

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN

-----***-----



BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM
LẬP TRÌNH JAVA

XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ QUÁN TRÀ SỮA

GVHD: ThS. Đào Mạnh Hà

Sinh viên: Nguyễn Văn Hậu

Nguyễn Lê Hoài Nam

Nguyễn Hồng Diễm

Nhóm: 17

Lớp: 20241IT6019004 **Khóa:** 18

Hà Nội - Năm 2024

MỤC LỤC

PHẦN 1: MỞ ĐẦU.....	1
1.1 Giới thiệu đề tài.....	1
1.2 Tính cấp thiết của đề tài.....	1
1.3 Lý do chọn đề tài.....	2
1.4 Công cụ thực hiện đề tài.....	2
PHẦN 2: KẾT QUẢ NGHIÊN CỨU	2
2.1 Giới thiệu	2
2.2 Khảo sát hệ thống.....	3
2.2.1 Khảo sát sơ bộ	3
2.2.2 Tài liệu đặc tả yêu cầu	4
2.3 Phân tích hệ thống.....	5
2.3.1 Mô hình hóa chức năng hệ thống.....	5
2.3.2 Mô hình hóa dữ liệu	8
2.4 Thực hiện bài toán.....	9
2.4.1 Đăng nhập (Nguyễn Văn Hậu)	9
2.4.2 Đăng ký (Nguyễn Văn Hậu).....	12
2.4.3 Đặt món (Nguyễn Lê Hoài Nam).....	15
2.4.4 Quản lý cửa hàng của quản trị viên (Nguyễn Hồng Diễm).....	18
2.4.5 Chức năng thao tác với tài khoản.....	19
2.4.6 Chức năng thêm mới sản phẩm.....	22

2.4.7 Chức năng xóa sản phẩm.....	25
2.4.8 Chức năng tạo tài khoản mới	28
2.4.9 Xem thông tin hóa đơn	30
2.4.10 Quên mật khẩu	32
2.4.11 Tính trừu tượng	35
PHẦN 3: KẾT QUẢ VÀ BÀI HỌC KINH NGHIỆM	37
3.1 Nội dung đã thực hiện	37
3.1.1 Những kiến thức và kỹ năng học được trong quá trình thực hiện đề tài.....	37
3.1.2 Chuẩn đầu ra đã đạt được trong quá trình thực hiện đề tài	38
3.1.3 Bài học kinh nghiệm sau khi kết thúc đề tài.....	38
3.2 Hướng phát triển	39
TÀI LIỆU THAM KHẢO	40

Danh mục hình ảnh

Hình 2.1: Biểu đồ use case	6
Hình 2.2: Mô hình liên kết	99
Hình 2.3: Màn hình đăng nhập	10
Hình 2.4: Màn hình đăng ký	12
Hình 2.5: Màn hình đặt món.....	15
Hình 2.6: Màn hình quản lý của Admin.....	18
Hình 2.7: Màn hình sau khi tìm kiếm tài khoản	32
Hình 2.8: Màn hình xóa tài khoản	21
Hình 2.9: Màn hình sau khi xóa tài khoản thành công	22
Hình 2.10: Màn hình thêm sản phẩm.....	Error! Bookmark not defined.
Hình 2.11: Màn hình bảng quản lí sản phẩm sau khi thêm.....	24
Hình 2.12: Màn hình xóa sản phẩm	25
Hình 2.13: Màn hình bảng quản lí sản phẩm sau khi xóa.....	27
Hình 2.14: Màn hình đăng kí tài khoản	32
Hình 2.15: Màn hình xem thông tin hóa đơn	31
Hình 2.16: Màn hình quên mật khẩu	32
Hình 2.17: Màn hình sau khi tìm lại được mật khẩu	35

PHẦN 1: MỞ ĐẦU

1.1 Giới thiệu đề tài

Trong những năm gần đây cùng với sự phát triển mạnh mẽ của nền kinh tế và sự phát triển của ngành CNTT. Tin học đang và sẽ phát triển mạnh mẽ không ngừng. Các ứng dụng của nó được áp dụng phổ biến trong các ngành như Quản lý nhân sự, Quản trị kinh doanh, Y tế, Giáo dục,... Ứng dụng tin học vào các công việc quen thuộc trong cuộc sống hàng ngày đã giúp xử lý lượng lớn các công việc không lồ một cách nhanh chóng và chính xác, đặc biệt phải kể đến là lĩnh vực quản lý. Việc sử dụng các phương pháp thủ công để quản lý các tác vụ như ghi chép bằng tay hoặc ghi chú bằng các ứng dụng thủ công thì rất khó khăn, tốn công sức trong việc tổng hợp thông tin, dẫn đến kết quả không được như mong đợi.

Với thời đại số hóa đang đi lên như bây giờ, mọi hoạt động kinh doanh đều đòi hỏi sự tiện lợi, nhanh chóng và đạt được kết quả chính xác. Vậy nên, việc xây dựng chương trình quản lý quán trà sữa là nhu cầu cấp thiết. Sự thay đổi không ngừng nghỉ của thị trường kinh doanh, từ cách thức pha chế đến mô hình phục vụ, đòi hỏi việc quản lý cần linh hoạt hơn bao giờ hết. Chính vì thế, chương trình quản lý quán trà sữa ra đời là một ví dụ cho việc tối ưu hóa toàn bộ quy trình từ quản lý nguyên liệu, nhân sự, khách hàng, hóa đơn,....

1.2 Tính cấp thiết của đề tài

Việc quản lý, bảo mật các thông tin cần thiết tại các quầy, quán có vai trò rất quan trọng trong việc tạo uy tín của quán, giúp độ uy tín được đẩy lên cao, xây dựng được niềm tin từ khách hàng.

Quản lý các công việc bằng phần mềm giúp công việc đạt tính chính xác cao, giúp việc theo công việc trở nên dễ dàng, khoa học và thuận tiện hơn nhiều.

1.3 Lý do chọn đề tài

Từ tính cấp thiết của việc điều phối các luồng hoạt động tại các cửa hàng kinh doanh, nhóm chúng em quyết định lựa chọn đề tài **“*Xây Dựng Phần Mềm Quản Lý Quán Trà Sữa*”**. Đề tài giúp chúng em áp dụng những kiến thức đã học để xây dựng chương trình hỗ trợ quản lý kinh doanh, giúp cho mọi hoạt động trở nên dễ dàng hơn, thuận tiện và nhanh chóng hơn.

1.4 Công cụ thực hiện đề tài

- Eclipse

Eclipse là một môi trường phát triển tích hợp (IDE) viết chủ yếu bằng Java và nó được dùng chủ yếu cho lập trình ứng dụng Java. Eclipse là một môi trường phát triển tích hợp dùng cho lập trình máy tính. Nó chứa một không gian làm việc cơ sở và một hệ thống plug-in để mở rộng để tùy chỉnh môi trường. Eclipse thích hợp với nhiều hệ điều hành như Window, MacOS, Linux,...

- JDK 23

JDK 23 là một bản phát hành của Java Development Kit, bao gồm môi trường phát triển và công cụ cần thiết để phát triển ứng dụng Java. Phiên bản này thường đi kèm với một số cải tiến, sửa lỗi, có các tính năng mới so với các phiên bản trước đó, cũng như ổn định hơn khi sử dụng. JDK 23 có thể được sử dụng để phát triển và triển khai các ứng dụng Java.

PHẦN 2: KẾT QUẢ NGHIÊN CỨU

2.1 Giới thiệu

Tên bài toán sẽ xây dựng: Xây dựng phần mềm quản lý quán trà sữa

Quy trình áp dụng triển khai bài tập lớn: Xây dựng phần mềm theo mô hình thu thập bản mẫu

Thực hiện quy trình 6 bước:

Bước 1: Thu thập và phân tích yêu cầu phần mềm thông qua khảo sát sơ bộ bằng hình thức phỏng vấn trực tiếp người quản lý quán. Từ kết quả khảo sát, phân tích các yêu cầu cần lấy của hệ thống bao gồm yêu cầu chức năng (yêu cầu nghiệp vụ, yêu cầu hệ thống), yêu cầu phi chức năng, miền ứng dụng.

Bước 2: Thiết kế nhanh: phân tích các chức năng thành các use case, mô tả chi tiết và thiết kế giao diện của từng use case, thiết kế dữ liệu của hệ thống.

Bước 3: Xây dựng bản nguyên mẫu: thực hiện bài toán từ những mô tả đã nêu ở trên

Bước 4: Chuyển giao kết quả cho khách hàng và nhận ý kiến đánh giá của khách hàng

Bước 5: Làm mịn nguyên mẫu: điều chỉnh, bổ sung các chức năng và tích hợp vào hệ thống nhằm hoàn thiện dần hệ thống.

Bước 6: Bàn giao cho khách hàng sản phẩm cuối cùng (Bàn giao, vận hành và bảo trì)

Hình thức sản phẩm: Sản phẩm ứng dụng

Kết quả đạt được: Cài đặt và triển khai thành công chương trình quản lý quán trà sữa bằng JAVA SWING.

2.2 Khảo sát hệ thống

2.2.1 Khảo sát sơ bộ

Mục tiêu chính của xây dựng chương trình quản lý quán trà sữa nhằm tối ưu hóa các quy trình liên quan, từ việc lưu trữ thông tin sản phẩm cho đến theo dõi quá trình làm việc của nhân viên cũng như thông tin hóa đơn.

Về phạm vi, hệ thống cần đáp ứng được các yêu cầu cơ bản như quản lý được danh mục sản phẩm của quán và ghi nhận được các giao dịch thanh toán xảy ra.

Ngoài ra, chương trình cũng cung cấp các tính năng nâng cao như việc thống kê danh sách hóa đơn. Qua đó, chương trình giúp quản lý tài nguyên hiệu quả hơn, nâng cao năng suất làm việc hơn.

2.2.2 Tài liệu đặc tả yêu cầu

2.2.2.1 Yêu cầu chức năng

- Đối với người dùng

- + Đăng ký: Cho phép người dùng tạo tài khoản mới trong hệ thống bằng cách cung cấp thông tin cá nhân cần thiết
- + Đăng nhập: Cho phép người dùng đăng nhập bằng tài khoản đã đăng ký trước đó bằng cách cung cấp email và mật khẩu để thực hiện các chức năng.
- + Đặt món: Cho phép người dùng đặt món bằng cách lựa chọn đồ ăn hoặc đồ uống được phân loại sẵn.
- + Xem thông tin hóa đơn: cho phép người dùng theo dõi quá trình xuất hóa đơn dựa theo ngày, tháng hoặc năm.

- Đối với quản trị viên

- + Đăng nhập: Cho phép người quản trị viên đăng nhập để thực hiện các chức năng tương ứng với quyền của mình
- + Quản lý sản phẩm: Lưu trữ, cập nhật hoặc xóa thông tin của món
- + Quản lý tài khoản: Quản lý các tài khoản của người dùng.
- + Xem thống kê: Quản trị viên xem thống kê lượng thu nhập.
- + Quản lý danh mục: lưu trữ và cập nhật các mục.
- + Thêm sản phẩm: Quản trị viên thêm món và giá tiền của món đó vào trong bảng menu.
- + Xem thông tin hóa đơn: Quản trị viên quản lý thông tin các hóa đơn mà người dùng đã thực hiện.

