**Maiza Falcon Rojas**

**February 29, 2024**

**Assignment: Milestone 5**

**Class: CST-239**

Github Link: https://github.com/maiza02/Milestone-5.git

Loom Link: <https://www.loom.com/share/f25459d9521a4066842da288c1a4caaf?sid=52e0efe1-6daa-4667-9d51-35143f4062a8>

<https://www.loom.com/share/c59fb40fb096419caf1c6349acc80ae8?sid=96007e20-e668-4e13-b4ea-9404e2cf888a>

**Computer Specs:**

Device name LAPTOP-SGHATDL7

Processor Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz 1.50 GHz

Installed RAM 8.00 GB (7.78 GB usable)

Device ID F1D3C4C5-7B38-41DB-A506-F5A23E5B853F

Product ID 00325-96703-32487-AAOEM

System type 64-bit operating system, x64-based processor

Pen and touch Touch support with 10 touch points

**Programming Conventions:**

**Naming Conventions:**

Class names use CamelCase (e.g., SalableProduct, InventoryManager).

Variable and method names use camelCase (e.g., inventory, getQuantity()).

Constants use uppercase with underscores (e.g., ASCENDING, DESCENDING).

**Indentation and Formatting:**

Indentation is consistent and typically four spaces.

Opening and closing braces follow the same line style (e.g., class InventoryManager {).

**Comments:**

Well-commented code for complex or non-trivial sections.

Javadoc comments for classes and methods describing their purpose and usage.

**Exception Handling:**

Proper exception handling using try-catch blocks.

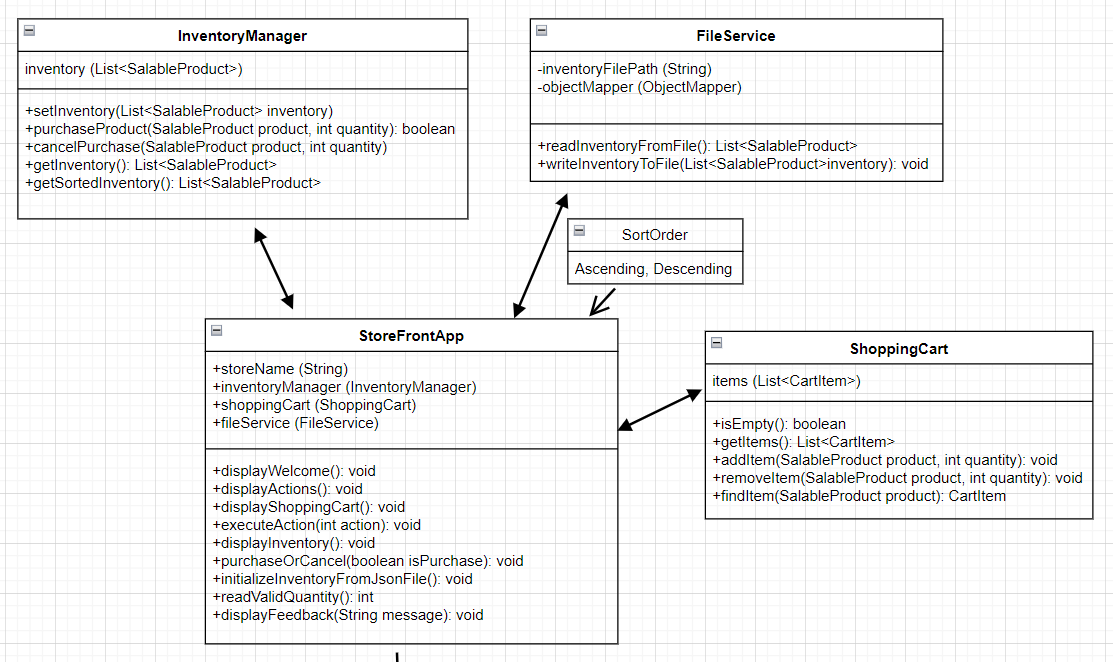
Meaningful error messages to aid debugging and user understanding.

**Consistency:**

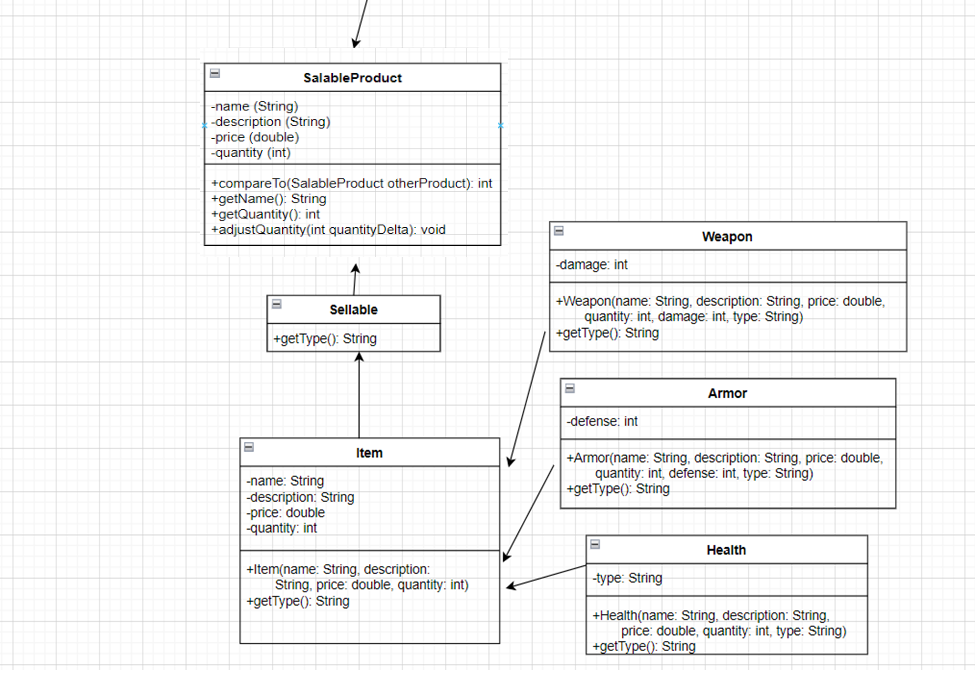
Consistent use of data structures (e.g., ArrayList for inventory).

Consistent use of access modifiers (e.g., private fields with public getters).

**Figure 1: UML Diagram 1**



**Figure 2: UML Diagram 2**



**Figure 3: Flowchart**

A diagram of a product

Description automatically generated

**Figure 4: Test Case**

A white rectangular box with black text

Description automatically generated

**Follow Up Question**

**What was challenging?**

The only trouble I ran into was updating all the list to the correct format.

**What did you learn?**

I learned how to convert all the arrays in my code to use generic.

**How would you improve on the project?**

There was nothing I would improve because it explained what and how I had to update the code.

**How can you use what you learned on the job?**

Use generics is going to be a common thing in the workplace so I am going to be able apply it to my code.