

|  |
| --- |
| **PROJECT DOCUMENT** |
| EduSys – CNTT (UDPM) |
| Hiểu qui trình thực hiện dự án phần mềm, sử dụng tài liệu để xây dựng phần mềm. |
| **NGHIÊN CỨU VÀ PHÁT TRIỂN CHƯƠNG TRÌNH FPOLY** |
| **26-10-2023** |

LỜI NÓI ĐẦU

##### Dự án mẫu là một môn học trong chương trình đào tạo CNTT của hệ cao đẳng thực hành FPT Polytechnic.

Mục tiêu của môn học này là cung cấp cho sinh viên một cái nhìn tổng quan về việc làm dự án phần mềm theo qui trình công nghiệp sát với thực tế hiện nay.

##### Qua môn học này sinh viên cần biết sử dụng tài liệu dự án được xây dựng sẵn để xây dựng ra một phần mềm – đó là phần mềm Quản lý đào tạo cho trung tâm đào tạo tin học ngắn hạn theo chuyên đề LapTrinhCity.

Trong chương trình đào tạo theo tiêu chuẩn CDIO thì sinh viên cần được huấn luyện các kỹ năng C, D, I, O (**C**onceiving — **D**esigning — **I**mplementing — **O**perating). Dự án này rèn luyện cho sinh viên các kỹ năng I và O đồng thời làm nền tảng cho sinh viên rèn luyện các kỹ năng còn lại C và D qua các môn học tiếp sau trong đó đặc biệt là dự án 1 và dự án 2 và cả thực tập tốt nghiệp.

##### Với phương pháp đào tạo đảo ngược như vậy chúng tôi mong muốn sinh viên nghề không những có đủ kỹ năng thực hiện mà còn được rèn luyện thêm kỹ phân tích và thiết kế nhằm giúp sinh viên phát triển sự nghiệp lâu dài học học tiếp lên các cấp độ cao hơn.

MỤC LỤC

1. [Phân tích 4](#_TOC_250030)
   1. [Hiện trạng 4](#_TOC_250029)
   2. [Yêu cầu hệ thống 4](#_TOC_250028)
   3. [Use case 4](#_TOC_250027)
2. [Thiết kế 7](#_TOC_250026)
   1. [Mô hình triển khai 7](#_TOC_250025)
   2. Thiết kế CSDL 8
      1. [Sơ đồ quan hệ thực thể 8](#_TOC_250024)
      2. [Thiết kế chi tiết các thực thể 9](#_TOC_250023)
   3. [Thiết kế giao diện 11](#_TOC_250022)
      1. [Sơ đồ tổ chức giao diện 11](#_TOC_250021)
      2. [Thiết kế giao diện cho các chức năng nghiệp vụ 12](#_TOC_250020)
3. [Thực hiện viết mã 38](#_TOC_250019)
   1. Viết mã tạo CSDL 38
      1. [Tạo CSDL 38](#_TOC_250018)
      2. [SQL truy vấn và thao tác bảng 40](#_TOC_250017)
      3. [Các thủ tục lưu tổng hợp thống kê 41](#_TOC_250016)
   2. Lập trình JDBC 43
      1. [Lớp hỗ trợ 43](#_TOC_250015)
      2. [Model class - Các lớp mô tả dữ liệu 46](#_TOC_250014)
      3. [DAO Class - Các lớp truy xuất dữ liệu 48](#_TOC_250013)
   3. [Viết mã cho ứng dụng 60](#_TOC_250012)
      1. [Các lớp tiện ích 60](#_TOC_250011)
      2. [Viết mã cho các chức năng 62](#_TOC_250010)
4. [Kiểm thử 108](#_TOC_250009)
   1. [Kiểm thử form quản lý nhân viên 108](#_TOC_250008)
   2. [Kiểm thử form quản lý người học 108](#_TOC_250007)
   3. [Kiểm thử form quản lý chuyên đề 108](#_TOC_250006)
   4. [Kiểm thử form quản lý Khóa học 109](#_TOC_250005)
   5. [Kiểm thử form quản lý học viên 109](#_TOC_250004)
5. [Đóng gói và triển khai 109](#_TOC_250003)
   1. [Hướng dẫn chuyển đổi jar thành exe 109](#_TOC_250002)
   2. [Hướng dẫn cài đặt triển khai 109](#_TOC_250001)
   3. [Hướng dẫn sử dụng phần mềm 109](#_TOC_250000)

# PHÂN TÍCH

### HIỆN TRẠNG

##### Trung tâm đào tạo tin học ngắn hạn LapTrinhCity đang đào tạo các khóa tin học ngắn hạn theo chuyên đề như Photoshop, Java, Web…

Việc quản lý khóa học, học viên, bảng điểm và doanh thu đang thực hiện thông qua excel. Hiện Trung tâm đang gặp khó khăn khi số lượng người học ngày một nhiều, dữ liệu ngày càng lớn nên việc quản lý excel gặp rất nhiều khó khăn, dễ sai sót mà không bảo mật.

### YÊU CẦU HỆ THỐNG

LapTrinhCity mong muốn xây dựng một phần mềm để giải quyết khó khăn trên.

### 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

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



* + - **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 năng tổng 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 xóa trong 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

##### Tên chuyên 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

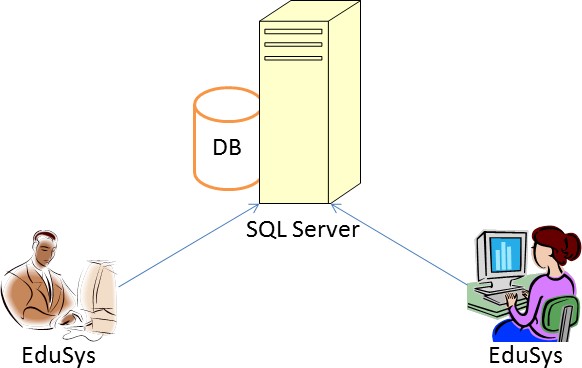
##### Họ và tên

* + - * + 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

##### Ứng dụng phần mềm được xây dựng để phục vụ cho nhiều người dùng nhưng cơ sở dữ liệu thì lưu trữ tập trung.



* + - 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

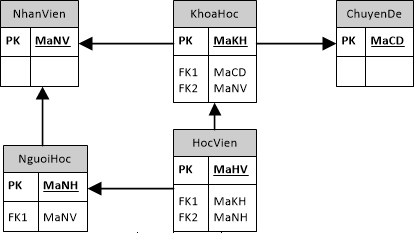
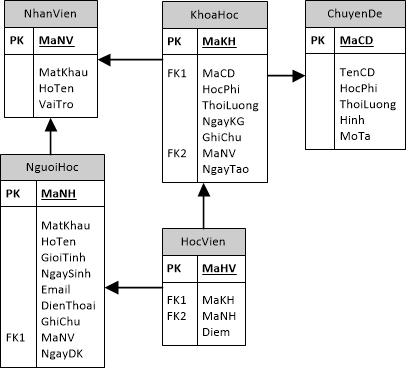
##### Khóa học: quản lý thông tin khóa học được tạo ra từ chuyên đề. Có nghĩa là mỗi chuyên đề có thể tạo ra nhiều khóa học.

* + - * 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

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

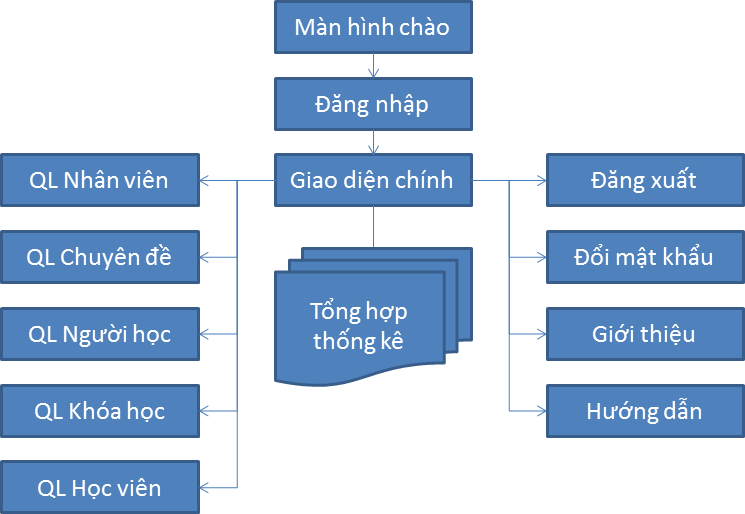
### THIẾT KẾ GIAO DIỆN

## Sơ đồ tổ chức giao diện

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

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

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



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

##### Trong phần thiết kế giao diện các bạn nên sử dụng gói hình ảnh được cung cấp sẵn trên tài nguyên học tập của LMS. Nếu các bạn không muốn sử dụng gói hình ảnh này thì bạn có thể chọn gói hình ảnh phù hợp của các bạn.

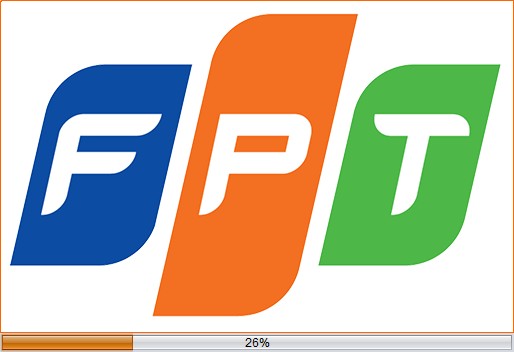
Lưu ý quan trọng:

##### Các bạn phải đặt tên đúng như hướng dẫn vì các tên này ảnh hưởng đến việc viết mã sau này. Nếu các bạn đặt tên khác thì phải chỉnh mã sau này theo tên mà các bạn đã đặt.

