



DỰ ÁN MẪU

ASSM

Giảng viên: Lê Văn Phụng

Sinh viên: Tô Minh Trí - PS15506

Lớp: IT16303

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.....	7
2.1	Mô hình triển khai.....	7
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.....	9
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.....	12
3	Thực hiện viết mã.....	38
3.1	Viết mã tạo CSDL.....	38
3.1.1	Tạo CSDL.....	38
3.1.2	SQL truy vấn và thao tác bảng.....	40
3.1.3	Các thủ tục lưu tổng hợp thống kê.....	41
3.2	Lập trình JDBC.....	43
3.2.1	Lớp hỗ trợ.....	43
3.2.2	Model class - Các lớp mô tả dữ liệu.....	46
3.2.3	DAO Class - Các lớp truy xuất dữ liệu.....	48
3.3	Viết mã cho ứng dụng.....	60
3.3.1	Các lớp tiện ích.....	60
3.3.2	Viết mã cho các chức năng.....	62
4	Kiểm thử.....	108
4.1	Kiểm thử form quản lý nhân viên.....	108
4.2	Kiểm thử form quản lý người học.....	108
4.3	Kiểm thử form quản lý chuyên đề.....	108
4.4	Kiểm thử form quản lý Khóa học.....	109
4.5	Kiểm thử form quản lý học viên.....	109

5	Đóng gói và triển khai.....	109
5.1	Hướng dẫn chuyển đổi jar thành exe.....	109
5.2	Hướng dẫn cài đặt triển khai.....	109
5.3	Hướng dẫn sử dụng phần mềm.....	109

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.

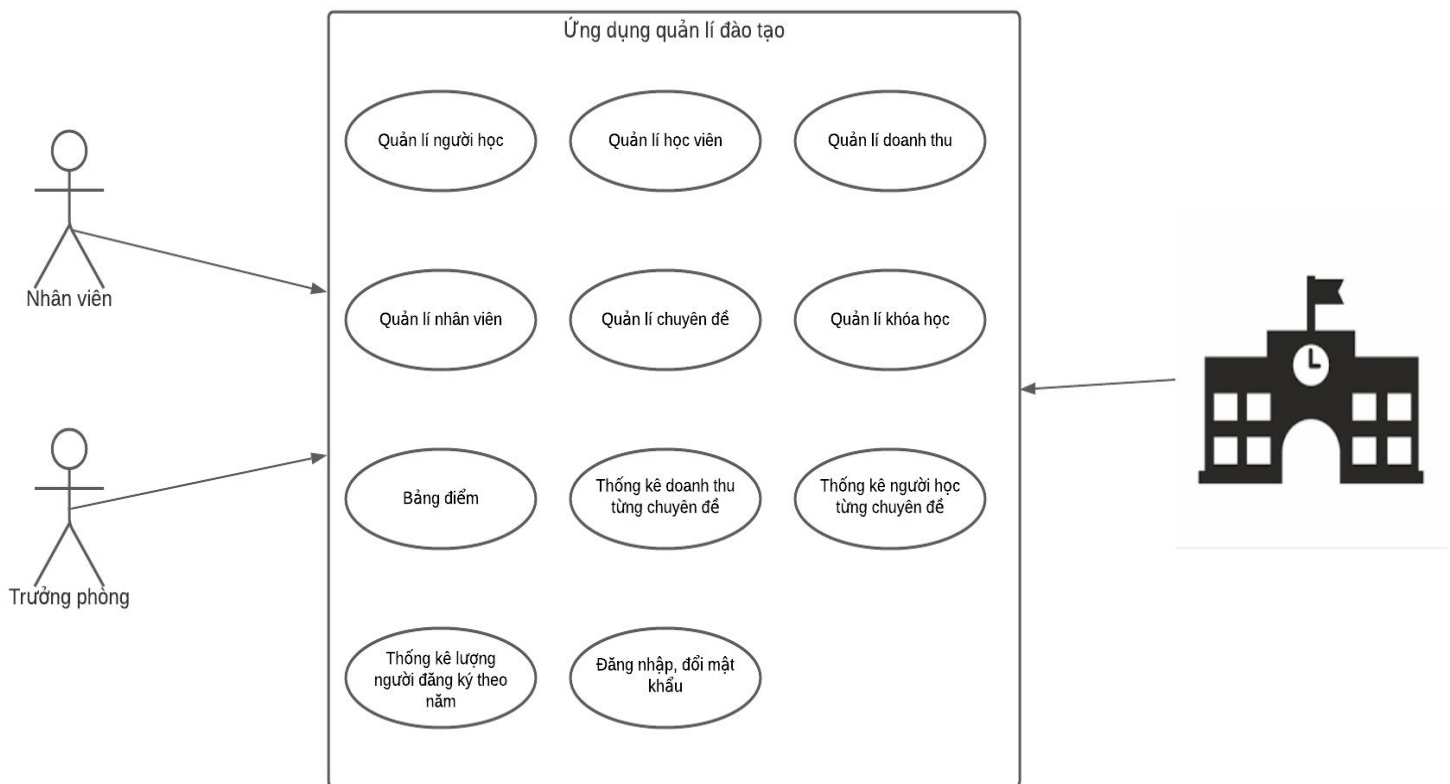
1.2 YÊU CẦU HỆ THỐNG

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

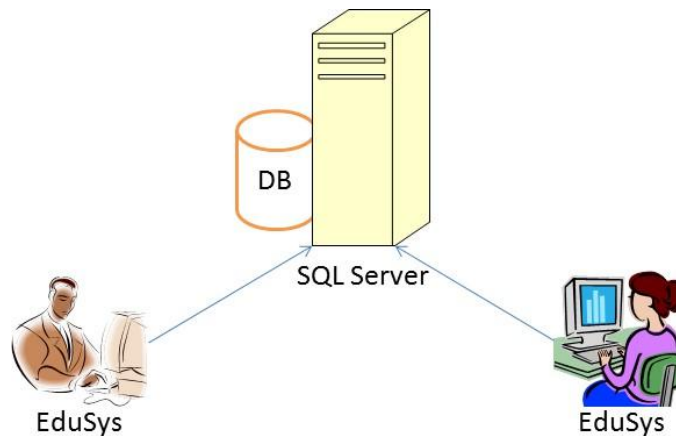
- Điểm trung bình
- Doanh thu từng chuyên đề theo năm phải có cấu trúc:
 - Chuyên đề
 - Số khóa
 - Số học viên
 - Doanh thu
 - Học phí thấp nhất
 - Học phí cao nhất
 - Học phí trung bình
- Thống kê số người học từng năm phải có cấu trúc:
 - Năm
 - Số học viên
 - Ngày đăng ký sớm nhất
 - Ngày đăng ký 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

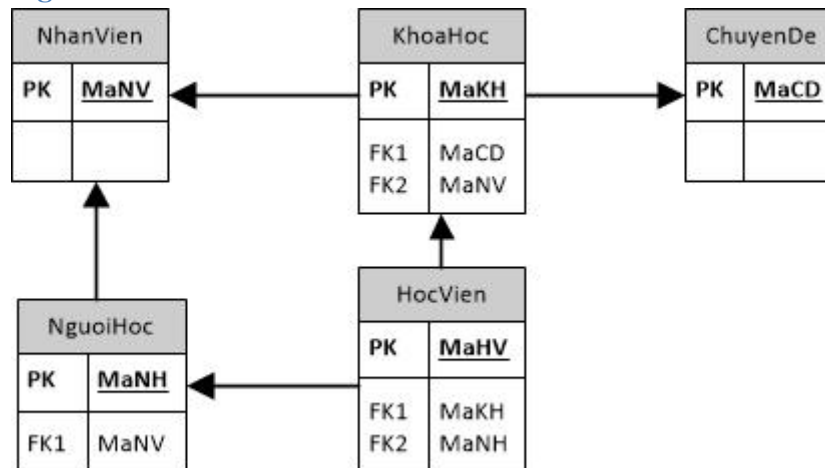
2.2.1 Sơ đồ quan hệ thực thể

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

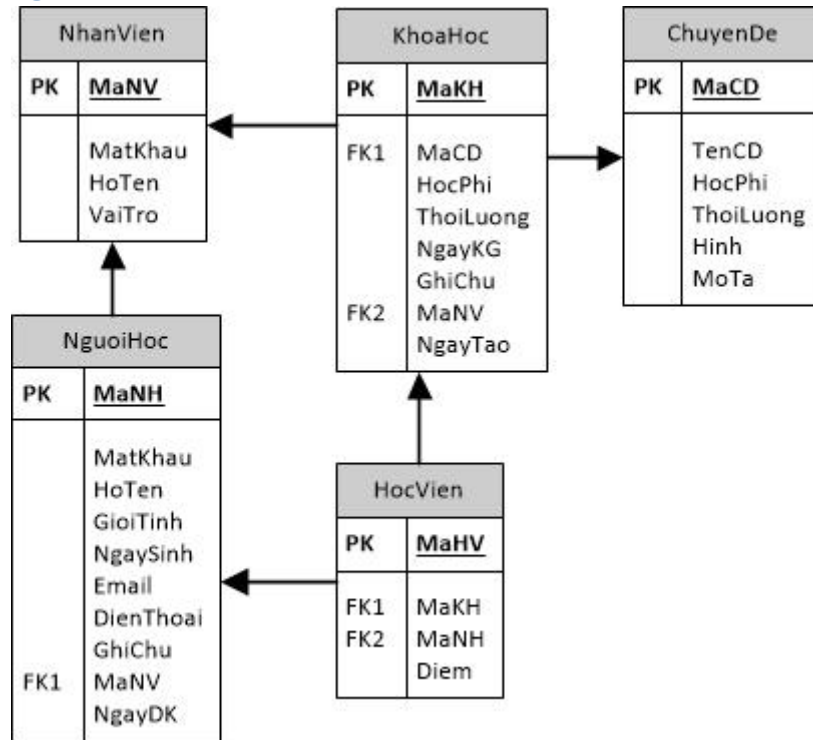
- ✓ Chuyên đề: quản lý chuyên đề đào tạo ngắn hạn
- ✓ Khóa học: quản lý thông tin khóa học được tạo ra từ chuyên đề. Có nghĩa là mỗi chuyên đề có thể tạo ra nhiều khóa học.
- ✓ Người học: quản lý thông tin của người đăng ký học
- ✓ Học viên: Học viên là người học đăng ký học chuyên đề. Mỗi học viên có 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 level2



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
MatKhou	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 ảnh
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
NgayDK	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 HọcVien 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

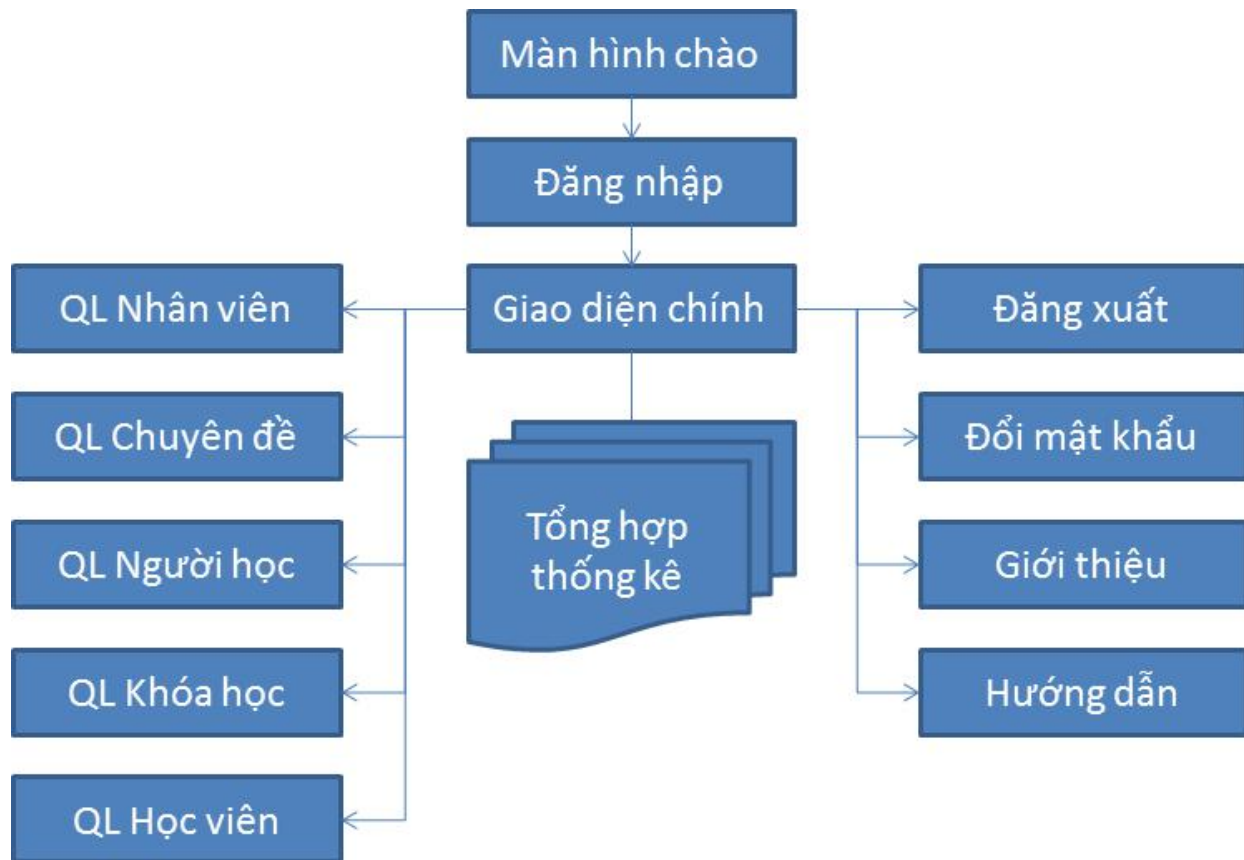
2.3 THIẾT KẾ GIAODIỆN

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 của các bạn.

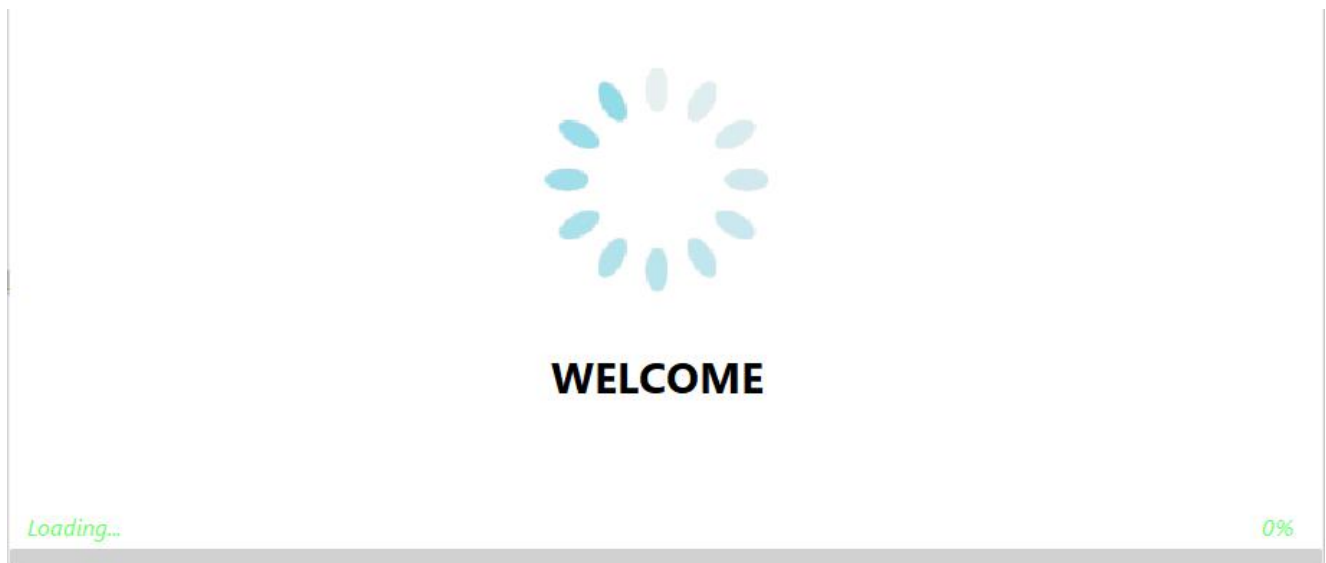
Lưu ý quan trọng:

- ✓ 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
TextField, 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	
...		

