**CST-239 Milestone 2 Assignment Report**

**Cover Page**

**CST-239 Final Assignment Report  
Store Front Application: Design, Implementation, and Demonstration  
Batossa Bakouma  
May 2025  
Grand Canyon University  
CST-239: Object-Oriented Programming  
Instructor: Komal Chhibber**

**Assignment Summary**

This project is a console-based Java application simulating a store front system where users can browse, purchase, and cancel transactions for various types of salable products such as weapons, armor, and health items. The application was designed and implemented using object-oriented programming principles including abstraction, inheritance, encapsulation, and polymorphism.

**Technologies Used**

* Java 11
* JavaDoc for code documentation
* GitHub for version control and public sharing
* PowerShell for compiling and running the application
* Loom for screencast recording

**UML Class Diagram**

The UML class diagram illustrates the structure of the application:

* SalableProduct is an abstract superclass defining common fields and behavior.
* Weapon, Armor, and HealthItem extend SalableProduct and introduce additional attributes.
* InventoryManager manages a collection of products using a map.
* StoreFrontApplication handles all user interactions.

A diagram of a company

AI-generated content may be incorrect.

**Figure 1: UML Class Diagram**

**Flowchart**

The flowchart represents the user interaction with the console-based application. It covers menu navigation, input prompts, validation, product purchasing, cancellation, and termination of the program.

A diagram of a company

AI-generated content may be incorrect.

**Figure 2: Application Flowchart**

**Code Implementation Summary**

**SalableProduct.java**

An abstract class that holds the basic attributes (name, description, price, quantity) and shared methods purchase() and cancel().

Weapon.java, Armor.java, HealthItem.java

Each subclass extends SalableProduct and includes a specific field: damage, defense, or heal respectively.

**InventoryManager.java**

Manages product inventory using a Map<Integer, SalableProduct>. It includes methods to list, buy, and cancel products.

StoreFrontApplication.java

Provides the main menu, reads user input, and calls the appropriate methods on the InventoryManager. It handles invalid input and loops until the user exits.

**Documentation (JavaDoc)**

Each class, method, and key field includes JavaDoc comments. The documentation was generated with the following command:

javadoc -d docs -sourcepath src\main\java -subpackages edu.gcu.storefront

The HTML output is located in the docs/ folder and can be opened via index.html.\

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 3: Screenshot of JavaDoc in Browser**

**Screencast Video**

You can watch a full demonstration of the project, including UML, flowchart, code, and functionality, at the following link:

**🔗** [**https://www.loom.com/share/67aa3a6805d541f58a79bb3768094f2e**](https://www.loom.com/share/67aa3a6805d541f58a79bb3768094f2e)

**GitHub Repository (Optional)**

The full source code is also available on GitHub:

**🔗** [**https://github.com/bebakouma/cst-239-milestone2-storefront**](https://github.com/bebakouma/cst-239-milestone2-storefront)