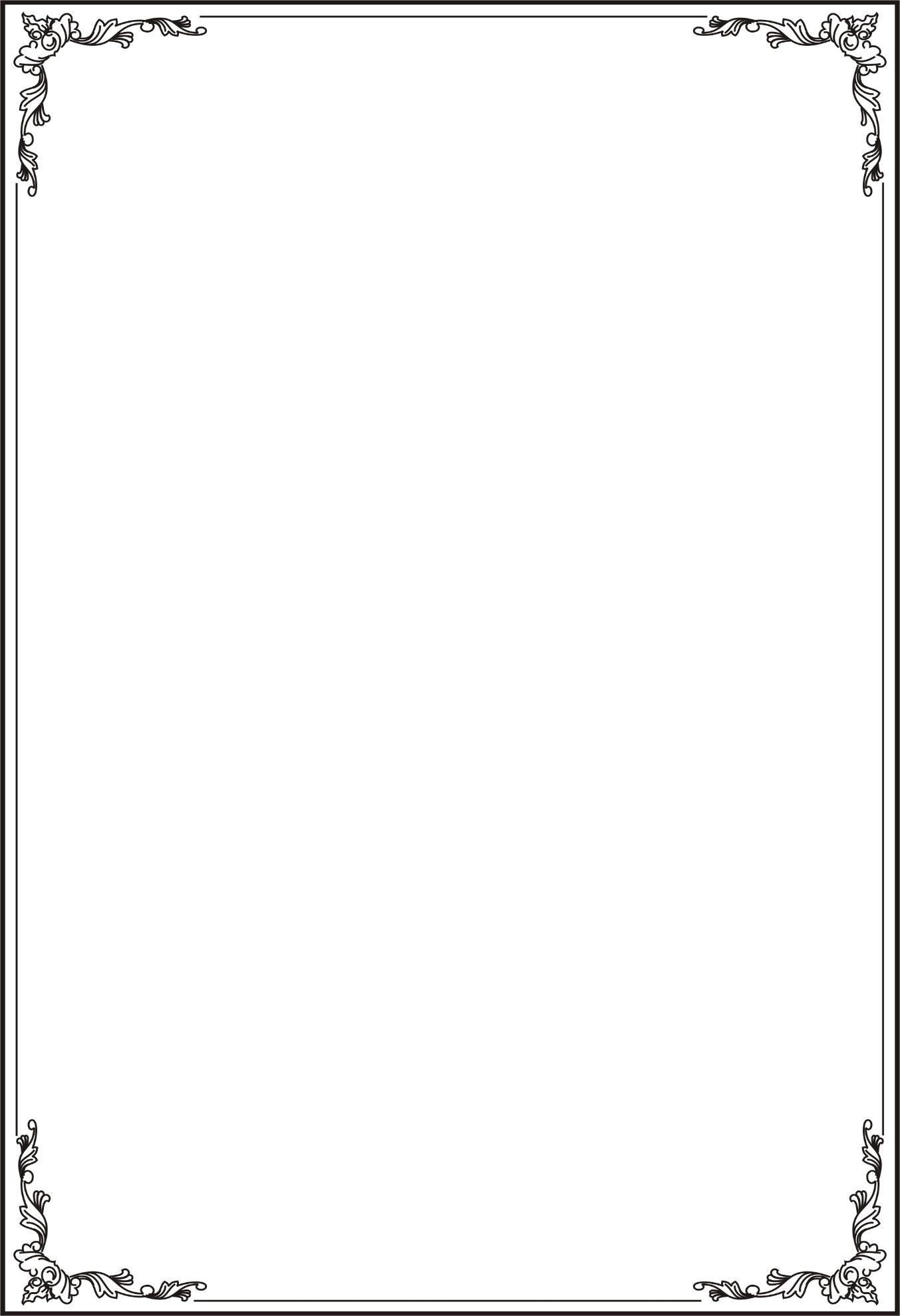
****

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

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



**BÁO CÁO BÀI TẬP LỚN THUỘC HỌC PHẦN: LẬP TRÌNH JAVA NÂNG CAO**

**ĐỀ TÀI**

**XÂY DỰNG PHẦM MỀM QUẢN LÝ KHO HÀNG BẰNG CÔNG NGHỆ JDBC, JAVA SWING**

Giáo viên hướng dẫn: Ths. Hoàng Quang Huy Nhóm – Lớp : 3 - 20232IT6020001

Thành viên : Chu Xuân Lộc– 2021601403

Phùng Công Thành– 2021601339

Nông Minh Trường– 2021602340

*Hà Nội, 2024*

# LỜI CẢM ƠN

Để hoàn thành chuyên đề báo cáo bài tập lớn, trước hết em xin gửi đến thầy ThS. Hoàng Quang Huy giáo viên hướng dẫn môn “Lập trình Java nâng cao” - người đã hết lòng giúp đỡ, dạy bảo, động viên và tạo điều kiện cho chúng em theo đuổi đề tài nghiên cứu này. Trong suốt quá trình thực hiện và tìm hiểu bài tập lớn thầy đã tận tình hướng dẫn, đưa lời khuyên bổ ích để chúng em có thể hoàn thành tốt đề tài. Và đó cũng là cơ hội cho chúng em tổng hợp và hệ thống hóa lại kiến thức đã học, đồng thời kết hợp với thực tế để nâng cao kiến thức chuyên môn.

Vì thế những đóng góp của thầy là kinh nghiệm quý báu giúp cho các thành viên trong nhóm sẽ có những dự tính sau này trong khi làm đồ án tốt nghiệp và sau khi tốt nghiệp.

Cuối cùng chúng em xin kính chúc thầy luôn luôn mạnh khỏe và ngày một thành công hơn trên con đường giảng dạy của mình.

Chúng em xin chân thành cảm ơn!.

**Nhóm 3**

Chu Xuân Lộc

Phùng Công Thành

Nông Minh Trường

# MỤC LỤC

[MỞ ĐẦU 1](#_Toc170410056)

[1. Lý do chọn đề tài 1](#_Toc170410057)

[2. Mục tiêu nghiên cứu 1](#_Toc170410058)

[3. Đối tượng, phạm vi nghiên cứu 2](#_Toc170410059)

[4. Kết quả mong muốn đạt được 3](#_Toc170410060)

[CHƯƠNG 1. GIỚI THIỆU VỀ CÔNG NGHỆ SỬ DỤNG 4](#_Toc170410061)

[1.1. Công nghệ JDBC 4](#_Toc170410062)

[1.1.1 Định nghĩa 4](#_Toc170410063)

[1.1.2 Ưu điểm của JDBC 4](#_Toc170410065)

[1.1.3 Nhược điểm của JDBC 5](#_Toc170410066)

[1.1.4 Kiến trúc JDBC 5](#_Toc170410067)

[1.1.5 Trình điều khiển JDBC 8](#_Toc170410071)

[1.1.6 Các bước để phát triển một ứng dụng JDBC 13](#_Toc170410077)

[1.1.7 Nạp trình điều khiển 14](#_Toc170410079)

[1.1.8 Xử lý kết quả và đóng kết nối cơ sở dữ liệu 15](#_Toc170410080)

[1.2. Công nghệ Java Swing 16](#_Toc170410081)

[1.2.1 Java AWT 16](#_Toc170410082)

[1.2.2 Java Swing 18](#_Toc170410083)

[1.2.3 So sánh Java swing với java awt 19](#_Toc170410085)

[1.3. Cơ sở dữ liệu MySQL 20](#_Toc170410086)

[CHƯƠNG 2. PHÂN TÍCH YÊU CẦU 24](#_Toc170410088)

[2.1 Xác định các yêu cầu cụ thể của phần mềm quản lý dự án 24](#_Toc170410089)

[2.1.1 Biểu đồ use case tổng quan 24](#_Toc170410090)

[2.1.2 Mối quan hệ giữa các use case 24](#_Toc170410092)

[2.1.3 Các yêu cầu chức năng 24](#_Toc170410093)

[2.2 Đặc tả các yêu cầu phần mềm 25](#_Toc170410094)

[2.2.1 Đặc tả use case Quản lý sản phẩm 25](#_Toc170410095)

[2.2.2 Đặc tả use case Quản lý tài khoản 26](#_Toc170410096)

[2.2.3 Đặc tả use case Quản lý phiếu nhập 28](#_Toc170410097)

[2.2.4 Đặc tả use case Thống kê kho 30](#_Toc170410098)

[2.2.5 Đặc tả use case Quản lý nhà cung cấp 31](#_Toc170410099)

[2.2.6 Đặc tả use case Quản lý phiếu xuất 33](#_Toc170410100)

[CHƯƠNG 3. THIẾT KẾ PHẦN MỀM 37](#_Toc170410101)

[3.1 Kiến trúc tổng quan của phần mềm 37](#_Toc170410102)

[3.1.1 Tổng quan kiến trúc phần mềm 37](#_Toc170410103)

[3.1.2 Mô hình kiến trúc – Mô hình MVC 39](#_Toc170410106)

[3.2 Thiết kế biểu đồ trình tự 41](#_Toc170410108)

[3.2.1 Biểu đồ trình tự use case Quản lý sản phẩm 41](#_Toc170410109)

[3.2.2 Biểu đồ trình tự use case Quản lý tài khoản 43](#_Toc170410111)

[3.2.3 Biểu đồ trình tự use case Quản lý phiếu nhập 44](#_Toc170410113)

[3.2.4 Biểu đồ trình tự use case Thống kê kho 46](#_Toc170410115)

[3.2.5 Biểu đồ trình tự use case Quản lý phiếu xuất 47](#_Toc170410117)

[3.2.6 Biểu đồ trình tự use case Quản lý nhà cung cấp 48](#_Toc170410119)

[3.3 Thiết kế cơ sở dữ liệu 49](#_Toc170410121)

[3.3.1 Biểu đồ thực thể liên kết mức vật lý 49](#_Toc170410122)

[3.3.2 Thiết kế bảng 49](#_Toc170410124)

[CHƯƠNG 4. TRIỂN KHAI VÀ KIỂM THỬ PHẦN MỀM 52](#_Toc170410132)

[4.1 Lựa chọn công nghệ triển khai 52](#_Toc170410133)

[4.1.1 Giới thiệu ngôn ngữ Java 52](#_Toc170410134)

[4.2 Lập trình và triển khai phần mềm 53](#_Toc170410135)

[CHƯƠNG 5. KẾT LUẬN LUẬN VÀ TỔNG KẾT 68](#_Toc170410166)

[5.1 Tóm tắt kết quả nghiên cứu 68](#_Toc170410167)

[5.2 Đánh giá đối với mục tiêu ban dầu 68](#_Toc170410168)

[5.3 Đề xuất hướng phát triển tương lai 68](#_Toc170410169)

[TÀI LIỆU THAM KHẢO 69](#_Toc170410170)

# DANH MỤC HÌNH ẢNH

[Hình 1. 1: Công nghệ JDBC 4](#_Toc170410199)

[Hình 1. 2: Mô hình 2 tầng 6](#_Toc170410203)

[Hình 1. 3: Mô hình 3 tầng 7](#_Toc170410204)

[Hình 1. 4: JDBC API 8](#_Toc170410205)

[Hình 1. 5: Trình điều khiển JDBC 9](#_Toc170410207)

[Hình 1. 6: Trình điều khiển loại 1 9](#_Toc170410208)

[Hình 1. 7: Trình điều khiển loại 2 10](#_Toc170410209)

[Hình 1. 8: Trình điều khiển loại 3 11](#_Toc170410210)

[Hình 1. 9: Trình điều khiển loại 4 12](#_Toc170410211)

[Hình 1. 10: Các bước để phát triển một ứng dụng JDBC 13](#_Toc170410213)

[Hình 1. 11: Hệ thống phân cấp API Java Swing 19](#_Toc170410219)

[Hình 1. 12: MySQL Workbench 23](#_Toc170410222)

[Hình 2. 1: Biểu đồ use case tổng quan 24](#_Toc170410226)

[Hình 3. 1: Kiến trúc của một ứng dụng doanh nghiệp 37](#_Toc170410239)

[Hình 3. 2: Mẫu kiến trúc phần mềm 38](#_Toc170410240)

[Hình 3. 3: Mô hình MVC 39](#_Toc170410242)

[Hình 3. 4: Biểu đồ trình tự use case Quản lý sản phẩm 42](#_Toc170410245)

[Hình 3. 5: Biểu đồ trình tự use case Quản tài khoản 43](#_Toc170410247)

[Hình 3. 6: Biểu đồ trình tự use case Quản lý phiếu nhập 45](#_Toc170410249)

[Hình 3.7: Biểu đồ trình tự use case thống kê kho 46](#_Toc170410251)

[Hình 3.8: Biểu dồ trình tự use case quản lý phiếu xuất 47](#_Toc170410253)

[Hình 3.9: Biểu dồ trình tự use case quản lý nhà cung cấp 48](#_Toc170410255)

[Hình 3.10: Biểu đồ thực thể liên kết mức vật lý 49](#_Toc170410258)

[Hình 3.11: Bảng maytinh 49](#_Toc170410260)

[Hình 3. 12: Bảng account 50](#_Toc170410261)

[Hình 3.13: Bảng chitietphieunhap 50](#_Toc170410262)

[Hình 3.14: Bảng chitietphieuxuat 50](#_Toc170410263)

[Hình 3.15: Bảng nhacungcap 51](#_Toc170410264)

[Hình 3.16: Bảng phieunhap 51](#_Toc170410265)

[Hình 3.17: Bảng phieuxuat 51](#_Toc170410266)

[Hình 4.2 Giao diện đăng nhập với tư cách Admin 54](#_Toc170410272)

[Hình 4.3 Giao diện đăng nhập với tư cách quản lí kho 54](#_Toc170410273)

[Hình 4.4 Giao diện đăng nhập với tư cách nhân viên nhập 55](#_Toc170410274)

[Hình 4.5 Giao diện đăng nhập với tư cách nhân viên xuất 55](#_Toc170410275)

[Hình 4.6 Giao diện thông tin cá nhân 56](#_Toc170410276)

[Hình 4.7 Giao diện thay đổi thông tin tài khoản 56](#_Toc170410277)

[Hình 4.8 Giao diện khi nhập mật khẩu mới không hợp lệ 57](#_Toc170410278)

[Hình 4.9 Giao diện khi nhập mật khẩu cũ sai 57](#_Toc170410279)

[Hình 4.10 Giao diện khi đổi mật khẩu thành công 58](#_Toc170410280)

[Hình 4.11 Mật khẩu được mã hóa BcryPassword và lưu vào cơ sở dữ liệu để tăng tính bảo mật 58](#_Toc170410281)

[Hình 4.12 Giao diện trang sản phẩm 58](#_Toc170410282)

[Hình 4.13 Thông tin chi tiết sản phẩm laptop 59](#_Toc170410283)

[Hình 4.14 Thông tin chi tiết sản phẩm PC 59](#_Toc170410284)

[Hình 4.15 Tìm kiếm sản phẩm 60](#_Toc170410285)

[Hình 4.16 Giao diện sửa sản phẩm 60](#_Toc170410286)

[Hình 4.17 Giao diện tạo phiếu nhập kho 61](#_Toc170410287)

[Hình 4.18 Chi tiết phiếu nhập 61](#_Toc170410288)

[Hình 4.19 Xuất file excel phiếu nhập 62](#_Toc170410289)

[Hình 4.20 Xuất file PDF phiếu nhập 62](#_Toc170410290)

[Hình 4.21 Giao diện tạo phiếu xuất kho 63](#_Toc170410291)

[Hình 4.22 Chi tiết phiếu xuất 63](#_Toc170410292)

[Hình 4.23 Xuất file excel phiếu xuất 64](#_Toc170410293)

[Hình 4.24 Xuất file PDF phiếu xuất 64](#_Toc170410294)

[Hình 4.25 Giao diện nhập thông tin hàng hóa 65](#_Toc170410295)

[Hình 4.26 Giao diện xóa thông tin hàng hóa 65](#_Toc170410296)

[Hình 4.27 Thống kê hàng hóa theo ngày 66](#_Toc170410297)

[Hình 4.28 Giao diện thêm nhân viên 66](#_Toc170410298)

[Hình 4.29 Giao diện xóa nhân viên 67](#_Toc170410299)

[Hình 4.30 Giao diện các biểu đồ 67](#_Toc170410300)

# MỞ ĐẦU

## Lý do chọn đề tài

Ngày nay, việc quản lý kho hàng đòi hỏi sự chính xác, nhanh chóng và tiện lợi. Một hệ thống quản lý kho hàng không chỉ giúp tiết kiệm thời gian mà còn mang lại trải nghiệm người dùng tốt hơn cho các doanh nghiệp. Đồng thời, việc bảo vệ thông tin sản phẩm và dữ liệu của doanh nghiệp cũng trở nên quan trọng hơn bao giờ hết, và một phần mềm có thể được thiết kế để đảm bảo tính an toàn và bảo mật cho mọi thông tin liên quan.

Ngoài ra, khả năng thống kê và tạo báo cáo tự động từ hệ thống giúp quản trị kho hàng có cái nhìn tổng quan và chi tiết về tình trạng nhập/xuất kho, giúp họ đưa ra những quyết định thông tin và phát triển chiến lược quản lý hiệu quả. Việc tích hợp các công cụ quản lý thông minh vào hệ thống cũng mở ra một thế giới mới của tiện ích cho doanh nghiệp, từ việc theo dõi tình trạng hàng hóa theo thời gian thực đến tối ưu hóa quy trình quản lý kho hàng.

Việc xây dựng một phần mềm quản lý kho hàng hiện đại sẽ mang lại nhiều lợi ích cho doanh nghiệp, giúp họ nâng cao hiệu quả hoạt động, giảm thiểu rủi ro và tăng cường khả năng cạnh tranh trên thị trường.offline

## Mục tiêu nghiên cứu

Mục tiêu của đề tài này là tạo ra một hệ thống quản lý kho hàng hiệu quả, tiện ích và linh hoạt, nhằm nâng cao hiệu suất và trải nghiệm người dùng trong quá trình quản lý và sử dụng kho hàng. Đồng thời, đề tài này cũng hướng đến việc đáp ứng các thách thức hiện đại mà các doanh nghiệp đang đối mặt, từ việc quản lý thông tin lớn đến sự đa dạng của sản phẩm và nguồn tài nguyên kho.

Một trong những mục tiêu hàng đầu là xây dựng một giao diện người dùng thân thiện, dễ sử dụng và đồng nhất trên nhiều thiết bị, từ máy tính cá nhân đến thiết bị di động. Việc này giúp tạo ra một trải nghiệm người dùng mượt mà và thuận tiện, giảm thiểu thời gian và công sức khi tìm kiếm thông tin sản phẩm, nhập/xuất kho, và tương tác với các tính năng khác của hệ thống.

