BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO

BÁO CÁO MÔN GIẢI THUẬT VÀ CẤU TRÚC DỮ LIỆU

ĐỀ TÀI: THIẾT KẾ CẤU TRÚC DỮ LIỆU, THUẬT TOÁN VÀ XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ BÁN THIẾT BỊ ĐIỆN TỬ

Mã môn học	**ALDS335764**
Mã lớp học phần	**ALDS335764_22_2_02CLC**
GVHD:	**PGS TS. Hoàng Văn Dũng**
Sinh viên thực hiện:	**Lê Phú Cường – 20119203**
	Lâm Tấn Phát – 20119264
	Nguyễn Trần Thành Trung - 20119302

BẢNG PHÂN CÔNG NHIỆM VỤ

STT	Nhiệm vụ	SV Thực hiện	Tiến độ
1	Lên ý tưởng, sắp xếp thuật toán, cấu trúc dữ liệu	Lê Phú Cường	100%
2	Viết hàm quản lý danh sách sản phẩm: • Hiển thị danh sách sản phẩm • Thêm sản phẩm • Tìm kiếm sản phẩm	Nguyễn Trần Thành Trung	100%
3	Viết hàm quản lý danh sách sản phẩm: • Chỉnh sửa thông tin sản phẩm • Xoá sản phẩm • Sắp xếp sản phẩm	Lê Phú Cường	100%
4	Viết hàm quản lý hoá đơn + Kết luận	Lâm Tấn Phát	100%
5	Test + Debug	Lâm Tấn Phát	100%
6	Viết báo cáo chương 1, 3, 4	Lê Phú Cường	100%
7	Viết báo cáo chương 2	Nguyễn Trần Thành Trung	100%

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU

1.1. Những vấn đề đặt ra và giải pháp

Cửa hàng bán linh kiện điện tử là nơi để phục vụ cho mọi khách hàng có nhu cầu tìm hiểu và mua sắm các sản phẩm trong lĩnh vực điện tử. Tại đây, khách hàng có thể tìm thấy các loại linh kiện phù hợp với nhu cầu và chuyên môn của mình và mua về để nghiên cứu, sử dụng hoặc sửa chữa theo nhu cầu của mình. Tuy nhiên, việc quản lý và đưa vào sử dụng hàng ngàn sản phẩm linh kiện trong cửa hàng là một công việc rất phức tạp. Bên cạnh việc nhập hàng, cửa hàng còn phải quản lý việc bán hàng đối với các khách hàng đến mua và tiếp nhận trả hàng khi xảy ra lỗi, hư hỏng của nhà sản xuất. Để giúp nhân viên cửa hàng thực hiện công việc này một cách dễ dàng hơn, cần có một chương trình quản lý đáp ứng các yêu cầu và tiêu chuẩn của cửa hàng.

Chương trình quản lý cửa hàng sẽ hỗ trợ cho việc quản lý các thông tin liên quan đến linh kiện, bao gồm thông tin chi tiết về linh kiện, thông tin về nhà sản xuất và giá cả, danh sách các sản phẩm trong kho, số lượng và trạng thái bán hàng của linh kiện, thông tin xuất nhập kho v.v. Chương trình sẽ cung cấp các tính năng để nhập sản phẩm mới, xóa sản phẩm, tìm kiếm và chỉnh sửa thông tin sản phẩm.

Ngoài ra, chương trình quản lý cửa hàng cũng sẽ giúp quản lý việc bán hàng có thể thực hiện việc quản lý danh sách các sản phẩm được bán ra và hoàn trả lại cửa hàng. Chương trình cũng cung cấp các tính năng để kiểm tra xem sản phẩm mà khách hàng muốn mua còn hay không và kiểm tra xem những sản phẩm đã hết hạn bảo hành để thực hiện các chính sách đổi trả sản phẩm.

1.2. Chức năng

Các chức năng chính của chương trình:

- Quản lý linh kiện (ID sản phẩm, tên sản phẩm, phân loại sản phẩm, số lượng sản phẩm, giá thành sản phẩm, phần trăm giảm giá)

Hiển thị danh sách sản phẩm bao gồm thông tin để theo dõi tình trạng của sản phẩm.

Thêm sản phẩm khi có sản phẩm mới.

Tìm kiếm sản phẩm để dễ dàng biết tình trạng sản phẩm với mã ID hoặc tên sản phẩm.

Chỉnh sửa thông tin sản phẩm khi có sai sót về ID sản phẩm, tên sản phẩm, số lượng và phân loại.

Xoá sản phẩm khi linh kiện đã không còn nhập hàng.

Sắp xếp đúng thứ tự sản phẩm (theo chữ cái A-Z, theo thứ tự ID từ nhỏ nhất đến lớn nhất, …).

- Quản lý hoá đơn

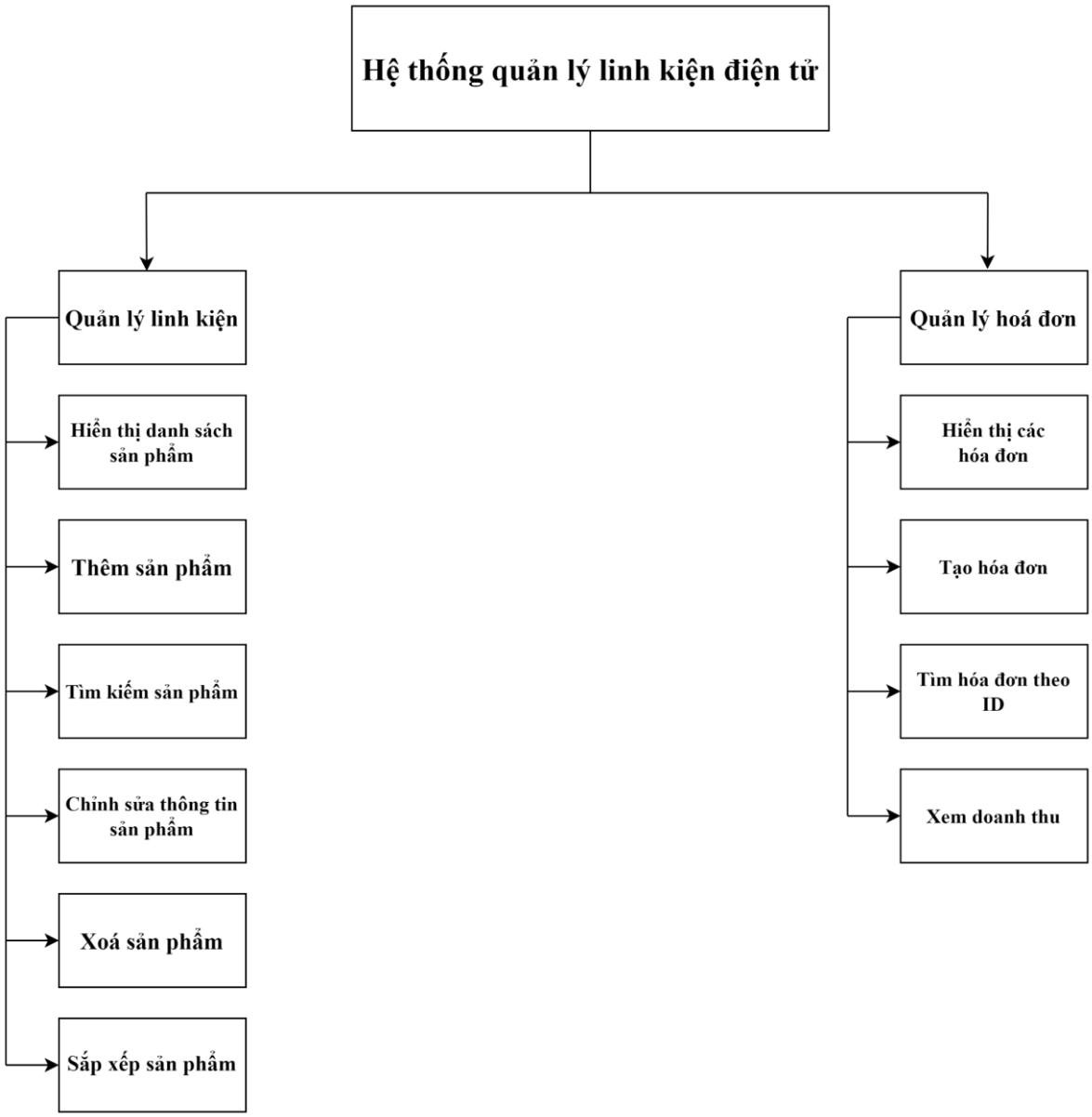
Chọn hàng: cho phép nhân viên chọn các sản phẩm và thêm chúng vào giỏ hàng.

Hiển thị toàn bộ hóa đơn: cho phép nhân viên xem tất cả sản phẩm trong giỏ hàng.

