

Group 3 Nozama Milestone 2: Application Requirements

Eyan Eubanks, Jamar Andrade, Jordan Diaz

Project Title: Nozama Shopping App

Java Swing Application

CRC Diagram

Class Name	Responsibility	Collaborators
Item	Converts to String Retrieves ID Retrieves Name Converts to JSON Retrieves Sell Price Answers "is Bundle?" Retrieves Quantity Retrieves Vendor Retrieves invoice price Retrieves Description Compares with objects Retrieves hashCode Changes ID Changes Name Change Sell Price Changes Description Changes Quantity Changes Vendor	Cart System Bundle
CustomerAccount	Perform Action on Login	User
SellerAccount	Converts to JSON Retrieves username Converts to String Retrieves Profit Retrieves Revenues Retrieves Costs Compares with objects Retrieves hashCode Adds to Revenues Adds to Costs Calculates profit	System User
User	Converts to JSON Converts to String Retrieves username Retrieves password Retrieves account	System

System	Logs user in Retrieves Inventory Retrieves Vendors Finds seller given a username Retrieves Cart Retrieves the last Items ID Checkout the cart Set current user Retrieve current user Add observers Notify observers	User SellerAccount CustomerAccount View Cart CouponDecorator Bundle Item JsonHandler
View	Gets updated by System On update, show screen	System
Cart	Retrieves Cart Retrieves the quantity for each Item in the cart Removes Item from the cart Print the Items in the cart Calculates the total and retrieves it Calculates the total with coupons Changes the total with coupons Retrieves quantity of a specific item in the cart	Item Bundle System CouponDecorator
JsonHandler	Gets JSON Array from JSON file Writes data to JSON	System
Bundle	Convert to String Retrieve sell price Answers "is Bundle?" Retrieves quantity Retrieves instance of vendor Retrieves name Converts to JSON Adds item to bundle Retrieve the list of Items Compares with other objects Retrieves hashCode Retrieves ID Changes quantity	Item System Cart

Class Diagram

Design Patterns

Singleton Pattern

Requirements:

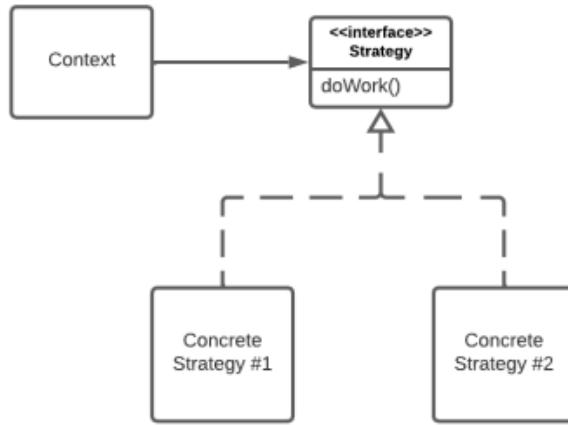
1. Define a class with a private constructor.
2. The class constructs a single instance of itself
3. Supply a static method that returns a reference to the single instance

NozamaSystem.java

```
private NozamaSystem(){}  
  
public static NozamaSystem getInstance()  
{  
    if (instance == null)  
        instance = new NozamaSystem();  
  
    return instance;  
}
```

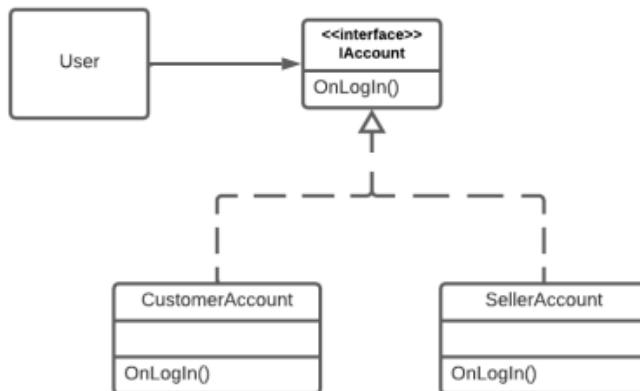
Strategy Pattern

Requirements:



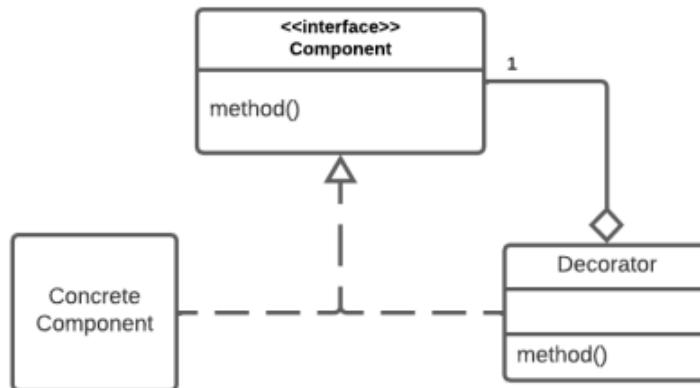
Implementation:

Name in Design Pattern	Actual Name (layout management)
Context	User
Strategy	IAccount
Concrete Strategy #1	CustomerAccount
Concrete Strategy #2	SellerAccount
doWork()	OnLogIn()



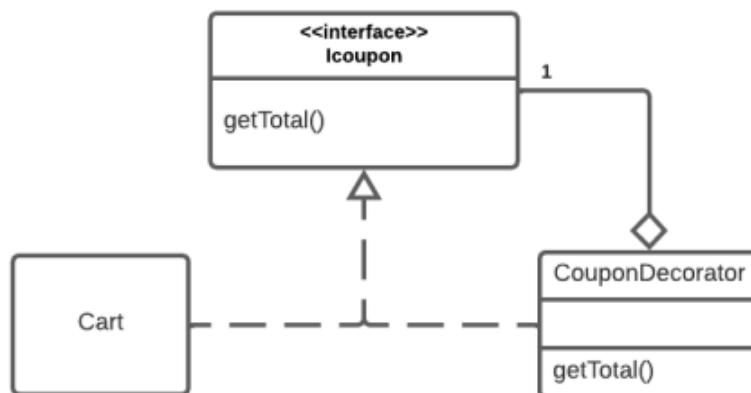
Decorator Pattern

Requirements:



