

|  |
| --- |
| TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC |
| Logo  Description automatically generated**Khoa Công Nghệ Thông Tin** |

**Lập trình JAVA**

PRJ – REPORT

**BÁO CÁO ĐỒ ÁN JAVA**

**Sinh viên:**

* Nguyễn Văn Long -20211TT0108

**Giảng viên hướng dẫn:**

* Nguyễn Thị Hồng Mỹ

**TP.HCM, tháng 12 năm 2022**

**Mục lục**

[1. Giới thiệu đồ án 3](#_Toc122991487)

[2. Mô tả đồ án 3](#_Toc122991488)

[2.1. Mô tả 3](#_Toc122991489)

[2.2. Sơ đồ các lớp 3](#_Toc122991490)

[2.3. Mô tả phác thảo đồ án 5](#_Toc122991491)

[3. Thực hiện đồ án 7](#_Toc122991492)

[3.1. Các thư viện sử dụng 7](#_Toc122991493)

[3.2. Cấu trúc package đồ án 7](#_Toc122991494)

[3.3. Hoàn thiện sản phẩm 8](#_Toc122991495)

[3.3.1. Chức năng đăng nhập hệ thống 8](#_Toc122991496)

[3.3.2. Chức năng quản lý sản phẩm 11](#_Toc122991497)

[3.3.3. Chức năng quản lý loại 22](#_Toc122991498)

[3.3.4. Chức năng quản lý kho 22](#_Toc122991499)

[3.3.5. Chức năng đăng xuất 23](#_Toc122991500)

[3.3.6. Các đoạn code nổi bật khác 24](#_Toc122991501)

[4. Tài liệu tham khảo 25](#_Toc122991502)

# Giới thiệu đồ án

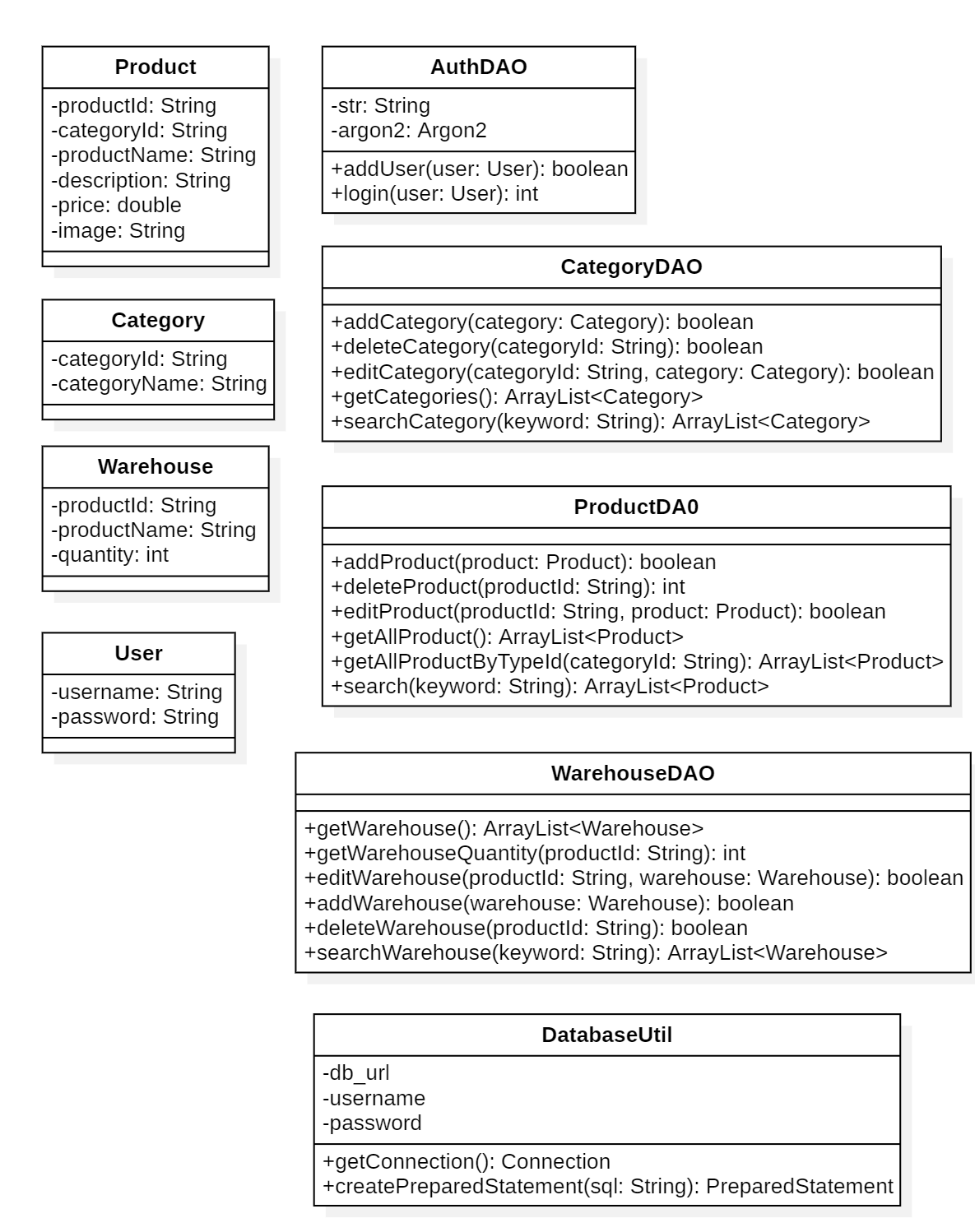
* Xây dựng ứng dụng quản lý hàng hoá sử dụng JAVA Swing với các tính năng cơ bản như thêm, xoá, sửa, đăng nhập, sắp xếp theo 1 tiêu chí, sắp xếp theo 2 tiêu chí, lấy sản phẩm theo loại, tìm kiếm sản phẩm theo mô hình MVC.