Mục tiêu khác là tối ưu hóa quy trình quản lý thông tin, từ việc nhập liệu đến theo dõi tình trạng kho hàng. Hệ thống sẽ cung cấp các báo cáo và thống kê tự động, giúp quản trị kho hàng có cái nhìn tổng quan và chi tiết về hoạt động của kho. Điều này không chỉ giúp họ ra quyết định thông tin mà còn tạo ra cơ sở để quy hoạch và phát triển dịch vụ quản lý kho hàng theo hướng hiệu quả nhất.

Một mục tiêu quan trọng nữa của đề tài là tích hợp chức năng quản lý thông minh, mở rộng khả năng tiếp cận và sử dụng tài nguyên của kho. Điều này bao gồm cả việc tối ưu hóa giao diện quản lý và khả năng kiểm tra tình trạng kho hàng theo thời gian thực, tạo ra sự linh hoạt cho người dùng trong việc quản lý nội dung.

Tổng cộng, mục tiêu nghiên cứu của đề tài là xây dựng một hệ thống quản lý kho hàng tích hợp, linh hoạt, và hiệu quả, không chỉ đáp ứng nhu cầu hiện tại mà còn chuẩn bị cho sự phát triển và thay đổi trong tương lai của ngành quản lý kho hàng.

## Đối tượng, phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Người Quản trị Kho hàng: Nghiên cứu sẽ tập trung vào cách hệ thống hỗ trợ người quản trị kho hàng trong việc quản lý thông tin về sản phẩm, nhà cung cấp, và các giao dịch nhập/xuất kho.

- Nhân viên Kho hàng: Đối tượng này bao gồm những người làm việc tại kho hàng, những người thực hiện các thao tác nhập/xuất kho và kiểm kê hàng hóa.

- Người Sử dụng Cuối cùng: Bao gồm các nhân viên bán hàng, quản lý và các bộ phận khác của doanh nghiệp sử dụng hệ thống để kiểm tra tình trạng hàng tồn kho, tạo đơn đặt hàng và theo dõi hàng hóa.

Phạm vi nghiên cứu:

- Quản lý Sản phẩm và Thông tin: Nghiên cứu sẽ xem xét cách hệ thống quản lý thông tin chi tiết về sản phẩm, bao gồm thông tin về nhà cung cấp, số lượng tồn kho, và vị trí lưu trữ.

- Quản lý Nhà cung cấp: Tính năng quản lý thông tin nhà cung cấp, bao gồm đăng ký mới, cập nhật thông tin và lịch sử giao dịch với mỗi nhà cung cấp.

- Giao diện Người dùng: Đối với nhân viên kho và các bộ phận liên quan, nghiên cứu sẽ xem xét thiết kế giao diện để đảm bảo tính thân thiện và dễ sử dụng.

- Bảo mật và Quyền Truy cập: Xem xét các biện pháp bảo mật để bảo vệ thông tin sản phẩm và dữ liệu của doanh nghiệp, đảm bảo rằng chỉ những người có quyền mới có thể truy cập vào các chức năng quan trọng.

- Báo cáo và Thống kê:Nghiên cứu cũng sẽ xem xét khả năng tạo báo cáo và thống kê để hỗ trợ người quản trị kho trong việc theo dõi hiệu suất và tình trạng kho hàng, từ đó đưa ra các quyết định quản lý hiệu quả.sách.

## Kết quả mong muốn đạt được

Chúng em mong muốn tạo ra một phần mềm quản lý kho hàng với các chức năng sau:

1. Quản lý Sản phẩm: Hệ thống sẽ lưu trữ và quản lý thông tin chi tiết về sản phẩm, bao gồm tên sản phẩm, mã sản phẩm, mô tả, nhà cung cấp, và số lượng tồn kho. Phần mềm sẽ có chức năng thêm, sửa, xóa thông tin sản phẩm.

2. Quản lý Nhà cung cấp: Hệ thống sẽ lưu trữ thông tin chi tiết về các nhà cung cấp, bao gồm tên, địa chỉ, số điện thoại, email, và lịch sử giao dịch. Phần mềm sẽ có chức năng thêm, sửa, xóa thông tin nhà cung cấp.

3. Quản lý Nhập/Xuất Kho: Hệ thống sẽ theo dõi và ghi nhận các giao dịch nhập/xuất kho, bao gồm ngày giờ, số lượng, và người thực hiện. Phần mềm sẽ có chức năng thêm, sửa, xóa và tìm kiếm các giao dịch nhập/xuất kho.

4. Tìm kiếm và Truy xuất Thông tin: Hệ thống sẽ cung cấp tính năng tìm kiếm nhanh thông tin sản phẩm, nhà cung cấp, và các giao dịch nhập/xuất kho.

5. Báo cáo và Thống kê: Hệ thống sẽ tự động tạo các báo cáo và biểu đồ thống kê về tình trạng kho hàng, lượng hàng nhập/xuất, và hiệu suất của kho hàng.

Tổng thể, chúng em mong muốn tạo ra một hệ thống quản lý kho hàng tích hợp, hiệu quả, và linh hoạt, đáp ứng nhu cầu hiện tại và chuẩn bị cho sự phát triển trong tương lai.kê.

# CHƯƠNG 1. GIỚI THIỆU VỀ CÔNG NGHỆ SỬ DỤNG

## 1.1. Công nghệ JDBC

### 1.1.1 Định nghĩa

JDBC là API Java cơ sở, mà nó cung cấp một số lớp và giao diện được viết bằng Java để truy xuất và thao tác với nhiều loại hệ cơ sở dữ liệu khác nhau.

Sự kết hợp của JDBC API và Java nền tảng cung cấp lợi thế cho việc truy xuất và tương tác với nhiều nguồn dữ liệu khác nhau và tính nhất quán của hoạt động trên một máy ảo Java (JVM). Đối với nhà phát triển, điều này giúp tránh việc phải viết một chương trình riêng biệt để truy cập các hệ cơ sở dữ liệu khác nhau như SQL Server, Oracle hoặc IBM DB2. Thay vào đó, một chương trình đơn lẻ triển khai JDBC có thể gửi Structured Query Language (SQL) hoặc gửi các câu lệnh khác tới các nguồn dữ liệu hoặc cơ sở dữ liệu tương ứng.

Ảnh có chứa văn bản, đồ dùng trong nhà vệ sinh, kem dưỡng da

Mô tả được tạo tự động

Hình 1. 1: Công nghệ JDBC

### 1.1.2 Ưu điểm của JDBC

JDBC có một số ưu điểm sau:

* *Độ linh hoạt cao*: JDBC cho phép tương tác với nhiều hệ quản trị cơ sở dữ liệu khác nhau, tạo ra sự linh hoạt trong việc kết nối và làm việc với các loại cơ sở dữ liệu khác nhau.
* *Hiệu suất cao*: JDBC cung cấp khả năng tối ưu hóa hiệu suất bằng cách sử dụng các câu lệnh SQL tối ưu và quản lý kết nối một cách hiệu quả.
* *Bảo mật mạnh mẽ*: JDBC hỗ trợ các tính năng bảo mật mạnh mẽ, bao gồm cơ chế xác thực và kiểm soát truy cập, giúp bảo vệ dữ liệu cơ sở dữ liệu.
* *Dễ học và sử dụng*: API của JDBC được thiết kế để dễ học và sử dụng, giúp nhà phát triển nhanh chóng làm quen và xây dựng ứng dụng kết nối cơ sở dữ liệu.
* *Khả năng tương thích mạnh mẽ*: JDBC tương thích với nhiều hệ thống cơ sở dữ liệu và được hỗ trợ rộng rãi trong cộng đồng Java, giúp đảm bảo tính tương thích và sẵn sàng sử dụng.

### 1.1.3 Nhược điểm của JDBC

JDBC có những nhược điểm sau:

* *Phức tạp*: Việc sử dụng JDBC có thể trở nên phức tạp đối với những ứng dụng lớn và phức tạp, đặc biệt là khi phải quản lý nhiều kết nối đồng thời.
* *Làm gián đoạn mã nguồn*: Sử dụng JDBC thường làm gián đoạn mã nguồn Java bởi các câu lệnh SQL, đặc biệt là khi các lệnh SQL được nhúng trực tiếp vào mã nguồn Java.
* *Không linh hoạt đối với thay đổi cơ sở dữ liệu*: Nếu có sự thay đổi trong cơ sở dữ liệu, việc thay đổi mã nguồn JDBC để phản ánh các thay đổi này có thể là công việc tốn kém và phức tạp.
* *Thiếu các tiện ích tích hợp*: JDBC thiếu một số tính năng tự động và tiện ích tích hợp mà các framework ORM (Object-Relational Mapping) như Hibernate có. Điều này có thể làm tăng khối lượng công việc cho nhà phát triển.
* *Bảo mật đòi hỏi quản lý riêng biệt*: Trong môi trường thực, việc quản lý bảo mật cho các kết nối cơ sở dữ liệu thường đòi hỏi sự quản lý riêng biệt, điều này có thể làm gia tăng khó khăn và rủi ro bảo mật.

### 1.1.4 Kiến trúc JDBC

* Mô hình 2 tầng: JDBC API hỗ trợ mô hình hai tầng cũng như mô hình ba tầng xử lý dữ liệu cho các mô hình truy xuất cơ sở dữ liệu. Trong mô hình hai tầng hệ thống máy khách/máy chủ, máy khách có thể liên hệ trực tiếp với cơ sở dữ liệu của máy chủ mà không cần bất kỳ công nghệ trung gian hoặc máy chủ khác. Trong mô hình hai tầng môi trường JDBC, các ứng dụng Java là khách và DBMS là cơ sở dữ liệu máy chủ.

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

Hình 1. 2: Mô hình 2 tầng

* Mô hình 3 tầng: Trong mô hình ba tầng, tầng giữa là tầng dịch vụ, một máy chủ thứ ba đảm nhận việc gửi yêu cầu của khách hàng tới máy chủ cơ sở dữ liệu. Tầng giữa giúp tách cơ sở dữ liệu máy chủ từ máy chủ WEB. Sự tham gia của máy chủ thứ 3 hoặc máy chủ Proxy tăng cường an ninh bằng cách đi qua tất cả các yêu cầu đến máy chủ cơ sở dữ liệu thông qua máy chủ Proxy. Máy chủ cơ sở dữ liệu xử lý yêu cầu và gửi lại kết quả tới tầng giữa (Proxy Server), một lần nữa kết quả được gửi trả về máy khách (Client). Mô hình ba tầng có lợi thế hơn so với mô hình hai tầng là nó đơn giản hóa và giảm chi phí triển khai ứng dụng, đồng thời giảm tải cho máy chủ cơ sở dữ liệu. Mô hình ba tầng cũng có thể được gọi là mô hình phân tán.

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

Hình 1. 3: Mô hình 3 tầng

* JDBC API:
* JDBC API là bộ sưu tập của các cách định nghĩa cơ sở dữ liệu theo nhiều cách khác nhau và các ứng dụng giao tiếp với nhau. Cốt lõi của JDBC API được dựa trên Java, vì vậy, nó được dùng như là nền tảng để xây dựng chung giữa ba tầng kiến trúc. Do đó, JDBC API là tầng giữa. Nó định nghĩa thế nào là mở kết nối của một ứng dụng và cơ sở dữ liệu, các yêu cầu được gửi tới cơ sở dữ liệu, các câu lệnh truy vấn SQL được thực thi, và kết quả của câu truy vấn đó được lấy ra, JDBC đã đạt được mục tiêu thông qua một tập các giao diện Java, đó là sự thực hiện một cách riêng biết của một lớp cho một cơ sở dữ liệu cụ thể và được gọi là trình điều khiển JDBC (JDBC Driver).

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

Hình 1. 4: JDBC API

### 1.1.5 Trình điều khiển JDBC

Trình điều khiển JDBC là cơ sở của JDBC API và chịu trách nhiệm đảm bảo ứng dụng có quyền truy cập nhất quán và đồng bộ với cơ sở dữ liệu. Các trình điều khiển chuyển đổi yêu cầu của khách hàng thành định dạng dễ hiểu và sau đó truyền nó vào cơ sở dữ liệu. Phản ứng này cũng được xử lý bởi trình điều khiển JDBC, sau đó được chuyển đổi sang mã Java và trình bày cho khách hàng.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình 1. 5: Trình điều khiển JDBC

Có bốn loại trình điều khiển được mô tả dưới đây:

* Trình điều khiển loại 1:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biểu đồ

Mô tả được tạo tự động

Hình 1. 6: Trình điều khiển loại 1

* Trình điều khiển loại 1 là một sản phẩm cầu nỗi phần mềm Java, còn được gọi là cầu nổi JDBC-ODBC cộng thêm với trình điều khiển ODBC.
* Những đặc tính: Trình điều khiển loại 1 sử dụng một công nghệ chuyển tiếp cung cấp ODBC truy cập thông qua trình điều khiển ODBC. Thiết lập này là một liên kết giữa JDBC API và ODBC APL OBDC API lần lượt thực hiện để có thể tiếp cận với cơ sở dữ liệu thông qua các trình điều khiển ODBC chuẩn. Máy khách cần phải cài đặt các kho hàng ODBC gốc, trình điều khiển, và các tập tin hỗ trợ cần thiết, và trong hầu hết các trường hợp, cơ sở dữ liệu của khách hàng là những đoạn mã. Đây sẽ là trình điều khiển thích hợp cho một mạng doanh nghiệp và khách hàng cài đặt ở đâu không phải là vấn đề lớn.
* Ưu điểm: Trình điều khiển loại 1 được viết cho phép truy cập và các cơ sở dữ liệu khác nhau thông qua việc thực hiện trình điều khiển ODBC. Trong một số trường hợp, sự lựa chọn duy nhất của khách hàng sẽ là MS Access hoặc Mircosoft SQL Server cho việc gọi tới ODBC.
* Nhược điểm: Trình điều khiển loại 1 không hỗ trợ hoàn toàn các ứng dụng được cài đặt trên máy khách hàng. Các kho hàng ODBC và các đoạn mã cơ sở dữ liệu phải nằm trên máy chủ, do đó nó làm giảm hiệu suất.
* Trình điều khiển loại 2:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Hình 1. 7: Trình điều khiển loại 2

* Trình điều khiển loại 2 được gọi là Native-API, là một phần của trình điều khiển Java.
* Những đặc tính: Trình điều khiển loại 2 bao gồm những đoạn mã Java được chuyển đổi từ lời gọi JDBC tới một cơ sở dữ liệu API địa phương như Oracle, Sybase, DB2, hoặc các hệ quản trị cơ sở dữ liệu khác. Đi kèm với các lời gọi trên là các phương thức để truy xuất vào cơ sở dữ liệu. Các đặc tính chung cơ bản của trình điều khiển này là việc đi kèm với các cơ sở dữ liệu để có thể giải thích được lời gọi JDBC đến cơ sở dữ liệu, ví dụ, Oracle OCI cung cấp trình điều khiển. Trình điều khiển loại 2 cũng cần có kho hàng gốc cho cơ sở dữ liệu của khách hàng và được cái đặt trên máy khách hàng giống như trình điều khiển loại 1.
* Ưu điểm. Trình điều khiển loại 2 có hiệu suất làm việc cao hơn so với trình điều khiển loại 1. Nó có các lời gọi được chuyển sang một cơ sở dữ liệu cụ thể.
* Nhược điểm: Trình điều khiển loại 2 không hỗ trợ các ứng dụng mà phần mềm đó khách hàng không cài đặt trên máy, như là các yêu cầu đoạn mã gốc được yêu cầu trên máy khách hàng. Cơ sở dữ liệu và kho hàng gốc đều nằm trên máy chủ, điều này sẽ làm giảm hiệu suất chương trình.
* Trình điều khiển loại 3:

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

Mô tả được tạo tự động

Hình 1. 8: Trình điều khiển loại 3

* Trình điều khiển loại 3 được biến đến như JDBC-NET, một trình điều khiển thuần Java.
* Những đặc tính: Trình điều khiển loại 3 là một trình điều khiển thuần Java, là sự chuyển đổi từ lời gọi JDBC vào một hệ quản trị cơ sở dữ liệu với một giao thức mạng độc lập, là một lần nữa chuyển dịch lời gọi tới cơ sở dữ liệu cụ thể bơi một máy chủ ở tầng trung gian. Trinh điều khiển này không yêu cầu bất kỳ cơ sở dữ liệu, kho hàng có nguồn gốc cụ thể sẽ được cài đặt trên máy khách. Trình điều khiển loại 3 hộ trợ các ứng dụng Web nên được thực hiện bởi trình điều khiển loại 3, trình điều khiển này có thể triển khai tốt trên Internet mà không cần phải cài đặt trên mây khách hàng.
* Ưu điểm: Trình điều khiển loại 3 có tính linh hoạt cao, nhất là nó không yêu cầu bất kỳ phần mềm hoặc dịch vụ có nguồn gốc được cái đặt trên máy khách hàng. Nó có một sự thích ứng cao để thay đổi và kiểm soát cơ sở dữ liệu mà không cần phải sửa đổi trình điều khiển ở máy khách hàng.
* Nhược điểm: Cơ sở dữ liệu phải được thực hiện ở tầng máy chủ cụ thể. Vì nó thực hiện các ứng dụng dựa trên Web, nó thực hiện các biện pháp an toàn giống như việc truy xuất qua tường lửa.
* Trình điều khiển loại 4:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Hình 1. 9: Trình điều khiển loại 4

* Trình điều khiển loại 4 được biến đến như Native-Protocol thuần Java hoặc giao thức Java tới cơ sở dữ liệu.
* Những đặc tính: Trình điều khiển loại 4 là một trình điều khiển thuần Java, là sự chuyển đổi lời gọi JDBC vào giao thức mạng để liên lạc trực tiếp với cơ sở dữ liệu. Đây là liên kết giữa khách hàng gọi trực tiếp tới máy chủ hệ quản trị cơ sở dữ liệu và cung cấp một giải pháp thiết thực cho việc truy xuất mạng nội bộ. Trong hầu các trường hợp, các trình điều khiển cung cấp bởi nhà cung cấp cơ sở dữ liệu. Các trình điều khiển này cũng không cần một kho hàng cụ thể cho cơ sở dữ liệu có nguồn gốc cụ thể để cấu hình trên máy của khách hàng và có thể triển khai trên Web mà không cần khách hàng cài đặt, theo yêu cầu cho các trình điều khiển loại 3.
* Ưu điểm: Trình điều khiển loại 4 có thể giao tiếp trực tiếp với cơ sở dữ liệu bằng cách sử dụng Java Socket, chứ không cần phải thông qua một tầng trung gian hay kho hàng gốc nào. Đây chính là lý do các trình điều khiển JDBC hoạt động nhanh nhất. Nó không cần phải bồ xung kho hàng gốc là cần thiết để cài đặt trên máy khách hàng.
* Nhược điểm: Hạn chế duy nhất của trình điều khiển loại 4 là họ được cơ sở dữ liệu cụ thể. Do đó, nếu trong trường hợp, các thay đổi về sau với cơ sở dữ liệu, thì các nhà phát triển ứng dụng có thể phải mua và triển khai một trình điều khiển loại 4 cụ thể mới cho các cơ sở dữ liệu.

