



**Dự án mẫu**

**Báo cáo**

Mã số sinh viên : PS27852

Họ tên sinh viên : Võ Thanh Tùng

Lớp : SD18401

Học kỳ : SPRING 2024

Giảng viên hướng dẫn : Lê Văn Phụng

Mục Lục

[1 Phân tích 3](#_Toc155689322)

[1.1 Hiện trạng 3](#_Toc155689323)

[1.2 Yêu cầu hệ thống 3](#_Toc155689324)

[1.3 Use case 4](#_Toc155689325)

[2 Thiết kế 7](#_Toc155689326)

[2.1 Mô hình triển khai 7](#_Toc155689327)

[2.2.1 Sơ đồ quan hệ thực thể 8](#_Toc155689328)

[2.2.1.1 ERD Diagram level 1 8](#_Toc155689329)

[2.2.2 Thiết kế chi tiết các thực thể 9](#_Toc155689330)

[2.2.2.1 Nhân viên 9](#_Toc155689331)

[2.2.2.2 Chuyên đề 9](#_Toc155689332)

[2.2.2.3 Người học 10](#_Toc155689333)

[2.2.2.4 Khóa học 10](#_Toc155689334)

[2.2.2.5 Học viên 11](#_Toc155689335)

[2.3 Thiết kế giao diện 13](#_Toc155689336)

[2.3.1 Sơ đồ tổ chức giao diện 13](#_Toc155689337)

[2.3.2 Thiết kế giao diện cho các chức năng nghiệp vụ 14](#_Toc155689338)

[2.3.2.1 Màn hình chào 15](#_Toc155689339)

[2.3.2.2 16](#_Toc155689340)

[2.3.2.3 Đăng nhập 16](#_Toc155689341)

[2.3.2.4 Giao diện chính 18](#_Toc155689342)

[2.3.2.5 Quản lý nhân viên 22](#_Toc155689343)

[2.3.2.6 Quản lý người học 26](#_Toc155689344)

[2.3.2.7 Quản lý chuyên đề 30](#_Toc155689345)

[2.3.2.8 Quản lý khóa học 34](#_Toc155689346)

[2.3.2.9 Quản lý học viên 38](#_Toc155689347)

[2.3.2.10 Tổng hợp thống kê 42](#_Toc155689348)

[2.3.2.11 Giới thiệu 48](#_Toc155689349)

[2.3.2.12 Hướng dẫn sử dụng 50](#_Toc155689350)

[3 Thực hiện viết mã 50](#_Toc155689351)

[3.1 Viết mã tạo CSDL 50](#_Toc155689352)

[3.1.1 Tạo CSDL 51](#_Toc155689353)

[3.1.2 SQL truy vấn và thao tác bảng 54](#_Toc155689354)

[3.1.3 Các thủ tục lưu tổng hợp thống kê 55](#_Toc155689355)

[3.2.1 Lớp hỗ trợ 58](#_Toc155689356)

# Phân tích

## Hiện trạng

LapTrinhCity, trung tâm hàng đầu về đào tạo tin học ngắn hạn, đang cung cấp những khóa học chất lượng cao với các chuyên đề đa dạng như Photoshop, Java, Web…

Quản lý thông tin về khóa học, học viên, bảng điểm và doanh thu tại Trung tâm đang sử dụng Excel. Tuy nhiên, với sự gia tăng đáng kể về số lượng học viên và dữ liệu, việc quản lý thông qua Excel trở nên cực kỳ khó khăn, dễ xảy ra sai sót và không đảm bảo tính bảo mật.

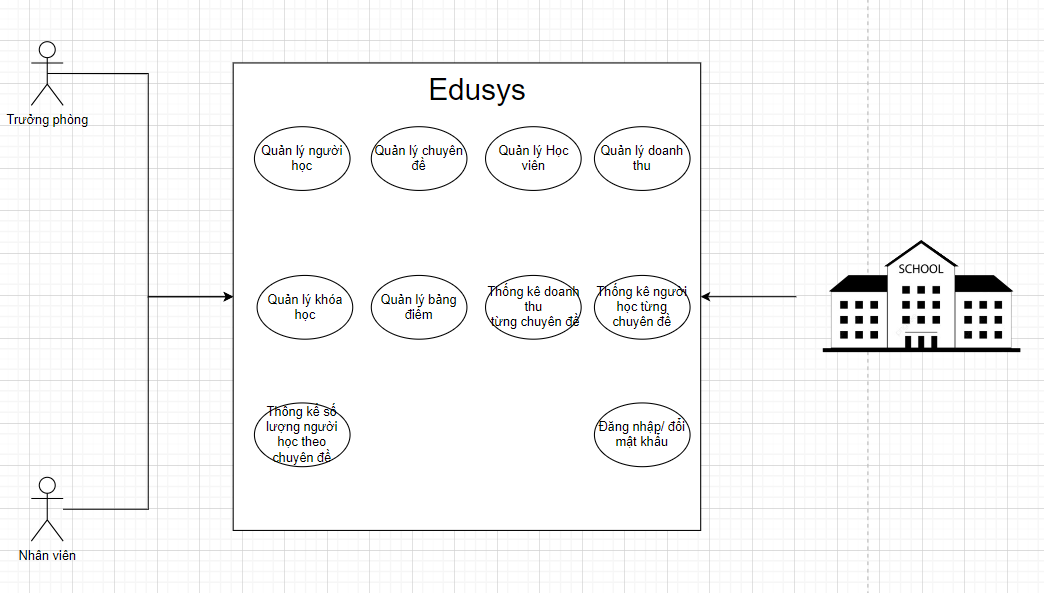
## Yêu cầu hệ thống

LapTrinhCity đặt hy vọng vào việc tạo ra một phần mềm giải quyết những thách thức đó.

* **Yêu cầu chức năng nghiệp vụ**
  + Quản lý người đăng ký học
  + Quản lý các chuyên đề
  + Quản lý các khóa học được tạo ra từ các chuyên đề
  + Quản lý học viên của các khóa học
  + Tổng hợp bảng điểm của các khóa học
  + Thống kê doanh thu, số lượng người học từng chuyên đề theo năm
  + Tổng hợp số lượng người học đăng ký theo tháng
* **Yêu cầu về bảo mật**
  + Tất cả mọi thành viên phải đăng nhập mới sử dụng được phần mềm
  + Trưởng phòng đào tạo được phép thực hiện tất cả các chức năng
  + Nhân viên phòng đào tạo không được phép xóa dữ liệu và cũng không được xem thông tin về doanh thu
* **Yêu cầu về môi trường công nghệ**
  + Ứng dụng phải được thực với công nghệ Swing và JDBC chạy trên mọi hệ điều hành với môi trường JDK tối thiểu 1.8
  + Hệ quản trị CSDL SQL Server 2008 trở lên

## Use case

Dựa trên yêu cầu hệ thống từ phía khách hàng, chúng ta có thể tạo ra một sơ đồ use case để trình bày một cách tổng quát về các chức năng cũng như vai trò của người sử dụng trong hệ thống.

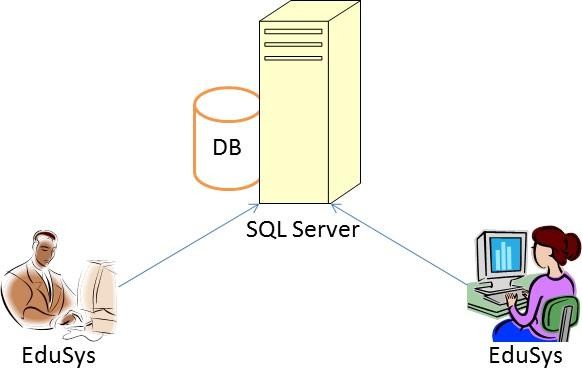


* **Chi tiết các chức năng**
  + Mỗi chức năng quản lý bao gồm các chức năng con
    - Xem: xem tất cả và chi tiết một mục
    - Thêm: thêm mới vào cơ sở dữ liệu
    - Xóa: xóa theo mã
    - Sửa: cập nhật dữ liệu đang xem
    - Tìm kiếm: tìm kiếm theo điều kiện
    - Điều hướng: di chuyển đến dữ liệu của bản ghi chi tiết trước và sau
    - Các chức năng thao tác dữ liệu (thêm, sửa, xóa) cần được kiểm lỗi hợp lý với dữ liệu
  + Chức năngtổng hợp thống kê số liệu
    - Bảng điểm theo khóa học phải có cấu trúc là:
      * Mã người học
      * Họ và tên
      * Điểm
    - Tổng hợp điểm từng chuyên đề
      * Chuyên đề
      * Số học viên
      * Điểm cao nhất
      * Điểm thấp nhất
      * Điểm trung bình
    - Doanh thu từng chuyên đề theo năm phải có cấu trúc:
      * Chuyên đề
      * Số khóa
      * Số học viên
      * Doanh thu
      * Học phí thấp nhất
      * Học phí cao nhất
      * Học phí trung bình
    - Thống kê số người học từng năm phải có cấu trúc:
      * Năm
      * Số học viên
      * Ngày đăng ký sớm nhất
      * Ngày đăng ký muộn nhất
  + Chức năng đăng nhập và đăng xuất
    - Đăng nhập: để sử dụng phần mềm, bắt buộc phải đăng nhập
    - Đăng xuất: đăng nhập lại với người dùng khác hoặc dừng lại đi đâu đó
* **Chi tiết về yêu cầu bảo mật**
  + Tất cả phải đăng nhập mới được sử dụng các chức năng trong phần mềm
  + Trưởng phòng: được phép sử dụng tất cả
  + Nhân viên:
    - Không được phép thực hiện chức năng xóatrong các chức năng quản lý
    - Không được phép xem chức năng thống kê doanh thu
* **Thông tin các thực thể**
  + Người học
    - Mã người đăng ký học
    - Họ và tên
    - Ngày sinh
    - Giới tính
    - Số điện thoại
    - Email
    - Ghi chú
  + Chuyên đề
    - Mã chuyên đề
    - Tênchuyên đề
    - Học phí
    - Thời lượng (tính theo giờ)
    - Hình logo
    - Mô tả chuyên đề
  + Khóa học
    - Mã khóa học
    - Mã chuyên đề
    - Học phí
    - Thời lượng
    - Ngày khai giảng
    - Ghi chú
  + Học viên
    - Mã học viên (số báo danh)
    - Mã khóa học
    - Mã người học
    - Điểm trung bình
  + Nhân viên
    - Mã nhân viên (dùng để đăng nhập)
    - Mật khẩu
    - Họ và tên
    - Vai trò (trưởng phòng hay nhân viên)
  + Chú ý: Học phí và thời lượng của chuyên đề có thể bị thay đổi mà không làm ảnh hưởng đến học phí và thời lượng của những người đã học.

# Thiết kế

## Mô hình triển khai

Phần mềm này được thiết kế để phục vụ đa người dùng, nhưng dữ liệu được lưu trữ tập trung tại một nơi.



* Cần một máy cài SQL Server 2008+. Máy này cần hệ điều hành window xp trở lên
* Các máy nhân viên phòng đào tạo cài phần mềm edusys. Các máy nhân viên cần JDK 1.8+ với hệ điều hành bất ký
  1. **THIẾT KẾ CSDL**

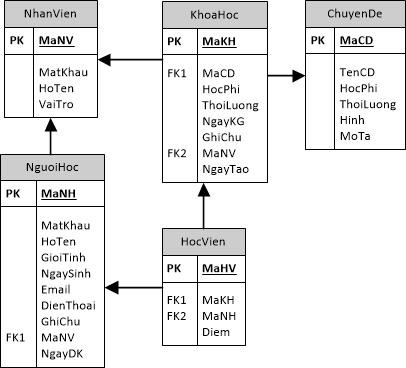
## Sơ đồ quan hệ thực thể

Trong hệ thống đã được mô tả trên chúng ta thấy có các thực thể sau

* Chuyên đề: quản lý chuyên đề đào tạo ngắn hạn
* Khóa học: quản lý thông tin khóa học được tạo ra từ chuyên đề. Có nghĩa là mỗi chuyên đề có thể tạo ra nhiều khóa học.
* Người học: quản lý thông tin của người đăng ký học
* Học viên: Học viên là người học đăng ký học chuyên đề. Mỗi học viên có thể đăng ký học nhiều chuyên đề khác nhau mà không cần phải nhập lại.
* Nhân viên: là người quản trị cần có thông tin đăng nhập đồng thời được ghi nhận mỗi khi nhân viên tạo người học mới hoặc khóa học mới.

Sau khi phân tích chúng ta có thể thấy các thực thể có mối quan hệ với nhau như sơ đồ ERD sau

### ERD Diagram level 1

* + - 1. ***ERD Diagram level 2***

Yêu cầu ràng buộc quan hệ:

* + - * + Tất cả các relationship đều là CASCADE (cập nhật dây chuyền)
        + Tất cả các relationship đều là NO ACTION ngoại trừ KhoaHoc-HocVien là CASCADE

## Thiết kế chi tiết các thực thể

### Nhân viên

Bảng NhanVien lưu thông tin nhân viên phòng đào tạo có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaNV | NVARCHAR(20) | PK, NOT NULL | Mã nhân viên |
| MatKhau | NVARCHAR(50) | NOT NULL | Mật khẩu đăng nhập |
| HoTen | NVARCHAR(50) | NOT NULL | Họ và tên |
| VaiTro | BIT | DEFAULT 0 | Vai trò, 1 là trưởng phòng |

### Chuyên đề

Bảng ChuyenDe lưu thông tin các chuyên đề có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |

|  |  |  |  |
| --- | --- | --- | --- |
| MaCD | NCHAR(5) | PK, NOT NULL | Mã chuyên đề |
| TenCD | NVARCHAR(50) | NOT NULL | Tên chuyên đề |
| HocPhi | FLOAT | NOT NULL | Học phí |
| ThoiLuong | INT | NOT NULL | Thời lượng |
| Hinh | NVARCHAR(50) | NOT NULL | Hình ảnh |
| MoTa | NVARCHAR(255) | NOT NULL | Mô tả |

### Người học

Bảng NguoiHoc lưu thông tin các những người đã đến trung tập đăng ký học có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaNH | NCHAR(7) | PK, NOT NULL | Mã người học |
| HoTen | NVARCHAR(50) | NOT NULL | Họ và tên |
| GioiTinh | BIT | DEFAULT 1 | Giới tính |
| NgaySinh | DATE | NOT NULL | Ngày sinh |
| DienThoai | NVARCHAR(24) | NOT NULL | Điện thoại |
| Email | NVARCHAR(50) | NOT NULL | Email |
| GhiChu | NVARCHAR(255) | NULL | Ghi chú |
| MaNV | NVARCHAR(20) | FK, NOT NULL | Mã nhân viên nhập |
| NgayDK | DATE | DEFAULT getdate() | Ngày đăng ký của người học |

### Khóa học

Bảng KhoaHoc lưu thông tin các khóa học được tạo ra từ các chuyên đề có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaKH | INT | PK, Tự tăng | Mã khóa học |
| MaCD | NCHAR(5) | FK, NOT NULL | Mã chuyên đề |
| HocPhi | FLOAT | NOT NULL | Học phí |
| ThoiLuong | INT | NOT NULL | Thời lượng, tính bằng giờ |

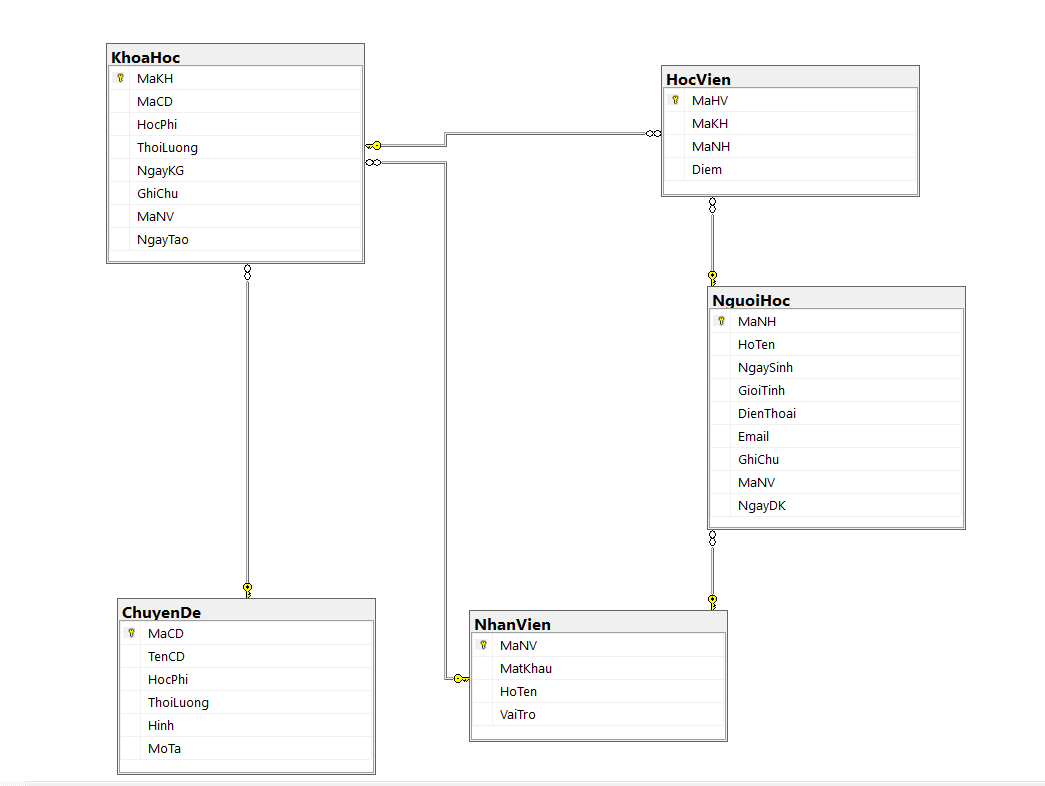
|  |  |  |  |
| --- | --- | --- | --- |
| NgayKG | DATE | NOT NULL | Ngày khai giảng |
| GhiChu | NVARCHAR(255) | NULL | Ghi chú về khóa học |
| MaNV | NVARCHAR(20) | FK, NOT NULL | Mã nhân viên nhập |
| NgayTao | DATE | DEFAULT getdate() | Ngày đăng ký của người học |

### Học viên

Bảng HocVien lưu thông tin các học viên tham gia vào các khóa học có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaHV | INT | PK, Tự tăng | Mã học viên (số báo danh) |
| MaKH | INT | FK, NOT NULL | Mã khóa học |
| MaNH | NCHAR(7) | FK, NOT NULL | Mã người học |
| Diem | FLOAT | DEFAULT -1 | Điểm cuối môn |

*Chú ý: (MaKH và MaNH) là duy nhất*

**

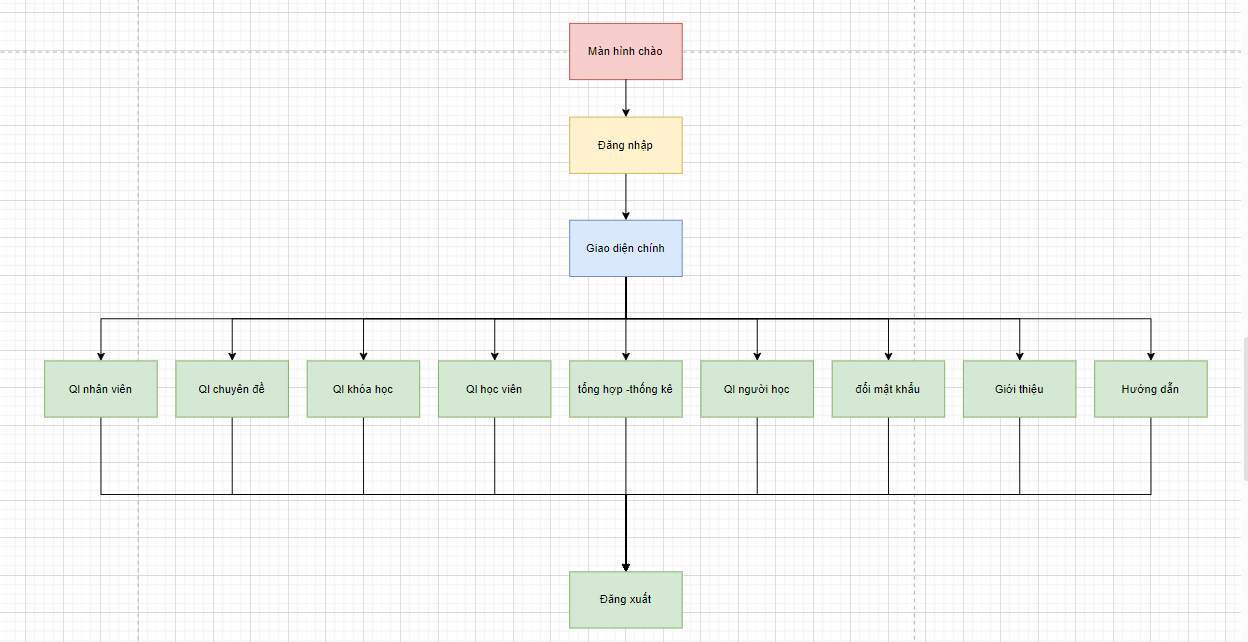
## Thiết kế giao diện

## Sơ đồ tổ chức giao diện

Giao diện chính là điểm truy cập vào menu chính và thanh công cụ, từ đó người dùng có thể tiếp cận các giao diện thành viên để thực hiện các chức năng trong hệ thống một cách thuận tiện và dễ dàng..