# Mô tả đồ án

## Mô tả

* Ứng dụng sử dụng java swing làm giao diện người dùng
* Ứng dụng sử dụng mysql làm hệ quản trị cơ bản sử dụng

## Sơ đồ các lớp



Hình ảnh 1. Sơ đồ lớp hình 1

Text

Description automatically generated

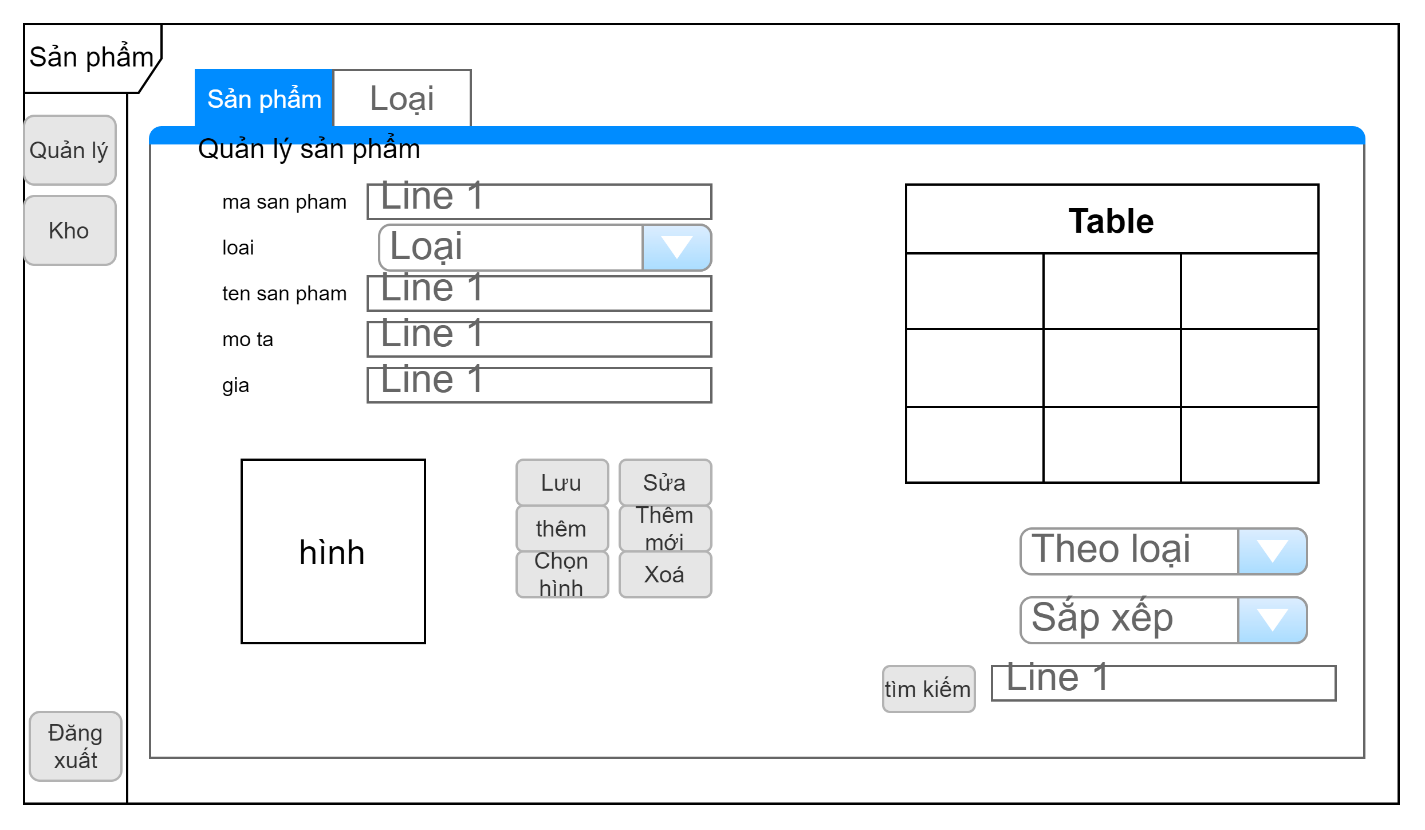
Hình ảnh 2. Sơ đồ các lớp hình 2

## Mô tả phác thảo đồ án

Graphical user interface, text, application

Description automatically generated

Hình ảnh 3. Phác thảo giao diện đăng nhập



Hình ảnh 4. Phác thảo giao diện quản lý sản phẩm

Graphical user interface

Description automatically generated

Hình ảnh 5. Phác thảo giao diện quản lý loại sản phẩm

Graphical user interface

Description automatically generated

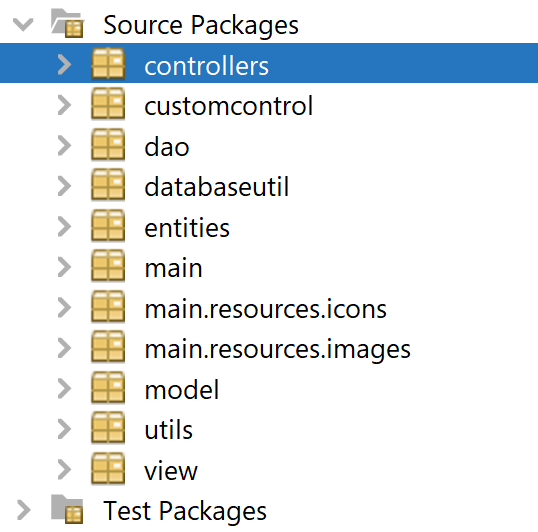
Hình ảnh 6. Phác thảo màn hình quản lý kho

# Thực hiện đồ án

## Các thư viện sử dụng

* **mysql-connector-java-8.0.28.jar**: thư viện hỗ trợ kết nối cơ sở dữ liệu mysql.
* **flatlaf-3.0.jar**: thư viện hỗ trợ theme cho ứng dụng.
