INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT108-3-1-PYP

PYTHON PROGRAMMING

APU1F2206-SE

HAND OUT DATE : 18th July 2022

HAND IN DATE : 25th September 2022

WEIGHTAGE : 100%

NAME : KEVIN TEH TING QIAN

TP_NUMBER : TP068904

INTAKE : APU1F2206SE

Table of Content

1.0 Introduction & Assumptions

This project focuses on an Online Groceries Management System (OGMS) that is utilized by FRESHCO Sdn Bhd. As more people with hectic schedules prefer to shop online, FRESHCO Sdn Bhd hired me to design this method. In this system, user available to make grocery order and payment, and for the non-registered user, they can have a sign-up section to become a member of this system user, for the system admin, they are available to make changes aggrading to the grocery details.

In this system, the user will initially be prompted for identify, including whether they are an admin, a registered customer, or a new customer. In the admin section, admin can log in to the system, after that, admin available to pick numbers for next move, such as view grocery list, upload/modify grocery details, add and delete grocery list information, available search for a specific item in the list, and search for a particular or general customer order. Admin don't want to do the next move; admin can choose to exit for quite this system.

Moving on, users can log in to access the system and examine details of their items in the part designated for registered customers. In addition, the user may place an order and make a payment in this part. Additionally, the user has access to view and review their own personal information.

Last but not least, the new customer section is identical to the other two parts in that the user can read grocery details and complete the registration process to utilize this system. In contrast, if the user does not belong to one of the three identities at the identity section part, the system will send an error and allow the user to try again.

2.0 Design of Program

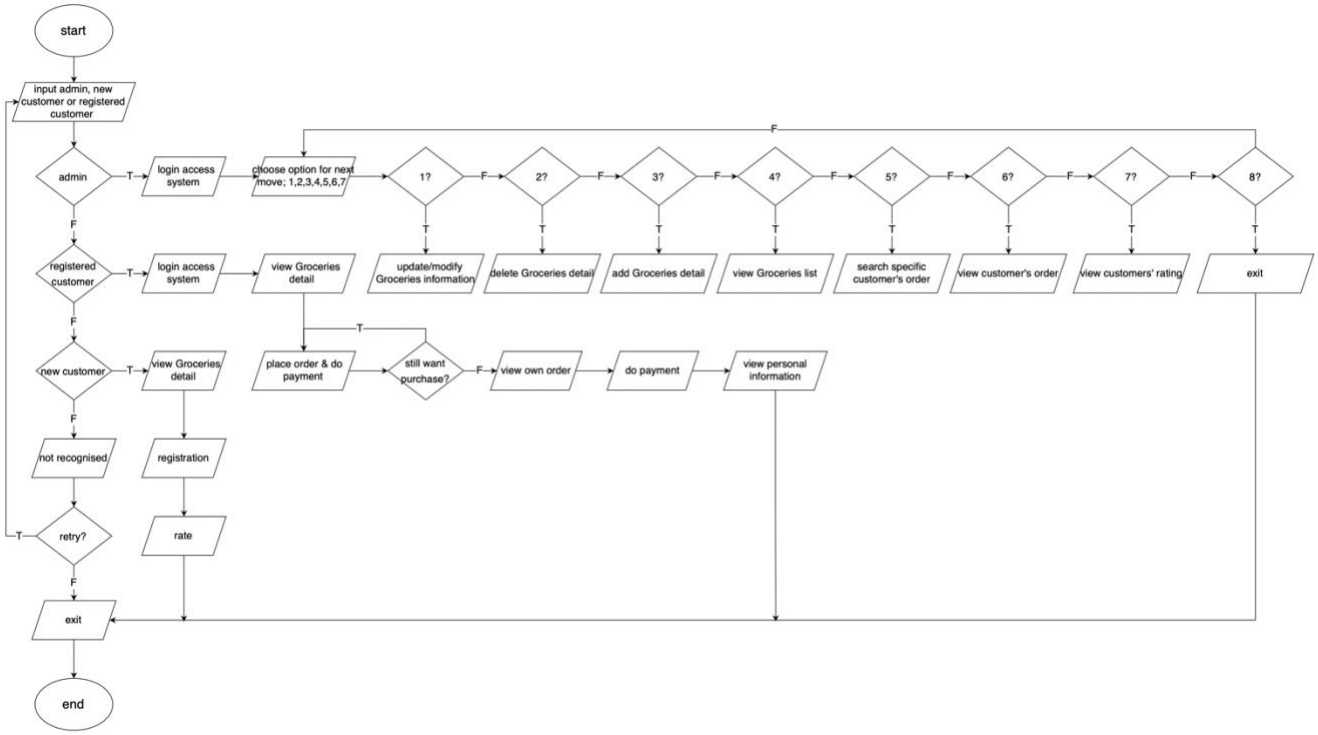

Figure 2.0.1: Flowchart of the System

```
START
User input identity(admin, registered customer or new customer)
While True
    If user = admin
        User insert username and password
        Print groceries list
        User input number 1,2,3,4,5,6,7
        If number = 1
            Prompt user update/modify groceries information
        Else if number = 2
            Prompt user delete groceries detail
        Else if number = 3
            Prompt user add groceries detail
        Else if number = 4
            Print groceries list
        Else if number = 5
            Prompt user search specific customer's order
        Else if number = 6
            Upload groceries detail
        Else if number = 7
            print customer rating list
        Else if number = 8
            Print Exit
        Else
            print ERROR
            Prompt user input number
        End if
    Else if user = registered customer
        Prompt user insert username and password
        Print groceries detail
        Prompt user place order
        If user still want purchase
            Prompt user place order
        Else
            Print user order
        End if
        Prompt user do payment
        Print user personal information
        Prompt user to give a rating
        Print Exit
    Else if user = new customer
        Print groceries detail
        User input for registration
    Else
        Print Error, retry
    Endif
End while
END
```

Figure 2.0.2: Pseudocode

The Online Groceries Management System (OGMS) is depicted in Figure 1 as it functions. For Figure 2, its pseudocode for the system flow. There are three sections in this flow chart: admin, registered customers, and new customers. Users can initially select the identity they want to use to log in and access the system.

Users who select the identity of admin must check in and demonstrate their admin status to access the website. The user can choose a number from 1 to 8 to carry out the following action after

completing the login process, which includes the following steps: 1. update/modify groceries information; 2. delete groceries detail; 3. add groceries detail; 4. view groceries list; 5. search for a specific customer's order; 6. view customer's order; 7. view customer rating and 8. exit. If the user chooses option 1, the name and price of the products on the grocery list will be changed. The user can modify the grocery list by deleting or adding products using choices 2 and 3. To determine whether there is a problem, the user can read the most recent grocery list in option 4. Options 5 and 6 both allow users to view the customer's order information, but option 5 only allows users to view the orders of all customers and cannot view the order information of a specific user. Option 6, on the other hand, allows users to search for a specific user's information, greatly reducing the amount of time required for the user to search through all orders one by one. For option 7, it allows user to review customers' rating. Option 8 allows the user to log out of the system, as a last step.

Compared to the admin interface, it will be considerably simpler in the registered customer. Like the previous interface, consumers who select this one must also log in to verify their identity before they can access their shopping list and make a purchasing decision. The consumer can repeat the shopping operation if they decide to repurchase the item, if not, the order for the previously purchased groceries will be printed out and charged. After taking this action, the user can examine his personal data, including name, address, birth year, etc. Before leaving, users must rate the system before they can proceed to the exit.

