



FPT POLYTECHNIC

QUẢN LÝ SHOP THỂ THAO BINGBONG

NGÀNH CÔNG NGHỆ THÔNG TIN (ỨNG DỤNG PHẦN MỀM)

GIẢNG VIÊN HƯỚNG DẪN: NGUYỄN TRUNG KIÊN

NHÓM THỰC HIỆN: NGUYỄN ĐOÀN CHÍ THỨC – PC01573 (Nhóm trưởng)

HUỲNH VĂN ĐẠT – PC01684

BÙI TRỌNG NHÂN – PC01721

NOVEMBER 18, 2021

BINGBONG SPORTS STORE MANAGER

1	Phân tích	5
1.1	Hiện trạng	5
1.2	Yêu cầu hệ thống	5
1.3	Use case	6
2	Thiết kế	7
2.1	Mô hình triển khai	7
2.2	Thiết kế CSDL	8
2.2.1	Sơ đồ quan hệ thực thể	8
2.2.2	Thiết kế chi tiết các thực thể	9
2.3	Thiết kế giao diện	15
2.3.1	Sơ đồ tổ chức giao diện	15
2.3.2	Thiết kế giao diện cho các chức năng nghiệp vụ	15
2.3.2.1	Thiết kế Form Giao Diện Chính	15
2.3.2.2	Thiết kế Form Nhân Viên	16
2.3.2.3	Thiết kế Form Chức Vụ	16
2.3.2.4	Thiết kế Form Thêm Tài Khoản	17
2.3.2.5	Thiết kế Form Sản Phẩm	17
2.3.2.6	Thiết kế Form Đơn Vị Tính	18
2.3.2.7	Thiết kế Form Loại Sản Phẩm	18
2.3.2.8	Thiết kế Form Lập Hóa Đơn	19
2.3.2.9	Thiết kế Form Xem Hóa Đơn	19
2.3.2.10	Thiết kế Form Khách Hàng	20
2.3.2.11	Thiết kế Form Doanh Thu	21
2.3.2.12	Thiết kế Form Login	21
2.3.2.13	Thiết kế Form ForgotPasswordDialog	22
2.3.2.14	Thiết kế Form OTPDialog	22
2.3.2.15	Thiết kế Form ChangePassDialog	23
3	Thực hiện viết mã	24
3.1	Viết mã tạo CSDL	24

3.1.1	Tạo CSDL	24
3.1.2	SQL truy vấn và thao tác	32
3.1.2.1	SQL đối với Bảng Nhân Viên	32
3.1.2.2	SQL đối với Bảng Sản Phẩm	33
3.1.2.3	SQL đối với Bảng Hóa Đơn	33
3.1.2.4	SQL đối với Bảng Hóa Đơn Chi Tiết	33
3.1.2.5	SQL đối với Bảng Khách Hàng	34
3.1.2.6	SQL đối với Bảng Tài Khoản	34
3.1.2.7	SQL đối với Bảng Chức Vụ	34
3.1.2.8	SQL đối với Bảng Loại Sản Phẩm	34
3.1.2.9	SQL đối với Bảng Đơn Vị Tính Sản Phẩm	35
3.1.3	Thủ tục lưu tổng hợp thống kê (SP_ThongKe)	35
3.2	Lập trình JDBC	36
3.2.1	Lớp hỗ trợ	36
3.2.1.1	Lớp tiện ích VNCharacterUtils	36
3.2.1.2	Lớp tiện ích MailSender	36
3.2.1.3	Lớp tiện ích MsgBox	36
3.2.1.4	Lớp tiện ích XImage	37
3.2.1.5	Lớp tiện ích XJdbc	37
3.2.2	Model class - Các lớp mô tả dữ liệu	38
3.2.2.1	Entity Nhân Viên	38
3.2.2.2	Entity Chức Vụ	39
3.2.2.3	Entity Tài Khoản	39
3.2.2.4	Entity Sản Phẩm	40
3.2.2.5	Entity Đơn Vị Tính	40
3.2.2.6	Entity Loại Sản Phẩm	41
3.2.2.7	Entity Hóa Đơn	41
3.2.2.8	Entity Hóa Đơn Chi Tiết	42
3.2.2.9	Entity Khách Hàng	42

3.2.3	DAO Class - Các lớp truy xuất dữ liệu	43
3.2.3.1	Lớp SoftwareDAO	43
3.2.3.2	Lớp Nhân Viên DAO	44
3.2.3.3	Lớp Chức Vụ DAO	45
3.2.3.4	Lớp Tài Khoản DAO	46
3.2.3.5	Lớp Sản Phẩm DAO	47
3.2.3.6	Lớp Đơn Vị Tính DAO	48
3.2.3.7	Lớp Loại Sản Phẩm DAO	49
3.2.3.8	Lớp Hóa Đơn DAO	50
3.2.3.9	Lớp Hóa Đơn Chi Tiết DAO	51
3.2.3.10	Lớp Khách Hàng DAO	52
3.2.3.11	Lớp Doanh Thu DAO	52
3.3	Viết mã cho ứng dụng	53
3.3.1.1	Xử lý Form Quản Lý Nhân Viên	53
3.3.1.2	Xử lý Form Quản Lý Chức Vụ	59
3.3.1.3	Xử lý Form Thêm Tài Khoản	62
3.3.1.4	Xử lý Form Quản Lý Sản Phẩm	63
3.3.1.5	Xử lý Form Quản Lý Đơn Vị Tính	68
3.3.1.6	Xử lý Form Quản Lý Loại Sản Phẩm	71
3.3.1.7	Xử lý Form Lập Hóa Đơn	75
3.3.1.8	Xử lý Form Xem Hóa Đơn	85
3.3.1.9	Xử lý Form Danh Mục Khách Hàng	89
3.3.1.10	Xử lý Form Xem Doanh Thu	93
3.3.1.11	Xử lý Form Login	95
3.3.1.12	Xử lý Form ForgotPasswordDialog	98
3.3.1.13	Xử lý Form OTPDialog	100
3.3.1.14	Xử lý Form ChangePassDialog	102
4	Kiểm thử	105
4.1	Kiểm thử form login dialog:	105

4.2	Kiểm thử form forgot password dialog:	106
4.3	Kiểm thử form otp dialog:	107
4.4	Kiểm thử form change pass dialog:	108
4.5	Kiểm thử form nhan vien	110
4.6	Kiểm thử form chuc vu:	112
4.7	Kiểm thử form them tai khoan:	113
4.8	Kiểm thử form san pham:	114
4.9	Kiểm thử form loai san pham:	115
4.10	Kiểm thử form don vi tinh:	116
4.11	Kiểm thử form khach hang:	117
4.12	Kiểm thử form lap hoa don:	119
4.13	Kiểm thử form xem lai hoa don:	122
4.14	Kiểm thử form doanh thu:	123
5	Đóng gói và triển khai	124
5.1	Hướng dẫn chuyển đổi jar thành exe	124
5.2	Hướng dẫn cài đặt triển khai	135
5.3	Hướng dẫn sử dụng phần mềm	135

1 PHÂN TÍCH

1.1 HIỆN TRẠNG

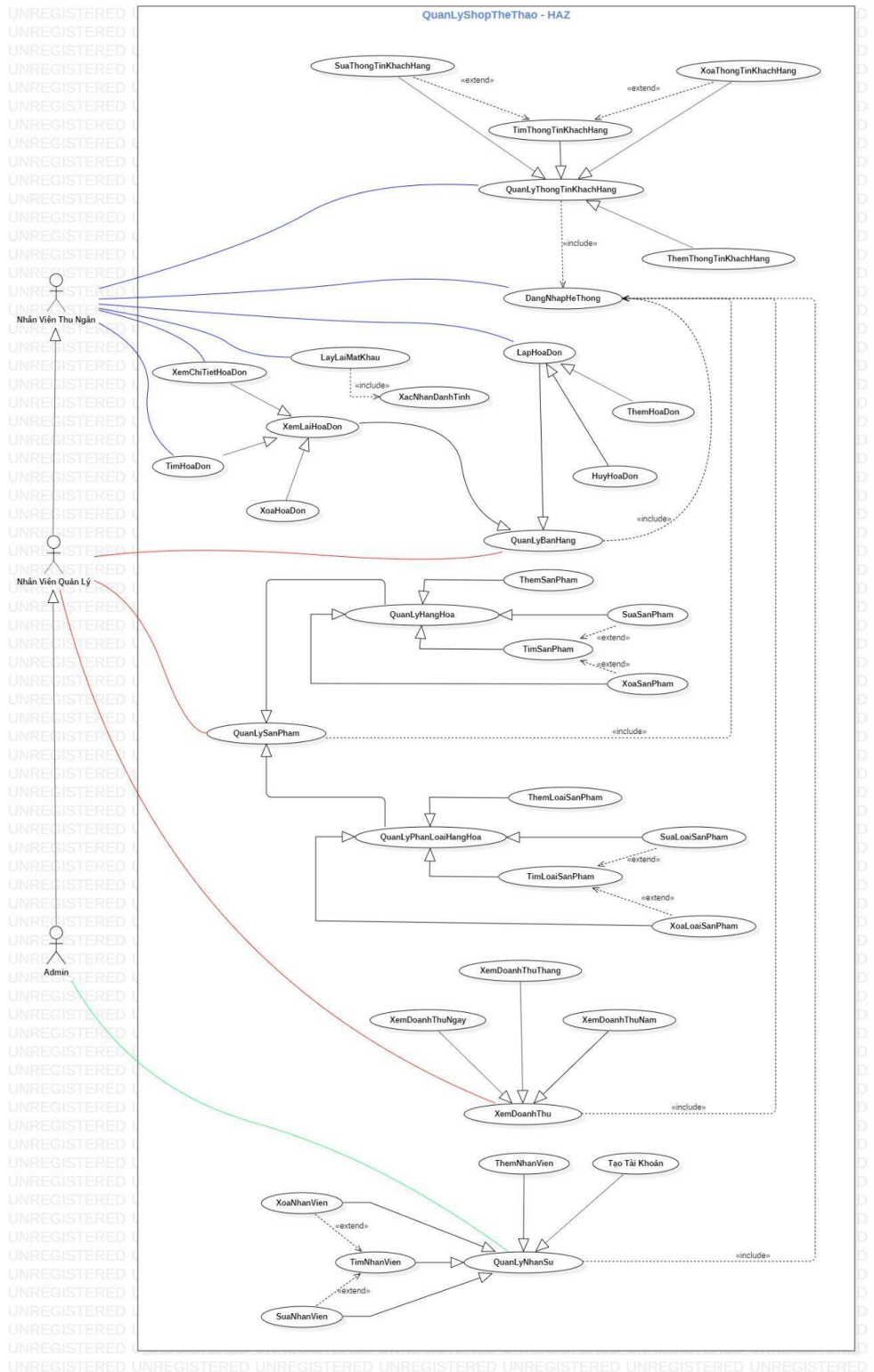
Shop thể thao BingBong là một cửa hàng thể thao chuyên kinh doanh về các mặt hàng như quần, áo, giày và các dụng cụ thể thao.

Shop quản lý bán hàng, thông tin sản phẩm và thông tin nhân viên, khách hàng, doanh thu của shop qua Excel. Với dung lượng dữ liệu ngày càng một tăng và sự bất tiện trong quá trình quản lý và tính doanh thu, vì vậy shop yêu cầu bên nhóm lập trình phần mềm HAZ làm một phần mềm quản lý bán hàng để quản lý dễ dàng hơn xem doanh thu thuận tiện hơn.

1.2 YÊU CẦU HỆ THỐNG

- Phần quản lý được thông tin Nhân Viên, Sản Phẩm, Khách Hàng, xuất và xem Hóa Đơn bán hàng, xem doanh thu theo ngày, theo tháng, theo năm.
- Giao diện thân thiện, màu sắc hài hòa.
- Có phân quyền: Admin, Quản lý và Thu Ngân.

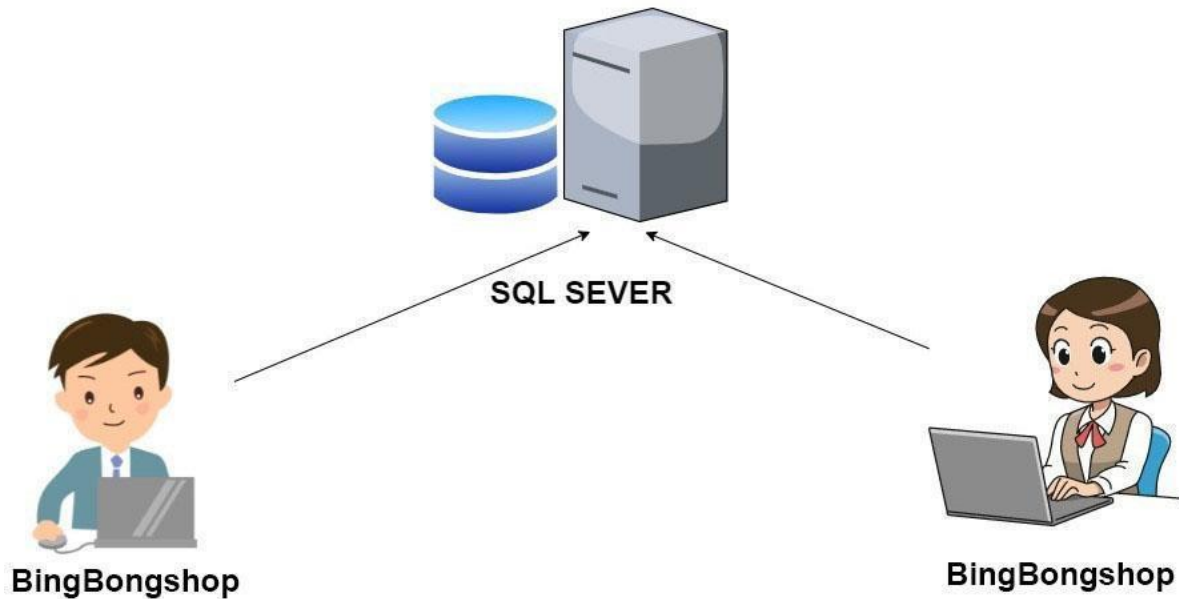
1.3 USE CASE



Sơ Đồ Use Case Chi Tiết (Ctrl + Click Vào Đây)

2 THIẾT KẾ

2.1 MÔ HÌNH TRIỂN KHAI

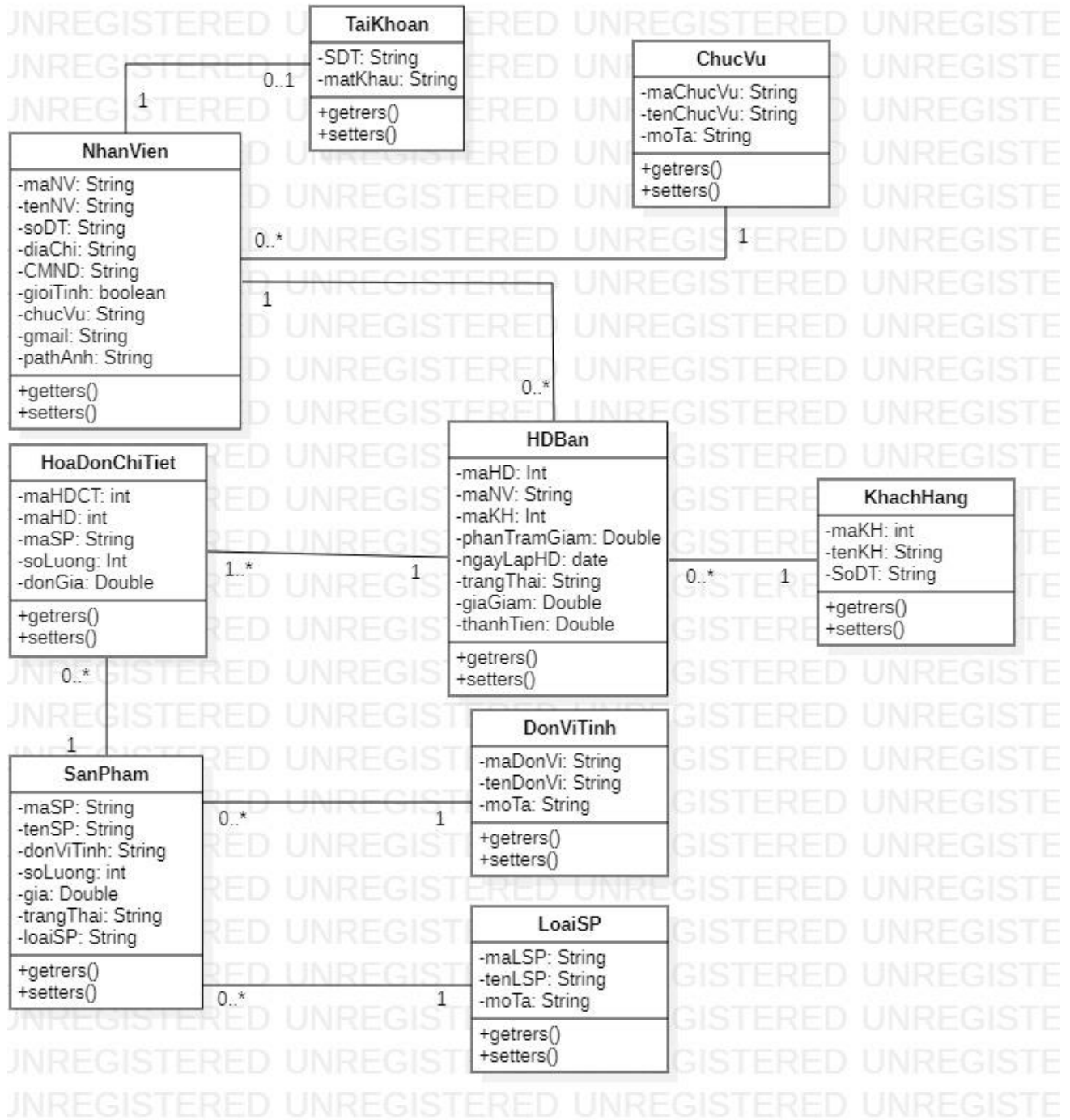


Ứng dụng phần mềm được xây dựng để phục vụ cho nhiều người dùng trong 1 công ty và cơ sở dữ liệu được lưu trữ tập trung.

- + Máy của nhân viên phải cài SQL Server 2008+. Máy này cần hệ điều hành Windows 8.1 trở lên.
- + Các máy nhân viên cài phần mềm BingBongShop. Các máy nhân viên cần JDK 1.8+ với hệ điều hành bất kì.

2.2 THIẾT KẾ CSDL

2.2.1 Sơ đồ quan hệ thực thể



Sơ Đồ Thực Thể Chi Tiết (Ctrl + Click Vào Đây)

2.2.2 Thiết kế chi tiết các thực thể

2.2.2.1 Thực thể Nhân Viên

Nhân Viên		
PK	MaNV TenNV SoDienThoai DiaChi CMND GioiTinh MaChucVu Gmail PathAnh	String String String String String boolean String String String

<i>Thuộc Tính</i>	<i>Kiểu Dữ Liệu</i>	<i>Mô Tả</i>
MaNV	String	Mã nhân viên
TenNV	String	Tên nhân viên
SoDT	String	Số điện thoại nhân viên
Gmail	String	Địa chỉ email của nhân viên
DiaChi	String	Địa chỉ thường trú của nhân viên
ChucVu	String	Chức vụ của nhân viên
GioiTinh	boolean	Giới tính của nhân viên
CMND	String	Chứng minh nhân dân của nhân viên
PathAnh	String	Đường path hình ảnh

2.2.2.2 Thực thể Sản Phẩm

SanPham		
PK	MaSP TenSP DonViTinh SoLuong Gia TrangThai Loai	String String String Integer Double String String

<i>Thuộc Tính</i>	<i>Kiểu Dữ Liệu</i>	<i>Mô Tả</i>
MaSP	String	Mã sản phẩm
TenSP	String	Tên sản phẩm
DonViTinh	String	Đơn vị tính của sản phẩm
SoLuong	Integer	Số lượng sản phẩm
Gia	Double	Đơn giá của sản phẩm
TrangThai	String	Trạng thái của sản phẩm(Ngừng/Dang kinh doanh hoặc sản phẩm mới...)
Loại	String	Phân loại sản phẩm

2.2.2.3 Thực thể Hóa Đơn Bán

HDBan		
PK	MaHD MaNV MaKH PhanTramGiam NgayLapHD TrangThai GiaGiam ThanhTien	Integer String Integer Integer Integer nvarchar(100) Double Double

<i>Thuộc Tính</i>	<i>Kiểu Dữ Liệu</i>	<i>Mô Tả</i>
MaHD	Integer	Mã hóa đơn bán hàng
MaNV	String	Mã nhân viên lập hóa đơn này
MaKH	Integer	Mã khách hàng
PhanTramGiam	Integer	Phần trăm giá giảm đi của hóa đơn
NgayLapHD	String	Ngày lập ra hóa đơn
TrangThai	String	Trạng thái hóa đơn
GiaGiam	Double	Giá giảm của hóa đơn
ThanhTien	String	Thành tiền của hóa đơn

2.2.2.4 Thực thể Hóa Đơn Chi Tiết

ChiTietHoaDon		
PK	MaHDCT	Integer
PK	MaHD	Integer
PK	MaSP	String
	SoLuong	Integer
	DonGia	Double

Thuộc Tính	Kiểu Dữ Liệu	Mô Tả
MaHD	Int	Mã hóa đơn
MaHDCT	Int	Mã hóa đơn chi tiết
MaSP	String	Mã Sản Phẩm
SoLuong	Int	Số lượng sản phẩm
DonGia	Float	Đơn giá của sản phẩm

2.2.2.5 Thực thể Khách Hàng

KhachHang		
PK	MaKH	Integer
	TenKH	String
	SoDT	String

Thuộc Tính	Kiểu Dữ Liệu	Mô Tả
MaKH	Int	Mã Khách Hàng
TenKH	String	Ten Khách Hàng
SoDT	String	Số điện thoại Khách Hàng

2.2.2.6 Thực thể Tài Khoản

TaiKhoan		
PK	SoDienThoai Pass	String String

Thuộc Tính	Kiểu Dữ Liệu	Mô Tả
SoDienThoai	String	Số điện thoại nhân viên
Pass	String	mật khẩu của nhân viên

2.2.2.7 Thực thể Chức Vụ

ChucVu		
PK	MaChucVu TenChucVu MoTa	String String String

Thuộc Tính	Kiểu Dữ Liệu	Mô Tả
MaChucVu	String	Mã Chức Vụ
TenChucVu	String	Tên Chức Vụ
MoTa	String	Mô tả chức vụ

2.2.2.8 Thực thể Loại Sản Phẩm

LoaiSP		
PK	MaLoai TenLoai MoTa	String String String

Thuộc Tính	Kiểu Dữ Liệu	Mô Tả
MaLoai	String	Mã Loại Sản Phẩm
TenLoai	String	Tên Loại Sản Phẩm
MoTa	String	Mô tả sản phẩm

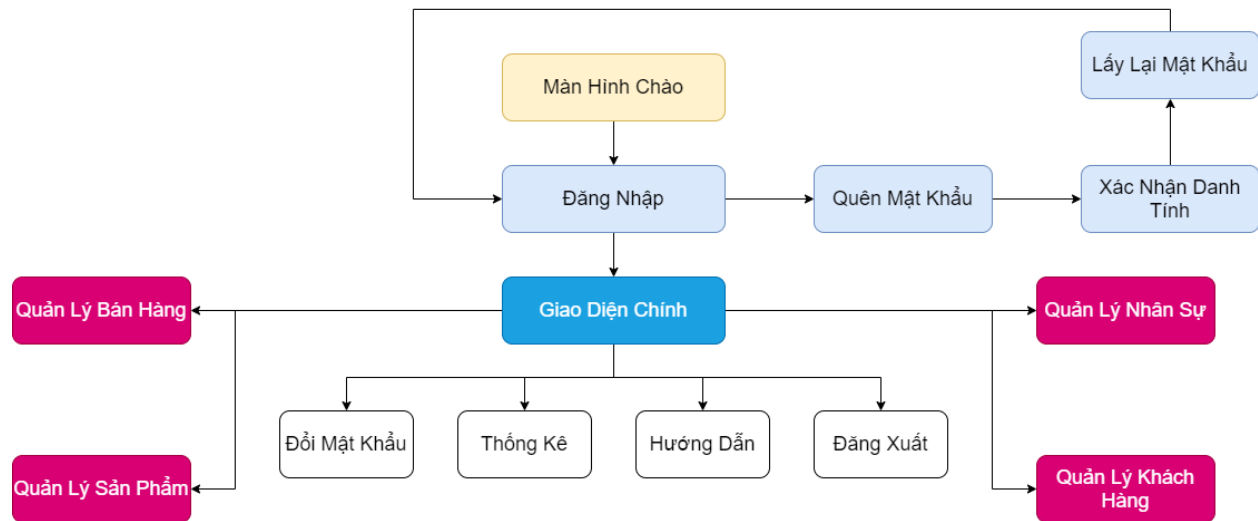
2.2.2.9 Thực thể Đơn Vị Tính Sản Phẩm

DonViTinh		
PK	MaDonVi TenDonVi MoTa	nvarchar(20) nvarchar(255) nvarchar(500)

Thuộc Tính	Kiểu Dữ Liệu	Mô Tả
MaDonVi	String	Mã đơn vị tính
TenDonVi	String	Tên đơn vị tính
MoTa	String	Mô tả đơn vị tính

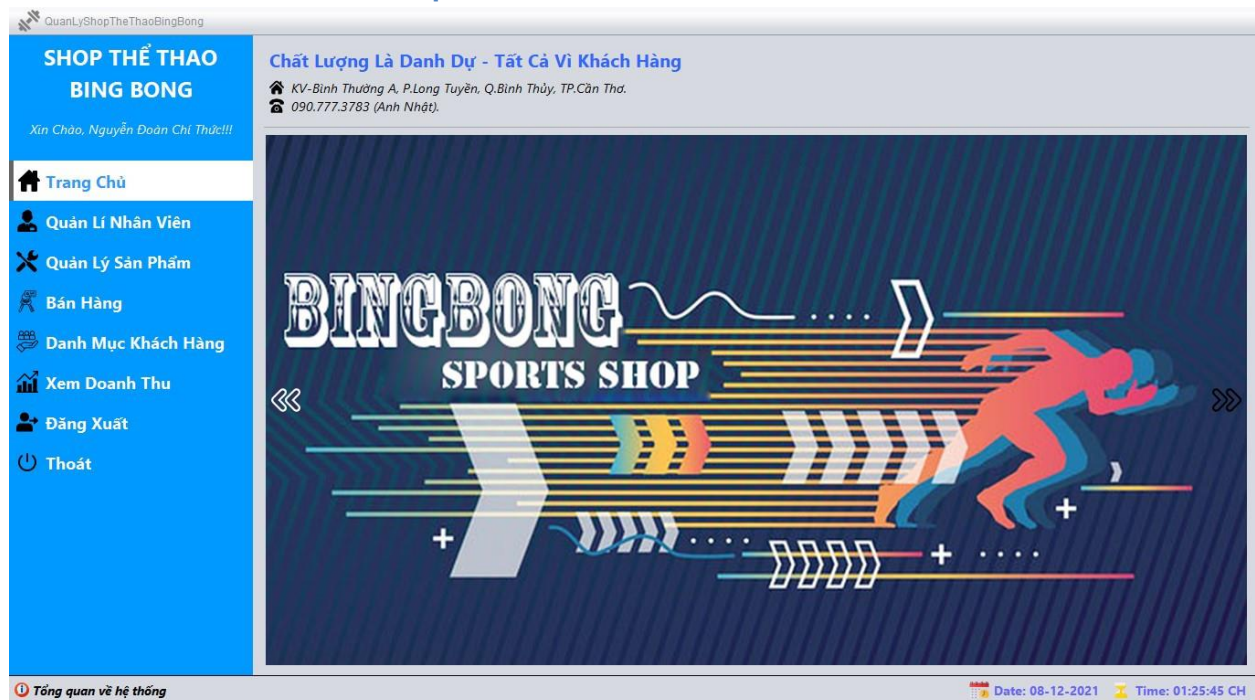
2.3 THIẾT KẾ GIAO DIỆN

2.3.1 Sơ đồ tổ chức giao diện



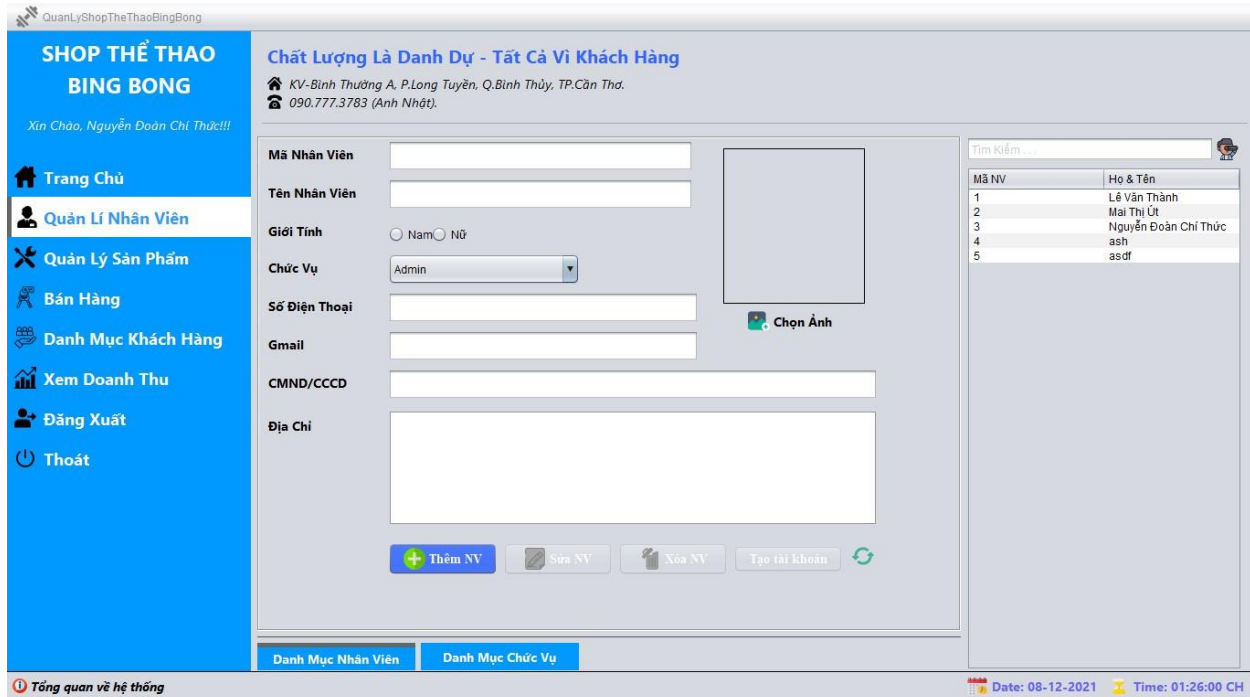
2.3.2 Thiết kế giao diện cho các chức năng nghiệp vụ

2.3.2.1 Thiết kế Form Giao Diện Chính



Mô Tả: Giao diện chính hiển thị các danh mục chức năng của phần mềm.

2.3.2.2 Thiết kế Form Nhân Viên



SHOP THỂ THAO BING BONG
Xin Chào, Nguyễn Đoàn Chí Thúc!!!

Chất Lượng Là Danh Dự - Tất Cả Vì Khách Hàng
KV-Bình Thường A, P.Long Tuyền, Q.Bình Thủy, TP.Cần Thơ.
090.777.3783 (Anh Nhật).