* **flatlaf-extras-3.0.jar**: thư viện hỗ trợ theme cho ứng dụng.
* **argon2-jvm-2.4.jar**: thư viện hỗ trợ mã hoá mật khẩu và kiểm tra mật khẩu.
* **jna-4.5.0.jar**: thư viện đi kèm theo argon2-jvm-2.4.jar.
* **svgSalamander-1.1.3.jar**: thư viện hỗ trợ hiển thị file svg (đi kèm theo flatlaf-extras-3.0.jar).

## Cấu trúc package đồ án



Hình ảnh 7. Cấu trúc package đồ án

* Controller: chứa các controller con và là nơi code xử lý chính của ứng dụng.
* Customcontrol: chứa custom một button cho việc hiển thị kèm icon từ file svg là thanh sidebar cho ứng dụng.
* Dao: chứa các code xử lý giao tiếp trực tiếp với cơ sở dữ liệu.
* Databaseutil: chứ lớp kết nối database, lớp này được các lớp trong “dao” kế thừa.
* Entities: chứa các lớp chung cơ bản.
* Main: Chứa hàm main chạy chính của ứng dụng.
* Model: Chứa các lớp model con giao tiếp ở giữa lớp “giao diện” và lớp “dao”.
* Utils: Chứa các lớp tiện ích tự định nghĩa.
* View: chứa các lớp tạo giao diện.

