

**Lập trình CITY EDUCATION CENTER - EDUSYS**

**HỆ THỐNG QUẢN LÝ ĐÀO TẠO - EduSys**



|  |  |
| --- | --- |
| Giảng viên | : Phạm Tiến Huy |
| Học viên | : Trương Minh Tiến – PY00023 |
| Lớp/Kỳ | : SD19301/SM24 |

June 18, 2024

LẬP TRÌNH CITY EDUCATION CENTER

MỤC LỤC

[1 Giới thiệu dự án 4](#_Toc43201191)

[1.1 Giới thiệu công ty LapTrinhCity 4](#_Toc43201192)

[1.2 Yêu cầu của công ty 4](#_Toc43201193)

[1.3 Lập kế hoạch dự án 4](#_Toc43201194)

[2 Phân tích yêu cầu khách hàng 4](#_Toc43201195)

[2.1 Sơ đồ Use Case 4](#_Toc43201196)

[2.2 Đặc tả yêu cầu hệ thống (SRS) 4](#_Toc43201197)

[2.2.1 Quản lý nhân viên 5](#_Toc43201198)

[2.2.2 Quản lý chuyên đề 5](#_Toc43201199)

[2.2.3 Quản lý người học 5](#_Toc43201200)

[2.2.4 Quản lý khóa học 5](#_Toc43201201)

[2.2.5 Quản lý học viên 5](#_Toc43201202)

[2.2.6 Quản lý tổng hợp – thống kê 5](#_Toc43201203)

[2.2.7 Đăng nhập 5](#_Toc43201204)

[2.2.8 Đổi mật khẩu 5](#_Toc43201205)

[2.3 Sơ đồ triển khai và yêu cầu hệ thống 5](#_Toc43201206)

[2.3.1 Sơ đồ triển khai 6](#_Toc43201207)

[2.3.2 Yêu cầu hệ thống 6](#_Toc43201208)

[3 Thiết kế ứng dụng 6](#_Toc43201209)

[3.1 Mô hình công nghệ ứng dụng 6](#_Toc43201210)

[3.2 Thực thể 6](#_Toc43201211)

[3.2.1 Sơ đồ quan hệ thực thể (ERD) 6](#_Toc43201212)

[3.2.2 Chi tiết thực thể 7](#_Toc43201213)

[3.3 Giao diện 8](#_Toc43201214)

[3.3.1 Sơ đồ tổ chức giao diện 8](#_Toc43201215)

[3.3.2 Giao diện cửa sổ chính 8](#_Toc43201216)

[3.3.3 Giao diện chức năng 9](#_Toc43201217)

[3.3.4 Các giao diện hỗ trợ khác 11](#_Toc43201218)

[4 Thực hiện dự án 13](#_Toc43201219)

[4.1 Tạo giao diện với Swing 13](#_Toc43201220)

[4.1.1 Cửa sổ chính (EduSysJFrame) 13](#_Toc43201221)

[4.1.2 Các cửa sổ quản lý 15](#_Toc43201222)

[4.1.3 Cửa sổ tổng hợp thống kê (ThongKeJDialog) 16](#_Toc43201223)

[4.1.4 Các giao diện hỗ trợ khác 16](#_Toc43201224)

[4.2 Tạo CSDL với SQL Server 17](#_Toc43201225)

[4.2.1 Sơ đồ quan hệ 17](#_Toc43201226)

[4.2.2 Chi tiết các bảng 17](#_Toc43201227)

[4.2.3 Thủ tục lưu 18](#_Toc43201228)

[4.3 Lập trình CSDL 19](#_Toc43201229)

[4.3.1 Mô hình tổ chức lập trình CSDL 19](#_Toc43201230)

[4.3.2 JdbcHelper (XJdbc) 19](#_Toc43201231)

[4.3.3 Entity Class và DAO 20](#_Toc43201232)

[4.4 Thư viện tiện ích 23](#_Toc43201233)

[4.5 Lập trình nghiệp vụ 23](#_Toc43201234)

[4.5.1 Cửa sổ chính EduSysJFrame 23](#_Toc43201235)

[4.5.2 Các cửa sổ hỗ trợ tổ chức 23](#_Toc43201236)

[4.5.3 Các cửa sổ chức năng quản lý 24](#_Toc43201237)

[4.5.4 Cửa sổ chức năng tổng hợp - thống kê 26](#_Toc43201238)

[5 Kiểm thử phần mềm và sửa lỗi 26](#_Toc43201239)

[5.1 DangNhapJDialog 26](#_Toc43201240)

[5.2 DoiMatKhauJDialog 26](#_Toc43201241)

[5.3 NhanVienJDialog 26](#_Toc43201242)

[5.4 ChuyenDeJDialog 26](#_Toc43201243)

[5.5 NguoiHocJDialog 26](#_Toc43201244)

[5.6 KhoaHocJDialog 26](#_Toc43201245)

[5.7 HocVienJDialog 27](#_Toc43201246)

[5.8 ThongKeJDialog 27](#_Toc43201247)

[6 Đóng gói và triển khai 27](#_Toc43201248)

[6.1 Sản phẩm phần mềm 27](#_Toc43201249)

[6.2 Hướng dẫn cài đặt 27](#_Toc43201250)

[7 KẾT LUẬN 27](#_Toc43201251)

[7.1 Khó khăn 27](#_Toc43201252)

[7.2 Thuận lợi 27](#_Toc43201253)

# Giới thiệu dự án

## Giới thiệu công ty LapTrinhCity

* Trung tâm đào tạo tin học ngắn hạn có tên là LapTrinhCity đang đào tạo các khóa học ngắn hạn theo chuyên đề như Photoshop, Java, Web,…
* Việc quản lý khóa học, học viên, bảng điểm và doanh thu hiện đang thực hiện thông qua excel.
* Hiện tại trung tâm đang gặp khó khăn khi số lượng người học ngày một nhiều nên việc quản lý excel gặp rất nhiều khó khăn.
* LapTrinhCity mong muốn xây dựng một phần mềm để giải quyết khó khăn trên.

## Yêu cầu của công ty

* **Yêu cầu chức năng ngiệp 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
* Nhân viên phòng đào tạo không được phép xóa và không được xem thông tin về doanh thu
* **Yêu cầu về công nghệ:**
* Ứng dụng phải được thực với công nghệ Swing và JDBC
* JDK 1.8+
* Hệ quản trị CSDL SQL Server 2008 trở lên
* **Đánh giá tính khả thi:** với năng lực của bản thân, mình chỉ cần thực sự chú tâm bỏ công sức và thời gian vào dự án này thì nó sẽ có khả năng hoàn thiện khá là tốt trong một khoảng thời gian nhất định.

## Lập kế hoạch dự án

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Hạng mục** | **Bắt đầu** | **Kết thúc** | **Kết quả** |
| 1 | Phân tích yêu cầu khách hàng | 08/05/2024 | 19/05/2024 | Hoàn thành |
| 1.1 | Vẽ sơ đồ use cases | 09/05/2024 | 09/05/2024 | Hoàn thành |
| 1.2 | Đặc tả yêu cầu hệ thống (SRS) | 12/05/2024 | 12/05/2024 | Hoàn thành |
| 1.3 | Sơ đồ triển khai và yêu cầu hệ thống | 14/05/2024 | 14/05/2024 | Hoàn thành |
| 2 | Thiết kế ứng dụng | 15/05/2024 | 25/06/2024 | Hoàn thành |
| 2.1 | Thiết kế mô hình công nghệ | 15/05/2024 | 15/05/2024 | Hoàn thành |
| 2.2 | Thực thể | 15/05/2024 | 15/05/2024 | Hoàn thành |
| 2.3 | Thiết kế giao diện | 16/05/2024 | 20/05/2024 | Hoàn thành |
| 3 | Thực hiện dự án | 21/05/2024 | 24/05/2024 | Hoàn thành |
| 3.1 | Tạo giao diện | 24/05/2024 | 27/05/2024 | Hoàn thành |
| 3.2 | Tạo CSDL với SQL Server | 28/05/2024 | 30/05/2024 | Hoàn thành |
| 3.3 | Lập trình CSDL | 01/06/2024 | 02/06/2024 | Hoàn thành |
| 4 | Kiểm thử | 03/06/2024 | 07/06/2024 | Hoàn thành |
| 4.1 | Xây dựng kịch bảng kiểm thử | 16/6/2024 | 16/6/2024 | Hoàn thành |
| 5 | Đóng gói & triển khai | 16/6/2024 | 16/6/2024 | Hoàn thành |
| 5.1 | Đóng gói sản phẩm | 16/6/2024 | 16/6/2024 | Hoàn thành |

# Phân tích yêu cầu khách hàng

## Sơ đồ Use Case

* *Vẽ sơ đồ use case để phác họa phạm vị của ứng dụng cũng như vai trò sử dụng. Nên chia nhiều mức để tránh rối sơ đồ*

## Đặc tả yêu cầu hệ thống (SRS)

### Quản lý nhân viên

* **Mô tả chức năng:**

Chức năng quản lý nhân viên được sử dụng để quản lý thông tin nhân viên. Yêu cầu của chức năng này là liệt kê danh sách nhân viên, xem thông tin chi tiết của mỗi nhân viên, thêm nhân viên mới, cập nhật thông tin hoặc xóa nhân viên đã tồn tài.

* **Dữ liệu liên quan:**

Thông tin của mỗi nhân viên gồm: mã nhân viên, họ và tên, mật khẩu đăng nhập, vai trò của nhân viên (gồm trưởng phòng và nhân viên bình thường).

* **Đối tượng sử dụng:**

Sau khi đăng nhập thì nhân viên nào cũng có thể sử dụng chức năng này, ngoại trừ xóa thì chỉ có trưởng phòng mới sử dụng được.

### Quản lý chuyên đề

* **MÔ TẢ CHỨC NĂNG:**

Chức năng quản lý chuyên đề được sử dụng để quản lý chuyên đề đạo tạo ngắn hạn. Yêu cầu của chức năng này là liệt kê danh sách thông tin của các chuyên đề, xem chi tiết chuyên đề, thêm chuyên đề mới, cập nhật chuyên đề hoặc xóa chuyên đề đã tồn tại

* **DỮ LIỆU LIÊN QUAN:**

Thông tin của mỗi chuyên đề bao gồm: mã chuyên đề, tên chuyên đề, thời lượng, học phí

* **ĐỐI TƯỢNG SỬ DỤNG:**

Sau khi đăng nhập thì nhân viên nào cũng có thể sử dụng chức năng này, ngoại trừ xóa thì chỉ có trưởng phòng mới sử dụng được.

### Quản lý người học

* **MÔ TẢ CHỨC NĂNG:**

Chức năng quản lý người học được sử dụng để quản lý thông tin của người đăng ký học. Yêu cầu của chức năng này là liệt kê danh sách thông tin của người học, xem chi tiết thông tin của mỗi người học, thêm người học mới, cập nhật hoặc xóa thông tin người học đã tồn tại

* **DỮ LIỆU LIÊN QUAN:**

Thông tin của mỗi người học bao gồm: mã người học, họ và tên, giới tính, ngày sinh, số điện thoại, địa chỉ email và ghi chú

* **ĐỐI TƯỢNG SỬ DỤNG:**

Sau khi đăng nhập thì nhân viên nào cũng có thể sử dụng chức năng này, ngoại trừ xóa thì chỉ có trưởng phòng mới sử dụng được

### Quản lý khóa học

* **MÔ TẢ CHỨC NĂNG:**

Chức năng quản lý khóa học được sử dụng để quản lý thông tin khóa học tạo ra từ chuyên đề. Yêu cầu của chức năng này là liệt kê danh sách thông tin của mỗi khóa học, xem chi tiết thông tin mỗi khóa học, thêm khóa học mới, cập nhật hoặc xóa thông tin khóa học đã tồn tại

* **DỮ LIỆU LIÊN QUAN:**

Thông tin của mỗi khóa học bao gồm: chuyên đề, ngày khai giảng, học phí, thời lượng giờ, người tạo, ngày tạo, ghi chú

* **ĐỐI TƯỢNG SỬ DỤNG:**

Sau khi đăng nhập thì nhân viên nào cũng có thể sử dụng chức năng này, ngoại trừ xóa thì chỉ có trưởng phòng mới sử dụng được

### Quản lý học viên

* **MÔ TẢ CHỨC NĂNG:**

Chức năng quản lý học viên được dùng để xem thông tin các học viên. Yêu cầu của chức năng này là xem danh sách các học viên, thêm và cập nhật học viên.

* **DỮ LIỆU LIÊN QUAN:**

Thông tin của mỗi học viên: mã học viên, mã người học, họ và tên, điểm

* **ĐỐI TƯỢNG SỬ DỤNG:**

Sau khi đăng nhập thì người dùng là trưởng phòng và nhân viên đều có thể sử dụng chức năng này

### Quản lý tổng hợp – thống kê

* **MÔ TẢ CHỨC NĂNG:**

Chức năng quản lý tổng hợp – thống kê được dùng để xem các thống kê về bảng điểm của học viên, thống kê về người học, thông kê về bảng điểm của các chuyên đề và đặc biệt là thống kê về doanh thu.

* **DỮ LIỆU LIÊN QUAN:**

Thông tin về thống kê bảng điểm bao gồm: Mã người học, họ tên, điểm, xếp loại

Thông tin về thống kê Người học bao gồm: Năm nhập học, số người đăng ký nhập học trong năm, người đăng ký sớm nhất, người đăng ký muộn nhất

Thông tin về thống kê điểm chuyên đề bao gồm: tên chuyên đề, số lượng học viên, điểm thấp nhất, điểm cao nhất, điểm trung bình

Thông tin về thống kê doanh thu bao gồm: tên chuyên đề, số khóa học, số học viên, tổng doanh thu, học phí thấp nhất, học phí cao nhất, học phí TB.

* **ĐỐI TƯỢNG SỬ DỤNG:**

Sau khi đăng nhập thì người dùng là trưởng phòng và nhân viên đều có thể sử dụng chức năng này

### Đăng nhập

* **MÔ TẢ CHỨC NĂNG:**

Chức năng đăng nhập dùng để đăng nhập tài khoản vào phần mềm.

* **DỮ LIỆU LIÊN QUAN:**

Thông tin của mỗi tài khoản đăng nhập bao gồm: tên đăng nhập và mật khẩu

* **ĐỐI TƯỢNG SỬ DỤNG:**

Dùng cho trưởng phòng và nhân viên

### Đổi mật khẩu

* **MÔ TẢ CHỨC NĂNG:**

Chức năng này dùng để đổi mật mật khẩu người dùng

* **DỮ LIỆU LIÊN QUAN:**

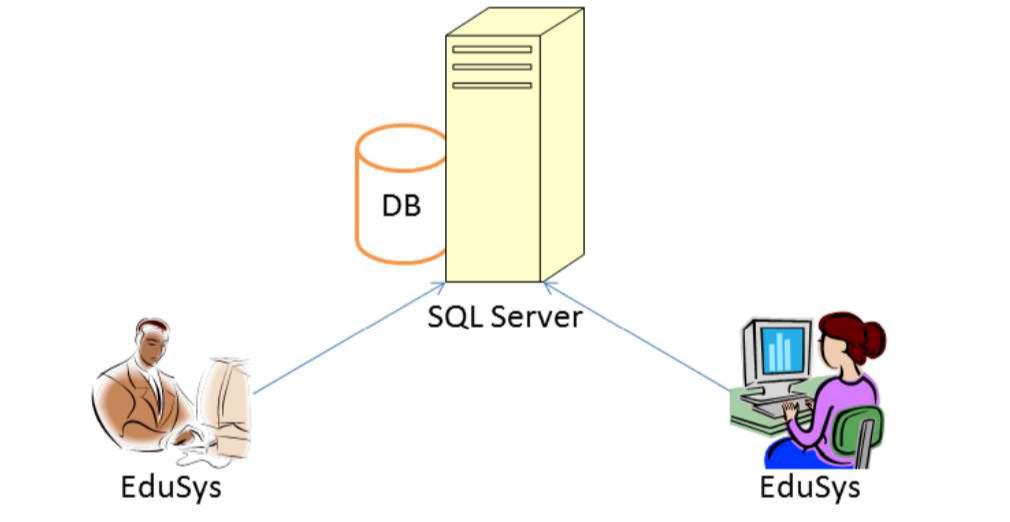
Thông tin đổi mật khẩu bao gồm: tên đăng nhập, mật khẩu cũ, mật khẩu mới, xác nhận mật khẩu mới

* **ĐỐI TƯỢNG SỬ DỤNG:**

Dùng cho trường phòng và nhân viên

## Sơ đồ triển khai và yêu cầu hệ thống

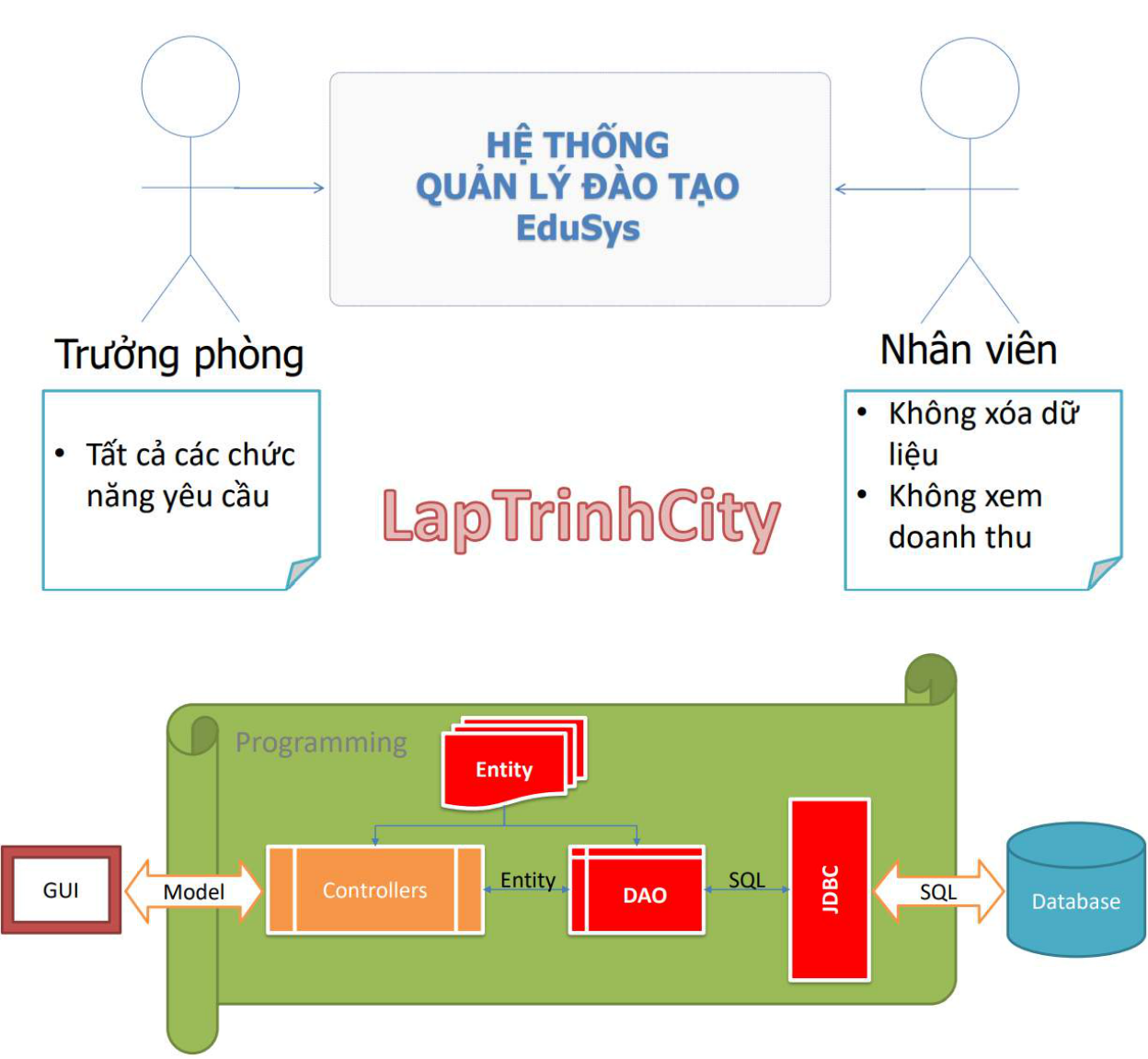
### Sơ đồ triển khai



### Yêu cầu hệ thống

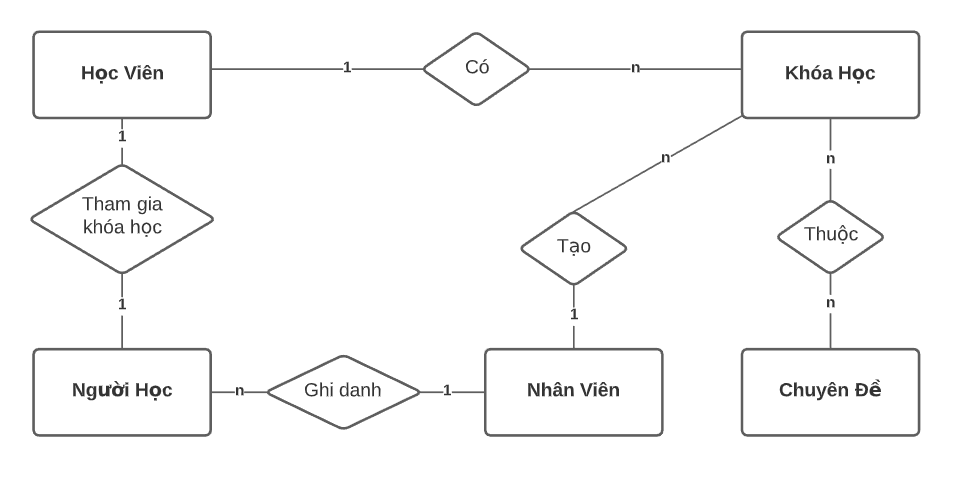
# Thiết kế ứng dụng

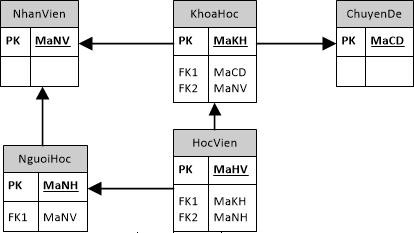
## Mô hình công nghệ ứng dụng

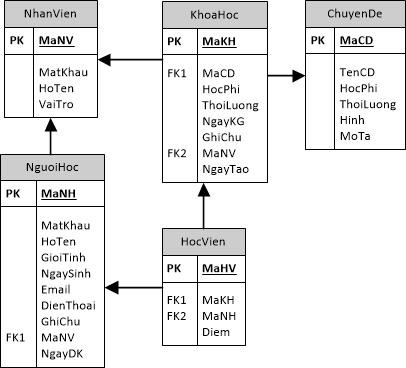


## Thực thể

### Sơ đồ quan hệ thực thể (ERD)

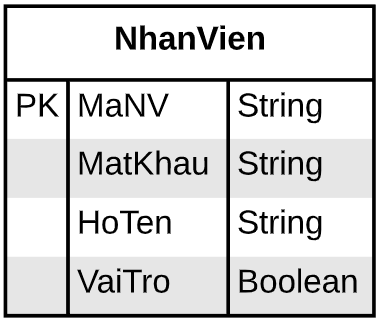






### Chi tiết thực thể

#### Thực thể Nhân viên



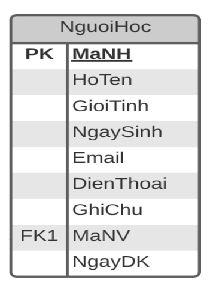
|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| MaNV | NVARCHAR(7) | Mã nhân viên |
| MatKhau | NVARCHAR(50) | Mật khẩu |
| HoTen | NVARCHAR(50) | Họ và tên nhân viên |
| VaiTro | BIT 0 | Vai trò gồm trưởng phòng và nhân viên bình thường |

#### Thực thể Chuyên đề

****

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| MaCD | NVARCHAR(7) | Mã chuyên đề |
| TenCD | NVARCHAR(50) | Tên Chuyên đề |
| HocPhi | FLOAT | Học phí của từng chuyên đề |
| ThoiLuong | INT | Thời gian học của từng chuyên đề |
| Hinh | NVARCHAR(50) | Hình ảnh biểu tượng cho chuyên đề |
| Mota | NVARCHAR(255) | Mô tả chi tiết về chuyên đề |

#### Thực thể Người học



|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| MaNH | NVARCHAR(50) | Mã người học |
| HoTen | NVARCHAR(50) | Họ và tên người học |
| GioiTinh | BIT 1 | Giới tính |
| NgaySinh | DATE | Ngày, tháng, năm sinh |
| Email | NVARCHAR(255) | Email của người học |
| DienThoai | NVARCHAR(24) | Số điện thoại |
| GhiChu | NVARCHAR(255) | Thông tin khác |
| MaNV | NVARCHAR(7) | Mã nhân viên them người học |
| NgayDK | DATE | Ngày người học đăng ký học |

#### Thực thể Khóa học



|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| MaKH | INT | Mã khoá học |
| MaCD | NVARCHAR(7) | Mã chuyên đề được them trong khoá |
| HocPhi | FLOAT | Học phí của khoá học |
| ThoiLuong | INT | Thời gian thực hiện khoá học |
| NgayKG | DATE | Ngày khai giảng |
| GhiChu | NVARCHAR(255) | Thông tin khác |
| MaNV | NVARCHAR(7) | Mã nhân viên tạo khoá học |
| NgayTao | DATE | Ngày tạo khoá học |

