|  |
| --- |
| FPT POLYTECHNIC |
| **DỰ ÁN 1** |
| NGÀNH CÔNG NGHỆ THÔNG TIN (ỨNG DỤNG PHẦN MỀM) |



|  |
| --- |
| 12/1/2019 |

|  |
| --- |
| GIẢNG VIÊN: HOÀNG QUỐC VIỆT |
| SINH VIÊN THỰC HIỆN |
| 1. Đào Duy Lương ( Trưởng nhóm ) 2. Nguyễn Ngọc Long 3. Trần Ngọc Hải |

NHẬN XÉT

**(Của giáo viên hướng dẫn)**

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Giáo viên hướng dẫn ký, ghi rõ họ tên

MỤC LỤC

[1 Phân tích 4](#_Toc26662182)

[1.1 Hiện trạng 4](#_Toc26662183)

[1.2 Yêu cầu hệ thống 4](#_Toc26662184)

[1.3 Use case 4](#_Toc26662185)

[2 Thiết kế 4](#_Toc26662186)

[2.1 Mô hình triển khai 4](#_Toc26662187)

[2.2 Thiết kế CSDL 8](#_Toc26662188)

[2.2.1 Sơ đồ quan hệ thực thể 8](#_Toc26662189)

[2.2.2 Thiết kế chi tiết các thực thể 8](#_Toc26662190)

[2.3 Thiết kế giao diện 8](#_Toc26662191)

[2.3.1 Sơ đồ tổ chức giao diện 8](#_Toc26662192)

[2.3.2 Bản MorkUp giao diện 16](#_Toc26662193)

[2.3.3 Thiết kế giao diện cho các chức năng nghiệp vụ 21](#_Toc26662194)

[3 Thực hiện viết mã 27](#_Toc26662195)

[3.1 Viết mã tạo CSDL 27](#_Toc26662196)

[3.1.1 Tạo CSDL 30](#_Toc26662197)

[3.1.2 SQL truy vấn và thao tác 30](#_Toc26662198)

[3.1.3 Các thủ tục lưu tổng hợp thống kê 30](#_Toc26662199)

[3.2 Lập trình JDBC 30](#_Toc26662200)

[3.2.1 Lớp hỗ trợ 30](#_Toc26662201)

[3.2.2 Model class - Các lớp mô tả dữ liệu 30](#_Toc26662202)

[3.2.3 DAO Class - Các lớp truy xuất dữ liệu 30](#_Toc26662203)

[3.3 Viết mã cho ứng dụng 30](#_Toc26662204)

[3.3.1 Xử lý Form X 30](#_Toc26662205)

[3.3.2 Xử lý Form Y 30](#_Toc26662206)

[4 Kiểm thử 30](#_Toc26662207)

[4.1 Kiểm thử form X 30](#_Toc26662208)

[4.2 Kiểm thử form Y 30](#_Toc26662209)

[5 Đóng gói và triển khai 31](#_Toc26662210)

[5.1 Hướng dẫn chuyển đổi jar thành exe 31](#_Toc26662211)

[5.2 Hướng dẫn cài đặt triển khai 31](#_Toc26662212)

[5.3 Hướng dẫn sử dụng phần mềm 31](#_Toc26662213)

# Phân tích

## Hiện trạng

Nhà trọ ABC đang có một số phòng trọ cho thuê, việc quản lý các phòng đang được thực hiện bằng tay và thủ công trên giấy , chỉ có Chủ nhà trọ quản lý việc thu tiền đến làm hợp đồng cho Khách trọ.

Hiện nay Nhà trọ đang gặp khá nhiều khó khắn do số lượng phòng đang tăng lên và việc quản lý đang trở nên khó khăn, dễ sai sót và không bảo mật. Mới nhất là Chủ trọ đã thuê thêm nhân viên để quản lý nhà trọ nên việc quản lý về phần hợp đồng và thanh toán trở nên không an toàn.

## Yêu cầu hệ thống

Nhà trọ ABC mong muốn xây dựng một phần mềm để giải quyết được những khó khắn trên.

* **Yêu cầu chức năng nghiệp vụ**
* Quản lý phòng
* Quản lý nhân viên
* Quản lý hợp đồng
* Quản lý các dịch vụ
* Quản lý khách trọ
* Thanh toán
* Hóa đơn chi tiết
* Hóa đơn dịch vụ
* **Yêu cầu về bảo mật**
* Tất cả mọi thành viên phải đăng nhập mới có thể sử dụng được phần mềm
* Chủ nhà trọ được phép thực hiện được hết tất cả các chức năng
* Nhân viên không được phép sử dụng Quản lý Nhân Viên
* **Yêu cầu về môi trường công nghệ**
  + Ứng dụng phải được thực với công nghệ Swing và JDBC chạy trên mọi hệ điều hành với môi trường JDK tối thiểu 1.8
* Hệ quản trị CSDL SQL Server 2008 trở lên

## Use case

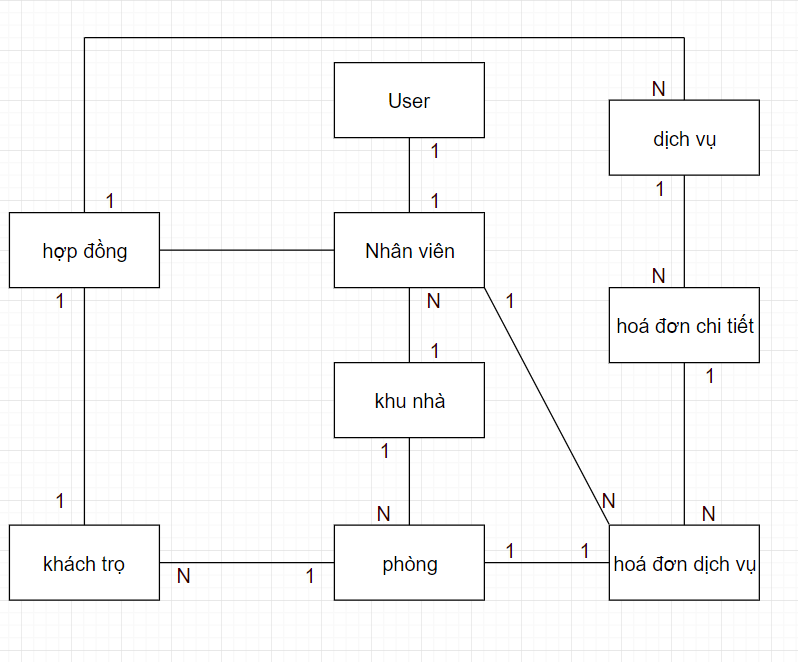
# Thiết kế

## Mô hình triển khai

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Chức năng** | **Nội dung** | **Input** | **Output** |
| 1 | Đăng nhập | Chủ nhà và nhân viên đăng nhập để sử dụng các chức năng có được từ chức năng phân quyền cấp cho. | User và password | Hiển thị kết quả đang nhập và chuyển hướng ra trang chủ nếu đăng nhập thành công. |
| 2 | Quản lý nhân viên | Chủ nhà là người duy nhất có quyền sử dụng chức năng này:  1. Hiển thị danh sách nhân viên  2. Thêm thông tin nhân viên  3. Chỉnh sửa thông tin nhân viên  4. Xóa thông tin nhân viên đó | 2. Thêm: thực hiện nhập userID, mã, tên, số ĐT, số CMT, email, chọn giới tính cho nhân viên và khu quản lý (ví dụ như khu A, B, C….)  3. Sửa: ĐT, số CMT, Email liên hệ và chọn khu quản lý (ví dụ như khu A, B, C….) thay thế cho nhân viên.  4. Xóa: điền mã Nhân viên muốn xóa | 2. Hệ thống tự động tạo user và password mặc định và vai trò của nhân viên và hiển thị thông báo nếu thêm thông tin thành công hoặc hiện thông báo lỗi hay các mục chủ nhà điền thiếu mục nào đó.  3. Hệ thống hiển thị thông báo nếu chỉnh sửa thông tin thành công hoặc hiện thông báo lỗi hay các mục chủ nhà điền thiếu mục nào đó.  4. Hệ thống thông báo xóa thành công và làm mới lại bảng thông tin nhân viên |
| 3 | Quản lý khách trọ | Nhân viên chỉ có quyền sử dụng các chức năng này:  1. Thêm thông tin khách trọ.  2. Tìm thông tin khách trọ theo mã của họ.  Chủ nhà có quyền sử dụng các chức năng ngoài chức năng trên:  3.Chỉnh sửa thông tin nhân viên | 1.Thêm: thực hiện nhập mã, tên, số ĐT, số CMT, email, chọn giới tính cho nhân viên và mã phòng trọ khách trọ thuê.  2. Tìm: mã khách trọ mà chủ nhà hoặc nhân viên muốn tìm  3. Sửa: điền số ĐT, số CMT, Email liên hệ của khách trọ. | 1.Hệ thống hiển thị thông báo nếu thêm thông tin thành công hoặc hiện thông báo lỗi hay các mục chủ nhà hoặc nhân viên điền thiếu mục nào đó.  2. Hệ thống hiện thị thông báo kết quả tìm kiếm và hiện thị thông tin lên bảng (nếu có thông tin)  3.Hệ thống hiển thị thông báo nếu chỉnh sửa thông tin thành công hoặc hiện thông báo lỗi hay các mục chủ nhà điền thiếu mục nào đó. |
| 4 | Quản lý dịch vụ | Chủ nhà là người duy nhất  quyền sử dụng các chức năng ngoài chức năng trên:  1. Thêm thông tin dịch vụ  2. Chỉnh sửa giá dịch vụ | 1.Thêm: điền mã dịch vụ, tên dịch vụ, giá dịch v  2. Sửa: điền giá dịch vụ mới cho dịch vụ mình muốn thay đổi giá | 1. Hệ thống hiển thị thông báo nếu thêm thông tin thành công hoặc hiện thông báo lỗi hay các mục chủ nhà điền thiếu mục nào đó.  2. Hệ thống hiển thị thông báo nếu chỉnh sửa thông tin thành công hoặc hiện thông báo lỗi hay các mục chủ nhà điền thiếu mục nào đó. |
| 5 | Quản lý phòng trọ | Nhân viên chỉ được sử dụng chức năng:  1.Tìm phòng trống  Chủ nhà ngoài chức năng trên còn sử dụng chức năng:  2. Thêm phòng trọ  3. Sửa giá phòng và thay đổi trạng thái của phòng | 1. Tìm: bấm vào nút chọn phòng với các nút là (Còn trống, đã thuê, đang sửa chữa) để biết thông tin phòng trọ  2. Thêm: điền mã phòng, tên phòng, giá phòng, diện tích, trạng thái phòng (Còn trống, đã thuê, đang sửa chữa)  3. Sửa: điền giá phòng mới hoặc trạng thái phòng (khi đã sửa chữa xong) | 1. Hệ thống hiện thông tin mà nút nhân viên hoặc chủ nhà muốn tìm hiện lên bảng thông tin phòng trọ  2. Hệ thống hiển thị thông báo nếu thêm thông tin thành công hoặc hiện thông báo lỗi hay các mục chủ nhà điền thiếu mục nào đó.  3. Hệ thống hiển thị thông báo nếu chỉnh sửa thành công hoặc hiện thông báo lỗi hay các mục chủ nhà điền thiếu mục sửa nào đó. |
| 6 | Hóa đơn dịch vụ | Chủ nhà và nhân viên đều có thể sử dụng chức năng :  1.Tìm hóa đơn dịch vụ theo phòng thuê | 1. Nhập mã phòng thuê muốn tìm | 1. Hệ thống hiển thị hóa đơn dịch vụ của phòng |
| 7 | Hóa đơn chi tiết | Chủ nhà hoặc nhân viên dùng chức năng:  1.Hiện thị thông tin hóa đơn chi tiết dịch vụ  1.Tìm hóa đơn chi tiết theo phòng thuê | 1. Nhập mã phòng thuê muốn tìm | 1. Hệ thống hiển thị hóa đơn chi tiết của phòng |
| 8 | Đổi mật khẩu | Cho nhân viên, chủ nhà đổi mật khẩu tài khoản đăng nhập của mình. | Mật khẩu hiện tại, mật khẩu mới và xác nhận mật khẩu mới | Thông báo sửa thành công hoặc thất bại. Thay đổi mật khẩu trong cơ sở dữ liệu. |
| 9 | Quản lí hợp đồng | Cho phép nhân viên và chủ nhà:  1.Thêm mới hợp đồng  2.Sửa hợp đồng  3.Xoá hợp đồng  4.Tìm kiếm hợp đồng | 1.Thêm: điền thông tin của khách trọ, phòng, số khách trọ, dịch vụ đăng kí, ngày thuê, ngày hết hạn và trạng thái.  2. Sửa: Mã hợp đồng cần sửa.Thông tin thay đổi : thông tin số khách trọ, dịch vụ, ngày hết hạn và trạng thái.  3.Xóa: Mã hợp đồng cần xoá.  4.Tìm: Mã hợp đồng cần tìm. | 1.Thông báo tình trạng thêm. Thay đổi dữ liệu trong cơ sở, hiển thị lại dữ liệu trên form  .  2. Thông báo tình trạng sửa. Thay đổi dữ liệu trong cơ sở, hiển thị lại dữ liệu trên form.  3. Thông báo tình trạng xoá. Thay đổi trường trạng thái trong cơ sở dữ liệu để đánh dấu đã xoá nhưng không xoá trên cơ sở dữ liệu. không hiển thị  4. Hiển thị thông tin tìm kiếm nếu có. Hoặc thông báo nếu không tìm thấy hoặc có lỗi. |

## Thiết kế CSDL

### Sơ đồ quan hệ thực thể

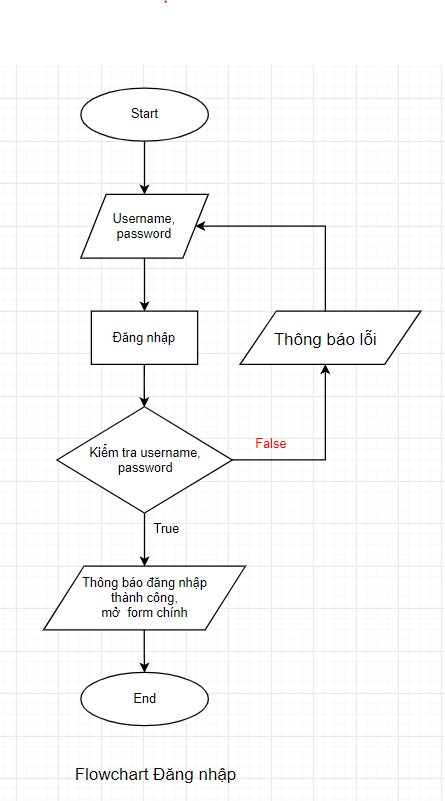
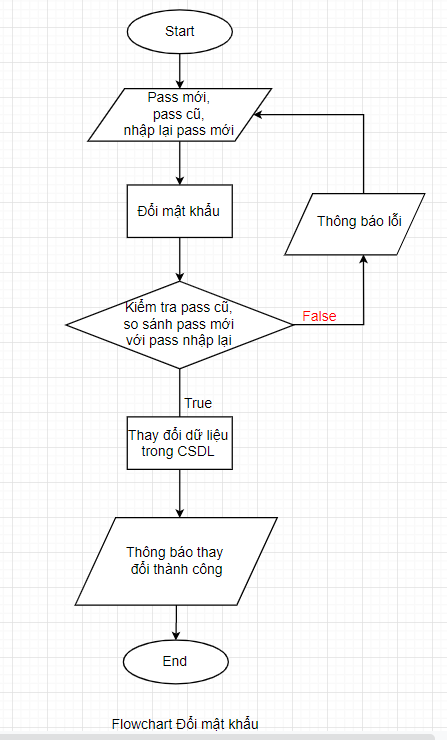


### Thiết kế chi tiết các thực thể

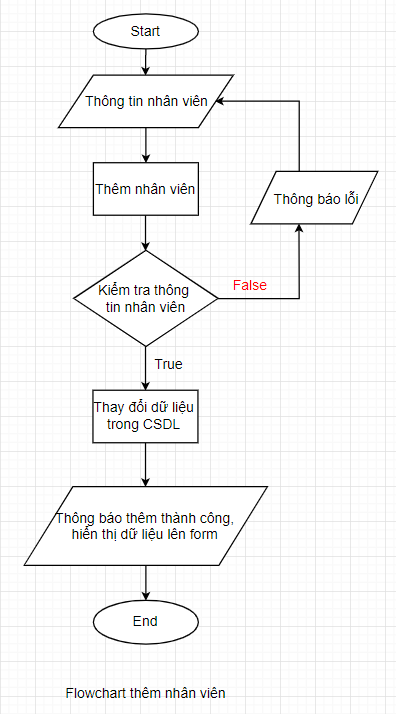
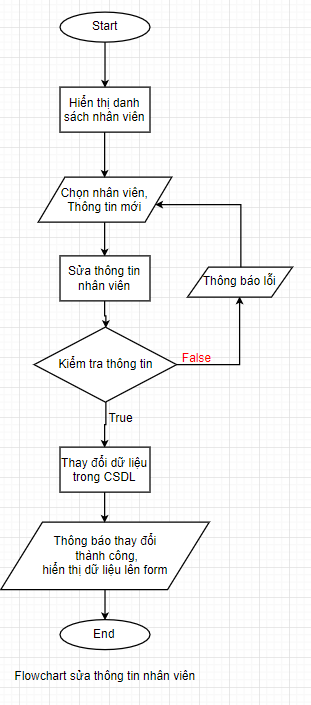
## Thiết kế giao diện

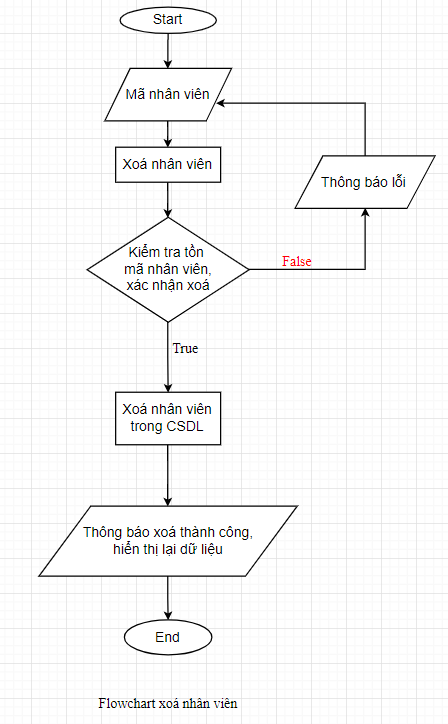
### Sơ đồ tổ chức giao diện

**Đăng nhập**

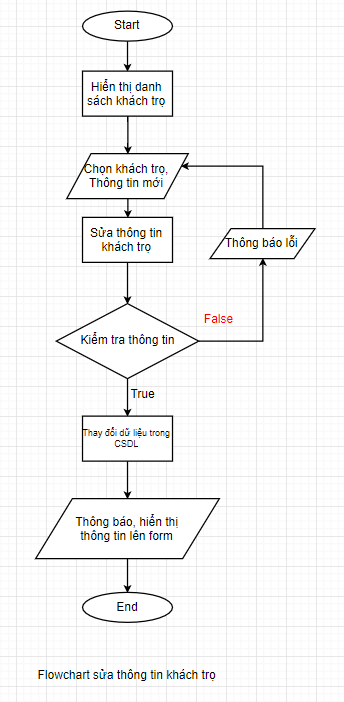
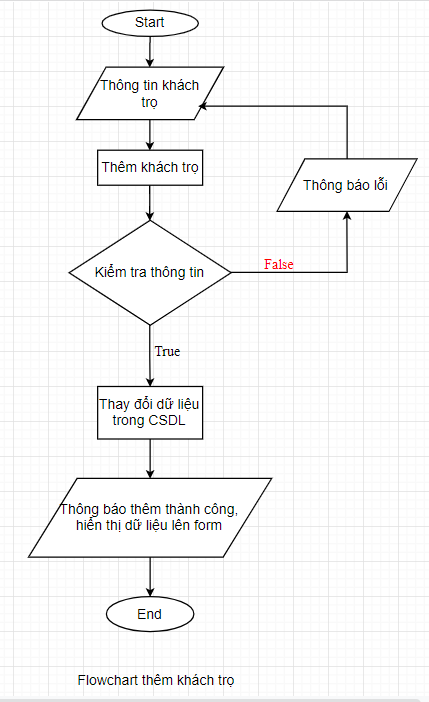
 

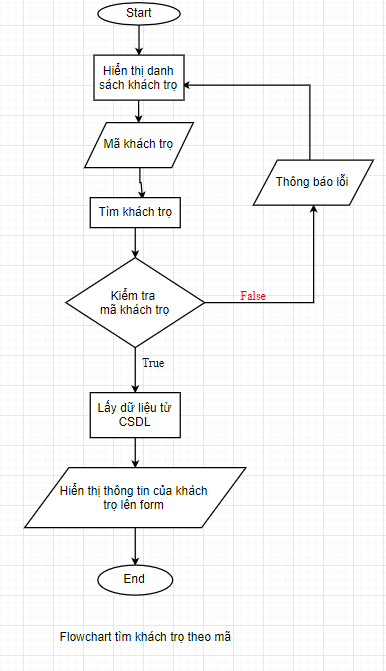
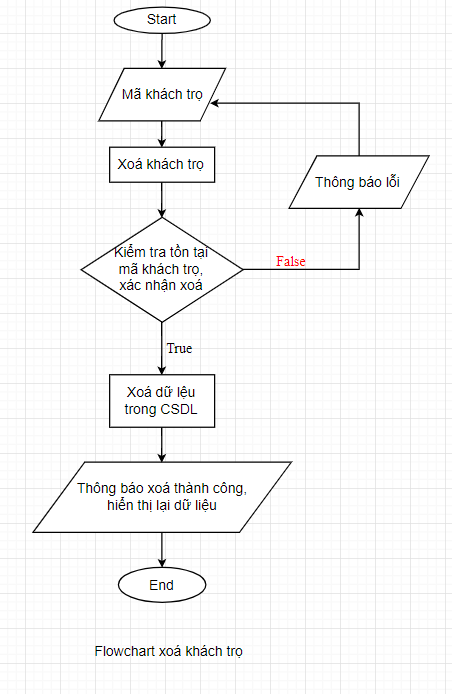
**Bảng nhân viên**

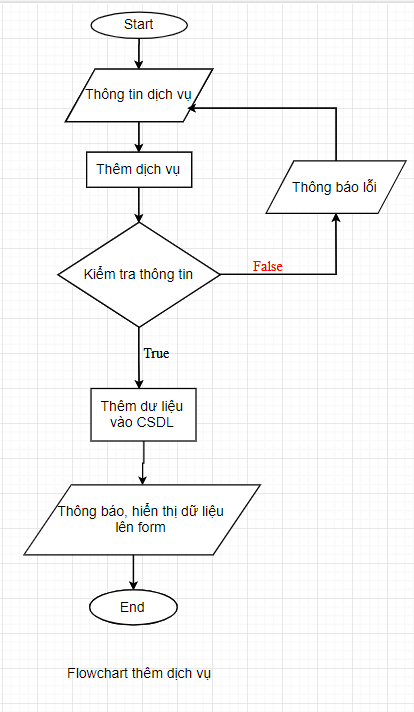
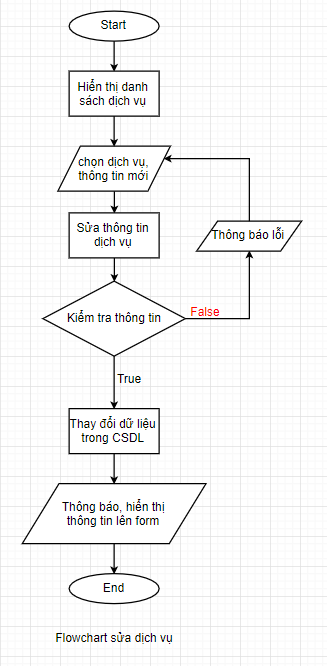


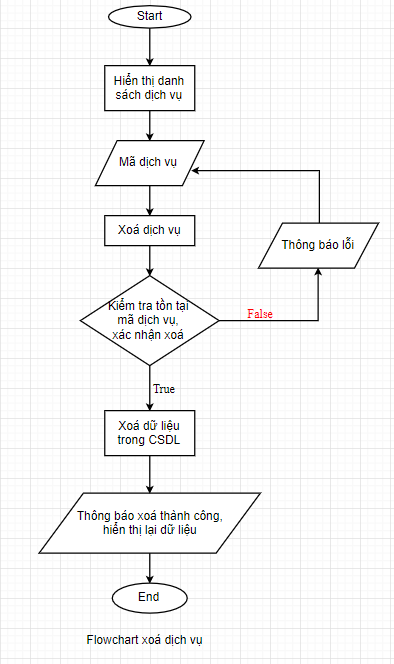
**Bảng khách trọ**



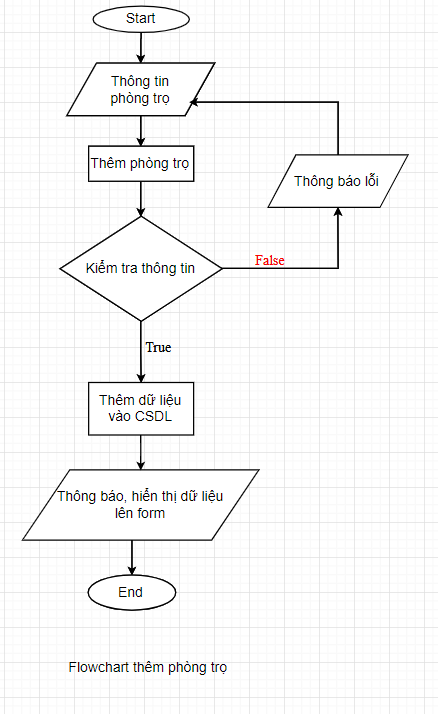
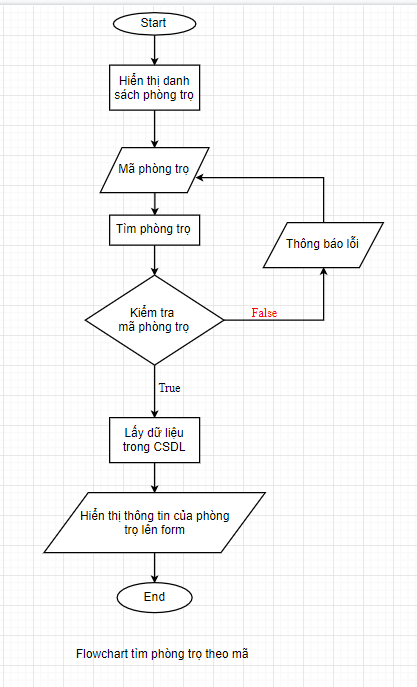


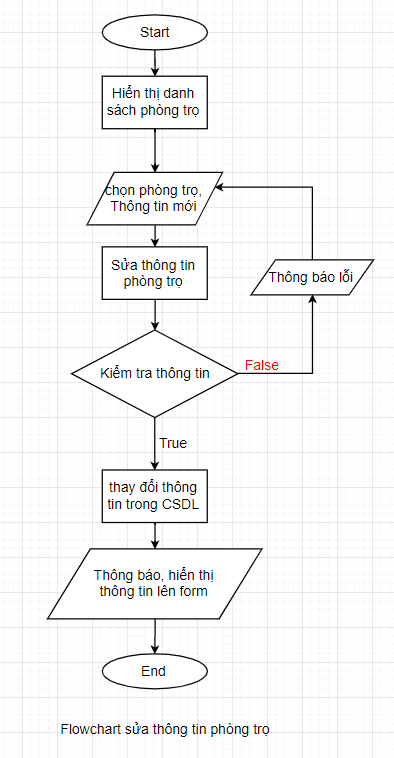
**Bảng Dịch vụ**

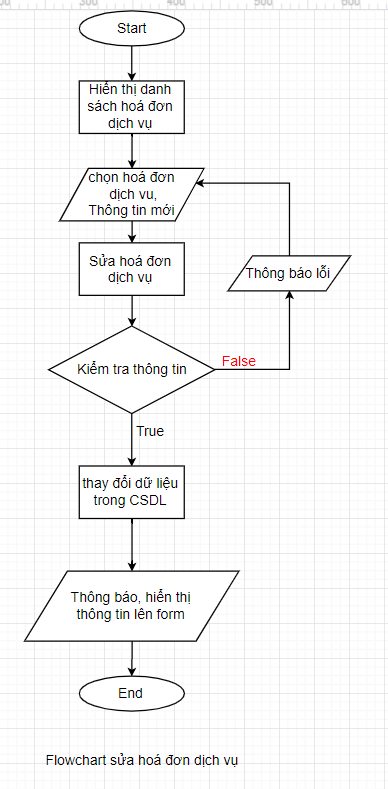
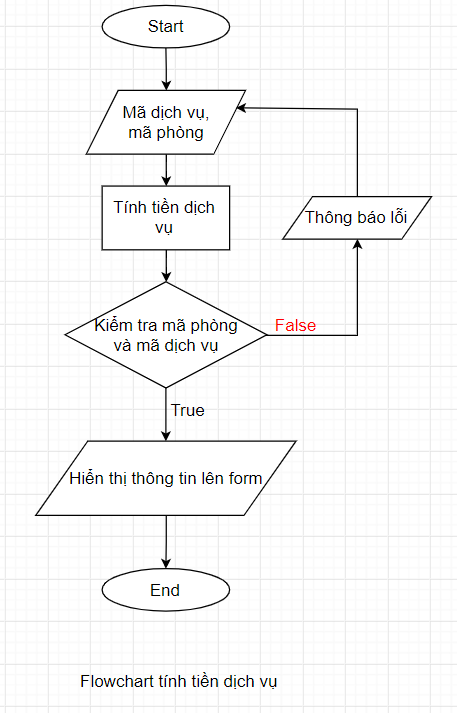


**Bảng phòng trọ**

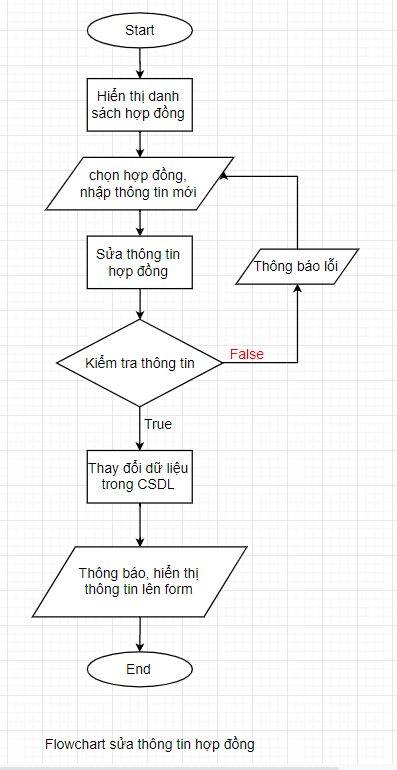
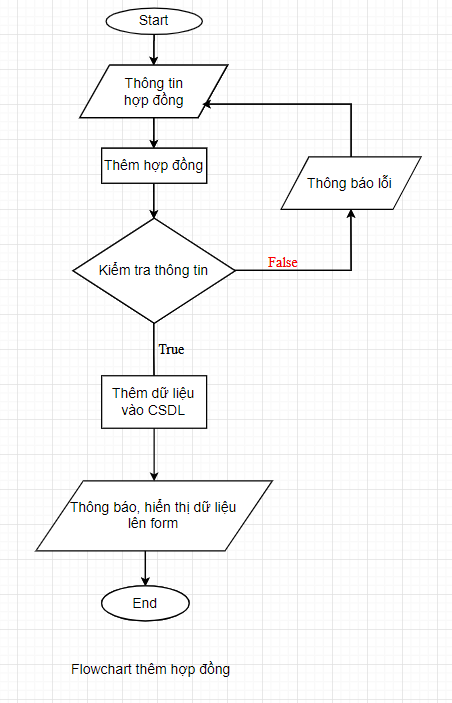