## Hoàn thiện sản phẩm

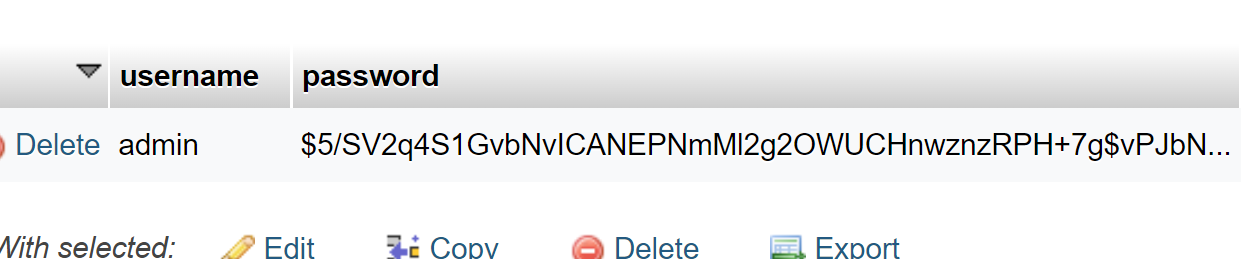
### Chức năng đăng nhập hệ thống

Graphical user interface, application

Description automatically generated

Hình ảnh 8. Giao diện đăng nhập hệ thống

* Giao diện đăng nhập hệ thống mở lên bắt đầu khởi động ứng dụng.
* Tài khoản và mật khẩu mặc định là “admin”, “admin”.
* Tài khoản và mật khẩu được lưu ở cơ sở dữ liệu và mật khẩu được mã hoá.



Hình ảnh 9. Tài khoản và mật khẩu lưu dưới cơ sở dữ liệu

1. **private** **static** String str = "$argon2id$v=19$m=15360,t=2,p=1";
2. **private** **static** Argon2 argon2 = Argon2Factory.create(Argon2Factory.Argon2Types.ARGON2id, 32, 64);
4. **public** **static** boolean addUser(User user) throws Exception {
5. **try** {
7. String sql = "INSERT INTO tbl\_user VALUE (?,?)";
9. PreparedStatement ps = createPreparedStatement(sql);
11. String hashString = argon2.hash(2, 15 \* 1024,user.getPassword().toCharArray());
12. String hassPassword = hashString.**replace**(str, "");
14. ps.setString(1, user.getUsername());
15. ps.setString(2, hassPassword);
17. int kq = ps.executeUpdate();
19. System.out.println("Them thanh cong");
21. **return** (kq == 1);
22. } **catch** (Exception e) {
23. System.out.println("Loi them du lieu " + e);
24. }
25. **return** **false**;
26. }

* Đây là đoạn code mã hoá mật khẩu khi thêm một tài khoản mới

1. **public** **static** int login(User user) throws Exception {
2. **try** {
3. String sql = "SELECT \* FROM tbl\_user WHERE username = ?";
4. PreparedStatement ps = createPreparedStatement(sql);
5. ps.setString(1, user.getUsername());
6. ResultSet data = ps.executeQuery();
7. **while** (data.next()) {
8. String hassPassword = data.getString(2);
10. **if** (!hassPassword.equals("")) {
12. boolean validPassword = argon2.verify(str + hassPassword, user.getPassword().toCharArray());
13. **if** (validPassword) {
14. System.out.println("Dang nhap thanh cong");
15. **return** 1;
16. } **else** {
17. System.out.println("dang nhap that bai, mat khau khong dung");
18. **return** 2;
19. }
20. }
21. }
22. **return** 3; // tai khoan khong ton tai
24. } **catch** (Exception e) {
25. System.out.println("Dang nhap that bai");
26. **return** 4;
27. }
28. }

* Đây là đoạn code check mật khẩu khi thực hiện đăng nhập

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình ảnh 10. Đăng nhập khi không bật cơ sở dữ liệu

