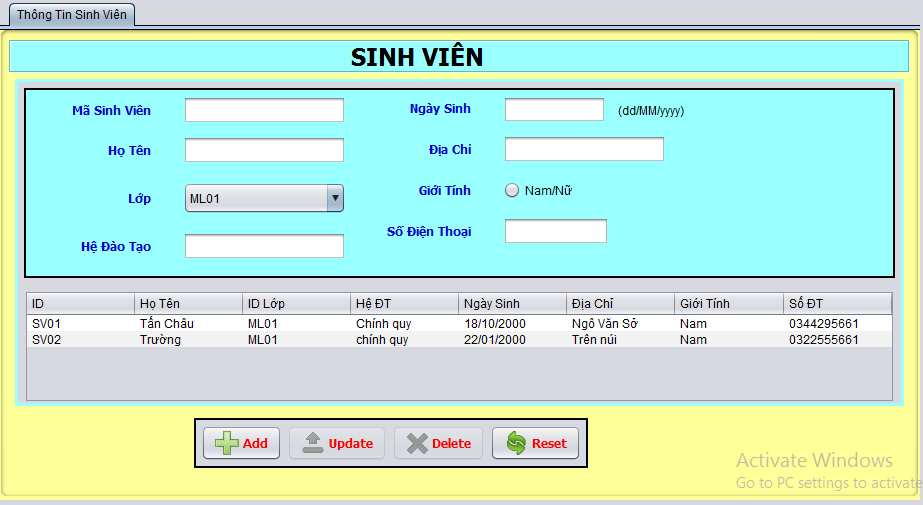




Các thành phần giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | Search | JFrame | Title: Tìm sinh viên theo mã lớp |
| 2 | tabs | JTabbedPane |  |
| 3 | pnltimtheomalop | JPanel | Tab Title: Tìm Sinh Viên Theo Mã Lớp |
| 4 | pnltimtheomasinhvien | JPanel | Tab Title: Tìm Điểm Theo Mã Sinh Viên |
| 5 | lblnhapmalop | JLabel | Text:Nhập mã lớp |
| 6 | txtnhapmalop | JTextField |  |
| 7 | btnsearch | JButton | Text:Search |
| 8 | tbltimsinhvientheomalop | JTable | Model: như hình |
| 9 | lblnhapmasinhvien | JLabel | Text:Nhập mã sinh viên |
| 10 | txtnhapmasinhvien | JTextField |  |
| 11 | btntim | JButton | Text:Search |
| 12 | tbltimdiemtheomasinhvien | JTable | Model: như hình |

#### Sinh Viên



Các thành phần giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | SinhVien | JFrame | Title: Thông tin sinh viên |
| 2 | lblmasinhvien | JLabel | Text:Mã sinh viên |
| 3 | lblhoten | JLabel | Text:Họ Tên |
| 4 | lbllop | JLabel | Text:Lớp |
| 5 | lblhedaotao | JLabel | Text:Hệ đào tạo |
| 6 | lblngaysinh | JLabel | Text:Ngày sinh |
| 7 | lbldiachi | JLabel | Text:Địa chỉ |
| 8 | lblgioitinh | JLabel | Text:Giới tính |
| 9 | lblsdt | JLabel | Text:Số Điện Thoại |
| 10 | txtmasinhvien | JTextField |  |
| 11 | txthoten | JTextField |  |
| 12 | cbolop | JCombobox |  |
| 13 | txthedaotao | JTextField |  |
| 14 | txtngaysinh | JTextField |  |
| 15 | txtdiachi | JTextField |  |
| 16 | rdogioitinh | JRadiobuton | Text:Nam/Nữ |
| 17 | txtsdt | JText Field |  |
| 18 | tblsinhvien | JTable | Model: như hình |
| 19 | btnnew | JButton | Text:New |
| 20 | btnupdate | JButton | Text:Update |
| 21 | btndelete | JButton | Text:Delete |
| 22 | btnreset | JButton | Text:Reset |

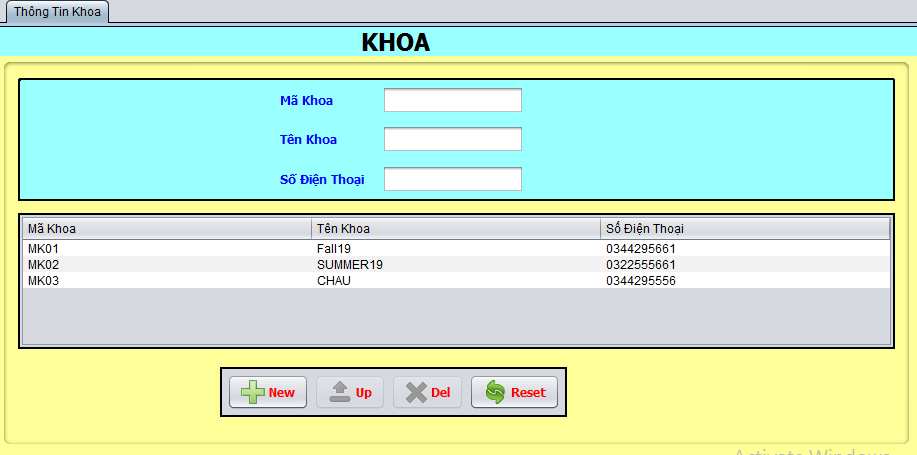
#### Giáo Viên



Các thành phần giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | GiaoVien | JFrame |  |
| 2 | lblTitel | JLabel | Text: THÔNG TIN GIÁO VIÊN |
| 3 | tblgiaovien | JTable |  |
| 4 | lblmaGV | JLable | Text: Mã Giảng Viên |
| 5 | txtmaGV | JTextField |  |
| 6 | lblhoTen | JLable | Text: Họ tên |
| 7 | txthoTen | JTextField |  |
| 8 | lblEmail | JLable | Text: Email |
| 9 | txtEmail | JTextField |  |
| 10 | lblmon | JLable | Text: Môn Giảng Dạy |
| 11 | txtmon | JTextField |  |
| 12 | lblngaysinh | JLable | Text: Ngày Sinh |
| 13 | txtngaysinh | JTextField |  |
| 14 | lbldiachi | JLable | Text: Địa Chỉ |
| 15 | txtdiachi | JTextField |  |
| 16 | lblgioitinh | JLable | Text: Giới Tính |
| 17 | rdosex | JRadiobutton | Text: Nam/Nữ |
| 18 | lblsdt | JLable | Text: Điện Thoại |
| 19 | txtsdt | JTextField |  |
| 20 | btnadd | JBuntton | Text: AddNew |
| 21 | btnupdate | JBuntton | Text: Update |
| 22 | btndelete | JBuntton | Text: Delete |
| 23 | btnreset | JBuntton | Text: Reset |

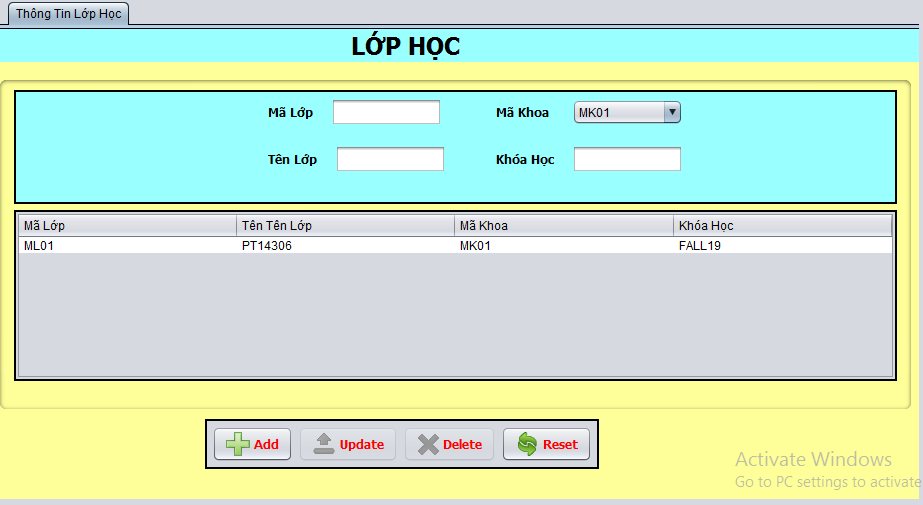
#### Khoa



Các thành phần giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | KHOA | JFame |  |
| 2 | lblTitel | JLable | Text: Thông tin Khóa học |
| 3 | lblmakhoa | JLable | Text: Mã Khoa |
| 4 | txtmakhoa | JTextField |  |
| 5 | lbltenkhoa | JLable | Text: Tên Khoa |
| 6 | txttenkhoa | JTextField |  |
| 7 | lblsdt | JLable | Text: Số điện thoại |
| 8 | txtsdt | JTextfield |  |
| 9 | tblkhoa | JTable |  |
| 10 | btnnew | JButton | Text: New |
| 11 | btnupdate | JButton | Text: Update |
| 12 | btndelete | JButton | Text: Delete |
| 13 | btnreset | JButton | Text: Reset |

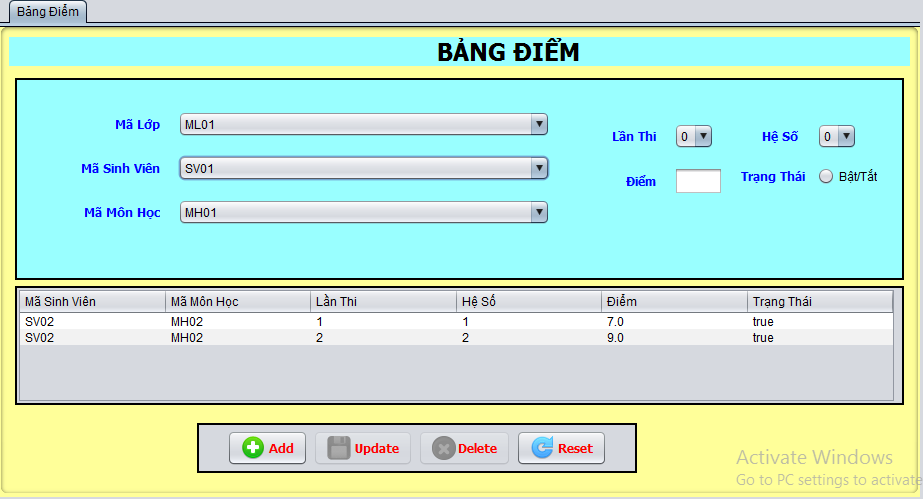
#### Lớp học:



Các thành phần giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | LOPHOC | JFrame |  |
| 2 | lblTitel | JLable | Text: Thông tin Lớp học |
| 3 | tbllophoc | JTable |  |
| 4 | lblmalop | JLable | Text: Mã lớp |
| 5 | txtmalop | JTextField |  |
| 6 | lblmakhoa | JLable | Text: Mã Khoa |
| 7 | cbomakhoa | JCombobox |  |
| 8 | lbltenlop | JLable | Text: Tên Lớp |
| 9 | txttenlop | JTextField |  |
| 10 | lblkhoahoc | JLable | Text: Khóa Học |
| 11 | txtkhoahoc | JTextField |  |
| 12 | btnadd | JButton | Text: Add |
| 13 | btnupdate | JButton | Text: Update |
| 14 | btndelete | JButton | Text: Delete |
| 15 | btnreset | JButton | Text: Reset |

#### Bảng điểm:



Các thành phần giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | frmBangDiem | JFrame | Title: Bảng điểm |
| 2 | lblmalop | Label | Text:Mã lớp |
| 3 | lblmasinhvien | Label | Text:Mã sinh viên |
| 4 | lblmamonhoc | Label | Text:Mã môn học |
| 5 | lbllanthi | Label | Text:Lần thi |
| 6 | lblheso | Label | Text:Hệ số |
| 7 | lbldiem | Label | Text:Điểm |
| 8 | lbltrangthai | Label | Text:Trạng thái |
| 9 | cbomalop | Combo box |  |
| 10 | cbomasinhvien | Combo box |  |
| 11 | cbomamonhoc | Combo box |  |
| 12 | cbolanthi | Combo box |  |
| 13 | cboheso | Combo box |  |
| 14 | txtdiem | Text Field |  |
| 15 | rdotrangthai | Radiobuton | Text:Bật/Tắt |
| 16 | tblangdiem | Table | Model: như hình |
| 17 | btnnew | Button | Text:New |
| 18 | btnupdate | Button | Text:Update |
| 19 | btndelete | Button | Text:Delete |
| 20 | btnreset | Button | Text:Reset |

#### Tính điểm:



Các thành phần giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | frmTinhDiem | JFrame | Title: Bảng điểm |
| 2 | lblmasinhvien | Label | Text:Mã sinh viên |
| 3 | cbomasinhvien | Combo box |  |
| 4 | lblmamon | Label | Text:Mã Môn |
| 5 | txtmamon | Text Field |  |
| 6 | tbltinhdiem | Table | Model: như hình |
| 7 | lbldiemtrungbinh | Label | Text:Điểm Trung Bình |
| 8 | txtdiemtrungbinh | Text Field |  |

# Thực hiện viết mã

## Viết mã tạo CSDL

### Tạo CSDL

CREATE DATABASE QLDSV

GO

USE QLDSV

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE DangNhap(

TenDangNhap nvarchar(50) NOT NULL,

MatKhau nvarchar(20) NOT NULL,

HoTen nvarchar(50) NULL,

Email nvarchar(50) NOT NULL,

CONSTRAINT PK\_DangNhap PRIMARY KEY CLUSTERED (TenDangNhap ASC)

WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE MonHoc(

MaMH nvarchar(10) NOT NULL,

TenMH nvarchar(20) NOT NULL,

SoTrinh int NULL,

HinhThucThi nvarchar(20) NULL,

HocKy int NULL,

PhongHoc nvarchar(10) NULL,

CONSTRAINT PK\_MonHoc PRIMARY KEY CLUSTERED (MaMH ASC)

WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE Khoa(

MaKhoa nvarchar(10) NOT NULL,

TenKhoa nvarchar(50) NOT NULL,

SDT nvarchar(15) NULL,

CONSTRAINT PK\_Khoa PRIMARY KEY CLUSTERED (MaKhoa ASC)

WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE GiangVien(

MaGV nvarchar(10) NOT NULL,

HoTenGV nvarchar(50) NOT NULL,

MaMH nvarchar(10) NOT NULL,

NgaySinh datetime NULL,

GioiTinh bit NULL,

Email nvarchar(50) NULL,

DiaChi nvarchar(70) NULL,

SDT nvarchar(10) NULL,

CONSTRAINT PK\_GiangVien PRIMARY KEY CLUSTERED (MaGV ASC)

WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE GiangVien WITH CHECK ADD CONSTRAINT FK\_GiangVien\_MonHoc FOREIGN KEY(MaMH)

REFERENCES MonHoc (MaMH)

GO

ALTER TABLE GiangVien CHECK CONSTRAINT FK\_GiangVien\_MonHoc

GO

REATE TABLE Lop(

MaLop nvarchar(10) NOT NULL,

TenLop nvarchar(50) NOT NULL,

MaKhoa nvarchar(10) NOT NULL,

KhoaHoc nvarchar(20) NULL,

CONSTRAINT PK\_Lop PRIMARY KEY CLUSTERED (MaLop ASC)

WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE Lop WITH CHECK ADD CONSTRAINT FK\_Lop\_Khoa FOREIGN KEY(MaKhoa)

REFERENCES Khoa (MaKhoa)

GO

ALTER TABLE Lop CHECK CONSTRAINT FK\_Lop\_Khoa

GO

CREATE TABLE SinhVien(

MaSV nvarchar(10) NOT NULL,

HoTen nvarchar(50) NOT NULL,

MaLop nvarchar(10) NOT NULL,

HeDaoTao nvarchar(50) NOT NULL,

NgaySinh datetime NULL,

DiaChi nvarchar(100) NULL,

GioiTinh bit NULL,

SDT nvarchar(15) NULL,

CONSTRAINT PK\_SinhVien PRIMARY KEY CLUSTERED (MaSV ASC)

WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE SinhVien WITH CHECK ADD CONSTRAINT FK\_SinhVien\_Lop FOREIGN KEY(MaLop)

REFERENCES Lop (MaLop)

GO

ALTER TABLE SinhVien CHECK CONSTRAINT FK\_SinhVien\_Lop

GO

CREATE TABLE Diem(

MaSV nvarchar(10) NOT NULL,

MaMH nvarchar(10) NOT NULL,

LanThi tinyint NOT NULL,

HeSo tinyint NULL,

Diem float NULL,

TrangThai bit NULL,

CONSTRAINT PK\_Diem PRIMARY KEY CLUSTERED (MaSV ASC,MaMH ASC,LanThi ASC)

WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

)ON [PRIMARY]

GO

ALTER TABLE Diem WITH CHECK ADD CONSTRAINT FK\_Diem\_MonHoc FOREIGN KEY(MaMH)

REFERENCES MonHoc (MaMH)

GO

ALTER TABLE Diem CHECK CONSTRAINT FK\_Diem\_MonHoc

GO

ALTER TABLE Diem WITH CHECK ADD CONSTRAINT FK\_Diem\_SinhVien FOREIGN KEY(MaSV)

REFERENCES SinhVien (MaSV)

GO

ALTER TABLE Diem CHECK CONSTRAINT FK\_Diem\_SinhVien

GO

### SQL truy vấn và thao tác

#### SQL đối với Bảng 1

#### SQL đối với Bảng 2

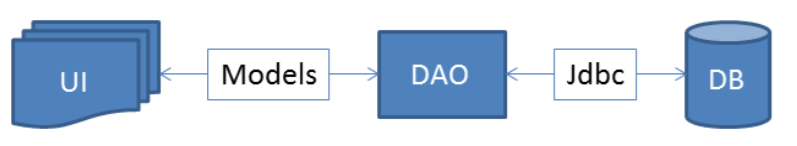
### Các thủ tục lưu tổng hợp thống kê

#### Procedure 1

#### Procedure 2

## Lập trình JDBC

Trong dự án này chúng ta chọn mô hình lập trình với CSDL như hình sau:



Trong đó:

* UI: các thành phần giao diện
* DB: cơ sở dữ liệu
* Jdbc: là lớp tiện ích cung cấp các hàm tiện ích làm việc với CSDL thông qua câu lệnh sql hoặc lời gọi thủ tục lưu
* Model: là các lớp mô tả dữ liệu theo cấu trúc các bảng trong CSDL
* DAO: là các lớp thao tác và truy vấn dữ liệu. Nó có nhiệm vụ chuyển đổi Model sang SQL và ngược lại.