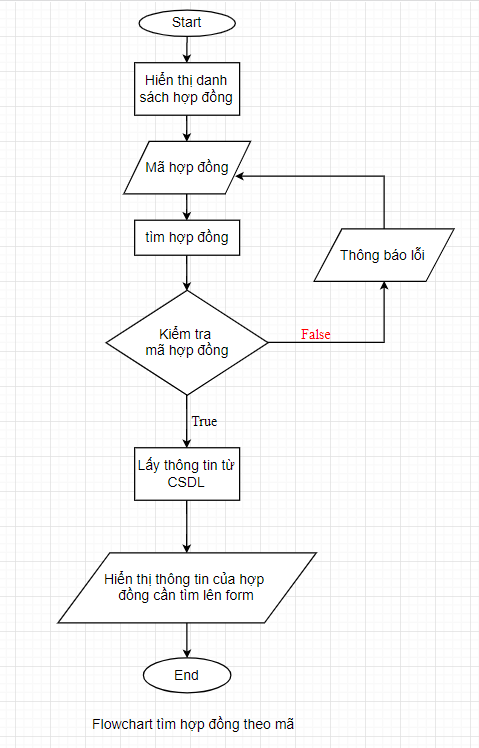


**Bảng hóa đơn dịch vụ**



**Bảng hợp đồng**

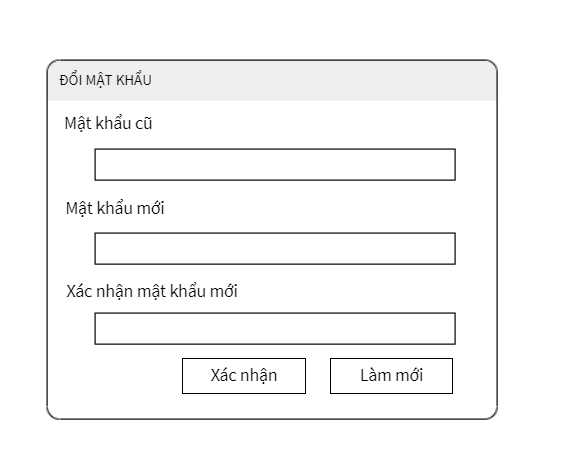




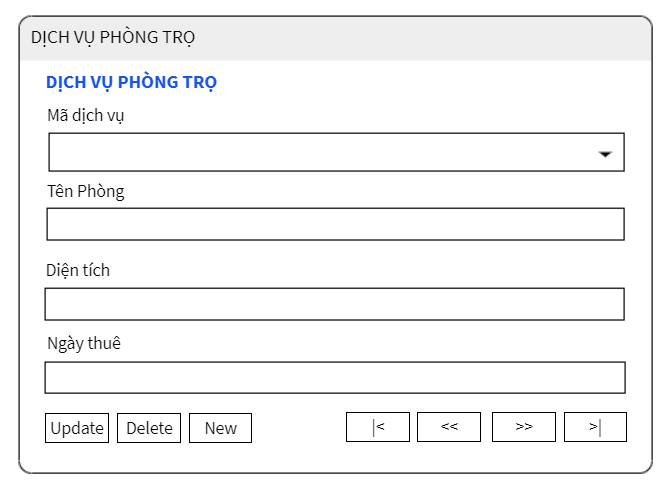
### Bản MorkUp giao diện

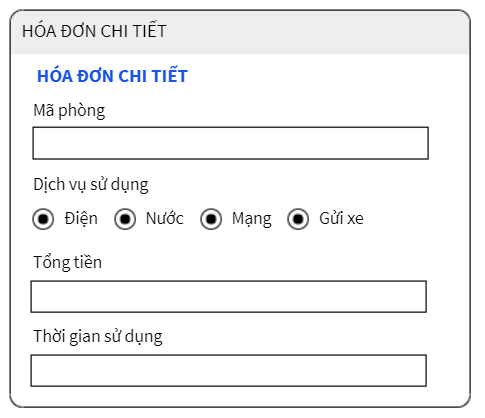
**Hello Form**

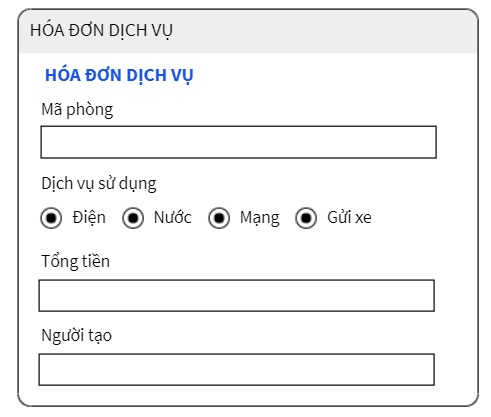
**Đăng nhập**

**Form ChangePassword**

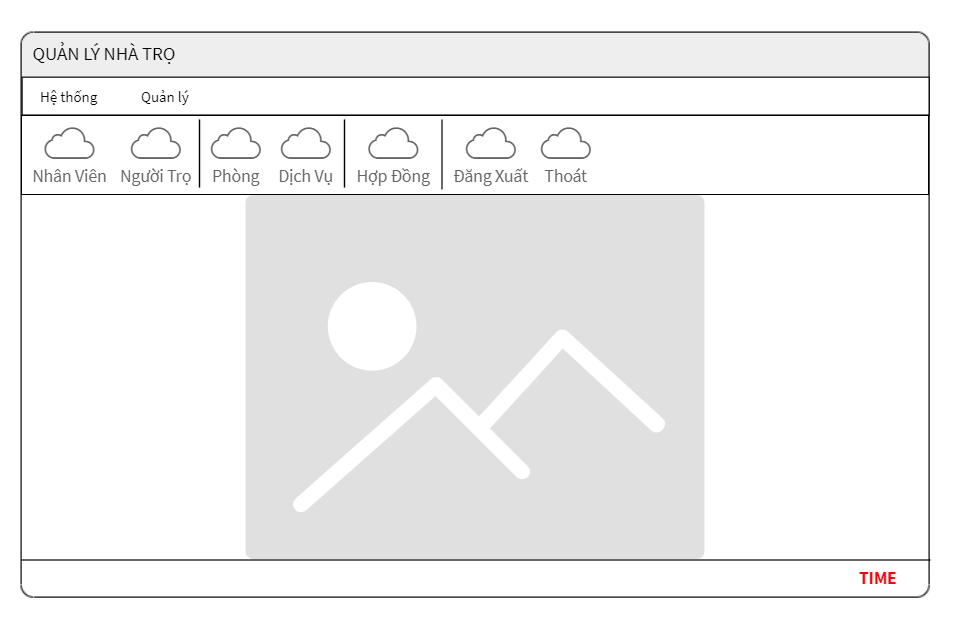
**Form Dịch vụ phòng trọ**

****

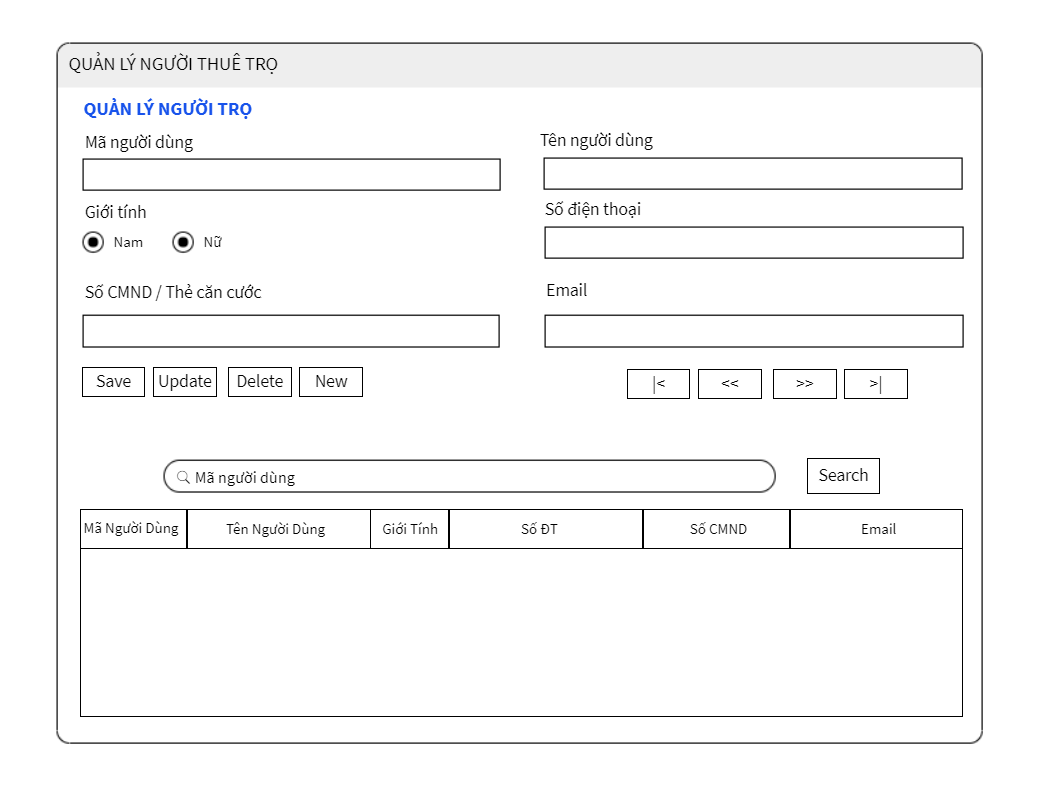
**Form Hóa đơn chi tiết**

**Form Hóa đơn dịch vụ**

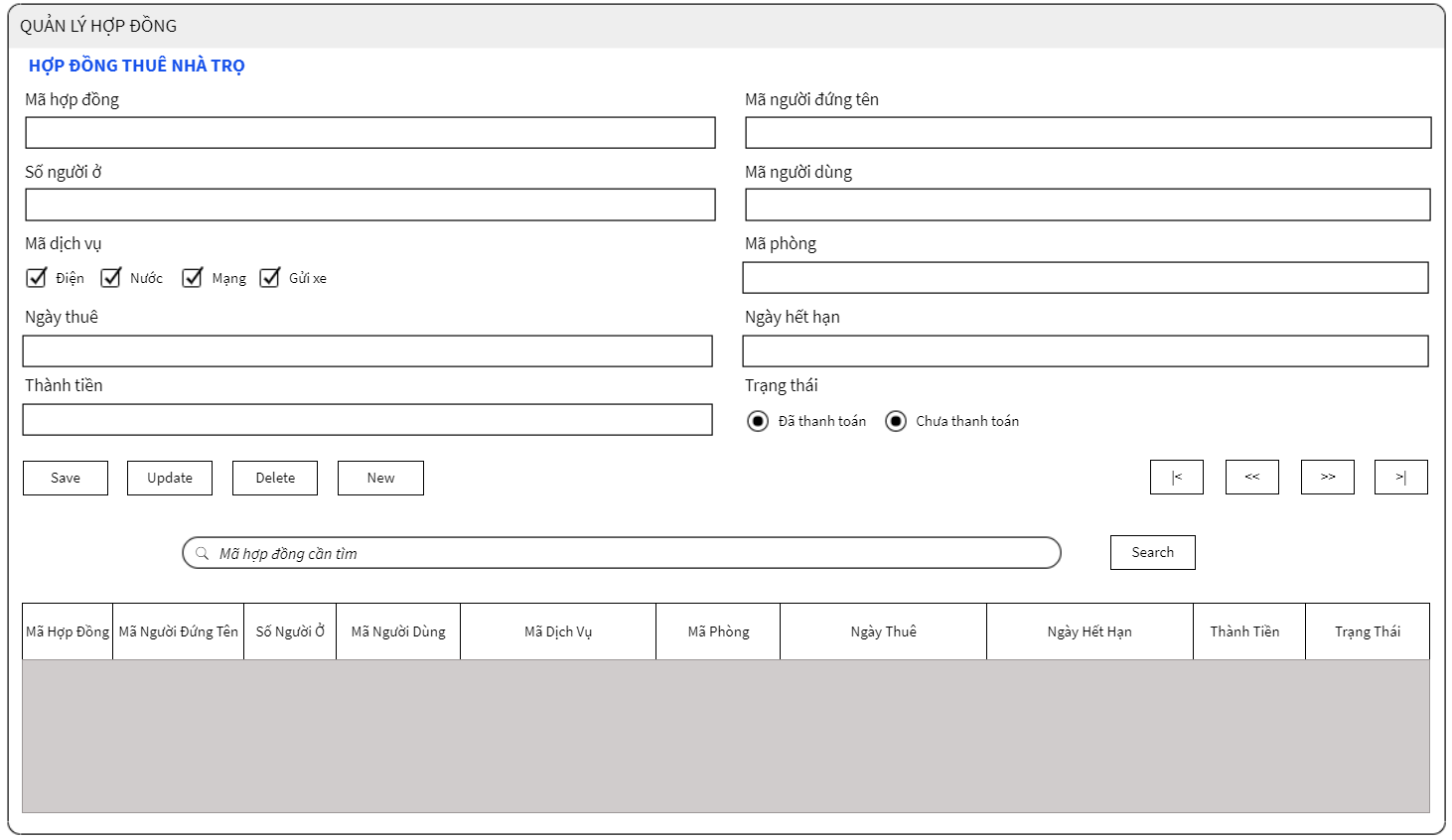
**Form Main**

****

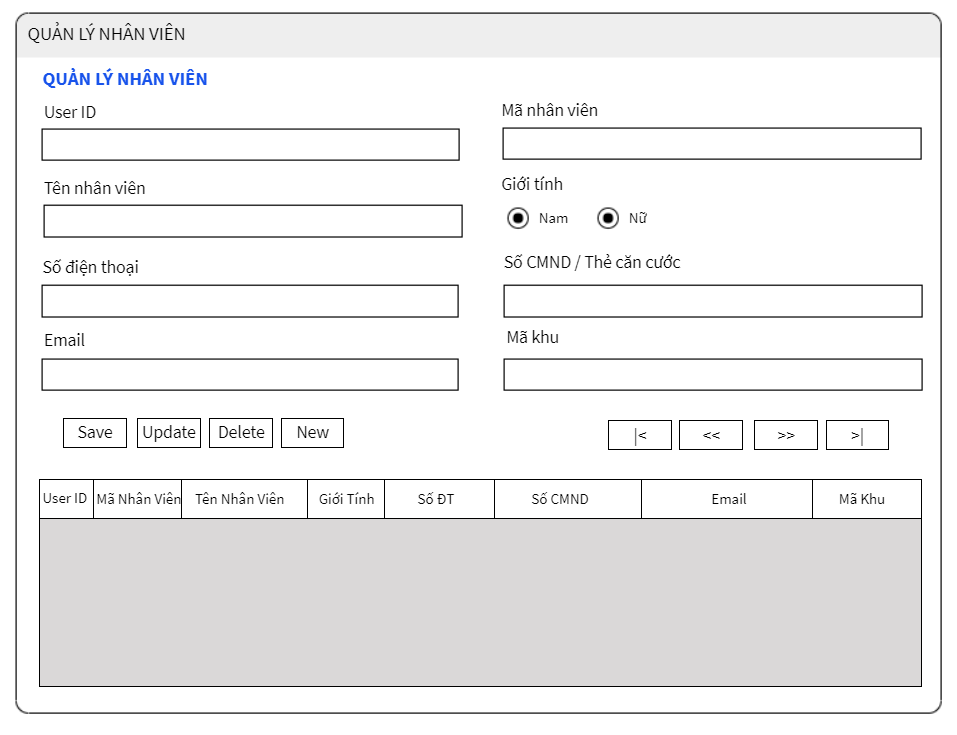
**Form Quản lý khách trọ**

****

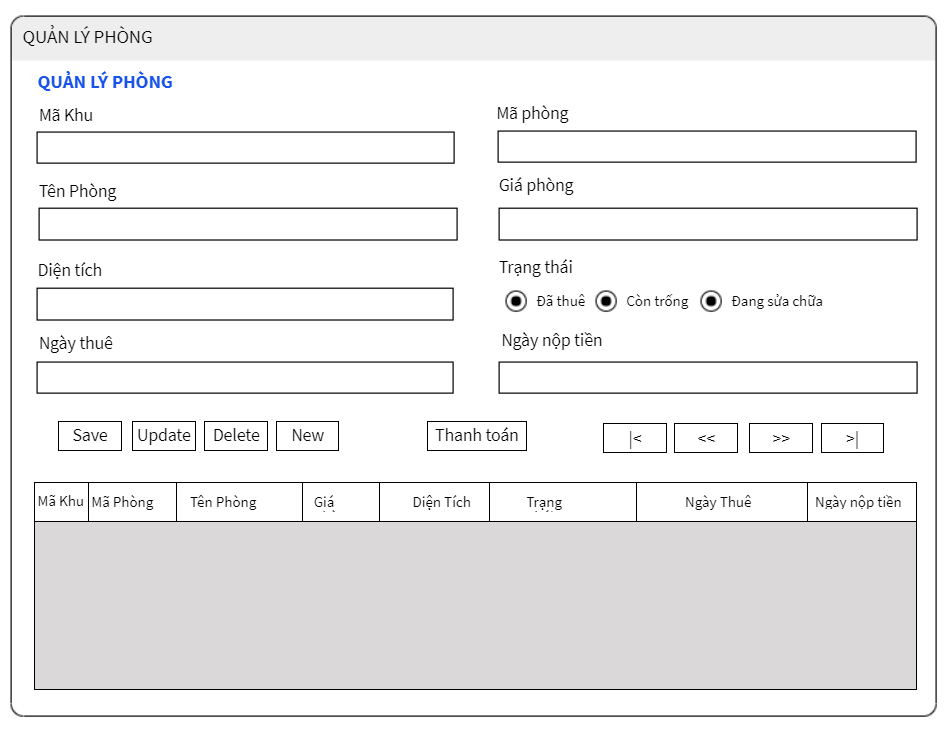
**Form Hợp đồng**

****

**Form Quản lý nhân viên**

****

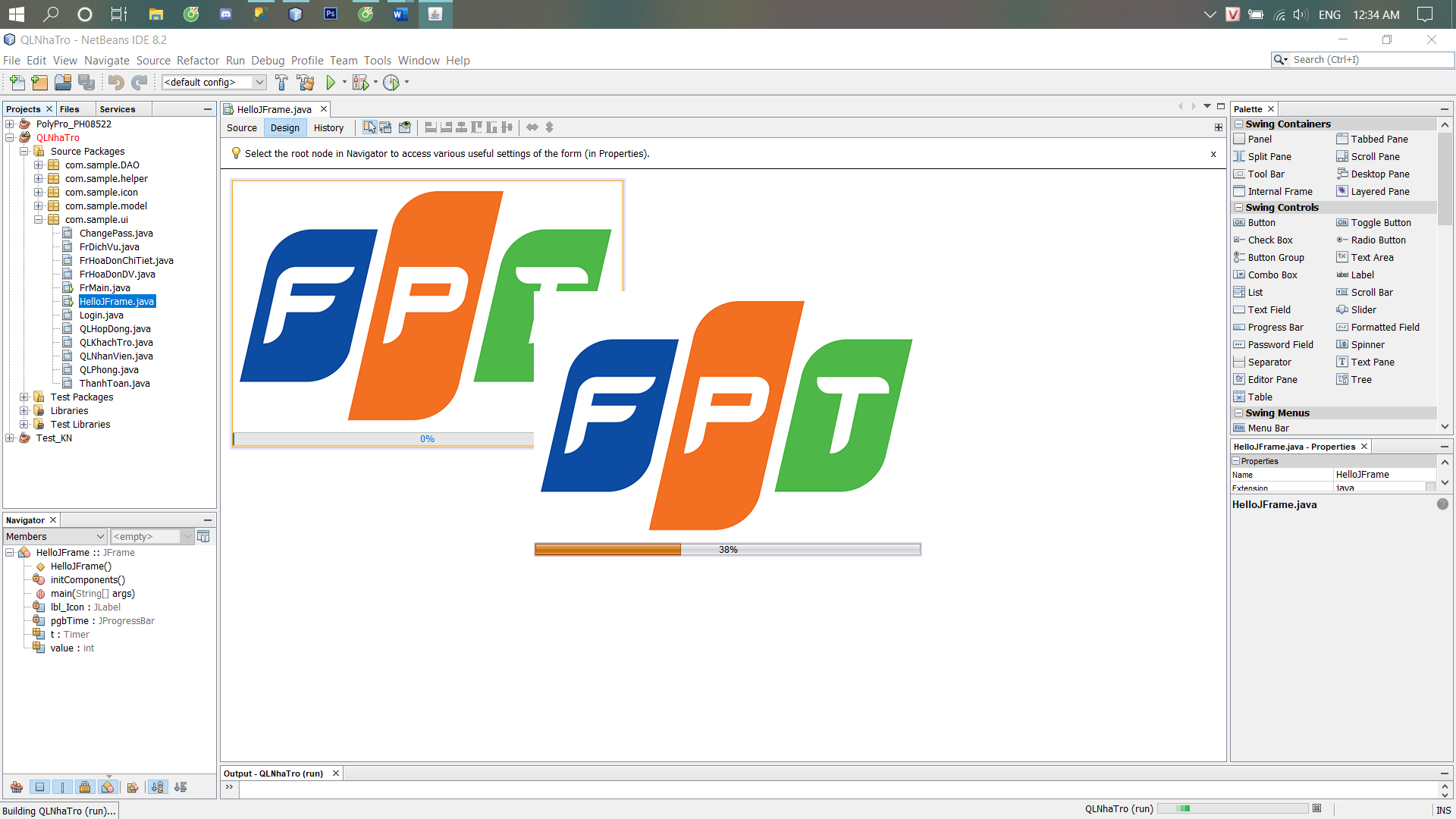
**Form Quản lý phòng**

****

**Form Thanh toán**

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

#### Màn hình chào



Các thành phần giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | ChaoJdialog | JDialog | Undecorate: True  Layout : Border Layout  PreferredSize: [514,530] |
| 2 | lblLogo | Jlabel | Icon: logo.png  Layout.Decorate: Center |
| 3 | Progress Bar | JProgressBar | Layout.Decorate: Last  StringPainted: True |

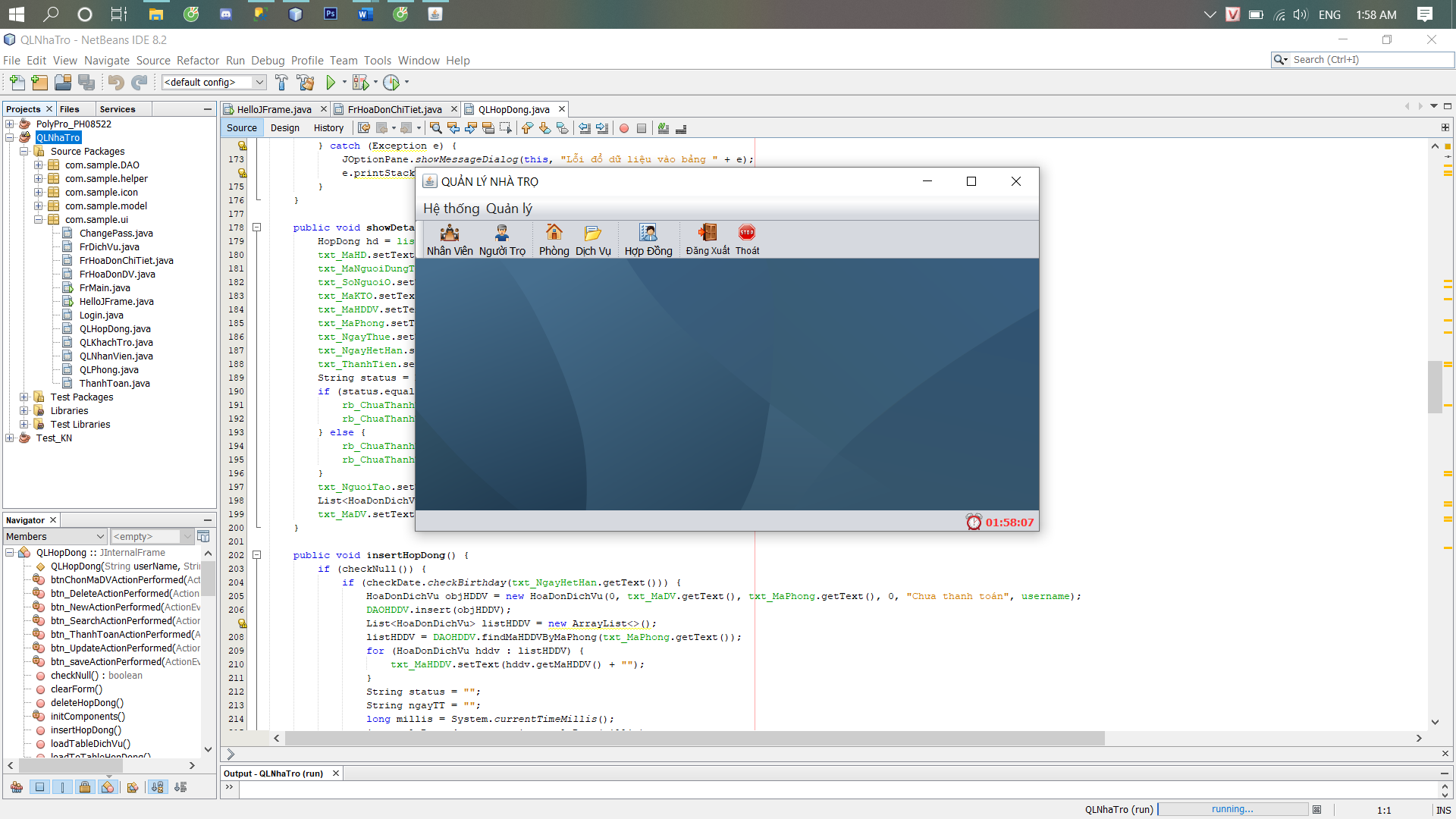
#### Màn hình đăng nhập

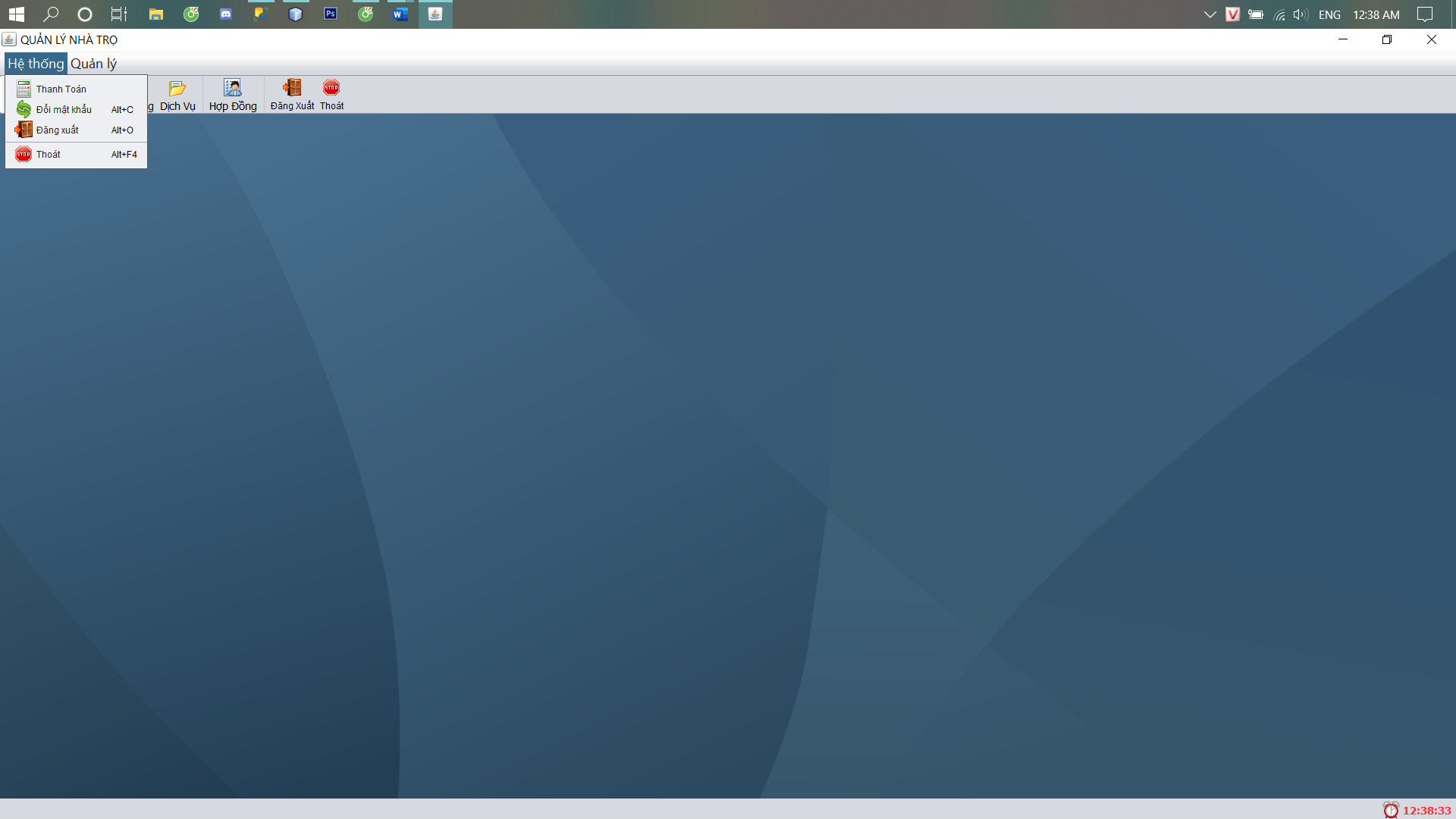
#### 

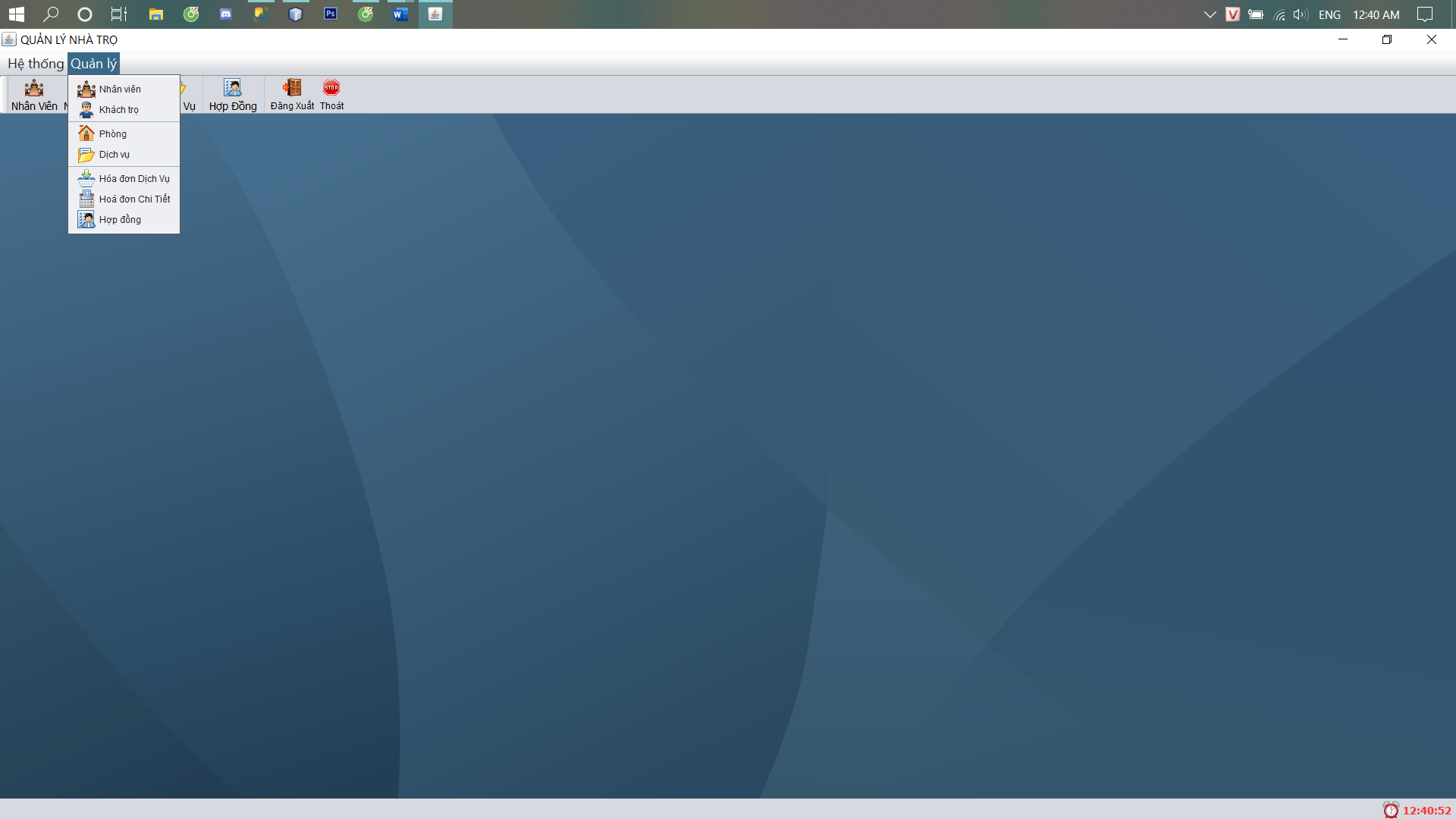
Các thành phần giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | DangNhapDialog | Jdialog | Title: HỆ THỐNG QUẢN LÝ NHÀ TRỌ  Layout: BorderLayout  defaultCloseOperation: Dispose |
| 2 | lblTitle | Jlabel | Text: ĐĂNG NHẬP  Font: Tahoma, Bold, 18  Foreground : [0,0,153] |
| 3 | lblImage | JLabel | Icon : Security.png  Layout.Direction: Before |
| 4 | pnlForm | JPanel | Layout.Derection:center |
| 5 | lblMaNV | JLabel | Text: Tên đăng nhập |
| 6 | txtMaNV | JTextField |  |
| 7 | lblMatKhau | JLabel | Text: Mật khẩu |
| 8 | txtMatKhau | JPassword |  |
| 9 | btnDangNhap | JButton | Text:Đăng nhập |
| 10 | btnKetThuc | JButton | Text:Kết thúc |

