Trường Đại Học Bách Khoa – Đại Học Đà Nẵng

A picture containing circle, colorfulness, graphics, sketch

Description automatically generatedKhoa Điện tử - Viễn Thông

Thuyết trình PBL1

**KĨ THUẬT LẬP TRÌNH**

**Đề tài: QUẢN LÍ KHO HÀNG**

Giảng viên hướng dẫn: TS. Nguyễn Duy Nhật Viễn

Lớp HP : 22N44A

Lớp SH : 22KTMT2

Sinh viên thực hiện : Lê Thanh Trà - MSSV: 106220273

Trần Đại Tín - MSSV: 106220272

*Đà Nẵng 2023*

**LỜI NÓI ĐẦU**

Dự án quản lý kho hàng được hoàn thiện dựa trên các kiến thức của bộ môn lập trình hướng đối tượng (OOP). Bao gồm các lớp và chức năng được liên kết với nhau từ đó hình thành nên các chức năng để tạo, hiển thị và quản lý các sản phẩm trong kho. Văn bản báo cáo bao gồm việc xây dựng bài toán, những vấn đề đã giải quyết cũng như còn tồn đọng.

**MỤC LỤC**

Bảng phân công nhiệm vụ……………………………………...4

1. Xây dựng chương trình…………………………………………5

**1**.Class thongtin………………………………………………...5

**1**.**1** Class thucpham……………………………………………..6

**1**.**2** Class hangtieudung……………..…………………………..6

**2**.Class thongke………………………………………………...7

**3**.Class quanli…………………………………………………..8

**4** .Định nghĩa các phương thức class 1(thongtin)………………9

**5**.Định nghĩa các phương thức class 2(thongke)……………...12

**6**.Định nghĩa các phương thức class 3(quanli)………………..14

**7**.Đăng kí / Đăng nhập – Main……………………………….22

1. Lưu đồ………………………………………………………...26
2. Chạy chương trình……………………………………………29
3. Tổng kết………………………………………………………36
4. Tài liệu tham khảo……………………………………………36

**BẢNG PHÂN CÔNG NHIỆM VỤ**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | HỌ VÀ TÊN | NHIỆM VỤ | KHỐI LƯỢNG |
| 01 | Trần Đại Tín | Viết class thông tin và thống kê; thiết kế Word | 50% |
| 02 | Lê Thanh Trà | Viết class quản lý, đăng kí đăng nhập, menu; thiết kế PPT | 50% |

1. **Xây dựng chương trình**

**1. Class thongtin**

private:

|  |  |
| --- | --- |
| **Tên biến** | **Kiểu dữ liệu** |
| tensp | string |
| ID | string |
| loai | string |
| hang | string |
| soluong | long long |
| gia | double |

public:

- Hàm tạo / hàm hủy

- Hàm nhập thông tin

- Hàm hiển thị

- Các hàm setter/ getter cho các thuộc tính

- Hàm tính tổng giá trị 1 sản phẩm (đơn giá nhân cho số lượng)

- Hàm đọc – ghi file

**Nội dung**: Xây dựng các thuộc tính cơ bản của sản phẩm và các hàm chức năng liên quan.

* 1. **Class thucpham ( thừa kế từ class thongtin )**

Nhập và hiển thị thông tin thực phẩm

class thucpham : public thongtin

{

public:

thucpham(){};

void nhap();

void hienthi();

};

* 1. **Class hangtieudung ( thừa kế từ class thongtin )**

Nhập và hiển thị thông tin hang tiêu dung

class hangtieudung : public thongtin

{

public:

hangtieudung(){};

void nhap();

void hienthi();

};



1. **Class thongke**

class thoigian

{

public:

long ngay, thang, nam; };

thoigian nhap, xuat, luu;

thongtin i;

long long sl\_daban, sl\_tonkho; // số lượng đã bán, tồn kho

long tg;

public:

- Hàm nhập số lượng đã bán

- Hàm tính số lượng hang hóa còn lại trong kho

- Hàm ghi lại thời gian nhập hàng

- Hàm ghi lại thời gian xuất hang

- Hàm ghi lại thời gian lưu kho

- Hàm hiển thị

- Hàm setter/ getter

- Hàm ghi – đọc file

**Nội dung:**

**-** Lồng thêm class thời gian với các đối tượng nhập, xuất, lưu.

- Chức năng ghi lại thời gian nhập/xuất hàng, thời gian tồn kho.

