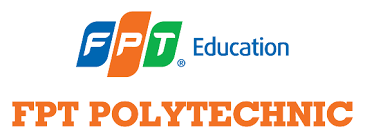
TRƯỜNG CAO ĐẲNG FPT POLYTECHNIC



**Dự Án 1**

**Đề tài: Phần mềm quản lý bán nước**

**Sinh viên thực hiện**:

Lê Viết Huy

Phan Tiểu Thiên

Võ Thị Thành

**Lớp**: SD18307

**Chuyên ngành**: Phát Triển Phần Mềm

**Giáo viên hướng dẫn**: Nguyễn Trần Nhật Kha

**Mục Lục**

[1. Phân Tích 4](#_Toc152942978)

[1.1 HIỆN TRẠNG 4](#_Toc152942979)

[1.2 Yêu cầu hệ thống 4](#_Toc152942980)

[✔ Yêu cầu chức năng nghiệp vụ 4](#_Toc152942981)

[✔ Yêu cầu về bảo mật 5](#_Toc152942982)

[✔ Yêu cầu về môi trường công nghệ 5](#_Toc152942983)

[1.3 USE CASE 5](#_Toc152942984)

[1.4 Phân công 10](#_Toc152942985)

[2. THIẾT KẾ 11](#_Toc152942987)

[2.1 MÔ HÌNH TRIỂN KHAI 11](#_Toc152942988)

[2.2 THIẾT KẾ CSDL 11](#_Toc152942989)

[2.2.1 Sơ đồ quan hệ thực thể 11](#_Toc152942990)

[2.2.2 Thiết kế chi tiết các thực thể 14](#_Toc152942991)

[2.3 THIẾT KẾ GIAO DIỆN 16](#_Toc152942992)

[2.3.1 Sơ đồ tổ chức giao diện 16](#_Toc152942993)

[2.3.2 Thiết kế giao diện cho các chức năng nghiệp vụ 17](#_Toc152942994)

[3. THỰC HIỆN VIẾT MÃ 39](#_Toc152942995)

[3.1 VIẾT MÃ TẠO CSDL 39](#_Toc152942996)

[3.1.1 Tạo CSDL 39](#_Toc152942997)

[3.1.2 SQL truy vấn và thao tác bảng 42](#_Toc152942998)

[3.1.3 Các thủ tục lưu tổng hợp thống kê 45](#_Toc152942999)

[3.2 LẬP TRÌNH JDBC 52](#_Toc152943000)

[3.2.1 Lớp hỗ trợ 52](#_Toc152943001)

[3.2.2 Model class - Các lớp mô tả dữ liệu 58](#_Toc152943002)

[3.2.3 DAO Class - Các lớp truy xuất dữ liệu 76](#_Toc152943003)

[3.3 VIẾT MÃ CHO ỨNG DỤNG 98](#_Toc152943004)

[3.3.1 Viết mã cho các chức năng 98](#_Toc152943005)

[4. KIỂM THỬ 181](#_Toc152943006)

[4.1 Kiểm thử form chung 181](#_Toc152943007)

[4.2 Kiểm thử đăng nhập 183](#_Toc152943008)

[4.3 Kiểm thử quản lý nhân viên 185](#_Toc152943009)

[4.4 Kiểm thử nhà cung cấp 188](#_Toc152943010)

[4.5 Kiểm thử quản lý khách hàng 192](#_Toc152943011)

[4.6 Kiểm thử sản phẩm 196](#_Toc152943012)

[4.7 Kiểm thử hóa đơn 200](#_Toc152943013)

[4.8 Kiểm thử thống kê 204](#_Toc152943014)

[5. ĐÓNG GÓI VÀ TRIỂN KHAI 206](#_Toc152943015)

[5.1 Dẫn chuyển đổi jar thành exe 206](#_Toc152943016)

[5.2 Hướng dẫn cài đặt triển khai 206](#_Toc152943017)

[5.3 Hướng dẫn sử dụng phần mềm 206](#_Toc152943018)

# 1. Phân Tích

## 1.1 HIỆN TRẠNG

Một doanh nghiệp bán nước mới mở cửa với sự tăng trưởng nhanh chóng trong lượng khách hàng. Để nâng cao chất lượng phục vụ và quản lý hiệu quả, chủ cửa hàng quyết định chuyển từ quản lý thủ công sang quản lý bằng phần mềm. Các vấn đề trong quản lý hiện tại bao gồm:

Ghi chép thủ công thông tin về sản phẩm và bán nước:

* Mất nhiều thời gian và có thể dẫn đến sai sót khi kiểm tra thông tin sản phẩm và tồn kho.
* Có khả năng mất thông tin quan trọng khi sử dụng hệ thống thủ công.

Quản lý hóa đơn và thanh toán:

* Quá trình lập hóa đơn và thu chi vẫn thực hiện chủ yếu thủ công, gây tốn thời gian và có thể dẫn đến sai sót trong ghi chép.

Thống kê doanh số bán hàng:

* Khó khăn trong việc đánh giá doanh số bán hàng hàng ngày và theo chu kỳ.
* Thiếu thông tin chi tiết về các sản phẩm bán chạy và tồn kho.

Quản lý khách hàng:

* Không có hệ thống tự động theo dõi thông tin khách hàng và lịch sử mua hàng.
* Gặp khó khăn khi tìm kiếm thông tin về khách hàng trong quá trình phục vụ.

Lưu trữ Thông tin:

* Hạn chế khả năng lưu trữ và truy xuất thông tin lâu dài do sử dụng hệ thống thủ công.

Trước tình hình đó vấn đề đặt ra là xây dựng một hệ thống thông tin đáp ứng được các yêu cầu cơ bản sau:

* Giảm khối lượng ghi chép nhằm lưu trữ thông tin.
* Cập nhật dữ liệu nhanh chóng, chính xác và kịp thời.
* Thống kê được thu chỉ và tình hình doanh thu của cửa hàng.
* Có khả năng lưu trữ thông tin lâu dài, đảm bảo truy vấn nhanh khi cần thiết

## 1.2 Yêu cầu hệ thống

Cửa hàng mong muốn xây dựng một phần mềm để giải quyết khó khăn trên.

### Yêu cầu chức năng nghiệp vụ

##### Quản lý nhân viên

* + - * Quản lý nhà cung cấp

##### Quản lý sản phẩm

##### Quản lý khách hàng

##### Quản lý bán hàng và hóa đơn

* + - * Thống kê doanh thu

### Yêu cầu về bảo mật

##### Tất cả mọi thành viên phải đăng nhập mới sử dụng được phần mềm

* + - * Quản lý 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

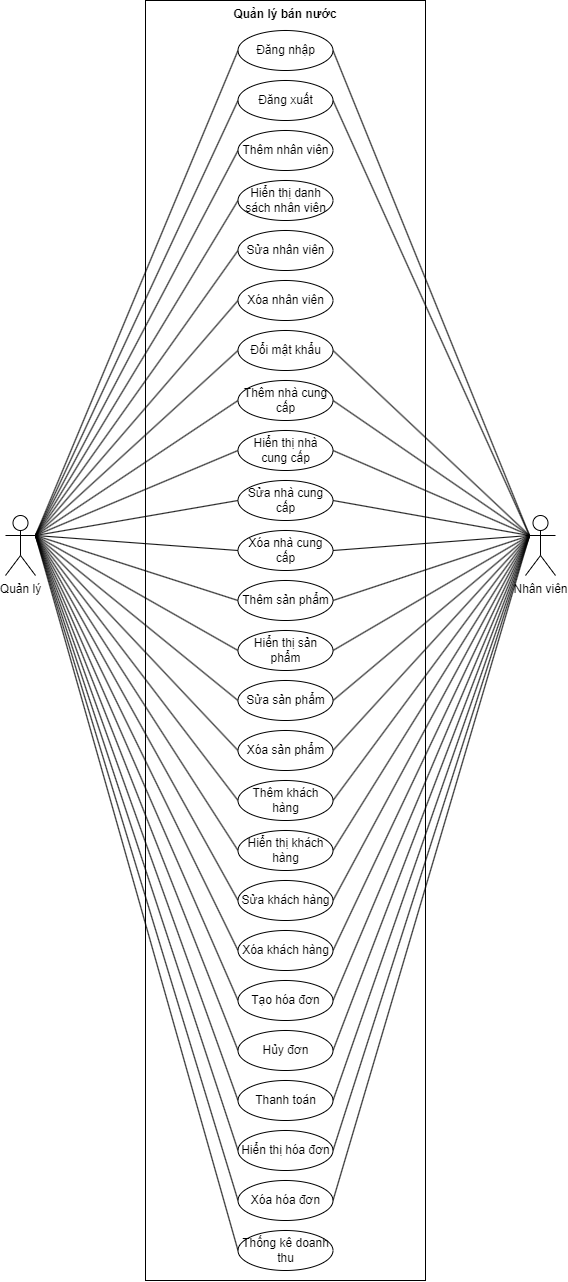
### Yêu cầu về môi trường công nghệ

##### Ứng dụng phải được thực với công nghệ Swing và JDBC chạy trên mọi hệ điều hành với môi trường JDK 17

* + - * Hệ quản trị CSDL SQL Server 2022

## 1.3 USE CASE

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



Chi tiết các chức năng

Đăng Nhập và Đăng Xuất:

* Đăng nhập :Đăng nhập bằng tài khoản người dùng vào hệ thống.
* Đăng xuất :Đăng xuất người dùng khỏi hệ thống.

Quản lý Nhân Viên:

* Thêm nhân viên :Thêm một nhân viên mới vào danh sách nhân viên.
* Xem danh sách nhân viên: Hiển thị danh sách tất cả nhân viên.
* Sửa nhân viên : Sửa thông tin của một nhân viên trong danh sách.
* Xóa nhân viên: Xóa một nhân viên khỏi danh sách nhân viên.
* Tìm kiếm nhân viên : Tìm một nhân viên bằng mã nhân viên trong danh sách.
* Đổi mật khẩu: Dùng email của nhân viên để đổi mật khẩu.

Quản lý Nhà Cung Cấp

* Thêm nhà cung cấp: Thêm một nhà cung cấp mới vào danh sách nhà cung cấp.
* Xem danh sách nhà cung cấp: Hiển thị danh sách tất cả nhà cung cấp.
* Sửa nhà cung cấp: Sửa thông tin của một nhà cung cấp trong danh sách.
* Xóa nhà cung cấp :Xóa một nhà cung cấp khỏi danh sách nhà cung cấp.
* Tìm kiếm nhà cung cấp: Tìm một nhà cung cấp bằng mã trong danh sách.

Quản lý Sản Phẩm:

* Thêm sản phẩm: Thêm một sản phẩm mới vào danh sách sản phẩm.
* Xem danh sách sản phẩm: Hiển thị danh sách tất cả sản phẩm.
* Sửa sản phẩm Sửa thông tin của một sản phẩm trong danh sách sản phẩm.
* Xóa sản phẩm: Xóa một sản phẩm khỏi danh sách sản phẩm.
* Tìm kiếm sản phẩm: Tìm một sản phẩm bằng mã sản phẩm trong danh sách.

Quản lý Khách Hàng:

* Thêm khách hàng: Thêm một khách hàng mới vào danh sách khách hàng.
* Xem danh sách khách hàng: Hiển thị danh sách tất cả khách hàng.
* Sửa khách hàng: Sửa thông tin của một khách hàng trong danh sách khách hàng.
* Xóa khách hàng: Xóa một khách hàng khỏi danh sách khách hàng.
* Tìm kiếm khách hàng: Tìm một khách hàng bằng mã khách hàng trong danh sách.

Quản lý Hóa Đơn:

* Tạo hóa đơn: Tạo một hóa đơn mới và thêm sản phẩm vào đó.
* Hủy đơn: Hủy một hóa đơn chưa thanh toán.
* Thanh toán: Thực hiện thanh toán cho một hóa đơn.
* In hóa đơn: Xuất file hóa đơn.
* Xem danh sách hóa đơn: Hiển thị danh sách tất cả hóa đơn đã tạo.
* Tìm kiếm hóa đơn: Tìm một hóa đơn bằng mã hóa đơn có trong danh sách.
* Thống kê doanh thu: Thống kê doanh thu của cửa hàng.

Chi tiết về yêu cầu bảo mật

Quản Lý: Được phép sử dụng tất cả.

Nhân viên: Không được phép thực hiện các chức năng quản lý nhân viên, quản lý nhà cung cấp. Không được phép xem chức năng thống kê doanh thu

* + - Thông tin các thực thể

##### Nhân viên

* + - * + Mã nhân viên

##### Mật khẩu

* + - * + Họ tên
        + Vai trò(trưởng phòng hay nhân viên)
        + Trạng Thái

##### Nhà cung cấp

* + - * + Mã nhà cung cấp

##### Tên nhà cung cấp

* + - * + Điện thoại

##### Địa chỉ

* + - * + Email

##### GhiChu

* + - * Sản phẩm
* Mã sản phẩm
* Tên sản phẩm
* Số lượng
* Đơn giá
* Hình ảnh
* Mô tả
* Mã nhà cung cấp
* Mã nhân viên
* Mã loại sản phẩm

##### Khách hàng

* + - * + Mã khách hàng

##### Tên khách hàng

* + - * + Ngày Sinh
        + Giới tính

##### Điện thoại

##### Email

##### Ghi chú

##### Ngày đăng ký

* + - * Hóa đơn

##### Mã hóa đơn

* + - * + Mã khách hàng
        + Mã nhân viên
        + Tổng tiền

##### Ngày mua hàng

* + - * + Trạng thái (đã hoặc chưa thanh toán)
      * Hóa đơn chi tiết
* Mã hóa đơn chi tiết
* Mã hóa đơn
* Mã sản phẩm
* Số lượng
* Thành tiền

## 1.4 Phân công

## 

# 2. THIẾT KẾ

## 2.1 MÔ HÌNH TRIỂN KHAI

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

* + - Cần một máy cài SQL Server 2022. Máy này cần hệ điều hành window xp trở lên

##### Các máy nhân viên phòng đào tạo cài phần mềm QLBN. Các máy nhân viên cần JDK 17 với hệ điều hành bất ký

## 2.2 THIẾT KẾ CSDL

### 2.2.1 Sơ đồ quan hệ thực thể

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

Quản lý (Manager):

* Mô tả: Quản lý hệ thống và chức năng của nhân viên. Có khả năng thêm, sửa, và xóa nhân viên, sản phẩm, loại sản phẩm, khách hàng, nhà cung cấp và thông tin hóa đơn.

Nhân viên (Employee):

* Mô tả: Thực hiện các chức năng thêm, sửa, và xóa. Có thể quản lý sản phẩm, khách hàng, loại sản phẩm và tạo hóa đơn.

Sản phẩm (Product):

* Mô tả: Đại diện cho các đồ uống bán trong quán. Bao gồm mã sản phẩm, tên sản phẩm, giá, hình ảnh và mô tả.

Khách hàng (Customer):

* Mô tả: Bao gồm mã khách hàng, tên, địa chỉ, số điện thoại và email.

Hóa đơn (Invoice):

* Mô tả: Ghi lại thông tin về giao dịch bán hàng. Bao gồm mã hóa đơn, ngày tạo, tổng tiền và trạng thái thanh toán.

Chi tiết hóa đơn (Invoice Detail):

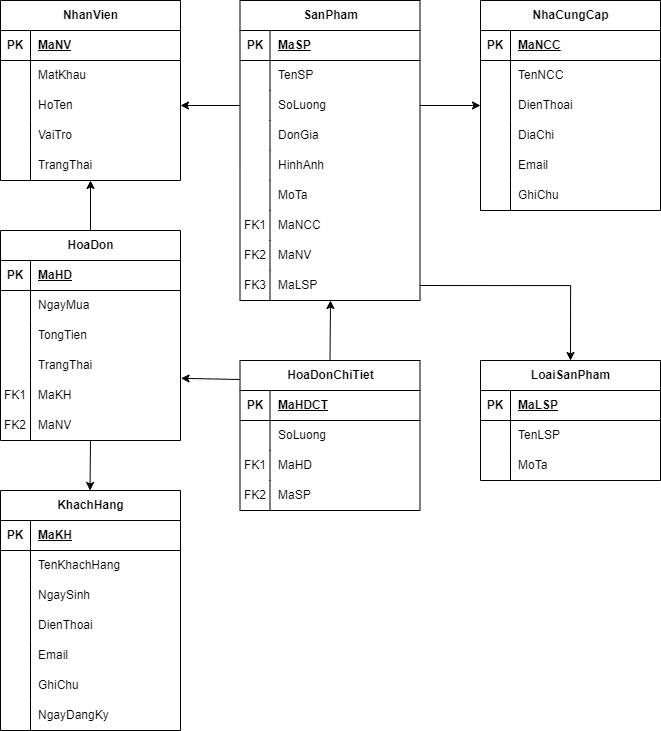
* Mô tả: Ghi chi tiết về từng sản phẩm trong hóa đơn. Bao gồm mã chi tiết hóa đơn, mã hóa đơn, mã sản phẩm, số lượng và giá.

Chức năng chính của mỗi vai trò:

* Quản lý: Quản lý toàn bộ hệ thống, bao gồm thêm, sửa, và xóa nhân viên, sản phẩm, khách hàng, và hóa đơn.
* Nhân viên: Thực hiện các chức năng cụ thể như quản lý sản phẩm, khách hàng, và tạo hóa đơn. Sau khi phân tích chúng ta có thể thấy các thực thể có mối quan hệ với nhau như sơ đồ ERD sau

#### 2.2.1.1 ERD Diagram level 1

#### 2.2.1.2 ERD Diagram level 2

******

##### Yêu cầu ràng buộc quan hệ:

* + - * + Tất cả các relationship đều là CASCADE (cập nhật dây chuyền)

### 2.2.2 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 | NVARCHAR(50) | PK, NOT NULL | Mã nhân viên |
| MatKhau | NVARCHAR(50) | NOT NULL | Mật khẩu đăng nhập |
| Email | NVARCHAR(50) | NOT NULL | Email |
| TenNV | NVARCHAR(50) | NOT NULL | Họ và tên |
| VaiTro | BIT | DEFAULT 0 | Vai trò, 1 là trưởng phòng |
| TrangThai | BIT | DEFAULT 1 | Trạng thái đi làm hay nghỉ |

#### Sản Phẩm

Bảng SanPham lưu thông tin các chuyên đề có cấu trúc như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu | Ràng buộc | Ghi chú |
| MaSP | NVARCHAR(50) | PK, NOT NULL | Mã Sản Phẩm |
| TenSP | NVARCHAR(50) | NOT NULL | Tên Sản Phẩm |
| DonGia | FLOAT | NOT NULL | Đơn Gía |
| SoLuong | INT | NOT NULL | Số Lượng |
| HinhAnh | NVARCHAR(50) | NOT NULL | Hình ảnh |
| MoTa | NVARCHAR(255) | NOT NULL | Mô tả |
| MaNCC | NVARCHAR(50) | FK,NOT NULL | Mã Nhà Cung Cấp |
| Mã NV | NVARCHAR(50) | FK, NOT NULL | Mã Nhân Viên |

#### Khách Hàng

Bảng KhachHang lưu thông tin các những người đã đến trung tập đăng ký học có cấu trúc như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu | Ràng buộc | Ghi chú |
| MaKH | NCHAR(7) | PK, NOT NULL | Mã người học |
| TenKH | NVARCHAR(50) | NOT NULL | Họ và tên |
| GhiChu | NVARCHAR(50) | NOT NULL | Ghi Chú |
| NgaySinh | DATE | NOT NULL | Ngày sinh |
| DienThoai | NVARCHAR(24) | NOT NULL | Điện thoại |
| Email | NVARCHAR(50) | NOT NULL | Email |
| ngayDk | DATE | NOT NULL | ngày đăng ký |

#### Nhà Cung Cấp

Bảng NhaCungCap lưu thông tin các khóa học được tạo ra từ các chuyên đề có cấu trúc như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu | Ràng buộc | Ghi chú |
| MaNCC | NVARCHAR(50) | PK, Tự tăng | Mã Nhà Cung Cấp |
| TenNCC | NVARCHAR(50) | NOT NULL | Tên Nhà Cung Cấp |
| DienThoai | NVARCHAR(50) | NOT NULL | Điện Thoại |
| DiaChi | NVARCHAR(50) | NOT NULL | Địa Chỉ |
| Email | NVARCHAR(50) | NOT NULL | Email |
| GhiChu | NVARCHAR(MAX) | NOT NULL | Ghi chú về nhà cung cấp |

#### Hóa Đơn

##### Bảng HoaDon lưu thông tin các học viên tham gia vào các khóa học có cấu trúc như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu | Ràng buộc | Ghi chú |
| MaHD | INT | PK, NOT NULL | Mã Hóa Đơn |
| NgayMua | DATE | NOT NULL | Ngày Mua |
| TrangThai | BIT | NOT NULL | Mã khách hàng |
| MaKH | NVARCHAR(50) | FK, NOT NULL | Mã Khách Hàng |
| MaNV | NVARCHAR(50) | FK, NOT NULL | Mã Nhân Viên |

#### Hóa Đơn chi tiết

##### Bảng HoaDonChiTiet lưu thông tin các học viên tham gia vào các khóa học có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu | Ràng buộc | Ghi chú |
| MaHDCT | INT | PK, NOT NULL | Mã Hóa Đơn Chi Tiết |
| SoLuong | INT | NOT NULL | Số Lượng |
| MaHD | INT | FK, NOT NULL | Mã Hóa Đơn |
| MaSP | NVARCHAR(50) | FK, NOT NULL | Mã Sản Phẩm |

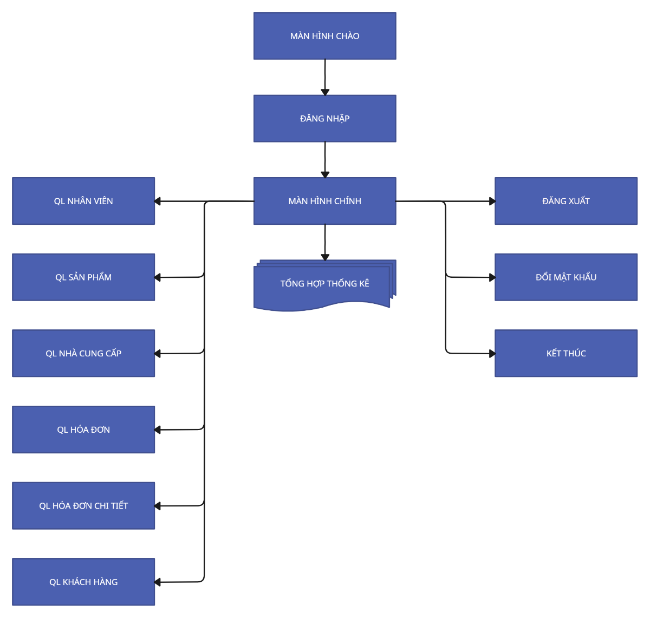
## 2.3 THIẾT KẾ GIAO DIỆN

### 2.3.1 Sơ đồ tổ chức giao diện

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

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

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



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

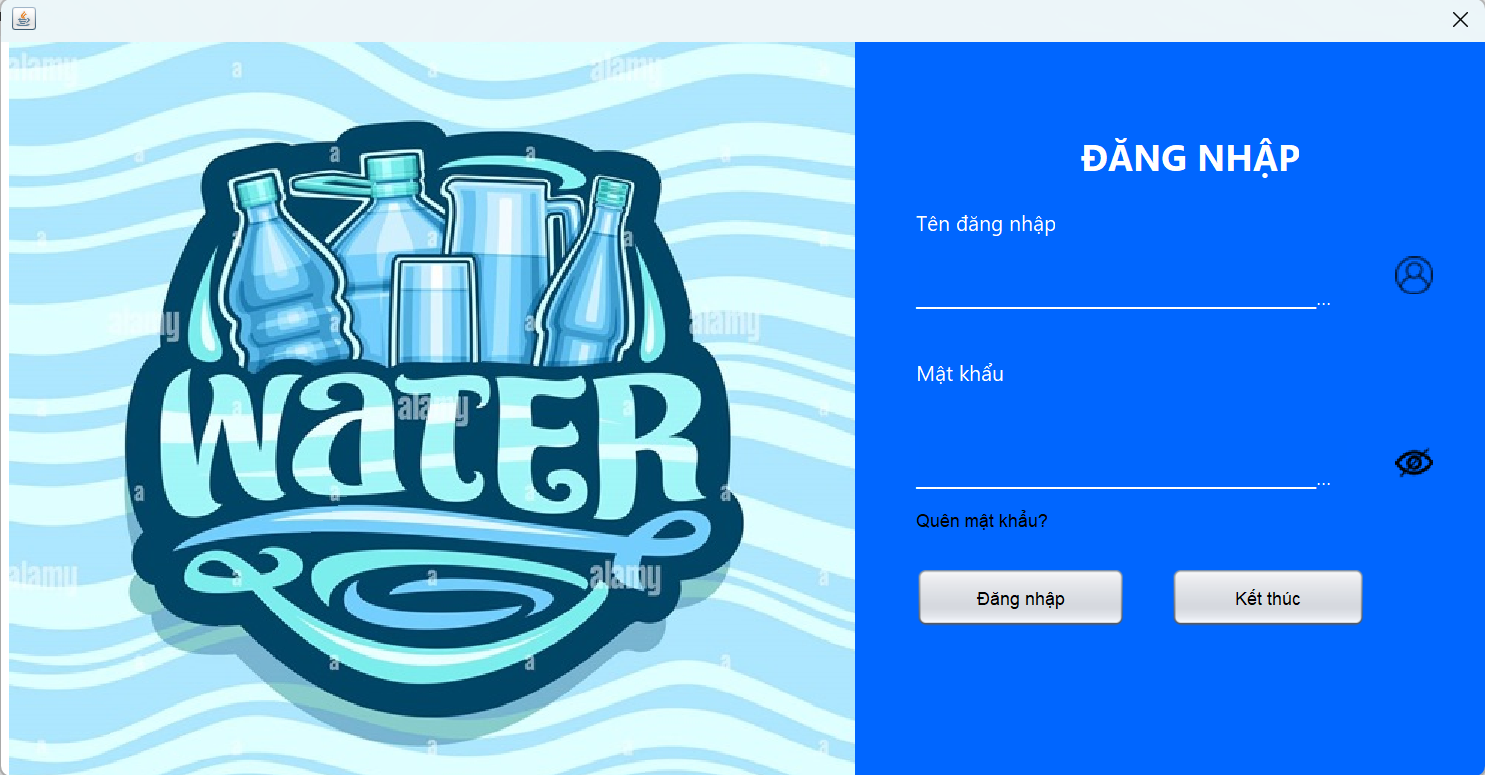
Trong phần thiết kế giao diện các bạn nên sử dụng gói hình ảnh phù hợp với ứng dụng.