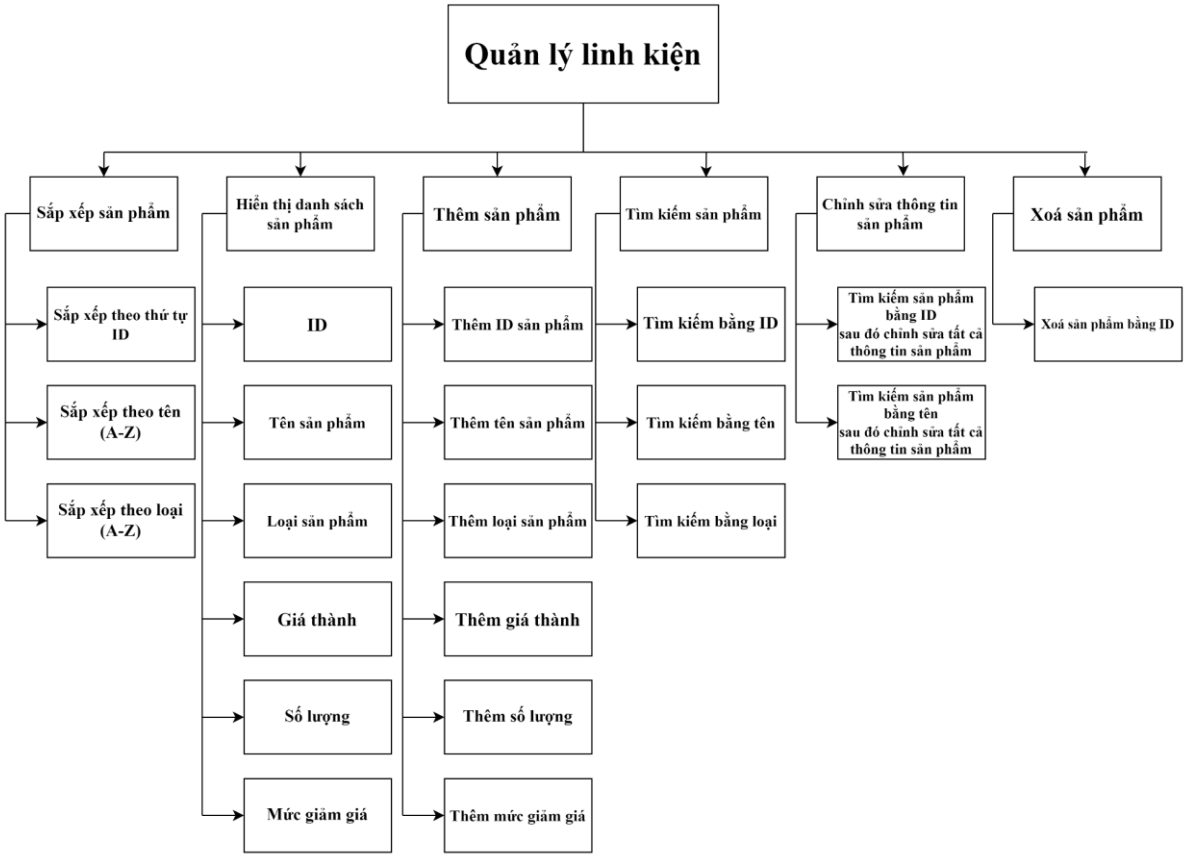
Tìm hoá đơn: cho phép nhân viên xem lại các sản phẩm đã chọn và số tiền cần thanh toán theo mã ID của hóa đơn. Hóa đơn thường bao gồm thông tin về các sản phẩm đã mua, số tiền phải trả, ngày mua.

Xem doanh thu: cho phép nhân viên xem doanh thu cửa hàng theo ngày, tháng hoặc năm mong muốn.

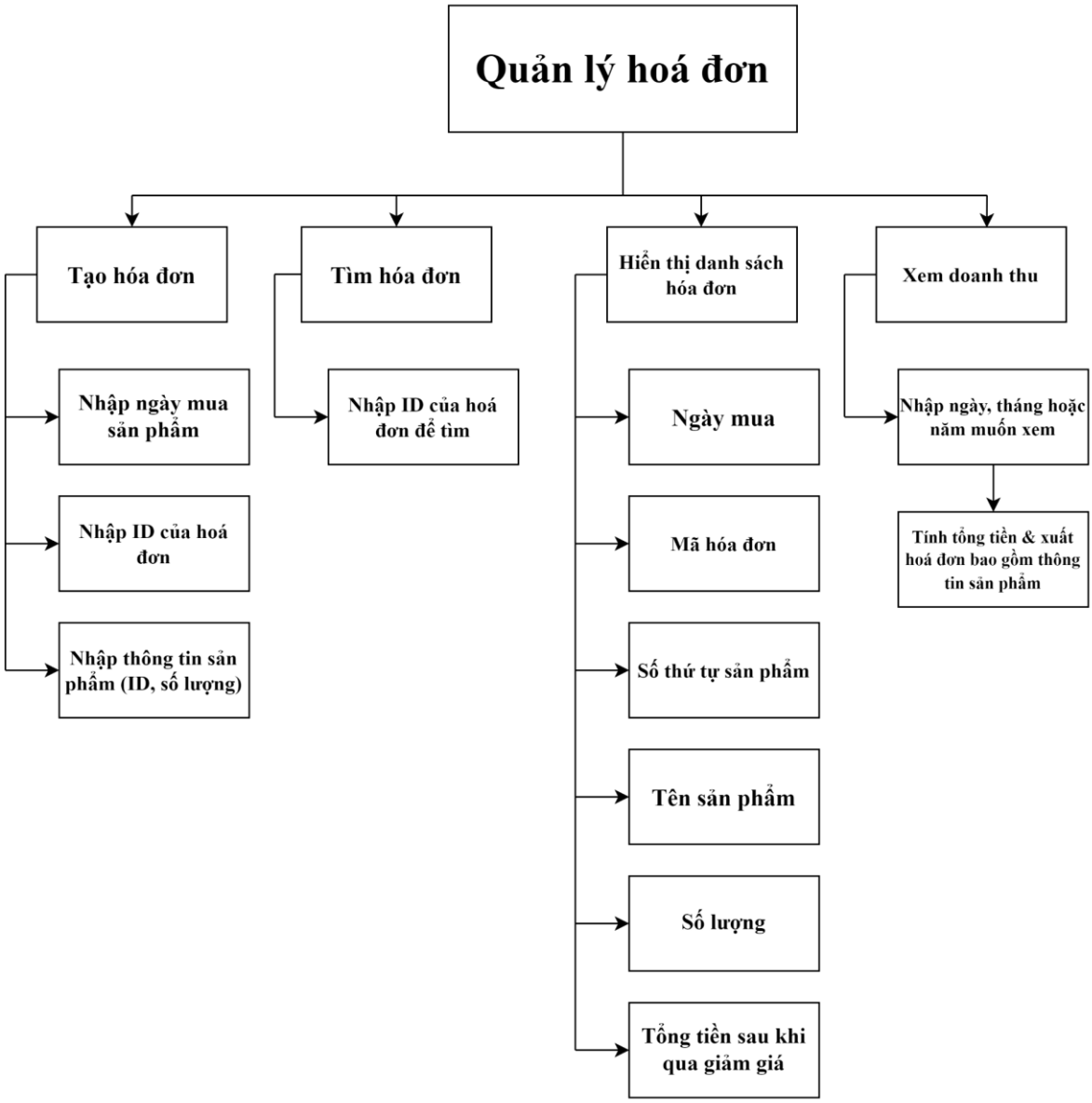
1.3. Biểu đồ phân rã chức năng



Hình 1.1. Biểu đồ phân rã chức năng của hệ thống



Hình 1.2. Biểu đồ chức năng của quản lý linh kiện



Hình 1.3. Biểu đồ chức năng của quản lý hoá đơn

1.4. Phạm vi thực hiện dự án

Dự án được xây dựng nhằm phục vụ các công việc quản lý thông tin các linh kiện, như tên sản phẩm, giá cả, số lượng tồn kho, v.v., quản lý thông tin khách hàng như tên, địa chỉ, số điện thoại, email, lịch sử mua hàng, v.v., quản lý đơn hàng từ khi khách hàng đặt hàng cho đến khi giao hàng xong. Thông tin trong đơn hàng bao gồm: sản phẩm, số lượng, giá, địa chỉ giao hàng, v.v.

1.5. Đối tượng thực hiện

Nhân viên kinh doanh: đối tượng này sẽ sử dụng hệ thống để quản lý thông tin sản phẩm, khách hàng, đơn hàng, hóa đơn, giá cả, giảm giá, v.v.

Nhân viên bán hàng: đối tượng này sẽ sử dụng hệ thống để thực hiện các hoạt động bán hàng như tạo hóa đơn, in hóa đơn, quản lý sản phẩm, v.v.

Nhân viên kho: đối tượng này sẽ sử dụng hệ thống để quản lý thông tin hàng tồn kho, nhập kho, xuất kho, kiểm kê hàng hóa, v.v.

CHƯƠNG 2: PHÂN TÍCH LẬP TRÌNH HỆ THỐNG QUẢN LÝ THÔNG TIN CỦA MỘT CỬA HÀNG BÁN LINH KIỆN ĐIỆN TỬ

2.1. Phân tích cấu trúc

Mô tả cấu trúc được yêu cầu, chọn cấu trúc dữ liệu để thể hiện, khai báo và định nghĩa cấu trúc:

2.1.1. Định nghĩa cấu trúc sản phẩm – linh kiện:

```
struct Product
{
    string productID;
    string name;
    string type;
    string field;
    string keyword;
    int quantity;
    float price;
    float discount;
    Product *next;
};
```

Thông tin sản phẩm linh kiện cần quản lý gồm:

- Product ID: Mã linh kiện
- Name: Tên linh kiện
- Type: Loại linh kiện
- Quantity: Số lượng
- Price: Giá thành
- Discount: Sản phẩm giảm giá

2.1.2. Định nghĩa cấu trúc hóa đơn:

```
struct Customer {
    string keyword;
    string field;
    string date;
    string billID;
    int serial;
    string productname;
    int productquantity;
    int total;
    Customer *next;
};
```

Thông tin khách hàng gồm:

- Date: Ngày mua hàng
- Bill ID: Mã hóa đơn
- Serial: Số thứ tự sản phẩm trong hóa đơn
- Product name: Tên sản phẩm được mua
- Product quantity: Số lượng của sản phẩm
- Total: Số tiền của sản phẩm x số lượng sau khi qua giảm giá

Cấu trúc dữ liệu hỗ trợ quản lý thông tin các linh kiện:

- Product ID: Chuỗi ký tự
- Name: Chuỗi ký tự
- Type: Chuỗi ký tự
- Product Field: Chuỗi ký tự
- Product Keyword: Chuỗi ký tự
- Quantity: Số nguyên không âm
- Price: Số thực dương
- Discount: Số thực dương
- Customer Keyword: Chuỗi ký tự
- Customer Field: Chuỗi ký tự
- Date: Chuỗi ký tự
- Bill ID: Chuỗi ký tự
- Serial: Số nguyên không âm
- Product Name: Chuỗi ký tự
- Product Quantity: Số nguyên không âm
- Total: Số thực dương

2.2. Quản lý danh sách sản phẩm

2.2.1. Hiển thị danh sách sản phẩm

Hàm hiển thị danh sách sản phẩm:

```
void displayProduct()
{
    Product *temp = head;
    cout << "+--------------+--------------------------------+--------------+--------------+--------------+--------------+" << endl;
    cout << "|   ProductID  |              Name              |     Type     |     Price    |   Quantity   |   Discount   |" << endl;
    cout << "+--------------+--------------------------------+--------------+--------------+--------------+--------------+" << endl;
    while (temp != nullptr)
    {
        cout << "| "  << setw(13) << left << temp->productID
             << " | " << setw(31) << left << temp->name
             << " | " << setw(13) << left << temp->type
             << " | " << setw(13) << left << temp->price
             << " | " << setw(13) << left << temp->quantity
             << " | " << setw(14) << left << temp->discount << "|" << endl;
        cout << "+--------------+--------------------------------+--------------+--------------+--------------+--------------+" << endl;
        temp = temp->next;
    }
}
```

Hình 2.1. Hàm hiển thị danh sách sản phẩm

Hàm displayProduct() được sử dụng để hiển thị danh sách sản phẩm đã được lưu trữ trong danh sách liên kết. Đầu tiên, hàm sẽ bắt đầu từ đầu danh sách bằng cách tạo một con trỏ temp và gán nó bằng head. Sau đó, hàm sẽ hiển thị một bảng chứa tiêu đề của các trường thông tin sản phẩm bao gồm ID, tên, loại, giá thành, số lượng và mức giảm giá. Sau đó, hàm sẽ duyệt qua toàn bộ danh sách liên kết, và mỗi lần duyệt qua một sản phẩm, hàm sẽ in ra các thông tin tương ứng của sản phẩm đó theo định dạng bảng. Cuối cùng, hàm sẽ cập nhật con trỏ temp để trỏ đến sản phẩm tiếp theo trong danh sách liên kết cho đến khi temp trỏ tới nullptr (kết thúc danh sách liên kết).