Do yêu cầu của hệ thống, mọi nhân viên cần đăng nhập trước khi sử dụng ứng dụng, vì vậy form đăng nhập được hiển thị trước để yêu cầu việc đăng nhập

Bên cạnh đó, mỗi ứng dụng trong quá trình khởi động cần hiển thị một màn hình chào đón kèm theo thanh tiến trình, giúp người dùng có cảm giác rõ ràng về việc ứng dụng đang được khởi động



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

Thiết kế giao diện nhằm phục vụ cho các chức năng nghiệp vụ trong hệ thống, đảm bảo tính thẩm mỹ và hiệu quả trong trải nghiệm người dùng.

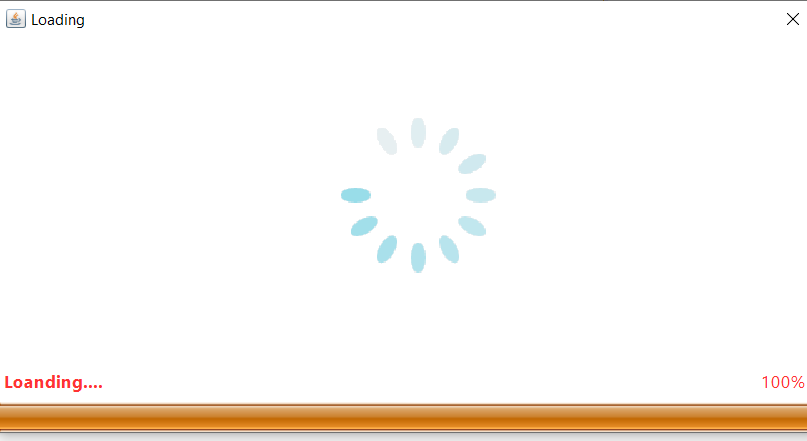
Lưu ý quan trọng:

* 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

|  |  |  |
| --- | --- | --- |
| **Control** | **Prefix** | **Mô tả** |
| JLabel | lbl | Nhãn |
| JButton | btn | Nút |
| JTextField, JTextArea, JPassword, JTextPane | txt | Các loại ô nhập |
| JRadioButton | rdo | Hộp chọn |

|  |  |  |
| --- | --- | --- |
| JCheckBox | chk | Hộp kiểm |
| ButtonGroup | bgr | Nhóm các button |
| JComboBox | cbo | Hộp chọn xổ xuống |
| JListBox | lst | Hộp chọn nhiều |
| JTree | tree | Cây |
| JMenu | mnu | Thực đơn |
| JMenuItem | mni | Mục menu |
| JPanel | pnl | Khung chứa |
| JTabbedPane | tab |  |
| JTable | tbl |  |
| … |  |  |

### Màn hình chào

******

Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | ChaoJDialog | JDialog | undecorated:true Layout:BorderLayout  preferredSize: [514, 350] |
| 2 | lblLogo | JLabel | Icon:logo.png Layout.Derection: center |
| 3 | progressBar | JProgressBar | Layout.Derection: last stringPainted:true |

### 

### Đăng nhập

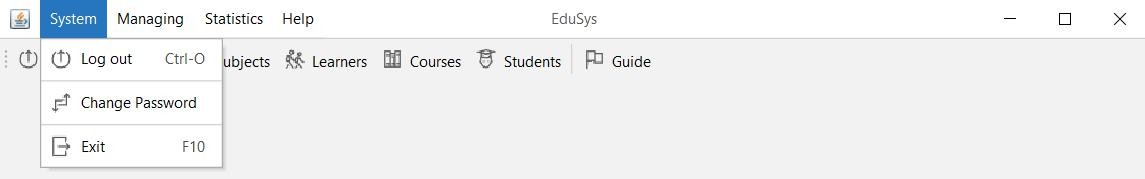
zCác thành phần giao diện

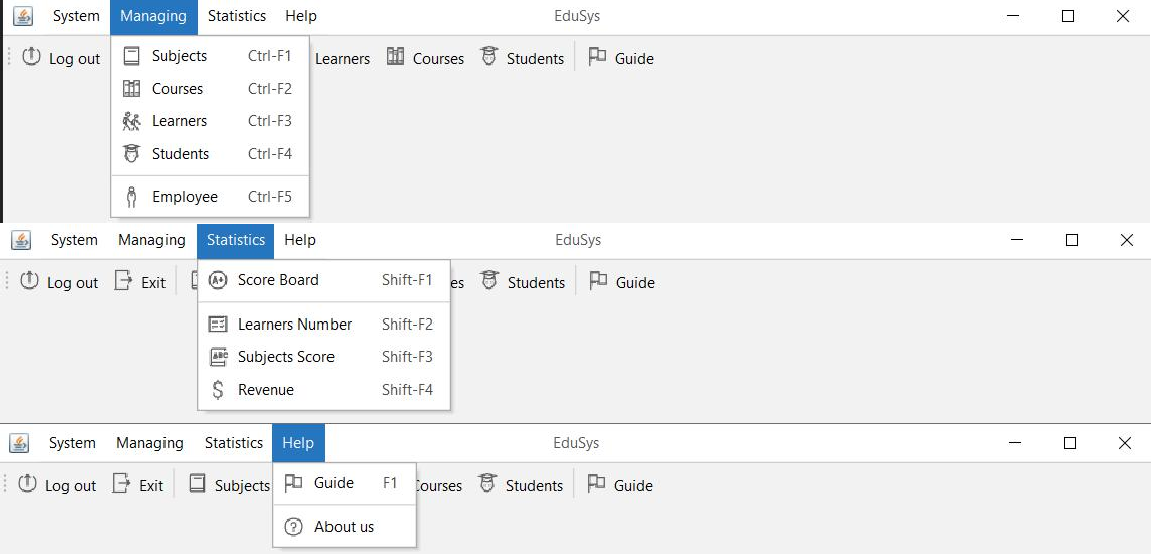
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Componel** | **Kiểu** | **Thuộc tính** |
| 1 | DangNhapJDialog | JDialog | title:HỆ THỐNG QUẢN LÝ ĐÀO TẠO  Layout: BorderLayout  defaultCloseOperation:DO\_NOTHIN G |
| 2 | lblTitle | JLabel | Text:ĐĂNG NHẬP  Font:Tahoma, Bold, 18  Foreground: [0,102,51] |

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | lblImage | JLabel | Icon: Security.png Layout.Direction:before |
| 4 | pnlForm | JPanel | Layout.Derection:center |

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | lblMaNV | JLabel | Text: Tên đăng nhập |
| 6 | txtMaNV | JTextField |  |
| 7 | lblMatKhau | JLabel | Text: Mật khẩu |
| 8 | txtMatKhau | JPassword |  |
| 9 | btnDangNhap | JButton | Text:Đăng nhập |
| 10 | btnKetThuc | JButton | Text:Kết thúc |

### Giao diện chính





Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Componel** | **Kiểu** | **Thuộc tính** |
| 1 | MainJFrame | JFrame | Title: HỆ THỐNG QUẢN LÝ ĐÀO TẠO  Layout:BorderLayout |
| 2 | mnuHeThong | JMenu | Text:Hệ thống |
| 3 | mniDangNhap | JMenuItem | Text:Đăng nhập Icon: Key.png Accelerator:  Ctrl+L |
| 4 | mniDangXuat | JMenuItem | Text:Đăng xuất Icon: Log out.png Accelerator:  Ctrl+O |

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | mniDoiMatKhau | JMenuItem | Text:Đổi mật khẩu Icon: Refresh.png |
| 6 | mniKetThuc | JMenuItem | Text:Kết thúc Icon: Stop.png Accelerator:  F10 |

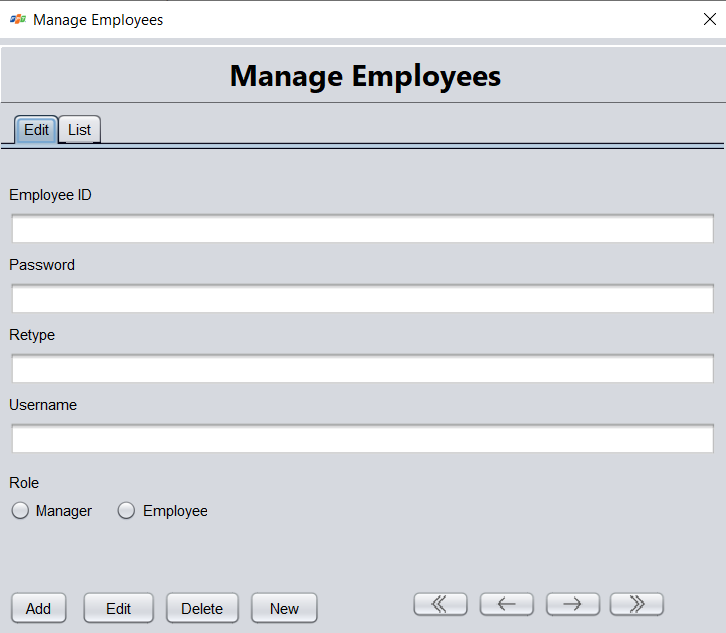
|  |  |  |  |
| --- | --- | --- | --- |
| 7 | mnuQuanLy | JMenu | Text:Quản lý |
| 8 | mniQLNguoiHoc | JMenuItem | Text:Người học Icon: Conference.png |
| 9 | mniQLNhanVien | JMenuItem | Text:Nhân viên Icon:  Lists.png |
| 10 | mniQLKhoaHoc | JMenuItem | Text:Khóa học  Icon: Certificate.png |
| 11 | mniQLChuyenDe | JMenuItem | Text:Chuyên đề Icon: User group.png |
| 12 | mnuThongKe | JMenu | Text:Thống kê |
| 13 | mniTKNguoiHoc | JMenuItem | Text:Người học từng năm Icon: Clien list.png |
| 14 | mniTKBangDem | JMenuItem | Text:Bảng điểm khóa… Icon: Card file.png |
| 15 | mniTKKhoaHoc | JMenuItem | Text:Điểm từng khóa Icon: Bar chart.png |
| 16 | mniTKDoanhThu | JMenuItem | Text:Doanh thu từng chuyên đề… Icon: Dollar.png |
| 17 | mnuTroGiup | JMenu | Text:Trợ giúp |
| 18 | mniHuongDan | JMenuItem | Text:Hướngdẫnsửdụng Icon: Globe.png  Accelerator: F1 |

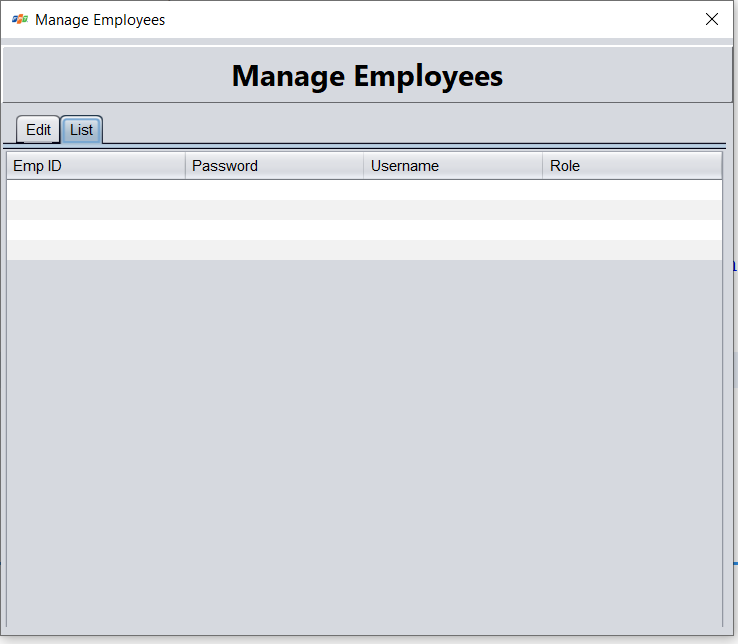
|  |  |  |  |
| --- | --- | --- | --- |
| 19 | mniGioiThieu | JMenuItem | Text:Giới thiệu sản phẩm Icon: Brick house.png |
| 20 | toolBar | JToolBar | Layout.Direction: first |
| 21 | btnDangXuat | JButton | Text:Đăng xuất Icon:Exit.png Margin: [2, 10, 2,  10] |

|  |  |  |  |
| --- | --- | --- | --- |
| 22 | btnKetThuc | JButton | Text:Kết thúc Icon:Stop.png Margin: [2, 10, 2,  10] |
| 23 | btnChuyenDe | JButton | Text:Chuyên đề Icon:Lists.png Margin: [2, 10, 2,  10] |
| 24 | btnNguoiHoc | JButton | Text:Người học Icon: Conference.png Margin: [2, 10, 2, 10] |
| 25 | btnKhoaHoc | JButton | Text:Khóa học  Icon: Certificate.png Margin: [2, 10, 2,  10] |
| 26 | btnHuongDan | JButton | Text:Hướng dẫn Icon: Globe.png Margin: [2, 10, 2,  10] |
| 27 | lblNoiDung | JLabel | Layout.Direction:center Icon:logo.png |
| 28 | pnlTrangThai | JPanel | Layout.Direction:last Layout:BorderLayou  t |

|  |  |  |  |
| --- | --- | --- | --- |
| 29 | lblTrangThai | JLabel | Layout.Direction:center Icon: Info.png  Text:Hệ quản lý đào tạo |
| 30 | lblDongHo | JLabel | Layout.Direction:after Icon:Alarm.png Text:10:55 PM |

### Quản lý nhân viên

******

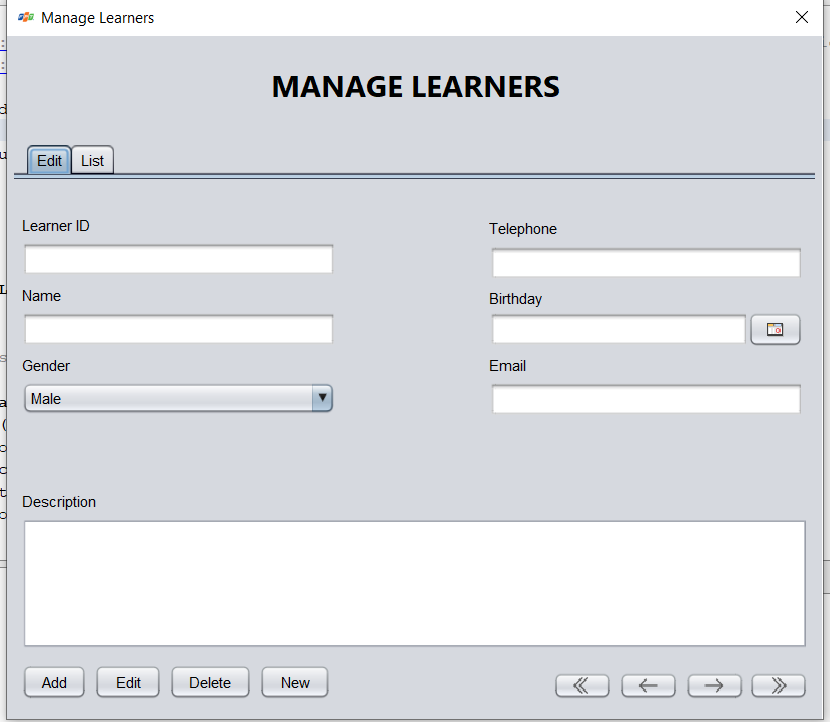


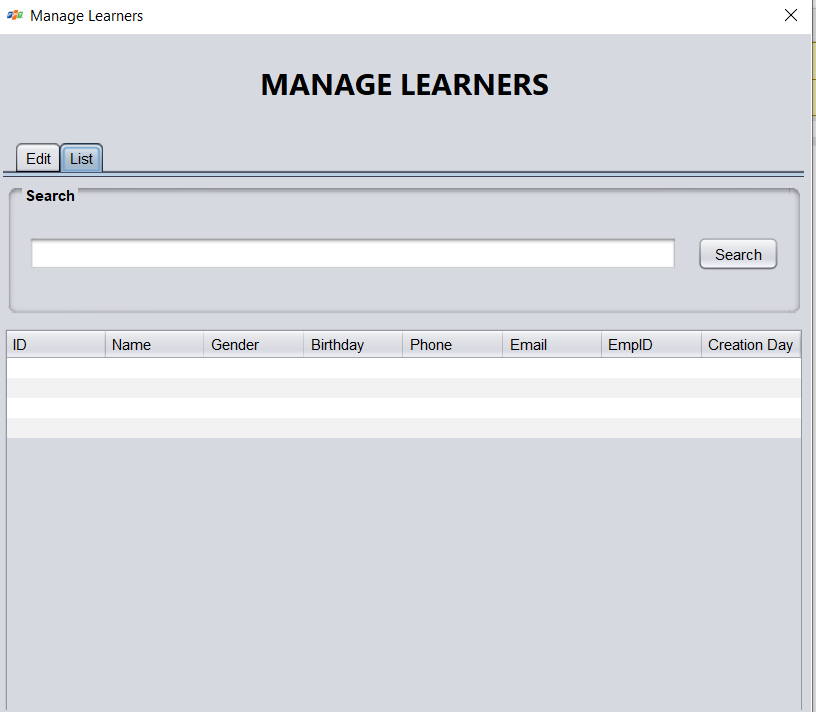
Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | NhanVienJFrame | JFrame | Title: QUẢN LÝ NHÂN VIÊN  defaultCloseOperation:DISPONSE |
| 2 | lblTitle | JLabel | Text: QUẢN LÝ NHÂN VIÊN QUẢN TRỊ  Foreground: [0,0,204]  Font:Tahoma, Bold, 14 |
| 3 | tabs | JTabbedPane |  |
| 4 | pnlEdit | JPanel | Tab Title: CẬP NHẬT |
| 5 | pnlList | JPanel | Tab Title: DANH SÁCH |
| 6 | lblMaNV | JLabel | Text: Mã nhân viên |
| 7 | txtMaNV | JTextField |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 8 | lblMatKhau | JLabel | Text: Mật khẩu |
| 9 | txtMatKhau | JPassword |  |
| 10 | lblXacNhanMK | JLabel | Text: Xác nhận mật khẩu |
| 11 | txtXacNhanMK | JPassword |  |
| 12 | lblHoTen | JLabel | Text: Họ và tên |
| 13 | txtHoTen | JTextField |  |
| 14 | lblVaiTro | JLabel | Text: Vai trò |
| 15 | bgrVaiTro | ButtonGroup |  |
| 16 | rdoTruongPhong | JRadioButton | Text: Trưởng phòng  Selected: true buttonGroup: bgrVaiTro |
| 17 | rdoNhanVien |  | Text: Nhân viên buttonGroup: bgrVaiTro |
| 18 | btnInsert | JButton | Text:Thêm |
| 19 | btnUpdate | JButton | Text:Sửa |
| 20 | btnDelete | JButton | Text:Xóa |
| 21 | btnClear | JButton | Text:Mới |
| 22 | btnFirst | JButton | Text: |< |
| 23 | btnPrev | JButton | Text: << |
| 24 | btnNext | JButton | Text: >> |
| 25 | btnLast | JButton | Text: >| |
| 26 | tblGridView | JTable | Model: như hình Height: 25 |