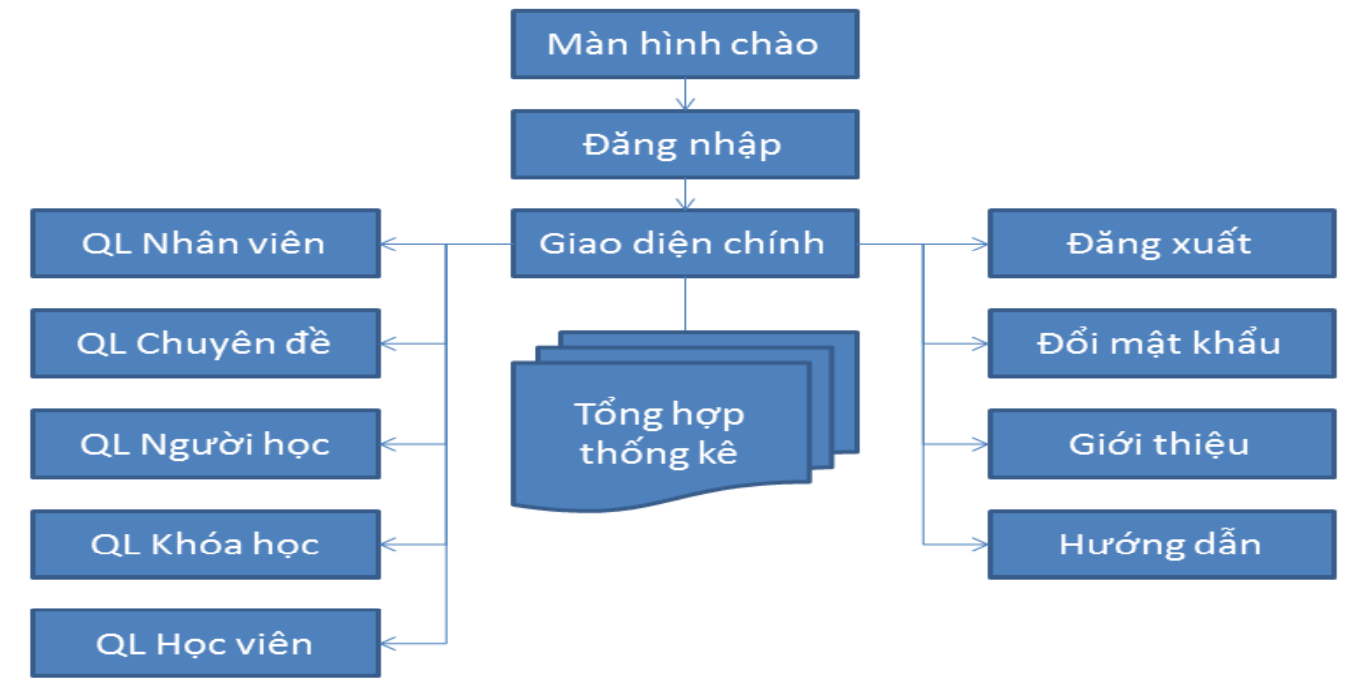
#### Thực thể Học viên



|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| MaHV | INT | Mã học viên |
| MaKH | NVARCHAR(7) | Mã khoá học |
| MaNH | NVARCHAR(7) | Mã người học |
| Diem | FLOAT | Điểm |

## Giao diện

### Sơ đồ tổ chức giao diện



### Giao diện cửa sổ chính

**Giao diện:**

**

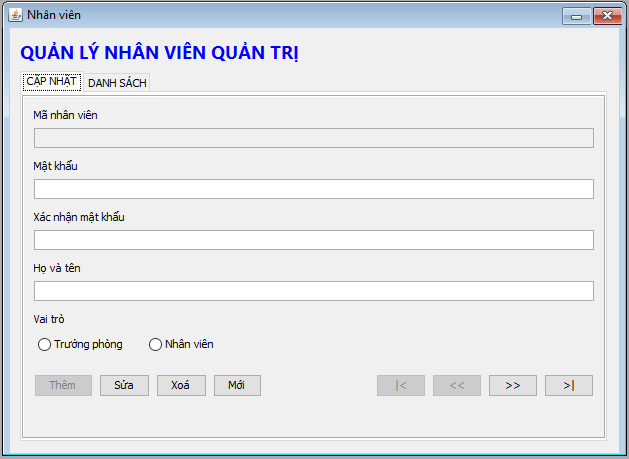
**Mô tả hoạt động**

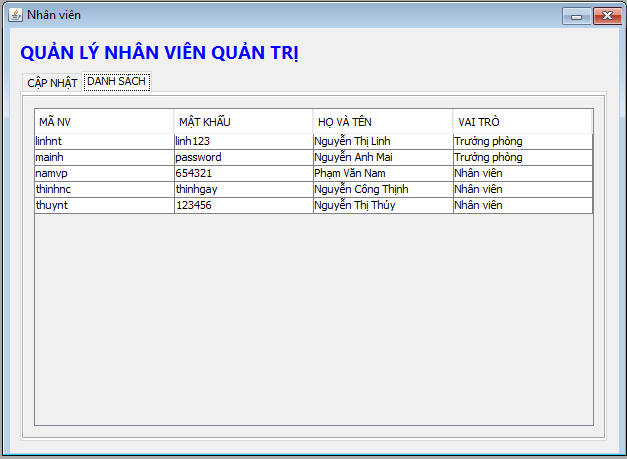
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Componel** | **Kiểu** | **Thuộc tính** |
| 1 | MainJFrame | JFrame | Title: HỆ THỐNG QUẢN LÝ ĐÀO TẠO  Layout:BorderLayout |
| 2 | mnuHeThong | JMenu | Text:Hệ thống |
| 3 | mniDangNhap | JMenuItem | Text:Đăng nhập Icon: Key.png Accelerator:Ctrl+L |
| 4 | mniDangXuat | JMenuItem | Text:Đăng xuất Icon: Log out.png Accelerator:Ctrl+O |
| 5 | mniDoiMatKhau | JMenuItem | Text:Đổi mật khẩu Icon: Refresh.png |
| 6 | mniKetThuc | JMenuItem | Text:Kết thúc Icon: Stop.png Accelerator:F10 |
| 7 | mnuQuanLy | JMenu | Text:Quản lý |
| 8 | mniQLNguoiHoc | JMenuItem | Text:Người học Icon: Conference.png |
| 9 | mniQLNhanVien | JMenuItem | Text:Nhân viên Icon:  Lists.png |
| 10 | mniQLKhoaHoc | JMenuItem | Text:Khóa học  Icon: Certificate.png |
| 11 | mniQLChuyenDe | JMenuItem | Text:Chuyên đề Icon: User group.png |
| 12 | mnuThongKe | JMenu | Text:Thống kê |
| 13 | mniTKNguoiHoc | JMenuItem | Text:Người học từng năm Icon: Clien list.png |
| 14 | mniTKBangDem | JMenuItem | Text:Bảng điểm khóa… Icon: Card file.png |
| 15 | mniTKKhoaHoc | JMenuItem | Text:Điểm từng khóa Icon: Bar chart.png |
| 16 | mniTKDoanhThu | JMenuItem | Text:Doanh thu từng chuyên đề… Icon: Dollar.png |
| 17 | mnuTroGiup | JMenu | Text:Trợ giúp |
| 18 | mniHuongDan | JMenuItem | Text:Hướngdẫnsửdụng Icon: Globe.png  Accelerator: F1 |
| 19 | mniGioiThieu | JMenuItem | Text:Giới thiệu sản phẩm Icon: Brickhouse.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  ayout:BorderLayout |
| 29 | lblTrangThai | JLabel | Layout.Direction:center Icon: Info.png  Text:Hệ quản lý đào tạo |
| 30 | lblDongHo | JLabel | Layout.Direction:after Icon:Alarm.png Text:10:55 PM |

### Giao diện chức năng

#### Cửa sổ quản lý nhân viên

**Giao diện**



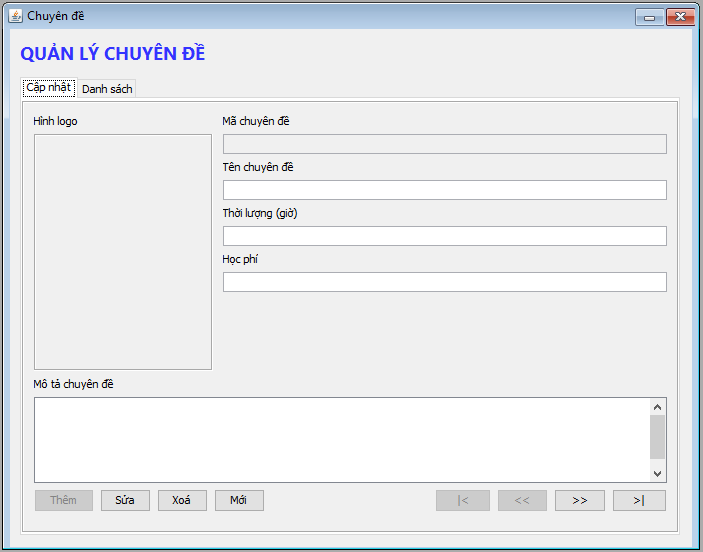


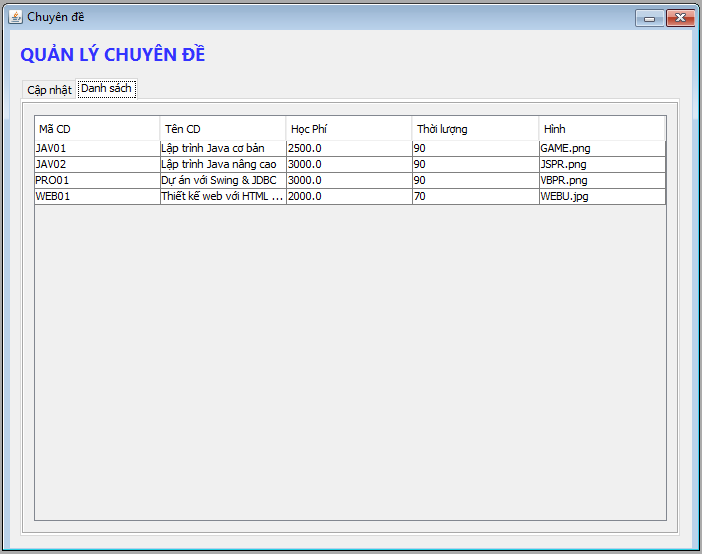
**Mô tả hoạt động:**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Điều khiển** | **Sự kiện** | **Mô tả hoạt động** |
| 1 | Cửa sổ | Initialize | Hiển thị tất cả nhân viên lên bảng |
| 2 | [Thêm] | Click | Validation, Thêm vào CSDL một nhân viên mới với dữ liệu nhập từ form |
| 3 | [Sửa] | Click | Validation, Cập nhật thông tin nhân viên đang xem trên form vào CSDL |
| 4 | [Xóa] | Click | Xóa nhân viên có mã đang xem trên form |
| 5 | [Mới] | Click | Xóa trắng form |
| 6 | Bảng | Click | Hiển thị thông tin của nhân viên trên hàng được chọn của bảng lên form để xem |
| 7 | [|<] | Click | Hiển thị thông tin của nhân viên của hàng đầu tiên của bảng lên form để xem |
| 8 | [<<] | Click | Hiển thị thông tin của nhân viên của hàng kế trước hàng được chọn của bảng lên form để xem |
| 9 | [>>] | Click | Hiển thị thông tin của nhân viên của hàng kế sau hàng được chọn của bảng lên form để xem |
| 10 | [>|] | Click | Hiển thị thông tin của nhân viên của hàng cuối cùng của bảng lên form để xem |

#### Cửa sổ quản lý chuyên đề

**Giao diện:**



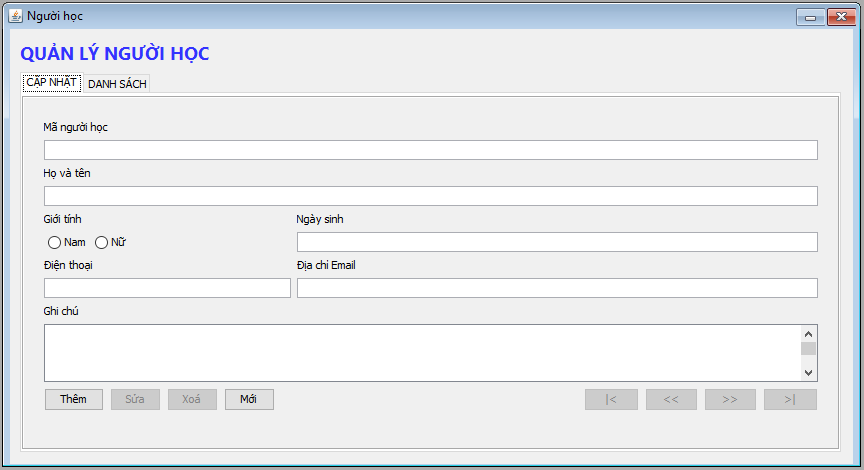


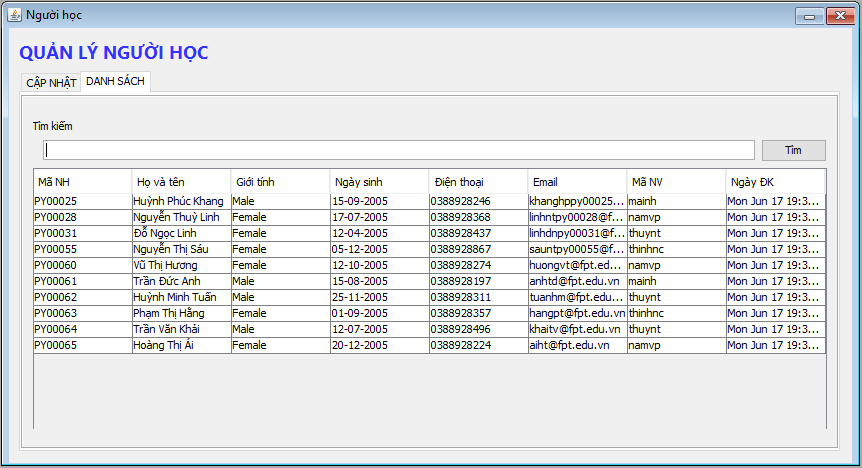
**Mô tả hoạt động:**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Điều khiển** | **Sự kiện** | **Mô tả hoạt động** |
| 1 | Cửa sổ | Initialize | Hiển thị tất cả chuyên đề lên bảng |
| 2 | [Thêm] | Click | Validation, Thêm vào CSDL một chuyên đề mới với dữ liệu nhập từ form |
| 3 | [Sửa] | Click | Validation, Cập nhật thông tin chuyên đề đang xem trên form vào CSDL |
| 4 | [Xóa] | Click | Xóa chuyên đề có mã đang xem trên form |
| 5 | [Mới] | Click | Xóa trắng form |
| 6 | Bảng | Click | Hiển thị thông tin của chuyên đề trên hàng được chọn của bảng lên form để xem |
| 7 | [|<] | Click | Hiển thị thông tin của chuyên đề của hàng đầu tiên của bảng lên form để xem |
| 8 | [<<] | Click | Hiển thị thông tin của chuyên đề của hàng kế trước hàng được chọn của bảng lên form để xem |
| 9 | [>>] | Click | Hiển thị thông tin của chuyên đề của hàng kế sau hàng được chọn của bảng lên form để xem |
| 10 | [>|] | Click | Hiển thị thông tin của chuyên đề của hàng cuối cùng của bảng lên form để xem |
| 11 | [mota] | Initalize | Hiển thị ghi chú chuyên đề |
| 12 | [Hình logo] | Click | Chọn logo cho chuyên đề |

#### Cửa sổ quản lý người học

**Giao diện:**



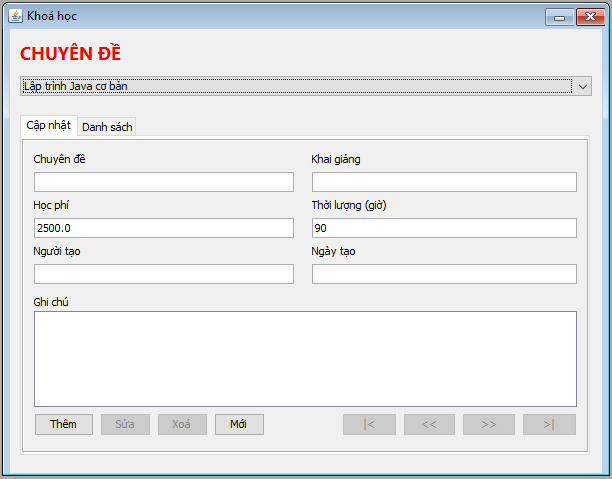


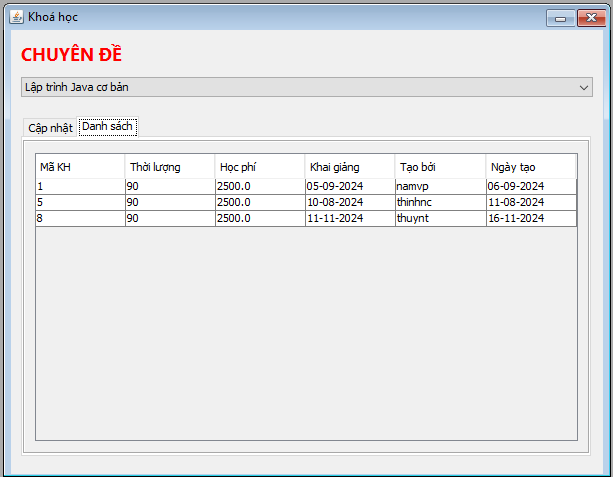
**Mô tả hoạt động:**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Điều khiển** | **Sự kiện** | **Mô tả hoạt động** |
| 1 | Cửa sổ | Initialize | Hiển thị tất cả người học lên bảng |
| 2 | [Thêm] | Click | Validation, Thêm vào CSDL một người học mới với dữ liệu nhập từ form |
| 3 | [Sửa] | Click | Validation, Cập nhật thông tin người học đang xem trên form vào CSDL |
| 4 | [Xóa] | Click | Xóa người học có đang xem trên form |
| 5 | [Mới] | Click | Xóa trắng form |
| 6 | Bảng | Click | Hiển thị thông tin của người học trên hàng được chọn của bảng lên form để xem |
| 7 | [|<] | Click | Hiển thị thông tin của người học của hàng đầu tiên của bảng lên form để xem |
| 8 | [<<] | Click | Hiển thị thông tin của người học của hàng kế trước hàng được chọn của bảng lên form để xem |
| 9 | [>>] | Click | Hiển thị thông tin của người học của hàng kế sau hàng được chọn của bảng lên form để xem |
| 10 | [>|] | Click | Hiển thị thông tin của người học của hàng cuối cùng của bảng lên form để xem |
| 11 | [mota] | Initalize | Hiển thị ghi chú người học |
| 12 | [Giới tính] | Click | Chọn giới tính cho người học |

#### Cửa sổ quản lý khóa học

**Giao diện:**



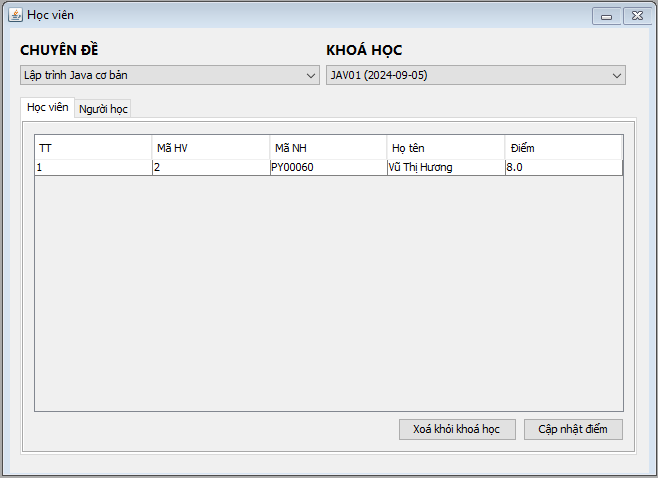


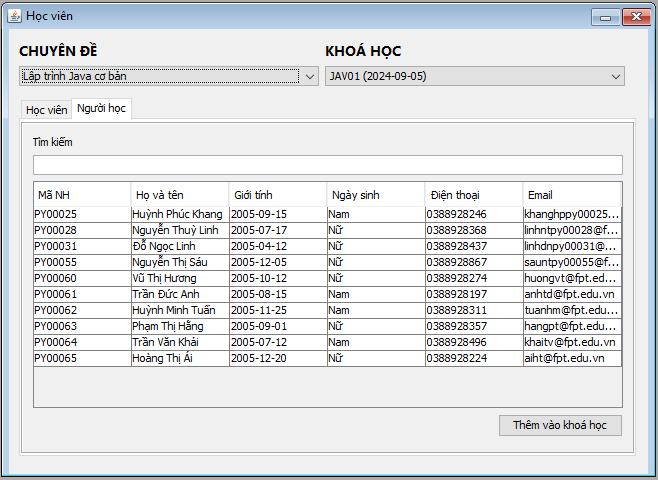
**Mô tả hoạt động:**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Điều khiển** | **Sự kiện** | **Mô tả hoạt động** |
| 1 | Cửa sổ | Initialize | Hiển thị tất cả khoá học lên bảng |
| 2 | [Thêm] | Click | Validation, Thêm vào CSDL một khoá học mới với dữ liệu nhập từ form |
| 3 | [Sửa] | Click | Validation, Cập nhật thông tin khoá học đang xem trên form vào CSDL |
| 4 | [Xóa] | Click | Xóa khoá học có mã đang xem trên form |
| 5 | [Mới] | Click | Xóa trắng form |
| 6 | Bảng | Click | Hiển thị thông tin của khoá học trên hàng được chọn của bảng lên form để xem |
| 7 | [|<] | Click | Hiển thị thông tin của khoá học của hàng đầu tiên của bảng lên form để xem |
| 8 | [<<] | Click | Hiển thị thông tin của khoá học của hàng kế trước hàng được chọn của bảng lên form để xem |
| 9 | [>>] | Click | Hiển thị thông tin của khoá học của hàng kế sau hàng được chọn của bảng lên form để xem |
| 10 | [>|] | Click | Hiển thị thông tin của khoá học của hàng cuối cùng của bảng lên form để xem |
| 11 | [mota] | Initalize | Hiển thị ghi chú khoá học |

#### Cửa sổ quản lý học viên

**Giao diện:**



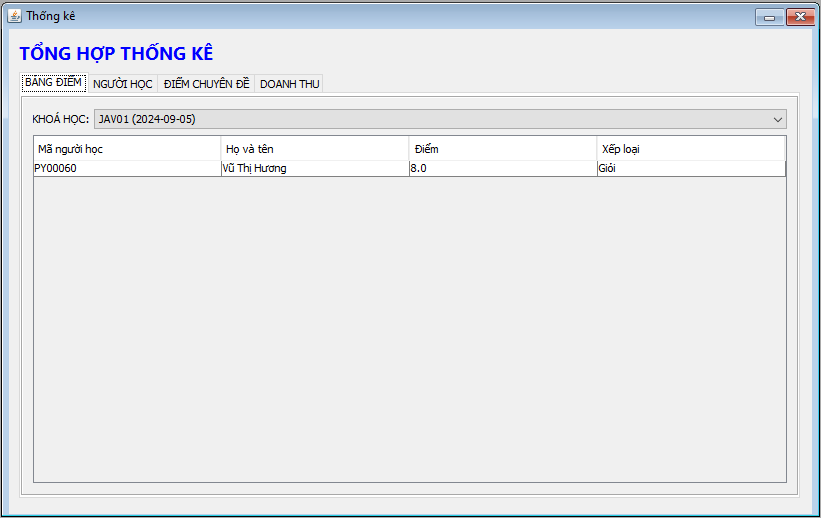


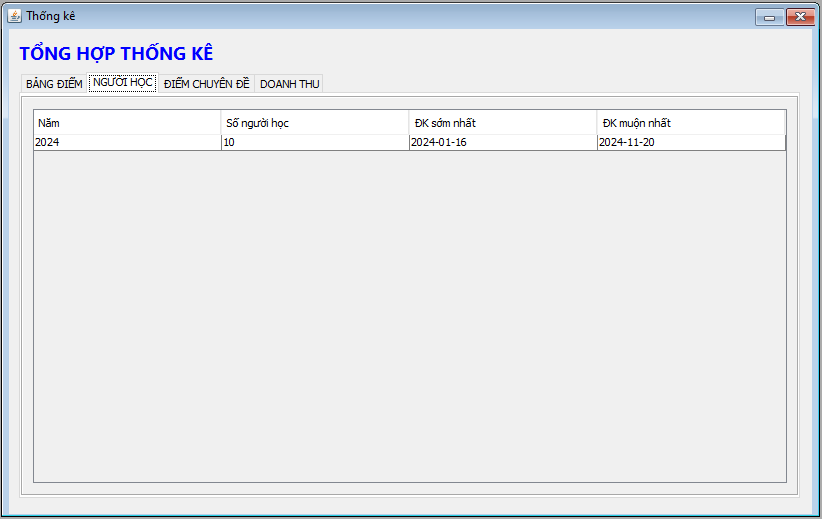
**Mô tả hoạt động:**

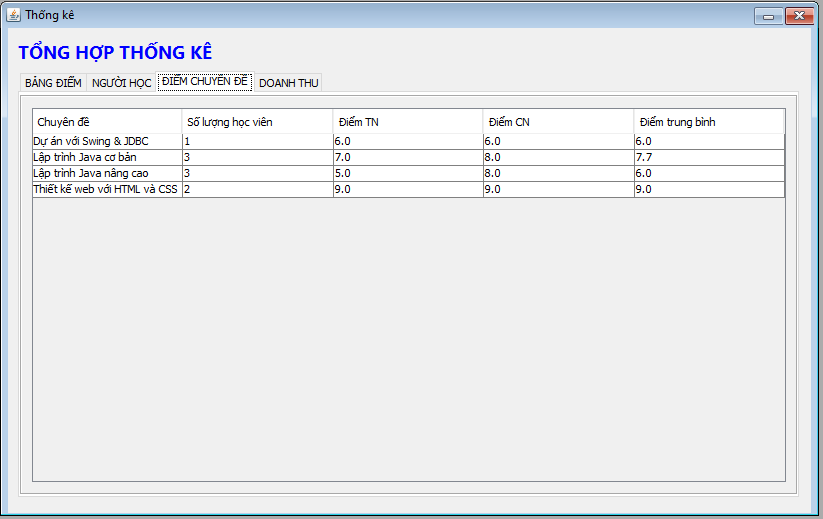
|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Điều khiển** | **Sự kiện** | **Mô tả hoạt động** |
| 1 | [CBO chuyên đề] | Click chọn | Chọn chuyên đề |
| 2 | Bảng học viên | Initalize | Hiển thị thông tin của học viên |
| 3 | [CBO khoá học] | Click chọn | Chọn khoá học |
| 4 | [Xóa khỏi khoá học] | Click | Xóa người học có đang xem trên bảng |
| 5 | [Cập nhật điểm] | Click | Cập nhật điểm của học viên đang được chọn |
| 6 | Bảng người học | Click | Hiển thị thông tin của người học |
| 7 | [Tìm kiếm] | Enter | Tìm kiếm thông tin người học theo thông tin được cung cấp trên thanh tìm kiếm |

