A Simple Point-of-Sale (POS) System

Start Assignment

Due Dec 20 by 11:59pm **Points** 100 **Submitting** a file upload
Available Nov 30 at 12am - Dec 20 at 11:59pm 21 days

A Simple Point-of-Sale (POS) System

OBJECTIVE

The objective of this assignment is to get you familiar with all (mostly) Python basics and some advance concepts, which include the proper syntax, Python input/output, loops, lists, dictionaries, classes, objects, inheritance. After performing this assignment, you should be able to master these python concepts. You NEED to use either Spyder or PyCharm for this assignment.

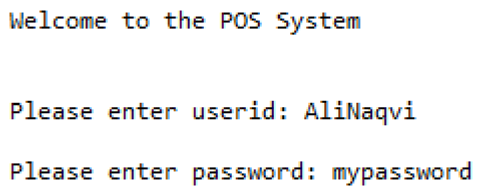
TASK

In this assignment, the overall task is to develop an application that does three things 1) verify that the right user enters the system, 2) provide ability for the cashier to perform sales type activities like selling an item, accepting returns, records sales, handle payments. etc. and 3) manage the backroom stock inventory by placing replenishment orders etc.

Study the **POS Project Requirements** below. You will write your application as a console application which means it will not have a GUI (Graphical User Interface).

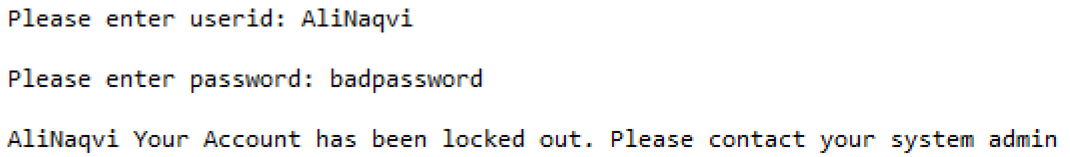
1)

1.1) In this POS system, a cashier will be asked to log in the system using his/her user id and password. (Maximum tries of incorrect passwords or userId is 3 after which the system will not allow this user to logon). The system logs in the cashier if proper credentials are provided otherwise a message is displayed to reenter the credentials. (If the user id or password is wrong, the cashier will be asked to re-enter). **(5 points)**



```
Welcome to the POS System

Please enter userid: AliNaqvi

Please enter password: mypassword
```

1.2) If the wrong combination of either the UserID or the password is supplied the system will lockout the user for entering the system **(5 Points)**



```
Please enter userid: AliNaqvi

Please enter password: badpassword

AliNaqvi Your Account has been locked out. Please contact your system admin
```

2)

2.1)You need to load the inventory data from a file called RetailStoreItemData file into your newly created item object. Item objects need to become part of a collection. You could use a dictionary to store the key/value pairs. Use the UPC as the key and store the Item object as the value to be stored in the dictionary. **(10 points)**

2.2) This file is provided to you with this assignment. This file contains all the items that are sold in your store. (Please look at the file to understand the data). The file has the following data.

Field	**Description**
UPC	UPC (unique key of the item)
Description	The description of the item
Item_Max_Qty	Max quantity that the shore should hold
Order_Threshold	New order needs to be place for replenishment once items_on_hand drop below the order_threshold

replenishment_order_qty	When a new order is placed a minimum of replenishment_order_qty has to be ordered
Item_on_hand	total items in store
Unit price	item price
Order_placed	This field indicates that the order has been placed for the items by placing Y or N in the field

Create a new class called Item which will have attributes like below: **(10 Points)**

UPC

Description

Item_Max_Qty

Order_Threshold

replenishment_order_qty

Item_on_hand

Unit price

Item class will have methods like below:

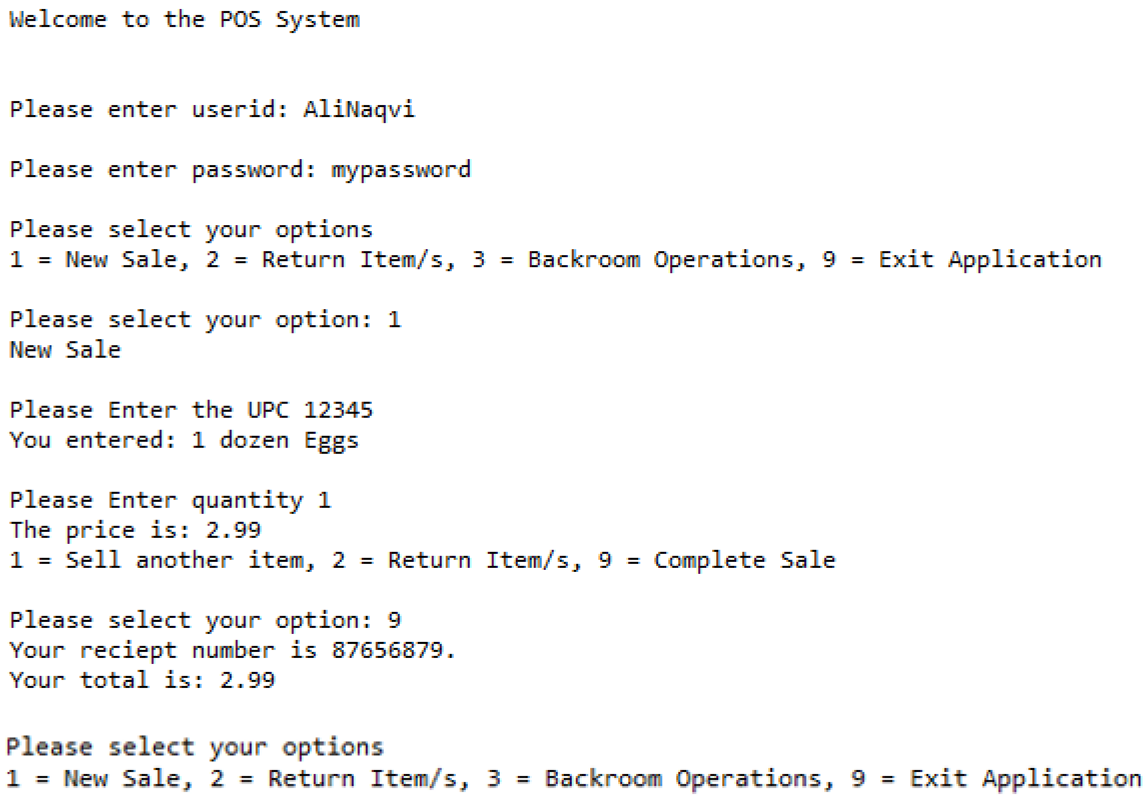
UpdateUnitOnHand(numberOfItems) #This method will update the unit_On_hand by either subtracting (in the case of sales) and adding (in the case of returns) to the file

3) POS Project Requirements

Point of Sale System is developed to support supermarket-type store operations. In particular software shall:

3.1) Allow the cashier to start a new sale and allow add/remove items to a new sale. **(5 Points)**

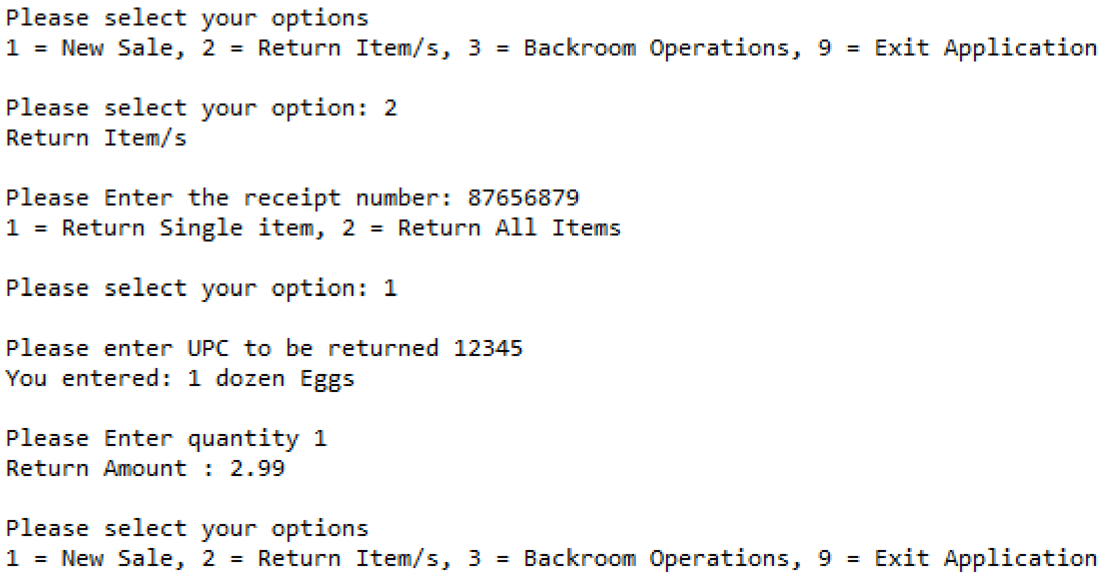
3.2) Once all items are added to the sale the cashier will request for cash to finalize the sale. **(20 Points)**



```
Welcome to the POS System


Please enter userid: AliNaqvi

Please enter password: mypassword

Please select your options
1 = New Sale, 2 = Return Item/s, 3 = Backroom Operations, 9 = Exit Application

Please select your option: 1
New Sale

Please Enter the UPC 12345
You entered: 1 dozen Eggs

Please Enter quantity 1
The price is: 2.99
1 = Sell another item, 2 = Return Item/s, 9 = Complete Sale

Please select your option: 9
Your reciept number is 87656879.
Your total is: 2.99

Please select your options
1 = New Sale, 2 = Return Item/s, 3 = Backroom Operations, 9 = Exit Application
```

