

PROJECT DOCUMENT

SOF2041 – Dự án mẫu UDMP-JAVA

Hiểu quy trình thực hiện dự án phần mềm, sử dụng tài liệu để xây dựng phần mềm.

**NGHIÊN CỨU VÀ PHÁT TRIỂN CHƯƠNG TRÌNH
FPOLY**

10/17/2023

MỤC LỤC

1. Phân tích	4
1.1. Hiện trạng.....	4
1.2. Yêu cầu hệ thống	4
1.3. Use case.....	4
2. Thiết kế	8
2.1. Mô hình triển khai.....	8
2.2. Thiết kế CSDL.....	8
2.2.1. Sơ đồ quan hệ thực thể.....	8
2.2.2. Thiết kế chi tiết các thực thể	10
2.3. Thiết kế giao diện	11
2.3.1. Sơ đồ tổ chức giao diện.....	11
2.3.2. Thiết kế giao diện cho các chức năng nghiệp vụ	11
3. Thực hiện viết mã	27
3.1. Viết mã tạo CSDL	27
3.1.1. Tạo CSDL	27
3.1.2. SQL truy vấn và thao tác bảng.....	29
3.1.3. Các thủ tục lưu tổng hợp thống kê.....	30
3.2. Lập trình JDBC	31
3.2.1. Lớp hỗ trợ.....	32
3.2.2. Entity class - Các lớp mô tả dữ liệu.....	36
3.2.3. DAO Class - Các lớp truy xuất dữ liệu	43
3.3. Viết mã cho ứng dụng	56
3.3.1. Các lớp tiện ích.....	56
3.3.2. Viết mã cho các chức năng.....	57
4. Kiểm thử	84
4.1. Kiểm thử form quản lý nhân viên.....	84
4.2. Kiểm thử form quản lý người học	84
4.3. Kiểm thử form quản lý chuyên đề.....	84
4.4. Kiểm thử form quản lý Khóa học.....	84
4.5. Kiểm thử form quản lý học viên	85
5. Đóng gói và triển khai	85
5.1. Hướng dẫn chuyển đổi jar thành exe	85

5.2. Hướng dẫn cài đặt triển khai.....	85
5.3. Hướng dẫn sử dụng phần mềm	85

1. PHÂN TÍCH

1.1. 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.

1.2. 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

1.3. 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.

2. THIẾT KẾ

2.1. 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 kỳ

2.2. THIẾT KẾ CSDL

2.2.1. Sơ đồ quan hệ thực thể

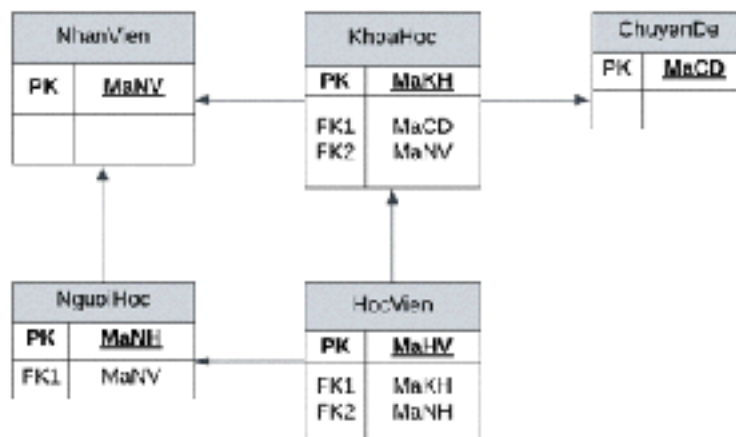
Trong hệ thống đã được mô tả trên chúng ta thấy có các thực thể sau

- ✓ 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.

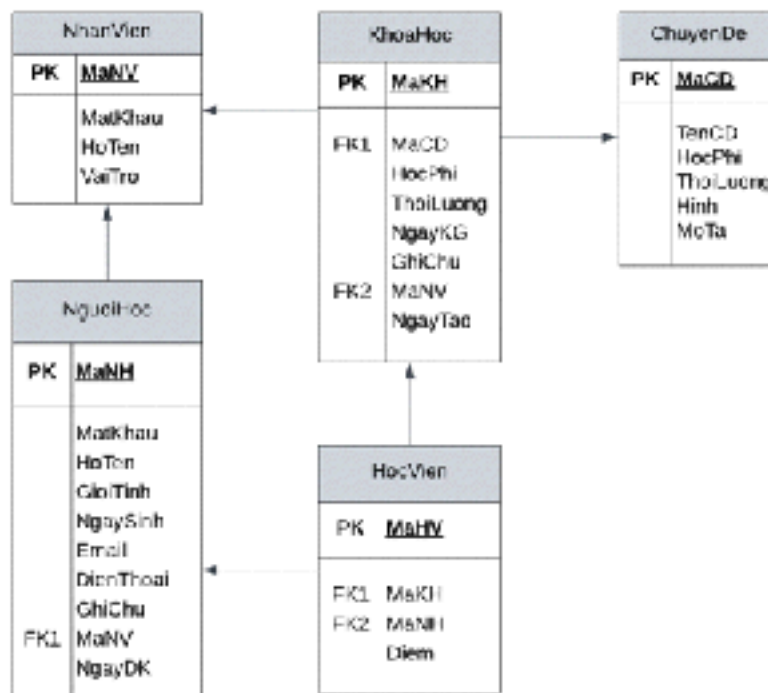
Sau khi phân tích chúng ta có thể thấy các thực thể có mối quan hệ với nhau như sơ đồ ERD sau

2.2.1.1. ERD Diagram level 1



2.2.1.2. 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

2.2.2. Thiết kế chi tiết các thực thể

2.2.2.1. 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

2.2.2.2. 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
MoTa	NVARCHAR(255)	NOT NULL	Mô tả

2.2.2.3. Người học

Bảng NguoiHoc lưu thông tin các những người đã đến trung tâm 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

NgàyDK	DATE	DEFAULT getdate()	Ngày đăng ký của người học
--------	------	-------------------	----------------------------

2.2.2.4. 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
NgàyKG	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
NgàyTao	DATE	DEFAULT getdate()	Ngày đăng ký của người học

2.2.2.5. 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 0.0	Điểm cuối môn

Chú ý: (MaKH và MaNH) là duy nhất

2.3. THIẾT KẾ GIAO DIỆN

2.3.1. 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.

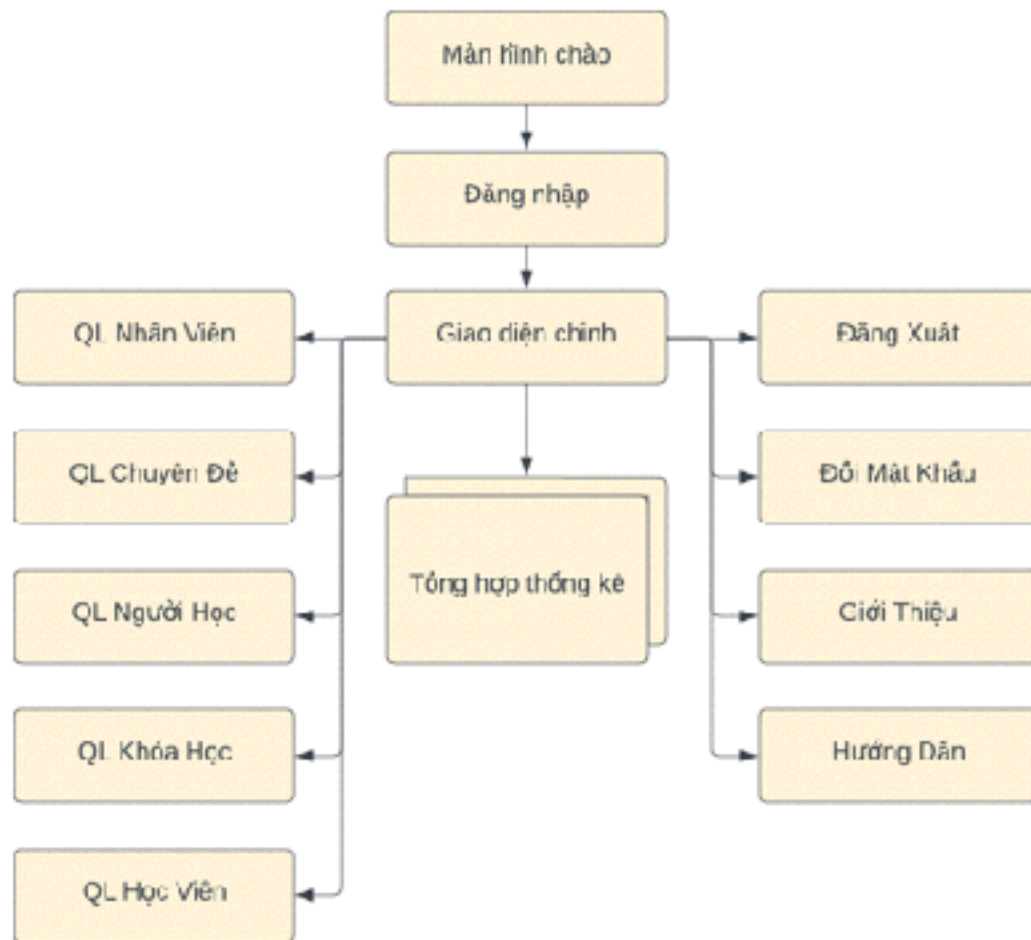
Ngoài ra mỗi ứng dụng trong thời gian khởi động cần có một màn hình chào cùng với thanh tiến trình để người có cảm giác ứng dụng đang khởi động.

2.3.2. 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

c ù a
c á c
bạn.

Lưu ý
q u a n
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
TextField, JTextArea, JPassword, JTextPane	txt	Các loại ô nhập
JRadioButton	rdo	Hộp chọn
JCheckBox	chk	Hộp kiểm
ButtonGroup	brg	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	Bảng
...		

2.3.2.1. Màn hình chào

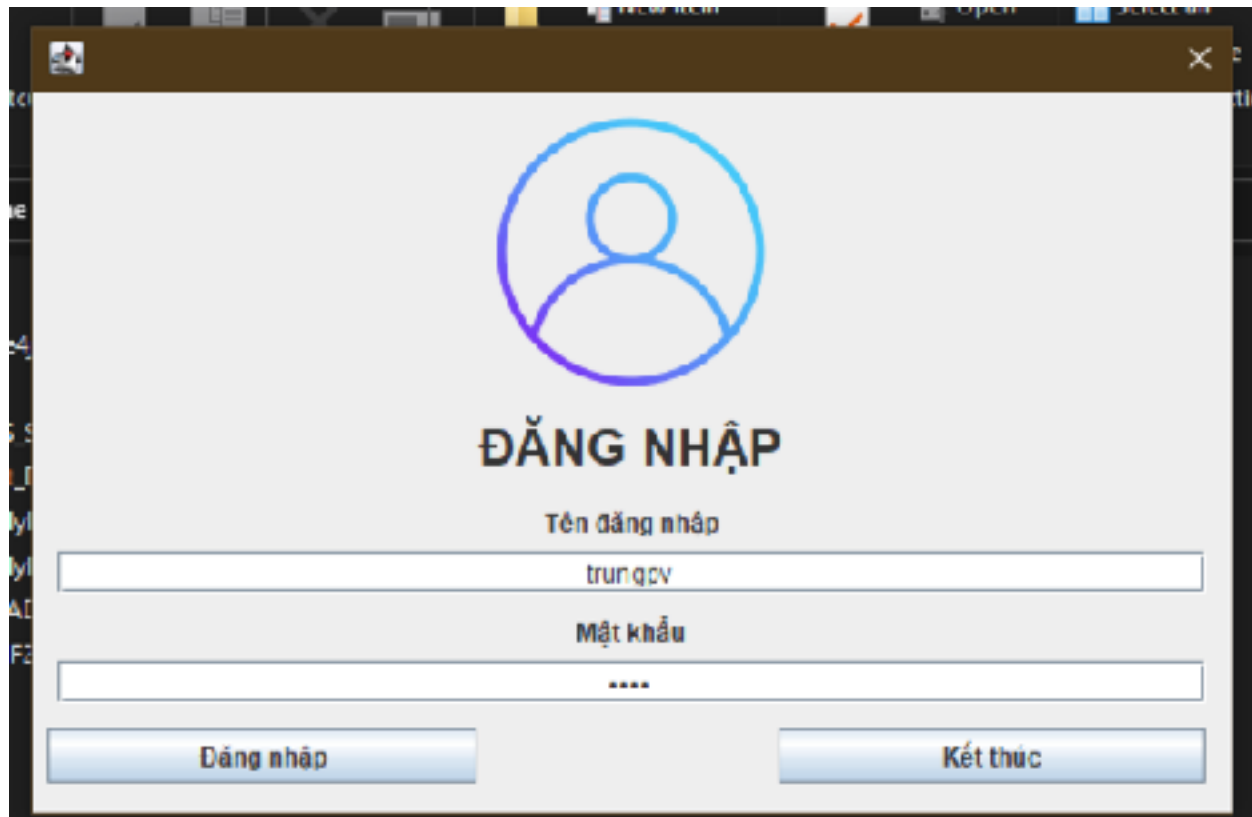


Các thành phần giao diện

	Thành phần	Kiểu	Thuộc tính
1	ChaoJDialog	JDialog	undecorated:true Layout:BorderLayout preferredSize: [514, 350]

