**TRƯỜNG CAO ĐẲNG FPT POLY TECHNICH**



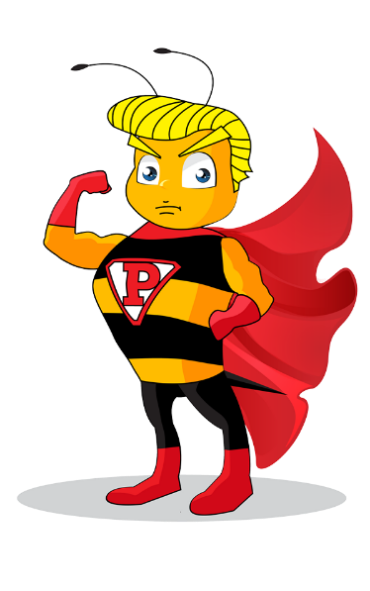
**BÁO CÁO THỰC HÀNH ASSIGMENT MÔN DỰ ÁN 1**

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

|  |
| --- |
| GIẢNG VIÊN HƯỚNG DẪN :  HUỲNH KHẮC DUY |
| SINH VIÊN THỰC HIỆN |
| BÙI NGUYỄN DUY NGUYÊN : PS24918 |

|  |
| --- |
|  |

**Đề tài : Xây dựng ứng dụng phần mềm Quản lí bán hàng**

****

|  |
| --- |
| LỜI NÓI ĐẦU  Chúng ta đang sống trong thời kì công nghệ 4.0, cuộc cách mạng khoa học công nghệ và khoa học kĩ thuật hiện đại can thiệp ngày càng sâu vào đời sống của con người. Nhờ có cuộc cách mạng công nghệ ấy mà, đời sống con người ngày càng được cải thiện về mọi mặt.  "Cách mạng Công nghiệp 4.0" đang diễn ra tại nhiều nước phát triển. Nó mang đến cho nhân loại cơ hội để thay đổi bộ mặt các nền kinh tế.  Ngày nay, Công nghệ thông tin có những bước phát triển mạnh mẽ, ứng dụng trong nhiều lĩnh vực đời sống và hiệu quả mang lại không thể phủ nhận, giúp giảm thiểu tối đa các khâu làm việc thủ công, kém hiệu quả, không chính xác của con người trước kia. Chính vì thế máy tính điện tử trở nên phổ biến và gần gũi với mọi người, việc tiếp cận với hệ thống |

MỤC LỤC

[1 Phân tích 3](#_Toc520132211)

[1.1 Hiện trạng 3](#_Toc520132212)

[1.2 Use case 6](#_Toc520132214)

[2 Thiết kế 7](#_Toc520132215)

[2.1 Liệt kê cac thực thể quan hệ 7](#_Toc520132216)

[2.2 Mô hình triển khai 8](#_Toc520132216)

[2.3 Thiết kế CSDL 8](#_Toc520132217)

[2.3.1 Sơ đồ quan hệ thực thể 8](#_Toc520132218)

[2.3.2 Thiết kế chi tiết các thực thể 9](#_Toc520132219)

[2.4 Thiết kế giao diện 11](#_Toc520132220)

[2.4.1 Sơ đồ tổ chức giao diện 11](#_Toc520132221)

[2.4.2 Thiết kế giao diện cho các chức năng nghiệp vụ 11](#_Toc520132222)

[3 Thực hiện viết mã 2](#_Toc520132223)8

[3.1 Viết mã tạo CSDL 2](#_Toc520132224)8

[3.1.1 Tạo CSDL 2](#_Toc520132225)8

[3.1.2 SQL truy vấn và thao tác 30](#_Toc520132226)

[3.1.3 Các thủ tục lưu tổng hợp thống kê 32](#_Toc520132227)

[3.2 Lập trình JDBC 35](#_Toc520132228)

[3.2.1 Lớp hỗ trợ 35](#_Toc520132229)

[3.2.2 Model class - Các lớp mô tả dữ liệu 42](#_Toc520132230)

[3.2.3 DAO Class - Các lớp truy xuất dữ liệu 44](#_Toc520132231)

[3.3 Viết mã cho ứng dụng 55](#_Toc520132232)

[3.3.1 Xử lý Form Chao\_Dialog 55](#_Toc520132233)

[3.3.2 Xử lý Form Login\_Dialog 56](#_Xử_lí_Form)

[3.3.3 Xử lý Form QuenMatKhau\_Dialog 57](#_Xử_lí_Form_1)

[3.3.4 Xử lý Form MainFrame 59](#_Toc520132234)

[3.3.5 Xử lý Form Home\_panel 60](#_Xử_lí_Form_2)

[3.3.6 Xử lý Form KhachHang\_panel 61](#_Xử_lí_Form_3)

[3.3.7 Xử lý Form NhanVien\_panel 63](#_Xử_lí_Form_4)

[3.3.8 Xử lý Form SanPham\_panel 63](#_Xử_lí_Form_5)

[3.3.9 Xử lý Form HoaDon\_panel 63](#_Xử_lí_Form_6)

[3.3.10 Xử lý Form ThongKe\_Dialog 64](#_Xử_lí_Form_7)

[4 Kiểm thử 66](#_Toc520132235)

[5 Đóng gói và triển khai 73](#_Toc520132238)

[5.1 Hướng dẫn chuyển đổi jar thành exe 73](#_Toc520132239)

[5.2 Hướng dẫn cài đặt triển khai 74](#_Toc520132240)

[5.3 Hướng dẫn sử dụng phần mềm 75](#_Toc520132241)

[6 Kết luận và hướng phát triển 76](#_Kết_luận)

[6.1 Kết luận 76](#_Hướng_phát_triển)

[6.2 Hướng phát triển 76](#_Kết_luận_1)

Phân tích

## Hiện trạng

### Thực trạng :

Trong thời đại công nghệ thông tin phát triển mạnh mẽ như vũ bão thì việc áp dụng các Phần mềm công nghệ được nhiều doanh nghiệp hay tư nhân sử dụng như một giải pháp quản lí tổng thể rất hiệu quả. Quản lý bán hàng là công tác quan trọng trong quá trình hoạt động của doanh nghiệp nhằm kiểm soát các hoạt động bán hàng, để dễ dàng và tiết kiệm chi phí nhiều doanh nghiệp đã sử dụng phần mềm quản lý bán hàng.

Cửa hàng bán đồ thể thao Sun Store là một cửa hàng mới được khai trương. Với lượng khách đến mua hàng ngày càng tăng, và gặp khó khăn trong việc quản lí sản phẩm, nhân viên cũng như khách hàng. Để phục vụ khách hàng được tốt hơn, chính xác và nhanh chóng hơn thì chủ cửa hàng muốn từng bước tin học hóa các khâu quản lí. Đặc biệt là trong công tác kế toán và quản lí cửa hàng. Bởi vì công tác thủ công mà cửa hàng đang thực hiện đã bộc lộ nhiều hạn chế như sau :

### HẠN CHẾ :

-Tra cứu thông tin sản phẩm, các nguồn hàng cung cấp cho cửa hàng và khách hàng mất nhiều thời gian và nhiều khi không chính xác.

-Lưu trữ thông tin nhập xuất hàng hóa, về thu chi, hóa đơn cần nhiều loại giấy tờ nên cồng kềnh và không đạt hiệu quả.

-Cập nhật thông tin sản phẩm hằng ngày tốn nhiều thời gian và công sức. Khó khăn trong việc thực hiện thống kê, báo cáo .

### Mục đích :

Trước những hạn chế đó, vấn đề được đặt ra là xây dụng một phần mềm quản lí đáp ứng các yêu cầu cơ bản sau :

-Giảm khối lượng ghi chép nhằm lưu trũ thông tin.

-Cập nhật dữ liệu sản phẩm nhanh chóng, chính xác.

-Xuất hóa đơn nhanh chóng khi khách hàng đến mua hàng.

-Có khả năng lưu trữ thông tin(sản phẩm, nhân viên, hóa đơn…) lâu dài, đảm bảo truy vấn nhanh khi cần thiết

### Đối tượng sử dụng hệ thống :

YÊU CẦU XÂY DỰNG MỘT HỆ THỐNG QUẢN LÍ BÁN HÀNG CỦA SUN STORE :

-Người quản lí : nắm được tình hình mua bán,quản lí sản phẩm, nhân viên của cửa hàng, doanh thu của cửa hàng theo ngày, tháng. Việc thống kê được thực hiện qua từng tháng.

- Nhân viên bán hàng : sẽ thực hiện tính tiền những sản phẩm mà khách hàng mua và lập hóa đơn cho khách, quản lí danh mục các sản phẩm của cửa hàng.

-Khách hàng : là người mua hàng ở tại cửa hàng.

### Nghiệp vụ hằng ngày :

- Người quản lí trực tiếp liên hệ với nhà sản xuất để nhập sản phẩm về cửa hàng, cũng như quan hệ đối tác để bán sản phẩm ra ngoài. Người quản lí sẽ quản lí thông tin khách hàng,sản phẩm, nhân viên của cửa hàng. Thống kê doanh thu của cửa hàng hằng ngày, hằng tháng.

- Nhân viên tại cửa hàng làm việc theo ca, khi đến ca làm việc của mình, nhân viên phải đăng nhập vào hệ thống. Khi khách hàng lựa chọn mua sản phẩm và yêu cầu tính tiền thì nhân viên sẽ tính tiền và lập hóa đơn cho khách.

- Khách hàng đến mua hàng có thể đăng kí thành viên để nhận được ưu đãi của cửa hàng khi đến mua lần sản phẩm tiếp theo.

### Yêu cầu hệ thống để sử dụng phần mềm

#### Yêu cầu phần cứng :

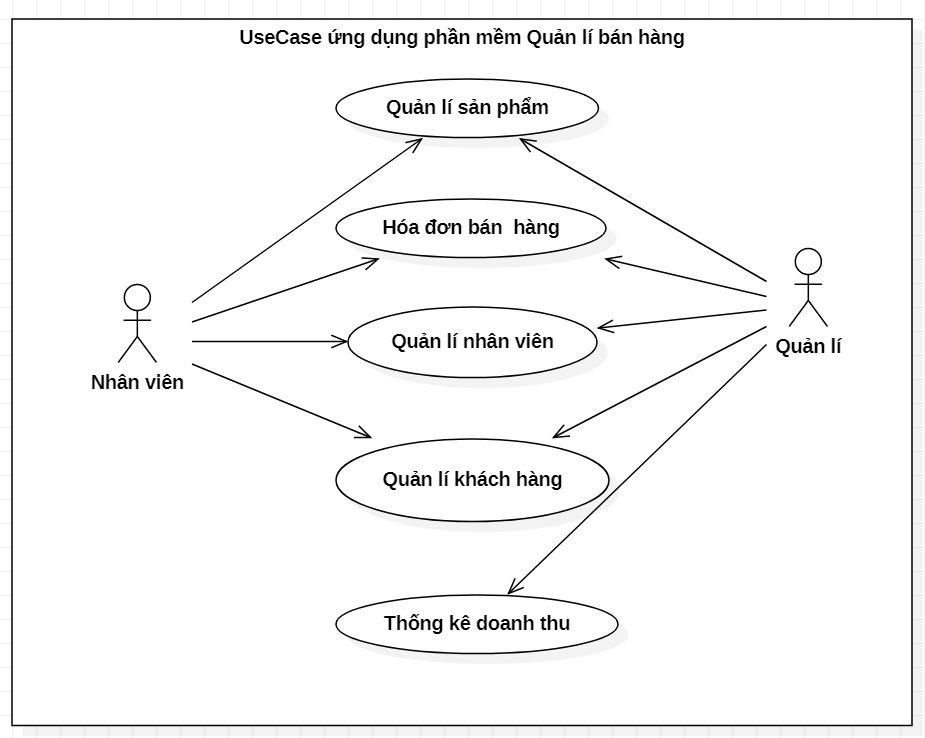
* Bộ vi xử lý(CPU): Intel Dual Core hoặc cao hơn.
* Bộ nhớ trong (RAM): 1GB trở lên.
* Đĩa cứng (HDD): 2GB trống hoặc nhiều hơn.
* Màn hình: Độ phân giải 1024×768 hoặc cao hơn.

#### Yêu cầu phần mềm :

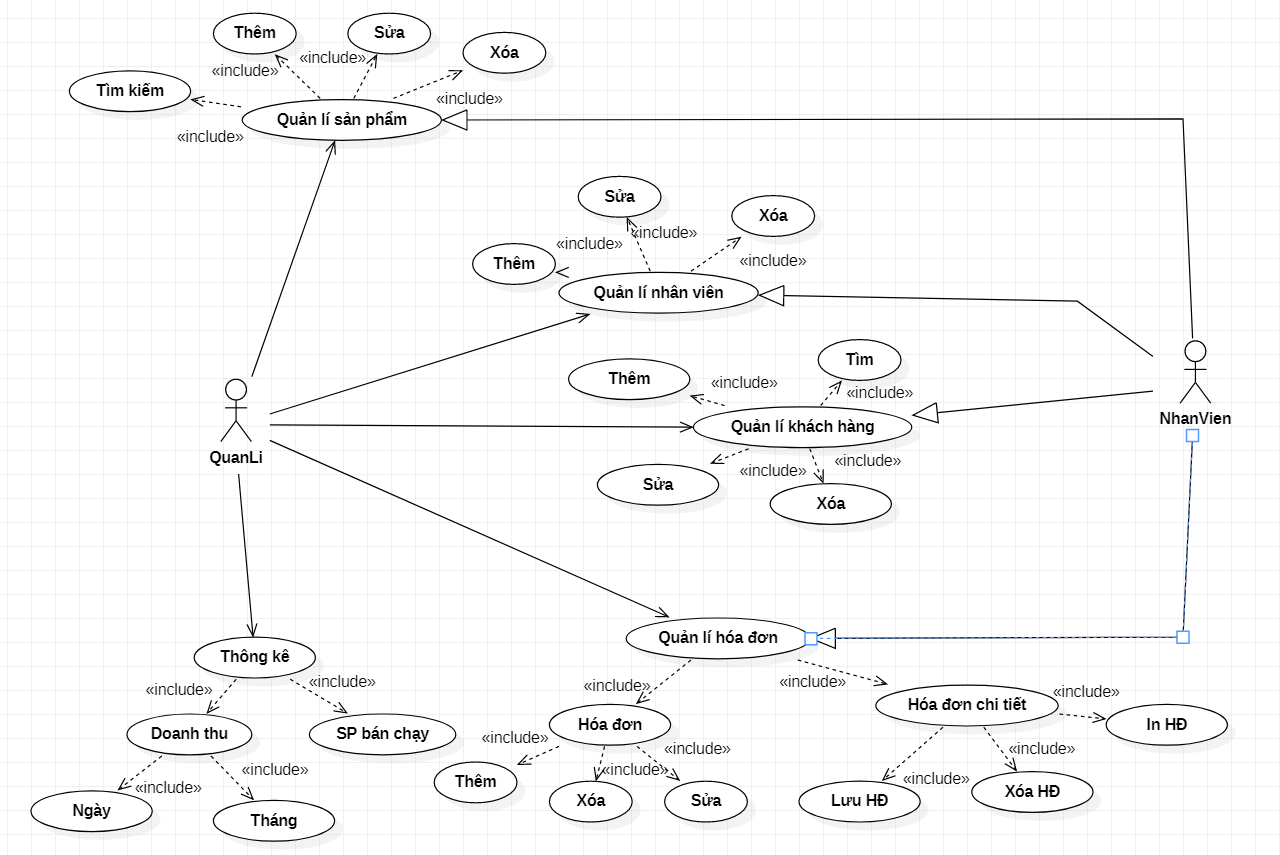
* Hệ điều hành: Windows 7/10 hoặc cao hơn.
* Bộ gõ tiếng Việt unicode: Unikey, VietKey.
* Cần một máy cài SQL Server 2008 hoặc mới hơn.

## Use case

### Sơ đồ Usecase mức 1:



### Sơ đồ Usecase mức 2:



# Thiết kế

## Liệt kê các thực thể quan hệ

Trong phần mềm quản lí bán hàng như trên ta có các thực thể :

+Khách hàng : lưu trữ thông tin khách hàng đến mua sản phẩm của cửa hàng

+Nhân viên : trực tiếp quản lí cửa hàng, thanh toán cho khách hàng khi mua sản phẩm

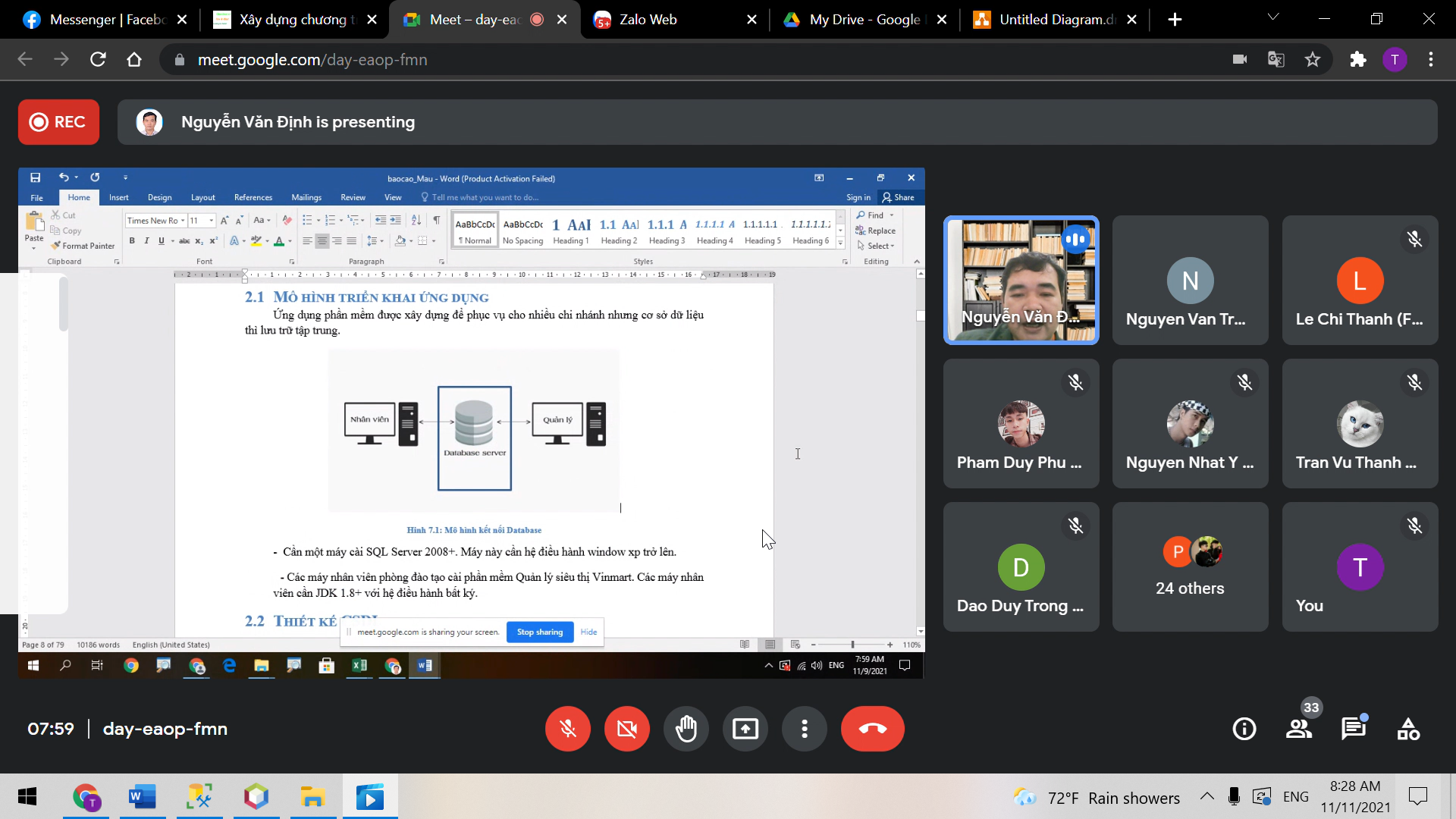
+Danh mục sản phẩm : mỗi sản phẩm sẽ ứng theo mỗi loại danh mục

+Sản phẩm : lưu trữ thông tin tất cả sản phẩm hiện có của cửa hàng

+Hóa đơn : khi khách thanh toán sản phẩm thì nhân viên sẽ lập hóa đơn cho khách hàng

+Hóa đơn chi tiết : lưu trữ thông tin chi tiết hóa đơn mua hàng (Sản phẩm, đơn giá, số lượng, thành tiền….)