Theo mô hình này thì lập trình giao diện làm việc với CSDL thông qua các lớp model. Model như là một gói dữ liệu được sử dụng để giao tiếp giữa giao diện và các thành phần DAO. DAO mới thực sự là lập trình CSDL nó làm nhiệm vụ nhận model từ giao diện, chuyển đổi sang SQL và sử dụng Jdbc để thực hiện các thao tác dữ liệu. Ngược lại thì truy vấn dữ liệu từ CSDL, chuyển đổi thành model để hiển thị lên các thành phần giao diện.

Việc tách thành các phần riêng biết giúp quản lý dự án tốt hơn, dễ bảo trì nâng cấp hơn

### Lớp hỗ trợ

Các lớp hỗ trợ chứa các phương thức tiện ích giúp chúng ta thực hiện các xử lý trong ứng dụng đơn giản hơn, ngắn gọn hơn, dễ hiểu hơn. Trong phần này chúng ta xây dựng Jdbc như sau:

#### Lớp tiện ích DBConnect:

package JDBC;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

public class DBConnect {

public static String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

public static String url = "jdbc:sqlserver://localhost:1433;databaseName=QLDSV";

public static String user = "PD03196";

public static String pass = "3181999";

public static Connection cnn;

public static boolean open() {

try {

if (cnn == null || cnn.isClosed()) {

Class.forName(driver);

cnn = DriverManager.getConnection(url, user, pass);

}

return true;

} catch (SQLException ex) {

Logger.getLogger(DBConnect.class.getName()).log(Level.SEVERE, null, ex);

} catch (ClassNotFoundException ex) {

Logger.getLogger(DBConnect.class.getName()).log(Level.SEVERE, null, ex);

}

return false;

}

public static void close() {

try {

if (cnn != null) {

cnn.close();

}

} catch (SQLException ex) {

Logger.getLogger(DBConnect.class.getName()).log(Level.SEVERE, null, ex);

}

}

public static void close(PreparedStatement ps){

try {

if (ps !=null) {

ps.close();

}

} catch (SQLException ex) {

Logger.getLogger(DBConnect.class.getName()).log(Level.SEVERE, null, ex);

}

close();

}

public static void close(PreparedStatement ps, ResultSet rs){

try {

if (rs !=null) {

rs.close();

}

} catch (SQLException ex) {

Logger.getLogger(DBConnect.class.getName()).log(Level.SEVERE, null, ex);

}

close(ps);

}

}

} catch (SQLException ex) {

Logger.getLogger(DBConnect.class.getName()).log(Level.SEVERE, null, ex);

}

}

public static void close(PreparedStatement ps){

try {

if (ps !=null) {

ps.close();

}

} catch (SQLException ex) {

Logger.getLogger(DBConnect.class.getName()).log(Level.SEVERE, null, ex);

}

close();

}

public static void close(PreparedStatement ps, ResultSet rs){

try {

if (rs !=null) {

rs.close();

}

} catch (SQLException ex) {

Logger.getLogger(DBConnect.class.getName()).log(Level.SEVERE, null, ex);

}

close(ps);

}

}

#### Lớp tiện ích ShareHelper

package JDBC;

import java.awt.Image;

import java.io.File;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.nio.file.StandardCopyOption;

import javax.swing.ImageIcon;

public class ShareHelper {

public static final Image APP\_ICON;

static {

String file = "/Icon/fpt.png";

APP\_ICON = new ImageIcon(ShareHelper.class.getResource(file)).getImage();

}

public static boolean saveLogo(File file) {

File dir = new File("logos");

if (!dir.exists()) {

dir.mkdirs();

}

File newFile = new File(dir, file.getName());

try {

Path source = Paths.get(file.getAbsolutePath());

Path destination = Paths.get(newFile.getAbsolutePath());

Files.copy(source, destination, StandardCopyOption.REPLACE\_EXISTING);

return true;

} catch (Exception ex) {

return false;

}

}

/\*\*

\* \* Đọc hình ảnh logo chuyên đề \* @param fileName là tên file logo \*

\* @return ảnh đọc được

\*/

public static ImageIcon readLogo(String fileName) {

File path = new File("logos", fileName);

return new ImageIcon(path.getAbsolutePath());

}

### Model class - Các lớp mô tả dữ liệu

Để tránh các sai sót về kiểu dữ liệu khi làm việc với CSDL, chúng ta cần phải xây dựng các lớp mô tả dữ liệu với kiểu phù hợp. Điều này giúp làm việc với CSDL chặt chẽ hơn, dễ hơn.

#### Model GiaoVien

package GiaoVien;

import MonHoc.MonHoc;

import java.util.ArrayList;

import java.util.Date;

public class GiaoVien {

private String magv;

private String hotengv;

private Date ngaysinh;

private boolean gioitinh;

private String email;

private String diachi;

private String sdt;

private String mamh;

private ArrayList<MonHoc> listMh;

public ArrayList<MonHoc> getListMh() {

return listMh;

}

public void setListMh(ArrayList<MonHoc> listMh) {

this.listMh = listMh;

}

public GiaoVien() {

}

public GiaoVien(String magv, String hotengv, Date ngaysinh, boolean gioitinh, String email, String diachi, String sdt, String mamh) {

this.magv = magv;

this.hotengv = hotengv;

this.magv = magv;

this.hotengv = hotengv;

this.ngaysinh = ngaysinh;

this.gioitinh = gioitinh;

this.email = email;

this.diachi = diachi;

this.sdt = sdt;

this.mamh = mamh;

}

public void setMamh(String mamh) {

this.mamh = mamh;

}

public String getMamh() {

return mamh;

}

public String getMagv() {

return magv;

}

public void setMagv(String magv) {

this.magv = magv;

}

public String getHotengv() {

return hotengv;

}

public void setHotengv(String hotengv) {

this.hotengv = hotengv;

}

public Date getNgaysinh() {

return ngaysinh;

}

public void setNgaysinh(Date ngaysinh) {

this.ngaysinh = ngaysinh;

}

public boolean isGioitinh() {

return gioitinh;

}

public void setGioitinh(boolean gioitinh) {

this.gioitinh = gioitinh;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getDiachi() {

return diachi;

}

public void setDiachi(String diachi) {

this.diachi = diachi;

} public String getSdt() {

return sdt;

}

public void setSdt(String sdt) {

this.sdt = sdt;

}

}

}

#### Model Khoa

package Khoa;

public class Khoa {

private String makhoa;

private String tenkhoa;

private String sdt;

public Khoa() {

}

public Khoa(String makhoa, String tenkhoa, String sdt) {

this.makhoa = makhoa;

this.tenkhoa = tenkhoa;

this.sdt = sdt;

}

public String getMakhoa() {

return makhoa;

}

public void setMakhoa(String makhoa) {

this.makhoa = makhoa;

}

public String getTenkhoa() {

return tenkhoa;

}

public void setTenkhoa(String tenkhoa) {

this.tenkhoa = tenkhoa;

}

public String getSdt() {

return sdt;

}

public void setSdt(String sdt) {

this.sdt = sdt;

}

}

#### Model LopHoc

package LopHoc;

public class LopHoc {

private String malop;

private String tenlop;

private String makhoa;

private String khoahoc;

public LopHoc() {

}

public LopHoc(String malop, String tenlop, String makhoa, String khoahoc) {

this.malop = malop;

this.tenlop = tenlop;

this.makhoa = makhoa;

this.khoahoc = khoahoc;

}

public String getMalop() {

return malop;

}

public void setMalop(String malop) {

this.malop = malop;

}

public String getTenlop() {

return tenlop;

}

public void setTenlop(String tenlop) {

this.tenlop = tenlop;

}

public String getMakhoa() {

return makhoa;

}

public void setMakhoa(String makhoa) {

this.makhoa = makhoa;

}

public String getKhoahoc() {

return khoahoc;

}

public void setKhoahoc(String khoahoc) {

this.khoahoc = khoahoc;

}

}

#### Model MonHoc

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package MonHoc;

/\*\*

\*

\* @author Anonymous

\*/

public class MonHoc {

private String mamh;

private int sotrinh;

private String tenmh;

private String hinhthucthi;

private int hocky;

private String phonghoc;

public MonHoc() {

}

public MonHoc(String mamh, int sotrinh, String tenmh, String hinhthucthi, int hocky, String phonghoc) {

this.mamh = mamh;

this.sotrinh = sotrinh;

this.tenmh = tenmh;

this.hinhthucthi = hinhthucthi;

this.hocky = hocky;

this.phonghoc = phonghoc;

}

public int getHocky() {

return hocky;

}

public void setHocky(int hocky) {

this.hocky = hocky;

}

public String getMamh() {

return mamh;

}

public void setMamh(String mamh) {

this.mamh = mamh;

}

public int getSotrinh() {

return sotrinh;

}

public void setSotrinh(int sotrinh) {

this.sotrinh = sotrinh;

}

public String getTenmh() {

return tenmh;

}

public void setTenmh(String tenmh) {

this.tenmh = tenmh;

}

public String getHinhthucthi() {

return hinhthucthi;

}

public void setHinhthucthi(String hinhthucthi) {

this.hinhthucthi = hinhthucthi;

}

public String getPhonghoc() {

return phonghoc;

}

public void setPhonghoc(String phonghoc) {

this.phonghoc = phonghoc;

}

}

this.tenmh = tenmh;

this.hinhthucthi = hinhthucthi;

this.hocky = hocky;

this.phonghoc = phonghoc;

}

public int getHocky() {

return hocky;

}

public void setHocky(int hocky) {

this.hocky = hocky;

}

public String getMamh() {

return mamh;

}

public void setMamh(String mamh) {

this.mamh = mamh;

}

public int getSotrinh() {

return sotrinh;

}

public void setSotrinh(int sotrinh) {

this.sotrinh = sotrinh;

}

public String getTenmh() {

return tenmh;

}

public void setTenmh(String tenmh) {

this.tenmh = tenmh;

}

public String getHinhthucthi() {

return hinhthucthi;

}

public void setHinhthucthi(String hinhthucthi) {

this.hinhthucthi = hinhthucthi;

}

public String getPhonghoc() {

return phonghoc;

}

public void setPhonghoc(String phonghoc) {

this.phonghoc = phonghoc;

}

}

public String getTenmh() {

return tenmh;

}

public void setTenmh(String tenmh) {

this.tenmh = tenmh;

}

public String getHinhthucthi() {

return hinhthucthi;

}

public void setHinhthucthi(String hinhthucthi) {

this.hinhthucthi = hinhthucthi;

}

public String getPhonghoc() {

return phonghoc;

}

public void setPhonghoc(String phonghoc) {

this.phonghoc = phonghoc;

}

}

#### Model SinhVien

package SinhVien;

import java.util.Date;

public class SinhVien {

private String masv;

private String tensv;

private String malop;

private String diachi;

private String hedaotao;

private Date ngaysinh;

private boolean gioitinh;

private String sodt;

public SinhVien() {

}

public SinhVien(String masv, String tensv, String malop, String diachi, String hedaotao, Date ngaysinh, boolean gioitinh, String sodt) {

this.masv = masv;

this.tensv = tensv;

this.malop = malop;

this.diachi = diachi;

this.hedaotao = hedaotao;

this.ngaysinh = ngaysinh;

this.gioitinh = gioitinh;

this.sodt = sodt;

}

public String getMasv() {

return masv;

}

public void setMasv(String masv) {

this.masv = masv;

}

public String getTensv() {

return tensv;

}

public void setTensv(String tensv) {

this.tensv = tensv;

}

public String getMalop() {

return malop;

}

public void setMalop(String malop) {

this.malop = malop;

}

public String getDiachi() {

public String getMasv() {

return masv;

}

public void setMasv(String masv) {

this.masv = masv;

}

public String getTensv() {

return tensv;

}

public void setTensv(String tensv) {

this.tensv = tensv;

}

public String getMalop() {

return malop;

}

public void setMalop(String malop) {

this.malop = malop;

}

public String getDiachi() {

return diachi;

}

public void setDiachi(String diachi) {

this.diachi = diachi;

}

public String getHedaotao() {

return hedaotao;

}

public void setHedaotao(String hedaotao) {

this.hedaotao = hedaotao;

}

public Date getNgaysinh() {

return ngaysinh;

}

public void setNgaysinh(Date ngaysinh) {

this.ngaysinh = ngaysinh;

}

public boolean isGioitinh() {

return gioitinh;

}

public void setGioitinh(boolean gioitinh) {

this.gioitinh = gioitinh;

}

public String getSodt() {

return sodt;

}

public void setSodt(String sodt) {

this.sodt = sodt;

}

}

public String getHedaotao() {

return hedaotao;

}

public void setHedaotao(String hedaotao) {

this.hedaotao = hedaotao;

}

public Date getNgaysinh() {

return ngaysinh;

}

public void setNgaysinh(Date ngaysinh) {

this.ngaysinh = ngaysinh;

}

public boolean isGioitinh() {

return gioitinh;

}

public void setGioitinh(boolean gioitinh) {

this.gioitinh = gioitinh;

}

public String getSodt() {

return sodt;

}

public void setSodt(String sodt) {

this.sodt = sodt;

}

}

#### Model BangDiem

package BangDiem;

public class BangDiem {

private String masv;

private String mamh;

private int lanthi;

private int heso;

private float diem;

private boolean trangthai;

public BangDiem() {

}

public BangDiem(String masv, String mamh, int lanthi, int heso, float diem, boolean trangthai) {

this.masv = masv;

this.mamh = mamh;

this.lanthi = lanthi;

this.heso = heso;

this.diem = diem;

this.trangthai = trangthai;

}

public String getMasv() {

return masv;

}

public void setMasv(String masv) {

this.masv = masv;

}

public String getMamh() {

return mamh;

}

public void setMamh(String mamh) {

this.mamh = mamh;

}

public int getLanthi() {

return lanthi;

}

public void setLanthi(int lanthi) {

this.lanthi = lanthi;

}

public int getHeso() {

return heso;

}

public void setHeso(int heso) {

this.heso = heso;

}

public float getDiem() {

return diem;

}

public void setDiem(float diem) {

this.diem = diem;

}

public boolean isTrangthai() {

return trangthai;

}

public void setTrangthai(boolean trangthai) {

this.trangthai = trangthai;

}

}

public String getMasv() {

return masv;

}

public void setMasv(String masv) {

this.masv = masv;

}

public String getMamh() {

return mamh;

}

public void setMamh(String mamh) {

this.mamh = mamh;

}

public int getLanthi() {

return lanthi;

}

public void setLanthi(int lanthi) {

this.lanthi = lanthi;

}

public int getHeso() {

return heso;

}

public void setHeso(int heso) {

this.heso = heso;

}

public float getDiem() {

return diem;

}

public void setDiem(float diem) {

this.diem = diem;

}

public boolean isTrangthai() {

return trangthai;

}

public void setTrangthai(boolean trangthai) {

this.trangthai = trangthai;

}

}

public float getDiem() {

return diem;

}

public void setDiem(float diem) {

this.diem = diem;

}

public boolean isTrangthai() {

return trangthai;

}

public void setTrangthai(boolean trangthai) {

this.trangthai = trangthai;

}

}

#### Model TimKiem

package Search;

public class TimKiem {

private String malop;

private String masv;

public TimKiem() {}

public TimKiem(String malop, String masv) {

this.malop = malop;

this.masv = masv;

}

public void setMasv(String masv) {

this.masv = masv;

}

public String getMasv() {

return masv;

}

public TimKiem(String malop) {

this.malop = malop;

}

public String getMalop() {

return malop;

}

public void setMalop(String malop) {

this.malop = malop

}

}

### DAO Class - Các lớp truy xuất dữ liệu

#### Lớp DAO làm việc với Bảng Môn Học

package GiaoVien;

import ConnectDatabase.DBConnect;

import MonHoc.MonHoc;

import MonHoc.MonHocDAO;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.Date;

import java.util.logging.Level;

import java.util.logging.Logger;

public class GiaoVienDAO implements IGiaoVienDAO {

@Override

public ArrayList<GiaoVien> getAll() {

// throw new UnsupportedOperationException("Not supported yet.");

ArrayList<GiaoVien> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \*from GiangVien");

rs = ps.executeQuery();

list = new ArrayList<GiaoVien>();

while (rs.next()) {

GiaoVien gv = new GiaoVien();

gv.setMagv(rs.getString(1));

gv.setHotengv(rs.getString(2));

gv.setMamh(rs.getString(3));

gv.setNgaysinh(new Date(rs.getDate(4).getTime()));

gv.setGioitinh(rs.getBoolean(5));

gv.setEmail(rs.getString(6));

gv.setDiachi(rs.getString(7));

gv.setSdt(rs.getString(8));

list.add(gv);

