A · P · U

ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

Individual Assignment

Python Programming

STUDENT NAME: GASSAR HAITHM SALEH BA HASHWAN

TP. NO: TP608741

INTAKE CODE: APD1F2206CS(CYB)

MODULE CODE: CT108-3-1-PYP

MODULE NAME: Python Programming

LECTURER NAME: Mr VIKNESH A/L RAMAMOORTHY

HAND-IN DATE: 9 October 2022

Table of Contents

Introduction and Assumptions

The rise of grocery stores companies on the internet has made it much simpler to stock up on supplies for the pantry. You will learn that there are a lot of advantages to doing your grocery shopping online, regardless of whether this is your first time utilising a grocery delivery service or if you are an experienced online shopper (Denisewitsch, 2021).

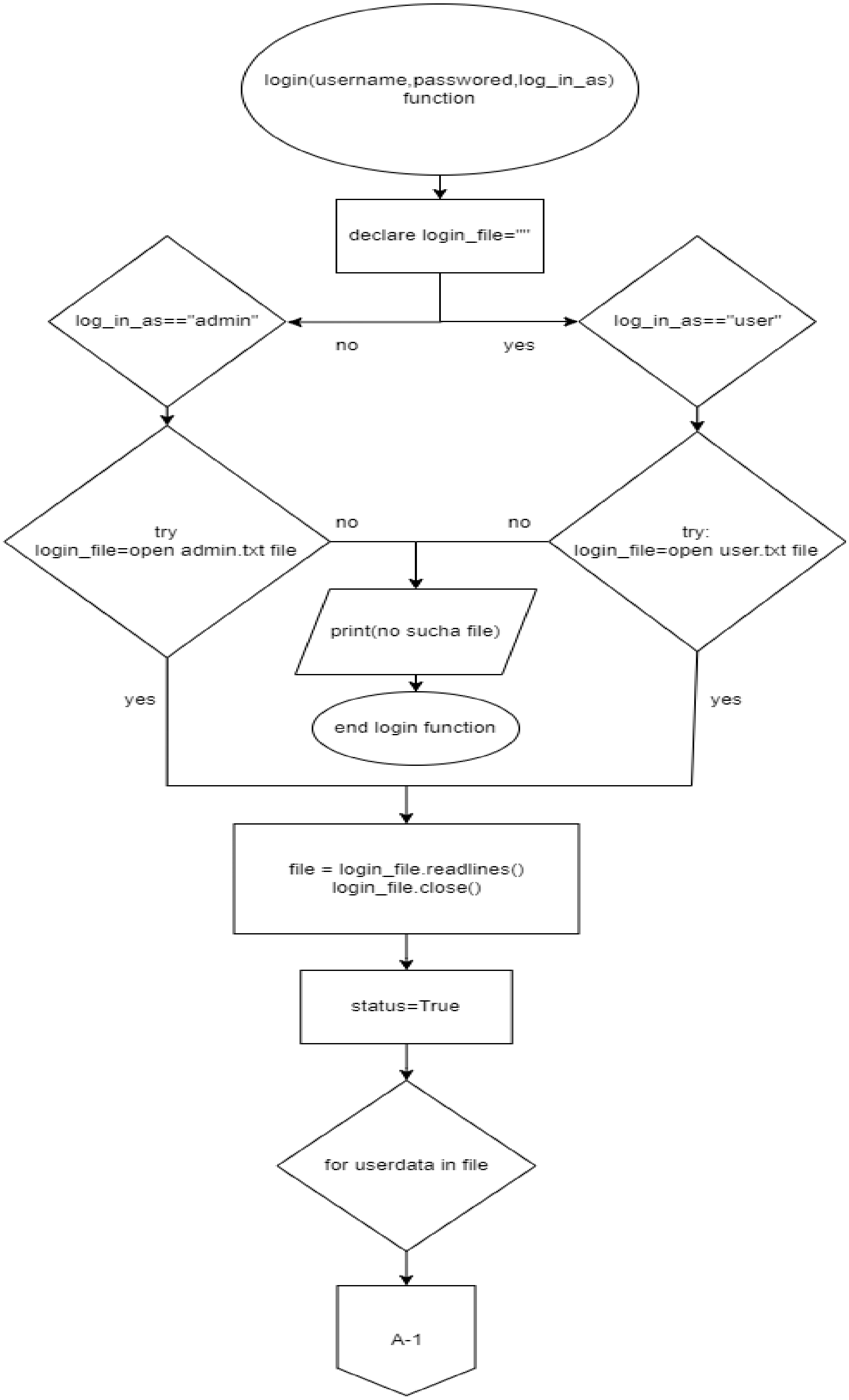
Online shopping was the greatest option to tackle the majority of concerns during the COVID-19 pandemic, including how to purchase groceries. However, the convenience of an online grocery shop ensured that customers didn't have to make the trip to the market on their own to buy their groceries.

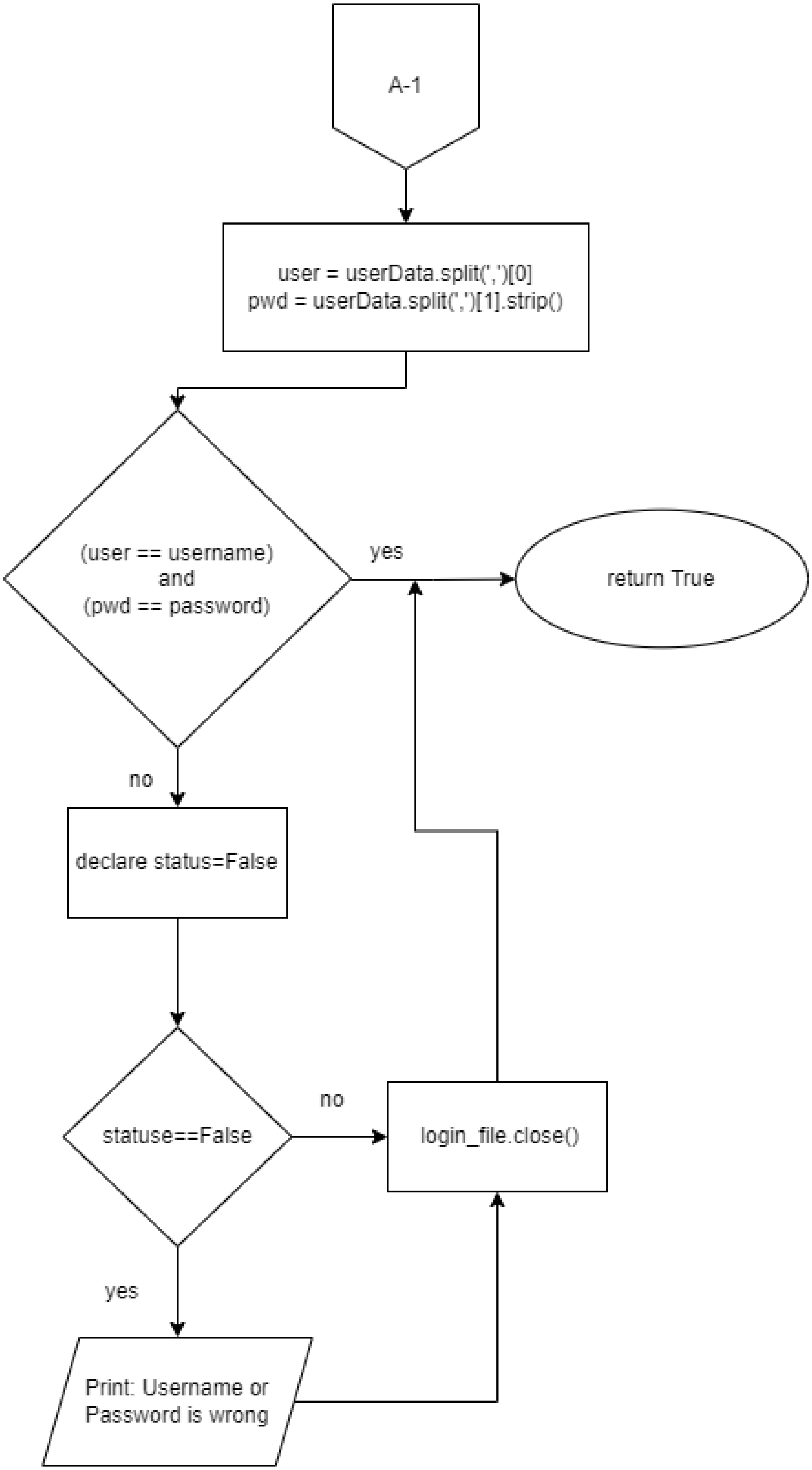
I was hired by FRESHCO Sdn Bhd, a major company in Malaysia, to develop a program for a grocery store for the company.

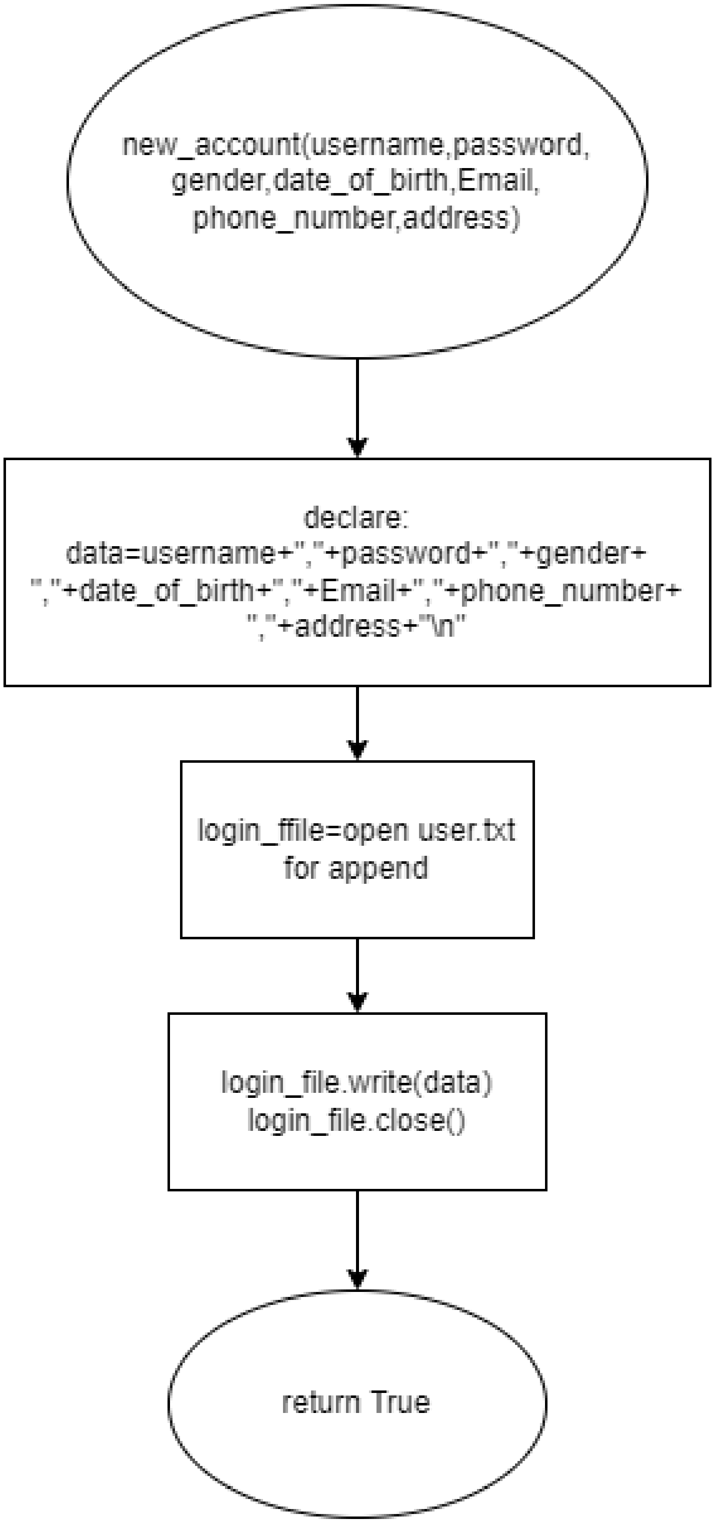
The first person to utilise this program will need to be a customer of the grocery store, and they must be able to access the menu, place a new order, see their own order, and view their own information. Additionally, this consumer has to have an account. The second user is known as the admin, and their responsibilities include adding, modifying, and deleting groceries, as well as going over orders placed by prior customers. The last user category is for guests, who may either browse the grocery store or create a new account if they so want.

