Luca du Toit

20220798

PRP SUMMATIVE PROJECT

Code Screenshots:

Full Overview:

```
from Table import tableDesign #importing table object from tableDesign

#creating 6 table objects for each table and placing them in a list
table1 = tableDesign("Table 1")
table2 = tableDesign("Table 2")
table3 = tableDesign("Table 3")
table4 = tableDesign("Table 4")
table5 = tableDesign("Table 5")
table6 = tableDesign("Table 6")
tableList = [table1, table2, table3, table4, table5, table6]

def userLogin(): #prompting the user for login details ···

def getLogins(): #function used for reading login text file and giving an output in list format ···

def getStock(): #function used for reading stock text file and giving an output in list format ···

def landingPage(): #Main landing page that shows all menu options and sends the user to correct function according to selection made ···

def availableTables(): #Used for the first option, checks tables that arent occupied and lets waiter choose their table. ···

def changeCustomers(): #used for the second option which is changing the amount of customers at a table ···

def addToOrder(): #used for adding items on the menu to the tables orders, which is also an attribute to the table object ···

def prepareBill(): #used for preparing the bill to print and creating a variable ready to send to a file later on ···

def completeSale(): #used to save printed bill to a text file ···

def cashUp(): #used to show money made in a day, option to clear value ···

loginList = getLogins() #using function to read logins text and save to list
stockList = getStock() #using function to read stock text and save to list

profits = 0 #setting initial value for profits to start at for a day
while True: #main loop for continuous selection until exit is chosen
    #login page that kicks off all other funtions
    print("_____")
    print("Welcome to Highlands Cafe Service!\n" )
    userInput = int(input("1.Login\n2.Exit\nPlease enter your choice: "))
    if (userInput == 1):

        userLogin()
    else:
        print("\nThank you for using Highlands Cafe Service!")
        break
    print()
```

```
class tableDesign:
    def __init__(self, tableName = "DefaultName", tableAssignment = "None", tableOrders = {}, tableCustomers = 0, preparedBill = False, printedBill = ""):
        self.tableName = tableName
        self.tableAssignment = tableAssignment
        self.tableCustomers = tableCustomers
        self.tableOrders = tableOrders
        self.preparedBill = preparedBill
        self.printedBill = printedBill
tableDesign()
```

Into Each Function:

```
def userLogin(): #prompting the user for login details
    global username
    username = input("\nPlease enter username: ")
    password = input("Please enter password: ")
    print()
    for names in loginList:
        if names == username: #login validation
            if password == loginList[names]:
                print(f"\nWelcome {username}!")
                landingPage() #sending to the main landing page after login
                break
            else:
                print("Invalid Password")
                break
    else:
        print("Invalid Username")
```

```
def getLogins(): #function used for reading login text file and giving an output in list format
    lineList = {}
    with open("Resources\Login.txt") as f:
        for line in f:
            (key, val) = line.split(",")
            lineList[key] = val.strip("\n")
    return lineList

def getStock(): #function used for reading stock text file and giving an output in list format
    lineList = []
    with open("Resources\Stock.txt") as f:
        for line in f:
            (key, val) = line.split(",")
            lineList.append([key, val.strip("\n")])
    return lineList
```

```
def landingPage(): #Main landing page that shows all menu options and sends the user to correct function according to selection made
    userInput = int(input("\nWhat would you like to do? \n\
                          \n1. Assign Table \
                          \n2. Change customers\
                          \n3. Add to Order\
                          \n4. Prepare Bill\
                          \n5. Complete Sale\
                          \n6. Cashup \
                          \n0. Log out \
                          \nPlease enter your choice: "))
    print()
    if userInput == 1:
        availableTables()
    elif userInput == 2:
        changeCustomers()
    elif userInput == 3:
        addToOrder()
    elif userInput == 4:
        prepareBill()
    elif userInput == 5:
        completeSale()
    elif userInput == 6:
        cashUp()
    elif userInput == 0:
        print("\nLogging Out!!!")
        pass
    else:
        print("Invalid input!")
        landingPage()
```

```python
def availableTables(): #Used for the first option, checks tables that arent occupied and lets waiter choose their table.
    print("These are the available tables: ")
    for i in tableList:
        if i.tableAssignment == "None":
            print(f"{i.tableName.strip('Table ')}. {i.tableName}")
    userInput = int(input("Please enter your choice, or 0 to cancel: "))
    if(userInput != 0):
        tableList[userInput - 1].tableAssignment = username #assigning the table through attributes in the table object

        print(f"\n{tableList[userInput - 1].tableName} successfully assigned to {username}")

        addCustomers = input("Do you want to add customers to the table? (y/n): ")
        if addCustomers == "y":
            changeCustomers() #sends the user straight into the change customers function
        else:
            landingPage()
    else:
        print("\nCancelling and Returning!!")
        landingPage()
```

```python
def changeCustomers(): #used for the second option which is changing the amount of customers at a table
    print(f"\nThese are the tables available to {username}")
    for i in tableList: #This loops shows the tables that the waiter is working on
        if i.tableAssignment == username:
            print(f"{i.tableName.strip('Table ')}. {i.tableName}")


    userInput = int(input("Please select the table you would like to add customers to, or 0 to exit: "))
    while True:
        if tableList[userInput - 1].tableAssignment == username :
            print(f"\nThere are currently {tableList[userInput - 1].tableCustomers} customers assigned to {tableList[userInput-1].tableName}")
            customerNumber = int(input("What is the amount of customers: "))
            tableList[userInput-1].tableCustomers = customerNumber

            print(f"\n{customerNumber} Customers successfully added to {tableList[userInput - 1].tableName}")
            break
        elif (userInput == 0):
            print("\nCancelling and Returning!")
            break
        else:
            print("Invalid selection")
            userInput = int(input("Please select the table you would like to add customers to, 0 to exit: "))

    landingPage()
```