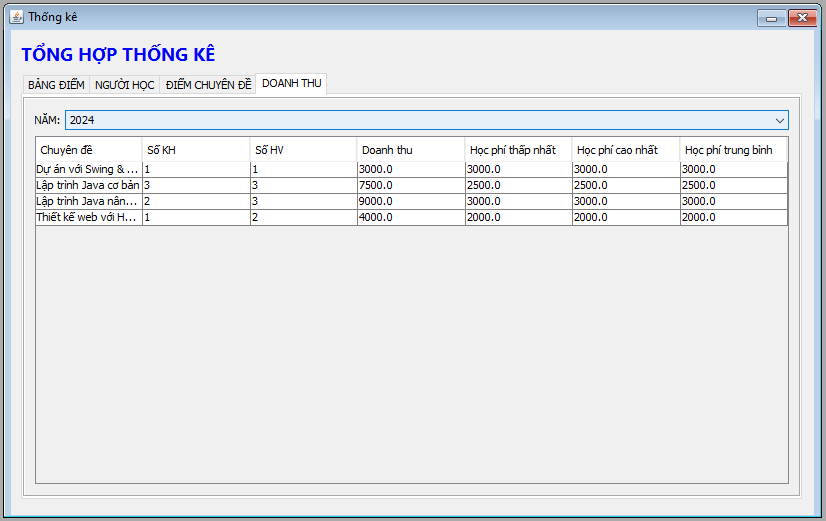
#### Cửa sổ thống kê

**Giao diện:**









**public NhanVienUI() {**

**initComponents();**

**setTitle("Nhân viên");**

**fillTable();**

**updateStatus();**

**}**

**NhanVienDAO dao = new NhanVienDAO();**

**int row = 0;**

**void fillTable() {**

**DefaultTableModel model = (DefaultTableModel) tblNhanVien.getModel();**

**model.setRowCount(0);**

**try {**

**List<NhanVien> list = dao.selectAll();**

**for (NhanVien nv : list) {**

**Object[] row = {**

**nv.getMaNV(),**

**nv.getMatKhau(),**

**nv.getHoTen(),**

**nv.isVaiTro() ? "Trưởng phòng" : "Nhân viên"**

**};**

**model.addRow(row);**

**}**

**} catch (Exception e) {**

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

**}**

**}**

**void edit() {**

**try {**

**String maNV = (String) tblNhanVien.getValueAt(this.row, 0);**

**NhanVien model = dao.selectbyId(maNV);**

**if (model != null) {**

**setForm(model);**

**updateStatus();**

**tabs.setSelectedIndex(0);**

**}**

**} catch (Exception e) {**

**Msgbox.alert(this, "Lỗi truy vấn dữ liệu");**

**}**

**}**

**void setForm(NhanVien model) {**

**txtMaNV.setText(model.getMaNV());**

**txtHoTen.setText(model.getHoTen());**

**txtMatKhau.setText(model.getMatKhau());**

**txtMatKhau2.setText(model.getMatKhau());**

**rdoTruongPhong.setSelected(model.isVaiTro());**

**rdoNhanVien.setSelected(!model.isVaiTro());**

**}**

**NhanVien getForm() {**

**NhanVien model = new NhanVien();**

**model.setMaNV(txtMaNV.getText());**

**model.setHoTen(txtHoTen.getText());**

**model.setMatKhau(new String(txtMatKhau.getPassword()));**

**model.setVaiTro(rdoTruongPhong.isSelected());**

**return model;**

**}**

**void updateStatus() {**

**boolean edit = this.row >= 0;**

**boolean first = this.row == 0;**

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

**// trang thái form**

**txtMaNV.setEditable(!edit);**

**btnThem.setEnabled(!edit);**

**btnSua.setEnabled(edit);**

**btnXoa.setEnabled(edit);**

**btnFirst.setEnabled(edit && !first);**

**btnPrev.setEnabled(edit && !first);**

**btnNext.setEnabled(edit && !last);**

**btnLast.setEnabled(edit && !last);**

**}**

**void clearForm() {**

**this.setForm(new NhanVien());**

**this.updateStatus();**

**row = -1;**

**updateStatus();**

**}**

**void insert() {**

**NhanVien model = getForm();**

**String mk2 = new String(txtMatKhau2.getPassword());**

**if (mk2.equals(model.getMatKhau())) {**

**try {**

**dao.insert(model);**

**this.fillTable();**

**this.clearForm();**

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

**} catch (Exception e) {**

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

**}**

**} else {**

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

**}**

**}**

**void update() {**

**NhanVien model = getForm();**

**String confirm = new String(txtMatKhau2.getPassword());**

**if (!confirm.equals(model.getMatKhau())) {**

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

**} else {**

**try {**

**dao.update(model);**

**this.fillTable();**

**Msgbox.alert(this, "Cập nhật thành công");**

**} catch (Exception e) {**

**Msgbox.alert(this, "Cập nhật thất bại");**

**}**

**}**

**}**

**void delete() {**

**if (!Auth.isManager()) {**

**Msgbox.alert(this, "Bạn không có quyền xoá nhân viên");**

**} else {**

**if (Msgbox.confirm(this, "Bạn thực sự muốn xoá nhân viên này?")) {**

**String manv = txtMaNV.getText();**

**try {**

**dao.delete(manv);**

**this.fillTable();**

**this.clearForm();**

**Msgbox.alert(this, "Bạn xoá thành công!");**

**} catch (Exception e) {**

**Msgbox.alert(this, "Xoá thất bại!");**

**}**

**}**

**}**

**}**

**void fisrt() {**

**row = 0;**

**edit();**

**}**

**void prev() {**

**if (row > 0) {**

**row--;**

**edit();**

**}**

**}**

**void next() {**

**if (row < tblNhanVien.getRowCount() - 1) {**

**row++;**

**edit();**

**}**

**}**

**void last() {**

**row = tblNhanVien.getRowCount() - 1;**

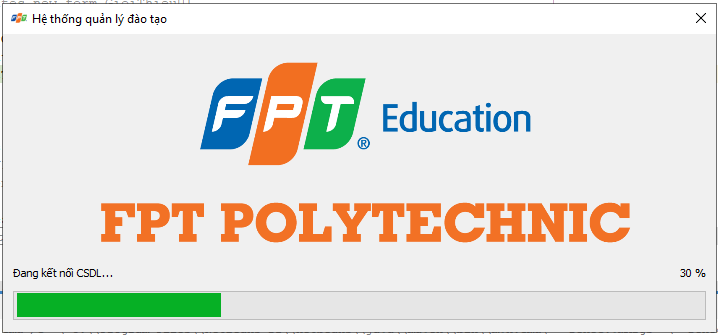
**edit();**

**}**

### Các giao diện hỗ trợ khác

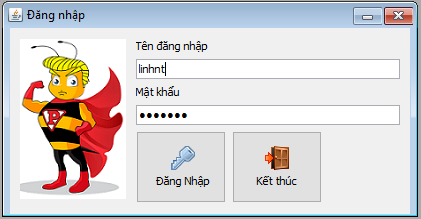
#### Cửa sổ chào

**Giao diện**



#### Cửa sổ đăng nhập

**Giao diện**

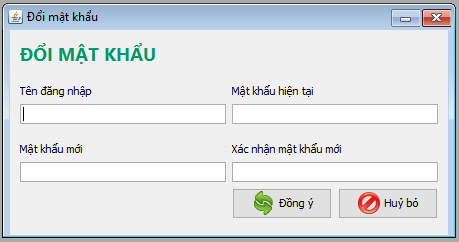


**Mô tả hoạt động**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Điều khiển** | **Sự kiện** | **Mô tả hoạt động** |
| 1 | [đăng nhập] | click | validation, đăng nhập tài khoản |
| 2 | [kết thúc] | click | tắt cửa sổ đăng nhập |

#### Cửa sổ đổi mật khẩu

**Giao diện:**



**Mô tả hoạt động:**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Điều khiển** | **Sự kiện** | **Mô tả hoạt động** |
| 1 | [đồng ý] | click | validation, đổi mật khẩu |
| 2 | [huỷ bỏ] | click | huỷ bỏ quá trình đổi mật khẩu, tắt cửa sổ |

#### Cửa sổ giới thiệu

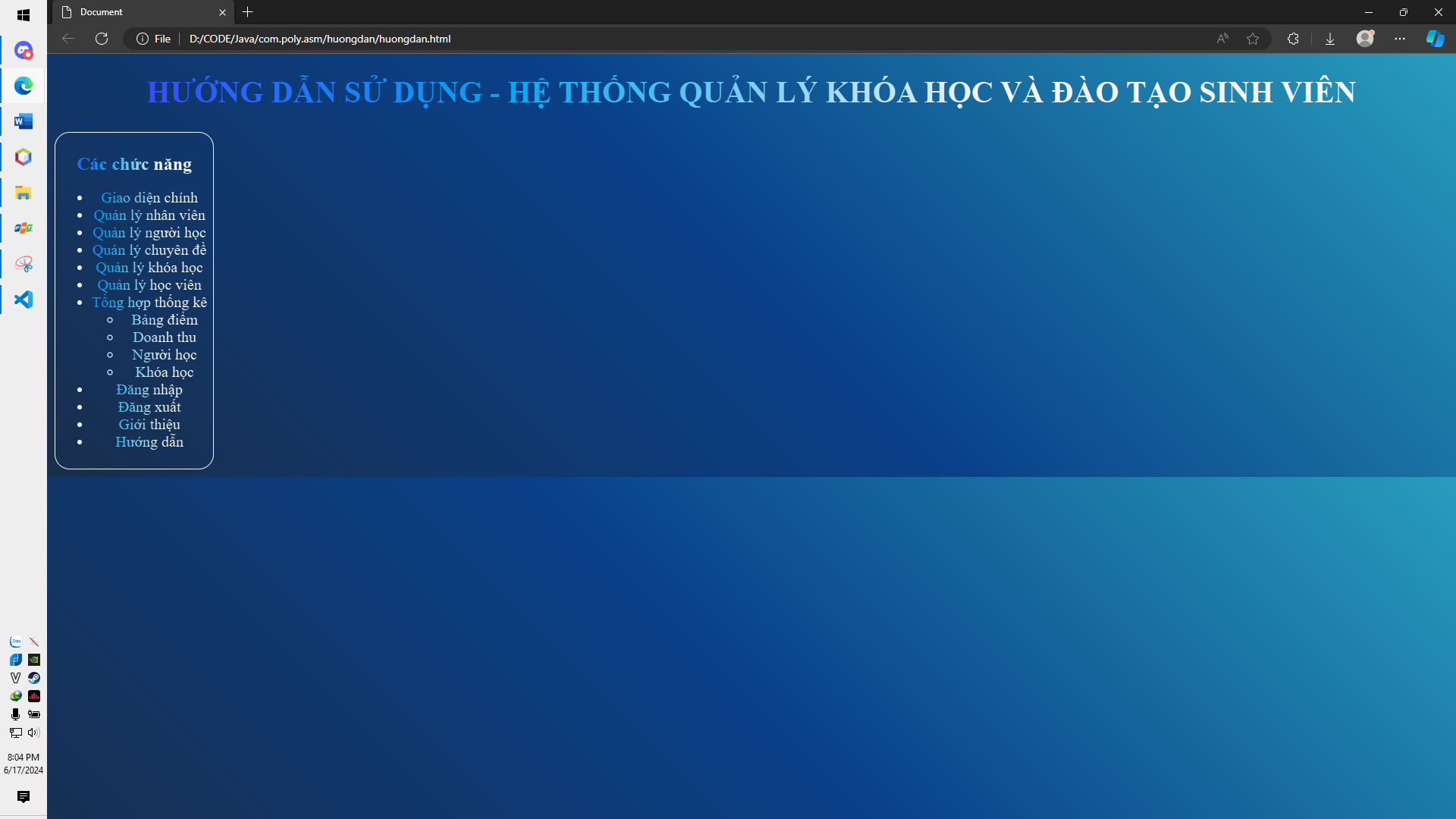
**Giao diện:**



**Mô tả hoạt động:** hiển thị các thông tin và các yêu cầu cần thiết để chạy app

#### Cửa sổ hướng dẫn

**Giao diện:**



**Mô tả hoạt động:** hướng dẫn sử dụng app

# Thực hiện dự án

## Tạo giao diện với Swing

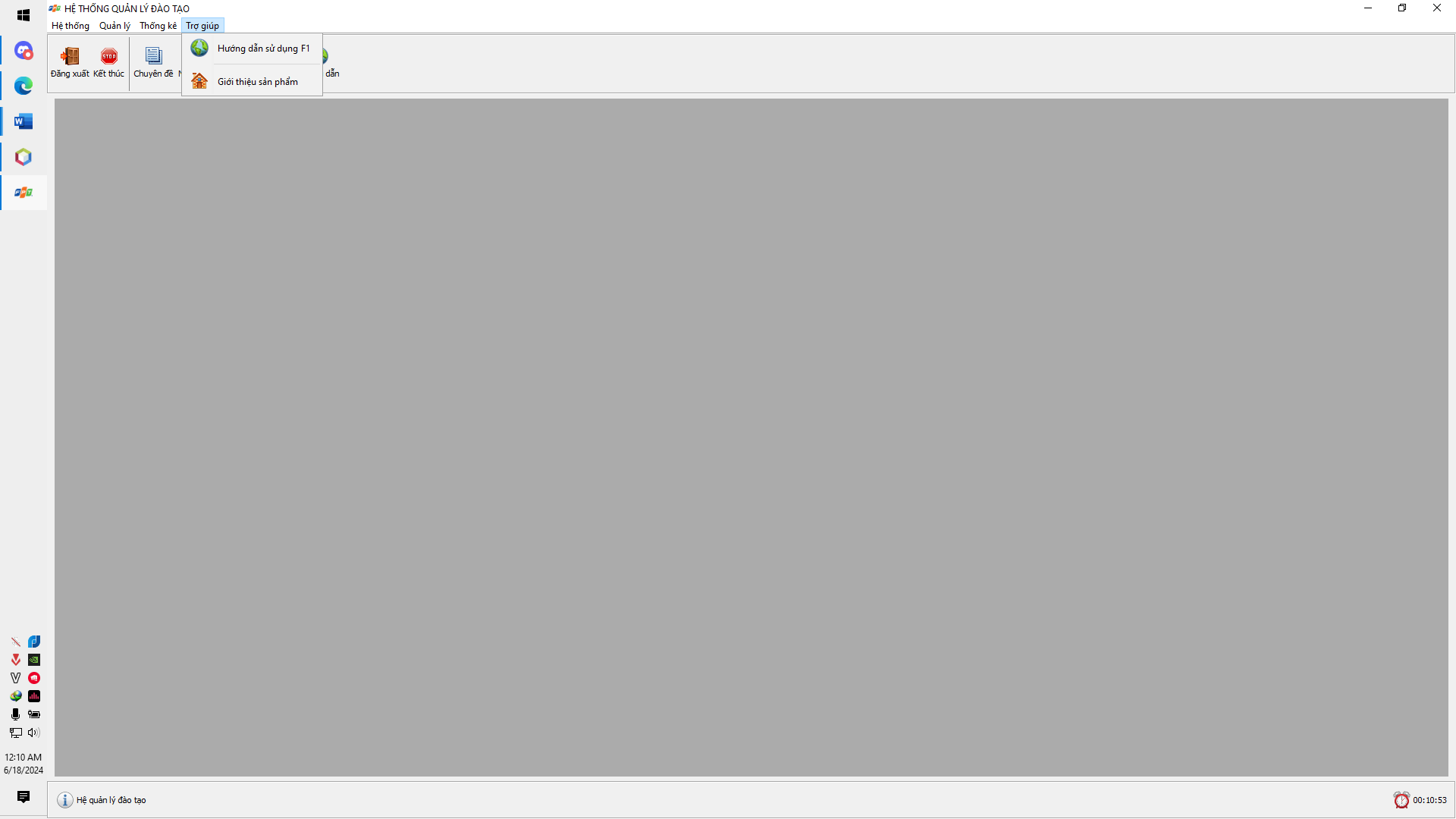
### Cửa sổ chính (EduSysJFrame)

**Giao diện**









**public MainAppUI() {**

**initComponents();**

**setLocationRelativeTo(null);**

**init();**

**initClock();**

**}**

**void init() {**

**setIconImage(XImage.getAppIcon());**

**setLocationRelativeTo(null);**

**setTitle("HỆ THỐNG QUẢN LÝ ĐÀO TẠO");**

**setExtendedState(JFrame.MAXIMIZED\_BOTH);**

**}**

**private Timer timer;**

**private void initClock() {**

**timer = new Timer(1000, new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**SimpleDateFormat formatter = new SimpleDateFormat("HH:mm:ss");**

**Date date = new Date();**

**lblDongHo.setText(formatter.format(date));**

**}**

**});**

**timer.start();**

**}**

**// phần hệ thống**

**public void openDoiMatKhauUI() {**

**if (Auth.isLogin()) {**

**DoiMatKhauUI DoiMK = new DoiMatKhauUI();**

**openWindow(Desktop, DoiMK);**

**DoiMK.setVisible(true);**

**} else {**

**Msgbox.alert(this, "Vui lòng đăng nhập");**

**}**

**}**

**public void openDangNhapUI() {**

**DangNhapUI dn = new DangNhapUI();**

**openWindow(Desktop, dn);**

**// Desktop.add(dn);**

**dn.setVisible(true);**

**// openWindow(Desktop,dn);**

**}**

**void dangXuat() {**

**Auth.clear();**

**closeAllInternalFrames();**

**openDangNhapUI();**

**}**

**private void closeAllInternalFrames() {**

**JInternalFrame[] frames = Desktop.getAllFrames();**

**for (JInternalFrame frame : frames) {**

**frame.dispose();**

**}**

**}**

**void ketThuc() {**

**if (Msgbox.confirm(this, "Bạn muốn kết thúc làm việc")) {**

**System.exit(0);**

**}**

**}**

**// phần quản lý**

**public void openChuyenDeUI() {**

**if (Auth.isLogin()) {**

**ChuyenDeUI qlcd = new ChuyenDeUI();**

**openWindow(Desktop, qlcd);**

**qlcd.setVisible(true);**

**} else {**

**Msgbox.alert(this, "Vui lòng đăng nhập");**

**}**

**}**

**public void openKhoaHocUI() {**

**if (Auth.isLogin()) {**

**KhoaHocUI qlkh = new KhoaHocUI();**

**openWindow(Desktop, qlkh);**

**qlkh.setVisible(true);**

**} else {**

**Msgbox.alert(this, "Vui lòng đăng nhập");**

**}**

**}**

**public void openNguoiHocUI() {**

**if (Auth.isLogin()) {**

**NguoiHocUI qlnh = new NguoiHocUI();**

**openWindow(Desktop, qlnh);**

**qlnh.setVisible(true);**

**} else {**

**Msgbox.alert(this, "Vui lòng đăng nhập");**

**}**

**}**

**public void openHocVienUI() {**

**if (Auth.isLogin()) {**

**HocVienUI qlhv = new HocVienUI();**

**openWindow(Desktop, qlhv);**

**qlhv.setVisible(true);**

**} else {**

**Msgbox.alert(this, "Vui lòng đăng nhập");**

**}**

**}**

**public void openNhanVienUI() {**

**if (Auth.isLogin()) {**

**NhanVienUI qlnv = new NhanVienUI();**

**openWindow(Desktop, qlnv);**

**qlnv.setVisible(true);**

**} else {**

**Msgbox.alert(this, "Vui lòng đăng nhập");**

**}**

**}**

**//phần thống kê**

**public void openThongKeUI(int index) {**

**if (Auth.isLogin()) {**

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

**Msgbox.alert(this, "Bạn không có quyền xem doanh thu!");**

**return;**

**} else {**

**ThongKeUI tk = new ThongKeUI();**

**openWindow(Desktop, tk);**

**tk.seclectTab(index);**

**}**

**} else {**

**Msgbox.alert(this, "Vui lòng đăng nhập lại!");**

**}**

**}**

**// phần trợ giúp**

**public void openGioiThieuUI() {**

**if (Auth.isLogin()) {**

**GioiThieuUI gt = new GioiThieuUI();**

**openWindow(Desktop, gt);**

**gt.setVisible(true);**

**} else {**

**Msgbox.alert(this, "Vui lòng đăng nhập");**

**}**

**}**

**public void openHuongDanUI() {**

**if (!java.awt.Desktop.isDesktopSupported()) {**

**JOptionPane.showMessageDialog(this, "Desktop API không được hỗ trợ trên nền tảng này.", "Lỗi",**

**JOptionPane.ERROR\_MESSAGE);**

**return;**

**}**

**java.awt.Desktop desktop = java.awt.Desktop.getDesktop();**

**if (!desktop.isSupported(java.awt.Desktop.Action.OPEN)) {**

**JOptionPane.showMessageDialog(this, "Hành động mở file không được hỗ trợ trên nền tảng này.", "Lỗi",**

**JOptionPane.ERROR\_MESSAGE);**

**return;**

**}**

**try {**

**// Đường dẫn tới file hướng dẫn**

**File file = new File("huongdan/huongdan.html");**

**if (!file.exists()) {**

**throw new FileNotFoundException("File hướng dẫn không tồn tại.");**

**}**

**desktop.open(file);**

**} catch (FileNotFoundException e) {**

**JOptionPane.showMessageDialog(this, "File hướng dẫn không tồn tại.", "Lỗi", JOptionPane.ERROR\_MESSAGE);**

**} catch (IOException e) {**

**JOptionPane.showMessageDialog(this, "Có lỗi xảy ra khi mở file hướng dẫn.", "Lỗi",**

**JOptionPane.ERROR\_MESSAGE);**

**} catch (Exception e) {**

**JOptionPane.showMessageDialog(this, "Một lỗi không xác định đã xảy ra.", "Lỗi", JOptionPane.ERROR\_MESSAGE);**

**}**

**}**

**// chỉnh màn hình**

**private void openWindow(JDesktopPane desktopPane, JInternalFrame jInternalFrame) {**

**Dimension desktopSize = desktopPane.getSize();**

**System.out.println("Desktop size width= " + desktopSize.width);**

**System.out.println("Desktop size height= " + desktopSize.height);**

**Dimension jInternalFrameSize = jInternalFrame.getSize();**

**System.out.println("Component size width= " + jInternalFrameSize.width);**

**System.out.println("Component size height= " + jInternalFrameSize.height);**

**int x = (desktopSize.width - jInternalFrameSize.width) / 2;**

**int y = (desktopSize.height - jInternalFrameSize.height) / 2;**

**System.out.println("Location x=" + x);**

**System.out.println("Location y=" + y);**

**jInternalFrame.setLocation(x, y);**

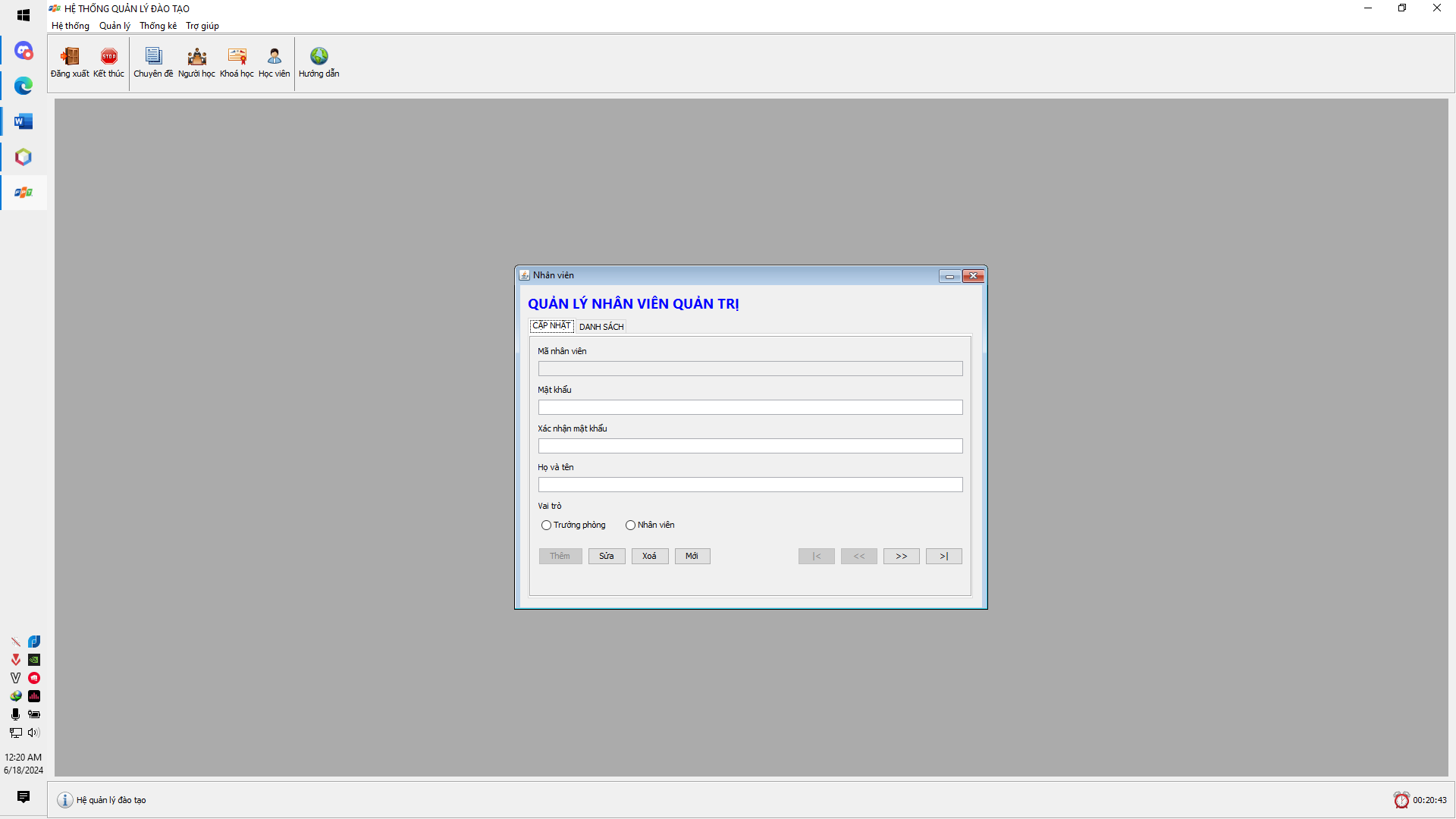
**desktopPane.add(jInternalFrame).setVisible(true);**