2	lblLogo	JLabel	Icon: logo.png horizontalAlignment: CENTER
3	lblText	JLabel	horizontalAlignment: CENTER
4	progressBar	JProgressBar	stringPainted: true

2.3.2.2. Đăng nhập

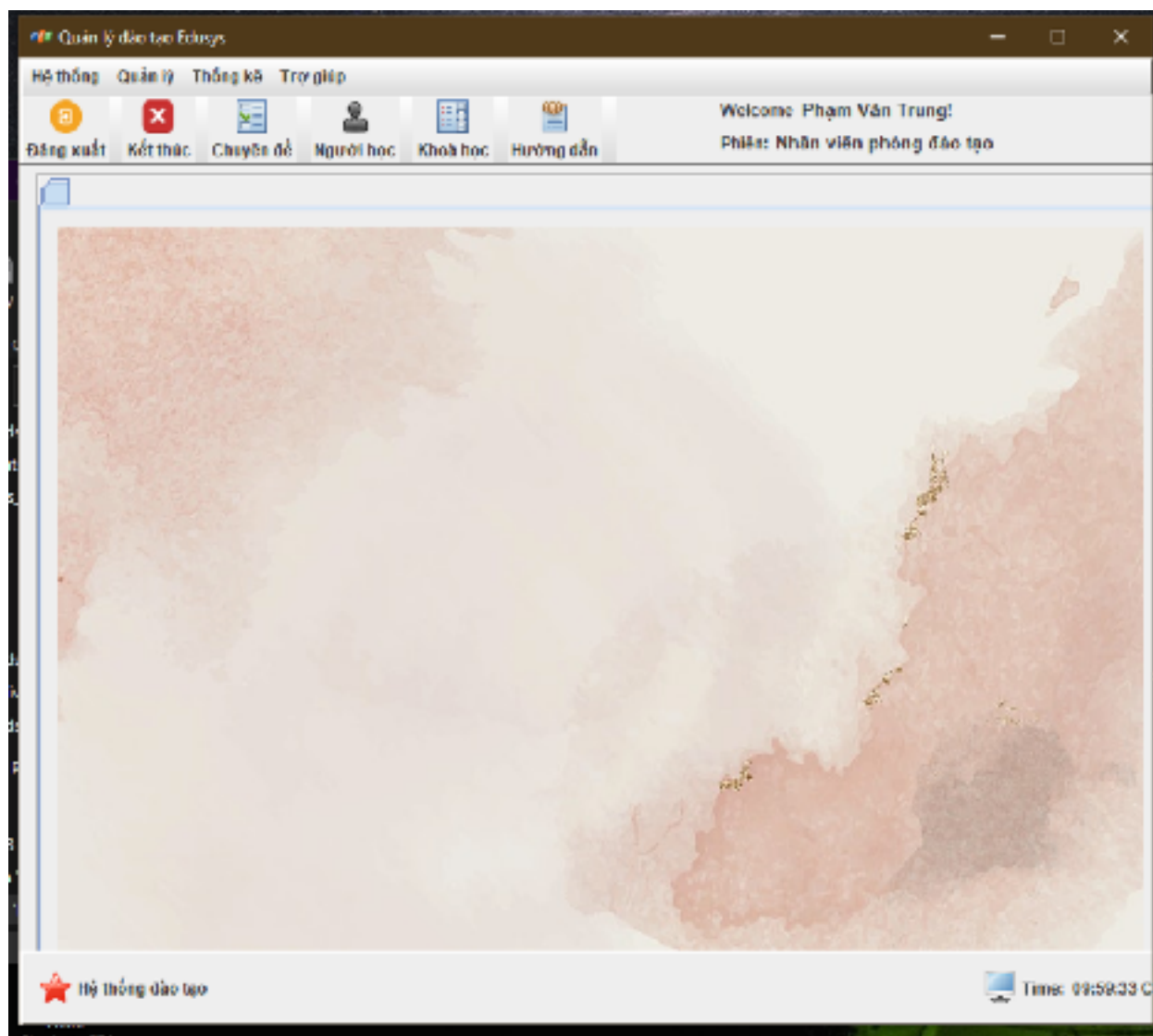


Các thành phần giao diện

	Componel	Kiểu	Thuộc tính
1	DangNhapJDialog	JDialog	title:HỆ THỐNG QUẢN LÝ ĐÀO TẠO Layout: BorderLayout defaultCloseOperation:DO_NOTHING
2	lblHinhAnh	JLabel	Icon: security.png
3	lblMaNV	JLabel	Text: Tên đăng nhập
4	txtMaNV	JTextField	

5	lblPass	JLabel	Text: Mật khẩu
6	txtPass	JPasswordField	
7	btnDangNhap	JButton	Text: Đăng Nhập
8	btnKetThuc	JButton	Text: Kết Thúc

2.3.2.3. *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: Exit.png Accelerator: Ctrl+O
5	mniDoiMatKhau	JMenuItem	Text: Đổi mật khẩu Icon: Cable.png
6	mniKetThuc	JMenuItem	Text: Kết thúc Icon: Stop.png Accelerator: F10
7	mnuQuanLy	JMenu	Text: Quản lý
8	mniNguoiHoc	JMenuItem	Text: Người học Icon: Conference.png
9	mniNhanVien	JMenuItem	Text: Nhân viên Icon: Lists.png
10	mniKhoaHoc	JMenuItem	Text: Khóa học Icon: Certificate.png
11	mniChuyenDe	JMenuItem	Text: Chuyên Đề Icon: User group.png
12	mnuThongKe	JMenu	Text: Thống kê
13	mniBangDiem	JMenuItem	Text: Bảng điểm Icon: Card file.png
14	mniLuongNH	JMenuItem	Text: Lượng người học Icon: Client list.png
15	mniDiemCD	JMenuItem	Text: Điểm chuyên đề Icon: Bar chart.png

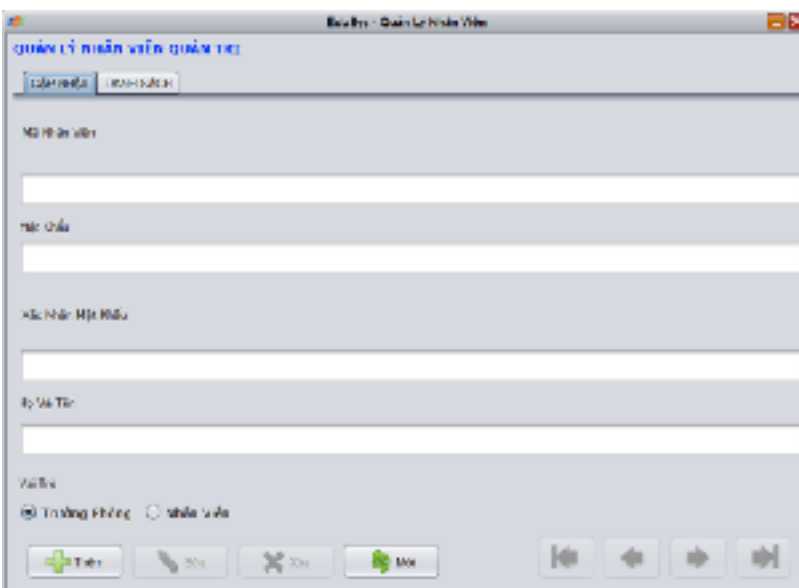
16	mniDoanhThu	JMenuItem	Text: Doanh thu Icon: Dollar.png
17	mnuTroGiup	JMenu	Text: Trợ giúp
18	mniHuongDan	JMenuItem	Text: Hướng dẫn sử dụng Icon: Globe.png Accelerator: F1
19	mniGioiThieu	JMenuItem	Text: Giới thiệu sản phẩm Icon: Brick house.png
20	toolBar	JToolBar	
21	btnDangXuat	JButton	Text: Đăng xuất Icon: Exit.png
22	btnKetThuc	JButton	Text: Kết thúc Icon: Stop sign.png
23	btnChuyenDe	JButton	Text: Chuyên đề Icon: Lists.png
24	btnNguoiHoc	JButton	Text: Người học Icon: Conference.png
25	btnKhoaHoc	JButton	Text: Khóa học Icon: Certificate.png
26	btnHuongDan	JButton	Text: Hướng dẫn Icon: Globe.png
27	lblNoiDung	JLabel	horizontalAlignment: CENTER Icon: logo.png
28	pnlTrangThai	JPanel	Layout: Border Layout
29	lblTrangThai	JLabel	Text: Hệ Quản Lý Đào Tạo Icon: Info.png
30	lblDongHo	JLabel	Text: Time Icon: Alarm.png

2.3.2.4. Quản lý nhân viên

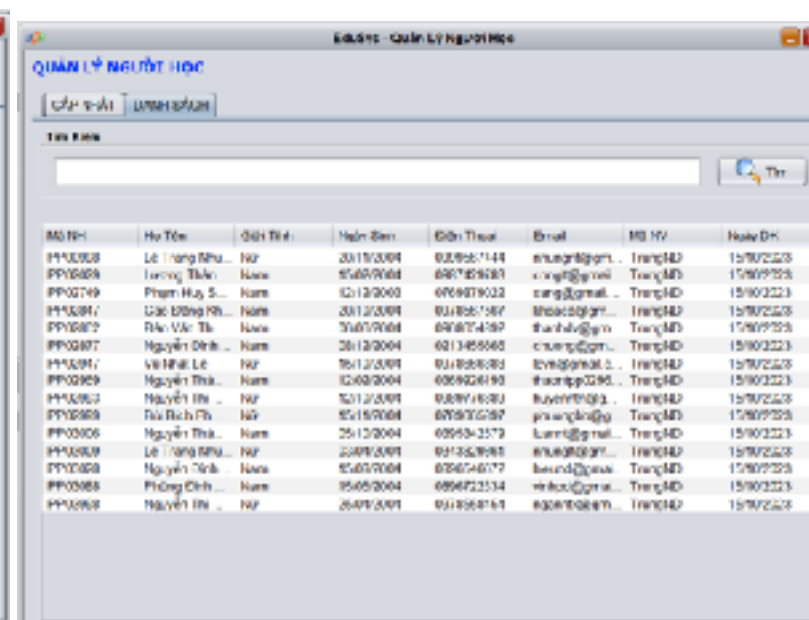
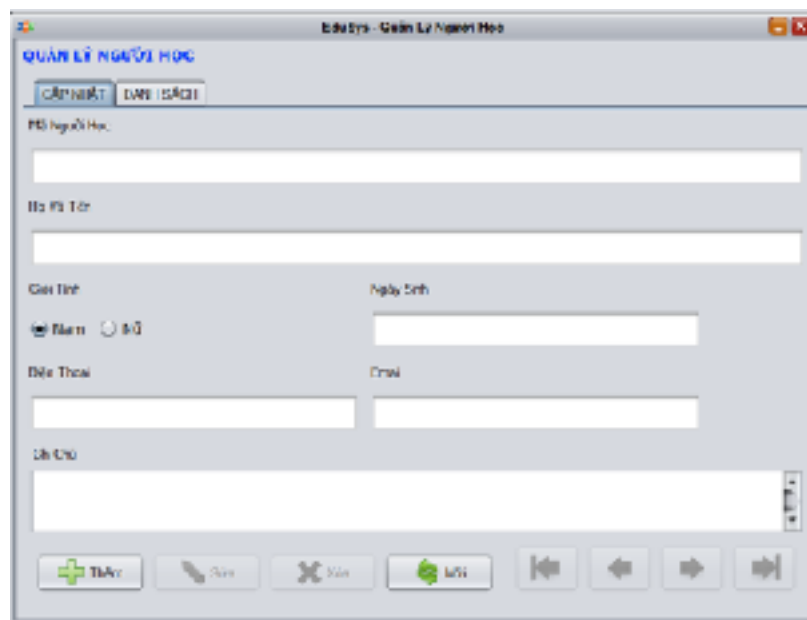
Các thành phần giao diện

	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: [0,51,255] Font: Tahoma 14 Bold
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	JPasswordField	
10	lblMatKhau2	JLabel	Text: Xác nhận mật khẩu
11	txtMatKhau2	JPasswordField	
12	lblHoTen	JLabel	Text: Họ và tên
13	txtHoTen	JTextField	
14	lblVaiTro	JLabel	Text: Vai trò
15	brgVaiTro	ButtonGroup	
16	rdoTruongPhong	JRadioButton	Text: Trưởng phòng Selected: true buttonGroup: brgVaiTro
17	rdoNhanVien	JRadioButton	Text: Nhân viên buttonGroup: brgVaiTro
18	btnThem	JButton	Text: Thêm Icon: Add.png
19	btnSua	JButton	Text: Sửa Icon: Edit.png
20	btnXoa	JButton	Text: Xóa Icon: Delete.png
21	btnMoi	JButton	Text: Mới Icon: Refresh.png
22	btnFirst	JButton	Icon: first_button.png

