**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**

**MÔN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**ĐỀ TÀI: QUẢN LÝ SIÊU THỊ**

Giảng viên hướng dẫn: TRẦN THỊ DUNG

Sinh viên thực hiện: LÊ THÁI BẢO – PHẠM QUỐC THUẬN – NHẬT TÂN

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá :64

Tp. Hồ Chí Minh, năm 2024

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**

**MÔN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**ĐỀ TÀI: QUẢN LÝ SIÊU THỊ**

Giảng viên hướng dẫn: TRẦN THỊ DUNG

Sinh viên thực hiện: LÊ THÁI BẢO – PHẠM QUỐC THUẬN – NHẬT TÂN

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá :64

Tp. Hồ Chí Minh, năm 2024

**LỜI CẢM ƠN**

Lời đầu tiên chúng em xin phép gửi đến Quý Thầy Cô của Bộ môn Công Nghệ Thông Tin trường Đại học Giao thông Vận Tải – Phân hiệu tại Tp Hồ Chí Minh lời chúc sức khỏe và lời cảm ơn sâu sắc nhất. Thầy cô đã tận tình hướng dẫn, truyền đạt kiến thức, kinh nghiệm cho chúng em trong suốt quá trình học tập và rèn luyện ở tại trường.

Kế tiếp, chúng em xin cảm ơn nhà trường đã tạo điều kiện tốt nhất cho chúng em, để chúng em có một môi trường học tập thật tốt. Đặc biệt cảm ơn đến các giảng viên của Bộ môn Công nghệ Thông tin đã truyền đạt và dạy dỗ chúng em rất nhiều kiến thức trong quãng thời gian học tập vừa qua.

Cảm ơn đến thầy cô giảng viên và các thầy cô thỉnh giảng khác đã dành nhiều thời gian và tâm huyết để truyền đạt những kiến thức cho chúng em.

Đặc biệt, chúng em xin gửi lời cảm ơn đến cô Trần Thị Dung, cô đã tận tình hướng dẫn chỉ bảo để chúng em có thể hoàn thành bài báo cáo.

Cảm ơn đến toàn thể các bạn thuộc lớp Công nghệ Thông tin K64 đã luôn giúp đỡ mình trong thời gian học tập cùng nhau.

Mặc dù em đã cố gắng hết sức trong quá trình thực hiện bài báo cáo này, nhưng vì thời gian có hạn và nhóm chúng em cũng chưa có kinh nghiệm làm việc với bài toán thực tế, nên chắc chắn sẽ có nhiều thiếu sót.

Lời sau cùng, chúc sức khỏe đến tất cả các thành viên trong gia đình, toàn thể giảng viên của trường Đại học Giao thông Vận tải phân hiệu tại Thành phố Hồ Chí Minh đặc biệt là các giảng viên thuộc Bộ Môn Công nghệ Thông tin, các bạn thuộc lớp Công nghệ Thông tin K64, các bạn bè khác của tôi. Chúng em xin chân thành cảm ơn!

# MỤC LỤC

[**CHƯƠNG 1. MỞ ĐẦU** 1](#_Toc183038451)

[1.1. Tổng quan về đề tài: 1](#_Toc183038452)

[1.2. Mục tiêu nghiên cứu: 1](#_Toc183038453)

[1.3. Phạm vi nghiên cứu: 1](#_Toc183038454)

[**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT** 3](#_Toc183038455)

[2.1. Tổng quan về ngôn ngữ lập trình C++ 3](#_Toc183038456)

[2.1.1 Khái niệm. 3](#_Toc183038457)

[2.1.2 Các đặc điểm nổi bật của ngôn ngữ lập trình C++ 3](#_Toc183038458)

[2.1.3 Ưu nhược điểm của ngôn ngữ C++ 3](#_Toc183038459)

[2.2. Lập trình hướng đối tượng trong ngôn ngữ C++ 4](#_Toc183038460)

[2.2.1 Lập trình hướng đối tượng là gì? 4](#_Toc183038461)

[2.2.2 Các tính chất cơ bản của OOP 4](#_Toc183038462)

[2.2.3 Các ưu điểm của lập trình hướng đối tượng 4](#_Toc183038463)

[2.3. Tổng quan về GitHub**.** 4](#_Toc183038464)

[2.4. Tổng quan về Jira 5](#_Toc183038465)

[**CHƯƠNG 3. PHÂN TÍCH, THIẾT KẾ, TRIỂN KHAI VÀ SỬ DỤNG CHƯƠNG TRÌNH THỰC TẾ** 6](#_Toc183038466)

[3.1 Phân tích yêu cầu hệ thống 6](#_Toc183038467)

[3.1.1 Yêu cầu chức năng 6](#_Toc183038468)

[3.1.2 Yêu cầu phi chức năng 7](#_Toc183038469)

[3.2 Thiết kế hệ thống và cơ sở dữ liệu 7](#_Toc183038470)

[3.2.1 Thiết kế hệ thống: 7](#_Toc183038471)

[3.2.2 Thiết kế cơ sở dữ liệu 10](#_Toc183038473)

[3.3 Triển khai hệ thống 12](#_Toc183038474)

[3.3.1 Công nghệ sử dụng 12](#_Toc183038475)

[3.3.2 Xây dựng chức năng chính 12](#_Toc183038476)

[3.3.3 Kiểm thử và tối ưu hóa 13](#_Toc183038477)

[3.4 Sử Dụng Chương Trình Thực Tế : 14](#_Toc183038478)

[3.4.1 Khách Hàng: 15](#_Toc183038479)

[3.4.2 Nhân viên: 23](#_Toc183038480)

[3.4.3 Admin: 24](#_Toc183038481)

[**CHƯƠNG 4. KẾT QUẢ VÀ KIẾN NGHỊ** 26](#_Toc183038482)

[4.1 Kết quả đạt được: 26](#_Toc183038483)

[4.2 Đánh giá và nhận xét 26](#_Toc183038484)

[4.3 Kiến nghị 27](#_Toc183038485)

[**TÀI LIỆU THAM KHẢO** 29](#_Toc183038486)

**DANH MỤC CHỮ VIẾT TẮT**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mô tả** | **Ý nghĩa** | **Ghi chú** |
| 1 | IDE | Integrated **D**evelopment **E**nvironment |  |
| 2 | OOP | Object Oriented Programming |  |
| 3 | UX | User Experience |  |
| 4 | UI | User Interface |  |

**DANH MỤC HÌNH ẢNH**

[Hình 3.1: Lớp user của chương trình 8](#_Toc183037844)

[Hình 3.2: Lớp user của chương trình 9](#_Toc183037845)

[Hình 3.3: Lớp sản phẩm của chương trình 9](#_Toc183037846)

[Hình 3.4: Giao diện khởi đầu chương trình 15](#_Toc183037847)

[Hình 3.5: Giao diện chương trình 15](#_Toc183037849)

[Hình 3.6: Giao diện đăng ký với vai trò khách hàng 16](#_Toc183037850)

[Hình 3.7: Giao diện đăng nhập thành công 17](#_Toc183037851)

[Hình 3.8: Giao diện chức năng của khách hàng 18](#_Toc183037852)

[Hình 3.9: Giao diện chức năng của nạp tiền 19](#_Toc183037853)

