

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



PBL2: DỰ ÁN CƠ SỞ LẬP TRÌNH
TÊN ĐỀ TÀI : HỆ THỐNG QUẢN LÍ
CỦA HÀNG ĐIỆN THOẠI

Giảng viên hướng dẫn : TS. ĐẶNG HOÀNG PHƯƠNG
Sinh viên thực hiện: NGUYỄN VĂN TRUNG ÁNH
ĐOÀN KIỀU NGÂN

Lớp : 23T_DT1
Khoa : 2023

Đà Nẵng, tháng 1 năm 2025

LỜI MỞ ĐẦU

Trong thời đại công nghệ phát triển vượt bậc, việc ứng dụng công nghệ thông tin vào các hoạt động kinh doanh đã trở thành xu hướng không thể thiếu. Đặc biệt, trong lĩnh vực bán lẻ, sự phổ biến của các nền tảng trực tuyến đã thay đổi cách khách hàng tiếp cận sản phẩm và dịch vụ. Khách hàng ngày nay ưu tiên các giải pháp mua sắm tiện lợi, nhanh chóng và tiết kiệm thời gian, đồng thời các doanh nghiệp cũng cần những công cụ hiện đại để quản lý và vận hành hiệu quả hơn.

Xuất phát từ nhu cầu thực tiễn này, chúng em đã xây dựng một hệ thống quản lý cửa hàng điện thoại nhằm hỗ trợ toàn diện cho các hoạt động kinh doanh. Hệ thống cung cấp các chức năng chính như quản lý thông tin sản phẩm, khách hàng, hóa đơn và các giao dịch. Ngoài ra, hệ thống còn giúp cập nhật chi tiết về các dòng điện thoại mới nhất, hỗ trợ tìm kiếm và phân loại sản phẩm dễ dàng, đảm bảo việc phục vụ khách hàng diễn ra nhanh chóng và chính xác.

Không chỉ dừng lại ở việc quản lý, hệ thống còn hướng đến việc nâng cao trải nghiệm người dùng, giúp cửa hàng tối ưu hóa quy trình vận hành và nâng cao chất lượng dịch vụ. Đây không chỉ là một công cụ hỗ trợ bán hàng, mà còn là nền tảng giúp các cửa hàng điện thoại bắt kịp xu hướng hiện đại hóa trong kinh doanh.

Dù đã nỗ lực hoàn thiện, nhưng do thời gian và kinh nghiệm còn hạn chế, chúng em hy vọng nhận được sự đóng góp và chỉ dẫn từ quý thầy cô và các bạn để hệ thống có thể hoàn thiện hơn nữa, đáp ứng tốt hơn yêu cầu thực tiễn.

Cuối cùng, chúng em xin bày tỏ lòng biết ơn sâu sắc đến giảng viên hướng dẫn **Đặng Hoài Phương**, người đã tận tình chỉ dẫn, cung cấp kiến thức và động viên chúng em trong suốt quá trình thực hiện đề tài này. Thầy đã không ngừng hỗ trợ, giải đáp các thắc mắc, đưa ra những định hướng quý báu, giúp chúng em hoàn thành đề tài một cách hiệu quả nhất.

Chúng em cũng xin gửi lời cảm ơn chân thành đến toàn thể quý thầy cô trong khoa **Công nghệ Thông tin**, những người đã truyền đạt những kiến thức nền tảng và chuyên sâu trong suốt thời gian học tập. Sự dùn dắt và những kinh nghiệm quý báu từ thầy cô chính là động lực và hành trang quý giá để chúng em hoàn thành tốt bài nghiên cứu này, cũng như chuẩn bị cho những thử thách trong tương lai.

MỤC LỤC

LỜI MỞ ĐẦU DANH MỤC HÌNH VẼ

1. TỔNG QUAN ĐỀ TÀI	1
2. CƠ SỞ LÝ THUYẾT	2
3. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG	2
3.1 Phân tích chức năng xác thực và phân quyền người dùng:	2
3.2 Phân tích chức năng phía khách hàng:	3
3.2.1 Tạo tài khoản mới (nếu chưa có tài khoản):	3
3.2.2 Đăng nhập vào tài khoản (đã có tài khoản):	4
3.2.3 Tìm kiếm sản phẩm trong cửa hàng:	4
3.2.4 Thêm, xóa sản phẩm trong giỏ hàng:	4
3.2.5 Xem giỏ hàng :	5
3.2.6 Đặt đơn hàng, chọn ngày giao, phương thức thanh toán:	5
3.2.7 Xem danh sách đơn hàng đã mua:	7
3.2.8 Xem thông tin, thay đổi thông tin tài khoản cá nhân	8
3.2.9. Xem các thông số thống kê	9
3.2.9.a) Thống kê tổng số hoá đơn, số tiền	9
3.2.9.b) Biểu đồ số tiền trên mỗi hoá đơn:	10
3.2.10 Đăng xuất	11
3.3 Phân tích chức năng phía quản lý:	11
3.3.1 Đăng nhập vào tài khoản quản lý	11
3.3.2 Xem danh sách sản phẩm trong cửa hàng	11
3.3.3 Xoá sản phẩm khỏi cửa hàng:	12
3.3.4 Thay đổi thông tin, số lượng sản phẩm	13
3.3.5 Xem danh sách khách hàng	15
3.3.6 Xem danh sách hoá đơn của cửa hàng	16
3.3.7 Xem thay đổi thông tin tài khoản cá nhân	17
3.3.8 Thống kê tổng số hoá đơn, số tiền, khách hàng	19
3.3.9 Biểu đồ cột các thống kê:	20
4. THIẾT KẾ CẤU TRÚC DỮ LIỆU	21
4.1 Phát biểu bài toán:	21
4.2 Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống:	23
5. ỨNG DỤNG HƯỚNG ĐỐI TƯỢNG VÀO BÀI TOÁN	24
5.1 Phân tích các lớp và đối tượng chính trong hệ thống	24
5.1.1 Chi tiết các lớp chính trong chương trình:	24
5.1.2 Chi tiết các lớp khác hỗ trợ :	25
5.2 Phân tích mối liên hệ giữa các lớp và đối tượng:	27
6. CHƯƠNG TRÌNH VÀ KẾT QUẢ	27
6.1 Tổ chức chương trình :	27
6.3 Nhận xét và đánh giá :	46
7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	47
7.1 Kết luận :	47
7.2 Hướng phát triển :	47
8. PHỤ LỤC	48
9. DANH MỤC TÀI LIỆU THAM KHẢO	48

DANH MỤC HÌNH VẼ

Hình 1 . Giao diện đăng nhập _____	28
Hình 2 . Giao diện đăng nhập không thành công _____	28
Hình 3 . Giao diện đăng kí thành công _____	29
Hình 4 . Thông báo đăng kí không thành công _____	29
Hình 5 . Giao diện chính cửa hàng _____	30
Hình 6 . Xem thông tin chi tiết sản phẩm _____	30
Hình 7 . Tìm kiếm sản phẩm theo tên sản phẩm _____	31
Hình 8 . Thêm sản phẩm vào giỏ hàng (không thành công) _____	31
Hình 9 . Thêm sản phẩm vào giỏ hàng (thành công) _____	32
Hình 10 . Thêm, giảm, xóa sản phẩm khỏi giỏ hàng: _____	32
Hình 11 . Chọn ngày giao hàng, phương thức thanh toán và đặt hàng _____	33
Hình 12 . Xem tất cả đơn hàng đã đặt _____	33
Hình 13 . Xem chi tiết hóa đơn _____	34
Hình 14 . Xem thông tin tài khoản _____	34
Hình 15 . Thay đổi thông tin tài khoản _____	35
Hình 16 . Xem tổng quan , thống kê _____	35
Hình 17 . Giao diện đăng nhập (manager) _____	36
Hình 18 . Giao diện tổng quan, thống kê sản phẩm bán được _____	36
Hình 19 . Giao diện thống kê số tiền kiếm được theo ngày _____	37
Hình 20 . Giao diện thống kê tên khách hàng theo số tiền đã mua _____	37
Hình 21 . Xem danh sách các sản phẩm đã bán _____	38
Hình 22 . Xem danh sách các sản phẩm trong cửa hàng _____	38
Hình 23 . Tìm kiếm sản phẩm theo tên _____	39
Hình 24 . Thêm sản phẩm mới vào cửa hàng _____	39
Hình 25 . Thêm sản phẩm thành công _____	40
Hình 26 . Xóa sản phẩm khỏi cửa hàng _____	40
Hình 27 . Xóa sản phẩm thành công _____	41
Hình 28 . Chính sửa thông tin sản phẩm trong cửa hàng _____	41
Hình 29 . Thay đổi thông tin sản phẩm _____	42
Hình 30 . Thay đổi thông tin thành công _____	42
Hình 31 . Xem danh sách các khách hàng của cửa hàng _____	43
Hình 32 . Tìm kiếm theo tên khách hàng _____	43
Hình 33 . Xóa thông tin khách hàng của cửa hàng _____	44
Hình 34 . Xem toàn bộ hóa đơn của cửa hàng _____	44
Hình 35 . Xem chi tiết hóa đơn bán hàng _____	45
Hình 36 . Xem và chỉnh sửa thông tin tài khoản _____	45

1. TỔNG QUAN ĐỀ TÀI

Đề tài “Hệ thống mua bán điện thoại” 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 điện thoại, 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 sản phẩm, khách hàng, đơn hàng và các hoạt động giao dịch có thể được tìm kiếm, quản lý và xử lý nhanh chóng bởi cả người dùng 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, mật khẩu, địa chỉ), Sản phẩm (Mã sản phẩm, tên sản phẩm, thể loại, giá tiền, số lượng sản phẩm, hệ điều hành, thông số kỹ thuật, thương hiệu), 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). Bên cạnh đó hệ thống còn quản lý các thông tin liên quan đến Manager (Người quản lý) với các chức năng (Thêm và xóa sản phẩm, quản lý thông tin khách hàng, xem doanh thu tổng hợp, 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 điện thoại, 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. CƠ SỞ LÝ THUYẾT

Cơ sở lý thuyết cho đề tài **Hệ thống quản lý cửa hàng điện thoại** là :

1. Lưu trữ dữ liệu :

Dữ liệu hệ thống được tổ chức và lưu trữ trong các tệp CSV với phân cấp theo từng mục riêng biệt, chẳng hạn như thông tin sản phẩm, khách hàng, hoặc hóa đơn. Việc quản lý dữ liệu được thực hiện thông qua các phương thức đọc/ghi file, đảm bảo tính chính xác và dễ dàng truy xuất.

2. Cấu trúc dữ liệu:

Hệ thống sử dụng cấu trúc dữ liệu template <Vector> sử dụng danh sách liên kết đôi để quản lý dữ liệu. Cách tiếp cận này đảm bảo tính linh hoạt trong lưu trữ và tối ưu hóa hiệu suất khi thao tác với các tập dữ liệu lớn. Ngoài ra sử dụng 1 template <Pair> để sử dụng hiệu quả với các cặp dữ liệu.

3. Thuật toán sắp xếp:

Trong các chức năng sắp xếp dữ liệu, hệ thống sử dụng thuật toán Insertion Sort.

Thuật toán này được tích hợp để đảm bảo việc sắp xếp các phần tử theo thứ tự mong muốn, đồng thời cân bằng giữa độ phức tạp và hiệu quả xử lý.

4. Tìm kiếm nhị phân:

Với các dữ liệu đã được sắp xếp, hệ thống áp dụng thuật toán tìm kiếm nhị phân (Binary Search) nhằm tối ưu hóa tốc độ tìm kiếm. Điều này giúp giảm thiểu thời gian xử lý, đặc biệt khi làm việc với danh sách lớn.

5. Thiết kế giao diện:

Giao diện người dùng được xây dựng bằng thư viện Qt, mang lại trải nghiệm trực quan, thân thiện và chuyên nghiệp. Các thành phần giao diện được triển khai thông qua các class Interface, giúp người dùng dễ dàng tương tác với hệ thống.

3. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG

3.1 Phân tích chức năng xác thực và phân quyền người dùng:

❖ **Mục tiêu:** Kiểm tra tính hợp lệ của thông tin đăng nhập (email và mật khẩu) và phân quyền người dùng dựa trên vai trò.

❖ **Cách triển khai trong mã:**

Người dùng nhập email và mật khẩu qua giao diện. Phương thức login() thực hiện kiểm tra tính hợp lệ của thông tin: kiểm tra trống (nếu email hoặc mật khẩu để trống, hiển thị thông báo lỗi) và kiểm tra định dạng email (sử dụng điều kiện kiểm tra chuỗi

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH

có chứa "@" và ".") . Sau khi kiểm tra sơ bộ, phương thức gửi thông tin đến appController->allLogin().

Vai trò của AppController: Gửi thông tin email và mật khẩu đến lớp quản lý dữ liệu (DataController) để kiểm tra tính hợp lệ. Phương thức allLogin() trả về **Kết quả** dưới dạng một đối tượng Pair<string, string>: role (vai trò người dùng) xác định loại người dùng (Customer, Manager hoặc không hợp lệ) ,userId: (Mã định danh người dùng).

Phân quyền (Authorization): Nếu người dùng là Customer: chuyển đến giao diện CustomerInterface. Nếu người dùng là Manager: chuyển đến giao diện ManagerInterface. Nếu người dùng không hợp lệ: hiển thị thông báo lỗi "Invalid email or password" và không cho phép truy cập vào hệ thống.

- ❖ **Kết quả:** Nếu thông tin hợp lệ: hiển thị thông báo "Login successful". Tùy thuộc vào vai trò người dùng, mở giao diện tương ứng (CustomerInterface hoặc ManagerInterface). Nếu thông tin không hợp lệ: hiển thị thông báo lỗi "Invalid email or password".

3.2 Phân tích chức năng phía khách hàng:

3.2.1 Tạo tài khoản mới (nếu chưa có tài khoản):

- ❖ **Mục tiêu:** Kiểm tra tính hợp lệ của thông tin đăng ký và lưu thông tin người dùng vào hệ thống.
- ❖ **Cách triển khai trong mã:**

Lấy thông tin: Từ các trường nhập liệu (tên, email, số điện thoại, mật khẩu, xác nhận mật khẩu, địa chỉ).

Kiểm tra hợp lệ:

Trống: Kiểm tra xem có trường nào trống không. Nếu có, hiển thị thông báo lỗi.

Email: Kiểm tra định dạng email (chứa "@" và "."). Nếu không hợp lệ, hiển thị thông báo lỗi.

Số điện thoại: Kiểm tra xem có phải chuỗi số và đúng độ dài 10 ký tự. Nếu sai, hiển thị thông báo lỗi.

Mật khẩu: Kiểm tra mật khẩu với xác nhận mật khẩu. Nếu không khớp, hiển thị thông báo lỗi.

Lưu thông tin: Nếu thông tin hợp lệ, vai trò người dùng là "Customer" và lưu vào hệ thống thông qua signin().

- ❖ **Kết quả:** Sau khi đăng ký thành công, tạo đối tượng CustomerInterface gắn với userID, mở giao diện CustomerInterface, và đóng giao diện đăng ký.

3.2.2 Đăng nhập vào tài khoản (đã có tài khoản):

- ❖ **Mục tiêu:** Kiểm tra tính hợp lệ của thông tin đăng nhập (email và mật khẩu)
- ❖ **Cách triển khai trong mã:** Sau khi đã xác thực , nếu đúng thì tạo đối tượng CustomerInterface gắn với userID, mở giao diện CustomerInterface, và đóng giao diện đăng nhập, nếu sai mật khẩu hoặc email sẽ thông báo ra màn hình .

3.2.3 Tìm kiếm sản phẩm trong cửa hàng:

- ❖ **Mục tiêu:** Cung cấp chức năng tìm kiếm sản phẩm trong cửa hàng dựa trên từ khóa người dùng nhập vào và hiển thị các sản phẩm phù hợp.
- ❖ **Cách triển khai trong mã:**

Lấy từ khóa tìm kiếm: Người dùng nhập từ khóa vào trường searchLineEdit, loại bỏ khoảng trắng thừa bằng trimmed().

Kiểm tra từ khóa: Nếu trường tìm kiếm trống (searchTerm.isEmpty()), tất cả sản phẩm được hiển thị. Nếu không trống, kiểm tra từng sản phẩm trong bảng.

Lọc sản phẩm: Duyệt qua các hàng trong bảng (productTable->rowCount()), lấy giá trị cột thứ 3 và kiểm tra xem nó có chứa từ khóa tìm kiếm hay không (contains()). Nếu khớp, hiển thị hàng; nếu không, ẩn hàng.

- ❖ **Kết quả:** Khi từ khóa tìm kiếm được nhập, chỉ các sản phẩm phù hợp sẽ hiển thị. Nếu trường tìm kiếm trống, tất cả sản phẩm sẽ được hiển thị.

3.2.4 Thêm, xóa sản phẩm trong giỏ hàng:

- ❖ **Mục tiêu:** Quản lý việc thêm và xóa sản phẩm trong giỏ hàng của khách hàng trong giao diện.

a) Thêm sản phẩm trong cửa sổ tất cả sản phẩm của cửa hàng:

Lấy thông tin sản phẩm: Tùy vào việc sản phẩm được chọn từ giỏ hàng (fromCart = true) hay danh sách sản phẩm (fromCart = false), thông tin sản phẩm như tên, giá và ID sản phẩm sẽ được lấy từ bảng tương ứng.

Kiểm tra tồn kho: Trước khi thêm sản phẩm vào giỏ, kiểm tra xem số lượng còn trong kho có đủ không (newProduct.getStock()). Nếu sản phẩm hết hàng, hiển thị thông báo cảnh báo.

Thêm vào giỏ: Tạo bản sao của sản phẩm và thêm vào giỏ hàng của khách hàng. Nếu thao tác từ giỏ hàng, cập nhật lại giao diện giỏ hàng (showCart()).

b) Thêm số lượng, giảm, xóa sản phẩm trong cửa sổ giỏ hàng:

Giảm số lượng/xóa sản phẩm: Nếu sản phẩm đã có trong giỏ, giảm số lượng hoặc xóa sản phẩm khi số lượng đạt 0. Khi xóa hoặc giảm số lượng, giỏ hàng sẽ được cập nhật. Cập nhật giỏ: Sau mỗi thao tác thêm, xóa hoặc giảm số lượng, nếu thao tác từ giỏ hàng, giao diện giỏ hàng sẽ được cập nhật lại.

❖ **Kết quả:**

Thêm sản phẩm: Nếu đủ hàng, sản phẩm được thêm vào giỏ.

Giảm/xóa sản phẩm: Số lượng sản phẩm giảm hoặc sản phẩm bị xóa khỏi giỏ khi số lượng bằng 0. Giao diện giỏ hàng sẽ luôn được cập nhật sau mỗi thay đổi.

3.2.5 Xem giỏ hàng :

❖ **Mục tiêu:** Cung cấp chức năng xem giỏ hàng của khách hàng và tải lại dữ liệu giỏ hàng từ cơ sở dữ liệu.

❖ **Cách triển khai trong mã:**

Lấy dữ liệu giỏ hàng:

Tạo một đối tượng cartOrigin kiểu unique_ptr<Cart>, trong đó dữ liệu giỏ hàng được tải từ cơ sở dữ liệu thông qua phương thức dataController->loadCartData(), sử dụng customer->getUserId() làm tham số để tải giỏ hàng của khách hàng hiện tại.

Lặp qua các sản phẩm trong giỏ hàng:

Sử dụng vòng lặp để duyệt qua các sản phẩm trong giỏ hàng (cartOrigin->getItems()), lấy thông tin sản phẩm và số lượng tương ứng.

Thêm sản phẩm vào giỏ:

Với mỗi sản phẩm trong giỏ hàng, phương thức customer->addToCart() được gọi để thêm sản phẩm và số lượng vào giỏ hàng của khách hàng.

❖ **Kết quả:**

Giỏ hàng của khách hàng sẽ được tái tạo từ dữ liệu đã lưu trữ trong cơ sở dữ liệu. Sau khi phương thức này được gọi, giỏ hàng sẽ đầy đủ các sản phẩm đã được lưu trữ từ trước, sẵn sàng để khách hàng tiếp tục thao tác.

3.2.6 Đặt đơn hàng, chọn ngày giao, phương thức thanh toán:

❖ **Mục tiêu:**

Hoàn tất quá trình đặt đơn hàng, từ việc kiểm tra giỏ hàng, hiển thị hóa đơn, cập nhật thông tin hóa đơn, thanh toán, cho đến việc làm trống giỏ hàng và cập nhật giao diện sau khi hoàn tất đơn hàng.

❖ **Cách triển khai trong mã:**

1. *Kiểm tra giỏ hàng:*

Khi người dùng bấm nút "Đặt hàng", hệ thống kiểm tra giỏ hàng qua hàm checkCart().

Nếu giỏ hàng trống: Hiển thị thông báo "Giỏ hàng đã trống"

Quá trình đặt hàng sẽ dừng lại, không chuyển sang giao diện hóa đơn.

Nếu giỏ hàng có sản phẩm:

Hệ thống chuyển sang giao diện hóa đơn bằng hàm showInvoiceScreen().

2. *Hiển thị hóa đơn và các trường thông tin:*

Sau khi giỏ hàng có sản phẩm: Giao diện hóa đơn sẽ được hiển thị.

Người dùng có thể chọn ngày giao hàng qua bộ chọn ngày (deliveryDateEdit) và phương thức thanh toán qua combo box (paymentMethodComboBox).

Cập nhật giao diện: Khi nhấn "Đặt hàng", nút "Thanh toán" (paymentButton) sẽ được hiển thị, trong khi nút "Đặt hàng" (orderButton) sẽ bị ẩn.

Điều này ngăn không cho người dùng thực hiện thao tác khác trong khi đang thực hiện thanh toán.

3. *Cập nhật và hiển thị thông tin hóa đơn:*

Cập nhật khi thay đổi thông tin:

Khi người dùng thay đổi thông tin ngày giao hàng (deliveryDateEdit.getDate()) hoặc phương thức thanh toán (paymentMethodComboBox.getSelectedItem()), hệ thống sẽ cập nhật và hiển thị lại thông tin này trên giao diện hóa đơn.

Thông tin giỏ hàng (sản phẩm, số lượng, giá trị, tổng tiền) sẽ được tính toán và hiển thị trong khu vực dành riêng cho hóa đơn (invoiceDisplay).

Cập nhật thông tin bằng các hàm như updateInvoiceDate() và updatePaymentMethod().

4. *Thanh toán và hoàn tất đơn hàng:*

Quy trình thanh toán:

Sau khi người dùng chọn phương thức thanh toán và ngày giao hàng, họ nhấn nút "Thanh toán" (paymentButton.addActionListener()).

Hệ thống thực hiện thanh toán, ví dụ: kiểm tra tính hợp lệ của thẻ tín dụng hoặc xác nhận thông tin chuyển khoản ngân hàng.

Kết quả thanh toán: Sau khi thanh toán thành công, hệ thống hiển thị thông báo "Mua hàng thành công" (showMessage("Mua hàng thành công")).

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH

Dữ liệu đơn hàng, bao gồm thông tin thanh toán và ngày giao hàng, sẽ được lưu trữ trong hệ thống (saveOrderData(order)).

Giỏ hàng sẽ được làm trống sau khi thanh toán thành công (clearCart()).

5. Làm trống giỏ hàng và cập nhật giao diện:

Sau khi thanh toán thành công:

Giỏ hàng sẽ được làm trống để chuẩn bị cho các lần mua sắm tiếp theo (clearCart()).

Các trường thông tin như ngày giao hàng và phương thức thanh toán sẽ bị ẩn đi (deliveryDateEdit.setVisible(false), paymentMethodComboBox.setVisible(false)).

Giao diện sẽ quay lại trạng thái ban đầu, hiển thị lại các sản phẩm trong cửa hàng và các chức năng mua sắm mới sẽ được hiển thị (showShoppingCart()).

