|  |
| --- |
| **TRƯỜNG ĐẠI HỌC BÁCH KHOA**  **KHOA CÔNG NGHỆ THÔNG TIN**      **ĐỒ ÁN LẬP TRÌNH CƠ SỞ**  **TÊN ĐỀ TÀI**  QUẢN LÝ BÁN HÀNG SIÊU THỊ WINMART  **Giảng viên hướng dẫn: T.S ĐẶNG THIÊN BÌNH**  **Sinh viên thực hiện**  **Nguyễn Đức Chung 20TCLC-Nhật 1 Nhóm:1**  **Đặng Nhật Minh 20TCLC-Nhật 1 Nhóm:1**  ***Đà Nẵng, 12/2021*** |

# MỤC LỤC

[MỤC LỤC 2](#_Toc90655923)

[DANH MỤC HÌNH VẼ VÀ BIỂU BẢNG 3](#_Toc90655924)

[LỜI MỞ ĐẦU 1](#_Toc90655925)

[1. GIỚI THIỆU ĐỀ TÀI 2](#_Toc90655926)

[2. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG 2](#_Toc90655927)

[2.1. Chức năng tìm sản phẩm. 2](#_Toc90655928)

[2.2. Chức năng thêm sản phẩm 3](#_Toc90655929)

[2.3. Chức năng sửa sản phẩm 3](#_Toc90655930)

[2.4. Chức năng tạo voucher 3](#_Toc90655931)

[3. THIẾT KẾ CẤU TRÚC DỮ LIỆU 5](#_Toc90655936)

[3.1. Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống. 7](#_Toc90655938)

[3.2.1. VECTOR 7](#_Toc90655939)

[3.2.2. Cây AVL 8](#_Toc90655940)

[3.2.3. PAIR 9](#_Toc90655941)

[4. PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG 13](#_Toc90655944)

[4.1. Cấu trúc hệ thống hướng đối tượng (class, object, relation) 13](#_Toc90655945)

[4.2. Kết quả 18](#_Toc90655946)

[4.2.1. Giao diện chính của chương trình 18](#_Toc90655947)

[4.2.2. Kết quả thực thi của chương trình 18](#_Toc90655948)

[4.2.3 Nhận xét đánh giá 26](#_Toc90655949)

[5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 26](#_Toc90655950)

[5.1. Kết luận 26](#_Toc90655951)

[5.2. Hướng phát triển 27](#_Toc90655952)

[TÀI LIỆU THAM KHẢO 28](#_Toc90655953)

# 

# DANH MỤC HÌNH VẼ VÀ BIỂU BẢNG

* **Bảng:**

[*Bảng 1: Các kiểu dữ liệu thực hiện tiến trình sử lý file 9*](#_Toc90652496)

[*Bảng 2: Bảng định nghĩa chế độ khi mở file 10*](#_Toc90652498)

[*Bảng 3: So sánh giữa lớp và đối tượng 13*](#_Toc90652502)

* **Hình:**

[*Hình 1: file ITEM. 4*](#_Toc90652163)

[*Hình 2: file 0.txt. 5*](#_Toc90652164)

[*Hình 3: file VOUCHER.txt. 6*](#_Toc90652165)

[*Hình 4: Linear Search. 10*](#_Toc90652166)

[*Hình 5: SkipSearch 11*](#_Toc90652167)

[*Hình 6: Giao diện ngoài chương trình 20*](#_Toc90652172)

[*Hình 7: Tìm tên sản phẩm bất kì 20*](#_Toc90652173)

[*Hình 8: Hiển thị sản phẩm có kí tự tìm kiếm 21*](#_Toc90652174)

[*Hình 9: Thêm sản phẩm 21*](#_Toc90652175)

[*Hình 10: Sửa sản phẩm 22*](#_Toc90652176)

[*Hình 11: Tạo VOUCHER 22*](#_Toc90652177)

# 

# LỜI MỞ ĐẦU

Lời đầu tiên chúng em xin gửi lời cảm ơn đến thầy giáo Đặng Tiên Bình, người đã giúp chúng em chọn đề tài, định hướng nghiên cứu, tận tình hướng dẫn và chỉ bảo chúng em trong quá trình hoàn thành đồ án lập trình tính toán.