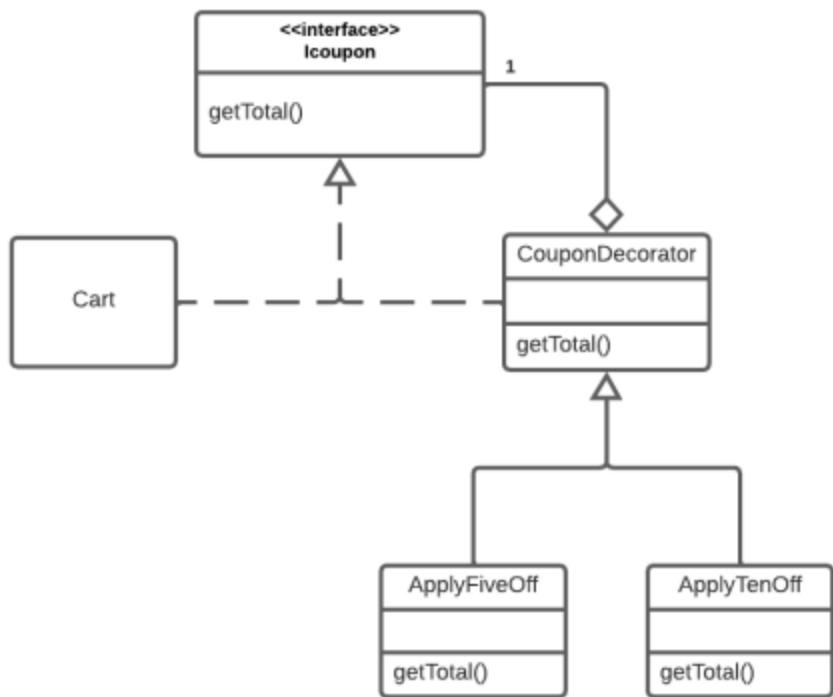
Implementation:

Name in Design Pattern	Actual Name (layout management)
Component	ICoupon
Concrete Component	Cart
Decorator	CouponDecorator
method()	getTotal()



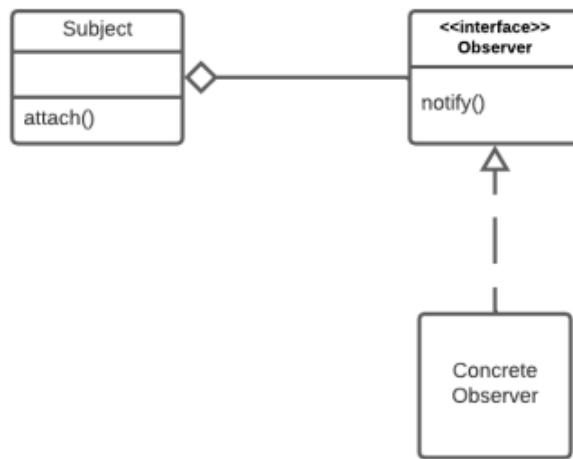
Sequence Diagrams

Additional:



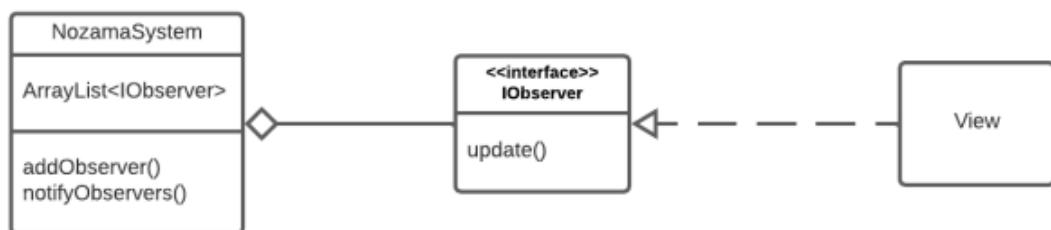
Observer Pattern

Requirements:



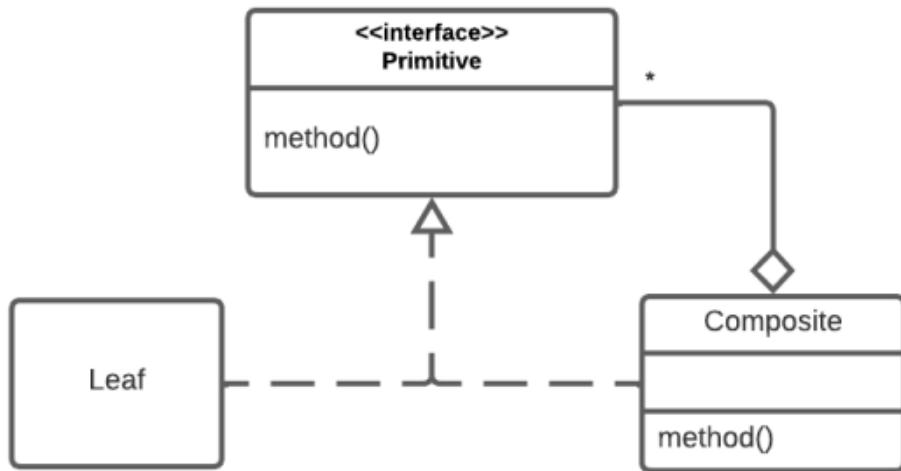
Implementation:

Name in Design Pattern	Actual Name (layout management)
Subject	NozamaSystem
Observer	IObserver
Concrete Observer	View
Attach()	addObserver()
Notify()	notifyObservers()



Composite Pattern

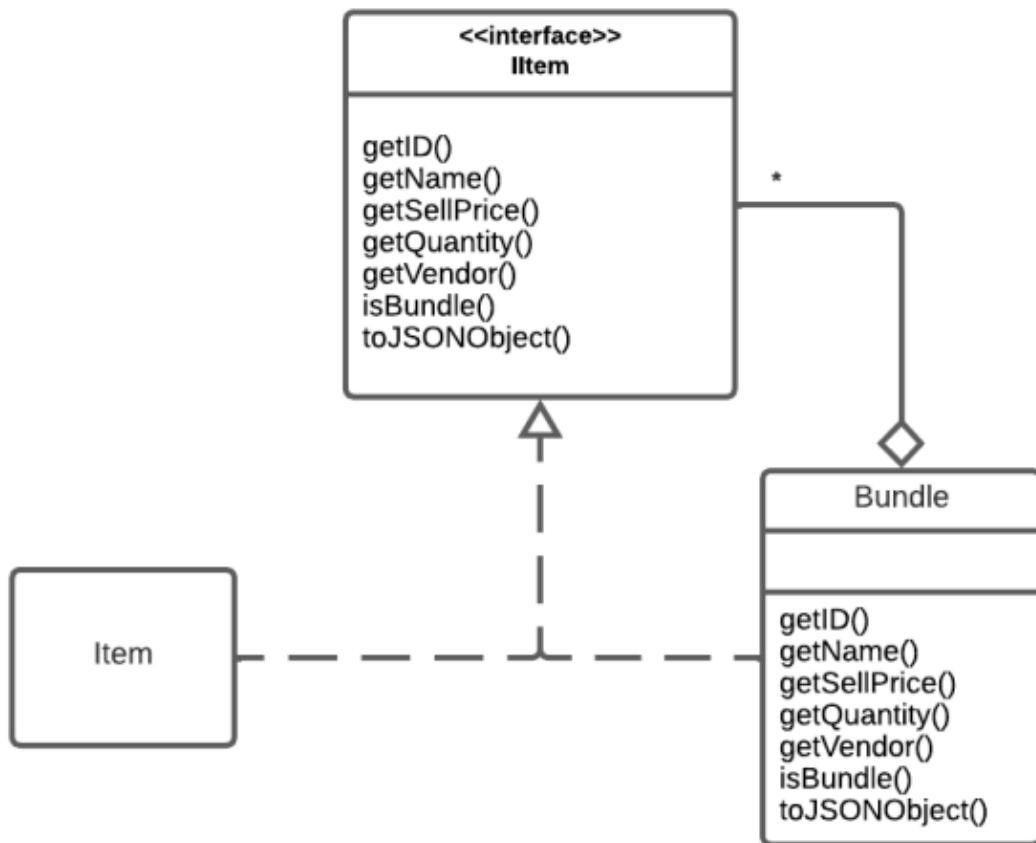
Requirements:



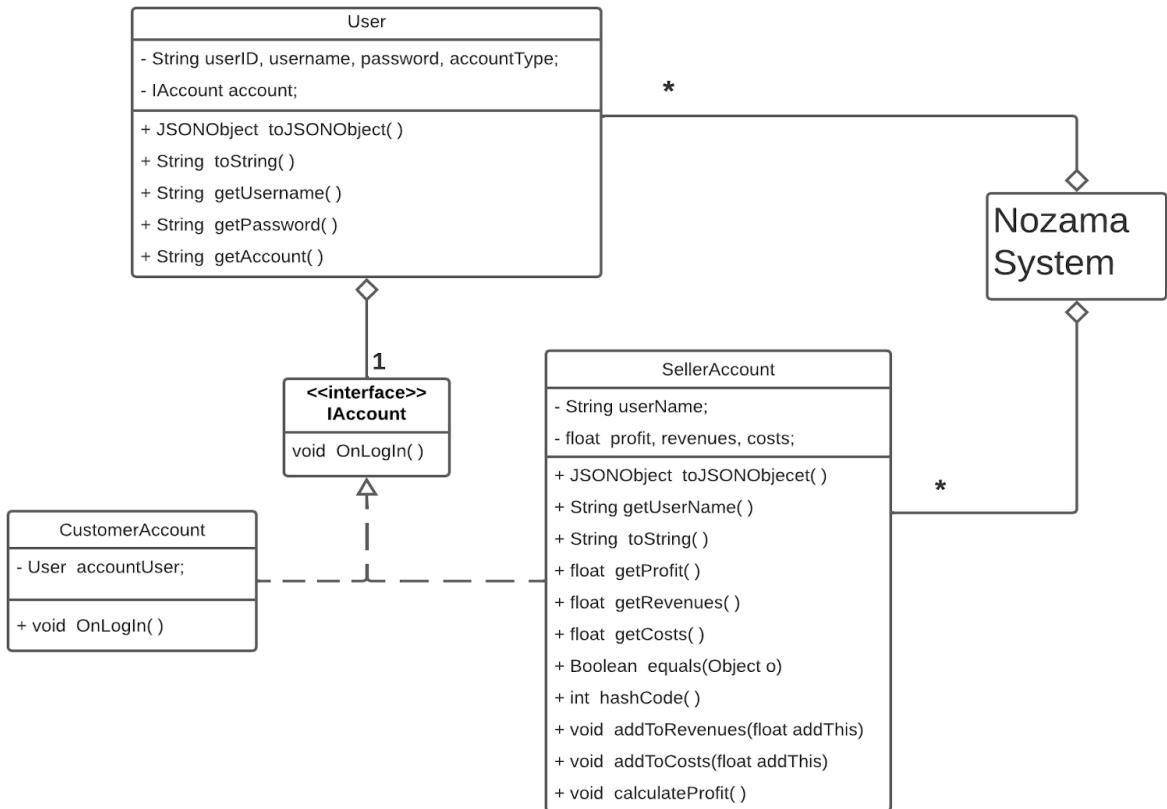
Implementation:

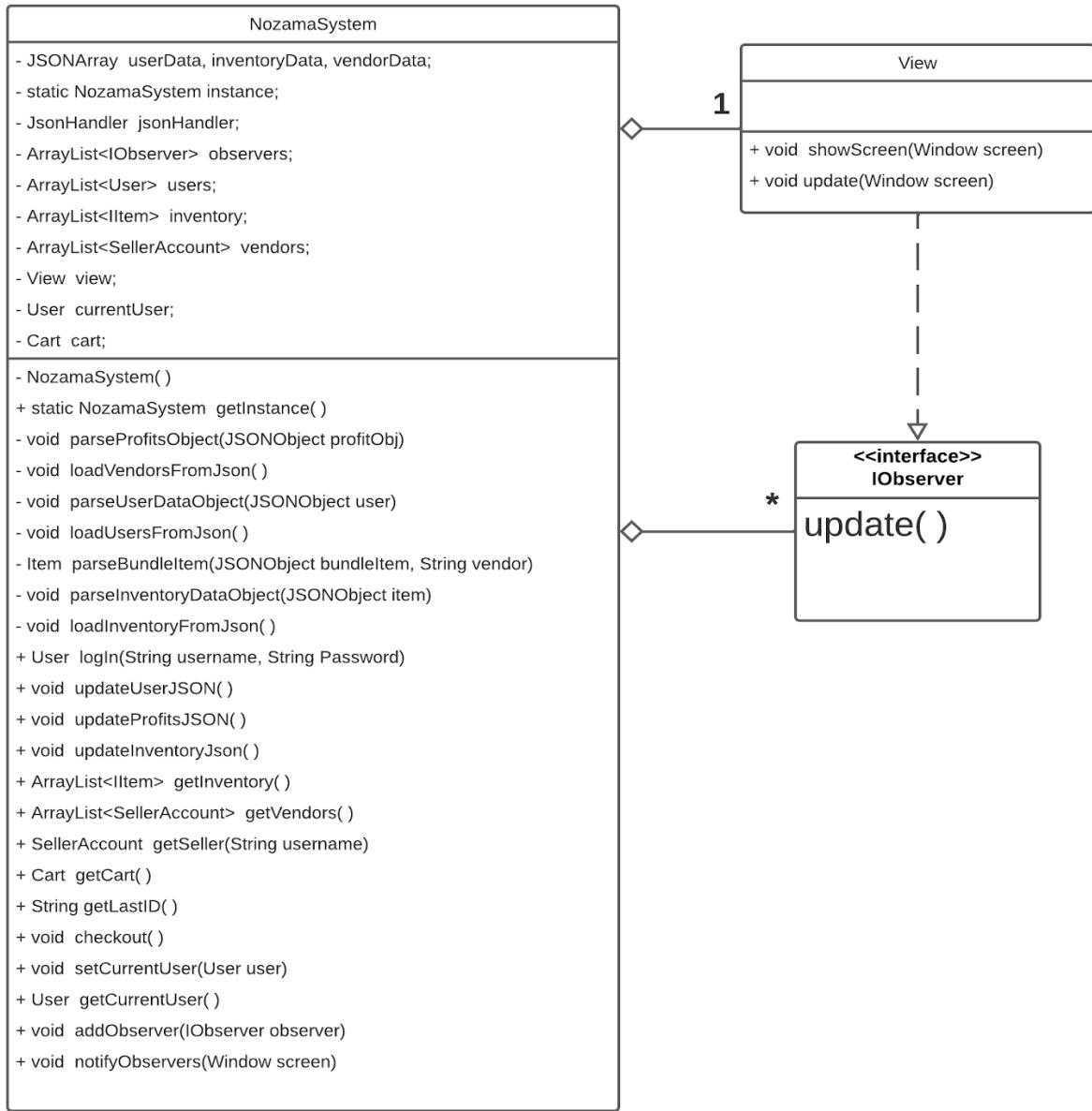
Name in Design Pattern	Actual Name (layout management)
Primitive	Item
Composite	Bundle
Leaf	Item
method()	<code>getID()</code> , <code>getName()</code> , <code>getSellPrice()</code> , <code>getQuantity()</code> , <code>getVendor()</code> , <code>isBundle()</code> , <code>toJSONObject()</code>

