|  |
| --- |
| `````**FPT POLYTECHNIC** |
| **DỰ ÁN 1**  **Quản lý kho hàng** |
| NGÀNH CÔNG NGHỆ THÔNG TIN (ỨNG DỤNG PHẦN MỀM) |



|  |
| --- |
| TP HCM, 2019 |

|  |
| --- |
| GIẢNG VIÊN HƯỚNG DẪN: TỐNG PHƯỚC QUAN |
| SINH VIÊN THỰC HIỆN |
| 1. Vương Thế Minh Thăng (PS09070), Trưởng nhóm 2. Đặng Công Hoàng Long (PS09098) 3. Hoàng Phi Long (PS09044) |

MỤC LỤC

[1 Phân tích 2](#_Toc27514938)

[1.1 Hiện trạng 2](#_Toc27514939)

[1.2 Yêu cầu hệ thống 2](#_Toc27514940)

[1.3 Use case 2](#_Toc27514941)

[2 Thiết kế 5](#_Toc27514942)

[2.1 Mô hình triển khai 5](#_Toc27514943)

[2.2 Thiết kế CSDL 6](#_Toc27514944)

[2.2.1 Sơ đồ quan hệ thực thể 6](#_Toc27514945)

[2.2.2 Thiết kế chi tiết các thực thể 7](#_Toc27514946)

[2.3 Thiết kế giao diện 11](#_Toc27514947)

[2.3.1 Sơ đồ tổ chức giao diện 11](#_Toc27514948)

[2.3.2 Thiết kế giao diện cho các chức năng nghiệp vụ 12](#_Toc27514949)

[3 Thực hiện viết mã 37](#_Toc27514950)

[3.1 Viết mã tạo CSDL 37](#_Toc27514951)

[3.1.1 Tạo CSDL 37](#_Toc27514952)

[3.1.2 SQL truy vấn và thao tác 42](#_Toc27514953)

[3.1.3 Các thủ tục lưu tổng hợp 42](#_Toc27514954)

[3.2 Lập trình hibernate 43](#_Toc27514955)

[3.2.1 Hibernate.config.xml 43](#_Toc27514956)

[3.2.2 Model class - Các lớp mô tả dữ liệu 44](#_Toc27514957)

[3.2.3 DAO Class - Các lớp truy xuất dữ liệu 47](#_Toc27514958)

[3.3 Viết mã cho ứng dụng 54](#_Toc27514959)

[3.4 Các lớp tiện ích 64](#_Toc27514960)

[4 Kiểm thử 66](#_Toc27514961)

[4.1 Kiểm thử form quản lý nhân viên 66](#_Toc27514962)

[4.2 Kiểm thử form quản lý khách hàng, nhà cung cấp 66](#_Toc27514963)

[4.3 Kiểm thử form quản lý sản phẩm 67](#_Toc27514964)

[4.4 Kiểm thử form quản lý đơn nhập, đơn xuất 67](#_Toc27514965)

[5 Đóng gói và triển khai 67](#_Toc27514966)

[5.1 Hướng dẫn chuyển đổi jar thành exe 67](#_Toc27514967)

[5.2 Hướng dẫn cài đặt triển khai 67](#_Toc27514968)

# Phân tích

## Hiện trạng

Công ty kho hàng FPT Polystorage đang nhập xuất kho các sản phẩm nước như Lavie, Aquafina, beer,... Việc quản lý Kho, nhân viên, bảng thống kê và hóa đơn đang thực hiện thông qua excel. Hiện công ty đang gặp khó khăn khi số lượng đơn hàng ngày một nhiều, dữ liệu ngày càng lớn nên việc quản lý excel gặp rất nhiều khó khăn, dễ sai sót mà không bảo mật.

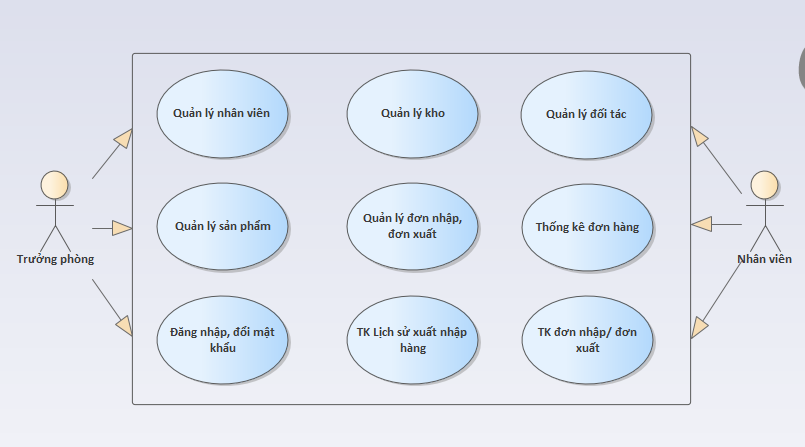
## Yêu cầu hệ thống

Xây dựng một phần mềm để giải quyết khó khăn trên.

* **Yêu cầu chức năng nghiệp vụ** 
  + Quản lý nhân viên
  + Quản lý sản phẩm
  + Quản lý đơn hàng
  + Quản lý Khách hàng
  + Quản lý nhà cung cấp
  + Quản lý hàng trong kho
  + Quản lý nhập hàng
  + Quản lý xuất hàng
* **Yêu cầu về bảo mật** 
  + Tất cả mọi thành viên phải đăng nhập mới sử dụng được phần mềm
  + Trưởng phòng đào tạo được phép thực hiện tất cả các chức năng
  + Nhân viên không được phép xóa dữ liệu và cũng không được xem thông tin về doanh thu
* **Yêu cầu về môi trường công nghệ** 
  + MySQL, hibernate, Swing, JDK 1.8

## Use case

Use case là sơ đồ tổng quan về mặt chức năng và phân vai trò người sử dụng. Dựa vào yêu cầu hệ thống của khách hàng, chúng ta có thể phác thảo sơ đồ use case như sau.

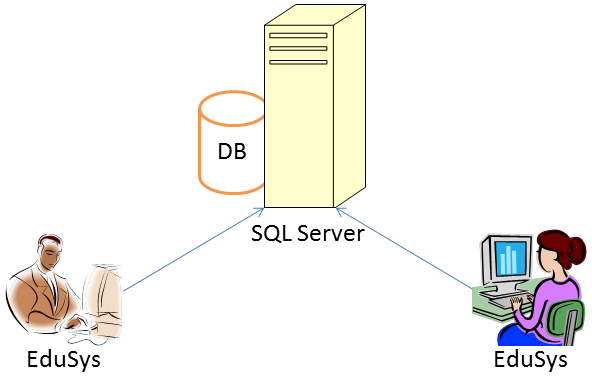


**QUẢN LÝ KHO HÀNG**

* **Chi tiết các chức năng**
  + Mỗi chức năng quản lý bao gồm các chức năng con
    - Xem: xem tất cả và chi tiết một mục
    - Thêm: thêm mới vào cơ sở dữ liệu
    - Xóa: xóa theo mã
    - Sửa: cập nhật dữ liệu đang xem
    - Tìm kiếm: tìm kiếm theo điều kiện
    - Điều hướng: di chuyển đến dữ liệu của bản ghi chi tiết trước và sau
    - Các chức năng thao tác dữ liệu (thêm, sửa, xóa) cần được kiểm lỗi hợp lý với dữ liệu
  + Chức năng đăng nhập và đăng xuất
    - Đăng nhập: để sử dụng phần mềm, bắt buộc phải đăng nhập
    - Đăng xuất: đăng nhập lại với người dùng khác hoặc dừng lại đi đâu đó
* **Chi tiết về yêu cầu bảo mật**
  + Tất cả phải đăng nhập mới được sử dụng các chức năng trong phần mềm
  + Trưởng phòng: được phép sử dụng tất cả chức năng
  + Nhân viên:
    - Không được phép thực hiện chức năng xóa trong các chức năng quản lý
    - Không được phép xem chức năng thống kê doanh thu
* **Thông tin các thực thể**
  + Nhân viên
    - Mã nhân viên(Tài khoản đăng nhập)
    - Họ và tên
    - Email
    - Giới tính
    - Số điện thoại
    - Mật khẩu
    - Hình
    - Vai trò
  + Sản phẩm
    - Mã sản phẩm
    - Tên sản phẩm
    - Mã nhà cung cấp
    - Mã loại
    - Đơn vị tính
    - Giá
  + Nhà cung cấp
    - Mã cung cấp
    - Tên cung cấp
    - Email
    - Số điện thoại
    - Địa chỉ
    - Ghi chú
  + Khách hàng
    - Mã khách hàng
    - Tên khách hàng
    - Email
    - Số điện thoại
    - Địa chỉ
    - Ghi chú
  + Loại
    - Mã loại
    - Tên loại
  + Kho
    - Mã kho
    - Địa chỉ
  + Chi tiết kho
    - Mã kho
    - Mã sản phẩm
    - Số lượng
  + Đơn nhập
    - Mã đơn nhập
    - Mã nhân viên
    - Ngày
    - Trạng thái
    - Mã kho
    - Mã nhà cung cấp
    - Ghi chú
  + Đơn xuất
    - Mã đơn xuất
    - Mã nhân viên
    - Ngày
    - Trạng thái
    - Mã kho
    - Mã khách hàng
    - Ghi chú

# Thiết kế

## Mô hình triển khai

Ứng dụng phần mềm được xây dựng để phục vụ cho nhiều người dùng nhưng cơ sở dữ liệu thì lưu trữ tập trung.

StorageSys

MY SQL

StorageSys

* Cần một máy cài My SQL . Máy này cần hệ điều hành window xp trở lên
* Các máy nhân viên cần JDK 1.8+ với hệ điều hành bất ký

## Thiết kế CSDL

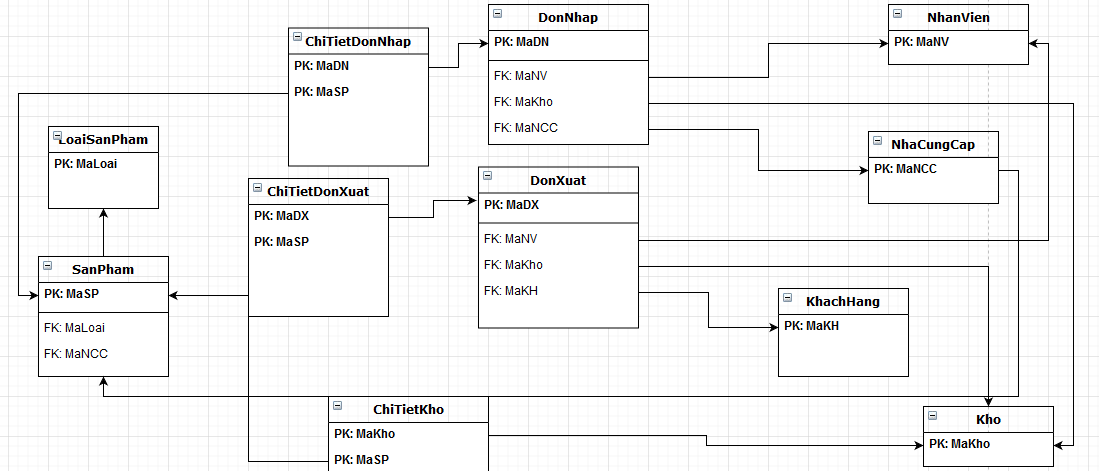
### Sơ đồ quan hệ thực thể

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

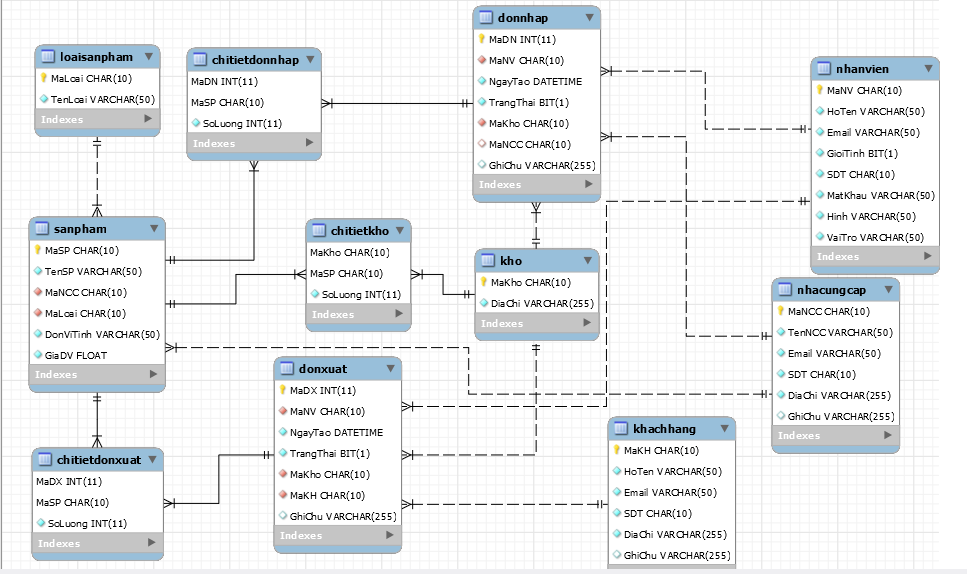
* Nhân viên: là người quản trị cần có thông tin đăng nhập đồng thời được ghi nhận mỗi khi nhân viên tạo
* Sản phẩm: quản lý sản phẩm
* Khách hàng: quản lý Khách
* Nhà cung cấp: quản lý nhà cung cấp
* Kho :là quản lí kho, sản phẩm trong kho
* Đơn nhập: là quản lí các đơn nhập kho
* Đơn nhập: là quản lí các đơn xuất kho