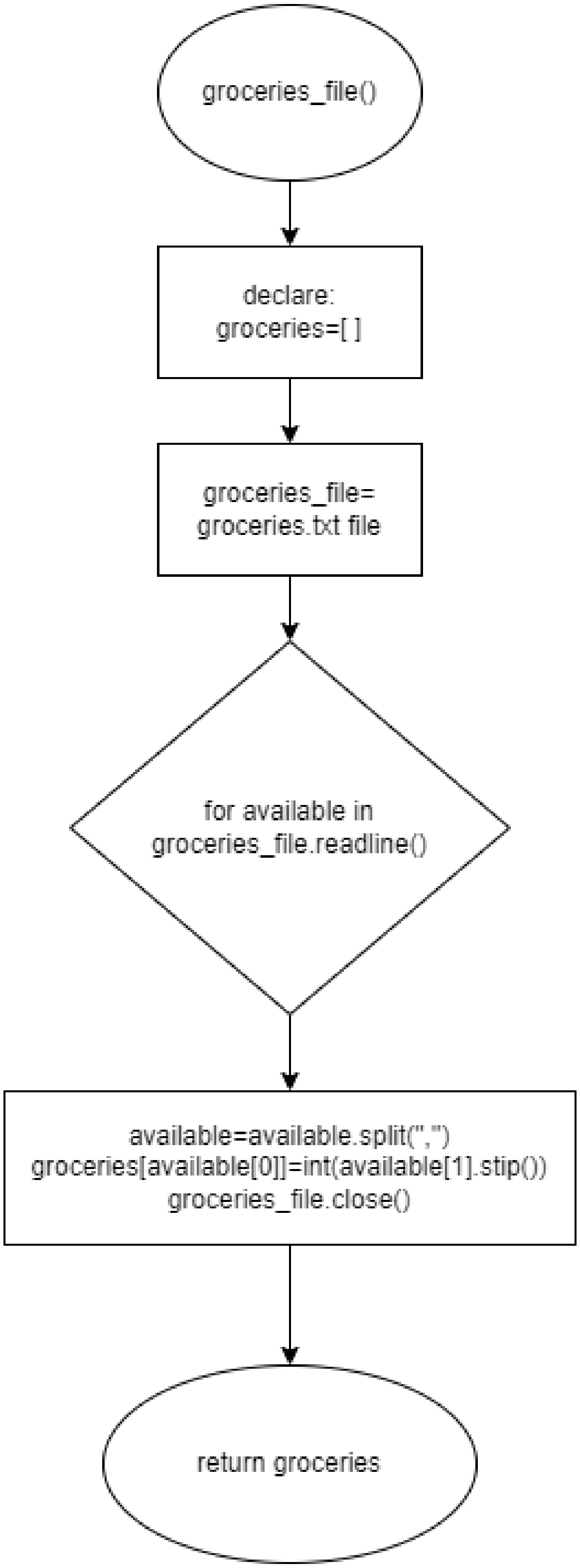
For this program, I will be using Python as the programing language. The creation of the program that fulfils the requirements of both the store and its customers is one of the things that are expected of me. I will be providing this program with a Flowchart and pseudocode, also samples of the program with justification. All the information will be saved to text files that will be sent with the project file.

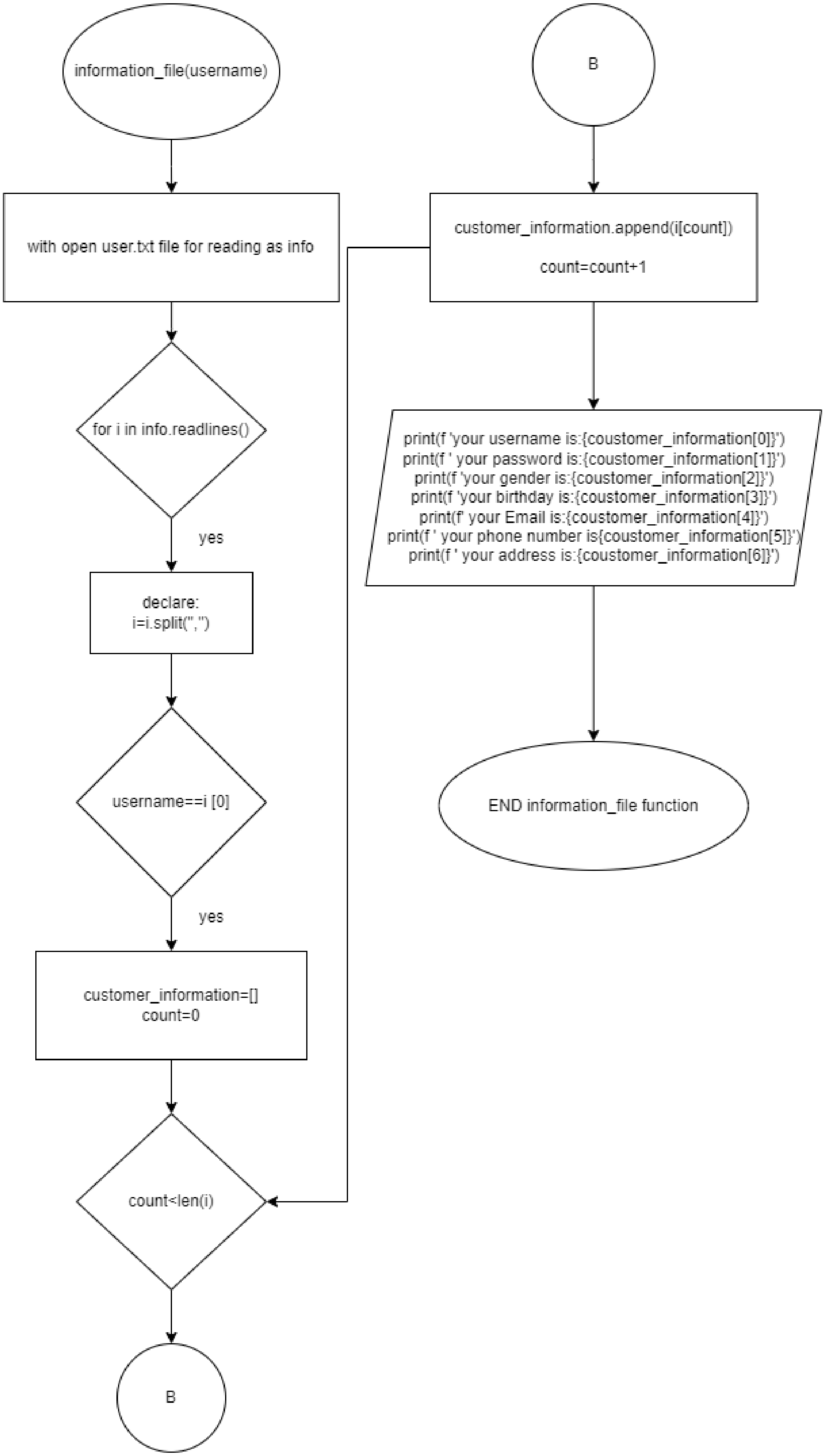
Flowchart

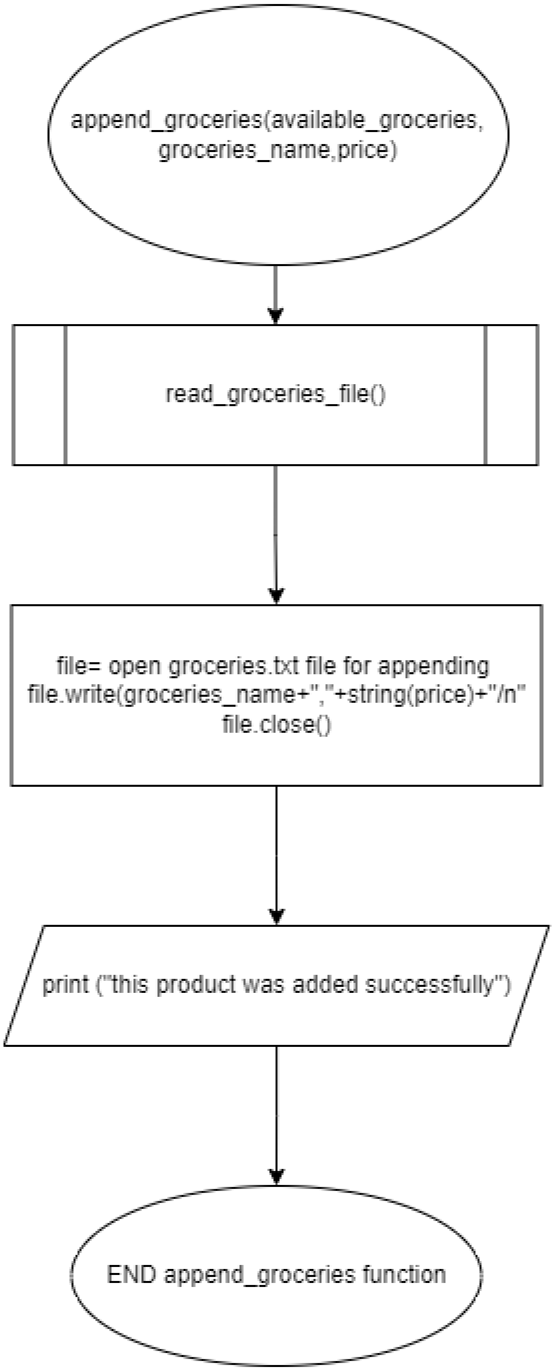

login(username,passwored,log_in_as)
function
declare login_file=""
log_in_as=="admin"
no
yes
log_in_as=="user"
try
login_file=open admin.txt file
no
no
try:
login_file=open user.txt file
print(no sucha file)
yes
end login function
yes
file = login_file.readlines()
login_file.close()
status=True
for userdata in file
A-1

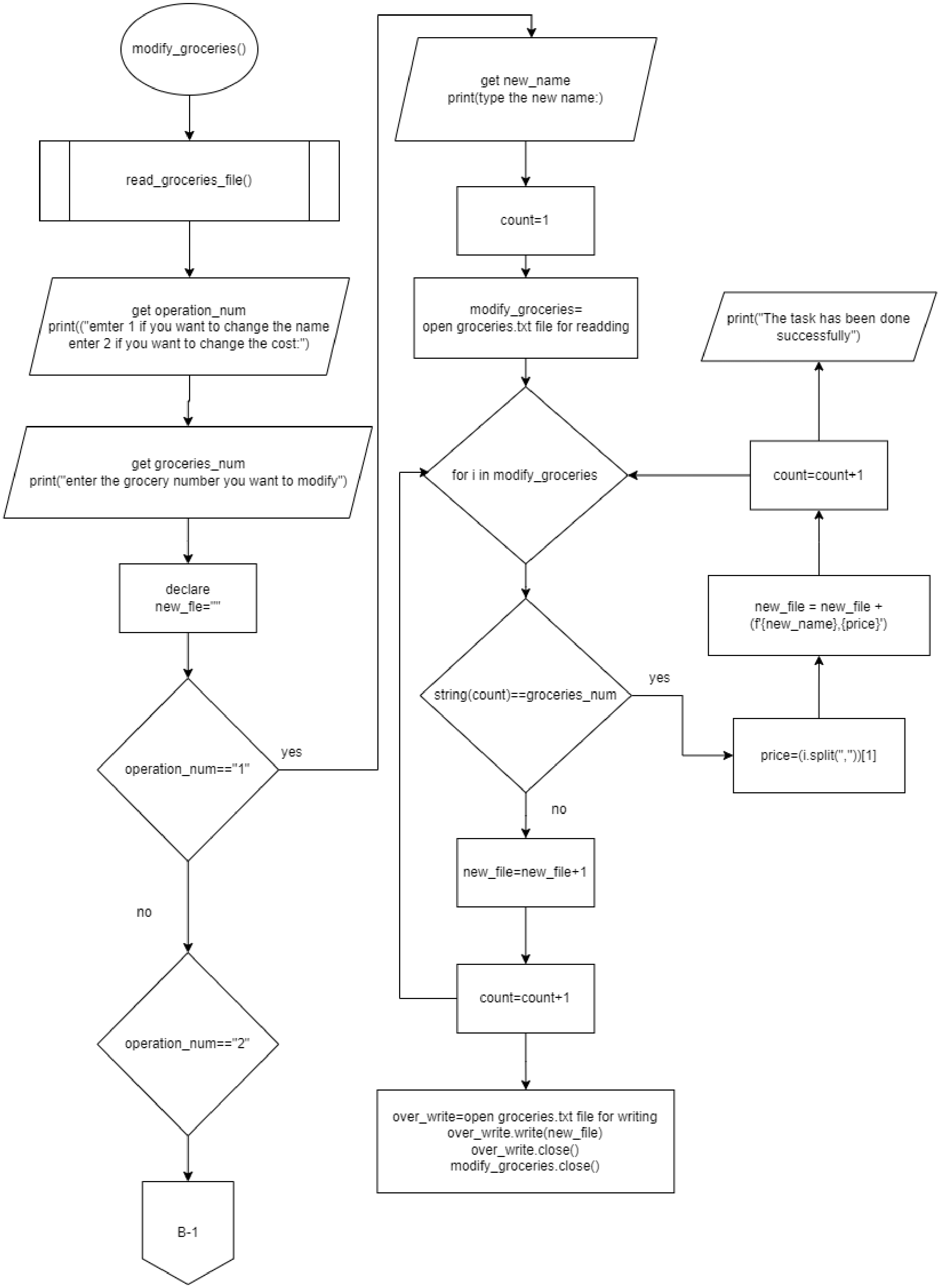

A-1
user = userData.split(',')[0]
pwd = userData.split(',')[1].strip()
(user == username)
and
(pwd == password)
yes
return True
no
declare status=False
statuse==False
no
login_file.close()
yes
Print: Username or
Password is wrong