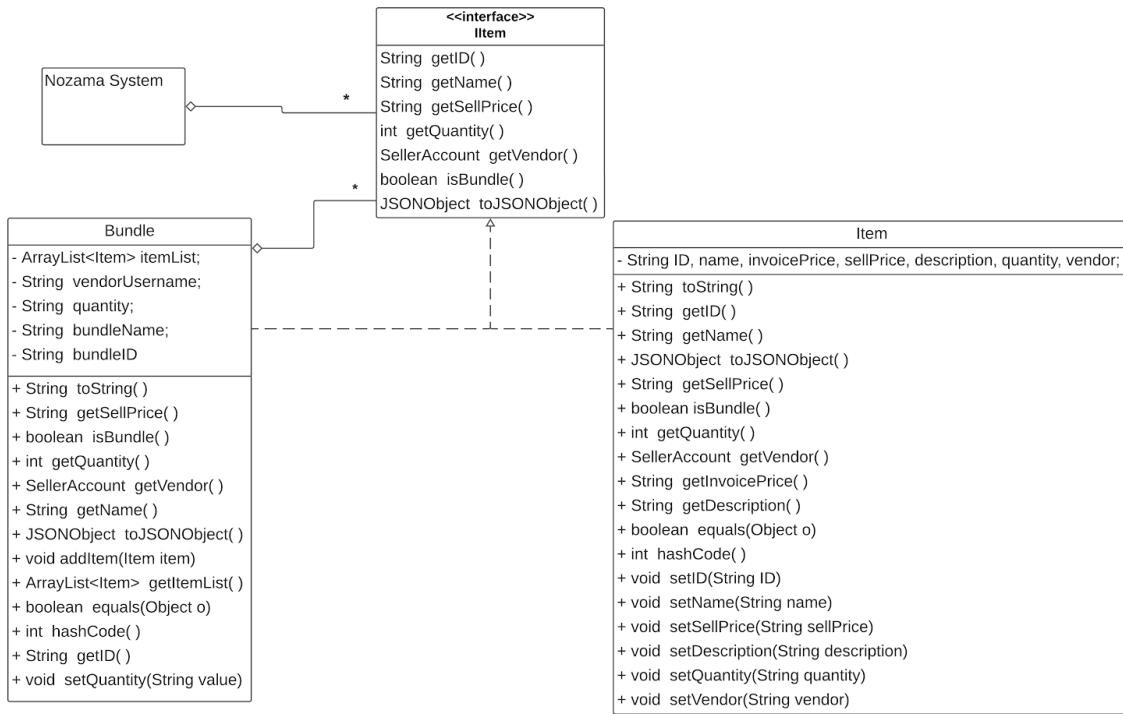
Note: Interfaces have an I before their name. so, the Interface IItem is not the class Item

