BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**ĐỀ TÀI: QUẢN LÍ KHO HÀNG**

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Sinh Viên | Họ và Tên | Ngày Sinh | Lớp |
| 1771020256 | Vũ Tuấn Hiệp | 13/08/2003 | CNTT17.01 |

### 

**Hà Nội, năm 2024**

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**ĐỀ TÀI: QUẢN LÍ KHO HÀNG**

**PHÂN HỆ:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1771020256 | Vũ Tuấn Hiệp | 13/08/2003 |  |  |

### 

### CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2

**Hà Nội, năm 2024**

**LỜI NÓI ĐẦU**

Trong thời đại thương mại điện tử bùng nổ, việc quản lý hàng tồn kho hiệu quả là yếu tố quyết định thành công của doanh nghiệp. Ứng dụng quản lý kho hàng ra đời nhằm giúp các nhà bán hàng online giải quyết bài toán quản lý kho phức tạp. Với giao diện trực quan, dễ sử dụng, ứng dụng sẽ giúp người dùng theo dõi đơn hàng, kiểm soát tồn kho, tối ưu hóa quy trình vận hành và nâng cao hiệu quả kinh doanh."

Hiện nay, có một số doanh nghiệp vừa và nhỏ đang áp dụng các ứng dụng quản lý kho hàng để cải thiện quy trình kinh doanh của họ. Những doanh nghiệp này thường gặp khó khăn trong việc theo dõi số lượng hàng hóa, quản lý đơn hàng và xử lý các vấn đề phát sinh trong kho.

Nhờ vào công nghệ, các ứng dụng này không chỉ giúp họ dễ dàng kiểm soát lượng hàng tồn kho mà còn hỗ trợ trong việc dự đoán nhu cầu, từ đó giảm thiểu tình trạng thiếu hàng hoặc thừa hàng. Ngoài ra, việc tích hợp với các nền tảng thương mại điện tử giúp tự động hóa quy trình bán hàng, từ đó tiết kiệm thời gian và chi phí cho doanh nghiệp.

Bên cạnh đó, nhiều ứng dụng còn cung cấp các tính năng phân tích dữ liệu, giúp doanh nghiệp nắm bắt xu hướng tiêu thụ và điều chỉnh chiến lược kinh doanh phù hợp. Điều này không chỉ góp phần nâng cao hiệu quả hoạt động mà còn tạo ra lợi thế cạnh tranh trên thị trường.

Với sự phát triển không ngừng của công nghệ, việc áp dụng các giải pháp quản lý kho hiện đại sẽ là một bước đi cần thiết cho các doanh nghiệp trong thời đại số hóa này.

MỤC LỤC

[CHƯƠNG 1. MÔ TẢ BÀI TOÁN 6](#_Toc183294434)

[1.1. Giới thiệu 6](#_Toc183294435)

[1.2. Chức năng chính 6](#_Toc183294436)

[1.3. Yêu cầu phi chức năng 7](#_Toc183294437)

[CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG 8](#_Toc183294438)

[2.1. Phân tích yêu cầu: 8](#_Toc183294439)

[2.2. Thiết kế hệ thống: 10](#_Toc183294440)

[KẾT LUẬN 23](#_Toc183294441)

# CHƯƠNG 1. MÔ TẢ BÀI TOÁN

## 1.1. Giới thiệu

Ứng dụng quản lý kho hàng là một phần mềm được thiết kế để giúp các doanh nghiệp quản lý hiệu quả hoạt động của kho hàng, từ việc theo dõi hàng hóa, nhập xuất kho đến việc kiểm kê và báo cáo. Ứng dụng này đóng vai trò quan trọng trong việc tối ưu hóa quy trình làm việc, giảm thiểu sai sót và tăng năng suất.

## 1.2. Chức năng chính

Quản lý hàng hóa:

* Thêm, sửa, xóa thông tin hàng hóa (tên, mã hàng, số lượng, đơn vị tính, giá thành, nhà cung cấp,...)
* Tìm kiếm hàng hóa nhanh chóng theo nhiều tiêu chí (mã hàng, tên hàng, loại hàng,...)
* Quản lý tồn kho: theo dõi số lượng hàng tồn kho thực tế, cảnh báo hàng sắp hết, hàng quá hạn.

Quản lý nhập xuất kho:

* Tạo phiếu nhập kho, phiếu xuất kho.
* Quản lý chi tiết các lần nhập xuất kho: ngày nhập/xuất, số lượng, giá trị.
* In báo cáo nhập xuất kho theo yêu cầu.

Kiểm kê:

* Thực hiện kiểm kê định kỳ hoặc đột xuất.
* So sánh số lượng hàng tồn kho thực tế với số liệu trên hệ thống.
* Tạo báo cáo chênh lệch.