Mã Nhân Viên

Tên Nhân Viên

Giới Tính ☐ Nam ☐ Nữ

Chức Vụ

Số Điện Thoại

Gmail

CMND/CCCD

Địa Chỉ

Chọn Ảnh

Danh Mục Nhân Viên

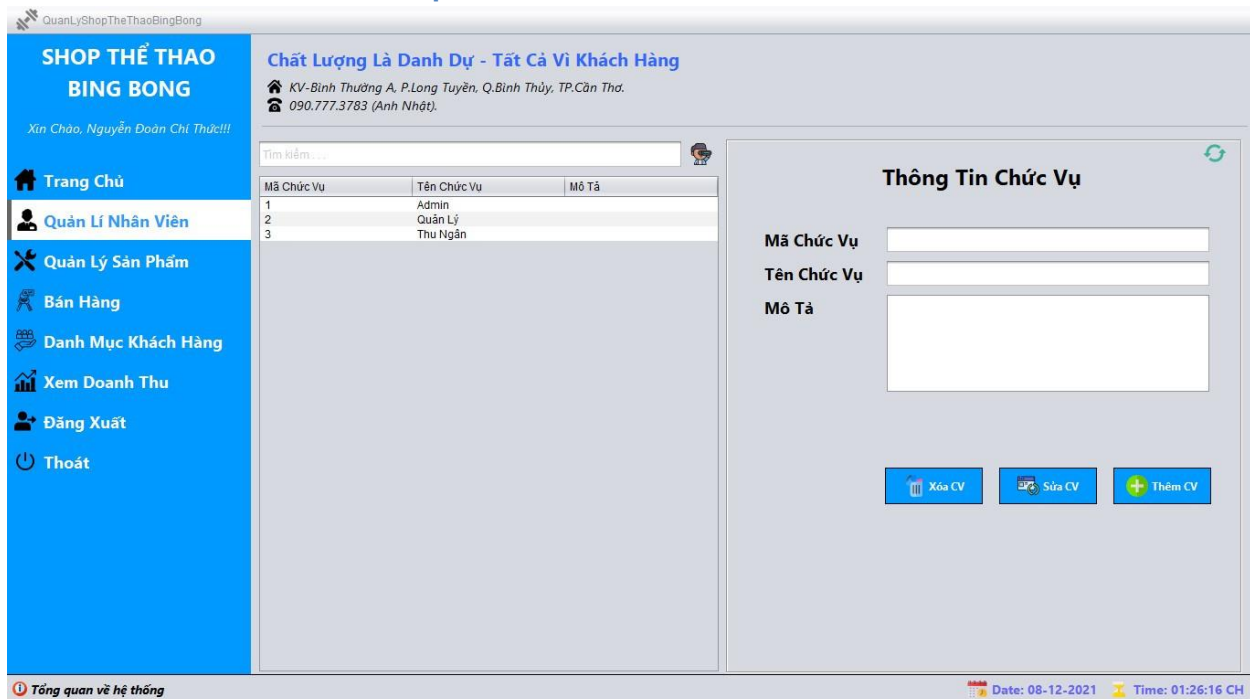
Mã NV	Họ & Tên
1	Lê Văn Thành
2	Mai Thị Út
3	Nguyễn Đoàn Chí Thúc
4	ash
5	asdf

Buttons: + Thêm NV, Sửa NV, Xóa NV, Tạo tài khoản

Footer: Tổng quan về hệ thống, Date: 08-12-2021, Time: 01:26:00 CH

Mô Tả: Giao diện nhân viên bao gồm các chức năng ; chọn nhân viên, thêm nhân viên, sửa nhân viên, xóa nhân viên, làm mới form , tạo tài khoản cho nhân viên.

2.3.2.3 Thiết kế Form Chức Vụ



SHOP THỂ THAO BING BONG
Xin Chào, Nguyễn Đoàn Chí Thúc!!!

Chất Lượng Là Danh Dự - Tất Cả Vì Khách Hàng
KV-Bình Thường A, P.Long Tuyền, Q.Bình Thủy, TP.Cần Thơ.
090.777.3783 (Anh Nhật).

Mã Chức Vụ

Tên Chức Vụ

Mô Tả

Chọn Ảnh

Danh Mục Chức Vụ

Mã Chức Vụ	Tên Chức Vụ	Mô Tả
1	Admin	
2	Quản Lý	
3	Thu Ngân	

Thông Tin Chức Vụ

Mã Chức Vụ

Tên Chức Vụ

Mô Tả

Buttons: Xóa CV, Sửa CV, + Thêm CV

Footer: Tổng quan về hệ thống, Date: 08-12-2021, Time: 01:26:16 CH

Mô Tả: Giao diện chức vụ bao gồm các chức năng , chọn chức vụ, thêm chức vụ, sửa chức vụ, xóa chức vụ, tìm kiếm chức vụ .

2.3.2.4 Thiết kế Form Thêm Tài Khoản

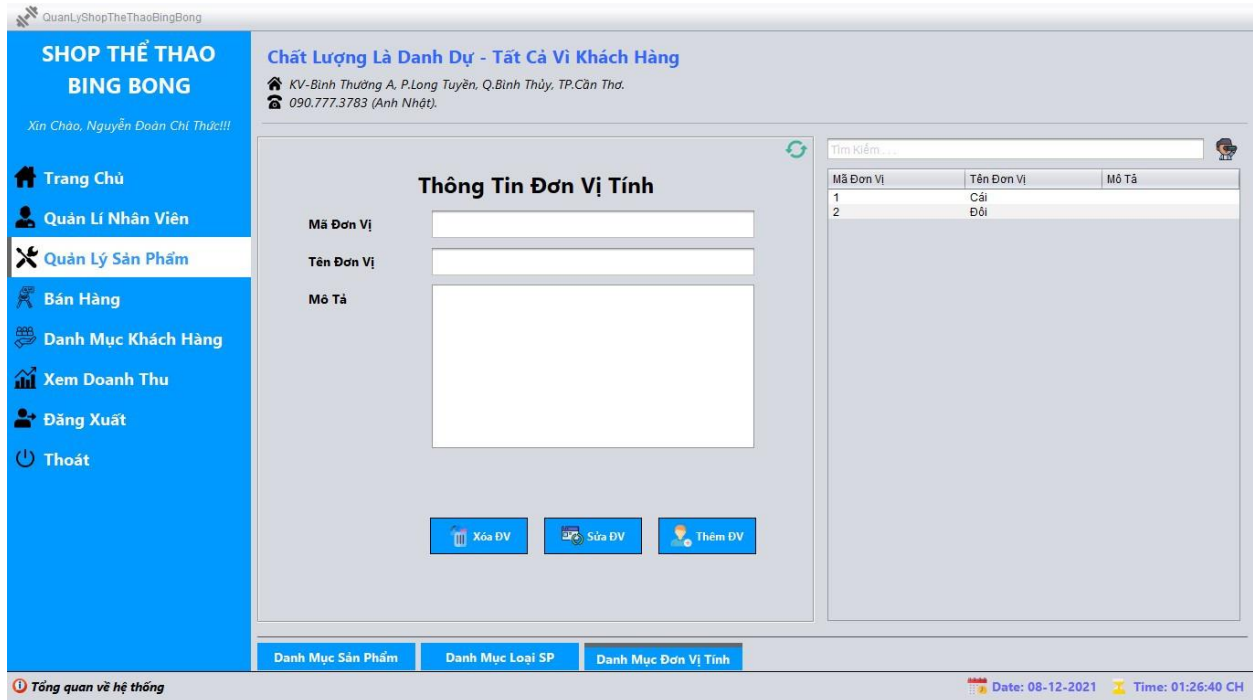
Mô Tả: Giao diện tạo tài khoản dùng để tạo một tài khoản mới cho một nhân viên.

2.3.2.5 Thiết kế Form Sản Phẩm

Mã Sản Phẩm	Tên Sản Phẩm	Số Lượng	Trạng Thái
1	Áo Thun Thái	85	Đang Kinh Doanh
2	Giày Đá Bóng	100	Ngừng Kinh Doa...
3	Quần Dài Thun	931	Đang Kinh Doanh

Mô Tả: Giao diện sản phẩm bao gồm các chức năng chọn sản phẩm, xóa sản phẩm, sửa sản phẩm, thêm sản phẩm, tìm kiếm sản phẩm.

2.3.2.6 Thiết kế Form Đơn Vị Tính



SHOP THỂ THAO BING BONG
Xin Chào, Nguyễn Đoàn Chí Thúc!!!

Chất Lượng Là Danh Dự - Tất Cả Vì Khách Hàng
KV-Bình Thường A, P.Long Tuyền, Q.Bình Thủy, TP.Cần Thơ.
090.777.3783 (Anh Nhật).

Thông Tin Đơn Vị Tính

Mã Đơn Vị:

Tên Đơn Vị:

Mô Tả:

Xóa ĐV Sửa ĐV Thêm ĐV

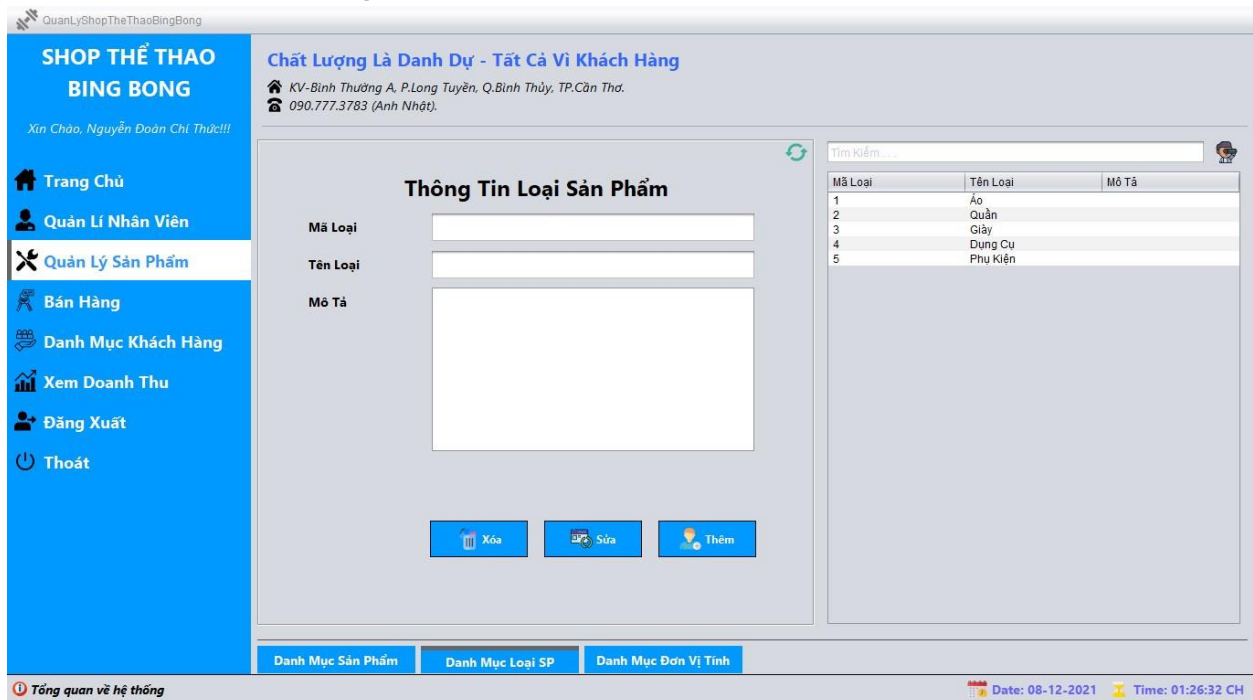
Mã Đơn Vị	Tên Đơn Vị	Mô Tả
1	Cái	
2	Đôi	

Danh Mục Sản Phẩm Danh Mục Loại SP Danh Mục Đơn Vị Tính

Tổng quan về hệ thống Date: 08-12-2021 Time: 01:26:40 CH

Mô Tả: Giao diện đơn vị tính bao gồm các chức năng chọn đơn vị tính, xóa đơn vị tính, sửa đơn vị tính, thêm đơn vị tính, tìm kiếm đơn vị tính.

2.3.2.7 Thiết kế Form Loại Sản Phẩm



SHOP THỂ THAO BING BONG
Xin Chào, Nguyễn Đoàn Chí Thúc!!!

Chất Lượng Là Danh Dự - Tất Cả Vì Khách Hàng
KV-Bình Thường A, P.Long Tuyền, Q.Bình Thủy, TP.Cần Thơ.
090.777.3783 (Anh Nhật).

Thông Tin Loại Sản Phẩm

Mã Loại:

Tên Loại:

Mô Tả:

Xóa Sửa Thêm

Mã Loại	Tên Loại	Mô Tả
1	Áo	
2	Quần	
3	Giày	
4	Dụng Cụ	
5	Phụ Kiện	

Danh Mục Sản Phẩm Danh Mục Loại SP Danh Mục Đơn Vị Tính

Tổng quan về hệ thống Date: 08-12-2021 Time: 01:26:32 CH

Mô Tả: Giao diện loại sản phẩm bao gồm các chức năng chọn loại sản phẩm, xóa loại sản phẩm, sửa loại sản phẩm, thêm loại sản phẩm, tìm kiếm loại sản phẩm.

2.3.2.8 Thiết kế Form Lập Hóa Đơn

QuanLyShopTheThaoBingBong

SHOP THỂ THAO BING BONG
Xin Chào, Nguyễn Đoàn Chí Thúc!!!

Chất Lượng Là Danh Dự - Tất Cả Vì Khách Hàng
KV-Bình Thường A, P.Long Tuyền, Q.Bình Thủy, TP.Cần Thơ.
090.777.3783 (Anh Nhật).

Trang Chủ
Quản Lý Nhân Viên
Quản Lý Sản Phẩm
Bán Hàng
Danh Mục Khách Hàng
Xem Doanh Thu
Đăng Xuất
Thoát

Lập Hóa Đơn
Hóa Đơn Đã Thanh Toán

Tìm khách hàng ...

Tên KH	Tên SP	SL
Khách Hàng Mua Lễ	Áo Thun Thái	85
Huỳnh Văn Đạt	Quần Dài Thun	931
Lê Văn Thanh		

Tìm sản phẩm ...

Tên SP	SL
Áo Thun Thái	85
Quần Dài Thun	931

Nhân Viên: Nguyễn Đoàn Chí Thúc Mã NV: 3
Tên KH: Mã KH:
SDT KH: Ngày Lập HD: 08-12-2021
Phần Trăm Giảm Giá: 0 Số Tiền Giảm: 0 đ
Số Lượng: **Ghi**

Mã Sản Phẩm	Tên SP	SL	Đơn Giá	Thành Tiền
-------------	--------	----	---------	------------

Tiền Cần Thanh Toán: 0 đ
IN HÓA ĐƠN THANH TOÁN

Tổng quan về hệ thống Date: 08-12-2021 Time: 01:26:54 CH

Mô Tả: Giao diện hóa đơn bao gồm các chức năng chọn khách hàng để thanh toán, chọn sản phẩm, lập hóa đơn, thanh toán hóa đơn và in hóa đơn, tìm kiếm khách hàng, tìm kiếm sản phẩm.

2.3.2.9 Thiết kế Form Xem Hóa Đơn

QuanLyShopTheThaoBingBong

SHOP THỂ THAO BING BONG
Xin Chào, Nguyễn Đoàn Chí Thúc!!!

Chất Lượng Là Danh Dự - Tất Cả Vì Khách Hàng
KV-Bình Thường A, P.Long Tuyền, Q.Bình Thủy, TP.Cần Thơ.
090.777.3783 (Anh Nhật).

Trang Chủ
Quản Lý Nhân Viên
Quản Lý Sản Phẩm
Bán Hàng
Danh Mục Khách Hàng
Xem Doanh Thu
Đăng Xuất
Thoát

Lập Hóa Đơn
Hóa Đơn Đã Thanh Toán

Tìm hóa đơn ...

Mã HD	Tên KH	Ngày Lập HD
1	Khách Hàng Mua Lễ	12-06-2021
2	Mai Văn Tài Anh	12-07-2021
3	Khách Hàng Mua Lễ	12-08-2021

Mã HD:
Mã Thu Ngân:
Mã KH:
SDT KH:
Phần Trăm Giảm Giá: Số Tiền Giảm: 0 đ

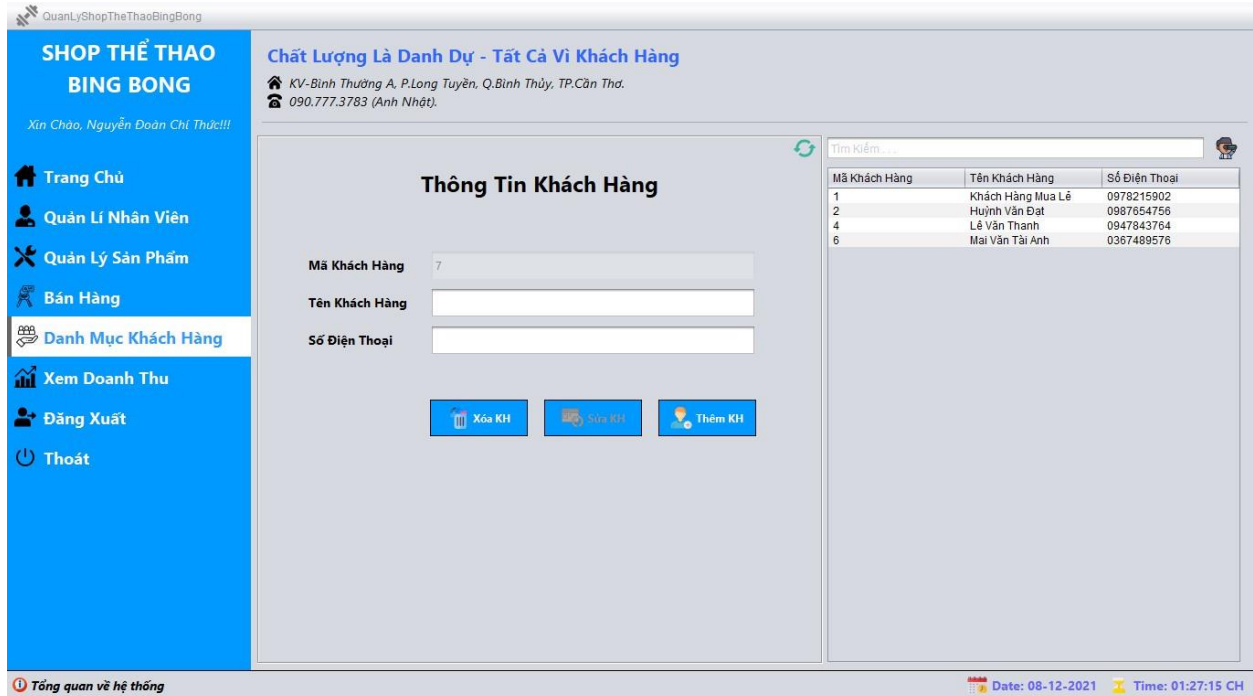
Mã SP	Tên SP	SL	Đơn Giá	Thành Tiền
-------	--------	----	---------	------------

Tiền Cần Thanh Toán: 0 đ
XÓA HÓA ĐƠN IN HÓA ĐƠN

Tổng quan về hệ thống Date: 08-12-2021 Time: 01:27:05 CH

Mô Tả: Giao diện Hóa đơn đã thanh toán dùng để xem lại hóa đơn đã thanh toán bao gồm chức năng chọn hóa đơn đã thanh toán, tìm hóa đơn, in hóa đơn, xóa hóa đơn (Chỉ admin và quản lý mới sử dụng được chức năng này).

2.3.2.10 Thiết kế Form Khách Hàng



SHOP THỂ THAO BING BONG
Xin Chào, Nguyễn Đoàn Chí Thúc!!!

Chất Lượng Là Danh Dự - Tất Cả Vì Khách Hàng
KV-Bình Thường A, P.Long Tuyền, Q.Bình Thủy, TP.Cần Thơ.
090.777.3783 (Anh Nhất).

Thông Tin Khách Hàng

Mã Khách Hàng:

Tên Khách Hàng:

Số Điện Thoại:

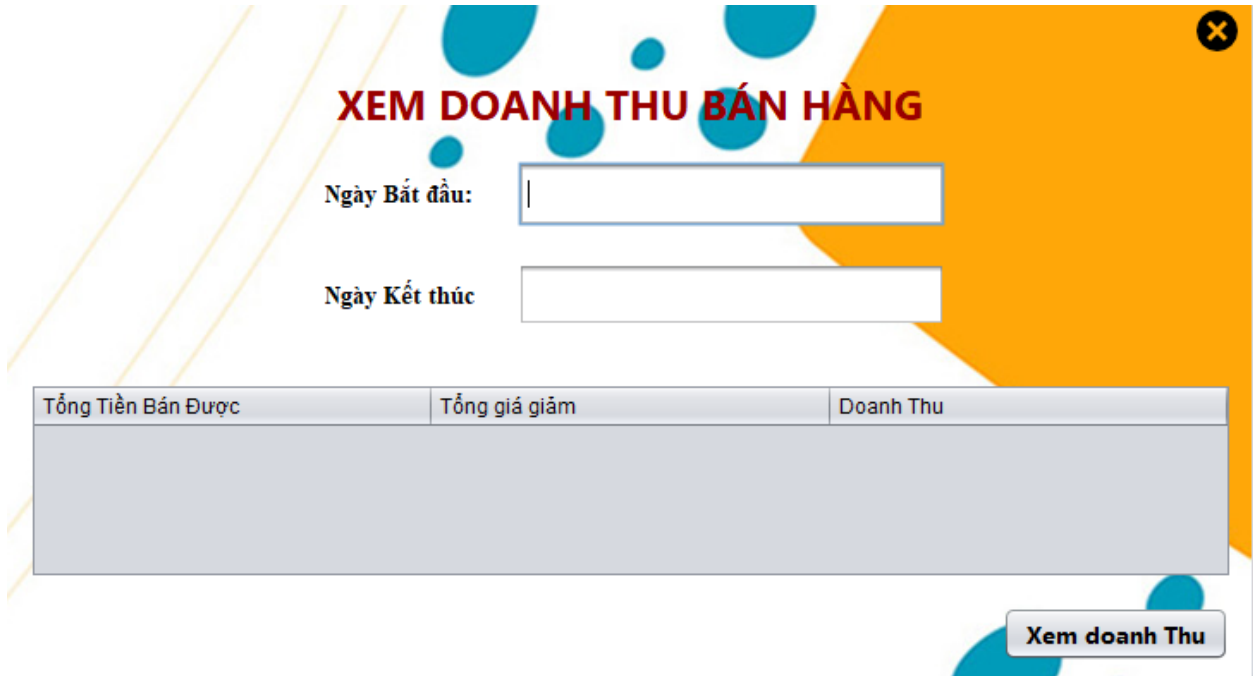
Mã Khách Hàng	Tên Khách Hàng	Số Điện Thoại
1	Khách Hàng Mua Lễ	0978215902
2	Huỳnh Văn Đạt	0987654756
4	Lê Văn Thanh	0947843764
6	Mai Văn Tài Anh	0367489576

Tổng quan về hệ thống

Date: 08-12-2021 Time: 01:27:15 CH

Mô Tả: Giao diện Khách Hàng dùng để xem thông tin khách hàng đã lưu, thêm khách hàng mới, sửa thông tin khách hàng, xóa thông tin khách hàng, tìm kiếm khách hàng.

2.3.2.11 Thiết kế Form Doanh Thu



XEM DOANH THU BÁN HÀNG

Ngày Bắt đầu:

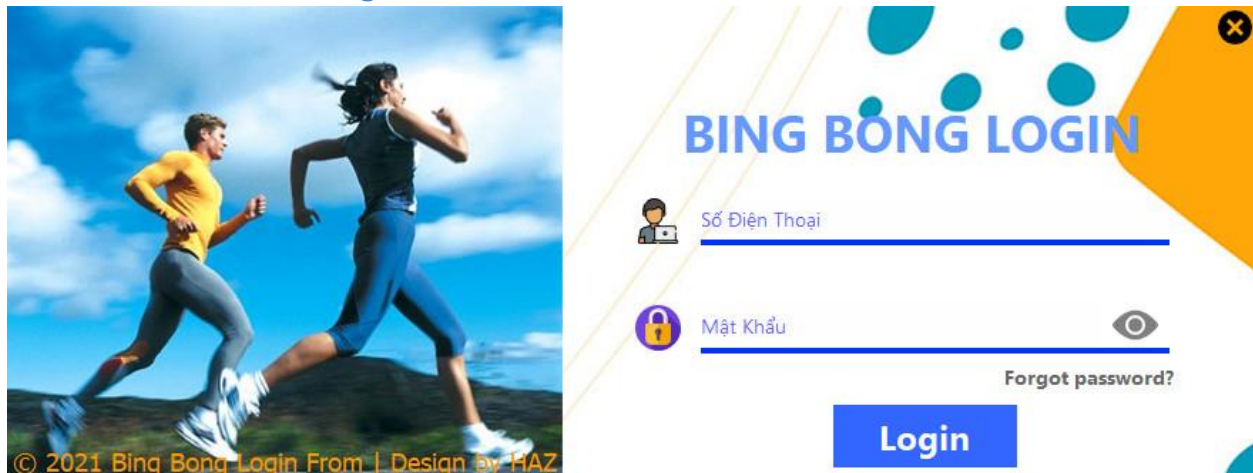
Ngày Kết thúc:

Tổng Tiền Bán Được	Tổng giá giảm	Doanh Thu

Xem doanh Thu


Mô Tả: Giao diện Xem Doanh Thu dùng để xem doanh thu nhập vào ngày bắt đầu và ngày kết thúc để xem được doanh thu trong khoảng ngày.

2.3.2.12 Thiết kế Form Login



BING BONG LOGIN

Số Điện Thoại

Mật Khẩu 

[Forgot password?](#)

Login

© 2021 Bing Bong Login Form | Design by HAZ

Mô Tả: Giao diện đăng nhập hệ thống dùng để đăng nhập tài khoản để sử dụng phần mềm.

2.3.2.13 Thiết kế Form ForgotPasswordDialog

Mô Tả: Giao diện nhập Gmail nhận thông tin mã OTP hệ thống gửi về Gmail đã nhận để xác nhận đổi mật khẩu.

2.3.2.14 Thiết kế Form OTPDialog

Mô Tả: Giao diện nhập mã OTP dùng để nhập mã xác nhận đã nhận từ Gmail trước đó.

2.3.2.15 Thiết kế Form ChangePassDialog

Mô Tả: Giao diện đổi mật dùng để đổi mật khẩu mới.

3 THỰC HIỆN VIẾT MÃ

3.1 VIẾT MÃ TẠO CSDL

3.1.1 Tạo CSDL

3.1.1.1 Lệnh tạo bảng Nhân Viên

Cấu Trúc Bảng

TÊN CỘT	Kiểu dữ liệu	Ràng buộc	Mô tả
MaNV	NVARCHAR(20)	PK, NOT NULL	Mã nhân viên
TenNV	nvarchar(100)	Not Null	Tên nhân viên
SoDienThoai	varchar(10)	Unique, Not Null	Số điện thoại nhân viên
DiaChi	nvarchar(500)	Not Null	Địa chỉ thường trú của nhân viên
CMND	nvarchar(12)	Unique, Not Null	Chứng minh nhân dân
GioiTinh	bit	Not Null	Giới tính (Nam/Nữ)
MaChucVu	nvarchar(20)	Foreign Key, Not Null	Mã chức vụ
Gmail	nvarchar(100)	Unique, Not Null	Địa chỉ Email
PathAnh	nvarchar(max)	Not Null	Địa chỉ file ảnh

Mã Lệnh Tạo Bảng

```
CREATE TABLE [dbo].[NhanVien](
    [MaNV] [nvarchar](20) NOT NULL,
    [TenNV] [nvarchar](500) NOT NULL,
    [SoDienThoai] [varchar](10) NOT NULL,
    [DiaChi] [nvarchar](500) NOT NULL,
    [CMND] [nvarchar](12) NOT NULL,
    [GioiTinh] [bit] NOT NULL,
    [MaChucVu] [nvarchar](20) NOT NULL,
    [Gmail] [nvarchar](100) NOT NULL,
    [PathAnh] [nvarchar](max) NULL,
    PRIMARY KEY CLUSTERED
    (
        [MaNV] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
```

```

IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY],
UNIQUE NONCLUSTERED
(
    [SoDienThoai] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY],
UNIQUE NONCLUSTERED
(
    [Gmail] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY],
UNIQUE NONCLUSTERED
(
    [CMND] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

ALTER TABLE [dbo].[NhanVien] WITH CHECK ADD CONSTRAINT
[FK_ChucVu_NhanVien] FOREIGN KEY([MaChucVu])
REFERENCES [dbo].[ChucVu] ([MaChucVu])
GO

ALTER TABLE [dbo].[NhanVien] CHECK CONSTRAINT [FK_ChucVu_NhanVien]

```

3.1.1.2 Lệnh tạo bảng Sản Phẩm

Cấu Trúc Bảng

TÊN CỘT	Kiểu dữ liệu	Ràng buộc	Mô tả
MaSp	nvarchar(20)	Primary key	Mã sản phẩm
TenSP	nvarchar(255)	Not Null	Tên sản phẩm
DonViTinh	nvarchar(20)	Foreign Key	Đơn vị tính
SoLuong	int	Not Null	Số lượng
Gia	float	Not Null	Giá

TrangThai	nvarchar(100)	Not Null	Trạng thái
Loai	nvarchar(20)	Foreign Key	Loại sản phẩm

Mã Lệnh Tạo Bảng

```

CREATE TABLE [dbo].[SanPham](
    [MaSP] [nvarchar](20) NOT NULL,
    [TenSP] [nvarchar](255) NOT NULL,
    [DonViTinh] [nvarchar](20) NOT NULL,
    [SoLuong] [int] NOT NULL,
    [Gia] [float] NOT NULL,
    [TrangThai] [nvarchar](100) NOT NULL,
    [Loai] [nvarchar](20) NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [MaSP] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[SanPham] WITH CHECK ADD CONSTRAINT
[FK_LoaiSanPham_SanPham] FOREIGN KEY([Loai])
REFERENCES [dbo].[LoaiSP] ([MaLoai])
GO

ALTER TABLE [dbo].[SanPham] CHECK CONSTRAINT
[FK_LoaiSanPham_SanPham]
GO

ALTER TABLE [dbo].[SanPham] WITH CHECK ADD CONSTRAINT
[FK_SanPham_DonVi] FOREIGN KEY([DonViTinh])
REFERENCES [dbo].[DonViTinh] ([MaDonVi])
GO

ALTER TABLE [dbo].[SanPham] CHECK CONSTRAINT [FK_SanPham_DonVi]

```

3.1.1.3 Lệnh tạo bảng Hóa Đơn

Cấu Trúc Bảng

TÊN CỘT	KIỂU DỮ LIỆU	RÀNG BUỘC	MÔ TẢ
MaHD	Int	Primary Key, Not Null	Mã Hóa đơn
MaNV	nvarchar(20)	Foreign Key, Not Null	Mã Nhân viên
MaKH	Int	Foreign Key, Not Null	Mã Khách Hàng
phanTramGiam	float	Not Null	Phần trăm giảm giá
NgayLapHD	date	Not Null	Ngày lập hóa đơn
TrangThai	nvarchar(100)	Not Null	Trạng thái hóa đơn
GiaGiam	float	Not Null	Giá giảm
ThanhTien	float	Not Null	Thành Tiền

Mã Lệnh Tạo Bảng

```

CREATE TABLE [dbo].[HDBan](
    [MaHD] [int] NOT NULL,
    [MaNV] [nvarchar](20) NOT NULL,
    [MaKH] [int] NOT NULL,
    [PhanTramGiam] [float] NOT NULL,
    [NgayLapHD] [date] NOT NULL,
    [TrangThai] [nvarchar](100) NOT NULL,
    [GiaGiam] [float] NOT NULL,
    [ThanhTien] [float] NOT NULL,
    PRIMARY KEY CLUSTERED
    (
        [MaHD] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
    = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[HDBan] WITH CHECK ADD CONSTRAINT
[FK_HoaDon_KhachHang] FOREIGN KEY([MaKH])
REFERENCES [dbo].[KhachHang] ([MaKH])
GO

