**BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG**

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

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



**BÁO CÁO ĐỒ ÁN MÔN LẬP TRÌNH HĐT**

**QUẢN LÝ KHO HOA TƯƠI**

Thành viên của nhóm:

1. Phạm Đức Phát (1050080151)
2. Trần Trọng Nhân (1050080148)
3. Lê Hoàng Phúc (1050080153)

Lớp: 10ĐH\_CNTT4

***TP. Hồ Chí Minh, 03 tháng 04 năm 2023***

MỤC LỤC

[1. Giới thiệu đồ án 3](#_Toc131358739)

[**1.1 Định nghĩa vấn đề** 3](#_Toc131358740)

[**1.2 Phạm vi của đồ án** 4](#_Toc131358741)

[**1.3 Mục tiêu và sự cần thiết của đề tài** 4](#_Toc131358742)

[2. Phân tích đề tài 4](#_Toc131358743)

[**2.1 Phân tích yêu cầu hệ thống** 4](#_Toc131358744)

[**2.1.1 Nhập hàng** 4](#_Toc131358745)

[**2.2.2 Bán hàng** 5](#_Toc131358746)

[**2.2 Yêu cầu chức năng** 5](#_Toc131358747)

[**2.2.1 Menu** 5](#_Toc131358748)

[**2.2.2 Nhập kho** 5](#_Toc131358749)

[**2.2.3 Xuất kho** 5](#_Toc131358750)

[**2.3 Yêu năng cầu phi chức năng** 5](#_Toc131358751)

[**2.4 Công việc cần giải quyết** 5](#_Toc131358752)

[3. Thiết kế 6](#_Toc131358753)

[**3.1 Mô hình các lớp** 6](#_Toc131358754)

[**3.1.1 Class Diagram** 6](#_Toc131358755)

[**3.1.2 Activity Diagram** 7](#_Toc131358756)

[**3.1.3 Sequence Diagram** 8](#_Toc131358757)

[**3.1.4 State Chart Diagram** 9](#_Toc131358758)

[**3.2 Xác định các lớp của đồ án** 9](#_Toc131358759)

[**3.2.1 Class KhoHoaTuoi** 9](#_Toc131358760)

[**3.2.2 Class Nhaphoa** 9](#_Toc131358761)

[**3.2.3 Class Xuathoa** 10](#_Toc131358762)

[**3.2.4 Class Danhsachnhap** 10](#_Toc131358763)

[**3.2.5 Class Danhsachxuat** 10](#_Toc131358764)

[4. Cài đặt bài toán 10](#_Toc131358765)

[**4.1 Class KhoHoaTuoi** 11](#_Toc131358766)

[**4.2 Class Nhaphoa** 13](#_Toc131358767)

[**4.3 Class Xuathoa** 15](#_Toc131358768)

[**4.3 Class Danhsachnhap** 16](#_Toc131358769)

[**4.4 Class DanhsachXuat** 19](#_Toc131358770)

[**4.5 Class Main** 20](#_Toc131358771)

[5. Kiểm thử 26](#_Toc131358772)

[**5.1 Menu** 27](#_Toc131358773)

[**5.2 Thêm hoa** 27](#_Toc131358774)

[**5.3 In danh sách hoa** 27](#_Toc131358775)

[**5.4 Sắp xếp số lượng hoa** 28](#_Toc131358776)

[**5.5 Xóa hoa** 28](#_Toc131358777)

[**5.6 Tìm kiếm hoa theo mã** 28](#_Toc131358778)

[**5.7 Xem danh sách kho có rỗng hay không** 28](#_Toc131358779)

[**5.8 Làm rỗng danh sách** 28](#_Toc131358780)

[**5.9 Số lượng hoa tươi** 29](#_Toc131358781)

[**5.10 Nhập hoa vào danh sách xuất** 29](#_Toc131358782)

[**5.11 In danh sách xuất hoa** 29](#_Toc131358783)

[6. Kết luận và định hướng phát triển: 29](#_Toc131358784)

[7. Tài liệu tham khảo: 30](#_Toc131358785)

**LỜI MỞ ĐẦU**

Trong thời đại ngày nay, công nghệ thông tin phát triển không ngừng không chỉ trên thế giới mà ngay cả đất nước ta công nghệ cũng đang là một ngành khá quan trọng và là một phần không thể thiếu trong cuộc sống thường ngày của con người chúng ta. Việc đưa công nghệ vào quản lý các công việc thường ngày cũng đã áp dụng rất thành công và thay thế con người giúp cho công việc quản lý dễ dàng hơn bao giờ hết, giảm bớt gánh nặng về thời gian, quản lý và sắp xếp công việc hợp lý, tạo ra hiệu quả cao và chất lượng công việc tốt hơn.