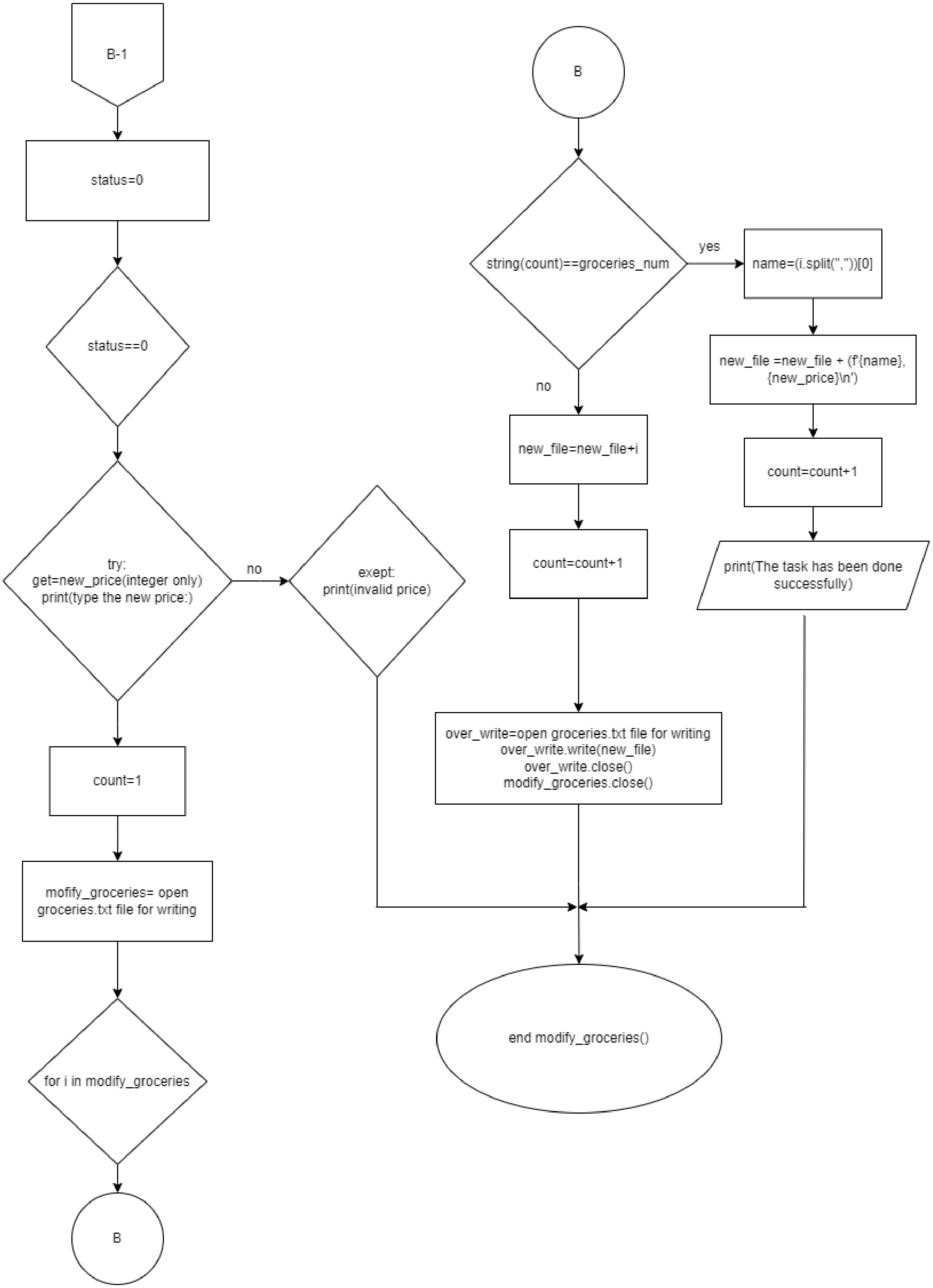

new_account(username,password,
gender,date_of_birth,Email,
phone_number,address)
declare:
data=username+","+password+","+gender+
","+date_of_birth+","+Email+","+phone_number+
","+address+"\n"
login_ffile=open user.txt
for append
login_file.write(data)
login_file.close()
return True

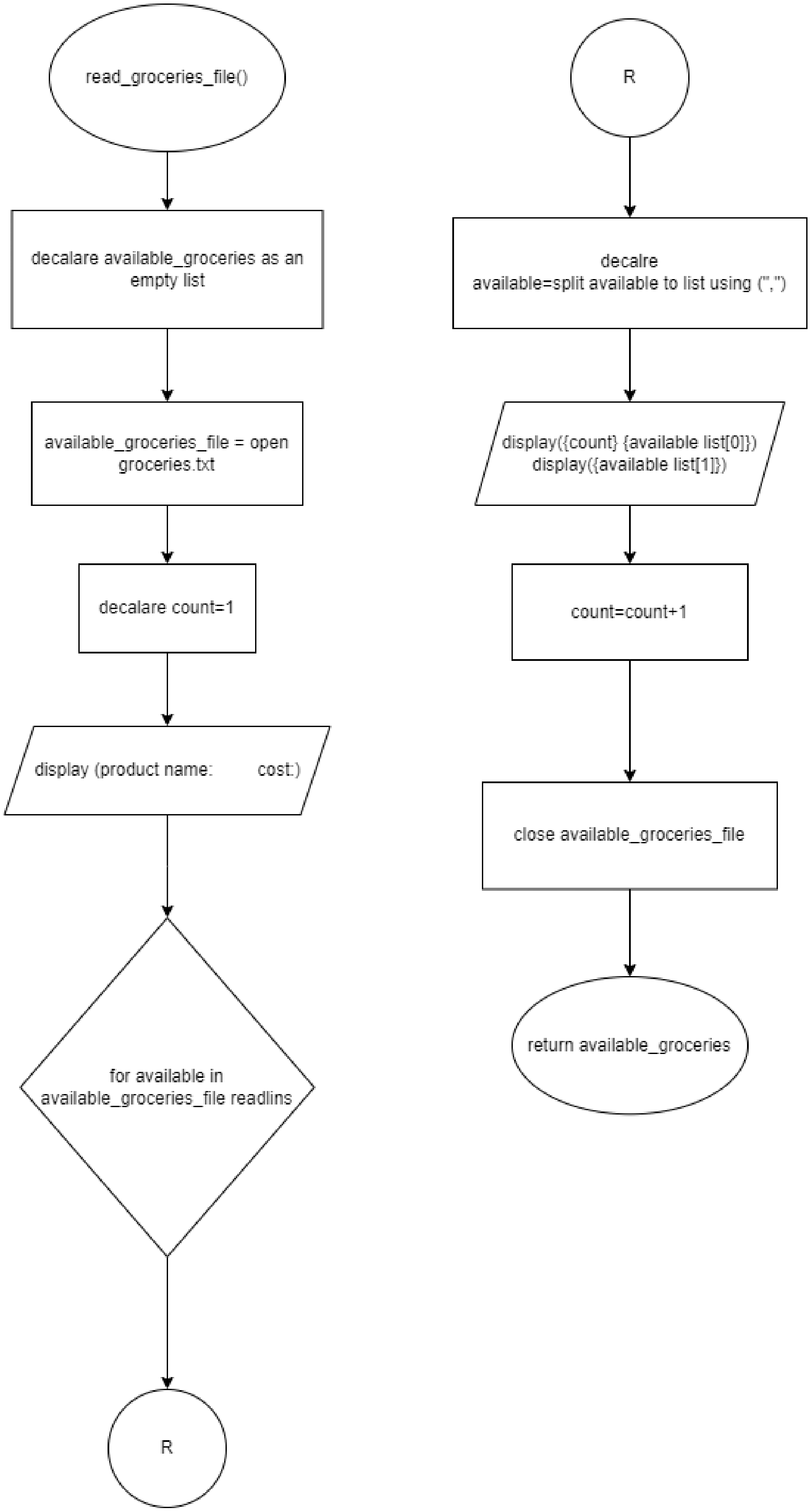

groceries_file()
declare:
groceries=[ ]
groceries_file=
groceries.txt file
for available in
groceries_file.readline()
available=available.split(",")
groceries[available[0]]=int(available[1].stip())
groceries_file.close()
return groceries

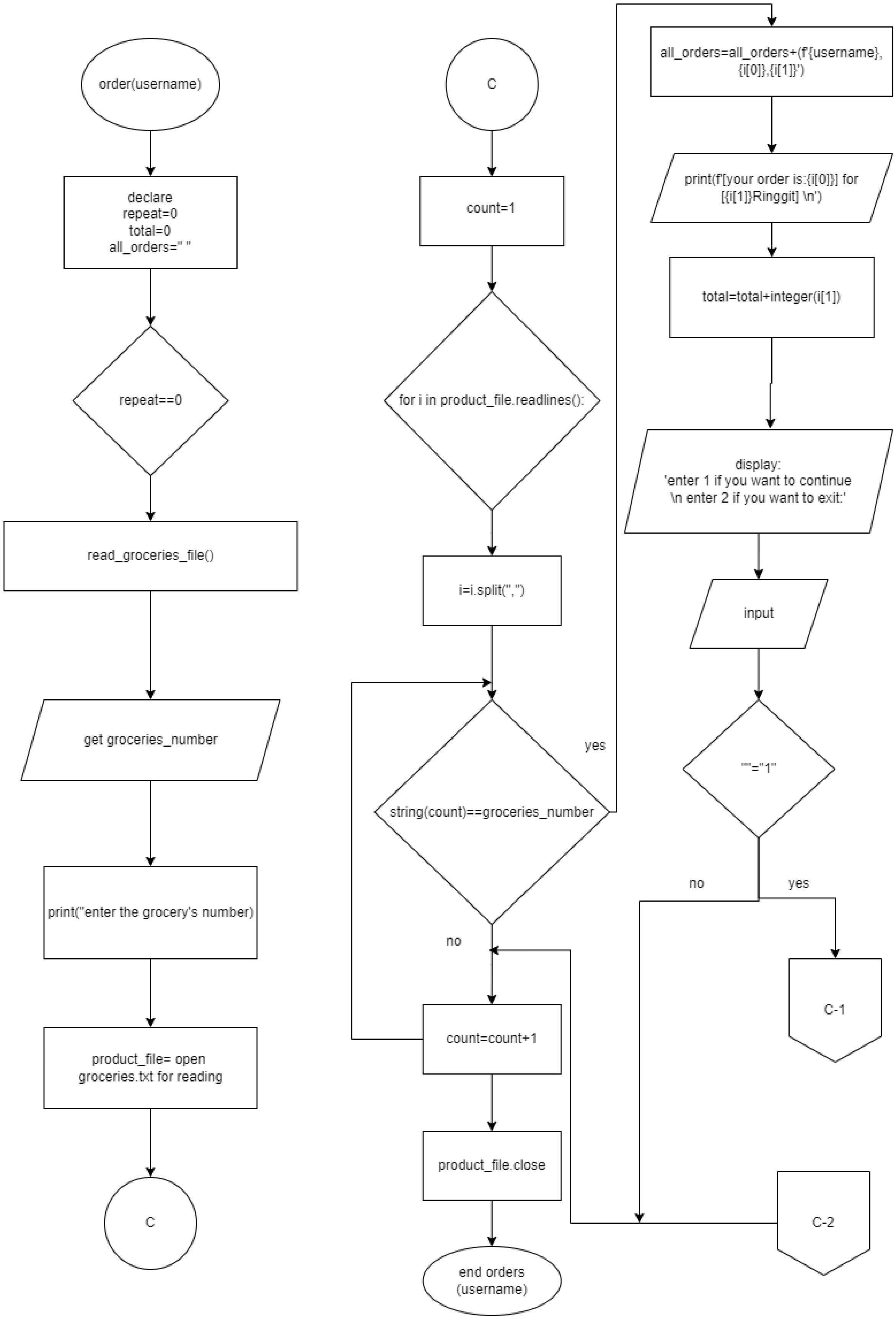

information_file(username)
with open user.txt file for reading as info
for i in info.readlines()
yes
declare:
i=i.split(",")
username==i [0]
yes
customer_information=[]
count=0
count<len(i)
B
B
customer_information.append(i[count])
count=count+1
print(f 'your username is:{coustomer_information[0]}')
print(f ' your password is:{coustomer_information[1]}')
print(f 'your gender is:{coustomer_information[2]}')
print(f 'your birthday is:{coustomer_information[3]}')
print(f' your Email is:{coustomer_information[4]}')
print(f ' your phone number is{coustomer_information[5]}')
print(f ' your address is:{coustomer_information[6]}')
END information_file function

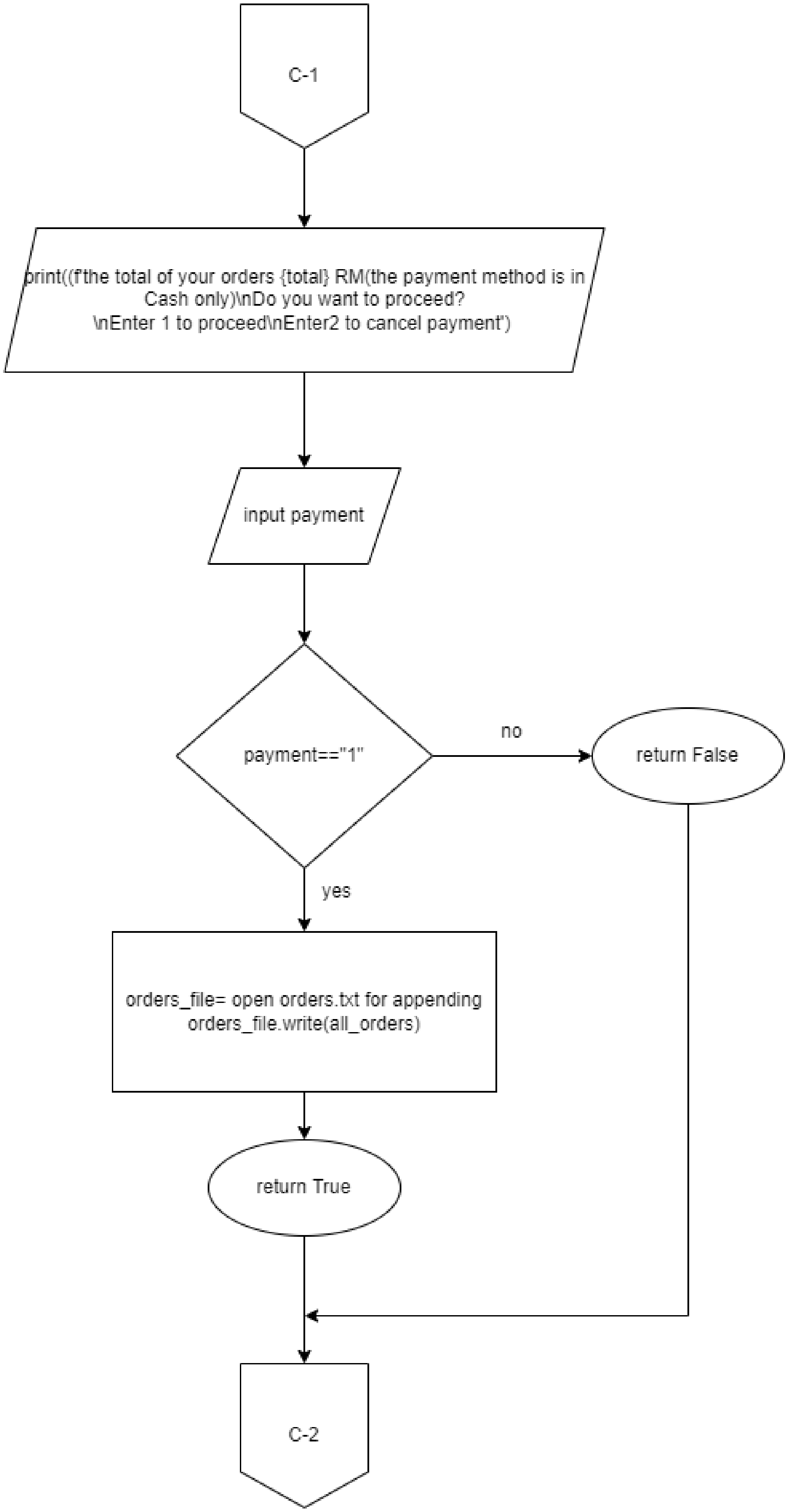

append_groceries(available_groceries, groceries_name,price)
read_groceries_file()
file= open groceries.txt file for appending
file.write(groceries_name+","+string(price)+"/n"
file.close()
print ("this product was added successfully")
END append_groceries function

