

**ĐẠI HỌC ĐÀ NẴNG  
ĐẠI HỌC BÁCH KHOA  
KHOA CÔNG NGHỆ THÔNG TIN**



**PBL2: DỰ ÁN CƠ SỞ LẬP TRÌNH**

**Đề tài: Hệ thống mua bán và quản lý thiết bị điện tử**

**SINH VIÊN THỰC HIỆN:**

<b>Lương Văn Võ</b>	<b>LỚP: 23T_DT1</b>	<b>NHÓM: 23.Nh10B</b>
<b>Nguyễn Hiếu</b>	<b>LỚP: 23T_DT1</b>	<b>NHÓM: 23.Nh10B</b>

**GIẢNG VIÊN HƯỚNG DẪN:**

**TS. Đặng Thiên Bình**

### LỜI MỞ ĐẦU

Trong những năm gần đây, công nghệ thông tin không ngừng phát triển một cách mạnh mẽ và hiện đại. Sự ra đời của công nghệ thông tin làm phong phú bộ mặt xã hội, đời sống con người được nâng cao rõ rệt, đóng góp to lớn cho sự phát triển của nhân loại. Với xu thế toàn cầu hoá nền kinh tế thế giới, đặc biệt là nhu cầu trao đổi hàng hoá của con người ngày càng tăng cả về số lượng và chất lượng, nhu cầu sử dụng Internet ngày càng nhiều và các hình thức kinh doanh trên mạng ngày càng đa dạng và phong phú rất được nhiều người sử dụng và ưa chuộng. Vì vậy, nhiều cửa hàng đã áp dụng phương pháp bán hàng qua mạng để khách hàng dễ dàng tìm kiếm thông tin và mua sản phẩm mà không cần đến tận cửa hàng, do đó có thể giảm được thời gian và chi phí đi lại cho khách hàng.

Việc xây dựng các ứng dụng ngày càng phổ biến và nhu cầu mua sắm của người dân ngày càng tăng cao. Để đáp ứng được nhu cầu tất yếu đó, chúng em đã xây dựng ứng dụng bán hàng trực tuyến. Ở ứng dụng này sẽ cập nhật đầy đủ thông tin mới nhất các loại thiết bị.

Mặc dù đã có nhiều cố gắng nhưng do thời gian tìm hiểu và trình độ có hạn nên không tránh khỏi những sai lầm và thiếu sót. Nên em rất mong đóng góp và chỉ bảo của quý thầy cô và các bạn.

Cuối cùng chúng em chân thành cảm ơn sự hướng dẫn và chỉ bảo tận tình của giảng viên hướng dẫn Đặng Thiên Bình và toàn thể thầy cô trong khoa công nghệ thông tin.

## MỤC LỤC

LỜI MỞ ĐẦU .....	1
MỤC LỤC.....	2
DANH MỤC HÌNH VẼ.....	4
1. GIỚI THIỆU ĐỀ TÀI.....	6
2. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG.....	7
2.1. Chức năng quản lý khách hàng, sản phẩm và các đơn hàng	7
2.2. Thao tác với dữ liệu trong cơ sở dữ liệu	7
2.3. Các phương thức chính của khách hàng	11
2.3.1. Thêm khách hàng (đăng ký).....	11
2.3.2. Quên và đổi mật khẩu đăng nhập .....	12
2.3.3. Xem lại lịch sử chi tiết đơn đặt hàng.....	12
2.3.4. Thêm sản phẩm vào giỏ hàng.....	13
2.3.5. Tăng số lượng sản phẩm trong giỏ hàng .....	13
2.3.6. Giảm số lượng sản phẩm trong giỏ hàng .....	14
2.3.7. Xóa sản phẩm ra khỏi giỏ hàng.....	14
2.3.8. Đặt đơn hàng .....	14
2.4. Các phương thức chính của quản trị viên (admin)	15
2.4.1. Sắp xếp theo tên khách hàng .....	15
2.4.2. Sắp xếp theo tên sản phẩm .....	16
2.4.3. Sắp xếp theo giá của sản phẩm.....	17
2.4.4. Tìm kiếm khách hàng .....	18
2.4.5. Tìm kiếm sản phẩm .....	19
2.4.6. Thêm sản phẩm mới .....	20
2.4.7. Chỉnh sửa thông tin sản phẩm .....	21
2.4.8. Xóa sản phẩm .....	22
2.4.9. Tạo mã giảm giá.....	22
2.4.10. Thống kê chi tiết.....	23
2.4.11. Thống kê doanh thu.....	23
3. THIẾT KẾ CẤU TRÚC DỮ LIỆU .....	24

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

3.1. Phát biểu bài toán	24
3.1.1. Mô tả đầu vào (Input): Đầu vào của chương trình gồm các file được lưu trong các thư mục.....	24
3.1.2. Yêu cầu đầu ra (Output):.....	26
3.2. Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống	27
4. ỨNG DỤNG HƯỚNG ĐỐI TƯỢNG VÀO BÀI TOÁN.....	28
4.1. Phân tích các lớp của hướng đối tượng	28
4.2. Mối liên kết phụ thuộc lẫn nhau giữa các thuộc tính của các lớp	37
4.3. Kết quả	38
5. CHƯƠNG TRÌNH VÀ KẾT QUẢ .....	38
5.1. Tổ chức chương trình	38
5.2. Kết quả thực thi của chương trình	38
5.2.1. Giao diện khách hàng.....	38
5.2.2. Giao diện quản trị viên.....	46
5.3. Nhận xét và đánh giá	54
6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	55
6.1. Kết luận	55
6.1.1. Đạt được .....	55
6.1.2. Chưa đạt được .....	55
6.2. Hướng phát triển	56
TÀI LIỆU THAM KHẢO.....	57

## DANH MỤC HÌNH VẼ

Hình 3.1-1. Mô tả đầu vào .....	24
Hình 3.1-2. Mô tả đầu vào thông tin khách hàng.....	24
Hình 3.1-3. Mô tả đầu vào thông tin sản phẩm.....	25
Hình 3.1-4. Mô tả đầu vào thông số kỹ thuật của sản phẩm.....	25
Hình 3.1-5. Mô tả đầu vào đường dẫn ảnh.....	25
Hình 3.1-6. Mô tả đầu vào mã giảm giá.....	26
Hình 3.1-7. Mô tả đầu vào các file giỏ hàng của khách hàng.....	26
Hình 4.1-1. Thuộc tính và phương thức lớp Customer .....	28
Hình 4.1-2. Thuộc tính và phương thức lớp Products.....	30
Hình 4.1-3. Thuộc tính và phương thức class Admin .....	35
Hình 4.1-4. Thuộc tính và phương thức class Discount.....	33
Hình 4.1-5. Thuộc tính và phương thức class History .....	34
Hình 4.1-6. Thuộc tính và phương thức class FileManager.....	36
Hình 4.1-7. Thuộc tính và phương thức class Order.....	32
Hình 4.2-1. Mô hình ER minh họa mối quan hệ giữa các thực thể .....	37
Hình 5.2-1. Giao diện đăng nhập .....	39
Hình 5.2-2. Giao diện đăng ký .....	39
Hình 5.2-3. Giao diện nhập email .....	40
Hình 5.2-4. Giao diện nhập mã OTP .....	40
Hình 5.2-5. Giao diện nhập mật khẩu mới.....	41
Hình 5.2-6. Giao diện chính của ứng dụng .....	42
Hình 5.2-7. Giao diện chi tiết của sản phẩm.....	42
Hình 5.2-8. Giao diện thông số kỹ thuật của sản phẩm .....	43
Hình 5.2-9. Giao diện giỏ hàng.....	44
Hình 5.2-10. Giao diện thanh toán .....	45
Hình 5.2-11. Giao diện xem lịch sử đơn hàng .....	45
Hình 5.2-12. Giao diện thông tin tài khoản.....	46
Hình 5.2-13. Giao diện chính.....	47
Hình 5.2-14. Giao diện danh sách tất cả khách hàng.....	47

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

Hình 5.2-15. Giao diện tìm kiếm tên khách hàng .....	48
Hình 5.2-16. Giao diện sắp xếp tên khách hàng .....	48
Hình 5.2-17. Giao diện danh sách tất cả sản phẩm .....	49
Hình 5.2-18. Giao diện tìm kiếm tên sản phẩm .....	49
Hình 5.2-19. Giao diện sắp xếp sản phẩm theo tên sản phẩm .....	50
Hình 5.2-20. Giao diện sắp xếp sản phẩm theo giá sản phẩm .....	50
Hình 5.2-21. Giao diện chỉnh sửa thông tin sản phẩm.....	51
Hình 5.2-22. Giao diện xem doanh thu .....	51
Hình 5.2-23. Giao diện tạo mã giảm giá .....	52
Hình 5.2-24. Giao diện thêm sản phẩm .....	53
Hình 5.2-25. Giao diện thống kê doanh thu khi không có đơn hàng trong ngày .....	53
Hình 5.2-26. Giao diện thống kê doanh thu khi có đơn hàng trong ngày .....	54

### 1. GIỚI THIỆU ĐỀ TÀI

Đề tài “Hệ thống mua bán và quản lý thiết bị điện tử” nhằm hướng đến hai mục tiêu chính.

- Đối với khách hàng: Cung cấp nền tảng giúp khách hàng dễ dàng tiếp cận, tìm hiểu thông tin chi tiết về các sản phẩm là các thiết bị điện tử, và thực hiện việc mua sắm trực tuyến. Điều này giúp khách hàng tiết kiệm thời gian và chi phí đi lại, mang lại trải nghiệm tiện lợi và hiệu quả hơn.
- Đối với người bán: Hỗ trợ người bán tối ưu hóa quy trình kinh doanh bằng cách đưa sản phẩm lên nền tảng trực tuyến, giúp mở rộng thị trường, tăng cơ hội tiếp cận khách hàng, và quản lý bán hàng hiệu quả hơn.

Một mục tiêu quan trọng khác của hệ thống là xây dựng cơ sở dữ liệu có cấu trúc rõ ràng và dễ dàng truy cập. Điều này đảm bảo thông tin về các thiết bị điện tử, khách hàng, đơn hàng và các hoạt động giao dịch có thể được tìm kiếm, quản lý và xử lý nhanh chóng bởi cả khách hàng lẫn người quản lý.

Hệ thống được xây dựng để quản lý thông tin liên quan đến khách hàng, sản phẩm và các đơn hàng. Các thông tin này bao gồm: Khách hàng (Mã khách hàng, họ và tên, email, số điện thoại khách hàng, tên đăng nhập, mật khẩu), Sản phẩm (Mã sản phẩm, tên sản phẩm, thể loại, giá tiền, nhà sản xuất, hệ điều hành, thông số kỹ thuật, số lượng sản phẩm), Đơn hàng (Mã đơn hàng, sản phẩm trong giỏ hàng, số lượng sản phẩm trong giỏ, mã của khách hàng đặt đơn hàng, địa chỉ của đơn hàng). Ngoài ra còn có các thông tin về lịch sử đơn hàng (ngày đặt hàng, mã đơn hàng, địa chỉ, thành tiền của đơn hàng, chi tiết các sản phẩm trong đơn hàng), Mã giảm giá (Tạo và cấp mã giảm giá cho khách hàng). Bên cạnh đó hệ thống còn quản lý các thông tin liên quan đến Admin (Người quản lý) với các chức năng (Thêm và xóa sản phẩm, quản lý thông tin khách hàng, xem doanh thu tổng hợp, cấp mã giảm giá cho khách hàng, xem thống kê doanh thu của từng ngày) . Hệ thống còn giúp người quản lý trong việc tìm kiếm và lọc nhanh chóng thông tin các sản phẩm, khách hàng, thêm sửa xóa các dữ liệu khi cần thiết. Hệ thống không chỉ là công cụ hỗ trợ mua bán và quản lý thiết bị điện tử, mà còn là cầu nối giữa khách hàng và người bán, tạo nên một môi trường giao dịch hiện đại, tiện ích và đáng tin cậy, góp phần thúc đẩy sự phát triển bền vững trong lĩnh vực thương mại điện tử.

### 2. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG

Trong phần này, chúng ta sẽ phân tích chi tiết các chức năng chính của hệ thống, bao gồm quản lý khách hàng, thiết bị điện tử và các đơn hàng. Các chức năng này bao gồm những thao tác cơ bản như thêm, tìm kiếm, lọc, xóa, hiển thị và thống kê thông tin về các đối tượng trong hệ thống.

Các chức năng được thiết kế với mục tiêu tối ưu hóa trải nghiệm người dùng, đảm bảo thao tác dễ dàng và hiệu quả. Cụ thể, hệ thống cung cấp các tính năng:

#### 2.1. Chức năng quản lý khách hàng, sản phẩm và các đơn hàng

- Chức năng quản lý khách bao gồm thêm khách hàng mới, tìm kiếm theo tên khách hàng, hiển thị danh sách khách hàng.
- Chức năng quản lý sản phẩm bao gồm thêm sản phẩm mới, xóa sản phẩm, chỉnh sửa thông tin sản phẩm, hiển thị danh sách sản phẩm.
- Chức năng quản lý các đơn hàng bao gồm thêm và xóa sản phẩm trong giỏ hàng, hiển thị nội dung giỏ hàng, tạo đơn hàng (đặt hàng).
- Khách hàng có thể dùng các chức năng thêm vào và xóa sản phẩm ra khỏi giỏ hàng, tăng giảm số lượng sản phẩm trong giỏ hàng, đặt đơn hàng, cập nhật tài khoản (đổi mật khẩu đăng nhập), quên mật khẩu đăng nhập, xem lịch sử mua hàng.
- Quản trị viên(admin) có thể dùng các chức năng như quản lý khách hàng (hiển thị danh sách, tìm kiếm khách hàng), quản lý sản phẩm (thêm, xóa, chỉnh sửa thông tin sản phẩm), quản lý các đơn hàng (xem lại lịch sử đơn đặt hàng, thống kê doanh thu, tìm kiếm đơn hàng), cấp phát mã giảm giá.

#### 2.2. Thao tác với dữ liệu trong cơ sở dữ liệu

Thao tác dữ liệu trong cơ sở dữ liệu là một trong những chức năng trọng tâm, giúp hệ thống xử lý và quản lý thông tin một cách đơn giản, hiệu quả. Đây là cầu nối giữa dữ liệu và hệ thống xử lý, đảm bảo dữ liệu được đọc, ghi, cập nhật và xóa một cách chính xác và nhanh chóng.

Chức năng này bao gồm các nhiệm vụ chính như:

- Đọc dữ liệu: lấy thông tin từ file, đưa vào hệ thống để xử lý và hiển thị
- Ghi dữ liệu: lưu trữ thông tin từ hệ thống vào lại file để lưu trữ.
- Cập nhật dữ liệu: sửa đổi thông tin hiện có trong file để đảm bảo dữ liệu được cập nhật.
- Xóa dữ liệu: loại bỏ thông tin không cần thiết để giảm thiểu sai sót về dữ liệu, tối ưu hóa bộ nhớ và đảm bảo hiệu suất hệ thống.



## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

Cụ thể trong hệ thống này, các thao tác này được đóng gói trong lớp quản lý dữ liệu là class FileManager, sử dụng các phương thức đọc/ghi từ file.

### **Phương thức đọc dữ liệu từ file và lưu vào hệ thống**

Trường hợp 1: Đọc và xử lý dữ liệu theo dòng và phân tích ký tự phân cách thành nhiều trường dữ liệu

Bước 1: Chuẩn bị và kiểm tra file

- Mở file cần đọc
- Kiểm tra xem file có mở thành công hay không

Bước 2: Đếm số lượng dữ liệu

- Mục tiêu: Xác định số lượng dòng dữ liệu để cấp phát bộ nhớ động phù hợp
- Cách thực hiện:
  - Đọc từng dòng dữ liệu bằng hàm getline
  - Tăng dần biến đếm số lượng dòng để biết tổng số dòng trong file

Bước 3: Đọc dữ liệu và lưu vào mảng động

- Quay lại đầu file để chuẩn bị đọc lại
- Cấp phát bộ nhớ cho mảng động với số lượng dòng đã đọc ở bước 2
- Đọc từng dòng dữ liệu, phân tích dữ liệu theo ký tự phân cách '|'
- Xử lý dữ liệu theo yêu cầu (chuyển đổi kiểu dữ liệu, lọc,...)
- Lưu dữ liệu đã xử lý vào mảng động

Bước 4: kết thúc

- Trả về mảng động chứa dữ liệu đã xử lý.
- Xử lý ngoại lệ nếu có lỗi trong quá trình đọc hoặc lưu dữ liệu, đảm bảo không làm gián đoạn chương trình

Trường hợp 2: Đọc và xử lý dữ liệu theo dòng

Bước 1: Chuẩn bị và kiểm tra file

- Mở file cần đọc
- Kiểm tra xem file có mở thành công hay không

Bước 2: Đếm số lượng dữ liệu

- Mục tiêu: Xác định số lượng dòng để cấp phát bộ nhớ động phù hợp.
- Cách thực hiện:
  - Đọc từng dòng dữ liệu bằng hàm getline.
  - Tăng dần biến đếm số lượng dòng để biết tổng số dòng trong file.

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

### Bước 3: Đọc và lưu dữ liệu vào mảng động

- Quay lại đầu file để chuẩn bị đọc lại dữ liệu.
- Cấp phát bộ nhớ cho mảng động theo số lượng dòng đã đếm ở bước 2.
- Đọc từng dòng dữ liệu nguyên vẹn, không tách ra từng phần, và lưu trữ vào mảng động.
- Xử lý dữ liệu theo yêu cầu (chuyển đổi kiểu dữ liệu, ...)
- Lưu dữ liệu đã xử lý vào mảng động

### Bước 4: Kết thúc

- Trả về mảng động chứa dữ liệu đã xử lý.
- Xử lý ngoại lệ trong trường hợp có lỗi xảy ra trong quá trình đọc hoặc lưu dữ liệu, đảm bảo không làm gián đoạn chương trình

### Trường hợp 3: Đọc và xử lý dữ liệu từ file với cặp dấu ‘\*’

#### Bước 1: Chuẩn bị và kiểm tra file

- Mở file cần đọc
- Kiểm tra xem file có mở thành công hay không

#### Bước 2: Đếm số lượng dữ liệu

- Mục tiêu: Xác định số lượng để cấp phát bộ nhớ động cho phù hợp
- Cách thực hiện:
  - Đọc từng dòng dữ liệu bằng hàm getline
  - Khi gặp dòng bắt đầu và kết thúc bằng dấu \*, tăng biến đếm lên 1

#### Bước 3: Đọc và lưu dữ liệu vào mảng động

- Quay lại đầu file để chuẩn bị đọc lại dữ liệu
- Cấp phát bộ nhớ cho mảng động theo số lượng dòng đã đếm ở bước 2
- Phân tích dữ liệu:
  - Mỗi mục dữ liệu được tạo bởi cặp dấu \*, với:
    - Dòng đầu chứa ID
    - Dòng thứ hai đến dấu \* chứa thông tin specifications
  - Xử lý dữ liệu theo yêu cầu (chuyển đổi kiểu dữ liệu, lọc thông tin, ...)
  - Lưu dữ liệu đã xử lý vào mảng động

#### Bước 4: Kết thúc

- Trả về mảng động chứa dữ liệu đã được xử lý.
- Xử lý ngoại lệ trong trường hợp có lỗi xảy ra trong quá trình đọc hoặc lưu dữ liệu, đảm bảo không làm gián đoạn chương trình.

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

### Phương thức ghi dữ liệu từ hệ thống vào file

#### Trường hợp 1:

Bước 1: Chuẩn bị file

- Xác định tên và đường dẫn của file cần ghi dữ liệu
- Kiểm tra xem file có mở thành công hay không

Bước 2: Lưu dữ liệu vào file

- Chuyển đổi dữ liệu thành định dạng phù hợp (nếu cần)
- Lưu vào theo từng dòng các thông tin và sử dụng dấu “|” để tách thành các trường dữ liệu

#### Trường hợp 2:

Bước 1: Chuẩn bị file

- Xác định tên và đường dẫn của file cần ghi dữ liệu
- Kiểm tra xem file có mở thành công hay không

Bước 2: Lưu dữ liệu

- Lưu dữ liệu thành từng dòng và nguyên một dòng

#### Trường hợp 3: Lưu dữ liệu vào file dựa trên cặp dấu \*

Bước 1: Chuẩn bị file

- Xác định file cần mở
- Kiểm tra xem file có mở thành công hay không. Nếu không, thông báo lỗi và kết thúc.