* + - * Trong lập trình giao diện người ta thường qui ước tiếp đầu ngữ của tên các thành phần giao diện như sau

|  |  |  |
| --- | --- | --- |
| **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 |  |
| … |  |  |

* + - 1. ***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

#### 

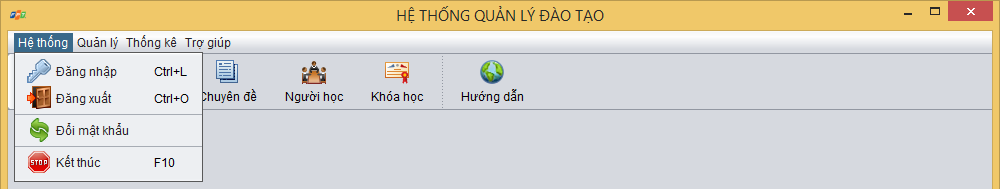
Các thành phần giao diện

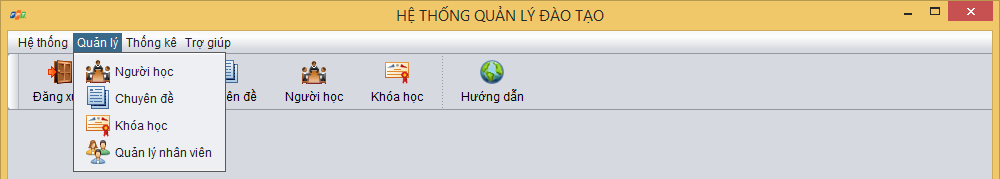
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Componel** | **Kiểu** | **Thuộc tính** |
| 1 | DangNhapJDialog | JDialog | title:HỆ THỐNG QUẢN LÝ ĐÀO TẠO  Layout: BorderLayout defaultCloseOperation:DO\_NOTHING |
| 2 | lblTitle | JLabel | Text:ĐĂNG NHẬP  Font:Tahoma, Bold, 18  Foreground: [0,102,51] |
| 3 | lblImage | JLabel | Icon: Security.png Layout.Direction:before |
| 4 | pnlForm | JPanel | Layout.Derection:center |

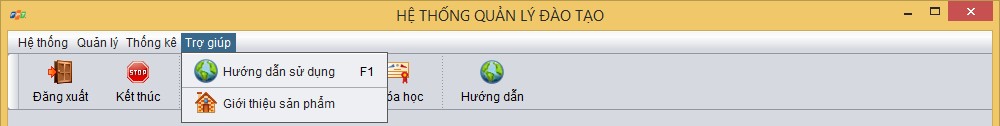
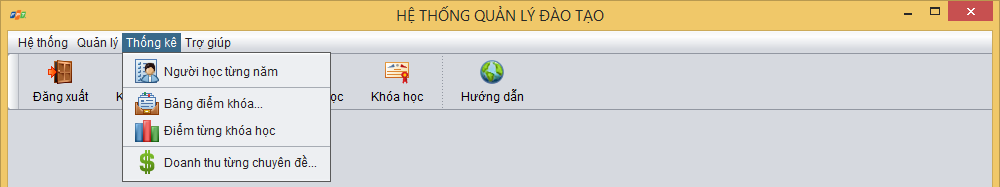
|  |  |  |  |
| --- | --- | --- | --- |
| 5 | lblMaNV | JLabel | Text: Tên đăng nhập |
| 6 | txtMaNV | JTextField |  |
| 7 | lblMatKhau | JLabel | Text: Mật khẩu |
| 8 | txtMatKhau | JPassword |  |
| 9 | btnDangNhap | JButton | Text:Đăng nhập |
| 10 | btnKetThuc | JButton | Text:Kết thúc |

#### Giao diện chính

#### 







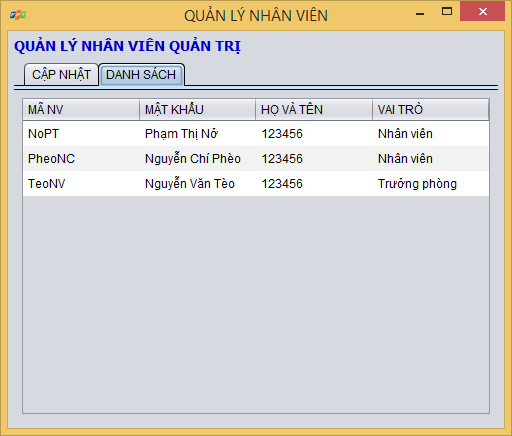
Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **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:Hướng dȁn sử dụng 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 dȁn  Icon: Globe.png Margin: [2, 10, 2, 10] |
| 27 | lblNoiDung | JLabel | Layout.Direction:center Icon:logo.png |
| 28 | pnlTrangThai | JPanel | Layout.Direction:last Layout:BorderLayout |
| 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 |

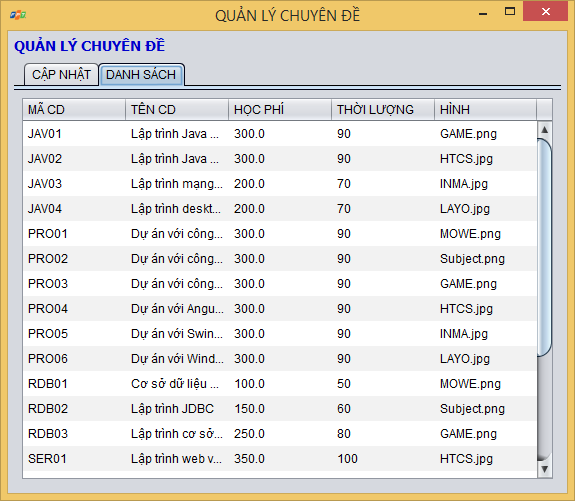
#### Quản lý người học

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

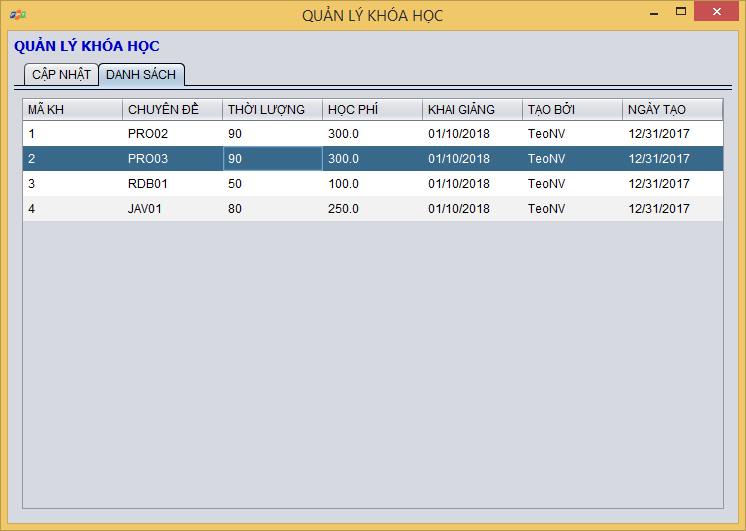


Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **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

#### 

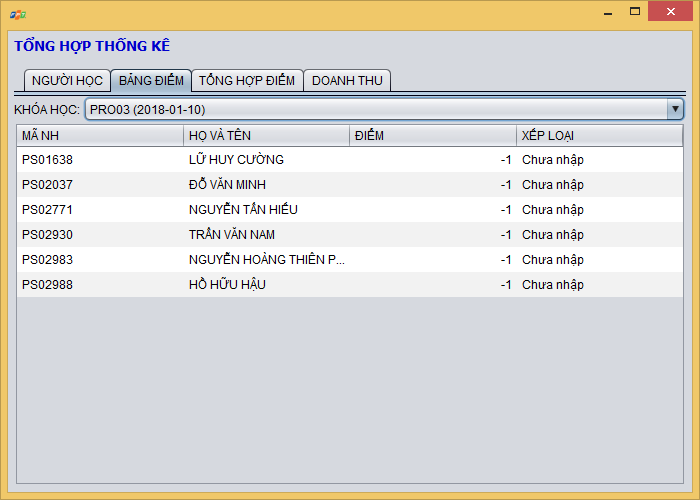
#### 

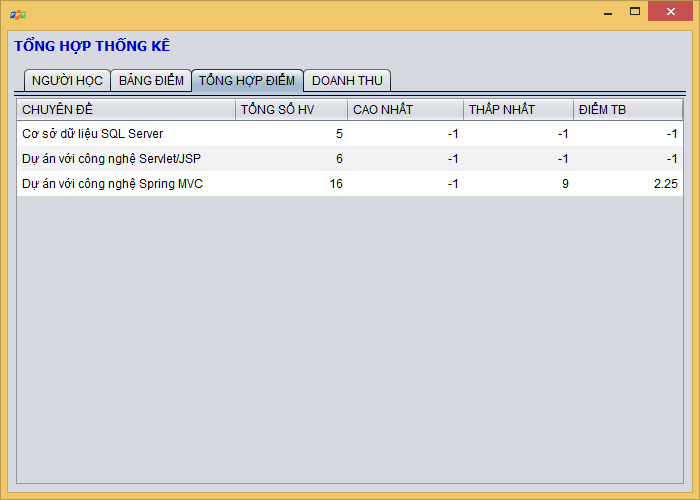
Các thành phần giao diện

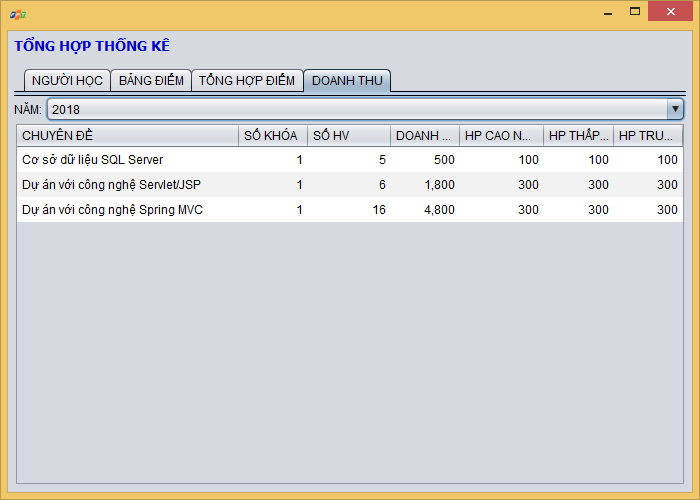
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | HocVienJFrame | JFrame | Title: QUẢN LÝ HỌC VIÊN KHÓA HỌC  defaultCloseOperation:DISPONSE |
| 2 | pnlHVKhac | JPanel | Border: TitledBorder(Title: HỌC VIÊN KHÁC) |
| 3 | pnlHVKH | JPanel | Border: TitledBorder(Title: 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 |
| 12 | btnCapNhat | JButton | Text: Cập nhật |

#### 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 |
| 5 | 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 dȁn sử dụng phần mềm. Trong phần viết mã sẽ hướng dȁn cách mở các trang web này khi người dùng nhấp vào mục hướng dȁn trong menu trợ giúp.

# THỰC HIỆN VIẾT MÃ

* 1. **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

#### 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=? |

#### 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

#### 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

#### 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 gọn hơn, dễ hiểu hơn. Trong phần này chúng ta xây dựng 2 lớp là XDate và Jdbc như sau.