❖ Kết quả:

Người dùng sẽ có thể thực hiện việc đặt hàng thành công nếu giỏ hàng có sản phẩm, sau đó xác nhận thông tin hóa đơn và thanh toán. Sau khi thanh toán thành công, giỏ hàng sẽ được làm trống và giao diện sẽ được cập nhật, sẵn sàng cho các lần mua sắm tiếp theo.

3.2.7 Xem danh sách đơn hàng đã mua:

❖ Mục tiêu:

Hiển thị danh sách đơn hàng đã mua, bao gồm chi tiết thông tin các hóa đơn như ngày giao hàng, phương thức thanh toán, và danh sách sản phẩm trong từng hóa đơn.

❖ Cách triển khai trong mã:

1. Truy xuất danh sách hóa đơn:

Khi người dùng chọn xem danh sách đơn hàng, hệ thống gọi phương thức getOrderHistory() từ lớp Customer. Sau đó lấy đối tượng Orders, sau đó gọi hàm getInvoice() để truy xuất danh sách hóa đơn (Vector<Invoice*>).

Triển khai:

Nếu danh sách hóa đơn rỗng thì hệ thống sẽ hiển thị thông báo "Bạn chưa có đơn hàng nào". Đồng thời giao diện danh sách đơn hàng sẽ không hiển thị.

Nếu danh sách hóa đơn có dữ liệu thì hệ thống hiển thị bảng danh sách hóa đơn trên giao diện bằng hàm showOrderList().

2. Hiển thị danh sách hóa đơn và các trường thông tin:

Sau khi truy xuất được danh sách hóa đơn, giao diện sẽ hiển thị:

Invoice ID: Mã hóa đơn.

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH

Delivery Date: Ngày giao hàng.

Payment Method: Phương thức thanh toán.

Total Amount: Tổng tiền của hóa đơn.

Tiếp đó hệ thống sẽ cập nhật giao diện và khi danh sách hóa đơn được hiển thị thì hệ thống sẽ cho phép người dùng bấm vào từng hóa đơn để xem chi tiết bằng hàm showInvoiceDetails().

3. Hiển thị chi tiết hóa đơn:

Khi người dùng chọn một hóa đơn cụ thể từ bảng, hệ thống sẽ lấy dữ liệu từ đối tượng Invoice tương ứng. Đồng thời hiển thị thông tin chi tiết như: Danh sách sản phẩm (tên sản phẩm, số lượng, giá từng sản phẩm), Tổng giá trị hóa đơn.

Triển khai:

Giao diện chi tiết hóa đơn sẽ bao gồm:

Bảng sản phẩm (Product Name, Quantity, Price).

Thông tin chung (Delivery Date, Payment Method, Total Amount).

3.2.8 Xem thông tin, thay đổi thông tin tài khoản cá nhân

❖ Mục tiêu:

Cho phép khách hàng xem và thay đổi thông tin cá nhân, đảm bảo khách hàng có thể cập nhật thông tin cá nhân như tên, số điện thoại, địa chỉ khi cần thiết.

❖ Cách triển khai trong mã:

1. Xem thông tin cá nhân:

Khi khách hàng nhấp vào nút "Xem thông tin cá nhân", hệ thống sẽ gọi hàm loadCustomerInfor(). Hàm này truy xuất dữ liệu từ cơ sở dữ liệu hoặc danh sách đối tượng khách hàng (Customer), sau đó hiển thị các thông tin cá nhân như: Họ tên, Email Số điện thoại, Địa chỉ.

Thông tin này được hiển thị trên giao diện người dùng thông qua các nhãn (Label).

2. Thay Đổi Thông Tin Cá Nhân:

Khi khách hàng nhấp vào nút "Chỉnh sửa thông tin cá nhân," các trường thông tin (tên, số điện thoại, địa chỉ) chuyển sang trạng thái có thể chỉnh sửa.

Khách hàng nhập thông tin mới, sau đó nhấp nút "Lưu thay đổi."

Hệ thống gọi hàm updateCustomerInfo() để:

Kiểm tra tính hợp lệ của thông tin (ví dụ: số điện thoại phải là số, email đúng định dạng).

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH

Cập nhật dữ liệu trong đối tượng Customer.

Lưu lại thay đổi vào cơ sở dữ liệu hoặc danh sách khách hàng qua hàm DataController:: updateCustomerData().

❖ **Kết quả:**

Sau khi lưu thành công, hiển thị thông báo "Thông tin đã được cập nhật thành công" và quay lại chế độ chỉ đọc.

3.2.9. Xem các thông số thống kê

3.2.9.a) Thông kê tổng số hóa đơn, số tiền

❖ **Mục Tiêu:**

Cung cấp thông tin về tổng số hóa đơn mà khách hàng đã thực hiện và tổng số tiền mà họ đã chi trả. Điều này giúp khách hàng theo dõi lịch sử mua sắm của mình một cách dễ dàng.

❖ **Cách Triển Khai Trong Mã:**

1. Kiểm tra lịch sử đơn hàng:

Khi người dùng yêu cầu xem thông kê hóa đơn, hệ thống sẽ kiểm tra lịch sử đơn hàng của khách hàng thông qua phương thức getOrderHistory().

Nếu lịch sử đơn hàng trống: Hiển thị thông báo "Chưa có đơn hàng nào".

Nếu có đơn hàng: Hệ thống tiếp tục lấy thông tin về các hóa đơn và tính toán thông kê tổng số hóa đơn và tổng số tiền đã chi.

2. Tính Toán Tổng Số Hóa Đơn và Tổng Số Tiền:

Sau khi kiểm tra lịch sử đơn hàng, hệ thống sẽ:

Tính tổng số hóa đơn: Duyệt qua tất cả các hóa đơn trong orderHistory và đếm số lượng hóa đơn.

Tính tổng số tiền: Duyệt qua từng hóa đơn, lấy giá trị totalAmount từ mỗi hóa đơn và cộng dồn vào tổng tiền.

3. Hiển Thị Thông Kê:

Khi hệ thống tính toán xong, sẽ hiển thị thông tin thống kê cho khách hàng.

Số hóa đơn: Hiển thị số lượng hóa đơn mà khách hàng đã thực hiện.

Tổng số tiền: Hiển thị tổng số tiền mà khách hàng đã chi cho các hóa đơn.

4. Cập Nhật Giao Diện:

Sau khi tính toán và hiển thị thông tin, hệ thống sẽ cập nhật giao diện để khách hàng có thể nhìn thấy các thông kê về tổng số hóa đơn và số tiền đã chi tiêu.

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH

Giao diện sẽ hiển thị tổng số hóa đơn và tổng số tiền.

Có thể hiển thị thêm các thông tin chi tiết về các hóa đơn nếu cần.

❖ Kết Quả:

Khách hàng sẽ có thể thấy tổng số hóa đơn mà họ đã thực hiện và tổng số tiền mà họ đã chi tiêu. Thông tin này giúp khách hàng theo dõi lịch sử mua sắm của mình một cách minh bạch và dễ dàng. Giao diện sẽ được cập nhật với các thông tin này, giúp khách hàng có cái nhìn rõ ràng hơn về các giao dịch của mình.

3.2.9.b) Biểu đồ số tiền trên mỗi hóa đơn:

❖ Mục Tiêu:

Hiển thị trực quan số tiền chi tiêu tương ứng với từng hóa đơn qua biểu đồ dạng cột. Điều này giúp khách hàng có cái nhìn tổng quan hơn về các khoản chi tiêu theo từng hóa đơn.

❖ Cách Triển Khai Trong Mã:

1. Kiểm tra lịch sử đơn hàng:

Khi khách hàng yêu cầu hiển thị biểu đồ, hệ thống thực hiện:

Kiểm tra lịch sử đơn hàng của khách hàng thông qua phương thức `getOrderHistory()`.

Nếu không có hóa đơn: Hiển thị thông báo "Không có hóa đơn nào để hiển thị biểu đồ."

Nếu có hóa đơn: Chuẩn bị dữ liệu cho biểu đồ bằng cách lấy thông tin ID hóa đơn và số tiền từ từng hóa đơn.

2. Chuẩn bị dữ liệu cho biểu đồ :

Duyệt qua danh sách hóa đơn để tạo một danh sách cặp dữ liệu (Mã hóa đơn, Số tiền).

Giới hạn số lượng hóa đơn hiển thị trên biểu đồ (tối đa 20 hóa đơn gần nhất).

Lưu thông tin này vào một cấu trúc như `Vector<Pair<QString, double>>` data.

3. Vẽ biểu đồ :

Sử dụng hàm `drawChart()` với các tham số:

+ Title: "Invoice ID by Amount"

+ X-axis Label: "Invoice ID"

+ Y-axis Label: "Amount"

+ Container: Vùng giao diện để hiển thị biểu đồ.

Các bước thực hiện trong `drawChart()`:

Xóa layout cũ của container để đảm bảo giao diện không bị chồng chéo.

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH

Sử dụng dữ liệu data để vẽ biểu đồ dạng cột, thêm cột vào trục Y và mã hóa đơn vào trục X.

Căn chỉnh và tối ưu hóa giao diện biểu đồ bằng cách bật tính năng khử răng cưa, hiển thị nhãn số liệu trên các cột, và tạo hiệu ứng mượt khi hiển thị.

4. Cập nhật giao diện:

Sau khi biểu đồ được vẽ, hệ thống cập nhật vào diagramBox (một QGroupBox) trong giao diện khách hàng.

Đảm bảo biểu đồ được hiển thị đúng vị trí và kích thước mong muốn trong giao diện chính.

❖ Kết Quả:

Khách hàng sẽ thấy một biểu đồ dạng cột hiển thị số tiền trên mỗi hóa đơn.

Biểu đồ chỉ hiển thị tối đa 20 hóa đơn gần nhất để duy trì giao diện rõ ràng.

Thông tin trực quan này giúp khách hàng dễ dàng nhận biết các giao dịch lớn, nhỏ và xu hướng chi tiêu của mình.

3.2.10 Đăng xuất

Người dùng thoát ứng dụng qua nút Logout , hệ thống tiến hành đóng cửa sổ CustomerInterface kết thúc.

3.3 Phân tích chức năng phía quản lý:

3.3.1 Đăng nhập vào tài khoản quản lý

- ❖ **Mục tiêu:** Kiểm tra tính hợp lệ của thông tin đăng nhập (email và mật khẩu)
- ❖ **Cách triển khai trong mã:** Sau khi đã xác thực , nếu đúng thì tạo đối tượng CustomerInterface gắn với userID, mở giao diện CustomerInterface, và đóng giao diện đăng nhập, nếu sai mật khẩu hoặc email sẽ thông báo ra màn hình .

3.3.2 Xem danh sách sản phẩm trong cửa hàng

- ❖ **Mục tiêu:** Mục tiêu của chức năng này là cho phép quản lý có thể xem tất cả sản phẩm hiện có trong cửa hàng, bao gồm các thông tin như mã sản phẩm, tên sản phẩm, loại sản phẩm, giá, số lượng trong kho và mô tả sản phẩm.

❖ **Cách triển khai trong mã:**

1. Tải Dữ Liệu Sản Phẩm:

Quản lý có thể xem danh sách sản phẩm thông qua chức năng loadProductData() trong lớp DataController. Phương thức này sẽ mở file chứa thông tin sản phẩm (tệp Data/ProductInformation.csv) và đọc từng dòng dữ liệu.

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH

Dữ liệu của mỗi sản phẩm sẽ được phân tích từ file CSV, và mỗi sản phẩm sẽ được tạo thành một đối tượng Product bằng phương thức parseProduct(). Các đối tượng Product hợp lệ (dữ liệu không bị thiếu hoặc sai) sẽ được lưu trữ vào trong một đối tượng Vector<Product>, đại diện cho danh sách sản phẩm.

2. Phương Thức loadProductData():

Phương thức này thực hiện các bước sau:

Mở tệp CSV chứa thông tin sản phẩm.

Đọc từng dòng trong tệp và phân tích thông tin thành các trường dữ liệu (mã sản phẩm, tên, loại, giá, số lượng, mô tả, chi tiết, và thương hiệu).

Sử dụng phương thức parseProduct() để tạo ra đối tượng Product từ mỗi dòng dữ liệu.

Thêm các đối tượng Product hợp lệ vào danh sách Vector<Product>.

3. Hiển Thị Danh Sách Sản Phẩm:

Sau khi danh sách sản phẩm đã được tải về, quản lý có thể dễ dàng duyệt qua các phần tử trong Vector<Product> để hiển thị thông tin của từng sản phẩm.

Thông tin của mỗi sản phẩm sẽ bao gồm mã sản phẩm, tên, giá cả, số lượng tồn kho, mô tả, chi tiết sản phẩm, và thương hiệu.

❖ Kết quả:

Quản lý có thể xem danh sách đầy đủ các sản phẩm trong cửa hàng, với thông tin chi tiết về từng sản phẩm như mã, tên, loại, giá, số lượng trong kho, mô tả, chi tiết và thương hiệu.

3.3.3 Xoá sản phẩm khỏi cửa hàng:

❖ Mục tiêu:

Mục tiêu của chức năng này là cho phép quản lý xoá sản phẩm khỏi cửa hàng dựa trên mã sản phẩm (productId). Khi sản phẩm bị xoá, nó sẽ bị loại bỏ khỏi danh sách sản phẩm hiện có trong hệ thống và thay đổi sẽ được lưu vào tệp dữ liệu.

❖ Cách triển khai trong mã:

1. Tìm và xoá sản phẩm:

Chức năng này được triển khai trong phương thức removeProduct() của lớp Manager. Phương thức này sẽ tìm kiếm sản phẩm trong danh sách các sản phẩm hiện có (dữ liệu từ tệp ProductInformation.csv).

Nếu sản phẩm có productId khớp với mã sản phẩm cần xoá, nó sẽ được loại bỏ khỏi danh sách.

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH

Sau khi xoá sản phẩm khỏi danh sách, phương thức sẽ gọi hàm saveProductsData() trong lớp DataController để lưu lại dữ liệu mới vào tệp.

2. Qui trình xóa sản phẩm :

Bước 1: Tải danh sách sản phẩm hiện có từ tệp ProductInformation.csv thông qua phương thức loadProductData(). Dữ liệu được lưu trữ trong Vector<Product>.

Bước 2: Duyệt qua danh sách sản phẩm (Vector<Product>) và tìm sản phẩm có productId khớp với mã sản phẩm cần xoá.

Bước 3: Nếu tìm thấy sản phẩm với mã productId khớp, sản phẩm đó sẽ được xoá khỏi danh sách bằng phương thức remove(i) của lớp Vector.

Bước 4: Sau khi xoá sản phẩm, lưu lại danh sách mới vào tệp CSV thông qua phương thức saveProductsData().

Bước 5: Nếu không tìm thấy sản phẩm với mã productId trong danh sách, một thông báo lỗi sẽ được hiển thị.

3. Phương thức removeProduct() trong lớp Manager:

Phương thức này tìm sản phẩm theo productId và loại bỏ nó khỏi danh sách sản phẩm.

Lý do xoá sản phẩm có thể là: khi quản lý muốn loại bỏ sản phẩm khỏi cửa hàng (chẳng hạn khi sản phẩm hết hàng, lỗi dữ liệu, hoặc cần xóa các sản phẩm không còn bán nữa).

Và sau khi sản phẩm được xoá, dữ liệu của cửa hàng sẽ được cập nhật và lưu lại vào tệp ProductInformation.csv để đảm bảo sự đồng bộ giữa hệ thống và cơ sở dữ liệu lưu trữ.

Nếu không tìm thấy sản phẩm, hệ thống sẽ trả về false và không có thay đổi nào.

❖ Kết Quả:

Khi chức năng xoá sản phẩm thành công, sản phẩm sẽ bị loại bỏ khỏi cửa hàng, và danh sách sản phẩm sẽ được cập nhật và lưu lại trong tệp dữ liệu. Quản lý có thể dễ dàng quản lý các sản phẩm trong cửa hàng, bao gồm việc xoá các sản phẩm không còn kinh doanh nữa.

3.3.4 Thay đổi thông tin, số lượng sản phẩm

❖ Mục tiêu:

Mục tiêu của chức năng này là cho phép quản lý thay đổi thông tin sản phẩm, bao gồm các thuộc tính như tên, giá, mô tả, số lượng tồn kho, v.v. Cụ thể, quản lý có thể cập

nhập thông tin sản phẩm (ví dụ: tên, giá, mô tả, v.v.) hoặc thay đổi số lượng tồn kho của sản phẩm.

❖ **Cách triển khai trong mã:**

Trong mã nguồn, việc thay đổi thông tin hoặc số lượng sản phẩm được thực hiện thông qua phương thức updateProduct() trong lớp Manager. Quy trình này giúp quản lý thay đổi thông tin của một sản phẩm đã tồn tại trong danh sách sản phẩm.

1. *Tải dữ liệu sản phẩm:*

Phương thức updateProduct(Product& product) trong lớp Manager sẽ gọi phương thức loadProductData() của lớp DataController để tải danh sách tất cả các sản phẩm từ tệp ProductInformation.csv.

2. *Duyệt qua danh sách sản phẩm:*

Sau khi tải được danh sách sản phẩm (Vector<Product>), hệ thống sẽ duyệt qua từng sản phẩm trong danh sách này.

Phương thức tìm kiếm sản phẩm thông qua productId (mã sản phẩm). Nếu tìm thấy sản phẩm có productId khớp, hệ thống sẽ thay thế sản phẩm cũ bằng sản phẩm mới có thông tin đã được cập nhật.

3. *Cập nhật thông tin sản phẩm:*

Khi tìm thấy sản phẩm cần cập nhật, hệ thống sẽ thay thế sản phẩm cũ trong danh sách bằng sản phẩm mới được truyền vào phương thức updateProduct().

Sau khi cập nhật thông tin, phương thức saveProductsData() sẽ được gọi để lưu lại danh sách sản phẩm đã thay đổi vào tệp ProductInformation.csv.

4. *Cập nhật số lượng tồn kho :*

Nếu muốn thay đổi số lượng tồn kho của sản phẩm, phương thức updateProduct() có thể được sử dụng để chỉnh sửa số lượng tồn kho trong đối tượng Product tương ứng.

Khi đó, khi lưu lại danh sách, số lượng tồn kho mới sẽ được cập nhật.

Hệ thống tìm kiếm sản phẩm trong danh sách sản phẩm, nếu tìm thấy sản phẩm với productId khớp, thông tin của sản phẩm sẽ được thay thế bằng thông tin mới.

Danh sách sản phẩm sau khi cập nhật sẽ được lưu lại vào tệp ProductInformation.csv.

❖ **Kết Quả:**

Khi chức năng cập nhật sản phẩm thành công, thông tin của sản phẩm sẽ được thay đổi và số lượng tồn kho sẽ được điều chỉnh. Sau khi thông tin được cập nhật, danh sách sản phẩm sẽ được lưu lại trong tệp ProductInformation.csv, đảm bảo dữ liệu luôn đồng

bộ với thông tin thực tế của cửa hàng. Quản lý có thể duy trì và điều chỉnh thông tin sản phẩm một cách linh hoạt trong suốt quá trình kinh doanh.

3.3.5 Xem danh sách khách hàng

❖ Mục tiêu:

Mục tiêu là cung cấp cho quản lý một chức năng để xem danh sách khách hàng, bao gồm việc truy xuất và sắp xếp thông tin khách hàng theo số tiền đã chi, giúp quản lý dễ dàng theo dõi và phân tích dữ liệu khách hàng.

❖ Cách triển khai trong mã:

1. Lấy dữ liệu khách hàng:

Khi quản lý muốn xem danh sách khách hàng, hệ thống sẽ sử dụng phương thức loadAllCustomersData() trong lớp DataController để lấy tất cả thông tin khách hàng từ cơ sở dữ liệu.

Dữ liệu khách hàng: Phương thức này trả về một danh sách các đối tượng Customer, mỗi đối tượng chứa thông tin về khách hàng như userId, name, email, phone, address, v.v.

Kiểm tra dữ liệu: Nếu danh sách khách hàng trống, hệ thống có thể hiển thị thông báo "Không có khách hàng" hoặc tiếp tục xử lý mà không hiển thị gì.

2. Sắp xếp khách hàng theo số tiền đã mua :

Quản lý có thể muốn xem danh sách khách hàng theo tổng số tiền họ đã chi. Để thực hiện điều này, phương thức sortCustomerByAmount() trong AppController được sử dụng.

Lấy danh sách khách hàng: Phương thức sortCustomerByAmount() gọi loadAllCustomersData() để lấy tất cả khách hàng.

Tính toán tổng số tiền: Sau đó, hệ thống sẽ tính tổng số tiền mà mỗi khách hàng đã chi tiêu thông qua các hóa đơn mà khách hàng đã thực hiện. Tổng tiền này được tính bằng cách duyệt qua tất cả hóa đơn của khách hàng, sử dụng các phương thức trong lớp Invoice.

Sắp xếp: Sau khi tính toán tổng số tiền, danh sách khách hàng được sắp xếp theo số tiền chi tiêu từ cao đến thấp, giúp quản lý dễ dàng nhận diện các khách hàng chi tiêu nhiều nhất.

3.Hiển thị danh sách khách hàng:

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH

Sau khi dữ liệu được sắp xếp, hệ thống sẽ hiển thị danh sách khách hàng theo thứ tự đã được sắp xếp.

Giao diện người dùng: Giao diện có thể hiển thị thông tin về khách hàng, bao gồm tên, email, số điện thoại, và tổng số tiền đã chi. Quản lý có thể dùng bảng hoặc danh sách để hiển thị dữ liệu này.

Thông báo: Nếu không có khách hàng hoặc có lỗi trong quá trình tải dữ liệu, hệ thống có thể hiển thị thông báo lỗi cho quản lý.

❖ Kết Quả:

Quản lý có thể dễ dàng truy xuất và xem danh sách khách hàng, sắp xếp họ theo mức độ chi tiêu, và thực hiện các thao tác quản lý khách hàng khác. Việc này giúp quản lý nắm bắt thông tin khách hàng nhanh chóng và đưa ra các quyết định kinh doanh hiệu quả hơn.

3.3.6 Xem danh sách hóa đơn của cửa hàng

❖ Mục tiêu:

Quản lý có thể xem danh sách hóa đơn của cửa hàng, bao gồm việc truy xuất tất cả hóa đơn, sắp xếp chúng theo ngày và hiển thị thông tin chi tiết để phục vụ cho việc phân tích doanh thu hoặc xử lý các vấn đề liên quan đến đơn hàng.

❖ Cách triển khai trong mã:

1. Lấy dữ liệu hóa đơn:

Quản lý có thể truy xuất danh sách tất cả hóa đơn thông qua phương thức loadAllInvoice() trong lớp AppController.

Đầu tiên là truy xuất hóa đơn của khách hàng: Phương thức loadAllInvoice() sẽ gọi loadAllCustomersData() từ DataController để lấy danh sách khách hàng, sau đó tiếp tục sử dụng thông tin của từng khách hàng để tải các hóa đơn của họ thông qua phương thức loadOrdersData() từ lớp DataController. Mỗi khách hàng có thể có nhiều hóa đơn.

Tiếp theo là lưu trữ hóa đơn: Hóa đơn của từng khách hàng sẽ được gom lại thành một danh sách lớn của tất cả hóa đơn trong cửa hàng.

2. Sắp xếp hóa đơn theo ngày:

Quản lý có thể muốn sắp xếp hóa đơn theo ngày để dễ dàng theo dõi doanh thu trong các khoảng thời gian cụ thể. Phương thức sortInvoiceByDate() được sử dụng để thực hiện điều này.

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH

Lấy danh sách hóa đơn: Phương thức sortInvoiceByDate() gọi loadAllCustomersData() và sau đó truy xuất hóa đơn từ tất cả khách hàng.

Sắp xếp hóa đơn: Hóa đơn được sắp xếp theo ngày từ mới nhất đến cũ nhất bằng cách so sánh ngày của mỗi hóa đơn. Việc sắp xếp này giúp quản lý dễ dàng theo dõi đơn hàng theo thời gian.