23	btnPrev	JButton	Icon: previous_button.png
24	btnNext	JButton	Icon: next_button.png
25	btnLast	JButton	Icon: last_button.png
26	tblNhanVien	JTable	Model: như hình



2.3.2.5. Quản lý người học

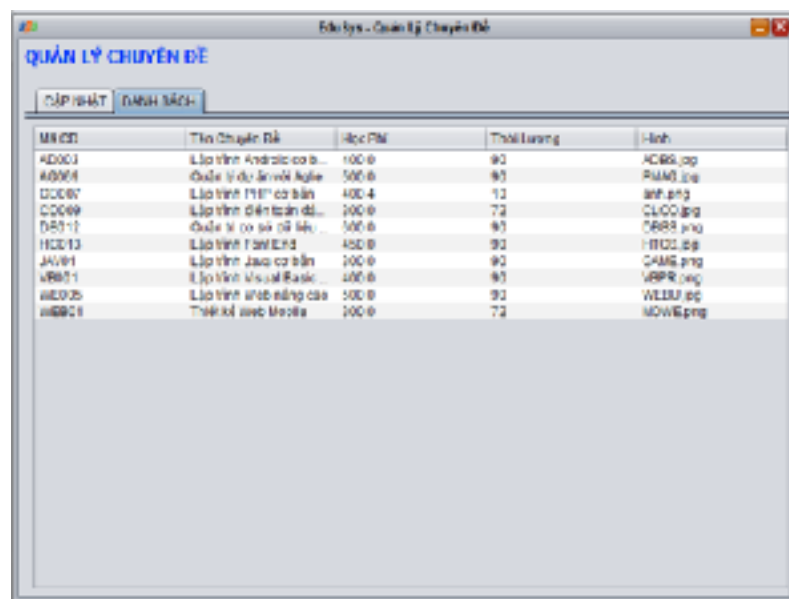
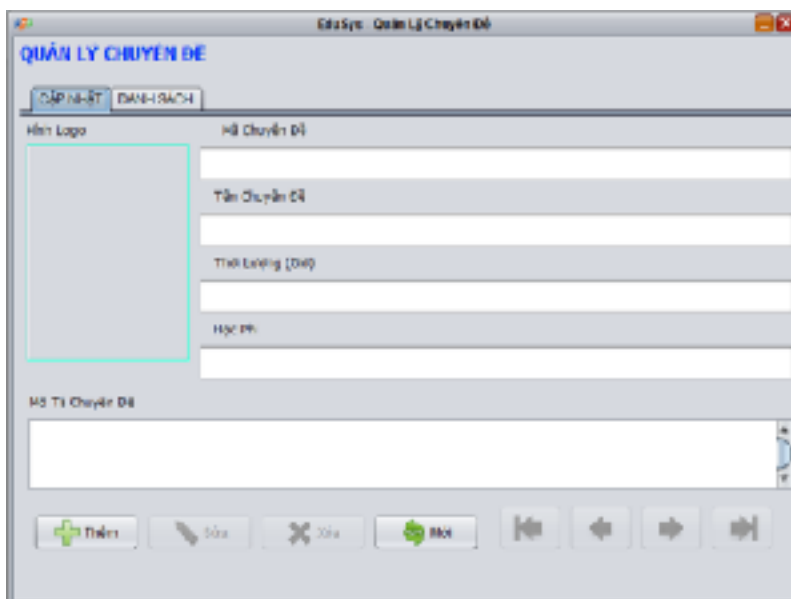


Các thành phần giao diện

	Component	Kiểu	Thuộc tính
1	NguoiHocJFrame	JFrame	Title: QUẢN LÝ NGƯỜI HỌC defaultCloseOperation:DISPONSE
2	lblTitle	JLabel	Text: QUẢN LÝ NGƯỜI HỌC Foreground: [0,51,255] Font: Tahoma 14 Bold
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	rdoNam	JRadioButton	Text: Nam
12	rdoNu	JRadioButton	Text: Nữ
13	brgGioiTinh	ButtonGroup	
14	lblNgaySinh	JLabel	Text: Ngày sinh
15	txtNgaySinh	JTextField	
16	lblDienThoai	JLabel	Text: Điện thoại
17	txtDienThoai	JTextField	
18	lblEmail	JLabel	Text: Email
19	txtEmail	JTextField	
20	lblGhiChu	JLabel	Text: Ghi chú
21	txtGhiChu	JTextArea	
22	btnThem	JButton	Text: Thêm Icon: Add.png
23	btnSua	JButton	Text: Sửa Icon: Edit.png
24	btnXoa	JButton	Text: Xóa Icon: Delete.png

25	btnMoi	JButton	Text: Mới Icon: Refresh.png
26	btnFirst	JButton	Icon: first_button.png
27	btnPrev	JButton	Icon: previous_button.png
28	btnNext	JButton	Icon: next_button.png
29	btnLast	JButton	Icon: last_button.png
30	pnlTimKiem	JPanel	Border: TitledBorder(Title: Tìm Kiếm)
31	txtTimKiem	JTextField	
32	btnTimKiem	JButton	Text: Tìm Icon: Search.png
33	tblNguoiHoc	JTable	Model: Như hình

2.3.2.6. Quản lý chuyên đề

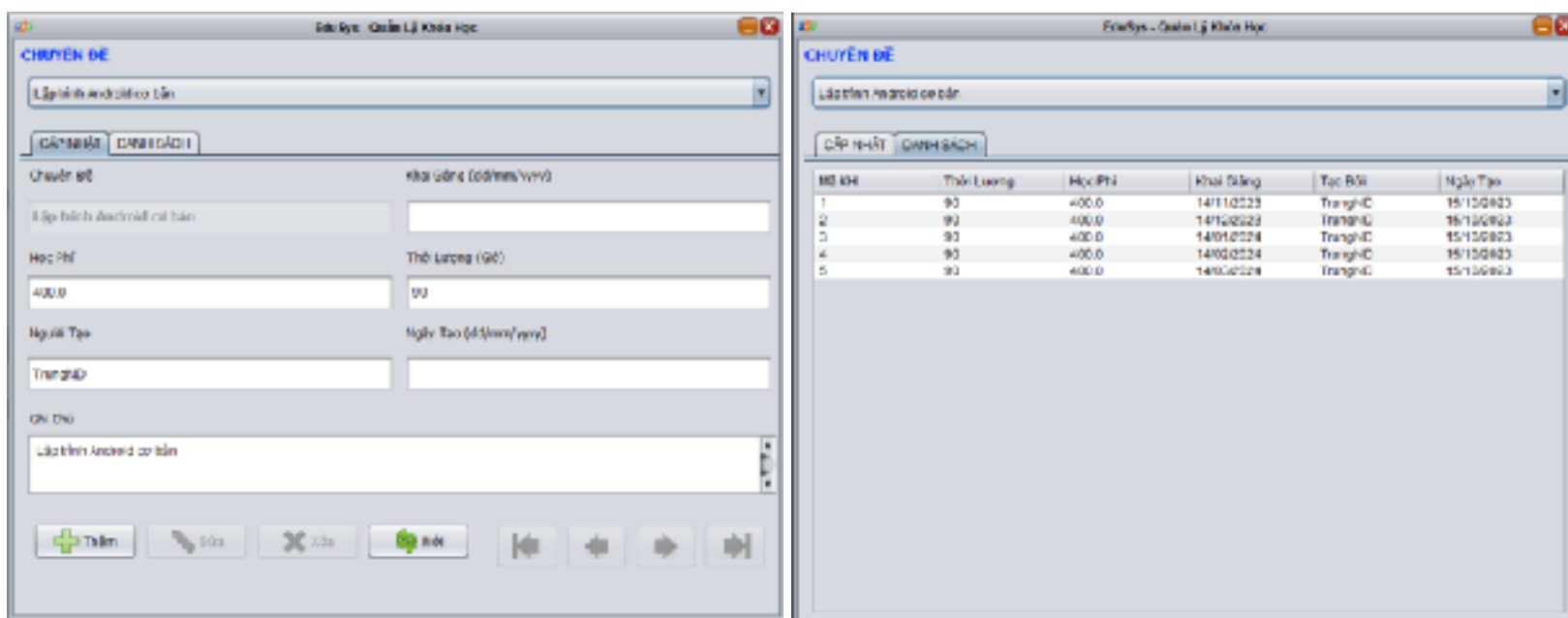


Các thành phần giao diện

	Component	Kiểu	Thuộc tính
1	ChuyenDeJFrame	JFrame	Title: QUẢN LÝ CHUYÊN ĐỀ defaultCloseOperation:DISPONSE
2	lblTitle	Jlabel	Text: QUẢN LÝ CHUYÊN ĐỀ Foreground: [0,51,255] Font: Tahoma 14 Bold

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	JTextField	
16	lblLogo	JLabel	Text: Hình logo
17	lblHinh	JLabel	
18	fileChooser	JFileChooser	dialogTitle: Chọn logo chuyên đề
19	btnThem	JButton	Text: Thêm Icon: Add.png
20	btnSua	JButton	Text: Sửa Icon: Edit.png
21	btnXoa	JButton	Text: Xóa Icon: Delete.png
22	btnMoi	JButton	Text: Mới Icon: Refresh.png
23	btnFirst	JButton	Icon: first_button.png
24	btnPrev	JButton	Icon: previous_button.png
25	btnNext	JButton	Icon: next_button.png
26	btnLast	JButton	Icon: last_button.png
27	tblChuyenDe	JTable	Model: Như hình

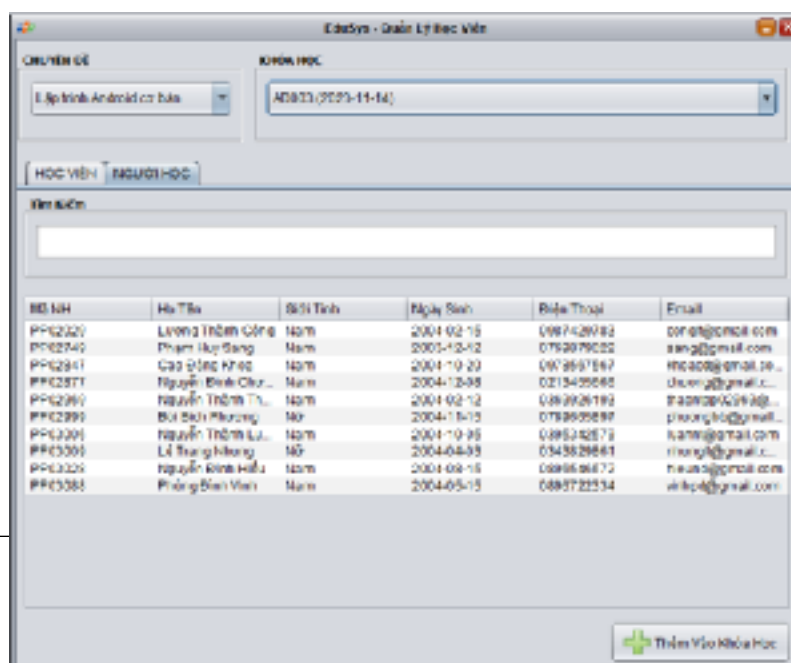
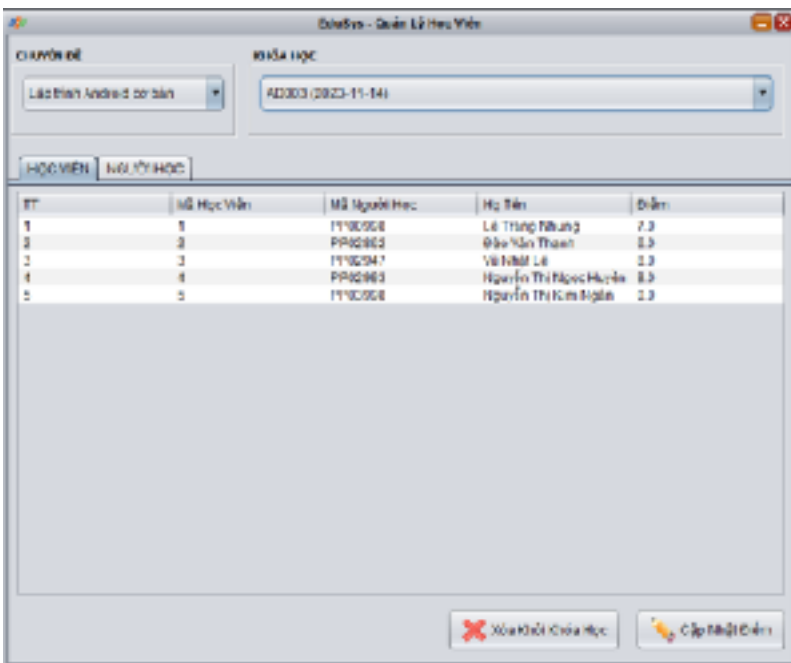
2.3.2.7. Quản lý khóa học