Thế mạnh của công nghệ thông tin ngày càng được chứng tỏ khi nó tham gia vào đa số các công việc phổ biến hiện nay như: quản lý nhân sự, quản lý lương bổng, quản lý thư viện,... Trong số đó không thể không kệ đến quản lý kho hoa tươi, đó là cách tối ưu để tiết kiệm thời gian và có thể xử lý chính xác nhất từng số liệu của việc nhập xuất mua bán cho đến giá cả mà khi quản lý thủ công rất dễ mắc phải sai sót. Vì vậy việc xây dựng một phần mềm quản lý có thể giúp chúng ta cập nhật được những thông tin mới nhất về từ các mặt hàng, sản phẩm hiện có trên thị trường, sự biến động về giá cả lẫn thuế giá trị gia tăng của từng sản phẩm đó. Phần mềm cũng là sự ghép nối giúp người bán và người mua từ đó giúp chúng ta quản lý tốt hơn những khách hàng đã nhiều lần mua hoa của tiệm. Ngoài ra, thay vì thống kê doanh thu bằng cách thủ công thì chúng ta có thể dựa vào số liệu có sẵn được máy vi tính lưu trữ trong bộ nhớ để có thể thực hiện tính toán một cách dễ dàng, tránh những sai số thường xảy ra có thể ảnh hưởng đến lợi nhuận cũng như sự quản lý của nhân viên tại cửa hàng.

Vì vậy xây dựng một phần mềm quản lý kho hoa tươi là điều cần thiết để giúp bạn quản lý kho hàng của mình một cách dễ dàng, tiết kiệm thời gian và công việc của những người sử dụng phần mềm sẽ đạt hiệu quả tốt nhất có thể.

# Giới thiệu đồ án

## **1.1 Định nghĩa vấn đề**

Việc  xây dựng một phần mềm quản lý kho hàng là một điều tất yếu trong thời đại ngày nay khi mà các doanh nghiệp lớn nhỏ đang mọc lên như nấm, điều này không chỉ giúp chúng ta cải thiện được năng suất làm việc khi dùng công nghệ phần mềm thay cho sức lao động của con người. Đặt biệt thời gian gần đây nhiều tiểu doanh nghiệp hoa tươi  liên tục xuất hiện ở nhiều nơi, điều này càng khiến cho nhu cầu mua sắm tăng cao tạo nên sự đang dạng về sản phẩm hoa tươi, chất lượng cũng như giá cả biến động theo. Vì vậy việc xây dựng phần mềm quản lý kho hoa tươi có thể quản lý tốt các sản phẩm, hàng hóa cũng chi phí giá cả của mặt hàng này, quản lý chặt chẽ từng khâu nhập-bán của cửa hàng một cách thuận tiện, dễ dàng và chính xác nhất.

## **1.2 Phạm vi của đồ án**

Phần mềm “Quản lý kho hoa tươi” được xây dựng với mục đích thực hiện các yêu cầu quản lý hàng hóa, xuất xứ, số lượng, giá cả vào ra của từng loại hoa,... Đó là các vấn đề mà phần mềm khi hoàn thành có thể đáp ứng hết những nhu cầu từ thông tin trên.

## **1.3 Mục tiêu và sự cần thiết của đề tài**

Thời gian qua do nhu cầu mua sắm của chúng ta ngày càng tăng nên sự đòi hỏi đáp ứng nhu cầu về hàng hóa cũng như sản phẩm ngày càng đa dạng. Mặt khác, với sự xuất hiện với số lượng lớn của những doanh nghiệp hoa tươi. Vì vậy để có thể đáp ứng lại nhu cầu đó của các chủ thể kinh doanh hoa tươi, nên nhóm 6 quyết định chọn đề tài “Quản lý kho hoa tươi” để báo cáo đồ án cho môn Lập trình hướng đối tượng mục đích là để tạo ra một phần mềm quản lý đơn giản, dễ tiếp cận, dễ sử dụng và có thể tiết kiệm thời gian cho việc quản lý kho hoa tươi, hạn chế những sai số và có thể đáp ứng nhu cầu mà người dũng đang cần.