2.2.2.2 Yêu cầu phi chức năng

- Độ khả dụng: là thuộc tính chất lượng đánh giá mức độ dễ sử dụng của giao diện người dùng
- Độ tin cậy: hệ thống bảo mật thông tin tốt
- Khả năng bảo trì hệ thống: hệ thống có khả năng bảo trì, nâng cấp và cập nhật các tính năng
- Khả năng sử dụng: để chỉ định sự dễ dàng mà khách truy cập vào ứng dụng của bạn tìm thấy thông tin mà họ đang tìm kiếm.
- Khả năng tương tác: là mức độ mà các yếu tố, thành phần trên ứng dụng, người dùng dễ dàng tiếp cận và sử dụng.
- Lưu trữ cơ sở dữ liệu: File txt.
- Ngôn ngữ lập trình: Java - Swing
- Môi trường cài đặt: Windows.

2.3 Phân tích hệ thống

2.3.1 Mô hình hóa chức năng hệ thống

- User chính sử dụng chương trình và vai trò:

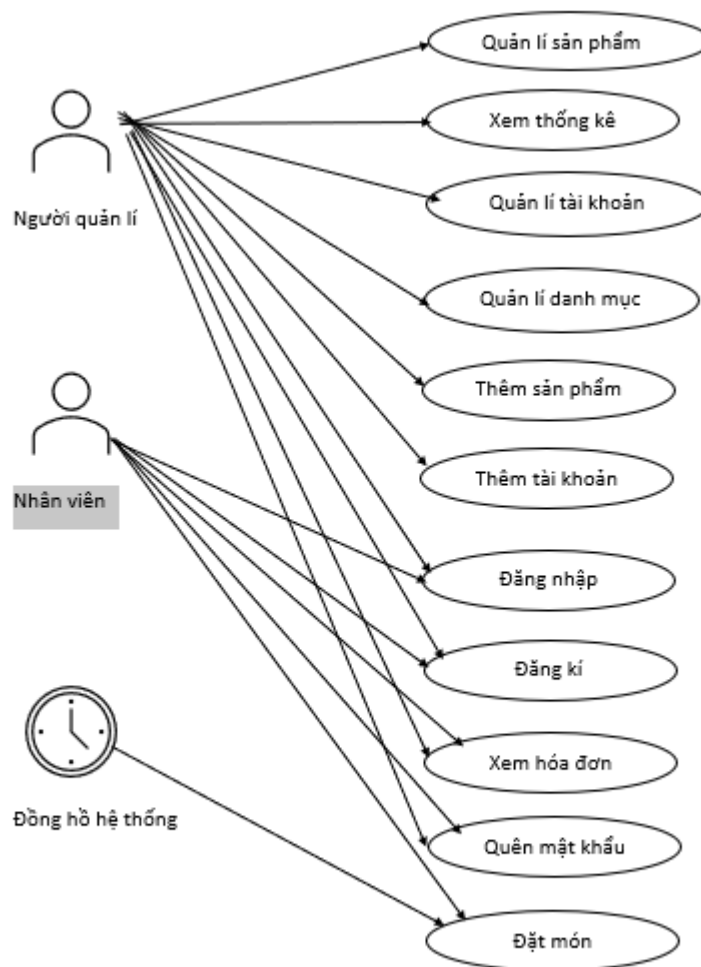
Người dùng: Đăng nhập, đặt món, quên mật khẩu, đăng xuất.

Quản trị viên: Đăng nhập, đăng kí, quên mật khẩu, quản lý sản phẩm, thêm sản phẩm, xem thông tin hóa đơn, thêm tài khoản người sử dụng.

Mô tả sơ bộ các use case:

- Đăng nhập: cho phép quản trị viên và người dùng đăng nhập tài khoản vào chương trình.
- Đăng ký: cho phép quản trị viên đăng ký tài khoản vào chương trình.
- Đặt món: cho phép người dùng gọi món theo nhu cầu.
- Xem thông tin hóa đơn: cho phép quản trị viên xem tổng hóa đơn.

- Quản lý sản phẩm: cho phép quản trị viên có quyền thêm, sửa, xóa thông tin của sản phẩm
- Tạo tài khoản: cho phép quản trị viên tạo tài khoản cần đăng nhập
- Thêm sản phẩm: cho phép quản trị viên thêm sản phẩm mới
- Quên mật khẩu: cho phép quản trị viên và người dùng tìm lại mật khẩu thông qua email và số điện thoại



Hình 2.1: Biểu đồ use case

Tên use case	Mô tả ngắn gọn	Chức năng	Ghi chú
Đăng nhập	Cho phép quản trị viên và người dùng đăng nhập vào hệ thống	Đăng nhập	Quản trị viên, Người dùng
Đăng kí	Cho phép quản trị viên đăng ký tài khoản trong hệ thống	Đăng ký	Quản trị viên
Thêm sản phẩm	Cho phép quản trị viên thêm thông tin sản phẩm mới	Thêm	Quản trị viên
Xem thông tin hóa đơn	Cho phép quản trị viên và người dùng xem danh sách hóa đơn	Xem	Quản trị viên, Người dùng
Đặt món	Cho phép người dùng đặt món	Thêm, xóa, xuất hóa đơn	Người dùng
Quản lí tài khoản	Cho phép quản trị viên quản lý tài khoản của người dùng	Cập nhật	Quản trị viên

Quản lí sản phẩm	Cho phép quản trị viên quản lý sản phẩm	Sửa	Quản trị viên
------------------	---	-----	---------------

2.3.2 Mô hình hóa dữ liệu

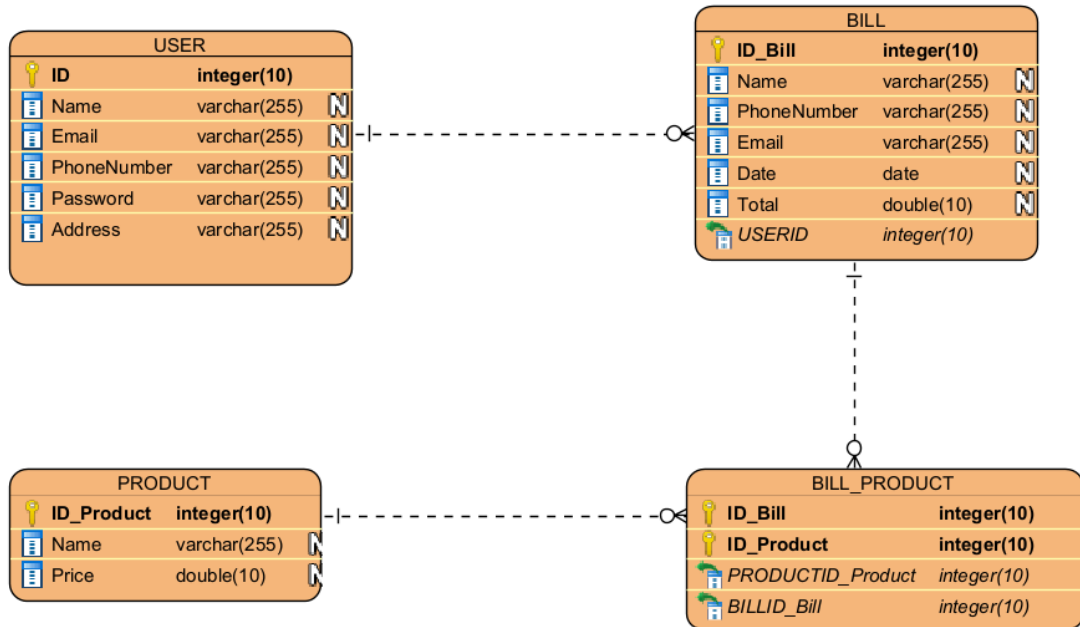
2.3.2.1 Thông tin lưu trữ

Thuộc tính của các lớp:

- Lớp User gồm có các thuộc tính:
 - + ID: thuộc tính khóa để xác định mã định danh của người dùng.
 - + Name: thuộc tính xác định tên của người dùng.
 - + Email: thuộc tính xác định email của người dùng.
 - + PhoneNumber: thuộc tính xác định số điện thoại của người dùng.
 - + Address: thuộc tính xác định địa chỉ của người dùng.
 - + Password: thuộc tính xác định mật khẩu của người dùng.
 - + Role: thuộc tính xác định vai trò của người dùng.
- Lớp Product có các thuộc tính:
 - + ID: thuộc tính khóa để xác định thứ tự của sản phẩm.
 - + Name: thuộc tính xác định tên của sản phẩm.
 - + Price: thuộc tính xác định giá của sản phẩm.
- Lớp Bill có các thuộc tính:
 - + ID: thuộc tính xác định mã định danh của hóa đơn.
 - + Name: thuộc tính xác định tên của hóa đơn.
 - + PhoneNumber: thuộc tính xác định số điện thoại của người dùng hóa đơn..
 - + Email: thuộc tính xác định email của người dùng hóa đơn.

- + Date: thuộc tính xác định thời gian tạo hóa đơn.
- + Total: thuộc tính xác định tổng số tiền của sản phẩm.

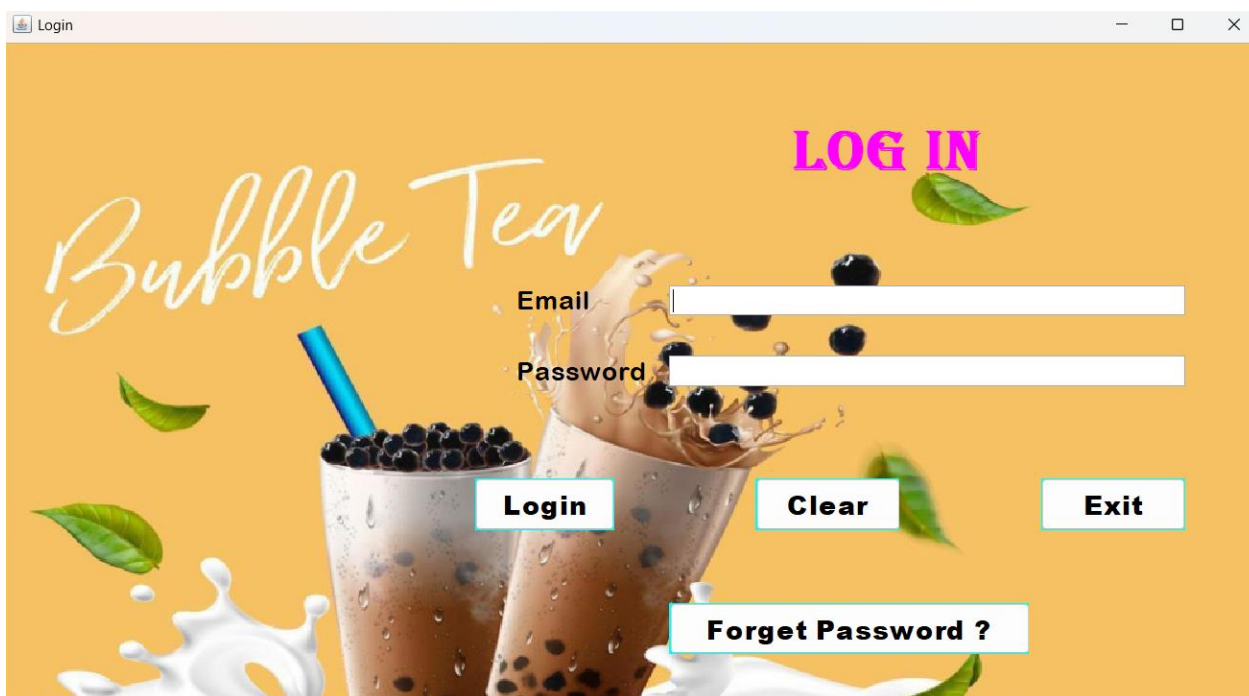
2.3.2.2 Biểu đồ lớp:



Hình 2.2: Mô hình liên kết

2.4 Thực hiện bài toán

2.4.1 Đăng nhập (Nguyễn Văn Hậu)



Hình 2.3: Màn hình đăng nhập

- Màn hình giao diện có chức năng đăng nhập bao gồm:
 - Trường nhập email, mật khẩu;
 - Nút button “Login” dùng để đăng nhập, button “Clear” để xóa các trường dữ liệu, button “Exit” để thoát chương trình
 - Button “Forgot Password” dùng để tìm lại tài khoản khi quên mật khẩu
- Hướng đối tượng:
 - Class Login được kế thừa từ javax.swing.JFrame và là thể hiện của lớp ActionListener.

```
public class LoginController implements ActionListener {
```

- Kiểm tra tài khoản có đúng không để đăng nhập

```

private void handleLogin() {
    String email = loginView.getTextFieldEmail().getText();
    String password = new String(loginView.getPasswordField().getPassword());

    User user = userRepository.findByEmailAndPassword(email, password);
    if (user != null) {
        JOptionPane.showMessageDialog(loginView, "Đăng Nhập Thành Công!", "Success", JOptionPane.INFORMATION_MESSAGE);
        if (user.getRole().equals("ADMIN")) {
            new AdminView().setVisible(true);
        } else {
            new ProductView().setVisible(true);
        }
        loginView.dispose();
    } else {
        JOptionPane.showMessageDialog(loginView, "Sai Email Hoặc Mật Khẩu!", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

- Thao tác với file

- Hệ thống thực hiện thao tác với file thông qua hàm saveUsersToFile() và loadUsersFromFile() giúp tạo luồng nhập xuất dữ liệu ra/vào file. Đối tượng nếu có sự thay đổi sẽ được lưu và đọc tại file Users.txt.