The user can only read the shopping list and complete the registration process in the last section, which is for new customers. On this page, they must enter their username, password, name, and address, among other details. After, it will enter the exit section after finishing the tasks.

3.0 Program Source Code and Explanation

Many various routines are utilised in this report to improve the system's performance, including the for loop, while loop, function, file, among others, to make the system function more smoothly.

3.1 WHILE LOOP

While loops are loops that run continually if a condition is met until the condition is not met or a break is used. This system employs a variety of while loop types.

```
if iden == a:
    while True:
        admin = host()
        print("\nUsername")
        username = input("")
        print("Password")
        password = input("")

        if username not in admin:
            print(f"\nUser not found {username}\n")
        elif admin[username] != password:
            print(f"\nIncorrect password for user {username}\n")
        else:
            print("\nWelcome\n")
            break
```

Figure 3.1.1: While Loop

If the admin's username and password are correct, the while loop will automatically loop in the True environment, and only then will it enter the Break in If else. Break can halt the hovering circle from continuing.

3.2 FOR LOOP

The for loop has an integrable object reading capability. The target that the for-loop will use to read data is behind the in. An integrable object, this target reads each element one at a time.

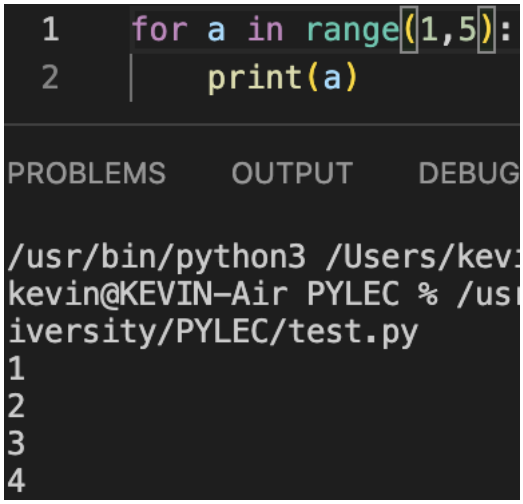

Figure 3.2.1: For Loop

This image illustrates how a will print the items in range (1,5) one by one without printing 5, which means that the actual print result will only be between 1 and 4.

3.3 IF ELSE

Using the execution outcome (True or False) of one or more statements, a Python conditional statement chooses whether to run the block of code. The following statement is carried out when the "judgement condition" is met (i.e., when it is non-zero). The execution content may consist of numerous lines, each of which is differentiated by indentation to denote a certain range. The related statement may be carried out when the content must be carried out even though the condition has not been established in the case of the optional statement "else."

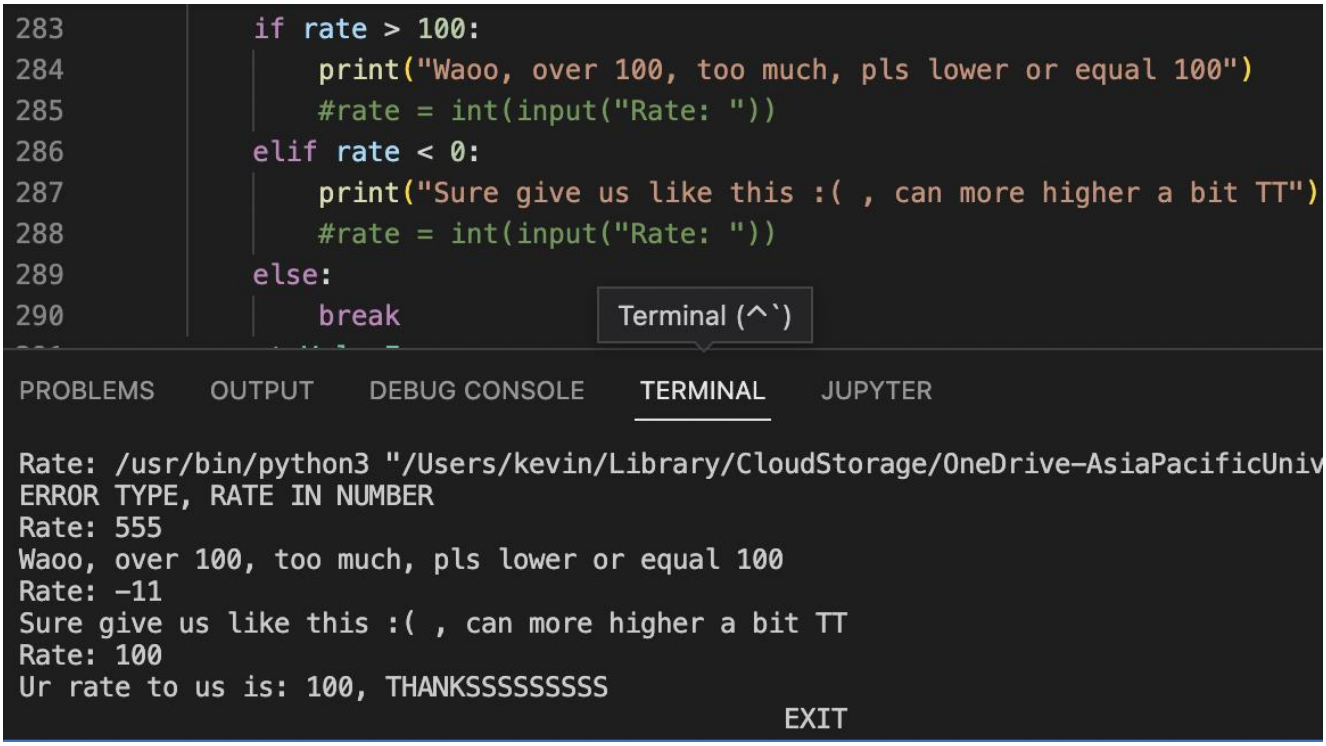

Figure 3.3.1: If Else - Conditional Statement

In this figure, if the input number is greater than 100, "Waoo, over 100, too much, pls lower or equal 100" will appear, and if the input number is less than 0, "Sure give us like this :(, can more higher a" will appear bit TT", this is elif(else if) - is another condition that is executed when the main condition is not established. If the input number is between 0 and 100, "Ur rate to us is: 100, THANKSSSSSSSSS" will appear, this is else

3.4 TRY/EXCEPT

Try/except statements can be used to catch exceptions. The try statement block uses the try/except statement to identify errors so that the except statement can store and handle exception data.

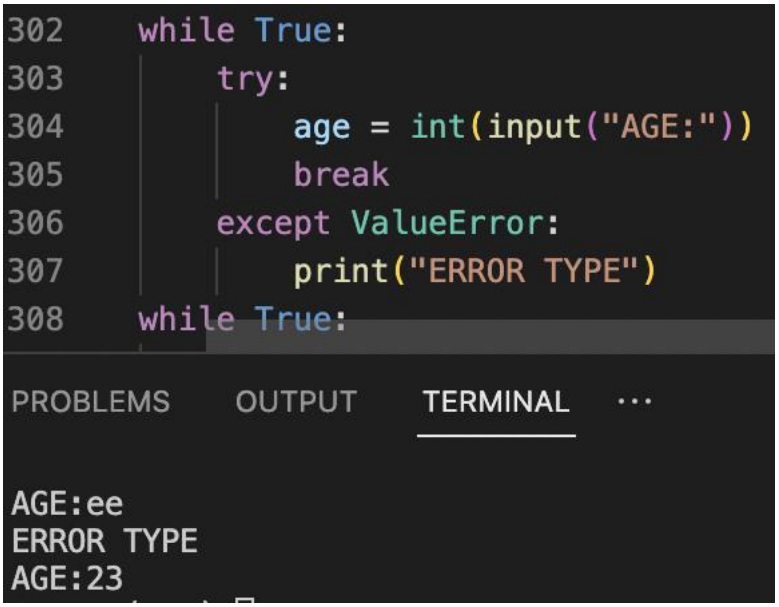

Figure 3.4.1: Try/Except

Here, the program will continue to loop until it is correct thanks to the use of the while loop. This code indicates that the user provides a numeric age; however, if the input in try is not a number, the program will go to the except block and display out the appropriate error type, as seen in the figure. Furthermore, if the first input in try is a number, the computer will bypass the except and move on to the following code.

