**BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG**

**TRƯỜNG ĐH TÀI NGUYÊN VÀ MÔI TRƯỜNG TP. HCM**

**KHOA HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**

Ảnh có chứa biểu đồ

Mô tả được tạo tự động

**BÁO CÁO ĐỒ ÁN MÔN LẬP TRÌNH HĐT**

**QUẢN LÝ KHO QUẦN ÁO**

Thành viên của nhóm:

1. Giang Thị Sông Hằng(1050080133)
2. Phạm Thùy Trang (1050080159)
3. Trần Lệ Uyên (1050080165)

Lớp: 10 ĐH\_CNTT4

***TP. Hồ Chí Minh, tháng 03 năm 2023***

# **Giới thiệu đồ án**

Ngành công nghệ thông tin đang phát triển khá mạnh mẽ được ứng dụng khá nhiều trong đời sống hiện nay. Việc tin học hóa vào đời sống giúp không gian lưu trữ tránh được thất lạc dữ liệu, tự động hệ thống hóa và cụ thể hóa theo nhu cầu. Và nền kinh tế đang phát triển như hiện nay, nhiều ngành nghề đang nổi trổi như bán hàng, thời trang, shop quần áo,…

Việc bán hàng cần lưu thông tin nhiều như các công ty xí nghiệp đều cần phải quản lí kho hàng hóa một cách chuẩn xác, hỗ trợ quá trình bán hàng, kiểm soát được nguồn hàng, giúp tiết kiệm thời gian và hiệu quả hơn. Việc quản lí kho còn giúp tránh thất thoát trong quá trình bán hàng.

Chính vì vậy tin học hóa trong lĩnh vực quản lí cửa hàng bán quần áo là một yêu cầu tất yếu và cần thiết muốn quản lí tốt cần có được các phần mềm tốt, phần mền phải đảm bảo được độ bảo mật cao, dễ sử dụng và nhiều tiện ích. Và đề tài “Quản lí kho quần áo” là một yêu cầu cần thiết của mọi cửa hàng, phần mềm quản lí kho quần áo nó được sử dụng theo dõi và kiểm soát những vấn đề như nhập kho, xuất kho, vận chuyển kho nội bộ, theo dõi số lượng hàng hóa hiện có.

**Mục Lục**

[**Giới thiệu đồ án** 2](#_Toc130808329)

[**II.Phân tích đề tài :** 5](#_Toc130808330)

[*1.1 Phân tích yêu cầu của hệ thống:* 5](#_Toc130808331)

[*1.2 Yêu cầu chức năng của hệ thống:* 5](#_Toc130808332)

[*1.3 Yêu cầu phi chức năng:* 6](#_Toc130808333)

[*1.4 Các dữ liệu:* 7](#_Toc130808334)

[**III. Xác định các lớp của bài toán** 7](#_Toc130808335)

[**1. Class KhoHang** 7](#_Toc130808336)

[**2. Class NhapHang kế thừa từ class KhoHang** 7](#_Toc130808337)

[**3. Class XuatHang kế thừa từ class KhoHang** 7](#_Toc130808338)

[**4. Class HangTon kế thừa từ class KhoHang** 8](#_Toc130808339)

[**5. Class QuayBan kế thừa từ class XuatHang** 8](#_Toc130808340)

[**IV. Mô tả thuật toán thao tác** 8](#_Toc130808341)

[**1. Chỉnh sửa thông tin Nhập, xuất, tồn, quầy:** 8](#_Toc130808342)

[**2. Hiện ra các danh sách quần áo trong kho dựa vào mã hàng** 8](#_Toc130808343)

[**3. Kiểm tra mạt hàng đó có tồn tại không:** 9](#_Toc130808344)

[**V. Mô hình UML** 9](#_Toc130808345)

[**VI. Cài đặt bài toán** 9](#_Toc130808346)

[**1.** **Class KhoHang** 9](#_Toc130808347)