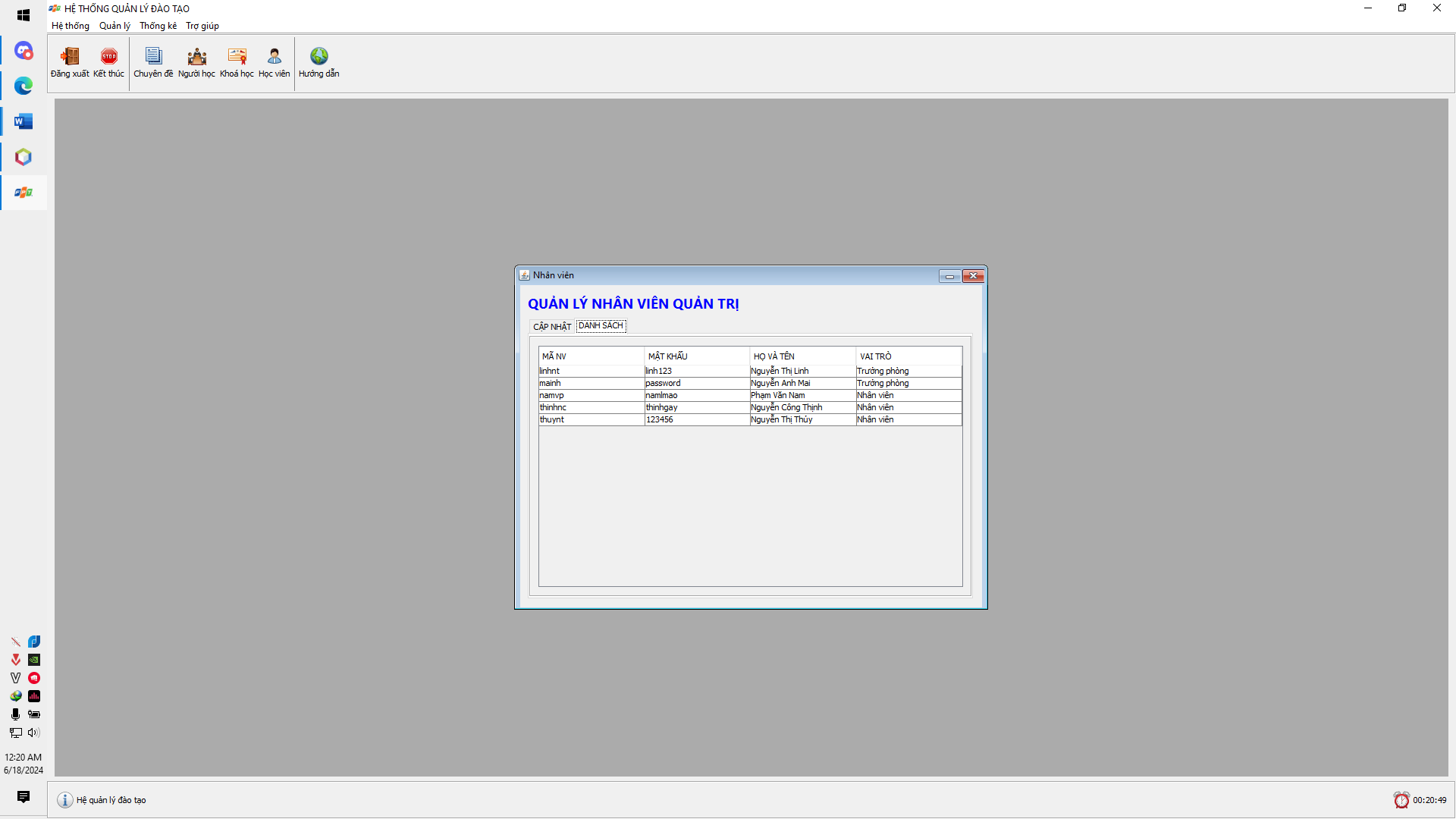
**}**

### Các cửa sổ quản lý

#### Cửa sổ quản lý nhân viên (NhanVienJInternalFrame)

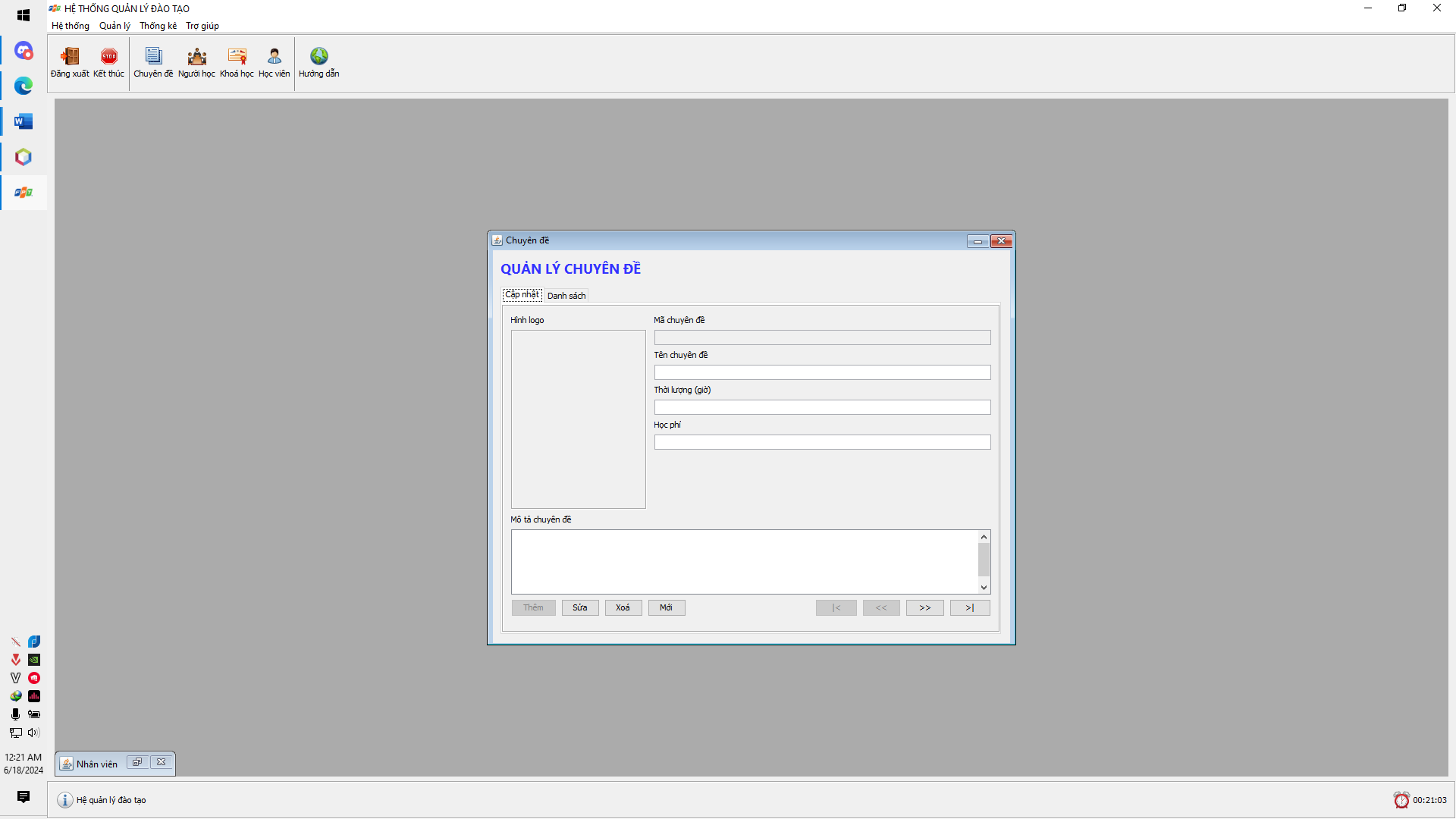
**Giao diện**

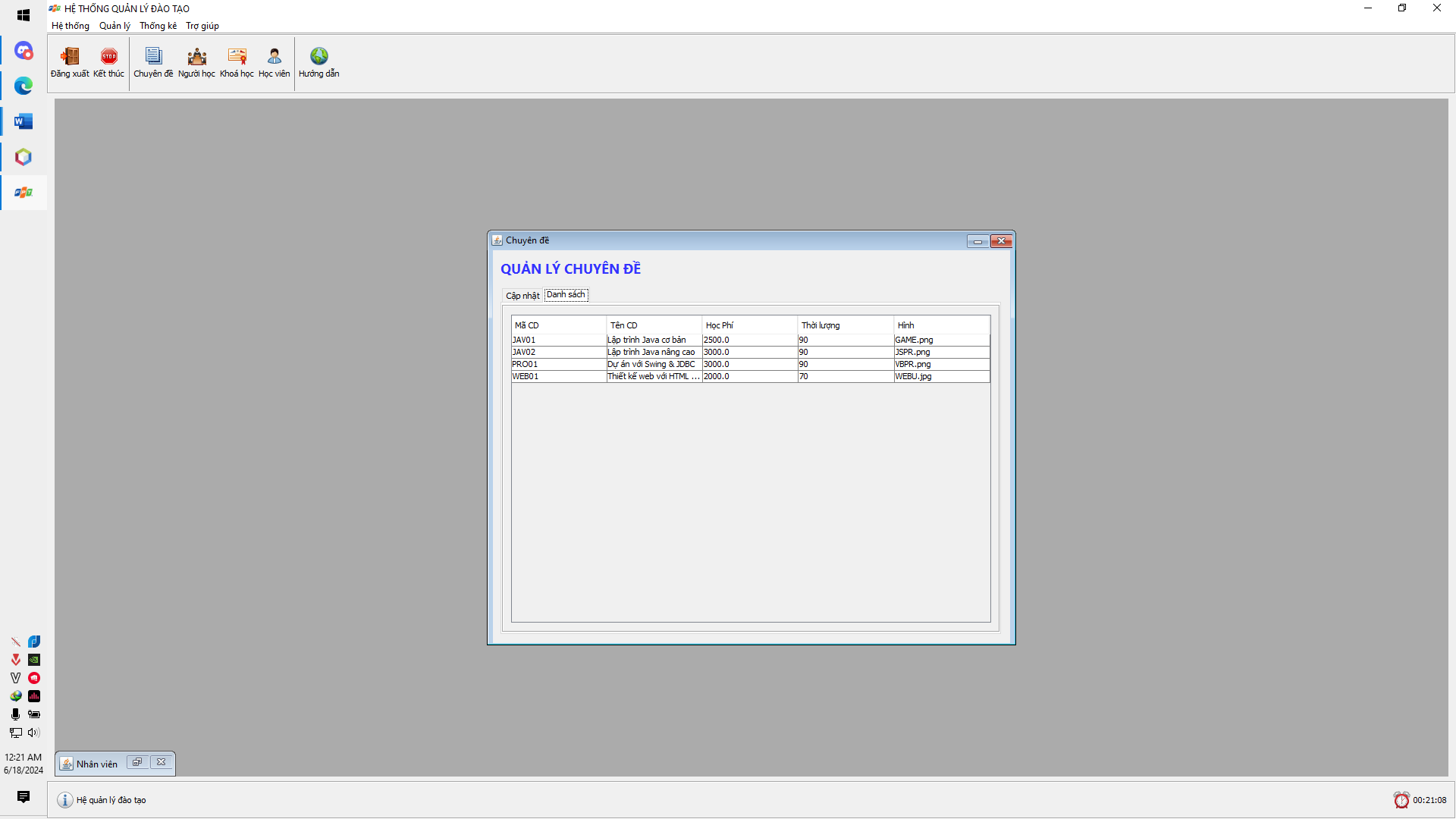




#### Cửa sổ quản lý chuyên đề (ChuyenDeJInternalFrame)

**Giao diện**





**public ChuyenDeUI() {**

**initComponents();**

**setTitle("Chuyên đề");**

**fillTable();**

**updateStatus();**

**}**

**ChuyenDeDAO dao = new ChuyenDeDAO();**

**JFileChooser fileChooser = new JFileChooser();**

**int row = 0;**

**public void fillTable() {**

**DefaultTableModel model = (DefaultTableModel) tblChuyenDe.getModel();**

**model.setRowCount(0);**

**try {**

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

**for (ChuyenDe cd : list) {**

**Object[] row = {**

**cd.getMaCD(),**

**cd.getTenCD(),**

**cd.getHocPhi(),**

**cd.getThoiLuong(),**

**cd.getHinh()**

**};**

**model.addRow(row);**

**}**

**} catch (Exception e) {**

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

**}**

**}**

**void chonAnh() {**

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

**File file = fileChooser.getSelectedFile();**

**XImage.save(file);**

**ImageIcon icon = XImage.read(file.getName());**

**lblAnh.setIcon(icon);**

**lblAnh.setToolTipText(file.getName());**

**}**

**}**

**void setForm(ChuyenDe model) {**

**txtMaCD.setText(model.getMaCD());**

**txtTenCD.setText(model.getTenCD());**

**txtThoiLuong.setText(String.valueOf(model.getThoiLuong()));**

**txtHocPhi.setText(String.valueOf(model.getHocPhi()));**

**txtMoTa.setText(model.getMoTa());**

**String hinh = model.getHinh();**

**if (hinh != null && !hinh.isEmpty()) {**

**lblAnh.setToolTipText(model.getHinh());**

**lblAnh.setIcon(XImage.read(model.getHinh()));**

**} else {**

**lblAnh.setIcon(null);**

**lblAnh.setToolTipText("");**

**}**

**}**

**ChuyenDe getForm() {**

**ChuyenDe cd = new ChuyenDe();**

**cd.setMaCD(txtMaCD.getText());**

**cd.setTenCD(txtTenCD.getText());**

**cd.setThoiLuong(Integer.parseInt(txtThoiLuong.getText()));**

**cd.setHocPhi(Double.parseDouble(txtHocPhi.getText()));**

**cd.setMoTa(txtMoTa.getText());**

**cd.setHinh(lblAnh.getToolTipText());**

**return cd;**

**}**

**void edit() {**

**try {**

**String maCD = (String) tblChuyenDe.getValueAt(this.row, 0);**

**ChuyenDe cd = dao.selectbyId(maCD);**

**if (cd != null) {**

**setForm(cd);**

**updateStatus();**

**tabs.setSelectedIndex(0);**

**}**

**} catch (Exception e) {**

**Msgbox.alert(this, "Lỗi truy vấn dữ liệu");**

**}**

**}**

**void updateStatus() {**

**boolean edit = this.row >= 0;**

**boolean first = this.row == 0;**

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

**// trang thái form**

**txtMaCD.setEditable(!edit);**

**btnThem.setEnabled(!edit);**

**btnSua.setEnabled(edit);**

**btnXoa.setEnabled(edit);**

**btnFirst.setEnabled(edit && !first);**

**btnPrev.setEnabled(edit && !first);**

**btnNext.setEnabled(edit && !last);**

**btnLast.setEnabled(edit && !last);**

**}**

**void clearForm() {**

**this.setForm(new ChuyenDe());**

**this.updateStatus();**

**this.row = -1;**

**updateStatus();**

**}**

**void insert() {**

**ChuyenDe cd = getForm();**

**try {**

**dao.insert(cd);**

**fillTable();**

**clearForm();**

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

**} catch (Exception e) {**

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

**}**

**}**

**void update() {**

**ChuyenDe cd = getForm();**

**try {**

**dao.update(cd);**

**fillTable();**

**Msgbox.alert(this, "Cập nhật thành công");**

**} catch (Exception e) {**

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

**}**

**}**

**// nhân viên tạo chuyên đề thì có thể xoá chuyên đề đc và cả trưởng phòng chú ý**

**void delete() {**

**if (!Auth.isManager()) {**

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

**} else {**

**String id = txtMaCD.getText();**

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

**try {**

**dao.delete(id);**

**fillTable();**

**clearForm();**

**Msgbox.alert(this, "Delete sucessfully!");**

**} catch (Exception e) {**

**Msgbox.alert(this, "Delete unsucessfully!");**

**}**

**}**

**}**

**}**

**void first() {**

**row = 0;**

**edit();**

**}**

**void prev() {**

**if (row > 0) {**

**row--;**

**edit();**

**}**

**}**

**void next() {**

**if (row < tblChuyenDe.getRowCount() - 1) {**

**row++;**

**edit();**

**}**

**}**

**void last() {**

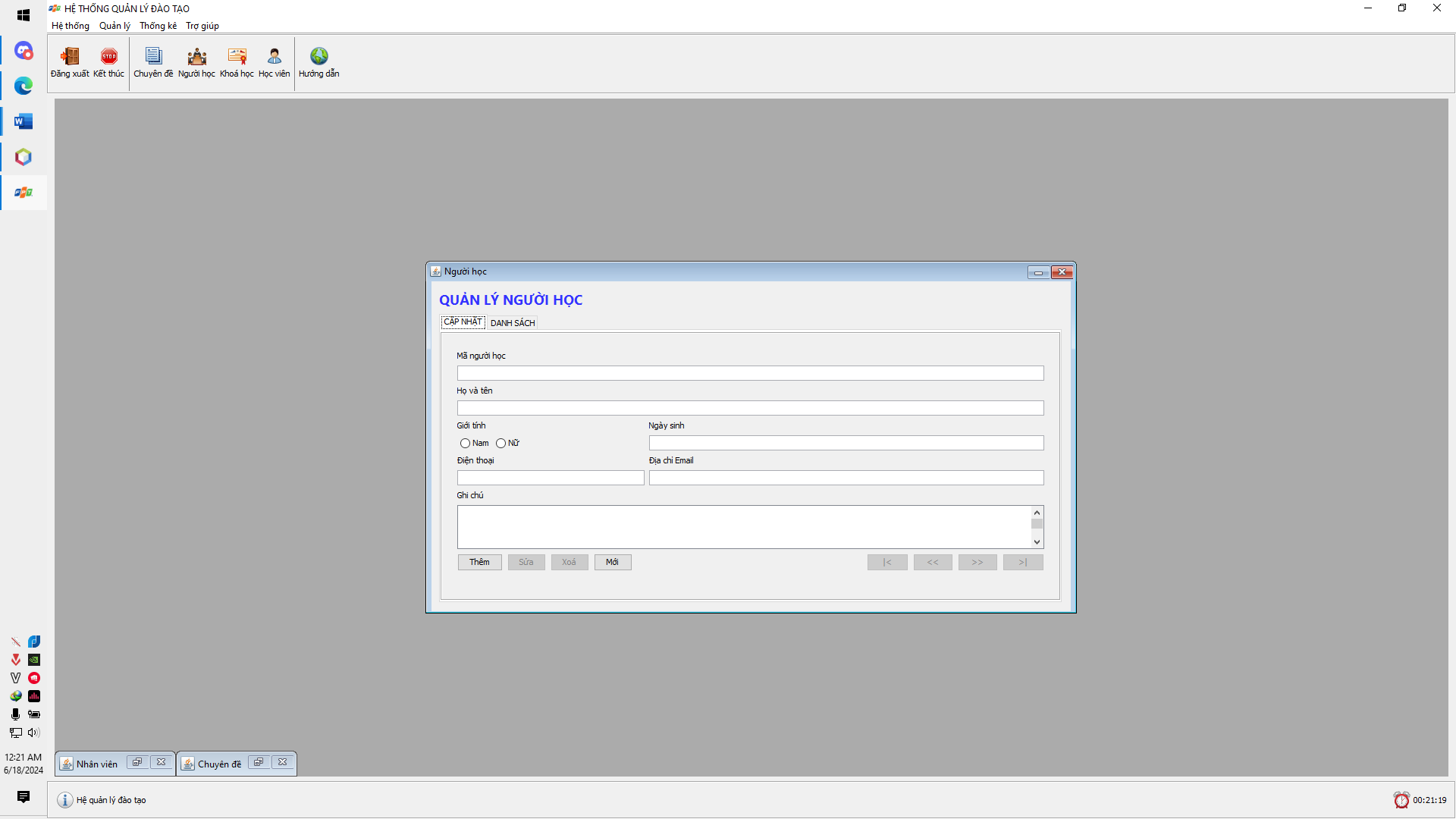
**row = tblChuyenDe.getRowCount() - 1;**

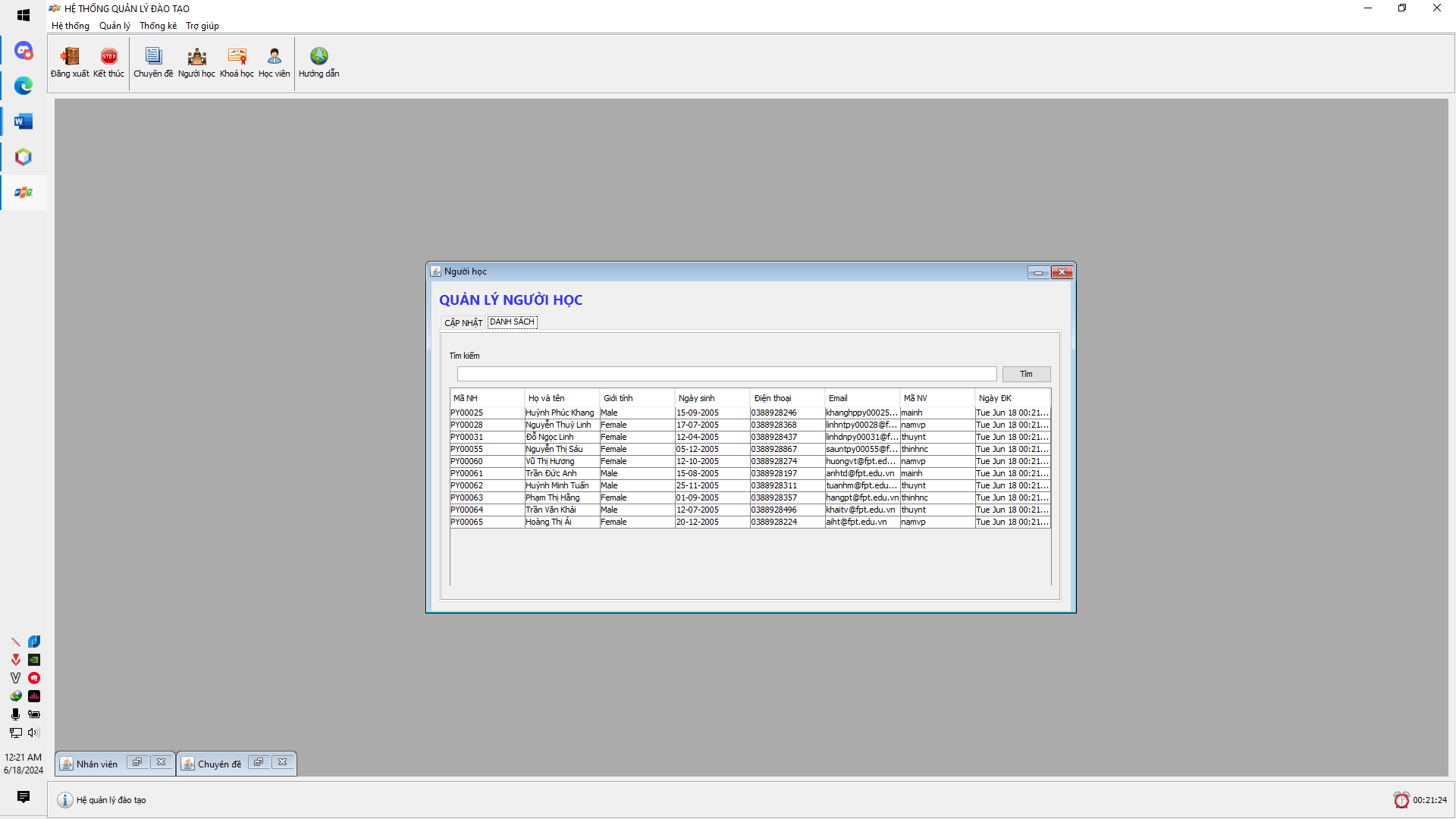
**edit();**

**}**

#### Cửa sổ quản lý người học (NguoiHocJInternalFrame)

**Giao diện**





public learner() {

initComponents();

fillTable();

updateStatus();

}

void fillTable() {

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

model.setRowCount(0);

try {

String keyword = txtTimKiem.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 edit() {

String manh = (String) tblNguoiHoc.getValueAt(this.row, 0);

NguoiHoc nh = dao.selectById(manh);

this.setForm(nh);

tabs.setSelectedIndex(0);

updateStatus();

}

void setForm(NguoiHoc nh) {

txtMaNH.setText(nh.getMaNH());

txtHoTen.setText(nh.getHoTen());

//txtNgaySinh.setText(nh.getNgaySinh());

if (nh.getNgaySinh() != null) {

txtNgaySinh.setText(XDate.toString(nh.getNgaySinh(), "dd-MM-yyyy"));

} else {

txtNgaySinh.setText("");

}

txtEmail.setText(nh.getEmail());

txtDienThoai.setText(nh.getDienThoai());

txtGhiChu.setText(nh.getGhiChu());

rdoMale.setSelected(nh.isGioiTinh());

rdoFemale.setSelected(!nh.isGioiTinh());

}

NguoiHoc getForm() {

NguoiHoc nh = new NguoiHoc();

nh.setMaNH(txtMaNH.getText());

nh.setHoTen(txtHoTen.getText());

nh.setDienThoai(txtDienThoai.getText());

nh.setGhiChu(txtGhiChu.getText());

nh.setNgayDK(new Date());

try {

String ngaySinhStr = txtNgaySinh.getText();

SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");

Date ngaySinh = sdf.parse(ngaySinhStr);

nh.setNgaySinh(ngaySinh);

}catch(ParseException e) {

e.printStackTrace();

// Xử lý lỗi chuyển đổi ngày tháng nếu cần

}

nh.setEmail(txtEmail.getText());

if (rdoMale.isSelected()) {

nh.setGioiTinh(true);

} else if (rdoFemale.isSelected()) {

nh.setGioiTinh(false);

}

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

return nh;

}

void updateStatus() {

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

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

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

// Form state

txtMaNH.setEditable(!edit);

btnThem.setEnabled(!edit);

btnSua.setEnabled(edit);

btnXoa.setEnabled(edit);

// Directional state

btnFirst.setEnabled(edit && !first);

btnPrev.setEnabled(edit && !first);

btnNext.setEnabled(edit && !last);

btnLast.setEnabled(edit && !last);

}

void clearForm() {

this.setForm(new NguoiHoc());

this.updateStatus();

row = -1;

updateStatus();

}

void insert() {

NguoiHoc nh = getForm();

try {

dao.insert(nh);

fillTable();

clearForm();

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

} catch (Exception e) {

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

}

}

void update() {

NguoiHoc nh = getForm();

try {

dao.update(nh);

fillTable();

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

} catch (Exception e) {

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

}

}

void delete() {

if (!Auth.isManager()) {

Msgbox.alert(this, "Bạn không có quyền xoá người học này!");

} else {

String manh = txtMaNH.getText();

if (Msgbox.confirm(this, "Bạn thực sự muốn xoá người học này?")){

try {

dao.delete(manh);

this.fillTable();

this.clearForm();

Msgbox.alert(this, "Bạn xoá thành công!");

} catch (Exception e) {

Msgbox.alert(this, "Xoá thất bại!");

}

}

}

}

void search() {

fillTable();

clearForm();

row = -1;

updateStatus();

}

void first() {

this.row = 0;

this.edit();

}

void prev() {

if (this.row > 0) {

this.row--;

this.edit();

}

}

void next() {

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

this.row++;

this.edit();

}

}

void last() {

this.row = tblNguoiHoc.getRowCount() - 1;