### Quản lý người học

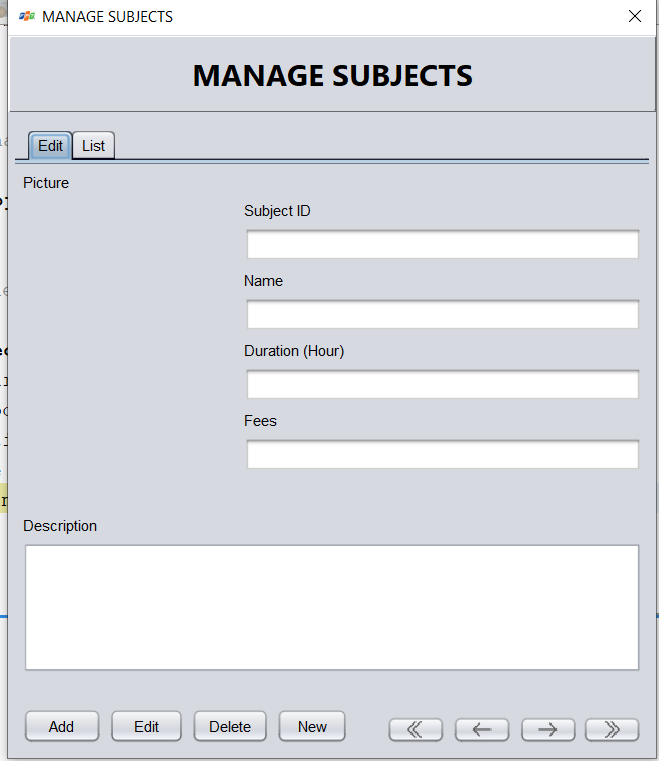


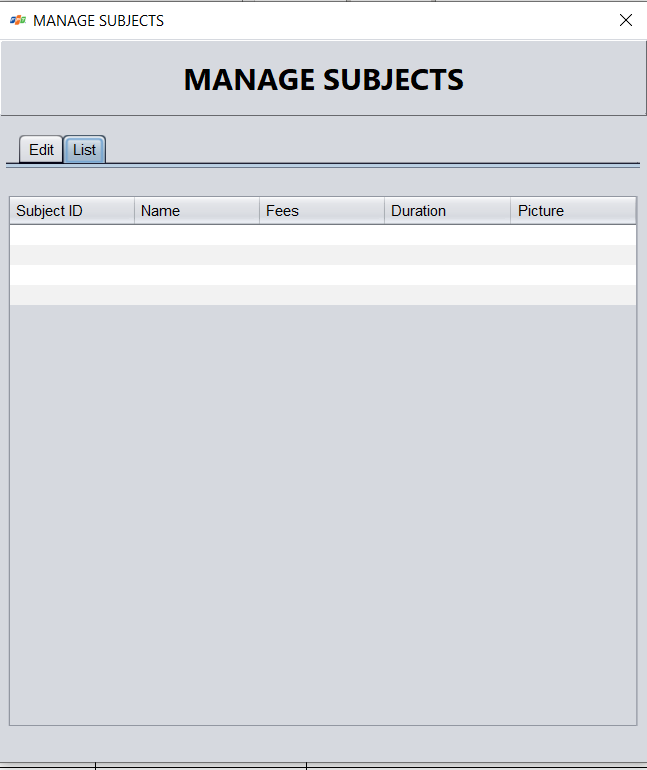
Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | NguoiHocJFrame | JFrame | Title: QUẢN LÝ NGƯỜI HỌC  defaultCloseOperation:DISPONSE |
| 2 | lblTitle | JLabel | Text: QUẢN LÝ NGƯỜI HỌC  Foreground: [0,0,204]  Font:Tahoma, Bold, 14 |
| 3 | tabs | JTabbedPane |  |
| 4 | pnlEdit | JPanel | Tab Title: CẬP NHẬT |
| 5 | pnlList | JPanel | Tab Title: DANH SÁCH |
| 6 | lblMaNH | JLabel | Text: Mã người học |
| 7 | txtMaNH | JTextField |  |
| 8 | lblHoTen | JLabel | Text: Họ và tên |
| 9 | txtHoTen | JTextField |  |
| 10 | lblGioiTinh | JLabel | Text: Giới tính |
| 11 | cboGioiTinh | JComboBox | Model: Nam, Nữ |
| 12 | lblNgaySinh | JLabel | Text: Ngày sinh |
| 13 | txtNgaySinh | JTextField |  |
| 14 | lblDienThoai | JLabel | Text: Điện thoại |
| 15 | txtDienThoai | JTextField |  |
| 16 | lblEmail | JLabel | Text: Email |
| 17 | txtEmail | JTextField |  |
| 18 | lblGhiChu | JLabel | Text: Ghi chú |
| 19 | txtGhiChu | JTextArea |  |
| 20 | btnInsert | JButton | Text:Thêm |

|  |  |  |  |
| --- | --- | --- | --- |
| 21 | btnUpdate | JButton | Text:Sửa |
| 22 | btnDelete | JButton | Text:Xóa |
| 23 | btnClear | JButton | Text:Mới |
| 24 | btnFirst | JButton | Text: |< |
| 25 | btnPrev | JButton | Text: << |
| 26 | btnNext | JButton | Text: >> |
| 27 | btnLast | JButton | Text: >| |
| 28 | tblGridView | JTable | Model: như hình Height: 25 |
| 28 | pnlTimKiem | JPanel | Border: TitledBorder(Title: Tìm Kiếm) |
| 30 | txtTimKiem | JTextField |  |
| 31 | btnTimKiem | JButton | Tìm |

### Quản lý chuyên đề

******

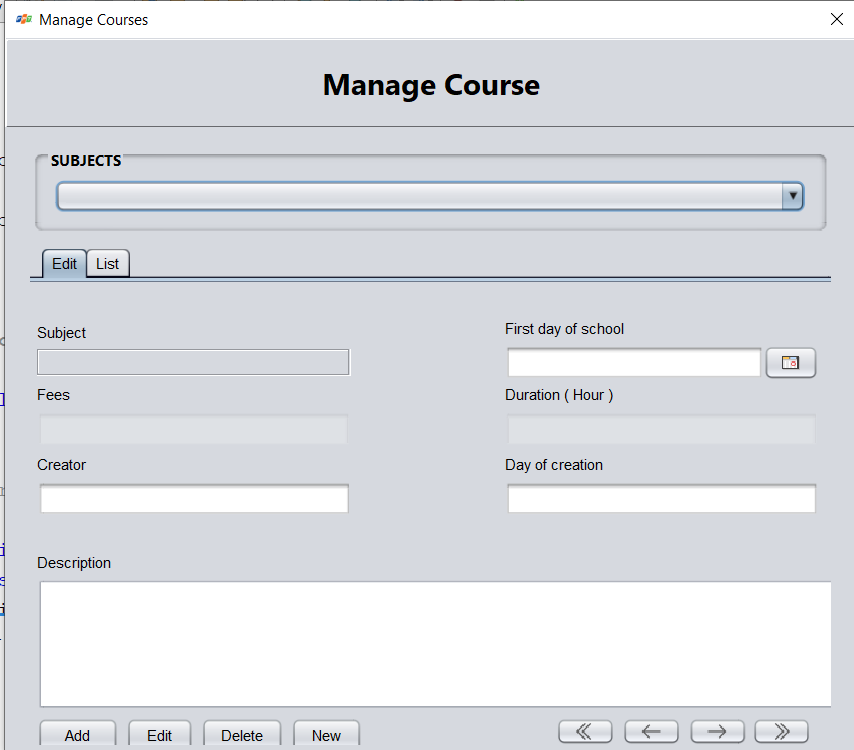


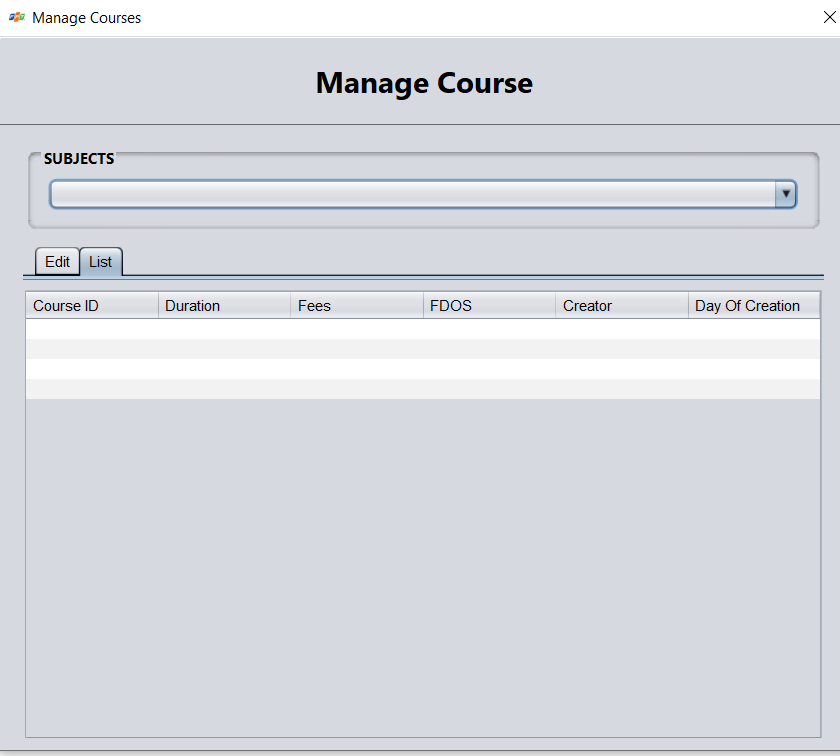
Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | ChuyenDeJFrame | JFrame | Title: QUẢN LÝ CHUYÊN ĐỀ  defaultCloseOperation:DISPONSE |
| 2 | lblTitle | JLabel | Text: QUẢN LÝ CHUYÊN ĐỀ  Foreground: [0,0,204]  Font:Tahoma, Bold, 14 |
| 3 | tabs | JTabbedPane |  |
| 4 | pnlEdit | JPanel | Tab Title: CẬP NHẬT |
| 5 | pnlList | JPanel | Tab Title: DANH SÁCH |

|  |  |  |  |
| --- | --- | --- | --- |
| 6 | lblMaCD | JLabel | Text: Mã chuyên đề |
| 7 | txtMaCD | JTextField |  |
| 8 | lblTenCD | JLabel | Text: Tên chuyên đề |
| 9 | txtTenCD | JTextField |  |
| 10 | lblHocPhi | JLabel | Text: Học phí |
| 11 | txtHocPhi | JTextField |  |
| 12 | lblThoiLuong | JLabel | Text: Thời lượng (giờ) |
| 13 | txtThoiLuong | JTextField |  |
| 14 | lblMoTa | JLabel | Text: Mô tả chuyên đề |
| 15 | txtMoTa | JTextArea |  |
| 16 | lblHinh | JLabel | Text: Hình logo |
| 17 | lblHinh | JLabel | Text: |
| 18 | btnInsert | JButton | Text:Thêm |
| 19 | btnUpdate | JButton | Text:Sửa |
| 20 | btnDelete | JButton | Text:Xóa |
| 21 | btnClear | JButton | Text:Mới |
| 22 | btnFirst | JButton | Text: |< |
| 23 | btnPrev | JButton | Text: << |
| 24 | btnNext | JButton | Text: >> |
| 25 | btnLast | JButton | Text: >| |
| 26 | tblGridView | JTable | Model: như hình Height: 25 |
| 27 | fileChooser | JFileChooser | dialogTitle: Chọn logo chuyên đề |

### Quản lý khóa học

******



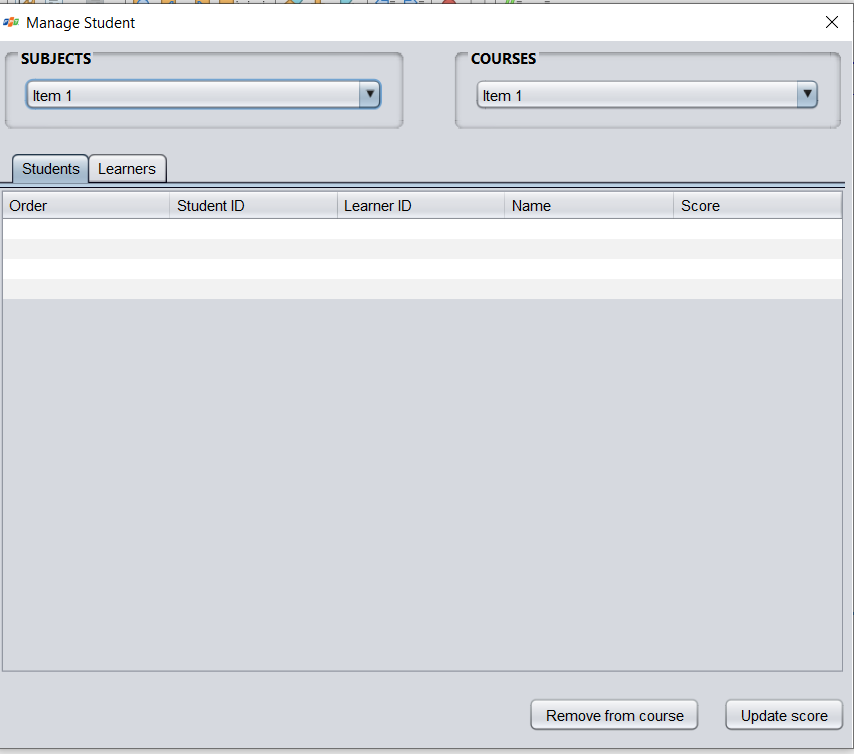
Các thành phần giao diện

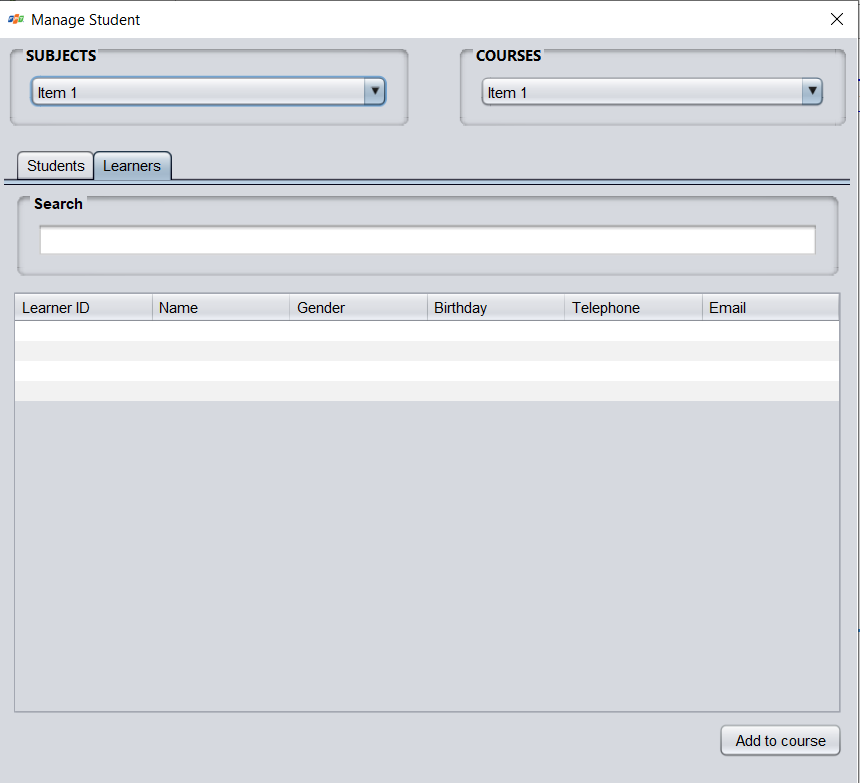
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | KhoaHocJFrame | JFrame | Title: QUẢN LÝ KHÓA HỌC |
| 2 | lblTitle | JLabel | Text: QUẢN LÝ KHÓA HỌC  Foreground: [0,0,204]  Font:Tahoma, Bold, 14 |
| 3 | tabs | JTabbedPane |  |
| 4 | pnlEdit | JPanel | Tab Title: CẬP NHẬT |
| 5 | pnlList | JPanel | Tab Title: DANH SÁCH |
| 6 | lblChuyenDe | JLabel | Text: Chuyên đề |
| 7 | cboChuyenDe | JComboBox | Model: Editable: false |

|  |  |  |  |
| --- | --- | --- | --- |
| 8 | lblNgayKG | JLabel | Text: Ngày khai giảng |
| 9 | txtNgayKG | JTextField |  |
| 10 | lblHocPhi | JLabel | Text: Học phí |
| 11 | txtHocPhi | JTextField | Editable: false Enable: false |
| 12 | lblThoiLuong | JLabel | Text: Thời lượng (giờ) |
| 13 | txtThoiLuong | JTextField | Editable: false Enable: false |
| 14 | lblMaNV | JLabel | Text: Người tạo |
| 15 | txtMaNV | JTextField | Editable: false Enable: false |
| 16 | lblNgayTao | JLabel | Text: Ngày tạo |
| 17 | txtNgayTao | JLabel | Editable: false Enable: false |
| 18 | lblGhiChu | JLabel | Text: Ghi chú |
| 19 | txtGhiChu | JTextArea |  |
| 20 | btnInsert | JButton | Text:Thêm |
| 21 | btnUpdate | JButton | Text:Sửa |
| 22 | btnDelete | JButton | Text:Xóa |
| 23 | btnClear | JButton | Text:Mới |
| 24 | btnFirst | JButton | Text: |< |
| 25 | btnPrev | JButton | Text: << |
| 26 | btnNext | JButton | Text: >> |
| 27 | btnLast | JButton | Text: >| |
| 28 | tblGridView | JTable | Model: như hình Height: 25 |

|  |  |  |  |
| --- | --- | --- | --- |
| 29 | btnHocVien | JButton | Text: Học viên |

### Quản lý học viên

******



Các thành phần giao diện

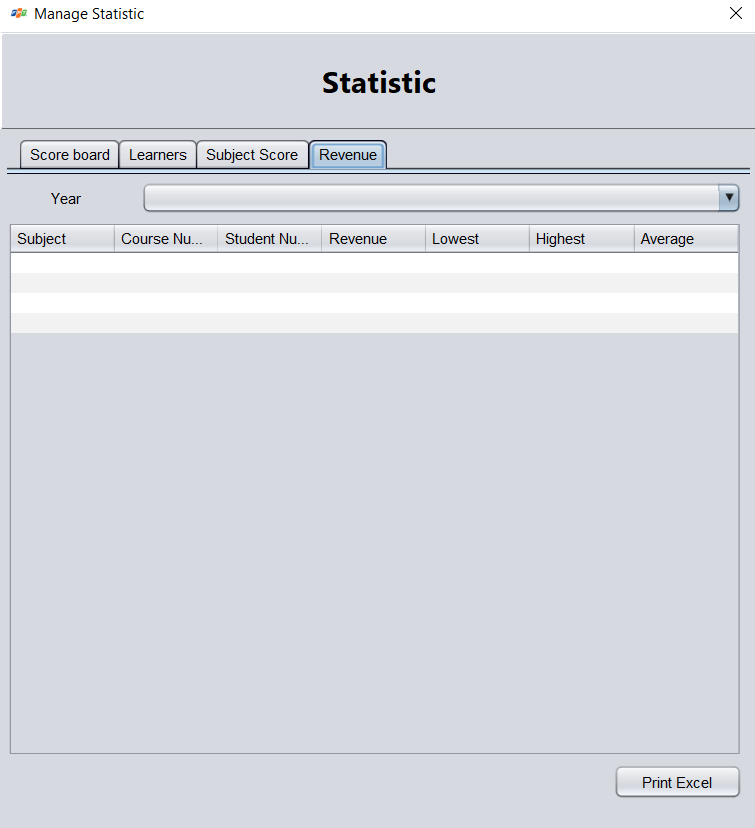
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | HocVienJFrame | JFrame | Title: QUẢN LÝ HỌC VIÊN KHÓA HỌC  defaultCloseOperation:DISPONSE |
| 2 | pnlHVKhac | JPanel | Border: TitledBorder(Title: HỌC VIÊN KHÁC) |
| 3 | pnlHVKH | JPanel | Border: TitledBorder(Title: HỌC VIÊN CỦA KHÓA HỌC) |
| 4 | cboNguoiHoc | JComboBox | Model: Editable: false |
| 5 | txtDiem | JTextField |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 6 | btnThem | JButton | Text: Thêm |
| 7 | tblGridView | JTable | Model: như hình Height: 25 |
| 8 | bgrPhanLoai | ButtonGroup |  |
| 9 | rdoTatCa | JButtonGroup | Text: Tất cả Selected: true  ButtonGroup:bgrPhanLoai |
| 10 | rdoDaNhap | JButtonGroup | Text: Đã nhập điểm ButtonGroup:bgrPhanLoai |
| 11 | rdoChuaNhap | JButtonGroup | Text: Chưa nhập điểm ButtonGroup:bgrPhanLoai |

### Tổng hợp thống kê







Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | ThongKeJFrame | JFrame | Title: TỔNG HỢP – THỐNG KÊ  defaultCloseOperation:DISPONSE |
| 2 | lblTitle | JLabel | Text: TỔNG HỢP – THỐNG KÊ |
| 3 | tabs | JTabbedPane |  |
| 4 | pnlNguoiHoc | JPanel | Tab Title: NGƯỜI HỌC Layout: BorderLayout |
|  | tblNguoiHoc | JTable | Model: như hình Height: 25 |
| 6 | pnlBangDiem | JPanel | Tab Title: BẢNG ĐIỂM Layout: BorderLayout |

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | pnlKhoaHoc | JPanel | Layout: BorderLayout Layout.Direction:  first |
| 8 | lblKhoaHoc | JLabel | Text: KHÓA HỌC  Layout.Direction: before |
| 9 | cboKhoaHoc | JComboBox | Model: Editable: false  Layout.Direction: center |
| 10 | tblBangDiem | JTable | Layout.Direction: center Model: như hình  Height: 25 |
| 11 | pnlKhoaHoc | JPanel | Tab Title: TỔNG HỢP Layout:  BorderLayout |
| 12 | tblTongHop | JTable | Model: như hình Height: 25  Layout.Direction: center |
| 13 | pnlDoanhThu | JPanel | Tab Title: DOANH  THU Layout: BorderLayout |
| 14 | pnlName | JPanel | Layout: BorderLayout Layout.Direction:  first |
| 15 | lblNam | JLabel | Text: NĂM Layout.Direction: before |
| 16 | cboNam | JComboBox | Model: Editable: false  Layout.Direction: center |

|  |  |  |  |
| --- | --- | --- | --- |
| 17 | tblDoanhThu | JTable | Model: như hình Height: 25  Layout.Direction: center |

### Giới thiệu



Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | GioiThieuJDialog | JDialog | Undecorated: True preferredSize: [512, 505] alwaysOnTop: True  Layout: BorderLayout |

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | lblTitle | JLabel | Icon: logo-small.png Layout.Direction:  first |
| 3 | txtGioiThieu | JTextPane | Text: như hình Layout.Direction: center |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | Editable: False |

### Hướng dẫn sử dụng

Thiết kế giao diện:

Thiết kế các trang web hướng dẫn sử dụng phần mềm. Trong phần viết mã sẽ hướng dẫn cách mở các trang web này khi người dùng nhấp vào mục hướng dẫn trong menu trợ giúp.

