Reflection – Flower Shop Manager

This application helps run a small flower shop with different items for sale. The different functions of the application include creating and managing items, inventory, a point of sale to perform a sale or a return, retrieving sales, and returns transactions from accounts history and with bulk item maintenance csv file features.

User Functionality

Menu

1. The main menu with 4 options to select sub-menus: Register, Items, Accounts, and Quit
2. Based on the option selected the next menu will be shown or quits the application

Items

1. An items menu with 5 options to select – View Item, Update Item, Add Item, Update Inventory and Go back to Main Menu
2. View Item - User will be able to view all existing items and their attributes
3. Update Item – User will be able to update any exiting item attribute other than inventory. If the user wants to exit after providing an item as input but without updating, the user can enter the same value as before and exit.
4. Add Item – User will be able to add a new item with all attributes (Name, Color, Cost, Price, Sale pack quantity, and Inventory)
5. Update Inventory – The user is provided an option to update inventory easily as this will be a frequently used option
6. User will be able to go back to the main menu
7. Item ID is auto-generated

Register

1. A register menu with 3 options to select – Perform a sale, Perform a return, and Go back to Main Menu
2. The user should be able to perform a sale transaction by entering the Item ID and Quantity.
   a. Note: As all items can be labeled with the item ID and a barcode, the user is expected to view and enter the item id during the sale
   b. User can enter the same item multiple times or different items and quantities in one transaction
   c. If the user wants to exit and complete one transaction, the user needs to enter "0" in the item id field
   d. The inventory is reduced for the items that are sold. The inventory can go below "0" during a sale as the sale to a customer is important at the POS. All adjustments to negative inventory due to system issues or manual errors can be corrected through the inventory adjustment option.
   e. If any error is made during a sale, the item can be returned using the return option.
3. The user should be able to perform a return transaction by entering the Item ID and Quantity.
   a. Note: As all items can be labeled with the item ID and a barcode, the user is expected to view and enter the item id during the sale
   b. User can enter the same item multiple times or different items and quantities in one transaction

  c. If the user wants to exit and complete one transaction, the user needs to enter "0" in the item id field

  d. The inventory is **not** added back for the items that are returned. This is to make sure the item is inspected and evaluated if it can be re-sold. All adjustments to inventory can be corrected through the inventory adjustment option if the items can be resold.

  e. If any error is made during a return, the item can be sold and re-returned.

4. The return transaction is not linked to the original sale transaction. Returns are based on original receipt and goodwill and can be performed even without an original sale transaction.

Accounts

1. An Accounts menu with 3 options to select – View all transactions, View today's transactions, and Go back to Main Menu

2. The user will be able to view all transactions (sales and returns totals) from accounts history

3. User will be able to view today's transactions (sales and returns totals) from accounts history. The system date will be considered as today's date.

4. There are two system-generated values for each transaction – Sequence and Transaction-ID.

  a. The Sequence number is a unique key and will not be repeated. Each line item scanned and sold is entered with a unique sequence number.

  b. All items in a single transaction will have the same randomly generated transaction ID.

  c. The Tran-ID identifies one transaction and can be present across multiple sequence ids.

  d. This Tran-ID is used only to check the different items in one transaction on a particular day and similar timeframe. The Tran-ID may be repeated across days due to its randomness and still, we can easily differentiate the transactions.

Bulk Maintenance:

User can upload a list of new items from a csv file from the same folder as the script. File attributes are Item Name (str), Color (str), Cost (upto 2 decimals), Price (up to 2 decimals), sale pack quantity (integer), Item inventory quantity (integer). If the file is uploaded again, duplicate entries with new item ids will be created. User needs to specify the file name with extension like bulkitem.csv to be uploaded.

User can also extract all available items from the application and the file is generated in the folder of the script with the name Itemextract.csv

Storage

Two pickle files (Items.pkl and Accounts.pkl) are required for the application. These will be automatically generated with no data when the application does not find these files. If there are existing pickle files with items and/or accounts data, these can be kept in the same directory as the program and the script will use them without creating new ones.

Class Bulk():

Instance Methods:
__init__: Initializes the Menu and instance different classes based on user selection

upload: Uploads new items from a user specified csv file

extract: Extract all available items into a csv file

Flower Shop Manager

Class Items():

Instance Methods:
__init__: Loads the stored objects from pickle file and displays item menu (View item, Update item, Add item, Update inventory, Back to Main menu)

getitems: Shows all available items

setitems: Creates or updates item entries

additems: Gets user input for new item attributes

upditems: Gets user input for changing item attributes

getinv: Returns the available inventory of the items

setinv: Updates inventory of the items

itemprint: Display the items to the user in a specified format

Class Menu():

Instance Methods:
__init__: Initializes the Menu and instance different classes based on user selection

Class Accounts():

Instance Methods:
__init__: Displays accounts menu (Display all transactions, Display today's transactions, Back to Main menu)

gettran: Loads the transactions from a pickle file and shows the available transactions and totals (Sales and Returns)

settran: Creates sales and return transactions and saves into a pickle file

Class Quitclass():

Instance Methods:
__init__: Calls the instance method leave to exit

leave: Exits the application

Static methods:
saveitems: Saves the items dictionary input parameter into a pickle file

saveaccounts: Saves the accounts dictionary input parameter into a pickle file

readitems: Reads the pickle file and returns an items dictionary

readaccounts: Reads the pickle file and returns an accounts dictionary

gettime: Returns current date and time

clearscreen: Clears the previous screen output

Class Register():

Instance Methods:
__init__: Loads the stored objects from pickle file and displays register menu (Perform Sale, Perform Return, Back to Main menu)

sale: Get user inputs to perform a sale transaction

return: Get user inputs to perform a return transaction