[**2.** **Class NhapHang** 11](#_Toc130808348)

[**3. Class XuatHang** 12](#_Toc130808349)

[**4.** **Class HangTon** 13](#_Toc130808350)

[**5.** **Class QuayBan** 14](#_Toc130808351)

[**6.** **Class Danhsachkhoquanao** 15](#_Toc130808352)

[**6.Class Main** 17](#_Toc130808353)

[**VII. Kiểm thử lập bộ test** 18](#_Toc130808354)

[Dữ liệu Mặt Hàng Trong Kho 18](#_Toc130808355)

[Dữ liệu Nhap của Kho 18](#_Toc130808356)

[Nhập lựa chọn và thông tin sản phẩm 20](#_Toc130808357)

[Nhập lựa chọn kiểm tra sản phẩm 20](#_Toc130808358)

[Nhập thông tin kiểm tra sản phẩm còn hay không 21](#_Toc130808359)

[Nhập lựa chọn xuất danh sách 21](#_Toc130808360)

[Nhập lựa chọn xóa mặt hàng ra khỏi danh sách 22](#_Toc130808361)

# **II.Phân tích đề tài :**

## *1.1 Phân tích yêu cầu của hệ thống:*

Đối tượng sử dụng hệ thống :

+  Người quản trị hệ thống

+ Người bán hàng

## *1.2 Yêu cầu chức năng của hệ thống:*

.Quản lý kho:

***Quản lý nhập*** :Sau mỗi lần lấy hàng bộ phận quản lí kho sẽ ghi lại chi tiết hàng hóa xuất vào,để tiện theo dõi hàng hóa trong kho.Khi mặt hàng nào đó sắp hết hoặc đã hết thì bộ phận quản lí kho sẽ tiến hanh báo cáo cho ban quản lí .Khi hàng đã được mua về kho ,bộ phận kho sẽ nhận phiếu giao hàng và kiểm tra số lượng và chất lượng xuất phiếu nhập kho.

Các công việc cần làm:

+Nhập hàng theo đung danh mục và số lượng trong hóa đơn mua hàng của cửa hàng bán quần áo.

+ Hàng nhập : được theo dõi dựa trên mã hàng,nhóm hàng,số lượng,đơn giá ,chất liệu ,nơi nhập,ngày nhập hàng,thành tiền ,tổng cộng số lượng.Số hàng nhận này có sự xác nhận của bên người quản lí kiểm đếm hàng kho.

+In ra báo cáo hàng nhập trong kỳ.

***Quản lý hàng xuất***  :Bộ phận này nhận yêu cầu mua hàng của khách hàng .Khách ở đây có thể mua lẻ hoặc mua sỉ.Bộ phận này có chức năng nhận yêu cầu và gửi đến bộ phận kho .Bộ phận kho sẽ kiểm tra số lượng hàng hóa có đủ đáp ứng hay không và thông báo lại cho bộ phận bán hàng và xuất phiếu xuất kho.Khi xuất hàng thông tin về mặt hàng xuất phải được đối chiếu kiểm tra.

Các công việc cần làm:

+Hàng xuất ra quầy được theo dõi qua mã hàng ,nhóm hàng ,chất liệu ,số lượng, ngày xuất

 +In báo cáo số lượng hàng xuất

***Quản lý hàng tồn***:Việc quản lí hàng tồn diễn ra ngay trong quá trinh nhập và xuất hàng.Hàng hóa được thực hiện qua máy tính mỗi khi phát sinh nhập hay xuất hàng.

Các công việc cần làm:

  + Tổng hợp những phát sinh xuất nhập làm tồn kho

   + in báo cáo số lượng tồn kho trong kỳ

***Quản lý quầy***:Bộ phận này có chức năng bán hàng tại các chi nhánh , mỗi cuối ca sẽ tổng hợp số lượng các mặt hàng đã bán được và nhập vào máy tính ,điều chỉnh các mặt hàng chênh lệch số liệu với nhau,và xuất hóa đơn cho khách hàng.