2.3.2.1 Màn hình chào

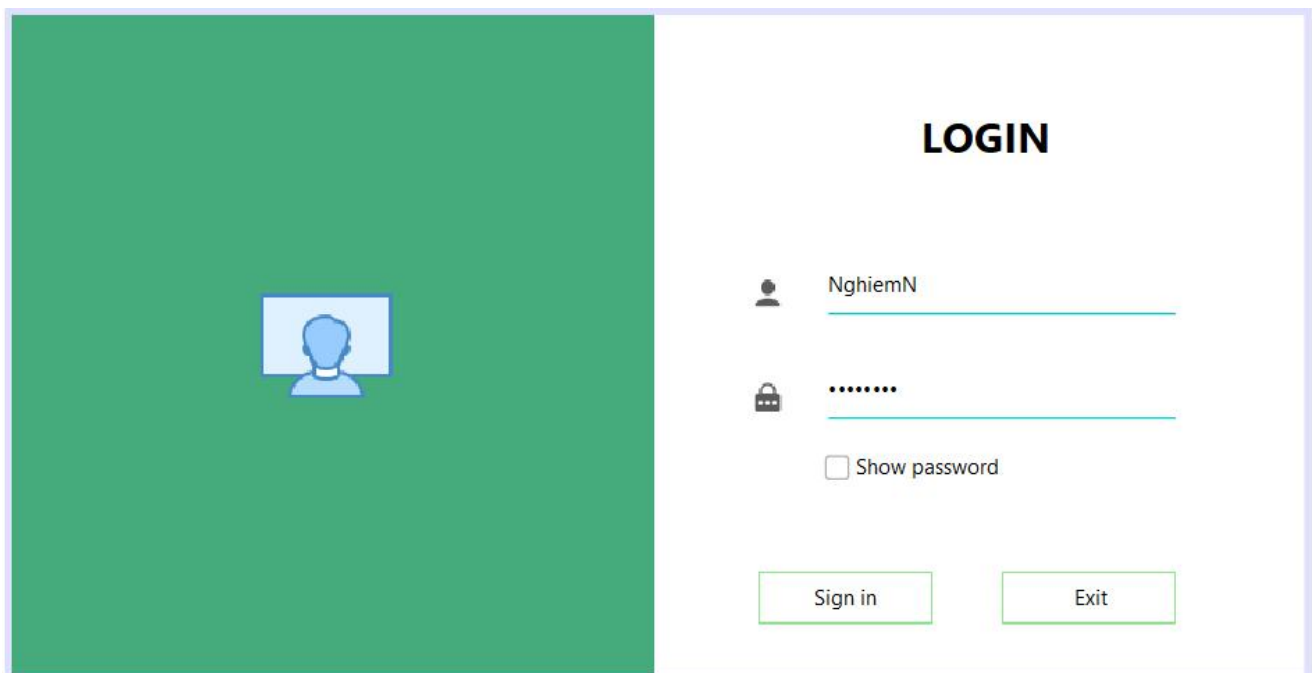


Các thành phần giao diện

	Thành phần	Kiểu	Thuộc tính
--	------------	------	------------

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

2.3.2.2 Đăng nhập

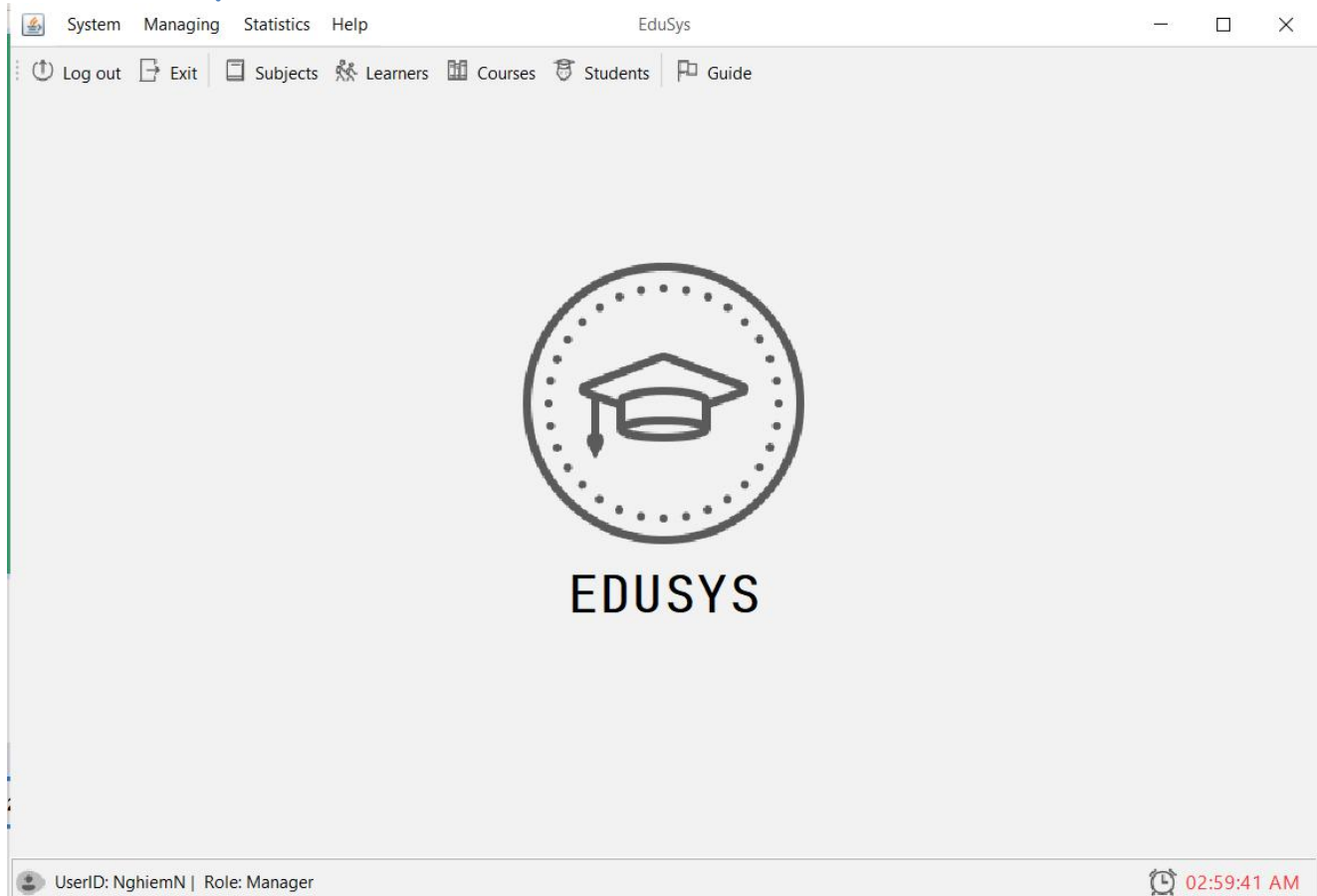


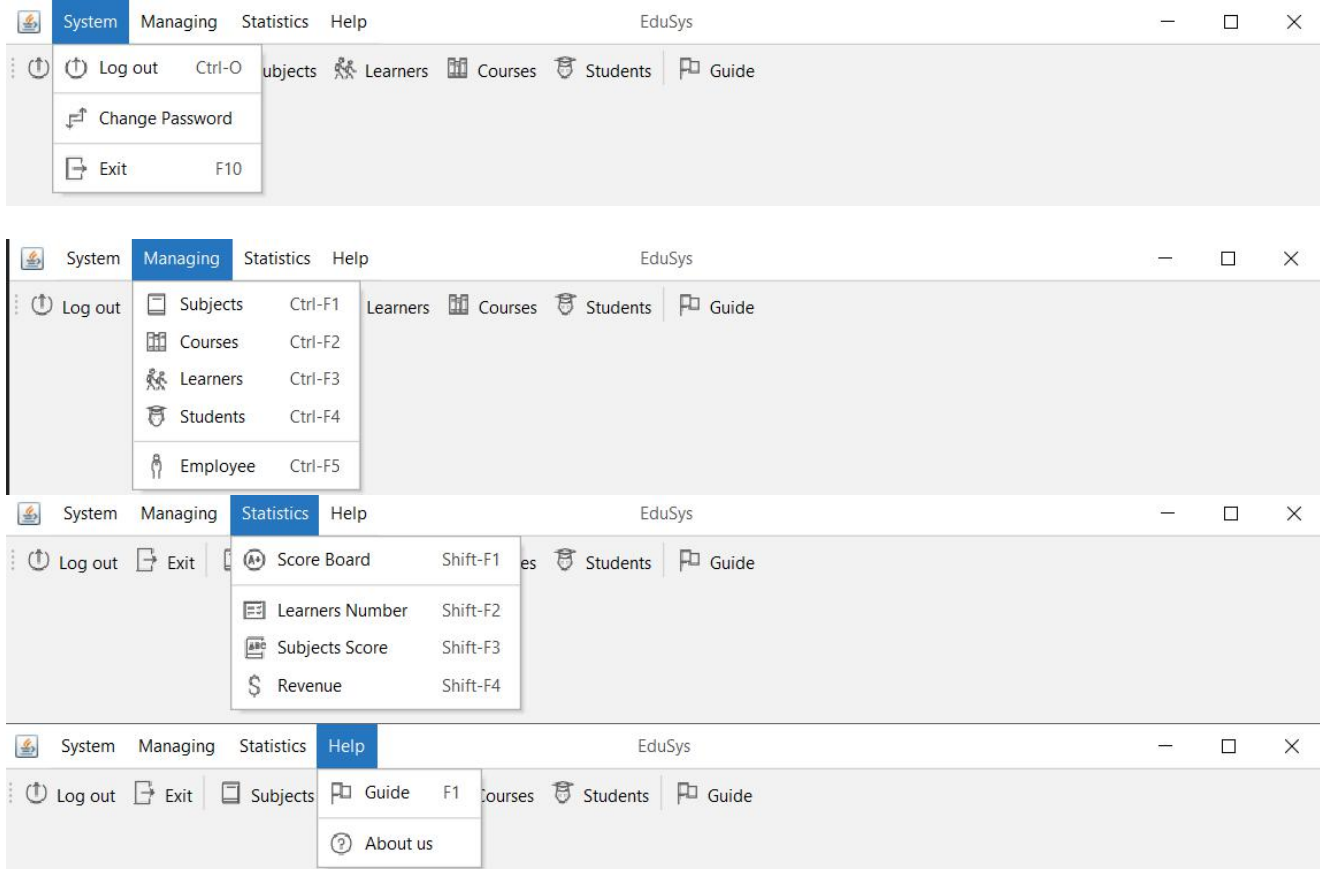
Các thành phần giao diện

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

3	lblImage	JLabel	Icon: Security.png Layout.Direction:before
4	pnlForm	JPanel	Layout.Derection:center
5	lblMaNV	JLabel	Text: Tên đăng nhập
6	txtMaNV	JTextField	
7	lblMatKhau	JLabel	Text: Mật khẩu
8	txtMatKhau	JPasswordField	
9	btnDangNhap	JButton	Text:Đăng nhập
10	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: 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: Clie 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 t

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

2.3.2.4 Quản lý nhân viên


Manage Employees
×

MANAGE EMPLOYEES

Edit
List

Employee ID

Password


Retype

Username

Role
☐ Manager
☐ Employee

Add
Edit
Delete
New

<<
<
>
>>

 Manage Employees
×

MANAGE EMPLOYEES			
Edit		List	
Emp ID	Password	Username	Role
LongNDH	*****	Nguyễn Đình Hoàng Lo...	Employee
LongNDT	*****	Nguyễn Đình Thiên Long	Employee
NghiemN	*****	Nguyễn Nghiệm	Manager
nhanvien	*****	nhanvien1	Employee
NoPT	*****	Phạm Thị Nở	Employee
PheoNC	*****	Nguyễn Chí Phèo	Employee
TeoNV	*****	Nguyễn Văn Tèo	Manager
ThaoLTH	*****	Lê Thị Hương Thảo	Employee

Các thành phần giao diện

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

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

Learners

MANAGE LEARNERS

Edit
List

Learner ID

Telephone

Name

Birthday

Gender
Male

Email

Description

Add
Edit
Delete
New

<<
<
>
>>

Learners

MANAGE LEARNERS

Edit
List

Search
Search

ID	Name	Gender	Birthday	Phone	Email	EmplID	Creation Day
PS01638	LỮ HUY CỨ...	Female	08-05-1991	0928768265	PS01638@fp...	TeoNV	Sun Oct 17 0...
PS02037	ĐỖ VĂN MI...	Male	24-10-1992	0968095685	PS02037@fp...	PheoNC	Sun Oct 17 0...
PS02771	NGUYỄN TẮ...	Male	15-09-1998	0927594734	PS02771@fp...	PheoNC	Sun Oct 17 0...
PS02867	NGUYỄN H...	Male	27-10-1997	0946984711	PS02867@fp...	TeoNV	Sun Oct 17 0...
PS02930	TRẦN VĂN ...	Male	03-06-2000	0924774498	PS02930@fp...	TeoNV	Sun Oct 17 0...
PS02979	ĐOÀN TRẦN...	Male	28-08-1994	0912374818	PS02979@fp...	TeoNV	Sun Oct 17 0...
PS02983	NGUYỄN H...	Male	04-04-1993	0912499836	PS02983@fp...	PheoNC	Sun Oct 17 0...
PS02988	HỒ HỮU H...	Male	08-02-1993	0924984876	PS02988@fp...	PheoNC	Sun Oct 17 0...
PS03031	PHAN TẤN ...	Male	05-04-1990	0924832716	PS03031@fp...	PheoNC	Sun Oct 17 0...
PS03046	NGUYỄN CA...	Male	28-01-1990	0977117727	PS03046@fp...	PheoNC	Sun Oct 17 0...
PS03080	HUỖNH TH...	Male	06-09-1994	0916436052	PS03080@fp...	PheoNC	Sun Oct 17 0...
PS03088	NGUYỄN H...	Male	02-09-1991	0938101529	PS03088@fp...	PheoNC	Sun Oct 17 0...
PS03096	ĐOÀN HỮU...	Male	21-02-1994	0945196719	PS03096@fp...	PheoNC	Sun Oct 17 0...
PS03104	LÊ THÀNH P...	Female	21-02-1993	0922948096	PS03104@fp...	TeoNV	Sun Oct 17 0...