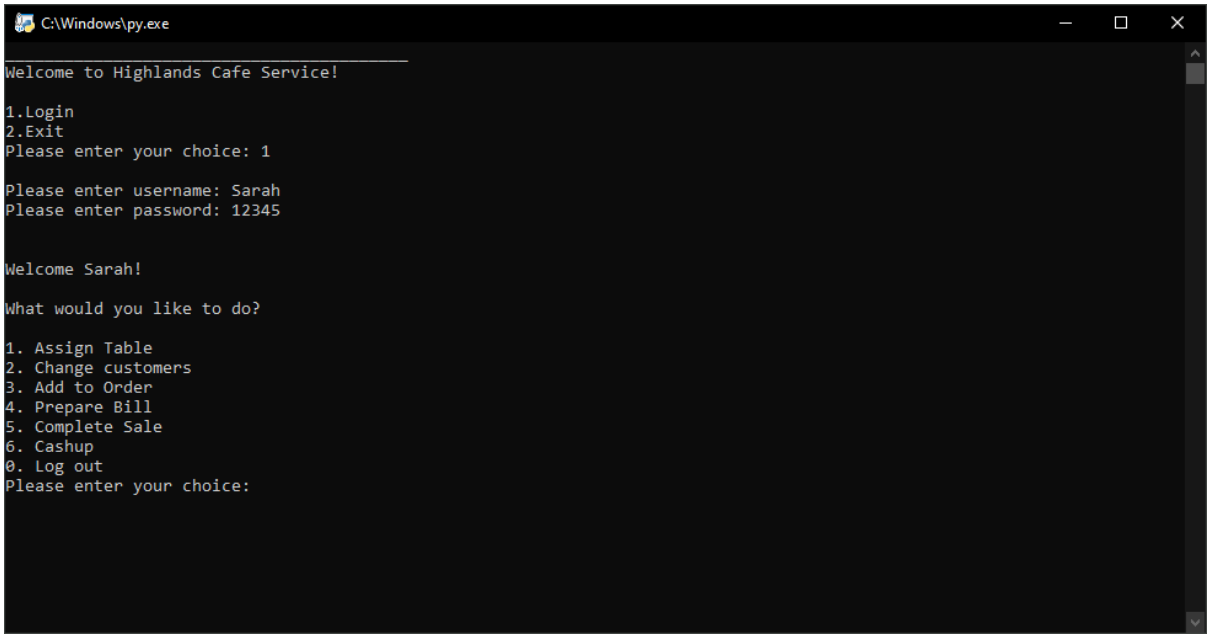
```python
def addToOrder(): #used for adding items on the menu to the tables orders, which is also an attribute to the table object
    print(f"\nThese are the tables available to {username}")
    for i in tableList:
        if i.tableAssignment == username:
            print(f"{i.tableName.strip('Table ')}. {i.tableName}")

    userInput = int(input("Please select the table you would like to add an order to, or 0 to exit: "))

    while True:
        if tableList[userInput - 1].tableAssignment == username :
            print("\nSelect an item: ")
            counter = 0
            for i in stockList:
                print(f"{counter + 1}. {i[0]} at R{i[1]}")
                counter += 1
            orderSelect = int(input("What item would you like to order: "))
            quantity = int(input(f"How many {stockList[orderSelect - 1][0]}/s would you like to add to the order: "))
            tableList[userInput - 1].tableOrders[stockList[orderSelect - 1][0]] = quantity
            #saved in the format of a diction(itemName: quantity)

            print(f"\n{quantity} {stockList[orderSelect - 1][0]}/s successfully added to {tableList[userInput - 1].tableName}")

            another = input("Would you like to add another order? (y/n): ")
            if another == "n":
                print()
                break

        elif (userInput == 0):
            print("\nCancelling and Returning!")
            break
        else:
            print("Invalid selection")
            userInput = int(input("Please select the table you would like to add an order to, 0 to exit: "))
    landingPage()
```