Báo cáo:

* Tạo các báo cáo về tình hình kinh doanh, tồn kho, nhập xuất hàng.
* Các báo cáo có thể được tùy chỉnh theo yêu cầu của người dùng.
* Quản lý nhà cung cấp, khách hàng: Quản lý thông tin nhà cung cấp, khách hàng. Theo dõi lịch sử giao dịch với từng đối tác.

## 1.3. Yêu cầu phi chức năng

* Dễ sử dụng: Giao diện console cần rõ ràng, dễ hiểu và dễ sử dụng.
* Hiệu năng: Ứng dụng cần hoạt động nhanh chóng và hiệu quả, ngay cả khi có nhiều công việc.
* Độ tin cậy: Dữ liệu công việc cần được lưu trữ an toàn và không bị mất mát.

# CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG

## 2.1. Phân tích yêu cầu:

**2.1.1Xác định người dùng:**

* Các doanh nghiệp vừa và nhỏ (SMEs) có một số đặc điểm và nhu cầu quản lý kinh doanh đặc thù:
* Nguồn lực hạn chế: SMEs thường có tài chính, nhân lực và cơ sở hạ tầng giới hạn, đòi hỏi giải pháp đơn giản, tiết kiệm chi phí và tối ưu hóa quy trình để phát triển bền vững.
* Quản lý hiệu quả: Cần tối ưu hóa hoạt động để tránh tồn kho quá nhiều hoặc thiếu hàng, quản lý kho và đơn hàng hiệu quả để giảm chi phí và đáp ứng nhu cầu thị trường đúng lúc.
* Theo dõi đơn hàng và khách hàng: Các công cụ quản lý đơn hàng, thông tin khách hàng và giao dịch bán hàng giúp duy trì sự hài lòng và tạo niềm tin từ khách hàng.

**Thu thập yêu cầu:**

* Dựa trên mô tả bài toán, ta đã xác định được các chức năng chính (thêm, hiển thị, chỉnh sửa, xóa,in báo cáo) và yêu cầu phi chức năng (dễ sử dụng, hiệu năng, độ tin cậy) của ứng dụng.

**Phân tích yêu cầu:**

* Lớp HangHoa (Sản phẩm)
* Mục đích: Đại diện cho một mặt hàng hoặc sản phẩm trong kho.
* Các thuộc tính:
* maHangHoa: Mã sản phẩm (định danh duy nhất).
* tenHangHoa: Tên sản phẩm.
* soLuong: Số lượng sản phẩm có trong kho.
* gia: Giá của sản phẩm.
* Phương thức:
* getMaHangHoa(): Lấy mã sản phẩm.
* hienThiThongTin(): Hiển thị thông tin sản phẩm.
* Các setter và getter khác để thao tác với các thuộc tính như tên, số lượng, và giá.
* Lớp QuanLyHangHoa
* Mục đích: Quản lý các thao tác với danh sách hàng hóa (thêm, sửa, xóa, tìm kiếm, lưu trữ vào và đọc từ file).
* Các thuộc tính:
* danhSachHangHoa: Danh sách chứa các đối tượng HangHoa.
* scanner: Để nhập thông tin từ người dùng.
* TEN\_FILE: Tên file chứa dữ liệu sản phẩm (danhSachHangHoa.dat).
* Các phương thức:
* File operations:
* luuVaoFile(): Lưu danh sách sản phẩm vào file.
* docTuFile(): Đọc dữ liệu sản phẩm từ file.
* docFileThuCong(): Tải dữ liệu từ file và hiển thị thông báo.
* Quản lý sản phẩm:
* themHangHoa(): Thêm sản phẩm mới vào danh sách.
* hienThiHangHoa(): Hiển thị danh sách tất cả các sản phẩm.
* timKiemHangHoaTheoMa(): Tìm kiếm sản phẩm theo mã.
* timKiemHangHoaTheoTen(): Tìm kiếm sản phẩm theo tên.
* capNhatHangHoa(): Cập nhật thông tin của sản phẩm.
* xoaHangHoaTheoMa(): Xóa sản phẩm theo mã.
* xoaHangHoaTheoTen(): Xóa sản phẩm theo tên.
* kiemTraHangHoaHetHan(): Chức năng kiểm tra hết hạn (chưa triển khai).
* Menu:
* menu(): Hiển thị menu điều khiển cho người dùng để lựa chọn các thao tác như thêm, tìm kiếm, cập nhật, xóa, và kiểm tra hàng hóa.