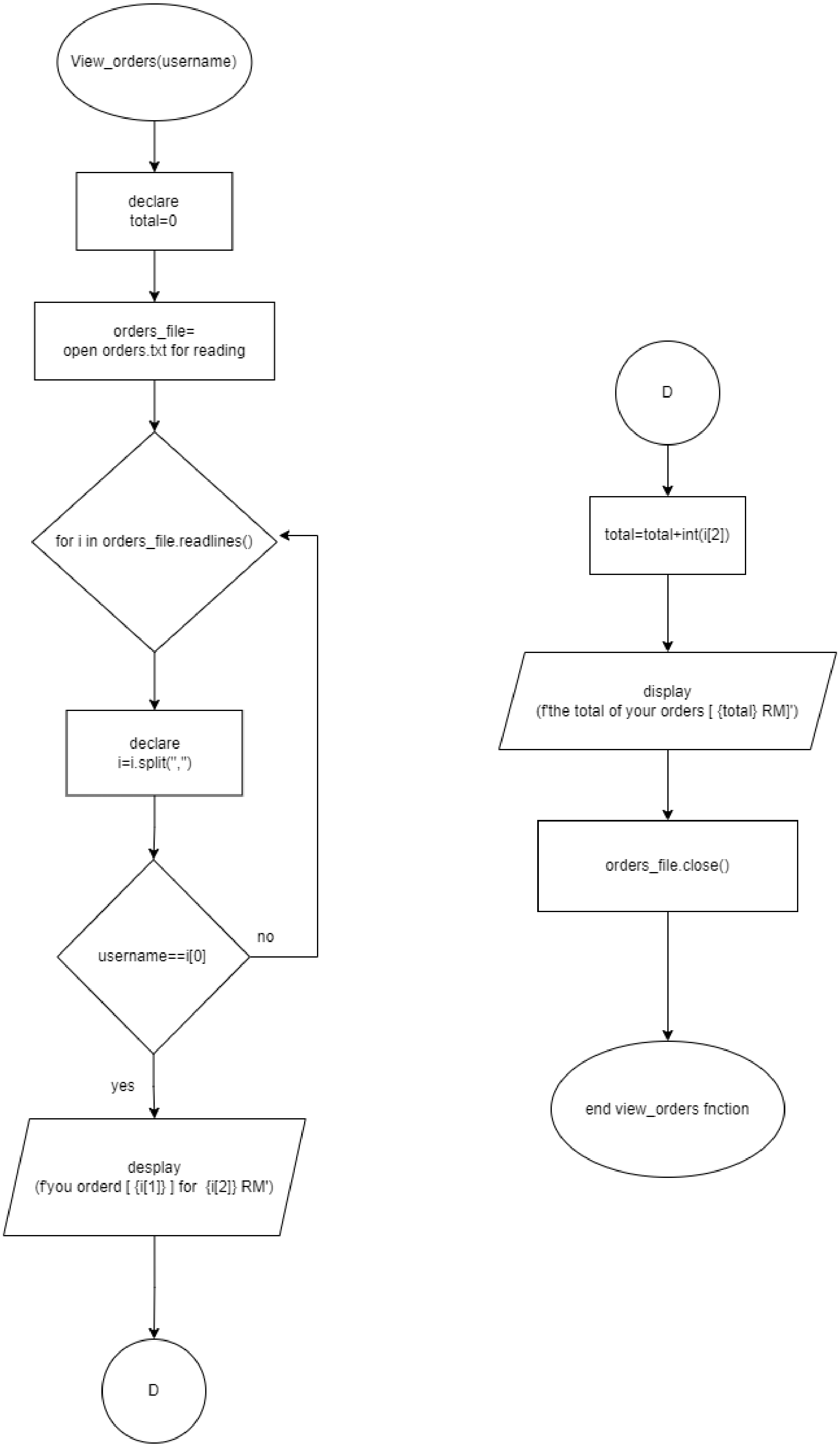

modify_groceries()
read_groceries_file()
get operation_num
print(("emter 1 if you want to change the name
enter 2 if you want to change the cost:")
get groceries_num
print("enter the grocery number you want to modify")
declare
new_fle=""
operation_num=="1"
yes
no
operation_num=="2"
B-1
get new_name
print(type the new name:)
count=1
modify_groceries=
open groceries.txt file for readding
for i in modify_groceries
string(count)==groceries_num
yes
no
new_file=new_file+1
count=count+1
over_write=open groceries.txt file for writing
over_write.write(new_file)
over_write.close()
modify_groceries.close()
price=(i.split(","))[1]
new_file = new_file +
(f'{new_name},{price}')
count=count+1
print("The task has been done
successfully")

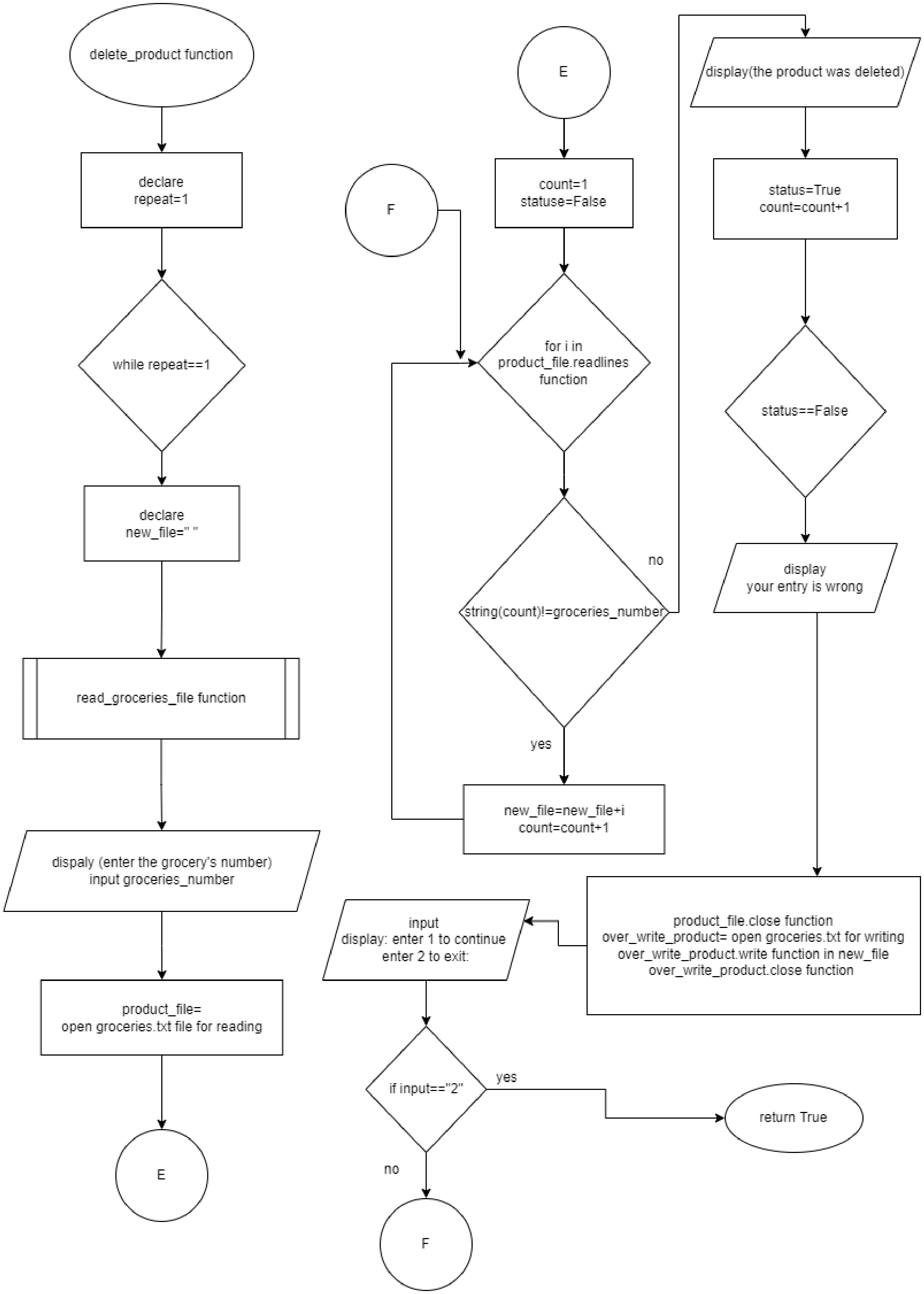

B-1
status=0
status==0
try:
get=new_price(integer only)
print(type the new price:)
no
exept:
print(invalid price)
count=1
mofify_groceries= open
groceries.txt file for writing
for i in modify_groceries
B
B
string(count)==groceries_num
yes
name=(i.split(","))[0]
new_file =new_file + (f'{name},
{new_price}\n')
count=count+1
print(The task has been done
successfully)
no
new_file=new_file+i
count=count+1
over_write=open groceries.txt file for writing
over_write.write(new_file)
over_write.close()
modify_groceries.close()
end modify_groceries()

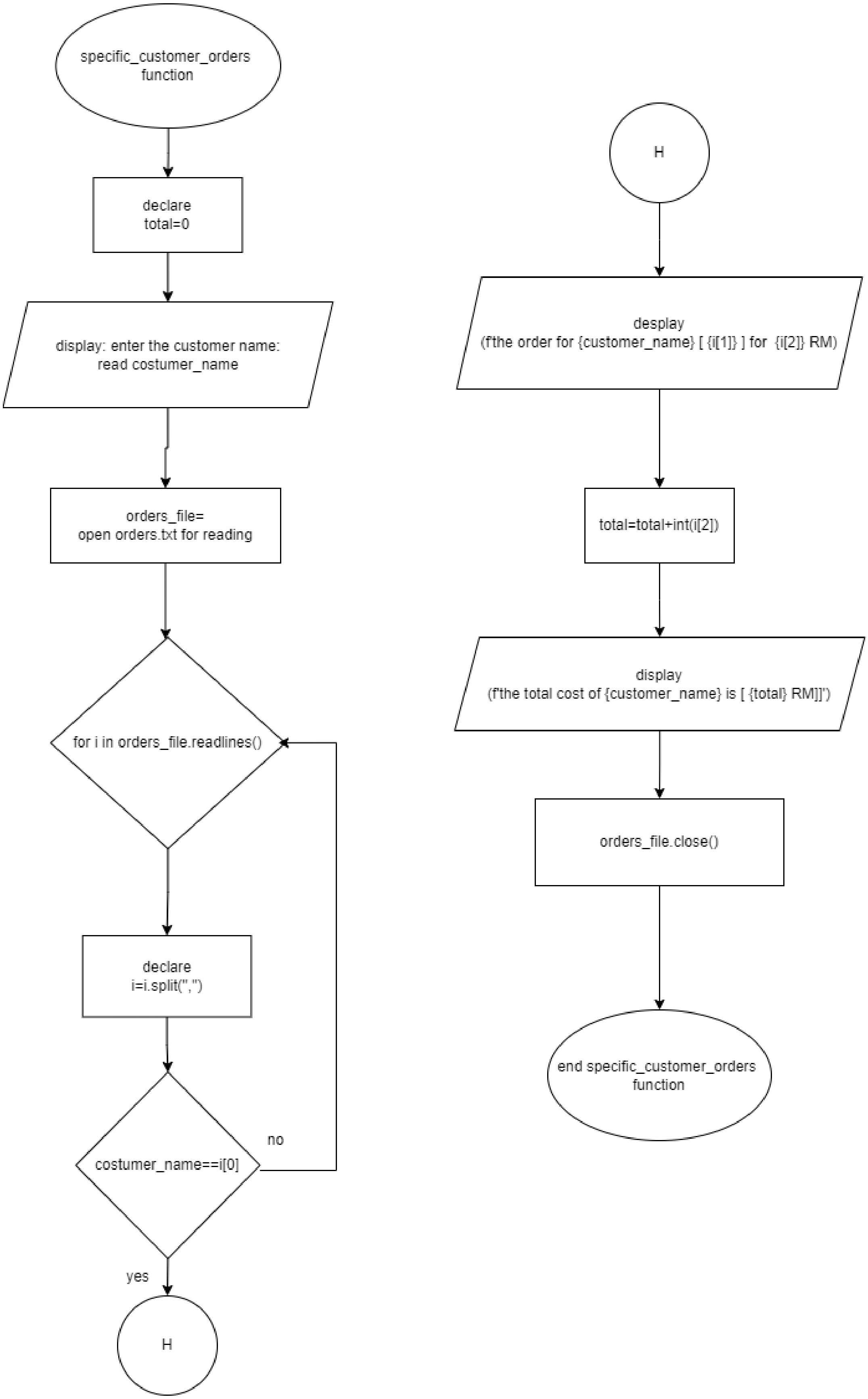

read_groceries_file()
decalare available_groceries as an empty list
available_groceries_file = open groceries.txt
decalare count=1
display (product name: cost:)
for available in available_groceries_file readlins
R
R
decalre available=split available to list using (",")
display({count} {available list[0]})
display({available list[1]})
count=count+1
close available_groceries_file
return available_groceries