Chúng em cũng xin cảm ơn các thầy cô giáo trong trường Đại học Bách Khoa – Đại học Đà Nẵng đã dạy bảo và truyền đạt cho chúng em nhiều kiến thức để ứng dụng vào trong đồ án.

C++ là một loại [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh" \o "Ngôn ngữ lập trình) bậc trung (middle-level). Đây là [ngôn ngữ lập trình đa năng](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_%C4%91a_n%C4%83ng" \o "Ngôn ngữ đa năng)  như một phần mở rộng của [ngôn ngữ lập trình C](https://vi.wikipedia.org/wiki/C_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)" \o "C (ngôn ngữ lập trình)), hoặc "C với các lớp Class", Ngôn ngữ đã được mở rộng đáng kể theo thời gian và C ++ hiện đại có các tính năng: [lập trình tổng quát](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_t%E1%BB%95ng_qu%C3%A1t" \o "Lập trình tổng quát), [lập trình hướng đối tượng](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng" \o "Lập trình hướng đối tượng), [lập trình thủ tục](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_th%E1%BB%A7_t%E1%BB%A5c), [ngôn ngữ đa mẫu hình](https://vi.wikipedia.org/w/index.php?title=Ng%C3%B4n_ng%E1%BB%AF_%C4%91a_m%E1%BA%ABu_h%C3%ACnh&action=edit&redlink=1" \o "Ngôn ngữ đa mẫu hình (trang chưa được viết)) tự do có [kiểu tĩnh](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh" \l "STATIC" \o "Ngôn ngữ lập trình), [dữ liệu trừu tượng](https://vi.wikipedia.org/w/index.php?title=D%E1%BB%AF_li%E1%BB%87u_tr%E1%BB%ABu_t%C6%B0%E1%BB%A3ng&action=edit&redlink=1" \o "Dữ liệu trừu tượng (trang chưa được viết)), và [lập trình đa hình](https://vi.wikipedia.org/w/index.php?title=L%E1%BA%ADp_tr%C3%ACnh_%C4%91a_h%C3%ACnh&action=edit&redlink=1" \o "Lập trình đa hình (trang chưa được viết)), ngoài ra còn có thêm các tính năng, công cụ để thao tác với [bộ nhớ cấp thấp](https://vi.wikipedia.org/w/index.php?title=B%E1%BB%99_nh%E1%BB%9B_c%E1%BA%A5p_th%E1%BA%A5p&action=edit&redlink=1" \o "Bộ nhớ cấp thấp (trang chưa được viết)). Vì vậy việc học C++ trở nên rất cấp thiết và quan trọng nếu chúng ta là người yêu lập trình, muốn học và phát triển trong ngành CNTT. Nói là nền tảng nhưng không phải C++ là một ngôn ngữ dễ tiếp cận, điển hình các phần như danh sách liên kết, lớp Class, đa năng hóa, … cũng đang làm khó các bạn lập trình viên của chúng ta nếu chưa hiểu bản chất của chúng. Ngoài ra việc sử dụng chúng để giải quyết vấn đề cũng trở nên nan giải với chúng ta. Vì vậy trong khuôn khổ của đồ án lập trình cơ sở này chúng em xin sử dụng các kiến thức mà thấy cô đã trang bị và sự tìm tòi từ sách vở và bạn bè để có thể làm rõ một phần những khó khăn của các bạn sinh viên trong quá trình học tập. Cụ thể chúng em sẽ dùng kiến thức về lập trình hướng đối tượng và cấu trúc dữ liệu để viết nên chương trình Quản lí bán hàng Siêu Thị Winmart.

Do thời gian có hạn, cũng như kinh nghiệm và kiến thức còn nhiều hạn chế nên trong

chương trình cũng như báo cáo này sẽ không tránh khỏi những thiếu sót, hạn chế nhất

định. Em rất mong nhận được nhận xét của thầy cô và các bạn.

Em xin chân thành cảm ơn!

