

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH**



LẬP TRÌNH WEB

ĐỀ TÀI:

WEBSITE BÁN PHỤ KIỆN ĐIỆN THOẠI

Giảng viên hướng dẫn: ThS. NGUYỄN TRUNG HIẾU

Sinh viên thực hiện:

NGUYỄN THỊ HIỀN	N20DCCN099
NGUYỄN THỊ THU HIỀN	N20DCCN100
NGUYỄN HOÀI HÂN	N20DCCN098
NGUYỄN TRỌNG NHÂN	N20DCCN122
VÕ QUỐC DINH	N20DCCN091

Lớp: D20CQCN02-N

Hồ Chí Minh, tháng 06 năm 2023

Mục lục

I.	KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG	4
1.	Tên đề tài: website bán phụ kiện điện thoại	4
2.	Sử dụng:	4
3.	Công cụ sử dụng	4
4.	Phân quyền và chức năng	4
-	Khách hàng chưa có tài khoản:	4
-	Khách hàng	4
-	Nhân viên:	4
-	Admin	4
5.	Phân tích hệ thống:	4
II.	GIAO DIỆN	7
1.	Trang chủ:	7
2.	Chi tiết sản phẩm	8
3.	Trang đăng ký, đăng nhập	8
4.	Trang thông tin khách hàng	9
5.	Trang giỏ hàng	9
6.	Trang đặt hàng	10
7.	Trang địa chỉ giao hàng	10
8.	Trang thông báo đặt hàng	11
9.	Trang đăng ký thông tin khách hàng	11
10.	Quản lý sản phẩm	11
11.	Quản lý phiếu nhập	12
12.	Quản lý phiếu đặt	13
13.	Quản lý nhà cung cấp	14
14.	Quản lý nhân viên	15
15.	Quản lý thông tin nhân viên	15
16.	Quản lý danh mục sản phẩm	17
III.	TÍNH NĂNG	17
1.	Tính năng quản lý:	17

a. Đăng ký, đăng nhập:	17
b. Xem, cập nhật thông tin tài khoản:.....	22
c. Quản lý tài khoản(khóa tài khoản nhân viên-khoá tài khoản khách hàng)	24
d. Xem, thêm, xóa, sửa sản phẩm:	24
e. Xem, thêm, xóa, sửa doanh mục sản phẩm:	26
f. Xem, thêm, xóa, xác nhận, hoàn thành phiếu nhập- Xem, thêm chi tiết phiếu nhập	29
g. Duyệt, hủy phiếu đặt - xem chi chi tiết phiếu đặt	33
h. Xem, thêm, xóa, sửa nhà cung cấp	34
2. Tính năng của khách hàng	35
a. Cập nhật thông tin tài khoản.	35
b. Đăng ký tài khoản khách hàng	36
c. Đăng ký thông tin khách hàng.....	38
d. Tìm kiếm:	40
e. Lọc:.....	40
f. Trang thông tin chi tiết sản phẩm;	42
g. Cập nhật thông tin khách hàng	44
h. Tình trạng đơn hàng	45

I. KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG

1. **Tên đề tài:** website bán phụ kiện điện thoại

2. **Sử dụng:**

- Spring MVC, my SQL
- JPA(Java Persistence API)
- HTML, CSS, Bootstrap và một vài thư viện được thiết kế sẵn.

3. **Công cụ sử dụng**

- Microsoft my SQL 8.0.32
- Eclipse IDE.

4. **Phân quyền và chức năng**

- Có 4 nhóm phân quyền: , khách hàng(chưa có tài khoản), khách hàng, admin, nhân viên.

- **Khách hàng chưa có tài khoản:**

- ✓ Tìm kiếm, lọc
- ✓ Xem thông tin sản phẩm
- ✓ Đăng kí tài khoản
- **Khách hàng**
- ✓ Có các quyền như khách hàng chưa có tài khoản
- ✓ Thông tin cá nhân như: giới tính, địa chỉ, sđt, ngày sinh
- ✓ Thêm sản phẩm vào giỏ hàng
- ✓ Đặt hàng
- ✓ Chỉnh sửa thông tin cá nhân

- **Nhân viên:**

- ✓ Chỉnh sửa thông tin cá nhân
- ✓ Quản lý sản phẩm
- ✓ Quản lý doanh mục sản phẩm
- ✓ Quản lý phiếu nhập
- ✓ Quản lý phiếu đặt
- ✓ Quản lý nhà cung cấp

- **Admin**

- ✓ Có các quyền của nhân viên
- ✓ Quản lí tài khoản(tài khoản nhân viên – khách hàng)

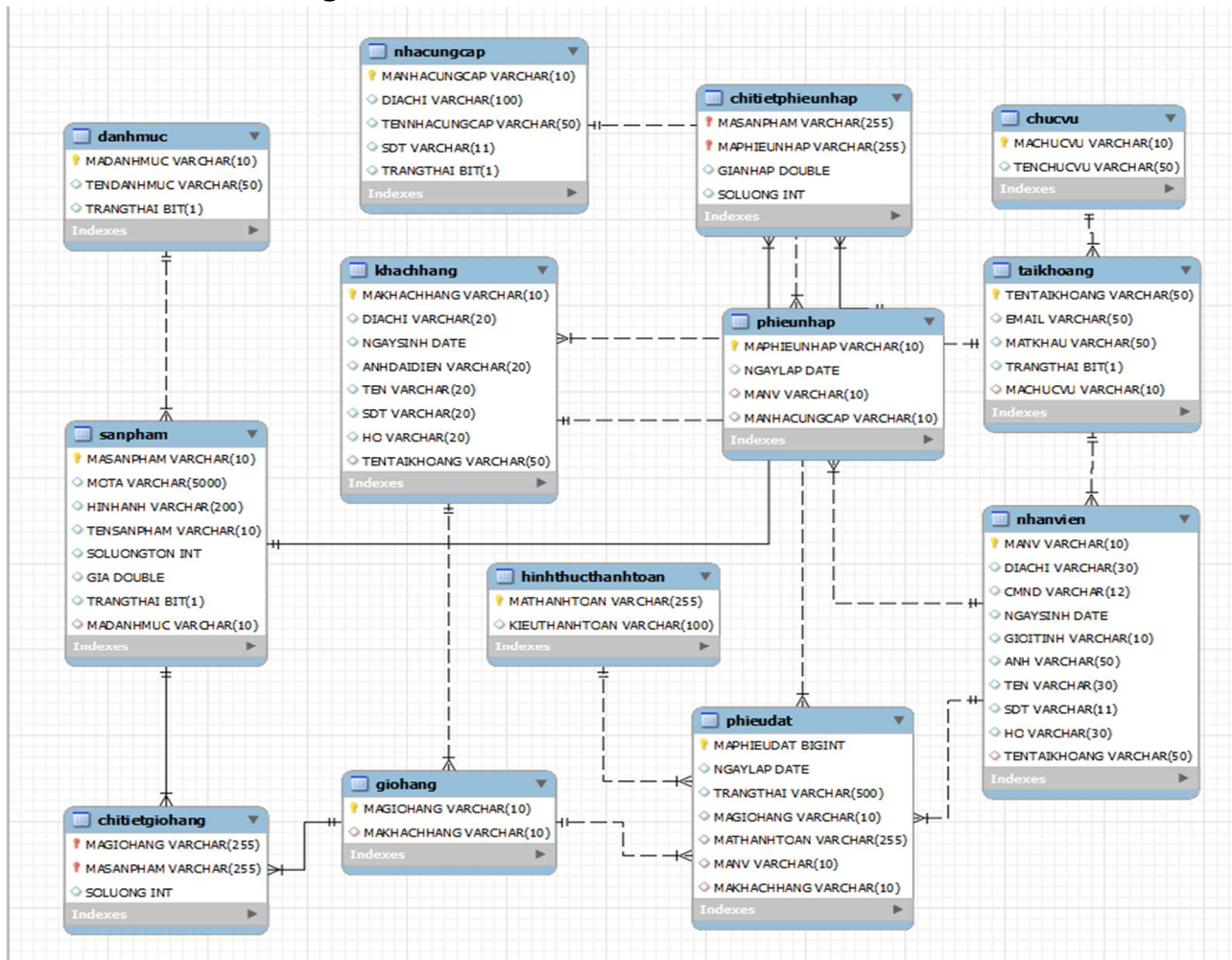
5. **Phân tích hệ thống:**

• **Gồm 13 bảng:**

- taikhoan
- nhanvien
- khachhang
- chucvu

- sanpham
- doanhmucsanpham
- phieunhap
- chitietphieudat
- phieudat
- giohang
- chitietgiohang
- hinhthucthanhtoan
- nhacungcap