### 1.1.6 Các bước để phát triển một ứng dụng JDBC

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Hình 1. 10: Các bước để phát triển một ứng dụng JDBC

* Đăng kí JDBC Driver

Bước đầu tiên trong một kết nối cơ sở dữ liệu các trình điều khiển JDBC với lớp DriverManager. DriverManager là một lớp tĩnh, trong đó quản lý các thiết lập của JDBC Driver có sẵn cho một ứng dụng. Đây là trách nhiệm của DriverManager, có trách nhiệm quản lý tất cả các tài liệu tham khảo cho tất cả các đối tượng điều khiển có sẵn cho một khách hàng JDBC.

* Thiết lập kết nối cơ sở dữ liệu

Sau khi trình điều khiển được tài, kết nối với cơ sở dữ liệu có thể được thành lập. Một URL (Uniform Resource Locator) cơ sở dữ liệu xác định một kết nổi JDBC và thông báo cho người quản lý trình điều khiển mà trình điều khiển và nguồn dữ liệu để sử dụng.

* Tạo và thực hiện một lệnh SQL

Khi kết nối với cơ sở dữ liệu được thành lập, các câu lệnh SQL có thể được nhập tới cơ sở dữ liệu để xử lý. Câu lệnh SQL là ngôn ngữ phố biển được chấp nhận cho ngôn ngữ truy vẫn để thực hiện hành động truy tim, chèn, cập nhật hoặc xóa trên cơ sở dữ liệu.

* Kết quả quá trình

Sau khi thực hiện các câu lệnh SQL, kết quả được xử lý và trích ra bằng cách sử dụng các đối tượng ResultSet.

* Đóng kết nối cơ sở dữ liệu

Kết nối cơ sở dữ liệu là một nguồn lực quan trọng và hạn chế, kết nối sẽ được đóng bằng cách thực hiện một lẫn phương thức close() là kết thúc.

### 1.1.7 Nạp trình điều khiển

Bước đầu tiên trong việc thành lập một kết nối với cơ sở dữ liệu và sử dụng trình điều khiển JDBC, là nạp lớp trình điều khiển bằng cách sử dụng phương thức Class.forName(). Phương thức Class.forName() chỉ định tạo ra một thể hiện của lớp trinh điều khiển và nó được đăng ký với lớp DriverManager. Sau khi trình điều khiển được nạp thành công, kết nối với cơ sở dữ liệu có thể được thành lập.

Cú pháp: Class.forName(<protocol>); Trong đó Protocol là những đặc tả cho các trình điều khiển được sử dụng.

Mẫu: Class.forName("jdbc.odbc.JdbcOdbcDriver"); Đây là đoạn mã thiết lập kết nối cho trình điều khiển loại 1, trình điều khiển JDBC-ODBC.

* Các kết nối:

Bước thứ hai để thực hiện kết nối cơ sở dữ liệu là kết nối các trình điều khiển với DBMS. Kết nối được yêu cầu bằng cách sử dụng phương thức DriverManage.getConnection(); Phương thức này sẽ kiểm tra tất cả các trình điều khiển sẵn có để thực hiện việc kết nối. Phương thức này cũng cho phép kiểm tra trình điều khiển nhận URL, và gửi tới máy khách.

* Tạo câu lệnh và truy vấn

Khi kết nối với cơ sở dữ liệu được thành lập, một câu lệnh cần đã được tạo ra từ query execution. Các câu lệnh là đối tượng thường xuyên nhất được sử dụng để thực thi các lệnh truy vấn SQL mà không cần tham số nào thông qua. Các câu lệnh được tạo ra bởi phương thức Connection.createStatement();

Một câu lệnh có thể được chia làm ba loại dựa trên câu lệnh SQL viết gửi tới cơ sở dữ liệu, Statement và PreparedStatement được kế thừa từ giao diện Statement. CallableStatement được kế thừa từ PreparedStatement. Một đối tượng PreparedStatment thực hiện lệnh biên dịch câu lệnh SQL có hoặc không có tham số bên trong.

* Sử dụng "executeQuery()" và "ResultSet"

Một đối tượng Statement khi được tạo ra có phương thức để thực hiện với các cơ sở dữ liệu khác nhau.

Phương thức executeQuery() là một trong những phương thức để lấy thông tin từ cơ sở dữ liệu. Nó xác nhận một lệnh đơn giản SQL SELECT như một tham số và trả lại các hàng cơ sở dữ liệu ở dạng một đối tượng ResultSet.

* Xử lý ngoại lệ trong ứng dụng JDBC

Trong khi làm việc với ứng dụng cơ sở dữ liệu và JDBC API, thinh thoảng có thể có tính huống mà có thể gây ra các trường hợp ngoại lệ. Và chúng ta thường sử dụng ClassNotFoundException và SQL Exception để bắt lỗi.

### 1.1.8 Xử lý kết quả và đóng kết nối cơ sở dữ liệu

* Điều cần để xử lý truy vấn:

Một khi các truy vấn tới cơ sở dữ liệu đã được thực thi và đổi tượng ResultSet đã được tạo, thì bước tiếp theo sẽ là xử lý và lấy kết quả trả về từ ResuilSet. Như dữ liệu trong ResultSet là bảng được sắp xếp có định dạng và vị trí con trỏ là ở trước dòng đầu tiên, nó cần phải sử dụng phương thức next) để tiếp tục.

Phương thức next() cho phép đi qua phía trước bằng cách di chuyền con trò một hàng về phía trước. Nó trả về kết quả đúng nều vị tri con trò hiện tại là hợp lệ và trả về giá trị sai khi con trỏ ở vị trí cuối cùng của bảng.

* Các phương thức xử lý truy vẫn

Các phương thức người nhận được sử dụng để trích xuất dữ liệu hàng hiện tại của đối tượng ResultSet và lưu nó vào biên của Java có kiểu dữ liệu tương ứng. Những phương thức này được khai bảo bởi giao diện ResultSet để sử dụng giá trị cột, hoặc các chỉ mục hoặc tên của cột. Nếu trong có nhiều hơn một cột có cùng tên, thì giá trị cột đầu tiên được kết hợp sẽ được trả lại. Có nhiều phương thức để thực hiện phương thực nhận này.

* getString()
* getInt()
* getFloat()
* getObject()
* Đóng kết nối cơ sở dữ liệu

Bước cuối cùng cũng là bước quan trọng nhất trong một ứng dụng cơ sở dữ liệu là hoàn thành tất cả các kết nối và các câu lệnh, sau khi xử lý xong. Kết nổi mở có thể gây rắc rồi an ninh. Đơn giản với phương thức close() được cung cấp bởi các đối tượng Connection, Statement và ResultSet để đạt được mục đích. Để giữ cho các phương thức thoát cơ sở dữ liệu được ngăn nắp và luôn được giải thoát (mặc dù có thể bị ngoại lệ), các kết nối cơ sở dữ liệu cần được đóng trong vòng cuối khối.

## 1.2. Công nghệ Java Swing

Swing và AWT là hai kho hàng quan trọng trong việc phát triển giao diện đồ họa người dùng (GUI) trong ứng dụng Java để tạo ra các ứng dụng đa nền tảng với giao diện đẹp và tương tác. Swing là một bộ mở rộng của AWT, được phát triển sau và có nhiều tính năng hơn.

### Java AWT

Java AWT (Abstract Window Toolkit) là một gói kho hàng của ngôn ngữ lập trình Java, được thiết kế để xây dựng ứng dụng đồ họa mở rộng. AWT được tạo ra để thay thế gói kho hàng AWT (Applet Window Toolkit) trong việc xây dựng các ứng dụng đồ họa mở rộng trên máy tính. Tuy nhiên, sau khi được phát triển, Java Swing (tên gọi là JFC Swing ở thời điểm ban đầu) được phát triển và được coi là giải pháp tốt hơn để xây dựng các ứng dụng đồ họa mở rộng. Do đó, Java AWT hiện được coi là cũ và không được khuyến khích để sử dụng trong việc phát triển các ứng dụng mới [1].

Java AWT cung cấp một số lớp và phương thức cơ bản để xây dựng các ứng dụng đồ họa. Một số lớp và phương thức cơ bản của AWT bao gồm:

* Các lớp cơ sở (đa số kế thừa từ lớp java.awt.Component): lớp Button, Label, TextField, Checkbox.
* Lớp cửa sổ (java.awt.Frame): lớp mặc định để xây dựng ứng dụng có giao diện người dùng.
* Các lớp điều khiển di chuột và bàn phím: lớp MouseEvent, KeyEvent.
* Các lớp hình đa giác (được sử dụng để vẽ hình): lớp Shape, Rectangle, Ellipse.

Một số đặc điểm của Java AWT:

* AWT chỉ hỗ trợ việc tạo ra các ứng dụng có giao diện người dùng dựa trên Windows. Điều này giới hạn AWT chỉ được sử dụng để xây dựng các ứng dụng đa năng trên Windows.
* AWT không hỗ trợ nhiều chức năng như định dạng, kích thước văn bản, hiệu ứng, hỗ trợ các ngôn ngữ khác nhau. Do đó, ứng dụng được xây dựng bằng AWT thường không đẹp mắt và không thân thiện đối với người dùng.
* Phụ Thuộc Vào Nền Tảng: Kho hàng AWT phụ thuộc vào nền tảng, vì vậy giao diện có thể trông khác nhau trên các hệ điều hành.
* Tùy Biến Hạn Chế: AWT cung cấp ít tính năng tùy chỉnh hơn so với Swing, giới hạn trong việc xây dựng giao diện đa dạng.
* Khối Lượng Lớn: Kho hàng AWT có nhiều lớp và phương thức, đôi khi làm cho quá trình học và phát triển phức tạp hơn.
* Phổ Biến Cổ Điển: Mặc dù AWT đã được sử dụng từ lâu trong phát triển Java, nó không còn phổ biến trong việc phát triển ứng dụng Java hiện đại.

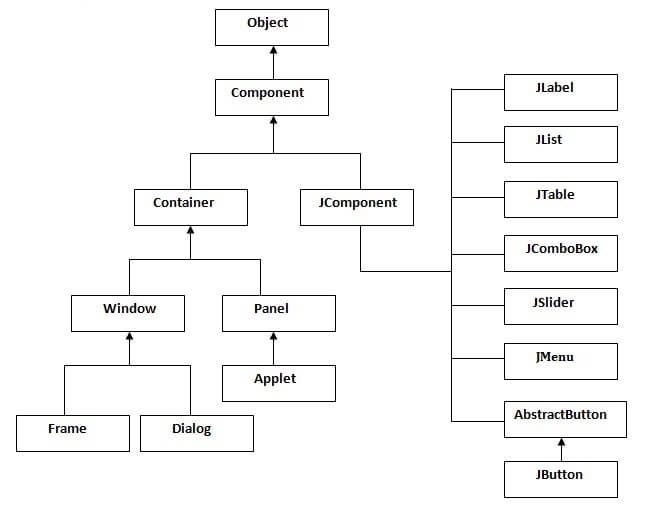
### 1.2.2 Java Swing

Java Swing là một kho hàng GUI mạnh mẽ và linh hoạt được xây dựng trên nền tảng Java Foundation Classes (JFC). Điều này cho phép Swing độc lập với nền tảng, có khả năng tùy biến cao để xây dựng các GUI tương tác và hấp dẫn về mặt hình ảnh và giúp tạo ra giao diện người dùng đa nền tảng

Java Swing với tính năng độc lập nền tảng và khả năng tùy biến cao đã trở thành một lựa chọn phổ biến trong việc phát triển ứng dụng GUI. Trong khi đó, AWT là kho hàng cổ điển ban đầu của Java, cung cấp các thành phần giao diện đơn giản, nhưng có hạn chế trong việc tạo ra giao diện phức tạp và đẹp mắt. Dưới đây là một số điểm cần biết về Java Swing:

* Khả năng Tùy Biến Cao: Swing cung cấp nhiều thành phần GUI như bảng, thanh trượt, thanh tiến trình, cây, bảng chọn màu, v.v. Điều này giúp bạn xây dựng giao diện phức tạp và đa dạng.
* Độc Lập Nền Tảng: Swing không phụ thuộc vào nền tảng hệ điều hành, vì vậy các ứng dụng Swing sẽ nhìn giống nhau trên các hệ điều hành khác nhau.
* Hiệu Suất Cao: Sử dụng mô hình xử lý sự kiện nhẹ (lightweight) giúp Swing có hiệu suất cao hơn và không cần gọi đến API gốc của hệ điều hành.
* Kết Hợp Với Công Nghệ Khác: Swing có khả năng tích hợp với các công nghệ khác như Java 2D, Java 3D, Java Sound, Java Accessibility, v.v. Điều này cho phép bạn xây dựng các ứng dụng đa phương tiện và đa dạng.
* Java Swing là một phần của Java Foundation Classes (JFC) được sử dụng để tạo các ứng dụng window-based. Nó được xây dựng trên API AWT (Abstract Windowing Toolkit) và được viết hoàn toàn bằng Java.
* Không giống như AWT, Java Swing cung cấp các thành phần không phụ thuộc vào nền tảng và nhẹ hơn.
* Gói javax.swing cung cấp các lớp cho java swing API như JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser.

**Hệ thống phân cấp của API java swing được đưa ra dưới đây.** [2]



Hình 1. 11: Hệ thống phân cấp API Java Swing

### 1.2.3 So sánh Java swing với java awt

**Ưu điểm**

* Swing có nhiều thành phần GUI hơn AWT, như các bảng, thanh trượt, thanh tiến trình, cây, bảng chọn màu.
* Swing có khả năng tùy biến cao hơn AWT, có thể thay đổi màu sắc, kiểu dáng, kích thước và hình dạng của các thành phần GUI.
* Swing không phụ thuộc vào nền tảng hệ điều hành, nghĩa là các thành phần GUI sẽ nhìn giống nhau trên các hệ điều hành khác nhau.
* Swing có hiệu suất cao hơn AWT, vì nó sử dụng mô hình xử lý sự kiện nhẹ (lightweight) và không cần gọi đến các API gốc của hệ điều hành.
* Swing có thể kết hợp với các công nghệ khác như Java 2D, Java 3D, Java Sound, Java Accessibility, v.v. để tạo ra các ứng dụng đa phương tiện.

**Nhược điểm**

* Swing khó học và sử dụng hơn AWT, vì nó có nhiều lớp và giao diện trừu tượng.
* Swing cần nhiều bộ nhớ hơn AWT, vì nó có nhiều đối tượng và phương thức.
* Swing có thể gặp vấn đề về tương thích với các phiên bản Java cũ hoặc các trình duyệt web.
* Swing có thể không hiển thị được một số thành phần GUI đặc biệt của hệ điều hành, như menu phải chuột hay thanh công cụ.

## 1.3. Cơ sở dữ liệu MySQL

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, FreeBSD, Unix, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS.

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ NodeJs, PHP, Perl, và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl ...

Một số đặc điểm của Mysql:

* *Khả năng mở rộng và linh hoạt*: Máy chủ cơ sở dữ liệu MySQL đáp ứng nhiều tính năng linh hoạt, nó có sức chứa để xử lý các ứng dụng được nhúng sâu với 1MB dung lượng để chạy kho dữ liệu khổng lồ lên đến hàng terabytes thông tin. Đặc tính đáng chú ý của MySQL là sự linh hoạt về platform với tất cả các phiên bản của Windows, Unix và Linux đang được hỗ trợ. Và đương nhiên, tính chất mã nguồn mở của MySQL cho phép tùy biến theo ý muốn để thêm các yêu cầu phù hợp cho database server.
* *Hiệu năng cao*: Với kiến trúc storage-engine cho phép các chuyên gia cơ sở dữ liệu cấu hình máy chủ cơ sở dữ liệu MySQL đặc trưng cho các ứng dụng chuyên biệt. Dù ứng dụng là website dung lượng lớn phục vụ hàng triệu người/ngày hay hệ thống xử lý giao dịch tốc độ cao thì MySQL đều đáp ứng được khả năng xử lý khắt khe của mọi hệ thống. Với những tiện ích tải tốc độ cao, cơ chế xử lý nâng cao khác và đặc biệt bộ nhớ cache, MySQL đưa ra tất cả nhưng tính năng cần có cho hệ thống doanh nghiệp khó tính hiện nay.
* *Tính sẵn sàng cao*: MySQL đảm bảo sự tin cậy và có thể sử dụng ngay. MySQL đưa ra nhiều tùy chọn có thể “mì ăn liền” ngay từ cấu hình tái tạo chủ/tớ tốc độ cao, để các nhà phân phối thứ 3 có thể đưa ra những điều hướng có thể dùng ngay duy nhất cho server cơ sở dữ liệu MySQL.
* *Hỗ trợ giao dịch mạnh mẽ*: MySQL đưa ra một trong số những engine giao dịch cơ sở dữ liệu tốt nhất trên thị trường. Các đặc trưng bao gồm, khóa mức dòng không hạn chế, hỗ trợ giao dịch ACID hoàn thiện, khả năng giao dịch được phân loại và hỗ trợ giao dịch đa dạng (multi-version) mà người đọc không bao giờ cản trở cho người viết và ngược lại. Dữ liệu được đảm bảo trong suốt quá trình server có hiệu lực, các mức giao dịch độc lập được chuyên môn hóa, khi phát hiện có lỗi khóa chết ngay tức thì.
* *Bảo vệ dữ liệu mạnh mẽ*: Việc quan trọng của các doanh nghiệp là bảo mật dữ liệu, MySQL tích hợp các tính năng bảo mật an toàn tuyệt đối. Với việc xác nhận truy cập cơ sở dữ liệu, MySQL trang bị các kỹ thuật mạnh, chỉ có người sử dụng đã được xác nhận mới truy cập được vào cơ sở dữ liệu. SSH và SSL cũng được hỗ trợ để đảm bảo kết nối an toàn và bảo mật. Tiện ích backup và recovery cung cấp bởi MySQL và các hãng phần mềm thứ 3 cho phép backup logic và vật lý cũng như recovery toàn bộ hoặc tại một thời điểm.
* *Phát triển ứng dụng toàn diện*: MySQL trở thành cơ sở dữ liệu mã nguồn mở phổ biến nhất hiện nay một phần là do cung cấp hỗ trợ hỗn hợp cho bất cứ sự phát triển ứng dụng nào cần. Trong cơ sở dữ liệu, hỗ trợ có thể được tìm thấy trong các trigger, stored procedure, cursor, view, ANSI-standard SQL,… MySQL cũng cung cấp các bộ kết nối như: JDBC, ODBC,… để tất cả các form của ứng dụng sử dụng MySQL như một server quản lý dữ liệu được đề xuất hàng đầu.
* *Quản lý dễ dàng*: Cài đặt MySQL khá nhanh và trung bình từ khi download phần mềm tới khi cài đặt thành công chỉ mất chưa đầy 15 phút. Cho dù platform là Linux, Microsoft Windows, Macintosh hoặc Unix thì quá trình cũng diễn ra nhanh chóng. Khi đã cài đặt, tính năng quản lý như tự khởi động lại, tự động mở rộng không gian và cấu hình động sẵn sàng cho người quản trị cơ sở dữ liệu bắt đầu làm việc. MySQL cung cấp toàn bộ công cụ quản lý đồ họa cho phép một DBA quản lý, sửa chữa và điều khiển hoạt động của nhiều server MySQL từ một máy trạm đơn.
* *Mã nguồn mở và tự do hỗ trợ 24/7*: Nhiều doanh nghiệp còn băn khoăn trong việc giao toàn bộ cho phần mềm mã nguồn mở bởi khó có thể tìm được hỗ trợ hay bảo mật an toàn phục vụ chuyên nghiệp. Với MySQL mọi sự cam kết đều rõ ràng, MySQL cam kết bồi thường khi gặp sự cố.
* *Chi phí sở hữu thấp*: Sử dụng MySQL cho các dự án, doanh nghiệp đều nhận thấy sự tiết kiệm chi phí đáng kể. Người dùng MySQL cũng không mất nhiều thời gian để sửa chữa hoặc vấn đề thời gian chết.

