CST 239 Milestone Report 2 Patrick Buhmbi 2/11/2024

My Plan for Managing Salable Products and Store Front Application:

In the first part of this task, we were tasked with understanding and designing UML class diagrams for Salable Product, Weapon, Armor, and Health classes. These classes must possess fundamental attributes such as Name, Description, Price, and Quantity. To accomplish this:

Identify Classes:

Salable Product: Represents a generic item available for sale.
Weapon: Refers to a product categorized as a weapon.
Armor: Represents a product classified as armor.
Health: Represents items related to health and healing.
Define Properties:

Salable Product: Mandatory attributes include Name, Description, Price, and Quantity.
Weapon, Armor, Health: These classes inherit properties from Salable Product and may incorporate additional attributes specific to their type.
Create UML Class Diagrams:

Illustrate each class along with its properties.
Use appropriate associations, like inheritance where applicable, to illustrate relationships between classes.
Include Minimum Properties:

Ensure that each class includes at least the essential attributes: Name, Description, Price, and Quantity.
Define data types for each attribute based on the requirements.
Consider Additional Properties:

Identify any extra properties unique to each subclass (Weapon, Armor, Health) and incorporate them into the class diagrams.
Establish Relationships:

Define connections between classes to indicate inheritance and any other relevant associations.
Review and Refine:

Carefully examine the UML diagrams to ensure they accurately represent the properties and relationships among the classes.
Make adjustments as needed to improve clarity and completeness.
In conclusion, by following these steps, we can effectively analyze and design the Salable Product, Weapon, Armor, and Health classes using UML class diagrams. This systematic approach aids in creating a clear and thorough structure for the system's data organization.

In the second part, we were tasked with analyzing and modeling a Store Front Application and Inventory Manager using UML class diagrams. Additionally, we needed to design a flowchart illustrating the logic of a Game User interacting with the Store Front. Here's how we can approach it:

Develop UML Class Diagrams:

Define the structure of the application using UML Class Diagrams, including classes like StoreFrontApp, InventoryManager, SalableProduct, Weapon, Armor, and Health.
Describe interactions such as StoreFrontApp utilizing InventoryManager, and InventoryManager overseeing SalableProducts.
Construct a Flowchart:

Create a visual representation detailing the steps a Game User follows while engaging with the Store Front Application, including presenting the Store Front's name, displaying a menu of available actions, executing selected actions based on user input, and presenting relevant feedback and error messages.
Implement Classes:

Design StoreFrontApp and InventoryManager classes to fulfill their designated roles, including rendering the welcome message and action menu, managing user input, overseeing the inventory, populating the initial inventory, and providing support for purchasing and canceling products.
Execute the Store Front Application:

Develop a Java application utilizing the established UML class designs, implementing necessary functionality to display the welcome message, action menu, and handle user input.
Utilize methods from InventoryManager to manage inventory and execute actions such as purchasing and canceling products.
Test and Address Issues:

Thoroughly test the Store Front Application to confirm its functionality aligns with expectations, debugging any identified issues, and refining the code as necessary.
Verify that feedback and error messages are appropriately displayed to the user throughout the interaction process.
By following these steps, we can effectively implement the Store Front Application and generate a flowchart illustrating the user interaction logic, ensuring clarity in design and execution while meeting the specified requirements.