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: [0,0,204] 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

2.3.2.6 Quản lý chuyên đề



Manage Subjects

×

MANAGE SUBJECTS

Edit
List

Picture



Subject ID
JAV01

Name
Lập trình Java cơ bản

Duration (Hour)
90

Fees
250.0

Description

JAV01 - Lập trình Java cơ bản

Add
Edit
Delete
New

<<
←
→
>>

Manage Subjects

MANAGE SUBJECTS

Edit List

Subject ID	Name	Fees	Duration	Picture
JAV01	Lập trình Java cơ ...	250.0	90	GAME.png
JAV02	Lập trình Java nân...	300.0	90	HTCS.jpg
JAV03	Lập trình mạng vớ...	200.0	70	INMA.jpg
JAV04	Lập trình desktop ...	200.0	70	ADAV.jpg
PRO01	Dự án với công n...	300.0	90	MOWE.png
PRO02	Dự án với công n...	300.0	90	Subject.png
PRO03	Dự án với công n...	300.0	90	GAME.png
PRO04	Dự án với Angular...	300.0	90	HTCS.jpg
PRO05	Dự án với Swing ...	300.0	90	INMA.jpg
PRO06	Dự án với Windo...	300.0	90	LAYO.jpg
RDB01	Cơ sở dữ liệu SQL ...	100.0	50	MOWE.png
RDB02	Lập trình JDBC	150.0	60	Subject.png
RDB03	Lập trình cơ sở dữ...	250.0	80	GAME.png
SER01	Lập trình web với ...	350.0	100	HTCS.jpg
SER02	Lập trình Spring M...	400.0	110	INMA.jpg
SER03	Lập trình MS.NET ...	400.0	110	LAYO.jpg
SER04	Xây dựng Web AP...	200.0	70	MOWE.png
WEB01	Thiết kế web với ...	200.0	70	Subject.png
WEB02	Thiết kế web với B...	0.0	40	GAME.png
WEB03	Lập trình front-en...	150.0	60	HTCS.jpg

Các thành phần giao diện

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

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

2.3.2.7 Quản lý khóa học

Courses

MANAGE COURSES

SUBJECTS

Lập trình desktop với Swing

Edit List

Subject

Lập trình desktop với Swing

Fees

200.0

Creator

First day of school

Duration (Hour)

70


Day of creation

Description

JAV04 - Lập trình desktop với Swing

Add Edit Delete New

<< < > >>

 Courses
 ×

MANAGE COURSES

SUBJECTS

▼

Edit List

Course ID	Duration	Fees	FDOS	Creator	Day Of Creation
2	90	300.0	11-02-2018	TeoNV	31-12-2017
3	50	100.0	12-03-2018	NghiemN	31-12-2017
4	80	250.0	01-04-2018	TeoNV	31-12-2017
1008	70	200.0	12-04-2020	TeoNV	02-05-2020
1015	70	200.0	07-10-2019	NghiemN	11-10-2021
1017	70	200.0	28-10-2021	NghiemN	11-10-2021
1020	70	200.0	29-01-1970	NghiemN	16-10-2021

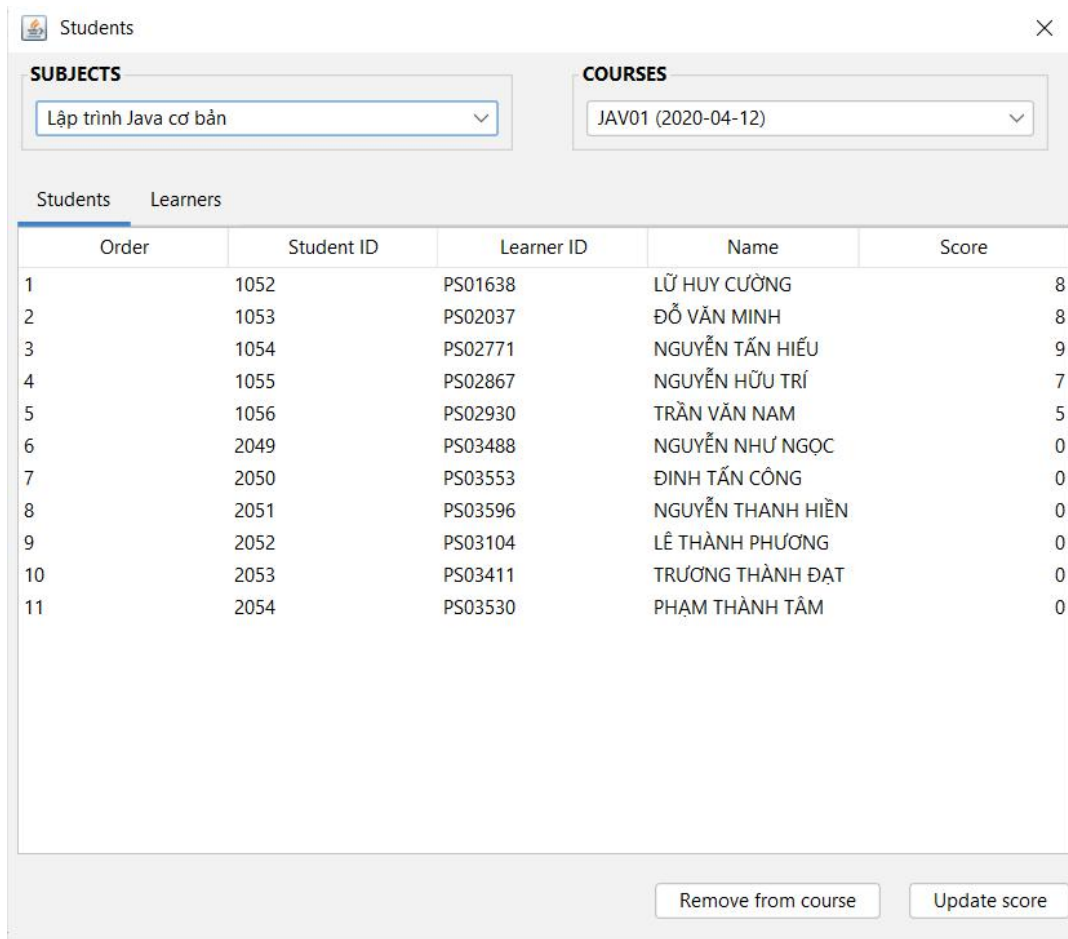
Các thành phần giao diện

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

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

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

2.3.2.8 Quản lý học viên



Order	Student ID	Learner ID	Name	Score
1	1052	PS01638	LỮ HUY CƯỜNG	8
2	1053	PS02037	ĐỖ VĂN MINH	8
3	1054	PS02771	NGUYỄN TẤN HIẾU	9
4	1055	PS02867	NGUYỄN HỮU TRÍ	7
5	1056	PS02930	TRẦN VĂN NAM	5
6	2049	PS03488	NGUYỄN NHƯ NGỌC	0
7	2050	PS03553	ĐINH TẤN CÔNG	0
8	2051	PS03596	NGUYỄN THANH HIỀN	0
9	2052	PS03104	LÊ THÀNH PHƯƠNG	0
10	2053	PS03411	TRƯƠNG THÀNH ĐẠT	0
11	2054	PS03530	PHẠM THÀNH TÂM	0


Các thành phần giao diện

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

5	txtDiem	TextField	
---	---------	-----------	--

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

2.3.2.9 Tổng hợp thống kê

 Statistics



STATISTIC

Score board

Learners

Subject Score

Revenue

Course

PRO02 (2018-01-10)



Learner ID	Name	Score	Classify
PS02037	ĐỖ VĂN MINH	9.0	Excellent
PS01638	LỮ HUY CƯỜNG	8.0	Good
PS02771	NGUYỄN TẤN HIẾU	8.0	Good
PS02930	TRẦN VĂN NAM	7.0	Above average
PS02983	NGUYỄN HOÀNG THIÊN ...	6.0	Average
PS02979	ĐOÀN TRẦN NHẬT VŨ	4.0	Not Qualified
PS02867	NGUYỄN HỮU TRÍ	3.0	Not Qualified
PS02988	HỒ HỮU HẬU	0.0	Not Qualified
PS03031	PHAN TẤN VIỆT	0.0	Not Qualified
PS03046	NGUYỄN CAO PHƯỚC	0.0	Not Qualified
PS03080	HUỖNH THANH HUY	0.0	Not Qualified
PS03088	NGUYỄN HOÀNG TRUNG	0.0	Not Qualified
PS03096	ĐOÀN HỮU KHANG	0.0	Not Qualified
PS03104	LÊ THÀNH PHƯƠNG	0.0	Not Qualified
PS03120	PHẠM NGỌC NHẬT TRƯ...	0.0	Not Qualified
PS03130	ĐẶNG BẢO VIỆT	0.0	Not Qualified



Statistics



STATISTIC

Score board

Learners

Subject Score

Revenue

Year	Learners Number	Earliest Signup	Latest Signup
1990	8	1990-01-02	1990-04-27
1991	3	1991-05-08	1991-09-02
1992	5	1992-02-15	1992-10-24
1993	6	1993-02-08	1993-12-01
1994	4	1994-02-21	1994-09-06
1995	5	1995-01-05	1995-05-09
1996	6	1996-02-04	1996-12-11
1997	3	1997-02-09	1997-10-27
1998	2	1998-04-08	1998-09-15
1999	3	1999-02-24	1999-07-09
2000	5	2000-03-10	2000-10-14




Statistics



STATISTIC

Score board Learners **Subject Score** Revenue

Subject	Students Number	Lowest	Highest	Average
Dự án với AngularJS...	27	0.0	9.0	0.52
Dự án với công ngh...	15	0.0	0.0	0.00
Dự án với công ngh...	16	0.0	0.0	0.00
Dự án với công ngh...	16	0.0	9.0	2.81
Lập trình desktop v...	13	0.0	9.0	4.15
Lập trình Java cơ bả...	31	0.0	10.0	3.03
Lập trình Java nâng ...	7	0.0	9.0	2.57
Lập trình mạng với J...	9	0.0	0.0	0.00

 Statistics
 ×

STATISTIC						
Score board		Learners	Subject Score	Revenue		
Year	2020					
Subject	Course Num...	Student Num...	Revenue	Lowest	Highest	Average
Dự án với An...	2	27	8100.0	300.0	300.0	300.00
Dự án với cô...	1	15	4500.0	300.0	300.0	300.00
Dự án với cô...	1	16	4800.0	300.0	300.0	300.00
Lập trình Jav...	3	31	8900.0	250.0	300.0	287.10
Lập trình Jav...	1	7	2100.0	300.0	300.0	300.00
Lập trình mậ...	1	9	1800.0	200.0	200.0	200.00

Print Excel

Các thành phần giao diện

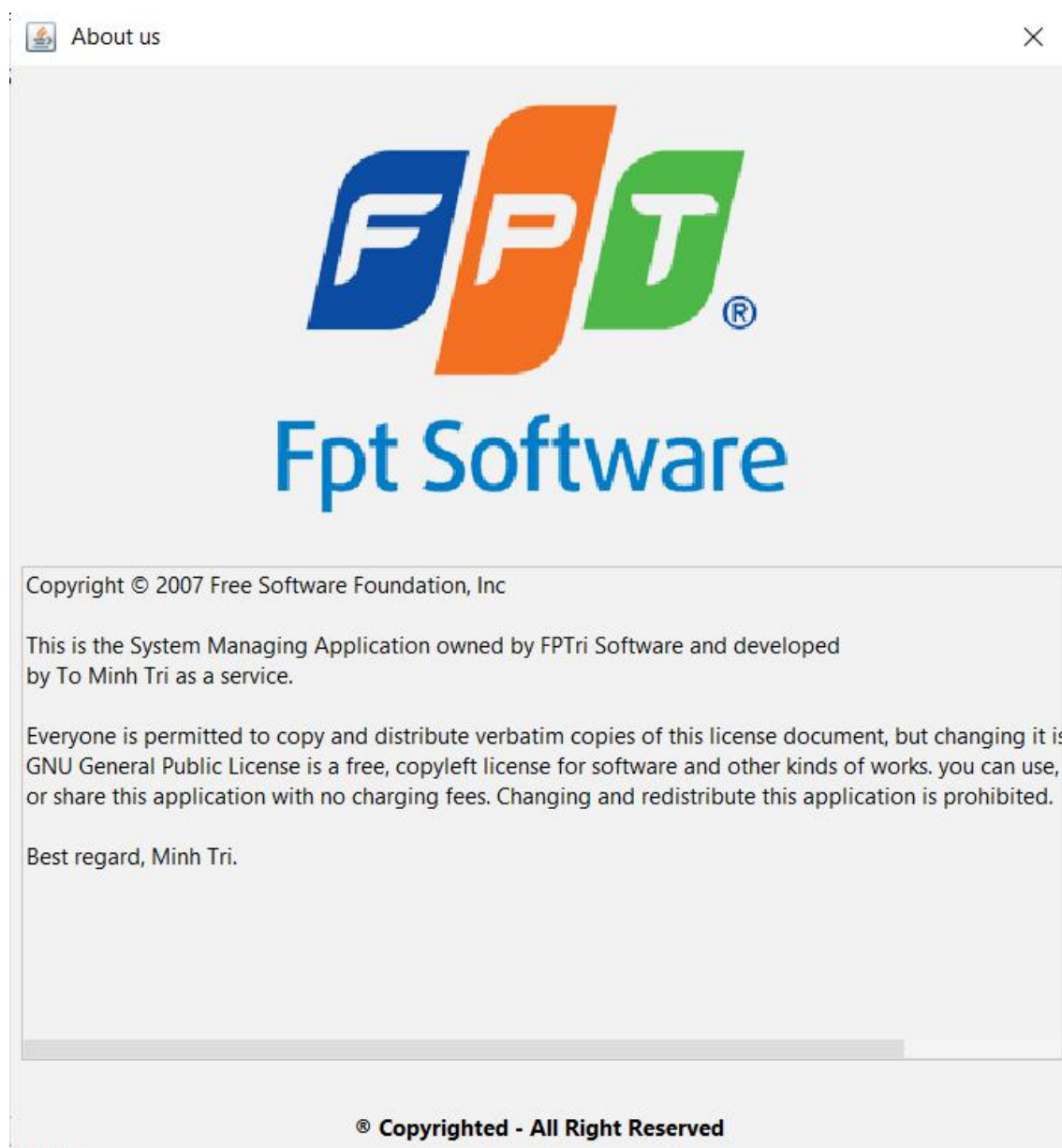
	Component	Kiểu	Thuộc tính
1	ThongKeJFrame	JFrame	Title: TỔNG HỢP – THỐNG KÊ defaultCloseOperation:DISPONSE
2	lblTitle	JLabel	Text: TỔNG HỢP – THỐNG KÊ
3	tabs	JTabbedPane	
4	pnlNguoiHoc	JPanel	Tab Title: NGƯỜI HỌC Layout: BorderLayout

5	tblNguoiHoc	JTable	Model: như hình Height: 25
6	pnlBangDiem	JPanel	Tab Title: BẢNG ĐIỂM Layout: BorderLayout

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

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

2.3.2.10 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
--	--	--	-----------------

2.3.2.11 Hướng dẫn sử dụng



Thiết kế giao diện:

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

3 THỰC HIỆN VIẾT MÃ

3.1 VIẾT MÃ TẠO CSDL

3.1.1 Tạo CSDL

```
CREATE DATABASE Polypro
GO
USE Edusys
GO
CREATE TABLE NhanVien(
    MaNV nvarchar(50) NOT NULL,
    MatKhu nvarchar(50) NOT NULL,
```

```

        HoTen nvarchar(50) NOT NULL,
        VaiTro bit NOT NULL DEFAULT 0,
        PRIMARY KEY(MaNV)
    )
GO
CREATE TABLE ChuyenDe(
    MaCD nchar(5) NOT NULL, TenCD
    nvarchar(50) NOT NULL,
    HocPhi float NOT NULL DEFAULT 0,
    ThoiLuong int NOT NULL DEFAULT 30,
    Hinh nvarchar(50) NOT NULL DEFAULT 'chuyen-de.png', MoTa
    nvarchar(255) NOT NULL,
    PRIMARY KEY(MaCD),
    UNIQUE(TenCD),
    CHECK(HocPhi >= 0 AND ThoiLuong > 0)
)
GO
CREATE TABLE NgoiHoc(
    MaNH nchar(7) NOT NULL, HoTen
    nvarchar(50) NOT NULL,
    NgaySinh date NOT NULL,
    GioiTinh bit NOT NULL DEFAULT 0,
    DienThoai nvarchar(50) NOT NULL,
    Email nvarchar(50) NOT NULL,
    GhiChu nvarchar(max) NULL,
    MaNV nvarchar(50) NOT NULL,
    NgayDK date NOT NULL DEFAULT getdate(),
    PRIMARY KEY(MaNH)
)
GO
CREATE TABLE KhoaHoc(
    MaKH int IDENTITY(1,1) NOT NULL,
    MaCD nchar(5) NOT NULL,
    HocPhi float NOT NULL DEFAULT 0,
    ThoiLuong int NOT NULL DEFAULT 0,
    NgayKG date NOT NULL,
    GhiChu nvarchar(50) NULL,
    MaNV nvarchar(50) NOT NULL,
    NgayTao date NOT NULL DEFAULT getdate(),
    PRIMARY KEY(MaKH),
    CHECK(HocPhi >= 0 AND ThoiLuong > 0),
    FOREIGN KEY (MaCD) REFERENCES ChuyenDe(MaCD) ON UPDATE CASCADE, FOREIGN
    KEY (MaNV) REFERENCES NhanVien(MaNV) ON UPDATE CASCADE
)
GO
CREATE TABLE HocVien(
    MaHV int IDENTITY(1,1) NOT NULL,

```

```

MaKH int NOT NULL,
MaNH nchar(7) NOT NULL,
Diem float NOT NULL,
PRIMARY KEY(MaHV),
UNIQUE(MaKH, MaNH),
FOREIGN KEY (MaKH) REFERENCES KhoaHoc(MaKH) ON DELETE CASCADE,
FOREIGN KEY (MaNH) REFERENCES NguoiHoc(MaNH) ON UPDATE CASCADE
)

```

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, MatKhai, HoTen, VaiTro) VALUES (?, ?, ?, ?)
Cập nhật theo mã	UPDATE NhanVien SET MatKhai=?, 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	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=?