1. **Class quanli**

private:

thucpham \*tp;

hangtieudung \*htd;

thongke \*h1, \*h2;

int luachon, soloai1, soloai2;

string ten, ID, loai, hang;

double gia; public:

- Hàm tạo/ hàm hủy

- Hàm hiển thị tất cả thông tin

- Hàm tìm kiếm chung ( có các case để lựa chọn )

+ Tìm theo ID

+ Tìm theo tên

+ Tìm theo giá

+Tìm theo loại

- Hàm cập nhật sản phẩm

+Hàm thêm sản phẩm

+Hàm xóa sản phẩm

- Hàm cập nhật số lượng

- Hàm tính tổng giá trị của kho

**Nội dung:**

Tham chiếu đến các thuộc tính, đối tượng của 2 lớp ở trên để quản lý.

1. **Định nghĩa các phương thức Class 1 (thongtin)**

* Các phương thức nhập và hiển thị thuộc tính sản phẩm

*Hàm nhập*

void thongtin::nhap()

{

cin.ignore(0);

fflush(stdin);

cout << "Nhap ten san pham: ";

getline(cin, tensp);

transform(tensp.begin(), tensp.end(), tensp.begin(), ::toupper);

fflush(stdin);

cout << "Nhap ma san pham: ";

getline(cin, ID);

transform(ID.begin(), ID.end(), ID.begin(), ::toupper);

fflush(stdin);

cout << "Nhap loai san pham: ";

getline(cin, loai);

transform(loai.begin(), loai.end(), loai.begin(), ::toupper);

fflush(stdin);

cout << "Nhap hang cua san pham: ";

getline(cin, hang);

transform(hang.begin(), hang.end(), hang.begin(), ::toupper);

do

{ fflush(stdin);

cout << "Nhap so luong: ";

cin >> soluong;

} while (soluong <=0);

do

{ fflush(stdin);

cout << "Nhap gia cua moi san pham(x1000 VND): ";

cin >> gia;

} while (gia <=0);

gia \*= 1000;}

*Hàm hiển thị*

void thongtin::hienthi()

{

cout <<"| " <<left << setw(15) << tensp << setw(6) << left << ID << setw(14) << left << loai << left << setw(12) << left << hang<< setw(10) << left << soluong << setw(15) << left << fixed << setprecision(0) << gia ;

}

* Các phương thức ghi và đọc thông tin

+ Ghi dữ liệu các biến vào tệp tin

+ Tính toán kích thước chuỗi, ghi chuỗi vào tệp tin

+ Đọc dữ liệu từ tệp tin và gán vào các đối tượng (các thuộc tính)

void thongtin::ghi(fstream &g)

{

g.write(reinterpret\_cast<const char \*>(&soluong), sizeof(soluong));

g.write(reinterpret\_cast<const char \*>(&gia), sizeof(gia));

size\_t size = tensp.size();

g.write(reinterpret\_cast<const char \*>(&size), sizeof(size));

g.write(tensp.data(), size);

size = ID.size();

g.write(reinterpret\_cast<const char \*>(&size), sizeof(size));

g.write(ID.data(), size);

size = loai.size();

//…

size = hang.size();

//…

}

void thongtin::doc(fstream &d)

{

d.read(reinterpret\_cast<char \*>(&soluong), sizeof(soluong));

d.read(reinterpret\_cast<char \*>(&gia), sizeof(gia));

size\_t size = tensp.size();

d.read(reinterpret\_cast<char \*>(&size), sizeof(size));

char \*ten\_data = new char[size]; // tạo một mảng ký tự để lưu dữ liệu chuỗi

std::memcpy(ten\_data, tensp.data(), size); // sao chép dữ liệu từ chuỗi sang mảng ký tự

d.read(ten\_data, size); // chuyển mảng ký tự vào hàm read

tensp.assign(ten\_data, size); // gán dữ liệu từ mảng ký tự vào chuỗi

delete[] ten\_data;

size = ID.size();

d.read(reinterpret\_cast<char \*>(&size), sizeof(size));

char \*ID\_data = new char[size]; // tạo một mảng ký tự để lưu dữ liệu chuỗi

std::memcpy(ID\_data, ID.data(), size); // sao chép dữ liệu từ chuỗi sang mảng ký tự

d.read(ID\_data, size); // chuyển mảng ký tự vào hàm read

ID.assign(ID\_data, size); // gán dữ liệu từ mảng ký tự vào chuỗi

delete[] ID\_data;

size = loai.size();