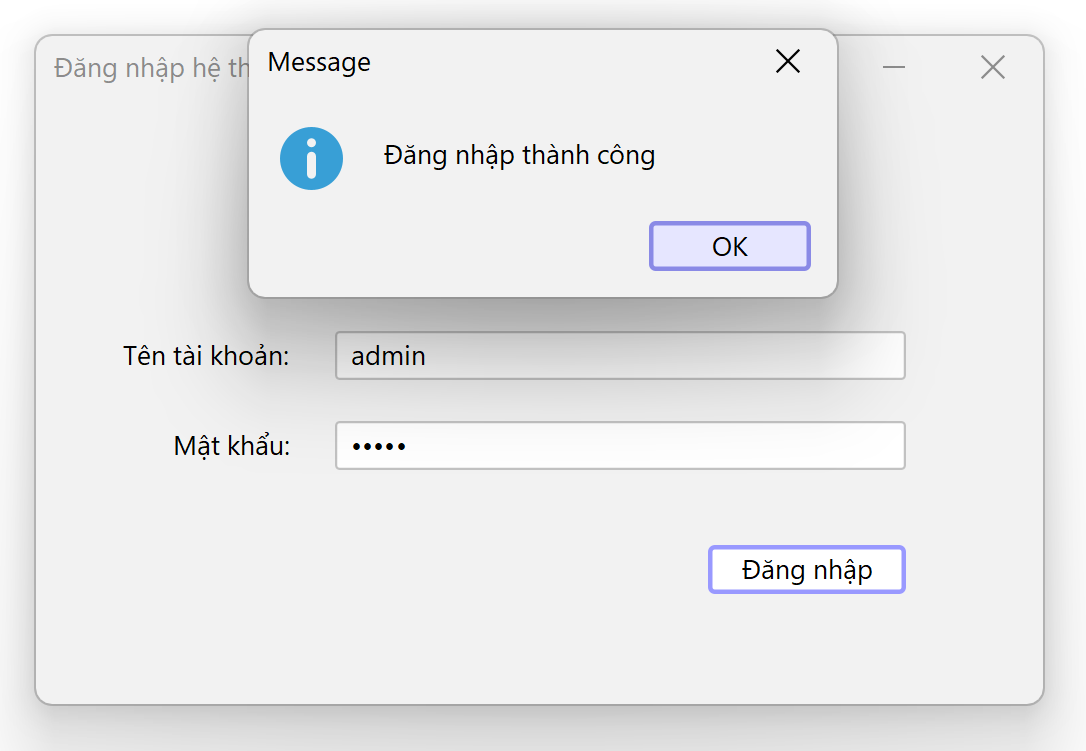
* Khi không bật cơ sở dữ liệu mà thực hiện đăng nhập sẽ hiên thông báo “Đăng nhập thất bại, không thể kết nối với máy chủ”.

Graphical user interface, text, application

Description automatically generated

Hình ảnh 11. Đăng nhập với tài khoản không tồn tại

* Khi đăng nhập với tài khoản không tồn tại sẽ hiển thị thông báo “Đăng nhập thất bại, tài khoản không tồn tại”.



Hình ảnh 12. Đăng nhập thành công

* Khi nhập nhập đúng tài khoản và mật khẩu sẽ hiển thị thông báo “Đăng nhập thành công” và sau đó sẽ hiển thị lên giao diện chính của ứng dụng.\

### Chức năng quản lý sản phẩm

Graphical user interface, application

Description automatically generated

Hình ảnh 13. Giao diện chính màn hình quản lý sản phẩm

* Mặc định khi hiện lên đã chọn vào mục quản lý
* Nút quản lý bên thanh sidebar được active và chuyển sang màu tím