3.1.2.5 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=?

3.1.3 Các thủ tục lưu tổng hợp thống kê

3.1.3.1 Số người học từng năm

```

CREATE PROC sp_ThongKeNguoiHoc
AS BEGIN
    SELECT
        YEAR(NgayDK) Nam,
        COUNT(*) SoLuong,
        MIN(NgayDK) DauTien,
        MAX(NgayDK) CuoiCung
    FROM NguoiHoc
    GROUP BY YEAR(NgayDK)
END

```

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;

/**
 *
 * @author balis
 */
public class XDate {

    public static SimpleDateFormat formatter = new SimpleDateFormat();
```

```
public static Date toDate(String date, String pattern) {
    try {
        formatter.applyPattern(pattern);
        return formatter.parse(date);
    } catch (ParseException e) {
        throw new RuntimeException(e);
    }
}

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

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

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.*;

/**
 *
 * @author balis
 */
public class XJdbc {

    public static String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";
    public static String dburl =
"jdbc:sqlserver://localhost:1433;databaseName=EDUSYS_ASSM";
    public static String username = "sa";
    public static String password = "songlong";

    static {
        try {
            Class.forName(driver);
        } catch (ClassNotFoundException e) {
            throw new RuntimeException(e);
        }
    }

    public static PreparedStatement preparedStatement(String sql, Object... args) throws
SQLException {
        Connection conn = DriverManager.getConnection(dburl, username, password);
        PreparedStatement pstmt = null;
        if (sql.trim().startsWith("{")) {
            pstmt = conn.prepareCall(sql); //proc
        } else {
            pstmt = conn.prepareStatement(sql); //SQL
        }
        for (int i = 0; i < args.length; i++) {
            pstmt.setObject(i + 1, args[i]);
        }
        return pstmt;
    }

    public static ResultSet executeQuery(String sql, Object... args) {
        try {
            PreparedStatement pstmt = preparedStatement(sql, args);
            try {
                return pstmt.executeQuery();
            } finally {
            }
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
    }
}
```

```

    }

    public static void executeUpdate(String sql, Object... args) {
        try {
            PreparedStatement pstmt = preparedStatement(sql, args);
            try {
                pstmt.executeUpdate();
            } finally {
                pstmt.getConnection().close();
            }
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
    }
}

```

3.2.1.3 *Lớp tiện ích XImage*

```

package com.edusys.utils;

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

/**
 *
 * @author balis
 */
public class XImage {

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

    public static void save(File src) {
        File dir = new File("logos", src.getName());
        if (!dir.exists()) {
            dir.mkdirs();
        }
    }
}

```

```

    }
    try {
        Path source = Paths.get(src.getAbsolutePath());
        Path destination = Paths.get(dir.getAbsolutePath());
        Files.copy(source, destination, StandardCopyOption.REPLACE_EXISTING);
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}

public static ImageIcon read(String filename) {
    File path = new File("logos", filename);
    return new ImageIcon(new
ImageIcon(path.getAbsolutePath()).getImage().getScaledInstance(180, 180,
Image.SCALE_DEFAULT));
}
}

```

3.2.1.4 *Lớp tiện ích MsgBox*

```

package com.edusys.utils;

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

/**
 *
 * @author balis
 */
public class MsgBox {

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

    public static boolean confirm(Component parent, String message) {
        int result = JOptionPane.showConfirmDialog(parent, message, "EduSys",
JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
        return result == JOptionPane.YES_OPTION;
    }
}

```

```

    }

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

```

3.2.2 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.

3.2.2.1 *NhanVien*

```

public class NhanVien {

    private String maNV;
    private String matKhai;
    private String hoTen;
    private boolean vaiTro = false;

    @Override
    public String toString() {
        return this.hoTen;
    }

    public String getMaNV() {
        return maNV;
    }

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

    public String getMatKhai() {
        return matKhai;
    }

    public void setMatKhai(String matKhai) {
        this.matKhai = matKhai;
    }

    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*

```
public class ChuyenDe {

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

    @Override
    public String toString() {
        return this.tenCD;
    }

    @Override
    public boolean equals(Object obj) {
        ChuyenDe other = (ChuyenDe) obj;
        return other.getMaCD().equals(this.getMaCD());
    }

    public String getMaCD() {
        return maCD;
    }

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

    public String getTenCD() {
        return tenCD;
    }

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

    public double getHocPhi() {
        return hocPhi;
    }

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

    public int getThoiLuong() {
        return thoiLuong;
    }
}
```

```
}  
  
public void setThoiLuong(int thoiLuong) {  
    this.thoiLuong = thoiLuong;  
}  
  
public String getHinh() {  
    return hinh;  
}  
  
public void setHinh(String hinh) {  
    this.hinh = hinh;  
}  
  
public String getMoTa() {  
    return moTa;  
}  
  
public void setMoTa(String moTa) {  
    this.moTa = moTa;  
}  
}
```


3.2.2.3 *NguoiHoc*

```
public class NguoiHoc {

    private String maNH;
    private String hoTen;
    private Date ngaySinh;
    private boolean gioiTinh;
    private String dienThoai;
    private String email;
    private String ghiChu;
    private String maNV;
    private Date ngayDK = new Date();

    @Override
    public String toString() {
        return this.hoTen;
    }

    public String getMaNH() {
        return maNH;
    }

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

    public String getHoTen() {
        return hoTen;
    }

    public void setHoTen(String hoTen) {
```

```
this.hoTen = hoTen;

}

public Date getNgaySinh() {
    return ngaySinh;
}

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

public boolean isGioiTinh() {
    return gioiTinh;
}

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

public String getDienThoai() {
    return dienThoai;
}

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

public String getEmail() {
    return email;
}

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

```
}
```

```
public String getGhiChu() {  
    return ghiChu;  
}
```

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

```
public String getMaNV() {  
    return maNV;  
}
```

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

```
public Date getNgayDK() {  
    return ngayDK;  
}
```

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

```
}
```

3.2.2.4 *KhoaHoc*

```
public class KhoaHoc {  
  
    private int maKH;  
    private String maCD;  
    private double hocPhi;  
    private int thoiLuong;  
    private Date ngayKG;  
    private String ghiChu;
```

```
private String maNV;  
private Date ngayTao = new Date();  
  
@Override  
public String toString() {  
    return this.maCD + " (" + this.ngayKG + ")";  
}  
  
public int getMaKH() {  
    return maKH;  
}  
  
public void setMaKH(int maKH) {  
    this.maKH = maKH;  
}  
  
public String getMaCD() {  
    return maCD;  
}  
  
public void setMaCD(String maCD) {  
    this.maCD = maCD;  
}  
  
public double getHocPhi() {  
    return hocPhi;  
}  
  
public void setHocPhi(double hocPhi) {  
    this.hocPhi = hocPhi;  
}  
  
public int getThoiLuong() {  
    return thoiLuong;  
}  
  
public void setThoiLuong(int thoiLuong) {  
    this.thoiLuong = thoiLuong;  
}  
  
public Date getNgayKG() {  
    return ngayKG;  
}  
  
public void setNgayKG(Date ngayKG) {  
    this.ngayKG = ngayKG;  
}  
  
public String getGhiChu() {  
    return ghiChu;  
}
```

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

public String getMaNV() {
    return maNV;
}

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

public Date getNgayTao() {
    return ngayTao;
}

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

}
```

3.2.2.5 *HocVien*

```
public class HocVien {  
  
    private int maHV;  
    private int maKH;  
    private String maNH;  
    private double diem = -1.0;
```

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

```
import java.util.List;  
  
/**  
 *  
 * @author balis  
 * @param <EntityType>  
 * @param <KeyType>  
 */  
abstract public class EduSysDAO<EntityType, KeyType> {  
  
    abstract public void insert(EntityType entity);  
  
    abstract public void update(EntityType entity);  
  
    abstract public void delete(KeyType id);  
  
    abstract public EntityType selectById(KeyType id);  
  
    abstract public List<EntityType> selectAll();  
  
    abstract protected List<EntityType> selectBySQL(String sql, Object... args);  
}
```

Sau đây là các lớp DAO cụ thể

3.2.3.1 *NhanVienDAO*

```
public class NhanVienDAO extends EduSysDAO<NhanVien, String> {

    String INSERT_SQL = "INSERT INTO NhanVien(MaNV, MatKhau, HoTen, VaiTro)
        VALUES(?,?,?,?)";
    String UPDATE_SQL = "UPDATE NhanVien SET MatKhau=?, HoTen=?, VaiTro=? WHERE
        MaNV=?";
    String DELETE_SQL = "DELETE FROM NhanVien WHERE MaNV=?";
    String SELECT_ALL_SQL = "SELECT * FROM NhanVien";
    String SELECT_BY_ID_SQL = "SELECT * FROM NhanVien WHERE MaNV=?";

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

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

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

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

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

```
@Override
protected List<NhanVien> selectBySQL(String sql, Object... args) {
    List<NhanVien> list = new ArrayList<>();
    try {
        ResultSet rs = XJdbc.executeQuery(sql, args);
        while (rs.next()) {
            NhanVien entity = new NhanVien();
            entity.setMaNV(rs.getString("MaNV"));
            entity.setHoTen(rs.getString("HoTen"));
            entity.setMatKhau(rs.getString("MatKhau"));
            entity.setVaiTro(rs.getBoolean("VaiTro"));
            list.add(entity);
        }
        rs.getStatement().getConnection().close();
        return list;
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}
```


3.2.3.2 ChuyenDeDAO

```
public class ChuyenDeDAO extends EduSysDAO<ChuyenDe, String> {

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

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

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

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

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

    @Override
    public List<ChuyenDe> selectAll() {
```

```
        return this.selectBySQL(SELECT_ALL_SQL);
    }

    @Override
    protected List<ChuyenDe> selectBySQL(String sql, Object... args) {
        List<ChuyenDe> list = new ArrayList<>();
        try {
            ResultSet rs = XJdbc.executeQuery(sql, args);
            while (rs.next()) {
                ChuyenDe entity = new ChuyenDe();
                entity.setMaCD(rs.getString("MaCD"));
                entity.setTenCD(rs.getString("TenCD"));
                entity.setHocPhi(rs.getDouble("HocPhi"));
                entity.setThoiLuong(rs.getInt("ThoiLuong"));
                entity.setHinh(rs.getString("Hinh"));
                entity.setMoTa(rs.getString("MoTa"));
                list.add(entity);
            }
            rs.getStatement().getConnection().close();
            return list;
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}
```

3.2.3.3 *NguoiHocDAO*

```
public class NguoiHocDAO extends EduSysDAO<NguoiHoc, String> {

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

    @Override
    public void insert(NguoiHoc entity) {
        XJdbc.executeUpdate(INSERT_SQL,
            entity.getMaNH(),
            entity.getHoTen(),
            entity.getNgaySinh(),
            entity.isGioiTinh(),
            entity.getDienThoai(),
            entity.getEmail(),
            entity.getGhiChu(),
            entity.getMaNV());
    }

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

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

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

```

        return list.get(0);
    }

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

    @Override
    protected List<NguoiHoc> selectBySQL(String sql, Object... args) {
        List<NguoiHoc> list = new ArrayList<>();
        try {
            ResultSet rs = XJdbc.executeQuery(sql, args);
            while (rs.next()) {
                NguoiHoc entity = new NguoiHoc();
                entity.setMaNH(rs.getString("MaNH"));
                entity.setHoTen(rs.getString("HoTen"));
                entity.setNgaySinh(rs.getDate("NgaySinh"));
                entity.setGioiTinh(rs.getBoolean("GioiTinh"));
                entity.setDienThoai(rs.getString("DienThoai"));
                entity.setEmail(rs.getString("Email"));
                entity.setGhiChu(rs.getString("GhiChu"));
                entity.setMaNV(rs.getString("MaNV"));
                list.add(entity);
            }
            rs.getStatement().getConnection().close();
            return list;
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
    }

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

    public List<NguoiHoc> selectNotInCourse(int makh, String keyword) {
        String SQL = "SELECT * FROM NguoiHoc WHERE HoTen LIKE ? AND "
            + "MaNH NOT IN(SELECT MaNH FROM HocVien WHERE MaKH = ?)";
        return this.selectBySQL(SQL, "%" + keyword + "%", makh);
    }
}

```

3.2.3.4 *KhoaHocDAO*

```
public class KhoaHocDAO extends EduSysDAO<KhoaHoc, Integer> {

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

    @Override
    public void insert(KhoaHoc entity) {
        XJdbc.executeUpdate(INSERT_SQL,
            entity.getMaCD(),
            entity.getHocPhi(),
            entity.getThoiLuong(),
            entity.getNgayKG(),
            entity.getGhiChu(),
            entity.getMaNV());
    }

    @Override
    public void update(KhoaHoc entity) {
        XJdbc.executeUpdate(UPDATE_SQL,
            entity.getMaCD(),
            entity.getHocPhi(),
            entity.getThoiLuong(),
            entity.getNgayKG(),
            entity.getGhiChu(),
            entity.getMaNV(),
            entity.getMaKH());
    }

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

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

```

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

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

public List<KhoaHoc> selectByChuyenDe(String macd) {
    String SQL = "SELECT * FROM KhoaHoc WHERE MaCD = ?";
    return this.selectBySQL(SQL, macd);
}

public List<Integer> selectYears() {
    String SQL = "SELECT DISTINCT year(NgayKG) Year FROM KhoaHoc ORDER BY
Year DESC";
    List<Integer> list = new ArrayList<>();
    try {
        ResultSet rs = XJdbc.executeQuery(SQL);
        while (rs.next()) {
            list.add(rs.getInt(1));
        }
        rs.getStatement().getConnection().close();
        return list;
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}
}

```

3.2.3.5 HocVienDAO

```
public class HocVienDAO extends EduSysDAO<HocVien, Integer> {

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

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

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

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

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

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

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

```
        try {
            ResultSet rs = XJdbc.executeQuery(sql, args);
            while (rs.next()) {
                HocVien entity = new HocVien();
                entity.setMaHV(rs.getInt("MaHV"));
                entity.setMaKH(rs.getInt("MaKH"));
                entity.setMaNH(rs.getString("MaNH"));
                entity.setDiem(rs.getDouble("Diem"));
                list.add(entity);
            }
            rs.getStatement().getConnection().close();
            return list;
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
    }

    public List<HocVien> selectByKhoaHoc(int makh) {
        String SQL = "SELECT * FROM HocVien WHERE MaKH = ?";
        return this.selectBySQL(SQL, makh);
    }
}
```


