

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

=====***=====



BÁO CÁO THÍ NGHIỆM/ THỰC NGHIỆM
LẬP TRÌNH JAVA

ĐỀ TÀI: XÂY DỰNG HỆ THỐNG QUẢN LÝ SẢN
PHẨM ĐIỆN TỬ CHO CÔNG TY YP

Giảng viên : TS. Hoàng Quang Huy

Nhóm : 15

Lớp : 20232IT6019006

Thành viên : Bùi Thị Ngọc Huyền -2022604722

Ngô Thành Đoàn -2022603699

Bùi Văn Đạt -2022604728

*****Hà Nội-2024*****

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn giảng viên hướng dẫn TS. Hoàng Quang Huy đã luôn tận tâm hỗ trợ và hướng dẫn chúng em. Những góp ý và sự chỉ dẫn của thầy trong suốt quá trình thực hiện đề tài đã giúp nhóm chúng em hoàn thiện bài tập một cách tốt nhất.

Qua quá trình nghiên cứu và học tập, chúng em đã tích lũy được nhiều kiến thức và kinh nghiệm quý báu. Những điều này sẽ là hành trang quan trọng giúp chúng em phát triển và hoàn thiện bản thân trong tương lai.

Trong quá trình thực hiện đề tài, do kiến thức còn hạn chế và thiếu kinh nghiệm thực tiễn, chúng em không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được ý kiến đóng góp từ thầy, cô và những người quan tâm để bài nghiên cứu của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

PHẦN 1: MỞ ĐẦU	6
1.1 Giới thiệu đề tài	6
1.2 Tính cấp thiết của đề tài	6
1.3 Lý do chọn đề tài	6
1.4 Công cụ thực hiện đề tài	7
PHẦN 2: KẾT QUẢ NGHIÊN CỨU	8
2.1 Giới thiệu	8
2.2 Khảo sát hệ thống	8
2.2.1 Khảo sát sơ bộ	8
2.2.2 Tài liệu đặc tả yêu cầu	9
2.3 Phân tích hệ thống	10
2.3.1 Mô hình hóa chức năng hệ thống	10
2.3.2 Mô hình hóa dữ liệu.....	12
2.4 Thực hiện bài toán	15
2.4.1 Đăng nhập	15
2.4.2 Đăng ký	17
2.4.3 Quên mật khẩu.....	21
2.4.4 Menu sản phẩm.....	24
2.4.5 Thêm sản phẩm.....	25
2.4.6 Sửa sản phẩm.....	27
2.4.7 Xóa sản phẩm	29
2.4.8 Tìm kiếm	30
2.4.9 Sắp xếp	33
PHẦN 3: KẾT QUẢ VÀ BÀI HỌC KINH NGHIỆM	35
3.1 Nội dung đã thực hiện.....	35
3.1.1 Những kiến thức và kỹ năng học được trong quá trình thực hiện đề tài	35
3.1.2 Chuẩn đầu ra đã đạt được trong quá trình thực hiện đề tài	35
3.1.3 Bài học kinh nghiệm sau khi kết thúc đề tài	36

3.2 Hướng phát triển.....	36
TÀI LIỆU THAM KHẢO	37

DANH SÁCH HÌNH ẢNH

Hình 2. 1 Biểu đồ use case	11
Hình 2. 2 Biểu đồ lớp	14
Hình 2. 3 Màn hình đăng nhập	15
Hình 2. 4 Màn hình đăng ký	18
Hình 2. 5 Khởi tạo bảng thông tin đăng ký người dùng	20
Hình 2. 6 Database quản lý thông tin đăng ký	21
Hình 2. 7 Màn hình quên mật khẩu.....	21
Hình 2. 8 Màn hình menu giao diện	24
Hình 2. 9 Màn hình thêm sản phẩm.....	25
Hình 2. 10 Màn hình sửa sản phẩm	27
Hình 2. 11 Màn hình xóa sản phẩm	29
Hình 2. 12 Màn hình tìm kiếm sản phẩm.....	30
Hình 2. 13 Màn hình sắp xếp sản phẩm.....	33

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 khổng lồ một cách nhanh gọn 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 sản phẩm điện tử 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 sản phẩm điện tử ra đời là một ví dụ cho việc tối ưu hóa toàn bộ quy trình từ quản lý nguyên liệu, nhân sự, đến linh hoạt trong quản lý chất lượng khu làm việc,...

1.2 Tính cấp thiết của đề tài

Việc quản lý 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.

Quản lý các tác vụ bằng phần mềm giúp công việc đạt tính chính xác cao, giúp việc theo dõi trở nên dễ dàng, khoa học, nhanh chóng hơn.

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 HỆ THỐNG QUẢN LÝ SẢN PHẨM ĐIỆN TỬ CHO CÔNG TY YP”. 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 IDE

Eclipse IDE là một môi trường phát triển tích hợp (IDE) mã nguồn mở phổ biến, chủ yếu được sử dụng cho lập trình Java nhưng cũng hỗ trợ nhiều ngôn ngữ khác thông qua các plugin. Nó cung cấp các công cụ mạnh mẽ cho viết mã, gỡ lỗi, và quản lý dự án. Eclipse có một cộng đồng lớn và nhiều tài nguyên hỗ trợ. Phiên bản mới nhất của Eclipse thường được cập nhật với các tính năng và cải tiến mới. .

CASE Studio

CASE Studio là công cụ vẽ mô hình dữ liệu chuyên nghiệp, có thể tùy chỉnh hỗ trợ lập trình viên hoặc nhân viên thiết kế dữ liệu trong việc vẽ mô hình Entity Relationship Diagrams (ERD) và Data Flow Diagrams (DFD) cũng như tạo scripts SQL cho nhiều cơ sở dữ liệu một cách tự động

IBM Rational Rose

Rational Rose là một công cụ lập mô hình trực quan mạnh mẽ giúp bạn phân tích và thiết kế các hệ thống phần mềm hướng đối tượng. Nó được dùng để lập mô hình hệ thống trước khi bạn viết mã lệnh

JDK 17

JDK 17 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 17 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 HỆ THỐNG QUẢN LÝ SẢN PHẨM ĐIỆN TỬ CHO CÔNG TY YP**

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. 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 sản phẩm điện tử cho công ty YP 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 sản phẩm điện tử cho công ty YP nhằm tối ưu hóa các quy trình liên quan, từ việc lưu trữ thông tin, tìm kiếm, thêm, sửa xóa, sắp xếp cho đến theo dõi quá trình làm việc cũng như mua bán.

Về phạm vi, hệ thống cần đáp ứng được các yêu cầu cơ bản như quản lý được danh mục sản phẩm của công ty qua các chức năng sản phẩm điện tử cho công ty YP.

2.2.2 Tài liệu đặc tả yêu cầu

2.2.2.1 Yêu cầu chức năng

- **Đối với người quản trị viên**
 - Đăng ký: Cho phép người quản trị viên tạo tài khoản mới trong hệ thống bằng cách cung cấp thông tin cá nhân cần thiết
 - 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 Laptop :
 - Sắp xếp sản phẩm theo mã sản phẩm, tên sản phẩm, giá, số lượng
 - Thêm sản phẩm
 - Sửa sản phẩm
 - Xoá sản phẩm
 - Hiển thị danh sách sản phẩm
 - Tìm kiếm sản phẩm
 - Quản lý sản phẩm Tivi :
 - Sắp xếp sản phẩm theo mã sản phẩm, tên sản phẩm, giá, số lượng
 - Thêm sản phẩm
 - Sửa sản phẩm
 - Xoá sản phẩm
 - Hiển thị danh sách sản phẩm
 - Tìm kiếm sản phẩm
 - Quản lý sản phẩm Mobile:
 - Sắp xếp sản phẩm theo mã sản phẩm, tên sản phẩm, giá, số lượng
 - Thêm sản phẩm
 - Sửa sản phẩm
 - Xoá sản phẩm
 - Hiển thị danh sách sản phẩm
 - Tìm kiếm sản phẩm

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 tốt được người dùng tin cậy, không rò rỉ thông tin người dùng
- 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 Dat.
- Ngôn ngữ lập trình: Java - Swing
- Môi trường cài đặt: Windows.