• **Mô hình Diagram:**



Chi tiết bảng:

a. Tài khoản:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
TENTAIKHOANG	VARCHAR(50)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
EMAIL	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
MATKHAU	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
TRANGTHAI	BIT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
MACHUCVU	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

b. Khách hàng:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
MAKHACHHANG	VARCHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
DIACHI	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
NGAYSINH	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
ANHDAIDJEN	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
TEN	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
SDT	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
HO	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
TENTAIKHOANG	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

c. Nhân viên:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
MANV	VARCHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
DIACHI	VARCHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
CMND	VARCHAR(12)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
NGAYSINH	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
GIOITINH	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
ANH	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
TEN	VARCHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
SDT	VARCHAR(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
HO	VARCHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
TENTAIKHOANG	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

d. Chức vụ:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
MACHUCVU	VARCHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
TENCHUCVU	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

e. Phiếu đặt:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
MAPHIEUDAT	BIGINT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
NGAYLAP	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
TRANGTHAI	VARCHAR(500)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
MAGIOHANG	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
MATHANHTOAN	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
MANV	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
MAKHACHHANG	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

f. Nhà cung cấp:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
MANHACUNGCAP	VARCHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
DIACHI	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
TENNHACUNGCAP	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
SDT	VARCHAR(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
TRANGTHAI	BIT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

g. Doanh mục:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
MADANHMUC	VARCHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
TENDANHMUC	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
TRANGTHAI	BIT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

h. Giỏ hàng:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default
MAGIOHANG	VARCHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
MAKHACHHANG	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

i. Chi tiết giỏ hàng:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
MAGIOHANG	VARCHAR(255)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
MASANPHAM	VARCHAR(255)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SOLUONG	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

j. Sản phẩm:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
MASANPHAM	VARCHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
MOTA	VARCHAR(5000)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
HINHANH	VARCHAR(200)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
TENSANPHAM	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
SOLUONGTON	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
GIA	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
TRANGTHAI	BIT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
MADANHMUC	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
MALOA	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

k. Phiếu nhập:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
MAPHIEUNHAP	VARCHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
NGAYLAP	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
MANV	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
MANHACUNGCAP	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

l. Hình thức thanh toán:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
MATHANHTOAN	VARCHAR(255)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
KIEUTHANHTOAN	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

m. Chi tiết phiếu nhập:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
MASANPHAM	VARCHAR(255)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
MAPHIEUNHAP	VARCHAR(255)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
GIANHAP	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
SOLUONG	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

1. Trang chủ:

Phụ Kiện

Giảm Đến 50%+

AirPods GIẢM SỐC

Giá chỉ 2.640.000đ

PHỤ KIỆN XÃ KHO

GIẢ RẺ QUÁ

Sạc dự phòng

Tai nghe Bluetooth

Cáp, sạc điện thoại

Củ sạc điện thoại

Giá đỡ điện thoại

Cường Lực

Tai nghe có dây

Thẻ nhớ

Loa Bluetooth

Giá tiền >

Sắp xếp >

Xóa bộ lọc

Ốp lưng Điện Thoại Vivo Y20 Y20i Y20S

50,000 đ

Tai nghe Bluetooth Samsung Galaxy Buds Pro SM-R190

3,900,000 đ

Tai nghe Có Dây Samsung IG935

300,000 đ

Bộ sạc nhanh Samsung A30| A30S chính hãng

100,000 đ

Miếng dán Kính Cường Lực cho Samsung Galaxy A50S

90,000 đ

Ốp lưng TPU dẻo nơ gắn thú nỉ dành cho Samsung A73

45,000 đ

Ốp lưng điện thoại TPU mờ điện thoại cho iPhone 12 pro max

50,000 đ

Ốp Lưng JIUMOO Cho Samsung Galaxy M51

56,000 đ

1

2

Giao hàng hỏa tốc trong 1 giờ


Thanh toán linh hoạt: tiền mặt, visa / master, trả góp

Trải nghiệm sản phẩm tại nhà

Lỗi đổi tại nhà trong 1 ngày


Hỗ trợ suốt thời gian sử dụng.
Hotline: 12345678

2. Chi tiết sản phẩm



[Giỏ hàng](#) [Tài khoản](#)

Tai nghe Bluetooth Samsung Galaxy Buds Pro SM-R190



3900000.0

KHUYẾN MÃI

- Khách hàng sẽ được thử máy miễn phí tại cửa hàng. Có thể đổi trả lỗi trong vòng 2 tháng.
- Trong hộp có: Dây đủ các dụng cụ hỗ trợ để sử dụng sản phẩm
- Bảo hành chính hãng 12 tháng.
- 1 đổi 1 trong 1 tháng nếu lỗi, đổi sản phẩm tại nhà trong 1 ngày.


Thêm vào giỏ hàng
Giao trong 1 giờ hoặc nhận lại cửa hàng

Mô tả


Mô tả chi tiết:

Thiết bị cải tiến âm thanh 3D công nghệ công nghệ Dolby Head Tracking™ cho chất lượng âm thanh sống động, tai nghe còn trang bị chống nước IPX7, kết nối thông minh giữa các thiết bị Samsung.


*** Bạn có thể thích ***



Tai nghe Bluetooth Samsung Galaxy Buds Pro SM-R190
3,900,000 đ



Tai nghe Có Dây Samsung IG935
300,000 đ



Tai nghe AKG kèm Galaxy S8
550,000 đ

3. Trang đăng ký, đăng nhập

Đăng kí tài khoản PhonesShop


Tên tài khoản

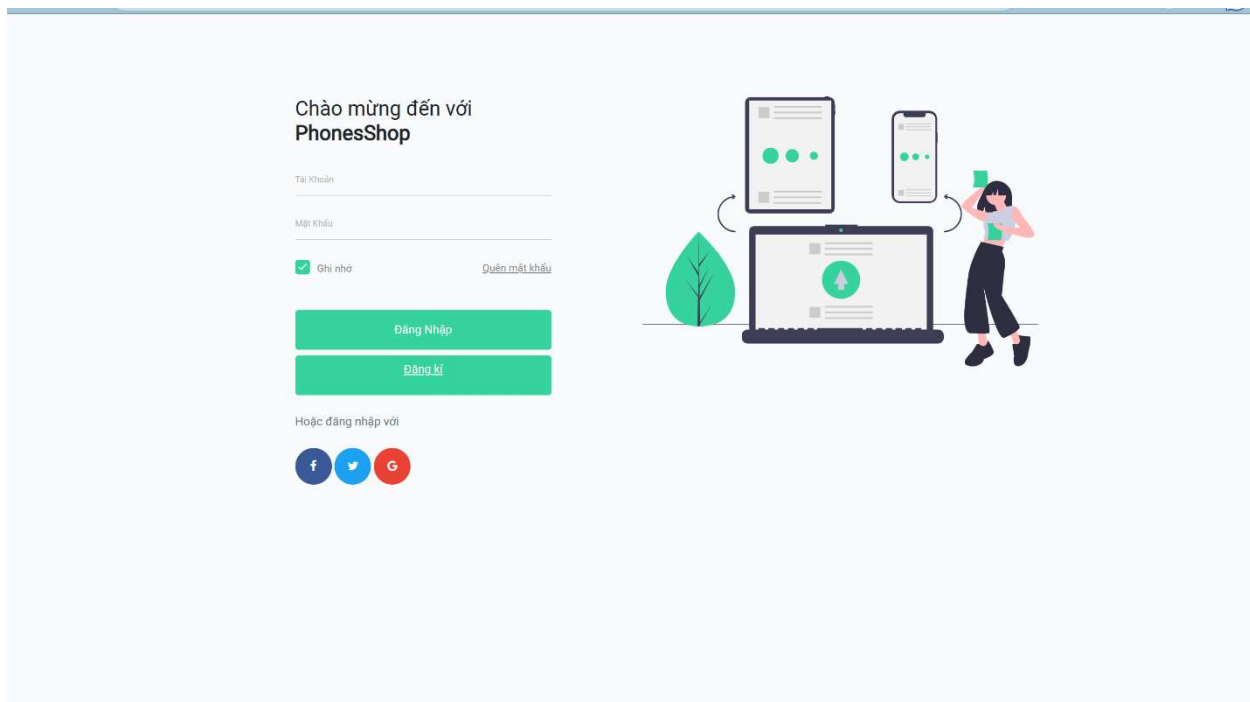
Email

Password

Xác nhận lại mật khẩu

Nếu bạn có tài khoản? [Đăng nhập](#)





4. Trang thông tin khách hàng

Hình ảnh

THÔNG TIN KHÁCH HÀNG

Tai khoản:

Họ:

Tên:

Email:

Danh sách đơn hàng

Mã phiếu đặt:

Trạng thái:

Địa chỉ:

Giao hàng hỏa tốc trong 1 giờ

Thanh toán linh hoạt: tiền mặt, visa / master, trả góp

Trải nghiệm sản phẩm tại nhà

Lỗi đổi tại nhà trong 1 ngày

Hỗ trợ suốt thời gian sử dụng.
Hotline: 12345678

5. Trang giỏ hàng

Sản phẩm	Đơn giá	Số lượng	Số tiền	Xóa
Ốp lưng Điện Thoại Vivo Y20i Y20S	50,000 đ	1	50,000 đ	<input type="button" value="X"/>

Giao tới:

Số điện thoại:

Địa chỉ:

Tổng cộng: 50,000 đ

Giao hàng hỏa tốc trong 1 giờ | Thanh toán linh hoạt: tiền mặt, visa / master, trả góp | Trải nghiệm sản phẩm tại nhà | Lỗi đổi tại nhà trong 1 ngày | Hỗ trợ suốt thời gian sử dụng. Hotline: [12345678](tel:12345678)

6. Trang đặt hàng

Địa chỉ nhận hàng

Số điện thoại: 0456875421

Địa chỉ: HCM

Sản phẩm	Đơn giá	Số lượng	Thành tiền
Ốp lưng Điện Thoại Vivo Y20i Y20S	50,000 đ	1	50,000 đ

Chọn hình thức thanh toán: Thanh toán khi nhận hàng

Giảm phí vận chuyển: Không có

Nhấn vào đặt hàng đồng nghĩa với việc bạn đồng ý với các điều khoản của Shop và có trách nhiệm với đơn hàng của mình.