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.

```
public class ThongKeDAO {

    private List<Object[]> getListOfArray(String sql, String[] cols, Object...
args) {
        try {
            List<Object[]> list = new ArrayList<>();
            ResultSet rs = XJdbc.executeQuery(sql, args);
            while (rs.next()) {
                Object[] vals = new Object[cols.length];
                for (int i = 0; i < cols.length; i++) {
                    vals[i] = rs.getObject(cols[i]);
                }
                list.add(vals);
            }
            rs.getStatement().getConnection().close();
            return list;
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
    }

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

    public List<Object[]> getLuongNguoiHoc() {
        String sql = "{CALL sp_LuongNguoiHoc}";
        String[] cols = {"Nam", "SoLuong", "DauTien", "CuoiCung"};
        return this.getListOfArray(sql, cols);
    }

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

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

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 *DialogHelper*

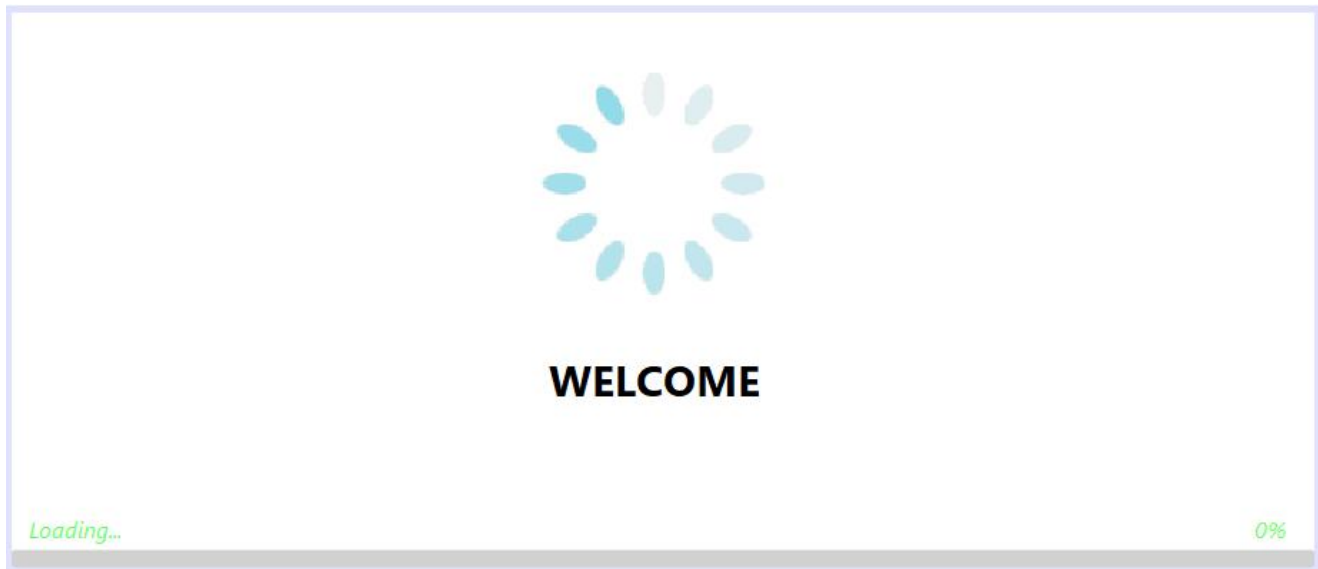
Lớp DialogHelper chứa các phương thức đơn giản hóa JOptionPane

```
public class MsgBox {  
  
    public static void alert(Component parent, String message) {  
        JOptionPane.showMessageDialog(parent, message, "EduSys",  
JOptionPane.INFORMATION_MESSAGE);  
    }  
  
    public static boolean confirm(Component parent, String message) {  
        int result = JOptionPane.showConfirmDialog(parent, message, "EduSys",  
JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);  
        return result == JOptionPane.YES_OPTION;  
    }  
  
    public static String prompt(Component parent, String message) {  
        return JOptionPane.showInputDialog(parent, message, "EduSys",  
JOptionPane.INFORMATION_MESSAGE);  
    }  
}
```

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
runLoadingBar()	Init

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

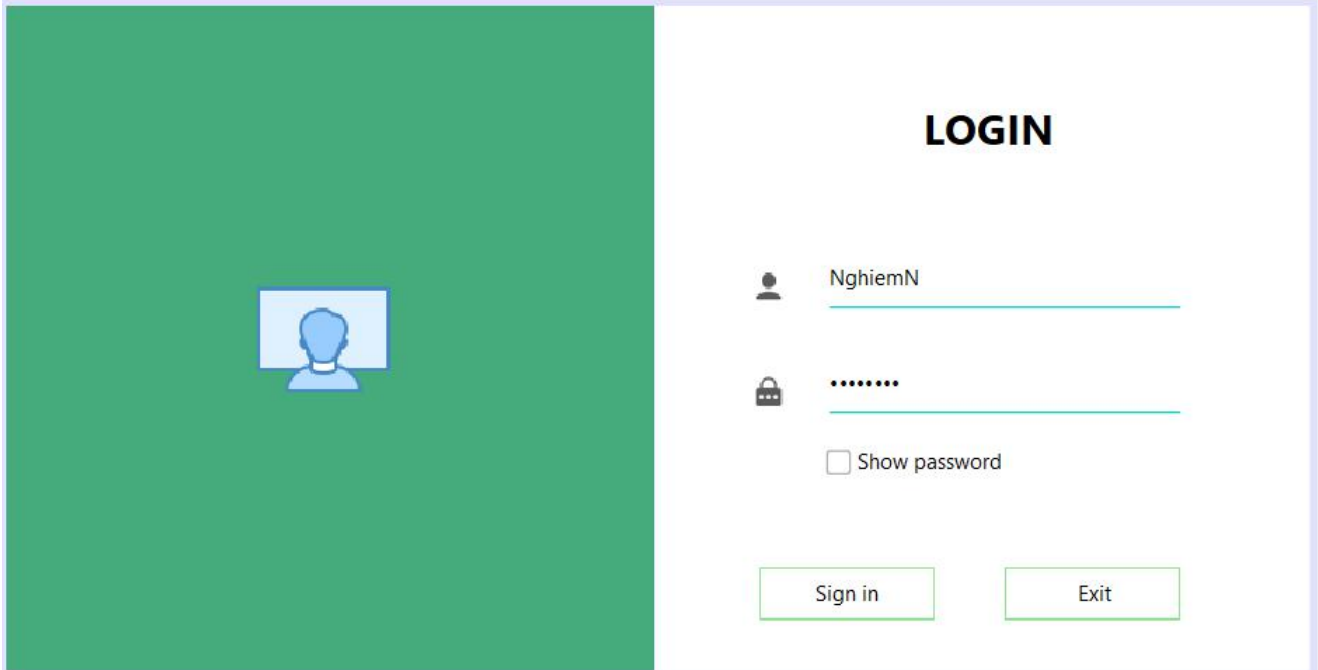
```
private void runLoadingBar() {
    class Loading extends Thread {

        @Override
        public void run() {

            for (int i = 0; i <= 100; i++) {
                try {
                    Thread.sleep(20);
                    lblLoadingValue.setText(i + "%");
                    progressBar.setValue(i);
                    if (i == 100) {
                        dispose();
                        new MainJFrame().setVisible(true);
                    }
                } catch (InterruptedException ex) {

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

3.3.2.2 Form đăng nhập (DangNhapJFrame)



MÔ TẢ HOẠT ĐỘNG

Sự kiện	Công việc
Constructor	Init()
btnDangNhap	Login()
btnKetThuc	Exit()

HƯỚNG DẪN CÀI ĐẶT MÃ

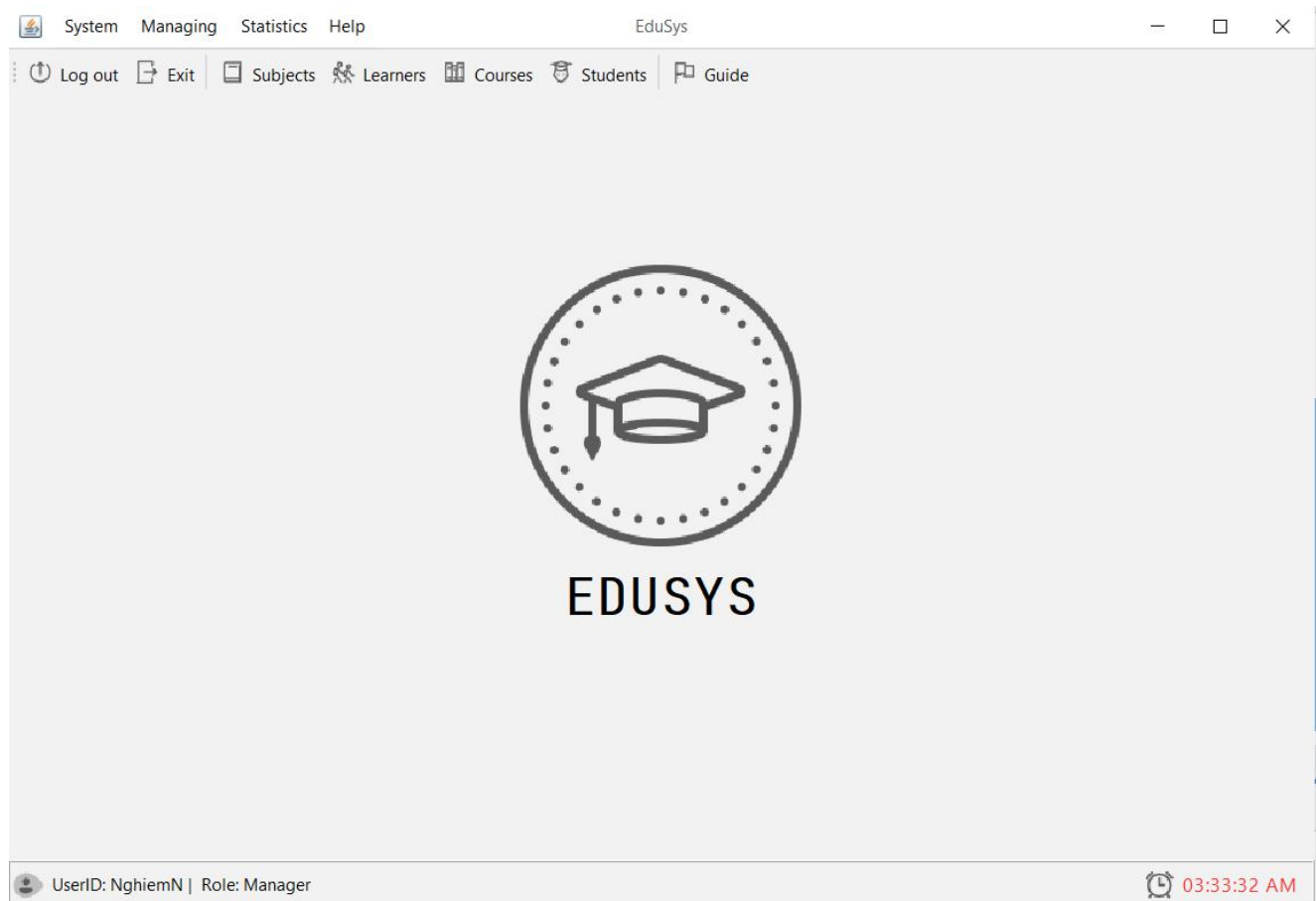
Bổ sung mã sau vào cuối lớp DangNhapJFrame.

```

void login() {
    String mamv = txtUser.getText();
    String password = new String(txtPassword.getPassword());
    NhanVien nhanVien = dao.selectById(mamv);
    if (nhanVien == null) {
        MsgBox.alert(this, "Wrong username!");
    } else if (!password.equals(nhanVien.getMatKhau())) {
        MsgBox.alert(this, "Wrong password!");
    } else {
        Auth.user = nhanVien;
        this.dispose();
        new GreetJDialog(this, rootPaneCheckingEnabled).setVisible(true);
    }
}

```

3.3.2.3 *Giao diện chính (MainJFrame)*



MÔ TẢ HOẠT ĐỘNG

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

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

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

```

void startClock() {
    class TimeClock extends Thread {

        @Override
        public void run() {
            while (true) {
                lblClock.setText(new SimpleDateFormat("hh:mm:ss
a").format(Calendar.getInstance().getTime()));
            }
        }
    }
    TimeClock timeClock = new TimeClock();
    timeClock.start();
}

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

void openChangePassword() {
    if (Auth.isLogin()) {
        new ChangePasswordJDialog(this,
rootPaneCheckingEnabled).setVisible(true);
    } else {
        MsgBox.alert(this, "Please login!");
    }
}

void LogOut() {
    Auth.clear();
    this.dispose();
    new LoginJDialog(this, rootPaneCheckingEnabled).setVisible(true);
}

```

```

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

void openSubjects() {
    new SubjectJDialog(this, rootPaneCheckingEnabled).setVisible(true);
}

void openCourses() {
    new CourseJDialog(this, rootPaneCheckingEnabled).setVisible(true);
}

void openLearners() {
    new LearnerJDialog(this, rootPaneCheckingEnabled).setVisible(true);
}

void openStudents() {
    new StudentJDialog(this, rootPaneCheckingEnabled).setVisible(true);
}

void openEmployees() {
    new EmployeeJDialog(this, rootPaneCheckingEnabled).setVisible(true);
}

void openGuide() {
    try {
        Desktop.getDesktop().browse(new
URI("https://www.kansastag.gov/advhtml_doc_upload/caplio_500se_software_user_guid
e.pdf"));
    } catch (IOException | URISyntaxException ex) {
        Logger.getLogger(MainJFrame.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

void openAboutUs() {
    new AboutJDialog(this, rootPaneCheckingEnabled).setVisible(true);
}

void openStatistics(int index) {
    if (Auth.isLogin()) {
        if (index == 3 && !Auth.isManager()) {
            MsgBox.alert(this, "You're not authorized to view revenue tab!");
        } else {
            StatisticJDialog statisticJDialog = new StatisticJDialog(this,
true);
            statisticJDialog.selectTab(index);
            statisticJDialog.setVisible(true);
        }
    }
}

