**TRƯỜNG** **ĐẠI** **HỌC** **BÁCH** **KHOA**

**KHOA** **ĐIỆN** **TỬ-VIỄN** **THÔNG**

**----------🙢 🕮 🙠----------**



**PBL1: LẬP** **TRÌNH**

**ĐỀ** **TÀI**

**PHẦN MỀM QUẢN LÍ CỬA HÀNG THỜI TRANG**

**GVHD:** TS.Văn Phú Tuấn

**SVTH:**

1. Lê Tán Nhất Quang Nhóm: 24.43A Lớp: 24KTMT1

2. Thái Văn Tuấn Nhóm: 24.43A Lớp: 24KTMT1

## *Đà Nẵng*, 06/2025

**THUYẾT MINH PBL1**

**LẬP TRÌNH**

**Bảng phân công nhiệm vụ và mức độ đóng góp của các thành viên**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên thành viên** | **Mã số sinh viên** | **Nhiệm vụ được phân công** | **Mức độ đóng góp (%)** | **Ký tên** |
| Lê Tán Nhất Quang | 106240245 | Code chính, sửa code, quay video  word,powerpoint | 55% |  |
| Thái Văn Toàn | 106240254 | Code phụ, kiểm tra code, word, lưu đồ | 45% |  |

**MỤC LỤC:**

*[Đà Nẵng](#_Toc23744)*[, 06/2025 1](#_Toc23744)

[NỘI DUNG 5](#_Toc7275)

[A. GIỚI THIỆU ĐỀ TÀI 5](#_Toc20888)

[1. Mô tả tổng quan về đề tài 5](#_Toc118)

[2. Tính cấp thiết và ý nghĩa của đề tài 5](#_Toc18394)

*[a. Tính cấp thiết](#_Toc30755)* [5](#_Toc30755)

*[b. Ý nghĩa của đề tài](#_Toc3221)* [6](#_Toc3221)

[3. Mục tiêu và phạm vi nghiên cứu 6](#_Toc30026)

*[a. Mục tiêu](#_Toc18237)* [6](#_Toc18237)

*[b. Phạm vi nghiên cứu](#_Toc29341)* [7](#_Toc29341)

[B. Xây dựng chương trình 8](#_Toc31876)

[1) Các thư viện đã dùng 8](#_Toc11176)

[1.1 Thư viện <iostream> 8](#_Toc9441)

[1.2 Thư viện<vector> 8](#_Toc7610)

[1.3 Thư viện<fstream> 8](#_Toc24872)

[1.4 Thư viện<string> 8](#_Toc7433)

[1.5 Thư viện<iomanip> 8](#_Toc9605)

[1.6 Thư viện<ctime> 8](#_Toc17763)

[1.7 Thư viện <sstream> 9](#_Toc12487)

[1.8 Thư viện<conio.h> (Thư viện của Windows,không chuẩn C++) 9](#_Toc26835)

[2) các lớp và hàm được sử dụng 9](#_Toc115)

[2.1 Lớp sản phẩm (class Product) 9](#_Toc12019)

[2.2 Lớp nhân viên (Class Seller) 10](#_Toc25925)

[2.3 Lớp Hóa đơn (Class Bill) 10](#_Toc22286)

[2.4 Hàm lưu và đọc danh sách sản phẩm vào file 12](#_Toc29930)

[2.5 Hàm lưu và đọc danh sách nhân viên 13](#_Toc28001)

[2.6 Hàm đăng nhập quản lý 14](#_Toc19147)

[2.7 Hàm trang quản lý sản phẩm 15](#_Toc26443)

[2.8 Hàm xem danh sách khách hàng 17](#_Toc6143)

[2.9 Hàm giao diện quản lí nhân viên và khách hàng: 18](#_Toc17909)

[2.10 Hàm tạo mã bill chạy tự động 20](#_Toc17361)

[2.11 Hàm tạo hóa đơn: 20](#_Toc6967)

[2.12 Hàm tính tổng doanh thu: 22](#_Toc10628)

[2.13 Hàm đổi mật khẩu ADMIN: 23](#_Toc27345)

[2.14 ) Hàm trang chính 24](#_Toc9535)

[3) Kết quả chính khi chạy thử chương trình 26](#_Toc29224)

[3.1 Màn hình hiển thị trang chính 26](#_Toc28994)

[3.2 Danh sách nhân viên 26](#_Toc29478)

[3.3 Danh sách sản phẩm hiện có 26](#_Toc7634)

[3.4 Hiển thị hóa đơn mua hàng 27](#_Toc17141)

[C. LƯU ĐỒ GIẢI THUẬT 29](#_Toc387)

[1. Lưu đồ tổng quát chương trình 29](#_Toc9127)

[2. Lưu đồ Trang chính 30](#_Toc11458)

[3.Lưu đồ quản lý sản phẩm 31](#_Toc26635)

[4. Lưu đồ quản lý sản nhân viên và khách hàng 32](#_Toc724)

[5.Lưu đồ tạo hóa đơn 33](#_Toc6974)

[6. Lưu đồ đổi mật khẩu ADMIN 34](#_Toc27554)

[7. Lưu đồ tổng doanh thu 35](#_Toc29117)

[D. PHÂN TÍCH VÀ ĐÁNH GIÁ ĐỒ ÁN QUẢN LÝ CỬA HÀNG THỜI TRANG 36](#_Toc27912)

[1. Tổng quan hệ thống 36](#_Toc14334)

[2. Các chức năng chính 36](#_Toc12167)

*[a. Quản lý sản phẩm](#_Toc21212)* [36](#_Toc21212)

*[b. Quản lý nhân viên (Seller)](#_Toc20864)* [36](#_Toc20864)

*[c. Quản lý khách hàng](#_Toc18522)* [36](#_Toc18522)

*[d. Tạo hóa đơn](#_Toc17772)* [36](#_Toc17772)

*[e. Thống kê doanh thu](#_Toc15240)* [36](#_Toc15240)

*[f. Hệ thống đăng nhập](#_Toc6826)* [36](#_Toc6826)

[3. Ưu điểm 37](#_Toc23700)

[4. Hạn chế và đề xuất cải tiến 37](#_Toc19925)

*[a. Quản lý dữ liệu](#_Toc5701)* [37](#_Toc5701)

*[b. Giao diện](#_Toc25800)* [37](#_Toc25800)

*[c. Chức năng mở rộng](#_Toc19576)* [37](#_Toc19576)

[5. Kết luận 37](#_Toc31667)

[TÀI LIỆU THAM KHẢO 39](#_Toc23378)

**NỘI DUNG**

1. **GIỚI THIỆU ĐỀ TÀI**

**1. Mô tả tổng quan về đề tài**

Đề tài **"Hệ thống quản lý cửa hàng thời trang TVQ"** là một phần mềm quản lý bán hàng được phát triển bằng ngôn ngữ C++, giúp tự động hóa các công việc như:

- **Quản lý sản phẩm** (thêm, sửa, xóa, kiểm tra tồn kho).

- **Quản lý nhân viên bán hàng** (Seller).

- **Quản lý khách hàng và lịch sử mua hàng**.

-  **Tạo hóa đơn, in bill tự động**.

-  **Thống kê doanh thu theo thời gian**.

Hệ thống lưu trữ dữ liệu dưới dạng file text (products.txt, sellers.txt, costumers.txt, bills\_data.txt) và hỗ trợ đăng nhập phân quyền (quản lý/nhân viên).