}

} catch (SQLException ex) {

Logger.getLogger(GiaoVienDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<GiaoVien> findByIDMonHoc(String mamh) {

ArrayList<GiaoVien> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \* from GiangVien where MaMH = ?");

ps.setString(1, mamh);

rs = ps.executeQuery();

list = new ArrayList<GiaoVien>();

while (rs.next()) {

GiaoVien gv = new GiaoVien();

gv.setMagv(rs.getString(1));

gv.setHotengv(rs.getString(2));

gv.setMamh(rs.getString(3));

gv.setNgaysinh(new Date(rs.getDate(4).getTime()));

gv.setGioitinh(rs.getBoolean(5));

gv.setEmail(rs.getString(6));

gv.setDiachi(rs.getString(7));

gv.setSdt(rs.getString(8));

list.add(gv);

}

} catch (SQLException ex) {

Logger.getLogger(GiaoVienDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public GiaoVien addNew(GiaoVien gv) {

PreparedStatement ps = null;

ps = DBConnect.cnn.prepareStatement("INSERT INTO GiangVien values (?,?,?,?,?,?,?,?)");

ps.setString(1, gv.getMagv());

ps.setString(2, gv.getHotengv());

ps.setString(3, gv.getMamh());

ps.setDate(4, new java.sql.Date(gv.getNgaysinh().getTime()));

ps.setBoolean(5, gv.isGioitinh());

ps.setString(6, gv.getEmail());

ps.setString(7, gv.getDiachi());

ps.setString(8, gv.getSdt());

int row = ps.executeUpdate();

if (row < 1) {

gv = null;

}

} catch (SQLException ex) {

Logger.getLogger(GiaoVienDAO.class.getName()).log(Level.SEVERE, null, ex);

gv = null;

} finally {

DBConnect.close(ps);

}

}

return gv;

}

@Override

public GiaoVien updateByID(GiaoVien gv) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("update GiangVien set HoTenGV =?,"

+ "MaMH= ?,NgaySinh=?,"

+ "GioiTinh=?, Email = ?, DiaChi = ?, "

+ "SDT = ? where MaGV = ?");

ps.setString(1, gv.getHotengv());

ps.setString(2, gv.getMamh());

//ps.setDate(3, new java.sql.Date(new Date().getTime()));

ps.setDate(3, new java.sql.Date(gv.getNgaysinh().getTime()));

ps.setBoolean(4, gv.isGioitinh());

ps.setString(5, gv.getEmail());

ps.setString(6, gv.getDiachi());

ps.setString(7, gv.getSdt());

ps.setString(8, gv.getMagv());

int row = ps.executeUpdate();

if (row < 1) {

gv = null;

}

} catch (SQLException ex) {

Logger.getLogger(GiaoVienDAO.class.getName()).log(Level.SEVERE, null, ex);

gv = null;

} finally {

DBConnect.close();

}

}

return gv;

}

public void deleteIDGV(String GiaoVienID)throws SQLException,ClassNotFoundException{

PreparedStatement ps = null;

if (DBConnect.open()) {

ps = DBConnect.cnn.prepareStatement("delete from GiangVien where MaGV= ?");

ps.setString(1, GiaoVienID);

ps.executeUpdate();

DBConnect.close();

}

}

public static void main(String[] args) {

System.out.println(new GiaoVienDAO().findByIDMonHoc("GV01").get(0).getMamh());

}

@Override

public ArrayList<GiaoVien> CheckID(String magv) {

ArrayList<GiaoVien> list = null;

PreparedStatement psCheck = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

psCheck = DBConnect.cnn.prepareStatement("select \*from GiangVien where MaGV=?");

psCheck.setString(1, magv);

rs = psCheck.executeQuery();

list = new ArrayList<GiaoVien>();

while (rs.next()) {

GiaoVien giaoVien = new GiaoVien();

giaoVien.setMagv(rs.getString(1));

list.add(giaoVien);

}

} catch (SQLException ex) {

Logger.getLogger(GiaoVienDAO.class.getName()).log(Level.SEVERE, null, ex);

}finally{

DBConnect.close(psCheck, rs);

}

}

return list;

}

}

#### Lớp DAO làm việc với Bảng Khoa

package Khoa;

import ConnectDatabase.DBConnect;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

public class KhoaDAO implements IKhoaDAO {

@Override

public ArrayList<Khoa> getAll() {

ArrayList<Khoa> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \*from Khoa");

rs = ps.executeQuery();

list = new ArrayList<Khoa>();

while (rs.next()) {

Khoa k = new Khoa();

k.setMakhoa(rs.getString(1));

k.setTenkhoa(rs.getString(2));

k.setSdt(rs.getString(3));

list.add(k);

}

} catch (SQLException ex) {

Logger.getLogger(KhoaDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public Khoa addNew(Khoa khoa) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("INSERT INTO Khoa values (?,?,?)");

ps.setString(1, khoa.getMakhoa());

ps.setString(2, khoa.getTenkhoa());

ps.setString(3, khoa.getSdt());

int row = ps.executeUpdate();

if (row < 1) {

khoa = null;

}

} catch (SQLException ex) {

Logger.getLogger(KhoaDAO.class.getName()).log(Level.SEVERE, null, ex);

khoa = null;

}finally{

DBConnect.close(ps);

}

}

return khoa;

}

@Override

public Khoa updateByID(Khoa khoa) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("update Khoa set TenKhoa =?, SDT = ? where MaKhoa = ?");

ps.setString(1, khoa.getTenkhoa());

ps.setString(2, khoa.getSdt());

ps.setString(3, khoa.getMakhoa());

int row = ps.executeUpdate();

if (row < 1) {

khoa = null;

}

} catch (SQLException ex) {

Logger.getLogger(KhoaDAO.class.getName()).log(Level.SEVERE, null, ex);

khoa = null;

} finally {

DBConnect.close(); }

} return khoa;

}

public void deleteKhoa(String KhoaID) throws SQLException, ClassNotFoundException{

PreparedStatement ps = null;

if (DBConnect.open()) {

ps = DBConnect.cnn.prepareStatement("delete from Khoa where MaKhoa = ?");

ps.setString(1, KhoaID);

ps.executeUpdate();

DBConnect.close();

}

}

@Override

public ArrayList<Khoa> checkID(String makhoa) {

ArrayList<Khoa> list = null;

PreparedStatement psCheck = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

psCheck = DBConnect.cnn.prepareStatement("select \* from Khoa where MaKhoa = ?");

psCheck.setString(1, makhoa);

rs = psCheck.executeQuery();

list = new ArrayList<Khoa>();

while (rs.next()) {

Khoa k = new Khoa();

k.setMakhoa(rs.getString(1));

list.add(k);

}

} catch (SQLException ex) {

Logger.getLogger(KhoaDAO.class.getName()).log(Level.SEVERE, null, ex);

}finally{

DBConnect.close(psCheck, rs);

}

}

return list;

}

}

#### Lớp DAO làm việc với Bảng Lớp Học

package LopHoc;

import ConnectDatabase.DBConnect;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

public class LopHocDAO implements ILopHocDAO {

@Override

public ArrayList<LopHoc> getAll() {

ArrayList<LopHoc> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \* from Lop");

rs = ps.executeQuery();

list = new ArrayList<LopHoc>();

while (rs.next()) {

LopHoc lh = new LopHoc();

lh.setMalop(rs.getString(1));

lh.setTenlop(rs.getString(2));

lh.setMakhoa(rs.getString(3));

lh.setKhoahoc(rs.getString(4));

list.add(lh);

}

} catch (SQLException ex) {

Logger.getLogger(LopHocDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<LopHoc> findByIDKhoa(String maKhoa) {

ArrayList<LopHoc> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \* from Lop where MaKhoa = ?");

ps.setString(1, maKhoa);

rs = ps.executeQuery();

list = new ArrayList<LopHoc>();

while (rs.next()) {

LopHoc lh = new LopHoc();

lh.setMalop(rs.getString(1));

lh.setTenlop(rs.getString(2));

lh.setMakhoa(rs.getString(3));

lh.setKhoahoc(rs.getString(4));

list.add(lh);

}

} catch (SQLException ex) {

Logger.getLogger(LopHocDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public LopHoc addNew(LopHoc lh) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("insert into Lop(MaLop,TenLop,MaKhoa,KhoaHoc) values(?,?,?,?)");

ps.setString(1, lh.getMalop());

ps.setString(2, lh.getTenlop());

ps.setString(3, lh.getMakhoa());

ps.setString(4, lh.getKhoahoc());

int row = ps.executeUpdate();

if (row < 1) {

lh = null;

}

} catch (SQLException ex) {

Logger.getLogger(LopHocDAO.class.getName()).log(Level.SEVERE, null, ex);

lh = null;

} finally {

DBConnect.close(ps);

}

}

return lh;

}

@Override

public LopHoc updateByID(LopHoc lh) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("update Lop set TenLop = ?,MaKhoa=?, KhoaHoc = ? where MaLop = ?");

ps.setString(1, lh.getTenlop());

ps.setString(2, lh.getMakhoa());

ps.setString(3, lh.getKhoahoc());

ps.setString(4, lh.getMalop());

int row = ps.executeUpdate();

if (row < 1) {

lh = null;

}

} catch (SQLException ex) {

Logger.getLogger(LopHocDAO.class.getName()).log(Level.SEVERE, null, ex);

lh = null;

} finally {

DBConnect.close();

}

}

return lh;

}

public void deleteLopHoc(String LopID)throws SQLException, ClassNotFoundException{

PreparedStatement ps = null;

if (DBConnect.open()) {

ps = DBConnect.cnn.prepareStatement("delete from Lop where MaLop= ?");

ps.setString(1, LopID);

ps.executeUpdate();

DBConnect.close();

}

}

@Override

public ArrayList<LopHoc> checkID(String malop) {

ArrayList<LopHoc> list = null;

PreparedStatement psCheck = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

psCheck = DBConnect.cnn.prepareStatement("select \* from Lop where MaLop");

psCheck.setString(1, malop);

rs = psCheck.executeQuery();

list = new ArrayList<LopHoc>();

while (rs.next()) {

LopHoc lh = new LopHoc();

lh.setMalop(rs.getString(1));

list.add(lh);

}

} catch (SQLException ex) {

Logger.getLogger(LopHocDAO.class.getName()).log(Level.SEVERE, null, ex);

}finally{

DBConnect.close(psCheck, rs);

}

}

return list;

}

}

#### Lớp DAO làm việc với Bảng Điểm

package BangDiem;

import JDBC.DBConnect;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

public class BangDiemDAO implements IBangDiemDAO {

@Override

public ArrayList<BangDiem> getAll() {

//throw new UnsupportedOperationException("Not supported yet.");

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \*from Diem");

rs = ps.executeQuery();

list = new ArrayList<BangDiem>();

while (rs.next()) {

BangDiem bd = new BangDiem();

bd.setMasv(rs.getString(1));

bd.setMamh(rs.getString(2));

bd.setLanthi(rs.getInt(3));

bd.setHeso(rs.getInt(4));

bd.setDiem(rs.getFloat(5));

bd.setTrangthai(rs.getBoolean(6));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<BangDiem> findByIDSinhVien(String masv) {

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \* from Diem where MaSV = ?");

ps.setString(1, masv);

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

BangDiem bd = new BangDiem();

bd.setMasv(rs.getString(1));

bd.setMamh(rs.getString(2));

bd.setLanthi(rs.getInt(3));

bd.setHeso(rs.getInt(4));

bd.setDiem(rs.getFloat(5));

bd.setTrangthai(rs.getBoolean(6));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<BangDiem> findByIDMonHoc(String mamh) {

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \* from Diem where MaMH = ?");

ps.setString(1, mamh);

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

BangDiem bd = new BangDiem();

}

}

return list;

}

@Override

public ArrayList<BangDiem> findByIDSinhVien(String masv) {

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \* from Diem where MaSV = ?");

ps.setString(1, masv);

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

BangDiem bd = new BangDiem();

bd.setMasv(rs.getString(1));

bd.setMamh(rs.getString(2));

bd.setLanthi(rs.getInt(3));

bd.setHeso(rs.getInt(4));

bd.setDiem(rs.getFloat(5));

bd.setTrangthai(rs.getBoolean(6));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<BangDiem> findByIDMonHoc(String mamh) {

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \* from Diem where MaMH = ?");

ps.setString(1, mamh);

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

BangDiem bd = new BangDiem();

bd.setMasv(rs.getString(1));

bd.setMamh(rs.getString(2));

bd.setLanthi(rs.getInt(3));

bd.setHeso(rs.getInt(4));

bd.setDiem(rs.getFloat(5));

bd.setTrangthai(rs.getBoolean(6));

list.add(bd);

}

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \* from Diem where MaMH = ?");

ps.setString(1, mamh);

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

BangDiem bd = new BangDiem();

bd.setMasv(rs.getString(1));

bd.setMamh(rs.getString(2));

bd.setLanthi(rs.getInt(3));

bd.setHeso(rs.getInt(4));

bd.setDiem(rs.getFloat(5));

bd.setTrangthai(rs.getBoolean(6));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public BangDiem addNew(BangDiem bd) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("insert into Diem (MaSV,MaMH,LanThi,HeSo,Diem,TrangThai) values (?,?,?,?,?,?)");

ps.setString(1, bd.getMasv());

ps.setString(2, bd.getMamh());

ps.setInt(3, bd.getLanthi());

ps.setInt(4, bd.getHeso());

ps.setFloat(5, bd.getDiem());

ps.setBoolean(6, bd.isTrangthai());

int row = ps.executeUpdate();

if (row < 1) {

bd = null;

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps);

}

}

return bd;

bd = null;

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps);

}

}

return bd;

}

@Override

public BangDiem upDate(BangDiem bd) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("update Diem set HeSo =?,Diem = ?,TrangThai=? where MaSV =? and MaMH =? and LanThi=?");

ps.setInt(1, bd.getHeso());

ps.setFloat(2, bd.getDiem());

ps.setBoolean(3, bd.isTrangthai());

ps.setString(4, bd.getMasv());

ps.setString(5, bd.getMamh());

ps.setInt(6, bd.getLanthi());

int row = ps.executeUpdate();

if (row < 1) {

bd = null;

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close();

}

}

return bd;

}

public void delBangDiem(String MaSV, String MaMH, int LanThi) throws SQLException, ClassNotFoundException {

PreparedStatement ps = null;

if (DBConnect.open()) {

ps = DBConnect.cnn.prepareStatement("delete from Diem where MaSV =? and MaMH =? and LanThi =?");

ps.setString(1, MaSV);

ps.setString(2, MaMH);

ps.setInt(3, LanThi);

ps.executeUpdate();

DBConnect.close();

}

}

if (DBConnect.open()) {

ps = DBConnect.cnn.prepareStatement("delete from Diem where MaSV =? and MaMH =? and LanThi =?");

ps.setString(1, MaSV);

ps.setString(2, MaMH);

ps.setInt(3, LanThi);

ps.executeUpdate();

DBConnect.close();

}

}

@Override

public boolean CheckID(String masv, String mamh, int lanthi) {

PreparedStatement psCheck = null;

ResultSet rs = null;

boolean result = true;

if (DBConnect.open()) {

try {

psCheck = DBConnect.cnn.prepareStatement("select \* from Diem where MaSV =? and MaMH =? and LanThi =?");

psCheck.setString(1, masv);

psCheck.setString(2, mamh);

psCheck.setInt(3, lanthi);

rs = psCheck.executeQuery();

result = rs.next();

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(psCheck, rs);

}

}

return result;

}

@Override

public ArrayList<BangDiem> findMaMH(String masv) {

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select DISTINCT mamh from Diem where trangthai is not null and masv=?");

ps.setString(1, masv);

rs = ps.executeQuery();

list = new ArrayList<BangDiem>();

while (rs.next()) {

BangDiem bd = new BangDiem();

bd.setMasv(rs.getString(1));

bd.setMamh(rs.getString("mamh"));

// bd.setLanthi(rs.getInt("fldLanThi"));

// bd.setHeso(rs.getInt("fldHeSo"));

// bd.setDiem(rs.getFloat("fldDiem"));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

ps = DBConnect.cnn.prepareStatement("select DISTINCT mamh from Diem where trangthai is not null and masv=?");

ps.setString(1, masv);

rs = ps.executeQuery();

list = new ArrayList<BangDiem>();

while (rs.next()) {

BangDiem bd = new BangDiem();

bd.setMasv(rs.getString(1));

bd.setMamh(rs.getString("mamh"));

// bd.setLanthi(rs.getInt("fldLanThi"));

// bd.setHeso(rs.getInt("fldHeSo"));

// bd.setDiem(rs.getFloat("fldDiem"));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

public ArrayList<BangDiem> loaddiem(String masv, String maMon) {

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select lanthi, heso, diem from Diem where mamh =? and masv=?");

ps.setString(1, maMon);

ps.setString(2, masv);

rs = ps.executeQuery();

list = new ArrayList<BangDiem>();

while (rs.next()) {

BangDiem bd = new BangDiem();

//bd.setMasv(rs.getString(1));

// bd.setMamh(rs.getString("fldmamh"));

bd.setLanthi(rs.getInt("LanThi"));

bd.setHeso(rs.getInt("HeSo"));

bd.setDiem(rs.getFloat("Diem"));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<BangDiem> findMasv() {