3.5 FUNCTION

An organized, reusable section of code called a function implements a single or group of related functions. Functions can increase the modularity and reuse of applications.

```
def viewcusorder():
    with open("orderlist.txt", "r") as file:
        print(file.read())
```

Figure 3.5.1: Function

As seen in Figure 3.6.1, a function code block starts with the def keyword and is followed by the name of the function and parenthesis ().

In the above figure, it opens a file and reads it, and is defined as viewcusorder(). The contents of the file will appear whenever viewcusorder() occurs anywhere next. As shown in Figure 3.6.2.

```
37      viewcusorder()
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   JUPYTER        Python

USER: sd {'ORANGE': {'QUANTITY': 2, 'SUBTOTAL': 24.0}} TOTAL: 24.0
USER: sd {'ORANGE': {'QUANTITY': 3, 'SUBTOTAL': 36.0}} TOTAL: 36.0
USER: sd {'ORANGE': {'QUANTITY': 3, 'SUBTOTAL': 36.0}} TOTAL: 36.0
USER: ww {} TOTAL: 0
USER: ww {} TOTAL: 0
USER: ww {} TOTAL: 0
USER: ww {} TOTAL: 0
USER: ww {} TOTAL: 0
USER: sd {} TOTAL: 0
USER: sd {} TOTAL: 0
USER: sd {'APPLE': {'QUANTITY': 1, 'SUBTOTAL': 69.0}, 'WATERMELON': {'QUANTITY': 1, 'SUBTOTAL': 3
4.0}} TOTAL: 103.0
USER: sd {'APPLE': {'QUANTITY': 1, 'SUBTOTAL': 69.0}, 'GRAPE': {'QUANTITY': 1, 'SUBTOTAL': 34.0}}
 TOTAL: 103.0
```

Figure 3.5.2: Example of Function

3.6 FILE

Unquestionably, this system's most crucial data are kept in a variety of text files.

```
with open("orderlist.txt", "r") as file:
```

Figure 3.6.0.1: Code to Open File

This line of code is crucial to be able to open the file. After the file has been opened, this line of code can be automatically closed, preventing the possibility that the user will forget to close the file. The file name, such as text.txt, must be typed in the first place in the () field. The method for opening this file is after the file name. For instance, r means read, a means appends, and w means write. After that, it is futile to just open the file without guiding him through its execution. To run the file, it must be specified how.

```
file.read()
```

Figure 3.6.0.2: Read File

Its function in the figure above is to read the file's whole contents.

```
file.readlines()
```

Figure 3.6.0.3: Read File Content Line by Line

Its function is to read the file's contents line by line in the figure above.

```
file.write("\n" + newid + ", " + newrepass)
```

Figure 3.6.0.4: Write Data into File

The purpose of the object in the previous figure is to write data, and the input data can be written in ().

3.6.1 GROSLIST.TXT

groslist.txt

```
1   WATERMELON 34
2   GRAPE 34
3   APPLE 69
```

Figure 3.6.1.1: Content of Grocery List

This text file will house all the grocery-related data for this system, including name.

3.6.2 HOSTLIST.TXT

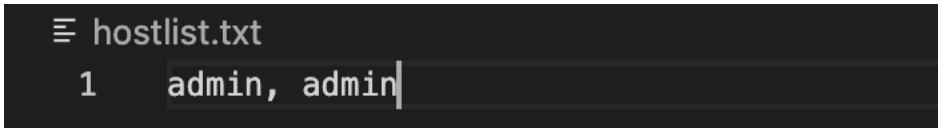

Figure 3.6.2.1: Admin List

The main purpose of this file is to store the username and password of the admin.

3.6.3 ORDERLIST.TXT

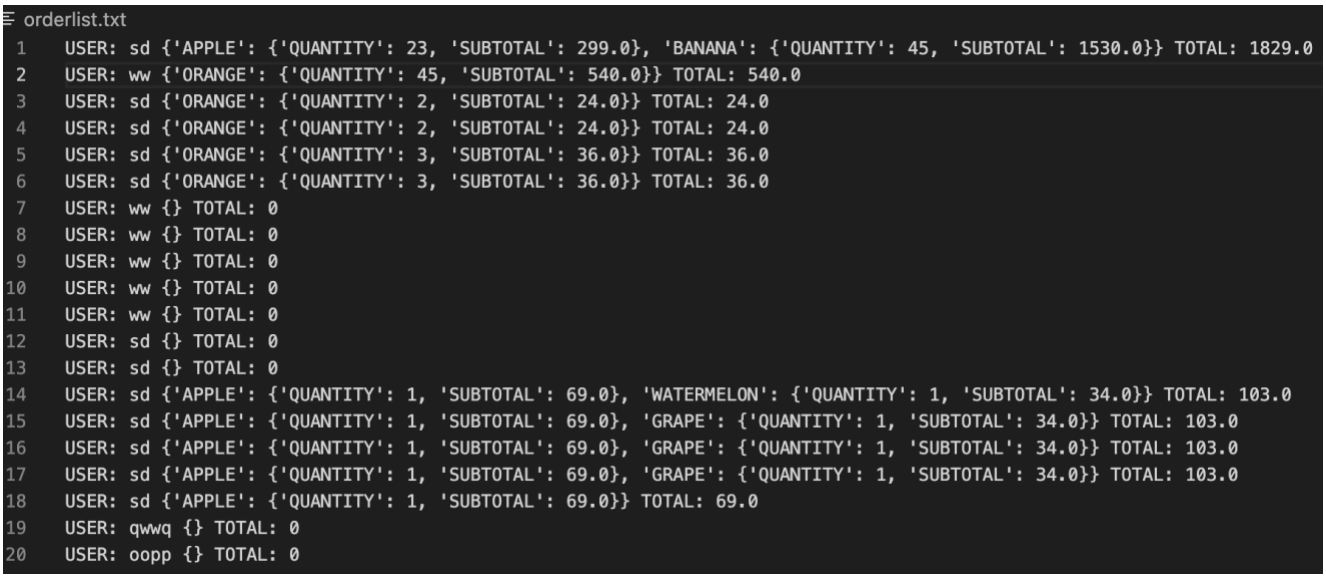

Figure 3.6.3.1: The Order List of Customer

The customer can access its current order list in this file, or the admin can examine the order list of every customer.