**2. Tính cấp thiết và ý nghĩa của đề tài**

***a. Tính cấp thiết***

**Nhu cầu quản lý kinh doanh thời trang ngày càng cao**, đặc biệt với các cửa hàng nhỏ và vừa.

**Quản lý thủ công** (sổ sách, Excel) dễ gây **nhầm lẫn, thất thoát, khó thống kê**.

**Cần hệ thống tự động hóa** để:

- Giảm thiểu sai sót trong quản lý kho.

- Tăng tốc độ thanh toán, tạo hóa đơn.

- Dễ dàng theo dõi doanh thu, khách hàng.

***b. Ý nghĩa của đề tài***

**Đối với cửa hàng**:

*-* **Tối ưu hóa quy trình bán hàng**, giảm thời gian quản lý.

**- Nâng cao trải nghiệm khách hàng** với hóa đơn rõ ràng, lưu trữ thông tin mua hàng.

**Đối với nhân viên**:

- Dễ dàng **tra cứu sản phẩm, tạo bill nhanh chóng**.

**- Giảm áp lực** khi quản lý tồn kho và doanh số.

**Đối với chủ cửa hàng (quản lý)**:

- Kiểm soát nội bộ cửa hàng.

**- Kiểm soát doanh thu** theo ngày/tháng/năm.

**- Phân tích hành vi khách hàng** để đưa ra chiến lược kinh doanh phù hợp.

**3. Mục tiêu và phạm vi nghiên cứu**

***a. Mục tiêu***

- **Xây dựng hệ thống quản lý bán hàng tự động** trên nền tảng C++.

**- Quản lý hiệu quả** sản phẩm, nhân viên, khách hàng.

**- Tạo hóa đơn tự động**, lưu trữ lịch sử giao dịch.

**- Thống kê doanh thu** để hỗ trợ ra quyết định kinh doanh.

***b. Phạm vi nghiên cứu***

**Đối tượng áp dụng**: Cửa hàng thời trang quy mô vừa và nhỏ.

**Chức năng chính**:

*-* Quản lý sản phẩm (thêm/xóa/sửa, kiểm kho).

*-*  Quản lý nhân viên bán hàng.

*-* Tạo hóa đơn, lưu trữ thông tin khách hàng.

*-*  Báo cáo doanh thu cơ bản.

**Giới hạn**:

*-* Chưa có **web/app quản lý từ xa**, chỉ chạy trên máy tính.

*-* Chưa tích hợp **thanh toán online** hoặc **quét mã vạch**.

1. **Xây dựng chương trình**
2. **Các thư viện đã dùng**

### **1.1 Thư viện <iostream>**

#### Công dụng: Thư viện chuẩn C++ để nhập/xuất dữ liệu (Input/Output Stream).

### **1.2 Thư viện<vector>**

#### Công dụng: Cung cấp mảng động (dynamic array) có thể thay đổi kích thước.

### **1.3 Thư viện<fstream>**

#### Công dụng: Đọc/ghi dữ liệu từ tệp tin (File Stream).

### **1.4 Thư viện<string>**

#### Công dụng: Làm việc với chuỗi ký tự (std::string) thay vì mảng ký tự kiểu C (char[]).

### **1.5 Thư viện<iomanip>**

#### Công dụng: Định dạng xuất dữ liệu (thường dùng với cout).

### **1.6 Thư viện<ctime>**

#### Công dụng: Làm việc với thời gian (ngày, giờ, timestamp).

### **1.7 Thư viện <sstream>**

#### Công dụng: Đọc/ghi dữ liệu từ chuỗi như một luồng (stream)

### **1.8 Thư viện<conio.h> (Thư viện của Windows,không chuẩn C++)**

#### Công dụng:  Nhập ký tự không hiển thị

1. **các lớp và hàm được sử dụng**

**2.1 Lớp sản phẩm (class Product)**

class Product {

    private:

        string id, name, size, gender;

        double price;

        int stock;

    public:

        Product(string id, string name, string size, double price, int stock, string gender)

            : id(id), name(name), size(size), price(price), stock(stock), gender(gender) {}

        string getId() const { return id; }

        string getName() const { return name; }

        string getSize() const { return size; }

        double getPrice() const { return price; }

        int getStock() const { return stock; }

        string getGender() const { return gender; }

        void setName(const string& newName) { name = newName; }

        void setSize(const string& newSize) { size = newSize; }

        void setPrice(double newPrice) { price = newPrice; }

        void setStock(int newStock) { stock = newStock; }

        void setGender(const string& newGender) { gender = newGender; }

        void reduceStock(int amount) { stock -= amount; }

        void display() const {

            cout << left << setw(10) << id

                << setw(20) << name

                << setw(6) << size

                << setw(10) << gender

                << setw(10) << fixed << setprecision(0) << price

                << setw(10) << stock << endl;

        }

    };

**- Các cấu trúc được sử dụng:**

#### **+ Access Modifiers (Phạm vi truy cập)**

#### **+ Constructor (Hàm khởi tạo)**

#### **+ Getter/Setter (Phương thức truy cập)**

#### **+ Method (Phương thức)**

**2.2 Lớp nhân viên (Class Seller)**

class Seller {

    private:

        string sellerID, fullName;

    public:

        Seller(string sellerID, string fullName)

            : sellerID(sellerID), fullName(fullName) {}

        string getID() const { return sellerID; }

        string getName() const { return fullName; }

        void display() const {

            cout << left << setw(10) << sellerID << setw(20) << fullName << endl;

        }};

**-Các cấu trúc được sử dụng**

#### + **Access Modifiers (Phạm vi truy cập)**

#### **+ Constructor (Hàm khởi tạo)**

#### **+ Getter (Phương thức truy cập)**

#### **+ Method (Phương thức)**

#### + **Formatting Output (Định dạng xuất)**

**2.3 Lớp Hóa đơn (Class Bill)**

class Bill {

    private:

        string billID, customerName, phone, email, address;

        vector<Product> items;

        string sellerID;

        string storeName = "Cua hang thoi trang TVQ";

        string hotline = "0909 000999";

        string storeEmail = "TvQfashion@gmail.com";

        string storeAddress = "193 Nguyen Luong Bang, Da Nang";

        time\_t billTime;

    public:

        Bill(string billID, string customerName, string phone, string email, string address, string sellerID, vector<Product> items)

            : billID(billID), customerName(customerName), phone(phone), email(email), address(address), sellerID(sellerID), items(items) {

            billTime = time(0);

        }

        time\_t getBillTime() const { return billTime; }

        string getCurrentTime() {

            time\_t now = time(0);

            tm t;

            localtime\_s(&t, &now);

            char buf[100];

            strftime(buf, sizeof(buf), "%Y-%m-%d %H:%M", &t);

            return string(buf);

        }

        void writeToFile() {

            string filename = "bill\_" + billID + ".txt";

            ofstream out(filename);

            if (!out) {

                cerr << "Khong the tao file hoa don!\n";

                return;

            }

            out << "======== HOA DON BAN HANG ========\n";

            out << "Ma hoa don: " << billID << endl;

            out << "Thoi gian: " << getCurrentTime() << endl;

            out << "Nhan vien ban hang: " << sellerID << endl;

            out << "-------------------------------\n";

            out << "Ten KH: " << customerName << endl;

            out << "Dia chi: " << address << endl;

            out << "SDT: " << phone << endl;

            out << "Email: " << email << endl;

            out << "-------------------------------\n";

            out << left << setw(20) << "Ten SP"