//…

size = hang.size(); //… }

1. **Định nghĩa các phương thức Class 2 (thongke)**

- Lồng thêm class thời gian với các đối tượng nhập, xuất, lưu.

- Chức năng ghi lại thời gian nhập/xuất hàng, thời gian tồn kho.

class thoigian

{

public:

long ngay, thang, nam;

};

thoigian nhap, xuat, luu;

thongtin i;

long long sl\_daban, sl\_tonkho; // số lượng đã bán, tồn kho

long tg;

*-Hàm ghi lại thời gian nhập xuất hàng*

void thongke::nhapvao()

{

fflush(stdin);

cout << "nhap thoi gian nhap hang: \n";

cout << "ngay nhap: ";

cin >> nhap.ngay;

cout << "thang: ";

cin >> nhap.thang;

cout << "nam: ";

cin >> nhap.nam;

}

void thongke::xuatdi()

{

fflush(stdin);

cout << "nhap thoi gian xuat hang: \n";

cout << "ngay xuat: ";

cin >> xuat.ngay;

cout << "thang: ";

cin >> xuat.thang;

cout << "nam: ";

cin >> xuat.nam;

}

*-Hàm đọc ghi file*

void thongke::ghi(fstream &g)

{

g.write(reinterpret\_cast<char \*>(&sl\_daban), sizeof(long long));

g.write(reinterpret\_cast<char \*>(&sl\_tonkho), sizeof(long long));

g.write(reinterpret\_cast<char \*>(&nhap), sizeof(thoigian));

g.write(reinterpret\_cast<char \*>(&xuat), sizeof(thoigian));

g.write(reinterpret\_cast<char \*>(&luu), sizeof(thoigian));

g.write(reinterpret\_cast<char \*>(&tg), sizeof(long long));

}

void thongke::doc(fstream &d)

{

d.read(reinterpret\_cast<char \*>(&sl\_daban), sizeof(long long));

d.read(reinterpret\_cast<char \*>(&sl\_tonkho), sizeof(long long));

d.read(reinterpret\_cast<char \*>(&nhap), sizeof(thoigian));

d.read(reinterpret\_cast<char \*>(&xuat), sizeof(thoigian));

d.read(reinterpret\_cast<char \*>(&luu), sizeof(thoigian));

d.read(reinterpret\_cast<char \*>(&tg), sizeof(long long));

}

1. **Định nghĩa các phương thức Class 3 (quanli)**

-Tham chiếu đến các thuộc tính, đối tượng của 2 lớp ở trên để quản lý.

-*Hàm Menu với các lựa chọn*

1.Cập nhật.

2.Hiển thị tất cả hàng hóa.

3.Tìm kiếm hàng hóa.

4.Thoát

int choice;

cout << "1.Them hang hoa. \n";

cout << "2.Xoa hang hoa. \n";

cout << "3.Cap nhat so luong xuat kho. \n";

cout << "0.Quay lai.\n";

fflush(stdin);

cin >> choice;

switch (choice)

{

case 1:

add\_sp();

break;

case 2:

del\_sp();

break;

case 3:

ud\_soluong();

break;

case 0:

system("cls");

menu();

default:

break;

}

//…..

bool count = true;

do

{

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << "| |" << endl;

cout << "| MENU |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| " << endl;

cout << "1.Cap nhat hang hoa. \n";

cout << "2.Hien thi tat ca hang hoa. \n";

cout << "3.Tim kiem hang hoa. \n";

cout << "4.Thoat!! \n";

fflush(stdin);

cin >> luachon;

switch (luachon)