Graphical user interface, application

Description automatically generated

Hình ảnh 14. Hiển thông tin chi tiết sản phẩm

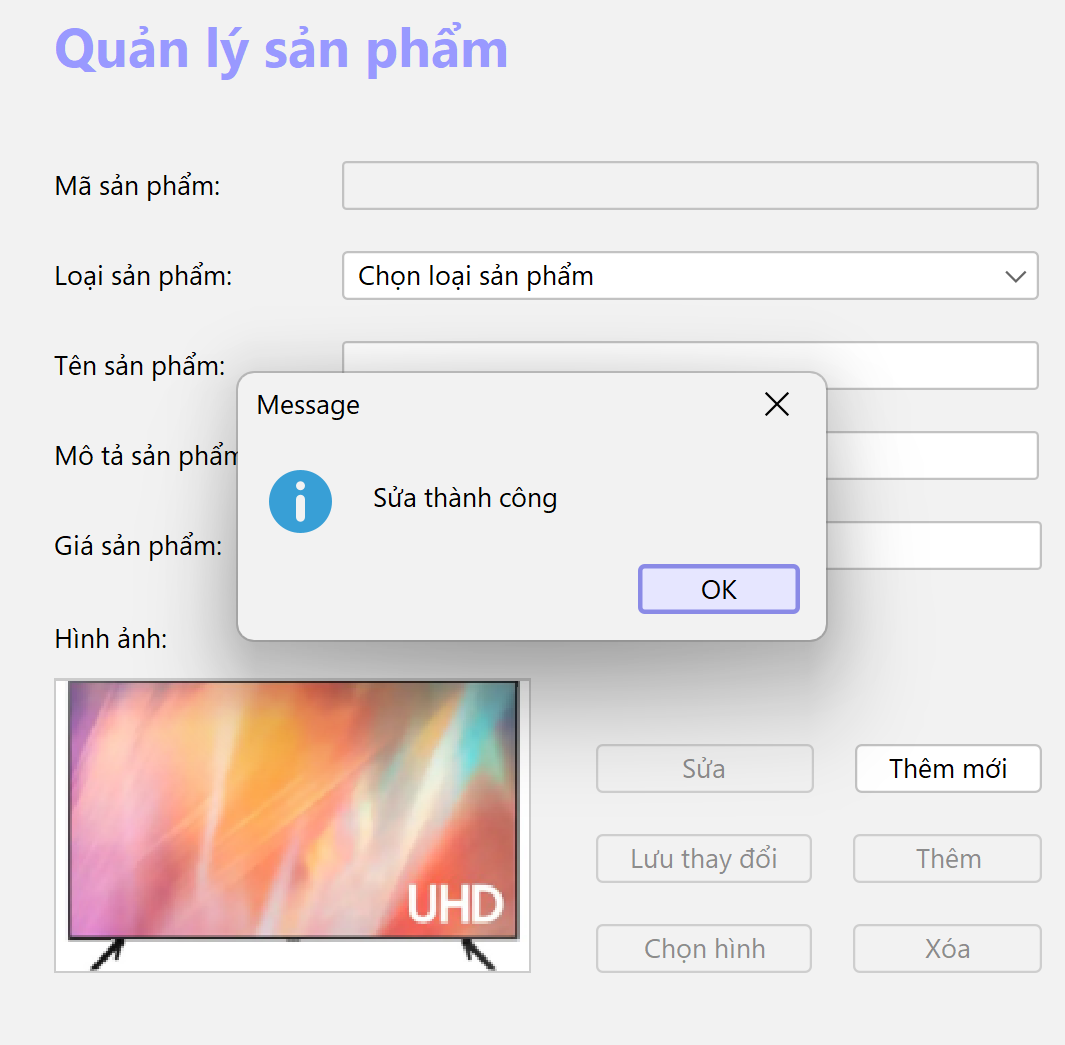
* Khi ấn vào danh sách sản phẩm ở bảng sản phẩm bên phải sẽ hiển thị thông tin sản phẩm chi tiết vào các trường bên trái, và hiển thị hình ảnh tương ứng vào ô chữ nhật bên dưới.

Graphical user interface, application

Description automatically generated

Hình ảnh 15. Sửa sản phẩm

* Khi chọn vào nút sửa thì các trường phía trên thay đổi trạng thái cho phép chỉnh sửa nội dung.
* Trường loại sản phẩm được ẩn và được chuyển thành combox với danh sách các loại sản phẩm.
* Nút xoá ẩn và chuyển thành nút xoá hình ảnh.
* Trường mã sản phẩm chỉ cho phép xem và không được sửa.



Hình ảnh 16. Thông báo sửa thành công

* Khi ấn nút lưu thay đổi thì dữ liệu được thay đổi thành công và hiện thông báo “Sửa thành công” và các nút trở về trạng thái ban đầu, đồng thời lữ liệu trên table được cập nhật lại.
* Sửa chỉ được thực hiện thành công khi chọn đầy đủ các trường, trừ trường hình ảnh có thể có hoặc không.