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 |
| ButtonGroup | bgr | Nhóm các button |
| JComboBox | cbo | Hộp chọn xổ xuống |
| JListBox | lst | Hộp chọn nhiều |
| JTree | tree | Cây |
| JMenu | mnu | Thực đơn |
| JMenuItem | mni | Mục menu |
| JPanel | pnl | Khung chứa |
| JTabbedPane | tab |  |
| JTable | tbl |  |
| … |  |  |

#### Đăng nhập



Các thành phần giao diện

|  |  |  |
| --- | --- | --- |
| **Componel** | **Kiểu** | **Thuộc tính** |
| DangNhapJDialog | JDialog | Layout:AbsoluteLaout defaultCloseOperation:DO\_NOTHING |
| lblTitle | JLabel | Text:ĐĂNG NHẬP  Font:Tahoma, Bold, 24  Foreground: [255,255,255] |
| lblView | JLabel | Icon: View |
| pnlForm | JPanel | Layout.Derection:center |

|  |  |  |
| --- | --- | --- |
| lblMaNV | JLabel | Text: Tên đăng nhập |
| txtMaNV | JTextField |  |
| lblMatKhau | JLabel | Text: Mật khẩu |
| txtMatKhau | JPassword |  |
| btnDangNhap | JButton | Text:Đăng nhập |
| lblQuenMK | JLabel | Text: Quên mật khẩu? |
| btnKetThuc | JButton | Text:Kết thúc |

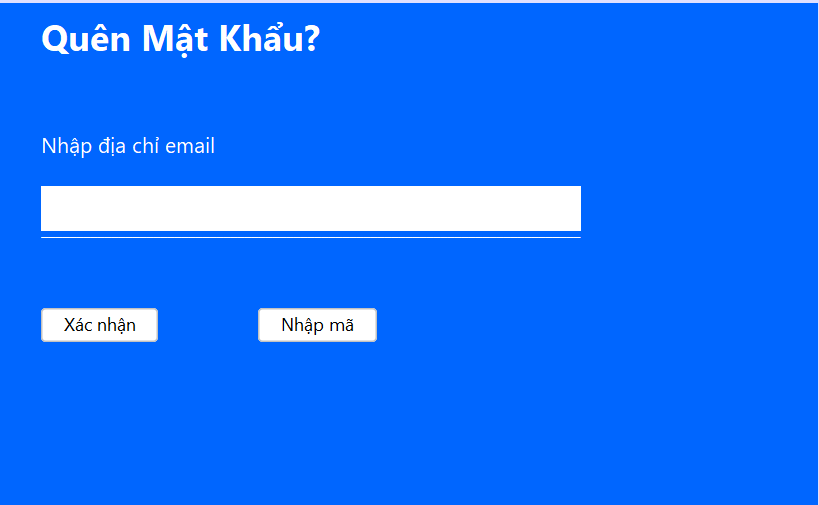
#### 2.3.2.2 Giao diện chính



Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Componel** | **Kiểu** | **Thuộc tính** |
| 1 | MainJFrame | JFrame | Title: HỆ THỐNG QUẢN LÝ BÁN NƯỚC  Layout:BorderLayout |
| 2 | btnDangXuat | Button | Text: Đăng Nhập  Icon : |
| 3 | btnNhanVien | button | Text: Nhân Viên  Icon: |
| 4 | btnNhaCungCap | button | Text : Nhà Cung Cấp  Icon |
| 5 | btnHoaDon | button | Text : Hóa Đơn  Icon : |
|  | btnSanPham | button | Text :Sản Phẩm  Icon : |
|  | btnKhachHang | button | Text : Khách Hàng  Icon : |
|  | btnThongKe | button | Text : Thống Kê  Icon : |
|  | btnKetThuc | button | Text : Kết Thúc  Icon : |

#### 2.3.2.3 Quên mật khẩu



Các thành phần giao diện

|  |  |  |
| --- | --- | --- |
| **Component** | **Kiểu** | **Thuộc tính** |
| txtEmail | JTextField | Email |
| lblTitle | JLabel | Quên mật khẩu |
| lblEmail | Jlabel | Nhập địa chỉ email |
| btnXacNhan | Jbutton | Xác nhận |
| btnNhapMa | Jbutton | Nhập mã |
| lblMatKhau | Jlabel | Mật khẩu mới |
| lblXacNhanMK | Jlabel | Xác Nhân Mật khẩu |
| txtMatKhau | JtextField |  |
| txtXacNhanMK | JtextField |  |

#### 2.3.2.4 Quản lý nhân viên

Các thành phẩn giao diện

|  |  |  |
| --- | --- | --- |
| **Component** | **Kiểu** | **Thuộc tính** |
| txtMaNV | JTextField | Text: Mã NV |
| txtMatKhau | JTextField | Text: Mật khẩu |
| lblMaNV | JLabel |  |
| txtMatKhau2 | JTextField |  |
| llblMatKhau | JLabel | Text: Mật Khẩu |
| lblMatKhau2 | JLabel |  |
| lblHoTen | JLabel | Text: Họ và tên |
| txtHoTen | JTextField |  |
| lblVaiTro | JLabel | Text: Vai trò nhân viên |
| lblTrangThai | JLabel | Text: Trạng Thái |
| rboQuanLy | JRadioButton | Text :Quản Lý |
| rboNhanVien | JRadioButton | Text : Nhân Viên |
| rboDangLamViec | JRadioButton | Text: Đang làm việc |
| rboDaNghiViec | JLabel | Text: Đã Nghỉ Việc |
| lblDanhSachNhanVien | JLabel | Text : Danh sách nhân viên |
| lblTimKiem | JLabel |  |
| txtTimKiem | JTextField |  |
| tblDanhSachNhanVien | JTable |  |
| btnInsert | JButton | Text:Thêm |
| btnUpdate | JButton | Text:Sửa |
| btnDelete | JButton | Text:Xóa |
| btnClear | JButton | Text:Mới |
| btnFirst | JButton | Text: |< |
| btnPrev | JButton | Text: << |
| btnNext | JButton | Text: >> |
| btnLast | JButton | Text: >| |

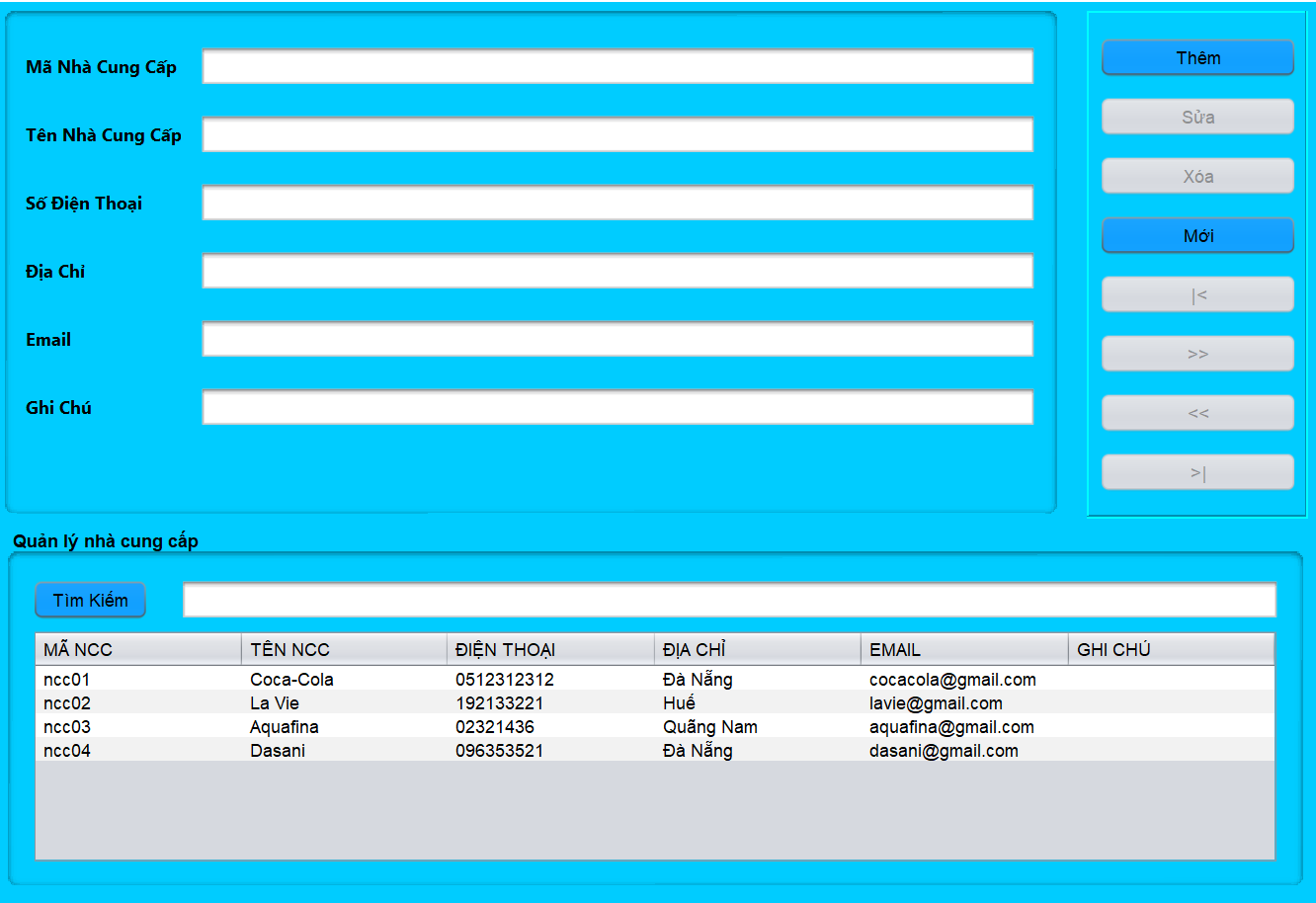
#### 2.3.3.5 Quản lý khách hàng

#### 

Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | txtMaKH | JTextField | Text: Mã khách hàng |
| 2 | txtTenKH | JTextField | Text: Tênkhách hàng |
| 3 | txtNgaySinh | JTextField | Text: Ngày sinh |
| 4 | txtDienThoai | JTextField | Text: Điện thoại |
| 5 | txtEmail | JTextField | Text: Email |
| 6 | txtGhiChu | JTextField | Text: Ghi Chú |
| 7 | txtNgayDangKy | JTextField | Text: Ngày Đăng Ký |
| 8 | lblKhachHang | JLabel | Text : Danh sách khách hàng |
| 9 | lblTimKiem | JLabel |  |
| 10 | txtTimKiem | JTextField |  |
| 11 | tblDanhSacKhachHang | JTable |  |
| 12 | btnInsert | JButton | Text:Thêm |
| 13 | btnUpdate | JButton | Text:Sửa |
| 14 | btnDelete | JButton | Text:Xóa |
| 15 | btnClear | JButton | Text:Mới |
| 16 | btnFirst | JButton | Text: |< |
| 17 | btnPrev | JButton | Text: << |
| 18 | btnNext | JButton | Text: >> |
| 19 | btnLast | JButton | Text: >| |

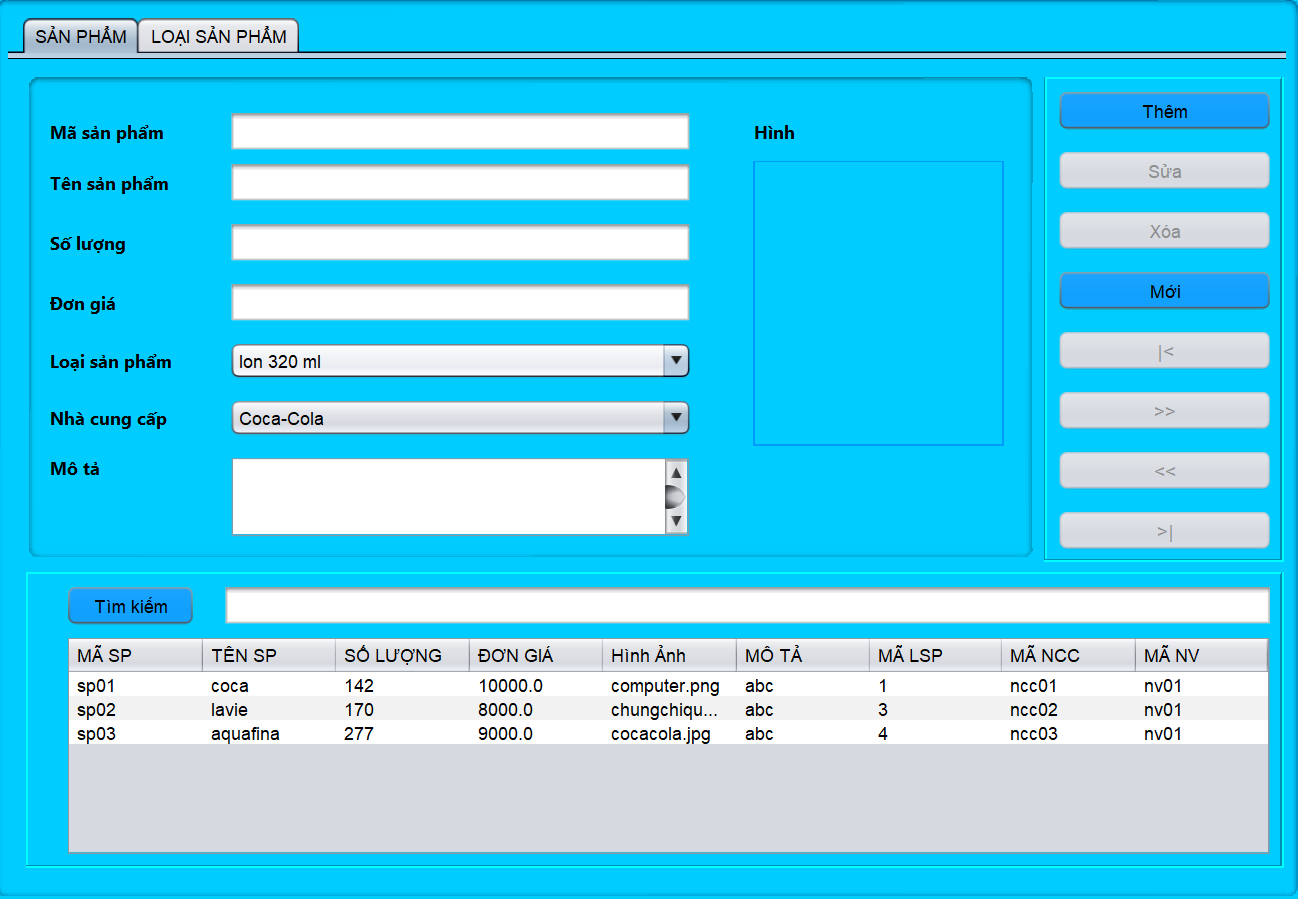
#### 2.3.3.6 Quản Lý Nhà Cung Cấp



Các thành phần giao diện

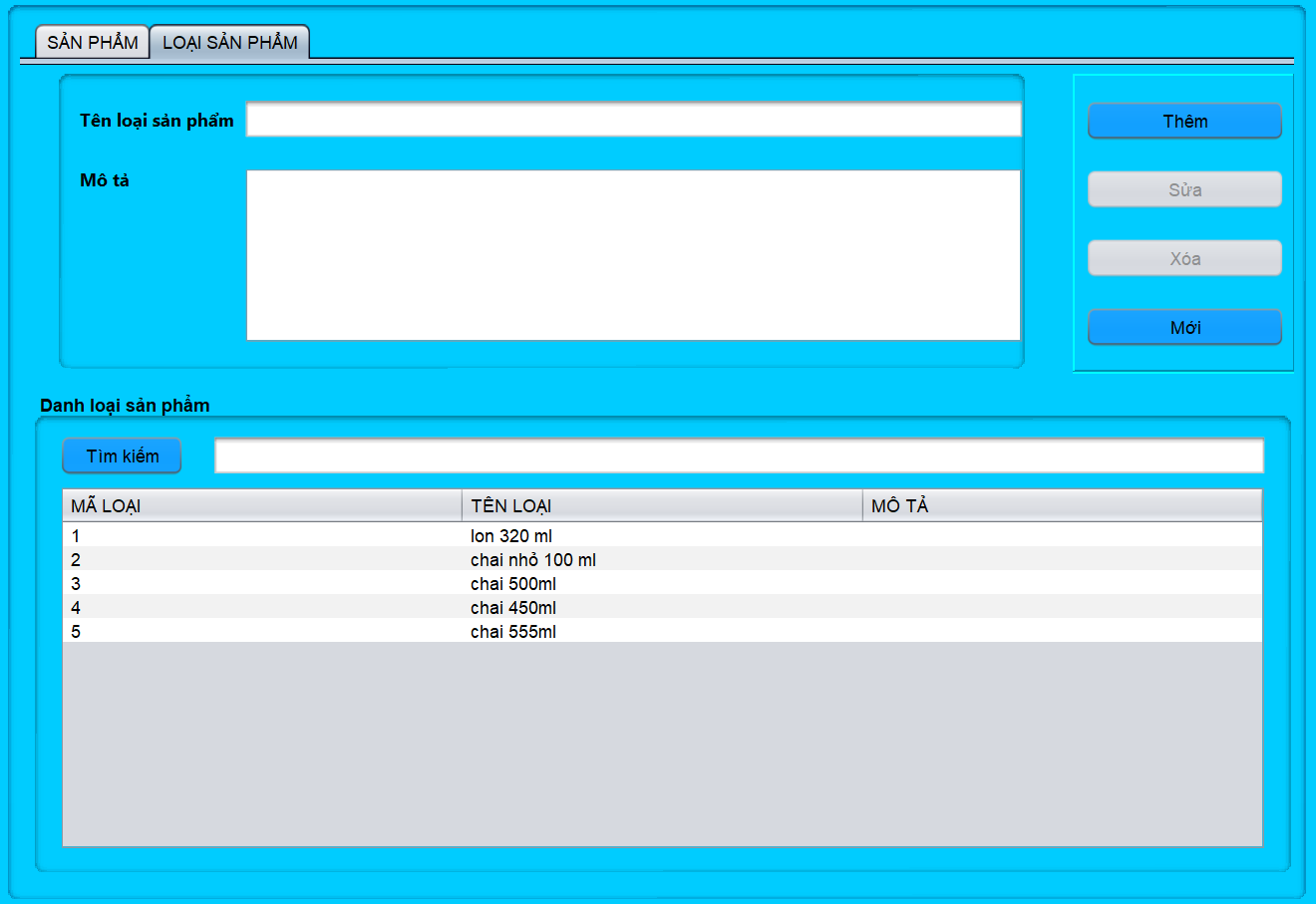
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | txtMaNCC | JTextField | Text: Mã nhà cung cấp |
| 2 | txtTenNCC | JTextField | Text: Tên nhà cung cấp |
| 3 | lblMaNCC | JLabel | Text: Mã Nhà Cung cấp |
| 4 | txtSDT | JTextField | Text : Số Điện Thoại |
| 5 | lblSDT | JLabel | Text: Số điện thoại |
| 6 | txtDiaChi | JTextField |  |
| 7 | lblTenNCC | JLabel |  |
| 8 | txtEmail | JTextField |  |
| 9 | lblDiaChi | JLabel |  |
| 10 | txtGhiChu | JTextArea |  |
| 11 | btnInsert | JButton | Text:Thêm |
| 12 | btnUpdate | JButton | Text:Sửa |
| 13 | btnDelete | JButton | Text:Xóa |
| 14 | btnClear | JButton | Text:Mới |
| 15 | btnFirst | JButton | Text: |< |
| 16 | btnPrev | JButton | Text: << |
| 17 | btnNext | JButton | Text: >> |
| 18 | btnLast | JButton | Text: >| |
| 19 | lblDanhSachNhaCungCap | JLabel | Text : Danh sách nhà cung cấp |
| 20 | lblTimKiem | JLabel |  |
| 21 | txtTimKiem | JTextField |  |
| 22 | tblDanhSachNhaCungCap | JTable |  |

#### 2.3.3.7 Quản Lý Sản Phẩm

Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | txtMaSP | JTextField | Text: Mã nhà cung cấp |
| 2 | txtTenSP | JTextField | Text: Tên nhà cung cấp |
| 3 | lblMaSP | JLabel | Text: Mã Nhà Cung cấp |
| 4 | txtSoLuong | JTextField | Text : Số Điện Thoại |
| 5 | lblTenSP | JLabel | Text: Số điện thoại |
| 6 | txtDonGia | JTextField |  |
| 7 | lblSoLuong | JLabel |  |
| 8 | txtLoaiSP | JTextField |  |
| 9 | lblDonGia | JLabel |  |
| 10 | txtNhaCC | JTextArea |  |
| 11 | lblLoaiSP | JLabel |  |
| 12 | lblMoTa | JLabel |  |
| 13 | txtNhaCC | JTextField |  |
| 14 | txtMoTa | JTextField |  |
| 15 | lblHinh | JLabel | Text: Hình logo |
| 16 | lblHinh | JLabel | Text: |
| 17 | btnInsert | JButton | Text:Thêm |
| 18 | btnUpdate | JButton | Text:Sửa |
| 19 | btnDelete | JButton | Text:Xóa |
| 20 | btnClear | JButton | Text:Mới |
| 21 | btnFirst | JButton | Text: |< |
| 22 | btnPrev | JButton | Text: << |
| 23 | btnNext | JButton | Text: >> |
| 24 | btnLast | JButton | Text: >| |
| 25 | lblHinhMousePressed | JLabel | if (evt.getClickCount() == 2) {  chonAnh();} |
| 26 | lblDanhSachSanPham | JLabel | Text : Danh sách sản phẩm |
| 27 | lblTimKiem | JLabel |  |
| 28 | txtTimKiem | JTextField |  |
| 29 | tblDanhSachSanPham | JTable |  |
| 30 | TabLoaiSP | JTabbedPane |  |
| 31 | LSP,SP | JToolBar | Text : Sản Phẩm  Text : Loại Sản Phẩm |

#### 2.3.3.8 Quản Lý Loại Sản Phẩm



|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | lblTenLoaiSP | JLabel | Text: Tên Loại Sản Phẩm |
| 2 | txtTenLoaiSP | JTextField |  |
| 3 | lblMoTa | JLabel | Text : Mô Tả |
| 4 | tenMoTa | JTextField |  |
| 5 | lblDanhSachLoaiSP | JLabel | Text : Danh sách Loại Sản Phẩm |
| 6 | lblTimKiem | JLabel |  |
| 7 | txtTimKiem | JTextField |  |
| 8 | tblDanhSachLoaiSP | JTable |  |
| 9 | btnInsert | JButton | Text:Thêm |
| 10 | btnUpdate | JButton | Text:Sửa |
| 11 | btnDelete | JButton | Text:Xóa |
| 12 | btnClear | JButton | Text:Mới |

#### 2.3.3.9 Quản lý hóa đơn

#### 

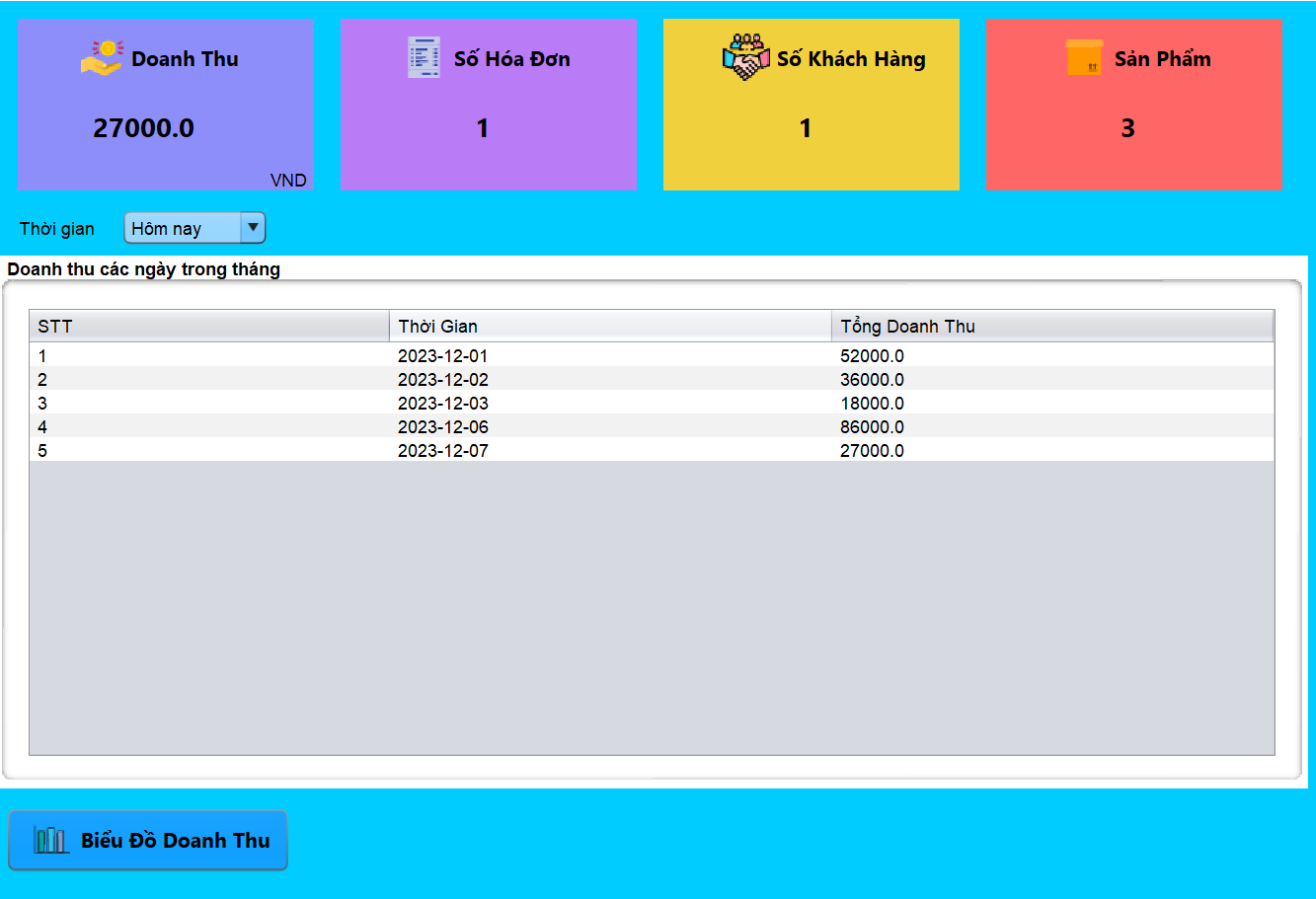
#### 

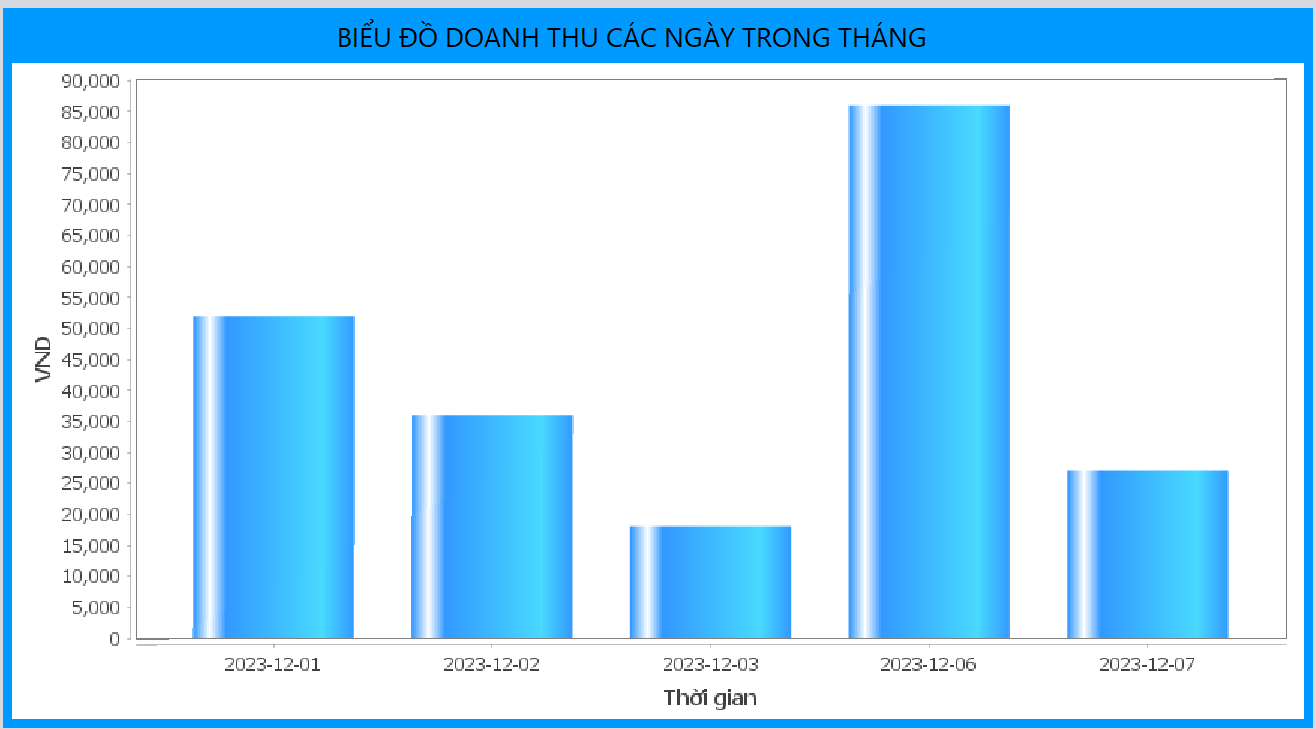
Các thành phần giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | btnTimKiem | JButton | Text: Tìm kiếm |
| 2 | txtTimKiem | JTextField |  |
| 5 | lblDanhSachHoaDon | JLabel | Text : Danh sách Hoá Đơn |
| 6 | PanelDs | JPanel |  |
| 8 | tblDanhSachHoaDon | JTable |  |
| 9 | btnThemHoaDon | JButton | Text:Thêm Hóa Đơn |
| 10 | btnCapNhatHoaDon | JButton | Text:Cập nhật hóa đơn |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | lblChiTietHoaDon | JLabel | Text : Chi tiết hóa đơn |
| 2 | Pn1 | JPanel |  |
| 3 | tblChiTiet | JTable |  |
| 4 | lblSanPham | JLabel | Text: Sản Phẩm |
| 5 | tblSanPham | JTable |  |
| 6 | lblTaoHoaDon | JLable | Text :Tạo hóa đơn |
| 7 | lblMaHD | JLable | Text : Mã Hóa Đơn |
| 8 | lblDienThoai | JLable | Text : Điện Thoại |
| 9 | lblNgayMua | JLable | Text : Ngày Mua |
| 10 | lblTongTienHang | JLable | Text : Tổng Tiền Hàng |
| 11 | lblTienThanhToan | JLable | Text : Tiền Thanh Toán |
| 12 | lblTienThua | JLable | Text : Tiền Thừa |
| 13 | txtMaHD | JTextField | Text : Mã KH |
| 14 | txtDienThoai | JTextField | Text : Điện Thoại |
| 15 | txtTenKhachHang | JTextField | Text : Tên Khách Hàng |
| 16 | txtNgayMua | JTextField | Text : Ngày Mua |
| 17 | txtTongTienHang | JTextField | Text : Tổng Tiền Hàng |
| 18 | txtTienThanhToan | JTextField | Text : Tiền Thanh Toán |
| 19 | txtTienThua | JTextField | Text : Tiền Thừa |
| 20 | btnTaoHoaDon |  | Text : Tạo Hóa Đơn |
| 21 | btnThanhToan |  | Text : Thanh Toán |
| 22 | btnHuyDon |  | Text : Hủy Đơn |
| 23 | btnTaoMoi |  | Text : Tạo Mới |