Bước 2: Lưu dữ liệu

- Đọc dữ liệu từng dòng của file.
- Khi gặp dấu \* đầu tiên, bắt đầu một sản phẩm.
- Dòng 1 sau dấu \*: Xác định là ID
- Các dòng tiếp theo đến dấu \* tiếp theo: Lưu dưới dạng specifications.
- Khi gặp dấu \* kết thúc, phần dữ liệu nằm trong cặp dấu \* là thông tin của 1 sản phẩm.

### Phương thức xoá dữ liệu: xoá một dòng trong file txt

Bước 1: Chuẩn bị file

- Mở file cần xoá
- Kiểm tra việc mở file: Xác nhận rằng file đã được mở thành công. Nếu không, tiến trình sẽ kết thúc ngay lập tức để tránh lỗi.

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

### Bước 2: Đọc file và lưu vào mảng

- Đọc từng dòng: Sử dụng vòng lặp để đọc từng dòng dữ liệu trong file và lưu vào mảng động
- Giới hạn tối đa là MAX\_LINES dòng (20 dòng)

### Bước 3: Xoá dòng cần xoá

- Xác định dòng cần xoá: Định vị dòng cần xoá thông qua biến lineToDelete
- Dời các dòng phía sau lên: Từ vị trí lineToDelete, dịch chuyển các dòng phía sau lên một vị trí trong mảng để ghi đè dòng cần xoá
- Giảm tổng số dòng: Giảm biến totalLines đi 1 để cập nhật lại số lượng dòng hiện có trong mảng

### Bước 4: Ghi lại mảng vào file

- Mở file ở chế độ ghi đè: Sử dụng chế độ ios::trunc để mở file, đảm bảo rằng nội dung cũ sẽ bị xoá và thay thế bằng nội dung mới
- Ghi từng dòng trong mảng vào file

## 2.3. Các phương thức chính của khách hàng

### 2.3.1. Thêm khách hàng (đăng ký)

Chức năng này cho phép tạo tài khoản mới để đăng nhập. Khách hàng phải nhập đầy đủ các thông tin như họ và tên, email, số điện thoại, tên đăng nhập, mật khẩu đăng nhập. Hệ thống sẽ kiểm tra tính duy nhất của tên đăng nhập. Sau đó hệ thống sẽ tự động cấp cho tài khoản một customerID và lưu trữ những thông tin này vào cơ sở dữ liệu.

Cách thức hoạt động:

- Khách hàng nhập đầy đủ các thông tin cá nhân như họ, tên đệm, tên, email, số điện thoại, tên đăng nhập và mật khẩu.
- Hệ thống sẽ kiểm tra tính duy nhất của tên đăng nhập.
- Nếu tên đăng nhập bị trùng lặp thì yêu cầu nhập lại tên đăng nhập.
- Ngược lại, nếu tên đăng nhập không bị trùng lặp, hệ thống sẽ
  - o Lưu thông tin vào cơ sở dữ liệu.
  - o Tự động tạo một file của riêng khách hàng đó để lưu thông tin giỏ hàng cho khách hàng.
- Kết thúc.

### 2.3.2. Quên và đổi mật khẩu đăng nhập

Chức năng này cho phép thay đổi mật khẩu của tài khoản đăng nhập. Khách hàng nhập địa chỉ email đã đăng ký tài khoản, hệ thống kiểm tra email trong cơ sở dữ liệu. Nếu email tồn tại, hệ thống gửi mã OTP (6 chữ số) tới email đó. Khách hàng kiểm tra và nhập mã OTP để xác thực, nếu hợp lệ, hệ thống yêu cầu đặt lại mật khẩu mới và nhập lại một lần nữa để xác nhận mật khẩu mới, nếu cả hai trùng khớp thì mật khẩu được đổi thành công. Sau khi cập nhật thành công, mật khẩu mới sẽ được lưu lại vào trong cơ sở dữ liệu.

Cách thức hoạt động:

- Khách hàng nhập email.
- Hệ thống kiểm tra email có tồn tại trong cơ sở dữ liệu không
  - o Nếu có: Hệ thống sẽ gửi mã OTP đến email của khách hàng
  - o Nếu không: Hệ thống thông báo lỗi và yêu cầu nhập lại
- Khách hàng nhập mã OTP đã nhận từ email.
- Hệ thống kiểm tra mã OTP
  - o Nếu đúng: Hệ thống cho phép khách hàng nhập mật khẩu mới
  - o Nếu sai: Hệ thống thông báo lỗi và yêu cầu nhập lại
- Khách hàng nhập mật khẩu mới
- Hệ thống cập nhật mật khẩu mới vào cơ sở dữ liệu.
- Hệ thống quay về màn hình đăng nhập và kết thúc.

### 2.3.3. Xem lại lịch sử chi tiết đơn đặt hàng

Chức năng này cho phép xem lại lịch sử chi tiết đơn hàng đã đặt. Khách hàng nhập mã đơn hàng, sau đó hệ thống kiểm tra cơ sở dữ liệu để lấy thông tin chi tiết của đơn đặt hàng và hiển thị ra màn hình dưới dạng danh sách các sản phẩm trong đơn hàng đó (nếu có). Danh sách bao gồm các thông tin như: mã sản phẩm, tên sản phẩm, thể loại, nhà sản xuất, hệ điều hành, giá tiền, số lượng.

Phân tích các hàm:

- Hàm *getItem()*: truy cập nội dung của đơn hàng
- Hàm *tach(const string& items, int& index)*: phân tách dữ liệu từng dòng được ngăn cách bởi dấu “|” thành các trường dữ liệu gồm mã sản phẩm, tên sản phẩm, thể loại, nhà sản xuất, hệ điều hành, giá tiền, số lượng

Cách thức hoạt động:

- Hệ thống thao tác với cơ sở dữ liệu để lấy thông tin lịch sử các đơn đặt hàng của khách hàng lưu vào mảng động

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

- Hệ thống kiểm tra mã đơn hàng nhập vào có trong mảng động không
  - o Nếu có: Lấy chỉ số count của mảng động
  - o Nếu không: giải phóng bộ nhớ và kết thúc
- Sử dụng hàm *tach(history[count].getItem(), count)*
- Hiển thị ra màn hình gồm các thông tin: mã sản phẩm, tên sản phẩm, thể loại, nhà sản xuất, hệ điều hành, giá tiền, số lượng

### 2.3.4. Thêm sản phẩm vào giỏ hàng

Chức năng này cho phép lựa chọn và thêm các sản phẩm mà họ muốn mua và giỏ hàng. Khi khách hàng chọn một sản phẩm và thêm vào giỏ hàng, hệ thống sẽ tự động lưu trữ thông tin của sản phẩm vào giỏ hàng (mã sản phẩm, tên sản phẩm, giá tiền, số lượng) thông qua sản phẩm khách hàng đã chọn.

Cách thức hoạt động:

- Khách hàng chọn sản phẩm cần thêm vào giỏ hàng
- Hệ thống kiểm tra sản phẩm này có trong giỏ hàng chưa
  - o Nếu có: hệ thống tự động tăng số lượng sản phẩm này lên thêm 1
  - o Nếu không: hệ thống sẽ kiểm tra giỏ hàng đầy hay chưa
    - Nếu chưa: thêm sản phẩm này vào giỏ hàng thông qua mã sản phẩm
    - Nếu đầy: không thêm được sản phẩm vào giỏ hàng và kết thúc
- Hệ thống sẽ cập nhật dữ liệu trong cơ sở dữ liệu và kết thúc

### 2.3.5. Tăng số lượng sản phẩm trong giỏ hàng

Chức năng này cho phép tăng số lượng sản phẩm trong giỏ hàng lên. Khách hàng nhấn vào dấu cộng bên phải sản phẩm trong giỏ hàng thì hệ thống tự động tăng số lượng của sản phẩm đó trong giỏ hàng lên thêm 1 và cập nhật lại dữ liệu trong cơ sở dữ liệu.

Cách thức hoạt động:

- Khách hàng chọn sản phẩm cần tăng số lượng trong giỏ hàng
- Duyệt và kiểm tra mã sản phẩm trong danh sách giỏ hàng khớp với mã sản phẩm truyền vào hay không (sử dụng vòng lặp *for*)
  - o Nếu không: tiếp tục vòng lặp *for* cho đến khi kết thúc vòng lặp
  - o Nếu có: tăng số lượng sản phẩm lên thêm 1 và cập nhật dữ liệu trong cơ sở dữ liệu
- Thoát vòng lặp *for* và kết thúc

### 2.3.6. Giảm số lượng sản phẩm trong giỏ hàng

Chức năng này cho phép giảm số lượng sản phẩm trong giỏ hàng. Khách hàng nhấn vào dấu trừ bên phải sản phẩm trong giỏ hàng thì hệ thống tự động giảm số lượng sản phẩm đi 1 và cập nhật lại dữ liệu trong cơ sở dữ liệu.

Cách thức hoạt động:

- Khách hàng chọn sản phẩm cần tăng số lượng trong giỏ hàng
- Duyệt và kiểm tra mã sản phẩm trong danh sách giỏ hàng khớp với mã sản phẩm truyền vào hay không (sử dụng vòng lặp *for*)
  - o Nếu không: tiếp tục vòng lặp *for* cho đến khi kết thúc vòng lặp
  - o Nếu có: giảm số lượng sản phẩm đi 1 nếu số lượng sản phẩm bằng 0 thì xoá sản phẩm ra khỏi giỏ hàng
- Cập nhật dữ liệu trong cơ sở dữ liệu và thoát vòng lặp *for*

### 2.3.7. Xoá sản phẩm ra khỏi giỏ hàng

Chức năng này cho phép khách hàng loại bỏ các sản phẩm không muốn mua ra khỏi giỏ hàng. Khi khách hàng nhấn vào nút "Xóa" bên cạnh sản phẩm cần loại bỏ, hệ thống sẽ xóa sản phẩm đó khỏi giỏ hàng, đồng thời cập nhật trạng thái giỏ hàng để hiển thị danh sách sản phẩm còn lại.

Cách thức hoạt động:

- Khách hàng chọn sản phẩm cần xoá ra khỏi giỏ hàng
- Hệ thống sẽ duyệt qua danh sách trong giỏ hàng sử dụng vòng lặp *for*
- Kiểm tra mã sản phẩm khớp với mã sản phẩm trong giỏ hàng không
  - o Nếu có: hệ thống sẽ xoá sản phẩm ra khỏi giỏ hàng thông qua mã sản phẩm và thoát khỏi vòng lặp *for*
  - o Nếu không: tiếp tục vòng lặp
- Hệ thống sẽ cập nhật dữ liệu trong cơ sở dữ liệu và kết thúc

### 2.3.8. Đặt đơn hàng

Chức năng này giúp khách hàng đặt mua các sản phẩm. Hệ thống yêu cầu khách hàng nhập địa chỉ giao hàng (bắt buộc) và nhập mã giảm giá (không bắt buộc). Sau đó, hệ thống sẽ lưu trữ thông tin đơn hàng vào cơ sở dữ liệu.

Cách thức hoạt động:

- Đọc file để lấy mã đơn hàng sử dụng hàm *loadOrderCode*
- Kiểm tra biến *countProduct* nếu bằng 0 thì sử dụng hàm *loadProducts* để lấy danh sách sản phẩm



## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

- Tạo mảng động thanh toán *Product\* pay = new Products[maxSize]*
- Kiểm tra và xử lý từng sản phẩm
  - o Lặp qua từng sản phẩm trong giỏ hàng
  - o Đối với mỗi sản phẩm, so khớp *productID* với danh sách *products*, kiểm tra số lượng trong kho
    - Nếu đủ hàng: trừ số lượng trong kho, thêm sản phẩm vào mảng *pay*, loại sản phẩm ra khỏi giỏ hàng (*data*)
    - Nếu không đủ hàng: hiển thị thông báo hết hàng
- Xử lý thanh toán: nếu có sản phẩm trong *pay* thì
  - o Tính giá giảm dựa trên mã giảm giá (nếu có)
  - o Ghi thông tin thanh toán vào file *Pay.txt*
  - o Tăng mã đơn hàng (*orderCode*) và lưu lại vào file *orderCode.txt*
- Cập nhật dữ liệu
  - o Giải phóng bộ nhớ mảng động *pay*
  - o Cập nhật danh sách sản phẩm vào file *Products.txt*
  - o Ghi lại sản phẩm còn lại vào file cá nhân (*customerID\_order.txt*)

### 2.4. Các phương thức chính của quản trị viên (admin)

#### 2.4.1. Sắp xếp theo tên khách hàng

Chức năng này cho phép sắp xếp thông tin khách hàng theo tên của khách hàng. Hệ thống sử dụng thuật toán sắp xếp quicksort, một thuật toán sắp xếp hiệu quả và nhanh chóng. Quản trị viên có thể điều chỉnh thứ tự sắp xếp (tăng dần hoặc giảm dần) thông qua các tham số truyền vào, cụ thể:

- Ascending (sắp xếp tăng dần): điều khiển bởi giá trị true.
- Decending (sắp xếp giảm dần): điều khiển bởi giá trị false.

Cách thức hoạt động: sử dụng thuật toán quick sort

- Hàm hoán đổi vị trí của hai khách hàng *swapCustomer*
- Hàm so sánh tên khách hàng *compareNameCustomer*
  - o Chuẩn hóa tên, tên đệm, họ của khách hàng sang chữ thường bằng *transform* với chế độ *::tolower*
  - o Kiểm tra hai tên khách hàng bằng câu lệnh *if*, đầu tiên kiểm tra tên nếu tên trùng nhau thì đến kiểm tra tên đệm nếu cũng trùng nhau thì kiểm tra họ
    - Nếu trùng nhau hết cả tên, tên đệm và họ: *return 0*
    - Nếu khác nhau: thì so sánh hai tên khách hàng nếu nhỏ hơn *return -1*, còn lớn hơn *return 1*



## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

- Hàm chia mảng theo *partitionNameCustomer*
  - Chọn *pivot* là phần tử cuối cùng của mảng *customer[high]*
  - Khởi tại biến *i* với giá trị *low - 1*
  - Vòng lặp for với phần tử *j* chạy từ *low* đến *high - 1*
  - Trong mỗi vòng lặp, so sánh tên khách hàng *customer[j]* với tên của *pivot* bằng cách gọi hàm *compareNameCustomer* và gán kết quả so sánh vào biến *condition*
  - Nếu *condition = true* thì tăng *i* lên 1 (*i++*) và gọi hàm đổi vị trí *swapCustomer* để đổi vị khách hàng *i* và *j*
  - Kết thúc vòng lặp, sử dụng hàm *swapCustomer* đổi vị trí hai khách hàng (*swapCustomer(&customer[i+1], &customer[high])*)
  - Trả về giá trị *i+1* (*return i+1*)
- Hàm sắp xếp *quicksortNameCustomer*
  - So sánh *low* và *high* (*if(low < high)*)
  - Nếu *low < high* thì gọi hàm
    - *partitionNameCustomer(customer, low, high, ascending)* để gán chỉ số phân đoạn vào biến *p*
    - *quicksortNameCustomer(customer, low, p-1, ascending)*
    - *quicksortNameCustomer(customer, p+1, high, ascending)*
  - Nếu *low >= high* thì kết thúc hàm

### 2.4.2. Sắp xếp theo tên sản phẩm

Chức năng này cho phép sắp xếp thông tin khách hàng theo tên sản phẩm. Hệ thống sử dụng thuật toán sắp xếp quicksort, một thuật toán sắp xếp hiệu quả và nhanh chóng. Quản trị viên có thể điều chỉnh thứ tự sắp xếp thông qua các tham số truyền vào, cụ thể:

- Ascending (sắp xếp tăng dần): điều khiển bởi giá trị *true*.
- Decending (sắp xếp giảm dần): điều khiển bởi giá trị *false*.

Cách thức hoạt động: sử dụng thuật toán quick sort

- Hàm đổi vị trí của hai sản phẩm *swapProducts*
- Hàm so sánh hai tên sản phẩm *compareNameProduct*
  - Chuẩn hóa tên của sản phẩm sang chữ thường bằng *transform* với chế độ *::tolower*
  - Kiểm tra hai tên sản phẩm bằng câu lệnh *if*
    - Nếu trùng nhau: *return 0*
    - Nếu khác nhau: thì so sánh hai tên sản phẩm nếu nhỏ hơn *return -1*, còn lớn hơn *return 1*

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

- Hàm chia mảng theo *partitionNameProduct*
  - Chọn *pivot* là phần tử cuối cùng của mảng *products[high]*
  - Khởi tại biến *i* với giá trị *low - 1*
  - Vòng lặp *for* với phần tử *j* chạy từ *low* đến *high - 1*
  - Trong mỗi vòng lặp, so sánh tên sản phẩm *products[j]* với tên của *pivot* bằng cách gọi hàm *compareNameProduct* và gán kết quả so sánh vào biến *condition*
  - Nếu *condition = true* thì tăng *i* lên 1 (*i++*) và gọi hàm đổi vị trí *swapCustomer* để đổi vị sản phẩm *i* và *j*
  - Kết thúc vòng lặp thì sử dụng hàm *swapProducts* đổi vị trí hai sản phẩm (*swapProducts(&products[i+1], &products[high])*)
  - Trả về giá trị *i+1* (*return i+1*)
- Hàm sắp xếp *quicksortNameProduct*
  - So sánh *low* và *high* (*if(low < high)*)
  - Nếu *low < high* thì gọi hàm
    - *partitionNameProduct(products, low, high, ascending)* để gán vào biến *p*
    - *quicksortNameProduct(products, low, p-1, ascending)*
    - *quicksortNameProduct(products, p+1, high, ascending)*
  - Nếu *low >= high* thì kết thúc

### 2.4.3. Sắp xếp theo giá của sản phẩm

Chức năng này cho phép sắp xếp thông tin khách hàng theo giá tiền của sản phẩm. Hệ thống sử dụng thuật toán sắp xếp quicksort, một thuật toán sắp xếp hiệu quả và nhanh chóng. Quản trị viên có thể điều chỉnh thứ tự sắp xếp thông qua các tham số truyền vào, cụ thể:

- Ascending (sắp xếp tăng dần): điều khiển bởi giá trị *true*.
- Decending (sắp xếp giảm dần): điều khiển bởi giá trị *false*.

Cách thức hoạt động: sử dụng thuật toán quick sort

- Hàm chia theo mảng *partitionPrice*
  - Gán *pivot = products[high]* và *i = low-1*
  - Tạo vòng lặp *for* từ *j = low* đến *high-1*
  - Trong mỗi vòng lặp
    - Nếu sắp xếp tăng dần (*ascending = true*) thì kiểm tra *products[j].getPriceProduct() < pivot.getPriceProduct()*
    - Nếu sắp xếp giảm dần (*ascending = false*) thì kiểm tra *products[j].getPriceProduct() > pivot.getPriceProduct()*

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

- Nếu điều kiện kiểm tra đúng thì tăng  $i$  lên 1 và đổi vị trí hai sản phẩm  $swapProducts(&products[i], &products[j])$
- Kết thúc vòng lặp đổi vị trí sản phẩm  $i+1$  và  $high$  sử dụng hàm  $swapProducts(&products[i+1], &products[high])$
- Trả về giá trị  $i+1$  ( $return i+1$ ) và kết thúc
- Hàm sắp xếp  $quicksortPrice$ 
  - Kiểm tra  $low$  với  $high$  ( $if(low < high)$ )
  - Nếu điều kiện kiểm tra đúng thì gọi hàm
    - $partitionPrice(products, low, high, ascending)$  để gán vào biến  $p$
    - $quicksortPrice(products, low, p-1, ascending)$
    - $quicksortPrice(products, p+1, high, ascending)$

### 2.4.4. Tìm kiếm khách hàng

Chức năng này hỗ trợ quản trị viên (admin) tìm kiếm thông tin khách hàng. Hệ thống sử dụng thuật toán tìm kiếm tuyến tính. Kết quả trả về bao gồm đầy đủ thông tin chi tiết của khách hàng: họ và tên, địa chỉ email, và số điện thoại.