# GIỚI THIỆU ĐỀ TÀI

Việc ứng dụng Công Nghệ Thông Tin vào đời sống ngày nay đã không còn là điều xa lạ đối với mỗi chúng ta.Công Nghệ Thông Tin đã phát triển với tốc độ nhanh chóng, chính xác đã giúp con người trong rất nhiều lĩnh vực. Do vậy việc áp dụng công nghệ vào việc quản lí bán hàng là điều cần thiết.Quản lí bán hàng siêu thị là một trong những mục tiêu cần đạt được, nhằm nâng cao hiệu quả quản lí, cũng như hỗ trợ tốt cho việc cập nhật dữ liệu hàng hóa.Phần lớn các công ty, chủ quản lí hiện tại chưa đáp ứng yêu cầu công việc như: Giao diện không thân thiện, người dùng gặp lỗi,…. Xuất phát từ những nhu cầu trên, chúng em đã chọn đề tài “ Quản lí bán hàng Siêu Thị Winmart “ nhằm giải quyết các vấn đề trên.

# PHÂN TÍCH CHỨC NĂNG HỆ THỐNG

## Chức năng tìm sản phẩm

Đây là chức năng giúp người quản lý xem hết được thông tin của tất cả các loại mặt hàng trong siêu thị bao gồm mã sản phẩm, tên sản phẩm, loại sản phẩm, số lượng mặt hàng, giá nhập, giá bán, giá khuyến mãi, số lượng trong kho bởi danh sách thông tin mặt hàng qua việc đọc và xuất ra trên màn hình.

## 2.2. Chức năng thêm sản phẩm

Đây là chức năng giúp người quản lí có thể thêm các mặt hàng mới chưa có và các thông tin các mặt hàng. Các thông tin về các mặt hàng bao gồm mã sản phẩm, tên sản phẩm, giá nhập, giá bán, số lượng.

Khi thêm mặt hàng người quản lý cần phải chọn danh mục cho mặt hàng.

## Chức năng sửa sản phẩm

Đây là chức năng hỗ trợ người quản lí có thể sửa các thông tin trên các loại mặt hàng. Ví dụ vào những dịp sale thì người quản lí có thể thay đổi các thông tin về giá bán, giá khuyến mãi

Các thao tác bao gồm :

* Chuyển danh mục của sản phẩm
* Thêm loại sản phẩm
* Thay đổi tên sản phẩm
* Thêm loại sản phẩm
* Xoá loại sản phẩm
* Sửa giá sản phẩm
* Nhập thêm sản phẩm
* Xoá sản phẩm

## Chức năng tạo voucher

Tạo voucher để cho khách hàng giảm giá khi mua sản phẩm.

# THIẾT KẾ CẤU TRÚC DỮ LIỆU

## Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống.

Các cấu trúc dữ liệu sử dụng trong hệ thống

* **vector template** : Gồm một con trỏ được cấp phát động để lưu trữ các đối tượng. Tự động tăng kích thước khi thêm sản phẩm.

Định nghĩa lớp **vector :**

#pragma once

#include <initializer\_list>

#include <stddef.h>

template <class DATA\_T>

class VECTOR

{

private:

    DATA\_T \*Container;

    size\_t CurSize;

    size\_t AlloSize;

private:

    void expand();

public:

    VECTOR();

    VECTOR(size\_t);

    VECTOR(size\_t, DATA\_T);

    VECTOR(const VECTOR<DATA\_T> &rvec);

    VECTOR(VECTOR<DATA\_T> &&rvec);

    VECTOR(std::initializer\_list<DATA\_T> &&init\_list);

    ~VECTOR();

public:

    void push\_back(const DATA\_T &KEY);

    void push\_back(DATA\_T &&KEY);

    void pop\_back();

    void push\_front(const DATA\_T &KEY);

    void push\_front(DATA\_T &&KEY);

    void pop\_front();

    void insert(const size\_t &idx, const DATA\_T &KEY);

    void insert(const size\_t &idx, DATA\_T &&KEY);

    DATA\_T &operator[](const size\_t &INDEX);

    DATA\_T &operator[](size\_t &&INDEX);

    const DATA\_T &operator[](const size\_t &INDEX) const;

    const DATA\_T &operator[](size\_t &&INDEX) const;

    void remove(const size\_t &INDEX);

    void remove(size\_t &&INDEX);

    void \*find(const DATA\_T &KEY);

    void \*find(DATA\_T &&KEY);

    const size\_t &size() const;

public:

    VECTOR &operator=(const VECTOR &rvec);