                << setw(6) << "Size"

                << setw(10) << "Gioi tinh"

                << setw(10) << "Don gia"

                << setw(6) << "SL"

                << setw(10) << "Thanh tien" << endl;

            double total = 0;

            out << fixed << setprecision(0);

            for (auto p : items) {

                double sub = p.getPrice() \* p.getStock();

                total += sub;

                out << left << setw(20) << p.getName()

                    << setw(6) << p.getSize()

                    << setw(10) << p.getGender()

                    << setw(10) << p.getPrice()

                    << setw(6) << p.getStock()

                    << setw(10) << sub << endl;

            }

            out << "-------------------------------\n";

            out << "Tong cong: " << total << " VND\n";

            out << "\n" << storeName << endl;

            out << "Dia chi: " << storeAddress << endl;

            out << "Hotline: " << hotline << " | Email: " << storeEmail << endl;

            out << "\nCam on quy khach va hen gap lai!\n";

            out.close();

        }

        void saveBillToFile() {

            string filename = "bills\_data.txt";

            ofstream out(filename, ios::app);

            if (!out.is\_open()) {

                cerr << "Khong the mo file de luu hoa don!" << endl;

                return;

            }

            out << "=== BILL " << billID << " ===\n";

            out << "Thoi gian: " << getCurrentTime() << "\n";

            out << "Khach hang: " << customerName << "\n";

            out << "Nhan vien: " << sellerID << "\n";

            out << "\nCHI TIET SAN PHAM:\n";

            out << left << setw(15) << "Ma SP"

                << setw(20) << "Ten SP"

                << setw(6)<< "Size"

                << setw(6) << "SL"

                << setw(12) << "Don gia"

                << setw(12) << "Thanh tien" << "\n";

            double total = 0;

            out << fixed << setprecision(0);

            for (const auto& item : items) {

                double subtotal = item.getPrice() \* item.getStock();

                total += subtotal;

                out << left << setw(15) << item.getId()

                    << setw(20) << item.getName()

                    << setw(6) << item.getSize()

                    << setw(6) << item.getStock()

                    << setw(12) << item.getPrice()

                    << setw(12) << subtotal << "\n";

            }

            out << "\nTONG CONG: " << total << " VND\n";

            out << "=================================\n\n";

            out.close();

            cout << "Da luu hoa don " << billID << " vao file " << filename << endl;

        }

        double tinhTongTien() const {

            double tongTien = 0.0;

            for (const auto& item : items) {

                tongTien += item.getPrice() \* item.getStock();

            }

            return tongTien;

        }

    };

**- CÁC CẤU TRÚC ĐƯỢC SỬ DỤNG**

#### + **Đóng gói (Encapsulation)**

#### **+ Constructor (Hàm khởi tạo)**

#### +**Định dạng Xuất (Formatting)**

+ Phương thức xử lí file hóa đơn:

****Ghi 2 loại file****:

File riêng cho từng hóa đơn (bill\_<ID>.txt).

File tổng hợp tất cả hóa đơn (bills\_data.txt).

****Tái sử dụng phương thức****:

getCurrentTime() dùng chung cho cả writeToFile() và saveBillToFile().

****Xử lý lỗi cơ bản****:

Kiểm tra if (!out) khi mở file.

**2.4 Hàm lưu và đọc danh sách sản phẩm vào file**

void saveProductsToFile(const vector<Product>& products) {

        ofstream out("products.txt");

        if (!out.is\_open()) {

            cerr << "Khong the mo file products.txt de ghi!\n";

            return;

        }

        for (auto p : products) {

            out << p.getId() << "," << p.getName() << "," << p.getSize() << ","

                << p.getGender() << "," << p.getPrice() << "," << p.getStock() << endl;

        }

        out.close();

    }

 void loadProductsFromFile(vector<Product>& products) {

        ifstream in("products.txt");

        if (!in.is\_open()) {

            cout << "Khong tim thay file products.txt. Tao file moi khi co san pham duoc them.\n";

            return;

        }

        products.clear();

        string line;

        while (getline(in, line)) {

            stringstream ss(line);

            string id, name, size, gender;

            double price;

            int stock;

            if (getline(ss, id, ',') &&

                getline(ss, name, ',') &&

                getline(ss, size, ',') &&

                getline(ss, gender, ',') &&

                (ss >> price) &&

                (ss.ignore(), ss >> stock)) {

                products.push\_back(Product(id, name, size, price, stock, gender));

            }

        }

        in.close();

    }

**- Các cấu trúc được sử dụng**

#### + **File I/O (Nhập/Xuất File)**

#### **+ Xử lý Lỗi File**

#### **+Định dạng CSV(giá trị phân tách bởi dấu “,”)**

**2.5 Hàm lưu và đọc danh sách nhân viên**

 void saveSellersToFile(const vector<Seller>& sellers) {

        ofstream out("sellers.txt");

        if (!out.is\_open()) {

            cerr << "Khong the mo file sellers.txt de ghi!\n";

            return;

        }

        for (auto s : sellers) {

            out << s.getID() << "," << s.getName() << endl;

        }

        out.close();

    }

    void loadSellersFromFile(vector<Seller>& sellers) {

        ifstream in("sellers.txt");

        if (!in.is\_open()) {

            cout << "Khong tim thay file sellers.txt. Tao file moi khi co seller duoc them.\n";

            return;

        }

        sellers.clear();

        string line;

        while (getline(in, line)) {

            size\_t pos = line.find(',');

            if (pos != string::npos) {

                string id = line.substr(0, pos);

                string name = line.substr(pos + 1);

                sellers.push\_back(Seller(id, name));

            }

        }

        in.close();

}

**-Các cấu trúc được sử dụng**

#### + **File I/O (Nhập/Xuất File)**

#### **+ Xử lý Lỗi File**

#### **+Định dạng CSV(giá trị phân tách bởi dấu “,”)**

**2.6 Hàm đăng nhập quản lý**

bool dangNhapQuanLy() {

        string user, pass;

        int demSai = 0;

        while (demSai < 3) {

            cout << "Nhap tai khoan: ";

            getline(cin, user);

            cout << "Nhap mat khau: ";

            pass = "";

            char c;

            while ((c = \_getch()) != '\r') {

                if (c == '\b' && !pass.empty()) {

                    pass.pop\_back();

                    cout << "\b \b";

                }

                else {

                    pass.push\_back(c);

                    cout << '\*';

                }

            }

            cout << endl;

            if (user == "TvQ" && pass == "QvT") {

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

                return true;

            }

            else {

                cout << "Sai tai khoan hoac mat khau. Thu lai!\n";

                demSai++;

            }

        }

        cout << "Nhap sai qua 3 lan. Quay ve menu chinh.\n";

        return false;

    }

**-Các cấu trúc được sử dụng**

#### + **Vòng lặp while với bộ đếm**

#### **+ Nhập mật khẩu ẩn (không hiển thị)**

#### **+ Kiểm tra tài khoản/mật khẩu**

#### **+ Kết thúc chương trình(sau 3 lần)**

**2.7 Hàm trang quản lý sản phẩm**

    void menuQuanLySanPham(vector<Product>& dsSanPham) {

        if (!dangNhapQuanLy()) return;

        int chon;

        do {

            cout << "\n================================= QUAN LY SAN PHAM =================================\n";

            cout << "==\t\t\t 1. Xem danh sach san pham \t\t\t\t  ==\n";

            cout << "==\t\t\t 2. Them san pham  \t\t\t\t\t  ==\n";

            cout << "==\t\t\t 3. Sua san pham  \t\t\t\t\t  ==\n";

            cout << "==\t\t\t 4. Xoa san pham \t\t\t\t\t  ==\n";