# Thực hiện viết mã

## Viết mã tạo CSDL

## Tạo CSDL

CREATE DATABASE Polypro GO

USE Edusys GO

CREATE TABLE NhanVien(

MaNV nvarchar(50) NOT NULL, MatKhau nvarchar(50) NOT NULL,

HoTen nvarchar(50) NOT NULL, VaiTro bit NOT NULL DEFAULT 0, PRIMARY KEY(MaNV)

) GO

CREATE TABLE ChuyenDe(

MaCD nchar(5) NOT NULL, TenCD nvarchar(50) NOT NULL,

HocPhi float NOT NULL DEFAULT 0, ThoiLuong int NOT NULL DEFAULT 30,

Hinh nvarchar(50) NOT NULL DEFAULT 'chuyen-de.png', MoTa nvarchar(255) NOT NULL,

PRIMARY KEY(MaCD), UNIQUE(TenCD),

CHECK(HocPhi >= 0 AND ThoiLuong > 0)

) GO

CREATE TABLE NguoiHoc(

MaNH nchar(7) NOT NULL, HoTen nvarchar(50) NOT NULL, NgaySinh date NOT NULL,

GioiTinh bit NOT NULL DEFAULT 0, DienThoai nvarchar(50) NOT NULL, Email nvarchar(50) NOT NULL, GhiChu nvarchar(max) NULL,

MaNV nvarchar(50) NOT NULL,

NgayDK date NOT NULL DEFAULT getdate(), PRIMARY KEY(MaNH)

) GO

CREATE TABLE KhoaHoc(

MaKH int IDENTITY(1,1) NOT NULL, MaCD nchar(5) NOT NULL,

HocPhi float NOT NULL DEFAULT 0, ThoiLuong int NOT NULL DEFAULT 0, NgayKG date NOT NULL,

GhiChu nvarchar(50) NULL, MaNV nvarchar(50) NOT NULL,

NgayTao date NOT NULL DEFAULT getdate(), PRIMARY KEY(MaKH),

CHECK(HocPhi >= 0 AND ThoiLuong > 0),

FOREIGN KEY (MaCD) REFERENCES ChuyenDe(MaCD) ON UPDATE CASCADE, FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV) ON UPDATE CASCADE

) GO

CREATE TABLE HocVien(

MaHV int IDENTITY(1,1) NOT NULL,

MaKH int NOT NULL,

MaNH nchar(7) NOT NULL, Diem float NOT NULL, PRIMARY KEY(MaHV), UNIQUE(MaKH, MaNH),

FOREIGN KEY (MaKH) REFERENCES KhoaHoc(MaKH) ON DELETE CASCADE,

FOREIGN KEY (MaNH) REFERENCES NguoiHoc(MaNH) ON UPDATE CASCADE

)

## SQL truy vấn và thao tác bảng

* + - 1. ***NhanVien***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO NhanVien (MaNV, MatKhau, HoTen, VaiTro) VALUES (?, ?, ?, ?) |
| Cập nhật theo mã | UPDATE NhanVien SET MatKhau=?, HoTen=?, VaiTro=? WHERE MaNV=? |
| Xóa theo mã | DELETE FROM NhanVien WHERE MaNV=? |
| Truy vấn tất cả | SELECT \* FROM NhanVien |
| Truy vấn theo mã | SELECT \* FROM NhanVien WHERE MaNV=? |

* + - 1. ***ChuyenDe***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO ChuyenDe (MaCD, TenCD, HocPhi, ThoiLuong, Hinh, MoTa) VALUES (?, ?,  ?, ?, ?, ?) |
| Cập nhật theo mã | UPDATE ChuyenDe SET TenCD=?, HocPhi=?, ThoiLuong=?, Hinh=?, MoTa=? WHERE MaCD=? |
| Xóa theo mã | DELETE FROM ChuyenDe WHERE MaCD=? |
| Truy vấn tất cả | SELECT \* FROM ChuyenDe |
| Truy vấn theo mã | SELECT \* FROM ChuyenDe WHERE MaCD=? |

* + - 1. ***NguoiHoc***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO NguoiHoc (MaNH, HoTen, NgaySinh, GioiTinh, DienThoai, Email, GhiChu, MaNV, NgayDK) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?) |
| Cập nhật theo mã | UPDATE NguoiHoc SET HoTen=?, NgaySinh=?, GioiTinh=?, DienThoai=?, Email=?, GhiChu=?, MaNV=?, NgayDK=? WHERE MaNH=? |

|  |  |
| --- | --- |
| Xóa theo mã | DELETE FROM NguoiHoc WHERE MaNH=? |
| Truy vấn tất cả | SELECT \* FROM NguoiHoc |
| Truy vấn theo mã | SELECT \* FROM NguoiHoc WHERE MaNH=? |

* + - 1. ***KhoaHoc***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO KhoaHoc (MaCD, HocPhi, ThoiLuong, NgayKG, GhiChu, MaNV, NgayTao) VALUES (?, ?, ?, ?, ?, ?, ?) |
| Cập nhật theo mã | UPDATE KhoaHoc SET MaCD=?, HocPhi=?, ThoiLuong=?, NgayKG=?, GhiChu=?, MaNV=?, NgayTao=? WHERE MaKH=? |
| Xóa theo mã | DELETE FROM KhoaHoc WHERE MaKH=? |
| Truy vấn tất cả | SELECT \* FROM KhoaHoc |
| Truy vấn theo mã | SELECT \* FROM KhoaHoc WHERE MaKH=? |

* + - 1. ***HocVien***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO HocVien (MaKH, MaNH, Diem) VALUES (?, ?, ?) |
| Cập nhật theo mã | UPDATE HocVien SET MaKH=?, MaNH=?, Diem=? WHERE MaHV=? |
| Xóa theo mã | DELETE FROM HocVien WHERE MaHV=? |
| Truy vấn tất cả | SELECT \* FROM HocVien |
| Truy vấn theo mã | SELECT \* FROM HocVien WHERE MaHV=? |

## Các thủ tục lưu tổng hợp thống kê

* + - 1. ***Số người học từng năm***

CREATE PROC sp\_ThongKeNguoiHoc AS BEGIN

SELECT

YEAR(NgayDK) Nam, COUNT(\*) SoLuong, MIN(NgayDK) DauTien, MAX(NgayDK) CuoiCung

FROM NguoiHoc

GROUP BY YEAR(NgayDK)

END

* + - 1. ***Doanh thu theo chuyên đề***

CREATE PROC sp\_ThongKeDoanhThu(@Year INT) AS BEGIN

SELECT

TenCD ChuyenDe, COUNT(DISTINCT kh.MaKH) SoKH, COUNT(hv.MaHV) SoHV,

SUM(kh.HocPhi) DoanhThu, MIN(kh.HocPhi) ThapNhat, MAX(kh.HocPhi) CaoNhat, AVG(kh.HocPhi) TrungBinh

FROM KhoaHoc kh

JOIN HocVien hv ON kh.MaKH=hv.MaKH JOIN ChuyenDe cd ON cd.MaCD=kh.MaCD

WHERE YEAR(NgayKG) = @Year GROUP BY TenCD

END

* + - 1. ***Học viên theo chuyên đề***

CREATE PROC sp\_ThongKeDiem AS BEGIN

SELECT

TenCD ChuyenDe, COUNT(MaHV) SoHV,

MIN(Diem) ThapNhat, MAX(Diem) CaoNhat, AVG(Diem) TrungBinh

FROM KhoaHoc kh

JOIN HocVien hv ON kh.MaKH=hv.MaKH JOIN ChuyenDe cd ON cd.MaCD=kh.MaCD

GROUP BY TenCD

END

* + - 1. ***Bảng điểm***

CREATE PROC sp\_BangDiem(@MaKH INT) AS BEGIN

SELECT

nh.MaNH, nh.HoTen, hv.Diem

FROM HocVien hv

JOIN NguoiHoc nh ON nh.MaNH=hv.MaNH WHERE hv.MaKH = @MaKH

ORDER BY hv.Diem DESC

END

* 1. **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 gọnhơn, dễhiểuhơn. Trongphần này chúng ta xây dựng 2 lớplà XDatevà Jdbcnhưsa

### Lớp tiện ích XDate

Chứa các phương thức chuyển đổi thời gian và chuỗi giúp lập trình sau này khi xử lý thời gian dễ dàng hơn.

package com.edusys.utils;

import java.text.ParseException; import java.text.SimpleDateFormat; import java.util.Date;

/\*\*

\*

\* @author balis

\*/

public class XDate {

public static SimpleDateFormat formatter = new SimpleDateFormat();

public static Date toDate(String date, String pattern) { try {

formatter.applyPattern(pattern); return formatter.parse(date);

} catch (ParseException e) {

throw new RuntimeException(e);

}

}

public static String toString(Date date, String pattern) { formatter.applyPattern(pattern);

return formatter.format(date);

}

public static Date addDays(Date date, long days) { date.setTime(date.getTime() + days \* 24 \* 60 \* 60 \* 1000); return date;

}

### Lớp tiện ích XJDBC

Chứa các phương thức thực hiện các câu lệnh thao tác và truy vấn được gọn nhẹ hơn.

package com.edusys.utils; import java.sql.\*;

/\*\*

\*

\* @author balis

\*/

public class XJdbc {

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

"jdbc:sqlserver://localhost:1433;databaseName=EDUSYS\_ASSM"; public static String username = "sa";

public static String password = "songlong";

static {

try {

Class.forName(driver);

} catch (ClassNotFoundException e) { throw new RuntimeException(e);

}

}

public static PreparedStatement preparedStatement(String sql, Object... args) throws SQLException {

Connection conn = DriverManager.getConnection(dburl, username, password); PreparedStatement pstmt = null;

if (sql.trim().startsWith("{")) {

pstmt = conn.prepareCall(sql); //proc

} else {

pstmt = conn.prepareStatement(sql); //SQL

}

for (int i = 0; i < args.length; i++) { pstmt.setObject(i + 1, args[i]);

}

return pstmt;

}

public static ResultSet executeQuery(String sql, Object... args) { try {

PreparedStatement pstmt = preparedStatement(sql, args); try {

return pstmt.executeQuery();

} finally {

}

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

public static void executeUpdate(String sql, Object... args) { try {

PreparedStatement pstmt = preparedStatement(sql, args); try {

pstmt.executeUpdate();

} finally {

pstmt.getConnection().close();

}

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

}

### Lớp tiện ích XImage

package com.edusys.utils;

import java.awt.Image; import java.io.File;

import java.io.IOException; import java.net.URL;

import java.nio.file.Files; import java.nio.file.Path; import java.nio.file.Paths;

import java.nio.file.StandardCopyOption; import javax.swing.ImageIcon;

/\*\*

\*

\* @author balis

\*/

public class XImage {

public static ImageIcon getAppIcon() {

URL url = XImage.class.getResource("/com/edusys/icons/fptf4.jpg"); return new ImageIcon(url);

}

public static void save(File src) {

File dir = new File("logos", src.getName()); if (!dir.exists()) {

dir.mkdirs();

}

try {

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

Path destination = Paths.get(dir.getAbsolutePath()); Files.copy(source, destination, StandardCopyOption.REPLACE\_EXISTING);

} catch (IOException e) {

throw new RuntimeException(e);

}

}

public static ImageIcon read(String filename) { File path = new File("logos", filename); return new ImageIcon(new

ImageIcon(path.getAbsolutePath()).getImage().getScaledInstance(180, 180, Image.SCALE\_DEFAULT));

}

}

### Lớp tiện ích MsgBox

package com.edusys.utils;

import java.awt.Component; import javax.swing.JOptionPane;

/\*\*

\*

\* @author balis

\*/

public class MsgBox {

public static void alert(Component parent, String message) { JOptionPane.showMessageDialog(parent, message, "EduSys",

JOptionPane.INFORMATION\_MESSAGE);

}

public static boolean confirm(Component parent, String message) {

int result = JOptionPane.showConfirmDialog(parent, message, "EduSys", JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE);

return result == JOptionPane.YES\_OPTION;

public static String prompt(Component parent, String message) { return JOptionPane.showInputDialog(parent, message, "EduSys",

JOptionPane.INFORMATION\_MESSAGE);

}

}

## 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.

### NhanVien

public class NhanVien {

private String maNV; private String matKhau; private String hoTen;

private boolean vaiTro = false;

@Override

public String toString() { return this.hoTen;

}

public String getMaNV() { return maNV;

}

public void setMaNV(String maNV) { this.maNV = maNV;

}

public String getMatKhau() { return matKhau;

}

public void setMatKhau(String matKhau) { this.matKhau = matKhau;

}

public String getHoTen() { return hoTen;

}

public void setHoTen(String hoTen) { this.hoTen = hoTen;

}

public boolean isVaiTro() { return vaiTro;

public void setVaiTro(boolean vaiTro) { this.vaiTro = vaiTro;

}

}

### ChuyenDe

public class ChuyenDe {

private String maCD; private String tenCD; private double hocPhi; private int thoiLuong; private String hinh; private String moTa;

@Override

public String toString() { return this.tenCD;

}

@Override

public boolean equals(Object obj) { ChuyenDe other = (ChuyenDe) obj;

return other.getMaCD().equals(this.getMaCD());

}

public String getMaCD() { return maCD;

}

public void setMaCD(String maCD) { this.maCD = maCD;

}

public String getTenCD() { return tenCD;

}

public void setTenCD(String tenCD) { this.tenCD = tenCD;

}

public double getHocPhi() { return hocPhi;

}

public void setHocPhi(double hocPhi) { this.hocPhi = hocPhi;

}

public int getThoiLuong() { return thoiLuong;

}

public void setThoiLuong(int thoiLuong) { this.thoiLuong = thoiLuong;

}

public String getHinh() { return hinh;

}

public void setHinh(String hinh) { this.hinh = hinh;

}

public String getMoTa() { return moTa;

}

public void setMoTa(String moTa) { this.moTa = moTa;

}

}

### NguoiHoc

public class NguoiHoc {

private String maNH; private String hoTen; private Date ngaySinh; private boolean gioiTinh; private String dienThoai; private String email; private String ghiChu; private String maNV;

private Date ngayDK = new Date();

@Override

public String toString() { return this.hoTen;

}

public String getMaNH() { return maNH;

}

public void setMaNH(String maNH) { this.maNH = maNH;

}

public String getHoTen() { return hoTen;

}

public void setHoTen(String hoTen) {

this.hoTen = hoTen;

}

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 getDienThoai() { return dienThoai;

}

public void setDienThoai(String dienThoai) { this.dienThoai = dienThoai;

}

public String getEmail() { return email;

}

public void setEmail(String email) { this.email = email;

}

public String getGhiChu() { return ghiChu;

}

public void setGhiChu(String ghiChu) { this.ghiChu = ghiChu;

}

public String getMaNV() { return maNV;

}

public void setMaNV(String maNV) { this.maNV = maNV;

}

public Date getNgayDK() { return ngayDK;

}

public void setNgayDK(Date ngayDK) { this.ngayDK = ngayDK;

}

}

### KhoaHoc

public class KhoaHoc {

private int maKH; private String maCD; private double hocPhi; private int thoiLuong; private Date ngayKG; private String ghiChu;

private String maNV;

private Date ngayTao = new Date();

@Override

public String toString() {

return this.maCD + " (" + this.ngayKG + ")";

}

public int getMaKH() { return maKH;

}

public void setMaKH(int maKH) { this.maKH = maKH;

}

public String getMaCD() { return maCD;

}

public void setMaCD(String maCD) { this.maCD = maCD;

}

public double getHocPhi() { return hocPhi;

}

public void setHocPhi(double hocPhi) { this.hocPhi = hocPhi;

}

public int getThoiLuong() { return thoiLuong;

}

public void setThoiLuong(int thoiLuong) { this.thoiLuong = thoiLuong;

}

public Date getNgayKG() { return ngayKG;

}

public void setNgayKG(Date ngayKG) { this.ngayKG = ngayKG;

}

public String getGhiChu() { return ghiChu;

}

public void setGhiChu(String ghiChu) { this.ghiChu = ghiChu;

}

public String getMaNV() { return maNV;

}

public void setMaNV(String maNV) { this.maNV = maNV;

}

public Date getNgayTao() { return ngayTao;

}

public void setNgayTao(Date ngayTao) { this.ngayTao = ngayTao;

}

}

### HocVien

public class HocVien {

private int maHV; private int maKH; private String maNH;

private double diem = -1.0;

## DAO Class - Các lớp truy xuất dữ liệu

DAO (Data Access Object) là các lớp giúp giúp làm việc với các bảng trong CSDL. Mỗi lớp được định nghĩa một cách đầy đủ nhất về các thao tác và truy vấn dữ liệu sẽ được sử dụng trong ứng dụng. DAO sau đó sẽ được sử dụng trong các điều khiển sự kiện của giao diện để làm việc với CSDL một cách thuận tiện.

Trong phạm vi ứng dụng này chúng ta cần xây dựng mỗi DAO chứa tối thiểu các hành động sau

import java.util.List;

/\*\*

\*

* @author balis
* @param <EntityType>
* @param <KeyType>

\*/

abstract public class EduSysDAO<EntityType, KeyType> { abstract public void insert(EntityType entity); abstract public void update(EntityType entity); abstract public void delete(KeyType id);

abstract public EntityType selectById(KeyType id); abstract public List<EntityType> selectAll();

abstract protected List<EntityType> selectBySQL(String sql, Object... args);

}

Sau đây là các lớp DAO cụ thể

### NhanVienDAO

public class NhanVienDAO extends EduSysDAO<NhanVien, String> {

String INSERT\_SQL = "INSERT INTO NhanVien(MaNV, MatKhau, HoTen, VaiTro) VALUES(?,?,?,?)";

String UPDATE\_SQL = "UPDATE NhanVien SET MatKhau=?, HoTen=?, VaiTro=? WHERE MaNV=?";

String DELETE\_SQL = "DELETE FROM NhanVien WHERE MaNV=?";

String SELECT\_ALL\_SQL = "SELECT \* FROM NhanVien";

String SELECT\_BY\_ID\_SQL = "SELECT \* FROM NhanVien WHERE MaNV=?";

@Override

public void insert(NhanVien entity) { XJdbc.executeUpdate(INSERT\_SQL,

entity.getMaNV(), entity.getMatKhau(), entity.getHoTen(), entity.isVaiTro());

}

@Override

public void update(NhanVien entity) { XJdbc.executeUpdate(UPDATE\_SQL,

entity.getMatKhau(), entity.getHoTen(), entity.isVaiTro(), entity.getMaNV());

}

@Override

public void delete(String id) { XJdbc.executeUpdate(DELETE\_SQL, id);

}

@Override

public NhanVien selectById(String id) {

List<NhanVien> list = this.selectBySQL(SELECT\_BY\_ID\_SQL, id); if (list.isEmpty()) {

return null;

}

return list.get(0);

}

@Override

public List<NhanVien> selectAll() {

return this.selectBySQL(SELECT\_ALL\_SQL);

}

@Override

protected List<NhanVien> selectBySQL(String sql, Object... args) { List<NhanVien> list = new ArrayList<>();

try {

ResultSet rs = XJdbc.executeQuery(sql, args); while (rs.next()) {

NhanVien entity = new NhanVien(); entity.setMaNV(rs.getString("MaNV")); entity.setHoTen(rs.getString("HoTen")); entity.setMatKhau(rs.getString("MatKhau")); entity.setVaiTro(rs.getBoolean("VaiTro")); list.add(entity);

}

rs.getStatement().getConnection().close(); return list;

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

}

### ChuyenDeDAO

public class ChuyenDeDAO extends EduSysDAO<ChuyenDe, String> {

String INSERT\_SQL = "INSERT INTO ChuyenDe (MaCD, TenCD, HocPhi, ThoiLuong, Hinh, MoTa) VALUES (?, ?, ?, ?, ?, ?)";

String UPDATE\_SQL = "UPDATE ChuyenDe SET TenCD=?, HocPhi=?, ThoiLuong=?, Hinh=?, MoTa=? WHERE MaCD=?";

String DELETE\_SQL = "DELETE FROM ChuyenDe WHERE MaCD=?";

String SELECT\_ALL\_SQL = "SELECT \* FROM ChuyenDe";

String SELECT\_BY\_ID\_SQL = "SELECT \* FROM ChuyenDe WHERE MaCD=?";

@Override

public void insert(ChuyenDe entity) { XJdbc.executeUpdate(INSERT\_SQL,

entity.getMaCD(), entity.getTenCD(), entity.getHocPhi(), entity.getThoiLuong(), entity.getHinh(), entity.getMoTa());

}

@Override

public void update(ChuyenDe entity) { XJdbc.executeUpdate(UPDATE\_SQL,

entity.getTenCD(), entity.getHocPhi(), entity.getThoiLuong(), entity.getHinh(), entity.getMoTa(), entity.getMaCD());

}

@Override

public void delete(String id) { XJdbc.executeUpdate(DELETE\_SQL, id);

}

@Override

public ChuyenDe selectById(String id) {

List<ChuyenDe> list = this.selectBySQL(SELECT\_BY\_ID\_SQL, id); if (list.isEmpty()) {

return null;

}

return list.get(0);