# Phân tích đề tài

## **2.1 Phân tích yêu cầu hệ thống**

Trong quá trình tham khảo thông tin các cửa hàng hoa bọn em đã thu thập được các các dữ liệu cần thiết. Việc quản lý kho hàng được dựa trên các nguyên tắc chung từ việc nhập hàng, bán hàng, tạo các báo cáo tổng kết theo định kỳ. Sau đây là các nguyên tắc quản lý:

### **2.1.1 Nhập hàng**

Cửa hàng được cung cấp hàng từ nhiều nhà cung cấp với nhiều loại hoa khác nhau như: hoa hồng, hoa tulip, hoa cúc, hoa cẩm tú cầu, … Khi nhận hàng từ nhà cung cấp nhân viên cửa hàng sẽ nhập thông tin hàng hóa như: tên hoa, màu sắc, xuất xứ, giá nhập, giá bán ra, số lượng hoa, và vài loại thuộc tính riêng biệt của từng loại hoa.

### **2.2.2 Bán hàng**

Cửa hàng sẽ hiển thị ra danh sách các loại hoa mà cửa hàng đang có ra màn hình cho khách hàng nhìn thấy các thông tin của các loại hoa như: tên hoa, màu hoa, xuất xứ, giá bán và số lượng của hoa còn trong cửa hàng.

## **2.2 Yêu cầu chức năng**

### **2.2.1 Menu**

Menu hiển thị ra các công việc mà người sử dụng có thể  thao tác được.

### **2.2.2 Nhập kho**

Nhập vào các thuộc tính của các loại hoa: mã hoa, tên hoa, màu sắc hoa, xuất xứ, giá nhập,số lượng, ngày nhập, Cho phép thực hiện các công việc như thêm hoa mới, in ra danh sách hoa, sắp xếp, xóa hoa, tìm kiếm, làm rỗng kho, và xe số lượng hoa có trong kho.

### **2.2.3 Xuất kho**

Cho phép nhận vào thông tin xuất kho đi như: mã hoa, tên hoa, màu, xuất xứ, giá xuất, số lượng. Và in ra danh sách những loại hoa được xuất đi cùng với từng thuộc tính.

## **2.3 Yêu năng cầu phi chức năng**

* Phần mềm dễ hiểu, dễ tiếp cận, dễ sử dụng.
* Thiết kế chỉn chu, gọn gàng, đảm bảo trình tự.
* Tính toán chuẩn xác số lượng, giá tiền, không xảy ra sai sót.
* Yêu cầu chạy nhanh, không bị lỗi,…

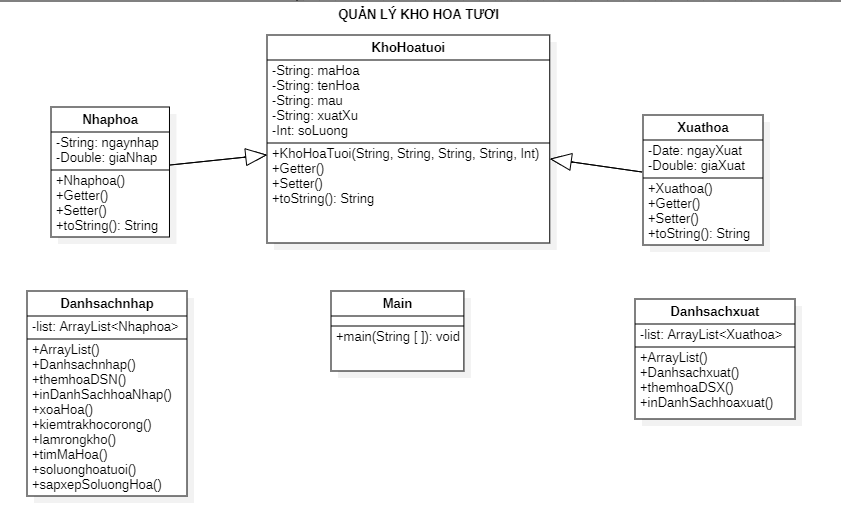
## **2.4 Công việc cần giải quyết**

* Thực hiện đủ các yêu cầu chức năng và phi chức năng, mô hình hóa các yêu cầu trên.
* Xác định Use case thiết kế các biểu đồ UML, biểu đồ quan hệ, xây dựng thuật toán cho mô hình.