#### Màn hình chính



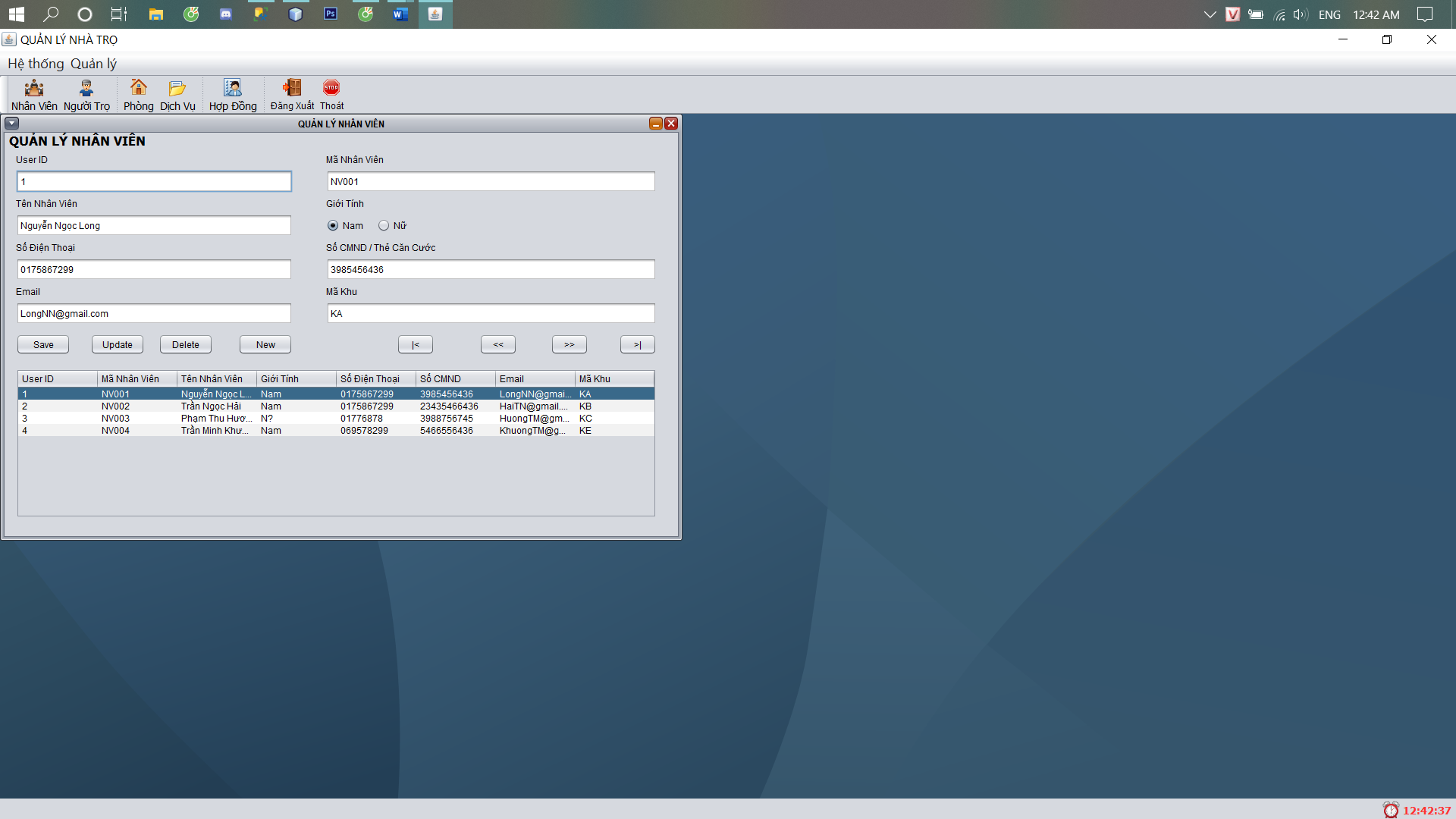




Các thành phần giao diện :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | MainJFrame | JFrame | Title: HỆ THỐNG QUẢN LÝ NHÀ TRỌ  Layout : BorderLayout |
| 2 | mnuHeThong | JMenu | Text:Hệ thống |
| 3 | mniThanhToan | JmenuItem | Text: Thanh Toán  Icon : Calculator.png |
| 4 | mni\_ChangePassword | JMenuItem | Text: Đổi mật khẩu  Icon: Refresh.png  Accelerator: Alt + C |
| 5 | mni\_LogOut | JMenuItem | Text:Đăng xuất  Icon: Log out.png  Accelerator: Alt + O |
| 6 | mni\_Exit | JMenuItem | Text:Kết thúc  Icon: Stop.png  Accelerator: Alt + F4 |
| 7 | mnuQuanLy | JMenu | Text:Quản lý |
| 8 | mni\_QLNhanVien | JMenuItem | Text: Nhân Viên  Icon: Conference.png |
| 9 | mni\_QLKhachTro | JmenuItem | Text: Khách Trọ  Icon: Boy.png |
| 10 | mni\_QLPhong | JmenuItem | Text: Phòng  Icon : Home.png |
| 11 | mni\_QLDichVu | JmenuIten | Text: Dịch vụ  Icon: Documents.png |
| 12 | mni\_HDDV | JmenuItem | Text: Hóa đơn Dịch vụ  Icon: Add to basket.png |
| 13 | mni\_HDCT | JmenuItem | Text: Hóa đơn Chi Tiết  Icon: Cash register.png |
| 14 | mni\_QLHopDong | JmenuItem | Text: Hợp đồng  Icon: Clien list.png |
| 15 | toolBar | JToolBar | Layout.Direction: first |
| 16 | btn\_QLNhanVien | JButton | Text: Nhân Viên  Icon: Conference.png  Margin: [2, 10, 2, 10] |
| 17 | btn\_QLNguoiTro | Jbutton | Text: Khách Trọ  Icon : Boy.png  Margin: [2, 10, 2, 10] |
| 18 | btn\_QLPhong | Jbutton | Text : Phòng  Icon : Home.png  Margin: [2, 10, 2, 10] |
| 19 | btn\_QL DichVu | Jbutton | Text : Dịch Vụ  Icon : Documents.png  Margin: [2, 10, 2, 10] |
| 20 | btn\_QLHopDong | Jbutton | Text : Hợp Đồng  Icon : Clien list.png  Margin: [2, 10, 2, 10] |
| 21 | btn\_LogOut | Jbutton | Text : Đăng Xuất  Icon : Log out.png  Margin: [2, 10, 2, 10] |
| 22 | btn\_Exit | Jbutton | Text : Thoát  Icon : Stop.png  Margin: [2, 10, 2, 10] |
| 23 | pnl\_Time | JPanel | Layout.Direction: Last  Layout : Border Layout |
| 24 | lbl\_Time | JLabel | Text : TIME  Icon : Alarm.png  Foreground : [255,51,51] |
| 25 | pnl\_Back | JPanel | Layout.Direction : Center  Layout : Border Layout |
| 26 | dsk\_Main | JDesktopPane |  |

#### Màn hình quản lý nhân viên



Các thành phần giao diện :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | NhanVienJInternalFrame | JInternalFrame | Title : QUẢN LÝ NHÂN VIÊN  Layout : BorderLayout  PreferredSize : [904, 578] |
| 2 | pnl\_Title | Jpanel | Layout.Direction : Fist |
| 3 | lbl\_Title | Jlabel | Text : QUẢN LÝ NHÂN VIÊN  Foreground : [0,0,153] |
| 4 | pnl\_Form | Jpanel | Layout.Direction : Last |
| 5 | lbl\_UserID | Jlabel | Text : User ID |
| 6 | txt\_UserID | JTextField |  |
| 7 | lbl\_MaNV | Jlabel | Text : Mã nhân viên |
| 8 | txt\_MaNV | JTextField |  |
| 9 | lbl\_TenNV | Jlabel | Text : Tên nhân viên |
| 10 | txt\_TenNV | JTextField |  |
| 11 | lbl\_GioiTinh | Jlabel | Text : Giới tính |
| 12 | rb\_Nam | JradioButton | Text : Nam |
| 13 | rb\_Nu | JradioButton | Text : Nữ |
| 14 | lbl\_SDT | Jlabel | Text : Số điện thoại |
| 15 | txt\_SDT | JTextField |  |
| 16 | lbl\_SoCMND | Jlabel | Text : Số CMND / Thẻ căn cước |
| 17 | txt\_SoCMND | JTextField |  |
| 18 | lbl\_Email | Jlabel | Text : Email |
| 19 | txt\_Email | JTextField |  |
| 20 | lbl\_MaKhu | Jlabel | Text : Mã khu |
| 21 | txt\_MaKhu | JTextField |  |
| 22 | btn\_Save | Jbutton | Text : Save |
| 23 | btn\_Update | Jbutton | Text : Update |
| 24 | btn\_Delete | Jbutton | Text : Delete |
| 25 | btn\_New | Jbutton | Text : New |
| 26 | btn\_Fist | Jbutton | Text : |< |
| 27 | btn\_Previous | Jbutton | Text : << |
| 28 | btn\_Next | Jbutton | Text : >> |
| 29 | btn\_Last | Jbutton | Text : >| |
| 30 | tbl\_NhanVien | Jtable |  |

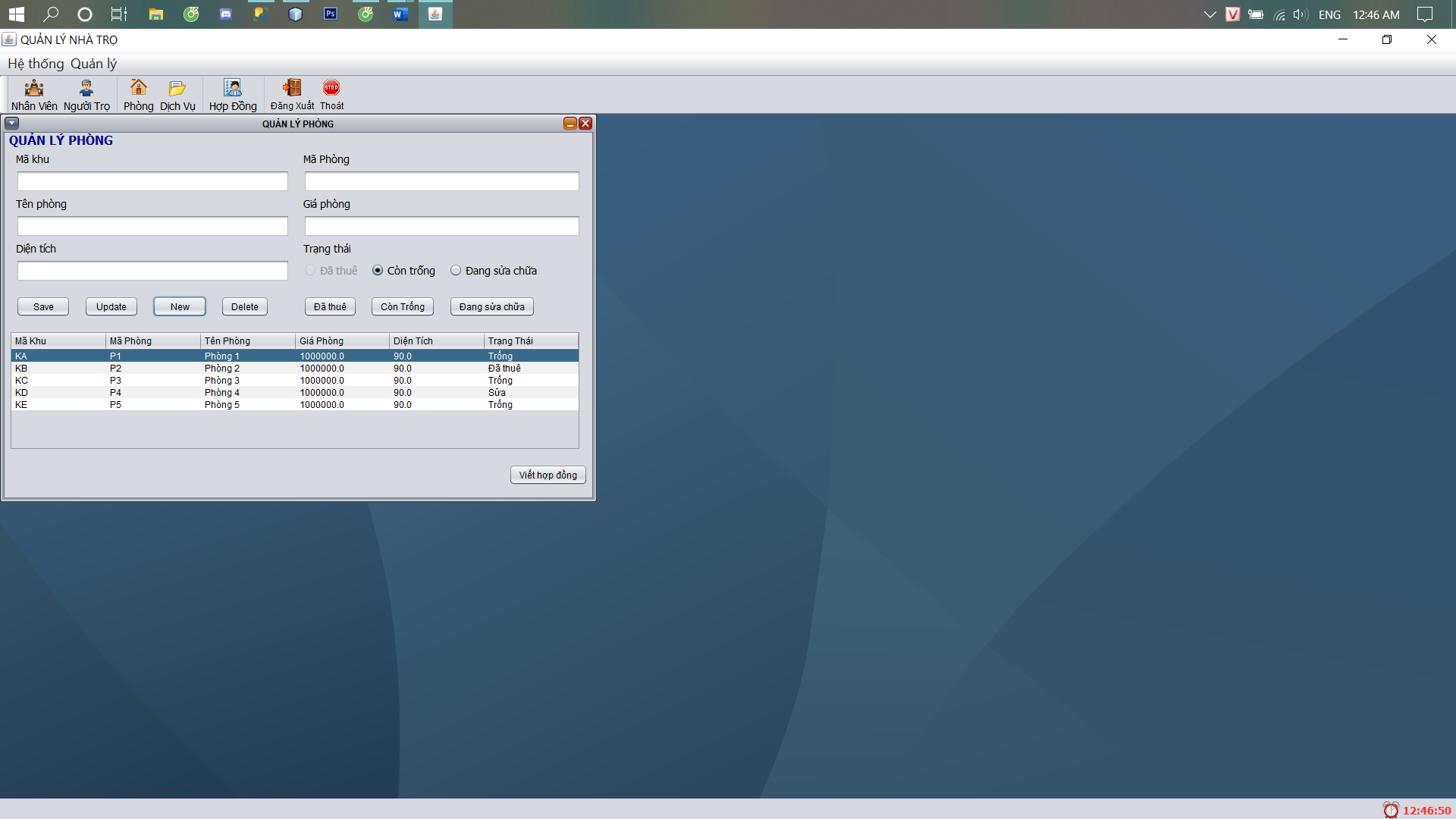
#### Màn hình quản lý khách trọ



Các thành phần giao diện :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | QLKhachTroInternalFrame | JInternalFrame | Title : QUẢN LÝ KHÁCH TRỌ  Layout : Border Layout |
| 2 | pnl\_Title | Jpanel | Layout.Direction : Fist |
| 3 | lbl\_Title | Jlabel | Text : QUẢN LÝ KHÁCH TRỌ  foreGround : [0,0,153] |
| 4 | pnl\_Form | Jpanel | Layout.Direction : Center |
| 5 | lbl\_MaKT | Jlabel | Text : Mã khách trọ |
| 6 | txt\_MaKT | JTextField |  |
| 7 | lbl\_TenKT | Jlabel | Text : Tên khách trọ |
| 8 | txt\_TenKT | JTextField |  |
| 9 | lbl\_GioiTinh | Jlabel | Text : Giới tính |
| 10 | rb\_Nam | JRadioButton | Text : Nam |
| 11 | rb\_Nu | JRadioButton | Text : Nữ |
| 12 | lbl\_SDT | Jlabel | Text : Số điện thoại |
| 13 | txt\_SDT | JtextField |  |
| 14 | lbl\_SoCMND | Jlabel | Text : Số CMND / Thẻ căn cước |
| 15 | txt\_SoCMND | JTextField |  |
| 16 | lbl\_Email | Jlabel | Text : Email |
| 17 | txt\_Email | JTextField |  |
| 18 | lbl\_MaPhongThue | Jlabel | Text : Mã phòng thuê |
| 19 | txt\_MaPhongThue | JtextField |  |
| 20 | pnl\_Button | Jpanel |  |
| 21 | btn\_Save | Jbutton | Text : Save |
| 22 | btn\_Update | Jbutton | Text : Update |
| 23 | btn\_New | Jbutton | Text : New |
| 24 | btn\_Fist | Jbutton | Text : |< |
| 25 | btn\_Previous | Jbutton | Text : << |
| 26 | btn\_Next | Jbutton | Text : >> |
| 27 | btn\_Last | Jbutton | Text : >| |
| 28 | pnl\_Search | Jpanel |  |
| 29 | txt\_Search | JtextField |  |
| 30 | btn\_Search | Jbutton | Text : Search |
| 31 | tbl\_KhachTro | JTable |  |

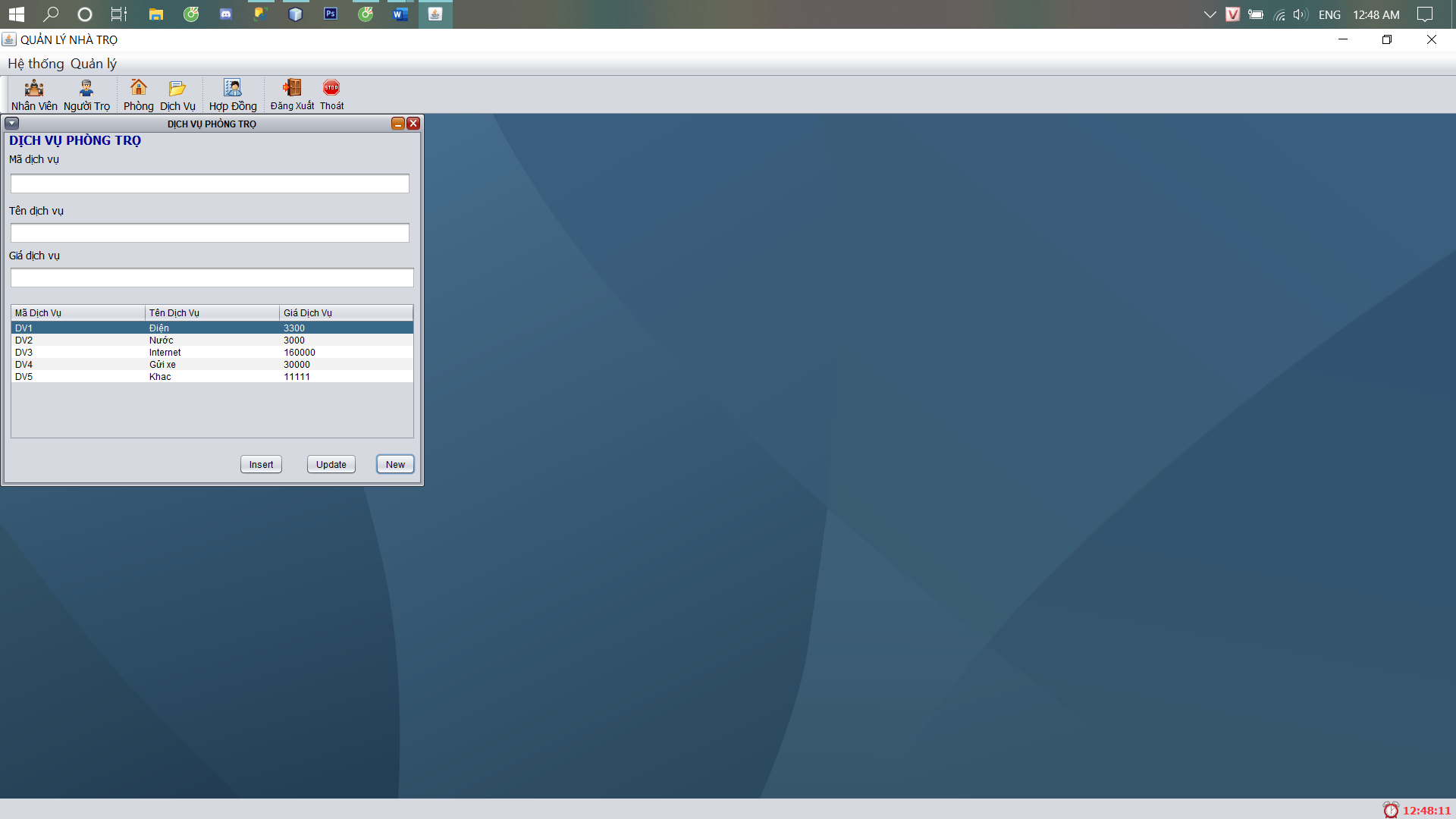
#### Màn hình quản lý phòng trọ



Các thành phần giao diện :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | QLPhongInternalFrame | JinternalFrame | Title : QUẢN LÝ PHÒNG  Layout : Border Layout  PreferredSize : [797, 533] |
| 2 | pnl\_Title | Jpanel | Layout.Direction : Fist |
| 3 | lbl\_Title | Jlabel | Text : QUẢN LÝ PHÒNG  foreGround : [0,0,153] |
| 4 | pnl\_Form | Jpanel | Layout.Direction : Last |
| 5 | lbl\_MaKhu | Jlabel | Text : Mã khu |
| 6 | txt\_MaKhu | JtextField |  |
| 7 | lbl\_MaPhong | Jlable | Text : Mã phòng |
| 8 | txt\_MaPhong | JtextField |  |
| 9 | lbl\_TenPhong | Jlabel | Text : Tên phòng |
| 10 | txt\_TenPhong | JtextField |  |
| 11 | lbl\_GiaPhong | Jlabel | Text : Gía phòng |
| 12 | txt\_GiaPhong | JtextField |  |
| 13 | lbl\_DienTich | Jlabel | Text : Diện tích |
| 14 | txt\_DienTich | JtextField |  |
| 15 | lbl\_TrangThai | Jlable | Text : Trạng thái |
| 16 | rb\_DaThue | JradioButtom | Text : Đã thuê |
| 17 | rb\_ConTrong | JradioButton | Text : Còn trống |
| 18 | rb\_DangSuaChua | JradioButton | Text : Đang sửa chữa |
| 19 | btn\_Save | Jbutton | Text : Save |
| 20 | btn\_Update | Jbutton | Text : Update |
| 21 | btn\_Delete | Jbutton | Text : Delete |
| 22 | btn\_New | Jbutton | Text : New |
| 23 | btn\_DaThue | Jbutton | Text : Đã thuê |
| 24 | btn\_ConTrong | Jbutton | Text : Còn trống |
| 25 | btn\_Fix | Jbutton | Text : Đang sửa chữa |
| 26 | btn\_ VietHD | Jbuton | Text : Viết hợp đồng |
| 27 | tbl\_QLPhong | Jtable |  |

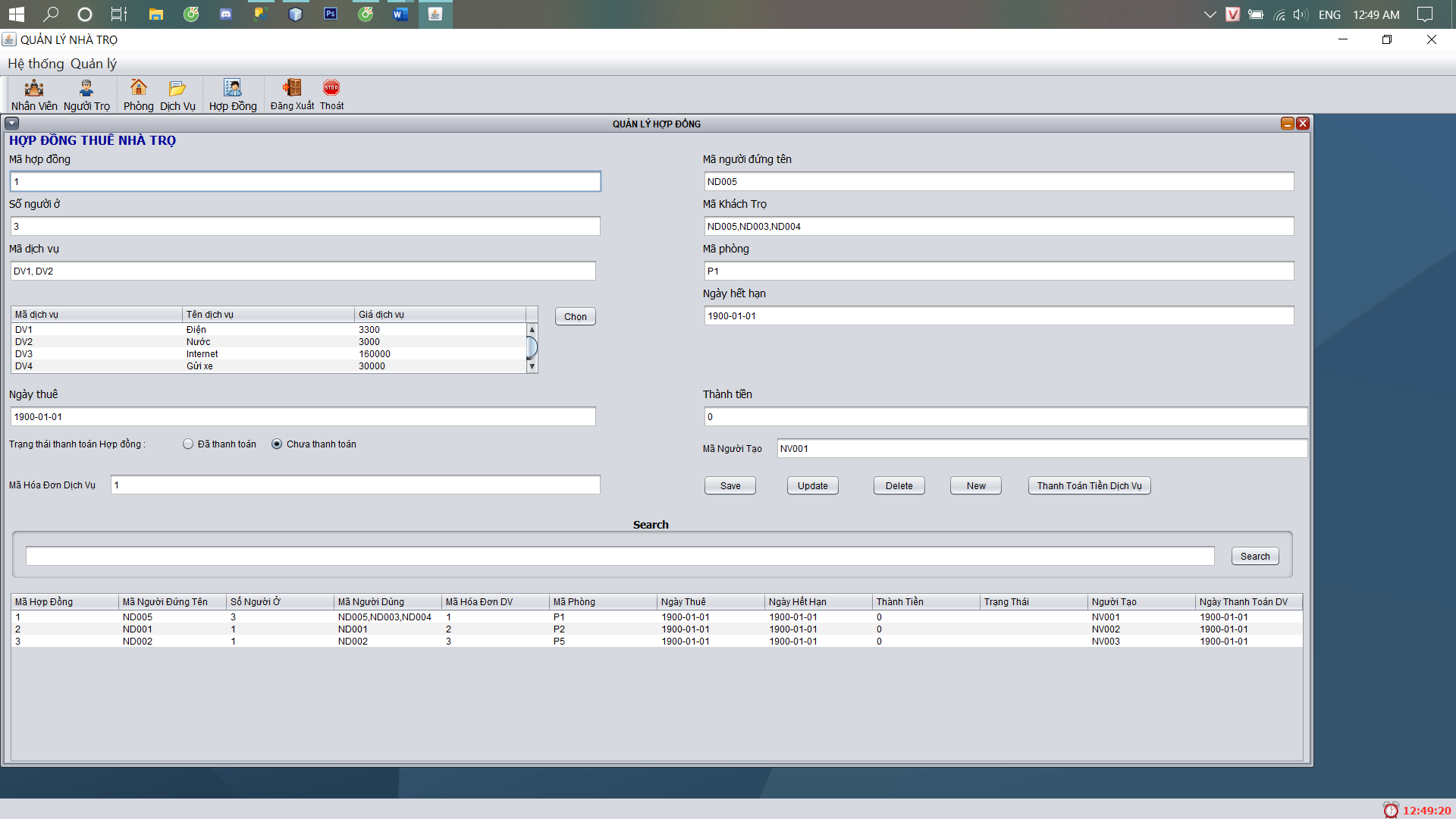
#### Màn hình quản lý dịch vụ



Các thành phần giao diện :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | DichVuInternalFrame | JInternalFrame | Title : DỊCH VỤ PHÒNG TRỌ  Layout : Border Layout  PreffedSize : [570, 515] |
| 2 | pnl\_Title | Jpanel | Layout.Direction : Fist |
| 3 | lbl\_Title | Jlabel | Text : DỊCH VỤ PHÒNG TRỌ  foreGround : [0,0,153] |
| 4 | pnl\_Form | Jpanel | Layout.Direction : Center |
| 5 | lbl\_MaDV | Jlabel | Text : Mã dịch vụ |
| 6 | txt\_MaDV | JtextField |  |
| 7 | lbl\_TenDV | Jlabel | Text: Tên dịch vụ |
| 8 | txt\_TenDV | Jtextfield |  |
| 9 | lbl\_GiaDV | Jlabel | Text : Giá dịch vụ |
| 10 | txt\_GiaDV | JtextField |  |
| 11 | btn\_Save | Jbutton | Text: Save |
| 12 | btn\_Update | Jbutton | Text : Update |
| 13 | btn\_New | Jbutton | Text : New |
| 14 | tbl\_DichVu | Jtable |  |

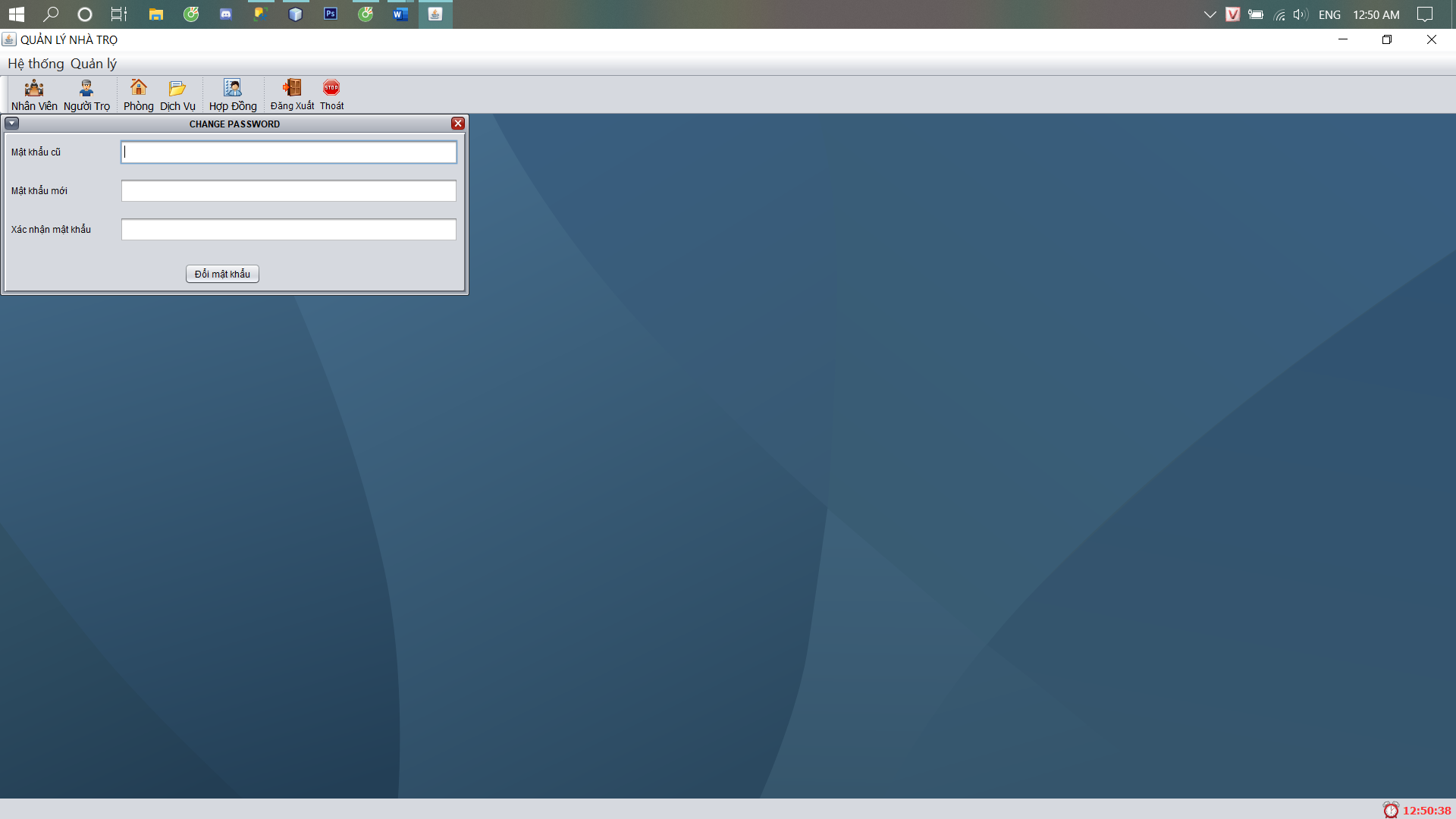
#### Màn hình quản lý hợp đồng



Các thành phần giao diện :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | QLHopDongInternalFrame | JinternalFrame | Title : QUẢN LÝ HỢP ĐỒNG  Layout : Border Layout |
| 2 | pnl\_Title | Jpanel | Layout Direction : Fist |
| 3 | lbl\_Title | Jlabel | Text : HỢP ĐỒNG THUÊ NHÀ TRỌ  ForeGround : [0,0,153] |
| 4 | pnl\_Form | Jpanel | Layout Direction : Center |
| 5 | lbl\_MaHD | Jlabel | Text : Mã hợp đồng |
| 6 | txt\_MaHD | JtextField |  |
| 7 | lbl\_MaNguoiDungTen | Jlabel | Text : Mã nguời đứng tên |
| 8 | txt\_MaNguoiDungTen | JTextField |  |
| 9 | lbl\_SoNguoiO | Jlabel | Text : Số người ở |
| 10 | txt\_SoNguoiO | JtextField |  |
| 11 | lbl\_MaKT | Jlabel | Text : Mã khách trọ |
| 12 | txt\_MaKT | JtextField |  |
| 13 | lbl\_MaDV | Jlabel | Text : Mã dịch vụ |
| 14 | txt\_MaDV | JtextField |  |
| 15 | lbl\_MaPhong | Jlabel | Text : Mã phòng |
| 16 | txt\_MaPhong | JtextField |  |
| 17 | lbl\_NgayHetHan | Jlabel | Text : Ngày hết hạn |
| 18 | txt\_NgayHetHan | JtextField |  |
| 19 | lbl\_NgayThue | Jlabel | Text : Ngày thuê |
| 20 | txt\_NgayThue | JtextField |  |
| 21 | lbl\_ThanhTiem | Jlabel | Text : Thành tiền |
| 22 | txt\_ThanhTien | JtextField |  |
| 23 | lbl\_TrangThai | Jlabel | Text : Trạng thái thanh toán hợp đồng |
| 24 | rb\_DaThanhToan | JradioButton | Text : Đã thanh toán |
| 25 | rb\_ChuaThanhToan | JradioButton | Text : Chưa thanh toán |
| 26 | lbl\_MaNguoiTao | Jlabel | Text : Mã người tạo |
| 27 | txt\_MaNguoiTao | JtextField | editable : False |
| 28 | lbl\_MaHDDV | Jlabel | Text : Mã hóa đơn dịch vụ |
| 29 | txt\_MaHDDV | JtextField | editable : False |
| 30 | tbl\_BangDV | Jtable |  |
| 31 | btn\_ChonMaDV | Jbutton | Text : Chọn |
| 32 | btn\_Save | Jbutton | Text : Save |
| 33 | btn\_Update | Jbutton | Text : Update |
| 34 | btn\_Delete | Jbutton | Text : Delete |
| 35 | btn\_New | Jbutton | Text : New |
| 36 | pnl\_Search | Jpanel | Border : Title Border |
| 37 | txt\_Search | JtextField |  |
| 38 | btn\_Search | Jbutton | Text : Search |
| 39 | tbl\_HopDong | Jtable |  |