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

#### ERD Diagram level 1



#### ERD Diagram level 2



### Thiết kế chi tiết các thực thể

#### Nhân viên

Bảng NhanVien lưu thông tin nhân viên phòng đào tạo có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaNV | CHAR(10) | PK, NOT NULL | Mã nhân viên |
| HoTen | VARCHAR(50) | NOT NULL | Họ và tên |
| Email | VARCHAR(50) | NOT NULL | Email |
| GioiTinh | BIT(1) | DEFAULT 0, NOT NULL | Giới tính, 0 là nữ, 1 là nam |
| SDT | CHAR(10) | NOT NULL | Số điện thoại |
| MatKhau | VARCHAR(50) | NOT NULL | Mật khẩu |
| Hinh | VARCHAR(50) | NOT NULL | Hình |
| VaiTro | VARCHAR(50) | NOT NULL | Nhân viên hoặc trưởng phòng |

#### Nhà cung cấp

Bảng đối tác thông tin các chuyên đề có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaNCC | CHAR(10) | PK, NOT NULL | Mã nhà cung cấp |
| TenNCC | VARCHAR(50) | NOT NULL | Tên nhà cung cấp |
| Email | VARCHAR(50) | NOT NULL | Email |
| SDT | CHAR(10) | NOT NULL | Số điện thoại |
| DiaChi | VARCHAR(255) | NOT NULL | Đia chỉ |
| GhiChu | VARCHAR(255) |  | Ghi chú |

#### Khách hàng

Bảng đối tác thông tin các chuyên đề có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaKH | CHAR(10) | PK, NOT NULL | Mã khách hàng |
| HoTen | VARCHAR(50) | NOT NULL | Tên khách hàng |
| Email | VARCHAR(50) | NOT NULL | Email |
| SDT | CHAR(10) | NOT NULL | Số điện thoại |
| DiaChi | VARCHAR(255) | NOT NULL | Đia chỉ |
| GhiChu | VARCHAR(255) |  | Ghi chú |

#### Loại

Bảng Loại thông tin các chuyên đề có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaLoai | CHAR(10) | PK, NOT NULL | Mã loại |
| TenLoai | VARCHAR(50) | NOT NULL | Tên loại |

#### Sản phẩm

Bảng sản phẩm thông tin các chuyên đề có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaSP | CHAR(10) | PK, NOT NULL | Mã sản phẩm |
| TenSP | VARCHAR(50) | NOT NULL | Tên sản phẩm |
| MaDT | CHAR(10) | FK, NOT NULL | Mã đối tác |
| MaLoai | CHAR(10) | FK, NOT NULL | Mã loại |
| DonViTinh | VARCHAR(50) | NOT NULL | Đơn vị tính |
| GiaDV | FLOAT | NOT NULL | Giá đơn vị |

#### Kho

Bảng Kho thông tin các chuyên đề có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| Makho | CHAR(10) | PK, NOT NULL | Mã kho |
| DiaChi | VARCHAR(255) | NOT NULL | Đia chỉ |

#### Chi tiết kho

Bảng chi tiết kho hàng thông tin các chuyên đề có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| Makho | CHAR(10) | PK, NOT NULL | Mã kho |
| MaSP | CHAR(10) | PK, NOT NULL | Mã sản phẩm |
| SoLuong | INT(11) | NOT NULL | Số lượng |

#### Đơn nhập

Bảng đơn hàng thông tin các chuyên đề có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaDN | INT(11) | PK, NOT NULL | Mã đơn nhập |
| MaNV | CHAR(10) | FK, NOT NULL | Mã nhân viên tạo đơn |
| NgayNhap | DATETIME | NOT NULL | Ngày tạo đơn |
| Trạng thái | BIT(1) | DEFAULT 0 | 0 là đang chờ, 1 là đã xác nhận |
| Makho | CHAR(10) | PK, NOT NULL | Mã kho |
| MaNCC | CHAR(10) | FK, NOT NULL | Mã nhà cung cấp |
| GhiChu | VARCHAR(255) |  | Ghi chú |

#### Chi tiết đơn nhập

Bảng chi tiết đơn hàng thông tin các chuyên đề có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaDN | INT(11) | PK, NOT NULL | Mã đơn hàng |
| MaSP | CHAR(10) | PK, NOT NULL | Mã sản phẩm |
| SoLuong | INT(11) | NOT NULL | Số lượng |

#### Đơn xuất

Bảng đơn hàng thông tin các chuyên đề có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaDX | INT(11) | PK, NOT NULL | Mã đơn xuất |
| MaNV | CHAR(10) | FK, NOT NULL | Mã nhân viên tạo đơn |
| NgayNhap | DATETIME | NOT NULL | Ngày tạo đơn |
| Trạng thái | BIT(1) | DEFAULT 0 | 0 là đang chờ, 1 là đã xác nhận |
| Makho | CHAR(10) | PK, NOT NULL | Mã kho |
| MaKH | CHAR(10) | FK, NOT NULL | Mã khách hàng |
| GhiChu | VARCHAR(255) |  | Ghi chú |

#### Chi tiết đơn nhập

Bảng chi tiết đơn hàng thông tin các chuyên đề có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaDN | INT(11) | PK, NOT NULL | Mã đơn hàng |
| MaSP | CHAR(10) | PK, NOT NULL | Mã sản phẩm |
| SoLuong | INT(11) | NOT NULL | Số lượng |

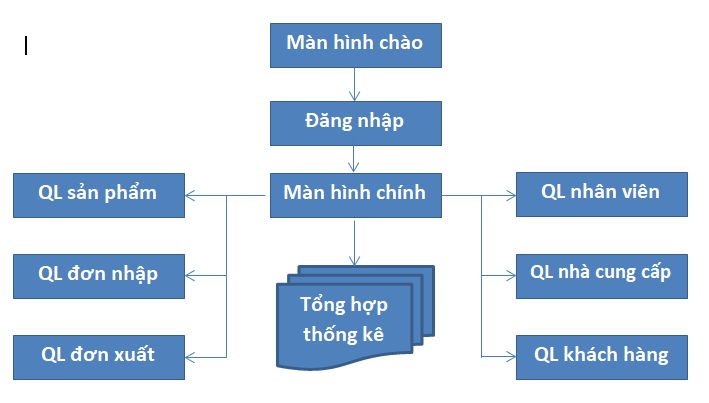
## Thiết kế giao diện

### Sơ đồ tổ chức giao diện

Giao diện chính là một cửa sổ chứa menu chính và toolbar. Thông qua đó để đi đến các giao diện thành viên để thực hiện các chức năng trong hệ thống.

Theo yêu cầu thì mọi nhân viên phải đăng nhập trước khi sử dụng ứng dụng nên form đăng nhập xuất hiện trước để yêu cầu đăng nhập.

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



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

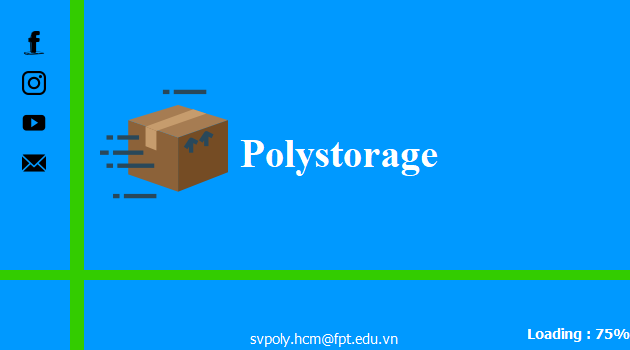
Trong phần thiết kế giao diện các bạn nên sử dụng gói hình ảnh được cung cấp sẵn trên tài nguyên học tập của LMS. Nếu các bạn không muốn sử dụng gói hình ảnh này thì bạn có thể chọn gói hình ảnh phù hợp của các bạn.

Lưu ý quan trọng:

* Các bạn phải đặt tên đúng như hướng dẫn vì các tên này ảnh hưởng đến việc viết mã sau này. Nếu các bạn đặt tên khác thì phải chỉnh mã sau này theo tên mà các bạn đã đặt.
* Trong lập trình giao diện người ta thường qui ước tiếp đầu ngữ của tên các thành phần giao diện như sau

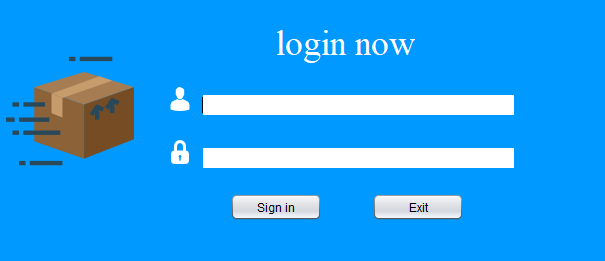
|  |  |  |
| --- | --- | --- |
| **Control** | **Prefix** | **Mô tả** |
| JLabel | lbl | Nhãn |
| JButton | btn | Nút |
| JTextField, JTextArea, JPassword, JTextPane | txt | Các loại ô nhập |
| JRadioButton | rdo | Hộp chọn |
| JCheckBox | chk | Hộp kiểm |
| JMenu | mnu | Thực đơn |
| JMenuItem | mni | Mục menu |
| JPanel | pnl | Khung chứa |
| JTable | tbl |  |
| … |  |  |

#### Chào



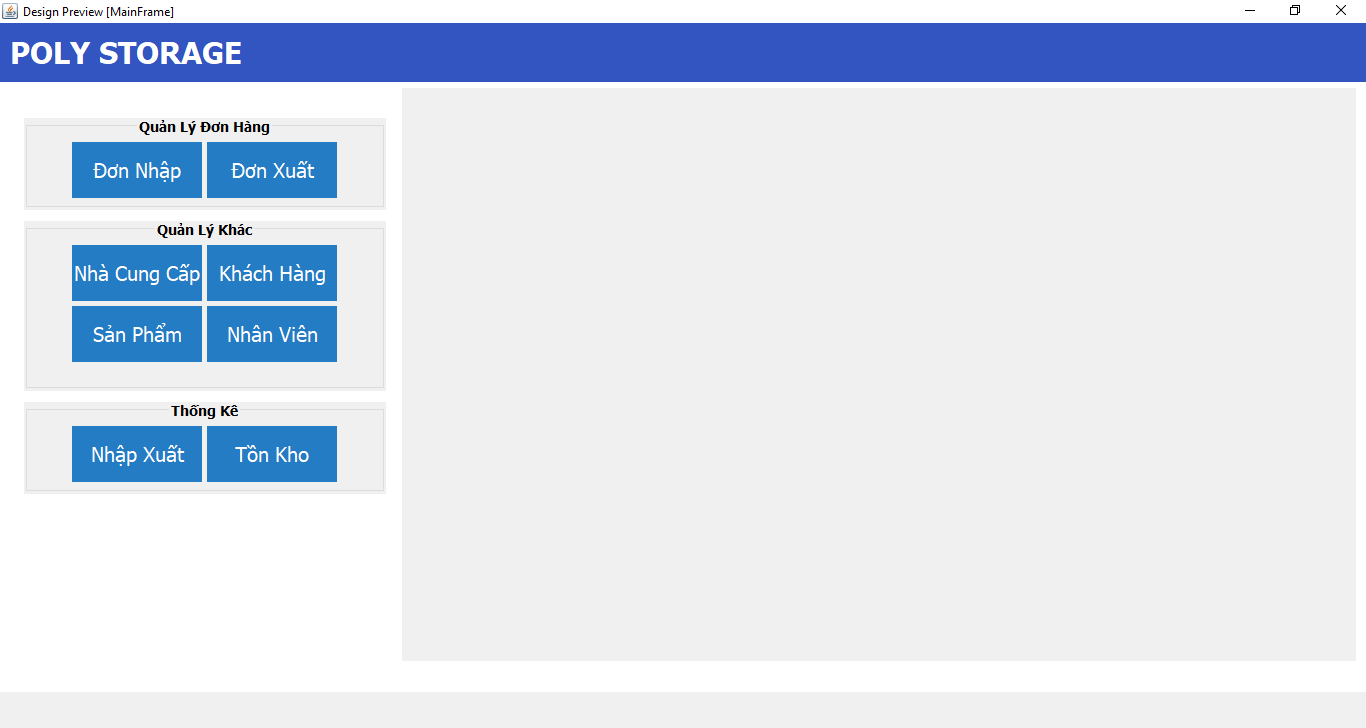
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Componel** | **Kiểu** | **Thuộc tính** |
| 1 | JDialogChao | JDialog | defaultCloseOperation: EXIT\_ON\_CLOSE |
| 2 | Pnlall | JPanel | Layout:Absolutelayout  Background: [0,153,255] |
| 3 | lblphantram | JLabel |  |
| 4 | Lbltitle | JLabel | Text: Polystorage |
| 5 | Lblfb | JLabel | Icon:facebook-sketched-logo-variant.png |
| 6 | Lblin | JLabel | Icon:instagram-logo.png |
| 7 | Lblyt | JLabel | Icon:youtube.png |
| 8 | Lblemail | JLabel | Icon:email\_1.png |
| 9 | Lblem | JLabel | Text: svpoly.hcm@fpt.edu.vn |