* + - 1. ***Lớp tiện ích DateHelper***

##### 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.polypro.helper;

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

public class DateHelper {

static final SimpleDateFormat DATE\_FORMATER = new SimpleDateFormat("MM/dd/yyyy");

/\*\*

* Chuyển đổi String sang Date
* @param date là String cần chuyển
* @param pattern là định dạng thời gian
* @return Date kết quả

\*/

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

if(pattern.length > 0){ DATE\_FORMATER.applyPattern(pattern[0]);

}

if(date == null){

return DateHelper.now();

}

return DATE\_FORMATER.parse(date);

}

catch (ParseException ex) {

throw new RuntimeException(ex);

}

}

/\*\*

* *Chuyển đổi từ Date sang String*
* *@param date là Date cần chuyển đổi*
* *@param pattern là định dạng thời gian*
* *@return String kết quả*

\*/

public static String toString(Date date, String...pattern) { if(pattern.length > 0){

DATE\_FORMATER.applyPattern(pattern[0]);

}

if(date == null){

date = DateHelper.now();

}

return DATE\_FORMATER.format(date);

}

/\*\*

* *Lấy thời gian hiện tại*
* *@return Date kết quả*

\*/

public static Date now() { return new Date();

}

/\*\*

* *Bổ sung số ngày vào thời gian*
* *@param date thời gian hiện có*
* *@param days số ngày cần bổ sung váo date*
* *@return Date kết quả*

\*/

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

}

/\*\*

* *Bổ sung số ngày vào thời gian hiện hành*
* *@param days số ngày cần bổ sung vào thời gian hiện tại*
* *@return Date kết quả*

\*/

public static Date add(int days) { Date now = DateHelper.now();

now.setTime(now.getTime() + days\*24\*60\*60\*1000); return now;

}

}

#### Lớp tiện ích JdbcHelper

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

package com.polypro.helper;

import java.sql.Connection; import java.sql.DriverManager;

import java.sql.PreparedStatement; import java.sql.ResultSet;

import java.sql.SQLException;

public class JdbcHelper {

private static String driver="com.microsoft.sqlserver.jdbc.SQLServerDriver"; private static String dburl="jdbc:sqlserver://localhost;database=Polypro"; private static String username="sa";

private static String password="123456";

/\*

\* Nạp driver

\*/ static{

try {

Class.forName(driver);

}

catch (ClassNotFoundException ex) { throw new RuntimeException(ex);

}

}

/\*\*

* *Xây dựng PreparedStatement*
* *@param sql là câu lệnh SQL chứa có thể chứa tham số. Nó có thể là một lời gọi thủ tục lưu*
* *@param args là danh sách các giá trị được cung cấp cho các tham số trong câu lệnh sql*
* *@return PreparedStatement tạo được*
* *@throws java.sql.SQLException lỗi sai cú pháp*

\*/

public static PreparedStatement prepareStatement(String sql, Object...args) throws SQLException{ Connection connection = DriverManager.getConnection(dburl, username, password); PreparedStatement pstmt = null;

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

pstmt = connection.prepareCall(sql);

}

else{

pstmt = connection.prepareStatement(sql);

}

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

}

return pstmt;

}

/\*\*

* *Thực hiện câu lệnh SQL thao tác (INSERT, UPDATE, DELETE) hoặc thủ tục lưu thao tác dữ liệu*
* *@param sql là câu lệnh SQL chứa có thể chứa tham số. Nó có thể là một lời gọi thủ tục lưu*
* *@param args là danh sách các giá trị được cung cấp cho các tham số trong câu lệnh sql \**

\*/

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

PreparedStatement stmt = prepareStatement(sql, args); try {

stmt.executeUpdate();

}

finally{

stmt.getConnection().close();

}

}

catch (SQLException e) {

throw new RuntimeException(e);

}

}

/\*\*

* Thực hiện câu lệnh SQL truy vấn (SELECT) hoặc thủ tục lưu truy vấn dữ liệu
* @param sql là câu lệnh SQL chứa có thể chứa tham số. Nó có thể là một lời gọi thủ tục lưu
* @param args là danh sách các giá trị được cung cấp cho các tham số trong câu lệnh sql

\*/

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

PreparedStatement stmt = prepareStatement(sql, args); return stmt.executeQuery();

}

catch (SQLException e) {

throw new RuntimeException(e);

}

}

}

## 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.

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

package com.polypro.model;

public class NhanVien { private String maNV; private String matKhau; private String hoTen;

private boolean vaiTro = false;

@Override

public String toString() { return this.hoTen;

}

getters/setters

}

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

package com.polypro.model;

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;

}

getters/setters

}

#### NguoiHoc

package com.polypro.model;

import com.polypro.helper.XDate; import java.util.Date;

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 = XDate.now();

@Override

public String toString() { return this.hoTen;

}

getters/setters

}

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

package com.polypro.model;

import com.polypro.helper.XDate; import java.util.Date;

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 = XDate.now();

@Override

public String toString() {

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

}

getters/setters

}

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

package com.polypro.model;

public class HocVien { private int maHV; private int maKH; private String maNH;

private double diem = -1.0;

@Override

public String toString() { return this.toString();

}

getters/setters

}

## 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

/\*\*

* Thêm mới thực thể vào CSDL
* @param entity là thực thể chứa thông tin bản ghi mới

\*/

public void insert(Entity entity){}

/\*\*

* Cập nhật thực thể vào CSDL
* @param entity là thực thể chứa thông tin bản ghi cần cập nhật

\*/

public void update(Entity entity){}

/\*\*

* Xóa bản ghi khỏi CSDL
* @param id là mã của bản ghi cần xóa

\*/

public void delete(String id){}

/\*\*

* Truy vấn tất cả các các thực thể
* @return danh sách các thực thể

\*/

public List<Entity> select(){}

/\*\*

* Truy vấn thực thể theo mã
* @param id là mã của bản ghi được truy vấn
* @return thực thể chứa thông tin của bản ghi

\*/

public Entity findById(String id){}

Sau đây là các lớp DAO cụ thể

* + - 1. ***NhanVienDAO***

package com.polypro.dao;

import com.polypro.helper.Jdbc; import com.polypro.model.NhanVien; import java.sql.ResultSet;

import java.sql.SQLException; import java.util.ArrayList; import java.util.List;

public class NhanVienDAO {

public void insert(NhanVien model){

String sql="INSERT INTO NhanVien (MaNV, MatKhau, HoTen, VaiTro) VALUES (?, ?, ?, ?)"; Jdbc.executeUpdate(sql,

model.getMaNV(), model.getMatKhau(), model.getHoTen(), model.getVaiTro());

}

public void update(NhanVien model){

String sql="UPDATE NhanVien SET MatKhau=?, HoTen=?, VaiTro=? WHERE MaNV=?"; Jdbc.executeUpdate(sql,

model.getMatKhau(), model.getHoTen(), model.getVaiTro(), model.getMaNV());

###### }

public void delete(String MaNV){

###### String sql="DELETE FROM NhanVien WHERE MaNV=?"; Jdbc.executeUpdate(sql, MaNV);

}

###### public List<NhanVien> select(){

String sql="SELECT \* FROM NhanVien"; return select(sql);

###### }

public NhanVien findById(String manv){

###### String sql="SELECT \* FROM NhanVien WHERE MaNV=?"; List<NhanVien> list = select(sql, manv);

return list.size() > 0 ? list.get(0) : null;

###### }

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

###### try {

ResultSet rs = null; try {

###### rs = Jdbc.executeQuery(sql, args); while(rs.next()){

NhanVien model=readFromResultSet(rs); list.add(model);

###### }

}

###### finally{

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

###### }

}

###### catch (SQLException ex) {

throw new RuntimeException(ex);

###### }

return list;

###### }

private NhanVien readFromResultSet(ResultSet rs) throws SQLException{ NhanVien model=new NhanVien(); model.setMaNV(rs.getString("MaNV")); model.setMatKhau(rs.getString("MatKhau")); model.setHoTen(rs.getString("HoTen")); model.setVaiTro(rs.getBoolean("VaiTro"));

###### return model;

}

}

#### ChuyenDeDAO

package com.polypro.dao;

###### import com.polypro.helper.Jdbc; import com.polypro.model.ChuyenDe; import java.sql.ResultSet;

import java.sql.SQLException; import java.util.ArrayList; import java.util.List;

###### public class ChuyenDeDAO {

public void insert(ChuyenDe model){

###### String sql="INSERT INTO ChuyenDe (MaCD, TenCD, HocPhi, ThoiLuong, Hinh, MoTa) VALUES (?, ?, ?, ?, ?, ?)"; Jdbc.executeUpdate(sql,

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

###### }

public void update(ChuyenDe model){

###### String sql="UPDATE ChuyenDe SET TenCD=?, HocPhi=?, ThoiLuong=?, Hinh=?, MoTa=? WHERE MaCD=?"; Jdbc.executeUpdate(sql,

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

###### }

public void delete(String MaCD){

###### String sql="DELETE FROM ChuyenDe WHERE MaCD=?"; Jdbc.executeUpdate(sql, MaCD);

}

###### public List<ChuyenDe> select(){

String sql="SELECT \* FROM ChuyenDe"; return select(sql);

###### }

public ChuyenDe findById(String macd){

###### String sql="SELECT \* FROM ChuyenDe WHERE MaCD=?"; List<ChuyenDe> list = select(sql, macd);

return list.size() > 0 ? list.get(0) : null;

###### }

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

###### try {

ResultSet rs = null; try {

rs = Jdbc.executeQuery(sql, args);

while(rs.next()){

ChuyenDe model=readFromResultSet(rs); list.add(model);

}

}

finally{

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

}

}

catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private ChuyenDe readFromResultSet(ResultSet rs) throws SQLException{ ChuyenDe model=new ChuyenDe(); model.setMaCD(rs.getString("MaCD")); model.setHinh(rs.getString("Hinh")); model.setHocPhi(rs.getDouble("HocPhi")); model.setMoTa(rs.getString("MoTa")); model.setTenCD(rs.getString("TenCD")); model.setThoiLuong(rs.getInt("ThoiLuong"));

return model;

}

}

#### NguoiHocDAO

package com.polypro.dao;

import com.polypro.helper.Jdbc; import com.polypro.model.NguoiHoc; import java.sql.ResultSet;

import java.sql.SQLException; import java.util.ArrayList; import java.util.List;

public class NguoiHocDAO {

public void insert(NguoiHoc model){

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

Jdbc.executeUpdate(sql, model.getMaNH(), model.getHoTen(), model.getNgaySinh(), model.getGioiTinh(), model.getDienThoai(), model.getEmail(), model.getGhiChu(), model.getMaNV());

}

###### public void update(NguoiHoc model){

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

###### Jdbc.executeUpdate(sql, model.getHoTen(), model.getNgaySinh(), model.getGioiTinh(), model.getDienThoai(), model.getEmail(), model.getGhiChu(), model.getMaNV(), model.getMaNH());

}

###### public void delete(String id){

String sql="DELETE FROM NguoiHoc WHERE MaNH=?"; Jdbc.executeUpdate(sql, id);

###### }

public List<NguoiHoc> select(){

###### String sql="SELECT \* FROM NguoiHoc"; return select(sql);

}

###### public List<NguoiHoc> selectByKeyword(String keyword){ String sql="SELECT \* FROM NguoiHoc WHERE HoTen LIKE ?"; return select(sql, "%"+keyword+"%");