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select DISTINCT masv from Diem where trangthai is not null");

rs = ps.executeQuery();

list = new ArrayList<BangDiem>();

while (rs.next()) {

BangDiem bd = new BangDiem();

bd.setMasv(rs.getString(1));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

public static void main(String[] args) {

System.out.println(new BangDiemDAO().findMaMH("B103104111").get(0).getMasv());

}

}

bd.setHeso(rs.getInt("HeSo"));

bd.setDiem(rs.getFloat("Diem"));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<BangDiem> findMasv() {

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select DISTINCT masv from Diem where trangthai is not null");

rs = ps.executeQuery();

list = new ArrayList<BangDiem>();

while (rs.next()) {

BangDiem bd = new BangDiem();

bd.setMasv(rs.getString(1));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

}

#### Lớp DAO làm việc với bảng MonHoc

package MonHoc;

import JDBC.DBConnect;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

public class MonHocDAO implements IMonHocDAO {

@Override

public ArrayList<MonHoc> getAll() {

ArrayList<MonHoc> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \*from MonHoc");

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

MonHoc mh = new MonHoc();

mh.setMamh(rs.getString(1));

mh.setTenmh(rs.getString(2));

mh.setSotrinh(rs.getInt(3));

mh.setHinhthucthi(rs.getString(4));

mh.setHocky(rs.getInt(5));

mh.setPhonghoc(rs.getString(6));

list.add(mh);

}

} catch (SQLException ex) {

Logger.getLogger(MonHocDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<MonHoc> findIDMaMH(String maMH) {

ArrayList<MonHoc> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \*from MonHoc where MaMH =?");

ps.setString(1, maMH);

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

MonHoc mh = new MonHoc();

mh.setMamh(rs.getString(1));

mh.setTenmh(rs.getString(2));

mh.setSotrinh(rs.getInt(3));

mh.setHinhthucthi(rs.getString(4));

mh.setHocky(rs.getInt(5));

mh.setPhonghoc(rs.getString(6));

list.add(mh);

}

} catch (SQLException ex) {

Logger.getLogger(MonHocDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public MonHoc addNew(MonHoc mh) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("insert into MonHoc(MaMH, TenMH,SoTrinh, HinhThucThi,HocKy,PhongHoc) values (?,?,?,?,?,?)");

ps.setString(1, mh.getMamh());

ps.setString(2, mh.getTenmh());

ps.setInt(3, mh.getSotrinh());

ps.setString(4, mh.getHinhthucthi());

ps.setInt(5, mh.getHocky());

ps.setString(6, mh.getPhonghoc());

int row = ps.executeUpdate();

if (row < 1) {

mh = null;

}

} catch (SQLException ex) {

Logger.getLogger(MonHocDAO.class.getName()).log(Level.SEVERE, null, ex);

mh = null;

} finally {

DBConnect.close(ps);

}

}

return mh;

}

@Override

public MonHoc updateByID(MonHoc mh) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("update MonHoc set TenMH=?,SoTrinh=?, HinhThucThi =?,HocKy =?,PhongHoc =? where MaMH =?");

ps.setString(1, mh.getTenmh());

ps.setInt(2, mh.getSotrinh());

ps.setString(3, mh.getHinhthucthi());

ps.setInt(4, mh.getHocky());

ps.setString(5, mh.getPhonghoc());

ps.setString(6, mh.getMamh());

int row = ps.executeUpdate();

if (row < 1) {

mh = null;

}

} catch (SQLException ex) {

Logger.getLogger(MonHocDAO.class.getName()).log(Level.SEVERE, null, ex);

mh = null;

} finally {

DBConnect.close();

}

}

return mh;

}

public void delMonHoc(String MonHocID) throws SQLException, ClassNotFoundException {

PreparedStatement ps = null;

if (DBConnect.open()) {

ps = DBConnect.cnn.prepareStatement("delete from MonHoc where MaMH= ?");

ps.setString(1, MonHocID);

ps.executeUpdate();

DBConnect.close();

}

}

@Override

public ArrayList<MonHoc> CheckID(String mamh) {

// throw new UnsupportedOperationException("Not supported yet.");

ArrayList<MonHoc> list = null;

PreparedStatement psCheck = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

psCheck = DBConnect.cnn.prepareStatement("select \*from MonHoc where MaMH=?");

psCheck.setString(1, mamh);

rs = psCheck.executeQuery();

list = new ArrayList<MonHoc>();

while (rs.next()) {

MonHoc monHoc = new MonHoc();

monHoc.setMamh(rs.getString(1));

list.add(monHoc);

}

} catch (SQLException ex) {

Logger.getLogger(MonHocDAO.class.getName()).log(Level.SEVERE, null, ex);

}finally{

DBConnect.close(psCheck, rs);

}

}

return list;

}

}

#### Lớp DAO làm việc với bảng SinhVien

package SinhVien;

import JDBC.DBConnect;

import com.sun.corba.se.impl.protocol.giopmsgheaders.Message;

import com.sun.org.apache.xerces.internal.impl.dv.xs.YearDV;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.Date;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class SinhVienDAO implements ISinhVienDAO {

@Override

public ArrayList<SinhVien> getAll() {

ArrayList<SinhVien> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \*from SinhVien");

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

SinhVien sv = new SinhVien();

sv.setMasv(rs.getString(1));

sv.setTensv(rs.getString(2));

sv.setMalop(rs.getString(3));

sv.setHedaotao(rs.getString(4));

sv.setNgaysinh(new Date(rs.getDate(5).getTime()));

sv.setDiachi(rs.getString(6));

sv.setGioitinh(rs.getBoolean(7));

sv.setSodt(rs.getString(8));

list.add(sv);

}

} catch (SQLException ex) {

Logger.getLogger(SinhVienDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<SinhVien> findByIDLop(String maLop) {

ArrayList<SinhVien> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \* from SinhVien where MaLop = ?");

ps.setString(1, maLop);

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

SinhVien sv = new SinhVien();

sv.setMasv(rs.getString(1));

sv.setTensv(rs.getString(2));

sv.setMalop(rs.getString(3));

sv.setHedaotao(rs.getString(4));

sv.setNgaysinh(new Date(rs.getDate(5).getTime()));

sv.setDiachi(rs.getString(6));

sv.setGioitinh(rs.getBoolean(7));

sv.setSodt(rs.getString(8));

list.add(sv);

}

} catch (SQLException ex) {

Logger.getLogger(SinhVienDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public SinhVien addNew(SinhVien sv) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("insert into SinhVien(MaSV,HoTen,MaLop,HeDaoTao,NgaySinh,DiaChi,GioiTinh,SDT) values (?,?,?,?,?,?,?,?)");

ps.setNString(1, sv.getMasv());

ps.setNString(2, sv.getTensv());

ps.setNString(3, sv.getMalop());

ps.setNString(4, sv.getHedaotao());

ps.setDate(5, new java.sql.Date(sv.getNgaysinh().getTime()));

ps.setNString(6, sv.getDiachi());

ps.setBoolean(7, sv.isGioitinh());

ps.setNString(8, sv.getSodt());

int row = ps.executeUpdate();

if (row < 1) {

sv = null;

}

} catch (SQLException ex) {

Logger.getLogger(SinhVienDAO.class.getName()).log(Level.SEVERE, null, ex);

sv = null;

} finally {

DBConnect.close(ps);

}

}

return sv;

}

@Override

public SinhVien updateByID(SinhVien sv) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("update SinhVien set HoTen = ?, MaLop = ?, HeDaoTao = ?, NgaySinh = ?, DiaChi = ?, GioiTinh = ?, SDT = ? where MaSV = ?");

ps.setString(1, sv.getTensv());

ps.setString(2, sv.getMalop());

ps.setString(3, sv.getHedaotao());

ps.setDate(4, new java.sql.Date(sv.getNgaysinh().getTime()));

ps.setString(5, sv.getDiachi());

ps.setBoolean(6, sv.isGioitinh());

ps.setString(7, sv.getSodt());

ps.setString(8, sv.getMasv());

if (ps.executeUpdate() < 1) {

sv = null;

}

} catch (SQLException ex) {

Logger.getLogger(SinhVienDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close();

}

}

return sv;

}

public void deleteIDSinhVien(String SinhVienID) throws SQLException, ClassNotFoundException {

PreparedStatement ps = null;

if (DBConnect.open()) {

ps = DBConnect.cnn.prepareStatement("delete from SinhVien where MaSV= ?");

ps.setString(1, SinhVienID);

ps.executeUpdate();

DBConnect.close();

}

}

public static void main(String[] args) {

System.out.println(new SinhVienDAO().findByIDLop("").get(0).getMalop());

}

@Override

public ArrayList<SinhVien> CheckID(String masv) {

ArrayList<SinhVien> list = null;

PreparedStatement psCheck = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

psCheck = DBConnect.cnn.prepareStatement("select \*from SinhVien where MaSV=?");

psCheck.setString(1, masv);

rs = psCheck.executeQuery();

list = new ArrayList<SinhVien>();

while (rs.next()) {

SinhVien sinhVien = new SinhVien();

sinhVien.setMasv(rs.getString(1));

list.add(sinhVien);

}

} catch (SQLException ex) {

Logger.getLogger(SinhVienDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(psCheck, rs);

}

}

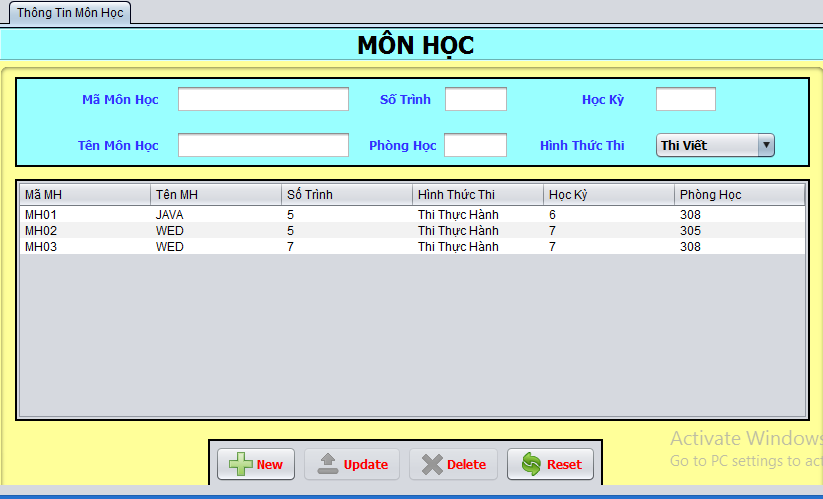
return list;

}

}

## Viết mã cho ứng dụng

### Thông Tin Môn Học (frmMonHoc)



|  |  |
| --- | --- |
| **Sự kiện** | **Công việc phải làm** |
| tblGridView.mouseClicked | jtfMaMH.setText(tblmonhoc.getValueAt(tblmonhoc.getSelectedRow(), 0).toString());  jtfTenMH.setText(tblmonhoc.getValueAt(tblmonhoc.getSelectedRow(), 1).toString());  jtfSoTrinh.setText(tblmonhoc.getValueAt(tblmonhoc.getSelectedRow(), 2).toString());  jcbThi.setSelectedItem(tblmonhoc.getValueAt(tblmonhoc.getSelectedRow(), 3).toString());  jtffhocky.setText(tblmonhoc.getValueAt(tblmonhoc.getSelectedRow(), 4).toString());  jtfPhongHoc.setText(tblmonhoc.getValueAt(tblmonhoc.getSelectedRow(), 5).toString());  btnnew.setEnabled(false);  btnupdate.setEnabled(true);  btndelete.setEnabled(true);  jtfMaMH.setEnabled(false); |
| btnInsert | String maMH = jtfMaMH.getText();  String tenMH = jtfTenMH.getText();  String soTrinh = jtfSoTrinh.getText();  String htThi = (String) jcbThi.getSelectedItem();  String hocKy = jtffhocky.getText();  String phongHoc = jtfPhongHoc.getText();  Check c = new Check();  if (!checkinfo()) {  return;  }  int strinh;  strinh = Integer.parseInt(soTrinh);  int hky;  hky = Integer.parseInt(hocKy);  MonHoc monHoc = new MonHoc(maMH, strinh, tenMH, htThi, hky, phongHoc);  ArrayList<MonHoc> listCheck = new MonHocDAO().CheckID(maMH);  if (listCheck.size() > 0) {  JOptionPane.showMessageDialog(this, "Mã ID bị trùng", "vui lòng kiểm tra lại", JOptionPane.ERROR\_MESSAGE);  jtfMaMH.setText("");  jtfMaMH.requestFocus();  } else {  MonHoc insert = new MonHocDAO().addNew(monHoc);  if (insert != null) {  showAll();  }  } |
| btnUpdate | String maMH = jtfMaMH.getText();  String tenMH = jtfTenMH.getText();  String soTrinh = jtfSoTrinh.getText();  String htThi = (String) jcbThi.getSelectedItem();  String hocKy = jtffhocky.getText();  String phongHoc = jtfPhongHoc.getText();  Check c = new Check();  if (!checkinfo()) {  return;  }  int strinh;  strinh = Integer.parseInt(soTrinh);  int hky;  hky = Integer.parseInt(hocKy);  MonHoc monHoc = new MonHoc(maMH, strinh, tenMH, htThi, hky, phongHoc);  MonHoc update = new MonHocDAO().updateByID(monHoc);  if (update != null) {  showAll();  }  btnnew.setEnabled(true);  btnupdate.setEnabled(false);  btndelete.setEnabled(false);  jtfMaMH.setEnabled(true); |
| btnDelete | String maMh = jtfMaMH.getText();  int b = JOptionPane.showConfirmDialog(null, "Bạn chắc chắn muốn xóa dữ liệu này?", "Thông Báo", JOptionPane.YES\_NO\_OPTION);  if (b == JOptionPane.YES\_OPTION) {  try {  new MonHocDAO().delMonHoc(maMh);  } catch (SQLException ex) {  JOptionPane.showMessageDialog(this, "Dữ liệu chưa thể xóa, có tồn tại điểm của môn này, hãy xóa dữ liệu điểm trước", "Thông báo", JOptionPane.ERROR\_MESSAGE);  } catch (ClassNotFoundException ex) {  JOptionPane.showMessageDialog(this, "Dữ liệu chưa thể xóa, có tồn tại điểm của môn này, hãy xóa dữ liệu điểm trước", "Thông báo", JOptionPane.ERROR\_MESSAGE);  }  while (dtm.getRowCount() > 0) {  dtm.removeRow(0);  }  resetForm();  loaddata();  btnnew.setEnabled(true);  btnupdate.setEnabled(false);  btndelete.setEnabled(false);  jtfMaMH.setEnabled(true);  } |
| btnClear | this.resetForm(); |

**Hướng Dẫn Đặt Mã Nguồn**

private DefaultTableModel dtm;

private DefaultComboBoxModel dcbmthi;

private ArrayList<MonHoc> list;

public frmMonHoc() {

initComponents();

dcbmthi = new DefaultComboBoxModel();

dtm = new DefaultTableModel();

dcbmthi.addElement("Thi Viết");

dcbmthi.addElement("Thi Thực Hành");

dcbmthi.addElement("Thi Vấn Đáp");

jcbThi.setModel(dcbmthi);

dtm.addColumn("Mã MH");

dtm.addColumn("Tên MH");

dtm.addColumn("Số Trình");

dtm.addColumn("Hình Thức Thi");

dtm.addColumn("Học Kỳ");

dtm.addColumn("Phòng Học");

loaddata();

}

private void loaddata() {

try {

list = new MonHocDAO().getAll();

for (MonHoc mh : list) {

dtm.addRow(toObjectsData(mh));

}

tblmonhoc.setModel(dtm);

} catch (InstantiationException ex) {

Logger.getLogger(frmMonHoc.class.getName()).log(Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

Logger.getLogger(frmMonHoc.class.getName()).log(Level.SEVERE, null, ex);

} catch (ClassNotFoundException ex) {

Logger.getLogger(frmMonHoc.class.getName()).log(Level.SEVERE, null, ex);

}

}

public static Object[] toObjectsData(MonHoc hoc) {

Object[] objectsData = {hoc.getMamh(), hoc.getTenmh(), hoc.getSotrinh(), hoc.getHinhthucthi(), hoc.getHocky(), hoc.getPhonghoc()};

return objectsData;

}

IMonHocDAO monHocDAO = (IMonHocDAO) Class.forName("DAO.MonHocDAO").newInstance();

list = new MonHocDAO().getAll();

for (MonHoc mh : list) {

dtm.addRow(toObjectsData(mh));

}

tblmonhoc.setModel(dtm);

} catch (InstantiationException ex) {

Logger.getLogger(frmMonHoc.class.getName()).log(Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

Logger.getLogger(frmMonHoc.class.getName()).log(Level.SEVERE, null, ex);

} catch (ClassNotFoundException ex) {

Logger.getLogger(frmMonHoc.class.getName()).log(Level.SEVERE, null, ex);

}

}

public static Object[] toObjectsData(MonHoc hoc) {

Object[] objectsData = {hoc.getMamh(), hoc.getTenmh(), hoc.getSotrinh(), hoc.getHinhthucthi(), hoc.getHocky(), hoc.getPhonghoc()};

return objectsData;

}

private void showAll() {

while (dtm.getRowCount() > 0) {

dtm.removeRow(0);

}

ArrayList<MonHoc> hocs = new MonHocDAO().getAll();

for (MonHoc mh : hocs) {

Vector v = new Vector();

v.add(mh.getMamh());

v.add(mh.getTenmh());

v.add(mh.getSotrinh());

v.add(mh.getHinhthucthi());

v.add(mh.getHocky());

v.add(mh.getPhonghoc());

dtm.addRow(v);

