Daniel Bell

PA-1 Use Case Realization

Sales501

1. User Selects and Operation

The program boots up running the class **Main** which calls **Sales** to present the user with a number list of possible operations. The user enters in a numeric value corresponding the desired operation. If the user input is invalid, an error message will be displayed and the function will run again.

1. Create Sale Transaction

The function CheckOut will create a new **Transaction** object and ask the user for the current *month* followed by the *item* and *price*. If the value of the price is invalid, the program will display an error message and ask for a new *item* and *price*. Upon successful input of an item and a price, the user will be asked if another items needs to be added. If the user enter ‘y’ the function will repeat. If the user answers ‘n’ the function will generate a total, show the receipt, and save the transaction in a list of *purchases*. Any other input at this stage will display an error message and the user will again be asked if another item needs to be added. Successful completion of this method will return to the main screen.

1. Return an Item

The function ReturnItem prompts the user for a *receipt\_num* and an *item*. An invalid input here will result in an error message. The program then searches through the list of purchases and determines if the item has been purchased, for how much, and whether or not the purchased item had been discounted from a rebate. If the item has been discounted, the program shows an error message and the function ends. Otherwise the *item*, *receipt number*, and *refund\_val* are displayed and the item is then removed from the list of returnable items..

1. Enter Rebate

The Function GenRebate will create a new **Promotion** object and the user will be promoted for the rebate month followed the item name and discount. An invalid discount value will display an error message and a new item name and discount will be asked. A valid entry will store the values in *month*, *name*, and *discount*, respectively and the whole object will be stored in *rebate*. The user will then be returned to the main screen. If a rebate is created when one already exists. The value will be overwritten.

1. Generate Rebate Check

The function GenChecks will read in the values from the *rebate* field. Invalid or empty values will display an error message and the user will be returned to the main screen. The computer will then locate all receipts matching the promotion’s month and item and display the dollar amounts the need to be refunded to each customer. Successful completion o f this method will return the user to the main screen.