{

case 1:

update();

dieukien();

break;

case 2:

hienthi();

dieukien();

break;

case 3:

timkiem();

break;

case 4:

count = false;

// cout << "Da ket thuc lenh" << endl;

break;

default:

break;

}

} while (count == true);

cout << "Da ket thuc lenh!!!" << endl;

exit(1); }

*-Hàm cập nhật sản phẩm (thêm sản phẩm)*

void quanli::add\_sp()

{

int \*sl;

cout << "Nhap so loai san pham can them: ";

cin >> \*sl;

doc\_sl();

// cout<<"so loai = "<<soloai<<endl;

fstream g("ttin.bin", ios::binary | ios::out | ios::app);

for (int i = soloai; i < soloai + \*sl; i++)

{

cout << "Nhap thong tin san pham thu:" << i + 1 << endl;

fflush(stdin);

t[i].nhap();

t[i].ghi(g);

h[i].nhaphd();

h[i].ghi(g);

system("cls");

}

g.close();

delete sl;}

*-Hàm kiểm tra ID*

Nếu nhập ID trùng sẽ thông báo và yêu cầu nhập lại.

void quanli::check\_ID(int i)

{

for (int j = 0; j < i; j++)

{

while (tp[i].get\_ID() == tp[j].get\_ID() || tp[i].get\_ID() == htd[j].get\_ID() )

{

cout << "Trung ID! Hay nhap lai san pham nay " << endl;

cout << "Nhap thong tin san pham thu: " << i + 1 << endl;

fflush(stdin);

tp[i].nhap();

}

}

for (int j=0; i < i; j++)

{

while (htd[i].get\_ID() == htd[j].get\_ID() || htd[i].get\_ID() == tp[j].get\_ID())

{

cout << "Trung ID! Hay nhap lai san pham nay " << endl;

cout << "Nhap thong tin san pham thu: " << i + 1 << endl;

fflush(stdin);

htd[i].nhap();

}

}

}

*-Hàm cập nhật sản phẩm ( xóa sản phẩm )*

void quanli::del\_sp()

{

doc\_sl();

string ma;

cout << "Nhap ID hang hoa can xoa: " << endl;

fflush(stdin);

getline(cin, ma);

transform(ma.begin(), ma.end(), ma.begin(), ::toupper);

for (int i = 0; i < soloai1; i++)

{

if (ma == tp[i].get\_ID())

{

cout << "Da xoa " << tp[i].get\_ten() << " ID: " << ma << endl;

for (int j = i; j < soloai1 - 1; j++)

{

{

tp[j] = tp[j + 1];

h1[j] = h1[j + 1];

}

}

soloai1--;

}

}

for (int i = 0; i < soloai2; i++)

{

if (ma == htd[i].get\_ID())

{

cout << "Da xoa " << htd[i].get\_ten() << " ID: " << ma << endl;

for (int j = i; j < soloai2 - 1; j++)

{

{

htd[j] = htd[j + 1];

h2[j] = h2[j + 1];

}

}

soloai2--;

}

}

fstream g1("thucpham.bin", ios::binary | ios::out); // mo file ghi thuc pham

fstream g2("hangtieudung.bin", ios::binary | ios::out); // mo file ghi hang tieu dung

if (!g1 || !g2)

{

cout << "Loi mo file!\n";

exit(0);

}

for (int i = 0; i < soloai1; i++)

{

tp[i].ghi(g1);

h1[i].ghi(g1);

}

for (int i = 0; i < soloai2; i++)

{

htd[i].ghi(g2);

h2[i].ghi(g2);

}

g1.close();

g2.close();

}

*-Hàm hiển thị*

void quanli::hienthi()

{

soloai = 0;

fstream d("ttin.bin", ios::binary | ios::in);

if (!d)

{

cout << "Loi mo file" << endl;

// exit(1);

}

else

{

while (!d.eof())

{

t[soloai].doc(d);

h[soloai].doc(d);

if (!d.eof())

soloai++;

}

cout << "SO LOAI: " << soloai << endl;

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \n";

cout << "| |\n";

cout << left << setw(18) << "| TEN" << setw(6) << left << "ID" << setw(14) << left << "LOAI" << setw(12) << left << "HANG" << setw(10) << left << "SL"