```

```
ALTER TABLE [dbo].[HDBan] CHECK CONSTRAINT [FK_HoaDon_KhachHang]
GO
```

```
ALTER TABLE [dbo].[HDBan] WITH CHECK ADD CONSTRAINT
[FK_HoaDon_NhanVien] FOREIGN KEY([MaNV])
REFERENCES [dbo].[NhanVien] ([MaNV])
```

3.1.1.4 Lệnh tạo bảng Hóa Đơn Chi Tiết

Cấu Trúc Bảng

TÊN CỘT	Kiểu dữ liệu	Ràng buộc	Mô tả
MaHDCT	int	Primary key, Not Null	Mã hóa đơn chi tiết
MaHD	int	Primary key, Foreign Key, Not Null	Mã hóa đơn
MaSP	nvarchar(20)	Primary key, Foreign Key, Not Null	Mã sản phẩm
SoLuong	int	Primary key, Not Null	Số Lượng
DonGia	float	Primary key. Not Null	Đơn giá

Mã Lệnh Tạo Bảng

```
CREATE TABLE [dbo].[ChiTietHoaDon](
    [MaHDCT] [int] NOT NULL,
    [MaHD] [int] NOT NULL,
    [MaSP] [nvarchar](20) NOT NULL,
    [SoLuong] [int] NOT NULL,
    [DonGia] [float] NOT NULL,
    PRIMARY KEY CLUSTERED
    (
        [MaHDCT] ASC,
        [MaHD] ASC,
        [MaSP] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
    = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[ChiTietHoaDon] WITH CHECK ADD CONSTRAINT
[FK_HoaDon_chiTietHoaDon] FOREIGN KEY([MaHD])
REFERENCES [dbo].[HDBan] ([MaHD])
GO
```

```
ALTER TABLE [dbo].[ChiTietHoaDon] CHECK CONSTRAINT
[FK_HoaDon_chiTietHoaDon]
GO
```

```
ALTER TABLE [dbo].[ChiTietHoaDon] WITH CHECK ADD CONSTRAINT
[FK_SanPham_HoaDonChiTiet] FOREIGN KEY([MaSP])
REFERENCES [dbo].[SanPham] ([MaSP])
GO
```

```
ALTER TABLE [dbo].[ChiTietHoaDon] CHECK CONSTRAINT
[FK_SanPham_HoaDonChiTiet]
```

3.1.1.5 Lệnh tạo bảng Khách Hàng

Cấu Trúc Bảng

TÊN CỘT	Kiểu dữ liệu	Ràng buộc	Mô tả
MaKH	int	Primary Key, Not Null	Mã Khách hàng
TenKH	nvarchar(255)	Not Null	Tên khách hàng
soDT	varchar(10)	Not Null	Số điện thoại

Mã Lệnh Tạo Bảng

```
CREATE TABLE [dbo].[KhachHang](
    [MaKH] [int] NOT NULL,
    [TenKH] [nvarchar](255) NOT NULL,
    [SoDT] [varchar](10) NOT NULL,
    PRIMARY KEY CLUSTERED
    (
        [MaKH] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
    = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY],
    UNIQUE NONCLUSTERED
    (
        [SoDT] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
```

```
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

3.1.1.6 Lệnh tạo bảng Tài Khoản

Cấu Trúc Bảng

TÊN CỘT	KIỂU DỮ LIỆU	RÀNG BUỘC	MÔ TẢ
SoDienThoai	varchar(10)	Primary key, Not Null	Số điện thoại
Pass	nvarchar(255)	Not Null	Mật khẩu của nhân viên

Mã Lệnh Tạo Bảng

```
CREATE TABLE [dbo].[TaiKhoan](
    [SoDienThoai] [varchar](10) NOT NULL,
    [Pass] [nvarchar](255) NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [SoDienThoai] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[TaiKhoan] WITH CHECK ADD CONSTRAINT
[FK_NhanVien_TaiKhoan] FOREIGN KEY([SoDienThoai])
REFERENCES [dbo].[NhanVien] ([SoDienThoai])
GO

ALTER TABLE [dbo].[TaiKhoan] CHECK CONSTRAINT [FK_NhanVien_TaiKhoan]
```

3.1.1.7 Lệnh tạo bảng Chức Vụ

Cấu Trúc Bảng

TÊN CỘT	KIỂU DỮ LIỆU	RÀNG BUỘC	MÔ TẢ
MaChucVu	nvarchar(20)	Primary key, Not Null	Mã chức vụ
TenChucVu	nvarchar(255)	Not Null	Tên chức vụ
MoTa	nvarchar(500)	Not Null	Mô tả

Mã Lệnh Tạo Bảng

```
CREATE TABLE [dbo].[ChucVu](
    [MaChucVu] [nvarchar](20) NOT NULL,
    [TenChucVu] [nvarchar](255) NOT NULL,
    [MoTa] [nvarchar](500) NULL,
    PRIMARY KEY CLUSTERED
(
    [MaChucVu] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

3.1.1.8 Lệnh tạo bảng Loại Sản Phẩm

Cấu Trúc Bảng

TÊN CỘT	KIỂU DỮ LIỆU	RÀNG BUỘC	MÔ TẢ
MaLoai	nvarchar(20)	Primary key, Not Null	Mã loại sản phẩm
TenLoai	nvarchar(255)	Not Null	Tên Loại sản phẩm
MoTa	nvarchar(500)	Not Null	Mô tả

Mã Lệnh Tạo Bảng

```
CREATE TABLE [dbo].[LoaiSP](
    [MaLoai] [nvarchar](20) NOT NULL,
    [TenLoai] [nvarchar](255) NOT NULL,
    [MoTa] [nvarchar](500) NULL,
    PRIMARY KEY CLUSTERED
(
    [MaLoai] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```


3.1.1.9 Lệnh tạo bảng Đơn Vị Tính Sản Phẩm

Cấu Trúc Bảng

TÊN CỘT	Kiểu dữ liệu	Ràng buộc	Mô tả
MaDonVi	nvarchar(20)	Primary key, Not Null	Mã đơn vị
TenDonVi	nvarchar(255)	Not Null	Tên đơn vị
MoTa	nvarchar(500)	Not Null	Mô tả

Mã Lệnh Tạo Bảng

```
CREATE TABLE [dbo].[DonViTinh](
    [MaDonVi] [nvarchar](20) NOT NULL,
    [TenDonVi] [nvarchar](255) NOT NULL,
    [MoTa] [nvarchar](500) NULL,
    PRIMARY KEY CLUSTERED
(
    [MaDonVi] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

3.1.2 SQL truy vấn và thao tác

3.1.2.1 SQL đối với Bảng Nhân Viên

```
INSERT INTO [dbo].[NhanVien]
([MaNV],[TenNV],[SoDienThoai],[DiaChi],[CMND],[GioiTinh],[MaChucVu],[PathAnh])
VALUES (?, ?, ?, ?, ?, ?, ?, ?)
```

```
UPDATE [dbo].[NhanVien] SET [TenNV] = ?, [SoDienThoai] = ?, [DiaChi] = ?,
[CMND] = ?, [GioiTinh] = ?, [MaChucVu] = ?, [Gmail] = ?, [PathAnh] = ? WHERE
[MaNV] = ?
```

```
DELETE FROM NhanVien WHERE MaNV = ?
```

```
SELECT * FROM NhanVien WHERE MaNV = ?
```

3.1.2.2 SQL đối với Bảng Sản Phẩm

```

INSERT INTO [dbo].[SanPham]
    ([MaSP],[TenSP],[DonViTinh],[SoLuong],[Gia],[TrangThai],[Loai]) VALUES
    (?, ?, ?, ?, ?, ?, ?)

UPDATE [dbo].[SanPham]
    SET [TenSP] = ?, [DonViTinh] = , [SoLuong] = ?, [Gia] = ?, [TrangThai] = ?, [Loai]
    = ?
    WHERE [MaSP] = ?

DELETE FROM SanPham WHERE MaSP = ?
SELECT * FROM SanPham WHERE MaSP = ?

```

3.1.2.3 SQL đối với Bảng Hóa Đơn

```

INSERT INTO [dbo].[HDBan]
    ([MaHD], [MaNV], [MaKH], [PhanTramGiam], [NgayLapHD], [TrangThai],
    [GiaGiam], [ThanhTien])
    VALUES ( ?, ?, ?, ?, ?, ?, ?, ?)

UPDATE [dbo].[HDBan]
    SET [MaNV] = ?, [MaKH] = ?, [PhanTramGiam] = ?, [NgayLapHD] = ?,
    [TrangThai] = ?, [GiaGiam] = ?, [ThanhTien] = ? WHERE [MaHD] = ?

DELETE FROM HDBan WHERE MaHD = ?
SELECT * FROM HDBan WHERE MaHD = ?

```

3.1.2.4 SQL đối với Bảng Hóa Đơn Chi Tiết

```

INSERT INTO [dbo].[ChiTietHoaDon]
    ([MaHDCT], [MaHD], [MaSP], [SoLuong], [DonGia])
    VALUES ( ?, ?, ?, ?, ?)

UPDATE [dbo].[ChiTietHoaDon]
    SET [MaHD] = ?, [MaSP] = ?, [SoLuong] = ?, [DonGia] = ? WHERE [MaHDCT] =
    ?

DELETE FROM ChiTietHoaDon WHERE MaHDCT = ?
SELECT * FROM ChiTietHoaDon WHERE MaHDCT = ?

```

3.1.2.5 SQL đối với Bảng Khách Hàng

```
INSERT INTO [dbo].[KhachHang]
    ([MaKH], [TenKH], [SoDT])
VALUES (?, ?, ?)

UPDATE [dbo].[KhachHang]
    SET [TenKH] = , [SoDT] = ? WHERE [MaKH] = ?

DELETE FROM KhachHang WHERE MaKH = ?
SELECT * FROM KhachHang WHERE MaKH = ?
```

3.1.2.6 SQL đối với Bảng Tài Khoản

```
INSERT INTO [dbo].[TaiKhoan]
    ([SoDienThoai], [Pass])
VALUES (?, ?)

UPDATE [dbo].[TaiKhoan]
    SET [SoDienThoai] = ? , [Pass] = ? WHERE [SoDienThoai] = ?

DELETE FROM TaiKhoan WHERE SoDienThoai = ?
SELECT * FROM TaiKhoan WHERE SoDienThoai = ?
```

3.1.2.7 SQL đối với Bảng Chức Vụ

```
INSERT INTO [dbo].[ChucVu]
    ([MaChucVu], [TenChucVu], [MoTa])
VALUES (?, ?, ?)

UPDATE [dbo].[ChucVu]
    SET [TenChucVu] = ? , [MoTa] = ? WHERE [MaChucVu] = ?

DELETE FROM ChucVu WHERE MaChucVu = ?
SELECT * FROM ChucVu WHERE MaChucVu = ?
```

3.1.2.8 SQL đối với Bảng Loại Sản Phẩm

```
INSERT INTO [dbo].[LoaiSP]
    ([MaLoai], [TenLoai], [MoTa])
VALUES (?, ?, ?)

UPDATE [dbo].[LoaiSP]
```

```
SET [TenLoai] = ?, [MoTa] = ? WHERE [MaLoai] = ?
```

```
DELETE FROM LoaiSP WHERE MaLoai = ?
SELECT * FROM LoaiSP WHERE MaLoai = ?
```

3.1.2.9 SQL đối với Bảng Đơn Vị Tính Sản Phẩm

```
INSERT INTO [dbo].[SanPham]
    ([MaSP], [TenSP], [DonViTinh], [SoLuong], [Gia], [TrangThai], [Loai])
VALUES (?, ?, ?, ?, ?, ?, ?)
```

```
UPDATE [dbo].[SanPham]
    SET [TenSP] = ?, [DonViTinh] = ?, [SoLuong] = ?, [Gia] = ?, [TrangThai] = ? ,
    [Loai] = ? WHERE [MaSP] = ?
```

```
DELETE FROM DonViTinh WHERE MaDonVi = ?
SELECT * FROM DonViTinh WHERE MaDonVi = ?
```

3.1.3 Thủ tục lưu tổng hợp thống kê (SP_ThongKe)

MÃ NGUỒN	<pre>Create proc [dbo].[sp_thongke] @ngayBatDau nvarchar(10), @ngayKetThuc nvarchar(10) as begin select sum(cthd.SoLuong * cthd.DonGia) as N'Tổng Tiền Bán Được', sum(hd.GiaGiam) as N'Tổng Giá Giảm', sum(cthd.SoLuong * cthd.DonGia)-sum(hd.GiaGiam) as N'Doanh Thu' from HDBan hd inner join ChiTietHoaDon cthd on hd.MaHD=cthd.MaHD where hd.NgayLapHD between @ngayBatDau and @ngayKetThuc end</pre>
MÔ TẢ	Procedure này được dùng để truy vấn doanh thu của từng hóa đơn theo ngày nhập vào
THAM SỐ	@ngayBatDau, @ngayKetThuc
KẾT QUẢ	[Tổng tiền bán được, Tổng giá giảm, Doanh Thu]

3.2 LẬP TRÌNH JDBC

3.2.1 Lớp hỗ trợ

3.2.1.1 Lớp tiện ích VNCharacterUtils

MÔ TẢ: VNCharacterUtils là lớp hỗ trợ tiện ích dùng để loại bỏ dấu Tiếng Việt .

HƯỚNG DẪN SỬ DỤNG: VNCharacterUtils.**removeAccent**(str)

- Dùng để loại bỏ dấu Tiếng Việt.
- Tham số str bao gồm: Chuỗi cần chuyển đổi.

3.2.1.2 Lớp tiện ích MailSender

MÔ TẢ: MailSender là lớp hỗ trợ tiện ích dùng để gửi và thông báo trạng thái của gmail.

HƯỚNG DẪN SỬ DỤNG: MailSender.**queue**(message)

- Dùng để gửi nội dung gmail đến người nhận.
- Tham số message bao gồm: Gmail người gửi, gmail người nhận, Tiêu đề gmail và nội dung của gmail.

3.2.1.3 Lớp tiện ích MsgBox

MÔ TẢ: MsgBox là lớp tiện ích hỗ trợ xử lý dữ liệu thời gian. Lớp này gồm 3 phương thức được mô tả trong phần hướng dẫn sử dụng sau đây.

HƯỚNG DẪN SỬ DỤNG:

- XJdbc.**alert**(Component parent,String message):
 - Dùng để gọi thông báo JOptionPane.showMessageDialog
 - Tham số parent :
 - Tham số message : nhập chuỗi muốn hiện thông báo
 - Kết quả: Hiện thị được thông báo
- XJdbc.**confirm**(Component parent,String message): boolean
 - Dùng để gọi thông báo JOptionPane.showConfirmDialog
 - Tham số parent :
 - Tham số message : nhập chuỗi muốn hiện thông báo
 - Kết quả: Hiện thị được thông báo
- XJdbc.**prompt**(Component parent,String message): String
 - Dùng để gọi thông báo JOptionPane.showInputDialog
 - Tham số parent :
 - Tham số message : nhập chuỗi muốn hiện thông báo
 - Kết quả: Hiện thị được thông báo

3.2.1.4 Lớp tiện ích XImage

Mô TẢ: XImage là lớp tiện ích hỗ trợ xử lý hình ảnh. Lớp này gồm 3 phương thức được mô tả trong phần hướng dẫn sử dụng sau đây.

HƯỚNG DẪN SỬ DỤNG:

- **XJdbc.getAppIcon(): Image**
 - Đưa hình ảnh icon được lưu trong thư mục lên title cửa sổ
 - Kết quả: số cửa sổ có gắn phương thức
- **XJdbc.save(File src):**
 - Thao tác: chọn một file từ bên ngoài
 - Tham số src: lấy đường dẫn của file đã chọn
 - Tạo thư mục logos nếu chưa tồn tại
 - Copy file vào thư mục logos
 - Kết quả: lấy được đường dẫn của file đã chọn
- **XJdbc.read(String fileName): ImageIcon**
 - Thao tác : chọn một file ảnh
 - Tham số fileName : lấy tên file hình ảnh đã chọn
 - Hiển hình ảnh lên label
 - Kết quả: hiển thị được hình ảnh trên label

3.2.1.5 Lớp tiện ích XJdbc

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

HƯỚNG DẪN SỬ DỤNG:

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

3.2.2 Model class - Các lớp mô tả dữ liệu

3.2.2.1 Entity Nhân Viên

NhanVien
-maNV: String -tenNV: String -soDT: String -diaChi: String -CMND: String -gioiTinh: boolean -chucVu: String -gmail: String -pathAnh: String
+getters() +setters()

NhanVien là lớp thực thể mô tả dữ liệu nhân viên.

3.2.2.2 Entity Chức Vụ

ChucVu
-maChucVu: String -tenChucVu: String -moTa: String
+getters() +setters()

ChucVu là lớp thực thể mô tả dữ liệu chức vụ nhân viên.

3.2.2.3 Entity Tài Khoản

TaiKhoan
-SDT: String -matKhau: String
+getters() +setters()

TaiKhoan là lớp thực thể mô tả dữ liệu tài khoản đăng nhập.

3.2.2.4 Entity Sản Phẩm

SanPham
-maSP: String -tenSP: String -donViTinh: String -soLuong: int -gia: Double -trangThai: String -loaiSP: String
+getters() +setters()

SanPham là lớp thực thể mô tả dữ liệu sản phẩm.

3.2.2.5 Entity Đơn Vị Tính

DonViTinh
-maDonVi: String -tenDonVi: String -moTa: String
+getters() +setters()

DonViTinh là lớp thực thể mô tả dữ liệu đơn vị của sản phẩm.

3.2.2.6 Entity Loại Sản Phẩm

LoaiSP
-maLSP: String -tenLSP: String -moTa: String
+getters() +setters()

LoaiSP là lớp thực thể mô tả dữ liệu loại sản phẩm.

3.2.2.7 Entity Hóa Đơn

HDBan
-maHD: Int -maNV: String -maKH: Int -phanTramGiam: Double -ngayLapHD: date -trangThai: String -giaGiam: Double -thanhTien: Double
+getters() +setters()

HDBan là lớp thực thể mô tả dữ liệu của hóa đơn.

3.2.2.8 Entity Hóa Đơn Chi Tiết

HoaDonChiTiet
-maHDCT: int -maHD: int -maSP: String -soLuong: Int -donGia: Double
+getters() +setters()

HoaDonChiTiet là lớp thực thể mô tả dữ liệu hóa đơn chi tiết.

3.2.2.9 Entity Khách Hàng

KhachHang
-maKH: int -tenKH: String -SoDT: String
+getters() +setters()

KhachHang là lớp thực thể mô tả dữ liệu khách hàng.

3.2.3 DAO Class - Các lớp truy xuất dữ liệu

3.2.3.1 Lớp SoftwareDAO

SoftwareDAO
+insert(Entity) +update(Entity) +delete(Key) +selectById(Key): Entity +selectAll(): List<Entity> +selectBySql(String, Object...): List<Entity>

SoftwareDAO là lớp trừu tượng, tổng quát (Generic) đặc tả những hành động truy xuất CSDL cơ bản nhất mà một lớp DAO cụ thể phải có. Các phương thức thực hiện truy xuất CSDL được mô tả như sau:

TT	PHƯƠNG THỨC	MÔ TẢ HOẠT ĐỘNG
1	insert(Entity)	Thêm mới một bản ghi với dữ liệu là entity E
2	update(Entity)	Cập nhật một bản ghi với dữ liệu là entity E
3	delete(Key)	Xóa một bản ghi với mã K
4	selectById(Key): Entity	Truy vấn một bản ghi với mã là K
5	selectAll(): List<Entity>	Truy vấn tất cả các bản ghi
6	selectBySql(String, Object...): List<Entity>	Truy vấn các bản ghi tùy vào sql và args

Trên đây chỉ là các hoạt động truy xuất CSDL cơ bản. Trong một lớp DAO cụ thể, các DAO con sẽ viết mã cho các phương thức này với Entity là lớp Entity và Key là kiểu khóa chính cụ thể đồng thời có thể bổ sung thêm các hoạt động truy xuất đặc thù riêng của từng DAO cần thiết.

3.2.3.2 Lớp Nhân Viên DAO

NhanVienDAO
+INSERT_SQL: String +UPDATE_SQL: String +DELETE_SQL: String +SELECT_ALL_SQL: String +SELECT_BY_ID_SQL: String +SELECT_BY_SDT_SQL: String
+insert(NhanVien) +update(NhanVien) +delete(String) +selectById(String): NhanVien +selectAll(): List<NhanVien> +selectBySql(String, Object...): List<NhanVien> +selectByKeyWord(String): List<NhanVien> +SelectByIDS(String): List<NhanVien> +SelectBySDT(String): List<NhanVien>

NhanVienDAO là lớp **DAO** cụ thể hóa các hoạt động truy xuất CSDL với bảng **NhanVien**

- ★ **E** được cụ thể hóa là **NhanVien**
- ★ **K** được cụ thể hóa là **String**
- ★ **SQL** là các câu lệnh thao tác dữ liệu trên bảng **NhanVien**

3.2.3.3 Lớp Chức Vụ DAO

ChucVuDAO
+INSERT_SQL: String +UPDATE_SQL: String +DELETE_SQL: String +SELECT_ALL_SQL: String +SELECT_BY_ID_SQL: String
+insert(ChucVu) +update(ChucVu) +delete(String) +selectById(String): ChucVu +selectAll(): List<ChucVu> +selectBySql(String, Object...): List<ChucVu>

ChucVuDAO là lớp **DAO** cụ thể hóa các hoạt động truy xuất CSDL với bảng **ChucVu**

- ★ **E** được cụ thể hóa là **ChucVu**
- ★ **K** được cụ thể hóa là **String**
- ★ **SQL** là các câu lệnh thao tác dữ liệu trên bảng **ChucVu**

3.2.3.4 Lớp Tài Khoản DAO

TaiKhoanDAO
+INSERT_SQL: String +UPDATE_SQL: String +DELETE_SQL: String +SELECT_ALL_SQL: String +SELECT_BY_ID_SQL: String
+insert(TaiKhoan) +update(TaiKhoan) +delete(String) +selectById(String): TaiKhoan +selectAll(): List<ChucVu> +selectBySql(String, Object...): List<ChucVu> +updatePass(String, String)

TaiKhoanDAO là lớp **DAO** cụ thể hóa các hoạt động truy xuất CSDL với bảng **TaiKhoan**

- ★ **E** được cụ thể hóa là **TaiKhoan**
- ★ **K** được cụ thể hóa là **String**
- ★ **SQL** là các câu lệnh thao tác dữ liệu trên bảng **TaiKhoan**

3.2.3.5 Lớp Sản Phẩm DAO

SanPhamDAO
+INSERT_SQL: String +UPDATE_SQL: String +DELETE_SQL: String +SELECT_ALL_SQL: String +SELECT_BY_ID_SQL: String
+insert(SanPham) +update(SanPham) +delete(String) +selectById(String): SanPham +selectAll(): List<SanPham> +selectBySql(String, Object...): List<SanPham> +selectByKeyWord(String): List<SanPham> +SelectByIds(String): List<SanPham> +SelectSanPhamDangKinhDoanh(): List<SanPham> +selectBySanPhamInHoaDon(String): List<SanPham>

SanPhamDAO là lớp **DAO** cụ thể hóa các hoạt động truy xuất CSDL với bảng **SanPham**

- ★ **E** được cụ thể hóa là **SanPham**
- ★ **K** được cụ thể hóa là **String**
- ★ **SQL** là các câu lệnh thao tác dữ liệu trên bảng **SanPham**

3.2.3.6 Lớp Đơn Vị Tính DAO

DonViTinhDAO
+INSERT_SQL: String +UPDATE_SQL: String +DELETE_SQL: String +SELECT_ALL_SQL: String +SELECT_BY_ID_SQL: String
+insert(DonViTinh) +update(DonViTinh) +delete(String) +selectById(String): DonViTinh +selectAll(): List<DonViTinh> +selectBySql(String, Object...): List<DonViTinh> +selectByKeyWord(String): List<DonViTinh> +selectByDonViInSanPham(String): List<SanPham>

DonViTinhDAO là lớp **DAO** cụ thể hóa các hoạt động truy xuất CSDL với bảng **DonViTinh**

- ★ **E** được cụ thể hóa là **DonViTinh**
- ★ **K** được cụ thể hóa là **String**
- ★ **SQL** là các câu lệnh thao tác dữ liệu trên bảng **DonViTinh**

3.2.3.7 Lớp Loại Sản Phẩm DAO

LoaiSanPhamDAO
+INSERT_SQL: String +UPDATE_SQL: String +DELETE_SQL: String +SELECT_ALL_SQL: String +SELECT_BY_ID_SQL: String
+insert(LoaiSanPham) +update(LoaiSanPham) +delete(String) +selectById(String): LoaiSanPham +selectAll(): List<LoaiSanPham> +selectBySql(String, Object...): List<LoaiSanPham> +selectByKeyWord(String): List<LoaiSanPham> +selectByLoaiInSanPham(String): List<LoaiSanPham>

LoaiSanPhamDAO là lớp **DAO** cụ thể hóa các hoạt động truy xuất CSDL với bảng **LoaiSP**

- ★ **E** được cụ thể hóa là **LoaiSanPham**
- ★ **K** được cụ thể hóa là **String**
- ★ **SQL** là các câu lệnh thao tác dữ liệu trên bảng **LoaiSP**

3.2.3.8 Lớp Hóa Đơn DAO

HoaDonDAO
+INSERT_SQL: String +UPDATE_SQL: String +DELETE_SQL: String +SELECT_ALL_SQL: String +SELECT_BY_ID_SQL: String +SELECT_BY_KEYWORD: String
+insert(HoaDon) +update(HoaDon) +delete(Integer) +selectById(Integer): HoaDon +selectAll(): List<HoaDon> +selectBySql(String, Object...): List<HoaDon > +selectByKeyWord(String): List<HoaDon > +selectByIDS(Integer): List<HoaDon>

HoaDonDAO là lớp **DAO** cụ thể hóa các hoạt động truy xuất CSDL với bảng **HDBan**

- ★ **E** được cụ thể hóa là **HoaDon**
- ★ **K** được cụ thể hóa là **Integer**
- ★ **SQL** là các câu lệnh thao tác dữ liệu trên bảng **HDBan**

3.2.3.9 Lớp Hóa Đơn Chi Tiết DAO

HoaDonChiTietDAO
+INSERT_SQL: String +UPDATE_SQL: String +DELETE_SQL: String +SELECT_ALL_SQL: String +SELECT_BY_ID_SQL: String +FIND_MAHDCCT_MAX: String
+insert(HoaDonChiTiet) +update(HoaDonChiTiet) +delete(Integer) +selectById(Integer): HoaDonChiTiet +selectAll(): List<HoaDonChiTiet> +selectBySql(String, Object...): List<HoaDonChiTiet> +findIdMax(): List<HoaDonChiTiet> +selectByIDS(Integer): List<HoaDonChiTiet> +selectByMaHD(Integer): List<HoaDonChiTiet>

HoaDonChiTietDAO là lớp **DAO** cụ thể hóa các hoạt động truy xuất CSDL với bảng **ChiTietHoaDon**

- ★ **E** được cụ thể hóa là **HoaDonChiTiet**
- ★ **K** được cụ thể hóa là **Integer**
- ★ **SQL** là các câu lệnh thao tác dữ liệu trên bảng **ChiTietHoaDon**

3.2.3.10 Lớp Khách Hàng DAO

KhachHangDAO
+INSERT_SQL: String +UPDATE_SQL: String +DELETE_SQL: String +SELECT_ALL_SQL: String +SELECT_BY_ID_SQL: String +SELECT_BY_KEYWORD: String
+insert(KhachHang) +update(KhachHang) +delete(Integer) +selectById(Integer): KhachHang +selectAll(): List<KhachHang> +selectBySql(String, Object...): List<KhachHang> +selectByKeyWord(String): List<KhachHang> +selectByIds(Integer): List<KhachHang> +selectByKhachHangInHoaDon(Integer): List<KhachHang>

KhachHangDAO là lớp **DAO** cụ thể hóa các hoạt động truy xuất CSDL với bảng **KhachHang**

- ★ **E** được cụ thể hóa là **KhachHang**
- ★ **K** được cụ thể hóa là **Integer**
- ★ **SQL** là các câu lệnh thao tác dữ liệu trên bảng **KhachHang**

3.2.3.11 Lớp Doanh Thu DAO

DoanhThuDAO
+getListOfArray(String, String[], Object...): List<Object> +getDoanhThu(String, String): List<Object>

DoanhThuDAO là lớp **DAO** cụ thể hóa các hoạt động truy xuất CSDL và **SQL** là các câu lệnh truy vấn thủ tục lưu trong CSDL

3.3 VIẾT MÃ CHO ỨNG DỤNG

3.3.1.1 Xử lý Form Quản Lý Nhân Viên

```

NhanVienDAO dao = new NhanVienDAO();
TaiKhoanDAO daoTK = new TaiKhoanDAO();
JFileChooser filenChooser = new JFileChooser();
List<NhanVien> listNV;
List<TaiKhoan> listTK;

public void fillTableNV() {
    DefaultTableModel modelNV = (DefaultTableModel)
tblNhanVienNV.getModel();
    modelNV.setRowCount(0);
    try {
        listNV = dao.SelectAll();
        for (NhanVien nv : listNV) {
            Object[] rowNV = new Object[]{nv.getMaNV(), nv.getTenNV()};
            modelNV.addRow(rowNV);
        }
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
    updateStatus();
}

public void setForm(NhanVien nv) {
    txtMaNV.setText(nv.getMaNV());
    txtTenNV.setText(nv.getTenNV());
    if (nv.isGioiTinh()) {
        rdoNamNV.setSelected(true);
    } else {
        rdoNuNV.setSelected(true);
    }
    cboChucVuNV.setSelectedIndex(Integer.parseInt(nv.getMaChucVu()) -
(1));
    txtGmailNV.setText(nv.getGmail());
    txtSoDienThoaiNV.setText(nv.getSoDienThoai());
    txtCmndNV.setText(nv.getCmnd());
    txtDiaChi.setText(nv.getDiaChi());
    if (nv.getPath() != null) {
        ImagemIcon hìnhAnh = new ImagemIcon(new
ImagemIcon(nv.getPath()).getImage()
        .getScaledInstance(lblAvatarNV.getWidth(),

```

```

lblAvatarNV.getHeight(), Image.SCALE_SMOOTH));

        lblAvatarNV.setIcon(hinhAnh);

    }
}

void chonAnh() {
    JFileChooser ch = new JFileChooser("src\\com\\software\\images");
    ch.showSaveDialog(null);
    File f = ch.getSelectedFile();
    if (f != null) {
        duongDan = "src\\com\\software\\images\\" + f.getName();
        ImageIcon hinhAnh = new ImageIcon(new
        ImageIcon(duongDan).getImage()
            .getScaledInstance(lblAvatarNV.getWidth(),
        lblAvatarNV.getHeight(), Image.SCALE_SMOOTH));

        lblAvatarNV.setIcon(hinhAnh);
    }
}

public NhanVien getForm() {
    NhanVien nv = new NhanVien();
    nv.setMaNV(txtMaNV.getText());
    nv.setTenNV(txtTenNV.getText());
    nv.setGioiTinh(rdoNamNV.isSelected());
    nv.setSoDienThoai(txtSoDienThoaiNV.getText());
    nv.setGmail(txtGmailNV.getText());
    nv.setCmnd(txtCmndNV.getText());
    nv.setDiaChi(txtDiaChi.getText());
    if (cboChucVuNV.getSelectedIndex() == 0) {
        nv.setMaChucVu("1");
    } else if (cboChucVuNV.getSelectedIndex() == 1) {
        nv.setMaChucVu("2");
    } else {
        nv.setMaChucVu("3");
    }
    nv.setPath(duongDan);
    return nv;
}

public void clearForm() {