Mô tả: Phương thức này sắp xếp tất cả hóa đơn theo ngày giảm dần (từ ngày mới nhất đến ngày cũ nhất).

3. Hiển thị danh sách hóa đơn:

Sau khi danh sách hóa đơn được lấy và sắp xếp, quản lý có thể hiển thị chúng trên giao diện. Mỗi hóa đơn có thể bao gồm thông tin chi tiết như:

- + Mã hóa đơn
- + Ngày mua
- + Tổng số tiền
- + Thông tin khách hàng
- + Các sản phẩm đã mua

Đồng thời giao diện người dùng sẽ có thể hiển thị danh sách này dưới dạng bảng hoặc danh sách. Quản lý có thể duyệt qua danh sách hóa đơn, tìm kiếm các hóa đơn cụ thể hoặc kiểm tra chi tiết của từng hóa đơn.

4. Quản lý thao tác với hóa đơn :

Sau khi danh sách hóa đơn được hiển thị, quản lý có thể thực hiện các thao tác khác như:

Tìm kiếm hóa đơn: Quản lý có thể tìm kiếm hóa đơn theo mã đơn hàng, tên khách hàng, hoặc ngày tháng.

Xem chi tiết hóa đơn: Quản lý có thể nhấp vào một hóa đơn cụ thể để xem chi tiết các sản phẩm, số lượng, giá trị của từng sản phẩm, và các thông tin liên quan.

❖ Kết Quả:

Quản lý có thể dễ dàng xem và theo dõi tất cả hóa đơn trong cửa hàng, từ đó phân tích doanh thu, đánh giá hành vi mua sắm của khách hàng hoặc kiểm tra tình trạng của các đơn hàng. Việc sắp xếp hóa đơn theo ngày giúp quản lý theo dõi dễ dàng hơn, đặc biệt khi cần báo cáo theo các khoảng thời gian cụ thể.

3.3.7 Xem thay đổi thông tin tài khoản cá nhân

❖ Mục tiêu:

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH

Quản lý có thể xem và thay đổi thông tin cá nhân của mình trong hệ thống, bao gồm các thao tác như xem thông tin hiện tại, chỉnh sửa thông tin (tên, email, số điện thoại, địa chỉ), và lưu các thay đổi vào cơ sở dữ liệu.

❖ Cách triển khai trong mã:

1. Xem thông tin cá nhân:

Quản lý có thể xem thông tin cá nhân thông qua việc truy xuất dữ liệu từ hệ thống.

Triển khai trong mã:

Phương thức liên quan gồm:

DataController::loadAllManagersData(): Tải danh sách tất cả quản lý từ cơ sở dữ liệu.

Manager class: Cung cấp các phương thức như getName(), getEmail(), getPhone(), getAddress() để lấy thông tin chi tiết của quản lý.

Quy trình:

Đầu tiên, tải danh sách tất cả quản lý thông qua loadAllManagersData().

Sau đó, so khớp userId của quản lý hiện tại (được truyền từ phiên đăng nhập) để xác định đối tượng quản lý cụ thể.

Cuối cùng, lấy thông tin như tên, email, số điện thoại, địa chỉ và hiển thị chúng trên giao diện.

2. Thay đổi thông tin cá nhân:

Quản lý có thể cập nhật các trường thông tin cá nhân như tên, email, số điện thoại hoặc địa chỉ.

Triển khai trong mã:

Phương thức liên quan gồm:

DataController::updateManagerData(Manager&): Lưu thông tin đã chỉnh sửa vào cơ sở dữ liệu.

Các phương thức setName(), setEmail(), setPhone(), setAddress() trong lớp Manager để cập nhật thông tin mới cho đối tượng quản lý.

Quy trình:

Xác định quản lý cần cập nhật bằng cách sử dụng userId.

Thu thập thông tin mới từ giao diện người dùng.

Sử dụng các phương thức set... để thay đổi thông tin trong đối tượng Manager.

Gọi phương thức updateManagerData() để lưu các thay đổi vào cơ sở dữ liệu.

3. Hiển thị giao diện thông báo:

Sau khi cập nhật thành công thông tin cá nhân, giao diện sẽ hiển thị thông báo xác nhận.

❖ **Kết Quả:**

Sau khi hoàn tất quá trình xem và chỉnh sửa thông tin, dữ liệu được cập nhật sẽ được lưu trữ vào hệ thống. Thông tin mới của quản lý sẽ hiển thị trên giao diện khi truy cập lại phần "Thông tin cá nhân."

3.3.8 Thông kê tổng số hóa đơn, số tiền, khách hàng

❖ **Mục tiêu:**

Chức năng cho phép quản lý xem các thông kê liên quan đến: tổng số hóa đơn của cửa hàng, tổng số tiền đã giao dịch, tổng số khách hàng của cửa hàng.

Những thông kê này hỗ trợ quản lý trong việc nắm bắt tình hình hoạt động của cửa hàng một cách tổng quan.

❖ **Cách triển khai trong mã:**

1. Tổng Số Hóa Đơn Của Cửa Hàng:

Triển khai trong mã:

AppController::loadAllInvoice(): Truy xuất tất cả hóa đơn từ dữ liệu.

DataController::loadOrdersData() và Customer::getUserId(): Truy xuất dữ liệu hóa đơn từ từng khách hàng.

Gọi phương thức loadAllInvoice() để tải tất cả hóa đơn trong hệ thống.

Sử dụng hàm getSize() của Vector<Invoice*> để xác định tổng số hóa đơn.

2. Tổng Số Tiền Đã Giao Dịch:

Invoice::getTotalAmount(): Lấy tổng tiền của từng hóa đơn.

AppController::loadAllInvoice(): Truy xuất tất cả hóa đơn.

Tải danh sách tất cả hóa đơn thông qua loadAllInvoice().

Sử dụng vòng lặp để cộng dồn giá trị từ các hóa đơn thông qua getTotalAmount().

3. Tổng Số Khách Hàng:

DataController::loadAllCustomersData(): Lấy danh sách tất cả khách hàng từ cơ sở dữ liệu.

Gọi loadAllCustomersData() để tải danh sách khách hàng.

Sử dụng hàm getSize() của Vector<Customer> để xác định tổng số khách hàng.

Và cuối cùng là tích hợp kết quả thống kê lên giao diện:

Khi quản lý yêu cầu thống kê, hệ thống sẽ hiển thị các thông tin sau:

Tổng số hóa đơn: Được tính bằng allInvoices.getSize().

Tổng số tiền giao dịch: Được tính bằng tổng giá trị getTotalAmount() của các hóa đơn.

Tổng số khách hàng: Được tính bằng allCustomers.getSize().

❖ **Kết quả:**

Khi sử dụng chức năng này, quản lý sẽ có thể: xem được tổng số hóa đơn của cửa hàng, xác định tổng doanh thu của cửa hàng, biết được tổng số lượng khách hàng hiện có trong hệ thống. Tất cả dữ liệu này được truy xuất từ cơ sở dữ liệu và đảm bảo tính chính xác, hỗ trợ quản lý trong việc phân tích hiệu suất kinh doanh.

3.3.9 Biểu đồ cột các thông kê:

❖ **Mục Tiêu:**

Chức năng Biểu đồ cột các thông kê cho phép quản lý xem các thông kê dưới dạng biểu đồ cột. Các thông kê này sẽ giúp quản lý cửa hàng dễ dàng nắm bắt thông tin về số lượng sản phẩm bán ra, tổng số tiền giao dịch theo từng khách hàng, và doanh thu theo ngày hoặc sản phẩm.

❖ **Các Chức Năng Cần Triển Khai:**

Sử dụng hàm drawChart() với các tham số:

- + Title: “Invoice ID by Amount”
- + X-axis Label: “Invoice ID”
- + Y-axis Label: “Amount”
- + Container: Vùng giao diện để hiển thị biểu đồ.

Các bước thực hiện trong drawChart():

Xóa layout cũ của container để đảm bảo giao diện không bị chồng chéo.

Sử dụng dữ liệu data để vẽ biểu đồ dạng cột, thêm cột vào trực Y và mã hóa đơn vào trực X.

Căn chỉnh và tối ưu hóa giao diện biểu đồ bằng cách bật tính năng khử răng cưa, hiển thị nhãn số liệu trên các cột, và tạo hiệu ứng mượt khi hiển thị.

a) Biểu Đồ Số Lượng Sản Phẩm Đã Bán:

Mục tiêu là hiển thị tổng số lượng của từng sản phẩm đã bán.

Cách triển khai trong mã:

Phương thức liên quan: dataController->loadSoldProductData().

Gọi phương thức loadSoldProductData() để lấy dữ liệu về sản phẩm đã bán.

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH

Tạo một mảng dữ liệu với định dạng `Pair<QString, double>` gồm tên sản phẩm và số lượng bán được.

Sử dụng hàm `drawChart()` để hiển thị biểu đồ cột.

b) *Biểu Đồ Tổng Số Tiền Theo Ngày (Doanh Thu Theo Ngày):*

Mục tiêu là vẽ biểu đồ thể hiện tổng số tiền giao dịch trong mỗi ngày trong khoảng thời gian cụ thể (ví dụ 14 ngày).

Cách triển khai trong mã:

Phương thức liên quan: `appController->loadAllInvoice()`, `Invoice::getInvoiceDate()`, `Invoice::getTotalAmount()`

Quy trình:

Lấy dữ liệu tất cả các hóa đơn trong khoảng thời gian từ `loadAllInvoice()`.

Xác định tổng số tiền cho mỗi ngày bằng cách cộng dồn giá trị `getTotalAmount()` của các hóa đơn.

Tạo danh sách ngày và số tiền ứng với mỗi ngày.

Gọi hàm `drawChart()` để hiển thị biểu đồ cột với dữ liệu ngày và doanh thu.

c) *Biểu Đồ Tổng Số Tiền Theo Khách Hàng:*

Mục tiêu là vẽ biểu đồ cột để hiển thị tổng số tiền đã chi tiêu của từng khách hàng.

Cách triển khai trong mã:

Phương thức liên quan: `dataController->loadAllCustomersData()`, `dataController->loadOrdersData()`, `Customer::getUserID()`

Quy trình:

Lấy danh sách tất cả khách hàng từ `loadAllCustomersData()`.

Với mỗi khách hàng, lấy danh sách hóa đơn của họ bằng cách sử dụng `loadOrdersData()`.

Cộng dồn giá trị `getTotalAmount()` của tất cả hóa đơn của khách hàng.

Tạo danh sách khách hàng và tổng số tiền của họ.

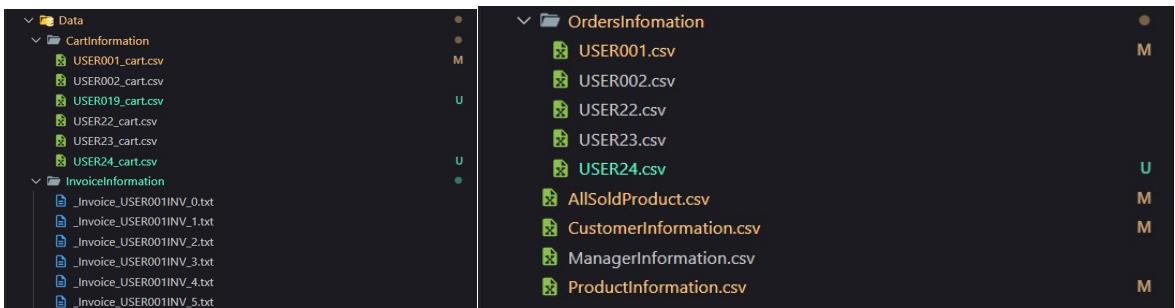
Gọi hàm `drawChart()` để hiển thị biểu đồ cột với tên khách hàng và số tiền đã chi.

4. THIẾT KẾ CẤU TRÚC DỮ LIỆU

4.1 Phát biểu bài toán:

- ❖ 4.1.1 Mô tả đầu vào : Các file CSV được phân cấp theo thư mục :

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH



❖ 4.1.2 Mô tả quá trình lưu trữ, trích xuất dữ liệu :

Lớp DataController đảm nhiệm vai trò lưu trữ dữ liệu, trích xuất dữ liệu : Product, Cart, Invoice, Order, Customer, Manager :

```
You, 2 weeks ago | 1 author (You)
class DataController {
public:
    Product findProductById(const string& productId);
    void saveProductsData(const Vector<Product>& product);
    Vector<Product> loadProductData();
    void removeProduct(const Invoice &invoice);
    void deleteProduct(const string& productId);
    void addNewProduct(const Product& product);

    void saveCartData(const Cart& cart);
    Cart loadCartData(const string& customerID);
    void saveInvoiceData(const Invoice& invoice);

    Vector<Pair<string, int>> loadSoldProductData();
    void saveOrdersData(const Orders &orders);
    Orders loadOrdersData(const string& customerID);

    void addToSoldProductData(const Invoice& invoice);
    Invoice loadInvoiceData(const string& invoiceID);
    bool findInvoiceByInvoiceID(const string& invoiceID , string& invoice);

    Vector<Customer> loadAllCustomersData();
    void saveAllCustomersData(const Vector<Customer>& customers);
    void addCustomer(const Customer& customer);
    void deleteCustomer(const string& customerID);
    Customer findCustomerById(const string& customerID);
    void updateCustomer(const Customer& customer);

    Vector<Manager> loadAllManagersData();
    void saveAllManagersData(const Vector<Manager>& );
    void addManager(const Manager& );
    void deleteManager(const string& );
    Manager findManagerById(const string& );
    void updateManager(const Manager&);

private:
    Product parseProduct(const string& line);
};
```

Cách thức hoạt động :

- Đọc dữ liệu từ tệp: Tệp sẽ được mở để đọc dữ liệu. Dữ liệu sẽ được đọc theo từng dòng hoặc từng phần tử.
- Chuyển đổi dữ liệu thành đối tượng: Dữ liệu đọc được sẽ được phân tích và chuyển đổi thành các đối tượng hoặc cấu trúc dữ liệu cần thiết cho hệ thống.
- Xử lý các dữ liệu không hợp lệ: Trong trường hợp gặp phải dữ liệu không hợp lệ, có thể bỏ qua hoặc ghi lại để xử lý sau.

❖ 4.1.3 Mô tả đầu ra :

Kết quả trả về là các đối tượng hoặc các thuộc tính cụ thể của đối tượng thỏa mãn yêu cầu cho trước.

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH

Ví dụ về quá trình và kết quả hoạt động của hàm saveAllCustomersData:

```
void DataController::saveAllCustomersData(const vector<Customer>& customers) {
    const string filename = "Data/customerInformation.csv";
    saveAllData<Customer>(customers, filename, [<] (const Customer& customer) {
        return customer.getUserId() + ";" +
            customer.getName() + ";" +
            customer.getEmail() + ";" +
            customer.getPhone() + ";" +
            customer.getPassword() + ";" +
            customer.getAddress();
    });
}
```

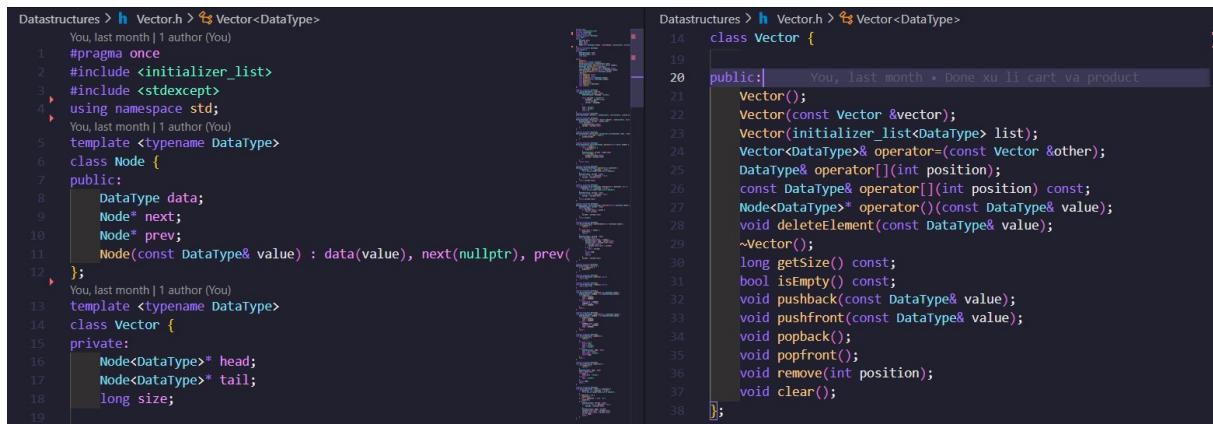
You, 4 weeks ago • thêm user customer

Data > CustomerInformation.csv > □ data

1 | USER001;Nguyễn Văn Huy;nhuhuy@gmail.com;0768406236;1222;15 Lê văn Hiến, Ngũ Hành Sơn, Đà Nẵng
2 | USER002;Trần Thị Thu Hồi;thuthu@gmail.com;0768406237;1333;12 Nguyễn Tri Phương, Thành Khê, Đà Nẵng
3 | USER003;Nguyễn Văn Hùng;hungnv@gmail.com;0768406237;1333;12 Nguyễn Tri Phương, Thành Khê, Đà Nẵng
4 | USER004;Phạm Minh Đức;minduc.pham@gmail.com;0768406238;4444;47 Tôn Đức Thắng, Liên Chiểu, Đà Nẵng
5 | USER005;Vũ Thị Thủ Hằng;hangvu@gmail.com;0768406239;5555;88 Phạm Văn Đồng, Hải Châu, Đà Nẵng
6 | USER006;Hoàng Văn Nam;namhoang@gmail.com;0768406240;6666;22 Hoàng Diệu, Hải Châu, Đà Nẵng
7 | USER007;Nguyễn Minh Tùng;tungminh@gmail.com;0768406241;7777;9 Hàm Nghi, Thành Khê, Đà Nẵng
8 | USER008;Lê Hồng Sơn;sonhong.soh@gmail.com;0768406242;8888;33 Nguyễn Văn Linh, Liên Chiểu, Đà Nẵng
9 | USER009;Nguyễn Văn Hùng;hungnv@gmail.com;0768406243;9999;55 Nguyễn Công Trứ, Sơn Trà, Đà Nẵng
10 | USER010;Trịnh Văn Hùng;hangtrinh@gmail.com;0768406244;1000;7 Trần Phú, Hải Châu, Đà Nẵng
11 | USER011;Đặng Văn Hoài;hoaidang@gmail.com;0768406245;1010;21 Nguyễn Văn Linh, Hải Châu, Đà Nẵng
12 | USER012;Lê Thị Mai;maitl@gmail.com;0768406246;1112;12 Ngô Quyền, Sơn Trà, Đà Nẵng
13 | USER013;Nguyễn Hoàng Phúc;phuchong@gmail.com;0768406247;1222;12 Lý Tự Trọng, Hải Châu, Đà Nẵng
14 | USER014;Phan Thị Minh Nguyệt;nguyetphan@gmail.com;0768406248;1333;12 Nguyễn Công Trứ, Sơn Trà, Đà Nẵng
15 | USER015;Lê Văn Hùng;hunglv@gmail.com;0768406249;1444;19 Ông Út Tát, Chợ Lớn, Hồ Chí Minh
16 | USER016;Nguyễn Thị Huyền;huanhuy@gmail.com;0768406250;1555;35 Hưng Vượng, Hải Châu, Đà Nẵng
17 | USER017;Nguyễn Thị Huyền;huanhuy@gmail.com;0768406251;1616;35 Hưng Vượng, Hải Châu, Đà Nẵng
18 | USER018;Lê Văn Phòng;phongle@gmail.com;0768406252;1717;48 Nguyễn Chí Thành, Hải Châu, Đà Nẵng
19 | USER019;Phạm Ngọc Quân;quannpham` --;1818;3 Hải Hồ, Hải Châu, Đà Nẵng
20 | USER020;Nguyễn Văn Hiệp;hiepgg@mail.com;0768406254;1919;29 Trung Nữ Vương, Hải Châu, Đà Nẵng
21 | USER021;Trần Văn Bình;bintran@gmail.com;0768406255;2020;10 Nguyễn Hữu Thọ, Hải Châu, Đà Nẵng
22 | USER022;Đoàn Kiều Ngân;nganng@gmail.com;0768406256;1111;67 Âu Cơ, Liên Chiểu, Đà Nẵng
23 | USER023;Nguyễn Thị Thanh Thảo;thanhthao@gmail.com;0768406257;2222;89 Liên Chiểu, Đà Nẵng
24 | USER001;Nguyễn Văn Trung Anh;trunganh@gmail.com;0768406258;2222;25 Duy Chất, Hòa Khánh Bắc, Liên Chiểu,

4.2 Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống:

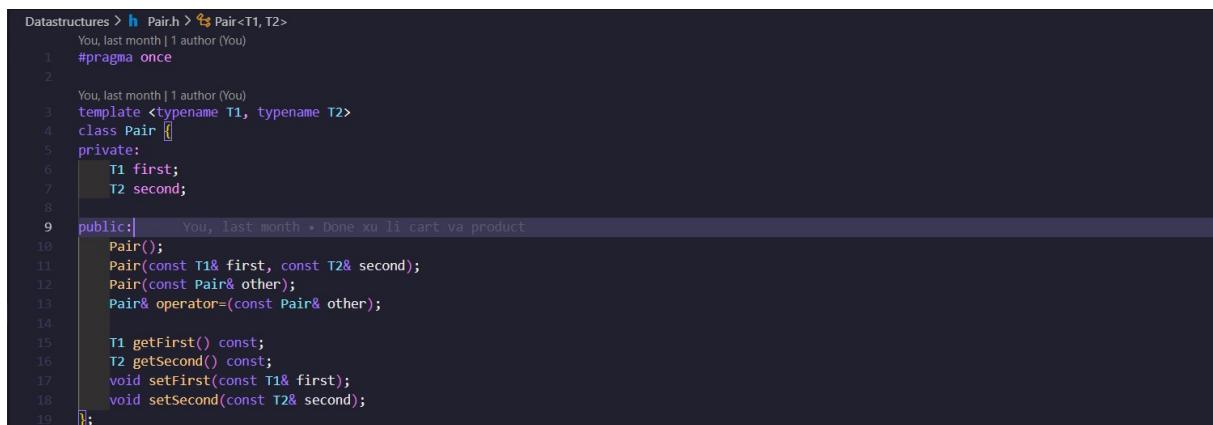
Xây dựng template Vector sử dụng danh sách liên kết đôi làm cấu trúc dữ liệu chính cho hệ thống:



The image shows two side-by-side code editors. The left editor contains the implementation of the `Node` class, which is a linked list node with fields for data, next, and prev pointers. The right editor contains the implementation of the `Vector` class, which is a linked list of nodes, providing methods for pushback, pushfront, remove, and clear.

```
Datastructures > h Vector.h > Vector.h
1 | You, last month | 1 author (You)
2 | #pragma once
3 | #include <initializer_list>
4 | #include <stdexcept>
5 | using namespace std;
6 | You, last month | 1 author (You)
7 | template <typename DataType>
8 | class Node {
9 | public:
10 |     DataType data;
11 |     Node* next;
12 |     Node* prev;
13 |     Node(DataType value) : data(value), next(nullptr), prev(nullptr) {}
14 | };
15 | You, last month | 1 author (You)
16 | template <typename DataType>
17 | class Vector {
18 | private:
19 |     Node<DataType>* head;
20 |     Node<DataType>* tail;
21 |     long size;
22 | };
23 |
24 | Datastructures > h Vector.h > Vector.h
25 | You, last month | Done xu li cart va product
26 | class Vector {
27 | public:
28 |     Vector();
29 |     Vector(const Vector &vector);
30 |     Vector(initializer_list<DataType> list);
31 |     Vector<DataType>& operator=(const Vector &other);
32 |     DataType& operator[](int position);
33 |     const DataType& operator[](int position) const;
34 |     Node<DataType>*& operator()(const DataType& value);
35 |     void deleteElement(const DataType& value);
36 |     ~Vector();
37 |     long getSize() const;
38 |     bool isEmpty() const;
```

Ngoài ra , để tối ưu trong quá trình thực thi hệ thống và xử lí dữ liệu , template Pair sẽ giải quyết hữu hiệu các dữ liệu theo cặp:



The image shows the implementation of the `Pair` class, which is a simple container for a key-value pair. It has private members `T1 first` and `T2 second`, and public methods for getting and setting both values.

```
Datastructures > h Pair.h > Pair.h
1 | You, last month | 1 author (You)
2 | #pragma once
3 | You, last month | 1 author (You)
4 | template <typename T1, typename T2>
5 | class Pair {
6 | private:
7 |     T1 first;
8 |     T2 second;
9 | public:
10 |     Pair();
11 |     Pair(const T1& first, const T2& second);
12 |     Pair(const Pair& other);
13 |     Pair& operator=(const Pair& other);
14 |
15 |     T1 getFirst() const;
16 |     T2 getSecond() const;
17 |     void setFirst(const T1& first);
18 |     void setSecond(const T2& second);
19 | };
```

5. ỨNG DỤNG HƯỚNG ĐỐI TƯỢNG VÀO BÀI TOÁN

5.1 Phân tích các lớp và đối tượng chính trong hệ thống

5.1.1 Chi tiết các lớp chính trong chương trình:

a) Lớp User :

Private	string userId	Mã định danh duy nhất của người dùng
Private	string name	Tên đầy đủ của người dùng
Private	string email	Địa chỉ email của người dùng
Private	string phone	Số điện thoại của người dùng
Private	string password	Mật khẩu của người dùng

b) Lớp Customer và Manager : kế thừa từ lớp User , sự khác nhau của 2 lớp đến từ phương thức (các hàm thành viên) . Bên cạnh đó , lớp Customer còn có các thuộc tính :

Private	Cart *cart	Giỏ hàng của khách hàng
Private	Orders *orderHistory	Danh sách đơn hàng của khách hàng

c) Lớp Product:

Private	string productId	Mã sản phẩm
Private	string name	Tên sản phẩm
Private	string category	Danh mục sản phẩm
Private	double price	Giá sản phẩm
Private	int stock	Số lượng sản phẩm trong kho
Private	string description	Đường dẫn mô tả sản phẩm
Private	Vector<string> detail	Chi tiết bổ sung của sản phẩm : màn hình , tần số quét , chip , ..
Private	string brand;	Thương hiệu của sản phẩm

d) Lớp Invoice:

Private	string invoiceId;	Mã hóa đơn
Private	string customerId;	Mã khách hàng
Private	Vector<Pair<Product*, int>> products;	Danh sách các sản phẩm trong hóa đơn (mỗi phần tử là một cặp bao gồm sản phẩm và số lượng)

Private	string invoiceDate;	Ngày lập hóa đơn
Private	double totalAmount;	Tổng số tiền của hóa đơn
Private	string deliveryDate;	Ngày giao hàng
Private	string paymentMethod;	Phương thức thanh toán

e) Lớp Orders : để thuận lợi cho công tác quản lý các hóa đơn , lớp Orders được xây dựng là 1 Vector<Invoice*> cùng với các phương thức để xử lí dữ liệu với hóa đơn đó.