## 2.2. Thiết kế hệ thống:

**Biểu đồ lớp cho MVC**

A diagram of a company

Description automatically generated

Hình 1

Kiến trúc MVC trong hệ thống quản lý kho hàng:

* Model:
* HangHoa: Đại diện cho một mặt hàng, chứa thông tin về mã hàng, tên hàng, số lượng tồn kho, loại hàng.
* PhieuNhapKho: Đại diện cho một phiếu nhập kho, chứa thông tin về mã phiếu, ngày nhập, danh sách hàng hóa nhập và số lượng nhập.
* HoaDon: Đại diện cho một hóa đơn xuất kho, chứa thông tin về mã hóa đơn, mã hàng, số lượng xuất, ngày xuất.
* View:
* Các phương thức toString() trong các lớp: Định dạng dữ liệu để hiển thị trên màn hình, tạo ra giao diện người dùng đơn giản.
* Controller:
* Các phương thức trong lớp QuanLyKhoHang: Thực hiện các nghiệp vụ như thêm hàng, sửa hàng, xuất hàng, nhập hàng, ... Đây là nơi điều khiển luồng thực thi của chương trình, tương tác với Model và View.
* Cách thức hoạt động:
* Người dùng tương tác: Người dùng nhập thông tin qua giao diện console (View).
* Controller xử lý: Controller nhận yêu cầu từ người dùng, gọi các phương thức tương ứng trong Model để cập nhật dữ liệu.
* Model cập nhật: Model cập nhật dữ liệu trong các đối tượng HangHoa, PhieuNhapKho, HoaDon.
* View hiển thị: Controller gọi các phương thức toString() để hiển thị kết quả ra màn hình.

|  |
| --- |
| **Dựa trên kiến trúc MVC đã chọn, ta có thể xác định các lớp sau:**   * **Lớp QuanLyKho:** * Các thuộc tính chính: * kho: Một HashMap<String, Integer> lưu trữ thông tin về mã hàng hóa (key) và số lượng (value). * scanner: Một đối tượng Scanner để nhập dữ liệu từ bàn phím. * TEN\_FILE: Một hằng số chứa tên file để lưu dữ liệu. * Các phương thức chính: * docTuFile(): Đọc dữ liệu từ file và cập nhật vào HashMap kho. * luuVaoFile(): Lưu dữ liệu từ HashMap kho vào file. * nhapKho(): Thêm một mặt hàng vào kho hoặc tăng số lượng của một mặt hàng đã tồn tại. * xuatKho(): Giảm số lượng của một mặt hàng đã tồn tại trong kho. * hienThiTonKho(): Hiển thị danh sách các mặt hàng và số lượng hiện có trong kho. * hienThiTonKhoTuFile(): Đọc dữ liệu từ file trước khi hiển thị, đảm bảo hiển thị thông tin mới nhất. * kiemKe(): Cho phép người dùng kiểm kê số lượng thực tế của một mặt hàng và cập nhật vào hệ thống. * **Lớp QuanLyHangHoa:** * Thuộc tính: * danhSachHangHoa: Một danh sách (List) lưu trữ các đối tượng HangHoa. Đây là nơi chứa tất cả thông tin về các mặt hàng trong kho. * Phương thức: * themHangHoa(): Cho phép người dùng nhập thông tin một mặt hàng mới.Thêm mặt hàng mới vào danh sách danhSachHangHoa. * xoaHangHoa(): Xóa một mặt hàng khỏi danh sách dựa trên mã hàng hóa. * suaHangHoa(): Cho phép người dùng cập nhật thông tin của một mặt hàng đã tồn tại. * hienThiDanhSachHangHoa(): Hiển thị danh sách tất cả các mặt hàng trong kho. * timHangHoaTheoMa(): Tìm kiếm một mặt hàng dựa trên mã hàng hóa. * xemBaoCaoTonKho(): Hiển thị một báo cáo tổng quan về số lượng và giá trị hàng hóa trong kho. * ghiFile(): Lưu toàn bộ danh sách danhSachHangHoa vào một file (ví dụ: file văn bản, file CSV, hoặc file nhị phân). * docFile(): Đọc dữ liệu từ file và cập nhật vào danh sách danhSachHangHoa. * xuatHangHoa(): Xuất hàng hóa ra ngoài, có thể là xuất sang một hệ thống khác hoặc in ra giấy. * taoPhieuNhapKho(): Tạo một phiếu nhập kho mới, bao gồm thông tin về ngày nhập, nhà cung cấp, danh sách hàng hóa nhập. * **Lớp NhanVien:** * Lớp này đại diện cho một nhân viên trong hệ thống quản lý nhân viên. * Thuộc tính: * maNhanVien: Mã nhân viên (kiểu String). * tenNhanVien: Tên nhân viên (kiểu String). * chucVu: Chức vụ của nhân viên (kiểu String). * Phương thức: * NhanVien(String maNhanVien, String tenNhanVien, String chucVu): Đây là constructor khởi tạo một đối tượng NhanVien mới với các thông tin như mã nhân viên, tên nhân viên, và chức vụ. * hienThiThongTin(): Phương thức này in ra thông tin của nhân viên theo định dạng bảng. Cụ thể là mã nhân viên, tên nhân viên, và chức vụ. * getMaNhanVien(): Phương thức này trả về mã nhân viên. * **Lớp QuanLyNhanVien:** * Lớp này quản lý danh sách nhân viên và các hoạt động quản lý nhân viên, bao gồm thêm, xóa, hiển thị, và lưu trữ dữ liệu vào file. * Thuộc tính: * danhSachNhanVien: Danh sách chứa các đối tượng NhanVien (kiểu ArrayList<NhanVien>), lưu trữ các nhân viên hiện có trong hệ thống. * scanner: Đối tượng Scanner để nhập dữ liệu từ người dùng. * TEN\_FILE: Tên file lưu trữ dữ liệu nhân viên (kiểu String), trong trường hợp này là "nhanvien.dat". * Phương thức: * QuanLyNhanVien(): Constructor của lớp, khi tạo đối tượng này sẽ tự động gọi phương thức docTuFile() để đọc dữ liệu từ file nếu có. * luuVaoFile(): Phương thức này dùng để lưu danh sách nhân viên vào file nhanvien.dat bằng cách sử dụng ObjectOutputStream. * docTuFile(): Phương thức này đọc dữ liệu từ file nhanvien.dat và chuyển nó thành một danh sách các đối tượng NhanVien sử dụng ObjectInputStream. * docFileThuCong(): Phương thức này tải dữ liệu từ file và hiển thị thông báo "Dữ liệu từ file đã được tải lên bộ nhớ". * themNhanVien(): Phương thức này thêm một nhân viên mới vào danh sách, yêu cầu người dùng nhập thông tin nhân viên, sau đó lưu lại vào file. * hienThiDanhSachNhanVien(): Phương thức này hiển thị danh sách tất cả các nhân viên đã được thêm vào, bao gồm mã nhân viên, tên nhân viên, và chức vụ. * xoaNhanVien(): Phương thức này xóa nhân viên khỏi danh sách theo mã nhân viên đã nhập. Sau khi xóa, danh sách được lưu lại vào file. * hienThiNhanVienTuFile(): Phương thức này đọc dữ liệu từ file và hiển thị danh sách nhân viên. * menu(): Phương thức này cung cấp giao diện người dùng cho các lựa chọn như thêm nhân viên, xóa nhân viên, hiển thị danh sách nhân viên và thoát chương trình. Phương thức này sẽ liên tục lặp lại yêu cầu lựa chọn cho đến khi người dùng chọn thoát. * **Mối quan hệ giữa các lớp:** * QuanLyKho sẽ sử dụng QuanLyHangHoa để quản lý thông tin về hàng hóa trong kho, như thêm, xóa hoặc sửa sản phẩm, và kiểm tra số lượng hàng hóa. * QuanLyKho cũng sẽ sử dụng QuanLyNhanVien để quản lý nhân viên trong kho, như phân công công việc hoặc giám sát hoạt động của nhân viên trong kho. * QuanLyHangHoa sẽ chứa thông tin chi tiết về các sản phẩm hàng hóa và có thể được sử dụng bởi QuanLyKho để thao tác với hàng hóa. * QuanLyNhanVien có thể có chức năng theo dõi nhân viên nào thực hiện công việc gì trong kho và có thể báo cáo về kho. |