3.6.4 RATE.TXT

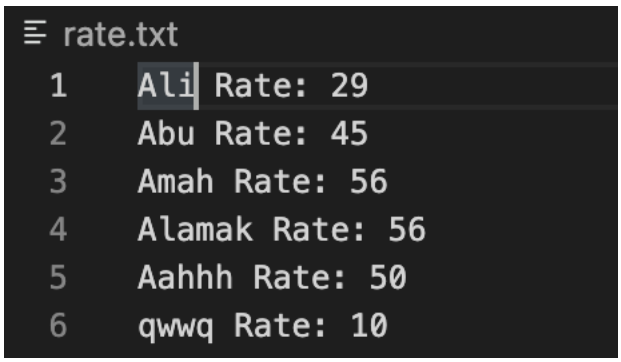

Figure 3.6.4.1: Customer Rating

The purpose of rate.txt is to gather user ratings on the website so that the administrator can review and improve it.

3.6.5 USER.TXT

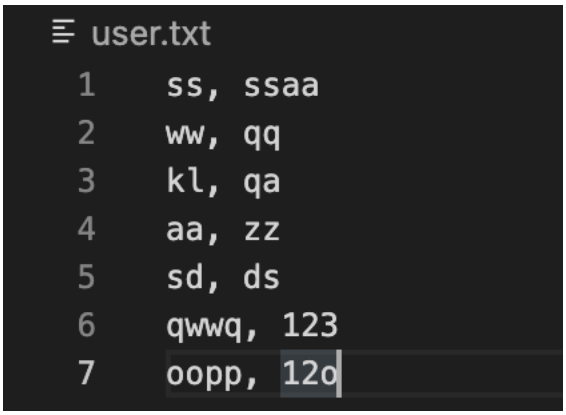

Figure 3.6.5.1: Registered Customers' Username and Password

This list is to store the username and password of the customer to avoid loss.

3.6.6 USERINFO.TXT

Figure 3.6.6.1: Registered Customers' Personal Information

The difference between this list and user.txt is that in addition to holding the customer's username and password in common, this list also stores the customer's personal data, including name, gender, and other details. The customer can also access this file to examine their own personal data.

4.0 Screenshots of Sample I/O and Explanation

4.1 WELCOME

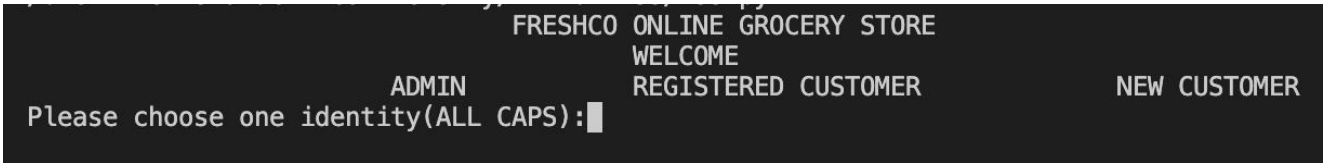

Figure 4.1.1: Welcome

The system's home page is this one. The user will be prompted to input an identification after this page, such as admin, registered customer, or new customer.

4.2 ADMIN

Going to the admin page will be the main topic of discussion in this section.

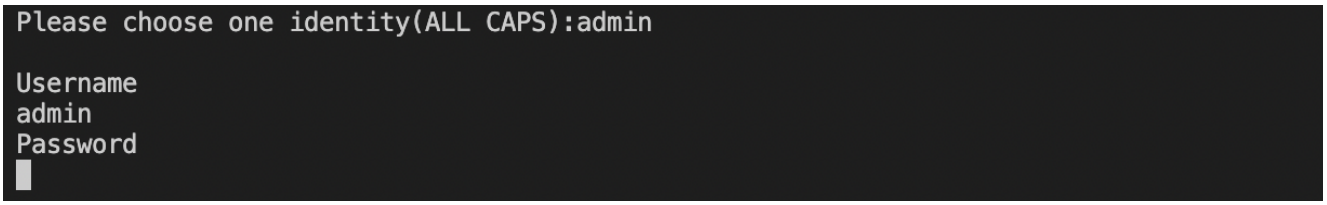

Figure 4.2.1: Admin Login

The user will be prompted to enter a username and password in Figure 4.2.1 to confirm the admin's identification. If the user enters wrong information, there will be 2 feedbacks. Its 2 are as follows:

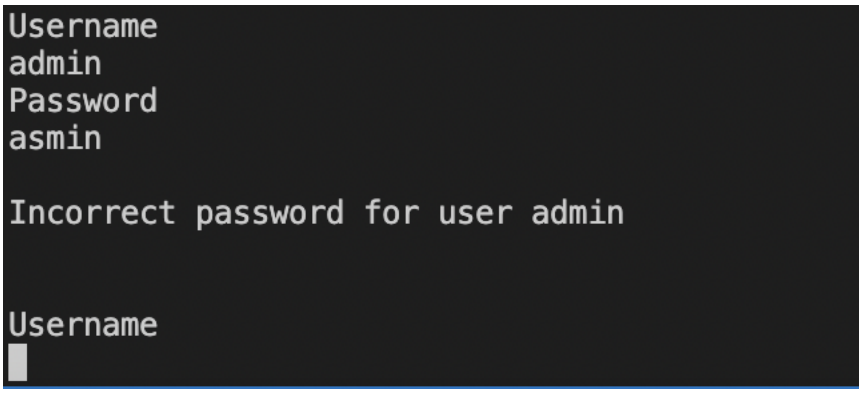

Figure 4.2.2: Login with Wrong Password

First, if the password entered by the user is incorrect, the system will require the user to re-enter it again.

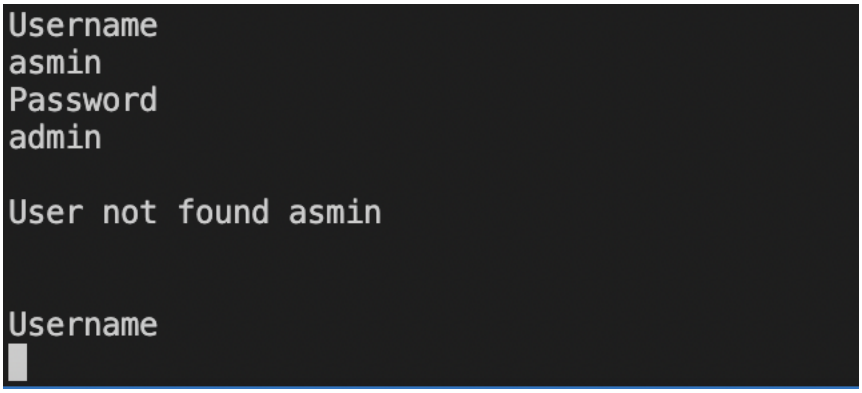

Figure 4.2.3: Login with Wrong Username

Or if the username entered by the user is wrong, the system will warn the user that the username does not exist, so that the user can enter one more time.

Figure 4.2.4: Successful Login

The system will display a successful login screen and the user's grocery list if the user's login username and password are accurate.

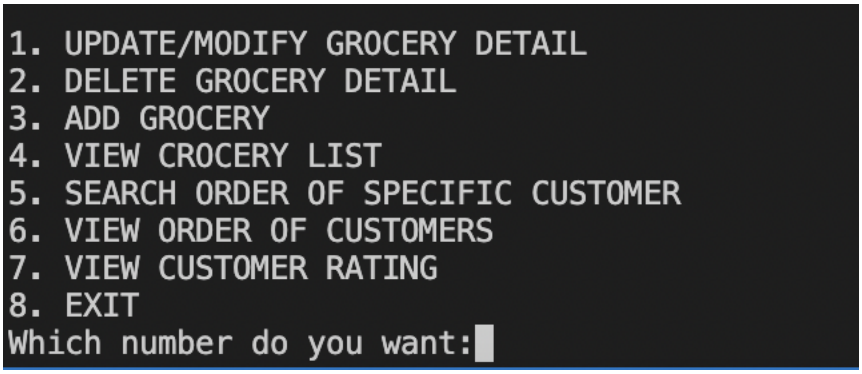

Figure 4.2.5: Choose for Next Move

After the user logs in and confirms the grocery list, he or she can choose the desired number in the above image to move on to the following step by clicking on it.

