We began by crafting UML diagrams for each class, which visually depicted how different parts of a store's system for managing inventory and shopping activities are set up and connected:

SalableProduct Class:

- Represents a product available for sale in the store.
- Contains attributes like productId, name, description, price, and quantity.
- Includes a constructor to initialize product details and methods to access and modify them.

InventoryManager Class**:**

- Oversees the store's inventory of products for sale.
- Maintains a list named "products" to store instances of SalableProduct.
- Provides functions to add, remove, and retrieve products from the inventory.

StoreFront Class:

- Acts as the interface for the store's front end.
- Contains an attribute called "inventoryManager" of InventoryManager type to manage stock operations.
- Implements methods to start the store, purchase products, and cancel orders.

ShoppingCart Class:

- Models a shopping cart for customers to manage their selections.
- Keeps track of SalableProduct instances in a list named "items".
- Provides functions to add or remove items, calculate the total cost, and simulate the checkout process.

The UML diagram illustrates the relationships among these classes, such as:

- StoreFront relies on InventoryManager for inventory tasks.
- ShoppingCart interacts with SalableProduct to adjust items and compute costs.
- StoreFront communicates with SalableProduct for purchasing and canceling orders, utilizing InventoryManager's features.

In essence, the UML diagram acts as a guide for understanding the structure and interactions within the store's management system, aiding developers in effectively building and managing the application.

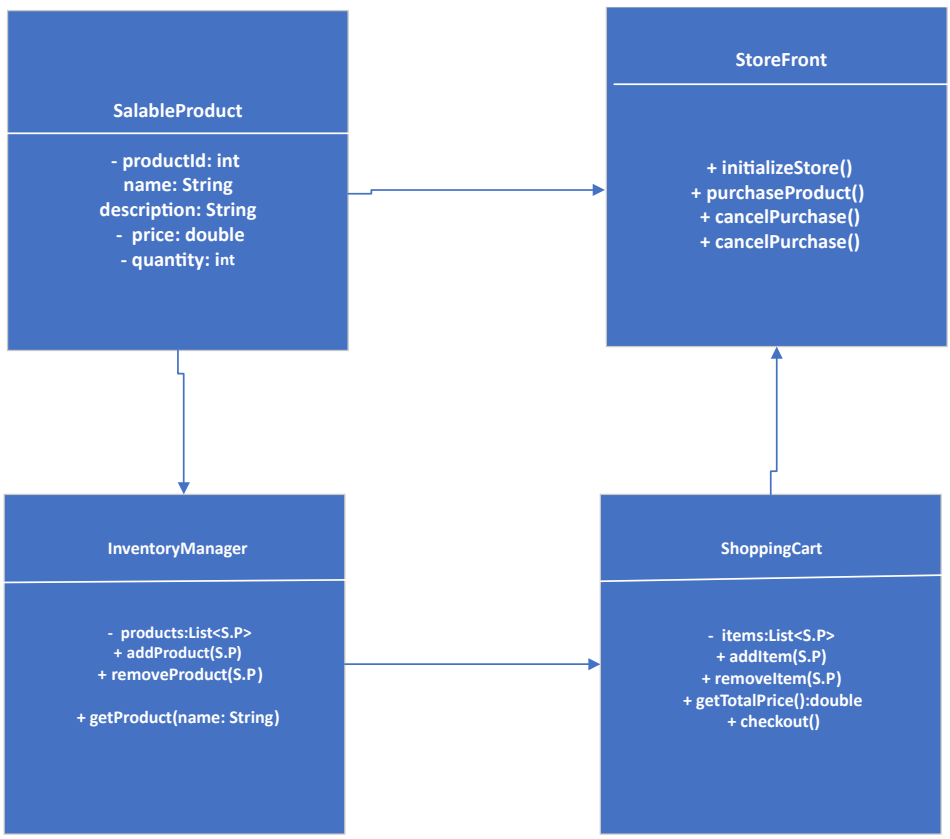
As for the main class:

- It simulates various store actions by creating instances of SalableProduct, InventoryManager, and ShoppingCart.
- Products are added to the inventory, and its contents are displayed.

- Products are added to the shopping cart, and its contents are displayed before removing a product.
- The updated contents of the shopping cart are displayed, and it is confirmed to be empty by emptying it.

In the code:

- The SalableProduct Class represents products available for sale, encapsulating attributes like productId and name.
- InventoryManager Class manages the store's inventory, including adding and removing products.
- StoreFront Class integrates InventoryManager and ShoppingCart components, but its method bodies are placeholders.
- ShoppingCart Class manages the shopping cart's contents yet lacks implementation details for adding and removing products.
- Weapon Class defines attributes and methods for a type of weapon product but lacks the implementation of the compareTo method required by the Comparable interface for proper functionality. Figure 1 below shows the UML Class diagram for the Store Front



Flowchart of User Interaction with the Store Front

This flowchart outlines the steps a user follows when they engage with the Store Front, including any interactions with the Inventory Manager and the Shopping Cart, if needed. It shows how the user moves through the store's interface and performs actions like buying and canceling products. To start, the user initializes the store and receives a friendly welcome message. Then, the Store Front presents a list of actions for the user to choose from. If the user decides to purchase a product, the Store Front displays the available inventory, lets the user pick a product, confirms the purchase, and gives feedback. In case the user wants to cancel a purchase, the Store Front shows the inventory again, allows the user to select the product they wish to cancel, confirms the cancellation, and provides feedback accordingly. Lastly, if the user decides to exit, the program closes.