## Mô hình triển khai



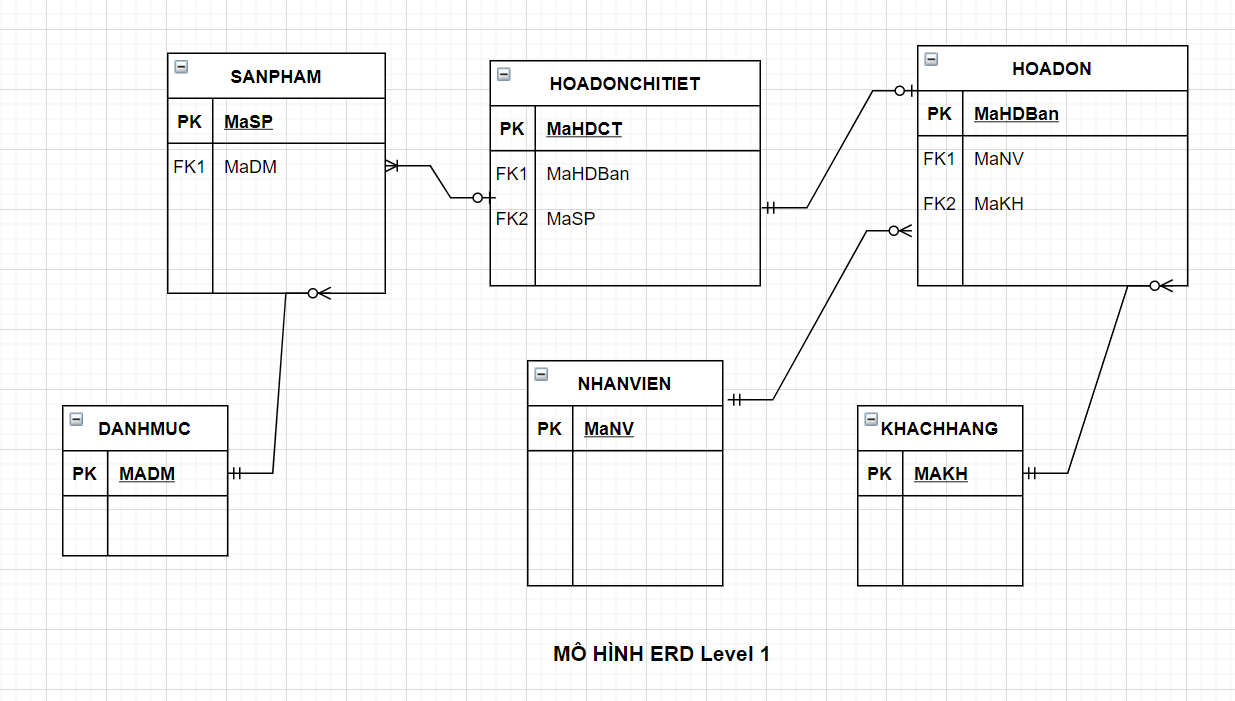
-Cần một máy cài SQL Server 2008 hoặc mới hơn. Máy này cần hệ điều hành Window XP trở lên.

-Các máy nhân viên của cửa hàng Nike Store cài phần mềm Quản lí bán hàng Nike Store. Các máy nhân viên cần có JDK 1.8 với hệ điều hành bất kì.

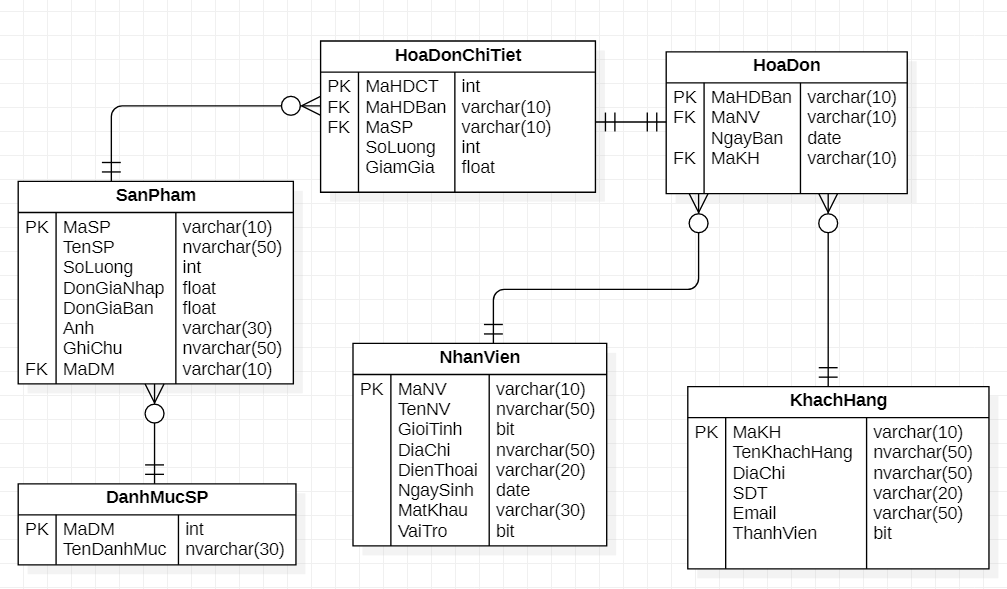
## Thiết kế CSDL

### Sơ đồ quan hệ thực thể

#### ERD Level 1 :



#### ERD Level 2 :



### Thiết kế chi tiết các thực thể

#### .Danh mục sản phẩm :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MaDM | varchar(10) | int, not null, indenty | Mã danh mục sản phẩm |
| TenDanhMuc | nvarchar(30) | Not null | Tên danh mục sản phẩm |

#### Sản phẩm :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MaSP | varchar(10) | PK, not null | Mã sản phẩm |
| TenSP | nvarchar(30) | Not null | Tên sản phẩm |
| SoLuong | int | Null | Số lượng sản phẩm |
| GiaNhap | float | Null | Đơn giá nhập sản phẩm |
| GiaBan | float | Null | Đơn giá bán sản phẩm |
| Anh | varchar(50) | Null | Ảnh sản phẩm |
| GhiChu | nvarchar(50) | Null | Ghi chú |
| MaDM | varchar(10) | FK, not null | Mã danh mục sản phẩm |

#### Nhân viên:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MaNV | varchar(10) | PK, not null | Mã nhân viên, tên đăng nhập |
| TenNV | nvarchar(30) | Not null | Tên nhân viên |
| GioiTinh | bit | Null | Giới tính |
| DiaChi | nvarchar(50) | Null | Địa chỉ |
| SDT | varchar(20) | Null | Số điện thoại NV |
| NgaySinh | date | Null | Ngày sinh |
| MatKhau | varcha(50) | Null | Mật khẩu |
| VaiTro | bit | Not null | Vai trò |

#### Khách hàng :

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| MaKH | varchar(10) | PK, not null | Mã khách hàng |
| TenKH | nvarchar(30) | Not null | Tên khách hàng |
| DiaChi | nvarchar(50) | Null | Địa chỉ |
| SDT | varchar(20) | Null | Số điện thoại KH |
| Email | varchar(40) | Null | Địa chỉ email |
| ThanhVien | bit | Null | Thành viên ? |

#### Hóa đơn:

|  |  |  |  |
| --- | --- | --- | --- |
| MaHDBan | varchar(10) | PK, not null | Mã hóa đơn |
| MaNV | varchar(10) | FK, not null | Mã nhân viên |
| NgayBan | date | Null | Ngày bán |
| MaKH | varchar(10) | FK, not null | Mã khách hàng |

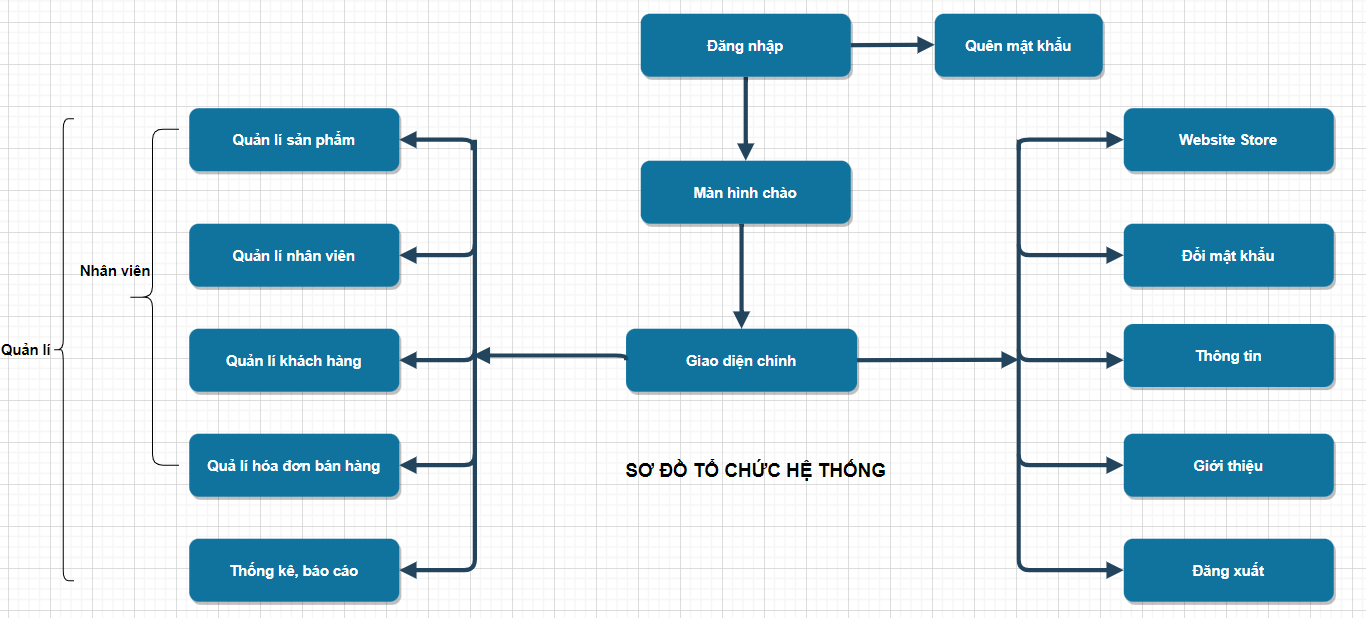
#### Hóa đơn chi tiết :

|  |  |  |  |
| --- | --- | --- | --- |
| MaHDCT | int | FK, not null, indentity | Mã chi tiết hóa đơn |
| MaHDBan | varchar(10) | FK, not null | Mã hóa đơn |
| MaSP | varchar(10) | FK, not null | Mã sản phẩm |
| SoLuong | int | Null | Số lượng sản phẩm |
| GiamGia | float | Null | Giảm giá |

## Thiết kế giao diện

### Sơ đồ tổ chức giao diện

Giao diện chính là một cửa sổ chứa menu chính và list menu. Thông qua đó để đi đến các giao diện thành phần 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. Vì thế nên form đăng nhập xuất hiện trước để yêu cầu đăng nhập. Ngoài ra mỗi ứng dụng trong thời gian khởi động cần có một màn hình chào cùng với thanh tiến trình để người có cảm giác ứng dụng đang khởi động



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

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 bấm |
| JTextField,JTextArea, JTextPassword,… | txt | Các loại ô nhập |
| JRadioButton | rdo | Ô chọn |
| ButtonGroup | bgr | Nhóm các rdo |
| JComboBox | cbo | Hộp chọn đổ xuống |
| JMenu | mnu | Thực đơn |
| JMenuItem | mni | Mục menu |
| JPanel | pnl | Khung chứa |
| JTabbedPane | tab | Tab |
| JTable | tb | Bảng |
| JFileChooser | fcs | Hộp chọn file |
| JCheckBox | chk | Hộp kiểm tra |
| ….. |  |  |

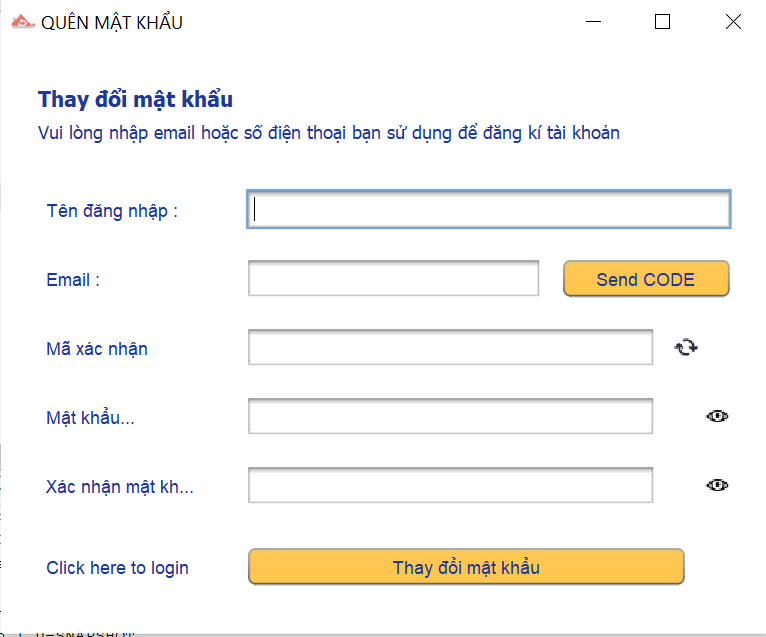
#### Thiết kế Form Đăng nhập



Các thành phần giao diện :

|  |  |  |
| --- | --- | --- |
| **Componel** | **Kiểu** | **Thuộc tính** |
| Login\_Dialog | Jdiog | Title : Login  defaultCloseOperation: DO\_NOTHING |
| lblLogo | Jlabel | Icon : 2250207.png |
| Pnl Form | Panel | Backgroud color : 255,255,255 |
| lblManv | JtextField | Text : Tên đăng nhập |
| lblMatKhau | JtextField | Text : Mật khẩu |
| txtManv | JtextField |  |
| txtMatKhau | JtextPassword |  |
| lblQuenMatKhau | Jlabel | Text : Quên Mật Khẩu ? |
| btnDangNhap | Button | Text : Đăng nhập |
| btnKetThuc | Button | Text : Kết thúc |
| chkShow | Jcheck |  |

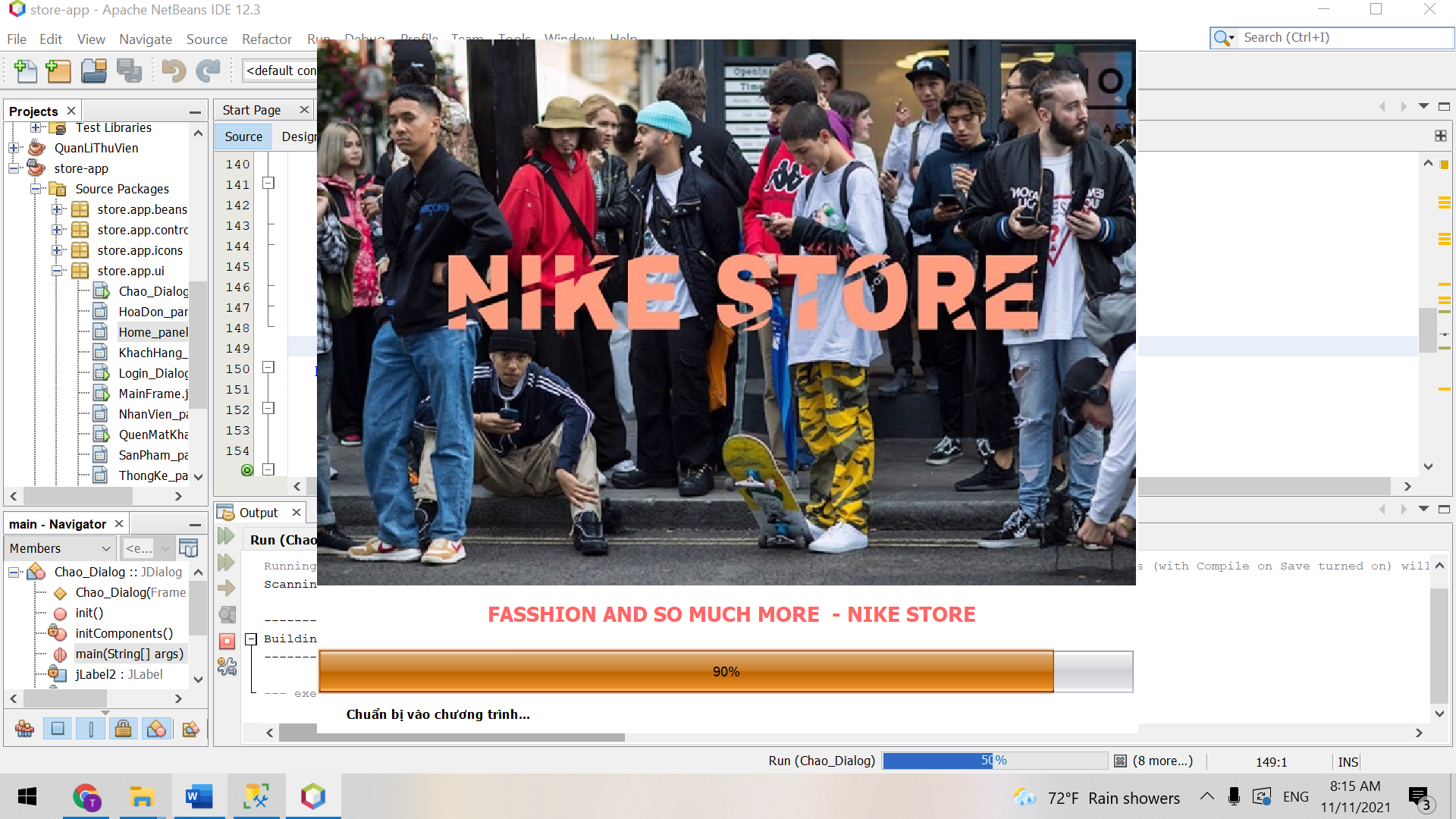
#### Thiết kế Form Quên mật khẩu



Các thành phần giao diện :

|  |  |  |
| --- | --- | --- |
| **Componel** | **Kiểu** | **Thuộc tính** |
| Login\_Dialog | Jdiog | Title : Login  defaultCloseOperation: DO\_NOTHING |
| lblTitle | Jlabel | Text : Thay đổi mật khẩu |
| lblNote | Jlabel | Text : Vui lòng nhập email hoặc số điện thoại bạn sử dụng để đăng kí tài khoản |
| lblMaNV | Jlabel | Text : Tên đăng nhập |
| lblEmail | Jlabel | Text : Email |
| lblMaXacNhan | Jlabel | Text : Mã xác nhận |
| lblMatKhau1 | Jlabel | Text : Mật khẩu mới |
| lblMatKhau2 | Jlabel | Text : Xác nhận mật khẩu |
| lblLogin | Jlabel | Text : Click here to login |
| Icon1 | Jlabel | Icon : show.png |
| Icon2 | Jlabel | Icon : hide.png |
| Icon3 | Jlable | Icon : show.png |
| Icon4 | Jlabel | Icone : hide.png |
| lblRefresh | Jlabel | Icon: refresh.png |
| lblreCode | Jlabel | Text : |
| btnChange | Button | Text : Thay đổi mật khẩu |
| btnSendEmail | Button | Text : Send Email |
| txtManv | JTextField |  |
| txtEmail | JTextFileld |  |
| txtCode | JTextField |  |
| txtPass1 | JPasswordFiled |  |
| txtPass2 | JPasswordFiled |  |

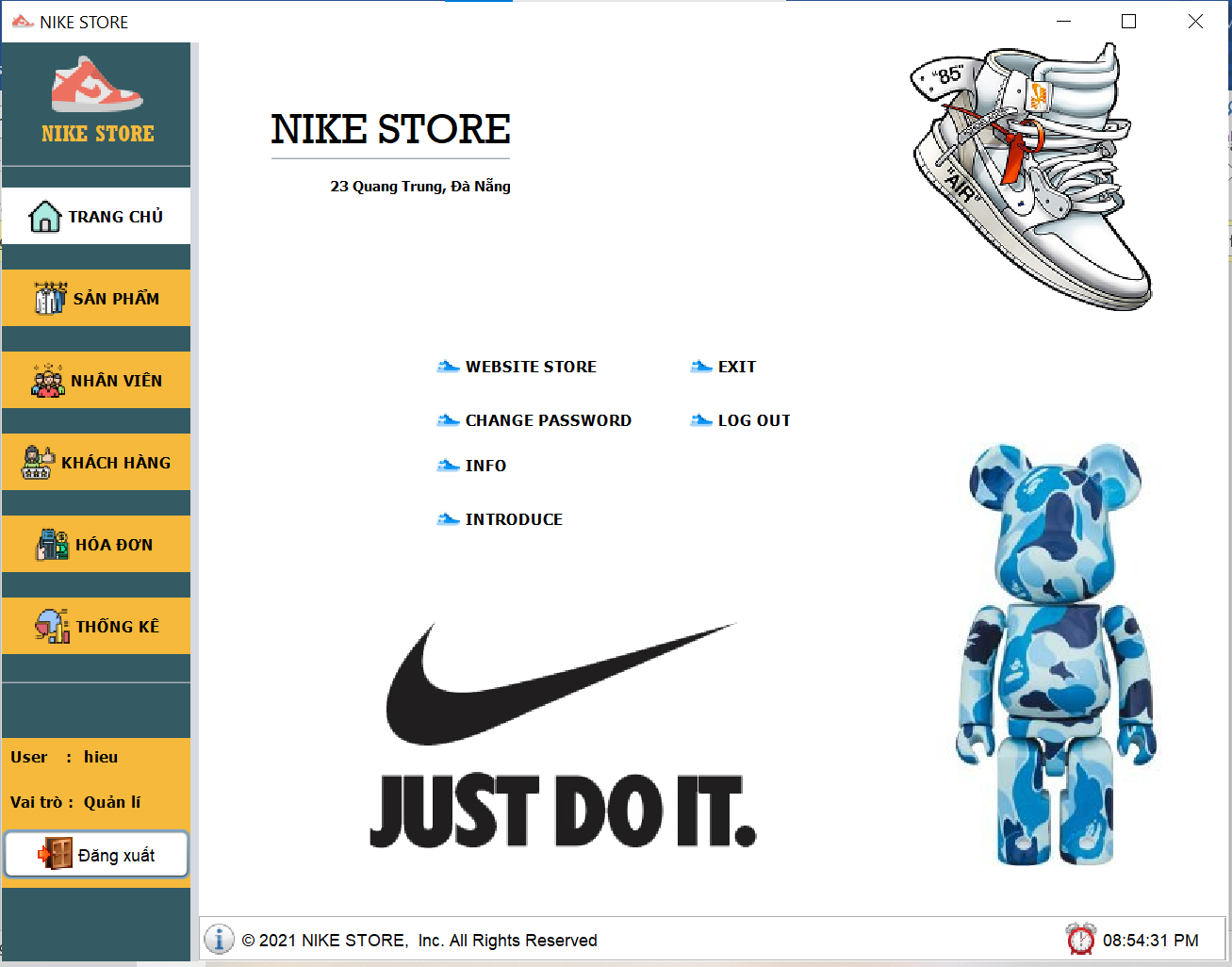
#### Thiết kế Form Chao\_Dialog :