#### 2.3.3.10 Tổng hợp thống kê





Các thành phẩn giao diện

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Component** | **Kiểu** | **Thuộc tính** |
| 1 | lblDoanhThu | JLabel |  |
| 2 | lblHoaDon | Jlabel |  |
| 5 | lblKhachHang | JLabel |  |
| 6 | lblSanPham | JPanel |  |
| 8 | cboThoiGian | JCombobox |  |
| 9 | lblThoiGian | JPanel | Text:Thời gian |
| 10 | tblDoanhThu | JTable |  |
| 11 | btnBieuDo | JButton | Text: Biểu đồ doanh thu |
| 12 | lblBieuDo | JLabel | Text:Biều đồ doanh thu các ngày trong thánh |
| 13 | pnlDoanhThu | JPanle |  |

# 3. THỰC HIỆN VIẾT MÃ

## 3.1 VIẾT MÃ TẠO CSDL

### 3.1.1 Tạo CSDL

|  |
| --- |
| 1. create database QLBN 2. go 3. use QLBN 4. --Tạo bảng nhân viên 5. create table NhanVien( 6. MaNV nvarchar(50) not null primary key, 7. MatKhau nvarchar(50) not null, 8. TenNV nvarchar(50) not null, 9. VaiTro bit not null, 10. TrangThai bit not null 11. ) 12. go 13. --Tạo bảng khách hàng 14. create table KhachHang( 15. MaKH nvarchar(50) not null primary key, 16. TenKH nvarchar(50) not null, 17. NgaySinh date not null, 18. DienThoai nvarchar(50) not null, 19. Email nvarchar(50) not null, 20. NgayDK date not null, 21. GhiChu nvarchar(max) 22. ) 23. go 24. --Tạo bảng nhà cung cấp 25. create table NhaCungCap( 26. MaNCC nvarchar(50) not null primary key, 27. TenNCC nvarchar(50) not null, 28. DienThoai nvarchar(50) not null, 29. DiaChi nvarchar(255) not null, 30. Email nvarchar(50) not null, 31. GhiChu nvarchar(max) 32. ) 33. go 34. -- Tạo bảng loại sản phẩm 35. create table LoaiSanPham( 36. MaLSP int identity(1,1) not null primary key, 37. TenLSP nvarchar(50) not null, 38. MoTa nvarchar(max) 39. ) 40. go 41. --Tạo bảng sản phẩm 42. create table SanPham( 43. MaSP nvarchar(50) not null primary key, 44. TenSP nvarchar(50) not null, 45. SoLuong int not null, 46. DonGia float not null, 47. HinhAnh nvarchar(50) not null, 48. MoTa nvarchar(max), 49. MaLSP int not null, 50. MaNCC nvarchar(50) not null, 51. MaNV nvarchar(50) not null, 52. constraint fk\_SanPhamLSP foreign key(MaLSP) references LoaiSanPham(MaLSP), 53. constraint fk\_SanPhamNCC foreign key(MaNCC) references NhaCungCap(MaNCC), 54. constraint fk\_SanPhamNV foreign key(MaNV) references NhanVien(MaNV) 55. ) 56. go 57. --Tạo bảng hóa đơn 58. create table HoaDon( 59. MaHD int identity(1,1) not null primary key, 60. NgayMua date not null, 61. TrangThai bit not null, 62. MaKH nvarchar(50) not null, 63. MaNV nvarchar(50) not null, 64. constraint fk\_HoaDonKH foreign key(MaKH) references KhachHang(MaKH), 65. constraint fk\_HoaDonNV foreign key(MaNV) references NhanVien(MaNV) 66. ) 67. go 68. --Tạo bảng hóa đơn chi tiết 69. create table HoaDonChiTiet( 70. MaHDCT int identity(1,1) not null primary key, 71. SoLuong int not null, 72. MaHD int not null, 73. MaSP nvarchar(50) not null, 74. constraint fk\_HoaDonChiTietHD foreign key(MaHD) references HoaDon(MaHD), 75. constraint fk\_HoaDonChiTietSP foreign key(MaSP) references SanPham(MaSP) 76. ) 77. go 78. -- Tạo danh sách hóa đơn 79. create proc sp\_DanhSachHD @tenKHCanTim nvarchar(50) 80. as 81. begin 82. select hd.MaHD, nv.TenNV, kh.TenKH, hd.NgayMua, hd.TrangThai from HoaDon hd 83. inner join NhanVien nv on hd.MaNV = nv.MaNV 84. inner join KhachHang kh on kh.MaKH = hd.MaKH 85. where kh.TenKH like @tenKHCanTim 86. end 87. go |

### 3.1.2 SQL truy vấn và thao tác bảng

#### 3.1.2.1 NhanVien

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

#### 3.1.2.3 SanPham

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO SanPham (MaSP, TenSP, SoLuong, DonGia,HinhAnh,MoTa,MaLSP,MaNCC,MaNV) VALUES (?,?,?,?,?,?,?) |
| Cập nhật theo mã | UPDATE SanPham SET TenSP=?, SoLuong=?, DonGia=?, HinhAnh=?, MoTa=?, MaLSP=? ,MaNCC=?, MaNV=? WHERE MaSP=? |
| Xóa theo mã | DELETE FROM SanPham WHERE MaSP=? |
| Truy vấn tất cả | SELECT \* FROM SanPham |
| Truy vấn theo mã | SELECT \* FROM SanPham WHERE MaSP=? |
| Cập nhật số lương theo mã | UPDATE SanPham SET SoLuong=? WHERE MaSP=? |

3.1.2.4 LoaiSanPham

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO LoaiSanPham (TenLSP,MoTa) VALUES (?,?) |
| Cập nhật theo mã | UPDATE LoaiSanPham SET TenLSP=?, MoTa=? WHERE MaLSP=? |
| Xóa theo mã | DELETE FROM LoaiSanPham WHERE MaSP=? |
| Truy vấn tất cả | SELECT \* FROM LoaiSanPham |
| Truy vấn theo mã | SELECT \* FROM LoaiSanPham WHERE MaLSP=? |

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO KhachHang (MaKH, TenKH, NgaySinh, DienThoai, Email,NgayDK,GhiChu) VALUES (?, ?, ?, ?,?,?,?) |
| Cập nhật theo mã | UPDATE KhachHang SET TenKH=?, NgaySinh=?, DienThoai=?, Email=?, NgayDK=?, GhiChu=? WHERE MaKH=? |
| Xóa theo mã | DELETE FROM KhachHang WHERE MaKH=? |
| Truy vấn tất cả | SELECT \* FROM KhachHang |
| Truy vấn theo mã | SELECT \* FROM KhachHang WHERE MaKH=? |

#### 3.1.2.5 KhachHang

#### 3.1.2.6 NhaCungCap

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO NhaCungCap (MaNCC, TenNCC, DienThoai, DiaChi,Email,GhiChu) VALUES (?, ?, ?, ?,?,?) |
| Cập nhật theo mã | UPDATE NhaCungCap SET TenNCC=?, DienThoai=?, DiaChi=?, Email=?, GhiChu=? WHERE MaNCC=? |
| Xóa theo mã | DELETE FROM NhaCungCap WHERE MaNCC=? |
| Truy vấn tất cả | SELECT \* FROM NhaCungCap |
| Truy vấn theo mã | SELECT \* FROM NhaCungCap WHERE MaNCC=? |

#### 3.1.2.7 HoaDon

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO HoaDon (NgayMua, TrangThai,MaKH,MaNV) VALUES (?, ?, ?,?) |
| Cập nhật theo mã | UPDATE HoaDon SET NgayMua=?, TrangThai=?, MaKH=?, MaNV=? WHERE MaHD=? |
| Xóa theo mã | DELETE FROM HoaDon WHERE MaHD=? |
| Truy vấn tất cả | SELECT \* FROM HoaDon |
| Truy vấn theo mã | SELECT \* FROM HoaDon WHERE MaHD=? |
| Câp nhật trạng thái theo mã | UPDATE HoaDon SET TrangThai=? WHERE MaHD=? |

#### 3.1.2.8 HoaDonChiTiet

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO HoaDonChiTiet (SoLuong, MaHD, MaSP) VALUES (?, ?, ?) |
| Cập nhật theo mã | UPDATE HoaDonChiTiet SET SoLuong=?, MaHD=?, MaSP=? WHERE MaHDCT=? |
| Xóa theo mã | DELETE FROM HoaDonChiTiet WHERE MaHDCT=? |
| Truy vấn tất cả | SELECT \* FROM HoaDonChiTiet |
| Truy vấn theo mã | SELECT \* FROM HoaDonChiTiet WHERE MaHDCT=? |
| Truy vấn theo mã hóa đơn | SELECT \* FROM HoaDonChiTiet WHERE MaHD=? |
| Cập nhật số lương | UPDATE HoaDonChiTiet SET SoLuong=? WHERE MaHDCT=? |
| Truy vấn theo mã hóa đơn, sản phẩm | select hdct.MaHDCT, hdct.SoLuong, hdct.MaHD, hdct.MaSP from HoaDon hd inner join HoaDonChiTiet hdct on hd.MaHD = hdct.MaHDCT where hdct.MaHD = ? and hdct.MaSP = ? |

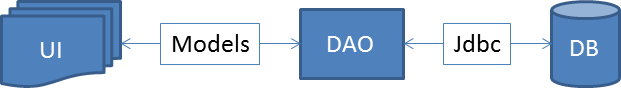
### 

### Các thủ tục lưu tổng hợp thống kê

|  |
| --- |
| -- Thống kê doanh thu các ngày trong tháng  create proc sp\_ThongKeDoanhThuNgay  as  begin  SELECT NgayMua, SUM(HoaDonChiTiet.SoLuong \* DonGia) AS DoanhThu  FROM HoaDonChiTiet  INNER JOIN HoaDon ON HoaDonChiTiet.MaHD = HoaDon.MaHD  INNER JOIN SanPham ON HoaDonChiTiet.MaSP = SanPham.MaSP  WHERE MONTH(NgayMua) = MONTH(GETDATE()) and HoaDon.TrangThai = 1  GROUP BY NgayMua  end  go  -- Thống kê lượng khách hàng theo năm  create proc sp\_LuongSanPhamNam @ngay Date  as  begin  Select  SUM(SoLuong) as LuongSanPham  from  HoaDon hd  inner join HoaDonChiTiet hdct  on hd.MaHD = hdct.MaHD  where  YEAR(hd.NgayMua) = YEAR(@ngay)  end  go  -- Thống kê lượng khách hàng theo thang  create proc sp\_LuongSanPhamThang @ngay Date  as  begin  Select  SUM(SoLuong) as LuongSanPham  from  HoaDon hd  inner join HoaDonChiTiet hdct  on hd.MaHD = hdct.MaHD  where  MONTH(hd.NgayMua) = MONTH(@ngay) and YEAR(hd.NgayMua) = YEAR(@ngay)  end  go  -- Thống kê lượng khách hàng theo năm  create proc sp\_LuongSanPhamNgay @ngay Date  as  begin  Select  SUM(SoLuong) as LuongSanPham  from  HoaDon hd  inner join HoaDonChiTiet hdct  on hd.MaHD = hdct.MaHD  where  hd.NgayMua = @ngay  end  go  -- Thống kê lượng khách hàng theo năm  create proc sp\_LuongKhachTheoNam @ngay Date  as  begin  SELECT  COUNT(DISTINCT MaKH) AS SoKhachHang  FROM  HoaDon  WHERE  YEAR(NgayMua) = YEAR(@ngay)  end  go  -- Thống kê lượng khách hàng theo tháng  create proc sp\_LuongKhachTheoThang @ngay Date  as  begin  SELECT  COUNT(DISTINCT HoaDon.MaKH) as SoKhachHang  FROM  HoaDon  WHERE  MONTH(HoaDon.NgayMua) = MONTH(@ngay) and YEAR(HoaDon.NgayMua) = YEAR(@ngay)  end  go  -- Thống kê lượng khách hàng theo ngày  create proc sp\_LuongKhachTheoNgay @ngay Date  as  begin  SELECT  COUNT(DISTINCT HoaDon.MaKH) as SoKhachHang  FROM  HoaDon  WHERE  HoaDon.NgayMua = @ngay  end  go  -- Thống kê lượng hóa đơn theo năm  create proc sp\_LuongHoaDonTheoNam @ngay Date  as  begin  SELECT  COUNT(HoaDon.MaHD) as SoHoaDon  FROM  HoaDon  WHERE  Year(HoaDon.NgayMua) = YEAR(@ngay)  GROUP BY  Year(HoaDon.NgayMua)  end  go  -- Thống kê lượng hóa đơn theo tháng  create proc sp\_LuongHoaDonTheoThang @ngay Date  as  begin  SELECT  COUNT(HoaDon.MaHD) as SoHoaDon  FROM  HoaDon  WHERE  MONTH(HoaDon.NgayMua) = MONTH(@ngay) and YEAR(HoaDon.NgayMua) = YEAR(@ngay)  GROUP BY  MONTH(HoaDon.NgayMua)  end  go  -- Thống kê lượng hóa đơn theo ngày  create proc sp\_LuongHoaDonTheoNgay @ngay Date  as  begin  SELECT  COUNT(HoaDon.MaHD) as SoHoaDon  FROM  HoaDon  WHERE  HoaDon.NgayMua = @ngay  GROUP BY  DAY(HoaDon.NgayMua)  end  go  -- Thống kê doanh thu theo ngày  create proc sp\_DoanhThuTheoNgay @ngay Date  as  begin  SELECT  SUM(HoaDonChiTiet.SoLuong \* SanPham.DonGia) AS DoanhThu  FROM  HoaDon  INNER JOIN HoaDonChiTiet ON HoaDon.MaHD = HoaDonChiTiet.MaHD  INNER JOIN SanPham ON HoaDonChiTiet.MaSP = SanPham.MaSP  WHERE  HoaDon.NgayMua = @ngay  GROUP BY  Day(HoaDon.NgayMua)  end  go  -- Thống kê doanh thu theo tháng  create proc sp\_DoanhThuTheoThang @ngay date  as  begin  SELECT  SUM(HoaDonChiTiet.SoLuong \* SanPham.DonGia) AS DoanhThu  FROM  HoaDon  INNER JOIN HoaDonChiTiet ON HoaDon.MaHD = HoaDonChiTiet.MaHD  INNER JOIN SanPham ON HoaDonChiTiet.MaSP = SanPham.MaSP  WHERE  Month(HoaDon.NgayMua) = MONTH(@ngay) and Year(HoaDon.NgayMua) = Year(@ngay) and HoaDon.TrangThai = 1  GROUP BY  Month(HoaDon.NgayMua)  end  go  -- Thống kê doanh thu theo năm  create proc sp\_DoanhThuTheoNam @nam date  as  begin  SELECT  SUM(HoaDonChiTiet.SoLuong \* SanPham.DonGia) AS DoanhThu  FROM  HoaDon  INNER JOIN HoaDonChiTiet ON HoaDon.MaHD = HoaDonChiTiet.MaHD  INNER JOIN SanPham ON HoaDonChiTiet.MaSP = SanPham.MaSP  WHERE  YEAR(HoaDon.NgayMua) = YEAR(@nam) and HoaDon.TrangThai = 1  GROUP BY  YEAR(HoaDon.NgayMua)  end |

## 3.2 LẬP TRÌNH JDBC

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



Trong đó:

##### UI: các thành phần giao diện

* + - DB: cơ sở dữ liệu

##### Jdbc: là lớp tiện ích cung cấp các hàm tiện ích làm việc với CSDL thông qua câu lệnh sql hoặc lời gọi thủ tục lưu

* + - Model: là các lớp mô tả dữ liệu theo cấu trúc các bảng trong CSDL

##### DAO: là các lớp thao tác và truy vấn dữ liệu. Nó có nhiệm vụ chuyển đổi Model sang SQL và ngược lại.

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

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

### 3.2.1 Lớp hỗ trợ

Các lớp hỗ trợ chứa các phương thức tiện ích giúp chúng ta thực hiện các xử lý trong ứng dụng đơn giản hơn, ngắn gọn hơn, dễ hiểu hơn. Trong phần này chúng ta xây dựng 2 lớp là XDate và Jdbc như sau.

#### 3.2.1.1 Lớp tiện ích XDate

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

|  |
| --- |
| static SimpleDateFormat formater = new SimpleDateFormat();  // Chuyển đổi String sang Date  // date là String cần chuyển đổi  // pattern là định dạng thời gian  // @return date là kết quả  public static Date toDate(String date, String pattern) {  try {  formater.applyPattern(pattern);  return formater.parse(date);  } catch (ParseException e) {  throw new RuntimeException(e);  }  }  // Chuyển đổi Date sang String  // date là Date cần chuyển đổi  // pattern là định dạng thời gian  // @return String là kết quả  public static String toString(Date date, String pattern) {  formater.applyPattern(pattern);  return formater.format(date);  }  public static Date now() {  return new Date();  }  // Bổ sung số ngày vào thời gian  //  // date thời gian hiện có  // days số ngày cần bổ sung vào date  // @return Date kết quả  public static Date addDays(Date date, int days) {  date.setTime(date.getTime() + days \* 24 \* 60 \* 60 \* 1000);  return date;  }  public static Date add(int days) {  Date now = XDate.now();  now.setTime(now.getTime() + days \* 24 \* 60 \* 60 \* 1000);  return now;  } |

#### 3.2.1.2 Lớp tiện ích JdbcHelper

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

|  |
| --- |
| package duan.qlbn.utils;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  public class JdbcHelper {  public static String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  public static String dbulr = "jdbc:sqlserver://localhost:1433; databaseName=QLBN";  public static String user = "sa";  public static String password = "123";      static{  try {  Class.forName(driver);  } catch (ClassNotFoundException e) {  throw new RuntimeException(e);  }  }    public static PreparedStatement getStmt(String sql, Object... args) throws SQLException{  Connection connection = DriverManager.getConnection(dbulr,user,password);  PreparedStatement pstmt = null;  if (sql.trim().startsWith("{")) {  pstmt = connection.prepareCall(sql);  }else{  pstmt = connection.prepareStatement(sql);  }  for (int i = 0; i < args.length; i++) {  pstmt.setObject(i+1, args[i]);  }  return pstmt;  }    public static int update(String sql, Object... args){  try {  PreparedStatement stmt = getStmt(sql, args);  try {  return stmt.executeUpdate();  } finally {  stmt.getConnection().close();  }  } catch (SQLException e) {  throw new RuntimeException(e);  }  }    public static ResultSet query(String sql, Object... args){  try {  PreparedStatement stmt = getStmt(sql, args);  return stmt.executeQuery();  } catch (SQLException e) {  throw new RuntimeException(e);  }  }    public static Object value(String sql, Object... args){  try {  ResultSet rs = query(sql, args);  if (rs.next()) {  return rs.getObject(0);  }  rs.getStatement().getConnection().close();  } catch (Exception e) {  throw new RuntimeException(e);  }  return null;  }  } |

#### 3.2.1.3 Lớp Auth

|  |
| --- |
| / Đối tượng này chứa thông tin người dùng sau khi đăng nhập  public static NhanVien user = null;  // Xóa thông tin người sử dụng sau khi đăng xuất  public static void clear() {  Auth.user = null;  }  // Kiểm tra xem đăng nhập hay chưa  // @return về đăng nhập hay chưa  public static boolean isLogin() {  return Auth.user != null;  }  // Kiểm tra vai trò quản lý hay nhân viên  public static boolean isManager() {  return Auth.isLogin() && user.isVaiTro();  } |

#### 3.2.1.4 Lớp XImage

|  |
| --- |
| public static Image getAppIcon(){  URL url = XImage.class.getResource("/duan/qlbn/icon/fpt.png");  return new ImageIcon(url).getImage();  }  public static boolean save(File src){  File dst = new File("src\\duan\\qlbn\\hinhanh", src.getName());  if (!dst.getParentFile().exists()) {  dst.getParentFile().mkdirs(); //Tạo thư mục  }  try {  Path from = Paths.get(src.getAbsolutePath());  Path to = Paths.get(dst.getAbsolutePath());  Files.copy(from, to, StandardCopyOption.REPLACE\_EXISTING);  return true;  } catch (Exception e) {  return false;  }  }  public static ImageIcon read(String fileName){  File path = new File("src\\duan\\qlbn\\hinhanh",fileName);  return new ImageIcon(path.getAbsolutePath());  } |

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

Để tránh các sai sót về kiểu dữ liệu khi làm việc với CSDL, chúng ta cần phải xây dựng các lớp mô tả dữ liệu với kiểu phù hợp. Điều này giúp làm việc với CSDL chặt chẽ hơn, dễ hơn.

#### 3.2.2.1 NhanVien

|  |
| --- |
| private String maNV;  private String matKhau;  private String email;  private String hoTen;  private boolean vaiTro;  private boolean trangThai;  public NhanVien() {  }  public NhanVien(String maNV, String matKhau, String email, String hoTen, boolean vaiTro, boolean trangThai) {  this.maNV = maNV;  this.matKhau = matKhau;  this.email = email;  this.hoTen = hoTen;  this.vaiTro = vaiTro;  this.trangThai = trangThai;  }  public String getMaNV() {  return maNV;  }  public void setMaNV(String maNV) {  this.maNV = maNV;  }  public String getMatKhau() {  return matKhau;  }  public void setMatKhau(String matKhau) {  this.matKhau = matKhau;  }  public String getEmail() {  return email;  }  public void setEmail(String email) {  this.email = email;  }  public String getHoTen() {  return hoTen;  }  public void setHoTen(String hoTen) {  this.hoTen = hoTen;  }  public boolean isVaiTro() {  return vaiTro;  }  public void setVaiTro(boolean vaiTro) {  this.vaiTro = vaiTro;  }  public boolean isTrangThai() {  return trangThai;  }  public void setTrangThai(boolean trangThai) {  this.trangThai = trangThai;  } |

#### 3.2.2.2 SanPham

|  |
| --- |
| private String maSP;  private String tenSP;  private int soLuong;  private double donGia;  private String hinhAnh;  private String moTa;  private int maLSP;  private String maNCC;  private String maNV;  public SanPham() {  }  public SanPham(String maSP, String tenSP, int soLuong, double donGia, String hinhAnh, String moTa, int maLSP, String maNCC, String maNV) {  this.maSP = maSP;  this.tenSP = tenSP;  this.soLuong = soLuong;  this.donGia = donGia;  this.hinhAnh = hinhAnh;  this.moTa = moTa;  this.maLSP = maLSP;  this.maNCC = maNCC;  this.maNV = maNV;  }  public String getMaSP() {  return maSP;  }  public void setMaSP(String maSP) {  this.maSP = maSP;  }  public String getTenSP() {  return tenSP;  }  public void setTenSP(String tenSP) {  this.tenSP = tenSP;  }  public int getSoLuong() {  return soLuong;  }  public void setSoLuong(int soLuong) {  this.soLuong = soLuong;  }  public double getDonGia() {  return donGia;  }  public void setDonGia(double donGia) {  this.donGia = donGia;  }  public String getHinhAnh() {  return hinhAnh;  }  public void setHinhAnh(String hinhAnh) {  this.hinhAnh = hinhAnh;  }  public String getMoTa() {  return moTa;  }  public void setMoTa(String moTa) {  this.moTa = moTa;  }  public int getMaLSP() {  return maLSP;  }  public void setMaLSP(int maLSP) {  this.maLSP = maLSP;  }  public String getMaNCC() {  return maNCC;  }  public void setMaNCC(String maNCC) {  this.maNCC = maNCC;  }  public String getMaNV() {  return maNV;  }  public void setMaNV(String maNV) {  this.maNV = maNV;  } |

#### 2.2.2.3 LoaiSanPham

|  |
| --- |
| private int MaLSP;private String tenLSP;private String moTa;public LoaiSanPham() {}public LoaiSanPham(int MaLSP, String tenLSP, String moTa) {this.MaLSP = MaLSP;this.tenLSP = tenLSP;this.moTa = moTa;}public int getMaLSP() {return MaLSP;}public void setMaLSP(int MaLSP) {this.MaLSP = MaLSP;}public String getTenLSP() {return tenLSP;}public void setTenLSP(String tenLSP) {this.tenLSP = tenLSP;}public String getMoTa() {return moTa;}public void setMoTa(String moTa) {this.moTa = moTa;} |

#### 3.2.2.4 KhachHang

|  |
| --- |
| private String maKH;  private String tenKH;  private Date ngaySinh;  private String dienThoai;  private String email;  private String ghiChu;  private Date ngayDangKy;  public KhachHang() {  }  public KhachHang(String maKH, String tenKH, Date ngaySinh, String dienThoai, String email, String ghiChu, Date ngayDangKy) {  this.maKH = maKH;  this.tenKH = tenKH;  this.ngaySinh = ngaySinh;  this.dienThoai = dienThoai;  this.email = email;  this.ghiChu = ghiChu;  this.ngayDangKy = ngayDangKy;  }  public String getMaKH() {  return maKH;  }  public void setMaKH(String maKH) {  this.maKH = maKH;  }  public String getTenKH() {  return tenKH;  }  public void setTenKH(String tenKH) {  this.tenKH = tenKH;  }  public Date getNgaySinh() {  return ngaySinh;  }  public void setNgaySinh(Date ngaySinh) {  this.ngaySinh = ngaySinh;  }  public String getDienThoai() {  return dienThoai;  }  public void setDienThoai(String dienThoai) {  this.dienThoai = dienThoai;  }  public String getEmail() {  return email;  }  public void setEmail(String email) {  this.email = email;  }  public String getGhiChu() {  return ghiChu;  }  public void setGhiChu(String ghiChu) {  this.ghiChu = ghiChu;  }  public Date getNgayDangKy() {  return ngayDangKy;  }  public void setNgayDangKy(Date ngayDangKy) {  this.ngayDangKy = ngayDangKy;  } |

#### 3.2.2.5 NhaCungCap

|  |
| --- |
| private String maNCC;  private String tenNCC;  private String dienThoai;  private String diaChi;  private String email;  private String ghiChu;  public NhaCungCap() {  }  public NhaCungCap(String maNCC, String tenNCC, String dienThoai, String diaChi, String email, String ghiChu) {  this.maNCC = maNCC;  this.tenNCC = tenNCC;  this.dienThoai = dienThoai;  this.diaChi = diaChi;  this.email = email;  this.ghiChu = ghiChu;  }  public String getMaNCC() {  return maNCC;  }  public void setMaNCC(String maNCC) {  this.maNCC = maNCC;  }  public String getTenNCC() {  return tenNCC;  }  public void setTenNCC(String tenNCC) {  this.tenNCC = tenNCC;  }  public String getDienThoai() {  return dienThoai;  }  public void setDienThoai(String dienThoai) {  this.dienThoai = dienThoai;  }  public String getDiaChi() {  return diaChi;  }  public void setDiaChi(String diaChi) {  this.diaChi = diaChi;  }  public String getEmail() {  return email;  }  public void setEmail(String email) {  this.email = email;  }  public String getGhiChu() {  return ghiChu;  }  public void setGhiChu(String ghiChu) {  this.ghiChu = ghiChu;  } |

#### 3.2.2.6 HoaDon

|  |
| --- |
| private int maHD;private Date ngayMua;private boolean trangThai;private String maKH;private String maNV;public HoaDon() {}public HoaDon(int maHD, Date ngayMua, boolean trangThai, String maKH, String maNV) {this.maHD = maHD;this.ngayMua = ngayMua;this.trangThai = trangThai;this.maKH = maKH;this.maNV = maNV;}public int getMaHD() {return maHD;}public void setMaHD(int maHD) {this.maHD = maHD;}public Date getNgayMua() {return ngayMua;}public void setNgayMua(Date ngayMua) {this.ngayMua = ngayMua;}public boolean isTrangThai() {return trangThai;}public void setTrangThai(boolean trangThai) {this.trangThai = trangThai;}public String getMaKH() {return maKH;}public void setMaKH(String maKH) {this.maKH = maKH;}public String getMaNV() {return maNV;}public void setMaNV(String maNV) {this.maNV = maNV;} |

#### 3.2.2.7 HoaDonChiTiet

|  |
| --- |
| private int maHDCT;  private int soLuong;  private int maHD;  private String maSP;  public HoaDonChiTiet() {  }  public HoaDonChiTiet(int maHDCT, int soLuong, int maHD, String maSP) {  this.maHDCT = maHDCT;  this.soLuong = soLuong;  this.maHD = maHD;  this.maSP = maSP;  }  public int getMaHDCT() {  return maHDCT;  }  public void setMaHDCT(int maHDCT) {  this.maHDCT = maHDCT;  }  public int getSoLuong() {  return soLuong;  }  public void setSoLuong(int soLuong) {  this.soLuong = soLuong;  }  public int getMaHD() {  return maHD;  }  public void setMaHD(int maHD) {  this.maHD = maHD;  }  public String getMaSP() {  return maSP;  }  public void setMaSP(String maSP) {  this.maSP = maSP;  } |

#### 3.2.2.8 MyCombobox

|  |
| --- |
| Object value; //chứa mã loại  Object text; //chứa tên loại  public MyCombobox(Object value, Object text) {  this.value = value;  this.text = text;  }    @Override  public String toString(){  return text.toString();  }    //Hàm lấy value kiểu int  public int MaInt(){  return Integer.parseInt(value.toString());  }  //Hàm lấy value kiểu String  public String MaString(){  return value.toString();  } |

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

##### DAO (Data Access Object) là các lớp giúp giúp làm việc với các bảng trong CSDL. Mỗi lớp được định nghĩa một cách đầy đủ nhất về các thao tác và truy vấn dữ liệu sẽ được sử dụng trong ứng dụng. DAO sau đó sẽ được sử dụng trong các điều khiển sự kiện của giao diện để làm việc với CSDL một cách thuận tiện.