Tổng tiền hàng: 50,000 đ

Phí vận chuyển: + 0 đ

Tổng cộng: 50,000 đ

Giao hàng hỏa tốc trong 1 giờ | Thanh toán linh hoạt: tiền mặt, visa / master, trả góp | Trải nghiệm sản phẩm tại nhà | Lỗi đổi tại nhà trong 1 ngày | Hỗ trợ suốt thời gian sử dụng. Hotline: [12345678](tel:12345678)

7. Trang địa chỉ giao hàng

ĐỊA CHỈ GIAO HÀNG

Mặc định

Số điện thoại: 0456875421

Địa chỉ: HCM

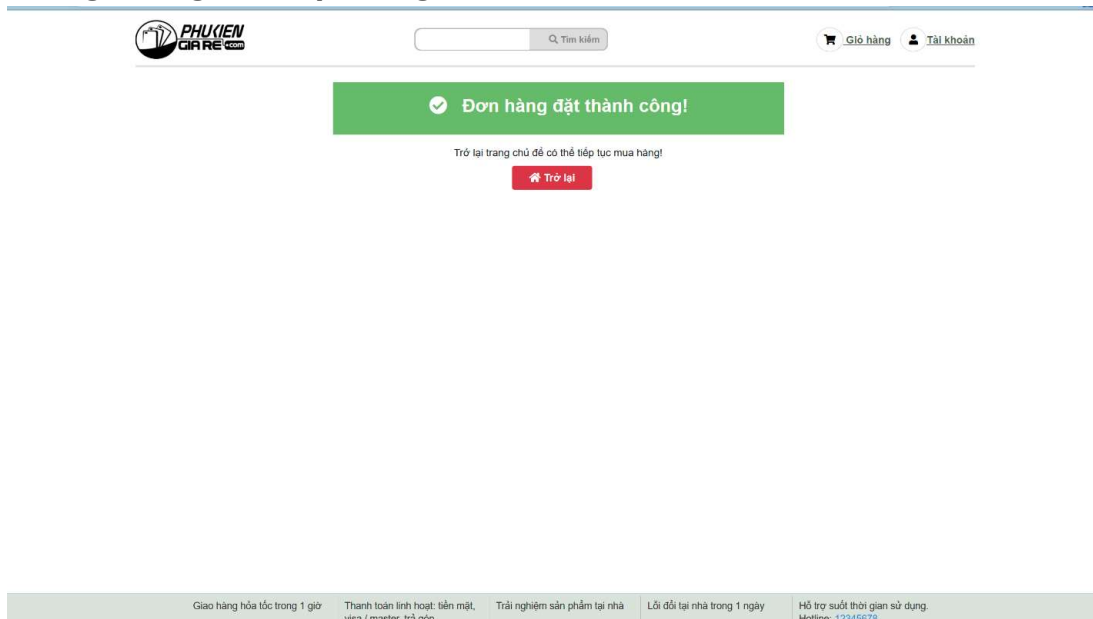
Địa chỉ mới

Điện thoại di động:

Địa chỉ:

Giao hàng hỏa tốc trong 1 giờ | Thanh toán linh hoạt: tiền mặt, visa / master, trả góp | Trải nghiệm sản phẩm tại nhà | Lỗi đổi tại nhà trong 1 ngày | Hỗ trợ suốt thời gian sử dụng. Hotline: [12345678](tel:12345678)

8. Trang thông báo đặt hàng



9. Trang đăng ký thông tin khách hàng

The screenshot shows the PHUKIEN GIARE.COM website header. Below the header, a form titled 'Đăng kí thông tin' (Register information) is displayed. The form contains the following fields: 'Họ' (Last name), 'Tên' (First name), 'Ngày sinh' (Date of birth) with a placeholder 'mm/dd/yyyy' and a calendar icon, 'Địa chỉ' (Address), and 'Số điện thoại' (Phone number). A blue 'Đăng kí' (Register) button is located at the bottom right of the form. The footer is identical to the previous screenshot, with five columns of text: 'Giao hàng hỏa tốc trong 1 giờ', 'Thanh toán linh hoạt: tiền mặt, visa / master, trả góp', 'Trải nghiệm sản phẩm tại nhà', 'Lỗi đổi tại nhà trong 1 ngày', and 'Hỗ trợ suốt thời gian sử dụng. Hotline: 12345678'.

10. Quản lý sản phẩm

1. Thêm sản phẩm

File hình ảnh
 No file chosen

Mã sản phẩm

Tên sản phẩm

Mô tả

Giá

Loại sản phẩm

Loại dòng máy

Thêm sản phẩm thành công

Tạo

2. Danh sách sản phẩm

Danh sách sản phẩm						
Home / Dashboard						
Ảnh sản phẩm	Mã sản phẩm	Tên sản phẩm	Đánh giá sản phẩm	Giá sản phẩm	Số lượng sản phẩm	Thao tác
	sp1	acre517	giá rẻ	2.0E7	10	<input type="button" value="Sửa"/> <input type="button" value="Xem cấu hình"/>
	sp2	lenovo12	sss	20.0	0	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/> <input type="button" value="Thêm cấu hình"/>

11. Quản lý phiếu nhập

1. Thêm phiếu nhập

Thêm phiếu nhập

Home / Dashboard

Thêm phiếu nhập mới

Mã phiếu nhập:

pn1

Ngày lập

mm/dd/yyyy

Nhà cung cấp

ncc1

Thêm mới

2. Danh sách phiếu nhập

Danh sách phiếu nhập

Home / Dashboard

Mã phiếu nhập	Ngày nhập	Nhân viên lập	Nhà cung cấp	Trạng thái	Thao tác
pn1		admin	test	0	<button>Thêm chi tiết phiếu nhập</button> <button>Xóa</button>

12. Quản lý phiếu đặt.

1. Giao diện danh sách phiếu đặt của khách hàng khi gửi phiếu

Danh sách phiếu đặt

Home / Dashboard

Mã phiếu đặt	Ngày đặt	Tên tài khoản	Nhân viên xác nhận	Hình thức thanh toán	Trạng thái	Thao tác
1	03/06/2023	trongnhan		Thanh toán khi nhận hàng	Đang xử lý	<button>Xem chi tiết phiếu đặt</button>

khi nhấn vào nút xem chi tiết sẽ bung ra chi tiết phiếu đặt.

2. Giao diện chi tiết phiếu đặt

Chi tiết phiếu đặt

Home / Dashboard

Thông tin khách hàng

Họ tên: Bùi Quốc Triệu

Số điện thoại: 0383460407

Địa chỉ: hcm

Sản phẩm	Số lượng	Giá sản phẩm	Tổng giá
acre517	3	2.0E7	6.0E7
lenovo12	3	20.0	60.0

[Quay lại](#)[Xác nhận](#)[Hủy](#)

3. Danh sách phiếu đặt

Danh sách phiếu đặt

Home / Dashboard

Mã phiếu đặt	Ngày đặt	Tên tài khoản	Nhân viên xác nhận	Hình thức thanh toán	Trạng thái	Thao tác
1	03/06/2023	trongnhan	admin	Thanh toán khi nhận hàng	Đã Xác Nhận	Xem chi tiết phiếu đặt Đóng gói Hoàn Tất

13. Quản lý nhà cung cấp.

1. Thêm nhà cung cấp

Thêm Nhà Cung Cấp

Home / Dashboard

Thêm nhà cung cấp

Mã nhà cung cấp:

Tên nhà cung cấp:

Địa chỉ nhà cung cấp:

SĐT nhà cung cấp:

[Thêm mới](#)

2. Danh sách nhà cung cấp

Danh sách nhà cung cấp				
Home / Dashboard				
Mã nhà cung cấp	Tên nhà cung cấp	Địa chỉ nhà cung cấp	SĐT nhà cung cấp	Thao tác
ncc1	test	hcm	0383460407	<button>Sửa</button> <button>Xóa</button>

14. Quản lý nhân viên

1. Thêm nhân viên

Thêm nhân viên

Tài Khoản

nhanvien

Email

sktt1tnhan@gmail.com

Mật Khẩu

Xác nhận mật khẩu

Chọn Quyền

☐ Nhân viên ☐ Admin

Đăng kí thành công

Đăng Kí

2. Danh sách nhân viên


Danh sách nhân viên						
Home / Dashboard						
Mã nhân viên	Họ Tên	Địa chỉ	Ngày sinh	SĐT	Tên tài khoản	Thao tác
pEfpI4H308	Nguyễn Hân	hcm	2002-01-03	0383460407	nhanvien	<button>Xóa</button>