}

@Override

public List<ChuyenDe> selectAll() {

return this.selectBySQL(SELECT\_ALL\_SQL);

}

@Override

protected List<ChuyenDe> selectBySQL(String sql, Object... args) { List<ChuyenDe> list = new ArrayList<>();

try {

ResultSet rs = XJdbc.executeQuery(sql, args); while (rs.next()) {

ChuyenDe entity = new ChuyenDe(); entity.setMaCD(rs.getString("MaCD")); entity.setTenCD(rs.getString("TenCD")); entity.setHocPhi(rs.getDouble("HocPhi")); entity.setThoiLuong(rs.getInt("ThoiLuong")); entity.setHinh(rs.getString("Hinh")); entity.setMoTa(rs.getString("MoTa")); list.add(entity);

}

rs.getStatement().getConnection().close(); return list;

} catch (Exception e) {

throw new RuntimeException(e);

}

}

### NguoiHocDAO

public class NguoiHocDAO extends EduSysDAO<NguoiHoc, String> {

String INSERT\_SQL = "INSERT INTO NguoiHoc (MaNH, HoTen, NgaySinh, GioiTinh, DienThoai, Email, GhiChu, MaNV) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

String UPDATE\_SQL = "UPDATE NguoiHoc SET HoTen=?, NgaySinh=?, GioiTinh=?, DienThoai=?, Email=?, GhiChu=?,MaNV=? WHERE MaNH=?";

String DELETE\_SQL = "DELETE FROM NguoiHoc WHERE MaNH=?";

String SELECT\_ALL\_SQL = "SELECT \* FROM NguoiHoc";

String SELECT\_BY\_ID\_SQL = "SELECT \* FROM NguoiHoc WHERE MaNH=?";

@Override

public void insert(NguoiHoc entity) { XJdbc.executeUpdate(INSERT\_SQL,

entity.getMaNH(), entity.getHoTen(), entity.getNgaySinh(), entity.isGioiTinh(), entity.getDienThoai(), entity.getEmail(), entity.getGhiChu(), entity.getMaNV());

}

@Override

public void update(NguoiHoc entity) { XJdbc.executeUpdate(UPDATE\_SQL,

entity.getHoTen(), entity.getNgaySinh(), entity.isGioiTinh(), entity.getDienThoai(), entity.getEmail(), entity.getGhiChu(), entity.getMaNV(), entity.getMaNH());

}

@Override

public void delete(String id) { XJdbc.executeUpdate(DELETE\_SQL, id);

}

@Override

public NguoiHoc selectById(String id) {

List<NguoiHoc> list = this.selectBySQL(SELECT\_BY\_ID\_SQL, id); if (list.isEmpty()) {

return null;

}

return list.get(0);

}

@Override

public List<NguoiHoc> selectAll() {

return this.selectBySQL(SELECT\_ALL\_SQL);

}

@Override

protected List<NguoiHoc> selectBySQL(String sql, Object... args) { List<NguoiHoc> list = new ArrayList<>();

try {

ResultSet rs = XJdbc.executeQuery(sql, args); while (rs.next()) {

NguoiHoc entity = new NguoiHoc(); entity.setMaNH(rs.getString("MaNH")); entity.setHoTen(rs.getString("HoTen")); entity.setNgaySinh(rs.getDate("NgaySinh")); entity.setGioiTinh(rs.getBoolean("GioiTinh")); entity.setDienThoai(rs.getString("DienThoai")); entity.setEmail(rs.getString("Email")); entity.setGhiChu(rs.getString("GhiChu")); entity.setMaNV(rs.getString("MaNV")); list.add(entity);

}

rs.getStatement().getConnection().close(); return list;

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

public List<NguoiHoc> selectByKeyword(String keyword) { String SQL = "SELECT \* FROM NguoiHoc WHERE HoTen LIKE ?"; return this.selectBySQL(SQL, "%" + keyword + "%");

}

public List<NguoiHoc> selectNotInCourse(int makh, String keyword) { String SQL = "SELECT \* FROM NguoiHoc WHERE HoTen LIKE ? AND "

+ "MaNH NOT IN(SELECT MaNH FROM HocVien WHERE MaKH = ?)";

return this.selectBySQL(SQL, "%" + keyword + "%", makh);

}

}

### KhoaHocDAO

public class KhoaHocDAO extends EduSysDAO<KhoaHoc, Integer> {

String INSERT\_SQL = "INSERT INTO KhoaHoc (MaCD, HocPhi, ThoiLuong, NgayKG, GhiChu, MaNV) VALUES (?, ?, ?, ?, ?, ?)";

String UPDATE\_SQL = "UPDATE KhoaHoc SET MaCD=?, HocPhi=?, ThoiLuong=?, NgayKG=?, GhiChu=?, MaNV=? WHERE MaKH=?";

String DELETE\_SQL = "DELETE FROM KhoaHoc WHERE MaKH=?";

String SELECT\_ALL\_SQL = "SELECT \* FROM KhoaHoc";

String SELECT\_BY\_ID\_SQL = "SELECT \* FROM KhoaHoc WHERE MaKH=?";

@Override

public void insert(KhoaHoc entity) { XJdbc.executeUpdate(INSERT\_SQL,

entity.getMaCD(), entity.getHocPhi(), entity.getThoiLuong(), entity.getNgayKG(), entity.getGhiChu(), entity.getMaNV());

}

@Override

public void update(KhoaHoc entity) { XJdbc.executeUpdate(UPDATE\_SQL,

entity.getMaCD(), entity.getHocPhi(), entity.getThoiLuong(), entity.getNgayKG(), entity.getGhiChu(), entity.getMaNV(), entity.getMaKH());

}

@Override

public void delete(Integer id) { XJdbc.executeUpdate(DELETE\_SQL, id);

}

@Override

public KhoaHoc selectById(Integer id) {

List<KhoaHoc> list = this.selectBySQL(SELECT\_BY\_ID\_SQL, id); if (list.isEmpty()) {

return null;

}

return list.get(0);

}

@Override

public List<KhoaHoc> selectAll() {

return this.selectBySQL(SELECT\_ALL\_SQL);

}

@Override

protected List<KhoaHoc> selectBySQL(String sql, Object... args) { List<KhoaHoc> list = new ArrayList<>();

try {

ResultSet rs = XJdbc.executeQuery(sql, args); while (rs.next()) {

KhoaHoc entity = new KhoaHoc(); entity.setMaKH(rs.getInt("MaKH")); entity.setMaCD(rs.getString("MaCD")); entity.setHocPhi(rs.getDouble("HocPhi")); entity.setThoiLuong(rs.getInt("ThoiLuong")); entity.setNgayKG(rs.getDate("NgayKG")); entity.setGhiChu(rs.getString("GhiChu")); entity.setMaNV(rs.getString("MaNV")); entity.setNgayTao(rs.getDate("NgayTao")); list.add(entity);

}

rs.getStatement().getConnection().close(); return list;

} catch (Exception e) {

throw new RuntimeException(e);

}

}

public List<KhoaHoc> selectByChuyenDe(String macd) { String SQL = "SELECT \* FROM KhoaHoc WHERE MaCD = ?";

return this.selectBySQL(SQL, macd);

}

public List<Integer> selectYears() {

String SQL = "SELECT DISTINCT year(NgayKG) Year FROM KhoaHoc ORDER BY Year DESC";

List<Integer> list = new ArrayList<>(); try {

ResultSet rs = XJdbc.executeQuery(SQL); while (rs.next()) {

list.add(rs.getInt(1));

}

rs.getStatement().getConnection().close(); return list;

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

}

### HocVienDAO

public class HocVienDAO extends EduSysDAO<HocVien, Integer> {

String INSERT\_SQL = "INSERT INTO HocVien(MaKH, MaNH, Diem) VALUES(?, ?, ?)"; String UPDATE\_SQL = "UPDATE HocVien SET MaKH=?, MaNH=?, Diem=? WHERE MaHV=?"; String DELETE\_SQL = "DELETE FROM HocVien WHERE MaHV=?";

String SELECT\_ALL\_SQL = "SELECT \* FROM HocVien";

String SELECT\_BY\_ID\_SQL = "SELECT \* FROM HocVien WHERE MaHV=?";

@Override

public void insert(HocVien entity) { XJdbc.executeUpdate(INSERT\_SQL,

entity.getMaKH(), entity.getMaNH(), entity.getDiem());

}

@Override

public void update(HocVien entity) { XJdbc.executeUpdate(UPDATE\_SQL,

entity.getMaKH(), entity.getMaNH(), entity.getDiem(), entity.getMaHV());

}

@Override

public void delete(Integer id) { XJdbc.executeUpdate(DELETE\_SQL, id);

}

@Override

public HocVien selectById(Integer id) {

List<HocVien> list = this.selectBySQL(SELECT\_BY\_ID\_SQL, id); if (list.isEmpty()) {

return null;

}

return list.get(0);

}

@Override

public List<HocVien> selectAll() {

return this.selectBySQL(SELECT\_ALL\_SQL);

}

@Override

protected List<HocVien> selectBySQL(String sql, Object... args) { List<HocVien> list = new ArrayList<>();

try {

ResultSet rs = XJdbc.executeQuery(sql, args); while (rs.next()) {

HocVien entity = new HocVien(); entity.setMaHV(rs.getInt("MaHV")); entity.setMaKH(rs.getInt("MaKH")); entity.setMaNH(rs.getString("MaNH")); entity.setDiem(rs.getDouble("Diem")); list.add(entity);

}

rs.getStatement().getConnection().close(); return list;

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

public List<HocVien> selectByKhoaHoc(int makh) {

String SQL = "SELECT \* FROM HocVien WHERE MaKH = ?";

return this.selectBySQL(SQL, makh);

}

}

### ThongKeDAO

Lớp DAO này chứa các phương thức gọi đến các thủ tục lưu được xây dựng trước đây để thức hiện các chức năngtổng hợp thống kê trong ứng dụng.

public class ThongKeDAO {

private List<Object[]> getListOfArray(String sql, String[] cols, Object... args) {

try {

List<Object[]> list = new ArrayList<>(); ResultSet rs = XJdbc.executeQuery(sql, args); while (rs.next()) {

Object[] vals = new Object[cols.length]; for (int i = 0; i < cols.length; i++) {

vals[i] = rs.getObject(cols[i]);

}

list.add(vals);

}

rs.getStatement().getConnection().close(); return list;

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

public List<Object[]> getBangDiem(Integer maKH) { String sql = "{CALL sp\_BangDiem(?)}"; String[] cols = {"MaNH", "HoTen", "Diem"}; return this.getListOfArray(sql, cols, maKH);

}

public List<Object[]> getLuongNguoiHoc() { String sql = "{CALL sp\_LuongNguoiHoc}";

String[] cols = {"Nam", "SoLuong", "DauTien", "CuoiCung"}; return this.getListOfArray(sql, cols);

}

public List<Object[]> getDiemChuyenDe() { String sql = "{CALL sp\_DiemChuyenDe}";

String[] cols = {"ChuyenDe", "SoHV", "ThapNhat", "CaoNhat", "TrungBinh"}; return this.getListOfArray(sql, cols);

}

public List<Object[]> getDoanhThu(int nam) { String sql = "{CALL sp\_DoanhThu(?)}";

String[] cols = {"ChuyenDe", "SoKH", "SoHV", "DoanhThu", "ThapNhat", "CaoNhat", "TrungBinh"};

return this.getListOfArray(sql, cols, nam);

}

## Viết mã cho ứngdụng

## Các lớp tiệních

Lập trình điều khiển hành vi tương tác của người dùng lên các giao diện của ứng dụng. Trước khi lập trình, chúng ta cần thực hiện 2 điều sau đây.

### DialogHelper

Lớp DialogHelper chứa các phương thức đơn giản hóa JOptionPane

public class MsgBox {

public static void alert(Component parent, String message) { JOptionPane.showMessageDialog(parent, message, "EduSys",

JOptionPane.INFORMATION\_MESSAGE);

}

public static boolean confirm(Component parent, String message) {

int result = JOptionPane.showConfirmDialog(parent, message, "EduSys", JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE);

return result == JOptionPane.YES\_OPTION;

}

public static String prompt(Component parent, String message) { return JOptionPane.showInputDialog(parent, message, "EduSys",

JOptionPane.INFORMATION\_MESSAGE);

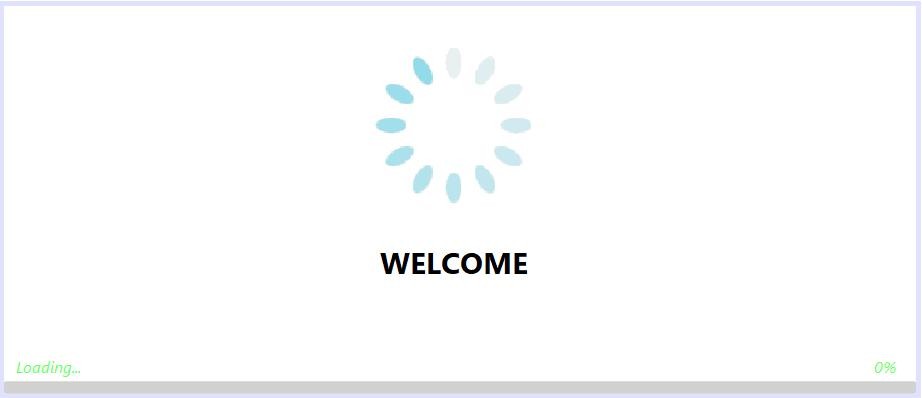
}

}

## Viết mã cho các chức năng

Trong phần này chúng ta sẽ tiến hành viết mã cho các giao diện đã thiết kế trước đây. Cụ thể chúng ta sẽ lần lược viết mã thực hiện các chức năng cho các giao diện sau đây.

### Màn hình chào(ChaoJDialog)



#### HOẠT ĐỘNG

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
| runLoadingBar() | Init |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

private void runLoadingBar() { class Loading extends Thread {

@Override

public void run() {

for (int i = 0; i <= 100; i++) { try {

Thread.sleep(20); lblLoadingValue.setText(i + "%"); progressBar.setValue(i);

if (i == 100) {

dispose();

new MainJFrame().setVisible(true);

}

} catch (InterruptedException ex) {

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

}

}

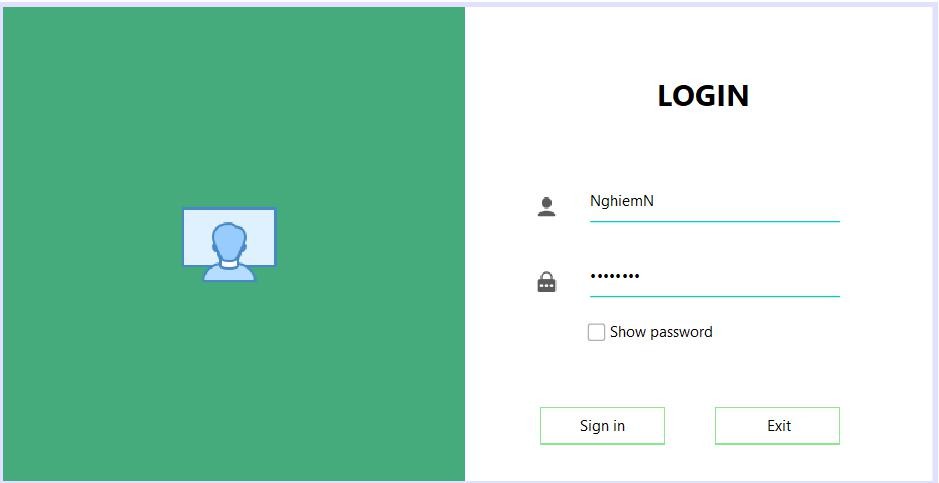
}

}

Loading l = new Loading(); l.start();

}

### Form đăng nhập(DangNhapJFrame)



#### MÔ TẢ HOẠT ĐỘNG

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
| Constructor | Init() |

|  |  |
| --- | --- |
| btnDangNhap | Login() |
| btnKetThuc | Exit() |

HƯỚNG DẪN CÀI ĐẶT MÃ

Bổ sung mã sau vào cuối lớp DangNhapJFrame.

void login() {

String mamv = txtUser.getText();

String password = new String(txtPassword.getPassword()); NhanVien nhanVien = dao.selectById(mamv);

if (nhanVien == null) {

MsgBox.alert(this, "Wrong username!");

} else if (!password.equals(nhanVien.getMatKhau())) { MsgBox.alert(this, "Wrong password!");

} else {

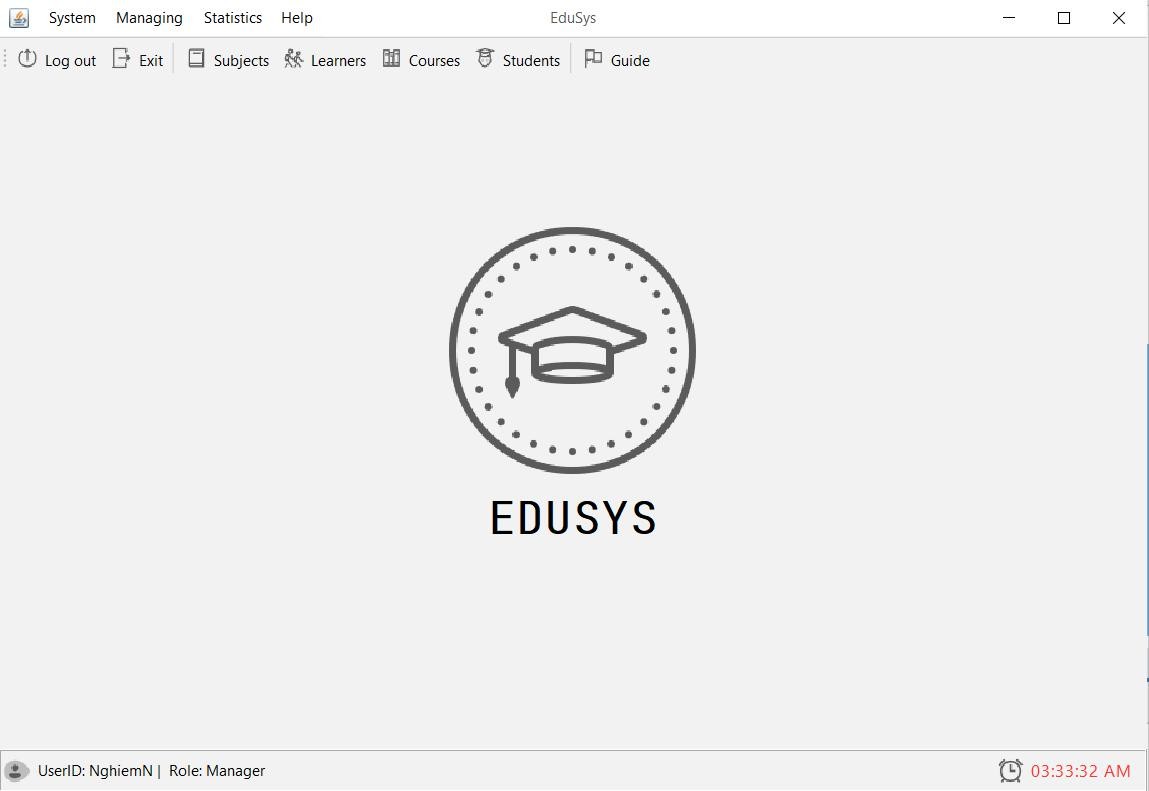
Auth.user = nhanVien; this.dispose();

new GreetJDialog(this, rootPaneCheckingEnabled).setVisible(true);

}

}

### Giao diện chính(MainJFrame)



#### MÔ TẢ HOẠT ĐỘNG

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
| Constructor | Init() |
| Đăng nhập | Login() |
| Đăng xuất | Logoff() |
| Kết thuc | Exit() |
| Chuyên đề | openChuyenDe() |
| Người học | openNguoiHoc() |
| Học viên | openHocVien() |
| Khóa học | openKhoaHoc() |

|  |  |
| --- | --- |
| Nhân viên | openNhanVien() |

|  |  |
| --- | --- |
| Thống kê người học từng năm | openThongKe(1) |
| Bảng điểm khóa | openThongKe(2) |
| Điểm từng khóa | openThongKe(3) |
| Doanh thư từng chuyên đề | openThongKe(4) |
| Hướng dẫn | openAbout() |
| Giới thiệu | openWebsite() |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

void startClock() {

class TimeClock extends Thread {

@Override

public void run() { while (true) {

lblClock.setText(new SimpleDateFormat("hh:mm:ss a").format(Calendar.getInstance().getTime()));

}

}

}

TimeClock timeClock = new TimeClock(); timeClock.start();

}

void displayUserInfo() {

String userID = Auth.user.getMaNV();

String role = Auth.user.isVaiTro() ? "Manager" : "Employee"; lblUserInfo.setText("UserID: " + userID + " | Role: " + role);

}

void openChangePassword() { if (Auth.isLogin()) {

new ChangePasswordJDialog(this, rootPaneCheckingEnabled).setVisible(true);

} else {

MsgBox.alert(this, "Please login!");

}

}

void LogOut() { Auth.clear(); this.dispose();

new LoginJDialog(this, rootPaneCheckingEnabled).setVisible(true);

}

void Exit() {

if (MsgBox.confirm(this, "Do you want to end this session?")) { System.exit(0);

}

}

void openSubjects() {

new SubjectJDialog(this, rootPaneCheckingEnabled).setVisible(true);