```

```

txtMaNV.setText("");
txtTenNV.setText("");
btgGioiTinh.clearSelection();
cboChucVuNV.setSelectedIndex(0);
txtGmailNV.setText("");
txtSoDienThoaiNV.setText("");
txtCmndNV.setText("");
txtDiaChi.setText("");
lblAvatarNV.setIcon(null);
btnThem.setEnabled(true);
try {
    tblNhanVienNV.setSelectionMode(-1);
} catch (Exception e) {
    System.out.println("Không gì đâu");
}
}
static String manv;

public void edit() {
    row = tblNhanVienNV.getSelectedRow();
    manv = (String) tblNhanVienNV.getValueAt(this.row, 0);
    NhanVien nv = dao.SelectByID(manv);
    this.setForm(nv);
    this.updateStatus();
}

public void insert() {
    NhanVien nv = getForm();
    try {
        dao.insert(nv);
        this.fillTableNV();
        this.clearForm();
        //MsgBox.alert(this, "thêm mới thành công!");
        JOptionPane.showMessageDialog(this, "thêm mới thành công!");
    } catch (Exception e) {
        System.out.println(e);
        JOptionPane.showMessageDialog(this, "thêm mới thất bại !");
    }
}

public void update() {
    NhanVien nv = getForm();
    int chon = JOptionPane.showConfirmDialog(this, "Bạn có chắc muốn cập

```



```

nhật nhân viên này không?");
    if (chon == JOptionPane.OK_OPTION) {
        try {
            dao.update(nv);
            this.fillTableNV();
            this.clearForm();
        } catch (Exception e) {
            System.out.println(e);
            MsgBox.alert(this, "Cập nhật thất bại!");
        }
    }
}

public void updateStatus() {
    row = tblNhanVienNV.getSelectedRow();
    if (row < 0) {
        btnTaoTaiKhoan.setEnabled(false);
        btnSua.setEnabled(false);
        btnThem.setEnabled(true);
        btnXoa.setEnabled(false);
    } else {
        btnTaoTaiKhoan.setEnabled(true);
        btnSua.setEnabled(true);
        btnThem.setEnabled(false);
        btnXoa.setEnabled(true);
        txtMaNV.setEditable(false);
    }
}

public void delete() {

    String manv = txtMaNV.getText();
    String soDienThoai = txtSoDienThoaiNV.getText();
    if (cboChucVuNV.getSelectedItem().toString().equals("Admin")) {
        JOptionPane.showMessageDialog(this, "Không thể thao tác với tài khoản của Admin", "Error", 1);
    } else {
        if (MsgBox.confirm(this, "Bạn thực sự muốn xóa nhân viên này?")) {
            try {

                daoTK.delete(soDienThoai);
                dao.delete(manv);
                this.fillTableNV();
            }
        }
    }
}

```

```

        this.clearForm();
    } catch (Exception e) {
        MsgBox.alert(this, "Nhân viên đã có thao tác đến dữ liệu hệ thống,
không thể xóa!");
    }
}

}

}

}

public static final Pattern VALID_EMAIL_ADDRESS_REGEX
    = Pattern.compile("^([A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,6})$",
Pattern.CASE_INSENSITIVE);

public static boolean validate(String emailStr) {
    Matcher matcher =
VALID_EMAIL_ADDRESS_REGEX.matcher(emailStr);
    return matcher.find();
}
boolean isUpdate = false;
public boolean checkForm() {
    NhanVien nv = getForm();
    if (isUpdate == false) {
        if (txtMaNV.getText().equals("")
            || txtTenNV.getText().equals("") ||
txtSoDienThoaiNV.getText().equals("") ||| cboChucVuNV.getSelectedIndex()
== 0
            || txtGmailNV.getText().equals("")
            || txtCmndNV.getText().equals("") || txtDiaChi.getText().equals("")
||| lblAvatarNV.getIcon() == null
        ) {
            JOptionPane.showMessageDialog(this, "Vui lòng nhập đầy đủ thông
tin trước khi nhấn nút thêm!", "Error", 1);
            return false;
        } else if (!(txtSoDienThoaiNV.getText()).matches("0[3789]{1}[\\d]{8}")) {
            JOptionPane.showMessageDialog(this, "Số điện thoại không đúng -
Vui lòng kiểm tra lại", "Error", 1);
            txtSoDienThoaiNV.requestFocus();
            return false;
        }
        else if (!validate(txtGmailNV.getText())) {
            JOptionPane.showMessageDialog(this, "Email không đúng - vui lòng
kiểm tra lại!", "Error", 1);

```

```

        txtGmailNV.requestFocus();
        return false;
    } else if (txtGmailNV.getText().contains(" ")) {
        JOptionPane.showMessageDialog(this, "Email không được chứa
khoản trắng!", "Error", 1);
        txtGmailNV.requestFocus();
        return false;
    } else if (!(txtCmndNV.getText()).matches("3[\\d]{8}")) {
        JOptionPane.showMessageDialog(this, "Số CMND không đúng - Vui
lòng kiểm tra lại", "Error", 1);
        txtCmndNV.requestFocus();
        return false;
    }
} else {
    if (nv.getMaNV().equals("admin")) {
        JOptionPane.showMessageDialog(this, "Không thể thao tác với tài
khoản admin!", "Error", 1);
        return false;
    }
}
for (int i = 0; i < dao.SelectAll().size(); i++) {
    if (isUpdate == true) {
    } else {
        if (txtMaNV.getText().equals(dao.SelectAll().get(i).getMaNV())) {
            JOptionPane.showMessageDialog(this, "Mã nhân viên không
được phép trùng!", "Error", 1);
            return false;
        } else if
(txtSoDienThoaiNV.getText().equals(dao.SelectAll().get(i).getSoDienThoai())) {
            JOptionPane.showMessageDialog(this, "Số điện thoại đã được sử
dụng ! vui lòng chọn một số khác", "Error", 1);
            return false;
        } else if
(txtSoDienThoaiNV.getText().equals(dao.SelectAll().get(i).getGmail())) {
            JOptionPane.showMessageDialog(this, "Email đã được sử dụng !
vui lòng chọn một số khác", "Error", 1);
            return false;
        }
    }
}
}
return true;
}

```

3.3.1.2 Xử lý Form Quản Lý Chức Vụ

```

int row = -1;
ChucVuDAO daoCV = new ChucVuDAO();
List<ChucVu> list;

public void FillCboChucVuNhanVien() {
    DefaultComboBoxModel model = (DefaultComboBoxModel)
cboChucVuNV.getModel();
    model.removeAllElements();
    list = daoCV.SelectAll();

    if (list.isEmpty()) {
    } else {
        for (int i = 0; i < list.size(); i++) {
            cboChucVuNV.addItem(list.get(i).getTenChucVu());
        }
    }
}

public void fillTableChucVu() {
    DefaultTableModel modelNV = (DefaultTableModel)
tblChucVuCV.getModel();
    modelNV.setRowCount(0);
    try {
        list = daoCV.SelectAll();
        for (ChucVu cv : list) {

            Object[] rowCV = new Object[]{cv.getMaChucVu(),
cv.getTenChucVu(), cv.getMoTa()};
            modelNV.addRow(rowCV);

        }
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}

public void setForm(ChucVu cv) {
    txtMaChucVuCV.setText(cv.getMaChucVu());
    txtTenChucVuCV.setText(cv.getTenChucVu());
    txtMoTaCV.setText(cv.getMoTa());
}

```

```

public ChucVu getFormCV() {
    ChucVu cv = new ChucVu();
    cv.setMaChucVu(txtMaChucVuCV.getText());
    cv.setTenChucVu(txtTenChucVuCV.getText());
    cv.setMoTa(txtMoTaCV.getText());
    return cv;
}

public void clearFormCV() {
    txtMaChucVuCV.setText("");
    txtTenChucVuCV.setText("");
    txtMoTaCV.setText("");
    try {
        tblChucVuCV.setSelectionMode(-1);
    } catch (Exception e) {
        System.out.println("Không gì đâu");
    }
}

public void editCV() {
    row = tblChucVuCV.getSelectedRow();
    String macv = (String) tblChucVuCV.getValueAt(row, 0);
    ChucVu nv = daoCV.SelectByID(macv);
    this.setForm(nv);
}

@SuppressWarnings("UseSpecificCatch")
public void insertCV() {
    ChucVu cv = getFormCV();
    try {
        daoCV.insert(cv);
        this.fillTableChucVu();
        this.clearFormCV();
        //MsgBox.alert(this, "thêm mới thành công!");
        JOptionPane.showMessageDialog(this, "thêm mới thành công!");
    } catch (Exception e) {
        System.out.println(e);
        JOptionPane.showMessageDialog(this, "thêm mới thất bại !");
    }
}

public void updateCV() {

```

```

        ChucVu cv = getFormCV();
        int chon = JOptionPane.showConfirmDialog(this, "Bạn có chắc muốn cập
nhật nhân viên này không?");
        if (chon == JOptionPane.OK_OPTION) {
            try {
                daoCV.update(cv);
                this.fillTableChucVu();
                this.clearForm();
            } catch (Exception e) {
                System.out.println(e);
                // MsgBox.alert(this, "Cập nhật thất bại!");
            }
        }
    }

    public void deleteCV() {
        String macv = txtMaChucVuCV.getText();
        for (int i = 0; i < dao.SelectAll().size(); i++) {
            if
(txtMaChucVuCV.getText().equals(dao.SelectAll().get(i).getMaChucVu())) {
                JOptionPane.showMessageDialog(this, "Chức vụ này đang sử dụng
bạn không thể xóa nó ", "Error", 1);
                break;
            } else {
                if (MsgBox.confirm(this, "Bạn thực sự muốn xóa nhân viên này?")) {
                    try {
                        daoCV.delete(macv);
                        this.fillTableChucVu();
                        this.clearFormCV();
                    } catch (Exception e) {
                        MsgBox.alert(this, "Chức vụ đã có thao tác đến dữ liệu hệ
thống, không thể xóa!");
                    }
                }
                break;
            }
        }
    }

    public boolean checkFormCV() {

        ChucVu cv = getFormCV();
    }

```

```

        if (isUpdate == false) {
            if (txtMaChucVuCV.getText().equals("") ||
txtTenChucVuCV.getText().equals("")) {
                JOptionPane.showMessageDialog(this, "Vui lòng nhập đầy đủ thông
tin trước khi nhấn nút thêm!", "Error", 1);
                return false;
            } else if (!(txtMaChucVuCV.getText()).matches("[\\d]*")) {
                JOptionPane.showMessageDialog(this, "Mã bạn nhập sai định dạng -
Vui lòng kiểm tra lại", "Error", 1);
                txtMaChucVuCV.requestFocus();
                return false;
            }
        }
        for (int i = 0; i < dao.SelectAll().size(); i++) {
            if (isUpdate == true) {
            } else {
                if
(txtMaChucVuCV.getText().equals(daoCV.SelectAll().get(i).getMaChucVu())) {
                    JOptionPane.showMessageDialog(this, "Mã không được phép
trùng!", "Error", 1);
                    return false;
                }
            }
        }
        return true;
    }
}

```

3.3.1.3 Xử lý Form Thêm Tài Khoản

```

public boolean checkTK() {
    listTK = daoTK.SelectAll();
    for (int i = 0; i < daoTK.SelectAll().size(); i++) {
        if (isUpdate == false) {
        } else {
            if
(txtGmailNV.getText().equals(daoTK.SelectAll().get(i).getSoDienThoai())) {
                JOptionPane.showMessageDialog(this, "Nhân viên này đã đăng ký
tài khoản trước đó", "Error", 1);
                return false;
            }
        }
    }
}

```

```

    }
    return true;
}

```

3.3.1.4 Xử lý Form Quản Lý Sản Phẩm

```

SanPhamDAO daoSP = new SanPhamDAO();
int indexSP = -1;
List<SanPham> listSP = daoSP.SelectAll();

LoaiSanPhamDAO daoLoaiSP = new LoaiSanPhamDAO();
List<LoaiSanPham> listLoaiSP = daoLoaiSP.SelectAll();

DonViTinhDAO daoDonViTinh = new DonViTinhDAO();
List<DonViTinh> listDonViTinh = daoDonViTinh.SelectAll();

int indexCboLoai = -1;
int indexCboDonVi = -1;

public void FillTableSP(int chon) {
    DefaultTableModel model = (DefaultTableModel)
tblSanPhamSP.getModel();
    model.setRowCount(0);
    if (chon == 1) {
        listSP = daoSP.SelectAll();
    }
    if (chon == 2) {
        listSP = daoSP.selectByKeyWord(txtTimKiemSanPhamSP.getText());
    }
    Object rowData[] = new Object[4];
    for (int i = 0; i < listSP.size(); i++) {
        rowData[0] = listSP.get(i).getMaSanPham();
        rowData[1] = listSP.get(i).getTenSanPham();
        rowData[2] = listSP.get(i).getSoLuong();
        rowData[3] = listSP.get(i).getTrangThai();
        model.addRow(rowData);
    }
    this.UpdateStatusSP();
}

public void TimSP() {
    if (txtTimKiemSanPhamSP.getText().isEmpty() ||

```



```

txtTimKiemSanPhamSP.getText().equals("Tìm Kiếm. . .")) {
    this.FillTableSP(1);
} else {
    this.FillTableSP(2);
}
}

public void FillCboDVT() {
    DefaultComboBoxModel model = (DefaultComboBoxModel)
cboDonViTinhSP.getModel();
    model.removeAllElements();
    List<DonViTinh> list = daoDVT.SelectAll();
    if (list.isEmpty()) {
        model.addElement("--Chọn đơn vị--");
    } else {
        for (int i = 0; i < list.size(); i++) {
            String dvt = list.get(i).getTenDonVi();
            model.addElement(dvt);
        }
    }
}

public void FillCboL() {
    DefaultComboBoxModel model = (DefaultComboBoxModel)
cboLoaiSP.getModel();
    model.removeAllElements();
    List<LoaiSanPham> list = daoLSP.SelectAll();
    if (list.isEmpty()) {
        model.addElement("--Chọn loại--");
    } else {
        for (int i = 0; i < list.size(); i++) {
            String lsp = list.get(i).getTenLoai();
            model.addElement(lsp);
        }
    }
}

public void SetFormSP(SanPham sp) {
    listDonViTinh = daoDonViTinh.SelectAll();
    listLoaiSP = daoLoaiSP.SelectAll();
    indexCboDonVi = 0;
    indexCboLoai = 0;
    txtTenSanPhamSP.setText(sp.getTenSanPham());
}

```

```

txtMaSanPhamSP.setText(sp.getMaSanPham());
txtSoLuongSP.setText(String.valueOf(sp.getSoLuong()));
txtDonGiaSP.setText(String.valueOf(sp.getGia()));
for (int i = 0; i < listDonViTinh.size(); i++) {
    if (listDonViTinh.get(i).getMaDonVi().equals(sp.getDonViTinh())) {
        indexCboDonVi = i;
        break;
    }
}
for (int i = 0; i < listLoaiSP.size(); i++) {
    if (listLoaiSP.get(i).getMaLoai().equals(sp.getLoaiSanPham())) {
        indexCboLoai = i;
        break;
    }
}
cboDonViTinhSP.setSelectedIndex(indexCboDonVi);
cboLoaiSP.setSelectedIndex(indexCboLoai);
if (sp.getTrangThai().equals("Đang Kinh Doanh")) {
    cboTrangThaiSP.setSelectedIndex(0);
} else {
    cboTrangThaiSP.setSelectedIndex(1);
}
}

public SanPham GetFormSP() {
    SanPham sp = new SanPham();
    listDonViTinh = daoDonViTinh.SelectAll();
    listLoaiSP = daoLoaiSP.SelectAll();
    indexCboDonVi = cboDonViTinhSP.getSelectedIndex();
    indexCboLoai = cboLoaiSP.getSelectedIndex();
    sp.setMaSanPham(txtMaSanPhamSP.getText());
    sp.setTenSanPham(txtTenSanPhamSP.getText());
    sp.setSoLuong(Integer.valueOf(txtSoLuongSP.getText()));
    sp.setLoaiSanPham(listLoaiSP.get(indexCboLoai).getMaLoai());
    sp.setDonViTinh(listDonViTinh.get(indexCboDonVi).getMaDonVi());
    sp.setGia(Double.valueOf(txtDonGiaSP.getText()));
    sp.setTrangThai(String.valueOf(cboTrangThaiSP.getSelectedItem()));
    return sp;
}

public void InsertSP() {
    SanPham sp = GetFormSP();
    if (CheckSP(1) == true) {

```

```

        daoSP.insert(sp);
        this.FillTableSP(1);
        this.ResetSP();
        MsgBox.alert(this, "Thêm sản phẩm thành công!");
    }
}

public void UpdateSP() {
    SanPham sp = GetFormSP();
    if (CheckSP(0) == true) {
        daoSP.update(sp);
        this.FillTableSP(1);
        MsgBox.alert(this, "Chỉnh sửa sản phẩm " +
txtMaSanPhamSP.getText() + " thành công!");
    }
}

public void DeleteSP() {
    if (indexSP == -1) {
        MsgBox.alert(this, "Vui lòng chọn sản phẩm!");
    } else {
        String maSanPhamSP =
String.valueOf(tblSanPhamSP.getValueAt(indexSP, 0));
        listSP = daoSP.selectBySanPhamInHoaDon(maSanPhamSP);
        if (listSP.isEmpty()) {
            String maSP = txtMaSanPhamSP.getText();
            boolean n = MsgBox.confirm(this, "Bạn chắc chắn muốn xóa sản
phẩm này không???");
            if (n == true) {
                daoSP.delete(maSP);
                this.FillTableLSP(1);
                this.ResetSP();
                MsgBox.alert(this, "Xóa thành công sản phẩm " + maSP + " !");
            } else {
                MsgBox.alert(this, "Đã hoàn tác lệnh xóa!");
            }
        } else {
            MsgBox.alert(this, "Sản phẩm đang được sử dụng!\nKhông được
xóa!");
        }
    }
}
}

```

```

public void ResetSP() {
    this.indexSP = -1;
    txtTenSanPhamSP.setText("");
    txtMaSanPhamSP.setText("");
    txtSoLuongSP.setText("");
    txtDonGiaSP.setText("");
    cboDonViTinhSP.setSelectedIndex(0);
    cboLoaiSP.setSelectedIndex(0);
    cboTrangThaiSP.setSelectedIndex(0);
    indexCboDonVi = -1;
    indexCboLoai = -1;
    this.UpdateStatusSP();
    txtMaSanPhamSP.setEnabled(true);
}

public void EditSP() {
    String msSP = (String) tblSanPhamSP.getValueAt(this.indexSP, 0);
    SanPham sp = daoSP.SelectByID(msSP);
    this.SetFormSP(sp);
    txtMaSanPhamSP.setEnabled(false);
}

public void UpdateStatusSP() {
    if (indexSP == -1) {
        lblSuaSP.setEnabled(false);
        lblThemSP.setEnabled(true);
    } else {
        lblSuaSP.setEnabled(true);
        lblThemSP.setEnabled(false);
    }
}

public boolean CheckSP(int mucDich) {
    booleankiemTraSP = true;
    listSP = daoSP.SelectAll();
    String maSP = txtMaSanPhamSP.getText();
    if (txtMaSanPhamSP.getText().trim().isEmpty()) {
        MsgBox.alert(this, "Vui lòng nhập mã sản phẩm!");
        kiemTraSP = false;
    } else if (txtTenSanPhamSP.getText().trim().isEmpty()) {
        MsgBox.alert(this, "Vui lòng nhập tên sản phẩm!");
        kiemTraSP = false;
    } else {

```

```

        for (int i = 0; i < listSP.size(); i++) {
//            1 là insert
            if (mucDich == 1) {
                if (maSP.equals(listSP.get(i).getMaSanPham())) {
                    MsgBox.alert(this, "Mã sản phẩm đã tồn tại!");
                   kiemTraSP = false;
                }
            }
        }
    }
    returnkiemTraSP;
}

```

3.3.1.5 Xử lý Form Quản Lý Đơn Vị Tính

```

LoaiSanPhamDAO daoLSP = new LoaiSanPhamDAO();
int indexLSP = -1;
List<LoaiSanPham> listLSP = daoLSP.SelectAll();

public void FillTableLSP(int chon) {
    DefaultTableModel model = (DefaultTableModel)
tblLoaiSanPham.getModel();
    model.setRowCount(0);
    if (chon == 1) {
        listLSP = daoLSP.SelectAll();
    }
    if (chon == 2) {
        listLSP = daoLSP.selectByKeyWord(txtTimLoaiSP.getText());
    }
    Object rowData[] = new Object[3];
    for (int i = 0; i < listLSP.size(); i++) {
        rowData[0] = listLSP.get(i).getMaLoai();
        rowData[1] = listLSP.get(i).getTenLoai();
        rowData[2] = listLSP.get(i).getMoTa();
        model.addRow(rowData);
    }
}

public void TimLSP() {
    if (txtTimLoaiSP.getText().isEmpty() || txtTimLoaiSP.getText().equals("Tìm
Kiếm. . .")) {
        this.FillTableLSP(1);
    }
}

```

```

    } else {
        this.FillTableLSP(2);
    }
}

public void SetFormLSP(LoaiSanPham lsp) {
    txtMaLoai.setText(lsp.getMaLoai());
    txtTenLoai.setText(lsp.getTenLoai());
    txtMoTaLoai.setText(lsp.getMoTa());
}

public LoaiSanPham GetFormLSP() {
    LoaiSanPham lsp = new LoaiSanPham();
    lsp.setMaLoai(txtMaLoai.getText());
    lsp.setTenLoai(txtTenLoai.getText());
    lsp.setMoTa(txtMoTaLoai.getText());
    return lsp;
}

public void InsertLSP() {
    LoaiSanPham lsp = GetFormLSP();
    if (CheckLSP(1) == true) {
        daoLSP.insert(lsp);
        this.FillTableLSP(1);
        this.ResetLSP();
        MsgBox.alert(this, "Thêm loại sản phẩm thành công!");
    }
}

public void UpdateLSP() {
    LoaiSanPham lsp = GetFormLSP();
    if (CheckLSP(0) == true) {
        daoLSP.update(lsp);
        this.FillTableLSP(1);
        MsgBox.alert(this, "Chỉnh sửa loại sản phẩm " + txtMaLoai.getText() + " thành công!");
    }
}

public void DeleteLSP() {
    if (indexLSP == -1) {
        MsgBox.alert(this, "Vui lòng chọn loại sản phẩm!");
    } else {

```

```

String maLoai = String.valueOf(tblLoaiSanPham.getValueAt(indexLSP,
0));
listLSP = daoLSP.selectByLoaiInSanPham(maLoai);
if (listLSP.isEmpty()) {
    String maLSP = txtMaLoai.getText();
    boolean n = MsgBox.confirm(this, "Bạn chắc chắn muốn xóa mã này
không???");
    if (n == true) {
        daoLSP.delete(maLSP);
        this.FillTableLSP(1);
        this.ResetLSP();
        MsgBox.alert(this, "Xóa thành công mã " + maLSP + " !");
    } else {
        MsgBox.alert(this, "Đã hoàn tác lệnh xóa!");
    }
} else {
    MsgBox.alert(this, "Loại sản phẩm đang được sử dụng!\nKhông
được xóa!");
}
}
}

public void ResetLSP() {
    LoaiSanPham lsp = new LoaiSanPham();
    this.SetFormLSP(lsp);
    this.indexLSP = -1;
    txtMaLoai.setText("");
    txtTenLoai.setText("");
    txtMoTaLoai.setText("");
    txtMaLoai.setEnabled(true);
    this.UpdateStatusLSP();
}

public void EditLSP() {
    String msLSP = (String) tblLoaiSanPham.getValueAt(this.indexLSP, 0);
    LoaiSanPham lsp = daoLSP.SelectByID(msLSP);
    this.SetFormLSP(lsp);
    txtMaLoai.setEnabled(false);
}

public void UpdateStatusLSP() {
    if (indexLSP == -1) {
        lblSuaLoai.setEnabled(false);
    }
}

```

```

        lblThemLoai.setEnabled(true);
    } else {
        lblSuaLoai.setEnabled(true);
        lblThemLoai.setEnabled(false);
    }
}

public boolean CheckLSP(int mucDich) {
    booleankiemTraLSP = true;
    listLSP = daoLSP.SelectAll();
    String maLSP = txtMaLoai.getText();
    if (txtMaLoai.getText().trim().isEmpty()) {
        MsgBox.alert(this, "Vui lòng nhập Loại sản phẩm!");
        kiemTraLSP = false;
    } else if (txtTenLoai.getText().trim().isEmpty()) {
        MsgBox.alert(this, "Vui lòng nhập tên Loại sản phẩm!");
        kiemTraLSP = false;
    } else {
        for (int i = 0; i < listLSP.size(); i++) {
//            1 là insert
            if (mucDich == 1) {
                if (maLSP.equals(listLSP.get(i).getMaLoai())) {
                    MsgBox.alert(this, "Mã Loại sản phẩm đã tồn tại!");
                    kiemTraLSP = false;
                }
            }
        }
    }
    return kiemTraLSP;
}

```

3.3.1.6 Xử lý Form Quản Lý Loại Sản Phẩm

```

DonViTinhDAO daoDVT = new DonViTinhDAO();
int indexDVT = -1;
List<DonViTinh> listDVT = daoDVT.SelectAll();

public void FillTableDVT(int chon) {
    DefaultTableModel model = (DefaultTableModel) tblDonViTinh.getModel();
    model.setRowCount(0);
    if (chon == 1) {
        listDVT = daoDVT.SelectAll();
    }
}

```



```

    }
    if (chon == 2) {
        listDVT = daoDVT.selectByKeyWord(txtTimKiemDV.getText());
    }
    Object rowData[] = new Object[3];
    for (int i = 0; i < listDVT.size(); i++) {
        rowData[0] = listDVT.get(i).getMaDonVi();
        rowData[1] = listDVT.get(i).getTenDonVi();
        rowData[2] = listDVT.get(i).getMoTa();
        model.addRow(rowData);
    }
}

public void TimDVT() {
    if (txtTimKiemDV.getText().isEmpty() ||
txtTimKiemDV.getText().equals("Tìm Kiếm. . .")) {
        this.FillTableDVT(1);
    } else {
        this.FillTableDVT(2);
    }
}

public void SetFormDVT(DonViTinh dvt) {
    txtMaDonViDV.setText(dvt.getMaDonVi());
    txtTenDonViDV.setText(dvt.getTenDonVi());
    txtMoTaDV.setText(dvt.getMoTa());
}

public DonViTinh GetFormDVT() {
    DonViTinh dvt = new DonViTinh();
    dvt.setMaDonVi(txtMaDonViDV.getText());
    dvt.setTenDonVi(txtTenDonViDV.getText());
    dvt.setMoTa(txtMoTaDV.getText());
    return dvt;
}

public void InsertDVT() {
    DonViTinh dvt = GetFormDVT();
    if (CheckDVT(1) == true) {
        daoDVT.insert(dvt);
        this.FillTableDVT(1);
        this.ResetDVT();
        MsgBox.alert(this, "Thêm loại mới thành công!");
    }
}

```

```

    }
}

public void UpdateDVT() {
    DonViTinh dvt = GetFormDVT();
    if (CheckDVT(0) == true) {
        daoDVT.update(dvt);
        this.FillTableDVT(1);
        MsgBox.alert(this, "Chỉnh sửa loại sản phẩm " +
txtMaDonViDV.getText() + " thành công!");
    }
}

public void DeleteDVT() {
    if (indexDVT == -1) {
        MsgBox.alert(this, "Vui lòng chọn đơn vị!");
    } else {
        String maDonVi = String.valueOf(tblDonViTinh.getValueAt(indexDVT,
0));
        listDVT = daoDVT.selectByDonViInSanPham(maDonVi);
        if (listDVT.isEmpty()) {
            String maDVT = txtMaDonViDV.getText();
            boolean n = MsgBox.confirm(this, "Bạn chắc chắn muốn xóa mã này
không???");
            if (n == true) {
                daoDVT.delete(maDVT);
                this.FillTableDVT(1);
                this.ResetDVT();
                MsgBox.alert(this, "Xóa thành công mã " + maDVT + " !");
            } else {
                MsgBox.alert(this, "Đã hoàn tác lệnh xóa!");
            }
        } else {
            MsgBox.alert(this, "Đơn vị đang được sử dụng!\nKhông được xóa!");
        }
    }
}

public void ResetDVT() {
    DonViTinh dvt = new DonViTinh();
    this.SetFormDVT(dvt);
    this.indexDVT = -1;
    txtMaDonViDV.setText("");
}

```

```

        txtTenDonViDV.setText("");
        txtMoTaDV.setText("");
        txtMaDonViDV.setEnabled(true);
        this.UpdateStatusDVT();
    }

    public void EditDVT() {
        String msDVT = (String) tblDonViTinh.getValueAt(this.indexDVT, 0);
        DonViTinh dvt = daoDVT.SelectByID(msDVT);
        this.SetFormDVT(dvt);
        txtMaDonViDV.setEnabled(false);
    }

    public void UpdateStatusDVT() {
        if (indexDVT == -1) {
            lblSuaDV.setEnabled(false);
            lblThemDV.setEnabled(true);
        } else {
            lblSuaDV.setEnabled(true);
            lblThemDV.setEnabled(false);
        }
    }

    public boolean CheckDVT(int mucDich) {
        booleankiemTraDVT = true;
        listDVT = daoDVT.SelectAll();
        String maDVT = txtMaDonViDV.getText();
        if (txtMaDonViDV.getText().trim().isEmpty()) {
            MsgBox.alert(this, "Vui lòng nhập đơn vị tính!");
            kiemTraDVT = false;
        } else if (txtTenDonViDV.getText().trim().isEmpty()) {
            MsgBox.alert(this, "Vui lòng nhập tên đơn vị tính!");
            kiemTraDVT = false;
        } else {
            for (int i = 0; i < listDVT.size(); i++) {
                // 1 là insert
                if (mucDich == 1) {
                    if (maDVT.equals(listDVT.get(i).getMaDonVi())) {
                        MsgBox.alert(this, "Mã đơn vị tính đã tồn tại!");
                        kiemTraDVT = false;
                    }
                }
            }
        }
    }

```

```

    }
    return kiemTraDVT;
}

```

3.3.1.7 Xử lý Form Lập Hóa Đơn

```

KhachHangDAO khachHangHD = new KhachHangDAO();
List<KhachHang> listKHHD;
SanPhamDAO sanPhamHD = new SanPhamDAO();
List<SanPham> listSPHD;
HoaDonDAO hoaDonHD = new HoaDonDAO();
HoaDonChiTietDAO hoaDonChiTietHD = new HoaDonChiTietDAO();
int indexKHHD = -1, indexSPHD = -1, indexSanPhamTrung = -1, indexHD =
-1;
int soLuong = 0, soLuongNew = 0;
int maHoaDonHD = 0, maHoaDonChiTietHD = 0;

public void TimSanPhamHD() {
    if (txtTimSanPhamHD.getText().isEmpty() ||
txtTimSanPhamHD.getText().equals("Tìm sản phẩm. . .")) {
        this.FillTableSPHD(1);
    } else {
        this.FillTableSPHD(2);
    }
}

public void TimKhachHangHD() {
    if (txtTimKhachHangHD.getText().isEmpty() ||
txtTimKhachHangHD.getText().equals("Tìm khách hàng. . .")) {
        this.FillTableKHHD(1);
    } else {
        this.FillTableKHHD(2);
    }
}

public void FillTableKHHD(int kieu) {
    DefaultTableModel model = (DefaultTableModel) tblChonKH.getModel();
    model.setRowCount(0);
    if (kieu == 1) {
        listKHHD = khachHangHD.SelectAll();
    }
    if (kieu == 2) {