Cách thức hoạt động:

Hàm ghép họ, tên đệm và tên  $getFullName$ :

- Trả về họ & tên ( $return lastName + " " + middleName + " " + firstName;$ )

Hàm tìm kiếm tên khách hàng  $searchNameCustomer$

- Gán  $searchNameLower = nameCustomer$
- Chuẩn hóa tên tìm kiếm  $searchNameCustomer$  về chữ thường sử dụng  $transform$  với chế độ  $::tolower$
- Lấy dữ liệu từ file lưu vào mảng động  $customers$  và biến số lượng  $countCustomer$  thông qua hàm  $loadCustomer$
- Gán biến đếm số lượng khách hàng tìm kiếm bằng 0 ( $foundCount = 0$ )
- Tạo vòng lặp  $for$  từ  $i = 0$  đến  $countCustomer-1$ 
  - Chuyển tên khách hàng sang chữ thường
  - Kiểm tra tên khách hàng có chứa trong chuỗi tìm kiếm không
    - Nếu có: tăng biến  $countCustomer$  lên 1
    - Nếu không: tiếp tục vòng lặp  $for$
- Kết thúc vòng lặp  $for$  thì khởi tạo mảng động với kích thước của mảng là  $countCustomer$  và biến  $index = 0$
- Tạo vòng lặp  $for$  từ  $i = 0$  đến  $countCustomer-1$ 
  - Chuyển tên khách hàng sang chữ thường

- Kiểm tra tên khách hàng có chứa trong chuỗi tìm kiếm không
  - Nếu có: thêm khách hàng vào mảng tìm kiếm và tăng biến *countCustomer* lên 1
  - Nếu không: tiếp tục vòng lặp *for*
- Kết thúc vòng lặp *for* thì giải phóng bộ nhớ *delete [] customers*
- Trả về mảng tìm kiếm *return foundCustomer*

### 2.4.5. Tìm kiếm sản phẩm

Chức năng này cho phép quản trị viên (admin) tìm kiếm thông tin sản phẩm. Hệ thống sử dụng thuật toán tìm kiếm nhị phân, cho phép người quản trị thực hiện tra cứu tên sản phẩm cần tìm. Kết quả tìm kiếm sẽ hiển thị đầy đủ các thông tin chi tiết bao gồm: tên sản phẩm, giá tiền, thể loại, nhà sản xuất, số lượng.

Cách thức hoạt động: thuật toán tìm kiếm nhị phân

- Lấy dữ liệu từ file lưu vào mảng động *products* và biến số lượng sản phẩm *countProduct* thông qua hàm *loadProducts*
- Nếu *products = nullptr* hoặc *countProduct = 0* thì số lượng tìm kiếm được *foundCount = 0* và *return nullptr* kết thúc tìm kiếm
- Nếu tên sản phẩm tìm kiếm rỗng sử dụng hàm *nameProduct.empty()* thì số lượng tìm thấy được bằng số sản phẩm *foundCount = countProduct* và trả về mảng động (*return products*) kết thúc tìm kiếm
- Gán *searchNameProduct = nameProduct*
- Chuyển tên tìm kiếm *searchNameProduct* về chữ thường sử dụng *transform* với chế độ *::tolower*
- Sắp xếp lại mảng động tăng dần sử dụng hàm *quicksortNameProduct*
- Khởi tạo mảng động *foundProduct = new Products[countProduct]* và *foundCount = 0*
- Khởi tạo *left = 0* và *right = countProduct - 1*
- Vòng lặp *while*
  - Khởi tạo *mid = (right - left) / 2*
  - Khởi tạo *midNameProduct = product[mid].getNameProduct()* và chuyển sang chữ thường *transform* chế độ *::tolower*
  - Kiểm tra nếu tên sản phẩm tại *mid* chứa trong chuỗi tìm kiếm *if(midNameProduct.find(searchNameProduct) != string::npos)*
  - Nếu điều kiện kiểm tra đúng thì
    - Thêm sản phẩm vào mảng tìm kiếm và tăng biến số lượng lên 1 (*foundProducts[foundCount++] = products[mid]*)

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

- Tìm kiếm các sản phẩm lân cận  $mid$  để lưu vào mảng tìm kiếm nghĩa là gán  $i = mid - 1$  và  $i = mid + 1$  để tạo hai vòng lặp  $while(i \geq 0)$  và  $while(i < countProduct)$  tìm kiếm thêm các sản phẩm lân cận
- Thoát khỏi vòng lặp  $while$  ( $break$ )
- Nếu điều kiện kiểm tra sai thì
  - Nếu nhỏ hơn chuỗi tìm kiếm, thu hẹp phạm vi tìm kiếm về phía bên phải  $left = mid + 1$
  - Nếu nhỏ hơn chuỗi tìm kiếm, thu hẹp phạm vi tìm kiếm về phía bên trái  $right = mid - 1$
- Kiểm tra nếu biến  $foundCount = 0$  hay không
  - Nếu có: giải phóng bộ nhớ ( $delete [] foundProduct$ ) và trả về  $nullptr$  ( $return nullptr$ ) kết thúc
  - Nếu không: trả về mảng tìm kiếm  $return foundProduct$  và kết thúc

### 2.4.6. Thêm sản phẩm mới

Chức năng này cho phép quản trị viên (admin) thêm sản phẩm mới vào hệ thống bán hàng trực tuyến. Quản trị viên cần cung cấp file ảnh sản phẩm và file thông tin chi tiết (bao gồm thiết bị và các đặc điểm nổi bật). Hệ thống yêu cầu nhập các thông tin như: tên file sản phẩm, tên file ảnh, mã sản phẩm, thể loại, tên sản phẩm, giá bán, nhà sản xuất, hệ điều hành và số lượng. Trong quá trình này, hệ thống sẽ tự động kiểm tra tính duy nhất của mã sản phẩm để tránh trùng lặp và tích hợp cơ chế kiểm tra tính hợp lệ của dữ liệu đầu vào.

Cách thức hoạt động:

- Khởi tạo các biến cần thiết để lưu các trường thông tin của sản phẩm cần thêm mới.
- Kiểm tra sự tồn tại của mã sản phẩm ( $productID$ ): mở tệp *Products.txt* và duyệt qua từng dòng để kiểm tra xem mã sản phẩm đã tồn tại trong hệ thống hay chưa. Nếu mã sản phẩm đã tồn tại, thông báo kết thúc việc thêm sản phẩm mới.
- Xử lý theo thể loại của sản phẩm ( $genre$ ): kiểm tra loại sản phẩm thuộc loại nào để ghi thông tin vào các tệp tương ứng, các tệp này được lưu theo các danh mục riêng biệt cho từng loại sản phẩm.
- Ghi thông tin sản phẩm vào tệp *Products.txt*: thông tin cơ bản về sản phẩm (mã sản phẩm, tên sản phẩm, loại sản phẩm, giá, nhà sản xuất, hệ điều hành và số lượng) được ngăn cách với nhau bởi ký tự '|' sẽ được ghi vào tệp *Products.txt* để lưu trữ trong hệ thống.

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

- Lưu thông số kỹ thuật vào tệp *ThongSoKyThuat.txt*: hàm mở tệp chứa thông số kỹ thuật của sản phẩm từ mục đã chuẩn bị trước và ghi vào tệp *ThongSoKyThuat.txt*. Thông số kỹ thuật của một sản phẩm được chứa giữa cặp dấu \*
- Cuối cùng, thông báo sản phẩm đã được thêm vào hệ thống thành công.

### 2.4.7. Chỉnh sửa thông tin sản phẩm

Chức năng này cho phép quản trị viên (admin) chỉnh sửa thông tin sản phẩm. Quản trị viên chọn sản phẩm cần chỉnh sửa. Hệ thống sẽ yêu cầu nhập các thông tin cần chỉnh sửa, bao gồm: tên sản phẩm, thể loại, giá tiền, hệ điều hành và số lượng. Dựa trên các thông tin chỉnh sửa, hệ thống tự động cập nhật cơ sở dữ liệu với các thay đổi.

Cách thức hoạt động:

- Thao tác với cơ sở dữ liệu để lưu thông tin sản phẩm vào mảng động *\*product* và số lượng sản phẩm *productSize* thông qua hàm *loadProducts*
- Kiểm tra lấy thông tin sản phẩm thành công hay không *if (product == NULL || productSize == 0)*
- Duyệt và tìm sản phẩm cần chỉnh sửa *for (int i = 0; i < productSize; i++)* và *if (productID == product[i].getProductID())*
- Nếu tìm thấy sản phẩm cần chỉnh sửa thì mở file *Products.txt* để đọc (*ifstream*)
- Tạo vòng lặp *while* để duyệt qua từng dòng
  - o Phân tách dữ liệu bởi dấu phân cách “|” thành các trường dữ liệu
  - o Cập nhật thông tin sản phẩm nếu kiểm tra đúng mã sản phẩm *if(stoi(id) == productID)*
  - o Lưu thông tin sản phẩm vào chuỗi *fileContent*
- Kết thúc vòng lặp *while* thì đóng file *Products.txt* để đọc *inFile.close()*
- Mở file *Products.txt* để ghi (*ofstream*) và kiểm tra mở file thành công hay không *if (!outFile.is\_open())*
  - o Nếu không: giải phóng bộ nhớ *delete [] product* và kết thúc *return*
  - o Nếu có: lưu thông tin sản phẩm vào file *outFile << fileContent* và thoát vòng lặp *for*
- Giải phóng bộ nhớ *delete [] product* và kết thúc *return*



### 2.4.8. Xoá sản phẩm

Chức năng này cho phép quản trị viên (admin) xoá sản phẩm khỏi hệ thống bán hàng trực tuyến. Để thực hiện, đầu tiên quản trị viên chọn sản phẩm cần xoá. Sau đó, hệ thống sẽ tự động xử lý và xoá sản phẩm khỏi cơ sở dữ liệu thông qua mã sản phẩm đó.

Cách thức hoạt động:

- Mở file thông số kỹ thuật để đọc (*ifstream*) và kiểm tra mở thành công hay không
- Tạo vòng lặp *while* đọc từng dòng
  - o Nếu đọc là dòng *dấu \** thì đánh dấu bắt đầu một sản phẩm
  - o Kiểm tra xem sản phẩm có phải là sản phẩm cần xoá (*productID*) không
    - Nếu không: ghi lại toàn bộ thông tin sản phẩm vào chuỗi *content* và ghi thêm *dấu \** để duy trì định dạng
    - Nếu có: bỏ qua thông tin sản phẩm này trong vòng lặp *while*
- Đóng file *ifstream*
- Mở file thông số kỹ thuật để ghi (*ofstream*) và ghi nội dung mới từ chuỗi *content* vào. Sau đó đóng file *ofstream*
- Mở file *Products.txt* để đọc (*ofstream*)
- Xác định sản phẩm cần xoá (*productsID*) có tồn tại hay không
  - o Nếu có: gán *productFound = true* và tiếp tục vòng lặp
  - o Nếu không: lưu thông tin sản phẩm vào mảng động *product*
- Mở file *Products.txt* để ghi (*ifstream*) và kiểm tra mở file thành công hay không.
  - o Nếu không: thì giải phóng bộ nhớ *delete [] product*
  - o Nếu có: thì lưu thông tin sản phẩm từ mảng động *product* vào file *Products.txt*
- Đóng file *Products.txt* để đọc (*ofstream*) và để ghi (*ifstream*)
- Giải phóng bộ nhớ *delete [] product*
- Thông qua ID của sản phẩm cần xoá, mở các file có liên quan để xoá các thông tin có trong đó (file hình ảnh, tính năng nổi bật, giá cả,...)
- Thông báo xoá sản phẩm thành công hay không tìm thấy

### 2.4.9. Tạo mã giảm giá

Chức năng này cho phép quản trị viên (admin) tạo mã giảm giá. Hệ thống sẽ yêu cầu quản trị viên nhập các thông tin (mã giảm giá, số tiền, số lượng mã). Sau đó, hệ thống sẽ lưu trữ những thông tin này vào cơ sở dữ liệu.

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

Cách thức hoạt động:

- Mở file để ghi với chế độ là ghi vào cuối file *ios::app*
- Kiểm tra mở file thành công hay không sử dụng hàm *is\_open()*, nếu không thành thì *return*
- Chuyển đổi tất cả sang chữ in hoa *transform* với chế độ *::toupper*
- Ghi vào file với những thông tin mã giảm giá, số tiền, số lượng được ngăn cách bởi dấu “|”
- Đóng file sử dụng hàm *close()*

### 2.4.10. Thống kê chi tiết

Chức năng này hỗ trợ quản trị viên (admin) xem chi tiết các đơn đặt hàng trong một ngày cụ thể. Quản trị viên nhập ngày (dd-mm-yyyy), sau đó hệ thống sẽ truy vấn cơ sở dữ liệu, tìm kiếm các đơn hàng phù hợp và hiển thị danh sách bao gồm các thông tin: số thứ tự (STT), mã khách hàng, mã đơn hàng, địa chỉ giao hàng, và tổng số tiền.

Cách thức hoạt động:

- Lấy dữ liệu từ file lưu vào mảng động thông qua hàm *loadPay*
- Hiển thị ra màn hình với dạng danh sách bao gồm các thông tin: số thứ tự, mã khách hàng, địa chỉ giao hàng và tổng số tiền của 1 đơn hàng
- Giải phóng bộ nhớ và kết thúc

### 2.4.11. Thống kê doanh thu

Chức năng này cho phép quản trị viên (admin) xem đồ thị thống kê doanh thu. Hệ thống sẽ truy vấn cơ sở dữ liệu, tổng hợp giá trị các đơn đặt hàng theo từng ngày và hiển thị doanh thu gần đây dưới dạng đồ thị cột.

Cách thức hoạt động:

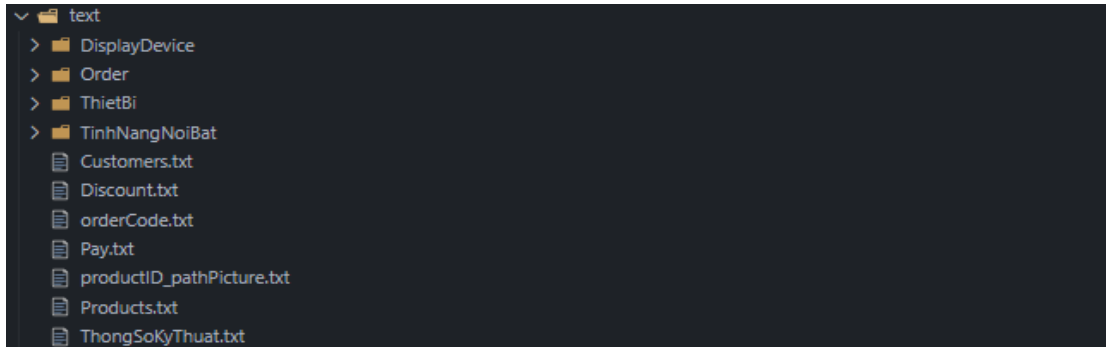
- Lấy dữ liệu từ file lưu vào mảng động thông qua hàm *loadPay* và lấy tổng số tiền của các đơn đặt hàng trong 1 ngày
- Hiển thị ra màn hình với với dạng đồ thị trực quan
- Giải phóng bộ nhớ và kết thúc



### 3. THIẾT KẾ CẤU TRÚC DỮ LIỆU

#### 3.1. Phát biểu bài toán

##### 3.1.1. Mô tả đầu vào (Input): Đầu vào của chương trình gồm các file được lưu trong các thư mục



Hình 3.1-1. Mô tả đầu vào

##### ❖ Thông tin khách hàng có trong hệ thống bao gồm:

- Mã khách hàng
- Họ & đệm
- Tên
- Email
- Số điện thoại
- Tên đăng nhập
- Mật khẩu đăng nhập

```
1|Lương|Văn|Võ|luongvanvo29@gmail.com|0865321921|vanvo299|voluong  
5|Nguyễn||Hiếu|nguyenhieu126@gmail.com|0987451743|hieu|hieu  
6|Lương|Văn|Vũ|vohocgioi@gmail.com|0987|vu|vu  
7|Nguyễn|Trần Ly|Na|nguyentranlyna2805@gmail.com|0938157041|lyna|lyna  
8|Nguyễn|Công|Nhật|congngat567@gmail.com|0383604320|congngat|ngat  
-1|admin|admin|admin|null|null|admin_1|1
```

Hình 3.1-2. Mô tả đầu vào thông tin khách hàng

##### ❖ Thông tin sản phẩm có trong hệ thống bao gồm:

- Mã sản phẩm
- Tên sản phẩm
- Thể loại
- Giá tiền
- Nhà sản xuất
- Hệ điều hành
- Thông số kỹ thuật
- Số lượng

PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

101		iPhone 15 Pro Max		Điện Thoại		1158.46		Apple		iOS		46
102		Samsung Galaxy S24 Ultra		Điện Thoại		1044.15		Samsung		Android		28
103		Xiaomi 14T Pro		Điện Thoại		669.69		Xiaomi		Android		30
104		OPPO Reno10 Pro+		Điện Thoại		551.44		OPPO		Android		29
105		Tecno Pova 6 Neo		Điện Thoại		161.66		Tecno		Android		29
106		Samsung Galaxy Z Flip 6		Điện Thoại		1066.8		Samsung		Android		25
107		OPPO Find X5 Pro		Điện Thoại		632.02		OPPO		Android		23
108		iPhone 16		Điện Thoại		877.08		Apple		iOS		35
113		Lenovo LOQ 15IAX9 83GS001RVN		Laptop		839.18		Lenovo		Windows		27
114		MSI Gaming GF63 12VE-460VN		Laptop		709.11		MSI		Windows		29
115		ASUS TUF Gaming F15 FX507ZC4-HN074W		Laptop		748.52		ASUS		windows		32

Hình 3.1-3. Mô tả đầu vào thông tin sản phẩm

* 101								
- Màn hình						Kích thước màn hình: 6.7 Inches Công nghệ màn hình: Super Retina XDR OLED Độ phân giải màn hình: 2796 x 1290-pixel Tần số quét: 120Hz Kiểu màn hình: Dynamic Island		
- Camera sau						Camera sau: Camera chính: 48MP, 24 mm f/1.78 Camera góc siêu rộng: 12 MP, 13mm, f/2.2 Camera tele: 12 MP Quay video: 4K@24/25/30/60 fps HD 1080p@25/30/60 fps HD 720p@30 fps		
- Camera trước						Camera trước: 12 MP, f/1.9 Quay video trước: 4K@24/25/30/60 fps HD 1080p@25/30/60 fps		
- Vi xử lý & đồ họa						Chipset: Apple A17 Pro 6 nhân Loại CPU: CPU 6 lõi mới với 2 lõi hiệu năng và 4 lõi hiệu suất GPU: GPU 6 nhân mới		
- Giao tiếp & kết nối						Công nghệ NFC: Có Thẻ SIM: 2 SIM(nano-SIM và eSIM) Jack tai nghe 3.5: Không Hỗ trợ mạng: 5G Wi-Fi: Wi-Fi 6E (802.11ax) Bluetooth v5.3		
- Pin & công nghệ sạc						Pin: 4422 mAh Công nghệ sạc: Sạc nhanh 20W Sạc không dây 15W Sạc không dây Cổng sạc: USB Type-C		
- Thiết kế & trọng lượng						Kích thước: 159.9 x 766.7 x 8.25 mm Trọng lượng: 221 g Chất liệu mặt lưng: Kính Chất liệu khung viền: Titanium		
*								