#### Đăng nhập



|  |  |  |  |
| --- | --- | --- | --- |
|  | **Componel** | **Kiểu** | **Thuộc tính** |
| 1 | JDialoglogin | JDialog | defaultCloseOperation: EXIT\_ON\_CLOSE |
| 2 | lblTitle | JLabel | Text:login Now  Font:Tahoma, Bold, 36  Foreground: [255,255,255] |
| 3 | lblImage | JLabel | Icon: box\_1.png |
| 4 | lblMaNV | JLabel | Icon: people.png |
| 5 | txtMaNV | JTextField |  |
| 6 | lblMatKhau | JLabel | Icon: key.png |
| 7 | txtMatKhau | JPassword |  |
| 8 | btnSignIn | JButton | Text:Sign in |
| 9 | btnExit | JButton | Text:Exit |

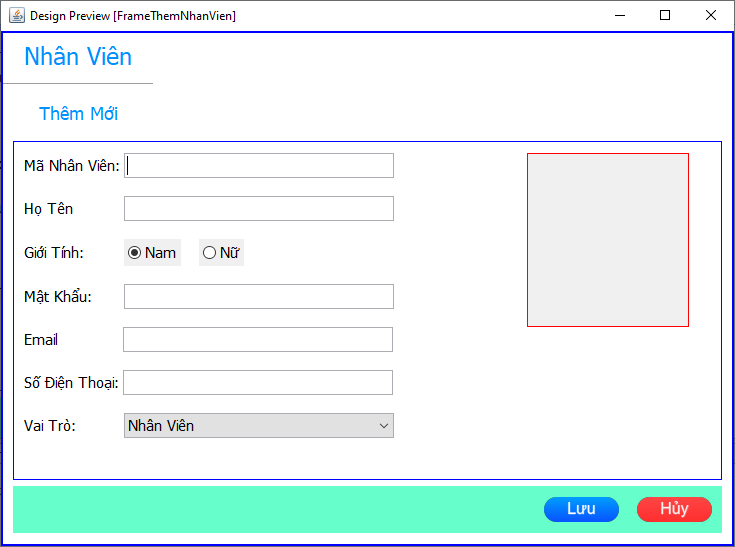
#### Giao diện chính

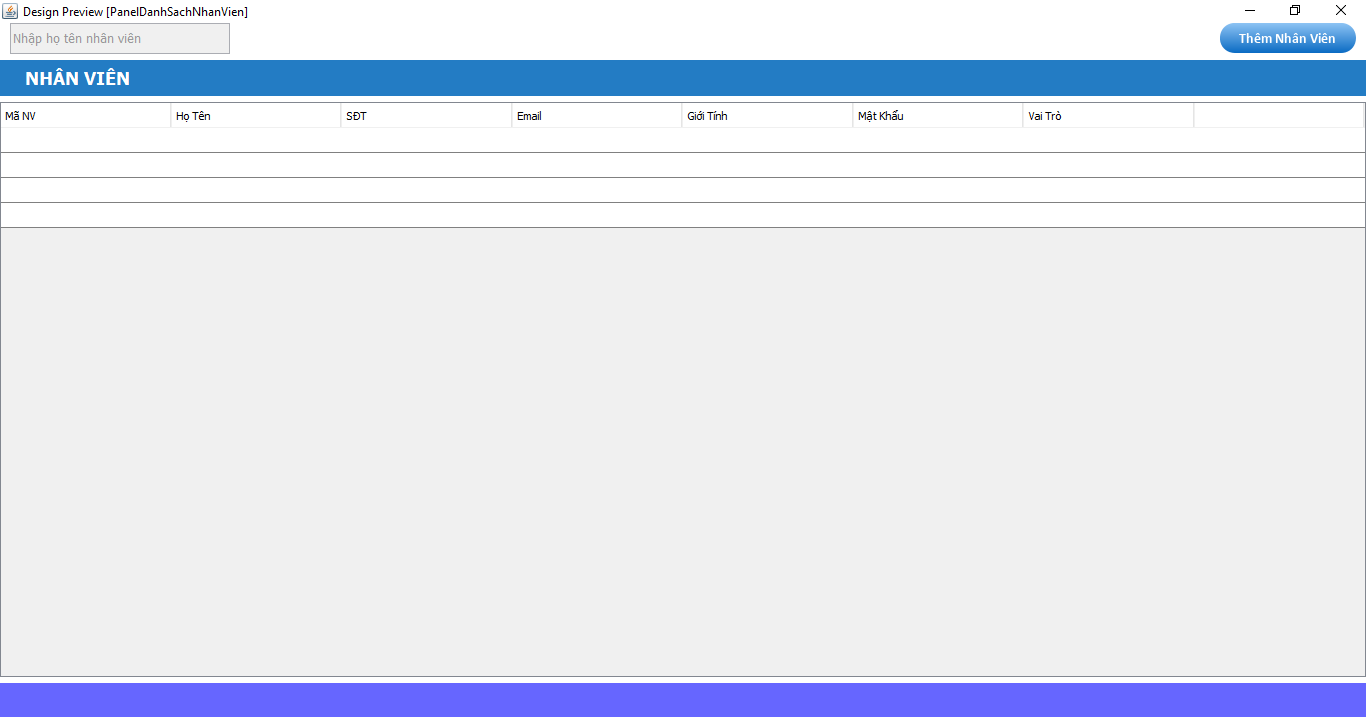


Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Componel** | **Kiểu** | **Thuộc tính** |
| 1 | MainJFrame | JFrame | Layout:BorderLayout |
| 2 | pnlAll | JPanel |  |
| 3 | pnRight | JPanel |  |
| 4 | pnlRoot | JPanel | Layout:GridLayout |
| 5 | pnllogo | JPanel |  |
| 6 | Lbllogo | JLabel | Text: POLY STORAGE  Font:Tahoma 30 Bold  Background: [240,240,240] |
| 7 | pnlQuanLyDonHang | JPanel | Layout:FlowLayout |
| 8 | pnlDonNhap | JPanel | Background: [35,124,196] |
| 9 | lblDonNhap | JLabel | Text: Đơn Nhập  Font:Tahoma 20 Plain |
| 10 | pnlDonXuat | JPanel | Background: [35,124,196] |
| 11 | lblDonXuat | JLabel | Text: Đơn Xuất |
| 12 | pnlQuanLyKhac | JPanel | Layout:FlowLayout |
| 13 | pnlNhaCungCap | JPanel |  |
| 14 | lblNhaCungCap | JLabel | Text: Nhà Cung Cấp |
| 15 | pnlKhachHang | JPanel |  |
| 16 | lblKhachHang | JLabel | Text: Khách Hàng |
| 17 | pnlSanPham | JPanel |  |
| 18 | lblSanPham | JLabel | Text: Sản Phẩm |
| 19 | pnlNhanVien | JPanel |  |
| 20 | lblNhanVien | JLabel | Text: Nhân Viên |
| 21 | pnlThongKe | JPanel | Layout:FlowLayout |
| 20 | pnlnhapxuat | JPanel |  |
| 21 | lblTongSanPhamNhapXuat | JLabel | Text: Nhập Xuất |
| 22 | pnltonkho | JPanel |  |
| 23 | lblTonKho | JLabel | Text: Tồn Kho  Icon: map-and-flags.png |
| 24 | lblBackGround | Label | Icon: warehouses.png |

#### Quản lý nhân viên

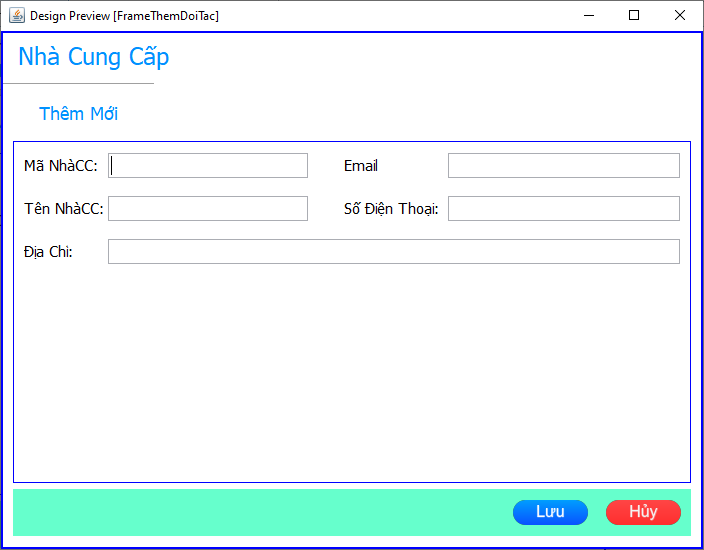


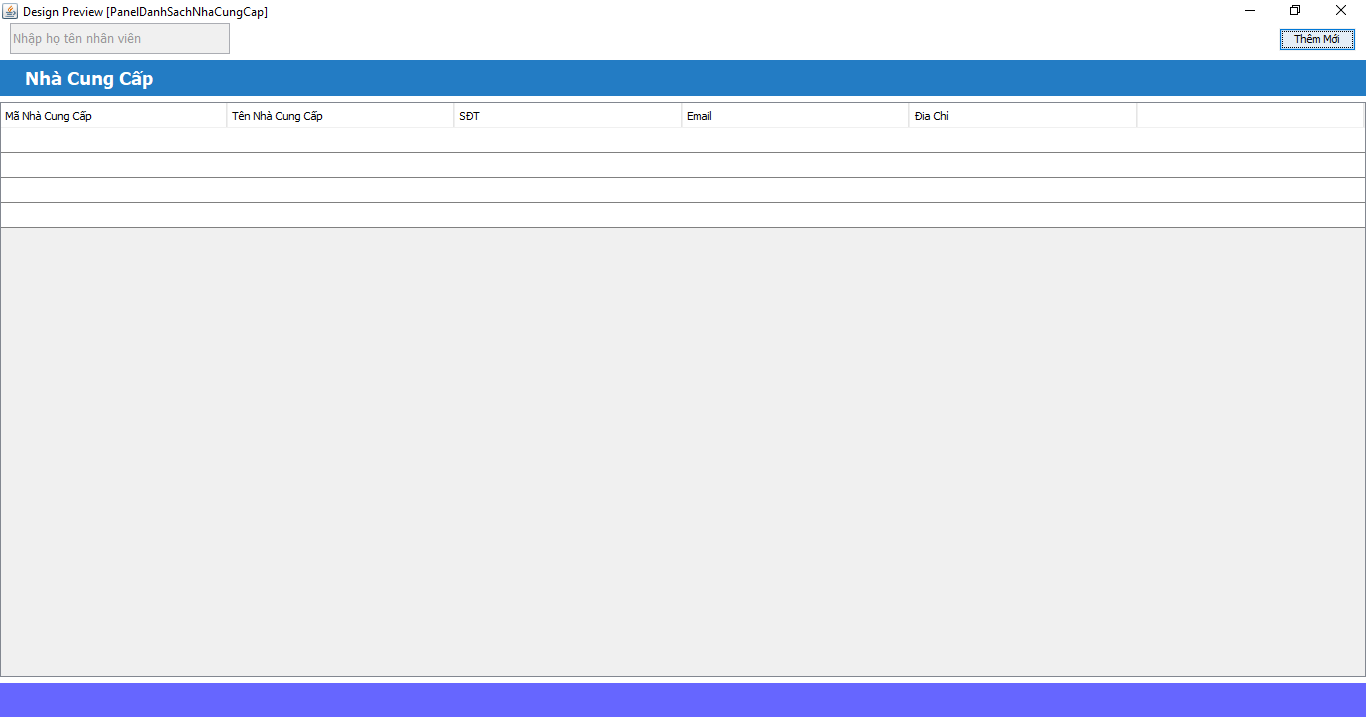


Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | FrameThemNhanVien | JFrame | defaultCloseOperation:DISPONSE |
| 2 | lblTitle | JLabel | Text: Nhân Viên  Foreground: [0,145,255]  Font:Tahoma, Bold, 24 |
| 3 | tabNhanVien | JTabbedPane |  |
| 4 | pnlThemMoi | JPanel | Tab Title: Thêm Mới |
| 5 | lblMaNV | JLabel | Text: Mã nhân viên |
| 6 | txtMaNV | JTextField |  |
| 7 | lblMatKhau | JLabel | Text: Mật khẩu |
| 8 | txtMatKhau | JPassword |  |
| 9 | lblHoTen | JLabel | Text: Họ và tên |
| 10 | txtHoTen | JTextField |  |
| 11 | lblEmail | JLabel | Text: Email |
| 12 | txtEmail | JTextField |  |
| 13 | lblSDT | JLabel | Text: Số điện thoại |
| 14 | lblVaiTro | JLabel | Text:Vai Trò |
| 15 | txtSDT | JTextField |  |
| 16 | lblVaiTro | JLabel | Text:Vai Trò |
| 17 | cboVaiTro | JComboBox | Model: Nhân Viên, Trưởng Phòng  Editable: false |
| 18 | lblGioiTinh | JLabel | Text:Giới Tính |
| 19 | cboGioiTinh | JComboBox | Model: Nam, Nữ  Editable: false |
| 20 | lblHinh | JLabel | Icon: unknow.png |
| 21 | btnInsert | JButton | Text:Lưu |
| 22 | btnUpdate | JButton | Text:Hủy |
|  |  |  |  |
| 1 | tabNhanVien | JTabbedPane |  |
| 2 | lblTitle | JLabel | Text: Nhân Viên  Foreground: [0,145,255]  Font:Tahoma, Bold, 24 |
| 3 | txtTimKiem | Text Field | Text: Nhập họ tên nhân viên |
| 4 | btnThemNV | JButton | Icon: button\_them-nhan-vien.png |
| 5 | tblQLNV | JTable | Model: như hình |
| 2 | lblTitle | JLabel | Text: Nhân Viên  Foreground: [0,145,255]  Font:Tahoma, Bold, 24 |
| 3 | txtTimKiem | Text Field | Text: Nhập họ tên nhân viên |
| 4 | btnThemNV | JButton | Icon: button\_them-nhan-vien.png |
| 5 | tblQLNV | JTable | Model: như hình |

