**Maiza Falcon Rojas**

**March 14, 2024**

**Assignment: Milestone 7**

**Class: CST-239**

Loom Link: https://www.loom.com/share/b4b22e2d04474be5837b9b44c734c705?sid=63bbd69f-61c8-4ab0-bad3-0e80d843d0ae

Github: https://github.com/maiza02/Milestone-7.git

**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**

**A screenshot of a computer screen

Description automatically generated**

**Figure 2: UML DIAGRAM**

A diagram of a function

Description automatically generated with medium confidence

**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?**

There were no challenges since I already knew how to set up the project up for Junit testing due to Activity 7. What was challenging was finding out if I needed to change my package name to Junit or not.

**What did you learn?**

I learned how to test classes that already exist and see if they run properly.

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

I would not change anything to the project because Activity 7 prepared me enough to add what was needed into this Milestone.

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

Knowing how to run junit for future projects is going to be very helpful in making sure things are running smoothly and properly.