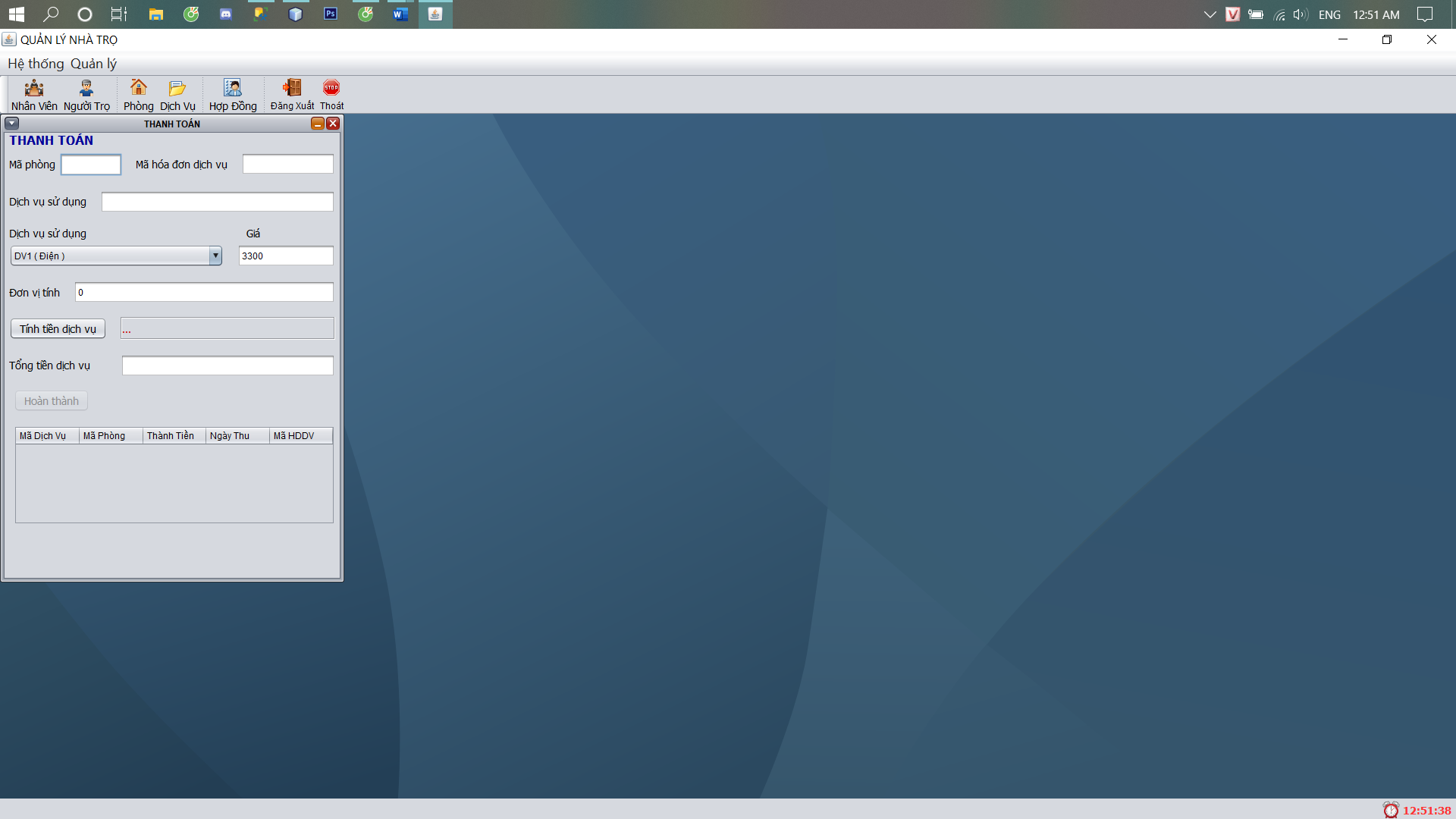
#### Màn hình đổi mật khẩu



Các thành phần giao diện :

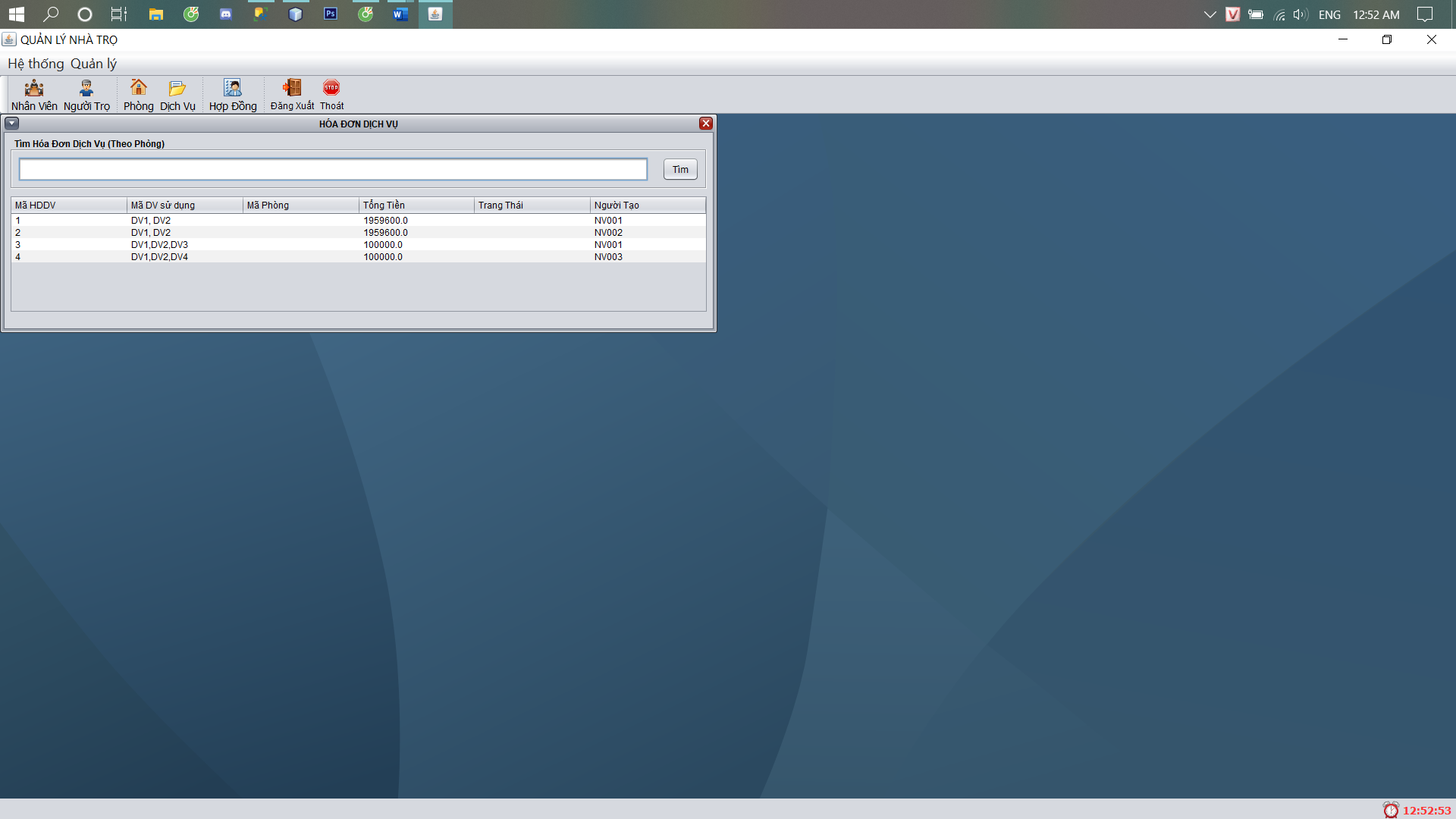
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | ChangePasswordInternalFrame | JinternalFrame | Title : CHANGE PASSWORD  Layout : Border Layout |
| 2 | pnl\_Form | Jpanel | Layout Direction : Center |
| 3 | lbl\_OldPass | Jlabel | Text : Mật khẩu cũ |
| 4 | ps\_OldPass | JpasswordField |  |
| 5 | lbl\_NewPass | Jlabel | Text : Mật khẩu mới |
| 6 | ps\_NewPass | JpasswordField |  |
| 7 | lbl\_ConfirmNewPass | Jlabel | Text : Xác nhận mật khẩu |
| 8 | ps\_ConfirmNewPass | JpasswordField |  |
| 9 | btn\_ChangePass | Jbutton | Text : Đổi mật khẩu |

#### Màn hình thanh toán



|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | ThanhToanInternalFrame | JinternalFrame | Text : THANH TOÁN  Layout : Border Layout |
| 2 | pnl\_Title | Jpanel | Layout Direction : Fist |
| 3 | lbl\_Title | Jlabel | Text : THANH TOÁN  foreGround : [0,0,153] |
| 4 | pnl\_Form | Jpanel | Layout Direction : Last |
| 5 | lbl\_MaPhong | Jlabel | Text : Mã phòng |
| 6 | txt\_MaPhong | JtextField | Editable : False |
| 7 | lbl\_MaHDDV | Jlabel | Text : Mã hóa đơn dịch vụ |
| 8 | txt\_MaHDDV | JtextField | Editable : False |
| 9 | lbl\_DVSD | Jlabel | Text : Dịch vụ sử dụng |
| 10 | txt\_DVSD | JtextField | Editable : False |
| 11 | lbl\_DVSDThem | Jlabel | Text : Dịch vụ sử dụng thêm |
| 12 | cbx\_DVSD | JcomboBox |  |
| 13 | lbl\_GiaDVSD | Jlabel | Text : Giá |
| 14 | txt\_GiaDVSD | JtextField | Editable : False |
| 15 | lbl\_DVTinh | Jlabel | Text : Đơn vị tính |
| 16 | txt\_DVTinh | JtextField |  |
| 17 | btn\_ThanhTien | Jbutton | Text : Tính tiền dịch vụ |
| 18 | lbl\_ShowTien | Jlabel | Text : …  Foreground : [204,0,0] |
| 19 | lbl\_TongTienDV | Jlabel | Text : Tổng tiền dịch vụ |
| 20 | txt\_TongTien | JtextField | Editable : False |
| 21 | btn\_ThuTien | Jbutton | Text : Hoàn thành |
| 22 | tbl\_ThanhToan | Jtbale |  |

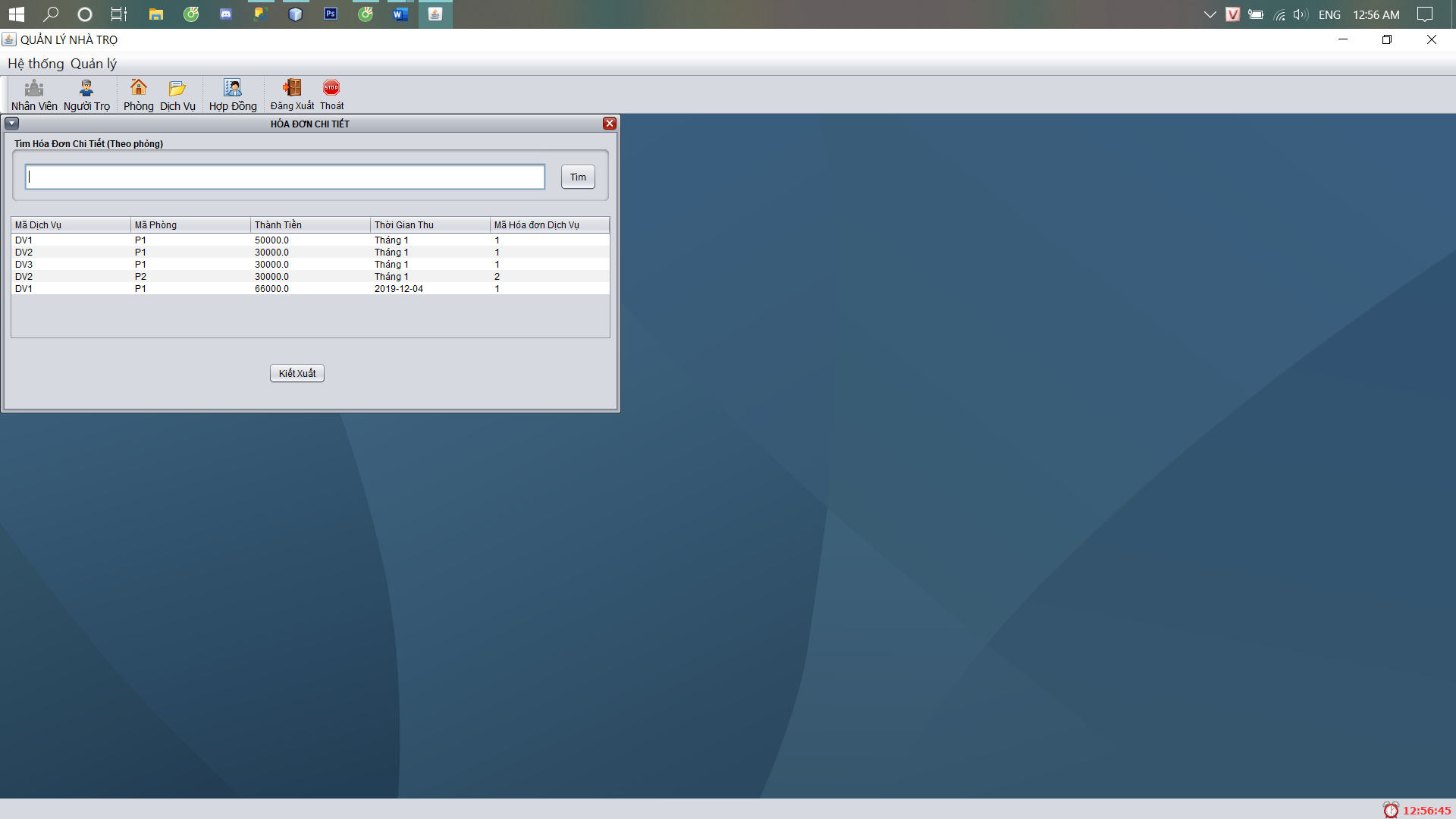
#### Màn hình hóa đơn dịch vụ



Các thành phần giao diện :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | HDDVInternalFrame | JinternalFrame | Title : HÓA ĐƠN DỊCH VỤ |
| 2 | pnl\_Search | JPanel | Border : Title Border |
| 3 | txt\_MaPhong | JtextField |  |
| 4 | btn\_Tim | Jbutton | Text : Tìm |
| 5 | tbl\_HDDV | Jtable |  |

#### Màn hình hóa đơn chi tiết



|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | HDCTInternalFrame | JinternalFrame | Title : HÓA ĐƠN CHI TIẾT |
| 2 | pnl\_Search | Jbutton | Border : Title Border |
| 3 | txt\_MaPhong | JtextField |  |
| 4 | btn\_Tim | Jbutton | Text : Tìm |
| 5 | tbl\_HDCT | Jtable |  |
| 6 | btn\_KietXuat | Jbutton | Text : Kiết xuất |

# Thực hiện viết mã

## Viết mã tạo CSDL

|  |
| --- |
| create database QLNhaTro  go  use QLNhaTro  go  create table **Users**(  userid int not null identity primary key,  username varchar(50) not null,  pass varchar(50) not null,  roles varchar(10) not null  )  create table **KhuNha**(  makhu varchar(10) not null primary key,  tenkhu nvarchar(100) not null  )  create table **DichVu**(  madv varchar(10) not null primary key,  tendv nvarchar(100) not null,  gia float not null default 1  )  create table **Phong**(  maphong varchar(10) not null primary key,  tenphong nvarchar(100) not null,  gia float not null,  dientich float not null,  trangthai nvarchar(100) not null,  makhu varchar(10) not null,  FOREIGN KEY (makhu) REFERENCES KhuNha(makhu)  )  create table **KhachTro**(  makt varchar(10) not null primary key,  tenkt nvarchar(100) not null,  gioitinh nvarchar(5) not null,  sodt varchar(15) not null,  socmt varchar(15) not null,  email nvarchar(100) not null,  maphong varchar(10) not null,  foreign key (maphong) references Phong(maphong)  )  create table **NhanVien**(  manv varchar(10) not null primary key,  userid int not null unique,  tennv nvarchar(100) not null,  gioitinh nvarchar(5) not null,  sodt varchar(15) not null,  socmt varchar(15) not null,  email nvarchar(100) not null,  makhu varchar(10) not null,  foreign key (userid) REFERENCES Users(userid),  foreign key (makhu) REFERENCES KhuNha(makhu)  )  create table **HoaDonDV**(  mahddv int identity not null primary key,  sudung nvarchar(255) not null,  maphong varchar(10) not null,  tongtien float not null,  trangthai nvarchar(50) null,  nguoitao varchar(10) not null,  foreign key (nguoitao) REFERENCES NhanVien(manv),  foreign key (maphong) REFERENCES Phong(maphong)  )  create table **HoaDonChiTiet**(  mahdct int identity not null primary key,  dichvu varchar(10) not null,  maphong varchar(10) not null,  thanhtien float not null,  thoigian nvarchar(50) not null,  nguoitao varchar(10) not null,  mahddv int not null,  foreign key (nguoitao) REFERENCES NhanVien(manv),  foreign key (dichvu) REFERENCES DichVu(madv),  foreign key (mahddv) REFERENCES HoaDonDV(mahddv),  foreign key (maphong) REFERENCES Phong(maphong)  )  create table **HopDong**(  mahd int identity not null primary key,  nguoidaidien varchar(10) not null,  songuoi int not null default 1,  nguoio nvarchar(200) not null,  mahddv int not null unique,  maphong varchar(10) not null,  ngaythue datetime not null default getdate(),  ngayhethan datetime not null default getdate(),  thanhtien float not null default 0,  trangthai varchar(10) null,  nguoitao varchar(10) not null,  foreign key (nguoidaidien) REFERENCES KhachTro(makt),  foreign key (nguoitao) REFERENCES NhanVien(manv),  foreign key (maphong) REFERENCES Phong(maphong),  foreign key (mahddv) REFERENCES HoaDonDV(mahddv)  )  Go  **Mã hóa password của user**  *update Users set pass = EncryptByPassPhrase('123', convert(varchar(100),pass) )*  **Mã hóa lại password của user hỗ trợ khi viết lập trình nút login của form login**  *Select userid, convert(varchar(10),convert(varchar(100),decryptbypassphrase('123',pass) )) as*  *mk from Users* |

### Tạo CSDL

### SQL truy vấn và thao tác

#### User

#### SQL đối với Bảng 2

### Các thủ tục

#### User

|  |
| --- |
| --chọn all user  create proc sp\_selectAllUsers  as  begin  select \* from Users  end;  --Select login  create proc sp\_Login  (@username varchar(50),  @pass varchar(50)  )  as  begin  select \* from Users where username = @username and pass = @pass  end  --chọn User  create proc sp\_selectUser  (@username varchar(50))  as  begin  select \* from Users where username = @username  end  --giả mã pass user chọn  create proc sp\_convertPasstoString  (@username varchar(50))  as  begin  update Users set pass = convert(varchar(10),convert(varchar(100),decryptbypassphrase('abc',pass) )) where username = @username  end  --mã hóa pass user chọn  create proc sp\_convertStringToPass  (@username varchar(50))  as  begin  update Users set pass = EncryptByPassPhrase('abc', convert(varchar(100),pass)) where username = @username  end  --Thêm user  create proc sp\_insertUser  (  @username varchar(50),  @pass varchar(50),  @roles varchar(10)  )  as  begin  insert into Users values(@username, @pass,@roles)  end  --Xóa user  create proc sp\_deleteUser  (  @userid int  )  as  begin  delete from Users where userid = @userid  end  --Đổi Mật Khẩu  create proc sp\_changePass  (  @username varchar(50),  @pass varchar(50)  )  as  begin  update Users set pass = @pass where username = @username  end |

#### Phòng

|  |
| --- |
| --chọn All Phòng  create proc sp\_selectAllRoom  as  begin  select \* from Phong  end;  --Thêm phòng  create proc sp\_insertRoom  (  @maphong varchar(10),  @tenphong nvarchar(100),  @gia float,  @dientich float,  @trangthai nvarchar(100),  @makhu varchar(10),  @ngaythanhtoandv date  )  as  begin  insert into Phong values(@maphong, @tenphong, @gia, @dientich, @trangthai, @makhu, @ngaythanhtoandv)  end  --Xóa phòng  create proc sp\_deleteRoom  (  @maphong varchar(10)  )  as  begin  delete from Phong where maphong = @maphong  end  --Sửa phòng  create proc sp\_updateRoom  (  @maphong varchar(10),  @gia float,  @trangthai nvarchar(100)  )  as  begin  update Phong set gia = @gia, trangthai =@trangthai where maphong = @maphong  end  --Tìm phòng  create proc sp\_findNVBySTT  (@trangthai nvarchar(100))  as  begin  select \* from Phong where trangthai = @trangthai  end |

#### Nhân viên

|  |
| --- |
| --Chọn All NV  create proc sp\_selectAllNV  as  begin  select \* from NhanVien where trangthai != 'deleted'  end;  --Thêm NV  create proc sp\_insertNV  (  @manv varchar(10),  @userid int,  @tennv nvarchar(100),  @gioitinh nvarchar(5),  @sodt varchar(15),  @socmt varchar(15),  @email nvarchar(100),  @makhu varchar(10),  @trangthai nvarchar(100)  )  as  begin  insert into NhanVien values(@manv,@userid,@tennv,@gioitinh,@sodt,@socmt,@email,@makhu,@trangthai)  end  --Sửa NV  create proc sp\_updateNV  (  @manv varchar(10),  @sodt varchar(15),  @socmt varchar(15),  @email nvarchar(100),  @makhu varchar(10)  )  as  begin  update NhanVien set sodt = @manv, socmt= @socmt ,email = @email, makhu = @makhu where manv= @manv  end  --Xóa NV  create proc sp\_deleteNV  (  @manv varchar(10)  )  as  begin  update NhanVien set trangthai= 'deleted' where manv = @manv  end  --Tìm NV  drop proc sp\_findNVByMANV  create proc sp\_findNVByMANV  (@manv varchar(10))  as  begin  select \* from NhanVien where manv = @manv and trangthai != 'deleted'  end |

#### 

#### Khách trọ

|  |
| --- |
| --Chọn all KT  create proc sp\_selectAllKT  as  begin  select \* from KhachTro where trangthai != 'deleted'  end;  --Thêm KT  create proc sp\_insertKT  (  @makt varchar(10),  @tenkt nvarchar(100),  @gioitinh nvarchar(5),  @sodt varchar(15),  @socmt varchar(15),  @email nvarchar(100),  @maphong varchar(10),  @trangthai nvarchar(100)  )  as  begin  insert into KhachTro values(@makt,@tenkt,@gioitinh,@sodt,@socmt,@email,@maphong,@trangthai)  end  --Sửa KT  create proc sp\_updateKT  (  @makt varchar(10),  @sodt varchar(15),  @socmt varchar(15),  @email nvarchar(100)  )  as  begin  update KhachTro set sodt = @sodt, socmt= @socmt,email = @email where makt= @makt  end  --Xóa KT  create proc sp\_deleteKT  (  @makt varchar(10)  )  as  begin  update KhachTro set trangthai = 'deleted' where makt = @makt  end  --Tìm KT  drop proc sp\_findKTByMAKT  create proc sp\_findKTByMAKT  (@makt varchar(10))  as  begin  select \* from KhachTro where makt = @makt and trangthai != 'deleted'  end |

#### Dịch vụ

|  |
| --- |
| --Chọn tất cả DV  create proc sp\_selectAllDV  as  begin  select \* from DichVu  end  --Thêm KT  create proc sp\_insertDV  (  @madv varchar(10),  @tendv nvarchar(100),  @gia float  )  as  begin  insert into DichVu values(@madv, @tendv, @gia)  end  --Sửa DV  create proc sp\_updateDV  (  @madv varchar(10),  @gia float  )  as  begin  update DichVu set gia = @gia where madv = @madv  end  --Xóa DV  create proc sp\_deleteDV  (  @madv varchar(10)  )  as  begin  delete from DichVu where madv = @madv  end |

#### Hợp đồng

|  |
| --- |
| -- Chọn All HD  create proc sp\_selectAllHD  as  begin  select \* from HopDong where trangthai !='deleted'  end  -- thêm hợp đồng  drop proc sp\_insertHD  create proc sp\_insertHD  (  @nguoidaidien varchar(10) ,  @songuoi int ,  @nguoio nvarchar(200) ,  @mahddv int ,  @maphong varchar(10) ,  @ngaythue date ,  @ngayhethan date ,  @thanhtien float,  @trangthai varchar(10),  @nguoitao varchar(10) ,  @madv varchar(10),  @ngaythanhtoandv date  )  as  begin  insert into HopDong values(@nguoidaidien , @songuoi , @nguoio, @mahddv, @maphong ,@ngaythue , @ngayhethan ,@thanhtien, @trangthai, @nguoitao, @madv,@ngaythanhtoandv )  update Phong set trangthai = N'Đã thuê' where maphong = @maphong  end  --Xóa hợp đồng  drop proc sp\_deleteHD  create proc sp\_deleteHD  (  @mahd int  )  as  begin  update HopDong set trangthai = 'deleted' where mahd= @mahd  update Phong set trangthai = N'Trống' where maphong= (select maphong from HopDong where mahd= @mahd)  update KhachTro set trangthai = N'deleted' where maphong = (select maphong from HopDong where mahd= @mahd)  end  select \* from NhanVien  --Update Hợp đồng  drop proc sp\_updateHD  create proc sp\_updateHD  (  @mahd int,  @songuoi int ,  @nguoio nvarchar(200),  @ngayhethan date,  @thanhtien float,  @trangthai varchar(10)  )  as  begin  update HopDong set songuoi = @songuoi, nguoio = @nguoio, ngayhethan = @ngayhethan, thanhtien = @thanhtien, trangthai = @trangthai where mahd=@mahd  end  --Tìm Hợp đồng  drop proc sp\_findHDByMaPhong  create proc sp\_findHDByMaPhong  (  @maphong varchar(10)  )  as  begin  select \* from HopDong where maphong = @maphong  end |

#### Hóa đơn dịch vụ

|  |
| --- |
| --Chọn all maHDDV  create proc sp\_selectAllHDDV  as  begin  select \* from HoaDonDV where trangthai !='deleted'  end  --Thêm Hóa Đơn DV  create proc sp\_insertHDDV  (  @sudung nvarchar(255),  @maphong varchar(10),  @tongtien float,  @trangthai nvarchar(50) ,  @nguoitao varchar(10)  )  as  begin  insert into HoaDonDV values(@sudung,@maphong,@tongtien,@trangthai,@nguoitao)  end  --update HDDV  drop proc sp\_insertHDDV  create proc sp\_updateHDDV  (  @sudung nvarchar(255),  @maphong varchar(10),  @tongtien float,  @trangthai nvarchar(50)  )  as  begin  update HoaDonDV set sudung = @sudung, tongtien= @tongtien, trangthai = @trangthai where maphong = @maphong  end  --Tìm theo mã phòng  create proc sp\_selectMaHDDVByMaPhong(  @maphong varchar(10)  )  as  begin  select \* from HoaDonDV where maphong = @maphong  end  ---7. HÓA ĐƠN CHI TIẾT  --Chọn all HDCT  create proc sp\_selectAllHDCT  as  begin  select \* from HoaDonChiTiet  end  ---Thêm HDCT  drop proc sp\_insertHDCT  create proc sp\_insertHDCT  (  @dichvu varchar(10),  @maphong varchar(10),  @thanhtien float,  @thoigian nvarchar(50),  @mahddv int  )  as  begin  insert into HoaDonChiTiet values(@dichvu,@maphong,@thanhtien,@thoigian,@mahddv)  end  --Tìm HDCT theo mã phòng  create proc sp\_selectMaHDCTByMaPhong(  @maphong varchar(10)  )  as  begin  select \* from HoaDonChiTiet where maphong = @maphong  end |

## Lập trình JDBC

### Lớp hỗ trợ

#### Lớp JDBCHelper

|  |
| --- |
| package com.sample.helper;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  public class JdbcHelper<JBDC> {  private static String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  private static String  url ="jdbc:sqlserver://DESKTOPFTMS285\\SQLEXPRESS:1433;databaseName=QLNhaTro";  private static String user = "sa";  private static String pass = "songlong";  static {  try {  Connection cnt = DriverManager.getConnection(url, user, pass);  } catch (Exception e) {  throw new RuntimeException(e);  }  }  public static PreparedStatement prepareStatement(String sql, Object... args) throws SQLException {  PreparedStatement ptsm = null;  Connection cnt = DriverManager.getConnection(url, user, pass);  if (sql.trim().startsWith("{")) {  ptsm = cnt.prepareCall(sql);  } else {  ptsm = cnt.prepareStatement(sql);  }  for (int i = 0; i < args.length; i++) {  ptsm.setObject(i + 1, args[i]);  }  return ptsm;  }  ;  public static void executeUpdate(String sql, Object... args) {  try {  PreparedStatement ptsm = prepareStatement(sql, args);  try {  ptsm.executeUpdate();  } finally {  try {  ptsm.close();  } catch (Exception e) {  throw new RuntimeException(e);  }  }  } catch (Exception e) {  throw new RuntimeException(e);  };  }  public static ResultSet executeQuery(String sql, Object... args) {  try {  PreparedStatement ptsm = prepareStatement(sql, args);  return ptsm.executeQuery();  } catch (Exception e) {  throw new RuntimeException(e);  }  }  } |

#### Lớp checkDate

|  |
| --- |
| package com.sample.helper;  import java.util.Date;  import java.util.Calendar;  public class checkDate {  public static boolean checkBirthday(String n) {  if (n == null || n.length() < 10) {  return false;  } else {  boolean flag = true;  String dd, mm, yyyy;  int d, m, y;  yyyy = n.substring(0, 4);  mm = n.substring(5, 6);  dd = n.substring(8, 9);  try {  Date dat = new Date();  Calendar date = Calendar.getInstance();  date.setTime(dat);  int y1 = date.get(Calendar.YEAR);  int m1 = date.get(Calendar.MONTH);  int d1 = date.get(Calendar.DAY\_OF\_MONTH);  m = Integer.parseInt(mm);  d = Integer.parseInt(dd);  y = Integer.parseInt(yyyy);  if ((d < 1) && (d > 31)) {  flag = false;  }  if ((m < 1) && (m > 12)) {  flag = false;  }  if (y < 1) {  flag = false;  }  if (y1 < y) {  flag = false;  }  } catch (Exception e) {  flag = false;  }  return flag;  }  }  } |

### Model class - Các lớp mô tả dữ liệu

#### DichVu Model

|  |
| --- |
| package com.sample.model;  public class DichVu {  String maDV, tenDV;  int gia;  public DichVu() {  }  public DichVu(String maDV, String tenDV, int gia) {  this.maDV = maDV;  this.tenDV = tenDV;  this.gia = gia;  }  public String getMaDV() {  return maDV;  }  public void setMaDV(String maDV) {  this.maDV = maDV;  }  public String getTenDV() {  return tenDV;  }  public void setTenDV(String tenDV) {  this.tenDV = tenDV;  }  public int getGia() {  return gia;  }  public void setGia(int gia) {  this.gia = gia;  }  @Override  public String toString() {  return maDV + " ( " + tenDV+" )" ;  }  } |

#### HoaDonChiTiet Model

|  |
| --- |
| package com.sample.model;  public class HoaDonChiTiet {  private int maHDCT;  private String maDVSD, maPhong;  private float thanhTien;  private String thoiGian;  private int maHDDV;  public HoaDonChiTiet() {  }  public HoaDonChiTiet(int maHDCT, String maDVSD, String maPhong, float thanhTien, String thoiGian, int maHDDV) {  this.maHDCT = maHDCT;  this.maDVSD = maDVSD;  this.maPhong = maPhong;  this.thanhTien = thanhTien;  this.thoiGian = thoiGian;  this.maHDDV = maHDDV;  }  public int getMaHDCT() {  return maHDCT;  }  public void setMaHDCT(int maHDCT) {  this.maHDCT = maHDCT;  }  public String getMaDVSD() {  return maDVSD;  }  public void setMaDVSD(String maDVSD) {  this.maDVSD = maDVSD;  }  public String getMaPhong() {  return maPhong;  }  public void setMaPhong(String maPhong) {  this.maPhong = maPhong;  }  public float getThanhTien() {  return thanhTien;  }  public void setThanhTien(float thanhTien) {  this.thanhTien = thanhTien;  }  public String getThoiGian() {  return thoiGian;  }  public void setThoiGian(String thoiGian) {  this.thoiGian = thoiGian;  }  public int getMaHDDV() {  return maHDDV;  }  public void setMaHDDV(int maHDDV) {  this.maHDDV = maHDDV;  }  } |

#### HoaDonDichVu Model

|  |
| --- |
| package com.sample.model;  public class HoaDonDichVu {  private int maHDDV;  private String suDung, maPhong;    private float tongTien;  private String trangThai, nguoiTao;  public HoaDonDichVu() {  }  public HoaDonDichVu(int maHDDV, String suDung, String maPhong, float tongTien, String trangThai, String nguoiTao) {  this.maHDDV = maHDDV;  this.suDung = suDung;  this.maPhong = maPhong;  this.tongTien = tongTien;  this.trangThai = trangThai;  this.nguoiTao = nguoiTao;  }  public int getMaHDDV() {  return maHDDV;  }  public void setMaHDDV(int maHDDV) {  this.maHDDV = maHDDV;  }  public String getSuDung() {  return suDung;  }  public void setSuDung(String suDung) {  this.suDung = suDung;  }  public String getMaPhong() {  return maPhong;  }  public void setMaPhong(String maPhong) {  this.maPhong = maPhong;  }  public float getTongTien() {  return tongTien;  }  public void setTongTien(float tongTien) {  this.tongTien = tongTien;  }  public String getTrangThai() {  return trangThai;  }  public void setTrangThai(String trangThai) {  this.trangThai = trangThai;  }  public String getNguoiTao() {  return nguoiTao;  }  public void setNguoiTao(String nguoiTao) {  this.nguoiTao = nguoiTao;  }  } |