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: [0,51,255] Font: Tahoma 14 Bold
3	tabs	JTabbedPane	
4	pnlEdit	JPanel	Tab Title: CẬP NHẬT
5	pnlList	JPanel	Tab Title: DANH SÁCH
6	lblChuyenDe	JLabel	Text: Chuyên đề
7	txtChuyenDe	JTextField	
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	TextField	Editable: false Enable: false
16	lblNgayTao	JLabel	Text: Ngày tạo
17	txtNgayTao	TextField	Editable: false Enable: false
18	lblGhiChu	JLabel	Text: Ghi chú
19	txtGhiChu	TextField	
20	btnThem	JButton	Text: Thêm Icon: Add.png
21	btnSua	JButton	Text: Sửa Icon: Edit.png
22	btnXoa	JButton	Text: Xóa Icon: Delete.png
23	btnMoi	JButton	Text: Mới Icon: Refresh.png
24	btnFirst	JButton	Icon: first_button.png
25	btnPrev	JButton	Icon: previous_button.png
26	btnNext	JButton	Icon: next_button.png
27	btnLast	JButton	Icon: last_button.png
28	tblKhoaHoc	JTable	Model: Như hình

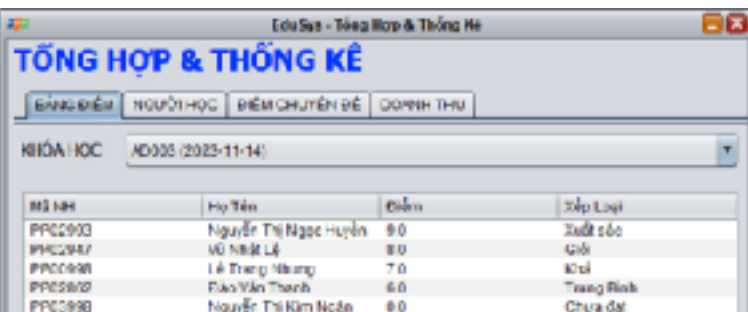


2.3.2.8. Quản lý học viên

Các thành phần giao diện

	Component	Kiểu	Thuộc tính
1	HocVienJFrame	JFrame	Title: QUẢN LÝ HỌC VIÊN KHÓA HỌC defaultCloseOperation:DISPONSE
2	pnlChuyenDe	JPanel	Border: TitledBorder(Title: Chuyên đề)
3	cboChuyenDe	JcomboBox	
4	pnlKhoaHoc	JPanel	Border: TitledBorder(Title: Khóa học)
5	cboKhoaHoc	JcomboBox	
6	tabs	JTabbedPane	
7	pnlHocVien	JPanel	Text: HỌC VIÊN
8	pnlNguoiHoc	JPanel	Text: NGƯỜI HỌC
9	tblHocVien	JTable	Model: Như hình
10	btnXoaHV	JButton	Text: Xóa khỏi khóa học Icon: Delete.png
11	btnSuaDiem	JButton	Text: Cập nhật điểm Icon: Edit.png
12	tblNguoiHoc	JTable	Model: Như hình
13	btnThemHV	JButton	Text: Thêm vào khóa học Icon: Add.png

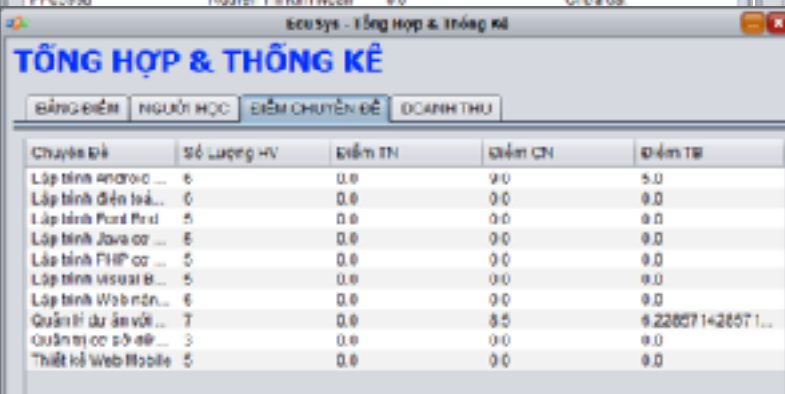
2.3.2.9. Tổng hợp thống kê



Mã NH	Họ Tên	Điểm	Sắp Loại
PR02903	Nguyễn Thị Ngọc Huyền	9.0	Đạt
PR02947	Vũ Nhật Lệ	8.0	Đạt
PR02948	Lê Thanh Nhung	7.0	Đạt
PR02902	Đào Văn Thanh	6.0	Trượt Đỉnh
PR02990	Nguyễn Thị Kim Ngọc	6.0	Chưa đạt



Năm	Số NH	ĐK Số mới	ĐK Muốn Nhứt
2023	15	2023-10-15	2023-10-15



Chuyên đề	Số lượng HV	Điểm TN	Điểm CN	Điểm TB
Lập trình Android ...	6	0.0	0.0	0.0
Lập trình điện toán ...	6	0.0	0.0	0.0
Lập trình Front End ...	5	0.0	0.0	0.0
Lập trình Java cơ ...	5	0.0	0.0	0.0
Lập trình PHP cơ ...	5	0.0	0.0	0.0
Lập trình Visual B ...	5	0.0	0.0	0.0
Lập trình Web nân ...	6	0.0	0.0	0.0
Quản lý dữ liệu với ...	7	0.0	8.5	6.228571428571...
Quản trị cơ sở dữ ...	3	0.0	0.0	0.0
Thiết kế Web Mobile	5	0.0	0.0	0.0



Chuyên đề	Số KH	Số HV	Điểm TB	Điểm CN	Điểm TB
Lập trình An...	1	8	2400.0	400.0	400.0
Lập trình B...	1	6	1600.0	300.0	300.0
Lập trình C...	1	5	2250.0	450.0	450.0
Lập trình J...	1	5	1500.0	300.0	300.0
Lập trình P...	1	5	2000.0	400.0	400.0
Lập trình V...	1	5	2000.0	400.0	400.0
Lập trình W...	1	6	3000.0	500.0	500.0
Quản lý dự...	1	7	3500.0	500.0	500.0
Quản trị cơ...	1	3	1800.0	600.0	600.0
Thiết kế W...	1	5	1500.0	300.0	300.0

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 Foreground: [0,51,255] Font: Tahoma 14 Bold
3	Tabs	JTabbedPane	
4	pnlBangDiem	JPanel	Text: BẢNG ĐIỂM
5	pnlNguoiHoc	JPanel	Text: NGƯỜI HỌC
6	pnlDiemChuyenDe	JPanel	Text: ĐIỂM CHUYÊN ĐỀ
7	pnlDoanhThu	JPanel	Text: DOANH THU
8	lblKhoaHoc	JLabel	Text: KHÓA HỌC
9	cboKhoaHoc	JComboBox	Editable: false
10	tblBangDiem	JTable	Model: Như hình
11	tblNguoiHoc	JTable	Model: Như hình
12	tblDiemChuyenDe	JTable	Mobile: Như hình
13	lblNam	JLabel	Text: Năm
14	cboNam	JComboBox	Editable: false
15	tblDoanhThu	JTable	Model: Như hình

3. THỰC HIỆN VIẾT MÃ

3.1. VIẾT MÃ TẠO CSDL

3.1.1. Tạo CSDL

```

go
create database edusys
go
use edusys
go

create TABLE NhanVien(
    MaNV nvarchar(20) not null,
    MatKhu NVARCHAR(50) not null,
    HoTen NVARCHAR(50) not null,
    VaiTro BIT DEFAULT 0,
    PRIMARY key (MaNV)
)

create TABLE NguoiHoc (
    MaNH NCHAR(7) not null,
    HoTen NVARCHAR(50) NOT NULL,
    GioiTinh BIT DEFAULT 1,
    NgaySinh DATE not null,
    DienThoai NVARCHAR(24) not null,
    Email NVARCHAR(50) not NULL,
    GhiChu NVARCHAR(255) null,
    MaNV NVARCHAR(20) not null,
    NgayDK Date DEFAULT GETDATE(),
    FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV) on UPDATE
CASCADE on Delete no action,
    PRIMARY key (MaNH)
)

CREATE TABLE ChuyenDe (
    MaCD NCHAR(5) not null,
    TenCD NVARCHAR(50) not null,
    HocPhi float not null,
    ThoiLuong int not null,
    Hinh NVARCHAR(255) not null,
    MoTa NVARCHAR(255) not null
    PRIMARY key (MaCD)
)

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) not null,

```

3.1.2. SQL truy vấn và thao tác bảng

3.1.2.1. NhanVien

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

3.1.2.2. 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=?

3.1.2.3. 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=?

3.1.2.4. KhoaHoc

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

3.1.2.5. HocVien

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

3.1.3. Các thủ tục lưu tổng hợp thống kê

3.1.3.1. Số người học từng năm

<pre> 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 </pre>

3.1.3.2. 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
```

3.1.3.3. 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
```

3.1.3.4. 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
```

3.2. LẬP TRÌNH JDBC

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



Trong đó:

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

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

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

3.2.1. 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.

3.2.1.1. Lớp tiện ích XDate

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

```

package com.edusys.utils;

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

public class XDate {

    static SimpleDateFormat formater = new SimpleDateFormat("dd/MM/yyyy");

    public static Date toDate(String date, String... pattern) {
        try {
            if (pattern.length > 0) {
                formater.applyPattern(pattern[0]);
            }
            if (date == null) {
                return XDate.now();
            }
            System.out.println(date);
            return formater.parse(date);
        } catch (ParseException ex) {
            throw new RuntimeException(ex);
        }
    }

    public static String toString(Date date, String... pattern) {
        if (pattern.length > 0) {
            formater.applyPattern(pattern[0]);
        }
        if (date == null) {
            date = XDate.now();
        }
        return formater.format(date);
    }

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

    public static Date add(int days) {
        Date now = XDate.now();
        now.setTime(now.getTime() + days * 24 * 60 * 60 * 1000);
        return now;
    }

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

    public static boolean isValidFormat(String date){
        try {
            Date chk = formater.parse(date);
            return true;
        } catch (Exception e) {
            return false;
        }
    }
}

```

3.2.1.2. *Lớp tiện ích XJdbc*

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

```

package com.edusys.utils;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class XJdbc {

    static String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";
    static String dburl = "jdbc:sqlserver://localhost:1433;databaseName=Laptrinhcity";
    static String user = "sa";
    static String pass = "123123";

    //Nap Driver
    static {
        try {
            Class.forName(driver);
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }

    //Xjdbc.GETSTMT()
    public static PreparedStatement getStmt(String sql, Object... args) throws SQLException {
        Connection conn = DriverManager.getConnection(dburl, user, pass);
        PreparedStatement stmt;

        if (sql.trim().startsWith("{")) {
            stmt = conn.prepareCall(sql); //PROC
        } else {
            stmt = conn.prepareStatement(sql); //SQL
        }

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

    //Xjdbc.QUERY()
    public static ResultSet query(String sql, Object... args) throws SQLException {
        PreparedStatement stmt = XJdbc.getStmt(sql, args);
        return stmt.executeQuery();
    }

    //Xjdbc.VALUE()
    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);
        }
    }

    //Xjdbc.UPDATE()
    public static int update(String sql, Object... args) {
        try {
            PreparedStatement stmt = XJdbc.getStmt(sql, args);
            return stmt.executeUpdate();
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}

```

3.2.2. Entity 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.

3.2.2.1. NhanVien

```
package com.edusys.entity;

public class NhanVien {

    String maNV;
    String matKhau;
    String hoTen;
    boolean vaiTro;

    public String getMaNV() {
        return maNV;
    }

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

    public String getMatKhau() {
        return matKhau;
    }

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

    public String getHoTen() {
        return hoTen;
    }

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

    public boolean isVaiTro() {
        return vaiTro;
    }

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

}
```

3.2.2.2. ChuyenDe

```
package com.edusys.entity;

public class ChuyenDe {

    String MaCD;
    String TenCD;
    double HocPhi;
    int ThoiLuong;
    String Hinh;
    String MoTa;

    public String getMaCD() {
        return MaCD;
    }

    public void setMaCD(String MaCD) {
        this.MaCD = MaCD;
    }

    public String getTenCD() {
        return TenCD;
    }

    public void setTenCD(String TenCD) {
        this.TenCD = TenCD;
    }

    public double getHocPhi() {
        return HocPhi;
    }

    public void setHocPhi(double HocPhi) {
        this.HocPhi = HocPhi;
    }

    public int getThoiLuong() {
        return ThoiLuong;
    }

    public void setThoiLuong(int ThoiLuong) {
        this.ThoiLuong = ThoiLuong;
    }

    public String getHinh() {
        return Hinh;
    }

    public void setHinh(String Hinh) {
        this.Hinh = Hinh;
    }

    public String getMoTa() {
        return MoTa;
    }
}
```

3.2.2.3. *NguoiHoc*


```
package com.edusys.entity;

import java.util.Date;

public class NguoiHoc {

    String MaNH;
    String HoTen;
    boolean GioiTinh;
    Date NgaySinh;
    String DienThoai;
    String Email;
    String GhiChu;
    String MaNV;
    Date NgayDK;

    public String getMaNH() {
        return MaNH;
    }

    public void setMaNH(String MaNH) {
        this.MaNH = MaNH;
    }

    public String getHoTen() {
        return HoTen;
    }

    public void setHoTen(String HoTen) {
        this.HoTen = HoTen;
    }

    public boolean isGioiTinh() {
        return GioiTinh;
    }

    public void setGioiTinh(boolean GioiTinh) {
        this.GioiTinh = GioiTinh;
    }

    public Date getNgaySinh() {
        return NgaySinh;
    }

    public void setNgaySinh(Date NgaySinh) {
        this.NgaySinh = NgaySinh;
    }

    public String getDienThoai() {
        return DienThoai;
    }

    public void setDienThoai(String DienThoai) {
        this.DienThoai = DienThoai;
    }
}
```

3.2.2.4. *KhoaHoc*