[Hình 3.9: Thông tin của khách hàng 19](#_Toc183037854)

[Hình 3.10: Giao diện chức năng mua hàng 20](#_Toc183037855)

[Hình 3.11: Sản phẩm sẽ in ra theo loại bạn chọn 21](#_Toc183037856)

[Hình 3.12: Sau khi đã dừng chọn sản phẩm của 1 loại 21](#_Toc183037857)

[Hình 3.13: Giao diện chức năng thanh toán của khách hàng 21](#_Toc183037858)

[Hình 3.14: Giao diện lịch sử mua hàng 22](#_Toc183037859)

[Hình 3.15: Giao diện sau khi sử dụng chức năng đăng xuất của khách hàng 22](#_Toc183037860)

[Hình 3.16: Giao diện dừng chương trình 23](#_Toc183037861)

[Hình 3.17: Giao diện đăng nhập của nhân viên 23](#_Toc183037862)

[Hình 3.18: Giao diện chức năng của nhân viên 24](#_Toc183037863)

[Hình 3.19: Giao diện đăng nhập của quản lý 24](#_Toc183037864)

[Hình 3.20: Giao diện chức năng của quản lý 25](#_Toc183037865)

# CHƯƠNG 1. MỞ ĐẦU

* 1. **Tổng quan về đề tài:**

Trong bối cảnh ngành bán lẻ phát triển không ngừng, việc quản lý hiệu quả các hoạt động trong siêu thị là yếu tố quan trọng giúp tối ưu hóa công việc, giảm thiểu sai sót và nâng cao trải nghiệm của khách hàng. Ứng dụng lập trình hướng đối tượng (OOP) là một phương pháp mạnh mẽ, cho phép tổ chức hệ thống một cách rõ ràng, dễ bảo trì và mở rộng. Báo cáo này tập trung vào việc xây dựng một phần mềm quản lý siêu thị sử dụng các nguyên tắc OOP trong C++, nhằm đảm bảo tính linh hoạt, hiệu quả và khả năng mở rộng của hệ thống.

* 1. **Mục tiêu nghiên cứu:**
* Áp dụng các khái niệm của lập trình hướng đối tượng như lớp, kế thừa, đa hình và đóng gói để xây dựng một hệ thống quản lý siêu thị hoàn chỉnh.
* Đảm bảo hệ thống có thể quản lý thông tin sản phẩm, khách hàng, hóa đơn và các hoạt động bán hàng một cách hiệu quả.
* Cung cấp giao diện người dùng dễ sử dụng, có khả năng cập nhật và hiển thị thông tin theo thời gian thực.
  1. **Phạm vi nghiên cứu:**

**Phạm vi nghiên cứu của đề tài bao gồm:**

* Xây dựng cấu trúc dữ liệu và các lớp chính phục vụ cho việc quản lý sản phẩm, khách hàng và giao dịch.
* Tạo một giao diện cơ bản để nhân viên siêu thị có thể thực hiện các thao tác như thêm, sửa, xóa thông tin sản phẩm và khách hàng, lập hóa đơn và báo cáo doanh số.
* Sử dụng ngôn ngữ lập trình C++ với các thư viện hỗ trợ để phát triển phần mềm, đảm bảo hiệu suất và tính ổn định của hệ thống.
* Giới hạn nghiên cứu trong các chức năng cơ bản của một hệ thống quản lý siêu thị, không bao gồm các tính năng phức tạp như quản lý kho hàng tự động hoặc tích hợp hệ thống thanh toán trực tuyến

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## **2.1. Tổng quan về ngôn ngữ lập trình C++**

### **2.1.1 Khái niệm** C++ là ngôn ngữ lập trình bậc trung phát triển bởi Bjarne Stroustrup vào năm 1979. Nó kết hợp tính năng của ngôn ngữ bậc cao và bậc thấp, được sử dụng để phát triển cả phần mềm hệ thống và ứng dụng bậc cao.

### **2.1.2 Các đặc điểm nổi bật của ngôn ngữ lập trình C++**

* Có thể sử dụng C++ để phát triển những chương trình bậc thấp và bậc cao, hoạt động tốt trên hầu hết phần cứng.
* Cú pháp của C++ dễ hiểu, khá giống tiếng Anh. Có thể dùng C++ để thiết kế các ứng dụng từ hợp ngữ (Assembly language).
* C++ được hỗ trợ trên nhiều nền tảng, hệ điều hành. Chương trình viết trên Linux có thể biên dịch trên Windows và chạy trên Mac OS.
* Hỗ trợ hiệu quả lập trình hướng đối tượng, bao gồm trừu tượng hóa, đóng gói, kế thừa và đa hình.
* C++ cung cấp công cụ con trỏ, cho phép lưu địa chỉ bộ nhớ, truy cập và thao tác dữ liệu từ bộ nhớ.

### **2.1.3 Ưu nhược điểm của ngôn ngữ C++** *Ưu điểm:*

* **Lập trình hướng đối tượng:** Hỗ trợ các tính chất như đóng gói, đa hình, trừu tượng và kế thừa.
* **Quản lý vùng nhớ:** Không có Garbage Collector, cho phép tái sử dụng vùng nhớ để cải thiện hiệu suất.
* **Hiệu suất cao:** Giao tiếp tốt với phần cứng, phù hợp phát triển hệ thống nhúng, game và ứng dụng yêu cầu hiệu suất cao.

*Nhược điểm:*

* **Tràn vùng nhớ:** Quản lý vùng nhớ không tốt sẽ dẫn đến lỗi tràn vùng nhớ.
* **OOP phức tạp:** Khái niệm như hàm thuần ảo, hàm bạn, lớp bạn, đa kế thừa… làm cho OOP trong C++ khó hiểu.
* **Con trỏ phức tạp:** Con trỏ giúp thao tác trực tiếp vùng nhớ nhưng khó học và dễ mắc lỗi.

## **2.2. Lập trình hướng đối tượng trong ngôn ngữ C++**

### **2.2.1 Lập trình hướng đối tượng là gì?** Lập trình hướng đối tượng (Object-Oriented Programming – OOP) là kỹ thuật lập trình quan trọng, hỗ trợ bởi hầu hết ngôn ngữ lập trình hiện nay như Java, PHP, .NET, Ruby, Python... OOP cho phép lập trình viên trừu tượng hóa và tổ chức code thành các đối tượng.

### **2.2.2 Các tính chất cơ bản của OOP**

* **Tính đóng gói (Encapsulation):** Thuộc tính và phương thức được đóng gói trong lớp, che giấu thông tin và hạn chế truy cập trực tiếp từ bên ngoài.
* **Tính kế thừa (Inheritance):** Cho phép lớp mới thừa hưởng thuộc tính, phương thức từ lớp cha, tối ưu và tái sử dụng mã nguồn.
* **Tính đa hình (Polymorphism):** Một hành động có thể được thực hiện theo nhiều cách khác nhau.
* **Tính trừu tượng (Abstraction):** Tổng quát hóa, chọn lọc các thuộc tính và phương thức cần thiết để giải quyết bài toán.

### **2.2.3 Các ưu điểm của lập trình hướng đối tượng**

* Loại bỏ chương trình lặp và dư thừa, mở rộng khả năng sử dụng mã nguồn.
* Rút ngắn thời gian xây dựng hệ thống, tăng năng suất.
* Giúp biểu diễn tốt hơn thế giới thực trên máy tính.