# Thiết kế

## **3.1 Mô hình các lớp**

### **3.1.1 Class Diagram**



### **3.1.2 Activity Diagram**

**Diagram

Description automatically generated**

### **3.1.3 Sequence Diagram**

**Diagram

Description automatically generated**

### **Diagram Description automatically generated****3.1.4 State Chart Diagram**

## **3.2 Xác định các lớp của đồ án**

### **3.2.1 Class KhoHoaTuoi**

* **Thuộc tính:** string tenHoa, String mau, String xuatXu, int giaNhap.
* **Phương thức:**
  + Constructor: Hàm tạo có đối số và hàm tạo không đối số.
  + Getter, setter: phương thức get, set cho các thuộc tính.
  + String toString: phương thức in ra màn hình dưới dạng chuỗi.

### **3.2.2 Class Nhaphoa**

* **Thuộc tính**: string ngayNhap, double giaNhap.
* **Phương thức:**
  + Constructor: Hàm tạo có đối số và hàm tạo không đối số.
  + Getter, setter: phương thức get, set cho các thuộc tính.
  + String toString: @Override lại phương thức in ra màn hình dưới dạng chuỗi của class KhoHoaTuoi.

### **3.2.3 Class Xuathoa**

* **Thuộc tính:** string ngayXuat, double giaXuat.
* **Phương thức:**
  + Constructor: Hàm tạo có đối số và hàm tạo không đối số.
  + Getter, setter: phương thức get, set cho các thuộc tính.
  + String toString: @Override lại phương thức in ra màn hình dưới dạng chuỗi của class KhoHoaTuoi.

### **3.2.4 Class Danhsachnhap**

* **Thuộc tính:** ArrayList <Nhaphoa> nh.
* **Phương thức:**
* themhoaDSN(): thêm hoa vào trong danh sách kho.
* inDanhSachhoaNhap(): in ra danh sách hoa đã thêm vào từ phương thức trên.
* xoaHoa(): xoa một loại hoa trong kho ra khỏi danh sách.
* kiemtrakhocorong(): phương thức kiểm tra xem kho có đang rỗng hay không. Nếu hoa đang rỗng sẽ hiện false và ngược lại.
* lamrongkho():giúp xóa toàn bộ hoa có trong danh sách.
* timMaHoa():tìm hoa dựa theo mã nhập vào.
* soluonghoatuoi(): phương thức đếm tổng số hoa có trong danh sách.
* sapxepSoLuongHoa(): sắp xếp hoa theo số lượng giảm dần.

### **3.2.5 Class Danhsachxuat**

* **Thuộc tính:** ArrayList <Xuathoa> xh.
* **Phương thức**:
* themhoaDSX(): thêm các loại hoa cần xuất vào trong danh sách xuất.
* inDanhSachhoaxuat(): in ra danh sách hoa cần xuất đã thêm vào từ phương thức trên.

1. **Cài đặt bài toán**

## **4.1 Class KhoHoaTuoi**

class KhoHoatuoi {

private String maHoa;

private String tenHoa;

private String mau;

private String xuatXu;

private int soLuong;

public KhoHoatuoi(String maHoa){

this.maHoa=maHoa;

}

public KhoHoatuoi(String maHoa, String tenHoa, String mau, String xuatXu,int soLuong){

super();

this.maHoa=maHoa;

this.tenHoa = tenHoa;

this.mau=mau;

this.xuatXu=xuatXu;

this.soLuong=soLuong;

}

public String getMahoa() {

return maHoa;

}

public void setMaHoa(String maHoa) {

this.maHoa=maHoa;

}

public String getTenhoa() {

return tenHoa;

}

public void setTenHoa(String tenHoa) {

this.tenHoa=tenHoa;

}

public String getMau() {

return mau;

}

public void setMau(String mau) {

this.mau=mau;

}

public String getXuatXu() {

return xuatXu;

}

public void setXuatXu(String xuatXu) {

this.xuatXu=xuatXu;

}

public int getSoLuong() {

return soLuong;

}

public void setSoLuong(int soLuong) {

this.soLuong=soLuong;