Trong phạm vi ứng dụng này chúng ta cần xây dựng mỗi DAO chứa tối thiểu các hành động sau

/\*\*

Thêm mới thực thể vào CSDL

@param entity là thực thể chứa thông tin bản ghi mới

\*/

public void insert(Entity entity){}

/\*\*

Cập nhật thực thể vào CSDL

@param entity là thực thể chứa thông tin bản ghi cần cập nhật

\*/

public void update(Entity entity){}

/\*\*

Xóa bản ghi khỏi CSDL

@param id là mã của bản ghi cần xóa

\*/

public void delete(String id){}

/\*\*

Truy vấn tất cả các các thực thể

@return danh sách các thực thể

\*/

public List<Entity> select(){}

/\*\*

Truy vấn thực thể theo mã

@param id là mã của bản ghi được truy vấn

@return thực thể chứa thông tin của bản ghi

\*/

public Entity findById(String id){}

Sau đây là các lớp DAO cụ thể

#### 3.2.3.1 NhanVienDAO

|  |
| --- |
| final String INSERT\_SQL = "INSERT INTO NhanVien (MaNV, MatKhau,email, TenNV, VaiTro, TrangThai) VALUES (?, ?,?, ?, ?,?)";final String UPDATE\_SQL = "UPDATE NhanVien SET MatKhau=?,email=?, TenNV=?, VaiTro=?, TrangThai=? WHERE MaNV=?";final String DELETE\_SQL = "DELETE FROM NhanVien WHERE MaNV=?";final String SELECT\_ALL\_SQL = "SELECT \* FROM NhanVien";final String SELECT\_BY\_ID\_SQL = "SELECT \* FROM NhanVien WHERE MaNV=?";final String UPDATE\_MatKhau\_SQL = "UPDATE NhanVien SET MatKhau=? WHERE MaNV=?";public List<NhanVien> selectByKeyword(String keyword) {String sql = "SELECT \* FROM NhanVien WHERE TenNV LIKE ?";return selectBySql(sql, "%" + keyword + "%");}@Overridepublic void insert(NhanVien entity) {JdbcHelper.update(INSERT\_SQL, entity.getMaNV(), entity.getMatKhau(), entity.getEmail(), entity.getHoTen(), entity.isVaiTro(), entity.isTrangThai());}@Overridepublic void update(NhanVien entity) {JdbcHelper.update(UPDATE\_SQL, entity.getMatKhau(), entity.getEmail(), entity.getHoTen(), entity.isVaiTro(), entity.isTrangThai(), entity.getMaNV());}public void updateMatKhau(String matKhau, String maNV) {JdbcHelper.update(UPDATE\_MatKhau\_SQL, matKhau, maNV);}@Overridepublic void delete(String id) {JdbcHelper.update(DELETE\_SQL, id);}@Overridepublic List<NhanVien> selectAll() {return selectBySql(SELECT\_ALL\_SQL);}@Overridepublic NhanVien selectByID(String id) {List<NhanVien> list = selectBySql(SELECT\_BY\_ID\_SQL, id);if (list.isEmpty()) {return null;}return list.get(0);}@Overridepublic List<NhanVien> selectBySql(String sql, Object... args) {List<NhanVien> list = new ArrayList<>();try {ResultSet rs = JdbcHelper.query(sql, args);while (rs.next()) {NhanVien entity = new NhanVien();entity.setMaNV(rs.getString("MaNV"));entity.setMatKhau(rs.getString("MatKhau"));entity.setEmail(rs.getString("email"));entity.setHoTen(rs.getString("TenNV"));entity.setVaiTro(rs.getBoolean("VaiTro"));entity.setTrangThai(rs.getBoolean("TrangThai"));list.add(entity);}} catch (Exception e) {throw new RuntimeException(e);}return list;} |

#### 3.2.3.2 SanPhamDAO

|  |
| --- |
| final String INSERT\_SQL = "INSERT INTO SanPham (MaSP, TenSP, SoLuong, DonGia,HinhAnh,MoTa,MaLSP,MaNCC,MaNV) VALUES (?,?,?,?,?,?,?,?,?)";  final String UPDATE\_SQL = "UPDATE SanPham SET TenSP=?, SoLuong=?, DonGia=?, HinhAnh=?, MoTa=?, MaLSP=? ,MaNCC=?, MaNV=? WHERE MaSP=?";  final String DELETE\_SQL = "DELETE FROM SanPham WHERE MaSP=?";  final String SELECT\_ALL\_SQL = "SELECT \* FROM SanPham";  final String SELECT\_BY\_ID\_SQL = "SELECT \* FROM SanPham WHERE MaSP=?";  final String UPDATESoLuongSp\_SQL = "UPDATE SanPham SET SoLuong=? WHERE MaSP=?";  public List<SanPham> selectByKeyword(String keyword) {  String sql = "SELECT \* FROM SanPham WHERE TenSP LIKE ?";  return selectBySql(sql, "%" + keyword + "%");  }  @Override  public void insert(SanPham entity) {  JdbcHelper.update(INSERT\_SQL, entity.getMaSP(), entity.getTenSP(), entity.getSoLuong(), entity.getDonGia(), entity.getHinhAnh(), entity.getMoTa(), entity.getMaLSP(), entity.getMaNCC(), entity.getMaNV());  }  @Override  public void update(SanPham entity) {  JdbcHelper.update(UPDATE\_SQL, entity.getTenSP(), entity.getSoLuong(), entity.getDonGia(), entity.getHinhAnh(), entity.getMoTa(), entity.getMaLSP(), entity.getMaNCC(), entity.getMaNV(), entity.getMaSP());  }  public void updateSoLuongSP(int soLuongSP, String maSP) {  JdbcHelper.update(UPDATESoLuongSp\_SQL, soLuongSP, maSP);  }  @Override  public void delete(String id) {  JdbcHelper.update(DELETE\_SQL, id);  }  @Override  public List<SanPham> selectAll() {  return selectBySql(SELECT\_ALL\_SQL);  }  @Override  public SanPham selectByID(String id) {  List<SanPham> list = selectBySql(SELECT\_BY\_ID\_SQL, id);  if (list.isEmpty()) {  return null;  }  return list.get(0);  }  @Override  public List<SanPham> selectBySql(String sql, Object... args) {  List<SanPham> list = new ArrayList<>();  try {  ResultSet rs = JdbcHelper.query(sql, args);  while (rs.next()) {  SanPham entity = new SanPham();  entity.setMaSP(rs.getString("MaSP"));  entity.setTenSP(rs.getString("TenSP"));  entity.setSoLuong(rs.getInt("SoLuong"));  entity.setDonGia(rs.getDouble("DonGia"));  entity.setHinhAnh(rs.getString("HinhAnh"));  entity.setMoTa(rs.getString("MoTa"));  entity.setMaLSP(rs.getInt("MaLSP"));  entity.setMaNCC(rs.getString("MaNCC"));  entity.setMaNV(rs.getString("MaNV"));  list.add(entity);  }  } catch (Exception e) {  throw new RuntimeException(e);  }  return list;  } |

#### 3.2.3.3 KhachHangDAO

|  |
| --- |
| final String INSERT\_SQL = "INSERT INTO KhachHang (MaKH, TenKH, NgaySinh, DienThoai, Email,NgayDK,GhiChu) VALUES (?, ?, ?, ?,?,?,?)";final String UPDATE\_SQL = "UPDATE KhachHang SET TenKH=?, NgaySinh=?, DienThoai=?, Email=?, NgayDK=?, GhiChu=? WHERE MaKH=?";final String DELETE\_SQL = "DELETE FROM KhachHang WHERE MaKH=?";final String SELECT\_ALL\_SQL = "SELECT \* FROM KhachHang";final String SELECT\_BY\_ID\_SQL = "SELECT \* FROM KhachHang WHERE MaKH=?";public List<KhachHang> selectByKeyword(String keyword) {String sql = "SELECT \* FROM KhachHang WHERE TenKH LIKE ?";return selectBySql(sql, "%" + keyword + "%");}@Overridepublic void insert(KhachHang entity) {JdbcHelper.update(INSERT\_SQL, entity.getMaKH(), entity.getTenKH(), entity.getNgaySinh(), entity.getDienThoai(), entity.getEmail(), entity.getNgayDangKy(), entity.getGhiChu());}@Overridepublic void update(KhachHang entity) {JdbcHelper.update(UPDATE\_SQL, entity.getTenKH(), entity.getNgaySinh(), entity.getDienThoai(), entity.getEmail(), entity.getNgayDangKy(), entity.getGhiChu(), entity.getMaKH());}@Overridepublic void delete(String id) {JdbcHelper.update(DELETE\_SQL, id);}@Overridepublic List<KhachHang> selectAll() {return selectBySql(SELECT\_ALL\_SQL);}@Overridepublic KhachHang selectByID(String id) {List<KhachHang> list = selectBySql(SELECT\_BY\_ID\_SQL, id);if (list.isEmpty()) {return null;}return list.get(0);}@Overridepublic List<KhachHang> selectBySql(String sql, Object... args) {List<KhachHang> list = new ArrayList<>();try {ResultSet rs = JdbcHelper.query(sql, args);while (rs.next()) {KhachHang entity = new KhachHang();entity.setMaKH(rs.getString("MaKH"));entity.setTenKH(rs.getString("TenKH"));entity.setNgaySinh(rs.getDate("NgaySinh"));entity.setDienThoai(rs.getString("DienThoai"));entity.setEmail(rs.getString("Email"));entity.setNgayDangKy(rs.getDate("NgayDK"));entity.setGhiChu(rs.getString("GhiChu"));list.add(entity);}} catch (Exception e) {throw new RuntimeException(e);}return list;} |

#### 3.2.3.4 NhaCungCapDAO

|  |
| --- |
| final String INSERT\_SQL = "INSERT INTO NhaCungCap (MaNCC, TenNCC, DienThoai, DiaChi,Email,GhiChu) VALUES (?, ?, ?, ?,?,?)";  final String UPDATE\_SQL = "UPDATE NhaCungCap SET TenNCC=?, DienThoai=?, DiaChi=?, Email=?, GhiChu=? WHERE MaNCC=?";  final String DELETE\_SQL = "DELETE FROM NhaCungCap WHERE MaNCC=?";  final String SELECT\_ALL\_SQL = "SELECT \* FROM NhaCungCap";  final String SELECT\_BY\_ID\_SQL = "SELECT \* FROM NhaCungCap WHERE MaNCC=?";  public List<NhaCungCap> selectByKeyword(String keyword) {  String sql = "SELECT \* FROM NhaCungCap WHERE TenNCC LIKE ?";  return selectBySql(sql, "%" + keyword + "%");  }  @Override  public void insert(NhaCungCap entity) {  JdbcHelper.update(INSERT\_SQL, entity.getMaNCC(), entity.getTenNCC(), entity.getDienThoai(), entity.getDiaChi(), entity.getEmail(), entity.getGhiChu());  }  @Override  public void update(NhaCungCap entity) {  JdbcHelper.update(UPDATE\_SQL, entity.getTenNCC(), entity.getDienThoai(), entity.getDiaChi(), entity.getEmail(), entity.getGhiChu(), entity.getMaNCC());  }  @Override  public void delete(String id) {  JdbcHelper.update(DELETE\_SQL, id);  }  @Override  public List<NhaCungCap> selectAll() {  return selectBySql(SELECT\_ALL\_SQL);  }  @Override  public NhaCungCap selectByID(String id) {  List<NhaCungCap> list = selectBySql(SELECT\_BY\_ID\_SQL, id);  if (list.isEmpty()) {  return null;  }  return list.get(0);  }  @Override  public List<NhaCungCap> selectBySql(String sql, Object... args) {  List<NhaCungCap> list = new ArrayList<>();  try {  ResultSet rs = JdbcHelper.query(sql, args);  while (rs.next()) {  NhaCungCap entity = new NhaCungCap();  entity.setMaNCC(rs.getString("MaNCC"));  entity.setTenNCC(rs.getString("TenNCC"));  entity.setDienThoai(rs.getString("DienThoai"));  entity.setDiaChi(rs.getString("DiaChi"));  entity.setEmail(rs.getString("Email"));  entity.setGhiChu(rs.getString("GhiChu"));  list.add(entity);  }  } catch (Exception e) {  throw new RuntimeException(e);  }  return list;  } |

#### 3.2.3.5 HoaDonDAO

|  |
| --- |
| final String INSERT\_SQL = "INSERT INTO HoaDon (NgayMua, TrangThai,MaKH,MaNV) VALUES (?, ?, ?,?)";  final String UPDATE\_SQL = "UPDATE HoaDon SET NgayMua=?, TrangThai=?, MaKH=?, MaNV=? WHERE MaHD=?";  final String DELETE\_SQL = "DELETE FROM HoaDon WHERE MaHD=?";  final String SELECT\_ALL\_SQL = "SELECT \* FROM HoaDon";  final String SELECT\_BY\_ID\_SQL = "SELECT \* FROM HoaDon WHERE MaHD=?";  final String UPDATE\_TrangThai\_SQL = "UPDATE HoaDon SET TrangThai=? WHERE MaHD=?";  public List<Object[]> getHoaDon(String tenKH) {  String sql = "{CALL sp\_DanhSachHD(?)}";  String[] cols = {"MaHD", "TenNV", "TenKH", "NgayMua", "TrangThai"};  return getListOfArray(sql, cols, "%"+ tenKH+ "%");  }  @Override  public void insert(HoaDon entity) {  JdbcHelper.update(INSERT\_SQL, entity.getNgayMua(), entity.isTrangThai(), entity.getMaKH(), entity.getMaNV());  }  @Override  public void update(HoaDon entity) {  JdbcHelper.update(UPDATE\_SQL, entity.getNgayMua(), entity.isTrangThai(), entity.getMaKH(), entity.getMaNV(), entity.getMaHD());  }  public void updateTrangThai(boolean trangThai, int maHD) {  JdbcHelper.update(UPDATE\_TrangThai\_SQL, trangThai, maHD);  }  @Override  public void delete(String id) {  JdbcHelper.update(DELETE\_SQL, id);  }  public void deleteByInt(int id) {  JdbcHelper.update(DELETE\_SQL, id);  }  @Override  public List<HoaDon> selectAll() {  return selectBySql(SELECT\_ALL\_SQL);  }  @Override  public HoaDon selectByID(String id) {  List<HoaDon> list = selectBySql(SELECT\_BY\_ID\_SQL, id);  if (list.isEmpty()) {  return null;  }  return list.get(0);  }  public HoaDon selectByIDInt(int id) {  List<HoaDon> list = selectBySql(SELECT\_BY\_ID\_SQL, id);  if (list.isEmpty()) {  return null;  }  return list.get(0);  }  @Override  public List<HoaDon> selectBySql(String sql, Object... args) {  List<HoaDon> list = new ArrayList<>();  try {  ResultSet rs = JdbcHelper.query(sql, args);  while (rs.next()) {  HoaDon entity = new HoaDon();  entity.setMaHD(rs.getInt("MaHD"));  entity.setNgayMua(rs.getDate("NgayMua"));  entity.setTrangThai(rs.getBoolean("TrangThai"));  entity.setMaKH(rs.getString("MaKH"));  entity.setMaNV(rs.getString("MaNV"));  list.add(entity);  }  } catch (Exception e) {  throw new RuntimeException(e);  }  return list;  }  private List<Object[]> getListOfArray(String sql, String[] cols, Object... args) {  try {  List<Object[]> list = new ArrayList<>();  ResultSet rs = JdbcHelper.query(sql, args);  while (rs.next()) {  Object[] vals = new Object[cols.length];  for (int i = 0; i < cols.length; i++) {  vals[i] = rs.getObject(cols[i]);  }  list.add(vals);  }  rs.getStatement().getConnection().close();  return list;  } catch (Exception e) {  throw new RuntimeException(e);  }  } |

#### 3.2.3.6 HoaDonChiTietDAO

|  |
| --- |
| final String INSERT\_SQL = "INSERT INTO HoaDonChiTiet (SoLuong, MaHD, MaSP) VALUES (?, ?, ?)";final String UPDATE\_SQL = "UPDATE HoaDonChiTiet SET SoLuong=?, MaHD=?, MaSP=? WHERE MaHDCT=?";final String DELETE\_SQL = "DELETE FROM HoaDonChiTiet WHERE MaHDCT=?";final String SELECT\_ALL\_SQL = "SELECT \* FROM HoaDonChiTiet";final String SELECT\_BY\_ID\_SQL = "SELECT \* FROM HoaDonChiTiet WHERE MaHDCT=?";final String SELECT\_BY\_MA\_HD\_SQL = "SELECT \* FROM HoaDonChiTiet WHERE MaHD=?";final String UPDATE\_SoLuong\_SQL = "UPDATE HoaDonChiTiet SET SoLuong=? WHERE MaHDCT=?";final String SELECT\_MaHDCT\_BY\_MHD\_MSP = "select hdct.MaHDCT, hdct.SoLuong, hdct.MaHD, hdct.MaSP from HoaDon hd inner join HoaDonChiTiet hdct on hd.MaHD = hdct.MaHDCT where hdct.MaHD = ? and hdct.MaSP = ?";final String SELECT\_ALLHDCTBYMaHD\_SQL = "select \* from HoaDonChiTiet where MaHD = ?";public List<HoaDonChiTiet> selectAllHDCTByMaHD(int maHD) {return selectBySql(SELECT\_ALLHDCTBYMaHD\_SQL, maHD);}public HoaDonChiTiet selectMaHDCT(int maHD, String maSP) {List<HoaDonChiTiet> list = selectBySql(SELECT\_MaHDCT\_BY\_MHD\_MSP, maHD, maSP);if (list.isEmpty()) {return null;}return list.get(0);}public List<HoaDonChiTiet> selectByHoaDon(int maKH) {return selectBySql(SELECT\_BY\_MA\_HD\_SQL, maKH);}@Overridepublic void insert(HoaDonChiTiet entity) {JdbcHelper.update(INSERT\_SQL, entity.getSoLuong(), entity.getMaHD(), entity.getMaSP());}@Overridepublic void update(HoaDonChiTiet entity) {JdbcHelper.update(UPDATE\_SQL, entity.getSoLuong(), entity.getMaHD(), entity.getMaSP(), entity.getMaHDCT());}public void updateSoLuong(int soLuong, int maHDCT) {JdbcHelper.update(UPDATE\_SoLuong\_SQL, soLuong, maHDCT);}@Overridepublic void delete(String id) {JdbcHelper.update(DELETE\_SQL, id);}public void deleteByInt(int id) {JdbcHelper.update(DELETE\_SQL, id);}@Overridepublic List<HoaDonChiTiet> selectAll() {return selectBySql(SELECT\_ALL\_SQL);}@Overridepublic HoaDonChiTiet selectByID(String id) {List<HoaDonChiTiet> list = selectBySql(SELECT\_BY\_ID\_SQL, id);if (list.isEmpty()) {return null;}return list.get(0);}@Overridepublic List<HoaDonChiTiet> selectBySql(String sql, Object... args) {List<HoaDonChiTiet> list = new ArrayList<>();try {ResultSet rs = JdbcHelper.query(sql, args);while (rs.next()) {HoaDonChiTiet entity = new HoaDonChiTiet();entity.setMaHDCT(rs.getInt("MaHDCT"));entity.setSoLuong(rs.getInt("SoLuong"));entity.setMaHD(rs.getInt("MaHD"));entity.setMaSP(rs.getString("MaSP"));list.add(entity);}} catch (Exception e) {throw new RuntimeException(e);}return list;} |

#### 3.2.3.7 ThongKeDAO

Lớp DAO này chứa các phương thức gọi đến các thủ tục lưu được xây dựng trước đây để thức hiện các chức năng tổng hợp thống kê trong ứng dụng.

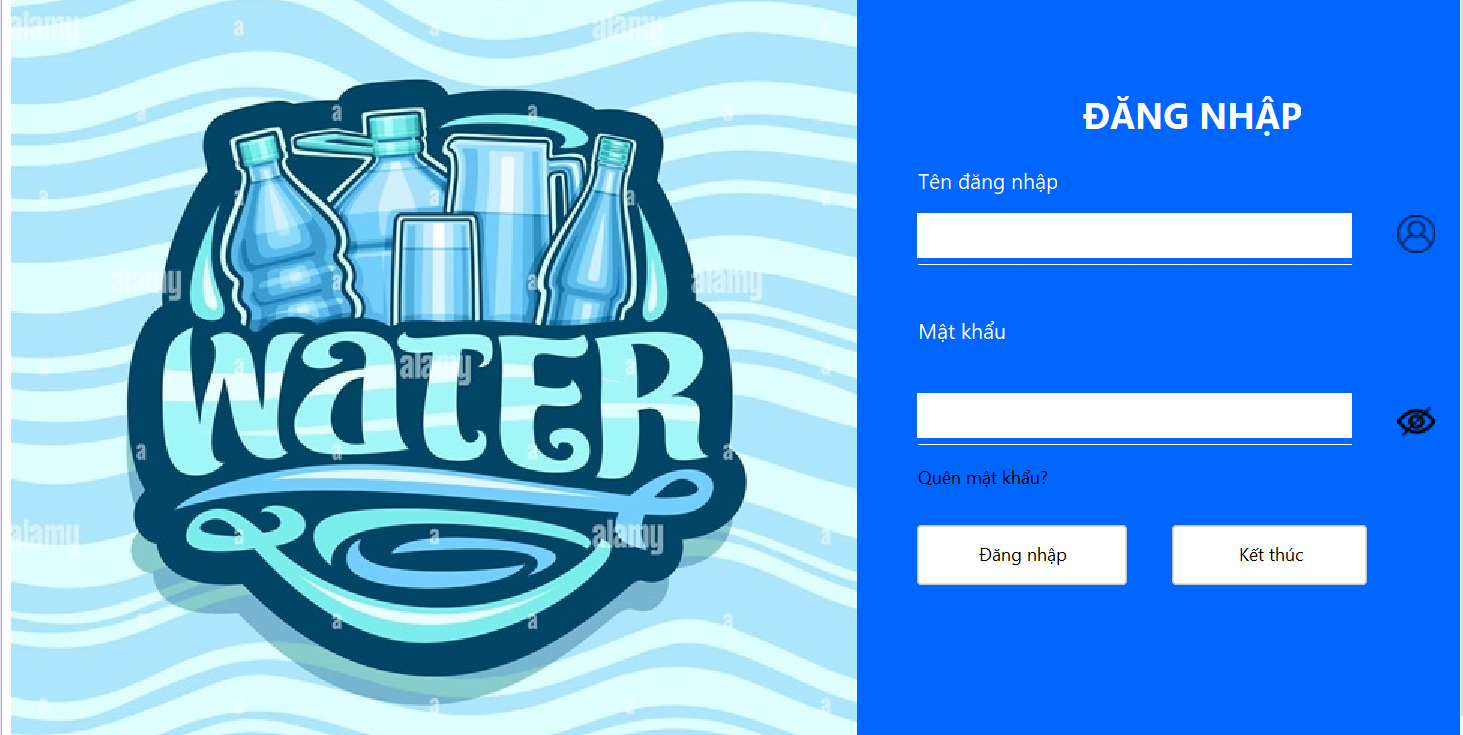
|  |
| --- |
| private List<Object[]> getListOfArray(String sql, String[] cols, Object... args) {  try {  List<Object[]> list = new ArrayList<>();  ResultSet rs = JdbcHelper.query(sql, args);  while (rs.next()) {  Object[] vals = new Object[cols.length];  for (int i = 0; i < cols.length; i++) {  vals[i] = rs.getObject(cols[i]);  }  list.add(vals);  }  rs.getStatement().getConnection().close();  return list;  } catch (Exception e) {  throw new RuntimeException(e);  }  }  public List<Object[]> getDoanhThuNgayTrongThang() {  String sql = "{CALL sp\_ThongKeDoanhThuNgay}";  String[] cols = {"NgayMua", "DoanhThu"};  return getListOfArray(sql, cols);  }    public List<Object[]> getLuongSanPhamNam(String nam) {  String sql = "{CALL sp\_LuongSanPhamNam(?)}";  String[] cols = {"LuongSanPham"};  return getListOfArray(sql, cols, nam);  }  public List<Object[]> getLuongSanPhamThang(String thang) {  String sql = "{CALL sp\_LuongSanPhamThang(?)}";  String[] cols = {"LuongSanPham"};  return getListOfArray(sql, cols, thang);  }  public List<Object[]> getLuongSanPhamNgay(String ngay) {  String sql = "{CALL sp\_LuongSanPhamNgay(?)}";  String[] cols = {"LuongSanPham"};  return getListOfArray(sql, cols, ngay);  }  public List<Object[]> getLuongKhachNam(String nam) {  String sql = "{CALL sp\_LuongKhachTheoNam(?)}";  String[] cols = {"SoKhachHang"};  return getListOfArray(sql, cols, nam);  }  public List<Object[]> getLuongKhachThang(String thang) {  String sql = "{CALL sp\_LuongKhachTheoThang(?)}";  String[] cols = {"SoKhachHang"};  return getListOfArray(sql, cols, thang);  }  public List<Object[]> getLuongKhachNgay(String ngay) {  String sql = "{CALL sp\_LuongKhachTheoNgay(?)}";  String[] cols = {"SoKhachHang"};  return getListOfArray(sql, cols, ngay);  }  public List<Object[]> getLuongHoaDonNam(String nam) {  String sql = "{CALL sp\_LuongHoaDonTheoNam(?)}";  String[] cols = {"SoHoaDon"};  return getListOfArray(sql, cols, nam);  }  public List<Object[]> getLuongHoaDonThang(String thang) {  String sql = "{CALL sp\_LuongHoaDonTheoThang(?)}";  String[] cols = {"SoHoaDon"};  return getListOfArray(sql, cols, thang);  }  public List<Object[]> getLuongHoaDonNgay(String ngay) {  String sql = "{CALL sp\_LuongHoaDonTheoNgay(?)}";  String[] cols = {"SoHoaDon"};  return getListOfArray(sql, cols, ngay);}  public List<Object[]> getDoanhThuNgay(String ngay) {  String sql = "{CALL sp\_DoanhThuTheoNgay(?)}";  String[] cols = {"DoanhThu"};  return getListOfArray(sql, cols, ngay); }  public List<Object[]> getDoanhThuThang(String thang) {  String sql = "{CALL sp\_DoanhThuTheoThang(?)}";  String[] cols = {"DoanhThu"};  return getListOfArray(sql, cols, thang);}  public List<Object[]> getDoanhThuNam(String nam) {  String sql = "{CALL sp\_DoanhThuTheoNam(?)}";  String[] cols = {"DoanhThu"};  return getListOfArray(sql, cols, nam);} |

## 3.3 VIẾT MÃ CHO ỨNG DỤNG

### 3.3.1 Viết mã cho các chức năng

Trong phần này chúng ta sẽ tiến hành viết mã cho các giao diện đã thiết kế trước đây. Cụ thể chúng ta sẽ lần lược viết mã thực hiện các chức năng cho các giao diện sau đây.