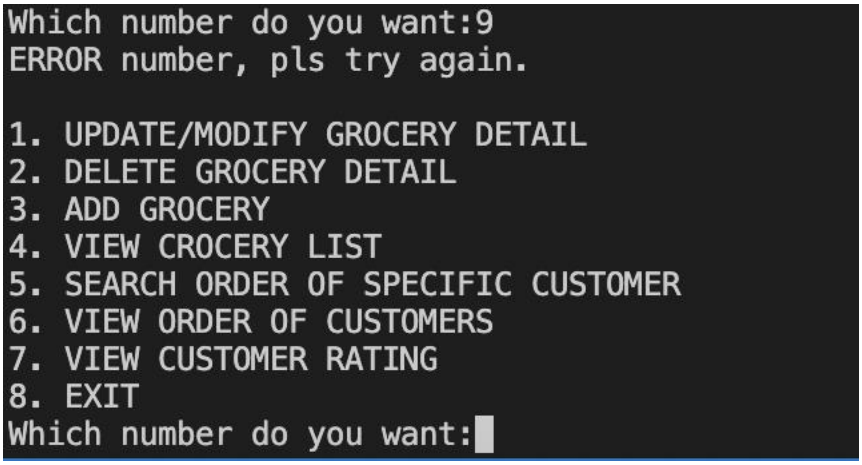

Figure 4.2.6: Wrong Insert

The user will be prompted to enter an error and be asked to enter another number if the number they entered is invalid.

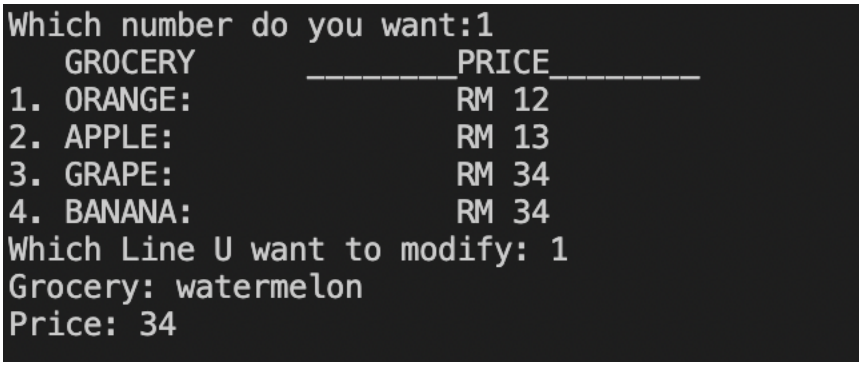

Figure 4.2.7: Update/Modify Grocery Detail

The user will be able to edit the contents of the grocery list, such as names and prices, if user enter No. 1.

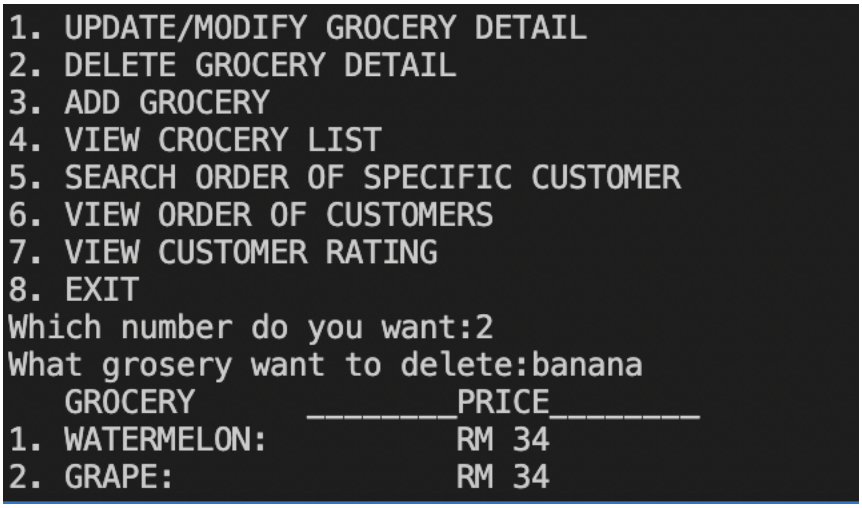

Figure 4.2.8: Delete Grocery

The user may remove any item from the grocery list for insert number 2. for instance, to remove banana.

```
1. UPDATE/MODIFY GROCERY DETAIL
2. DELETE GROCERY DETAIL
3. ADD GROCERY
4. VIEW CROCERY LIST
5. SEARCH ORDER OF SPECIFIC CUSTOMER
6. VIEW ORDER OF CUSTOMERS
7. VIEW CUSTOMER RATING
8. EXIT
Which number do you want:3
What grocery want to add:apple
Price:69
```

Figure 4.2.9: Add Grocery

Option 3 allows the user to add new product details.

```
1. UPDATE/MODIFY GROCERY DETAIL
2. DELETE GROCERY DETAIL
3. ADD GROCERY
4. VIEW CROCERY LIST
5. SEARCH ORDER OF SPECIFIC CUSTOMER
6. VIEW ORDER OF CUSTOMERS
7. VIEW CUSTOMER RATING
8. EXIT
Which number do you want:4
                                        GROCERY LIST:
   GROCERY      ________PRICE________
1. WATERMELON:          RM 34
2. GRAPE:               RM 34
3. APPLE:               RM 69
```

Figure 4.2.10: View Grocery List

Users can read the most recent grocery list if they choose option number 4.

```
Which number do you want:5
Insert the username that U want to view:sd


USER: sd {'APPLE': {'QUANTITY': 23, 'SUBTOTAL': 299.0}, 'BANANA': {'QUANTITY': 45, 'SUBTOTAL': 1530.0}} TOTAL: 1829.0

USER: sd {'ORANGE': {'QUANTITY': 3, 'SUBTOTAL': 36.0}} TOTAL: 36.0

USER: sd {'APPLE': {'QUANTITY': 1, 'SUBTOTAL': 69.0}, 'WATERMELON': {'QUANTITY': 1, 'SUBTOTAL': 34.0}} TOTAL: 103.0

USER: sd {'APPLE': {'QUANTITY': 1, 'SUBTOTAL': 69.0}} TOTAL: 69.0
```

Figure 4.2.11: Search Order of Specific Customer

Option 5 is without a doubt the best course of action if the user wants to locate the order list of a specific customer. You can only print the order list for that customer after inputting the name of the customer you're looking for.

```
1. UPDATE/MODIFY GROCERY DETAIL
2. DELETE GROCERY DETAIL
3. ADD GROCERY
4. VIEW CROCERY LIST
5. SEARCH ORDER OF SPECIFIC CUSTOMER
6. VIEW ORDER OF CUSTOMERS
7. VIEW CUSTOMER RATING
8. EXIT
Which number do you want:6
USER: sd {'APPLE': {'QUANTITY': 23, 'SUBTOTAL': 299.0}, 'BANANA': {'QUANTITY': 45, 'SUBTOTAL': 1530.0}} TOTAL: 1829.0
USER: ww {'ORANGE': {'QUANTITY': 45, 'SUBTOTAL': 540.0}} TOTAL: 540.0
USER: sd {'ORANGE': {'QUANTITY': 2, 'SUBTOTAL': 24.0}} TOTAL: 24.0
USER: sd {'ORANGE': {'QUANTITY': 2, 'SUBTOTAL': 24.0}} TOTAL: 24.0
USER: sd {'ORANGE': {'QUANTITY': 3, 'SUBTOTAL': 36.0}} TOTAL: 36.0
USER: sd {'ORANGE': {'QUANTITY': 3, 'SUBTOTAL': 36.0}} TOTAL: 36.0
```