```

```

        listKHHD =
khachHangHD.selectByKeyWord(txtTimKhachHangHD.getText());
    }
    Object rowData[] = new Object[1];
    for (int i = 0; i < listKHHD.size(); i++) {
        rowData[0] = listKHHD.get(i).getTenKhachHang();
        model.addRow(rowData);
    }
}

public void FillThongTinThuNgan() {
    lblChao.setText("Xin Chào, " + nhanVienDangNhap.get(0).getTenNV() +
"!!!");
    txtMaNhanVienHD.setText(nhanVienDangNhap.get(0).getMaNV());
    txtTenNhanVienHD.setText(nhanVienDangNhap.get(0).getTenNV());
}

public void FillTableSPHD(int mucDich) {
    DefaultTableModel model = (DefaultTableModel) tblChonSP.getModel();
    if (tblHoaDon.getRowCount() == 0) {
        model.setRowCount(0);
        if (mucDich == 1) {
            listSPHD = sanPhamHD.SelectSanPhamDangKinhDoanh();
        }
        if (mucDich == 2) {
            listSPHD =
sanPhamHD.selectByKeyWord(txtTimSanPhamHD.getText());
        }
        Object rowData[] = new Object[2];
        for (int i = 0; i < listSPHD.size(); i++) {
            rowData[0] = listSPHD.get(i).getTenSanPham();
            rowData[1] = listSPHD.get(i).getSoLuong();
            model.addRow(rowData);
        }
    } else {
        model.setRowCount(0);
        if (mucDich == 1) {
            listSPHD = sanPhamHD.SelectSanPhamDangKinhDoanh();
        }
        if (mucDich == 2) {
            listSPHD =
sanPhamHD.selectByKeyWord(txtTimSanPhamHD.getText());
        }
    }
}

```

```

Object rowData[] = new Object[2];
for (int i = 0; i < listSPHD.size(); i++) {
    rowData[0] = listSPHD.get(i).getTenSanPham();
    rowData[1] = listSPHD.get(i).getSoLuong();
    model.addRow(rowData);
}
for (int i = 0; i < listSPHD.size(); i++) {
    for (int j = 0; j < tblHoaDon.getRowCount(); j++) {
        String maSP = (String) tblHoaDon.getValueAt(j, 0);
        if (listSPHD.get(i).getMaSanPham().equals(maSP)) {
            Integer slSP = listSPHD.get(i).getSoLuong();
            Integer slHD = (Integer) tblHoaDon.getValueAt(j, 2);
            tblChonSP.setValueAt(slSP - slHD, i, 1);
            break;
        }
    }
}
}

public void ChonKH() {
    listKHHD = kháchHangHD.SelectAll();
    indexKHHD = tblChonKH.getSelectedRow();

    txtMaKhachHangHD.setText(String.valueOf(listKHHD.get(indexKHHD).getMaKhachHang()));

    txtTenKhachHangHD.setText(listKHHD.get(indexKHHD).getTenKhachHang());
    txtSdtKhachHangHD.setText(listKHHD.get(indexKHHD).getSoDT());
    indexKHHD = -1;
}

public void CapNhatSoLuongTamThoi() throws SQLException {
    int soLuongHT = (int) tblChonSP.getValueAt(indexSPHD, 1);
    int soLuongThucTai = 0;
    soLuongThucTai = soLuongHT - 1;
    tblChonSP.setValueAt(soLuongThucTai, indexSPHD, 1);
}

public boolean KiemTraNhapSoLuongHD() {
    boolean kiemTra = true;
    try {
        if (txtSoLuongHD.getText().isEmpty()) {

```

```

        MsgBox.alert(this, "Vui lòng nhập số lượng!!!");
        txtSoLuongHD.requestFocus();
        return kiemTra = false;
    } else {
        Double slThuc = Double.valueOf(txtSoLuongHD.getText());
        if (slThuc != slThuc.intValue()) {
            MsgBox.alert(this, "Vui lòng nhập số nguyên!!!");
            txtSoLuongHD.requestFocus();
            return kiemTra = false;
        } else if (slThuc < 0) {
            MsgBox.alert(this, "Vui lòng nhập số lượng lớn hơn hoặc bằng
0!!!");
            txtSoLuongHD.requestFocus();
            return kiemTra = false;
        } else if (slThuc == 0) {
            return kiemTra = true;
        } else {
            return kiemTra = true;
        }
    }
} catch (NumberFormatException e) {
    MsgBox.alert(this, "Vui lòng nhập số!!!");
    txtSoLuongHD.requestFocus();
    return kiemTra = false;
}
}

public void CapNhatSoLuongTamThoiTBLHD() {
    DefaultTableModel model = (DefaultTableModel) tblHoaDon.getModel();
    int indexSanPhamCanTim = -1;
    indexHD = tblHoaDon.getSelectedRow();
    String maSP = (String) tblHoaDon.getValueAt(indexHD, 0);
    listSPHD = sanPhamHD.SelectSanPhamDangKinhDoanh();
    for (int i = 0; i < listSPHD.size(); i++) {
        if (listSPHD.get(i).getMaSanPham().equals(maSP)) {
            indexSanPhamCanTim = i;
            break;
        }
    }
}
soLuongNew = Integer.valueOf(txtSoLuongHD.getText());
if (soLuongNew == 0) {
    model.removeRow(indexHD);
    indexHD = tblHoaDon.getSelectedRow();
}

```

```

        int tongSL = (int) tblChonSP.getValueAt(indexSanPhamCanTim, 1) +
soLuong;
        tblChonSP.setValueAt(tongSL - soLuongNew, indexSanPhamCanTim,
1);
        txtSoLuongHD.setText(null);
        txtSoLuongHD.setEditable(false);
        this.TienPhaiTra();
    } else if (soLuongNew >
listSPHD.getValueAt(indexSanPhamCanTim).getSoLuong()) {
        txtSoLuongHD.setText(String.valueOf(soLuong));
        MsgBox.alert(this, "Số lượng sản phẩm không đủ!!!");
        this.TienPhaiTra();
    } else {
        int tongSL = (int) tblChonSP.getValueAt(indexSanPhamCanTim, 1) +
soLuong;
        tblChonSP.setValueAt(tongSL - soLuongNew, indexSanPhamCanTim,
1);
        tblHoaDon.setValueAt(soLuongNew, indexHD, 2);
        this.TinhTienTungSP();
        this.TienPhaiTra();
    }
    txtSoLuongHD.setText(null);
    soLuong = 0;
    soLuongNew = 0;
    indexHD = -1;
    txtSoLuongHD.setEditable(false);
    lblGhi.requestFocus();
    indexSanPhamCanTim = -1;
}

@SuppressWarnings("empty-statement")
public void chonSP() throws SQLException {
    DefaultTableModel model = (DefaultTableModel) tblHoaDon.getModel();
    DecimalFormat decimalFormat = new DecimalFormat("#.##");
    indexSPHD = tblChonSP.getSelectedRow();
    Integer kiemSL = (Integer) tblChonSP.getValueAt(indexSPHD, 1);
    if (kiemSL == 0) {
        MsgBox.alert(this, "Sản Phẩm Hiện Đã Hết Hàng!!!");
        return;
    }
    @SuppressWarnings("MismatchedReadAndWriteOfArray")
    Object rowData[] = new Object[5];
    if (KiemTraSanPhamDaCo() == true) {

```



```

        @SuppressWarnings("BoxingBoxedValue")
        Double tongTien = 1 * (Double) listSPHD.get(indexSPHD).getGia();
        rowData[0] = listSPHD.get(indexSPHD).getMaSanPham();
        rowData[1] = listSPHD.get(indexSPHD).getTenSanPham();
        rowData[2] = 1;
        rowData[3] =
decimalFormat.format(listSPHD.get(indexSPHD).getGia());
        rowData[4] = decimalFormat.format(tongTien);;
        model.addRow(rowData);
    } else {
        Integer soLuog = (Integer) tblHoaDon.getValueAt(indexSanPhamTrung,
2) + 1;
        @SuppressWarnings("BoxingBoxedValue")
        Double tongTien = soLuog * Double.valueOf((String)
tblHoaDon.getValueAt(indexSanPhamTrung, 3));
        tblHoaDon.setValueAt(soLuog, indexSanPhamTrung, 2);
        tblHoaDon.setValueAt(decimalFormat.format(tongTien),
indexSanPhamTrung, 4);
    }
    this.CapNhatSoLuongTamThoi();
    indexSanPhamTrung = -1;
    indexSPHD = -1;
}

public boolean KiemTraSanPhamDaCo() {
    boolean kiemTra = true;
    indexSPHD = tblChonSP.getSelectedRow();
    String maSanPhamDangChon =
listSPHD.get(indexSPHD).getMaSanPham();
    String maSanPhamCoTrongHD = null;
    for (int i = 0; i < tblHoaDon.getRowCount(); i++) {
        maSanPhamCoTrongHD = (String) tblHoaDon.getValueAt(i, 0);
        if (maSanPhamDangChon.equals(maSanPhamCoTrongHD)) {
            kiemTra = false;
            indexSanPhamTrung = i;
            break;
        } else {
            kiemTra = true;
        }
    }
    return kiemTra;
}

```

```

public void ResetHD() {
    DefaultTableModel model = (DefaultTableModel) tblHoaDon.getModel();
    model.setRowCount(0);
    txtMaKhachHangHD.setText(null);
    txtTenKhachHangHD.setText(null);
    txtSdtKhachHangHD.setText(null);
    lblThanhTienGiamHD.setText("0 đ");
    lblTongTienHD.setText("0 đ");
    txtPhanTramGiam.setText("0");
    maHoaDonChiTietHD = 0;
    maHoaDonHD = 0;
    this.FillTableKHHD(1);
    this.FillTableSPHD(1);
}

public String ChuyenDoiSoThuc(Double tien) {
    String ketQua = null;
    if (tien == tien.intValue()) {
        ketQua = String.valueOf(tien.intValue());
    } else {
        ketQua = String.valueOf(tien);
    }
    return ketQua;
}

public void TinhTienTungSP() {
    DecimalFormat decimalFormat = new DecimalFormat("#.##");
    int sl = (int) tblHoaDon.getValueAt(indexHD, 2);
    Double donGia = Double.valueOf((String) tblHoaDon.getValueAt(indexHD,
3));
    tblHoaDon.setValueAt(decimalFormat.format(sl * donGia), indexHD, 4);
}

public Double TinhTongTienHD() {
    Double tongTien = 0.0;
    for (int i = 0; i < tblHoaDon.getRowCount(); i++) {
        Double tien = Double.valueOf((String) tblHoaDon.getValueAt(i, 4));
        tongTien += tien;
    }
    return tongTien;
}

public Double TinhTienGiam() {

```

```

Double tongTien = TinhTongTienHD();
Double giam = 0.0;
try {
    giam = Double.valueOf(txtPhanTramGiam.getText());
} catch (NumberFormatException e) {
    giam = 0.0;
}
Double tienDuocGiam = tongTien * giam / 100;
Locale localeVN = new Locale("vi", "VN");
NumberFormat currencyFormat =
NumberFormat.getCurrencyInstance(localeVN);
String str = currencyFormat.format(tienDuocGiam);
lblThanhTienGiamHD.setText(str);
return tienDuocGiam;
}

public Double TienPhaiTra() {
    Double tienGiam = TinhTienGiam();
    Double tongTien = TinhTongTienHD();
    Double tienPhaiTra = tongTien - tienGiam;
    Locale localeVN = new Locale("vi", "VN");
    NumberFormat currencyFormat =
NumberFormat.getCurrencyInstance(localeVN);
    String str = currencyFormat.format(tienPhaiTra);
    lblTongTienHD.setText(str);
    return tienPhaiTra;
}

public boolean KiemTraNhapGiamGia() {
    boolean kiemTra = true;
    try {
        if (txtPhanTramGiam.getText().isEmpty()) {
            txtPhanTramGiam.setText("0");
            lblThanhTienGiamHD.setText("0 đ");
            kiemTra = true;
        } else {
            Double phanTramGiam =
Double.valueOf(txtPhanTramGiam.getText());
            if (phanTramGiam != phanTramGiam.intValue()) {
                MsgBox.alert(this, "Vui lòng nhập số nguyên!!!");
                txtPhanTramGiam.setText("0");
                kiemTra = true;
            } else if (Double.valueOf(txtPhanTramGiam.getText()) < 0 ||

```

```

Double.valueOf(txtPhanTramGiam.getText()) > 100) {
    MsgBox.alert(this, "Phần trăm giảm hợp lệ 0 - 100!!!");
    txtPhanTramGiam.setText("0");
   kiemTra = true;
} else if (!txtPhanTramGiam.getText().matches("[0-9]{0,}")) {
    MsgBox.alert(this, "Vui lòng nhập số!!!");
    txtPhanTramGiam.setText("0");
   kiemTra = true;
} else {
}
}
} catch (NumberFormatException e) {
    MsgBox.alert(this, "Vui lòng nhập số!!!");
    txtPhanTramGiam.setText("0");
    txtPhanTramGiam.requestFocus();
   kiemTra = true;
}
return kiemTra;
}

public Integer TaoMaHoaDon() {
    Integer maHoaDon = 0;
    List<HoaDon> list = new ArrayList<>();
    list = hoaDonHD.SelectAll();
    if (list.isEmpty()) {
        maHoaDon = 1;
    } else {
        maHoaDon = list.size() + 1;
    }
    return maHoaDon;
}

@SuppressWarnings({"BoxingBoxedValue", "BoxedValueEquality",
"NumberEquality"})
public Integer TaoMaHoaDonChiTiet() {
    Integer maHoaDonCT = 0;
    List<HoaDonChiTiet> list = new ArrayList<>();
    list = hoaDonChiTietHD.FindIdMax();
    if (list.isEmpty()) {
        maHoaDonCT = 1;
    } else {
        maHoaDonCT = list.get(list.size() - 1).getMaHDCT() + 1;
    }
}

```

```

        return maHoaDonCT;
    }

    public HoaDon getFormHoaDonHD() {
        HoaDon hd = new HoaDon();
        hd.setMaHD(maHoaDonHD);
        hd.setMaNV(nhanVienDangNhap.get(0).getMaNV());
        hd.setMaKH(Integer.valueOf(txtMaKhachHangHD.getText()));
        hd.setPhanTramGiam(Integer.valueOf(txtPhanTramGiam.getText()));
        hd.setGiaGiam(TinhTienGiam());
        hd.setNgayLapHD(txtNgayLapHD.getText());
        hd.setTrangThai("Đã thanh toán");
        hd.setThanhTien(TienPhaiTra());
        return hd;
    }

    public void insertHoaDonHD() {
        HoaDon hd = getFormHoaDonHD();
        if (true) {
            hoaDonHD.insert(hd);
            MsgBox.alert(this, "Thêm hóa đơn thành công!!!");
        }
    }

    public void insertHoaDonChiTietHD() throws SQLException {
        HoaDonChiTiet hdct;
        DefaultTableModel model = (DefaultTableModel) tblHoaDon.getModel();
        for (int i = 0; i < model.getRowCount(); i++) {
            maHoaDonChiTietHD = this.TaoMaHoaDonChiTiet();
            String maSP = (String) model.getValueAt(i, 0);
            Integer sl = (Integer) model.getValueAt(i, 2);
            String donGia = (String) model.getValueAt(i, 3);
            hdct = new HoaDonChiTiet();
            hdct.setMaHD(maHoaDonHD);
            hdct.setMaHDCT(maHoaDonChiTietHD);
            hdct.setMaSP(maSP);
            hdct.setSoLuong(sl);
            hdct.setDonGia(Double.valueOf(donGia));
            hoaDonChiTietHD.insert(hdct);
            int index = 0;
            listSPHD = sanPhamHD.SelectSanPhamDangKinhDoanh();
            for (int j = 0; j < listSPHD.size(); j++) {

```

```

        if (listSPHD.get(j).getMaSanPham().equals(maSP)) {
            index = j;
            break;
        }
    }
    Integer soLuongTon = listSPHD.get(index).getSoLuong();
    Integer ketQua = soLuongTon - sl;
    XJdbc.update("UPDATE SanPham SET SoLuong = ? WHERE MaSP = ?", ketQua, maSP);
}
}

public boolean KiemTraThanhToanHD() {
    booleankiemTra = true;
    if (txtMaKhachHangHD.getText().isEmpty()) {
        MsgBox.alert(this, "Vui lòng chọn khách hàng!!!");
        kiemTra = false;
    } else if (tblHoaDon.getRowCount() == 0) {
        MsgBox.alert(this, "Vui lòng chọn sản phẩm!!!");
        kiemTra = false;
    } else {
        kiemTra = true;
    }
    return kiemTra;
}

public void ThanhToanHoaDonHD() throws SQLException {
    if (KiemTraThanhToanHD() == true) {
        maHoaDonHD = TaoMaHoaDon();
        this.insertHoaDonHD();
        this.insertHoaDonChiTietHD();
        this.ResetHD();
    }
}
}

```

3.3.1.8 Xử lý Form Xem Hóa Đơn

```

int maHDCT = -1;
HoaDonDAO hoaDonHDCT = new HoaDonDAO();
List<HoaDon> listDanhSachHDCT;
KhachHangDAO khachHangHDCT = new KhachHangDAO();
List<KhachHang> listKhachHangHDCT;

```

```

HoaDonChiTietDAO hoaDonChiTietHDCT = new HoaDonChiTietDAO();
List<HoaDonChiTiet> listHoaDonChiTietHDCT;
NhanVienDAO nhanVienHDCT = new NhanVienDAO();
List<NhanVien> listNhanVienHDCT;
SanPhamDAO sanPhamHDCT = new SanPhamDAO();
List<SanPham> listSanPhamHDCT;
int indexChonXemHD = -1;

public void FillDanhSachHDCT(int kieu) {
    DefaultTableModel modelHDCT = (DefaultTableModel)
tblDanhSachHDCT.getModel();
    modelHDCT.setRowCount(0);
    if (kieu == 1) {
        listDanhSachHDCT = hoaDonHDCT.SelectAll();
    }
    if (kieu == 2) {
        listDanhSachHDCT =
hoaDonHDCT.selectByKeyWord(txtTimKiemHDCT.getText());
    }
    Object rowData[] = new Object[3];
    for (int i = 0; i < listDanhSachHDCT.size(); i++) {
        rowData[0] = listDanhSachHDCT.get(i).getMaHD();
        rowData[1] = listDanhSachHDCT.get(i).getMaKH();
        rowData[2] = listDanhSachHDCT.get(i).getNgayLapHD();
        modelHDCT.addRow(rowData);
    }
    if (tblDanhSachHDCT.getRowCount() != 0) {
        for (int i = 0; i < tblDanhSachHDCT.getRowCount(); i++) {
            List<KhachHang> listKhachHangHDCT;
            KhachHangDAO khDAO = new KhachHangDAO();
            int maKH = (Integer) tblDanhSachHDCT.getValueAt(i, 1);
            listKhachHangHDCT = khDAO.SelectByIDS(maKH);
            String tenKH = listKhachHangHDCT.get(0).getTenKhachHang();
            tblDanhSachHDCT.setValueAt(tenKH, i, 1);
        }
    }
}

public void ChonHoaDon() {
    DefaultTableModel modelHDCT = (DefaultTableModel)
tblThongTinHDCT.getModel();
    modelHDCT.setRowCount(0);
    DecimalFormat decimalFormat = new DecimalFormat("#.##");

```

```

        int maHD = (Integer) tblDanhSachHDCT.getValueAt(indexChonXemHD,
0);
        listDanhSachHDCT = hoaDonHDCT.SelectByIDS(maHD);
        txtMaHoaDonHDCT.setText(String.valueOf(maHD));
        txtMaThuNganHDCT.setText(listDanhSachHDCT.get(0).getMaNV());

txtMaKhachHangHDCT.setText(String.valueOf(listDanhSachHDCT.get(0).getM
aKH()));

txtPhanTramGiamHDCT.setText(String.valueOf(listDanhSachHDCT.get(0).get
PhanTramGiam()));
        txtNgayLapHDCT.setText(listDanhSachHDCT.get(0).getNgayLapHD());
        listKhachHangHDCT =
khachHangHDCT.SelectByIDS(Integer.valueOf(txtMaKhachHangHDCT.getTex
t()));

txtTenKhachHangHDCT.setText(listKhachHangHDCT.get(0).getTenKhachHan
g());
        txtSdtKhachHangHDCT.setText(listKhachHangHDCT.get(0).getSoDT());
        listNhanVienHDCT =
nhanVienHDCT.SelectByIDS(txtMaThuNganHDCT.getText());
        txtTenThuNganHDCT.setText(listNhanVienHDCT.get(0).getTenNV());
        listHoaDonChiTietHDCT = hoaDonChiTietHDCT.SelectByMaHD(maHD);
        maHDCT = listHoaDonChiTietHDCT.get(0).getMaHDCT();
        listHoaDonChiTietHDCT = hoaDonChiTietHDCT.SelectByMaHD(maHD);
        Double tienPhaiTra = 0.0, tienDcGiam = 0.0, tongTien = 0.0;
        Object rowData[] = new Object[5];
        for (int i = 0; i < listHoaDonChiTietHDCT.size(); i++) {
            String maSP = listHoaDonChiTietHDCT.get(i).getMaSP();
            listSanPhamHDCT = sanPhamHDCT.SelectByIDS(maSP);
            rowData[0] = maSP;
            rowData[1] = listSanPhamHDCT.get(0).getTenSanPham();
            rowData[2] =
decimalFormat.format(listHoaDonChiTietHDCT.get(i).getSoLuong());
            rowData[3] =
decimalFormat.format(listHoaDonChiTietHDCT.get(i).getDonGia());
            rowData[4] =
decimalFormat.format(listHoaDonChiTietHDCT.get(i).getSoLuong() *
listHoaDonChiTietHDCT.get(i).getDonGia());
            modelHDCT.addRow(rowData);
            tongTien += (listHoaDonChiTietHDCT.get(i).getSoLuong() *
listHoaDonChiTietHDCT.get(i).getDonGia());
        }

```



```

        Locale localeVN = new Locale("vi", "VN");
        NumberFormat currencyFormat =
NumberFormat.getCurrencyInstance(localeVN);
        tienDcGiam = (tongTien *
Integer.valueOf(txtPhanTramGiamHDCT.getText()) / 100);
        tienPhaiTra = tongTien - tienDcGiam;
        lblTienDcGiamHDCT.setText(currencyFormat.format(tienDcGiam));
        lblTienPhaiTraHDCT.setText(currencyFormat.format(tienPhaiTra));
    }

    public void ResetHoaDonDangXem() {
        DefaultTableModel model = (DefaultTableModel)
tblThongTinHDCT.getModel();
        txtMaHoaDonHDCT.setText(null);
        txtMaThuNganHDCT.setText(null);
        txtMaKhachHangHDCT.setText(null);
        txtPhanTramGiamHDCT.setText(null);
        txtNgayLapHDCT.setText(null);
        txtTenKhachHangHDCT.setText(null);
        txtSdtKhachHangHDCT.setText(null);
        txtTenThuNganHDCT.setText(null);
        lblTienDcGiamHDCT.setText("0 đ");
        lblTienPhaiTraHDCT.setText("0 đ");
        model.setRowCount(0);
        maHDCT = 0;
        txtTimKiemHDCT.setText("Tìm hóa đơn. . .");
        txtTimKiemHDCT.nextFocus();
        this.FillDanhSachHDCT(1);
    }

    public void XoaHoaDon() throws SQLException {
        if (tblThongTinHDCT.getRowCount() != 0) {
            boolean chon = MsgBox.confirm(this, "Bạn chắc chắn muốn xóa HD
này?");
            if (chon == true) {
                Integer maHD = Integer.valueOf(txtMaHoaDonHDCT.getText());
                @SuppressWarnings("UnnecessaryBoxing")
                String maSP = null;
                int sl = 0;
                listSanPhamHDCT = sanPhamHDCT.SelectAll();
                for (int i = 0; i < listSanPhamHDCT.size(); i++) {
                    for (int j = 0; j < tblThongTinHDCT.getRowCount(); j++) {
                        if

```

```
(listSanPhamHDCT.get(i).getMaSanPham().equals(tblThongTinHDCT.getValueAt(j, 0))) {  
    maSP = listSanPhamHDCT.get(i).getMaSanPham();  
    sl = listSanPhamHDCT.get(i).getSoLuong() +  
Integer.valueOf((String) tblThongTinHDCT.getValueAt(j, 2));  
    XJdbc.update("UPDATE SanPham SET SoLuong = ?  
WHERE MaSP = ?", sl, maSP);  
    break;  
}  
}  
}  
hoaDonChiTietHDCT.delete(maHD);  
hoaDonHDCT.delete(maHD);  
this.ResetHoaDonDangXem();  
this.FillDanhSachHDCT(1);  
MsgBox.alert(this, "Đã xóa thành công hóa đơn " + maHD + ".");  
}  
} else {  
    MsgBox.alert(this, "Vui lòng chọn hóa đơn!!!");  
}  
}
```

3.3.1.9 Xử lý Form Danh Mục Khách Hàng

```

KhachHangDAO daoKH = new KhachHangDAO();
int indexKH = -1;
List<KhachHang> listKH = daoKH.SelectAll();

public void FillTableKH(int chon) {
    DefaultTableModel model = (DefaultTableModel)
tblThongTinKH.getModel();
    model.setRowCount(0);
    if (chon == 1) {
        listKH = daoKH.SelectAll();
    }
    if (chon == 2) {
        listKH = daoKH.selectByKeyWord(txtTimKiemKH.getText());
    }
    Object rowData[] = new Object[3];
    for (int i = 0; i < listKH.size(); i++) {
        rowData[0] = listKH.get(i).getMaKhachHang();
        rowData[1] = listKH.get(i).getTenKhachHang();
    }
}

```

```

        rowData[2] = listKH.get(i).getSoDT();
        model.addRow(rowData);
    }
    this.UpdateStatusKH();
}

public void TimKhachHangKH() {
    if (txtTimKiemKH.getText().isEmpty() ||
txtTimKiemKH.getText().equals("Tìm Kiếm. . .")) {
        this.FillTableKH(1);
    } else {
        this.FillTableKH(2);
    }
}

public void SetFormKH(KhachHang kh) {
    txtMaKH.setText(String.valueOf(kh.getMaKhachHang()));
    txtTenKhachHangKH.setText(kh.getTenKhachHang());
    txtSdtKH.setText(kh.getSoDT());
}

public KhachHang GetFormKH() {
    KhachHang kh = new KhachHang();
    kh.setMaKhachHang(Integer.valueOf(txtMaKH.getText()));
    kh.setTenKhachHang(txtTenKhachHangKH.getText());
    kh.setSoDT(txtSdtKH.getText());
    return kh;
}

public Integer TaoMaKhachHang() {
    Integer maKhachHang = 0;
    KhachHangDAO dao = new KhachHangDAO();
    List<KhachHang> list = new ArrayList<>();
    list = dao.SelectAll();
    if (list.isEmpty()) {
        maKhachHang = 1;
    } else {
        maKhachHang = list.get(list.size() - 1).getMaKhachHang() + 1;
    }
    return maKhachHang;
}

public void InsertKH() {

```

```

        KhachHang kh = GetFormKH();
        if (CheckKH(1) == true) {
            daoKH.insert(kh);
            this.FillTableKH(1);
            this.ResetKH();
            MsgBox.alert(this, "Thêm khách hàng mới thành công!");
        }
        lblThemKH.requestFocus();
    }

    public void UpdateKH() {
        KhachHang dvt = GetFormKH();
        if (CheckKH(0) == true) {
            daoKH.update(dvt);
            this.FillTableKH(1);
            MsgBox.alert(this, "Chỉnh sửa khách hàng " + txtMaKH.getText() + " thành công!");
        }
        lblSuaKH.requestFocus();
    }

    public void DeleteKH() {
        if (indexKH == -1) {
            MsgBox.alert(this, "Vui lòng chọn khách hàng!");
        } else {
            Integer maKhachHang = Integer.valueOf(String.valueOf(tblThongTinKH.getValueAt(indexKH, 0)));
            listKH = daoKH.selectByKhachHangngInHoaDon(maKhachHang);
            if (listKH.isEmpty()) {
                if (maKhachHang == 1) {
                    MsgBox.alert(this, "Thông Tin Khách Hàng Mặc Định!\nKhông Được Xóa Thông Tin Đây!");
                } else {
                    boolean n = MsgBox.confirm(this, "Bạn chắc chắn muốn xóa khách hàng này không???");
                    if (n == true) {
                        daoKH.delete(maKhachHang);
                        this.FillTableKH(1);
                        this.ResetKH();
                        MsgBox.alert(this, "Xóa thành công khách hàng " + maKhachHang + " !");
                    } else {
                        MsgBox.alert(this, "Đã hoàn tác lệnh xóa!");
                    }
                }
            }
        }
    }

```

```

    }
    }
    } else {
        MsgBox.alert(this, "Khách hàng đã được lập hóa đơn!\nKhông được
xóa khách hàng này!");
    }
}
lblXoaKH.requestFocus();
}

public void ResetKH() {
    KhachHang kh = new KhachHang();
    this.SetFormKH(kh);
    this.indexKH = -1;
    txtMaKH.setText(String.valueOf(TaoMaKhachHang()));
    txtTenKhachHangKH.setText("");
    txtSdtKH.setText("");
    this.UpdateStatusKH();
    lblResetKH.requestFocus();
}

public void EditKH() {
    Integer maKH =
Integer.valueOf(String.valueOf(tblThongTinKH.getValueAt(this.indexKH, 0)));
    KhachHang kh = daoKH.SelectByID(maKH);
    this.SetFormKH(kh);
    txtMaKH.setEnabled(false);
}

public void UpdateStatusKH() {
    if (indexKH == -1) {
        lblSuaKH.setEnabled(false);
        lblThemKH.setEnabled(true);
    } else {
        lblSuaKH.setEnabled(true);
        lblThemKH.setEnabled(false);
    }
}

public boolean CheckKH(int mucDich) {
    booleankiemTraKH = true;
    listKH = daoKH.SelectAll();
    String sdt = txtSdtKH.getText();

```

```

if (txtTenKhachHangKH.getText().trim().isEmpty()) {
    MsgBox.alert(this, "Vui lòng nhập tên khách hàng!");
   kiemTraKH = false;
} else if (txtSdtKH.getText().isEmpty()) {
    MsgBox.alert(this, "Vui lòng nhập số điện thoại!");
   kiemTraKH = false;
} else {
    for (int i = 0; i < listKH.size(); i++) {
        if (mucDich == 1) {
            if (sdt.equals(listKH.get(i).getSoDT())) {
                MsgBox.alert(this, "Số điện thoại khách hàng đã tồn tại!");
               kiemTraKH = false;
            }
        }
    }
}
returnkiemTraKH;
}

```

3.3.1.10 Xử lý Form Xem Doanh Thu

```

public boolean KiemTraNgay() {
    if (txtNBD.getText().trim().isEmpty()) {
        MsgBox.alert(this, "Vui lòng nhập ngày bắt đầu!!!");
        txtNBD.requestFocus();
        return false;
    } else if (txtNKT.getText().trim().isEmpty()) {
        MsgBox.alert(this, "Vui lòng nhập ngày kết thúc!!!");
        txtNKT.requestFocus();
        return false;
    }
    DateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
    sdf.setLenient(false);
    try {
        String date1 = txtNBD.getText().replace("/", "-");
        String date2 = date1.replace(".", "-");
        sdf.parse(date2);
    } catch (ParseException e) {
        MsgBox.alert(this, "Ngày bắt đầu không hợp lệ!\nVui lòng nhập ngày theo định dạng dd-MM-YYYY!");
        txtNBD.requestFocus();
        return false;
    }
}