2.3 Phân tích hệ thống

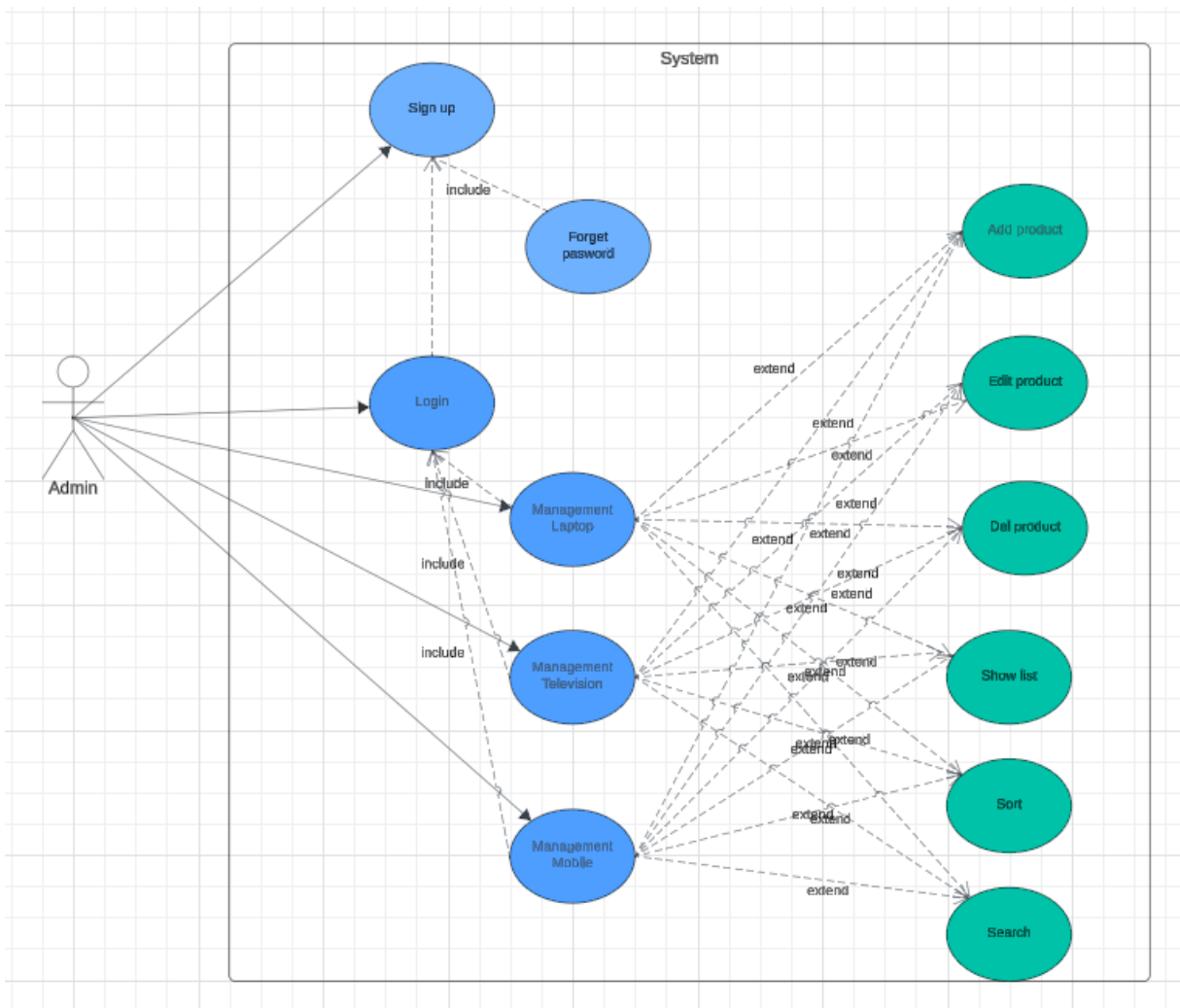
2.3.1 Mô hình hóa chức năng hệ thống

Actor chính sử dụng chương trình và vai trò:

Quản trị viên: Đăng nhập, đăng ký, thay đổi mật khẩu, quản lý Laptop, quản lý Tivi, quản lý Mobile

Mô tả sơ bộ các use case

- Đăng nhập: cho phép quản trị viên và nhân viên đăng nhập tài khoản vào chương trình
- Quên mật khẩu: cho phép người quản trị sửa lại mật khẩu đăng nhập
- Đăng ký: cho phép quản trị viên đăng ký tài khoản vào chương trình
- Quản lý Laptop: cho phép quản trị viên sắp xếp sản phẩm, Thêm sản phẩm, Sửa sản phẩm, Xoá sản phẩm, Hiển thị danh sách sản phẩm, Tìm kiếm sản phẩm Laptop
- Quản lý Tivi: cho phép quản trị viên sắp xếp sản phẩm, Thêm sản phẩm, Sửa sản phẩm, Xoá sản phẩm, Hiển thị danh sách sản phẩm, Tìm kiếm sản phẩm Tivi
- Quản lý Mobile: cho phép quản trị viên sắp xếp sản phẩm, Thêm sản phẩm, Sửa sản phẩm, Xoá sản phẩm, Hiển thị danh sách sản phẩm, Tìm kiếm sản phẩm.



Hình 2. 1 Biểu đồ use case

ID	Tên use case	Mô tả ngắn gọn	Chức năng	Ghi chú
UC_001	Login	Cho phép quản trị viên và nhân viên đăng nhập vào hệ thống	Đăng nhập	Admin
UC_002	Sign up	Cho phép nhân viên đăng ký tài khoản trong hệ thống	Đăng ký	Admin
UC_003	Management Laptop	Cho phép quản trị viên sắp xếp sản phẩm, Thêm sản phẩm, Sửa sản phẩm, Xóa sản phẩm, Hiện thị danh sách sản phẩm, Tìm kiếm sản phẩm Laptop	Quản lý Laptop	Admin

UC_004	Management Television	Cho phép quản trị viên sắp xếp sản phẩm, Thêm sản phẩm, Sửa sản phẩm, Xoá sản phẩm, Hiển thị danh sách sản phẩm, Tìm kiếm sản phẩm Tivi	Quản lý Tivi	Admin
UC_005	Management Mobile	Cho phép quản trị viên sắp xếp sản phẩm, Thêm sản phẩm, Sửa sản phẩm, Xoá sản phẩm, Hiển thị danh sách sản phẩm, Tìm kiếm sản phẩm Mobile	Quản lý Mobile	Admin
UC_006	Forget password	Cho phép nhân viên lấy lại mật khẩu của tài khoản trong hệ thống	Quên mật khẩu	Admin