Các thành phần giao diện :

|  |  |  |
| --- | --- | --- |
| Chao\_Dialog | JDialog | SetLocationRelavetive : null  undecorated:true Layout:BorderLayou |
| progressBar | JProgressBar | Layout.Derection:last stringPainted:true |
| lblStatus | Jlabel |  |
| lblImage | Jlabel | Icon : banner.jpg |
| lblTitle | Jlabel | Text :Fashion and so much more – Nike Store |

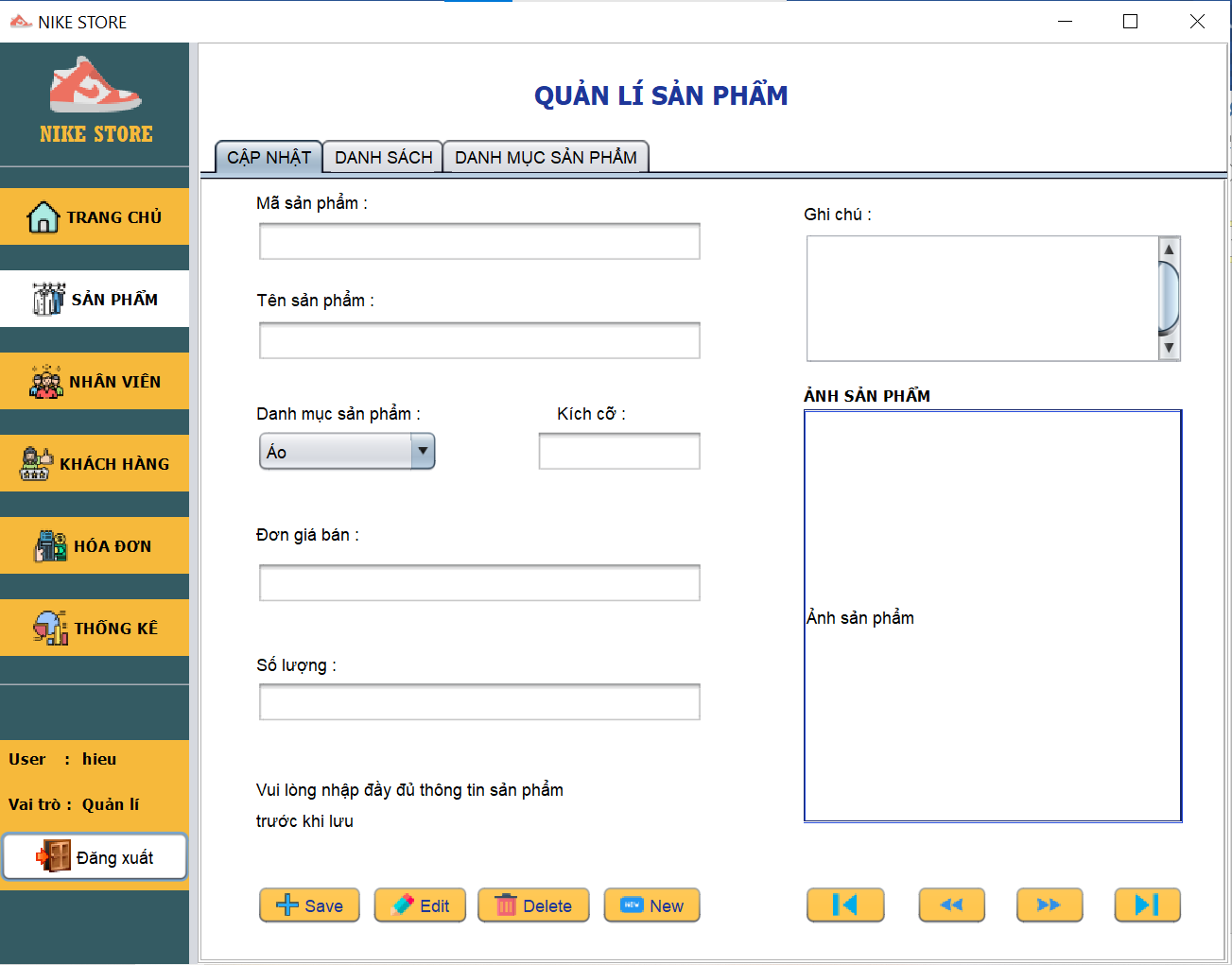
#### Thiết kế Form MainFrame :

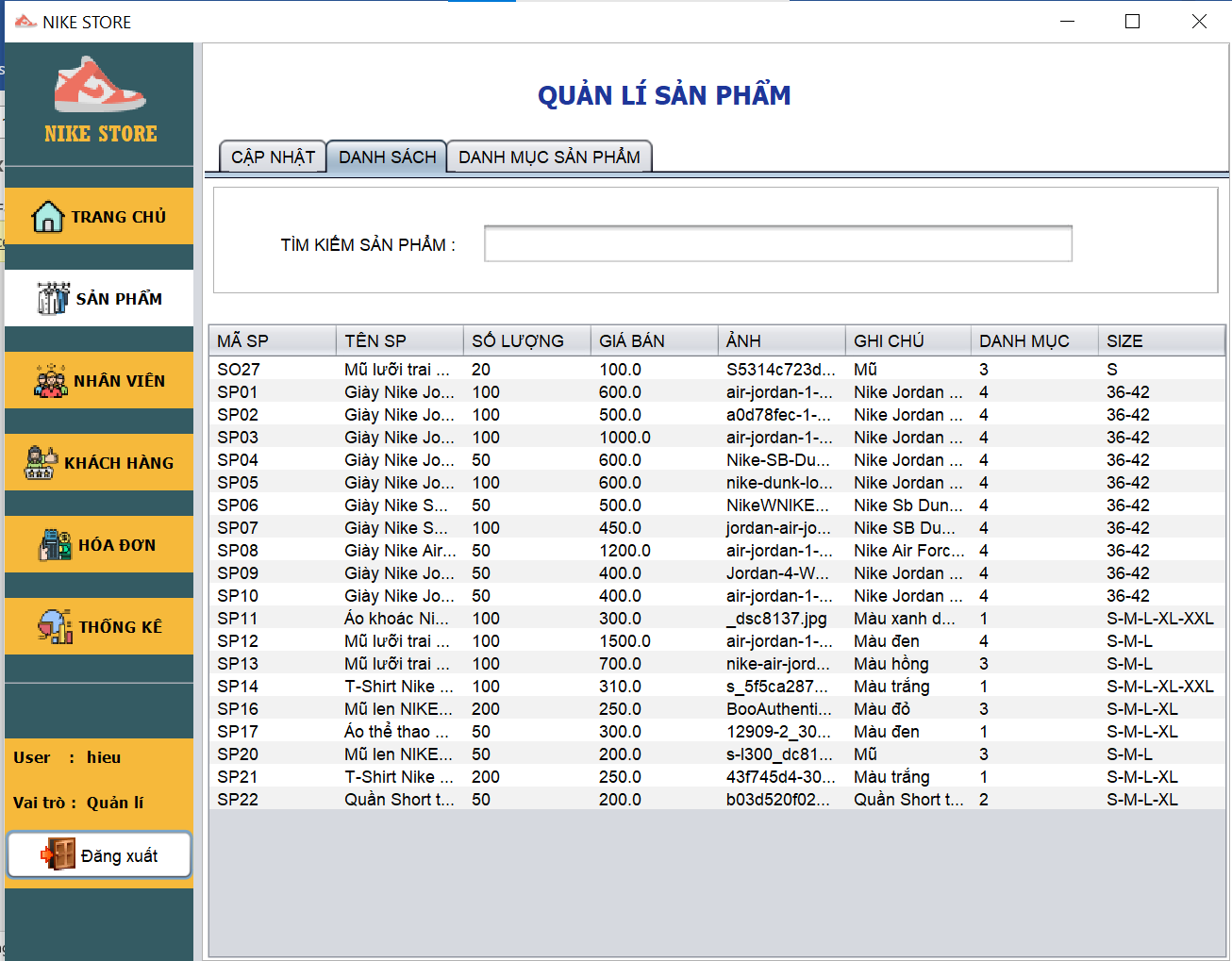


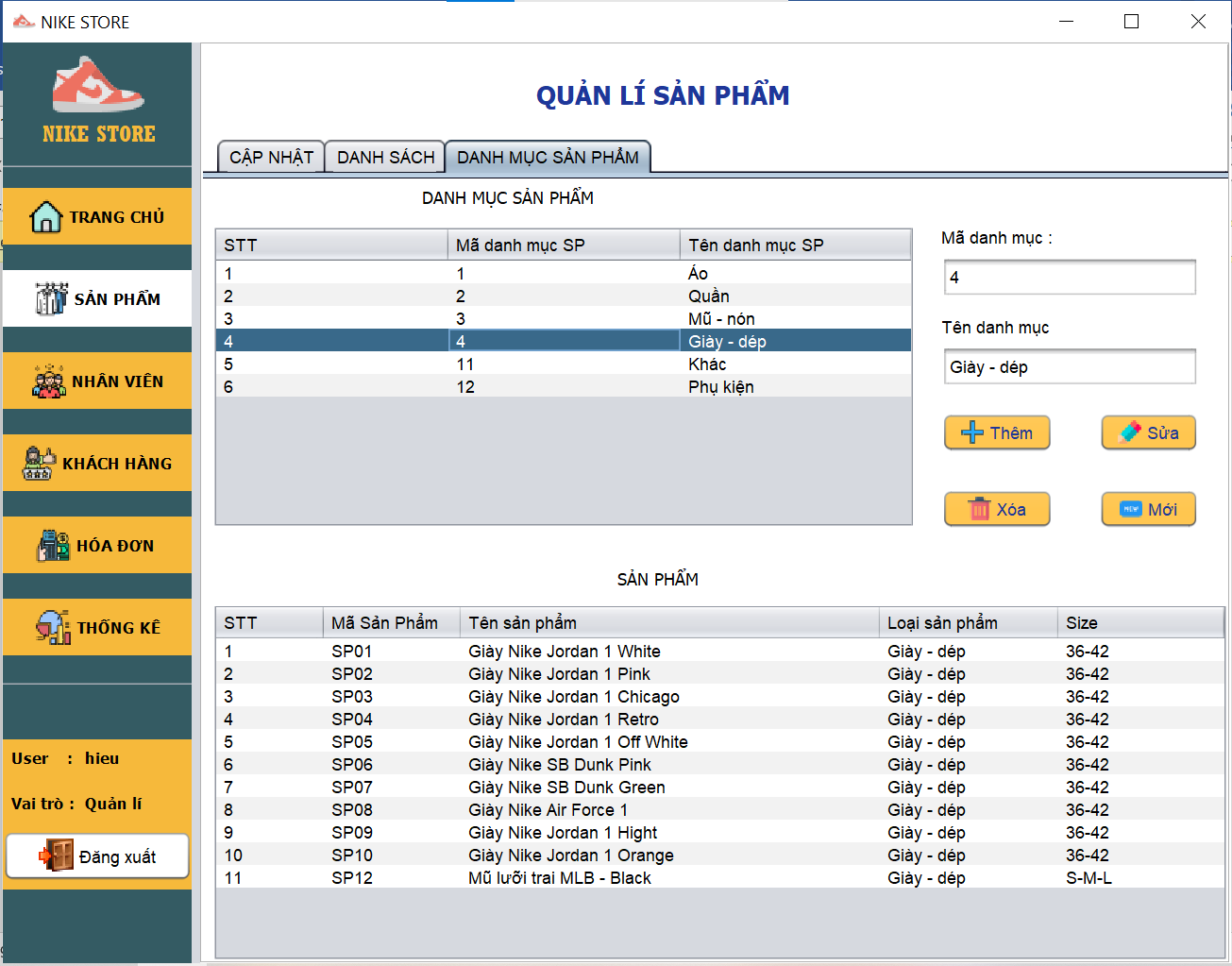
Các thành phần giao diện :

|  |  |  |
| --- | --- | --- |
| MainFrame | JFrame | SetTitle : Nike Store  SetLocationRelavetive : null |
| pnlMenu | Panel |  |
| logoApp | Jlabel | Icon : logoapp.png |
| pnlSanPham | Panel |  |
| lblSanPham | Jlabel | Text : Sản phẩm |
| pnlNhanVien | Panel |  |
| lblNhanVien | Jlable | Text : Nhân Viên |
| pnlKhachHang | Panel |  |
| lblKhachHang | Jlabel | Text : Khách hàng |
| pnlHoaDon | Panel |  |
| lblHoaDon | Jlabel | Text : Hóa đơn |
| pnlThongKe | Panel |  |
| lblThongKe | Jlabel | Text : Thống kê |
| pnlUser | Panel |  |
| lblUser | Jlabel | Text : User |
| lblRole | Jlabel | Text : Vai trò |
| User | Jlabel |  |
| vaiTro | Jlabel |  |
| btnDangXuat | Button | Icon : logout.png  Text :Đăng xuất |

#### Thiết kế Form Quản lí sản phẩm :

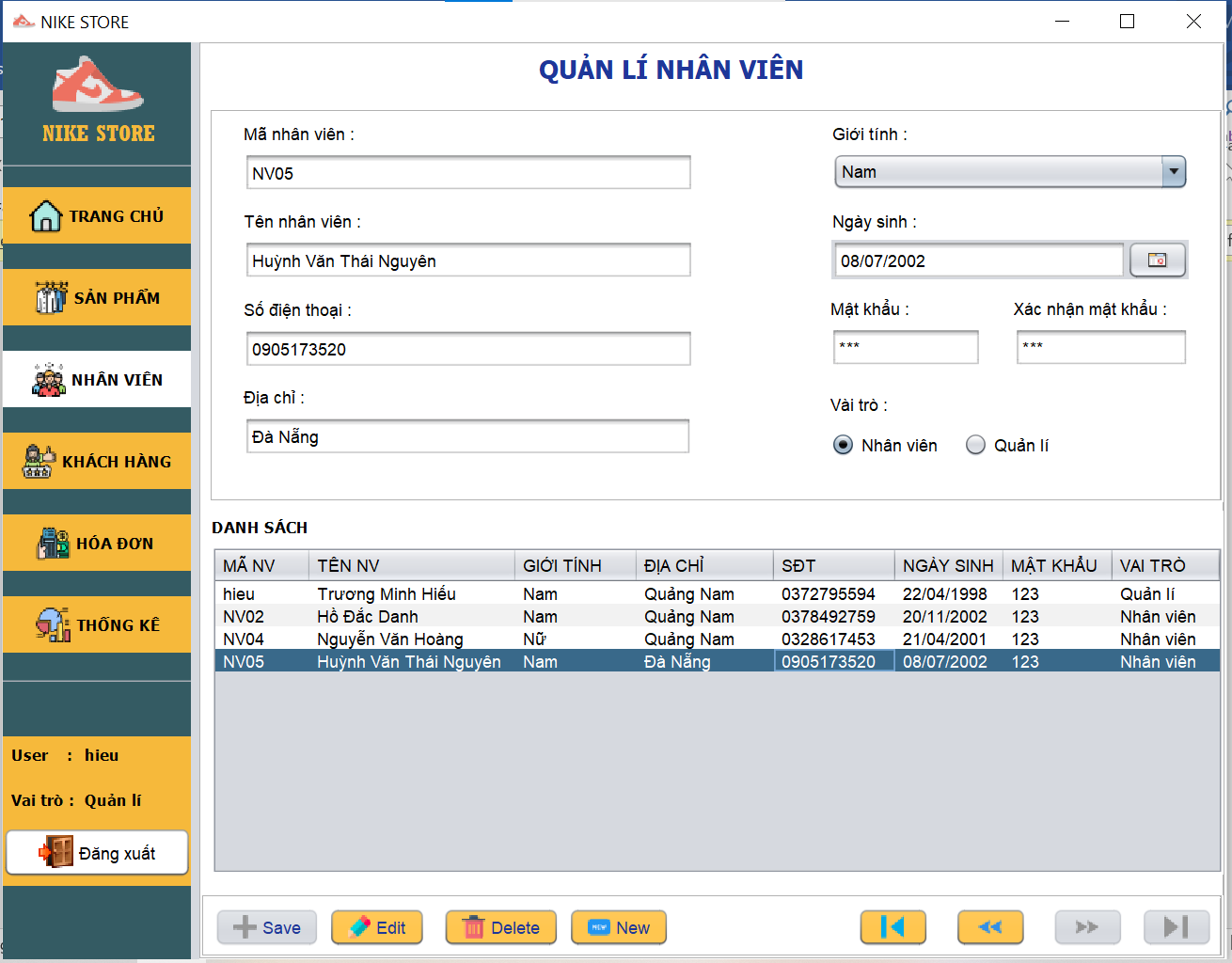






Các thành phần giao diện :

#### Thiết kế Form Quản lí nhân viên :



#### Thiết kế Form Quản lí khách hàng :



Các thành phần giao diện :

|  |  |  |
| --- | --- | --- |
| KhachHang\_Panel | Panel | Border : EtchedBorder |
| pnlCapNhat | Panel |  |
| lblTitle | JLabel | Text : Quản lí khách hàng |
| lblMaKH | JLabel | Text : Mã khách hàng |
| txtMaKH | JTextField |  |
| lblTenKH | JLabel | Text : Tên khách hàng |
| txtTenKH | JTextField |  |
| lblSDT | JLabel | Text : Số điện thoại |
| txtSDT | JTextField |  |
| lblDiaChi | JLabel | Text : Địa chỉ |
| txtDiaChi | JTextField |  |
| lblEmail | JLabel | Text : Email |
| txtEmail | JTextField |  |
| lblThanhVien | JLabel | Text : Thành viên |
| rdoCo | Radio Button |  |
| rdoKhong | Radio Button |  |
| bgrThanhVien | Button Group |  |
| pnlTimKiem | Panel | Border : EtchedBorder |
| lblTimKiem | Label | Text : Tìm kiếm |
| txtTimKiem | JTextField |  |
| btnXoa | Button |  |
| bntMoi | Button |  |
| btnFirst | Button |  |
| btnNext | Button |  |
| btnPrev | Button |  |
| btnLast | Button |  |
| pnlDanhSach | Panel | Border : Etched Border |
| tbKhachHang | JTable |  |
| pnlChucNang | Panel | Border : Etched Border |

