University of Engineering and Technology, Taxila

Department of Software Engineering

Project

Computer programming

Submitted to:

Dr. Kanwal Yousaf

Muhammad Arslan Fareed

24-SE-36

Khuram Sohail

24-SE-72

Grocery Store Manager

Introduction

The Grocery Store Management System is a console-based application developed to manage the inventory of a grocery store efficiently. The primary goal of this project is to ease the process of adding, updating, displaying, and deleting items from the inventory, as well as calculating the total cost of purchased items. This application is designed to provide a simple and effective solution for small business owners and retail employees to handle daily inventory tasks easily.

Key Features

1. **Load Inventory**:
    - Loads existing inventory data from a file when the program starts, ensuring you have all previous data without needing to re-enter everything.
2. **Save Inventory**:
    - Saves the current inventory to a file before exiting the program, keeping your data safe and ensuring you don't lose any changes.
3. **Add Item**:
    - Allows users to add new items to the inventory by inputting the item's name, price, and quantity, then stores them in arrays.
4. **Display Inventory**: ○ Shows a list of all items in the inventory along with their details, providing a clear overview of the stock and making it easy to check what items are available and their current status.
5. **Update Item**:
    - Enables users to update the quantity and price of existing items in the inventory, ensuring information remains accurate.
6. **Delete Item**:
    - Provides functionality to remove items from the inventory if they are no longer needed.

7. **Calculate Total Cost**:
    - Calculates the total cost of items purchased by the user and updates the inventory accordingly, simplifying the checkout process and ensuring that inventory levels reflect recent sales.
8. **Interactive Menu**:
    - Provides an easy-to-navigate, user-friendly menu to access the various functions, making the application easy to operate even for users with limited technical skills.

Libraries used

1. **<iostream>**: This library is used for input and output operations. It provides the functionalities cin and cout for reading from and writing to the console, which allows the application to interact with the user.
2. **<fstream>**: This library is used for file input and output operations. It provides classes ifstream for reading from files and ofstream for writing to files. This enables the application to save and load inventory data from a file, ensuring data persistence.
3. **<string>**: This library is used for handling string operations. It provides the string class to manipulate sequences of characters, facilitating the storage and manipulation of item details like names, prices, and quantities.

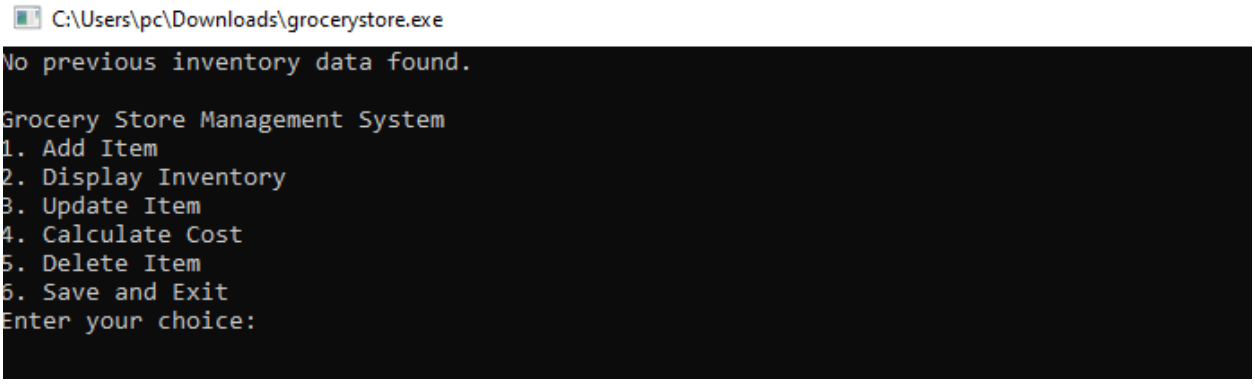
Functions used

1. **loadInventory():**
    - Loads inventory data from a file when the program starts. It reads the number of items and their details (names, prices, quantities) from the file into arrays.
2. **saveInventory():**
    - Saves the current inventory data to a file before exiting the program. It writes the number of items and their details from the arrays to the file.
3. **addItem():**
    - Adds a new item to the inventory. It asks the user for the item's name, price, and quantity and then stores these details.