Graphical user interface, application

Description automatically generated

Hình ảnh 17. Thêm sản phẩm mới

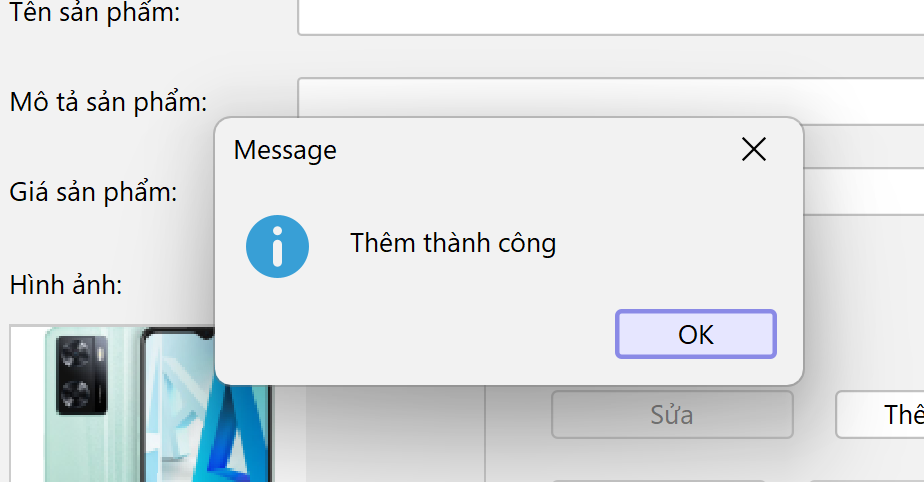
* Khi bấm vào nút thêm mới thì các trường dữ liệu trả về dữ liệu trống
* Các nút chuyển về trạng thái thêm mới

Graphical user interface, text, application

Description automatically generated

Hình ảnh 18. Hộp thoại chọn hình

* Khi bấm vào nút chọn hình thì hộp thoại chọn hình sẽ hiện lên mặc định với định dạng là hình ảnh.
* Hình ảnh sẽ được lưu vào thư mục hình ảnh trong đồ án khi được thêm thành công.
* Tên hình ảnh là định dạng “long” của ngày giờ hiện tại.
* String extension = ""; // đuôi của hình ảnh
* **if** (fileSelect != **null**) {
* int i = fileSelect.getName().lastIndexOf('.');
* **if** (i > 0) {
* extension = fileSelect.getName().toString().substring(i + 1);
* }
* }
* // dùng thời gian để làm tên hình ảnh cho đẹp và để không trùng nhau
* String newImageName = (fileSelect != **null**) ? (**new** Date().getTime() + "." + extension) : "";
* Đây là đoạn code tạo tên hình ảnh bằng ngày giờ với “fileSelect” là hình ảnh trong được chọn từ hộp thoại.



Hình ảnh 19. Thông báo thêm thành công

* Sau khi thêm thành công sẽ hiện thông báo thêm thành công và hình ảnh được copy và thư mục hình ảnh
* **public** **static** void copyFile(String source, String dest) throws IOException {
* FileInputStream inputStream = **null**;
* FileOutputStream outputStream = **null**;
* // kiem tra thu muc chua hinh anh co ton tai, khong co thi tao thu muc
* File directory = **new** File("src/main/resources/images/");
* **if** (!directory.exists()) {
* directory.mkdir();
* }
* **try** {
* inputStream = **new** FileInputStream(source);
* outputStream = **new** FileOutputStream(dest);
* int c;
* **while** ((c = inputStream.read()) != -1) {
* outputStream.write(c);
* }
* } **finally** {
* **if** (inputStream != **null**) {
* inputStream.close();
* }
* **if** (outputStream != **null**) {
* outputStream.close();
* }
* }
* }
* Đây là đoạn code copy hình ảnh

Graphical user interface, application

Description automatically generated

Hình ảnh 20. Xoá sản phẩm

* Khi ấn vào nút xoá sản phẩm thì sẽ kiểm tra số lượng sản phẩm trong kho nếu lớn hơn 0 thì không cho xoá, nếu bằng 0 thì sẽ hiển thị thông báo xoá thành công.

Graphical user interface

Description automatically generated with medium confidence

Hình ảnh 21. Lấy sản phẩm theo loại

* Khi chọn sản phẩm theo loại thì bảng hiển thị sản phẩm sẽ hiển thị sản phẩm theo loại tương ứng.

Graphical user interface

Description automatically generated with medium confidence

Hình ảnh 22. Sắp xếp sản phẩm

* Khi chọn và sắp xếp theo tiêu chí thì bảng sản phẩm sẽ cập nhật theo các tiêu chí sắp xếp được chọn.

Graphical user interface, text, application, email

Description automatically generated

Hình ảnh 23. Tìm kiếm sản phẩm

* Với tìm kiếm sản phẩm sẽ tìm kiếm theo tên sản phẩm và hiển thị kết quả tương ướng trên bản sản phẩm.
* Nếu bỏ dấu tích tìm kiếm đi thì bảng sản phẩm hiển thị trở lại đầy đủ như ban đầu.