```




```

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

```

3.3.2.4 Quản lý nhân viên (NhanVienJFrame)


Manage Employees
×

MANAGE EMPLOYEES

Edit
List

Employee ID

Password

Retype

Username

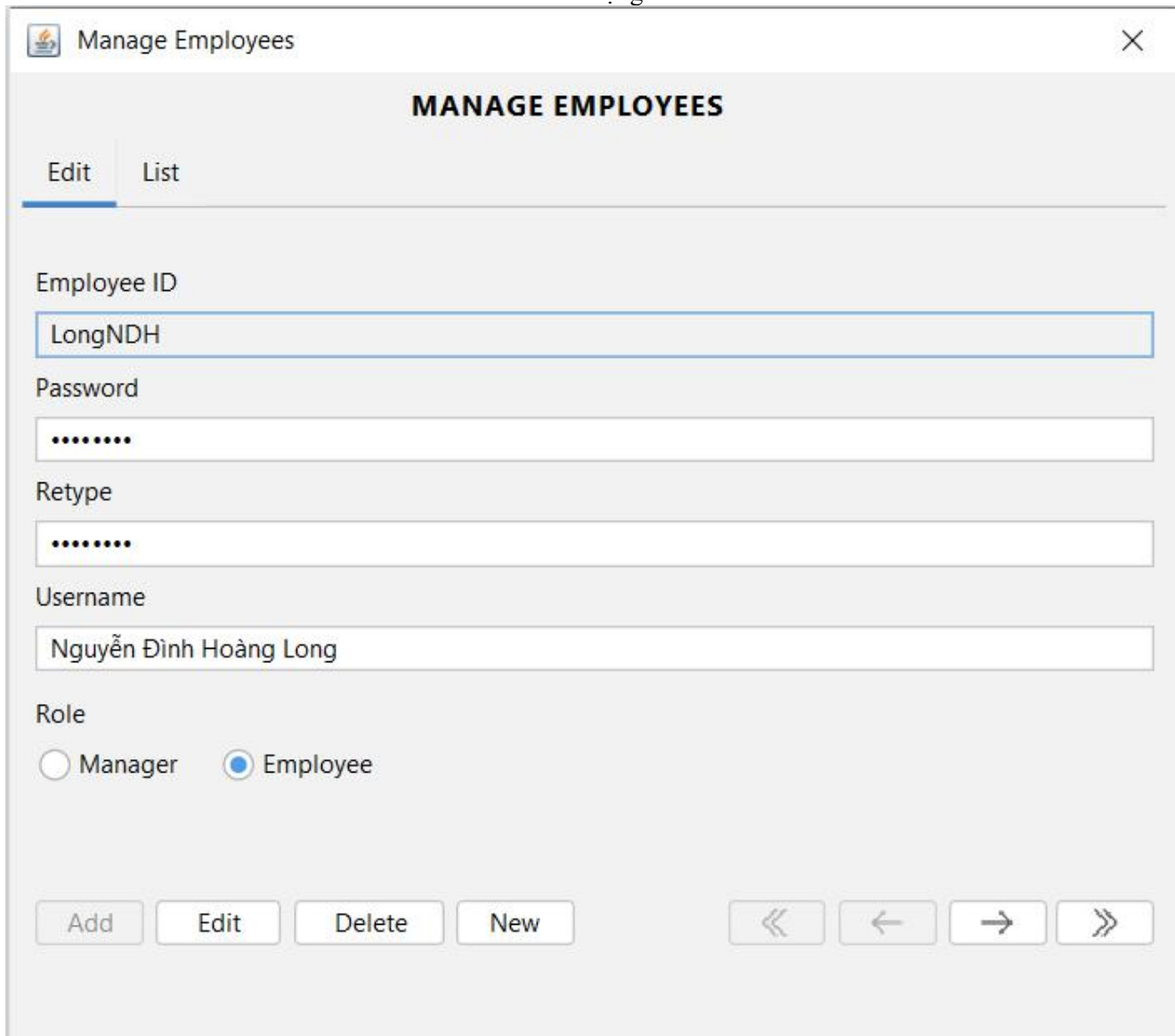
Role

☐ Manager
 ☐ Employee

Add
Edit
Delete
New


«
←
→
»

Hình: Form ở trạng thái insert



The screenshot shows a web application window titled "Manage Employees" with a close button (X) in the top right corner. The main heading is "MANAGE EMPLOYEES". Below the heading are two tabs: "Edit" (selected) and "List". The form contains several input fields: "Employee ID" with the value "LongNDH", "Password" with masked characters ".....", "Retype" with masked characters ".....", and "Username" with the value "Nguyễn Đình Hoàng Long". There is a "Role" section with two radio buttons: "Manager" (unselected) and "Employee" (selected). At the bottom, there are four buttons: "Add", "Edit", "Delete", and "New". To the right of these buttons are four navigation buttons: "<<", "<", ">", and ">>".

Hình: Form ở trạng thái update


Manage Employees
×

MANAGE EMPLOYEES			
Edit		List	
Emp ID	Password	Username	Role
LongNDH	*****	Nguyễn Đình Hoàng Lo...	Employee
LongNDT	*****	Nguyễn Đình Thiên Long	Employee
NghiemN	*****	Nguyễn Nghiệm	Manager
nhanvien	*****	nhanvien1	Employee
NoPT	*****	Phạm Thị Nở	Employee
PheoNC	*****	Nguyễn Chí Phèo	Employee
TeoNV	*****	Nguyễn Văn Tèo	Manager
ThaoLTH	*****	Lê Thị Hương Thảo	Employee

Hình: Bảng chứa danh sách nhân viên

MÔ TẢ HOẠT ĐỘNG

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

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

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

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

```

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

/**
 * Creates new form employeeJDialog
 */
public EmployeeJDialog(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    this.fillTable();
    this.updateStatus();
    setTitle("Manage Employees");
    setLocationRelativeTo(parent);
}

void insert() {
    NhanVien nv = getForm();
    String mk2 = new String(txtRetype.getPassword());
    if (!mk2.equals(nv.getMatKhau())) {
        MsgBox.alert(this, "Retype password didn't match!");
    } else {
        if (!Auth.isManager() && rdoManager.isSelected()) {
            MsgBox.alert(this, "Employee can't add new Manager!");
        } else {
            try {
                dao.insert(nv);
                this.fillTable();
            } catch (Exception e) {
                MsgBox.alert(this, "Error: " + e.getMessage());
            }
        }
    }
}

```

```

        this.clearForm();
        MsgBox.alert(this, "Insert successfully");
    } catch (Exception e) {
        MsgBox.alert(this, "Insert unsuccessfully!");
    }
}

}

}

void update() {
    NhanVien nv = getForm();
    String mk2 = new String(txtRetype.getPassword());
    if (!mk2.equals(nv.getMatKhau())) {
        MsgBox.alert(this, "Retype password didn't match!");
    } else {
        try {
            dao.update(nv);
            this.fillTable();
            MsgBox.alert(this, "Update successfully");
        } catch (Exception e) {
            MsgBox.alert(this, "Update unsuccessfully!");
        }
    }
}

void delete() {
    if (!Auth.isManager()) {
        MsgBox.alert(this, "You're not authorized to delete employee!");
    } else {
        String manv = txtEmpID.getText();
        if (manv.equals(Auth.user.getMaNV())) {
            MsgBox.alert(this, "You can't delete yourself!");
        } else if (MsgBox.confirm(this, "Do you want to delete this
employee?")) {
            try {
                dao.delete(manv);
                this.fillTable();
                this.clearForm();
                MsgBox.alert(this, "Delete successfully!");
            } catch (Exception e) {
                MsgBox.alert(this, "Delete unsuccessfully!");
            }
        }
    }
}

void edit() {
    String manv = (String) tblEmployee.getValueAt(this.row, 0);
    NhanVien nv = dao.selectById(manv);
    this.setForm(nv);
    tabs.setSelectedIndex(0);
    this.updateStatus();
}

```

```

}

void clearForm() {
    NhanVien nv = new NhanVien();
    this.setForm(nv);
    this.row = -1;
    this.updateStatus();
}

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

void prev() {
    if (this.row > 0) {
        this.row--;
        this.edit();
    }
}

void next() {
    if (this.row < tblEmployee.getRowCount() - 1) {
        this.row++;
        this.edit();
    }
}

void last() {
    this.row = tblEmployee.getRowCount() - 1;
    this.edit();
}

void fillTable() {
    DefaultTableModel model = (DefaultTableModel) tblEmployee.getModel();
    model.setRowCount(0);
    try {
        List<NhanVien> list = dao.selectAll();
        for (NhanVien nv : list) {
            Object[] rows = {nv.getMaNV(), nv.getMatKhau().replaceAll(".",
"*"), nv.getHoTen(), nv.isVaiTro() ? "Manager" : "Employee"};
            model.addRow(rows);
        }
    } catch (Exception e) {
        MsgBox.alert(this, "Data Query Error!");
    }
}

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


```

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

private NhanVien getForm() {
    NhanVien nv = new NhanVien();
    nv.setMaNV(txtEmpID.getText());
    nv.setHoTen(txtUsername.getText());
    nv.setMatKhau(new String(txtPassword.getPassword()));
    nv.setVaiTro(rdoManager.isSelected());
    return nv;
}

void updateStatus() {
    boolean edit = (this.row >= 0);
    boolean first = (this.row == 0);
    boolean last = (this.row == tblEmployee.getRowCount() - 1);
    //Form state
    txtEmpID.setEditable(!edit);
    btnAdd.setEnabled(!edit);
    btnEdit.setEnabled(edit);
    btnDelete.setEnabled(edit);
    //Directional state
    btnFirst.setEnabled(edit && !first);
    btnPrev.setEnabled(edit && !first);
    btnNext.setEnabled(edit && !last);
    btnLast.setEnabled(edit && !last);
}
```


3.3.2.5 Quản lý người học (NguoiHocJFrame)


Learners
×

MANAGE LEARNERS

Edit


List

Learner ID

Telephone

Name

Birthday



Gender

Male

▼

Email

Description

Add

Edit

Delete

New


«

←

→

»

Hình: form trạng thái thêm mới


Learners
×


MANAGE LEARNERS

Edit
List

Learner ID

Telephone

Name

Birthday


Gender

Email

Description

Add
Edit
Delete
New

<<
<
>
>>

Hình: form trạng thái cập nhật

Learners

×

MANAGE LEARNERS

Edit

List

Search

Search

ID	Name	Gender	Birthday	Phone	Email	EmpID	Creation Day
PS01638	LỮ HUY CƯ...	Female	08-05-1991	0928768265	PS01638@fp... TeoNV		Sun Oct 17 0...
PS02037	ĐỖ VĂN MI...	Male	24-10-1992	0968095685	PS02037@fp... PheoNC		Sun Oct 17 0...
PS02771	NGUYỄN TẮ...	Male	15-09-1998	0927594734	PS02771@fp... PheoNC		Sun Oct 17 0...
PS02867	NGUYỄN H...	Male	27-10-1997	0946984711	PS02867@fp... TeoNV		Sun Oct 17 0...
PS02930	TRẦN VĂN ...	Male	03-06-2000	0924774498	PS02930@fp... TeoNV		Sun Oct 17 0...
PS02979	ĐOÀN TRẦN...	Male	28-08-1994	0912374818	PS02979@fp... TeoNV		Sun Oct 17 0...
PS02983	NGUYỄN H...	Male	04-04-1993	0912499836	PS02983@fp... PheoNC		Sun Oct 17 0...
PS02988	HỒ HỮU H...	Male	08-02-1993	0924984876	PS02988@fp... PheoNC		Sun Oct 17 0...
PS03031	PHAN TẤN ...	Male	05-04-1990	0924832716	PS03031@fp... PheoNC		Sun Oct 17 0...
PS03046	NGUYỄN CA...	Male	28-01-1990	0977117727	PS03046@fp... PheoNC		Sun Oct 17 0...
PS03080	HUỲNH TH...	Male	06-09-1994	0916436052	PS03080@fp... PheoNC		Sun Oct 17 0...
PS03088	NGUYỄN H...	Male	02-09-1991	0938101529	PS03088@fp... PheoNC		Sun Oct 17 0...
PS03096	ĐOÀN HỮU...	Male	21-02-1994	0945196719	PS03096@fp... PheoNC		Sun Oct 17 0...
PS03104	LÊ THÀNH P...	Female	21-02-1993	0922948096	PS03104@fp... TeoNV		Sun Oct 17 0...

Hình: Bảng chứa danh sách người học

MÔ TẢ HOẠT ĐỘNG

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

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

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

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

```

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

/**
 * Creates new form LearnerJDialog
 */
public LearnerJDialog(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    this.fillTable();
    this.updateStatus();
    setTitle("Learners");
    setLocationRelativeTo(parent);
}

void insert() {
    NguoiHoc nh = getForm();
    try {
        dao.insert(nh);
        this.fillTable();
        this.clearForm();
        MsgBox.alert(this, "Insert successfully");
    } catch (Exception e) {
        MsgBox.alert(this, "Insert unsuccessfully!");
    }
}

```

```

    }

    void update() {
        NguoiHoc nh = getForm();
        try {
            dao.update(nh);
            this.fillTable();
            MsgBox.alert(this, "Update successfully");
        } catch (Exception e) {
            MsgBox.alert(this, "Update unsuccessfully!");
        }
    }

    void delete() {
        if (!Auth.isManager()) {
            MsgBox.alert(this, "You're not authorized to delete learner!");
        } else {
            String manh = txtLearnerID.getText();
            if (MsgBox.confirm(this, "Do you want to delete this employee?")) {
                try {
                    dao.delete(manh);
                    this.fillTable();
                    this.clearForm();
                    MsgBox.alert(this, "Delete sucessfully!");
                } catch (Exception e) {
                    MsgBox.alert(this, "Delete unsuccessfully!");
                }
            }
        }
    }

    void fillTable() {
        DefaultTableModel model = (DefaultTableModel) tblLearner.getModel();
        model.setRowCount(0);
        try {
            String keyword = txtSearch.getText();
            List<NguoiHoc> list = dao.selectByKeyword(keyword);
            for (NguoiHoc nh : list) {
                Object[] rows = {nh.getMaNH(), nh.getHoTen(), nh.isGioiTinh() ?
"Male" : "Female", XDate.toString(nh.getNgaySinh(), "dd-MM-yyyy"),
                nh.getDienThoai(), nh.getEmail(), nh.getMaNV(),
nh.getNgayDK()};
                model.addRow(rows);
            }
        } catch (Exception e) {
        }
    }

    void setForm(NguoiHoc nh) {
        txtLearnerID.setText(nh.getMaNH());
        txtName.setText(nh.getHoTen());
        txtBirth.setDate(nh.getNgaySinh());
    }

```

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

}

void clearForm() {
    NguoiHoc nh = new NguoiHoc();
    nh.setMaNV(Auth.user.getMaNV());
    nh.setNgayDK(new Date());
    this.setForm(nh);
    this.row = -1;
    this.updateStatus();
}

NguoiHoc getForm() {
    NguoiHoc nh = new NguoiHoc();
    nh.setMaNH(txtLearnerID.getText());
    nh.setHoTen(txtName.getText());
    nh.setDienThoai(txtTelephone.getText());
    nh.setGhiChu(txtDesc.getText());
    nh.setNgayDK(new Date());
    nh.setNgaySinh(txtBirth.getDate());
    nh.setEmail(txtEmail.getText());
    nh.setGioiTinh(cboGender.getSelectedIndex() == 0);
    nh.setMaNV(Auth.user.getMaNV());
    return nh;
}

void edit() {
    String manh = (String) tblLearner.getValueAt(this.row, 0);
    NguoiHoc nh = dao.selectById(manh);
    this.setForm(nh);
    tabs.setSelectedIndex(0);
    this.updateStatus();
}

void updateStatus() {
    boolean edit = (this.row >= 0);
    boolean first = (this.row == 0);
    boolean last = (this.row == tblLearner.getRowCount() - 1);
    //Form state
    txtLearnerID.setEditable(!edit);
    btnAdd.setEnabled(!edit);
    btnEdit.setEnabled(edit);
    btnDelete.setEnabled(edit);
    //Directional state
    btnFirst.setEnabled(edit && !first);
    btnPrev.setEnabled(edit && !first);
    btnNext.setEnabled(edit && !last);
}
```

```
        btnLast.setEnabled(edit && !last);
    }

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


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

    void prev() {
        if (this.row > 0) {
            this.row--;
            this.edit();
        }
    }

    void next() {
        if (this.row < tblLearner.getRowCount() - 1) {
            this.row++;
            this.edit();
        }
    }

    void last() {
        this.row = tblLearner.getRowCount() - 1;
        this.edit();
    }
}
```


3.3.2.6 Quản lý chuyên đề (ChuyenDeJFrame)


Manage Subjects

MANAGE SUBJECTS

Edit

List

Picture

Subject ID

Name

Duration (Hour)

Fees

Description

Add

Edit

Delete

New


<<

<

>

>>


Hình: trạng thái insert


Manage Subjects
×

MANAGE SUBJECTS

Edit
List

Picture



Subject ID

JAV03

Name

Lập trình mạng với Java

Duration (Hour)