Trước khi đến với khái niệm MySQL Workbench, chúng ta cần phải biết MySQL là gì. Một tập hợp nhiều dữ liệu sẽ được gọi là một cơ sở dữ liệu. Có nhiều loại cơ sở dữ liệu khác nhau từ đơn giản đến phức tạp cũng như nhiều cách lưu trữ cơ sở dữ liệu, truyền thống nhất là trên văn bản, giấy tờ hoặc có thể lưu trữ bằng những phần mềm thông qua các tập tin.

Tuy nhiên, đối với những cơ sở dữ liệu khó và phức tạp như lập trình, chúng ta cần phải sử dụng hệ quản trị cơ sở dữ liệu. Và MySQL là một hệ thống quản trị cơ sở dữ liệu nhưng có mã nguồn mở dùng để tạo và quản trị các cơ sở dữ liệu và được sử dụng với ngôn ngữ truy vấn có cùng một cấu trúc là Structured Query Language – viết tắt là SQL.

MySQL Workbench chính là một chương trình giúp cho người lập trình có thể giao tiếp với hệ cơ sở dữ liệu MySQL thay vì phải sử dụng các lệnh Command-line phức tạp và mất thời gian. MySQL Workbench được thiết kế đơn giản, dễ sử dụng và có thể thích ứng với nhiều hệ điều hành như là Microsoft Windowns, Max OS, Linux hay Ubuntu.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

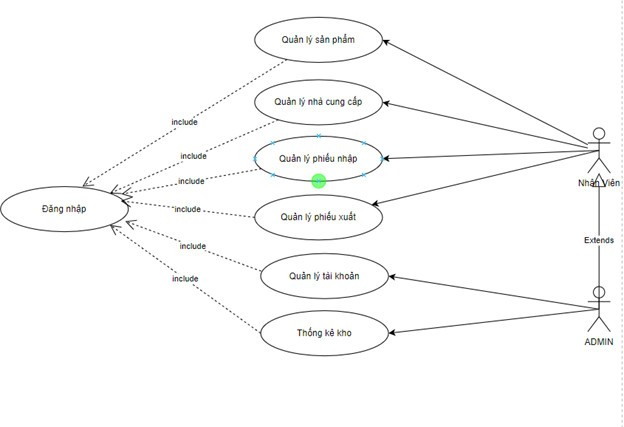
Mô tả được tạo tự động

Hình 1. 12: MySQL Workbench

# CHƯƠNG 2. PHÂN TÍCH YÊU CẦU

## Xác định các yêu cầu cụ thể của phần mềm quản lý dự án

### Biểu đồ use case tổng quan

****

Hình 2. 1: Biểu đồ use case tổng quan

### Mối quan hệ giữa các use case

### 2.1.3 Các yêu cầu chức năng

Các yêu cầu chức năng bao gồm:

* Quản lý tài khoản: tác nhân có thể chỉnh sửa thông tin hồ sơ (thêm, sửa, xóa) và cập nhật lại lên hệ thống.
* Tạo phiếu xuất kho: Phiếu được tạo khi hàng hoá được xuất đi.
* Đăng xuất: các tác nhân thoát khỏi hệ thống.
* Đăng nhập: các tác nhân đăng nhập vào hệ thống.
* Đổi mật khẩu: các tác nhân thay đổi mật khẩu đăng nhập của bản thân hệ thống.
* Quản lí hàng hóa: tác nhân có thể chỉnh sửa thông tin của hàng hóa (thêm, sửa, xóa) và cập nhật lại lên hệ thống.
* Xem thống kê: Xem bảng thống kê lại doanh thu qua các ngày, tháng, quý, năm.

## Đặc tả các yêu cầu phần mềm

### Đặc tả use case Quản lý sản phẩm

**Mô tả các use case:**

**1.Thêm sản phẩm**

* Mô tả: Cho phép người dùng thêm một sản phẩm mới vào hệ thống.
* Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống.
* Hành động chính:
* Người dùng nhập thông tin chi tiết về sản phẩm (tên, mô tả, giá, số lượng, danh mục, v.v.).
* Hệ thống lưu trữ thông tin mới của sản phẩm vào cơ sở dữ liệu.
* Kết quả: Sản phẩm mới được thêm thành công vào hệ thống.

**2.Sửa sản phẩm**

* Mô tả: Cho phép người dùng chỉnh sửa thông tin của một sản phẩm đã có trong hệ thống.
* Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống và có quyền chỉnh sửa sản phẩm.
* Hành động chính:
* Người dùng chọn sản phẩm cần sửa đổi và cập nhật thông tin sản phẩm (tên, mô tả, giá, số lượng, danh mục, v.v.).
* Hệ thống lưu trữ các thay đổi vào cơ sở dữ liệu.
* Kết quả: Thông tin của sản phẩm được cập nhật thành công.

**3.Xóa sản phẩm**

* Mô tả: Cho phép người dùng xóa một sản phẩm khỏi hệ thống.
* Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống và có quyền xóa sản phẩm.
* Hành động chính:
* Người dùng chọn sản phẩm cần xóa.
* Hệ thống xác nhận việc xóa sản phẩm và xóa thông tin sản phẩm khỏi cơ sở dữ liệu.
* Kết quả: Sản phẩm được xóa khỏi hệ thống thành công.

**4.Tìm kiếm sản phẩm**

* Mô tả: Cho phép người dùng tìm kiếm sản phẩm trong hệ thống dựa trên các tiêu chí nhất định.
* Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống.
* Hành động chính:
* Người dùng nhập các tiêu chí tìm kiếm (ví dụ: tên sản phẩm, danh mục, giá, v.v.).
* Hệ thống trả về danh sách các sản phẩm phù hợp với tiêu chí tìm kiếm.
* Kết quả: Người dùng có thể xem được danh sách các sản phẩm thỏa mãn yêu cầu tìm kiếm.

**5.Xuất Excel danh sách sản phẩm**

* Mô tả: Cho phép người dùng xuất danh sách các sản phẩm hiện có trong hệ thống ra file Excel.
* Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống.
* Hành động chính:
* Người dùng chọn tùy chọn xuất Excel từ menu hoặc giao diện.
* Hệ thống tạo và xuất file Excel chứa danh sách các sản phẩm (bao gồm các thông tin như tên, mô tả, giá, số lượng, danh mục, v.v.).
* Kết quả: File Excel chứa thông tin các sản phẩm được tạo ra và có sẵn để tải xuống.

**Quan hệ giữa các use case:**

* Thêm sản phẩm và Sửa sản phẩm có thể có mối quan hệ "include", vì một phần của chức năng sửa sản phẩm có thể bao gồm các bước tương tự như thêm sản phẩm.
* Tìm kiếm sản phẩm và Xuất Excel danh sách sản phẩm có thể có mối quan hệ "extend", vì chức năng tìm kiếm sản phẩm có thể được mở rộng bằng việc cho phép người dùng xuất kết quả tìm kiếm ra file Excel.

### Đặc tả use case Quản lý tài khoản

**Mô tả các use case:**

**1.Thêm Tài khoản**

* Mô tả: Cho phép người quản trị hệ thống thêm một tài khoản mới.
* Tiền điều kiện: Người dùng đăng nhập vào hệ thống với vai trò quản trị viên.
* Hành động chính:

Người quản trị nhập thông tin chi tiết của tài khoản mới (tên đăng nhập, mật khẩu, quyền truy cập, v.v.).

* Hệ thống lưu trữ thông tin tài khoản mới vào cơ sở dữ liệu.
* Kết quả: Tài khoản mới được thêm thành công vào hệ thống.

**2.Sửa Tài khoản**

* Mô tả: Cho phép người quản trị chỉnh sửa thông tin của một tài khoản đã tồn tại.
* Tiền điều kiện: Người dùng đăng nhập vào hệ thống với vai trò quản trị viên và có quyền sửa đổi tài khoản.
* Hành động chính:

Người quản trị chọn tài khoản cần sửa đổi và cập nhật thông tin tài khoản (mật khẩu mới, quyền truy cập, v.v.).

* Hệ thống lưu trữ các thay đổi vào cơ sở dữ liệu.
* Kết quả: Thông tin của tài khoản được cập nhật thành công.

**3.Xóa Tài khoản**

* Mô tả: Cho phép người quản trị xóa một tài khoản khỏi hệ thống.
* Tiền điều kiện: Người dùng đăng nhập vào hệ thống với vai trò quản trị viên và có quyền xóa tài khoản.
* Hành động chính:

Người quản trị chọn tài khoản cần xóa.

* Hệ thống xác nhận việc xóa tài khoản và xóa thông tin tài khoản khỏi cơ sở dữ liệu.
* Kết quả: Tài khoản được xóa khỏi hệ thống thành công.

**4.Tìm kiếm Tài khoản**

* Mô tả: Cho phép người quản trị tìm kiếm tài khoản trong hệ thống dựa trên các tiêu chí nhất định.
* Tiền điều kiện: Người dùng đăng nhập vào hệ thống với vai trò quản trị viên.
* Hành động chính:

Người quản trị nhập các tiêu chí tìm kiếm (ví dụ: tên đăng nhập, quyền truy cập, v.v.).

* Hệ thống trả về danh sách các tài khoản phù hợp với tiêu chí tìm kiếm.
* Kết quả: Người quản trị có thể xem được danh sách các tài khoản thỏa mãn yêu cầu tìm kiếm.

**5.Xuất Excel danh sách Tài khoản**

* Mô tả: Cho phép người quản trị xuất danh sách các tài khoản hiện có trong hệ thống ra file Excel.
* Tiền điều kiện: Người dùng đăng nhập vào hệ thống với vai trò quản trị viên.
* Hành động chính:

Người quản trị chọn tùy chọn xuất Excel từ menu hoặc giao diện.

* Hệ thống tạo và xuất file Excel chứa danh sách các tài khoản (bao gồm các thông tin như tên đăng nhập, quyền truy cập, v.v.).
* Kết quả: File Excel chứa thông tin các tài khoản được tạo ra và có sẵn để tải xuống.

**Quan hệ giữa các use case:**

* Thêm Tài khoản và Sửa Tài khoản có thể có mối quan hệ "include", vì một phần của chức năng sửa tài khoản có thể bao gồm các bước tương tự như thêm tài khoản.
* Tìm kiếm Tài khoản và Xuất Excel danh sách Tài khoản có thể có mối quan hệ "extend", vì chức năng tìm kiếm tài khoản có thể được mở rộng bằng việc cho phép người quản trị xuất kết quả tìm kiếm ra file Excel.

### 2.2.3 Đặc tả use case Quản lý phiếu nhập

1. Tên use case

Quản lý phiếu nhập

2. Mô tả vắn tắt.

Use case này cho phép thêm, sửa và xóa các phiếu nhập trong bảng phieunhap  
3. Luồng các sự kiện

3.1. Luồng cơ bản

1) a.Use case này bắt đầu khi nhấn vào nút “Phiếu nhập” trên thanh Menu. Hệ thống sẽ lấy các thông tin về các phiếu nhập bao gồm: mã phiếu nhập, nhà cung cấp, người tạo, thời gian tạo và tổng tiền từ bảng phieunhap và hiển thị các thông tin đó ra màn hình dưới dạng bảng.

2) Xem chi tiết phiếu nhập:

a) Kích vào 1 sản phẩm và bấm nút “Xem chi tiết”. Hệ thống sẽ lấy các thông tin về chi tiết phiếu nhập bao gồm: mã phiếu nhập, mã máy, tên máy, số lượng, đơn giá từ bảng chitietphieunhap và nhà cung cấp. người tạo, thời gian tạo từ bảng phieunhap hiển thị các thông tin đó ra màn hình.

b) Xuất PDF: Kích vào nút “Xuất PDF” trên màn hình chi tiết phiếu nhập. ”. Hệ thống sẽ lấy các thông tin về chi tiết phiếu nhập bao gồm: mã phiếu nhập, mã máy, tên máy, số lượng, đơn giá từ bảng chitietphieunhap và nhà cung cấp. người tạo, thời gian tạo từ bảng phieunhap và xuất ra file PDF.

3) Thêm phiếu nhập:

a) Click vào vào nút “Nhập hàng” trên thanh Menu. . Hệ thống sẽ lấy các thông tin về chi tiết phiếu nhập bao gồm: mã phiếu nhập, mã máy, tên máy, số lượng, đơn giá từ bảng chitietphieunhap và nhà cung cấp từ bảng phieunhap hiển thị các thông tin đó ra màn hình dưới dạng bảng.

b) Chọn vào 1 sản phẩm, nhập số lượng, chọn nhà cung cấp sau đó kích nút “Nhập hàng”. Hệ thống sẽ thêm phiếu nhập mới vào bảng phieunhap và hiển thị lên màn hình Phiếu nhập danh sách các phiếu nhập đã được cập nhật.

4) Sửa phiếu nhập

a) Nhấn vào một phiếu nhập cần sửa và bấm nút “Sửa” . Hệ thống sẽ thông tin về chi tiết phiếu nhập bao gồm: mã phiếu nhập, mã máy, tên máy, số lượng, đơn giá từ bảng chitietphieunhap và nhà cung cấp từ bảng phieunhap hiển thị các thông tin đó ra màn hình dưới dạng bảng.

b) Sửa phiếu nhập bao gồm: nhà cung cấp, người tạo phiếu, thêm sửa xóa sản phẩm. Sau kích nút “Lưu thay đổi”. Hệ thống sẽ cập nhật dữ liệu mới của phiếu nhập vào bảng phieunhap và hiển thị lên màn hình danh sách các phiếu nhập đã được cập nhật.

5) Xóa phiếu nhập:

a) Chọn 1 phiếu nhập và kích nút “Xóa”. Hệ thống sẽ cập nhật lại bảng phieunhap và hiển thị lên màn hình danh sách phiếu nhập đã được cập nhật.

6) Xuất phiếu nhập:

a) Kích vào nút “Xuất Excel” trên màn hình. Hệ thống sẽ lấy các thông tin về các phiếu nhập bao gồm: mã phiếu nhập, nhà cung cấp, người tạo, thời gian tạo và tổng tiền từ bảng phieunhap và xuất ra file Excel.

Use case kết thúc.

3.2. Các luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo “Lỗi kết nối” và use case kết thúc.

2) Tại bước 3b hoặc 4b trong luồng cơ bản nếu thủ thư nhập thông tin phiếu nhập không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại.

3) Tại bước 4b trong luồng cơ bản nếu thủ thư kích vào nút “x” hệ thống sẽ bỏ qua thao tác sửa và hiển thị danh sách các phiếu nhập trong phieunhap và use case kết thúc.

4) Tại bước 5a trong luồng cơ bản nếu vào nút “Không đồng ý” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các phiếu nhập trong bảng phieunhap và use case kết thúc.

4. Yêu cầu đặc biệt

Không có.

5. Tiền điều kiện

Không có.

6. Hậu điều kiện

Nếu use case kết thúc thành công thì thông tin về phiếu nhập sẽ được cập nhập trong cơ sở dữ liệu.

7. Điểm mở rộng

Không có.

### 2.2.4 Đặc tả use case Thống kê kho

1. Tên use case.

Thống kê kho.

2. Mô tả vắn tắt.

Use case này cho phép xem chi tiết danh sách các sản phẩm được nhập, xuất khỏi kho hàng.

3. Luồng các sự kiện

3.1. Luồng cơ bản

1) xem thống kê kho

a. Use case này bắt đầu khi click vào nút “Tồn Kho” trên thanh Menu. Hệ thống sẽ lấy các thông tin về sản phẩm bao gồm: mã máy, tên máy, số lượng, đơn giá, bộ xử lý, RAM, bộ nhớ, loại máy từ bảng maytinh và hiển thị các thông tin đó ra màn hình dưới dạng bảng.

2) Xuất excel

a. Kích vào nút “Xuất Excel” trên màn hình. Hệ thống sẽ lấy các thông tin về sản phẩm bao gồm: mã máy, tên máy, số lượng, đơn giá, bộ xử lý, RAM, bộ nhớ, loại máy từ bảng maytinh và xuất ra file Excel.

3.2. Các luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo “Lỗi kết nối” và use case kết thúc.

4. Yêu cầu đặc biệt

Không có.

5. Tiền điều kiện

Không có.

6. Hậu điều kiện

Không có.

7. Điểm mở rộng

Không có.

### 2.2.5 Đặc tả use case Quản lý nhà cung cấp

1. Tên use case.

Quản lý Nhà cung cấp.

2. Mô tả vắn tắt.

Use case này cho phép người quản lý kho thêm, sửa và xóa các nhà cung cấp trong bảng NHACUNGCAP.

3. Luồng các sự kiện

3.1. Luồng cơ bản

1) Use case này bắt đầu khi. Người thủ thư nhấn vào nút “Nhà cung cấp” trên màn hình Trang chủ. Hệ thống sẽ lấy các thông tin về các nhà cung cấp bao gồm: Mã nhà cung cấp, Tên nhà cung cấp, Địa chỉ, Điện thoại từ bảng NHACUNGCAP và hiển thị các thông tin đó ra màn hình dưới dạng bảng.