- Trong mỗi lớp, để đảm bảo tính rõ ràng và dễ duy trì, ta sẽ triển khai các thành phần cơ bản như Constructor, Destructor, các phương thức Setter, Getter, và những phương thức đặc thù (business logic) riêng biệt của từng lớp.

5.1.2 Chi tiết các lớp hỗ trợ :

- a) Lớp DataController:** Như đã đề cập trước đó , lớp DataController phụ trách các công việc liên quan đến xử lí dữ liệu , ghi đọc dữ liệu từ file .
- b) Lớp AppController :**Đối với việc xác thực và phân quyền người dùng , lớp AppController sẽ đảm nhiệm cùng với các chức năng sắp xếp dữ liệu theo 1 định dạng cho trước.

```
Controller > h AppController.h ...
You, 2 weeks ago | author (You)
1 #pragma once
2 #include "Model/Product.h"
3 #include "Model/Cart.h"
4 #include "Model/Invoice.h"
5 #include "Model/Orders.h"
6 #include "Model/Customer.h"
7 #include "Model/Manager.h"
8 #include "Datastructures/Vector.h"
9 #include "Datastructures/Pair.h"
10 #include "Controller/DataController.h"
11 #include <string>
12 using namespace std;
You, 2 weeks ago | author (You)
13 class AppController{
14 public:
15     Pair<string,string> allogin(const string& email , const string& password);
16     string login(const string& email , const string& password,const string& role);
17     string signin( const string& name, const string& email, const string& phone, const string& password, const string& address,const string& role);
18     Vector<Invoice*> sortInvoicebyDate();
19     Vector<Invoice*> loadAllInvoice();
20     Vector<Pair<string,int>> sortSoldProducts();
21     Vector<Pair<Customer, double>> sortCustomerByAmount();
22     string createProductId();
23 };
You, 2 weeks ago * thêm user customer
```

c) Các lớp liên quan đến xử lí giao diện:

i. Lớp LoginSignupInterface:

```
View > h LoginSignupInterface.h ...
You, 2 weeks ago | author (You)
1 #pragma once
2 #include <QWidget>
3 #include <QLineEdit>
4 #include <QPushButton>
5 #include <QCheckBox>
6 #include <QLabel>
7 #include <QStackedWidget>
8 #include <QVBoxLayout>
9 #include <QGroupBox>
10 #include "Controller/DataController.h"
11 #include "Controller/AppController.h"
12
You, 2 weeks ago | author (You)
13 class LoginSignupInterface : public QWidget {
14     Q_OBJECT
15
16 public:
17     explicit LoginSignupInterface(QWidget *parent = nullptr);
18     void showMessage(QWidget *parent, bool status, const QString &message);
19     ~LoginSignupInterface();
20     private slots:
21     void showSignupPage();
22     void showLoginPage();
23     void login();
24     void signup();
25     private:

```

```
View > h LoginSignupInterface.h ...
You, 2 weeks ago | author (You)
13 class LoginSignupInterface : public QWidget {
25     private:
26     QLineEdit *emailInput;
27     QLineEdit *passwordInput;
28     QPushButton *loginButton;
29     QLabel *loginImage;
30     QLineEdit *nameInput;
31     QLineEdit *signupEmailInput;
32     QLineEdit *phoneInput;
33     QLineEdit *signupPasswordInput;
34     QLineEdit *confirmSignupPasswordInput;
35     QLineEdit *addressInput;
36     QPushButton *signupButton;
37     QLabel *signupImage;
38     QStackedWidget *stackedWidget;
39     QPushButton *switchToSignupButton;
40     QPushButton *switchToLoginButton;
41     DataController *datacontroller;
42     AppController *appcontroller;
43 };
```

ii. Lớp ManagerInterface:

```

View > h ManagerInterface.h > ManagerInterface
1 #include <QMap>
2 #include <QString>
3 #include "Model/Cart.h"
4 #include "Model/Manager.h"
5 #include "Model/Invoice.h"
6 #include "Controller/DataController.h"
7 #include "Controller/AppController.h"
8 #include <string>
9 #include <QComboBox>
10 #include <QMessageBox>
11 #include "Model/Customer.h"
12 using namespace std;
13 You, 2 weeks ago | 1 author (You)
14 class ManagerInterface : public QWidget {
15     Q_OBJECT
16
17 public:
18     ManagerInterface(QWidget *parent = nullptr, const string &managerid = "");
19     void showMessage(QWidget *parent, bool status, const QString &message);
20     void drawChart(const Vector<Pair<QString, double>>& data, const QString& chartType);
21     ~ManagerInterface();
22     private slots:
23     void showOverview();
24     void showProducts();
25     void filterProducts();
26     void checkout();
27     void showAccount(bool change = false);
28     void showCustomers();
29     void showInvoices();
30     void filterCustomer();
31     void filterInvoice();
32     void addNewProduct();
33     void showAmountByDate();
34     void showProductId();
35     void showCustomerByAmount();
36     void showSoldProduct();
37     signals:
38     private:
39
View > h ManagerInterface.h > ManagerInterface
1 ManagerInterface : public QWidget {
2     signals:
3     Manager* manager;
4     QPushbutton *showOverviewButton;
5     QPushbutton *showProductsButton;
6     QPushbutton *checkoutButton;
7     QPushbutton *showCustomersButton;
8     QPushbutton *showInvoicesButton;
9     QLineEdit *searchProductLine;
10    QLineEdit *searchCustomerLine;
11    QLineEdit *searchInvoiceLine;
12    QPushbutton *showAccountButton;
13    QStackedWidget *stackWidget;
14    QStackedWidget *diagramStack;
15    QTextEdit *invoiceDisplay;
16    QTableWidget *productTable;
17    QTableWidget *customerTable;
18    QTableWidget *invoicesTable;
19    QGroupBox *overviewBox;
20    QGroupBox *managerInfoBox;
21    QPushbutton *addNewProductButton;
22    QGroupBox *diagramBox;
23    QPushbutton *applyButton;
24    QPushbutton *changeButton;
25    void addProductsData();
26    void addCustomersData();
27    void addInvoicesData();
28    void deleteProductById(const QString& productId);
29    void updateProductById(const QString& productId);
30    void deleteCustomer(int row);
31    void deleteDetailsProducts(int row);
32    void showInvoiceDetail(int row);
33    DataController * dataController;
34    AppController * appController;
35 };

```

iii. Lớp CustomerInterface:

```

View > h CustomerInterface.h > CustomerInterface
1 #include <QWidget>
2 #include <QLabel>
3 #include <QString>
4 #include "Model/Cart.h"
5 #include "Model/Customer.h"
6 #include "Model/Invoice.h"
7 #include <string>
8 #include <QComboBox>
9 #include <QMessageBox>
10 #include "Controller/DataController.h"
11 using namespace std;
12 You, 2 weeks ago | 1 author (You)
13 class CustomerInterface : public QWidget {
14     Q_OBJECT
15
16 public:
17     CustomerInterface(QWidget *parent = nullptr, const string &customerid = "");
18     void showMessage(QWidget *parent, bool status, const QString &message);
19     void drawChart(const Vector<Pair<QString, double>>& data, const QString& chartType);
20     ~CustomerInterface();
21     private slots:
22     void showOverview();
23     void showProducts();
24     void filterProducts();
25     void showCart();
26     void showOrders();
27     void showAccount(bool change = false);
28     void showInvoice();
29     void payment();
30     void checkout();
31     void cartOrigin();
32     void ordersOrigin();
33     void clearCart();
34     void onDeliveryDateChanged();
35     void onPaymentMethodChanged();
36     signals:
37     private:
38
View > h CustomerInterface.h > CustomerInterface
1 CustomerInterface : public QWidget {
2     signals:
3     private:
4         Customer* customer;
5         QPushbutton *showOverviewButton;
6         QPushbutton *showProductsButton;
7         QPushbutton *checkoutButton;
8         QPushbutton *showCartButton;
9         QPushbutton *showOrdersButton;
10        QPushbutton *showInvoicesButton;
11        QPushbutton *showAccountButton;
12        QStackedWidget *stackWidget;
13        QLineEdit *searchLineEdit;
14        QTextEdit *invoiceDisplay;
15        QPushbutton *paymentButton;
16        QPushbutton *clearCartButton;
17        QPushbutton *placeOrderButton;
18        QTableWidget *productTable;
19        QTableWidget *cartTable;
20        QTableWidget *ordersTable;
21        QLabel *totalPrice;
22        QLabel *dateLabel;
23        QDateEdit *deliveryDateEdit;
24        QComboBox *paymentMethodComboBox;
25        QLabel *paymentMethodLabel;
26        QGroupBox *overviewBox;
27        QGroupBox *customerInfoBox;
28        DataController *dataController;
29        QGroupBox *diagramBox;
30        QPushbutton *applyButton;
31        QPushbutton *changeButton;
32        void addProducts(int row, bool fromCart);
33        void deleteProducts(int row, bool fromCart);
34        void showDetailsProducts(int row);
35        void addProductsData();
36        void showInvoiceDetail(int row);
37 };

```

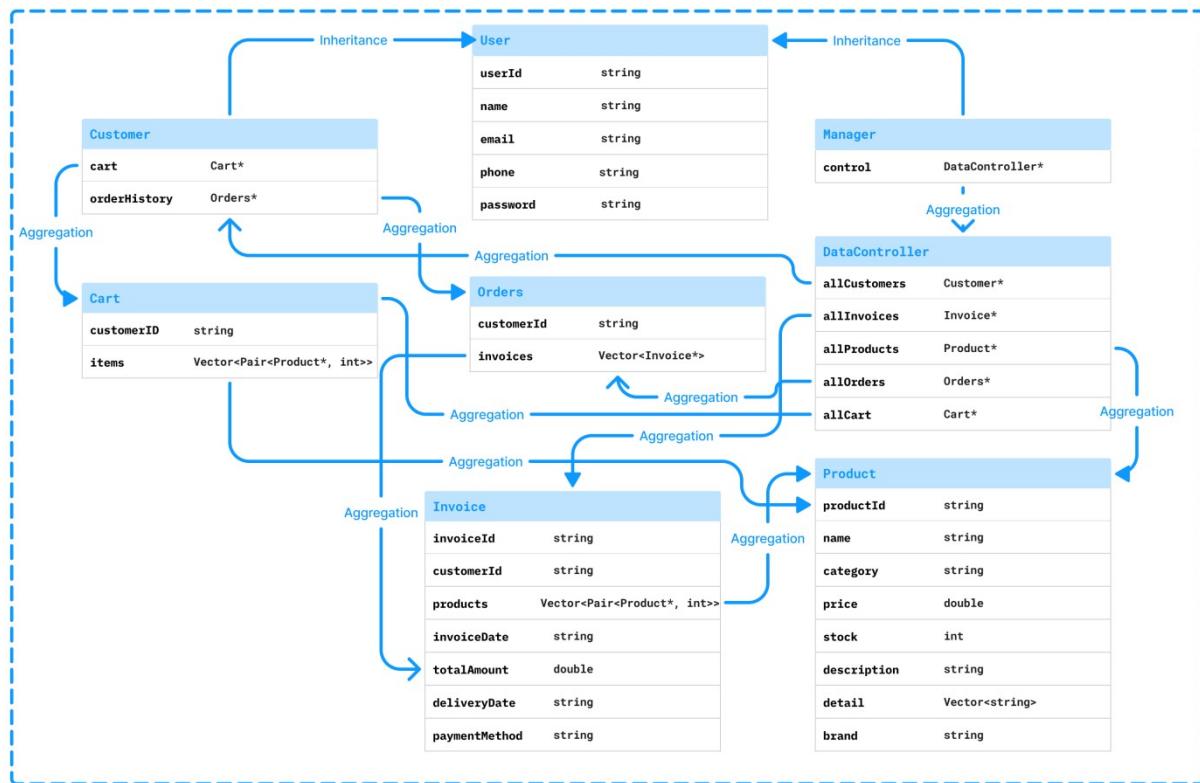
iv. Lớp AddNewProductInterface:

```

View > h AddNewProduct.h > ...
1 #pragma once
2
3 #include <QWidget>
4 #include <QLineEdit>
5 #include <QSpinBox>
6 #include <QDoubleSpinBox>
7 #include <QPushButton>
8 #include <QVBoxLayout>
9 #include <QFormLayout>
10 #include <QString>
11 #include <QStringList>
12 #include "Model/Product.h"
13 #include "Controller/DataController.h"
14 #include "Controller/AppController.h"
15
16 You, 2 weeks ago | 1 author (You)
17 class AddProductWidget : public QWidget {
18     Q_OBJECT
19
20 public:
21     explicit AddProductWidget(QWidget *parent = nullptr, Product *product = new Product());
22     void showMessage(QWidget *parent, bool status, const QString &message);
23     private slots:
24     void onOkButtonClicked();
25     void onCancelButtonClicked();
26     signals:
27     void productAdded();
28     private:
29         Product *product;
30         QLineEdit *nameEdit;
31         QLineEdit *categoryEdit;
32         QDoubleSpinBox *priceSpin;
33         QSpinBox *stockSpin;
34         QLineEdit *descriptionEdit;
35         QLineEdit *detailEdit;
36         QLineEdit *brandEdit;
37 };

```

5.2 Phân tích mối liên hệ giữa các lớp và đối tượng:



Mối quan hệ kế thừa (Inheritance) :

Lớp Manager và Customer kế thừa từ lớp User. Kế thừa cho phép một lớp con sử dụng lại các thuộc tính và phương thức đã được định nghĩa trong lớp cha

Mối quan hệ kết tập (Aggregation):

Các class trong hệ thống quan hệ với nhau chủ yếu ở mối quan hệ Aggregation vì:

- + Dùng con trỏ : Đối tượng thuộc lớp này không sở hữu các đối tượng thuộc lớp khác. Nó chỉ tham chiếu đến các đối tượng thuộc lớp khác được tạo và quản lý ở nơi khác
- + Tách biệt vòng đời: Nếu đối tượng lớp này bị hủy, các đối tượng thuộc lớp khác mà nó tham chiếu đến sẽ không bị hủy theo.

6. CHƯƠNG TRÌNH VÀ KẾT QUẢ

6.1 Tổ chức chương trình :

Ngôn ngữ lập trình : C++

Thư viện hỗ trợ : Qt

Ngoài ra , chương trình sử dụng :

- + resoure.qsc : quản lí tài nguyên hệ thống.
- + style.qss : tệp định kiểu cho giao diện người dùng.
- + MakeFile : tệp định nghĩa các lệnh để biên dịch và liên kết chương trình.

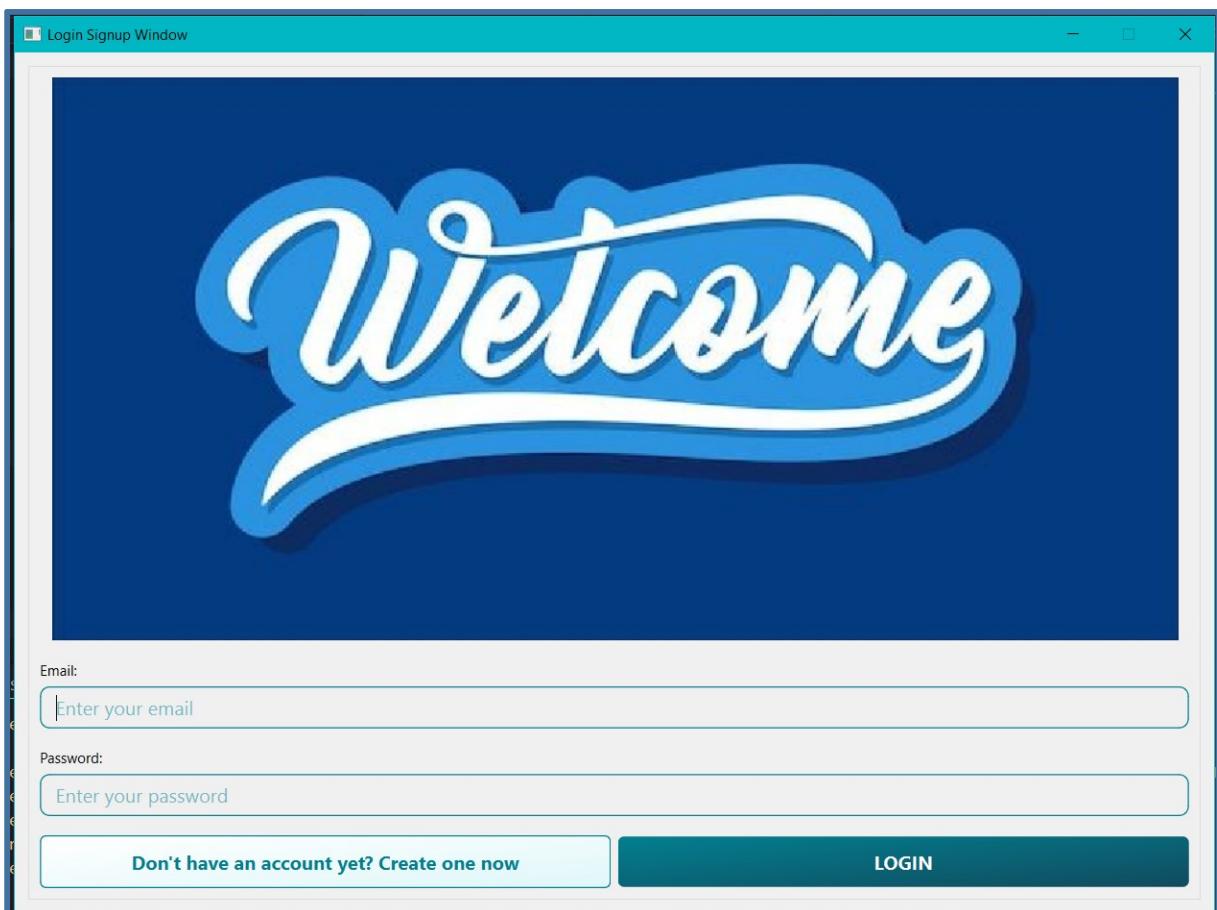
6.2 Kết quả thực thi chương trình:

I .Giao diện phía khách hàng:

1. Giao diện đăng nhập :

a) Giao diện đăng nhập :

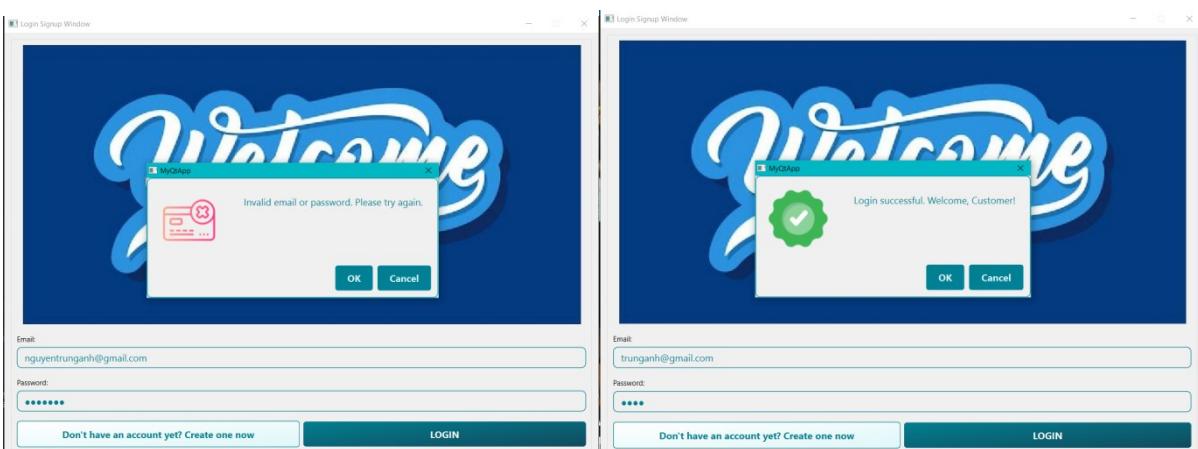
Khi khởi chạy, giao diện đăng nhập/đăng ký xuất hiện. Người dùng nhập email, mật khẩu hoặc chọn Create one now để tạo tài khoản mới



Hình 1. Giao diện đăng nhập

b) Thông báo đăng nhập thành công - không thành công:

Nếu nhập sai, thông báo Invalid email or password sẽ hiển thị. Nếu nhập đúng, thông báo Login successful. Welcome, Customer! xác nhận đăng nhập thành công.