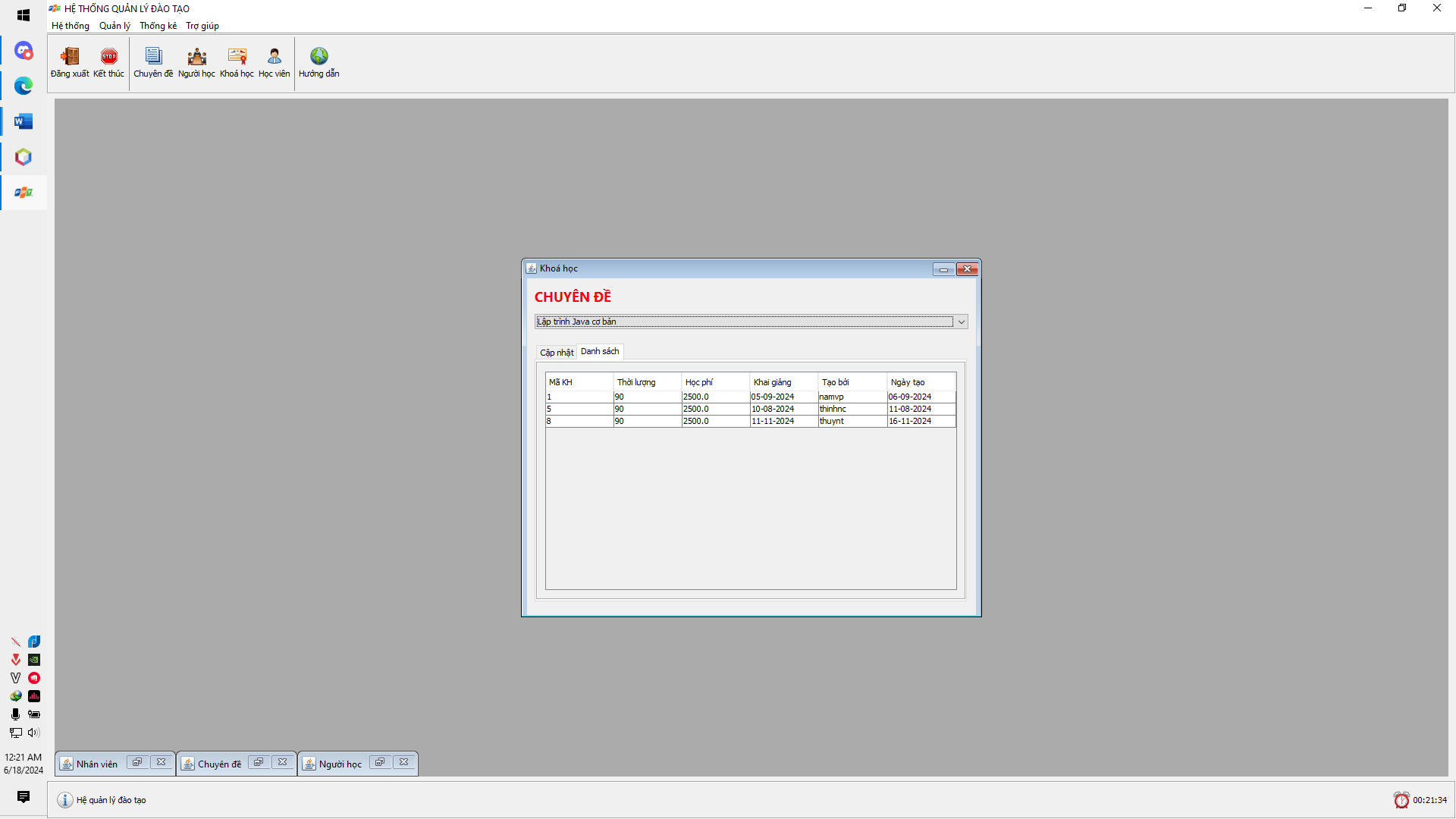
this.edit();

}

#### Cửa sổ quản lý khóa học (KhoaHocJInternalFrame)

**Giao diện**





public course() {

initComponents();

this.fillComboBoxChuyenDe();

this.fillTable();

this.updateStatus();

}

void fillComboBoxChuyenDe() {

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

model.removeAllElements();

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

for (ChuyenDe cd : list) {

model.addElement(cd);

}

}

void fillTable() {

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

model.setRowCount(0);

try {

ChuyenDe chuyenDe = (ChuyenDe) cboChuyenDe.getSelectedItem();

if (chuyenDe != null) {

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) {

e.printStackTrace();

}

}

void chonChuyenDe() {

ChuyenDe chuyenDe = (ChuyenDe) cboChuyenDe.getSelectedItem();

if (chuyenDe != null) {

txtThoiLuong.setText(String.valueOf(chuyenDe.getThoiLuong()));

txtHocPhi.setText(String.valueOf(chuyenDe.getHocPhi()));

//String moTa = chuyenDe.getMoTa();

String tenCD = chuyenDe.getTenCD();

// Các thao tác khác liên quan đến chuyenDe

} else {

// Xử lý trường hợp chuyenDe là null nếu cần

txtThoiLuong.setText("");

txtHocPhi.setText("");

// Hiển thị hoặc xử lý trường hợp chuyenDe là null

}

this.fillTable();

this.row = -1;

tabs.setSelectedIndex(1);

this.updateStatus();

}

void updateStatus() {

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

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

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

// Form state

btnThem.setEnabled(!edit);

btnSua.setEnabled(edit);

btnXoa.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) tblKhoaHoc.getValueAt(this.row, 0);

KhoaHoc kh = khDAO.selectById(makh);

this.setForm(kh);

tabs.setSelectedIndex(0);

this.updateStatus();

}

void clearForm() {

txtNguoiTao.setText("");

txtNgayKG.setText(""); // Đặt giá trị rỗng cho JTextField

txtHocPhi.setText("");

txtThoiLuong.setText("");

txtGhiChi.setText("");

txtNgayTao.setText(""); // Đặt giá trị rỗng cho JTextField

this.row = -1;

this.updateStatus();

}

void setForm(KhoaHoc kh) {

ChuyenDe chuyenDe = cdDAO.selectById(kh.getMaCD());

if (chuyenDe != null) {

cboChuyenDe.setSelectedItem(chuyenDe);

}

selectedCourseID = kh.getMaKH();

SimpleDateFormat dateFormat = new SimpleDateFormat("dd-MM-yyyy");

if (kh.getNgayKG() != null) {

String ngayKG = dateFormat.format(kh.getNgayKG());

txtNgayKG.setText(ngayKG);

} else {

txtNgayKG.setText("");

}

txtNguoiTao.setText(kh.getMaNV());

txtGhiChi.setText(kh.getGhiChu());

txtHocPhi.setText(String.valueOf(kh.getHocPhi()));

txtThoiLuong.setText(String.valueOf(kh.getThoiLuong()));

if (kh.getNgayTao() != null) {

String ngayTao = dateFormat.format(kh.getNgayTao());

txtNgayTao.setText(ngayTao);

} else {

txtNgayTao.setText("");

}

}

KhoaHoc getForm() {

KhoaHoc kh = new KhoaHoc();

ChuyenDe cd = (ChuyenDe) cboChuyenDe.getSelectedItem();

if (txtNguoiTao.getText().equals("")) {

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

} else {

kh.setMaNV(txtNguoiTao.getText());

}

kh.setMaCD(cd.getMaCD());

kh.setThoiLuong(Integer.parseInt(txtThoiLuong.getText()));

// Chuyển đổi chuỗi ngày tháng thành Date

SimpleDateFormat dateFormat = new SimpleDateFormat("dd-MM-yyyy");

try {

Date ngayKG = dateFormat.parse(txtNgayKG.getText());

kh.setNgayKG(ngayKG);

} catch (ParseException e) {

e.printStackTrace();

// Xử lý lỗi nếu cần thiết

}

kh.setHocPhi(Double.parseDouble(txtHocPhi.getText()));

kh.setGhiChu(txtGhiChi.getText());

kh.setMaKH(Integer.parseInt(cboChuyenDe.getToolTipText()));

try {

Date ngayTao = dateFormat.parse(txtNgayTao.getText());

kh.setNgayTao(ngayTao);

} catch (ParseException e) {

e.printStackTrace();

// Xử lý lỗi nếu cần thiết

}

return kh;

}

void insert() {

KhoaHoc nh = getForm();

try {

khDAO.insert(nh);

fillTable();

clearForm();

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

} catch (Exception e) {

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

}

}

void update() {

KhoaHoc nh = getForm();

try {

khDAO.update(nh);

fillTable();

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

} catch (Exception e) {

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

}

}

void delete() {

if (!Auth.isManager()) {

Msgbox.alert(this, "Bạn không có quyền xoá người học này!");

} else {

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

if (Msgbox.confirm(this, "Bạn thực sự muốn xoá người học này?")){

try {

khDAO.delete(id);

this.fillTable();

this.clearForm();

Msgbox.alert(this, "Bạn xoá thành công!");

} catch (Exception e) {

Msgbox.alert(this, "Xoá thất bại!");

}

}

}

}

void first() {

this.row = 0;

this.edit();

}

void prev() {

if (this.row > 0) {

this.row--;

this.edit();

}

}

void next() {

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

this.row++;

this.edit();

}

}

void last() {

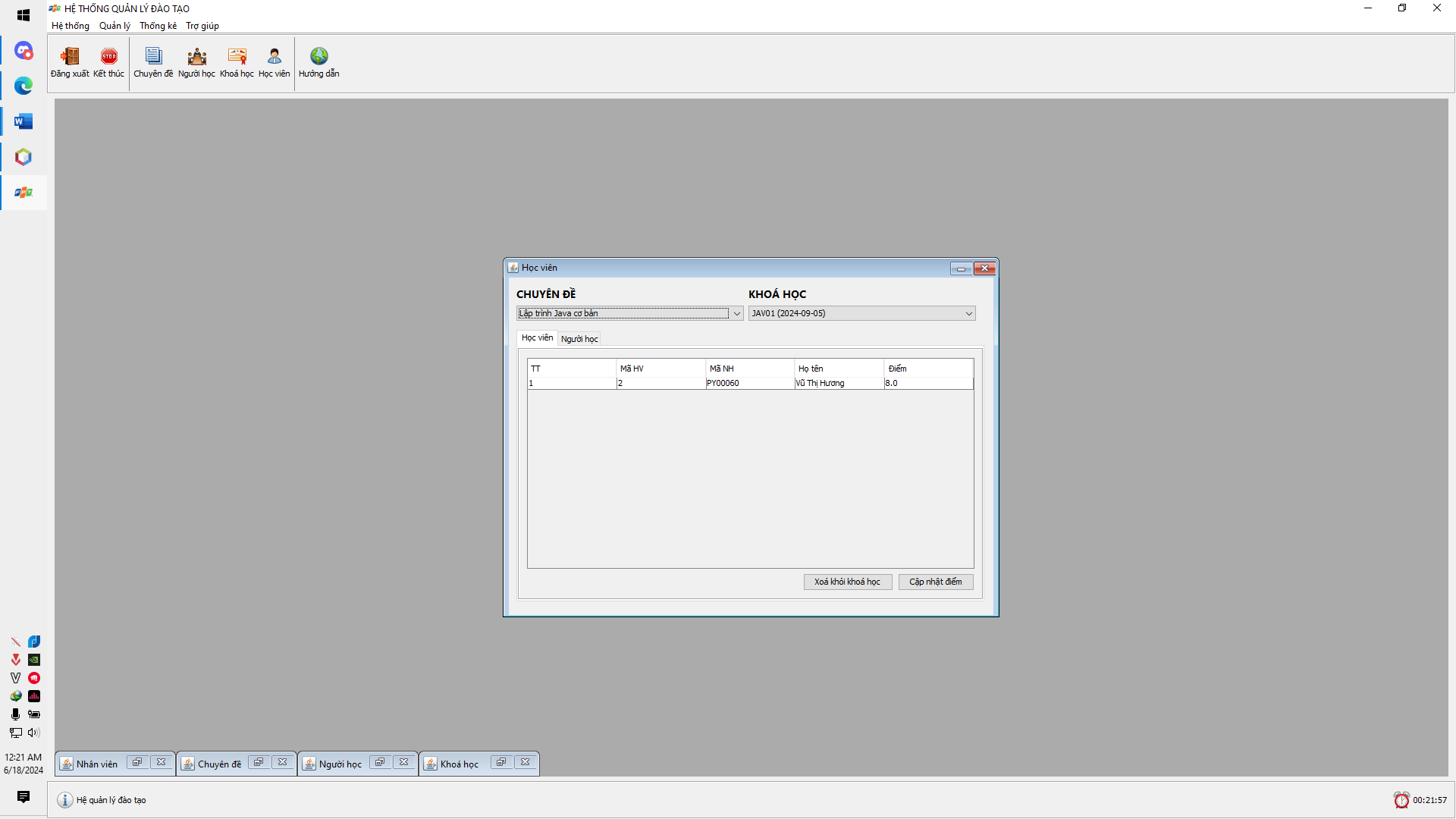
this.row = tblKhoaHoc.getRowCount() - 1;

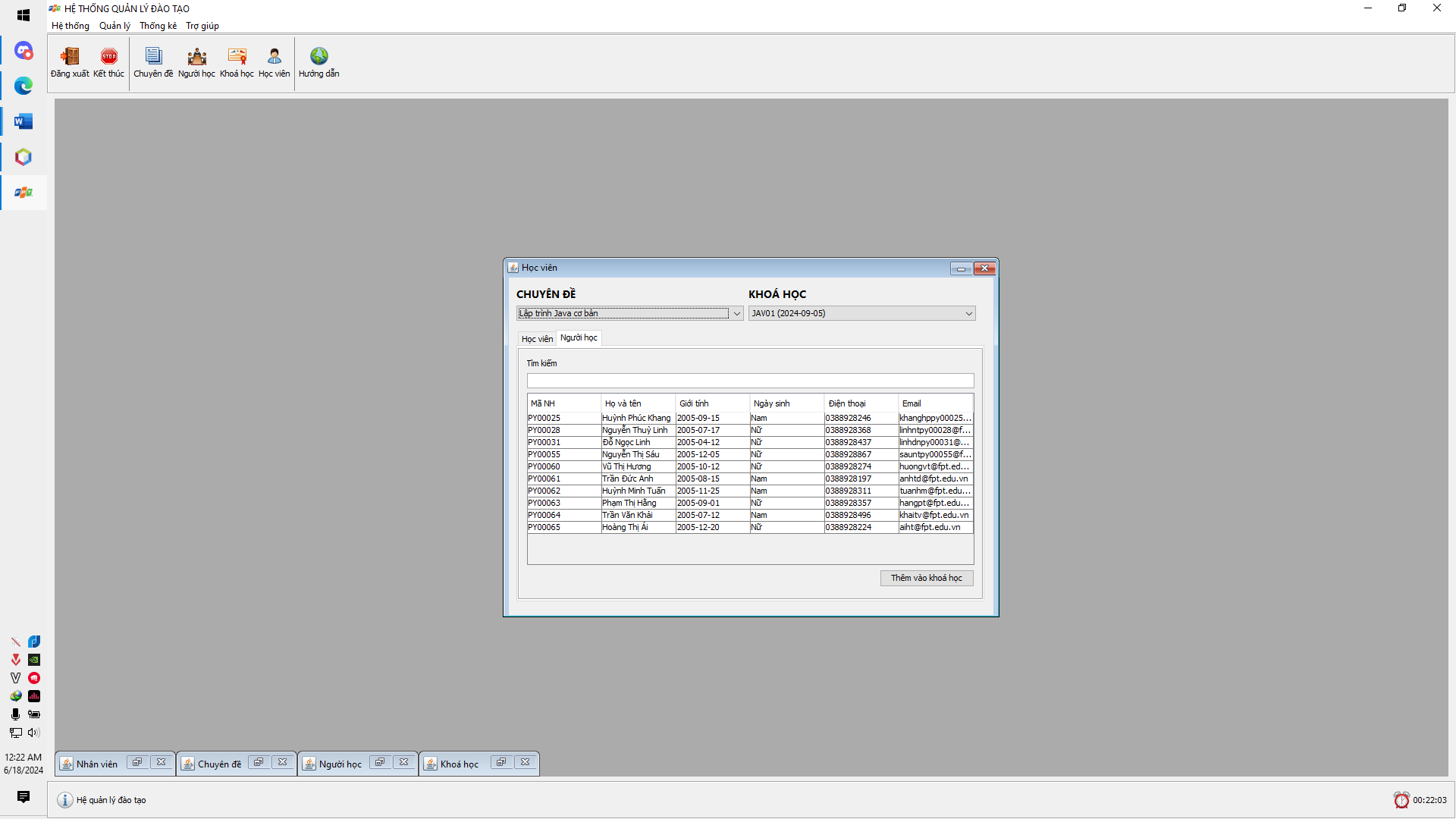
this.edit();

}

#### Cửa sổ quản lý học viên (HocVienJInternalFrame)

**Giao diện**





public Student() {

initComponents();

fillComboboxChuyenDe();

}

void fillComboboxChuyenDe(){

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

model.removeAllElements();

try{

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

for(ChuyenDe cd : list){

model.addElement(cd);

}

fillComboboxKhoaHoc();

}catch(Exception e){

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

}

}

void fillComboboxKhoaHoc() {

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

model.removeAllElements();

try {

ChuyenDe chuyenDe = (ChuyenDe) cboChuyenDe.getSelectedItem();

if (chuyenDe != null) {

List<KhoaHoc> list = khDAO.selectKhoaHocByChuyenDe(chuyenDe.getMaCD());

for (KhoaHoc khoaHoc : list) {

model.addElement(khoaHoc);

}

// Sau khi tải dữ liệu khoá học, cập nhật bảng người học

fillTableNguoiHoc();

}

} catch (Exception e) {

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

}

}

public void fillTableHocVien() {

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

model.setRowCount(0);

try {

KhoaHoc kh = (KhoaHoc) cboKhoaHoc.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();

Object[] row = {

i + 1, hv.getMaHV(),

hv.getMaNH(),

hoTen,

hv.getDiem()

};

model.addRow(row);

}

}

// Sau khi điền dữ liệu cho bảng học viên, cập nhật bảng người học

fillTableNguoiHoc();

} catch (Exception e) {

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

}

}

public void fillTableNguoiHoc(){

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

model.setRowCount(0);

try{

KhoaHoc kh = (KhoaHoc) cboKhoaHoc.getSelectedItem();

if(kh != null){

System.out.println("MaKH:" + kh.getMaKH());

String keyword = txtTimKiem.getText();

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

for(NguoiHoc nh : list){

Object[] row = {

nh.getMaNH(),

nh.getHoTen(),

nh.getNgaySinh(),

nh.isGioiTinh() ? "Nam" : "Nữ",

nh.getDienThoai(),

nh.getEmail()

};

model.addRow(row);

}

}

}catch(Exception e){

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

}

}

void addHocVien(){

KhoaHoc khoaHoc = (KhoaHoc)cboKhoaHoc.getSelectedItem();

for(int row : tblNguoiHoc.getSelectedRows()){

HocVien hv = new HocVien();

hv.setMaKH(khoaHoc.getMaKH());

hv.setDiem(0);

hv.setMaNH((String)tblNguoiHoc.getValueAt(row, 0));

System.out.println("=>"+hv.getMaKH()+"-"+hv.getMaNH()+"-"+hv.getDiem());

hvDAO.insert(hv);

}

fillTableHocVien();

tabs.setSelectedIndex(0);

}

void removeHocVien(){

if(!Auth.isManager()){

Msgbox.alert(this, "Bạn không đủ quyền để xoá học viên");

}else{

if(Msgbox.confirm(this, "Bạn muốn xoá các học viên được chọn")){

for(int row:tblHocVien.getSelectedRows()){

int maHV = (Integer)tblHocVien.getValueAt(row, 1);

hvDAO.delete(maHV);

}

fillTableHocVien();

}

}

}

void updateDiem(){

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

int maHV = (Integer)tblHocVien.getValueAt(i, 1);

HocVien hocVien = hvDAO.selectById(maHV);

hocVien.setDiem(Double.parseDouble(tblHocVien.getValueAt(i, 4).toString()));

hvDAO.update(hocVien);

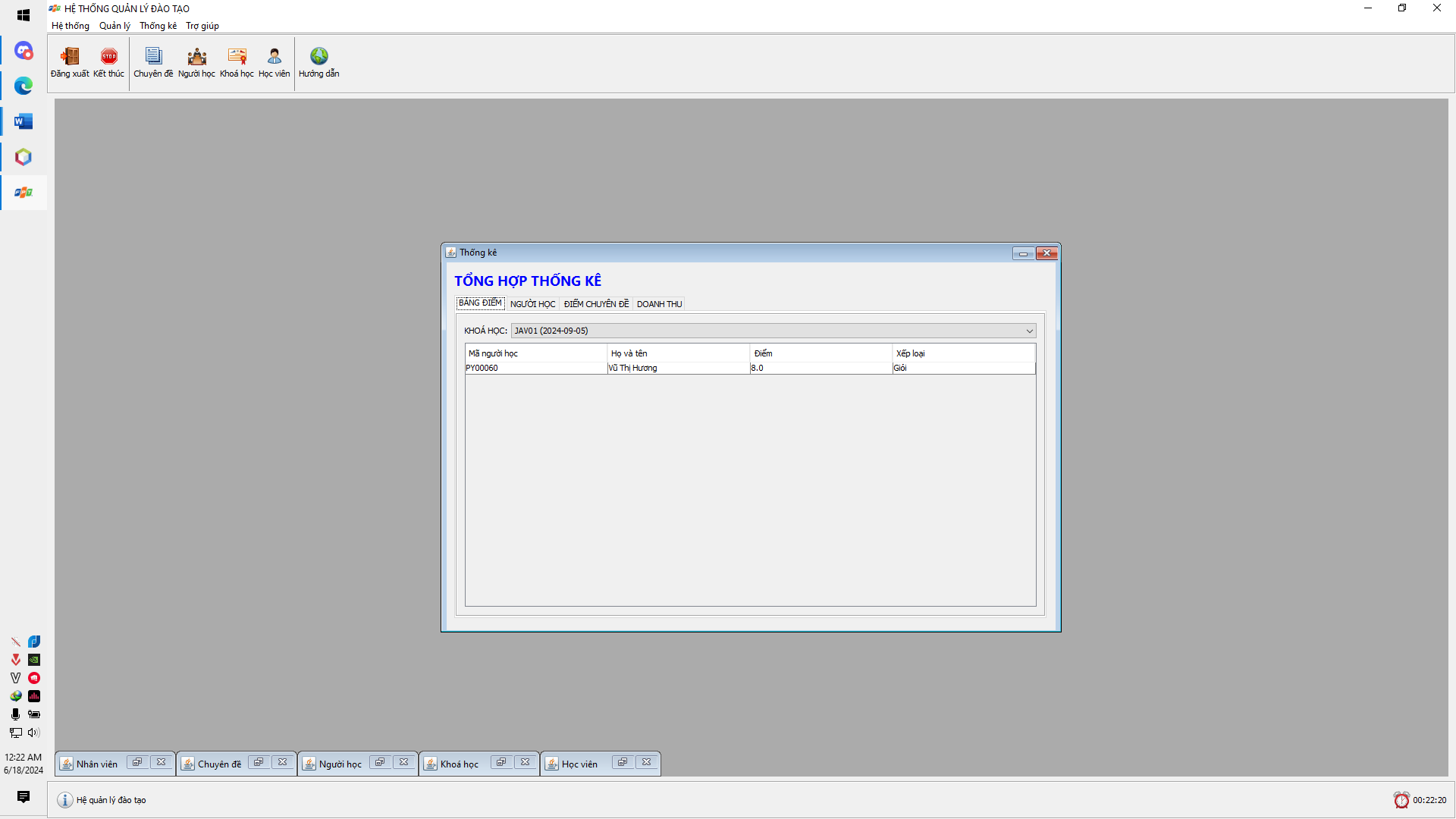
}

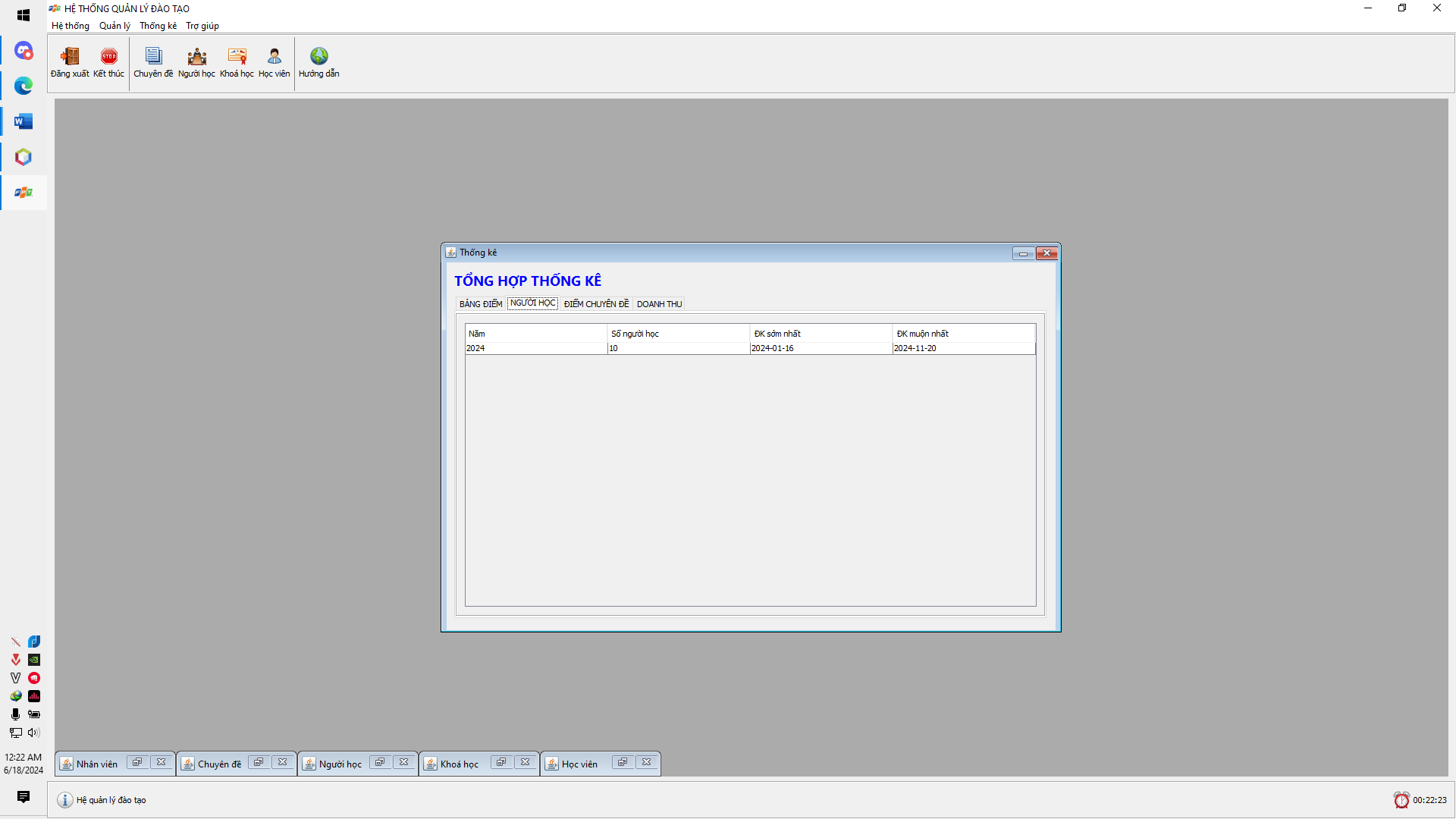
Msgbox.alert(this, "Cập nhật điểm thành công");