**Thiết kế giao diện:**

* Hiển thị menu chính với các lựa chọn:
  + 1. Quản lý hàng hoá
    2. Quản lý nhập/xuất kho
    3. Quản lý nhân viên kho
    4. Thoát
* Sử dụng các thông báo rõ ràng để hướng dẫn người dùng nhập liệu và hiển thị kết quả.

1. Nếu người dùng nhập lựa chọn không hợp lệ:

[LỖI] Lựa chọn không hợp lệ! Vui lòng nhập số từ 1 đến 4.

1. Khi người dùng chọn 1. Quản lý hàng hóa:

-----------------------------------------

QUẢN LÝ HÀNG HÓA

1. Thêm hàng hóa mới

2. Sửa thông tin hàng hóa

3. Xóa hàng hóa

4. Hiển thị danh sách hàng hóa

5. Quay lại menu chính

-----------------------------------------

Vui lòng chọn một tùy chọn (1-5):

1. Khi người dùng chọn 2. Quản lý nhập/xuất kho:

-----------------------------------------

QUẢN LÝ NHẬP/XUẤT KHO

1. Thêm phiếu nhập kho

2. Thêm phiếu xuất kho

3. Hiển thị danh sách phiếu nhập/xuất

4. Quay lại menu chính

-----------------------------------------

Vui lòng chọn một tùy chọn (1-4):