}

###### public List<NguoiHoc> selectByCourse(Integer makh){

String sql="SELECT \* FROM NguoiHoc WHERE MaNH NOT IN (SELECT MaNH FROM HocVien WHERE MaKH=?)";

###### return select(sql, makh);

}

###### public NguoiHoc findById(String manh){

String sql="SELECT \* FROM NguoiHoc WHERE MaNH=?"; List<NguoiHoc> list = select(sql, manh);

###### return list.size() > 0 ? list.get(0) : null;

}

###### private List<NguoiHoc> select(String sql, Object...args){ List<NguoiHoc> list=new ArrayList<>();

try {

###### ResultSet rs = null; try {

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

###### NguoiHoc model=readFromResultSet(rs); list.add(model);

}

###### }

finally{

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

}

}

catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private NguoiHoc readFromResultSet(ResultSet rs) throws SQLException{ NguoiHoc model=new NguoiHoc(); model.setMaNH(rs.getString("MaNH")); model.setHoTen(rs.getString("HoTen")); model.setNgaySinh(rs.getDate("NgaySinh")); model.setGioiTinh(rs.getBoolean("GioiTinh")); model.setDienThoai(rs.getString("DienThoai")); model.setEmail(rs.getString("Email")); model.setGhiChu(rs.getString("GhiChu")); model.setMaNV(rs.getString("MaNV")); model.setNgayDK(rs.getDate("NgayDK"));

return model;

}

}

#### KhoaHocDAO

package com.polypro.dao;

import com.polypro.helper.Jdbc; import com.polypro.model.KhoaHoc; import java.sql.ResultSet;

import java.sql.SQLException; import java.util.ArrayList; import java.util.List;

public class KhoaHocDAO {

public void insert(KhoaHoc model){

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

?)";

Jdbc.executeUpdate(sql, model.getMaCD(), model.getHocPhi(), model.getThoiLuong(), model.getNgayKG(), model.getGhiChu(), model.getMaNV());

}

public void update(KhoaHoc model){

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

Jdbc.executeUpdate(sql,

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

}

###### public void delete(Integer MaKH){

String sql="DELETE FROM KhoaHoc WHERE MaKH=?"; Jdbc.executeUpdate(sql, MaKH);

###### }

public List<KhoaHoc> select(){

###### String sql="SELECT \* FROM KhoaHoc"; return select(sql);

}

###### public KhoaHoc findById(Integer makh){

String sql="SELECT \* FROM KhoaHoc WHERE MaKH=?"; List<KhoaHoc> list = select(sql, makh);

###### return list.size() > 0 ? list.get(0) : null;

}

###### private List<KhoaHoc> select(String sql, Object...args){ List<KhoaHoc> list=new ArrayList<>();

try {

###### ResultSet rs = null; try {

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

###### KhoaHoc model = readFromResultSet(rs); list.add(model);

}

###### }

finally{

###### rs.getStatement().getConnection().close();

}

###### }

catch (SQLException ex) {

###### throw new RuntimeException(ex);

}

###### return list;

}

###### private KhoaHoc readFromResultSet(ResultSet rs) throws SQLException{ KhoaHoc model=new KhoaHoc();

model.setMaKH(rs.getInt("MaKH")); model.setHocPhi(rs.getDouble("HocPhi")); model.setThoiLuong(rs.getInt("ThoiLuong")); model.setNgayKG(rs.getDate("NgayKG")); model.setGhiChu(rs.getString("GhiChu"));

model.setMaNV(rs.getString("MaNV")); model.setNgayTao(rs.getDate("NgayTao")); model.setMaCD(rs.getString("MaCD"));

return model;

}

}

#### HocVienDAO

package com.polypro.dao;

import com.polypro.helper.Jdbc; import com.polypro.model.HocVien; import java.sql.ResultSet;

import java.sql.SQLException; import java.util.ArrayList; import java.util.List;

public class HocVienDAO {

public void insert(HocVien model){

String sql="INSERT INTO HocVien(MaKH, MaNH, Diem) VALUES(?, ?, ?)"; Jdbc.executeUpdate(sql,

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

}

public void update(HocVien model){

String sql="UPDATE HocVien SET MaKH=?, MaNH=?, Diem=? WHERE MaHV=?"; Jdbc.executeUpdate(sql,

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

}

public void delete(Integer MaHV){

String sql="DELETE FROM HocVien WHERE MaHV=?"; Jdbc.executeUpdate(sql, MaHV);

}

public List<HocVien> select(){

String sql="SELECT \* FROM HocVien"; return select(sql);

}

public HocVien findById(Integer mahv){

String sql="SELECT \* FROM HocVien WHERE MaHV=?"; List<HocVien> list = select(sql, mahv);

return list.size() > 0 ? list.get(0) : null;

}

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

try {

ResultSet rs = null; try {

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

HocVien model=readFromResultSet(rs); list.add(model);

}

}

finally{

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

}

}

catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private HocVien readFromResultSet(ResultSet rs) throws SQLException{ HocVien model=new HocVien();

model.setMaHV(rs.getInt("MaHV")); model.setMaKH(rs.getInt("KH")); model.setMaNH(rs.getString("MaNH")); model.setDiem(rs.getDouble("Diem")); return model;

}

}

* + - 1. ***ThongKeDAO***

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

package com.polypro.dao;

import com.polypro.helper.Jdbc; import java.sql.ResultSet; import java.sql.SQLException; import java.util.ArrayList;

import java.util.List;

public class ThongKeDAO {

public List<Object[]> getNguoiHoc(){ List<Object[]> list=new ArrayList<>(); try {

ResultSet rs = null; try {

String sql="{call sp\_ThongKeNguoiHoc}"; rs = Jdbc.executeQuery(sql);

###### while(rs.next()){ Object[] model={

rs.getInt("Nam"), rs.getInt("SoLuong"), rs.getDate("DauTien"), rs.getDate("CuoiCung")

###### };

list.add(model);

###### }

}

###### finally{

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

###### }

}

###### catch (SQLException ex) {

throw new RuntimeException(ex);

###### }

return list;

###### }

public List<Object[]> getBangDiem(Integer makh){ List<Object[]> list=new ArrayList<>();

###### try{

ResultSet rs = null; try {

###### String sql="{call sp\_BangDiem (?)}"; rs = Jdbc.executeQuery(sql, makh); while(rs.next()){

double diem = rs.getDouble("Diem"); String xepLoai = "Xuất sắc";

###### if(diem < 0){

xepLoai = "Chưa nhập";

###### }

else if(diem < 3){ xepLoai = "Kém";

###### }

else if(diem < 5){ xepLoai = "Yếu";

###### }

else if(diem < 6.5){ xepLoai = "Trung bình";

###### }

else if(diem < 7.5){ xepLoai = "Khá";

###### }

else if(diem < 9){ xepLoai = "Giỏi";

###### }

Object[] model={ rs.getString("MaNH"), rs.getString("HoTen"), diem,

###### xepLoai

};

###### list.add(model);

}

###### }

finally{

###### rs.getStatement().getConnection().close();

}

###### }

catch(SQLException ex){ ex.printStackTrace();

###### throw new RuntimeException(ex);

}

###### return list;

}

###### public List<Object[]> getDiemTheoChuyenDe(){ List<Object[]> list=new ArrayList<>();

try {

###### ResultSet rs = null; try {

String sql="{call sp\_ThongKeDiem}"; rs = Jdbc.executeQuery(sql); while(rs.next()){

###### Object[] model={ rs.getString("ChuyenDe"), rs.getInt("SoHV"), rs.getDouble("ThapNhat"), rs.getDouble("CaoNhat"), rs.getDouble("TrungBinh")

};

###### list.add(model);

}

###### }

finally{

###### rs.getStatement().getConnection().close();

}

###### }

catch (SQLException ex) {

###### throw new RuntimeException(ex);

}

###### return list;

}

###### public List<Object[]> getDoanhThu(int nam){ List<Object[]> list=new ArrayList<>();

try {

###### ResultSet rs = null; try {

String sql="{call sp\_ThongKeDoanhThu (?)}"; rs = Jdbc.executeQuery(sql, nam); while(rs.next()){

Object[] model={ rs.getString("ChuyenDe"),

rs.getInt("SoKH"),

rs.getInt("SoHV"), rs.getDouble("DoanhThu"), rs.getDouble("ThapNhat"), rs.getDouble("CaoNhat"), rs.getDouble("TrungBinh")

};

list.add(model);

}

}

finally{

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

}

}

catch (SQLException e) {

throw new RuntimeException(e);

}

return list;

}

}

### VIẾT MÃ CHO ỨNG DỤNG

## Các lớp tiện ích

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.

* + - 1. ***ShareHelper***

Lớp ShareHelper cung cấp phương thức tiện ích quản lý người đăng nhập và làm việc với hình ảnh chia sẽ trong tất cả các form của ứng dụng

*package com.polypro.helper;*

*import com.polypro.model.NhanVien; import java.awt.Image;*

*import java.io.File; import java.nio.file.Files; import java.nio.file.Path; import java.nio.file.Paths;*

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

*public class ShareHelper {*

*/\*\**

* *Ảnh biểu tượng của ứng dụng, xuất hiện trên mọi cửa sổ*

*\*/*

*public static final Image APP\_ICON; static{*

*// Tải biểu tượng ứng dụng*

*String file = "/com/polypro/icon/fpt.png";*

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

*}*

*/\*\**

* *Sao chép file logo chuyên đề vào thư mục logo*
* *@param file là đối tượng file ảnh*
* *@return chép được hay không*

*\*/*

*public static boolean saveLogo(File file){ File dir = new File("logos");*

*// Tạo thư mục nếu chưa tồn tại if(!dir.exists()){*

*dir.mkdirs();*

*}*

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

*// Copy vào thư mục logos (đè nếu đã tồn tại) Path source = Paths.get(file.getAbsolutePath());*

*Path destination = Paths.get(newFile.getAbsolutePath()); Files.copy(source, destination, StandardCopyOption.REPLACE\_EXISTING); return true;*

*}*

*catch (Exception ex) { return false;*

*}*

*}*

*/\*\**

* *Đọc hình ảnh logo chuyên đề*
* *@param fileName là tên file logo*
* *@return ảnh đọc được*

*\*/*

*public static ImageIcon readLogo(String fileName){ File path = new File("logos", fileName);*

*return new ImageIcon(path.getAbsolutePath());*

*}*

*/\*\**

* *Đối tượng này chứa thông tin người sử dụng sau khi đăng nhập*

*\*/*

*public static NhanVien USER = null;*

*/\*\**

* *Xóa thông tin của người sử dụng khi có yêu cầu đăng xuất*

*\*/*

*public static void logoff() { ShareHelper.USER = null;*

*}*

*/\*\**

* *Kiểm tra xem đăng nhập hay chưa*
* *@return đăng nhập hay chưa*

*\*/*

*public static boolean authenticated() { return ShareHelper.USER != null;*

*}*

*}*

#### DialogHelper

Lớp DialogHelper chứa các phương thức đơn giản hóa JOptionPane

package com.polypro.helper;