    VECTOR &operator=(VECTOR &&rvec);

public:

    DATA\_T \*begin() const;

    DATA\_T \*end() const;

};

#include "VECTOR.cc"

* **cây AVL template** : cây tìm kiếm nhị phân tự cân bằng

Định nghĩa lớp **AVLTree**:

#pragma once

#include "AVLNode.h"

template <class T>

void LNR(std::ostream &os, const std::shared\_ptr<AVLNode<T>> &node);

template <class T>

class AVLTree

{

    friend std::ostream &operator<<(std::ostream &os, const AVLTree &avlt)

    {

        LNR<T>(os, avlt.TRoot);

        return os;

    }

    static size\_t NumberOfElements;

protected:

    std::shared\_ptr<AVLNode<T>> TRoot;

private:

    std::shared\_ptr<AVLNode<T>> LLRotation(std::shared\_ptr<AVLNode<T>> Nptr);

    std::shared\_ptr<AVLNode<T>> LRRotation(std::shared\_ptr<AVLNode<T>> Nptr);

    std::shared\_ptr<AVLNode<T>> RRRotation(std::shared\_ptr<AVLNode<T>> Nptr);

    std::shared\_ptr<AVLNode<T>> RLRotation(std::shared\_ptr<AVLNode<T>> Nptr);

private:

    std::shared\_ptr<AVLNode<T>> RInsert(std::shared\_ptr<AVLNode<T>> Nptr, const T &Key, bool \*SUCCESS = nullptr);

    std::shared\_ptr<AVLNode<T>> RDelete(std::shared\_ptr<AVLNode<T>> Nptr, const T &Key, bool \*SUCCESS = nullptr);

    std::shared\_ptr<AVLNode<T>> RFindByKey(std::shared\_ptr<AVLNode<T>> Nptr, const T &Key);

public:

    void Insert(const T &Key, bool \*SUCCESS = nullptr);

    void Delete(const T &Key, bool \*SUCCESS = nullptr);

    std::shared\_ptr<AVLNode<T>> FindByKey(const T &Key);

public:

    AVLTree() = default;

    ~AVLTree() = default;

public:

    static size\_t Size()

    {

        return AVLTree::NumberOfElements;

    }

    std::shared\_ptr<AVLNode<T>> getRoot()

    {

        return TRoot;

    }

};

#include "AVLTree.cc"

* **pair template** : lưu trữ các cặp đối tượng

Định nghĩa lớp **PAIR** :

#pragma once

template <class DATA\_T1, class DATA\_T2>

class PAIR

{

public:

    DATA\_T1 first;

    DATA\_T2 second;

public:

    PAIR(const PAIR &);

    PAIR(PAIR &&);

    PAIR(const DATA\_T1 &, const DATA\_T2 &);

    PAIR(DATA\_T1 && = DATA\_T1(), DATA\_T2 && = DATA\_T2());

public:

    ~PAIR() = default;

public:

    PAIR operator=(const PAIR &);

    PAIR operator=(PAIR &&);

    bool operator<(const PAIR &);

};

#include "PAIR.cc"

**Các đối tượng và thuộc tính cần lưu trữ:**

**Mặt hàng:**

+ Mã mặt hàng(8 chữ số, bắt đầu bằng 1)

+ Tên mặt hàng

+ Số loại

+ Thông tin của mỗi loại (Tên loại, giá nhập, giá bán, giá khuyến mãi, số lượng trong kho)

+ Số bài đánh giá và điểm trung bình

Cách thức lưu trữ:

A picture containing table

Description automatically generated Background pattern

Description automatically generated with low confidence

Gồm có 19 danh mục đánh số từ 0 đến 18 là các folder. Trong mỗi folder là các file đánh số từ 0 là các danh mục con.