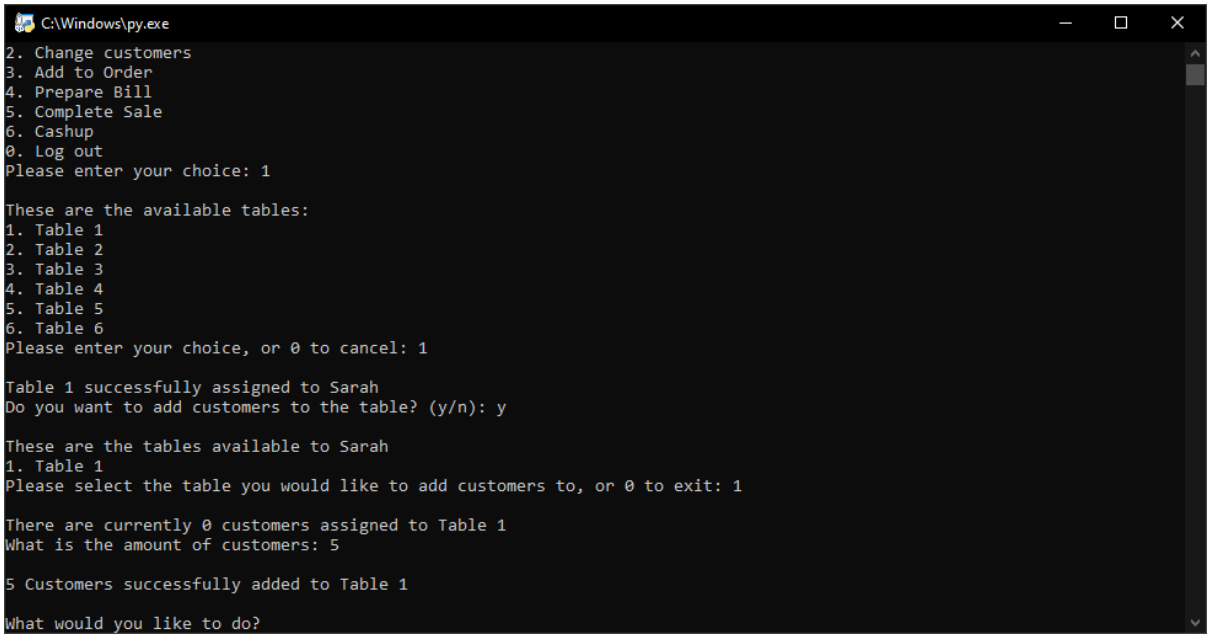
```
def prepareBill(): #used for preparing the bill to print and creating a variable ready to send to a file later on
    global profits
    print(f"\nThese are the tables available to {username}")
    for i in tableList:
        if i.tableAssignment == username:
            print(f"{i.tableName.strip('Table ')}. {i.tableName}")

    userInput = int(input("Please select the table you would like to prepare bill for, or 0 to exit: "))

    convertedTableList = list((tableList[userInput - 1].tableOrders).items())
    print (convertedTableList)
    while True:
        if tableList[userInput - 1].tableAssignment == username :
            tableList[userInput - 1].printedBill = f"- - - - - - - - - - - - - - - - - - - - - - - \
                \nThe bill for {tableList[userInput - 1].tableName}\n\
                \n\tItem\tQuantity\tPrice"
            counter = 0
            total = 0
            for i in convertedTableList:
                for x in stockList:
                    if (i[0]) == x[0]:
                        tableList[userInput - 1].printedBill += (f"\n\t{i[0]}\t{i[1]}\tR\t{(int(x[1]) * i[1])}")
                        total += (int(x[1]) * i[1])
                counter+=1
            tableList[userInput - 1].printedBill += f"\nThe total of you bill was R{total}\
                \nRecommended Tip is R{round(total*0.1, 2)}\
                \nGrand Total would be R{total + total*0.1}"
            if (tableList[userInput - 1].tableCustomers > 1):
                tableList[userInput - 1].printedBill += f"\nYour bill split evenly amongst {tableList[userInput - 1].tableCustomers} would be R{(total + total*0.1)/tableList[userInput - 1].tableCustomers} each"
            tableList[userInput - 1].printedBill += f"\nYou were helped by {username}\
                \n- - - - - - - - - - - - - - - - - - - - -"
            tableList[userInput - 1].preparedBill = True

            print(tableList[userInput - 1].printedBill)

            profits += total
            break

        elif (userInput == 0):
            print("\nCancelling and Returning!")
            break
        else:
            print("Invalid selection")
            userInput = int(input("Please select the table you would like to prepare bill for, 0 to exit: "))
    landingPage()
```