}

resetForm();

// jtfthi.setText("");

}

public void resetForm() {

jtfMaMH.setText("");

jtfTenMH.setText("");

jtfPhongHoc.setText("");

jtfSoTrinh.setText("");

jtffhocky.setText("");

jtfMaMH.requestFocus();

}

public boolean checkinfo() {

Check c = new Check();

if (!c.checkID(jtfMaMH.getText())) {

JOptionPane.showMessageDialog(this, "Nhập mã sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfMaMH.setText("");

jtfMaMH.requestFocus();

return false;

} else if (!c.checkSpace(jtfTenMH.getText()) || !c.check(jtfTenMH.getText())) {

JOptionPane.showMessageDialog(this, "Nhập tên môn học sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfTenMH.setText("");

jtfTenMH.requestFocus();

return false;

} else if (!c.checkSpace(jtfSoTrinh.getText()) || !c.checkNumber(jtfSoTrinh.getText())) {

JOptionPane.showMessageDialog(this, "Kiểm tra lại số trình", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfSoTrinh.setText("");

jtfSoTrinh.requestFocus();

return false;

} else if (!c.checkSpace(jtffhocky.getText()) || !c.checkNumber(jtffhocky.getText())) {

JOptionPane.showMessageDialog(this, "Nhập sai học kỳ phải là số", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtffhocky.setText("");

jtffhocky.requestFocus();

return false;

} else if (!c.checkSpace(jtfPhongHoc.getText()) || !c.check(jtfPhongHoc.getText())) {

JOptionPane.showMessageDialog(this, "Nhập tên phòng học sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfPhongHoc.setText("");

jtfPhongHoc.requestFocus();

return false;

}

return true;

}

jtffhocky.setText("");

jtfMaMH.requestFocus();

}

public boolean checkinfo() {

Check c = new Check();

if (!c.checkID(jtfMaMH.getText())) {

JOptionPane.showMessageDialog(this, "Nhập mã sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfMaMH.setText("");

jtfMaMH.requestFocus();

return false;

} else if (!c.checkSpace(jtfTenMH.getText()) || !c.check(jtfTenMH.getText())) {

JOptionPane.showMessageDialog(this, "Nhập tên môn học sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfTenMH.setText("");

jtfTenMH.requestFocus();

return false;

} else if (!c.checkSpace(jtfSoTrinh.getText()) || !c.checkNumber(jtfSoTrinh.getText())) {

JOptionPane.showMessageDialog(this, "Kiểm tra lại số trình", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfSoTrinh.setText("");

jtfSoTrinh.requestFocus();

return false;

} else if (!c.checkSpace(jtffhocky.getText()) || !c.checkNumber(jtffhocky.getText())) {

JOptionPane.showMessageDialog(this, "Nhập sai học kỳ phải là số", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtffhocky.setText("");

jtffhocky.requestFocus();

return false;

} else if (!c.checkSpace(jtfPhongHoc.getText()) || !c.check(jtfPhongHoc.getText())) {

JOptionPane.showMessageDialog(this, "Nhập tên phòng học sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfPhongHoc.setText("");

jtfPhongHoc.requestFocus();

return false;

}

return true;

}

jtffhocky.setText("");

jtffhocky.requestFocus();

return false;

} else if (!c.checkSpace(jtfPhongHoc.getText()) || !c.check(jtfPhongHoc.getText())) {

JOptionPane.showMessageDialog(this, "Nhập tên phòng học sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfPhongHoc.setText("");

jtfPhongHoc.requestFocus();

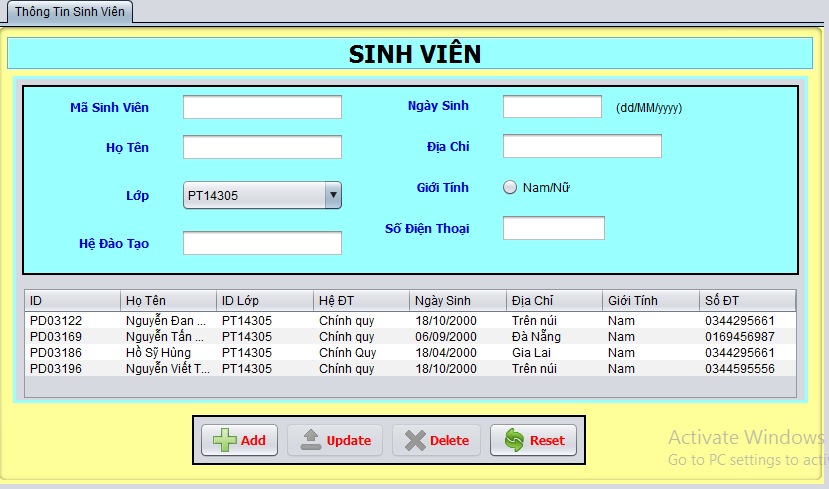
return false;

}

return true;

}

### Thông Tin Sinh Viên (frmSinhVien)



|  |  |
| --- | --- |
| **Sự kiện** | **Công việc phải làm** |
| tblGridView.mouseClicked | jtfMaSV.setText(tbsinhvien.getValueAt(tbsinhvien.getSelectedRow(), 0).toString());  jtfHoTen.setText(tbsinhvien.getValueAt(tbsinhvien.getSelectedRow(), 1).toString());  jcblop.setSelectedItem(tbsinhvien.getValueAt(tbsinhvien.getSelectedRow(), 2).toString());  jtfHeDaoTao.setText(tbsinhvien.getValueAt(tbsinhvien.getSelectedRow(), 3).toString());  jtfngaysinh.setText(tbsinhvien.getValueAt(tbsinhvien.getSelectedRow(), 4).toString());  // jdate.setDateFormatString(jtbTTSV.getValueAt(jtbTTSV.getSelectedRow(), 4).toString());  jtfDiaChi.setText(tbsinhvien.getValueAt(tbsinhvien.getSelectedRow(), 5).toString());  jtfDienThoai.setText(tbsinhvien.getValueAt(tbsinhvien.getSelectedRow(), 7).toString());  String sex = tbsinhvien.getValueAt(tbsinhvien.getSelectedRow(), 6).toString();  if (sex.equals("true")) {  jrbsex.setSelected(true);  } else {  jrbsex.setSelected(false);  }  btnnew.setEnabled(false);  btnupdate.setEnabled(true);  btndelete.setEnabled(true);  jtfMaSV.setEnabled(false); |
| btnInsert | String masv = jtfMaSV.getText();  String tensv = jtfHoTen.getText();  String hedt = jtfHeDaoTao.getText();  String diachi = jtfDiaChi.getText();  String sdt = jtfDienThoai.getText();  String ngaysinh = jtfngaysinh.getText();  boolean gioitinh = jrbsex.isSelected();  String malop = allClass.get(jcblop.getSelectedIndex()).getMalop();  Check c = new Check();  if (!checkinfo()) {  return;  }  Date d = null;  if (ngaysinh == null || ngaysinh.equals("") || !ngaysinh.matches(regexDDMMYYYY)) {  JOptionPane.showMessageDialog(this, "Lỗi ngày sinh", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);  // JOptionPane.showMessageDialog(null, "Lỗi ngày sinh");  jtfngaysinh.setText("");  jtfngaysinh.requestFocus();  return;  } else {  try {  d = new SimpleDateFormat("dd/MM/yyyy").parse(ngaysinh);  } catch (ParseException ex) {  Logger.getLogger(frmSinhVien.class.getName()).log(Level.SEVERE, null, ex);  return;  }  }  SinhVien sinhVien = new SinhVien(masv, tensv, malop, diachi, hedt, d, gioitinh, sdt);  ArrayList<SinhVien> listCheck = new SinhVienDAO().CheckID(masv);  if (listCheck.size() > 0) {  JOptionPane.showMessageDialog(this, "Mã ID bị trùng. Vui lòng kiểm tra lại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  jtfMaSV.setText("");  jtfMaSV.requestFocus();  } else {  SinhVien insert = new SinhVienDAO().addNew(sinhVien);  if (insert != null) {  showAll();  }  } |
| btnUpdate | String masv = jtfMaSV.getText();  String tensv = jtfHoTen.getText();  String hedt = jtfHeDaoTao.getText();  String diachi = jtfDiaChi.getText();  String sdt = jtfDienThoai.getText();  String ngaysinh = jtfngaysinh.getText();  // String ngaysinh = jdate.getDateFormatString();  // String strPattern = "[^a-z ]";  boolean gioitinh = jrbsex.isSelected();  String malop = allClass.get(jcblop.getSelectedIndex()).getMalop();  Check c = new Check();  if (!checkinfo()) {  return;  }  Date d = null;  if (ngaysinh == null || ngaysinh.equals("") || !ngaysinh.matches(regexDDMMYYYY)) {  JOptionPane.showMessageDialog(this, "Lỗi ngày sinh", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);  // JOptionPane.showMessageDialog(null, "Lỗi ngày sinh");  jtfngaysinh.setText("");  jtfngaysinh.requestFocus();  return;  } else {  try {  d = new SimpleDateFormat("dd/MM/yyyy").parse(ngaysinh);  } catch (ParseException ex) {  Logger.getLogger(frmSinhVien.class.getName()).log(Level.SEVERE, null, ex);  return;  }  }  SinhVien sinhVien = new SinhVien(masv, tensv, malop, diachi, hedt, d, gioitinh, sdt);  SinhVien updateByID = new SinhVienDAO().updateByID(sinhVien);  if (updateByID != null) {  showAll();  }  btnnew.setEnabled(true);  btnupdate.setEnabled(false);  btndelete.setEnabled(false);  jtfMaSV.setEnabled(true); |
| btnDelete | int b = JOptionPane.showConfirmDialog(null, "Bạn chắc chắn muốn xóa dữ liệu này?", "Thông Báo", JOptionPane.YES\_NO\_OPTION);  if (b == JOptionPane.YES\_OPTION) {  try {  String masv = jtfMaSV.getText();  new SinhVienDAO().deleteIDSinhVien(masv);  } catch (SQLException ex) {  JOptionPane.showMessageDialog(this, "Sinh viên này có điểm, hãy xóa điểm của sinh viên này trước", "Thông báo", JOptionPane.ERROR\_MESSAGE);  } catch (ClassNotFoundException ex) {  JOptionPane.showMessageDialog(this, "Sinh viên này có điểm, hãy xóa điểm của sinh viên này trước", "Thông báo", JOptionPane.ERROR\_MESSAGE);  }  while (dtm.getRowCount() > 0) {  dtm.removeRow(0);  }  loaddata();  resetForm();  btnnew.setEnabled(true);  btnupdate.setEnabled(false);  btndelete.setEnabled(false);  jtfMaSV.setEnabled(true);  } |
| btnClear | this.resetForm(); |

**Hướng Dẫn Đặt Mã Nguồn**

|  |
| --- |
| private DefaultTableModel dtm;  private DefaultComboBoxModel dcbm;  ArrayList<LopHoc> allClass = null;  ArrayList<SinhVien> list = null;  public static String regexDDMMYYYY = "^(0[1-9]|[12][0-9]|3[01])[- /.](0[1-9]|1[012])[- /.](19|20)\\d\\d$";  public frmSinhVien() {  try {  initComponents();  dcbm = new DefaultComboBoxModel();  dtm = new DefaultTableModel();  ILopHocDAO lopHocDAO = (ILopHocDAO) Class.forName("DAO.LopHocDAO").newInstance();  allClass = lopHocDAO.getAll();  for (LopHoc lh : allClass) {  dcbm.addElement(lh.getMalop());  }  jcblop.setModel(dcbm);  dtm.addColumn("ID");  dtm.addColumn("Họ Tên");  dtm.addColumn("ID Lớp");  dtm.addColumn("Hệ ĐT");  dtm.addColumn("Ngày Sinh");  dtm.addColumn("Địa Chỉ");  dtm.addColumn("Giới Tính");  dtm.addColumn("Số ĐT");  loaddata();  } catch (InstantiationException ex) {  Logger.getLogger(frmSinhVien.class.getName()).log(Level.SEVERE, null, ex);  } catch (IllegalAccessException ex) {  Logger.getLogger(frmSinhVien.class.getName()).log(Level.SEVERE, null, ex);  } catch (ClassNotFoundException ex) {  Logger.getLogger(frmSinhVien.class.getName()).log(Level.SEVERE, null, ex);  }  }  public void resetForm() {  jtfMaSV.setText("");  jtfHoTen.setText("");  jtfDienThoai.setText("");  jtfDiaChi.setText("");  jtfHeDaoTao.setText("");  jtfngaysinh.setText("");  jtfMaSV.requestFocus();  }  private void showAll() {  while (dtm.getRowCount() > 0) {  dtm.removeRow(0);  }  int selectedIndex = jcblop.getSelectedIndex();  LopHoc getLopHoc = allClass.get(selectedIndex);  ArrayList<SinhVien> svs = new SinhVienDAO().findByIDLop(getLopHoc.getMalop());  //ArrayList<SinhVien> sinhViens = new SinhVienDAO().CheckID(g)  for (SinhVien sv : svs) {  Vector vector = new Vector();  vector.add(sv.getMasv());  vector.add(sv.getTensv());  vector.add(sv.getMalop());  vector.add(sv.getHedaotao());  SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");  vector.add(dateFormat.format(sv.getNgaysinh()));  vector.add(sv.getDiachi());  // vector.add(sv.isGioitinh());  if (sv.isGioitinh() == true) {  vector.add("Nam");  } else {  vector.add("Nữ");  }  vector.add(sv.getSodt());  dtm.addRow(vector);  }  resetForm();  }  public boolean checkinfo() {  Check c = new Check();  if (!c.checkID(jtfMaSV.getText())) {  JOptionPane.showMessageDialog(this, "Nhập mã sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);  jtfMaSV.setText("");  jtfMaSV.requestFocus();  return false;  } else if (!c.checkSpace(jtfHoTen.getText()) || !c.check(jtfHoTen.getText())) {  JOptionPane.showMessageDialog(this, "Nhập tên sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);  jtfHoTen.setText("");  jtfHoTen.requestFocus();  return false;  } else if (!c.checkSpace(jtfHeDaoTao.getText()) || !c.check(jtfHeDaoTao.getText())) {  JOptionPane.showMessageDialog(this, "Kiểm tra lại nhập hệ đào tạo", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);  jtfHeDaoTao.setText("");  jtfHeDaoTao.requestFocus();  return false;  // } else if (!c.checkSpace(jtfngaysinh.getText())|| !c.check(jtfHeDaoTao.getText()) || !c.checkDate(jtfngaysinh.getText())) {  // JOptionPane.showMessageDialog(this, "Kiểm tra lại ngày sinh", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);  // jtfngaysinh.setText("");  // jtfngaysinh.requestFocus();  // return false;  } else if (!c.checkSpace(jtfDiaChi.getText())) {  JOptionPane.showMessageDialog(this, "Nhập địa chỉ sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);  jtfDiaChi.setText("");  jtfDiaChi.requestFocus();  return false;  } else if (!c.checkSpace(jtfDienThoai.getText()) || !c.checkPhone(jtfDienThoai.getText())) {  JOptionPane.showMessageDialog(this, "Nhập phone sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);  jtfDienThoai.setText("");  jtfDienThoai.requestFocus();  return false;  }  return true;  } |