Các công việc cần làm:

\_Sổ giao ca:

+Số lượng tồn đầu ca và số lượng bán trong ca,cuối mỗi ca sẽ bàn giao cho người ca sau.

+Điều chỉnh số lượng tồn nếu phát hiện số lượng tồn thực tế chênh lệch với số lượng tồn của ca trước .

## *1.3 Yêu cầu phi chức năng:*

- An ninh: Hệ thống phải được bảo mật khỏi sự truy cập trái phép.

- Hiệu suất: Tốc độ xử lý nhập xuất hàng, giao dịch của hệ thống nhanh, đưa sản phẩm tới khách hàng trong thời gian ngắn nhất có thể.

- Khả năng mở rộng: Hệ thống phải có thể tăng hoặc giảm quy mô khi cần thiết.

- Khả dụng: Mức độ sử dụng được và làm hài lòng người sử dụng, phù hợp với nhu cầu.

- Tính di động: Hệ thống phải có thể chạy trên các nền tảng khác nhau.

- Khả năng tương thích: Hệ thống phải tương thích với các hệ thống khác.

- Tuân thủ: Hệ thống phải tuân thủ tất cả các luật và quy định hiện hành.

- Phục vụ cho công cụ tìm kiếm, thống kê, xây dựng, báo cáo nhập xuất sản phẩm

## *1.4 Các dữ liệu:*

**Quản lí kho**: Mã hàng, nhóm hàng, Số lượng, Chất liệu,Đơn giá

**Quản lí nhập**: Ngày nhập, Thành tiền, mã hàng

**Quản lí xuất**: Ngày xuất, Thành tiền, Mã hàng

**Quản lí tồn**: Tiền thiệt hại, Mã hàng,

**Quản lí quầy**: Hoá đơn, Ngày trả, Lí do đổi trả

# **III. Xác định các lớp của bài toán**

## **1. Class KhoHang**

∙ Thuộc tính: String maHang, nhomhang, chatLieu; Double soLuong,donGia

∙ Phương thức:

⮚ Constructor: Hàm tạo có đối số và hàm tạo không đối số

⮚ Getter, Setter: Phương thức get, set cho các thuộc tính kiểu private

⮚ Nhap: nhập thông tin các thuộc tính

⮚ Hien: Hiện ra các thông tin

## **2. Class NhapHang kế thừa từ class KhoHang**

∙ Thuộc tính: Date ngayNhap; Double thanhTien; String matHang

 ∙ Phương thức:

⮚ Nhap: override nhap() ở class Kho

⮚ Hien: override hien() ở class Kho, hiện thông tin Nhập

⮚ ngayNhap các ngày nhập hàng, gọi hàm ngayNhap từ class Nhaphang

⮚ thanhTien tiền nhập hàng, gọi hàm thanhTientừ class NhapHang

⮚ matHang: tên loại sản phẩm gọi hàm matHang từ class NhapHang

## **3. Class XuatHang kế thừa từ class KhoHang**

∙ Thuộc tính: Date ngayXuat; Double thanhTien; String: matHang

∙ Phương thức:

⮚ Nhap: override nhap() ở class Kho

⮚ Hien: override hien() ở class Kho, hiện thông tin Xuất

⮚ ngayXuat các ngày nhập hàng, gọi hàm ngayXuat từ class NhapHang

⮚ thanhTien: tiền nhập hàng, gọi hàm thanhTien từ class NhapHang

⮚ matHang: : tên loại sản phẩm gọi hàm matHang từ class NhapHang

## **4. Class HangTon kế thừa từ class KhoHang**

∙ Thuộc tính: Double tienThietHai, String matHang

∙ Phương thức:

⮚ Nhap: override phương thức nhap() ở class KhoHang

⮚ Hien: override phương thức hien() ở class KhoHang

## **5. Class QuayBan kế thừa từ class XuatHang**

∙ Thuộc tính: String hoaDon, Date ngayBan, String liDoDoiTra

∙ Phương thức:

⮚ Nhap: override phương thức nhap() ở class KhoHang

⮚ Hien: override phương thức hien() ở class KhoHang

# **IV. Mô tả thuật toán thao tác**

**1. Chỉnh sửa thông tin Nhập, xuất, tồn, quầy:**

Thuật toán:

∙ Nhập vào maHang/nhomHang/chatLieu/donGia/soLuong

∙ Dùng vòng lặp và câu lệnh so sánh, nếu từ khóa truyền vào trùng với dữ liệu đã có sẽ hiện thị ra để thao tác chỉnh sửa, không có sẽ không thông bảo  gì

∙ Thêm: Nhập vào thông tin đối tượng mới

∙ Xóa: Trong ArrayList dùng phép remove

**2. Hiện ra các danh sách quần áo trong kho dựa vào mã hàng**

Thuật toán:

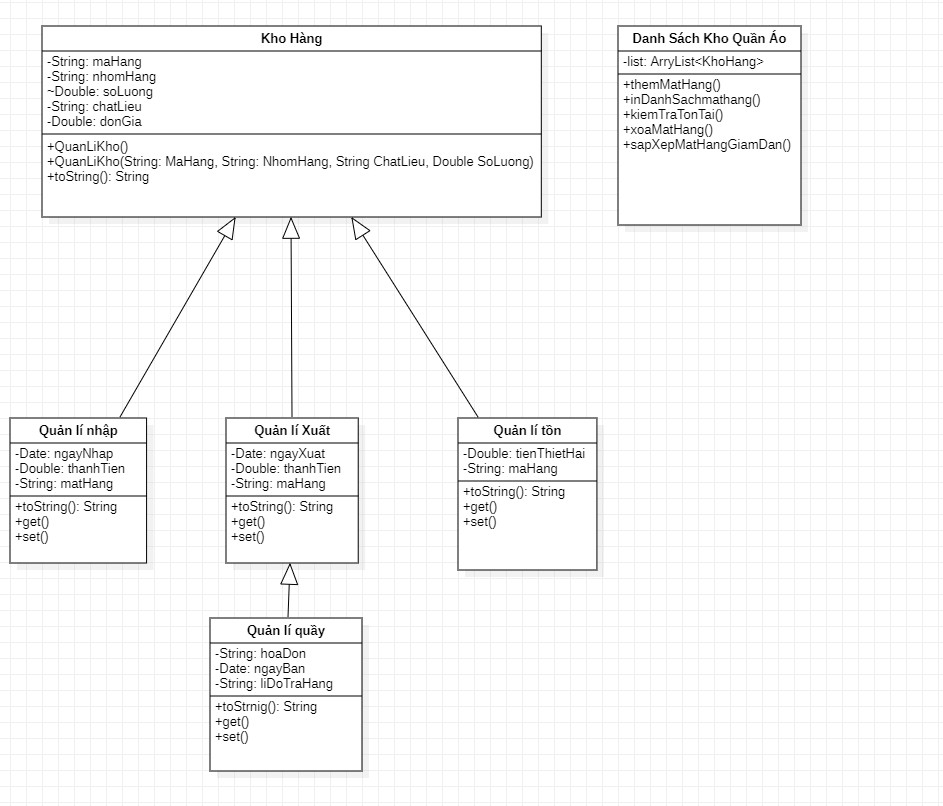
∙ Trong ArrayList đã nhập, tìm kiếm và hiện ra những thí sinh có điểm chuẩn  và mã ngành trùng với thông tin nhập vào.