}

public String toString(){

return "KhoHoatuoi["+" Ma hoa: "+ maHoa +

" Ten Hoa: "+tenHoa +

" Mau hoa: "+ mau +

" Xuat xu: " +xuatXu +

" So luong: "+soLuong +"]";

}

}

## **4.2 Class Nhaphoa**

class Nhaphoa extends KhoHoatuoi {

private String ngayNhap;

private double giaNhap;

public Nhaphoa(String maHoa){

super(maHoa);

}

public Nhaphoa(String maHoa, String tenHoa, String mau, String xuatXu, double giaNhap, int soLuong, String ngayNhap) {

super(maHoa,tenHoa, mau, xuatXu,soLuong);

this.ngayNhap = ngayNhap;

this.giaNhap=giaNhap;

}

public String getNgayNhap() {

return ngayNhap;

}

public void setNgayNhap(String ngayNhap) {

this.ngayNhap = ngayNhap;

}

public double getGiaNhap() {

return giaNhap;

}

public void setGiaNhap(double giaNhap) {

this.giaNhap=giaNhap;

}

@Override

public String toString() {

return "Nhap hoa:" +super.toString()+"\nGia nhap:"+this.giaNhap+"\nNgayNhap:"+this.ngayNhap;

}

}

## **4.3 Class Xuathoa**

class Xuathoa extends KhoHoatuoi {

private String ngayXuat;

private double giaXuat;

public Xuathoa(String maHoa, String tenHoa, String mau, String xuatXu,int soLuong, double giaXuat,

String ngayXuat){

super(maHoa, tenHoa, mau, xuatXu, soLuong);

this.giaXuat=giaXuat;

this.ngayXuat=ngayXuat;

}

public String getNgayXuat() {

return ngayXuat;

}

public void setNgayXuat(String ngayXuat) {

this.ngayXuat=ngayXuat;

}

public double getGiaxuat() {

return giaXuat;

}

public void setGiaXuat(double giaXuat) {

this.giaXuat=giaXuat;

}

@Override

public String toString() {

return "Thong tin xuat hoa:" +super.toString()+"\nNgayXuat:"+this.ngayXuat+"\nGia ban:"+this.giaXuat;

}

}

## **4.3 Class Danhsachnhap**

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.Scanner;

class Danhsachnhap{

private ArrayList<Nhaphoa> nh;

public Danhsachnhap(){

this.nh = new ArrayList<Nhaphoa>();

}

public Danhsachnhap(ArrayList<Nhaphoa> nh){

this.nh = nh;

}

public void themhoaDSN(Nhaphoa nhn){

this.nh.add(nhn);

}

public void inDanhSachhoaNhap() {

for(Nhaphoa nhap: nh ){

System.out.println(nhap);

}

}

public boolean kiemtrahoaton(Nhaphoa nhn) {

return this.nh.contains(nhn);

}

public boolean xoaHoa(Nhaphoa nhn) {

return this.nh.remove(nhn);

}

public boolean kiemtrakhocorong(){

return this.nh.isEmpty();

}

public void lamrongkho(){

this.nh.removeAll(nh);

}

public void timMaHoa(String maHoa){

for (KhoHoatuoi kht : nh) {

if(kht.getMahoa().indexOf(maHoa)>=0){

System.out.println(kht);

}

}

}

public int soluonghoatuoi(){

return this.nh.size();

}

public boolean kiemtrahoa(Nhaphoa nhn){

return this.nh.contains(nhn);

}

public void sapxepSoluongHoa() {

Collections.sort(this.nh, new Comparator<Nhaphoa>(){

public int compare(Nhaphoa nh1, Nhaphoa nh2) {

if(nh1.getSoLuong()<nh2.getSoLuong()) {

return 1;

}else if(nh1.getSoLuong()>nh2.getSoLuong()){

return -1;

}else {

return 0;

}

}

});

}

}

## **4.4 Class DanhsachXuat**

import java.util.ArrayList;

class Danhsachxuat{

private ArrayList<Xuathoa> xh;

public Danhsachxuat(){

this.xh = new ArrayList<Xuathoa>();

}

public Danhsachxuat(ArrayList<Xuathoa> xh){

this.xh = xh;

}

public void themhoaDSX(Xuathoa xhn){

this.xh.add(xhn);

}

public void inDanhSachhoaxuat() {

for(Xuathoa nhap: xh ){

System.out.println(nhap);

}

}

}

## **4.5 Class Main**

import java.util.Scanner;