Hình 3.1-4. Mô tả đầu vào thông số kỹ thuật của sản phẩm

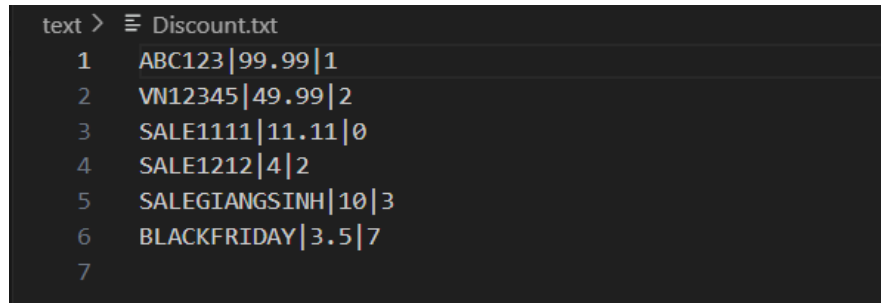
101		Graphics/Picture/iPhone15Promax.png
102		Graphics/Picture/samsungS24Ultra.png
103		Graphics/Picture/xiaomi14TPro.png
104		Graphics/Picture/OPPOReno10pro+.jpg
105		Graphics/Picture/TecnoPova6Neo.png
106		Graphics/Picture/SamsungZFlip6.jpg
107		Graphics/Picture/OPPOFindX5Pro.jpg

Hình 3.1-5. Mô tả đầu vào đường dẫn ảnh

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

### ❖ Thông tin mã giảm giá có trong hệ thống bao gồm:

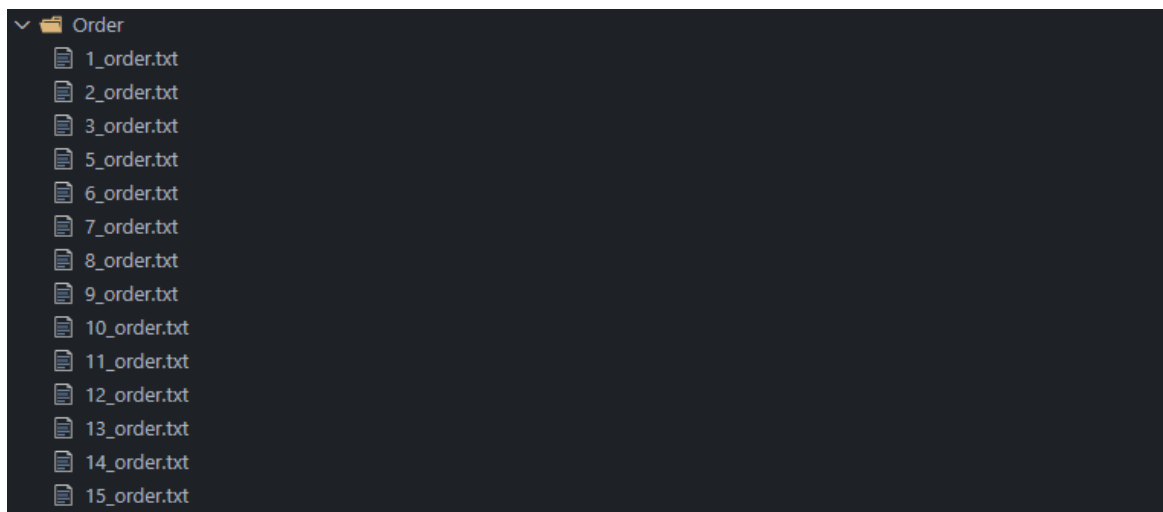
- Mã giảm giá
- Số tiền giảm
- Số lượng phiếu giảm giá



```
text > Discount.txt
1  ABC123|99.99|1
2  VN12345|49.99|2
3  SALE1111|11.11|0
4  SALE1212|4|2
5  SALEGIANGSINH|10|3
6  BLACKFRIDAY|3.5|7
7
```

Hình 3.1-6. Mô tả đầu vào mã giảm giá

### ❖ Thông tin giỏ hàng: các file lưu dữ liệu giỏ hàng của từng khách hàng:



```
Order
├── 1_order.txt
├── 2_order.txt
├── 3_order.txt
├── 5_order.txt
├── 6_order.txt
├── 7_order.txt
├── 8_order.txt
├── 9_order.txt
├── 10_order.txt
├── 11_order.txt
├── 12_order.txt
├── 13_order.txt
├── 14_order.txt
└── 15_order.txt
```

Hình 3.1-7. Mô tả đầu vào các file giỏ hàng của khách hàng

### 3.1.2. Yêu cầu đầu ra (Output):

#### ❖ Quản lý khách hàng

- Đăng ký và đăng nhập
- Quên mật khẩu và đổi mật khẩu
- Tìm kiếm theo tên của khách hàng
- Lưu trữ thông tin khách hàng vào cơ sở dữ liệu

#### ❖ Quản lý thiết bị điện tử

- Thêm, chỉnh sửa và xóa sản phẩm
- Tìm kiếm theo tên sản phẩm
- Sắp xếp theo tên và giá của sản phẩm
- Lưu trữ thông tin sản phẩm vào cơ sở dữ liệu

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

### ❖ **Quản lý mã giảm giá**

- Thêm mã giảm giá
- Lưu trữ thông tin mã giảm giá vào cơ sở dữ liệu

### ❖ **Quản lý giỏ hàng**

- Thêm sản phẩm vào giỏ hàng
- Tăng giảm số lượng trong giỏ hàng
- Xoá sản phẩm ra khỏi giỏ hàng
- Lưu trữ thông tin giỏ hàng vào cơ sở dữ liệu

### ❖ **Quản lý các đơn đặt hàng**

- Nhập địa chỉ giao hàng
- Kiểm tra mã giảm giá (nếu có)
- Chức năng thanh toán
- Lưu trữ thông tin đơn hàng vào cơ sở dữ liệu

### ❖ **Xem lịch sử và thống kê doanh thu**

- Xem lịch sử giao dịch
- Thống kê chi tiết
- Đồ thị doanh thu

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

**Mảng động:** sử dụng các mảng động có kiểu dữ liệu là các class riêng biệt để lưu trữ thông tin khách hàng, thiết bị điện tử.

- Mảng động Customer: dùng để lưu thông tin khách hàng
- Mảng động Products: dùng để lưu thông tin sản phẩm
- Mảng động History: dùng để lưu thông tin lịch sử các đơn đặt hàng
- Mảng động Discount: dùng để lưu thông tin mã giảm giá

### **Tìm kiếm:**

- Sử dụng thuật toán tìm kiếm nhị phân (Binary Search) để tìm kiếm thông tin của sản phẩm cần tìm
- Sử dụng thuật toán tìm kiếm tuyến tính để tìm kiếm thông tin của khách hàng thông qua tên khách hàng

**Sắp xếp:** sử dụng thuật toán quicksort, một thuật toán sắp xếp hiệu quả và nhanh chóng. Để sắp xếp thông tin khách hàng và sản phẩm.

- Sắp xếp thông tin khách hàng theo tên khách hàng.
- Sắp xếp thông tin sản phẩm theo tên sản phẩm.
- Sắp xếp thông tin sản phẩm theo giá của sản phẩm.

## 4. ỨNG DỤNG HƯỚNG ĐỐI TƯỢNG VÀO BÀI TOÁN

### 4.1. Phân tích các lớp của hướng đối tượng

#### a/ Class Customer

```
1  class Customer {
2  private:
3      int customerID;
4      string lastName;
5      string firstName;
6      string middleName;
7      string email;
8      string phone;
9      string userName;
10     string password;
11 public:
12     Customer(int customerID, string lastName, string middleName, string firstName,
13             string email, string phone, string userName, string password);
14     Customer();
15     // Getter
16     int getCustomerID();
17     string getLastName();
18     string getMiddleName();
19     string getFirstName();
20     string getEmail();
21     string getPhone();
22     string getUserName();
23     string getPassword();
24     string getFullName() const;
25
26     void resizeCustomerArray(Customer*& customers, int &size, const Customer& newCustomer);
27
28     void setPassword(string &password);
29     // Verify login
30     bool verifyLogin(const string &userName, const string &password);
31
32     int login(const string& username, const string &password, Customer *&loggedInCustomer);
33     void registerAccount(string lastName, string middleName, string firstName, string email,
34                        string phone, string userName, string password);
35
36     int verifyPhone(const string &phone); // Hàm kiểm tra số điện thoại nhập vào có đúng
37     int verifyEmail(const string &email); // Hàm kiểm tra email nhập vào có đúng
38     // Quên mật khẩu => đổi lại mật khẩu thông qua số điện thoại
39     void forgetPassword(const string& phone, string &newPassword, string& confirmNewPassword);
40     void changePassword();
41     void updateCustomerInfo();
42     void displayCustomer();
43
44     // Hoán đổi hai khách hàng với nhau
45     static void swapCustomer(Customer *customer1, Customer *customer2);
46     static int compareNameCustomer(Customer *customer1, Customer *customer2);
47
48     // Sắp xếp theo tên khách hàng
49     static int partitionNameCustomer(Customer *customers, int low, int high, bool ascending);
50     static void quicksortNameCustomer(Customer *customers, int low, int high, bool ascending);
51
52     static Customer* searchNameCustomer(string nameCustomer, int &foundCount);
53 };
54
```

Hình 4.1-1. Thuộc tính và phương thức lớp Customer

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

\* Lớp Customer gồm các thuộc tính:

- customerID: mã của khách hàng
- lastName: họ của khách hàng
- middleName: tên đệm của khách hàng
- firstName: tên của khách hàng
- email: địa chỉ email của khách hàng
- phone: số điện thoại của khách hàng
- userName: tên đăng nhập của khách hàng
- password: mật khẩu tài khoản của khách hàng

\* Lớp Customer gồm các phương thức:

- Phương thức Getter, Setter:
  - + getCustomerID(): truy cập mã khách hàng
  - + getLastName(): truy cập họ khách hàng
  - + getMiddleName(): truy cập tên đệm của khách hàng
  - + getFirstName(): truy cập tên của khách hàng
  - + getEmail(): truy cập địa chỉ email của khách hàng
  - + getPhone(): truy cập số điện thoại của khách hàng
  - + getUserName(): truy cập tên đăng nhập của khách hàng
  - + getPassword(): truy cập mật khẩu của khách hàng
  - + getFullName(): truy cập họ và tên đầy đủ của khách hàng
  - + setPassword(): cập nhật thuộc tính password
- resizeCustomerArray: mở rộng mảng động và chứa thêm 1 phần tử mới
- verifyLogin: xác thực thông tin đăng nhập
- login: đăng nhập tài khoản
- registerAccount: tạo tài khoản mới
- verifyEmail: kiểm tra email
- forgetPassword: quên mật khẩu và đổi mật khẩu
- swapCustomer (static): hoán đổi vị trí hai khách hàng
- compareNameCustomer: so sánh tên của hai khách hàng
- partitionNameCustomer: phân chia mảng theo giá trị pivot
- quicksortNameCustomer: sắp xếp tên khách hàng
- searchNameCustomer: tìm kiếm theo tên khách hàng



### b/ Class Products

```
1 class Products{
2     protected:
3         int productID; // mã sản phẩm
4         string nameProduct; // tên sản phẩm
5         string genre; // thể loại của sản phẩm
6         double priceProduct; // giá
7         string manufacturer; // nhà sản xuất
8         string operatingSystem; // hệ điều hành
9         string specifications; // Thông số kỹ thuật
10        int Count; // số lượng sản phẩm
11    public:
12        Products();
13        Products(int productID, string nameProduct, string genre, double priceProduct,
14            string manufacturer, string operatingSystem, int Count);
15        Products(int ProductID, string specifications);
16        Products(int productID, string nameProduct, string genre, double priceProduct,
17            string manufacturer, string operatingSystem, string specifications, int Count);
18        ~Products();
19
20        // getter
21        int getProductID();
22        string getNameProduct();
23        string getGenre();
24        double getPriceProduct();
25        string getManufacturer();
26        string getOperatingSystem();
27        string getSpecifications();
28        int getCount();
29
30        // setter
31        void setProductID(int productID);
32        void setNameProduct(string nameProduct);
33        void setGenre(string genre);
34        void setPriceProduct(double priceProduct);
35        void setManufacturer(string manufacturer);
36        void setOperatingSystem(string operatingSystem);
37        void setCount(int count);
38
39        // display
40        void displayProducts(); // Hiển thị ra sản phẩm
41        // Hiển thị cấu hình chi tiết của sản phẩm
42        string displaySpecification(int productID);
43
44        friend ostream& operator<<(ostream& out, const Products& product);
45
46        static void swapProducts(Products *product1, Products *product2);
47        static int compareNameProduct(Products *product1, Products *product2);
48
49        // Sắp xếp theo tên sản phẩm
50        static int partitionNameProduct(Products *products, int low, int high, bool ascending);
51        static void quicksortNameProduct(Products *products, int low, int high, bool ascending);
52
53        static Products* searchNameProducts(const string& nameProduct, int& foundCount);
54
55        // Lấy thông số kỹ thuật của sản phẩm
56        string getSpecificationByID(const string &productFileName, const string &indexFileName, int productID);
57
58        // Sắp xếp sản phẩm theo giá tiền
59        static int partitionPrice(Products *products, int low, int high, bool ascending);
60        static void quicksortPrice(Products *products, int low, int high, bool ascending);
61    };
62
```

Hình 4.1-2. Thuộc tính và phương thức lớp Products

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

Lớp Products gồm các thuộc tính:

- productID: mã của sản phẩm
- nameProduct: tên của sản phẩm
- genre: thể loại của sản phẩm
- priceProduct: giá của sản phẩm
- manufacturer: nhà sản xuất của sản phẩm
- operatingSystem: hệ điều hành của sản phẩm
- specifications: thông số kỹ thuật của sản phẩm
- Count: số lượng sản phẩm

Lớp Products gồm các phương thức:

- Phương thức Getter, Setter:
  - + getProductID: truy cập mã sản phẩm
  - + getNameProduct: truy cập tên sản phẩm
  - + getPriceProduct: truy cập giá sản phẩm
  - + getManufacturer: truy cập nhà sản xuất
  - + getOperatingSystem: truy cập hệ điều hành
  - + getSpecification: truy cập thông số kỹ thuật
  - + getCount: truy cập số lượng
  - + setProductID: cập nhật thuộc tính mã sản phẩm
  - + setNameProduct: cập nhật thuộc tính tên sản phẩm
  - + setPriceProduct: cập nhật thuộc tính giá sản phẩm
  - + setManufacturer: cập nhật thuộc tính nhà sản xuất
  - + setOperatingSystem: cập nhật thuộc tính hệ điều hành
  - + setSpecification: cập nhật thuộc tính thông số kỹ thuật
  - + setCount: cập nhật thuộc tính số lượng
- displayProducts(): hiển thị sản phẩm ra màn hình
- displaySpecification(): hiển thị cấu hình chi tiết của sản phẩm
- operator <<(): xuất sản phẩm ra màn hình
- swapProducts(): hoán đổi hai sản phẩm
- compareNameProduct(): so sánh tên hai sản phẩm
- searchNameProduct(): tìm kiếm theo tên của sản phẩm
- Phương thức sắp xếp:
  - + partitionNameProduct(), partitionPrice(): hàm phân hoạch Lomuto, là bước quan trọng trong thuật toán sắp xếp quicksort
  - + quicksortNameProduct(): sắp xếp theo tên của sản phẩm
  - + quicksortPrice(): sắp xếp theo giá của sản phẩm



### c/ Class Order

```
1  class Order : public Customer, public FileManager, public Discount, public History {
2      static Products* products;
3      static int countProduct;
4      Products* data;    // con trỏ động chữ danh sách
5      int count;         // số lượng hiện tại
6      static int customerID; // ID của khách hàng
7      string addressOrder; // Địa chỉ của đơn hàng
8      static long orderCode; // Mã đơn hàng
9  public:
10     Products *getData() const;
11     Order();
12     Order(const Order& other);
13     Order& operator=(const Order& other);
14     ~Order();
15     int getCountP();
16     void displayProduct();
17     void saveToFile(const string fileProducts);
18     // lấy ID của khách hàng
19     void setID(int ID);
20     void update();
21     // thêm vào
22     void addData(int ID);
23     // xóa
24     void deleteData(int ID);
25     void deleteData(string name);
26     // in ra
27     void displayOrder();
28     // tìm kiếm đơn hàng trong order
29     int findData(int& ID) const;
30     int findData(const string& name) const;
31     //cập nhập thông tin đơn hàng khi có sự thay đổi
32     void updateData(int&, const string&, const string&, const string&, const string&, double, int);
33     void clearOrder(); // xóa toàn bộ đơn hàng
34
35     // thanh toán
36     void payment(string &addressOrder, string discountCode, double &priceDiscount);
37
38     // Xem lịch sử giao dịch
39     void displayHistory();
40     void displayDetailsHistory(long orderCode, Products*& data, int &count);
41
42     // Thống kê doanh thu
43     double statistic(const string& date); // 1 ngày
44
45     int getCount();
46     void increaseProductCount(int productID);
47     void decreaseProductCount(int productID);
48
49     void setAddressOrder(string addressOrder);
50     string getAddressOrder();
51 };
```

Hình 4.1-3. Thuộc tính và phương thức class Order

Lớp Order gồm các thuộc tính:

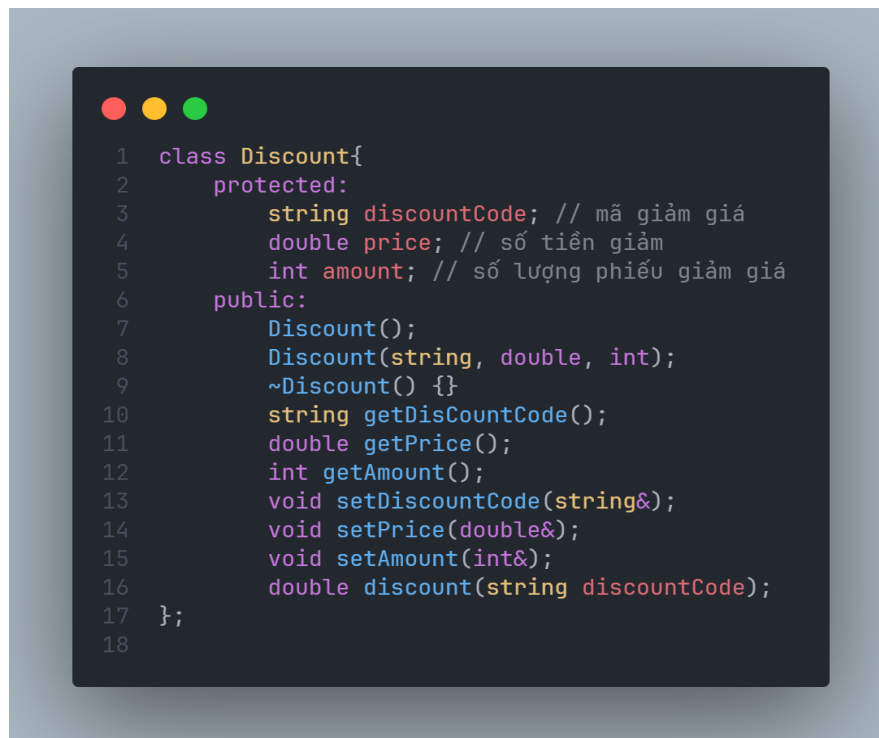
- \*products: mảng động lưu thông tin sản phẩm trong hệ thống
- countProducts: số lượng sản phẩm khác nhau trong hệ thống
- \*data: mảng động lưu thông tin sản phẩm trong giỏ hàng
- count: số lượng sản phẩm khác nhau trong giỏ hàng
- customerID: mã khách hàng
- addressOrder: địa chỉ của đơn hàng

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

Lớp Order gồm các phương thức:

- setID(): cập nhật thuộc tính mã sản phẩm
- update(): cập nhật thông tin cho giỏ hàng
- addData(): thêm sản phẩm vào giỏ hàng
- deleteData(): xoá sản phẩm ra khỏi giỏ hàng
- displayOrder(): hiển thị thông tin sản phẩm trong giỏ hàng ra màn hình
- findData(): tìm kiếm thông tin sản phẩm
- updateData(): cập nhật thông tin sản phẩm trong giỏ hàng nếu sản phẩm đó có sự thay đổi
- clearOrder(): xoá toàn bộ sản phẩm ra khỏi giỏ hàng
- payment(): thanh toán toàn bộ sản phẩm trong giỏ hàng
- displayHistory(): xem lịch sử các đơn đặt hàng
- displayDetailsHistory(): xem chi tiết lịch sử các đơn đặt hàng
- statistic(): thống kê doanh thu
- getCount(): số lượng sản phẩm hiện tại trong giỏ hàng
- increaseProductCount(): tăng số lượng một sản phẩm trong giỏ hàng
- decreaseProductCount(): giảm số lượng một sản phẩm trong giỏ hàng
- setAddressOrder(): cập nhật thuộc tính địa chỉ đơn hàng
- getAddressOrder(): truy cập địa chỉ đơn hàng