#### 3.3.1.1 Form Đăng Nhập



Hướng dẫn mã nguồn

|  |
| --- |
| NhanVienDAO dao = new NhanVienDAO();  public DangNhapJDialog(java.awt.Frame parent, boolean modal) {  super(parent, modal);  initComponents();  setLocationRelativeTo(null);  txtMaNV.setBackground(new java.awt.Color(0, 0, 0, 1));  txtMatKhau.setBackground(new java.awt.Color(0, 0, 0, 1));  }  void dangNhap() {  String strMaNV = txtMaNV.getText();  String strPassword = new String(txtMatKhau.getPassword());  NhanVien nv = dao.selectByID(strMaNV);  if (nv == null) {  MsgBox.alert(this, "Sai tên đăng nhập");  } else {  if (!nv.getMatKhau().equals(strPassword)) {  MsgBox.alert(this, "Sai mật khẩu");  } else {  Auth.user = nv;  this.dispose();  new QLBNJFrame().setVisible(true);  }  }  }  private void disviewMouseClicked(java.awt.event.MouseEvent evt) {  // TODO add your handling code here:  txtMatKhau.setEchoChar((char) 0);  disview.setVisible(false);  disview.setEnabled(false);  showview.setVisible(true);  showview.setVisible(true);  }  private void showviewMouseClicked(java.awt.event.MouseEvent evt) {  // TODO add your handling code here:  txtMatKhau.setEchoChar((char) 8226);  disview.setVisible(true);  disview.setEnabled(true);  showview.setVisible(false);  showview.setVisible(false);  }  private void btnketThucActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  System.exit(0);  }  private void btndangNhapActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  dangNhap();  }  private void lblQuenMatKhauMouseClicked(java.awt.event.MouseEvent evt) {  // TODO add your handling code here:  QuenMatKhauJDialog qmkjd= new QuenMatKhauJDialog((Frame) SwingUtilities.getWindowAncestor(this), true);  qmkjd.setVisible(true);  } |

#### Giao diện chính (MainJFrame)



Mô tả hoạt động

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
| Constructor | Init() |
| Đăng xuất |  |
| Nhân viên | childPanel = new NhanVienPanel();  pnMain.removeAll();  pnMain.add(childPanel, BorderLayout.EAST);  pnMain.revalidate(); |
| Nhà cung cấp | childPanel = new NhaCungCapPanel();  pnMain.removeAll();  pnMain.add(childPanel, BorderLayout.EAST);  pnMain.revalidate(); |
| Hóa đơn | childPanel = new HoaDonPanel();  pnMain.removeAll();  pnMain.add(childPanel, BorderLayout.EAST);  pnMain.revalidate(); |
| Sản phẩm | childPanel = new SanPhamPanel();  pnMain.removeAll();  pnMain.add(childPanel, BorderLayout.EAST);  pnMain.revalidate(); |
| Khách hàng | childPanel = new KhachHangPanel();  pnMain.removeAll();  pnMain.add(childPanel, BorderLayout.EAST);  pnMain.revalidate(); |
| Thống kê | openKhoaHoc() |
| Kết thúc | System.exit(0) |

Hướng dẫn mã nguồn

|  |
| --- |
| private JPanel childPanel;  public QLBNJFrame() {  initComponents();  init();  }  void init() {  setIconImage(XImage.getAppIcon());  setLocationRelativeTo(null);  setTitle("QUẢN LÝ BÁN NƯỚC");  childPanel = new LogoPanel();  pnMain.removeAll();  pnMain.add(childPanel, BorderLayout.EAST);  pnMain.revalidate();  }  boolean kiemTraVaiTro() {  if (Auth.user.isVaiTro() == false) {  return false;  }  return true;  }  void moHoaDon() {  childPanel = new HoaDonPanel();  pnMain.removeAll();  pnMain.add(childPanel, BorderLayout.EAST);  pnMain.revalidate();  }  private void btnHoaDonActionPerformed(java.awt.event.ActionEvent evt) {  moHoaDon();  }  private void btnNhanVienActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  if (Auth.isLogin()) {  if (kiemTraVaiTro()) {  childPanel = new NhanVienPanel();  pnMain.removeAll();  pnMain.add(childPanel, BorderLayout.EAST);  pnMain.revalidate();  } else {  MsgBox.alert(this, "Chỉ quản lý mới có quyền");  }  } else {  MsgBox.alert(this, "Vui lòng đăng nhập!");  }  }  private void btnNhaCungCapActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  if (Auth.isLogin()) {  if (kiemTraVaiTro()) {  childPanel = new NhaCungCapPanel();  pnMain.removeAll();  pnMain.add(childPanel, BorderLayout.EAST);  pnMain.revalidate();  } else {  MsgBox.alert(this, "Chỉ quản lý mới có quyền");  }  } else {  MsgBox.alert(this, "Vui lòng đăng nhập!");  }  }  private void btnSanPhamActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  if (Auth.isLogin()) {  childPanel = new SanPhamPanel();  pnMain.removeAll();  pnMain.add(childPanel, BorderLayout.EAST);  pnMain.revalidate();  } else {  MsgBox.alert(this, "Vui lòng đăng nhập!");  }  }  private void btnKetThucActionPerformed(java.awt.event.ActionEvent evt) {  if (MsgBox.confirm(this, "Ban muon ket thuc cong viec?")) {  System.exit(0);  }  }  private void btnKhachHangActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  if (Auth.isLogin()) {  childPanel = new KhachHangPanel();  pnMain.removeAll();  pnMain.add(childPanel, BorderLayout.EAST);  pnMain.revalidate();  } else {  MsgBox.alert(this, "Vui lòng đăng nhập!");  }  }  private void btnThongKeActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  if (Auth.isLogin()) {  if (kiemTraVaiTro()) {  childPanel = new ThongKeJPanel();  pnMain.removeAll();  pnMain.add(childPanel, BorderLayout.EAST);  pnMain.revalidate();  } else {  MsgBox.alert(this, "Chỉ quản lý mới có quyền");  }  } else {  MsgBox.alert(this, "Vui lòng đăng nhập!");  }  }  private void btnDangXuatActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  Auth.clear();  this.dispose();  new DangNhapJDialog(this, true).setVisible(true);  } |

#### Quản lý nhân viên (NhanVienJFrame)



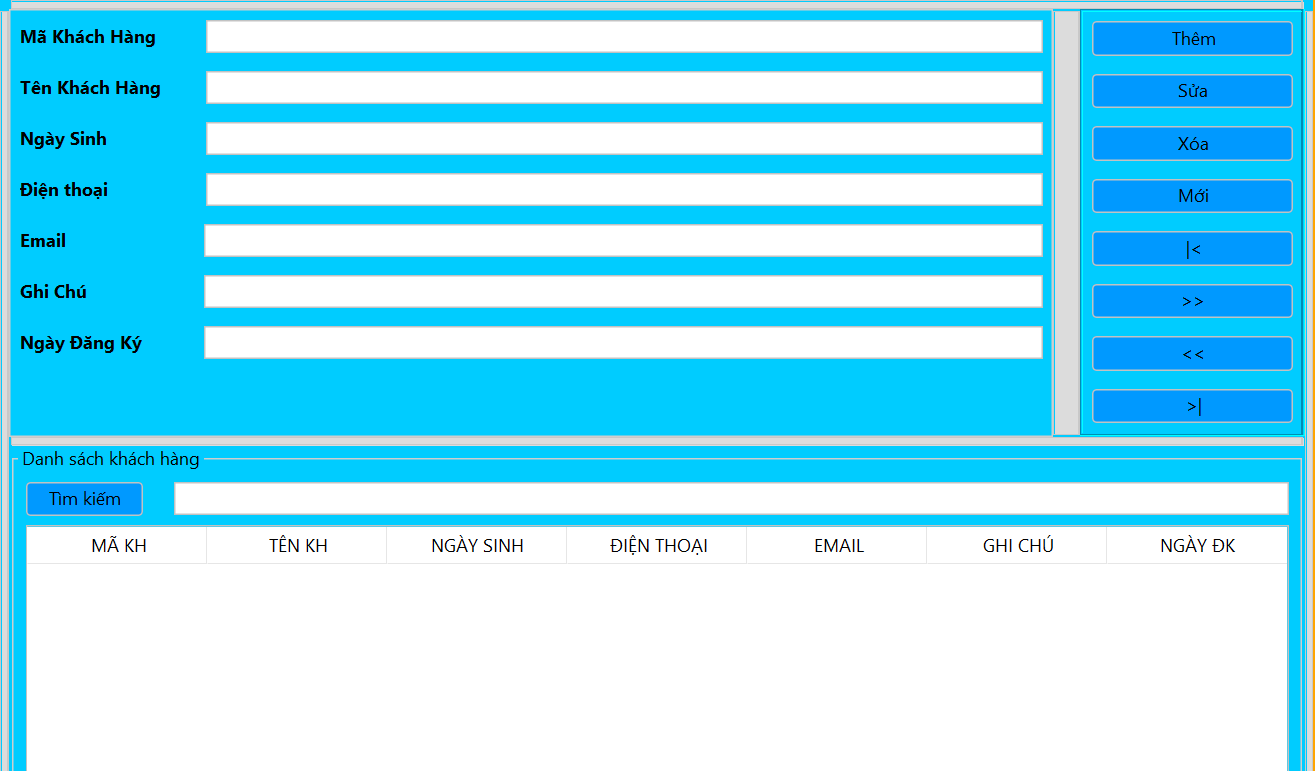
Mô tả hoạt động

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc phải làm** |
| Constructor | Init() |
| windowOpened | this.load(); this.setStatus(true); |
| tblNhanVienMouseClicked | if (evt.getClickCount() == 1) {  this.row = tblNhanVien.rowAtPoint(evt.getPoint());  edit();} |
| btnInsert | Insert() |
| btnUpdate | Update() |
| btnDelete | Delete() |
| btnClear | Clear() |
| btnFirst | this.index =0; this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++; this.edit(); |
| btnLast | this.index = tblGridView.getRowCount() - 1; this.edit(); |

Hướng dẫn cài đặt mã nguồn

|  |
| --- |
| NhanVienDAO nvDAO = new NhanVienDAO();  int row = -1;  public NhanVienPanel() {  initComponents();  init();  }  void init() {  fillToTable();  updateStatus();  }  void clearForm() {  this.setForm(new NhanVien());  this.updateStatus();  row = -1;  updateStatus();  rdoQuanLy.setSelected(true);  rdoDangLam.setSelected(true);  }  void updateStatus() {  boolean edit = this.row >= 0;  boolean first = this.row == 0;  boolean last = this.row == tblNhanVien.getRowCount() - 1;  txtMaNV.setEditable(!edit);  //Khi insert thì không update, delete  btnThem.setEnabled(!edit);  btnSua.setEnabled(edit);  btnXoa.setEnabled(edit);  btnDau.setEnabled(edit && !first);  btnSau.setEnabled(edit && !first);  btnTruoc.setEnabled(edit && !last);  btnCuoi.setEnabled(edit && !last);  lblTrangThai.setVisible(edit);  rdoDangLam.setVisible(edit);  rdoDaNghi.setVisible(edit);  }  void delete() {  if (MsgBox.confirm(this, "Bạn thực sự muốn xóa nhân viên này!")) {  String maNV = txtMaNV.getText();  try {  nvDAO.delete(maNV);  this.fillToTable();  this.clearForm();  MsgBox.alert(this, "Xóa thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Xóa thất bại!");  }  }  }  void edit() {  try {  String maNV = (String) tblNhanVien.getValueAt(this.row, 0);  NhanVien model = nvDAO.selectByID(maNV);  if (model != null) {  setForm(model);  updateStatus();  }  } catch (Exception e) {  MsgBox.alert(this, "Lỗi truy vấn dữ liệu");  }  }  void setForm(NhanVien model) {  txtMaNV.setText(model.getMaNV());  txtHoTen.setText(model.getHoTen());  txtMatKhau.setText(model.getMatKhau());  txtXacNhanMK.setText(model.getMatKhau());  txtEmail.setText(model.getEmail());  rdoQuanLy.setSelected(model.isVaiTro());  rdoNhanVien.setSelected(!model.isVaiTro());  rdoDangLam.setSelected(model.isTrangThai());  rdoDaNghi.setSelected(!model.isTrangThai());  }  void update() {  NhanVien nvModel = getForm();  String confirm = new String(txtXacNhanMK.getPassword());  if (!confirm.equals(nvModel.getMatKhau())) {  MsgBox.alert(this, "Xác nhận mật khẩu không đúng!");  } else {  try {  nvDAO.update(nvModel);  this.fillToTable();  MsgBox.alert(this, "Cập nhật thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Cập nhật thất bại!");  }  }  }  void fillToTable() {  DefaultTableModel tblModel = (DefaultTableModel) tblNhanVien.getModel();  tblModel.setRowCount(0);  try {  String keyword = txtTimKiem.getText();  List<NhanVien> list = nvDAO.selectByKeyword(keyword);  for (NhanVien nv : list) {  Object[] row = {  nv.getMaNV(),  nv.getMatKhau(),  nv.getEmail(),  nv.getHoTen(),  nv.isVaiTro() ? "Quản lý" : "Nhân viên",  nv.isTrangThai() ? "Đang làm" : "Đã nghỉ"  };  tblModel.addRow(row);  }  } catch (Exception e) {  }  }  void insert() {  NhanVien nv = getForm();  String confirm = new String(txtXacNhanMK.getPassword());  if (confirm.equals(nv.getMatKhau())) {  try {  nvDAO.insert(nv);  fillToTable();  clearForm();  MsgBox.alert(this, "Thêm mới thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Thêm mới thất bại!");  }  } else {  MsgBox.alert(this, "Mật khẩu xác nhận không đúng!");  }  }  NhanVien getForm() {  NhanVien nv = new NhanVien();  nv.setMaNV(txtMaNV.getText());  nv.setMatKhau(new String(txtMatKhau.getPassword()));  nv.setEmail(txtEmail.getText());  nv.setHoTen(txtHoTen.getText());  nv.setVaiTro(rdoQuanLy.isSelected());  nv.setTrangThai(rdoDangLam.isSelected());  return nv;  }  boolean validateForm() {  if (txtMaNV.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập mã!");  return false;  }  if (txtMatKhau.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập mật khẩu!");  return false;  }  if (txtMatKhau.getText().toString().length() < 3) {  MsgBox.alert(this, "Mật khẩu ít nhất 3 ký tự!");  return false;  }  if (txtXacNhanMK.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập xác nhận mật khẩu!");  return false;  }  if (txtEmail.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập email");  txtEmail.requestFocus();  return false;  }  if (txtEmail.getText().matches("\\w+@\\w+(\\.\\w+){1,2}") == false) {  MsgBox.alert(this, "Email sai định dạng!");  return false;  }  if (txtHoTen.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập họ tên!");  return false;  }  if (txtHoTen.getText().matches("^[a-zA-Z\\s\\p{L}]+$") == false) {  MsgBox.alert(this, "Chỉ chứa alphabet và ký tự trắng!");  txtHoTen.requestFocus();  return false;  }  return true;  }  void first() {  row = 0;  edit();  }  void prev() {  if (row > 0) {  row--;  edit();  }  }  void next() {  if (row < tblNhanVien.getRowCount() - 1) {  row++;  edit();  }  }  void last() {  row = tblNhanVien.getRowCount() - 1;  edit();  }  private void tblNhanVienMouseClicked(java.awt.event.MouseEvent evt) {  // TODO add your handling code here:  if (evt.getClickCount() == 1) {  this.row = tblNhanVien.rowAtPoint(evt.getPoint());  edit();  }  }  private void tblNhanVienMousePressed(java.awt.event.MouseEvent evt) {  // TODO add your handling code here:  }  private void btnThemActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  if (validateForm()) {  insert();  }  }  private void btnSuaActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  if (validateForm()) {  update();  }  }  private void btnXoaActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  delete();  }  private void btnMoiActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  clearForm();  }  private void btnDauActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  first();  }  private void btnSauActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  prev();  }  private void btnTruocActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  next();  }  private void btnCuoiActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  last();  }  private void rdoQuanLyActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  }  private void rdoDangLamActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  }  private void rdoDaNghiActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  }  private void btnTimKiemActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  fillToTable();  } |

#### Quản lý khách hàng



Mô tả hoạt động

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc phải làm** |
| Constructor | Init() |
| windowOpened | this.load(); this.setStatus(true); |
| tblKhachHangMouseClicked | if (evt.getClickCount() == 1) {  this.row = tblKhachHang.rowAtPoint(evt.getPoint());  edit();} |
| btnInsert | Insert() |
| btnUpdate | Update() |
| btnDelete | Delete() |
| btnClear | Clear() |
| btnFirst | this.index = 0; this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++; this.edit(); |
| btnLast | this.index = tblGridView.getRowCount() - 1; this.edit(); |
| btnTimKiem | filltToTab() |

Hướng dẫn cài đặt mã nguồn

|  |
| --- |
| KhachHangDAO khDAO = new KhachHangDAO();  int row = -1;  public KhachHangPanel() {  initComponents();  init();  }  void init() {  fillToTable();  updateStatus();  txtNgayDK.setText(String.valueOf(java.time.LocalDate.now()));  }  private void fillToTable() {  DefaultTableModel model = (DefaultTableModel) tblKhachHang.getModel();  model.setRowCount(0);  try {  String keyword = txtTimKiem.getText();  List<KhachHang> list = khDAO.selectByKeyword(keyword);  SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");  for (KhachHang kh : list) {  Object[] row = {  kh.getMaKH(),  kh.getTenKH(),  dateFormat.format(kh.getNgaySinh()),  kh.getDienThoai(),  kh.getEmail(),  kh.getGhiChu(),  dateFormat.format(kh.getNgayDangKy())  };  model.addRow(row);  }  } catch (Exception e) {  e.printStackTrace();  MsgBox.alert(this, "Lỗi truy vấn dữ liệu!");  }  }  void edit() {  try {  String maKH = (String) tblKhachHang.getValueAt(this.row, 0);  KhachHang model = khDAO.selectByID(maKH);  if (model != null) {  setForm(model);  updateStatus();  }  } catch (Exception e) {  MsgBox.alert(this, "Lỗi truy vấn dữ liệu");  }  }  void insert() {  KhachHang kh = getForm();  try {  khDAO.insert(kh);  fillToTable();  clearForm();  MsgBox.alert(this, "Thêm mới thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Thêm mới thất bại dsfsd!");  }  }  void update() {  KhachHang khModel = getForm();  try {  khDAO.update(khModel);  this.fillToTable();  this.clearForm();  MsgBox.alert(this, "Cập nhật thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Cập nhật thất bại!");  }  }  void delete() {  if (MsgBox.confirm(this, "Bạn thực sự muốn xóa khách hàng này!")) {  String maKH = txtMaKH.getText();  try {  khDAO.delete(maKH);  this.fillToTable();  this.clearForm();  MsgBox.alert(this, "Xóa thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Xóa thất bại!");  }  }  }  void clearForm() {  this.setForm(new KhachHang());  this.updateStatus();  row = -1;  updateStatus();  txtNgayDK.setText(String.valueOf(java.time.LocalDate.now()));  }  void setForm(KhachHang model) {  txtMaKH.setText(model.getMaKH());  txtTenKH.setText(model.getTenKH());  txtNgaySinh.setText(String.valueOf(model.getNgaySinh()));  txtDienThoai.setText(model.getDienThoai());  txtEmail.setText(model.getEmail());  txtGhiChu.setText(model.getGhiChu());  txtNgayDK.setText(String.valueOf(model.getNgayDangKy()));  }  private KhachHang getForm() {  KhachHang kh = new KhachHang();  kh.setMaKH(txtMaKH.getText());  kh.setTenKH(txtTenKH.getText());  kh.setNgaySinh(XDate.toDate(txtNgaySinh.getText(), "yyyy-MM-dd"));  kh.setDienThoai(txtDienThoai.getText());  kh.setEmail(txtEmail.getText());  kh.setGhiChu(txtGhiChu.getText());  kh.setNgayDangKy(XDate.toDate(txtNgayDK.getText(), "yyyy-MM-dd"));  return kh;  }  void updateStatus() {  boolean edit = this.row >= 0;  boolean first = this.row == 0;  boolean last = this.row == tblKhachHang.getRowCount() - 1;  txtMaKH.setEditable(!edit);  //Khi insert thì không update, delete  btnThem.setEnabled(!edit);  btnSua.setEnabled(edit);  btnXoa.setEnabled(edit);  btnDau.setEnabled(edit && !first);  btnSau.setEnabled(edit && !first);  btnTruoc.setEnabled(edit && !last);  btnCuoi.setEnabled(edit && !last);  }  boolean validateForm() {  if (txtMaKH.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập mã!");  return false;}  if (txtTenKH.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập tên!");  return false;}  if (txtTenKH.getText().matches("^[a-zA-Z\\s\\p{L}]+$") == false) {  MsgBox.alert(this, "Chỉ chứa alphabet và ký tự trắng!");  txtTenKH.requestFocus();  return false;}  if (txtNgaySinh.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập ngày sinh!");  return false;}  if (txtDienThoai.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập điện thoại!");  return false;}  if (txtDienThoai.getText().matches("[0-9]+") == false) {  MsgBox.alert(this, "Chỉ cho nhập số!");  txtDienThoai.requestFocus();  return false;}  if (txtEmail.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập email!");  return false;}  if (txtEmail.getText().matches("\\w+@\\w+(\\.\\w+){1,2}") == false) {  MsgBox.alert(this, "Email sai định dạng!");  return false;}  if (txtGhiChu.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập ghi chú!");  return false;}  if (txtNgayDK.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập ngày đăng ký");  return false;}  return true;}  void first() {  row = 0;  edit();}  void prev() {  if (row > 0) {  row--;  edit();}}  void next() {  if (row < tblKhachHang.getRowCount() - 1) {  row++;  edit();}}  void last() {  row = tblKhachHang.getRowCount() - 1;  edit();} |

#### Quản lý nhà cung cấp



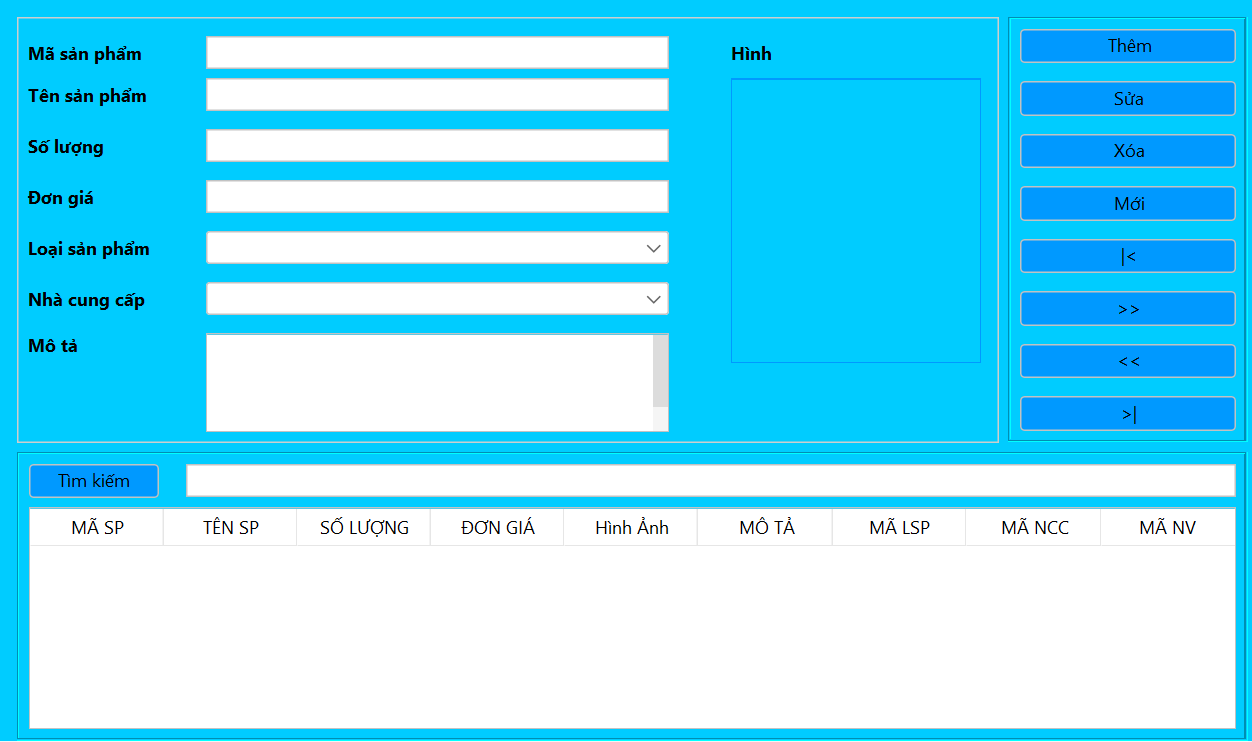
Mô tả hoạt động

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc phải làm** |
| Constructor | Init() |
| windowOpened | this.load(); this.setStatus(true); |
| tblNhaCungCapMousePressed | if (evt.getClickCount() == 2) {  this.row = tblNhaCungCap.rowAtPoint(evt.getPoint());  edit();} |
| btnInsert | Insert() |
| btnUpdate | Update() |
| btnDelete | Delete() |
| btnClear | Clear() |
| btnFirst | this.index = 0; this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++; this.edit(); |
| btnLast | this.index = tblGridView.getRowCount() - 1; this.edit(); |
| btnTimKiem | fiilToTab(); |

Hướng dẫn cài đặt mã nguồn

|  |
| --- |
| NhaCungCapDAO nCCDAO = new NhaCungCapDAO();  int row = -1;  /\*\*  \* Creates new form NhaCungCapPanel  \*/  public NhaCungCapPanel() {  initComponents();  init();  }  void init() {  fillToTable();  updateStatus();  }  void clearForm() {  this.setForm(new NhaCungCap());  this.updateStatus();  row = -1;  updateStatus();  }  void updateStatus() {  boolean edit = this.row >= 0;  boolean first = this.row == 0;  boolean last = this.row == tblNhaCungCap.getRowCount() - 1;  txtMaNCC.setEditable(!edit);  //Khi insert thì không update, delete  btnThem.setEnabled(!edit);  btnSua.setEnabled(edit);  btnXoa.setEnabled(edit);  btnDau.setEnabled(edit && !first);  btnSau.setEnabled(edit && !first);  btnTruoc.setEnabled(edit && !last);  btnCuoi.setEnabled(edit && !last);  }  void setForm(NhaCungCap model) {  txtMaNCC.setText(model.getMaNCC());  txtTenNCC.setText(model.getTenNCC());  txtSDT.setText(model.getDienThoai());  txtDiaChi.setText(model.getDiaChi());  txtEmail.setText(model.getEmail());  txtGhiChu.setText(model.getGhiChu());  }  NhaCungCap getForm() {  NhaCungCap ncc = new NhaCungCap();  ncc.setMaNCC(txtMaNCC.getText());  ncc.setTenNCC(txtTenNCC.getText());  ncc.setDienThoai(txtSDT.getText());  ncc.setDiaChi(txtDiaChi.getText());  ncc.setEmail(txtEmail.getText());  ncc.setGhiChu(txtGhiChu.getText());  return ncc;  }  void fillToTable() {  DefaultTableModel tblModel = (DefaultTableModel) tblNhaCungCap.getModel();  tblModel.setRowCount(0);  try {  String keyword = txtTimKiem.getText();  List<NhaCungCap> list = nCCDAO.selectByKeyword(keyword);  for (NhaCungCap ncc : list) {  Object[] row = {  ncc.getMaNCC(),  ncc.getTenNCC(),  ncc.getDienThoai(),  ncc.getDiaChi(),  ncc.getEmail(),  ncc.getGhiChu(),};  tblModel.addRow(row);  }  } catch (Exception e) {  }  }  void edit() {  try {  String maNCC = (String) tblNhaCungCap.getValueAt(this.row, 0);  NhaCungCap model = nCCDAO.selectByID(maNCC);  if (model != null) {  setForm(model);  updateStatus();  }  } catch (Exception e) {  MsgBox.alert(this, "Lỗi truy vấn dữ liệu");  }  }  void insert() {  NhaCungCap ncc = getForm();  try {  nCCDAO.insert(ncc);  fillToTable();  clearForm();  MsgBox.alert(this, "Thêm mới thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Thêm mới thất bại!");  }  }  void update() {  NhaCungCap nCCModel = getForm();  try {  nCCDAO.update(nCCModel);  this.fillToTable();  this.clearForm();  MsgBox.alert(this, "Cập nhật thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Cập nhật thất bại!");  }  }  void delete() {  if (MsgBox.confirm(this, "Bạn thực sự muốn xóa nhà cung cấp này!")) {  String maNCC = txtMaNCC.getText();  try {  nCCDAO.delete(maNCC);  this.fillToTable();  this.clearForm();  MsgBox.alert(this, "Xóa thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Xóa thất bại!");  }  }  }  boolean validateForm() {  if (txtMaNCC.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập mã!");  txtMaNCC.requestFocus();  return false;  }  if (txtTenNCC.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập tên!");  txtTenNCC.requestFocus();  return false;  }  if (txtSDT.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập số điện thoại");  txtSDT.requestFocus();  return false;  }  if (txtSDT.getText().matches("[0-9]+") == false) {  MsgBox.alert(this, "Chỉ cho nhập số!");  txtSDT.requestFocus();  return false;  }  if (txtDiaChi.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập địa chỉ");  txtDiaChi.requestFocus();  return false;  }  if (txtEmail.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập email");  txtEmail.requestFocus();  return false;  }  if (txtEmail.getText().matches("\\w+@\\w+(\\.\\w+){1,2}") == false) {  MsgBox.alert(this, "Email sai định dạng!");  return false;  }  if (txtGhiChu.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập ghi chú");  txtGhiChu.requestFocus();  return false;  } return true;}  void first() {  row = 0;  edit();}  void prev() {  if (row > 0) {  row--;  edit();}}  void next() {  if (row < tblNhaCungCap.getRowCount() - 1) {  row++;  edit();}}  void last() {  row = tblNhaCungCap.getRowCount() - 1;  edit();} |