70

Fees

200.0

Description

JAV03 - Lập trình mạng với Java

Add

Edit

Delete

New


«

←

→

»

Hình: trạng thái update


Manage Subjects
×

MANAGE SUBJECTS

Edit
List

Subject ID	Name	Fees	Duration	Picture
JAV01	Lập trình Java cơ ...	250.0	90	GAME.png
JAV02	Lập trình Java nân...	300.0	90	HTCS.jpg
JAV03	Lập trình mạng vớ...	200.0	70	INMA.jpg
JAV04	Lập trình desktop ...	200.0	70	ADAV.jpg
PRO01	Dự án với công n...	300.0	90	MOWE.png
PRO02	Dự án với công n...	300.0	90	Subject.png
PRO03	Dự án với công n...	300.0	90	GAME.png
PRO04	Dự án với Angular...	300.0	90	HTCS.jpg
PRO05	Dự án với Swing ...	300.0	90	INMA.jpg
PRO06	Dự án với Windo...	300.0	90	LAYO.jpg
RDB01	Cơ sở dữ liệu SQL ...	100.0	50	MOWE.png
RDB02	Lập trình JDBC	150.0	60	Subject.png
RDB03	Lập trình cơ sở dữ...	250.0	80	GAME.png
SER01	Lập trình web với ...	350.0	100	HTCS.jpg
SER02	Lập trình Spring M...	400.0	110	INMA.jpg
SER03	Lập trình MS.NET ...	400.0	110	LAYO.jpg
SER04	Xây dựng Web AP...	200.0	70	MOWE.png
WEB01	Thiết kế web với ...	200.0	70	Subject.png
WEB02	Thiết kế web với B...	0.0	40	GAME.png
WEB03	Lập trình front-en...	150.0	60	HTCS.jpg

Hình: bảng dữ liệu

MÔ TẢ HOẠT ĐỘNG

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

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

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

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

```

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

/**
 * Creates new form ManageSubject
 */
public SubjectJDialog(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    this.fillTable();
    this.updateStatus();
    setTitle("Manage Subjects");
    setLocationRelativeTo(parent);
}

void insert() {
    ChuyenDe cd = getForm();
    try {
        dao.insert(cd);
        this.fillTable();
    }
}

```

```

        this.clearForm();
        MsgBox.alert(this, "Insert Successfully!");
    } catch (Exception e) {
        MsgBox.alert(this, "Insert Unsuccessfully!");
    }
}

void update() {
    ChuyenDe cd = getForm();
    try {
        dao.update(cd);
        this.fillTable();
        MsgBox.alert(this, "Update successfully");
    } catch (Exception e) {
        MsgBox.alert(this, "Update unsuccessfully!");
    }
}

void delete() {
    if (!Auth.isManager()) {
        MsgBox.alert(this, "You're not authorized to delete employee!");
    } else {
        String id = txtSubjectID.getText();
        if (MsgBox.confirm(this, "Do you want to delete this subject?")) {
            try {
                dao.delete(id);
                this.fillTable();
                this.clearForm();
                MsgBox.alert(this, "Delete sucessfully!");
            } catch (Exception e) {
                MsgBox.alert(this, "Delete unsuccessfully!");
            }
        }
    }
}

void choosePicture() {
    if (fileChooser.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
        fileChooser.setDialogTitle("Choose Image");
        File file = fileChooser.getSelectedFile();
        XImage.save(file);
        ImageIcon icon = XImage.read(file.getName());
        lblPicture.setIcon(icon);
        lblPicture.setToolTipText(file.getName());
    }
}

void fillTable() {
    DefaultTableModel model = (DefaultTableModel) tblSubject.getModel();
    model.setRowCount(0);
    try {
        List<ChuyenDe> list = dao.selectAll();

```

```

        for (ChuyenDe cd : list) {
            Object[] rows = {cd.getMaCD(), cd.getTenCD(), cd.getHocPhi(),
cd.getThoiLuong(), cd.getHinh()};
            model.addRow(rows);
        }
    } catch (Exception e) {
        MsgBox.alert(this, "Data query error!");
    }
}

void edit() {
    String macd = (String) tblSubject.getValueAt(this.row, 0);
    ChuyenDe cd = dao.selectById(macd);
    this.setForm(cd);
    tabs.setSelectedIndex(0);
    this.updateStatus();
}

void updateStatus() {
    boolean edit = (this.row >= 0);
    boolean first = (this.row == 0);
    boolean last = (this.row == tblSubject.getRowCount() - 1);
    //form state
    txtSubjectID.setEditable(!edit);
    btnAdd.setEnabled(!edit);
    btnEdit.setEnabled(edit);
    btnDelete.setEnabled(edit);
    //Directional state
    btnPrev.setEnabled(edit && !first);
    btnNext.setEnabled(edit && !last);
    btnFirst.setEnabled(edit && !first);
    btnLast.setEnabled(edit && !last);
}

void setForm(ChuyenDe cd) {
    txtSubjectID.setText(cd.getMaCD());
    txtName.setText(cd.getTenCD());
    txtFees.setText(String.valueOf(cd.getHocPhi()));
    txtDuration.setText(String.valueOf(cd.getThoiLuong()));
    txtDesc.setText(cd.getMoTa());
    if (cd.getHinh() != null) {
        lblPicture.setToolTipText(cd.getHinh());
        lblPicture.setIcon(XImage.read(cd.getHinh()));
    }
}

private ChuyenDe getForm() {
    ChuyenDe cd = new ChuyenDe();
    cd.setMaCD(txtSubjectID.getText());
    cd.setTenCD(txtName.getText());
    cd.setHocPhi(Double.valueOf(txtFees.getText()));

```

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

    void clearForm() {
        ChuyenDe cd = new ChuyenDe();
        this.setForm(cd);
        this.row = -1;
        this.updateStatus();
    }


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

    void prev() {
        if (row > 0) {
            this.row--;
            this.edit();
        }
    }

    void next() {
        if (row < tblSubject.getRowCount() - 1) {
            this.row++;
            this.edit();
        }
    }

    void last() {
        this.row = tblSubject.getRowCount() - 1;
        this.edit();
    }
}
```


3.3.2.7 Quản lý khóa học(KhoaHocJFrame)

 Courses ×

MANAGE COURSES

SUBJECTS


Lập trình Java cơ bản ▼

Edit

List

Course ID	Duration	Fees	FDOS	Creator	Day Of Creation
1005	90	300.0	12-04-2020	TeoNV	02-05-2020
1013	90	300.0	10-04-2020	TeoNV	02-05-2020
1014	90	250.0	08-04-2020	TeoNV	02-05-2020
1019	90	300.0	12-01-1970	NghiemN	11-10-2021

Hình: trạng thái insert

 Courses
×

MANAGE COURSES

SUBJECTS

Lập trình Java cơ bản

Edit
List

Subject

Lập trình Java cơ bản

Fees

250.0

Creator

TeoNV

Description

Lập trình Java cơ bản

First day of school

08/04/2020

Duration (Hour)

90

Day of creation

02-05-2020

Add
Edit
Delete
New

<<
←
→
>>

Hình: Trạng thái update

Courses
 ×

MANAGE COURSES

SUBJECTS

Lập trình Java cơ bản

Edit
 List

Course ID	Duration	Fees	FDOS	Creator	Day Of Creation
1005	90	300.0	12-04-2020	TeoNV	02-05-2020
1013	90	300.0	10-04-2020	TeoNV	02-05-2020
1014	90	250.0	08-04-2020	TeoNV	02-05-2020
1019	90	300.0	12-01-1970	NghiemN	11-10-2021

Hình: bảng dữ liệu

MÔ TẢ HOẠT ĐỘNG

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

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

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

```

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

/**
 * Creates new form courseJDialog
 */
public CourseJDialog(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    this.fillComboBoxChuyenDe();
    this.fillTable();
    this.updateStatus();
    setTitle("Courses");
    setLocationRelativeTo(parent);
}

```

```

}

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

void fillTable() {
    DefaultTableModel model = (DefaultTableModel) tblCourse.getModel();
    model.setRowCount(0);
    try {
        ChuyenDe chuyenDe = (ChuyenDe) cboSubjects.getSelectedItem();
        List<KhoaHoc> list = khDAO.selectByChuyenDe(chuyenDe.getMaCD());
        for (KhoaHoc kh : list) {
            Object[] row = {kh.getMaKH(), kh.getThoiLuong(), kh.getHocPhi(),
                XDate.toString(kh.getNgayKG(), "dd-MM-yyyy"), kh.getMaNV(),
                XDate.toString(kh.getNgayTao(), "dd-MM-yyyy")};
            model.addRow(row);
        }
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}

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

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

void insert() {
    KhoaHoc kh = getForm();
    try {
        khDAO.insert(kh);
        this.fillTable();
        MsgBox.alert(this, "Insert Successfully!");
    } catch (Exception e) {
        MsgBox.alert(this, "Insert Unsuccessfully!");
    }
}

```

```
}

void update() {
    KhoaHoc kh = getForm();
    try {
        khDAO.update(kh);
        this.fillTable();
        MsgBox.alert(this, "Update successfully");
    } catch (Exception e) {
        MsgBox.alert(this, "Update unsuccessfully!");
    }
}

void delete() {
    if (!Auth.isManager()) {
        MsgBox.alert(this, "You're not authorized to delete employee!");
    } else {
        int id = Integer.parseInt(cboSubjects.getToolTipText());
        if (MsgBox.confirm(this, "Do you want to delete this subject?")) {
            try {
                khDAO.delete(id);
                this.fillTable();
                MsgBox.alert(this, "Delete successfully!");
            } catch (Exception e) {
                MsgBox.alert(this, "Delete unsuccessfully!");
            }
        }
    }
}

void updateStatus() {
    boolean edit = (this.row >= 0);
    boolean first = (this.row == 0);
    boolean last = (this.row == tblCourse.getRowCount() - 1);
    //Form state
    btnAdd.setEnabled(!edit);
    btnEdit.setEnabled(edit);
    btnDel.setEnabled(edit);
    //Directional state
    btnFirst.setEnabled(edit && !first);
    btnPrev.setEnabled(edit && !first);
    btnNext.setEnabled(edit && !last);
    btnLast.setEnabled(edit && !last);
}

void edit() {
    int makh = (int) tblCourse.getValueAt(this.row, 0);
    KhoaHoc kh = khDAO.selectById(makh);
    this.setForm(kh);
    tabs.setSelectedIndex(0);
    this.updateStatus();
}
```

```

}

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

void setForm(KhoaHoc kh) {
    cboSubjects.setSelectedItem(cdDAO.selectById(kh.getMaCD()));
    cboSubjects.setToolTipText(String.valueOf(kh.getMaKH()));
    txtFDOS.setDate(kh.getNgayKG());
    txtCreator.setText(kh.getMaNV());
    txtDesc.setText(kh.getGhiChu());
    txtFees.setText(String.valueOf(kh.getHocPhi()));
    txtDuration.setText(String.valueOf(kh.getThoiLuong()));
    txtDayCreated.setText(XDate.toString(kh.getNgayTao(), "dd-MM-yyyy"));
}

KhoaHoc getForm() {
    KhoaHoc kh = new KhoaHoc();
    ChuyenDe cd = (ChuyenDe) cboSubjects.getSelectedItem();
    if (txtCreator.getText().equals("")) {
        kh.setMaNV(Auth.user.getMaNV());
    } else {
        kh.setMaNV(txtCreator.getText());
    }
    kh.setMaCD(cd.getMaCD());
    kh.setThoiLuong(Integer.parseInt(txtDuration.getText()));
    kh.setNgayKG(txtFDOS.getDate());
    kh.setHocPhi(Double.parseDouble(txtFees.getText()));
    kh.setGhiChu(txtDesc.getText());
    kh.setMaKH(Integer.parseInt(cboSubjects.getToolTipText()));
    return kh;
}

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

void prev() {
    if (this.row > 0) {
        this.row--;
        this.edit();
    }
}

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

```

```


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

void last() {
    this.row = tblCourse.getRowCount() - 1;
    this.edit();
}

```

3.3.2.8 Quản lý học viên (HocVienJFrame)

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


Students

×

SUBJECTS

Lập trình Java cơ bản ▼

COURSES

JAV01 (2020-04-12) ▼

Students

Learners

Order	Student ID	Learner ID	Name	Score
1	1052	PS01638	LỮ HUY CƯỜNG	8
2	1053	PS02037	ĐỖ VĂN MINH	8
3	1054	PS02771	NGUYỄN TẤN HIẾU	9
4	1055	PS02867	NGUYỄN HỮU TRÍ	7
5	1056	PS02930	TRẦN VĂN NAM	5
6	2049	PS03488	NGUYỄN NHƯ NGỌC	0
7	2050	PS03553	ĐINH TẤN CÔNG	0
8	2051	PS03596	NGUYỄN THANH HIỀN	0
9	2052	PS03104	LÊ THÀNH PHƯƠNG	0
10	2053	PS03411	TRƯƠNG THÀNH ĐẠT	0
11	2054	PS03530	PHẠM THÀNH TÂM	0

Remove from course

Update score

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

cboChuaNhap	
btnCapNhat	Update()

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

```

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

/**
 * Creates new form StudentJDialog
 */
public StudentJDialog(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    this.fillComboBoxChuyenDe();
    setTitle("Students");
    setLocationRelativeTo(parent);
}

void fillComboBoxChuyenDe() {
    DefaultComboBoxModel model = (DefaultComboBoxModel)
cboSubjects.getModel();
    model.removeAllElements();
    List<ChuyenDe> list = cdDAO.selectAll();
    for (ChuyenDe cd : list) {
        //add name of the subject to cbo
        model.addElement(cd);
    }
    this.fillComboBoxKhoaHoc();
}

void fillComboBoxKhoaHoc() {
    DefaultComboBoxModel model = (DefaultComboBoxModel) cboCourses.getModel();
    model.removeAllElements();
    ChuyenDe cd = (ChuyenDe) cboSubjects.getSelectedItem();
    if (cd != null) {
        List<KhoaHoc> list = khDAO.selectByChuyenDe(cd.getMaCD());
        for (KhoaHoc kh : list) {
            model.addElement(kh);
        }
        this.fillTableHocVien();
    }
}

void fillTableHocVien() {
    DefaultTableModel model = (DefaultTableModel) tblStudent.getModel();

```

```

        model.setRowCount(0);
        KhoaHoc kh = (KhoaHoc) cboCourses.getSelectedItem();
        if (kh != null) {
            List<HocVien> list = hvDAO.selectByKhoaHoc(kh.getMaKH());
            for (int i = 0; i < list.size(); i++) {
                HocVien hv = list.get(i);
                String hoten = nhDAO.selectById(hv.getMaNH()).getHoTen();
                model.addRow(new Object[]{i + 1, hv.getMaHV(), hv.getMaNH(),
hoten, hv.getDien()});
            }
            this.fillTableNguoiHoc();
        }
    }

    void fillTableNguoiHoc() {
        DefaultTableModel model = (DefaultTableModel) tblLearner.getModel();
        model.setRowCount(0);
        KhoaHoc kh = (KhoaHoc) cboCourses.getSelectedItem();
        String keyword = txtSearch.getText();
        List<NguoiHoc> list = nhDAO.selectNotInCourse(kh.getMaKH(), keyword);
        for (NguoiHoc nh : list) {
            model.addRow(new Object[]{nh.getMaNH(), nh.getHoTen(),
nh.isGioiTinh() ? "Male" : "Female",
                XDate.toString(nh.getNgaySinh(), "dd-MM-yyyy"), nh.getDienThoai(),
nh.getEmail()});
        }
    }

    void addHocVien() {
        KhoaHoc kh = (KhoaHoc) cboCourses.getSelectedItem();
        int[] rows = tblLearner.getSelectedRows();
        for (int row : rows) {
            String manh = (String) tblLearner.getValueAt(row, 0);
            HocVien hv = new HocVien();
            hv.setMaKH(kh.getMaKH());
            hv.setDien(0);
            hv.setMaNH(manh);
            hvDAO.insert(hv);
        }
        this.fillTableHocVien();
        this.fillTableNguoiHoc();
    }

    void removeHocVien() {
        if (!Auth.isManager()) {
            MsgBox.alert(this, "You're not authorized to delete employee!");
        } else {
            int[] rows = tblStudent.getSelectedRows();
            if (rows.length > 0 && MsgBox.confirm(this, "Do you want to delete
student?")) {

```