15. Quản lý thông tin nhân viên

1. Thêm thông tin của nhân viên





Profile

Home / Users / Profile



Nguyễn Hân

EMPLOYEE

Tổng quan

Chỉnh sửa thông tin

Đổi mật khẩu

Hình ảnh

Choose File

No file chosen

Họ

Nguyễn

Tên

Hân

CMND

123456789

Địa chỉ

hcm

Giới Tính

nam

SĐT

0383460407

Ngày Sinh


06/03/2002

Save Changes

2. Giao diện thông tin nhân viên





Profile

Home / Users / Profile



Nguyễn Hân

EMPLOYEE

Tổng quan

Chỉnh sửa thông tin

Đổi mật khẩu

Thông tin

Chi tiết thông tin

Họ Tên

Nguyễn Hân

Cửa Hàng

PhoneShop

Công việc

nhanvien

Địa chỉ

hcm


SĐT

0383460407

3. Giao diện thay đổi mật khẩu





Profile

Home / Users / Profile



Nguyễn Hân

EMPLOYEE

Tổng quan

Chỉnh sửa thông tin

Đổi mật khẩu

Current Password

New Password

Re-enter New Password

Change Password

16. Quản lý danh mục sản phẩm

1. Danh sách danh mục

Mã thể loại	Tên thể loại	Thao tác
dm2	acer	<button>Sửa</button> <button>Xóa</button>
dm3	lenovo	<button>Sửa</button> <button>Xóa</button>

© Copyright NiceAdmin. All Rights Reserved
Designed by BootstrapMade

III. TÍNH NĂNG

1. Tính năng quản lý:

a. Đăng ký, đăng nhập:

* Đăng nhập:

@RequestMapping("/login")

public ModelAndView login() {

ModelAndView mav = **new** ModelAndView("login/login");

return mav;

}

Cho phép đăng nhập vào trang web.

Dùng security để xử lý login: cấu hình security.xml kiểm tra đăng nhập thành công hay thất bại nếu thành công thì xem tài khoản thuộc quyền nào nếu là user thì sẽ redirect tới trang chủ user để sử dụng chức năng user còn tài khoản quyền admin và employer sẽ redirect tới trang admin để quản lý trang admin

<?xml version="1.0" encoding="UTF-8"?>

<beans:beans xmlns="http://www.springframework.org/schema/security"

xmlns:beans="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.2.xsd

http://www.springframework.org/schema/security

http://www.springframework.org/schema/security/spring-security-

3.2.xsd">

```

<http auto-config="false" use-expressions="true" disable-url-rewriting="true">

    <intercept-url pattern="/admin/**"
access="hasAnyRole('ADMIN','EMPLOYEE')" />

    <access-denied-handler error-page="/accessDenied" />
    <form-login login-page="/login"
        username-parameter="j_username"
        password-parameter="j_password"
        login-processing-url="/j_spring_security_check"
        authentication-success-handler-ref="customSuccessHandler"
        authentication-failure-url="/login?incorrectAccount"/>
    <logout delete-cookies="JSESSIONID"/>
</http>

<authentication-manager >
    <authentication-provider user-service-ref="CustomUserDetailsService">
        <password-encoder ref="bcryptEncoder"/>
    </authentication-provider>
</authentication-manager>

<beans:bean id="bcryptEncoder"
class="org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder" />
<beans:bean id="CustomUserDetailsService"
class="phoneaccessories.service.impl.CustomUserDetailsService" />
<beans:bean id="customSuccessHandler"
class="phoneaccessories.securiry.CustomSuccessHandler" />

</beans:beans>

```

Đây là phần xử lý đăng nhập thành công hay thất bại.

```

public class CustomUserDetailsService implements UserDetailsService{

    @Autowired
    private AccountRepository accountRepository;

    @Override
    public UserDetails loadUserByUsername(String username) throws
UsernameNotFoundException {
        Account user = accountRepository.findByUsername(username);

        if (user == null) {

```

```

        throw new UsernameNotFoundException("User name not
fond");
    }

    List<GrantedAuthority> authorities = new ArrayList<>();
    authorities.add(new
SimpleGrantedAuthority(user.getPosition().getName()));

    MyUser myUser = new MyUser(user.getUsername(),
user.getPasword(), true, true, true, true, authorities);

    if (user.getPosition().getName().equals("ADMIN") ||
user.getPosition().getName().equals("EMPLOYEE")) {
        myUser.setFullName(user.getStaff().getSurname() + " "+
user.getStaff().getName());
        myUser.setImage(user.getStaff().getImage());
        myUser.setPosition(user.getPosition().getName());
    }
    else {
        myUser.setFullName(user.getUser().getSurname() + " "+
user.getUser().getName());
        myUser.setImage(user.getUser().getImage());
        myUser.setPosition(user.getPosition().getName());
    }
    return myUser;
}
}
}

```

Khi đăng nhập thành công xét phân quyền và hướng redirect

```

@Component
public class CustomSuccessHandler extends
SimpleUrlAuthenticationSuccessHandler {
    private RedirectStrategy redirectStrategy = new DefaultRedirectStrategy();
    @Override
    public void handle(HttpServletRequest request, HttpServletResponse response,
Authentication authentication)
    throws IOException {
        String targetUrl = determineTargetUrl(authentication);
        if (response.isCommitted()) {
            return;
        }
    }
}

```

```

redirectStrategy.sendRedirect(request, response, targetUrl);
}
private String determineTargetUrl(Authentication authentication) {
String url = "";
List<String> roles = SecurityUtils.getAuthorities();
if (isAdmin(roles)) {
url = "/admin/home";
} else if (isUser(roles)) {
url = "/trang-chu";
}
return url;
}
private boolean isAdmin(List<String> roles) {
if (roles.contains("ADMIN")||roles.contains("EMPLOYEE")) {
return true;
}
return false;
}
private boolean isUser(List<String> roles) {
if (roles.contains("USER")) {
return true;
}
return false;
}
public RedirectStrategy getRedirectStrategy() {
return redirectStrategy;
}
public void setRedirectStrategy(RedirectStrategy redirectStrategy) {
this.redirectStrategy = redirectStrategy;
}
}

```

* Đăng ký:

```

@RequestMapping(value = "/staff/register",method = RequestMethod.GET)
public ModelAndView singin() {
ModelAndView mav = new ModelAndView("/admin/staff/add-staff");
Account account = new Account();
mav.addObject("account",account );
return mav;
}
@SuppressWarnings("null")
@PostMapping("/staff/signup")

```

```

public ModelAndView creatAccount(@ModelAttribute("account") Account
account , @RequestParam("passwordConfirm") String passwordCf, BindingResult
bindingResult,
@RequestParam("role") String role ) {
ModelAndView mav = new ModelAndView("/admin/staff/add-staff");
mav.addObject("check", false);
String message = "";
if (account.getUsername() == "") {
message = "Tài khoản không được để trống";
mav.addObject("checktk", true);
mav.addObject("errorun", message);
return mav;
}
mav.addObject("checktk", false);
if (account.getEmail() == "") {
message = "Email không được để trống";
mav.addObject("checkemail", true);
mav.addObject("errorem", message);
return mav;
}
mav.addObject("checkemail", false);
if (account.getPasword() == "") {
message = "Mật khẩu không được để trống";
mav.addObject("checkmk", true);
mav.addObject("errorpw", message);
return mav;
}
mav.addObject("checkmk", false);
if (bindingResult.hasErrors()) {
message = "Đăng kí thất bại";
mav.addObject("message", message);
return mav;
}
if ( !passwordCf.equals(account.getPasword())) {
mav.addObject("checkPassword", true);
mav.addObject("error", "Xác thực mật khẩu thất bại");
return mav;
}
if (role.equals("ADMIN")) {
Position position = positionService.findOneByName(role);
if (position == null) {
position = new Position();
position.setld("admin");

```

```

position.setName(role);
positionService.save(position);
}
account.setPosition(position);
}
if (role.equals("EMPLOYEE")) {
Position position = positionService.findOneByName(role);
if (position == null) {
position = new Position();
position.setld("employee");
position.setName(role);
positionService.save(position);
}
account.setPosition(position);
}
message = "Đăng kí thành công";
mav.addObject("message", message);
mav.addObject("check", true);
accountService.saveStaff(account);
Staffs staffs = new Staffs();
staffs.setAccount(account);
staffs.setld(RandomStringGenerator.generateRandomString(10));
staffService.save(staffs);
return mav;
}

```

Khi một nhân viên mới vào sẽ được cấp cho một tài khoản, sau khi đăng nhập vào thì nhân viên tự điền các thông tin của mình và thay đổi mật khẩu. Chỉ admin mới được quyền đăng ký tài khoản cho nhân viên.

b. Xem, cập nhập thông tin tài khoản:

* Xem thông tin tài khoản:

```

@RequestMapping("/profile")
public ModelAndView proFile() {
ModelAndView mav = new ModelAndView();
Account account =
accountService.findByUsername(SecurityUtils.getPrincipal().getUsername());
Staffs staff = staffService.findOneByAccount(account);
mav.addObject("staff", staff);
mav.setViewName("/admin/staff/profile");
}

```



```
return mav;
}
```

Nhân viên có thể xem được thông tin cá nhân của mình khi đăng nhập vào hệ thống.

