



|  |
| --- |
| **DOCUMENT** |
| PHẦN MỀM:  QUẢN LÝ ĐÀO TẠO – CNTT (UDPM) |
|  |
| **Thực hiện: Hồ Hoàng Phú - PS17588**  **Lớp: IT16312**  **Giảng Viên Hướng Dẫn: Thầy Lê Văn Phụng** |
|  |

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.

# 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**

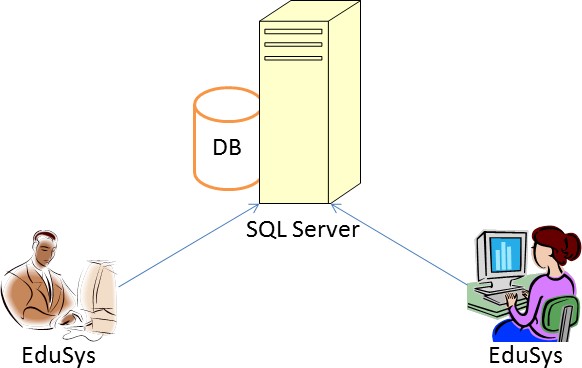
##### Người học

* + - * + 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.



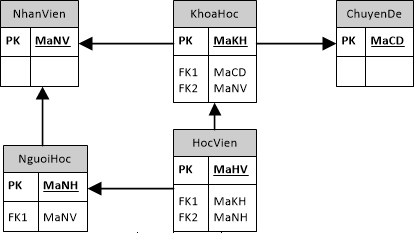
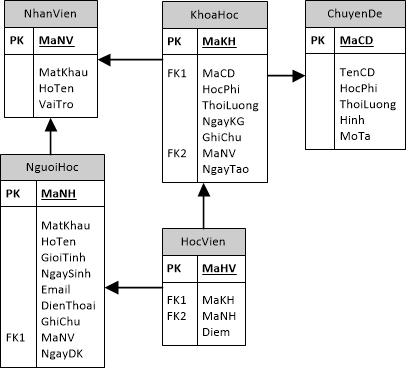
* + - Cần một máy cài SQL Server 2008+. Máy này cần hệ điều hành window xp trở lên
    - Các máy nhân viên phòng đào tạo cài phần mềm EduVN. Các máy nhân viên cần JDK 1.8+ với hệ điều hành bất ký
  1. **THIẾT KẾ CSDL**

## 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

* + - 1. ***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 |
| Email | NVARCHAR(50) | NOT NULL | Email của nhân viên |
| QuenMK | NVARCHAR(15) |  | Code khi thực hiện Quên MK |

* + - 1. ***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 |

* + - 1. ***Set up Mail***

Bảng Mailer lưu thông tin địa chỉ Email của hệ thống dùng để thao tác gửi mail có cấu trúc như sau.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| UsernameMailer | VARCHAR(50) | FK, NOT NULL | Mã khóa học |
| PasswordMailer | VARCHAR(255) | NOT NULL | Mã chuyên đề |

* + - 1. ***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 |
| Hinh | NVARCHAR(255) | DEFAULT ‘default.png’ | Hình đại diện |
| 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 |

* + - 1. ***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*

* + - 1. ***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 |

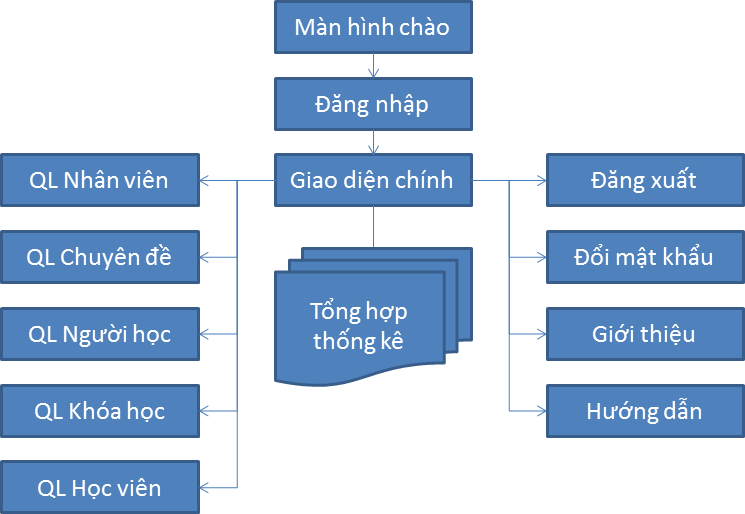
### THIẾT KẾ GIAO DIỆN

## 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.

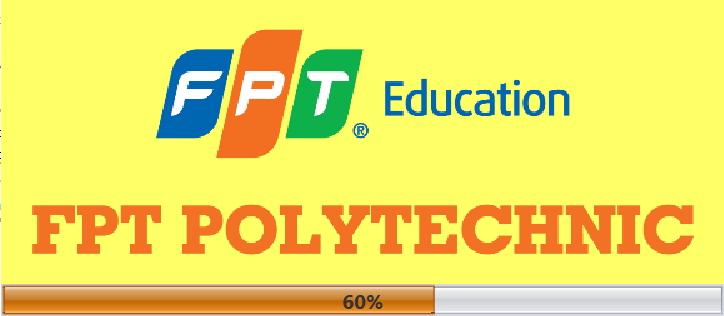
Lưu ý quan trọng:

* + - * 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 |

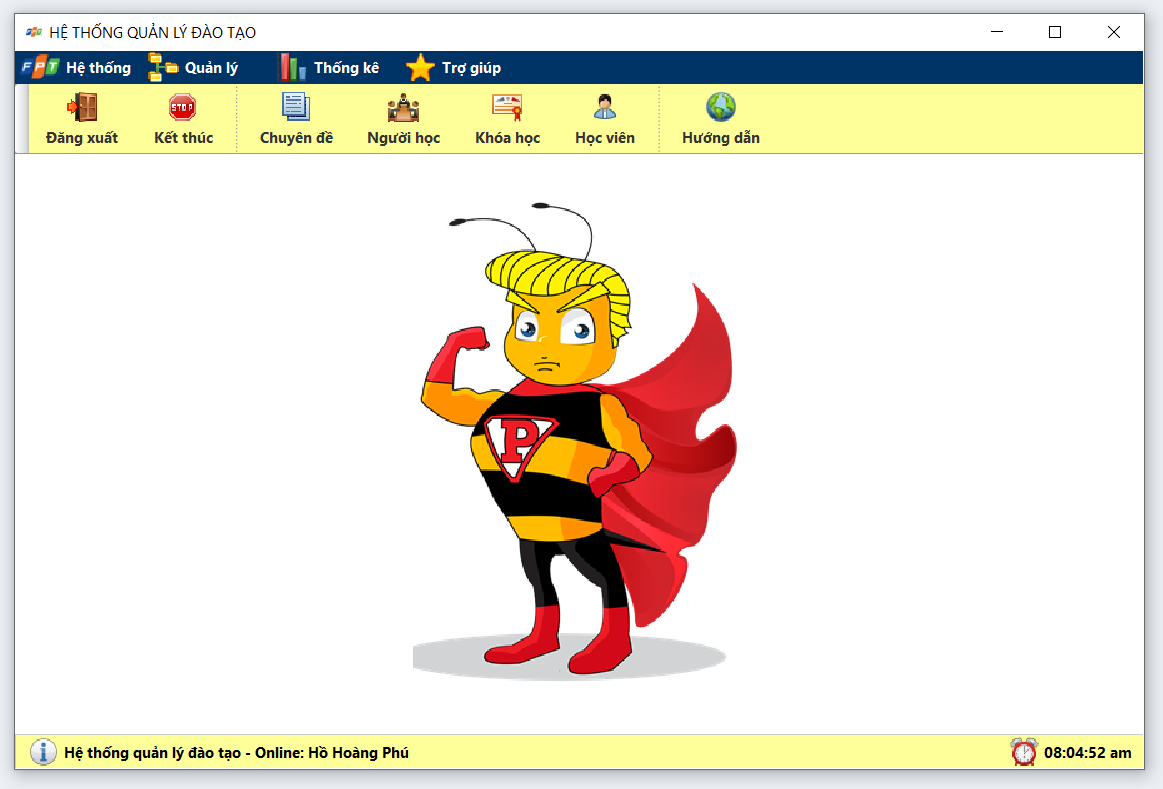
* + - 1. ***Đăng nhập***



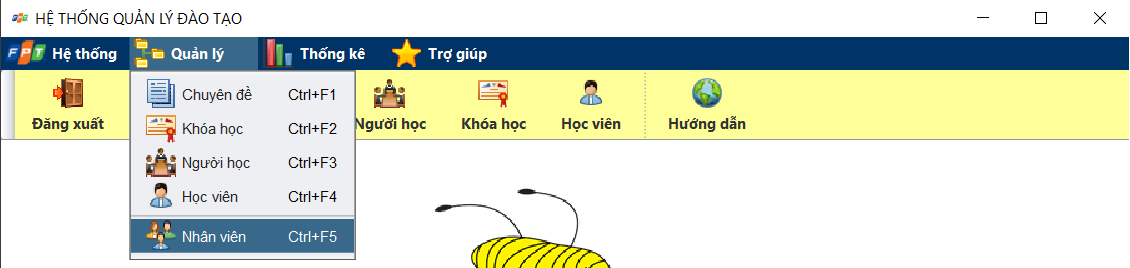
Các thành phần giao diện

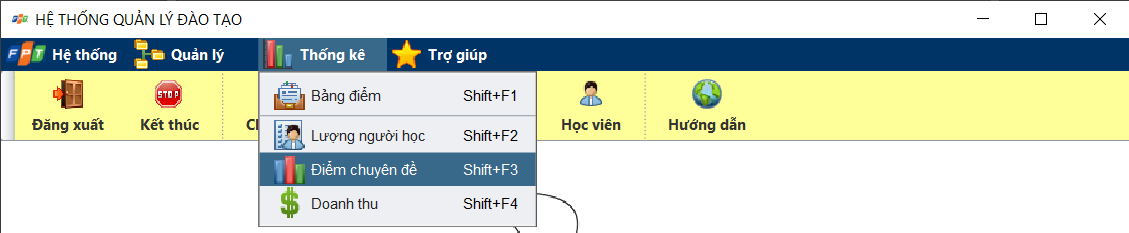
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Componel** | **Kiểu** | **Thuộc tính** |
| 1 | DangNhapJDialog | JDialog | title:HỆ THỐNG QUẢN LÝ ĐÀO TẠO  Layout: BorderLayout defaultCloseOperation:DO\_NOTHING |
| 2 | lblTitle | JLabel | Text: HỆ THỐNG QUẢN LÝ ĐÀO TẠO  Font:Tahoma, Bold, 18  Foreground: [0,102,51] |
| 3 | lblImage | JLabel | Icon: fpt.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 |
| 11 | btnQuenMayKhau | JButton | Text:Quên mật khẩu |

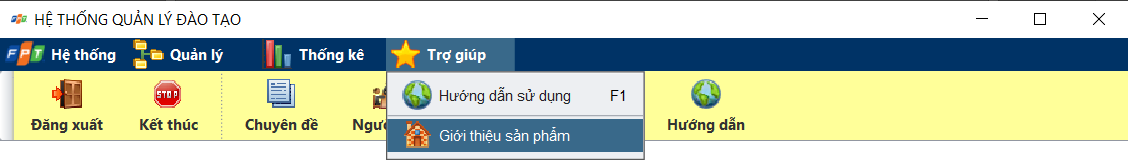
* + - 1. ***Giao diện chính***

******

******

****

****

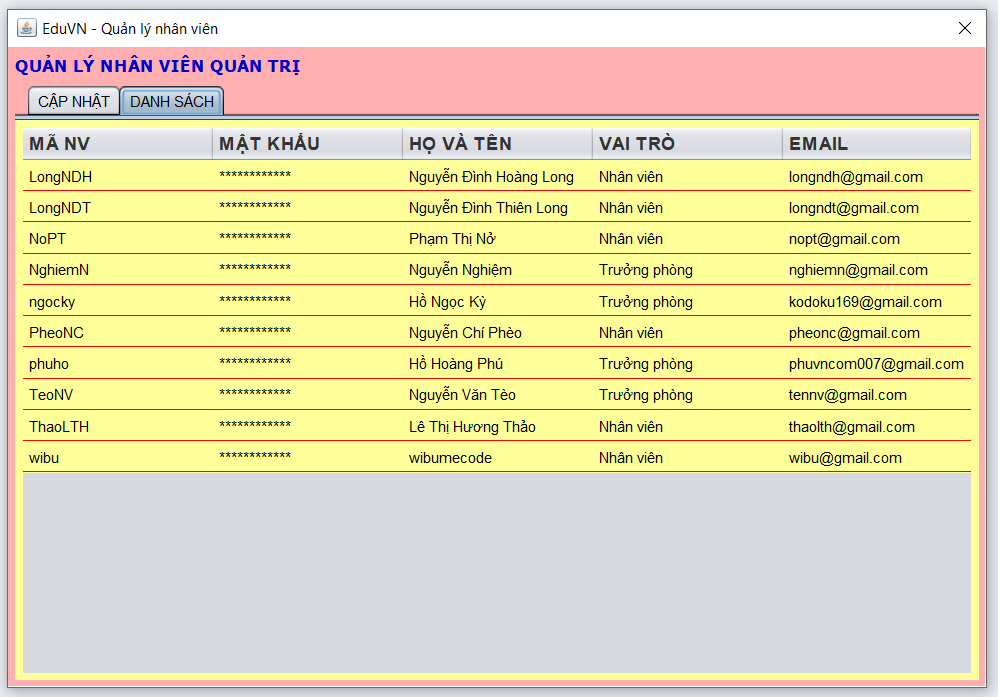
****

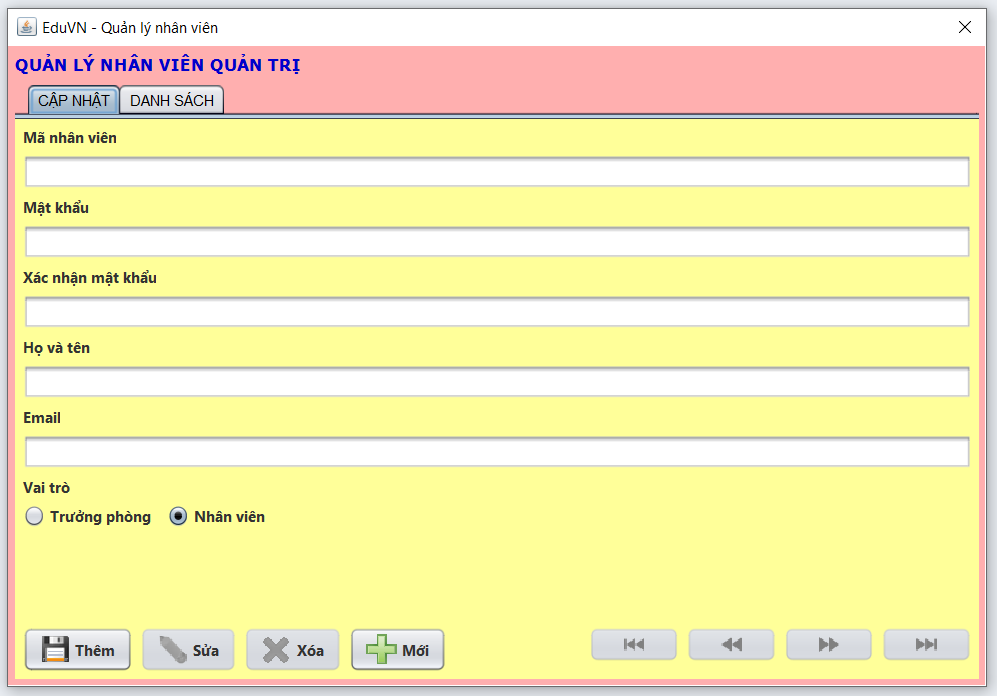
Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Componel** | **Kiểu** | **Thuộc tính** |
| 1 | MainJFrame | JFrame | Title: HỆ THỐNG QUẢN LÝ ĐÀO TẠO  Layout:BorderLayout |
| 2 | mnuHeThong | JMenu | Text:Hệ thống |
| 3 | mniSetup | JMenuItem | Text:Set up Email |
| 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 dẫn 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 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: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 |