2) Thêm nhà cung cấp:

a) Người quản lý kho nhập nhấn nút thêm trên màn hình, nhập các dữ liệu vào các ô Mã nhà cung cấp, Tên nhà cung cấp, Số điện thoại, Địa chỉ sau đó kích nút “Thêm”. Hệ thống sẽ thêm nhà cung cấp mới vào bảng NHACUNGCAPvà hiển thị lên màn hình thông báo thêm thành công.

b) Người quản lý kho kích nút “OK”. Hệ thống sẽ lấy các thông tin về các nhà cung cấp bao gồm: Mã nhà cung cấp, Tên nhà cung cấp, Địa chỉ, Điện thoại từ bảng NHACUNGCAP và hiển thị các thông tin đó ra màn hình.

3) Sửa nhà cung cấp:

a) Người quản lý kho kích chọn vào một nhà cung cấp cần sửa. Hệ thống sẽ hiển thị thông tin của nhà cung cấp đó lên các ô Mã nhà cung cấp, Tên nhà cung cấp, Số điện thoại, Địa chỉ.

b) Người quản lý kho sửa lại nhà cung cấp sau đó kích nút “Sửa”. Hệ thống sẽ cập nhật dữ liệu mới của nhà cung cấp vào bảng NHACUNGCAP và hiển thị lên màn hình danh sách các nhà cung cấp đã được cập nhật.

4) Xóa nhà cung cấp:

a) Người quản lý kho ấn vào một nhà cung cấp cần xóa và kích nút “Xóa”. Hệ thống sẽ hiển thị thông báo “bạn có chắc muốn xóa nhà cung cấp”.

b) Người quản lý kho kích “Yes”. Hệ thống sẽ cập nhật lại bảng NHACUNGCAP và hiển thị lên màn hình danh sách nhà cung cấp đã được cập nhật.

Use case kết thúc.

3.2. Các luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo “Lỗi kết nối” và use case kết thúc.

2) Tại bước 2a hoặc 3b trong luồng cơ bản nếu người quản lý kho nhập thông tin nhà cung cấp không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi. Người quản lý kho có thể kích nút “Hủy” và use case kết thúc.

3) Tại bước 2a hoặc 3b trong luồng cơ bản nếu người quản lý kích vào nút “Hủy” hệ thống sẽ bỏ qua thao tác thêm mới hoặc sửa tương ứng và hiển thị danh sách các phiếu mượn trong bảng NHACUNGCAP và use case kết thúc.

4) Tại bước 4b trong luồng cơ bản nếu thủ thư kích vào nút “No” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các nhà cung cấp trong bảng NHACUNGCAP và use case kết thúc.

4. Yêu cầu đặc biệt

Không có.

5. Tiền điều kiện

Người quản lý kho cần đăng nhập vào phần mềm.

6. Hậu điều kiện

Nếu use case kết thúc thành công thì thông tin về nhà cung cấp sẽ được cập nhập trong cơ sở dữ liệu.

7. Điểm mở rộng

Không có.

### 2.2.6 Đặc tả use case Quản lý phiếu xuất

1. Tên use case.

Quản lý Phiếu xuất.

2. Mô tả vắn tắt.

Use case này cho phép quản lý kho thêm, sửa xóa các phiếu xuất hàng trong bảng PHIEUXUAT.

Usecase cho phép nhân viên xuất kho Thêm phiếu xuất.

3. Luồng các sự kiện

3.1. Luồng cơ bản

1) Xem chi tiết Phiếu xuất:

a) Use case này bắt đầu khi. Người quản lý kho nhấn vào nút “Phiếu xuất” trên màn hình Trang chủ. Hệ thống sẽ lấy các thông tin về các phiếu xuất bao gồm: Số thứ tự, Mã phiếu, Người tạo, Thời gian tạo, Tổng tiền từ bảng bảng PHIEUXUAT và hiển thị các thông tin đó ra màn hình dưới dạng bảng.

b) Người quản lý kho chọn một Phiếu xuất trên bảng các phiếu xuất và kích nút “Xem chi tiết”. Hệ thống sẽ lấy thông về phiếu xuất bao gồm: Mã phiếu, Người tạo, Thời gian tạo, Tổng tiền, Mã máy, Tên máy, Số lượng, Đơn giá từ bảng CHITIETPHIEUXUAT và hiển thị lên màn hình.

2) Thêm Phiếu xuất:

a) Người quản lý kho kích “Xuất hàng” trên màn hình trang chủ. Hệ thống sẽ lấy thông tin Mã máy, Tên máy, Số lượng, Đơn giá, từ bảng MAYTINH và hiển thị lên màn danh sách hàng hóa.

b) Người quản lý kho kích chọn các hàng hóa và số lượng mong muốn, và kích “Thêm”. Hệ thống sẽ lấy thông tin các sản phẩm gồm Mã máy, Tên máy, Số lượng, Đơn giá, từ bảng MAYTINH hiển thị lên phiếu màn hình dưới dạng danh sách.

c) Người quản lý kích nút “Xuất hàng”. Hệ thống sẽ hiển thị lên thông báo “Bạn có chắc muốn xuất hàng”.

d) Người quản lý kích nút “Yes”. Hệ thống sẽ cập nhật thông tin vào bảng PHIEUXUAT.

3) Sửa Phiếu xuất:

a) Người quản lý kho kích “Phiếu xuất” trên màn hình trang chủ. Hệ thống sẽ lấy thông tin Số thứ tự, Mã phiếu, Người tạo, Thời gian tạo, Tổng tiền từ bảng PHIEUXUAT và hiển thị lên màn danh sách hàng hóa.

b) Người quản lý kho kích chọn một phiếu xuất và kích nút “Sửa” trên thanh công cụ. Hệ thống sẽ lấy thông tin gồm Mã phiếu, Mã máy, Số lượng, Đơn giá, từ bảng bảng CHITIETPHIEUXUAT và hiển thị các thông tin đó ra màn hình.

c) Người quản lý kho thêm, xóa sản phẩm, sửa số lượng và kích nút “Lưu thay đổi”. Hệ thống sẽ cập nhật bảng PHIEUXUAT hiển thị lên thông báo “Sửa thành công”.

d) Người quản lý kích “OK”. Hệ thống lấy thông tin từ bảng PHIEUXUAT và hiển thị danh sách phiếu xuất lên màn hình.

4) Xóa Phiếu xuất:

a) Người quản lý ấn vào một phiếu xuất cần xóa và kích nút “Xóa”. Hệ thống sẽ lấy Mã phiếu xuất từ bảng PHIEUXUAT hiển thị thông báo “Bạn có chắc chắn muốn xóa PX .. ”.

b) Người quản lý kho kích nút “Yes”. Hệ thống sẽ cập nhật lại bảng PHIEUXUAT và hiển thị lên màn hình danh sách các phiếu xuất đã được cập nhật.

Use case kết thúc.

3.2. Các luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo “Lỗi kết nối” và use case kết thúc.

2) Tại bước 2a hoặc 3b trong luồng cơ bản nếu người quản lý kho nhập thông tin phiếu xuất không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi. Người quản lý có thể kích nút “Hủy” và use case kết thúc.

3) Tại bước 2a hoặc 4b trong luồng cơ bản nếu thủ thư kích vào nút “Hủy” hệ thống sẽ bỏ qua thao tác thêm mới, hiển thị danh sách các phiếu xuất trong bảng PHIEUXUAT và use case kết thúc.

4) Tại bước 4b trong luồng cơ bản nếu thủ thư kích vào nút “No” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các phiếu xuất trong bảng PHIEUXUAT và use case kết thúc.

4. Yêu cầu đặc biệt

Không có.

5. Tiền điều kiện

Người quản lý kho cần đăng nhập vào phần mềm.

6. Hậu điều kiện

Nếu use case kết thúc thành công thì thông tin về phiếu xuất sẽ được cập nhập trong cơ sở dữ liệu.

7. Điểm mở rộng

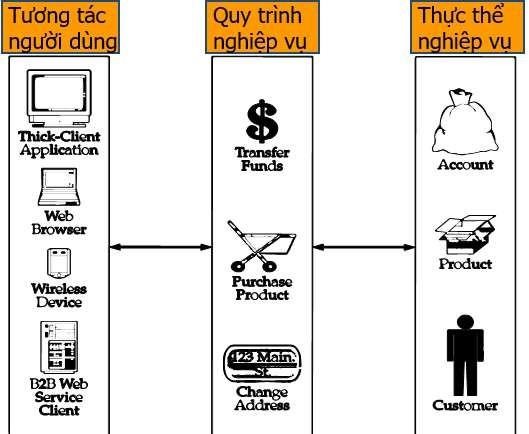
Không có.

# CHƯƠNG 3. THIẾT KẾ PHẦN MỀM

## Kiến trúc tổng quan của phần mềm

### Tổng quan kiến trúc phần mềm

Kiến trúc của một ứng dụng doanh nghiệp bao gồm 3 phần: phần thực thể nghiệp vụ, phần quy trình nghiệp vụ và phần tương tác người dùng.



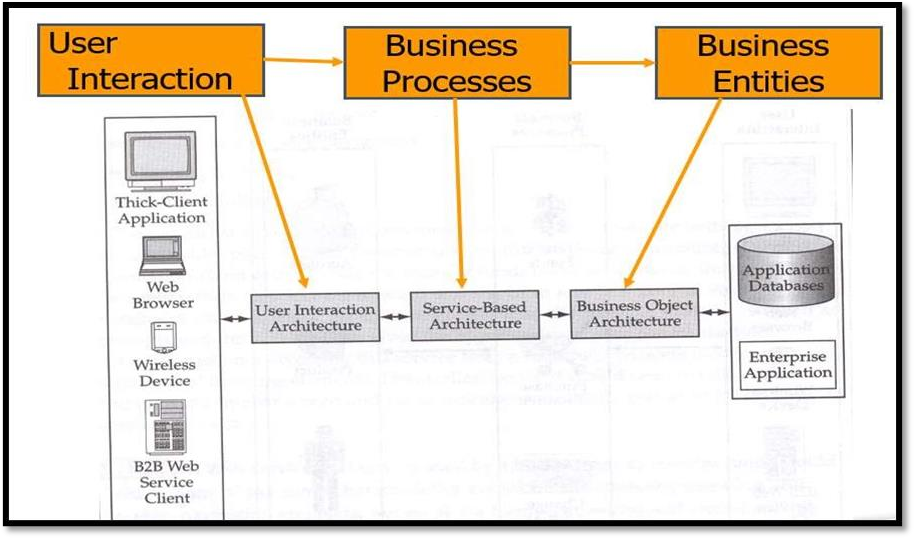
Hình 3. 1: Kiến trúc của một ứng dụng doanh nghiệp

* **Phần thực thể nghiệp vụ (BE)**
* Các phần tử này có chung các đặc điểm:
  + - Các hành vi (Behaviors)
    - Các tính chất (Properties)
    - Các quan hệ với các thực thể khác
    - Các luật hay các đường lối hành động
* Các đặc điểm này sẽ có một ảnh hưởng lớn đến kiến trúc của thực thể nghiệp vụ
* **Phần quy trình nghiệp vụ**
  + Là một luồng công việc cụ thể trong đó có một hoặc nhiều thực thể nghiệp vụ tham gia.
  + Các bước tuân theo đúng quy tắc nghiệp vụ của hệ thống.
  + Là các nghiệp vụ cơ bản trong hệ thống.
  + Cần được truy nhập từ các client khác nhau như: trình duyệt, các điều khiển không dây và web service.
* **Phần tương tác người dùng**

Các đặc điểm chung là:

* + Trình bày ứng dụng, chẳng hạn như HTML hay XML trên HTTP
  + Truy nhập các chức năng và dịch vụ của doanh nghiệp
  + Biểu diễn nội dung tĩnh và động
  + Có luồng màn hình, hay điều hướng giữa các trang
  + Xử lý các Form
  + Điều khiển lỗi

Kiến trúc phần mềm là Các phần tử chính của các ứng dụng doanh nghiệp tương ứng với các tầng (layers) và tạo thành một kiến trúc chung cho các ứng dụng phần mềm. Kiến trúc phần mềm mô hình hóa 3 phần tử chính (UI, BP, BE) này và cung cấp một kỹ thuật cài đặt cho mỗi phần tử.



Hình 3. 2: Mẫu kiến trúc phần mềm

Kiểu kiến trúc là một chuyên môn hóa của phần tử và các loại liên quan, cùng với một bộ những hạn chế về cách nó có thể được sử dụng. Một vài tác giả chỉ ra một số kiểu kiến trúc chính như sau:

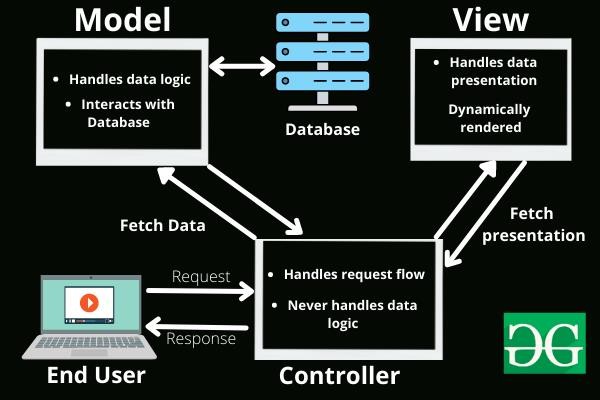
Kiến trúc thường (ví dụ, phân tầng, pipes and filter, blackboard)

* + Các hệ thống phân tán (ví dụ client- server, three- tiers, broker)
  + Các hệ thống tương tác (ví dụ, MVC, Presentation- Abstraction- Control, WPF)
  + Các hệ thống mô phỏng (ví dụ, microkernel, reflection)

Các kiểu khác (ví dụ, batch, interperters, process control, rule- based)

### Mô hình kiến trúc – Mô hình MVC

#### Tổng quan về mô hình MVC



Hình 3. 3: Mô hình MVC

**M (Model)**

* + - * + Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng.
        + Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller.
        + Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một file XML bình thường.
        + Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho xem, truy xuất, xử lý dữ liệu, …

**V (View)**

* + - * + Đây là phần giao diện giao tiếp với người dùng. Người dùng có thể gửi hoặc nhận thông tin từ hệ thống.
        + Các ứng dụng web sử dụng MVC View như một phần của hệ thống, các View được tạo thành bằng mã HTML.
        + View cũng có chức năng ghi nhận hoạt động của người dùng để tương tác với Controller.
        + View không có mối quan hệ trực tiếp với Controller, cũng không được lấy dữ liệu từ Controller mà chỉ hiển thị yêu cầu chuyển cho Controller mà thôi.

**C (Control)**

* + - * + Là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view.
        + Controller kết nối với Model để lấy dữ liệu và đưa lên View các thông tin mà người dùng yêu cầu.

#### Ưu điểm mô hình MVC

Mô hình MVC phù hợp với các dự án với các đội ngũ làm việc độc lập với nhau, như vậy các công việc đều có thể được thực hiện một cách linh hoạt. Các ưu điểm là:

* + - * + Hỗ trợ quá trình phát triển nhanh chóng: Với đặc điểm hoạt động độc lập của từng thành phần, các lập trình viên có thể làm việc đồng thời trên từng bộ phận khác nhau của mô hình này. MVC giúp bạn tiết kiệm rất nhiều thời gian.
        + Khả năng cung cấp đồng thời nhiều khung View: Với mô hình MVC, bạn có thể tạo ra đồng thời nhiều khung View cho Model.
        + Hỗ trợ các kỹ thuật không đồng bộ: MVC có thể hoạt động trên nền tảng JavaScript. Điều này có nghĩa là các ứng dụng MVC có thể hoạt động với các file PDF, các trình duyệt web cụ thể, và cả các widget máy tính.
        + Dễ dàng thao tác chỉnh sửa: Bộ phận Model hoạt động tách biệt với View đồng nghĩa với việc bạn có thể đưa ra các thay đổi, chỉnh sửa hoặc cập nhật dễ dàng ở từng bộ phận.
        + Giữ nguyên trạng thái data: Mô hình MVC truyền lại dữ liệu nhưng không định dạng lại dữ liệu. Do đó, các dữ liệu này có thể được dùng lại cho các thay đổi sau này.
        + Hỗ trợ các nền tảng phát triển SEO: Với mô hình MVC, bạn có thể dễ dàng tạo ra các mã SEO URL để thu hút lượng truy cập đối với ứng dụng bất kỳ.

#### Nhược điểm mô hình MVC

* Khó khăn trong quá trình điều hướng code: Điều hướng khung có thể phức tạp vì mô hình này bao gồm nhiều lớp và yêu cầu người dùng thích ứng với các tiêu chí phân tách của MVC.
* Không thích hợp việc phát triển các ứng dụng nhỏ vì mô hình này yêu cầu bạn lưu trữ một số lượng lớn các file.
* Nhiều khung hoạt động đồng thời: Việc phân tách một tính năng thành ba bộ phận khác nhau dễ dẫn đến hiện tượng phân tán. Do đó, đòi hỏi các nhà phát triển phải duy trì tính nhất quán của nhiều bộ phận cùng một lúc.