1. Khi người dùng chọn 3. Quản lý nhân viên kho:

-----------------------------------------

QUẢN LÝ NHÂN VIÊN KHO

1. Thêm nhân viên mới

2. Sửa thông tin nhân viên

3. Xóa nhân viên

4. Hiển thị danh sách nhân viên

5. Quay lại menu chính

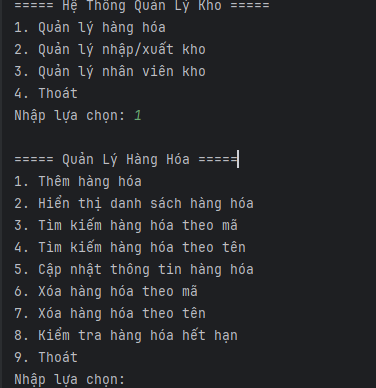
-----------------------------------------

Vui lòng chọn một tùy chọn (1-5):

1. Khi người dùng chọn 4. Thoát:

Bạn có chắc chắn muốn thoát? (y/n):

+ Minh họa



Hình 2

**2.3. Triển khai:**

* **Viết code:** Sử dụng Java để cài đặt các class trong mô hình MVC, đọc/ghi file, xử lý dữ liệu và hiển thị giao diện console.
  + Cài đặt GoogleSheetHandler sử dụng thư viện API của Google Sheets (ví dụ: Google API Client Library for Java).
  + Xử lý xác thực để ứng dụng có quyền truy cập vào Google Sheet của bạn.
  + Thay đổi code trong TaskController để sử dụng GoogleSheetHandler cho việc đọc/ghi dữ liệu.
* **Kiểm thử:** Viết các test case để kiểm tra các chức năng của ứng dụng, đảm bảo ứng dụng hoạt động đúng theo yêu cầu.

**2.4. Vận hành và bảo trì:**

* Cài đặt và triển khai:
  + Hướng dẫn người dùng cách chạy ứng dụng từ console (ví dụ: java -jar TodoApp.jar).
  + Ngoài hướng dẫn chạy ứng dụng, cần hướng dẫn người dùng cách tạo Google Sheet và chia sẻ quyền truy cập cho ứng dụng.
  + Có thể cần cài đặt thêm các thư viện cần thiết cho việc kết nối với Google Sheets.
* Bảo trì: Sửa lỗi phát sinh, cập nhật chức năng mới (nếu có) và cải thiện hiệu năng của ứng dụng.

**CHƯƠNG 3. KẾT QUẢ THỰC HIỆN**

**3.1. Công nghệ đã sử dụng**

* Ngôn ngữ lập trình: Java
* Công cụ: IntelliJ IDEA
* Thư viện: Scanner, List, ArrayList

**3.2. Tiến độ thực hiện**

Link github tới dự án: https://github.com/tuanhiep23/detai4-VuTuanHiep.git

**Hướng dẫn các bước đã thực hiện:**

**B1. Tạo dự án mới:**

* Mở IntelliJ IDEA.
* Chọn "Create New Project".
* Chọn "Java" làm ngôn ngữ lập trình.
* Chọn JDK phù hợp (ví dụ: JDK 11 hoặc mới hơn).
* Nhập tên dự án (ví dụ: "TodoApp").
* Chọn vị trí lưu trữ dự án.
* Nhấn "Finish" để tạo dự án.

**B2. Tạo các package:**

* Trong cửa sổ "Project", click chuột phải vào thư mục "src".
* Chọn "New" -> "Package".
* Tạo các package sau:
  + model: chứa các lớp Task và TaskList.
  + view: chứa lớp TaskView.
  + controller: chứa lớp TaskController.
  + handler: chứa lớp GoogleSheetHandler.

**B3. Tạo các lớp:**

* Trong mỗi package, click chuột phải và chọn "New" -> "Java Class" để tạo các lớp tương ứng.
* Cài đặt các thuộc tính và phương thức cho từng lớp dựa trên thiết kế đã phân tích.

**B4. Cài đặt thư viện Google Sheets API:**

* Mở file pom.xml (nếu bạn đang sử dụng Maven) hoặc build.gradle (nếu bạn đang sử dụng Gradle).
* Thêm dependency cho thư viện Google API Client Library for Java. Ví dụ, trong Maven:
* IntelliJ IDEA sẽ tự động tải về và thêm thư viện vào dự án.

**B5. Viết code:**

* Bắt đầu viết code cho từng lớp, thực hiện các chức năng của ứng dụng:
  + **Task:** cài đặt các thuộc tính và phương thức cơ bản.
  + **TaskList:** cài đặt các phương thức để quản lý danh sách công việc.
  + **GoogleSheetHandler:** cài đặt các phương thức để kết nối, đọc, ghi, cập nhật, xóa dữ liệu trên Google Sheet.
  + **TaskView:** cài đặt các phương thức để hiển thị giao diện console và tương tác với người dùng.
  + **TaskController:** cài đặt logic xử lý yêu cầu của người dùng, điều phối các lớp khác để hoàn thành các chức năng.