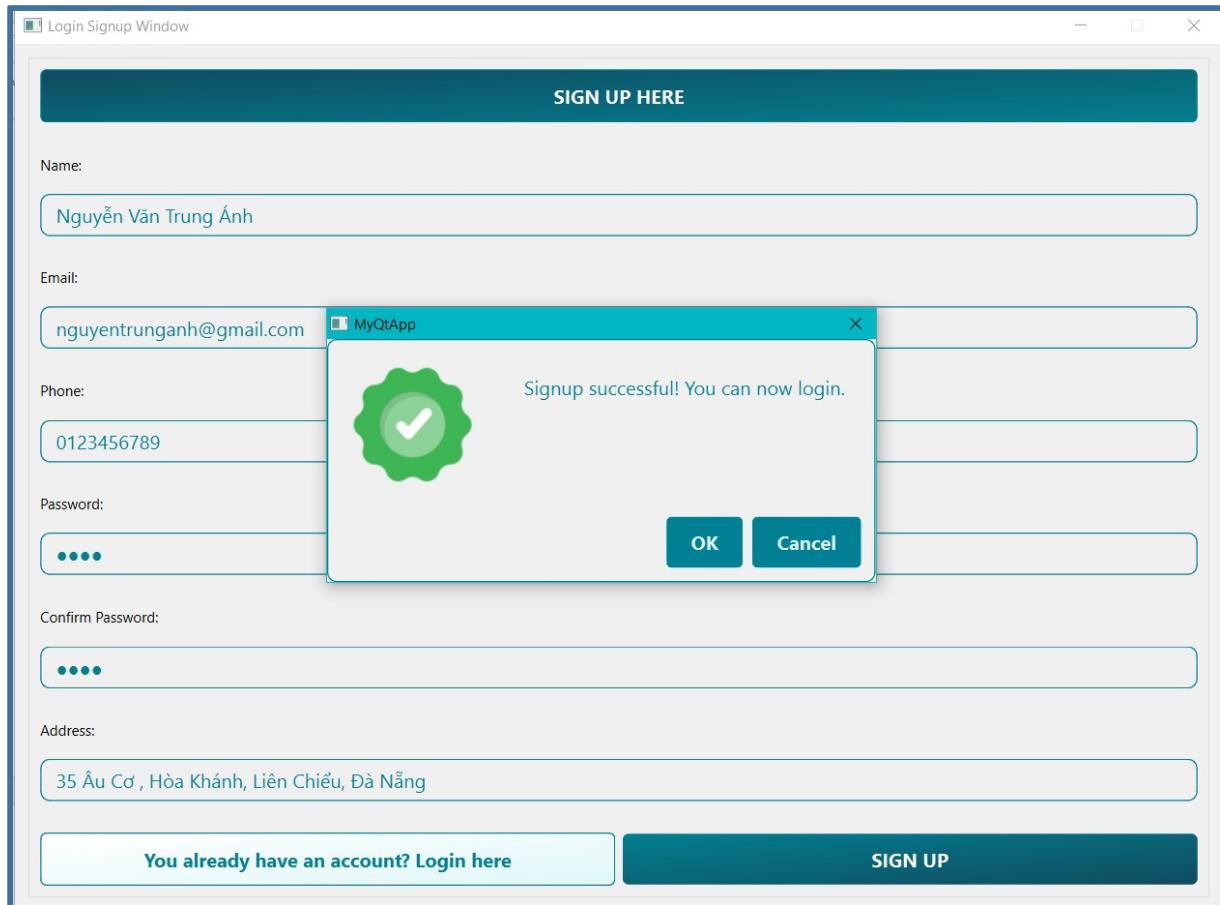


Hình 2. Giao diện đăng nhập không thành công

2. Giao diện đăng ký :

a) Giao diện đăng ký thành công:

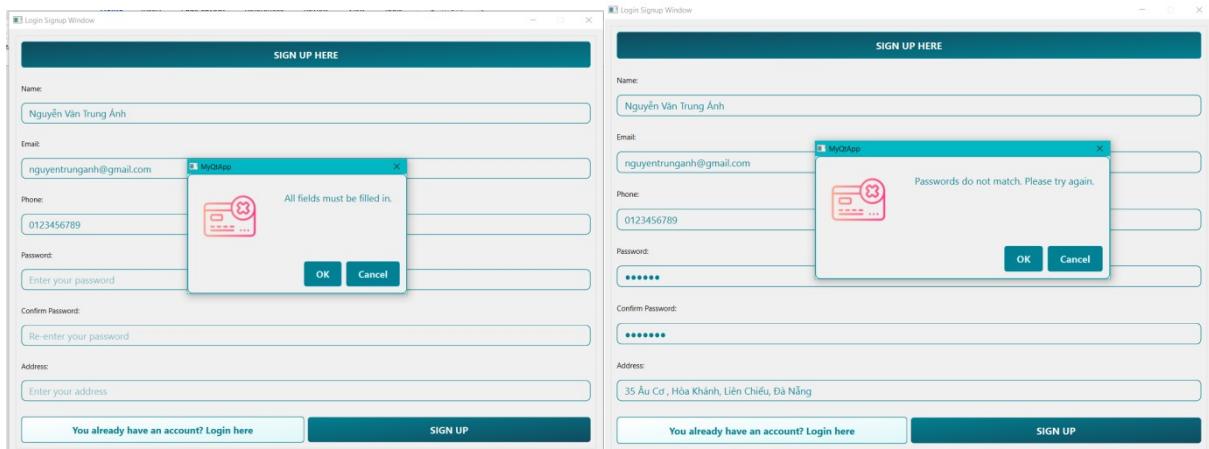
Sau khi người dùng nhập đầy đủ và chính xác thông tin đăng ký, hệ thống xử lý và hiển thị thông báo 'Signup successful! You can now login.' để xác nhận quá trình đăng ký đã hoàn tất thành công.



Hình 3. Giao diện đăng ký thành công

b) Thông báo lỗi đăng ký - đăng ký không thành công:

Nếu nhập thiếu hoặc sai thông tin đăng ký (email không hợp lệ, mật khẩu không khớp, hoặc đê trống), hệ thống sẽ báo lỗi “All fields must be filled in” hoặc “Passwords do not match. Please try again” và yêu cầu nhập lại.



Hình 4. Thông báo đăng ký không thành công

3. Giao diện chính cửa hàng:

Sau khi đăng nhập thành công, giao diện hiển thị thanh bên trái với các tùy chọn: Overview, Products, Cart, Orders, Account, Logout. Giao diện khởi đầu sẽ là danh sách các sản phẩm trong cửa hàng.

No.	Description	Product ID	Product Name	Price	Quantity	Add to Cart
4		PRD004	Điện thoại iPhone 13 Pro	1200	250	
5		PRD005	Điện thoại iPhone 13	950	500	
6		PRD006	Điện thoại Apple iPhone SE (2022)	500	0	
7		PRD007	Điện thoại Oppo Find X3	700	58	

Hình 5. Giao diện chính cửa hàng

4. Xem thông tin chi tiết sản phẩm:

Nếu người dùng muốn xem chi tiết sản phẩm và nhấn vào nút "Xem thêm thông tin"

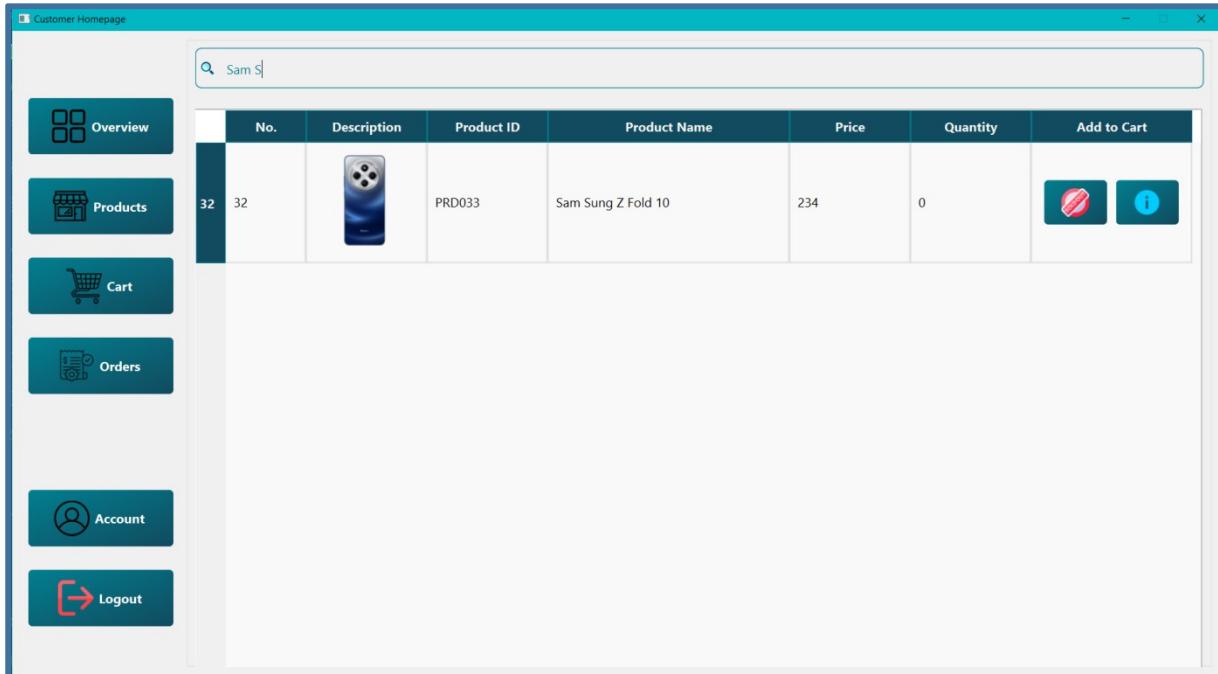
- i Hệ thống hiển thị chi tiết sản phẩm. Các thông tin chi tiết về sản phẩm bao gồm thông số kỹ thuật, thương hiệu và các tùy chọn mua hàng.

No.	Description	Product ID	Product Name	Price	Quantity	Add to Cart
4		PRD004	Điện thoại iPhone 13 Pro	1200	250	
5		PRD005	Điện thoại iPhone 13	950	500	
6		PRD006	Điện thoại Apple iPhone SE (2022)	500	0	
7		PRD007	Điện thoại Oppo Find X3	700	58	

Hình 6. Xem thông tin chi tiết sản phẩm

5. Tìm kiếm sản phẩm theo tên sản phẩm :

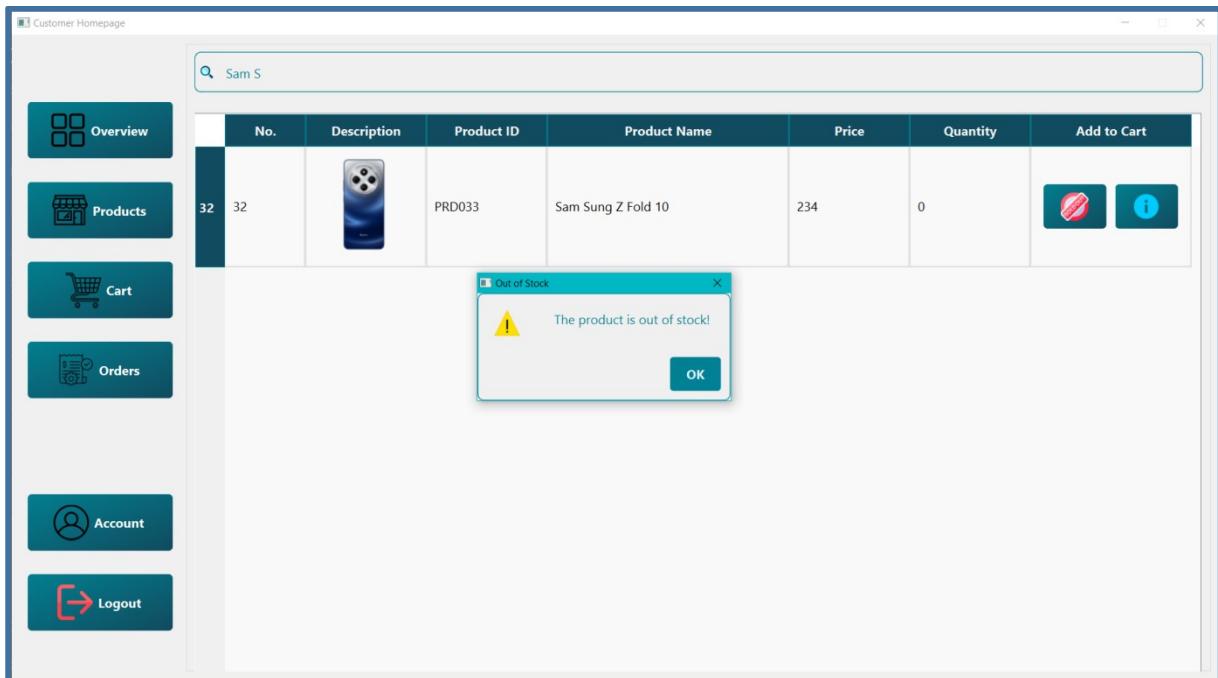
Sau khi người dùng nhập từ khóa vào thanh tìm kiếm , giao diện sẽ hiển thị kết quả tìm kiếm sản phẩm tương ứng, bao gồm các thông tin chi tiết về sản phẩm như tên sản phẩm, mã sản phẩm, giá và số lượng có sẵn.



Hình 7. Tìm kiếm sản phẩm theo tên sản phẩm

6. Thêm sản phẩm vào giỏ hàng (không thành công):

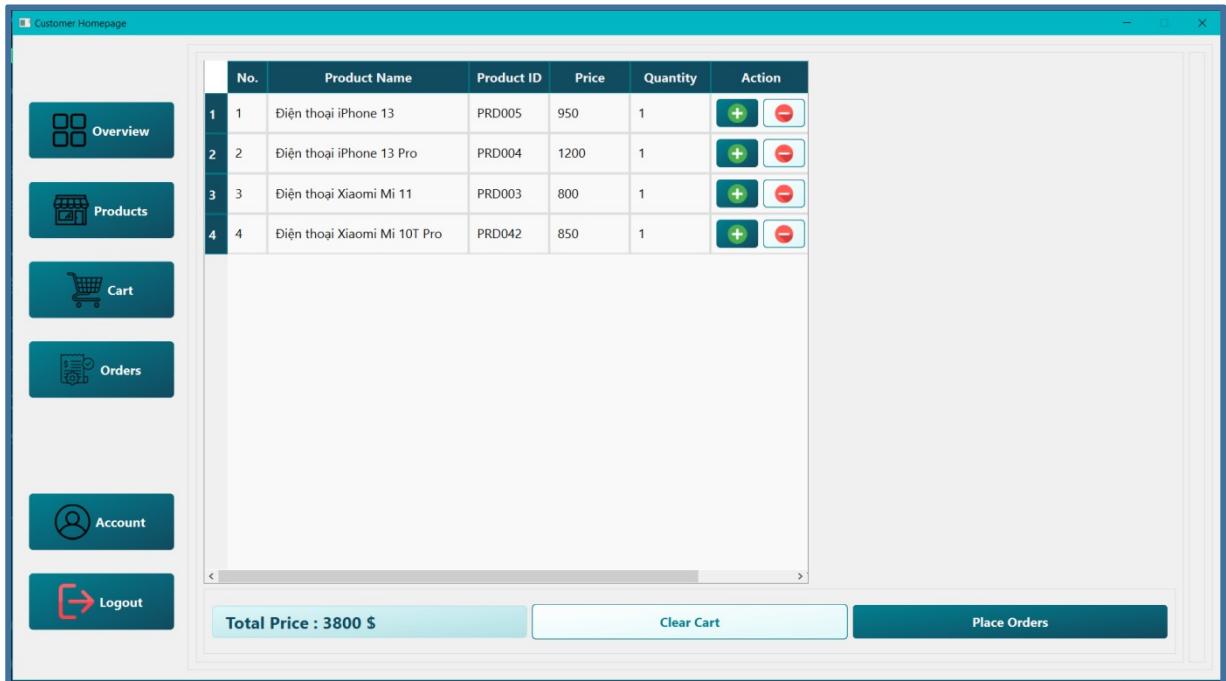
Nếu khách hàng muốn thêm sản phẩm vào giỏ hàng và nhấn nút "Thêm sản phẩm vào giỏ hàng". Tuy nhiên, sản phẩm này hiện đang hết hàng thì hệ thống sẽ hiển thị một thông báo lỗi với nội dung "The product is out of stock!"



Hình 8. Thêm sản phẩm vào giỏ hàng (không thành công)

7. Thêm sản phẩm vào giỏ hàng (thành công):

Khi nhấn Thêm vào giỏ hàng thành công, giao diện hiển thị danh sách sản phẩm trong giỏ, gồm số thứ tự, tên, mã, giá, số lượng, cùng các nút tăng/giảm số lượng.



The screenshot shows a customer dashboard titled "Customer Homepage". On the left sidebar, there are buttons for Overview, Products, Cart, Orders, Account, and Logout. The main area displays a table of products in the cart:

No.	Product Name	Product ID	Price	Quantity	Action
1	Điện thoại iPhone 13	PRD005	950	1	+/-
2	Điện thoại iPhone 13 Pro	PRD004	1200	1	+/-
3	Điện thoại Xiaomi Mi 11	PRD003	800	1	+/-
4	Điện thoại Xiaomi Mi 10T Pro	PRD042	850	1	+/-

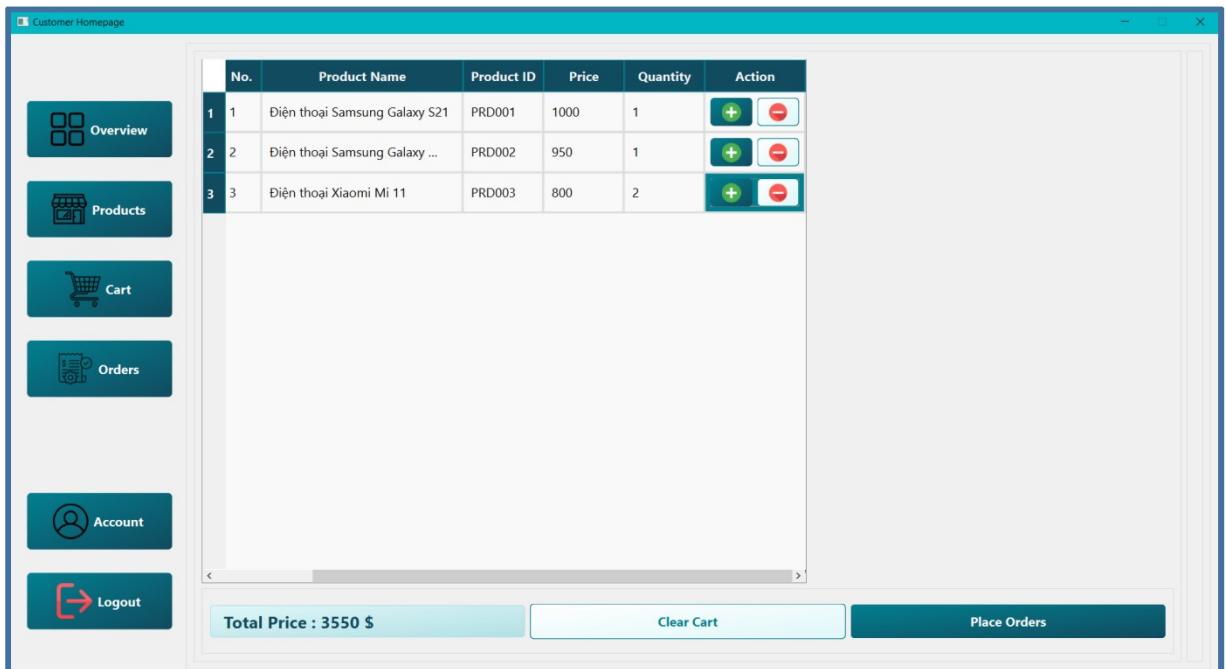
At the bottom, it shows "Total Price : 3800 \$" and buttons for "Clear Cart" and "Place Orders".

Hình 9. Thêm sản phẩm vào giỏ hàng (thành công)

8. Thêm, giảm, xóa sản phẩm khỏi giỏ hàng:

Khi nhấn Thêm (+), số lượng sản phẩm trong giỏ tăng 1.

Khi nhấn Giảm (-), số lượng giảm 1. Nếu số lượng về 0, sản phẩm sẽ bị xóa khỏi giỏ hàng.



The screenshot shows a customer dashboard titled "Customer Homepage". On the left sidebar, there are buttons for Overview, Products, Cart, Orders, Account, and Logout. The main area displays a table of products in the cart:

No.	Product Name	Product ID	Price	Quantity	Action
1	Điện thoại Samsung Galaxy S21	PRD001	1000	1	+/-
2	Điện thoại Samsung Galaxy ...	PRD002	950	1	+/-
3	Điện thoại Xiaomi Mi 11	PRD003	800	2	+/-

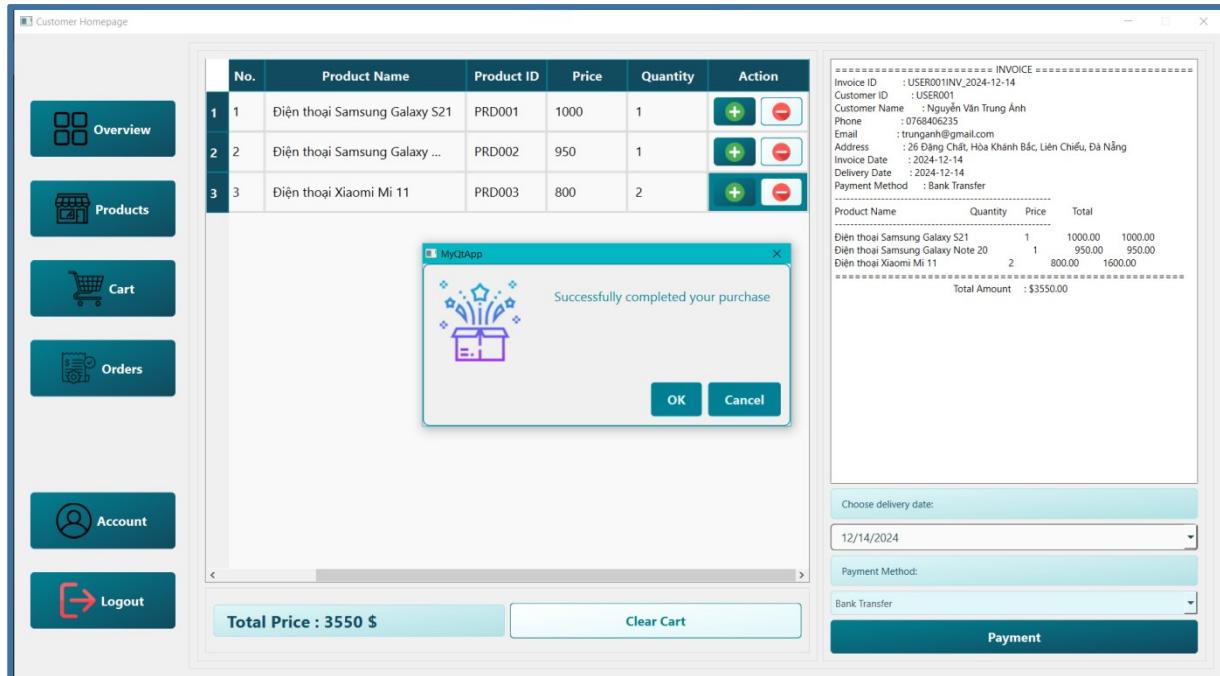
At the bottom, it shows "Total Price : 3550 \$" and buttons for "Clear Cart" and "Place Orders".

Hình 10. Thêm, giảm, xóa sản phẩm khỏi giỏ hàng:

9. Chọn ngày giao hàng, phương thức thanh toán và đặt hàng:

Sau khi khách hàng chọn các sản phẩm muốn mua và thêm vào giỏ hàng, giao diện sẽ chuyển đến trang thanh toán. Trên trang này, người dùng có thể chọn ngày giao hàng, phương thức thanh toán và nhấn vào nút "Payment" để hoàn tất quá trình đặt hàng.

Giao diện sẽ hiển thị thông báo xác nhận rằng đơn hàng đã được đặt thành công và cung cấp mã đơn hàng để theo dõi.