#### Thiết kế Form Quản lí hóa đơn:

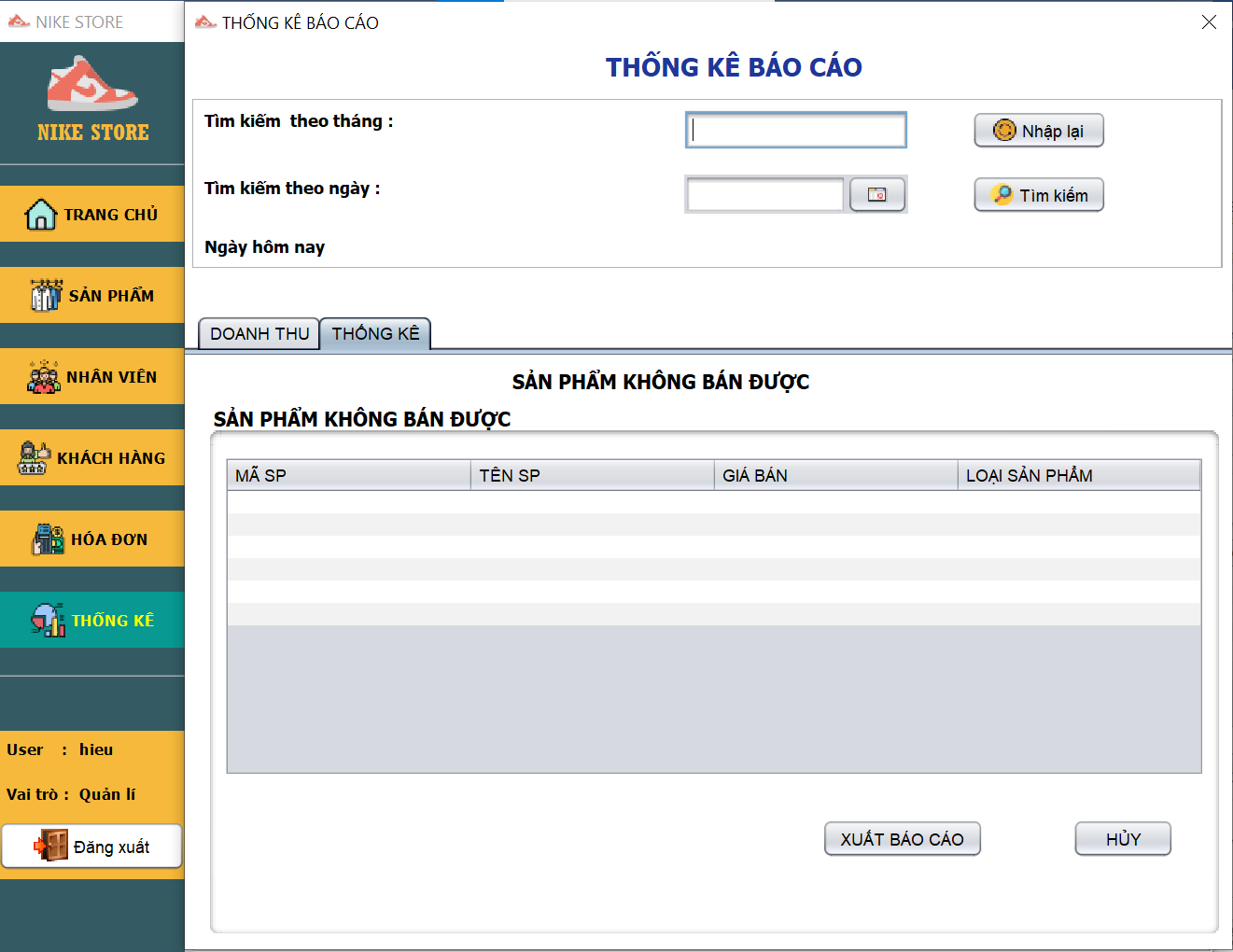


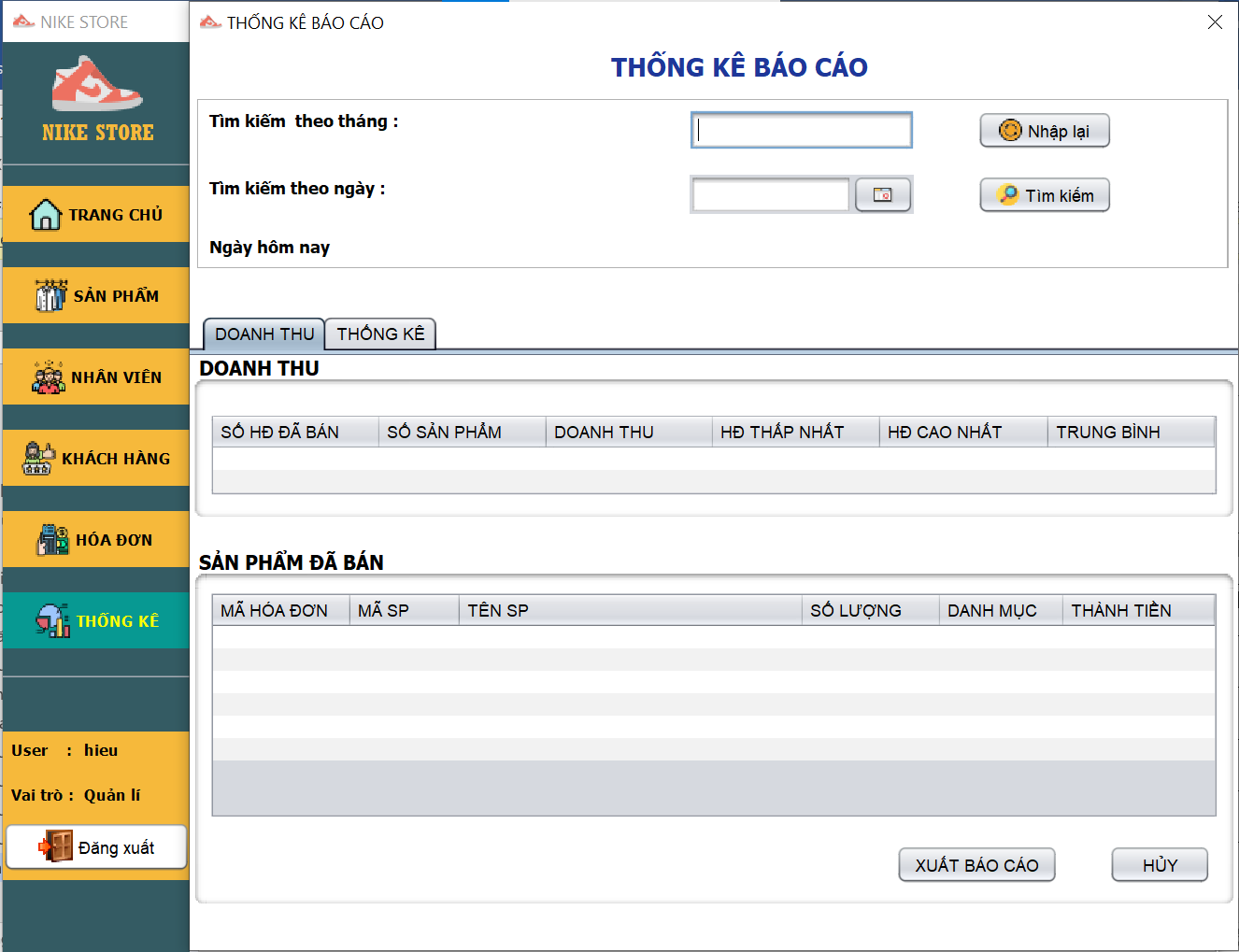


Các thành phần giao diện :

|  |  |  |
| --- | --- | --- |
| HoaDon\_Panel | Panel | Border : EtchedBorder |
| Tab | JTabbedPane |  |
| pnlThongTinChung | Panel | Border : TitleBorder : Thông tin chung |
| lblTitle | JLabel | Text : Quản lí hóa đơn |
| lblMaHD | JLabel | Text : Mã hóa đơn |
| txtMaHD | JTextField |  |
| lblNgayBan | JLabel | Text : Ngày bán |
| txtNgayBan | JTextField |  |
| lblMaNV | JLabel | Text : Mã nhân viên |
| txtMaNV | JTextField |  |
| lblTenNV | JLabel | Text : Tên nhân viên |
| txtTenNV | JTextField |  |
| lblMaKH | JLabel | Text : Mã khách hàng |
| cboMaKH | JComboBox |  |
| lblTenKH | JLabel | Text : Tên khách hàng |
| txtTenKH | JTextArea |  |
| lblDiaChi | JLabel | Text : Địa chỉ |
| txtDiaChi | JTextField |  |
| lblSDT | JLabel | Text : Số điện thoại |
| txtSDT | JTextField |  |
| tbHoaDon | JTable |  |
| btnSave |  |  |
| btnEdit | Button |  |
| btnDelete | Button |  |
| bntNew | Button |  |
| btnFirst | Button |  |
| btnNext | Button |  |
| btnPrev | Button |  |
| btnLast | Button |  |
| pnlHoaDonChiTiet | Panel | Border : TitleBorder : Thông tin chi tiết |
| lblMaSP | JLabel | Text : Mã sản phẩm |
| cboSP | JComboBox |  |
| lblTenSP | JLabel | Text : Tên sản phẩm |
| txtTenSP | JTextField |  |
| lblSoLuong | JLabel | Text : Số lượng |
| txtSoLuong | JTextField |  |
| lblDonGia | JLabel | Text : Đơn giá |
| txtDonGia | JTextField |  |
| lblGiamGia | JLabel | Text : Giảm giá |
| txtGiamGia | JTextField |  |
| lblThanhTien | JLabel | Text : Thành tiền |
| txtThanhTien | JTextField |  |
| tbHoaDonChiTiet | JTable |  |
| btnThem | Button |  |
| btnLuu | Button |  |
| btnXoa | Button |  |
| btnIn | Button |  |
| btnTimKiem | Button |  |

#### Thiết kế Form Báo cáo thống kê:





Các thành phần giao diện :

|  |  |  |
| --- | --- | --- |
| ThongKe\_Dialog | Jdialog | Title : Báo cáo thống kê |
| lblTitle | JLabel | Text : Báo cáo thống kê |
| tab | JTabbedPane |  |
| pnlDoanhThu | Panel |  |
| tbDoanhThu | JTable |  |
| lblNgay | JLabel | Text : Ngày |
| cboNgay | JComboBox |  |
| lblThang | JLabel | Text : Tháng |
| cboThang | JComboBox |  |
| pnlTop | Panel |  |
| tbTopSP | JTable |  |

# Thực hiện viết mã

## Viết mã tạo CSDL

### Tạo CSDL

#### CÂU LỆNH TẠO CSDL CÂU LỆNH TẠO CSDL

/\* câu lệnh tạo database \*/

Create database QuanLyBanHang

go

use QuanLyBanHang

/\*Function tạo mã sản phẩm tự động tăng \*/

CREATE FUNCTION AUTO\_MaSP()

RETURNS VARCHAR(5)

AS

BEGIN

DECLARE @ID VARCHAR(5)

IF (SELECT COUNT(MASP) FROM SanPham) = 0

SET @ID = '0'

ELSE

SELECT @ID = MAX(RIGHT(MASP, 3)) FROM SanPham

SELECT @ID = CASE

WHEN @ID >= 0 and @ID < 9 THEN 'SP00' + CONVERT(CHAR, CONVERT(INT, @ID) + 1)

WHEN @ID >= 9 and @ID <99 THEN 'SP0'+ CONVERT(CHAR, CONVERT(INT, @ID) + 1)

WHEN @ID >= 99 THEN 'SP'+ CONVERT(CHAR, CONVERT(INT, @ID) + 1)

END

RETURN @ID

END

/\*Function tạo mã nhân viên tự động tăng \*/

/\*Function tạo mã khách hàng tự động tăng \*/

/\* câu lệnh tạo bảng khách hàng \*/

Create Table KhachHang(

MaKH varchar(10) not null,

TenKH nvarchar(50) not null,

DiaChi nvarchar(50),

Sdt varchar(20),

Email varchar(50),

Thanhvien bit,

primary key (MaKh asc)

);

/\* câu lệnh tạo bảng danh mục sản phẩm\*/

Create table DanhMucSP(

MaDM int not null Identity (1,1),

TenDanhMuc nvarchar(50) not null,

primary key (MaDM asc)

);

/\* câu lệnh tạo bảng sản phẩm\*/

Create Table SanPham(

MaSP varchar(10) not null,

TenSP nvarchar(50) not null,

SoLuong int,

DonGiaBan float,

Anh varchar(max),

GhiChu nvarchar(50),

MaDM int,

Size nvarchar(50),

primary key (MaSP asc)

);

Create Table NhanVien(

MaNV varchar(10) not null,

TenNV nvarchar(50) not null,

GioiTinh bit,

DiaChi nvarchar(50),

DienThoai varchar(20),

NgaySinh date,

MatKhau varchar(30) not null,

VaiTro bit,

primary key (MaNV asc)

);

/\* câu lệnh tạo bảng hóa đơn\*/

Create Table HoaDon(

MaHDBan varchar(10) not null,

MaNV varchar(10) not null,

NgayBan date,

MaKH varchar(10),

primary key (MaHDBan asc)

);

/\* câu lệnh tạo bảng hóa đơn chi tiet\*/

Create Table HoaDonChiTiet(

MaHDCT int Identity(1,10),

MaHDBan varchar(10) not null,

MaSP varchar(10) not null,

SoLuong int,

GiamGia float,

primary key (maHDCT asc)

);

#### CÂU LỆNH TẠO RÀNG BUỘC GIỮA CÁC BẢNG

**A.Ràng buộc bảng HoaDon-HoaDonChiTiet**

ALTER TABLE [dbo].[HoaDonChiTiet] WITH CHECK ADD CONSTRAINT [FK\_ChiTietHoaDon\_HoaDon] FOREIGN KEY([MaHDBan])

REFERENCES [dbo].[HoaDon] ([MaHDBan])

ON UPDATE CASCADE

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[HoaDonChiTiet] CHECK CONSTRAINT [FK\_ChiTietHoaDon\_HoaDon]

GO

**B.Ràng buộc bảng HoaDonChiTiet-SanPham**

ALTER TABLE [dbo].[HoaDonChiTiet] WITH CHECK ADD CONSTRAINT [FK\_ChiTietHoaDon\_SanPham] FOREIGN KEY([MaSP])

REFERENCES [dbo].[SanPham] ([MaSP])

ON UPDATE CASCADE

GO

ALTER TABLE [dbo].[HoaDonChiTiet] CHECK CONSTRAINT [FK\_ChiTietHoaDon\_SanPham]

GO

**C.Ràng buộc bảng HoaDon-KhachHang**

ALTER TABLE [dbo].[HoaDon] WITH CHECK ADD CONSTRAINT [FK\_HoaDon\_KhachHang] FOREIGN KEY([MaKH])

REFERENCES [dbo].[KhachHang] ([MaKH])

ON UPDATE CASCADE

GO

ALTER TABLE [dbo].[HoaDon] CHECK CONSTRAINT [FK\_HoaDon\_KhachHang]

GO

**D.Ràng buộc bảng HoaDon-NhanVien**

ALTER TABLE [dbo].[HoaDon] WITH CHECK ADD CONSTRAINT [FK\_HoaDon\_NhanVien] FOREIGN KEY([MaNV])

REFERENCES [dbo].[NhanVien] ([MaNV])

ON UPDATE CASCADE

GO

ALTER TABLE [dbo].[HoaDon] CHECK CONSTRAINT [FK\_HoaDon\_NhanVien]

GO

**E.Ràng buộc bảng SanPham-DanhMucSP**

ALTER TABLE [dbo].[SanPham] WITH CHECK ADD CONSTRAINT [FK\_DanhMuc\_SanPham] FOREIGN KEY([MaDM])

REFERENCES [dbo].[DanhMucSP] ([MaDM])

GO

ALTER TABLE [dbo].[SanPham] CHECK CONSTRAINT [FK\_DanhMuc\_SanPham]

GO

### SQL truy vấn và thao tác

#### SQL đối với bảng SanPham

|  |  |
| --- | --- |
| Hành động | Câu lệnh SQL |
| Thêm mới Sản phẩm | INSERT INTO SanPham VALUES ([dbo].[AUTO\_MaSP](), ?, ?, ?, ?, ?, ?, ?) |
| Cập nhật sản phẩm | UPDATE SanPham SET TenSP =?, SoLuong=? ,DonGiaBan=? ,Anh=? , GhiChu=? ,MaDM = ?, Size = ? WHERE MaSP = ? |
| Xóa sản phẩm theo mã | DELETE FROM SanPham WHERE MaSP = ? |
| Select SanPham theo mã | SELECT \* FROM SanPham WHERE MaSP=? |
| Select tất cả SanPham | SELECT \* FROM SanPham |

#### SQL đối với bảng KhachHang

|  |  |
| --- | --- |
| Hành động | Câu lệnh SQL |
| Thêm mới Khách Hàng | INSERT INTO KhachHang values ([dbo].[AUTO\_MaKH](), ?, ?, ?, ?, ?) |
| Cập nhật khách hàng | UPDATE KhachHang SET TenKH =? ,DiaChi=? ,Sdt=? ,Email=? , ThanhVien=? WHERE MaKH = ? |
| Xóa khách hàng theo mã | DELETE FROM KhachHang WHERE MaKH = ? |
| Select KhachHang theo mã | SELECT \* FROM KhachHang WHERE MaKH=? |
| Select tất cả KhachHang | SELECT \* FROM KhachHang |

#### SQL đối với bảng NhanVien

|  |  |
| --- | --- |
| Hành động | Câu lệnh SQL |
| Thêm mới NhanVien | INSERT INTO NhanVien VALUES ([dbo].[AUTO\_MaNV](), ?, ?, ?, ?, ?, ?, ?) |
| Cập nhật NhanVien theo mã NV | UPDATE NhanVien SET TenNV =?, GioiTinh=? ,DiaChi=? ,DienThoai=? ,NgaySinh=? , MatKhau=? , VaiTro=? WHERE MaNV = ? |
| Xóa khách hàng theo mã nv | DELETE FROM NhanVien WHERE MaNV = ? |
| Select NhanVien theo mã nv | SELECT \* FROM NhanVien WHERE MaNV=? |
| Select tất cả NhanVien | SELECT \* FROM NhanVien |

#### SQL đối với bảng HoaDon

|  |  |
| --- | --- |
| Hành động | Câu lệnh SQL |
| Thêm mới HoaDon | INSERT INTO HoaDon (MaHDBan,MaNV,NgayBan,MaKH) VALUES (?, ?, ?, ?) |
| Cập nhật HoaDon theo mã HDBan | UPDATE HoaDon SET MaNV =?, NgayBan=? ,MaKH=? WHERE MaHDBan = ? |
| Xóa khách hàng theo mã HDBan | DELETE FROM HoaDon WHERE MaHDBan = ? |
| Select HoaDon theo mã HDBan | SELECT \* FROM HoaDOn WHERE MaHDBan=? |
| Select tất cả HoaDon | select hd.MaHDBan, MaNV, NgayBan, MaKH, SUM(((sp.DonGiaBan\*ct.SoLuong)\*(100-ct.GiamGia)/100)) as TongTien from HoaDon hd  inner join HoaDonChiTiet ct on ct.MaHDBan = hd.MaHDBan  inner join SanPham sp on sp.MaSP = ct.MaSP  GROUP BY hd.MaHDBan,MaNV, NgayBan, MaKH  ORDER BY NgayBan desc |

#### SQL đối với bảng HoaDonChiTiet

|  |  |
| --- | --- |
| Hành động | Câu lệnh SQL |
| Thêm mới HoaDonChiTiet | INSERT INTO HoaDonChiTiet ( MaHDBan,MaSP,SoLuong,GiamGia) VALUES (?, ?, ?, ?) |
| Cập nhật HoaDonChiTiet theo mã HDCT | UPDATE HoaDonChiTiet SET MaHDBan =?, MaSP=? ,SoLuong=?, GiamGia = ? WHERE MaHDCT = ? |
| Xóa khách hàng theo mã HDCT | DELETE FROM HoaDonChiTiet WHERE MaHDCT = ? |
| Select HoaDonChiTiet theo mã HDCT | SELECT \* FROM HoaDonChiTiet WHERE MaHDCT=? |
| Select tất cả HoaDonChiTiet | SELECT \* FROM HoaDonChiTiet |

#### SQL đối với bảng DanhMucSP

|  |  |
| --- | --- |
| Hành động | Câu lệnh SQL |
| Thêm mới DanhMucSP | INSERT INTO DanhMucSP ( TenDanhMuc) VALUES ( ?) |
| Cập nhật DanhMucSP theo mã DmSP | UPDATE DanhMucSP SET TenDanhMuc=? WHERE MaDM=? |
| Xóa DanhMucSP theo mã DMSP | DELETE FROM DanhMucSP WHERE MaDM = ? |
| Select DanhMucSP theo mã DMSP | SELECT \* FROM DanhMucSP WHERE MaDM=? |
| Select tất cả DanhMucSP | Select \* from DanhMucSP |

select MaSP,SanPham.TenSP,DanhMucSP.TenDanhMuc as TenDanhMuc,SanPham.Size from SanPham inner join DanhMucSP on DanhMucSP.MaDM = SanPham.MaDM

where DanhMucSP.MaDM = ?