#### HopDong Model

|  |
| --- |
| package com.sample.model;  public class HopDong {  private String maHD, maNguoiDaiDien;  private int soNguoi;  private String maNguoiO;  private int maHDDV;  private String maPhong, ngayThue, ngayTra;  private int thanhTien;  private String trangthai, nguoiTao, maDV, ngayThanhToan;  public HopDong() {  }  public HopDong(String maHD, String maNguoiDaiDien, int soNguoi, String maNguoiO, int maHDDV, String maPhong, String ngayThue, String ngayTra, int thanhTien, String trangthai, String nguoiTao, String maDV, String ngayThanhToan) {  this.maHD = maHD;  this.maNguoiDaiDien = maNguoiDaiDien;  this.soNguoi = soNguoi;  this.maNguoiO = maNguoiO;  this.maHDDV = maHDDV;  this.maPhong = maPhong;  this.ngayThue = ngayThue;  this.ngayTra = ngayTra;  this.thanhTien = thanhTien;  this.trangthai = trangthai;  this.nguoiTao = nguoiTao;  this.maDV = maDV;  this.ngayThanhToan = ngayThanhToan;  }  public String getMaHD() {  return maHD;  }  public void setMaHD(String maHD) {  this.maHD = maHD;  }  public String getMaNguoiDaiDien() {  return maNguoiDaiDien;  }  public void setMaNguoiDaiDien(String maNguoiDaiDien) {  this.maNguoiDaiDien = maNguoiDaiDien;  }  public int getSoNguoi() {  return soNguoi;  }  public void setSoNguoi(int soNguoi) {  this.soNguoi = soNguoi;  }  public String getMaNguoiO() {  return maNguoiO;  }  public void setMaNguoiO(String maNguoiO) {  this.maNguoiO = maNguoiO;  }  public int getMaHDDV() {  return maHDDV;  }  public void setMaHDDV(int maHDDV) {  this.maHDDV = maHDDV;  }  public String getMaPhong() {  return maPhong;  }  public void setMaPhong(String maPhong) {  this.maPhong = maPhong;  }  public String getNgayThue() {  return ngayThue;  }  public void setNgayThue(String ngayThue) {  this.ngayThue = ngayThue;  }  public String getNgayTra() {  return ngayTra;  }  public void setNgayTra(String ngayTra) {  this.ngayTra = ngayTra;  }  public int getThanhTien() {  return thanhTien;  }  public void setThanhTien(int thanhTien) {  this.thanhTien = thanhTien;  }  public String getTrangthai() {  return trangthai;  }  public void setTrangthai(String trangthai) {  this.trangthai = trangthai;  }  public String getNguoiTao() {  return nguoiTao;  }  public void setNguoiTao(String nguoiTao) {  this.nguoiTao = nguoiTao;  }  public String getMaDV() {  return maDV;  }  public void setMaDV(String maDV) {  this.maDV = maDV;  }  public String getNgayThanhToan() {  return ngayThanhToan;  }  public void setNgayThanhToan(String ngayThanhToan) {  this.ngayThanhToan = ngayThanhToan;  }  } |

#### KhachTro Model

|  |
| --- |
| package com.sample.model;  public class KhachTro {  private String maKT, tenKT, gioiTinh, soDT, soCMT, eMail, maPhongThue, trangthai;  public KhachTro() {  }  public KhachTro(String maKT, String tenKT, String gioiTinh, String soDT, String soCMT, String eMail, String maPhongThue, String trangthai) {  this.maKT = maKT;  this.tenKT = tenKT;  this.gioiTinh = gioiTinh;  this.soDT = soDT;  this.soCMT = soCMT;  this.eMail = eMail;  this.maPhongThue = maPhongThue;  this.trangthai = trangthai;  }  public String getMaKT() {  return maKT;  }  public void setMaKT(String maKT) {  this.maKT = maKT;  }  public String getTenKT() {  return tenKT;  }  public void setTenKT(String tenKT) {  this.tenKT = tenKT;  }  public String getGioiTinh() {  return gioiTinh;  }  public void setGioiTinh(String gioiTinh) {  this.gioiTinh = gioiTinh;  }  public String getSoDT() {  return soDT;  }  public void setSoDT(String soDT) {  this.soDT = soDT;  }  public String getSoCMT() {  return soCMT;  }  public void setSoCMT(String soCMT) {  this.soCMT = soCMT;  }  public String geteMail() {  return eMail;  }  public void seteMail(String eMail) {  this.eMail = eMail;  }  public String getMaPhongThue() {  return maPhongThue;  }  public void setMaPhongThue(String maPhongThue) {  this.maPhongThue = maPhongThue;  }  public String getTrangthai() {  return trangthai;  }  public void setTrangthai(String trangthai) {  this.trangthai = trangthai;  }  } |

#### NhanVien Model

|  |
| --- |
| package com.sample.model;  public class NhanVien {  private String maNV;  private int userID;  private String tenNV, gioiTinh, soDT, soCMT, eMail, maKhu, trangThai;  public NhanVien() {  }  public NhanVien(String maNV, int userID, String tenNV, String gioiTinh, String soDT, String soCMT, String eMail, String maKhu, String trangThai) {  this.maNV = maNV;  this.userID = userID;  this.tenNV = tenNV;  this.gioiTinh = gioiTinh;  this.soDT = soDT;  this.soCMT = soCMT;  this.eMail = eMail;  this.maKhu = maKhu;  this.trangThai = trangThai;  }  public String getMaNV() {  return maNV;  }  public void setMaNV(String maNV) {  this.maNV = maNV;  }  public int getUserID() {  return userID;  }  public void setUserID(int userID) {  this.userID = userID;  }  public String getTenNV() {  return tenNV;  }  public void setTenNV(String tenNV) {  this.tenNV = tenNV;  }  public String getGioiTinh() {  return gioiTinh;  }  public void setGioiTinh(String gioiTinh) {  this.gioiTinh = gioiTinh;  }  public String getSoDT() {  return soDT;  }  public void setSoDT(String soDT) {  this.soDT = soDT;  }  public String getSoCMT() {  return soCMT;  }  public void setSoCMT(String soCMT) {  this.soCMT = soCMT;  }  public String geteMail() {  return eMail;  }  public void seteMail(String eMail) {  this.eMail = eMail;  }  public String getMaKhu() {  return maKhu;  }  public void setMaKhu(String maKhu) {  this.maKhu = maKhu;  }  public String getTrangThai() {  return trangThai;  }  public void setTrangThai(String trangThai) {  this.trangThai = trangThai;  }  } |

#### Phong Model

|  |
| --- |
| package com.sample.model;  public class Phong {  private String maPhong, tenPhong;  private float giaTien,dienTich;  private String trangThai, maKhu;  private String ngayThanhToan;  public Phong() {  }  public Phong(String maPhong, String tenPhong, float giaTien, float dienTich, String trangThai, String maKhu, String ngayThanhToan) {  this.maPhong = maPhong;  this.tenPhong = tenPhong;  this.giaTien = giaTien;  this.dienTich = dienTich;  this.trangThai = trangThai;  this.maKhu = maKhu;  this.ngayThanhToan = ngayThanhToan;  }  public String getMaPhong() {  return maPhong;  }  public void setMaPhong(String maPhong) {  this.maPhong = maPhong;  }  public String getTenPhong() {  return tenPhong;  }  public void setTenPhong(String tenPhong) {  this.tenPhong = tenPhong;  }  public float getGiaTien() {  return giaTien;  }  public void setGiaTien(float giaTien) {  this.giaTien = giaTien;  }  public float getDienTich() {  return dienTich;  }  public void setDienTich(float dienTich) {  this.dienTich = dienTich;  }  public String getTrangThai() {  return trangThai;  }  public void setTrangThai(String trangThai) {  this.trangThai = trangThai;  }  public String getMaKhu() {  return maKhu;  }  public void setMaKhu(String maKhu) {  this.maKhu = maKhu;  }  public String getNgayThanhToan() {  return ngayThanhToan;  }  public void setNgayThanhToan(String ngayThanhToan) {  this.ngayThanhToan = ngayThanhToan;  }  } |

#### User Model

|  |
| --- |
| package com.sample.model;  public class User {  private String userID;  private String user, passWord, roles;  public User() {  }  public User(String userID, String user, String passWord, String roles) {  this.userID = userID;  this.user = user;  this.passWord = passWord;  this.roles = roles;  }  public String getUserID() {  return userID;  }  public void setUserID(String userID) {  this.userID = userID;  }  public String getUser() {  return user;  }  public void setUser(String user) {  this.user = user;  }  public String getPassWord() {  return passWord;  }  public void setPassWord(String passWord) {  this.passWord = passWord;  }  public String getRoles() {  return roles;  }  public void setRoles(String roles) {  this.roles = roles;  }  } |

### DAO Class - Các lớp truy xuất dữ liệu

#### Lớp DAO chính

|  |
| --- |
| package com.sample.DAO;  import java.util.List;  public interface DAO<DAO> {    List<DAO> select();    void insert(DAO model);    void update(DAO model);    void delete(DAO model);  } |

#### DichVuDAO

|  |
| --- |
| package com.sample.DAO;  import com.sample.helper.JdbcHelper;  import com.sample.model.DichVu;  import java.sql.ResultSet;  import java.util.ArrayList;  import java.util.List;  public class DichVuDAO implements DAO<DichVu>{  @Override  public List<DichVu> select() {  String sql = "exec sp\_selectAllDV";  return select(sql);  }  private List<DichVu> select(String sql, Object... args) {  List<DichVu> list = new ArrayList<>();  ResultSet rs = null;  try {  rs = JdbcHelper.executeQuery(sql, args);  while (rs.next()) {  DichVu model = new DichVu();  model.setMaDV(rs.getString(1));  model.setTenDV(rs.getString(2));  model.setGia(rs.getInt(3));  list.add(model);  }  } catch (Exception ex) {  throw new RuntimeException(ex);  } finally {  try {  rs.close();  } catch (Exception e) {  throw new RuntimeException(e);  };  }  return list;  }  @Override  public void insert(DichVu model) {  String sql = "exec sp\_insertDV ?, ?, ? ";  JdbcHelper.executeUpdate(sql, model.getMaDV(), model.getTenDV(), model.getGia());  }  @Override  public void update(DichVu model) {  String sql = "exec sp\_updateDV ?, ? ";  JdbcHelper.executeUpdate(sql, model.getMaDV(), model.getGia());  }  @Override  public void delete(DichVu model) {  String sql = "exec sp\_deleteDV ? ";  JdbcHelper.executeUpdate(sql, model.getMaDV());  }  } |

#### HoaDonChiTietDAO

|  |
| --- |
| package com.sample.DAO;  import com.sample.helper.JdbcHelper;  import com.sample.model.HoaDonChiTiet;  import java.sql.ResultSet;  import java.util.ArrayList;  import java.util.List;  /\*\*  \*  \* @author akais  \*/  public class HoaDonChiTietDAO implements DAO<HoaDonChiTiet>{  @Override  public List<HoaDonChiTiet> select() {  String sql = "exec sp\_selectAllHDCT ";  return select(sql);  }  private List<HoaDonChiTiet> select(String sql, Object... args) {  List<HoaDonChiTiet> list = new ArrayList<>();  ResultSet rs = null;  try {  rs = JdbcHelper.executeQuery(sql, args);  while (rs.next()) {  HoaDonChiTiet model = new HoaDonChiTiet();  model.setMaHDCT(rs.getInt(1));  model.setMaDVSD(rs.getString(2));  model.setMaPhong(rs.getString(3));  model.setThanhTien(rs.getInt(4));  model.setThoiGian(rs.getString(5));  model.setMaHDDV(rs.getInt(6));  list.add(model);  }  } catch (Exception ex) {  throw new RuntimeException(ex);  } finally {  try {  rs.close();  } catch (Exception e) {  throw new RuntimeException(e);  };  }  return list;  }    @Override  public void insert(HoaDonChiTiet model) {  String sql = "exec sp\_insertHDCT ?, ?, ?, ?, ?";  JdbcHelper.executeUpdate(sql,  model.getMaDVSD(),  model.getMaPhong(),  model.getThanhTien(),  model.getThoiGian(),  model.getMaHDDV());  }  @Override  public void update(HoaDonChiTiet model) {  throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  }  @Override  public void delete(HoaDonChiTiet model) {  throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  }    public List<HoaDonChiTiet> findMaHDCTByMaPhong(String maPhong){  String sql = "exec sp\_selectMaHDCTByMaPhong ? ";  return select(sql, maPhong);  }  } |

#### HoaDonDichVuDAO

|  |
| --- |
| package com.sample.DAO;  import com.sample.helper.JdbcHelper;  import com.sample.model.HoaDonDichVu;  import java.sql.ResultSet;  import java.util.ArrayList;  import java.util.List;  /\*\*  \*  \* @author akais  \*/  public class HoaDonDichVuDAO implements DAO<HoaDonDichVu> {  @Override  public List<HoaDonDichVu> select() {  String sql = "exec sp\_selectAllHDDV";  return select(sql);  }  private List<HoaDonDichVu> select(String sql, Object... args) {  List<HoaDonDichVu> list = new ArrayList<>();  ResultSet rs = null;  try {  rs = JdbcHelper.executeQuery(sql, args);  while (rs.next()) {  HoaDonDichVu model = new HoaDonDichVu();  model.setMaHDDV(rs.getInt(1));  model.setSuDung(rs.getString(2));  model.setNguoiTao(rs.getString(3));  model.setTongTien(rs.getInt(4));  model.setTrangThai(rs.getString(5));  model.setNguoiTao(rs.getString(6));  list.add(model);  }  } catch (Exception ex) {  throw new RuntimeException(ex);  } finally {  try {  rs.close();  } catch (Exception e) {  throw new RuntimeException(e);  };  }  return list;  }  @Override  public void insert(HoaDonDichVu model) {  String sql = "exec sp\_insertHDDV ?, ?, ?, ?, ?";  JdbcHelper.executeUpdate(sql,  model.getSuDung(),  model.getMaPhong(),  model.getTongTien(),  model.getTrangThai(),  model.getNguoiTao()  );  }  @Override  public void update(HoaDonDichVu model) {  String sql = "exec sp\_updateHDDV ?, ?, ?, ?";  JdbcHelper.executeUpdate(sql,  model.getSuDung(),  model.getMaPhong(),  model.getTongTien(),  model.getTrangThai());  }  @Override  public void delete(HoaDonDichVu model) {  throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  }    public List<HoaDonDichVu> findMaHDDVByMaPhong(String maPhong){  String sql = "exec sp\_selectMaHDDVByMaPhong ? ";  return select(sql, maPhong);  }  } |

#### HopDongDAO

|  |
| --- |
| package com.sample.DAO;  import com.sample.helper.JdbcHelper;  import com.sample.model.HopDong;  import java.sql.ResultSet;  import java.util.ArrayList;  import java.util.List;  public class HopDongDAO implements DAO<HopDong> {  @Override  public List<HopDong> select() {  String sql = "exec sp\_selectAllHD";  return select(sql);  }  private List<HopDong> select(String sql, Object... args) {  List<HopDong> list = new ArrayList<>();  ResultSet rs = null;  try {  rs = JdbcHelper.executeQuery(sql, args);  while (rs.next()) {  HopDong model = new HopDong();  model.setMaHD(rs.getString(1));  model.setMaNguoiDaiDien(rs.getString(2));  model.setSoNguoi(rs.getInt(3));  model.setMaNguoiO(rs.getString(4));  model.setMaHDDV(rs.getInt(5));  model.setMaPhong(rs.getString(6));  model.setNgayThue(rs.getString(7));  model.setNgayTra(rs.getString(8));  model.setThanhTien(rs.getInt(9));  model.setTrangthai(rs.getString(10));  model.setNguoiTao(rs.getString(11));  model.setMaDV(rs.getString(12));  model.setNgayThanhToan(rs.getString(13));  list.add(model);  }  } catch (Exception ex) {  throw new RuntimeException(ex);  } finally {  try {  rs.close();  } catch (Exception e) {  throw new RuntimeException(e);  };  }  return list;  }  @Override  public void insert(HopDong model) {  String sql = "exec sp\_insertHD ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ? ";  JdbcHelper.executeUpdate(sql,  model.getMaNguoiDaiDien(),  model.getSoNguoi(),  model.getMaNguoiO(),  model.getMaHDDV(),  model.getMaPhong(),  model.getNgayThue(),  model.getNgayTra(),  model.getThanhTien(),  model.getTrangthai(),  model.getNguoiTao(),  model.getMaDV(),  model.getNgayThanhToan());  }  @Override  public void update(HopDong model) {  String sql = "exec sp\_updateHD ?, ?, ?, ?, ?,? ";  JdbcHelper.executeUpdate(sql,  model.getMaHD(),  model.getSoNguoi(),  model.getMaNguoiO(),  model.getNgayTra(),  model.getThanhTien(),  model.getTrangthai());  }  @Override  public void delete(HopDong model) {  String sql = "exec sp\_deleteHD ? ";  JdbcHelper.executeUpdate(sql, model.getMaHD());  }    public List<HopDong> findByMaPhong(String maPhong){  String sql = "exec sp\_findHDByMaPhong ? ";  return select(sql, maPhong);  }  } |

#### KhachTroDAO

|  |
| --- |
| package com.sample.DAO;  import com.sample.helper.JdbcHelper;  import com.sample.model.KhachTro;  import java.sql.ResultSet;  import java.util.ArrayList;  import java.util.List;  public class KhachTroDAO implements DAO<KhachTro> {  /\*\*  \*  \* @param model  \*/  @Override  public void insert(KhachTro model) {  String sql = "exec sp\_insertKT ?, ?, ?, ?, ?, ?, ?, ? ";  JdbcHelper.executeUpdate(sql,  model.getMaKT(),  model.getTenKT(),  model.getGioiTinh(),  model.getSoDT(),  model.getSoCMT(),  model.geteMail(),  model.getMaPhongThue(),  model.getTrangthai());  }  @Override  public List<KhachTro> select() {  String sql = "exec sp\_selectAllKT";  return select(sql);  }  private List<KhachTro> select(String sql, Object... args) {  List<KhachTro> list = new ArrayList<>();  ResultSet rs = null;  try {  rs = JdbcHelper.executeQuery(sql, args);  while (rs.next()) {  KhachTro model = new KhachTro();  model.setMaKT(rs.getString(1));  model.setTenKT(rs.getString(2));  model.setGioiTinh(rs.getString(3));  model.setSoDT(rs.getString(4));  model.setSoCMT(rs.getString(5));  model.seteMail(rs.getString(6));  model.setMaPhongThue(rs.getString(7));  list.add(model);  }  } catch (Exception ex) {  throw new RuntimeException(ex);  } finally {  try {  rs.close();  } catch (Exception e) {  throw new RuntimeException(e);  };  }  return list;  }  @Override  public void delete(KhachTro model) {  String sql = "exec sp\_deleteKT ?";  JdbcHelper.executeUpdate(sql, model.getMaKT());  }  @Override  public void update(KhachTro model) {  String sql = "exec sp\_updateKT ?, ?, ?, ?";  JdbcHelper.executeUpdate(sql, model.getMaKT(), model.getSoDT(), model.getSoCMT(), model.geteMail());  }  public List<KhachTro> findByMaKT(String maKT) {  String sql = "exec sp\_findKTByMAKT ? ";  return select(sql, maKT);  }  } |

#### NhanVienDAO

|  |
| --- |
| package com.sample.DAO;  import com.sample.helper.JdbcHelper;  import com.sample.model.NhanVien;  import java.sql.ResultSet;  import java.util.ArrayList;  import java.util.List;  public class NhanVienDAO implements DAO<NhanVien> {  public void insert(NhanVien model) {  String sql = "exec sp\_insertNV ?, ?, ?, ?, ?, ?, ?, ?, ?";  JdbcHelper.executeUpdate(sql,  model.getMaNV(),  model.getUserID(),  model.getTenNV(),  model.getGioiTinh(),  model.getSoDT(),  model.getSoCMT(),  model.geteMail(),  model.getMaKhu(),  model.getTrangThai());  }  @Override  public List<NhanVien> select() {  String sql = "exec sp\_selectAllNV";  return select(sql);  }  private List<NhanVien> select(String sql, Object... args) {  List<NhanVien> list = new ArrayList<>();  ResultSet rs = null;  try {  rs = JdbcHelper.executeQuery(sql, args);  while (rs.next()) {  NhanVien model = new NhanVien();  model.setMaNV(rs.getString(1));  model.setUserID(rs.getInt(2));  model.setTenNV(rs.getString(3));  model.setGioiTinh(rs.getString(4));  model.setSoDT(rs.getString(5));  model.setSoCMT(rs.getString(6));  model.seteMail(rs.getString(7));  model.setMaKhu(rs.getString(8));  list.add(model);  }  } catch (Exception ex) {  throw new RuntimeException(ex);  } finally {  try {  rs.close();  } catch (Exception e) {  throw new RuntimeException(e);  };  }  return list;  }  @Override  public void update(NhanVien model) {  String sql = "exec sp\_updateNV ?, ?, ?, ?, ? ";  JdbcHelper.executeUpdate(sql, model.getMaNV(), model.getSoDT(), model.getSoCMT(), model.geteMail(), model.getMaKhu());  }  @Override  public void delete(NhanVien model) {  String sql = "exec sp\_deleteNV ?";  JdbcHelper.executeUpdate(sql, model.getMaNV());  }    public List<NhanVien> findByMaNV(String maNV){  String sql = "exec sp\_findNVByMANV ? ";  return select(sql, maNV);  }  } |

#### PhongDAO

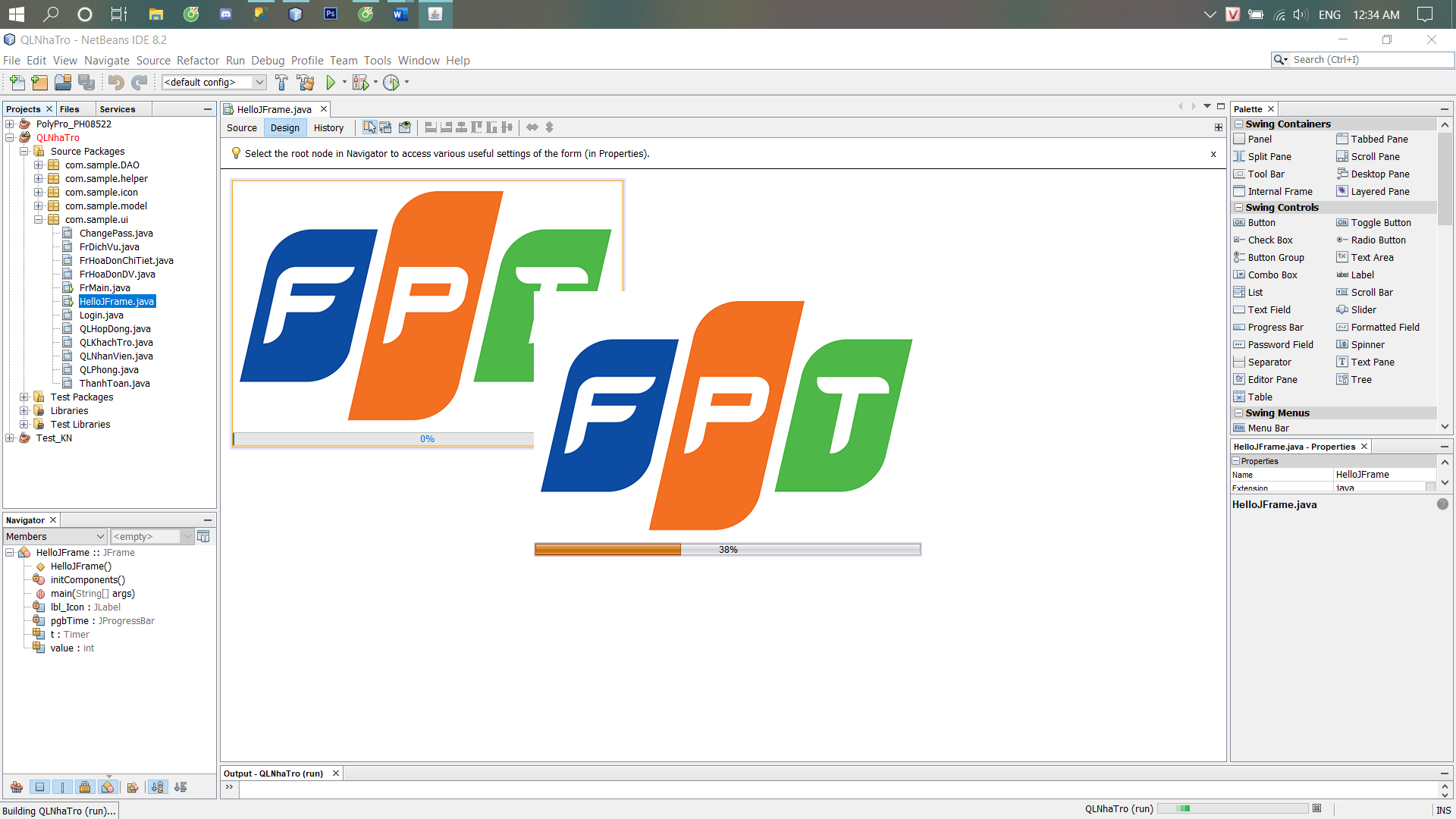
|  |
| --- |
| package com.sample.DAO;  import com.sample.helper.JdbcHelper;  import com.sample.model.Phong;  import java.sql.ResultSet;  import java.util.ArrayList;  import java.util.List;  public class PhongDAO implements DAO<Phong>{  public void insert(Phong model) {  String sql = "exec sp\_insertRoom ?, ?, ?, ?, ?, ?,? ";  JdbcHelper.executeUpdate(sql,  model.getMaPhong(),  model.getTenPhong(),  model.getGiaTien(),  model.getDienTich(),  model.getTrangThai(),  model.getMaKhu(),  model.getNgayThanhToan());  }  public List<Phong> select() {  String sql = "exec sp\_selectAllRoom";  return select(sql);  }  private List<Phong> select(String sql, Object... args) {  List<Phong> list = new ArrayList<>();  ResultSet rs = null;  try {  rs = JdbcHelper.executeQuery(sql, args);  while (rs.next()) {  Phong model = new Phong();  model.setMaPhong(rs.getString(1));  model.setTenPhong(rs.getString(2));  model.setGiaTien(rs.getInt(3));  model.setDienTich(rs.getInt(4));  model.setTrangThai(rs.getString(5));  model.setMaKhu(rs.getString(6));  model.setNgayThanhToan(rs.getString(7));  list.add(model);  }  } catch (Exception ex) {  throw new RuntimeException(ex);  } finally {  try {  rs.close();  } catch (Exception e) {  throw new RuntimeException(e);  };  }  return list;  }  public void update(Phong model) {  String sql = "exec sp\_updateRoom ?, ?, ?";  JdbcHelper.executeUpdate(sql, model.getMaPhong(), model.getGiaTien(), model.getTrangThai());  }  public List<Phong> findBySatus(String STT) {  String sql = "exec sp\_findNVBySTT ? ";  return select(sql, STT);  }  @Override  public void delete(Phong model) {  String sql =" exec sp\_deleteRoom ? ";  JdbcHelper.executeUpdate(sql, model.getMaPhong());  }  } |

#### UserDAO

|  |
| --- |
| package com.sample.DAO;  import com.sample.helper.JdbcHelper;  import com.sample.model.User;  import java.sql.ResultSet;  import java.util.ArrayList;  import java.util.List;  public class UserDAO implements DAO<User> {  private List<User> list = new ArrayList<>();  @Override  public List<User> select() {  String sql = "select \* from Users";  return select(sql);  }  private List<User> select(String sql, Object... args) {  ResultSet rs = null;  try {  rs = JdbcHelper.executeQuery(sql, args);  while (rs.next()) {  User model = new User();  model.setUserID(rs.getString(1));  model.setUser(rs.getString(2));  model.setPassWord(rs.getString(3));  model.setRoles(rs.getString(4));  list.add(model);  }  } catch (Exception ex) {  throw new RuntimeException(ex);  } finally {  try {  rs.close();  } catch (Exception e) {  throw new RuntimeException(e);  };  }  return list;  }  @Override  public void insert(User model) {  String sql = "insert into Users values(?, ?, ?)";  JdbcHelper.executeUpdate(sql,  model.getUser(),  model.getPassWord(),  model.getRoles());  }  @Override  public void update(User model) {  String sql = "exec sp\_changePass ?, ? ";  JdbcHelper.executeUpdate(sql, model.getUser(),model.getPassWord());  }  @Override  public void delete(User model) {  String sql = "exec sp\_deleteUser ? ";  JdbcHelper.executeUpdate(sql, model.getUserID());  }  public List<User> findByUser(String userID) {  String sql = "exec sp\_selectUser ? ";  return select(sql, userID);  }  public void convertPassToString(String userName) {  String sql = "exec sp\_convertPasstoString ? ";  JdbcHelper.executeUpdate(sql, userName);  }    public void convertStringtoPass(String userNamePass) {  String sql = "exec sp\_convertStringToPass ? ";  JdbcHelper.executeUpdate(sql, userNamePass);  }  public List<User> selectLogin(String userName, String passWord) {  String sql = "exec sp\_Login ?, ? ";  return select(sql,userName, passWord);  }  } |

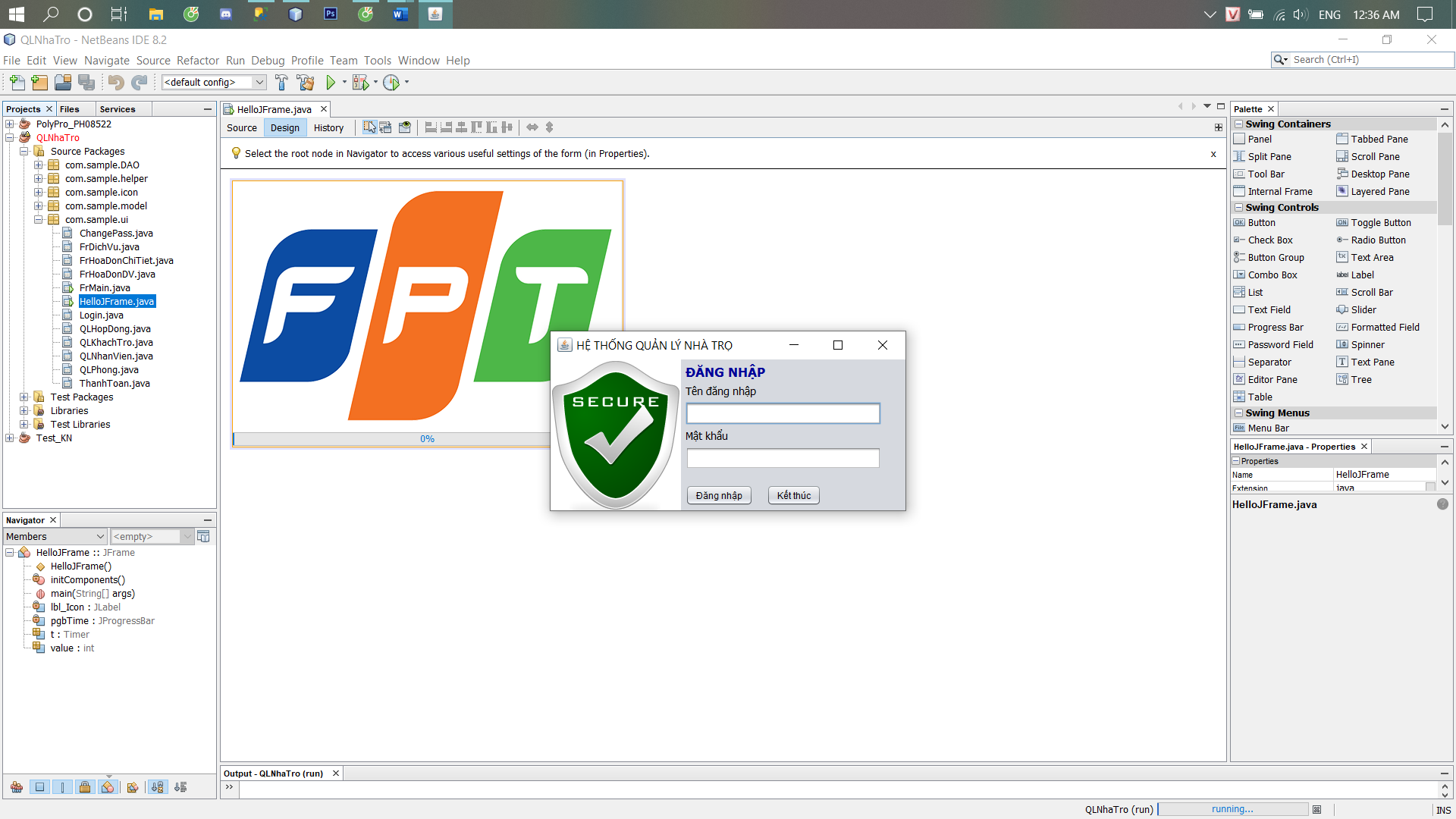
## Viết mã cho ứng dụng

### Màn hình chào



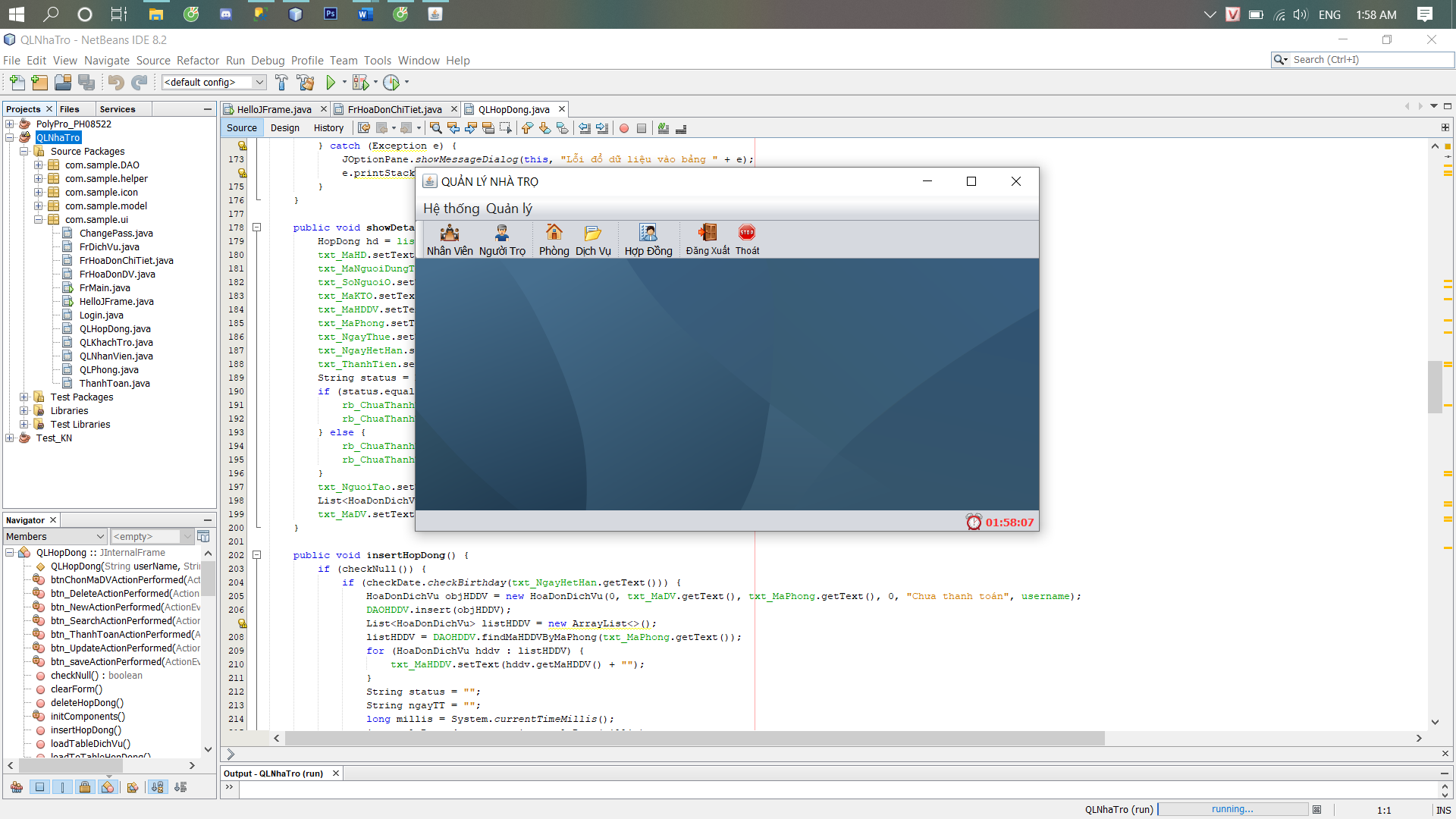
|  |
| --- |
| package com.sample.ui;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import javax.swing.Timer;  public class HelloJFrame extends javax.swing.JFrame {  Timer t;  int value;  public HelloJFrame() {  initComponents();  setLocationRelativeTo(null);  t = new Timer(50, new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  value = pgbTime.getValue() + 1;  pgbTime.setValue(value > 100 ? 0 : value);  if(pgbTime.getValue() == 100){  t.stop();  new Login().setVisible(true);  dispose();  }  }  });  t.start();  } |