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

public class DialogHelper {

/\*\*

* Hiển thị thông báo cho người dùng
* @param parent là cửa sổ chứa thông báo
* @param message là thông báo

\*/

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

"Hệ thống quản lý đào tạo", JOptionPane.INFORMATION\_MESSAGE);

}

/\*\*

* Hiển thị thông báo và yêu cầu người dùng xác nhận
* @param parent là cửa sổ chứa thông báo
* @param message là câu hỏi yes/no
* @return là kết quả nhận được true/false

\*/

public static boolean confirm(Component parent, String message) { int result = JOptionPane.showConfirmDialog(parent, message,

"Hệ thống quản lý đào tạo",

JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE); return result == JOptionPane.YES\_OPTION;

}

/\*\*

* Hiển thị thông báo yêu cầu nhập dữ liệu
* @param parent là cửa sổ chứa thông báo
* @param message là thông báo nhắc nhở nhập
* @return là kết quả nhận được từ người sử dụng nhập vào

\*/

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

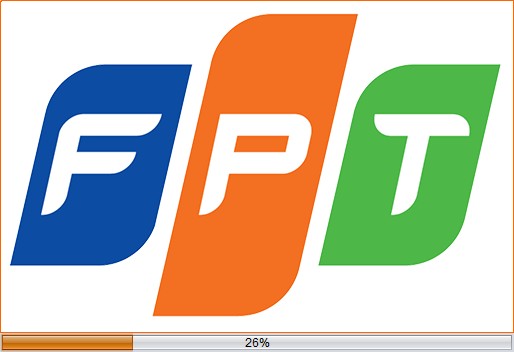
"Hệ thống quản lý đào tạo", JOptionPane.INFORMATION\_MESSAGE);

}

}

## 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.

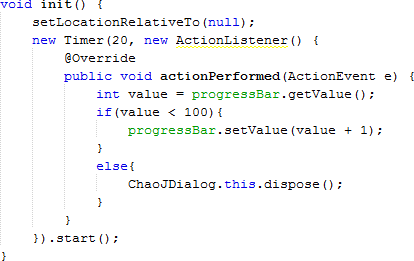
* + - 1. ***Màn hình chào (ChaoJDialog)***

##### HOẠT ĐỘNG

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

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

Bổ sung vào cuối cửa sổ hàm init() sau đây



#### 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.

NhanVienDAO dao = new NhanVienDAO(); void init(){

setLocationRelativeTo(null);

}

void login() {

String manv = txtMaNV.getText();

String matKhau = new String(txtMatKhau.getPassword()); try {

NhanVien nhanVien = dao.findById(manv); if(nhanVien != null){

String matKhau2 = nhanVien.getMatKhau(); if(matKhau.equals(matKhau2)){

ShareHelper.USER = nhanVien;

DialogHelper.alert(this, "Đăng nhập thành công!"); this.dispose();

}

else{

DialogHelper.alert(this, "Sai mật khẩu!");

}

}

else{

DialogHelper.alert(this, "Sai tên đăng nhập!");

}

}

catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");

}

}

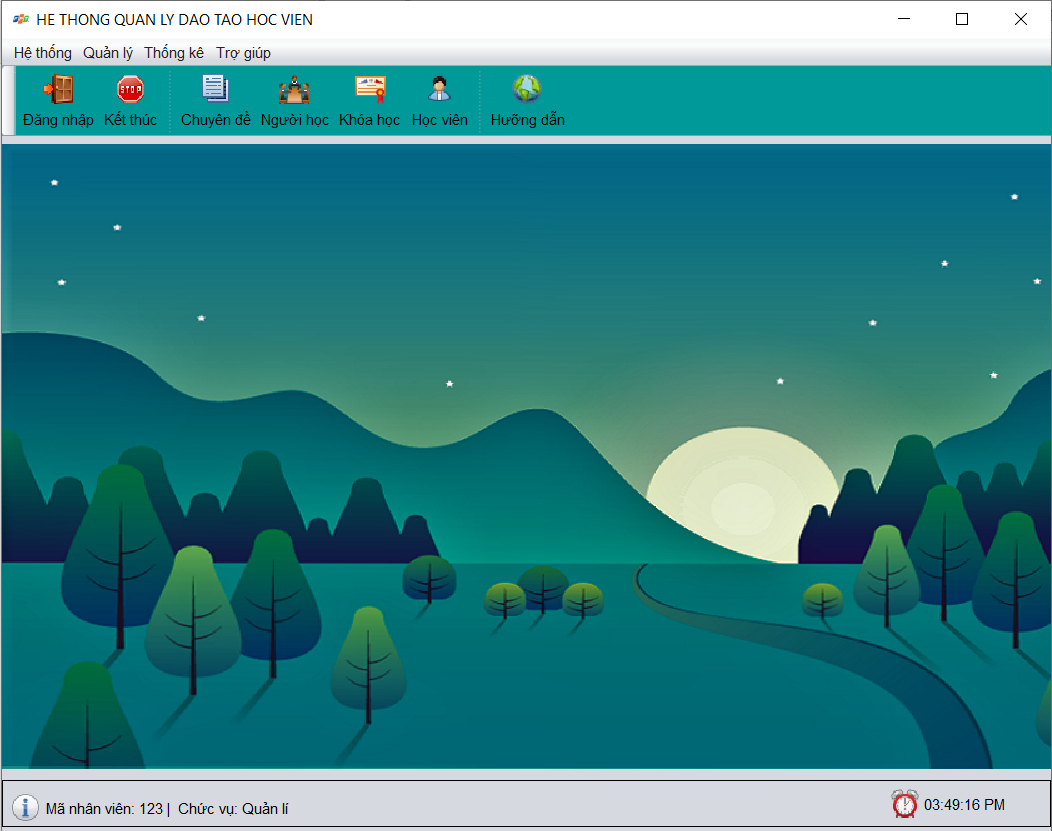
void exit(){

if(DialogHelper.confirm(this, "Bạn có muốn thoát khỏi ứng dụng không?")){ System.exit(0);

}

}

#### 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 dȁn | openAbout() |
| Giới thiệu | openWebsite() |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

void init() { setSize(1000, 600);

setIconImage(ShareHelper.APP\_ICON); setLocationRelativeTo(null);

new Timer(1000, new ActionListener() {

SimpleDateFormat format = new SimpleDateFormat("hh:mm:ss a"); @Override

public void actionPerformed(ActionEvent e) { lblDongHo.setText(format.format(new Date()));

}

}).start(); this.openWelcome(); this.openLogin();

}

void openLogin(){

new DangNhapJDialog(this, true).setVisible(true);

}

void openWelcome(){

new ChaoJDialog(this, true).setVisible(true);

}

void logoff(){ ShareHelper.logoff(); this.openLogin();

}

void exit(){

if(DialogHelper.confirm(this, "Bạn thực sự muốn kết thúc?")){ System.exit(0);

}

}

void openThongKe(int index){ if(ShareHelper.authenticated()){

new ThongKeJFrame(index).setVisible(true);

}

else{

DialogHelper.alert(this, "Vui lòng đăng nhập!");

}

}

void openNhanVien(){ if(ShareHelper.authenticated()){

new NhanVienJFrame().setVisible(true);

}

else{

DialogHelper.alert(this, "Vui lòng đăng nhập!");

}

}

void openKhoaHoc(){ if(ShareHelper.authenticated()){

new KhoaHocJFrame().setVisible(true);

}

else{

DialogHelper.alert(this, "Vui lòng đăng nhập!");

}

}

void openChuyenDe(){ if(ShareHelper.authenticated()){

new ChuyenDeJFrame().setVisible(true);

}

else{

DialogHelper.alert(this, "Vui lòng đăng nhập!");

}

}

void openNguoiHoc(){ if(ShareHelper.authenticated()){

new NguoiHocJFrame().setVisible(true);

}

else{

DialogHelper.alert(this, "Vui lòng đăng nhập!");

}

}

void openAbout(){

new GioiThieuJDialog(this, true).setVisible(true);

}

void openWebsite(){ try {

Desktop.getDesktop().browse(new File("help/index.html").toURI());

}

catch (IOException ex) {

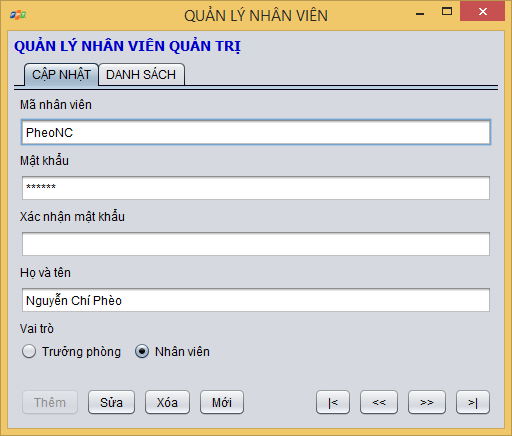
DialogHelper.alert(this, "Không tìm thấy file hướng dẫn!");

}

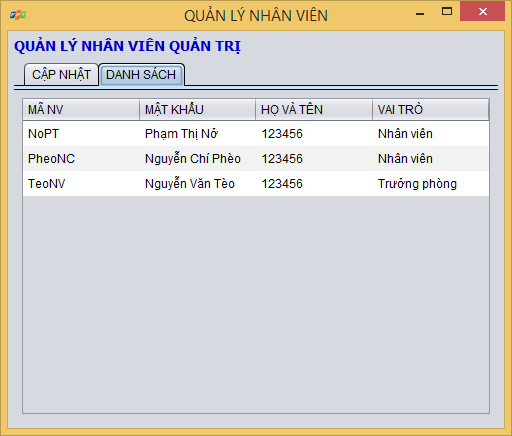
}

#### 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

int index = 0; // vị trí của nhân viên đang hiển thị trên form NhanVienDAO dao = new NhanVienDAO();

void init(){ setIconImage(ShareHelper.APP\_ICON); setLocationRelativeTo(null);

}

void load() {

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

try {

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

Object[] row = { nv.getMaNV(), nv.getMatKhau(), nv.getHoTen(),

nv.getVaiTro()?"Trưởng phòng":"Nhân viên"

};

model.addRow(row);

}

}

catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");

}

}

void insert(){

NhanVien model = getModel();

String confirm = new String(txtXacNhanMK.getPassword()); if(confirm.equals(model.getMatKhau())){

try {

dao.insert(model); this.load();

this.clear();

DialogHelper.alert(this, "Thêm mới thành công!");

}

catch (Exception e) {

DialogHelper.alert(this, "Thêm mới thất bại!");

}

}

else{

DialogHelper.alert(this, "Xác nhận mật khẩu không đúng!");