### d/ Class Discount

A screenshot of a code editor showing the implementation of the Discount class in C++. The code is written in a dark-themed editor with line numbers from 1 to 18. The class has protected attributes for discount code, price, and amount, and public methods for initialization, getters, setters, and a discount calculation method.

```
1  class Discount{
2      protected:
3          string discountCode; // mã giảm giá
4          double price; // số tiền giảm
5          int amount; // số lượng phiếu giảm giá
6      public:
7          Discount();
8          Discount(string, double, int);
9          ~Discount() {}
10         string getDiscountCode();
11         double getPrice();
12         int getAmount();
13         void setDiscountCode(string&);
14         void setPrice(double&);
15         void setAmount(int&);
16         double discount(string discountCode);
17     };
18
```

Hình 4.1-4. Thuộc tính và phương thức class Discount

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

Lớp Discount gồm các thuộc tính:

- discountCode: mã giảm giá
- price: số tiền giảm
- amount: số lượng phiếu giảm giá

Lớp Discount gồm các phương thức:

- Phương thức Getter, Setter:
  - + getDiscountCode: truy cập mã giảm giá
  - + getPrice: truy cập số tiền giảm
  - + getAmount: truy cập số lượng phiếu giảm giá
  - + setDiscountCode: cập nhật thuộc tính mã giảm giá
  - + setPrice: cập nhật thuộc tính số tiền giảm
  - + setAmount: cập nhật thuộc tính số lượng phiếu giảm giá

### e/ Class History

```
1  class History {
2      protected:
3          string currentDate;
4          int customerID;
5          long orderCode;
6          string addressOrder;
7          double priceTotal;
8          string item;
9      public:
10         History();
11         History(string currentDate, int customerID, long orderCode,
12             string addressOrder, double priceTotal);
13         History(string currentDate, int customerID, long orderCode,
14             string addressOrder, double priceTotal, string item);
15         ~History();
16         string getCurrentDate();
17         int getCustomerID();
18         long getOrderCode();
19         string getAddressOrder();
20         double getPriceTotal();
21         string getItem();
22     };
23
```

Hình 4.1-5. Thuộc tính và phương thức class History

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

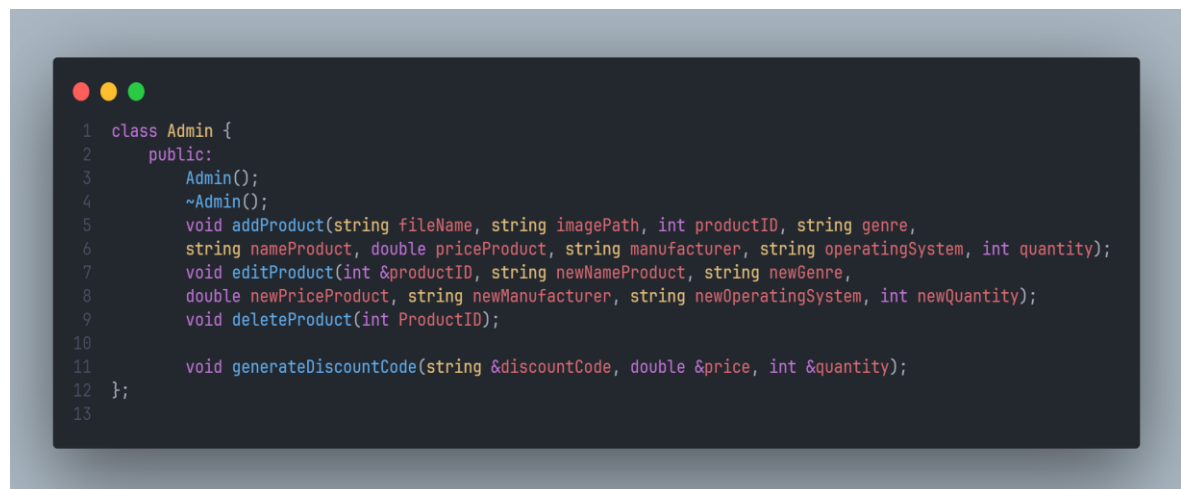
Lớp History gồm các thuộc tính:

- currentDate: ngày tháng năm đặt hàng
- customerID: mã sản phẩm của khách hàng đặt hàng
- orderCode: mã của đơn đặt hàng
- addressOrder: địa chỉ giao hàng
- priceTotal: giá tiền của đơn đặt hàng
- item: nội dung của đơn đặt hàng

Lớp History gồm các phương thức:

- Phương thức Getter, Setter:
  - + getCurrentDate: truy cập ngày tháng năm
  - + getCustomerID: truy cập mã khách hàng
  - + getOrderCode: truy cập mã đơn đặt hàng
  - + getAddressOrder: truy cập địa chỉ giao hàng
  - + getPriceTotal: truy cập giá tiền
  - + getItem: truy cập nội dung của đơn hàng

### f/ Class Admin



Hình 4.1-6. Thuộc tính và phương thức class Admin

Lớp Admin gồm các phương thức:

- addProduct(): thêm sản phẩm vào hệ thống
- editProduct(): chỉnh sửa thông tin của sản phẩm
- deleteProduct(): xoá sản phẩm ra khỏi hệ thống
- generateDiscountCode(): tạo mã giảm giá

### g/ Class FileManager

```

1 class FileManager
2 {
3 public:
4     FileManager();
5     ~FileManager();
6     static Customer *loadCustomer(const std::string &fileCustomer, int &countCustomer);
7     static void saveCustomer(const std::string &fileCustomer, Customer &customer);
8
9     static Products *loadSpecifications(const string &fileThongSoKyThuat, int &countProducts);
10    static Products *loadProducts(const std::string &fileProducts, int &countProducts);
11    static void saveProducts(const string fileProducts, Products *products, int &countProduct);
12
13    static Products *loadOrder(const string &fileOrder, int &countOrder);
14    static void saveOrder(const string &filename, Products *data, int &count);
15
16    static int loadOrderCode(const string &filename);
17    static void saveOrderCode(const string &filename, long &orderCode);
18
19    static History *loadPay(const string &filename, int &countPay, int &customerID);
20    static History *loadPay(const string &filename, const string &currentDate, int &countPay, double &totalPrice);
21    static void savePay(const string &filename, Products *pay, int &count, int customerID,
22        string addressOrder, long orderCode, double &priceDiscount);
23
24    static string getCurrentDate(); // Lấy thời gian hiện tại
25
26    static void saveDiscount(const string &filename, Discount *discount, int &count);
27    static Discount *loadDiscount(const string &filename, int &count);
28
29    // Đọc file trong TỉnhNangNoiBat
30    string readTextTinhNangNoiBat(const string &filePath);
31
32    // Đọc file và trả về mảng string
33    static string *readFromFile(const string &filePath, int &count);
34    // Đọc file chứa id và path đường dẫn sản phẩm
35    static vector<pair<int, string>> readFromFile(const string &fileName);
36
37    // Xóa 1 dòng trong file
38    void deleteLineFromFile(const string &filePath, int lineToDelete);
39
40    // Tạo file chỉ mục
41    void createIndexFile(const string &productFileName, const string &indexFileName);
42
43    static int *loadLineNumbersFromFile(const string &filePath, int &lineCount);
44
45    // load ID từ file thành 1 mảng số nguyên
46    static int *loadIDFromFile(const string &filePath, int &count);
47 };
48

```

Hình 4.1-7. Thuộc tính và phương thức class FileManager

Lớp FileManager gồm các phương thức:

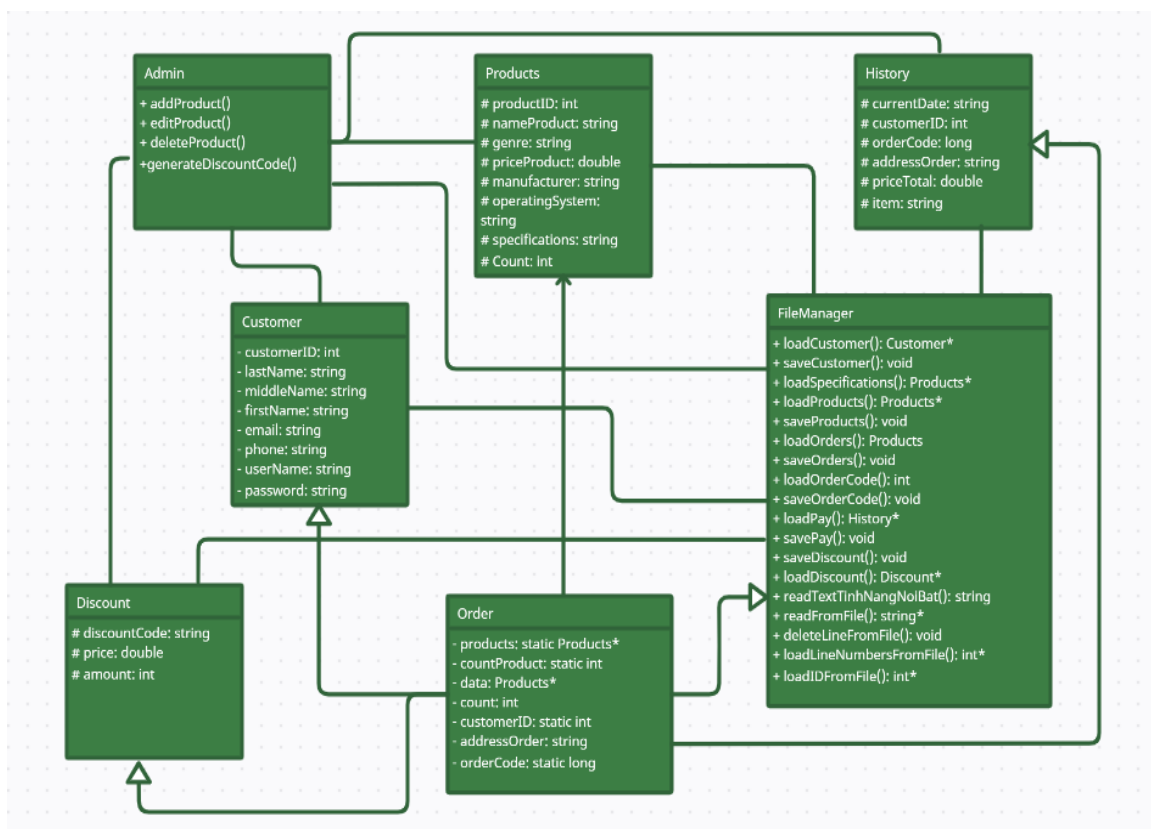
- loadCustomer(): đọc thông tin khách hàng từ file và lưu vào mảng động
- saveCustomer(): lưu thông tin khách hàng vào cơ sở dữ liệu
- loadSpicification(): đọc lưu thông tin tham số kỹ thuật của sản phẩm từ file và lưu vào mảng động
- loadProducts(): đọc thông tin sản phẩm từ file và lưu vào mảng động
- saveProducts():lưu thông tin sản phẩm vào cơ sở dữ liệu
- loadOrder(): đọc thông tin giỏ hàng từ file và lưu vào mảng động
- saveOrder(): lưu thông tin giỏ hàng vào cơ sở dữ liệu

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

- loadOrderCode(): đọc thông tin mã đơn hàng từ file và lưu vào biến
- saveOrderCode(): lưu thông tin mã đơn hàng vào cơ sở dữ liệu
- loadPay(): đọc thông tin đơn đặt hàng từ file và lưu vào mảng động
- savePay(): lưu thông tin đơn đặt hàng vào cơ sở dữ liệu
- getCurrentDate(): lấy thời gian hiện tại
- saveDiscount(): lưu thông tin mã giảm giá vào cơ sở dữ liệu
- loadDiscount(): đọc thông tin mã giảm giá từ file và lưu vào mảng động
- readTextTinhNangNoiBat(): đọc file trong tính năng nổi bật
- readFromFile(): đọc từ file
  - + trả về kiểu vecto là đọc ID + link ảnh
  - + trả về kiểu string là lưu vào mảng string
- deleteLineFromFile(): xóa 1 dòng trong file txt
- createIndexFile(): tạo file chỉ mục
- loadLineNumbersFromFile():
- loadIDFromFile(): đọc ID từ file thành 1 mảng số nguyên

### 4.2. Mỗi liên kết phụ thuộc lẫn nhau giữa các thuộc tính của các lớp

- Mô hình ER để minh họa mối quan hệ giữa các thực thể:



Hình 4.2-1. Mô hình ER minh họa mối quan hệ giữa các thực thể

### 4.3. Kết quả

Hệ thống quản lý và mua bán thiết bị điện tử được thiết kế nhằm hỗ trợ người dùng (khách hàng, nhân viên bán hàng, và quản trị viên) thao tác trực tiếp trên giao diện. Hệ thống cung cấp các chức năng chính như thêm mới thiết bị, tìm kiếm sản phẩm, thống kê doanh thu, và quản lý thông tin khách hàng. Mỗi chức năng của hệ thống đều được minh họa rõ ràng qua các đoạn mã chi tiết và ảnh giao diện chương trình. Mỗi thao tác trên hệ thống đều đảm bảo tính tiện lợi và trực quan cho người dùng, với kết quả được hiển thị đầy đủ và chi tiết, giúp nâng cao hiệu quả quản lý và tối ưu hoá trải nghiệm cho người dùng.

## 5. CHƯƠNG TRÌNH VÀ KẾT QUẢ

### 5.1. Tổ chức chương trình

- Ngôn ngữ lập trình: C++
- Thư viện đồ họa: SFML
- Ngoài ra, chương trình sử dụng:
  - Makefile: tệp định nghĩa các lệnh để biên dịch & liên kết chương trình
  - cURL: được sử dụng để thực hiện yêu cầu mạng, ở đây được dùng để gửi mã OTP về gmail của khách hàng trong việc đổi mật khẩu.
  - JsonCpp: được sử dụng để xử lý dữ liệu JSON, giúp dễ dàng làm việc với các tệp cấu hình hoặc giao tiếp với các API web.

### 5.2. Kết quả thực thi của chương trình

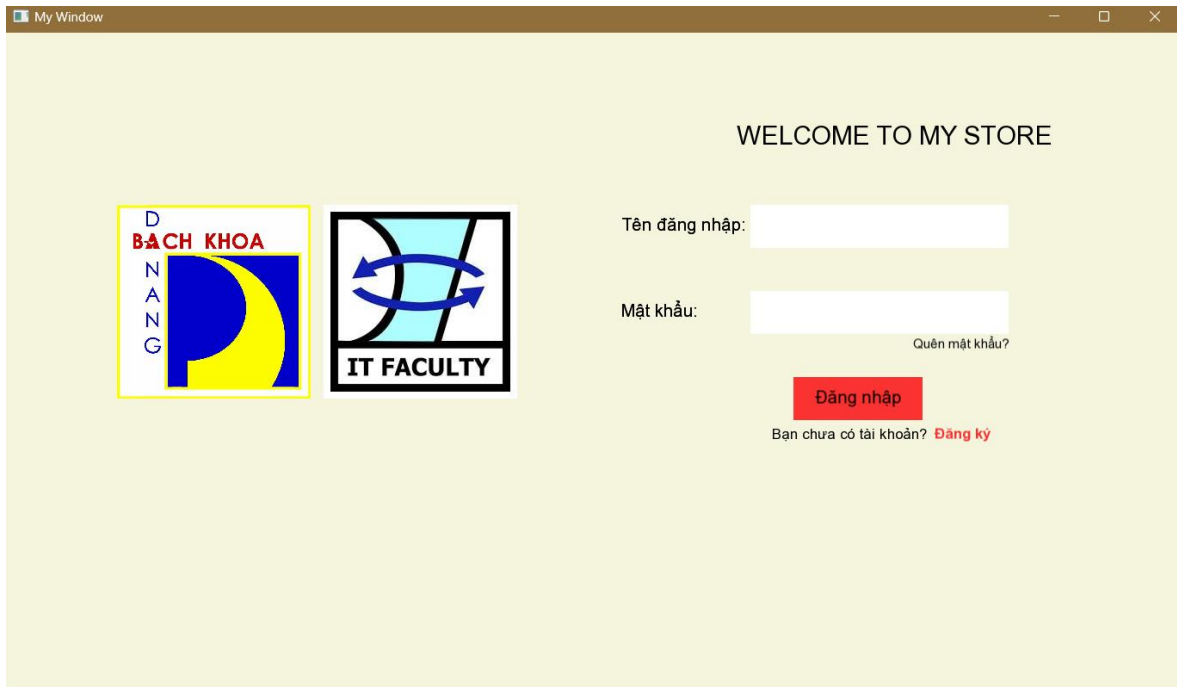
#### 5.2.1. Giao diện khách hàng



## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

### 1. Giao diện đăng nhập:

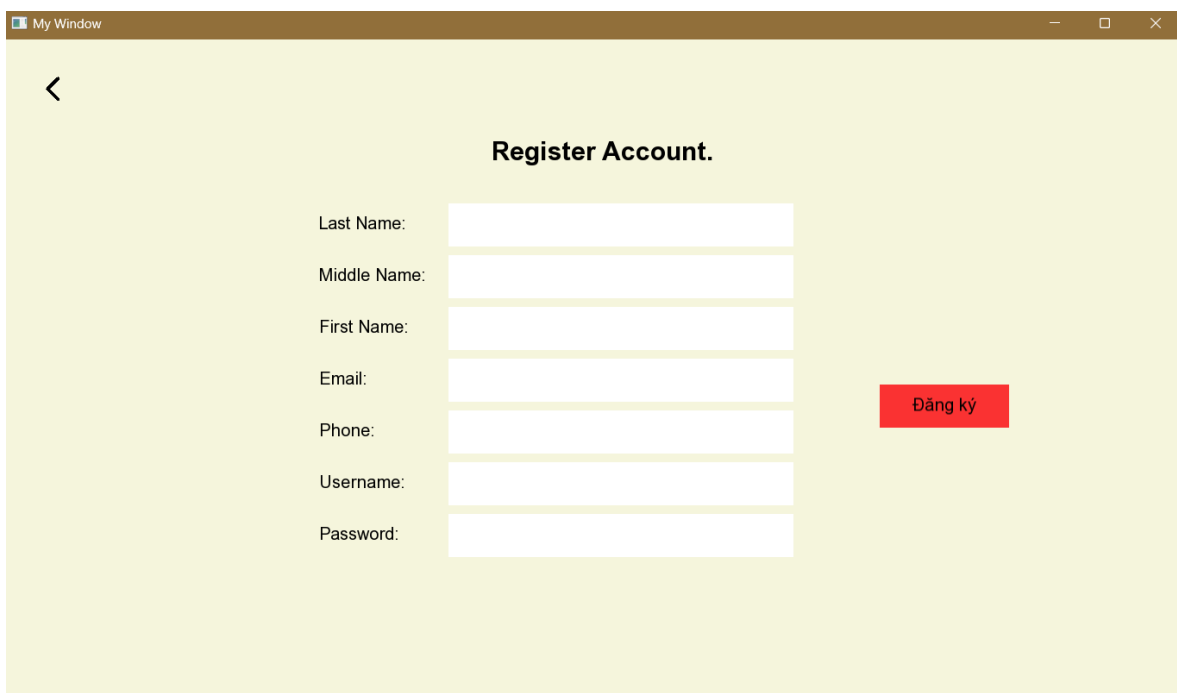
- Khi khởi chạy, giao diện đăng nhập xuất hiện. Người dùng nhập tên đăng nhập và mật khẩu để đăng nhập hoặc chọn *Đăng ký* để tạo tài khoản mới.