### Các thủ tục lưu tổng hợp thống kê

#### Procedure sp\_HDCT\_MaHD

/\*proc gọi ra hóa đơn chi tiết theo mã hóa đơn bán\*/

if OBJECT\_ID ('sp\_HDCT\_MaHD') is not null

drop proc sp\_HDCT\_MaHD

go

CREATE PROCEDURE [dbo].[sp\_HDCT\_MaHD]

@MaHDBan varchar(10)

AS

BEGIN

select ct.MaHDCT,hd.MaHDBan, sp.TenSP, ct.SoLuong, sp.DonGiaBan, ct.GiamGia, ((sp.DonGiaBan\*ct.SoLuong)\*(100-ct.GiamGia)/100) as ThanhTien from HoaDonChiTiet ct

inner join HoaDon hd on ct.MaHDBan = hd.MaHDBan

inner join SanPham sp on sp.MaSP = ct.MaSP

where ct.MaHDBan = @MaHDBan

GROUP BY ct.MaHDCT,hd.MaHDBan, sp.TenSP,ct.SoLuong,sp.DonGiaBan, ct.GiamGia

END

GO

#### Procedure sp\_ThongKeSPDaBanTheoThang

/\*Thống kê số sản phẩm đã bán theo tháng \*/

if OBJECT\_ID ('sp\_ThongKeSanPhamDaBanTheoThang') is not null

drop proc sp\_ThongKeSanPhamDaBanTheoThang

go

CREATE PROCEDURE [dbo].[sp\_ThongKeSanPhamDaBanTheoThang]

@Thang int

AS

BEGIN

select ct.MaHDBan, sp.MaSP,sp.TenSP,SUM(ct.SoLuong) as SoLuong, dmsp.TenDanhMuc,

((sp.DonGiaBan\*ct.SoLuong)\*(100-ct.GiamGia)/100) as ThanhTien

from HoaDonChiTiet ct

inner join SanPham sp on sp.MaSP = ct.MaSP

inner join HoaDon h on h.MaHDBan=ct.MaHDBan

inner join DanhMucSP dmsp on dmsp.MaDM = sp.MaDM

where ct.MaHDBan is not null and month (h.NgayBan)=@Thang

group by ct.MaHDBan, sp.MaSP,sp.TenSP,ct.SoLuong,h.NgayBan, dmsp.TenDanhMuc,

((sp.DonGiaBan\*ct.SoLuong)\*(100-ct.GiamGia)/100)

END

GO

#### Procedure sp\_ThongKeSPDaBanTheoNgay

/\*Thống kê số sản phẩm đã bán theo ngày \*/

if OBJECT\_ID ('sp\_ThongKeSanPhamDaBanTheoNgay') is not null

drop proc sp\_ThongKeSanPhamDaBanTheoNgay

go

CREATE PROCEDURE [dbo].[sp\_ThongKeSanPhamDaBanTheoNgay]

@NgayBan date

AS

BEGIN

select ct.MaHDBan, sp.MaSP,sp.TenSP,SUM(ct.SoLuong) as SoLuong, dmsp.TenDanhMuc,

((sp.DonGiaBan\*ct.SoLuong)\*(100-ct.GiamGia)/100) as ThanhTien

from HoaDonChiTiet ct

inner join SanPham sp on sp.MaSP = ct.MaSP

inner join HoaDon h on h.MaHDBan=ct.MaHDBan

inner join DanhMucSP dmsp on dmsp.MaDM = sp.MaDM

where ct.MaHDBan is not null and h.NgayBan=@NgayBan

group by ct.MaHDBan, sp.MaSP,sp.TenSP,ct.SoLuong,h.NgayBan, dmsp.TenDanhMuc,((sp.DonGiaBan\*ct.SoLuong)\*(100-ct.GiamGia)/100)

END

GO

#### Procedure sp\_NhanVienCoDon

if OBJECT\_ID ('sp\_ThongKeNhanVien\_CoDon') is not null

drop proc sp\_ThongKeNhanVien\_CoDon

go

CREATE PROCEDURE [dbo].[sp\_ThongKeNhanVien\_CoDon]

@ngay date

AS

BEGIN

select hd.MaHDBan,sum(hdct.SoLuong) as SoLuongSP,nv.MaNV,nv.TenNV,hd.NgayBan,SUM(((sp.DonGiaBan\*hdct.SoLuong)\*(100-hdct.GiamGia)/100)) as TongTien

from HoaDonChiTiet hdct

inner join SanPham sp on hdct.MaSP = sp.MaSP

inner join HoaDon hd on hd.MaHDBan = hdct.MaHDBan

inner join NhanVien nv on hd.MaNV = nv.MaNV

where hd.NgayBan = @ngay and hd.MaKH is not null

group by hd.MaHDBan,nv.MaNV,nv.TenNV,hd.NgayBan

END

GO

#### Procedure sp\_HoaDon

/\*proc gọi ra tất cả hóa đơn \*/

if OBJECT\_ID ('sp\_HoaDon') is not null

drop proc sp\_HoaDon

go

CREATE PROCEDURE [dbo].[sp\_HoaDon]

AS

BEGIN

select hd.MaHDBan, MaNV, NgayBan, MaKH, SUM(((sp.DonGiaBan\*ct.SoLuong)\*(100-ct.GiamGia)/100)) as TongTien from HoaDon hd

inner join HoaDonChiTiet ct

on ct.MaHDBan = hd.MaHDBan

inner join SanPham sp

on sp.MaSP = ct.MaSP

GROUP BY hd.MaHDBan,MaNV, NgayBan, MaKH

ORDER BY NgayBan desc

END

GO

#### View xem DoanhThu

/\*create view dùng để tính doanh thu\*/

if OBJECT\_ID ('DoanhThu\_View') is not null

drop view DoanhThu\_View

go

CREATE VIEW DoanhThu\_View As

select hd.MaHDBan,hd.NgayBan, SUM(((sp.DonGiaBan\*ct.SoLuong)\*(100-ct.GiamGia)/100)) as TongTien,

SUM(ct.SoLuong) as SPDaBan

from HoaDon hd

inner join HoaDonChiTiet ct on ct.MaHDBan = hd.MaHDBan

inner join SanPham sp on sp.MaSP = ct.MaSP

GROUP BY hd.MaHDBan, hd.NgayBan

#### Procedure sp\_DoanhThuTheoNgay

/\*Tính doanh thu theo ngày \*/

if OBJECT\_ID ('sp\_DoanhThuTheoNgay') is not null

drop proc sp\_DoanhThuTheoNgay

go

CREATE PROC [dbo].[sp\_DoanhThuTheoNgay]

(@Ngay date)

AS BEGIN

select COUNT(DISTINCT MaHDBan) as TongHoaDonDaBan,

SUM(SPDaban) as SPDaBan,

Sum(TongTien) as DoanhThu,

Min(TongTien) as HDThapNhat,

Max (TongTien) as HDCaoNhat,

ROUND(AVG(TongTien),2) as TrungBinh

from DoanhThu\_View where NgayBan = @Ngay

END

#### Procedure sp\_DoanhThuTheoThang

/\*Thống kê doanh thu theo tháng \*/

if OBJECT\_ID ('sp\_DoanhThuTheoThang') is not null

drop proc sp\_DoanhThuTheoThang

go

CREATE PROC [dbo].[sp\_DoanhThuTheoThang]

(@Thang int)

AS BEGIN

select COUNT(DISTINCT MaHDBan) as TongHoaDonDaBan,

SUM(SPDaban) as SPDaBan,

Sum(TongTien) as DoanhThu,

Min(TongTien) as HDThapNhat,

Max (TongTien) as HDCaoNhat,

ROUND(AVG(TongTien),2) as TrungBinh

from DoanhThu\_View where MONTH(NgayBan) = @Thang

END

#### Procedure sp\_SPKhBanDuocThang

/\* Procedure danh sách các sản phẩm (MASP,TENSP,DonGiaBan) không bán được theo tháng \*/

if OBJECT\_ID ('sp\_KhBanDuocThang') is not null

drop proc sp\_KhBanDuocThang

go

CREATE PROC [dbo].[sp\_KhBanDuocThang]

(@Thang int)

AS BEGIN

SELECT sp.MaSP,sp.TenSP,DonGiaBan, dm.TenDanhMuc as LoaiSP

FROM SANPHAM sp

inner join DanhMucSP dm on dm.MaDM = sp.MaDM

WHERE sp.MASP not in ( SELECT hdct.MASP FROM HOADON hd,

HoaDonChiTiet hdct

WHERE hd.MaHDBan = hdct.MaHDBan AND month(hd.NgayBan)=@Thang)

END

GO

#### Procedure sp\_SPKhBanDuocNgay

/\* Procedure danh sách các sản phẩm (MASP,TENSP,DonGiaBan) không bán được theo ngày \*/

if OBJECT\_ID ('sp\_KhBanDuocNgay') is not null

drop proc sp\_KhBanDuocNgay

go

CREATE PROC [dbo].[sp\_KhBanDuocNgay]

(@Ngay date)

AS BEGIN

SELECT sp.MaSP,sp.TenSP,DonGiaBan, dm.TenDanhMuc as LoaiSP

FROM SANPHAM sp

inner join DanhMucSP dm on dm.MaDM = sp.MaDM

WHERE sp.MASP not in ( SELECT hdct.MASP FROM HOADON hd,

HoaDonChiTiet hdct

WHERE hd.MaHDBan = hdct.MaHDBan AND (hd.NgayBan)=@Ngay)

END

GO

## Lập trình JDBC

### Lớp hỗ trợ

#### Lớp tiện ích ChuyenManHinh

**public class ChuyenManHinh** {

private JPanel root;

private String kindSelect = "";

private List<DanhMucBeans> listItem = null;

**public ChuyenManHinh**(JPanel jpnroot){

this.root = jpnroot;

}

**public void setView**(JPanel jpnItem, JLabel jlbItem){

kindSelect = "TrangChu";

root.removeAll();

root.setLayout(new BorderLayout());

root.add(new Home\_panel());

// root.validate();

// root.repaint();

}

**public void setEvent**(List<DanhMucBeans> listItem){

this.listItem = listItem;

for (DanhMucBeans item : listItem) {

item.getJlb().addMouseListener(new LabelEvent(item.getKind(), item.getJpn(), item.getJlb()));

}

}

class LabelEvent implements MouseListener{

private JPanel node;

private String kind;

private JPanel jpnItem;

private JLabel jlbItem;

public LabelEvent(String kind, JPanel jpnItem, JLabel jlbItem) {

this.kind = kind;

this.jpnItem = jpnItem;

this.jlbItem = jlbItem;

}

@Override

public void mouseClicked(MouseEvent e) {

switch(kind){

case "SanPham":

node = new SanPham\_panel();

break;

case "NhanVien":

node = new NhanVien\_panel();

break;

case "KhachHang":

node = new KhachHang\_panel();

break;

case "HoaDon":

node = new HoaDon\_panel();

break;

default:

node = new Home\_panel();

break;

}

root.removeAll();

root.setLayout(new BorderLayout());

root.add(node);

root.validate();

// root.repaint();

setChangeBackground(kind);

}

@Override

public void mousePressed(MouseEvent e) {//khi click vào menu thì sẽ có hiệu ứng nháy màu chọn

kindSelect = kind;

jpnItem.setBackground(new Color(7, 153, 146));

jlbItem.setBackground(new Color(7, 153, 146));

jlbItem.setForeground(Color.yellow);

}

@Override

public void mouseReleased(MouseEvent e) {

}

@Override

public void mouseEntered(MouseEvent e) {// khi rê vào menu thì sẽ đổi màu

jpnItem.setBackground(Color.white);

}

@Override

public void mouseExited(MouseEvent e) { //khi thoát rê chuột ở menu

if (!kindSelect.equalsIgnoreCase(kind)) {

jpnItem.setBackground( new Color(246,185,59));

jlbItem.setBackground(Color.black);

}

}

private void setChangeBackground(String kind){

for (DanhMucBeans item : listItem) {

if (item.getKind().equalsIgnoreCase(kind)) {

item.getJlb().setBackground(Color.white);

item.getJpn().setBackground(Color.white); // khi click vào menu thì sẽ đổi màu panel

jlbItem.setForeground(Color.black);

}else{

item.getJlb().setBackground(new Color(246,185,59)); //kh click vào menu thì sẽ giữ nguyên màu lúc ban đầu

item.getJpn().setBackground(new Color(246,185,59));

}

}

}

}

}

#### Lớp tiện ích Auth (thông tin đăng nhập)

**public class Auth** {

/\*

\* Đối tượng này chứa thông tin người sử dụng sau khi đăng nhập

\*/

**public static NhanVien** user = null; //người dùng chưa login

**public static KhachHang khachHang = null;**

/\*Xóa thông tin của người dùng khi có yêu cầu đăng xuất

\*

\*/

**public static void clear()**{//log out

Auth.user = null;

}

/\*

\* Kiểm tra đăng nhập hay chưa

\*/

**public static boolean isLogin()**{

return Auth.user != null;

}

/\*

\* Kiểm tra có phải trường phòng hay không

\*/

**public static boolean isManager(){**

return Auth.isLogin() && user.isVaiTro();

}

/\*

\* Kiểm tra có phải trường phòng hay không

\*/

**public static boolean isThanhVien(){**

return khachHang.isThanhVien();

}

}

#### Lớp tiện ích MsgBox (các hộp thoại thông báo)

**public class MsgBox** {

**public static void showMessageDialog(**Component parent, String content){

JOptionPane.showMessageDialog(parent, content, "Hệ thống quản lí bán hàng NIKE STORE", JOptionPane.INFORMATION\_MESSAGE);

}

**public static boolean showConfirmDialog**(Component parent, String content){

int choose = JOptionPane.showConfirmDialog(parent, content, "Hệ thống quản lí bán hàng NIKE STORE",

JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE);

return choose == JOptionPane.YES\_OPTION;

}

**public static void showErrorDialog**(Component parent, String content, String title){

JOptionPane.showMessageDialog(parent, content, title, JOptionPane.ERROR\_MESSAGE);

}

}

#### Lớp tiệc ích validator (kiểm tra hợp lệ )

#### Lớp tiện ích Xdate (xử lí định dạng thời gian)

**public class XDate** {

static SimpleDateFormat formater = new SimpleDateFormat();

**public static Date toDate**(String date, String pattern) {

try {

formater.applyPattern(pattern); //pattern :"dd-mm-yyyy"

return formater.parse(date);

}

catch (ParseException ex) {

throw new RuntimeException(ex);

}

}

**public static String toString**(Date date, String...pattern) {

if(pattern.length > 0){

formater.applyPattern(pattern[0]);

}

if(date == null){

date = XDate.now();

}

return formater.format(date);

}

**public static Date addDays**(Date date, long days) {

date.setTime(date.getTime() + days\*24\*60\*60\*1000);

return date;

}

**public static Date now**() {

return new Date();

}

**public static Date add**(int days){

Calendar cal=Calendar.getInstance();

cal.add(Calendar.DAY\_OF\_MONTH,days);

return cal.getTime();

}

}

#### Lớp tiện ích XImages (xử lí định dạng hình ảnh)

**public class XImage** {

**public static final Image APP\_ICON**;

**static{**

// Tải biểu tượng ứng dụng

String file = "/store/app/icons/icon.png";

APP\_ICON = new ImageIcon(XImage.class.getResource(file)).getImage();

}

**public static boolean saveLogo**(File file){

File dir = new File("D:\\Du an 1\\store-app\\src\\main\\images"); //khai báo thư mục logos ngang hàng với src

// Tạo thư mục nếu chưa tồn tại

if(!dir.exists()){

dir.mkdirs();

}

File newFile = new File(dir, file.getName());

try {

// Copy vào thư mục logos (đè nếu đã tồn tại)

Path source = Paths.get(file.getAbsolutePath());

Path destination = Paths.get(newFile.getAbsolutePath());

Files.copy(source, destination, StandardCopyOption.REPLACE\_EXISTING);

return true;

}

catch (Exception ex) {

return false;

}

}

**public static ImageIcon readLogo**(String fileName){

File path = new File("D:\\Du an 1\\store-app\\src\\main\\images", fileName);

return new ImageIcon(new ImageIcon(path.getAbsolutePath()).getImage().getScaledInstance(300, 300, Image.SCALE\_DEFAULT));

}

}

#### Lớp tiện ích XJDBC (xử lí kết nối cơ sở dữ liệu)

**public class XJDBC** {

private static String driver="com.microsoft.sqlserver.jdbc.SQLServerDriver";

private static String dburl="jdbc:sqlserver://localhost;database=QuanLyBanHang";

private static String username="sa";

private static String password="minhhieu21";

public static store.app.utils.clsConnectDB connection = new store.app.utils.clsConnectDB ();

/\*

\* Nạp driver

\*/

**static{**

try {

Class.forName(driver);

}

catch (ClassNotFoundException ex) {

throw new RuntimeException(ex);

}

}

**public static PreparedStatement getStmt**(String sql, Object...args) throws SQLException{

Connection connection = DriverManager.getConnection(dburl, username, password);

PreparedStatement pstmt = null;

if(sql.trim().startsWith("{")){ //goi thủ tục lưu trữ

pstmt = connection.prepareCall(sql); //(store procedure)

}

else{

pstmt = connection.prepareStatement(sql);

}

for(int i=0;i<args.length;i++){

pstmt.setObject(i + 1, args[i]); // ps.setString(1, hv.gẹtHoTen());

}

return pstmt;

}

**public static void update**(String sql, Object...args) { //insert, update, delete

try {

PreparedStatement stmt = XJDBC.getStmt(sql, args);

try {

stmt.executeUpdate();

}

finally{

stmt.getConnection().close();

}

}

catch (SQLException e) {

throw new RuntimeException(e);

}

}

**public static ResultSet query**(String sql, Object...args) {

try {

PreparedStatement stmt = XJDBC.getStmt(sql, args);

return stmt.executeQuery();

}

catch (SQLException e) {

e.printStackTrace();

throw new RuntimeException(e);

}

}

public static Object value(String sql, Object...args){

try {

ResultSet rs = XJDBC.query(sql, args);

if (rs.next()){

return rs.getObject(0);

}

rs.getStatement().getConnection().close();

return null;

} catch (Exception e) {

throw new RuntimeException(e);

}

}

}

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

#### SanPham

public class SanPham {

private String maSP;

private String tenSP;

private int soLuong;

private float giaBan;

private String anh;

private String ghiChu;

private int maDM;

private String size;

public SanPham() {

}

@Override

public String toString(){ //do du lieu ra combobox

return this.tenSP;

}

Contruster

Getter/setter

}

#### DanhMucSP

private int maDM;

private String tenDM;

@Override

public String toString(){ //do du lieu ra combobox

return this.tenDM;

}

Contruster

Getter/setter

}

#### HoaDon

public class HoaDon {

private String maHDBan;

private String maNV;

private String maKH;

private Date ngayBan;

contruster

getter/setter

}

#### HoaDonChiTiet

public class HoaDonChiTiet {

private int MaHDCT;

private String maHDBan;

private String maSP;

private int soLuong;

private float giamGia;

private float thanhTien;

contruster

getter/setter

}

#### NhanVien

public class NhanVien {

private String maNV;

private String tenNV;

private boolean gioiTinh; //mặc định là nam

private String diaChi;

private String sdt;

private Date ngaySinh;

private String matKhau;

private boolean vaiTro = false; //mặc định là nhân viên

contruster

getter/setter

}

#### KhachHang

public class KhachHang {

private String maKH;

private String tenKH;

private String diaChi;

private String sdt;

private String email;

private boolean thanhVien = false; //mặc định là không

public KhachHang() {

}

@Override

public String toString(){ //do du lieu ra combobox

return this.tenKH;

}

Contruster

Getter/setter

}

### 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

abstract public class storeAppDAO<EntityType, KeyType> { //generic

//thêm mới thực thể vào CSDL

abstract public void insert(EntityType entity);//EntityType: SanPham, NhanVien, KhachHang...

//Cập nhật thực thể vào CSDL

abstract public void update(EntityType entity);

//Xóa bản ghi khỏi CSDL

abstract public void delete(KeyType id);//KeyType: String, Integer, Double...

//Truy vấn bản ghi theo ID

abstract public EntityType selectById(KeyType id);

//Truy vấn tất cả bản ghi của thực thể

abstract public List<EntityType> selectAll();

abstract protected List<EntityType> selectBySql(String sql, Object...args);

}

#### Lớp DAO làm việc với Bảng DanhMucSP

public class DanhMucSpDAO extends storeAppDAO<DanhMucSP, String>{

@Override

public void insert(DanhMucSP model) { (thực hiện insert dữ liệu vào CSDL)

String sql="INSERT INTO DanhMucSP ( TenDanhMuc) VALUES ( ?)";

XJDBC.update(sql,

model.getTenDM());

}

@Override

public void update(DanhMucSP model) { (cập nhật đối tượng ở CSDL)

String sql="UPDATE DanhMucSP SET TenDanhMuc=? WHERE MaDM=?";

XJDBC.update(sql,

model.getTenDM(),

model.getMaDM());

}

@Override

public void delete(String maDM) { (Xóa đối tượng theo mã)

String sql = "DELETE FROM DanhMucSP WHERE MaDM = ? ";

XJDBC.update(sql, maDM);

}

@Override

public DanhMucSP selectById(String maDM) { (truy vấn đối tượng theo mã)

String sql = "SELECT \* FROM DanhMucSP WHERE MaDM=?";

List<DanhMucSP> list = selectBySql(sql, maDM);

return list.size() > 0 ? list.get(0) : null;

}

@Override

public List<DanhMucSP> selectAll() { (truy vấn tất cả thông tin của đối tượng)

String sql = "Select \* from DanhMucSP";

return this.selectBySql(sql);

}

@Override

protected List<DanhMucSP> selectBySql(String sql, Object... args) {

List<DanhMucSP> list=new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = XJDBC.query(sql, args);

while(rs.next()){

DanhMucSP entity=new DanhMucSP();

entity.setMaDM(rs.getInt("MaDM"));

entity.setTenDM(rs.getString("TenDanhMuc"));

list.add(entity);

}

}

finally{

rs.getStatement().getConnection().close();

}

}

catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

}

#### Lớp DAO làm việc với Bảng SanPham

public class SanPhamDAO extends storeAppDAO<SanPham, String>{

@Override

public void insert(SanPham model) {

String sql="INSERT INTO SanPham VALUES ([dbo].[AUTO\_MaSP](), ?, ?, ?, ?, ?, ?, ?)”;

XJDBC.update(sql,

model.getMaSP(),

model.getTenSP(),

model.getSoLuong(),

model.getGiaBan(),

model.getAnh(),

model.getGhiChu(),

model.getMaDM(),

model.getSize());

}

@Override

public void update(SanPham model) {

String sql="UPDATE SanPham SET TenSP =?, SoLuong=? ,DonGiaBan=? ,Anh=? , GhiChu=? ,MaDM = ?, Size = ? WHERE MaSP = ?";

XJDBC.update(sql,

model.getTenSP(),

model.getSoLuong(),

model.getGiaBan(),

model.getAnh(),

model.getGhiChu(),

model.getMaDM(),

model.getSize(),

model.getMaSP());

}

@Override

public void delete(String maSP) {

String sql = "DELETE FROM SanPham WHERE MaSP = ? ";

XJDBC.update(sql, maSP);

}

@Override

public SanPham selectById(String maSP) {

String sql = "SELECT \* FROM SanPham WHERE MaSP=?";

List<SanPham> list = selectBySql(sql, maSP);

return list.size() > 0 ? list.get(0) : null;

}

@Override

public List<SanPham> selectAll() {

String sql = "SELECT \* FROM SanPham";

return this.selectBySql(sql);

}

@Override

protected List<SanPham> selectBySql(String sql, Object... args) {

List<SanPham> list=new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = XJDBC.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.setGiaBan(rs.getFloat("DonGiaBan"));

entity.setAnh(rs.getString("Anh"));

entity.setGhiChu(rs.getString("GhiChu"));

entity.setMaDM(rs.getInt("MaDM"));

entity.setSize(rs.getString("Size"));

list.add(entity);

}

}

finally{

rs.getStatement().getConnection().close();

}

}

catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

public List<SanPham> selectSPByDanhMuc(String maDM){ //tìm kiếm sản phẩm theo danh mục sản phẩm