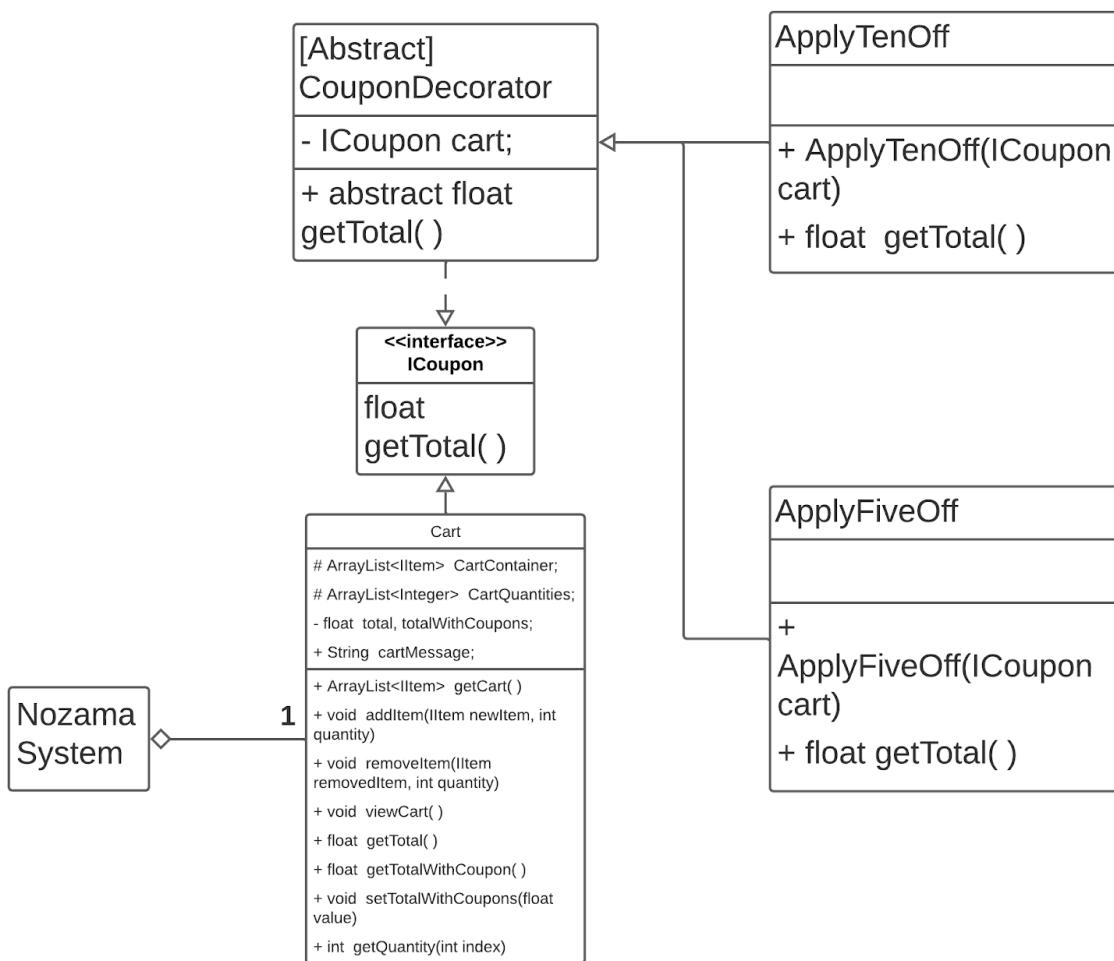


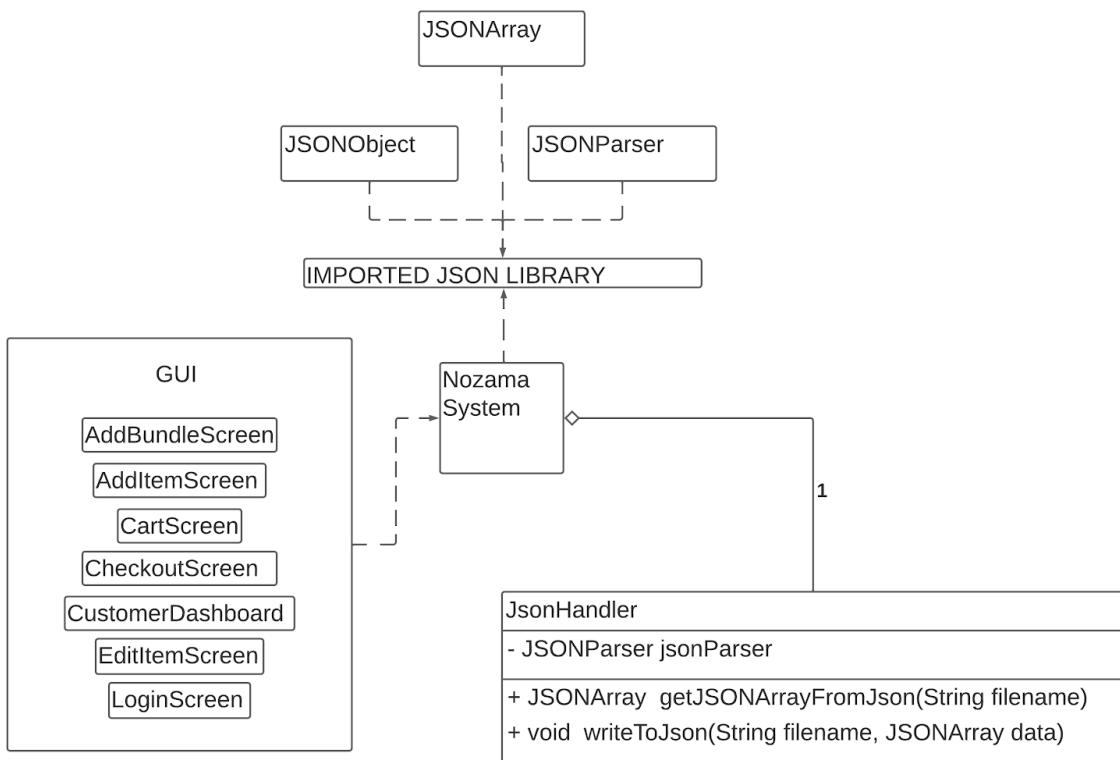
Class Diagram

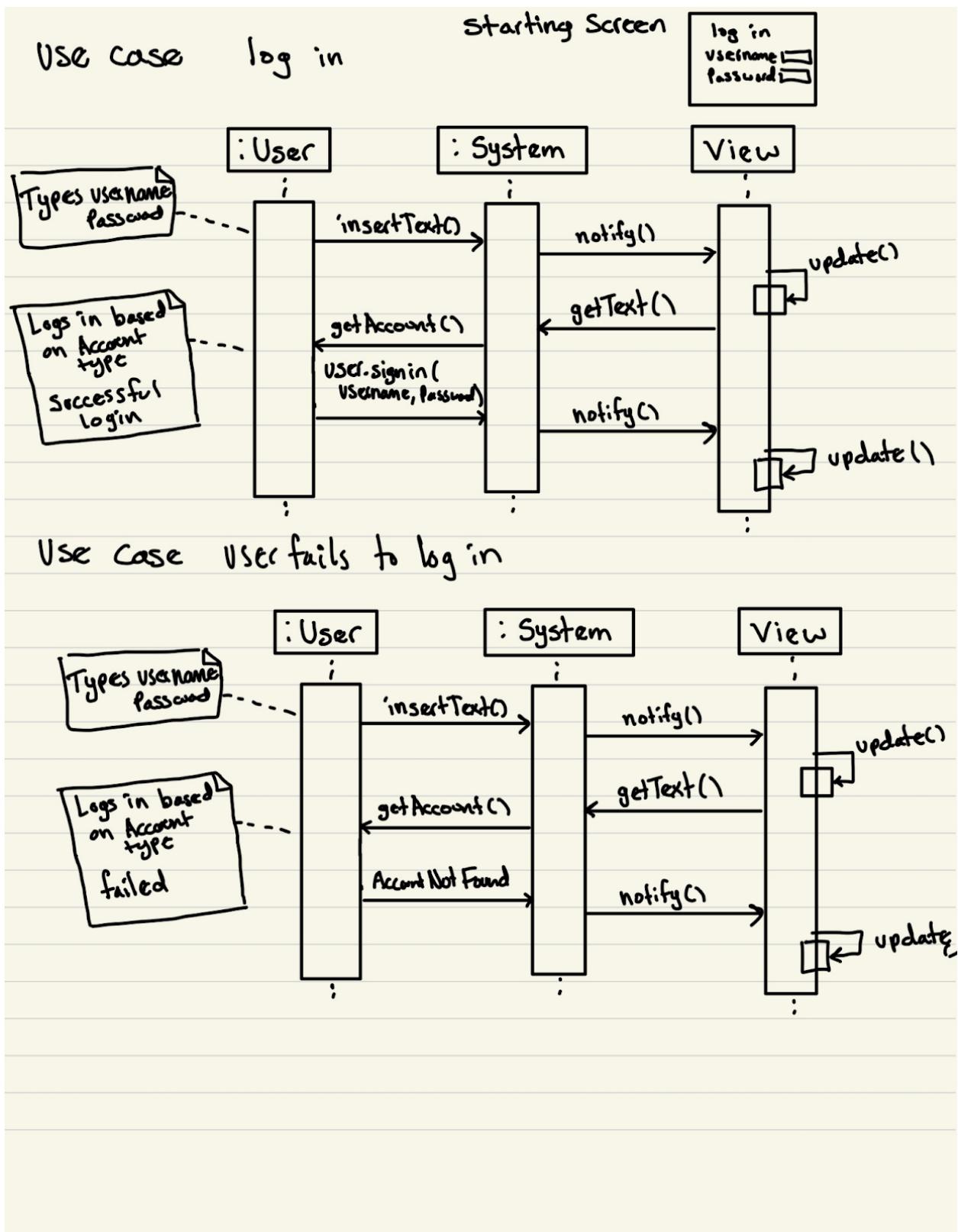




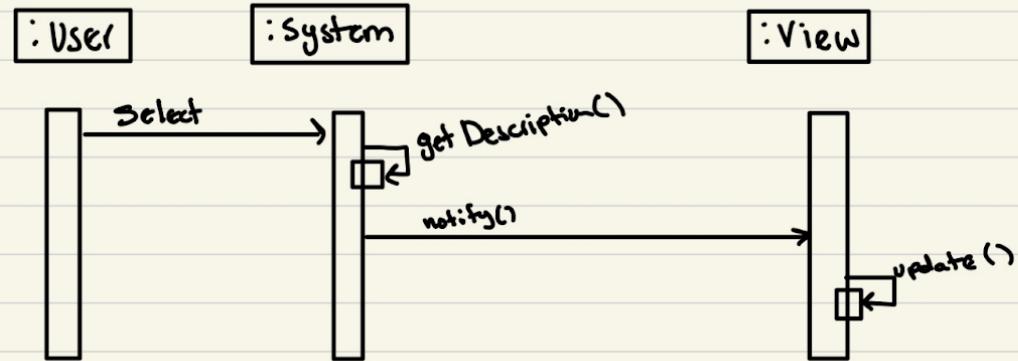




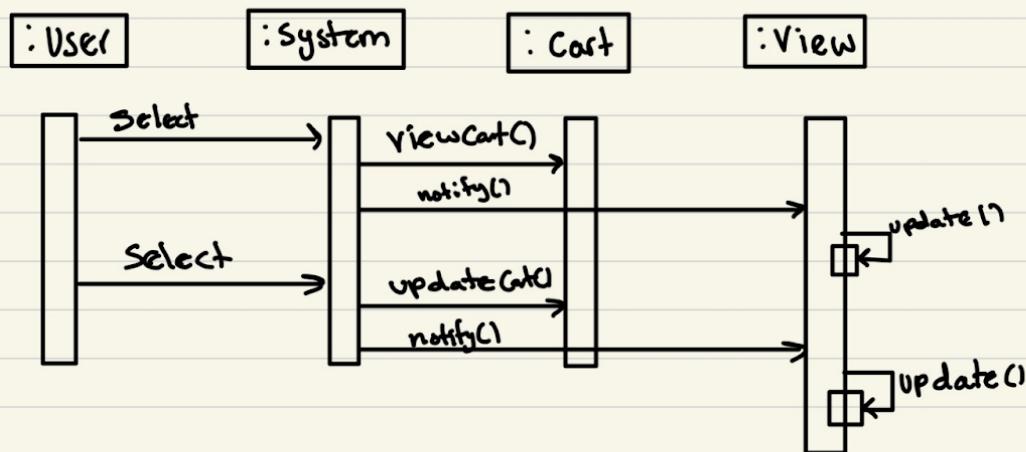




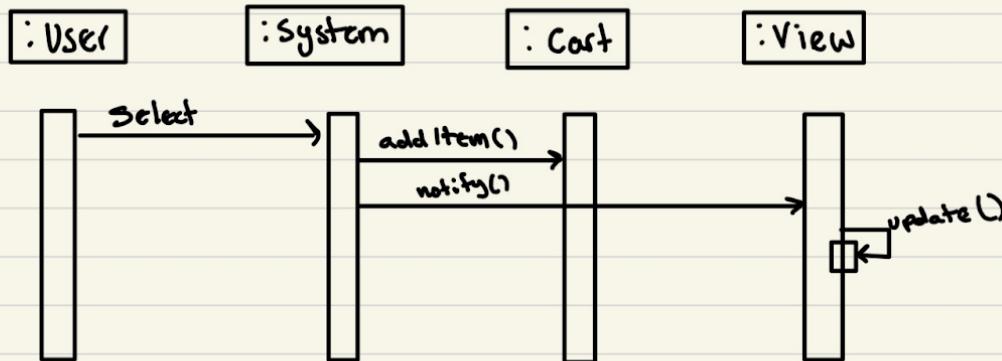