### Màn hình đăng nhập



|  |
| --- |
| package com.sample.ui;  import com.sample.DAO.UserDAO;  import com.sample.model.User;  import java.awt.Frame;  import java.util.ArrayList;  import java.util.List;  import javax.swing.JOptionPane;  public class Login extends javax.swing.JFrame {  List<User> listUser = new ArrayList<>();  String roles = "";  UserDAO userDAO = new UserDAO();  String userName = "";  String pass = "";  public boolean checkNull() {  userName = txt\_TenDN.getText();  pass = new String(ps\_MatKhau.getPassword());  if (userName.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống tên đăng nhập");  return false;  } else if (pass.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống mật khẩu");  return false;  }  return true;  }  public void login() {  if (checkNull()) {  userDAO.convertPassToString(userName);  List<User> list = userDAO.selectLogin(userName, pass);  if (userName.equalsIgnoreCase(list.get(0).getUser()) && pass.equalsIgnoreCase(list.get(0).getPassWord())) {  roles = list.get(0).getRoles();  userDAO.convertStringtoPass(userName);  JOptionPane.showMessageDialog(this, "Đăng nhập thành công!!");  if (roles.equalsIgnoreCase("NV")) {  new FrMain(userName, roles).setVisible(true);  } else if (roles.equalsIgnoreCase("CN")) {  new FrMain(userName, roles).setVisible(true);  }  }  }  }  public Login() {  initComponents();  setLocationRelativeTo(null);  }  private void btn\_DangNhapActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.login();  this.dispose();  } catch (Exception e) {  userDAO.convertStringtoPass(userName);  JOptionPane.showMessageDialog(this, "Sai tên đăng nhập hoặc password!!");  }  } |

### Màn hình chính

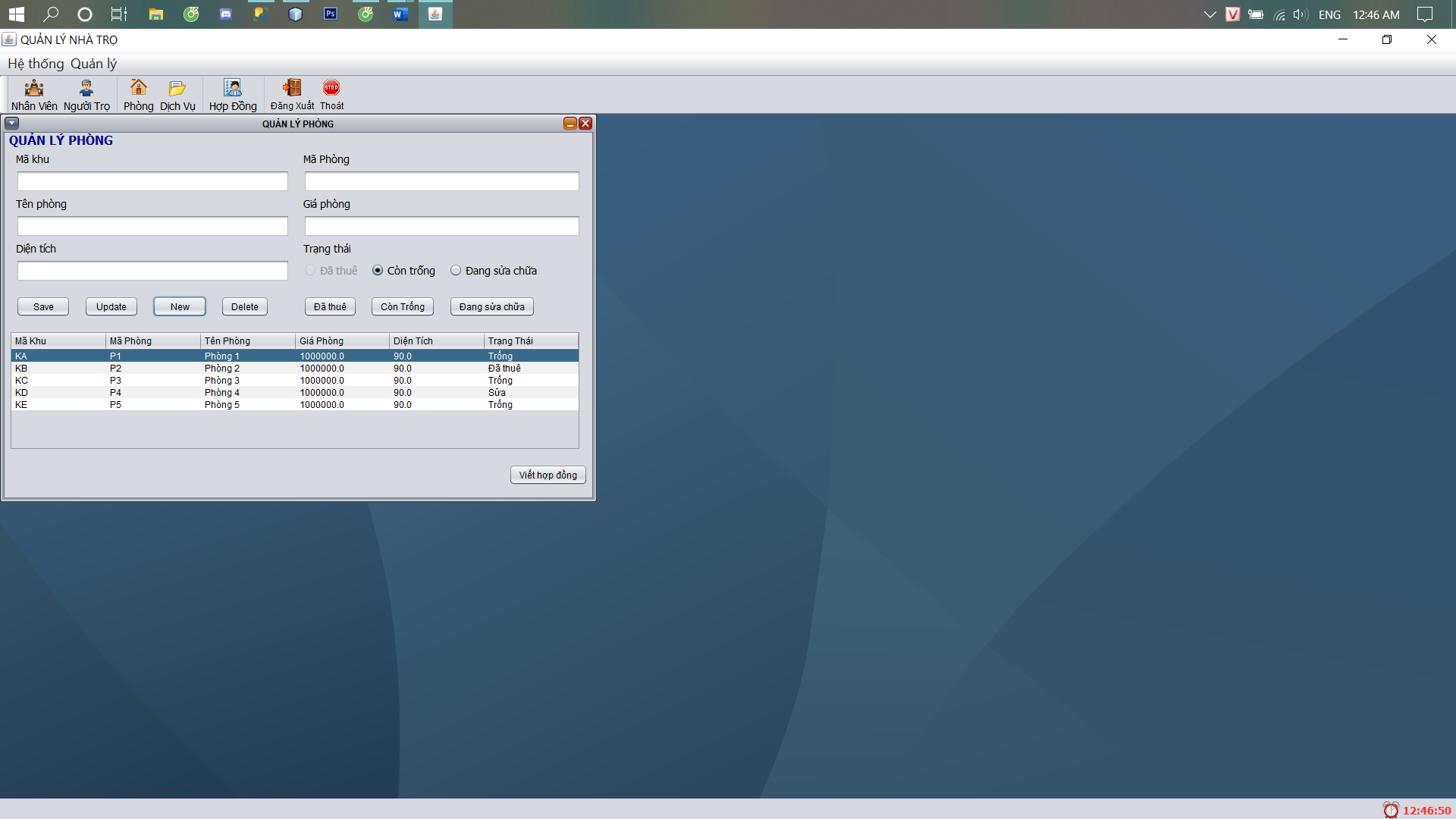


|  |
| --- |
| package com.sample.ui;  import com.sample.model.NhanVien;  import java.awt.HeadlessException;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.text.SimpleDateFormat;  import java.util.Date;  import javax.swing.JButton;  import javax.swing.JDesktopPane;  import javax.swing.JLabel;  import javax.swing.JMenu;  import javax.swing.JMenuBar;  import javax.swing.JMenuItem;  import javax.swing.JOptionPane;  import javax.swing.JPanel;  import javax.swing.JPopupMenu;  import javax.swing.JToolBar;  import javax.swing.Timer;  public class FrMain extends javax.swing.JFrame {  Timer t;  String roles;  String userName;  public static FrMain main;  public FrMain() {  main = this;  initComponents();  }  public FrMain(String roles, String userName, JButton btn\_Exit, JButton btn\_LogOut, JButton btn\_QLDichVu, JButton btn\_QLHopDong, JButton btn\_QLNguoiTro, JButton btn\_QLNhanVien, JButton btn\_QLPhong, JDesktopPane dsk\_main, JLabel jLabel1, JMenuBar jMenuBar1, JMenuItem jMenuItem1, JPanel jPanel1, JPanel jPanel3, JPopupMenu.Separator jSeparator1, JToolBar.Separator jSeparator2, JToolBar.Separator jSeparator4, JPopupMenu.Separator jSeparator5, JPopupMenu.Separator jSeparator6, JToolBar.Separator jSeparator7, JToolBar jToolBar2, JMenuItem mni\_ChangePassword, JMenuItem mni\_Exit, JMenuItem mni\_LogOut, JMenuItem mni\_QLDichVu, JMenuItem mni\_QLHoaDon, JMenuItem mni\_QLHopDong, JMenuItem mni\_QLKhachTro, JMenuItem mni\_QLNhanVien, JMenuItem mni\_QLPhong, JMenu mnu\_HeThong, JMenu mnu\_QuanLy) throws HeadlessException {  this.roles = roles;  this.userName = userName;  this.btn\_Exit = btn\_Exit;  this.btn\_LogOut = btn\_LogOut;  this.btn\_QLDichVu = btn\_QLDichVu;  this.btn\_QLHopDong = btn\_QLHopDong;  this.btn\_QLNguoiTro = btn\_QLNguoiTro;  this.btn\_QLNhanVien = btn\_QLNhanVien;  this.btn\_QLPhong = btn\_QLPhong;  this.dsk\_main = dsk\_main;  this.lbl\_Time = jLabel1;  this.jMenuBar1 = jMenuBar1;  this.jMenuItem1 = jMenuItem1;  this.pnl\_Time = jPanel1;  this.pnl\_Back = jPanel3;  this.jSeparator1 = jSeparator1;  this.jSeparator2 = jSeparator2;  this.jSeparator4 = jSeparator4;  this.jSeparator5 = jSeparator5;  this.jSeparator6 = jSeparator6;  this.jSeparator7 = jSeparator7;  this.jToolBar2 = jToolBar2;  this.mni\_ChangePassword = mni\_ChangePassword;  this.mni\_Exit = mni\_Exit;  this.mni\_LogOut = mni\_LogOut;  this.mni\_QLDichVu = mni\_QLDichVu;  this.mni\_HDCT = mni\_QLHoaDon;  this.mni\_QLHopDong = mni\_QLHopDong;  this.mni\_QLKhachTro = mni\_QLKhachTro;  this.mni\_QLNhanVien = mni\_QLNhanVien;  this.mni\_QLPhong = mni\_QLPhong;  this.mnu\_HeThong = mnu\_HeThong;  this.mnu\_QuanLy = mnu\_QuanLy;  }  public String getRoles() {  return roles;  }  public void setRoles(String roles) {  this.roles = roles;  }  public String getUserName() {  return userName;  }  public void setUserName(String userName) {  this.userName = userName;  }  public static FrMain getMain() {  return main;  }  public static void setMain(FrMain main) {  FrMain.main = main;  }  public JButton getBtn\_Exit() {  return btn\_Exit;  }  public void setBtn\_Exit(JButton btn\_Exit) {  this.btn\_Exit = btn\_Exit;  }  public JButton getBtn\_LogOut() {  return btn\_LogOut;  }  public void setBtn\_LogOut(JButton btn\_LogOut) {  this.btn\_LogOut = btn\_LogOut;  }  public JButton getBtn\_QLDichVu() {  return btn\_QLDichVu;  }  public void setBtn\_QLDichVu(JButton btn\_QLDichVu) {  this.btn\_QLDichVu = btn\_QLDichVu;  }  public JButton getBtn\_QLHopDong() {  return btn\_QLHopDong;  }  public void setBtn\_QLHopDong(JButton btn\_QLHopDong) {  this.btn\_QLHopDong = btn\_QLHopDong;  }  public JButton getBtn\_QLNguoiTro() {  return btn\_QLNguoiTro;  }  public void setBtn\_QLNguoiTro(JButton btn\_QLNguoiTro) {  this.btn\_QLNguoiTro = btn\_QLNguoiTro;  }  public JButton getBtn\_QLNhanVien() {  return btn\_QLNhanVien;  }  public void setBtn\_QLNhanVien(JButton btn\_QLNhanVien) {  this.btn\_QLNhanVien = btn\_QLNhanVien;  }  public JButton getBtn\_QLPhong() {  return btn\_QLPhong;  }  public void setBtn\_QLPhong(JButton btn\_QLPhong) {  this.btn\_QLPhong = btn\_QLPhong;  }  public JDesktopPane getDsk\_main() {  return dsk\_main;  }  public void setDsk\_main(JDesktopPane dsk\_main) {  this.dsk\_main = dsk\_main;  }  public JLabel getjLabel1() {  return lbl\_Time;  }  public void setjLabel1(JLabel jLabel1) {  this.lbl\_Time = jLabel1;  }  public JMenuBar getjMenuBar1() {  return jMenuBar1;  }  public void setjMenuBar1(JMenuBar jMenuBar1) {  this.jMenuBar1 = jMenuBar1;  }  public JMenuItem getjMenuItem1() {  return jMenuItem1;  }  public void setjMenuItem1(JMenuItem jMenuItem1) {  this.jMenuItem1 = jMenuItem1;  }  public JPanel getjPanel1() {  return pnl\_Time;  }  public void setjPanel1(JPanel jPanel1) {  this.pnl\_Time = jPanel1;  }  public JPanel getjPanel3() {  return pnl\_Back;  }  public void setjPanel3(JPanel jPanel3) {  this.pnl\_Back = jPanel3;  }  public JPopupMenu.Separator getjSeparator1() {  return jSeparator1;  }  public void setjSeparator1(JPopupMenu.Separator jSeparator1) {  this.jSeparator1 = jSeparator1;  }  public JToolBar.Separator getjSeparator2() {  return jSeparator2;  }  public void setjSeparator2(JToolBar.Separator jSeparator2) {  this.jSeparator2 = jSeparator2;  }  public JToolBar.Separator getjSeparator4() {  return jSeparator4;  }  public void setjSeparator4(JToolBar.Separator jSeparator4) {  this.jSeparator4 = jSeparator4;  }  public JPopupMenu.Separator getjSeparator5() {  return jSeparator5;  }  public void setjSeparator5(JPopupMenu.Separator jSeparator5) {  this.jSeparator5 = jSeparator5;  }  public JPopupMenu.Separator getjSeparator6() {  return jSeparator6;  }  public void setjSeparator6(JPopupMenu.Separator jSeparator6) {  this.jSeparator6 = jSeparator6;  }  public JToolBar.Separator getjSeparator7() {  return jSeparator7;  }  public void setjSeparator7(JToolBar.Separator jSeparator7) {  this.jSeparator7 = jSeparator7;  }  public JToolBar getjToolBar2() {  return jToolBar2;  }  public void setjToolBar2(JToolBar jToolBar2) {  this.jToolBar2 = jToolBar2;  }  public JMenuItem getMni\_ChangePassword() {  return mni\_ChangePassword;  }  public void setMni\_ChangePassword(JMenuItem mni\_ChangePassword) {  this.mni\_ChangePassword = mni\_ChangePassword;  }  public JMenuItem getMni\_Exit() {  return mni\_Exit;  }  public void setMni\_Exit(JMenuItem mni\_Exit) {  this.mni\_Exit = mni\_Exit;  }  public JMenuItem getMni\_LogOut() {  return mni\_LogOut;  }  public void setMni\_LogOut(JMenuItem mni\_LogOut) {  this.mni\_LogOut = mni\_LogOut;  }  public JMenuItem getMni\_QLDichVu() {  return mni\_QLDichVu;  }  public void setMni\_QLDichVu(JMenuItem mni\_QLDichVu) {  this.mni\_QLDichVu = mni\_QLDichVu;  }  public JMenuItem getMni\_QLHoaDon() {  return mni\_HDCT;  }  public void setMni\_QLHoaDon(JMenuItem mni\_QLHoaDon) {  this.mni\_HDCT = mni\_QLHoaDon;  }  public JMenuItem getMni\_QLHopDong() {  return mni\_QLHopDong;  }  public void setMni\_QLHopDong(JMenuItem mni\_QLHopDong) {  this.mni\_QLHopDong = mni\_QLHopDong;  }  public JMenuItem getMni\_QLKhachTro() {  return mni\_QLKhachTro;  }  public void setMni\_QLKhachTro(JMenuItem mni\_QLKhachTro) {  this.mni\_QLKhachTro = mni\_QLKhachTro;  }  public JMenuItem getMni\_QLNhanVien() {  return mni\_QLNhanVien;  }  public void setMni\_QLNhanVien(JMenuItem mni\_QLNhanVien) {  this.mni\_QLNhanVien = mni\_QLNhanVien;  }  public JMenuItem getMni\_QLPhong() {  return mni\_QLPhong;  }  public void setMni\_QLPhong(JMenuItem mni\_QLPhong) {  this.mni\_QLPhong = mni\_QLPhong;  }  public JMenu getMnu\_HeThong() {  return mnu\_HeThong;  }  public void setMnu\_HeThong(JMenu mnu\_HeThong) {  this.mnu\_HeThong = mnu\_HeThong;  }  public JMenu getMnu\_QuanLy() {  return mnu\_QuanLy;  }  public void setMnu\_QuanLy(JMenu mnu\_QuanLy) {  this.mnu\_QuanLy = mnu\_QuanLy;  }  public FrMain(String userName, String roles) {  main = this;  initComponents();  setLocationRelativeTo(null);  this.roles = roles;  this.userName = userName;  if (this.roles.equalsIgnoreCase("NV")) {  btn\_QLNhanVien.setEnabled(false);  mni\_QLNhanVien.setEnabled(false);  }  setExtendedState(MAXIMIZED\_BOTH);  SimpleDateFormat sdf = new SimpleDateFormat("hh:mm:ss");  t = new Timer(100, new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  Date d = new Date();  lbl\_Time.setText(sdf.format(d));  }  });  t.start();  }  private void mni\_QLDichVuActionPerformed(java.awt.event.ActionEvent evt) {  try {  FrDichVu dv = new FrDichVu();  dsk\_main.add(dv);  dv.setVisible(true);  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_QLNhanVienActionPerformed(java.awt.event.ActionEvent evt) {  QLNhanVien qlnv = new QLNhanVien();  dsk\_main.add(qlnv);  qlnv.setVisible(true);  }  private void btn\_QLNguoiTroActionPerformed(java.awt.event.ActionEvent evt) {  QLKhachTro qlkt = new QLKhachTro(roles);  dsk\_main.add(qlkt);  qlkt.setVisible(true);  }  private void btn\_QLPhongActionPerformed(java.awt.event.ActionEvent evt) {  QLPhong qlp = new QLPhong(userName, roles);  dsk\_main.add(qlp);  qlp.setVisible(true);  }  private void mni\_ChangePasswordActionPerformed(java.awt.event.ActionEvent evt) {  try {  ChangePass cp = new ChangePass(userName);  dsk\_main.add(cp);  cp.setVisible(true);  } catch (Exception e) {  }  }  private void btn\_QLDichVuActionPerformed(java.awt.event.ActionEvent evt) {  try {  FrDichVu dv = new FrDichVu();  dsk\_main.add(dv);  dv.setVisible(true);  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_ExitActionPerformed(java.awt.event.ActionEvent evt) {  try {  int ex = JOptionPane.showConfirmDialog(this, "Bạn có muốn thoát?");  if (ex == JOptionPane.YES\_OPTION) {  System.exit(0);  }  } catch (Exception e) {  }  }  private void btn\_QLHopDongActionPerformed(java.awt.event.ActionEvent evt) {  try {  QLHopDong hd = new QLHopDong(userName, roles, "");  dsk\_main.add(hd);  hd.setVisible(true);  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  e.printStackTrace();  }  }  private void btn\_LogOutActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.dispose();  new Login().setVisible(true);  } catch (Exception e) {  }  }  private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {  try {  ThanhToan hd = new ThanhToan(userName, "", "", "");  dsk\_main.add(hd);  hd.setVisible(true);  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  e.printStackTrace();  }  }  private void mni\_LogOutActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.dispose();  new Login().setVisible(true);  } catch (Exception e) {  }  }  private void mni\_ExitActionPerformed(java.awt.event.ActionEvent evt) {  System.exit(0);  }  private void mni\_QLNhanVienActionPerformed(java.awt.event.ActionEvent evt) {  QLNhanVien qlnv = new QLNhanVien();  dsk\_main.add(qlnv);  qlnv.setVisible(true);  }  private void mni\_QLKhachTroActionPerformed(java.awt.event.ActionEvent evt) {  QLKhachTro qlkt = new QLKhachTro(roles);  dsk\_main.add(qlkt);  qlkt.setVisible(true);  }  private void mni\_QLPhongActionPerformed(java.awt.event.ActionEvent evt) {  QLPhong qlp = new QLPhong(userName, roles);  dsk\_main.add(qlp);  qlp.setVisible(true);  }  private void mni\_QLHopDongActionPerformed(java.awt.event.ActionEvent evt) {  try {  QLHopDong hd = new QLHopDong(userName, roles, "");  dsk\_main.add(hd);  hd.setVisible(true);  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  e.printStackTrace();  }  }  private void mni\_HDCTActionPerformed(java.awt.event.ActionEvent evt) {  try {  FrHoaDonChiTiet hdct = new FrHoaDonChiTiet();  dsk\_main.add(hdct);  hdct.setVisible(true);  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  e.printStackTrace();  }  }  private void mni\_HDDVActionPerformed(java.awt.event.ActionEvent evt) {  try {  FrHoaDonDV hddv = new FrHoaDonDV();  dsk\_main.add(hddv);  hddv.setVisible(true);  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  e.printStackTrace();  }  }  public static void main(String args[]) {  java.awt.EventQueue.invokeLater(new Runnable() {  public void run() {  new FrMain("", "").setVisible(true);  }  });  } |

### Màn quản lý nhân viên

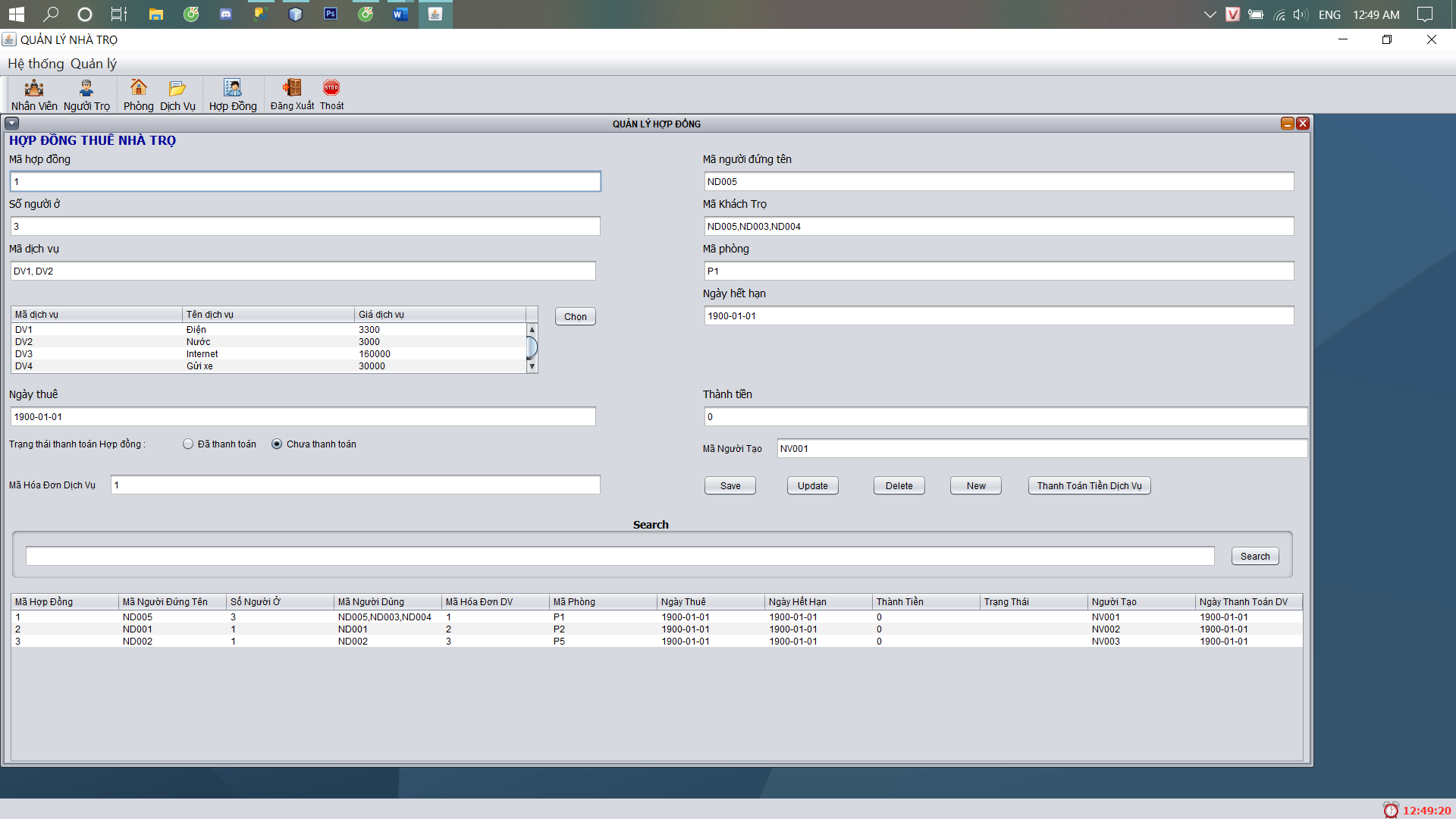
|  |
| --- |
| package com.sample.ui;  import com.sample.DAO.NhanVienDAO;  import com.sample.DAO.UserDAO;  import com.sample.helper.JdbcHelper;  import com.sample.model.NhanVien;  import com.sample.model.User;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.util.ArrayList;  import java.util.List;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel;  public class QLNhanVien extends javax.swing.JInternalFrame {  DefaultTableModel model;  int index = 0;  List<NhanVien> listNhanVien = new ArrayList<>();  NhanVienDAO NhanVienDAO = new NhanVienDAO();  List<User> listUser = new ArrayList<>();  User userModel = new User();  UserDAO userDAO = new UserDAO();  String maNV = "";  String tenNV = "";  String soDT = "";  String soCMT = "";  String eMail = "";  String maKhu = "";  public boolean checkRepeat(String maNVcheck, String userIDcheck) {  boolean trung = false;  for (int i = 0; i < listNhanVien.size(); i++) {  if (maNVcheck.equalsIgnoreCase(listNhanVien.get(i).getMaNV())) {  JOptionPane.showMessageDialog(this, "Trùng mã Nhân Viên");  trung = false;  return trung;  } else if (userIDcheck.equals(listNhanVien.get(i).getUserID() + "")) {  JOptionPane.showMessageDialog(this, "Trùng userID");  trung = false;  return trung;  } else {  trung = true;  }  }  return trung;  }  public boolean checkNull() {  maNV = txt\_MaNV.getText();  tenNV = txt\_TenNV.getText();  soDT = txt\_SoDT.getText();  soCMT = txt\_SoCMND.getText();  eMail = txt\_Email.getText();  maKhu = txt\_MaKhu.getText();  String inputSoDT = "\\d{9,10}";  String inputSoCMT = "\\d{12}";  String eMailForm = "\\w+\\w@+\\w.+\\w";  if (maNV.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống mã Nhân viên!!");  return false;  } else if (tenNV.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống Họ tên!!");  return false;  } else if (soDT.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống SĐT");  return false;  } else if (!(soDT).matches(inputSoDT)) {  JOptionPane.showMessageDialog(this, "Số điện thoại phải là số từ 9 đến 10 số");  return false;  } else if (soCMT.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống số CMT");  return false;  } else if (!(soCMT).matches(inputSoCMT)) {  JOptionPane.showMessageDialog(this, "Số CMT phải là 12 số");  return false;  } else if (eMail.equals("")) {  JOptionPane.showMessageDialog(this, "không bỏ trống email");  return false;  } else if (!eMail.matches(eMailForm)) {  JOptionPane.showMessageDialog(this, "sai định dang, định dạng email: abc@abc.abc");  return false;  }  return true;  }  public void loadTableNhanVien() {  model = (DefaultTableModel) tbl\_NhanVien.getModel();  model.setRowCount(0);  try {  listNhanVien = NhanVienDAO.select();  for (NhanVien nh : listNhanVien) {  model.addRow(new Object[]{nh.getUserID(), nh.getMaNV(), nh.getTenNV(), nh.getGioiTinh(), nh.getSoDT(), nh.getSoCMT(), nh.geteMail(), nh.getMaKhu()});  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ dữ liệu vào bảng" + e);  e.printStackTrace();  }  }  public void showDetail() {  NhanVien nv = listNhanVien.get(index);  txt\_UserID.setText(nv.getUserID() + "");  txt\_MaNV.setText(nv.getMaNV());  txt\_TenNV.setText(nv.getTenNV());  String gioiTinh = nv.getGioiTinh();  if (gioiTinh.equalsIgnoreCase("Nam")) {  rb\_Nam.setSelected(true);  rb\_Nu.setSelected(false);  } else {  rb\_Nam.setSelected(false);  rb\_Nu.setSelected(true);  }  txt\_SoDT.setText(nv.getSoDT());  txt\_SoCMND.setText(nv.getSoCMT());  txt\_Email.setText(nv.geteMail());  txt\_MaKhu.setText(nv.getMaKhu());  tbl\_NhanVien.setRowSelectionInterval(index, index);  }  public void clearForm() {  txt\_MaNV.setEditable(true);  txt\_TenNV.setEditable(true);  txt\_MaNV.setText("");  txt\_UserID.setText("");  txt\_TenNV.setText("");  txt\_SoDT.setText("");  txt\_SoCMND.setText("");  txt\_Email.setText("");  rb\_Nam.setSelected(true);  rb\_Nu.setSelected(false);  txt\_MaKhu.setText("");  }  public void insertNhanVien() {  if (checkRepeat(txt\_MaNV.getText(), txt\_UserID.getText()) && checkNull()) {  String gioiTinh = "";  if (rb\_Nam.isSelected()) {  gioiTinh = "Nam";  } else {  gioiTinh = "Nữ";  }  userModel = new User(null, maNV, "123", "NV");  userDAO.insert(userModel);  listUser = userDAO.findByUser(maNV);  for (User us : listUser) {  txt\_UserID.setText(us.getUserID());  }  NhanVien nv = new NhanVien(maNV, Integer.parseInt(txt\_UserID.getText()), tenNV, gioiTinh, soDT, soCMT, eMail, maKhu, "");  NhanVienDAO.insert(nv);  loadTableNhanVien();  JOptionPane.showMessageDialog(this, "Thêm thành công!!");  }  }  public void updateNhanVien() {  try {  if (checkNull()) {  NhanVien objNhanVien = new NhanVien(maNV, Integer.parseInt(txt\_UserID.getText()), null, null, soDT, soCMT, eMail, maKhu, null);  NhanVienDAO.update(objNhanVien);  loadTableNhanVien();  JOptionPane.showMessageDialog(this, "Update thành công");  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi update" + e);  e.printStackTrace();  }  }  public void deleteNhanVien() {  int ask = JOptionPane.showConfirmDialog(this, "Bạn có muốn xóa");  if (ask == JOptionPane.YES\_OPTION) {  NhanVien objNhanVien = new NhanVien(txt\_MaNV.getText(), Integer.parseInt(txt\_UserID.getText()), null, null, soDT, soCMT, eMail, maKhu, null);  NhanVienDAO.delete(objNhanVien);  JOptionPane.showMessageDialog(this, "Xóa thành công!!");  loadTableNhanVien();  if (listNhanVien.size() > 0) {  if (index == listNhanVien.size()) {  index--;  }  showDetail();  } else {  clearForm();  }  }  }  private void troVeFirst() {  try {  index = 0;  showDetail();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi: " + e);  }  }  private void troVeTruoc() {  try {  if (index > 0) {  index--;  showDetail();  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi: " + e);  }  }  private void troVeSau() {  try {  if (index < listNhanVien.size() - 1) {  index++;  showDetail();  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "đang ở cuối rồi ");  }  }  private void troVeLast() {  try {  index = listNhanVien.size() - 1;  showDetail();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi: " + e);  }  }  public QLNhanVien() {  initComponents();  loadTableNhanVien();  if (listNhanVien.size() >= 0) {  index = 0;  showDetail();  }  }  private void btn\_SaveActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.insertNhanVien();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi: " + e);  }  }  private void tbl\_NhanVienMouseClicked(java.awt.event.MouseEvent evt) {  try {  index = tbl\_NhanVien.getSelectedRow();  if (index >= 0) {  showDetail();  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ thông tin lên form");  }  }  private void btn\_FistActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.troVeFirst();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_PreviousActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.troVeTruoc();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_NextActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.troVeSau();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_LastActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.troVeLast();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_NewActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.clearForm();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_DeleteActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.deleteNhanVien();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_UpdateActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.updateNhanVien();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  } |