2.3.2 Mô hình hóa dữ liệu

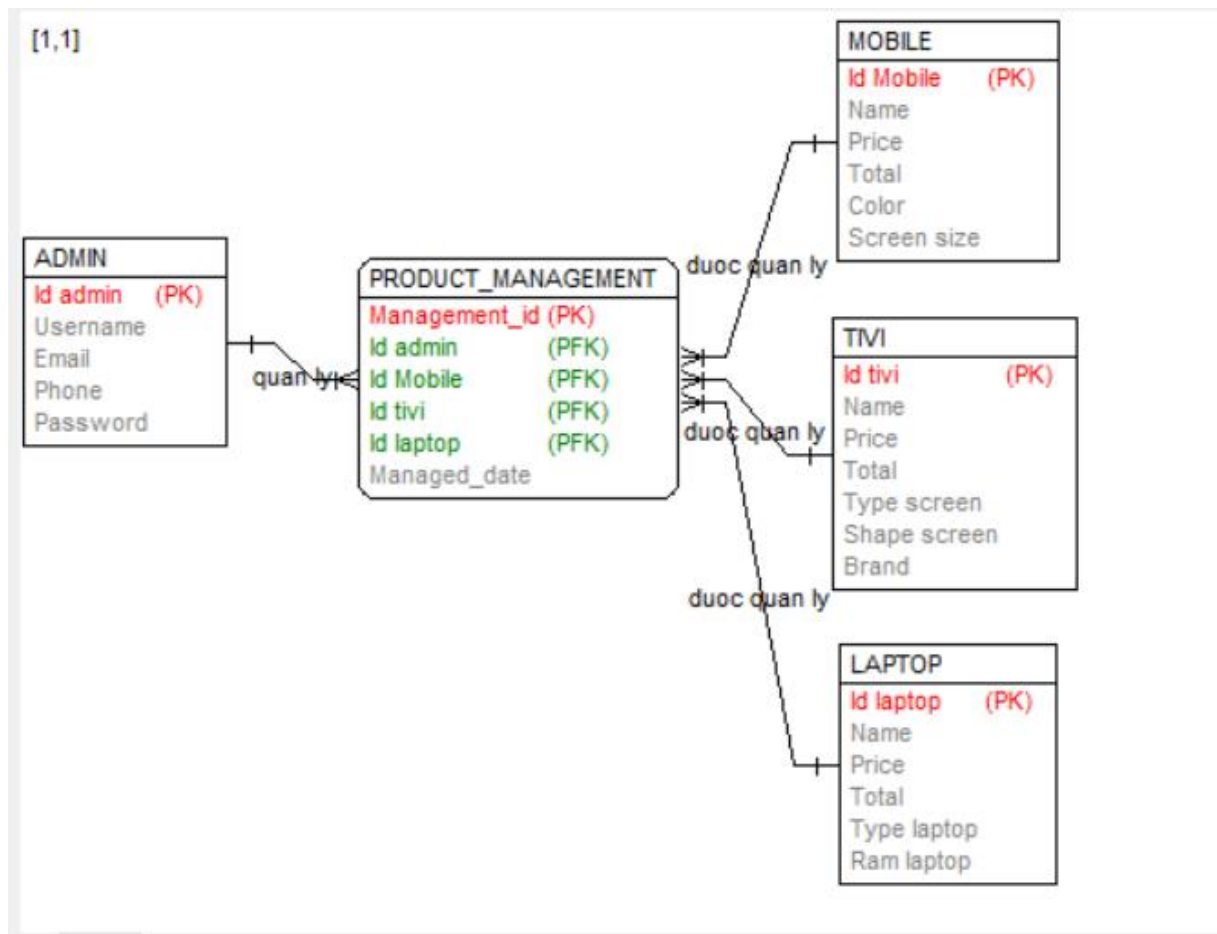
2.3.2.1 Thông tin lưu trữ

Thuộc tính của các lớp:

- Lớp Admin gồm có các thuộc tính:
 - Id admin: thuộc tính khóa xác định mã định danh của người quản trị
 - Username: thuộc tính xác định tên của người quản trị.
 - Email: thuộc tính xác định email của người quản trị
 - Phone: thuộc tính xác định số điện thoại của người quản trị.
 - Password: thuộc tính xác định mật khẩu của người quản trị
- Lớp Product Management các thuộc tính:
 - Management id: thuộc tính khóa để xác định thứ tự của sản phẩm
 - Management date: thuộc tính xác định ngày quản lý sản phẩm
 - ID Mobile: thuộc tính khóa để xác định thứ tự của Mobile
 - Id admin: thuộc tính khóa xác định mã định danh của người quản trị
 -
- Lớp Mobile có các thuộc tính:
 - ID Mobile: thuộc tính khóa để xác định thứ tự
 - Name: thuộc tính xác định tên của sản phẩm
 - Price: thuộc tính xác định giá của sản phẩm

- Total: thuộc tính xác định số lượng sản phẩm
- Color: thuộc tính xác định màu của sản phẩm
- Screen size: thuộc tính xác định kích thước của sản phẩm
- Lớp Tivi có các thuộc tính:
 - ID Tivi: thuộc tính khóa để xác định thứ tự của Tivi
 - Name: thuộc tính xác định tên của sản phẩm
 - Price: thuộc tính xác định giá của sản phẩm
 - Total: thuộc tính xác định số lượng sản phẩm
 - Type screen: thuộc tính xác định kiểu màn hình của sản phẩm
 - Shape screen: thuộc tính xác định kích thước màn hình của sản phẩm
 - Branch: thuộc tính xác định hãng của sản phẩm
- Lớp Laptop có các thuộc tính:
 - ID Laptop: thuộc tính khóa để xác định thứ tự của Laptop
 - Name: thuộc tính xác định tên của sản phẩm
 - Price: thuộc tính xác định giá của sản phẩm
 - Total: thuộc tính xác định số lượng sản phẩm
 - Type Laptop: thuộc tính xác định kiểu màn hình của sản phẩm
 - Ram Laptop: thuộc tính xác định ram lưu trữ của sản phẩm

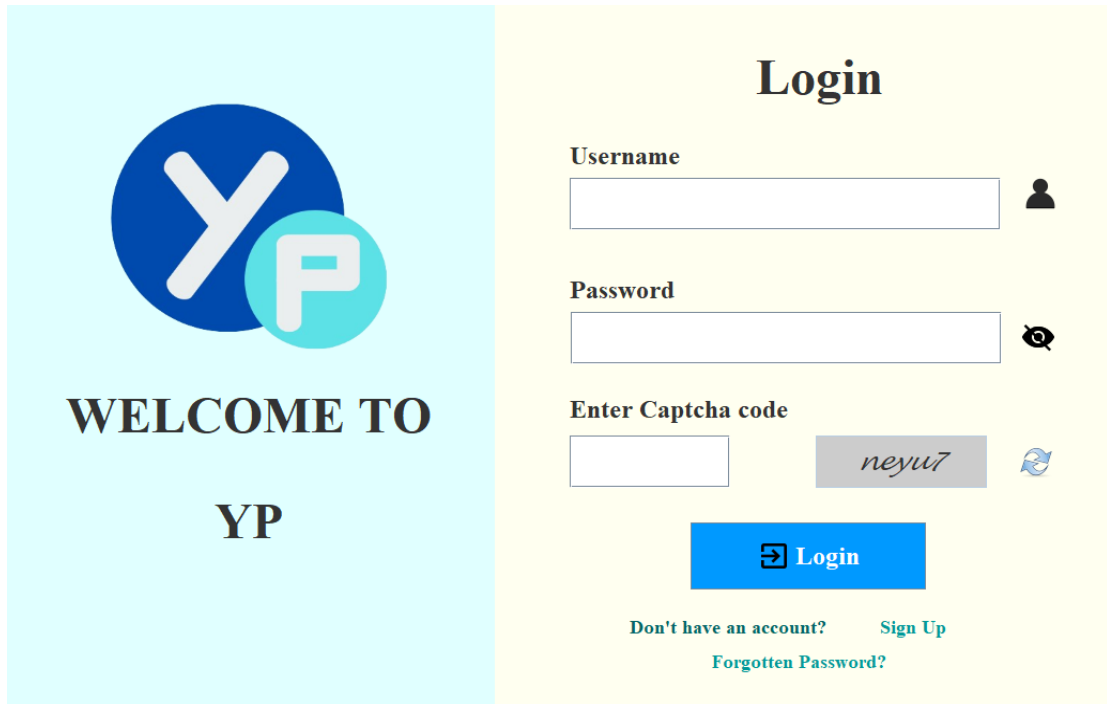
2.3.2.2 Biểu đồ lớp



Hình 2. 2 Biểu đồ lớp

2.4 Thực hiện bài toán

2.4.1 Đăng nhập



Hình 2. 3 Màn hình đăng nhập

- Màn hình giao diện chức năng đăng nhập bao gồm:
- Trường nhập username, password, captcha code.
- Nút button “Login” dùng để đăng nhập.
- Label “Sign up” dẫn đến trang đăng ký tài khoản.
- Label “Forgot Password” dẫn đến trang tìm lại tài khoản và thay đổi mật khẩu mới khi quên mật khẩu.
- Admin đăng nhập với username là “admin” với password là “admin” để thực hiện các quyền của admin.
- Hướng đối tượng:
- Class Login được kế thừa từ javax.swing.JFrame và là thể hiện của lớp ActionListener.

```
public class Login extends javax.swing.JFrame implements ActionListener {
```