Hình 11. Chọn ngày giao hàng, phương thức thanh toán và đặt hàng

10. Xem tất cả đơn hàng đã đặt :

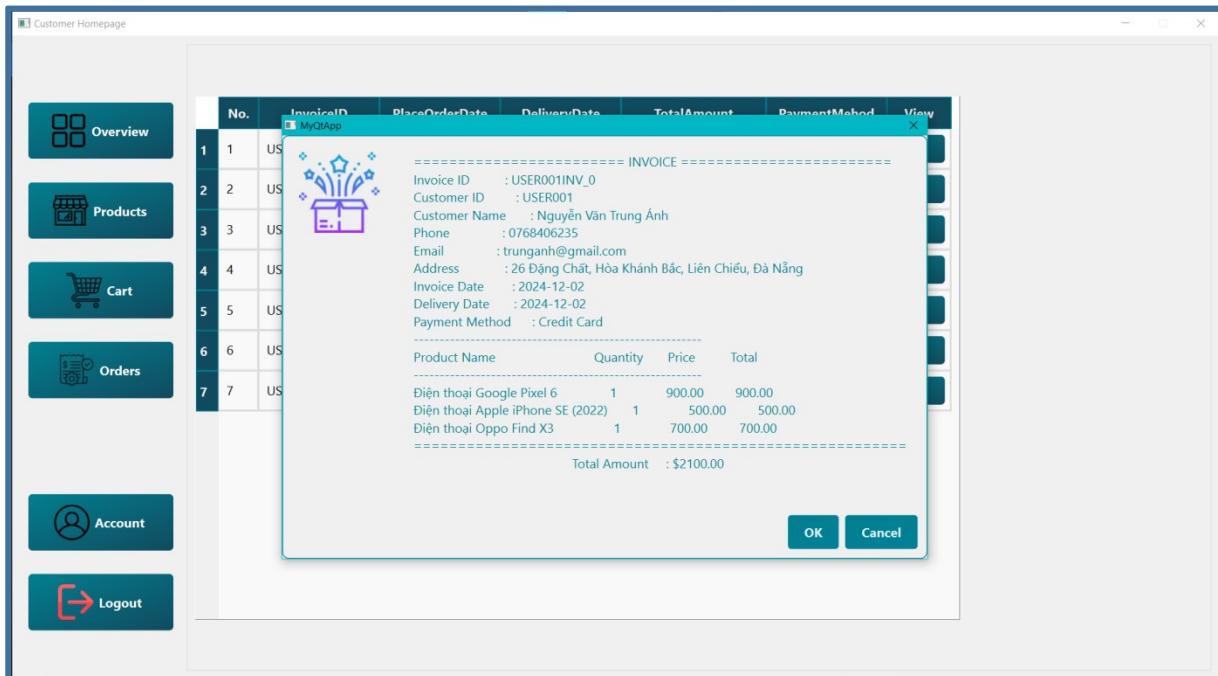
Sau khi khách hàng chọn mục "Orders" trên thanh điều hướng, giao diện sẽ hiển thị danh sách tất cả các đơn hàng mà người dùng đã đặt.

No.	InvoiceID	PlaceOrderDate	DeliveryDate	TotalAmount	PaymentMehod	View
1	USER001INV_0	2024-12-02	2024-12-02	2100	Credit Card	
2	USER001INV_1	2024-12-02	2024-12-13	3100	Bank Transfer	
3	USER001INV_2	2024-12-02	2024-12-20	1800	Bank Transfer	
4	USER001INV_3	2024-12-05	2024-12-05	2100	Bank Transfer	
5	USER001INV_4	2024-12-05	2025-01-17	4200	Bank Transfer	
6	USER001INV_5	2024-12-08	2024-12-21	4200	Bank Transfer	
7	USER001INV_6	2024-12-14	2024-12-14	3550	Bank Transfer	

Hình 12. Xem tất cả đơn hàng đã đặt

11. Xem chi tiết hóa đơn :

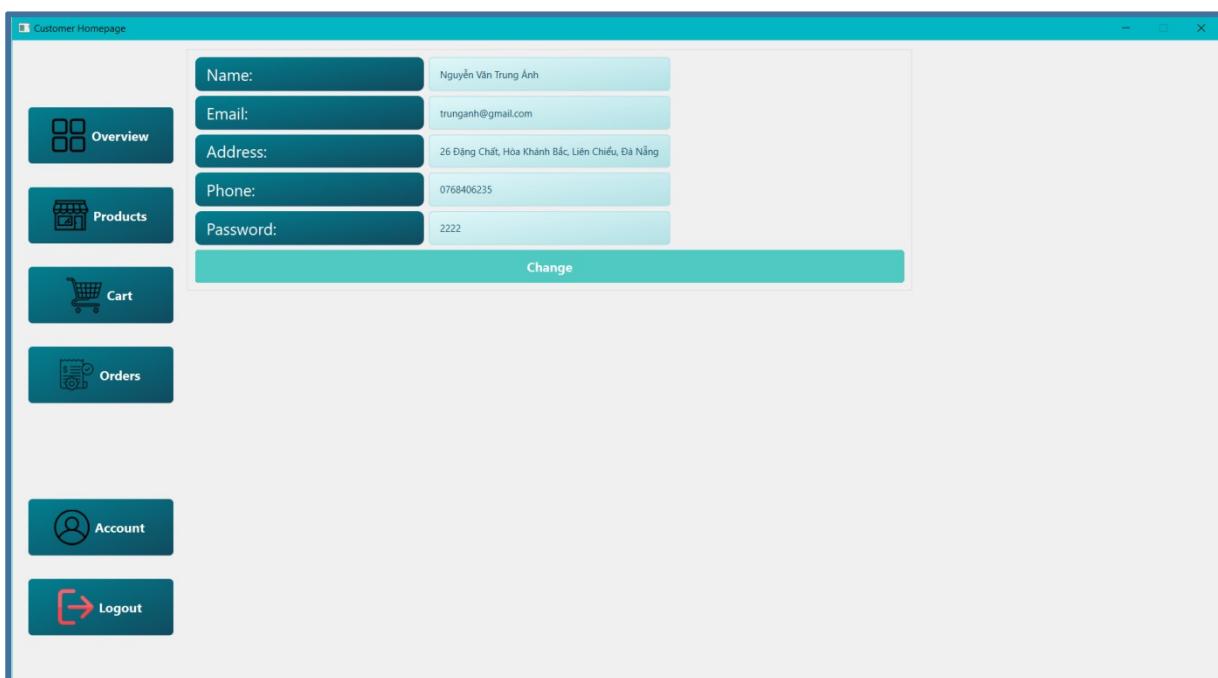
Sau khi khách hàng chọn mục "Orders" trên thanh điều hướng và nhấn vào nút "Xem chi tiết"  bên cạnh mã hóa đơn, giao diện sẽ hiển thị thông tin chi tiết về hóa đơn của khách hàng.



Hình 13. Xem chi tiết hóa đơn

12. Xem thông tin tài khoản:

Sau khi người dùng chọn mục "Account" trên thanh điều hướng, giao diện sẽ hiển thị thông tin tài khoản của họ bao gồm tên, email, địa chỉ, số điện thoại, mật khẩu. Bên cạnh đó, người dùng thay đổi thông tin của mình qua nút “Change”.

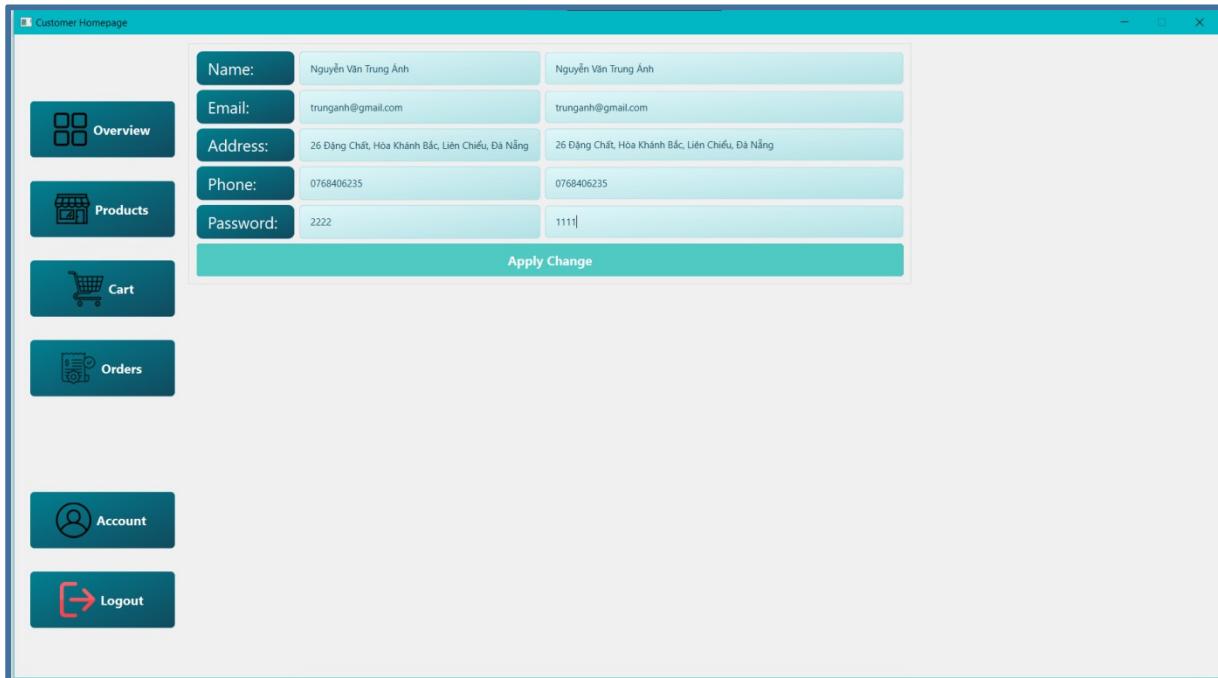


Hình 14. Xem thông tin tài khoản

PBL2 : ĐỒ ÁN CƠ SỞ LẬP TRÌNH

13. Thay đổi thông tin tài khoản:

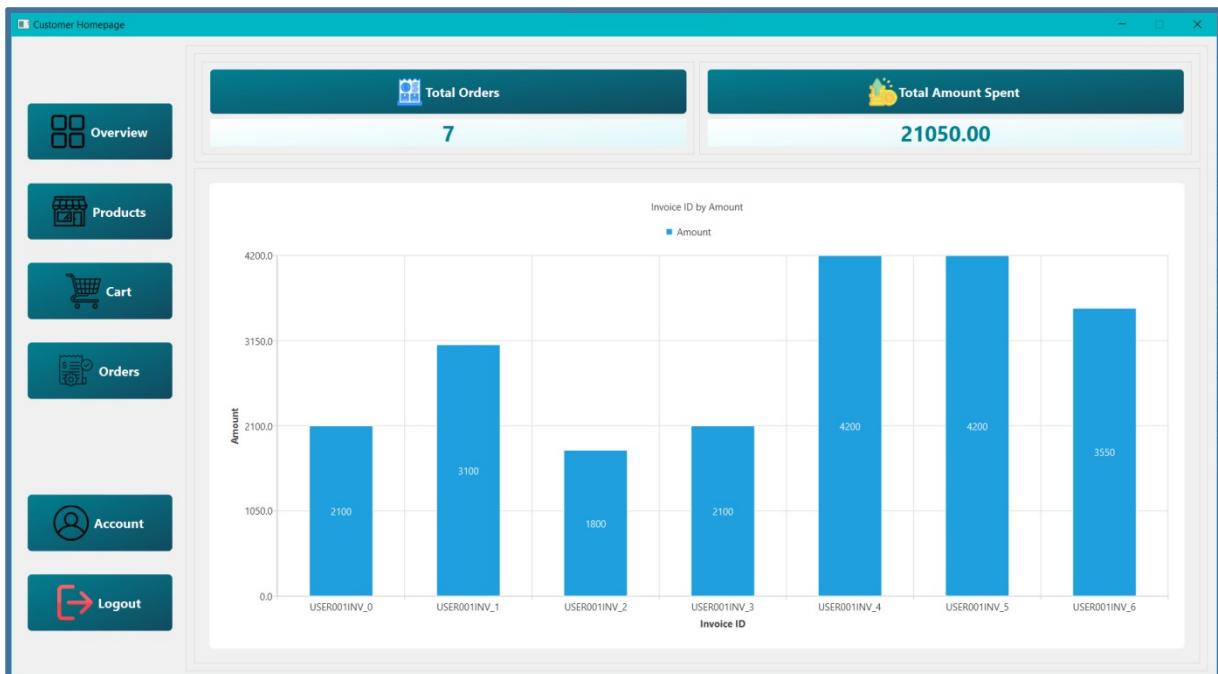
Người dùng có thể nhấn vào nút "Apply Change" để thay đổi thông tin tài khoản của mình như tên, email, địa chỉ, số điện thoại và mật khẩu.



Hình 15. Thay đổi thông tin tài khoản

14. Xem tổng quan , thống kê :

Sau khi người dùng chọn mục "Overview" trên thanh điều hướng, giao diện sẽ hiển thị trang tổng quan và thống kê hoạt động của người dùng trên nền tảng thương mại điện tử bao gồm tổng số đơn hàng, tổng số tiền đã chi tiêu và biểu đồ cột hiển thị số tiền đã chi tiêu cho từng mã hóa đơn.

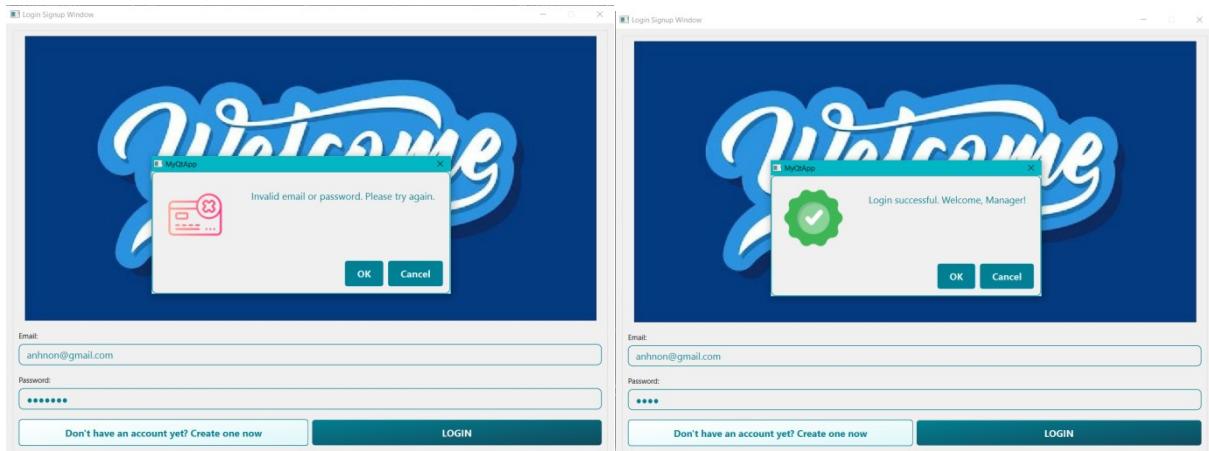


Hình 16. Xem tổng quan , thống kê

II. Giao diện phía quản lý:

1. Giao diện đăng nhập :

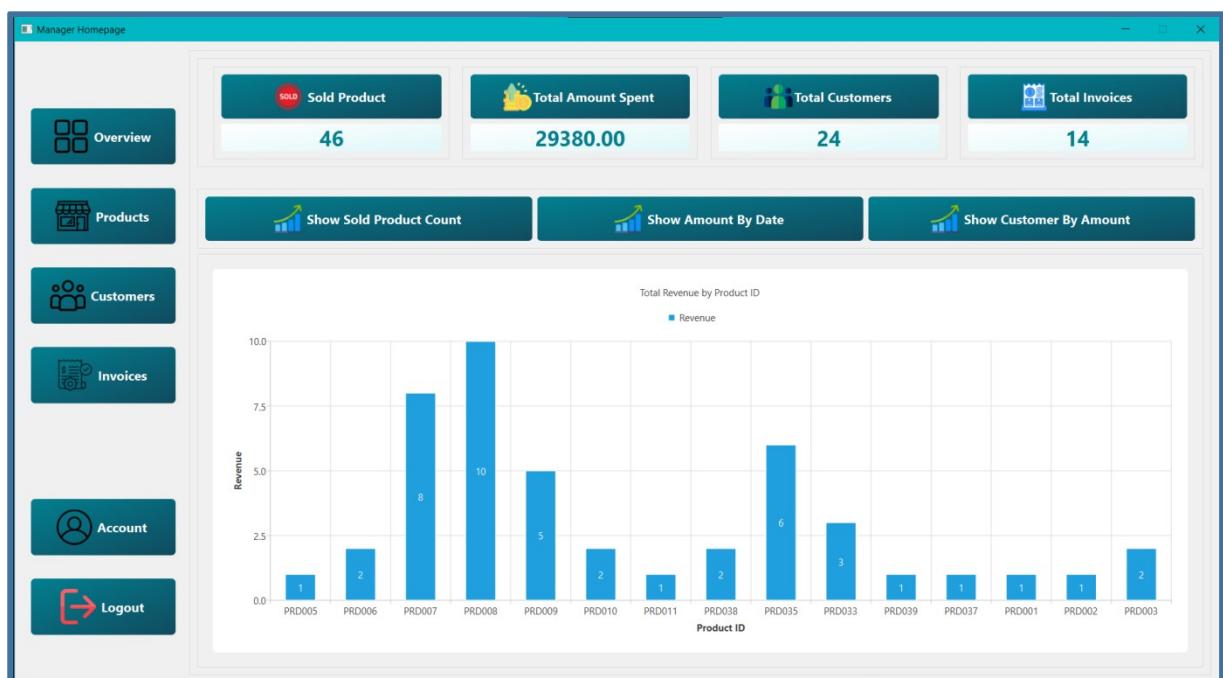
Nếu nhập sai thông tin đăng nhập, giao diện báo lỗi Invalid email or password. Please try again. Nếu nhập đúng, thông báo Login successful. Welcome, Manager! sẽ xuất hiện, xác nhận đăng nhập thành công.



Hình 17. Giao diện đăng nhập (manager)

2. Giao diện tổng quan, thống kê sản phẩm bán được :

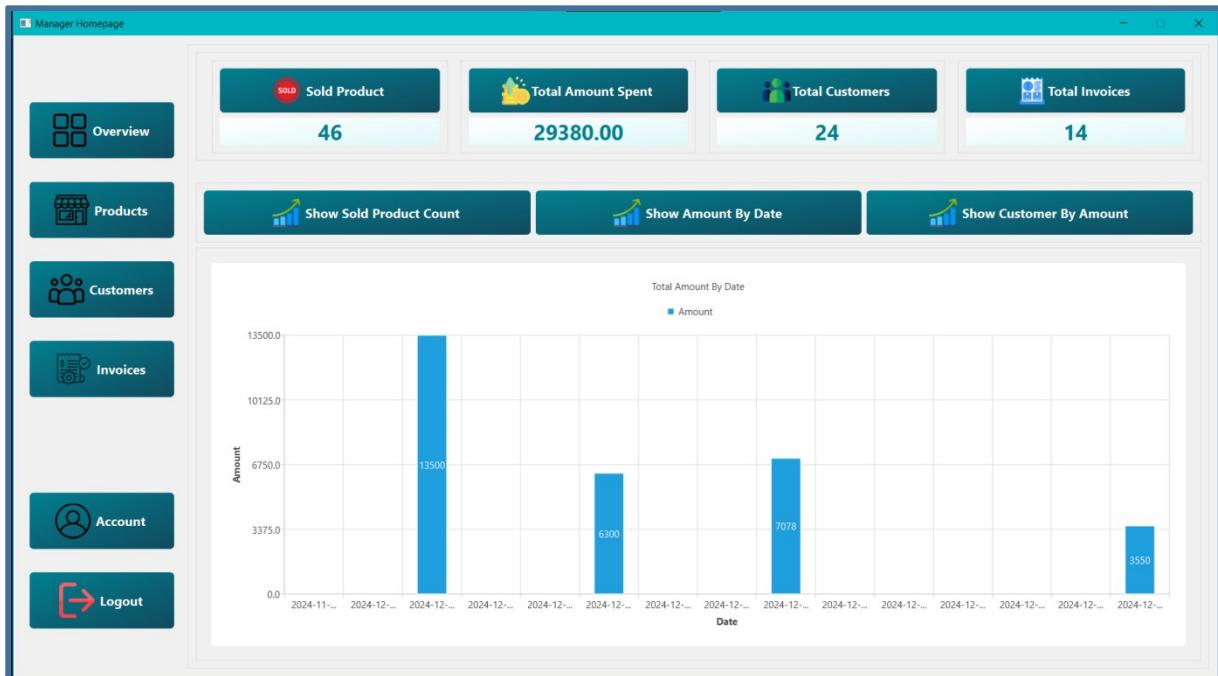
Sau khi quản lý đăng nhập vào hệ thống và chọn mục "Overview" trên thanh điều hướng, giao diện sẽ hiển thị trang tổng quan và thống kê sản phẩm bán được bao gồm **tổng số lượng sản phẩm đã bán** trên nền tảng, **tổng số tiền thu được** từ việc bán hàng, **tổng số khách hàng đã mua sản phẩm** trên nền tảng, **tổng số hóa đơn đã phát hành** và **biểu đồ cột hiển thị doanh thu theo mã sản phẩm**.



Hình 18. Giao diện tổng quan, thống kê sản phẩm bán được

3. Giao diện thống kê số tiền kiếm được theo ngày:

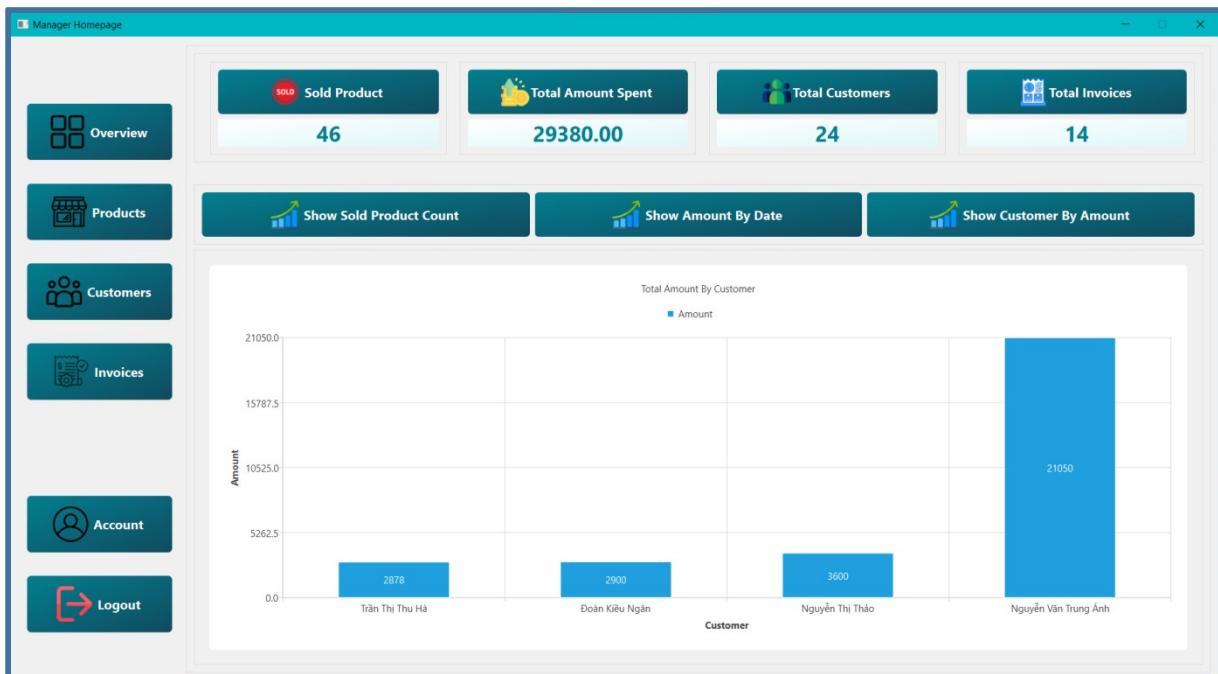
Sau khi quản lý đăng nhập vào hệ thống và chọn mục "Overview" trên thanh điều hướng, giao diện sẽ hiển thị trang tổng quan sau đó nhấn chọn "Show Amount By Date" để xuất hiện biểu đồ cột thống kê số tiền kiếm được theo ngày.