### Thông Tin Bảng Điểm (frmBangDiem)



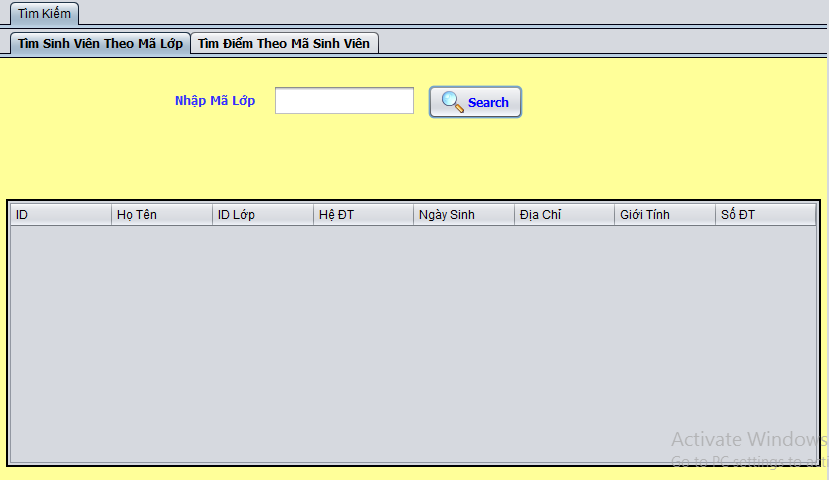
|  |  |
| --- | --- |
| **Sự kiện** | **Công việc phải làm** |
| tblbangdiemMouseReleased | jcbMasv.setSelectedItem(tblbangdiem.getValueAt(tblbangdiem.getSelectedRow(), 0).toString());  jcbMaMH.setSelectedItem(tblbangdiem.getValueAt(tblbangdiem.getSelectedRow(), 1).toString());  jcbLanthi.setSelectedItem(tblbangdiem.getValueAt(tblbangdiem.getSelectedRow(), 2).toString());  jcbHeSo.setSelectedItem(tblbangdiem.getValueAt(tblbangdiem.getSelectedRow(), 3).toString());  jtfDiem.setText(tblbangdiem.getValueAt(tblbangdiem.getSelectedRow(), 4).toString());  String trangthai = tblbangdiem.getValueAt(tblbangdiem.getSelectedRow(), 5).toString();  if (trangthai.equals("true")) {  jrbTrangthai.setSelected(true);  } else {  jrbTrangthai.setSelected(false);  }  btnnew.setEnabled(false);  btnsave.setEnabled(true);  btndelete.setEnabled(true);  jcbMasv.setEnabled(false);  jcbMaMH.setEnabled(false);  jcbLanthi.setEnabled(false);  jcbMaLop.setEnabled(false); |
| btnInsert | if (!checkinfo()) {  return;  }  String diem = jtfDiem.getText();  int lanthi = jcbLanthi.getSelectedIndex();  int heso = jcbHeSo.getSelectedIndex();  boolean trangthai = jrbTrangthai.isSelected();  String masv = listsv.get(jcbMasv.getSelectedIndex()).getMasv();  String mamh = listmh.get(jcbMasv.getSelectedIndex()).getMamh();  float diemsv;  diemsv = Float.parseFloat(diem);  if (diemsv > 10 || diemsv < 0) {  JOptionPane.showMessageDialog(this, "Điểm phải nhập là số từ 0 - 9. Vui lòng nhập lại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  return;  }  if (lanthi == 0) {  JOptionPane.showMessageDialog(this, "Lần Thi phải lớn hơn 0.Vui lòng nhập lại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  return;  }  if (heso == 0) {  JOptionPane.showMessageDialog(this, "Lần Thi phải lớn hơn 0.Vui lòng nhập lại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  return;  }  BangDiem bd = new BangDiem(masv, mamh, lanthi, heso, diemsv, trangthai);  boolean result = new BangDiemDAO().CheckID(masv, mamh, lanthi);  if (result) {  JOptionPane.showMessageDialog(this, "Dữ liệu bị trùng, vui lòng kiểm tra lại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  } else {  BangDiem insert = new BangDiemDAO().addNew(bd);  if (insert != null) {  showAll();  }  } |
| btnUpdate | if (!checkinfo()) {  return;  }  String diem = jtfDiem.getText();  int lanthi = jcbLanthi.getSelectedIndex();  int heso = jcbHeSo.getSelectedIndex();  boolean trangthai = jrbTrangthai.isSelected();  String masv = listsv.get(jcbMasv.getSelectedIndex()).getMasv();  String mamh = listmh.get(jcbMasv.getSelectedIndex()).getMamh();  float diemsv;  diemsv = Float.parseFloat(diem);  if (diemsv > 10 || diemsv < 0) {  JOptionPane.showMessageDialog(this, "Điểm phải nhập là số từ 0 - 9. Vui lòng nhập lại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  jtfDiem.requestFocus();  return;  }  if (lanthi == 0) {  JOptionPane.showMessageDialog(this, "Lần Thi phải lớn hơn 0. Vui lòng nhập lại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  return;  }  if (heso == 0) {  JOptionPane.showMessageDialog(this, "Hệ số phải lớn hơn 0. Vui lòng nhập lại", "Lỗi", JOptionPane.ERROR\_MESSAGE);  return;  }  BangDiem bd = new BangDiem(masv, mamh, lanthi, heso, diemsv, trangthai);  BangDiem updateID = new BangDiemDAO().upDate(bd);  if (updateID != null) {  showAll();  }  btnnew.setEnabled(true);  btnsave.setEnabled(false);  btndelete.setEnabled(false);  jcbMasv.setEnabled(true);  jcbMaMH.setEnabled(true);  jcbLanthi.setEnabled(true);  jcbMaLop.setEnabled(true); |
| btnDelete | int b = JOptionPane.showConfirmDialog(null, "Bạn chắc chắn muốn xóa dữ liệu này?", "Thông Báo", JOptionPane.YES\_NO\_OPTION);  if (b == JOptionPane.YES\_OPTION) {  String masv = listsv.get(jcbMasv.getSelectedIndex()).getMasv();  String mamh = listmh.get(jcbMasv.getSelectedIndex()).getMamh();  int lanthi = jcbLanthi.getSelectedIndex();  try {  new BangDiemDAO().delBangDiem(masv, mamh, lanthi);  } catch (SQLException ex) {  Logger.getLogger(frmBangDiem.class.getName()).log(Level.SEVERE, null, ex);  } catch (ClassNotFoundException ex) {  Logger.getLogger(frmBangDiem.class.getName()).log(Level.SEVERE, null, ex);  }  while (dtm.getRowCount() > 0) {  dtm.removeRow(0);  }  resetForm();  loadData();  btnnew.setEnabled(true);  btnsave.setEnabled(false);  btndelete.setEnabled(false);  jcbMasv.setEnabled(true);  jcbMaMH.setEnabled(true);  jcbLanthi.setEnabled(true);  jcbMaLop.setEnabled(true);  } |
| btnClear | resetForm(); |

**Hướng Dẫn Đặt Mã Nguồn**

|  |
| --- |
| private DefaultTableModel dtm;  private DefaultComboBoxModel dcbmHS;  private DefaultComboBoxModel dcbmLanThi;  private DefaultButtonModel dbm;  private DefaultComboBoxModel dcbmMaSV;  private DefaultComboBoxModel dcbmMaMH;  private DefaultComboBoxModel dcbmMaLop;  ArrayList<SinhVien> listsv = null;  ArrayList<MonHoc> listmh = null;  ArrayList<BangDiem> listbdiem = null;  ArrayList<LopHoc> listML = null;  public frmBangDiem() {  try {  initComponents();  dtm = new DefaultTableModel();  dcbmHS = new DefaultComboBoxModel();  dcbmLanThi = new DefaultComboBoxModel();  dcbmMaSV = new DefaultComboBoxModel();  dcbmMaMH = new DefaultComboBoxModel();  dcbmMaLop = new DefaultComboBoxModel();  dbm = new DefaultButtonModel();  //Load Mon Hoc  IMonHocDAO monHocDAO = (IMonHocDAO) Class.forName("DAO.MonHocDAO").newInstance();  listmh = monHocDAO.getAll();  for (MonHoc mh : listmh) {  dcbmMaMH.addElement(mh.getMamh());  }  jcbMaMH.setModel(dcbmMaMH);  //Load Ma Lop  ILopHocDAO lopHocDAO = (ILopHocDAO) Class.forName("DAO.LopHocDAO").newInstance();  listML = lopHocDAO.getAll();  for (LopHoc lopHoc : listML) {  dcbmMaLop.addElement(lopHoc.getMalop());  }  jcbMaLop.setModel(dcbmMaLop);  //Load Ma sv  try {  ISinhVienDAO sinhVienDAO = (ISinhVienDAO) Class.forName("DAO.SinhVienDAO").newInstance();  listsv = sinhVienDAO.getAll();  dcbmMaSV.removeAllElements();  for (SinhVien sv : listsv) {  dcbmMaSV.addElement(sv.getMasv());  }  jcbMasv.setModel(dcbmMaSV);  } catch (ClassNotFoundException ex) {  Logger.getLogger(frmBangDiem.class.getName()).log(Level.SEVERE, null, ex);  } catch (InstantiationException ex) {  Logger.getLogger(frmBangDiem.class.getName()).log(Level.SEVERE, null, ex);  } catch (IllegalAccessException ex) {  Logger.getLogger(frmBangDiem.class.getName()).log(Level.SEVERE, null, ex);  }  //Load Lan Thi  dcbmLanThi.addElement("0");  dcbmLanThi.addElement("1");  dcbmLanThi.addElement("2");  dcbmLanThi.addElement("3");  dcbmLanThi.addElement("4");  dcbmLanThi.addElement("5");  dcbmLanThi.addElement("6");  jcbLanthi.setModel(dcbmLanThi);  //Load He So  dcbmHS.addElement("0");  dcbmHS.addElement("1");  dcbmHS.addElement("2");  dcbmHS.addElement("3");  dcbmHS.addElement("4");  jcbHeSo.setModel(dcbmHS);  //Ten Bang  dtm.addColumn("Mã Sinh Viên");  dtm.addColumn("Mã Môn Học");  dtm.addColumn("Lần Thi");  dtm.addColumn("Hệ Số");  dtm.addColumn("Điểm");  dtm.addColumn("Trạng Thái");  loadData();  } catch (ClassNotFoundException ex) {  Logger.getLogger(frmBangDiem.class.getName()).log(Level.SEVERE, null, ex);  } catch (InstantiationException ex) {  Logger.getLogger(frmBangDiem.class.getName()).log(Level.SEVERE, null, ex);  } catch (IllegalAccessException ex) {  Logger.getLogger(frmBangDiem.class.getName()).log(Level.SEVERE, null, ex);  }  }  public void loadData() {  try {  IBangDiemDAO bangDiemDAO = (IBangDiemDAO) Class.forName("DAO.BangDiemDAO").newInstance();  listbdiem = bangDiemDAO.getAll();  for (BangDiem bd : listbdiem) {  dtm.addRow(toObjectData(bd));  }  tblbangdiem.setModel(dtm);  } catch (ClassNotFoundException ex) {  Logger.getLogger(frmBangDiem.class.getName()).log(Level.SEVERE, null, ex);  } catch (InstantiationException ex) {  Logger.getLogger(frmBangDiem.class.getName()).log(Level.SEVERE, null, ex);  } catch (IllegalAccessException ex) {  Logger.getLogger(frmBangDiem.class.getName()).log(Level.SEVERE, null, ex);  }  }  public static Object[] toObjectData(BangDiem bd) {  Object[] objects = {bd.getMasv(), bd.getMamh(), bd.getLanthi(), bd.getHeso(), bd.getDiem(), bd.isTrangthai()};  return objects;  }  public void showAll() {  while (dtm.getRowCount() > 0) {  dtm.removeRow(0);  }  ArrayList<BangDiem> bangDiems = new BangDiemDAO().getAll();  for (BangDiem bd : bangDiems) {  Vector v = new Vector();  v.add(bd.getMasv());  v.add(bd.getMamh());  v.add(bd.getLanthi());  v.add(bd.getHeso());  v.add(bd.getDiem());  if (bd.isTrangthai() == true) {  v.add("Bật");  } else {  v.add("Tắt");  }  dtm.addRow(v);  }  resetForm();  }  public void resetForm() {  jtfDiem.setText("");  // jcbHeSo.setSelectedItem(null);  // jcbLanthi.setSelectedItem(null);  // jcbMaMH.setSelectedItem(null);  // jcbMaLop.setSelectedItem(null);  // jcbMasv.setSelectedItem(null);  // jrbTrangthai.setSelected(true);  }  public boolean checkinfo() {  Check c = new Check();  if (!c.checkSpace(jtfDiem.getText())) {  JOptionPane.showMessageDialog(this, "Nhập điểm sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);  jtfDiem.setText("");  jtfDiem.requestFocus();  return false;  }  return true;  } |

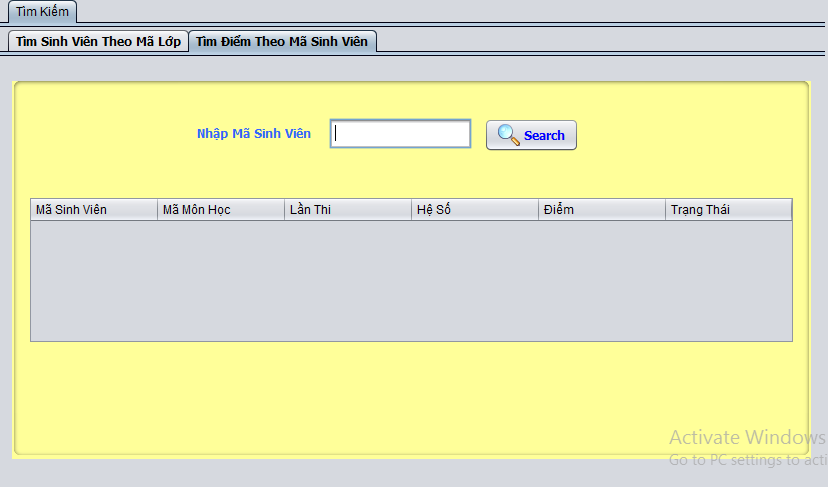
### Tìm Kiếm (frmTimKiem)

**A.Tìm theo mã lớp**



|  |  |
| --- | --- |
| **Sự Kiện** | **Công việc phải làm** |
| btnsearch | timkiem(); |

**B.Tìm theo mã lớp**



|  |  |
| --- | --- |
| **Sự Kiện** | **Công việc phải làm** |
| btntim | timKiemDiem(); |

**Hướng Dẫn Đặt Mã Nguồn**

|  |
| --- |
| private DefaultTableModel dtm;  private DefaultTableModel dtmMark;  ArrayList<BangDiem> listbdiem = null;  /\*\*  \* Creates new form Search  \*/  public Search() {  initComponents();  dtm = new DefaultTableModel();  dtmMark = new DefaultTableModel();  //TIm kiem sinh vien theo ma lop  dtm.addColumn("ID");  dtm.addColumn("Họ Tên");  dtm.addColumn("ID Lớp");  dtm.addColumn("Hệ ĐT");  dtm.addColumn("Ngày Sinh");  dtm.addColumn("Địa Chỉ");  dtm.addColumn("Giới Tính");  dtm.addColumn("Số ĐT");  tbltimsinhvientheomalop.setModel(dtm);  //TIm Kiem diem theo ma sv  dtmMark.addColumn("Mã Sinh Viên");  dtmMark.addColumn("Mã Môn Học");  dtmMark.addColumn("Lần Thi");  dtmMark.addColumn("Hệ Số");  dtmMark.addColumn("Điểm");  dtmMark.addColumn("Trạng Thái");  tbltimdiemtheomasinhvien.setModel(dtmMark);  }  private static Object[] toObjectData(SinhVien sv) {  SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");  String date = dateFormat.format(sv.getNgaysinh());  boolean flag = sv.isGioitinh();  String gioitinh = "Nam";  if (!flag) {  gioitinh = "Nữ";  }  Object[] objectData = {sv.getMasv(), sv.getTensv(), sv.getMalop(),  sv.getHedaotao(), date, sv.getDiachi(), gioitinh, sv.getSodt()};  return objectData;  }  public void timkiem() {  while (dtm.getRowCount() > 0) {  dtm.removeRow(0);  }  String malop = txtmalop.getText();  try {  try {  ISinhVienDAO sinhVienDAO = (ISinhVienDAO) Class.forName("DAO.SinhVienDAO").newInstance();  ArrayList<SinhVien> list = sinhVienDAO.findByIDLop(malop);  for (SinhVien sv : list) {  dtm.addRow(toObjectData(sv));  }  } catch (InstantiationException ex) {  Logger.getLogger(Search.class.getName()).log(Level.SEVERE, null, ex);  } catch (IllegalAccessException ex) {  Logger.getLogger(Search.class.getName()).log(Level.SEVERE, null, ex);  }  } catch (ClassNotFoundException ex) {  Logger.getLogger(Search.class.getName()).log(Level.SEVERE, null, ex);  }  }  public void timKiemDiem() {  while (dtmMark.getRowCount() > 0) {  dtmMark.removeRow(0);  }  try {  String masv = txtmasv.getText();  IBangDiemDAO bangDiemDAO = (IBangDiemDAO) Class.forName("DAO.BangDiemDAO").newInstance();  ArrayList<BangDiem> listbd = bangDiemDAO.findByIDSinhVien(masv);  for (BangDiem bd : listbd) {  dtmMark.addRow(toOjectBd(bd));  }  } catch (ClassNotFoundException ex) {  Logger.getLogger(Search.class.getName()).log(Level.SEVERE, null, ex);  } catch (InstantiationException ex) {  Logger.getLogger(Search.class.getName()).log(Level.SEVERE, null, ex);  } catch (IllegalAccessException ex) {  Logger.getLogger(Search.class.getName()).log(Level.SEVERE, null, ex);  }  }  public static Object[] toOjectBd(BangDiem bd) {  Object[] objects = {bd.getMasv(), bd.getMamh(), bd.getLanthi(), bd.getHeso(), bd.getDiem(), bd.isTrangthai()};  return objects;  } |

### Login (frmLogin)



|  |  |
| --- | --- |
| **Sự Kiện** | **Công việc phải làm** |
| btnlogin | send(); |
| btncancle | System.exit(0); |

**Hướng Dẫn Đặt Mã Nguồn**

public static String nameLogin = "";

PreparedStatement ps = null;

ResultSet rs = null;

public Login() {

setIconImage(ShareHelper.APP\_ICON);

initComponents();

}

public void resetForm() {

txtUsername.setText("");

txtPassword.setText("");

txtUsername.requestFocus();

}

public void send() {

String strnull = "";

String user = txtUsername.getText();

String pass = txtPassword.getText();

if (user.equals(strnull) || pass.equals(strnull)) {

JOptionPane.showMessageDialog(null, "Username hoặc Password không được để trống!");

this.resetForm();

} else if (DBConnect.open()) {

try {

boolean blean = true;

ps = DBConnect.cnn.prepareStatement("select TenDangNhap,MatKhau from DangNhap");

rs = ps.executeQuery();

while (rs.next()) {

String username = "" + rs.getString("TenDangNhap");

String password = "" + rs.getString("MatKhau");

if (user.equals(username) && pass.equals(password)) {

nameLogin = username;

new Main().setVisible(true);

dispose();

} else {

//blean = false;

//this.setVisible(false);

JOptionPane.showMessageDialog(this, "Username hoặc Password bị sai! \nKhông thể đăng nhập. Vui lòng đăng nhập lại!", "Login Fails", JOptionPane.ERROR\_MESSAGE);

this.resetForm();