```

```

    }
    try {
        String date1 = txtNKT.getText().replace("/", "-");
        String date2 = date1.replace(".", "-");
        sdf.parse(date2);
    } catch (ParseException e) {
        MsgBox.alert(this, "Ngày kết thúc không hợp lệ!\nVui lòng nhập ngày theo định dạng dd-MM-YYYY!");
        txtNKT.requestFocus();
        return false;
    }
    return true;
}

public String ChuyenNgay(JTextField txtField, String dinhDangBegin, String dinhDangEnd) throws ParseException {
    String dateBD1 = txtField.getText().replace("/", "-");
    String dateBD2 = dateBD1.replace(".", "-");
    Date date = new SimpleDateFormat("dd-MM-yyyy").parse(dateBD2);
    SimpleDateFormat f = new SimpleDateFormat("yyyy-MM-dd");
    String kq = f.format(date);
    return kq;
}

public void fillTableDiemChuyenDe() throws ParseException {
    DoanhThuDAO daoDT = new DoanhThuDAO();
    DefaultTableModel model = (DefaultTableModel) tblDoanhThu.getModel();
    model.setRowCount(0);
    String ngayBD = ChuyenNgay(txtNBD, "dd-MM-yyyy", "yyyy-MM-dd");
    String ngayKT = ChuyenNgay(txtNKT, "dd-MM-yyyy", "yyyy-MM-dd");
    List<Object[]> list = daoDT.getDoanhThu(ngayBD, ngayKT);
    for (Object[] row : list) {
        model.addRow(new Object[]{row[0], row[1], row[2]});
    }
    Locale localeVN = new Locale("vi", "VN");
    NumberFormat currencyFormat =
    NumberFormat.getCurrencyInstance(localeVN);

    tblDoanhThu.setValueAt(currencyFormat.format(tblDoanhThu.getValueAt(0, 0)), 0, 0);

    tblDoanhThu.setValueAt(currencyFormat.format(tblDoanhThu.getValueAt(0, 1)), 0, 1);

```

```
tblDoanhThu.setValueAt(currencyFormat.format(tblDoanhThu.getValueAt(0,
2)), 0, 2);
}
```

3.3.1.11 Xử lý Form Login

```
static String phoneLogin = null;
static String positionLogin = null;
NhanVienDAO nhanVienDAO = new NhanVienDAO();
TaiKhoanDAO taiKhoanDAO = new TaiKhoanDAO();
List<NhanVien> listNhanVien;
List<TaiKhoan> listTaiKhoan;
int dem = 0;
int clickFirst = 0;

public LoginDialog() {
    initComponents();
    this.setLocationRelativeTo(null);
    txtUser.setBackground(new java.awt.Color(0, 0, 0, 1));
    pwdPass.setBackground(new java.awt.Color(0, 0, 0, 1));
    lblTitle.requestFocus();
    this.setIconImage(XImage.getAppIcon());
    pwdPass.setEchoChar((char) 0);
    lblAnHienPass.setEnabled(false);
}

public boolean ValidateFormLogin() {
    String isPhone = "^([0]+[3_5_7-9]+[0-9]){8}";
    String isPassWord = ".{8}";
    if (txtUser.getText().equals("Số Điện Thoại") || txtUser.getText().isEmpty())
    {
        MsgBox.alert(this, "Vui lòng nhập Username");
        return false;
    } else if (pwdPass.getText().equals("Mật Khẩu") ||
pwdPass.getText().isEmpty()) {
        MsgBox.alert(this, "Vui lòng nhập Password");
        return false;
    } else if (!txtUser.getText().matches(isPhone)) {
        MsgBox.alert(this, "Số điện thoại không hợp lệ!");
        return false;
    } else if (!pwdPass.getText().matches(isPassWord)) {
```



```

        MsgBox.alert(this, "Mật khẩu không hợp lệ!\nMật khẩu ít nhất 8 ký tự");
        return false;
    }
    return true;
}

@SuppressWarnings("null")
public void Login() {
    listTaiKhoan = taiKhoanDAO.SelectAll();
    listNhanVien = nhanVienDAO.SelectAll();
    int check = 0;
    String phone = null;
    String pass = null;
    String position = null;
    for (int i = 0; i < listNhanVien.size(); i++) {
        if (txtUser.getText().equals(listNhanVien.get(i).getSoDienThoai())) {
            check = 1;
            phone = txtUser.getText();
            position = listNhanVien.get(i).getMaChucVu();
            break;
        }
    }
    if (check == 0) {
        MsgBox.alert(this, "Số điện thoại hoặc mật khẩu không đúng!\nVui lòng kiểm tra lại!");
        reset();
    } else {
        check = 0;
        for (int i = 0; i < listTaiKhoan.size(); i++) {
            if (phone.equals(listTaiKhoan.get(i).getSoDienThoai())) {
                check = 1;
                pass = listTaiKhoan.get(i).getMatKhau();
                break;
            }
        }
    }
    if (check == 1) {
        if (pwdPass.getText().equals(pass)) {
            MsgBox.alert(this, "Đăng nhập thành công!");
            phoneLogin = phone;
            positionLogin = position;
            this.dispose();
            new MainInterfaceDialog().setVisible(true);
        } else {

```

```

        MsgBox.alert(this, "Số điện thoại hoặc mật khẩu không đúng!\nVui lòng kiểm tra lại!");
        reset();
    }
} else {
    MsgBox.alert(this, "Số điện thoại hoặc mật khẩu không đúng!\nVui lòng kiểm tra lại!");
    reset();
}
}

public void clickPassword() {
    if (clickFirst == 0) {
        pwdPass.setText("");
        lblAnHienPass.setIcon(new
        ImageIcon(getClass().getResource("/com/software/icon/hidden.png")));
        lblAnHienPass.setEnabled(true);
        pwdPass.setEchoChar('●');
        dem = 0;
    }
    clickFirst ++;
}

public void clickHidePass() {
    if (!pwdPass.getText().equals("Mật Khẩu")) {
        if (dem % 2 != 0) {
            lblAnHienPass.setIcon(new
            ImageIcon(getClass().getResource("/com/software/icon/hidden.png")));
            pwdPass.setEchoChar('●');
        } else {
            lblAnHienPass.setIcon(new
            ImageIcon(getClass().getResource("/com/software/icon/view.png")));
            pwdPass.setEchoChar((char) 0);
        }
        dem++;
    }
}

public void reset() {
    dem = 0;
    clickFirst = 0;
    txtUser.setText("Số Điện Thoại");
}

```

```

        pwdPass.setText("Mật Khẩu");
        lblAnHienPass.setIcon(new
        ImageIcon(getClass().getResource("/com/software/icon/view.png")));
        pwdPass.setEchoChar((char) 0);
        lblAnHienPass.setEnabled(false);
        lblLogin.requestFocus();
    }

```

3.3.1.12 Xử lý Form ForgotPasswordDialog

```

public ForgotPasswordDialog() {
    initComponents();
    this.setLocationRelativeTo(null);
    txtGmail.setBackground(new java.awt.Color(0, 0, 1, 0));
    lblTitle.requestFocus();
    this.setIconImage(XImage.getApplIcon());
}

static int randomInt;
static String mailTo = null;

@SuppressWarnings({"CallToPrintStackTrace", "UseSpecificCatch"})
public void GuiMail() {
    for (int i = 1; i < 2; i++) {
        double randomDouble = Math.random();
        randomDouble = randomDouble * 1000000 + 1;
        randomInt = (int) randomDouble;
        if (String.valueOf(randomInt).length() < 6) {
            i = 1;
        } else {
            break;
        }
    }
    try {
        String taiKhoan = "hotroshopbingbong@gmail.com";
        String matKhau = "Bingbong123@";

        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");
    }
}

```

```

        Session session = Session.getDefaultInstance(prop, new
javax.mail.Authenticator() {
            @Override
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(taiKhoan, matKhau);
            }
        });

```

```

String from = "hotroshopbingbong@gmail.com";
String to = txtGmail.getText();
String subject = "Mã xác nhận Bing Bong Shop ";
String body = "Mã xác nhận của bạn là : " + randomInt;

```

```

MimeMessage message = new MimeMessage(session);
message.setFrom(new InternetAddress(from));
message.setRecipients(
    Message.RecipientType.TO,
    InternetAddress.parse(to)
);
message.setSubject(subject);
message.setText(body);

```

```

    MailSender.queue(message);
} catch (Exception e) {
    JOptionPane.showMessageDialog(this, e.getMessage());
    e.printStackTrace();
}
}

```

```

boolean checkForm() {
    boolean check = true;
    int kiemTra = 0;
    String isGmail= "([a-zA-z]{3,})+[pP]+[cC]+[0-9]{5}+@fpt.edu.vn)|([a-zA-Z0-9]{6,30}+@gmail.com)";
    NhanVienDAO nvDAO = new NhanVienDAO();
    List<NhanVien> listNV = nvDAO.SelectAll();
    for (int i = 0; i < listNV.size(); i++) {
        if (txtGmail.getText().equals(listNV.get(i).getGmail())) {
            kiemTra = 1;
            break;
        }
    }
}

```

```

    if (txtGmail.getText().isEmpty() || txtGmail.getText().equals("Nhập Gmail"))
    {
        check = false;
        MsgBox.alert(this, "Vui lòng nhập gmail của bạn!");
    } else if (!txtGmail.getText().matches(isGmail)){
        check = false;
        MsgBox.alert(this, "Gmail chưa hợp lệ.\nVui lòng nhập lại!");
    } else if (kiemTra == 0) {
        check = false;
        MsgBox.alert(this, "Gmail chưa được đăng ký!\nVui lòng nhập lại!");
    }
    else {
        check = true;
    }
    return check;
}

```

3.3.1.13 Xử lý Form OTPDialog

```

int second = 60;
Timer timer;

public OTPDialog() {
    initComponents();
    this.setLocationRelativeTo(null);
    lblTitle.requestFocus();
    this.setIconImage(XImage.getAppIcon());
    setTime();
}

@SuppressWarnings({"UseSpecificCatch", "CallToPrintStackTrace"})
public void GuiMail() {
    for (int i = 1; i < 2; i++) {
        double randomDouble = Math.random();
        randomDouble = randomDouble * 1000000 + 1;
        randomInt = (int) randomDouble;
    }
    try {
        String taiKhoan = "hotroshopbingbong@gmail.com";
        String matKhau = "Bingbong123@";

        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");

Session session = Session.getDefaultInstance(prop, new
javax.mail.Authenticator() {
    @Override
    protected PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication(taiKhoan, matKhau);
    }
});

String from = "hotroshopbingbong@gmail.com";
String to = ForgotPasswordDialog.mailTo;
String subject = "Mã xác nhận Bing Bong Shop ";
String body = "Mã xác nhận của bạn là : " + randomInt;

MimeMessage message = new MimeMessage(session);
message.setFrom(new InternetAddress(from));
message.setRecipients(
    Message.RecipientType.TO,
    InternetAddress.parse(to)
);
message.setSubject(subject);
message.setText(body);
MailSender.queue(message);
} catch (Exception e) {
    JOptionPane.showMessageDialog(this, e.getMessage());
    e.printStackTrace();
}
}

public void setTime() {
    timer = new Timer(1000, (e) -> {
        second--;
        lblTime.setText("00:" + String.valueOf(second));
        if (second == 0) {
            ForgotPasswordDialog.randomInt = 0;
            lblTime.setText("Mã xác nhận đã hết hiệu lực!");
            timer.stop();
        }
    });
}

```

```

        timer.start();
    }

    boolean checkCode() {
        try {
            String code = txtNumber1.getText() + txtNumber2.getText() +
txtNumber3.getText() + txtNumber4.getText() + txtNumber5.getText() +
txtNumber6.getText();
            if (Integer.valueOf(code) != ForgotPasswordDialog.randomInt) {
                MsgBox.alert(this, "Mã xác nhận chưa chính xác vui lòng kiểm tra
lại!!!");
                return false;
            } else if (code.isEmpty()) {
                MsgBox.alert(this, "Vui lòng nhập mã xác thực!!!");
                return false;
            }
        } catch (NumberFormatException e) {
            MsgBox.alert(this, "Mã xác nhận chưa chính xác vui lòng kiểm tra
lại!!!");
            return false;
        }
        return true;
    }

    public void PasteCode(JTextField txtOne, JTextField txtTwo) {
        if (txtOne.getText().length() == 1) {
            txtOne.nextFocus();
            txtTwo.requestFocus();
        } else if (txtOne.getText().length() > 1) {
            int index1 = txtOne.getText().charAt(0);
            char kiTu = (char) index1;
            txtOne.setText("");
            txtOne.setText(String.valueOf(kiTu));
        }
    }
}

```

3.3.1.14 Xử lý Form ChangePassDialog

```

int demPass = 0, demConfirm = 0;
int clickFirstPass = 0, clickFirstConfirm = 0;

```

```

public ChangePassDialog() {
    initComponents();
    this.setLocationRelativeTo(null);
    lblTitle.requestFocus();
    pwdPass.setBackground(new java.awt.Color(0, 0, 1, 0));
    pwdConfirm.setBackground(new java.awt.Color(0, 0, 1, 0));
    this.setIconImage(XImage.getAppIcon());
    pwdPass.setEchoChar((char) 0);
    pwdConfirm.setEchoChar((char) 0);
}

public void clickPasswordField(int click, JPasswordField PwField, JLabel
lblLabel, int index) {
    if (click == 0) {
        PwField.setText("");
        lblLabel.setIcon(new
ImageIcon(getClass().getResource("/com/software/icon/hidden.png")));
        lblLabel.setEnabled(true);
        PwField.setEchoChar('●');
        index = 0;
    }
    click++;
}

public void clickHide(JPasswordField PwField, JLabel lblLabel, int index,
String chuoi) {
    if (!PwField.getText().equals(chuoi)) {
        if (index % 2 != 0) {
            lblLabel.setIcon(new
ImageIcon(getClass().getResource("/com/software/icon/hidden.png")));
            PwField.setEchoChar('●');
        } else {
            lblLabel.setIcon(new
ImageIcon(getClass().getResource("/com/software/icon/view.png")));
            PwField.setEchoChar((char) 0);
        }
        index++;
    }
}

public boolean CheckForm() {
    String isPassWord = "^.{8,}";
    boolean check = true;

```



```

        if (pwdPass.getText().isEmpty() || pwdPass.getText().equals("Mật Khẩu"))
        {
            MsgBox.alert(this, "Vui lòng nhập mật khẩu!");
            check = false;
        } else if (pwdConfirm.getText().isEmpty() ||
pwdConfirm.getText().equals("Xác Nhận Mật Khẩu")) {
            MsgBox.alert(this, "Vui lòng xác nhận mật khẩu!");
            check = false;
        } else if (!pwdPass.getText().matches(isPassWord)) {
            MsgBox.alert(this, "Mật khẩu không hợp lệ!\nMật khẩu ít nhất 8 ký tự");
            check = false;
        } else if (!pwdConfirm.getText().equals(pwdPass.getText())) {
            MsgBox.alert(this, "Xác nhận mật khẩu không đúng\nVui lòng kiểm tra
lại!");
            check = false;
        }
        return check;
    }

    public void UpdatePass() {
        TaiKhoanDAO tkDAO = new TaiKhoanDAO();
        NhanVienDAO nvDAO = new NhanVienDAO();
        List<NhanVien> listNV = nvDAO.SelectAll();
        String soDienThoai = null;
        for (int i = 0; i < listNV.size(); i++) {
            if (ForgotPasswordDialog.mailTo.equals(listNV.get(i).getGmail())) {
                soDienThoai = listNV.get(i).getSoDienThoai();
                break;
            }
        }
        tkDAO.updatePass(pwdPass.getText(), soDienThoai);
        MsgBox.alert(this, "Đổi mật khẩu thành công!");
    }

```

4 KIỂM THỬ

4.1 KIỂM THỬ FORM LOGIN DIALOG:

TH	Mô Tả Tình Huống	Dữ Liệu Mẫu	Kết Quả Mong Đợi	Fix
1	Đăng nhập đúng tên đăng nhập và mật khẩu	Tài khoản: 0946492294 Mật khẩu: admin@123	Đăng nhập thành công và hiện thông báo đăng nhập thành công, đóng cửa sổ hiện tại và mở lên cửa sổ chính	Hoàn thành
2	Đăng nhập với tên đăng nhập bỏ trống	Mật khẩu: admin@123	Hệ thống hiển thị thông báo vui lòng nhập tên đăng nhập	Hoàn thành
3	Đăng nhập với mật khẩu bỏ trống	Tài khoản: 0946492294	Hệ thống hiển thị thông báo vui lòng nhập mật khẩu	Hoàn thành
4	Đăng nhập bỏ trống cả tên đăng nhập và mật khẩu		Hệ thống hiển thị thông báo vui lòng nhập tên đăng nhập và mật khẩu	Hoàn thành
5	Đăng nhập với tên đăng nhập sai	Tài khoản: 0946492295 Mật khẩu: admin@123	Hệ thống hiển thị thông báo tên đăng nhập không tồn tại	Hoàn thành
6	Đăng nhập với mật khẩu sai	Tài khoản: 0946492294 Mật khẩu: admin@124	Hệ thống hiển thị thông báo sai mật khẩu	Hoàn thành

7	Chọn kết thúc thay vì đăng nhập		Hệ thống đóng phần mềm	Hoàn thành
8	Click vào icon hiện mật khẩu		Hiển thị được mật khẩu và icon hiển thị thành icon ẩn	Hoàn thành
9	Click vào icon ẩn mật khẩu		Ẩn được mật khẩu và icon ẩn thành icon hiển thị	Hoàn thành
10	Click vào Forgot password?		Hệ thống chuyển sang Form nhập Gmail để lấy lại mật khẩu	Hoàn thành

4.2 KIỂM THỬ FORM FORGOT PASSWORD DIALOG:

TH	Mô Tả Tình Huống	Dữ Liệu Mẫu	Kết Quả Mong Đợi	Fix
1	Nhập vào đúng Gmail đã đăng ký	quangdat0156@gmail.com	Hệ thống đóng Form nhập Gmail chuyển sang Form nhập mã OTP DiaLog	Hoàn thành
2	Nhập vào sai định dạng Gmail	quangdat0156@gmail	Hệ thống thông báo Gmail chưa hợp lệ. Vui lòng nhập lại!	Hoàn thành
3	Nhập vào Gmail chưa đăng ký	quangdat01567@gmail.com	Hệ thống thông báo Gmail chưa được đăng ký! Vui lòng nhập lại!	Hoàn thành
4	Nhập vào sai định dạng Gmail dưới 6 ký tự	quang@gmail.com	Hệ thống thông báo Gmail chưa hợp lệ. Vui lòng nhập lại!	Hoàn thành

5	Nhập vào sai định dạng Gmail trên 30 ký tự	a01234567890 123456789012 3456789@gmail.com	Hệ thống thông báo Gmail chưa hợp lệ. Vui lòng nhập lại!	Hoàn thành
6	Nhập vào Gmail có ký tự đặc biệt	quangdat,0156@gmail.com	Hệ thống thông báo Gmail chưa hợp lệ. Vui lòng nhập lại!	Hoàn thành
7	Click vào icon mũi tên hướng sang trái		Hệ thống quay lại Form Login Dialog	Hoàn thành
8	Click vào Button CONFIRM		Hệ thống chuyển sang Form nhập mã OTP Dialog	Hoàn thành
9	Chọn kết thúc thay vì CONFIRM		Hệ thống đóng phần mềm	Hoàn thành

4.3 KIỂM THỬ FORM OTP DIALOG:

TH	Mô Tả Tình Huống	Dữ Liệu Mẫu	Kết Quả Mong Đợi	Fix
1	Nhập vào 6 chữ số ngẫu nhiên gửi về Gmail	999999	Hệ thống đóng Form OTP Dialog chuyển sang Form ChangePass Dialog	Hoàn thành
2	Nhập vào 6 chữ số ngẫu nhiên gửi về Gmail nhưng quá thời gian		Hệ thống thông báo Mã xác nhận hết hiệu lực!	Hoàn thành
3	Nhập vào không đúng 6 số OTP	999998	Hệ thống thông báo mã xác nhận chưa chính xác vui lòng kiểm tra lại!!!	Hoàn thành

4	Nhập vào dãy OTP bằng chữ	abcdef	Hệ thống thông báo mã xác nhận chưa chính xác vui lòng kiểm tra lại!!!	Hoàn thành
5	Bỏ trống dãy OTP		Hệ thống thông báo mã xác nhận chưa chính xác vui lòng kiểm tra lại!!!	Hoàn thành
6	Click vào Resend Code		Hệ thống tự động gửi lại mã OTP mới và reset lại thời gian nhập mã OTP	Hoàn thành
7	Click vào icon mũi tên hướng sang trái		Hệ thống quay lại Form Forgot Password Dialog	Hoàn thành
8	Click vào Button CONFIRM		Hệ thống chuyển sang Form nhập mã OTP Dialog	Hoàn thành
9	Chọn kết thúc thay vì CONFIRM		Hệ thống đóng phần mềm	Hoàn thành

4.4 KIỂM THỬ FORM CHANGE PASS DIALOG:

TH	Mô Tả Tình Huống	Dữ Liệu Mẫu	Kết Quả Mong Đợi	Fix
1	Nhập vào mật khẩu mới và nhập lại xác nhận mật khẩu mới	admin@1234 admin@1234	Hệ thống thông báo Đổi mật khẩu thành công! và chuyển sang Form Login Dialog	Hoàn thành

2	Nhập vào mật khẩu mới	admin@1234	Hệ thống thông báo Xác nhận mật khẩu không đúng Vui lòng kiểm tra lại!	Hoàn thành
3	Xác nhận mật khẩu mới	admin@1234	Hệ thống thông báo Vui lòng nhập mật khẩu!	Hoàn thành
4	Nhập vào mật khẩu mới và nhập sai xác nhận mật khẩu mới	admin@1234 admin@12345	Hệ thống thông báo Xác nhận mật khẩu không đúng Vui lòng kiểm tra lại!	Hoàn thành
5	Nhập vào mật khẩu mới dưới 8 ký tự	admin	Hệ thống thông báo Mật khẩu không hợp lệ! Mật khẩu ít nhất 8 ký tự!	Hoàn thành
6	Click vào icon hiện mật khẩu		Hiện thị được mật khẩu và icon hiển thị thành icon ẩn	Hoàn thành
7	Click vào icon ẩn mật khẩu		Ẩn được mật khẩu và icon ẩn thành icon hiển thị	Hoàn thành
8	Click vào icon mũi tên hướng sang trái		Hệ thống quay lại Form Forgot Password Dialog	Hoàn thành
9	Click vào Button CONFIRM		Hệ thống chuyển sang Form Login Dialog	Hoàn thành
10	Chọn kết thúc thay vì CONFIRM		Hệ thống đóng phần mềm	Hoàn thành

4.5 KIỂM THỬ FORM NHÂN VIÊN

TH	Mô Tả Tình Huống	Dữ Liệu Mẫu	Kết Quả Mong Đợi	Fix
1	Click vào một dòng trong danh sách nhân viên		Hiển thị thông tin nhân viên đó lên form	Hoàn thành
2	Thêm nhân viên thất bại, bỏ trống mã nhân viên, họ tên và mật khẩu		Hệ thống hiển thị thông báo không được bỏ trống	Hoàn thành
3	Thêm nhân viên thất bại, nhập sai định dạng số điện thoại	NV01,Trọng Nhân ,Nam , quản lý , 111111111,tron gnhan@gamil.com, 364236567, TP. cần thơ	Hệ thống hiển thị thông báo: "Số điện thoại không đúng - Vui lòng kiểm tra lại"	Hoàn thành
4	Thêm nhân viên thất bại, nhập sai định dạng Gmail	NV01,Trọng Nhân ,Nam , quản lý , 0769331203,tron gnhan122, 364236567, TP. cần thơ	Hệ thống hiển thị thông báo: "Email không đúng - Vui lòng kiểm tra lại"	Hoàn thành
5	Thêm nhân viên thất bại, nhập Gmail chứa khoảng trắng	NV01,Trọng Nhân ,Nam , quản lý , 0769331203,tron gnhan 122, 364236567, TP. cần thơ	Hệ thống hiển thị thông báo: "Email không được chứa khoảng trắng!"	Hoàn thành
6	Thêm nhân viên thất bại, số điện thoại không được phép trùng	Nhập một số điện thoại đã có trong danh sách "0769331203"	Hệ thống hiển thị thông báo: "số điện thoại không được phép trùng - Vui lòng kiểm tra lại"	Hoàn thành

7	Thêm nhân viên thất bại, Email không được phép trùng	Nhập một số số điện thoại đã có trong danh sách "TrongNhan@gmail.com"	Hệ thống hiển thị thông báo: "Email không được phép trùng - Vui lòng kiểm tra lại"	Hoàn thành
8	Thêm nhân viên thất bại, MaNV không được phép trùng	Nhập một số số điện thoại đã có trong danh sách "NV01"	Hệ thống hiển thị thông báo: "Mã Nhân Viên không được phép trùng - Vui lòng kiểm tra lại"	Hoàn thành
9	Thêm nhân viên thất bại, bỏ trống một trong những dữ liệu		Hệ thống hiển thị thông báo không được bỏ trống	Hoàn thành
10	Thêm nhân viên thành công	NV01, Trọng Nhân, Nam, quản lý, 0769331203, trongnhan@gmail.com, 364236567, TP. cần thơ	Hệ thống hiển thị thông báo "Thêm nhân viên thành công"	Hoàn thành
11	Kiểm tra Enabled của nút sửa nhân viên (chưa chọn dữ liệu)		btnSua.setEnabled(false)	Hoàn thành
12	Kiểm tra Enabled của nút sửa nhân viên (click vào một dòng dữ liệu trong bảng)		btnSua.setEnabled(true)	Hoàn thành
13	Cập nhật thất bại với tài khoản admin	chọn một dòng tài liệu có vai trò là admin	Hệ thống hiển thị thông báo: "Không thể thao tác với tài khoản admin!"	Hoàn thành
14	Kiểm tra Enabled của nút Xóa nhân viên (chưa chọn dữ liệu)		btnXoa.setEnabled(false)	Hoàn thành

15	Kiểm tra Enabled của nút Xóa nhân viên (click vào một dòng dữ liệu trong bảng)		btnXoa setEnabled(true)	Hoàn thành
16	Xóa thất bại , không thể xóa tài khoản admin	chọn một dòng tài liệu có vai trò là admin	Hệ thống hiển thị thông báo: “Không thể thao tác với tài khoản admin!”	Hoàn thành
17	Xóa nhân viên thành công	chọn một dòng dữ liệu vai trò khác admin	Hệ thống hiển thị thông báo xác nhận từ người dùng , Ok : xóa nhân viên thành công Cancel : quay lại	hoàn thành
18	Kiểm tra nút làm mới		xóa trống form , set Enable các nút	hoàn thành

4.6 KIỂM THỬ FORM CHỨC VỤ:

TH	Mô Tả Tình Huống	Dữ Liệu Mẫu	Kết Quả Mong Đợi	Fix
1	Thêm chức vụ thất bại, bỏ trống một trong những dữ liệu		Hệ thống hiển thị thông báo không được bỏ trống	Hoàn thành
2	Thêm nhân viên thất bại, Nhập sai định dạng mã	CV1	Hệ thống hiển thị thông báo “ Mã bạn nhập sai định dạng - Vui lòng kiểm tra lại”	Hoàn thành
3	Thêm chức vụ thất bại, Nhập trùng mã chức vụ	1	Hệ thống hiển thị thông báo “ Mã không được phép trùng!”	Hoàn thành

4	Thêm chức vụ thành công	1 , Thu ngân	Hệ thống hiển thị thông báo “ Thêm thành công”	Hoàn thành
5	Thêm chức vụ thành công và set vào combobox chức vụ bên form nhân viên		set chức vụ mới trên combobox	Hoàn thành

4.7 KIỂM THỬ FORM THÊM TÀI KHOẢN:

TH	Mô Tả Tình Huống	Dữ Liệu Mẫu	Kết Quả Mong Đợi	Fix
1	Thêm Tài khoản thất bại, bỏ trống mật khẩu		Hệ thống hiển thị thông báo “ Vui lòng nhập mật khẩu của bạn “	Hoàn thành
2	Thêm tài khoản thất bại , mật khẩu phải từ 8 ký tự trở lên	1234566	Hệ thống hiển thị thông báo “ Mật khẩu bao gồm 8 đến 18 ký tự”	Hoàn thành
3	Thêm Tài khoản thất bại, Xác nhận mật khẩu không đúng	Mật khẩu : 12345678 xác nhận mật khẩu ; 12345677	Hệ thống hiển thị thông báo “ Xác nhận mật khẩu chưa chính xác!”	Hoàn thành
4	Thêm tài khoản thành công	số điện thoại lấy bên nhân viên , mật khẩu : 12345678 xác nhận mật khẩu : 12345678	Hệ thống thông báo “ Successfully “	Hoàn thành

4.8 KIỂM THỬ FORM SẢN PHẨM:

TH	Mô Tả Tình Huống	Dữ Liệu Mẫu	Kết Quả Mong Đợi	Fix
1	Double vào Danh Mục Sản Phẩm		Hệ thống hiển thị danh sách sản phẩm đang có	Hoàn thành
2	Double click vào sản phẩm		Hệ thống hiển thị thông tin chi tiết sản phẩm qua bảng bên bên trái	Hoàn thành
3	Cập nhật hoặc thêm sản phẩm thành công.	Tên SP:Giày đá bóng Mã SP: 6 Số Lượng: 100 Đơn Giá: 100000 Đơn vị tính: Trái Loại: Bóng Trạng thái: Đang kinh doanh	Thông báo thêm thành công và load dữ liệu sản phẩm vừa thêm mới vào bảng.	Hoàn thành
4	Cập nhật hoặc thêm thất bại, nhập mã sản phẩm bị trùng	Giày bóng đá mã SP: 6 Áo polo mã SP: 6	Thông báo Mã sản phẩm đã tồn tại!	Hoàn thành
5	Cập nhật hoặc thêm thất bại, bỏ trống tên sản phẩm		Thông báo Vui lòng nhập tên sản phẩm!	Hoàn thành
6	Cập nhật hoặc thêm thất bại, bỏ trống mã sản phẩm		Thông báo Vui lòng nhập mã sản phẩm!	Hoàn thành
7	Cập nhật hoặc thêm thất bại, bỏ trống số lượng sản phẩm		Thông báo Vui lòng nhập số lượng!	Hoàn thành

8	Cập nhật hoặc thêm thất bại, bỏ trống đơn giá sản phẩm		Thông báo Vui lòng nhập đơn giá!	Hoàn thành
9	Cập nhật hoặc thêm thất bại, nhập chữ trong số lượng sản phẩm	abcdef	Thông báo Vui lòng nhập số!	Hoàn thành
10	Cập nhật hoặc thêm thất bại, nhập chữ trong số lượng sản phẩm	abcdef	Thông báo Vui lòng nhập số!	Hoàn thành
11	Xóa sản phẩm thành công, Click chọn sản phẩm trong bảng danh sách sản phẩm		Thông báo xóa thành công và load dữ liệu sản phẩm.	Hoàn thành
10	Reset form		Xóa trắng form khách hàng	Hoàn Thành
12	Trên thanh tìm kiếm nhập vào keyword cần tìm	Keyword: “Bóng” Hiển thị sản phẩm: - Bóng - Giày đá bóng - Túi đựng bóng	Hiển thị thông tin cần tìm trong bảng sản phẩm	Hoàn Thành

4.9 KIỂM THỬ FORM LOẠI SẢN PHẨM:

TH	Mô Tả Tình Huống	Dữ Liệu Mẫu	Kết Quả Mong Đợi	Fix
1	Double vào Danh Mục Loại Sản Phẩm		Hệ thống hiển thị danh sách loại sản phẩm đang có	Hoàn thành
2	Double click vào loại sản phẩm		Hệ thống hiển thị thông tin chi tiết loại sản	Hoàn thành

			phẩm qua bảng bên bên trái	
3	Cập nhật hoặc thêm loại sản phẩm thành công.	Mã Loại: 4 Tên Loại: Áo Ghi Chú:	Thông báo thêm thành công và load dữ liệu loại sản phẩm vừa thêm mới vào bảng.	Hoàn thành
4	Xóa loại sản phẩm thành công, Click chọn loại sản phẩm trong bảng danh sách sản phẩm (Loại chưa có thêm thông tin sản phẩm)		Thông báo xóa thành công và load dữ liệu loại sản phẩm.	Hoàn thành
5	Xóa loại sản phẩm không thành công, Click chọn loại sản phẩm trong bảng danh sách sản phẩm (Loại đã thêm thông tin sản phẩm)		Thông báo loại sản phẩm đang được sử dụng. Không được xóa!	Hoàn thành
6	Reset form		Xóa trắng form khách hàng	Hoàn Thành
7	Trên thanh tìm kiếm nhập vào keyword cần tìm	Keyword: “Á” Hiển thị loại sản phẩm: - Áo - Giày đá bóng	Hiển thị thông tin cần tìm trong bảng loại sản phẩm	Hoàn Thành

