Inventory Management System using Java Collections

Aim: The goal of this project is to design and implement an efficient Inventory Management System using **Java Collection Framework** to manage products, track transactions, handle undo operations, and generate inventory statistics.

Objectives:

- Ensure unique product identification using SKU
- Support sorting and searching of products using different criteria
- Maintain transaction history
- Implement undo functionality for quantity updates
- Detect and manage low stock alerts
- Demonstrate practical use of advanced data structures

This project emphasizes **data structure selection, performance awareness, and clean object-oriented design**, which are critical skills for a Java developer role.

Setup Instructions

Prerequisites:

- Java JDK 8 or higher
- VS Code / IntelliJ IDEA
- Basic knowledge of Java and OOP

Folder Structure:

InventoryManagementSystem/

├── src/

```
│   │   ├── Product.java
│   │   └── Transaction.java
│   ├── comparators/
│   │   ├── PriceComparator.java
│   │   ├── ValueComparator.java
│   │   └── NameComparator.java
│   ├── collections/
│       └── InventoryManagementSystem.java
├── docs/
│   └── diagrams.png
└── README.md
```

Code Structure Explanation

Product.java

- Represents a product in inventory
- Implements Comparable<Product> for **natural sorting by SKU**
- Contains:
  - SKU
  - Name
  - Category
  - Price
  - Quantity

Transaction.java

- Stores **previous state** for undo operations
- Fields:
    - SKU
    - Old Quantity

Comparator Classes

- PriceComparator → Sort by product price
- ValueComparator → Sort by inventory value (price × quantity)
- NameComparator → Sort alphabetically by name

Each comparator follows **Single Responsibility Principle**.

InventoryManagementSystem.java

Core business logic:

- Add product
- Update quantity
- Undo updates
- Display sorted products
- Low stock alerts
- Transaction history
- Inventory statistics

Main.java

- Menu-driven console interface
- Handles user input and delegates logic to InventoryManagementSystem

Collection Types Used & Why

HashSet<Product>

Purpose: Store unique products

Reason:

- Prevents duplicate SKUs
- Average time complexity: **O(1)**

TreeSet<Product>

Purpose: Sorted product view

Reason:

- Automatically sorted using Comparable
- Time complexity: **O(log n)**

LinkedList<String>

Purpose: Transaction history

Reason:

- Frequent insertions at the beginning
- Efficient add/remove operations

Stack<Transaction>

Purpose: Undo functionality

Reason:

- LIFO behavior matches undo operations exactly

Queue<Product> (LinkedList)

Purpose: Low stock alerts

Reason:

- FIFO processing of alerts
- Simple and predictable alert handling

Sorting Strategies

- **Natural Sorting:** SKU using Comparable
- **Custom Sorting:**

    - Price → PriceComparator
    - Value → ValueComparator
    - Name → NameComparator