Hình 5.2-1. Giao diện đăng nhập

### 2. Giao diện đăng ký:

- Người dùng nhập đầy đủ và chính xác thông tin đăng ký, hệ thống xử lý và đưa người dùng trở lại màn hình đăng nhập khi đã đăng ký thành công.



Hình 5.2-2. Giao diện đăng ký

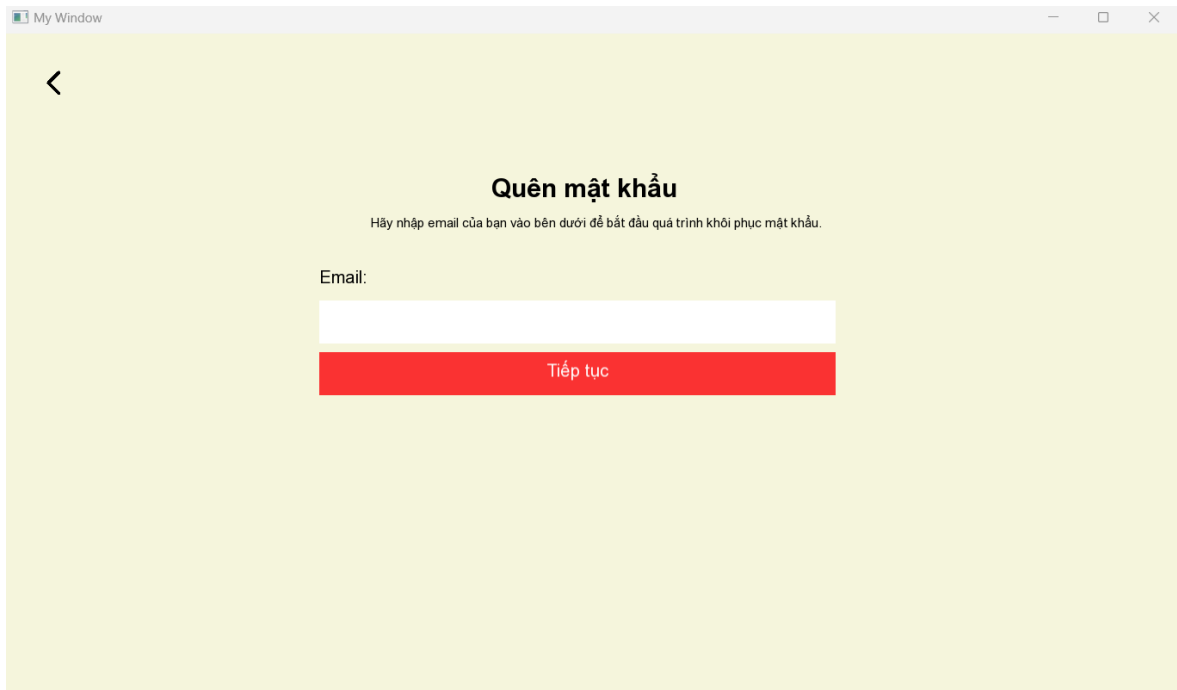


## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

### 3. Giao diện quên mật khẩu

#### a. Giao diện nhập email người dùng

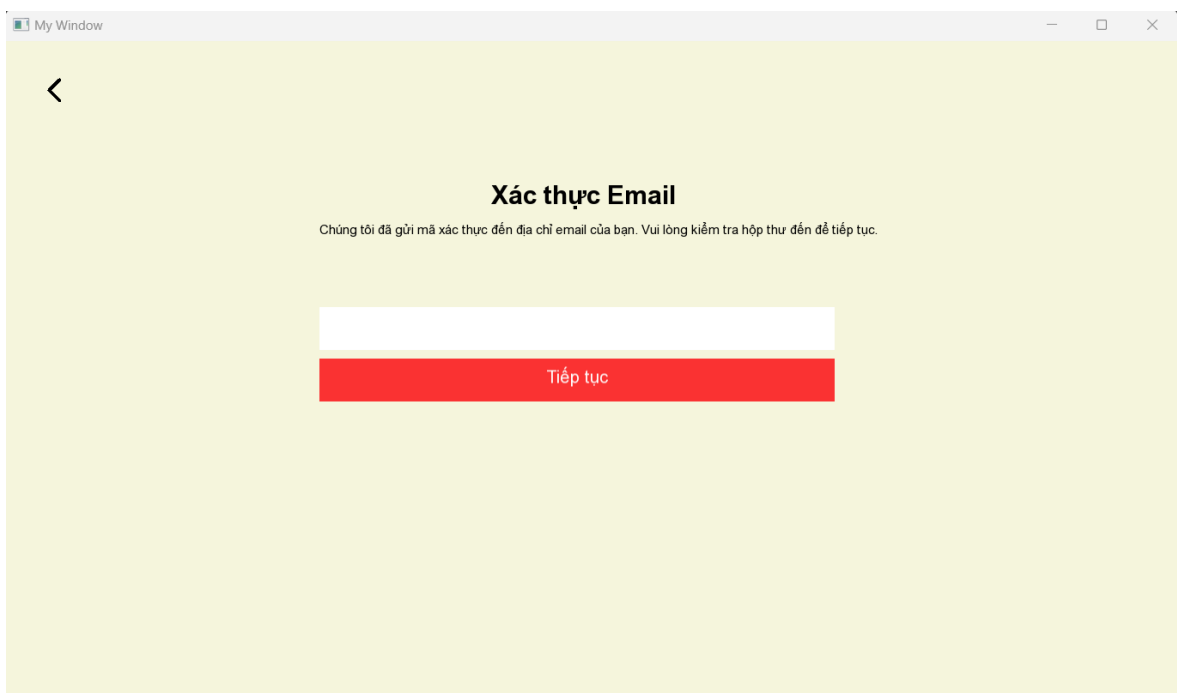
- Người dùng nhập email khi đăng ký tài khoản và chọn nút *Tiếp tục*



*Hình 5.2-3. Giao diện nhập email*

#### b. Giao diện nhập mã OTP

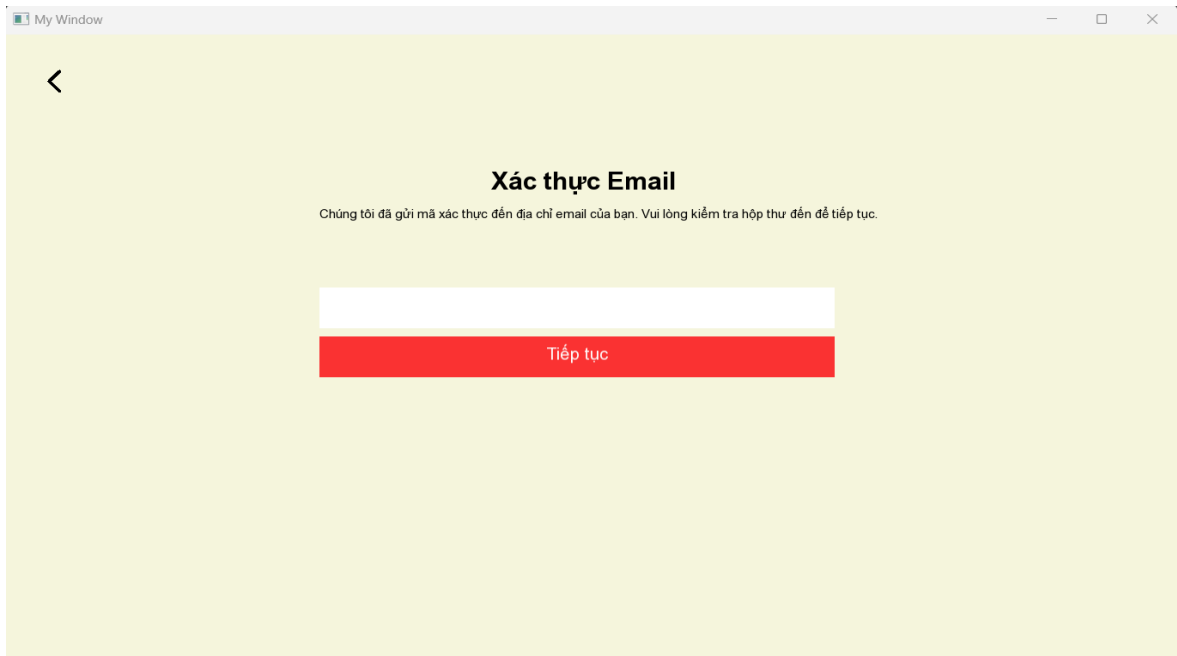
- Người dùng kiểm tra gmail và nhập chính xác mã OTP (6 chữ số) vào và bấm vào nút *Tiếp tục*.



*Hình 5.2-4. Giao diện nhập mã OTP*


### c. Giao diện thay đổi mật khẩu mới

- Người dùng nhập mật khẩu mới và nhập lại chính xác thêm một lần nữa để xác nhận mật khẩu mới. Tiếp theo chọn vào nút *Xác nhận*. Người dùng sẽ được đưa về màn hình đăng nhập để đăng nhập lại tài khoản sau khi thay đổi mật khẩu.

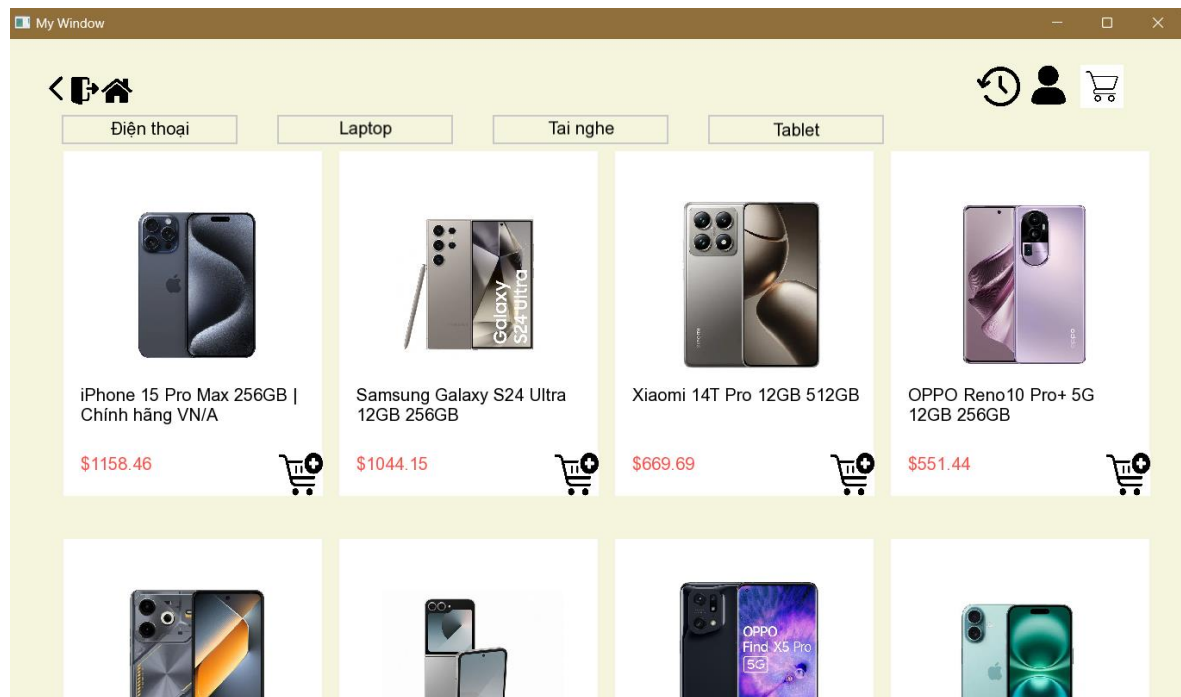


Hình 5.2-5. Giao diện nhập mật khẩu mới

### 4. Giao diện chính của ứng dụng

- Sau khi đăng nhập thành công, giao diện hiển thị màn hình có các sản phẩm, bên trên có thể loại của sản phẩm, chọn vào loại nào thì sẽ hiển thị sản phẩm tương ứng với loại sản phẩm đó.
- Trong màn hình sản phẩm này còn có các tùy chọn ở bên trên: Home, Back, Log out, Lịch sử, Thông tin tài khoản, Xem giỏ hàng.
- Khi chọn vào nút *Thêm vào giỏ hàng*  ở mỗi khối sản phẩm thì sản phẩm đó sẽ được thêm vào giỏ hàng.

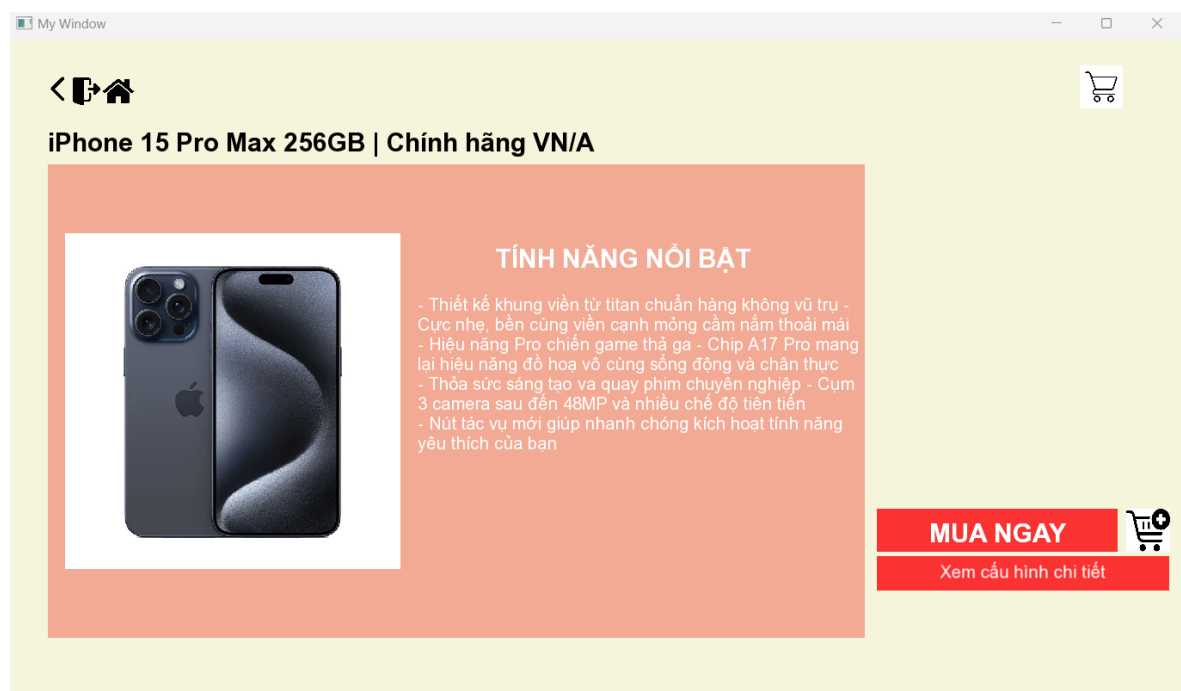
## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ



Hình 5.2-6. Giao diện chính của ứng dụng

### 5. Giao diện chi tiết của sản phẩm

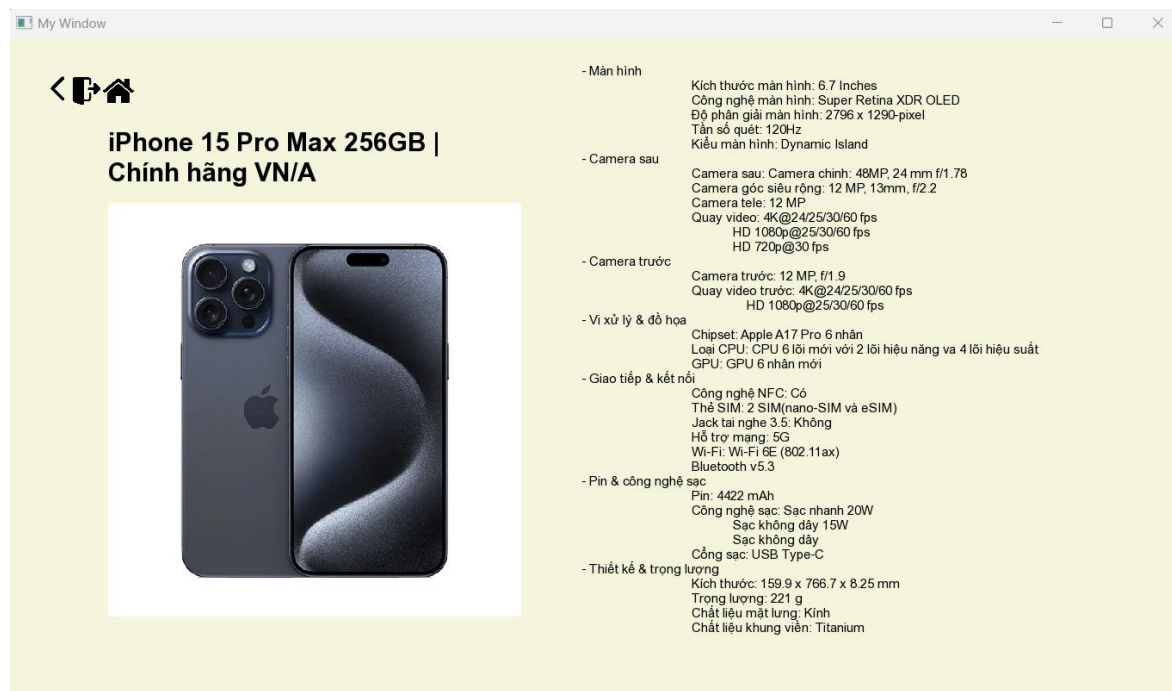
- Khi chọn vào một sản phẩm nào đó, màn hình chi tiết của sản phẩm đó sẽ được hiển thị.
- Trong này sẽ có tính năng nổi bật của sản phẩm, nút *Mua ngay*, *Thêm vào giỏ hàng*, *Xem cấu hình chi tiết*.
- Khi click vào nút *Mua ngay* thì sẽ được đưa đến màn hình chứa các sản phẩm trong giỏ hàng.



Hình 5.2-7. Giao diện chi tiết của sản phẩm

### 6. Giao diện xem cấu hình chi tiết của sản phẩm

- Ở màn hình chi tiết của sản phẩm, khi người dùng chọn vào nút *Xem cấu hình chi tiết* thì giao diện màn hình xem thông số kỹ thuật chi tiết của sản phẩm sẽ hiển thị.

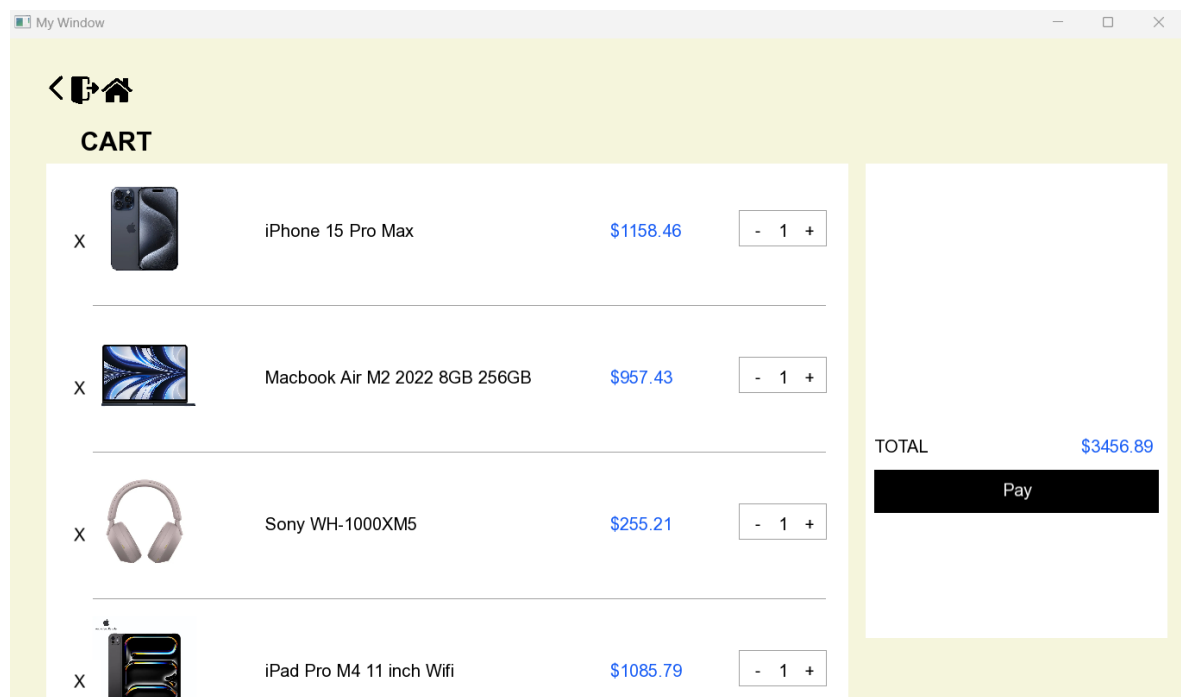


Hình 5.2-8. Giao diện thông số kỹ thuật của sản phẩm

### 7. Giao diện giỏ hàng

- Trong giao diện giỏ hàng, các sản phẩm được người dùng thêm vào sẽ xuất hiện ở đây. Mỗi sản phẩm sẽ có số lượng sản phẩm, *tăng/giảm* số lượng và *xóa sản phẩm* “X” ra khỏi giỏ hàng.
- Bên phải sẽ có tổng số tiền của các sản phẩm trong giỏ hàng. Ngoài ra còn có nút *Pay* để thanh toán các đơn hàng.