}

void openCourses() {

new CourseJDialog(this, rootPaneCheckingEnabled).setVisible(true);

}

void openLearners() {

new LearnerJDialog(this, rootPaneCheckingEnabled).setVisible(true);

}

void openStudents() {

new StudentJDialog(this, rootPaneCheckingEnabled).setVisible(true);

}

void openEmployees() {

new EmployeeJDialog(this, rootPaneCheckingEnabled).setVisible(true);

}

void openGuide() { try {

Desktop.getDesktop().browse(new URI("https://[www.kansastag.gov/advhtml\_doc\_upload/caplio\_500se\_software\_user\_guid](http://www.kansastag.gov/advhtml_doc_upload/caplio_500se_software_user_guid) e.pdf"));

} catch (IOException | URISyntaxException ex) { Logger.getLogger(MainJFrame.class.getName()).log(Level.SEVERE, null,

ex);

}

}

void openAboutUs() {

new AboutJDialog(this, rootPaneCheckingEnabled).setVisible(true);

}

void openStatistics(int index) { if (Auth.isLogin()) {

if (index == 3 && !Auth.isManager()) {

MsgBox.alert(this, "You're not authorized to view revenue tab!");

} else {

StatisticJDialog statisticJDialog = new StatisticJDialog(this,

true);

statisticJDialog.selectTab(index); statisticJDialog.setVisible(true);

}

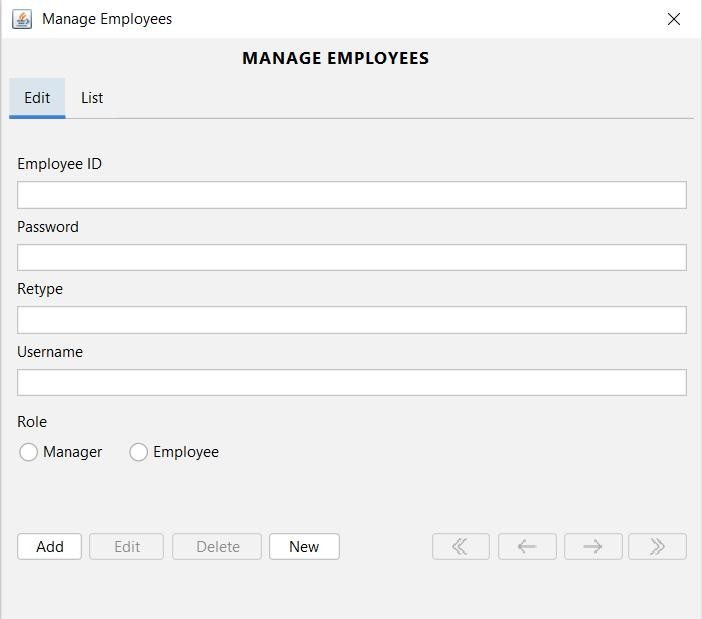
} else {

MsgBox.alert(this, "Please login!");

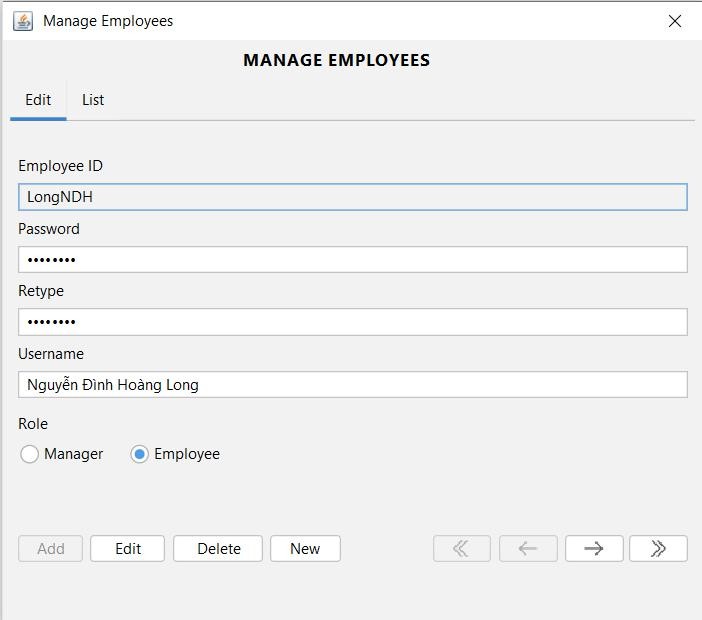
}

}

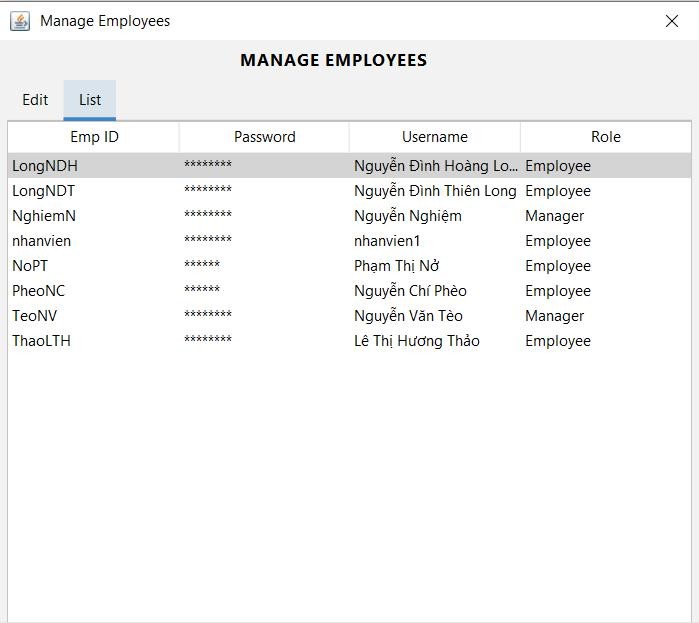
### Quản lý nhân viên(NhanVienJFrame)



Hình: Form ở trạng thái insert



Hình: Form ở trạng thái update



Hình: Bảng chứa danh sách nhân viên

#### MÔ TẢ HOẠT ĐỘNG

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc phải làm** |
| Constructor | Init() |
| windowOpened | this.load(); this.setStatus(true); |

|  |  |
| --- | --- |
| tblGridView.mouseClicked | if(evt.getClickCount() == 2){ this.index =  tblGridView.rowAtPoint(evt.getPoint()); if (this.index >= 0) {  this.edit(); tabs.setSelectedIndex(0);  }  } |

|  |  |
| --- | --- |
| btnInsert | Insert() |
| btnUpdate | Update() |
| btnDelete | Delete() |
| btnClear | Clear() |
| btnFirst | this.index = 0; this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++  ; this.edit(); |
| btnLast | this.index = tblGridView.getRowCount() - 1; this.edit(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

NhanVienDAO dao = new NhanVienDAO(); int row = -1;

/\*\*

\* Creates new form employeeJDialog

\*/

public EmployeeJDialog(java.awt.Frame parent, boolean modal) { super(parent, modal);

initComponents(); this.fillTable(); this.updateStatus(); setTitle("Manage Employees"); setLocationRelativeTo(parent);

}

void insert() {

NhanVien nv = getForm();

String mk2 = new String(txtRetype.getPassword()); if (!mk2.equals(nv.getMatKhau())) {

MsgBox.alert(this, "Retype password didn't match!");

} else {

if (!Auth.isManager() && rdoManager.isSelected()) { MsgBox.alert(this, "Employee can't add new Manager!");

} else {

try {

dao.insert(nv); this.fillTable();

this.clearForm();

MsgBox.alert(this, "Insert successfully");

} catch (Exception e) {

MsgBox.alert(this, "Insert unsucessfully!");

}

}

}

}

void update() {

NhanVien nv = getForm();

String mk2 = new String(txtRetype.getPassword()); if (!mk2.equals(nv.getMatKhau())) {

MsgBox.alert(this, "Retype password didn't match!");

} else {

try {

dao.update(nv); this.fillTable();

MsgBox.alert(this, "Update successfully");

} catch (Exception e) {

MsgBox.alert(this, "Update unsucessfully!");

}

}

}

void delete() {

if (!Auth.isManager()) {

MsgBox.alert(this, "You're not authorized to delete employee!");

} else {

String manv = txtEmpID.getText();

if (manv.equals(Auth.user.getMaNV())) { MsgBox.alert(this, "You can't delete yourself!");

} else if (MsgBox.confirm(this, "Do you want to delete this employee?")) {

try {

dao.delete(manv); this.fillTable(); this.clearForm();

MsgBox.alert(this, "Delete sucessfully!");

} catch (Exception e) {

MsgBox.alert(this, "Delete unsucessfully!");

}

}

}

}

void edit() {

String manv = (String) tblEmployee.getValueAt(this.row, 0); NhanVien nv = dao.selectById(manv);

this.setForm(nv); tabs.setSelectedIndex(0);

this.updateStatus();

}

void clearForm() {

NhanVien nv = new NhanVien(); this.setForm(nv);

this.row = -1; this.updateStatus();

}

void first() { this.row = 0; this.edit();

}

void prev() {

if (this.row > 0) { this.row--;

this.edit();

}

}

void next() {

if (this.row < tblEmployee.getRowCount() - 1) { this.row++;

this.edit();

}

}

void last() {

this.row = tblEmployee.getRowCount() - 1; this.edit();

}

void fillTable() {

DefaultTableModel model = (DefaultTableModel) tblEmployee.getModel(); model.setRowCount(0);

try {

List<NhanVien> list = dao.selectAll(); for (NhanVien nv : list) {

Object[] rows = {nv.getMaNV(), nv.getMatKhau().replaceAll(".", "\*"), nv.getHoTen(), nv.isVaiTro() ? "Manager" : "Employee"};

model.addRow(rows);

}

} catch (Exception e) {

MsgBox.alert(this, "Data Query Error!");

}

}

void setForm(NhanVien nv) { txtEmpID.setText(nv.getMaNV());

txtUsername.setText(nv.getHoTen());

txtPassword.setText(nv.getMatKhau()); txtRetype.setText(nv.getMatKhau()); rdoManager.setSelected(nv.isVaiTro()); rdoEmployee.setSelected(!nv.isVaiTro());

}

private NhanVien getForm() { NhanVien nv = new NhanVien(); nv.setMaNV(txtEmpID.getText());

nv.setHoTen(txtUsername.getText()); nv.setMatKhau(new String(txtPassword.getPassword())); nv.setVaiTro(rdoManager.isSelected());

return nv;

}

void updateStatus() {

boolean edit = (this.row >= 0); boolean first = (this.row == 0);

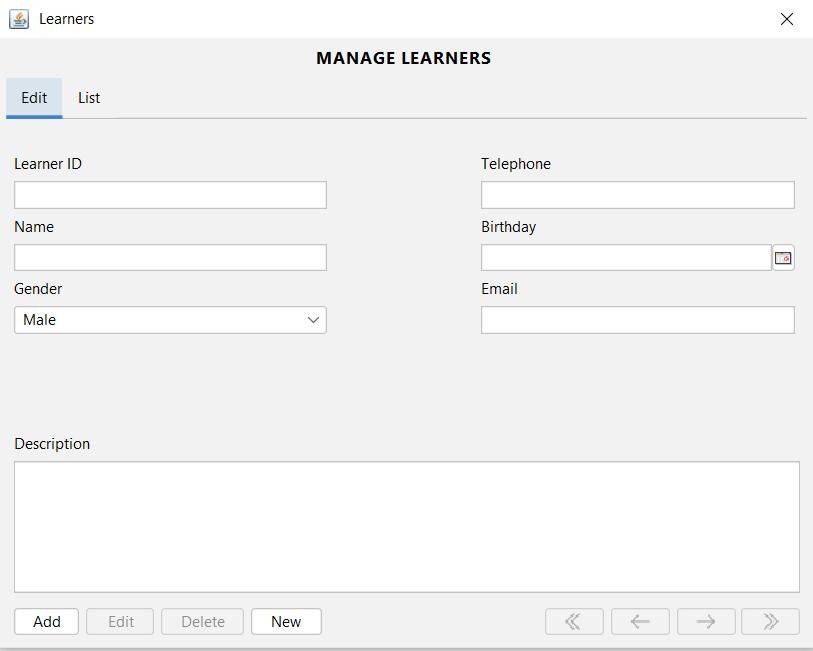
boolean last = (this.row == tblEmployee.getRowCount() - 1);

//Form state txtEmpID.setEditable(!edit); btnAdd.setEnabled(!edit); btnEdit.setEnabled(edit); btnDelete.setEnabled(edit);

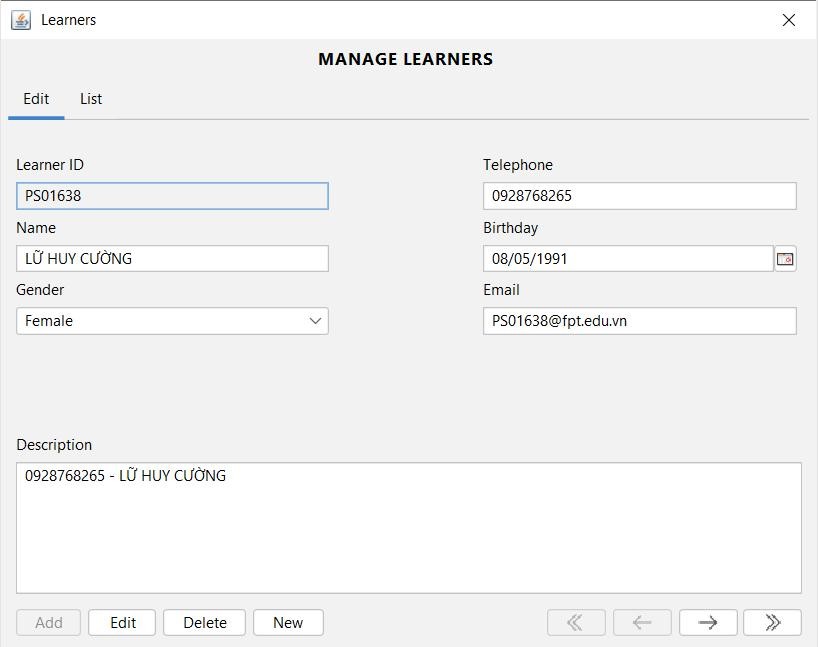
//Directional state btnFirst.setEnabled(edit && !first); btnPrev.setEnabled(edit && !first); btnNext.setEnabled(edit && !last); btnLast.setEnabled(edit && !last);

}

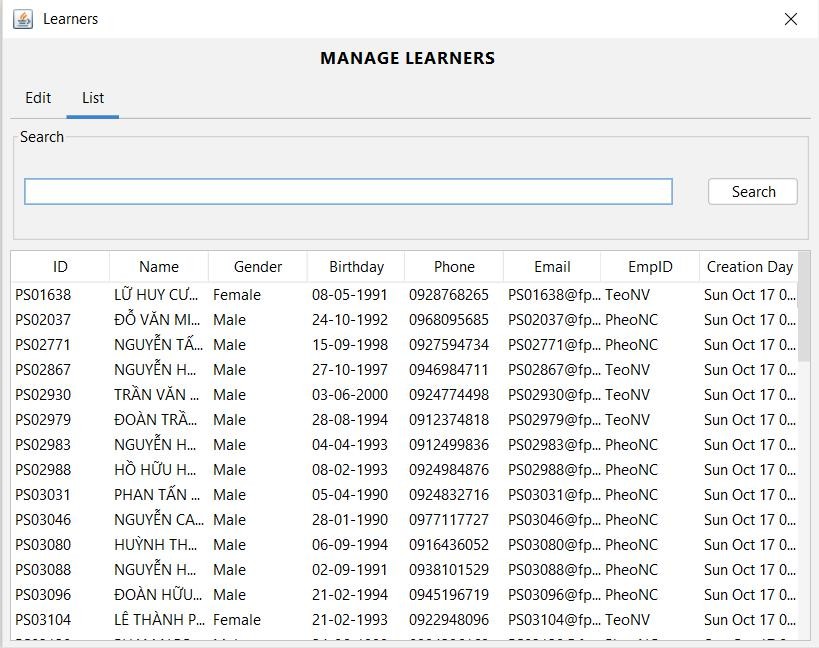
### Quản lý người học(NguoiHocJFrame)



Hình: form trạng thái thêm mới



Hình: form trạng thái cập nhật



Hình: Bảng chứa danh sách người học

#### MÔ TẢ HOẠT ĐỘNG

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc phải làm** |
| Constructor | Init() |
| windowOpened | this.load(); this.setStatus(true); |

|  |  |
| --- | --- |
| tblGridView.mouseClicked | if(evt.getClickCount() == 2){ this.index =  tblGridView.rowAtPoint(evt.getPoint()); if (this.index >= 0) {  this.edit(); tabs.setSelectedIndex(0);  }  } |
| btnInsert | Insert() |

|  |  |
| --- | --- |
| btnUpdate | Update() |
| btnDelete | Delete() |
| btnClear | Clear() |
| btnFirst | this.index = 0; this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++  ; this.edit(); |
| btnLast | this.index = tblGridView.getRowCount() - 1; this.edit(); |
| btnTimKiem | this.load();  this.clear(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

NguoiHocDAO dao = new NguoiHocDAO(); int row = -1;

/\*\*

\* Creates new form LearnerJDialog

\*/

public LearnerJDialog(java.awt.Frame parent, boolean modal) { super(parent, modal);

initComponents(); this.fillTable(); this.updateStatus(); setTitle("Learners"); setLocationRelativeTo(parent);

}

void insert() {

NguoiHoc nh = getForm(); try {

dao.insert(nh); this.fillTable(); this.clearForm();

MsgBox.alert(this, "Insert successfully");

} catch (Exception e) {

MsgBox.alert(this, "Insert unsucessfully!");

}

}

void update() {

NguoiHoc nh = getForm(); try {

dao.update(nh); this.fillTable();

MsgBox.alert(this, "Update successfully");

} catch (Exception e) {

MsgBox.alert(this, "Update unsucessfully!");

}

}

void delete() {

if (!Auth.isManager()) {

MsgBox.alert(this, "You're not authorized to delete learner!");

} else {

String manh = txtLearnerID.getText();

if (MsgBox.confirm(this, "Do you want to delete this employee?")) { try {

dao.delete(manh); this.fillTable(); this.clearForm();

MsgBox.alert(this, "Delete sucessfully!");

} catch (Exception e) {

MsgBox.alert(this, "Delete unsucessfully!");

}

}

}

}

void fillTable() {

DefaultTableModel model = (DefaultTableModel) tblLearner.getModel(); model.setRowCount(0);

try {

String keyword = txtSearch.getText(); List<NguoiHoc> list = dao.selectByKeyword(keyword); for (NguoiHoc nh : list) {

Object[] rows = {nh.getMaNH(), nh.getHoTen(), nh.isGioiTinh() ? "Male" : "Female", XDate.toString(nh.getNgaySinh(), "dd-MM-yyyy"),

nh.getDienThoai(), nh.getEmail(), nh.getMaNV(),

nh.getNgayDK()};

model.addRow(rows);

}

} catch (Exception e) {

}

}

void setForm(NguoiHoc nh) { txtLearnerID.setText(nh.getMaNH()); txtName.setText(nh.getHoTen());

txtBirth.setDate(nh.getNgaySinh());

// txtBirth.setText(XDate.toString(nh.getNgaySinh(), "dd-MM-yyyy")); txtEmail.setText(nh.getEmail()); txtTelephone.setText(nh.getDienThoai()); txtDesc.setText(nh.getGhiChu()); cboGender.setSelectedIndex(nh.isGioiTinh() ? 0 : 1);

}

void clearForm() {

NguoiHoc nh = new NguoiHoc(); nh.setMaNV(Auth.user.getMaNV()); nh.setNgayDK(new Date()); this.setForm(nh);

this.row = -1; this.updateStatus();

}

NguoiHoc getForm() {

NguoiHoc nh = new NguoiHoc(); nh.setMaNH(txtLearnerID.getText()); nh.setHoTen(txtName.getText()); nh.setDienThoai(txtTelephone.getText()); nh.setGhiChu(txtDesc.getText()); nh.setNgayDK(new Date()); nh.setNgaySinh(txtBirth.getDate()); nh.setEmail(txtEmail.getText());

nh.setGioiTinh(cboGender.getSelectedIndex() == 0); nh.setMaNV(Auth.user.getMaNV());

return nh;

}

void edit() {

String manh = (String) tblLearner.getValueAt(this.row, 0); NguoiHoc nh = dao.selectById(manh);

this.setForm(nh); tabs.setSelectedIndex(0); this.updateStatus();

}

void updateStatus() {

boolean edit = (this.row >= 0); boolean first = (this.row == 0);

boolean last = (this.row == tblLearner.getRowCount() - 1);

//Form state txtLearnerID.setEditable(!edit); btnAdd.setEnabled(!edit); btnEdit.setEnabled(edit); btnDelete.setEnabled(edit);

//Directional state btnFirst.setEnabled(edit && !first); btnPrev.setEnabled(edit && !first);

btnNext.setEnabled(edit && !last);

btnLast.setEnabled(edit && !last);

}

void search() { this.fillTable(); this.clearForm(); this.row = -1; this.updateStatus();

}

void first() { this.row = 0; this.edit();

}

void prev() {

if (this.row > 0) { this.row--;

this.edit();