- Thao tác với cơ sở dữ liệu

Hệ thống thực hiện thao tác với database thông qua lớp MyConnection dùng để lưu trữ thông tin đăng ký của người dùng. Đối tượng nếu có sự thay đổi sẽ được lưu tại database.

```

package connection;

import java.sql.*;

public class MyConnection {
    public static final String url = "jdbc:mysql://localhost:3306/qlspdt_yp";
    public static final String username = "root";
    public static final String password = "ngochuyen0527";
    public static Connection con = null;

    public static Connection getConnection() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            // Tạo kết nối tới cơ sở dữ liệu
            con = DriverManager.getConnection(url, username, password);
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException ex) {
            ex.printStackTrace();
        }
        return con;
    }
}

```

- Không cho phép bỏ trống các trường dữ liệu

```

private boolean isEmpty() {
    if (txtUsername.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Username is required", "Warning", 2);
        return false;
    }
    if (String.valueOf(jPasswordField1.getPassword()).isEmpty()) {
        JOptionPane.showMessageDialog(this, "Password is required", "Warning", 2);
        return false;
    }

    if (txtCode.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Captcha code is required", "Warning", 2);
        return false;
    }
    return true;
}

```

- Kiểm tra tính hợp lệ của username, password, captcha code đã nhập đúng hay chưa.


```

public boolean checkValidation() {
    // kiểm tra tính hợp lệ của username.
    if (!lblErrorUsernameL.getText().equalsIgnoreCase("")) {
        JOptionPane.showMessageDialog(this, "Incorrect username", "Error",
            JOptionPane.ERROR_MESSAGE);
        return false;
    }

    // kiểm tra password.
    if (!lblErrorPassL.getText().equalsIgnoreCase("")) {
        JOptionPane.showMessageDialog(this, "Incorrect password", "Error",
            JOptionPane.ERROR_MESSAGE);
        return false;
    }
    //kiểm tra mã captcha
    if (!txtCode.getText().equalsIgnoreCase(txtCaptchaL.getText())) {
        JOptionPane.showMessageDialog(this, "Captcha code does not match", "Error",
            JOptionPane.ERROR_MESSAGE);
        return false;
    }
    return true;
}

```

- Xử lý đăng nhập: lấy thông tin đăng nhập và kiểm tra với giá trị tài khoản khởi tạo, nếu hợp lệ thì chuyển sang màn hình trang chủ, nếu không đưa ra thông báo.

```

private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (isEmpty()) {
        if (checkValidation()) {
            UserDashboard ud = new UserDashboard();
            ud.setVisible(true);
            ud.pack();
            this.dispose();
        }
    }
}
} //GEN-LAST:event_btnLoginActionPerformed

```

2.4.2 Đăng ký

- Giao diện sau khi nhấn “Sign up”

Hình 2. 4 Màn hình đăng ký

- Màn hình giao diện chức năng đăng ký bao gồm: Trường nhập liệu tên đăng nhập, email, mật khẩu, xác thực mật khẩu, số điện thoại, mã captcha.
- 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 SignUp extends javax.swing.JFrame {
```

Thao tác với cơ sở dữ liệu

Hệ thống thực hiện thao tác với database thông qua lớp UserDao dùng để lưu trữ thông tin đăng ký của người dùng. Đối tượng nếu có sự thay đổi sẽ được lưu tại database.

```
UserDao user = new UserDao();
```

- Không cho phép bỏ trống các trường dữ liệu

```

* @return
*/
public boolean isEmpty() {
    if (txtUsername.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Username is required", "Warning", 2);
        return false;
    }

    if (txtEmail.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Email is required", "Warning", 2);
        return false;
    }

    if (String.valueOf(txtPassword.getPassword()).isEmpty()) {
        JOptionPane.showMessageDialog(this, "Password is required", "Warning", 2);
        return false;
    }

    if (String.valueOf(txtVerifyPassword.getPassword()).isEmpty()) {
        JOptionPane.showMessageDialog(this, "Verify password is required", "Warning", 2);
        return false;
    }

    if (txtPhone.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Phone number is required", "Warning", 2);
        return false;
    }

    if (txtCode.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Captcha code is required", "Warning", 2);
        return false;
    }
    return true;
}

```

- Kiểm tra tính hợp lệ của username, email, verify password, phone, captcha code đã nhập đúng hay chưa.

```

//
public boolean checkValidation() {
    // kiểm tra tính hợp lệ của username.
    if (!lblErrorUsername.getText().equalsIgnoreCase("")) {
        JOptionPane.showMessageDialog(this, "Invalid username. Please enter username again", "Error",
            JOptionPane.ERROR_MESSAGE);
        return false;
    }

    // kiểm tra tính hợp lệ của địa chỉ email.
    if (!lblErrorEmail.getText().equalsIgnoreCase("")) {
        JOptionPane.showMessageDialog(this, "Invalid email address. Please enter email again", "Error",
            JOptionPane.ERROR_MESSAGE);
        return false;
    }

    // kiểm tra password.
    if (!lblErrorPass.getText().equalsIgnoreCase("")) {
        JOptionPane.showMessageDialog(this, "Password does not match. Please enter password again", "Error",
            JOptionPane.ERROR_MESSAGE);
        return false;
    }

    // sdt phải bắt đầu bằng số 0 và có 10 chữ số
    if (!lblErrorPhone.getText().equalsIgnoreCase("")) {
        JOptionPane.showMessageDialog(this, "Invalid phone number. Please enter phone number again", "Error",
            JOptionPane.ERROR_MESSAGE);
        return false;
    }

    if (!txtCode.getText().equalsIgnoreCase(txtCaptchaS.getText())) {
        JOptionPane.showMessageDialog(this, "Captcha code does not match. Please enter code again", "Error",
            JOptionPane.ERROR_MESSAGE);
        return false;
    }

    return true;
}

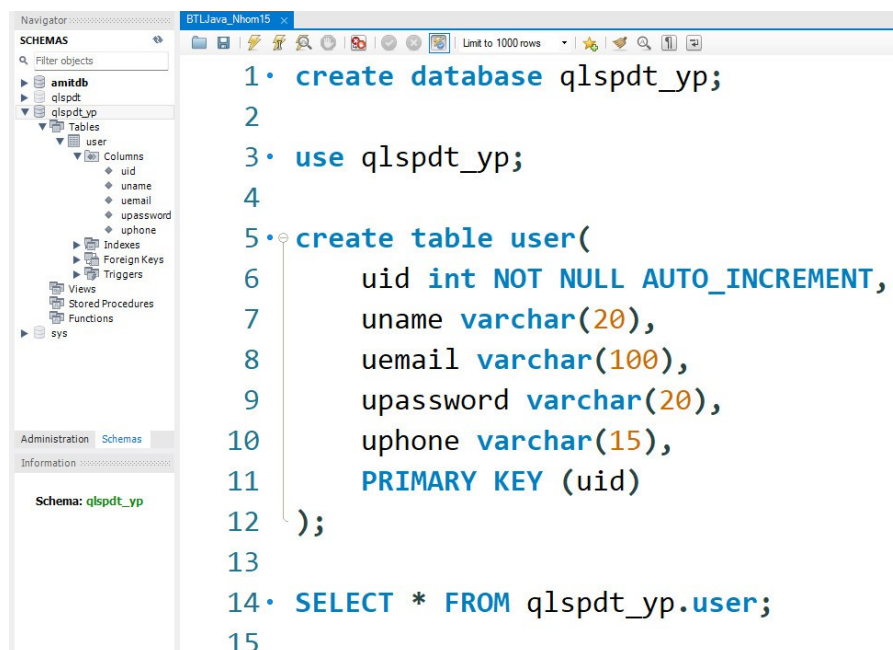
```

- Xử lý đăng ký: lấy thông tin đăng ký và kiểm tra trong database, nếu hợp lệ thì thêm thông tin người dùng vào database và chuyển sang màn hình login để đăng nhập, nếu không thì đưa ra thông báo.

