**Grocery Store Management**

**Phase 4 Full Report**

**MIS 307 Group Code: PB8**

**Group members:**

**Yen Vo, Michael Ammons, Sam Prisco**

1. **Problem Statement**

A local grocery store is in need of an improved system to track and update the inventory and customer information. The grocery store requires a system that keeps track of the store’s customers, daily sales, monthly sales, and information of the cashiers. Databases are used to store and update customer ids, cashier ids, and receipt ids. This data will be used to track sales and returns. Along with those, data of customers and cashiers will be stored and searchable. The manager needs to be able to add and remove cashiers for employees that are new or have left. The employees need to be able to add and remove products to keep accurate information on the inventory. The employees also need to be able to add customer information to add, remove, and find customers from the database. The database will provide a much more efficient tracking of the inventory and sales in the store. The data that is kept will be used to provide a monthly sales report for the store.

1. **Classes and Methods**

**Classes**

**GroceryProgram**- main class that allows store’s employees use the program. It manages all CustomerManager, SaleManager, ReportManager, ProductManager, and CashierManager. There are different menus based on the selection of users. After selecting a menu, a user can access the information he or she wants to know.

**CustomerManager**-class modifies customer’s information. In this class, grocery’s employees can **print** a list of customers, **add** and **search** customers by name. This will help grocery’s staff easily maintain and retrieve customer’s information.

**SaleManager**- this class is used to **print** a list of receipts, **add** a new receipt and **search** a receipt based on the receipt's id. The class helps employees find item or receipt numbers quickly when customers want to return their items to store.

**ReportManager**- class prints **daily** and **monthly** sales for the store. For daily report, employee will enter a day, month, and year. For report sales by month, the user will need to enter month and year. After getting all information from users, the program will generate and show receiptID, name of customers, and calculate total sale on that day or month. This class helps store tracking total sales of the store.

**ProductManager**-class maintains and tracks products in store. This class allows employees to **add** products and **search** products based on the product's name. This will help employees track their product level in the store and update the number available product to customers.

**SimpleDataSource**- external class that creates a database connection.

**DatabaseManager** – class initialize grocery database , all of create tables (Products, customers, Receipt, Receipt\_Details) queries will store in this class

**Class Method**

**GroceryProgram**

**main**-. The method connects to ProductManager, SaleManager, CustomerManager, and Report Manager. It provides a different menu to users. There are 11 menus in the main method. This will help users access the information they want easily.

**CustomerManager**

**InitializeData**-

**printAllCustomers** - print all of customers information, this will help manager or cashier know

**addCustomer**- a customer is added when they.

**searchCustomer**-a customer can be searched by name, id or phone number. This method can be helpful when a customer returns an item, or they want to update their information

**SaleManager**

**addReceip**- this method will help cashiers add new receipts for each customer because each customer has each transaction when they buy products at store; and a receipt is needed to print each time.

**printReceipt -** this method helps employees print all the receipts they made in a day. This will help employees compare the receipt ‘s information with the receipt number they want to search.

**searchReceipt-** this method helps cashiers find the right receipt to the right customer. It ensures that cashiers find the correct price and correct returned items when customers return their stuff at the store.

**ReportManager**

**printDailySale -** This method prints the amount sold at the end of the day for the managers to see trends in the daily data

**printMonthlySale -** This method prints the amount sold at the end of the month for managers to see trends in data.

**ProductManager**

**addProduct -** the method allows employees or cashiers to replenish products when they run out of stock. This will help provide enough products to adapt customer’s demand.

**searchProduct**- the method allows employees to delete a product from the system when it was sold. This method provides accurate information about the number of available products in store.

**DatabaseManager**

**creatDBTables**- the method creates product, customer, receipt, and receipt\_detail tables.

**dropTables**- the method will help drop table if it exists

**4. Table of Components**

|  |  |  |  |
| --- | --- | --- | --- |
| Required components | Corresponding classes | Corresponding methods | Corresponding SQL tables |
| Reading/Writing form text files | -CustomerManager  -ProductManager  -SaleManager | InitializeCustomerData  InitializeProductData  initializeReceiptData  initializeReceiptDetailsData | -Customer  -Product  -Receipt  -Receipts\_Details |
| * Relational databases and SQL   Use of multiple classes/ or multiple tables | DatabaseManager  GroceryProgram  -CustomerManager  -ProductManager  -ReportManager  -SaleManager | -createDBTables  -dropTables  -main method- Initialize Database Connection- to all th Managers  1. PrintAll Customers  2.addCustomer  3.SearchCustomer byName  4.CreateQueryInsert Customer  2.PrintAllProducts  3.addProducts  4.SearchProductByName  5.CreateQueryInsertProduct  1.PrintAllReceipts  2.PrintReceiptByID  3.Report Sale By Date  4.ReportSale ByMonth  1.addReceipt  2.addReceiptDetail  3.searchReceiptById  4.createQueryInsertReceipt | -Customer  -Product  -Receipt  -Receipt\_Details  -Customer  -Product  -Using ‘Join’ query to join multiple tables together  -ReceiptDetails  -Receipts  -Customer  - Receipt  -Receipt\_Detail |
| Reading and processing html files from web pages | LocationLookUp | main |  |
| Prompt user enter  A location from providing HTML file ( Jamestown or Grafton) | LocationLookUp | main |  |
| Extract store’s address, location, and hours | LocationLookUp | main |  |

**4. Implementation Results**

The program is implemented so that the user can search, find, and show receipts of customers and find and add available products. The user can search for the receipt of the customer and see all of the receipts of that customer. The program is also designed to show you all the products that are in the store, and a list of all the customers that the store has. The program can also show all of the receipts that were created on a certain date, or in a specific month. This allows the program to create daily and monthly sales of all products for the store. The program also allows for customers to be able to search for a store by town name. If a store is found in that town, the store's phone number, address, store hours are displayed. We believe that the program does implement the processes as they were initially planned.

**5. Discussion**

All of our goals have been met for this project. We set out to create a user friendly program that allows for a business to be able store information about their inventory, customers, and sales. The program we created does all three. As mentioned above, the program offers a wide variety of capabilities that could be useful for a store and the customers. The program is efficient and user friendly. Multiple improvements could be made to increase the program's benefits to a store. The first is an inventory tracking class. This class would be able to automatically track and update the inventory and send a notice when inventory is low. Another improvement would be a goal setting program. It would allow the user to set a daily or monthly goal for sales and give updates on how close they are to achieving those goals. The last idea of an improvement is a customer rewards program that gives the customers rewards for spending money at the store. This would provide more incentives for shopping there. While there are improvements that could be made, the overall goals we set were met and the design and implementation of the program went smoothly.