}

}

void next() {

if (this.row < tblLearner.getRowCount() - 1) { this.row++;

this.edit();

}

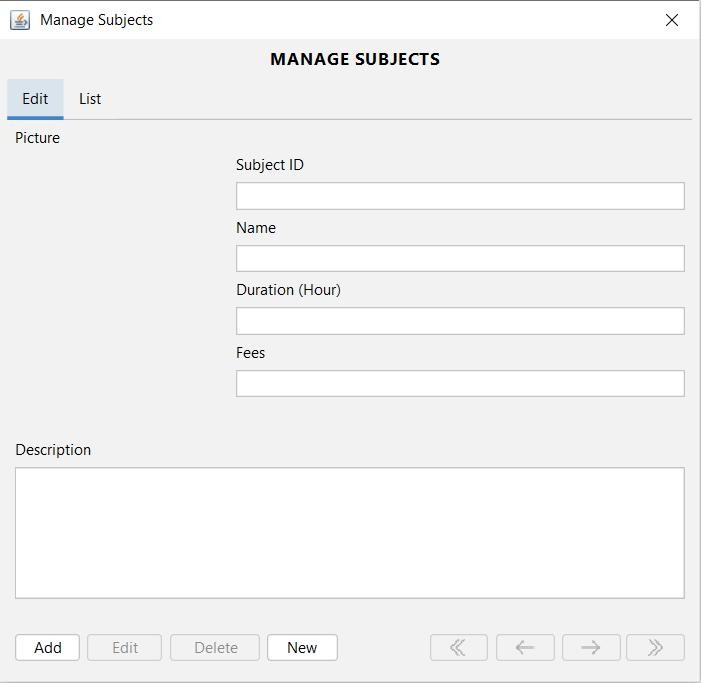
}

void last() {

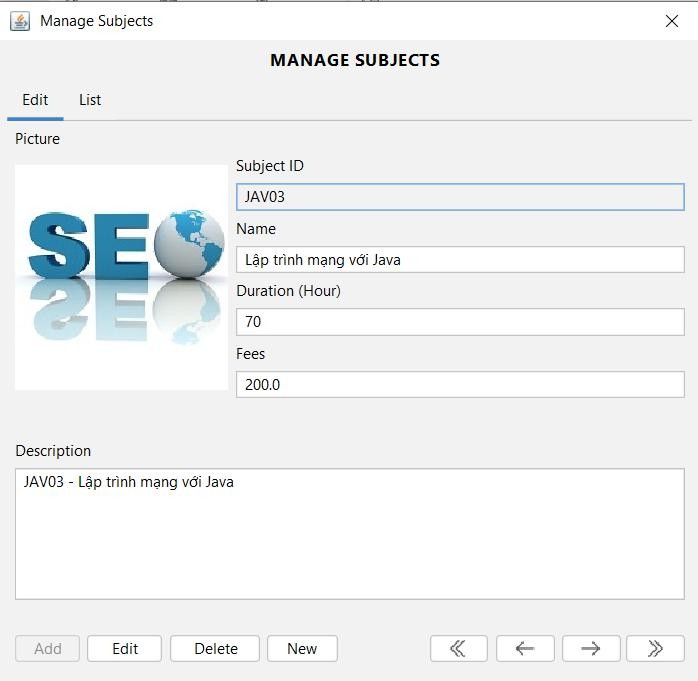
this.row = tblLearner.getRowCount() - 1; this.edit();

}

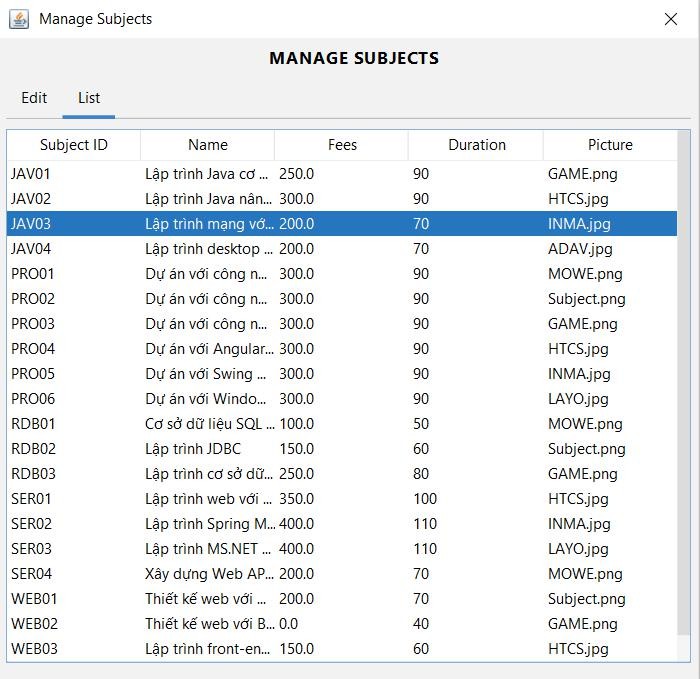
### Quản lý chuyên đề(ChuyenDeJFrame)



Hình: trạng thái insert



Hình: trạng thái update



Hình: bảng dữ liệu

#### MÔ TẢ HOẠT ĐỘNG

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc phải làm** |
| Constructor | Init() |
| windowOpened | this.load(); this.setStatus(true); |

|  |  |
| --- | --- |
| tblGridView.mouseClicked | if(evt.getClickCount() == 2){ this.index =  tblGridView.rowAtPoint(evt.getPoint()); if (this.index >= 0) {  this.edit();  tabs.setSelectedIndex(0); |

|  |  |
| --- | --- |
|  | }  } |
| btnInsert | Insert() |
| btnUpdate | Update() |
| btnDelete | Delete() |
| btnClear | Clear() |
| btnFirst | this.index = 0; this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++  ; this.edit(); |
| btnLast | this.index = tblGridView.getRowCount() - 1; this.edit(); |
| lblHinh | this.selectImage(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

ChuyenDeDAO dao = new ChuyenDeDAO(); JFileChooser fileChooser = new JFileChooser(); int row = -1;

/\*\*

\* Creates new form ManageSubject

\*/

public SubjectJDialog(java.awt.Frame parent, boolean modal) { super(parent, modal);

initComponents(); this.fillTable(); this.updateStatus(); setTitle("Manage Subjects"); setLocationRelativeTo(parent);

}

void insert() {

ChuyenDe cd = getForm(); try {

dao.insert(cd); this.fillTable();

this.clearForm();

MsgBox.alert(this, "Insert Successfully!");

} catch (Exception e) {

MsgBox.alert(this, "Insert Unsuccessfully!");

}

}

void update() {

ChuyenDe cd = getForm(); try {

dao.update(cd); this.fillTable();

MsgBox.alert(this, "Update successfully");

} catch (Exception e) {

MsgBox.alert(this, "Update unsucessfully!");

}

}

void delete() {

if (!Auth.isManager()) {

MsgBox.alert(this, "You're not authorized to delete employee!");

} else {

String id = txtSubjectID.getText();

if (MsgBox.confirm(this, "Do you want to delete this subject?")) { try {

dao.delete(id); this.fillTable(); this.clearForm();

MsgBox.alert(this, "Delete sucessfully!");

} catch (Exception e) {

MsgBox.alert(this, "Delete unsucessfully!");

}

}

}

}

void choosePicture() {

if (fileChooser.showOpenDialog(this) == JFileChooser.APPROVE\_OPTION) { fileChooser.setDialogTitle("Choose Image");

File file = fileChooser.getSelectedFile(); XImage.save(file);

ImageIcon icon = XImage.read(file.getName()); lblPicture.setIcon(icon); lblPicture.setToolTipText(file.getName());

}

}

void fillTable() {

DefaultTableModel model = (DefaultTableModel) tblSubject.getModel(); model.setRowCount(0);

try {

List<ChuyenDe> list = dao.selectAll();

for (ChuyenDe cd : list) {

Object[] rows = {cd.getMaCD(), cd.getTenCD(), cd.getHocPhi(), cd.getThoiLuong(), cd.getHinh()};

model.addRow(rows);

}

} catch (Exception e) {

MsgBox.alert(this, "Data query error!");

}

}

void edit() {

String macd = (String) tblSubject.getValueAt(this.row, 0); ChuyenDe cd = dao.selectById(macd);

this.setForm(cd); tabs.setSelectedIndex(0); this.updateStatus();

}

void updateStatus() {

boolean edit = (this.row >= 0); boolean first = (this.row == 0);

boolean last = (this.row == tblSubject.getRowCount() - 1);

//form state txtSubjectID.setEditable(!edit); btnAdd.setEnabled(!edit); btnEdit.setEnabled(edit); btnDelete.setEnabled(edit);

//Directional state btnPrev.setEnabled(edit && !first); btnNext.setEnabled(edit && !last); btnFirst.setEnabled(edit && !first); btnLast.setEnabled(edit && !last);

}

void setForm(ChuyenDe cd) { txtSubjectID.setText(cd.getMaCD()); txtName.setText(cd.getTenCD()); txtFees.setText(String.valueOf(cd.getHocPhi())); txtDuration.setText(String.valueOf(cd.getThoiLuong())); txtDesc.setText(cd.getMoTa());

if (cd.getHinh() != null) { lblPicture.setToolTipText(cd.getHinh()); lblPicture.setIcon(XImage.read(cd.getHinh()));

}

}

private ChuyenDe getForm() { ChuyenDe cd = new ChuyenDe();

cd.setMaCD(txtSubjectID.getText()); cd.setTenCD(txtName.getText());

cd.setHocPhi(Double.valueOf(txtFees.getText()));

cd.setThoiLuong(Integer.valueOf(txtDuration.getText())); cd.setHinh(lblPicture.getToolTipText()); cd.setMoTa(txtDesc.getText());

return cd;

}

void clearForm() {

ChuyenDe cd = new ChuyenDe(); this.setForm(cd);

this.row = -1; this.updateStatus();

}

void first() { this.row = 0; this.edit();

}

void prev() {

if (row > 0) { this.row--;

this.edit();

}

}

void next() {

if (row < tblSubject.getRowCount() - 1) { this.row++;

this.edit();

}

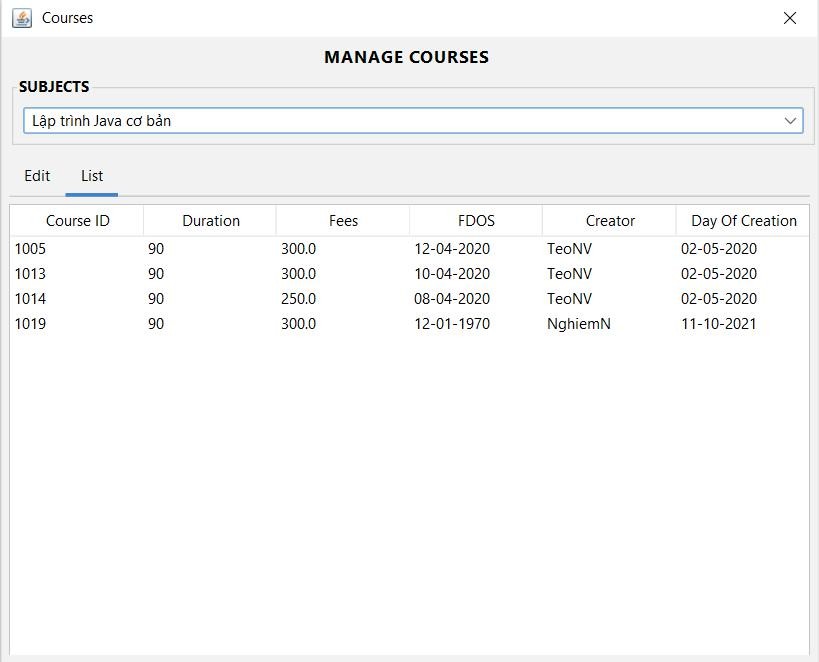
}

void last() {

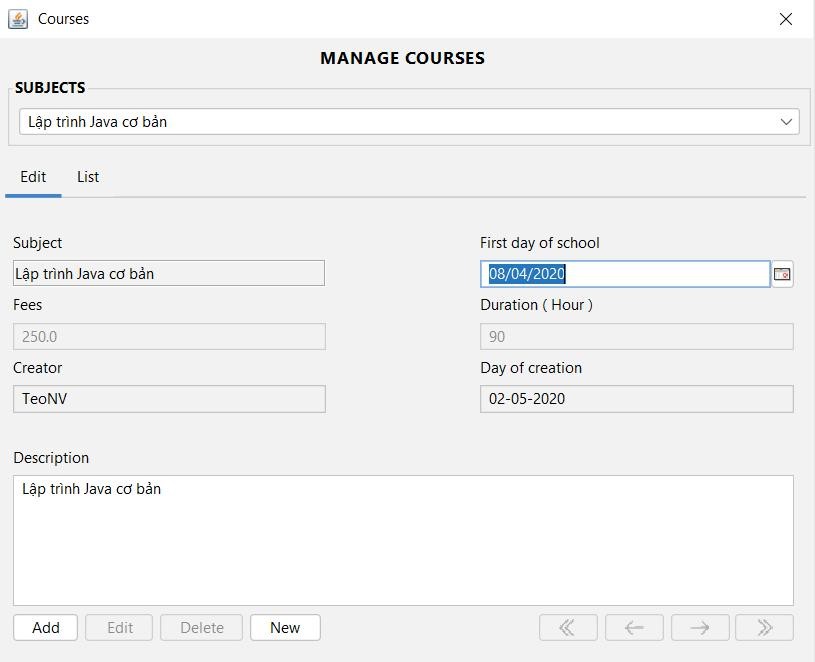
this.row = tblSubject.getRowCount() - 1; this.edit();

}

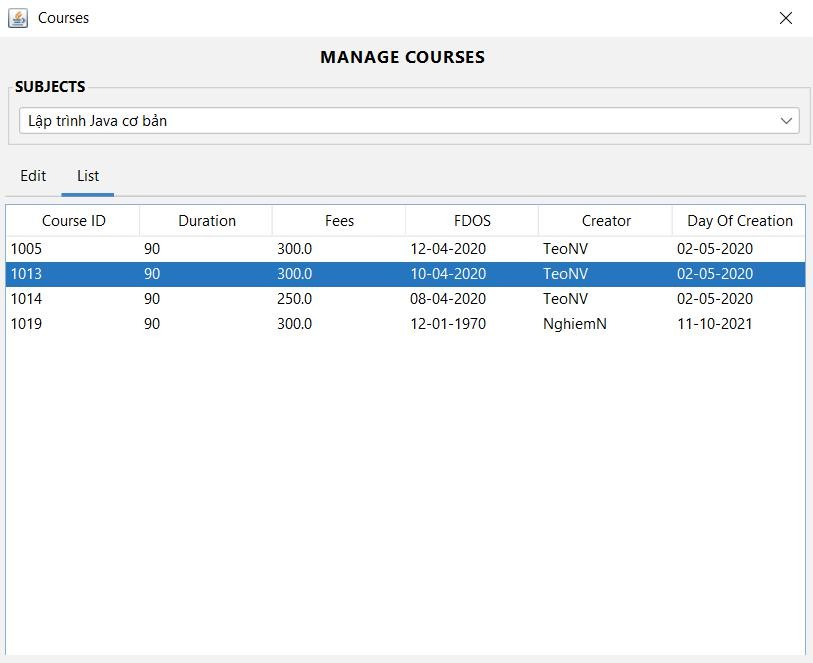
### Quản lý khóa học(KhoaHocJFrame)



Hình: trạng thái insert



Hình: Trạng thái update



Hình: bảng dữ liệu

#### MÔ TẢ HOẠT ĐỘNG

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc phải làm** |
| Constructor | Init() |
| windowOpened | this.fillComboBox(  ); this.load(); this.clear(); this.setStatus(true); |
| tblGridView.mouseClicked | if(evt.getClickCount() == 2){  this.index = tblGridView.rowAtPoint(evt.getPoint()); |

|  |  |
| --- | --- |
|  | if (this.index >= 0)  { this.edit(); tabs.setSelectedIndex(0);  }  } |
| btnInsert | Insert() |
| btnUpdate | Update() |
| btnDelete | Delete() |
| btnClear | Clear() |
| btnFirst | this.index = 0; this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++  ; this.edit(); |
| btnLast | this.index = tblGridView.getRowCount() - 1; this.edit(); |
| cboChuyenDe | selectComboBox(); |
| btnStudents | this.openHocVien(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

KhoaHocDAO khDAO = new KhoaHocDAO(); ChuyenDeDAO cdDAO = new ChuyenDeDAO(); int row = -1;

/\*\*

\* Creates new form courseJDialog

\*/

public CourseJDialog(java.awt.Frame parent, boolean modal) { super(parent, modal);

initComponents(); this.fillComboBoxChuyenDe(); this.fillTable(); this.updateStatus(); setTitle("Courses");

setLocationRelativeTo(parent);

}

void fillComboBoxChuyenDe() {

DefaultComboBoxModel model = (DefaultComboBoxModel) cboSubjects.getModel();

model.removeAllElements(); List<ChuyenDe> list = cdDAO.selectAll(); for (ChuyenDe cd : list) {

model.addElement(cd);

}

}

void fillTable() {

DefaultTableModel model = (DefaultTableModel) tblCourse.getModel(); model.setRowCount(0);

try {

ChuyenDe chuyenDe = (ChuyenDe) cboSubjects.getSelectedItem(); List<KhoaHoc> list = khDAO.selectByChuyenDe(chuyenDe.getMaCD()); for (KhoaHoc kh : list) {

Object[] row = {kh.getMaKH(), kh.getThoiLuong(), kh.getHocPhi(), XDate.toString(kh.getNgayKG(), "dd-MM-yyyy"), kh.getMaNV(),

XDate.toString(kh.getNgayTao(), "dd-MM-yyyy")};

model.addRow(row);

}

} catch (Exception e) {

throw new RuntimeException(e);

}

}

void chonChuyenDe() {

ChuyenDe chuyenDe = (ChuyenDe) cboSubjects.getSelectedItem(); txtDuration.setText(String.valueOf(chuyenDe.getThoiLuong())); txtFees.setText(String.valueOf(chuyenDe.getHocPhi())); txtDesc.setText(chuyenDe.getMoTa()); lblSubject.setText(chuyenDe.getTenCD());

this.fillTable(); this.row = -1; tabs.setSelectedIndex(1); this.updateStatus();

}

void insert() {

KhoaHoc kh = getForm(); try {

khDAO.insert(kh); this.fillTable();

MsgBox.alert(this, "Insert Successfully!");

} catch (Exception e) {

MsgBox.alert(this, "Insert Unsuccessfully!");

}

}

void update() {

KhoaHoc kh = getForm(); try {

khDAO.update(kh); this.fillTable();

MsgBox.alert(this, "Update successfully");

} catch (Exception e) {

MsgBox.alert(this, "Update unsucessfully!");

}

}

void delete() {

if (!Auth.isManager()) {

MsgBox.alert(this, "You're not authorized to delete employee!");

} else {

int id = Integer.parseInt(cboSubjects.getToolTipText());

if (MsgBox.confirm(this, "Do you want to delete this subject?")) { try {

khDAO.delete(id); this.fillTable();

MsgBox.alert(this, "Delete sucessfully!");

} catch (Exception e) {

MsgBox.alert(this, "Delete unsucessfully!");

}

}

}

}

void updateStatus() {

boolean edit = (this.row >= 0); boolean first = (this.row == 0);

boolean last = (this.row == tblCourse.getRowCount() - 1);

//Form state btnAdd.setEnabled(!edit); btnEdit.setEnabled(edit); btnDel.setEnabled(edit);

//Directional state btnFirst.setEnabled(edit && !first); btnPrev.setEnabled(edit && !first); btnNext.setEnabled(edit && !last); btnLast.setEnabled(edit && !last);

}

void edit() {

int makh = (int) tblCourse.getValueAt(this.row, 0); KhoaHoc kh = khDAO.selectById(makh); this.setForm(kh);

tabs.setSelectedIndex(0); this.updateStatus();

}

void clearForm() { txtCreator.setText(""); txtFDOS.setDate(new Date(0)); this.row = -1; this.updateStatus();

}

void setForm(KhoaHoc kh) { cboSubjects.setSelectedItem(cdDAO.selectById(kh.getMaCD())); cboSubjects.setToolTipText(String.valueOf(kh.getMaKH())); txtFDOS.setDate(kh.getNgayKG()); txtCreator.setText(kh.getMaNV()); txtDesc.setText(kh.getGhiChu()); txtFees.setText(String.valueOf(kh.getHocPhi())); txtDuration.setText(String.valueOf(kh.getThoiLuong()));

txtDayCreated.setText(XDate.toString(kh.getNgayTao(), "dd-MM-yyyy"));

}

KhoaHoc getForm() {

KhoaHoc kh = new KhoaHoc();

ChuyenDe cd = (ChuyenDe) cboSubjects.getSelectedItem(); if (txtCreator.getText().equals("")) {

kh.setMaNV(Auth.user.getMaNV());

} else {

kh.setMaNV(txtCreator.getText());

}

kh.setMaCD(cd.getMaCD()); kh.setThoiLuong(Integer.parseInt(txtDuration.getText())); kh.setNgayKG(txtFDOS.getDate()); kh.setHocPhi(Double.parseDouble(txtFees.getText())); kh.setGhiChu(txtDesc.getText()); kh.setMaKH(Integer.parseInt(cboSubjects.getToolTipText())); return kh;