2.2.2. Thêm sản phẩm

Hàm thêm sản phẩm như sau:

```
void addProduct(string productID, string name, string type, int price, int quantity, int discount)
{
    Product *newProduct = new Product;
    newProduct->productID = productID;
    newProduct->name = name;
    newProduct->type = type;
    newProduct->price = price;
    newProduct->quantity = quantity;
    newProduct->discount = discount;
    newProduct->next = head;
    head = newProduct;
}
```

Hình 2.2. Hàm thêm sản phẩm

Hàm "addProduct" được sử dụng để thêm một sản phẩm mới vào danh sách liên kết đơn của các sản phẩm. Cụ thể, hàm này có các tham số đầu vào là "productID", "name", "type", "price", "quantity" và "discount", lần lượt là thông tin về mã sản phẩm,

tên sản phẩm, loại sản phẩm, giá sản phẩm, số lượng sản phẩm và mức giảm giá của sản phẩm.

Hàm bắt đầu bằng cách tạo một con trỏ "newProduct" đến một sản phẩm mới và đặt các giá trị của các thông tin sản phẩm vào các trường tương ứng của sản phẩm đó.

Sau đó, sản phẩm mới được đặt làm "head" của danh sách.

2.2.3. Tìm kiếm sản phẩm

Hàm tìm kiếm sản phẩm như sau:

```
void findProduct(string keyword, string field)
{
    vector<Product*> productsToFind = searchProduct(keyword, field);
    if (productsToFind.empty())
    {
        cout << "  No product found in the store." << endl;
        return;
    }
    else{
        cout << "+--------------+--------------------------------+--------------+--------------+--------------+--------------+" << endl;
        cout << "|   ProductID  |              Name              |      Type    |     Price    |    Quantity  |    Discount  |" << endl;
        cout << "+--------------+--------------------------------+--------------+--------------+--------------+--------------+" << endl;
        for (Product* product : productsToFind)
        {
            cout << "| " << setw(13) << left << product->productID
                 << " | " << setw(31) << left << product->name
                 << " | " << setw(13) << left << product->type
                 << " | " << setw(13) << left << product->price
                 << " | " << setw(13) << left << product->quantity
                 << " | " << setw(13) << left << product->discount << " |" << endl;
        cout << "+--------------+--------------------------------+--------------+--------------+--------------+--------------+" << endl;
        }
    }
}
```

Hình 2.3. Hàm tìm kiếm sản phẩm

Hàm findProduct(string keyword, string field) có chức năng tìm kiếm sản phẩm trong danh sách các sản phẩm của cửa hàng dựa trên từ khóa và trường dữ liệu được chỉ định.

Hàm này sẽ gọi hàm searchProduct(keyword, field) để tìm kiếm các sản phẩm phù hợp và lưu chúng vào một vector các con trỏ sản phẩm.

Nếu không tìm thấy sản phẩm nào phù hợp, hàm sẽ in ra thông báo "No product found in the store."

Nếu tìm thấy sản phẩm phù hợp, hàm sẽ hiển thị danh sách các sản phẩm đó trên màn hình bằng cách sử dụng các lệnh cout và setw để định dạng dữ liệu sao cho đẹp mắt. Sau đó, hàm sẽ sử dụng vòng lặp for để duyệt qua danh sách các sản phẩm phù hợp và in ra thông tin chi tiết của từng sản phẩm trên màn hình.

2.2.4. Chỉnh sửa thông tin sản phẩm

Hàm chỉnh sửa thông tin sản phẩm như sau:

```
void adjustProduct(string keyword, string field, string newID, string newname_product, string newtype, int newprice, int newquantity, int newdiscount)
{
    vector<Product*> productsToAdjust = searchProduct(keyword, field);
    if (productsToAdjust.empty()) {
        cout << "  No products found to adjust." << endl;
        return;
    }
    for (Product* product : productsToAdjust) {
        product->productID = newID;
        product->name = newname_product;
        product->type = newtype;
        product->price = newprice;
        product->quantity = newquantity;
        product->discount = newdiscount;
    }
    cout << "  Product information updated successfully." << endl;
}
```

Hình 2.4. Hàm chỉnh sửa thông tin sản phẩm

Hàm adjustProduct() là một hàm để sửa đổi thông tin của một hoặc nhiều sản phẩm trong cửa hàng. Đầu tiên, nó sẽ tìm kiếm và trả về một danh sách các sản phẩm khớp với từ khóa tìm kiếm trong trường dữ liệu được chỉ định. Nếu không có sản phẩm nào khớp, hàm sẽ in ra thông báo "Không tìm thấy sản phẩm nào để điều chỉnh" và kết thúc. Nếu tìm thấy các sản phẩm, hàm sẽ sửa đổi thông tin của chúng bằng các giá trị mới được truyền vào, sau đó in ra thông báo "Thông tin sản phẩm đã được cập nhật thành công".

2.2.5. Xoá sản phẩm

Hàm xoá sản phẩm như sau:

```
void deleteProduct(string productID)
{
    if (head == nullptr)
    {
        cout << "  The list is empty." << endl;
        return;
    }
    Product *temp = head;
    Product *prev = nullptr;
    while (temp != nullptr && temp->productID != productID)
    {
        prev = temp;
        temp = temp->next;
    }
    if (temp == nullptr)
    {
        cout << "  Product not found in the library." << endl;
        return;
    }
    if (prev == nullptr)
    {
        head = head->next;
    }
    else
    {
        prev->next = temp->next;
    }
    delete temp;
    cout << "  Product deleted successfully." << endl;
}
```

Hình 2.5. Hàm xoá sản phẩm

Hàm deleteProduct có chức năng xóa một sản phẩm khỏi danh sách sản phẩm. Đầu vào của hàm là productID là mã sản phẩm cần xóa.

Trước khi xóa, hàm sẽ kiểm tra xem danh sách sản phẩm có rỗng không. Nếu rỗng, hàm sẽ thông báo cho người dùng biết rằng danh sách sản phẩm đang trống và không thực hiện thao tác xóa.

Nếu danh sách sản phẩm không rỗng, hàm sẽ duyệt qua danh sách để tìm sản phẩm cần xóa. Nếu sản phẩm không tồn tại trong danh sách, hàm sẽ thông báo cho người dùng biết rằng sản phẩm không được tìm thấy và không thực hiện thao tác xóa.

Nếu sản phẩm được tìm thấy, hàm sẽ xóa sản phẩm bằng cách liên kết node trước của sản phẩm đó với node sau của sản phẩm đó. Nếu sản phẩm cần xóa là nút đầu tiên của danh sách sản phẩm, con trỏ đầu danh sách sẽ được cập nhật để trỏ đến node kế tiếp.

Cuối cùng, hàm sẽ giải phóng bộ nhớ đã được cấp phát cho sản phẩm được xóa và thông báo cho người dùng biết rằng sản phẩm đã được xóa thành công.

2.2.6. Sắp xếp sản phẩm

Hàm sắp xếp sản phẩm như sau:

```
void displaySortedProduct(string sortBy)
{
    if (head == nullptr)
    {
        cout << "  The information system is empty." << endl;
        return;
    }
    int listSize = 0;
    Product *temp = head;
    while (temp != nullptr)
    {
        listSize++;
        temp = temp->next;
    }
    Product *productArray = new Product[listSize];
    temp = head;
    for (int i = 0; i < listSize; i++)
    {
        productArray[i] = *temp;
        temp = temp->next;
    }
    for (int i = 1; i < listSize; i++)
    {
        Product key = productArray[i];
        int j = i - 1;
        while (j >= 0 && ((sortBy == "productID" && productArray[j].productID > key.productID) ||
                          (sortBy == "name" && productArray[j].name > key.name) ||
                          (sortBy == "type" && productArray[j].type > key.type)))
        {
            productArray[j + 1] = productArray[j];
            j--;
        }
        productArray[j + 1] = key;
    }

    cout << "-----------------------------------" << endl;
    cout << "  List of products sorted by " << sortBy << ":" << endl;
    cout << "O==============O==========================================O==================O==============O==============O==============O" << endl;
    cout << "|   ProductID   |                Product Name              |       Type       |     Price    |   Quantity   |   Discount   |" << endl;
    cout << "O==============O==========================================O==================O==============O==============O==============O" << endl;
    for (int i = 0; i < listSize; i++)
    {
        cout << "|" << setw(15) << left << productArray[i].productID
             << "|" << setw(60) << left << productArray[i].name
             << "|" << setw(20) << left << productArray[i].type
             << "|" << setw(15) << left << productArray[i].price
             << "|" << setw(15) << left << productArray[i].quantity
             << "|" << setw(15) << left << productArray[i].discount << "|" << endl;
        cout << "|---------------|------------------------------------------|------------------|--------------|--------------|--------------|" << endl;
    }
    delete[] productArray;
}
```

Hình 2.6. Hàm sắp xếp sản phẩm

Các hàm này đều liên quan đến việc sắp xếp danh sách sản phẩm theo một trường dữ liệu nào đó.

Hàm displaySortedProduct được sử dụng để hiển thị danh sách sản phẩm được sắp xếp theo một trường cho trước (tên, loại hoặc mã sản phẩm) và được lưu trữ dưới dạng một mảng động.

Các sản phẩm được lưu trữ trong danh sách liên kết được duyệt và sao chép vào mảng động, sau đó sắp xếp mảng theo thứ tự tăng dần của trường được chỉ định bởi biến "sortBy".