```
// insert data into user table
public void insert(int id, String username, String email, String password, String phone) {
    String sql = "insert into user values(?,?,?,?,?)";
    try {
        ps = con.prepareStatement(sql);
        ps.setInt(1, id);
        ps.setString(2, username);
        ps.setString(3, email);
        ps.setString(4, password);
        ps.setString(5, phone);

        if (ps.executeUpdate() > 0) {
            JOptionPane.showMessageDialog(null, "User added successfully");
        }
    } catch (SQLException ex) {
        Logger.getLogger(UserDao.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int id = Integer.parseInt(txtId.getText());
    String username = txtUsername.getText();
    String email = txtEmail.getText();
    String password = String.valueOf(txtPassword.getPassword());
    String phone = txtPhone.getText();
    if (isEmpty()) {
        if (checkValidation()) {
            user.insert(id, username, email, password, phone);
            new Login().setVisible(true);
            this.dispose();
        }
    }
}
} // GEN-LAST:event_btnSaveActionPerformed
```



Hình 2. 5 Khởi tạo bảng thông tin đăng ký người dùng

uid	uname	uemail	upassword	uphone
1	Ngochuyenn	huyen527@gmail.com	123	4564575647
2	huyen	user@gmail.com	123	56547457
3	btnh527	huyen@gmail.com	123	5264245624
4	huyenne	huyenne@gmail.com	12345	435435234
5	huynne	huynne@gmail.com	12345	56356356
6	hello	helloe@gmail.com	123	3435346576
7	huyen	h@gmail.com	1234	0964721692
8	huyenn	hh@gmail.com	123	0956748289
9	admin	admin@gmail.com	admin	0123456789

Hình 2. 6 Database quản lý thông tin đăng ký

2.4.3 Quên mật khẩu

- Giao diện sau khi nhấn “Forgotten password”

Forgot Password

Email

New Password

Verify Password

Enter Captcha code

neyu7

Hình 2. 7 Màn hình quên mật khẩu

- Màn hình giao diện chức năng thay đổi mật khẩu bao gồm: Trường nhập liệu email, mật khẩu mới, xác thực mật khẩu, mã captcha.
- Hướng đối tượng:
 - Encapsulation (Đóng gói): Các thuộc tính (fields) và phương thức (methods) được gói gọn trong lớp ForgotPass .
 - Lớp ForgotPass kế thừa từ javax.swing.JFrame, do đó ForgotPass có thể sử dụng các phương thức và thuộc tính của JFrame.

```
public class ForgotPass extends javax.swing.JFrame {
```

- Thao tác với cơ sở dữ liệu
- Hệ thống thực hiện thao tác với database thông qua lớp ForgotPasswordDao dùng để tìm kiếm thông tin email đăng ký của người dùng có tồn tại hay chưa. Đối tượng nếu có sự thay đổi sẽ được lưu tại database.

```
ForgotPasswordDao fg = new ForgotPasswordDao();
```

- Không cho phép bỏ trống các trường dữ liệu

```
public boolean isEmpty() {
    if (txtEmailF.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Email is required to search", "Warning", 2);
        return false;
    }

    if (String.valueOf(txtPasswordF.getPassword()).isEmpty()) {
        JOptionPane.showMessageDialog(this, "Password is required", "Warning", 2);
        return false;
    }

    if (String.valueOf(txtVerifyPasswordF.getPassword()).isEmpty()) {
        JOptionPane.showMessageDialog(this, "Verify password is required", "Warning", 2);
        return false;
    }

    if (txtCode.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Captcha code is required", "Warning", 2);
        return false;
    }
    return true;
}
```

Kiểm tra tính hợp lệ của verify password, captcha code đã nhập đúng hay chưa.

```

public boolean checkValidation() {
    // kiểm tra password.
    if (!lblErrorPassF.getText().equalsIgnoreCase("")) {
        JOptionPane.showMessageDialog(this, "Password does not match.", "Error",
            JOptionPane.ERROR_MESSAGE);
        return false;
    }

    if (!txtCode.getText().equalsIgnoreCase(txtCaptchaF.getText())) {
        JOptionPane.showMessageDialog(this, "Captcha code does not match.", "Error",
            JOptionPane.ERROR_MESSAGE);
        return false;
    }

    return true;
}

```

- Kiểm tra email đã dùng để đăng ký hay chưa, nếu đã tồn tại thì mới có thể thay đổi password, không thì hiện thông báo.

```

public boolean checkEmailExist() {
    if (!lblErrorEmailF.getText().equalsIgnoreCase("")) {
        JOptionPane.showMessageDialog(this, "Email address doesn't exists", "Error",
            return false;
        }
    }
    return true;
}

```

- Xử lý đăng ký: lấy thông tin đăng ký và kiểm tra trong database, nếu hợp lệ thì thêm thông tin người dùng vào database và chuyển sang màn hình login để đăng nhập, nếu không thì đưa ra thông báo.

```

// insert data into user table
public void insert(int id, String username, String email, String password, String phone) {
    String sql = "insert into user values(?,?,?,?,?)";
    try {
        ps = con.prepareStatement(sql);
        ps.setInt(1, id);
        ps.setString(2, username);
        ps.setString(3, email);
        ps.setString(4, password);
        ps.setString(5, phone);

        if (ps.executeUpdate() > 0) {
            JOptionPane.showMessageDialog(null, "User added successfully");
        }
    } catch (SQLException ex) {
        Logger.getLogger(UserDao.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```


- Encapsulation (Đóng gói): Các thuộc tính (fields) và phương thức (methods) được gói gọn trong lớp MainMenu.
- Lớp MainMenu kế thừa từ javax.swing.JFrame, do đó MainMenu có thể sử dụng các phương thức và thuộc tính của JFrame.

```
public class MainMenu extends javax.swing.JFrame {
```

2.4.5 Thêm sản phẩm

The screenshot shows the 'Laptop Management System' interface. On the left, there is a form for adding a new product with fields for Product ID, Product Name, Product Price, Product Total, Laptop Type, and Laptop's Ram. Below the form is a 'Function' section with buttons for Add, Edit, Delete, and Show list. On the right, there is a search bar and a table displaying a list of products with columns for Product ID, Product Name, Product Price (\$), Product Total, Laptop Type, and Laptop's Ram.

Product ID	Product Name	Product Price (\$)	Product Total	Laptop Type	Laptop's Ram
1	Asus FTX 15	312.0	1	Laptop Gami...	8
89	Dell XPS 13	617.0	47	Laptop Gami...	8
30	Razer Blade ...	882.0	68	Ultrabook	16
67	Acer nitro 5	608.0	8	Laptop Gami...	16
46	Lenovo Thin...	764.0	25	Student Lapt...	16
54	Acer nitro 5	383.0	28	Student Lapt...	16
90	Microsoft Su...	797.0	26	Budget Lapt...	16
55	Microsoft Su...	442.0	97	Ultrabook	16
41	Lenovo Thin...	508.0	63	Budget Lapt...	8
52	Lenovo Thin...	780.0	63	Ultrabook	16
98	Razer Blade ...	740.0	99	Student Lapt...	16
69	Dell XPS 13	613.0	0	Student Lapt...	16
25	Microsoft Su...	766.0	26	Ultrabook	16
24	Acer nitro 5	486.0	31	Laptop Gami...	8
32	Dell XPS 13	581.0	64	Laptop Gami...	8
8	Dell XPS 13	525.0	56	Student Lapt...	8
47	Dell XPS 13	881.0	11	Ultrabook	8
63	Dell XPS 13	603.0	78	Laptop Gami...	8

Hình 2. 9 Màn hình thêm sản phẩm

- Màn hình giao diện chức năng thêm sản phẩm bao gồm: Trường nhập dữ liệu mã sản phẩm, tên sản phẩm, giá sản phẩm, số lượng sản phẩm, loại laptop, số RAM của laptop, bảng thông tin của các sản phẩm, button “Add” để thêm sản phẩm.