## **2.3. Tổng quan về GitHub**

## GitHub là nền tảng quản lý mã nguồn và cộng tác dành cho các dự án phần mềm, sử dụng hệ thống kiểm soát phiên bản Git.

**Ý nghĩa và lợi ích của GitHub:**

**Quản lý mã nguồn:** Lưu trữ và tổ chức mã nguồn hiệu quả.

**Cộng tác:** Hỗ trợ làm việc nhóm dễ dàng.

**Kiểm soát phiên bản:** Theo dõi và quản lý thay đổi mã nguồn.

**Mã nguồn mở:** Chia sẻ và đóng góp vào các dự án mã nguồn mở.

**Tích hợp và tự động hóa:** Liên kết với các công cụ DevOps.

**Quản lý dự án:** Theo dõi tiến độ và công việc liên quan.

## **2.4. Tổng quan về Jira**

## Jira là công cụ quản lý dự án và theo dõi vấn đề (issue tracking) do Atlassian phát triển. Ban đầu dùng để theo dõi lỗi trong phần mềm, nay đã trở thành công cụ quản lý đa năng.

**Ý nghĩa và lợi ích của Jira:**

* **Theo dõi công việc và vấn đề:** Quản lý nhiệm vụ và các lỗi phát sinh.
* **Hỗ trợ quản lý Agile:** Phương pháp Agile được tích hợp trực tiếp.
* **Quản lý và báo cáo dự án:** Theo dõi tiến độ và tạo báo cáo.
* **Tùy chỉnh linh hoạt:** Thích ứng với nhu cầu của từng dự án.
* **Tích hợp với các công cụ khác:** Kết nối với GitHub, Jenkins...
* **Quản lý quy trình làm việc:** Tự động hóa và tối ưu hóa quy trình.

# CHƯƠNG 3. PHÂN TÍCH, THIẾT KẾ, TRIỂN KHAI VÀ SỬ DỤNG CHƯƠNG TRÌNH THỰC TẾ

**3.1 Phân tích yêu cầu hệ thống**

**3.1.1 Yêu cầu chức năng**

**Quản lý:**

* **Nhập hàng:** Thêm sản phẩm mới vào danh sách sản phẩm.
* **Xóa sản phẩm**: Loại bỏ hoàn toàn một sản phẩm khỏi hệ thống.
* **Quản lý nhân viên:** Thêm, chỉnh sửa và xóa thông tin nhân viên.
* **Đăng nhập với giao diện nhân viên:** Truy cập giao diện làm việc như một nhân viên để hỗ trợ công việc.
* **Đăng nhập với giao diện khách hàng:** Đăng nhập để kiểm tra trải nghiệm của khách hàng.

**Nhân viên:**

* **Quản lý khách hàng:** Thêm, sửa và xóa thông tin khách hàng.
* **Xem hóa đơn:** Truy cập và xem chi tiết hóa đơn đã phát hành.
* **Xóa tài khoản:** Xóa tài khoản khách hàng khi cần thiết.

**Khách hàng:**

* **Đổi mật khẩu**: Cập nhật mật khẩu để bảo mật tài khoản.
* **Mua sản phẩm và thanh toán:** Thêm sản phẩm vào giỏ hàng và thực hiện thanh toán.
* **Nạp tiền vào tài khoản:** Tăng số dư để sử dụng cho việc mua sắm.
* **Xem lịch sử mua hàng:** Kiểm tra các đơn hàng đã thực hiện.