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

            cout << "======================================================================================\n";

            cout << "Chon: ";

            cin >> chon;

            cin.ignore();

            switch (chon) {

            case 1: {

                cout << left << setw(10) << "Ma"

                    << setw(20) << "Ten"

                    << setw(6) << "Size"

                    << setw(10) << "Gioi tinh"

                    << setw(10) << "Gia"

                    << setw(10) << "Ton kho" << endl;

                for (auto& p : dsSanPham) {

                    p.display();

                }

                break;

            }

            case 2: {

                string id, name, size, gender;

                double price;

                int stock;

                cout << "Nhap ma sp: "; getline(cin, id);

                cout << "Nhap ten sp: "; getline(cin, name);

                cout << "Nhap size: "; getline(cin, size);

                cout << "Nhap gioi tinh: "; getline(cin, gender);

                cout << "Nhap gia: "; cin >> price;

                cout << "Nhap so luong: "; cin >> stock;

                cin.ignore();

                dsSanPham.push\_back(Product(id, name, size, price, stock, gender));

                saveProductsToFile(dsSanPham);

                cout << "Them thanh cong.\n";

                break;

            }

            case 3: {

                string id;

                cout << "Nhap ma sp can sua: ";

                getline(cin, id);

                bool found = false;

                for (auto& p : dsSanPham) {

                    if (p.getId() == id) {

                        string name, size, gender;

                        double price;

                        int stock;

                        cout << "Ten moi (bo qua de giu): "; getline(cin, name);

                        if (!name.empty()) p.setName(name);

                        cout << "Size moi (bo qua de giu): "; getline(cin, size);

                        if (!size.empty()) p.setSize(size);

                        cout << "Gioi tinh moi (bo qua de giu): "; getline(cin, gender);

                        if (!gender.empty()) p.setGender(gender);

                        cout << "Gia moi (0 de giu): "; cin >> price;

                        if (price > 0) p.setPrice(price);

                        cout << "So luong moi (0 de giu): "; cin >> stock;

                        cin.ignore();

                        if (stock > 0) p.setStock(stock);

                        saveProductsToFile(dsSanPham);

                        cout << "Cap nhat thanh cong.\n";

                        found = true;

                        break;

                    }

                }

                if (!found) cout << "Khong tim thay san pham.\n";

                break;

            }

            case 4: {

                string id;

                cout << "Nhap ma sp can xoa: ";

                getline(cin, id);

                bool found = false;

                for (auto it = dsSanPham.begin(); it != dsSanPham.end(); ++it) {

                    if (it->getId() == id) {

                        found = true;

                        cout << "Xac nhan xoa " << it->getName() << "? (y/n): ";

                        char x;

                        cin >> x;

                        cin.ignore();

                        if (x == 'y' || x == 'Y') {

                            dsSanPham.erase(it);

                            saveProductsToFile(dsSanPham);

                            cout << "Da xoa.\n";

                        }

                        else {

                            cout << "Huy xoa.\n";

                        }

                        break;

                    }

                }

                if (!found) cout << "Khong tim thay san pham.\n";

                break;

            }

            case 0:

                cout << "Quay ve menu chinh...\n";

                break;

            default:

                cout << "Lua chon khong hop le.\n";

            }

        } while (chon != 0);

    }

**-Các cấu trúc được sử dụng**

#### + **Cấu trúc Menu vòng lặp**

#### **+ Cấu trúc rẽ nhánh**

#### **+ Thao tác trên Vector**

#### **+ Nhập/Xuất cơ bản**

#### **+ Xử lý file**

#### **+ Kiểm tra điều kiện**

#### **+ Định dạng xuất (Formatting)**

**2.8 Hàm xem danh sách khách hàng**

void xemDanhSachKhachHang(const vector<Product>& dsSanPham) {

ifstream fin("customers.txt");

if (!fin) {

cout << "Khong mo duoc file customers.txt\n";

return;

}

cout << "\n--- DANH SACH KHACH HANG ---\n\n";

string line;

while (getline(fin, line)) {

size\_t pos1 = line.find(',');

size\_t pos2 = line.find(',', pos1 + 1);

size\_t pos3 = line.find(',', pos2 + 1);

size\_t pos4 = line.find(',', pos3 + 1);

if (pos1 == string::npos || pos2 == string::npos || pos3 == string::npos || pos4 == string::npos) {

cout << "Dong loi: " << line << endl;

continue;

}

string ten = line.substr(0, pos1);

string sdt = line.substr(pos1 + 1, pos2 - pos1 - 1);

string email = line.substr(pos2 + 1, pos3 - pos2 - 1);

string diachi = line.substr(pos3 + 1, pos4 - pos3 - 1);

string dsSp = line.substr(pos4 + 1);

if (!dsSp.empty() && dsSp.front() == '[') dsSp = dsSp.substr(1);

if (!dsSp.empty() && dsSp.back() == ']') dsSp.pop\_back();

cout << "Ten KH : " << ten << endl;

cout << "SDT : " << sdt << endl;

cout << "Email : " << email << endl;

cout << "Dia chi : " << diachi << endl;

size\_t start = 0;

double tongTien = 0;

while (true) {

size\_t end = dsSp.find(';', start);

string tenSp = dsSp.substr(start, end - start);

for (auto sp : dsSanPham) {

if (sp.getName() == tenSp) {

tongTien += sp.getPrice();

cout << "San pham da mua: " << sp.getName()

<< " - gia: " << fixed << setprecision(0) << sp.getPrice()

<< " VND\n";

break;

}

}

if (end == string::npos) break;

start = end + 1;

}

time\_t now = time(0);

tm t;

localtime\_s(&t, &now);

char buf[100];

strftime(buf, sizeof(buf), "%Y-%m-%d %H:%M", &t);

cout << "Ngay in bill : " << buf << endl;

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

}

fin.close();

}

**- Các cấu trúc đã sữ dụng**

+ File Handling (Xử lý file)

+ Loop (Vòng lặp - if - while - for)

+ Conditionals (Câu lệnh điều kiện)

+ String Processing (Xử lý chuỗi)

+ Output Formatting (Định dạng xuất)

**2.9 Hàm giao diện quản lí nhân viên và khách hàng:**

void menuQuanLyNhanVien\_KhachHang(vector<Seller>& dsSeller, const vector<Product>& dsSanPham) {

int chon;

do {

cout << "\n=========================== QUAN LY NHAN VIEN & KHACH HANG ===========================\n";

cout << "==\t\t\t 1. Xem ds seller \t\t\t\t\t ==" << endl;

cout << "==\t\t\t 2. Them seller (yeu cau tai khoan quan ly)\t\t ==" << endl;

cout << "==\t\t\t 3. Xoa seller (yeu cau tai khoan quan ly)\t\t ==" << endl;

cout << "==\t\t\t 4. Xem ds khach hang \t\t\t\t\t ==" << endl;

cout << "==\t\t\t 0. Quay lai \t\t\t\t\t\t ==" << endl;

cout << "======================================================================================\n";

cout << "Chon: ";

cin >> chon;

cin.ignore();

switch (chon) {

case 1: {

cout << "\n--- DANH SACH SELLER ---\n";

for (auto& s : dsSeller) {

s.display();