* **Chỉnh sửa thông tin tài khoản:**

```
@PostMapping("/profile/edit")
public ModelAndView editProfile(@ModelAttribute("staff") Staffs staff
, @RequestParam("avatar") MultipartFile avatar,
@RequestParam("ids") String ids) {
    ModelAndView mav = new ModelAndView();
    Account account =
accountService.findByUsername(SecurityUtils.getPrincipal().getUsername());
    staff.setImage(avatar.getOriginalFilename());
    staff.setId(ids);
    staff.setAccount(account);
    System.out.println(staff.getSurname());
    System.out.println(staff.getName());
    staffService.save(staff);
    String message = "";
    mav.setViewName("/admin/staff/profile");
    try {
        MultipartFile multipartFile = avatar;
        String fileName = multipartFile.getOriginalFilename();
        File file = new
File("D:\\ProjectLTweb\\PhoneAccessories\\src\\main\\webapp\\templates\\image
s\\",
fileName);
        multipartFile.transferTo(file);
        message = "Thêm sản phẩm thành công";
        mav.addObject("message", message);
    } catch (Exception e) {
        e.printStackTrace();
        mav.addObject("message", "Lưu file ảnh thất bại");
    }
    mav.setViewName("/admin/staff/profile");
    return mav;
}
```

Nhân viên có quyền chỉnh sửa thông tin cá nhân của bản thân vào lưu thông tin chỉnh sửa vào CSDL.

c. Quản lý tài khoản(khóa tài khoản nhân viên-khoá tài khoản khách hàng)

```
@RequestMapping("/staff/delete")
public ModelAndView deleteStaff(@RequestParam("usernameac") String
username) {
    ModelAndView mav = new ModelAndView();
    staffService.delete(username);
    mav.setViewName("redirect:/admin/staff/list");
    return mav;
}
```

Admin có quyền khóa tài khoản của nhân viên và khách hàng. Khi khóa thì không thể dùng tài khoản đó để đăng nhập vào hệ thống.

d. Xem, thêm, xóa, sửa sản phẩm:

* Xem sản phẩm:

```
@RequestMapping("product/list")
public ModelAndView listProduct() {
    ModelAndView mav = new ModelAndView("admin/product/listproduct");
    List<Product> listProduct = pepository.findAllByStatus(true);
    mav.addObject("listProduct", listProduct);
    return mav;
}
```

Khi đăng nhập vào hệ thống với đúng quyền hạn, thì có thể xem được danh sách tất cả các sản phẩm.

* Thêm sản phẩm:

```
@GetMapping("product/add")
public ModelAndView showPageAddProdcut() {
    ModelAndView mav = new ModelAndView();
    mav.setViewName("admin/product/add-product");
    Product product = new Product();
    mav.addObject("addProdcut", product);
    return mav;
}

@PostMapping("product/add")
public ModelAndView addProdcut(@ModelAttribute("addProdcut") Product
product,
@RequestParam("avatar") MultipartFile avatar) {
    ModelAndView mav = new ModelAndView();
    mav.setViewName("admin/product/add-product");
}
```

```

String message = "";
if (avatar.isEmpty()) {
mav.addObject("imgnull", "Hình ảnh trống");
return mav;
}
if (product.getId().isEmpty()) {
mav.addObject("idnull", "Mã sản phẩm trống");
return mav;
}
if (product.getName().isEmpty()) {
mav.addObject("namenull", "Tên sản phẩm trống");
return mav;
}
product.setImage(avatar.getOriginalFilename());
mav.addObject("success", "Thêm sản phẩm thành công");
productService.save(product);
try {
MultipartFile multipartFile = avatar;
String fileName = multipartFile.getOriginalFilename();
File file = new
File("D:\\ProjectL2web\\PhoneAccessories\\src\\main\\webapp\\templates\\image
s\\",
fileName);
multipartFile.transferTo(file);
message = "Thêm sản phẩm thành công";
mav.addObject("message", message);
} catch (Exception e) {
e.printStackTrace();
mav.addObject("message", "Lưu file ảnh thất bại");
}
return mav;
}

```

Cho phép Admin và nhân viên thêm cái sản phẩm vào danh sách sản phẩm. Khi bấm vào nút submit thông tin của sản phẩm sẽ được đẩy lên CSDL.

* Sửa sản phẩm:

```

@GetMapping("product/edit")
public ModelAndView viewEdit (@RequestParam("idp") String idProduct) {
ModelAndView mav = new ModelAndView();
mav.setViewName("admin/product/edit-product");
Product product = productService.getProductById(idProduct);
mav.addObject("product", product);
return mav;
}

```

```

}
@PostMapping("product/edit")
public ModelAndView editProduct(@ModelAttribute("product") Product product
, @RequestParam("avatar") MultipartFile avatar, @RequestParam("idp") String
idp) {
    ModelAndView mav = new ModelAndView();
    String message = "";
    mav.setViewName("admin/product/edit-product");
    product.setImage(avatar.getOriginalFilename());
    productService.update(product, idp);
    try {
        String filename = avatar.getOriginalFilename();
        File file = new
        File("D:\\ProjectLTweb\\PhoneAccessories\\src\\main\\webapp\\templates\\image
s\\"+filename);
        avatar.transferTo(file);
        message = "Cập Nhật sản phẩm thành công";
        mav.addObject("message", message);
    } catch (Exception e) {
        e.printStackTrace();
        mav.addObject("message", "Lưu file ảnh thất bại");
    }
    return mav;
}

```

Cho phép chỉnh sửa thông tin của sản phẩm, nhấn vào nút [Sửa] thông tin của sản phẩm sẽ được hiển thị và cho phép chỉnh sửa. Khi đã sửa xong hết thông tin bấm vào nút [Lưu] dữ liệu vừa được chỉnh sửa sẽ được lưu trên CSDL.

* Xóa sản phẩm:

```

@GetMapping("product/delete")
public ModelAndView deleteProduct(@RequestParam("idp") String idProduct) {
    ModelAndView mav = new ModelAndView();
    mav.setViewName("redirect:/admin/product/list");
    Product product = productService.getProductById(idProduct);
    product.setStatus(false);
    repository.save(product);
    return mav;
}

```

Chỉ cho phép xóa những sản phẩm nào có số lượng tồn bằng 0. Xóa chỉ là chuyển trạng thái của sản phẩm chứ không phải là xóa sản phẩm khỏi database.

e. Xem, thêm, xóa, sửa danh mục sản phẩm:

*** Xem doanh mục sản phẩm:**

```
@RequestMapping("category/list")
public ModelAndView listCategory() {
    ModelAndView mav = new ModelAndView("admin/category/list-category");
    List<Category> listCategory = new ArrayList<>();
    listCategory = categoryImpl.getListCategoryByStatus(true);
    mav.addObject("listCategory", listCategory);
    return mav;
}
```

Khi đăng nhập vào hệ thống, vào mục danh mục sản phẩm bấm vào danh sách thì tất cả các danh mục sản phẩm để người quản lý xem.

*** Thêm doanh mục sản phẩm:**

```
@GetMapping("category/add")
public ModelAndView viewAddCategory() {
    ModelAndView mav = new ModelAndView("admin/category/add");
    Category category = new Category();
    mav.addObject("addcategory", category);
    return mav;
}

@PostMapping("category/add")
public ModelAndView addCategory(@ModelAttribute("addcategory") Category category) {
    ModelAndView mav = new ModelAndView("admin/category/add");
    if (category.getId().isEmpty()) {
        mav.addObject("idnull", "Id thư mục trống");
        return mav;
    }
    if (category.getName().isEmpty()) {
        mav.addObject("namenull", "Tên thư mục trống");
        return mav;
    }
    mav.addObject("success", "Thêm thư mục thành công");
    categoryImpl.addCategory(category);
    return mav;
}
```

Khi đăng nhập vào hệ thống, tìm đến mục doanh mục sản phẩm, nhấn vào “Thêm” giao diện thêm doanh mục sản phẩm sẽ hiển thị lên cho phép nhân viên tiến hành thêm doanh mục sản phẩm. Khi điền đầy đủ thông tin nhấn vào nút [Tạo] một doanh mục sản phẩm sẽ được thêm vào CSDL.

*** Sửa doanh mục sản phẩm:**

```
@GetMapping("category/edit")
```

```

public ModelAndView viewEditCategory(@RequestParam("idc") String
idcategory) {
    ModelAndView mav = new ModelAndView("admin/category/edit");
    Category category = categoryImpl.findOneById(idcategory);
    mav.addObject("editcategory", category);
    return mav;
}
@PostMapping("category/edit")
public ModelAndView editCategory(@ModelAttribute("editcategory") Category
category,
@RequestParam("idc") String idcategory) {
    ModelAndView mav = new ModelAndView("admin/category/edit");
    categoryImpl.update(category, idcategory);
    Category categoryold = categoryImpl.findOneById(idcategory);
    if (categoryold.getId().equals(category.getId())) {
    } else {
        categoryImpl.deleteCategory(categoryold.getId());
    }
    return mav;
}

```

Phần mềm cho phép nhân viên chỉnh sửa thông tin của doanh mục sản phẩm. Khi chỉnh sửa các thông tin muốn chỉnh sửa bấm vào nút [Lưu] để lưu thông tin chỉnh sửa vào cơ sở dữ liệu. Bấm [Hủy] để hủy các thông tin đã chỉnh sửa trước đó.