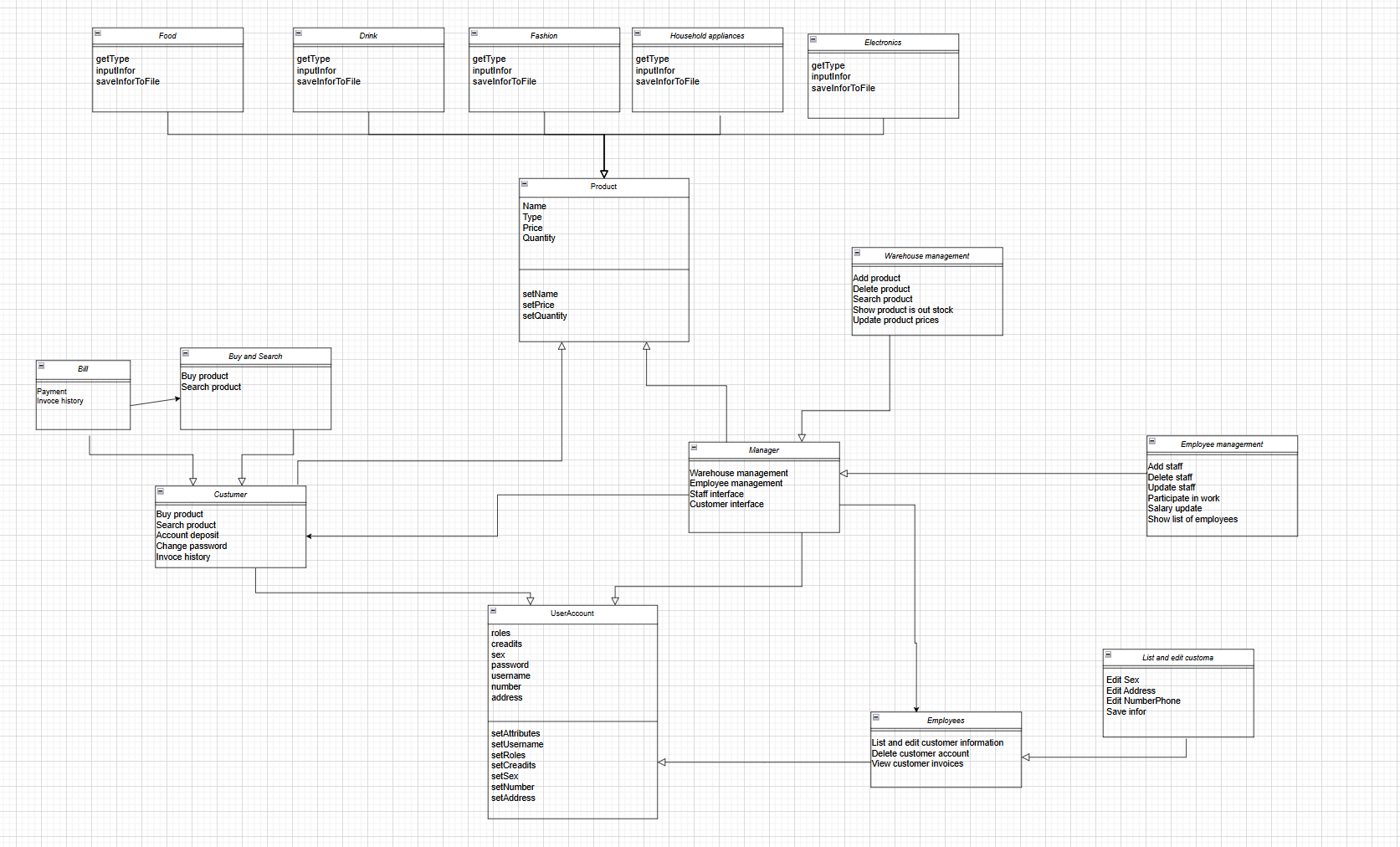
**3.1.2 Yêu cầu phi chức năng**

* **Hiệu suất**: Hệ thống cần được thiết kế để xử lý khối lượng dữ liệu lớn một cách nhanh chóng và hiệu quả, đảm bảo thời gian phản hồi ngắn khi thực hiện các thao tác như truy vấn, cập nhật thông tin và xử lý giao dịch. Điều này yêu cầu việc sử dụng các thuật toán tối ưu và cấu trúc dữ liệu hiệu quả
* **Tính bảo mật**: Hệ thống phải bảo vệ dữ liệu người dùng bằng cách áp dụng các phương pháp mã hóa dữ liệu và cơ chế xác thực người dùng nghiêm ngặt. Bảo mật cần được đặt lên hàng đầu để đảm bảo rằng thông tin cá nhân và giao dịch không bị truy cập trái phép hoặc lộ ra bên ngoài.
* **Tính dễ sử dụng**: Giao diện người dùng cần được thiết kế thân thiện, dễ hiểu và trực quan. Người dùng ở mọi trình độ, từ nhân viên đến quản lý, đều có thể thao tác hệ thống mà không gặp khó khăn. Cách bố trí các chức năng cần hợp lý để giảm thiểu thời gian học cách sử dụng.
* **Tính ổn định và bảo trì**: Hệ thống cần hoạt động liên tục, không bị gián đoạn và không xảy ra lỗi không mong muốn. Mã nguồn phải được viết sạch sẽ và có cấu trúc rõ ràng, cho phép việc bảo trì, nâng cấp hoặc sửa chữa dễ dàng khi cần thiết. Điều này giúp hệ thống có tuổi thọ dài hơn và dễ dàng cập nhật thêm tính năng mới.

**3.2 Thiết kế hệ thống và cơ sở dữ liệu**

**3.2.1 Thiết kế hệ thống:**

* **Sơ đồ cây lớp của chương trình:**

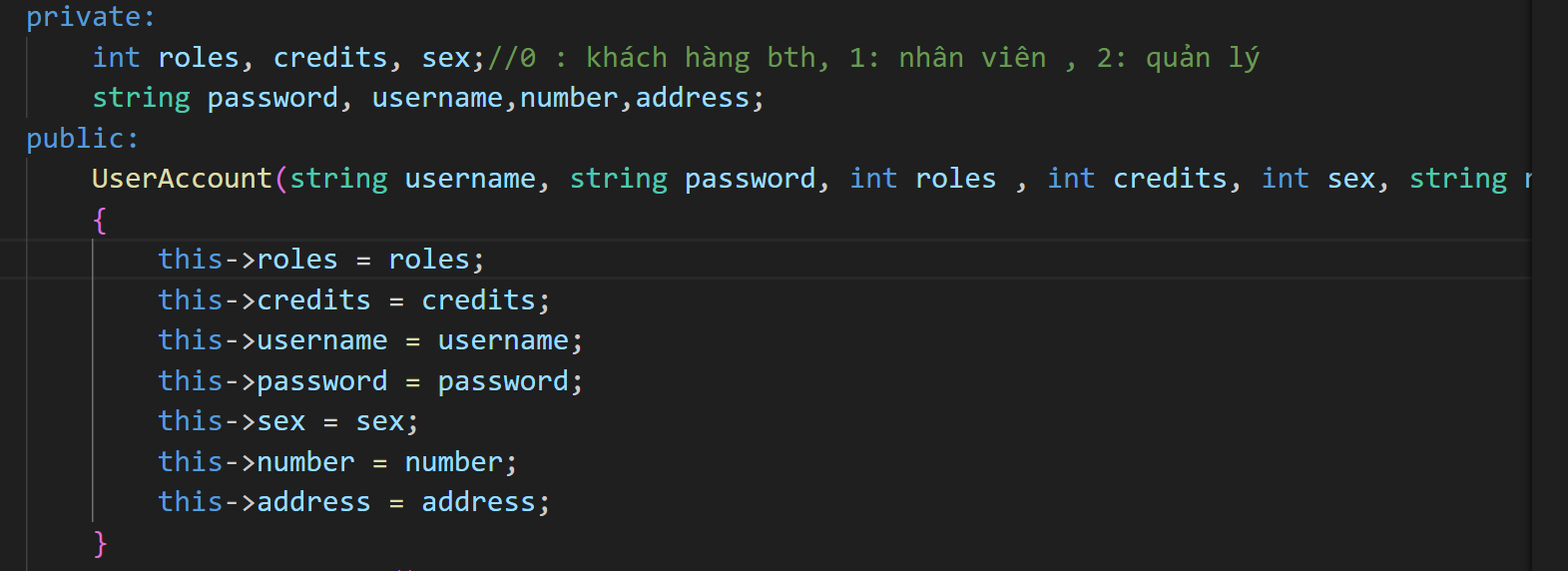


Hình 3.1: Lớp user của chương trình

* **Kiến trúc hệ thống:**

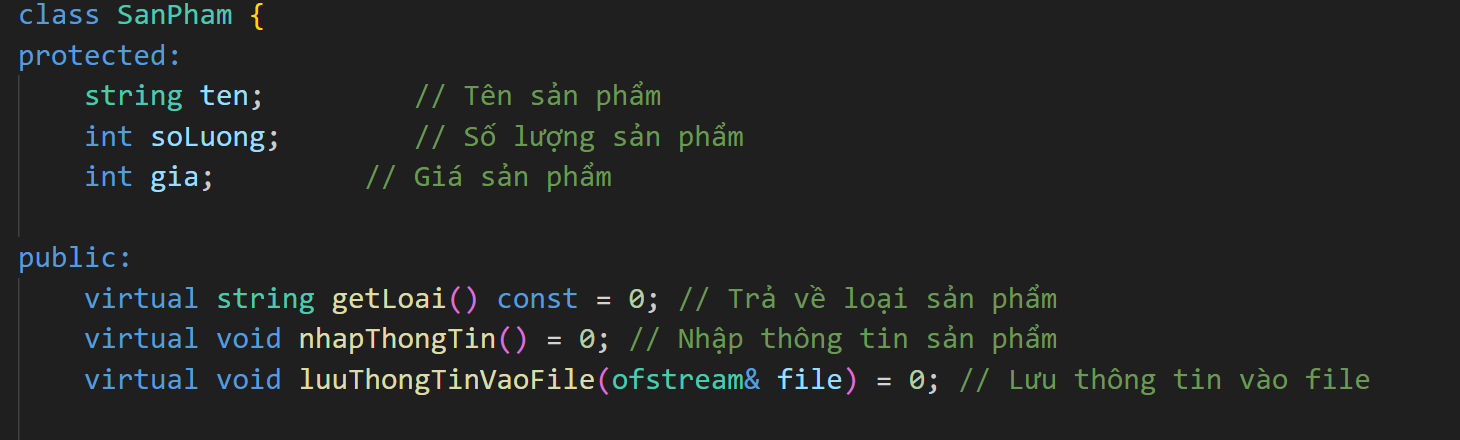
Hệ thống sẽ được xây dựng dựa trên mô hình lập trình hướng đối tượng (OOP), với các lớp sau:

* + - User: Lớp này đại diện cho người dùng trong hệ thống, có thể là quản lý, nhân viên hoặc khách hàng. Lớp này bao gồm các thuộc tính như tên đăng nhập, mật khẩu, và quyền truy cập.