**B6. Xử lý xác thực Google Sheets:**

* Tạo tài khoản dịch vụ trên Google Cloud Platform và tải xuống file JSON chứa khóa API.
* Tham khảo tài liệu của Google Sheets API để biết cách sử dụng file JSON này để xác thực ứng dụng của bạn.
* Cài đặt code xử lý xác thực trong GoogleSheetHandler.

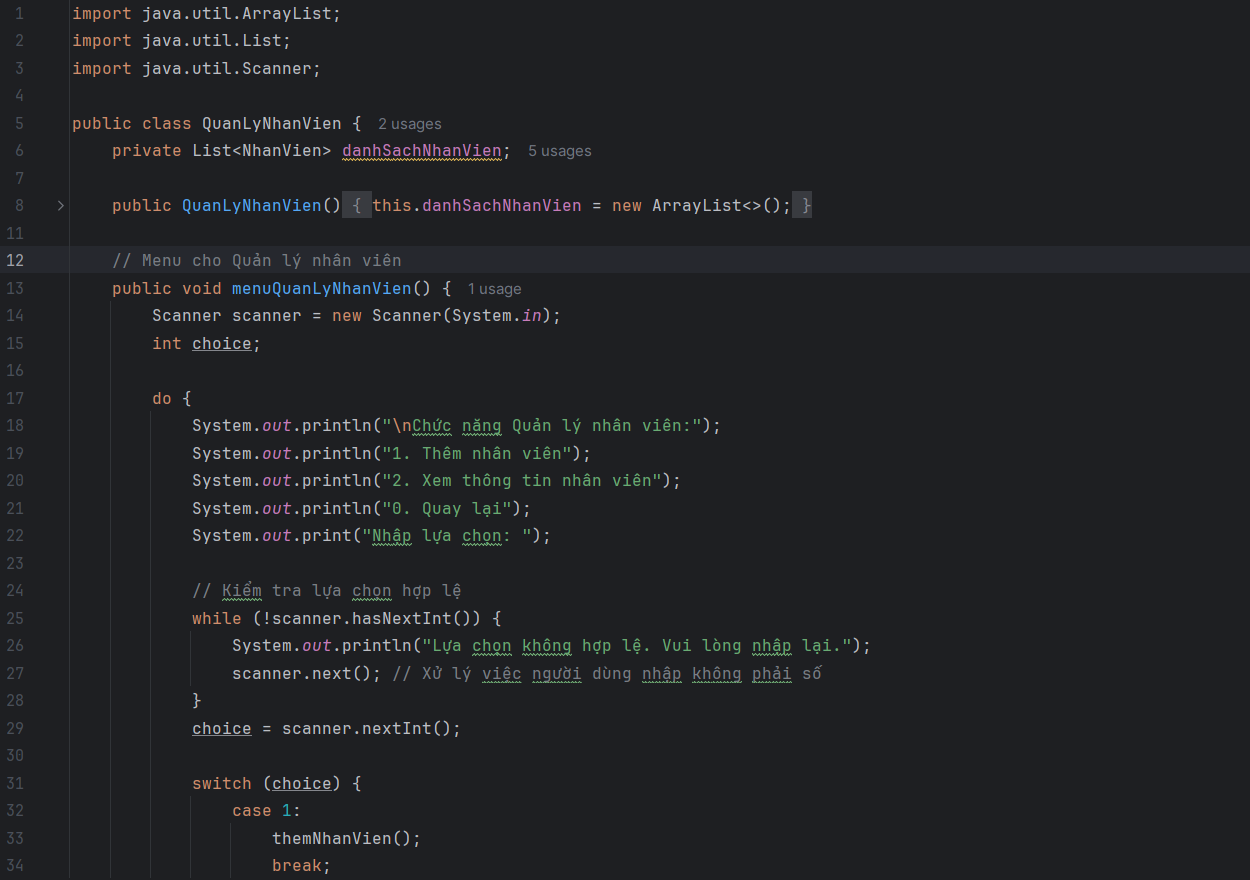
**B7. Chạy và kiểm thử:**

* Chạy ứng dụng từ IntelliJ IDEA bằng cách click chuột phải vào lớp Main (hoặc lớp chứa phương thức main) và chọn **"Run 'Main.main()'"**.
* Kiểm tra các chức năng của ứng dụng, sửa lỗi và hoàn thiện code.