## 3.2 Thiết kế biểu đồ trình tự

### 3.2.1 Biểu đồ trình tự use case Quản lý sản phẩm



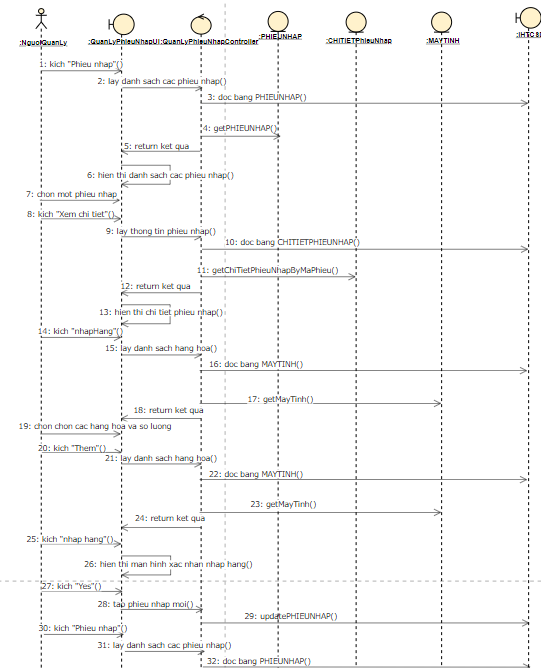
Hình 3. 4: Biểu đồ trình tự use case Quản lý sản phẩm

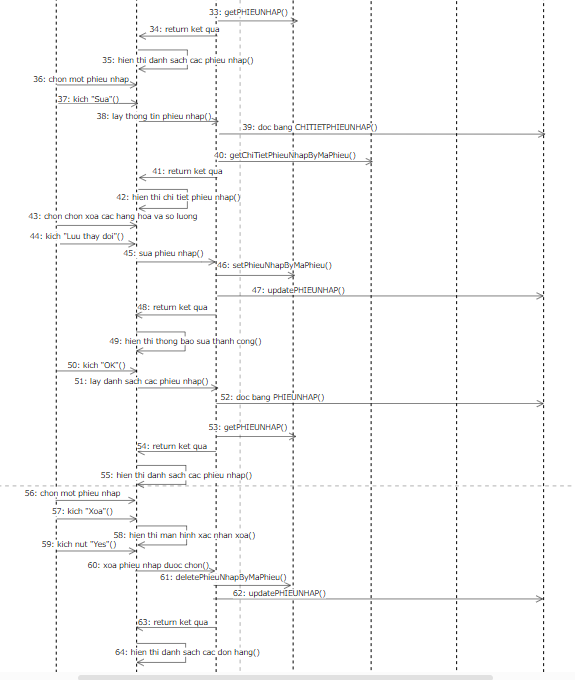
### 3.2.2 Biểu đồ trình tự use case Quản lý tài khoản



Hình 3. 5: Biểu đồ trình tự use case Quản tài khoản

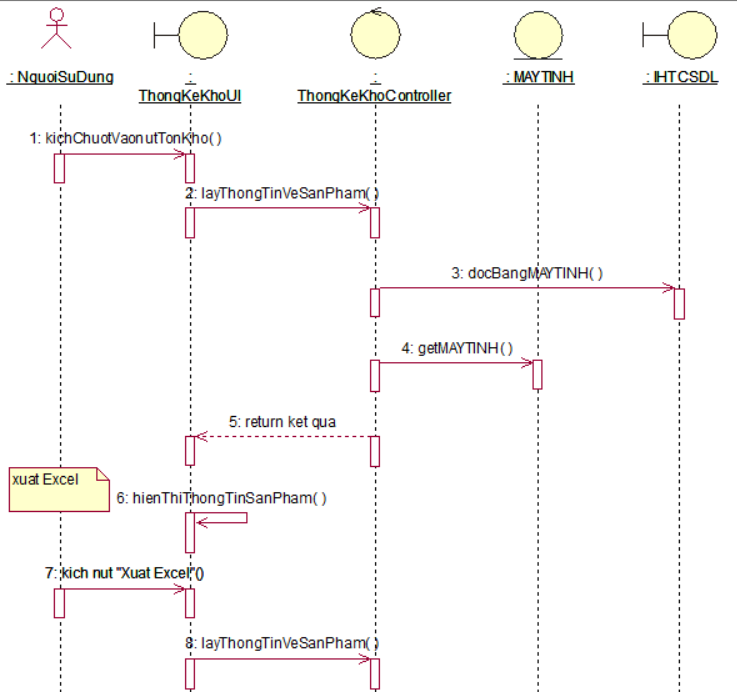
### 3.2.3 Biểu đồ trình tự use case Quản lý phiếu nhập

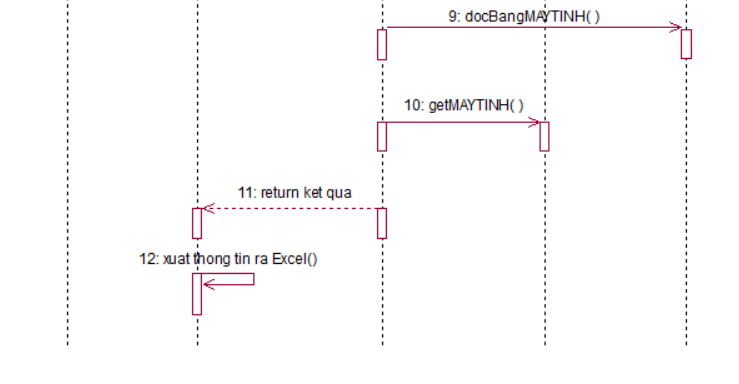




Hình 3. 6: Biểu đồ trình tự use case Quản lý phiếu nhập

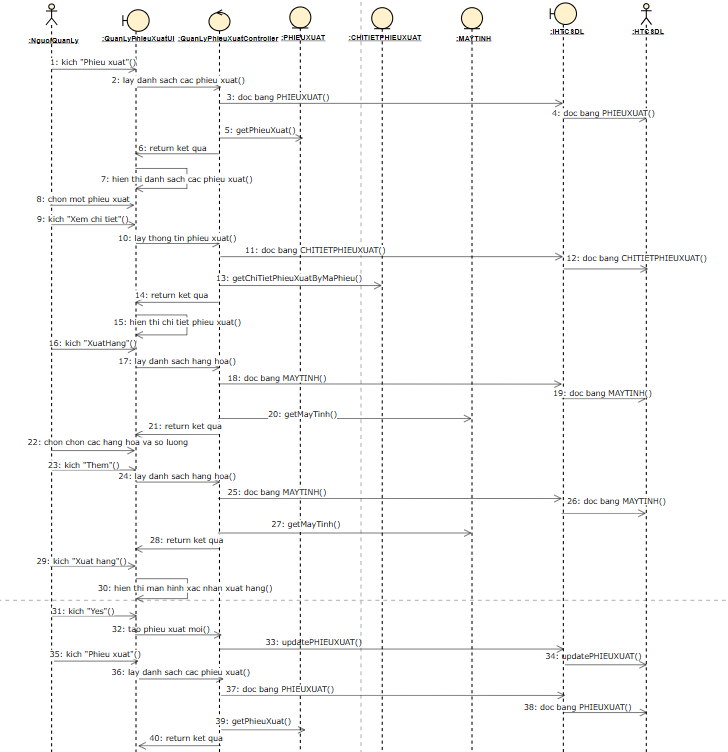
### 3.2.4 Biểu đồ trình tự use case Thống kê kho

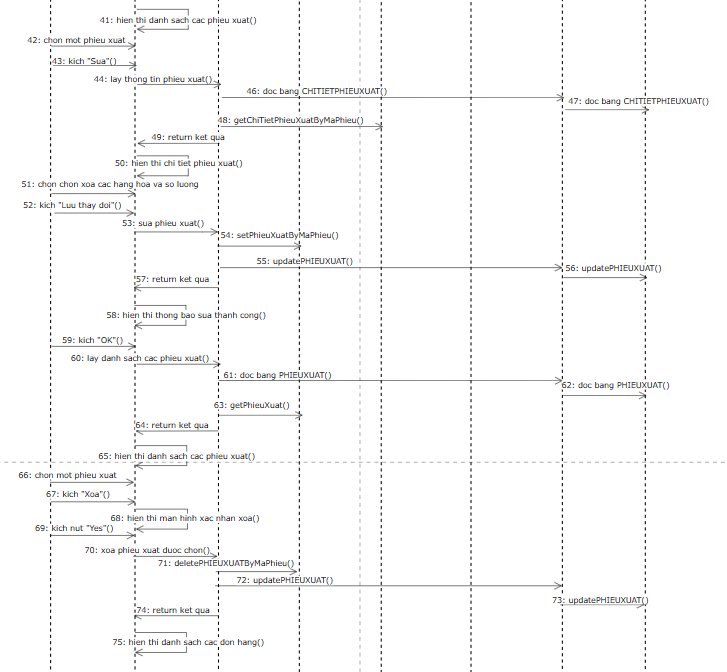
****

****

Hình 3.7: Biểu đồ trình tự use case thống kê kho

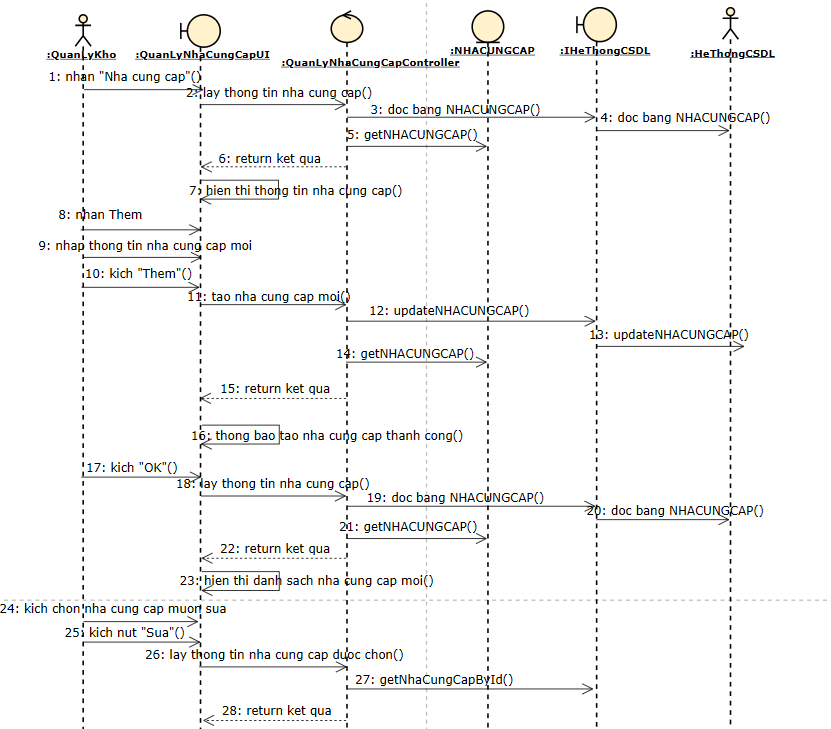
### 3.2.5 Biểu đồ trình tự use case Quản lý phiếu xuất





Hình 3.8: Biểu dồ trình tự use case quản lý phiếu xuất

### 3.2.6 Biểu đồ trình tự use case Quản lý nhà cung cấp

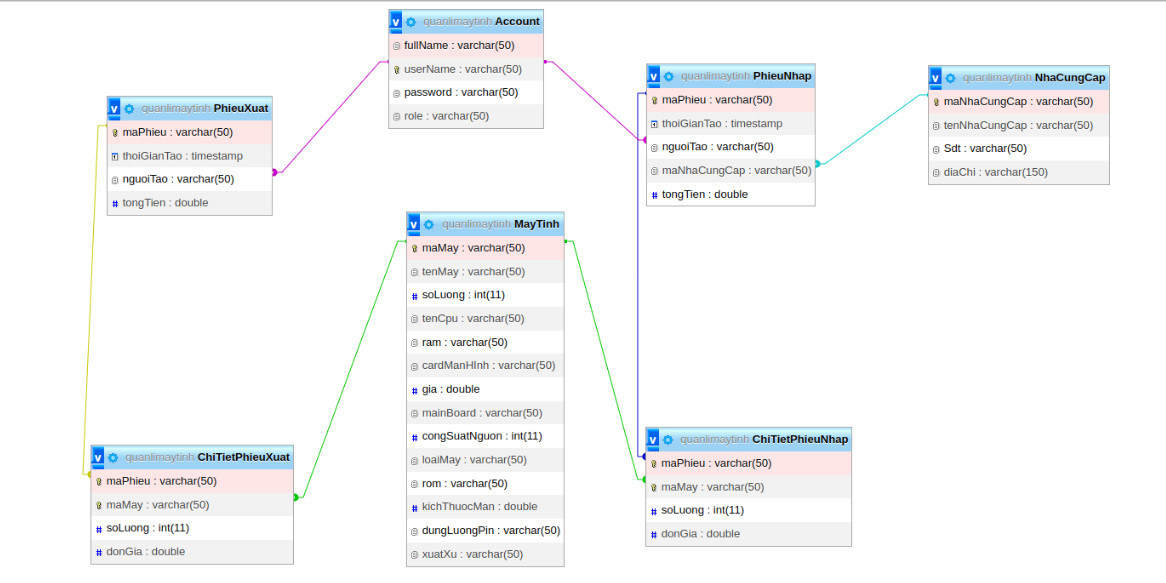




Hình 3.9: Biểu dồ trình tự use case quản lý nhà cung cấp

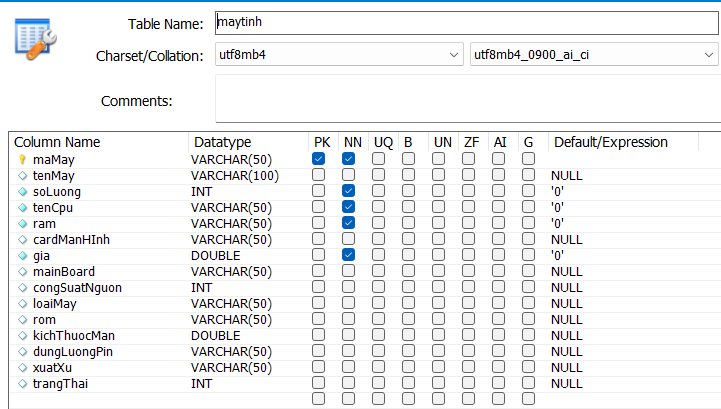
## *3.3* Thiết kế cơ sở dữ liệu

### Biểu đồ thực thể liên kết mức vật lý

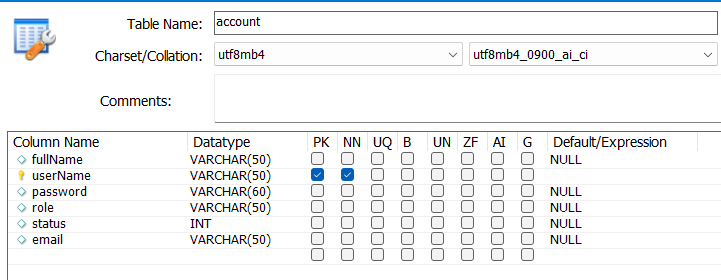


Hình 3.10: Biểu đồ thực thể liên kết mức vật lý

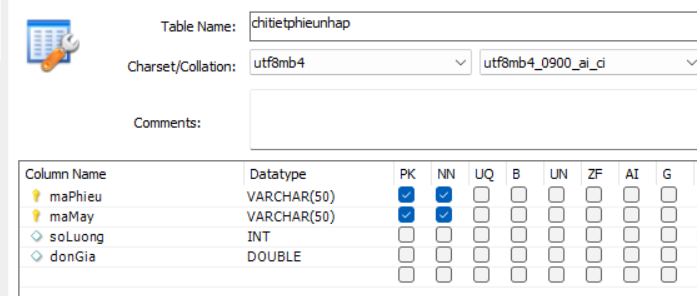
### Thiết kế bảng

****

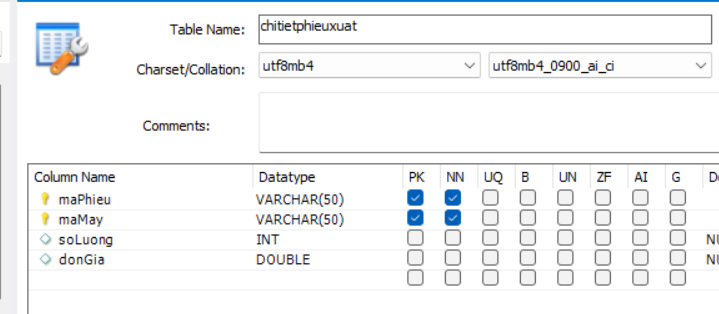
Hình 3.11: Bảng maytinh

**

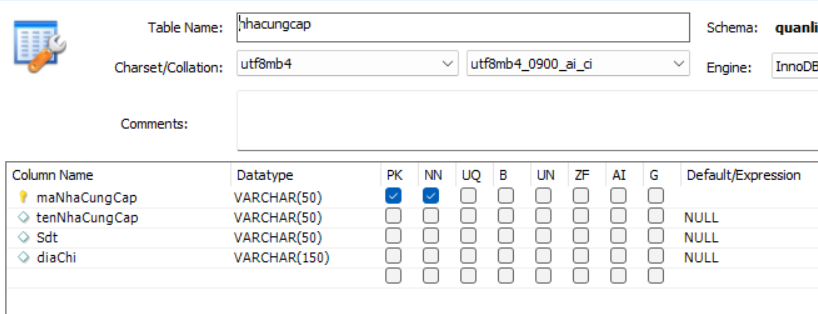
Hình 3. 12: Bảng account

**

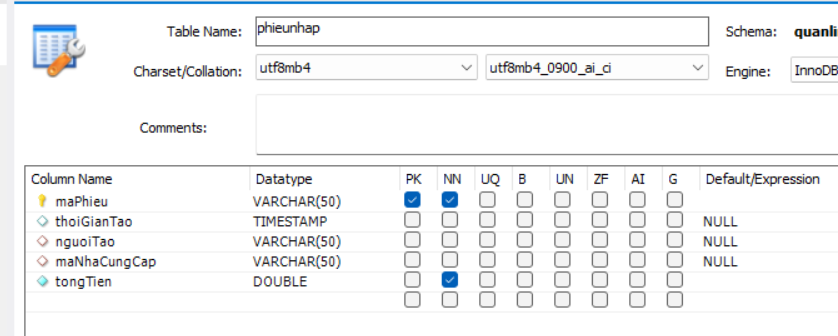
Hình 3.13: Bảng chitietphieunhap



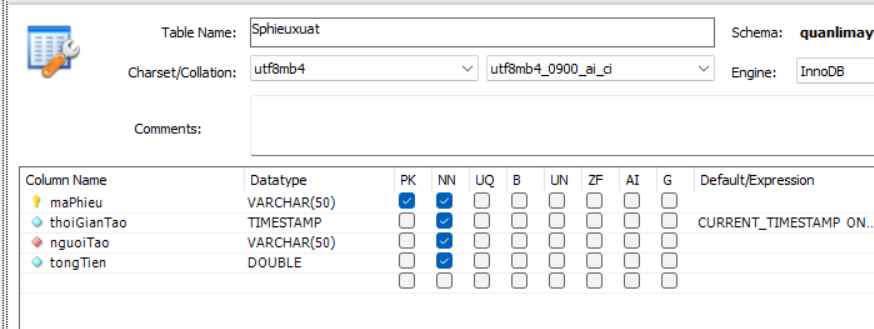
Hình 3.14: Bảng chitietphieuxuat



Hình 3.15: Bảng nhacungcap



Hình 3.16: Bảng phieunhap



Hình 3.17: Bảng phieuxuat

# CHƯƠNG 4. TRIỂN KHAI VÀ KIỂM THỬ PHẦN MỀM

## Lựa chọn công nghệ triển khai

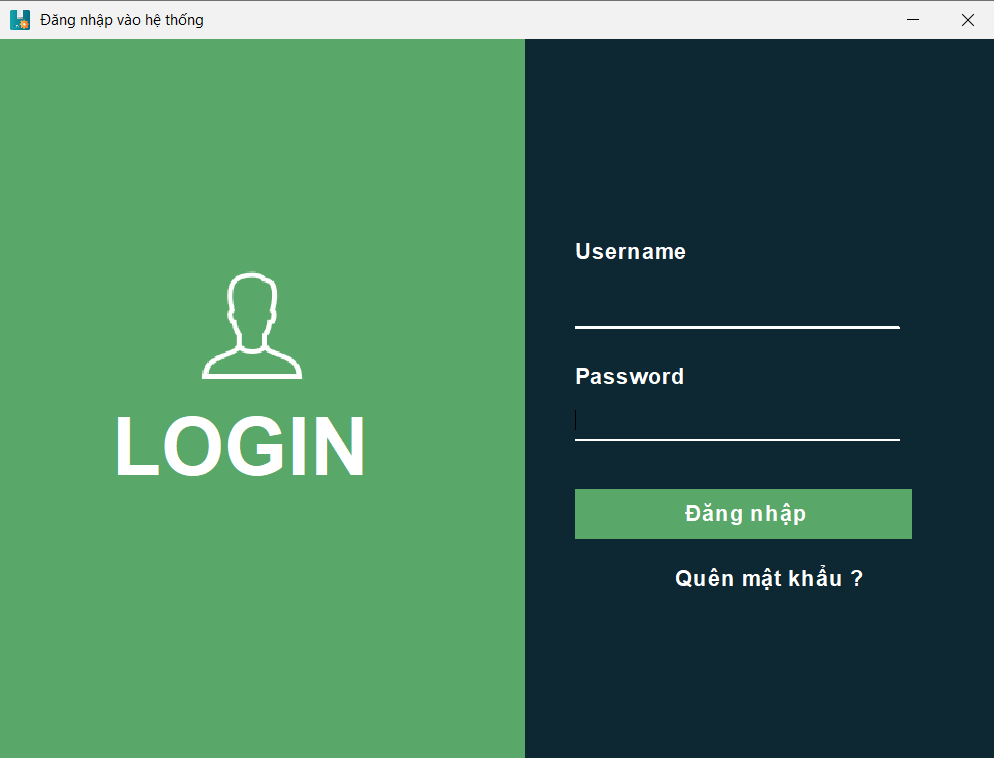
### Giới thiệu ngôn ngữ Java

Java là ngôn ngữ lập trình hướng đối tượng mạnh mẽ và được sử dụng rộng rãi, được biết đến với tính đơn giản, nền tảng độc lập và kho hàng phong phú. Được tạo bởi James Gosling và nhóm của ông tại Sun Microsystems vào giữa những năm 1990, Java đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên toàn cầu. Tính linh hoạt, hỗ trợ cộng đồng mạnh mẽ và hệ sinh thái phong phú khiến nó trở thành lựa chọn tuyệt vời để phát triển các ứng dụng phần mềm khác nhau, bao gồm các chương trình mã hóa như RSA.