}

break;

}

case 2: {

cout << "\n--- THEM SELLER MOI ---\n";

if (!dangNhapQuanLy()) break;

string id, name;

cout << "Nhap ma seller: ";

getline(cin, id);

cout << "Nhap ten seller: ";

getline(cin, name);

dsSeller.push\_back(Seller(id, name));

saveSellersToFile(dsSeller);

cout << "Them thanh cong.\n";

break;

}

case 3: {

cout << "\n--- XOA SELLER ---\n";

if (!dangNhapQuanLy()) break;

string id;

cout << "Nhap ma seller can xoa: ";

getline(cin, id);

bool found = false;

for (auto it = dsSeller.begin(); it != dsSeller.end(); ++it) {

if (it->getID() == id) {

found = true;

cout << "Ban chac chan muon xoa " << it->getName() << "? (y/n): ";

char x;

cin >> x;

cin.ignore();

if (x == 'y' || x == 'Y') {

dsSeller.erase(it);

saveSellersToFile(dsSeller);

cout << "Da xoa seller.\n";

}

else {

cout << "Huy xoa.\n";

}

break;

}

}

if (!found) cout << "Khong tim thay seller.\n";

break;

}

case 4: {

xemDanhSachKhachHang(dsSanPham);

break;

}

case 0:

cout << "Quay ve menu chinh...\n";

break;

default:

cout << "Lua chon khong hop le!\n";

}

} while (chon != 0);

}

**- CÁC CẤU TRÚC ĐƯỢC SỬ DỤNG:**

+ Function (Hàm)

+ Loop (Vòng lặp)

+ Switch-case (Câu lệnh chọn lựa)

+ Function Calling (Gọi hàm)

+ User Input (Nhập dữ liệu từ người dùng)

+ Screen Control (Xử lý màn hình: system("cls"))

**2.10 Hàm tạo mã bill chạy tự động**

string taoMaBillTuDong() {

int so = 1;

string ma;

do {

ma = "A" + string(3 - to\_string(so).length(), '0') + to\_string(so);

ifstream fin("bill\_" + ma + ".txt");

if (!fin) break;

so++;

fin.close();

} while (true);

return ma;

}

**- Các cấu trúc được sử dụng:**

+ Loop (Vòng lặp - do\_while - if)

+ File Handling (Xử lý file)

+ String Processing (Xử lý chuỗi)

**2.11 Hàm tạo hóa đơn:**

void menuTaoHoaDon(vector<Product>& dsSanPham, const vector<Seller>& dsSeller) {

cout << "\n=== TAO HOA DON BAN HANG ===\n";

string sellerID;

cout << "Nhap ma nhan vien ban hang: ";

getline(cin, sellerID);

bool sellerTonTai = false;

string sellerName;

for (const auto& s : dsSeller) {

if (s.getID() == sellerID) {

sellerTonTai = true;

sellerName = s.getName();

break;

}

}

if (!sellerTonTai) {

cout << "Khong tim thay nhan vien nay.\n";

return;

}

string tenKH, sdt, email, diachi;

cout << "Nhap ten khach hang: "; getline(cin, tenKH);

cout << "Nhap SDT: "; getline(cin, sdt);

cout << "Nhap email: "; getline(cin, email);

cout << "Nhap dia chi: "; getline(cin, diachi);

cout << "\n--- DANH SACH SAN PHAM HIEN CO ---\n";

cout << left << setw(10) << "Ma" << setw(20) << "Ten"

<< setw(6) << "Size" << setw(10) << "Gioi tinh"

<< setw(10) << "Gia" << setw(10) << "Ton kho" << endl;

for (const auto& sp : dsSanPham) {

sp.display();

}

vector<Product> gioHang;

while (true) {

string id;

int sl;

cout << "\nNhap ma san pham (nhap 0 de ket thuc): ";

getline(cin, id);

if (id == "0") break;

bool timThay = false;

for (auto& sp : dsSanPham) {

if (sp.getId() == id) {

cout << "Nhap so luong: ";

cin >> sl;

cin.ignore();

if (sl <= 0 || sl > sp.getStock()) {

cout << "So luong khong hop le. Ton kho: " << sp.getStock() << endl;

break;

}

gioHang.push\_back(Product(sp.getId(), sp.getName(), sp.getSize(),

sp.getPrice(), sl, sp.getGender()));

sp.reduceStock(sl);

timThay = true;

cout << "Da them " << sl << " " << sp.getName() << " vao gio hang.\n";

break;

}

}

if (!timThay) {

cout << "Khong tim thay san pham.\n";

}

}

if (gioHang.empty()) {

cout << "Khong co san pham nao duoc chon. Huy tao hoa don.\n";

return;

}

string maBill = taoMaBillTuDong();

Bill bill(maBill, tenKH, sdt, email, diachi, sellerID, gioHang);

danhSachHoaDon.push\_back(bill);

bill.writeToFile();

bill.saveBillToFile();

saveProductsToFile(dsSanPham);

cout << "\n=== HOA DON VUA TAO ===\n";

ifstream fin("bill\_" + maBill + ".txt");

if (fin) {

string line;

while (getline(fin, line)) {

cout << line << endl;

}

fin.close();

}

cout << "\nDa tao hoa don thanh cong!" << endl;

cout << "Ma hoa don: " << maBill << endl;

cout << "Nhan vien: " << sellerName << endl;

cout << "Tong cong: " << fixed << setprecision(0)

<< bill.tinhTongTien() << " VND" << endl;

}

**- CÁC CẤU TRÚC ĐƯỢC SỬ DỤNG:**

+ Loop (Vòng lặp - if - while)

+ Conditionals (Câu lệnh điều kiện)

+ File Handling (Xử lý file)

+ String Processing (Xử lý chuỗi)

+ Output Formatting (Định dạng xuất)

**2.12 Hàm tính tổng doanh thu:**

void tinhTongTienTheoThoiGian(const vector<Bill>& danhSachHoaDon) {

if (danhSachHoaDon.empty()) {

cout << "Khong co hoa don nao trong he thong!\n";

return;

}

cout << "\n=== TINH TONG TIEN THEO THOI GIAN ===";

cout << "\n Theo ngay (dd/mm/yyyy)";

cout << "Nhap thoi gian: ";

string input;

getline(cin, input);

int year = 0, month = 0, day = 0;

char sep;

istringstream iss(input);

if (sscanf(input.c\_str(), "%d/%d/%d", &day, &month, &year) != 3) {

cout << "Dinh dang ngay khong hop le!\n";

return;

}

if (day < 1 || day > 31 || month < 1 || month > 12 || year < 1900) {

cout << "Ngay thang nam khong hop le!\n";

return;

}

double tongTien = 0;

int soHoaDon = 0;

for (const auto& bill : danhSachHoaDon) {

time\_t billTime = bill.getBillTime();

tm t;

localtime\_s(&t, &billTime);

bool match = false;

match = (t.tm\_mday == day && t.tm\_mon + 1 == month && t.tm\_year + 1900 == year);

if (match) {

tongTien += bill.tinhTongTien();

soHoaDon++;

}

}

cout << "\n=== KET QUA ===";

cout << "\nThoi gian: " << input;

cout << "\nTong so hoa don: " << soHoaDon;

cout << "\nTong tien: " << fixed << setprecision(0) << tongTien << " VND\n";