Hình 3.2: Lớp user của chương trình

* + - SanPham: Lớp này sẽ lưu trữ thông tin về sản phẩm, bao gồm mã sản phẩm, tên sản phẩm, giá, số lượng tồn kho và loại sản phẩm. Sản phẩm sẽ được phân loại thành các loại như:



Hình 3.3: Lớp sản phẩm của chương trình

* + - * ThucPham: Lớp con của SanPham, đại diện cho các sản phẩm thực phẩm.
      * DoUong: Lớp con của SanPham, đại diện cho các sản phẩm đồ uống.
      * DoGiaDung: Lớp con của SanPham, đại diện cho các sản phẩm đồ gia dụng.
      * ThoiTrang: Lớp con của SanPham, đại diện cho các sản phẩm thời trang.
      * DoDienTu: Lớp con của SanPham, đại diện cho các sản phẩm điện tử.
    - QuanLy: Lớp này đại diện cho quản lý, có các phương thức để quản lý các chức năng trong hệ thống như nhập hàng, xóa sản phẩm, và quản lý nhân viên.
    - KhachHang: Lớp này lưu trữ thông tin của khách hàng, bao gồm mã khách hàng, tên, địa chỉ, số điện thoại và các phương thức để quản lý thông tin khách hàng.
    - NhanVien: Lớp này đại diện cho nhân viên, có các thuộc tính như mã nhân viên, tên, và các chức năng để quản lý khách hàng và hóa đơn.
* **Lưu đồ hoạt động:**
  + Lưu đồ sẽ mô tả quy trình từ khi khách hàng chọn sản phẩm đến khi hoàn tất thanh toán, bao gồm các bước chính như:
    - Khách hàng chọn sản phẩm từ danh sách.
    - Thêm sản phẩm vào giỏ hàng.
    - Kiểm tra giỏ hàng.
    - Thực hiện thanh toán.
    - Nhận hóa đơn.
* **Sơ đồ lớp:**
  + Sơ đồ lớp sẽ thể hiện các lớp đã nêu cùng với các mối quan hệ giữa chúng. Sơ đồ này sẽ chỉ ra cách các lớp SanPham, ThucPham, DoUong, DoGiaDung, ThoiTrang, và DoDienTu kết nối với nhau, cũng như mối quan hệ của chúng với các lớp KhachHang, NhanVien, và QuanLy.

**3.2.2 Thiết kế cơ sở dữ liệu**

**Cấu trúc cơ sở dữ liệu:**

Cơ sở dữ liệu bao gồm các bảng chính như sau:

**Bảng Products:**

* ProductID: Mã, INT.
* ProductName: String.
* Price: FLOAT.
* QuantityInStock: INT.
* ProductType: String (Thực phẩm, Đồ uống, Đồ gia dụng, Thời trang, Đồ điện tử).

**Bảng Users:**

* UserID: Mã, INT.
* Username: String.
* Password: String.
* UserType: String (Quản lý, Nhân viên, Khách hàng).

**Bảng Customers:**

* CustomerID: Mã, INT.
* CustomerName: String.
* Address: String.
* Phone: String.

**Bảng Invoice:**

InvoiceDetailID: Mã, INT.

Quantity: INT.

Date: DATETIME (ngày mua hàng).Mô hình ER (Entity-Relationship)

Mô hình ER minh họa mối quan hệ giữa các thực thể trong cơ sở dữ liệu:

* Products có mối quan hệ một hay nhiều với Invoice (mỗi sản phẩm có thể xuất hiện trong nhiều chi tiết hóa đơn).
* Customers có mối quan hệ một-nhiều với Invoices (mỗi khách hàng có thể có nhiều hóa đơn).
* Users liên kết với cả Customers và Invoices để quản lý thông tin người dùng và quyền truy cập.

=> Mô hình ER giúp hình dung cấu trúc dữ liệu và kết nối giữa các bảng, hỗ trợ hiệu quả trong thiết kế và phát triển hệ thống quản lý siêu thị.

**3.3 Triển khai hệ thống**

**3.3.1 Công nghệ sử dụng**

**Ngôn ngữ lập trình:**

* Sử dụng C++ để phát triển hệ thống quản lý siêu thị. C++ là ngôn ngữ lập trình hướng đối tượng mạnh mẽ, cho phép xây dựng các lớp và đối tượng, giúp quản lý mã nguồn hiệu quả.
* Các thư viện hỗ trợ bao gồm:

**<fstream>:** Đọc/ghi dữ liệu từ/đến tệp.

**<string>:** Xử lý và thao tác với chuỗi.

**<vector>:** Mảng động, lưu trữ dữ liệu linh hoạt.

**<sstream>:** Xử lý chuỗi như dòng nhập, chuyển đổi kiểu dữ liệu.

**<chrono>:** Làm việc với thời gian, đo thời gian thực hiện.

**<thread>:** Quản lý luồng, thực hiện nhiệm vụ đồng thời.

**<iomanip>:** Định dạng đầu ra, điều chỉnh hiển thị dữ liệu.

**Công cụ phát triển:**

* Sử dụng các IDE (Môi trường phát triển tích hợp) như Visual Studio Code hoặc Dev-C++ để viết, biên dịch và gỡ lỗi mã nguồn. Các công cụ này cung cấp giao diện thân thiện, hỗ trợ tính năng tự động hoàn thành mã, và dễ dàng quản lý dự án.

**3.3.2 Xây dựng chức năng chính**

1. **Quản lý sản phẩm:**

* **Thêm sản phẩm:** Xây dựng một phương thức để người dùng có thể nhập thông tin sản phẩm mới (tên, giá, số lượng tồn kho, loại sản phẩm) và lưu trữ vào danh sách sản phẩm.
* **Sửa sản phẩm:** Cung cấp chức năng cho phép người quản lý cập nhật thông tin của sản phẩm đã tồn tại, như thay đổi giá, số lượng tồn kho, hoặc loại sản phẩm.
* **Xóa sản phẩm:** Cho phép xóa một sản phẩm khỏi danh sách dựa trên mã sản phẩm hoặc tên sản phẩm.
* **Tìm kiếm sản phẩm:** Cung cấp phương thức tìm kiếm sản phẩm theo tên hoặc loại để dễ dàng truy xuất thông tin.

1. **Quản lý khách hàng:**

* **Lưu trữ thông tin khách hàng:** Viết mã để tạo mới thông tin khách hàng, bao gồm tên, địa chỉ, và số điện thoại.
* **Cập nhật thông tin:** Cung cấp chức năng cho phép sửa đổi thông tin khách hàng khi có thay đổi, như thay đổi địa chỉ hoặc số điện thoại.

1. **Lập hóa đơn:**

* **Tạo hóa đơn:** Phát triển một phương thức cho phép người dùng chọn sản phẩm từ danh sách và thêm vào hóa đơn. Tính toán tổng tiền dựa trên số lượng và giá của sản phẩm.
* **Lưu hóa đơn:** Lưu trữ hóa đơn đã tạo vào cơ sở dữ liệu để có thể truy xuất và xem lại sau này.