#### Quản lý sản phẩm



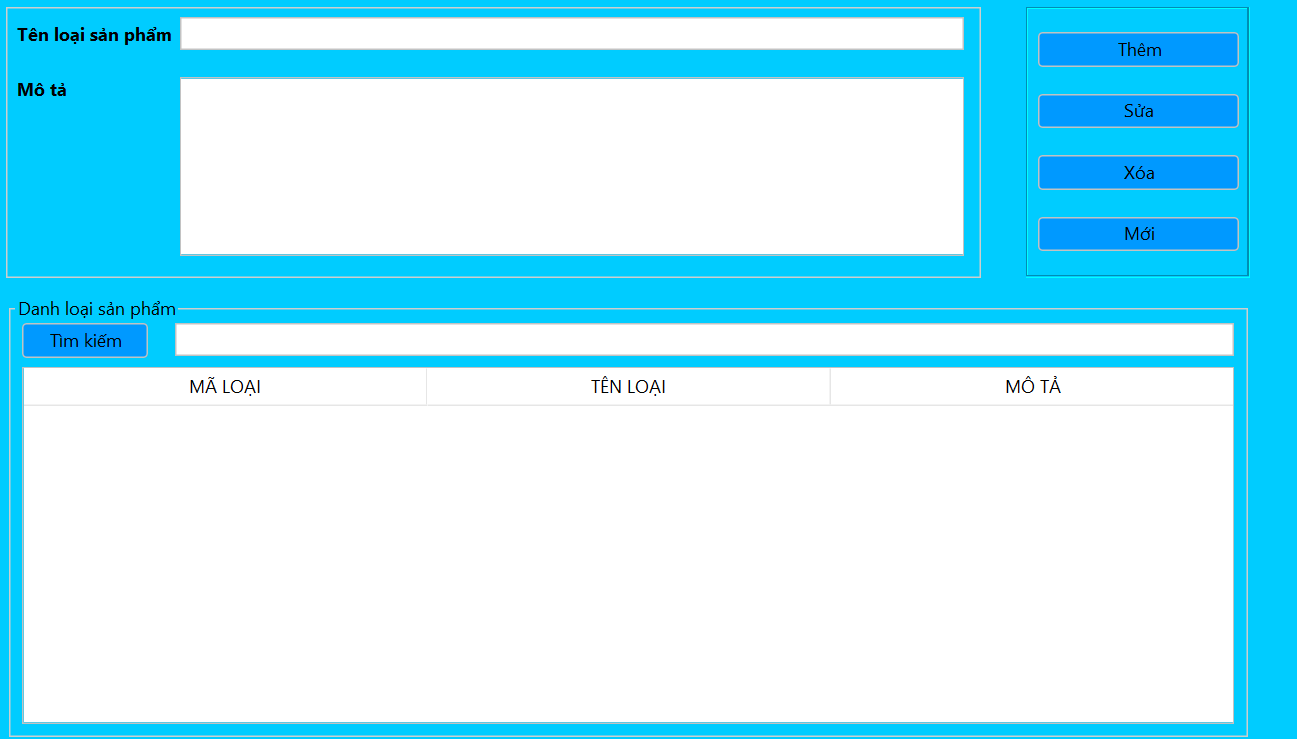
Mô tả hoạt động

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc phải làm** |
| Constructor | Init() |
| tblGridView.mouseClicked | if (evt.getClickCount() == 2) {  this.row = tblSanPham.rowAtPoint(evt.getPoint());  editSanPham();  } |
| btnInsert | Insert() |
| btnUpdate | Update() |
| btnDelete | Delete() |
| btnClear | Clear() |
| btnFirst | this.index = 0; this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++; this.edit(); |
| btnLast | this.index = tblGridView.getRowCount() - 1; this.edit(); |
| cboLoaiSP |  |
| cboNCC |  |

Hướng dẫn mã nguồn

|  |
| --- |
| JFileChooser fileChooser = new JFileChooser();  LoaiSanPhamDAO lspDAO = new LoaiSanPhamDAO();  NhaCungCapDAO nccDAO = new NhaCungCapDAO();  SanPhamDAO spDAO = new SanPhamDAO();  int rowLSP = -1;  int rowSP = -1;  boolean test = false;  public SanPhamPanel() {  initComponents();  init();  }  void init() {  fillToTableLSP();  fillComboBoxLSP();  fillComboBoxNCC();  fillToTableSP();  updateStatusSP();  updateStatusLSP();  }  //Sản phẩm  void updateStatusSP() {  boolean edit = this.rowSP >= 0;  boolean first = this.rowSP == 0;  boolean last = this.rowSP == tblSanPham.getRowCount() - 1;  txtMaSP.setEditable(!edit);  //Khi insert thì không update, delete  btnThemSP.setEnabled(!edit);  btnSuaSP.setEnabled(edit);  btnXoaSP.setEnabled(edit);  btnDauSP.setEnabled(edit && !first);  btnSauSP.setEnabled(edit && !first);  btnTruocSP.setEnabled(edit && !last);  btnCuoiSP.setEnabled(edit && !last);  }  void deleteSP() {  if (MsgBox.confirm(this, "Bạn thực sự muốn xóa sản phẩm này!")) {  String maSP = txtMaSP.getText();  try {  spDAO.delete(maSP);  test = true;  this.fillToTableSP();  this.clearFormSP();  MsgBox.alert(this, "Xóa thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Xóa thất bại!");  }  }  }  void updateSP() {  SanPham model = getFormSP();  try {  spDAO.update(model);  this.fillToTableSP();  MsgBox.alert(this, "Cập nhật thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Cập nhật thất bại!");  }  }  void editSP() {  try {  String maSP = (String) tblSanPham.getValueAt(this.rowSP, 0);  SanPham model = spDAO.selectByID(maSP);  if (model != null) {  this.setFormSP(model);  updateStatusSP();  }  } catch (Exception e) {  MsgBox.alert(this, "Lỗi truy vấn dữ liệu");  }  }  void setFormSP(SanPham model) {  txtMaSP.setText(model.getMaSP());  txtTenSP.setText(model.getTenSP());  txtSoLuong.setText(String.valueOf(model.getSoLuong()));  txtDonGia.setText(String.valueOf(model.getDonGia()));  int demLSP = 0;  try {  List<LoaiSanPham> list = lspDAO.selectAll();  for (LoaiSanPham lsp : list) {  if (lsp.getMaLSP() == model.getMaLSP()) {  break;  }  demLSP++;  }  } catch (Exception e) {  }  cboLoaiSanPham.setSelectedIndex(demLSP);  int demNCC = 0;  try {  List<NhaCungCap> list = nccDAO.selectAll();  for (NhaCungCap ncc : list) {  if (ncc.getMaNCC().equals(model.getMaNCC())) {  break;  }  demNCC++;  }  } catch (Exception e) {  }  cboNhaCungCap.setSelectedIndex(demNCC);  txtMoTaSP.setText(model.getMoTa());  if (model.getHinhAnh() != null) {  lblHinh.setIcon(XImage.read(model.getHinhAnh()));  lblHinh.setToolTipText(model.getHinhAnh());  }  }  void clearFormSP() {  updateStatusSP();  txtMaSP.setText("");  txtTenSP.setText("");  txtSoLuong.setText("");  txtDonGia.setText("");  cboLoaiSanPham.setSelectedIndex(0);  cboNhaCungCap.setSelectedIndex(0);  txtMoTaSP.setText("");  lblHinh.setIcon(null);  rowSP = -1;  updateStatusSP();  }  void chonAnh() {  if (fileChooser.showOpenDialog(this) == JFileChooser.APPROVE\_OPTION) {  File file = fileChooser.getSelectedFile();  XImage.save(file);  ImageIcon icon = XImage.read(file.getName());  lblHinh.setIcon(icon);  lblHinh.setToolTipText(file.getName());  }  }  SanPham getFormSP() {  SanPham sp = new SanPham();  sp.setMaSP(txtMaSP.getText());  sp.setTenSP(txtTenSP.getText());  sp.setSoLuong(Integer.parseInt(txtSoLuong.getText()));  sp.setDonGia(Double.parseDouble(txtDonGia.getText()));  sp.setHinhAnh(lblHinh.getToolTipText());  sp.setMoTa(txtMoTaSP.getText());  MyCombobox maLSP = (MyCombobox) cboLoaiSanPham.getSelectedItem();  sp.setMaLSP(maLSP.MaInt());  MyCombobox maNCC = (MyCombobox) cboNhaCungCap.getSelectedItem();  sp.setMaNCC(maNCC.MaString());  sp.setMaNV(Auth.user.getMaNV());  return sp;  }  void insertSP() {  SanPham sp = getFormSP();  try {  spDAO.insert(sp);  fillToTableSP();  fillComboBoxLSP();  clearFormSP();  MsgBox.alert(this, "Thêm mới thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Thêm mới thất bại!");  }  }  void fillToTableSP() {  DefaultTableModel tblModeSP = (DefaultTableModel) tblSanPham.getModel();  tblModeSP.setRowCount(0);  try {  String keyword = txtTimKiemSP.getText();  List<SanPham> list = spDAO.selectByKeyword(keyword);  for (SanPham sp : list) {  Object[] row = {  sp.getMaSP(),  sp.getTenSP(),  sp.getSoLuong(),  sp.getDonGia(),  sp.getHinhAnh(),  sp.getMoTa(),  sp.getMaLSP(),  sp.getMaNCC(),  sp.getMaNV()  };  tblModeSP.addRow(row);  }  } catch (Exception e) {  MsgBox.alert(this, "Lỗi truy vấn dữ liệu sản phẩm!");  }  }  void fillComboBoxNCC() {  DefaultComboBoxModel cboModelNCC = (DefaultComboBoxModel) cboNhaCungCap.getModel();  cboModelNCC.removeAllElements();  try {  List<NhaCungCap> list = nccDAO.selectAll();  for (NhaCungCap ncc : list) {  MyCombobox mycbb = new MyCombobox(ncc.getMaNCC(), ncc.getTenNCC());  cboModelNCC.addElement(mycbb);  }  } catch (Exception e) {  }  }  void fillComboBoxLSP() {  DefaultComboBoxModel cboModelLSP = (DefaultComboBoxModel) cboLoaiSanPham.getModel();  cboModelLSP.removeAllElements();  try {  List<LoaiSanPham> list = lspDAO.selectAll();  for (LoaiSanPham lsp : list) {  MyCombobox mycbb = new MyCombobox(lsp.getMaLSP(), lsp.getTenLSP());  cboModelLSP.addElement(mycbb);  }  } catch (Exception e) {  }  }  boolean validateFormSP() {  if (txtMaSP.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập mã!");  return false;  }  if (txtTenSP.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập tên sản phẩm!");  return false;  }  if (txtSoLuong.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập số lượng!");  return false;  }  if (txtSoLuong.getText().matches("[0-9]+") == false) {  MsgBox.alert(this, "Chỉ cho nhập số!");  txtSoLuong.requestFocus();  return false;  }  if (txtDonGia.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập đơn giá!");  return false;  }  if (txtDonGia.getText().matches("[0-9]+") == false) {  MsgBox.alert(this, "Chỉ cho nhập số!");  txtDonGia.requestFocus();  return false;  }  if (txtMoTaSP.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập mô tả!");  return false;  }  return true;  } |

#### Quản lý loại sản phẩm



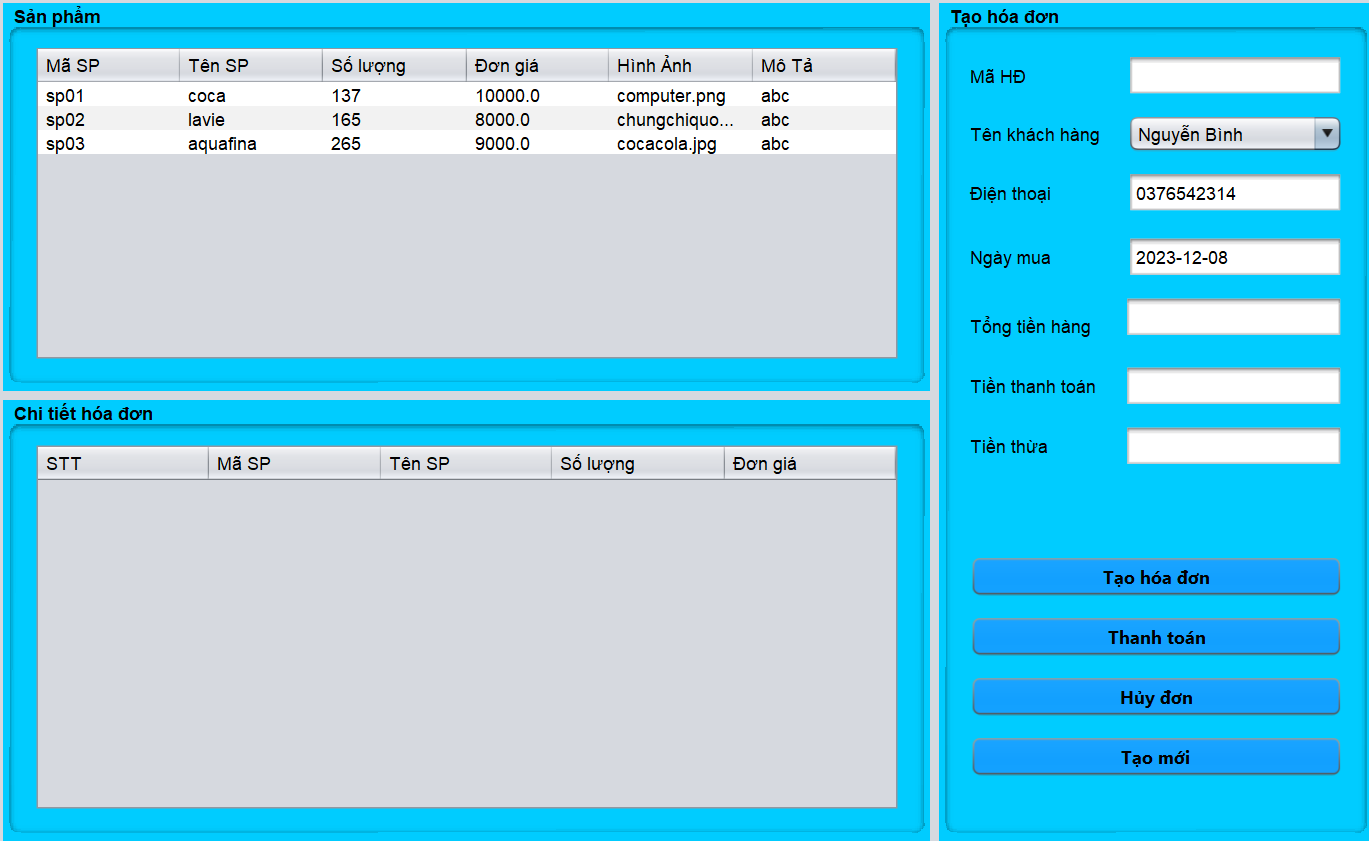
Mô tả hoạt động

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
| Constructor | Init()  this.MaKH = MaKH; |
| btnThem | Insert() |
| btnSua | Update() |
| btnXoa | Delete() |
| btnMoi | ClearForm() |
| tblLoaiSanPhamMousePressed | if (evt.getClickCount() == 2) {  this.rowLSP = tblLoaiSanPham.rowAtPoint(evt.getPoint());  editLoaiSanPham();  } |

Hướng dẫn cài đặt mã nguồn

|  |
| --- |
| void updateStatusLSP() {  boolean edit = this.rowLSP >= 0;  boolean first = this.rowLSP == 0;  boolean last = this.rowLSP == tblLoaiSanPham.getRowCount() - 1;  //Khi insert thì không update, delete  btnThemLoaiSP.setEnabled(!edit);  btnSuaLSP.setEnabled(edit);  btnXoaLSP.setEnabled(edit);  }  void clearFormLSP() {  txtTenLSP.setText("");  txtMoTaLSP.setText("");  this.updateStatusLSP();  rowLSP = -1;  updateStatusLSP();  }  void updateLSP() {  int maLSP = (int) tblLoaiSanPham.getValueAt(rowLSP, 0);  LoaiSanPham lspModel = new LoaiSanPham();  lspModel.setMaLSP(maLSP);  lspModel.setTenLSP(txtTenLSP.getText());  lspModel.setMoTa(txtMoTaLSP.getText());  try {  lspDAO.update(lspModel);  this.fillToTableLSP();  fillComboBoxLSP();  this.clearFormLSP();  MsgBox.alert(this, "Cập nhật thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Cập nhật thất bại!");  }  }  void deleteLSP() {  if (MsgBox.confirm(this, "Bạn thực sự muốn xóa nhà cung cấp này!")) {  int maLSP = (int) tblLoaiSanPham.getValueAt(rowLSP, 0);  try {  lspDAO.delete(String.valueOf(maLSP));  this.fillToTableLSP();  fillComboBoxLSP();  this.clearFormLSP();  MsgBox.alert(this, "Xóa thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Xóa thất bại!");  }  }  }  void editLSP() {  try {  int maLSP = (int) tblLoaiSanPham.getValueAt(this.rowLSP, 0);  LoaiSanPham model = lspDAO.selectByID(String.valueOf(maLSP));  if (model != null) {  setFormLSP(model);  updateStatusLSP();  }  } catch (Exception e) {  MsgBox.alert(this, "Lỗi truy vấn dữ liệu");  }  }  void setFormLSP(LoaiSanPham model) {  txtTenLSP.setText(model.getTenLSP());  txtMoTaLSP.setText(model.getMoTa());  }  void fillToTableLSP() {  DefaultTableModel tblModel = (DefaultTableModel) tblLoaiSanPham.getModel();  tblModel.setRowCount(0);  try {  String keyword = txtTimKiemLSP.getText();  List<LoaiSanPham> list = lspDAO.selectByKeyword(keyword);  for (LoaiSanPham lsp : list) {  Object[] row = {  lsp.getMaLSP(),  lsp.getTenLSP(),  lsp.getMoTa(),};  tblModel.addRow(row);  }  } catch (Exception e) {  }  }  void insertLSP() {  LoaiSanPham lsp = getFormLSP();  try {  lspDAO.insert(lsp);  fillToTableLSP();  clearFormLSP();  MsgBox.alert(this, "Thêm mới thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Thêm mới thất bại!");  }  }  LoaiSanPham getFormLSP() {  LoaiSanPham lsp = new LoaiSanPham();  lsp.setTenLSP(txtTenLSP.getText());  lsp.setMoTa(txtMoTaLSP.getText());  return lsp;  }  void first() {  rowSP = 0;  editSP();  }  void prev() {  if (rowSP > 0) {  rowSP--;  editSP();}}  void next() {  if (rowSP < tblSanPham.getRowCount() - 1) {  rowSP++;  editSP();}}  void last() {  rowSP = tblSanPham.getRowCount() - 1;  editSP();}  boolean validateFormLSP() {  if (txtTenLSP.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập tên sản phẩm!");  return false;}return true;} |

#### Quản Lý Hóa Đơn

  
Mô tả hoạt động

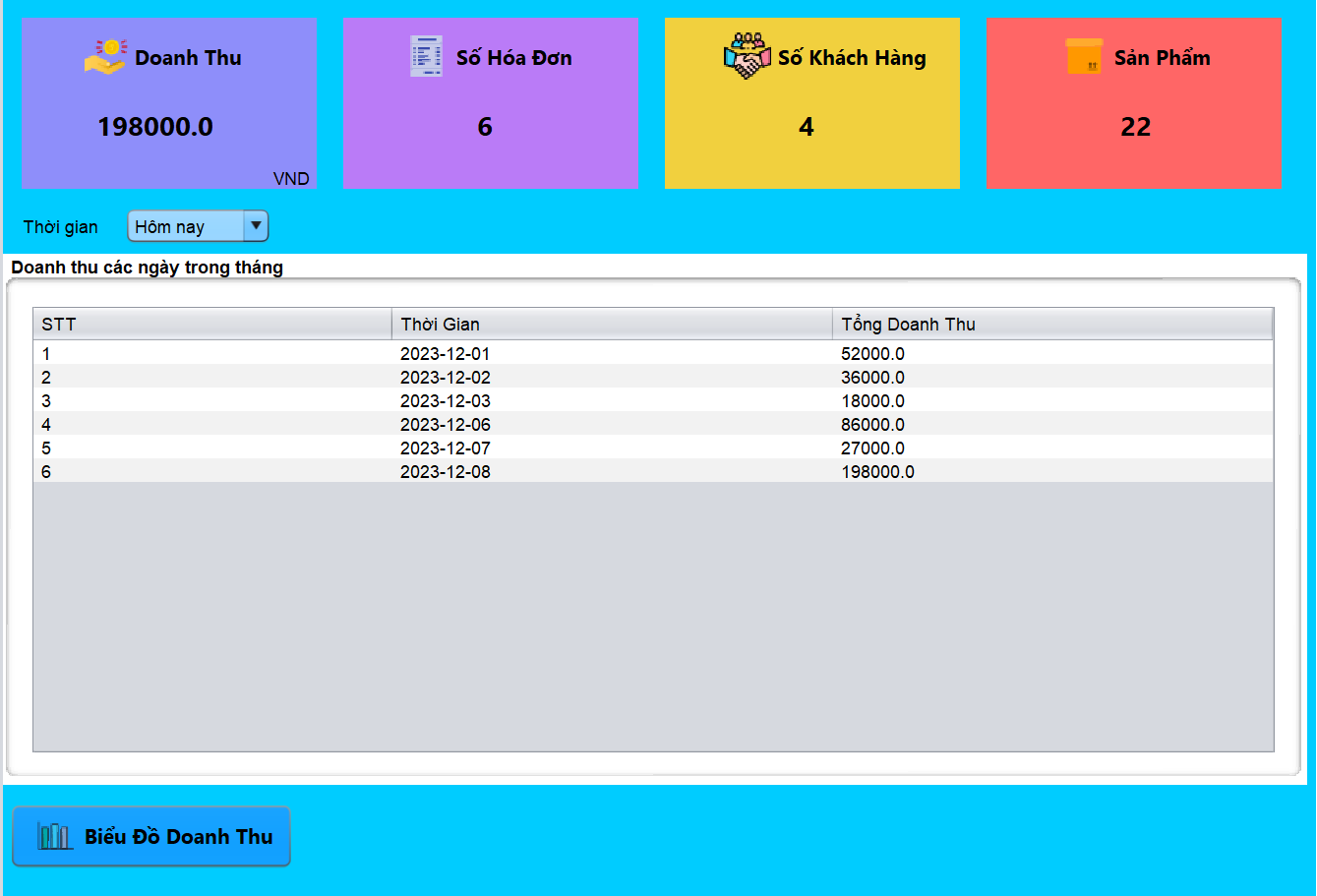
|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init() |
| btnThemHoaDon | fillTable = true;  HoaDonJDialog dshd = new HoaDonJDialog((Frame) SwingUtilities.getWindowAncestor(this), true);  dshd.setVisible(true);  if (fillTable) {  fillTableHoaDon();  fillTable = false;} |
| btnUpdateHoaDon | fillTable = true;  if (row != -1) {  int maHD = (Integer) tblHoaDon.getValueAt(row, 0);  HoaDon hd = hdDAO.selectByID(String.valueOf(maHD));  if (hd.isTrangThai()) {  MsgBox.alert(this, "Hóa đơn đã thanh toán!");} else {  HoaDonJDialog dshd = new HoaDonJDialog((Frame) SwingUtilities.getWindowAncestor(this), true);  dshd.setEditData(hd);  dshd.setVisible(true);  fillTableHoaDon();}  fillTable = false;  row = -1;} else {MsgBox.alert(this, "Chưa chọn hóa đơn!");} |
| tblHoaDonMouseClicked | this.row = tblHoaDon.rowAtPoint(evt.getPoint()); |

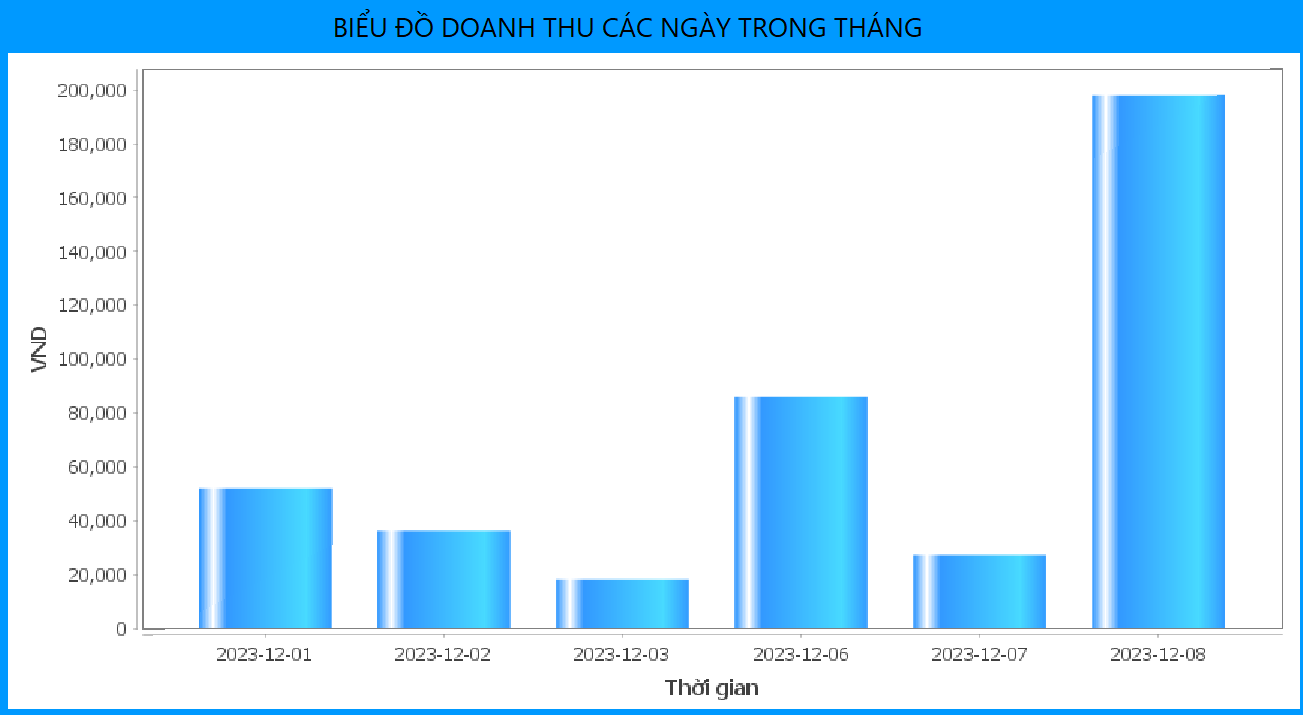
Hướng dẫn cài đặt mã nguồn

|  |
| --- |
| HoaDonDAO hdDAO = new HoaDonDAO();  private int row = -1;  boolean fillTable = false;  public HoaDonPanel() {  initComponents();  fillTableHoaDon();  }  void fillTableHoaDon() {  DefaultTableModel model = (DefaultTableModel) tblHoaDon.getModel();  model.setRowCount(0);    String tenKH = txtTenKH.getText();  List<Object[]> list = hdDAO.getHoaDon(tenKH);  for (Object[] row : list) {  boolean trangThai = (boolean) row[4];  model.addRow(new Object[]{row[0], row[1], row[2], row[3], trangThai ? "Đã thanh toán" : "Chưa thanh toán"});  }  }  private void tblHoaDonMousePressed(java.awt.event.MouseEvent evt) {  // TODO add your handling code here:  }  private void btnThemDonActionPerformed(java.awt.event.ActionEvent evt) {  fillTable = true;  HoaDonJDialog dshd = new HoaDonJDialog((Frame) SwingUtilities.getWindowAncestor(this), true);  dshd.setVisible(true);  if (fillTable) {  fillTableHoaDon();  fillTable = false;  }  }  private void tblHoaDonMouseClicked(java.awt.event.MouseEvent evt) {  // TODO add your handling code here:  this.row = tblHoaDon.rowAtPoint(evt.getPoint());  }  private void btnCapNhatDonActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  fillTable = true;  if (row != -1) {  int maHD = (Integer) tblHoaDon.getValueAt(row, 0);  HoaDon hd = hdDAO.selectByID(String.valueOf(maHD));  if (hd.isTrangThai()) {  MsgBox.alert(this, "Hóa đơn đã thanh toán!");  } else {  HoaDonJDialog dshd = new HoaDonJDialog((Frame) SwingUtilities.getWindowAncestor(this), true);  dshd.setEditData(hd);  dshd.setVisible(true);  fillTableHoaDon();  }  fillTable = false;  row = -1;  } else {  MsgBox.alert(this, "Chưa chọn hóa đơn!");  }  }  private void btnTimKiemActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  fillTableHoaDon();  } |