}

**-CÁC CẤU TRÚC ĐƯỢC SỬ DỤNG**:

+ Loop (Vòng lặp-if\_else)

+ Conditionals (Câu lệnh điều kiện)

+ String Processing (Xử lý chuỗi)

+ Output Formatting (Định dạng xuất)

+ Time Handling (Xử lý thời gian)

**2.13 Hàm đổi mật khẩu ADMIN:**

void menuDoiMatKhau() {

cout << "\n=== DOI MAT KHAU QUAN LY ===\n";

string user, oldPass;

cout << "Nhap ten tai khoan: ";

getline(cin, user);

if (user != "TvQ") {

cout << "Sai ten tai khoan. Chi cho phep doi mat khau cua tai khoan quan ly.\n";

return;

}

cout << "Nhap mat khau cu: ";

oldPass = "";

char c;

while ((c = \_getch()) != '\r') {

if (c == '\b' && !oldPass.empty()) {

oldPass.pop\_back();

cout << "\b \b";

}

else {

oldPass.push\_back(c);

cout << '\*';}

}

cout << endl;

if (oldPass != matKhauQL) {

cout << "Mat khau sai.\n";

return;

}

string newPass;

cout << "Nhap mat khau moi: ";

newPass = "";

while ((c = \_getch()) != '\r') {

if (c == '\b' && !newPass.empty()) {

newPass.pop\_back();

cout << "\b \b";

}

else {

newPass.push\_back(c);

cout << '\*';

}

}

cout << endl;

matKhauQL = newPass;

cout << "Doi mat khau thanh cong.\n";

}

**- CÁC CẤU TRÚC ĐƯỢC SỬ DỤNG:**

+ Conditionals (Câu lệnh điều kiện)

+ String Processing (Xử lý chuỗi)

+ Output Formatting (Định dạng xuất)

+ Input Masking (Ẩn ký tự nhập – dùng \_getch())

**2.14 ) Hàm trang chính**

void trangChinh() {

vector<Product> dsSanPham;

vector<Seller> dsSeller;

loadProductsFromFile(dsSanPham);

loadSellersFromFile(dsSeller);

int chon;

do {

cout << "\n================================= Quan ly cua Hang Q&T ===============================\n";

cout << "===================================== MENU CHINH =====================================\n";

cout << "==" << right << setw(48) << "1. Quan ly san pham" << setw(37) << "==\n";

cout << "==" << right << setw(59) << "2. Quan ly seller & khach hang" << setw(26) << "==\n";

cout << "==" << right << setw(52) << "3. Tao hoa don ban hang" << setw(33) << "==\n";

cout << "==" << right << setw(52) << "4. Doi mat khau quan ly" << setw(33) << "==\n";

cout << "==" << right << setw(66) << "5. Tong doanh thu theo ngay thang nam" << setw(19) << "==\n";

cout << "==" << right << setw(37) << "0. Thoat" << setw(48) << "==\n";

cout << "======================================================================================\n";

cout << "======================================================================================\n";

cout << "Chon: ";

cin >> chon;

cin.ignore();

switch (chon) {

case 1:

menuQuanLySanPham(dsSanPham);

break;

case 2:

menuQuanLyNhanVien\_KhachHang(dsSeller, dsSanPham);

break;

case 3:

menuTaoHoaDon(dsSanPham, dsSeller);

break;

case 4:

menuDoiMatKhau();

break;

case 5:

tinhTongTienTheoThoiGian(danhSachHoaDon);

break;

case 0:

cout << "Thoat chuong trinh...\n";

break;

default:

cout << "Lua chon khong hop le.\n";

}

} while (chon != 0);

}

-CÁC CẤU TRÚC ĐƯỢC SỬ DỤNG:

+ Loop (Vòng lặp - switch\_case - do\_while)

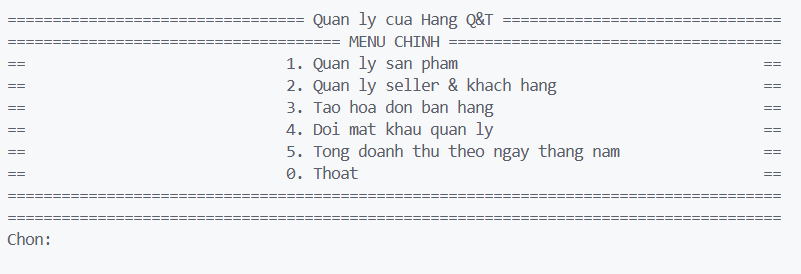
+ Conditionals (Câu lệnh điều kiện)

+ Output Formatting (Định dạng xuất)

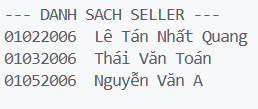
+ File Handling (Xử lý file)