String sql = "select MaSP,SanPham.TenSP,DanhMucSP.TenDanhMuc\n" +

" as TenDanhMuc,SanPham.GhiChu from SanPham,DanhMucSP where\n" +

" SanPham.MaDM=DanhMucSP.MaDM and DanhMucSP.MaDM = ?";

return this.selectBySql(sql, maDM);

}

public List<SanPham> selectByKeyword(String keyword){ //tìm kiếm sản phẩm theo từ khóa

String sql = "SELECT \* FROM SanPham WHERE TenSP LIKE ?";

return this.selectBySql(sql, "%" + keyword + "%");

}

}

#### Lớp DAO làm việc với bảng NhanVien

public class NhanVienDao extends storeAppDAO<NhanVien, String>{

@Override

public void insert(NhanVien model) {

String sql="INSERT INTO NhanVien VALUES ([dbo].[AUTO\_MaNV](), ?, ?, ?, ?, ?, ?, ?)";

XJDBC.update(sql,

model.getMaNV(),

model.getTenNV(),

model.isGioiTinh(),

model.getDiaChi(),

model.getSdt(),

model.getNgaySinh(),

model.getMatKhau(),

model.isVaiTro());

}

@Override

public void update(NhanVien model) {

String sql="UPDATE NhanVien SET TenNV =?, GioiTinh=? ,DiaChi=? ,DienThoai=? ,NgaySinh=? , MatKhau=? , VaiTro=? WHERE MaNV = ?";

XJDBC.update(sql,

model.getTenNV(),

model.isGioiTinh(),

model.getDiaChi(),

model.getSdt(),

model.getNgaySinh(),

model.getMatKhau(),

model.isVaiTro(),

model.getMaNV());

}

@Override

public void delete(String maNV) {

String sql = "DELETE FROM NhanVien WHERE MaNV = ? ";

XJDBC.update(sql, maNV);

}

@Override

public NhanVien selectById(String maNV) {

String sql = "SELECT \* FROM NhanVien WHERE MaNV=?";

List<NhanVien> list = selectBySql(sql, maNV);

return list.size() > 0 ? list.get(0) : null;

}

@Override

public List<NhanVien> selectAll() {

String sql = "SELECT \* FROM NhanVien";

return this.selectBySql(sql);

}

@Override

protected List<NhanVien> selectBySql(String sql, Object... args) {

List<NhanVien> list=new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = XJDBC.query(sql, args);

while(rs.next()){

NhanVien entity=new NhanVien();

entity.setMaNV(rs.getString("MaNV"));

entity.setTenNV(rs.getString("TenNV"));

entity.setGioiTinh(rs.getBoolean("GioiTinh"));

entity.setDiaChi(rs.getString("DiaChi"));

entity.setSdt(rs.getString("DienThoai"));

entity.setNgaySinh(rs.getDate("NgaySinh"));

entity.setMatKhau(rs.getString("MatKhau"));

entity.setVaiTro(rs.getBoolean("VaiTro"));

list.add(entity);

}

}

finally{

rs.getStatement().getConnection().close();

}

}

catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

public void qmk(NhanVien model) {

String sql="UPDATE NhanVien SET MatKhau=? WHERE MaNV=?";

XJDBC.update(sql,

model.getMatKhau(),

model.getMaNV());

}

}

#### Lớp DAO làm việc với bảng KhachHang

public class KhachHangDAO extends storeAppDAO<KhachHang, String>{

@Override

public void insert(KhachHang model) {

String sql="INSERT INTO KhachHang values ([dbo].[AUTO\_MaKH](), ?, ?, ?, ?, ?)";

XJDBC.update(sql,

model.getMaKH(),

model.getTenKH(),

model.getDiaChi(),

model.getSdt(),

model.getEmail(),

model.isThanhVien());

}

@Override

public void update(KhachHang model) {

String sql="UPDATE KhachHang SET TenKH =? ,DiaChi=? ,Sdt=? ,Email=? , ThanhVien=? WHERE MaKH = ?";

XJDBC.update(sql,

model.getTenKH(),

model.getDiaChi(),

model.getSdt(),

model.getEmail(),

model.isThanhVien(),

model.getMaKH());

}

@Override

public void delete(String maKH) {

String sql = "DELETE FROM KhachHang WHERE MaKH = ? ";

XJDBC.update(sql, maKH);

}

@Override

public KhachHang selectById(String maKH) {

String sql = "SELECT \* FROM KhachHang WHERE MaKH=?";

List<KhachHang> list = selectBySql(sql, maKH);

return list.size() > 0 ? list.get(0) : null;

}

@Override

public List<KhachHang> selectAll() {

String sql = "SELECT \* FROM KhachHang";

return this.selectBySql(sql);

}

@Override

protected List<KhachHang> selectBySql(String sql, Object... args) {

List<KhachHang> list=new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = XJDBC.query(sql, args);

while(rs.next()){

KhachHang entity=new KhachHang();

entity.setMaKH(rs.getString("MaKH"));

entity.setTenKH(rs.getString("TenKH"));

entity.setDiaChi(rs.getString("DiaChi"));

entity.setSdt(rs.getString("Sdt"));

entity.setEmail(rs.getString("Email"));

entity.setThanhVien(rs.getBoolean("ThanhVien"));

list.add(entity);

}

}

finally{

rs.getStatement().getConnection().close();

}

}

catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

public List<KhachHang> selectByKeyword(String keyword){ //tìm kiếm khách hàng theo từ khóa

String sql = "SELECT \* FROM KhachHang WHERE TenKH LIKE ?";

return this.selectBySql(sql, "%" + keyword + "%");

}

}

#### Lớp DAO làm việc với bảng HoaDon

public class HoaDonDAO extends storeAppDAO<HoaDon, String>{

@Override

public void insert(HoaDon model) {

String sql="INSERT INTO HoaDon (MaHDBan,MaNV,NgayBan,MaKH) VALUES (?, ?, ?, ?)";

XJDBC.update(sql,

model.getMaHDBan(),

model.getMaNV(),

model.getNgayBan(),

model.getMaKH());

}

@Override

public void update(HoaDon model) {

String sql="UPDATE HoaDon SET MaNV =?, NgayBan=? ,MaKH=? WHERE MaHDBan = ?";

XJDBC.update(sql,

model.getMaNV(),

model.getNgayBan(),

model.getMaKH(),

model.getMaHDBan());

}

@Override

public void delete(String maHD) {

String sql = "DELETE FROM HoaDon WHERE MaHDBan = ? ";

XJDBC.update(sql, maHD);

}

@Override

public HoaDon selectById(String maHD) {

String sql = "SELECT \* FROM HoaDOn WHERE MaHDBan=?";

List<HoaDon> list = selectBySql(sql, maHD);

return list.size() > 0 ? list.get(0) : null;

}

@Override

public List<HoaDon> selectAll() {

String sql = "SELECT \* FROM HoaDon";

return this.selectBySql(sql);

}

@Override

protected List<HoaDon> selectBySql(String sql, Object... args) {

List<HoaDon> list=new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = XJDBC.query(sql, args);

while(rs.next()){

HoaDon entity=new HoaDon();

entity.setMaHDBan(rs.getString("MaHDBan"));

entity.setMaNV(rs.getString("MaNV"));

entity.setNgayBan(rs.getDate("NgayBan"));

entity.setMaKH(rs.getString("MaKH"));

list.add(entity);

}

}

finally{

rs.getStatement().getConnection().close();

}

}

catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

public List<HoaDon> selectByKhachHang(String MaKH){ //tìm kiếm Khóa học theo chuyên đề

String sql = "select hd.MaHDBan, MaNV, NgayBan, MaKH, SUM(sp.DonGiaBan\*ct.SoLuong-ct.GiamGia) as ThanhTien from HoaDon hd \n" +

"inner join HoaDonChiTiet ct \n" +

"on ct.MaHDBan = hd.MaHDBan\n" +

"inner join SanPham sp \n" +

"on sp.MaSP = ct.MaSP\n" +

"where hd.MaKH = ? \n" +

"GROUP BY hd.MaHDBan,MaNV, NgayBan, MaKH";

return this.selectBySql(sql, MaKH);

}

private List<Object[]> getListOfArray(String sql, String[] cols, Object...arg){

try {

List<Object[]> list = new ArrayList<>();

ResultSet rs = XJDBC.query(sql, arg);

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; //danh sách mảng ->> filltable

} catch (Exception e) {

throw new RuntimeException();

}

}

public List<Object[]> getHoaDon(){ (dùng để đổ tất cả thông tin hóa đơn ra Table)

String sql = "{CALL sp\_HoaDon}";

String[] cols = {"MaHDBan","MaNV","NgayBan","MaKH","TongTien"};// chứa tên cột khi đổ vào table

return this.getListOfArray(sql, cols);//chứa kết quả câu lệnh

}

public List<Object[]> getHoaDonTheoKH(String maKH){

String sql = "{CALL sp\_HoaDonKhachHang(?)}";

String[] cols = {"MaHDBan","MaNV","NgayBan","MaKH","TongTien"};// chứa tên cột khi đổ vào table

return this.getListOfArray(sql, cols, maKH);//chứa kết quả câu lệnh

}

}

#### Lớp dao làm việc với bảng HoaDonChiTiet

public class HoaDonChiTietDAO extends storeAppDAO<HoaDonChiTiet, Integer>{

@Override

public void insert(HoaDonChiTiet model) {

String sql="INSERT INTO HoaDonChiTiet ( MaHDBan,MaSP,SoLuong,GiamGia) VALUES (?, ?, ?, ?)";

XJDBC.update(sql,

model.getMaHDBan(),

model.getMaSP(),

model.getSoLuong(),

model.getGiamGia());

}

@Override

public void update(HoaDonChiTiet model) {

String sql="UPDATE HoaDonChiTiet SET MaHDBan =?, MaSP=? ,SoLuong=?, GiamGia = ? WHERE MaHDCT = ?";

XJDBC.update(sql,

model.getMaHDBan(),

model.getMaSP(),

model.getSoLuong(),

model.getGiamGia(),

model.getMaHDCT());

}

@Override

public void delete(Integer MaHDCT) {

String sql = "DELETE FROM HoaDonChiTiet WHERE MaHDCT = ? ";

XJDBC.update(sql, MaHDCT);

}

@Override

public HoaDonChiTiet selectById(Integer MaHDCT) {

String sql = "SELECT \* FROM HoaDonChiTiet WHERE MaHDCT=?";

List<HoaDonChiTiet> list = selectBySql(sql, MaHDCT);

return list.size() > 0 ? list.get(0) : null;

}

@Override

public List<HoaDonChiTiet> selectAll() {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

@Override

protected List<HoaDonChiTiet> selectBySql(String sql, Object... args) {

List<HoaDonChiTiet> list=new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = XJDBC.query(sql, args);

while(rs.next()){

HoaDonChiTiet entity=new HoaDonChiTiet();

entity.setMaHDBan(rs.getString("MaHDBan"));

entity.setMaSP(rs.getString("MaSP"));

entity.setSoLuong(rs.getInt("SoLuong"));

entity.setGiamGia(rs.getFloat("GiamGia"));

list.add(entity);

}

}

finally{

rs.getStatement().getConnection().close();

}

}

catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private List<Object[]> getListOfArray(String sql, String[] cols, Object...arg){

try {

List<Object[]> list = new ArrayList<>();

ResultSet rs = XJDBC.query(sql, arg);

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; //danh sách mảng ->> filltable

} catch (Exception e) {

throw new RuntimeException();

}

}

public List<Object[]> getHDCTByMaHDBan(String MaHD){

String sql = "{CALL sp\_HDCT\_MaHD(?)}";

String[] cols = {"MaHDCT","MaHDBan","TenSP","SoLuong","DonGiaBan","GiamGia","ThanhTien"};// chứa tên cột khi đổ vào table

return this.getListOfArray(sql, cols, MaHD);//chứa kết quả câu lệnh

}

}

## Viết mã cho ứng dụng

### Xử lý Form Chao\_Dialog

**public void init(){**

setLocationRelativeTo(null);

Thread t=new Thread(){ //Thread (luồng) về cơ bản là một tiến trình con (sub-process)

int i=-1;

@Override

**public void run(){**

while(true){

try {

i++;

progressBar.setValue(i);

if(i==20)lblStatus.setText("Đang khởi tạo cái modun...");

if(i==50)lblStatus.setText("Đang kết nối CSDL...");

if(i==70)lblStatus.setText("Chuẩn bị vào chương trình...");

if(i==90)lblStatus.setText("Thành công !");

if(i==100){

Chao\_Dialog.this.dispose(); //đóng ChaoJDialog

break;

}

Thread.sleep(20); //thread tạm dừng hoạt động trong 20 ms

} catch (InterruptedException ex) {

break;

}

}

}

};

t.start(); //thread bắt đầu hoạt động

}

### Xử lí Form Login\_Dialog

**void init(){**

setLocationRelativeTo(null);

}

**void exit (){**

int i=JOptionPane.showConfirmDialog(this, "Bạn có muốn kết thúc ứng dụng ?",

"THÔNG BÁO !",JOptionPane.YES\_OPTION

,JOptionPane.QUESTION\_MESSAGE);

if(i==0){

System.exit(0);

}

}

NhanVienDao dao = new NhanVienDao();

**void dangnhap(){**

StringBuilder sb = new StringBuilder();

Validator.validateEmty(txtUsername, sb, "Tên đăng nhập không được để trống !");

Validator.validateEmty(txtPassword, sb, "Mật khẩu không được để trống !");

if (sb.length() > 0) {

MsgBox.showErrorDialog(this, sb.toString(), "LỖI");

return;

}

String manv = txtUsername.getText();//manv là usernam

String matkhau = new String(txtPassword.getPassword());

NhanVien nv = dao.selectById(manv);//tìm nhân viên theo mã nv

if (nv==null) {//không tìm thấy username

MsgBox.showErrorDialog(this, "Sai tên đăng nhập","Lỗi");

}else if(!matkhau.equals(nv.getMatKhau())){

MsgBox.showErrorDialog(this, "Sai mật khẩu","Lỗi");

}else{

Auth.user = nv; // gán user login thành công

// MsgBox.showMessageDialog(this, "Đăng nhập thành công !\nNgười dùng " + "'" +Auth.user.getMaNV() + "'");

MainFrame mainfr = new MainFrame();

mainfr.setVisible(true);

this.dispose();

}

}

### Xử lí Form QuenMatKhau\_Dialog

**void exit (){**

int i=JOptionPane.showConfirmDialog(this, "Bạn có muốn kết thúc ứng dụng ?",

"THÔNG BÁO !",JOptionPane.YES\_OPTION

,JOptionPane.QUESTION\_MESSAGE);

if(i==0){

System.exit(0);

}

}

**public void sendMail(){**

final String username = "haideptrai210402@gmail.com";

final String password = "minhhieu21";

Properties prop = new Properties();

prop.put("mail.smtp.host", "smtp.gmail.com");

prop.put("mail.smtp.port", "587");

prop.put("mail.smtp.auth", "true");

prop.put("mail.smtp.starttls.enable", "true"); //TLS

//Đăng nhập gmail

Session session = Session.getInstance(prop,

new javax.mail.Authenticator() {

protected javax.mail.PasswordAuthentication getPasswordAuthentication() {

return new javax.mail.PasswordAuthentication(username, password);

}

});

try {

Message message = new MimeMessage(session);

message.setFrom(new InternetAddress("hieutd21042002@gmail.com"));

message.setRecipients(

Message.RecipientType.TO,

InternetAddress.parse(txtEmail.getText())

);

message.setSubject("Thay doi mat khau !");

message.setText("Ma xac nhan cua ban la : "+lblCode.getText());

Transport.send(message);

JOptionPane.showMessageDialog(this, "Email đã được gửi !");

System.out.println("Done");

} catch (MessagingException e) {

e.printStackTrace();

}

}

**void quenMatKhau(){**

if (validateForm()) {

NhanVienDao dao = new NhanVienDao();

try {

//So sánh mật khẩu có trùng hay không ?

String s1 = txtPass.getText();

String s2 = txtRepass.getText();

if (s1.equals(s2)==false) {

MsgBox.showMessageDialog(this, "Mật khẩu không khớp");

txtRepass.setBackground(Color.red);

return;

}

//So sánh mã xác nhận có trùng hay không ?

String s3 = lblCode.getText();

String s4 = txtRecode.getText();

if (s3.equals(s4)==false) {

MsgBox.showMessageDialog(this, "Mã xác nhận không khớp");

txtRecode.setBackground(Color.red);

return;

}

NhanVien nv = dao.selectById(txtUsername.getText()); // Tìm kiếm mã người dùng

if (nv == null) {

MsgBox.showErrorDialog(this, "Tên đăng nhập không hợp lệ", "LỖI !");

txtUsername.setBackground(Color.red);

}else{

int i=JOptionPane.showConfirmDialog(this, "Bạn có muốn thay đổi mật khẩu không ?",

"THÔNG BÁO !",JOptionPane.YES\_OPTION

,JOptionPane.QUESTION\_MESSAGE);

if(i==0){

//câu lệnh để thay đổi mật khẩu

try {

NhanVien user = new NhanVien();

user.setMaNV(txtUsername.getText());

user.setMatKhau(txtPass.getText());

dao.qmk(user);

JOptionPane.showMessageDialog(this, "Mật khẩu đã được thay đổi !");

clearForm();

} catch (Exception e) {

JOptionPane.showMessageDialog(this, "LỖI !!!\n"+e.getMessage());

// e.printStackTrace();

}

}

}

} catch (Exception e) {

e.printStackTrace();

MsgBox.showErrorDialog(this, e.getMessage(), "LỖI");

}

}else{

MsgBox.showMessageDialog(this, "Bạn chưa nhập đầy đủ thông tin");

return;

}

}

### Xử lý Form MainFrame

**public void chuyenManHinh(){**

ChuyenManHinh cmh = new ChuyenManHinh(pnlView);

cmh.setView(pnlTrangChu, lblTrangChu);

List<DanhMucBeans> listItem = new ArrayList<>();

listItem.add(new DanhMucBeans("TrangChu",pnlTrangChu,lblTrangChu));

listItem.add(new DanhMucBeans("SanPham",pnlSanPham,lblSanPham));

listItem.add(new DanhMucBeans("NhanVien",pnlNhanVien,lblNhanVien));

listItem.add(new DanhMucBeans("KhachHang",pnlKhachHang,lblKhachHang));

listItem.add(new DanhMucBeans("HoaDon",pnlHoaDon,lblHoaDon));

listItem.add(new DanhMucBeans("ThongKe",pnlThongKe,lblThongKe));

cmh.setEvent(listItem);

}

**void init(){**

setIconImage(XImage.APP\_ICON);

setLocationRelativeTo(null);

}

**void openThongKe() {**

if (Auth.isLogin()) {

if (!Auth.isManager()) {

MsgBox.showMessageDialog(this, "Bạn không có quyền xem thông tin doanh thu");

return;

}

new ThongKe\_Dialog(this, true).setVisible(true);

} else {

MsgBox.showMessageDialog(this, "Vui lòng đăng nhập");

}

}

**void dangxuat()** {

Auth.clear();

new Login\_Dialog(this, true).setVisible(true);

processLoginSuccessful();

this.dispose();

}

**private void processLoginSuccessful(){**

try {

user.setText(Auth.user.getMaNV());

if (Auth.user.isVaiTro()==true) {

vaitro.setText("Quản lí");

}else {

vaitro.setText("Nhân viên");

}

} catch (Exception e) {

MsgBox.showMessageDialog(this, "Vui lòng đăng nhập để sử dụng hệ thống !");

this.dispose();

}

}

### Xử lí Form Home\_panel

**void dongho()** {

new Timer(1000, new ActionListener() {

SimpleDateFormat format = new SimpleDateFormat("hh:mm:ss a");

@Override

public void actionPerformed(ActionEvent e) {

lblDongHo.setText(format.format(new Date()));

}

}).start();

}

**void openWeb(){**

try {

Desktop.getDesktop().browse(new File("D:\\WEB\_PT16305\\ASM1//index.html").toURI());

//thư mục help đặt ngang hàng với src

} catch (IOException ex) {

MsgBox.showMessageDialog(this, "Không tìm thấy file hướng dẫn!");

}

}

**private void doimatkhau(){** //thực hiện đổi mật khẩu

String manv = txtTenDangNhap.getText();

String matkhau = new String(txtPassCu.getPassword());

String matkhaumoi = new String(txtPassMoi.getPassword());

String matkhau2 = new String(txtXacNhanPass.getPassword());

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

MsgBox.showMessageDialog(this, "Sai tên đăng nhập !");

txtTenDangNhap.setBackground(Color.pink);

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

MsgBox.showMessageDialog(this, "Mật khẩu hiện tại không đúng !");

txtPassCu.setBackground(Color.pink);

}else if(!matkhaumoi.equals(matkhau2)){

MsgBox.showMessageDialog(this, "Mật khẩu không trùng khớp");

txtXacNhanPass.setBackground(Color.pink);

}else {

NhanVienDao dao = new NhanVienDao();

Auth.user.setMatKhau(matkhaumoi);

dao.update(Auth.user);

MsgBox.showMessageDialog(this, "Đổi mật khẩu thành công");

ChangePass\_Dialog.hide();

}

}

**public void Info(){ //đưa thông tin người đăng nhập lên form**

lblMaNV.setText(Auth.user.getMaNV());

lblTenNV.setText(Auth.user.getTenNV());

lblSdt.setText(Auth.user.getSdt());

lblDiaChi.setText(Auth.user.getDiaChi());

lblNgaySinh.setText(XDate.toString(Auth.user.getNgaySinh(), "dd/MM/yyyy"));

if (Auth.user.isVaiTro()==true) {

rdoNam.setSelected(true);