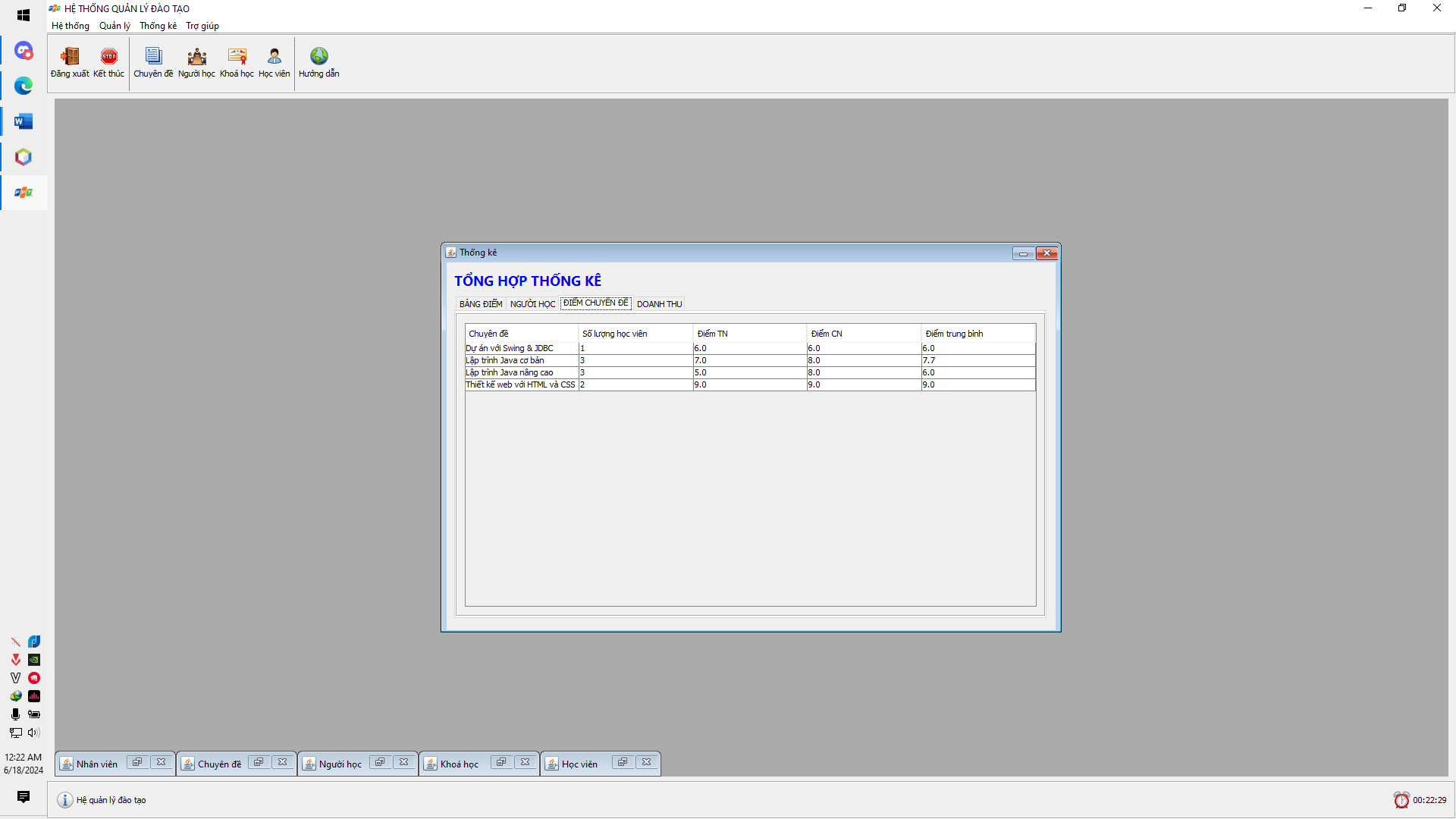
}

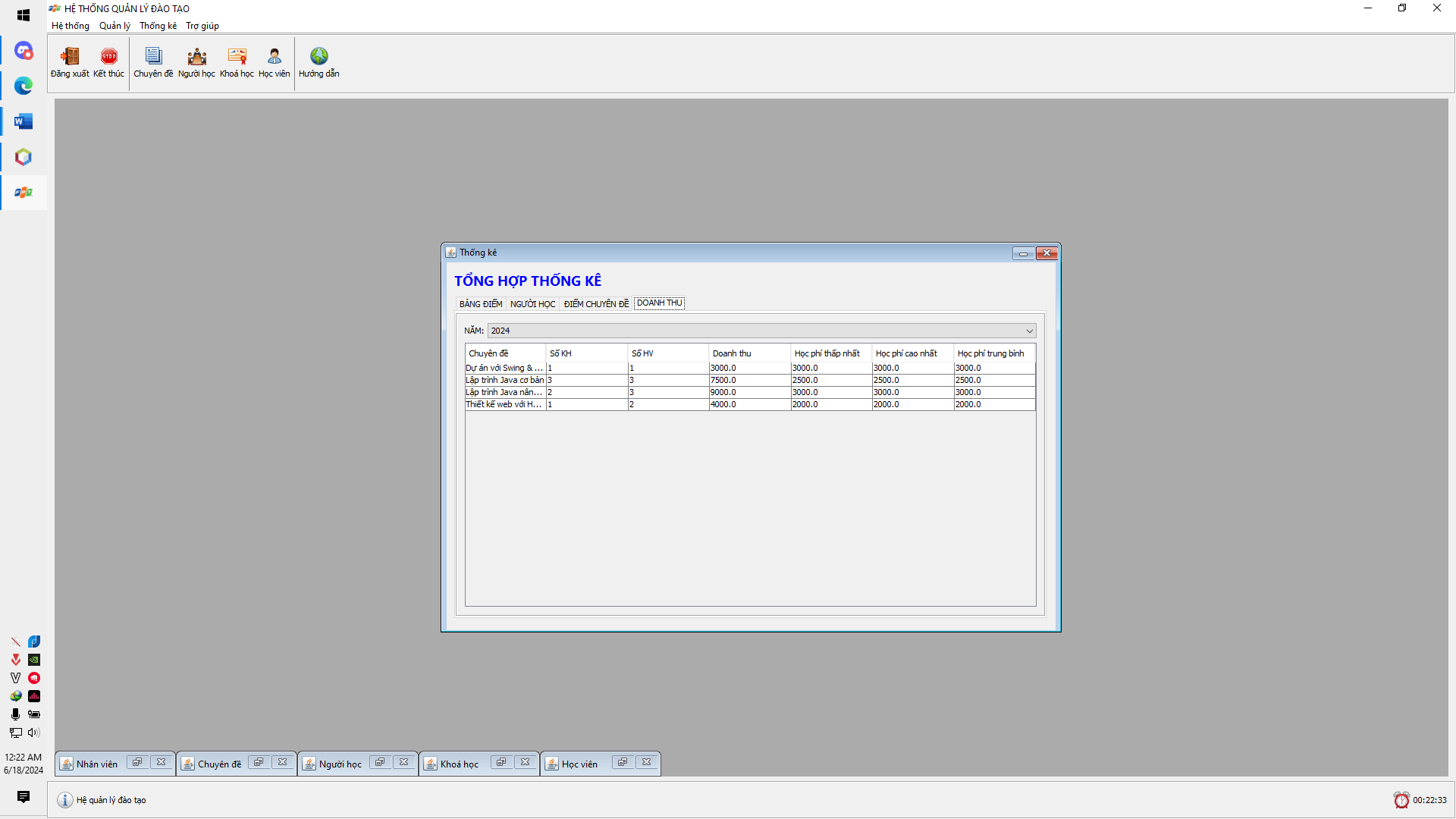
### Cửa sổ tổng hợp thống kê (ThongKeJInternalFrame)

**Giao diện**









public Statistical() {

initComponents();

setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

initComponents();

init();

}

void init() {

fillComboKhoahoc();

fillTableBangdiem();

fillTableNguoiHoc();

fillTableDiemchuyende();

fillComboNam();

fillTableDoanhthu();

this.selectTab(0);

if(!Auth.isManager()) {

tabs.remove(3);

}

}

private void fillComboKhoahoc() {

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

model.removeAllElements();

List<KhoaHoc> list = khdao.selectAll();

if (!list.isEmpty()) { // Kiểm tra danh sách có rỗng hay không

for (KhoaHoc kh : list) {

model.addElement(kh);

}

}

}

private void fillTableBangdiem() {

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

model.setRowCount(0);

KhoaHoc kh = (KhoaHoc) cboKhoaHoc.getSelectedItem();

if (kh != null) { // Kiểm tra khác null trước khi sử dụng

List<Object[]> list = dao.getBangDiem(kh.getMaKH());

for (Object[] row : list) {

float diem = Float.valueOf(row[2].toString());

model.addRow(new Object[]{

row[0], row[1], diem, getXeploai(diem)

});

}

}

}

private void fillTableNguoiHoc() {

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

model.setRowCount(0);

List<Object[]> list = dao.getLuongNguoiHoc();

for (Object[] row : list) {

model.addRow(row);

}

}

private void fillTableDiemchuyende() {

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

model.setRowCount(0);

List<Object[]> list = dao.getDiemChuyenDe();

for (Object[] row : list) {

model.addRow(new Object[]{

row[0],row[1],row[2],row[3],row[4]

});

}

}

private void fillComboNam() {

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

model.removeAllElements();

List<Integer> list = khdao.selectYears();

for (Integer year : list) {

model.addElement(year);

}

}

private void fillTableDoanhthu() {

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

model.setRowCount(0);

Integer selectedNam = (Integer) cboNam.getSelectedItem();

if (selectedNam != null) {

int nam = selectedNam.intValue();

List<Object[]> list = dao.getDoanhThu(nam);

for (Object[] row : list) {

model.addRow(new Object[]{

row[0], // Thay thế bằng tên cột 1

row[1], // Thay thế bằng tên cột 2

row[2], // Thay thế bằng tên cột 3

// và tiếp tục cho các cột còn lại nếu có

});

}

}

}

public void selectTab(int index) {

tabs.setSelectedIndex(index);

}

private String getXeploai(float diem) {

if(diem<5) {

return "Chưa đạt";

}

if(diem<6.5) {

return "Trung bình";

}

if(diem<7.5) {

return "Khá";

}

if(diem<9) {

return "Giỏi";

}

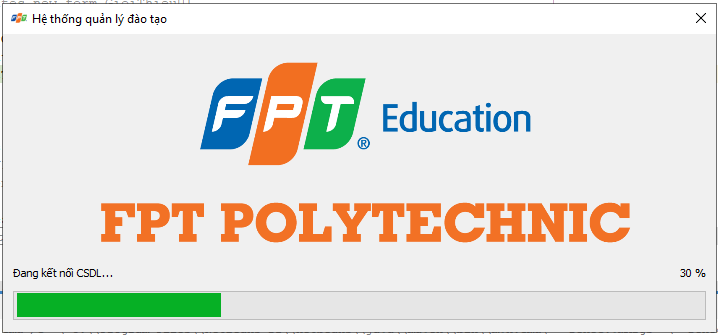
return "Xuất sắc";

}

### Các giao diện hỗ trợ khác

#### Cửa sổ chào (ChaoJInternalFrame)

**Giao diện**



void init(){

setIconImage(XImage.getAppIcon());

setLocationRelativeTo(null);

setTitle("Hệ thống quản lý đào tạo");

}

public void Loading(int value, String message){

pgbLoading.setValue(value);

lblLoading.setText(message);

lblValue.setText(String.valueOf(value)+" %");

try {

Thread.sleep(value\*100);

} catch (InterruptedException ex) {

}

}

private void formWindowOpened(java.awt.event.WindowEvent evt) {

// TODO add your handling code here:

Thread taiTienTrinh = new Thread(){

public void run(){

Loading(10, "Đang khởi động...");

Loading(30, "Đang kết nối CSDL...");

Loading(50, "Đang tải...");

Loading(100, "Hoàn thành");

dispose();

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new HinhChinh().setVisible(true);

}

});

}

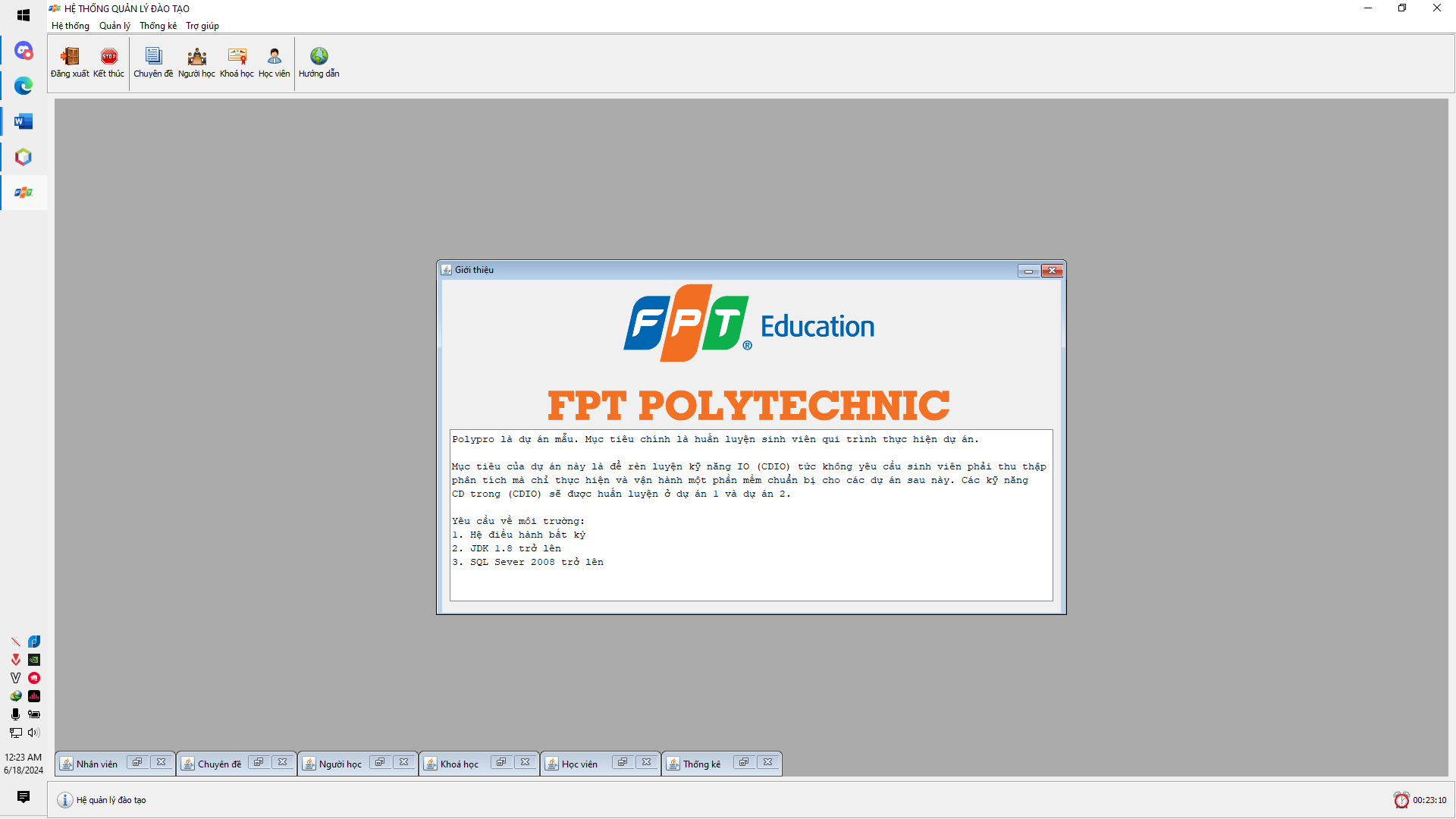
};

taiTienTrinh.start();

}

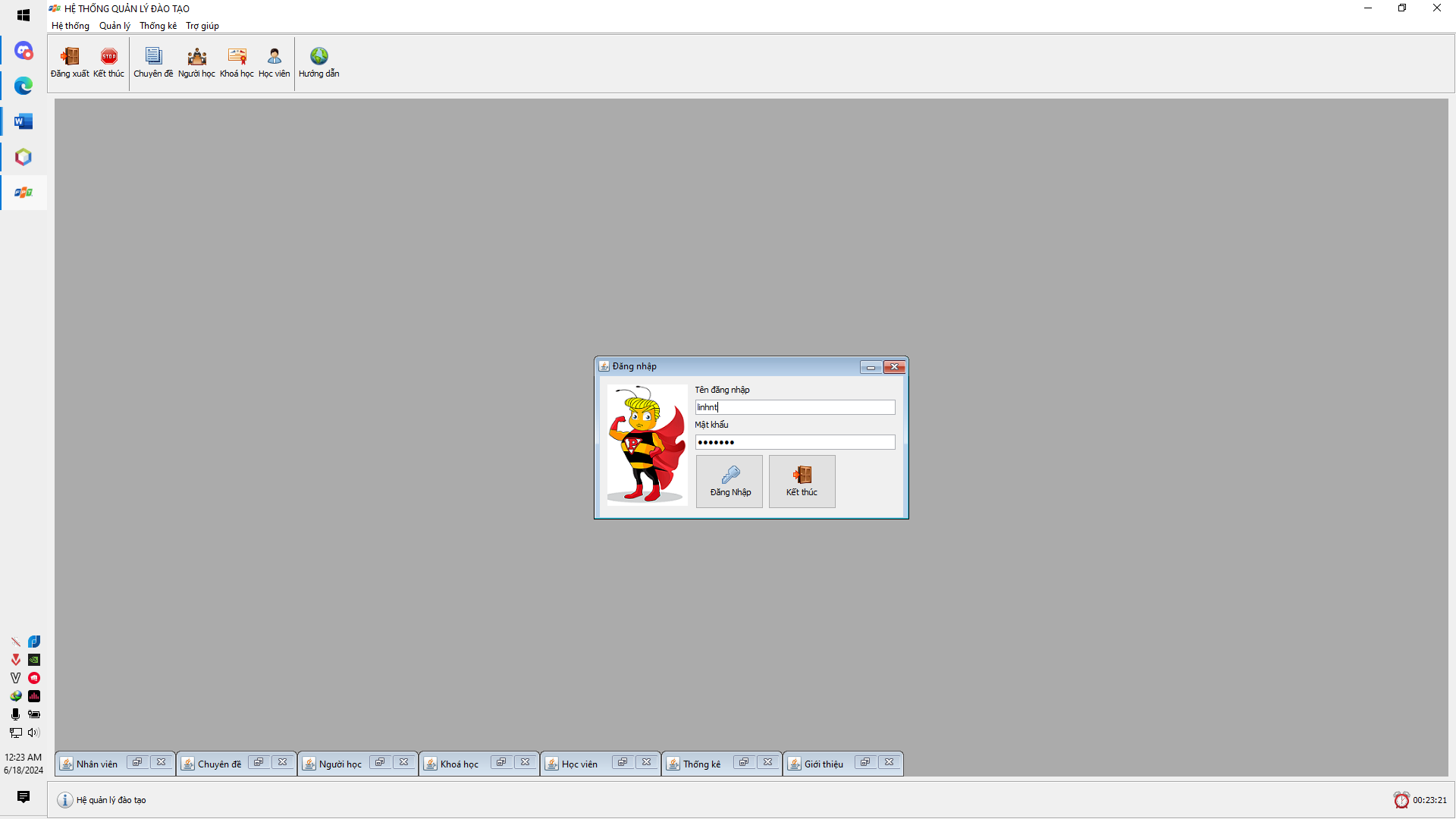
#### Cửa sổ giới thiệu (GioiThieuJInternalFrame)

**Giao diện**



#### Cửa sổ đăng nhập (DangNhapJInternalFrame)

**Giao diện**



public DangNhap() {

initComponents();

}

void dangNhap(){

String MaNV = txtMaNV.getText();

String password = new String(txtMatKhau.getPassword());

NhanVien nv = dao.selectById(MaNV);

if(nv == null){

Msgbox.alert(this, "Sai đăng nhập");

}else{

if(!nv.getMatKhau().equals(password)){

Msgbox.alert(this, "sai mật khẩu");

}else{

Auth.user = nv;

this.dispose();

}

}

}

void ketThuc(){

if(Msgbox.confirm(this, "Bạn muốn kết thúc ứng dụng?")){

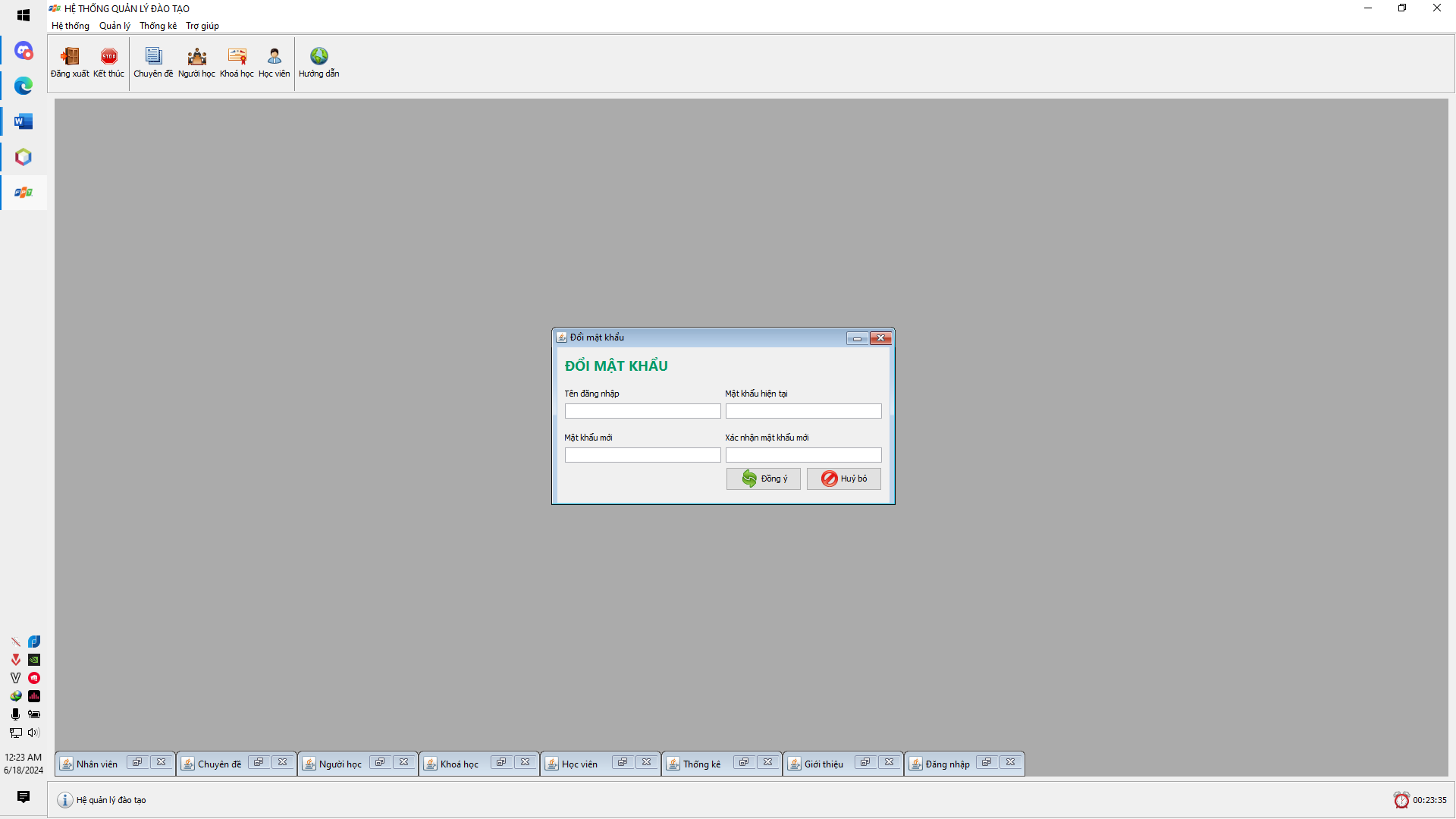
System.exit(0);

}

}

#### Cửa sổ đổi mật khẩu (DoiMatKhauJInternalFrame)

**Giao diện**



public DoiMatKhau() {

initComponents();

}

void doiMatKhau(){

String manv = txtMaNV.getText();

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

String matKhauMoi1 = new String(txtMatKhau1.getPassword());

String matKhauMoi2 = new String(txtMatKhau2.getPassword());

if(!manv.equalsIgnoreCase(Auth.user.getMaNV())){

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

}else if(!matKhau.equals(Auth.user.getMatKhau())){

Msgbox.alert(this, "Sai mật khẩu");

}else if(!matKhauMoi1.equals(matKhauMoi2)){

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

}else{

Auth.user.setMatKhau(matKhauMoi1);

dao.update(Auth.user);

Msgbox.alert(this, "Đổi mật khẩu thành công!");

}

}

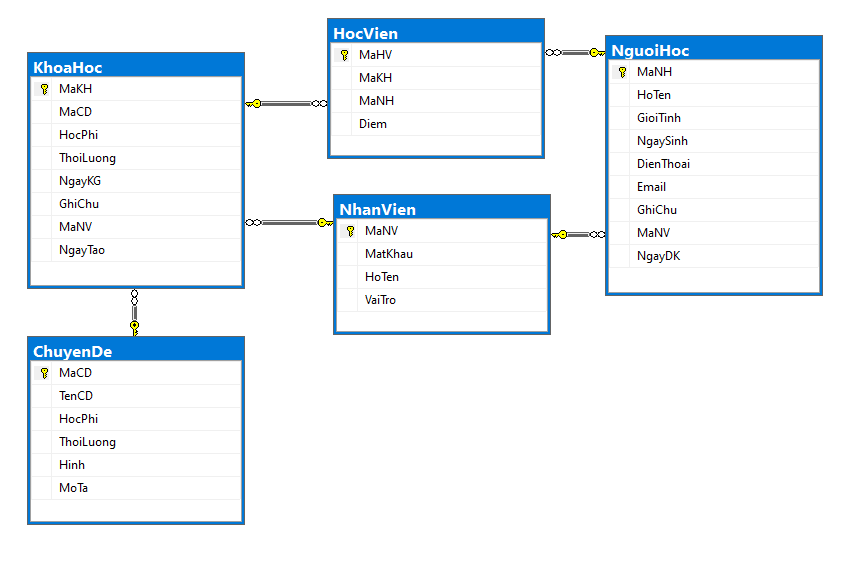
void huyBo(){

this.dispose();

};

## Tạo CSDL với SQL Server

### Sơ đồ quan hệ



### Chi tiết các bảng

#### Bảng NhanVien

**Cấu trúc bảng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| MaNV | NVARCHAR(20) | PK, NOT NULL | Mã nhân viên |
| MatKhau | NVARCHAR(50) | NOT NULL | Mật khẩu |
| HoTen | NVARCHAR(50) | NOT NULL | Họ và tên |
| VaiTro | BIT | DEFAULT 0 | 1 là trưởng phòng, 0 là nhân viên |

**Mã lệnh tạo bảng**

|  |
| --- |
| CREATE TABLE NhanVien  (  MaNV NVARCHAR(50) NOT NULL,  MatKhau NVARCHAR(50) NOT NULL,  HoTen NVARCHAR(50) NOT NULL,  VaiTro BIT NOT NULL,  PRIMARY KEY(MaNV)  ) |

**Câu lệnh SQL cơ bản**

|  |
| --- |
| INSERT INTO NhanVien (MaNV, MatKhau, HoTen, VaiTro) VALUES (?, ?, ?, ?)  UPDATE NhanVien SET MatKhau = ?, HoTen = ?, VaiTro = ? WHERE MaNV = ?  DELETE FROM NhanVien WHERE MaNV = ?  SELECT \* FROM NhanVien WHERE MaNV = ? |