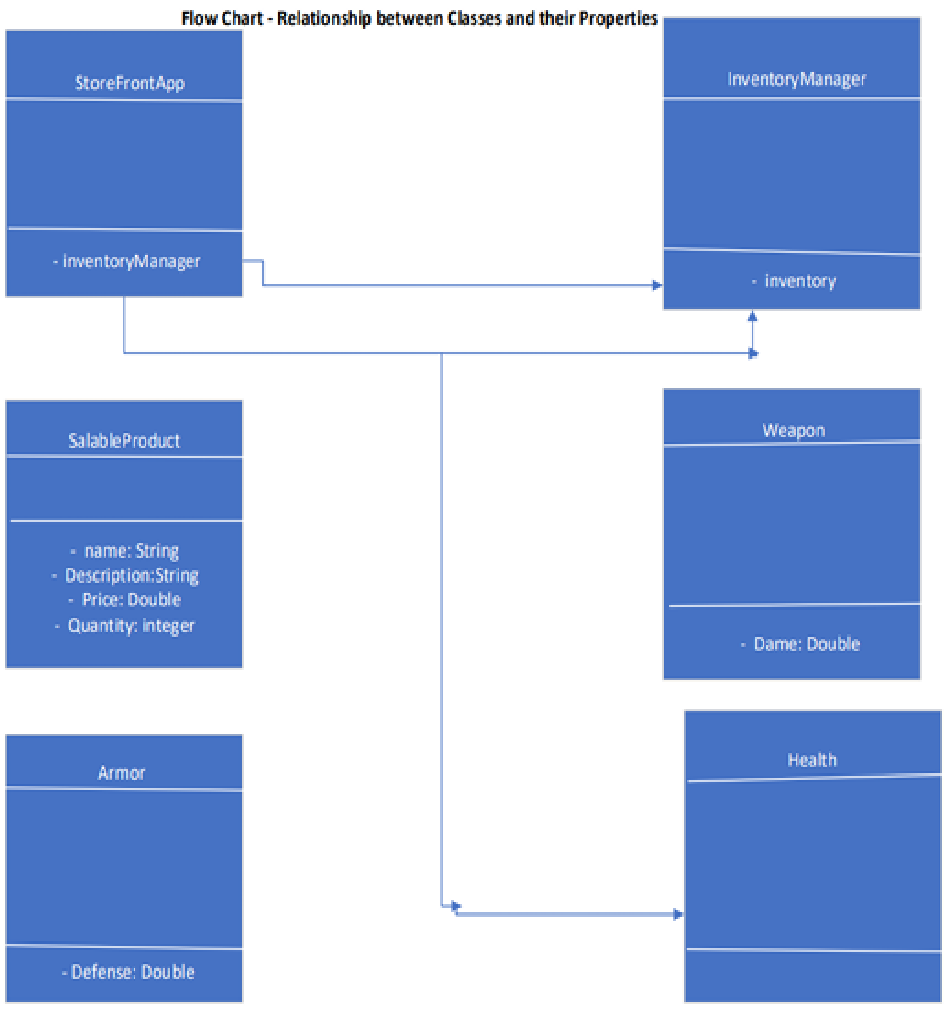

Figure 1 Show the flow chart of the store.