```
        for (int row : rows) {
            int mahv = (Integer) tblStudent.getValueAt(row, 1);
            hvDAO.delete(mahv);
        }
        this.fillTableHocVien();
        this.fillTableNguoiHoc();
    }
}

void updateDiem() {
    int n = tblStudent.getRowCount();
    for (int i = 0; i < n; i++) {
        int mahv = (Integer) tblStudent.getValueAt(i, 1);
        double diem = (Double) tblStudent.getValueAt(i, 4);
        HocVien hv = hvDAO.selectById(mahv);
        hv.setDiem(diem);
        hvDAO.update(hv);
    }
    this.fillTableHocVien();
    MsgBox.alert(this, "Update score successfully!");
}
```

3.3.2.9 Thống kê dữ liệu (ThongKeJFrame)

STATISTIC			
Score board	Learners	Subject Score	Revenue
Year	Learners Number	Earliest Signup	Latest Signup
1990	8	1990-01-02	1990-04-27
1991	3	1991-05-08	1991-09-02
1992	5	1992-02-15	1992-10-24
1993	6	1993-02-08	1993-12-01
1994	4	1994-02-21	1994-09-06
1995	5	1995-01-05	1995-05-09
1996	6	1996-02-04	1996-12-11
1997	3	1997-02-09	1997-10-27
1998	2	1998-04-08	1998-09-15
1999	3	1999-02-24	1999-07-09
2000	5	2000-03-10	2000-10-14

Hình: Thống kê số đăng ký từng năm

Statistics

STATISTIC

Score board

Learners

Subject Score

Revenue

Course

PRO02 (2018-01-10)

Learner ID	Name	Score	Classify
PS02037	ĐỖ VĂN MINH	9.0	Excellent
PS01638	LỮ HUY CƯỜNG	8.0	Good
PS02771	NGUYỄN TẤN HIẾU	8.0	Good
PS02930	TRẦN VĂN NAM	7.0	Above average
PS02983	NGUYỄN HOÀNG THIÊN ...	6.0	Average
PS02979	ĐOÀN TRẦN NHẬT VŨ	4.0	Not Qualified
PS02867	NGUYỄN HỮU TRÍ	3.0	Not Qualified
PS02988	HỒ HỮU HẬU	0.0	Not Qualified
PS03031	PHAN TẤN VIỆT	0.0	Not Qualified
PS03046	NGUYỄN CAO PHƯỚC	0.0	Not Qualified
PS03080	HUỲNH THANH HUY	0.0	Not Qualified
PS03088	NGUYỄN HOÀNG TRUNG	0.0	Not Qualified
PS03096	ĐOÀN HỮU KHANG	0.0	Not Qualified
PS03104	LÊ THÀNH PHƯƠNG	0.0	Not Qualified
PS03120	PHẠM NGỌC NHẬT TRƯ...	0.0	Not Qualified
PS03130	ĐẶNG BẢO VIỆT	0.0	Not Qualified

Hình: Bảng điểm của từng khóa



Statistics

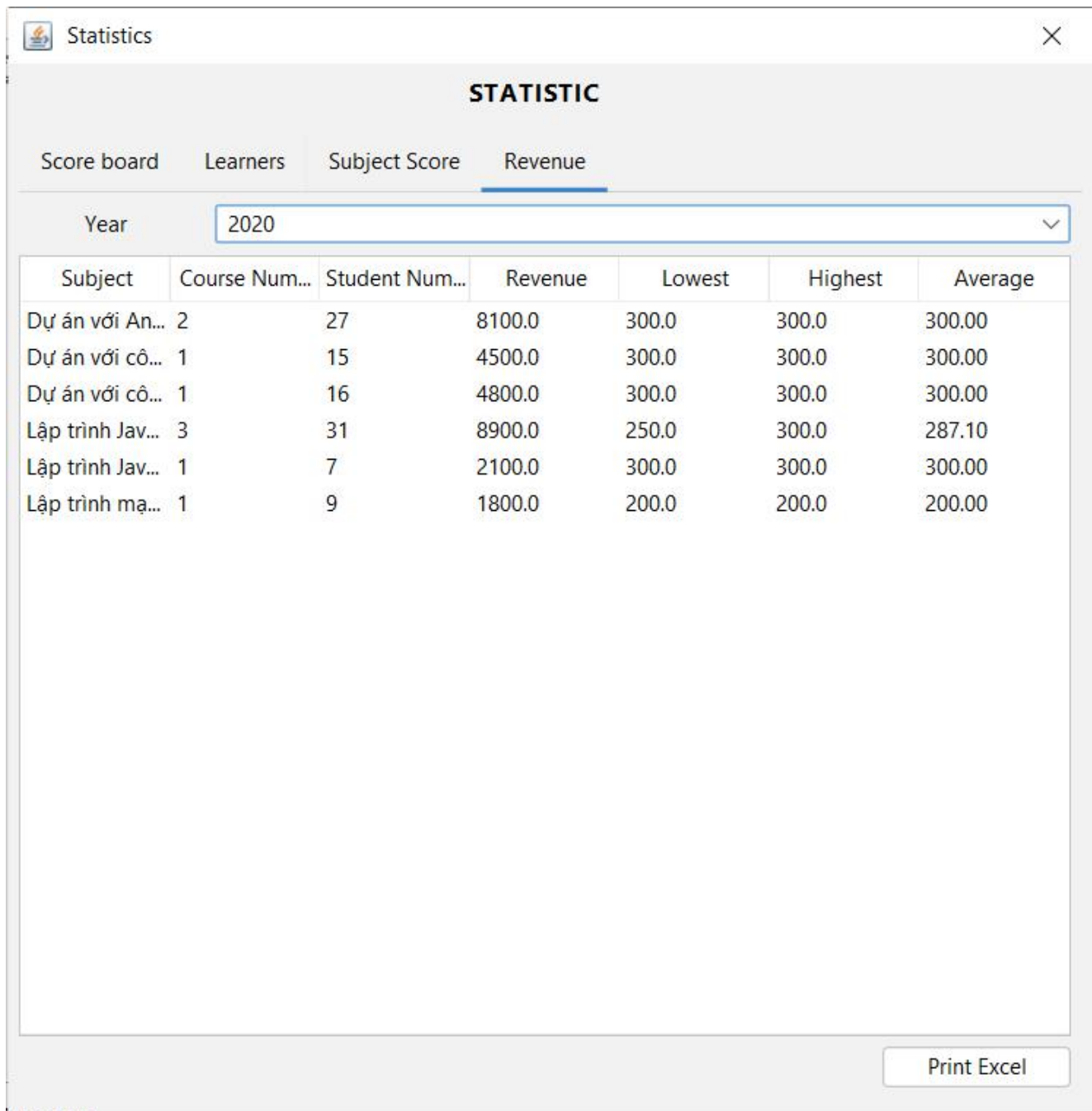


STATISTIC

Score board Learners **Subject Score** Revenue

Subject	Students Number	Lowest	Highest	Average
Dự án với AngularJS...	27	0.0	9.0	0.52
Dự án với công ngh...	15	0.0	0.0	0.00
Dự án với công ngh...	16	0.0	0.0	0.00
Dự án với công ngh...	16	0.0	9.0	2.81
Lập trình desktop v...	13	0.0	9.0	4.15
Lập trình Java cơ bả...	31	0.0	10.0	3.03
Lập trình Java nâng ...	7	0.0	9.0	2.57
Lập trình mạng với J...	9	0.0	0.0	0.00

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



Hình: Thống kê doanh thu từng chuyên đề theo năm

MÔ TẢ HOẠT ĐỘNG

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

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

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

```

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

/**
 * Creates new form StatisticJDialog
 */
public StatisticJDialog(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    init();
    setTitle("Statistics");
    setLocationRelativeTo(parent);
}

void init() {
    fillComboYear();
    fillComboKhoaHoc();
    fillTableBangDiem();
    fillTableDiemChuyenDe();
    fillTableNguoiHoc();
    fillTableDoanhThu();
    this.selectTab(0);
    if (!Auth.isManager()) {
        tabs.remove(3);
    }
}

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

void fillTableBangDiem() {
    DefaultTableModel model = (DefaultTableModel)
tblScoreBoard.getModel();
    model.setRowCount(0);

```

```

        KhoaHoc kh = (KhoaHoc) cboCourse.getSelectedItemAt();
        List<Object[]> list = tkDAO.getBangDiem(kh.getMaKH());
        for (Object[] row : list) {
            double score = (double) row[2];
            model.addRow(new Object[]{row[0], row[1], score,
getXepLoai(score)});
        }
    }

    void fillTableNguoiHoc() {
        DefaultTableModel model = (DefaultTableModel) tblLearners.getModel();
        model.setRowCount(0);
        List<Object[]> list = tkDAO.getLuongNguoiHoc();
        for (Object[] row : list) {
            model.addRow(row);
        }
    }

    void fillTableDiemChuyenDe() {
        DefaultTableModel model = (DefaultTableModel)
tblSubjectScore.getModel();
        model.setRowCount(0);
        List<Object[]> list = tkDAO.getDiemChuyenDe();
        for (Object[] row : list) {
            model.addRow(new Object[]{row[0], row[1], row[2], row[3], new
DecimalFormat("####0.00").format(row[4])});
        }
    }

    void fillComboYear() {
        DefaultComboBoxModel model = (DefaultComboBoxModel)
cboYear.getModel();
        model.removeAllElements();
        List<Integer> list = khDAO.selectYears();
        for (Integer year : list) {
            model.addElement(year);
        }
    }

    void fillTableDoanhThu() {
        DefaultTableModel model = (DefaultTableModel) tblRevenue.getModel();
        model.setRowCount(0);
        int year = (Integer) cboYear.getSelectedItemAt();
        List<Object[]> list = tkDAO.getDoanhThu(year);
        for (Object[] row : list) {
            model.addRow(new Object[]{row[0], row[1], row[2], row[3],
row[4], row[5], new
DecimalFormat("####0.00").format(row[6])});
        }
    }

    void fillComboKhoaHoc() {

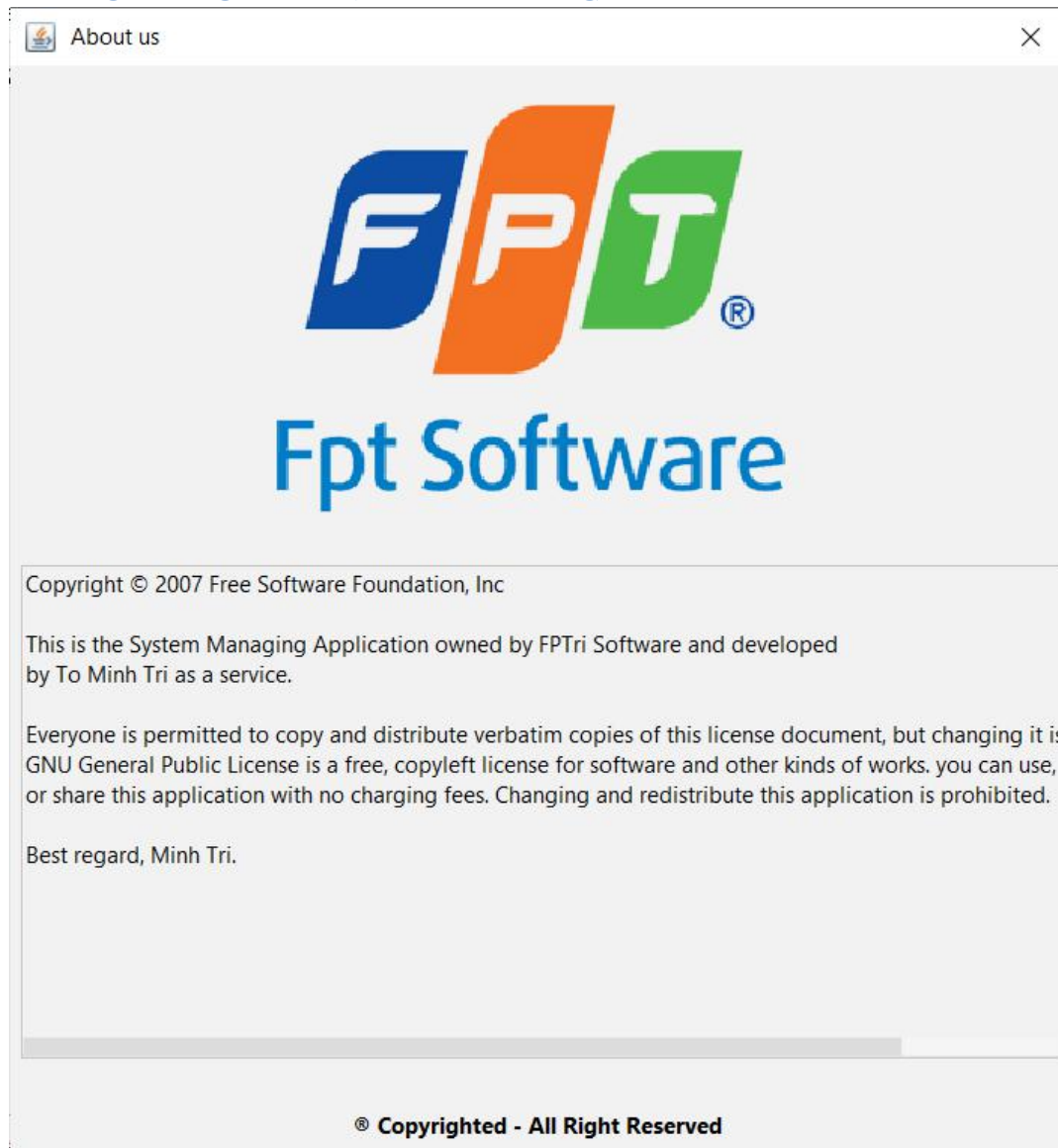
```

```
        DefaultComboBoxModel model = (DefaultComboBoxModel)
cboCourse.getModel();
        model.removeAllElements();
        List<KhoaHoc> list = khDAO.selectAll();
        for (KhoaHoc kh : list) {
            model.addElement(kh);
        }
    }

String getXepLoai(double score) {
    if (score < 5) {
        return "Not Qualified";
    } else if (score < 6.5) {
        return "Average";
    } else if (score < 7.5) {
        return "Above average";
    } else if (score < 9) {
        return "Good";
    }
    return "Excellent";
}

void printReport(JTable table, File file) {
    try {
        File f = new File(file + ".xlsx");
        TableModel model = table.getModel();
        FileWriter excel = new FileWriter(f);
        for (int i = 0; i < model.getColumnCount(); i++) {
            excel.write(model.getColumnName(i) + "\t");
        }
        excel.write("\n");
        for (int i = 0; i < model.getRowCount(); i++) {
            for (int j = 0; j < model.getColumnCount(); j++) {
                excel.write(model.getValueAt(i, j).toString() + "\t");
            }
            excel.write("\n");
        }
        if (MsgBox.confirm(this, "Print Successfully! Do you want to
open it?")) {
            Desktop.getDesktop().browse(f.toURI());
        }
        excel.close();
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
```

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);
lblLogo txtContent	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
NgàyKG	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 EduSys bằng cách chạy file EduSys.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: NghiemH và password: songlong
- ✓ 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