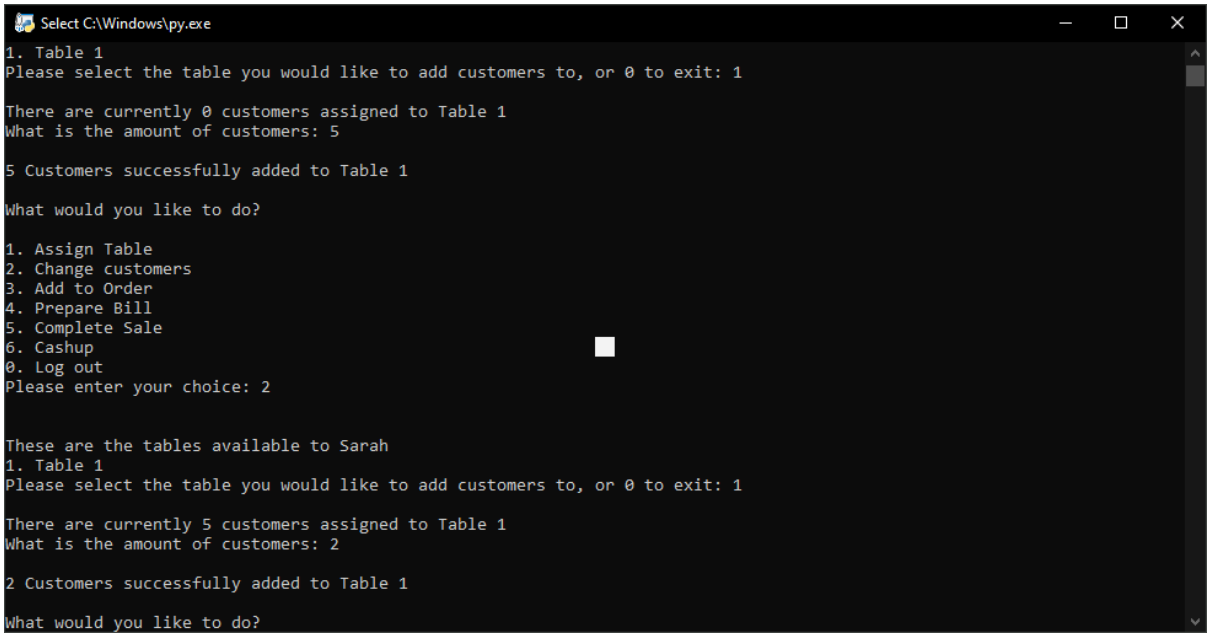
```
def completeSale(): #used to save printed bill to a text file
    print(f"\nThese are the tables available to {username}")
    for i in tableList:
        if i.tableAssignment == username:
            print(f"{i.tableName.strip('Table ')}. {i.tableName}")

    userInput = int(input("Please select the table you would like to complete sale for, or 0 to exit: "))

    while True:
        if tableList[userInput - 1].tableAssignment == username :
            if tableList[userInput - 1].preparedBill == True:
                fileName = input("Please enter a file name: ")
                f = open(f"CompletedOrders\{fileName}.txt", "x")
                f.write(tableList[userInput - 1].printedBill)

                print("\nSale was successfully completed")
                tableList[userInput - 1] = tableDesign(f"Table {userInput}")
                break
            else:
                print("\nBill is NOT prepared!")
                break

        elif (userInput == 0):
            print("\nCancelling and Returning!")
            break
        else:
            print("Invalid selection")
            userInput = int(input("Please select the table you would like to complete sale for, 0 to exit: "))
    landingPage()
```

