CS 3354

Project: Inventory Management System

Authors: Ashley Cook, Sarah Gonzalez
Instructor: **Tarek Salah Uddin Mahmud**
December 10, 2023

1 Git Link

```
https://github.com/ash-cook99/Java.git
```

2 Introduction

The purpose of this project is to develop an Inventory Management System for a business franchise, which holds summary detail of all the business' stores, as well as the product inventory stock. This program overall is a management system that provides users with the ability to carry out transactions and generate summary reports. The students implemented this system using Java coding language.

3 Main.java

This class is the entry point of the program and contains the main method. It provides a command-line interface for users to interact with the Inventory Management System. Users can add products, stores, perform incoming and outgoing transactions, and generate reports using the prompts and input provided through the console. The loop allows users to continuously interact with the system until they choose to exit.

4 Product.java

Product class is needed to carry relevant attributes for the product placed in the system. Methods are needed to access the product data and amend the product detail such as the ID, Name, and count.

5 Store.java

The class signifies the stores within the business franchise. The constructor initializes the store with the data/attibutes "name" and "address". Methods with getters and setters provide access and ability to edit the store's attributes.

6 StockManager.java

Product inventory and relevant stores are managed by the stockmanager class. The constructor is used to manage the products and store data. We then use many methods to provide functionality. To complete our goal the class needs methofs to add data to the list, by taking the corresponding objects. Also, return the lists of all the products in stock and stores.

7 Transactions.java

There are two types of transactions, but a Transaction class was needed for general processing of all the Transactions and their details. The HashMap is helpful in storing data into the product list. This class uses methods to return the list of products and number of products involved in each transaction. The addProduct method allows the addition of a product and it's quanity to the product list.

8 IncomingTransactions.java

This class has a constructor that inherent data from the Transaction class and all of its functionality. Store, Product, and numberOfItems is used as input to initialize the transaction. No methods are needed. The items are added to the inventory

9 OutgoingTransactions.java

This class has a constructor that inherent data from the Transaction class and all of it's functinoality. Store, Product, and numberOfItems is used a s input to initialize the transaction. No methods are needed. The items from the inventory are sent to a store.

10 TransactionsManager.java

TransactionsManager is an important class that manages incoming and outgoing transactions. The constructor is used to initialize our transaction list. Methods are used to help manage transaction actions and reporting purposes. The class can add new incoming/outgoing transactions to our system