- Kiểm tra xem có ô nào bị trống dữ liệu:

```
if (Product_id.isEmpty() || Product_name.isEmpty() || Product_price.isEmpty()
    || Product_total.isEmpty() || Laptop_type.isEmpty() || Laptop_ram.isEmpty()) {
    JOptionPane.showMessageDialog(contentPane, "There is a cell that has not been entered!",
        "Error", JOptionPane.ERROR_MESSAGE);
    return;
}
```

- Báo lỗi khi mã sản phẩm bị trùng:

```
if (Product_id.isEmpty() || Product_name.isEmpty() || Product_price.isEmpty()
    || Product_total.isEmpty() || Laptop_type.isEmpty() || Laptop_ram.isEmpty()) {
    JOptionPane.showMessageDialog(contentPane, "There is a cell that has not been entered!"
        , "Error", JOptionPane.ERROR_MESSAGE);
    return;
}
```

- Thêm sản phẩm vào danh sách ,sản phẩm sẽ hiển thị trên bảng theo nếu đồng ý thêm sẽ thêm và báo thực hiện thành công:

```
int c = JOptionPane.showConfirmDialog(this, "Do you want to add this product?");
if (c == JOptionPane.OK_OPTION) {
    // vector chứa thông tin để cho vào bảng

    Vector a = new Vector<Laptop>();
    a.add(Product_id);
    a.add(Product_name);
    a.add(Product_price);
    a.add(Product_total);
    a.add(Laptop_type);
    a.add(Laptop_ram);
    dtm.addRow(a); // thêm dữ liệu vào bảng

    dtm.fireTableDataChanged(); // cập nhật dữ liệu

    // thêm vào danh sách
    this.list.add(new Laptop(pi, Product_name, pp, pt, Laptop_type, lr));
} else {
    return;
}

JOptionPane.showMessageDialog(contentPane, "Successfully added");
// set các ô nhập dữ liệu về trống để tiện cho việc nhập thông tin mới
this.txtProduct_id.setText("");
this.txtProduct_name.setText("");
this.txtProduct_price.setText("");
this.txtProduct_total.setText("");
this.txtLaptop_type.setText("");
this.txtLaptop_ram.setText("");
```

- Sử dụng tập hợp List:

```
public List<Laptop> list = new ArrayList<Laptop>();
```

1. Lí do sử dụng:

- 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. Mục đích sử dụng tập hợp List trong đoạn mã:
- Chứa danh sách các sản phẩm: List<Laptop> được sử dụng để chứa danh sách các đối tượng Laptop, mỗi đối tượng Laptop đại diện cho một sản phẩm với các thuộc tính như tên sản phẩm, giá cả, mô tả, v.v.
3. Các phép toán đã sử dụng:
- Duyệt qua danh sách: sử dụng vòng lặp for

```
for (Laptop laptop : list) {
```

- Reset lại ô nhập thông tin khi đã thêm sản phẩm vào bảng:

```
this.txtProduct_id.setText("");
this.txtProduct_name.setText("");
this.txtProduct_price.setText("");
this.txtProduct_total.setText("");
this.txtLaptop_type.setText("");
this.txtLaptop_ram.setText("");
```

2.4.6 Sửa sản phẩm

Laptop Managerment System

Information

Product ID: 1

Product Name: Asus FTX 15

Product Price: 312.0

Product Total: 1

Laptop Type: Laptop Gaming

Laptop's Ram: 8

Function

Sort: Product_id Asc Sort

+ Add Edit Delete Show list

Q Input search Product_id Search

Product ID	Product Name	Product Price (\$)	Product Total	Laptop Type	Laptop's Ram
1	Asus FTX 15	312.0	1	Laptop Gami...	8
89	Dell XPS 13	617.0	47	Laptop Gami...	8
30	Razer Blade ...	882.0	68	Ultrabook	16
67	Acer nitro 5	608.0	8	Laptop Gami...	16
46	Lenovo Thin...	764.0	25	Student Lapt...	16
54	Acer nitro 5	383.0	28	Student Lapt...	16
90	Microsoft Su...	797.0	26	Budget Lapt...	16
55	Microsoft Su...	442.0	97	Ultrabook	16
41	Lenovo Thin...	508.0	63	Budget Lapt...	8
52	Lenovo Thin...	780.0	63	Ultrabook	16
98	Razer Blade ...	740.0	99	Student Lapt...	16
69	Dell XPS 13	613.0	0	Student Lapt...	16
25	Microsoft Su...	766.0	26	Ultrabook	16
24	Acer nitro 5	486.0	31	Laptop Gami...	8
32	Dell XPS 13	581.0	64	Laptop Gami...	8
8	Dell XPS 13	525.0	56	Student Lapt...	8
47	Dell XPS 13	881.0	11	Ultrabook	8
63	Dell XPS 13	603.0	78	Laptop Gami...	8

Hình 2. 10 Màn hình sửa sản phẩm

- Màn hình giao diện chức năng sửa sản phẩm gồm: Trường nhập dữ liệu mã sản phẩm, tên sản phẩm, giá sản phẩm, số lượng sản phẩm, loại laptop, số RAM của laptop, bảng thông tin của các sản phẩm, button “Edit” để sửa sản phẩm.

- Cập nhật lại thông tin sản phẩm trong danh sách và trong bảng :

```
// set lại thông tin cho sản phẩm đã sửa
this.list.set(selectedRow, editedLaptop);

// set lại dữ liệu cho các dòng và cột tương ứng <dtm.setValueAt("thong tin", số
// dòng, số cột)>
dtm.setValueAt(Product_id, selectedRow, 0);
dtm.setValueAt(Product_name, selectedRow, 1);
dtm.setValueAt(Product_price, selectedRow, 2);
dtm.setValueAt(Product_total, selectedRow, 3);
dtm.setValueAt(Laptop_type, selectedRow, 4);
dtm.setValueAt(Laptop_ram, selectedRow, 5);
```

- Không được để trống dữ liệu:

```
if (Product_id.isEmpty() || Product_name.isEmpty() || Product_price.isEmpty()
    || Product_total.isEmpty() || Laptop_type.isEmpty() || Laptop_ram.isEmpty()) {
    JOptionPane.showMessageDialog(contentPane, "There is a cell that has not been entered!"
        , "Error", JOptionPane.ERROR_MESSAGE);
    return;
}
```

- Sử dụng tập hợp List:

```
public List<Laptop> list = new ArrayList<Laptop>();
```

Lí do sử dụng:

- 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.

Mục đích sử dụng tập hợp List trong đoạn mã:

- Chứa danh sách các sản phẩm: List<Product> được sử dụng để chứa danh sách các đối tượng Product, mỗi đối tượng Product đại diện cho một sản phẩm với các thuộc tính như tên sản phẩm, giá cả, mô tả, v.v.

Các phép toán đã sử dụng:

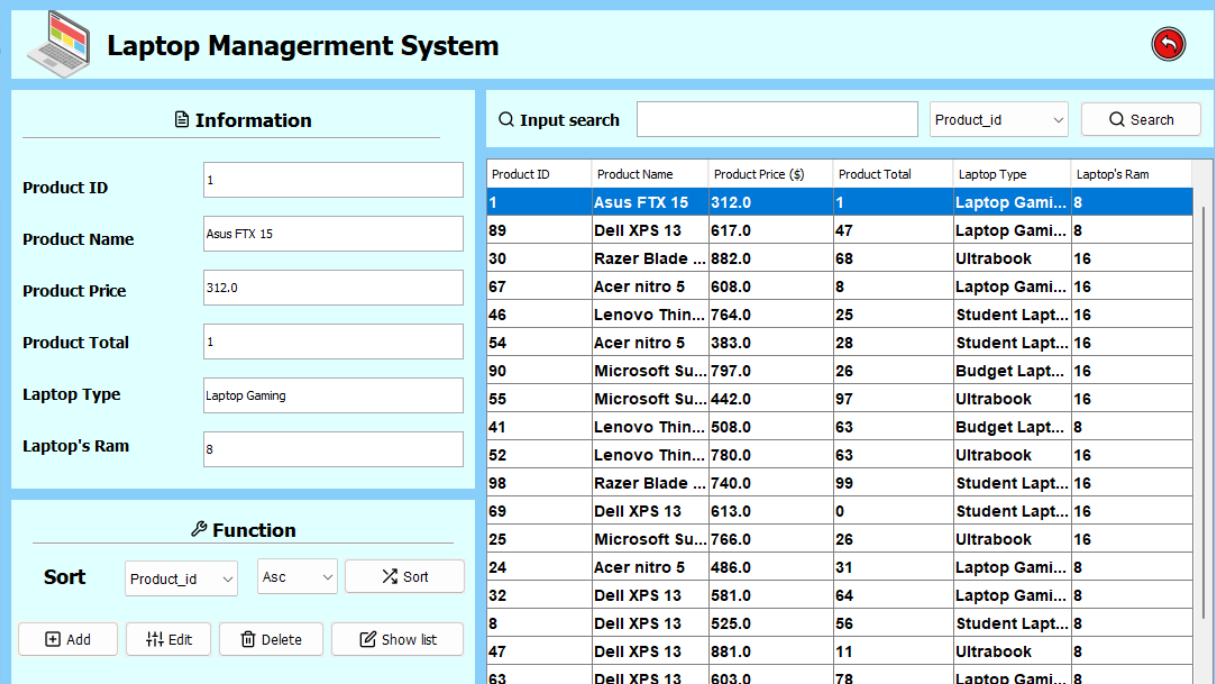
- Duyệt qua danh sách: sử dụng vòng lặp for-each

```
for (Laptop laptop : list) {
```