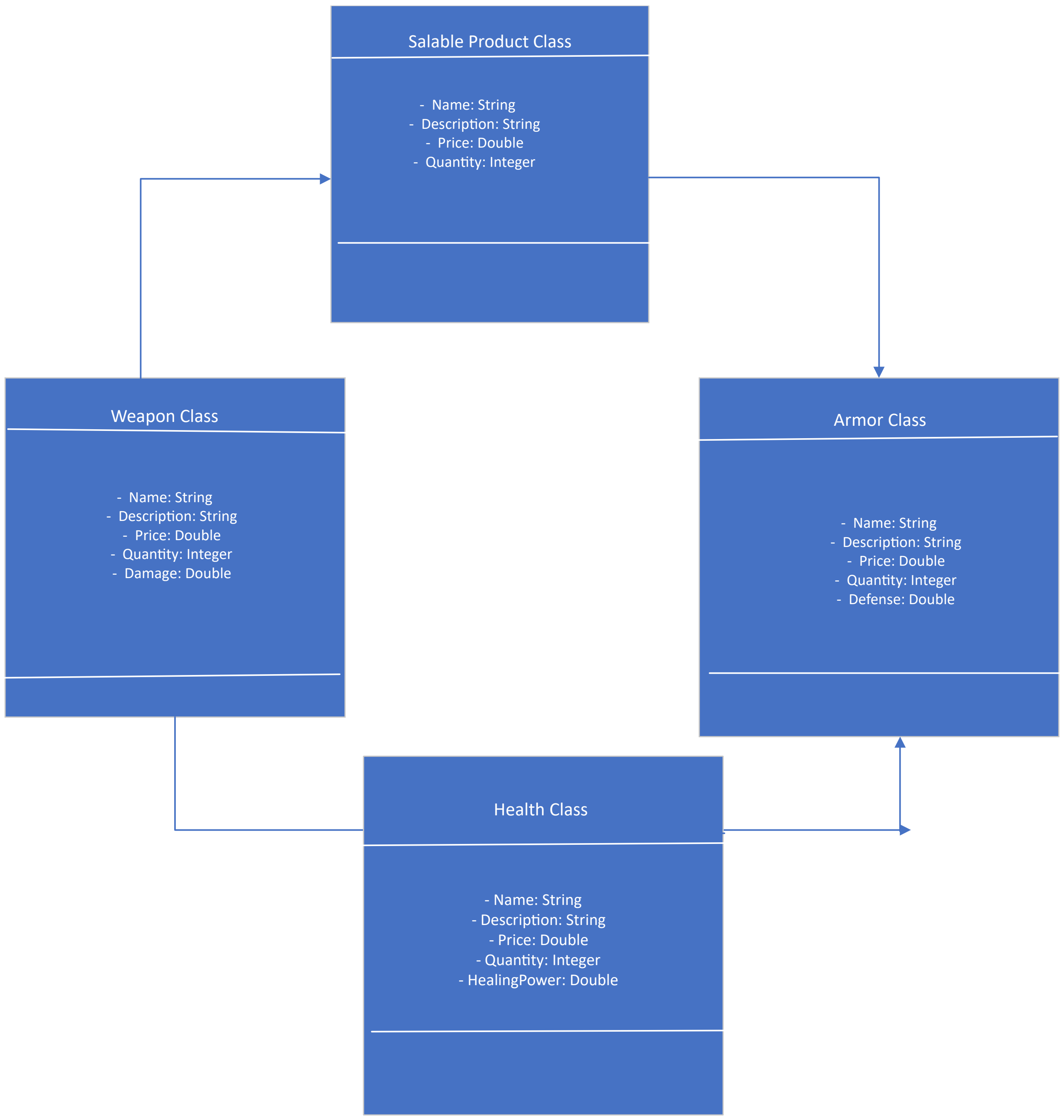

Figure 2. Shows the UML Class Diagram.
Let's create UML diagrams that show how different items like Salable
Products, Weapons, Armors, and Health-related items are organized in our system. We'll focus on the key details required for a Salable Product, which are its Name, Description, Price, and Quantity.
In these diagrams, each type of item has its own list of details. These include the fundamental details common to all items like Name,
Description, Price, and Quantity. The classes for Weapons, Armors, and

Health items are built on top of the Salable Product class. They inherit the basic properties like Name, Description, Price, and Quantity from the
Salable Product class but also have additional features specific to their type. This setup helps us organize our code in a way that makes it easier to reuse and maintain.