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

Danhsachnhap dsn = new Danhsachnhap();

Danhsachxuat dsx = new Danhsachxuat();

int n=0;

do {

System.out.println("\n\_\_\_\_\_\_\_Quan ly kho hoa tuoi\_\_\_\_\_\_\_");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MENU\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println(

"\n1.Them hoa" +

"\n2.In danh sach hoa" +

"\n3.Sap xep so luong hoa" +

"\n4.Xoa hoa" +

"\n5.Tim kiem hoa"+

"\n6.Danh sach kho co rong khong?"+

"\n7.Lam rong kho"+

"\n8.So luong hoa tuoi "+

"\n9.Nhap hoa vao danh sach xuat "+

"\n10.In danh sach hoa da xuat di"+

"\n0.\_\_\_\_Nhan 0 de thoat\_\_\_\_");

System.out.println("Xin hay nhap lua chon: ");

n=sc.nextInt();

sc.nextLine();

if(n==1)

{

System.out.println("Nhap ma hoa: ");

String maHoa=sc.nextLine();

System.out.println("Nhap ten hoa: ");

String tenHoa=sc.nextLine();

System.out.println("Nhap mau hoa: ");

String mau=sc.nextLine();

System.out.println("Nhap xuat xu: ");

String xuatXu=sc.nextLine();

System.out.println("Nhap so luong hoa: ");

int soLuong=sc.nextInt();

System.out.println("Nhap vao gia nhap: ");

double giaNhap=sc.nextDouble();

sc.nextLine();

System.out.println("Nhap vao ngay nhap hoa ");

String ngayNhap=sc.nextLine();

Nhaphoa nh= new Nhaphoa(maHoa, tenHoa, mau, xuatXu, giaNhap, soLuong, ngayNhap);

dsn.themhoaDSN(nh);

System.out.println("Them hoa moi thanh cong!");

System.out.println("Thong tin hoa moi: \n" +

" Ma hoa: " + maHoa +

" Ten hoa: "+ tenHoa +

" Mau hoa: "+ mau +

" Xuat xu: " +xuatXu +

" So luong: "+soLuong+

" Gia nhap: " +giaNhap +

" Ngay nhap: " +ngayNhap);

}

else if(n==2)

{

dsn.inDanhSachhoaNhap();

}

else if(n==3)

{

dsn.sapxepSoluongHoa();

dsn.inDanhSachhoaNhap();

}

else if(n==4)

{

System.out.println("Nhap ma hoa muon xoa: ");

String maHoa=sc.nextLine();

Nhaphoa nh=new Nhaphoa(maHoa);

System.out.println("Xoa hoa:"+dsn.xoaHoa(nh));

}

else if(n==5)

{

System.out.println("Nhap ma hoa can tim: ");

String maHoa=sc.nextLine();

Nhaphoa nh=new Nhaphoa(maHoa);

System.out.println("Tim hoa: ");

dsn.timMaHoa(maHoa);

}

else if(n==6)

{

System.out.println("Danh sach co rong khong: "+dsn.kiemtrakhocorong());

}

else if(n==7)

{

dsn.lamrongkho();

System.out.println("Da lam rong kho!");

}else if(n==8)

{

if(dsn.soluonghoatuoi()==0){

System.out.println("Khong co hoa nao !");

}else{

System.out.println("So luong hoa la:"+dsn.soluonghoatuoi());

}

} else if(n==9)

{

System.out.println("Nhap ma hoa: ");

String maHoa=sc.nextLine();

System.out.println("Nhap ten hoa: ");

String tenHoa=sc.nextLine();

System.out.println("Nhap mau hoa: ");

String mau=sc.nextLine();

System.out.println("Nhap xuat xu: ");

String xuatXu=sc.nextLine();

System.out.println("Nhap so luong hoa: ");

int soLuong=sc.nextInt();

System.out.println("Nhap ngay xuat kho:");

String ngayxuat = sc.nextLine();

System.out.println("Nhap gia xuat hoa :");

double giaxuat = sc.nextDouble();

Xuathoa xhx= new Xuathoa(maHoa, tenHoa, mau, xuatXu, soLuong, giaxuat, null);

dsx.themhoaDSX(xhx);

System.out.println("Xuat hoa thanh cong!");