Cuối cùng, danh sách sản phẩm được hiển thị bằng cách sử dụng các thông tin sản phẩm được lưu trữ trong mảng, bao gồm mã sản phẩm, tên, loại, giá, số lượng và giảm giá

2.3. Quản lý hoá đơn

2.3.1. Tạo hóa đơn

Hàm tạo hóa đơn như sau:

```
system("cls");
displayProduct();
string billID;
cout << "\tEnter date: ";
string date;
bool valid_date = false;
do {
    cin >> date;
    valid_date = editdate(date);
} while (!valid_date);
cin.ignore();
cout << "\tEnter Bill ID: ";
editbillID(billID);
do {
    serial++;
    string productID = "productID";
    string ProductID;
    cout << "\tEnter product ID: ";
    cin >> ProductID;
    int quantitybuy;
    cout << "\tHow many do you want to buy: ";
    cin >> quantitybuy;
    vector<Product*> BuyProduct = searchProduct(ProductID, productID);
    if (BuyProduct.empty())
    {
        cout << "\tProduct not found." << endl;
        system("cls");
        return;
    }
```

```
if (BuyProduct[0]->quantity == 0)
{
    cout << "\tWe currently don't have this product'." << endl;
    return;
}
int total;
int remain;
float dis;
remain = BuyProduct[0]->quantity - quantitybuy;
if(BuyProduct[0]->discount == 0)
{
    total = quantitybuy * BuyProduct[0]->price;
}
else {
    dis = BuyProduct[0]->discount / 100;
    total = quantitybuy * (BuyProduct[0]->price - BuyProduct[0]->price * dis);
    }
if (remain < 0)
{
    cout << "\tSorry we don't have enough" << endl;
    remain = BuyProduct[0]->quantity;
    cout << "\tChange a mount of quantity product: ";
    cin >> quantitybuy;
}
BuyProduct[0]->quantity=remain;
exportData("Product.txt");
addCustomer(billID, date, serial, BuyProduct[0]->name, quantitybuy, total);
exportDatabill("bill.txt");
cout << "\tPress 'Y' or 'y' to countinue to buy another product ?: " << endl;
cin >> YN;
} while(YN=="y" || YN=="Y");
serial = 0;
cin.ignore();
break;
```

Hình 2.7. Hàm tạo hóa đơn

Hàm tạo hóa đơn trong chương trình quản lí hoá đơn là chức năng cho phép nhân viên lựa chọn sản phẩm mua, cập nhật số lượng sản phẩm sau khi mua, tính tổng giá trị đơn hàng, thêm thông tin hoá đơn và xuất ra file dữ liệu hoá đơn.

Đầu tiên, hàm hiển thị danh sách sản phẩm thông qua hàm displayProduct() và yêu cầu người dùng nhập ngày mua và mã hoá đơn. Tiếp theo, trong vòng lặp do-while, người dùng được yêu cầu nhập mã sản phẩm và số lượng sản phẩm muốn mua. Sau đó, hàm searchProduct() được sử dụng để tìm kiếm thông tin sản phẩm theo mã sản phẩm và danh mục sản phẩm. Nếu sản phẩm không tồn tại hoặc số lượng sản phẩm không đủ để bán, thông báo lỗi sẽ xuất hiện và vòng lặp sẽ kết thúc.

Nếu sản phẩm có đủ số lượng để bán, hàm tính tổng giá trị đơn hàng và cập nhật số lượng sản phẩm còn lại sau khi khách hàng đã mua. Tiếp theo, thông tin hoá đơn được thêm vào bảng bill thông qua hàm addCustomer() và xuất ra file dữ liệu hoá đơn. Sau đó, người dùng được yêu cầu xác nhận có muốn tiếp tục mua sản phẩm khác hay

không. Nếu người dùng chọn tiếp tục, vòng lặp sẽ tiếp tục và yêu cầu nhập lại mã sản phẩm và số lượng sản phẩm. Nếu người dùng chọn dừng lại, hàm kết thúc và quay về menu chính.

2.3.2. Hiển thị danh sách hóa đơn

```
void DisplayBill() {
    Customer *temp = headC;
    cout << "O-------------o-------------o----------o-------------------------------------------------o--------------o--------------O" << endl;
    cout << "|      Date    |    BillID    |  Serial  |                   Product Name                  |   Quantity   |     Total    |" << endl;
    cout << "o-------------o-------------o----------o-------------------------------------------------o--------------o--------------o" << endl;
    while (temp != nullptr) {
        cout << "|"  << setw(14) << left << temp->date
             << "|" << setw(14) << left << temp->billID
             << "|" << setw(11)  << left << temp->serial
             << "|" << setw(60) << left << temp->productname
             << "|" << setw(15) << left << temp->productquantity
             << "|" << setw(15) << left << temp->total << "|" << endl;
        cout << "o-------------o-------------o----------o-------------------------------------------------o--------------o--------------" << endl;
        temp = temp->next;
    }
}
```

```
void returnToPreviousFunction() {
    int input;
    do {
        std::cout << "  Enter 0 to return to the previous function: ";
        std::cin >> input;
    } while (input != 0);
    system("cls");
    cin.ignore();
}
```

```
case 1:
{
    system("cls");
    DisplayBill();
    returnToPreviousFunction();
    break;
}
```

Hình 2.8. Hàm xem tất cả hóa đơn

Hàm DisplayCustomer trong chương trình quản lí hoá đơn được sử dụng để hiển thị danh sách hoá đơn của khách hàng.

Trước tiên, con trỏ temp được khởi tạo và trỏ tới headC (đầu danh sách hoá đơn).

Sau đó, trong vòng lặp while, chương trình sẽ duyệt qua từng phần tử của danh sách hoá đơn và hiển thị thông tin của từng hoá đơn. Thông tin được hiển thị gồm ngày (Date), mã hoá đơn (Bill ID), số thứ tự (Serial), tên sản phẩm (Productname), số lượng sản phẩm (Productquantity), và tổng cộng (Total).

Mỗi dòng thông tin hoá đơn được định dạng bằng cách sử dụng hàm setw để xác định độ rộng của các cột và left để căn lề trái. Sau mỗi hoá đơn, một dòng ngăn cách được hiển thị để tạo thành bảng danh sách.

Cuối cùng, khi hết danh sách hoá đơn, vòng lặp kết thúc và hàm kết thúc.

2.3.3. Tìm hoá đơn

Hàm tìm hoá đơn như sau:

```
vector<Customer *> searchBill(string keyword, string field)
{
    vector<Customer *> show;
    Customer *temp = headC;
    while (temp != nullptr)
    {
        if (field == "date")
        {
            temp->date.erase(0, temp->date.find_first_not_of(' '));
            temp->date.erase(temp->date.find_last_not_of(' ') + 1);
            if (temp->date == removeSpaces(keyword))
            {
                show.push_back(temp);
            }
        }
        else if (field == "month")
        {
            size_t pos = temp->date.find_first_of("/");
            size_t pos2 = temp->date.find_last_not_of(' ') + 1;
            string month = temp->date.substr(pos, pos2-pos);
            if(month == "/"+removeSpaces(keyword))
            {
                show.push_back(temp);
            }
        }
        else if (field == "year")
        {
            size_t pos = temp->date.find_first_of("/");
            size_t pos2 = temp->date.find_first_of("/", pos+1);
            size_t pos3 = temp->date.find_last_not_of(' ') + 1;
            string year = temp->date.substr(pos2, pos3-pos2);
            if(year == "/"+removeSpaces(keyword))
            {
                show.push_back(temp);
            }
        }
        else if (field == "billID")
        {
            temp->billID.erase(0, temp->billID.find_first_not_of(' '));
            temp->billID.erase(temp->billID.find_last_not_of(' ') + 1);
            if (temp->billID == removeSpaces(keyword))
            {
                show.push_back(temp);
            }
        }
        temp = temp->next;
    }
    return show;
}
```

```
void findBill(string keyword, string field)
{
    vector<Customer *> billsToFind = searchBill(keyword, field);
    if (billsToFind.empty())
    {
        cout << "  Not found in the system." << endl;
        return;
    }
    else{
    cout << "O--------------o--------------o----------o-------------------------------------------------------------o--------------o--------------O" << endl;
    cout << "|      Date    |    BillID    |   Serial  |                        Product Name                         |   Quantity   |      Total    |" << endl;
    cout << "o--------------o--------------o----------o-------------------------------------------------------------o--------------o--------------o" << endl;
    for (Customer *customer : billsToFind)
    {
        cout << "|" << setw(14) << left << customer->date
             << "|" << setw(14) << left << customer->billID
             << "|" << setw(11)  << left << customer->serial
             << "|" << setw(60) << left << customer->productname
             << "|" << setw(15) << left << customer->productquantity
             << "|" << setw(15) << left << customer->total << "|" << endl;
        cout << "o--------------o--------------o----------o-------------------------------------------------------------o--------------o--------------o" << endl;
    }
    }
}
```

```
case 3:
    {
        system("cls");
        string billID;
        cout << "  Enter the Bill ID to find: ";
        editnumber(billID);
        findBill(billID,"billID");
        returnToPreviousFunction();
        break;
    }
```

Hình 2.9. Hàm tìm hoá đơn

Đầu tiên, tạo ra biến ID để nhập mã hoá đơn.

Sau đó, hàm searchBill được gọi để tìm hoá đơn có mã tương ứng. Nếu không tìm thấy hoá đơn, thông báo "Not found in the system" sẽ được hiển thị và chương trình kết thúc.

Nếu tìm thấy hoá đơn, hàm findBill được gọi để hiển thị thông tin hoá đơn.