- Không để mã sản phẩm mới sửa lại bị trùng với mã sản phẩm đã tồn tại:

```
if (selectedRow != -1) {
    for (Laptop laptop : this.list) {
        count++;
        if (laptop.getProduct_id() == pi && count != selectedRow) {
            JOptionPane.showMessageDialog(contentPane, "This id already exists in 1 other place",
                "Error", JOptionPane.ERROR_MESSAGE);
            // gán cờ nếu có id trùng thì hiển thị thông báo lỗi và ko thêm sản phẩm mới vào
            // danh sách
            flag = false;
            return;
        }
    }
}
```

2.4.7 Xóa sản phẩm



Product ID	Product Name	Product Price (\$)	Product Total	Laptop Type	Laptop's Ram
1	Asus FTX 15	312.0	1	Laptop Gami...	8
89	Dell XPS 13	617.0	47	Laptop Gami...	8
30	Razer Blade ...	882.0	68	Ultrabook	16
67	Acer nitro 5	608.0	8	Laptop Gami...	16
46	Lenovo Thin...	764.0	25	Student Lapt...	16
54	Acer nitro 5	383.0	28	Student Lapt...	16
90	Microsoft Su...	797.0	26	Budget Lapt...	16
55	Microsoft Su...	442.0	97	Ultrabook	16
41	Lenovo Thin...	508.0	63	Budget Lapt...	8
52	Lenovo Thin...	780.0	63	Ultrabook	16
98	Razer Blade ...	740.0	99	Student Lapt...	16
69	Dell XPS 13	613.0	0	Student Lapt...	16
25	Microsoft Su...	766.0	26	Ultrabook	16
24	Acer nitro 5	486.0	31	Laptop Gami...	8
32	Dell XPS 13	581.0	64	Laptop Gami...	8
8	Dell XPS 13	525.0	56	Student Lapt...	8
47	Dell XPS 13	881.0	11	Ultrabook	8
63	Dell XPS 13	603.0	78	Laptop Gami...	8

Hình 2. 11 Màn hình xóa sản phẩm

- Màn hình giao diện chức năng xóa sản phẩm bao gồm: Trường nhập dữ liệu mã sản phẩm, tên sản phẩm, giá sản phẩm, số lượng sản phẩm, loại laptop, số

RAM của laptop, bảng thông tin của các sản phẩm, button “Delete” để xóa dữ liệu

- Kiểm tra đã chọn sản phẩm để xóa hay chưa nếu chọn thì sẽ xóa ở bảng và trong danh sách

```
if (selectedRow != -1) {
    int c = JOptionPane.showConfirmDialog(contentPane, "Do you want to delete this product?");
    if (c == JOptionPane.OK_OPTION) {
        Laptop laptopRemove = this.list.get(selectedRow);
        dtm.removeRow(selectedRow); // Xóa dòng này khỏi bảng
        this.list.remove(laptopRemove); // xóa laptop khỏi danh sách
    } else {
        return;
    }
} else {
    JOptionPane.showMessageDialog(contentPane, "You haven't selected a product to delete!");
}
```

- Thông báo sửa thành công và cập nhật lại thông tin ở bảng :

```
JOptionPane.showMessageDialog(contentPane, "Successfully deleted");
dtm.fireTableDataChanged(); // cập nhật lại thông tin của bảng
```

2.4.8 Tìm kiếm

The screenshot displays the 'Laptop Management System' interface. On the left, there is an 'Information' panel with input fields for Product ID (1), Product Name (Asus FTX 15), Product Price (312.0), Product Total (1), Laptop Type (Laptop Gaming), and Laptop's Ram (8). Below this is a 'Function' panel with a 'Sort' dropdown set to 'Product_id' and 'Asc', and buttons for 'Add', 'Edit', 'Delete', and 'Show list'. On the right, there is a search bar with the text 'Input search' and a 'Product_id' dropdown. Below the search bar is a table with 6 columns: Product ID, Product Name, Product Price (\$), Product Total, Laptop Type, and Laptop's Ram. The table contains 20 rows of data, including products like Asus FTX 15, Dell XPS 13, Razer Blade, Acer nitro 5, Lenovo Thin, Microsoft Su, and Dell XPS 13.

Product ID	Product Name	Product Price (\$)	Product Total	Laptop Type	Laptop's Ram
1	Asus FTX 15	312.0	1	Laptop Gami...	8
89	Dell XPS 13	617.0	47	Laptop Gami...	8
30	Razer Blade ...	882.0	68	Ultrabook	16
67	Acer nitro 5	608.0	8	Laptop Gami...	16
46	Lenovo Thin...	764.0	25	Student Lapt...	16
54	Acer nitro 5	383.0	28	Student Lapt...	16
90	Microsoft Su...	797.0	26	Budget Lapt...	16
55	Microsoft Su...	442.0	97	Ultrabook	16
41	Lenovo Thin...	508.0	63	Budget Lapt...	8
52	Lenovo Thin...	780.0	63	Ultrabook	16
98	Razer Blade ...	740.0	99	Student Lapt...	16
69	Dell XPS 13	613.0	0	Student Lapt...	16
25	Microsoft Su...	766.0	26	Ultrabook	16
24	Acer nitro 5	486.0	31	Laptop Gami...	8
32	Dell XPS 13	581.0	64	Laptop Gami...	8
8	Dell XPS 13	525.0	56	Student Lapt...	8
47	Dell XPS 13	881.0	11	Ultrabook	8
63	Dell XPS 13	603.0	78	Laptop Gami...	8

Hình 2. 12 Màn hình tìm kiếm sản phẩm

- Màn hình giao diện chức năng tìm kiếm bao gồm: Trường nhập dữ liệu mã sản phẩm, tên sản phẩm, giá sản phẩm, số lượng sản phẩm, loại laptop, thông tin tìm kiếm của sản phẩm số RAM của laptop, bảng thông tin của các sản phẩm và nút button “Search” để tìm kiếm, comboBox để lựa chọn thông tin muốn tìm kiếm.