🡺Các chức năng này sẽ tạo nên một hệ thống quản lý siêu thị toàn diện, cho phép người quản lý theo dõi và điều hành các hoạt động mua bán, đồng thời quản lý thông tin sản phẩm và khách hàng một cách hiệu quả.

**3.3.3 Kiểm thử và tối ưu hóa**

1. **Kiểm thử:**
   * **Kiểm thử đơn vị:** Thực hiện kiểm thử trên từng hàm hoặc phương thức riêng lẻ để đảm bảo rằng từng phần của mã hoạt động chính xác. Điều này bao gồm việc kiểm tra các chức năng như thêm, sửa, xóa sản phẩm, cũng như lập hóa đơn .
   * **Kiểm thử tích hợp:** Sau khi các đơn vị đã được kiểm thử thành công, tiến hành kiểm thử tích hợp để đảm bảo rằng các phần khác nhau của hệ thống hoạt động tốt cùng nhau. Ví dụ, kiểm tra xem việc thêm sản phẩm vào hóa đơn có làm cập nhật số lượng tồn kho đúng không, và việc tính toán doanh thu có chính xác hay không.
2. **Tối ưu hóa:**
   * **Điều chỉnh mã nguồn:** Xem xét và điều chỉnh mã nguồn để cải thiện hiệu suất. Điều này có thể bao gồm việc tối ưu hóa các thuật toán tìm kiếm, sử dụng cấu trúc dữ liệu hiệu quả hơn, hoặc giảm thiểu số lần truy cập cơ sở dữ liệu không cần thiết.
   * **Giảm thiểu độ trễ:** Phân tích các phần của hệ thống có thể gây ra độ trễ, như việc xử lý lớn dữ liệu, và tìm cách cải thiện tốc độ xử lý. Ví dụ, sử dụng đa luồng để thực hiện các tác vụ đồng thời khi cần thiết.
   * **Tối ưu hóa hiệu suất xử lý dữ liệu:** Sử dụng các công cụ và kỹ thuật để theo dõi hiệu suất của hệ thống trong thời gian thực, từ đó xác định các điểm nghẽn và cải thiện chúng. Có thể áp dụng các giải pháp caching để giảm tải cho hệ thống khi truy cập dữ liệu thường xuyên.

🡺Quy trình kiểm thử và tối ưu hóa là rất quan trọng để đảm bảo hệ thống không chỉ hoạt động đúng mà còn hiệu quả và đáp ứng nhu cầu của người dùng. Việc thực hiện kiểm thử đầy đủ và tối ưu mã nguồn sẽ giúp hệ thống hoạt động trơn tru, đáp ứng tốt với lượng người dùng lớn và tránh gặp phải các lỗi không mong muốn trong quá trình vận hành.

**3.4 Sử Dụng Chương Trình Thực Tế :**

Mở đầu của chương trình:



Hình 3.4: Giao diện khởi đầu chương trình



Hình 3.5: Giao diện chương trình

**3.4.1 Khách Hàng:**

**Bước 1: Đăng ký đối với khách chưa có tài khoản (hoặc đăng nhập nếu khách hàng đã có tài khoản) .**

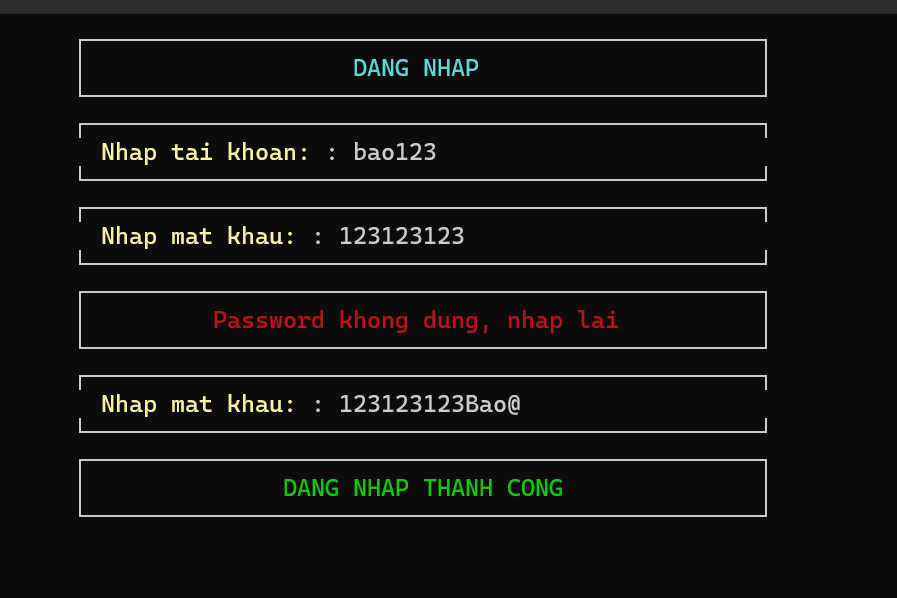
-Điền đầy đủ thông tin để được đăng ký.



Hình 3.6: Giao diện đăng ký với vai trò khách hàng

**Bước 2: Đăng nhập:**

-Điền thông tin để đăng nhập.



Hình 3.7: Giao diện đăng nhập thành công

**-**Sau khi đăng nhập thành công với vai trò khách hàng.

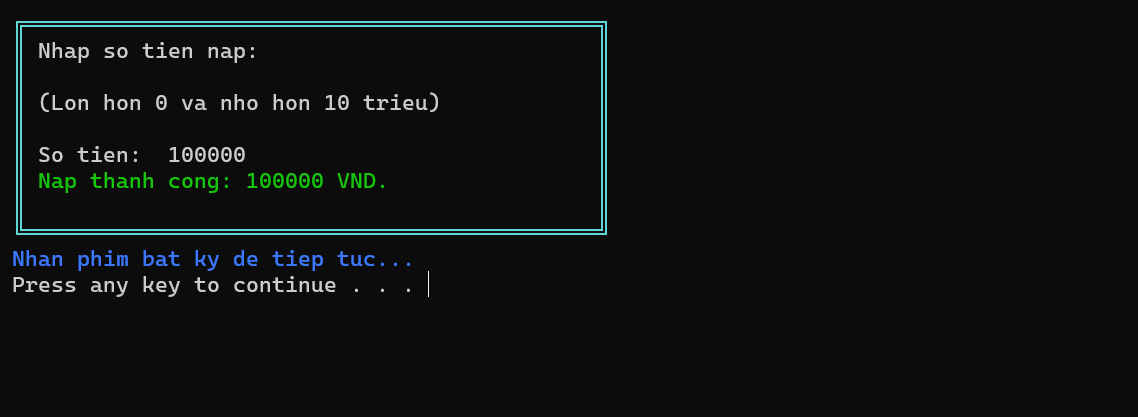


Hình 3.8: Giao diện chức năng của khách hàng

-Bạn cần nạp tiền để mua sản phẩm vì lúc này tài khoản của bạn đang có số dư là (0 $)