```

private void saveUsersToFile() {
    File file = new File(FILE_PATH);

    try (OutputStream fos = new FileOutputStream(file);
        ObjectOutputStream oos = new ObjectOutputStream(fos)) {
        oos.writeObject(users);
        System.out.println("Danh sách User đã được ghi vào file thành công.");
    } catch (IOException e) {
        System.err.println("Lỗi khi ghi file: " + e.getMessage());
        e.printStackTrace();
    }
}

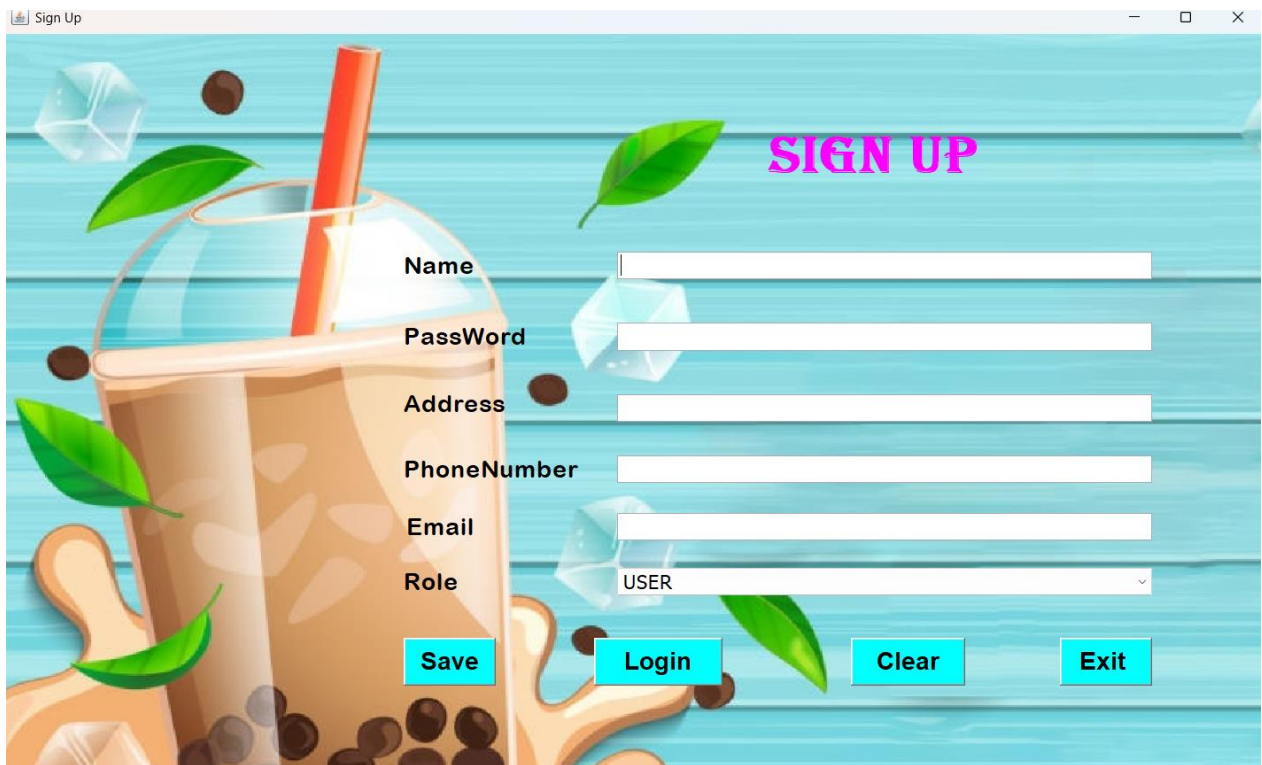
// Đọc danh sách User từ file
private void loadUsersFromFile() {
    File file = new File(FILE_PATH);
    if (!file.exists()) {
        System.out.println("File không tồn tại. Trả về danh sách rỗng.");
        return;
    }

    try (InputStream fis = new FileInputStream(file);
        ObjectInputStream ois = new ObjectInputStream(fis)) {
        Object obj = ois.readObject();

        if (obj instanceof List<?>) {
            List<?> list = (List<?>) obj;
            if (!list.isEmpty() && list.get(0) instanceof User) {
                users = new ArrayList<>();
                for (Object item : list) {
                    users.add((User) item);
                }
                System.out.println("Danh sách User đã được đọc từ file thành công.");
            } else {
                throw new IOException("Dữ liệu trong file không phải là List<User>");
            }
        } else {
            throw new IOException("Dữ liệu trong file không phải là List<?>");
        }
    } catch (IOException | ClassNotFoundException e) {
        System.err.println("Lỗi khi đọc file: " + e.getMessage());
        e.printStackTrace();
    }
}

```


2.4.2 Đăng ký (Nguyễn Văn Hậu)



Hình 2.4: Màn hình đăng ký

- Màn hình giao diện chức năng đăng ký gồm: Trường nhập tên, email, số điện thoại, địa chỉ, mật khẩu, vai trò tài khoản.
- Hướng đối tượng:
 - Encapsulation (Đóng gói): Các thuộc tính (fields) và phương thức (methods) được gói gọn trong lớp Signup.
 - Lớp Signup kế thừa từ javax.swing.JFrame, do đó Signup có thể sử dụng các phương thức và thuộc tính của JFrame.

```
public class SignupView extends JFrame {
```

- Polymorphism (Đa hình): Đoạn code sử dụng đa hình thông qua việc sử dụng ActionListener. Nhiều nút bấm (buttons) khác nhau sử dụng ActionListener để

định nghĩa các hành vi khác nhau khi được nhấn.

```
Button buttonExit = new Button("Exit ");
buttonExit.setFont(new Font("Arial Black", Font.BOLD, 22));
buttonExit.setBackground(UIManager.getColor("Button.highlight"));
buttonExit.setBounds(944, 528, 83, 43);
contentPane.add(buttonExit);

buttonExit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
});

JLabel notice = new JLabel("Sign Up");
notice.setHorizontalAlignment(SwingConstants.CENTER);
notice.setFont(new Font("Algerian", Font.BOLD, 50));
notice.setBounds(500, 97, 280, 48);
contentPane.add(notice);

SignUpController signUpController = new SignUpController(this);
buttonSave.addActionListener(signUpController);
buttonClear.addActionListener(signUpController);
}
```

- Thao tác với cơ sở dữ liệu được đọc từ file :
 - isEmpty tránh người dùng để trống dữ liệu.
 - Kiểm tra tài khoản người dùng với dữ liệu từ file User.txt thông qua userRepository nếu đã có thì thông báo đã tồn tại còn chưa thì lưu vào file.

```

private void handleSignUp() {
    try {
        String name = signUpView.getTextFieldName().getText();
        String email = signUpView.getTextFieldEmail().getText();
        String password = new String(signUpView.getPasswordField().getPassword());
        String address = signUpView.getTextFieldAddress().getText();
        int phoneNumber = Integer.parseInt(signUpView.getTextFieldPhoneNumber().getText());
        String role = signUpView.getSelectedRole();

        if (name.isEmpty() || email.isEmpty() || password.isEmpty() || address.isEmpty() || role.isEmpty()) {
            JOptionPane.showMessageDialog(signUpView, "Bạn Phải Điền Đủ Tất Cả các Ô", "Error", JOptionPane.ERROR_MESSAGE);
            return;
        }

        User existingUser = userRepository.findByEmailAndPassword(email, password);
        if (existingUser != null) {
            JOptionPane.showMessageDialog(signUpView, "Email Đã Tồn Tại", "Error", JOptionPane.ERROR_MESSAGE);
            return;
        }

        User newUser = new User(name, email, phoneNumber, address, password, role);
        userRepository.save(newUser);
        JOptionPane.showMessageDialog(signUpView, "Đăng Ký Thành Công", "Success", JOptionPane.INFORMATION_MESSAGE);
        signUpView.xoaForm();
    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(signUpView, "Số Điện Thoại Không Hợp Lệ", "Error", JOptionPane.ERROR_MESSAGE);
    } catch (Exception ex) {
        ex.printStackTrace();
        JOptionPane.showMessageDialog(signUpView, "Đã Xảy Ra Lỗi Trong quá Trình Đăng Ký: " + ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

- Tính trừu tượng:

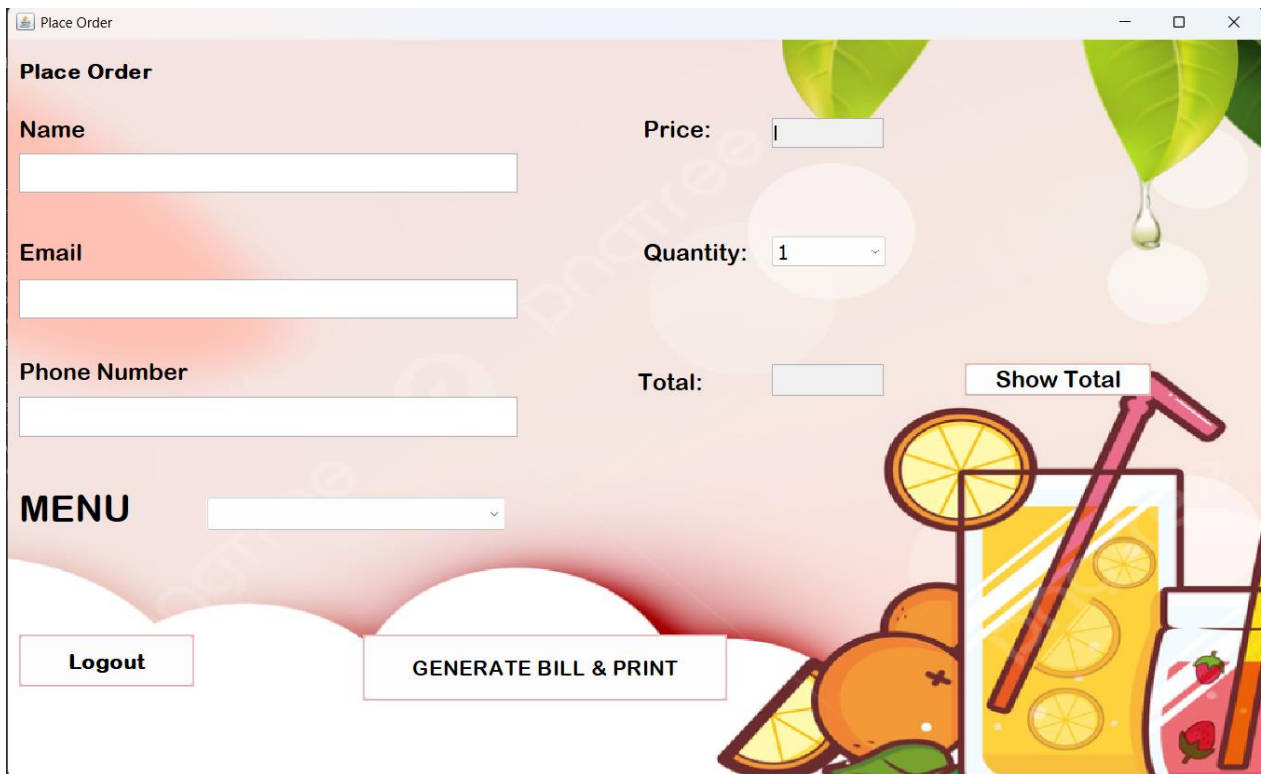
- Các phương thức được triển khai thông qua Interface IUserRepository giúp có thể ứng dụng đa kế thừa, làm cho mã nguồn dễ bảo trì, mở rộng và kiểm thử hơn.

```

public interface IUserRepository {
    public User save(User user);
    public User findByEmailAndPassword(String email, String password);
    public User findByEmail(String email);
    public User update(User user);
    public List<User> findAll();
    public void clear();
    public void delete(User user);
}