}

}

void update(){

NhanVien model = getModel();

String confirm = new String(txtXacNhanMK.getPassword()); if(!confirm.equals(model.getMatKhau())){

DialogHelper.alert(this, "Xác nhận mật khẩu không đúng!");

}

else{

try {

dao.update(model); this.load();

DialogHelper.alert(this, "Cập nhật thành công!");

}

catch (Exception e) {

DialogHelper.alert(this, "Cập nhật thất bại!");

}

}

}

void delete(){

if(DialogHelper.confirm(this, "Bạn thực sự muốn xóa nhân viên này?")){ String manv = txtMaNV.getText();

try {

dao.delete(manv); this.load();

this.clear();

DialogHelper.alert(this, "Xóa thành công!");

}

catch (Exception e) { DialogHelper.alert(this, "Xóa thất bại!");

}

}

}

void edit() { try {

String manv = (String) tblGridView.getValueAt(this.index, 0); NhanVien model = dao.findById(manv);

if(model != null){ this.setModel(model); this.setStatus(false);

}

}

catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");

}

}

void clear(){

this.setModel(new NhanVien()); this.setStatus(true);

}

void setModel(NhanVien model){ txtMaNV.setText(model.getMaNV()); txtHoTen.setText(model.getHoTen()); txtMatKhau.setText(model.getMatKhau()); txtXacNhanMK.setText(model.getMatKhau()); rdoTruongPhong.setSelected(model.getVaiTro()); rdoNhanVien.setSelected(!model.getVaiTro());

}

NhanVien getModel(){

NhanVien model = new NhanVien(); model.setMaNV(txtMaNV.getText()); model.setHoTen(txtHoTen.getText()); model.setMatKhau(new String(txtMatKhau.getPassword())); model.setVaiTro(rdoTruongPhong.isSelected());

return model;

}

void setStatus(boolean insertable){ txtMaNV.setEditable(insertable); btnInsert.setEnabled(insertable); btnUpdate.setEnabled(!insertable); btnDelete.setEnabled(!insertable);

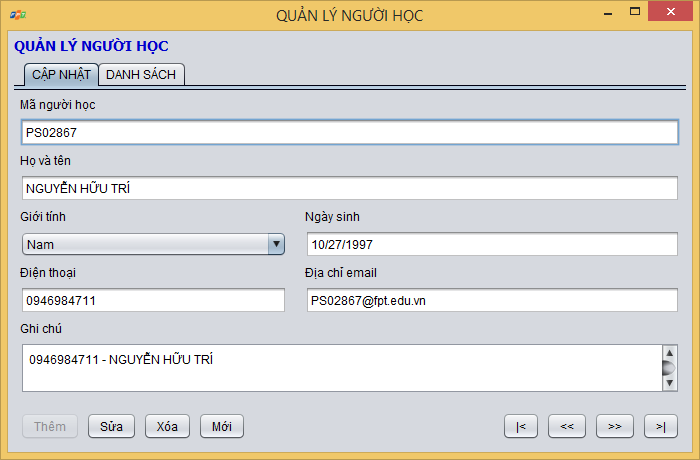
boolean first = this.index > 0;

boolean last = this.index < tblGridView.getRowCount() - 1; btnFirst.setEnabled(!insertable && first); btnPrev.setEnabled(!insertable && first); btnNext.setEnabled(!insertable && last); btnLast.setEnabled(!insertable && last);

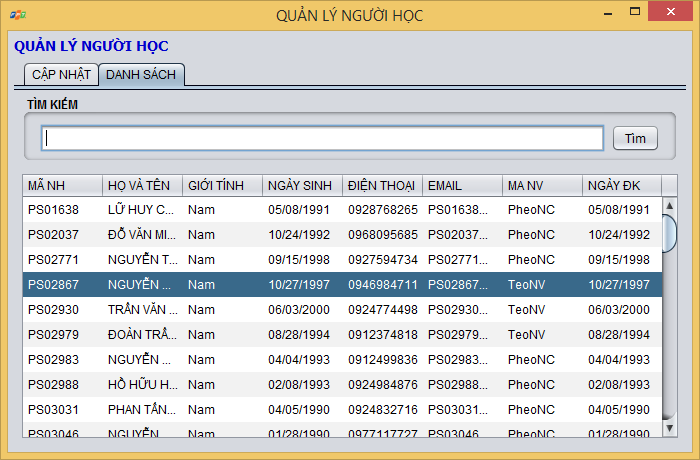
}

#### 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

int index = 0;

NguoiHocDAO dao = new NguoiHocDAO();

void init() { setIconImage(ShareHelper.APP\_ICON); setLocationRelativeTo(null);

}

void load() {

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

try {

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

Object[] row = { nh.getMaNH(), nh.getHoTen(),

nh.getGioiTinh()?"Nam":"Nữ", DateHelper.toString(nh.getNgaySinh()), nh.getDienThoai(),

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

DateHelper.toString(nh.getNgayDK())

};

model.addRow(row);

}

}

catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");

}

}

void insert(){

NguoiHoc model = getModel(); try {

dao.insert(model); this.load();

this.clear();

DialogHelper.alert(this, "Thêm mới thành công!");

}

catch (Exception e) {

DialogHelper.alert(this, "Thêm mới thất bại!");

}

}

void update(){

NguoiHoc model = getModel(); try {

dao.update(model); this.load();

DialogHelper.alert(this, "Cập nhật thành công!");

}

catch (Exception e) {

DialogHelper.alert(this, "Cập nhật thất bại!");

}

}

void delete(){

if(DialogHelper.confirm(this, "Bạn thực sự muốn xóa người học này?")){ String manh = txtMaNH.getText();

try {

dao.delete(manh); this.load();

this.clear();

DialogHelper.alert(this, "Xóa thành công!");

}

catch (HeadlessException e) { DialogHelper.alert(this, "Xóa thất bại!");

}

}

}

void clear(){

NguoiHoc model = new NguoiHoc(); model.setMaNV(ShareHelper.USER.getMaNV()); model.setNgayDK(DateHelper.now()); this.setModel(model);

}

void edit() { try {

String manh = (String) tblGridView.getValueAt(this.index, 0); NguoiHoc model = dao.findById(manh);

if(model != null){ this.setModel(model); this.setStatus(false);

}

}

catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");

}

}

void setModel(NguoiHoc model){ txtMaNH.setText(model.getMaNH()); txtHoTen.setText(model.getHoTen()); cboGioiTinh.setSelectedIndex(model.getGioiTinh()?0:1); txtNgaySinh.setText(DateHelper.toString(model.getNgaySinh())); txtDienThoai.setText(model.getDienThoai()); txtEmail.setText(model.getEmail()); txtGhiChu.setText(model.getGhiChu());

}

NguoiHoc getModel() {

NguoiHoc model = new NguoiHoc(); model.setMaNH(txtMaNH.getText()); model.setHoTen(txtHoTen.getText()); model.setGioiTinh(cboGioiTinh.getSelectedIndex() == 0); model.setNgaySinh(DateHelper.toDate(txtNgaySinh.getText())); model.setDienThoai(txtDienThoai.getText()); model.setEmail(txtEmail.getText()); model.setGhiChu(txtGhiChu.getText()); model.setMaNV(ShareHelper.USER.getMaNV()); model.setNgayDK(DateHelper.now());

return model;

}

void setStatus(boolean insertable){ txtMaNH.setEditable(insertable); btnInsert.setEnabled(insertable); btnUpdate.setEnabled(!insertable); btnDelete.setEnabled(!insertable);

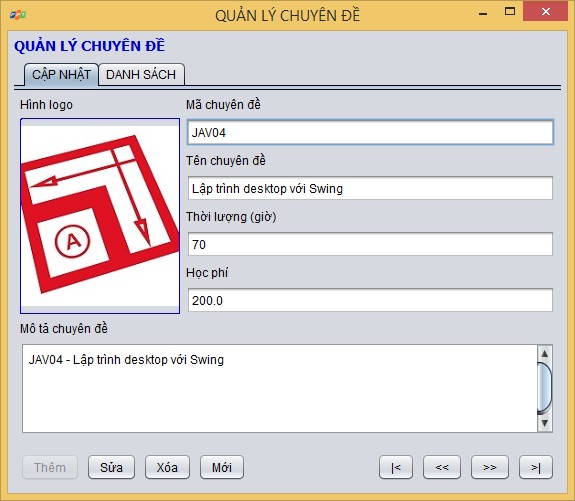
boolean first = this.index > 0;

boolean last = this.index < tblGridView.getRowCount() - 1; btnFirst.setEnabled(!insertable && first); btnPrev.setEnabled(!insertable && first); btnLast.setEnabled(!insertable && last); btnNext.setEnabled(!insertable && last);

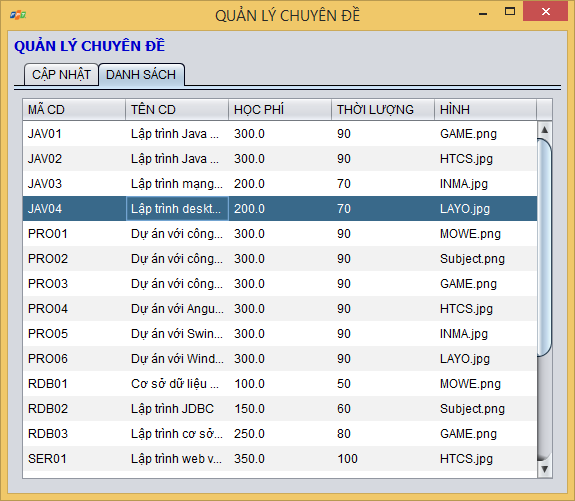
}

#### 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

int index = 0;

ChuyenDeDAO dao = new ChuyenDeDAO();

void init() { setIconImage(ShareHelper.APP\_ICON); setLocationRelativeTo(null);

}

void load() {

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

try {

List<ChuyenDe> list = dao.select(); for (ChuyenDe cd : list) {

Object[] row = {

cd.getMaCD(), cd.getTenCD(), cd.getHocPhi(), cd.getThoiLuong(), cd.getHinh()

};

model.addRow(row);

}

}

catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");

}

}

void insert(){

ChuyenDe model = getModel(); try {

dao.insert(model); this.load();

this.clear();

DialogHelper.alert(this, "Thêm mới thành công!");

}

catch (Exception e) {

DialogHelper.alert(this, "Thêm mới thất bại!");

}

}

void update(){

ChuyenDe model = getModel(); try {

dao.update(model); this.load();

DialogHelper.alert(this, "Cập nhật thành công!");

}

catch (Exception e) {

DialogHelper.alert(this, "Cập nhật thất bại!");

}

}

void delete(){

if(DialogHelper.confirm(this, "Bạn có muốn xóa hay không?")){ String macd = txtMaCD.getText();

try {

dao.delete(macd); this.load();

this.clear();