#### Bảng ChuyenDe

**Cấu trúc bảng**

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

**Mã lệnh tạo bảng**

|  |
| --- |
| CREATE TABLE ChuyenDe (  MaCD NCHAR(5) PRIMARY KEY NOT NULL,  TenCD NVARCHAR(50) NOT NULL,  HocPhi FLOAT NOT NULL,  ThoiLuong INT NOT NULL,  Hinh NVARCHAR(50) NOT NULL,  MoTa NVARCHAR(255) NOT NULL  ); |

#### Bảng NguoiHoc

**Cấu trúc bảng**

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

**Mã lệnh tạo bảng**

|  |
| --- |
| CREATE TABLE NguoiHoc (  MaNH NCHAR(7) PRIMARY KEY NOT NULL,  HoTen NVARCHAR(50) NOT NULL,  GioiTinh BIT DEFAULT 1,  NgaySinh DATE NOT NULL,  DienThoai NVARCHAR(24) NOT NULL,  Email NVARCHAR(50) NOT NULL,  GhiChu NVARCHAR(255),  MaNV NVARCHAR(20) NOT NULL,  NgayDK DATE DEFAULT getdate()  ); |

#### Bảng KhoaHoc

**Cấu trúc bảng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaKH | INT | PK, Tự tăng | Mã khóa học |
| MaCD | NCHAR(5) | FK, NOT NULL | Mã chuyên đề |
| HocPhi | FLOAT | NOT NULL | Học phí |
| ThoiLuong | INT | NOT NULL | Thời lượng, tính bằng giờ |
| NgayKG | DATE | NOT NULL | Ngày khai giảng |
| GhiChu | NVARCHAR(255) | NULL | Ghi chú về khóa học |
| MaNV | NVARCHAR(20) | FK, NOT NULL | Mã nhân viên nhập |
| NgayTao | DATE | DEFAULT getdate() | Ngày đăng ký của người học |

**Mã lệnh tạo bảng**

|  |
| --- |
| CREATE TABLE KhoaHoc (  MaKH INT PRIMARY KEY IDENTITY,  MaCD NCHAR(5) NOT NULL,  HocPhi FLOAT NOT NULL,  ThoiLuong INT NOT NULL,  NgayKG DATE NOT NULL,  GhiChu NVARCHAR(255),  MaNV NVARCHAR(20) NOT NULL,  NgayTao DATE DEFAULT getdate(),  FOREIGN KEY (MaCD) REFERENCES ChuyenDe(MaCD),  FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)  ); |

#### Bảng HocVien

**Cấu trúc bảng**

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

**Mã lệnh tạo bảng**

|  |
| --- |
| CREATE TABLE HocVien (  MaHV INT PRIMARY KEY IDENTITY,  MaKH INT NOT NULL,  MaNH NCHAR(7) NOT NULL,  Diem FLOAT DEFAULT -1,  FOREIGN KEY (MaKH) REFERENCES KhoaHoc(MaKH),  FOREIGN KEY (MaNH) REFERENCES NguoiHoc(MaNH)  ); |

### Thủ tục lưu

* *Mỗi thủ tục lưu cần làm rõ các vấn đề sau đây:*
  + *Mô tả, mục đích sử dụng*
  + *Tham số vào*
  + *Kết quả ra*

#### Sp\_BangDiem()

|  |  |
| --- | --- |
| **Mã nguồn** | 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 |
| **Mô tả** | Proc này được sử dụng để truy vấn bảng điểm của khóa học |
| **Tham số** | @MaKH là mã khóa học cần truy vấn bảng điểm |
| **Kết quả** | [MaNH, TenNH, Diem] |

#### Sp\_DoanhThu()

|  |  |
| --- | --- |
| **Mã nguồn** | CREATE PROC sp\_DoanhThu(@Year INT)  AS BEGIN  SELECT  TenCD ChuyenDe,  COUNT(DISTINCT kh.MaKH) SoKH,  COUNT(hv.MaHV) SoHV,  SUM(kh.HocPhi) DoanhThu,  MIN(kh.HocPhi) ThapNhat,  MAX(kh.HocPhi) CaoNhat,  AVG(kh.HocPhi) TrungBinh  FROM KhoaHoc kh  JOIN HocVien hv ON kh.MaKH=hv.MaKH  JOIN ChuyenDe cd ON cd.MaCD = kh.MaCD  WHERE YEAR(NgayKG) = @Year  GROUP BY TenCD  END |
| **Mô tả** | Proc này được sử dụng để truy vấn doanh thu theo năm |
| **Tham số** | @Year là năm cần truy vấn |
| **Kết quả** | [ChuyenDe, SoKH, DoanhThu, ThapNhat, CaoNhat, TrungBinh] |

#### Sp\_LuongNguoiHoc()

|  |  |
| --- | --- |
| **Mã nguồn** | CREATE PROC sp\_LuongNguoiHoc  AS BEGIN  SELECT  YEAR(NgayDK) Nam,  COUNT(\*) SoLuong,  MIN(NgayDK) DauTien,  MAX(NgayDK) CuoiCung  FROM NguoiHoc  GROUP BY YEAR(NgayDK)  END |
| **Mô tả** | Proc này được sử dụng để truy vấn số lượng người học trong năm |
| **Tham số** |  |
| **Kết quả** | [Nam, SoLuong, DauTien, CuoiCung] |

#### Sp\_DiemChuyenDe()

|  |  |
| --- | --- |
| **Mã nguồn** | CREATE PROC sp\_DiemChuyenDe  AS BEGIN  SELECT  TenCD ChuyenDe,  COUNT(MaHV) SoHV,  MIN(Diem) ThapNhat,  MAX(Diem) CaoNhat,  AVG(Diem) TrungBinh  FROM KhoaHoc kh  JOIN HocVien hv ON kh.MaKH = hv.MaKH  JOIN ChuyenDe cd ON cd.MaCD = kh.MaCD  GROUP BY TenCD  END |
| **Mô tả** | Proc này được sử dụng để truy vấn điểm chuyên đề |
| **Tham số** |  |
| **Kết quả** | [ChuyenDe, SoHV, ThapNhat, CaoNhat, TrungBinh] |

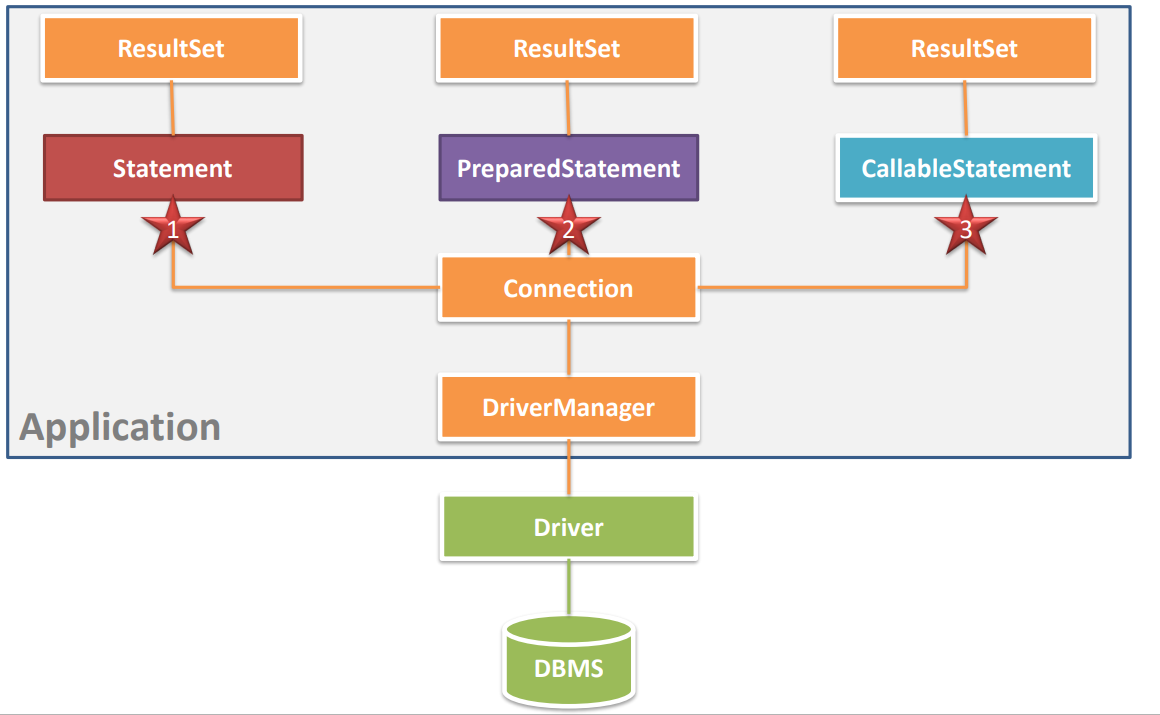
## Lập trình CSDL

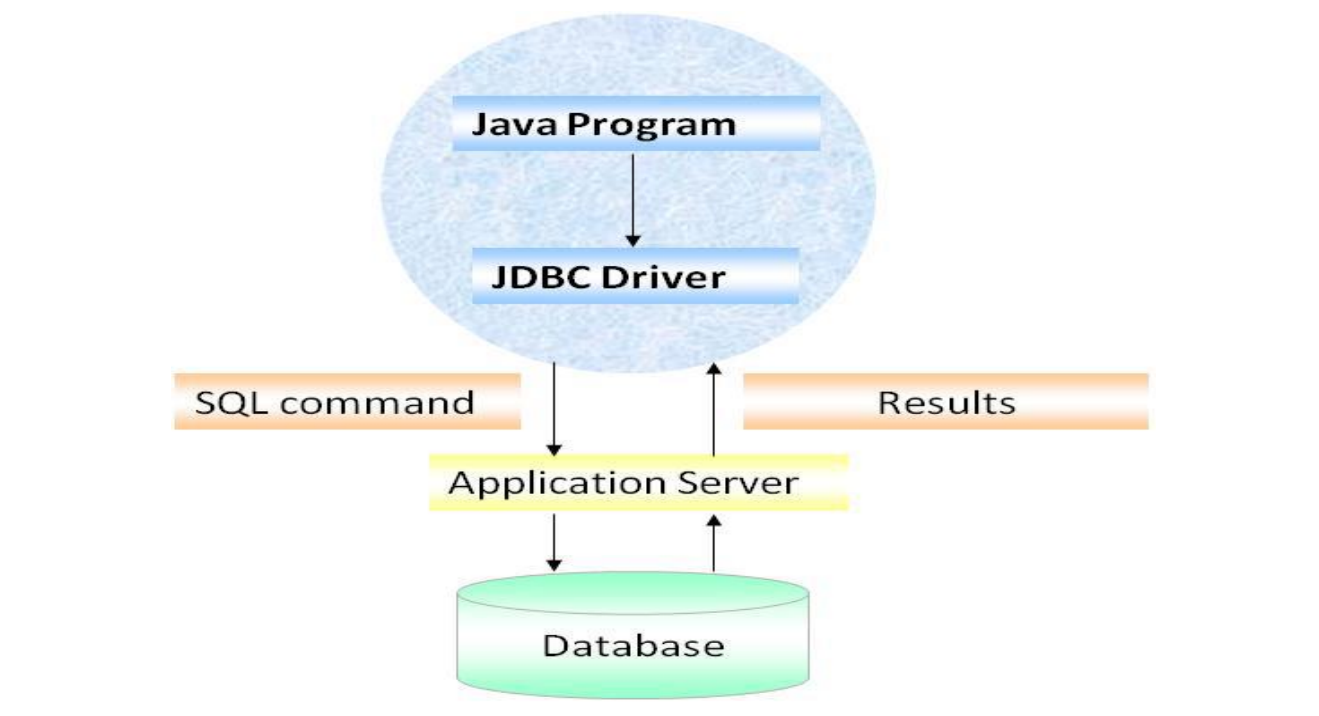
### Mô hình tổ chức lập trình CSDL

#### CSDL

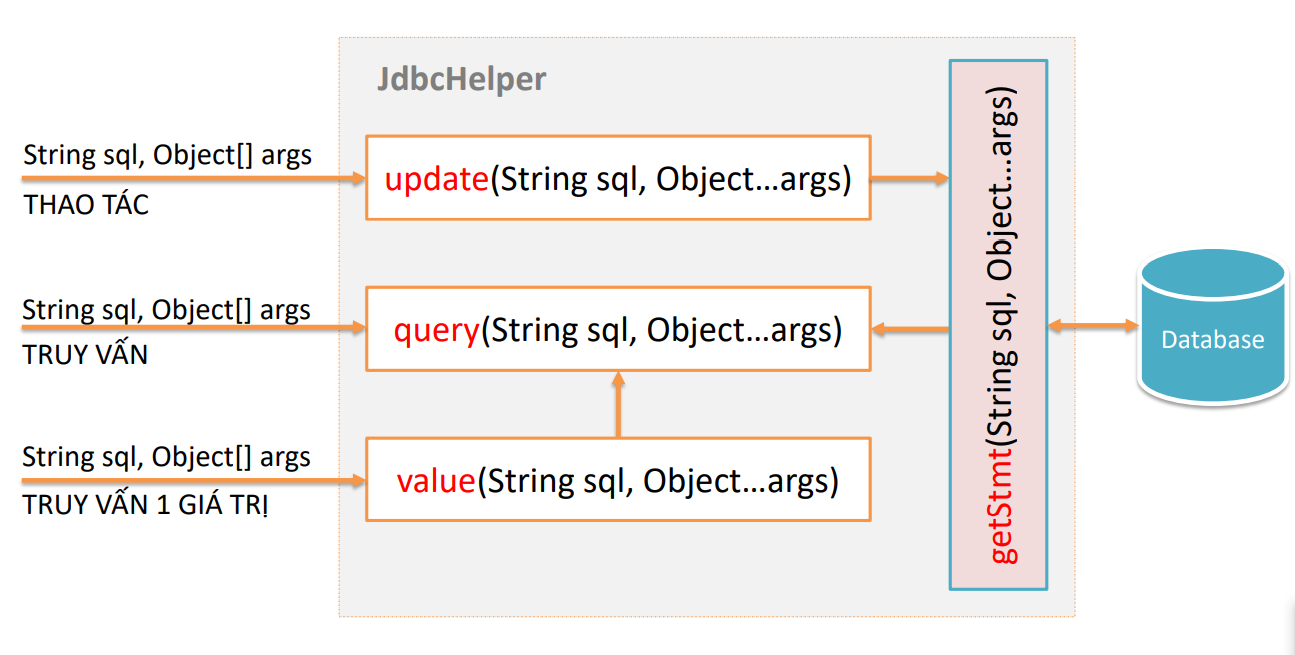
* Table
  + Chuyên Đề
  + Khóa Học
  + Người Học
  + Học Viên
  + Nhân Viên
* Stored Procedure
  + Sp\_BangDiem(INT MaKH)
  + Sp\_DiemChuyenDe()
  + Sp\_DoanhThu(INT year)
  + Sp\_LuongNguoiHoc()

#### Mô hình ứng dụng JDBC





#### Mô hình JdbcHelper



### JdbcHelper (XJdbc)

**Mô tả:**

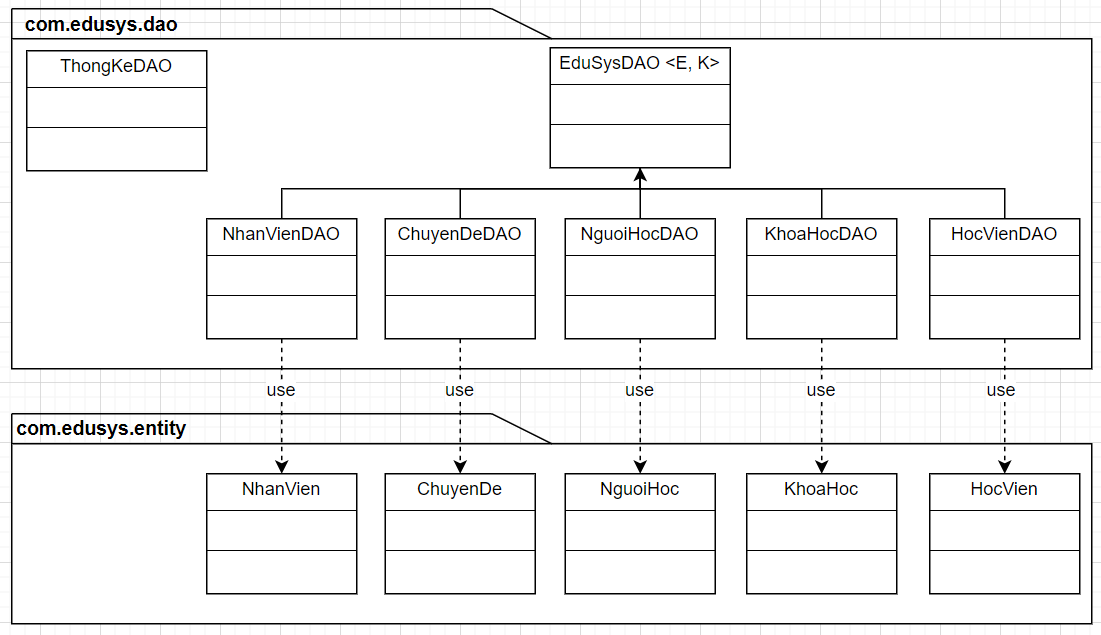
XJdbc là lớp tiện ích hỗ trợ lập trình JDBC. Lớp này gồm 3 phương thức được mô tả trong phần hướng dẫn sử dụng sau đây.

**Hướng dẫn sử dụng:**

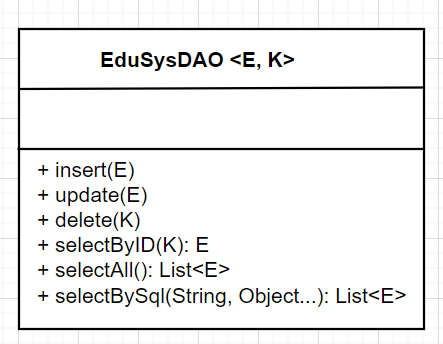
* XJdbc.**query**(String sql, Object…args): ResultSet
  + Truy vấn dữ liệu từ CSDL
  + Tham số sql: câu lệnh SELECT truy vấn có chứa ?
  + Tham số: args là các giá trị cung cấp cho các dấu ? của SQL
  + Kết quả: ResultSet
* XJdbc.**update**(String sql, Object…args): int
  + Thao tác CSDL (INSERT, UPDATE, DELETE)
  + Tham số sql: câu lệnh INSERT, UPDATE, DELETE có chứa ?
  + Tham số: args là các giá trị cung cấp cho các dấu ? của SQL
  + Kết quả: số bản ghi có ảnh hưởng
* XJdbc.**value**(String sql, Object…args): Object
  + Truy vấn một giá trị từ CSDL
  + Tham số sql: câu lệnh SELECT truy vấn có chứa ?
  + Tham số: args là các giá trị cung cấp cho các dấu ? của SQL
  + Kết quả: đối tượng chứa giá trị truy vấn

### Entity Class và DAO

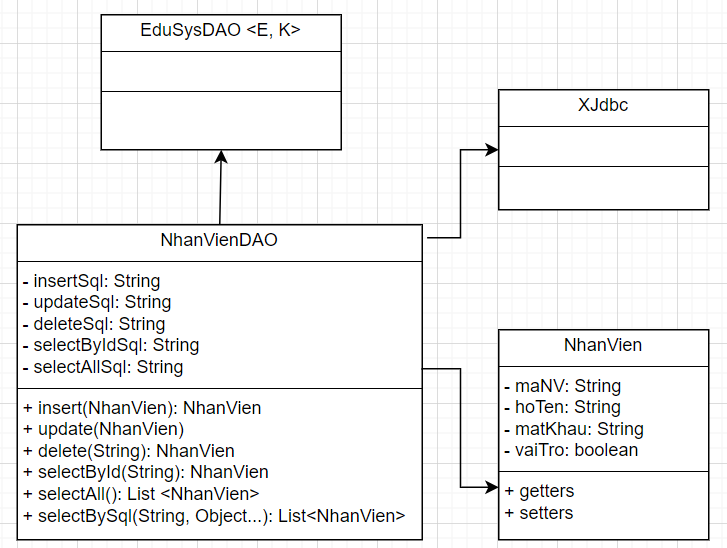
#### Class Diagram



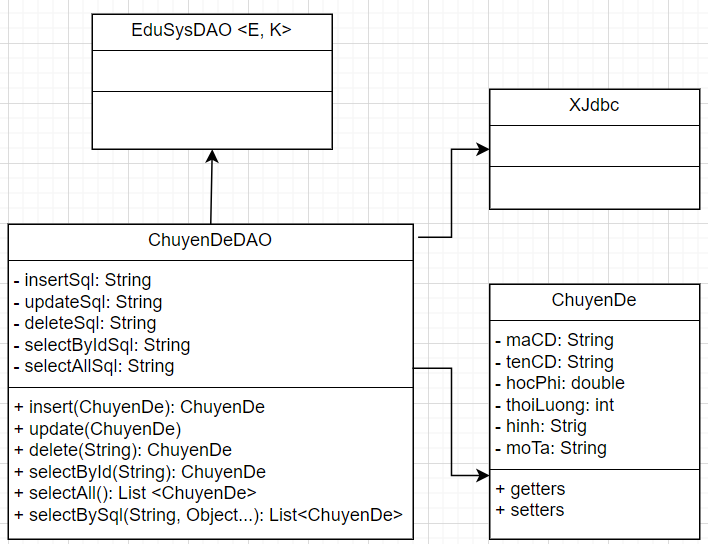
#### EduSysDAO



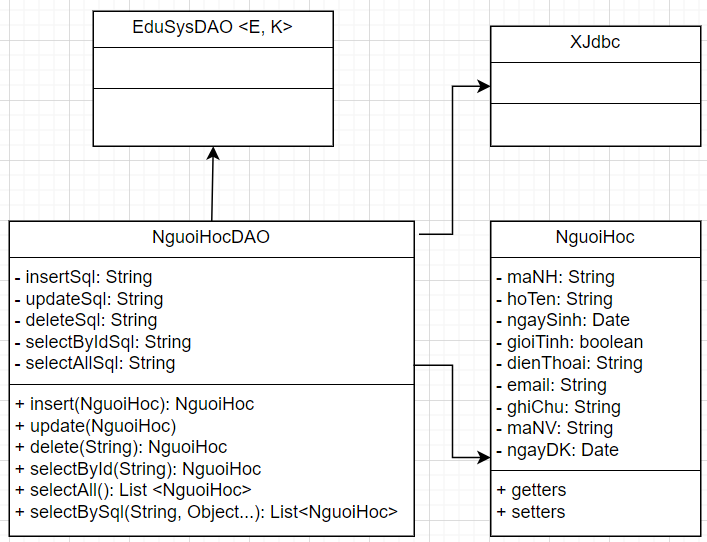
#### NhanVien và NhanVienDAO



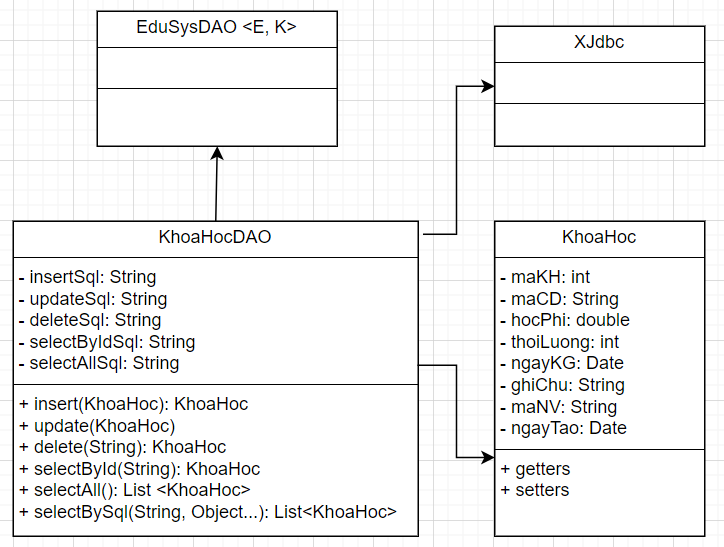
#### ChuyenDe & ChuyenDeDAO



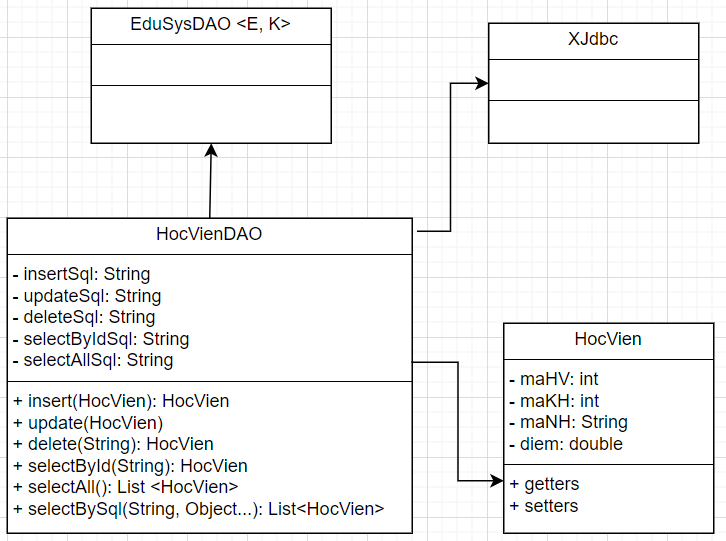
#### NguoiHoc & NguoiHocDAO



#### KhoaHoc & KhoaHocDAO



#### HocVien & HocVienDAO



#### ThongKeDAO

….

## Thư viện tiện ích

### XImage

**Mô tả:**

XImage là lớp tiện ích hỗ trợ cho việc làm việc với hình ảnh trong ứng dụng Java. Lớp này gồm 3 phương thức được mô tả trong phần hướng dẫn sử dụng sau đây.

**Hướng dẫn sử dụng:**

* XImage.**getAppIcon**(): Image
  + Trả về biểu tượng của ứng dụng.
  + Kết quả: Image
* XImage.**save**(File src):
  + Lưu trữ một tệp hình ảnh vào thư mục “logos”
  + Tham số: src là địa chỉ được cung cấp vào một thư mục được chỉ định ("logos")
* XImage.**read**(String fileName): ImageIcon
  + Đọc một tệp hình ảnh từ thư mục "logos"
  + Tham số: fileName là tên tệp đã được cung cấp
  + Kết quả: ImageIcon

### XDate

**Mô tả:**

XDate là lớp tiện ích được thiết kế để xử lý các thao tác liên quan đến ngày tháng. Lớp này gồm 3 phương thức được mô tả trong phần hướng dẫn sử dụng sau đây.