Hình 19. Giao diện thống kê số tiền kiếm được theo ngày

4. Giao diện thống kê tên khách hàng theo số tiền đã mua :

Sau khi quản lý đăng nhập vào hệ thống và chọn mục "Overview" trên thanh điều hướng, giao diện sẽ hiển thị trang tổng quan sau đó nhấn chọn "Show Customer By Amount" để xuất hiện biểu đồ cột thống kê số tiền kiếm được theo tên khách hàng.



Hình 20. Giao diện thống kê tên khách hàng theo số tiền đã mua

5. Xem danh sách các sản phẩm đã bán :

Sau khi quản lý đăng nhập vào hệ thống và chọn mục "Overview" trên thanh điều hướng, giao diện sẽ hiển thị trang tổng quan sau đó nhấp chọn "Sold Product" để xuất hiện danh sách các sản phẩm đã bán.

Sold Product

Product ID	Product Name	Quantity Sold	Price
PRD008	Điện thoại Vivo X60	750	10
PRD007	Điện thoại Oppo Find X3	700	8
PRD035	Điện thoại Samsung Galaxy A72	650	6
PRD009	Điện thoại Realme Narzo 30	350	5
PRD033	Sam Sung Z Fold 10	234	3
PRD006	Điện thoại Apple iPhone SE (2022)	500	2
PRD010	Điện thoại Realme GT	600	2
PRD038	Điện thoại Xiaomi Redmi Note 9 Pro	400	2
PRD003	Điện thoại Xiaomi Mi 11	800	2
PRD005	Điện thoại iPhone 13	950	1
PRD011	Điện thoại Sony Xperia 1 II	1400	1
PRD039	Điện thoại Huawei Y7a	350	1
PRD037	Điện thoại RedMi Note 9	678	1
PRD001	Điện thoại Samsung Galaxy S21	1000	1
PRD002	Điện thoại Samsung Galaxy Note 20	650	1

Hình 21. Xem danh sách các sản phẩm đã bán

6. Xem danh sách các sản phẩm trong cửa hàng :

Sau khi người quản lý đăng nhập vào hệ thống và chọn mục "Products" trên thanh điều hướng, giao diện sẽ hiển thị danh sách các sản phẩm có trong cửa hàng.

Search by product name...

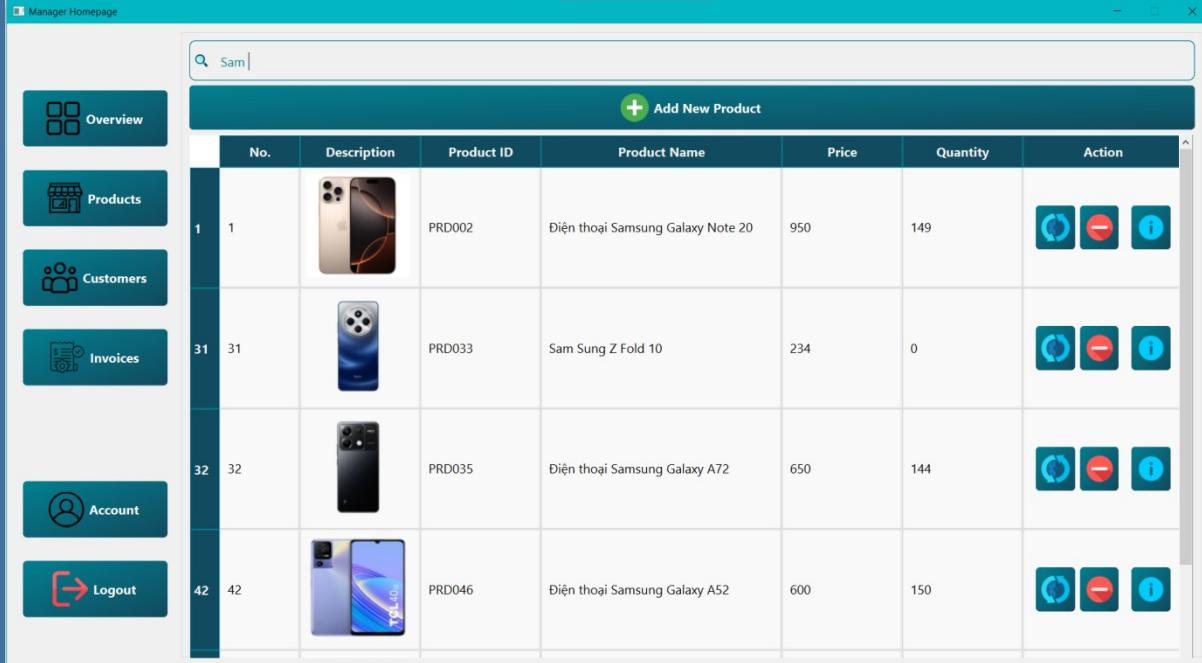
Add New Product

No.	Description	Product ID	Product Name	Price	Quantity	Action
1		PRD001	Điện thoại Samsung Galaxy S21	1000	199	
2		PRD002	Điện thoại Samsung Galaxy Note 20	950	149	
3		PRD003	Điện thoại Xiaomi Mi 11	800	298	
4		PRD004	Điện thoại iPhone 13 Pro	1200	250	

Hình 22. Xem danh sách các sản phẩm trong cửa hàng

7. Tìm kiếm sản phẩm theo tên:

Sau khi người quản lý sử dụng thanh tìm kiếm ở phía trên cùng của trang để tìm kiếm sản phẩm theo tên, giao diện sẽ hiển thị danh sách các sản phẩm liên quan đến từ khóa đã nhập.



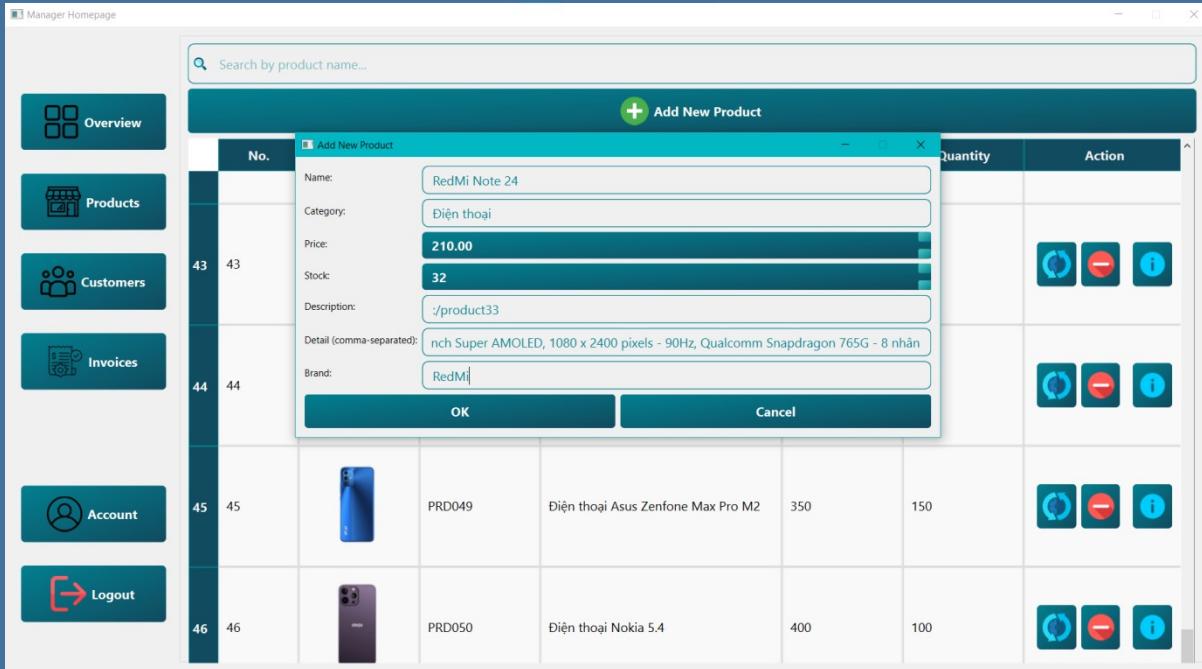
The screenshot shows a search bar at the top with the text "Sam". Below it is a table titled "Add New Product" containing four rows of product data. The columns are labeled: No., Description, Product ID, Product Name, Price, Quantity, and Action. The products listed are:

No.	Description	Product ID	Product Name	Price	Quantity	Action
1		PRD002	Điện thoại Samsung Galaxy Note 20	950	149	
31		PRD033	Sam Sung Z Fold 10	234	0	
32		PRD035	Điện thoại Samsung Galaxy A72	650	144	
42		PRD046	Điện thoại Samsung Galaxy A52	600	150	

Hình 23. Tìm kiếm sản phẩm theo tên

8. Thêm sản phẩm mới vào cửa hàng :

Sau khi người quản lý nhấp vào nút "Add New Product" trong trang quản lý sản phẩm, giao diện sẽ hiển thị một cửa sổ bật lên với tiêu đề "Add New Product" (Thêm sản phẩm mới).

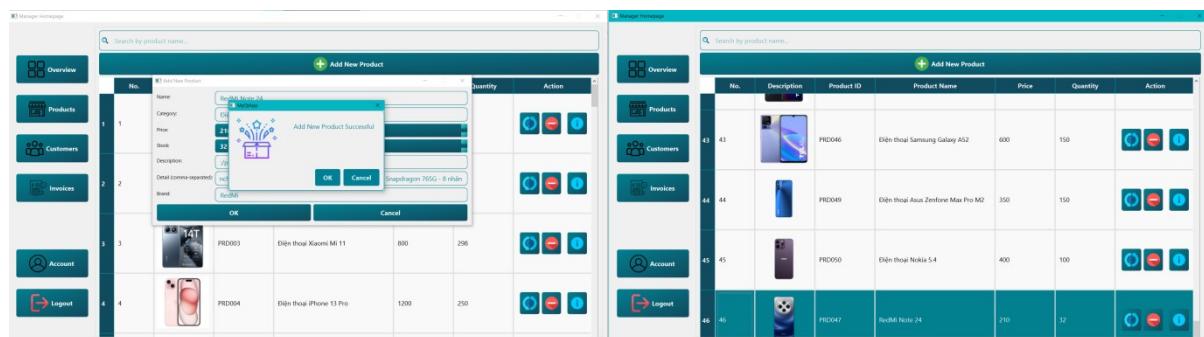


The screenshot shows a modal dialog box titled "Add New Product" overlaid on the main product list. The dialog contains fields for Name, Category, Price, Stock, and Description. The Name field is filled with "RedMi Note 24", Category is "Điện thoại", Price is "210.00", Stock is "32", and Description is "/product33". Below these fields is a "Detail (comma-separated):" section with the value "inch Super AMOLED, 1080 x 2400 pixels - 90Hz, Qualcomm Snapdragon 765G - 8 nhân". The dialog also has a "Brand:" field containing "RedMi". At the bottom are "OK" and "Cancel" buttons.

Hình 24. Thêm sản phẩm mới vào cửa hàng

9. Thêm sản phẩm thành công :

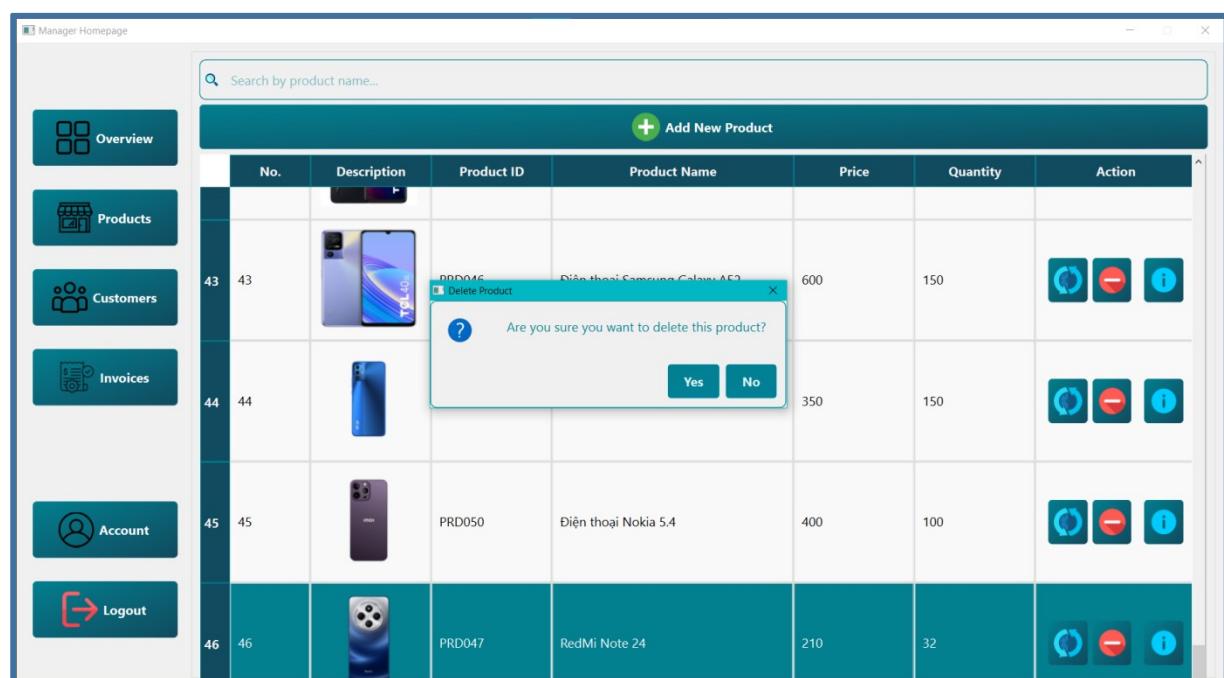
Sau khi người quản lý thêm sản phẩm mới vào cửa hàng thành công, giao diện sẽ hiển thị thông báo xác nhận sản phẩm đã được thêm thành công. Sản phẩm mới sẽ xuất hiện trong danh sách sản phẩm của cửa hàng, và ID của sản phẩm sẽ được hệ thống tạo để đảm bảo tính duy nhất để phân biệt các sản phẩm với nhau.



Hình 25. Thêm sản phẩm thành công

10. Xóa sản phẩm khỏi cửa hàng :

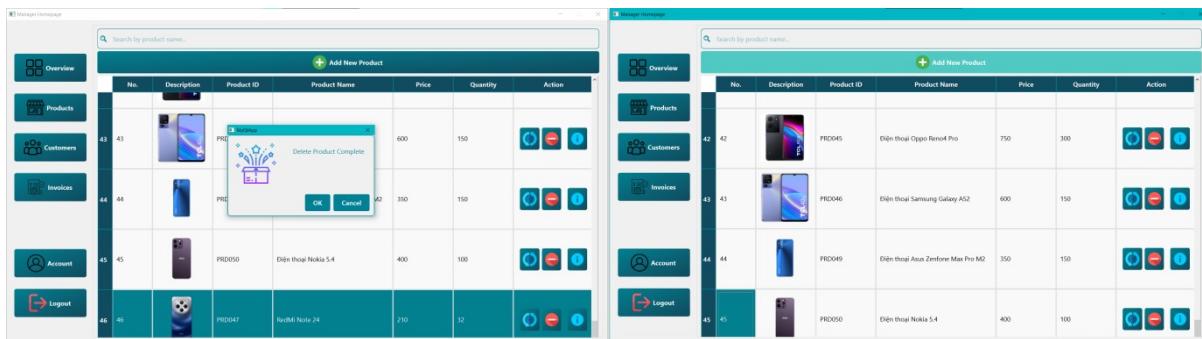
Sau khi người quản lý nhấn vào nút "Xóa" bên cạnh sản phẩm cần xóa trong danh sách sản phẩm, giao diện sẽ hiện lên một cửa sổ xác nhận với câu hỏi "Are you sure you want to delete this product?" và hai lựa chọn "Yes" và "No".



Hình 26. Xóa sản phẩm khỏi cửa hàng

11.Xóa sản phẩm thành công :

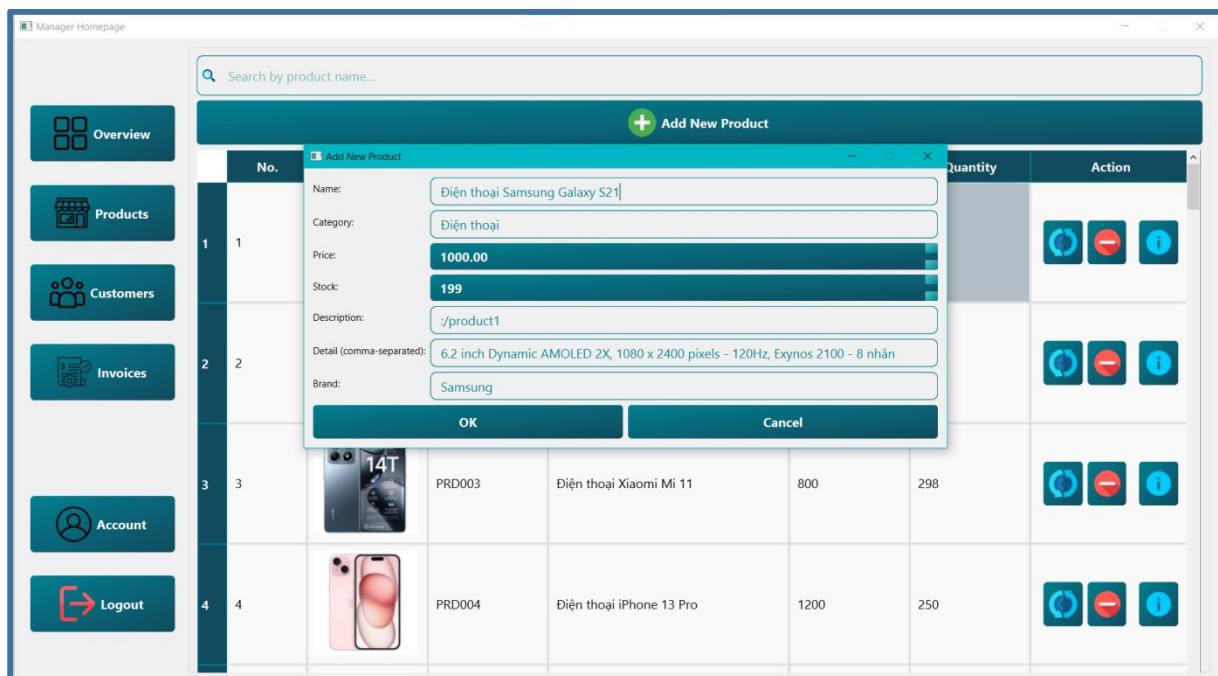
Sau khi người quản lý nhấn vào nút "Xóa" và xác nhận xóa sản phẩm, giao diện sẽ hiển thị một thông báo bật lên với nội dung "Delete Product Complete". Danh sách sản phẩm được cập nhật lại đảm bảo sản phẩm đã được xóa.



Hình 27. Xóa sản phẩm thành công

12. Chính sửa thông tin sản phẩm trong cửa hàng :

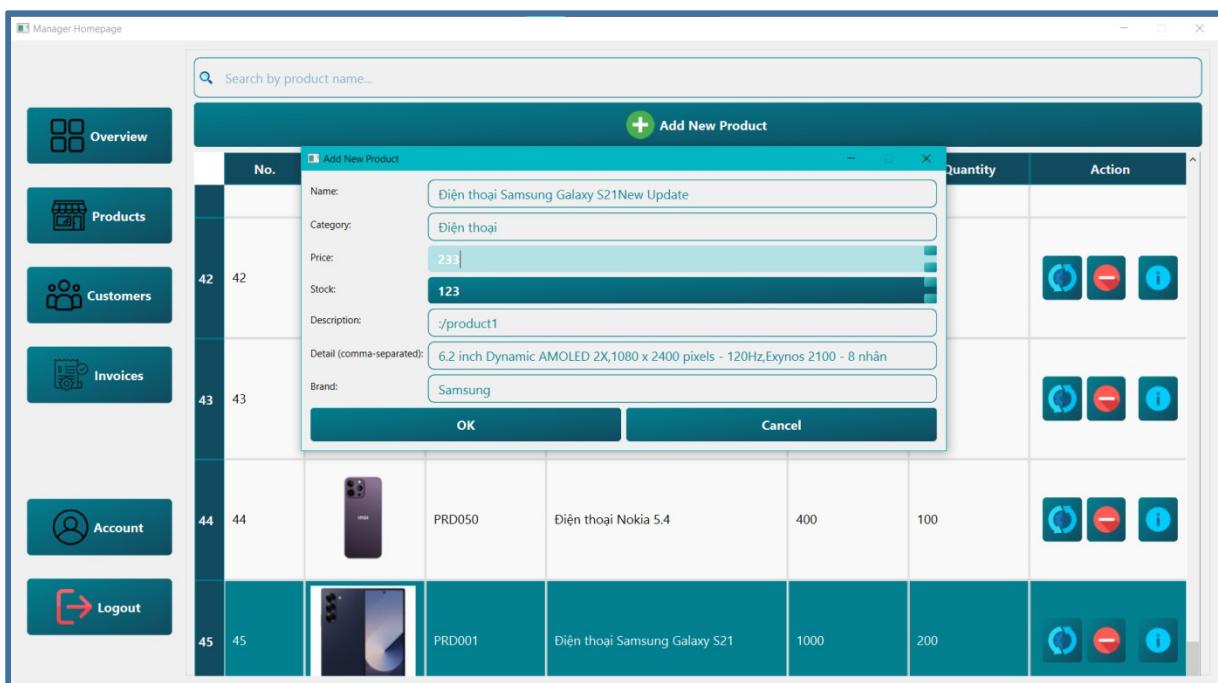
Sau khi người quản lý nhấn vào nút "Edit" (Chỉnh sửa) bên cạnh sản phẩm cần chỉnh sửa trong danh sách sản phẩm, giao diện sẽ hiển thị một cửa sổ bật lên với nội dung của sản phẩm và quản lí sẽ thay đổi trực tiếp nó.



Hình 28. Chính sửa thông tin sản phẩm trong cửa hàng

13. Thay đổi thông tin sản phẩm :

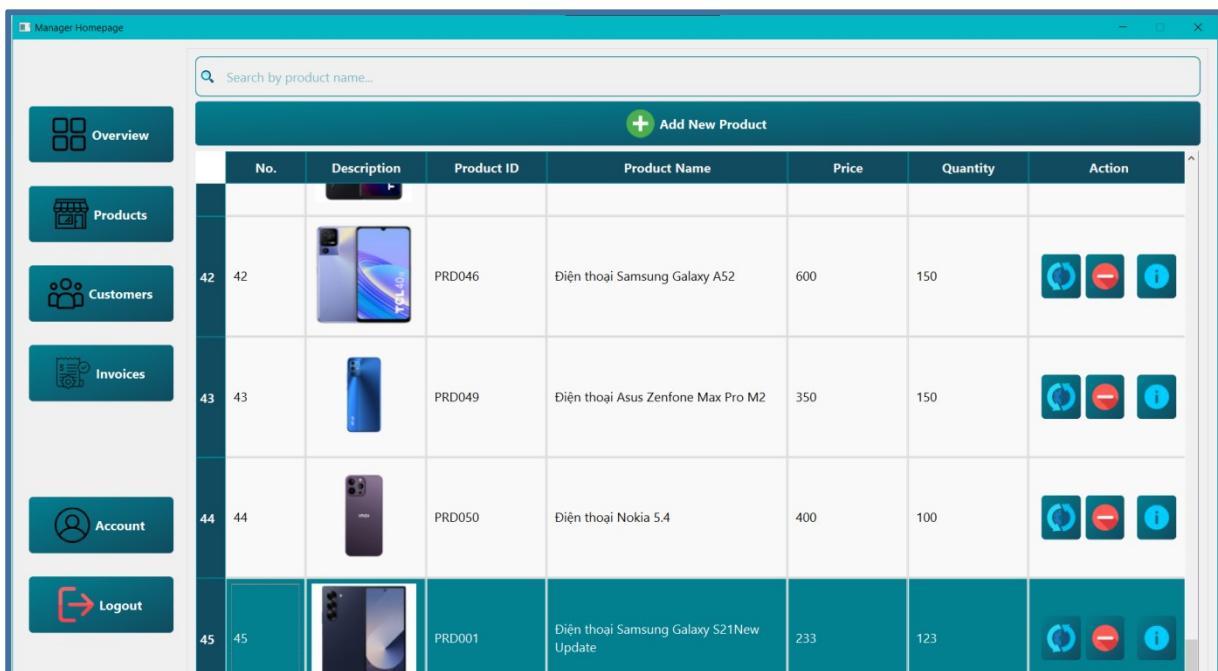
Sau khi đã thay đổi các thông số của sản phẩm , quản lí nhấn “OK” để xác nhận việc thay đổi thông tin sản phẩm.