```

2.4.3 Đặt món (Nguyễn Lê Hoài Nam)



Hình 2.5: Màn hình đặt món

- Màn hình giao diện chức năng đặt món bao gồm: Trường nhập dữ liệu tên, số điện thoại, email, menu, giá, số lượng, tổng tiền, tạo hóa đơn.
- Hướng đối tượng:
 - Tính kế thừa (Inheritance): Lớp ProductView kế thừa từ javax.swing.JFrame, điều này cho phép ProductView thừa hưởng các thuộc tính và phương thức của JFrame.

```
public class ProductView extends JFrame {
```

- Tính đa hình (Polymorphism):

```

comboBoxMenu.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String selectedProduct = (String) comboBoxMenu.getSelectedItem();
        for (Product product : listProduct) {
            if (product.getTenSP().equals(selectedProduct)) {
                textFieldPrice.setText(String.valueOf(product.getGia()));
                break;
            }
        }
    }
});

btnLogout.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
        new LoginView().setVisible(true);
    }
});

```

- Tính đóng gói:

```

public String getUsername() {
    return textFieldName.getText();
}

public String getUserEmail() {
    return textFieldEmail.getText();
}

public String getUserPhone() {
    return textFieldPhone.getText();
}

public String getSelectedProduct() {
    return (String) comboBoxMenu.getSelectedItem();
}

public String getProductPrice() {
    return textFieldPrice.getText();
}

public String getSelectedQuantity() {
    return (String) comboBoxQuantity.getSelectedItem();
}

public void setTotalPrice(double total) {
    textFieldTotal.setText(String.valueOf(total));
}

```

- Không được để trống các trường dữ liệu:

```

String name = productView.getUserName();
String email = productView.getUserEmail();
String phone = productView.getUserPhone();
String selectedProduct = productView.getSelectedProduct();
String priceStr = productView.getProductPrice();
String quantityStr = productView.getSelectedQuantity();

if (name.isEmpty() || email.isEmpty() || phone.isEmpty() || selectedProduct.isEmpty() || priceStr.isEmpty() || quantityStr.isEmpty()) {
    JOptionPane.showMessageDialog(productView, "Hãy Điền Đầy Đủ Vào Chỗ Trống!");
    return;
}

```

- Sau khi hoàn thành mua hàng bill sẽ được lưu trữ vào file thông qua các hàm loadBillsFromFile() và saveBillsToFile(). Giúp cho quản trị viên có thể lấy ra thông tin Bill cần xem.

```

----- save ----- repository,

Bill newBill = new Bill(0, name, Integer.parseInt(phone), email, java.time.LocalDate.now().toString(), total);
billRepository.save(newBill);

JOptionPane.showMessageDialog(productView, "Lưu Hóa Đơn Thành Công!\nName: " + name + "\nEmail: " + email + "\nPhone: " + phone
}

```

- Sử dụng tập hợp List để lưu và đọc Bill từ file:

```

private void loadBillsFromFile() {
    try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(FILE_PATH))) {
        Object obj = ois.readObject();

        if (obj instanceof List<?>) {
            List<?> list = (List<?>) obj;
            if (!list.isEmpty() && list.get(0) instanceof Bill) {
                bills = new ArrayList<>();
                for (Object item : list) {
                    bills.add((Bill) item);
                }
                System.out.println("Danh sách hóa đơn đã được đọc từ file thành công.");
            } else {
                throw new IOException("Dữ liệu trong file không phải là List<Bill>");
            }
        } else {
            throw new IOException("Dữ liệu trong file không phải là List<?>");
        }
    } catch (FileNotFoundException e) {
        System.out.println("Chưa Có Hóa Đơn Nào");
    } catch (IOException | ClassNotFoundException e) {
        e.printStackTrace();
    }
}

private void saveBillsToFile() {
    try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(FILE_PATH))) {
        oos.writeObject(bills);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

1. Lí do sử dụng List để lưu trữ thông tin Bill:

- Thứ tự các phần tử: List duy trì thứ tự các phần tử theo thứ tự chúng được thêm vào. Điều này rất hữu ích khi bạn muốn duy trì và hiển thị danh sách các sản phẩm theo thứ tự cụ thể.
- List cho phép truy cập ngẫu nhiên (random access) đến các phần tử của nó thông qua chỉ số (index). Điều này rất hữu ích khi bạn cần truy cập hoặc cập nhật các phần tử một cách nhanh chóng.
- Dễ dàng thao tác: List cung cấp nhiều phương thức tiện lợi để thêm, xóa, và duyệt qua các phần tử, giúp cho việc thao tác với danh sách các đối tượng trở nên dễ dàng hơn.

2.4.4 Quản lý cửa hàng của quản trị viên (Nguyễn Hồng Diễn)

ID	Name	Email	Phone	Address
4	Nguyễn Văn Hậu	nguyenvanhau@gmail.com	123456789	Address2
6	Nguyễn Hồng Diễn	nguyenhongdien@gmail.com	1234567	Address3
8	Nguyễn Văn A	nguyenvana@gmail.com	987654321	Address4
10	Nguyễn Văn B	nguyenvanb@gmail.com	987654322	Address5
12	Nguyễn Văn C	nguyenvanc@gmail.com	987654323	Address6
14	Nguyễn Văn D	nguyenvand@gmail.com	987654324	Address7
16	Nguyễn Văn E	nguyenvane@gmail.com	987654325	Address8
18	Nguyễn Văn F	nguyenvanf@gmail.com	987654326	Address9
20	Nguyễn Văn G	nguyenvang@gmail.com	987654327	Address10
22	Nguyễn Văn H	nguyenvanh@gmail.com	987654328	Address11
24	Nguyễn Văn I	nguyenvani@gmail.com	987654329	Address12

ID	Name	Price
1	Moxue	45000.0
2	The Alley	50000.0
3	Trà Sữa Thiên Đường	40000.0
4	Trà Sữa Koi Thé	45000.0
5	Cà phê Trung Nguyên	50000.0
6	Cà phê Arabica	55000.0
7	Cà phê Brasil	60000.0
8	Cà phê VIP	70000.0
9	Mimi tea	35000.0
10	Heekca	40000.0
11	Share Tea	35000.0

Hình 2.6: Màn hình quản lý của admin

- Màn hình giao diện chức năng bao gồm: tên tài khoản, thêm sản phẩm, xóa sản phẩm, tạo tài khoản mới, xem thông tin bill, hiện thông tin sản phẩm.

- Hướng đối tượng:

- Tính kế thừa: class AdminView kế thừa javax.swing.JFrame

```
public class AdminView extends JFrame {
```

- Tính đóng gói: lấy thông tin qua phương thức get():

```
public String getUsernameInput() {  
    return textFieldUsername.getText();  
}
```

```
public String getNewItemName() {  
    return textFieldNewItemName.getText();  
}
```

```
public double getNewItemPrice() {  
    try {  
        return Double.parseDouble(textFieldNewItemPrice.getText());  
    } catch (NumberFormatException e) {  
        return -1;  
    }  
}
```

```
public String getDeleteItemName() {  
    return textFieldDeleteItemName.getText();  
}
```

- Tính đa hình : Đoạn code sử dụng đa hình thông qua việc sử dụng ActionListener. Nhiều nút bấm (buttons) khác nhau sử dụng ActionListener để định nghĩa các hành vi khác nhau khi được nhấn.

```
AdminController adminController = new AdminController(this, loggedInUser);  
btnGetAccountByUsername.addActionListener(adminController);  
btnDeleteAccount.addActionListener(adminController);  
btnAddNewItem.addActionListener(adminController);  
btnDeleteItem.addActionListener(adminController);  
btnGetTotalBills.addActionListener(adminController);  
btnSignUp.addActionListener(adminController);
```

2.4.5 Chức năng thao tác với tài khoản:

- Chức năng tìm kiếm theo tên thông tin tài khoản(hoặc xóa tài khoản):

```
private void getAccountByName() {
    String name = adminView.getUsernameInput();
    List<User> users = userRepository.findAll();
    User foundUser = null;
    for (User user : users) {
        if (user.getTen().equalsIgnoreCase(name)) {
            foundUser = user;
            break;
        }
    }

    if (foundUser != null) {
        adminView.updateUserList(List.of(foundUser)); // Cập nhật bảng chỉ với tài khoản tìm được
    } else {
        JOptionPane.showMessageDialog(adminView, "Không tìm thấy tài khoản", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void deleteAccount() {
    String name = adminView.getUsernameInput();
    List<User> users = userRepository.findAll();
    User foundUser = null;
    for (User user : users) {
        if (user.getTen().equalsIgnoreCase(name)) {
            foundUser = user;
            break;
        }
    }

    if (foundUser != null) {
        userRepository.delete(foundUser);
        JOptionPane.showMessageDialog(adminView, "Xóa thành công", "Success", JOptionPane.INFORMATION_MESSAGE);
        loadAccounts(); // Tải lại danh sách sau khi xóa tài khoản
    } else {
        JOptionPane.showMessageDialog(adminView, "Không tìm thấy tài khoản", "Error", JOptionPane.ERROR_MESSAGE);
    }
}
```

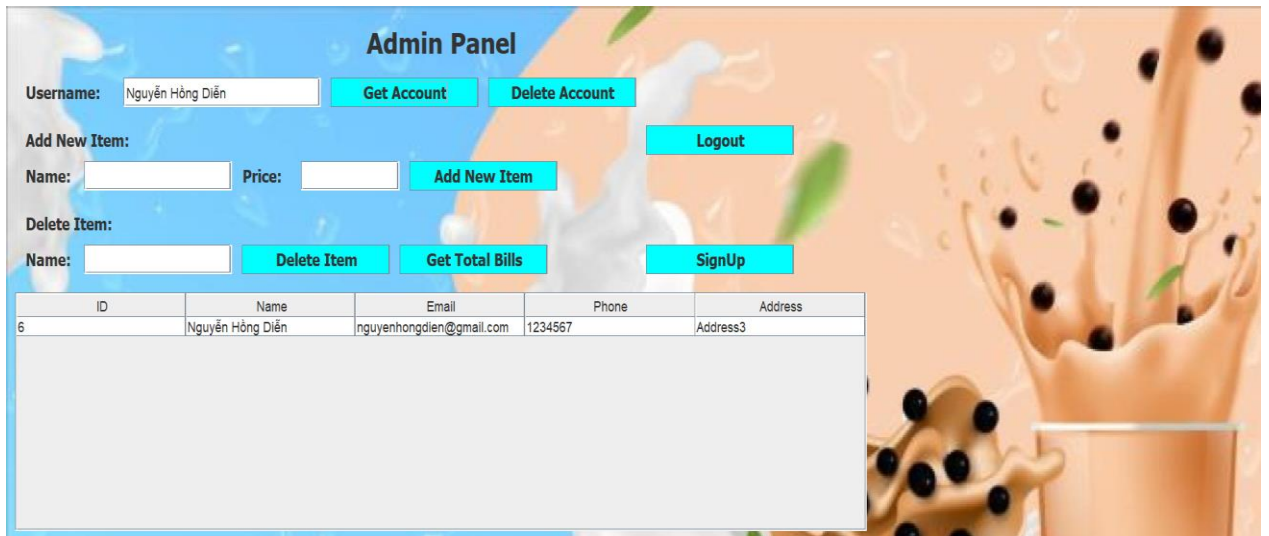
- Cập nhập thông tin tài khoản xuống bảng:

```
public void updateUserList(List<User> users) {
    userTableModel.setRowCount(0);
    for (User user : users) {
        userTableModel.addRow(new Object[]{user.getMaNV(), user.getTen(), user.getEmail(), user.getSdt(), user.getDiaChi()});
    }
}
```

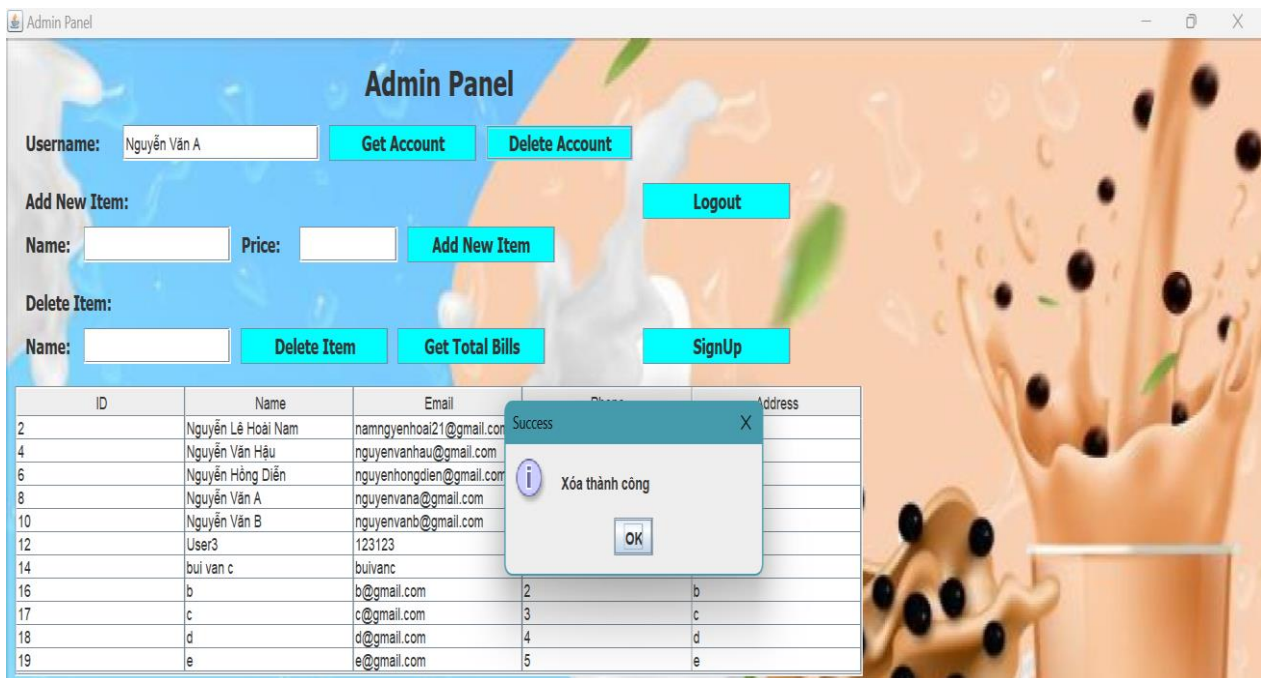
- Làm việc với file dữ liệu User.txt

- Việc thay đổi dữ liệu tài khoản sẽ được ghi lại vào file User.txt thông qua phương thức saveUsersToFile().

```
@Override
public User update(User user) {
    for (int i = 0; i < users.size(); i++) {
        if (users.get(i).getEmail().equals(user.getEmail())) {
            users.set(i, user);
            saveUsersToFile();
            return user;
        }
    }
    return null;
}
```



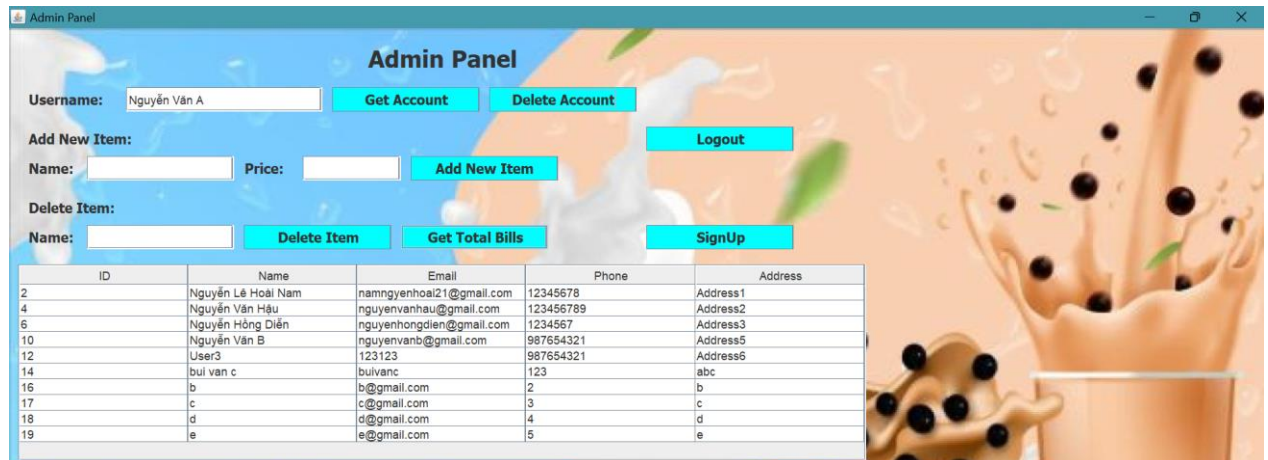
Hình 2.7: Màn hình sau khi tìm kiếm thông tin tài khoản



Hình 2.8: Màn hình khi xóa tài khoản.

- Cập nhật thông tin tài khoản xuống bảng:

```
public void updateUserList(List<User> users) {
    userTableModel.setRowCount(0);
    for (User user : users) {
        userTableModel.addRow(new Object[]{user.getMaNV(), user.getTen(), user.getEmail(), user.getSdt(), user.getDiaChi()});
    }
}
```



Hình 2.9: Màn hình sau khi xóa tài khoản thành công.

2.4.6 Chức năng thêm mới sản phẩm:

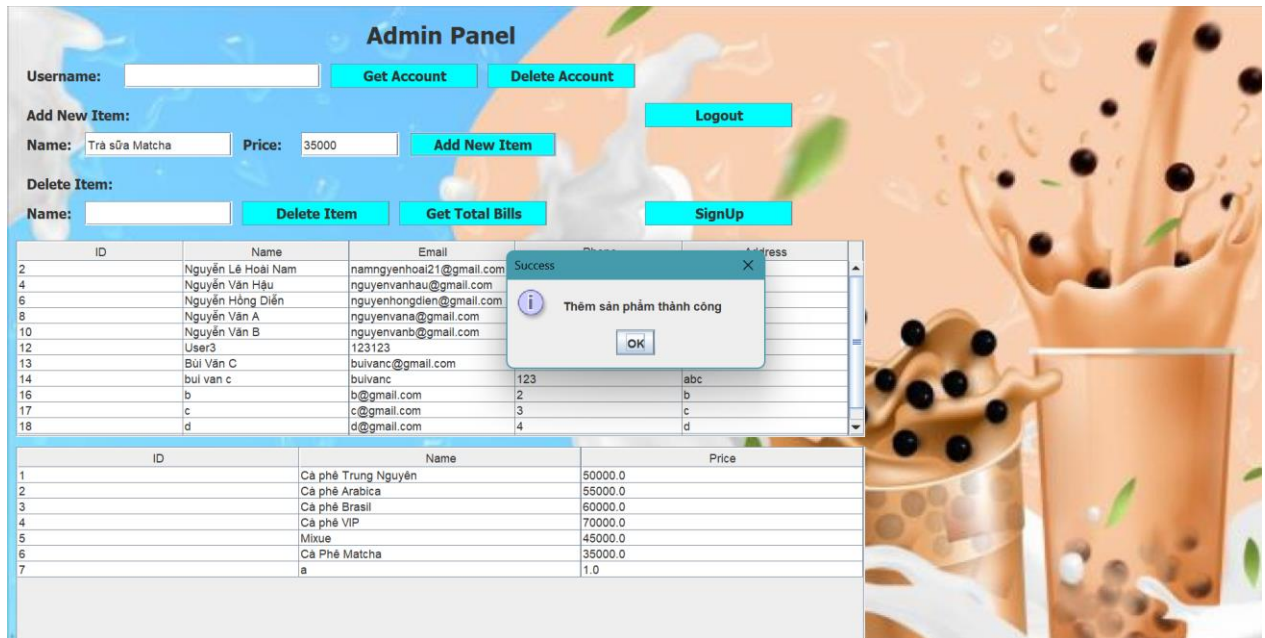
- Thêm mới sản phẩm
 - Quản trị viên nhập tên và giá sản phẩm (tên không được để trống và giá phải ≥ 0) thông tin sản phẩm sẽ được thêm vào file thông qua interface IproductRepository.
 - Lớp loadProduct giúp tải lại dữ liệu sản phẩm sau khi thêm sản phẩm mới

```
private void addNewItem() {
    String name = adminView.getNewItemName();
    double price = adminView.getNewItemPrice();
    if (name.isEmpty() || price <= 0) {
        JOptionPane.showMessageDialog(adminView, "Sản phẩm không hợp lệ", "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }
    Product newProduct = new Product(0, name, price);
    productRepository.save(newProduct);
    JOptionPane.showMessageDialog(adminView, "Thêm sản phẩm thành công", "Success", JOptionPane.INFORMATION_MESSAGE);
    adminView.clearNewItemForm();
    loadProducts(); // Tải lại danh sách sản phẩm sau khi thêm sản phẩm mới
}
```

```

@Override
public Product save(Product product) {
    int newId = generateNewId(); // Tao ID mới
    product.setMaSP(newId); // Gán ID mới cho sản phẩm
    products.add(product);
    saveProductsToFile();
    return product;
}

```



Hình 2.10: Màn hình thêm sản phẩm

- Cập nhật sản phẩm và hiện thông tin xuống bảng.