|  |
| --- |
| KhachHangDAO khDAO = new KhachHangDAO();  SanPhamDAO spDAO = new SanPhamDAO();  HoaDonDAO hdDAO = new HoaDonDAO();  HoaDonChiTietDAO hdctDAO = new HoaDonChiTietDAO();  int rowSP = -1;  int rowCTHD = -1;  /\*\*  \* Creates new form DSHoaDonJDialog  \*/  public HoaDonJDialog(java.awt.Frame parent, boolean modal) {  super(parent, modal);  initComponents();  this.setLocationRelativeTo(null);  fillTableSanPham();  fillComboboxKhachHang();  txtNgayMua.setText(XDate.toString(XDate.now(), "yyyy-MM-dd"));  txtMaHD.setEditable(false);  txtTongTienHang.setEditable(false);  txtTienThua.setEditable(false);  }  void capNhatHoaDonChiTiet() {  try {  String maSP = (String) tblChiTietHoaDon.getValueAt(rowCTHD, 1);  int maHDCT = hdctDAO.selectMaHDCT(Integer.parseInt(txtMaHD.getText()), maSP).getMaHDCT();  HoaDonChiTiet hdct = hdctDAO.selectByID(String.valueOf(maHDCT));  int soLuongDaCoHDCT = hdct.getSoLuong();  int soLuongNhapHDCT = Integer.parseInt(MsgBox.prompt(this, "Nhập số lượng sản phẩm"));  SanPham sp = spDAO.selectByID(String.valueOf(hdct.getMaSP()));  if (soLuongNhapHDCT == 0) {  spDAO.updateSoLuongSP(sp.getSoLuong() + hdct.getSoLuong(), hdct.getMaSP());  hdctDAO.delete(String.valueOf(maHDCT));  fillTableHoaDonChiTiet();  fillTableSanPham();  } else if (soLuongNhapHDCT > soLuongDaCoHDCT) {  int soLuongThem = soLuongNhapHDCT - soLuongDaCoHDCT;  if (sp.getSoLuong() < soLuongThem) {  MsgBox.alert(this, "Không đủ sản phẩm!");  } else {  spDAO.updateSoLuongSP(sp.getSoLuong() - soLuongThem, sp.getMaSP());  hdctDAO.updateSoLuong(soLuongNhapHDCT, maHDCT);  fillTableHoaDonChiTiet();  fillTableSanPham();  }  } else {  int soLuongGiam = soLuongDaCoHDCT - soLuongNhapHDCT;  spDAO.updateSoLuongSP(sp.getSoLuong() + soLuongGiam, sp.getMaSP());  hdctDAO.updateSoLuong(soLuongNhapHDCT, maHDCT);  fillTableHoaDonChiTiet();  fillTableSanPham();  }  } catch (Exception e) {  }  }  void huyHoaDon() {  try {  List<HoaDonChiTiet> list = hdctDAO.selectByHoaDon(Integer.parseInt(txtMaHD.getText()));  for (HoaDonChiTiet hdct : list) {  int soLuongDaNhap = hdct.getSoLuong();  int soLuongConLai = spDAO.selectByID(hdct.getMaSP()).getSoLuong();  int soLuongSP = soLuongDaNhap + soLuongConLai;  spDAO.updateSoLuongSP(soLuongSP, hdct.getMaSP());  hdctDAO.deleteByInt(hdct.getMaHDCT());  }  hdDAO.deleteByInt(Integer.parseInt(txtMaHD.getText()));  clearForm();  fillTableSanPham();  MsgBox.alert(this, "Hủy hóa đơn thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Hủy hóa đơn thất bại!");  }  }  void thanhToan() {  try {  hdDAO.updateTrangThai(true, Integer.parseInt(txtMaHD.getText()));  MsgBox.alert(this, "Thanh toán thành công!");  } catch (Exception e) {  MsgBox.alert(this, "Thanh toán thất bại!");  }  }  boolean validateForm() {  if (txtMaHD.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa có hóa đơn!");  return false;  }  if (txtTongTienHang.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa có sản phẩm!");  return false;  }  if (txtTienThanhToan.getText().trim().isEmpty()) {  MsgBox.alert(this, "Chưa nhập tiền thanh toán!");  return false;  }  if (txtTienThanhToan.getText().matches("[0-9]+") == false) {  MsgBox.alert(this, "Chỉ cho nhập số!");  txtTienThanhToan.requestFocus();  return false;  }  if (Double.parseDouble(txtTienThanhToan.getText()) < Double.parseDouble(txtTongTienHang.getText())) {  MsgBox.alert(this, "Không đủ tiền thanh toán!");  return false;  }  return true;  }  void themSanPhamVaoHoaDon() {  try {  HoaDonChiTiet hdct = new HoaDonChiTiet();  int soLuongSPNhap = Integer.parseInt(MsgBox.prompt(this, "Nhập số lượng"));  SanPham sp = spDAO.selectByID((String) tblSanPham.getValueAt(rowSP, 0));  boolean kiemTraSP = false;  List<HoaDonChiTiet> listHDCT = hdctDAO.selectAllHDCTByMaHD(Integer.parseInt(txtMaHD.getText()));  for (HoaDonChiTiet hoaDonChiTiet : listHDCT) {  if (hoaDonChiTiet.getMaSP().equals(sp.getMaSP())) {  kiemTraSP = true;  }  }  if (kiemTraSP) {  MsgBox.alert(this, "Sản phẩm đã có trong hóa đơn!");  kiemTraSP = false;  } else {  if (soLuongSPNhap > sp.getSoLuong()) {  MsgBox.alert(this, "Không đủ sản phẩm!");  } else {  hdct.setSoLuong(soLuongSPNhap);  hdct.setMaHD(Integer.parseInt(txtMaHD.getText()));  hdct.setMaSP((String) tblSanPham.getValueAt(rowSP, 0));  hdctDAO.insert(hdct);  int soLuongSPConLai = sp.getSoLuong() - soLuongSPNhap;  spDAO.updateSoLuongSP(soLuongSPConLai, (String) tblSanPham.getValueAt(rowSP, 0));  MsgBox.alert(this, "Thêm sản phẩm thành công");  fillTableHoaDonChiTiet();  fillTableSanPham();  }    }  } catch (Exception e) {  }  }  void fillTableHoaDonChiTiet() {  DefaultTableModel model = (DefaultTableModel) tblChiTietHoaDon.getModel();  model.setRowCount(0);  try {  List<HoaDonChiTiet> list = hdctDAO.selectByHoaDon(Integer.parseInt(txtMaHD.getText()));  double tongTien = 0;  for (int i = 0; i < list.size(); i++) {  HoaDonChiTiet hdct = list.get(i);  String tenSP = spDAO.selectByID(hdct.getMaSP()).getTenSP();  double donGia = spDAO.selectByID(hdct.getMaSP()).getDonGia();  Object[] row = {  i + 1, hdct.getMaSP(), tenSP, hdct.getSoLuong(), donGia  };  model.addRow(row);  tongTien += (hdct.getSoLuong() \* donGia);  txtTongTienHang.setText(String.valueOf(tongTien));  }  } catch (Exception e) {  MsgBox.alert(this, "Hiển thị hóa đơn chi tiết thất bại");  }  }  void taoHoaDon() {  HoaDon hd = getForm();  try {  hdDAO.insert(hd);  MsgBox.alert(this, "Tạo hóa đơn thành công!");  btnThanhToan.setEnabled(true);  btnHuyDon.setEnabled(true);  } catch (Exception e) {  MsgBox.alert(this, "Tạo hóa đơn thất bại!");  }  }  HoaDon getForm() {  HoaDon hd = new HoaDon();  hd.setNgayMua(XDate.toDate(txtNgayMua.getText(), "yyyy-MM-dd"));  hd.setTrangThai(false);  MyCombobox KHDuocChon = (MyCombobox) cboKhachHang.getSelectedItem();  hd.setMaKH(KHDuocChon.MaString());  hd.setMaNV(Auth.user.getMaNV());  return hd;  }  void clearForm() {  txtMaHD.setText("");  cboKhachHang.setSelectedIndex(0);  DefaultTableModel model = (DefaultTableModel) tblChiTietHoaDon.getModel();  model.setRowCount(0);  txtTongTienHang.setText("");  txtTienThanhToan.setText("");  txtTienThua.setText("");  rowSP = -1;  }  void setForm(HoaDon model) {  txtMaHD.setText(String.valueOf(model.getMaHD()));  txtNgayMua.setText(XDate.toString(model.getNgayMua(), "yyyy-MM-dd"));  int dem = 0;  // Hiển thị khách hàng của hóa đơn ở combobox khách hàng  try {  List<KhachHang> list = khDAO.selectAll();  for (KhachHang kh : list) {  if (kh.getMaKH().equals(model.getMaKH())) {  break;  }  dem++;  }  } catch (Exception e) {  }  cboKhachHang.setSelectedIndex(dem);  }  void selectComboBox() {  MyCombobox KHDuocChon = (MyCombobox) cboKhachHang.getSelectedItem();  txtDienThoai.setText(khDAO.selectByID(KHDuocChon.MaString()).getDienThoai());  }  void fillComboboxKhachHang() {  DefaultComboBoxModel cboModelKH = (DefaultComboBoxModel) cboKhachHang.getModel();  cboModelKH.removeAllElements();  try {  List<KhachHang> list = khDAO.selectAll();  for (KhachHang kh : list) {  MyCombobox mycbb = new MyCombobox(kh.getMaKH(), kh.getTenKH());  cboModelKH.addElement(mycbb);  }  } catch (Exception e) {  }  }  void fillTableSanPham() {  DefaultTableModel tblModelSP = (DefaultTableModel) tblSanPham.getModel();  tblModelSP.setRowCount(0);  try {  List<SanPham> list = spDAO.selectAll();  for (SanPham sp : list) {  Object[] row = {  sp.getMaSP(),  sp.getTenSP(),  sp.getSoLuong(),  sp.getDonGia(),  sp.getHinhAnh(),  sp.getMoTa(),};  tblModelSP.addRow(row);  }  } catch (Exception e) {  }  }  private void btnTaoMoiActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  clearForm();  }  private void btnThanhToanActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  if (validateForm()) {  thanhToan();  clearForm();  }  }  private void btnHuyDonActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  if (txtMaHD.getText().equals("")) {  MsgBox.alert(this, "Chưa chọn đơn hàng!");  } else {  huyHoaDon();  }  this.dispose();  }  private void btnTaoHDActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  if (txtMaHD.getText().equals("")) {  taoHoaDon();  int index = hdDAO.selectAll().size() - 1;  List list = hdDAO.selectAll();  HoaDon hd = (HoaDon) list.get(index);  txtMaHD.setText(String.valueOf(hd.getMaHD()));  } else {  MsgBox.alert(this, "Hóa đơn đã tồn tại!");  }  }  private void cboKhachHangActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  selectComboBox();  }  private void tblSanPhamMouseClicked(java.awt.event.MouseEvent evt) {  // TODO add your handling code here:  if (txtMaHD.getText().equals("")) {  MsgBox.alert(this, "Chưa có hóa đơn!");  } else {  if (evt.getClickCount() == 1) {  this.rowSP = tblSanPham.rowAtPoint(evt.getPoint());  themSanPhamVaoHoaDon();  }  }  }  private void tblChiTietHoaDonMouseClicked(java.awt.event.MouseEvent evt) {  // TODO add your handling code here:  if (evt.getClickCount() == 1) {  rowCTHD = tblChiTietHoaDon.rowAtPoint(evt.getPoint());  capNhatHoaDonChiTiet();  }  }  private void txtTienThanhToanKeyReleased(java.awt.event.KeyEvent evt) {  // TODO add your handling code here:  txtTienThua.setText(String.valueOf(Double.parseDouble(txtTienThanhToan.getText())-Double.parseDouble(txtTongTienHang.getText())));  } |

#### Thống kê doanh thu





Mô tả hoạt động

|  |  |
| --- | --- |
| **Sự kiện** | **Công việc** |
| Constructor | init();  doanhThuTheoNgay();  soLuongHoaDonTheoNgay();  soLuongKhachTheoNgay();  luongSanPhamTheoNgay();  doanhThuNgayTrongThang(); |
| cboThoiGian | if (cboThoiGian.getSelectedIndex() == 0) {  doanhThuTheoNgay();  soLuongHoaDonTheoNgay();  soLuongKhachTheoNgay();  luongSanPhamTheoNgay();  }  if (cboThoiGian.getSelectedIndex() == 1) {  doanhThuTheoThang();  soLuongHoaDonTheoThang();  soLuongKhachTheoThang();  luongSanPhamTheoThang();  }  if (cboThoiGian.getSelectedIndex() == 2) {  doanhThuTheoNam();  soLuongHoaDonTheoNam();  soLuongKhachTheoNam();  luongSanPhamTheoNam();  } |

Hướng dẫn cài đặt mã nguồn

|  |
| --- |
| void doanhThuNgayTrongThang() {  DefaultTableModel model = (DefaultTableModel) tblDoanhThu.getModel();  model.setRowCount(0);  List<Object[]> list = tkDAO.getDoanhThuNgayTrongThang();  int i = 1;  for (Object[] row : list) {  model.addRow(new Object[]{i, row[0], row[1]});  i++;  }  }  void luongSanPhamTheoNam() {  String ngay = String.valueOf(java.time.LocalDate.now());  List<Object[]> list = tkDAO.getLuongSanPhamNam(ngay);  for (Object[] row : list) {  lblSanPham.setText(String.valueOf(row[0]));  }  }  void luongSanPhamTheoThang() {  String ngay = String.valueOf(java.time.LocalDate.now());  List<Object[]> list = tkDAO.getLuongSanPhamThang(ngay);  for (Object[] row : list) {  lblSanPham.setText(String.valueOf(row[0]));  }  }  void luongSanPhamTheoNgay() {  lblSanPham.setText("");  String ngay = String.valueOf(java.time.LocalDate.now());  List<Object[]> list = tkDAO.getLuongSanPhamNgay(ngay);  for (Object[] row : list) {  lblSanPham.setText(String.valueOf(row[0]));  }  }  void soLuongKhachTheoNam() {  String ngay = String.valueOf(java.time.LocalDate.now());  List<Object[]> list = tkDAO.getLuongKhachNam(ngay);  for (Object[] row : list) {  lblKhachHang.setText(String.valueOf(row[0]));  }  }  void soLuongKhachTheoThang() {  String ngay = String.valueOf(java.time.LocalDate.now());  List<Object[]> list = tkDAO.getLuongKhachThang(ngay);  for (Object[] row : list) {  lblKhachHang.setText(String.valueOf(row[0]));  }  }  void soLuongKhachTheoNgay() {  lblKhachHang.setText("");  String ngay = String.valueOf(java.time.LocalDate.now());  List<Object[]> list = tkDAO.getLuongKhachNgay(ngay);  for (Object[] row : list) {  lblKhachHang.setText(String.valueOf(row[0]));  }  }  void soLuongHoaDonTheoNam() {  String ngay = String.valueOf(java.time.LocalDate.now());  List<Object[]> list = tkDAO.getLuongHoaDonNam(ngay);  for (Object[] row : list) {  lblSoHoaDon.setText(String.valueOf(row[0]));  }  }  void soLuongHoaDonTheoThang() {  String ngay = String.valueOf(java.time.LocalDate.now());  List<Object[]> list = tkDAO.getLuongHoaDonThang(ngay);  for (Object[] row : list) {  lblSoHoaDon.setText(String.valueOf(row[0]));  }  }  void soLuongHoaDonTheoNgay() {  lblSoHoaDon.setText("");  String ngay = String.valueOf(java.time.LocalDate.now());  List<Object[]> list = tkDAO.getLuongHoaDonNgay(ngay);  for (Object[] row : list) {  lblSoHoaDon.setText(String.valueOf(row[0]));  }  }  void doanhThuTheoNgay() {  lblDoanhThu.setText("");  String ngay = String.valueOf(java.time.LocalDate.now());  List<Object[]> list = tkDAO.getDoanhThuNgay(ngay);  for (Object[] row : list) {  lblDoanhThu.setText(String.valueOf(row[0]));  }  }  void doanhThuTheoThang() {  String thang = String.valueOf(java.time.LocalDate.now());  List<Object[]> list = tkDAO.getDoanhThuThang(thang);  for (Object[] row : list) {  lblDoanhThu.setText(String.valueOf(row[0]));  }}  void doanhThuTheoNam() {  String nam = String.valueOf(java.time.LocalDate.now());  List<Object[]> list = tkDAO.getDoanhThuNam(nam);  for (Object[] row : list) {  lblDoanhThu.setText(String.valueOf(row[0]));  }}  private void cboThoiGianActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  if (cboThoiGian.getSelectedIndex() == 0) {  doanhThuTheoNgay();  soLuongHoaDonTheoNgay();  soLuongKhachTheoNgay();  luongSanPhamTheoNgay(); }  if (cboThoiGian.getSelectedIndex() == 1) {  doanhThuTheoThang();  soLuongHoaDonTheoThang();  soLuongKhachTheoThang();  luongSanPhamTheoThang(); }  if (cboThoiGian.getSelectedIndex() == 2) {  doanhThuTheoNam();  soLuongHoaDonTheoNam();  soLuongKhachTheoNam();  luongSanPhamTheoNam();} }  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  BieuDoDoanhThuJDialog dshd = new BieuDoDoanhThuJDialog((Frame) SwingUtilities.getWindowAncestor(this), true);  dshd.setVisible(true);} |

Jdialog biểu đồ doanh thu

|  |
| --- |
| ThongKeDAO tkDAO = new ThongKeDAO();  public BieuDoDoanhThuJDialog(java.awt.Frame parent, boolean modal) {  super(parent, modal);  initComponents();  setLocationRelativeTo(null);  showBarChart();  }  public void showBarChart() {  DefaultCategoryDataset dataset = new DefaultCategoryDataset();  List<Object[]> list = tkDAO.getDoanhThuNgayTrongThang();  for (Object[] row : list) {  dataset.addValue((Number) row[1], "Doanh Thu", (Comparable) row[0]);  }  JFreeChart chart = ChartFactory.createBarChart("", "Thời gian", "VND",  dataset, PlotOrientation.VERTICAL, false, true, false);  CategoryPlot categoryPlot = chart.getCategoryPlot();  //categoryPlot.setRangeGridlinePaint(Color.BLUE);  categoryPlot.setBackgroundPaint(Color.WHITE);  BarRenderer renderer = (BarRenderer) categoryPlot.getRenderer();  Color clr3 = new Color(51, 153, 255);  renderer.setSeriesPaint(0, clr3);  ChartPanel barpChartPanel = new ChartPanel(chart);  pnlDoanhThu.removeAll();  pnlDoanhThu.add(barpChartPanel, BorderLayout.CENTER);  pnlDoanhThu.validate();} |

# 4. KIỂM THỬ

## 4.1 Kiểm thử form chung

|  |  |  |  |
| --- | --- | --- | --- |
| **II. Sàng lọc các trường hợp thử nghiệm chung** | |  |  |
| **#** | **Hoạt động** | **Kết quả mong đợi** | **Đáp ứng hoặc không** |
| **A** | **Kiểm tra giao diện** |  |  |
| **1** | **Thuộc tính trang** |  |  |
| 1.1 | Kiểm tra xem bố cục có phù hợp với yêu cầu của khách hàng chưa | Tiêu đề của giao diện đặt tại vị trí không phù hợp | Không |
| 1.2 | Kiểm tra chính tả tên của các thành phần đã đúng chưa | Tên của các thành phần đã đúng chính tả theo tài liệu thiết kế | Đáp ứng |
| 1.3 | Nút hình ảnh luôn chứa chú giải công cụ | Hình ảnh dưới dạng liên kết có mẹo/gợi ý thông điệp phù hợp | Đáp ứng |
| 1.4 | Các mục/dữ liệu trong hộp thả xuống được sắp xếp theo thứ tự bảng chữ cái | Các mục trong menu chưa được sắp xếp theo bảng chữ cái | Không |
| **2** | **Kích thước** |  |  |
| 2.1 | Kiểm tra kích thước của các button, label, text,... | Kích thước đúng theo trong tài liệu thiết kế | Đáp ứng |
| **3** | **Căn chỉnh giao diện** |  |  |
| 3.1 | Kiểm tra căn chỉnh khoảng cách của các button, label, text,... | Khoảng cách phù hợp đúng trong tài liệu thiết kế | Đáp ứng |
| 3.2 | Kiểm tra căn chỉnh khoảng cách, vị trí của văn bản hiển thị trong list | Văn bản căn giữa | Đáp ứng |
| **4** | **Màu sắc** |  |  |
| 4.1 | Màu của giao diện có phù hợp với yêu cầu người dùng | Màu của giao diện hài hòa đúng trong tài liệu thiết kế | Đáp ứng |
| **B.** | **Kiểm tra chức năng** |  |  |
| **1** | **Menu** |  |  |
| 1.1 | Kiểm tra xem menu có chưa đủ các chức năng chưa | Menu chứa đầy đủ các chức năng trong tài liệu thiết kế | Đáp ứng |
| 1.2 | Khi chọn vào các chức năng thì đường dẫn đã đúng chưa | Sau khi chọn các chức năng thì hiển thị form giao diện tương ứng với các chức năng đó | Đáp ứng |
| **2** | **Button** |  |  |
| 2.1 Quản lý thông tin | Kiểm tra có hiện form quản lý thông tin sau khi chọn | Hiển thị form quản lý thông tin sau khi chọn | Đáp ứng |
| 2.2 Quản lý lương | Kiểm tra có hiện form quản lý lương sau khi chọn | Hiển thị form quản lý lương sau khi chọn | Đáp ứng |
| 2.3 Thống kê | Kiểm tra có hiện form thống kê sau khi chọn | Hiển thị form thống kê sau khi chọn | Đáp ứng |
| **3** | **Đăng xuất khỏi ứng dụng** |  |  |
| 3.1 | Kiển tra xem có hiện thông báo xác nhận đăng xuất và chọn có | Quay trở lại trang đăng nhập | Đáp ứng |
| 3.2 | Kiển tra xem có hiện thông báo xác nhận đăng xuất và chọn không | Không có gì thay đổi | Đáp ứng |

## 4.2 Kiểm thử đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Hoạt động** | **Kết quả mong đợi** | **Đáp ứng hoặc không** |
| **A** | **Kiểm tra giao diện** |  |  |
| **1** | **Thuộc tính trang** |  |  |
| 1.1 | Kiểm tra xem bố cục có phù hợp với màn hình có độ phân giải tối thiểu được xác định không | Màn hình hiển thị không phù hợp | Không |
| 1.2 | Kiểm tra tiêu đề (chính tả) của các thành phần | Tiêu đề đúng chính tả ở tất cả các ngôn ngữ của ứng dụng theo tài liệu thiết kế | Đáp ứng |
| 1.3 | Kiểm tra hình ảnh icon giao diện | Hình ảnh rõ nét , đúng vị trí |  |
| 1.4 | Kiểm tra giao diện thu nhỏ | Không có thanh cuộn ngang. Không có thanh cuộn dọc | Không |
| **2** | **Kích thước** |  |  |
| 2.1 | Kiểm tra icon, font size, font style, font color của các text trên màn hình | Kích thước của:  phông chữ -> như trong tài liệu thiết kế  những cái khác -> phải phù hợp (đẹp) theo bố cục | Đáp ứng |
| **3** | **Căn chỉnh của mục** |  |  |
| 3.1 | Kiểm tra căn chỉnh khoảng cách của các text field, button, label,... | Khoảng cách phù hợp đúng trong tài liệu thiết kế | Đáp ứng |
| 3.2 | Kiểm tra căn chỉnh các trường trong bố cục | Tất cả các trường phải được căn chỉnh với nhau | Đáp ứng |
| **4** | **Màu sắc** |  |  |
| 4.1 | Màu của giao diện có phù hợp với yêu cầu người dùng | Màu của giao diện hài hòa đúng trong tài liệu thiết kế | Đáp ứng |
| **B** | **KIỂM TRA CHỨC NĂNG** |  |  |
| **1** | **Đăng nhập** |  |  |
| 1.1 | Kiểm tra chức năng đăng nhập với tài khoản chưa đã tồn tại điền đầy đủ thông tin và đúng định dạng | Thông báo tài khoản không tồn tại | Đáp ứng |
| 1.2 | Kiểm tra click hiển thị mật khẩu | Mật khẩu được hiển thị sau khi click | Đáp ứng |
| 1.3 | Kiểm tra chức năng đăng nhập không điền tên đăng nhập | Không hiển thị thông báo chưa điền tên đăng nhập | Không |
| 1.4 | Kiểm tra chức năng đăng nhập không điền mật khẩu | Không hiển thị thông báo chưa điền tên đăng nhập | Không |
| 1.5 | Kiểm tra chức năng đăng nhập điền đầy đủ thông tin và sai định dạng | Thông báo sai định dạng được hiển thị | Đáp ứng |
| 1.6 | Kiểm tra chức năng đăng nhập với tài khoản đã tồn tại điền đầy đủ thông tin và đúng định dạng | Đăng nhập thành công | Đáp ứng |

## 4.3 Kiểm thử quản lý nhân viên

|  |  |  |  |
| --- | --- | --- | --- |
| **II. Sàng lọc các trường hợp thử nghiệm chung** | |  |  |
| **#** | **Hoạt động** | **Kết quả mong đợi** | **Đáp ứng hoặc không** |
| **A** | **Kiểm tra giao diện** |  |  |
| **1** | **Thuộc tính trang** |  |  |
| 1.1 | Kiểm tra chính tả tên của các thành phần đã đúng chưa | Tên của các thành phần đã đúng chính tả theo tài liệu thiết kế | Đáp ứng |
| 1.2 | Nút hình ảnh luôn chứa chú giải công cụ | Hình ảnh dưới dạng liên kết có mẹo/gợi ý thông điệp phù hợp | Đáp ứng |
| 1.3 | Kiểm tra thanh cuộn trong danh sách thông tin nhân viên | Không hiển thị thanh cuộn ngang và dọc | Không |
| **2** | **Kích thước** |  |  |
| 2.1 | Kiểm tra kích thước của các button, label, text, list,... | Kích thước đúng theo trong tài liệu thiết kế | Đáp ứng |
| **3** | **Căn chỉnh giao diện** |  |  |
| 3.1 | Kiểm tra căn chỉnh khoảng cách của các text field, button, label, list... | Khoảng cách phù hợp đúng trong tài liệu thiết kế | Đáp ứng |
| 3.2 | Kiểm tra căn chỉnh khoảng cách, vị trí của văn bản hiển thị trong list | Văn bản căn giữa | Đáp ứng |
| **4** | **Màu sắc** |  |  |
| 4.1 | Màu của giao diện có phù hợp với yêu cầu người dùng | Màu của giao diện hài hòa đúng trong tài liệu thiết kế | Đáp ứng |
| **B.** | **Kiểm tra chức năng** |  |  |
| **1** | Thêm nhân viên |  |  |
| 1.1 | Kiểm tra chức năng thêm nhân viên với dữ liệu được điền đầy đủ và đúng định dạng | Thêm thành công nhân viên | Đáp ứng |
| 1.2 | Kiểm tra chức năng thêm nhân viên với dữ liệu bỏ trống | Thêm thất bại và thông báo lỗi yêu cầu người dùng nhập mục bỏ trống | Đáp ứng |
| 1.3 | Kiểm tra chức năng thêm nhân viên với dữ liệu sai định dạng | Thêm thành công nhân viên | Không |
| 1.4 | Kiểm tra chức năng thêm nhân viên với dữ liệu đã tồn tại trước đó | Thêm thất bại và hiển thị thông báo nhân viên đã tồn tại | Đáp ứng |
| **2** | **Cập nhật nhân viên** |  |  |
| 2.1 | Kiểm tra chức năng cập nhật với dữ liệu đầy đủ, đúng định dạng | Cập nhật thành công | Đáp ứng |
| 2.2 | Kiểm tra chức năng cập nhật với dữ liệu bỏ trống | Cập nhật thất bại, hiển thị thông báo lỗi | Đáp ứng |
| 2.3 | Kiểm tra chức năng cập nhật với dữ liệu sai định dạng | Cập nhật thành công | Không |
| **3** | **Xóa nhân viên** |  |  |
| 3.1 | Kiểm tra chức năng xóa nhân viên thành công | Hiển thị thông báo thành công và nhân viên được xóa khỏi dử liệu | Đáp ứng |
| 3.2 | Kiểm tra chức năng xóa nhân viên thất bại | Xóa không thành công nhưng không hiển thị thông báo | Không |
| 3.3 | Xóa nhân viên có chứa thông tin ở các bản khác | Xóa không thành công và hiển thị thông báo | Đáp ứng |
| **4** | **Tìm kiếm nhân viên** |  |  |
| 4.1 | Tìm nhân viên tồn tại trong dữ liệu | Hiển thị thành công nhân viên lên form | Đáp ứng |
| 4.2 | Tìm nhân viên không tồn tại | Không hiển thị nhân viên và thông báo | Đáp ứng |
| 4.3 | Tìm nhân viên thành công theo mã nhân viên, họ tên, mã căn cước | Hiển thị thông tin nhân viên lên form | Đáp ứng |
| **5** | **Danh sách nhân viên** |  |  |
| 5.1 | Chọn nhân viên trong danh sách hiển thị lên form | Thông tin được hiển thị lên form | Đáp ứng |
| 5.2 | Thông tin nhân viên được lưu xuống thành công sau khi thêm, sửa xóa | Thông tin nhân viên được hiển thị lên danh sách | Đáp ứng |
| 5.3 | Thông tin nhân viên được lưu xuống thất bại sau khi thêm, sửa xóa | Không hiển thị thông tin nhân viên đó vào danh sách | Đáp ứng |