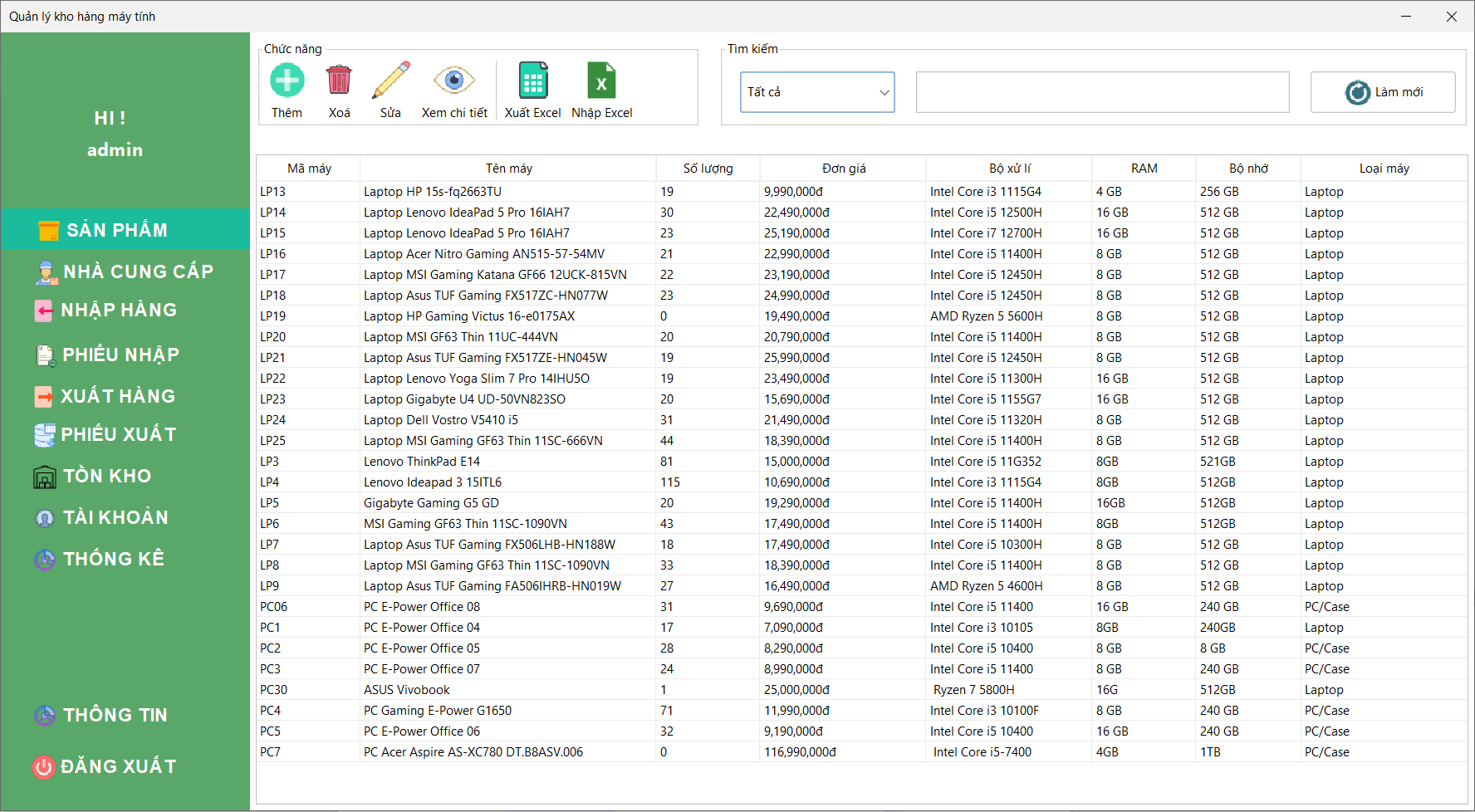
Lý do chọn ngôn ngữ Java: Khi thiết kế chương trình mã hóa RSA, việc chọn ngôn ngữ lập trình phù hợp là rất quan trọng. Dưới đây là một số lý do thuyết phục tại sao Java là một lựa chọn lý tưởng:

* + - 1. **Độc lập nền tảng**: Nguyên tắc "viết một lần, chạy mọi nơi" của Java cho phép các chương trình được viết bằng Java chạy trên bất kỳ nền tảng nào với Máy ảo Java (JVM). Điều này có nghĩa là chương trình mã hóa RSA được phát triển bằng Java sẽ có thể mang theo được và có thể được thực thi trên nhiều hệ điều hành khác nhau, bao gồm Windows, macOS và Linux. Hơn nữa, đặc điểm này cho phép dễ dàng phân phối và triển khai chương trình trên các thiết bị khác nhau.
      2. **Mạnh mẽ và Bảo mật**: Thiết kế của Java nhấn mạnh đến tính bảo mật, làm cho nó trở thành một ngôn ngữ tuyệt vời cho các tác vụ liên quan đến mã hóa. Nó cung cấp các cơ chế tích hợp để quản lý bộ nhớ, xử lý ngoại lệ và phát hiện lỗi thời gian chạy, giảm khả năng xảy ra lỗ hổng và cải thiện độ bền của chương trình. Ngoài ra, kho hàng tiêu chuẩn mở rộng của Java bao gồm các API mật mã tạo thuận lợi cho việc triển khai các thuật toán mã hóa an toàn như RSA.
      3. **Hỗ trợ kho hàng rộng lớn**: Java tự hào có một bộ sưu tập lớn các kho hàng và khung giúp đơn giản hóa các tác vụ lập trình phức tạp. Đối với mã hóa RSA, các kho hàng như Java Cryptography Architecture (JCA) và Bouncy Castle cung cấp các chức năng mã hóa toàn diện, bao gồm tạo khóa, mã hóa và giải mã. Các kho hàng này tuân thủ các giao thức mã hóa tiêu chuẩn, đảm bảo tính tương thích và khả năng tương tác với các hệ thống mật mã khác.

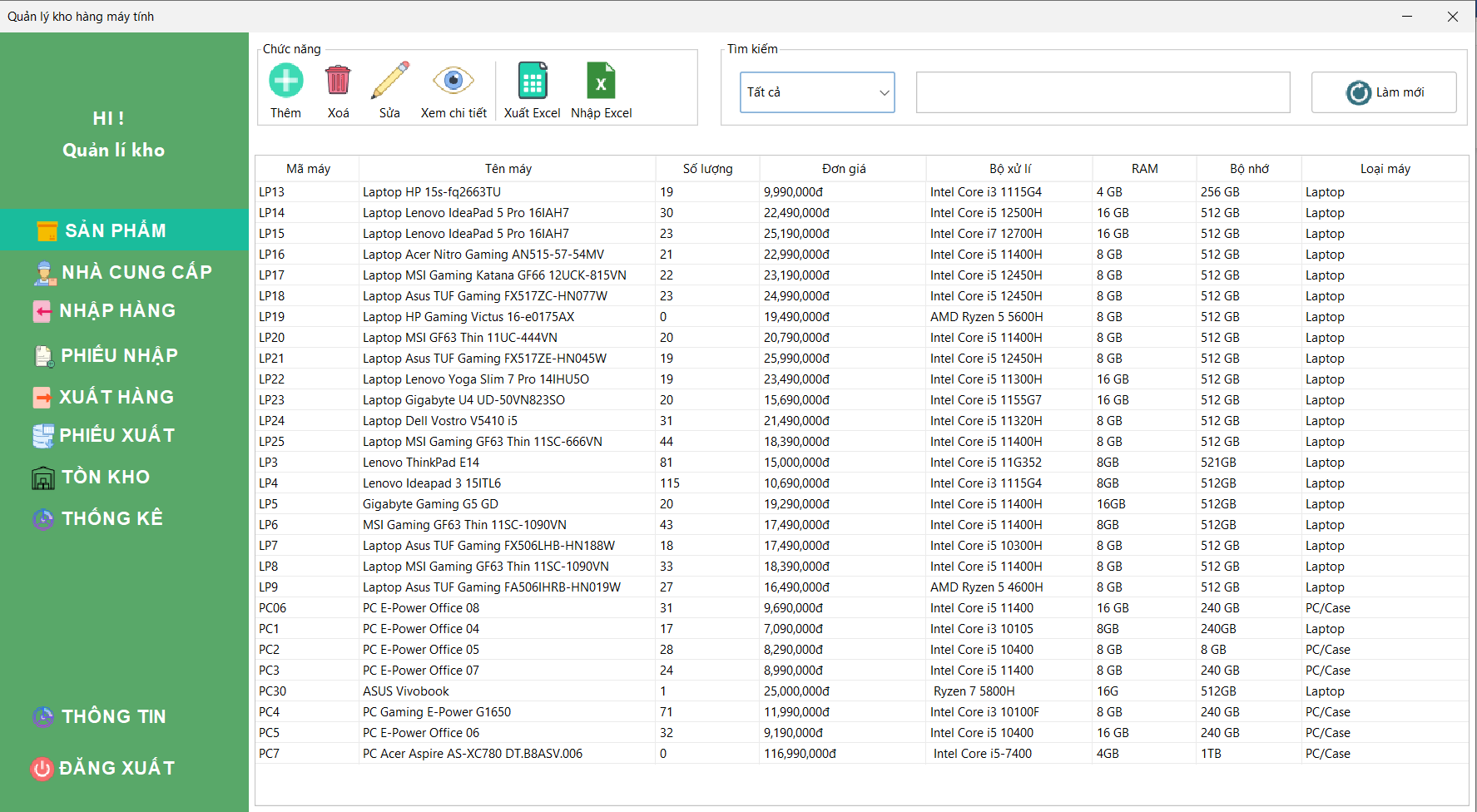
## Lập trình và triển khai phần mềm



Hình 4.1 Giao diện đăng nhập



Hình 4.2 Giao diện đăng nhập với tư cách Admin



Hình 4.3 Giao diện đăng nhập với tư cách quản lí kho

Graphical user interface, text, application

Description automatically generated

Hình 4.4 Giao diện đăng nhập với tư cách nhân viên nhập

Graphical user interface, text

Description automatically generated

Hình 4.5 Giao diện đăng nhập với tư cách nhân viên xuất

Graphical user interface

Description automatically generated

Hình 4.6 Giao diện thông tin cá nhân

Graphical user interface

Description automatically generated

Hình 4.7 Giao diện thay đổi thông tin tài khoản

Graphical user interface

Description automatically generated

Hình 4.8 Giao diện khi nhập mật khẩu mới không hợp lệ

Graphical user interface

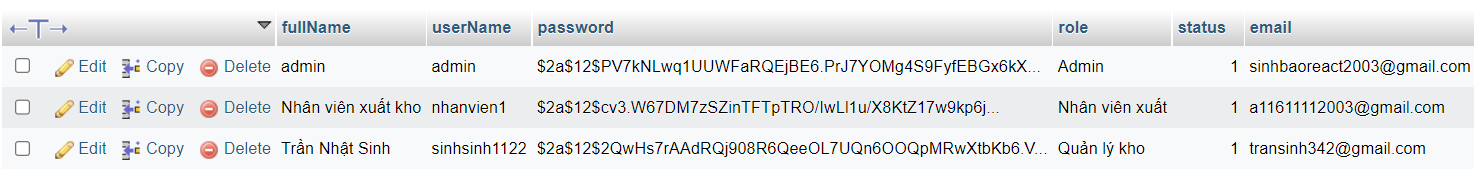
Description automatically generated

Hình 4.9 Giao diện khi nhập mật khẩu cũ sai

Graphical user interface

Description automatically generated

Hình 4.10 Giao diện khi đổi mật khẩu thành công



Hình 4.11 Mật khẩu được mã hóa BcryPassword và lưu vào cơ sở dữ liệu để tăng tính bảo mật