## **3. Kiểm tra mạt hàng đó có tồn tại không:**

Thuật toán:

.Dùng thuật toán Contains

# **V. Mô hình UML**



# **VI. Cài đặt bài toán**

* 1. **Class KhoHang**

package main;

import java.util.Objects;

import java.util.Objects;

public class KhoHang implements Comparable{

private String maHang;

private String nhomHang;

private String chatLieu;

private Double soLuong;

private Double donGia;

public KhoHang(String maHang,String nhomHang, String chatLieu,Double soLuong,Double donGia) {

this.maHang=maHang;

this.nhomHang=nhomHang;

this.chatLieu=chatLieu;

this.soLuong=soLuong;

this.donGia=donGia;

}

public String getMaHang() {

return maHang;

}

public void setMaHang(String maHang) {

this.maHang = maHang;

}

public String getNhomHang() {

return nhomHang;

}

public void setNhomHang(String nhomHang) {

this.nhomHang = nhomHang;

}

public String getChatLieu() {

return chatLieu;

}

public void setChatLieu(String chatLieu) {

this.chatLieu = chatLieu;

}

public Double getSoLuong() {

return soLuong;

}

public void setSoLuong(Double soLuong) {

this.soLuong = soLuong;

}

public Double getDonGia() {

return donGia;

}

public void setDonGia(Double donGia) {

this.donGia = donGia;

}

@Override

public String toString() {

return "KhoHang["+"MaHang:'"+maHang+'\''+",NhomHang:'"+nhomHang+'\''+",chatLieu:'"+chatLieu+'\''+",SoLuong:'"+soLuong+'\''+",DonGia:'"+donGia+'\''+']';

}

public int compareTo(KhoHang o) {

return this.maHang.compareTo(maHang);

}

@Override

public int hashCode() {

return Objects.hash(nhomHang,chatLieu,donGia,soLuong);

}

@Override

public boolean equals(Object obj) {

if (this == obj)

return true;

if (obj == null)

return false;

if (getClass() != obj.getClass())

return false;

KhoHang other = (KhoHang) obj;

return Objects.equals(maHang, other.maHang);

}

}

* 1. **Class NhapHang**

package main;

import java.util.Date;

public class NhapHang extends KhoHang {

private Date ngayNhap;

private Double thanhTien;

private String matHang;

public NhapHang(String maHang, String nhomHang, String chatLieu, Double soLuong, Double donGia, Date ngayNhap,

Double thanhTien, String matHang) {

super(maHang, nhomHang, chatLieu, soLuong, donGia);

this.ngayNhap = ngayNhap;

this.thanhTien = thanhTien;

this.matHang = matHang;

}

public Date getNgayNhap() {

return ngayNhap;

}

public void setNgayNhap(Date ngayNhap) {

this.ngayNhap = ngayNhap;

}

public Double getThanhTien() {

return thanhTien;

}

public void setThanhTien(Double thanhTien) {

this.thanhTien = thanhTien;

}

public String getMatHang() {

return matHang;

}

public void setMatHang(String matHang) {

this.matHang = matHang;

}

@Override

public String toString() {

return "NhapHang" +super.toString()+"\nNgayNhap:"+this.ngayNhap+"\nThanhTien:"+this.thanhTien+"\nMatHang:"+this.matHang;

}

}

## **3. Class XuatHang**

import java.util.Date;

public class XuatHang extends KhoHang {

private Date ngayXuat;

private Double thanhTien;

private String matHang;

public XuatHang(String maHang, String nhomHang, String chatLieu, Double soLuong, Double donGia, Date ngayXuat,

Double thanhTien, String matHang) {

super(maHang, nhomHang, chatLieu, soLuong, donGia);

this.ngayXuat = ngayXuat;

this.thanhTien = thanhTien;

this.matHang = matHang;

}

public Date getNgayXuat() {

return ngayXuat;

}

public void setNgayXuat(Date ngayXuat) {

this.ngayXuat = ngayXuat;

}

public Double getThanhTien() {

return thanhTien;

}

public void setThanhTien(Double thanhTien) {

this.thanhTien = thanhTien;