Flow Chart - Relationship between Classes and their Properties

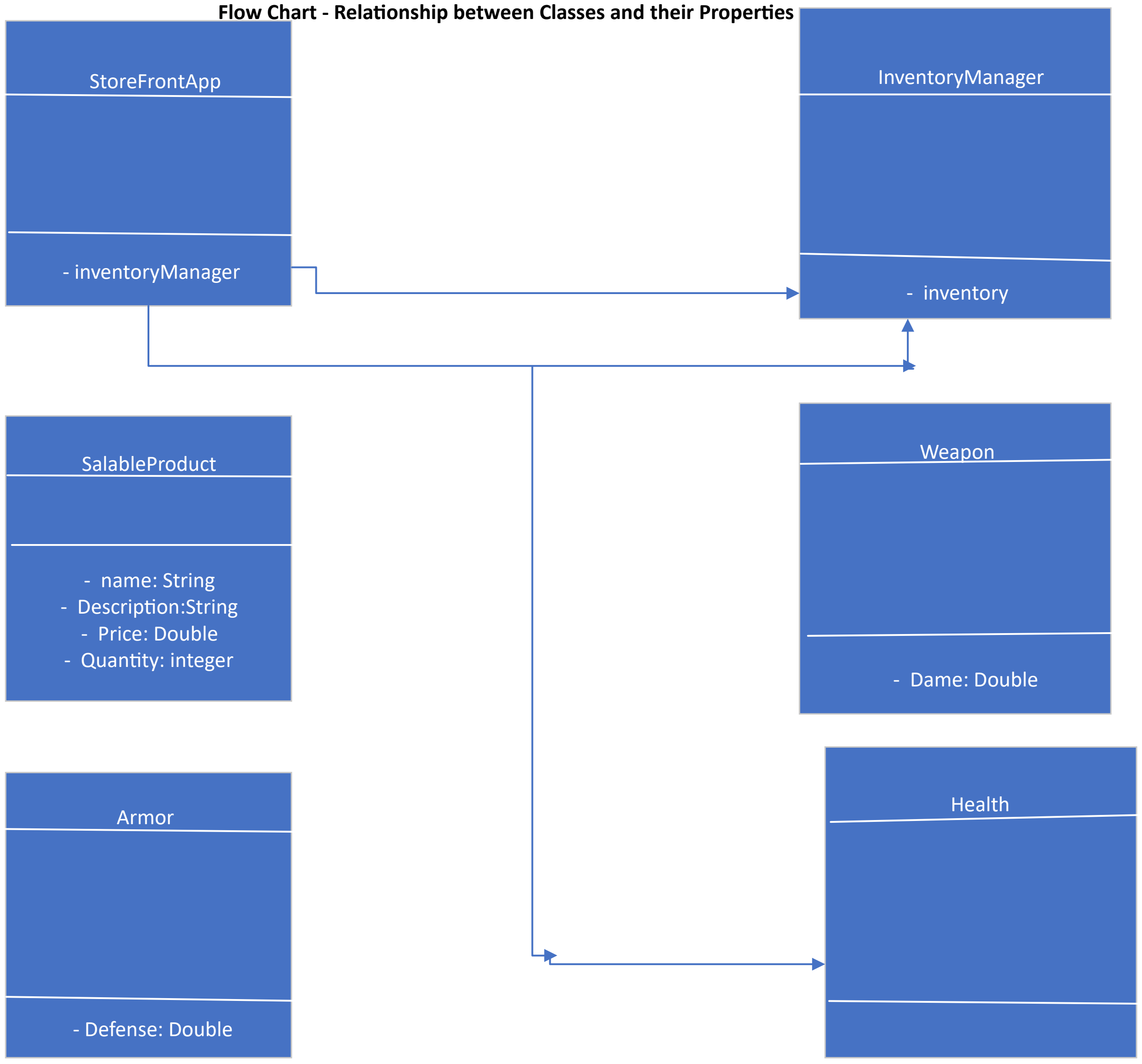

Figure 3 shows Here's the UML class diagram based on the provided classes and relationships
StoreFrontApp: Represents the application's user interface.
InventoryManager: Manages the inventory of salable products.
SalableProduct: Represents a product that can be sold.
Weapon, Armor, Health: Different types of salable products.
Relationships: StoreFrontApp uses InventoryManager for inventory management.
InventoryManager manages the inventory of SalableProducts.
Each SalableProduct type (Weapon, Armor, Health) is a subtype of SalableProduct, inheriting its properties and behavior.

This diagram illustrates how the classes are related to each other and their basic properties. Depending on the specific requirements and functionalities of your application, you may need to add more details, methods, or associations to the classes.

Below are links to the following.

1) Milestone 2 Javadoc
https://github.com/Buokwifoy/MY_GCU_COURSES/tree/main/Milestone2_JavaDoc.

2) Milestone 2 code zip
https://github.com/Buokwifoy/MY_GCU_COURSES/blob/main/Milestone%202_code_zip.zip

3) Milestone 2 Video Presentation
https://www.loom.com/share/a2084f6da4584a4ebf17dd24756c8ddc