Graphical user interface, application

Description automatically generated

Hình 4.12 Giao diện trang sản phẩm

A screenshot of a computer

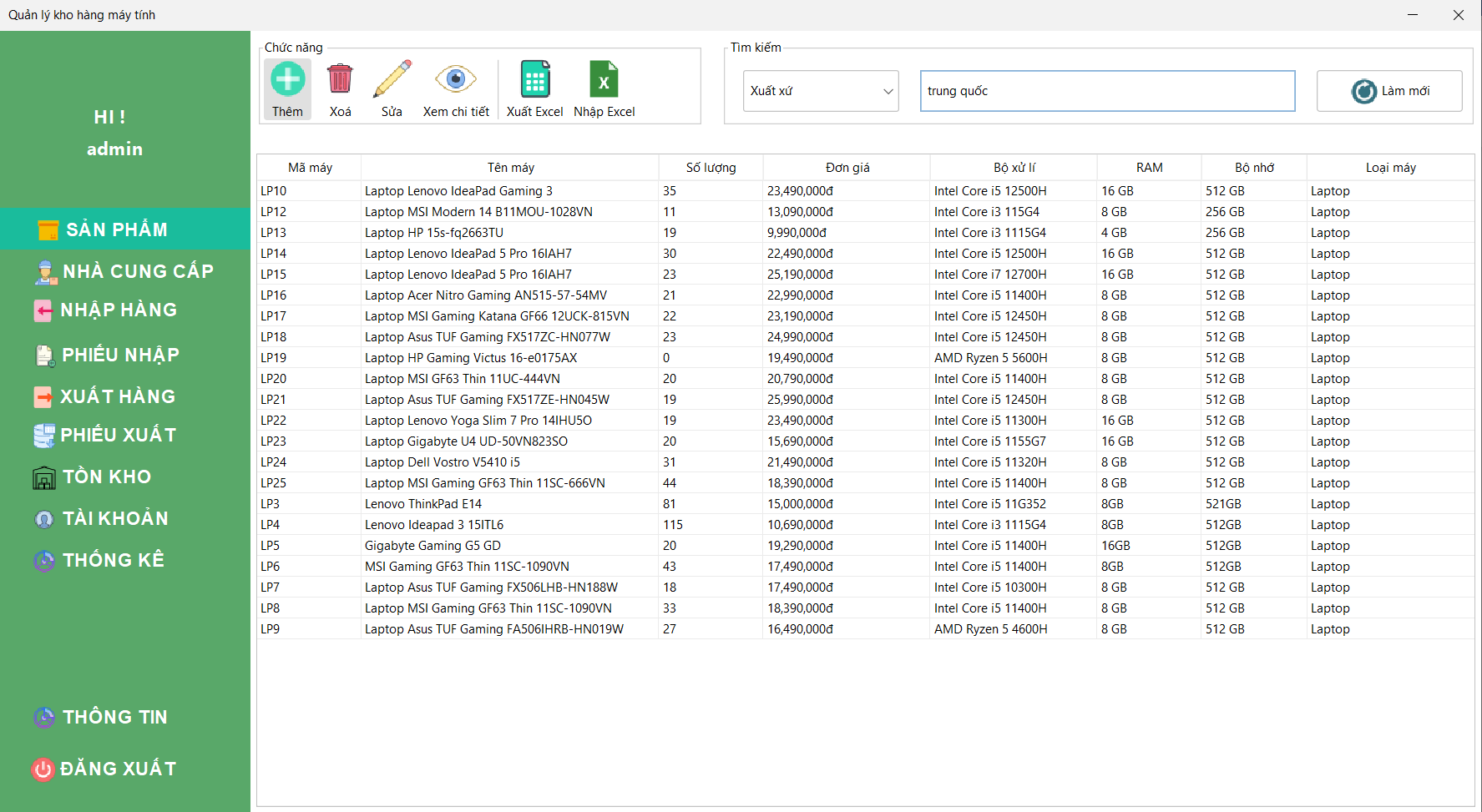
Description automatically generated

Hình 4.13 Thông tin chi tiết sản phẩm laptop

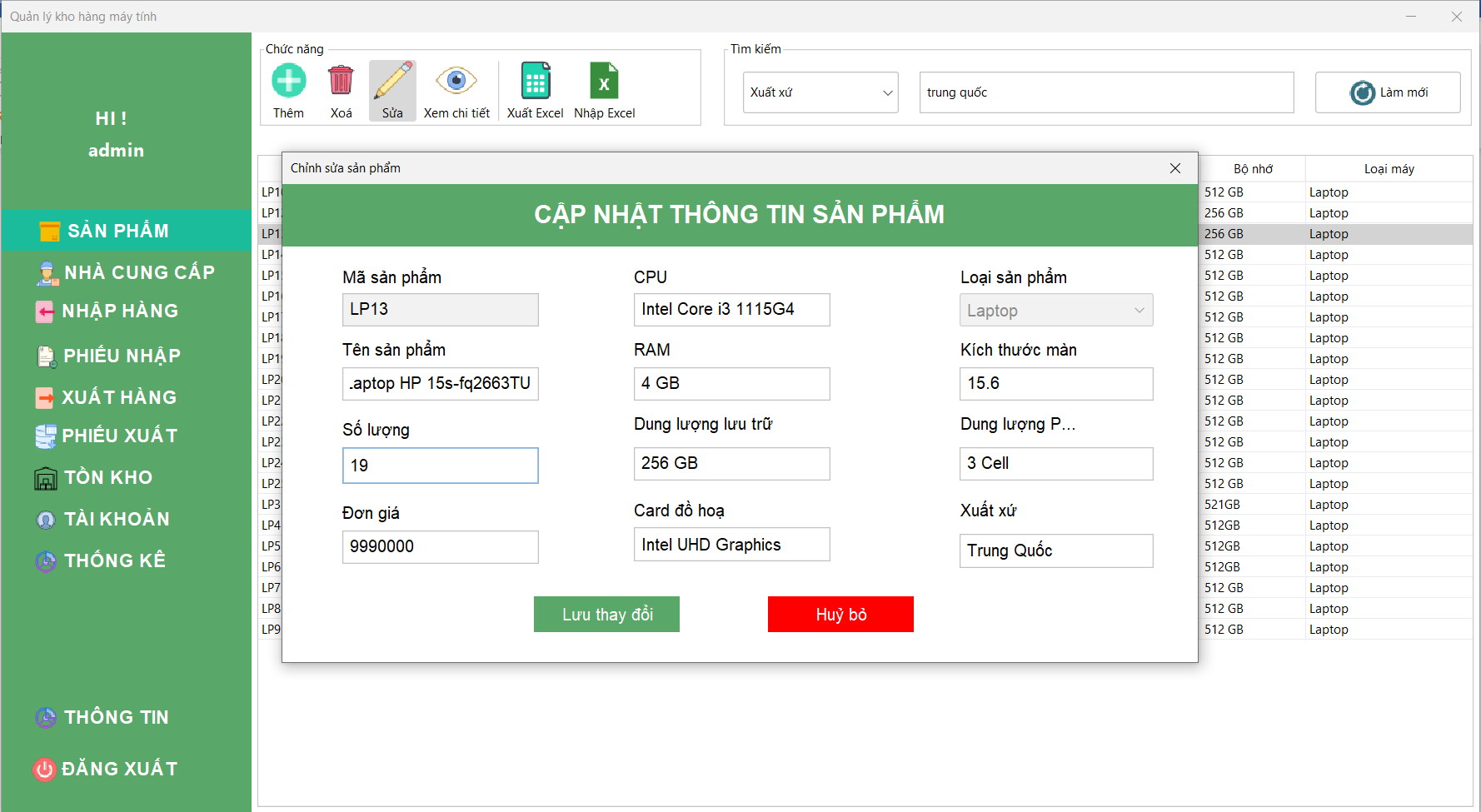
Graphical user interface, application

Description automatically generated

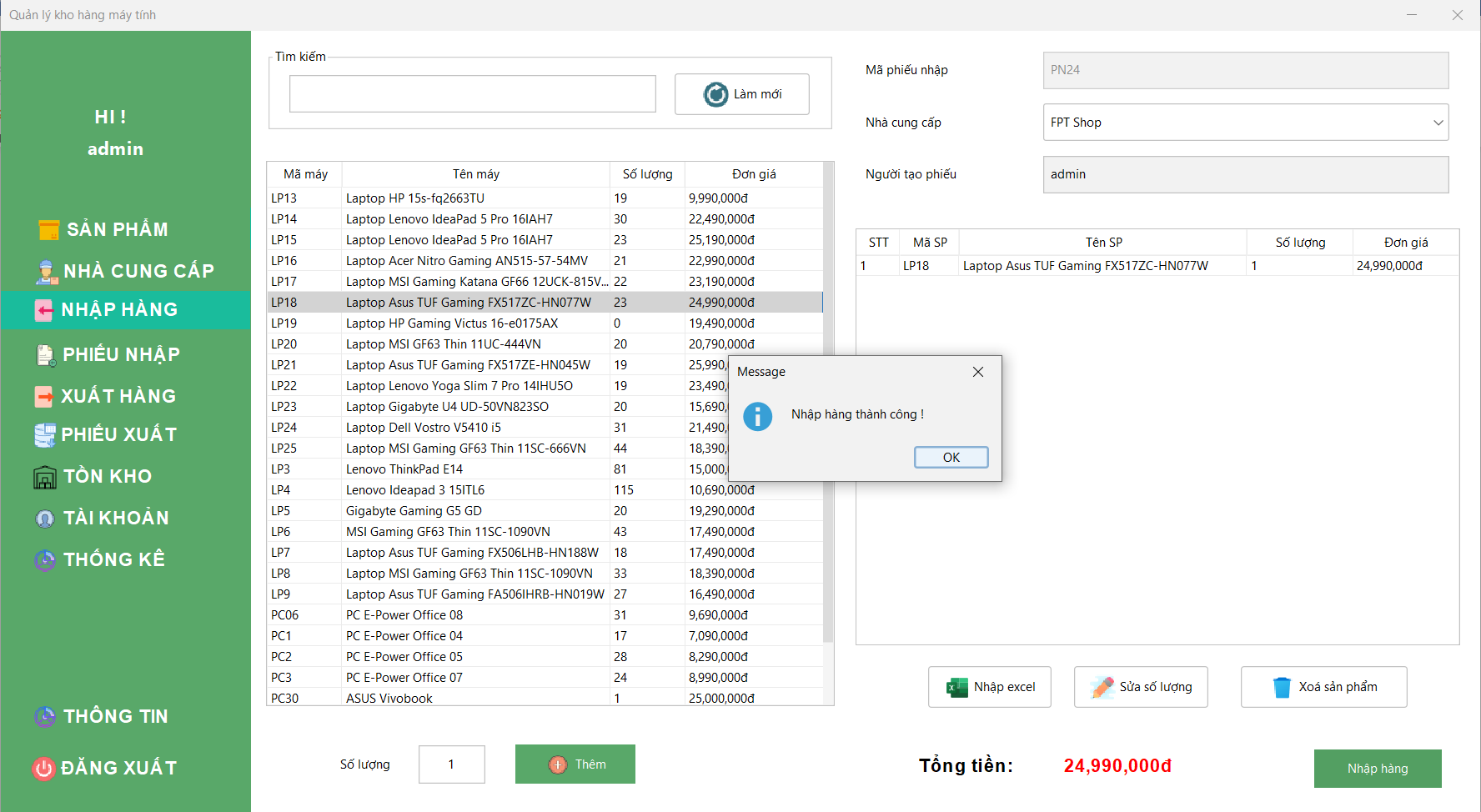
Hình 4.14 Thông tin chi tiết sản phẩm PC



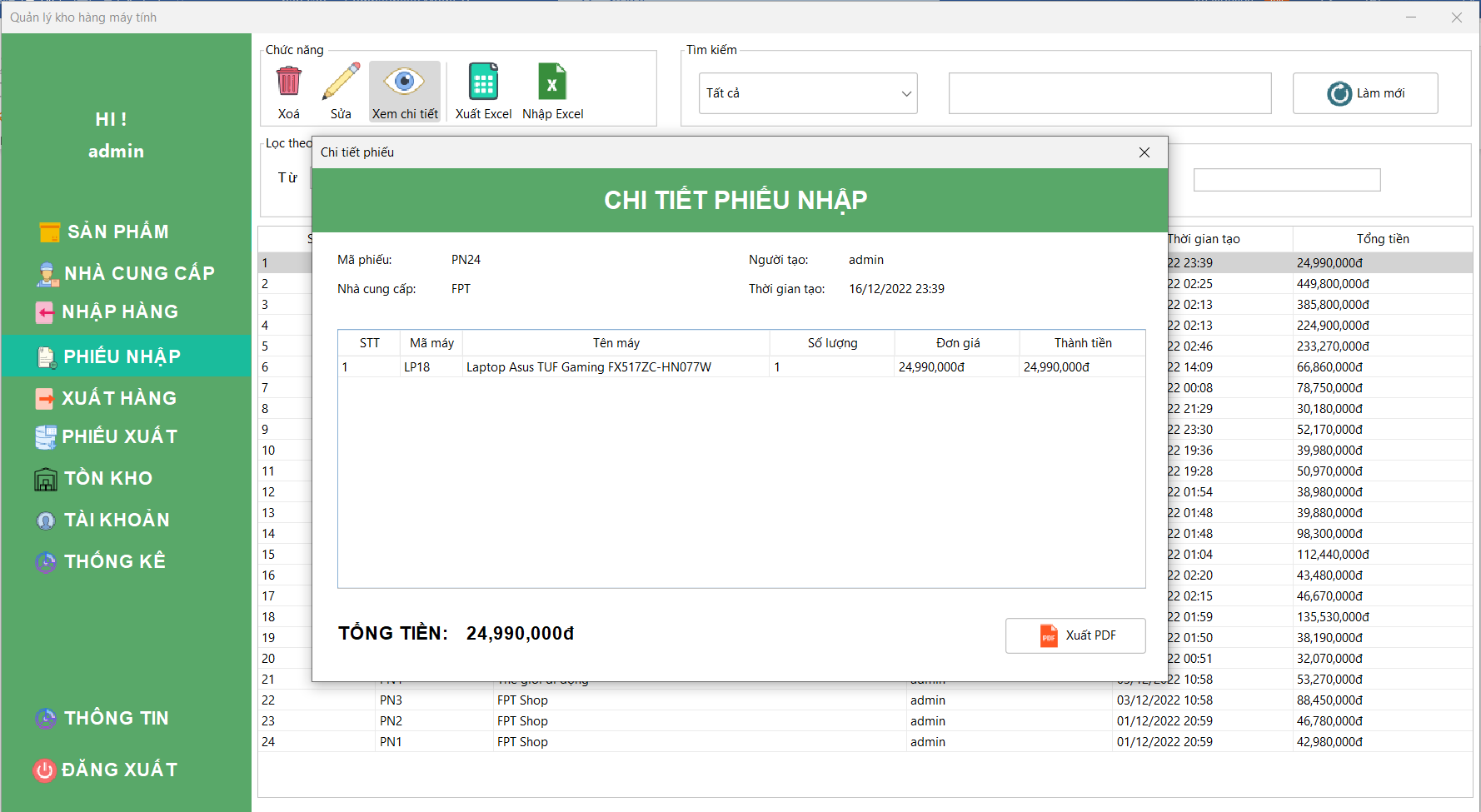
Hình 4.15 Tìm kiếm sản phẩm



Hình 4.16 Giao diện sửa sản phẩm



Hình 4.17 Giao diện tạo phiếu nhập kho



Hình 4.18 Chi tiết phiếu nhập

Graphical user interface, application, table, Excel

Description automatically generated

Hình 4.19 Xuất file excel phiếu nhập

Graphical user interface

Description automatically generated

Hình 4.20 Xuất file PDF phiếu nhập

Graphical user interface, text

Description automatically generated

Hình 4.21 Giao diện tạo phiếu xuất kho

Graphical user interface, application, Word

Description automatically generated

Hình 4.22 Chi tiết phiếu xuất

Graphical user interface, application, table, Excel

Description automatically generated

Hình 4.23 Xuất file excel phiếu xuất

Graphical user interface, text, email

Description automatically generated

Hình 4.24 Xuất file PDF phiếu xuất

Graphical user interface, application

Description automatically generated

Hình 4.25 Giao diện nhập thông tin hàng hóa

Graphical user interface, application, table, Excel

Description automatically generated

Hình 4.26 Giao diện xóa thông tin hàng hóa

Graphical user interface, application, table

Description automatically generated

Hình 4.27 Thống kê hàng hóa theo ngày

Graphical user interface

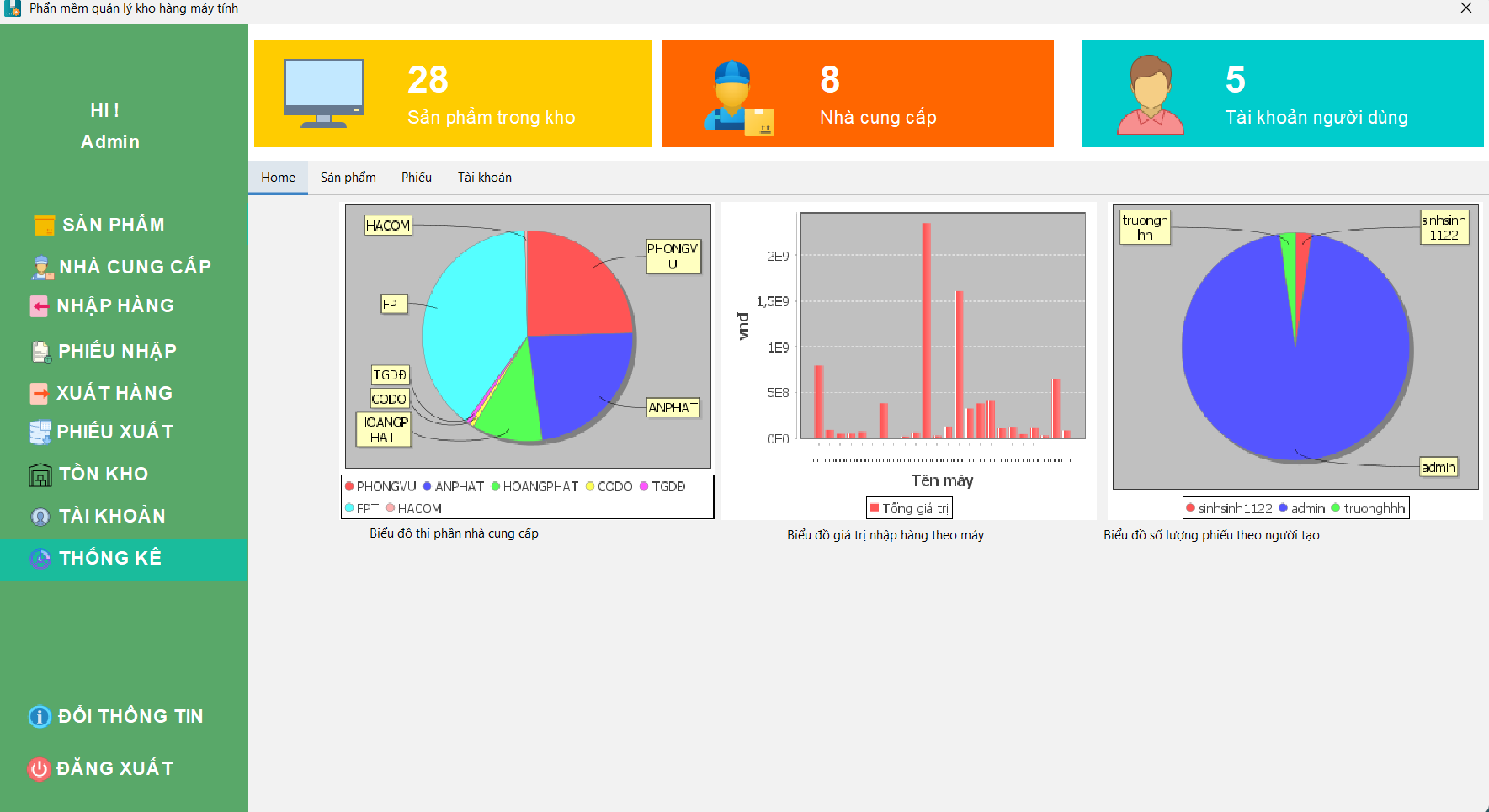
Description automatically generated

Hình 4.28 Giao diện thêm nhân viên

Graphical user interface, text, application

Description automatically generated

Hình 4.29 Giao diện xóa nhân viên



Hình 4.30 Giao diện các biểu đồ

# CHƯƠNG 5. KẾT LUẬN LUẬN VÀ TỔNG KẾT

## Tóm tắt kết quả nghiên cứu

Với sự phát triển của công nghệ, việc phát triển ứng dụng để quản lý kho hàng trở nên quan trọng. Đề tài này được chọn để cung cấp một giải pháp tiện lợi cho việc quản lý các kho hàng. Qua quá trình nghiên cứu, chúng em đã xây dựng được đầy đủ các chức năng như mục tiêu ban đầu đặt ra. Để đề tài có thể nâng cao thêm các tính năng, thì chúng em còn tìm hiểu và đặt ra các mục tiêu để phát triển dự án, giúp sản phẩm có thể phục vụ tốt hơn trong tương lai.

## Đánh giá đối với mục tiêu ban dầu

Hệ thống quản lý kho hàng không chỉ giúp theo dõi tiến độ một cách chi tiết mà còn quản lý nguồn lực hiệu quả. Dự án đã vượt qua các mục tiêu đặt ra từ giai đoạn đầu, mang lại giải pháp quản lý kho hàng toàn diện cho các kho hàng đang được thực hiện trong trường học. Việc tích hợp các tính năng quan trọng đã cung cấp cơ sở hạ tầng mạnh mẽ cho quản lý tiến trình xây dựng. Điều này không chỉ thúc đẩy sự hiệu quả trong theo dõi tiến độ mà còn tối ưu hóa việc quản lý nguồn lực, từ nhân sự đến nguồn tài chính.

## Đề xuất hướng phát triển tương lai

Sau quá trình nghiên cứu và xây dựng dự án, chúng em đã hoàn thành được những mục tiêu nhất định, nhưng dự án còn những hướng phát triển tiềm năng hơn, giúp phù hợp với từng thời điểm mà công nghệ đang phát triển hằng ngày và những chức năng có thể xem xét để tích hợp vào trong dự án như:

Tích hợp trí tuệ nhân tạo vào trong việc dự đoán các rủi ro trong quá trình thực thi, áp dụng các công nghệ mới để cải tiến dự án, như mở ra các diễn đàn có trong dự án giúp các trường học có thể trao đổi các ý tưởng cũng như các vướng mắc trong việc thực hiện dự án.

# TÀI LIỆU THAM KHẢO

[1] J a v a Swing và Java AWT. <https://hocjava.com/swing-va-awt-trong-java/#1_AWT_va_Java_Swing_la_gi>.

[2] Java Swing – Swing trong Java. http://hocjava.com/java-swing-swing-trong-java/