**Hướng dẫn sử dụng:**

* XDate.**toDate**(String date, String pattern): Date
  + Chuyển đổi một chuỗi đại diện cho ngày tháng thành đối tượng “Date”.
  + Tham số: date là chuỗi ngày tháng cần phân tích.
  + Tham số: pattern là mẫu định dạng ngày tháng.
  + Kết quả: Date
* XDate.**toString**(Date date, String pattern): String
  + Chuyển đổi một đối tượng “Date” thành chuỗi đại diện dựa trên mẫu định dạng được chỉ định.
  + Tham số: date là đối tượng “Date” cần định dạng.
  + Tham số: pattern là mẫu định dạng mong muốn.
  + Kết quả: String
* XDate.**addDays**(Date date, long days): Date
  + Cộng thêm một số ngày vào đối tượng “Date” hiện tại.
  + Tham số: date là đối tượng “Date” ban đầu.
  + Tham số: days là số ngày cần cộng thêm.
  + Kết quả: Date

### MsgBox

**Mô tả:**

MsgBox là lớp tiện ích cung cấp các phương thức để hiển thị các hộp thoại thông báo, xác nhận và nhập liệu sử dụng JOptionPane trong Java Swing. Lớp này gồm 3 phương thức được mô tả trong phần hướng dẫn sử dụng sau đây.

**Hướng dẫn sử dụng:**

* MsgBox.**alert**(Component parent, String message):
  + Hiển thị một hộp thoại thông báo với thông điệp cho trước.
  + Tham số: parent là thành phần cha của hộp thoại.
  + Tham số: message là thông điệp cần hiển thị trong hộp thoại.
* MsgBox.**confirm**(Component parent, String message): boolean
  + Hiển thị một hộp thoại xác nhận với thông điệp cho trước và chờ người dùng chọn Yes hoặc No.
  + Tham số: parent là thành phần cha của hộp thoại.
  + Tham số: message là thông điệp cần hiển thị trong hộp thoại.
  + Kết quả: boolean
* MsgBox.**prompt**(Component parent, String message): String
  + Hiển thị một hộp thoại nhập liệu với thông điệp cho trước và chờ người dùng nhập dữ liệu.
  + Tham số: parent là thành phần cha của hộp thoại.
  + Tham số: message là thông điệp cần hiển thị trong hộp thoại.
  + Kết quả: String

### Auth

**Mô tả:**

Auth là lớp tiện ích để quản lý xác thực người dùng trong hệ thống. Lớp này chủ yếu để kiểm tra trạng thái đăng nhập của người dùng và quyền hạn của họ. Lớp này gồm 3 phương thức được mô tả trong phần hướng dẫn sử dụng sau đây.

**Hướng dẫn sử dụng:**

* Auth.**clear**():
  + Đặt giá trị của “user” về “null”, tức là đăng xuất người dùng hiện tại.
* Auth.**isLogin**(): boolean
  + Kiểm tra xem có người dùng nào đang đăng nhập hay không.
  + Kết quả: boolean
* Auth.**isManager**(): boolean
  + Kiểm tra xem người dùng hiện tại có vai trò là quản lý hay không.

## Lập trình nghiệp vụ

### Cửa sổ chính EduSysJFrame

|  |  |  |
| --- | --- | --- |
| **TT** | **Phương thức** | **Mô tả** |
| 1 | init() | Mở cửa sổ chào  Mở cửa sổ đăng nhập  Bắt đầu hiển thị đồng hồ hệ thống |
| 2 | openDoiMatKhau() | Mở cửa sổ đổi mật khẩu |
| 3 | openLogin() | Mở cửa sổ đăng nhập |
| 4 | openQLChuyende() | Mở cửa sổ quản lý chuyên đề |
| 5 | openQlhv() | Mở cửa sổ quản lý học viên |
| 6 | openQlkh() | Mở cửa sổ quản lý khóa học |
| 7 | openQlnh() | Mở cửa sổ quản lý người học |
| 8 | openQlnv() | Mở cửa sổ quản lý nhân viên |
| 9 | openChao() | Mở cửa sổ Splash\_Screen |
| 10 | dangXuat() | Đăng xuất người dùng hiện tại và mở cửa sổ đăng nhập |
| 11 | ketThuc() | Kết thúc ứng dụng sau khi xác nhận từ người dùng |
| 12 | openThongKe() | Mở cửa sổ thống kê |
| 13 | openGioiThieu() | Mở cửa sổ giới thiệu sản phẩm |
| 14 | openHuongDan() | Mở file hướng dẫn ‘bai.html’ |
| 15 | openWindow(JDesktopPane desktopPane, JInternalFrame jInternalFrame) | Mở và hiển thị một cửa sổ con (JInternalFrame) trong cửa sổ chính (JDesktopPane). |

### Các cửa sổ hỗ trợ tổ chức

#### ChaoInternalFrame

|  |  |  |
| --- | --- | --- |
| **TT** | **Phương thức** | **Mô tả** |
| 1 | dangNhap() | Thực hiện chức năng đăng nhập vào hệ thống |
| 2 | ketThuc() | Kết thúc ứng dụng sau khi xác nhận từ người dùng |

#### DangNhapJInternalFrame

|  |  |  |
| --- | --- | --- |
| **TT** | **Phương thức** | **Mô tả** |
| 1 | dangNhap() | Thực hiện chức năng đăng nhập vào hệ thống |
| 2 | ketThuc() | Kết thúc ứng dụng sau khi xác nhận từ người dùng |

#### DoiMatKhauJInternalFrame

|  |  |  |
| --- | --- | --- |
| **TT** | **Phương thức** | **Mô tả** |
| 1 | doiMatKhau() | Thực hiện chức năng đổi mật khẩu cho người dùng hiện tại |
| 2 | huyBo() | Đóng cửa sổ hiện tại |

#### GioiThieuJInteralFrame

|  |  |  |
| --- | --- | --- |
| **TT** | **Phương thức** | **Mô tả** |
| 1 | init() | Hiển thị cửa sổ ở giữa màn hình |

### Các cửa sổ chức năng quản lý

#### NhanVienJnternalFrame

|  |  |  |
| --- | --- | --- |
| **TT** | **Phương thức** | **Mô tả** |
| 1 | init() | Khởi tạo dữ liệu và trạng thái ban đầu cho giao diện |
| 2 | Insert() | Thêm mới một nhân viên |
| 3 | Update() | Cập nhật thông tin nhân viên |
| 4 | Delete() | Xóa một nhân viên |
| 5 | getForm() | Lấy thông tin từ form và tạo đối tượng “NhanVien” |
| 6 | setForm() | Đặt thông tin nhân viên vào form |
| 7 | clearForm() | Xóa nội dụng trong form và đặt trạng thái ban đầu |
| 8 | Edit() | Chỉnh sửa thông tin nhân viên đang chọn |
| 9 | fillTable() | Tải danh sách nhân viên từ CSDL và hiển thị trong bảng |
| 10 | updateStatus() | Cập nhật trạng thái của các nút và trường nhập liệu dựa trên tình trạng hiện tại của form |
| 11 | First() | Dòng đầu tiên của bảng |
| 12 | Prev() | Dòng kề phía trước |
| 13 | Next() | Dòng kề phía sau |
| 14 | Last() | Dòng cuối của bảng |

#### ChuyenDeJInternalFrame

|  |  |  |
| --- | --- | --- |
| **TT** | **Phương thức** | **Mô tả** |
| 1 | init() | Khởi tạo dữ liệu và trạng thái ban đầu cho giao diện |
| 2 | Insert() | Thêm mới một chuyên đề |
| 3 | Update() | Cập nhật thông tin chuyên đề |
| 4 | Delete() | Xóa một chuyên đề |
| 5 | getForm() | Lấy thông tin từ form và tạo đối tượng “ChuyenDe” |
| 6 | setForm() | Đặt thông tin chuyên đề vào form |
| 7 | clearForm() | Xóa nội dụng trong form và đặt trạng thái ban đầu |
| 8 | Edit() | Chỉnh sửa thông tin chuyên đề đang chọn |
| 9 | fillTable() | Tải danh sách chuyên đề từ CSDL và hiển thị trong bảng |
| 10 | updateStatus() | Cập nhật trạng thái của các nút và trường nhập liệu dựa trên tình trạng hiện tại của form |
| 11 | First() | Dòng đầu tiên của bảng |
| 12 | Prev() | Dòng kề phía trước |
| 13 | Next() | Dòng kề phía sau |
| 14 | Last() | Dòng cuối của bảng |
| 15 | chonAnh() | Chọn và tải ảnh cho chuyên đề |

#### NguoiHocJIternalFrame

|  |  |  |
| --- | --- | --- |
| **TT** | **Phương thức** | **Mô tả** |
| 1 | init() | Khởi tạo dữ liệu và trạng thái ban đầu cho giao diện |
| 2 | Insert() | Thêm mới một người học |
| 3 | Update() | Cập nhật thông tin người học |
| 4 | Delete() | Xóa một người học |
| 5 | getForm() | Lấy thông tin từ form và tạo đối tượng “NguoiHoc” |
| 6 | setForm() | Đặt thông tin người học vào form |
| 7 | clearForm() | Xóa nội dụng trong form và đặt trạng thái ban đầu |
| 8 | Edit() | Chỉnh sửa thông tin chuyên đề đang chọn |
| 9 | fillTable() | Tải danh sách người học từ CSDL và hiển thị trong bảng |
| 10 | updateStatus() | Cập nhật trạng thái của các nút và trường nhập liệu dựa trên tình trạng hiện tại của form |
| 11 | First() | Dòng đầu tiên của bảng |
| 12 | Prev() | Dòng kề phía trước |
| 13 | Next() | Dòng kề phía sau |
| 14 | Last() | Dòng cuối của bảng |
| 15 | fillFind() | Tìm kiếm người học theo tiêu chí và hiển thị kết quả trong bảng |
| 16 | Ngaydk() | Lấy ngày đăng kí hiện tại |

#### KhoaHocJInternalFrame

|  |  |  |
| --- | --- | --- |
| **TT** | **Phương thức** | **Mô tả** |
| 1 | init() | Khởi tạo dữ liệu và trạng thái ban đầu cho giao diện |
| 2 | Insert() | Thêm mới một khóa học |
| 3 | Update() | Cập nhật thông tin khóa học |
| 4 | Delete() | Xóa một khóa học |
| 5 | getForm() | Lấy thông tin từ form và tạo đối tượng “KhoaHoc” |
| 6 | setForm() | Đặt thông tin khóa học vào form |
| 7 | clearForm() | Xóa nội dụng trong form và đặt trạng thái ban đầu |
| 8 | Edit() | Chỉnh sửa thông tin một khóa học |
| 9 | fillTable() | Tải danh sách khóa học từ CSDL và hiển thị trong bảng |
| 10 | updateStatus() | Cập nhật trạng thái của các nút và trường nhập liệu dựa trên tình trạng hiện tại của form |
| 11 | First() | Dòng đầu tiên của bảng |
| 12 | Prev() | Dòng kề phía trước |
| 13 | Next() | Dòng kề phía sau |
| 14 | Last() | Dòng cuối của bảng |
| 15 | chonChuyenDe() | Xử lý sự kiện khi người dùng chọn chuyên đề, hiển thị thông tin chuyên đề và danh sách khóa học tương ứng |

#### HocVienJInternalFrame

|  |  |  |
| --- | --- | --- |
| **TT** | **Phương thức** | **Mô tả** |
| 1 | init() | Khởi tạo ban đầu khi mở ứng dụng |
| 2 | fillComboBoxChuyenDe() | Điền dữ liệu vào ComboBoxChuyenDe với danh sách chuyên đề từ CSDL |
| 3 | fillComboBoxKhoaHoc() | Điền dữ liệu vào ComboBoxKhoaHoc với danh sách khóa học thuộc chuyên đề được chọn từ ComboBoxChuyenDe |
| 4 | fillTableHocVien() | Điền dữ liệu vào bảng HocVien với danh sách học viên của khóa học được chọn từ ComboBoxKhoaHoc |
| 5 | fillNguoiHoc() | Điền dữ liệu vào bảng NguoiHoc với danh sách người học chưa tham gia khóa học được chọn từ ComboBoxKhoaHoc |
| 6 | removeHocVien() | Xóa các học viên được chọn từ bảng HocVien và cập nhật lại bảng |
| 7 | addHocVien() | Thêm người học được chọn từ bảng NguoiHoc vào khóa học được chọn từ ComboBoxKhoaHoc |
| 8 | updateDiem() | Cập nhật điểm của các học viên được chọn từ bảng HocVien |
| 9 | fillFind() | Chức năng tìm kiếm người học theo các tiêu chí và hiển thị kết quả trong bảng |

### Cửa sổ chức năng tổng hợp - thống kê

|  |  |  |
| --- | --- | --- |
| **TT** | **Phương thức** | **Mô tả** |
| 1 | init() | Khởi tạo ban đầu khi mở ứng dụng |
| 2 | fillComboBoxKhoaHoc() | Điền dữ liệu vào ComboBoxKhoaHoc với danh sách khóa học từ CSDL |
| 3 | fillTableBangDiem() | Điền dữ liệu vào bảng BangDiem với bảng điểm của khóa học được chọn từ ComboBoxKhoaHoc |
| 4 | getXepLoai(double diem) | Trả về xếp loại dựa trên điểm số |
| 5 | fillTableNguoiHoc() | Ddieefnf dữ liệu vào bảng NguoiHoc với thống kê lượng người học |
| 6 | fillTableDiemChuyenDe() | Điền dữ liệu vào bảng DiemChuyenDe với thống kê điểm của từng chuyên đề |
| 7 | fillComboBoxNam() | Điền dữ liệu vào ComboBoxNam với danh sách các năm có khóa học được tổ chức |
| 8 | fillTableDoanhThu() | Điền dữ liệu vào bảng DoanhThu với thống kê doanh thu theo năm được chọn từ ComboBoxNam |

### 

# Kiểm thử phần mềm và sửa lỗi

## DangNhapJInternalframe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TH** | **Mô tả tình huống** | **Dữ liệu mẫu** | **Kết quả trông đợi** | **Fix** |
| 1 | Đăng nhập để trống tên và mật khẩu |  | Yêu cầu nhập | ok |
| 2 | Đăng nhập sai tên | TeoVN, songsong | Sai tên đăng nhập | not |
| 3 | Đăng nhập sai mật khẩu | TeoNV, 123 | Sai mật khẩu | 95% |
| 4 | Đăng nhập đúng | TeoNV, songlong | Đóng cửa sổ |  |

## DoiMatKhauJInternalframe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TH** | **Mô tả tình huống** | **Dữ liệu mẫu** | **Kết quả trông đợi** | **Fix** |
| 1 | Để trống mã, mật khẩu hiện tại, mật khẩu mới |  | Yêu cầu nhập | ok |
| 2 | Xác nhận sai mã, mật khẩu | NoPT, duanmau | Thông báo sai, yêu cầu nhập lại | ok |
| 3 | Xác nhận mật khẩu mới sai | TeoNV, 123 | Thông báo xác nhận sai, yêu cầu nhập lại | ok |
| 4 | Đổi mật khẩu thành công | TeoNV, 123 | Xóa sạch form | ok |

## NhanVienJInternalframe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TH** | **Mô tả tình huống** | **Dữ liệu mẫu** | **Kết quả trông đợi** | **Fix** |
| 1 | Để trông form |  | Yêu cầu nhập | Ok |
| 2 | Kiểm tra trùng mã | TeoNV | Thông báo trùng | Ok |
| 3 | Thêm, sửa, xóa dựa vào vai trò | TeoNV, NoPT | Thông báo không được thực hiện hành động nếu là nhân viên | Ok |
| 4 | Check event MouseClicked |  | Hiển thị lại thông tin nhân viên lên form | ok |
| 5 | Nút điều hướng |  | Form thay đổi dữ liệu | ok |

## ChuyenDeJInternalframe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TH** | **Mô tả tình huống** | **Dữ liệu mẫu** | **Kết quả trông đợi** | **Fix** |
| 1 | Để trông form |  | Yêu cầu nhập | Ok |
| 2 | Kiểm tra trùng mã | JAV01 | Thông báo trùng | Ok |
| 3 | Kiểm tra định dạng | -1, aA@,… | Yêu cầu nhập lại | ok |
| 4 | Xóa dựa vào vai trò | TeoNV, NoPT | Thông báo không được thực hiện hành động nếu là nhân viên | Ok |
| 5 | Check event MouseClicked |  | Hiển thị lại thông tin nhân viên lên form | ok |
| 6 | Nút điều hướng |  | Form thay đổi dữ liệu | ok |

## NguoiHocJInternalframe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TH** | **Mô tả tình huống** | **Dữ liệu mẫu** | **Kết quả trông đợi** | **Fix** |
| 1 | Để trông form |  | Yêu cầu nhập | Ok |
| 2 | Kiểm tra trùng mã | JAV01 | Thông báo trùng | Ok |
| 3 | Kiểm tra định dạng | -1, aA@,… | Yêu cầu nhập lại | ok |
| 4 | Nút điều hướng |  | Form thay đổi dữ liệu | ok |
| 5 | Check event MouseClicked |  | Hiển thị lại thông tin nhân viên lên form | ok |
| 6 | Nút tìm kiếm | Abc, tèo | Tìm kiếm theo tên | ok |

## KhoaHocJInternalframe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TH** | **Mô tả tình huống** | **Dữ liệu mẫu** | **Kết quả trông đợi** | **Fix** |
| 1 | Kiểm tra định dạng | -1, aA@,… | Yêu cầu nhập lại | ok |
| 2 | Nút điều hướng |  | Form thay đổi dữ liệu | ok |
| 3 | Check event MouseClicked |  | Hiển thị lại thông tin nhân viên lên form | ok |
| 4 | Form và table thay đổi theo comboBox |  | Form và table hiện thị theo comboBox | ok |
| 5 | Kiểm tra quyền xóa | TeoNV, NoPT | Thông báo không có quyền nếu là nhân viên | ok |

## HocVienJInternalframe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TH** | **Mô tả tình huống** | **Dữ liệu mẫu** | **Kết quả trông đợi** | **Fix** |
| 1 | Xóa học viên |  | Xóa một hoặc nhiều học viên một lúc | Ok |
| 2 | Thêm học viên |  | Thêm một hoặc nhiều học viên một lúc | Ok |
| 3 | Nhập, sửa điểm | A,-1,11 | Thông báo sai dữ liệu | Ok |
| 4 | ComboBox Khóa Học |  | Dựa vào comboBox chuyên đề | ok |
| 5 | TextField Tìm kiếm | Abc, XYZ, tèo | Hiện thị thông viên theo họ tên | ok |

## ThongKeJInternalframe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TH** | **Mô tả tình huống** | **Dữ liệu mẫu** | **Kết quả trông đợi** | **Fix** |
| 1 | Bảng Khóa học |  | Thay đổi theo comboBox | ok |
| 2 | Bảng doanh thu |  | Thay đổi theo năm | ok |

# Đóng gói và triển khai

## Sản phẩm phần mềm

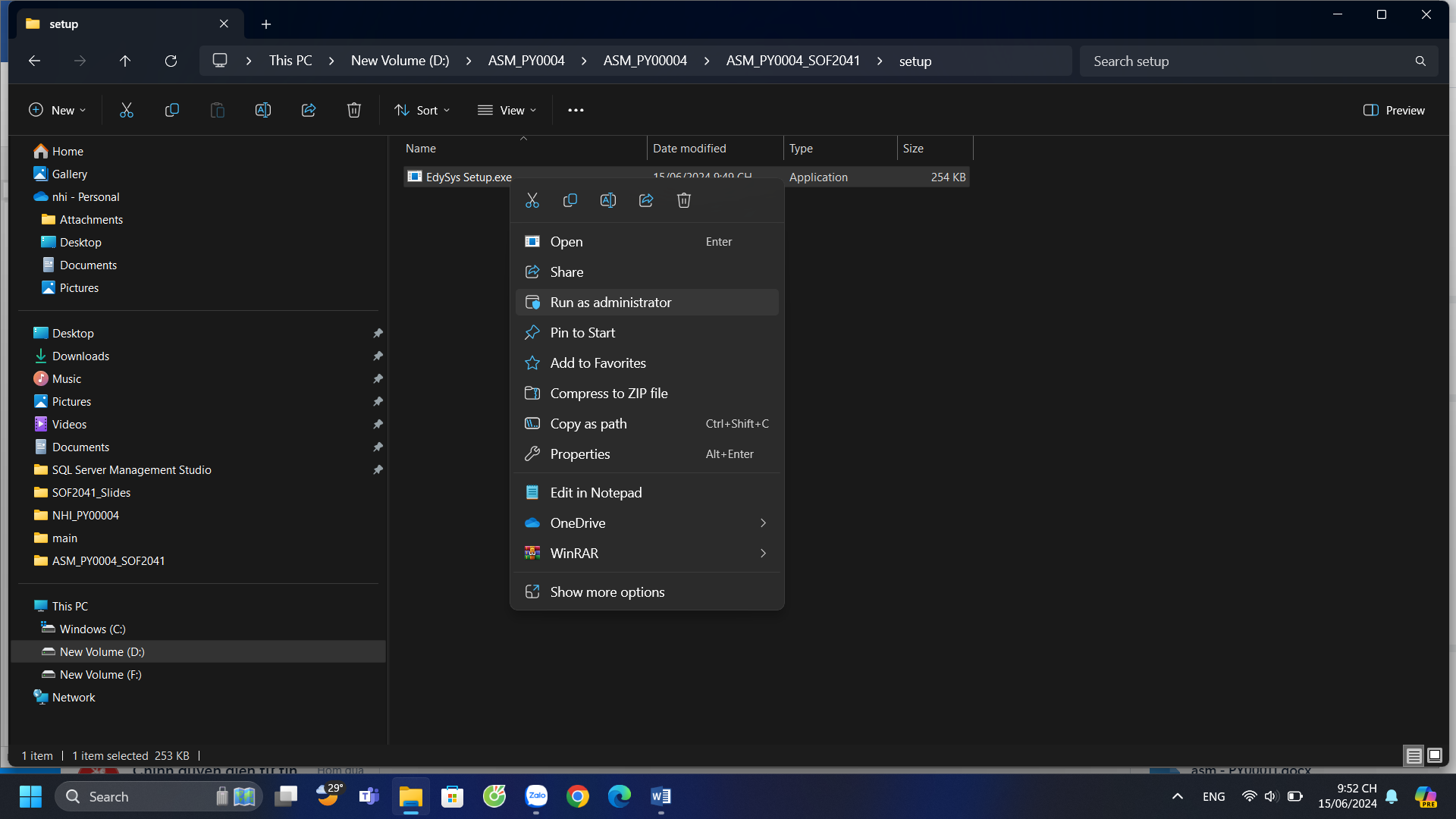
|  |  |  |
| --- | --- | --- |
| **TT** | **Thành phần** | **Mô tả** |
| 1 | Setup.exe | File cài đặt |
| 2 | EduSys.sql | Cơ sở dữ liệu |
| 3 | Readme.txt | Hướng dẫn cài đặt |
| 4 | Guide.docx | Hướng dẫn sử dụng |

## Hướng dẫn cài đặt

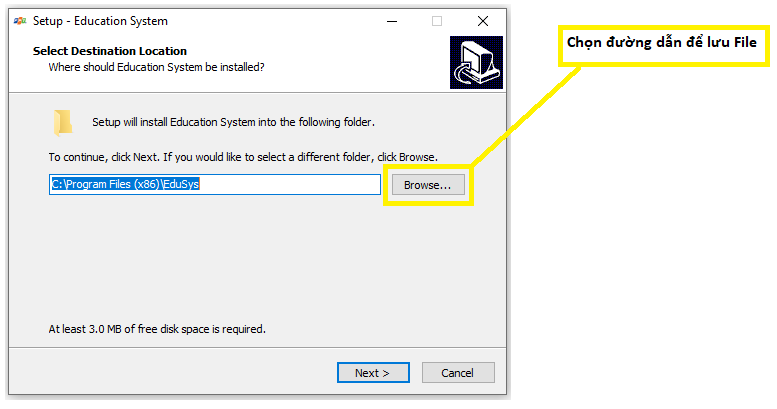
* Bước 1: Cài đặt file EduSys Setup.exe



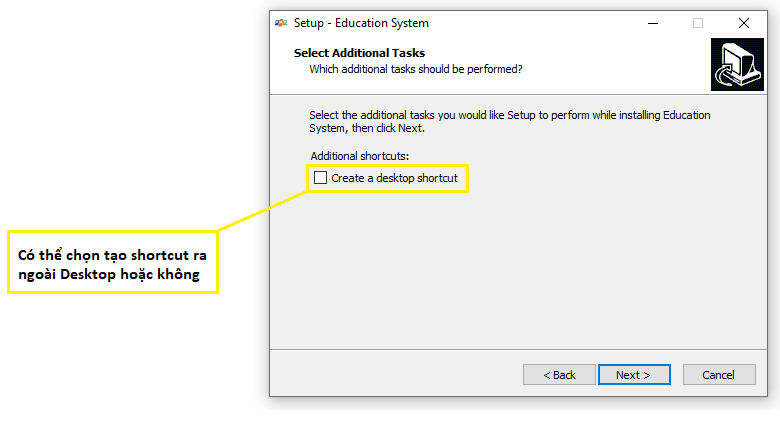
* Bước 2: Khởi động file



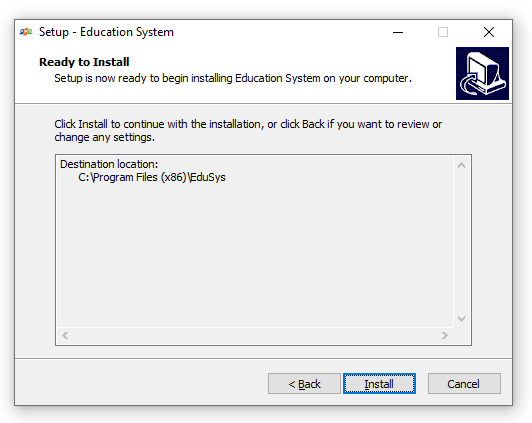
* Bước 3: Chọn đường dẫn để lưu => Next



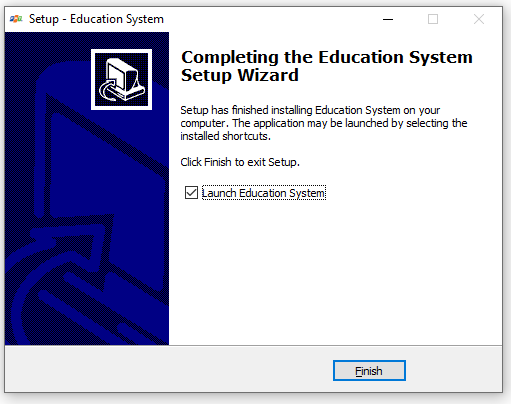
* Bước 4: Tiếp tục Next



* Bước 5:Bấm Install để cài đặt



* Bước 6: Bấm Finish để hoàn thành cài đặt



# KẾT LUẬN

## Khó khăn

* Phải tự túc tìm tòi và phát triển dự án theo cá nhân
* Không có nhóm làm cùng dự án
* Có những sai sót nhỏ và phải tự mình tìm rồi sửa
* Lần đầu làm một dự án lớn
* Chưa học Java3 nên gặp 1 số khó khăn

## Thuận lợi

* Có sẵn tài liệu tham khảo trên CMS
* Có code để tham khảo và phát triển