```

public void updateProductList(List<Product> products) {
    productTableModel.setRowCount(0);
    for (Product product : products) {
        productTableModel.addRow(new Object[]{product.getMaSP(), product.getTenSP(), product.getGia()});
    }
}

```

Admin Panel

Username: **Get Account** **Delete Account**

Add New Item: **Add New Item** **Logout**

Name: Price:

Delete Item: **Delete Item** **Get Total Bills** **SignUp**

ID	Name	Email	Phone	Address
2	Nguyễn Lê Hoài Nam	namnguyenhoai21@gmail.com	12345678	Address1
4	Nguyễn Văn Hậu	nguyenvanhau@gmail.com	123456789	Address2
6	Nguyễn Hồng Diên	nguyenhongdien@gmail.com	1234567	Address3
8	Nguyễn Văn A	nguyenvana@gmail.com	987654321	Address4
10	Nguyễn Văn B	nguyenvanb@gmail.com	987654321	Address5
12	User3	123123	987654321	Address6
13	Bùi Văn C	buivanc@gmail.com	123	abc
14	bui van c	buivanc@gmail.com	123	abc
16	b	b@gmail.com	2	b
17	c	c@gmail.com	3	c
18	d	d@gmail.com	4	d

ID	Name	Price
1	Cà phê Trung Nguyên	50000.0
2	Cà phê Arabica	55000.0
3	Cà phê Brasil	60000.0
4	Cà phê VIP	70000.0
5	Mixue	45000.0
6	Cà Phê Matcha	35000.0
7	a	1.0
8	Trà sữa Matcha	35000.0

Hình 2.11: Màn hình bảng quản lí sản phẩm sau khi thêm.

- Thao tác với file:
 - Hệ thống thực hiện thao tác với file thông qua hàm loadProductsFromFile() và saveProductsToFile() giúp tạo luồng nhập xuất dữ liệu ra/vào file. Đối tượng nếu có sự thay đổi sẽ được lưu và đọc tại file Product.txt

```
private void loadProductsFromFile() {
    try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(FILE_PATH))) {
        Object obj = ois.readObject();

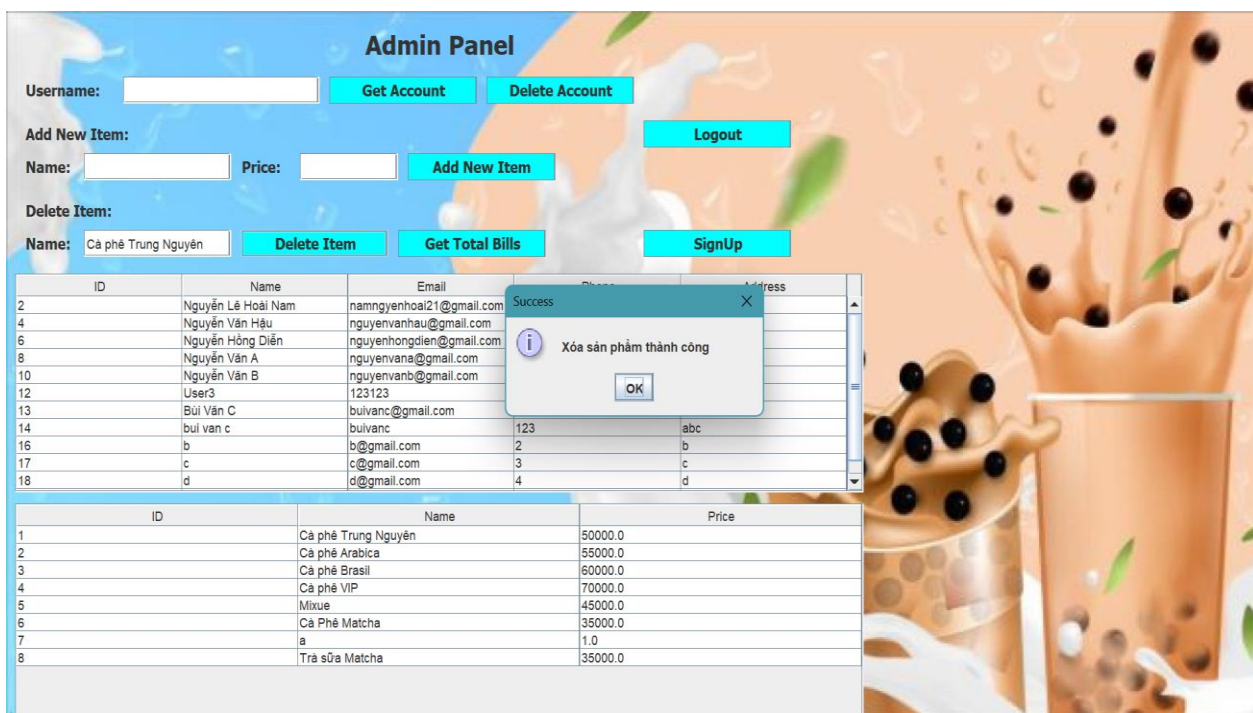
        if (obj instanceof List<?>) {
            List<?> list = (List<?>) obj;
            if (!list.isEmpty() && list.get(0) instanceof Product) {
                products = new ArrayList<>();
                for (Object item : list) {
                    products.add((Product) item);
                }
                System.out.println("Danh sách sản phẩm đã được đọc từ file thành công.");
            } else {
                throw new IOException("Dữ liệu trong file không phải là List<Product>");
            }
        } else {
            throw new IOException("Dữ liệu trong file không phải là List<?>");
        }
    } catch (FileNotFoundException e) {
        System.out.println("Chưa Có Món Đồ Uống Nào");
    } catch (IOException | ClassNotFoundException e) {
        e.printStackTrace();
    }
}

private void saveProductsToFile() {
    try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(FILE_PATH))) {
        oos.writeObject(products);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```


2.4.7 Chức năng xóa sản phẩm

- Quản trị viên nhập tên sản phẩm và nếu tồn tại sản phẩm thì xóa được sẽ, sau đó được hiển thị xuống bảng bên dưới

```
private void deleteItem() {  
    String itemName = adminView.getDeleteItemName();  
    Product product = productRepository.findByName(itemName);  
    if (product != null) {  
        productRepository.delete(product);  
        JOptionPane.showMessageDialog(adminView, "Xóa sản phẩm thành công", "Success", JOptionPane.INFORMATION_MESSAGE);  
        loadProducts(); // Tải lại danh sách sản phẩm sau khi xóa sản phẩm  
    } else {  
        JOptionPane.showMessageDialog(adminView, "Không tìm thấy sản phẩm", "Error", JOptionPane.ERROR_MESSAGE);  
    }  
}
```



Hình 2.12: Màn hình xóa sản phẩm

- Làm việc với file:
 - Thông qua các phương thức loadProductsFromFile() và saveBProductToFile() thì các hóa đơn được lưu trữ dưới dạng file txt. Các phương thức trên có sử dụng try_catch (bắt ngoại lệ) giúp đảm bảo chương trình diễn ra suôn sẻ.

```
private void loadProductsFromFile() {
    try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(FILE_PATH))) {
        Object obj = ois.readObject();

        if (obj instanceof List<?>) {
            List<?> list = (List<?>) obj;
            if (!list.isEmpty() && list.get(0) instanceof Product) {
                products = new ArrayList<>();
                for (Object item : list) {
                    products.add((Product) item);
                }
                System.out.println("Danh sách sản phẩm đã được đọc từ file thành công.");
            } else {
                throw new IOException("Dữ liệu trong file không phải là List<Product>");
            }
        } else {
            throw new IOException("Dữ liệu trong file không phải là List<?>");
        }
    } catch (FileNotFoundException e) {
        System.out.println("Chưa Có Món Đồ Uống Nào");
    } catch (IOException | ClassNotFoundException e) {
        e.printStackTrace();
    }
}

private void saveProductsToFile() {
    try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(FILE_PATH))) {
        oos.writeObject(products);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

- Cập nhập sản phẩm và hiện thông tin xuống bảng.

```
public void updateProductList(List<Product> products) {
    productTableModel.setRowCount(0);
    for (Product product : products) {
        productTableModel.addRow(new Object[]{product.getMaSP(), product.getTenSP(), product.getGia()});
    }
}
```

Admin Panel

Username:

Add New Item: Name: Price:

Delete Item: Name:

ID	Name	Email	Phone	Address
2	Nguyễn Lê Hoài Nam	namnguyenhoai21@gmail.com	12345678	Address1
4	Nguyễn Văn Hậu	nguyenvanhau@gmail.com	123456789	Address2
6	Nguyễn Hồng Diên	nguyenhongdien@gmail.com	1234567	Address3
8	Nguyễn Văn A	nguyenvana@gmail.com	987654321	Address4
10	Nguyễn Văn B	nguyenvanb@gmail.com	987654321	Address5
12	User3	123123	987654321	Address6
13	Bùi Văn C	buivanc@gmail.com	123	abc
14	bui van c	buivanc	123	abc
16	b	b@gmail.com	2	b
17	c	c@gmail.com	3	c
18	d	d@gmail.com	4	d

ID	Name	Price
2	Cà phê Arabica	55000.0
3	Cà phê Brasil	60000.0
4	Cà phê VIP	70000.0
5	Mixue	45000.0
6	Cà Phê Matcha	35000.0
7	a	1.0
8	Trà sữa Matcha	35000.0

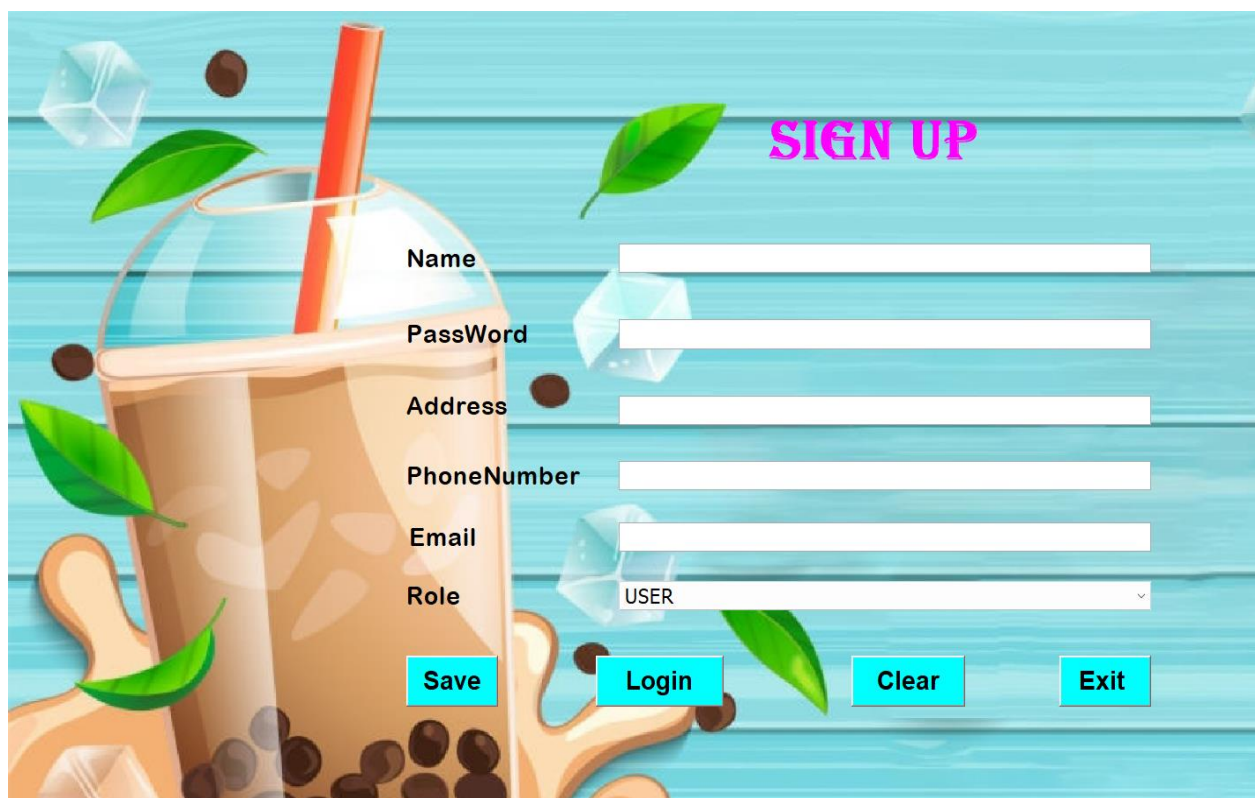
Hình 2.13: Màn hình bảng quản lý sản phẩm sau khi xóa.

- Tính đóng gói: Lấy thông tin qua phương thức get(), giúp bảo vệ dữ liệu bên trong khỏi sự truy cập và thay đổi trực tiếp từ bên ngoài.