2.3.4. Xem doanh thu

Hàm xem doanh thu như sau:

```
case 1:
{
    system("cls");
    string date;
    cout << "  Type date you want to see revenue (dd/mm/yyyy): ";
    bool valid_date = false;
    do {
    cin >> date;
    valid_date = editdate(date);
    } while (!valid_date);
    cin.ignore();
    findBill(date,"date");
    ifstream revenue("bill.txt");
    string line;
    while (getline(revenue, line)) {
    size_t pos = line.find_first_of(";");
    size_t pos2 = line.find_first_of(";",pos+1);
    string date2 = line.substr(pos, pos2 - pos);
    removeSpaces(date2);
    if(date2 == "; "+date)
    {
        size_t pos1 = line.find_last_of(';');
        string last_value_str = line.substr(pos1 + 1);
        int total = stoi(last_value_str);
        sum = sum + total;
    }
    else
    {
        sum = sum + 0;
    }
}
    cout << "                    Revenue: " << sum << endl;
    sum = 0;
    returnToPreviousFunction();
    break;
}
```

```
case 2:
{
    system("cls");
    string month;
    cout << "  Type month you want to see revenue (mm/yyyy): ";
    bool valid_month = false;
    do {
    cin >> month;
    valid_month = editmonth(month);
    } while (!valid_month);
    cin.ignore();
    findBill(month,"month");
    ifstream revenue("bill.txt");
    string line;
    while (getline(revenue, line)) {
    size_t pos = line.find_first_of(";");
    size_t pos2 = line.find_first_of(";",pos+1);
    size_t pos3 = line.find_first_of("/");
    string month2 = line.substr(pos3, pos2 - pos3);
    removeSpaces(month2);
    if(month2 == "/"+month)
    {
        size_t pos1 = line.find_last_of(';');
        string last_value_str = line.substr(pos1 + 1);
        int total = stoi(last_value_str);
        sum = sum + total;
    }
    else
    {
        sum = sum + 0;
    }
}
    cout << "                    Revenue: " << sum << endl;
    sum = 0;
    returnToPreviousFunction();
    break;
}
```

```
case 3:
{
    system("cls");
    string year;
    cout << "  Type year you want to see revenue (yyyy): ";
    bool valid_year = false;
    do {
    cin >> year;
    valid_year = edityear(year);
    } while (!valid_year);
    cin.ignore();
    findBill(year,"year");
    ifstream revenue("bill.txt");
    string line;
    while (getline(revenue, line)) {
    size_t pos = line.find_first_of(";");
    size_t pos2 = line.find_first_of(";",pos+1);
    size_t pos3 = line.find_first_of("/");
    size_t pos4 = line.find_first_of("/",pos3+1);
    string year2 = line.substr(pos4, pos2 - pos4);
    removeSpaces(year2);
    if(year2 == "/"+year)
    {
        size_t pos1 = line.find_last_of(';');
        string last_value_str = line.substr(pos1 + 1);
        int total = stoi(last_value_str);
        sum = sum + total;
    }
    else
    {
        sum = sum + 0;
    }
}
    cout << "                    Revenue: " << sum << endl;
    sum = 0;
    returnToPreviousFunction();
    break;
}
```

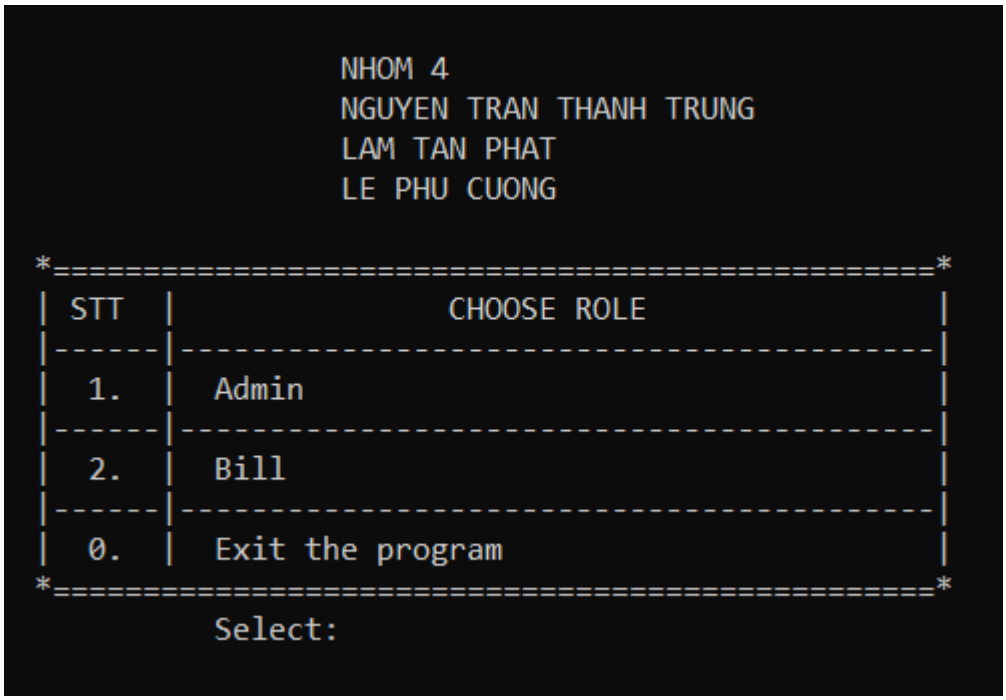
Hình 2.10. Hàm xem doanh thu

Người dùng sẽ được yêu cầu nhập ngày muốn xem doanh thu trong định dạng "dd/mm/yyyy" bằng câu lệnh "cin". Ngày này sẽ được kiểm tra tính hợp lệ bằng cách gọi hàm "editdate(date)".

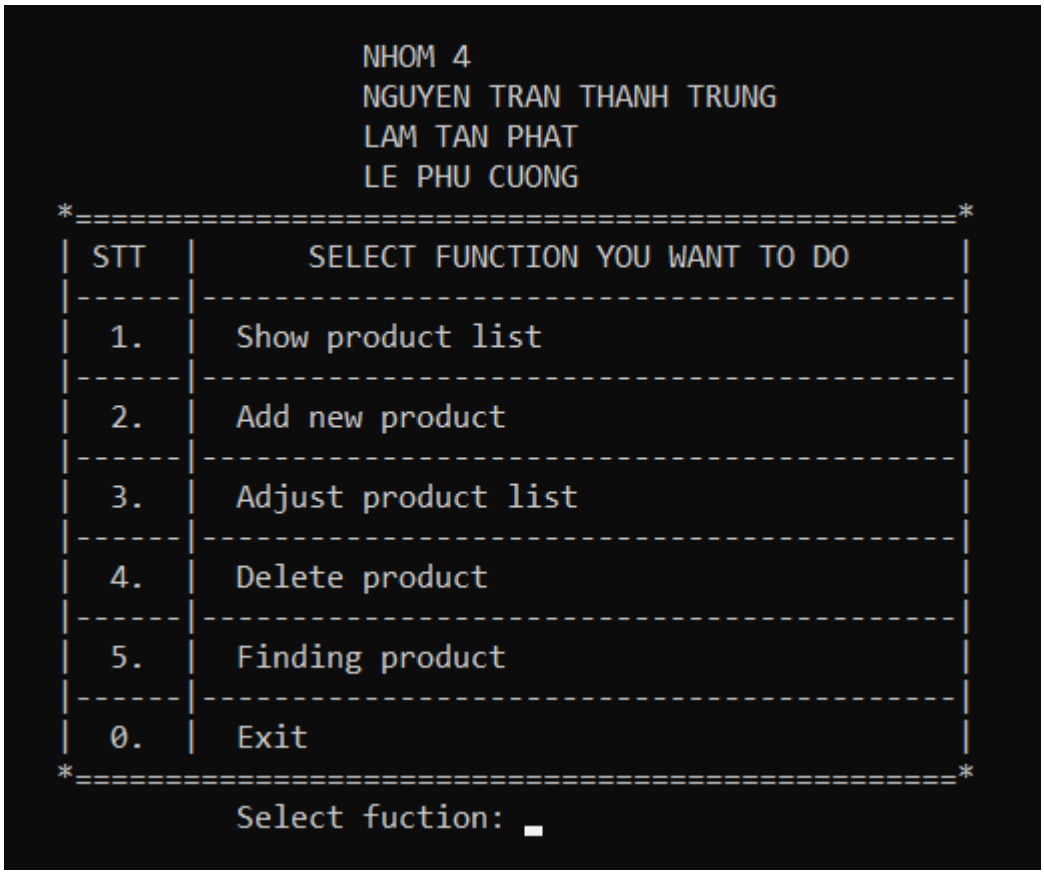
Sau khi xác định ngày hợp lệ, hàm "findBill(date, "date")" sẽ được gọi để tìm kiếm các hóa đơn trong tệp văn bản "bill.txt" chứa ngày này. Các hóa đơn được tìm thấy sẽ được đọc và tính toán tổng số tiền bán hàng bằng cách lấy giá trị cuối cùng trong mỗi dòng của tệp văn bản. Tổng số tiền này sẽ được cộng vào biến "sum".

Cuối cùng, tổng doanh thu của ngày được hiển thị trên màn hình console và biến "sum" được đặt lại thành 0. Sau đó, chương trình sẽ trở lại chức năng trước đó bằng cách gọi hàm "returnToPreviousFunction()".

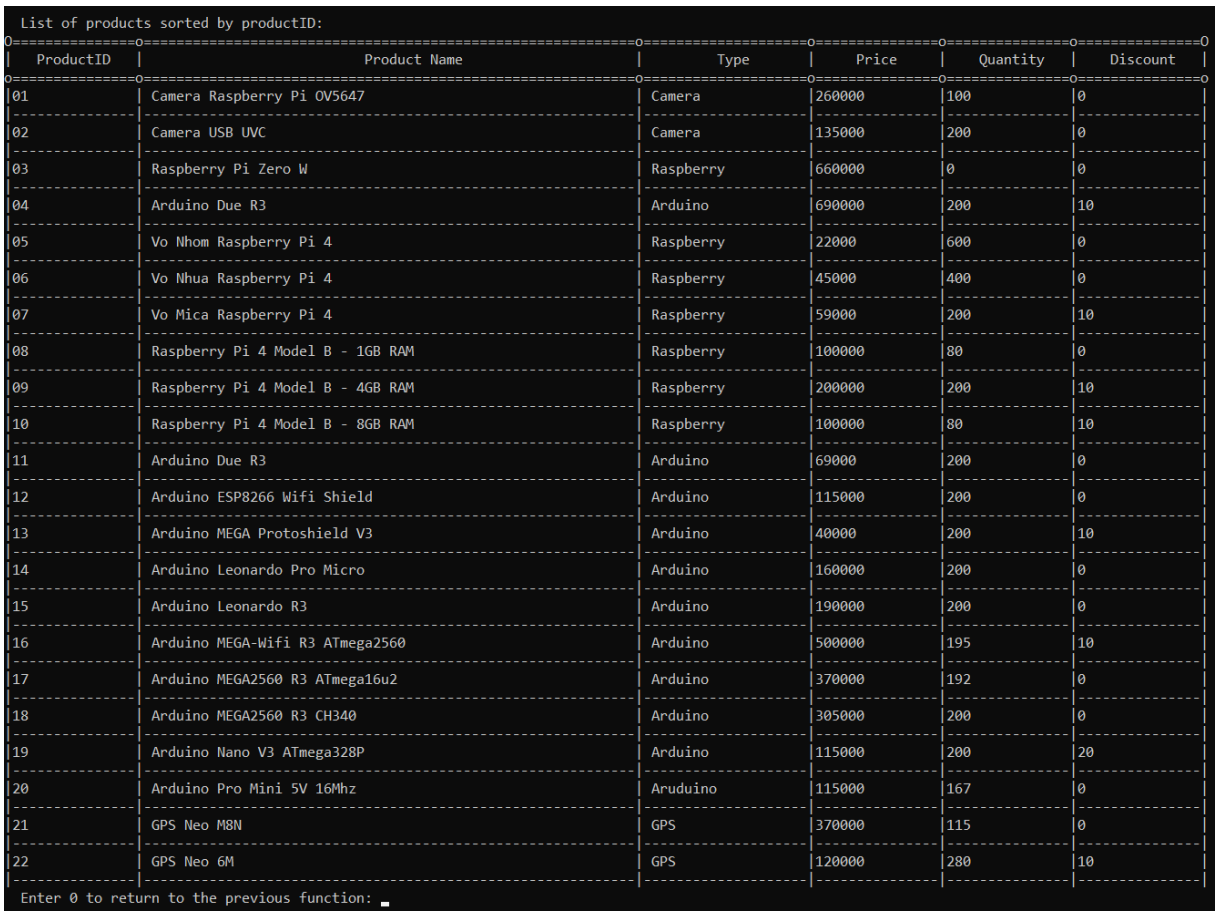
Tương tự với xem doanh thu của tháng và của năm.