Hình 29. Thay đổi thông tin sản phẩm

14. Thay đổi thông tin thành công :

Sau khi người quản lí nhập đầy đủ thông tin và nhấn chọn "OK" thì giao diện sẽ hiển thị thay đổi thông tin thành công. Danh sách các sản phẩm của cửa hàng sẽ được cập nhật lại và đảm bảo ID của sản phẩm và giữ nguyên để đảm bảo tính nhất quán dữ liệu.



Hình 30. Thay đổi thông tin thành công

15. Xem danh sách các khách hàng của cửa hàng :

Sau khi người quản lý đăng nhập vào hệ thống và chọn mục "Customers" trên thanh điều hướng, giao diện sẽ hiển thị danh sách các khách hàng của cửa hàng.

	CustomerID	Name	Email	Phone	Address	Total Price	Action
1	USER001	Nguyễn Văn Trung Ánh	trunganh@gmail.com	0768406235	26 Đặng Chất, Hòa Khánh Bắc, Liên Chiểu, Đà Nẵng	42100	
2	USER23	Nguyễn Thị Thảo	thao@gmail.com	0123456789	89 Liên Chiểu Đà Nẵng	7200	
3	USER22	Đoàn Kiều Ngân	ngan@gmail.com	0768406235	67 Âu Cơ Liên Chiểu Đà Nẵng	5800	
4	USER002	Trần Thị Thu Hà	thuhatr@gmail.com	0768406236	15 Lê Văn Hiến, Ngũ Hành Sơn, Đà Nẵng	5756	
5	USER003	Nguyễn Văn Hưng	hungnv@gmail.com	0768406237	12 Nguyễn Tri Phương, Thanh Khê, Đà Nẵng	0	
6	USER004	Phạm Minh Đức	minhduc.pham@gmail...	0768406238	47 Tôn Đức Thắng, Liên Chiểu, Đà Nẵng	0	
7	USER005	Vũ Thị Thu Hằng	hangvu@gmail.com	0768406239	88 Phạm Văn Đồng, Hải Châu, Đà Nẵng	0	
8	USER006	Hoàng Văn Nam	namhv@gmail.com	0768406240	22 Hoàng Diệu, Hải Châu, Đà Nẵng	0	
9	USER007	Nguyễn Minh Tùng	tungminh@gmail.com	0768406241	9 Hàm Nghi, Thanh Khê, Đà Nẵng	0	

Hình 31. Xem danh sách các khách hàng của cửa hàng

16.Tìm kiếm theo tên khách hàng :

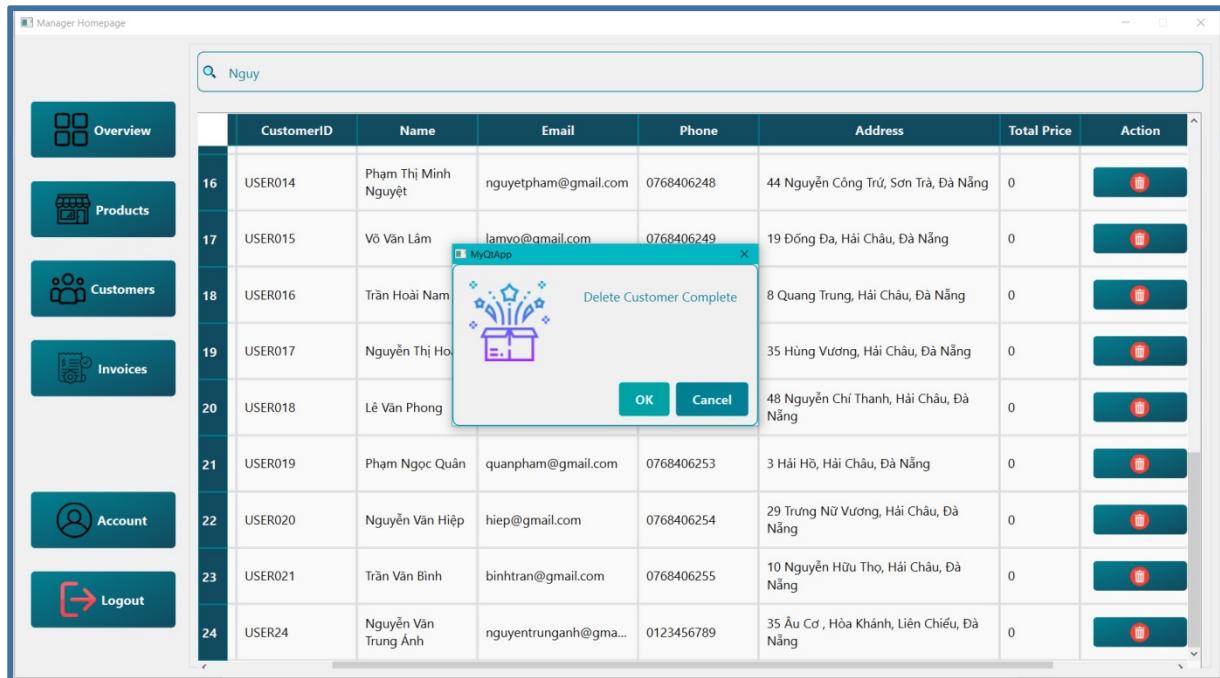
Sau khi người dùng nhập từ khóa vào thanh tìm kiếm , giao diện sẽ hiển thị kết quả tìm kiếm khách hàng tương ứng đồng thời với các kí tự nhập vào để tối ưu trải nghiệm.

	CustomerID	Name	Email	Phone	Address	Total Price	Action
16	USER014	Phạm Thị Minh Nguyệt	nguyetpham@gmail.com	0768406248	44 Nguyễn Công Trứ, Sơn Trà, Đà Nẵng	0	
19	USER017	Nguyễn Thị Hoa	hoanguyen@gmail.com	0768406251	35 Hùng Vương, Hải Châu, Đà Nẵng	0	
24	USER24	Nguyễn Văn Trung Ánh	nguyentrunganh@gmail...	0123456789	35 Âu Cơ, Hòa Khánh, Liên Chiểu, Đà Nẵng	0	

Hình 32. Tìm kiếm theo tên khách hàng

17. Xóa thông tin khách hàng của cửa hàng :

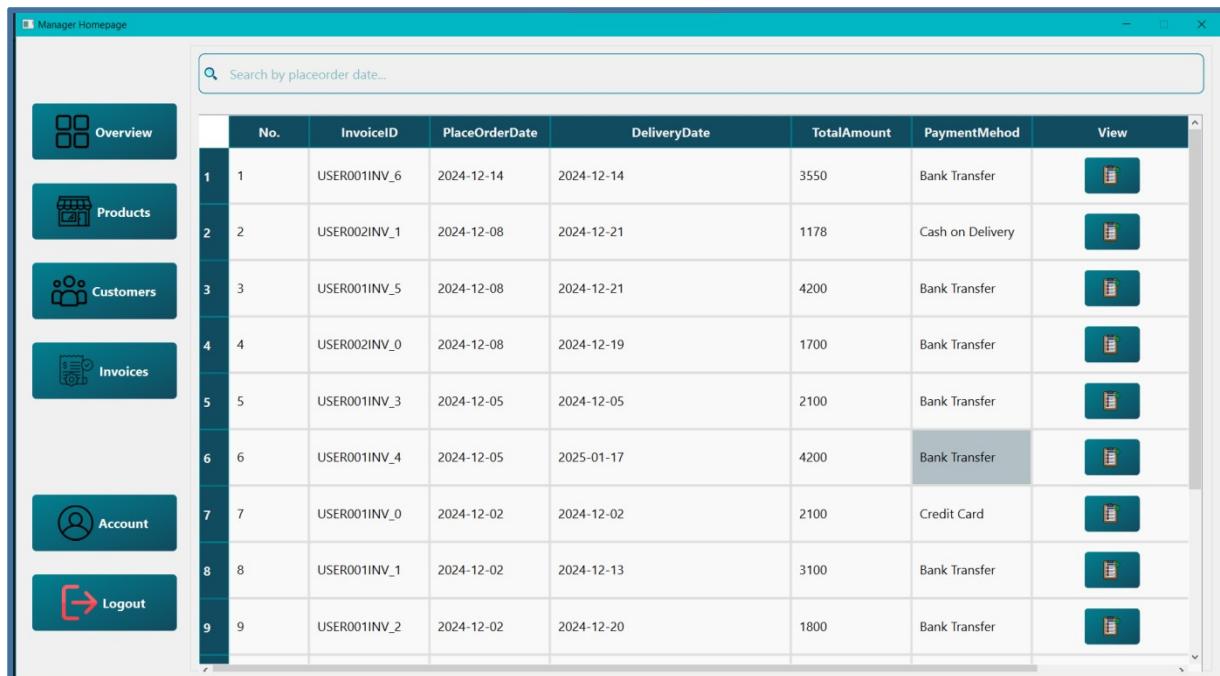
Sau khi người quản lý nhấn vào nút "Xóa" (biểu tượng thùng rác) bên cạnh thông tin khách hàng cần xóa trong danh sách khách hàng, giao diện sẽ hiện lên một cửa sổ xác nhận với câu hỏi "Are you sure you want to delete this customer?" và hai lựa chọn "Yes" và "No".



Hình 33. Xóa thông tin khách hàng của cửa hàng

18. Xem toàn bộ hóa đơn của cửa hàng :

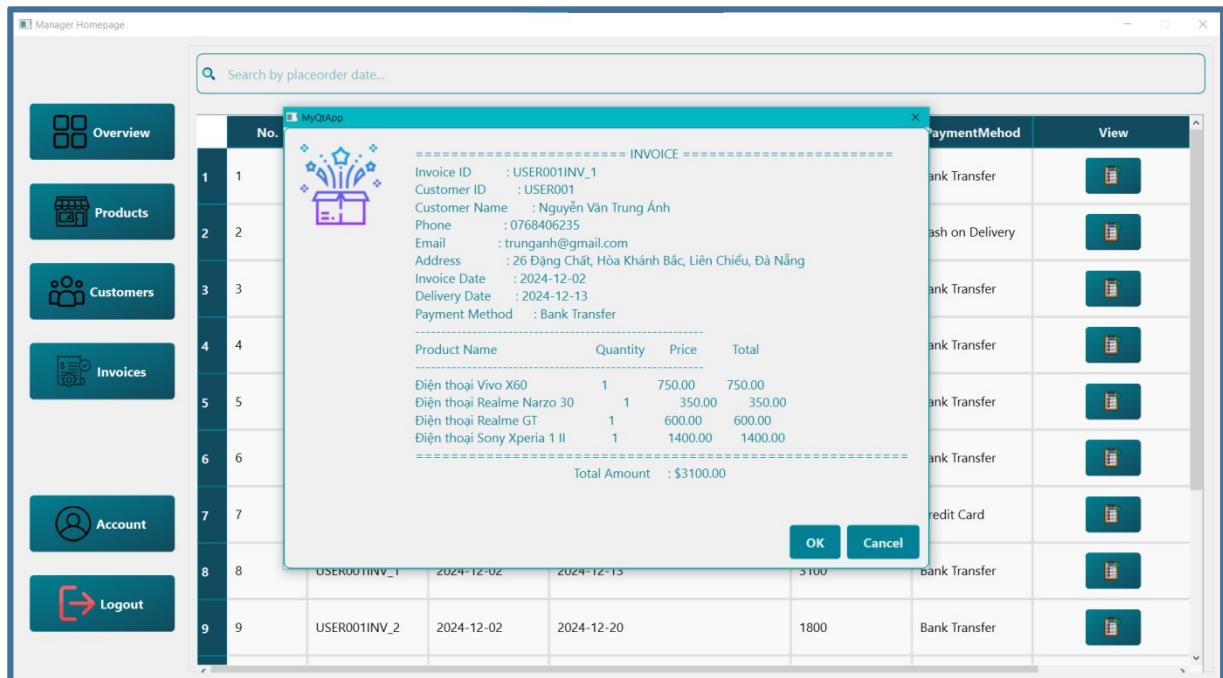
Sau khi người quản lý đăng nhập vào hệ thống và chọn mục "Invoices" trên thanh điều hướng, giao diện sẽ hiển thị danh sách toàn bộ hóa đơn của cửa hàng.



Hình 34. Xem toàn bộ hóa đơn của cửa hàng

19. Xem chi tiết hóa đơn bán hàng :

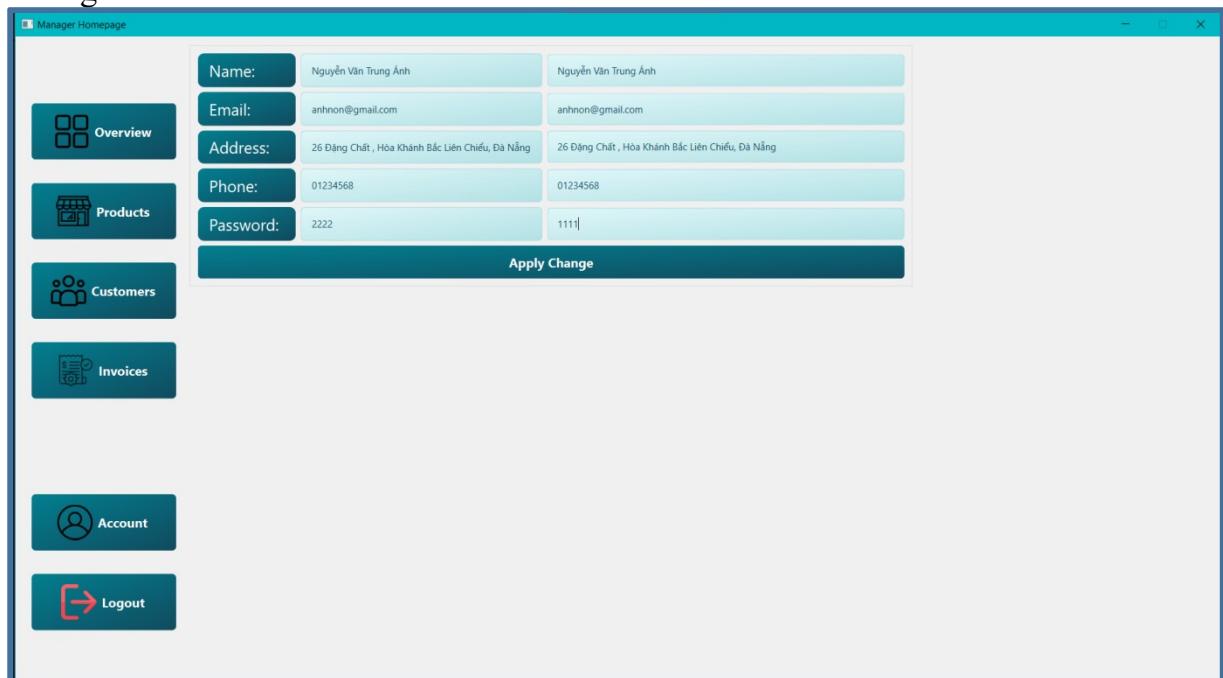
Sau khi người quản lý nhấn vào nút "View" trong danh sách hóa đơn, giao diện sẽ hiển thị chi tiết về hóa đơn bán hàng cụ thể.



Hình 35. Xem chi tiết hóa đơn bán hàng

20. Xem và chỉnh sửa thông tin tài khoản :

Sau khi người quản lí chọn mục "Account" trên thanh điều hướng, giao diện sẽ hiển thị thông tin tài khoản của họ bao gồm tên, email, địa chỉ, số điện thoại, mật khẩu. Và người quản lí cũng có thể nhấn vào nút "Change" để thay đổi thông tin tài khoản của mình như tên, email, địa chỉ, số điện thoại và mật khẩu và xác nhận bằng nút "Apply Change".



Hình 36. Xem và chỉnh sửa thông tin tài khoản

6.3 Nhận xét và đánh giá :

1.Tính thực tiễn của đề tài

Ưu điểm: Đề tài mang tính ứng dụng cao, phù hợp với xu hướng hiện nay khi mua sắm trực tuyến ngày càng trở nên phổ biến. Hệ thống không chỉ giải quyết nhu cầu thực tế của khách hàng trong việc tìm kiếm và mua sắm sản phẩm mà còn hỗ trợ người bán trong việc quản lý và mở rộng thị trường.

Nhược điểm: Đề tài tập trung vào lĩnh vực điện thoại di động, một thị trường cạnh tranh cao, đòi hỏi sự khác biệt và tiện ích vượt trội để thu hút khách hàng. Nếu không tích hợp các tính năng độc đáo hoặc mang tính đột phá, hệ thống có thể khó nổi bật giữa các đối thủ lớn.

2.Tính khả thi trong triển khai

Ưu điểm: Đề tài có phạm vi chức năng rõ ràng, từ quản lý thông tin khách hàng, sản phẩm, đến đơn hàng và doanh thu. Điều này giúp dễ dàng triển khai từng phần, đảm bảo tiến độ và chất lượng.

Nhược điểm: Đề triển khai thành công, cần một đội ngũ chuyên môn tốt, đặc biệt là trong việc tối ưu hóa trải nghiệm người dùng (UI/UX) và bảo mật hệ thống.

Hệ thống cần được kiểm thử kỹ lưỡng để đảm bảo hoạt động ổn định trong điều kiện có lượng lớn người dùng đồng thời.

3. Tính sáng tạo và độc đáo

Ưu điểm: Đề tài hướng đến việc tích hợp nhiều chức năng, từ tìm kiếm, đặt hàng, đến thống kê và quản lý, tạo nên một nền tảng toàn diện cho cả khách hàng và người bán. Nếu được phát triển thêm các tính năng như cá nhân hóa trải nghiệm hoặc hỗ trợ khách hàng bằng AI, hệ thống sẽ trở nên hấp dẫn hơn.

Nhược điểm: Mặc dù toàn diện, hệ thống có thể thiếu yếu tố mới mẻ so với các nền tảng thương mại điện tử lớn hiện nay.

4. Đóng góp cho lĩnh vực thương mại điện tử

Ưu điểm: Đề tài góp phần thúc đẩy quá trình số hóa trong lĩnh vực bán lẻ điện thoại di động. Đồng thời, hệ thống này cũng mở ra cơ hội tiếp cận thị trường online cho các doanh nghiệp nhỏ lẻ.

Nhược điểm: Trong bối cảnh cạnh tranh, để thực sự đóng góp hiệu quả, hệ thống cần có chiến lược tiếp cận thị trường cụ thể và rõ ràng.

5. Đánh giá tổng quan

Điểm mạnh:

Đáp ứng đúng nhu cầu thực tế của người dùng.

Cơ cấu chức năng hệ thống rõ ràng và hợp lý.

Tiềm năng ứng dụng cao với nhiều tính năng hữu ích.

Điểm yếu:

Tính cạnh tranh chưa cao nếu không có các điểm khác biệt nổi bật.

Cần cải thiện yếu tố sáng tạo để tăng tính hấp dẫn.

7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

7.1 Kết luận :

Hệ thống mua bán điện thoại là một giải pháp công nghệ thiết thực, góp phần đơn giản hóa quy trình mua sắm và kinh doanh trong lĩnh vực điện thoại di động. Hệ thống không chỉ mang đến trải nghiệm tiện lợi, tiết kiệm thời gian và chi phí cho khách hàng mà còn hỗ trợ người bán quản lý và phát triển hoạt động kinh doanh một cách hiệu quả, minh bạch.

Điểm mạnh của hệ thống nằm ở cơ sở dữ liệu được tổ chức rõ ràng, dễ dàng truy cập và quản lý, cùng các chức năng đáp ứng đầy đủ nhu cầu của cả khách hàng, người bán và quản lý. Bên cạnh đó, việc hỗ trợ quản lý thông tin khách hàng, sản phẩm và đơn hàng cũng như các công cụ báo cáo doanh thu và thống kê bán hàng đã tạo nên nền tảng vững chắc để tối ưu hóa mọi khía cạnh trong hoạt động thương mại điện tử.

Hệ thống không chỉ là một công cụ hỗ trợ mà còn là cầu nối quan trọng giữa khách hàng và người bán, giúp xây dựng mối quan hệ tin cậy và bền vững, đồng thời thúc đẩy ngành thương mại điện tử phát triển mạnh mẽ hơn.

7.2 Hướng phát triển :

Để nâng cao hiệu quả hoạt động và tăng cường khả năng cạnh tranh, hệ thống cần được phát triển thêm các tính năng thực tế và phù hợp với nhu cầu của thị trường:

Hỗ trợ đa nền tảng:

Xây dựng phiên bản ứng dụng dành cho thiết bị di động với giao diện thân thiện, tối ưu hóa hiệu suất trên các hệ điều hành như iOS và Android.

Tối ưu hóa giao diện trên trình duyệt để đảm bảo trải nghiệm nhất quán trên cả máy tính và điện thoại.

Tích hợp phương thức thanh toán đa dạng:

Tích hợp các cổng thanh toán trực tuyến phổ biến như thẻ tín dụng, ví điện tử (Momo, ZaloPay, PayPal), và mã QR để tăng sự tiện lợi.

Cải thiện trải nghiệm cá nhân hóa:

Sử dụng công nghệ trí tuệ nhân tạo (AI) để phân tích hành vi người dùng, từ đó đề xuất sản phẩm phù hợp với sở thích và nhu cầu của khách hàng.

Mở rộng danh mục sản phẩm:

Cho phép hệ thống hỗ trợ bán các phụ kiện điện thoại như ốp lưng, tai nghe, sạc dự phòng, hoặc các sản phẩm công nghệ liên quan để tăng cơ hội tiếp cận khách hàng.

Tăng cường bảo mật:

Cải tiến hệ thống bảo vệ dữ liệu người dùng và giao dịch bằng các công nghệ mã hóa tiên tiến.

Áp dụng các biện pháp xác thực hai yếu tố (2FA) để bảo vệ tài khoản khách hàng.

Hỗ trợ tính năng vận chuyển:

Kết nối với các đơn vị vận chuyển uy tín để tự động hóa quy trình giao nhận hàng hóa.

Cung cấp tính năng theo dõi trạng thái đơn hàng theo thời gian thực cho khách hàng.

Phân tích và báo cáo chuyên sâu:

Bổ sung các công cụ phân tích dữ liệu và báo cáo chi tiết để giúp người bán đưa ra các quyết định kinh doanh dựa trên dữ liệu.

Với những cải tiến này, hệ thống sẽ trở thành một nền tảng thương mại điện tử toàn diện và bền vững, không chỉ đáp ứng tốt nhu cầu hiện tại mà còn sẵn sàng thích nghi với sự thay đổi của thị trường trong tương lai.

8.PHỤ LỤC

Github: https://github.com/hnagnurtme/PBL_2.git

9.DANH MỤC TÀI LIỆU THAM KHẢO