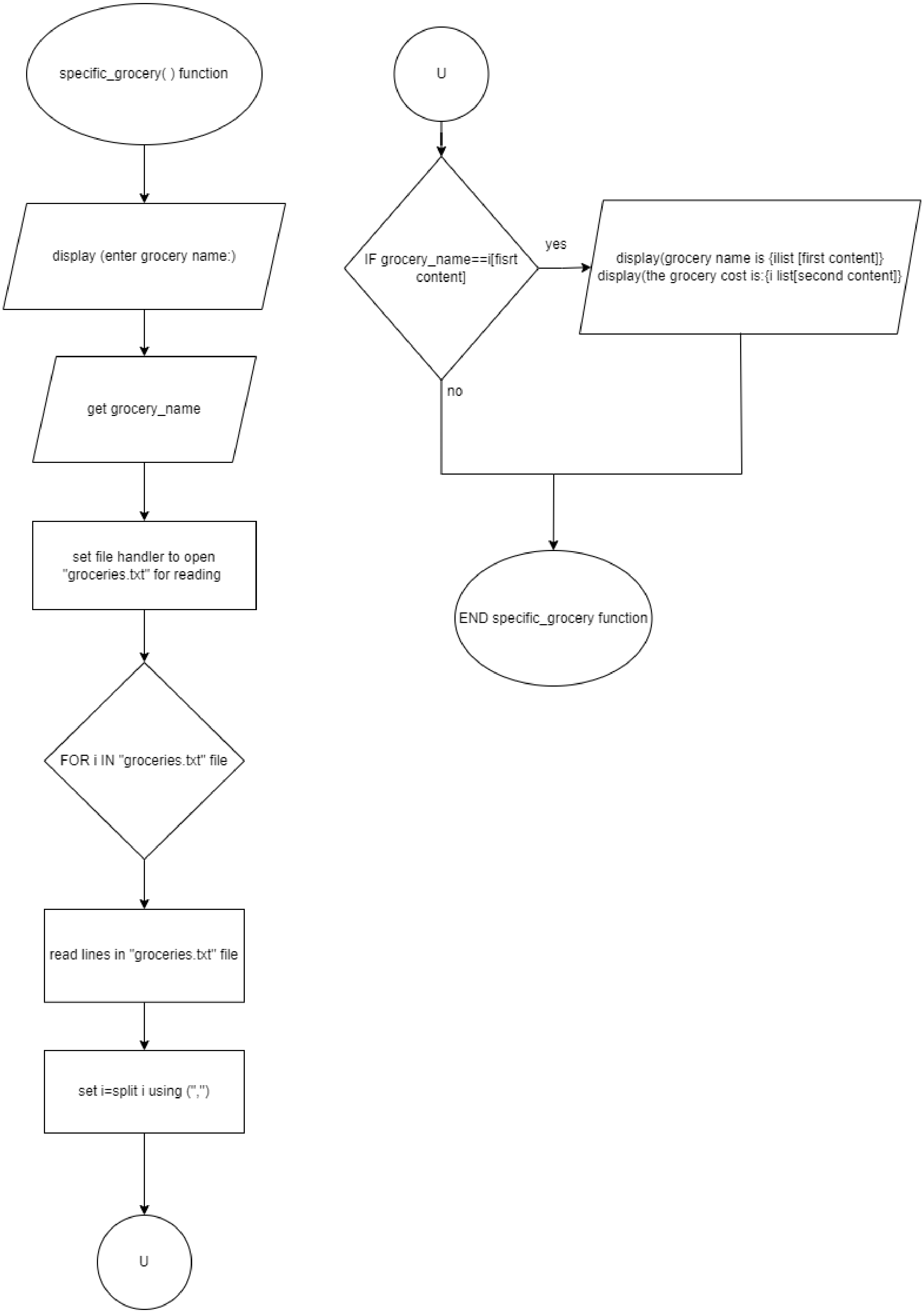

order(username)
declare
repeat=0
total=0
all_orders=" "
repeat==0
read_groceries_file()
get groceries_number
print("enter the grocery's number)
product_file= open groceries.txt for reading
C
C
count=1
for i in product_file.readlines():
i=i.split(",")
yes
string(count)==groceries_number
no
count=count+1
product_file.close
end orders (username)
all_orders=all_orders+(f'{username},{i[0]},{i[1]}')
print(f'[your order is:{i[0]}] for [{i[1]}Ringgit] \n')
total=total+integer(i[1])
display:
'enter 1 if you want to continue
\n enter 2 if you want to exit:'
input
""="1"
no
yes
C-1
C-2

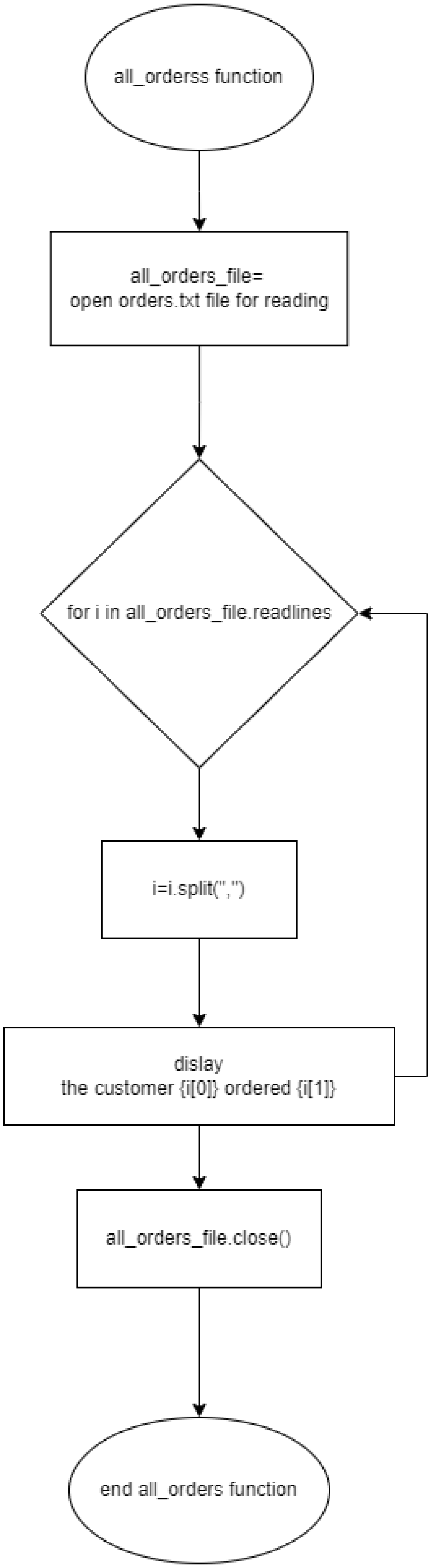

C-1
print((f'the total of your orders {total} RM(the payment method is in Cash only)\nDo you want to proceed?
\nEnter 1 to proceed\nEnter2 to cancel payment')
input payment
payment=="1"
no
return False
yes
orders_file= open orders.txt for appending
orders_file.write(all_orders)
return True
C-2

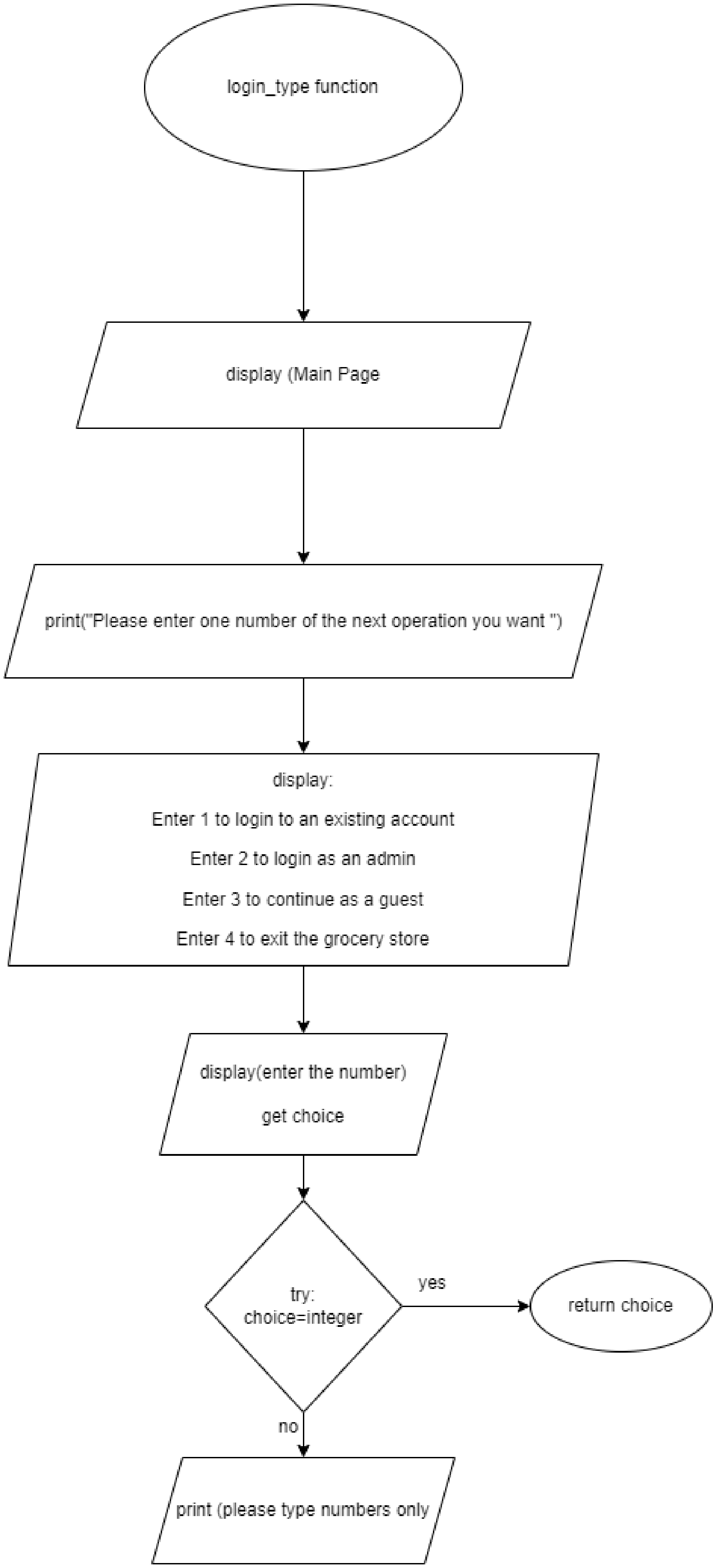

View_orders(username)
declare
total=0
orders_file=
open orders.txt for reading
for i in orders_file.readlines()
declare
i=i.split(",")
username==i[0]
no
yes
desplay
(f'you orderd [{i[1]}] for {i[2]} RM')
D
D
total=total+int(i[2])
display
(f'the total of your orders [{total} RM]')
orders_file.close()
end view_orders fnction