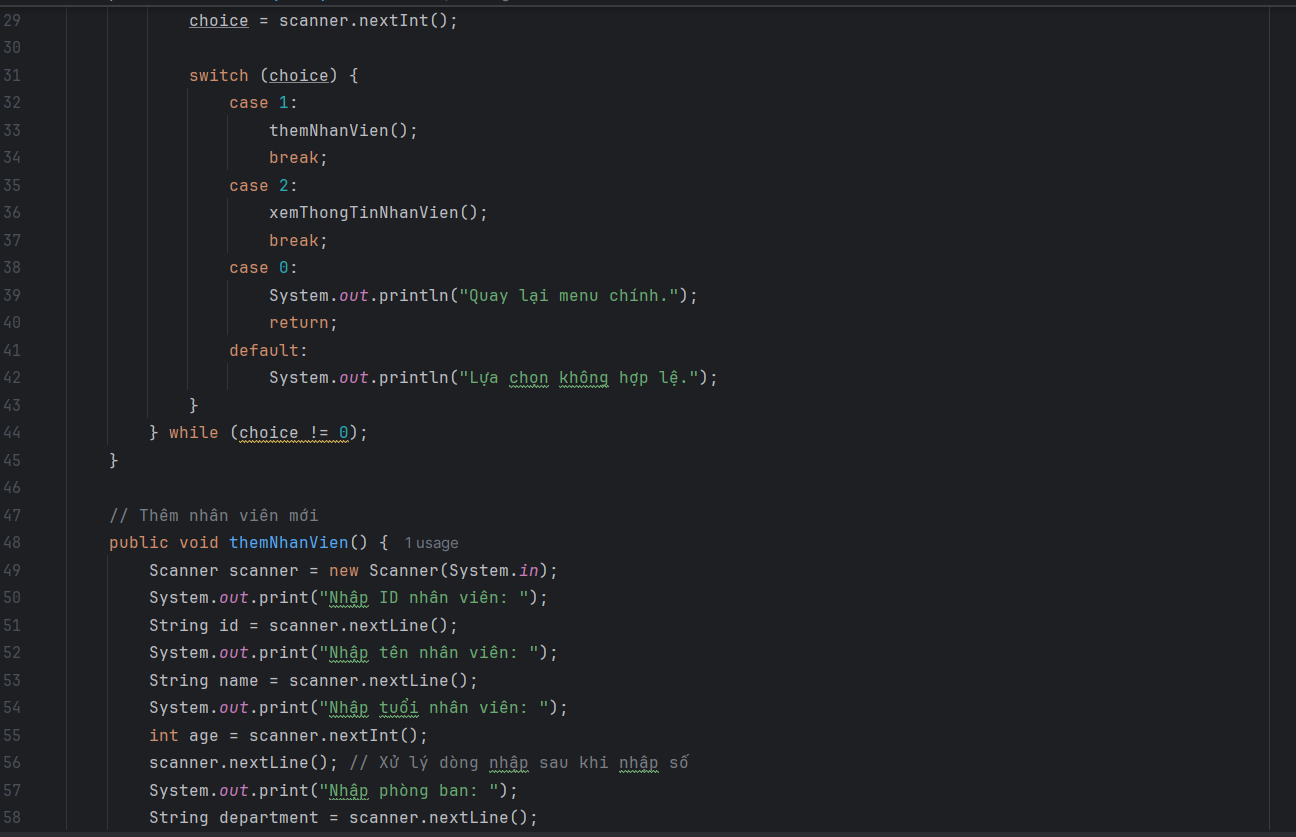
**B8. Triển khai (tùy chọn):**

* Nếu muốn đóng gói ứng dụng thành file JAR để dễ dàng chia sẻ và chạy trên các máy khác, bạn có thể sử dụng chức năng "Build Artifacts" của IntelliJ IDEA.

