CST-239 Milestone 3: Inventory Manager and Shopping Cart Report

**Student Name:** Batossa Bakouma  
**Course:** CST-239: Object-Oriented Programming  
**Instructor:** Komal Chhibber  
**Date:** May 25, 2025  
**Title:** Milestone 3 – Inventory Manager and Shopping Cart Implementation

**1. Project Overview**

This milestone extends the StoreFront application by implementing two key modules: InventoryManager and ShoppingCart. The goal is to demonstrate object-oriented design principles such as abstraction, inheritance, composition, and interface implementation in a Java console-based shopping system.

**2. UML Class Diagrams**

The following UML diagrams were created to represent the design of the system:

**🔹 2.1 Salable Product Hierarchy**

* Includes SalableProduct, Weapon, Armor, and HealthItem
* Weapon implements Comparable<Weapon> for alphabetical sorting

A diagram of a system

AI-generated content may be incorrect.

*Salable Product Hierarchy UML Diagram*

**🔹 2.2 InventoryManager**

* Manages store inventory
* Adds, removes, and lists items

A diagram of a product

AI-generated content may be incorrect.

*InventoryManager UML Diagram*

**🔹 2.3 ShoppingCart**

* Manages cart operations (add, remove, empty, view)

A screenshot of a computer

AI-generated content may be incorrect.  
 *ShoppingCart UML Diagram*

**🔹 2.4 StoreFrontApp Integration**

* Shows composition of StoreFrontApplication with InventoryManager and ShoppingCart

A screenshot of a diagram

AI-generated content may be incorrect.  
 *StoreFrontApp UML Diagram*

**3. Application Design Features**

**✅ InventoryManager:**

* Initializes inventory when the store starts
* Supports purchasing and canceling (restocking) products
* Assigns internal product IDs

**✅ ShoppingCart:**

* Stores purchased items
* Supports viewing and emptying the cart
* Fully integrated with InventoryManager

**✅ Weapon Comparison:**

* Weapon implements Comparable<Weapon>
* Case-insensitive alphabetical sorting by name

**4. Flowchart of User Interaction**

The flowchart below shows the logic of user interaction with the store, including InventoryManager and ShoppingCart operations.

A diagram of a company

AI-generated content may be incorrect.

*Flowchart of Game User Interaction*

**5. Java Code Implementation**

All functionality was implemented using Java with a clean, modular structure:

* **Packages:**
  + edu.gcu.storefront.model – product-related classes
  + edu.gcu.storefront.service – Inventory and Cart logic
  + edu.gcu.storefront – application entry point
* **JavaDoc:**
  + All public classes and methods are documented
  + JavaDoc HTML files generated in /docs using the javadoc tool

A screenshot of a computer

AI-generated content may be incorrect.

*Screenshot of folder structure*

**6. Application Demonstration**

The console-based UI allows the user to interact with the store using a menu. The following actions are supported:

* List available products
* Purchase product (adds to cart, reduces inventory)
* Cancel purchase (removes from cart, restocks)
* View cart
* Empty cart
* Exit

***The outputs are demos in the Screencast Video***

**7. Conclusion**

This milestone demonstrates how key object-oriented principles , abstraction, inheritance, composition, and interfaces, are applied in practice. The result is a maintainable and extensible Java application that models a functional inventory and cart system.

**8. Screencast Video**

🎥 Watch the full demonstration (UML, code walkthrough, and app execution):  
<https://www.loom.com/share/64555883e5de4822a8aecca73db763f6>

**9. GitHub Repository (Optional)**

📁 Full source code is available at:  
<https://github.com/bebakouma/cst-239-milestone3-storefront>