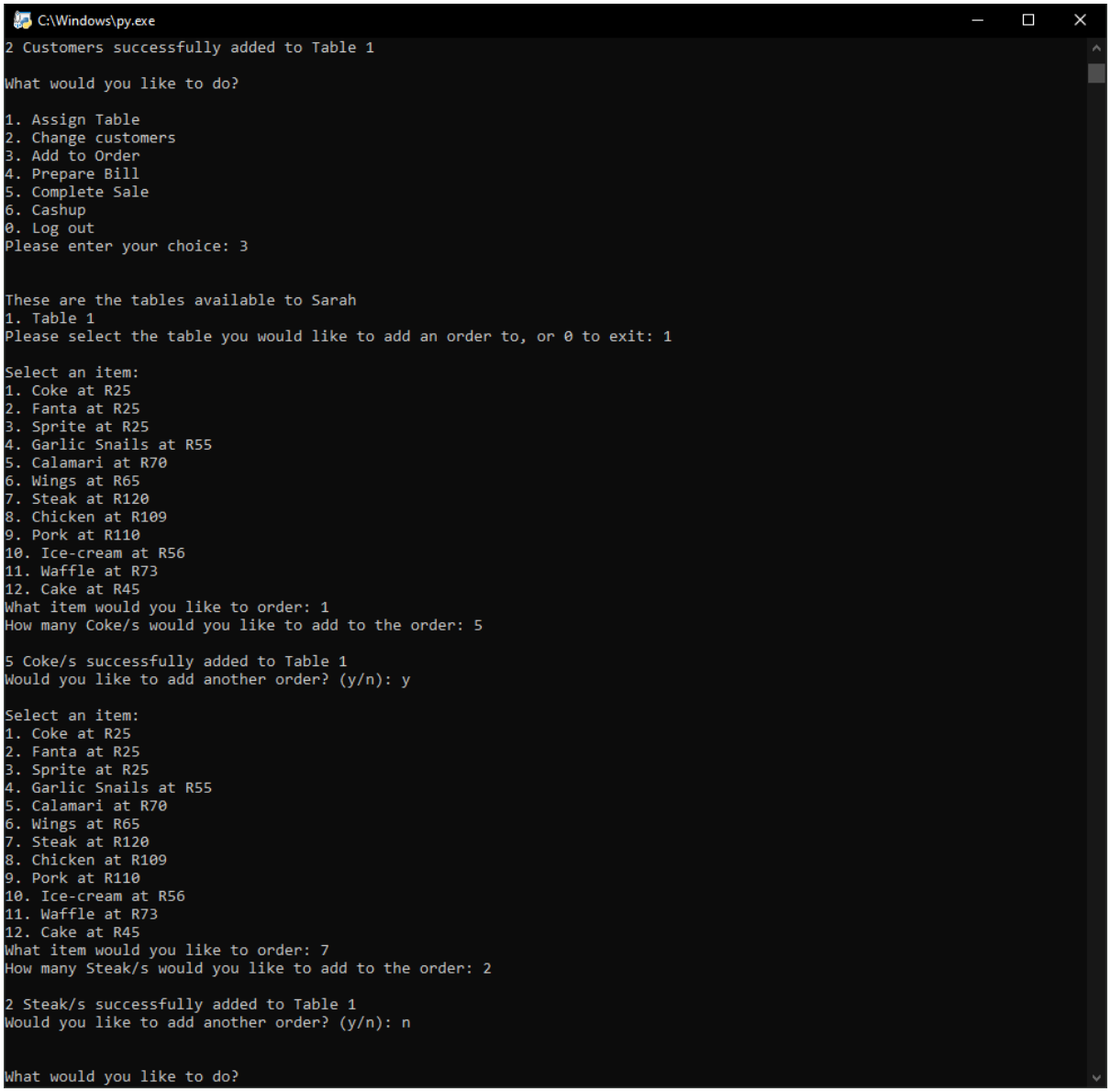
```
def cashUp(): #used to show money made in a day, option to clear value
    global profits
    print(f"\nToday we made R{profits}")
    clearProfits = input("Do you wish to clear todays profit?(y/n): ")
    if clearProfits == "y":
        profits = 0
    landingPage()
```