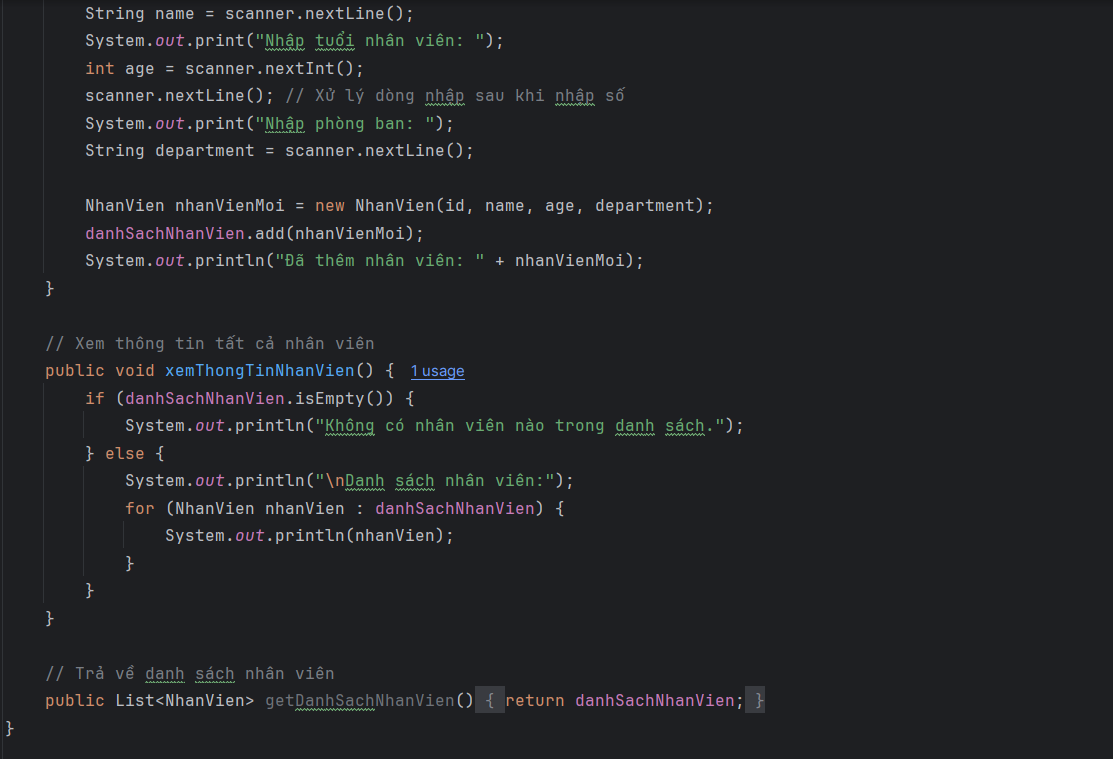
**3.3. Hình ảnh sản phẩm**

****

Hình 3



Hình 4



Hình 5

# KẾT LUẬN

+ Ưu điểm của quản lý kho bằng Java

Quản lý kho bằng Java mang lại nhiều ưu điểm nổi bật. Đầu tiên, Java là ngôn ngữ lập trình đa nền tảng, cho phép ứng dụng chạy trên nhiều hệ điều hành khác nhau, từ Windows đến Linux và macOS. Điều này giúp các doanh nghiệp dễ dàng triển khai và sử dụng ứng dụng quản lý kho mà không gặp phải rào cản về hệ thống. Thêm vào đó, Java hỗ trợ xây dựng ứng dụng quy mô lớn với khả năng mở rộng linh hoạt, phù hợp với sự phát triển của doanh nghiệp. Cùng với kho thư viện phong phú và tính năng bảo mật cao, Java tạo điều kiện thuận lợi cho việc phát triển các giải pháp quản lý kho hiệu quả.

+ Nhược điểm của quản lý kho bằng Java

Mặc dù có nhiều lợi thế, việc quản lý kho bằng Java cũng gặp phải một số nhược điểm. Một trong những vấn đề chính là hiệu suất, khi ứng dụng Java có thể chạy chậm hơn so với các ngôn ngữ như C++ do phụ thuộc vào Java Virtual Machine (JVM). Ngoài ra, ứng dụng Java thường có kích thước lớn hơn vì cần bao gồm JVM và các thư viện liên quan, điều này có thể gây khó khăn trong việc triển khai trên các thiết bị hạn chế về tài nguyên. Hơn nữa, đối với người mới bắt đầu, việc nắm bắt các khái niệm lập trình hướng đối tượng và xử lý ngoại lệ trong Java có thể là một thách thức.

+ Hướng phát triển trong tương lai

Trong tương lai, các ứng dụng quản lý kho bằng Java có thể hướng tới việc tích hợp các công nghệ mới để nâng cao hiệu quả hoạt động. Việc sử dụng trí tuệ nhân tạo và máy học để dự đoán nhu cầu hàng hóa và tối ưu hóa quy trình tồn kho sẽ là một xu hướng quan trọng. Đồng thời, phát triển ứng dụng di động và chuyển đổi lên nền tảng đám mây cũng sẽ giúp cải thiện tính linh hoạt và khả năng mở rộng của hệ thống. Cuối cùng, việc tích hợp Internet of Things (IoT) để theo dõi hàng hóa theo thời gian thực sẽ mang lại sự chính xác và hiệu quả cao hơn trong quản lý kho, giúp doanh nghiệp nâng cao khả năng cạnh tranh trong thị trường.

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. Nguyễn Hồng Sơn (2007), *Giáo trình hệ thống Mạng máy tính CCNA* (Semester 1), NXB Lao động xã hội.
2. Phạm Quốc Hùng (2017), *Đề cương bài giảng Mạng máy tính*, Đại học SPKT Hưng Yên.
3. James F. Kurose and Keith W. Ross (2013), *Computer Networking: A top-down approach sixth Edition*, Pearson Education.