Hình 3.1. Giao diện trang chủ



Hình 3.2. Giao diện quản lý linh kiện

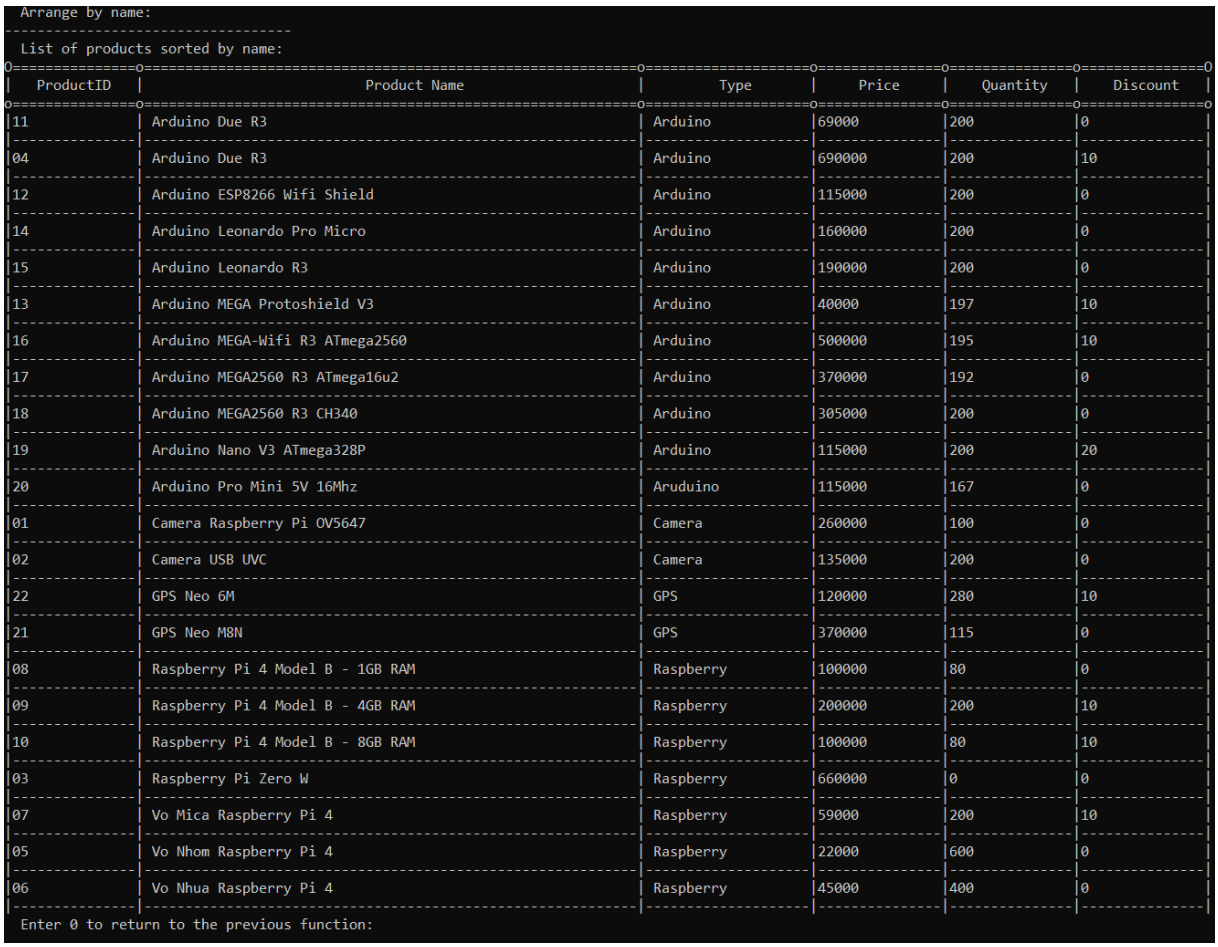


List of products sorted by productID:

ProductID	Product Name	Type	Price	Quantity	Discount
01	Camera Raspberry Pi OV5647	Camera	260000	100	0
02	Camera USB UVC	Camera	135000	200	0
03	Raspberry Pi Zero W	Raspberry	660000	0	0
04	Arduino Due R3	Arduino	690000	200	10
05	Vo Nhom Raspberry Pi 4	Raspberry	22000	600	0
06	Vo Nhua Raspberry Pi 4	Raspberry	45000	400	0
07	Vo Mica Raspberry Pi 4	Raspberry	59000	200	10
08	Raspberry Pi 4 Model B - 1GB RAM	Raspberry	100000	80	0
09	Raspberry Pi 4 Model B - 4GB RAM	Raspberry	200000	200	10
10	Raspberry Pi 4 Model B - 8GB RAM	Raspberry	100000	80	10
11	Arduino Due R3	Arduino	69000	200	0
12	Arduino ESP8266 Wifi Shield	Arduino	115000	200	0
13	Arduino MEGA Protoshield V3	Arduino	40000	200	10
14	Arduino Leonardo Pro Micro	Arduino	160000	200	0
15	Arduino Leonardo R3	Arduino	190000	200	0
16	Arduino MEGA-Wifi R3 ATmega2560	Arduino	500000	195	10
17	Arduino MEGA2560 R3 ATmega16u2	Arduino	370000	192	0
18	Arduino MEGA2560 R3 CH340	Arduino	305000	200	0
19	Arduino Nano V3 ATmega328P	Arduino	115000	200	20
20	Arduino Pro Mini 5V 16Mhz	Aruduino	115000	167	0
21	GPS Neo M8N	GPS	370000	115	0
22	GPS Neo 6M	GPS	120000	280	10

Enter 0 to return to the previous function:

Hình 3.3. Giao diện hiển thị danh sách sản phẩm theo ID

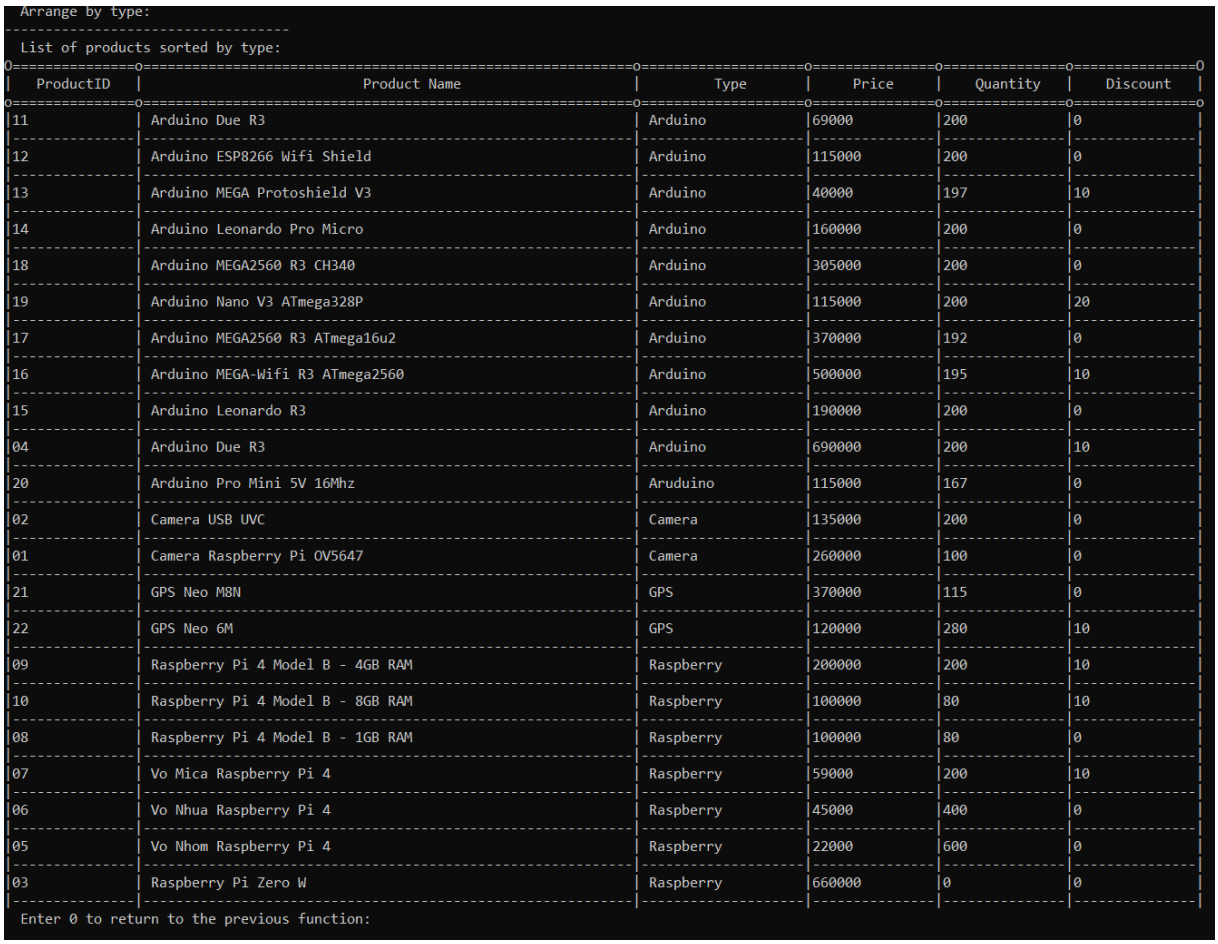


```
 Arrange by name:
----------------------------------
 List of products sorted by name:
```