1. **Kết quả chính khi chạy thử chương trình**

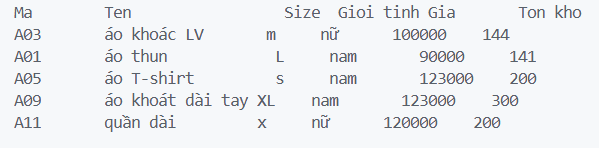
**3.1 Màn hình hiển thị trang chính**



**3.2 Danh sách nhân viên**



**3.3 Danh sách sản phẩm hiện có**

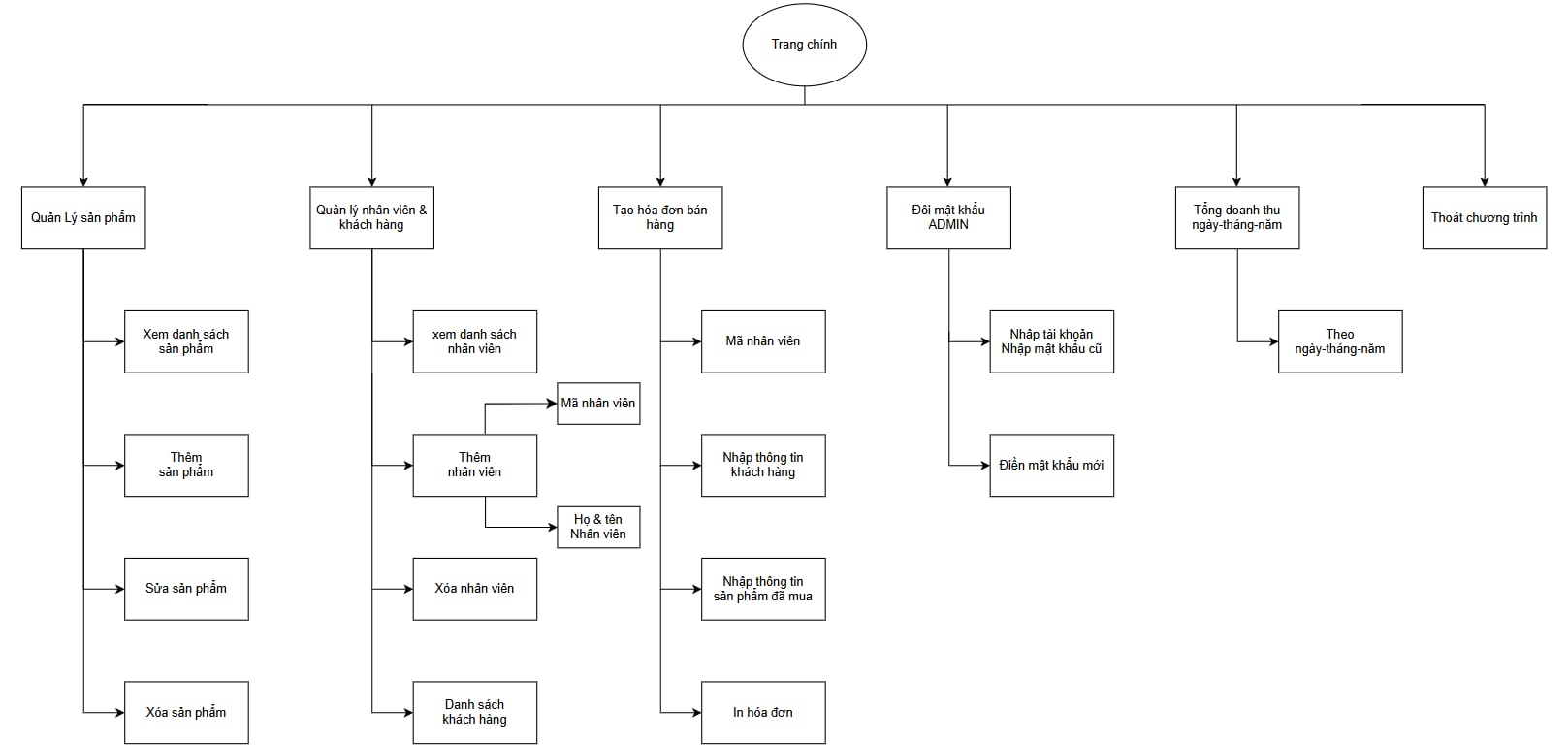


**3.4 Hiển thị hóa đơn mua hàng**

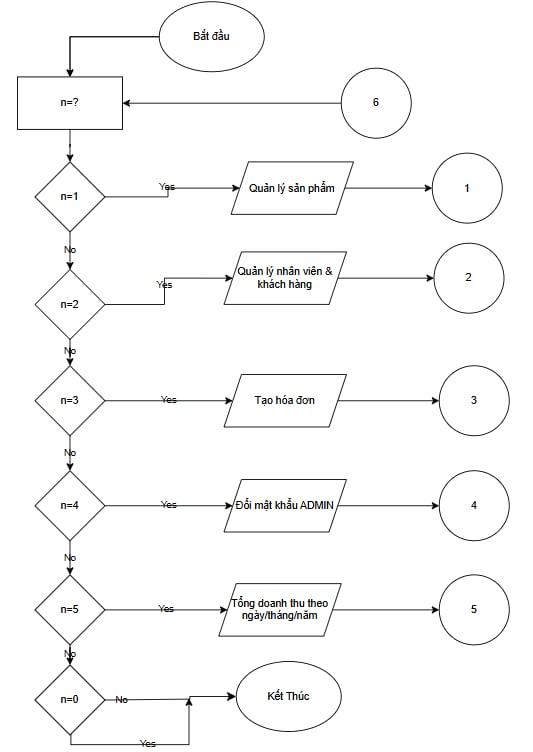


**C.** **LƯU ĐỒ GIẢI THUẬT**

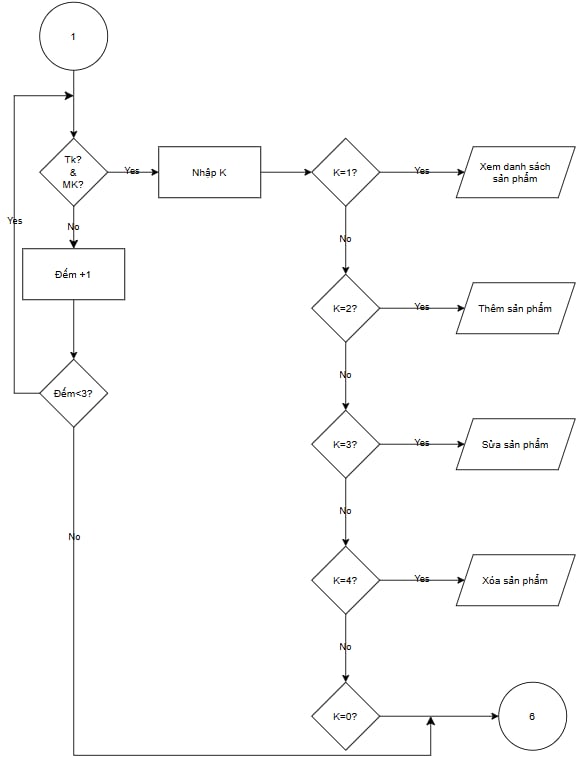
**1. Lưu đồ tổng quát chương trình**



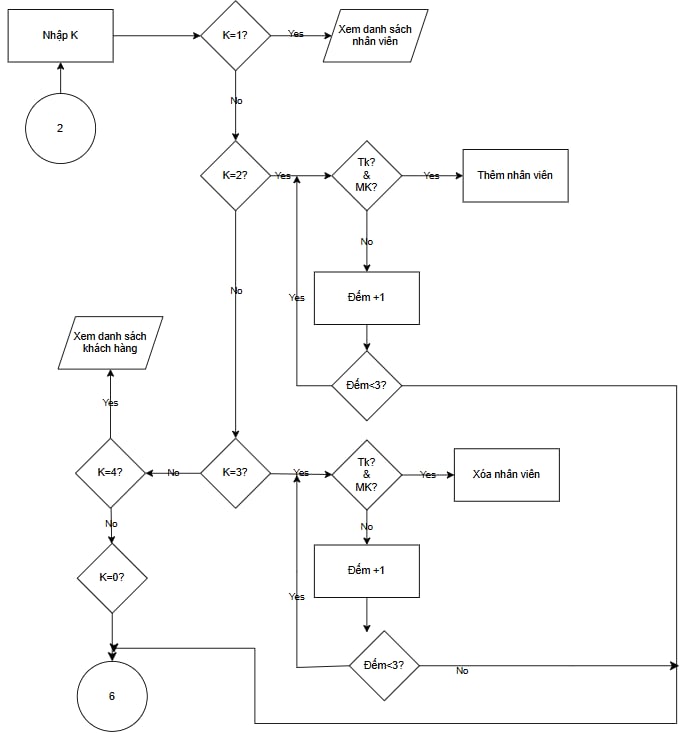
**2. Lưu đồ Trang chính**



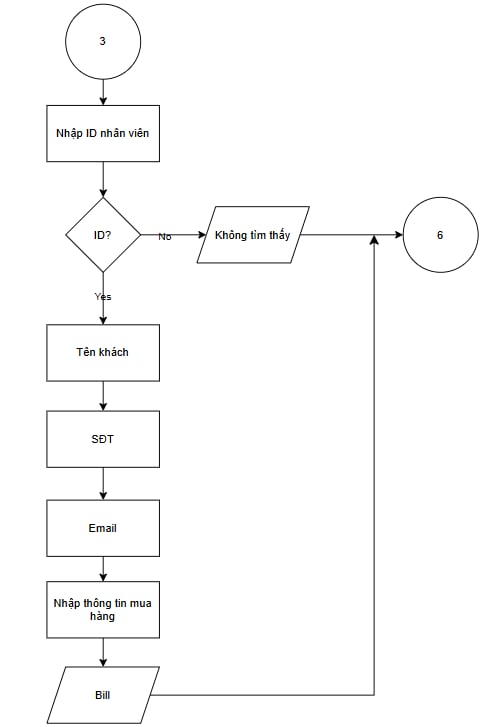
**3.Lưu đồ quản lý sản phẩm**

****

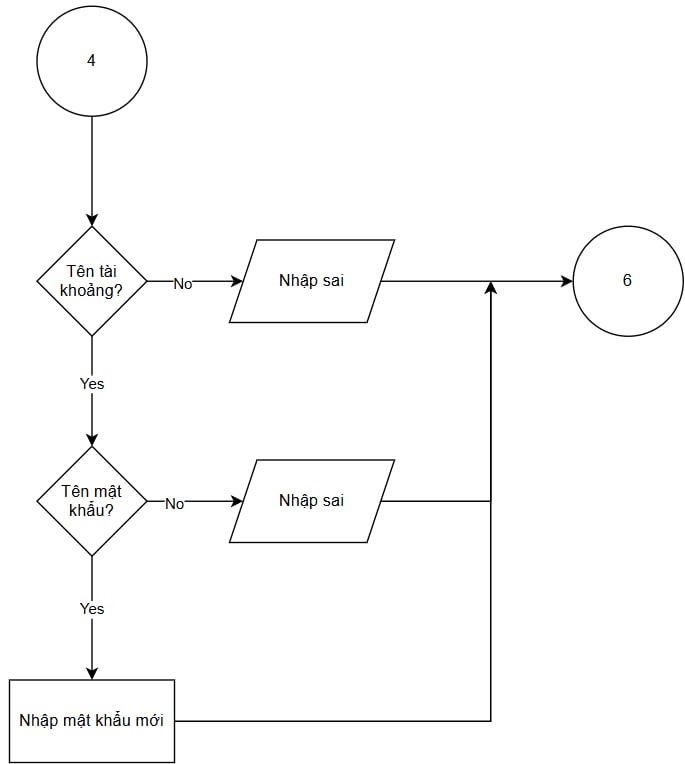
**4. Lưu đồ quản lý sản nhân viên và khách hàng**



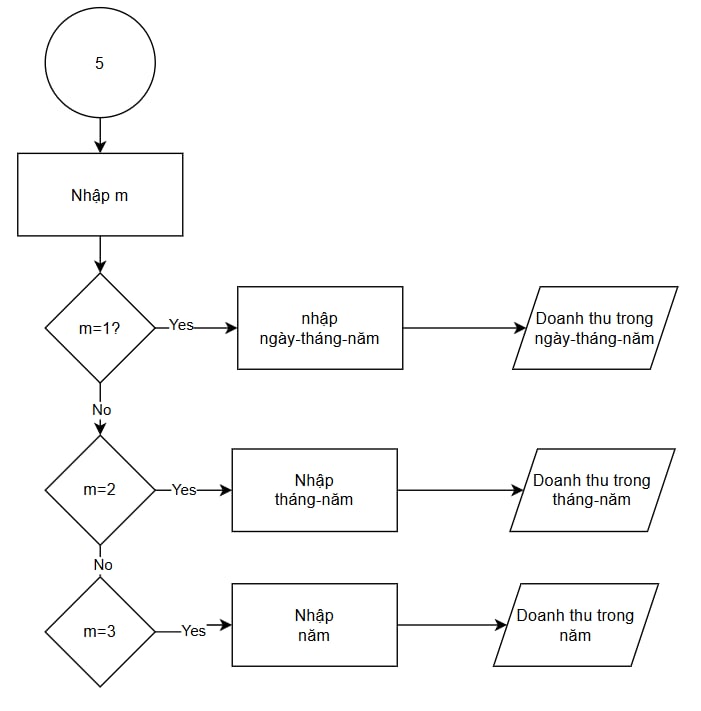
**5.Lưu đồ tạo hóa đơn**

****

**6. Lưu đồ đổi mật khẩu ADMIN**

****

**7. Lưu đồ tổng doanh thu**

****

**D. PHÂN TÍCH VÀ ĐÁNH GIÁ ĐỒ ÁN QUẢN LÝ CỬA HÀNG THỜI TRANG**

**1. Tổng quan hệ thống**

**Ngôn ngữ**: C++.

**Mục đích**: Quản lý bán hàng thời trang với các chức năng cơ bản như quản lý sản phẩm, nhân viên, khách hàng và hóa đơn.

**2. Các chức năng chính**

***a. Quản lý sản phẩm***

- Xem danh sách sản phẩm.

- Thêm/sửa/xóa sản phẩm.

- Lưu vào file products.txt.

***b. Quản lý nhân viên (Seller)***

- Xem danh sách nhân viên.

- Thêm/xoá nhân viên (yêu cầu quyền admin).

- Lưu vào file sellers.txt.

***c. Quản lý khách hàng***

- Xem danh sách khách hàng từ file customers.txt.

- Hiển thị thông tin mua hàng của khách.

***d. Tạo hóa đơn***

- Tự động sinh mã hóa đơn (A001, A002,...).

- Lưu hóa đơn vào file bill\_[mã].txt và tổng hợp vào bills\_data.txt.

- Cập nhật tồn kho sau khi bán.

***e. Thống kê doanh thu***

Tính tổng tiền theo ngày/tháng/năm.