```
public String getDeleteItemName() {
    return textFieldDeleteItemName.getText();
}
```


2.4.8 Chức năng tạo mới tài khoản:

- Tạo tài khoản:
 - Quản trị viên có quyền tạo mới tài khoản với vai trò là admin hoặc user. Điều này giúp cho quản trị viên có thể khởi tạo, quản lý được các thông tin tài khoản. Tăng tính bảo mật, an toàn với các thông tin của người sử dụng.



The image shows a 'SIGN UP' form overlaid on a background illustration of a bubble tea drink. The form includes the following fields and controls:

- NAME**: Text input field.
- PassWord**: Text input field.
- Address**: Text input field.
- PhoneNumber**: Text input field.
- Email**: Text input field.
- Role**: Dropdown menu with 'USER' selected.
- Buttons**: 'Save', 'Login', 'Clear', and 'Exit' buttons.

Hình 2.14: Màn hình đăng kí tài khoản

- Tính đóng gói giúp bảo vệ an toàn dữ liệu người dùng:

```
public JTextField getTextFieldName() {  
    return textField_name;  
}  
  
public JTextField getTextFieldEmail() {  
    return textField_email;  
}  
  
public JTextField getTextFieldAddress() {  
    return textField_address;  
}  
  
public JTextField getTextFieldPhoneNumber() {  
    return textField_phonenumber;  
}  
  
public JPasswordField getPasswordField() {  
    return passwordField;  
}  
  
public String getSelectedRole() {  
    return (String) comboBoxRole.getSelectedItem();  
}
```

- Thao tác với file:
 - Hệ thống thực hiện thao tác với file thông qua hàm saveUsersToFile() và loadUsersFromFile() giúp tạo luồng nhập xuất dữ liệu ra/vào file. Đối tượng nếu có sự thay đổi sẽ được lưu và đọc tại file User.txt.

```

// Ghi danh sách User vào file
private void saveUsersToFile() {
    File file = new File(FILE_PATH);

    try (OutputStream fos = new FileOutputStream(file);
        ObjectOutputStream oos = new ObjectOutputStream(fos)) {
        oos.writeObject(users);
        System.out.println("Danh sách User đã được ghi vào file thành công.");
    } catch (IOException e) {
        System.err.println("Lỗi khi ghi file: " + e.getMessage());
        e.printStackTrace();
    }
}

// Đọc danh sách User từ file
private void loadUsersFromFile() {
    File file = new File(FILE_PATH);
    if (!file.exists()) {
        System.out.println("File không tồn tại. Trả về danh sách rỗng.");
        return;
    }

    try (InputStream fis = new FileInputStream(file);
        ObjectInputStream ois = new ObjectInputStream(fis)) {
        Object obj = ois.readObject();

        if (obj instanceof List<?>) {
            List<?> list = (List<?>) obj;
            if (!list.isEmpty() && list.get(0) instanceof User) {
                users = new ArrayList<>();
                for (Object item : list) {
                    users.add((User) item);
                }
                System.out.println("Danh sách User đã được đọc từ file thành công.");
            } else {
                throw new IOException("Dữ liệu trong file không phải là List<User>");
            }
        }
    }
}

```

2.4.9 Xem thông tin hóa đơn

- Màn hình giao diện gồm các dữ liệu bao gồm: Mã hóa đơn, tên khách hàng, gmail, số điện thoại, ngày in hóa đơn, tổng tiền.

Bill List					
ID	Name	Email	Phone	Date	Total
2	Nguyễn Văn A	a.nguyen@gmail.com	123456789	2024-12-01	200000.0
4	Trần Thị B	b.tran@gmail.com	987654321	2024-12-02	150000.0
6	Lê Văn C	c.le@gmail.com	456789123	2024-12-03	300000.0
8	Phạm Thị D	d.pham@gmail.com	789123456	2024-12-04	250000.0
10	Hoàng Văn E	e.hoang@gmail.com	321654987	2024-12-05	100000.0
12	Hoàng Văn D	d.hoang@gmail.com	321654986	2024-12-05	120000.0
14	Hoàng Văn F	f.hoang@gmail.com	321654985	2024-12-05	400000.0
16	Hoàng Văn G	g.hoang@gmail.com	321654984	2024-12-06	300000.0
18	Hoàng Văn H	h.hoang@gmail.com	321654983	2024-12-07	2000000.0
20	Hoàng Văn I	i.hoang@gmail.com	321654982	2024-12-08	45000.0

Hình 2.15: Màn hình xem thông tin hóa đơn

- Hướng đối tượng:
 - Tính kế thừa: lớp BillView() kế thừa từ lớp JFrame().

```
public class BillView extends JFrame {
```

- Tính đóng gói: Lấy thông tin qua phương thức get()

```
if (bills != null) {
    for (Bill bill : bills) {
        tableModel.addRow(new Object[]{bill.getMaHD(),
            bill.getTen(),
            bill.getEmail(),
            bill.getSDT(),
            bill.getNgayDat(),
            bill.getTongTien()});
    }
}
```

- Làm việc với file:
 - Thông qua các phương thức loadBillsFromFile() và saveBillsToFile() thì các hóa đơn được lưu trữ dưới dạng file txt. Các phương thức trên có sử dụng try_catch (bắt ngoại lệ) giúp đảm bảo chương trình diễn ra suôn sẻ.

```

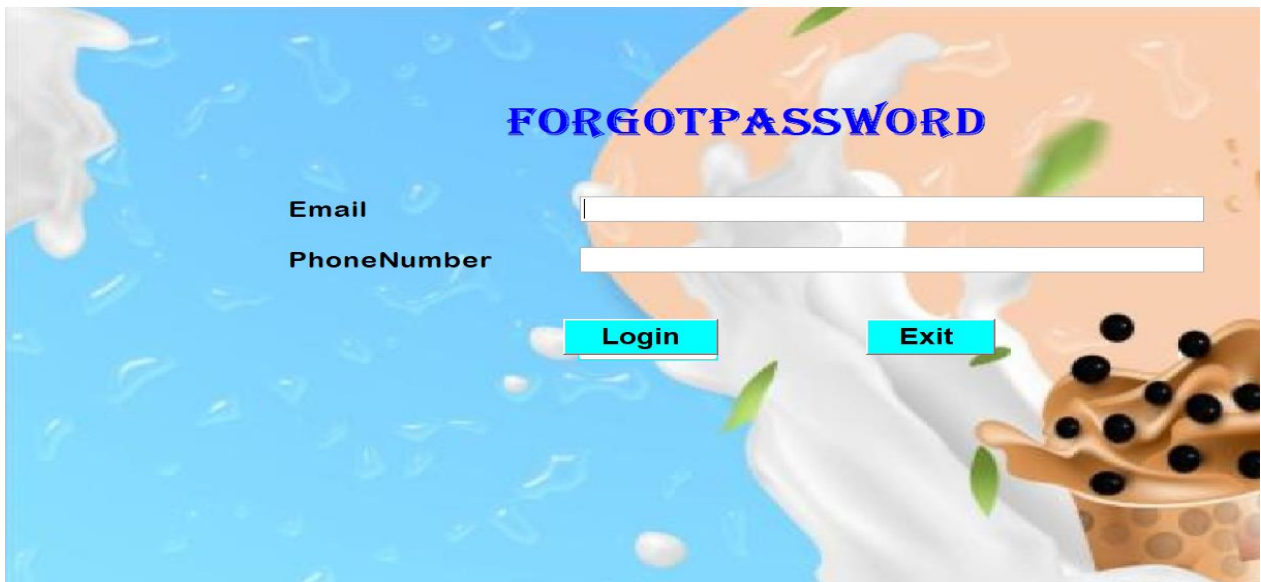
private void loadBillsFromFile() {
    try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(FILE_PATH)))
        Object obj = ois.readObject();

        if (obj instanceof List<?>) {
            List<?> list = (List<?>) obj;
            if (!list.isEmpty() && list.get(0) instanceof Bill) {
                bills = new ArrayList<>();
                for (Object item : list) {
                    bills.add((Bill) item);
                }
                System.out.println("Danh sách hóa đơn đã được đọc từ file thành công.");
            } else {
                throw new IOException("Dữ liệu trong file không phải là List<Bill>");
            }
        } else {
            throw new IOException("Dữ liệu trong file không phải là List<?>");
        }
    } catch (FileNotFoundException e) {
        System.out.println("Chưa Có Hóa Đơn Nào");
    } catch (IOException | ClassNotFoundException e) {
        e.printStackTrace();
    }
}

private void saveBillsToFile() {
    try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(FILE_PATH)))
        oos.writeObject(bills);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

2.4.10 Quên mật khẩu (Nguyễn Lê Hoài Nam)



Hình 2.16: Màn hình quên mật khẩu

- Màn hình giao diện chức năng xác thực người dùng bao gồm: trường nhập liệu Email, trường nhập số điện thoại, các button.
- Hướng đối tượng

- Tính kế thừa: Lớp ForgotPasswordView() kế thừa từ lớp JFrame

```
public class ForgotPasswordView extends JFrame {
```

- Tính đóng gói: Lấy thông tin qua phương thức get()

```
public JTextField getTextFieldEmail() {
    return textField;
}
```

```
public JTextField getTextFieldPhoneNumber() {
    return textField_1;
}
```

- Tính đa hình: Nút bấm và trường nhập dữ liệu khác nhau sử dụng ActionListener để định nghĩa các hành vi khác nhau khi thao tác

```
Button button_login = new Button("Login");
button_login.addActionListener(new ActionListener() {
    Button button_exit = new Button("Exit ");
    button_exit.addActionListener(new ActionListener())
```

- Lấy dữ liệu email và số điện thoại được người sử dụng nhập để so sánh với thông tin tài khoản có trong file User.txt nếu đúng thì sẽ trả về mật khẩu thông qua phương thức getmatKhau(), còn nếu sai thì hiện ra thông báo cho người dùng.

```

private void handleFindUser() {
    String email = forgotPasswordView.getTextFieldEmail().getText();
    String phoneNumberStr = forgotPasswordView.getTextFieldPhoneNumber().getText();
    boolean userFound = false;

    try {
        int phoneNumber = Integer.parseInt(phoneNumberStr);
        List<User> users = userRepository.findAll();

        for (User user : users) {
            if (user.getEmail().equals(email) && user.getSdt() == phoneNumber) {
                userFound = true;
                String message = "Tìm Thấy Người Dùng: " + user.getTen() + "\nPassword: " + user.getMatKhau();
                JOptionPane.showMessageDialog(forgotPasswordView, message, "User Found", JOptionPane.INFORMATION_MESSAGE);
                break;
            }
        }

        if (!userFound) {
            JOptionPane.showMessageDialog(forgotPasswordView, "Email hoặc PhoneNumber Không đúng.", "Error", JOptionPane.ERROR_MESSAGE);
        }
    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(forgotPasswordView, "Nhập sai PhoneNumber.", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

- Thao tác với file:
 - Thông qua phương thức loadUsersFromFile() để có thể lấy dữ liệu tài khoản.