```
package com.edusys.entity;

import java.util.Date;

public class KhoaHoc {

    int MaKH;
    String MaCD;
    double HocPhi;
    int ThoiLuong;
    Date NgayKG;
    String GhiChu;
    String MaNV;
    Date NgayTao;

    public int getMaKH() {
        return MaKH;
    }

    public void setMaKH(int MaKH) {
        this.MaKH = MaKH;
    }

    public String getMaCD() {
        return MaCD;
    }

    public void setMaCD(String MaCD) {
        this.MaCD = MaCD;
    }

    public double getHocPhi() {
        return HocPhi;
    }

    public void setHocPhi(double HocPhi) {
        this.HocPhi = HocPhi;
    }

    public int getThoiLuong() {
        return ThoiLuong;
    }

    public void setThoiLuong(int ThoiLuong) {
        this.ThoiLuong = ThoiLuong;
    }

    public Date getNgayKG() {
        return NgayKG;
    }

    public void setNgayKG(Date NgayKG) {
        this.NgayKG = NgayKG;
    }
}
```

3.2.2.5. *HocVien*

```

package com.edusys.entity;

public class HocVien {

    int MaHV;
    int MaKH;
    String MaNH;
    double Diem;

    public int getMaHV() {
        return MaHV;
    }

    public void setMaHV(int MaHV) {
        this.MaHV = MaHV;
    }

    public int getMaKH() {
        return MaKH;
    }

    public void setMaKH(int MaKH) {
        this.MaKH = MaKH;
    }

    public String getMaNH() {
        return MaNH;
    }

    public void setMaNH(String MaNH) {
        this.MaNH = MaNH;
    }

    public double getDiem() {
        return Diem;
    }

    public void setDiem(double Diem) {
        this.Diem = Diem;
    }
}

```

3.2.3. 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 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

3.2.3.1. *NhanVienDAO*

```

package com.edusys.dao;

import com.edusys.entity.NhanVien;
import com.edusys.utils.XJdbc;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

public class NhanVienDAO extends EduSysDAO<NhanVien, String>{
    String INSERT_SQL = "INSERT INTO NhanVien(MaNV, MatKhau, HoTen, VaiTro) VALUES(?,?,?,?)";
    String UPDATE_SQL = "UPDATE NhanVien SET MatKhau = ?, HoTen = ?, VaiTro = ? WHERE MaNV = ?";
    String DELETE_SQL = "DELETE FROM NhanVien WHERE MaNV = ?";
    String SELECT_ALL_SQL = "SELECT * FROM NhanVien";
    String SELECT_BY_ID_SQL = "SELECT * FROM NhanVien WHERE MaNV = ?";

    @Override
    public void insert(NhanVien entity) {
        XJdbc.update(INSERT_SQL,
            entity.getMaNV(), entity.getMatKhau(), entity.getHoTen(), entity.isVaiTro());
    }

    @Override
    public void update(NhanVien entity) {
        XJdbc.update(UPDATE_SQL,
            entity.getMatKhau(), entity.getHoTen(), entity.isVaiTro(), entity.getMaNV());
    }

    @Override
    public void delete(String id) {
        XJdbc.update(DELETE_SQL, id);
    }

    @Override
    public List<NhanVien> selectAll() {
        return this.selectBySQL(SELECT_ALL_SQL);
    }

    @Override
    public NhanVien selectByID(String id) {
        List<NhanVien> list = this.selectBySQL(SELECT_BY_ID_SQL, id);
        if(list.isEmpty()){
            return null;
        }
        return list.size() > 0 ? list.get(0) : null;
    }

    @Override
    protected List<NhanVien> selectBySQL(String sql, Object... args) {
        List<NhanVien> list = new ArrayList<NhanVien>();
        try {
            ResultSet rs = XJdbc.query(sql, args);
            while(rs.next()){
                NhanVien nv = new NhanVien();
                nv.setMaNV(rs.getString("MaNV"));
                nv.setMatKhau(rs.getString("MatKhau"));
                nv.setHoTen(rs.getString("HoTen"));
                nv.setVaiTro(rs.getBoolean("VaiTro"));
                list.add(nv);
            }
        }
    }
}

```

3.2.3.2. *ChuyenDeDAO*


```

package com.edusys.dao;

import com.edusys.entity.ChuyenDe;
import com.edusys.utils.XJdbc;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

public class ChuyenDeDAO extends EduSysDAO<ChuyenDe, String> {

    String INSERT_SQL = "INSERT INTO ChuyenDe(MaCD, TenCD, HocPhi, ThoiLuong, Hinh, MoTa)
VALUES(?,?,?,?,?)";
    String UPDATE_SQL = "UPDATE ChuyenDe SET TenCD = ?, HocPhi = ?, ThoiLuong = ?, Hinh = ?, MoTa
= ? WHERE MaCD = ?";
    String DELETE_SQL = "DELETE FROM ChuyenDe WHERE MaCD = ?";
    String SELECT_ALL_SQL = "SELECT * FROM ChuyenDe";
    String SELECT_BY_ID_SQL = "SELECT * FROM ChuyenDe WHERE MaCD = ?";

    @Override
    public void insert(ChuyenDe entity) {
        XJdbc.update(INSERT_SQL,
            entity.getMaCD(), entity.getTenCD(), entity.getHocPhi(), entity.getThoiLuong(), entity.getHinh(),
entity.getMoTa());
    }

    @Override
    public void update(ChuyenDe entity) {
        XJdbc.update(UPDATE_SQL,
            entity.getTenCD(), entity.getHocPhi(), entity.getThoiLuong(), entity.getHinh(), entity.getMoTa(),
entity.getMaCD());
    }

    @Override
    public void delete(String id) {
        XJdbc.update(DELETE_SQL, id);
    }

    @Override
    public List<ChuyenDe> selectAll() {
        return this.selectBySQL(SELECT_ALL_SQL);
    }

    @Override
    public ChuyenDe selectByID(String id) {
        List<ChuyenDe> list = this.selectBySQL(SELECT_BY_ID_SQL, id);
        if(list.isEmpty()){
            return null;
        }
        return list.size() > 0 ? list.get(0) : null;
    }

    @Override
    protected List<ChuyenDe> selectBySQL(String sql, Object... args) {
        List<ChuyenDe> list = new ArrayList<ChuyenDe>();
        try {
            ResultSet rs = XJdbc.query(sql, args);
            while(rs.next()){
                ChuyenDe cd = new ChuyenDe();
                cd.setMaCD(rs.getString("MaCD"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return list;
    }
}

```

3.2.3.3. *NguoiHocDAO*

```

package com.edusys.dao;

import com.edusys.entity.NguoiHoc;
import com.edusys.utils.XJdbc;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

public class NguoiHocDAO extends EduSysDAO<NguoiHoc, String> {

    String INSERT_SQL = "INSERT INTO NguoiHoc(MaNH, HoTen, GioiTinh, NgaySinh, DienThoai, Email,
    GhiChu, MaNV) VALUES(?,?,?,?,?,?,?)";
    String UPDATE_SQL = "UPDATE NguoiHoc SET HoTen = ?, GioiTinh = ?, NgaySinh = ?, DienThoai = ?,
    Email = ?, GhiChu = ?, MaNV = ? WHERE MaNH = ?";
    String DELETE_SQL = "DELETE FROM NguoiHoc WHERE MaNH = ?";
    String SELECT_ALL_SQL = "SELECT * FROM NguoiHoc";
    String SELECT_BY_ID_SQL = "SELECT * FROM NguoiHoc WHERE MaNH = ?";

    @Override
    public void insert(NguoiHoc entity) {
        XJdbc.update(INSERT_SQL,
            entity.getMaNH(), entity.getHoTen(), entity.isGioiTinh(),
            new java.sql.Date(entity.getNgaySinh().getTime()),
            entity.getDienThoai(), entity.getEmail(), entity.getGhiChu(), entity.getMaNV());
    }

    @Override
    public void update(NguoiHoc entity) {
        XJdbc.update(UPDATE_SQL,
            entity.getHoTen(), entity.isGioiTinh(), new java.sql.Date(entity.getNgaySinh().getTime()),
            entity.getDienThoai(), entity.getEmail(), entity.getGhiChu(), entity.getMaNV(),
            entity.getMaNH());
    }

    @Override
    public void delete(String id) {
        XJdbc.update(DELETE_SQL, id);
    }

    @Override
    public List<NguoiHoc> selectAll() {
        return this.selectBySQL(SELECT_ALL_SQL);
    }

    @Override
    public NguoiHoc selectByID(String id) {
        List<NguoiHoc> list = this.selectBySQL(SELECT_BY_ID_SQL, id);
        if(list.isEmpty()){
            return null;
        }
        return list.get(0);
    }

    @Override
    protected List<NguoiHoc> selectBySQL(String sql, Object... args) {
        List<NguoiHoc> list = new ArrayList<NguoiHoc>();
        try {
            ResultSet rs = XJdbc.query(sql, args);
            while(rs.next()){

```

3.2.3.4. *KhoaHocDAO*

```

package com.edusys.dao;

import com.edusys.entity.KhoaHoc;
import com.edusys.utils.XJdbc;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

public class KhoaHocDAO extends EduSysDAO<KhoaHoc, Integer> {

    String INSERT_SQL = "INSERT INTO KhoaHoc(MaCD, HocPhi, ThoiLuong, NgayKG, GhiChu, MaNV,
    NgayTao) VALUES(?,?,?,?,?,GETDATE())";
    String UPDATE_SQL = "UPDATE KhoaHoc SET MaCD = ?, HocPhi = ?, ThoiLuong = ?, NgayKG = ?,
    GhiChu = ?, MaNV = ? WHERE MaKH = ?";
    String DELETE_SQL = "DELETE FROM KhoaHoc WHERE MaKH = ?";
    String SELECT_ALL_SQL = "SELECT * FROM KhoaHoc";
    String SELECT_BY_ID_SQL = "SELECT * FROM KhoaHoc WHERE MaKH = ?";

    @Override
    public void insert(KhoaHoc entity) {
        XJdbc.update(INSERT_SQL,
            entity.getMaCD(), entity.getHocPhi(), entity.getThoiLuong(),
            new java.sql.Date(entity.getNgayKG().getTime()), entity.getGhiChu(), entity.getMaNV());
    }

    @Override
    public void update(KhoaHoc entity) {
        XJdbc.update(UPDATE_SQL,
            entity.getMaCD(), entity.getHocPhi(), entity.getThoiLuong(),
            new java.sql.Date(entity.getNgayKG().getTime()), entity.getGhiChu(), entity.getMaNV(),
            entity.getMaKH());
    }

    @Override
    public List<KhoaHoc> selectAll() {
        return this.selectBySQL(SELECT_ALL_SQL);
    }

    @Override
    protected List<KhoaHoc> selectBySQL(String sql, Object... args) {
        List<KhoaHoc> list = new ArrayList<KhoaHoc>();
        try {
            ResultSet rs = XJdbc.query(sql, args);
            while (rs.next()) {
                KhoaHoc kh = new KhoaHoc();
                kh.setMaKH(rs.getInt("MaKH"));
                kh.setMaCD(rs.getString("MaCD"));
                kh.setHocPhi(rs.getDouble("HocPhi"));
                kh.setThoiLuong(rs.getInt("ThoiLuong"));
                kh.setNgayKG(rs.getDate("NgayKG"));
                kh.setMaNV(rs.getString("MaNV"));
                kh.setNgayTao(rs.getDate("NgayTao"));
                list.add(kh);
            }
            rs.getStatement().getConnection().close();
            return list;
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}

```

3.2.3.5. *HocVienDAO*

```

package com.edusys.dao;

import com.edusys.entity.HocVien;
import com.edusys.utils.XJdbc;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

public class HocVienDAO extends EduSysDAO<HocVien, Integer> {

    String INSERT_SQL = "INSERT INTO HocVien(MaKH, MaNH, Diem) VALUES(?,?,?)";
    String UPDATE_SQL = "UPDATE HocVien SET MaKH = ?, MaNH = ?, Diem = ? WHERE MaHV = ?";
    String DELETE_SQL = "DELETE FROM HocVien WHERE MaHV = ?";
    String SELECT_ALL_SQL = "SELECT * FROM HocVien";
    String SELECT_BY_ID_SQL = "SELECT * FROM HocVien WHERE MaHV = ?";

    @Override
    public void insert(HocVien entity) {
        XJdbc.update(INSERT_SQL,
            entity.getMaKH(), entity.getMaNH(), entity.getDiem());
    }