ProductID	Product Name	Type	Price	Quantity	Discount
11	Arduino Due R3	Arduino	69000	200	0
04	Arduino Due R3	Arduino	690000	200	10
12	Arduino ESP8266 Wifi Shield	Arduino	115000	200	0
14	Arduino Leonardo Pro Micro	Arduino	160000	200	0
15	Arduino Leonardo R3	Arduino	190000	200	0
13	Arduino MEGA Protoshield V3	Arduino	40000	197	10
16	Arduino MEGA-Wifi R3 ATmega2560	Arduino	500000	195	10
17	Arduino MEGA2560 R3 ATmega16u2	Arduino	370000	192	0
18	Arduino MEGA2560 R3 CH340	Arduino	305000	200	0
19	Arduino Nano V3 ATmega328P	Arduino	115000	200	20
20	Arduino Pro Mini 5V 16Mhz	Aruduino	115000	167	0
01	Camera Raspberry Pi OV5647	Camera	260000	100	0
02	Camera USB UVC	Camera	135000	200	0
22	GPS Neo 6M	GPS	120000	280	10
21	GPS Neo M8N	GPS	370000	115	0
08	Raspberry Pi 4 Model B - 1GB RAM	Raspberry	100000	80	0
09	Raspberry Pi 4 Model B - 4GB RAM	Raspberry	200000	200	10
10	Raspberry Pi 4 Model B - 8GB RAM	Raspberry	100000	80	10
03	Raspberry Pi Zero W	Raspberry	660000	0	0
07	Vo Mica Raspberry Pi 4	Raspberry	59000	200	10
05	Vo Nhom Raspberry Pi 4	Raspberry	22000	600	0
06	Vo Nhua Raspberry Pi 4	Raspberry	45000	400	0

```
 Enter 0 to return to the previous function:
```

Hình 3.4. Giao diện hiển thị danh sách sản phẩm theo tên (A->Z)



```
 Arrange by type:
---------------------------------
 List of products sorted by type:
o==============o============================================================o==================o==============o==============o==============o
|   ProductID  |                        Product Name                        |        Type      |     Price    |    Quantity  |    Discount  |
o==============o============================================================o==================o==============o==============o==============o
|11            | Arduino Due R3                                             | Arduino          |69000         |200           |0             |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|12            | Arduino ESP8266 Wifi Shield                                | Arduino          |115000        |200           |0             |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|13            | Arduino MEGA Protoshield V3                                | Arduino          |40000         |197           |10            |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|14            | Arduino Leonardo Pro Micro                                 | Arduino          |160000        |200           |0             |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|18            | Arduino MEGA2560 R3 CH340                                  | Arduino          |305000        |200           |0             |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|19            | Arduino Nano V3 ATmega328P                                 | Arduino          |115000        |200           |20            |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|17            | Arduino MEGA2560 R3 ATmega16u2                             | Arduino          |370000        |192           |0             |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|16            | Arduino MEGA-Wifi R3 ATmega2560                            | Arduino          |500000        |195           |10            |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|15            | Arduino Leonardo R3                                        | Arduino          |190000        |200           |0             |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|04            | Arduino Due R3                                             | Arduino          |690000        |200           |10            |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|20            | Arduino Pro Mini 5V 16Mhz                                  | Aruduino         |115000        |167           |0             |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|02            | Camera USB UVC                                             | Camera           |135000        |200           |0             |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|01            | Camera Raspberry Pi OV5647                                 | Camera           |260000        |100           |0             |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|21            | GPS Neo M8N                                                | GPS              |370000        |115           |0             |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|22            | GPS Neo 6M                                                 | GPS              |120000        |280           |10            |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|09            | Raspberry Pi 4 Model B - 4GB RAM                           | Raspberry        |200000        |200           |10            |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|10            | Raspberry Pi 4 Model B - 8GB RAM                           | Raspberry        |100000        |80            |10            |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|08            | Raspberry Pi 4 Model B - 1GB RAM                           | Raspberry        |100000        |80            |0             |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|07            | Vo Mica Raspberry Pi 4                                     | Raspberry        |59000         |200           |10            |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|06            | Vo Nhua Raspberry Pi 4                                     | Raspberry        |45000         |400           |0             |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|05            | Vo Nhom Raspberry Pi 4                                     | Raspberry        |22000         |600           |0             |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
|03            | Raspberry Pi Zero W                                        | Raspberry        |660000        |0             |0             |
|--------------|------------------------------------------------------------|------------------|--------------|--------------|--------------|
 Enter 0 to return to the previous function:
```

Hình 3.5. Giao diện hiển thị danh sách sản phẩm theo loại (A->Z)

```
 Arrange by price:
---------------------------------
 List of products sorted by price:
```

ProductID	Product Name	Type	Price	Quantity	Discount
05	Vo Nhom Raspberry Pi 4	Raspberry	22000	600	0
13	Arduino MEGA Protoshield V3	Arduino	40000	197	10
06	Vo Nhua Raspberry Pi 4	Raspberry	45000	400	0
07	Vo Mica Raspberry Pi 4	Raspberry	59000	200	10
11	Arduino Due R3	Arduino	69000	200	0
10	Raspberry Pi 4 Model B - 8GB RAM	Raspberry	100000	80	10
08	Raspberry Pi 4 Model B - 1GB RAM	Raspberry	100000	80	0
12	Arduino ESP8266 Wifi Shield	Arduino	115000	200	0
19	Arduino Nano V3 ATmega328P	Arduino	115000	200	20
20	Arduino Pro Mini 5V 16Mhz	Aruduino	115000	167	0
22	GPS Neo 6M	GPS	120000	280	10
02	Camera USB UVC	Camera	135000	200	0
14	Arduino Leonardo Pro Micro	Arduino	160000	200	0
15	Arduino Leonardo R3	Arduino	190000	200	0
09	Raspberry Pi 4 Model B - 4GB RAM	Raspberry	200000	200	10
01	Camera Raspberry Pi OV5647	Camera	260000	100	0
18	Arduino MEGA2560 R3 CH340	Arduino	305000	200	0
21	GPS Neo M8N	GPS	370000	115	0
17	Arduino MEGA2560 R3 ATmega16u2	Arduino	370000	192	0
16	Arduino MEGA-Wifi R3 ATmega2560	Arduino	500000	195	10
03	Raspberry Pi Zero W	Raspberry	660000	0	0
04	Arduino Due R3	Arduino	690000	200	10

```
 Enter 0 to return to the previous function:
```

Hình 3.6. Giao diện hiển thị danh sách sản phẩm theo giá (nhỏ -> lớn

```
 Enter product ID: 22
 Enter product name: GPS NEO 7M
 Enter product type: GPS
 Enter protudct price: 58000
 Enter quantity: 228
 Enter discount: 0
 Please check your enter again
```

ProductID	Name	Type	Price	Quantity	Discount
22	GPS NEO 7M	GPS	58000	228	0

```
 If it true, enter 1; false, enter 0: 1
 Enter 0 to return to the previous function:
```

Hình 3.7. Giao diện thêm sản phẩm

```
 NHOM 4
+=================================================+
| STT  |       SELECT ADJUST FUNCTION             |
|------|------------------------------------------|
|  1.  |  Adjust by ID                            |
|------|------------------------------------------|
|  2.  |  Adjust by product name                  |
|------|------------------------------------------|
|  0.  |  Exit                                    |
+=================================================+
          Select function:
```

Hình 3.8. Giao diện lựa chọn chỉnh sửa sản phẩm

```
 Find by ID: 22
+---------------+---------------------------------+--------------+--------------+--------------+---------------+
|   ProductID   |              Name               |     Type     |     Price    |    Quantity  |    Discount   |
+---------------+---------------------------------+--------------+--------------+--------------+---------------+
| 22            | GPS NEO 7M                      | GPS          | 58000        | 228          | 0             |
+---------------+---------------------------------+--------------+--------------+--------------+---------------+
 Enter new ID: 23
 Enter new product name: GPS NEO N7M
 Enter new type: GPS
 Enter new price: 45000
 Enter new quantity: 220
 Enter new discount: 10
 Please check your enter again
+---------------+---------------------------------+--------------+--------------+--------------+---------------+
|   ProductID   |              Name               |     Type     |     Price    |    Quantity  |    Discount   |
+---------------+---------------------------------+--------------+--------------+--------------+---------------+
| 23            | GPS NEO N7M                     | GPS          | 45000        | 220          | 10            |
+---------------+---------------------------------+--------------+--------------+--------------+---------------+
 If it true, enter 1; false, enter 0: 1
 Product information updated successfully.
 Enter 0 to return to the previous function: 0
```

Hình 3.9. Giao diện chỉnh sửa thông tin sản phẩm

```
Enter the productID to be deleted: 23
Product deleted successfully.
Enter 0 to return to the previous function:
```

Hình 3.10. Giao diện xoá sản phẩm

```
                NHOM 4
                NGUYEN TRAN THANH TRUNG
                LAM TAN PHAT
                LE PHU CUONG
*================================================*
| STT  |          SELECT SEARCH FUNCTION         |
|------|-----------------------------------------|
|  1.  |  Find by ID                             |
|------|-----------------------------------------|
|  2.  |  Find by name                           |
|------|-----------------------------------------|
|  3.  |  Find by type                           |
|------|-----------------------------------------|
|  0.  |  Exit                                   |
*================================================*
              Select function:
```

Hình 3.11. Giao diện lựa chọn tìm kiếm sản phẩm

```
 Find by ID: 21
+---------------+---------------------------------+--------------+--------------+--------------+---------------+
|   ProductID   |              Name               |     Type     |     Price    |    Quantity  |    Discount   |
+---------------+---------------------------------+--------------+--------------+--------------+---------------+
| 21            | GPS NEO M8N                     | GPS          | 370000       | 170          | 0             |
+---------------+---------------------------------+--------------+--------------+--------------+---------------+
 Enter 0 to return to the previous function:
```

Hình 3.12. Giao diện tìm kiếm sản phẩm bằng ID

```
Find by name: Camera Raspberry Pi OV5647 5MP
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
|  ProductID  |              Name              |     Type     |     Price    |   Quantity   |   Discount   |
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
| 1           | Camera Raspberry Pi OV5647 5MP | RASPBERRY PI | 260000       | 100          | 0            |
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
Enter 0 to return to the previous function:
```