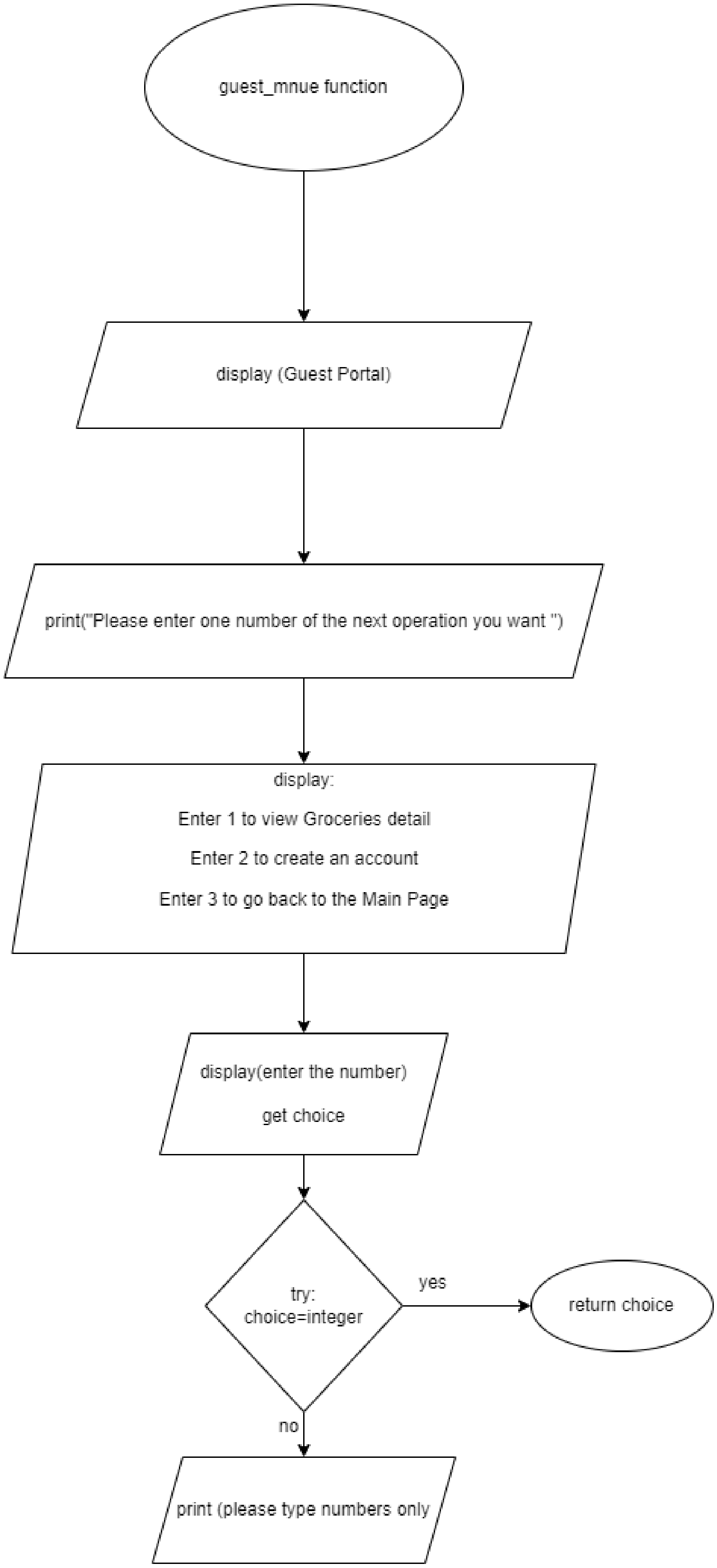

delete_product function
declare
repeat=1
while repeat==1
declare
new_file=" "
read_groceries_file function
dispaly (enter the grocery's number)
input groceries_number
product_file=
open groceries.txt file for reading
E
F
E
count=1
statuse=False
for i in
product_file.readlines
function
string(count)!=groceries_number
no
yes
new_file=new_file+i
count=count+1
display(the product was deleted)
status=True
count=count+1
status==False
display
your entry is wrong
product_file.close function
over_write_product= open groceries.txt for writing
over_write_product.write function in new_file
over_write_product.close function
input
display: enter 1 to continue
enter 2 to exit:
if input=="2"
yes
return True
no
F

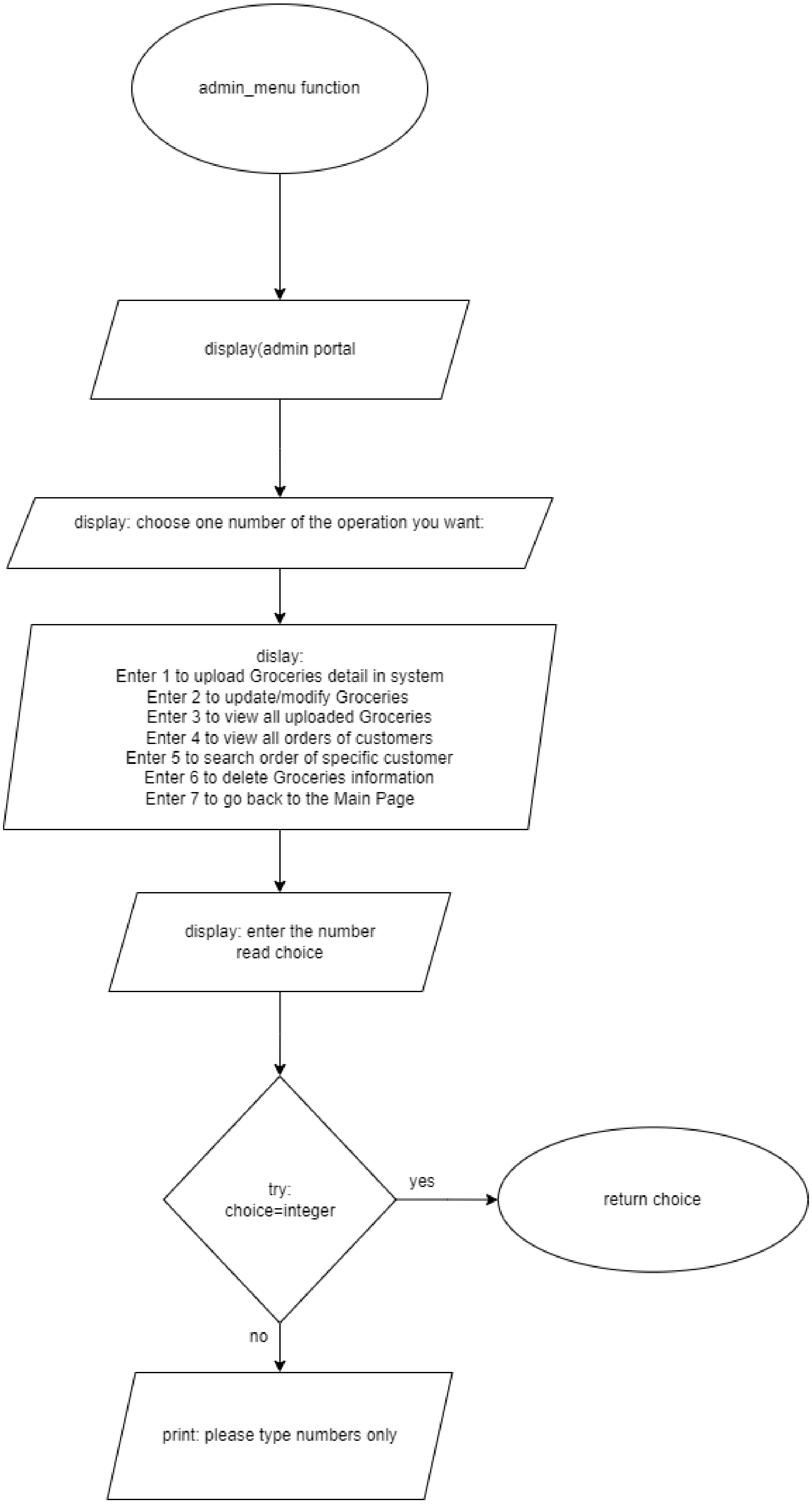

specific_customer_orders
function
declare
total=0
display: enter the customer name:
read costumer_name
orders_file=
open orders.txt for reading
for i in orders_file.readlines()
declare
i=i.split(",")
costumer_name==i[0]
no
yes
H
H
desplay
(f'the order for {customer_name} [{i[1]}] for {i[2]} RM)
total=total+int(i[2])
display
(f'the total cost of {customer_name} is [{total} RM]]')
orders_file.close()
end specific_customer_orders
function

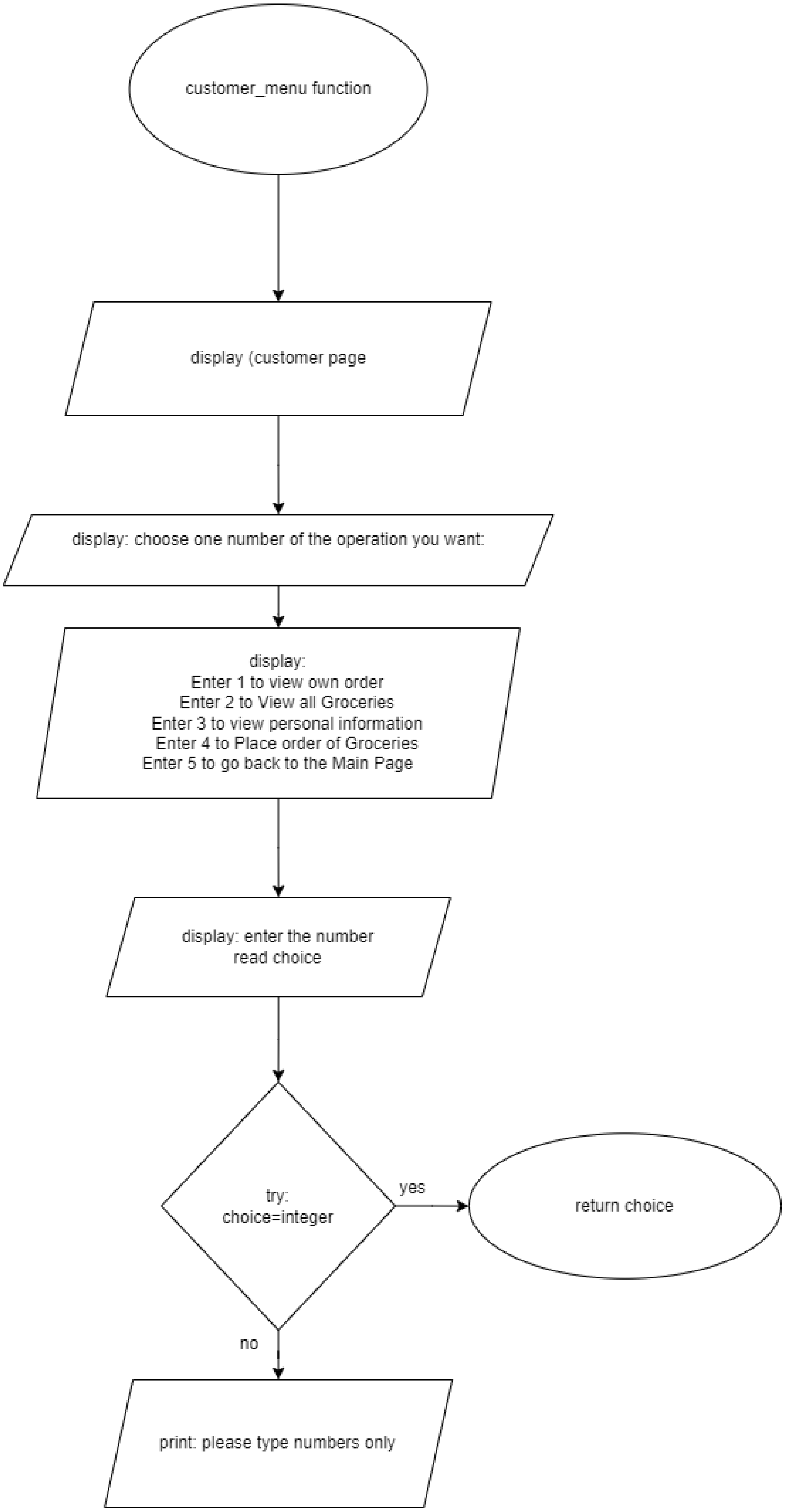

specific_grocery( ) function
display (enter grocery name:)
get grocery_name
set file handler to open "groceries.txt" for reading
FOR i IN "groceries.txt" file
read lines in "groceries.txt" file
set i=split i using (",")
U
U
IF grocery_name==i[fisrt content]
yes
display(grocery name is {ilist [first content]}
display(the grocery cost is:{i list[second content]}
no
END specific_grocery function

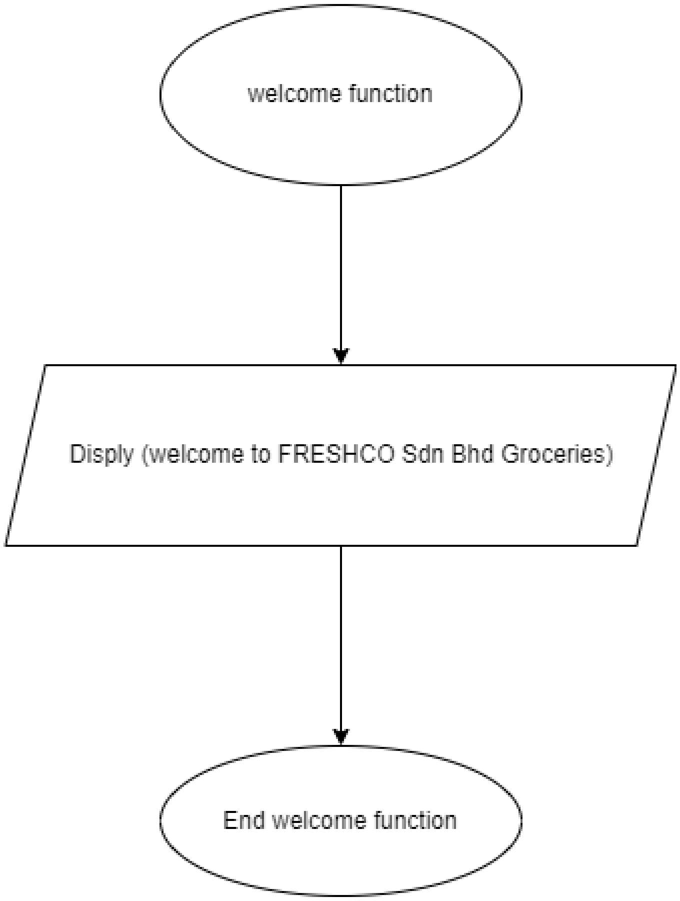

all_orderss function
all_orders_file=
open orders.txt file for reading
for i in all_orders_file.readlines
i=i.split(",")
dislay
the customer {i[0]} ordered {i[1]}
all_orders_file.close()
end all_orders function