## 4.4 Kiểm thử nhà cung cấp

|  |  |  |  |
| --- | --- | --- | --- |
| **II. Sàng lọc các trường hợp thử nghiệm chung** | |  |  |
| **#** | **Hoạt động** | **Kết quả mong đợi** | **Đáp ứng hoặc không** |
| **A** | **Kiểm tra giao diện** |  |  |
| **1** | **Thuộc tính trang** |  |  |
| 1.1 | Kiểm tra chính tả tên của các thành phần đã đúng chưa | Tên của các thành phần đã đúng chính tả theo tài liệu thiết kế | Đáp ứng |
| 1.2 | Nút hình ảnh luôn chứa chú giải công cụ | Hình ảnh dưới dạng liên kết có mẹo/gợi ý thông điệp phù hợp | Đáp ứng |
| 1.3 | Kiểm tra thanh cuộn trong danh sách thông tin nhân viên | Không hiển thị thanh cuộn ngang và dọc | Không |
| **2** | **Kích thước** |  |  |
| 2.1 | Kiểm tra kích thước của các button, label, text, list,... | Kích thước đúng theo trong tài liệu thiết kế | Đáp ứng |
| **3** | **Căn chỉnh giao diện** |  |  |
| 3.1 | Kiểm tra căn chỉnh khoảng cách của các text field, button, label, list... | Khoảng cách phù hợp đúng trong tài liệu thiết kế | Đáp ứng |
| 3.2 | Kiểm tra căn chỉnh khoảng cách, vị trí của văn bản hiển thị trong list | Văn bản căn giữa | Đáp ứng |
| **4** | **Màu sắc** |  |  |
| 4.1 | Màu của giao diện có phù hợp với yêu cầu người dùng | Màu của giao diện hài hòa đúng trong tài liệu thiết kế | Đáp ứng |
| **B.** | **Kiểm tra chức năng** |  |  |
| **1** | Thêm nhà cung cấp |  |  |
| 1.1 | Kiểm tra chức năng thêm nhà cung cấp với dữ liệu được điền đầy đủ và đúng định dạng | Thêm thành công nhà cung cấp | Đáp ứng |
| 1.2 | Kiểm tra chức năng thêm nhà cung cấp với dữ liệu bỏ trống | Thêm thất bại và thông báo lỗi yêu cầu người dùng nhập mục bỏ trống | Đáp ứng |
| 1.3 | Kiểm tra chức năng thêm nhà cung cấp với dữ liệu sai định dạng | Thêm thành công nhà cung cấp | Không |
| 1.4 | Kiểm tra chức năng thêm nhà cung cấp với dữ liệu đã tồn tại trước đó | Thêm thất bại và hiển thị thông báo nhà cung cấp đã tồn tại | Đáp ứng |
| **2** | **Cập nhật nhà cung cấp** |  |  |
| 2.1 | Kiểm tra chức năng cập nhật với dữ liệu đầy đủ, đúng định dạng | Cập nhật thành công | Đáp ứng |
| 2.2 | Kiểm tra chức năng cập nhật với dữ liệu bỏ trống | Cập nhật thất bại, hiển thị thông báo lỗi | Đáp ứng |
| 2.3 | Kiểm tra chức năng cập nhật với dữ liệu sai định dạng | Cập nhật thành công | Không |
| **3** | **Xóa nhà cung cấp** |  |  |
| 3.1 | Kiểm tra chức năng xóa nhà cung cấp thành công | Hiển thị thông báo thành công và nhà cung cấp được xóa khỏi dử liệu | Đáp ứng |
| 3.2 | Kiểm tra chức năng xóa nhà cung cấp thất bại | Xóa không thành công nhưng không hiển thị thông báo | Không |
| 3.3 | Xóa nhà cung cấp có chứa thông tin ở các bản khác | Xóa không thành công và hiển thị thông báo | Đáp ứng |
| **4** | **Tìm kiếm nhà cung cấp** |  |  |
| 4.1 | Tìm nhà cung cấp tồn tại trong dữ liệu | Hiển thị thành công nhà cung cấp lên form | Đáp ứng |
| 4.2 | Tìm nhà cung cấp không tồn tại | Không hiển thị nhân viên và thông báo | Đáp ứng |
| 4.3 | Tìm nhà cung cấp thành công theo mã nhà cung cấp, họ tên, | Hiển thị thông tin nhà cung cấp lên form | Đáp ứng |
| **5** | **Danh sách nhà cung cấp** |  |  |
| 5.1 | Chọn nhà cung cấp trong danh sách hiển thị lên form | Thông tin được hiển thị lên form | Đáp ứng |
| 5.2 | Thông tin nhà cung cấp được lưu xuống thành công sau khi thêm, sửa xóa | Thông tin nhà cung cấp được hiển thị lên danh sách | Đáp ứng |
| 5.3 | Thông tin nhà cung cấp được lưu xuống thất bại sau khi thêm, sửa xóa | Không hiển thị thông tin nhà cung cấp đó vào danh sách | Đáp ứng |
| **6** | Xóa form |  |  |
| 6.1 | Xóa thông tin trên form | Form được xóa trở về mặt định ban đầu | Đáp ứng |
| **7** | **Nút "First"** |  |  |
| 7.1 | Kiểm tra khi nhấn nút "First" khi đang ở trang đầu tiên | Hiển thị thông báo hoặc không thực hiện thay đổ | Đáp ứng |
| 7.2 | Kiểm tra khi nhấn nút "First" khi đang ở trang không phải đầu tiên | Di chuyển đến trang đầu tiên | Đáp ứng |
| **8** | **Nút "Prev"** |  |  |
| 8.1 | Kiểm tra khi nhấn nút "Prev" khi đang ở trang đầu tiên | Hiển thị thông báo hoặc không thực hiện thay đổi | Đáp ứng |
| 8.2 | Kiểm tra khi nhấn nút "Prev" khi đang ở trang không phải đầu tiên | Di chuyển đến trang trước đó | Đáp ứng |
| **9** | **Nút "Next"** |  |  |
| 9.1 | Kiểm tra khi nhấn nút "Next" khi đang ở trang cuối cùng | Hiển thị thông báo hoặc không thực hiện thay đổi | Đáp ứng |
| 9.2 | Kiểm tra khi nhấn nút "Next" khi đang ở trang không phải cuối cùng | Kiểm tra khi nhấn nút "Next" khi đang ở trang không phải cuối cùng | Đáp ứng |
| **10** | **Nút "Last"** |  |  |
| 10.1 | Kiểm tra khi nhấn nút "Last" khi đang ở trang cuối cùng | Hiển thị thông báo hoặc không thực hiện thay đổi | Đáp ứng |
| 10.2 | Kiểm tra khi nhấn nút "Last" khi đang ở trang không phải cuối cùng | Kiểm tra khi nhấn nút "Last" khi đang ở trang không phải cuối cùng | Đáp ứng |

## 4.5 Kiểm thử quản lý khách hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **II. Sàng lọc các trường hợp thử nghiệm chung** | |  |  |
| **#** | **Hoạt động** | **Kết quả mong đợi** | **Đáp ứng hoặc không** |
| **A** | **Kiểm tra giao diện** |  |  |
| **1** | **Thuộc tính trang** |  |  |
| 1.1 | Kiểm tra chính tả tên của các thành phần đã đúng chưa | Tên của các thành phần đã đúng chính tả theo tài liệu thiết kế | Đáp ứng |
| 1.2 | Nút hình ảnh luôn chứa chú giải công cụ | Hình ảnh dưới dạng liên kết có mẹo/gợi ý thông điệp phù hợp | Đáp ứng |
| 1.3 | Kiểm tra thanh cuộn trong danh sách thông tin nhân viên | Không hiển thị thanh cuộn ngang và dọc | Không |
| **2** | **Kích thước** |  |  |
| 2.1 | Kiểm tra kích thước của các button, label, text, list,... | Kích thước đúng theo trong tài liệu thiết kế | Đáp ứng |
| **3** | **Căn chỉnh giao diện** |  |  |
| 3.1 | Kiểm tra căn chỉnh khoảng cách của các text field, button, label, list... | Khoảng cách phù hợp đúng trong tài liệu thiết kế | Đáp ứng |
| 3.2 | Kiểm tra căn chỉnh khoảng cách, vị trí của văn bản hiển thị trong list | Văn bản căn giữa | Đáp ứng |
| **4** | **Màu sắc** |  |  |
| 4.1 | Màu của giao diện có phù hợp với yêu cầu người dùng | Màu của giao diện hài hòa đúng trong tài liệu thiết kế | Đáp ứng |
| **B.** | **Kiểm tra chức năng** |  |  |
| **1** | Thêm nhà khách hàng |  |  |
| 1.1 | Kiểm tra chức năng khách hàng với dữ liệu được điền đầy đủ và đúng định dạng | Thêm thành công khách hàng | Đáp ứng |
| 1.2 | Kiểm tra chức năng thêm khách hàng với dữ liệu bỏ trống | Thêm thất bại và thông báo lỗi yêu cầu người dùng nhập mục bỏ trống | Đáp ứng |
| 1.3 | Kiểm tra chức năng thêm khách hàng với dữ liệu sai định dạng | Thêm thành công khách hàng | Không |
| 1.4 | Kiểm tra chức năng thêm khách hàngvới dữ liệu đã tồn tại trước đó | Thêm thất bại và hiển thị thông báo khách hàng đã tồn tại | Đáp ứng |
| **2** | **Cập nhật khách hàng** |  |  |
| 2.1 | Kiểm tra chức năng cập nhật với dữ liệu đầy đủ, đúng định dạng | Cập nhật thành công | Đáp ứng |
| 2.2 | Kiểm tra chức năng cập nhật với dữ liệu bỏ trống | Cập nhật thất bại, hiển thị thông báo lỗi | Đáp ứng |
| 2.3 | Kiểm tra chức năng cập nhật với dữ liệu sai định dạng | Cập nhật thành công | Không |
| **3** | **Xóa khách hàng** |  |  |
| 3.1 | Kiểm tra chức năng xóa khách hàng thành công | Hiển thị thông báo thành công và khách hàng được xóa khỏi dử liệu | Đáp ứng |
| 3.2 | Kiểm tra chức năng xóa khách hàng thất bại | Xóa không thành công nhưng không hiển thị thông báo | Không |
| 3.3 | Xóa khách hàng có chứa thông tin ở các bản khác | Xóa không thành công và hiển thị thông báo | Đáp ứng |
| **4** | **Tìm kiếm khách hàng** |  |  |
| 4.1 | Tìm khách hàng tồn tại trong dữ liệu | Hiển thị thành công khách hàng lên form | Đáp ứng |
| 4.2 | Tìm khách hàng không tồn tại | Không hiển thị khách hàng và thông báo | Đáp ứng |
| 4.3 | Tìm khách hàng thành công theo mã nhà cung cấp, họ tên, | Hiển thị thông tin khách hàng lên form | Đáp ứng |
| **5** | **Danh sách khách hàng** |  |  |
| 5.1 | Chọn khách hàng trong danh sách hiển thị lên form | Thông tin được hiển thị lên form | Đáp ứng |
| 5.2 | Thông tin khách hàng được lưu xuống thành công sau khi thêm, sửa xóa | Thông tin khách hàng được hiển thị lên danh sách | Đáp ứng |
| 5.3 | Thông tin khách hàng được lưu xuống thất bại sau khi thêm, sửa xóa | Không hiển thị thông tin khách hàng đó vào danh sách | Đáp ứng |
| **6** | Xóa form |  |  |
| 6.1 | Xóa thông tin trên form | Form được xóa trở về mặt định ban đầu | Đáp ứng |
| **7** | **Nút "First"** |  |  |
| 7.1 | Kiểm tra khi nhấn nút "First" khi đang ở trang đầu tiên | Hiển thị thông báo hoặc không thực hiện thay đổ | Đáp ứng |
| 7.2 | Kiểm tra khi nhấn nút "First" khi đang ở trang không phải đầu tiên | Di chuyển đến trang đầu tiên | Đáp ứng |
| **8** | **Nút "Prev"** |  |  |
| 8.1 | Kiểm tra khi nhấn nút "Prev" khi đang ở trang đầu tiên | Hiển thị thông báo hoặc không thực hiện thay đổi | Đáp ứng |
| 8.2 | Kiểm tra khi nhấn nút "Prev" khi đang ở trang không phải đầu tiên | Di chuyển đến trang trước đó | Đáp ứng |
| **9** | **Nút "Next"** |  |  |
| 9.1 | Kiểm tra khi nhấn nút "Next" khi đang ở trang cuối cùng | Hiển thị thông báo hoặc không thực hiện thay đổi | Đáp ứng |
| 9.2 | Kiểm tra khi nhấn nút "Next" khi đang ở trang không phải cuối cùng | Kiểm tra khi nhấn nút "Next" khi đang ở trang không phải cuối cùng | Đáp ứng |
| **10** | **Nút "Last"** |  |  |
| 10.1 | Kiểm tra khi nhấn nút "Last" khi đang ở trang cuối cùng | Hiển thị thông báo hoặc không thực hiện thay đổi | Đáp ứng |
| 10.2 | Kiểm tra khi nhấn nút "Last" khi đang ở trang không phải cuối cùng | Kiểm tra khi nhấn nút "Last" khi đang ở trang không phải cuối cùng | Đáp ứng |

## 4.6 Kiểm thử sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| **II. Sàng lọc các trường hợp thử nghiệm chung** | |  |  |
| **#** | **Hoạt động** | **Kết quả mong đợi** | **Đáp ứng hoặc không** |
| **A** | **Kiểm tra giao diện** |  |  |
| **1** | **Thuộc tính trang** |  |  |
| 1.1 | Kiểm tra chính tả tên của các thành phần đã đúng chưa | Tên của các thành phần đã đúng chính tả theo tài liệu thiết kế | Đáp ứng |
| 1.2 | Nút hình ảnh luôn chứa chú giải công cụ | Hình ảnh dưới dạng liên kết có mẹo/gợi ý thông điệp phù hợp | Đáp ứng |
| 1.3 | Kiểm tra thanh cuộn trong danh sách thông tin nhân viên | Hiển thị thanh cuộn ngang và dọc | Đáp ứng |
| **2** | **Kích thước** |  |  |
| 2.1 | Kiểm tra kích thước của các button, label, text, list,... | Kích thước đúng theo trong tài liệu thiết kế | Đáp ứng |
| **3** | **Căn chỉnh giao diện** |  |  |
| 3.1 | Kiểm tra căn chỉnh khoảng cách của các text field, button, label, list... | Khoảng cách phù hợp đúng trong tài liệu thiết kế | Đáp ứng |
| 3.2 | Kiểm tra căn chỉnh khoảng cách, vị trí của văn bản hiển thị trong list | Văn bản căn giữa | Đáp ứng |
| **4** | **Màu sắc** |  |  |
| 4.1 | Màu của giao diện có phù hợp với yêu cầu người dùng | Màu của giao diện hài hòa đúng trong tài liệu thiết kế | Đáp ứng |
| **B.** | **Kiểm tra chức năng** |  |  |
| **1** | **Thêm sản phẩm** |  |  |
| 1.1 | Kiểm tra chức năng thêm sản phẩm với dữ liệu được điền đầy đủ và đúng định dạng | Thêm thành công sản phẩm | Đáp ứng |
| 1.2 | Kiểm tra chức năng thêm sản phẩm với dữ liệu bỏ trống | Thêm thất bại và thông báo lỗi yêu cầu người dùng nhập mục bỏ trống | Đáp ứng |
| 1.3 | Kiểm tra chức năng thêm sản phẩm với dữ liệu sai định dạng | Thêm thất bại và thông báo lỗi yêu cầu người dùng nhập đúng định dạng | Đáp ứng |
| 1.4 | Kiểm tra chức năng thêm sản phẩm với dữ liệu đã tồn tại trước đó | Thêm thất bại và hiển thị thông báo nhân viên đã tồn tại | Đáp ứng |
| **2** | **Cập nhật sản phẩm** |  |  |
| 2.1 | Kiểm tra chức năng cập nhật với dữ liệu đầy đủ, đúng định dạng | Cập nhật thành công | Đáp ứng |
| 2.2 | Kiểm tra chức năng cập nhật với dữ liệu bỏ trống | Cập nhật thất bại, hiển thị thông báo lỗi yêu cầu nhập mục bỏ trống | Đáp ứng |
| 2.3 | Kiểm tra chức năng cập nhật với dữ liệu sai định dạng | Cập nhật thất bại, hiển thị thông báo lỗi yêu cầu nhập đúng định dạng | Đáp ứng |
| **3** | **Xóa sản phẩm** |  |  |
| 3.1 | Kiểm tra chức năng xóa sản phẩm thành công | Hiển thị thông báo thành công và nhân viên được xóa khỏi dử liệu | Đáp ứng |
| 3.2 | Kiểm tra chức năng xóa sản phẩm thất bại | Xóa không thành công hiển thị thông báo xóa thất bại | Đáp ứng |
| 3.3 | Xóa sản phẩm có chứa thông tin ở các bản khác | Xóa không thành công và hiển thị thông báo | Đáp ứng |
| **4** | **Tìm kiếm sản phẩm** |  |  |
| 4.1 | Tìm sản phẩm tồn tại trong dữ liệu | Hiển thị thành công nhân viên lên form | Đáp ứng |
| 4.2 | Tìm sản phẩm không tồn tại | Không hiển thị nhân viên và thông báo | Đáp ứng |
| 4.3 | Tìm sản phẩm theo tên sản phẩm | Hiển thị thông tin nhân viên lên form | Đáp ứng |
| **5** | **Danh sách sản phẩm** |  |  |
| 5.1 | Chọn sản phẩm trong danh sách hiển thị lên form | Thông tin được hiển thị lên form | Đáp ứng |
| 5.2 | Thông tin sản phẩm được lưu xuống thành công sau khi thêm, sửa xóa | Thông tin nhân viên được hiển thị lên danh sách | Đáp ứng |
| 5.3 | Thông tin sản phẩm được lưu xuống thất bại sau khi thêm, sửa xóa | Không hiển thị thông tin nhân viên đó vào danh sách | Đáp ứng |
| **6** | Xóa form |  |  |
| 6.1 | Xóa thông tin trên form | Form được xóa trở về mặt định ban đầu | Đáp ứng |
| **7** | **Nút "First"** |  |  |
| 7.1 | Kiểm tra khi nhấn nút "First" khi đang ở trang đầu tiên | Hiển thị thông báo hoặc không thực hiện thay đổ | Đáp ứng |
| 7.2 | Kiểm tra khi nhấn nút "First" khi đang ở trang không phải đầu tiên | Di chuyển đến trang đầu tiên | Đáp ứng |
| **8** | **Nút "Prev"** |  |  |
| 8.1 | Kiểm tra khi nhấn nút "Prev" khi đang ở trang đầu tiên | Hiển thị thông báo hoặc không thực hiện thay đổi | Đáp ứng |
| 8.2 | Kiểm tra khi nhấn nút "Prev" khi đang ở trang không phải đầu tiên | Di chuyển đến trang trước đó | Đáp ứng |
| **9** | **Nút "Next"** |  |  |
| 9.1 | Kiểm tra khi nhấn nút "Next" khi đang ở trang cuối cùng | Hiển thị thông báo hoặc không thực hiện thay đổi | Đáp ứng |
| 9.2 | Kiểm tra khi nhấn nút "Next" khi đang ở trang không phải cuối cùng | Kiểm tra khi nhấn nút "Next" khi đang ở trang không phải cuối cùng | Đáp ứng |
| **10** | **Nút "Last"** |  |  |
| 10.1 | Kiểm tra khi nhấn nút "Last" khi đang ở trang cuối cùng | Hiển thị thông báo hoặc không thực hiện thay đổi | Đáp ứng |
| 10.2 | Kiểm tra khi nhấn nút "Last" khi đang ở trang không phải cuối cùng | Kiểm tra khi nhấn nút "Last" khi đang ở trang không phải cuối cùng | Đáp ứng |

## 4.7 Kiểm thử hóa đơn

|  |  |  |  |
| --- | --- | --- | --- |
| **II. Sàng lọc các trường hợp thử nghiệm chung** | |  |  |
| **#** | **Hoạt động** | **Kết quả mong đợi** | **Đáp ứng hoặc không** |
| **A** | **Kiểm tra giao diện** |  |  |
| **1** | **Thuộc tính trang** |  |  |
| 1.1 | Kiểm tra chính tả tên của các thành phần đã đúng chưa | Tên của các thành phần đã đúng chính tả theo tài liệu thiết kế | Đáp ứng |
| 1.2 | Nút hình ảnh luôn chứa chú giải công cụ | Hình ảnh dưới dạng liên kết có mẹo/gợi ý thông điệp phù hợp | Đáp ứng |
| 1.3 | Kiểm tra thanh cuộn trong danh sách thông tin nhân viên | Hiển thị thanh cuộn ngang và dọc | Đáp ứng |
| **2** | **Kích thước** |  |  |
| 2.1 | Kiểm tra kích thước của các button, label, text, list,... | Kích thước đúng theo trong tài liệu thiết kế | Đáp ứng |
| **3** | **Căn chỉnh giao diện** |  |  |
| 3.1 | Kiểm tra căn chỉnh khoảng cách của các text field, button, label, list... | Khoảng cách phù hợp đúng trong tài liệu thiết kế | Đáp ứng |
| 3.2 | Kiểm tra căn chỉnh khoảng cách, vị trí của văn bản hiển thị trong list | Văn bản căn giữa | Đáp ứng |
| **4** | **Màu sắc** |  |  |
| 4.1 | Màu của giao diện có phù hợp với yêu cầu người dùng | Màu của giao diện hài hòa đúng trong tài liệu thiết kế | Đáp ứng |
| **B.** | **Kiểm tra chức năng** |  |  |
| **1** | **Thêm hóa đơn** |  |  |
| 1.1 | Kiểm tra chức năng hiển thị màn hình thêm hóa đơn | Màn hình thêm hóa đơn được hiển thị | Đáp ứng |
| 1.2 | Kiểm tra chức năng thêm hóa đơn với dữ liệu được điền đầy đủ và đúng định dạng | Thêm thành công sản phẩm | Đáp ứng |
| 1.3 | Kiểm tra chức năng thêm hóa đơn với dữ liệu bỏ trống | Thêm thất bại và thông báo lỗi yêu cầu người dùng nhập mục bỏ trống | Đáp ứng |
| 1.4 | Kiểm tra chức năng thêm hóa đơn với dữ liệu sai định dạng | Thêm thất bại và thông báo lỗi yêu cầu người dùng nhập đúng định dạng | Đáp ứng |
| 1.5 | Kiểm tra chức năng thêm hóa đơn với dữ liệu đã tồn tại trước đó | Thêm thất bại và hiển thị thông báo nhân viên đã tồn tại | Đáp ứng |
| **2** | **Cập nhật hóa đơn** |  |  |
|  | Kiểm tra màn hình hiển thị cập nhật hóa đơn nhưng chưa chọn hóa đơn trên danh sách | Màn hình chập nhật hóa đơn không được hiển thị và thông báo lỗi yêu cầu chọn hóa đơn | Đáp ứng |
|  | Kiểm tra màn hình hiển thị cập nhật hóa đơn đã chọn hóa đơn trên danh sách với trạng thái chưa thanh toán | Màn hình chập nhật hóa đơn được hiển thị | Đáp ứng |
|  | Kiểm tra màn hình hiển thị cập nhật hóa đơn đã chọn hóa đơn trên danh sách với trạng thái đã thanh toán | Màn hình chập nhật hóa đơn không hiển thị và thông báo hóa đơn đã thanh toán | Đáp ứng |
| 2.1 | Kiểm tra chức năng cập nhật với dữ liệu đầy đủ, đúng định dạng | Cập nhật thành công | Đáp ứng |
| 2.2 | Kiểm tra chức năng cập nhật với dữ liệu bỏ trống | Cập nhật thất bại, hiển thị thông báo lỗi yêu cầu nhập mục bỏ trống | Đáp ứng |
| 2.3 | Kiểm tra chức năng cập nhật với dữ liệu sai định dạng | Cập nhật thất bại, hiển thị thông báo lỗi yêu cầu nhập đúng định dạng | Đáp ứng |
| **3** | **Hủy đơn** |  |  |
| 3.1 | Kiểm tra chức năng hủy đơn nhưng không chọn đơn hàng | Hiển thị thông báo lỗi và yêu cầu chọn đơn hàng | Đáp ứng |
|  | Kiểm tra chức năng hủy đơn đã chọn đơn hàng với trạng thái chưa thanh toán | Hủy đơn hàng thành công | Đáp ứng |
| 3.2 | Kiểm tra chức năng hủy đơn với trạng thái đã thanh toán | Màn hình chập nhật hóa đơn không hiển thị và thông báo hóa đơn đã thanh toán | Đáp ứng |
| **4** | **Thanh toán** |  |  |
| 4.1 | Kiểm tra chức năng thanh toán nhưng chưa chọn hóa đơn | Hiển thị thông báo chưa chọn hóa đơn | Đáp ứng |
| 4.2 | Kiểm tra chức năng thanh toán đã chọn hóa đơn với trạng thái chưa thanh toán | Hiển thị thông báo thanh toán thành công và cập nhật trạng thái đã thanh toán | Đáp ứng |
| **5** | **Tìm kiếm hóa đơn** |  |  |
| 5.1 | Tìm hóa đơn tồn tại trong dữ liệu | Hiển thị thành công hóa đơn trên danh sách | Đáp ứng |
| 5.2 | Tìm hóa đơn không tồn tại | Không hiển thị thông tin lên danh sách | Đáp ứng |
| 5.3 | Tìm hóa đơn theo tên khách hàng | Hiển thị thông tin trên danh sách hóa đơn | Đáp ứng |
| **6** | **Danh sách hóa đơn** |  |  |
| 6.2 | Thông tin hóa đơn được lưu xuống thành công sau khi cập nhật | Thông tin hóa đơn được hiển thị lên danh sách | Đáp ứng |
| **7** | **Tạo mới** |  |  |
| 7.1 | Xóa thông tin trên form | Form được xóa trở về mặt định ban đầu | Đáp ứng |

## 4.8 Kiểm thử thống kê

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Hoạt động** | **Kết quả mong đợi** | **Đáp ứng hoặc không** |
| **A** | **Kiểm tra giao diện** |  |  |
| **1** | **Thuộc tính trang** |  |  |
| 1.1 | Kiểm tra xem bố cục có phù hợp với màn hình có độ phân giải tối thiểu được xác định không | Màn hình hiển thị không phù hợp | Không |
| 1.2 | Kiểm tra tiêu đề (chính tả) của các thành phần | Tiêu đề đúng chính tả ở tất cả các ngôn ngữ của ứng dụng theo tài liệu thiết kế | Đáp ứng |
| 1.3 | Kiểm tra hình ảnh icon giao diện | Hình ảnh rõ nét , đúng vị trí |  |
| 1.4 | Kiểm tra giao diện thu nhỏ | Không có thanh cuộn ngang. Không có thanh cuộn dọc | Không |
| **2** | **Kích thước** |  |  |
| 2.1 | Kiểm tra icon, font size, font style, font color của các text trên màn hình | Kích thước của:  phông chữ -> như trong tài liệu thiết kế  những cái khác -> phải phù hợp (đẹp) theo bố cục | Đáp ứng |
| **3** | **Căn chỉnh của mục** |  |  |
| 3.1 | Kiểm tra căn chỉnh khoảng cách của các text field, button, label,... | Khoảng cách phù hợp đúng trong tài liệu thiết kế | Đáp ứng |
| 3.2 | Kiểm tra căn chỉnh các trường trong bố cục | Tất cả các trường phải được căn chỉnh với nhau | Đáp ứng |
| **4** | **Màu sắc** |  |  |
| 4.1 | Màu của giao diện có phù hợp với yêu cầu người dùng | Màu của giao diện hài hòa đúng trong tài liệu thiết kế | Đáp ứng |
| **B** | **Kiểm tra chức năng** |  |  |
| **1** | **Hiển thị dữ liệu** |  |  |
| 1.1 | Kiểm tra dữ liệu hiển thị lên bảng | Hiển thị đầy đủ thông tin trong dử liệu | Đáp ứng |
| 1.2 | Kiểm tra dữ liệu đã hiển thị đúng các trường trong bảng | Dữ liệu được đưa vào đúng các trường trong bảng | Đáp ứng |
| 1.3 | Kiểm tra thông tin trên bảng có thay đổi sau khi dự liệu được thêm mới | Thông tin trên bảng thay đổi sau khi thêm mới | Đáp ứng |
| 1.4 | Kiểm tra thông tin trên bảng có thay đổi sau khi dự liệu được chỉnh sửa | Thông tin trên bảng thay đổi sau khi chỉnh sửa | Đáp ứng |
| 1.5 | Kiểm tra thông tin trên bảng có thay đổi sau khi dữ liệu được xóa | Thông tin trên bảng không thay đổi sau khi xóa | Không |
| 1.6 | Kiểm tra tính năng xuất file dạng excel | Xuất file thành công | Đáp ứng |

# 5. ĐÓNG GÓI VÀ TRIỂN KHAI

## 5.1 Dẫn chuyển đổi jar thành exe

##### Sử dụng phần mềm EXE4J để chuyển đổi jar sang exe

* + - Sử dụng phần mềm InnoSetup để đóng gói

## 5.2 Hướng dẫn cài đặt triển khai

##### Cài SQL Server 2020

* + - Tạo CSDL Polypro bằng cách chạy file Polypro.sql
    - Cài đặt JDK 17
    - Chạy file setup

## 5.3 Hướng dẫn sử dụng phần mềm

##### Đăng nhập với tài khoản username: admin và password:admin

* + - Cấp các tài khoản có vai trò là quản lý

##### Đăng nhập tài khoản mới và xóa tài khoản admin

* + - Xem hướng dȁn sử dụng trong menu trợ giúp của phần mềm