#### Quản lý nhà cung cấp

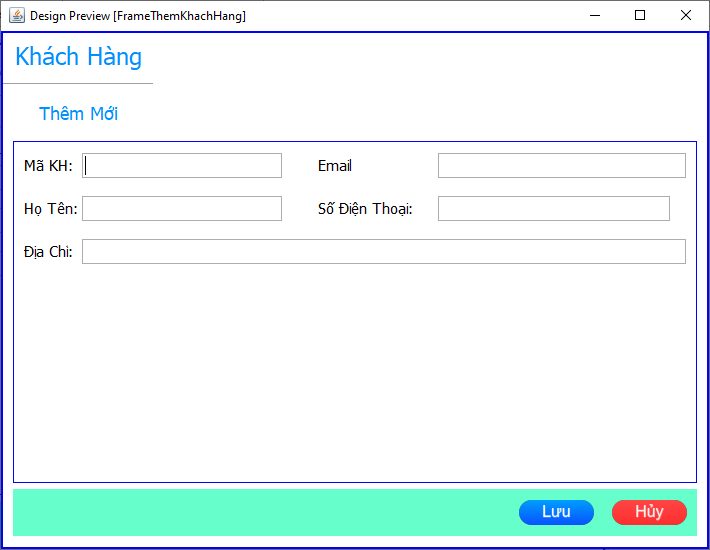
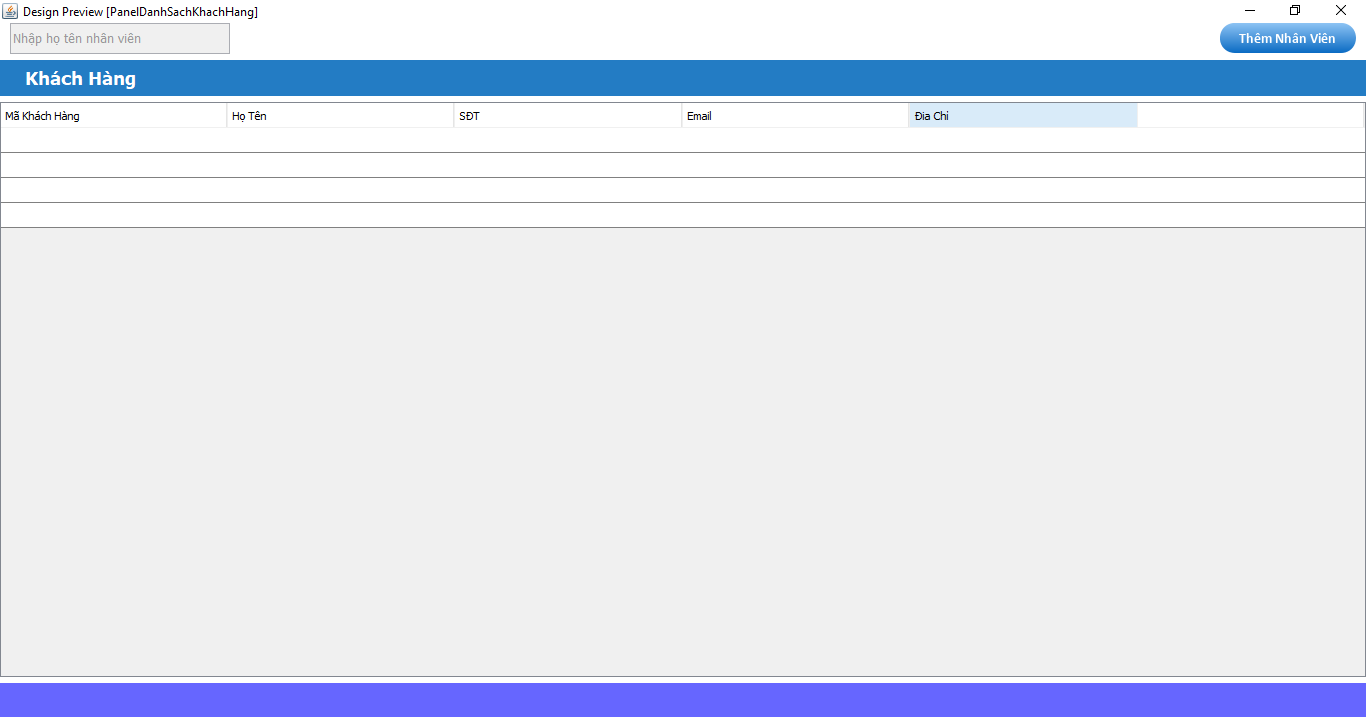




Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | FrameThemDoiTac | JFrame | defaultCloseOperation:DISPONSE |
| 2 | lblTitle | JLabel | Text: Nhà cung Cấp  Foreground:[0,145,255]  Font:Tahoma 24 Plain |
| 3 | lblChucNang | JLabel | Text:Thêm Mới  Foreground:[0,145,255]  Font:Tahoma 18 Plain |
| 4 | pnlCenter | JPanel |  |
| 5 | jLabel3 | JLabel | Text: Mã NhàCC |
| 6 | txtMaDT | JTextField |  |
| 7 | jLabel4 | JLabel | Text: Tên NhàCC: |
| 8 | txtTenDT | JTextField |  |
| 9 | lblDiaChi | JLabel | Text: Địa chỉ |
| 10 | txtDiaChi | JTextField |  |
| 11 | lblEmail | JLabel | Text: Email |
| 12 | txtEmail | JTextField |  |
| 13 | lblSDT | JLabel | Text: Số điện thoại |
| 14 | txtSDT | JTextField |  |
| 15 | pnlControl | JPanel | Background: [102,255,204] |
| 16 | lblSave | JLabel | Icon: button\_luu.png |
|  |  |  |  |
| 1 | PanelDanhSachNhaCungCap | JPanel |  |
| 2 | pnlControl | JPanel | Text:Thêm Mới  Foreground:[0,145,255]  Font:Tahoma 18 Plain |
| 3 | txtTimKiem | JTextField | Text: Nhập họ tên nhân viên |
| 4 | pnlTitle | JPanel | background:[35,124,196] |
| 5 | lblTitle | JLabel | Font:Tahoma 18 Bold  Foreground:[255,255,255] |
| 6 | btnThemMoi | JButtom | Text: Thêm Mới |
| 7 | jScrollPane1 | jScrollPane |  |
| 8 | tblNhaCungCap | JTable |  |
| 9 | jPanel9 | JPanel | Background:[102,102,255] |

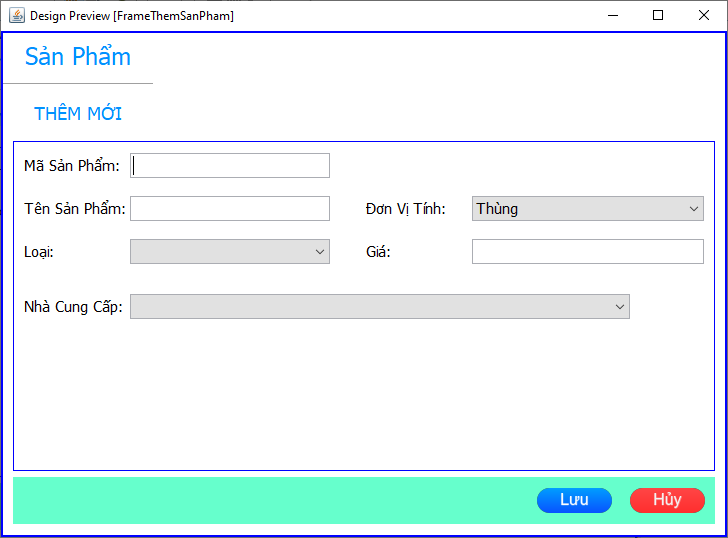
#### Quản lý khách hàng

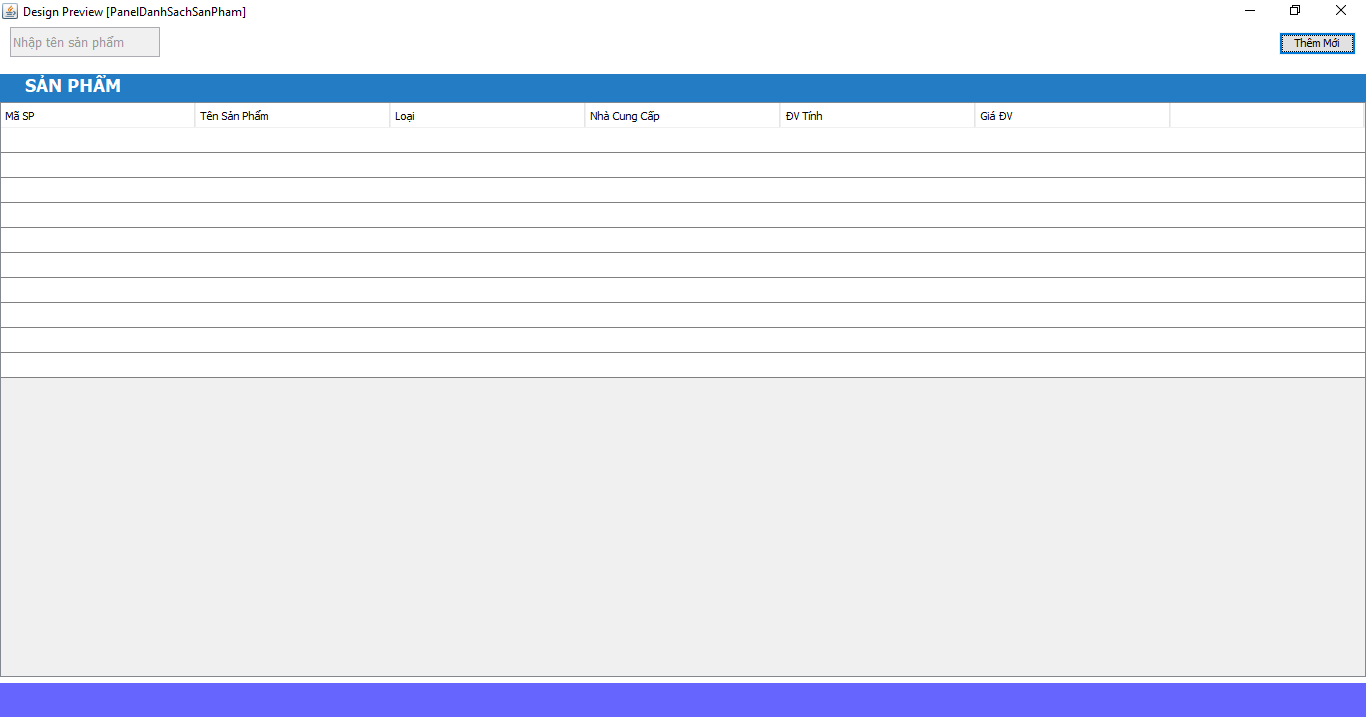
 

Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | FrameThemKhachHang | JFrame |  |
| 2 | jPanel1 | JPanel |  |
| 3 | jLabel1 | JLabel | Text: Khách Hàng  Foreground:[0,145,255]  Font:Tahoma 24 Plain |
| 4 | lblChucNang | JLabel | Text:Thêm Mới  Foreground:[0,145,255]  Font:Tahoma 24 Plain |
| 5 | pnlCenter | JPanel |  |
| 6 | jLabel3 | JLabel | Text: Mã KH |
| 7 | jLabel4 | JLabel | Text: Họ Tên |
| 8 | txtMaDT | JTextField |  |
| 9 | txtHoTen | JTextField |  |
| 10 | jLabel7 | JLabel | Text: Email |
| 11 | jLabel8 | JLabel | Text:Số Điện Thoại |
| 12 | txtSDT | JTextField |  |
| 13 | txtEmail | JTextField |  |
| 14 | jLabel9 | JLabel | Text: Địa Chỉ |
| 15 | txtDiaChi | JTextField |  |
| 16 | pnlControl | JPanel | Background:[102,255,204] |
| 17 | lblSave | JLabel | Icon:button\_luu.png |
| 18 | lblCancel | JLabel | Icon:button\_huy(1).png |
|  |  |  |  |
| 1 | PanelDanhSachKhachHang | JPanel | defaultCloseOperation:DISPONSE |
| 2 | pnlControl | JPanel |  |
| 3 | txtTimKiem | JTextField |  |
| 4 | btnThemNhanVien | JLabel | Icon:button\_them-nhan-vien.png |
| 5 | pnlTitle | JPanel | Background:[35,124,196] |
| 6 | lblTitle | JLabel | Text: Khách Hàng  Font:Tahoma 18 Bold  Foreground:[255,255,255] |
| 7 | jScrollPane1 | JScrollPane |  |
| 8 | tblDoiTac | JTable |  |
| 9 | jPanel9 | jPanel9 | Background:[102,102,255] |