4. **displayInventory():**
    - Displays all items currently in the inventory. It prints each item's name, price, and quantity.
5. **updateItem():**
    - Updates the quantity and price of an existing item. It asks the user to enter the name of the item to update and then updates its quantity and price based on user input.
6. **deleteItem():**
    - Deletes an item from the inventory. It asks the user for the item's name and then removes the item from the arrays.
7. **calculateTotalCost():**
    - Calculates the total cost of items purchased by the user and updates the inventory. It asks the user for the names and quantities of items to purchase, adds up the total cost, and adjusts the quantities in the inventory.

Console UI

Main Menu



Add item

```
Enter your choice: 1

Enter item name: Apple
Enter item price: 20
Enter item quantity: 5
New item added!
```

Display Inventory

```
Enter your choice: 2

Inventory:
-----------------------------
Name    Price   Quantity
-----------------------------
Apple   20      5
```

Update Item

```
Enter your choice: 3

Inventory:
-----------------------------
Name    Price   Quantity
-----------------------------
Apple   20      5

Enter item name to update: Apple
Enter new quantity for Apple: 7
Enter new price for Apple: 25
Item updated!
```

Display Inventory

```
Enter your choice: 2

Inventory:
-----------------------------
Name    Price   Quantity
-----------------------------
Apple   25      7
```

Calculate cost

```
Enter your choice: 4
Enter item name and quantity to purchase (type 'done' to finish):
Item name: Apple
Quantity: 4
Item name: done
Total cost: 100
```

Display Inventory

```
Enter your choice: 2

Inventory:
-----------------------------
Name    Price   Quantity
-----------------------------
Apple   25      3
```

Delete Item

```
Enter your choice: 5

Inventory:
-----------------------------
Name    Price   Quantity
-----------------------------
Apple   25      3

Enter item name to delete: Apple
Item deleted!
```

Save and Exit

```
Enter your choice: 6
Exiting...
```

File Handling

File handling is used in the Grocery Store Management System to ensure that inventory data is stored persistently between sessions. This means that when the program is closed and reopened, the inventory information remains intact. Here's how file handling is implemented:

1. **Loading Inventory Data**:
    - The loadInventory() function is responsible for reading inventory data from a file named "inventory.txt".
    - When the program starts, this function opens the file, reads the number of items in the inventory, and then reads the details of each item (name, price, quantity). ○ This data is stored in arrays, allowing the program to work with the existing inventory data right from the start.
    - If the file does not exist, it notifies the user that no previous inventory data was found, a fresh start.
2. **Saving Inventory Data**:
    - The saveInventory() function is used to write the current inventory data to "inventory.txt" before the program exits.
    - This function opens the file for writing, writes the number of items, and then writes each item's details (name, price, quantity) to the file. ○ This ensures that any changes made to the inventory during the session are preserved and will be available the next time the program is run.

How the Application Addresses the Needs of the Target Audience

The target audience for this project is **small grocery store owners** or **managers** who need a lightweight, efficient, and cost-effective solution to manage inventory. Here's how the application caters to their needs:

- **Efficiency**: Simplifies inventory tracking and purchase management.
- **Cost-Effective**: Eliminates the need for expensive inventory software.
- **Scalability**: Handles up to 1,000 items and can be expanded.
- **Reliability**: Saves data to prevent loss during application closure.

Implementation Insights

- **File I/O**: Used to load and save inventory to `inventory.txt`.
- **Arrays**: Store item details like names, prices, and quantities.
- **Validation**: Prevents errors such as duplicate entries and invalid operations.
- **Dynamic Updates**: Real-time changes in stock and cost tracking.

Flowchart