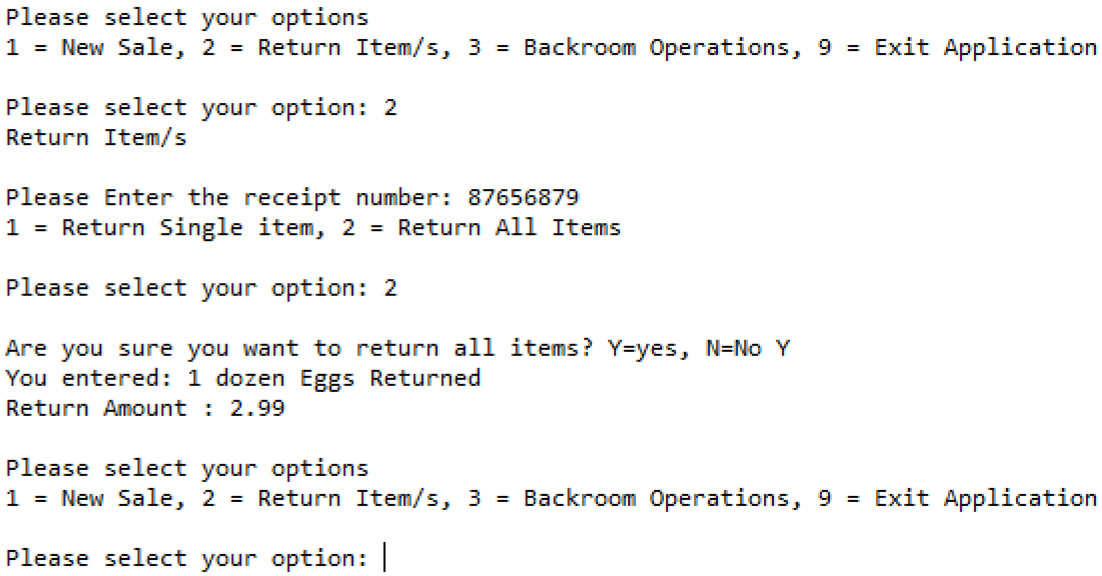
3.3) For returns - Support cancellation of the entire sale as well as return of an individual item. **(20 Points)**

Single item return use case:



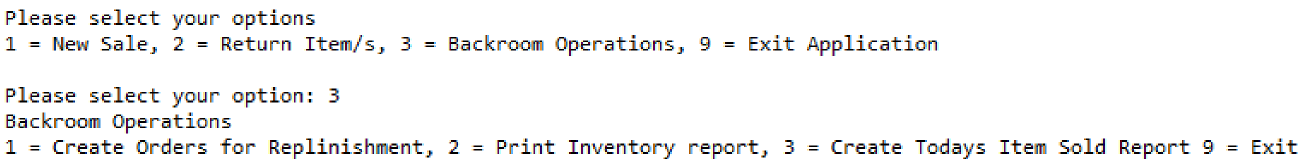
```
Please select your options
1 = New Sale, 2 = Return Item/s, 3 = Backroom Operations, 9 = Exit Application

Please select your option: 2
Return Item/s

Please Enter the receipt number: 87656879
1 = Return Single item, 2 = Return All Items

Please select your option: 1

Please enter UPC to be returned 12345
You entered: 1 dozen Eggs

Please Enter quantity 1
Return Amount : 2.99

Please select your options
1 = New Sale, 2 = Return Item/s, 3 = Backroom Operations, 9 = Exit Application
```

All items returned use case:



```
Please select your options
1 = New Sale, 2 = Return Item/s, 3 = Backroom Operations, 9 = Exit Application

Please select your option: 2
Return Item/s

Please Enter the receipt number: 87656879
1 = Return Single item, 2 = Return All Items

Please select your option: 2

Are you sure you want to return all items? Y=yes, N=No Y
You entered: 1 dozen Eggs Returned
Return Amount : 2.99

Please select your options
1 = New Sale, 2 = Return Item/s, 3 = Backroom Operations, 9 = Exit Application

Please select your option: |
```

4) Backroom Operations



```
Please select your options
1 = New Sale, 2 = Return Item/s, 3 = Backroom Operations, 9 = Exit Application

Please select your option: 3
Backroom Operations
1 = Create Orders for Replinishment, 2 = Print Inventory report, 3 = Create Todays Item Sold Report 9 = Exit
```

4.1) Update the Item_on_hand inventory field once an item has been sold or returned. Example of inventory: subtract or add number of items sold from the master file (RetailStoreItemData) which contains Item_on_hand. **(5 points)**

4.2) Support inventory management – Write a method that would create an order file to replenish items in the store by looking at the current item_on_hand and the Order_threshold of the items. If the item_on_hand is less than or equal to the Order_threshold then you should order the items for the set quantity in the **(5 points)**

5) Support report generation:

5.1) Inventory report : listing off all inventory items with name, quantity, threshold, and quantity of items in pending order state. Pending orders means – the order has been placed however the items have not been received as yet to be put on the shelves or backroom. i.e. simply the order file that you generate nightly for ordering items for your store. **(5 Points)**

5.2) Create a report of all items sold for "Today" and calculate the total sales **(5 Points)**

5.3) The system will keep track of the items and the total sales amount at the register. You could do this by simply appending the sold items and the dollar amount into a file. **(5 Points)**

You need to submit your python code as well as the screen shots of all the functionality that you have implemented. (best way is to cut and paste all screenshots in one word document)