    @Override
    public void update(HocVien entity) {
        XJdbc.update(UPDATE_SQL,
            entity.getMaKH(), entity.getMaNH(), entity.getDiem(), entity.getMaHV());
    }

    @Override
    public void delete(Integer id) {
        XJdbc.update(DELETE_SQL, id);
    }

    @Override
    public List<HocVien> selectAll() {
        return this.selectBySQL(SELECT_ALL_SQL);
    }

    @Override
    public HocVien selectByID(Integer id) {
        List<HocVien> list = this.selectBySQL(SELECT_BY_ID_SQL, id);
        if (list.isEmpty()) {
            return null;
        }
        return list.get(0);
    }

    @Override
    protected List<HocVien> selectBySQL(String sql, Object... args) {
        List<HocVien> list = new ArrayList<HocVien>();
        try {
            ResultSet rs = XJdbc.query(sql, args);
            while (rs.next()) {
                HocVien hv = new HocVien();
                hv.setMaHV(rs.getInt("MaHV"));
                hv.setMaKH(rs.getInt("MaKH"));
                hv.setMaNH(rs.getString("MaNH"));
                hv.setDiem(rs.getDouble("Diem"));
                list.add(hv);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return list;
    }
}

```

3.2.3.6. *ThongKeDAO*

Lớp DAO này chứa các phương thức gọi đến các thủ tục lưu được xây dựng trước đây để thực 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_ThongKeNguoiHoc}" ;
        String[] cols = {"Nam", "SoLuong", "DauTien", "CuoiCung"};
        return this.getListOfArray(sql, cols);
    }

    public List<Object[]> getDiemChuyenDe() {
        String sql = "{CALL sp_ThongKeDiem}" ;
        String[] cols = {"ChuyenDe", "SoHV", "ThapNhat", "CaoNhat", "TrungBinh"};
        return this.getListOfArray(sql, cols);
    }

    public List<Object[]> getDoanhThu(int nam) {
        String sql = "{CALL sp_ThongKeDoanhThu(?)}" ;
        String[] cols = {"ChuyenDe", "SoKH", "SoHV", "DoanhThu", "ThapNhat", "CaoNhat", "TrungBinh"};
        return this.getListOfArray(sql, cols, nam);
    }
}
```


3.3. VIẾT MÃ CHO ỨNG DỤNG

3.3.1. 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.

3.3.1.1. *XImage*

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

```
package com.edusys.utils;

import java.awt.Image;
import java.io.File;
import java.net.URL;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardCopyOption;
import javax.swing.ImageIcon;

public class XImage {
    public static Image getAppIcon(){
        URL url = XImage.class.getResource("/com/edusys/utils/fpt.png");
        return new ImageIcon(url).getImage();
    }

    public static void save(File src){
        File dst = new File("logos", src.getName());
        if(!dst.getParentFile().exists()){
            dst.getParentFile().mkdirs(); //Tao thu muc logos neu chua ton tai
        }
        try {
            Path from = Paths.get(src.getAbsolutePath());
            Path to = Paths.get(dst.getAbsolutePath());
            Files.copy(from, to, StandardCopyOption.REPLACE_EXISTING); //Copy file vao thu muc
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }

    public static ImageIcon read(String fileName){
        File path = new File("logos", fileName);
        return new ImageIcon(path.getAbsolutePath());
    }
}
```

3.3.1.2. *MsgBox*

Lớp MsgBox chứa các phương thức đơn giản hóa JOptionPane

```
package com.edusys.utils;

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

public class MsgBox {
    public static void alert(Component parent, String message){
        JOptionPane.showMessageDialog(parent, message,
            "Hệ thống quản lý đào tạo", JOptionPane.INFORMATION_MESSAGE);
    }

    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;
    }

    public static String prompt(Component parent, String message){
        return JOptionPane.showInputDialog(parent, message,
            "Hệ thống quản lý đào tạo", JOptionPane.INFORMATION_MESSAGE);
    }
}
```

3.3.2. 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 lượt viết mã thực hiện các chức năng cho các giao diện sau đây.

3.3.2.1. Màn hình chào (ChaoJDialog)



HOẠT ĐỘNG

Sự kiện	Công việc
Constructor	init() setIconImage(XImage.getAppIcon());

CÀI ĐẶT MÃ NGUỒN

```

void init() {
    setLocationRelativeTo(null);
    timer = new Timer(10, new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            int value = ProgressBar.getValue();
            if (value < 100) {
                ProgressBar.setValue(value + 1);
                if (value == 50) {
                    lblText.setText("Đang kết nối cơ sở dữ liệu!");
                }
                if (value == 90) {
                    lblText.setText("Chương trình sẵn sàng!");
                }
            } else {
                CuaSoChaoJDialog.this.dispose();
                new DangNhapJFrame().setVisible(true);
                timer.stop();
            }
        }
    });
}
    
```

```
);  
    timer.start();  
}
```

3.3.2.2. Form đăng nhập (DangNhapJFrame)



MÔ TẢ HOẠT ĐỘNG

Sự kiện	Công việc
Constructor	setLocationRelativeTo(null); setIconImage(XImage.getAppIcon());
btnDangNhap	dangNhap()
btnKetThuc	ketThuc()

HƯỚNG DẪN CÀI ĐẶT MÃ

Bổ sung mã sau vào cuối lớp DangNhapJFrame.

```

NhanVienDAO dao = new NhanVienDAO();

void dangNhap() {
    String manv = txtMaNV.getText();
    String matkhau = new String(txtPass.getPassword());

    if (manv.length() == 0) {
        MsgBox.alert(this, "Không được để trống Tên Đăng Nhập!");
        txtMaNV.requestFocus();
        return;
    } else if (matkhau.length() == 0) {
        MsgBox.alert(this, "Không được để trống Mật Khẩu!");
        txtPass.requestFocus();
        return;
    }

    NhanVien nv = dao.selectByID(manv);

    if (nv == null) {
        MsgBox.alert(this, "Sai tên đăng nhập!");
        txtMaNV.requestFocus();
    } else if (!matkhau.equals(nv.getMatKhau())) {
        MsgBox.alert(this, "Sai mật khẩu!");
        txtPass.requestFocus();
    } else {
        MsgBox.alert(this, "Đăng nhập thành công.");
        Auth.user = nv;
        this.dispose();
        new MainFrame().setVisible(true);
    }
}

void ketThuc() {
    if (MsgBox.confirm(this, "Bạn muốn kết thúc ứng dụng?")) {
        System.exit(0);
    }
}

```

3.3.2.3. *Giao diện chính (MainJFrame)*

MÔ TẢ HOẠT ĐỘNG

Sự kiện	Công việc
Constructor	setIconImage(XImage.getAppIcon()); starDongHo();
Đăng nhập	dangNhap();
Đăng xuất	dangXuat();
Kết thúc	ketThuc();
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 thu từng chuyên đề	openThongKe(4)
Hướng dẫn	openHuongDan();
Giới thiệu	openGioiThieu();



CÀI ĐẶT MÃ NGUỒN

```

void starDongHo() {
    //Đồng hồ
    new Timer(10, new ActionListener() {
        SimpleDateFormat format = new SimpleDateFormat("hh:mm:ss a");

        @Override
        public void actionPerformed(ActionEvent e) {
            lblTime.setText(format.format(new Date()));
        }
    }).start();
}

void ketThuc() {
    if (MsgBox.confirm(this, "Bạn có muốn kết thúc chương trình?")) {
        System.exit(0);
    }
}

void dangXuat() {
    if (MsgBox.confirm(this, "Bạn có muốn đăng xuất không?")) {
        Auth.clear();
        this.dispose();
        new DangNhapJFrame().setVisible(true);
    }
}

void openDoiMatKhau() {
    if (Auth.isLogin()) {
        JFrame internalFrame = new DoiMatKhauFrame();
        DesktopPane.add(internalFrame);
        internalFrame.setVisible(true);
    }
}

void openNhanVien() {
    if (Auth.isLogin()) {
        NhanVienFrame frame = new NhanVienFrame();
        DesktopPane.add(frame);
        frame.setVisible(true);
    }
}

```


3.3.2.4. Quản lý nhân viên (NhanVienJFrame)

QUẢN LÝ NHÂN VIÊN QUẢN TRỊ

CẬP NHẬT **DANH SÁCH**

Mã Nhân Viên

Mật Khẩu

Xác Nhận Mật Khẩu

Họ và Tên

Vai Trò

☒ Trưởng Phòng ☐ Nhân Viên

Thêm **Sửa** **Xóa** **Mới**

QUẢN LÝ NHÂN VIÊN QUẢN TRỊ

CẬP NHẬT **DANH SÁCH**

Mã Nhân Viên

TrungND

Mật Khẩu

123456

Xác Nhận Mật Khẩu

123456

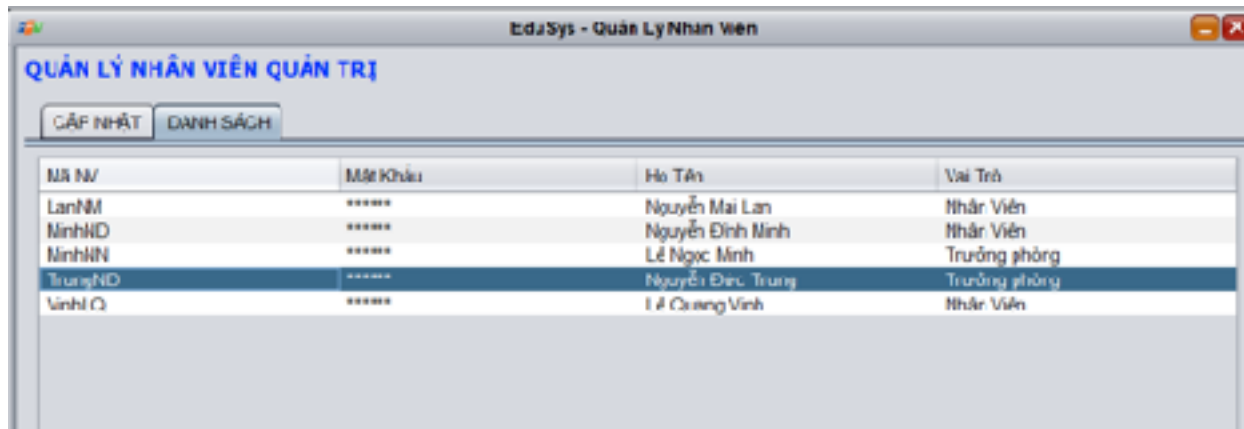
Họ và Tên

Nguyễn Đức Trung

Vai Trò

☒ Trưởng Phòng ☐ Nhân Viên

Thêm **Sửa** **Xóa** **Mới**



MÔ TẢ HOẠT ĐỘNG

Sự kiện	Công việc phải làm
Constructor	Init()
tblNhanVienMouseClicked	<pre>if (evt.getClickCount() == 2) { this.row = tblNhanVien.getSelectedRow(); this.edit(); }</pre>
btnFirstActionPerformed	first();
btnPrevActionPerformed	prev();
btnNextActionPerformed	next();
btnLastActionPerformed	last();
btnXoaActionPerformed	delete();
btnSuaActionPerformed	<pre>if(validateForm(false)){ update(); }</pre>
btnThemActionPerformed	<pre>if (validateForm(true)) { insert(); }</pre>
btnMoiActionPerformed	clearForm();