4.10 KIỂM THỬ FORM ĐƠN VỊ TÍNH:

TH	Mô Tả Tình Huống	Dữ Liệu Mẫu	Kết Quả Mong Đợi	Fix
1	Double vào Danh Mục Đơn Vị Tính		Hệ thống hiển thị danh sách đơn vị tính đang có	Hoàn thành

2	Double click vào đơn vị tính		Hệ thống hiển thị thông tin chi tiết đơn vị tính qua bảng bên bên trái	Hoàn thành
3	Cập nhật hoặc thêm đơn vị tính thành công.	Mã đơn vị: 4 Tên đơn vị: Đôi Ghi Chú:	Thông báo thêm thành công và load dữ liệu đơn vị tính vừa thêm mới vào bảng.	Hoàn thành
4	Xóa đơn vị tính thành công, Click chọn đơn vị tính trong bảng danh sách đơn vị tính (đơn vị tính chưa có thêm thông tin sản phẩm)		Thông báo xóa thành công và load dữ liệu đơn vị tính.	Hoàn thành
5	Xóa đơn vị tính không thành công, Click chọn đơn vị tính trong bảng danh sách đơn vị tính (đơn vị tính đã thêm thông tin sản phẩm)		Thông báo đơn vị đang được sử dụng. Không được xóa!	Hoàn thành
6	Reset form		Xóa trắng form khách hàng	Hoàn Thành
7	Trên thanh tìm kiếm nhập vào keyword cần tìm	Keyword: “Đôi” Hiển thị đơn vị tính: - Đôi	Hiển thị thông tin cần tìm trong bảng đơn vị tính	Hoàn Thành

4.11 KIỂM THỬ FORM KHÁCH HÀNG:

TH	Mô Tả Tình Huống	Dữ Liệu Mẫu	Kết Quả Mong Đợi	Fix
1	Double click vào danh mục khách hàng.		Đổ dữ liệu khách hàng vào bảng khách hàng.	Hoàn Thành

2	Cập nhật hoặc thêm khách hàng thành công.	Nguyễn Văn A, Số điện thoại hợp lệ.	Thông báo thêm thành công và load dữ liệu khách hàng vừa thêm mới vào bảng.	Hoàn Thành
3	Cập nhật hoặc thêm thất bại, nhập số điện thoại không hợp lệ nhiều hơn hoặc ít hơn 10 ký tự số.	098732628638 21, 091283721, . . .	Thông báo nhập sai và bắt nhập lại	Hoàn Thành
4	Cập nhật hoặc thêm thất bại, nhập số điện thoại không hợp lệ nhập bao gồm chữ cái và ký tự đặc biệt.	09327hf873, 0932f2@43e,	Thông báo nhập sai và bắt nhập lại	Hoàn Thành
5	Cập nhật hoặc thêm thất bại, bỏ trống tên khách hàng.		Thông báo nhập sai và bắt nhập lại	Hoàn Thành
6	Cập nhật hoặc thêm thất bại, bỏ trống số điện thoại khách hàng.		Thông báo nhập sai và bắt nhập lại	Hoàn Thành
7	Cập nhật hoặc thêm thất bại, bỏ trống tên và số điện thoại khách hàng.		Thông báo nhập sai và bắt nhập lại	Hoàn Thành
8	Xóa khách hàng thành công, Click chọn khách hàng trong bảng khách hàng (Khách hàng chưa lập hóa đơn lần nào).		Thông báo xóa thành công và load dữ liệu khách hàng.	Hoàn Thành
8	Xóa khách hàng thất bại, Click chọn khách hàng trong bảng khách hàng (Khách hàng đã lập hóa đơn).		Thông báo xóa thất bại.	Hoàn Thành

9	Reset form		Xóa trắng form khách hàng	Hoàn Thành
10	Trên thanh tìm kiếm nhập vào keyword cần tìm	Keyword: “Nguyễn” Hiện thị tên khách hàng: - Nguyễn Hoàng Anh - Nguyễn Việt Khái	Hiện thị thông tin cần tìm trong bảng Khách hàng	Hoàn Thành

4.12 KIỂM THỬ FORM LẬP HOA ĐƠN:

TH	Mô Tả Tình Huống	Dữ Liệu Mẫu	Kết Quả Mong Đợi	Fix
1	Double click chuột vào danh mục lập hóa đơn.		Đổ dữ liệu khách hàng vào bảng khách hàng, đổ dữ liệu sản phẩm vào bảng sản phẩm.	Hoàn Thành
2	Double click chuột vào bảng khách hàng.		Thông tin khách hàng bao gồm tên, mã và số điện thoại khách hàng hiển thị lên form Hóa Đơn Bán Hàng.	Hoàn Thành
3	Double click chuột vào Bảng Sản Phẩm.		Thông tin sản phẩm bao gồm mã, tên, số lượng, đơn giá và tính thành tiền của sản phẩm lên bảng Sản Phẩm chờ thanh toán.	Hoàn Thành

4	Double click chuột vào Bảng Sản Phẩm chờ thanh toán.		Hiển thị số lượng hiện tại của sản phẩm trong bảng sản phẩm chờ thanh toán lên Text Field số lượng.	Hoàn Thành
5	Nhập số lượng vượt quá số lượng trong kho. vd: Số lượng sản phẩm a hiện tại còn 50.	51, 52, 53, 54, . . .	Hiển thị thông báo nhập sai và bắt nhập lại.	Hoàn Thành
6	Nhập số lượng bé hơn 0.	-1, -2, -3, -4, . . .	Hiển thị thông báo nhập sai và bắt nhập lại.	Hoàn Thành
7	Nhập số lượng là số thực.	0.1, 1.1, 1.2, . . .	Hiển thị thông báo nhập sai và bắt nhập lại.	Hoàn Thành
8	Nhập số lượng là các kí tự không phải là số.	a, b, c, . . .	Hiển thị thông báo nhập sai và bắt nhập lại.	Hoàn Thành
9	Nhập phần trăm giảm giá cho toàn hóa đơn bé hơn 0 và lớn hơn 100.	-1, -2, -3, . . . 101, 102, 103, . .	Hiển thị thông báo nhập sai và bắt nhập lại.	Hoàn Thành
10	Nhập phần trăm giảm giá cho toàn hóa là số thực.	0.1, 1.1, 1.2, . . .	Hiển thị thông báo nhập sai và bắt nhập lại.	Hoàn Thành
11	Nhập phần trăm giảm giá cho toàn hóa là các kí tự không phải là số.	a, b, c, . . .	Hiển thị thông báo nhập sai và bắt nhập lại.	Hoàn Thành
12	Click vào Button Ghi.		Lưu số lượng trên Text Field cho sản phẩm trong bảng Sản Phẩm chờ thanh toán.	Hoàn Thành

13	Click vào Button Thanh Toán.		Lưu Hóa Đơn và Hóa Đơn Chi Tiết vào Database. Thông báo thêm hóa đơn thành công.	Hoàn Thành
14	Click vào Button In Hóa Đơn.		Chuyển hướng mở file hóa đơn để tiến hành xem trước khi in.	Hoàn Thành
15	Nhập tên hoặc mã sản phẩm vào Textfield tìm sản phẩm.		Tìm được sản phẩm để sản phẩm tìm được vào bảng sản phẩm.	Hoàn Thành
15	Nhập tên hoặc số điện thoại khách hàng vào Textfield tìm khách hàng.		Tìm được khách hàng để khách hàng tìm được vào bảng khách hàng.	Hoàn Thành
16	Reset form		Xóa trắng form hóa đơn.	Hoàn Thành
17	Trên thanh tìm kiếm khách hàng nhập vào keyword cần tìm	Keyword: "Nguyễn" Hiện thị tên khách hàng: - Nguyễn Hoàng Anh - Nguyễn Việt Khải	Hiện thị thông tin cần tìm trong bảng Khách hàng bên tay trái	Hoàn Thành

18	Trên thanh tìm kiếm sản phẩm nhập vào keyword cần tìm	Keyword: “Bóng” Hiển thị sản phẩm: - Bóng - Giày đá bóng - Túi đựng bóng	Hiển thị thông tin cần tìm trong bảng sản phẩm nằm giữa bảng khách hàng và hóa đơn bán hàng	Hoàn Thành
----	---	--	---	------------

4.13 KIỂM THỬ FORM XEM LẠI HÓA ĐƠN:

TH	Mô Tả Tình Huống	Dữ Liệu Mẫu	Kết Quả Mong Đợi	Fix
1	Double click vào danh mục Xem Lại Hóa Đơn.		Đổ thông tin hóa đơn bao gồm mã HD, tên khách hàng và ngày lập hóa đơn vào bảng Hóa đơn đã thanh toán.	Hoàn Thành
2	Double click vào bảng Hóa đơn đã thanh toán.		Hiển thị thông tin hóa đơn và hóa đơn chi tiết của hóa đơn vừa click lên form.	Hoàn Thành
3	Click vào Button Xóa hóa đơn.	Đăng nhập là Admin hoặc quản lý.	Hỏi lựa chọn chắc chắn xóa. - Yes: Xóa hóa đơn. - No: Hoàn tác thao tác xóa	Hoàn Thành
4	Click vào Button Xóa hóa đơn.	Đăng nhập là thu ngân.	Thông báo không cho xóa hóa đơn.	Hoàn Thành

5	Click vào Button In Hóa Đơn.		Chuyển hướng mở file hóa đơn để tiến hành xem trước khi in.	Hoàn Thành
6	Reset form		Xóa trắng form	Hoàn Thành

4.14 KIỂM THỬ FORM DOANH THU:

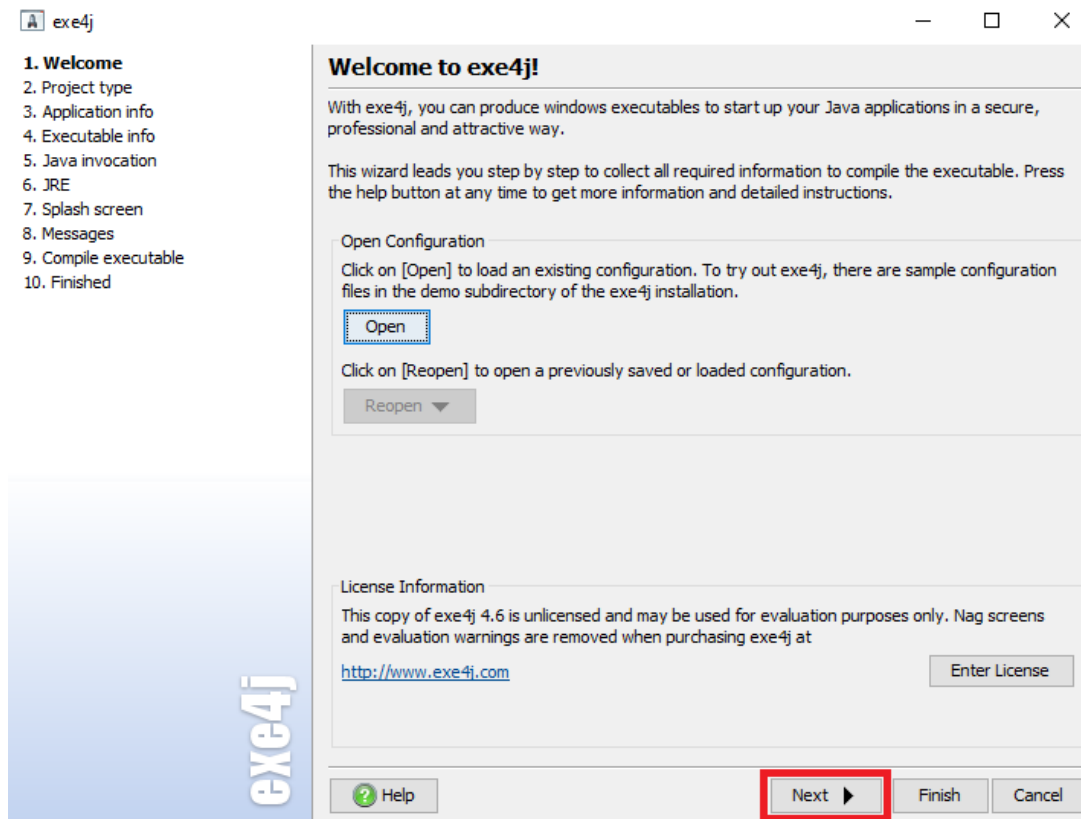
TH	Mô Tả Tình Huống	Dữ Liệu Mẫu	Kết Quả Mong Đợi	Fix
1	Nhập ngày bắt đầu và ngày kết thúc đúng định dạng	01-01-2021	Hiển thị doanh thu	Hoàn thành
2	Nhập ngày bắt đầu và ngày kết thúc sai định dạng	sad, jahuy, 13-14-2021, . . .	Thông báo nhập sai và yêu cầu nhập lại	Hoàn thành
3	Bỏ trống không nhập ngày bắt đầu hoặc ngày kết thúc		Thông báo yêu cầu nhập lại.	Hoàn thành

5 ĐÓNG GÓI VÀ TRIỂN KHAI

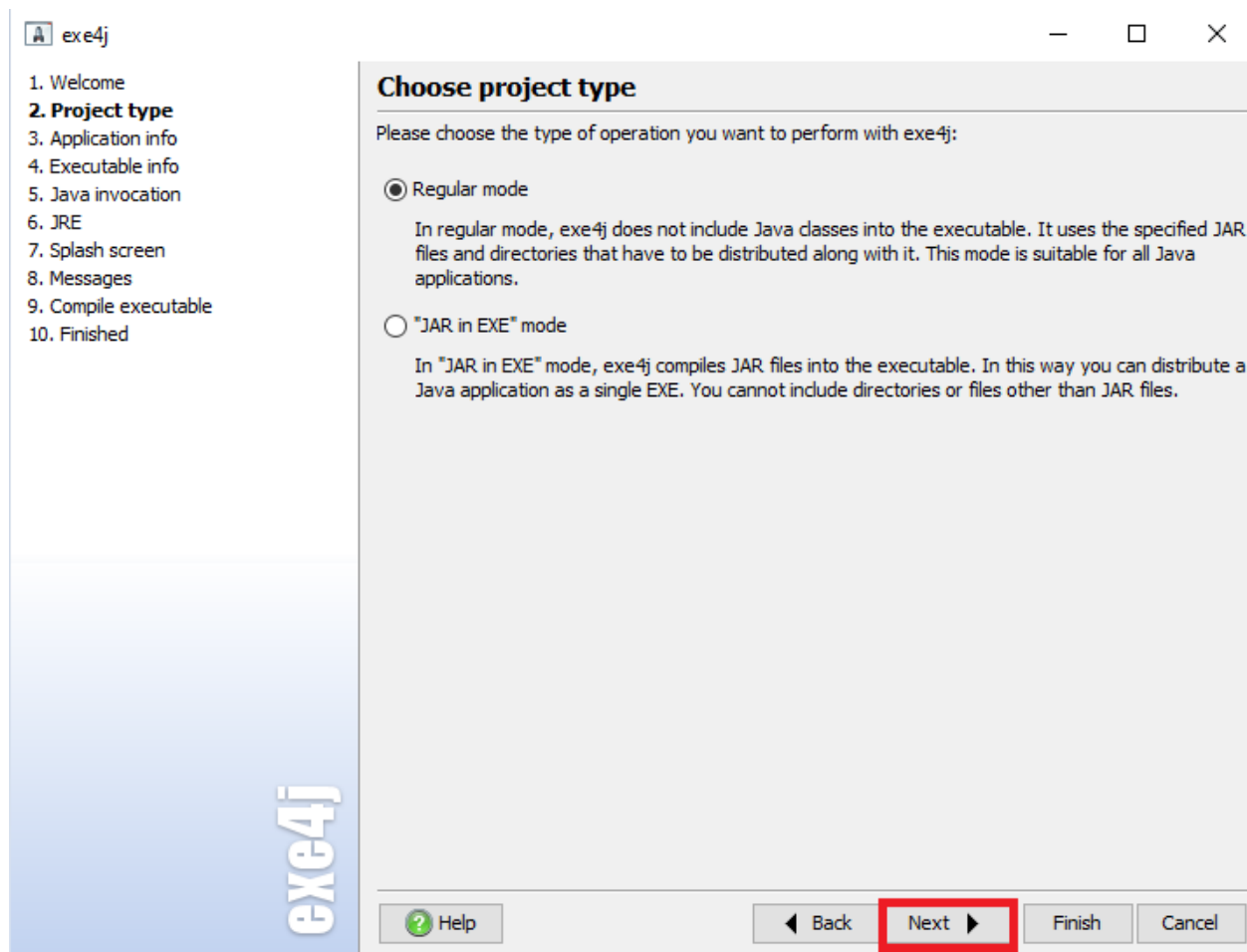
5.1 HƯỚNG DẪN CHUYỂN ĐỔI JAR THÀNH EXE

B1: Mở chương trình exe4j Wizard:

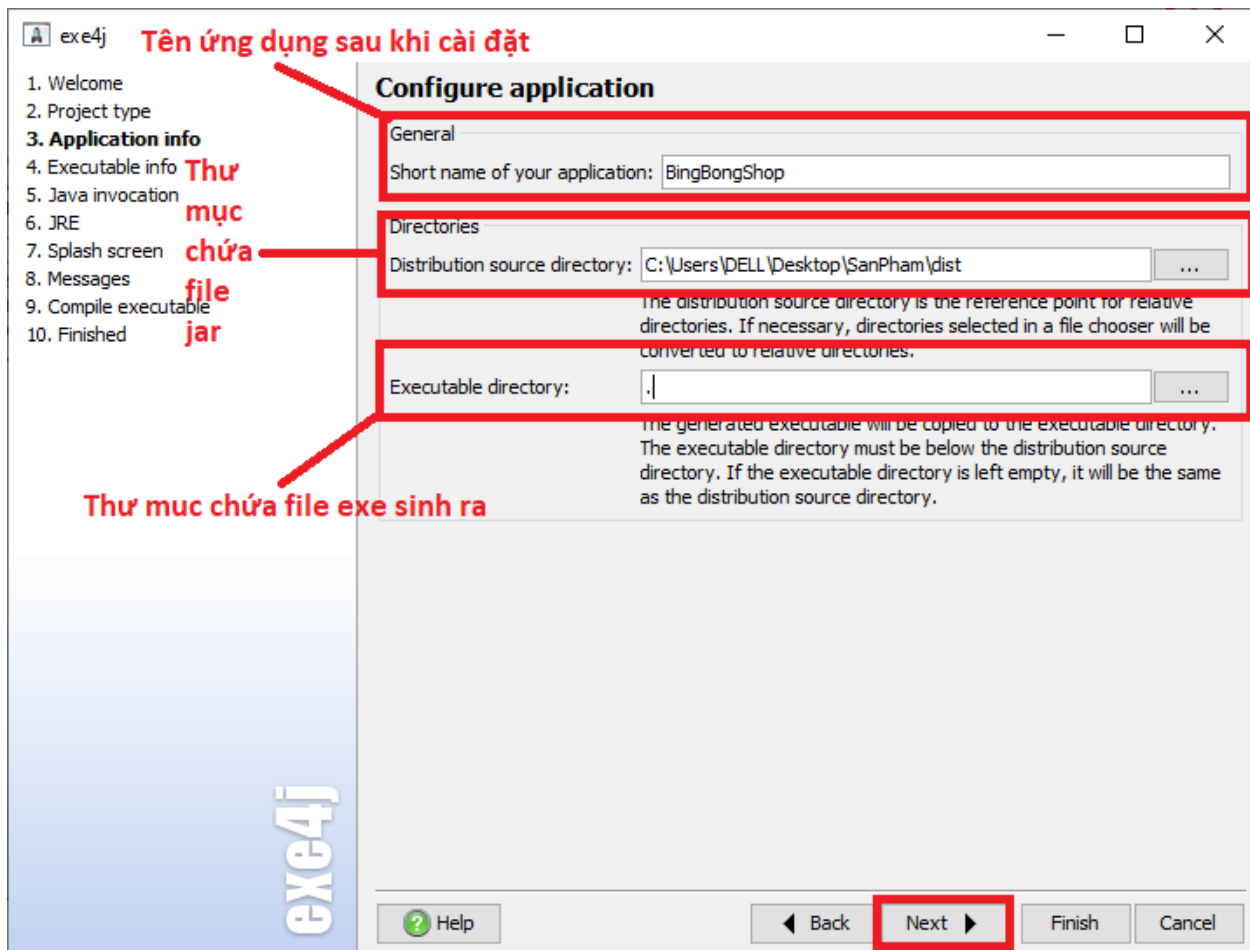
B2: Trong hộp thoại Welcome to exe4j! → Chọn Next.



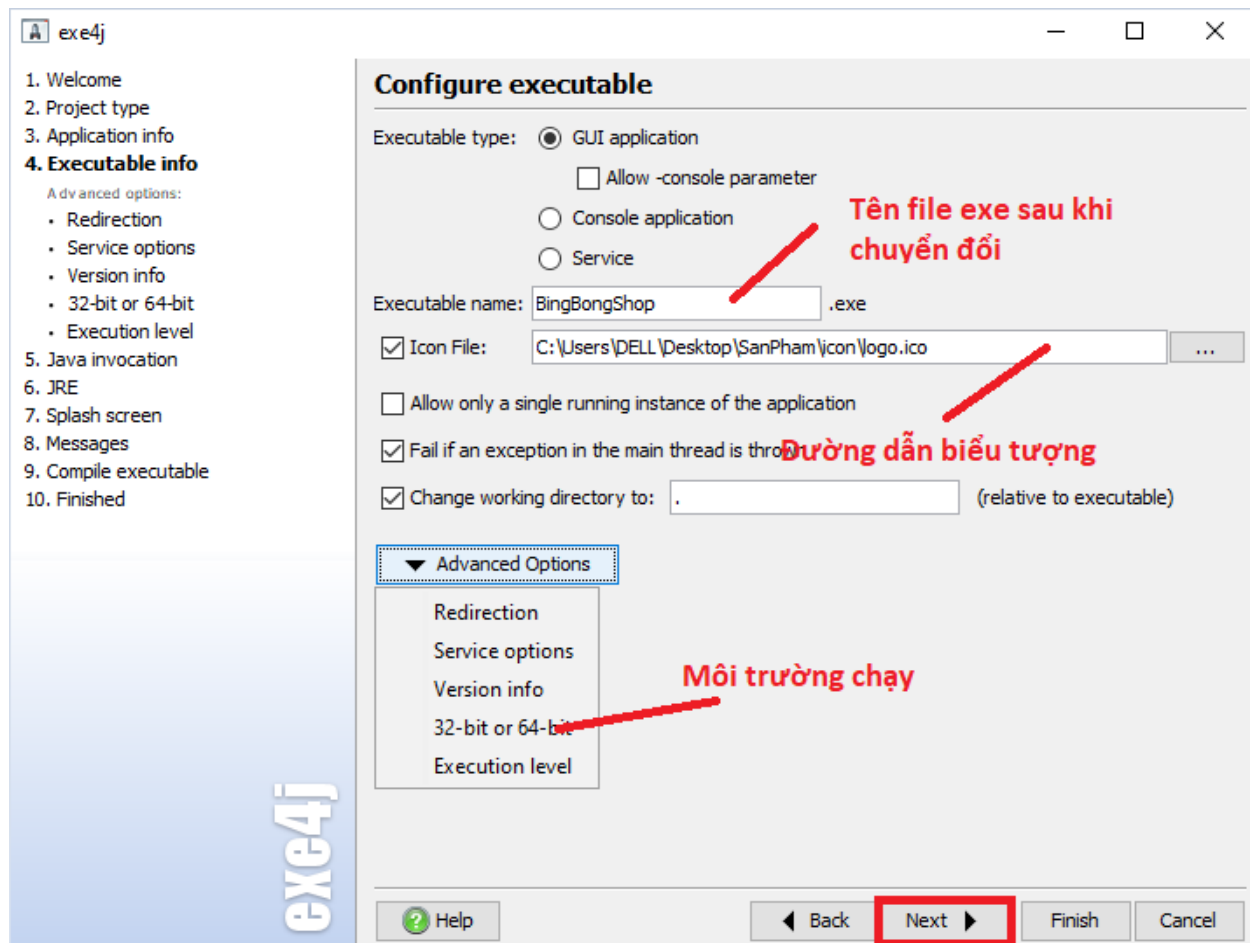
B3: Trong hộp thoại Choose project type → Chọn Next.



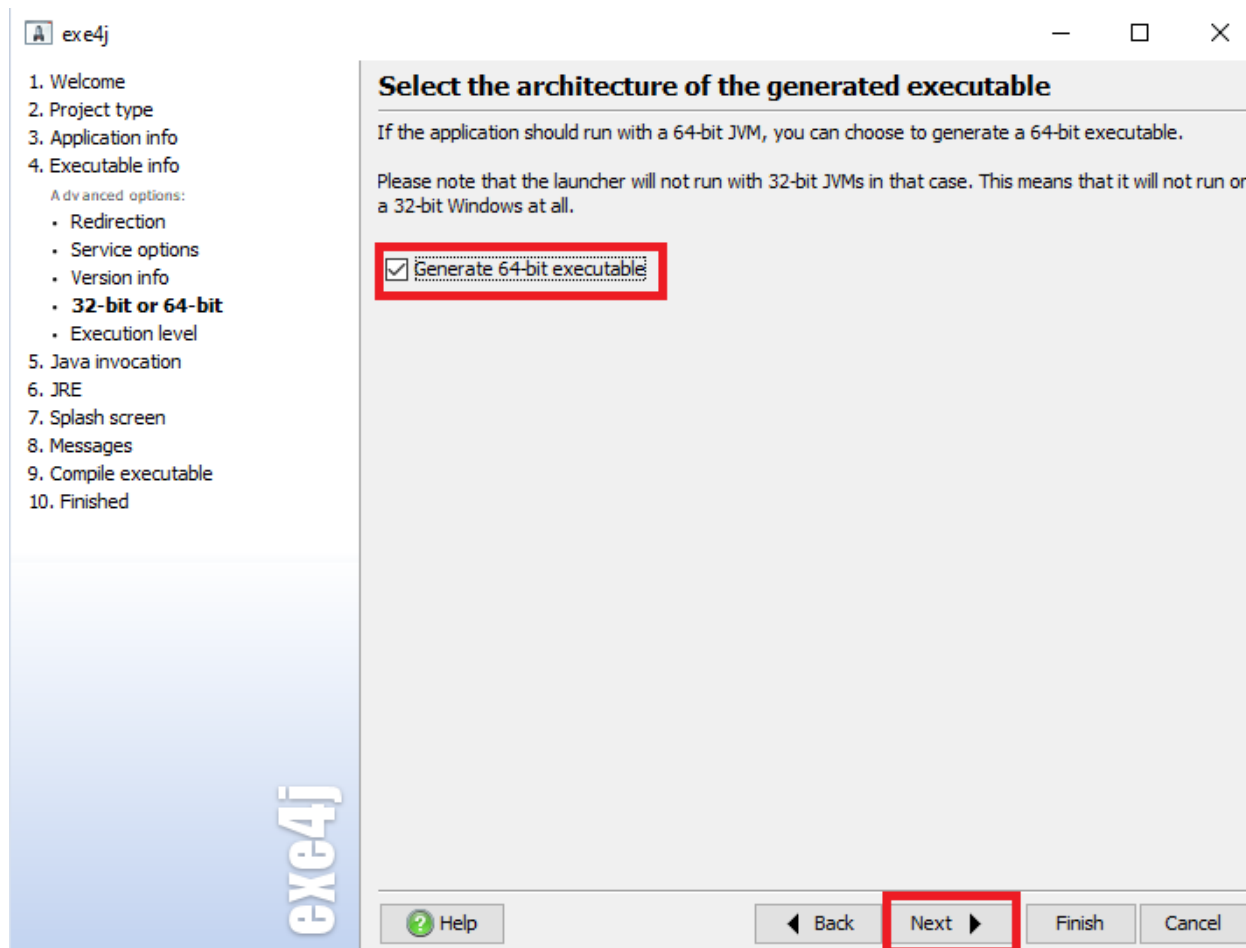
B4: Trong hộp thoại Configure application → Nhập tên ứng dụng sau khi cài đặt → Chọn thư mục chứa file jar → Chọn thư mục chứa file exe sinh ra → Chọn Next.



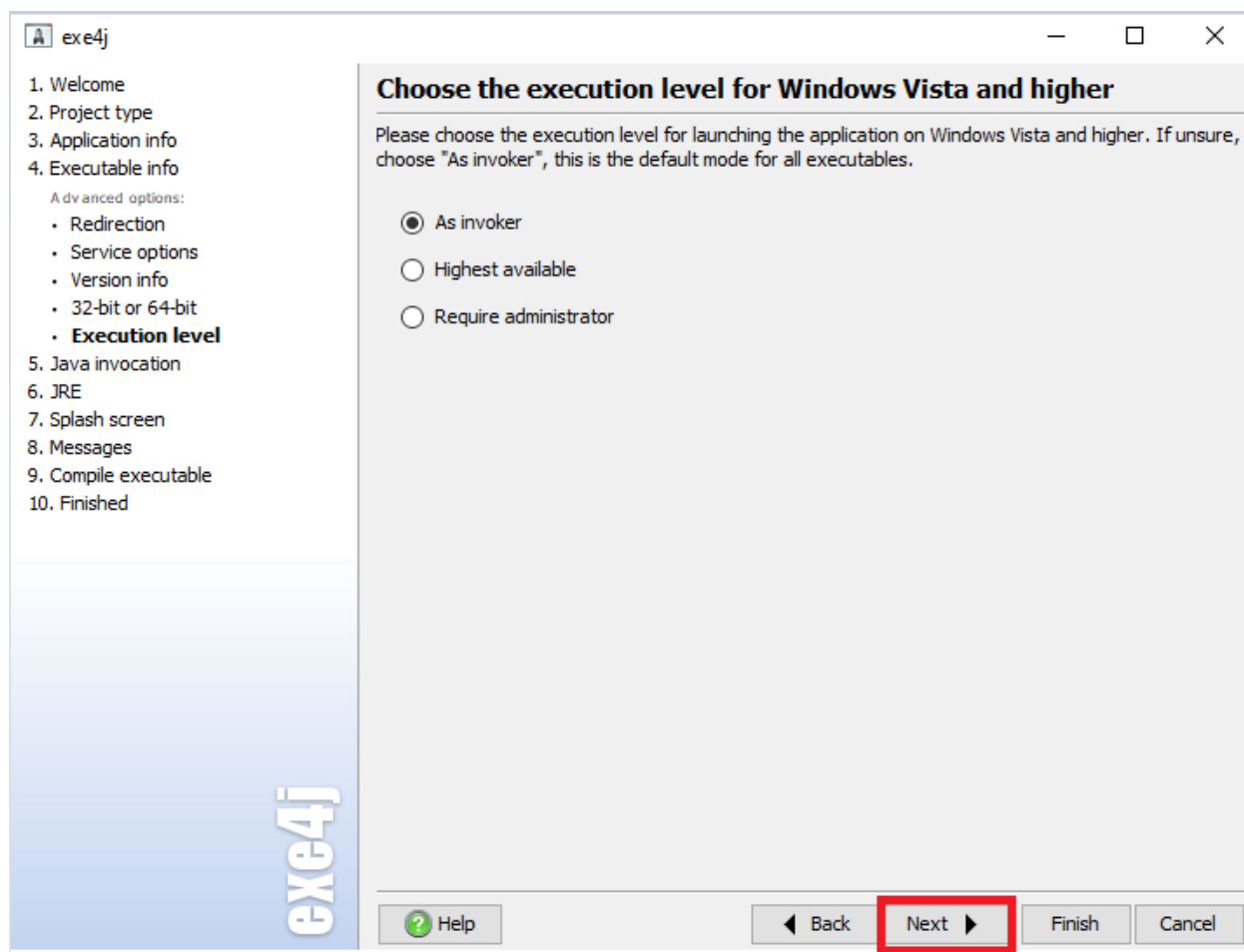
B5: Trong hộp thoại Configure executable → Nhập tên file sau khi chuyển đổi → Chọn đường dẫn biểu tượng (nếu có) → Chọn môi trường chạy → Chọn Next.