* + - 1. ***Quản lý nhân viên***



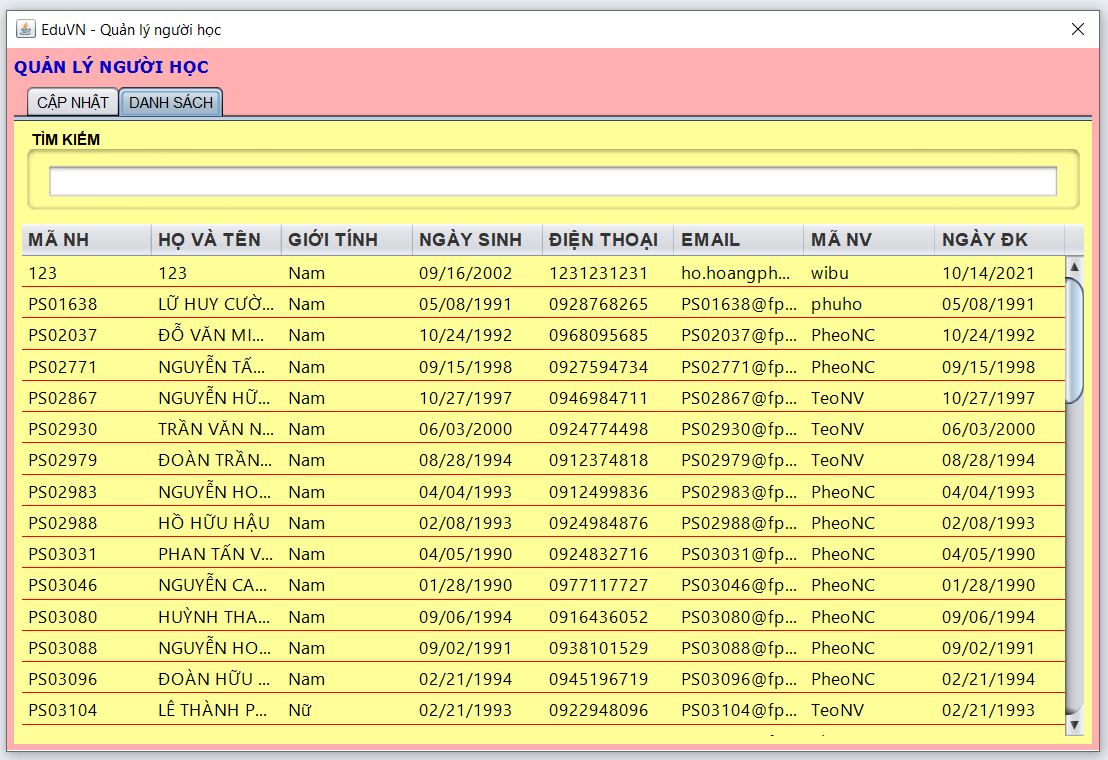


Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | NhanVienJDialog | JDialog | 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 | lblEmail | JLabel | Text: Email |
| 15 | txtEmail | JTextField |  |
| 16 | lblVaiTro | JLabel | Text: Vai trò |
| 17 | bgrVaiTro | ButtonGroup |  |
| 18 | rdoTruongPhong | JRadioButton | Text: Trưởng phòng  Selected: true buttonGroup: bgrVaiTro |
| 19 | rdoNhanVien |  | Text: Nhân viên buttonGroup: bgrVaiTro |
| 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 |

* + - 1. ***Quản lý người học***



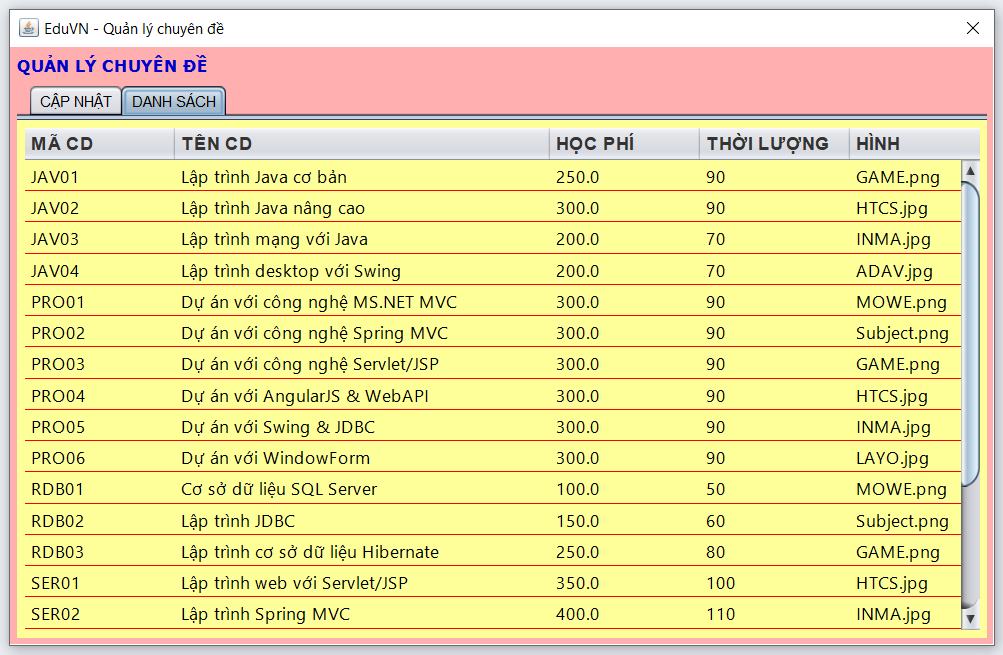


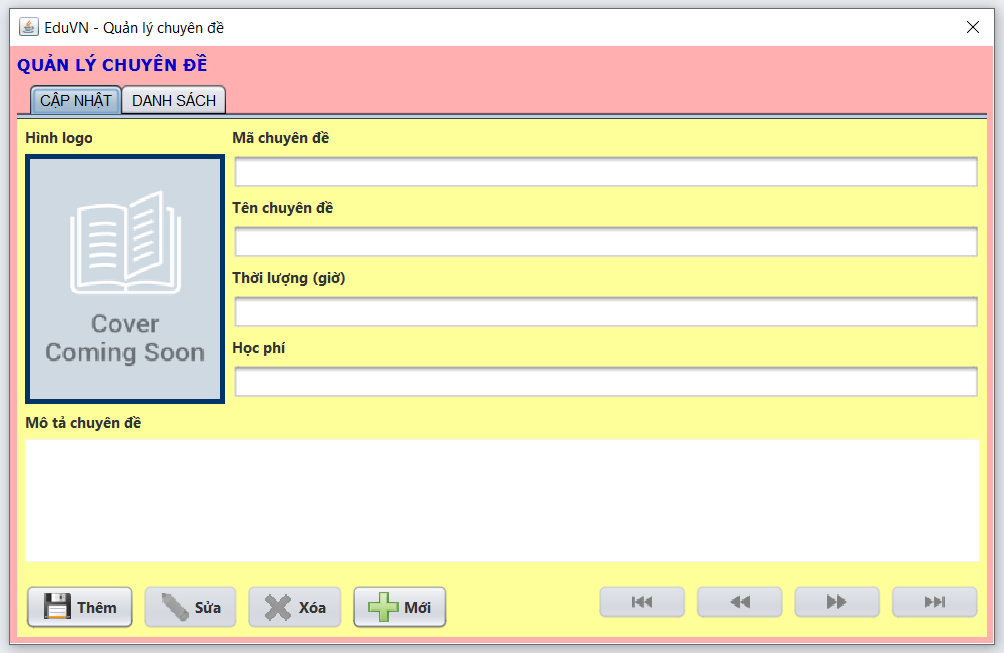
Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | NguoiHocJDialog | JDialog | 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 |
| 32 | lblHinh | Jlabel | Hình ảnh đại diện |