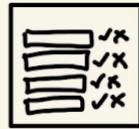
Use Case Customer reviews Product Details



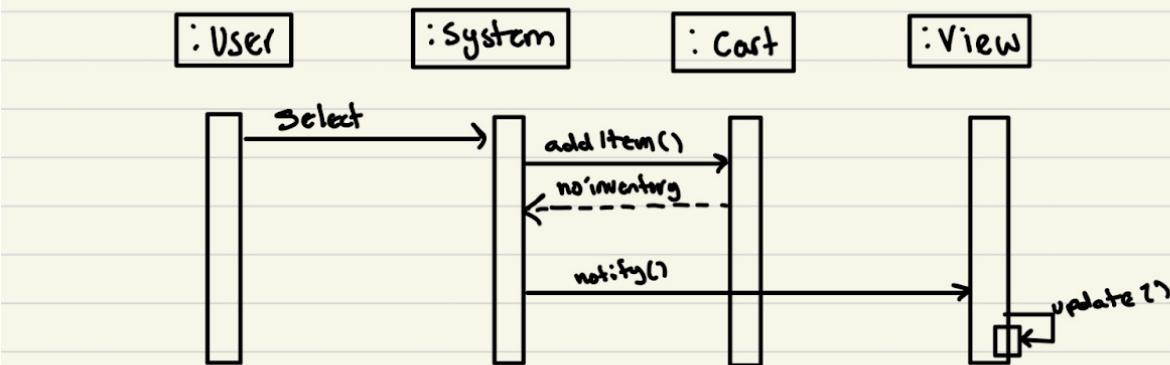
Use Case Customer reviews / updates shopping Cart



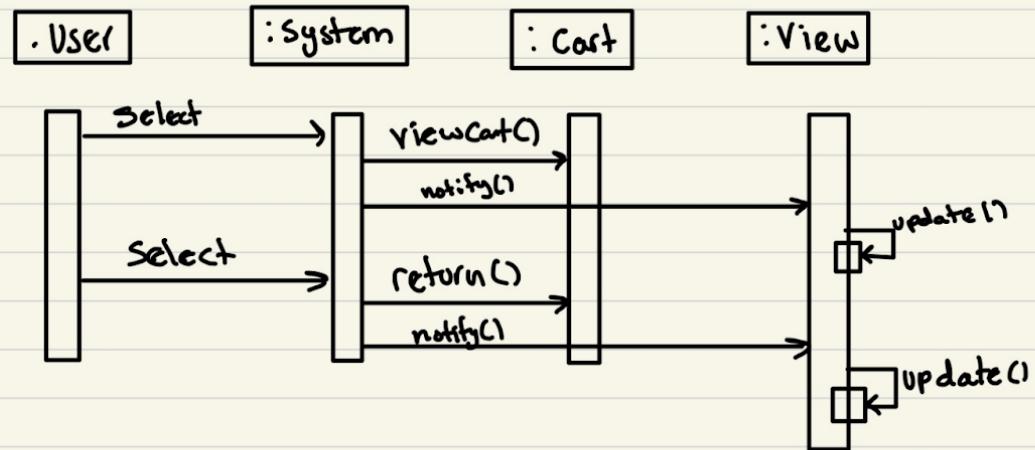
Use Case Customer Adds items to shopping Cart



