Justin Hurd

Professor Arias

Software Development 1

Due: 10 May 2019

This is OK, but a little too simple. 42/50

Final Project Write-Up

Abstract. A brief description of the work.

The entire project is a composite of Java classes that are enough to describe a financial transaction on an Online Retail Application. Some classes the project contain are Account, Transaction, Bank, and Item. For this project, I had created my own separate database to use by simply creating classes that act as a database and contain only a few accounts and two transactions. The Account class contains an account number, username, password, and bank account. The Item class contains the item ID number, item name, and the price of the item. The Transaction class contains the transaction ID number, a cart of items, and the account that is making the transaction.

Introduction. Describes the motivation of the work and provides an outline of the paper.

Being someone who makes a lot of online retail purchases, I have always been fascinated with how online transactions occur, are actually stored, and how seemingly simple it is for customers to have products delivered right to their door. Online retail has revolutionized the consumer economy – and consequently the producer side of the economy as well. I hope to one day possibly use this knowledge of creating my own online retail application to assist in my dream of creating a side business that sells various merchandise to people worldwide. In this report I will give a detailed description of the system as a UML, the program requirements, a literature survey, a user manual, and a conclusion.

Detailed System Description. Describes what the system does and how specific users interact with it. It also describes how classes interact (in UML).

This system specifically helps the user access the store by signing in. The user types the correct username and password into the store and causes the items and prices to appear. After the prices appear, the user is prompted to enter the specific numbers that correlate with a specific item. Each item has a price linked between the item number and the item name. As you enter in the numbers, the system creates a cart that you can keep adding to. Once you type number 11, which is "None" then the system prompts you that your purchase either was successful or not successful due to the funds that are in your linked bank account. The program will show the total price for the items in your cart and then it will show how much money is left in the account after the purchase.

Account
accountID: int username: String password: String -bankAccNum: int
+account(user: String, pass: String, bankAcc: int) +getPass( ):String +getUser( ): String

Item
itemID: int itemName: String price: double
+Item(id: int, name: String, p: double) +getItemId( ): int +getItemName( ): String +getItemPrice( ): double

Transaction
transactionID: int <Item>: ArrayList account: Account

+Transaction(acc: account)
+addToCart(item: Item)
+getTransactionPrice( ): double
totalPrice: double

Bank
bankName: String bankAccountNumber: int bankBalance: double
+Bank(bname: String, baccountID: int, balance: double) +getBankName( ): String +getBankAccount( ): int +getBankBalance( ): double

Requirements. Describes what the specific details of the problem that the system is addressing.

The specific problems that this system is addressing is the aspect of how to incorporate the database with the java programming. This system uses a quick and easy way to access the purchase process throughout the specific online retail store. This program is the non-digital side of the store. When you see an online retail store, you're able to see the item you are buying and in multiple colors. My retail store, you just see the name of the item and its price. If this was further developed, someone may be able to use this to combine with their webpage.

Literature Survey. Describes other work that has been done to address the same or similar problems.

The work that has been done to address these problems contain my four java classes: Account, Bank, Transaction, & item. There is one extra class used to test my program, going by the name of Test. I found that the use of figuring out how to combine the price and item so that when you purchase the item, the money would be pulled from your account, leaving only the users account balance. This is an easy system to use, definitely with room to advance forward but fulfills the needed requirements in order to have a running system for an online retail store.

User Manual. Briefly describes how the system should be used.

- Welcome to my store!
- Enter usernames: bwayne, clarkkent, hulk
  - If username is not in system: "Not a member " appears
- Enter correct passwords: batman!!, 21super12@@, greenman321$
  - IF wrong password is entered: incorrect password appears. You have to rerun system if that appears
- What do you want to buy?
  - Type in specific number of the displayed item that you want to add to your cart.

    - You can enter as many as you want until you type in 11 – none. The purchase will end.
- The program now displays the total for the user
    - The bank funds that are left
    - Purchase either: Successful or Unsuccessful due to how much is in the bank account.

Conclusion. Summarizes the goals accomplished by the system.

Since I am a person who really enjoys online shopping, I believe that if my system was incorporated into an actual database for an online retail store, that this would work out in a simplified manner. If someone further advanced my work and combined a visual feed into this project, the system would work in a great way. Now that I see how financials may be stored, I had learned much more than I had expected to. The system using my five classes connects them all really well and provides the information needed for fulfilling my online retail store.