}

public String getMatHang() {

return matHang;

}

public void setMatHang(String matHang) {

this.matHang = matHang;

}

@Override

public String toString() {

return "NhapHang" +super.toString()+"\nNgayXuat:"+this.ngayXuat+"\nThanhTien:"+this.thanhTien+"\nMatHang:"+this.matHang;

}

}

1. **Class HangTon**

package main;

public class HangTon extends KhoHang {

private Double tienThietHai;

private String maHang;

public HangTon(String maHang, String nhomHang, String chatLieu, Double soLuong, Double donGia, Double tienThietHai,

String maHang2) {

super(maHang, nhomHang, chatLieu, soLuong, donGia);

this.tienThietHai = tienThietHai;

maHang = maHang2;

}

public Double getTienThietHai() {

return tienThietHai;

}

public void setTienThietHai(Double tienThietHai) {

this.tienThietHai = tienThietHai;

}

public String getMaHang() {

return maHang;

}

public void setMaHang(String maHang) {

this.maHang = maHang;

}

@Override

public String toString() {

return "HangTon:" +super.toString()+"\n TienThietHai:"+this.tienThietHai+"\nMaHang"+this.maHang;

}

}

1. **Class QuayBan**

package main;

import java.util.Date;

public class QuayBan extends XuatHang{

private String hoaDon;

private Date ngayBan;

private String liDoTraHang;

public QuayBan(String maHang, String nhomHang, String chatLieu, Double soLuong, Double donGia, Date ngayXuat,

Double thanhTien, String matHang, String hoaDon, Date ngayBan, String liDoTraHang) {

super(maHang, nhomHang, chatLieu, soLuong, donGia, ngayXuat, thanhTien, matHang);

this.hoaDon = hoaDon;

this.ngayBan = ngayBan;

this.liDoTraHang = liDoTraHang;

}

public String getHoaDon() {

return hoaDon;

}

public void setHoaDon(String hoaDon) {

this.hoaDon = hoaDon;

}

public Date getNgayBan() {

return ngayBan;

}

public void setNgayBan(Date ngayBan) {

this.ngayBan = ngayBan;

}

public String getLiDoTraHang() {

return liDoTraHang;

}

public void setLiDoTraHang(String liDoTraHang) {

this.liDoTraHang = liDoTraHang;

}

@Override

public String toString() {

return "QuayHang:" +super.toString()+"\n HoaDon:"+this.hoaDon+"\nNgayBan"+this.ngayBan+"\nLiDoTraHang"+this.liDoTraHang;

}

}

1. **Class Danhsachkhoquanao**

package main;

import java.util.ArrayList;

import java.util.Collection;

import java.util.Collections;

import java.util.Comparator;

public class Danhsachkhoquanao {

private ArrayList<KhoHang> danhSach;

public Danhsachkhoquanao() {

this.danhSach=new ArrayList<KhoHang>();

}

public Danhsachkhoquanao(ArrayList<KhoHang> danhSach) {

this.danhSach = danhSach;

}

public void themMatHang(KhoHang kh) {

this.danhSach.add(kh);

}

public void inDanhsachmathang() {

for(KhoHang khohang: danhSach ) {

System.out.println(khohang);

}

}

public boolean kiemTraTonTai(KhoHang kh) {

return this.danhSach.contains(kh);

}

public boolean xoaMatHang(KhoHang kh) {

return this.danhSach.remove(kh);

}

public void sapXepMatHangGiamDan() {

Collections.sort(this.danhSach, new Comparator<KhoHang>() {

public int compare(KhoHang kh1, KhoHang kh2) {

if(kh1.getDonGia()<kh2.getDonGia()) {

return -1;

}else if(kh1.getDonGia()>kh2.getDonGia()){

return 1;

}else {

return 0;

}

}

});

}

}

## **6.Class Main**

package main;

import java.util.Scanner;

public class test {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

Danhsachkhoquanao dskqa=new Danhsachkhoquanao();

int luaChon=0;

do {

System.out.println("------Menu-----");