*** Xóa doanh mục sản phẩm:**

```

@GetMapping("category/delete")
public ModelAndView removeCategory(@RequestParam("idc") String
idcategory) {
    ModelAndView mav = new ModelAndView("admin/category/list-category");
    List<Product> productds = productService.findAllByCategory_Id(idcategory);
    if (productds.isEmpty()) {
        categoryImpl.deleteCategory(idcategory);
    }
    else {
        mav.addObject("idc", idcategory);
        mav.addObject("message", "Sản phẩm tồn tại không thể xóa");
    }
    List<Category> listCategory = new ArrayList<>();
    listCategory = categoryImpl.getListCategoryByStatus(true);
    mav.addObject("listCategory", listCategory);
    return mav;
}

```

Chỉ được phép xóa doanh mục sản phẩm khi không có sản phẩm nào trong doanh mục đó. Xóa chỉ là chuyển trạng thái chứ không phải là xóa doanh mục sản phẩm khỏi CSDL.

f. Xem, thêm, xóa, xác nhận, hoàn thành phiếu nhập- Xem, thêm chi tiết phiếu nhập

*** Xem phiếu nhập:**

```
@RequestMapping("receipt/list")
public ModelAndView listReceipt() {
    ModelAndView mav = new ModelAndView();
    mav.setViewName("admin/receipt/list-receipt");
    List<Receipt> listRc = receiptService.getListReceipt();
    mav.addObject("listReceipt", listRc);
    return mav;
```

Cho phép nhân viên xem được danh sách các phiếu nhập đã lập trước đó

*** Thêm phiếu nhập:**

```
@GetMapping("receipt/add")
public ModelAndView viewAddReceipt() {
    ModelAndView mav = new ModelAndView("admin/receipt/add-receipt");
    Receipt receipt = new Receipt();
    mav.addObject("addreceipt", receipt);
    return mav;
}

@PostMapping("receipt/add")
public ModelAndView addReceipt(@ModelAttribute("addreceipt") Receipt receipt) {
    ModelAndView mav = new ModelAndView("admin/receipt/add-receipt");
    Account account =
        accountService.findByUsername(SecurityUtils.getPrincipal().getUsername());
    Staffs staffs = staffService.findOneByAccount(account);
    receipt.setStaff(staffs);
    receipt.setStatus(0); // status = 0 vừa lập phiếu
    receiptService.save(receipt);
    return mav;
}
```

Cho phép nhân viên thêm phiếu nhập, nhân viên vào hệ thống nhấn vào [Thêm phiếu nhập] và tiến hành thêm, sau đó nhấn vào nút [Thêm mới] để dữ liệu được lưu vào CSDL.

*** Xóa phiếu nhập:**

```
@GetMapping("receipt/delete")
```



```

public ModelAndView deleteReceipt(@RequestParam("idr") String idr) {
    ModelAndView mav = new ModelAndView();
    mav.setViewName("redirect:/admin/receipt/list");
    Receipt receipt = receiptService.findById(idr);
    receiptService.delete(receipt);
    return mav;
}

```

Phiếu nhập chỉ được xóa khi chưa xác nhận phiếu nhập.

*** Xác nhận, hủy phiếu nhập:**

```

@GetMapping("receipt-detail/confirmz")
public ModelAndView confirmReceiptDetail(@RequestParam("idr") String idr ,
    @RequestParam(value = "status",required = false) String status) {
    ModelAndView mav = new ModelAndView();
    Receipt receipt = receiptService.findById(idr);
    if (status == null) {
        receipt.setStatus(1);// 1 đã xác nhận
    }
    else
    {
        receipt.setStatus(2);// 2 đã hủy
    }
    receiptRepository.save(receipt);
    mav.setViewName("redirect:/admin/receipt/list");
    return mav;
}

```

Khi nhà cung cấp báo là đã nhận được thông tin đơn hàng, thì nhân viên sẽ chuyển phiếu nhập về trạng thái đã xác nhận. Hủy khi xảy ra sai sót và tiến hành lập phiếu nhập mới.

*** Hoàn thành phiếu nhập:**

```

@GetMapping("receipt/success")
public ModelAndView successReceipt(@RequestParam("idr") String idr) {
    ModelAndView mav = new ModelAndView();
    Receipt receipt = receiptService.findById(idr);
    receipt.setStatus(3); // 3 đã hoàn thành phiếu nhập
    List<ReceiptDetail> receiptDetails = receipt.getListReceiptDetail();
    for (ReceiptDetail receiptDetail : receiptDetails) {
        Product product =
        productService.getProductById(receiptDetail.getProduct().getId());
    }
}

```

```

product.setNumber(receiptDetail.getQuantity()+receiptDetail.getProduct().getNumber());
repository.save(product);
}
receiptRepository.save(receipt);
mav.setViewName("redirect:/admin/receipt/list");
return mav;
}

```

Khi đã nhận đủ đơn hàng sẽ xác nhận phiếu nhập được hoàn thành. (Khi phiếu nhập được xác nhận hoàn thành thông tin về số lượng sẽ được cộng dồn vào số lượng tồn và giá sản phẩm sẽ được lưu vào CSDL).

* Xem chi tiết phiếu nhập

```

@GetMapping("receipt-detail/list")
public ModelAndView getListReceiptDetail(@RequestParam("idr") String idr) {
    ModelAndView mav = new ModelAndView();
    List<ReceiptDetail> listReceiptDetail =
        receiptDetailRepository.findAllById_Receipt(idr);
    mav.addObject("listReceiptDetail", listReceiptDetail);
    float sum = 0;
    for (ReceiptDetail receiptDetail : listReceiptDetail) {
        sum += receiptDetail.getQuantity() * receiptDetail.getPrice();
    }
    mav.addObject("sum", sum);
    Receipt receipt = receiptService.findById(idr);
    mav.addObject("receipt", receipt);
    mav.setViewName("admin/receipt/receipt-detail/list-receiptdetail");
    return mav;
}

```

Ở cột thao tác của phiếu nhập có mục xem chi tiết phiếu nhập. Khi nhấn vào đó nhân viên sẽ thấy được chi tiết của phiếu nhập đó.

* Thêm chi tiết phiếu nhập:

```

@GetMapping("receipt-detail/quantity")
public ModelAndView getQuantityRcDI(@RequestParam("idr") String idr) {
    ModelAndView mav = new ModelAndView("/admin/receipt/quantityRcDI");
    mav.addObject("idr", idr);
    return mav;
}

@PostMapping("receipt-detail/quantity")
public ModelAndView setUpQuantityRcDI(@RequestParam("idr") String idr,HttpServletRequest request) {

```

```

ModelAndView mav = new ModelAndView("/admin/receipt/quantityRcDI");
String quantityRcDI = request.getParameter("quantityRcDI");
mav.addObject("idr", idr);
int quantity = Integer.valueOf(quantityRcDI);
if (quantity < 0) {
mav.setViewName("/admin/receipt/quantityRcDI");
}else {
Receipt receipt = new Receipt();
List<ReceiptDetail> listReceiptDetail = new ArrayList<>();
mav.setViewName("/admin/receipt/receipt-detail/create-receiptdetail");
for (int i = 0; i < quantity ; i++) {
ReceiptDetail receiptDetail = new ReceiptDetail();
receiptDetail.setPrice(0);
receiptDetail.setQuantity(0);
listReceiptDetail.add(receiptDetail);
}
receipt.setListReceiptDetail(listReceiptDetail);
receipt.setId(idr);
mav.addObject("chiTietPhieuNhap", receipt);
}
return mav;
}
@PostMapping("receipt-detail/create")
public ModelAndView
viewCreateReceiptDetail(@ModelAttribute("chiTietPhieuNhap") Receipt receipt
,@RequestParam("idr")
String idr) {
ModelAndView mav = new ModelAndView();
mav.setViewName("/admin/receipt/receipt-detail/create-receiptdetail");
receipt.setId(idr);
receiptService.save(receipt);
List<ReceiptDetail> receiptDetails = receipt.getListReceiptDetail();
float sum = 0;
for (ReceiptDetail receiptDetail : receiptDetails) {
sum += receiptDetail.getQuantity() * receiptDetail.getPrice();
}
mav.addObject("sum", sum);
return mav;
}

```

Khi bấm thêm chi tiết phiếu nhập, phải chọn loại số lượng sản phẩm, sau khi xác nhận sẽ tạo table gồm: n dòng đúng với loại sản phẩm vừa nhập, điền thông tin các sản phẩm vào từng dòng, khi xác nhận thì chi tiết của phiếu nhập sẽ được lưu vào cơ sở dữ liệu.

g. Duyệt, hủy phiếu đặt - xem chi chi tiết phiếu đặt

* Xem phiếu đặt:

```
@RequestMapping("order/list")
public ModelAndView listOrder() {
    ModelAndView mav = new ModelAndView();
    mav.setViewName("/admin/order/list-order");
    List<Order> listOrder = orderService.getListOrder();
    mav.addObject("listOrder", listOrder);
    return mav;
}
```

Khi khách hàng đặt đơn, nhân viên vào mục phiếu đặt để xem danh sách các phiếu đặt.