}

void first() { this.row = 0; this.edit();

}

void prev() {

if (this.row > 0) { this.row--;

this.edit();

}

}

void next() {

if (this.row < tblCourse.getRowCount() - 1) {

this.row++; this.edit();

}

}

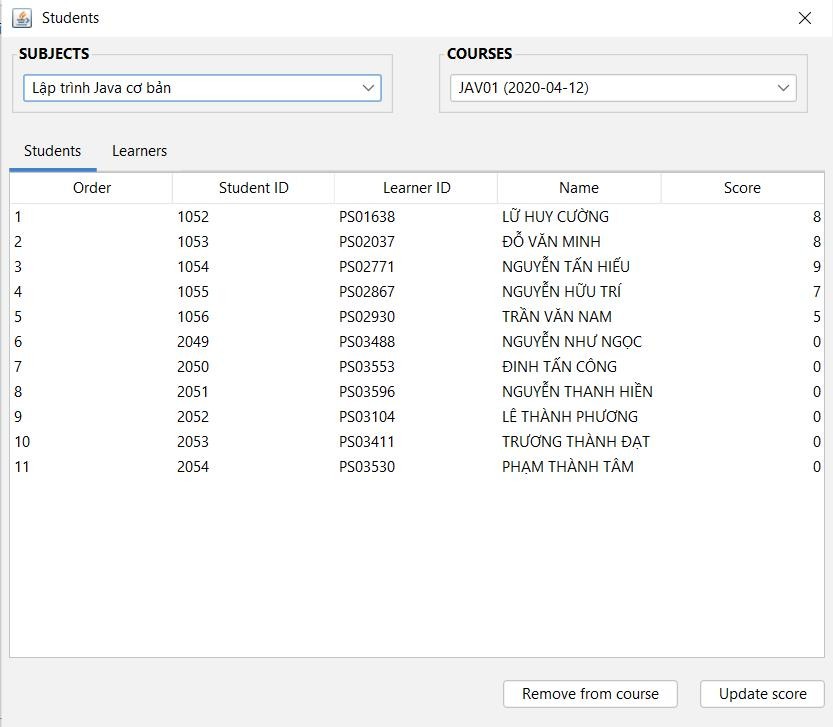
void last() {

this.row = tblCourse.getRowCount() - 1; this.edit();

}

### Quản lý học viên(HocVienJFrame)

Form chức năng này quản lý học viên của khóa học đang xem



MÔ TẢ HOẠT ĐỘNG

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
| Constructor | Init()  this.MaKH = MaKH; |
| windowOpened | this.fillComboBox(  );  this.fillGridView(); |
| btnThem | Insert() |
| cboTatCa cboDaNhap | this.fillGridView(); |

|  |  |
| --- | --- |
| cboChuaNhap |  |
| btnCapNhat | Update() |

#### HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

ChuyenDeDAO cdDAO = new ChuyenDeDAO(); KhoaHocDAO khDAO = new KhoaHocDAO(); NguoiHocDAO nhDAO = new NguoiHocDAO(); HocVienDAO hvDAO = new HocVienDAO();

/\*\*

\* Creates new form StudentJDialog

\*/

public StudentJDialog(java.awt.Frame parent, boolean modal) { super(parent, modal);

initComponents(); this.fillComboBoxChuyenDe(); setTitle("Students"); setLocationRelativeTo(parent);

}

void fillComboBoxChuyenDe() {

DefaultComboBoxModel model = (DefaultComboBoxModel) cboSubjects.getModel();

model.removeAllElements(); List<ChuyenDe> list = cdDAO.selectAll(); for (ChuyenDe cd : list) {

//add name of the subject to cbo model.addElement(cd);

}

this.fillComboBoxKhoaHoc();

}

void fillComboBoxKhoaHoc() {

DefaultComboBoxModel model = (DefaultComboBoxModel) cboCourses.getModel(); model.removeAllElements();

ChuyenDe cd = (ChuyenDe) cboSubjects.getSelectedItem(); if (cd != null) {

List<KhoaHoc> list = khDAO.selectByChuyenDe(cd.getMaCD()); for (KhoaHoc kh : list) {

model.addElement(kh);

}

this.fillTableHocVien();

}

}

void fillTableHocVien() {

DefaultTableModel model = (DefaultTableModel) tblStudent.getModel();

model.setRowCount(0);

KhoaHoc kh = (KhoaHoc) cboCourses.getSelectedItem(); if (kh != null) {

List<HocVien> list = hvDAO.selectByKhoaHoc(kh.getMaKH()); for (int i = 0; i < list.size(); i++) {

HocVien hv = list.get(i);

String hoten = nhDAO.selectById(hv.getMaNH()).getHoTen(); model.addRow(new Object[]{i + 1, hv.getMaHV(), hv.getMaNH(),

hoten, hv.getDiem()});

}

this.fillTableNguoiHoc();

}

}

void fillTableNguoiHoc() {

DefaultTableModel model = (DefaultTableModel) tblLearner.getModel(); model.setRowCount(0);

KhoaHoc kh = (KhoaHoc) cboCourses.getSelectedItem(); String keyword = txtSearch.getText();

List<NguoiHoc> list = nhDAO.selectNotInCourse(kh.getMaKH(), keyword); for (NguoiHoc nh : list) {

model.addRow(new Object[]{nh.getMaNH(), nh.getHoTen(), nh.isGioiTinh() ? "Male" : "Female",

XDate.toString(nh.getNgaySinh(), "dd-MM-yyyy"), nh.getDienThoai(), nh.getEmail()});

}

}

void addHocVien() {

KhoaHoc kh = (KhoaHoc) cboCourses.getSelectedItem(); int[] rows = tblLearner.getSelectedRows();

for (int row : rows) {

String manh = (String) tblLearner.getValueAt(row, 0); HocVien hv = new HocVien();

hv.setMaKH(kh.getMaKH()); hv.setDiem(0); hv.setMaNH(manh); hvDAO.insert(hv);

}

this.fillTableHocVien(); this.fillTableNguoiHoc();

}

void removeHocVien() {

if (!Auth.isManager()) {

MsgBox.alert(this, "You're not authorized to delete employee!");

} else {

int[] rows = tblStudent.getSelectedRows();

if (rows.length > 0 && MsgBox.confirm(this, "Do you want to delete student?")) {

for (int row : rows) {

int mahv = (Integer) tblStudent.getValueAt(row, 1); hvDAO.delete(mahv);

}

this.fillTableHocVien(); this.fillTableNguoiHoc();

}

}

}

void updateDiem() {

int n = tblStudent.getRowCount(); for (int i = 0; i < n; i++) {

int mahv = (Integer) tblStudent.getValueAt(i, 1); double diem = (Double) tblStudent.getValueAt(i, 4); HocVien hv = hvDAO.selectById(mahv); hv.setDiem(diem);

hvDAO.update(hv);

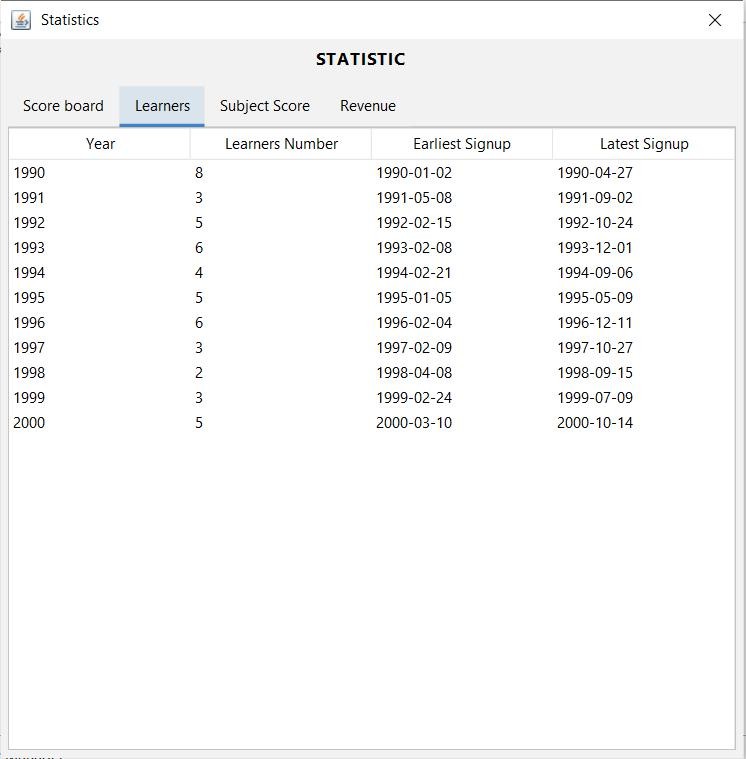
}

this.fillTableHocVien();

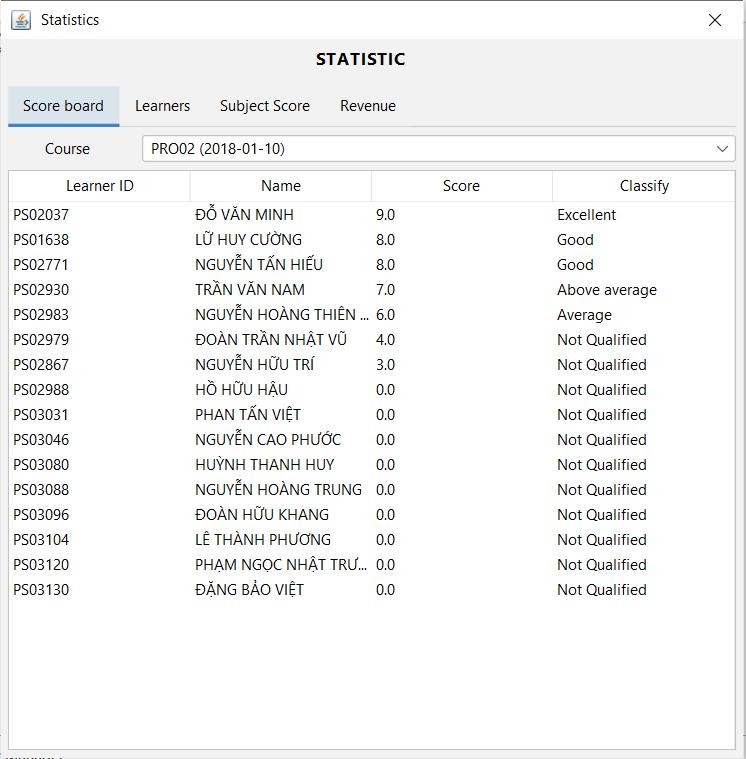
MsgBox.alert(this, "Update score successfully!");

}

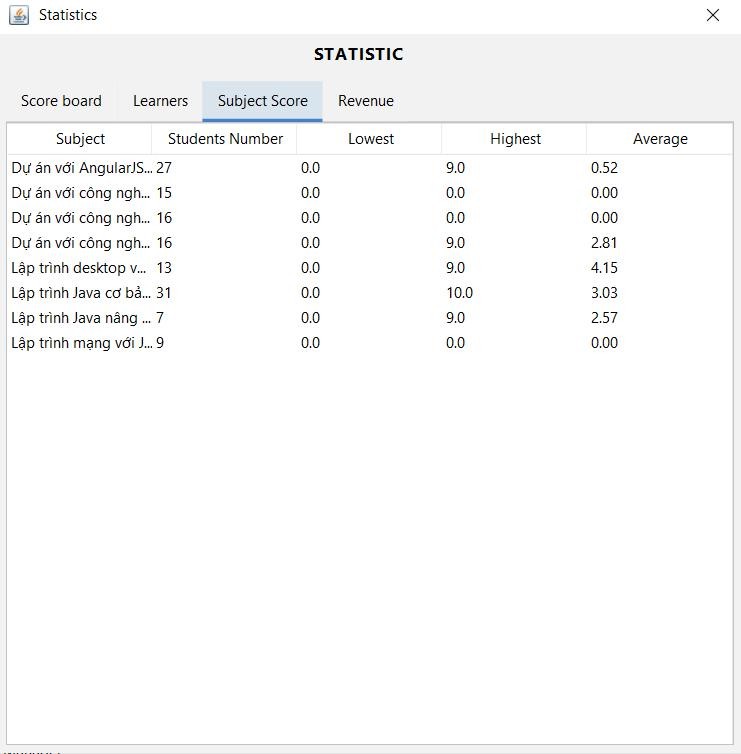
### Thống kê dữ liệu(ThongKeJFrame)



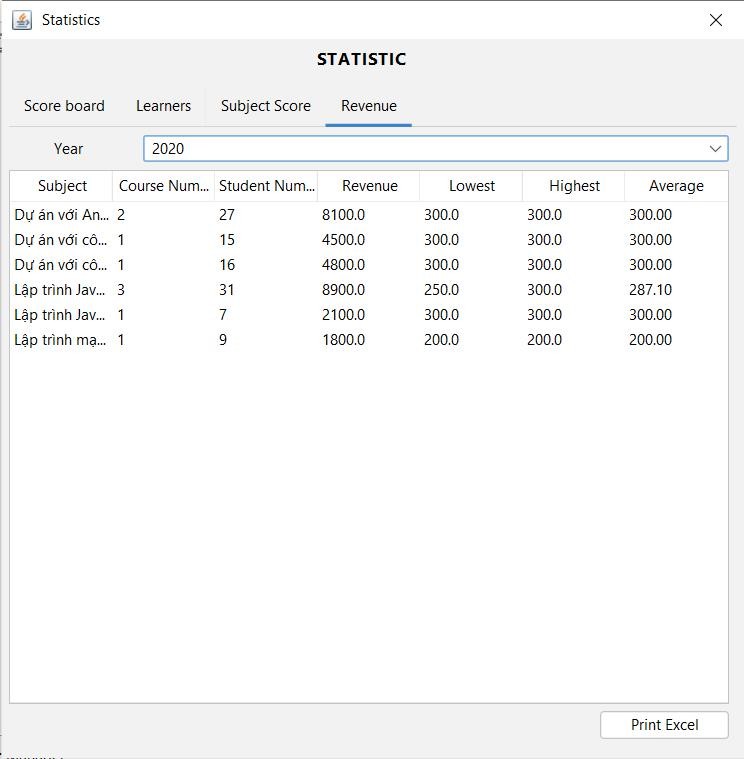
Hình: Thống kê số đăng ký từng năm



Hình: Bảng điểm của từng khóa



Hình: Thống kê số học viên và điểm theo chuyên đề



Hình: Thống kê doanh thu từng chuyên đề theo năm

#### MÔ TẢ HOẠT ĐỘNG

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
| Constructor | init(); tabs.setSelectedIndex(tabIndex); |

|  |  |
| --- | --- |
| windowOpened | fillComboBoxKhoaHoc(); fillTableBangDiem(); fillTableNguoiHoc(); fillTableKhoaHoc(); fillComboBoxNam(); fillTableDoanhThu(); |
| cboNam | fillTableDoanhThu(); |
| cboKhoaHoc | fillTableBangDiem(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

ThongKeDAO tkDAO = new ThongKeDAO(); KhoaHocDAO khDAO = new KhoaHocDAO();

/\*\*

\* Creates new form StatisticJDialog

\*/

public StatisticJDialog(java.awt.Frame parent, boolean modal) { super(parent, modal);

initComponents(); init(); setTitle("Statistics");

setLocationRelativeTo(parent);

}

void init() {

fillComboYear(); fillComboKhoaHoc(); fillTableBangDiem(); fillTableDiemChuyenDe(); fillTableNguoiHoc(); fillTableDoanhThu(); this.selectTab(0);

if (!Auth.isManager()) { tabs.remove(3);

}

}

public void selectTab(int index) { tabs.setSelectedIndex(index);

}

void fillTableBangDiem() {

DefaultTableModel model = (DefaultTableModel) tblScoreBoard.getModel();

model.setRowCount(0);

KhoaHoc kh = (KhoaHoc) cboCourse.getSelectedItem(); List<Object[]> list = tkDAO.getBangDiem(kh.getMaKH()); for (Object[] row : list) {

double score = (double) row[2]; model.addRow(new Object[]{row[0], row[1], score,

getXepLoai(score)});

}

}

void fillTableNguoiHoc() {

DefaultTableModel model = (DefaultTableModel) tblLearners.getModel(); model.setRowCount(0);

List<Object[]> list = tkDAO.getLuongNguoiHoc(); for (Object[] row : list) {

model.addRow(row);

}

}

void fillTableDiemChuyenDe() {

DefaultTableModel model = (DefaultTableModel) tblSubjectScore.getModel();

model.setRowCount(0);

List<Object[]> list = tkDAO.getDiemChuyenDe(); for (Object[] row : list) {

model.addRow(new Object[]{row[0], row[1], row[2], row[3], new

DecimalFormat("####0.00").format(row[4])});

}

}

void fillComboYear() {

DefaultComboBoxModel model = (DefaultComboBoxModel) cboYear.getModel();

model.removeAllElements();

List<Integer> list = khDAO.selectYears(); for (Integer year : list) {

model.addElement(year);

}

}

void fillTableDoanhThu() {

DefaultTableModel model = (DefaultTableModel) tblRevenue.getModel(); model.setRowCount(0);

int year = (Integer) cboYear.getSelectedItem(); List<Object[]> list = tkDAO.getDoanhThu(year); for (Object[] row : list) {

model.addRow(new Object[]{row[0], row[1], row[2], row[3], row[4], row[5], new

DecimalFormat("####0.00").format(row[6])});

}

}

void fillComboKhoaHoc() {

DefaultComboBoxModel model = (DefaultComboBoxModel) cboCourse.getModel();

model.removeAllElements(); List<KhoaHoc> list = khDAO.selectAll(); for (KhoaHoc kh : list) {

model.addElement(kh);

}

}

String getXepLoai(double score) { if (score < 5) {

return "Not Qualified";

} else if (score < 6.5) { return "Average";

} else if (score < 7.5) { return "Above average";

} else if (score < 9) { return "Good";

}

return "Excellent";

}

void printReport(JTable table, File file) { try {

File f = new File(file + ".xlsx"); TableModel model = table.getModel(); FileWriter excel = new FileWriter(f);

for (int i = 0; i < model.getColumnCount(); i++) { excel.write(model.getColumnName(i) + "\t");

}

excel.write("\n");

for (int i = 0; i < model.getRowCount(); i++) {

for (int j = 0; j < model.getColumnCount(); j++) { excel.write(model.getValueAt(i, j).toString() + "\t");

}

excel.write("\n");

}

if (MsgBox.confirm(this, "Print Successfully! Do you want to open it?")) {

Desktop.getDesktop().browse(f.toURI());

}

excel.close();

} catch (IOException e) {

throw new RuntimeException(e);

}

}

### Đóng cửa sổ giới thiệu (GioiThieuJDialog)

Hình: Giới thiệu sản phẩm

MÔ TẢ HOẠT ĐỘNG

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
| Constructor | setLocationRelativeTo(null); |
| lblLogo txtContent | this.dispose(); |

# Kiểm thử

## Kiểm thử form quản lý nhân viên

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Thao tác** | | |
| **Thêm mới** | **Cập nhật** | **Xóa** |
| Mã nhân viên | Không để trống Không trùng |  | Không xóa chính mình |
| Mật khẩu | Ít nhất 3 ký tự | |  |
| Họ và tên | Chỉ chứa alphabet và ký tự trắng | |  |

## Kiểmthửformquảnlýngườihọc

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Thao tác** | |
| **Thêm mới** | **Cập nhật** |
| Mã người học | Đúng7kýtự Không trùng |  |
| Họ và tên | Chỉ chứa alphabet và ký tự trắng | |
| Ngày sinh | Trước ngày hiện tại ít nhất 16 năm | |
| Địa chỉ email | Phải đúng định dạng email | |

## Kiểm thử form quản lý chuyên đề

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Thao tác** | |
| **Thêm mới** | **Cập nhật** |
| Mã chuyên đề | Đúng 5 ký tự |  |
| TenCD | Chỉ chứa alphabet và ký tự trắng | |
| HocPhi | Số dương (>=0) | |
| ThoiLuong | Số dương (>=0) | |
| Hinh | Không để trống | |

## Kiểm thử form quản lý Khóa học

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Thao tác** | |
| **Thêm mới** | **Cập nhật** |
| NgayKG | Sau ngày hiện tại | |

## Kiểm thử form quản lý học viên

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Thao tác** | |
| **Thêm mới** | **Cập nhật** |
| Điểm | Phải là số thực từ 0 đến 10 hoặc -1 | |

# Đóng gói và triểnkhai

## Hướng dẫn chuyển đổi jar thành exe

* Sử dụng phần mềm EXE4J để chuyển đổi jar sang exe
* Sử dụng phần mềm InnoSetup để đóng gói

## Hướng dẫn cài đặt triển khai

* Cài SQL Server 2008 trở lên
* Tạo CSDL EduSys bằng cách chạy file EduSys.sql
* Cài đặt JDK 1.8 trởlên
* Chạy file setup

## Hướng dẫn sử dụng phần mềm

* Đăng nhập với tài khoản username:thanhtung vàpassword: tung123
* Cấp các tài khoản có vai trò là manager
* Đăngnhập tài khoản mới và xóa tài khoản admin
* Xem hướng dẫn sử dụng trong menu trợ giúp của phần mềm