System.out.println("Chon chuc nang");

System.out.println(

"1.Them mat hang vao kho.\n"

+ "2.In danh sach co trong kho trong kho.\n"

+"3.Kiem tra mat hang co ton tai khong dua tren ma hang.\n"

+"4.Xoa mot mat hang ra khoi danh sach.\n"

+"5.Xuat ra danh sach mat hang giam dan theo don gia.\n"

+"0.Thoat chuong trinh" );

luaChon=sc.nextInt();

sc.nextLine();

if(luaChon==1) {

System.out.println("Nhap ma hang:");String maHang=sc.nextLine();

System.out.println("Nhap nhom hang:");String nhomHang=sc.nextLine();

System.out.println("Nhap chat ieu:");String chatLieu=sc.nextLine();

System.out.println("Nhap don gia");Double donGia=sc.nextDouble();

System.out.println("Nhap so luong:");Double soLuong=sc.nextDouble();

KhoHang kh=new KhoHang(maHang, nhomHang, chatLieu, soLuong, donGia);

dskqa.themMatHang(kh);

}else if(luaChon==2) {

dskqa.inDanhsachmathang();

}else if(luaChon==3) {

System.out.println("Nhap ma hang: ");String maHang=sc.nextLine();

KhoHang kh= new KhoHang( maHang, null, null, null,null);

System.out.println("Kiểm tra : "+dskqa.kiemTraTonTai(kh));

}else if(luaChon==4) {

System.out.println("Nhap ma hang: ");String maHang=sc.nextLine();

KhoHang kh= new KhoHang(maHang, null, null, null, null);

System.out.println("Xóa mat hang ra khoi danh sach :"+dskqa.xoaMatHang(kh));

}else if(luaChon==5) {

dskqa.sapXepMatHangGiamDan();

dskqa.inDanhsachmathang();

}

} while (luaChon!=0);

}

# **VII. Kiểm thử lập bộ test**

# Dữ liệu Mặt Hàng Trong Kho

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã Hàng | Nhóm Hàng | Số Lượng | Chất Liệu | Đơn Giá |
| 101 | Áo | 50 | Cotton | 100.000 |
| 102 | Quần | 40 | Kaki | 200.000 |
| 103 | Đầm | 30 | Lụa | 200.000 |
| 104 | Hoodie | 40 | Nỉ Bông | 300.000 |
| 105 | Váy | 30 | Cotton | 100.000 |

## Dữ liệu Nhap của Kho

| Ngày Nhập | Đơn Giá  Nhập | Thành Tiền | Nhóm Hàng | Mặt Hàng | Chất Liệu | Mã Hàng | Số Lượng |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 01/03/2023 | 100.000 | 5.000.000 | Áo | Áo thun | Cotton | 101 | 50 |
| Áo sơ mi |
| 04/03/2023 | 200.000 | 8.000.000 | Quần | Quần tây | Kaki | 102 | 40 |
| Quần ống loe |
| 10/03/2023 | 200.000 | 6.000.000 | Đầm | Đầm suông | Lụa | 103 | 30 |
| Đầm maxi |
| 15/03/2023 | 300.000 | 12.000.000 | Hoodie | Hoodie | Nỉ Bông | 104 | 40 |
| Hooide zip |
| 20/03/2023 | 100.000 | 3.000.000 | Váy | Váy chữ A | Cotton | 105 | 30 |
| Váy dáng dài |

## Nhập lựa chọn và thông tin sản phẩm



## Nhập lựa chọn kiểm tra sản phẩm

Text

Description automatically generated

## Nhập thông tin kiểm tra sản phẩm còn hay không

Text

Description automatically generated

## Nhập lựa chọn xuất danh sách

Text

Description automatically generated

## Nhập lựa chọn xóa mặt hàng ra khỏi danh sách