### Màn hình quản lý phòng



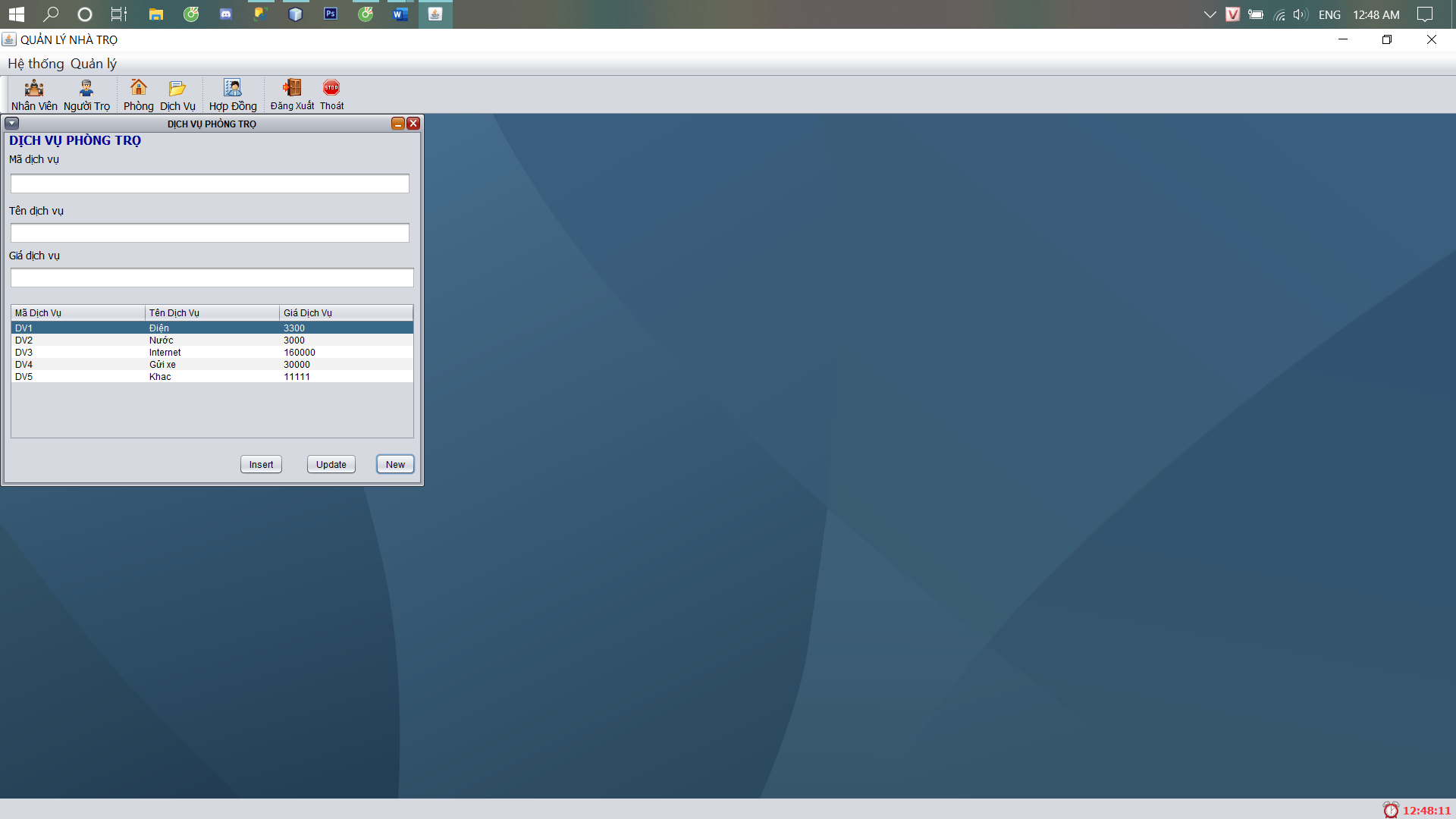
|  |
| --- |
| package com.sample.ui;  import com.sample.DAO.PhongDAO;  import com.sample.model.Phong;  import java.util.ArrayList;  import java.util.List;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel;  public class QLPhong extends javax.swing.JInternalFrame {  String roles;  String userName;  DefaultTableModel model;  int index = 0;  List<Phong> listPhong = new ArrayList<>();  PhongDAO phongDAO = new PhongDAO();  String maPhong = "";  String tenPhong = "";  String giaTien = "";  String dienTich = "";  String trangThai = "";  String maKhu = "";  public boolean checkRepeat(String maPhongCheck) {  boolean trung = false;  for (int i = 0; i < listPhong.size(); i++) {  if (maPhongCheck.equalsIgnoreCase(listPhong.get(i).getMaPhong())) {  JOptionPane.showMessageDialog(this, "Trùng mã phòng");  trung = false;  return trung;  } else {  trung = true;  }  }  return trung;  }  public boolean checkNull() {  maPhong = txt\_MaPhong.getText();  tenPhong = txt\_TenPhong.getText();  giaTien = txt\_GiaPhong.getText();  dienTich = txt\_DienTich.getText();  maKhu = txt\_MaKhu.getText();  String inputNumber = "\\d{6,10}";  String inputS = "\\d{1,2}";  if (maPhong.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống mã phòng!!");  return false;  } else if (tenPhong.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống tên phòng!!");  return false;  } else if (giaTien.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống giá tiền");  return false;  } else if (!(giaTien).matches(inputNumber)) {  JOptionPane.showMessageDialog(this, "Giá không hơp lệ!! Giá phải có từ 100000 trở lên");  return false;  } else if (dienTich.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống diện tích phòng");  return false;  } else if (!(dienTich).matches(inputS)) {  JOptionPane.showMessageDialog(this, "Diện tích không hợp lệ!! Diện tích không quá 100 mét vuông");  return false;  } else if (maKhu.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống mã khu");  return false;  }  return true;  }  public void loadTablePhong() {  model = (DefaultTableModel) tbl\_QLPhong.getModel();  model.setRowCount(0);  try {  listPhong = phongDAO.select();  for (Phong nh : listPhong) {  model.addRow(new Object[]{nh.getMaKhu(), nh.getMaPhong(), nh.getTenPhong(), nh.getGiaTien(), nh.getDienTich(), nh.getTrangThai()});  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ dữ liệu vào bảng" + e);  e.printStackTrace();  }  }  public void loadTablePhongByStatus(String tinhTrang) {  model = (DefaultTableModel) tbl\_QLPhong.getModel();  model.setRowCount(0);  try {  listPhong = phongDAO.findBySatus(tinhTrang);  for (Phong nh : listPhong) {  model.addRow(new Object[]{nh.getMaKhu(), nh.getMaPhong(), nh.getTenPhong(), nh.getGiaTien(), nh.getDienTich(), nh.getTrangThai()});  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ dữ liệu vào bảng" + e);  e.printStackTrace();  }  }  public void insertPhong() {  try {  if (checkRepeat(txt\_MaPhong.getText()) && checkNull()) {  String status = "";  if (rb\_ConTrong.isSelected()) {  status = "Trống";  } else if (rb\_DangSuaChua.isSelected()) {  status = "Sửa";  }  Phong objPhong = new Phong(maPhong, tenPhong, Integer.parseInt(txt\_GiaPhong.getText()), Integer.parseInt(txt\_DienTich.getText()), status, maKhu, "");  phongDAO.insert(objPhong);  loadTablePhong();  JOptionPane.showMessageDialog(this, "Thêm thành công!!");  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi Thêm" + e);  }  }  public void updatePhong() {  try {  String status = "";  if (rb\_ConTrong.isSelected()) {  status = "Trống";  } else if (rb\_DangSuaChua.isSelected()) {  status = "Sửa";  }  Phong objPhong = new Phong(txt\_MaPhong.getText(), tenPhong, Integer.parseInt(txt\_GiaPhong.getText()), Integer.parseInt(txt\_DienTich.getText()), status, maKhu, "");  phongDAO.update(objPhong);  loadTablePhong();  JOptionPane.showMessageDialog(this, "Update thành công");  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi update" + e);  }  }  public void deletePhong() {  try {  int ask = JOptionPane.showConfirmDialog(this, "Bạn có muốn xóa");  if (ask == JOptionPane.YES\_OPTION) {  Phong objPhong = new Phong(txt\_MaPhong.getText(), null, 0, 0, null, null, null);  phongDAO.delete(objPhong);  JOptionPane.showMessageDialog(this, "Xóa thành công!!");  loadTablePhong();  if (listPhong.size() > 0) {  if (index == listPhong.size()) {  index--;  }  showDetail();  } else {  clearForm();  }  }  } catch (Exception e) {  }  }  public void clearForm() {  rb\_DaThue.setEnabled(false);  txt\_MaPhong.setEditable(true);  txt\_TenPhong.setEditable(true);  txt\_MaKhu.setEditable(true);  txt\_MaPhong.setText("");  txt\_TenPhong.setText("");  txt\_GiaPhong.setText("");  txt\_DienTich.setText("");  txt\_MaKhu.setText("");  }  public void showDetail() {  if (model.getValueAt(index, 5).equals("Đã thuê")) {  btn\_Update.setEnabled(false);  btn\_Delete.setEnabled(false);  } else {  btn\_Update.setEnabled(true);  btn\_Delete.setEnabled(true);  }  if (model.getValueAt(index, 5).equals("Trống")) {  btn\_VietHD.setEnabled(true);  } else {  btn\_VietHD.setEnabled(false);  }  rb\_DaThue.setEnabled(false);  Phong nv = listPhong.get(index);  txt\_MaPhong.setText(nv.getMaPhong());  txt\_TenPhong.setText(nv.getTenPhong());  txt\_GiaPhong.setText(nv.getGiaTien() + "");  txt\_DienTich.setText(nv.getDienTich() + "");  txt\_MaKhu.setText(nv.getMaKhu());  String status = nv.getTrangThai();  if (status.equalsIgnoreCase("Đã thuê")) {  rb\_DaThue.setSelected(true);  rb\_ConTrong.setSelected(false);  rb\_DangSuaChua.setSelected(false);  } else if (status.equalsIgnoreCase("Trống")) {  rb\_DaThue.setSelected(false);  rb\_ConTrong.setSelected(true);  rb\_DangSuaChua.setSelected(false);  } else if (status.equalsIgnoreCase("Sửa")) {  rb\_DaThue.setSelected(false);  rb\_ConTrong.setSelected(false);  rb\_DangSuaChua.setSelected(true);  }  tbl\_QLPhong.setRowSelectionInterval(index, index);  }  public QLPhong(String userName, String roles) {  this.roles = roles;  this.userName = userName;  initComponents();  loadTablePhong();  if (roles.equalsIgnoreCase("NV")) {  btn\_Save.setEnabled(false);  btn\_new.setEnabled(false);  btn\_Update.setEnabled(false);  btn\_Delete.setEnabled(false);  }  if (listPhong.size() >= 0) {  index = 0;  showDetail();  }  }  private void tbl\_QLPhongMouseClicked(java.awt.event.MouseEvent evt) {  try {  index = tbl\_QLPhong.getSelectedRow();  if (index >= 0) {  showDetail();  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ thông tin lên form");  }  }  private void btn\_UpdateActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.updatePhong();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  }  private void btn\_newActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.clearForm();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  }  private void btn\_SaveActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.insertPhong();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  }  private void btn\_DaThueActionPerformed(java.awt.event.ActionEvent evt) {  try {  btn\_Update.setEnabled(false);  btn\_Save.setEnabled(false);  btn\_Delete.setEnabled(false);  loadTablePhongByStatus("Đã thuê");  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  }  private void btnTrongActionPerformed(java.awt.event.ActionEvent evt) {  try {  loadTablePhongByStatus("Trống");  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  }  private void btn\_FixActionPerformed(java.awt.event.ActionEvent evt) {  try {  loadTablePhongByStatus("Sửa");  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  }  private void btn\_DeleteActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.deletePhong();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  }  private void btn\_VietHDActionPerformed(java.awt.event.ActionEvent evt) {  try {  QLHopDong qlhd = new QLHopDong(userName, roles,txt\_MaPhong.getText());    FrMain.main.getDsk\_main().add(qlhd);  qlhd.show();    } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  } |

### Màn hình quản lý hợp đồng



|  |
| --- |
| package com.sample.ui;  import com.sample.DAO.DichVuDAO;  import com.sample.DAO.HoaDonDichVuDAO;  import com.sample.DAO.HopDongDAO;  import com.sample.helper.checkDate;  import com.sample.model.DichVu;  import com.sample.model.HoaDonDichVu;  import com.sample.model.HopDong;  import java.util.ArrayList;  import java.util.List;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel;  public class QLHopDong extends javax.swing.JInternalFrame {  String roles;  String username;  DefaultTableModel model;  DefaultTableModel modelDV;  int index = 0;  List<HopDong> listHopDong = new ArrayList<>();  HopDongDAO hopDongDAO = new HopDongDAO();  DichVuDAO dicVuDAO = new DichVuDAO();  HoaDonDichVuDAO DAOHDDV = new HoaDonDichVuDAO();  String maNguoiDD = "";  String soNguoiO = "";  String maNguoiO = "";  String maPhong;  String ngayThue = "";  String ngayTra = "";  String thanhTien = "";  String maSuDung = "DV1, DV2";  public boolean checkNull() {  maNguoiDD = txt\_MaNguoiDungTen.getText();  soNguoiO = txt\_SoNguoiO.getText();  maNguoiO = txt\_MaKT.getText();  maPhong = txt\_MaPhong.getText();  ngayThue = txt\_NgayThue.getText();  ngayTra = txt\_NgayHetHan.getText();  thanhTien = txt\_ThanhTien.getText();  String gioiHan = "\\d{1}";  String giaTien = "\\d{6,8}";  if (maNguoiDD.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống mã người đại diện");  return false;  } else if (!soNguoiO.matches(gioiHan)) {  JOptionPane.showMessageDialog(this, "Tối đa 9 người ở!!");  return false;  } else if (maNguoiO.equals("")) {  if (!soNguoiO.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống mã người ở");  return false;  }  } else if (maPhong.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống mã phòng");  return false;  } else if (ngayThue.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống ngày thuê");  return false;  } else if (ngayTra.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống ngày trả");  return false;  } else if (thanhTien.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống thành tiền");  return false;  } else if (!thanhTien.matches(giaTien)) {  JOptionPane.showMessageDialog(this, "Giá không hợp lệ!! Tiền từ: 100000 -> 99999999");  return false;  }  return true;  }  public void clearForm() {  txt\_MaHD.setText("");  txt\_MaHDDV.setText("");  txt\_MaKT.setText("");  txt\_MaNguoiDungTen.setText("");  txt\_MaPhong.setText("");  txt\_NgayHetHan.setText("");  txt\_NgayThue.setText("");  txt\_SoNguoiO.setText("");  txt\_ThanhTien.setText("");  txt\_MaNguoiTao.setText(username);  }  public void loadTableDichVu() {  modelDV = (DefaultTableModel) tbl\_BangDV.getModel();  modelDV.setRowCount(0);  try {  List<DichVu> listDV = dicVuDAO.select();  for (DichVu dv : listDV) {  modelDV.addRow(new Object[]{dv.getMaDV(), dv.getTenDV(), dv.getGia()});  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ dữ liệu vào bảng" + e);  e.printStackTrace();  }  }  public void loadToTableHopDong() {  model = (DefaultTableModel) tbl\_HopDong.getModel();  model.setRowCount(0);  try {  listHopDong = hopDongDAO.select();  for (HopDong hd : listHopDong) {  model.addRow(new Object[]{  hd.getMaHD(),  hd.getMaNguoiDaiDien(),  hd.getSoNguoi(),  hd.getMaNguoiO(),  hd.getMaHDDV(),  hd.getMaPhong(),  hd.getNgayThue(),  hd.getNgayTra(),  hd.getThanhTien(),  hd.getTrangthai(),  hd.getNguoiTao(),  hd.getNgayThanhToan()  });  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ dữ liệu vào bảng " + e);  e.printStackTrace();  }  }  public void loadToTableHopDongByMAPhong() {  model = (DefaultTableModel) tbl\_HopDong.getModel();  model.setRowCount(0);  try {  List<HopDong> listFHD = hopDongDAO.findByMaPhong(txt\_Search.getText());  for (HopDong hd : listFHD) {  model.addRow(new Object[]{  hd.getMaHD(),  hd.getMaNguoiDaiDien(),  hd.getSoNguoi(),  hd.getMaNguoiO(),  hd.getMaHDDV(),  hd.getMaPhong(),  hd.getNgayThue(),  hd.getNgayTra(),  hd.getThanhTien(),  hd.getTrangthai(),  hd.getNguoiTao(),  hd.getNgayThanhToan()  });  }  if (listFHD.size() > 0) {  JOptionPane.showMessageDialog(this, "Tìm thành công");  index = 0;  tbl\_HopDong.setRowSelectionInterval(index, index);  showDetail();  } else {  loadToTableHopDong();  JOptionPane.showMessageDialog(this, "Không có thông tin mã phòng này");  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ dữ liệu vào bảng " + e);  e.printStackTrace();  }  }  public void showDetail() {  HopDong hd = listHopDong.get(index);  txt\_MaHD.setText(hd.getMaHD());  txt\_MaNguoiDungTen.setText(hd.getMaNguoiDaiDien());  txt\_SoNguoiO.setText(hd.getSoNguoi() + "");  txt\_MaKT.setText(hd.getMaNguoiO());  txt\_MaHDDV.setText(hd.getMaHDDV() + "");  txt\_MaPhong.setText(hd.getMaPhong());  txt\_NgayThue.setText(hd.getNgayThue());  txt\_NgayHetHan.setText(hd.getNgayTra());  txt\_ThanhTien.setText(hd.getThanhTien() + "");  String status = hd.getTrangthai();  if (status.equalsIgnoreCase("Chưa thanh toán")) {  rb\_ChuaThanhToan.setSelected(true);  rb\_ChuaThanhToan.setSelected(false);  } else {  rb\_ChuaThanhToan.setSelected(false);  rb\_ChuaThanhToan.setSelected(true);  }  txt\_MaNguoiTao.setText(hd.getNguoiTao());  List<HoaDonDichVu> listHDDV = DAOHDDV.findMaHDDVByMaPhong(txt\_MaPhong.getText());  txt\_MaDV.setText(listHDDV.get(0).getSuDung());  }  public void insertHopDong() {  if (checkNull()) {  if (checkDate.checkBirthday(txt\_NgayHetHan.getText())) {  HoaDonDichVu objHDDV = new HoaDonDichVu(0, txt\_MaDV.getText(), txt\_MaPhong.getText(), 0, "Chưa thanh toán", username);  DAOHDDV.insert(objHDDV);  List<HoaDonDichVu> listHDDV = new ArrayList<>();  listHDDV = DAOHDDV.findMaHDDVByMaPhong(txt\_MaPhong.getText());  for (HoaDonDichVu hddv : listHDDV) {  txt\_MaHDDV.setText(hddv.getMaHDDV() + "");  }  String status = "";  String ngayTT = "";  long millis = System.currentTimeMillis();  java.sql.Date date = new java.sql.Date(millis);  if (rb\_ChuaThanhToan.isSelected()) {  status = "Chưa thanh toán";  } else if (rb\_DaThanhToan.isSelected()) {  status = "Đã thanh toán";  ngayTT = date + "";  }  HopDong objHD = new HopDong("0",  txt\_MaNguoiDungTen.getText(),  Integer.parseInt(txt\_SoNguoiO.getText()),  txt\_MaKT.getText(),  Integer.parseInt(txt\_MaHDDV.getText()),  txt\_MaPhong.getText(),  txt\_NgayThue.getText(),  txt\_NgayHetHan.getText(),  Integer.parseInt(thanhTien),  status,  username,  "DV1",  ngayTT);  hopDongDAO.insert(objHD);  loadToTableHopDong();  JOptionPane.showMessageDialog(this, "Thêm thành công!!");  } else {  JOptionPane.showMessageDialog(this, "Sai định dạng ngày: YYYY-MM-DD ");  }  }  }  public void updateHopDong() {  try {  long millis = System.currentTimeMillis();  java.sql.Date date = new java.sql.Date(millis);  DAOHDDV.update(new HoaDonDichVu(0, txt\_MaDV.getText(), maPhong, 0, "Chưa thanh toán", username));  String status = "";  String ngayTT = "";  if (rb\_ChuaThanhToan.isSelected()) {  status = "Chưa thanh toán";  } else if (rb\_DaThanhToan.isSelected()) {  status = "Đã Thanh Toán";  ngayTT = date + "";  }  HopDong objHopDong = new HopDong(txt\_MaHD.getText(), null, Integer.parseInt(txt\_SoNguoiO.getText()), txt\_MaKT.getText(), 0, txt\_MaPhong.getText(), null, txt\_NgayHetHan.getText(), Integer.parseInt(txt\_ThanhTien.getText()), status, null, null, null);  hopDongDAO.update(objHopDong);  loadToTableHopDong();  JOptionPane.showMessageDialog(this, "Update thành công");  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi update" + e);  e.printStackTrace();  }  }  public void deleteHopDong() {  int ask = JOptionPane.showConfirmDialog(this, "Bạn có muốn xóa");  if (ask == JOptionPane.YES\_OPTION) {  HopDong objHopDong = new HopDong(txt\_MaHD.getText(), null, Integer.parseInt(txt\_SoNguoiO.getText()), txt\_MaKT.getText(), 0, txt\_MaPhong.getText(), null, txt\_NgayHetHan.getText(), Integer.parseInt(txt\_ThanhTien.getText()), null, null, null, null);  hopDongDAO.delete(objHopDong);  JOptionPane.showMessageDialog(this, "Xóa thành công!!");  loadToTableHopDong();  if (listHopDong.size() > 0) {  if (index == listHopDong.size()) {  index--;  }  showDetail();  } else {  clearForm();  }  }  }  public QLHopDong(String userName, String roles, String maPhong) {  initComponents();  loadTableDichVu();  loadToTableHopDong();  this.username = userName;  this.roles = roles;  this.maPhong = maPhong;  if (this.roles.equalsIgnoreCase("NV")) {  btn\_Delete.setEnabled(false);  btn\_Update.setEnabled(false);  }  if (maPhong.equals("")) {  if (listHopDong.size() >= 0) {  index = 0;  showDetail();  }  } else {  clearForm();  txt\_MaPhong.setText(maPhong);  txt\_MaNguoiTao.getText();  }  }  private void btn\_saveActionPerformed(java.awt.event.ActionEvent evt) {  try {  insertHopDong();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  e.printStackTrace();  }  }  private void tbl\_HopDongMouseClicked(java.awt.event.MouseEvent evt) {  try {  index = tbl\_HopDong.getSelectedRow();  if (index >= 0) {  showDetail();  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  }  private void btn\_SearchActionPerformed(java.awt.event.ActionEvent evt) {  try {  loadToTableHopDongByMAPhong();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  }  private void btn\_UpdateActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.updateHopDong();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  }  private void btn\_DeleteActionPerformed(java.awt.event.ActionEvent evt) {  try {  deleteHopDong();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  }  private void btn\_NewActionPerformed(java.awt.event.ActionEvent evt) {  try {  clearForm();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  }  private void btn\_ThanhToanActionPerformed(java.awt.event.ActionEvent evt) {  try {  List<HoaDonDichVu> listHDDV = DAOHDDV.findMaHDDVByMaPhong(txt\_MaPhong.getText());  maSuDung = listHDDV.get(0).getSuDung();  ThanhToan ttdv = new ThanhToan(username, txt\_MaPhong.getText(), txt\_MaHDDV.getText(), maSuDung);  FrMain.main.getDsk\_main().add(ttdv);  ttdv.show();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  }  private void btn\_ChonMaDVActionPerformed(java.awt.event.ActionEvent evt) {  try {  int idx = tbl\_BangDV.getSelectedRow();  if (idx >= 0) {  if (txt\_MaDV.getText().equals("")) {  txt\_MaDV.setText(modelDV.getValueAt(idx, 0).toString());  } else if (!txt\_MaDV.getText().equals("")) {  txt\_MaDV.setText(txt\_MaDV.getText() + ", " + modelDV.getValueAt(idx, 0).toString());  }  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps! Error: " + e);  }  } |

### Màn hình quản lý dịch vụ



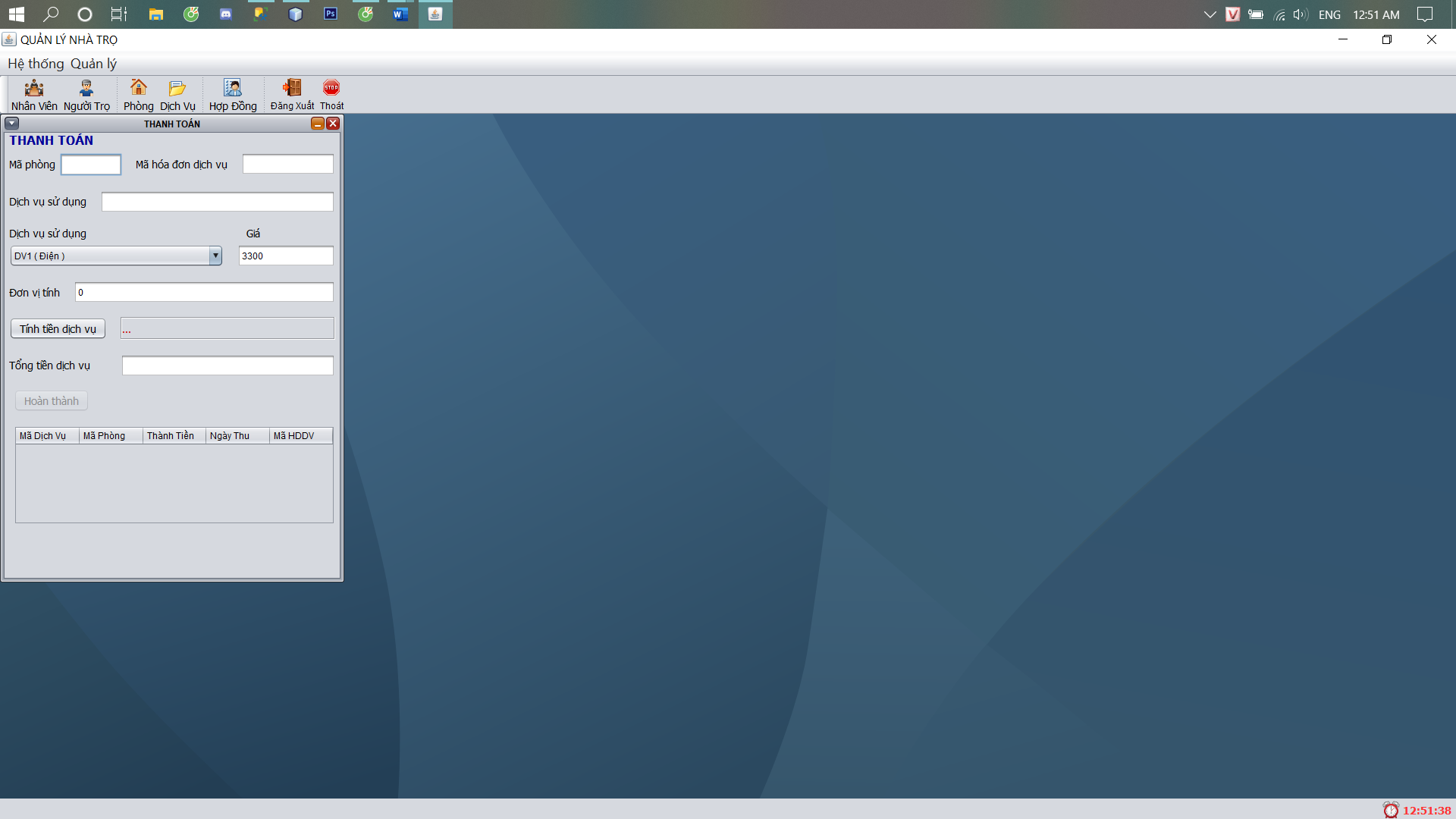
|  |
| --- |
| package com.sample.ui;  import java.util.ArrayList;  import java.util.List;  import javax.swing.table.DefaultTableModel;  import com.sample.DAO.DichVuDAO;  import com.sample.model.DichVu;  import javax.swing.JOptionPane;  public class FrDichVu extends javax.swing.JInternalFrame {  DefaultTableModel model;  int index = 0;  List<DichVu> listDV = new ArrayList<>();  DichVuDAO dichVuDAO = new DichVuDAO();  String maDV = "";  String tenDV = "";  String giaDV = "";  public boolean checkNull() {  giaDV = txt\_GiaDV.getText();  String inputPrice = "\\d{4,6}";  if(giaDV.equals("")){  JOptionPane.showConfirmDialog(this, "Không bỏ trống giá");  return false;  }  if (!giaDV.matches(inputPrice)) {  JOptionPane.showConfirmDialog(this, "Giá không hợp lệ!!! giá phải từ 1000-999999");  return false;  }  return true;  }  public void loadTableDichVu() {  model = (DefaultTableModel) tbl\_DichVu.getModel();  model.setRowCount(0);  try {  listDV = dichVuDAO.select();  for (DichVu dv : listDV) {  model.addRow(new Object[]{dv.getMaDV(), dv.getTenDV(), dv.getGia()});  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ dữ liệu vào bảng" + e);  e.printStackTrace();  }  }  public void showDetail() {  DichVu kt = listDV.get(index);  txt\_maDV.setText(kt.getMaDV());  txt\_TenDV.setText(kt.getTenDV());  txt\_GiaDV.setText(kt.getGia() + "");  tbl\_DichVu.setRowSelectionInterval(index, index);  }  public void insertDV(){  try {  if(checkNull()){  DichVu dv = new DichVu(txt\_maDV.getText(),txt\_TenDV.getText(), Integer.parseInt(txt\_GiaDV.getText()));  dichVuDAO.insert(dv);  loadTableDichVu();  JOptionPane.showMessageDialog(this, "Insert dịch vụ thành công!!!");  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: "+e);  }  }  public void clearForm(){  txt\_maDV.setText("");  txt\_TenDV.setText("");  txt\_GiaDV.setText("");  txt\_maDV.setEditable(true);  }  public FrDichVu() {  initComponents();  loadTableDichVu();  if (listDV.size() > 0) {  index = 0;  showDetail();  }  }  private void tbl\_DichVuMouseClicked(java.awt.event.MouseEvent evt) {  try {  index = tbl\_DichVu.getSelectedRow();  if (index >= 0) {  showDetail();  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_UpdateActionPerformed(java.awt.event.ActionEvent evt) {  try {  if(checkNull()){  DichVu dv = new DichVu(maDV, tenDV, Integer.parseInt(txt\_GiaDV.getText()));  dichVuDAO.update(dv);  loadTableDichVu();  JOptionPane.showMessageDialog(this, "Update giá thành công!!!");  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: "+e);  }  }  private void btn\_InsertActionPerformed(java.awt.event.ActionEvent evt) {  this.insertDV();  }  private void btn\_NewActionPerformed(java.awt.event.ActionEvent evt) {  this.clearForm();  } |