}

}

}

}

catch (SQLException ex) {

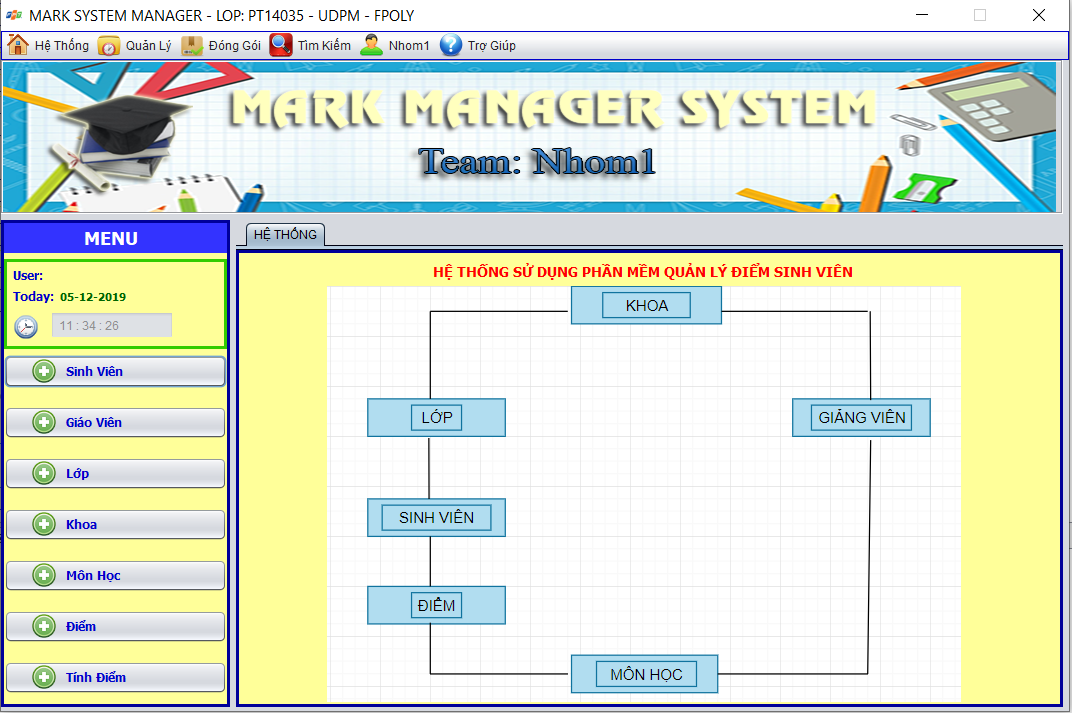
Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

### Main (frmMain)



|  |  |
| --- | --- |
| **Sự Kiện** | **Công việc phải làm** |
| btnInputSinhvien | frmSinhVien sinhVien = new frmSinhVien();  jtpContent.removeAll();  jtpContent.add("Thông Tin Sinh Viên", sinhVien); |
| btnInputGiaovien | frmGiaoVien giaoVien = new frmGiaoVien();  jtpContent.removeAll();  jtpContent.add("Thông Tin Giáo Viên", giaoVien); |
| btnLop | frmLopHoc lophoc = new frmLopHoc();  jtpContent.removeAll();  jtpContent.add("Thông Tin Lớp Học", lophoc); |
| btnKhoa | frmKhoa khoa = new frmKhoa();  jtpContent.removeAll();  jtpContent.add("Thông Tin Khoa", khoa); |
| btnMonhoc | frmMonHoc monhoc = new frmMonHoc();  jtpContent.removeAll();  jtpContent.add("Thông Tin Môn Học", monhoc); |
| btnDiem | frmBangDiem bangDiem = new frmBangDiem();  jtpContent.removeAll();  jtpContent.add("Bảng Điểm", bangDiem); |
| btnTinhdiem | frmTinhDiem tinhDiem = new frmTinhDiem();  jtpContent.removeAll();  jtpContent.add("Tính Điểm",tinhDiem); |

**Hướng Dẫn Đặt Mã Nguồn**

public static String nameLogin;

private Date today = new Date();

Locale local = new Locale("vi", "VI");

DateFormat d = DateFormat.getDateInstance(DateFormat.MEDIUM, local);

String date = d.format(today);

public Main() {

setIconImage(ShareHelper.APP\_ICON);

initComponents();

setUser();

Timer dongho = new Timer(1000, new ActionListener() {

public void actionPerformed(ActionEvent e) {

Calendar lich = Calendar.getInstance();

int gio = lich.get(Calendar.HOUR);

int phut = lich.get(Calendar.MINUTE);

int giay = lich.get(Calendar.SECOND);

jtfCurrent.setText(" " + gio + " : " + phut + " : " + giay);

}

});

dongho.start();

}

private void setUser() {

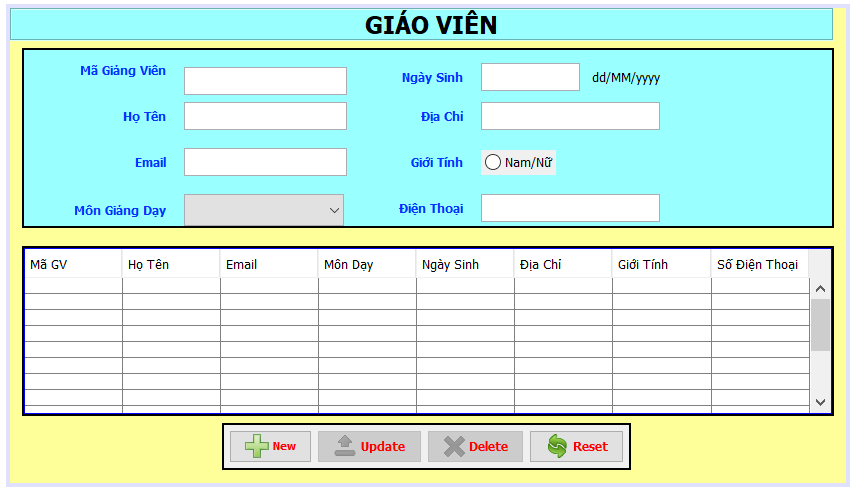
nameLogin = Login.nameLogin;

jlbUser.setText(nameLogin);

jlbToday.setText(date);

}

### Giáo Viên (frmGiaovien)



|  |  |
| --- | --- |
| **Sự Kiện** | **Công việc phải làm** |
| btnnew | String magv = jtfMaGV.getText();  String hoten = jtfHoTen.getText();  String email = jtfEmail.getText();  String sdt = jtfDienThoai.getText();  String ngaysinh = jtfngaysinh.getText();  boolean gioitinh = jrbsex.isSelected();  String diachi = jtfDiaChi.getText();  String mamh = listmh.get(jcbgiangday.getSelectedIndex()).getMamh();  Check c = new Check();  if (!checkinfo()) {  return;  }  Date d = null;  if (ngaysinh == null || ngaysinh.equals("") || !ngaysinh.matches(regexDDMMYYYY)) {  JOptionPane.showMessageDialog(this, "Nhập ngày sinh sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);  jtfngaysinh.setText("");  jtfngaysinh.requestFocus();  return;  } else {  try {  d = new SimpleDateFormat("dd/MM/yyyy").parse(ngaysinh);  } catch (ParseException ex) {  JOptionPane.showMessageDialog(this, "Nhập ngày sinh sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);  return;  }  }  GiaoVien giaoVien = new GiaoVien(magv, hoten, d, gioitinh, email, diachi, sdt, mamh);  ArrayList<GiaoVien> listCheck = new GiaoVienDAO().CheckID(magv);  if (listCheck.size() > 0) {  JOptionPane.showMessageDialog(this, "Mã ID bị trùng", "vui lòng kiểm tra lại", JOptionPane.ERROR\_MESSAGE);  jtfMaGV.setText("");  jtfMaGV.requestFocus();  } else {  GiaoVien insert = new GiaoVienDAO().addNew(giaoVien);  if (insert != null) {  showAll();  }  } |
| btnupdate | String magv = jtfMaGV.getText();  String hoten = jtfHoTen.getText();  String email = jtfEmail.getText();  String sdt = jtfDienThoai.getText();  String ngaysinh = jtfngaysinh.getText();  boolean gioitinh = jrbsex.isSelected();  String diachi = jtfDiaChi.getText();  String mamh = listmh.get(jcbgiangday.getSelectedIndex()).getMamh();  Date d = null;  if (ngaysinh == null || ngaysinh.equals("") || !ngaysinh.matches(regexDDMMYYYY)) {  JOptionPane.showMessageDialog(this, "Nhập ngày sinh sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);  jtfngaysinh.setText("");  jtfngaysinh.requestFocus();  return;  } else {  try {  d = new SimpleDateFormat("dd/MM/yyyy").parse(ngaysinh);  } catch (ParseException ex) {  JOptionPane.showMessageDialog(this, "Nhập ngày sinh sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);  return;  }  }  GiaoVien giaoVien = new GiaoVien(magv, hoten, d, gioitinh, email, diachi, sdt, mamh);  GiaoVien update = new GiaoVienDAO().updateByID(giaoVien);  if (update != null) {  showAll();  }  btnnew.setEnabled(true);  btnupdate.setEnabled(false);  btndelete.setEnabled(false);  jtfMaGV.setEnabled(true); |
| btndelete | int b = JOptionPane.showConfirmDialog(null, "Bạn chắc chắn muốn xóa dữ liệu này?", "Thông Báo", JOptionPane.YES\_NO\_OPTION);  if (b == JOptionPane.YES\_OPTION) {  String magv = jtfMaGV.getText();  try {  new GiaoVienDAO().deleteIDGV(magv);  } catch (SQLException ex) {  Logger.getLogger(frmGiaoVien.class.getName()).log(Level.SEVERE, null, ex);  } catch (ClassNotFoundException ex) {  Logger.getLogger(frmGiaoVien.class.getName()).log(Level.SEVERE, null, ex);  }  while (dtm.getRowCount() > 0) {  dtm.removeRow(0);  }  resetForm();  data();  btnnew.setEnabled(true);  btnupdate.setEnabled(false);  btndelete.setEnabled(false);  jtfMaGV.setEnabled(true);  } |
| btnreset | this.resetForm(); |

**Hướng Dẫn Đặt Mã Nguồn**

private DefaultTableModel dtm;

private DefaultComboBoxModel dcbm;

ArrayList<MonHoc> listmh = null;

ArrayList<GiaoVien> listgv = null;

Public static String regexDDMMYYYY = "^(0[1-9]|[12][0-9]|3[01])[- /.](0[1-9]|1[012])[- /.](19|20)\\d\\d$";

public frmGiaoVien() {

try {

initComponents();

dcbm = new DefaultComboBoxModel();

dtm = new DefaultTableModel();

IMonHocDAO monHocDAO = (IMonHocDAO) Class.forName("DAO.MonHocDAO").newInstance();

listmh = monHocDAO.getAll();

for (MonHoc mh : listmh) {

dcbm.addElement(mh.getMamh());

}

jcbgiangday.setModel(dcbm);

dtm.addColumn("ID Giáo Viên");

dtm.addColumn("Họ Tên");

dtm.addColumn("Môn Dạy");

dtm.addColumn("Ngày Sinh");

dtm.addColumn("Giới Tính");

dtm.addColumn("Email");

dtm.addColumn("Địa Chỉ");

dtm.addColumn("Số ĐT");

data();

// loaddata();

} catch (ClassNotFoundException ex) {

Logger.getLogger(frmGiaoVien.class.getName()).log(Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

Logger.getLogger(frmGiaoVien.class.getName()).log(Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

Logger.getLogger(frmGiaoVien.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void data() {

try {

IGiaoVienDAO giaoVienDAO = (IGiaoVienDAO) Class.forName("DAO.GiaoVienDAO").newInstance();

listgv = new GiaoVienDAO().getAll();

for (GiaoVien gv : listgv) {

dtm.addRow(toObjectsData(gv));

}

tblgiaovien.setModel(dtm);

} catch (ClassNotFoundException ex) {

Logger.getLogger(frmGiaoVien.class.getName()).log(Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

Logger.getLogger(frmGiaoVien.class.getName()).log(Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

Logger.getLogger(frmGiaoVien.class.getName()).log(Level.SEVERE, null, ex);

}

}

private static Object[] toObjectsData(GiaoVien gv) {

SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");

String date = dateFormat.format(gv.getNgaysinh());

boolean flag = gv.isGioitinh();

String gioitinh = "Nam";

if (!flag) {

gioitinh = "Nữ";

}

Object[] objectsData = {gv.getMagv(), gv.getHotengv(), gv.getMamh(), date, gioitinh, gv.getEmail(), gv.getDiachi(), gv.getSdt()};

return objectsData;

}

public boolean checkinfo() {

Check c = new Check();

if (!c.checkID(jtfMaGV.getText())) {

JOptionPane.showMessageDialog(this, "Nhập mã sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfMaGV.setText("");

jtfMaGV.requestFocus();

return false;

} else if (!c.checkSpace(jtfHoTen.getText()) || !c.check(jtfHoTen.getText())) {

JOptionPane.showMessageDialog(this, "Nhập tên sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfHoTen.setText("");

jtfHoTen.requestFocus();

return false;

} else if (!c.checkSpace(jtfEmail.getText()) || !c.check(jtfEmail.getText())) {

JOptionPane.showMessageDialog(this, "Email nhập sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfEmail.setText("");

jtfEmail.requestFocus();

return false;

} else if (!c.checkSpace(jtfDiaChi.getText())) {

JOptionPane.showMessageDialog(this, "Nhập địa chỉ sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfDiaChi.setText("");

jtfDiaChi.requestFocus();

return false;

} else if (!c.checkSpace(jtfDienThoai.getText()) || !c.checkPhone(jtfDienThoai.getText())) {

JOptionPane.showMessageDialog(this, "Nhập phone sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfDienThoai.setText("");

jtfDienThoai.requestFocus();

return false;

}

return true;

}

public void resetForm() {

jtfMaGV.setText("");

jtfHoTen.setText("");

jtfEmail.setText("");

jtfDiaChi.setText("");

jtfDienThoai.setText("");

jtfngaysinh.setText("");

jtfMaGV.requestFocus();

}

private void showAll() {

while (dtm.getRowCount() > 0) {

dtm.removeRow(0);

}

int selectedIndex = jcbgiangday.getSelectedIndex();

MonHoc getmh = listmh.get(selectedIndex);

ArrayList<GiaoVien> list = new GiaoVienDAO().findByIDMonHoc(getmh.getMamh());

for (GiaoVien gv : list) {

Vector vector = new Vector();

vector.add(gv.getMagv());

vector.add(gv.getHotengv());

vector.add(gv.getMamh());

SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");

vector.add(dateFormat.format(gv.getNgaysinh()));

if (gv.isGioitinh() == true) {

vector.add("Nam");

} else {

vector.add("Nữ");

}

vector.add(gv.getEmail());

vector.add(gv.getDiachi());

vector.add(gv.getSdt());

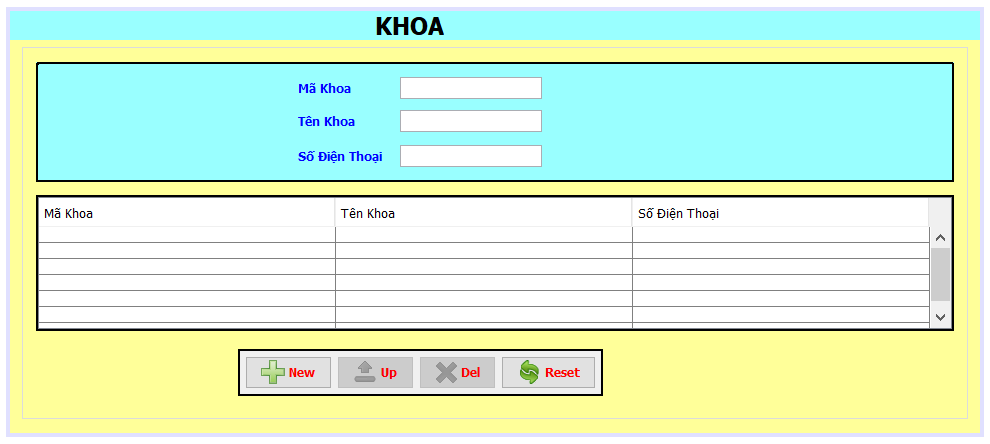
dtm.addRow(vector);

}

resetForm();