* + - 1. ***Quản lý chuyên đề***



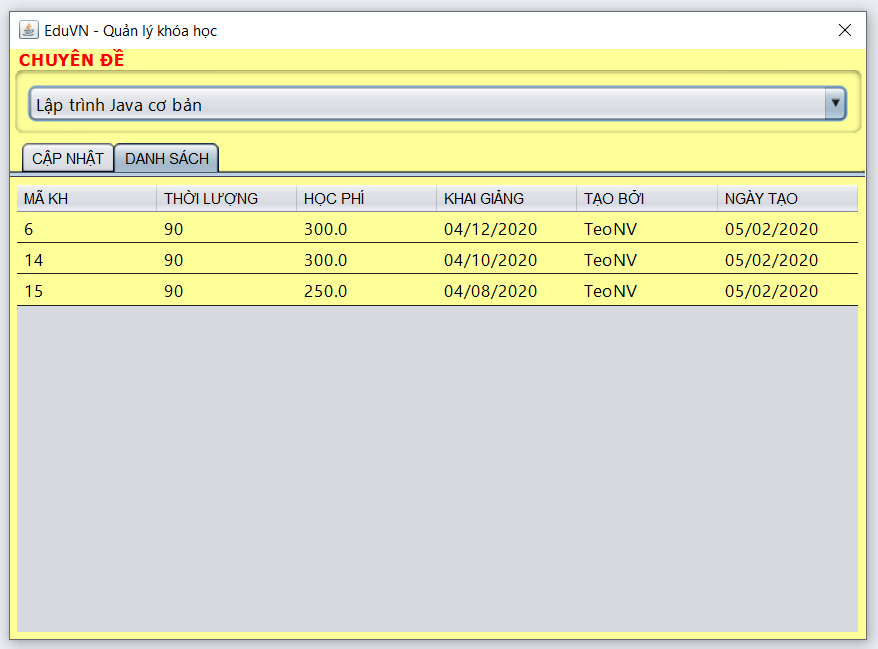


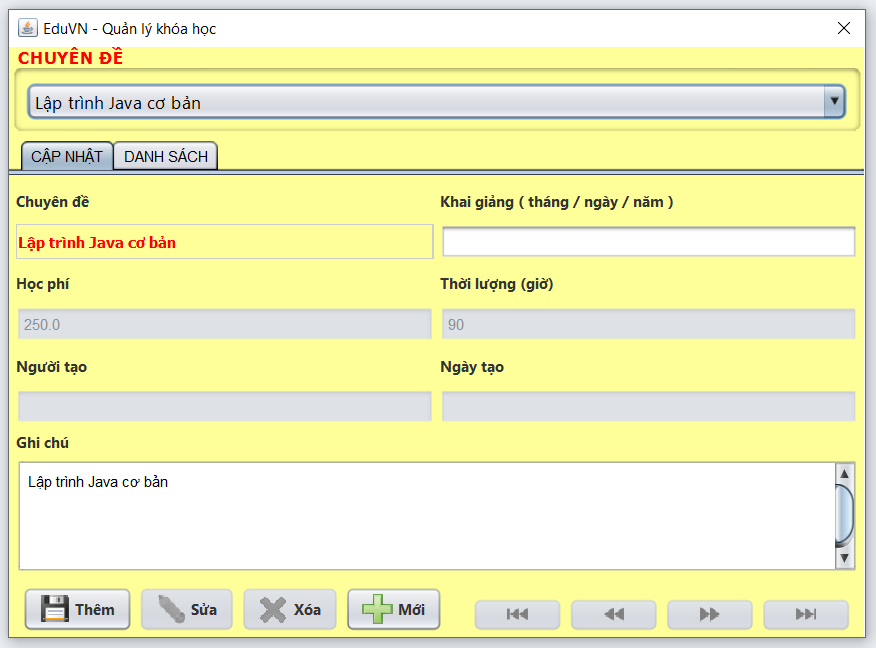
Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | ChuyenDeJDialog | JDialog | 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 |