- Các kiểu tìm kiếm

```
* Phương thức danh sách tìm kiếm theo Laptop_name[]
public List<Laptop> searchByLaptopType(String name) {}

* Phương thức danh sách tìm kiếm theo Laptop_ram[]
public List<Laptop> searchByLaptopRam(int ram) {}

* Phương thức danh sách tìm kiếm theo Product_id[]
public List<Laptop> searchByProductId(int id) {}

* Phương thức danh sách tìm kiếm theo Product_name[]
public List<Laptop> searchByProductName(String name) {}

* Phương thức danh sách tìm kiếm theo Product_total[]
public List<Laptop> searchByProductTotal(int total) {}

* Phương thức danh sách tìm kiếm theo Product_price[]
public List<Laptop> searchByProductPrice(double price) {}
```

- Lấy thông tin từ các comboBox và trường nhập thông tin tìm kiếm

```
String selectedItem = (String) comboBox_Search.getSelectedItem();
String searchText = txtSearch.getText();
```

- Xóa thông tin ở bảng

```
public void clearTable() {
    DefaultTableModel dtm = (DefaultTableModel) table.getModel();
    dtm.setRowCount(0);
}
```

- Phương thức tìm kiếm


```

public void search() {
    String selectedItem = (String) comboBox_Search.getSelectedItem();
    String searchText = txtSearch.getText();
    DefaultTableModel dtm = (DefaultTableModel) table.getModel();
    List<Laptop> results = new ArrayList<Laptop>();
    switch (selectedItem) {
        case "Product ID":
            results = searchByProductId(Integer.parseInt(searchText));
            break;
        case "Product Name":
            results = searchByProductName(searchText);
            break;
        case "Product Price":
            results = searchByProductPrice(Double.parseDouble(searchText));
            break;
        case "Product Total":
            results = searchByProductTotal(Integer.parseInt(searchText));
            break;
        case "Laptop Type":
            results = searchByLaptoptype(searchText);
            break;
        case "Laptop's Ram":
            results = searchByLaptopRam(Integer.parseInt(searchText));
            break;
    }
    addData(results);
}

```


2.4.9 Sắp xếp

The screenshot displays the 'Laptop Management System' interface. On the left, there is an 'Information' section with input fields for Product ID (1), Product Name (Asus FTX 15), Product Price (312.0), Product Total (1), Laptop Type (Laptop Gaming), and Laptop's Ram (8). Below this is a 'Function' section with a 'Sort' button, a dropdown menu set to 'Product_id', a dropdown menu set to 'Asc', and a 'Sort' button with a refresh icon. At the bottom of the Function section are buttons for '+ Add', 'Edit', 'Delete', and 'Show list'. On the right, there is a search bar with the text 'Q Input search', a dropdown menu set to 'Product_id', and a 'Search' button. Below the search bar is a table with the following data:

Product ID	Product Name	Product Price (\$)	Product Total	Laptop Type	Laptop's Ram
1	Asus FTX 15	312.0	1	Laptop Gami...	8
89	Dell XPS 13	617.0	47	Laptop Gami...	8
30	Razer Blade ...	882.0	68	Ultrabook	16
67	Acer nitro 5	608.0	8	Laptop Gami...	16
46	Lenovo Thin...	764.0	25	Student Lapt...	16
54	Acer nitro 5	383.0	28	Student Lapt...	16
90	Microsoft Su...	797.0	26	Budget Lapt...	16
55	Microsoft Su...	442.0	97	Ultrabook	16
41	Lenovo Thin...	508.0	63	Budget Lapt...	8
52	Lenovo Thin...	780.0	63	Ultrabook	16
98	Razer Blade ...	740.0	99	Student Lapt...	16
69	Dell XPS 13	613.0	0	Student Lapt...	16
25	Microsoft Su...	766.0	26	Ultrabook	16
24	Acer nitro 5	486.0	31	Laptop Gami...	8
32	Dell XPS 13	581.0	64	Laptop Gami...	8
8	Dell XPS 13	525.0	56	Student Lapt...	8
47	Dell XPS 13	881.0	11	Ultrabook	8
63	Dell XPS 13	603.0	78	Laptop Gami...	8

Hình 2. 13 Màn hình sắp xếp sản phẩm

- Màn hình giao diện chức năng sắp xếp bao gồm: Trường nhập dữ liệu mã sản phẩm, tên sản phẩm, giá sản phẩm, số lượng sản phẩm, loại laptop, số RAM của laptop, bảng thông tin của các sản phẩm và nút button “Sort” để sắp xếp, 1 comboBox để lựa chọn thông tin muốn sắp xếp và 1 comboBox để chọn kiểu sắp xếp tăng dần hay giảm dần.
- Các phương thức sắp xếp

```
* Phương thức sắp xếp theo Loại Laptop[]  
public List<Laptop> sortedByLaptopType(List<Laptop> list1, boolean isSort) {  
  
* Phương thức sắp xếp theo RAM laptop[]  
public List<Laptop> sortedByLaptopRam(List<Laptop> list1, boolean isSort) {  
  
* Phương thức sắp xếp theo mã sản phẩm[]  
public List<Laptop> sortedByProductId(List<Laptop> list1, boolean isSort) {  
  
* Phương thức sắp xếp theo tên sản phẩm[]  
public List<Laptop> sortedByProductName(List<Laptop> list1, boolean isSort) {  
  
* Phương thức sắp xếp theo giá sản phẩm[]  
public List<Laptop> sortedByProductPrice(List<Laptop> list1, boolean isSort) {  
  
* Phương thức sắp xếp theo số lượng sản phẩm[]  
public List<Laptop> sortedByProductTotal(List<Laptop> list1, boolean isSort) {
```

- Lấy thông tin từ các comboBox

```
String selectedItem = (String) comboBox_Sort.getSelectedItemAt();
String selectedItem1 = (String) comboBox_Sorted_by.getSelectedItemAt();
```

- Xóa thông tin ở bảng

```
public void clearTable() {
    DefaultTableModel dtm = (DefaultTableModel) table.getModel();
    dtm.setRowCount(0);
}
```

- Hiển thị thông tin từ danh sách ra bảng

```
public void addData(List<Laptop> results) {
    clearTable();
    DefaultTableModel dtm = (DefaultTableModel) table.getModel();
    for (Laptop laptop : results) {
        Vector<Object> row = new Vector<Object>();
        row.add(laptop.getProduct_id());
        row.add(laptop.getProduct_name());
        row.add(laptop.getProduct_price());
        row.add(laptop.getProduct_total());
        row.add(laptop.getLaptop_type());
        row.add(laptop.getLaptop_ram());
        dtm.addRow(row);
    }
    dtm.fireTableDataChanged();// cập nhật lại thông tin của bảng
}
```

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 Hệ thống Quản lý sản phẩm điện tử Cho Công Ty YP”, chúng em đã được học và áp dụng những kiến thức như là:

- Kiến thức về Java Swing: Để xây dựng giao diện người dùng cho chương trình, nhóm 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 mảng liên kết, 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 mang tính thẩm mỹ, chúng em đã tìm hiểu cách thiết kế giao diện bằng việc sắp xếp khéo léo vị trí của các thành phần như nút bấm, trường nhập dữ liệu và bảng hiển thị.
- 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 Hệ thống Quản lý sản phẩm điện tử cho công ty YP”, 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.

- Giao diện được thiết kế phù hợp, thân thiện với người dùng.
- Áp dụng được kỹ năng phân tích hệ thống, kỹ năng quản lý dự án, kiến thức liên quan đến thiết kế và kỹ năng lập trình.

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 Hệ thống Quản lý sản phẩm điện tử Cho Công Ty YP”, chúng em nhận thấy có một số hướng có thể được phát triển:

- Cải thiện giao diện người dùng: Dù cho giao diện hiện tại đã đáp ứng được cơ bản yêu cầu về mặt thiết kế, nhưng vẫn còn không gian để cải tiến. 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 thông báo có thể được thêm vào chương trình nhằm hiển thị thông tin của một số thay đổi trong quá trình làm việc.
- 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.
- Chuyển hóa thành ứng dụng di động: Không chỉ giới hạn trên phiên bản Windows, có một hướng phát triển khác đó là phát triển trên nền tảng Mobile. Tích hợp thêm khả năng mua hàng trực tuyến dành cho khách hàng để mở rộng doanh thu.

TÀI LIỆU THAM KHẢO

- [1]. IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
- [2] Cay S. Horstmann; Core Java - Volume II - Advanced Features; NXB Prentice Hall, 2017.
- [3] H. M. Deitel; P. J. Deitel; S. E. Santry, Advanced Java 2 Platform, Prentice Hall, 2001.
- [4] Đoàn Văn Ban, Lập trình Java nâng cao, NXB Khoa học và Kỹ thuật, Hà Nội 201.