<< setw(15) << left << "GIA(VND)" << setw(15) << left << "SL DA XUAT" << setw(15) << left << "SL TON KHO" << setw(15) << left << "NGAY NHAP" << setw(15) << left

<< "NGAY XUAT" << setw(15) << left << "TONG GIA TRI SAN PHAM(VND) |" << endl;

cout << "|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|\n";

for (int i = 0; i < soloai; i++)

{

t[i].doc(d);

h[i].doc(d);

t[i].hienthi();

h[i].hienthi(t[i]);

cout << setw(26) << left << t[i].get\_tongsp() <<"|"<< endl;

cout << "|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|\n";

}

cout << endl;

cout << "--------------------------------------- \n";

cout << "Tong gia tri trong kho: " << tongGT\_kho() << " VND" << endl;

d.close();

}

// system("cls");

}

*-Hàm tìm kiếm theo ID*

void quanli::tim\_ID()

{

fflush(stdin);

cin.ignore(0);

cout << "---------nhap ID hang hoa can tim kiem------" << endl;

getline(cin, ID);

transform(ID.begin(), ID.end(), ID.begin(), ::toupper);

fstream d("ttin.bin", ios::binary | ios::in);

if (!d)

{cout << "Loi mo file!\n";}

else

{cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \n";

cout << "| |\n";

cout << left << setw(18) << "| TEN" << setw(6) << left << "ID" << setw(14) << left << "LOAI" << setw(12) << left << "HANG" << setw(10) << left << "SL"

<< setw(15) << left << "GIA(VND)" << setw(15) << left << "SL DA XUAT" << setw(15) << left << "SL TON KHO" << setw(15) << left << "NGAY NHAP" << setw(15) << left

<< "NGAY XUAT" << setw(15) << left << "TONG GIA TRI SAN PHAM(VND) |" << endl;

cout << "|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|\n";

for (int i = 0; i < soloai; i++)

{

t[i].doc(d);

h[i].doc(d);

if (ID == t[i].get\_ID())

{

t[i].hienthi();

h[i].hienthi(t[i]);

cout << setw(26) << left << t[i].get\_tongsp() <<"|"<< endl;

cout << "|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|\n";

}

}d.close();

}

}

-Hàm tìm theo tên //…

-Hàm tìm theo giá //…

-Hàm tìm theo loại //…

1. **Đăng kí / Đăng nhập – Main**

USER

Hệ thống

Tài khoản

GỬI YÊU CẦU ĐĂNG NHẬP

YÊU CẦU NHẬP TK VÀ MK

NHẬP TK VÀ MK

GỬI TK VÀ MK

TRẢ VỀ THÔNG BÁO

Sử dụng đọc/ghi file nhị phân để lưu Tài khoản – Mật khẩu người dùng.

Mở file acc.bin để lưu TK đăng kí

void sign\_up()

{

accout acc;

fstream w("acc.bin", ios::binary | ios::out);

if (!w)

{

cout << "Khong the mo file!" << endl;

exit(1);

}

// w.clear();

acc.nhap();

w.write(reinterpret\_cast<const char \*>(&acc), sizeof(accout));

cout << "Dang ki thanh cong !!!\n";

w.close();

}

* Nhập thông tin đăng nhập và kiểm tra với thông tin đăng kí được lưu trong file

void sign\_in()

{

accout acc;

static int choice, dem = 0, j=0;

static string user, password, tim\_acc, tim\_pass;

fstream r("acc.bin", ios::binary | ios::in);

if (!r)

{

cout << "Khong the mo file!" << endl;

exit(1);

}

tt:

fflush(stdin);

cout << "Nhap ten dang nhap: \n";

cin >> user;

system("cls");

cout << "Nhap mat khau: \n";

fflush(stdin);

cin >> password;

system("cls");

r.read(reinterpret\_cast<char \*>(&acc), sizeof(accout));

if (user == acc.username && password == acc.password)

{

cout << "\nDang nhap thanh cong! \n";

cout << "------------------------\n";

}

* Nếu nhập sai thông tin đăng nhập
* Thông báo và yêu cầu chọn quên mật khẩu / quên tên đăng nhập để tìm lại tài khoản

if (user != acc.username || password != acc.password)