CÀI ĐẶT MÃ NGUỒN

```

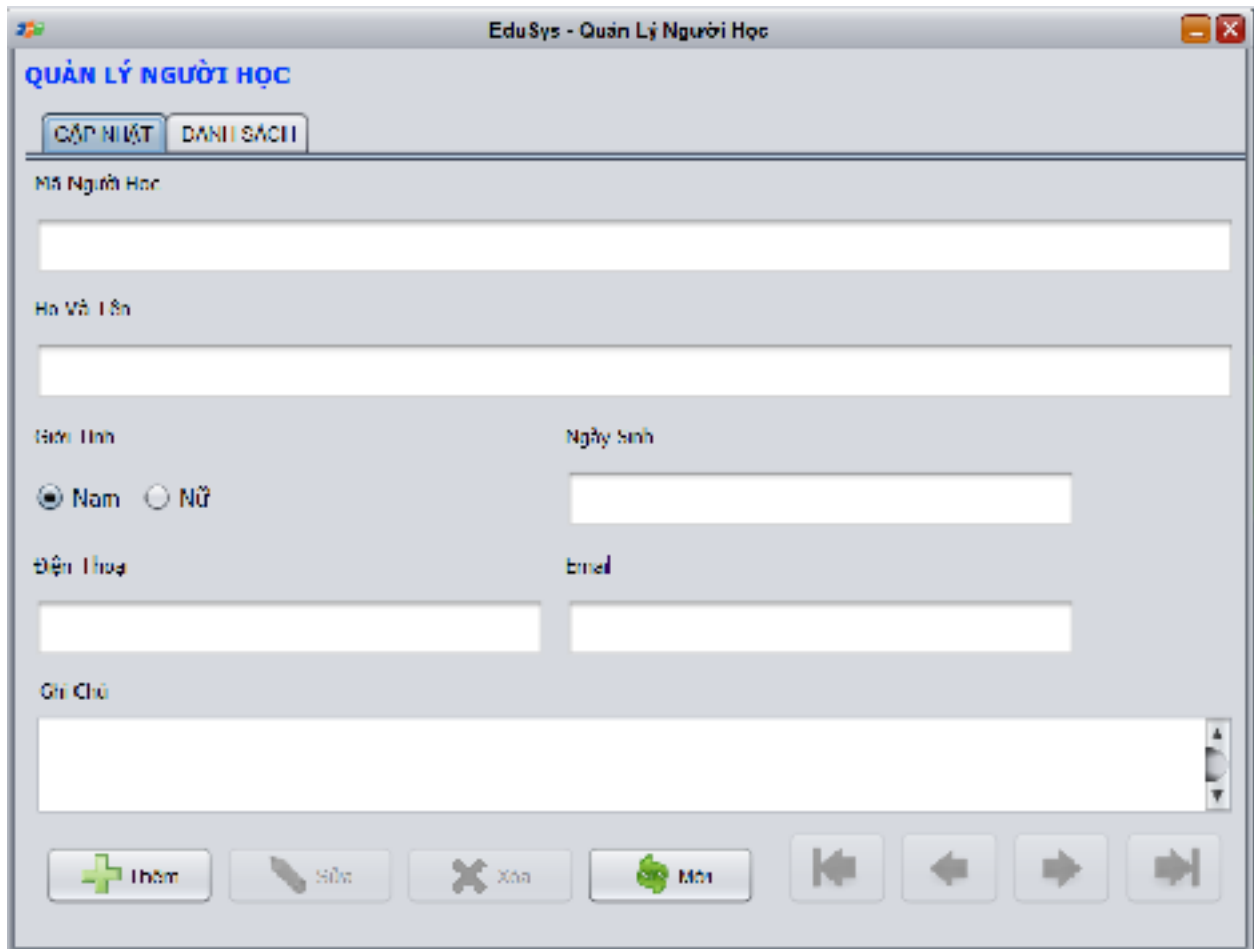
NhanVienDAO dao = new NhanVienDAO();
int row = -1;
void init() {
    this.fillTable();
    this.row = -1;
    this.updateStatus();
}

void insert() {
    NhanVien nv = getForm();
    String mk2 = new String(txtMatKhau2.getPassword());
    if (!Auth.isManager()) {
        MsgBox.alert(this, "Bạn không có quyền thêm nhân viên!");
    } else {
        if (!mk2.equals(nv.getMatKhau())) {
            MsgBox.alert(this, "Xác nhận mật khẩu không đúng!");
            txtMatKhau2.requestFocus();
        } else {
            try {
                dao.insert(nv);
                this.fillTable();
                this.clearForm();
                MsgBox.alert(this, "Thêm mới thành công!");
            } catch (Exception e) {
                MsgBox.alert(this, "Thêm mới thất bại!");
            }
        }
    }
}

void update() {
    NhanVien nv = getForm();
    String mk2 = new String(txtMatKhau2.getPassword());
    if (!Auth.isManager()) {
        MsgBox.alert(this, "Bạn không có quyền sửa nhân viên!");
    } else {
        if (!mk2.equals(nv.getMatKhau())) {
            MsgBox.alert(this, "Xác nhận mật khẩu không đúng!");
        } else {

```

3.3.2.5. Quản lý người học (NguoiHocJFrame)

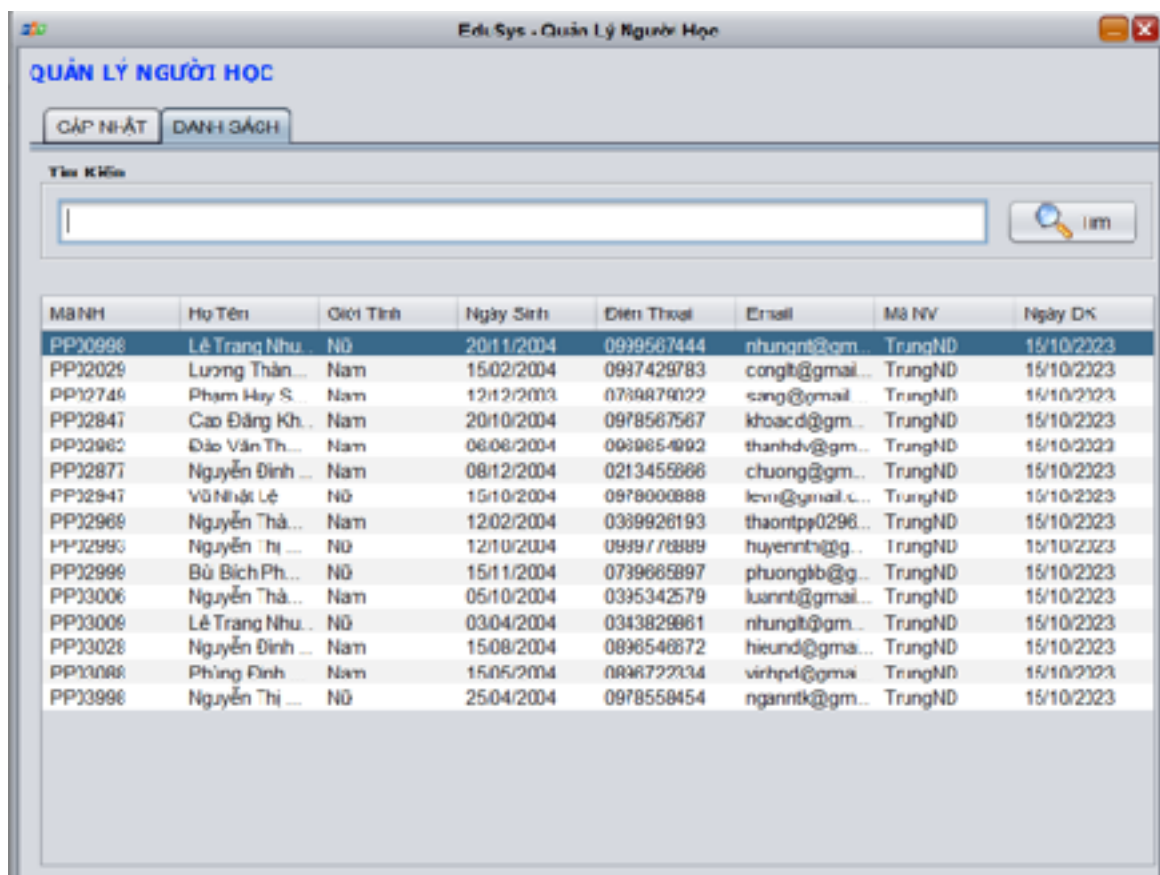


Hình: form trạng thái thêm mới

MÔ TẢ HOẠT ĐỘNG

Sự kiện	Công việc phải làm
Constructor	Init()
btnThemActionPerformed	<pre>if (validateForm(true)) { insert(); }</pre>
btnMoiActionPerformed	clearForm();
btnSuaActionPerformed	<pre>if (validateForm(false)) { update(); }</pre>

btnXoaActionPerformed	delete();
btnFirstActionPerformed	first();
btnPrevActionPerformed	prev();
btnNextActionPerformed	next();
btnLastActionPerformed	last();
btnTimKiemActionPerformed	timkiem();
tblNguoiHocMouseClicked	<pre> if (evt.getClickCount() == 2) { this.row = tblNguoiHoc.rowAtPoint(evt.getPoint()); if (this.row >= 0) { this.edit(); tabs.setSelectedIndex(0); } } </pre>



CÀI ĐẶT MÃ NGUỒN

```

NguoiHocDAO dao = new NguoiHocDAO();
int row = -1;
private void timkiem() {
    this.fillTable();
    this.clearForm();
    this.row = -1;
    this.updateStatus();
}

void init() {
    this.fillTable();
    this.row = -1;
    this.updateStatus();
}

void insert() {
    NguoiHoc nh = getForm();

    try {
        dao.insert(nh);
        this.fillTable();
        this.clearForm();
        MsgBox.alert(this, "Thêm mới thành công!");
    } catch (Exception e) {
        MsgBox.alert(this, "Thêm mới thất bại!" + e);
    }
}

void update() {
    NguoiHoc nh = getForm();

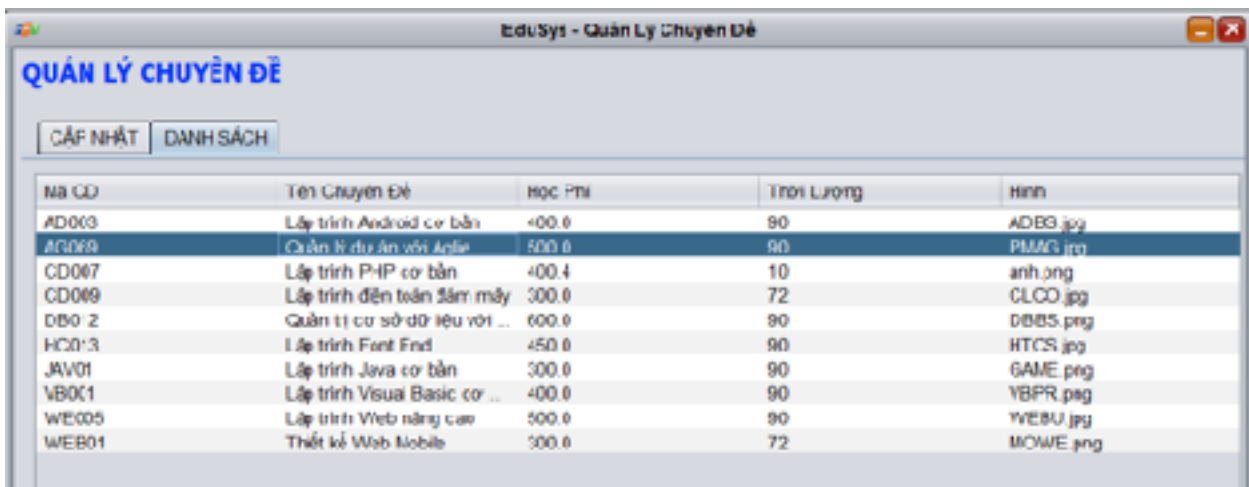
    try {
        dao.update(nh);
        this.fillTable();
        MsgBox.alert(this, "Cập nhật thành công!");
    } catch (Exception e) {
        MsgBox.alert(this, "Cập nhật thất bại!");
    }
}

```

3.3.2.6. Quản lý chuyên đề (ChuyenDeJFrame)

Hình: trạng thái insert

Hình: trạng thái update



Mã CD	Tên Chuyên Đề	Học Phí	Thời Lượng	Hình
AD003	Lập trình Android cơ bản	400.0	90	ADB3.jpg
AD009	Quản lý dự án với Agile	500.0	90	PMAG.jpg
CD007	Lập trình PHP cơ bản	400.4	10	anh.png
CD009	Lập trình điện toán đám mây	300.0	72	CLCD.jpg
DB0 2	Quản trị cơ sở dữ liệu với ...	600.0	90	DBBS.png
HC013	Lập trình Front End	450.0	90	HTCS.jpg
JAV01	Lập trình Java cơ bản	300.0	90	GAME.png
VB001	Lập trình Visual Basic cơ ...	400.0	90	VBPR.png
WE005	Lập trình Web nâng cao	500.0	90	WEBU.jpg
WEB01	Thiết kế Web Mobile	300.0	72	MOWE.png

Hình: bảng dữ liệu

MÔ TẢ HOẠT ĐỘNG

Sự kiện	Công việc phải làm
Constructor	Init()
tblChuyenDeMouseClicked	<pre>if (evt.getClickCount() == 2) { this.row = tblChuyenDe.getSelectedRow(); this.edit(); }</pre>
lblLogoMouseClicked	chonAnh();
btnThemActionPerformed	<pre>if (validateForm(true)) { insert(); }</pre>
btnMoiActionPerformed	clearForm();
btnSuaActionPerformed	<pre>if (validateForm(false)) { update(); }</pre>
btnXoaActionPerformed	delete();
btnFirstActionPerformed	first();
btnPrevActionPerformed	prev();
btnNextActionPerformed	next();
btnLastActionPerformed	last();

CÀI ĐẶT MÃ NGUỒN

```

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

        void chonAnh() {
            if (fileChooser.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
                File file = fileChooser.getSelectedFile();
                XImage.save(file); //luu hinh vao thu muc logos
                ImageIcon icon = XImage.read(file.getName()); //Doc hinh tu logos
                lblLogo.setIcon(icon);
                lblLogo.setToolTipText(file.getName()); //giu ten hinh trong tooltip
            }
        }

        void init() {
            this.fillTable();
            this.row = -1;
            this.updateStatus();
        }

        void insert() {
            ChuyenDe cd = getForm();

            try {
                dao.insert(cd);
                this.fillTable();
                this.clearForm();
                MsgBox.alert(this, "Thêm mới thành công!");
            } catch (Exception e) {
                MsgBox.alert(this, "Thêm mới thất bại!");
            }
        }

        void update() {
            ChuyenDe cd = getForm();

            try {
                dao.update(cd);
                this.fillTable();
            }
        }
    
```