Hình 3.13. Giao diện tìm kiếm sản phẩm bằng tên

```
Find by type:ARDUINO
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
|  ProductID  |              Name              |     Type     |     Price    |   Quantity   |   Discount   |
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
| 4           | Arduino Due R3                 | ARDUINO      | 690000       | 200          | 10           |
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
| 11          | Arduino Due R3                 | ARDUINO      | 690000       | 200          | 10           |
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
| 12          | Arduino ESP8266 Wifi Shield    | ARDUINO      | 115000       | 200          | 10           |
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
| 13          | Arduino Leonardo pro micro     | ARDUINO      | 160000       | 200          | 10           |
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
| 14          | Arduino Leonardo R3            | ARDUINO      | 190000       | 200          | 10           |
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
| 15          | Arduino MEGA Protoshield V3    | ARDUINO      | 40000        | 200          | 10           |
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
| 16          | Arduino mega+WiFi R3 ATmega2560| ARDUINO      | 500000       | 200          | 10           |
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
| 17          | Arduino MEGA2560 R3 Atmega16u2 | ARDUINO      | 370000       | 200          | 10           |
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
| 18          | Arduino MEGA2560 R3 CH340      | ARDUINO      | 305000       | 200          | 10           |
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
| 19          | Arduino nano V3.0 ATmega328P   | ARDUINO      | 115000       | 200          | 10           |
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
| 20          | Arduino pro mini 5V 16Mhz      | ARDUINO      | 115000       | 180          | 10           |
+-------------+--------------------------------+--------------+--------------+--------------+--------------+
Enter 0 to return to the previous function:
```

Hình 3.14. Giao diện tìm kiếm sản phẩm bằng loại

```
                NHOM 4
                NGUYEN TRAN THANH TRUNG
                LAM TAN PHAT
                LE PHU CUONG
*================================================*
|  STT  |      SELECT FUNCTION YOU WANT TO DO      |
|-------|-------------------------------------------|
|   1.  |  Show all bill                            |
|-------|-------------------------------------------|
|   2.  |  Add new bill                             |
|-------|-------------------------------------------|
|   3.  |  Find bill ID                             |
|-------|-------------------------------------------|
|   4.  |  Revenue                                  |
|-------|-------------------------------------------|
|   0.  |  Exit                                     |
*================================================*
          Select function:
```

Hình 3.15. Giao diện lựa chọn quản lý hoá đơn

Date	BillID	Serial	Product Name	Quantity	Total
10/02/2023	6	2	Arduino MEGA2560 R3 ATmega16u2	8	2960000
10/02/2023	6	1	GPS Neo 6M	10	1080000
12/12/2020	5	2	Arduino Pro Mini 5V 16Mhz	3	345000
12/12/2020	5	1	GPS Neo M8N	5	1850000
12/10/2023	4	1	Arduino Pro Mini 5V 16Mhz	10	1150000
17/10/2022	3	1	Arduino MEGA2560 R3 ATmega16u2	8	2960000
15/10/2020	2	2	GPS Neo 6M	10	1080000
15/10/2020	2	1	Arduino MEGA-Wifi R3 ATmega2560	5	2250000
12/02/2023	1	1	Raspberry Pi Zero W	100	66000000

Enter 0 to return to the previous function:

Hình 3.16. Giao diện xem toàn bộ hóa đơn

ProductID	Product Name	Type	Price	Quantity	Discount
09	Raspberry Pi 4 Model B - 4GB RAM	Raspberry	200000	200	10
10	Raspberry Pi 4 Model B - 8GB RAM	Raspberry	100000	80	10
11	Arduino Due R3	Arduino	69000	200	0
12	Arduino ESP8266 Wifi Shield	Arduino	115000	200	0
13	Arduino MEGA Protoshield V3	Arduino	40000	200	10
14	Arduino Leonardo Pro Micro	Arduino	160000	200	0
18	Arduino MEGA2560 R3 CH340	Arduino	305000	200	0
19	Arduino Nano V3 ATmega328P	Arduino	115000	200	20
20	Arduino Pro Mini 5V 16Mhz	Aruduino	115000	167	0
21	GPS Neo M8N	GPS	370000	115	0
22	GPS Neo 6M	GPS	120000	280	10
17	Arduino MEGA2560 R3 ATmega16u2	Arduino	370000	192	0
16	Arduino MEGA-Wifi R3 ATmega2560	Arduino	500000	195	10
15	Arduino Leonardo R3	Arduino	190000	200	0
08	Raspberry Pi 4 Model B - 1GB RAM	Raspberry	100000	80	0
07	Vo Mica Raspberry Pi 4	Raspberry	59000	200	10
06	Vo Nhua Raspberry Pi 4	Raspberry	45000	400	0
05	Vo Nhom Raspberry Pi 4	Raspberry	22000	600	0
04	Arduino Due R3	Arduino	690000	200	10
03	Raspberry Pi Zero W	Raspberry	660000	0	0
02	Camera USB UVC	Camera	135000	200	0
01	Camera Raspberry Pi OV5647	Camera	260000	100	0

Enter date: 12/10/2022
Enter Bill ID: 7
Enter product ID: 13
How many do you want to buy: 3
Press 'Y' or 'y' to countinue to buy another product ?:

Hình 3.17. Giao diện tạo thêm hóa đơn mới

Enter the Bill ID to find: 7

Date	BillID	Serial	Product Name	Quantity	Total
12/10/2022	7	1	Arduino MEGA Protoshield V3	3	108000

Enter 0 to return to the previous function:

Hình 3.18. Giao diện tìm hoá đơn theo ID

```
 Type date you want to see revenue (dd/mm/yyyy): 12/12/2020
O-------------o-------------o----------o-------------------------------------------o--------------o--------------O
|      Date   |    BillID   |   Serial |                Product Name               |   Quantity   |      Total   |
o-------------o-------------o----------o-------------------------------------------o--------------o--------------o
|12/12/2020   |5            |2         | Arduino Pro Mini 5V 16Mhz                 |3             |345000        |
o-------------o-------------o----------o-------------------------------------------o--------------o--------------o
|12/12/2020   |5            |1         | GPS Neo M8N                               |5             |1850000       |
o-------------o-------------o----------o-------------------------------------------o--------------o--------------o
                Revenue: 2195010
 Enter 0 to return to the previous function:
```

Hình 3.19. Giao diện xem doanh thu theo ngày

```
 Type month you want to see revenue (mm/yyyy): 02/2023
O-------------o-------------o----------o-------------------------------------------o--------------o--------------O
|      Date   |    BillID   |   Serial |                Product Name               |   Quantity   |      Total   |
o-------------o-------------o----------o-------------------------------------------o--------------o--------------o
|10/02/2023   |6            |2         | Arduino MEGA2560 R3 ATmega16u2            |8             |2960000       |
o-------------o-------------o----------o-------------------------------------------o--------------o--------------o
|10/02/2023   |6            |1         | GPS Neo 6M                                |10            |1080000       |
o-------------o-------------o----------o-------------------------------------------o--------------o--------------o
|12/02/2023   |1            |1         | Raspberry Pi Zero W                       |100           |66000000      |
o-------------o-------------o----------o-------------------------------------------o--------------o--------------o
                Revenue: 70040000
 Enter 0 to return to the previous function:
```

Hình 3.20. Giao diện xem doanh thu theo tháng

```
 Type year you want to see revenue (yyyy): 2023
O-------------o-------------o----------o-------------------------------------------o--------------o--------------O
|      Date   |    BillID   |   Serial |                Product Name               |   Quantity   |      Total   |
o-------------o-------------o----------o-------------------------------------------o--------------o--------------o
|10/02/2023   |6            |2         | Arduino MEGA2560 R3 ATmega16u2            |8             |2960000       |
o-------------o-------------o----------o-------------------------------------------o--------------o--------------o
|10/02/2023   |6            |1         | GPS Neo 6M                                |10            |1080000       |
o-------------o-------------o----------o-------------------------------------------o--------------o--------------o
|12/10/2023   |4            |1         | Arduino Pro Mini 5V 16Mhz                 |10            |1150000       |
o-------------o-------------o----------o-------------------------------------------o--------------o--------------o
|12/02/2023   |1            |1         | Raspberry Pi Zero W                       |100           |66000000      |
o-------------o-------------o----------o-------------------------------------------o--------------o--------------o
                Revenue: 71190000
 Enter 0 to return to the previous function:
```

Hình 3.21. Giao diện xem doanh thu theo năm

CHƯƠNG 4: KẾT LUẬN

Trong quá trình xây dựng cấu trúc dữ liệu quản lý cửa hàng bán linh kiện, đã được tạo ra một hệ thống quản lí linh hoạt và tiện lợi. Cấu trúc dữ liệu này cho phép người dùng thực hiện các hoạt động quản lí như thêm, sửa, xoá và tìm kiếm sản phẩm trong cửa hàng.

Một số cấu trúc dữ liệu quan trọng đã được sử dụng trong hệ thống bao gồm danh sách liên kết đơn để lưu trữ thông tin sản phẩm, hàm tìm kiếm để tạo ra một hàm tìm kiếm hiệu quả dựa trên các trường quan trọng như giá và số lượng, và danh sách liên kết để quản lí thông tin khách hàng và hoá đơn.

Hệ thống cũng cung cấp các chức năng quan trọng như tìm kiếm sản phẩm theo từ khóa, hiển thị thông tin sản phẩm, thêm hóa đơn mới, in hoá đơn. Đồng thời, việc nhập liệu và kiểm tra hợp lệ được thực hiện để đảm bảo tính chính xác và an toàn của dữ liệu.

Hệ thống này cho phép người dùng quản lí linh hoạt các sản phẩm và hoá đơn trong cửa hàng bán linh kiện. Nó cung cấp tính năng tìm kiếm nhanh chóng, thêm và xoá dữ liệu dễ dàng và in hoá đơn tự động. Điều này giúp tăng hiệu suất làm việc và quản lí dễ dàng, đồng thời giảm thiểu sai sót và thời gian tìm kiếm thông tin.

Tổng quan, cấu trúc dữ liệu đã xây dựng mang lại một hệ thống quản lí cửa hàng bán linh kiện hiệu quả, giúp quản lí sản phẩm, khách hàng và hoá đơn một cách thuận tiện và nhanh chóng. Nó có thể được mở rộng và tùy chỉnh để phù hợp với yêu cầu và quy mô của cửa hàng, đồng thời tăng cường hiệu suất và tối ưu hóa quá trình quản lí.