} else {

rdoNu.setSelected(true);

}

if (Auth.user.isVaiTro()==true) {

lblVaiTro.setText("Quản lí");

}else {

lblVaiTro.setText("Nhân viên");

}

}

### Xử lí Form KhachHang\_panel

import java.awt.Color;

import static java.awt.Color.white;

import java.util.List;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import javax.swing.JOptionPane;

import javax.swing.JTable;

import javax.swing.JTextField;

import javax.swing.table.DefaultTableModel;

import store.app.dao.KhachHangDAO;

import store.app.entity.KhachHang;

import store.app.entity.NhanVien;

import store.app.utils.Auth;

import store.app.utils.MsgBox;

import store.app.utils.XDate;

**public boolean checkTrungMa**(JTextField txt) {//kiểm tra trùng mã khách hàng

}

**void fillTable(){**

//tìm kiếm khách hàng theo keyword rồi đổ list khách hàng vào bảng

//nếu không có keywork trong ô tìm kiếm thì đổ toàn bộ khách hàng trong CSDL vào bảng

}

**KhachHang getForm()** {

//lấy thông tin trên form cho vào đt khách hàng thực hiện update hoặc insert

//return khách hàng

}

**void setForm(KhachHang kh)** {

//lấy thông tin đối tượng khách hàng đưa lên form

}

int row = -1; //chứa vị trí hiện hành của khách hàng

**void updateStatus(**){ //cập nhật trạng thái form

}

**void edit()**{ //điền thông tin đt nhanVien lên form (theo vị trí row)

}

**void insert(){**

//lấy thông tin trên form để

//thêm (đăng kí) khách hàng - quản lí vào CSDL

}

**void update(){**

//lấy thông tin trên form để

//cập nhật nhanVien theo maKH

}

**void delete()** {

//lấy maKH trên form, xóa khách hàng theo maKH

//xóa trắng form

}

**public static boolean checkEmail(JTextField field, StringBuilder sb)** {

//kiểm tra định dạng Email theo JtextField truyền vào

Pattern pattern = Pattern.compile("\\w+@\\w+\\.\\w+");

}

**public static boolean checkSDT(JTextField field, StringBuilder sb){**

//kiểm tra định dạng số điện thoại theo JtextField truyền vào

Pattern pattern = Pattern.compile("(0?)(3[2-9]|5[6|8|9]"

+ "|7[0|6-9]|8[0-6|8|9]|9[0-4|6-9])[0-9]{7}$"); //biểu thức chính quy số điện thoại VN tìm từ trên mạng

**}**

### Xử lí Form NhanVien\_panel

**public String matKhauToSao**(String pass){

//nếu nv đăng nhập với vai trò là Quản lí thì có thể xem mật khẩu và ngược lại

}

**Tương tự như FORM KHACHHANG**

### Xử lí Form SanPham\_panel

JFileChooser jFileChooser = new JFileChooser("D:\\Du an 1\\store-app\\src\\main\\images"); //đường dẫn tới thư mục hình ảnh

DanhMucSpDAO dmdao = new DanhMucSpDAO();

String strHinhAnh = null;

**public void LayDuLieuSanPhamTheoDanhMucSP (String MaDM)** {

//Lấy dữ liệu sản phẩm theo mã DMSP truyền vào

}

**void fillComboBoxDmSP()** { //đổ dữ liệu danh mục sản phẩm lên combobox

}

**void chonAnh()** {

//thực hiện việc chọn ảnh sản phẩm

}

### Xử lí Form HoaDon\_panel

HoaDonChiTietDAO ctdao = new HoaDonChiTietDAO();

KhachHangDAO khdao = new KhachHangDAO();

HoaDonDAO hddao = new HoaDonDAO();

**public boolean checkSo() {**

//kiểm tra định dạng số và phạm vi giá trị

**}**

**void fillTableHDCTbyMaHD()** {//đổ danh sách hóa đơn chi tiết vào bảng điểm theo maHDBan

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

model.setRowCount(0);

String mahd = txtMaHD.getText();

List<Object[]> list = ctdao.getHDCTByMaHDBan(mahd); //gọi store procedure sp\_bangdiem (lấy theo mã khóa học)

for (Object[] row : list) { //cho vòng lặp duyệt qua mảng líst

model.addRow(new Object[]{row[0], row[1], row[2], row[3], row[4],row[5],row[6]});

}

}

**void fillComboBoxSP()** {

//đổ dữ liệu danh mục sản phẩm lên combobox

}

**void TinhTien()** {

//thực hiện tính tiền hóa đơn khi nhân viên chọn sản phẩm bán, số lượng và giảm giá

}

**void fillTableHoaDon()** { //đổ danh sách hóa đơn được tính tiền theo công thức từ procedure lưu trong CSDL

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

model.setRowCount(0);

List<Object[]> list = hddao.getHoaDon();//gọi store procedure sp\_HoaDon

for (Object[] row : list) {//cho vòng lặp duyệt qua mảng líst

model.addRow(row);

}

}

**void showBill(String maHD**) {

//hiển thị tất cả thông tin hóa đơn lên TextAre cho nhân viên xem trước khi in

}

**void printBill()** {

//thực hiện in hóa đơn

}

**void getFom/setForm (){**

//tương tự như cách sử lí của Form KhachHang

}

### Xử lí Form ThongKe\_Dialog

**SimpleDateFormat date\_format = new SimpleDateFormat("yyyy/MM/dd");** //định dạng ngày

**public boolean checkSo(String thang) {**

//Kiểm tra dịnh dạng tháng

**}**

**void fillTableDoanhThuTheoThang() {**

//Điền dữ liệu vào bảng doanh thu theo tháng tương ứng

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

model.setRowCount(0);

int thang = (Integer.valueOf(txtThang.getText()));

List<Object[]> list = tkdao.getDoanhThuTheoThang(thang);

for (Object[] row : list) {//cho vòng lặp duyệt qua mảng líst

model.addRow(row);

}

}

**void fillTableDoanhThuTheoNgay()** {//Điền dữ liệu vào bảng doanh thu theo ngày tương ứng

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

model.setRowCount(0);

String ngay = ((JTextField)dateChooser.getDateEditor().getUiComponent()).getText();

List<Object[]> list = tkdao.getDoanhThuTheoNgay(ngay);

for (Object[] row : list) {//cho vòng lặp duyệt qua mảng líst

model.addRow(row);

}

}

**void fillTableDoanhThuHomNay()** {//Điền dữ liệu vào bảng doanh thu theo ngày hôm nay tương ứng

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

model.setRowCount(0);

String ngay = lblNgayHomNay.getToolTipText();

List<Object[]> list = tkdao.getDoanhThuTheoNgay(ngay);

for (Object[] row : list) {//cho vòng lặp duyệt qua mảng líst

model.addRow(row);

}

}

**void fillTableSanPhamHomNay/thang/ngay()** {//điền dữ liệu vào bảng Sản phẩm bán được

List<Object[]> list = tkdao.getSanPhamTheoNgay(ngay/thang);

}

**void fillTableSPKhBanDuocHomNayThang/Ngay()** {//điền dữ liệu vào bảng Sản phẩm **KHÔNG** bán được

List<Object[]> list = tkdao.getSanPhamKhBanDuocTheoThang(thang/ngay);

}

**void fillTableSPBanChayHomnay/Thang/Ngay()** {//điền dữ liệu vào bảng Sản phẩm bán chạy

List<Object[]> list = tkdao.getSPBanChayTheoNgay(ngay/thang);

}

**void fillTableNhanVienHomNay/Thang/Ngay()** {//Điền dữ liệu vào bảng nhân viên có đơn

List<Object[]> list = tkdao.getNhanVienThang(thang/ngay);

}

# Kiểm thử

Kiểm thử phần mềm là quá trình thực thi 1 chương trình với mục đích tìm ra lỗi.

Kiểm thử phần mềm đảm bảo sản phẩm phần mềm đáp ứng chính xác, đầy đủ và đúng theo yêu cầu của khách hàng, yêu cầu của sản phẩm đề đã đặt ra.

Kiểm thử phần mềm cũng cung cấp mục tiêu, cái nhìn độc lập về phần mềm, điều này cho phép việc đánh giá và hiểu rõ các rủi ro khi thực thi phần mềm.

Kiểm thử phần mềm tạo điều kiện cho bạn tận dụng tối đa tư duy đánh giá và sáng tạo để bạn có thể phát hiện ra những điểm mà người khác chưa nhìn thấy

**MỤC ĐÍCH KIỂM THỬ :**

Xác định những lỗi và khiếm khuyết có thể xảy ra trong quá trình phát triển phần mềm

Khắc phục những lỗi của phần mềm để đưa đến trải nghiệm tốt nhất cho người dùng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Chức năng** | **Trường hợp** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| 1 | Giao diện | Giao diện | Giao diện dễ nhìn, thân thiện với người dùng | Giao diện dễ nhìn, thân thiện với người dùng | Pass |
| Phân chia bố cục | Phân chia bố cục hợp lí | Phân chia bố cục hợp lí | Pass |
| Vị trí các thành phần trong form | Vị trí các thành phần trong form hợp lí, thông minh | Vị trí các thành phần trong form hợp lí, thông minh | Pass |
| Kiểm tra tất cả lỗi về chính tả, cấu trúc câu, ngữ pháp trên màn hình | Kiểm tra tất cả lỗi về chính tả, cấu trúc câu, ngữ pháp trên màn hình đạt yêu cầu | Kiểm tra tất cả lỗi về chính tả, cấu trúc câu, ngữ pháp trên màn hình đạt yêu cầu | Pass |
| Truy cập đến màn hình tương ứng với 1 mục được chọn | Truy cập đến màn hình tương ứng với 1 mục được chọn thành công | Truy cập đến màn hình tương ứng với 1 mục được chọn thành công | Pass |
| Thống nhất 1 Style hiển thị chung | Thống nhất 1 Style hiển thị chung | Thống nhất 1 Style hiển thị chung | Pass |
| 2 | Form Login | Không nhập bất cứ dữ liệu gì vào form | Hiển thị thông báo tên đăng nhập và mật khẩu kh được để trống | Hiển thị thông báo tên đăng nhập và mật khẩu kh được để trống | Pass |
| Nhập username, không nhập password và ngược lại | Hiển thị thông báo mật khẩu không được để trống và ngược lại | Hiển thị thông báo mật khẩu không được để trống và ngược lại | Pass |
| Nhập sai username hoặc mật khẩu | Hiển thị thông báo Sai tên đăng nhập hoặc Sai mật khẩu | Hiển thị thông báo Sai tên đăng nhập hoặc Sai mật khẩu | Pass |
| Nhập đúng username và password | Chuyển tới màn hình MainFrame | Chuyển tới màn hình MainFrame | Pass |
| Click vào label Quên mật khẩu ? | Hiển thị Form Quên mật khẩu | Hiển thị Form Quên mật khẩu | Pass |
| Click chọn hoặc không vào checkBox showpassword ? | Hiển thị mật khẩu và ngược lại | Hiển thị mật khẩu và ngược lại | Pass |
| Nhập đúng Username và Password | Chạy màn hình chào 🡪 vào MainFrame | Chạy màn hình chào 🡪 vào MainFrame | Pass |
| 3 | Form Quên mật khẩu | Bỏ trống 1 trong các trường dữ liệu | Hiển thị thông báo Bạn chưa nhập đầy đủ thông tin | Hiển thị thông báo Bạn chưa nhập đầy đủ thông tin | Pass |
| Nhập sai tên đăng nhập | Hiển thị thông báo Tên đăng nhập không hợp lệ | Hiển thị thông báo Tên đăng nhập không hợp lệ | Pass |
| Nhập email và nhấn nút SendCode | Gửi mail mã xác nhận đến email và hiển thị thông báo Email đã được gửi | Gửi mail mã xác nhận đến email và hiển thị thông báo Email đã được gửi | Pass |
| Nhập sai mã xác nhận từ mail gửi về | Hiển thị thông báo mã xác nhận không đúng | Hiển thị thông báo mã xác nhận không đúng | Pass |
| Nhập xác nhận mật khẩu không trùng khớp với mật khẩu mới | Hiển thị thông báo Mật khẩu không khớp | Hiển thị thông báo Mật khẩu không khớp | Pass |
| Click vào Icon Refresh | Đổi mã xác nhận | Đổi mã xác nhận | Pass |
| Click vào icon Show hoặc Hide | Hiển thị hoặc ẩn mật khẩu | Hiển thị hoặc ẩn mật khẩu | Pass |
| Click vào label Click here to login | Hiển thị Form Login | Hiển thị Form Login | Pass |
| 4 | Form MainFrame | Nhấn vào từng mục trong Menu | Chuyển đến màn hình tương ứng | Chuyển đến màn hình tương ứng | Pass |
| Khi đăng nhập vào sẽ hiển thị Username và Vai trò của người đăng nhập | Khi đăng nhập vào sẽ hiển thị Username và Vai trò của người đăng nhập | Khi đăng nhập vào sẽ hiển thị Username và Vai trò của người đăng nhập | Pass |
| Kiểm tra vai trò đăng nhập, nếu là Quản lí thì có thể xem thống kê và ngược lại | Quản lí sẽ được phép xem thống kê còn nhân viên thì không | Quản lí sẽ được phép xem thống kê còn nhân viên thì không | Pass |
| Click vào button Đăng xuất | Hiển thị hộp chọn Yes/No để đăng xuất hoặc không | Hiển thị hộp chọn Yes/No để đăng xuất hoặc không | Pass |
| Click vào lebel Website Store | Mở trang web của cửa hàng | Mở trang web của cửa hàng | Pass |
| Click vào label Change Password | Hiển thị Form đổi mật khẩu | Hiển thị Form đổi mật khẩu | Pass |
| Click vào label Exit | Hiển thị hộp chọn yes/no để thoát hoặc không | Hiển thị hộp chọn yes/no để thoát hoặc không | Pass |
| Click vào label Info | Hiển thị form thông tin của người đăng nhập | Hiển thị form thông tin của người đăng nhập | Pass |
|  |  |  |  |
| 5 | Panel SanPham | Để trống một trong các trường dữ liệu | Hiển thị thông báo Bạn chưa nhập đầy đủ thông tin | Hiển thị thông báo Bạn chưa nhập đầy đủ thông tin | Pass |
| Nhập mã sản phẩm trùng với mã sản phẩm đã có trong Database | Hiển thị thông báo Mã sản phẩm trùng, ô mã sản phẩm chuyển sang màu hồng | Hiển thị thông báo Mã sản phẩm trùng, ô mã sản phẩm chuyển sang màu hồng | Pass |
| Nhập đơn giá bán, số lượng là chữ | Hiển thị thông báo Phải là số | Hiển thị thông báo Phải là số | Pass |
| Nhập đơn giá bán, số lượng là số âm | Hiển thị thông báo Phải lớn hơn 0 | Hiển thị thông báo Phải lớn hơn 0 | Pass |
| Để trống hình ảnh | Hiển thị thống báo Không được để trống hình ảnh | Hiển thị thống báo Không được để trống hình ảnh | Pass |
| Click vào 1 trong các nút điều hướng | Duyệt đến từng bản ghi trong table tương ứng | Duyệt đến từng bản ghi trong table tương ứng | Pass |
| Click chọn vào 1 bản ghi trên bảng | Chuyển đến tab Cập nhật, đưa dữ liệu lên Form | Chuyển đến tab Cập nhật, đưa dữ liệu lên Form | Pass |
| Nhập dữ liệu vào ô tìm kiếm | Tìm kiếm tên sản phẩm theo từ khóa tương ứng | Tìm kiếm tên sản phẩm theo từ khóa tương ứng | Pass |
| Để trống tên danh mục sản phẩm | Hiển thị thông báo bạn chưa nhập tên danh mục sản phẩm | Hiển thị thông báo bạn chưa nhập tên danh mục sản phẩm | Pass |
| Chọn một trong các bản ghi trên bảng sản phẩm và thực hiện cập nhật hoặc xóa | Cập nhật hoặc xóa khỏi CSDL | Cập nhật hoặc xóa khỏi CSDL | Pass |
| Chọn một trong các bản ghi trên bảng danh mục sản phẩm và thực hiện cập nhật hoặc xóa | Cập nhật hoặc xóa khỏi CSDL | Cập nhật hoặc xóa khỏi CSDL | Pass |
| Click vào button New | Clear Form, nút Sửa,xóa và các nút điều hướng không dùng được | Clear Form, nút Sửa,xóa và các nút điều hướng không dùng được | Pass |
| 6 | Panel Nhân viên | Nếu người dùng đăng nhập với vai trò Quản lí hoặc nhân viên | Nếu là quản lí thì sẽ xem được mật khẩu và ngược lại | Nếu là quản lí thì sẽ xem được mật khẩu và ngược lại | Pass |
| Để trống một trong các trường dữ liệu | Hiển thị thông báo Bạn chưa nhập đầy đủ thông tin | Hiển thị thông báo Bạn chưa nhập đầy đủ thông tin | Pass |
| Nhập mã nhân viên trùng với mã nhân viên đã có trong Database | Hiển thị thông báo Mã nhân viên trùng, ô mã nhân viên chuyển sang màu hồng | Hiển thị thông báo Mã nhân viên trùng, ô mã nhân viên chuyển sang màu hồng | Pass |
| Nhập sai định dạng số điện thoại | Hiển thị thông báo Số điện thoại không hợp lệ | Hiển thị thông báo Số điện thoại không hợp lệ | Pass |
| Nhập xác nhận mật khẩu không khớp với mật khẩu | Hiển thị thông báo Mật khẩu không khớp,ô xác nhận mật khẩu chuyển sang màu hồng | Hiển thị thông báo Mật khẩu không khớp,ô xác nhận mật khẩu chuyển sang màu hồng | Pass |
| Click vào 1 trong các nút điều hướng | Duyệt đến từng bản ghi trong table tương ứng | Duyệt đến từng bản ghi trong table tương ứng | Pass |
| Click chọn vào 1 bản ghi trên bảng | Đưa dữ liệu lên Form, nút Edit, Delete dùng được còn nút Save không dùng được | Đưa dữ liệu lên Form, nút Edit, Delete dùng được còn nút Save không dùng được | Pass |
| Chọn một trong các bản ghi trên bảng nhân viên và thực hiện cập nhật hoặc xóa | Hiển thị hộp chọn hỏi người dùng. Cập nhật hoặc xóa khỏi CSDL | Hiển thị hộp chọn hỏi người dùng. Cập nhật hoặc xóa khỏi CSDL | Pass |
| Click vào button New | Clear Form, nút Sửa,xóa và các nút điều hướng không dùng được | Clear Form, nút Sửa,xóa và các nút điều hướng không dùng được | Pass |
| 7 | Panel KhachHang | Để trống một trong các trường dữ liệu | Hiển thị thông báo Bạn chưa nhập đầy đủ thông tin | Hiển thị thông báo Bạn chưa nhập đầy đủ thông tin | Pass |
| Nhập mã khách hàng trùng với Mã khách hàng đã có trong Database | Hiển thị thông báo Mã khách hàng trùng, ô mã khách hàng chuyển sang màu hồng | Hiển thị thông báo Mã khách hàng trùng, ô mã khách hàng chuyển sang màu hồng | Pass |
| Nhập sai định dạng số điện thoại | Hiển thị thông báo Số điện thoại không hợp lệ | Hiển thị thông báo Số điện thoại không hợp lệ | Pass |
| Nhập sai định dạng Email | Hiển thị thông báo Email không hợp lệ | Hiển thị thông báo Email không hợp lệ | Pass |
| Click vào 1 trong các nút điều hướng | Duyệt đến từng bảng ghi trong table tương ứng | Duyệt đến từng bảng ghi trong table tương ứng | Pass |
| Click chọn vào 1 bản ghi trên bảng | Đưa dữ liệu lên Form, nút Edit, Delete dùng được còn nút Save không dùng được | Đưa dữ liệu lên Form, nút Edit, Delete dùng được còn nút Save không dùng được | Pass |
| Chọn một trong các bản ghi trên bảng khách hàng và thực hiện cập nhật hoặc xóa | Hiển thị hộp chọn hỏi người dùng. Cập nhật hoặc xóa khỏi CSDL | Hiển thị hộp chọn hỏi người dùng. Cập nhật hoặc xóa khỏi CSDL | Pass |
| Click vào button New | Clear Form, nút Sửa,xóa và các nút điều hướng không dùng được | Clear Form, nút Sửa,xóa và các nút điều hướng không dùng được | Pass |
| Nhập dữ liệu vào ô tìm kiếm | Hiển thị thông tin khách hàng trên bảng theo từ khóa nhập vào | Hiển thị thông tin khách hàng trên bảng theo từ khóa nhập vào | Pass |
| 8 | Panel HoaDon | Tự động điền mã nhân viên, tên nhân viên là người đăng nhập và đang sử dụng phần mềm | Tự động điền mã nhân viên và tên nhân viên là người đăng nhập và đang sử dụng phần mềm | Tự động điền mã nhân viên và tên nhân viên là người đăng nhập và đang sử dụng phần mềm | Pass |
| Tự động điền ngày bán là ngày hiện tại | Tự động điền ngày bán là ngày hiện tại | Tự động điền ngày bán là ngày hiện tại | Pass |
| Chọn một khách hàng bất kì trong combobox khachhang | Thông tin Mã khách hàng, Địa chỉ, Điện thoại tự động thay đổi theo khách hàng tương ứng | Thông tin Mã khách hàng, Địa chỉ, Điện thoại tự động thay đổi theo khách hàng tương ứng | Pass |
| Để trống ô Mã hóa đơn | Hiển thị thông báo Mã hóa đơn không được để trống | Hiển thị thông báo Mã hóa đơn không được để trống | Pass |
| Chọn một trong các bản ghi trên bảng hóa đơn và thực hiện cập nhật hoặc xóa | Hiển thị hộp chọn hỏi người dùng. Cập nhật hoặc xóa khỏi CSDL | Hiển thị hộp chọn hỏi người dùng. Cập nhật hoặc xóa khỏi CSDL | Pass |
| Chọn một trong các bản ghi trên bảng hóa đơn | Chọn một trong các bản ghi trên bảng hóa đơn, chuyển sang tab HoaDonChiTiet | Chọn một trong các bản ghi trên bảng hóa đơn, chuyển sang tab HoaDonChiTiet | Pass |
| Click vào button New | Clear Form, nút Sửa,xóa và các nút điều hướng không dùng được | Clear Form, nút Sửa,xóa và các nút điều hướng không dùng được | Pass |
| Click vào 1 trong các nút điều hướng | Duyệt đến từng bản ghi trong table tương ứng | Duyệt đến từng bản ghi trong table tương ứng | Pass |
| Nhập mã hóa đơn và nhấn nút Save | Hiển thị thông báo hỏi người dùng tiếp tục thêm sản phẩm vào hóa đơn, chuyển sang tab HoaDonChiTiet | Hiển thị thông báo hỏi người dùng tiếp tục thêm sản phẩm vào hóa đơn, chuyển sang tab HoaDonChiTiet | Pass |
| Chọn sản phẩm bất kì trong combobox SanPham | Thông tin Mã sản phẩm, Giá bán tự động thay đổi theo sản phẩm tương ứng | Thông tin Mã sản phẩm, Giá bán tự động thay đổi theo sản phẩm tương ứng | Pass |
| Nhập số lượng hoặc giảm giá là chữ | Hiển thị thông báo Phải là số | Hiển thị thông báo Phải là số | Pass |
| Nhập số lượng hoặc giảm giá là số âm | Hiển thị thông báo Phải lớn hơn 0 | Hiển thị thông báo Phải lớn hơn 0 | Pass |
| Nhập số lượng, giảm giá là số lớn hơn 0 | Tự động tính và hiển thị lên ô thành tiền | Tự động tính và hiển thị lên ô thành tiền | Pass |
| Chọn một trong các bản ghi trên bảng hóa đơn và thực hiện cập nhật hoặc xóa | Hiển thị hộp chọn hỏi người dùng. Cập nhật hoặc xóa khỏi CSDL | Hiển thị hộp chọn hỏi người dùng. Cập nhật hoặc xóa khỏi CSDL | Pass |
| Click vào nút Xong | Hiển thị thông báo Hoàn thành, chuyển sang tab HoaDon và clear form và table HoaDonChiTiet | Hiển thị thông báo Hoàn thành, chuyển sang tab HoaDon và clear form và table HoaDonChiTiet | Pass |
| Click vào nút in | Hiển thị form In cho phép nhân viên xem lại hóa đơn. Có thể in ra hoặc không | Hiển thị form In cho phép nhân viên xem lại hóa đơn. Có thể in ra hoặc không | Pass |
| Thực hiện thêm, sửa, hoặc xóa HoaDonChiTiet | Tự động tính toán thay đổi tổng tiền | Tự động tính toán thay đổi tổng tiền | Pass |
| 9 | Dialog ThongKe | Click vào label Ngày hôm nay | Hiển thị thông tin các bảng theo ngày hôm nay | Hiển thị thông tin các bảng theo ngày hôm nay | Pass |
| Nhập dữ liệu vào ô tháng là chữ | Hiển thị thông báo phải là số | Hiển thị thông báo phải là số | Pass |
| Nhập dữ liệu số khách số từ 1-12 | Hiển thị thông báo tháng phải từ 1-12 | Hiển thị thông báo tháng phải từ 1-12 | Pass |
| Nhập đúng tháng | Hiển thị thông tin các bảng theo ngày hôm tháng | Hiển thị thông tin các bảng theo ngày hôm tháng | Pass |
| Chọn ngày | Hiển thị thông tin các bảng theo ngày được chọn | Hiển thị thông tin các bảng theo ngày được chọn | Pass |
| Click vào nút Hủy | Đóng form ThongKe | Đóng form ThongKe | Pass |
| Click vào nút Xuất báo cáo | Xuất báo cáo ra file Excel | Chưa xuất được báo cáo | NA |