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ



Hình 5.2-9. Giao diện giỏ hàng

### 8. Giao diện thanh toán

- Khi chọn vào nút *Pay* ở giao diện giỏ hàng, màn hình thanh toán sẽ hiển thị.
- Tại đây, người dùng được yêu cầu nhập địa chỉ đơn hàng, và mã giảm giá (nếu có). Sau khi nhập đủ thông tin thì chọn vào nút *Pay* để xác nhận thanh toán.
- Nếu người dùng không muốn thanh toán, người dùng có thể chọn vào nút *Home* để quay về màn hình sản phẩm, chọn vào nút *Cart* hoặc nút *Back* để quay về giỏ hàng.

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

Payment information

Lương Văn Võ (luongvanvo10@gmail.com)  
Log out

Lương Văn Võ

0865321921

Address:

Cart Pay

iPhone 15 Pro Max x1 \$1158.46

Macbook Air M2 2022 8GB 256GB x1 \$957.43

Sony WH-1000XM5 x1 \$255.21

iPad Pro M4 11 inch Wifi x1 \$1085.79

Discount

Use

Price product \$3456.89

Discount - \$0

TOTAL \$3456.89

Hình 5.2-10. Giao diện thanh toán

### 9. Giao diện xem lịch sử đơn hàng

- Người dùng chọn vào nút *Xem lịch sử đơn hàng* Giao diện màn hình xem lịch sử sẽ hiển thị.
- Tại đây, người dùng cần nhập mã đơn hàng để kiểm tra chi tiết đơn hàng. Sau khi nhập theo yêu cầu thì người dùng chọn vào nút *Xem ngay* để xem các sản phẩm trong đơn hàng.

Chi tiết lịch sử đơn hàng

Kiểm tra đơn hàng

Số điện thoại:  
0865321921

Mã đơn hàng:  
100021


Nếu quý khách có bất kỳ thắc mắc nào, xin vui lòng gọi 0865321921

Xem ngay

STT	Tên sản phẩm	Tiền	Số lượng
1	iPhone 15 Pro Max	\$1158.46	1
2	Samsung Galaxy S24 Ultra	\$1044.15	1

Hình 5.2-11. Giao diện xem lịch sử đơn hàng

### 10. Giao diện thông tin tài khoản

- Người dùng chọn vào nút *Xem thông tin tài khoản*  để xem thông tin của tài khoản.
- Tại đây, hiển thị các thông tin của người dùng và thông tin các đơn hàng mà người dùng đã thanh toán.
- Người dùng có thể chọn vào nút *Log out* để đăng xuất tài khoản hoặc nút *Change password* để thay đổi mật khẩu.



Hình 5.2-12. Giao diện thông tin tài khoản

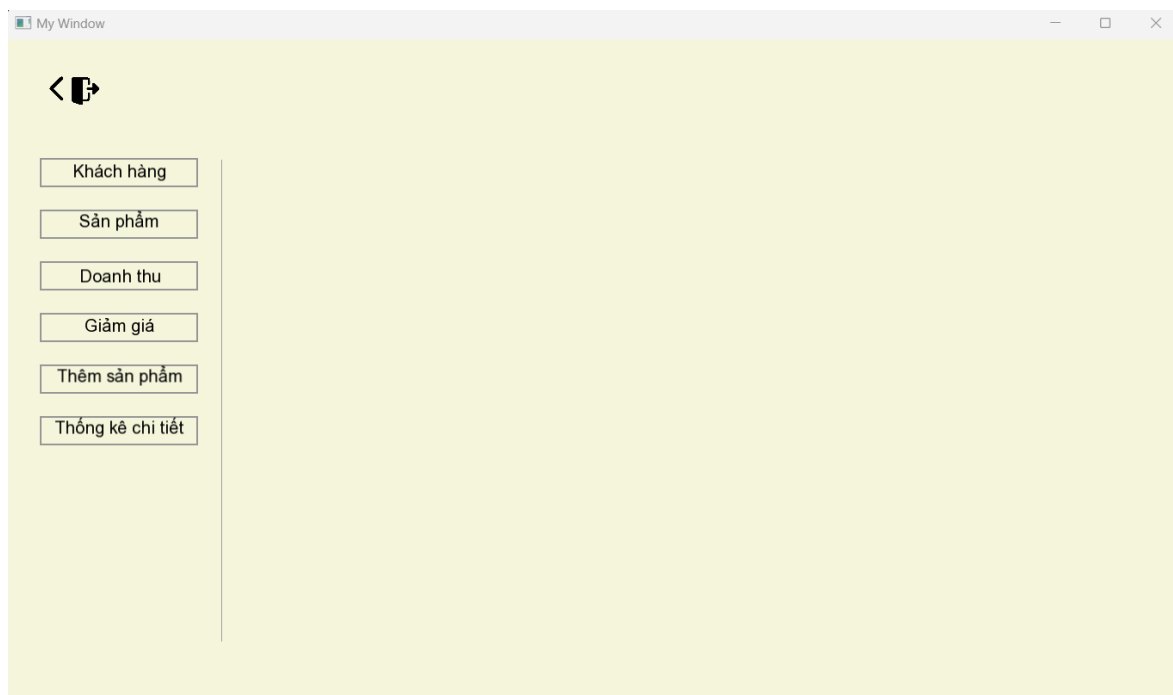
### 5.2.2. Giao diện quản trị viên

#### 1. Giao diện chính của ứng dụng

- Quản trị viên (admin) đăng nhập vào hệ thống với tài khoản đặc biệt được cấp riêng. Khi đăng nhập thành công, giao diện hiển thị thanh bên trái với các tùy chọn: *Khách hàng, Sản phẩm, Doanh thu, Giảm giá, Thêm sản phẩm, Thống kê chi tiết*.




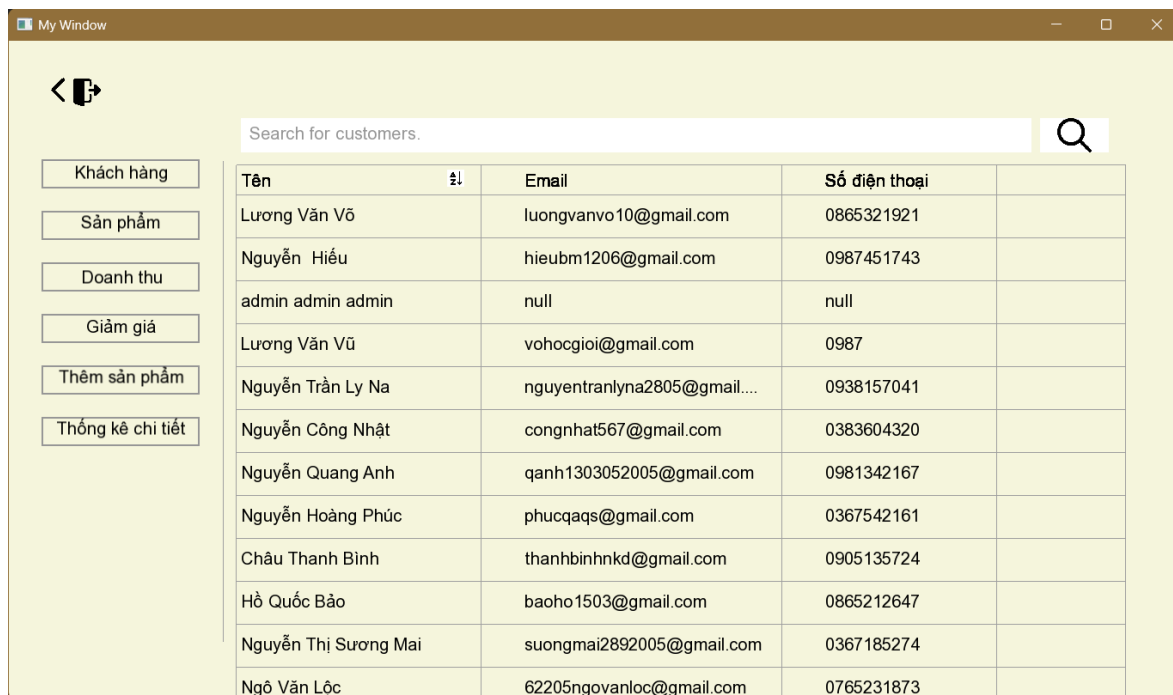
## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ



Hình 5.2-13. Giao diện chính

### 2. Giao diện hiển thị danh sách khách hàng

- Chọn vào ô *Khách hàng* để hiển thị giao diện để xem danh sách khách hàng.
- Ngoài ra, còn có thể tìm kiếm khách hàng khi nhập tên của khách hàng vào ô tìm kiếm, hoặc có thể sắp xếp tên sản phẩm từ A-Z hoặc từ Z-A khi chọn vào ô 

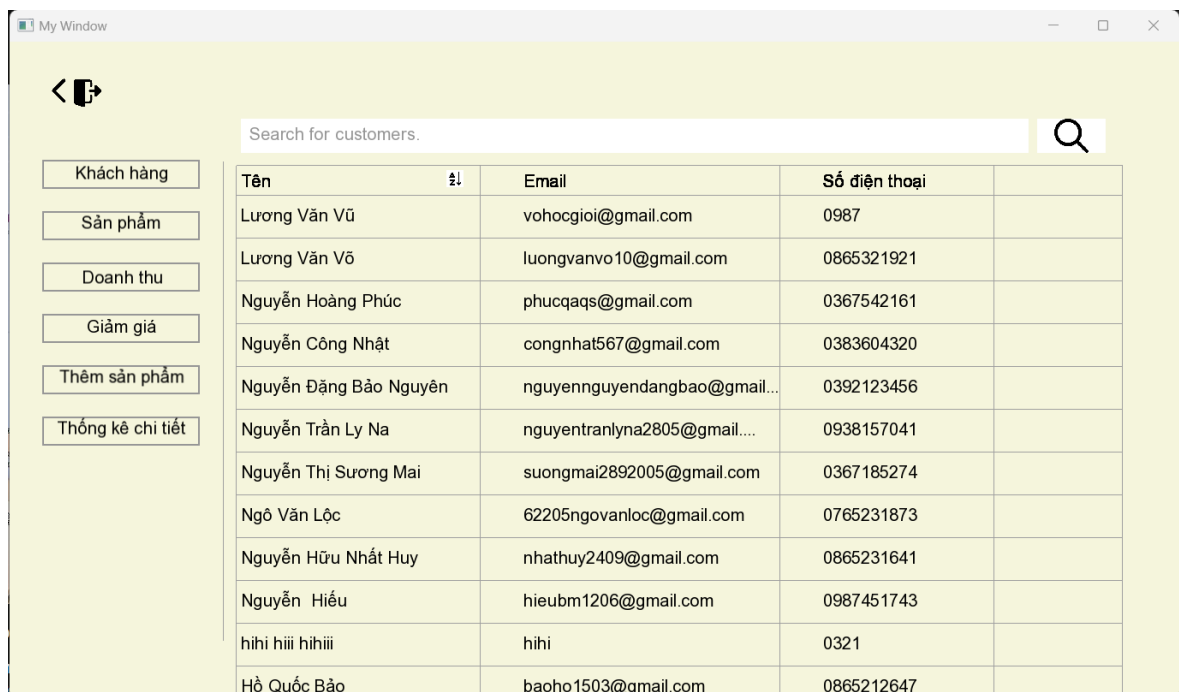


Hình 5.2-14. Giao diện danh sách tất cả khách hàng

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ



Hình 5.2-15. Giao diện tìm kiếm tên khách hàng



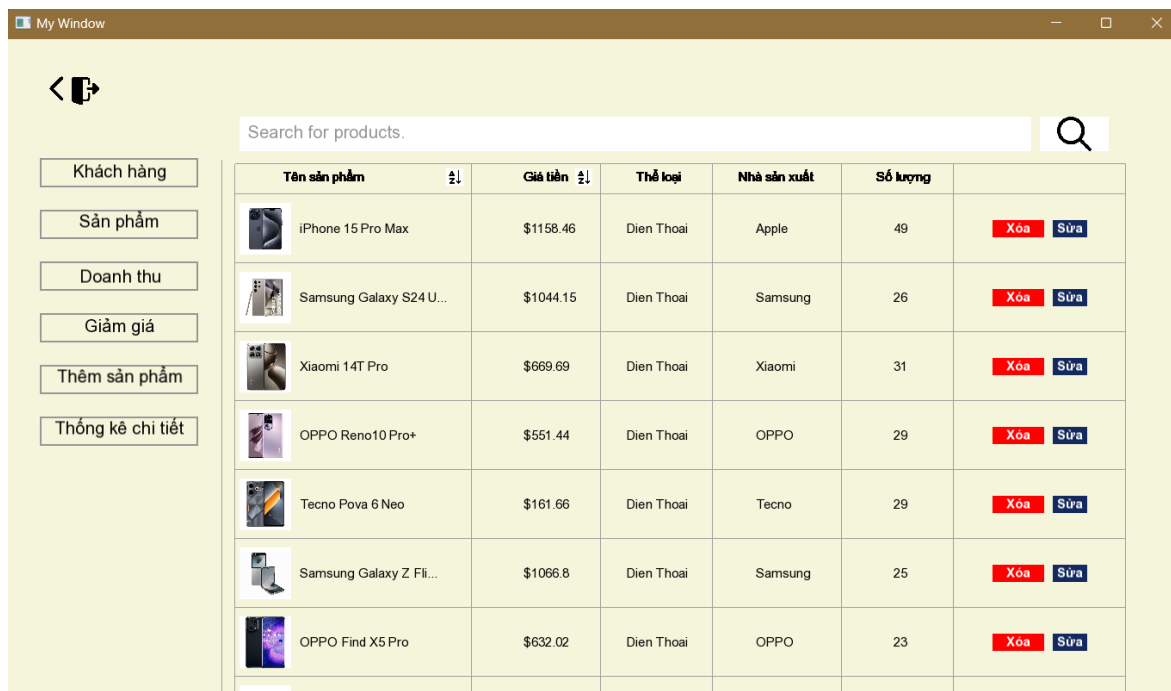
Hình 5.2-16. Giao diện sắp xếp tên khách hàng

### 3. Giao diện hiển thị danh sách sản phẩm

- Chọn vào ô *Sản phẩm* để hiển thị giao diện để xem danh sách sản phẩm.
- Ngoài ra, còn có thể tìm kiếm sản phẩm khi nhập tên của sản phẩm vào ô tìm kiếm, hoặc có thể sắp xếp tên sản phẩm từ A-Z hoặc từ Z-A khi chọn vào ô

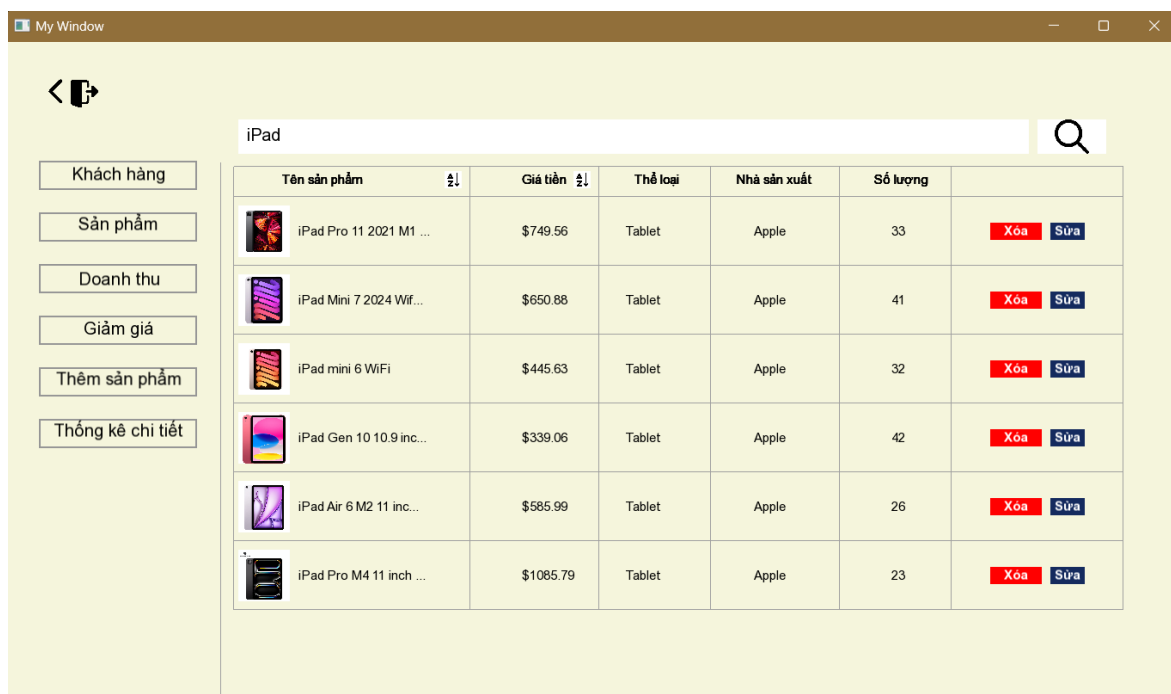
## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

- Quản trị viên còn có thể sửa thông tin của mỗi sản phẩm khi chọn vào ô *Sửa* hoặc xóa sản phẩm ra khỏi hệ thống khi chọn vào ô *Xóa*.
- Quản trị viên có thể sắp xếp danh sách theo giá tiền sản phẩm từ thấp đến cao hoặc từ cao đến thấp khi chọn vào ô  $\updownarrow$



Tên sản phẩm	Giá tiền	Thế loại	Nhà sản xuất	Số lượng	
iPhone 15 Pro Max	\$1158.46	Điện Thoại	Apple	49	Xóa Sửa
Samsung Galaxy S24 U...	\$1044.15	Điện Thoại	Samsung	26	Xóa Sửa
Xiaomi 14T Pro	\$669.69	Điện Thoại	Xiaomi	31	Xóa Sửa
OPPO Reno10 Pro+	\$551.44	Điện Thoại	OPPO	29	Xóa Sửa
Tecno Pova 6 Neo	\$161.66	Điện Thoại	Tecno	29	Xóa Sửa
Samsung Galaxy Z Fl...	\$1066.8	Điện Thoại	Samsung	25	Xóa Sửa
OPPO Find X5 Pro	\$632.02	Điện Thoại	OPPO	23	Xóa Sửa