Outputs(Code test run):

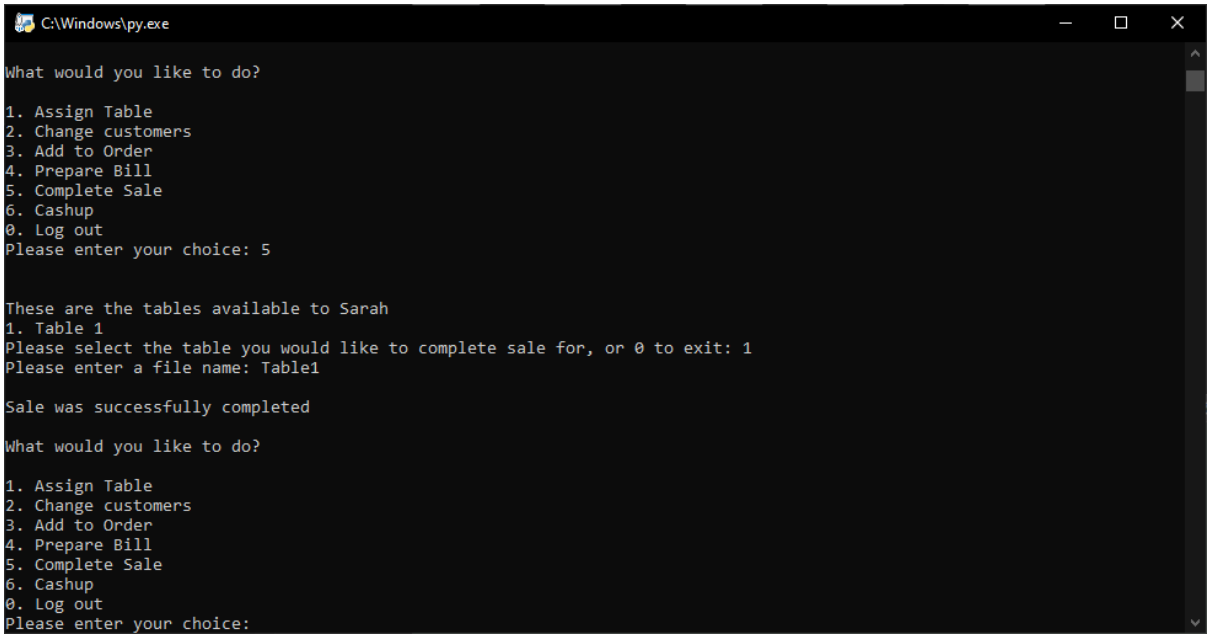


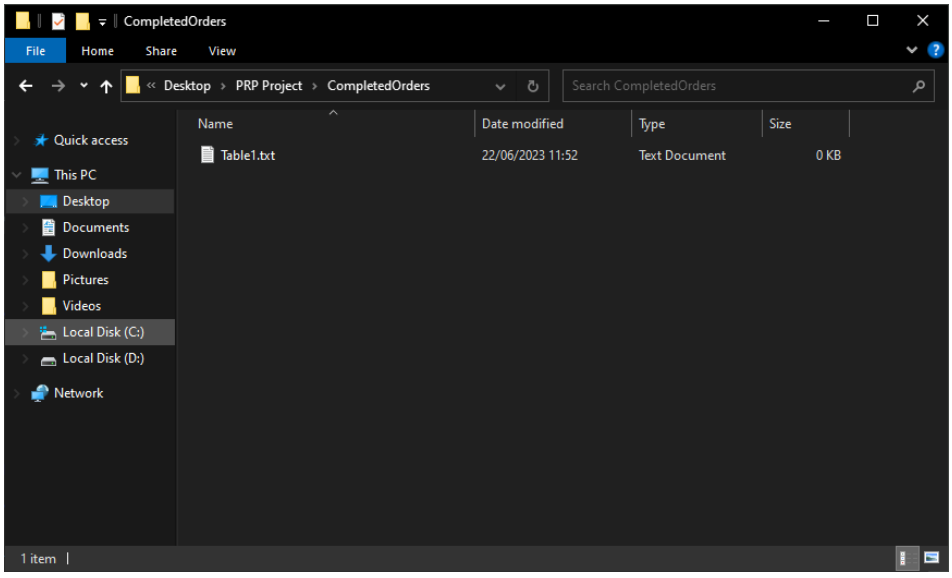


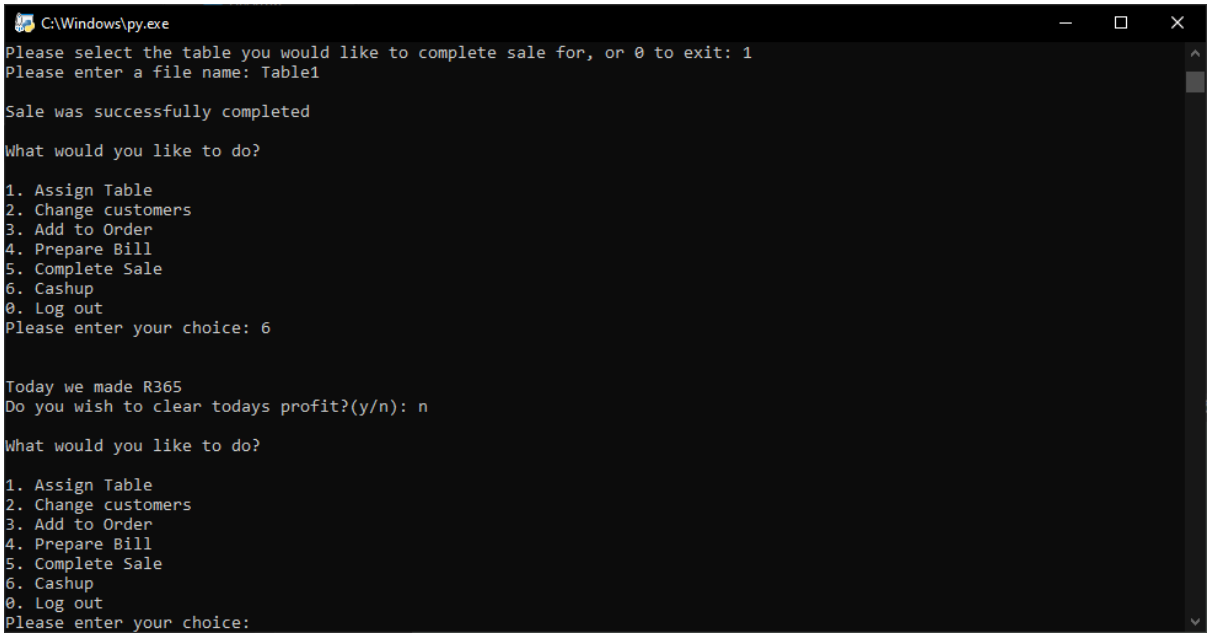

Select C:\Windows\py.exe
1. Table 1
Please select the table you would like to add customers to, or 0 to exit: 1
There are currently 0 customers assigned to Table 1
What is the amount of customers: 5
5 Customers successfully added to Table 1
What would you like to do?
1. Assign Table
2. Change customers
3. Add to Order
4. Prepare Bill
5. Complete Sale
6. Cashup
0. Log out
Please enter your choice: 2
These are the tables available to Sarah
1. Table 1
Please select the table you would like to add customers to, or 0 to exit: 1
There are currently 5 customers assigned to Table 1
What is the amount of customers: 2
2 Customers successfully added to Table 1
What would you like to do?