# Đóng gói và triển khai

## Hướng dẫn chuyển đổi jar thành exe

-Sử dụng phần mềm LAUNCH4J để chuyển đổi jar sang exe

-Sử dụng phần mềm InnoSetup để đóng gói

## Hướng dẫn cài đặt triển khai

-Cài SQL Server 2008 trở lên

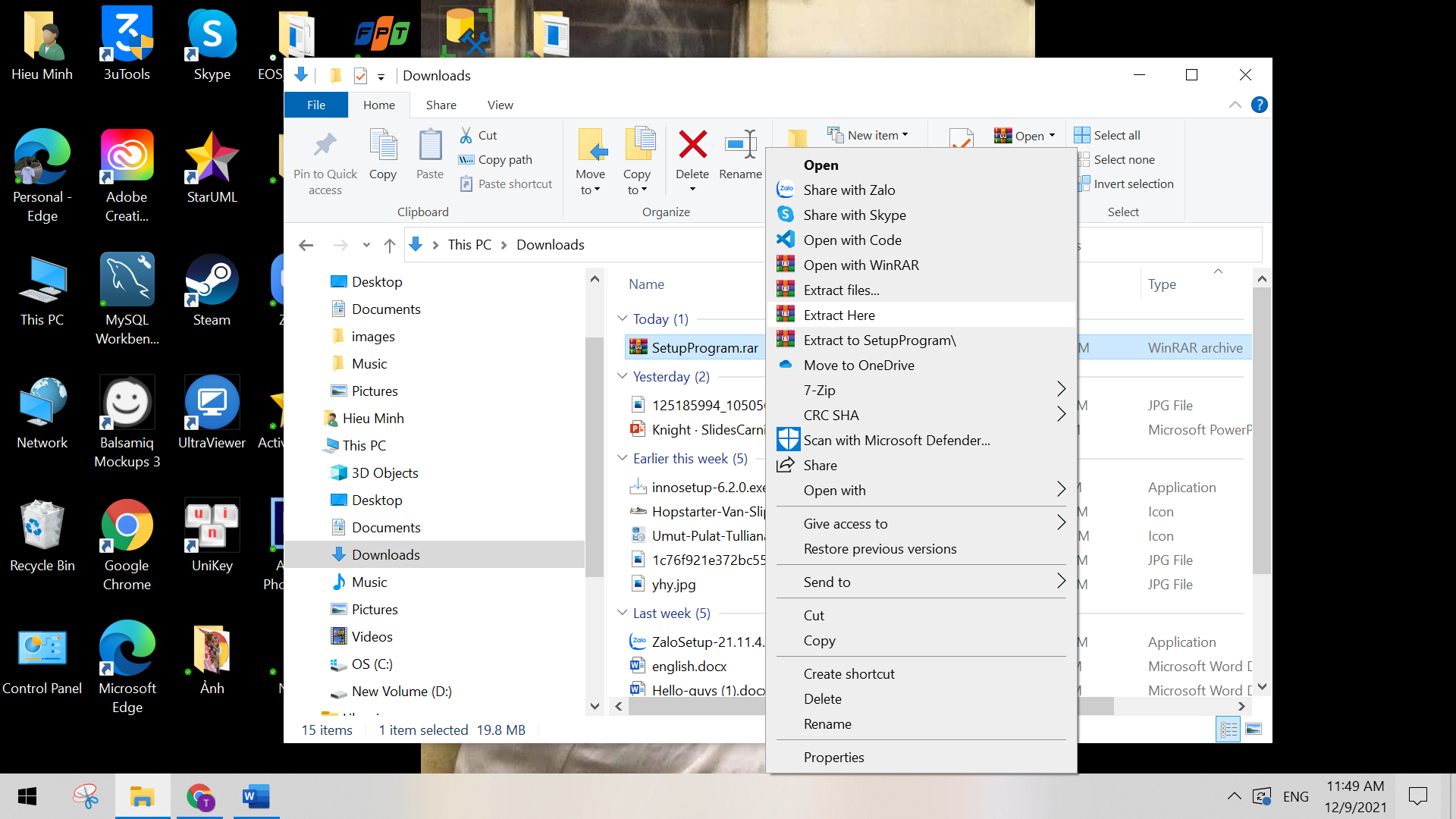
-Tạo CSDL QuanLiBangHang bằng cách chạy file QuanLiBanHang.sql

-Cài đặt JDK 1.8 trở lên

-Chạy file SetupProgram.exe

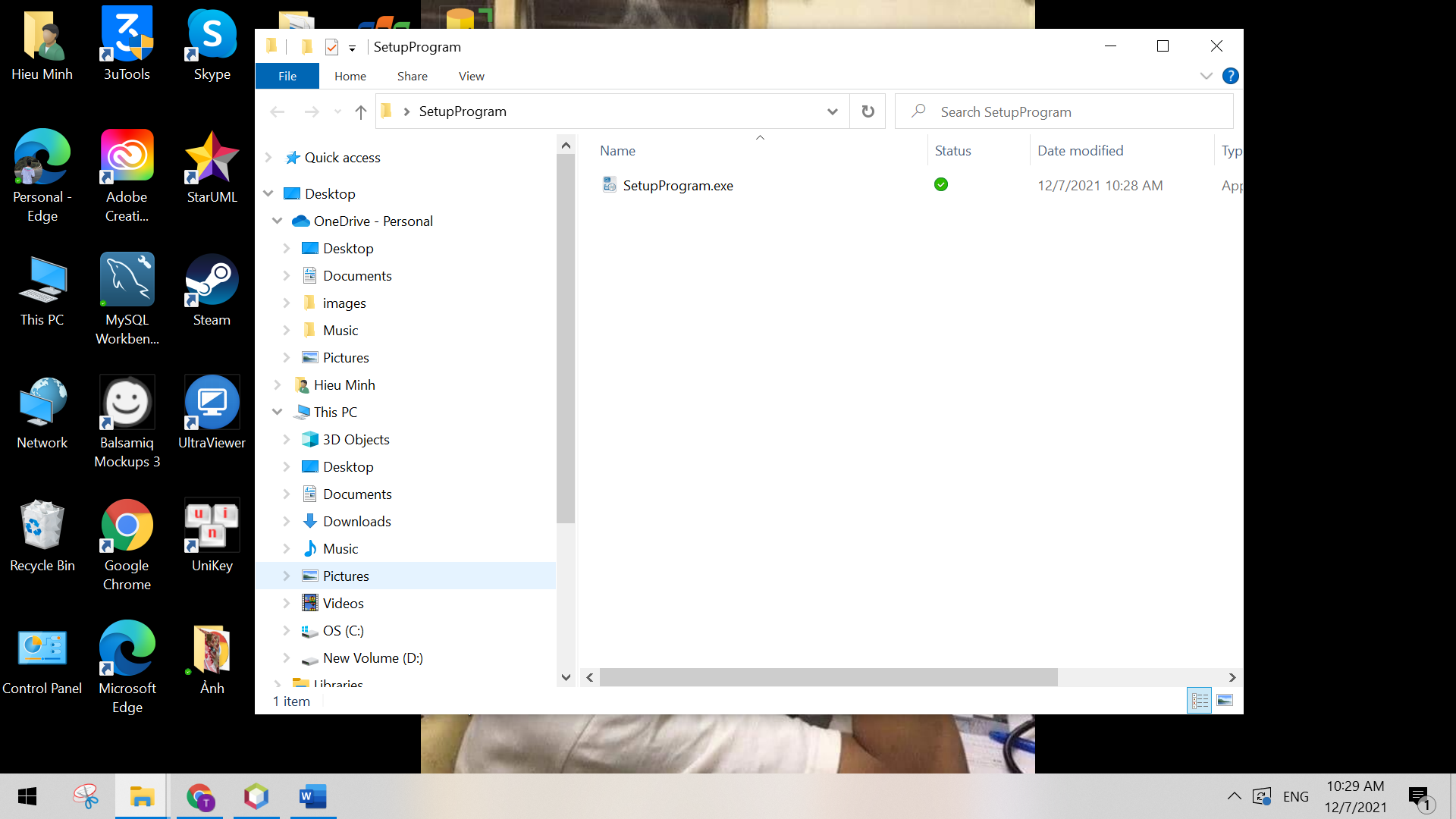
Hướng dẫn cài đặt sử dụng phần mềm :

**+B1 : Tải và giải nén File SetupProgram**



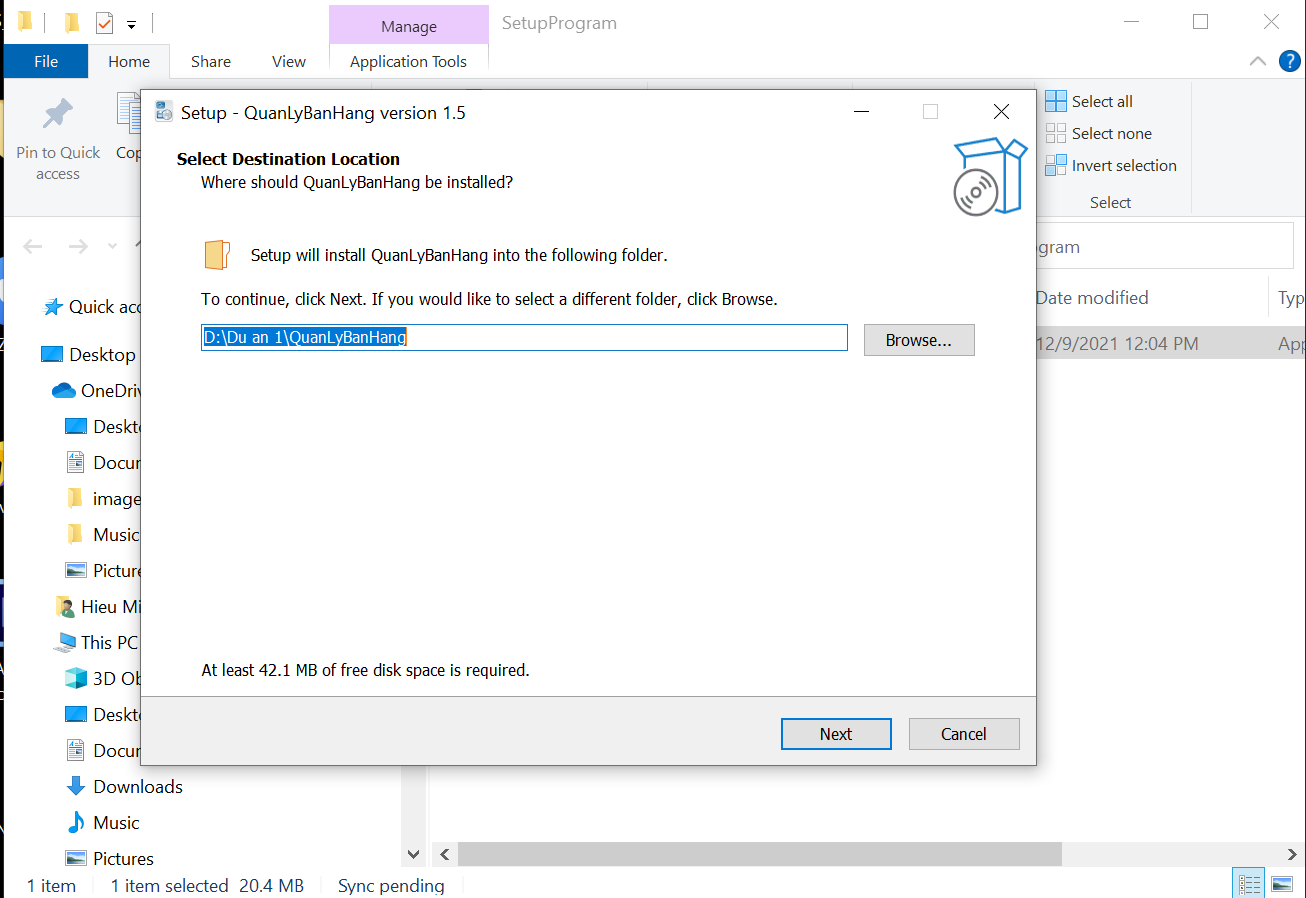
Giải nén file SetupProgram

**+B2 : Chạy file SetupProgram và cài đặt theo hướng dẫn :**



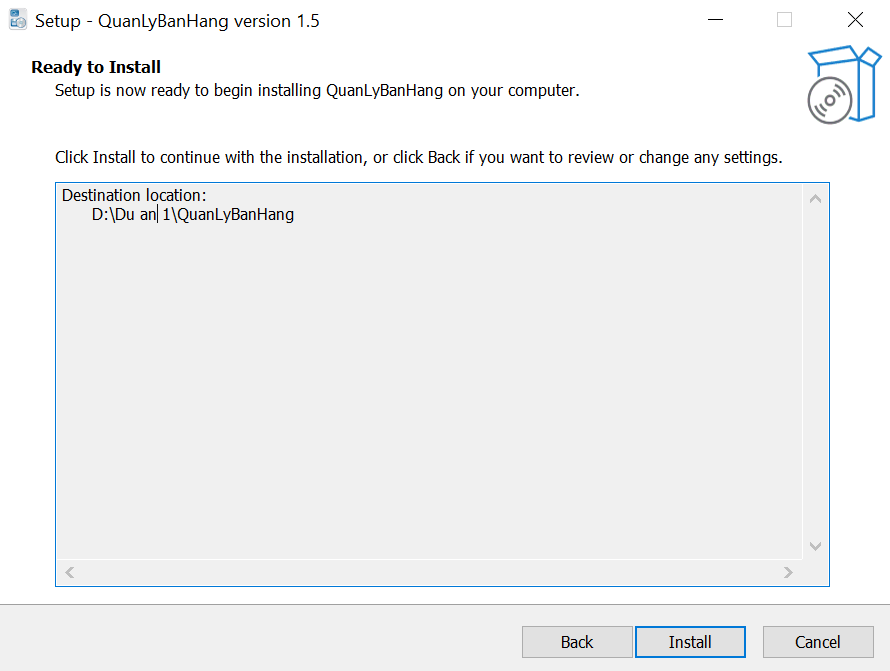
Chạy SetupProgram

+B3 : Chọn nơi lưu thư mục chứa phần mềm



Chọn đường dẫn chứa file phần phần mềm

**+B4 : Nhấn Install để cài đặt phần mềm**



Nhấn Install để tiến hàng cài đặt phần mềm

## Hướng dẫn sử dụng phần mềm

- Đăng nhập với tài khoản username: sa và password:songlong

- Cấp các tài khoản có vai trò là nhân viên hoặc quản lí

- Đăng nhập tài khoản mới và xóa tài khoản hieu

- Xem hướng dẫn sử dụng trong menu trợ giúp của phần mềm

# Kết luận

## Kết luận

Do trình độ và thời gian thực hiện dự án có hạn nên dự án đạt được những kết quả như sau:

-Phân tích toàn bộ bài toán từ bước phân tích nghiệp vụ đến việc phân tích dữ liệu và các chức năng của hệ thống.

- Hiểu rõ quy trình làm việc thực hiện dự án công nghệ.

- Nâng cao kiến thức, kỹ năng làm việc khi thực hiện dự án.

- Xử lí nghiệp vụ hiệu quả hơn.

- Xây dựng được phần mềm quản lí đáp ứng tương đối kì vọng đặt ra.

- Kĩ năng lập trình, quản lí thời gian được phát triển.

- Tạo dựng được các tài liệu báo cáo liên quan đến ứng dụng

- Đóng gói sản phẩm phần mềm đưa đến người dùng dễ dàng.

- Sau khi thực hiện dự án và được sự giúp đỡ của thầy (cô), em đã trau dồi cho bản thân khả năng tư duy logic, có thể nghiên cứu độc lập một vấn đề.

## Hướng phát triển

Đây là một bài toán có nhiều tiềm năng trong quá trình hội nhập, để phát triển thành một hệ thống hoàn chỉnh và có thể đưa ra ứng dụng thực tế một cách rộng rãi chương trình cần:

* Cải tiến, hoàn thiện một số chức năng chưa hoàn chỉnh trong chương trình
* Nâng cấp hệ thống để có thể áp dụng trong việc kinh doanh trên mạng diện rộng và sử dụng được trên các hệ quản trị khác.
* Thiết kế giao diện chương trình trở nên chuyên nghiệp hơn.
* Bổ sung thêm những chức năng cần thiết khác cho ứng dụng
* Phần mềm ứng dụng được áp dụng cho hầu hết các doanh nghiệp kinh doanh , chứ không chỉ là doanh nghiệp vừa và nhỏ.

Với những kiến thức đã học được trong quá trình học tập, tham khảo các tài liệu chuyên nghành cùng với sự nỗ lực của bản thân và sự giúp đỡ, hướng dẫn nhiệt tình của Thầy. Bản thân em đã xây dựng được một phần mềm QUẢN LÍ BÁN HÀNG, chương trình tuy chưa phải là một sản phẩm phần mềm hoàn hảo nhưng cũng góp phần thể hiện một hướng nghiên cứu mới cho ngành CÔNG NGHỆ THÔNG TIN.

RẤT MONG NHẬN ĐƯỢC NHỮNG ĐÁNH GIÁ, GÓP Ý CỦA THẦY CÔ VÀ CÁC BẠN ĐỂ CHƯƠNG TRÌNH NGÀY CÀNG HOÀN THIỆN HƠN.