#### Quản lý sản phẩm

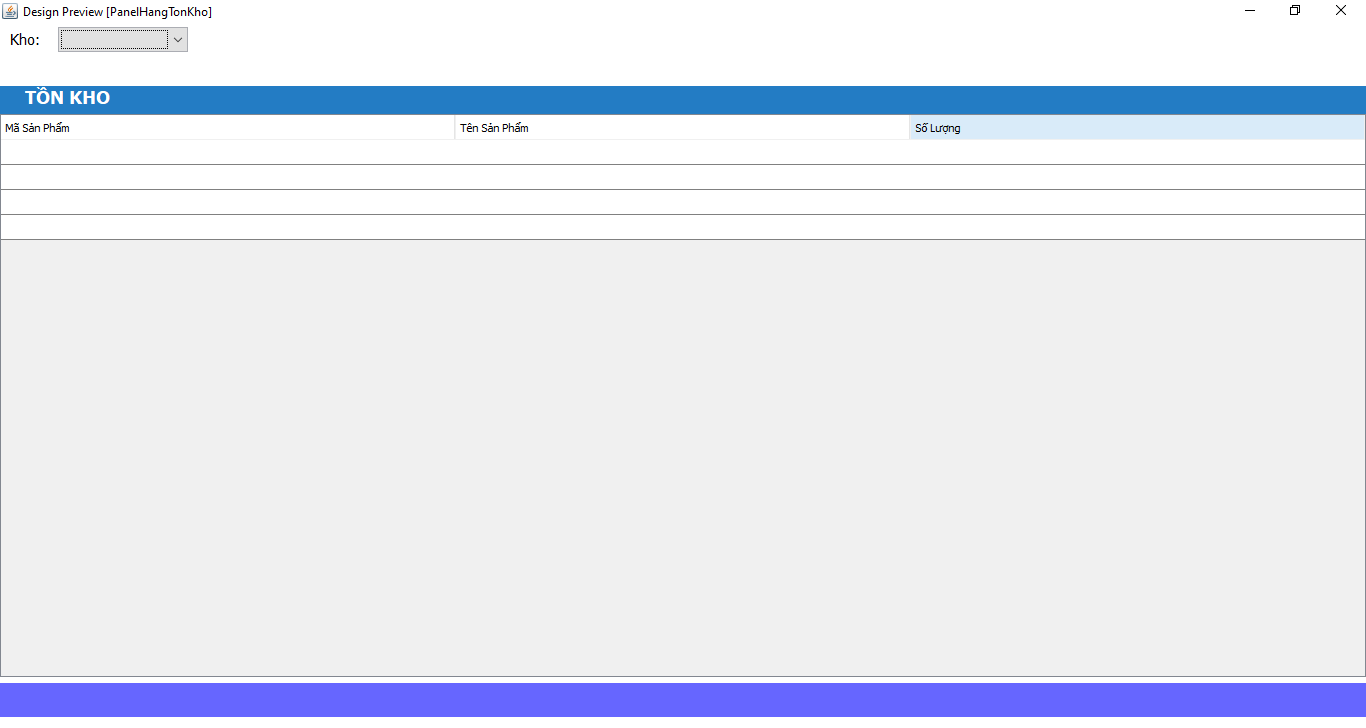




Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | FrameThemSanPham | JFrame | defaultCloseOperation:DISPONSE |
| 2 | lblTitle | JLabel | Text: sản phẩm  Foreground:[0,145,255]  Font:Tahoma 24 Plain |
| 3 | lblChucNang | JLabel | Text:THÊM MỚI  font:Tahoma 18 Plain  foreground:[0,145,255] |
| 4 | pnlCenter | JPanel |  |
| 5 | lblMaSP | JLabel | Text: Mã sản phẩm |
| 6 | txtMaSP | JTextField |  |
| 7 | lblTenSP | JLabel | Text: Tên sản phẩm |
| 8 | TxtTenSP | JTextField |  |
| 9 | lblNhaCC | JLabel | Text:Nhà cung cấp |
| 10 | cboNhaCC | JComboBox | Model: |
| 11 | lblgia | JLabel | Text: Giá |
| 12 | txtGia | JTextField |  |
| 13 | lblGiaDV | JLabel | Text: ĐV Tính |
| 14 | cboGiaDV | JComboBox | Model:Thùng,chai |
| 15 | lblSave | JLabel | Icon:button\_luu.png |
| 16 | lblCancel | JLabel | Icon:button\_huy(1).png |
| 17 | pnlControl | JPanel | Background:[102,255,204] |
|  |  |  |  |
| 1 | PanelDanhSachSanPham | JPanel |  |
| 2 | pnlControl | JPanel | Text:Sản Phẩm  Foreground:[0,145,255]  Font:Tahoma 18 Plain |
| 3 | txtTimKiem | JTextField | Text: Nhập họ tên sản phẩm |
| 4 | pnlTitle | JPanel | background:[35,124,196] |
| 5 | lblTitle | JLabel | Texr: SẢN PHẨM  Font:Tahoma 18 Bold  Foreground:[255,255,255] |
| 6 | btnThemMoi | JButtom | Text: Thêm Mới |
| 7 | jScrollPane1 | jScrollPane |  |
| 8 | tblSanPham | JTable |  |
| 9 | jPanel9 | JPanel | Background:[102,102,255] |
| 1 | PanelDanhSachNhaCungCap | JPanel |  |

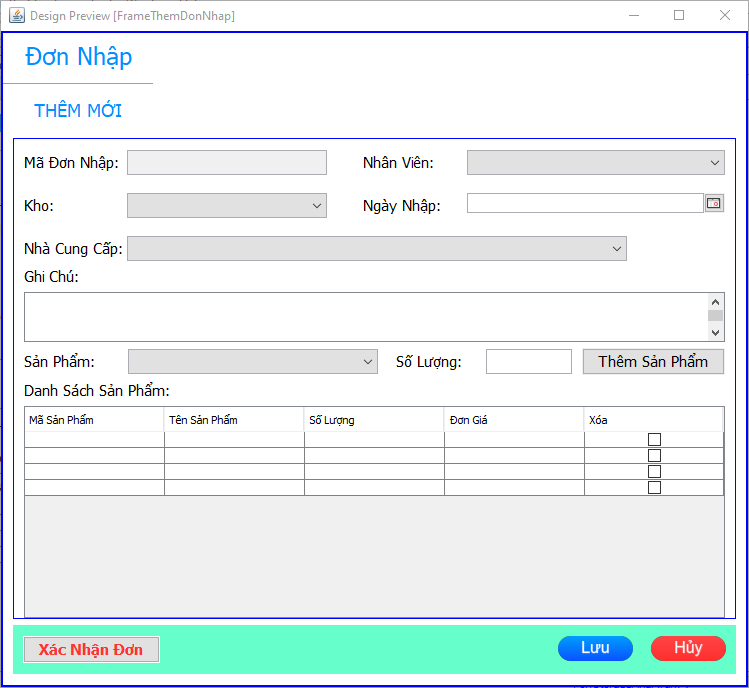
#### Quản lý kho

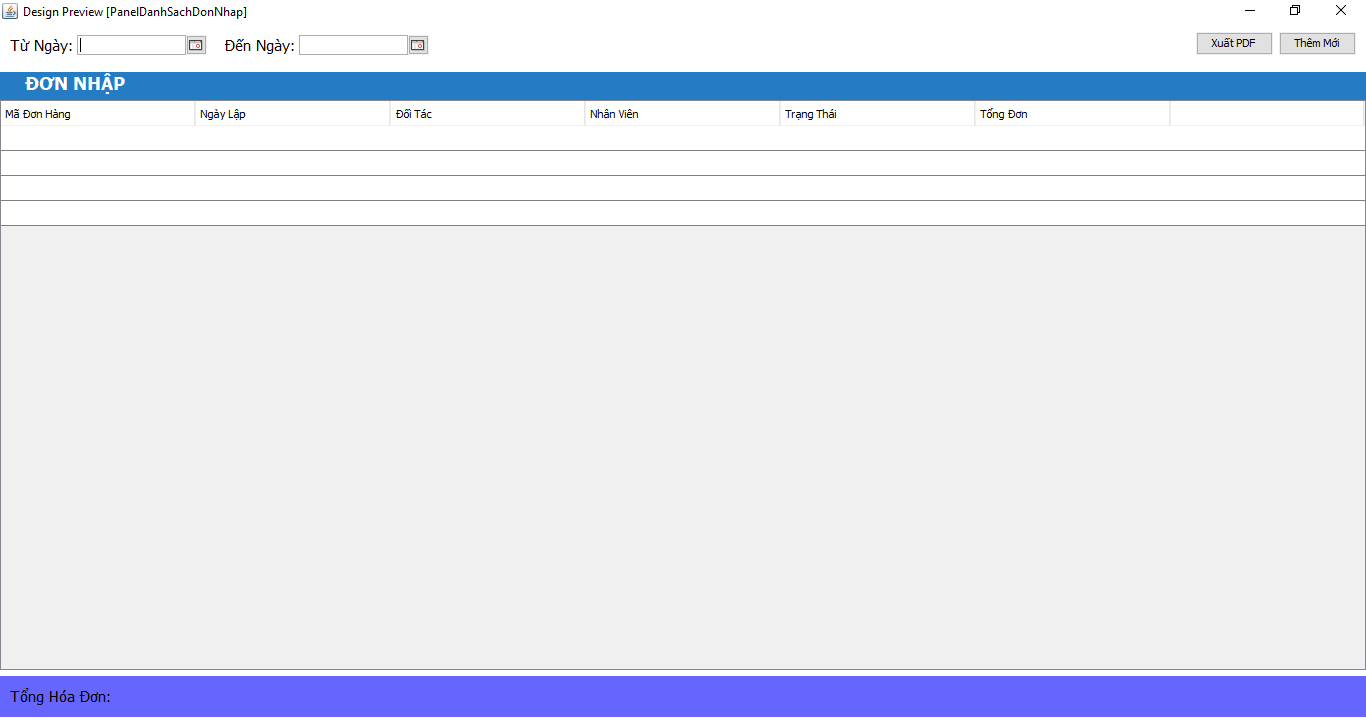


Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | PanelHangTonKho | JPanel |  |
| 3 | pnlControl | JPanel | Foreground:[0,145,255]  Font:Tahoma 18 Plain |
| 4 | jLabel1 | JLabel | Text: Kho |
| 5 | cboMaKho | JComboBox |  |
| 6 | pnlRoot | JPanel |  |
| 7 | lblTitle | JLabel | Text: TỒN KHO  Font:Tahoma 18 Bold |
| 8 | jScrollPane1 | jScrollPane |  |
| 15 | tblTonKho | JTable |  |
| 16 | jPanel2 | JPannel | Background:[102,102,255] |