11 Demo/Output

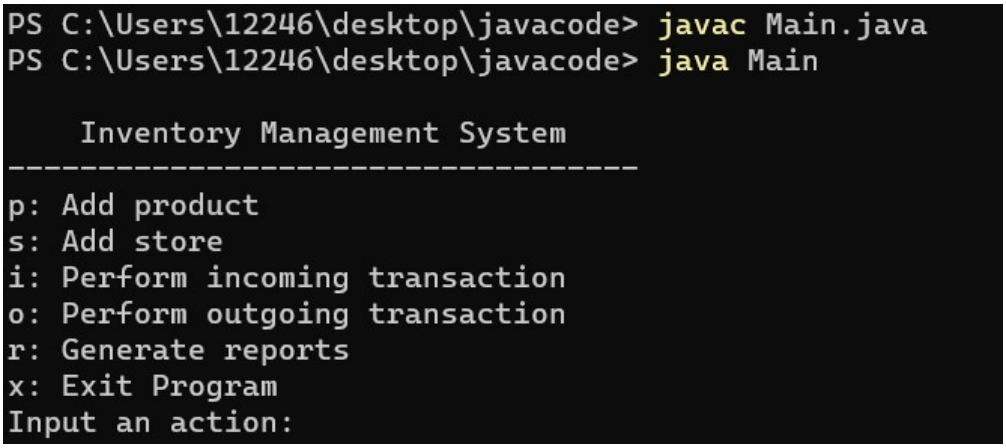


Fig. 1: fig: Screenshot of menu from console

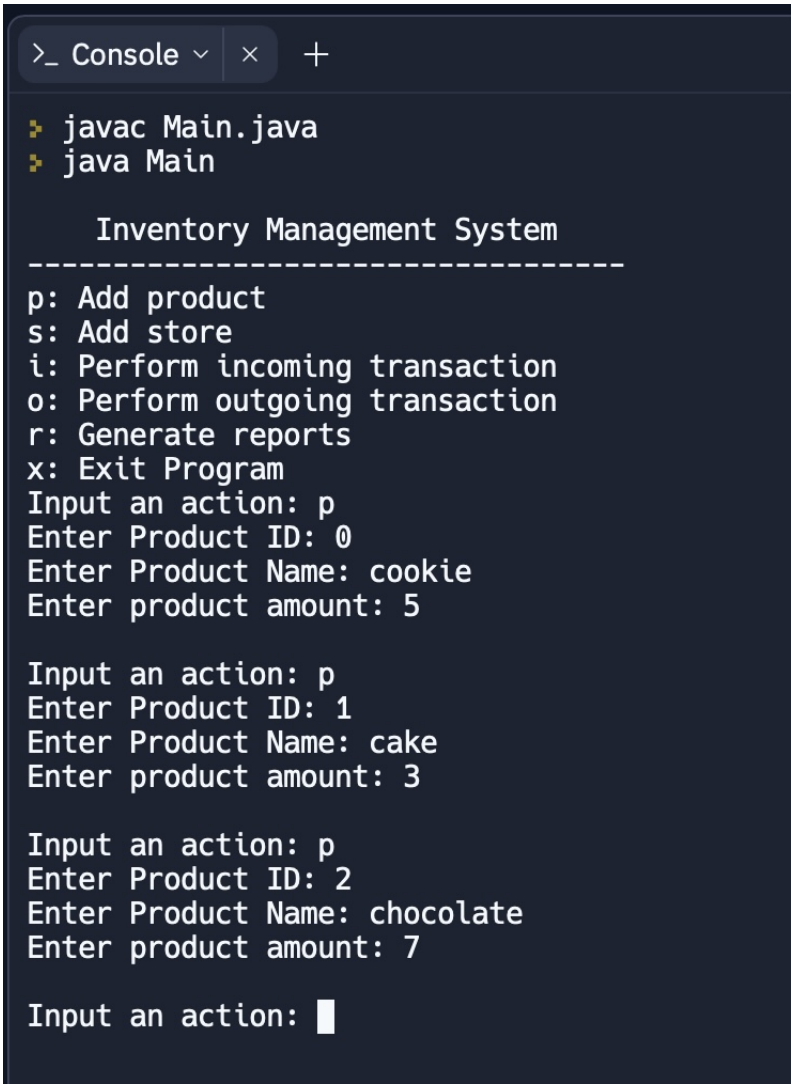


Fig. 2: fig: Screenshot of menu from console
The user can then enter 'p' to Add product, and is then prompted to enter a numerical product ID, product name, and quantity of the product.

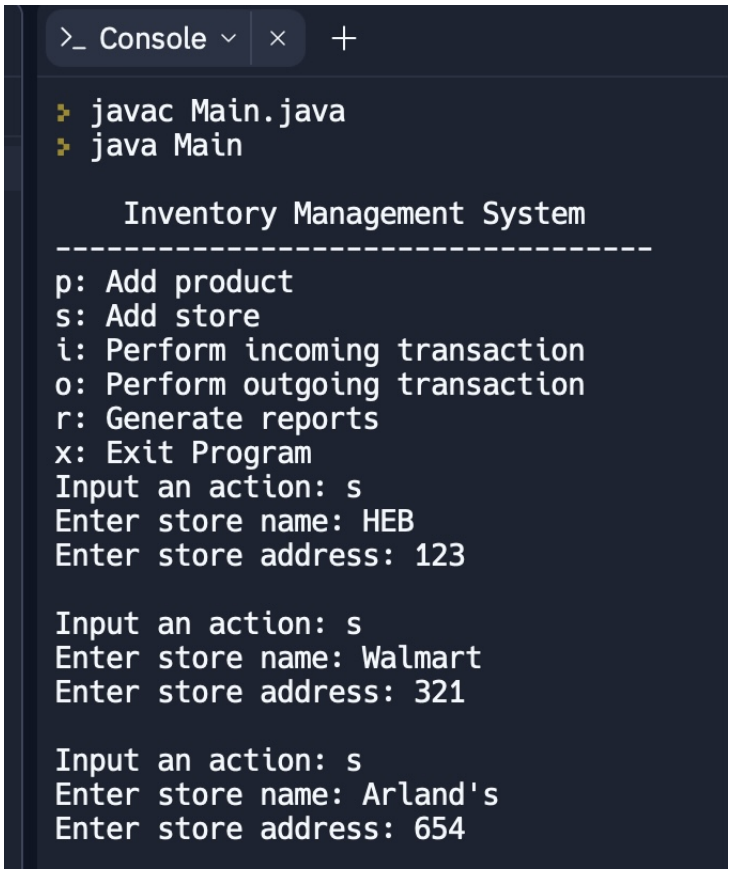
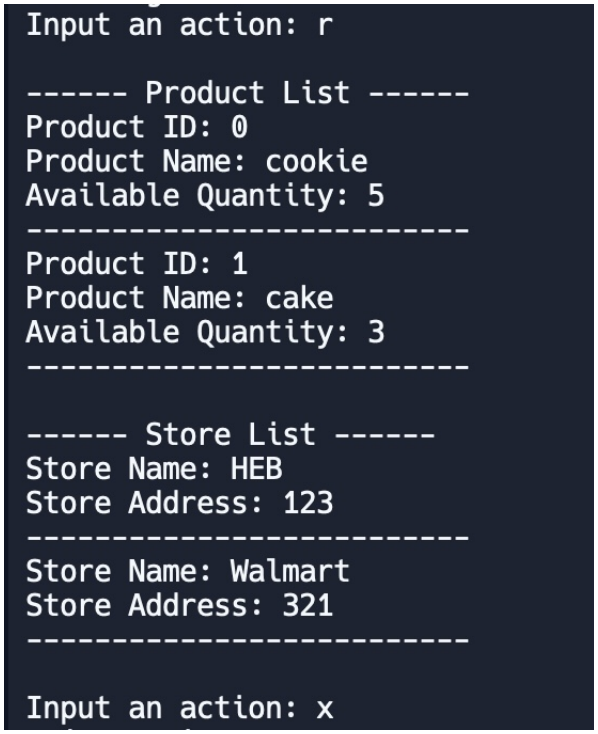