* Duyệt, hủy phiếu đặt:

```
@RequestMapping("order/confirm")
public ModelAndView confirmDetail(@RequestParam("ido") String ido
, @RequestParam(value = "status", required = false) String status) {
    Account account =
    accountService.findByUsername(SecurityUtils.getPrincipal().getUsername());
    Staffs staff = staffService.findOneByAccount(account);
    ModelAndView mav = new ModelAndView();
    mav.setViewName("redirect:/admin/order/list");
    Long number = Long.valueOf(ido);
    Order order = orderService.findOneById(number);
    if (status!=null) {
        order.setStatus("Đã Bị Hủy");
    }
    else
    {
        order.setStatus("Đã Xác Nhận");
    }
    order.setStaff(staff);
    orderService.save(order);
    return mav;
}
```

Khi khách hàng đặt hàng, nhân viên vào mục phiếu đặt → xem chi tiết phiếu đặt khi đó phiếu đặt đang ở trạng thái chờ xử lý, nhân viên có quyền xác nhận hoặc hủy phiếu đặt đồng thời phiếu đặt sẽ chuyển qua trạng thái mà nhân viên đã chọn.

* Xem chi tiết phiếu đặt:

```
@RequestMapping("order/detail")
```

```

public ModelAndView orderDetail(@RequestParam("idc") String idc) {
    ModelAndView mav = new ModelAndView();
    mav.setViewName("/admin/order/detail-order");
    List<CardDetail> listOrderDetail = orderDetailService.findAllByCardId(idc);
    mav.addObject("listOrderDetail", listOrderDetail);
    Order order = orderService.findOneByCard_Id(idc);
    mav.addObject("order", order);
    return mav;
}

```

Ở cột thao tác có mục xem chi tiết phiếu đặt, khi nhấn vào chi tiết của phiếu đặt đó sẽ được hiển thị lên cho nhân viên xem.

h. Xem, thêm, xóa, sửa nhà cung cấp

* Xem nhà cung cấp:

```

@GetMapping("supplier/list")
public ModelAndView listSupplier() {
    ModelAndView mav = new ModelAndView("admin/supplier/list-supplier");
    List<Supplier> listSupplier = supplierService.getListSupplierByStatus(true);
    mav.addObject("listSupplier", listSupplier);
    return mav;
}

```

Khi đăng nhập vào hệ thống, nhân viên tìm đến mục nhà cung cấp, nhấn vào danh sách khi đó danh sách các nhà cung cấp sẽ được hiển thị lên để xem.

* Thêm nhà cung cấp:

```

@GetMapping("supplier/add")
public ModelAndView viewAddSupplier() {
    ModelAndView mav = new ModelAndView();
    mav.setViewName("admin/supplier/add-supplier");
    Supplier supplier = new Supplier();
    mav.addObject("supplier", supplier);
    return mav;
}

@PostMapping("supplier/add")
public ModelAndView addSupplier(@ModelAttribute("supplier") Supplier
supplier) {
    ModelAndView mav = new ModelAndView();
    mav.setViewName("admin/supplier/add-supplier");
    supplierService.save(supplier);
    return mav;
}

```

Nhân viên đăng nhập vào hệ thống, vào nhà cung cấp chọn [Thêm] và tiến hành nhập thông tin, sau đó bấm [Thêm mới] dữ liệu sẽ được thêm vào CSDL

* Sửa nhà cung cấp:

```
@GetMapping("supplier/edit")
public ModelAndView viewEditSupplier(@RequestParam("ids") String
idsupplier) {
    ModelAndView mav = new ModelAndView();
    mav.setViewName("admin/supplier/edit-supplier");
    Supplier supplier = supplierService.findById(idsupplier);
    mav.addObject("supplier", supplier);
    return mav;
}

@PostMapping("supplier/edit")
public ModelAndView editSupplier(@RequestParam("ids") String
idsupplier, @ModelAttribute("supplier") Supplier supplier) {
    ModelAndView mav = new ModelAndView();
    mav.setViewName("admin/supplier/edit-supplier");
    supplierService.update(supplier, idsupplier);
    return mav;
}
```

Cho phép chỉnh sửa thông tin nhà cung cấp nếu muốn sửa đổi thông tin.

2. Tính năng của khách hàng

a. Cập nhật thông tin tài khoản.

```
@PostMapping("/profile/edit")
public ModelAndView editProfile(@ModelAttribute("staff") Staffs staff
, @RequestParam("avatar") MultipartFile avatar,
    @RequestParam("ids") String ids) {
    ModelAndView mav = new ModelAndView();

    Account account =
accountService.findByUsername(SecurityUtils.getPrincipal().getUsername());

    staff.setImage(avatar.getOriginalFilename());
    staff.setId(ids);
    staff.setAccount(account);
    System.out.println(staff.getSurname());
    System.out.println(staff.getName());
    staffService.save(staff);

    String message = "";
```

```

        mav.setViewName("/admin/staff/profile");
        try {
            MultipartFile multipartFile = avatar;
            String fileName = multipartFile.getOriginalFilename();
            File file = new
File("D:\\ProjectL\\Tweb\\PhoneAccessories\\src\\main\\webapp\\templates\\image
s\\",
                fileName);
            multipartFile.transferTo(file);
            message = "Thêm sản phẩm thành công";
            mav.addObject("message", message);
        } catch (Exception e) {
            e.printStackTrace();
            mav.addObject("message", "Lưu file ảnh thất bại");
        }
        mav.setViewName("/admin/staff/profile");
        return mav;
    }

```

b. Đăng ký tài khoản khách hàng

```

@RequestMapping(value = "/signinUser", method = RequestMethod.GET)
public ModelAndView signinUS() {
    ModelAndView mav = new ModelAndView("cart-pay/signinUS");
    Account account = new Account();
    mav.addObject("account", account);
    return mav;
}

@PostMapping("/signupUser")
public ModelAndView creatAccount( HttpSession
session, @ModelAttribute("account") Account account,
        @RequestParam("passwordConfirm") String passwordCf,
        @RequestParam("role") String role, BindingResult bindingResult) {
    ModelAndView mav = new ModelAndView("cart-pay/signinUS");
    mav.addObject("check", true);
    String message = "";
    if (account.getUsername() == "") {
        message = "Tài khoản không được để trống";
        mav.addObject("errorTK", message);
    }
}

```



```

        return mav;
    }
    if (account.getEmail() == "") {
        message = "Email không được để trống";
        mav.addObject("errorem", message);
        return mav;
    }

    if (account.getPasword() == "") {
        message = "Mật khẩu không được để trống";
        mav.addObject("errorpw", message);
        return mav;
    }
    if (bindingResult.hasErrors()) {
        message = "Đăng kí thất bại";
        mav.addObject("message", message);

        return mav;
    }

    if (!passwordCf.equals(account.getPasword())) {
        mav.addObject("checkPassword", true);
        mav.addObject("error", "Xác thực mật khẩu thất bại");
        message = "Đăng kí thất bại";
        mav.addObject("message", message);
        return mav;
    }

    mav.addObject("message", message);
    Position position = positionService.findOneByName(role);
    if (position == null) {
        position = new Position();
        position.setId("user");
        position.setName(role);
        positionService.save(position);
    }

```

```
account.setPosition(position);
```

```
accountService.saveUser(account);  
//accountImpl.save(account);
```

```
Tenuser = account.getUsername();  
session.setAttribute("account", account);  
message = "Đăng kí thành công";  
mav.setViewName("redirect:/DKuser");  
return mav;
```

```
}
```

Điền các thông tin: tên tài khoản, Email, mật khẩu, xác nhận mật khẩu. Nếu tài khoản chưa tồn tại thì khi bấm đăng ký thì sẽ đăng ký tài khoản thành công, các thông tin không được để trống nếu để trống sẽ bị báo lỗi. Các thông tin đăng ký sẽ lưu vào cơ sở dữ liệu. Sau đó chuyển sang trang đăng ký thông tin khách hàng.

c. Đăng ký thông tin khách hàng

```
@RequestMapping(value = "/DKuser")  
public ModelAndView userDK() {  
    ModelAndView mav = new ModelAndView("cart-pay/userInfo");  
    User user = new User();  
    mav.addObject("user", user);  
    return mav;  
}  
  
@PostMapping("/userDK1")  
public ModelAndView createUser(HttpSession  
session, @ModelAttribute("user") User user) {  
    ModelAndView mav = new ModelAndView("cart-pay/userInfo");  
    Account account = (Account) session.getAttribute("account")  
    Account account1 =  
accountImpl.getAccountById(account.getUsername());  
    mav.addObject("check", true);  
    String message = "";
```

```

    if (user.getSurname() == "") {
        message = "Họ không được để trống";
        mav.addObject("errorHo", message);
        return mav;
    }

    if (user.getName() == "") {
        message = "Tên không được để trống";
        mav.addObject("errorTen", message);
        return mav;
    }

    if (user.getAddress() == "") {
        message = "Địa không được để trống";
        mav.addObject("errorDC", message);
        return mav;
    }

    if (user.getSdt() == "") {
        message = "Số điện không được để trống";
        mav.addObject("errorSDT", message);
        return mav;
    }

    Long count = userRpsrt.countUserById()
    Long newcount = count+1;
    String newId = Long.toString(newcount);
    user.setId(newId);
    user.setAccount(account1);

    userImpl.save(user);
    message = "Đăng kí thông tin thành công";
    mav.addObject("message", message);
    mav.setViewName("redirect:/login");
    return mav;
}

```

Khách hàng điền các thông tin họ, tên, ngày sinh, địa chỉ vào ô. Nếu ô thông tin bị bỏ trống sẽ báo lỗi trừ ô ngày sinh. Sau khi điền đầy đủ thông tin thì

đăng ký thông tin thành công dữ liệu được lưu vào bảng trong cơ sở dữ liệu. Sau khi đăng ký thành công trả về trang đăng nhập , khách hàng đăng nhập tài khoản vừa đăng ký để mua sắm.