***f. Hệ thống đăng nhập***

- Đăng nhập quản lý với tài khoản/mật khẩu.

- Chức năng đổi mật khẩu.

**3. Ưu điểm**

- Code có cấu trúc rõ ràng, sử dụng OOP.

- Lưu dữ liệu vào file text đơn giản.

- Giao diện console dễ theo dõi.

- Có xử lý nhập mật khẩu ẩn (sử dụng \_getch()).

**4. Hạn chế và đề xuất cải tiến**

***a. Quản lý dữ liệu***

- Chưa có cơ chế sao lưu/phục hồi dữ liệu.

**Đề xuất**: Thêm chức năng backup/restore.

***b. Giao diện***

- Giao diện console cơ bản.

**Đề xuất**: Có thể phát triển thành ứng dụng GUI.

***c. Chức năng mở rộng***

- Thiếu báo cáo thống kê chi tiết.

- Chưa có quản lý nhập hàng.

**Đề xuất**: Thêm các báo cáo và quản lý kho.

**5. Kết luận**

Đồ án hiện tại đã đáp ứng được các yêu cầu cơ bản của hệ thống quản lý bán hàng. Code có cấu trúc tốt và dễ dàng mở rộng. Cần bổ sung thêm các chức năng nâng cao và cải thiện bảo mật để hoàn thiện hệ thống.

**TÀI LIỆU THAM KHẢO**

1. 28Tech Blog, Trang web chia sẻ kiến thức lập trình.

Truy cập tại: [https://28tech.com.vn/](https://28tech.com.vn/" \t "_new)

1. 28Tech. (n.d.). Kênh YouTube học lập trình. YouTube.

Truy cập tại: [https://www.youtube.com/@28tech](https://www.youtube.com/@28tech" \t "_new)

1. Learn C++. (n.d.). Skill up with our free tutorials.

Truy cập tại: [https://www.learncpp.com/](https://www.learncpp.com/" \t "_new)

1. Tự học C. (n.d.). Trang web học C/C++ online miễn phí.

Truy cập tại: [http://c.tuhoc.cc/](http://c.tuhoc.cc/" \t "_new)

1. Vũ Nguyễn Coder. (n.d.). Kênh YouTube học lập trình C/C++. YouTube.

Truy cập tại: [https://www.youtube.com/@VuNguyenCoder](https://www.youtube.com/@VuNguyenCoder" \t "_new)