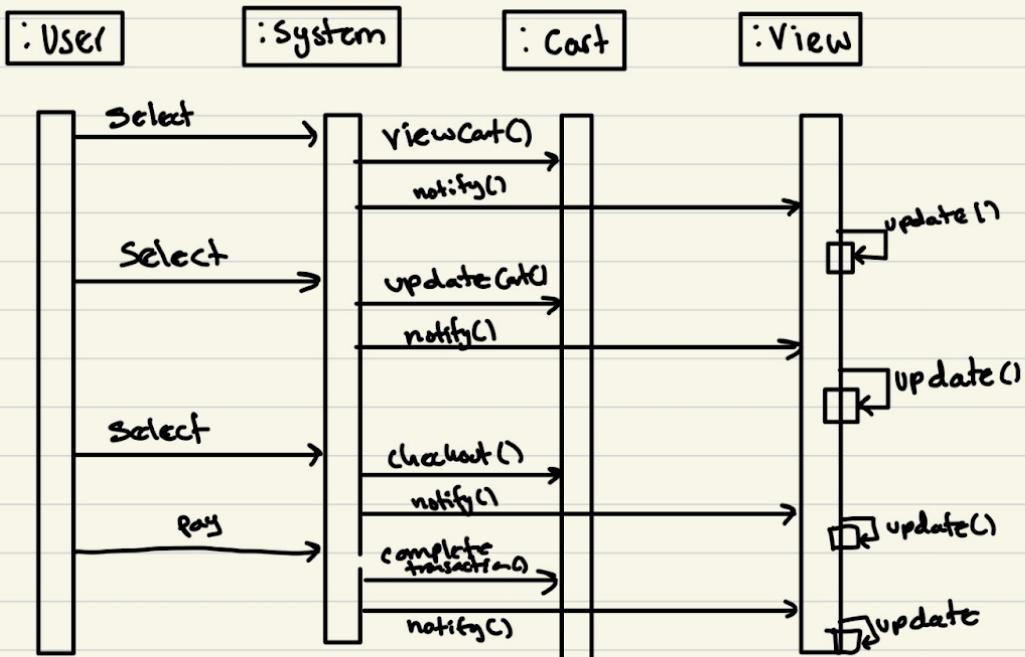
Use Case Customer Adds item but no inventory



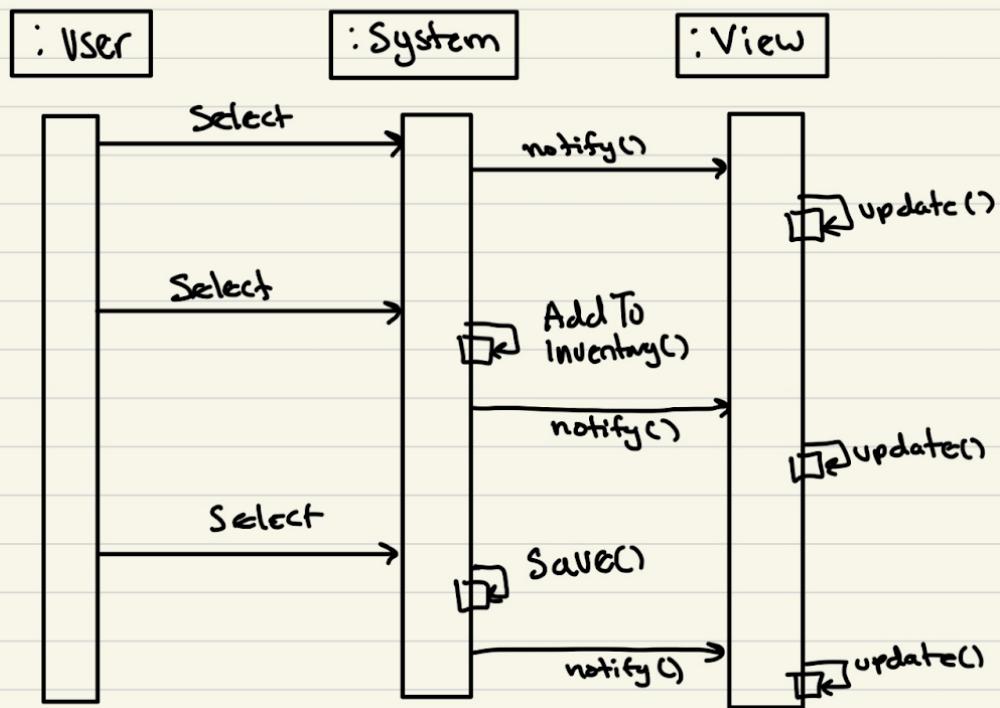
Use Case Customer exits review shopping Cart