### Chức năng quản lý loại

Graphical user interface, application, Word

Description automatically generated

Hình ảnh 24. Màn hình quản lý loại loại

* Các chức năng thêm, xoá, sửa tìm kiếm cũng có các thao tác và thông báo giống bên màn hình sản phẩm.

### Chức năng quản lý kho

Graphical user interface, application

Description automatically generated

Hình ảnh 25. Màn hình quản lý kho

* Khi chuyển sang màn hình quản lý kho thì trạng thái nút bên sidebar chuyển trạng thái active màu tím sang nút “Kho hàng”.
* Các chức năng sửa, tìm kiếm vào sắp xếp tương tự thao tác như màn hình sản phẩm.
* Màn hình này không có mục thêm và được tự động thêm khi thêm sản phẩm và có có số lượng mặc định là 1, khi số lượng được chỉnh về 0 và lúc xoá sản phẩm thì bên kho cũng sẽ được xoá theo.

### Chức năng đăng xuất

Graphical user interface, text, application

Description automatically generated

Hình ảnh 26. Nút đăng xuất

* Khi ấn vào nút đăng xuất ở góc dưới sidebar sẽ hiện lên thông báo xác nhận có đăng xuất hay không. Nếu chọn có thì ứng dụng sẽ thoát và hiện lên màn hình login.

Graphical user interface, text, application

Description automatically generated

Hình ảnh 27. Thông báo xác nhận đăng xuất

### Các đoạn code nổi bật khác

1. **private** **static** void setTheme() {
2. **try** {
3. FlatLightLaf.setGlobalExtraDefaults(Collections.singletonMap("@accentColor", "#9999FF")); // đổi màu chủ đạo
4. // UIManager.setLookAndFeel(theme);
5. FlatLightLaf.setup();
6. UIManager.put("TextComponent.arc", 4);
7. UIManager.put("TabbedPane.selectedBackground", "#FFF");
8. UIManager.put("Component.innerFocusWidth", 1);
9. UIManager.put("ScrollBar.thumbArc", 999);
10. UIManager.put("ScrollBar.thumbInsets", **new** Insets(2, 2, 2, 2));// theme
11. } **catch** (Exception e) {
12. }
13. }

* Đây là đoạn code thực hiện đổi theme cho ứng dụng sử dụng thư viện “FlatLight”.

1. **package** customcontrol;
3. **import** com.formdev.flatlaf.extras.FlatSVGIcon;
4. **import** java.awt.Color;
5. **import** java.util.**function**.Function;
6. **import** javax.swing.JButton;
7. **import** javax.swing.SwingConstants;
9. /\*\*
10. \*
11. \* @author long
12. \*/
13. **public** **class** FlatSvgButton **extends** JButton {
14. **private** FlatSVGIcon icon;
16. **public** void createButton(String iconPath) {
17. icon = **new** FlatSVGIcon(iconPath, 20, 20);
18. setIcon(icon);
19. setVerticalTextPosition(SwingConstants.BOTTOM);
20. setHorizontalTextPosition(SwingConstants.CENTER);
21. setBackground(**new** java.awt.Color(238, 238, 238));
22. setFont(**new** java.awt.Font("Segoe UI", 0, 10)); // NOI18N
23. setBorder(javax.swing.BorderFactory.createEmptyBorder(1, 1, 1, 1));
24. setMargin(**new** java.awt.Insets(2, 2, 3, 2));
26. }
28. **public** void setButtonText(String text) {
29. setText(text);
30. }
32. **public** void setButtonContentColor(Color color) {
33. icon.setColorFilter(**new** FlatSVGIcon.ColorFilter(**new** Function<Color, Color>() {
34. @Override
35. **public** Color apply(Color t) {
36. **return** color;
37. }
38. }));
39. setForeground(color);
40. }
42. **public** void setButtonDefaultContentColor() {
43. setForeground(**null**);
44. icon.setColorFilter(**null**);
45. }
46. }

* Đây là đoạn code tạo custom một component để làm phần nút thao tác bên sidebar kèm icon được hiển thị từ file svg ở giao diện chính.

# 4. Tài liệu tham khảo

- <https://www.formdev.com/flatlaf/>

- <https://www.youtube.com/watch?v=fuyJi9qw6Ag>

- <https://planetb.troye.io/> trang web hỗ trợ tạo các đoạn code đẹp để copy vào word