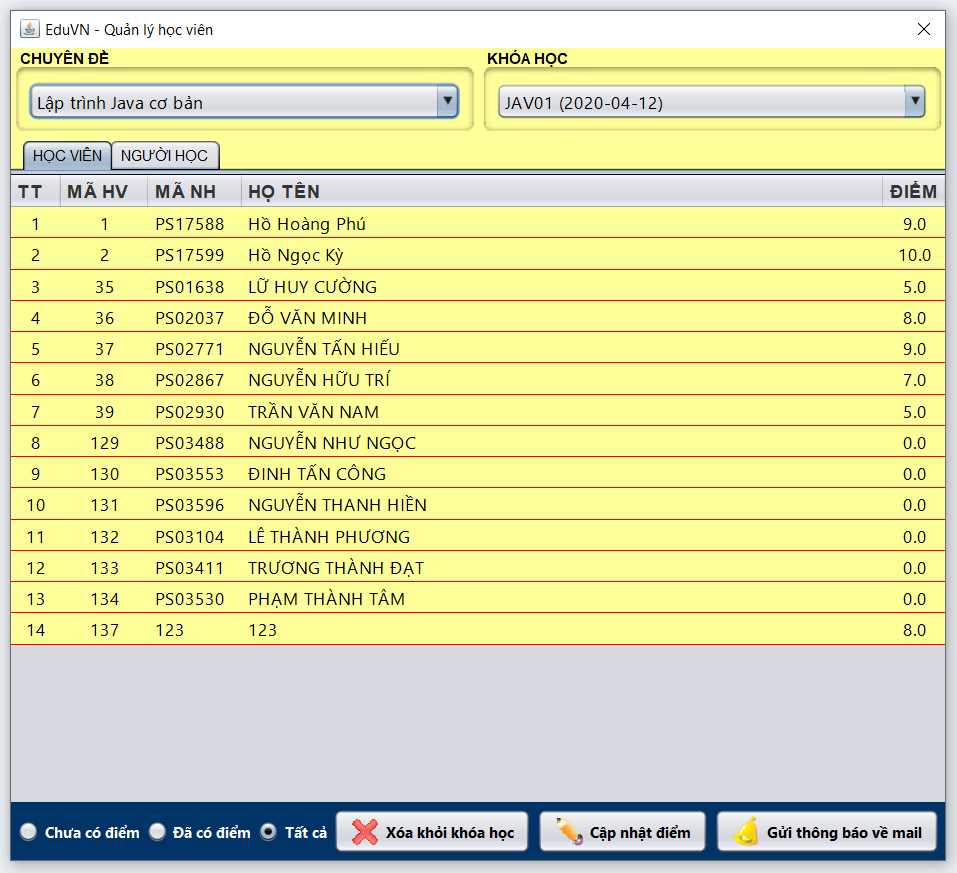
* + - 1. ***Quản lý khóa học***

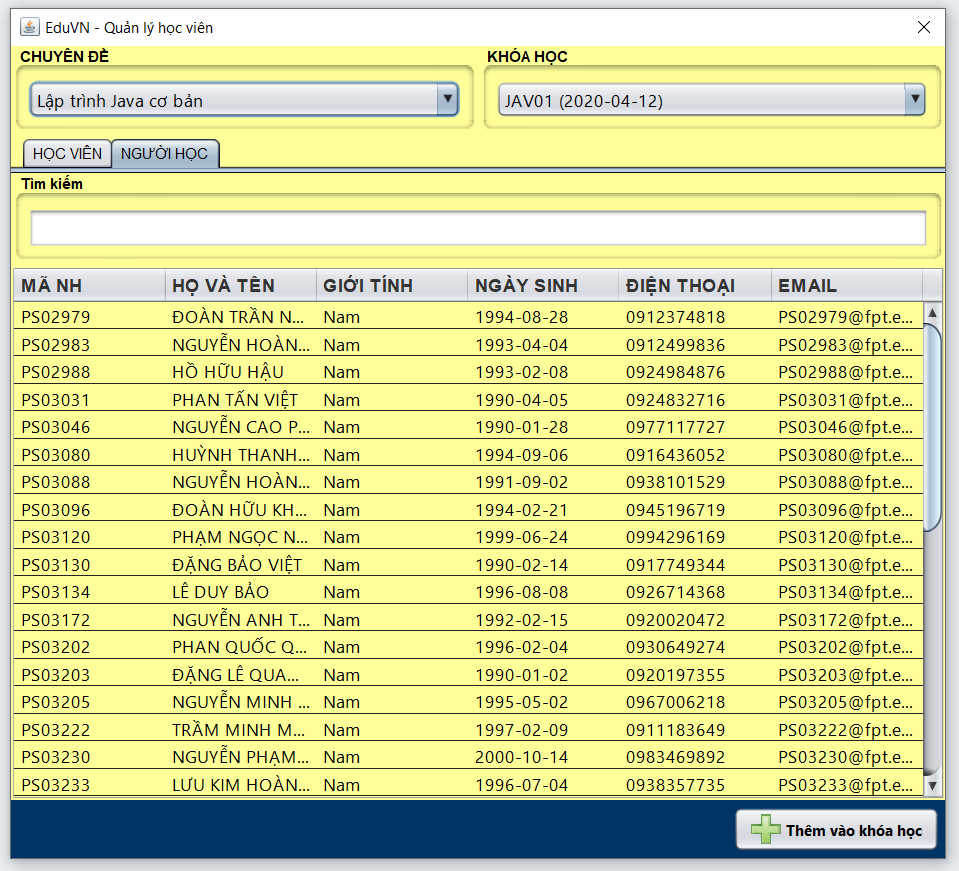




Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | KhoaHocJDialog | JDialog | 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 |