Fig. 3: fig: Screenshot of menu from console
The user can then enter 's' to Add store, and is then prompted to enter store name and address.

```
Input an action: i
Available Stores:
0: HEB
1: Walmart
2: Arland's
Select a store: 1

Available Products:
0: cookie 5
1: cake 3
2: chocolate 7
Select a product: 2
Enter product amount: 5
Input an action: o
Available Stores:
0: HEB
1: Walmart
2: Arland's
Select a store: 1
Available Products:
0: cookie 5
1: cake 3
2: chocolate 2
Select a product: 0
Enter product amount: 3
Input an action:
```

Fig. 4: fig: Screenshot of menu from console
The user can then enter 'i' to specify the incoming transaction. Products and Stores that were previously added are listed. Once the choices are made, that transaction is saved in the system.



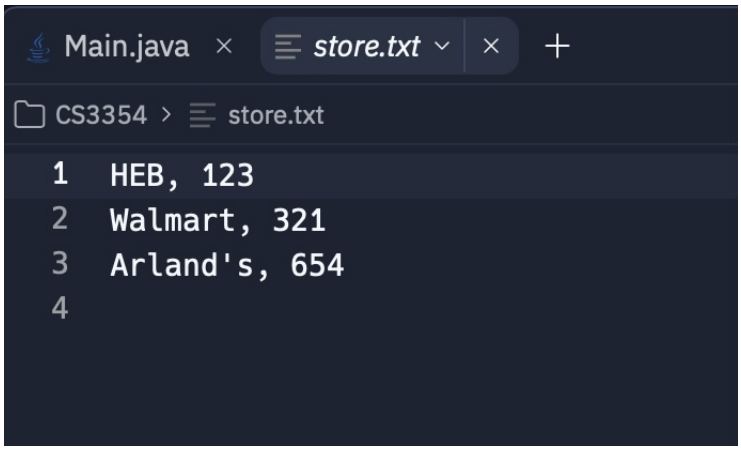
```
Input an action: r

------ Product List ------
Product ID: 0
Product Name: cookie
Available Quantity: 5
--------------------------
Product ID: 1
Product Name: cake
Available Quantity: 3
--------------------------

------ Store List ------
Store Name: HEB
Store Address: 123
--------------------------
Store Name: Walmart
Store Address: 321
--------------------------

Input an action: x
```

Fig. 5: fig: Screenshot of menu from console
The user can then enter 'r' to see Product List and Store List reports.



```
Main.java   store.txt
CS3354 > store.txt
1  HEB, 123
2  Walmart, 321
3  Arland's, 654
4
```

Fig. 6: fig: Screenshot of menu from console
This output txt file displays the stores in the system and the address.

