**Project 2**

**SPECIFICATION AND TEST CASES**

|  |  |
| --- | --- |
| Specifications | Test Case |
| Read the CSVs into main. | Test Case 1, Part 1: Demonstrate that the product list is read.  Test Case 2 & 3, Part 1: Demonstrate that the sales data is read. |
| Store the data from the product in a product ADT, sales data into line items ADT, and line items into a receipt ADT. Line items, and receipts should have the capacity to generate their totals. | Test Case 1, Part 2: Output one product.  Test Case 2, Part 2: Out one sales line item.  Test Case 3, Part 2: Out one complete sales receipt. |
| Create a complete sales inventory of all products sold, by scanning the receipts, and have the capacity to print the sales inventory. | Test Case 4, Part 1: State when the inventory is being populated, and when complete.  Test Case 4, Part 2: Output a sales inventory that accurately matches the totals of the sales data. |

|  |
| --- |
| salesReceipt |
| -saleID: int  -customerID: int  -saledateTime: Date  -items: Linkedlist <saleItem>  -total: Double  -df: DecimalFormat |
| +salesReceipt(cID: int, sID: int)  +setSaleID(sID: int)  +getSaleID(): int  +setCustomerID(cID: int)  +getCustomerID(): int  +setSaleDateTime()  +getDate():Date  +addItem(id: int, u: int, currentInventory: LinkedList<product>)  +calculate()  +printSalesReceipt() |

|  |
| --- |
| Main |
| -inventoryCSV: String  -salesCSV: String  -inventory: LinkedList<Product>  -sales: LinkedList<salesReceipt> |
| +inventory(fileName: String): LinkedList<Product>  +receipts(filename: String):LinkedList<salesRecipet>  +salesInventory(totalSales: LinkedList<salesReceipt>) |

**Project 2**

**UML DIAGRAM**

|  |
| --- |
| saleItem |
| -Productid: Int  -units: Int  -productDescription: String  -unitPrice: Double  -lineTotal: Double  -df: DecimalFormat |
| +saleItem(id: int, u: int, currentInventory: LinkedList<product>)  +setProductID(id: int)  +getProductID()  +setUnits(units: Int)  +getUnits(){query}  +setUnitPrice(double p)  +getUnitPrice()  +getLineTotal()  +printSaleItem() |

|  |
| --- |
| Product |
| -productID: Int  -description: string  -price: Int |
| +product(id: int, d: string, p: double)  +setDescription(d: string)  +getDescription()  +setProductID(id: int)  +getProductID()  +setPrice(p: double)  +getPrice()  +printdetails() |

**Project 2**

**PSUEDO CODE:**

Define Variables

-File names

Populate linked list of products Inventory via a method that contains a buffered reader

-Use a loop to keep adding products to a linked list of Product objects until complete

Print test cast 1: Part 2: A Product

Populate linked list of Receipts with a method containing a buffered reader

-Print Test Case 1 Part 1, half at start, confirmation at the end.

-Use Register 0 to store the complete sales data outside of loop, populate all productid, but no units(nothing to populate with yet)

Use a loop that continues until all data is read

-add line items to the current receipts items list until customer ID changes

-Create Each New Receipt when customer ID changes

Print test case 2, part 2: Line items

Calculate receipt total

Print test case 3, part 2: Print total receipt

Populate Receipts by calling a method that scans sales receipts, and adds each line item to the appropriate productid sum

Calculate Sales total

Print test case 4: Complete Sales Inventory