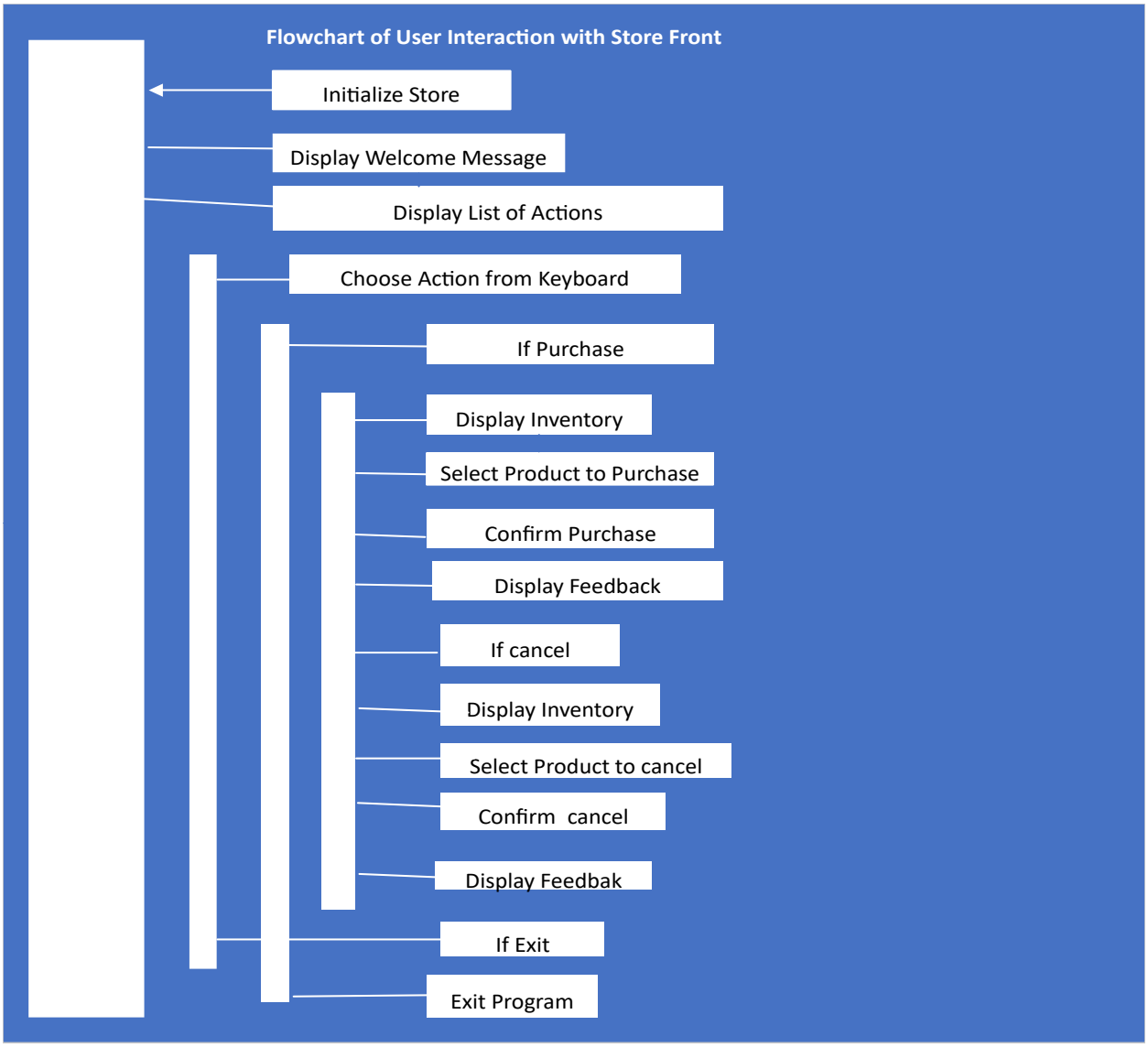


Figure 2 Shows the Flowchart of the User Interaction with a Store Front

Next, we'll proceed to put into action the provided UML class designs for SalableProduct, InventoryManager, ShoppingCart, and StoreFront. This code is equipped with features to set up the store, buy products, and cancel purchases. The Main class is responsible for trying out these functions and showing what's inside the inventory and shopping cart.