#### Quản lý đơn nhập

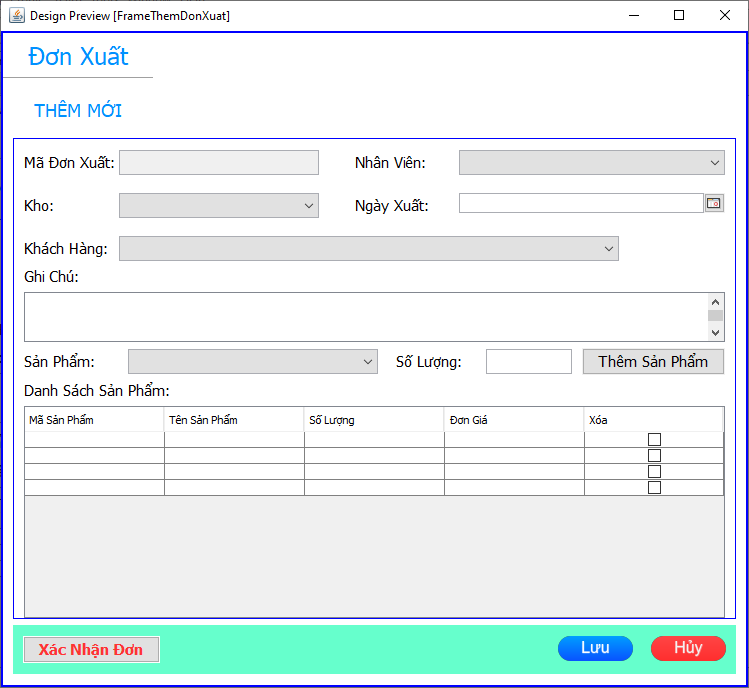


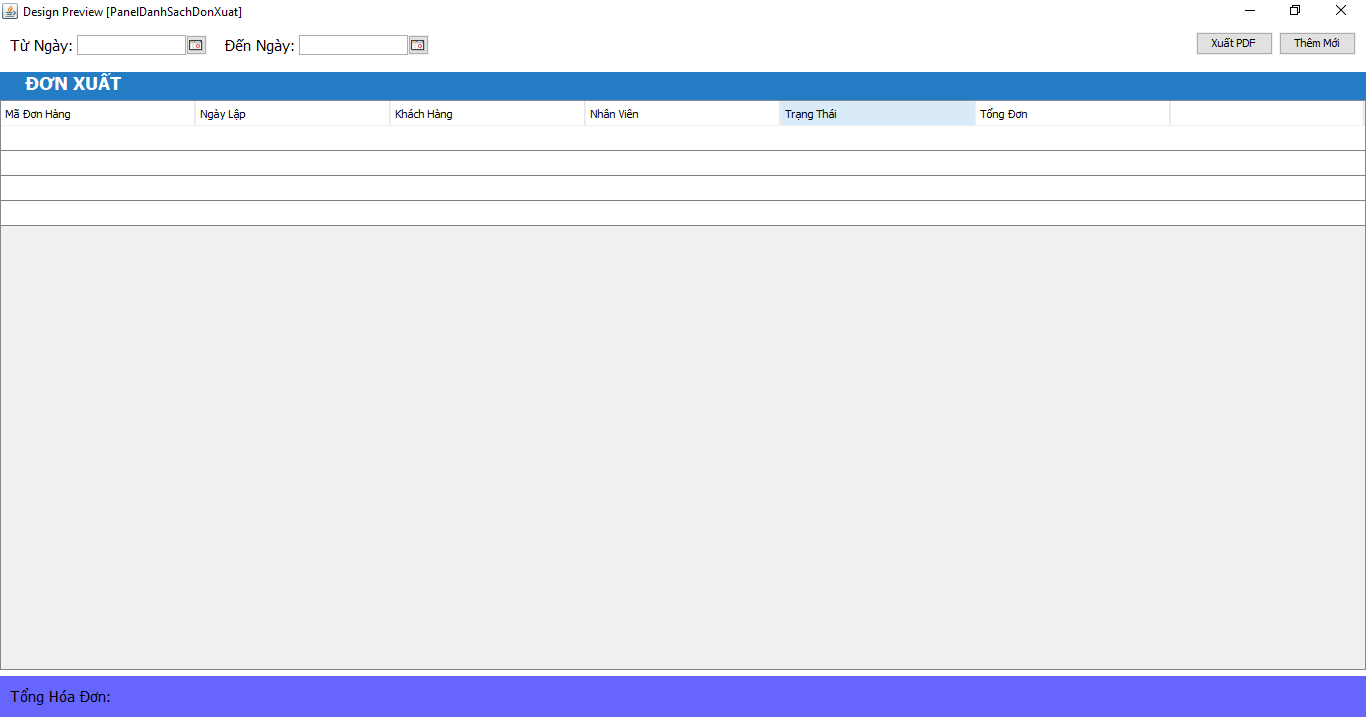


Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | FrameThemDonNhap | JFrame |  |
| 2 | Pnlall | JPanel |  |
| 3 | jLabel1 | JLabel | Tab Title: Đơn Nhập |
| 4 | lblChucNang | JLabel | Text: THÊM MỚI |
| 5 | jLabel3 | JLabel | Text: Mã Đơn Nhập |
| 6 | jLabel4 | JLabel | Text: Nhân Viên |
| 7 | txtMaDN | JTextField |  |
| 8 | cboKho | JComboBox |  |
| 9 | jLabel6 | JLabel | Text: Nhà Cung Cấp |
| 10 | jLabel7 | JLabel | Text: Ghi Chú |
| 11 | jLabel8 | JLabel | Text: Ngày Nhập |
| 12 | txtNgayNhap | JDateChooser |  |
| 13 | lblGhiChu | JLabel | Text:Ghi Chú |
| 14 | cboNhaCungCap | JComboBox |  |
| 15 | cboNhanVien | JComboBox |  |
| 16 | jScrollPane1 | JScrollPane |  |
| 17 | txtGhiChu | JTextArea | Text:Xóa  Icon: delete.png |
| 18 | jLabel9 | JLabel | Text: Sản Phẩm |
| 19 | cboSanPham | JComboBox |  |
| 20 | jLabel10 | JLabel | Text: Số Lượng |
| 21 | btnThemSanPham | JButton | Text:Thêm Sản Phẩm |
| 22 | jLabel11 | JLabel | Text: Danh Sách Sản Phẩm |
| 23 | jScrollPane3 | JSrollPane |  |
| 24 | tblChiTietDonNhap | JTable |  |
| 25 | pnlControl | JPanel |  |
| 26 | lblSave | JLabel | Icon:button\_luu.png |
| 27 | lblCanel | JLabel | Icon:button\_huy(1).png |
| 28 | btnXacNhanDon | JButton | Text:Xác Nhận Đơn  Font:Tahoma 15 Bold  Foreground:[255,51,51] |
|  |  |  |  |
| 1 | PanelDanhSachDonNhap | JPanel |  |
| 2 | pnlControl | JPanel |  |
| 3 | btnThemMoi | JButton | Text: Thêm Mới |
| 4 | btnXuatPDF | JButton | Text: Xuất PDF |
| 5 | txtNgayMin | JDateChooser |  |
| 6 | txtNgayMax | JDateChooser |  |
| 7 | jLabel1 | JLabel | Text: Từ Ngày |
| 8 | jLabel2 | JLabel | Text: Đến Ngày |
| 9 | pnlRoot | JPanel |  |
| 10 | lblTitle | JLabel | Text: ĐƠN NHẬP  Font:Tahoma 18 Bold  Foreground:[255,255,255] |
| 11 | jScrollPane1 | JScrollPane |  |
| 12 | tblDonNhap | JTable |  |
| 14 | jPanel2 | JPanel | Background:[102,102,255] |
| 15 | jLabel3 | JLabel | Text:Tổng Hóa Đơn  Font:Tahoma 15 Plain |
| 16 | lblTongHoaDon | JLabel | Text:  Font:Tahoma 15 Plain |

#### Quản lý đơn xuất





|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | FrameThemDonXuat | JFrame |  |
| 2 | jLabel1 | Jlabel | Text: Đơn Xuất  Font:Tahoma 24 Plain  Foreground:[0,145,255] |
| 3 | lblChucNang | Jlabel | Text: THÊM MỚI  Font:Tahoma 24 Plain  Foreground:[0,145,255] |
| 4 | pnlCenter | JPanel |  |
| 5 | jLabel3 | JLabel | Text:Mã Đơn Xuất |
| 6 | jLabel4 | JLabel | Text:Nhân Viên |
| 7 | txtMaDX | JTextField |  |
| 8 | jLabel5 | JLabel | Text: Kho |
| 9 | cboKho | JComboBox |  |
| 10 | jLabel6 | JLabel | Text:Khách Hàng |
| 11 | jLabel7 | JLabel | Text:Ghi Chú |
| 12 | jLabel8 | JLabel | Text:Ngày Xuất |
| 13 | cboKhachHang | JComboBox |  |
| 14 | cboNhanVien | JComboBox |  |
| 15 | txtNgayXuat | JDateChooser |  |
| 16 | jScrollPane1 | JScrollPane |  |
| 17 | txtGhiChu | JTextArea |  |
| 18 | jLabel9 | JLabel | Text: Sản Phẩm |
| 19 | cboSanPham | JComboBox |  |
| 20 | jLabel10 | JLabel | Text:Số Lượng |
| 21 | txtSoLuong | JTextField |  |
| 22 | btnThemSanPham | JButton | Text: Thêm Sản Phẩm |
| 23 | jLabel11 | JLabel | Text:Danh Sách Sản Phẩm |
| 24 | jScrollPane3 | JScrollPane |  |
| 25 | tblChiTietDonNhap | JTable |  |
| 26 | pnlControl | JPanel |  |
| 27 | lblSave | JLabel | Icon:button\_luu.png |
| 28 | lblCancel | JLabel | Icon:button\_huy(1).png |
| 29 | btnXacNhanDon | JButton | Text:Xác Nhận Đơn  Foreground:[255,51,51] |
|  |  |  |  |
| 1 | PanelDanhSachDonXuat | JPanel |  |
| 2 | pnlControl | JPanel |  |
| 3 | btnThemMoi | JButtont | Text: Thêm Mới |
| 4 | jLabel1 | JLabel | Text:Từ Ngày |
| 5 | txtNgayMin | JDateChooser |  |
| 6 | jLabel2 | JLabel | Text: Đến Ngày |
| 7 | txtNgayMax | JDateChooser |  |
| 8 | btnXuatPDF | JButtont | Text: Xuất PDF |
| 9 | pnlRoot | JPanel |  |
| 10 | lblTitle |  | Text:ĐƠN XUẤT  Font:Tahoma 18 Bold  Foreground:[255,255,255] |
| 11 | jScrollPane1 | JScrollPane |  |
| 12 | tblDonXuat | JTable |  |
| 13 | jPanel2 | JPanel |  |
| 14 | jLabel3 | JLabel | Text: Tổng Hóa Đơn |
| 15 | lblTongHoaDon | JLabel | Text: |

# Thực hiện viết mã

## Viết mã tạo CSDL

### Tạo CSDL

CREATE DATABASE polystorage

USE polystorage

CREATE TABLE `chitietdonnhap` (

`MaDN` int(11) NOT NULL,

`MaSP` char(10) NOT NULL,

`SoLuong` int(11) NOT NULL,

PRIMARY KEY (`MaDN`,`MaSP`),

KEY `FK\_ChiTietDonNhap\_SanPham` (`MaSP`),

CONSTRAINT `FK\_ChiTietDonNhap\_DonNhap` FOREIGN KEY (`MaDN`) REFERENCES `donnhap` (`MaDN`) ON UPDATE CASCADE,

CONSTRAINT `FK\_ChiTietDonNhap\_SanPham` FOREIGN KEY (`MaSP`) REFERENCES `sanpham` (`MaSP`) ON UPDATE CASCADE))

CREATE TABLE `chitietkho` (

`MaKho` char(10) NOT NULL,

`MaSP` char(10) NOT NULL,

`SoLuong` int(11) NOT NULL,

PRIMARY KEY (`MaKho`,`MaSP`),

KEY `FK\_ChiTietKho\_SanPham` (`MaSP`),

CONSTRAINT `FK\_ChiTietKho\_Kho` FOREIGN KEY (`MaKho`) REFERENCES `kho` (`MaKho`) ON UPDATE CASCADE,

CONSTRAINT `FK\_ChiTietKho\_SanPham` FOREIGN KEY (`MaSP`) REFERENCES `sanpham` (`MaSP`) ON UPDATE CASCADE

)

CREATE TABLE `donnhap` (

`MaDN` int(11) NOT NULL AUTO\_INCREMENT,

`MaNV` char(10) NOT NULL,

`NgayNhap` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`TrangThai` bit(1) NOT NULL DEFAULT b'0',

`MaKho` char(10) NOT NULL,

`MaNCC` char(10) NOT NULL,

`GhiChu` varchar(255) DEFAULT NULL,

PRIMARY KEY (`MaDN`),

KEY `FK\_DonNhap\_Kho` (`MaKho`),

KEY `FK\_DonNhap\_NhanVien` (`MaNV`),

KEY `FK\_DonNhap\_NhaCungCap` (`MaNCC`),

CONSTRAINT `FK\_DonNhap\_Kho` FOREIGN KEY (`MaKho`) REFERENCES `kho` (`MaKho`) ON UPDATE CASCADE,

CONSTRAINT `FK\_DonNhap\_NhaCungCap` FOREIGN KEY (`MaNCC`) REFERENCES `nhacungcap` (`MaNCC`) ON UPDATE CASCADE,

CONSTRAINT `FK\_DonNhap\_NhanVien` FOREIGN KEY (`MaNV`) REFERENCES `nhanvien` (`MaNV`) ON UPDATE CASCADE

)

CREATE TABLE `donxuat` (

`MaDX` int(11) NOT NULL AUTO\_INCREMENT,

`MaNV` char(10) NOT NULL,

`NgayXuat` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`TrangThai` bit(1) NOT NULL DEFAULT b'0',

`MaKho` char(10) NOT NULL,

`MaKH` char(10) NOT NULL,

PRIMARY KEY (`MaDX`),

KEY `FK\_DonXuat\_Kho` (`MaKho`),

KEY `FK\_DonXuat\_NhanVien` (`MaNV`),

KEY `FK\_DonXuat\_KhachHang` (`MaKH`),

CONSTRAINT `FK\_DonXuat\_KhachHang` FOREIGN KEY (`MaKH`) REFERENCES `khachhang` (`MaKH`) ON UPDATE CASCADE,

CONSTRAINT `FK\_DonXuat\_Kho` FOREIGN KEY (`MaKho`) REFERENCES `kho` (`MaKho`) ON UPDATE CASCADE,

CONSTRAINT `FK\_DonXuat\_NhanVien` FOREIGN KEY (`MaNV`) REFERENCES `nhanvien` (`MaNV`) ON UPDATE CASCADE

)

CREATE TABLE `khachhang` (

`MaKH` char(10) NOT NULL,

`HoTen` varchar(50) NOT NULL,

`Email` varchar(50) NOT NULL,

`SDT` char(10) NOT NULL,

`DiaChi` varchar(255) NOT NULL,

`GhiChu` varchar(255) DEFAULT NULL,

PRIMARY KEY (`MaKH`)

)

CREATE TABLE `kho` (

`MaKho` char(10) NOT NULL,

`DiaChi` varchar(255) NOT NULL,

PRIMARY KEY (`MaKho`)

)

CREATE TABLE `loaisanpham` (

`MaLoai` char(10) NOT NULL,

`TenLoai` varchar(50) NOT NULL,

PRIMARY KEY (`MaLoai`)

)

CREATE TABLE `nhacungcap` (

`MaNCC` char(10) NOT NULL,

‘TenNCC` varchar(50) NOT NULL,

`Email` varchar(50) NOT NULL,

`SDT` char(10) NOT NULL,

`DiaChi` varchar(255) NOT NULL,

`GhiChu` varchar(255) DEFAULT NULL,

PRIMARY KEY (`MaNCC`)

)