3.3.2.7. Quản lý khóa học (KhoaHocJFrame)

Hình: trạng thái insert

Hình: Trạng thái update

Mã KI	Thời Lượng	Học Phí	Khởi Giảng	Tạo Bởi	Ngày Tạo
1	90	400.0	14/11/2023	TrungND	15/10/2023
2	90	400.0	14/12/2023	TrungND	15/10/2023
3	90	400.0	14/01/2024	TrungND	15/10/2023
4	90	400.0	14/02/2024	TrungND	15/10/2023
5	90	400.0	14/03/2024	TrungND	15/10/2023

Hình: bảng dữ liệu

MÔ TẢ HOẠT ĐỘNG

Sự kiện	Công việc phải làm
Constructor	Init()
lblLogoMouseClicked	chonAnh();
btnThemActionPerformed	<pre> if (validateForm(true)) { insert(); } </pre>
btnMoiActionPerformed	clearForm();
btnSuaActionPerformed	<pre> if (validateForm(false)) { update(); } </pre>
btnXoaActionPerformed	delete();
btnFirstActionPerformed	first();
btnPrevActionPerformed	prev();
btnNextActionPerformed	next();
btnLastActionPerformed	last();
tblChuyenDeMouseClicked	<pre> if (evt.getClickCount() == 2) { this.row = tblChuyenDe.getSelectedRow(); this.edit(); } </pre>

CÀI ĐẶT MÃ NGUỒN

```

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

void chonAnh() {
    if (fileChooser.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
        File file = fileChooser.getSelectedFile();
        XImage.save(file); //luu hinh vao thu muc logos
        ImageIcon icon = XImage.read(file.getName()); //Doc hinh tu logos
        lblLogo.setIcon(icon);
        lblLogo.setToolTipText(file.getName()); //giu ten hinh trong tooltip
    }
}

void init() {
    this.fillTable();
    this.row = -1;
    this.updateStatus();
}

void insert() {
    ChuyenDe cd = getForm();

    try {
        dao.insert(cd);
        this.fillTable();
        this.clearForm();
        MsgBox.alert(this, "Thêm mới thành công!");
    } catch (Exception e) {
        MsgBox.alert(this, "Thêm mới thất bại!");
    }
}

void update() {
    ChuyenDe cd = getForm();

    try {
        dao.update(cd);
        this.fillTable();
    }
}

```

3.3.2.8. Quản lý học viên (HocVienJFrame)

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

The screenshot shows the 'HocVienJFrame' window. At the top, there are two dropdown menus: 'CHUYÊN ĐỘ' (set to 'Lập trình Android cơ bản') and 'KHÓA HỌC' (set to 'AD000 (2023-11-14)'). Below these are two tabs: 'HOC VIÊN' (selected) and 'NGƯỜI HỌC'. The main area contains a table with the following data:

TT	Mã Học Viên	Mã Người Học	Họ Tên	Điểm
1	1	PP00000	Lê Trang Nhung	7.0
2	2	PP00001	Đinh Văn Thuận	5.0
3	3	PP00002	Vũ Nhật Lệ	8.0
4	4	PP00003	Nguyễn Thị Ngọc Huyền	9.0
5	5	PP00004	Nguyễn Thị Kim Ngân	3.0
6	53	PP00005	Nguyễn Thanh Thảo	3.0

At the bottom right, there are two buttons: 'Xóa khỏi Khóa Học' (with a red X icon) and 'Cập Nhật Điểm' (with a pencil icon).

The screenshot shows the 'HocVienJFrame' window. At the top, there are two dropdown menus: 'CHUYÊN ĐỘ' (set to 'Lập trình Android cơ bản') and 'KHÓA HỌC' (set to 'AD000 (2023-11-14)'). Below these are two tabs: 'HOC VIÊN' (selected) and 'NGƯỜI HỌC'. The main area contains a search bar labeled 'Tìm Kiếm' and a table with the following data:

Mã NH	Họ Tên	Giới Tính	Ngày Sinh	Điện Thoại	Email
PP00009	Lương Thành Công	Nam	2004-08-10	0967429733	conglt@gmail.com
PP00010	Phạm Huy Sang	Nam	2003-12-12	0766879022	sangp@gmail.com
PP00011	Cao Đăng Khoa	Nam	2004-10-20	0971567557	khoacd@gmail.com
PP00012	Nguyễn Đình Chú	Nam	2004-12-00	0210465833	chuongd@gmail.com
PP00013	Đỗ Bình Phương	Nữ	2004-11-10	0766002897	phuongbb@gmail.com
PP00014	Nguyễn Thành Luân	Nam	2004-10-05	0367342579	luann@gmail.com
PP00015	Lê Trang Nhung	Nữ	2004-06-03	0341020031	nhunglt@gmail.com
PP00016	Nguyễn Bình Hòa	Nam	2004-08-10	0860640072	hienuc@gmail.com
PP00017	Phùng Đình Vinh	Nam	2004-05-10	0860722334	vinhpd@gmail.com

At the bottom right, there is a button labeled 'Thêm Vào Khóa Học' (with a green plus icon).

MÔ TẢ HOẠT ĐỘNG

Sự kiện	Công việc
Constructor	Init()
cboChuyenDeActionPerformed	fillComboBoxKhoaHoc();
cboKhoaHocActionPerformed	fillTableHocVien();
btnSuaDiemActionPerformed	<pre> try { double diem = 0; for (int i = 0; i < tblHocVien.getRowCount(); i++) { diem = Double.parseDouble(tblHocVien.getValueAt(i, 4).toString()); if (diem < 0) { MsgBox.alert(this, "Điểm phải lớn hơn hoặc bằng 0!"); return; } else if (diem > 10) { MsgBox.alert(this, "Điểm phải nhỏ hơn hoặc bằng 10!"); return; } } if (diem >= 0 diem <= 10) { updateDiem(); } } catch (Exception e) { MsgBox.alert(this, "Điểm không đúng định dạng!"); return; } </pre>
btnXoaHVActionPerformed	removeHocVien();
btnThemHVActionPerformed	addHocVien();
txtTimKiemActionPerformed	fillTableNguoiHoc();

CÀI ĐẶT MÃ NGUỒN

```

ChuyenDeDAO cddao = new ChuyenDeDAO();
KhoaHocDAO khdao = new KhoaHocDAO();
NguoiHocDAO nhdao = new NguoiHocDAO();
HocVienDAO hvdao = new HocVienDAO();

void init() {
    fillComboBoxChuyenDe();
}

void fillComboBoxChuyenDe() {
    DefaultComboBoxModel model = (DefaultComboBoxModel) cboChuyenDe.getModel();
    model.removeAllElements();
    List<ChuyenDe> list = cddao.selectAll();
    for (ChuyenDe cd : list) {
        model.addElement(cd);
    }
    fillComboBoxKhoaHoc();
}

void fillComboBoxKhoaHoc() {
    DefaultComboBoxModel model = (DefaultComboBoxModel) cboKhoaHoc.getModel();
    model.removeAllElements();
    ChuyenDe chuyende = (ChuyenDe) cboChuyenDe.getSelectedItem();
    if (chuyende != null) {
        List<KhoaHoc> list = khdao.selectByChuyenDe(chuyende.getMaCD());
        for (KhoaHoc kh : list) {
            model.addElement(kh);
        }
        fillTableHocVien();
    }
}

void fillTableHocVien() {
    DefaultTableModel model = (DefaultTableModel) tblHocVien.getModel();
    model.setRowCount(0);
    KhoaHoc khoahoc = (KhoaHoc) cboKhoaHoc.getSelectedItem();
    if (khoahoc != null) {
        List<HocVien> list = hvdao.selectByKhoaHoc(khoahoc.getMaKH());
        for (int i = 0; i < list.size(); i++) {
            HocVien hv = list.get(i);

```


3.3.2.9. Thống kê dữ liệu (ThongKeJFrame)

EduSys - Tổng Hợp & Thống Kê

TỔNG HỢP & THỐNG KÊ

BẢNG ĐIỂM NGƯỜI HỌC ĐIỂM CHUYÊN ĐỀ DOANH THU

KHOA HỌC: A0003 (2023-11-14)

Mã NH	Họ Tên	Điểm	Xếp Loại
PP02883	Nguyễn Thị Ngọc Huyền	8.0	Xuất sắc
PP02847	Vũ Nhật Lệ	8.0	Giỏi
PP00880	Lê Trang Nhung	7.0	Khá
PP02882	Hàn Văn Thành	6.0	Trung Bình
PP00880	Nguyễn Thị Kim Ngân	0.0	Chưa đạt
PP02888	Nguyễn Thành Thuận	0.0	Chưa đạt

EduSys - Tổng Hợp & Thống Kê

TỔNG HỢP & THỐNG KÊ

BẢNG ĐIỂM NGƯỜI HỌC ĐIỂM CHUYÊN ĐỀ DOANH THU

Năm	Số NH	ĐK Sớm Nhất	ĐK Muộn Nhất
2023	15	2023-10-15	2023-10-15

EduSys - Tổng Hợp & Thống Kê

TỔNG HỢP & THỐNG KÊ

BẢNG ĐIỂM NGƯỜI HỌC ĐIỂM CHUYÊN ĐỀ DOANH THU

NĂM: 2023

Chuyên Đề	Số KH	Số HV	Doanh Thu	HP TN	HP CN	HP TR
Lập trình An...	1	6	2400.0	400.0	400.0	400.0
Lập trình điệ...	1	6	1800.0	300.0	300.0	300.0
Lập trình Fo...	1	5	2250.0	450.0	450.0	450.0
Lập trình Jav...	1	5	1500.0	300.0	300.0	300.0
Lập trình PHP...	1	5	2002.0	400.4	400.4	400.4
Lập trình Vis...	1	5	2000.0	400.0	400.0	400.0
Lập trình W...	1	6	3000.0	500.0	500.0	500.0
Quản lý dự...	1	7	3500.0	500.0	500.0	500.0
Quản lý cơ...	1	3	1800.0	600.0	600.0	600.0
Thiết kế We...	1	5	1500.0	300.0	300.0	300.0

Chuyên Đề	Số Lượng HV	Điểm TN	Điểm CN	Điểm TB
Lập trình Android c...	6	0.0	9.0	5.0
Lập trình điện toán ...	6	0.0	0.0	0.0
Lập trình Fort End	5	0.0	0.0	0.0
Lập trình Java cơ b...	5	0.0	0.0	0.0
Lập trình PHP cơ b...	5	0.0	0.0	0.0
Lập trình Visual Ba...	5	0.0	0.0	0.0
Lập trình Web nân...	6	0.0	0.0	0.0
Quản lý dự án với ...	7	0.0	8.5	6.228571428571429
Quản trị cơ sở dữ l...	3	0.0	0.0	0.0
Thiết kế Web Mobile	5	0.0	0.0	0.0

MÔ TẢ HOẠT ĐỘNG

Sự kiện	Công việc
Constructor	init();
cboKhoaHocActionPerformed	fillTableBangDiem();
cboNamActionPerformed	fillTableDoanhThu();

CÀI ĐẶT MÃ NGUỒN

```

        ThôngKeDAO tkdao = new ThôngKeDAO();
        KhoaHocDAO khdao = new KhoaHocDAO();

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

        void init() {
            this.selectTab(0);
            if (!Auth.isManager()) {
                tabs.remove(3);
            }

            fillComboBoxKhoaHoc();
            fillTableBangDiem();
            fillTableNguoiHoc();
            fillTableDiemChuyenDe();
            fillComboBoxNam();
            fillTableDoanhThu();
        }

        void fillComboBoxKhoaHoc() {
            DefaultComboBoxModel model = (DefaultComboBoxModel) cboKhoaHoc.getModel();
            model.removeAllElements();
            List<KhoaHoc> list = khdao.selectAll();
            for (KhoaHoc kh : list) {
                model.addElement(kh);
            }
        }

        private String getXepLoai(double diem) {
            if (diem < 5) {
                return "Chưa đạt";
            }
            if (diem < 6.5) {
                return "Trung Bình";
            }
            if (diem < 7.5) {
                return "Khá";
            }
            if (diem < 8) {
                return "Giỏi";
            }
        }
    
```



3.3.2.10. Đóng cửa sổ giới thiệu (GioiThieuJDialog)

Hình: Giới thiệu sản phẩm

MÔ TẢ HOẠT ĐỘNG

Sự kiện	Công việc
Constructor	setLocationRelativeTo(null);
lblLogoMouseClicked	this.dispose();

4. KIỂM THỬ

4.1. 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		

4.2. 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	

4.3. 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	

4.4. KIỂM THỬ FORM QUẢN LÝ KHÓA HỌC

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

4.5. 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	

5. ĐÓNG GÓI VÀ TRIỂN KHAI

5.1. 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

5.2. HƯỚNG DẪN CÀI ĐẶT TRIỂN KHAI

- ✓ Cài SQL Server 2008 trở lên
- ✓ Tạo CSDL Polypro bằng cách chạy file [...sql]
- ✓ Cài đặt JDK 1.8 trở lên
- ✓ Chạy file setup

5.3. 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 dẫn sử dụng trong menu trợ giúp của phần mềm