DialogHelper.alert(this, "Xóa thành công!");

}

catch (Exception e) { DialogHelper.alert(this, "Xóa thất bại!");

}

}

}

void clear(){

this.setModel(new ChuyenDe()); this.setStatus(true);

}

void edit() { try {

String macd = (String) tblGridView.getValueAt(this.index, 0); ChuyenDe model = dao.findById(macd);

if(model != null){ this.setModel(model); this.setStatus(false);

}

}

catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");

}

}

void setModel(ChuyenDe model){

txtMaCD.setText(model.getMaCD()); txtTenCD.setText(model.getTenCD()); txtThoiLuong.setText(String.valueOf(model.getThoiLuong())); txtHocPhi.setText(String.valueOf(model.getHocPhi())); txtMota.setText(model.getMoTa()); lblHinh.setToolTipText(model.getHinh());

if(model.getHinh() != null){ lblHinh.setIcon(ShareHelper.readLogo(model.getHinh()));

}

}

ChuyenDe getModel(){

ChuyenDe model = new ChuyenDe(); model.setMaCD(txtMaCD.getText()); model.setTenCD(txtTenCD.getText()); model.setThoiLuong(Integer.valueOf(txtThoiLuong.getText())); model.setHocPhi(Double.valueOf(txtHocPhi.getText())); model.setHinh(lblHinh.getToolTipText()); model.setMoTa(txtMota.getText());

return model;

}

void setStatus(boolean insertable){ txtMaCD.setEditable(insertable); btnInsert.setEnabled(insertable); btnUpdate.setEnabled(!insertable); btnDelete.setEnabled(!insertable);

boolean first = this.index > 0;

boolean last = this.index < tblGridView.getRowCount() - 1; btnFirst.setEnabled(!insertable && first); btnPrev.setEnabled(!insertable && first); btnLast.setEnabled(!insertable && last); btnNext.setEnabled(!insertable && last);

}

void selectImage() {

if(fileChooser.showOpenDialog(this) == JFileChooser.APPROVE\_OPTION){

File file = fileChooser.getSelectedFile(); if(ShareHelper.saveLogo(file)){

// Hiển thị hình lên form lblHinh.setIcon(ShareHelper.readLogo(file.getName())); lblHinh.setToolTipText(file.getName());

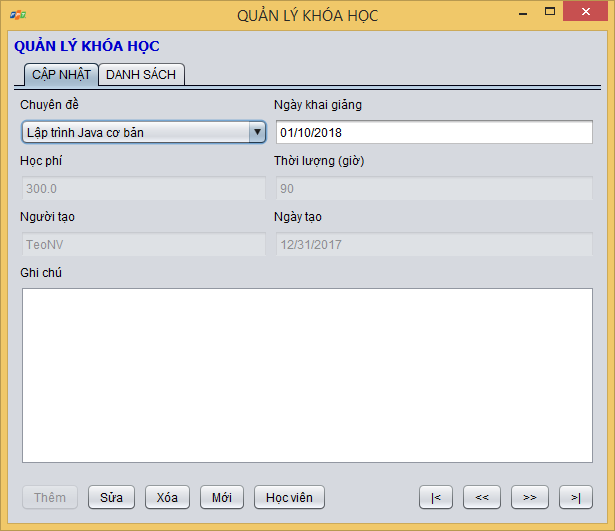
}

}

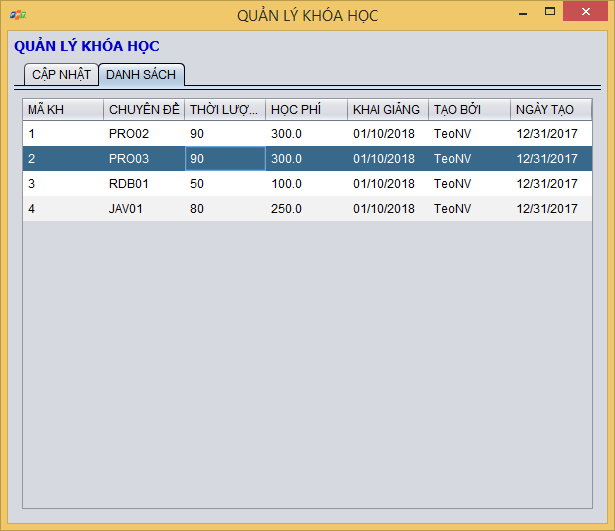
}

#### 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

int index = 0;

KhoaHocDAO dao = new KhoaHocDAO(); ChuyenDeDAO cddao = new ChuyenDeDAO();

void init() { setIconImage(ShareHelper.APP\_ICON); setLocationRelativeTo(null);

}

void load() {

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

try {

List<KhoaHoc> list = dao.select(); for (KhoaHoc kh : list) {

Object[] row = { kh.getMaKH(),

kh.getMaCD(), kh.getThoiLuong(), kh.getHocPhi(),

DateHelper.toString(kh.getNgayKG()), kh.getMaNV(), DateHelper.toString(kh.getNgayTao())

};

model.addRow(row);

}

}

catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");

}

}

void insert(){

KhoaHoc model = getModel(); model.setNgayTao(new Date()); try {

dao.insert(model); this.load();

this.clear();

DialogHelper.alert(this, "Thêm mới thành công!");

}

catch (HeadlessException e) { DialogHelper.alert(this, "Thêm mới thất bại!");

}

}

void update(){

KhoaHoc model = getModel();

try {

dao.update(model); this.load();

DialogHelper.alert(this, "Cập nhật thành công!");

}

catch (Exception e) {

DialogHelper.alert(this, "Cập nhật thất bại!");

}

}

void delete(){

if(DialogHelper.confirm(this, "Bạn thực sự muốn xóa khóa học này?")){ Integer makh = Integer.valueOf(cboChuyenDe.getToolTipText());

try {

dao.delete(makh); this.load();

this.clear();

DialogHelper.alert(this, "Xóa thành công!");

}

catch (Exception e) { DialogHelper.alert(this, "Xóa thất bại!");

}

}

}

void clear(){

KhoaHoc model = new KhoaHoc();

ChuyenDe chuyenDe = (ChuyenDe) cboChuyenDe.getSelectedItem(); model.setMaCD(chuyenDe.getMaCD()); model.setMaNV(ShareHelper.USER.getMaNV()); model.setNgayKG(DateHelper.add(30)); model.setNgayTao(DateHelper.now());

this.setModel(model);

}

void edit() { try {

Integer makh = (Integer) tblGridView.getValueAt(this.index, 0); KhoaHoc model = dao.findById(makh);

if(model != null){ this.setModel(model); this.setStatus(false);

}

}

catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");

}

}

void setModel(KhoaHoc model){ cboChuyenDe.setToolTipText(String.valueOf(model.getMaKH())); cboChuyenDe.setSelectedItem(cddao.findById(model.getMaCD())); txtNgayKG.setText(DateHelper.toString(model.getNgayKG())); txtHocPhi.setText(String.valueOf(model.getHocPhi())); txtThoiLuong.setText(String.valueOf(model.getThoiLuong())); txtMaNV.setText(model.getMaNV()); txtNgayTao.setText(DateHelper.toString(model.getNgayTao())); txtGhiChu.setText(model.getGhiChu());

}

KhoaHoc getModel(){

KhoaHoc model = new KhoaHoc();

ChuyenDe chuyenDe = (ChuyenDe) cboChuyenDe.getSelectedItem(); model.setMaCD(chuyenDe.getMaCD()); model.setNgayKG(DateHelper.toDate(txtNgayKG.getText())); model.setHocPhi(Double.valueOf(txtHocPhi.getText())); model.setThoiLuong(Integer.valueOf(txtThoiLuong.getText())); model.setGhiChu(txtGhiChu.getText()); model.setMaNV(ShareHelper.USER.getMaNV()); model.setNgayTao(DateHelper.toDate(txtNgayTao.getText())); model.setMaKH(Integer.valueOf(cboChuyenDe.getToolTipText()));

return model;

}

void setStatus(boolean insertable){ btnInsert.setEnabled(insertable); btnUpdate.setEnabled(!insertable); btnDelete.setEnabled(!insertable);

boolean first = this.index > 0;

boolean last = this.index < tblGridView.getRowCount() - 1; btnFirst.setEnabled(!insertable && first); btnPrev.setEnabled(!insertable && first); btnLast.setEnabled(!insertable && last); btnNext.setEnabled(!insertable && last);

btnStudents.setVisible(!insertable);

}

void selectComboBox(){

ChuyenDe chuyenDe = (ChuyenDe) cboChuyenDe.getSelectedItem(); txtThoiLuong.setText(String.valueOf(chuyenDe.getThoiLuong())); txtHocPhi.setText(String.valueOf(chuyenDe.getHocPhi()));

}

void openHocVien() {

Integer id = Integer.valueOf(cboChuyenDe.getToolTipText()); new HocVienJFrame(id).setVisible(true);

}

void fillComboBox(){

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

try {

List<ChuyenDe> list = cddao.select(); for(ChuyenDe cd : list){

model.addElement(cd);

}

}

catch (Exception e) {

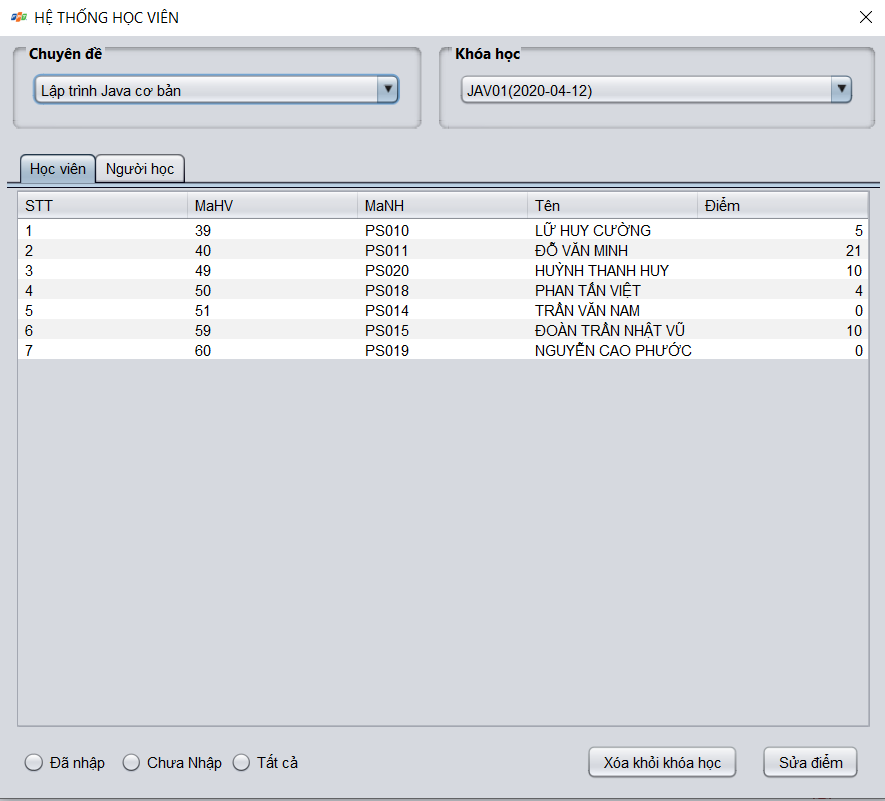
DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");