Figure 4.2.12: View Customers' Order

Option 6 allows you to print the order list for all customers if you don't want to print the order list for just one customer. The customer's purchase history and total amount can be checked by the user simultaneously.

```
1. UPDATE/MODIFY GROCERY DETAIL
2. DELETE GROCERY DETAIL
3. ADD GROCERY
4. VIEW CROCERY LIST
5. SEARCH ORDER OF SPECIFIC CUSTOMER
6. VIEW ORDER OF CUSTOMERS
7. VIEW CUSTOMER RATING
8. EXIT
Which number do you want:7
sd Rate: 29
sd Rate: 45
sd Rate: 56
ww Rate: 56
```

Figure 4.2.13: View Customer Rating

The user can choose option 7 to find out how the client feels about the way the system works, whether they are happy or whether there is room for improvement.

Figure 4.2.14: Exit System

Finally, if the user selects option 8, the system will depart.

4.3 REGISTERED CUSTOMER

Moving on, this is the area for registered users, which is also the most crucial element of the entire system.

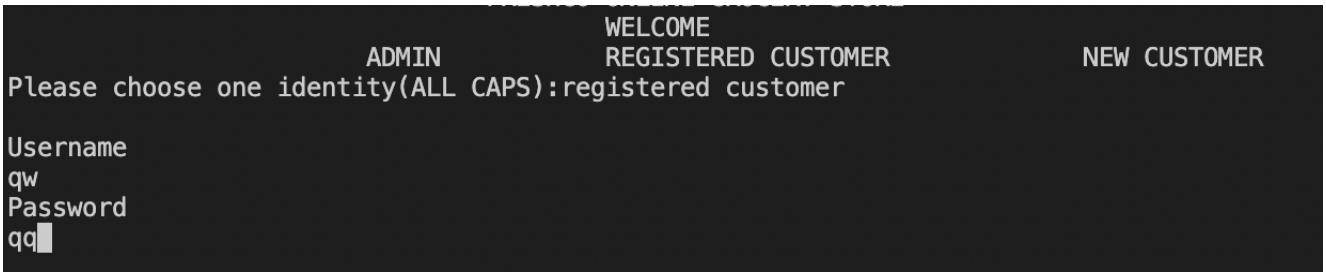

Figure 4.3.1: Registered Customer Login

The user will additionally need to complete a login action if they select the identity of a registered customer. If the user enters wrong information, there will be 2 feedbacks. Its 2 are as follows:

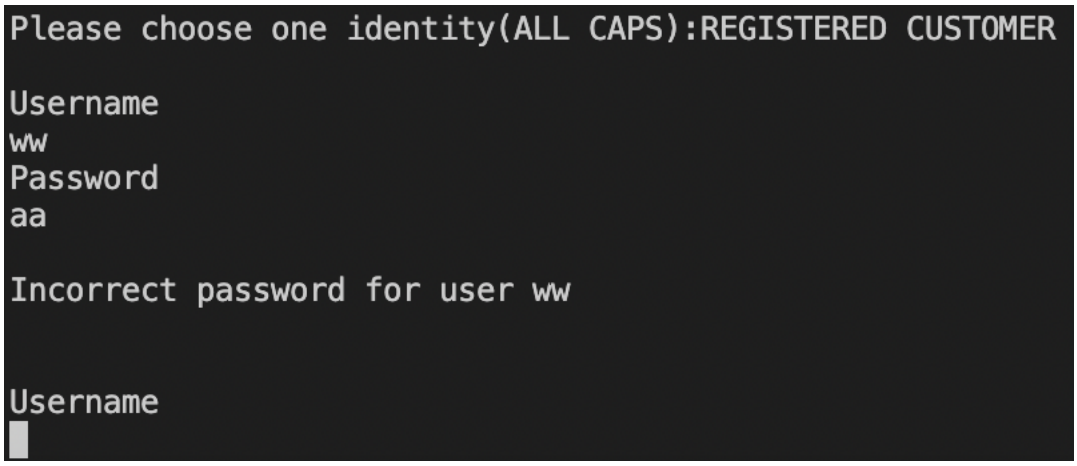

Figure 4.3.2: Login with Wrong Password

First, if the password entered by the user is incorrect, the system will require the user to re-enter it again.

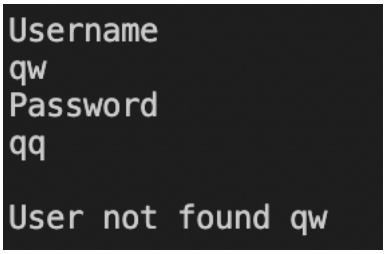

Figure 4.3.3: Login with Wrong Username

Or if the username entered by the user is wrong, the system will warn the user that the username does not exist, so that the user can enter one more time.

```
Username
ww
Password
qq

Welcome ww


                                        Groceries
   GROCERY       ________PRICE________
1. WATERMELON:          RM 34
2. GRAPE:               RM 34
3. APPLE:               RM 69
Want to purchase?(yes/no):
```

Figure 4.3.4: Login Successful and do Purchase

The system will display a successful login screen and the user's grocery list if the user's login username and password are accurate. The user will be prompted to purchase groceries after seeing their list of necessary items.

```
                                        Groceries
   GROCERY       ________PRICE________
1. WATERMELON:          RM 34
2. GRAPE:               RM 34
3. APPLE:               RM 69
Want to purchase?(yes/no):no
```

Figure 4.3.5: User No Do Purchase

If the user decides not to buy the product, the system will immediately reveal the user's personal data, like name, gender, and so on. Details are shown in Figure 4.3.13.

```
                                        Groceries
   GROCERY       ________PRICE________
1. WATERMELON:          RM 34
2. GRAPE:               RM 34
3. APPLE:               RM 69
Want to purchase?(yes/no):yes
Add grocery U want:apple
Quantity:1
{'APPLE': {'QUANTITY': 1, 'SUBTOTAL': 69.0}}
Want do more purchase?(yes/no):
```

Figure 4.3.6: User Make Purchase

The system will ask the user what groceries and how much they want to buy if they choose to purchase and will then print out the shopping list for them to review.