Hình ảnh của danh mục trong chương trình:

A screenshot of a computer

Description automatically generated with medium confidence

Cấu trúc dữ liệu dùng để lưu trữ : VECTOR

**VECTOR<VECTOR<VECTOR<ITEM>>> ITEM\_CONTAINER;**

Để truy cập một mặt hàng, cần 3 chỉ số : i,j,k lần lượt là số thứ tự của danh mục lớn, số thứ tự của danh mục con, số thứ tự của mặt hàng trong danh mục đó

* Voucher:

+ Mã voucher

+ Giá trị tối thiểu cần đạt để sử dụng voucher

+ Phần trăm giảm giá

+ Số tiền giảm giá

+ Ngày hết hạn

Định nghĩa lớp **vouche**r :

class VOUCHER

{

    std::string \_vid;

    size\_t \_vlowerBound;

    unsigned short \_vpercent;

    size\_t \_vdiscount;

    DATE \_vexpiry;

};

Cách thức lưu trữ voucher:

Dùng biến tĩnh kiểu AVLTree được quản lý bởi một class:

class VOUCHER\_MANAGER\_CLASS

{

private:

    static AVLTree<VOUCHER> VOUCHER\_DS;

};

Các phương thức cung cấp :

Khởi tạo danh sách cây VOUCHER từ file, thêm VOUCHER mới, cập nhật voucher vào file.

# PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG

# Cấu trúc hệ thống hướng đối tượng (class, object, relation)

Lớp ITEMPROP: lưu trữ các thuộc tính của một loại mặt hang

Định nghĩa lớp ITEMPROP

class ITEMPROP

{

protected:

    std::string \_unit;

    size\_t \_cost;

    size\_t \_price;

    size\_t \_promoPrice;

    size\_t \_quantity;

};

Lớp ITEM sẽ có một VECTOR các ITEMPROP

Định nghĩa lớp ITEM:

class ITEM

{

protected:

    std::string \_sku;

    std::string \_name;

    std::string \_priname;

    std::string \_origin\_priname;

    VECTOR<ITEMPROP> \_classify;

    short unsigned \_nreview;

    float \_medscore;

};

Hai thuộc tính \_priname và \_origin\_priname để giúp tìm kiếm mặt hàng.

Ví dụ một mặt hàng có mã mặt hàng là **10023128**, tên mặt hàng là **Bộ drap gối bọc cotton hoa 160 x 200 cm (Nhiều màu)** thì \_priname và \_origin\_priname có dạng :

\_priname : 10023128 bo drap goi boc cotton hoa 160 x 200 cm nhieu mau

\_origin\_priname: 10023128 bộ drap gối bọc cotton hoa 160 x 200 cm nhiều màu

Hình ảnh các mặt hàng được lưu trữ trong file:

Text

Description automatically generated

Các mặt hàng được lưu trữ trong một biến tĩnh được quản lý bởi một class

class ITEM\_DB\_MANAGER

{

static VECTOR<VECTOR<VECTOR<ITEM>>> ITEM\_CONTAINER;

};

Class này cung cấp các phương thức khởi tạo, tìm kiếm, sửa, cập nhật dữ liệu.

**Phương thức khởi tạo và cập nhật dữ liệu do class cung cấp:**

Sử dụng các luồng để đọc, ghi dữ liệu từ các file, tổng cộng có 74 luồng.

**Phương thức tìm kiếm mặt hàng:**

Sử dụng các luồng và thuật toán SkipSearch để tìm chuỗi từ người dùng nhập vào với tên sản phẩm.

Cài đặt thuật toán SkipSearch:

#pragma once

#include <cstring>

#include <iostream>

using namespace std;

#define ASize 256

class \_cell

{

public:

    unsigned int element;

    \_cell \*next;

public:

    \_cell(unsigned int element = 0, \_cell \*next = nullptr)

        : element{element}, next{next}

    {}

};

typedef \_cell \*List;

void Allocate(List \*X, const char \*str, const size\_t &len)

{

    List p{nullptr};

    for (size\_t &&i{0}; i < len; ++i)

    {

        p = new \_cell(i, X[str[i]]);

        X[str[i]] = p;

    }

}

void DeAllocate(List \*X, const char \*str, const size\_t &len)

{

    List p{nullptr};

    for (size\_t &&i{0}; i < len; i++)

    {

        while (X[str[i]])

        {

            p = X[str[i]];

            X[str[i]] = p->next;

            delete p;

        }

    }

}

bool SKIPSEARCH(const char \*pat, const char \*txt)