{

cout << "Sai 3 lan se phai dang ki lai! \n";

for (int i = j; i < 3; i++)

{

j++;

goto tt;

}

cout << "1.Quen mat khau \n";

cout << "2.Quen ten dang nhap \n";

cin >> choice;

switch (choice)

{

case 1:

do

{

cout << "Nhap ten dang nhap de tim mat khau \n";

cin >> tim\_acc;

if (tim\_acc == acc.username)

{

cout << "Mat khau cua tai khoan( " << acc.username << " )la: " << acc.password << endl;

int lc;

cout<<"Nhap 0 de quay lai.";

cin>>lc;

if(lc==0)

goto tt;

}

else

{

dem++;

cout << "Sai 3 lan se dang ki lai \n";

cout << "Sai lan thu: " << dem << endl;

if (dem == 3)

{

sign\_up();

goto tt;

}

}

} while (true);

break;

case 2:

sign\_up();

break;

default:

break;

}

}

r.close();

1. **LƯU ĐỒ**

* Lưu đồ hàm nhập / hàm xuất các thuộc tính của sản phẩm

Nhập đơn giá

Nhập số lượng

Nhập hãng

Nhập loại

Nhập ID

Nhập tên

Nhập tên

Nhập ID

Nhập loại

Nhập hãng

Nhập số lượng

Nhập đơn giá

|  |
| --- |
| tensp[i]  =tensp cần tìm  i< số loại  Nhập tên sản phẩm cần tìm  Xuất thông tin các sản phẩm có trong kho  i = 0  i+1  T  T  F |

|  |  |
| --- | --- |
| Nhập ID sản phẩm cần tìm  i< số loại  F  T  T  i+1  i = 0  ID[i]=ID cần tìm  Xuất thông tin các sản phẩm có trong kho   |  | | --- | | Lưu đồ hàm tìm sản phẩm theo tên | |

|  |
| --- |
| Lưu đồ hàm tìm sản phẩm theo ID |

|  |
| --- |
|  |

|  |
| --- |
| Lưu đồ hàm nhập lựa chọn |
|  |

|  |
| --- |
| Nhập lựa chọn  Hiển thị lựa chọn  1.Cập nhật  2.Hiển thị  3.Tìm kiếm  4.Thoát  Gọi các hàm tương ứng với lựa chọn |

1. **Chạy chương trình**

Bắt đầu với chức năng đăng kí / đăng nhập

|  |
| --- |
| A screen shot of a computer  Description automatically generated with medium confidence |

|  |
| --- |
| Sau khi đăng kí người dùng sẽ đăng nhập.  Nếu nhập sai thông tin sẽ thông báo. |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
| Hiển thị Menu với chức năng để lựa chọn |

|  |
| --- |
|  |

|  |
| --- |
| Ban đầu sẽ chưa có thông tin của bất cứ sản phẩm gì |

|  |
| --- |
|  |

Cập nhật – Thêm sản phẩm

|  |
| --- |
|  |

|  |
| --- |
| Thêm số lượng xuất hàng và thời gian xuất hàng |

|  |
| --- |
|  |

|  |
| --- |
|  |

Tìm kiếm hàng hóa theo ID

|  |
| --- |
|  |

|  |
| --- |
| Tìm kiếm hàng hóa theo giá |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

1. **Tổng kết**

Ưu điểm :

* Các chức năng của chương trình chạy khá hoàn thiện.
* Code khá sạch và gọn, dễ hiểu.
* Giao diện dễ sử dụng.

Nhược điểm :

* Có thể gặp lỗi bộ nhớ khi đọc file nhị phân trong quá trình sử dụng

1. **Nguồn tham khảo**

* Link youtube hướng dẫn làm việc với file nhị phân <https://www.youtube.com/watch?v=v7IVPtDDh18&ab_channel=th%C3%A2ntri%E1%BB%87u>
* Link trang web học tập về kiến thức Lập trình HĐT

<https://nhatvien.com/courses/lap-trinh-huong-doi-tuong/>

* Sách thao khảo: How to C program – Paul Deitel & Harvey Deitel

**THE END**