### Màn hình quản lý khách trọ



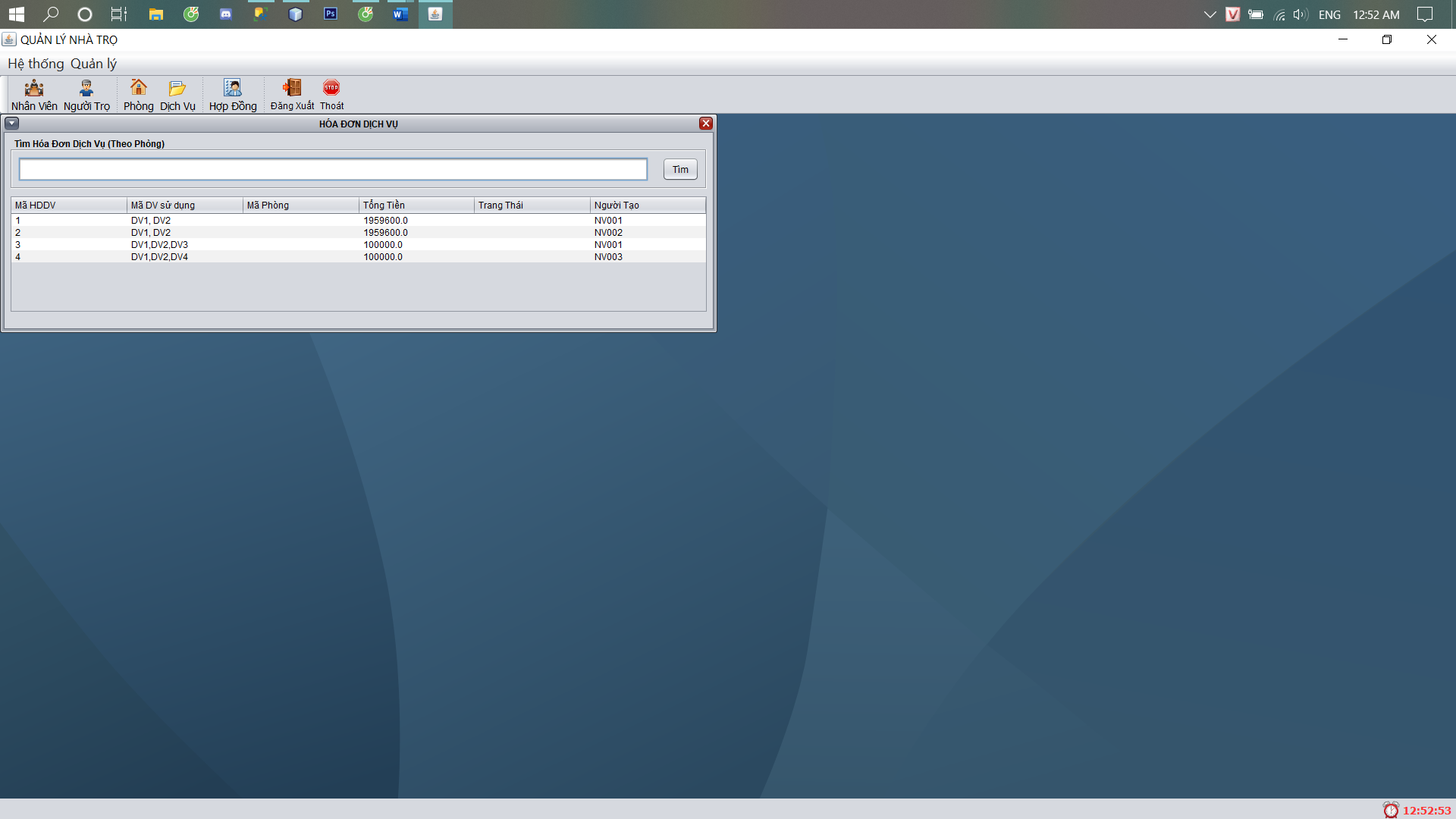
|  |
| --- |
| package com.sample.ui;  import com.sample.DAO.KhachTroDAO;  import com.sample.model.KhachTro;  import java.util.ArrayList;  import java.util.List;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel;  public class QLKhachTro extends javax.swing.JInternalFrame {  String roles;  DefaultTableModel model;  int index = 0;  List<KhachTro> listKhachTro = new ArrayList<>();  KhachTroDAO KhachTroDAO = new KhachTroDAO();  String maKT = "";  String tenKT = "";  String soDT = "";  String soCMT = "";  String eMail = "";  String maPhongThue = "";  public boolean checkRepeat(String maKTCheck) {  boolean trung = false;  for (int i = 0; i < listKhachTro.size(); i++) {  if (maKTCheck.equalsIgnoreCase(listKhachTro.get(i).getMaKT())) {  JOptionPane.showMessageDialog(this, "Trùng mã Nhân Viên");  trung = false;  return trung;  } else {  trung = true;  }  }  return trung;  }  public boolean checkNull() {  maKT = txt\_MaKT.getText();  tenKT = txt\_TenKT.getText();  soDT = txt\_SoDT.getText();  soCMT = txt\_SoCMND.getText();  eMail = txt\_Email.getText();  maPhongThue = txt\_maPhongThue.getText();  String inputSoDT = "\\d{9,10}";  String inputSoCMT = "\\d{12}";  String eMailForm = "\\w+\\w@+\\w.+\\w";  if (maKT.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống mã Khách Trọ!!");  return false;  } else if (tenKT.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống Họ tên!!");  return false;  } else if (soDT.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống SĐT");  return false;  } else if (!(soDT).matches(inputSoDT)) {  JOptionPane.showMessageDialog(this, "Số điện thoại phải là số từ 9 đến 10 số");  return false;  } else if (soCMT.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống số CMT");  return false;  } else if (!(soCMT).matches(inputSoCMT)) {  JOptionPane.showMessageDialog(this, "Số CMT phải là 12 số");  return false;  } else if (eMail.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống email");  return false;  } else if (!eMail.matches(eMailForm)) {  JOptionPane.showMessageDialog(this, "sai định dang, định dạng email: abc@abc.abc");  return false;  } else if (maPhongThue.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống Mã phòng thuê");  return false;  }  return true;  }  public void loadTableKhachTro() {  model = (DefaultTableModel) tbl\_KhachTro.getModel();  model.setRowCount(0);  try {  listKhachTro = KhachTroDAO.select();  for (KhachTro kt : listKhachTro) {  model.addRow(new Object[]{kt.getMaKT(), kt.getTenKT(), kt.getGioiTinh(), kt.getSoDT(), kt.getSoCMT(), kt.geteMail(), kt.getMaPhongThue()});  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ dữ liệu vào bảng" + e);  e.printStackTrace();  }  }  public void showDetail() {  KhachTro kt = listKhachTro.get(index);  txt\_MaKT.setText(kt.getMaKT());  txt\_TenKT.setText(kt.getTenKT());  String gioiTinh = kt.getGioiTinh();  if (gioiTinh.equalsIgnoreCase("Nam")) {  rb\_Nam.setSelected(true);  rb\_Nu.setSelected(false);  } else {  rb\_Nam.setSelected(false);  rb\_Nu.setSelected(true);  }  txt\_SoDT.setText(kt.getSoDT());  txt\_SoCMND.setText(kt.getSoCMT());  txt\_Email.setText(kt.geteMail());  txt\_maPhongThue.setText(kt.getMaPhongThue());  tbl\_KhachTro.setRowSelectionInterval(index, index);  }  public void clearForm() {  txt\_MaKT.setEditable(true);  txt\_TenKT.setEditable(true);  txt\_MaKT.setText("");  txt\_TenKT.setText("");  txt\_SoDT.setText("");  txt\_SoCMND.setText("");  txt\_Email.setText("");  rb\_Nam.setSelected(true);  rb\_Nu.setSelected(false);  txt\_maPhongThue.setText("");  }  public void insertKhachTro() {  if (checkRepeat(txt\_MaKT.getText()) && checkNull()) {  String gioiTinh = "";  if (rb\_Nam.isSelected()) {  gioiTinh = "Nam";  } else {  gioiTinh = "Nữ";  }  KhachTro objKhachTro = new KhachTro(maKT, tenKT, gioiTinh, soDT, soCMT, eMail, maPhongThue,"");  KhachTroDAO.insert(objKhachTro);  loadTableKhachTro();  JOptionPane.showMessageDialog(this, "Thêm thành công!!");  }  }  public void updateNhanVien() {  try {  if (checkNull()) {  KhachTro objKhachTro = new KhachTro(maKT, tenKT, null, soDT, soCMT, eMail, maPhongThue,null);  KhachTroDAO.update(objKhachTro);  loadTableKhachTro();  JOptionPane.showMessageDialog(this, "Update thành công");  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi update" + e);  e.printStackTrace();  }  }  public void deleteNguoiHoc() {  KhachTro objKhachTro = new KhachTro(txt\_MaKT.getText(), tenKT, null, soDT, soCMT, eMail, maPhongThue,null);  KhachTroDAO.delete(objKhachTro);  JOptionPane.showMessageDialog(this, "Xóa thành công!!");  loadTableKhachTro();  if (listKhachTro.size() > 0) {  if (index == listKhachTro.size()) {  index--;  }  showDetail();  } else {  clearForm();  }  }  public void findByMaKT() {  DefaultTableModel model = (DefaultTableModel) tbl\_KhachTro.getModel();  model.setRowCount(0);  try {  String maKTS = txt\_Search.getText();  List<KhachTro> list = KhachTroDAO.findByMaKT(maKTS);  for (KhachTro kt : list) {  model.addRow(new Object[]{kt.getMaKT(), kt.getTenKT(), kt.getGioiTinh(), kt.getSoDT(), kt.getSoCMT(), kt.geteMail(), kt.getMaPhongThue()});  }  if (list.size() > 0) {  JOptionPane.showMessageDialog(this, "Tìm thành công");  index = 0;  tbl\_KhachTro.setRowSelectionInterval(index, index);  showDetail();  } else {  loadTableKhachTro();  JOptionPane.showMessageDialog(this, "Không có thông tin mã khách trọ này");  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi câu truy vấn " + e);  }  }  private void troVeFirst() {  try {  index = 0;  showDetail();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi: " + e);  }  }  private void troVeTruoc() {  try {  if (index > 0) {  index--;  showDetail();  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi: " + e);  }  }  private void troVeSau() {  try {  if (index < listKhachTro.size() - 1) {  index++;  showDetail();  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "đang ở cuối rồi ");  }  }  private void troVeLast() {  try {  index = listKhachTro.size() - 1;  showDetail();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi: " + e);  }  }  public QLKhachTro(String roles) {  this.roles = roles;  initComponents();  loadTableKhachTro();  if (listKhachTro.size() >= 0) {  index = 0;  showDetail();  }  }  private void tbl\_KhachTroMouseClicked(java.awt.event.MouseEvent evt) {  try {  index = tbl\_KhachTro.getSelectedRow();  if (index >= 0) {  showDetail();  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_UpdateActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.updateNhanVien();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_SaveActionPerformed(java.awt.event.ActionEvent evt) {  try {  if (checkRepeat(txt\_MaKT.getText()) && checkNull()) {  this.insertKhachTro();  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_NewActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.clearForm();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_SearchActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.findByMaKT();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_LastActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.troVeLast();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_NextActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.troVeSau();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_FistActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.troVeFirst();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  }  private void btn\_PreviousActionPerformed(java.awt.event.ActionEvent evt) {  try {  this.troVeTruoc();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  } |

### Màn hình thanh toán



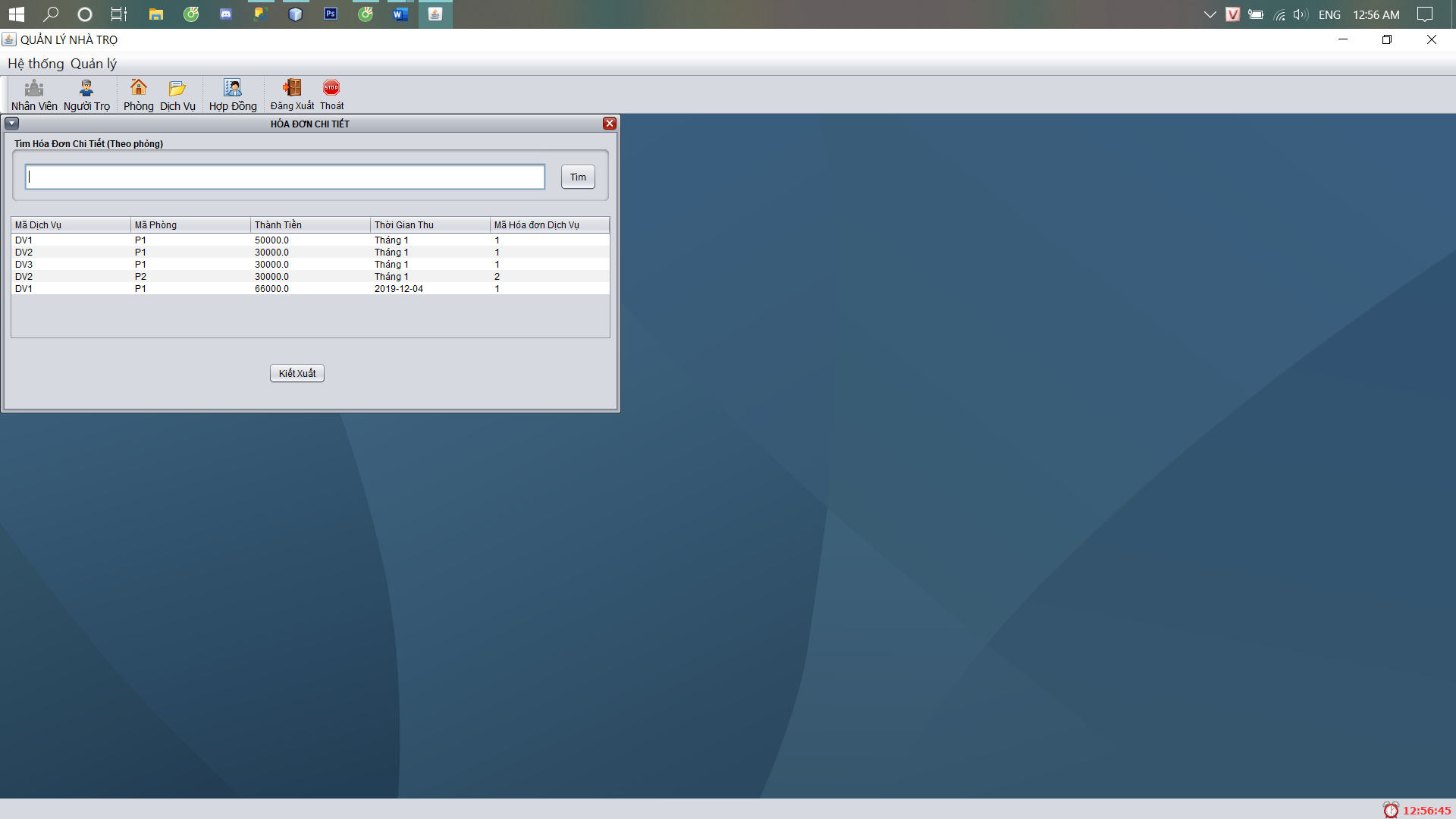
|  |
| --- |
| package com.sample.ui;  import com.sample.DAO.DichVuDAO;  import com.sample.DAO.HoaDonChiTietDAO;  import com.sample.DAO.HoaDonDichVuDAO;  import com.sample.model.DichVu;  import com.sample.model.HoaDonDichVu;  import com.sample.model.HoaDonChiTiet;  import java.io.BufferedOutputStream;  import java.io.FileOutputStream;  import java.util.ArrayList;  import java.util.List;  import javax.swing.DefaultComboBoxModel;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel;  import org.apache.poi.ss.usermodel.Sheet;  import org.apache.poi.xssf.usermodel.XSSFCell;  import org.apache.poi.xssf.usermodel.XSSFRow;  import org.apache.poi.xssf.usermodel.XSSFSheet;  import org.apache.poi.xssf.usermodel.XSSFWorkbook;  public class ThanhToan extends javax.swing.JInternalFrame {  String userName;  String maPhong;  String maHDDV;  String DVSD;  float thanhTienDV;  float tongTien;  String donViTinh;  DichVuDAO dichVuDAO = new DichVuDAO();  HoaDonDichVuDAO DAO\_HDDV = new HoaDonDichVuDAO();  HoaDonChiTietDAO DAO\_HDCT = new HoaDonChiTietDAO();  DefaultTableModel model;  List<HoaDonChiTiet> listHDCT = new ArrayList<>();  public void LoadToCombo() {  DefaultComboBoxModel model = (DefaultComboBoxModel) cbx\_DVSD.getModel();  model.removeAllElements();  try {  List<DichVu> list = dichVuDAO.select();  for (DichVu dv : list) {  model.addElement(dv);  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi truy vấn dữ liệu!");  }  }  public void ExportFileExxcel() {  FileOutputStream fos = null;  XSSFWorkbook exelWorkbook = null;  BufferedOutputStream bfos = null;  try {  exelWorkbook = new XSSFWorkbook();  XSSFSheet excelSheet = exelWorkbook.createSheet("HDCT");  for (int i = 0; i < model.getRowCount(); i++) {  Sheet firstSheet = exelWorkbook.getSheetAt(0);  XSSFRow exCellRow = excelSheet.createRow(i);  for (int j = 0; j < model.getColumnCount(); j++) {  XSSFCell eCell = exCellRow.createCell(j);  eCell.setCellValue(model.getValueAt(i, j).toString());  }  }  fos = new FileOutputStream("D:\\" + txt\_MaPhong.getText() + ".xlsx");  bfos = new BufferedOutputStream(fos);  exelWorkbook.write(bfos);  exelWorkbook.close();  bfos.close();  fos.close();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "lỗi: " + e);  }  }  public void loadToTable() {  model = (DefaultTableModel) tbl\_ThanhToan.getModel();  try {  DichVu dv = (DichVu) cbx\_DVSD.getSelectedItem();  String maDV = "";  long millis = System.currentTimeMillis();  java.sql.Date date = new java.sql.Date(millis);  maDV = String.valueOf(dv.getMaDV());  model.addRow(new Object[]{maDV, txt\_MaPhong.getText(), lbl\_ShowTien.getText(), date, txt\_MaHDDV.getText()});  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ dữ liệu vào bảng " + e);  e.printStackTrace();  }  }  public void clearForm() {  txt\_DVTinh.setText("0");  btn\_ThuTien.setEnabled(false);  }  public boolean checkGia() {  donViTinh = txt\_DVTinh.getText();  String inputSo = "\\d{1,4}";  if (!donViTinh.matches(inputSo)) {  JOptionPane.showMessageDialog(this, "Số không hợp lệ!! Giá từ 1->9999");  return false;  }  return true;  }  public void insertHoaDonChiTiet() {  if (checkGia()) {  long millis = System.currentTimeMillis();  java.sql.Date date = new java.sql.Date(millis);  HoaDonChiTiet objHDCT = new HoaDonChiTiet(0, cbx\_DVSD.getSelectedItem().toString(), txt\_MaPhong.getText(), thanhTienDV, date + "", Integer.parseInt(txt\_MaHDDV.getText()));  DAO\_HDCT.insert(objHDCT);  JOptionPane.showMessageDialog(this, "Thanh toán thành công!!!");  btn\_ThuTien.setEnabled(false);  }  }  public ThanhToan(String userName, String maPhong, String maHDDV, String DVSD) {  initComponents();  LoadToCombo();  btn\_ThuTien.setEnabled(false);  this.userName = userName;  this.maPhong = maPhong;  this.maHDDV = maHDDV;  this.DVSD = DVSD;  if (maPhong.equals("") && maHDDV.equals("") && DVSD.equals("")) {  clearForm();  } else {  txt\_MaPhong.setText(maPhong);  txt\_MaHDDV.setText(maHDDV);  txt\_DVSD.setText(DVSD);  }  }  private void btn\_ThuTienActionPerformed(java.awt.event.ActionEvent evt) {  try {  int confirm = JOptionPane.showConfirmDialog(this, "Bạn có muốn hoàn thành?");  if (confirm == JOptionPane.YES\_OPTION) {  HoaDonDichVu objHDDV = new HoaDonDichVu(0, txt\_DVSD.getText(), txt\_MaPhong.getText(), Float.parseFloat(txt\_TongTien.getText()), "Đã thanh toán", userName);  DAO\_HDDV.update(objHDDV);  ExportFileExxcel();  JOptionPane.showMessageDialog(this, "Thu tiền thành công");  this.dispose();  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  e.printStackTrace();  }  }  private void btn\_ThanhTienActionPerformed(java.awt.event.ActionEvent evt) {  try {  thanhTienDV = Float.parseFloat(txt\_DVTinh.getText()) \* Float.parseFloat(txt\_GiaDVSD.getText());  lbl\_ShowTien.setText(thanhTienDV + "");  if (txt\_TongTien.getText().equals("")) {  txt\_TongTien.setText(thanhTienDV + "");  } else {  txt\_TongTien.setText(Float.parseFloat(txt\_TongTien.getText()) + thanhTienDV + "");  }  loadToTable();  insertHoaDonChiTiet();  btn\_ThuTien.setEnabled(true);  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi tính toán " + e);  }  }  private void cbx\_DVSDItemStateChanged(java.awt.event.ItemEvent evt) {  try {  DichVu dv = (DichVu) cbx\_DVSD.getSelectedItem();  txt\_GiaDVSD.setText(String.valueOf(dv.getGia()));  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi: " + e);  }  } |

### Màn hình hóa đơn dịch vụ



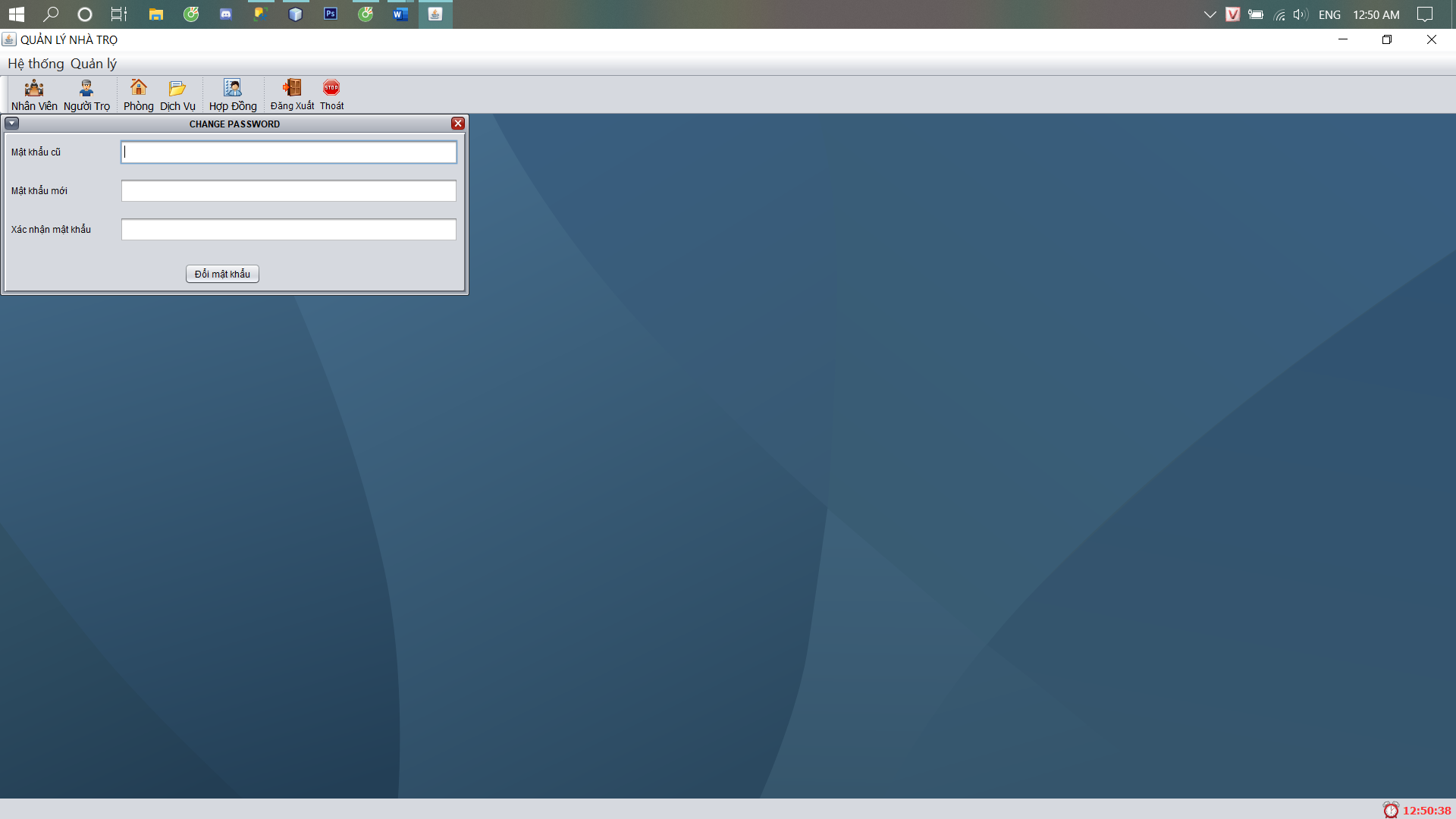
|  |
| --- |
| package com.sample.ui;  import com.sample.DAO.HoaDonDichVuDAO;  import com.sample.model.HoaDonDichVu;  import java.util.ArrayList;  import java.util.List;  import javax.swing.JOptionPane;  import javax.swing.table.DefaultTableModel;  public class FrHoaDonDV extends javax.swing.JInternalFrame {  DefaultTableModel model;  List<HoaDonDichVu> list = new ArrayList<>();  HoaDonDichVuDAO DAO\_HDDV = new HoaDonDichVuDAO();  public void loadToTableHDCT() {  model = (DefaultTableModel) tbl\_HDDV.getModel();  model.setRowCount(0);  try {  list = DAO\_HDDV.select();  for (HoaDonDichVu hddv : list) {  model.addRow(new Object[]{  hddv.getMaHDDV(),  hddv.getSuDung(),  hddv.getMaPhong(),  hddv.getTongTien(),  hddv.getTrangThai(),  hddv.getNguoiTao()  });  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ dữ liệu vào bảng" + e);  e.printStackTrace();  }  }  public void loadToTableHDCTByMaPhong() {  model = (DefaultTableModel) tbl\_HDDV.getModel();  model.setRowCount(0);  try {  List<HoaDonDichVu> listHDCT = DAO\_HDDV.findMaHDDVByMaPhong(txt\_maPhong.getText());  for (HoaDonDichVu hddv : list) {  model.addRow(new Object[]{  hddv.getMaHDDV(),  hddv.getSuDung(),  hddv.getMaPhong(),  hddv.getTongTien(),  hddv.getTrangThai(),  hddv.getNguoiTao()  });  }  if (listHDCT.size() > 0) {  JOptionPane.showMessageDialog(this, "Tìm thành công");  } else {  loadToTableHDCT();  JOptionPane.showMessageDialog(this, "Không có thông tin mã phòng này");  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ dữ liệu vào bảng" + e);  e.printStackTrace();  }  }  public FrHoaDonDV() {  initComponents();  loadToTableHDCT();  }  private void btn\_TimActionPerformed(java.awt.event.ActionEvent evt) {  try {  loadToTableHDCTByMaPhong();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi: " + e);  }  } |

### Màn hình hóa đơn chi tiết



|  |
| --- |
| package com.sample.ui;  import com.sample.DAO.HoaDonChiTietDAO;  import com.sample.model.HoaDonChiTiet;  import java.io.BufferedOutputStream;  import java.io.FileOutputStream;  import java.util.ArrayList;  import java.util.List;  import javax.swing.JFileChooser;  import javax.swing.JOptionPane;  import javax.swing.filechooser.FileNameExtensionFilter;  import javax.swing.table.DefaultTableModel;  import org.apache.poi.ss.usermodel.Sheet;  import org.apache.poi.xssf.usermodel.XSSFCell;  import org.apache.poi.xssf.usermodel.XSSFRow;  import org.apache.poi.xssf.usermodel.XSSFSheet;  import org.apache.poi.xssf.usermodel.XSSFWorkbook;  public class FrHoaDonChiTiet extends javax.swing.JInternalFrame {  DefaultTableModel model;  List<HoaDonChiTiet> list = new ArrayList<>();  HoaDonChiTietDAO DAO\_HDCT = new HoaDonChiTietDAO();  public void loadToTableHDCT() {  model = (DefaultTableModel) tbl\_HDCT.getModel();  model.setRowCount(0);  try {  list = DAO\_HDCT.select();  for (HoaDonChiTiet hdct : list) {  model.addRow(new Object[]{  hdct.getMaDVSD(),  hdct.getMaPhong(),  hdct.getThanhTien(),  hdct.getThoiGian(),  hdct.getMaHDDV()  });  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ dữ liệu vào bảng" + e);  e.printStackTrace();  }  }  public void loadToTableHDCTByMaPhong() {  model = (DefaultTableModel) tbl\_HDCT.getModel();  model.setRowCount(0);  try {  List<HoaDonChiTiet> listHDCT = DAO\_HDCT.findMaHDCTByMaPhong(txt\_maPhong.getText());  for (HoaDonChiTiet hdct : list) {  model.addRow(new Object[]{  hdct.getMaDVSD(),  hdct.getMaPhong(),  hdct.getThanhTien(),  hdct.getThoiGian(),  hdct.getMaHDDV()  });  }  if (listHDCT.size() > 0) {  JOptionPane.showMessageDialog(this, "Tìm thành công");  } else {  loadToTableHDCT();  JOptionPane.showMessageDialog(this, "Không có thông tin mã phòng này");  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi đổ dữ liệu vào bảng" + e);  e.printStackTrace();  }  }  public FrHoaDonChiTiet() {  initComponents();  loadToTableHDCT();  }  private void btn\_TimActionPerformed(java.awt.event.ActionEvent evt) {  try {  loadToTableHDCTByMaPhong();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Lỗi: " + e);  }  }  private void btn\_KietXuatActionPerformed(java.awt.event.ActionEvent evt) {  FileOutputStream fos = null;  XSSFWorkbook exelWorkbook = null;  BufferedOutputStream bfos = null;  try {  JFileChooser exelFileChoser = new JFileChooser("D:\\");  exelFileChoser.setDialogTitle("Save as ..");  FileNameExtensionFilter fnef = new FileNameExtensionFilter("Files", "xls", "xlsx", "slsm");  exelFileChoser.setFileFilter(fnef);  int chooser = exelFileChoser.showSaveDialog(null);  if (chooser == JFileChooser.APPROVE\_OPTION) {  exelWorkbook = new XSSFWorkbook();  XSSFSheet excelSheet = exelWorkbook.createSheet("HDCT");  for (int i = 0; i < model.getRowCount(); i++) {  Sheet firstSheet = exelWorkbook.getSheetAt(0);  XSSFRow exCellRow = excelSheet.createRow(i);  for (int j = 0; j < model.getColumnCount(); j++) {  XSSFCell eCell = exCellRow.createCell(j);  eCell.setCellValue(model.getValueAt(i, j).toString());  }  }  }  if(chooser == JFileChooser.CANCEL\_OPTION) {  JOptionPane.showMessageDialog(this, "Lưu không thành công");  return;  }  fos = new FileOutputStream(exelFileChoser.getSelectedFile() + ".xlsx");  bfos = new BufferedOutputStream(fos);  exelWorkbook.write(bfos);  JOptionPane.showMessageDialog(this, "Thành công");  exelWorkbook.close();  bfos.close();  fos.close();  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "lỗi: " + e);  }  } |

### Màn hình change password



|  |
| --- |
| package com.sample.ui;  import com.sample.DAO.UserDAO;  import com.sample.model.User;  import javax.swing.JOptionPane;  public class ChangePass extends javax.swing.JInternalFrame {  String userName;  UserDAO usDAO = new UserDAO();  String oldPass = "";  String newPass = "";  String confirmPass = "";  public boolean checkError() {  oldPass = new String(ps\_OldPass.getPassword());  newPass = new String(ps\_NewPass.getPassword());  confirmPass = new String(ps\_ConfirmNewPass.getPassword());  if (oldPass.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống mật khẩu cũ");  return false;  } else if (newPass.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống mật khẩu mới");  return false;  } else if (confirmPass.equals("")) {  JOptionPane.showMessageDialog(this, "Không bỏ trống Xác nhận mật khẩu");  return false;  } else if (!confirmPass.equals(newPass)) {  JOptionPane.showMessageDialog(this, "Mật khẩu xác nhận không trùng");  return false;  }  return true;  }  public ChangePass(String userName) {  this.userName = userName;  initComponents();  }  private void bnt\_ChangePassActionPerformed(java.awt.event.ActionEvent evt) {  try {  if (checkError()) {  int ask = JOptionPane.showConfirmDialog(this, "Bạn muốn đổi mật khẩu???");  if (ask == JOptionPane.YES\_OPTION) {  User objuser = new User(null, userName, newPass, null);  usDAO.update(objuser);  usDAO.convertStringtoPass(userName);  JOptionPane.showMessageDialog(this, "Đổi mật khẩu thành công!!");  this.dispose();  }  }  } catch (Exception e) {  JOptionPane.showMessageDialog(this, "Opps!! Error: " + e);  }  } |

# Kiểm thử

## Kiểm thử form X

## Kiểm thử form Y

# Đóng gói và triển khai

## Hướng dẫn chuyển đổi jar thành exe

* Sử dụng phần mềm EXE4J để chuyển đổi jar sang exe
* Sử dụng phần mềm InnoSetup để đóng gói

## Hướng dẫn cài đặt triển khai

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

## Hướng dẫn sử dụng phần mềm

* Đăng nhập với tài khoản username: longnn và password: 123
* Cấp các tài khoản có vai trò là chủ nhà
* Đăng nhập tài khoản mới và xóa tài khoản admin