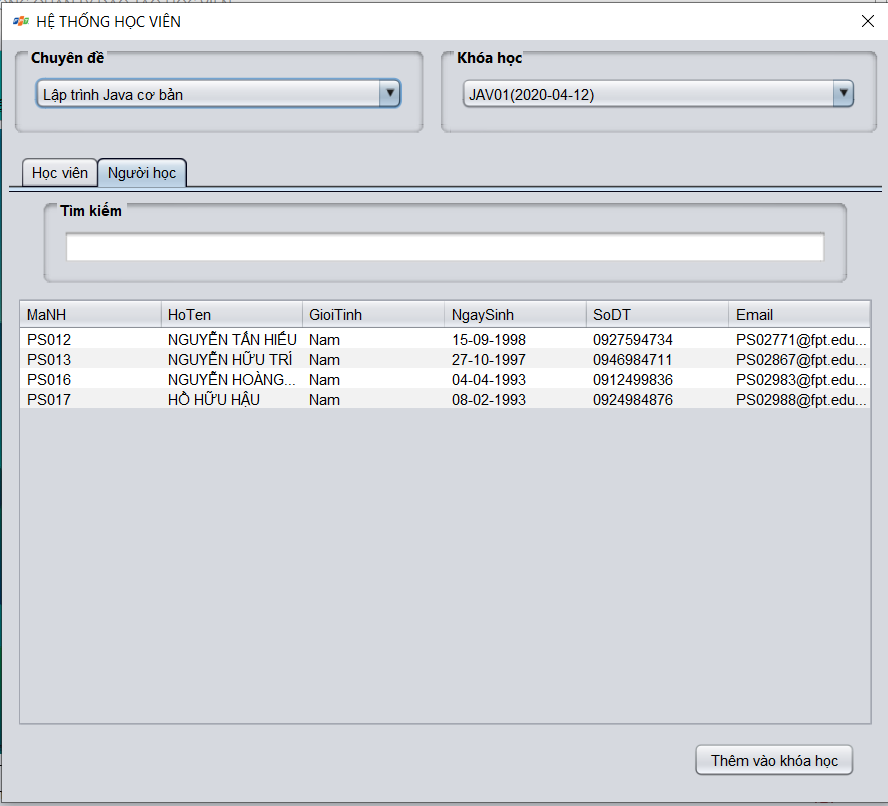
}

}

#### 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

public Integer MaKH;

HocVienDAO dao = new HocVienDAO(); NguoiHocDAO nhdao = new NguoiHocDAO();

void init() { setIconImage(ShareHelper.APP\_ICON); setLocationRelativeTo(null);

}

void fillComboBox() {

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

try {

List<NguoiHoc> list = nhdao.selectByCourse(MaKH); for (NguoiHoc nh : list) {

model.addElement(nh);

}

}

catch (Exception e) {

DialogHelper.alert(this, "Lỗi truy vấn học viên!");

}

}

void fillGridView() {

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

try {

String sql = "SELECT hv.\*, nh.HoTen FROM HocVien hv "

+ " JOIN NguoiHoc nh ON nh.MaNH=hv.MaNH WHERE MaKH=?"; ResultSet rs = JdbcHelper.executeQuery(sql, MaKH);

while (rs.next()) {

double diem = rs.getDouble("Diem"); Object[] row = {

rs.getInt("MaHV"), rs.getString("MaNH"), rs.getString("HoTen"), diem, false

};

if(rdoTatCa.isSelected()){ model.addRow(row);

}

else if(rdoDaNhap.isSelected() && diem >= 0){ model.addRow(row);

}

else if(rdoChuaNhap.isSelected() && diem < 0){ model.addRow(row);

}

}

}

catch (SQLException e) {

DialogHelper.alert(this, "Lỗi truy vấn học viên!");

}

}

void insert() {

NguoiHoc nguoiHoc = (NguoiHoc) cboNguoiHoc.getSelectedItem();

HocVien model = new HocVien(); model.setMaKH(MaKH); model.setMaNH(nguoiHoc.getMaNH()); model.setDiem(Double.valueOf(txtDiem.getText()));

try {

dao.insert(model); this.fillComboBox(); this.fillGridView();

}

catch (Exception e) {

DialogHelper.alert(this, "Lỗi thêm học viên vào khóa học!");

}

}

void update() {

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

Integer mahv = (Integer) tblGridView.getValueAt(i, 0); String manh = (String) tblGridView.getValueAt(i, 1); Double diem = (Double) tblGridView.getValueAt(i, 3); Boolean isDelete = (Boolean) tblGridView.getValueAt(i, 4);

if(isDelete){ dao.delete(mahv);

}

else{

HocVien model = new HocVien(); model.setMaHV(mahv); model.setMaKH(MaKH); model.setMaNH(manh); model.setDiem(diem);

dao.update(model);

}

}

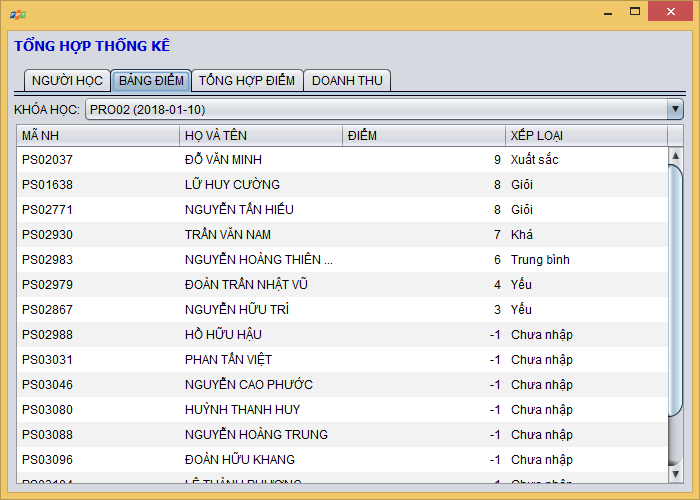
this.fillComboBox(); this.fillGridView();

DialogHelper.alert(this, "Cập nhật thành công!");

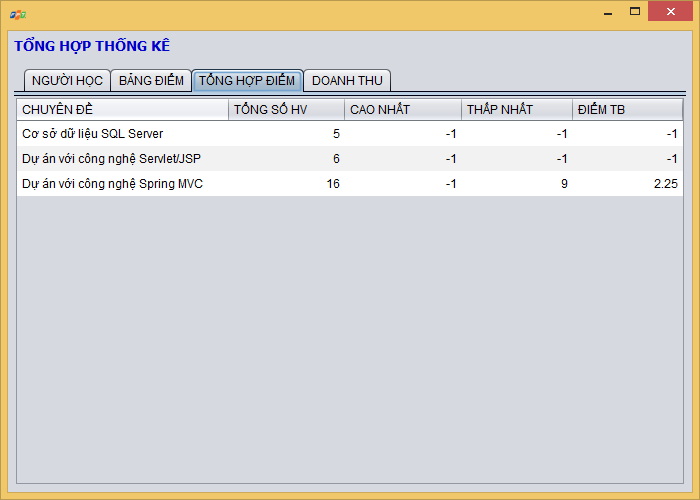
}

#### Thống kê dữ liệu (ThongKeJFrame)

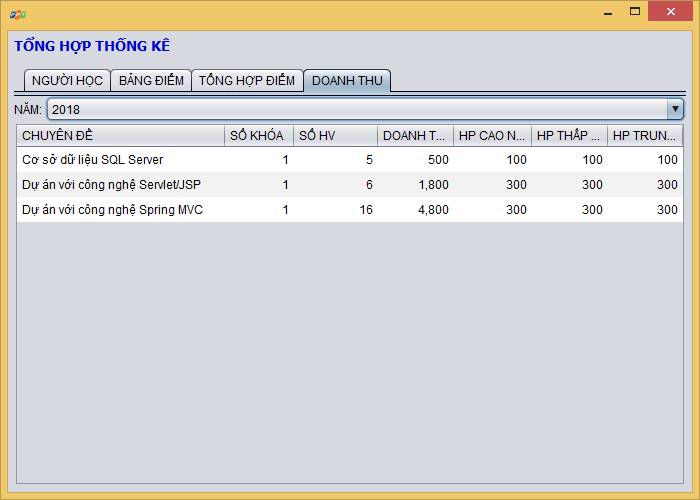
##### Hình: Thống kê số đăng ký từng năm



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 dao = new ThongKeDAO(); KhoaHocDAO khdao = new KhoaHocDAO();

void init() { setIconImage(ShareHelper.APP\_ICON); setLocationRelativeTo(null);

}

void fillComboBoxKhoaHoc(){

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

List<KhoaHoc> list = khdao.select(); for(KhoaHoc kh : list){

model.addElement(kh);

}

cboKhoaHoc.setSelectedIndex(0);

}

void fillComboBoxNam(){

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

List<KhoaHoc> list = khdao.select(); for(KhoaHoc kh : list){

int nam = kh.getNgayKG().getYear() + 1900; if(model.getIndexOf(nam) < 0){

model.addElement(nam);

}

}

cboNam.setSelectedIndex(0);

}

void fillTableBangDiem(){

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

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

model.addRow(row);

}

}

void fillTableNguoiHoc(){

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

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

model.addRow(row);

}

}

void fillTableKhoaHoc() {

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

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

model.addRow(row);

}

}

void fillTableDoanhThu() {

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

int nam = Integer.parseInt(cboNam.getSelectedItem().toString()); List<Object[]> list = dao.getDoanhThu(nam);

for(Object[] row : list){ model.addRow(row);

}

}

#### 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 | |  |

### KIỂM THỬ FORM QUẢN LÝ NGƯỜI HỌC

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Thao tác** | |
| **Thêm mới** | **Cập nhật** |
| Mã người học | Đúng 7 ký tự 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 TRIỂN KHAI

### 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 Polypro bằng cách chạy file Polypro.sql

##### Cài đặt JDK 1.8 trở lên

* + - Chạy file setup

### HƯỚNG DẪN SỬ DỤNG PHẦN MỀM

##### Đăng nhập với tài khoản username: admin và password:admin

* + - Cấp các tài khoản có vai trò là trưởng phòng

##### Đăng nhập tài khoản mới và xóa tài khoản admin

* + - Xem hướng dȁn sử dụng trong menu trợ giúp của phần mềm