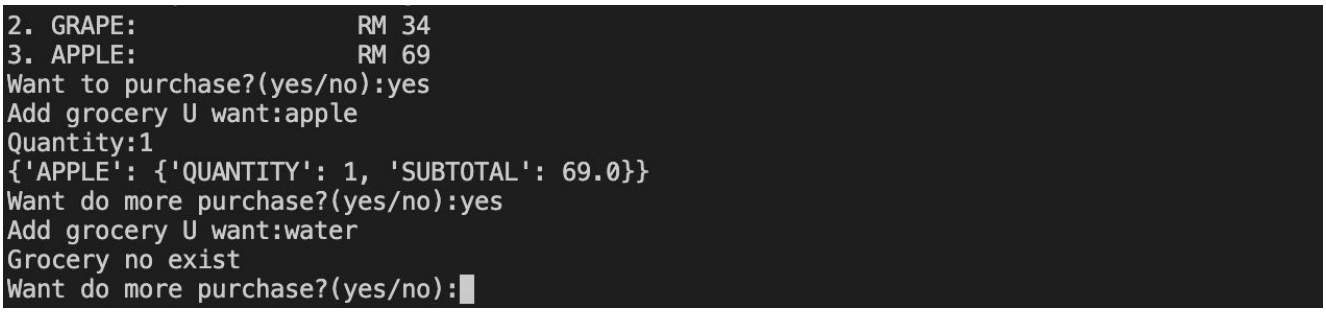

Figure 4.3.7: Non-existent Grocery

The system will prompt the user to enter again if the user's chosen groceries are not included in the list and will notify them as such.

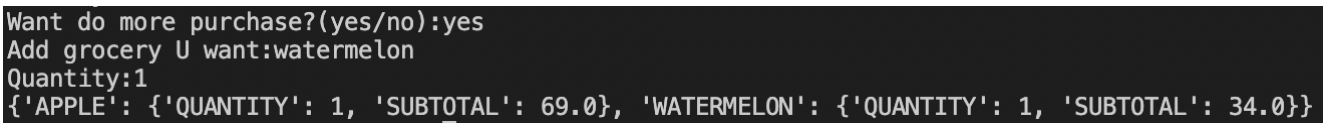

Figure 4.3.8: Make Purchase Again

The user will be questioned about making additional purchases after completing their initial transaction.

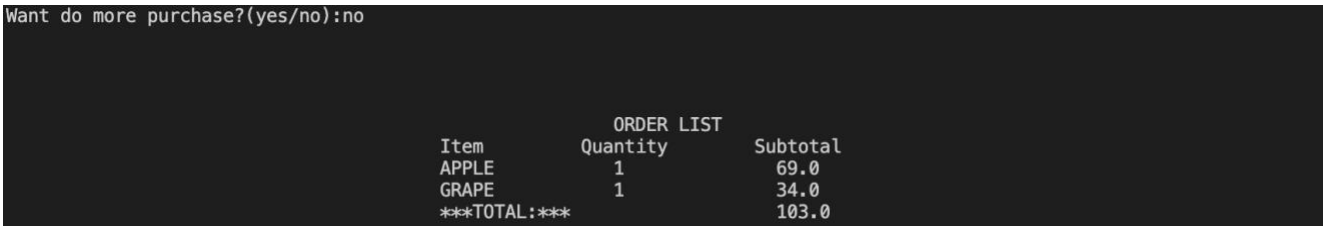

Figure 4.3.9: End for Purchase

The system will print out the user's purchase list, which includes the groceries purchased, the quantity, and the total amount, if the user decides not to purchase the item again.

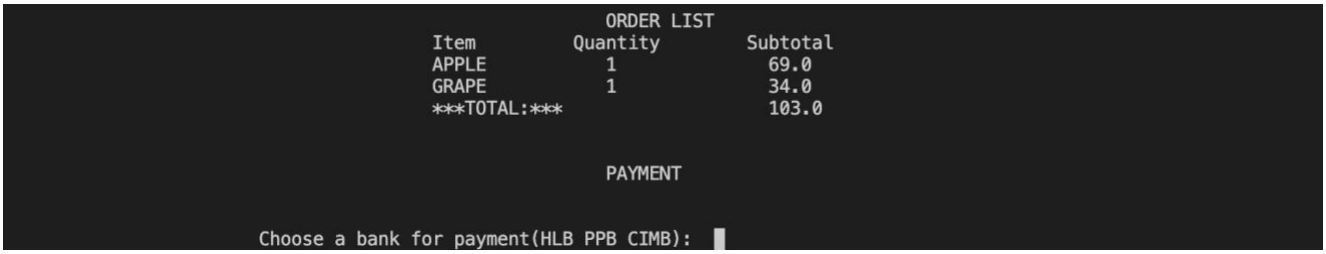

Figure 4.3.10: Make Payment

The user will examine the purchase list before continuing with the payment procedure. The user's only payment option currently is through the designated bank.

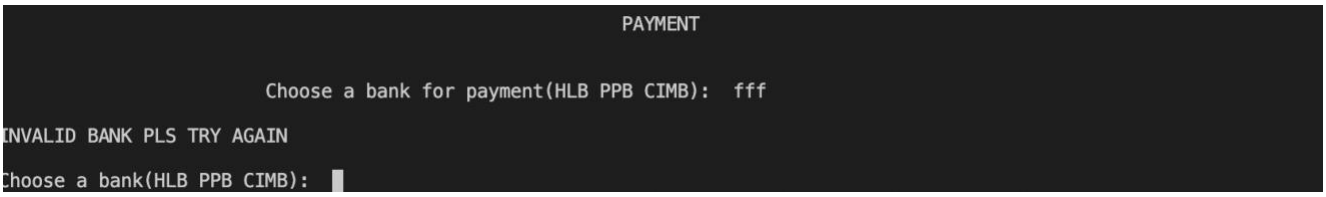

Figure 4.3.11: Non-existent Bank

The system will notify the user that the bank that entered does not exist and allow user to try again if it does not appear in the list that has been provided.

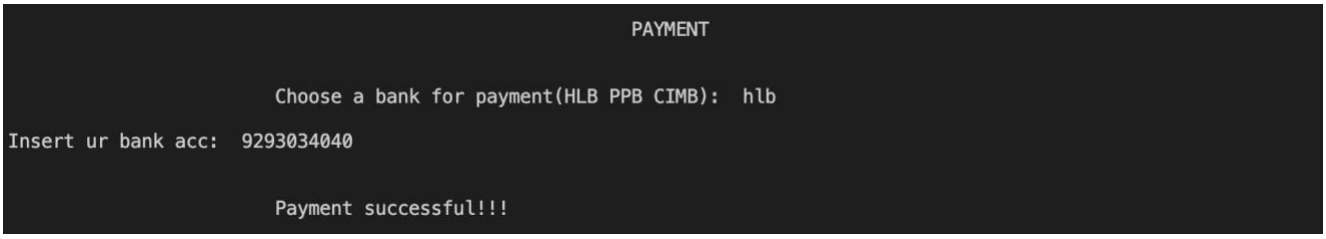

Figure 4.3.12: Payment Successful

When the user types in the right bank and account number, the payment is considered successful.

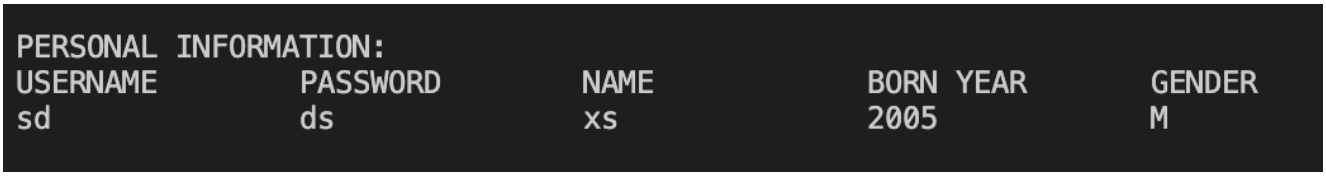

Figure 4.3.13: User Personal Information

The system will skip to this section, which contains the user's personal information, if the user has finished the purchase or decides not to buy the item. Name, gender, birth year, and other information are included in the material.