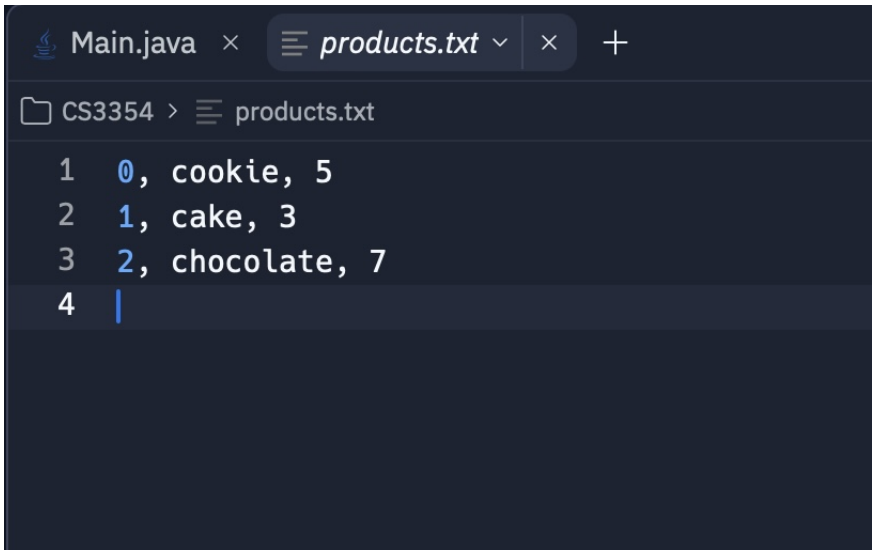


Fig. 7: fig: Screenshot of menu from console
This output text file displays the list of products in the system and their quantity.

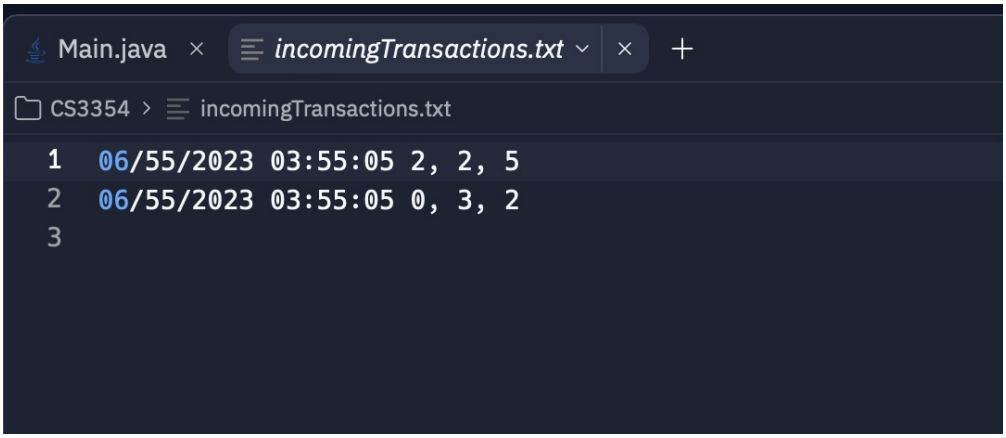


Fig. 8: fig: Screenshot of menu from console
This output txt file displays the incoming transactions with the date, productID and quantity.

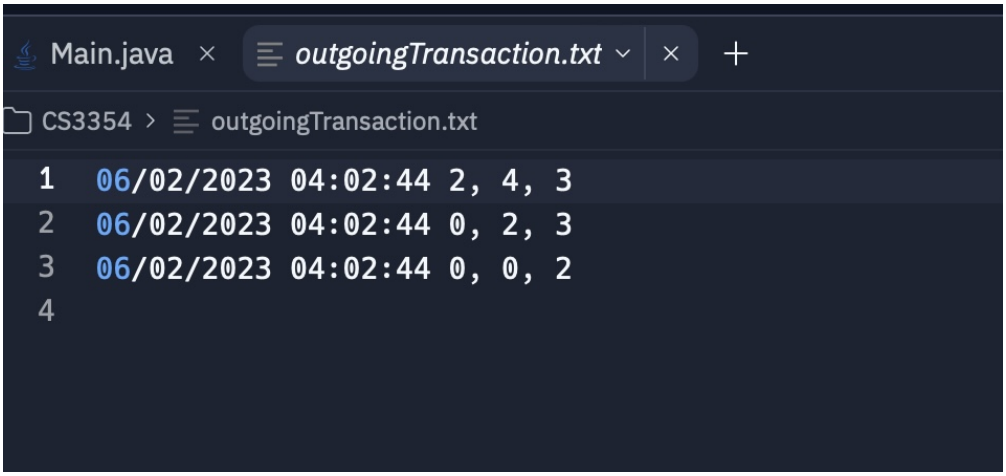


Fig. 9: fig: Screenshot of menu from console
This output txt file displays the outgoing transactions with the date, productID and quantity.

12 Group Participation

- Ashley Cook primarily worked on ensuring Main could function and provide the initial menu to the screen; Add store and Add product menu options worked; and worked on ensuring report files were created.
- Sarah Gonzales primary work on the transaction code files, making sure data was being successfully saved to the system, and outlining the report.

13 Conclusion

In summary, the creation of this Inventory Management System needed a comprehensive understanding of Java classes, constructors, and objects. Each class was dependent on the other to perform basic tasks. Many of our code bugs arose from incorrect parameters and missing objects. After fixing our mistakes, we were able to create a system without any errors.

14 References

1. https://www.geeksforgeeks.org/hashmap-class-methods-java-examples-set-1-put-get-isempty-size/
1. https://www.javatpoint.com/banking-application-in-java