```

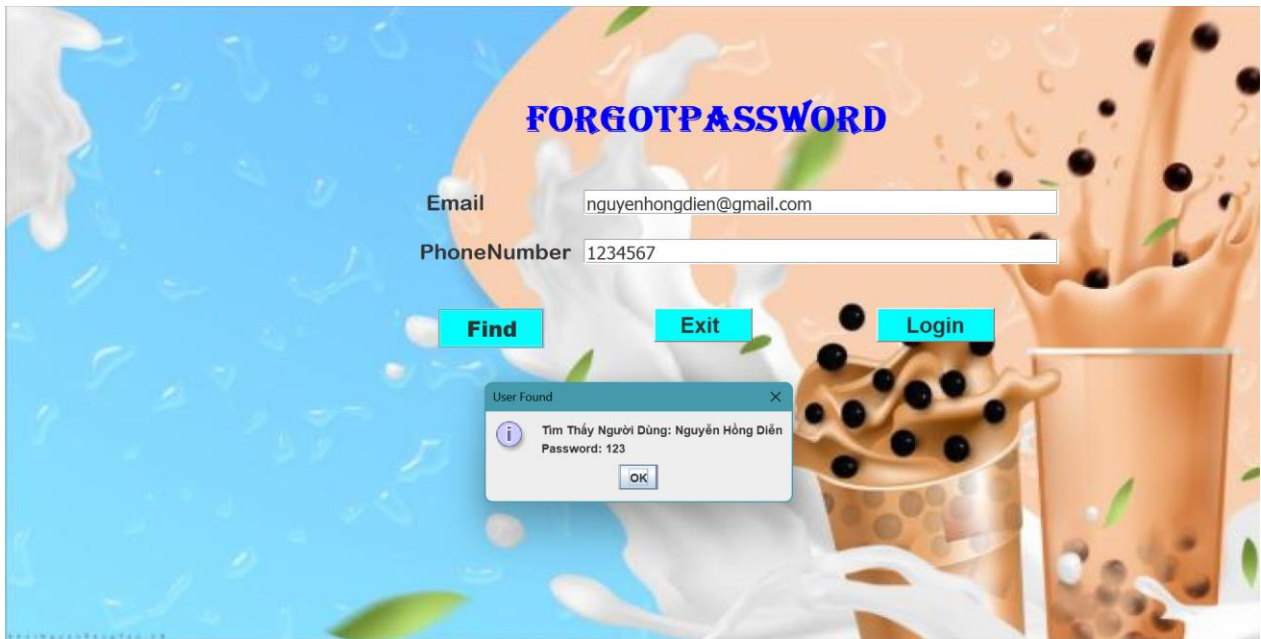
// Đọc danh sách User từ file
private void loadUsersFromFile() {
    File file = new File(FILE_PATH);
    if (!file.exists()) {
        System.out.println("File không tồn tại. Trả về danh sách rỗng.");
        return;
    }

    try (InputStream fis = new FileInputStream(file);
        ObjectInputStream ois = new ObjectInputStream(fis)) {
        Object obj = ois.readObject();

        if (obj instanceof List<?>) {
            List<?> list = (List<?>) obj;
            if (!list.isEmpty() && list.get(0) instanceof User) {
                users = new ArrayList<>();
                for (Object item : list) {
                    users.add((User) item);
                }
                System.out.println("Danh sách User đã được đọc từ file thành công.");
            } else {
                throw new IOException("Dữ liệu trong file không phải là List<User>");
            }
        } else {
            throw new IOException("Dữ liệu trong file không phải là List<?>");
        }
    } catch (IOException | ClassNotFoundException e) {
        System.err.println("Lỗi khi đọc file: " + e.getMessage());
        e.printStackTrace();
    }
}

```


- Màn hình sau khi được cấp lại mật khẩu:



Hình 2.17: Màn hình sau khi tìm lại được mật khẩu.

2.4.11 Tính trừu tượng

- Tính trừu tượng được thể hiện thông qua các Interface giúp cho việc các đối tượng thuộc các lớp khác nhau có thể được sử dụng chung thông qua interface mà chúng triển khai. Đồng thời giúp mã nguồn dễ bảo trì, mở rộng và kiểm thử hơn.
- Interface IUserRepository:
 - Khởi tạo một interface IUserRepository với các phương thức cần thiết để làm việc với các đối tượng User. Các phương thức này bao gồm:
 - save(User user): Lưu một đối tượng User vào file.
 - findByEmailAndPassword(String email, String password): Tìm và trả về đối tượng User dựa trên email và mật khẩu.
 - findByEmail(String email): Tìm và trả về đối tượng User dựa trên email.
 - update(User user): Cập nhật thông tin của một đối tượng User.
 - findAll(): Trả về danh sách tất cả các đối tượng User từ file.
 - clear(): Xóa toàn bộ dữ liệu.

- delete(User user): Xóa một đối tượng User.

```
public interface IUserRepository {  
    public User save(User user);  
    public User findByEmailAndPassword(String email, String password);  
    public User findByEmail(String email);  
    public User update(User user);  
    public List<User> findAll();  
    public void clear();  
    public void delete(User user);  
}
```

- Interface IProductRepository:
 - Khởi tạo một interface IProductRepository với các phương thức cơ bản để làm việc với các đối tượng Product.

```
public interface IProductRepository {  
    public Product save(Product product);  
    public List<Product> findAll();  
    public Product findByName(String name);  
    public void delete(Product product);  
    public void clear();  
}
```

- Interface IBillRepository:
 - Khởi tạo một interface IBillRepository với các phương thức cơ bản để làm việc với các đối tượng Bill.

```
public interface IBillRepository {  
    public Bill save(Bill bill);  
    public List<Bill> findAll();  
    public void clear();  
}
```

PHẦN 3: KẾT QUẢ VÀ BÀI HỌC KINH NGHIỆM

3.1 Nội dung đã thực hiện

3.1.1 Những kiến thức và kỹ năng học được trong quá trình thực hiện đề tài

Trong quá trình thực hiện đề tài “Xây dựng chương trình quản lý quán trà sữa” chúng em đã được học và áp dụng những kiến thức:

- Kiến thức về Java Swing: Để xây dựng giao diện người dùng cho chương trình, chúng em đã được học về các thành phần cơ bản của Java Swing như JPanel, JFrame, JButton, JTable và cách sử dụng chúng để tạo ra các thành phần giao diện và xử lý sự kiện.
- Kỹ năng lập trình Java: Đồ án này giúp chúng em cải thiện được kỹ năng lập trình Java. Chúng em đã học cách sử dụng các cú pháp, các lớp, các phương thức để xây dựng chương trình quản lý. Ngoài ra, kỹ năng debug và kỹ năng tìm kiếm các tài liệu liên quan cũng được cải thiện hơn trong quá trình lập trình.
- Kiến thức về cấu trúc dữ liệu: Trong khi xây dựng chương trình, chúng em đã học cách sử dụng các cấu trúc dữ liệu như các cấu trúc collection, mảng để lưu trữ và quản lý dữ liệu. Điều đó giúp chúng em biết cách lựa chọn cấu trúc dữ liệu sao cho phù hợp với từng bài toán đề ra.
- Kỹ năng thiết kế giao diện: Để tạo ra một giao diện người dùng dễ sử dụng và có tính thẩm mỹ, chúng em đã tìm hiểu cách thiết kế giao diện bằng việc sắp xếp vị trí của các thành phần như nút bấm, trường nhập dữ liệu và bảng hiển thị một cách có tổ chức, phù hợp.

- Kỹ năng làm việc nhóm: Để xây dựng một chương trình quản lý phù hợp, chúng em đã áp dụng các kỹ năng quản lý dự án như lập kế hoạch, phân chia công việc hợp lý và theo dõi tiến độ. Điều đó giúp chúng em tạo ra được chương trình một cách nhanh chóng và hoàn thành đúng hạn.

Qua quá trình thực hiện đề tài “Xây dựng chương trình quản lý quán trà sữa”, chúng em đã được học và biết thêm được nhiều kỹ năng mới như kỹ năng sử dụng Java Swing, sử dụng cấu trúc dữ liệu hợp lý, cách phân tích yêu cầu các bài toán, hiểu sâu hơn về lập trình Java và cách quản lý dự án hiệu quả. Những kiến thức và kỹ năng này giúp cho chúng em có một cái nhìn sâu và vững chắc để xây dựng và phát triển các ứng dụng liên quan sau này.

3.1.2 Chuẩn đầu ra đã đạt được trong quá trình thực hiện đề tài

- Xây dựng thành công chương trình sử dụng Java Swing. Chương trình cung cấp các tính năng cơ bản như thêm, sửa, xóa, cập nhật thông tin. Ngoài ra, chương trình còn cung cấp tính năng nâng cao hơn đó là tìm lại mật khẩu của tài khoản thông qua email và số điện thoại. Các thông tin và hóa đơn người dùng được lưu một cách bảo mật trong file thông qua các phương thức của lớp `ObjectInputStream` và `ObjectOutputStream`.
- Giao diện được thiết kế phù hợp, dễ dàng sử dụng.

3.1.3 Bài học kinh nghiệm sau khi kết thúc đề tài

- Kỹ năng lập trình và thiết kế giao diện: Cần trang bị cho bản thân thật chắc các kiến thức lập trình căn bản và tìm hiểu về cách thiết kế giao diện sao cho bố cục hài hòa.
- Quy trình phân tích và quản lý: Phân tích yêu cầu bài toán và quản lý dự án là quan trọng để xây dựng chương trình chất lượng. Chúng em còn hiểu ra được

tầm quan trọng của việc lên kế hoạch cụ thể trước khi bắt đầu viết mã giúp việc xây dựng chương trình.

- Kỹ năng giao tiếp và làm việc nhóm: Trong quá trình thực hiện đề tài, chúng em nhận thấy khả năng giao tiếp tốt giúp cho quá trình làm việc hiệu quả hơn. Hơn nữa, chúng em còn học được cách lắng nghe ý kiến, học cách trao đổi thông tin sao cho rõ ràng, dễ hiểu để từ đó đưa ra các hướng đi cùng giải pháp phù hợp.

3.2 Hướng phát triển

Sau khi hoàn thành đề tài “Xây dựng chương trình quản lý quán trà sữa”, chúng em nhận thấy có một số hướng có thể được phát triển:

- Cải thiện giao diện : Mặc dù cho giao diện hiện tại đã đáp ứng được cơ bản yêu cầu về mặt thiết kế, dễ dàng sử dụng song vẫn còn những không gian để cải tiến nhằm tối ưu hóa. Tùy chỉnh bố cục hoặc thêm hiệu ứng bắt mắt sẽ làm tăng trải nghiệm người dùng, làm chương trình trở nên hấp dẫn, thú vị hơn.
- Bổ sung tính năng thông báo: Tính năng xem thông tin đồ uống có thể được thêm vào để giúp người sử dụng có cái nhìn chi tiết để dễ dàng đưa ra lựa chọn.
- Tích hợp cơ sở dữ liệu: Một hướng phát triển khác đó là tích hợp cơ sở dữ liệu thực để lưu trữ các thông tin, điều đó giúp việc quản lý hiệu quả hơn.

TÀI LIỆU THAM KHẢO

- [1] Giáo trình lập trình Java
- [2] <https://www.javatpoint.com/java-tutorial>
- [3] <https://www.w3schools.com/java/>
- [4] <https://dev.java/learn/eclipse/>
- [5] Giáo trình lập trình hướng đối tượng