C:\Windows\py.exe
2 Customers successfully added to Table 1
What would you like to do?
1. Assign Table
2. Change customers
3. Add to Order
4. Prepare Bill
5. Complete Sale
6. Cashup
0. Log out
Please enter your choice: 3
These are the tables available to Sarah
1. Table 1
Please select the table you would like to add an order to, or 0 to exit: 1
Select an item:
1. Coke at R25
2. Fanta at R25
3. Sprite at R25
4. Garlic Snails at R55
5. Calamari at R70
6. Wings at R65
7. Steak at R120
8. Chicken at R109
9. Pork at R110
10. Ice-cream at R56
11. Waffle at R73
12. Cake at R45
What item would you like to order: 1
How many Coke/s would you like to add to the order: 5
5 Coke/s successfully added to Table 1
Would you like to add another order? (y/n): y
Select an item:
1. Coke at R25
2. Fanta at R25
3. Sprite at R25
4. Garlic Snails at R55
5. Calamari at R70
6. Wings at R65
7. Steak at R120
8. Chicken at R109
9. Pork at R110
10. Ice-cream at R56
11. Waffle at R73
12. Cake at R45
What item would you like to order: 7
How many Steak/s would you like to add to the order: 2
2 Steak/s successfully added to Table 1
Would you like to add another order? (y/n): n
What would you like to do?


Select C:\Windows\py.exe
What would you like to do?
1. Assign Table
2. Change customers
3. Add to Order
4. Prepare Bill
5. Complete Sale
6. Cashup
0. Log out
Please enter your choice: 4
These are the tables available to Sarah
1. Table 1
Please select the table you would like to prepare bill for, or 0 to exit: 1
[('Coke', 5), ('Steak', 2)]
- -
The bill for Table 1
Item Quantity Price
Coke 5 R 125
Steak 2 R 240
The total of you bill was R365
Recommended Tip is R36.5
Grand Total would be R401.5
Your bill split evenly amongst 2 would be R200.75 each
You were helped by Sarah


C:\Windows\py.exe
What would you like to do?
1. Assign Table
2. Change customers
3. Add to Order
4. Prepare Bill
5. Complete Sale
6. Cashup
0. Log out
Please enter your choice: 5
These are the tables available to Sarah
1. Table 1
Please select the table you would like to complete sale for, or 0 to exit: 1
Please enter a file name: Table1
Sale was successfully completed
What would you like to do?
1. Assign Table
2. Change customers
3. Add to Order
4. Prepare Bill
5. Complete Sale
6. Cashup
0. Log out
Please enter your choice:


CompletedOrders
File
Home
Share
View
Desktop
PRP Project
CompletedOrders
Search CompletedOrders
Quick access
This PC
Desktop
Documents
Downloads
Pictures
Videos
Local Disk (C:)
Local Disk (D:)
Network
Name
Date modified
Type
Size
Table1.txt
22/06/2023 11:52
Text Document
0 KB
1 item



Declaration of Authenticity:

Completed Declaration of Authenticity

I, Luca du Toit, solemnly declare that the work presented in this Summative is entirely my own. I have not plagiarized or copied the work of others without proper acknowledgment. I affirm that the content, ideas, and arguments presented herein are the result of my independent effort and intellectual contributions.

I understand the significance of academic integrity and the detrimental consequences of engaging in plagiarism or other forms of dishonesty. Therefore, I assure you the following:

1. All sources used in this work, including but not limited to books, articles, websites, and personal communications, have been appropriately cited and referenced according to the specified guidelines or referencing style.
2. Any direct quotations or paraphrased information from external sources have been identified by using quotation marks or proper citation methods.
3. I have not received any unauthorized assistance or collaboration from others in completing this work, except for instances explicitly permitted by the instructor or clearly stated in the assignment guidelines.
4. The ideas, arguments, and interpretations expressed in this work are my own and have not been submitted for assessment in any other academic setting unless explicitly mentioned and properly acknowledged.
5. I acknowledge that failure to adhere to these principles of academic honesty and integrity may result in severe penalties, including but not limited to the rejection of this work, loss of marks, academic probation, or disciplinary action as deemed appropriate by the educational institution.

Signature:  Date: 22/06/2023