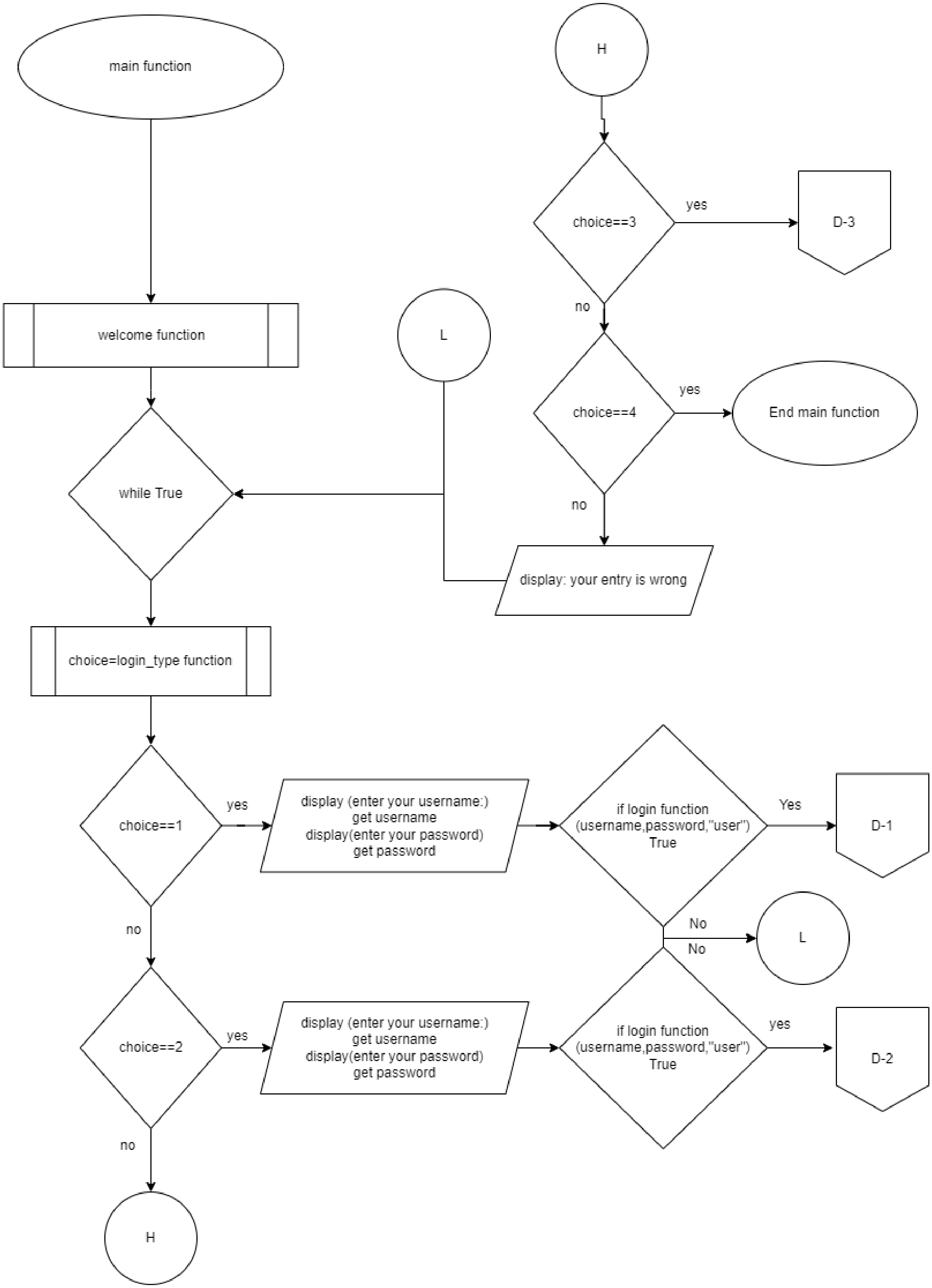

login_type function
display (Main Page
print("Please enter one number of the next operation you want ")
display:
Enter 1 to login to an existing account
Enter 2 to login as an admin
Enter 3 to continue as a guest
Enter 4 to exit the grocery store
display(enter the number)
get choice
try:
choice=integer
yes
return choice
no
print (please type numbers only

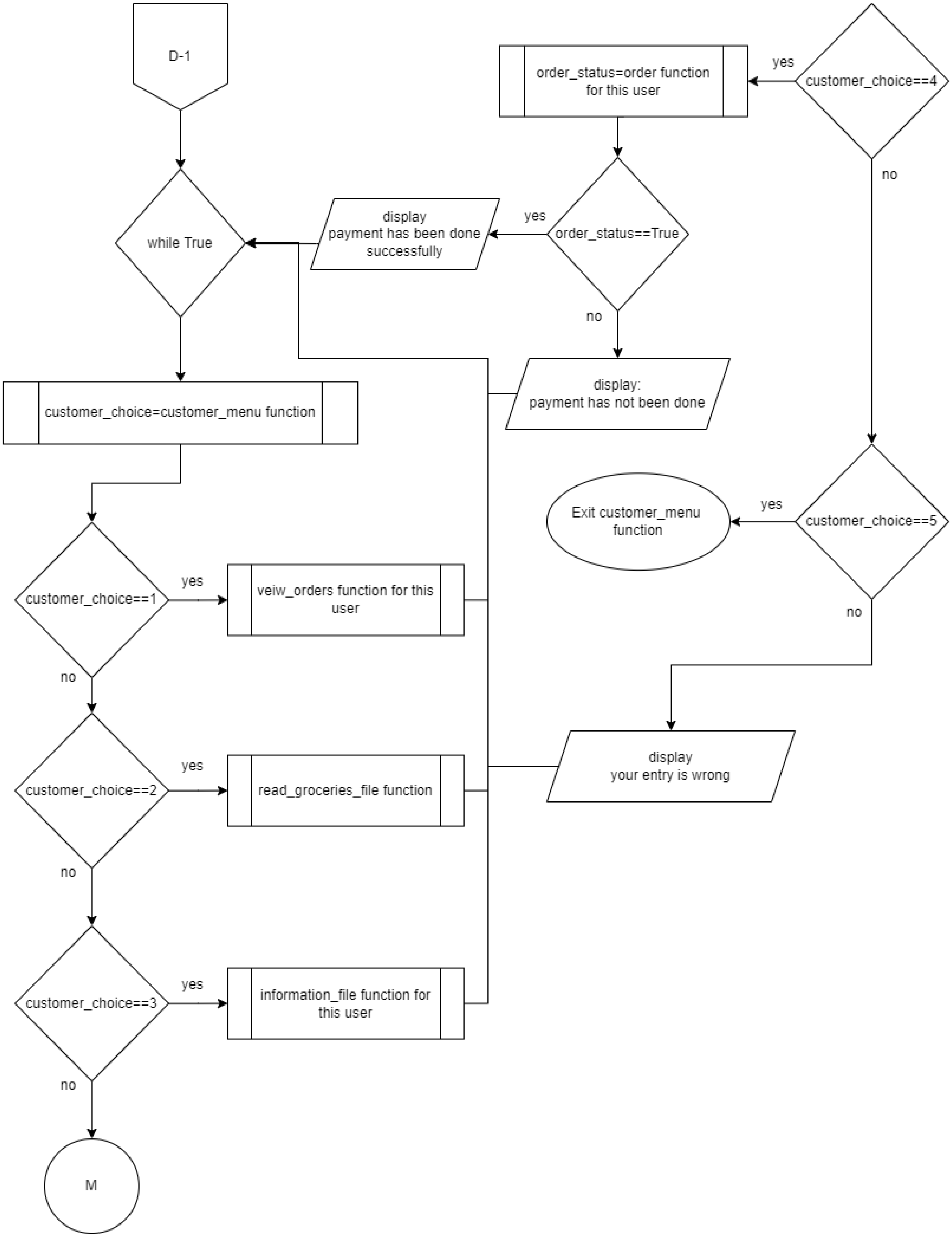

guest_mnue function
display (Guest Portal)
print("Please enter one number of the next operation you want ")
display:
Enter 1 to view Groceries detail
Enter 2 to create an account
Enter 3 to go back to the Main Page
display(enter the number)
get choice
try:
choice=integer
yes
return choice
no
print (please type numbers only

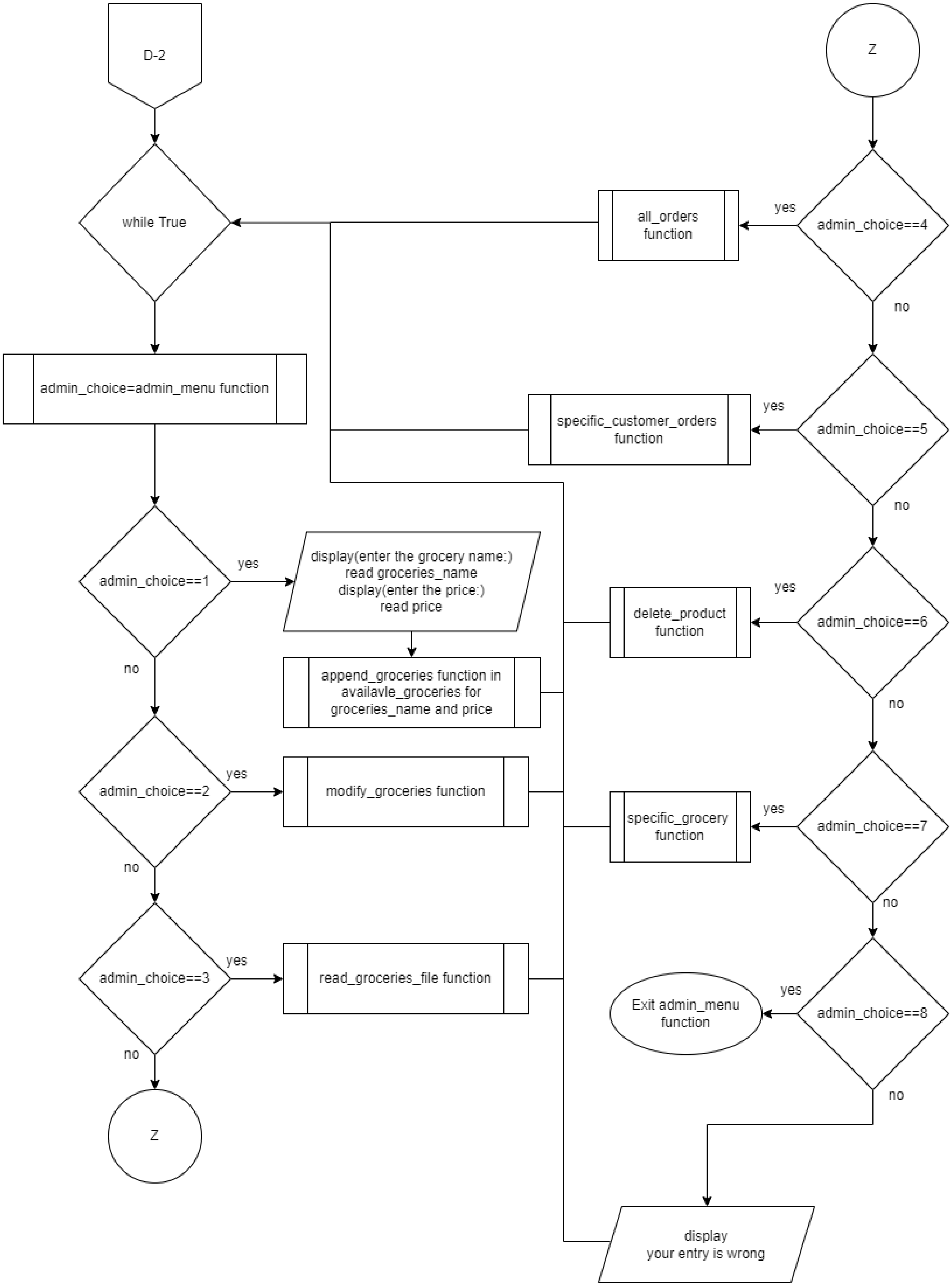

admin_menu function
display(admin portal
display: choose one number of the operation you want:
dislay:
Enter 1 to upload Groceries detail in system
Enter 2 to update/modify Groceries
Enter 3 to view all uploaded Groceries
Enter 4 to view all orders of customers
Enter 5 to search order of specific customer
Enter 6 to delete Groceries information
Enter 7 to go back to the Main Page
display: enter the number
read choice
try:
choice=integer
yes
return choice
no
print: please type numbers only