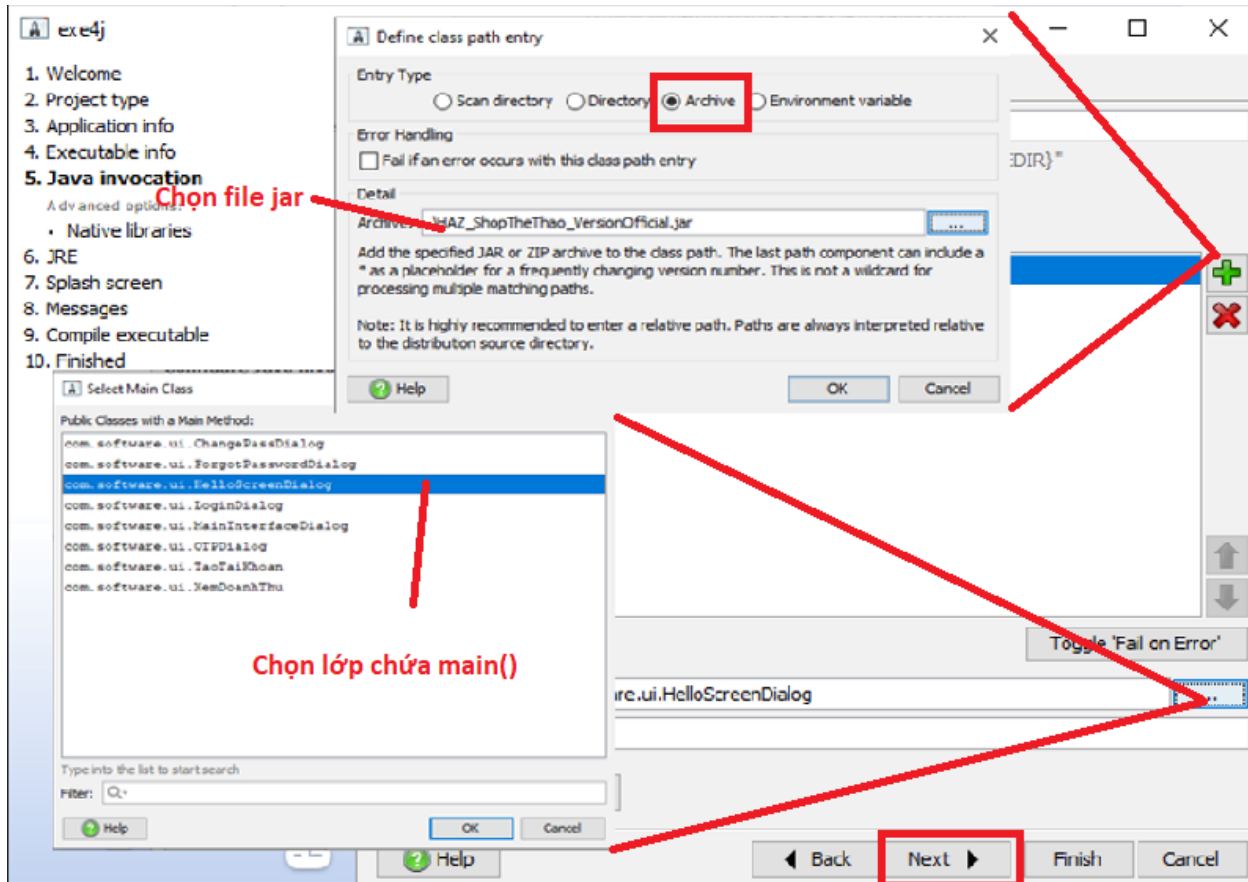
B6: Trong hộp thoại Select the architecture of the generated executable → Chọn Generate 64-bit executable → Chọn Next.



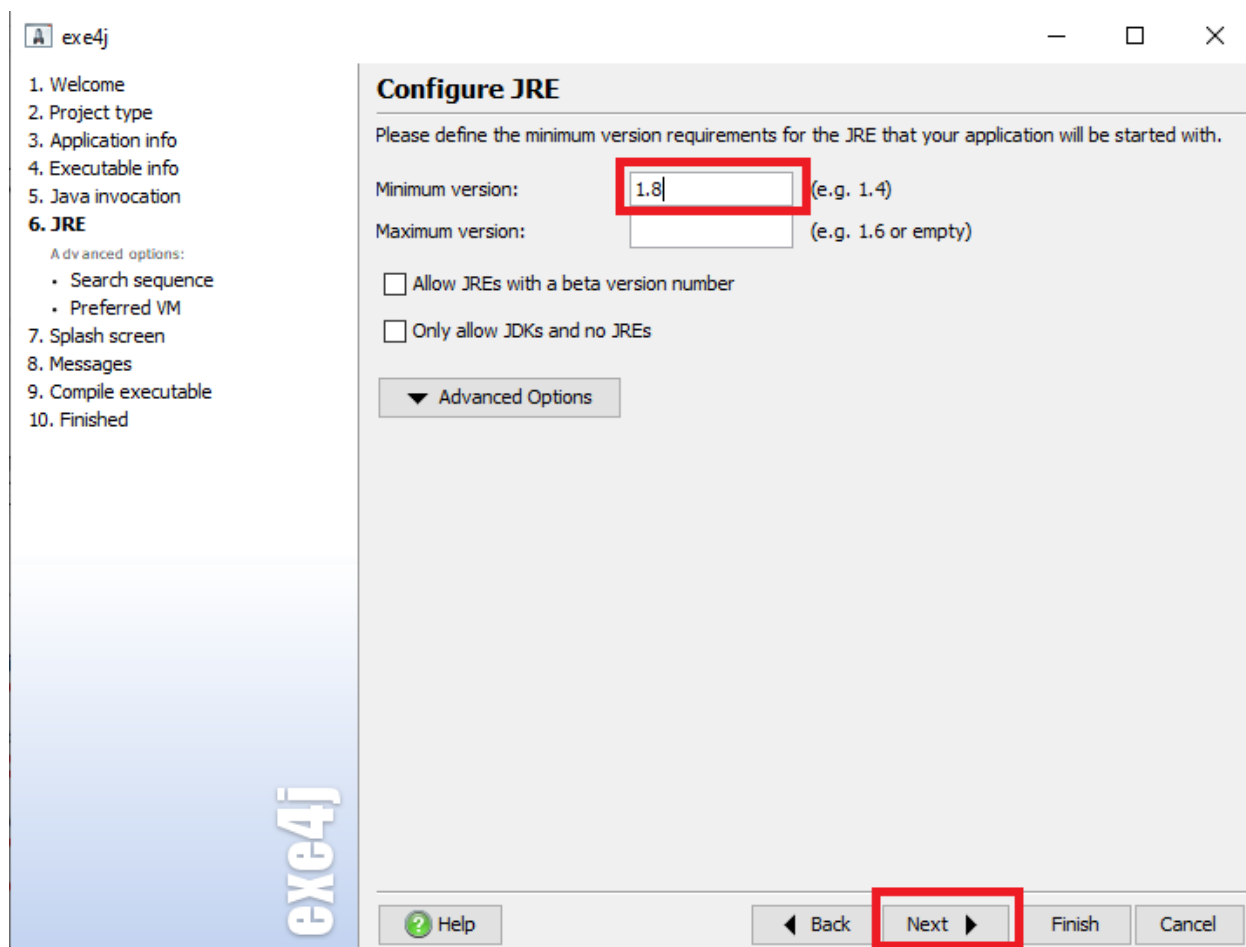
B7: Trong hộp thoại Choose the execution level for Windows Vista and higher → Chọn Next.



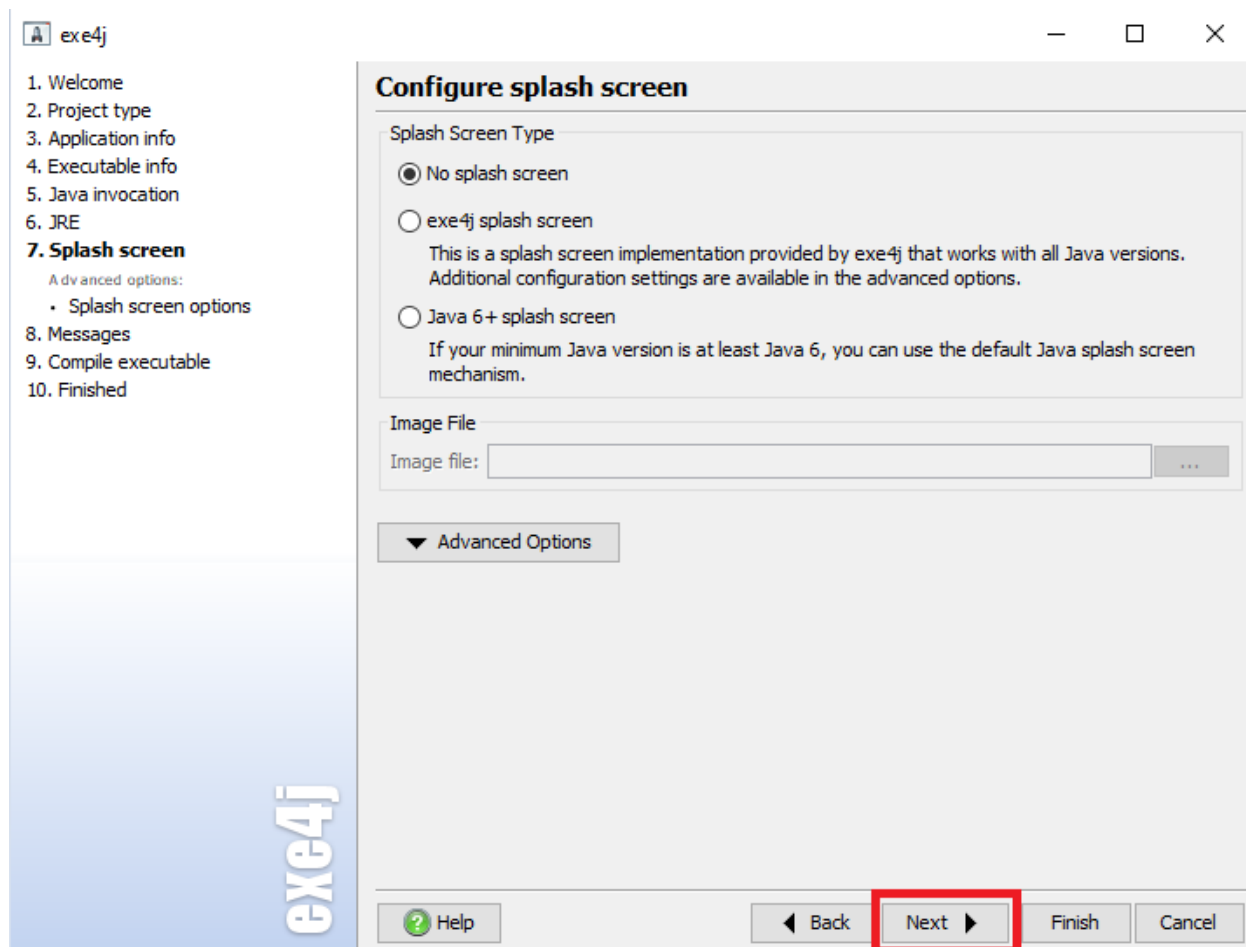
B8: Trong hộp thoại Configure Java invocation → Chọn Class path (đầu cộng màu xanh dưới mô tả). → Chọn Archivve → Chọn file jar → Chọn OK → Trong hộp thoại Configure Java invocation tiếp tục chọn Main class → Trong hộp thoại Select Main Class → Chọn main class → Chọn OK → Trở về hộp thoại Configure Java invocation chọn Next.



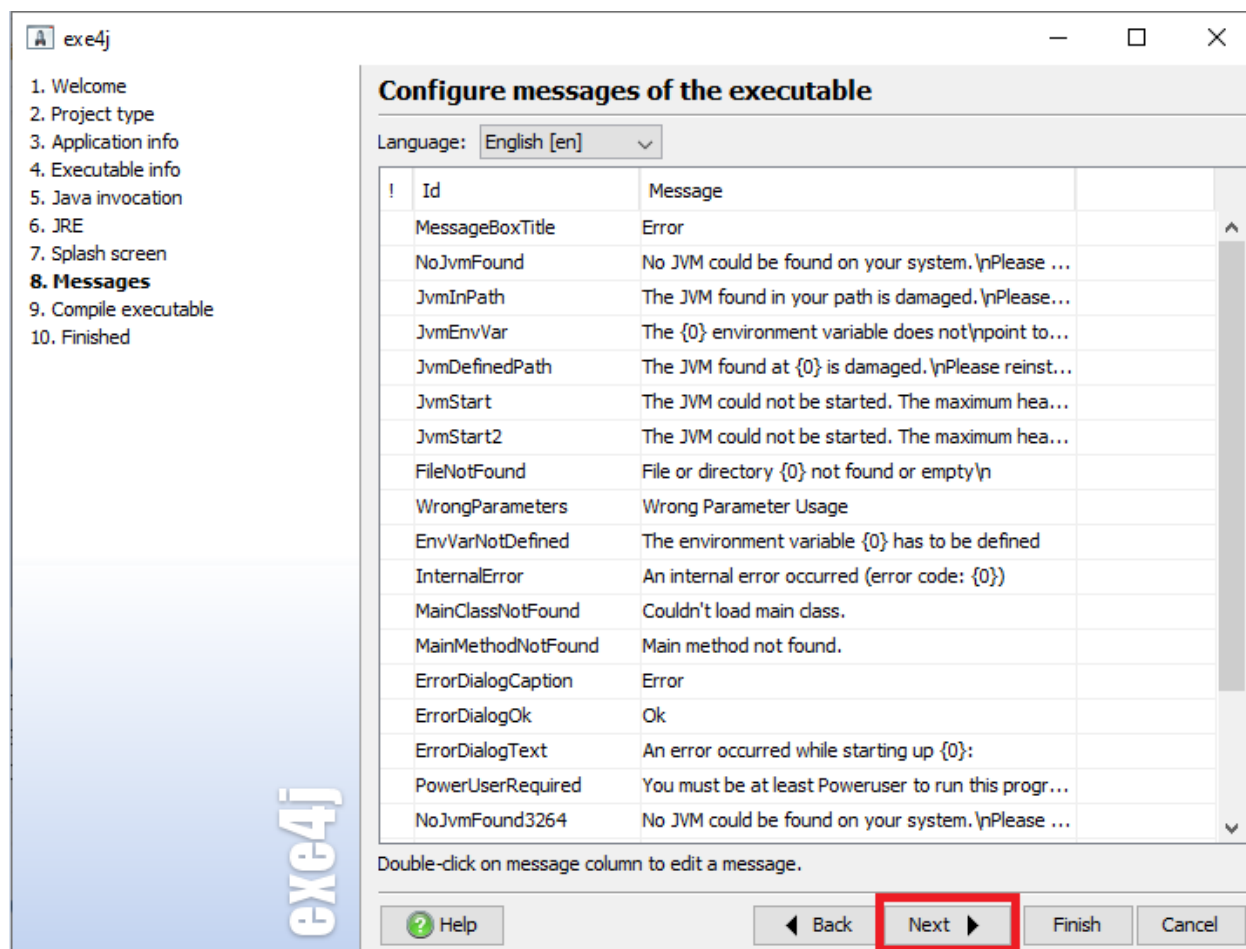
B9: Trong hộp thoại Configure JRE → Nhập Minimum version → Chọn Next.



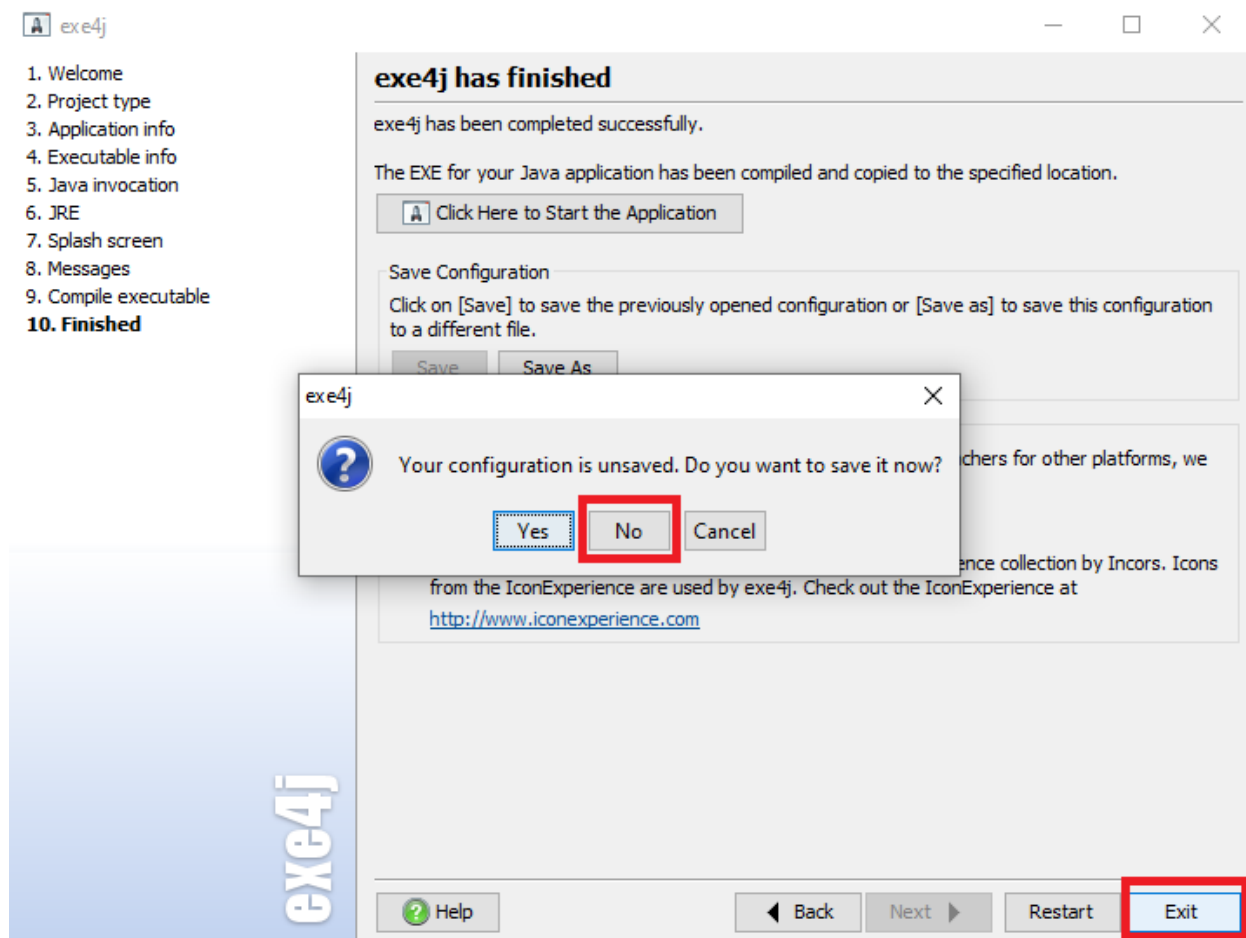
B10: Trong hộp thoại Configure splash screen → Chọn Next.



B11: Trong hộp thoại Configure messages of the executable → Chọn Next.



B12: Trong hộp thoại exe4j has finished → Chọn Exit → Chọn No.



5.2 HƯỚNG DẪN CÀI ĐẶT TRIỂN KHAI

- Bước 1: Chạy data.sql để tạo CSDL QuanLyShopTheThaoPingPong.
- Bước 2: Đặt mật khẩu sa của SQL Server là 123.
- Bước 3: Chạy file Setup.exe mật khẩu cài đặt là admin@123 (chú ý môi trường JDK1.8+).
- Bước 4: Sau khi cài đặt mở ứng dụng lên đăng nhập với số điện thoại là 0978215902 và mật khẩu là admin@123.

5.3 HƯỚNG DẪN SỬ DỤNG PHẦN MỀM

B1: Click đúp chuột vào phần mềm để khởi chạy chương trình.

B2: Chờ chương trình load dữ liệu và tiến hành đăng nhập.

B3: Nhập Số Điện Thoại là 0978215902, Mật Khẩu là admin@123 để đăng nhập.

B4: Click vào nút “Yes” để tiến hành vào giao diện chính.

B5: Chọn chức năng ở giao diện chính:

- Danh Mục Nhân Viên:
 - Thêm Nhân Viên:

B1: Nhập Mã nhân viên : (NV01)

Nhập Tên nhân viên : (Bui Trong Nhan)

Chọn giới tính (Nam)

Chọn chức vụ (tùy chọn)

Nhập số điện thoại (0769331203)

Nhập Email : Trongnhan140902@gmail.com

Nhập CMND : 364236567

Nhập địa chỉ : Tp . Cần thơ

Ảnh nhấp vào chọn ảnh mình mong muốn .

B2: Nhấn vào button thêm nhân viên .

B3: Nhân viên mới được thêm vào danh sách nhân viên.

- Sửa Nhân Viên:

B1 : Chọn một nhân viên muốn cập nhật bằng cách click vào một dòng trên bảng.

B2 : Chọn và sửa thông tin mong muốn (vd : sửa tên : Trọng Nhân).

B3 : Nhấn vào button sửa nhân viên.

B4 : Nhân viên được cập nhật và hiển thị trên danh sách.

- Xóa Nhân Viên:

B1: Chọn một nhân viên muốn Xóa bằng cách click vào một dòng trên bảng (trừ nhân viên có vai trò là admin).

B2 : Nhấn vào button Xóa nhân viên.

B3 : Nhân viên đã được xóa và không còn hiển thị trên danh sách.

- Thêm tài khoản nhân viên:

B1: Chọn một nhân viên muốn Thêm tài khoản bằng cách click vào một dòng trên bảng (ngoại trừ nhân viên đã có tài khoản)

B2 : Nhấn vào button tạo tài khoản

B3 : chuyển ta đến một jframe mới để tạo tài khoản

B4 : Nhập mật khẩu phải trên 8 ký tự (12345678)

B5 :Điền xác nhận mật khẩu y như trên (12345678)

B6 : Nhấn button hoàn tất để tạo tài khoản

- Tìm kiếm nhân viên:

B1 : Nhập từ khóa của tên nhân viên vào txtTimKiemNV.

B2 : Nhân viên tương tự từ khóa trên sẽ hiển thị lên bảng .

- Danh Mục Chức Vụ:

- Thêm Chức Vụ:

B1 : Nhập mã chức vụ (1) - phải là số nguyên

B2 : Nhập tên chức vụ (Thu Ngân)

B3 : Nhập mô tả chức vụ mới có

B4 : Nhấn button thêm chức vụ

B5 : Chức vụ vừa được thêm hiển thị lên bảng

- Sửa Chức Vụ:

B1 : Chọn một chức vụ muốn cập nhật . (Tránh các chức vụ đã có nhân viên sử dụng)

B2 : Thay đổi thông tin theo ý muốn

B3 : Nhấn nút sửa chức vụ

B4 : Chức vụ được sửa hiển thị lên bảng chức vụ

- Xóa Chức Vụ:

B1 : Chọn một chức vụ muốn xóa .

B2 : Nhấn nút xóa chức vụ (Tránh các chức vụ đã có nhân viên sử dụng)

B4 : Chức vụ được xóa không còn hiển thị lên bảng chức vụ.

- Tìm kiếm Chức Vụ:

B1 : Nhập từ khóa của tên chức vụ vào thanh tìm kiếm phía trên cùng của bảng.

B2 : chức vụ tương tự từ khóa trên sẽ hiển thị lên bảng .

- Danh mục Sản Phẩm:

- Thêm Sản Phẩm:

B1: Nhập tên sản phẩm mới.

B2: Nhập mã sản phẩm mới (Không được trùng)

B3: Nhập vào số lượng sản phẩm.

B4: Nhập vào đơn giá sản phẩm.

B5: Chọn đơn vị tính.

B6: Chọn loại sản phẩm.

B7: Chọn trạng thái

B8: Chọn thêm sản phẩm.

B9: Sản phẩm đã được thêm vào bảng và hiển thị trên bảng sản phẩm.

Sửa Sản Phẩm:

B1: Double click chọn sản phẩm muốn sửa trong bảng sản phẩm.

B2: Thay đổi nội dung muốn sửa. Riêng mã sản phẩm không được sửa.

B3: Nhấn nút sửa sản phẩm

B4: Sản phẩm đã được thay đổi trong bảng sản phẩm.

○ Xóa Sản Phẩm:

B1: Double click chọn sản phẩm muốn xóa trong bảng sản phẩm.

B2: Nhấn nút xóa sản phẩm. Riêng những sản phẩm đã được lập hóa đơn không được xóa , chỉ thay đổi được trạng thái của sản phẩm.

B3: Sản phẩm đã xóa không còn hiện lên bảng sản phẩm.

○ Tìm kiếm Sản Phẩm:

B1: Nhập từ khóa của tên sản phẩm vào thanh tìm kiếm phía trên cùng của bảng.

B2: Các sản phẩm liên quan sẽ hiển thị lên bảng sản phẩm.

● Danh mục Loại Sản Phẩm:

○ Thêm Loại Sản Phẩm:

B1: Nhập vào mã loại sản phẩm (Mã loại không được trùng).

B2: Nhập vào tên loại sản phẩm.

B3: Nhập vào mô tả (có thể nhập hoặc không).

B4: Chọn thêm loại sản phẩm.

B5: Loại sản phẩm được thêm vào bảng và hiển thị trên bảng loại sản phẩm.

- Sửa Loại Sản Phẩm:
 - B1: Double click chọn loại sản phẩm muốn sửa trong bảng loại sản phẩm.
 - B2: Thay đổi nội dung muốn sửa. Riêng mã loại sản phẩm không được sửa.
 - B3: Nhấn nút sửa loại sản phẩm.
 - B4: Loại sản phẩm đã được thay đổi trong bảng loại sản phẩm.
- Xóa Loại Sản Phẩm:
 - B1: Double click chọn loại sản phẩm muốn xóa trong bảng loại sản phẩm.
 - B2: Nhấn nút xóa loại sản phẩm. Riêng những loại sản phẩm đã được áp dụng cho sản phẩm không được xóa.
 - B3: Loại sản phẩm đã xóa không còn hiện lên bảng sản phẩm.
- Tìm kiếm Loại Sản Phẩm:
 - B1: Nhập từ khóa của tên loại sản phẩm vào thanh tìm kiếm phía trên cùng của bảng.
 - B2: Các sản phẩm liên quan sẽ hiển thị lên bảng loại sản phẩm.
- Danh mục Đơn Vị Tính:
 - Thêm Đơn Vị Tính:
 - B1: Nhập vào mã đơn vị tính (Mã đơn vị tính không được trùng).
 - B2: Nhập vào tên đơn vị tính.
 - B3: Nhập vào mô tả (có thể nhập hoặc không).
 - B4: Chọn thêm đơn vị tính.
 - Sửa Đơn Vị Tính:
 - B1: Double click chọn đơn vị tính muốn sửa trong bảng đơn vị tính.
 - B2: Thay đổi nội dung muốn sửa. Riêng mã đơn vị tính không được sửa.

B3: Nhấn nút sửa đơn vị tính.

B4: Đơn vị tính đã được thay đổi trong bảng đơn vị tính.

- Xóa Đơn Vị Tính:

B1: Double click chọn đơn vị tính muốn xóa trong bảng đơn vị tính.

B2: Nhấn nút xóa đơn vị tính. Riêng những đơn vị tính đã được áp dụng cho sản phẩm không được xóa.

B3: Đơn vị tính đã xóa không còn hiện lên bảng đơn vị tính.

- Tìm kiếm Đơn Vị Tính:

B1: Nhập từ khóa của tên đơn vị tính vào thanh tìm kiếm phía trên cùng của bảng.

B2: Các đơn vị tính liên quan sẽ hiển thị lên bảng đơn vị tính.

- Danh mục Bán Hàng:

- Lập Hóa Đơn:

- Thêm Hóa Đơn:

B1: Chọn khách hàng trong bảng khách hàng.

B2: Chọn ít nhất 1 sản phẩm trong bảng sản phẩm.

B3: Chọn sản phẩm trong bảng sản phẩm chờ thanh toán để chỉnh sửa số lượng.

B4: Nhập vào phần trăm giảm giá cho toàn hóa đơn (nếu có), phần trăm giảm giá chỉ giới hạn từ 0 - 100 và không bao gồm số thực (vd: 0.1, 0.2, 0.3, . . .).

B5: Nhấn vào nút thanh toán để lưu hóa đơn.

B6: Hệ thống thông báo lưu thành công và hiển thị hộp thoại chọn in hóa đơn vừa lưu hay không, chọn Yes để tiến hành xem trước khi in hoặc No để tiến hành trạng thái lập hóa đơn mới.

- Tìm Khách Hàng: Nhập tên hoặc số điện thoại vào thanh tìm kiếm khách hàng để tìm khách hàng.
- Tìm Sản Phẩm: Nhập tên hoặc mã sản phẩm để tìm sản phẩm.
- In Hóa Đơn:

B1: Kiểm tra xem thông tin hóa đơn đã đầy đủ (Bao gồm thông tin khách hàng và sản phẩm).

B2: Chọn nút in hóa đơn để tiến hành xem trước khi in.

- Xem Lại Hóa Đơn

- Tìm Hóa Đơn: Nhập tên khách hàng , mã hóa đơn, hoặc số điện thoại khách hàng vào thanh tìm kiếm hóa đơn để tìm hóa đơn
- Xem Hóa Đơn: Click chọn hóa đơn trong bảng để xem hóa đơn đã chọn.
- Xóa Hóa Đơn (Chỉ thao tác được với tài khoản là admin hay quản lý):

B1: Click chọn hóa đơn trong bảng để tiến hành xóa HĐ.

B2: Click vào nút xóa.

B3: Trong hộp thoại lựa chọn, chọn Yes để xóa hóa đơn, chọn No để hoàn tác lệnh.

- In Hóa Đơn:

B1: Click chọn hóa đơn trong bảng để tiến hành.

B2: Chọn nút in hóa đơn để tiến hành xem trước khi in.

- Danh mục Khách Hàng:

- Thêm Khách Hàng:

B1: Nhập tên khách hàng, nhập số điện thoại (10 số và không bao gồm chữ cái và bắt đầu là số 0).

B2: Click chuột vào nút thêm.

B3: Click chuột vào nút Ok.

- Sửa Khách Hàng:

B1: Click chọn khách hàng trong bảng để sửa thông tin khách hàng.

B2: Nhập tên khách hàng, nhập số điện thoại (10 số và không bao gồm chữ cái và bắt đầu là số 0).

B3: Click chuột vào nút sửa.

B4: Click chuột vào nút Ok.

- Xóa Khách Hàng:

B1: Click chọn khách hàng trong bảng để tiến hành xóa thông tin khách hàng (Khách hàng chưa lập hóa đơn).

B2: Click chuột vào nút xóa.

B3: Chọn Yes để xóa hoặc chọn No để trở lại.

- Tìm kiếm Khách Hàng: Nhập tên hoặc số điện thoại vào thanh tìm kiếm khách hàng để tìm khách hàng.

- Đăng xuất: Trở về giao diện đăng nhập.
- Thoát: Đóng chương trình.
- Chức năng khác ngoài giao diện chính (Quên mật khẩu):

B1: Click chuột vào dòng Forgot password trong phần Login.

B2: Nhập Gmail đã đăng ký trước đó của tài khoản cần đổi mật khẩu rồi ấn Confirm.

B3: Mở Gmail lên lấy mã xác nhận vừa nhận trong Gmail sau đó nhập vào Form OTP rồi ấn Confirm.

B4: Tiến hành nhập mật khẩu mới và xác nhận mật khẩu mới rồi ấn Confirm. Mật khẩu đã đổi thành công!