d. Tìm kiếm:

```
@GetMapping("/search")

    public ModelAndView
searchProductsByName(@RequestParam("search") String name) {

    List<Product> productList =
productService.searchProductByName(name);

    ModelAndView mav = new ModelAndView("user/product");

    mav.addObject("listProduct", productList);

    return mav;

}
```

Khi khách hàng vào trang sản phẩm, mục tìm kiếm sẽ được hiển thị, khách hàng tiến hành nhập vào ô tìm kiếm thông tin về sản phẩm mà mình muốn tìm kiếm, sau đó nhấn tìm kiếm các sản phẩm liên quan sẽ hiển thị lên trang sản phẩm để khách hàng lựa chọn.

e. Lọc:

* Lọc theo doanh mục sản phẩm:

```
@RequestMapping("/product")
    public ModelAndView
getProductListByCategoryId(@RequestParam(value = "categoryId", required =
false) String id) {
    List<Product> productList = new ArrayList<>();

    if (id == null) {
        productList = productService.getListProduct();
        System.out.println("AAa");
    } else {
        productList = productService.getProductByCategoryId(id);
        System.out.println("DD");
    }

    ModelAndView mav = new ModelAndView("user/product");
    mav.addObject("listProduct", productList);
}
```

```

    }
    return mav;
}

```

Khách hàng vào trang chủ, nó hiển thị các doanh mục sản phẩm, khách hàng chỉ cần nhấn vào doanh mục sản phẩm mà mình muốn lựa chọn. Khi đó dựa vào id của doanh mục sản phẩm mà các sản phẩm có trong doanh mục đó sẽ hiển thị lên trên khung sản phẩm cho khách hàng lựa chọn.

* Lọc theo giá tiền:

```

@GetMapping("/findbycenterprice")
public ModelAndView filterByPriceRange(@RequestParam("price1")
double price1,
                                     @RequestParam("price2") double price2) {
    List<Product> productList = productService.filterByPriceRange(price1,
price2);
    ModelAndView mav = new ModelAndView("user/product");
    mav.addObject("listProduct", productList);
    return mav;
}

@GetMapping("/findprice")
public ModelAndView filterByPrice(@RequestParam("price") double
price) {
    List<Product> productList = productService.filterByPrice(price);
    ModelAndView mav = new ModelAndView("user/product");
    mav.addObject("listProduct", productList);
    return mav;
}

```

Khách hàng vào trang sản phẩm, lựa chọn giá tiền phù hợp các sản phẩm có giá tiền tương ứng sẽ được hiển thị trong trang sản phẩm để khách hàng lựa chọn.

* Lọc theo giá tiền tăng dần, giảm dần, lọc theo tên từ A-Z, Z-A;

```

@GetMapping("/price")
public ModelAndView
getProductsBySortType(@RequestParam("sortType") String sortType) {
    List<Product> products = new ArrayList<>();
    if (sortType.equals("price-asc")) {
        products = productService.getProductsOrderByPriceAsc();
    } else if (sortType.equals("price-desc")) {
        products = productService.getProductsOrderByPriceDesc();
    }
}

```

```

    } else if (sortType.equals("name-asc")) {
        products = productService.getProductsOrderByNameAsc();
    } else if (sortType.equals("name-desc")) {
        products = productService.getProductsOrderByNameDesc();
    }
    ModelAndView mv = new ModelAndView("user/product");
    mv.addObject("listProduct", products);
    return mv;
}

```

Khách hàng vào trang sản phẩm chọn vào mục lọc theo giá tiền tăng dần, giảm dần hay lọc theo tên từ A-Z, Z-A, các sản phẩm sẽ được sắp xếp theo lựa chọn mà khách hàng đã chọn.

f. Trang thông tin chi tiết sản phẩm;

*** Xem chi tiết sản phẩm:**

```

@RequestMapping (value = "/product_detail")
public ModelAndView product_detail(@RequestParam(value="id",
required=false) String id) {
    ModelAndView mv = new ModelAndView("user/product_detail");
    List<Product> listProduct = new ArrayList<>();
    listProduct = productService.getListProduct();
    mv.addObject("listProduct", listProduct);

    Product product = productService.getProductById(id);
    mv.addObject("product", product);
    return mv;
}
}

```

Khi nhấn vào một sản phẩm bất kỳ nó sẽ chuyển đến trang chi tiết sản phẩm, dựa vào mã sản phẩm của sản phẩm đó mà các thông tin liên qua đến sản phẩm sẽ được hiển thị ra ngoài giao diện của trang chi tiết sản phẩm để khách hàng hiểu thêm về sản phẩm đó.

*** Ép sản phẩm vào giỏ hàng:**

```

@RequestMapping("/addProduct")
public ModelAndView cart(HttpSession session, @RequestParam(name
="id") String id) {
    ModelAndView mv = new ModelAndView("cart-pay/cart");
    Account account = (Account) session.getAttribute("account");
}

```

```

        if (account == null) {
            // Xử lý khi không có tài khoản, ví dụ: chuyển hướng đến trang đăng
            nhập, thông báo lỗi, vv.
            // Ở đây tôi sẽ chuyển hướng về trang đăng ký
            return new ModelAndView("redirect:/loginUS");
        }

        Account account1 =
accountServiceImpl.getAccountByld(account.getUsername());
        User user = account1.getUser();

        Cart cart;
        List<Cart> carts = cartservice.getCartByUser(user);
        if (carts.isEmpty()) {
            cart = new Cart();
            User user1 = userSV.getUserByld(user.getId());
            Long cartCount = cartRpstr.countCartByld();
            Long newCount = cartCount + 1;
            String count = newCount.toString();
            cart.setld(count);
            cart.setUser(user1);
            cartRpstr.save(cart);
            System.out.println("Giỏ hàng vừa tạo: " + count);
        } else {
            cart = carts.get(carts.size() - 1); // Lấy giỏ hàng cuối cùng trong danh
            sách giỏ hàng
        }

        System.out.println("Username:" + account.getUsername());
        String a = account.getUsername(); // Đã thêm kiểu dữ liệu String
        Account account11 = accountServiceImpl.getAccountByld(a);

        Product product = productService.getProductByld(id);

        CardDetailPk ids = new CardDetailPk(cart.getld(), product.getld());
        int count = 1;

        CardDetail existingCardDetail =
cardDetailService.getCardDetailBylds(ids);
        if (existingCardDetail != null) {
            // Sản phẩm đã tồn tại trong giỏ hàng, thông báo đã có sản phẩm
            String message = "Sản phẩm đã tồn tại trong giỏ hàng.";

```

```

        mv.addObject("message", message); // Thêm message vào
        ModelAndView
        System.out.println("Thông báo đã được đặt: " + message);
    } else {
        CardDetail cardDetail = new CardDetail(ids, count, product, cart);
        cartDetailService.save(cardDetail);
    }
    List<CardDetail> listcart =
    cart_DetailRepository.findById(cart.getId());
    mv.addObject("listCart", listcart);

    return mv;
}

```

Khi đã chọn được sản phẩm phù hợp, người dùng tiến hành chọn vào nút thêm sản phẩm vào giỏ hàng, khi đó sản phẩm được chọn sẽ được thêm vào chi tiết giỏ hàng với id giỏ hàng tương ứng. Đồng thời nó sẽ chuyển hướng đến trang giỏ hàng của khách hàng, dựa vào id của sản phẩm mà thông tin về sản phẩm được chọn sẽ hiện thị trên trang giao diện giỏ hàng của khách hàng.

g. Cập nhật thông tin khách hàng

```

@PostMapping("/profile/update")
public ModelAndView updateProfile(@ModelAttribute("account1") User
user,
                                @RequestParam("avatar") MultipartFile avatar,
                                @RequestParam("ids") String ids) {
    ModelAndView mav = new ModelAndView();
    user.setId(ids);
    user.setImage(avatar.getOriginalFilename());

    System.out.println(user.getId());
    userService.updateUser(user);

    String message = "";
    mav.setViewName("/user/profile");
    try {
        MultipartFile multipartFile = avatar;
        String fileName = multipartFile.getOriginalFilename();
        File file = new File(
            "D:\\eclipseWorkspace\\thuctapcoso\\PhoneAccessories\\src\\main\\webap
p\\templates\\user\\img",

```



```

        fileName);
        multipartFile.transferTo(file);
        message = "Thêm thành công";
        mav.addObject("message", message);
    } catch (Exception e) {
        e.printStackTrace();
        mav.addObject("message", "Lưu file ảnh thất bại");
    }

    Account account =
accountService.getAccountById(SecurityUtils.getPrincipal().getUsername());
    mav.addObject("account1", account.getUser());
    mav.addObject("order", account.getUser().getListOther());

    mav.addObject("account", account);

    mav.setViewName("/user/profile");

    return mav;
}

```

h. Tình trạng đơn hàng

```

@GetMapping("/profile")
    public ModelAndView showProfile() {ss
        ModelAndView mav = new ModelAndView("user/profile");
        Account account =
accountService.getAccountById(SecurityUtils.getPrincipal().getUsername());
        mav.addObject("account1", account.getUser());
        mav.addObject("order", account.getUser().getListOther());

        mav.addObject("account", account);

        return mav;
    }
}

```