**Bước 3: Nạp tiền vào tài khoản**

-Nhập số tiền bạn muốn để nạp

****

Hình 3.9: Giao diện chức năng của nạp tiền

**-**Sau khi nạp xong thì tiền đã thêm vào tài khoản



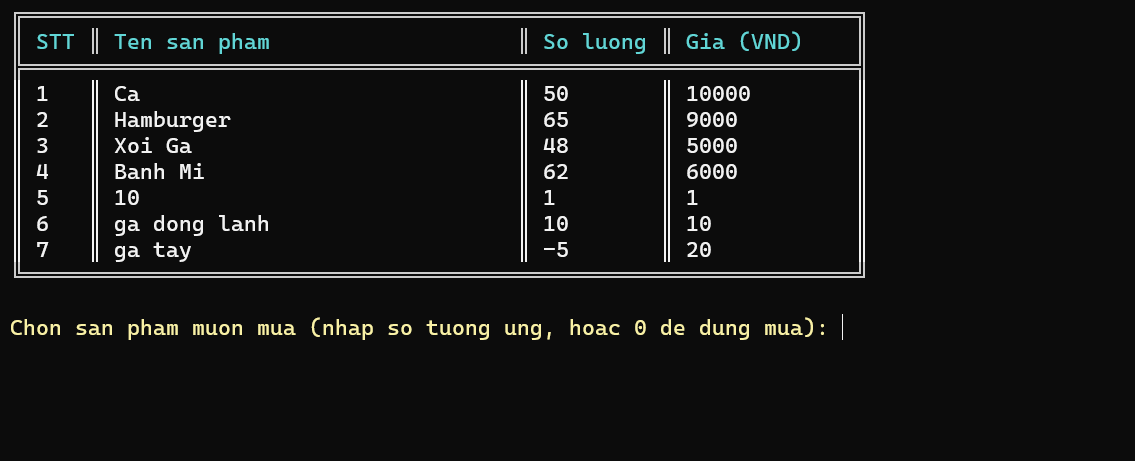
Hình 3.9: Thông tin của khách hàng

**Bước 4: Chọn quầy sản phẩm để mua sắm:**

****

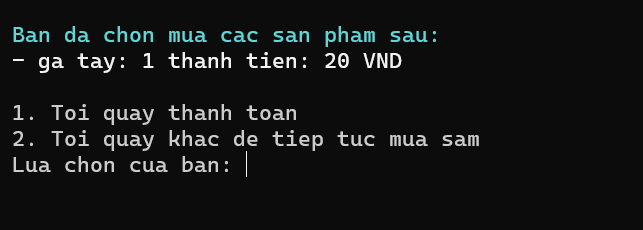
Hình 3.10: Giao diện chức năng mua hàng

-Sau đó chọn loại sản phẩm bạn muốn mua



Hình 3.11: Sản phẩm sẽ in ra theo loại bạn chọn

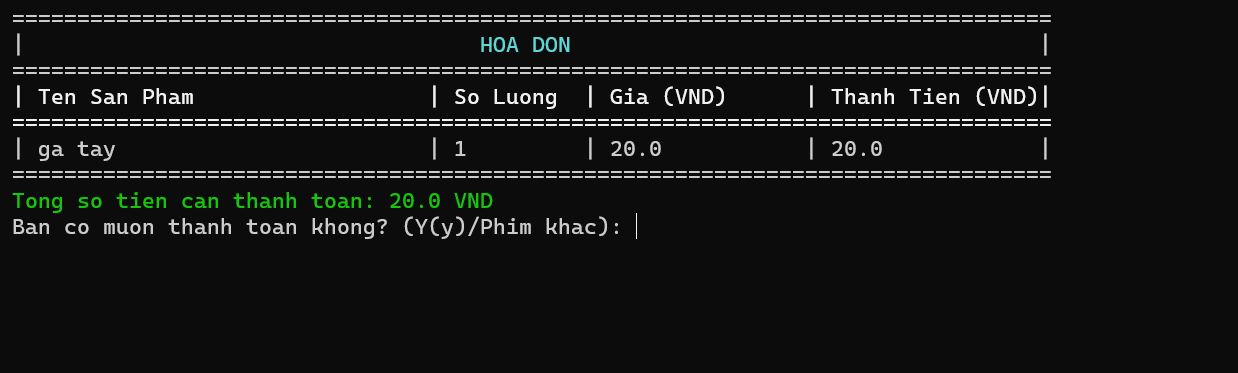
-Nhập không để dừng lại



Hình 3.12: Sau khi đã dừng chọn sản phẩm của 1 loại

-Sau đó bạn dừng thì sẽ thêm tất cả sản phẩm bạn vừa lựa vào giỏ hàng tạm và lúc này lựa chọn tới quầy thanh toán hoặc là tiếp tục mua tiếp thì các bước tương tự trên (nếu thanh toán thì tới bước tiếp theo)

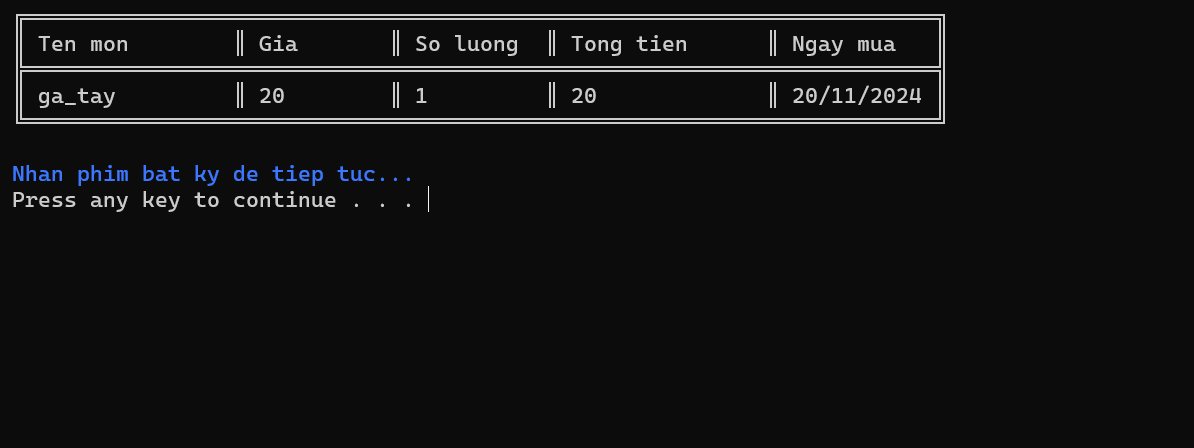
**Bước 5: Chọn quầy thanh toán:**

****

Hình 3.13: Giao diện chức năng thanh toán của khách hàng

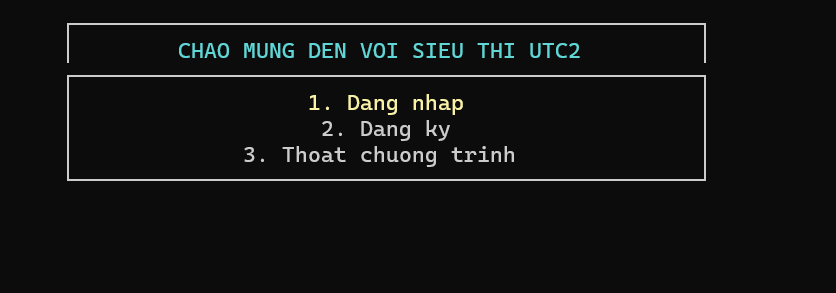
-Bạn sẽ thanh toán tại đây và sau khi thanh toán thì số tiền trong tài khoảng sẽ trừ ra và bạn có muốn tiếp tục chương trình hay không