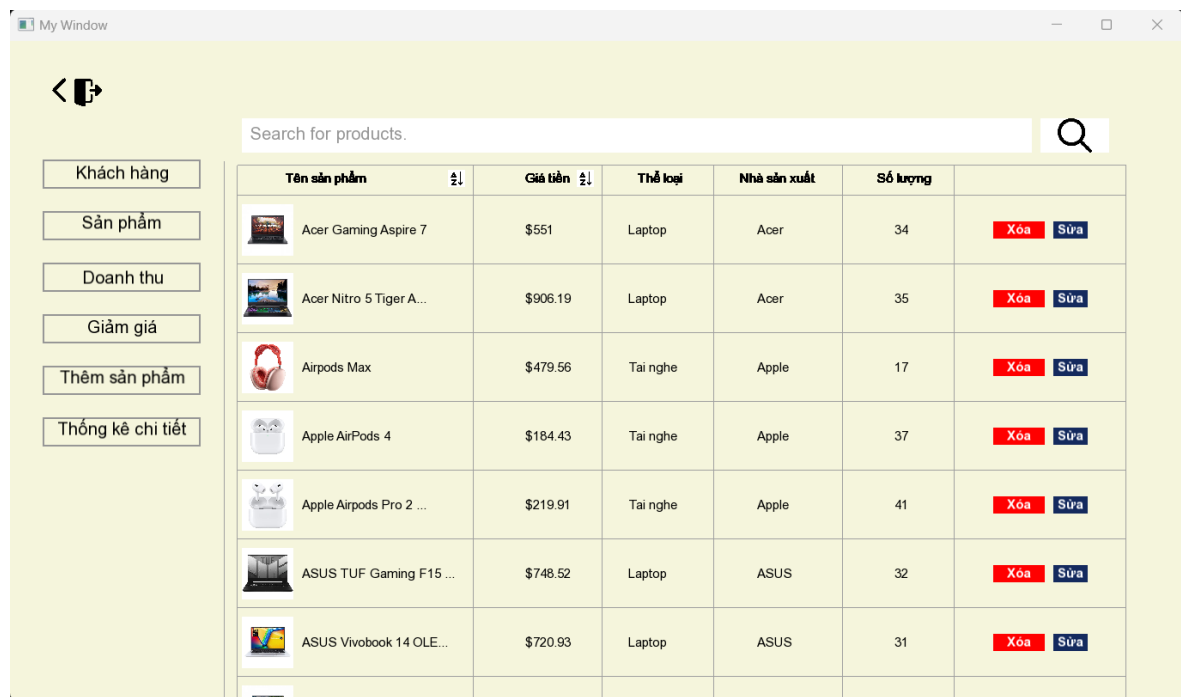
Hình 5.2-17. Giao diện danh sách tất cả sản phẩm



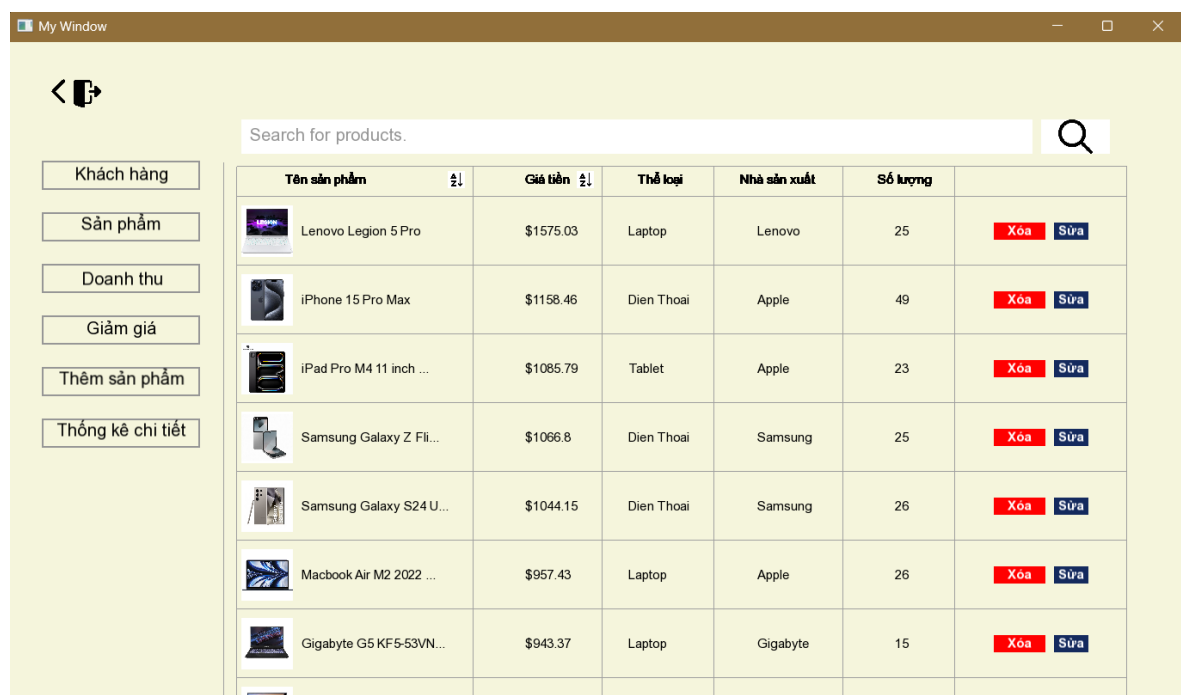
Tên sản phẩm	Giá tiền	Thế loại	Nhà sản xuất	Số lượng	
iPad Pro 11 2021 M1 ...	\$749.56	Tablet	Apple	33	Xóa Sửa
iPad Mini 7 2024 Wif...	\$650.88	Tablet	Apple	41	Xóa Sửa
iPad mini 6 WiFi	\$445.63	Tablet	Apple	32	Xóa Sửa
iPad Gen 10 10.9 inc...	\$339.06	Tablet	Apple	42	Xóa Sửa
iPad Air 6 M2 11 inc...	\$585.99	Tablet	Apple	26	Xóa Sửa
iPad Pro M4 11 inch ...	\$1085.79	Tablet	Apple	23	Xóa Sửa

Hình 5.2-18. Giao diện tìm kiếm tên sản phẩm

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ



Hình 5.2-19. Giao diện sắp xếp sản phẩm theo tên sản phẩm



Hình 5.2-20. Giao diện sắp xếp sản phẩm theo giá sản phẩm

### 4. Giao diện chỉnh sửa thông tin sản phẩm

- Khi người dùng nhấn nút *Sửa* trên từng sản phẩm trong giao diện danh sách sản phẩm, giao diện chỉnh sửa sản phẩm tương ứng sẽ được hiển thị.
- Hệ thống yêu cầu quản trị viên nhập đầy đủ thông tin yêu cầu của sản phẩm để cập nhật thông tin sản phẩm trên hệ thống.

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

- Sau khi nhập thì nhấn vào nút *Xác nhận* để lưu thông tin thay đổi.



CHỈNH SỬA THÔNG TIN SẢN PHẨM

Tên sản phẩm:

Thẻ loại:

Giá tiền:

Nhà sản xuất:

Hệ điều hành:

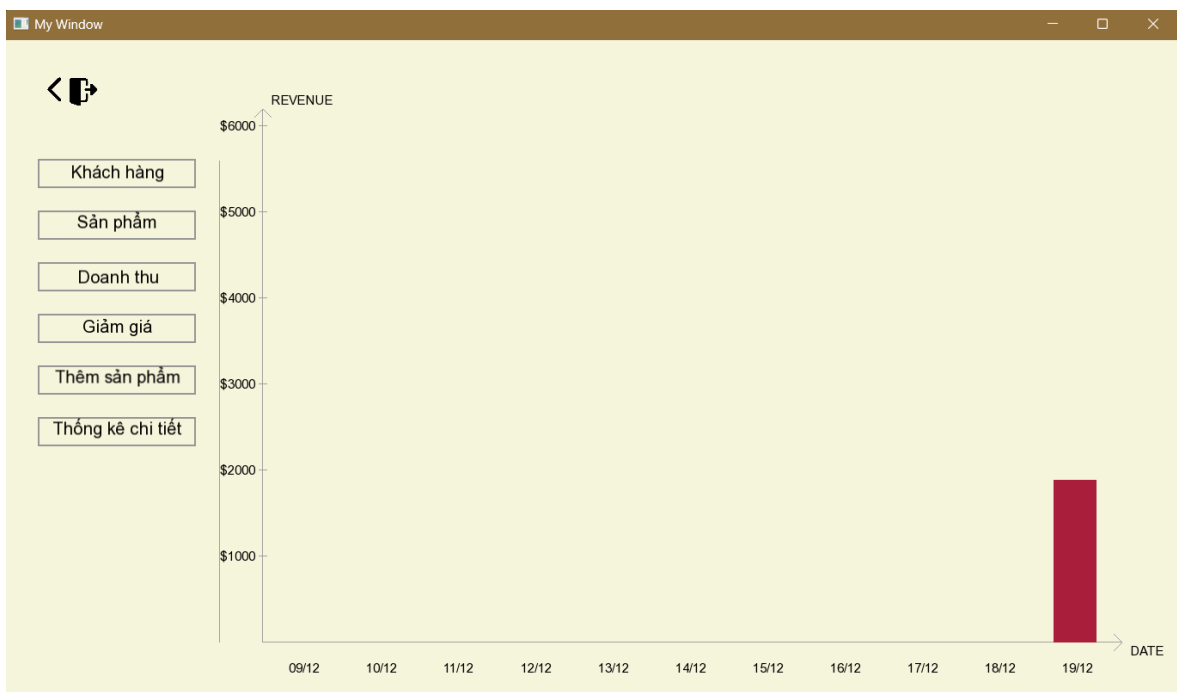
Số lượng:

Xác nhận

Hình 5.2-21. Giao diện chỉnh sửa thông tin sản phẩm

### 5. Giao diện xem doanh thu

- Khi nhấn vào ô *Doanh thu*, bạn sẽ xem được doanh thu của hệ thống dưới dạng biểu đồ cột cho những ngày gần nhất.



Hình 5.2-22. Giao diện xem doanh thu

### 6. Giao diện tạo mã giảm giá

- Khi nhấn vào ô *Giảm giá*, giao diện tạo mã giảm giá sẽ được hiển thị, nơi quản trị viên có thể cấp mã giảm giá.
- Để cấp mã giảm giá, quản trị viên cần nhập đủ các thông tin: *Mã giảm giá*, *Số tiền được giảm*, *Số lượng mã*. Sau đó nhấn vào ô *Cấp* để hoàn thành việc tạo ra mã giảm giá cho khách hàng.

The screenshot shows a web application window titled "My Window". On the left side, there is a vertical sidebar with several buttons: "Khách hàng", "Sản phẩm", "Doanh thu", "Giảm giá" (which is highlighted), "Thêm sản phẩm", and "Thống kê chi tiết". The main content area has a light yellow background and is titled "TẠO MÃ GIẢM GIÁ" in bold. Below the title, there are three input fields with labels: "Mã giảm giá:", "Tiền:", and "Số lượng mã:". To the right of these fields is a red button labeled "Cấp".

Hình 5.2-23. Giao diện tạo mã giảm giá

### 7. Giao diện thêm sản phẩm

- Khi nhấn vào ô *Thêm sản phẩm*, giao diện thêm sản phẩm sẽ được hiển thị, nơi quản trị viên có thể thêm sản phẩm vào hệ thống.
- Quản trị viên cần nhập đầy đủ thông tin cần thiết của sản phẩm, sau đó nhấn vào nút *Xác nhận* để hoàn thành việc thêm sản phẩm vào hệ thống.

## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ

My Window

<

**THÊM SẢN PHẨM.**

Khách hàng

Sản phẩm

Doanh thu

Giảm giá

Thêm sản phẩm

Thống kê chi tiết

File sản phẩm:

File ảnh sản phẩm:

Mã sản phẩm:

Thẻ loại:

Tên sản phẩm:

Giá sản phẩm:

Nhà sản xuất:

Hệ điều hành:

Số lượng:

Xác nhận

Hình 5.2-24. Giao diện thêm sản phẩm

### 8. Giao diện thống kê doanh số theo từng ngày

- Khi nhấn vào ô *Thống kê chi tiết*, giao diện thống kê chi tiết doanh số sẽ được hiển thị.
- Quản trị viên muốn xem doanh số của ngày nào thì nhập vào ô tìm kiếm với mẫu dd-mm-yyyy để tìm kiếm. Thông tin các đơn hàng sẽ được hiển thị dưới dạng danh sách.

My Window

<

**THỐNG KÊ CHI TIẾT DOANH SỐ**

18-12-2024

STT

Mã khách hàng

Mã đơn hàng

Địa chỉ

Thành tiền

Không có đơn hàng nào.

Hình 5.2-25. Giao diện thống kê doanh thu khi không có đơn hàng trong ngày



## PBL 2: DỰ ÁN LẬP TRÌNH CƠ SỞ



STT	Mã khách hàng	Mã đơn hàng	Địa chỉ	Thành tiền
1	10	100039	55 Lý Thái Tông, Đà Nẵng	\$921.69
2	2	100040	10 Hồ Tùng Mậu, Đà Nẵng	\$668.77
3	2	100041	10 Hồ Tùng Mậu, Đà Nẵng	\$168.7
4	2	100042	14 Phan Châu Trinh, Đà Nẵng	\$101.89

Hình 5.2-26. Giao diện thống kê doanh thu khi có đơn hàng trong ngày

### 5.3. Nhận xét và đánh giá

Xây dựng được chương trình hệ thống mua bán và quản lý thiết bị điện tử, đáp ứng được các chức năng cơ bản của người dùng

Đáp ứng được yêu cầu của đề tài:

- Về kiến thức: áp dụng các kiến thức đã học trong môn lập trình hướng đối tượng, cơ sở dữ liệu, cấu trúc dữ liệu và giải thuật
- Về chức năng và kỹ thuật
  - o Xây dựng chương trình có khả năng tương tác và lưu trữ dữ liệu trong các file txt.
  - o Thiết kế giao diện người dùng trực quan, đơn giản, dễ sử dụng, phù hợp với mọi đối tượng.
  - o Đảm bảo chương trình hỗ trợ gần như đầy đủ các chức năng cần thiết của một hệ thống quản lý mua bán bao gồm:
    - Thêm, sửa, xóa, tìm kiếm và sắp xếp thông tin thiết bị
    - Quản lý danh sách khách hàng và đơn hàng
    - Thống kê và báo cáo doanh thu
  - o Đảm bảo chương trình hoạt động ổn định, xử lý nhanh và không có lỗi nghiêm trọng trong quá trình vận hành.

## 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1. Kết luận

#### 6.1.1. Đạt được

Hệ thống mua bán và quản lý thiết bị điện tử đã xây dựng thành công, đáp ứng được nhiều chức năng cốt lõi như thêm, chỉnh sửa, tìm kiếm, sắp xếp, xóa và hiển thị thông tin sản phẩm và khách hàng. Những chức năng này giúp cải thiện đáng kể việc quản lý kho hàng, giao dịch mua bán và thông tin khách hàng, đảm bảo tính chính xác và minh bạch của dữ liệu. Nhờ đó, hệ thống không chỉ giảm tải được công việc thủ công mà còn hỗ trợ hiệu quả trong việc tối ưu hóa vận hành và quản lý.

Việc áp dụng mô hình lập trình hướng đối tượng (OOP) là một điểm sáng trong dự án này. Hệ thống sử dụng các lớp chính như **Product**, **Customer**, **Admin**, **Order** và **History** cùng các phương thức liên quan, cho phép mã nguồn được tổ chức rõ ràng, dễ bảo trì và tái sử dụng. OOP giúp mở rộng tính năng dễ dàng mà không ảnh hưởng đến các phần đã triển khai. Ví dụ, việc sử dụng các **hàm bạn bè** (friend functions) để truy cập các thuộc tính riêng tư, cũng như tích hợp các **hàm tĩnh** như `getProductID()` hoặc `getCustomerID()`, góp phần đảm bảo tính toàn vẹn và hiệu quả của dữ liệu.

Một điểm nổi bật khác là hệ thống hỗ trợ chức năng tìm kiếm mạnh mẽ, cho phép quản trị viên tra cứu thông tin sản phẩm hoặc khách hàng theo tên sản phẩm hoặc khách hàng. Chức năng này giúp nâng cao khả năng quản lý kho hàng, chăm sóc khách hàng và lập kế hoạch bán hàng, đồng thời đảm bảo dữ liệu được truy xuất nhanh chóng và chính xác. Bên cạnh đó, hệ thống còn tích hợp chức năng quản lý tài khoản người dùng, với hai phân quyền chính là **Admin** và **User**, tạo sự linh hoạt và an toàn trong việc sử dụng.

Với những cải tiến vượt bậc này, hệ thống quản lý và mua bán thiết bị điện tử không chỉ đáp ứng được các yêu cầu hiện tại mà còn sẵn sàng mở rộng trong tương lai, góp phần nâng cao hiệu quả quản lý và phát triển trong tương lai.

#### 6.1.2. Chưa đạt được

Mặc dù hệ thống quản lý và mua bán thiết bị điện tử đã đạt được nhiều kết quả tích cực, vẫn còn một số hạn chế cần khắc phục để đáp ứng tốt hơn nhu cầu phát triển trong tương lai:

- Chưa tích hợp với cơ sở dữ liệu lớn hoặc các giải pháp lưu trữ: hiện tại, hệ thống sử dụng tệp văn bản (text file) để lưu trữ dữ liệu. Điều này

không chỉ làm giảm hiệu suất mà còn hạn chế khả năng mở rộng khi dữ liệu ngày càng lớn. Việc xử lý và truy xuất dữ liệu lớn dựa trên tệp văn bản có thể gây chậm trễ và khó khăn trong quản lý.

- Thiếu các chức năng phân tích và thống kê chi tiết: Hệ thống hiện chưa hỗ trợ các báo cáo chi tiết về hiệu suất bán hàng, doanh thu theo từng sản phẩm hoặc khoảng thời gian cụ thể, cũng như phân tích hành vi mua sắm của khách hàng. Những chức năng này rất cần thiết để hỗ trợ quản lý đưa ra các quyết định chiến lược và tối ưu hóa hoạt động kinh doanh.
- Chưa tích hợp các tính năng bảo mật nâng cao: Hệ thống chưa triển khai các cơ chế mã hóa dữ liệu hoặc quản lý lịch sử truy cập, dẫn đến nguy cơ mất mát hoặc rò rỉ thông tin quan trọng nếu xảy ra sự cố hoặc bị tấn công bảo mật. Việc bổ sung các tính năng như quản lý lịch sử hoạt động của tài khoản, phân quyền chi tiết hơn, hoặc sử dụng giao thức bảo mật sẽ nâng cao mức độ an toàn của hệ thống.
- Vấn đề về tính thẩm mỹ còn hạn chế và thiếu các thông báo cho người dùng, gây khó khăn trong việc sử dụng.
- Chưa triển khai các tính năng hỗ trợ khách hàng và quản lý hiện đại:
  - Chức năng theo dõi trạng thái đơn hàng theo thời gian thực vẫn chưa được tích hợp, làm giảm trải nghiệm khách hàng
  - Các tính năng quản lý kho nâng cao như dự báo tồn kho hoặc tối ưu hóa quy trình nhập hàng chưa được triển khai, gây khó khăn khi quy mô kinh doanh mở rộng
  - Hệ thống chưa hỗ trợ tích hợp với các nền tảng di động hoặc website bán hàng, hạn chế khả năng tiếp cận và phục vụ khách hàng trực tuyến

### 6.2. Hướng phát triển

- Nâng cấp lưu trữ:
  - Chuyển từ lưu trữ bằng file txt sang cơ sở dữ liệu như MySQL, PostgreSQL, ....
- Phát triển ứng dụng:
  - Xây dựng ứng dụng trên các nền tảng chuyên nghiệp hơn như: web, mobile (iOS, Android) để đáp ứng nhu cầu đa dạng của người dùng.
  - Phát triển ứng dụng với nhiều tính năng hơn để có thể hỗ trợ tối đa cho người dùng nhằm nâng cao trải nghiệm và quản trị viên thuận tiện hơn trong việc quản lý hệ thống.

## TÀI LIỆU THAM KHẢO

- [1] Giáo trình lập trình hướng đối tượng.
- [2] *Introduction to algorithms* by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.
- [3] SFML. (n.d.). *SFML: Simple and Fast Multimedia Library*. Truy cập ngày 21/10/2024 tại <https://www.sfm1-dev.org/>
- [4] cURL. (n.d.). *cURL: Transfer data with URL syntax*. Truy cập ngày 15/11/2024 tại <https://curl.se/>