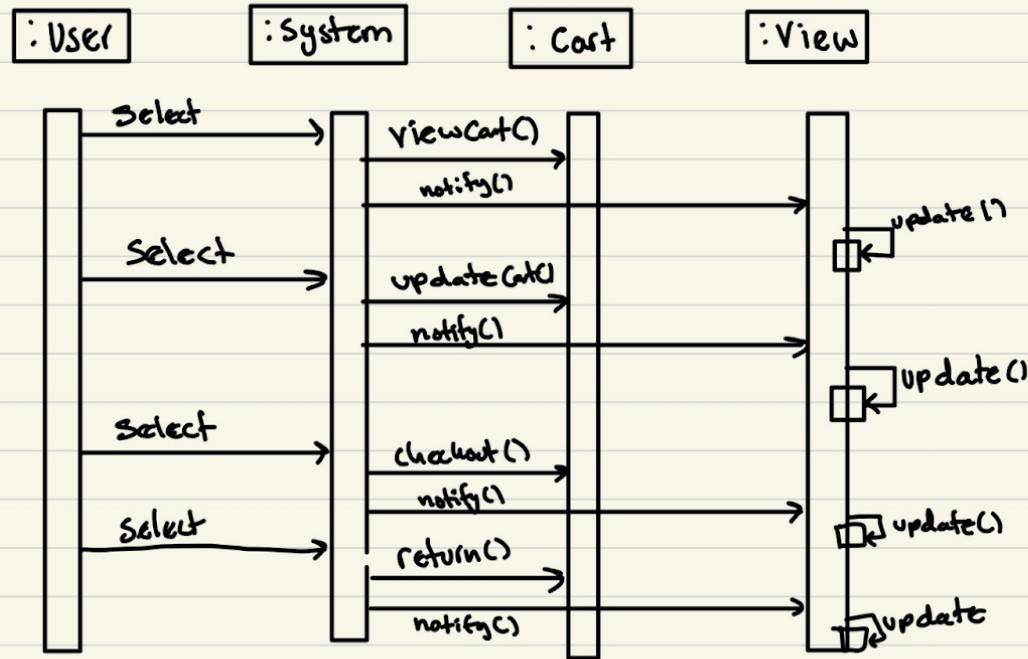
Use Case Customer Checks out



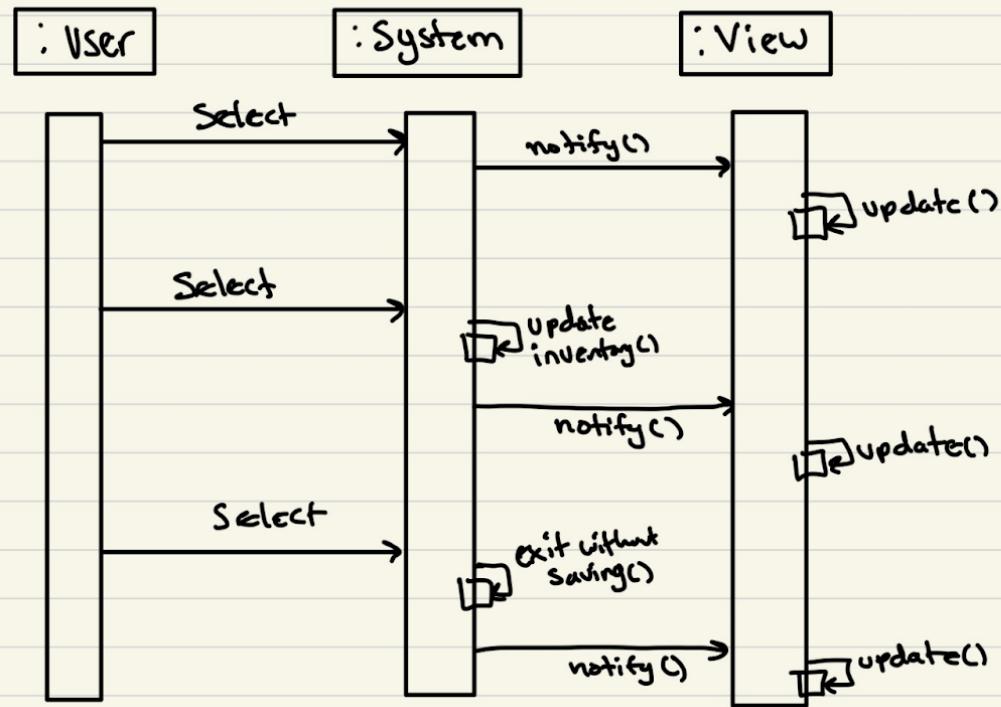
Use Case Seller Adds New Product



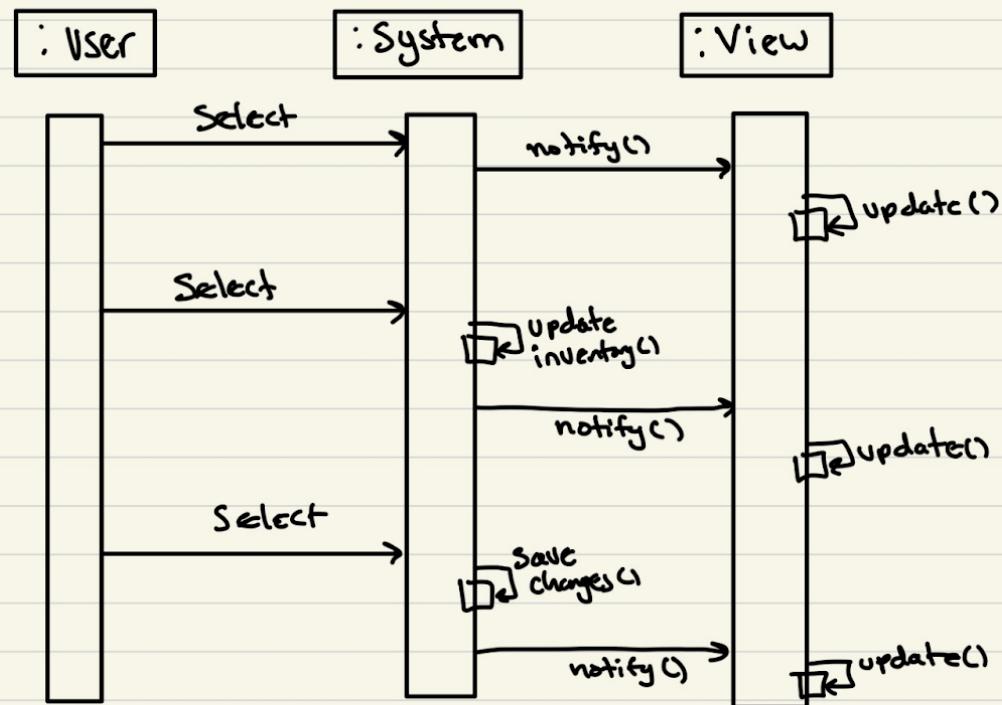
Use Case Customer exits checkout before payment



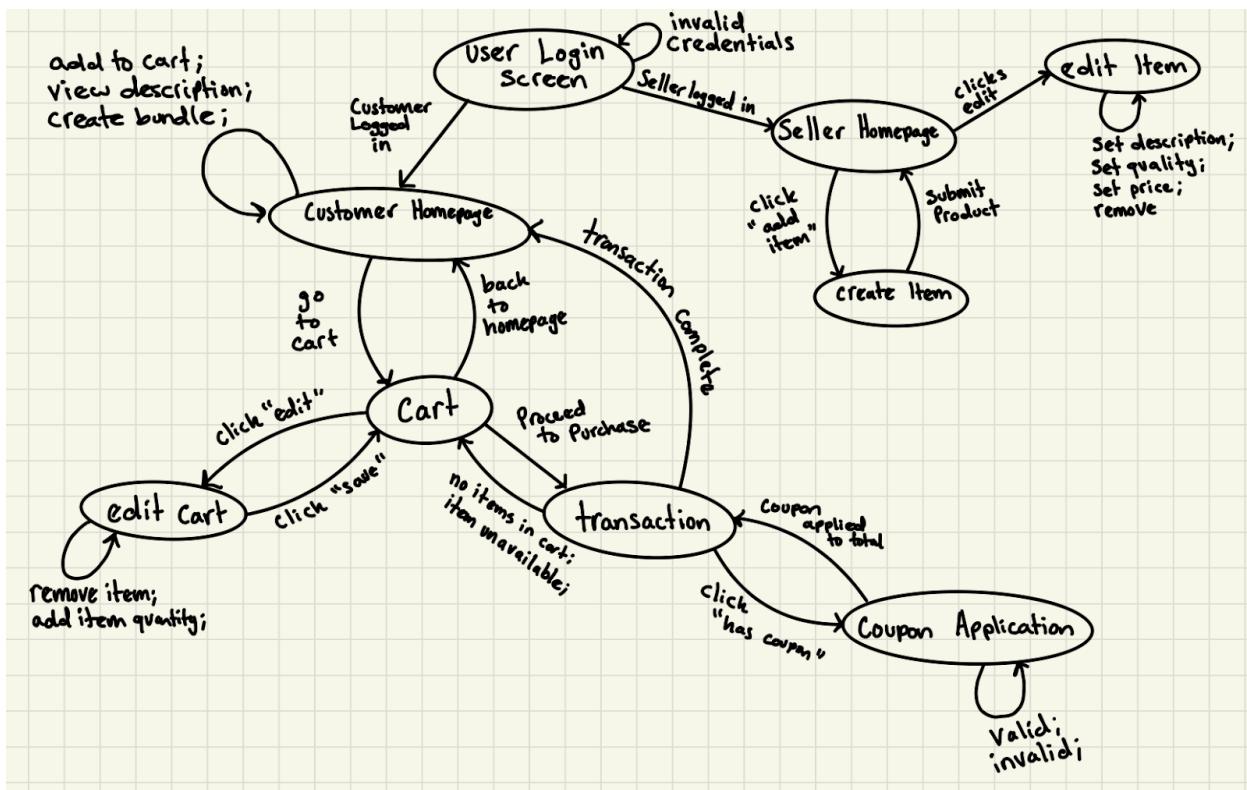
Use Case Seller exits before confirming update



Use Case Seller Review / updates inventory



State Diagram



-seller homepage is where the seller can see revenue, sales, and profit