CREATE TABLE `nhanvien` (

`MaNV` char(10) NOT NULL,

`HoTen` varchar(50) NOT NULL,

`Email` varchar(50) NOT NULL,

`GioiTinh` bit(1) NOT NULL DEFAULT b'1',

`SDT` char(10) NOT NULL,

`MatKhau` varchar(50) NOT NULL,

`Hinh` varchar(50) NOT NULL DEFAULT 'unknow.png',

`VaiTro` varchar(50) NOT NULL DEFAULT 'Nhân Viên',

PRIMARY KEY (`MaNV`)

)

CREATE TABLE `sanpham` (

`MaSP` char(10) NOT NULL,

`TenSP` varchar(50) NOT NULL,

`MaNCC` char(10) NOT NULL,

`MaLoai` char(10) NOT NULL,

`DonViTinh` varchar(50) NOT NULL,

`GiaDV` float NOT NULL,

PRIMARY KEY (`MaSP`),

KEY `FK\_SanPham\_NhaCungCap` (`MaNCC`),

KEY `FK\_SanPham\_Loai` (`MaLoai`),

CONSTRAINT `FK\_SanPham\_Loai` FOREIGN KEY (`MaLoai`) REFERENCES `loaisanpham` (`MaLoai`) ON UPDATE CASCADE,

CONSTRAINT `FK\_SanPham\_NhaCungCap` FOREIGN KEY (`MaNCC`) REFERENCES `nhacungcap` (`MaNCC`) ON UPDATE CASCADE

)

### SQL truy vấn và thao tác

Tất cả các bảng đều có CRUD như:

#### SQL đối với Bảng chitietdonnhap

|  |  |
| --- | --- |
| Hành động | Câu lệnh SQL |
| Thêm mới | INSERT INTO chitietdonnhap(MaDN,MaSP,SoLuong) VALUES(?,?,?,?) |
| Cập nhật theo mã | UPDATE chitetdonnhap SET MaDN=?, MaSP=?, SoLuong=? |
| Xoá theo mã | DELETE FROM chitetdonnhap WHERE MaDN=? |
| Truy vấn tất cả | SELECT \* FROM chitietdonnhap |
| Truy vấn theo mã | SELECT \* FROM chitietdonnhap WHERE MaDN |

#### SQL đối với Bảng donnhap

|  |  |
| --- | --- |
| Hành động | Câu lệnh SQL |
| Thêm mới | INSERT INTO chitietdonhang(MaDX,MaSP,SoLuong) VALUES(?,?,?,?) |
| Cập nhật theo mã | UPDATE chitetdonhang SET MaDX=?, MaSP=?, SoLuong=? |
| Xoá theo mã | DELETE FROM chitetdonhang WHERE MaDX=? |
| Truy vấn tất cả | SELECT \* FROM chitietdonhang |
| Truy vấn theo mã | SELECT \* FROM chitietdonhang WHERE MaDX |

### Các thủ tục lưu tổng hợp

#### Tổng sản phẩm nhập vào

CREATE DEFINER=`root`@`localhost` PROCEDURE `spGetTongSanPhamNhap`()

BEGIN

SELECT sanpham.MaSP, TenSP, sum(SoLuong) TongNhap FROM polystorage.chitietdonnhap, sanpham

WHERE sanpham.MaSP = chitietdonnhap.MaSP

GROUP BY sanpham.MaSP

ORDER BY TongNhap DESC;

END

#### Tổng giá trị đơn nhập

CREATE DEFINER=`root`@`localhost` PROCEDURE `spGetTongDonNhap`(IN \_MaDN int)

BEGIN

SELECT SUM(GiaDV \* SoLuong) TongGia FROM chitietdonnhap,donnhap,sanpham

WHERE chitietdonnhap.MaDN = donnhap.MaDN

AND chitietdonnhap.MaSP = sanpham.MaSP

AND chitietdonnhap.MaDN = \_MaDN

GROUP BY chitietdonnhap.MaDN;

END

## Lập trình hibernate

### Hibernate.config.xml

<hibernate-configuration>

<session-factory>

<property name="hibernate.connection.driver\_class">com.mysql.jdbc.Driver</property>

<property name="hibernate.connection.url">jdbc:mysql://localhost/PolyStorage?useUnicode=true&amp;characterEncoding=utf-8</property>

<property name="hibernate.connection.username">root</property>

<property name="hibernate.connection.password"></property>

<property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

<property name="hibernate.show\_sql">true</property>

<property name="hibernate.current\_session\_context\_class">thread</property>

<property name="hibernate.query.factory\_class">org.hibernate.hql.internal.ast.ASTQueryTranslatorFactory</property>

<mapping resource="com/polystorage/entity/ChiTietDonNhap.hbm.xml"/>

<mapping resource="com/polystorage/entity/ChiTietDonXuat.hbm.xml"/>

<mapping resource="com/polystorage/entity/NhanVien.hbm.xml"/>

<mapping resource="com/polystorage/entity/NhaCungCap.hbm.xml"/>

<mapping resource="com/polystorage/entity/KhachHang.hbm.xml"/>

<mapping resource="com/polystorage/entity/LoaiSanPham.hbm.xml"/>

<mapping resource="com/polystorage/entity/ChiTietKho.hbm.xml"/>

<mapping resource="com/polystorage/entity/SanPham.hbm.xml"/>

<mapping resource="com/polystorage/entity/DonNhap.hbm.xml"/>

<mapping resource="com/polystorage/entity/DonXuat.hbm.xml"/>

<mapping resource="com/polystorage/entity/Kho.hbm.xml"/>

</session-factory>

</hibernate-configuration>

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

#### Sản Phẩm

package com.polystorage.entity;

import java.util.HashSet;

import java.util.Set;

public class SanPham implements java.io.Serializable {

private String maSp;

private LoaiSanPham loaiSanPham;

private NhaCungCap nhaCungCap;

private String tenSp;

private String donViTinh;

private float giaDv;

private Set chiTietDonXuats = new HashSet(0);

private Set chiTietKhos = new HashSet(0);

private Set chiTietDonNhaps = new HashSet(0);

public SanPham() {

}

public SanPham(String maSp, LoaiSanPham loaisanpham, NhaCungCap nhacungcap, String tenSp, String donViTinh, float giaDv) {

this.maSp = maSp;

this.loaiSanPham = loaisanpham;

this.nhaCungCap = nhacungcap;

this.tenSp = tenSp;

this.donViTinh = donViTinh;

this.giaDv = giaDv;

}

public SanPham(String maSp, LoaiSanPham loaisanpham, NhaCungCap nhacungcap, String tenSp, String donViTinh, float giaDv, Set chitietdonxuats, Set chitietkhos, Set chitietdonnhaps) {

this.maSp = maSp;

this.loaiSanPham = loaisanpham;

this.nhaCungCap = nhacungcap;

this.tenSp = tenSp;

this.donViTinh = donViTinh;

this.giaDv = giaDv;

this.chiTietDonXuats = chitietdonxuats;

this.chiTietKhos = chitietkhos;

this.chiTietDonNhaps = chitietdonnhaps;

}

Geter/seter

#### Nhà cung cấp

package com.polystorage.entity;

import java.util.HashSet;

import java.util.Set;

public class NhaCungCap implements java.io.Serializable {

private String maNCC;

private String tenNCC;

private String email;

private String sdt;

private String diaChi;

private String ghiChu;

private Set sanPhams = new HashSet(0);

private Set donNhaps = new HashSet(0);

public NhaCungCap() {

}

public NhaCungCap(String maNcc, String tenNcc, String email, String sdt, String diaChi) {

this.maNCC = maNcc;

this.tenNCC = tenNcc;

this.email = email;

this.sdt = sdt;

this.diaChi = diaChi;

}

public NhaCungCap(String maNcc, String tenNcc, String email, String sdt, String diaChi, String ghiChu, Set sanphams, Set donnhaps) {

this.maNCC = maNcc;

this.tenNCC = tenNcc;

this.email = email;

this.sdt = sdt;

this.diaChi = diaChi;

this.ghiChu = ghiChu;

this.sanPhams = sanphams;

this.donNhaps = donnhaps;

}

Getter/Setter

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

#### DonNhapDAO

package com.polystorage.dao;

import com.polystorage.entity.DonNhap;

import com.polystorage.util.HibernateUtil;

import java.util.Date;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

public class DonNhapDAO {

public List<DonNhap> getListDonNhap(int maDN) {

List<DonNhap> list = null;

Session session = HibernateUtil.getSessionFactory().openSession();

session.beginTransaction();

String sql = "from DonNhap";

if (maDN != -1) {

sql += " and MaDN = '" + maDN + "'";

}

Query query = session.createQuery(sql);

list = query.list();

session.close();

return list;

}

public List<DonNhap> getListDonNhap(Date min, Date max) {

List<DonNhap> list = null;

Session session = HibernateUtil.getSessionFactory().openSession();

session.beginTransaction();

String sql = "FROM DonNhap where NgayNhap >= :min AND NgayNhap <= :max";

Query query = session.createQuery(sql);

query.setParameter("min", min);

query.setParameter("max", max);

list = query.list();

session.close();

return list;

}

public int getNextId() {

Session session = HibernateUtil.getSessionFactory().openSession();

session.beginTransaction();

String sql = "SELECT AUTO\_INCREMENT FROM information\_schema.TABLES"

+ " WHERE TABLE\_SCHEMA = 'polystorage'"

+ " AND TABLE\_NAME = 'donnhap'";

Query query = session.createSQLQuery(sql);

int nextId = Integer.parseInt(query.list().get(0).toString());

session.close();

return nextId;

}

public double getTongDonNhap(int maDN) {

Session session = HibernateUtil.getSessionFactory().openSession();

session.beginTransaction();

String sql = "CALL spGetTongDonNhap(" + maDN + ")";

Query query = session.createSQLQuery(sql);

List<Double> list = query.list();

if (list.size() == 0){

return 0;

}

session.close();

return list.get(0);

}

public DonNhap getInfoDonNhap(int maDN) {

Session session = HibernateUtil.getSessionFactory().openSession();

session.beginTransaction();

DonNhap donHang = (DonNhap) session.get(DonNhap.class, maDN);

session.close();

return donHang;

}

public boolean insertDonNhap(DonNhap dn) {

Session session = HibernateUtil.getSessionFactory().openSession();

try {

session.beginTransaction();

session.save(dn);

session.getTransaction().commit();

return true;

} catch (Exception e) {

session.getTransaction().rollback();

e.printStackTrace();

return false;

} finally {

session.close();

}

}

public boolean updateDonNhap(DonNhap dn) {

if (getInfoDonNhap(dn.getMaDN()) == null) {

return false;

}

Session session = HibernateUtil.getSessionFactory().openSession();

try {

session.beginTransaction();

session.update(dn);

session.getTransaction().commit();

return true;

} catch (Exception e) {

session.getTransaction().rollback();

e.printStackTrace();

return false;

} finally {

session.close();

}

}

public boolean deleteDonNhap(int maDH) {

DonNhap donNhap = getInfoDonNhap(maDH);

Session session = HibernateUtil.getSessionFactory().openSession();

try {

session.beginTransaction();

session.delete(donNhap);

session.getTransaction().commit();

return true;

} catch (Exception e) {

session.getTransaction().rollback();

e.printStackTrace();

return false;

} finally {

session.close();

}

}

}

#### ChiTietDonNhapDAO

package com.polystorage.dao;

import com.polystorage.entity.ChiTietDonNhap;

import com.polystorage.entity.ChiTietDonNhapId;

import com.polystorage.util.HibernateUtil;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

public class ChiTietDonNhapDAO {

public List<ChiTietDonNhap> getListChiTietDonNhap(int maDN) {

Session session = HibernateUtil.getSessionFactory().openSession();

session.beginTransaction();

String sql = "FROM ChiTietDonNhap WHERE MaDN = " + maDN;

Query query = session.createQuery(sql);

List<ChiTietDonNhap> list = query.list();

session.close();

return list;

}

public ChiTietDonNhap getInfoChiTietDonNhap(ChiTietDonNhapId chiTietDonHangId) {

Session session = HibernateUtil.getSessionFactory().openSession();

session.beginTransaction();

ChiTietDonNhap chiTietDonNhap = (ChiTietDonNhap) session.get(ChiTietDonNhap.class, chiTietDonHangId);

session.close();

return chiTietDonNhap;

}

public boolean insertChiTietDonNhap(ChiTietDonNhap chiTietDonNhap) {

Session session = HibernateUtil.getSessionFactory().openSession();

try {

session.beginTransaction();

session.save(chiTietDonNhap);

session.getTransaction().commit();

return true;

} catch (Exception e) {

session.getTransaction().rollback();

e.printStackTrace();

return false;

} finally {

session.close();

}

}

public boolean updateChiTietDonNhap(ChiTietDonNhap chiTietDonNhap) {

if (getInfoChiTietDonNhap(chiTietDonNhap.getId()) == null) {

return false;

}

Session session = HibernateUtil.getSessionFactory().openSession();

try {

session.beginTransaction();

session.update(chiTietDonNhap);

session.getTransaction().commit();

return true;

} catch (Exception e) {

session.getTransaction().rollback();

e.printStackTrace();

return false;

} finally {

session.close();

}

}

public boolean deleteChiTietDonNhap(int maDN) {

Session session = HibernateUtil.getSessionFactory().openSession();

try {

session.beginTransaction();

Query query = session.createQuery("DELETE FROM ChiTietDonNhap where maDN = :maDN");

query.setParameter("maDN", maDN);

if (query.executeUpdate() > 0) {

session.getTransaction().commit();

return true;

} else {

session.getTransaction().commit();

return false;