{

    size\_t patlen = strlen(pat);

    size\_t txtlen = strlen(txt);

    if (patlen > txtlen)

        return false;

    char \*pattmp = new char[patlen + 1];

    strcpy(pattmp, pat);

    char \*txttmp = new char[txtlen + 1];

    strcpy(txttmp, txt);

    for (size\_t &&i{0}; i < patlen; ++i)

        if (pattmp[i] < 0)

            pattmp[i] += 128;

    for (size\_t &&i{0}; i < txtlen; ++i)

        if (txttmp[i] < 0)

            txttmp[i] += 128;

    List X[ASize] = {nullptr};

    List p{nullptr};

    Allocate(X, pattmp, patlen);

    for (size\_t i = patlen - 1; i < txtlen; i += patlen)

    {

        for (p = X[txttmp[i]]; p != nullptr; p = p->next)

        {

            if (memcmp(pattmp, txttmp + i - p->element, patlen) == 0 &&

                (i - p->element) <= (txtlen - patlen))

            {

                DeAllocate(X, pattmp, patlen);

                delete[] pattmp;

                delete[] txttmp;

                return true;

            }

        }

    }

    DeAllocate(X, pattmp, patlen);

    delete[] pattmp;

    delete[] txttmp;

    return false;

}

# Kết quả

# 4.2.1 Giao diện chính của chương trình

Timeline

Description automatically generated

*Table

Description automatically generated*

### Kết quả thực thi của chương trình

* **Chức năng tìm sản phẩm**

Graphical user interface, application

Description automatically generated

* Hiển thị thông tin sản phẩm có chứa chuỗi ký tự input:

Graphical user interface, application

Description automatically generated

* **Chức năng thêm sản phẩm**

*Table

Description automatically generated*

*Graphical user interface, text, application

Description automatically generated*

* **Chức năng sửa sản phẩm**

Background pattern

Description automatically generated with low confidence

**A picture containing table

Description automatically generated**

**Chức năng tạo VOUCHER**

### A picture containing graphical user interface Description automatically generated

Voucher được tạo lưu trong file:

Table

Description automatically generated

Kết thúc chương trình

### Chart Description automatically generated with medium confidence4.2.3 Nhận xét đánh giá

-Ưu điểm:

+) Giao diện đồ họa bắt mắt

+) Chương trình đã có những chức năng cơ bản dễ dàng sử dụng trong việc mô phỏng chương trình quản lí bán hàng siêu thị winmart

+) Ứng dụng được các thuật toán tìm kiếm tuần tự, skipsearch vào trong chương trình

-Nhược điểm:

+) Tuy giao diện đồ họa bắt mặt nhưng chương trình vẫn còn hết sức đơn giản

+) Nguồn dữ liệu các mặt hàng vẫn còn khá ít so với dự liệu trên thực tế

+) Tính năng chưa đa dạng

+) Chương trình còn nhiều bất cập khi nhập nhập input không phù hợp với kiều dữ liệu

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Trong quá trình làm đồ án PBL2 chúng em đã có thể tiếp thu những kiến thức của C++, lập trình hướng đối tượng(OOP), cấu trúc dữ liệu và đã biết kết hợp những kiến thức đó để cho ra được một chương trình tương đối hoàn thiện với mục tiêu và điều kiện của đề tài. Tuy nhiên để so sánh mới một hệ thống chương trình Quản lý bán hàng siêu thị Winmart thực tế, chúng em cảm thấy chương trình của mình còn đang rất đơn giản, lạc hậu.

## Hướng phát triển

-Cải thiện thêm giao diện cho chương trình.

-Tiếp tục sửa các lỗi còn tồn tại trong chương trình để chương trình hoàn thiện.

-Sử dụng cấu trúc dữ liệu một cách tối ưu để rút gọn phần viết code của chương trình nhằm giảm tối thiểu bộ nhớ và thời gian thực thi chương trình.

-Phát triển thêm nhiều tính năng tiện ích để làm phong phú hơn chức năng của ứng dụng

# TÀI LIỆU THAM KHẢO

[1] Beginning C++ Programming – From Beginer to Beyond

Dr.Frank Mitropoulos

https://www.udemy.com/course/beginning-c-plus-plus-programming/

[2] Mastering Data Structures & Algorithms using C and C++

Abdul Bari

https://www.udemy.com/course/datastructurescncpp/

[3] Learn Multithreading with Modern C++

James Raynard

<https://www.udemy.com/course/learn-modern-cplusplus-concurrency/>