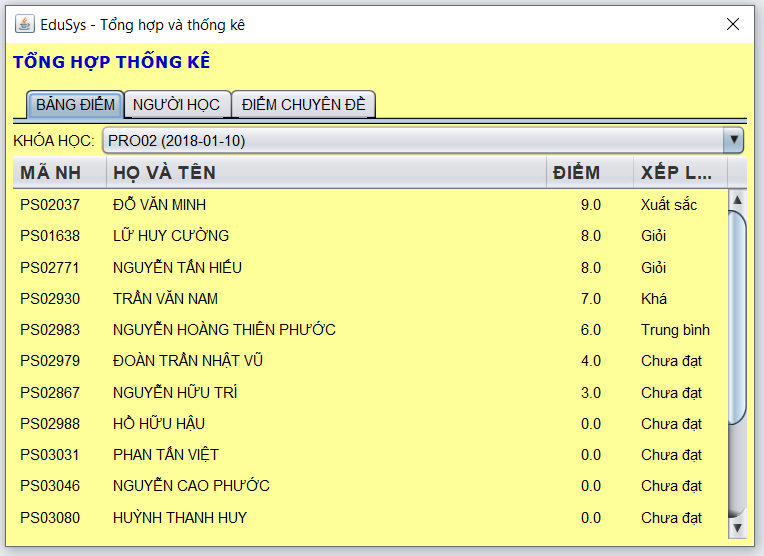
* + - 1. ***Quản lý học viên***

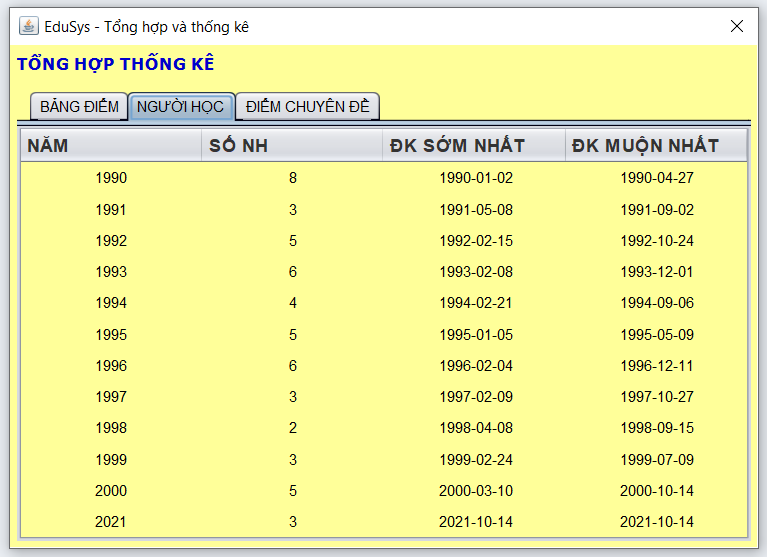


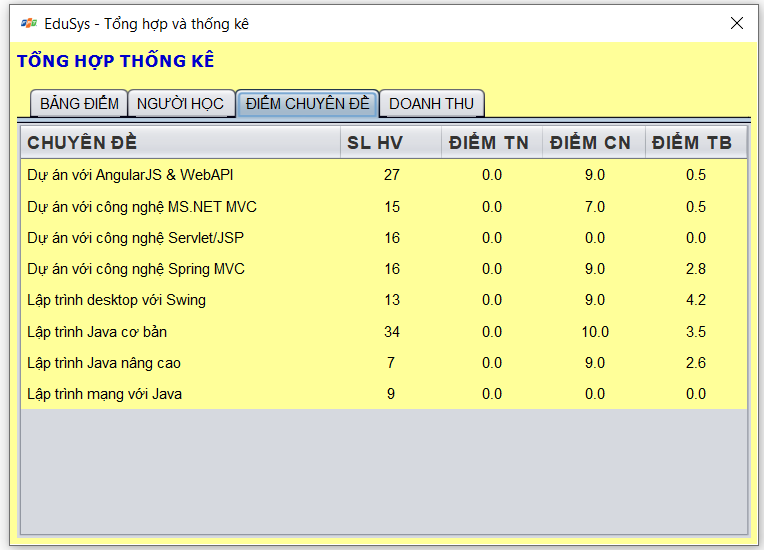
* + - 1. Các thành phần giao diện

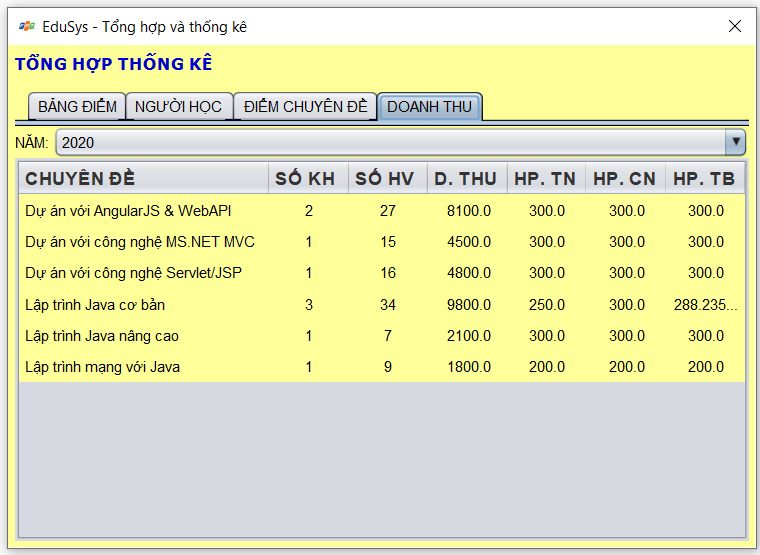
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | HocVienJDialog | JDialog | Title: QUẢN LÝ HỌC VIÊN  defaultCloseOperation:DISPONSE |
| 2 | pnlNguoiHoc | JPanel | Border: TitledBorder(Title: Người Học) |
| 3 | pnlHocVien | JPanel | Border: TitledBorder(Title: HỌC VIÊN) |
| 4 | cboKhoaHoc | JComboBox |  |
| 5 | cboChuyenDe | JComboBox |  |
| 6 | tableHocVien | JTable |  |
| 7 | tableNguoiHoc | JTable |  |
| 8 | rdoCCD | JRadio | Text: Chưa có điểm |
| 9 | rdoDCD | JRadio | Text: Đã có điểm |
| 10 | rdoAll | JRadio | Text: Tất cả |
| 11 | btnRemove | JButton | Text: Xóa khỏi khóa học |
| 12 | btnUpdate | JButton | Text: Cập nhật điểm |
| 13 | btnGuiMail | JButton | Text: Gửi thông báo về mail |
| 14 | btnAddHocVien | JButton | Text: Thêm vào khóa học |

* + - 1. ***Tổng hợp thống kê***





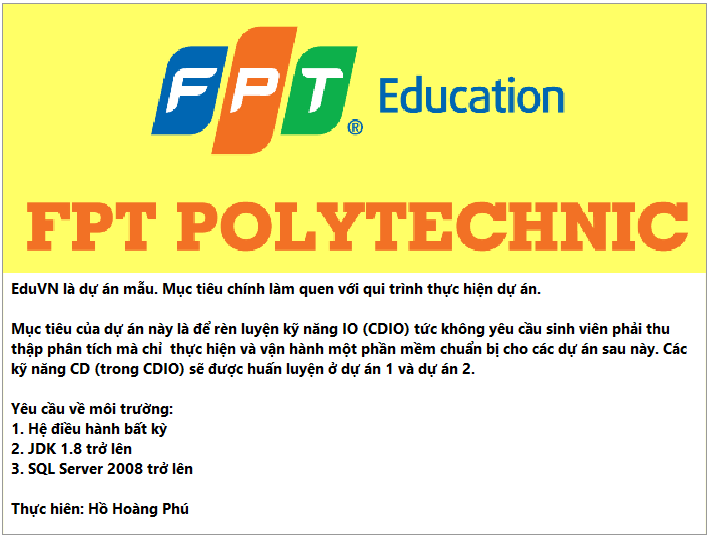




Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | ThongKeJFrame | JFrame | Title: TỔNG HỢP VÀ THỐNG KÊ  defaultCloseOperation:DISPONSE |
| 2 | lblTitle | JLabel | Text: TỔNG HỢP VÀ 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 |

* + - 1. ***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 |

# THỰC HIỆN VIẾT MÃ

* 1. **VIẾT MÃ TẠO CSDL**

## Tạo CSDL

CREATE DATABASE DuAnMau

GO

USE DuAnMau

GO

CREATE TABLE Mailer(

UsernameMailer varchar(50) NOT NULL,

PasswordMailer varchar(255) NOT NULL,

PRIMARY KEY(UsernameMailer)

)

GO

CREATE TABLE NhanVien(

MaNV nvarchar(50) NOT NULL,

MatKhau nvarchar(32) NOT NULL,

HoTen nvarchar(50) NOT NULL,

VaiTro bit NOT NULL DEFAULT 0,

Email nvarchar(50) NOT NULL,

QuenMK nvarchar(15),

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(255) NOT NULL DEFAULT 'chuyen-de.png',

MoTa nvarchar(255) NOT NULL,

PRIMARY KEY(MaCD),

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,

Hinh nvarchar(255) NOT NULL DEFAULT 'default.png',

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),

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

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

)

GO

## SQL truy vấn và thao tác bảng

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

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

IF OBJECT\_ID('sp\_LuongNguoiHoc') IS NOT NULL

DROP PROC sp\_LuongNguoiHoc

GO

CREATE PROC sp\_LuongNguoiHoc AS

BEGIN

SELECT

YEAR(NgayDK) Nam,

COUNT(\*) SoLuong,

MIN(NgayDK) DauTien,

MAX(NgayDK) CuoiCung

FROM NguoiHoc

GROUP BY YEAR(NgayDK)

END

***Doanh thu theo chuyên đề***

IF OBJECT\_ID('sp\_DoanhThu') IS NOT NULL

DROP PROC sp\_DoanhThu

GO

CREATE PROC sp\_DoanhThu(@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***

IF OBJECT\_ID('sp\_DiemChuyenDe') IS NOT NULL

DROP PROC sp\_DiemChuyenDe

GO

CREATE PROC sp\_DiemChuyenDe 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***

IF OBJECT\_ID('sp\_BangDiem') IS NOT NULL

DROP PROC sp\_BangDiem

GO

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 JdbcHelper***

##### 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.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class XJdbc {

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

private static String dburl = "jdbc:sqlserver://localhost;databaseName=DuAnMau";

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 getStmt(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 update(String sql, Object... args) {

try {

PreparedStatement stmt = XJdbc.getStmt(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 query(String sql, Object... args) {

try {

PreparedStatement stmt = XJdbc.getStmt(sql, args);

return stmt.executeQuery();

} catch (SQLException e) {

e.printStackTrace();

throw new RuntimeException(e);

}

}

public static Object value(String sql, Object... args) {

try {

ResultSet rs = XJdbc.query(sql, args);

if (rs.next()) {

return rs.getObject(0);

}

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

return null;

} catch (Exception e) {

throw new RuntimeException(e);

}

}

/\*\*

\* Thực hiện kiểm tra việc kết nối tới database name

\*

\*

\* @return Trả về "OK" nếu kết nối thành công. Hoặc trả về thông báo lỗi nếu kết nối thất bại

\*/

public static String checkConnectSQL() {

try {

Connection connection = DriverManager.getConnection(dburl, username, password);

} catch (SQLException ex) {

String[] a = ex.getMessage().split("\\.");

return a[0];

}

return "OK";

}

}

* + - 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.edusys.utils;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

public class XDate {

static SimpleDateFormat formatDate = new SimpleDateFormat();

/\*\*

\* Chuyển đổi String sang Date

\*

\* @param date là String cần chuyển

\* @param pattern là định dạng thời gian

\*/

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

try {

formatDate.applyPattern(pattern);

return formatDate.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

\*/

public static String toString(Date date, String pattern) {

formatDate.applyPattern(pattern);

return formatDate.format(date);

}

/\*\*

\* Bổ sung số ngày vào thời gian

\*

\* @param date thời gian hiện có

\* @param days số ngày cần bổ sung vào date

\* @return Date kết quả

\*/

public static Date addDays(Date date, long days) {

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

return date;

}

}

* + - 1. ***Lớp tiện ích MsgBox***

Chứa các phương thức Joptionpane giúp lập trình sau này khi xử lý thời gian dễ dàng hơn.

package com.edusys.utils;

import java.awt.Component;

import javax.swing.JOptionPane;

public class MsgBox {

/\*\*

\* 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,

"Message", 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,

"Message",

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,

"Message", 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.