}

} catch (Exception e) {

session.getTransaction().rollback();

e.printStackTrace();

return false;

} finally {

session.close();

}

}

}

## Viết mã cho ứng dụng

#### MainFrame

package com.polystorage.view;

import com.polystorage.entity.NhanVien;

import com.polystorage.helper.DialogHelper;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class MainFrame extends javax.swing.JFrame {

public static NhanVien nhanVien = null;

boolean isNhanVien = true;

public MainFrame() {

new JDialogChao(this, true).setVisible(true);

new JDialogLogin(this, true).setVisible(true);

if (nhanVien.getVaiTro().equals("Trưởng Phòng")) {

isNhanVien = false;

}

initComponents();

this.setExtendedState(JFrame.MAXIMIZED\_BOTH);

reloadRootPanel(new PanelDanhSachDonNhap(isNhanVien));

lblNhanVien.setText(nhanVien.getVaiTro() + ": " + nhanVien.getHoTen());

}

private void reloadRootPanel(JPanel panel) {

pnlRoot.removeAll();

pnlRoot.add(panel);

pnlRoot.updateUI();

}

private void btnDonNhapMouseClicked(java.awt.event.MouseEvent evt) {

reloadRootPanel(new PanelDanhSachDonNhap(isNhanVien));

}

private void btnNhaCungCapMouseClicked(java.awt.event.MouseEvent evt) {

reloadRootPanel(new PanelDanhSachNhaCungCap(isNhanVien));

}

private void btnKhachHangMouseClicked(java.awt.event.MouseEvent evt) {

reloadRootPanel(new PanelDanhSachKhachHang(isNhanVien));

}

private void btnSanPhamMouseClicked(java.awt.event.MouseEvent evt) {

reloadRootPanel(new PanelDanhSachSanPham(isNhanVien));

}

private void btnNhanVienMouseClicked(java.awt.event.MouseEvent evt) {

if (isNhanVien){

DialogHelper.showMessageDialog(null, "Bạn không có quyền truy cập");

} else {

reloadRootPanel(new PanelDanhSachNhanVien(isNhanVien));

}

}

private void btnDonXuatMouseClicked(java.awt.event.MouseEvent evt) {

reloadRootPanel(new PanelDanhSachDonXuat(isNhanVien));

}

private void btnTongSanPhamNhapXuatMouseClicked(java.awt.event.MouseEvent evt) {

reloadRootPanel(new PanelTongSanPhamNhapXuat(isNhanVien));

}

private void btnTonKhoMouseClicked(java.awt.event.MouseEvent evt) {

reloadRootPanel(new PanelHangTonKho());

}

#### PanelDanhSachSanPham

package com.polystorage.view;

import com.polystorage.dao.LoaiSanPhamDAO;

import com.polystorage.dao.NhaCungCapDAO;

import com.polystorage.dao.SanPhamDAO;

import com.polystorage.entity.LoaiSanPham;

import com.polystorage.entity.NhaCungCap;

import com.polystorage.entity.SanPham;

import com.polystorage.helper.DialogHelper;

import java.awt.Color;

import java.awt.Cursor;

import java.util.List;

import javax.swing.DefaultComboBoxModel;

public class FrameThemSanPham extends javax.swing.JFrame {

public FrameThemSanPham(SanPham sp) {

initComponents();

setLocationRelativeTo(null);

this.sp = sp;

this.fillToNhaCC();

this.fillToLoaiSanPham();

this.updateUI();

if (sp != null) {

capNhat = true;

txtMaSP.setEditable(false);

lblChucNang.setText("CẬP NHẬT");

this.showDetailSanPham();

}

}

private void showDetailSanPham() {

txtMaSP.setText(sp.getMaSp());

txtTenSP.setText(sp.getTenSp());

txtGia.setText(String.valueOf(sp.getGiaDv()));

cboNhaCungCap.setSelectedItem("(" + sp.getNhaCungCap().getMaNCC() + ") " + sp.getNhaCungCap().getTenNCC());

cboLoai.setSelectedItem("(" + sp.getLoaiSanPham().getMaLoai() + ") " + sp.getLoaiSanPham().getTenLoai());

if (sp.getDonViTinh().equals("Thùng")) {

cboDonViTinh.setSelectedIndex(0);

} else {

cboDonViTinh.setSelectedIndex(1);

}

}

private void fillToNhaCC() {

List<NhaCungCap> list = nhaCungCapDAOImpl.getListNhaCungCap(null);

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

model.removeAllElements();

for (NhaCungCap x : list) {

model.addElement("(" + x.getMaNCC() + ") " + x.getTenNCC());

}

cboNhaCungCap.setSelectedIndex(0);

}

private void fillToLoaiSanPham() {

List<LoaiSanPham> list = loaiSanPhamDAOImpl.getListLoai(null);

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

model.removeAllElements();

for (LoaiSanPham x : list) {

model.addElement("(" + x.getMaLoai() + ") " + x.getTenLoai());

}

cboLoai.setSelectedIndex(0);

}

private void updateUI() {

btnCancel.setCursor(new Cursor(Cursor.HAND\_CURSOR));

btnSave.setCursor(new Cursor(Cursor.HAND\_CURSOR));

}

private String checkInput() {

if (txtMaSP.getText().isEmpty()) {

txtMaSP.setBackground(Color.yellow);

txtMaSP.requestFocus();

return "Mã sản phẩm trống.";

} else if (!txtMaSP.getText().matches("[a-zA-Z0-9]+")) {

txtMaSP.setBackground(Color.yellow);

txtMaSP.requestFocus();

return "Sai định dạng mã sản phẩm.";

}

if (txtTenSP.getText().isEmpty()) {

txtTenSP.setBackground(Color.yellow);

txtTenSP.requestFocus();

return "Tên sản phẩm trống.";

} else if (!txtTenSP.getText().matches("[\\p{L} ]+")) {

txtTenSP.setBackground(Color.yellow);

txtTenSP.requestFocus();

return "Sai định dạng tên sản phẩm.";

}

if (txtTenSP.getText().isEmpty()) {

txtTenSP.setBackground(Color.yellow);

txtTenSP.requestFocus();

return "Tên sản phẩm trống.";

} else {

try {

Float.valueOf(txtTenSP.getText());

} catch (Exception e) {

txtTenSP.setBackground(Color.yellow);

txtTenSP.requestFocus();

return "Sai định dạng sản phẩm.";

}

}

return "";

}

private void setWhite() {

txtMaSP.setBackground(Color.white);

txtTenSP.setBackground(Color.white);

txtGia.setBackground(Color.white);

cboNhaCungCap.setSelectedIndex(0);

cboDonViTinh.setSelectedIndex(0);

cboLoai.setSelectedIndex(0);

}

private SanPham getModel() {

String maSP = txtMaSP.getText();

String tenSP = txtTenSP.getText();

float giaSP = Float.parseFloat(txtGia.getText());

String maNCC = (String) cboNhaCungCap.getSelectedItem();

NhaCungCap nhaCungCap = nhaCungCapDAOImpl.getInfoNhaCungCap(maNCC.substring(1, maNCC.lastIndexOf(")")));

String maLoai = (String) cboLoai.getSelectedItem();

LoaiSanPham loaiSanPham = loaiSanPhamDAOImpl.getInfoLoai(maLoai.substring(1, maLoai.lastIndexOf(")")));

String donViTinh = (String) cboDonViTinh.getSelectedItem();

SanPham sanPham = new SanPham(maSP, loaiSanPham, nhaCungCap, tenSP, donViTinh, giaSP);

return sanPham;

}

#### PanelDanhSachSanPham

package com.polystorage.view;

import com.polystorage.dao.SanPhamDAO;

import com.polystorage.entity.SanPham;

import com.polystorage.helper.DialogHelper;

import com.polystorage.helper.RemoveButton;

import java.awt.Color;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.List;

import javax.swing.JButton;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

public class PanelDanhSachSanPham extends javax.swing.JPanel {

public static PanelDanhSachSanPham reload;

private boolean isNhanVien;

public PanelDanhSachSanPham(boolean isNhanVien) {

reload = this;

this.isNhanVien = isNhanVien;

initComponents();

this.setTableUI();

this.fillToTableSanPham(null);

} JButton btnUpdate = new JButton();

List<SanPham> listSP;

int indexSP = 0;

SanPhamDAO sanPhamDAOImpl = new SanPhamDAO();

private void setTableUI() {

tblSanPham.getTableHeader().setFont(new Font("Segoe UI", Font.PLAIN, 15));

tblSanPham.getTableHeader().setOpaque(false);

//tblNhanVien.getTableHeader().setBackground(new Color(32, 136, 203));

tblSanPham.getTableHeader().setBackground(new Color(255, 255, 255));

tblSanPham.getTableHeader().setForeground(new Color(0, 0, 0));

tblSanPham.setRowHeight(25);

}

public void fillToTableSanPham(String tenSP) {

listSP = sanPhamDAOImpl.getListSanPham(tenSP);

if (listSP != null) {

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

ActionListener actionListener = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

removeNhanVien();

}

};

tblSanPham.getColumnModel().getColumn(6).setCellRenderer(new RemoveButton(tblSanPham, 6, actionListener));

model.setRowCount(0);

for (SanPham x : listSP) {

Object[] row = new Object[]{x.getMaSp(), x.getTenSp(), x.getLoaiSanPham().getTenLoai(), x.getNhaCungCap().getTenNCC(), x.getDonViTinh(), x.getGiaDv()};

model.addRow(row);

}

}

}

private void removeNhanVien() {

if (isNhanVien) {

DialogHelper.showMessageDialog(null, "Nhân viên không có quyền xóa");

} else {

int index = tblSanPham.getSelectedRow();

String maSP = (String) tblSanPham.getValueAt(index, 0);

int ok = JOptionPane.showConfirmDialog(null, "Xác nhận xóa sản phẩm " + maSP, "Bạn muốn xóa?", JOptionPane.YES\_NO\_OPTION);

if (ok == 0) {

if (sanPhamDAOImpl.deleteSanPham(maSP)) {

PanelDanhSachSanPham.reload.fillToTableSanPham(null);

DialogHelper.showMessageDialog(null, "Xóa sản phẩm " + maSP + " thành công");

} else {

DialogHelper.showMessageDialog(null, "Xóa sản phẩm " + maSP + " thất bại");

}

}

}

}

## Các lớp tiện ích

#### DialogHelper

Lớp DialogHelper chứa các phương thức đơn giản hóa JOptionPane

package com.polystorage.helper;

import java.awt.Component;

import javax.swing.JOptionPane;

public class DialogHelper {

static String title = "Hệ thống quản lý kho";

public static void showMessageDialog(Component parent, String mess) {

JOptionPane.showMessageDialog(parent, mess, title, JOptionPane.INFORMATION\_MESSAGE);

}

public static boolean confirm(Component parent, String mess) {

int ok = JOptionPane.showConfirmDialog(parent, mess, title, JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE);

return ok == JOptionPane.YES\_OPTION;

}

}

#### ProcessImage

package com.polystorage.helper;

import java.awt.Image;

import javax.swing.ImageIcon;

import javax.swing.JLabel;

public class ProcessImage {

private static String pathAva = System.getProperty("user.dir") + "/src/image/";

public static void setAvatar(String path, JLabel lblAvatar) {

ImageIcon imageIcon = new ImageIcon(pathAva + path); //get and set avt

Image image = imageIcon.getImage(); //

Image newimg = image.getScaledInstance(160, 175, java.awt.Image.SCALE\_SMOOTH); // resize ảnh thành 160x175

imageIcon = new ImageIcon(newimg); // transform it back

lblAvatar.setIcon(imageIcon); //set Avatar vào label.

lblAvatar.setToolTipText(path);

}

public static String getPathAva() {

return pathAva;

}

public static void setPathAva(String pathAva) {

ProcessImage.pathAva = pathAva;

}

}

# Kiểm thử

## Kiểm thử form quản lý nhân viên

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Thao tác** | | |
| **Thêm mới** | **Cập nhật** | **Xóa** |
| Mã nhân viên | Không để trống  Không trùng |  | Không xóa chính mình |
| Mật khẩu | Ít nhất 6 ký tự | |  |
| Họ và tên | Chỉ chứa alphabet và ký tự trắng | |  |

## Kiểm thử form quản lý khách hàng, nhà cung cấp

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Thao tác** | |
| **Thêm mới** | **Cập nhật** |
| Mã Nhà Cung Cấp, Mã Khách hàng | 0-10 ký tự  Không trùng |  |
| Họ và tên | Chỉ chứa alphabet và ký tự trắng | |
| Số điện thoại | Phải là 10 số nguyên dương | |
| Địa chỉ email | Phải đúng định dạng email | |

## Kiểm thử form quản lý sản phẩm

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Thao tác** | |
| **Thêm mới** | **Cập nhật** |
| Mã sản phẩm | 0-10 ký tự  Không trùng |  |
| Tên sản phẩm | Chỉ chứa alphabet và ký tự trắng | |
| Gía | Phải là số | |

## Kiểm thử form quản lý đơn nhập, đơn xuất

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Thao tác** | |
| **Thêm mới** | **Cập nhật** |
| Ngày xuất | Không được lớn hơn ngày hiện tại | |

# 

# 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 MySQL
* Tạo CSDL PolyStorage 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: admin và password: admin
* Cấp các tài khoản có vai trò là trưởng phòng
* Đăng nhập tài khoản mới và xóa tài khoản admin
* Xem hướng dẫn sử dụng trong menu trợ giúp của phần mềm