}

### Thông tin Khoa (frmKhoa)



|  |  |
| --- | --- |
| **Sự Kiện** | **Công việc phải làm** |
| btnreset | this.ResetForm(); |
| btndelete | int b = JOptionPane.showConfirmDialog(null, "Bạn chắc chắn muốn xóa dữ liệu này?", "Thông Báo", JOptionPane.YES\_NO\_OPTION);  if (b == JOptionPane.YES\_OPTION) {  try {  String makhoa = jtfMaKhoa.getText();  new KhoaDAO().deleteKhoa(makhoa);  } catch (SQLException ex) {  Logger.getLogger(frmKhoa.class.getName()).log(Level.SEVERE, null, ex);  } catch (ClassNotFoundException ex) {  Logger.getLogger(frmKhoa.class.getName()).log(Level.SEVERE, null, ex);  }  while (dtm.getRowCount() > 0) {  dtm.removeRow(0);  }  ResetForm();  loaddata();  btnnew.setEnabled(true);  btnupdate.setEnabled(false);  btndelete.setEnabled(false);  jtfMaKhoa.setEnabled(true);  } |
| btnupdate | if (!checkinfo()) {  return;  }  String makhoa = jtfMaKhoa.getText();  String tenkhoa = jtfTenKhoa.getText();  String sdt = jtfSDT.getText();  Khoa k = new Khoa(makhoa, tenkhoa, sdt);  Khoa update = new KhoaDAO().updateByID(k);  if (update != null) {  showAll();  }  btnnew.setEnabled(true);  btnupdate.setEnabled(false);  btndelete.setEnabled(false);  jtfMaKhoa.setEnabled(true); |
| btnew | if (!checkinfo()) {  return;  }  String makhoa = jtfMaKhoa.getText();  String tenkhoa = jtfTenKhoa.getText();  String sdt = jtfSDT.getText();  Khoa k = new Khoa(makhoa, tenkhoa, sdt);  ArrayList<Khoa> listCheck = new KhoaDAO().checkID(makhoa);  if (listCheck.size() > 0) {  JOptionPane.showMessageDialog(this, "Mã ID bị trùng", "vui lòng kiểm tra lại", JOptionPane.ERROR\_MESSAGE);  jtfMaKhoa.setText("");  jtfMaKhoa.requestFocus();  } else {  Khoa insert = new KhoaDAO().addNew(k);  if (insert != null) {  showAll();  }  } |
| btnkhoaMouseCicked | jtfMaKhoa.setText(btnkhoa.getValueAt(btnkhoa.getSelectedRow(), 0).toString());  jtfTenKhoa.setText(btnkhoa.getValueAt(btnkhoa.getSelectedRow(), 1).toString());  jtfSDT.setText(btnkhoa.getValueAt(btnkhoa.getSelectedRow(), 2).toString());  btnnew.setEnabled(false);  btnupdate.setEnabled(true);  btndelete.setEnabled(true);  jtfMaKhoa.setEnabled(false); |

**Hướng Dẫn Đặt Mã Nguồn**

private DefaultTableModel dtm;

ArrayList<Khoa> list = null;

public frmKhoa() {

initComponents();

dtm = new DefaultTableModel();

dtm.addColumn("Mã Khoa");

dtm.addColumn("Tên Khoa");

dtm.addColumn("Số Điện Thoại");

loaddata();

}

private void loaddata() {

try {

IKhoaDAO khoaDAO = (IKhoaDAO) Class.forName("DAO.KhoaDAO").newInstance();

list = new KhoaDAO().getAll();

for (Khoa khoa : list) {

dtm.addRow(toObjectsData(khoa));

}

btnkhoa.setModel(dtm);

} catch (ClassNotFoundException ex) {

Logger.getLogger(frmKhoa.class.getName()).log(Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

Logger.getLogger(frmKhoa.class.getName()).log(Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

Logger.getLogger(frmKhoa.class.getName()).log(Level.SEVERE, null, ex);

}

}

private static Object[] toObjectsData(Khoa k) {

Object[] objectsData = {k.getMakhoa(), k.getTenkhoa(), k.getSdt()};

return objectsData;

}

public void ResetForm() {

jtfMaKhoa.setText("");

jtfTenKhoa.setText("");

jtfSDT.setText("");

jtfMaKhoa.requestFocus();

} private void showAll() {

while (dtm.getRowCount() > 0) {

dtm.removeRow(0);

}

ArrayList<Khoa> khoas = new KhoaDAO().getAll();

for (Khoa khoa : khoas) {

Vector v = new Vector();

v.add(khoa.getMakhoa());

v.add(khoa.getTenkhoa());

v.add(khoa.getSdt());

dtm.addRow(v);

}

ResetForm();

}

public boolean checkinfo() {

Check c = new Check();

if (!c.checkID(jtfMaKhoa.getText())) {

JOptionPane.showMessageDialog(this, "Nhập Mã khoa sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfMaKhoa.setText("");

jtfMaKhoa.requestFocus();

return false;

} else if (!c.checkSpace(jtfTenKhoa.getText()) || !c.check(jtfTenKhoa.getText())) {

JOptionPane.showMessageDialog(this, "Nhập tên Khoa sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfTenKhoa.setText("");

jtfTenKhoa.requestFocus();

return false;

} else if (!c.checkSpace(jtfSDT.getText()) || !c.checkPhone(jtfSDT.getText())) {

JOptionPane.showMessageDialog(this, "Lỗi Số Điện Thoại", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfSDT.setText("");

jtfSDT.requestFocus();

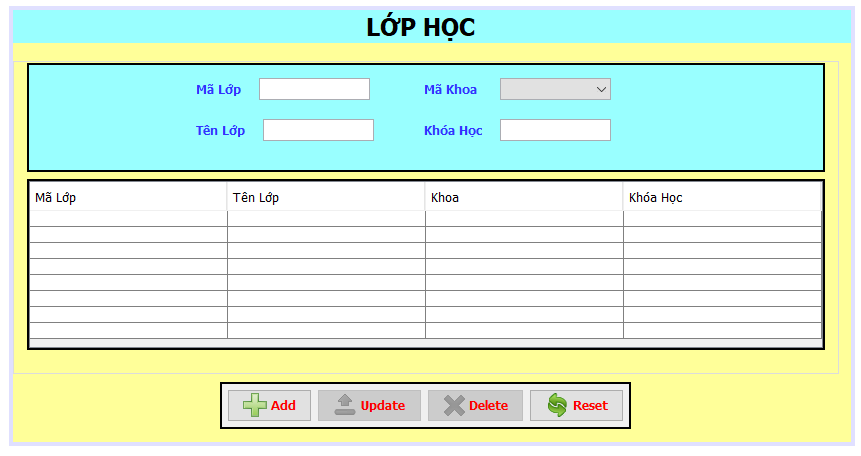
return false;

}

return true;

}

### Thông tin Lớp học (frmLophoc)



|  |  |
| --- | --- |
| **Sự Kiện** | **Công việc phải làm** |
| btnreset | this.ResetForm(); |
| btnnew | String malop = jtfMaLop.getText();  String tenlop = jtfTenLop.getText();  String khoahoc = jtfKhoaHoc.getText();  String makhoa = allKhoa.get(jcbMaKhoa.getSelectedIndex()).getMakhoa();  if (!checkinfo()) {  return;  }  LopHoc lh = new LopHoc(malop, tenlop, makhoa, khoahoc);  ArrayList<LopHoc> listCheck = new LopHocDAO().checkID(malop);  if (listCheck.size() > 0) {  JOptionPane.showMessageDialog(this, "Mã ID bị trùng", "vui lòng kiểm tra lại", JOptionPane.ERROR\_MESSAGE);  jtfMaLop.setText("");  jtfMaLop.requestFocus();  } else {  LopHoc insert = new LopHocDAO().addNew(lh);  if (insert != null) {  showAll();  }  } |
| btndelete | int b = JOptionPane.showConfirmDialog(null, "Bạn chắc chắn muốn xóa dữ liệu này?", "Thông Báo", JOptionPane.YES\_NO\_OPTION);  if (b == JOptionPane.YES\_OPTION) {  try {  String malop = jtfMaLop.getText();  new LopHocDAO().deleteLopHoc(malop);  } catch (SQLException ex) {  Logger.getLogger(frmLopHoc.class.getName()).log(Level.SEVERE, null, ex);  } catch (ClassNotFoundException ex) {  Logger.getLogger(frmLopHoc.class.getName()).log(Level.SEVERE, null, ex);  }  while (dtm.getRowCount() > 0) {  dtm.removeRow(0);  }  loaddata();  ResetForm();  btnnew.setEnabled(true);  btnupdate.setEnabled(false);  btndelete.setEnabled(false);  jtfMaLop.setEnabled(true);  } |
| btnupdate | if (!checkinfo()) {  return;  }  String malop = jtfMaLop.getText();  String tenlop = jtfTenLop.getText();  String khoahoc = jtfKhoaHoc.getText();  String makhoa = allKhoa.get(jcbMaKhoa.getSelectedIndex()).getMakhoa();  LopHoc lh = new LopHoc(malop, tenlop, makhoa, khoahoc);  LopHoc update = new LopHocDAO().updateByID(lh);  if (update != null) {  showAll();  }  btnnew.setEnabled(true);  btnupdate.setEnabled(false);  btndelete.setEnabled(false);  jtfMaLop.setEnabled(true); |
| tbllophocMouseClicked | jtfMaLop.setText(tbllophoc.getValueAt(tbllophoc.getSelectedRow(), 0).toString());  jtfTenLop.setText(tbllophoc.getValueAt(tbllophoc.getSelectedRow(), 1).toString());  jcbMaKhoa.setSelectedItem(tbllophoc.getValueAt(tbllophoc.getSelectedRow(), 2).toString());  jtfKhoaHoc.setText(tbllophoc.getValueAt(tbllophoc.getSelectedRow(), 3).toString());  btnnew.setEnabled(false);  btnupdate.setEnabled(true);  btndelete.setEnabled(true);  jtfMaLop.setEnabled(false); |

**Hướng Dẫn Đặt Mã Nguồn**

private DefaultComboBoxModel dcbm;

private DefaultTableModel dtm;

ArrayList<Khoa> allKhoa = null;

ArrayList<LopHoc> allLop = null;

public frmLopHoc() {

try {

initComponents();

dcbm = new DefaultComboBoxModel();

dtm = new DefaultTableModel();

IKhoaDAO khoaDAO = (IKhoaDAO) Class.forName("DAO.KhoaDAO").newInstance();

allKhoa = khoaDAO.getAll();

for (Khoa k : allKhoa) {

dcbm.addElement(k.getMakhoa());}

jcbMaKhoa.setModel(dcbm);

dtm.addColumn("Mã Lớp");

dtm.addColumn("Tên Tên Lớp");

dtm.addColumn("Mã Khoa");

dtm.addColumn("Khóa Học");

loaddata();

} catch (InstantiationException ex) {

Logger.getLogger(frmLopHoc.class.getName()).log(Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

Logger.getLogger(frmLopHoc.class.getName()).log(Level.SEVERE, null, ex);

} catch (ClassNotFoundException ex) {

Logger.getLogger(frmLopHoc.class.getName()).log(Level.SEVERE, null, ex);

}

}

public void ResetForm() {

jtfMaLop.setText("");

jtfTenLop.setText("");

jtfKhoaHoc.setText("");

jtfMaLop.requestFocus();

}

private void loaddata() {

try {

try {

ILopHocDAO lopHocDAO = (ILopHocDAO) Class.forName("DAO.LopHocDAO").newInstance();

allLop = new LopHocDAO().getAll();

for (LopHoc lh : allLop) {

dtm.addRow(toObjectsData(lh));

}

tbllophoc.setModel(dtm);

} catch (ClassNotFoundException ex) {

Logger.getLogger(frmLopHoc.class.getName()).log(Level.SEVERE, null, ex);

}

} catch (InstantiationException ex) {

Logger.getLogger(frmLopHoc.class.getName()).log(Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

Logger.getLogger(frmLopHoc.class.getName()).log(Level.SEVERE, null, ex);

}

}

private static Object[] toObjectsData(LopHoc lh) {

Object[] objectsdata = {lh.getMalop(), lh.getTenlop(), lh.getMakhoa(), lh.getKhoahoc()};

return objectsdata;

}

public boolean checkinfo() {

Check c = new Check();

if (!c.checkID(jtfMaLop.getText())) {

JOptionPane.showMessageDialog(this, "Nhập mã lớp sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfMaLop.setText("");

jtfMaLop.requestFocus();

return false;

} else if (!c.checkSpace(jtfTenLop.getText()) || !c.check(jtfTenLop.getText())) {

JOptionPane.showMessageDialog(this, "Nhập tên lơp học sai", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfTenLop.setText("");

jtfTenLop.requestFocus();

return false;

} else if (!c.checkSpace(jtfKhoaHoc.getText()) || !c.check(jtfKhoaHoc.getText())) {

JOptionPane.showMessageDialog(this, "Lỗi khóa học", "Hãy nhập lại", JOptionPane.ERROR\_MESSAGE);

jtfKhoaHoc.setText("");

jtfKhoaHoc.requestFocus();

return false;

}

return true;

}

private void showAll() {

while (dtm.getRowCount() > 0) {

dtm.removeRow(0);

}

int selectedIndex = jcbMaKhoa.getSelectedIndex();

Khoa getkhoa = allKhoa.get(selectedIndex);

ArrayList<LopHoc> lhs = new LopHocDAO().findByIDKhoa(getkhoa.getMakhoa());

for (LopHoc lopHoc : lhs) {

Vector v = new Vector();

v.add(lopHoc.getMalop());

v.add(lopHoc.getTenlop());

v.add(lopHoc.getMakhoa());

v.add(lopHoc.getKhoahoc());

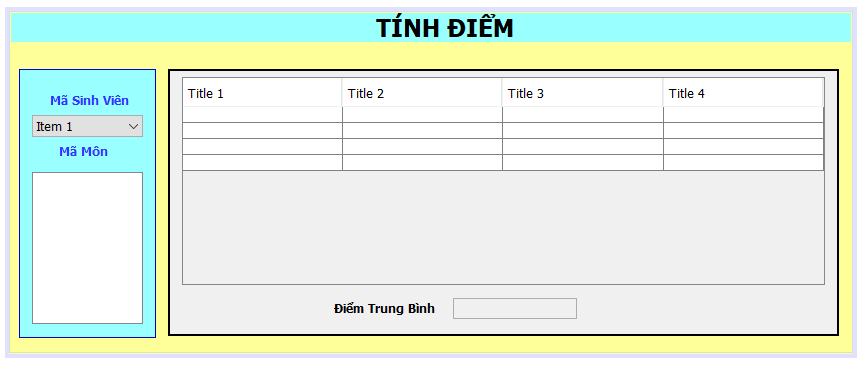
dtm.addRow(v);

}

ResetForm();

}

### Tính điểm (frmTinhdiem)



|  |  |
| --- | --- |
| **Sự Kiện** | **Công việc phải làm** |
| cboMasv | try {  IBangDiemDAO diemDAO = (IBangDiemDAO) Class.forName("DAO.BangDiemDAO").newInstance();  listmh = diemDAO.findMaMH(listsv.get(jcbMasv.getSelectedIndex()).getMasv());  dlm.removeAllElements();  for (BangDiem bangDiem : listmh) {  dlm.addElement(bangDiem.getMamh());  }  jlMaMon.setModel(dlm);  } catch (ClassNotFoundException ex) {  Logger.getLogger(frmTinhDiem.class.getName()).log(Level.SEVERE, null, ex);  } catch (InstantiationException ex) {  Logger.getLogger(frmTinhDiem.class.getName()).log(Level.SEVERE, null, ex);  } catch (IllegalAccessException ex) {  Logger.getLogger(frmTinhDiem.class.getName()).log(Level.SEVERE, null, ex);  } |
| lblMamon | if (jlMaMon.getSelectedValue() != null) {  try {  while (dtm.getRowCount() > 0) {  dtm.removeRow(0);  }  String maSV = jcbMasv.getSelectedItem().toString();  String maMon = jlMaMon.getSelectedValue().toString();  IBangDiemDAO diemDAO = (IBangDiemDAO) Class.forName("DAO.BangDiemDAO").newInstance();  ArrayList<BangDiem> list = diemDAO.loaddiem(maSV, maMon);  float diemTB = 0;  int tongHS = 0;  for (int i = 0; i < list.size(); i++) {  Vector v = new Vector();  v.add(list.get(i).getLanthi());  v.add(list.get(i).getHeso());  v.add(list.get(i).getDiem());  tongHS += list.get(i).getHeso();  diemTB += list.get(i).getDiem() \* list.get(i).getHeso();  dtm.addRow(v);  }  jtfDTB.setText((diemTB / tongHS) + "");  } catch (ClassNotFoundException ex) {  Logger.getLogger(frmTinhDiem.class.getName()).log(Level.SEVERE, null, ex);  } catch (InstantiationException ex) {  Logger.getLogger(frmTinhDiem.class.getName()).log(Level.SEVERE, null, ex);  } catch (IllegalAccessException ex) {  Logger.getLogger(frmTinhDiem.class.getName()).log(Level.SEVERE, null, ex);  }  } |

**Hướng Dẫn Đặt Mã Nguồn**

private DefaultTableModel dtm;

private DefaultListModel dlm;

private DefaultComboBoxModel dcbmMasv;

private DefaultComboBoxModel dcbmMamh;

ArrayList<BangDiem> listsv = null;

ArrayList<BangDiem> listmh = null;

public frmTinhDiem() {

initComponents();

dlm = new DefaultListModel();

dtm = new DefaultTableModel();

dcbmMamh = new DefaultComboBoxModel();

dcbmMasv = new DefaultComboBoxModel();

try {

//Load ma lop tai bang diem

IBangDiemDAO bangDiemDAO = (IBangDiemDAO) Class.forName("DAO.BangDiemDAO").newInstance();

listsv = bangDiemDAO.findMasv();

for (BangDiem bd : listsv) {

dcbmMasv.addElement(bd.getMasv());

}

}

jcbMasv.setModel(dcbmMasv);

dtm.addColumn("Lần Thi");

dtm.addColumn("Hệ Số");

dtm.addColumn("Điểm");

jtbtinhdiem.setModel(dtm);

} catch (ClassNotFoundException ex) {

Logger.getLogger(frmTinhDiem.class.getName()).log(Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

Logger.getLogger(frmTinhDiem.class.getName()).log(Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

Logger.getLogger(frmTinhDiem.class.getName()).log(Level.SEVERE, null, ex);

}

}

# Kiểm thử

## Kiểm thử form X

## Kiểm thử form Y

# Đóng gói và triển khai

## Hướng dẫn chuyển đổi jar thành exe

## Hướng dẫn cài đặt triển khai

## Hướng dẫn sử dụng phần mềm