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

package com.edusys.entity;

public class NhanVien {

private String maNV;

private String matKhau;

private String hoTen;

private String Email;

private String VerifyQMK;

private boolean vaiTro = false;

//--getter/setter

}

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

package com.edusys.entity;

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;

}

public ChuyenDe(String maCD, String tenCD, double hocPhi, int thoiLuong, String hinh, String moTa) {

this.maCD = maCD;

this.tenCD = tenCD;

this.hocPhi = hocPhi;

this.thoiLuong = thoiLuong;

this.hinh = hinh;

this.moTa = moTa;

}

//--getter/setter

}

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

package com.edusys.entity;

import java.util.Date;

public class NguoiHoc {

private String maNH;

private String hoTen;

private Date ngaySinh = new Date();

private boolean gioiTinh;

private String dienThoai;

private String email;

private String hinh;

private String ghiChu;

private String maNV;

private Date ngayDK = new Date();

//--getter/setter

}

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

package com.edusys.entity;

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 = new Date();

@Override

public String toString() {

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

}

//--getter/setter

}

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

package com.edusys.entity;

public class HocVien {

private int maHV;

private int maKH;

private String maNH;

private double diem;

//--getter/setter

}

## 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.edusys.dao;

import com.edusys.utils.XJdbc;

import com.edusys.entity.NhanVien;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

public class NhanVienDAO extends EduVNDAO<NhanVien, String> {

@Override

public void insert(NhanVien model) {

String sql = "INSERT INTO NhanVien (MaNV, MatKhau, HoTen, VaiTro, Email, QuenMK) VALUES (?, ?, ?, ?, ?, ?)";

XJdbc.update(sql,

model.getMaNV(),

model.getMatKhau(),

model.getHoTen(),

model.getVaiTro(),

model.getEmail(),

model.getVerifyQMK());

}

@Override

public void update(NhanVien model) {

String sql = "UPDATE NhanVien SET MatKhau=?, HoTen=?, VaiTro=?, Email=?, QuenMK=? WHERE MaNV=?";

XJdbc.update(sql,

model.getMatKhau(),

model.getHoTen(),

model.getVaiTro(),

model.getEmail(),

model.getVerifyQMK(),

model.getMaNV());

}

@Override

public void delete(String MaNV) {

String sql = "DELETE FROM NhanVien WHERE MaNV=?";

XJdbc.update(sql, MaNV);

}

@Override

public List<NhanVien> selectAll() {

String sql = "SELECT \* FROM NhanVien";

return this.selectBySql(sql);

}

@Override

public NhanVien selectById(String manv) {

String sql = "SELECT \* FROM NhanVien WHERE MaNV=?";

List<NhanVien> list = this.selectBySql(sql, manv);

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

}

@Override

protected List<NhanVien> selectBySql(String sql, Object... args) {

List<NhanVien> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = XJdbc.query(sql, args);

while (rs.next()) {

NhanVien entity = new NhanVien();

entity.setMaNV(rs.getString("MaNV"));

entity.setMatKhau(rs.getString("MatKhau"));

entity.setHoTen(rs.getString("HoTen"));

entity.setVaiTro(rs.getBoolean("VaiTro"));

entity.setEmail(rs.getString("Email"));

entity.setVerifyQMK(rs.getString("QuenMK"));

list.add(entity);

}

} finally {

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

}

} catch (SQLException ex) {

ex.printStackTrace();

throw new RuntimeException(ex);

}

return list;

}

}

* + - 1. ***ChuyenDeDAO***

package com.edusys.dao;

import com.edusys.utils.XJdbc;

import com.edusys.entity.ChuyenDe;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

public class ChuyenDeDAO extends EduVNDAO<ChuyenDe, String> {

@Override

public void insert(ChuyenDe model) {

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

XJdbc.update(sql,

model.getMaCD(),

model.getTenCD(),

model.getHocPhi(),

model.getThoiLuong(),

model.getHinh(),

model.getMoTa());

}

@Override

public void update(ChuyenDe model) {

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

XJdbc.update(sql,

model.getTenCD(),

model.getHocPhi(),

model.getThoiLuong(),

model.getHinh(),

model.getMoTa(),

model.getMaCD());

}

@Override

public void delete(String MaCD) {

String sql = "DELETE FROM ChuyenDe WHERE MaCD=?";

XJdbc.update(sql, MaCD);

}

@Override

public List<ChuyenDe> selectAll() {

String sql = "SELECT \* FROM ChuyenDe";

return selectBySql(sql);

}

@Override

public ChuyenDe selectById(String macd) {

String sql = "SELECT \* FROM ChuyenDe WHERE MaCD=?";

List<ChuyenDe> list = selectBySql(sql, macd);

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

}

@Override

protected List<ChuyenDe> selectBySql(String sql, Object... args) {

List<ChuyenDe> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = XJdbc.query(sql, args);

while (rs.next()) {

ChuyenDe entity = new ChuyenDe();

entity.setMaCD(rs.getString("MaCD"));

entity.setHinh(rs.getString("Hinh"));

entity.setHocPhi(rs.getDouble("HocPhi"));

entity.setMoTa(rs.getString("MoTa"));

entity.setTenCD(rs.getString("TenCD"));

entity.setThoiLuong(rs.getInt("ThoiLuong"));

list.add(entity);

}

} finally {

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

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

}

* + - 1. ***NguoiHocDAO***

package com.edusys.dao;

import com.edusys.utils.XJdbc;

import com.edusys.entity.NguoiHoc;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

public class NguoiHocDAO extends EduVNDAO<NguoiHoc, String> {

@Override

public void insert(NguoiHoc model) {

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

XJdbc.update(sql,

model.getMaNH(),

model.getHoTen(),

model.getNgaySinh(),

model.getGioiTinh(),

model.getDienThoai(),

model.getEmail(),

model.getHinh(),

model.getGhiChu(),

model.getMaNV());

}

@Override

public void update(NguoiHoc model) {

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

XJdbc.update(sql,

model.getHoTen(),

model.getNgaySinh(),

model.getGioiTinh(),

model.getDienThoai(),

model.getEmail(),

model.getHinh(),

model.getGhiChu(),

model.getMaNV(),

model.getMaNH());

}

@Override

public void delete(String id) {

String sql = "DELETE FROM NguoiHoc WHERE MaNH=?";

XJdbc.update(sql, id);

}

@Override

public List<NguoiHoc> selectAll() {

String sql = "SELECT \* FROM NguoiHoc";

return selectBySql(sql);

}

@Override

public NguoiHoc selectById(String manh) {

String sql = "SELECT \* FROM NguoiHoc WHERE MaNH=?";

List<NguoiHoc> list = selectBySql(sql, manh);

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

}

@Override

protected List<NguoiHoc> selectBySql(String sql, Object... args) {

List<NguoiHoc> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = XJdbc.query(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.setHinh(rs.getString("Hinh"));

entity.setGhiChu(rs.getString("GhiChu"));

entity.setMaNV(rs.getString("MaNV"));

entity.setNgayDK(rs.getDate("NgayDK"));

list.add(entity);

}

} finally {

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

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

public List<NguoiHoc> selectByKeyword(String keyword) {

String sql = "SELECT \* FROM NguoiHoc WHERE HoTen + MaNH + DienThoai + Email LIKE ?";

return this.selectBySql(sql, "%" + keyword + "%");

}

public List<NguoiHoc> selectNotInCourse(int makh, String keyword) {

String sql = "SELECT \* FROM NguoiHoc "

+ " WHERE HoTen + MaNH + DienThoai + Email LIKE ? AND "

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

return this.selectBySql(sql, "%" + keyword + "%", makh);

}

}

* + - 1. ***KhoaHocDAO***

package com.edusys.dao;

import com.edusys.utils.XJdbc;

import com.edusys.entity.KhoaHoc;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

public class KhoaHocDAO extends EduVNDAO<KhoaHoc, Integer> {

@Override

public void insert(KhoaHoc model) {

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

XJdbc.update(sql,

model.getMaCD(),

model.getHocPhi(),

model.getThoiLuong(),

model.getNgayKG(),

model.getGhiChu(),

model.getMaNV());

}

@Override

public void update(KhoaHoc model) {

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

XJdbc.update(sql,

model.getMaCD(),

model.getHocPhi(),

model.getThoiLuong(),

model.getNgayKG(),

model.getGhiChu(),

model.getMaNV(),

model.getMaKH());

}

@Override

public void delete(Integer MaKH) {

String sql = "DELETE FROM KhoaHoc WHERE MaKH=?";