System.out.println("Thong tin hoa duoc xuat: \n" +

" Ma hoa: "+maHoa +

" Ten hoa: "+tenHoa +

" Mau hoa: "+mau +

" Xuat xu: "+xuatXu +

" So luong: "+soLuong +

" Ngay xuat:"+ngayxuat +

" Gia xuat"+ giaxuat);

}else if(n==10){

dsx.inDanhSachhoaxuat();

}

}while (n!=0);

}

}

1. **Kiểm thử**

## **5.1 Menu**

Text

Description automatically generated

## **5.2 Thêm hoa**

Text

Description automatically generated

## **5.3 In danh sách hoa**

Text

Description automatically generated

## **5.4 Sắp xếp số lượng hoa**

Text

Description automatically generated

## **5.5 Xóa hoa**

Graphical user interface, text

Description automatically generated

## **5.6 Tìm kiếm hoa theo mã**

Text

Description automatically generated with medium confidence

## **5.7 Xem danh sách kho có rỗng hay không**

A screenshot of a computer

Description automatically generated with medium confidence

## **5.8 Làm rỗng danh sách**

A screenshot of a computer

Description automatically generated with medium confidence

## **5.9 Số lượng hoa tươi**

A screenshot of a computer

Description automatically generated with medium confidence

## **5.10 Nhập hoa vào danh sách xuất**

Text

Description automatically generated with low confidence

## **5.11 In danh sách xuất hoa**

A screenshot of a computer

Description automatically generated with medium confidence

1. **Kết luận và định hướng phát triển:**

Hệ thống quản lý kho hoa tươi mà nhóm 6 xây dựng đã đáp ứng được phần lớn các yêu cầu về chức năng và cả phi chức năng được đề ra. Tuy nhiên dự án này vẫn chưa hoàn toàn là một sản phẩm hoàn chỉnh, so với nhu cầu tiêu dùng của thị trường hiện tại thì các yêu cầu về tính tiện ích và độ hiệu quả cũng như giao diện thô sơ mà hệ thống mang lại vẫn còn nhiều thiếu sót cần cải tiến.

Xây dựng mô hình dễ hiểu và dễ sử dụng hơn nữa là mục tiêu mà nhóm 6 nhắm đến trong quá trình cải tiến sản phẩm. Bên cạnh đó, đa chức năng cũng là một yếu tố quan trọng mà nhóm 6 hướng tới. Cụ thể, dự án sẽ được bổ sung nhiều ứng dụng thực tế, và các chức năng mới cũng sẽ được thêm thắt dựa trên quá trình khảo nghiệm thị trường thực tiễn. Các yêu cầu về phi chức năng sẽ dần được cải thiện, nâng cao sự chuẩn xác, rút ngắn độ trễ, đảm bảo an toàn thông tin, bảo mật,… từ đó gầy dựng sự tín nhiệm cao nhất có thể cho người sử dụng.

Tuy Hệ thống quản lý kho hoa tươi của nhóm 6 vẫn đang trong giai đoạn nâng cấp và phải cập nhật thêm về sau. Song, mô hình này vẫn là phiên bản hoàn thiện nhất trong tất cả phiên bản mà nhóm 6 sản sinh ra, hơn hết thì dự án cũng đã đạt được mục tiêu đề ra ban đầu nên cũng sẽ giúp đỡ rất nhiều cho người sử dụng trong việc quản lý kho hoa của mình.

1. **Tài liệu tham khảo:**
2. **Studocu:** Ứng dụng Java sử dụng Netbeans - Phần mềm quản lý kho thực phẩm.
3. **Techover.io:** Tổng quan về Activity Diagram.
4. **VietTuts:** ArrayList trong java.
5. **VOER:** Biểu đồ trạng thái (State Diagram).
6. **Freetuts:** Bài tập Java OOP – Chương trình kiểm tra thực phẩm.
7. **123docz:** Viết chương trình nhập họ tên, điểm toán, điểm văn của 1 học sinh tính điểm trung bình và xuất ra kết quả.
8. **Niithanoi:** Cách sử dụng TRY CATH trong JAVA xửa lý tất cả Ngoại lệ.
9. **Alias:** Lập trinh Java – Kiểu dữ lieu date và những cú pháp cực hay ho.
10. **TITV / Lê Nhật Tùng:** Java 61. Bài tập quản lý danh sách sinh viên trong Java sử dụng ArrayList.
11. **19LearningCurve:** How to Draw Sequence Diagram in StarUML.