**Bước 6: Bạn có thể kiểm tra lịch sử mua hàng hoặc đăng xuất:**

****

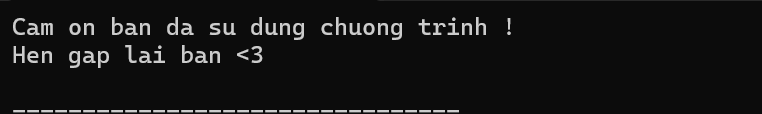
Hình 3.14: Giao diện lịch sử mua hàng

-Hoặc đăng xuất:



Hình 3.15: Giao diện sau khi sử dụng chức năng đăng xuất của khách hàng

-Thoát chương trình:



Hình 3.16: Giao diện dừng chương trình

**3.4.2 Nhân viên:**

**Bước 1: Đăng nhâp:**

****

Hình 3.17: Giao diện đăng nhập của nhân viên

**Bước 2: Bạn có thể sử dụng tất cả các chức năng trong vai trò nhân viên**

****

Hình 3.18: Giao diện chức năng của nhân viên

**3.4.3 Admin:**

**Bước 1: Đăng nhâp:**

****

Hình 3.19: Giao diện đăng nhập của quản lý

**Bước 2: Bạn có thể sử dụng tất cả các chức năng trong vai trò quản lý**

****

Hình 3.20: Giao diện chức năng của quản lý

# CHƯƠNG 4. KẾT QUẢ VÀ KIẾN NGHỊ

# 4.1 Kết quả đạt được:

**Hoàn thành chức năng chính của chương trình:** Tất cả các chức năng chính của hệ thống đã được triển khai đầy đủ, bao gồm quản lý sản phẩm, quản lý khách hàng và lập hóa đơn. Hệ thống cho phép người dùng thực hiện các tác vụ một cách hiệu quả và nhanh chóng, đáp ứng nhu cầu của người dùng. Nhờ việc áp dụng các nguyên lý OOP, tôi đã có thể thiết kế các lớp (class) và đối tượng (object) một cách hợp lý, giúp cho mã nguồn trở nên dễ hiểu và dễ bảo trì hơn.

**Đảm bảo trải nghiệm người dùng (UX/UI):** Giao diện người dùng đã được thiết kế thân thiện và dễ sử dụng. Các chức năng được bố trí hợp lý, giúp người dùng dễ dàng tìm kiếm và tương tác với các tính năng của hệ thống, mang lại trải nghiệm người dùng tốt

**Kiểm thử và bảo mật:** Hệ thống đã trải qua quá trình kiểm thử nghiêm ngặt, đảm bảo rằng tất cả các chức năng hoạt động chính xác mà không có lỗi. Các biện pháp bảo mật đã được triển khai để bảo vệ thông tin người dùng và dữ liệu giao dịch, nhằm tăng cường tính an toàn cho hệ thống. Qua việc sử dụng các phương thức kế thừa và đa hình trong OOP, tôi đã tạo ra các lớp kiểm thử (test class) một cách có hệ thống, giúp phát hiện và xử lý lỗi một cách nhanh chóng và hiệu quả.

**Học hỏi từ OOP:** Trong quá trình phát triển hệ thống, tôi đã học được nhiều điều bổ ích từ lập trình hướng đối tượng. Tôi hiểu rõ hơn về cách thiết kế lớp và đối tượng, cách sử dụng kế thừa để tái sử dụng mã, và cách áp dụng tính đóng gói để bảo vệ dữ liệu. Những kiến thức này không chỉ giúp tôi hoàn thành dự án mà còn chuẩn bị cho tôi những kỹ năng cần thiết cho các dự án trong tương lai.

# 4.2 Đánh giá và nhận xét

* **Điểm mạnh:**
  + Hệ thống được xây dựng dựa trên mô hình hướng đối tượng, giúp mã nguồn dễ bảo trì và mở rộng.
  + Tính năng quản lý sản phẩm và khách hàng hoạt động hiệu quả, hỗ trợ người quản lý theo dõi và điều hành công việc tốt.
  + Giao diện người dùng thân thiện, cung cấp trải nghiệm tốt cho người dùng cuối, giúp họ dễ dàng thực hiện các tác vụ.
* **Điểm hạn chế:**
  + Một số chức năng như báo cáo doanh số có thể cần bổ sung thêm nhiều tùy chọn để người dùng linh hoạt hơn trong việc xuất báo cáo.
  + Vẫn còn một số vấn đề nhỏ về hiệu suất khi xử lý một lượng lớn dữ liệu, ảnh hưởng đến tốc độ phản hồi của hệ thống.

# 4.3 Kiến nghị

* **Cải thiện và mở rộng các tính năng:**
  + Nên bổ sung các tính năng như quản lý khuyến mãi và theo dõi lịch sử giao dịch của khách hàng để tăng cường khả năng phục vụ và hỗ trợ người dùng tốt hơn.
* **Tăng cường bảo mật và tối ưu hiệu năng:**
  + Cần áp dụng các biện pháp bảo mật tiên tiến hơn, như mã hóa dữ liệu và xác thực đa yếu tố, để bảo vệ thông tin người dùng tốt hơn.
  + Tối ưu hóa hiệu suất xử lý dữ liệu, đặc biệt trong các tác vụ liên quan đến báo cáo và tìm kiếm sản phẩm, nhằm cải thiện tốc độ hoạt động của hệ thống.
* **Định hướng phát triển lâu dài:**
  + Xây dựng kế hoạch phát triển bền vững để cập nhật công nghệ mới và cải tiến hệ thống theo nhu cầu thị trường.
  + Cần thường xuyên thu thập phản hồi từ người dùng để cải thiện sản phẩm, đảm bảo đáp ứng tốt nhất nhu cầu sử dụng.

**Tóm lại:** Chương này tóm tắt những kết quả đã đạt được trong quá trình phát triển hệ thống quản lý siêu thị, cùng với các đánh giá và kiến nghị nhằm nâng cao hiệu quả hoạt động của hệ thống trong tương lai. Những đề xuất này không chỉ giúp cải thiện hệ thống mà còn đảm bảo đáp ứng nhu cầu ngày càng cao của người dùng.

# TÀI LIỆU THAM KHẢO

[1]. Nguyễn Văn A, Lập trình hướng đối tượng trong C++, NXB Giáo dục, 2020.

[2]. Trần Thị B, Phạm Văn C, Phát triển phần mềm quản lý siêu thị bằng OOP, NXB Khoa học và Kỹ thuật, 2018.

[3]. Lê Huy Dũng, Thực hành lập trình OOP với ứng dụng thực tế, NXB ĐHQG, 2021.

[4]. Phạm Quốc Hưng, Quản lý bán lẻ và công nghệ trong siêu thị, NXB Kinh tế, 2019.

[5]. Lập trình Việt, "Ứng dụng OOP trong quản lý bán lẻ," https://www.laptrinhviet.com/oop-quan-ly-sieu-thi