XJdbc.update(sql, MaKH);

}

@Override

public List<KhoaHoc> selectAll() {

String sql = "SELECT \* FROM KhoaHoc";

return selectBySql(sql);

}

@Override

public KhoaHoc selectById(Integer makh) {

String sql = "SELECT \* FROM KhoaHoc WHERE MaKH=?";

List<KhoaHoc> list = selectBySql(sql, makh);

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

}

@Override

protected List<KhoaHoc> selectBySql(String sql, Object... args) {

List<KhoaHoc> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = XJdbc.query(sql, args);

while (rs.next()) {

KhoaHoc entity = new KhoaHoc();

entity.setMaKH(rs.getInt("MaKH"));

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"));

entity.setMaCD(rs.getString("MaCD"));

list.add(entity);

}

} finally {

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

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

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.query(sql);

while (rs.next()) {

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

}

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

return list;

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

}

}

* + - 1. ***HocVienDAO***

package com.edusys.dao;

import com.edusys.utils.XJdbc;

import com.edusys.entity.HocVien;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

public class HocVienDAO extends EduVNDAO<HocVien, Integer> {

@Override

public void insert(HocVien model) {

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

XJdbc.update(sql,

model.getMaKH(),

model.getMaNH(),

model.getDiem());

}

@Override

public void update(HocVien model) {

String sql = "UPDATE HocVien SET MaKH=?, MaNH=?, Diem=? WHERE MaHV=?";

XJdbc.update(sql,

model.getMaKH(),

model.getMaNH(),

model.getDiem(),

model.getMaHV());

}

@Override

public void delete(Integer MaHV) {

String sql = "DELETE FROM HocVien WHERE MaHV=?";

XJdbc.update(sql, MaHV);

}

@Override

public List<HocVien> selectAll() {

String sql = "SELECT \* FROM HocVien";

return selectBySql(sql);

}

@Override

public HocVien selectById(Integer mahv) {

String sql = "SELECT \* FROM HocVien WHERE MaHV=?";

List<HocVien> list = selectBySql(sql, mahv);

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

}

@Override

protected List<HocVien> selectBySql(String sql, Object... args) {

List<HocVien> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = XJdbc.query(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);

}

} finally {

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

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

public List<HocVien> selectByKhoaHoc(int maKH) {

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

return this.selectBySql(sql, maKH);

}

}

* + - 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.edusys.dao;

import com.edusys.utils.XJdbc;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

public class ThongKeDAO {

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

try {

List<Object[]> list = new ArrayList<>();

ResultSet rs = XJdbc.query(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 (Exception 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);

}

}

# KIỂM THỬ

### KIỂM THỬ FORM QUẢN LÝ NHÂN VIÊN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Mô tả tình huống | Dữ liệu mẫu | Kết quả mông đợi | Fix |
| 1 | Thêm nhân viên, để trống tất cả |  | Yêu cầu nhập | OK |
| 2 | Thêm nhân viên nhập đủ | Mã: PS17588,MK: 123456Xác nhận MK: 123456Họ tên: Hồ Hoàng PhúMail: [phuho@gmail.com](mailto:phuho@gmail.com)Vai trò: Trưởng phòng | Thông báo thêm thành công | OK |
| 3 | Thêm nhân viên nhập đủ, xác nhận pass lại sai | Mã: PS17588,MK: 123456Xác nhận MK: 1234567Họ tên: Hồ Hoàng PhúMail: [phuho@gmail.com](mailto:phuho@gmail.com)Vai trò: Trưởng phòng | Xác nhận lại mật khẩu sai | OK |
| 4 | Nhân viên: Xóa nhân viên |  | Bạn không thể thực hiện hành động này | OK |
| 5 | Nhân viên: Sửa nhân viên |  | Bạn không thể thực hiện hành động này | OK |

### KIỂM THỬ FORM QUẢN LÝ NGƯỜI HỌC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Mô tả tình huống | Dữ liệu mẫu | Kết quả mông đợi | Fix |
| 1 | Nhấn đúp vào bảng người học |  | Chuyển sang tab Cập nhật, show thông tin người học lên. | OK |
| 2 | Tìm kiếm sinh viên | PS17588 | Bảng còn lại 1 record của sinh viên mã số: PS17588 | OK |
| 3 | Thêm người học, để trống tất cả thông tin |  | Thông báo vui lòng nhập tất cả thông tin | OK |
| 4 | Thêm người học | Nhập đủ tất cả | Thông báo thêm thành công | OK |

### KIỂM THỬ FORM QUẢN LÝ CHUYÊN ĐỀ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Mô tả tình huống | Dữ liệu mẫu | Kết quả mông đợi | Fix |
| 1 | Nhấn đúp vào bảng chuyên đề |  | Chuyển sang tab Cập nhật, show thông tin chuyên đề lên. | OK |
| 2 | Thêm chuyên đề, để trống tất cả thông tin |  | Thông báo vui lòng nhập tất cả thông tin | OK |
| 3 | Thêm chuyên đề | Nhập đủ tất cả | Thông báo thêm thành công | OK |

### KIỂM THỬ FORM QUẢN LÝ KHÓA HỌC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Mô tả tình huống | Dữ liệu mẫu | Kết quả mông đợi | Fix |
| 1 | Chọn sinh viên và nhấn thêm vào khóa học ở bảng người học |  | Chuyển sang tab học viên, show thông tin học viên lên. | OK |
| 2 | Sửa điểm của học viên | abc | Sửa không thành công. Dữ liệu không thay đổi | OK |
| 3 | Sửa điểm của học viên | 11 | Sửa không thành công. Dữ liệu không thay đổi | OK |
| 4 | Sửa điểm của học viên | -100 | Sửa không thành công. Dữ liệu không thay đổi | OK |
| 3 | Chọn sinh viên và nhấn xóa khỏi khóa học ở bảng học viên |  | Xóa học viên khỏi khóa học | OK |
| 4 | Chọn combobox |  | Show dữ liệu ra bảng ở tab Danh sách | OK |
| 5 | Tìm kiếm người học | PS17588 | Bảng còn lại 1 record của người học mã số: PS17588 | OK |

### KIỂM THỬ FORM QUẢN LÝ HỌC VIÊN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Mô tả tình huống | Dữ liệu mẫu | Kết quả mông đợi | Fix |
| 1 | Nhấn đúp vào bảng khóa học |  | Chuyển sang tab Cập nhật, show thông tin khóa học lên. | OK |
| 2 | Thêm khóa học, để trống tất cả thông tin |  | Thông báo vui lòng nhập tất cả thông tin | OK |
| 3 | Thêm khóa học | Nhập đủ tất cả | Thông báo thêm thành công | OK |
| 4 | Chọn combobox |  | Show dữ liệu ra giống với database | OK |

### Kiểm thử form đăng nhập

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Mô tả tình huống | Dữ liệu mẫu | Kết quả mông đợi | Fix |
| 1 | Đăng nhập để trống tài khoản mật khẩu |  | Yêu cầu nhập | OK |
| 2 | Đăng nhập sai tài khoản | phho, 123456 | Sai tên đăng nhập | OK |
| 3 | Đăng nhập sai mật khẩu | phuho, 12345 | Sai mật khẩu | OK |
| 4 | Đăng nhập đúng | phho, 123456 | Đóng form login, mở form main | OK |

# ĐÓNG GÓI VÀ TRIỂN KHAI

### 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 DuAnMau bằng cách chạy file DuAnMau\_SQL.sql
    - Cài đặt JDK 1.8 trở lên
    - Chạy file setup

### HƯỚNG DẪN SỬ DỤNG PHẦN MỀM

* + - Đăng nhập với tài khoản username: phuho và password: 123456
    - Cấp các tài khoản có vai trò là trưởng phòng
    - Đăng nhập tài khoản mới và xóa tài khoản phuho
    - Xem hướng dẫn sử dụng trong menu trợ giúp của phần mềm