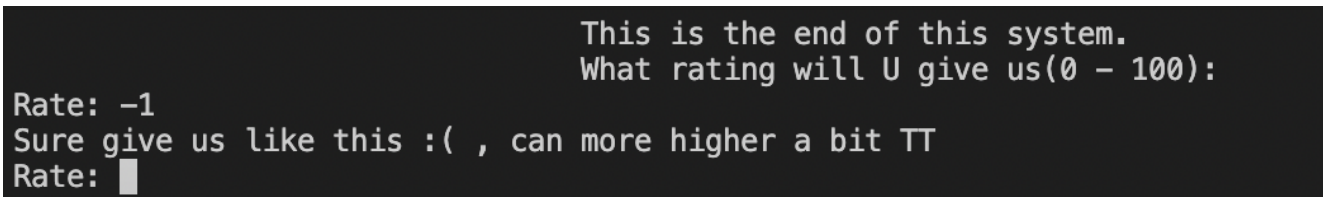

Figure 4.3.14: Rating

Before leaving the system, the user will be asked to rate it to help it get better upgrades in the future.

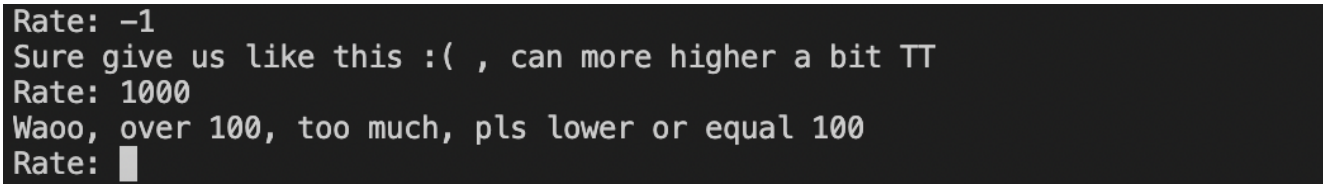

Figure 4.3.15: Rating Out of Value

User will be prompted to reapply if the rating is below or out of the range.

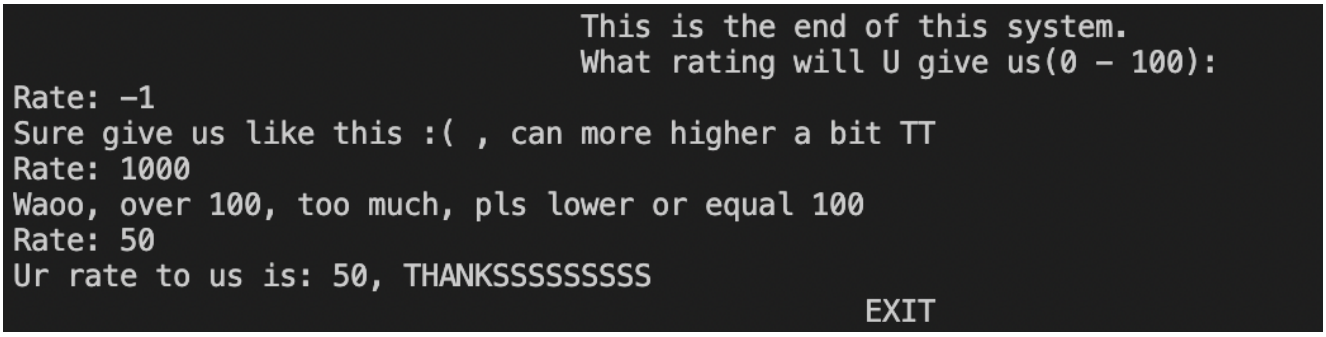

Figure 4.3.16: Rating Successful

The system will prompt the user to thank them for their rating and quit the system after the rating is complete.

4.4 NEW CUSTOMER

The new customer system will be easier to understand and simpler than the other two identify systems.

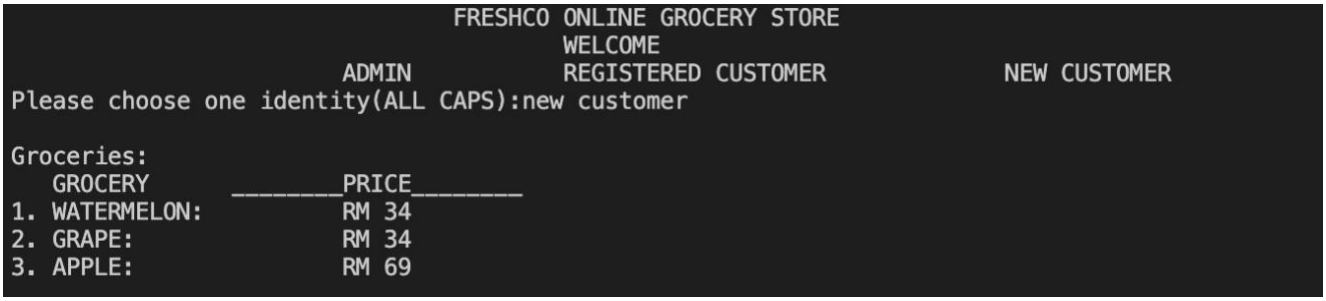

Figure 4.4.1: New Customer

The user will get a list of groceries when they first use this system.

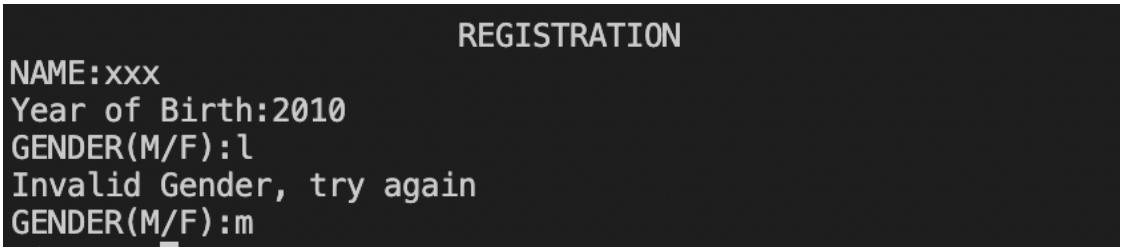

Figure 4.3.1: Registration

The user enters the registration section after viewing the list and providing their name, gender, and year of birth. The user will be prompted to submit their information again if they enter a gender that is not M or F.

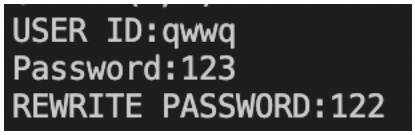

Figure 4.3.1: User ID and Password

The most crucial component is next: the login and password. The user must check this box to see if the password is the same to finish the registration process. The user must repeatedly input the passwords until they match if they don't.

Figure 4.3.1: Registration Successful

The system will exit automatically if your password matches, which indicates that your registration was successful.

5.0 Conclusion

The system runs smoothly in all respects, and it can carry out the user's requests across the board. Although it's a little software, I'm extremely proud of this system because it could easily complete this report on its own. This method will therefore live on in my memory and in my heart forever.

This report taught me not only the prerequisites and skills needed to build a system, but also the challenges that must be overcome along the way, such as those related to programming, data search, and other issues. But it was because of these challenges that a successful system could be developed, and it did so successfully.

On the other hand, because of this report, my research skills and logical thinking have also improved. I have learned how to locate resources accurately and effectively as well as how to use the system's logic to gather information and direct me in the right direction to obtain the information I require for this project.

(3234 words)