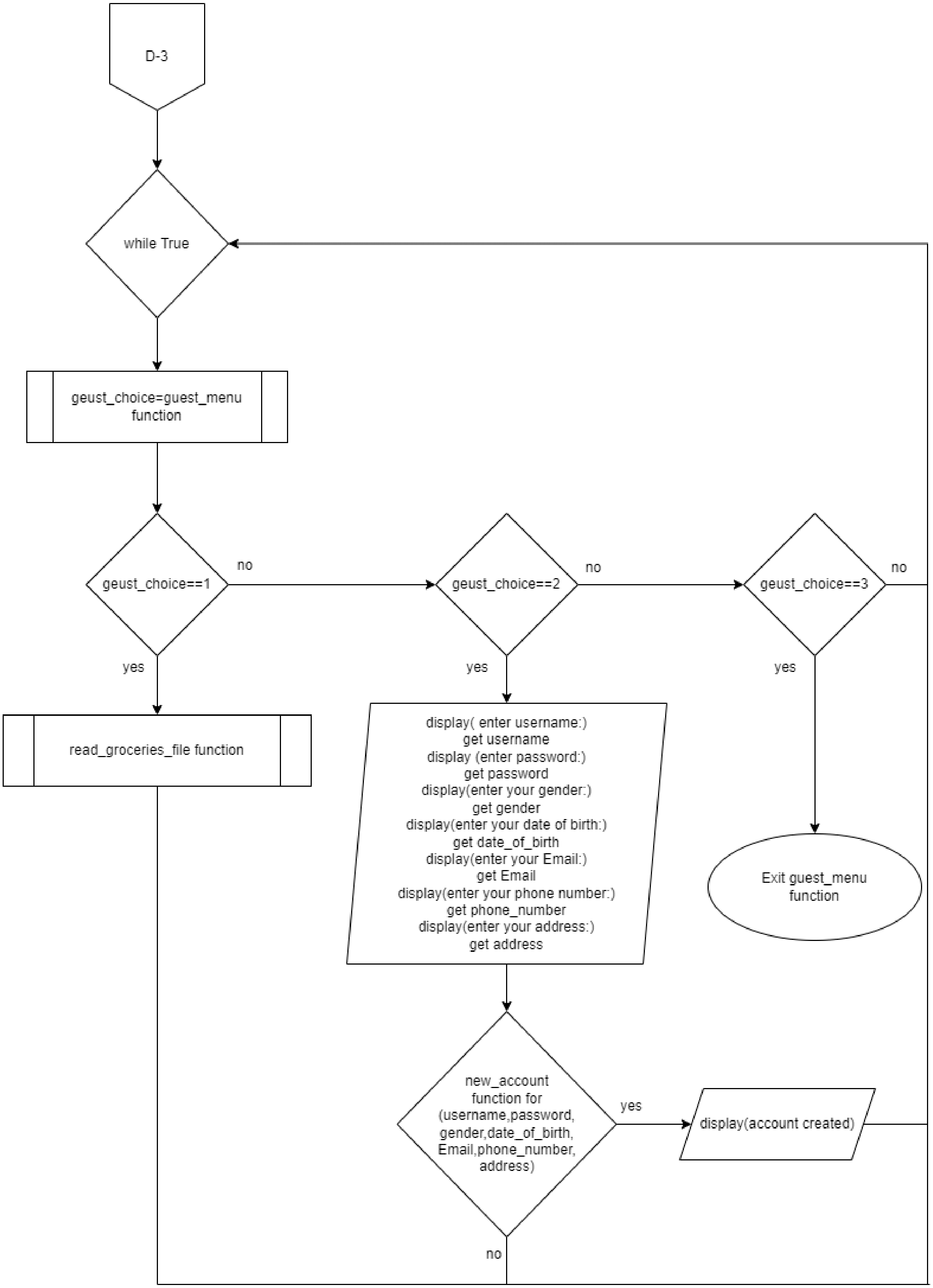

customer_menu function
display (customer page
display: choose one number of the operation you want:
display:
Enter 1 to view own order
Enter 2 to View all Groceries
Enter 3 to view personal information
Enter 4 to Place order of Groceries
Enter 5 to go back to the Main Page
display: enter the number
read choice
try:
choice=integer
yes
return choice
no
print: please type numbers only


welcome function
Disply (welcome to FRESHCO Sdn Bhd Groceries)
End welcome function


main function
welcome function
while True
choice=login_type function
choice==1
yes
display (enter your username:)
get username
display(enter your password)
get password
if login function
(username,password,"user")
True
Yes
D-1
No
L
no
choice==2
yes
display (enter your username:)
get username
display(enter your password)
get password
if login function
(username,password,"user")
True
No
yes
D-2
no
H
L
H
choice==3
yes
D-3
no
choice==4
yes
End main function
no
display: your entry is wrong


D-1
while True
customer_choice=customer_menu function
customer_choice==1
yes
no
veiw_orders function for this user
customer_choice==2
yes
no
read_groceries_file function
customer_choice==3
yes
no
information_file function for this user
M
customer_choice==4
yes
no
order_status=order function for this user
order_status==True
yes
no
display payment has been done successfully
display: payment has not been done
customer_choice==5
yes
no
Exit customer_menu function
display your entry is wrong


D-2
while True
admin_choice=admin_menu function
admin_choice==1
yes
no
display(enter the grocery name:)
read groceries_name
display(enter the price:)
read price
append_groceries function in availavle_groceries for groceries_name and price
admin_choice==2
modify_groceries function
admin_choice==3
read_groceries_file function
Z
Z
admin_choice==4
all_orders function
admin_choice==5
specific_customer_orders function
admin_choice==6
delete_product function
admin_choice==7
specific_grocery function
admin_choice==8
Exit admin_menu function
display
your entry is wrong


D-3
while True
geust_choice=guest_menu function
geust_choice==1
no
yes
read_groceries_file function
geust_choice==2
no
yes
display(enter username:)
get username
display (enter password:)
get password
display(enter your gender:)
get gender
display(enter your date of birth:)
get date_of_birth
display(enter your Email:)
get Email
display(enter your phone number:)
get phone_number
display(enter your address:)
get address
geust_choice==3
no
yes
Exit guest_menu function
new_account function for (username,password, gender,date_of_birth, Email,phone_number, address)
yes
display(account created)
no

Pseudocode

```
Define login (PARAMETER: username, password,log_in_as) function

    1. Start login function
    2. Set Login_file=" "
    3. IF log_in_as == "user" THEN
        3.1. Try:
            3.1.1. Set file handler to open "user.txt" file
        3.2. Except:
            3.2.1. print "no such a file"
    4. ELSE IF log_in_as == "admin"
        4.1. Try:
            4.1.1. Set file handler to open "admin.txt"
        4.2. Except:
            4.2.1. print "No such a file"
    5. ENDIF
    6. Read lines in "user.txt" file and "admin.txt" file
    7. Close file handler
    8. Set Status=True
    9. FOR userData IN "user.txt" file and "admin.txt" file
        9.1. User = userData line split to list using (,) [first content]
        9.2. pwd = userData line split to list using (,) [second content] using STRIP function
            8.3.1. IF (user == username) and (pwd == password) THEN
                8.3.1.1. Return True
            8.3.2. ELSE
                8.3.2.1.Status=False
            8.3.3. ENDIF
            8.3.4. IF Status==False THEN
                8.3.4.1.Print "Username or Password is wrong"
            8.3.5. ENDIF
    9. ENDFOR
```

10. Close file handler
11. END login function

Define new_account (PARAMETER: username, password, gender, date_of_birth, Email, phone_number, address) function

```
1. Start new_account function
2. Set Data=username+","+password+","+gender+","+date_of_birth+","+Email+
                  ","+phone_number+","+address+"\n"
3. Set file handler to open “user.txt” file for APPENDING
4. Write into “user.txt” file (Data)
5. Close file handler
6. Return True
7. END new_account function
```

Define groceries_file( ) function

```
1. Start groceries_file function
2. Set Groceries as list to save into it
3. Set file handler to open “groceries.txt” file
4. FOR available IN “groceries.txt”
   4.1. Read lines in “groceries.txt” file
   4.2. Set available=split to list using (“,”)
   4.3. For groceries in available list [first content]=convert to for groceries integer in
        available list [second content]
   4.4.Close file handler
   4.5.Return groceries
5. END groceries_file function
```

```
Define information_file (PAREMETR:username)

    1.  Start information_file function
    2.  Set file handler to open “user.txt” file for reading
    3.  FOR i IN “user.txt” file
        3.1. Read lines in “user.txt” file
        3.2. Set i=split i to list using(“,”)
        3.3. IF (username= i list [first content] THEN
            3.3.1.   Set customer_information as an empty LIST
            3.3.2.   Set count=0
        3.4.ENDIF
            3.4.1.   DOWHILE (count<string length in i LIST)
                3.4.1.1. Append into “user.txt” file i LIST [count]
                3.4.1.2. Calculate count=count+1
            3.4.2.   ENDDO
    4.  ENDFOR
    5.  Display your username is: [ the first content in customer_information list]
    6.  Display your password is: [ the second content in customer_information list]
    7.  Display your gender is: [ the third content in customer_information list]
    8.  Display your birthday is: [ the fourth content in customer_information list]
    9.  Display your Email is: [ the fifth content in customer_information list]
    10. Display your phone number is: [ the sixth content in customer_information list]
    11. Display your address is: [ the seventh content in customer_information list]
    12. END information_file function
```

Define append_groceries (PAREMETER: available_groceries,groceries_name,price) function

```
1. Start append_groceries function
2. Read_groceries_file function
3. Set file handle to open “groceries.txt” file for APPENDING
4. Write into “groceries.txt” file
5. Close file handler
6. Display (this product was added successfully)
7. END append_groceries function
```

Define modofy_groceries( ) function

```
1. Start modify_groceries function
2. Read_groceries_file function
3. Display (enter 1 if you want to change the name, enter 2 if you want to change the cost:)
4. Get operation_num
5. Display (enter the grocery number you want to modify:)
6. Get groceries_num
7. Set new_file=””
8. IF operation_num==”1” THEN
   8.1.Display (type the new amount:)
   8.2.Get new_name
   8.3.Set count=1
   8.4.Set file1 handler to open “groceries.txt” file for READING
   8.5.FOR i IN modify_grocereis
       8.5.1. IF string(count)==groceries_num THEN
           8.5.1.1. Set price=split i to list using (“,”)[the second content]
           8.5.1.2.Set new_file=new_file+(new_name,price)
           8.5.1.3.Calculate count=count+1
           8.5.1.4.Display (the task has been done successfully)
       8.5.2. ELSE
           8.5.2.1. Set new_file=new_file+i
           8.5.2.2. Calculate count=count+1
       8.5.3. ENDIF
   8.6.Set file2 handler to open “groceries.txt” file for WRITING
   8.7.Write into “groceries.txt” file (new_file)
   8.8.Close file2 handler
   8.9.Close file1 handler
   8.10.     ENDFOR
9. ELSEIF operation_num==”2”
   9.1.Status =0
```

```
    9.2.DOWHILE status==0
        9.2.1.   Try:
            9.2.1.1.Display (type the new price:)
            9.2.1.2.Get new_price as INTEGER only
            9.2.1.3.Set status=1
        9.2.2.   Except:
            9.2.2.1.Display (invalid price)
    9.3.ENDDO
    9.4.Set count=1
    9.5.Set file3 handler to open “groceries.txt” file for READING
    9.6. FOR i IN “groceries.txt” file
        9.6.1.   IF String(count)==”groceries.txt” file THEN
            9.6.1.1.Set name= split i to list using (“,”)[the first content]
            9.6.1.2.Set new_file=new_file+(name,new_price)
            9.6.1.3.Calculate count=count+1
            9.6.1.4.Display (the task has been done successfully)
        9.6.2.   ELSE
            9.6.2.1.Set new_file=new_file+i
            9.6.2.2.Calculate count=count+1
        9.6.3.   ENDIF
    9.7.Set file4 handler to open “groceries.txt” for WRITING
    9.8.Write into “groceries.txt” file (new_file)
    9.9.Close file4 handler
    9.10.        Close file3 handler
    9.11.        ENDFOR
10. END modify_grocereis( ) function
```

Define order(PAREMETER: username) function

```
1. Start order function
2. Set repeat=0
3. Set total=0
4. Set all_orders to save into it
5. DOWHILE repeat==0
   5.1.read_groceries_file( ) function
   5.2.Display (enter the grocery’s number:)
   5.3.Get groceries_number
   5.4.Set file1 handler to open “groceries.txt” file for READING
   5.5.Set count=1
   5.6.FOR i IN  “groceries.txt” file
       5.6.1. Read lines in “groceries.txt” file
       5.6.2. Set i=split i to list using(“,”)
       5.6.3. IF string(count)==groceries_nmber THEN
           5.6.3.1.Set all_orders=all_orders+(username,
 i list[first content], i list[second content])
           5.6.3.2.Display (your order is: (i list[first content]) for (i list[second content]
                   Ringgit)
           5.6.3.3.Calculate total=total+INTEGER(i list[second content])
           5.6.3.4.Display (enter 1 if you want to continue, enter 2 if you want to proceed
                   to payment:)
           5.6.3.5.Get input
           5.6.3.6.IF input==”1” THEN
               5.6.3.6.1. Break
           5.6.3.7.ELSE
               5.6.3.7.1. Display (the total of  your orders {total} RM (the payment is in
                          cash only) Do you want to proceed? Enter 1 to proceed, enter 2
                          to cancel payment)
           5.6.3.8.ENDIF
               5.6.3.8.1. Get payment
```

```
                5.6.3.8.2.   IF payment==”1” THEN
                    5.6.3.8.2.1.Set file2 handler to open “orders.txt.” file for APPENDING
                    5.6.3.8.2.2.Write into “orders.txt” file (all_orders)
                    5.6.3.8.2.3.Return True
                5.6.3.8.3.   ELSE
                    5.6.3.8.3.1.Return False
                5.6.3.8.4.   ENDIF
        5.6.4.   ELSE
            5.6.4.1.Calculate count=count+1
        5.6.5.   ENDIF
    5.7.ENDFOR
6.   Close file1 handler
7.   END order function
```

Define view_orders(PAREMETE: username) function

```
1. Start view_orders function
2. Set total=0
3. Set file handler to open “orders.txt” file for READING
4. FOR i IN “orders.txt” file
   4.1.Read lines in “orders.txt” file
   4.2.Set i=split i to list using (“,”)
   4.3.IF username == i list[first content] THEN
       4.3.1. Display (you ordered {i list[second content]} for [i list[third content]} RM)
       4.3.2. Calculate total=total+INTEGER(i list[third content]}
   4.4.ENDIF
5. ENDFOR
6. Display (the total of your orders  {total} RM)
7. Close file handler
8. END view_orders function
```

Define read_file_groceries( ) fnction

```
1. Start read_groceries_file function
2. Set available_groceries as empty list
3. Set file handler to open “groceries.txt” file
4. Set count=1
5. Display(‘\n’)
6. Display(product name:      cost:)
7. FOR available IN “groceries.txt” file
   7.1.Set available=split available to list using (“,”)
   7.2.Display({count} {available list[first content]})
   7.3.Display({available list[second content]})
   7.4.Calculate count=count+1
   7.5.Close file handler
   7.6.Return available_groceries
8. ENDFOR
9. END groceries_file function
```

Define specific_customer_orders( ) function

```
1. Start specific_customer_orders function
2. Set total=0
3. Display (enter the customer name:)
4. get customer_name
5. Set file handler to open “orders.txt” file for READING
6. FOR i IN “orders.txt” file
   6.1.Read lines in “orders.txt” file
   6.2.Set i=split i to list using (“,”)
   6.3.IF customer_name == i list[first content] THEN
       6.3.1. Display (the order for {customer_name} [ {i list[second content]} ] for {i list[third content]} RM
       6.3.2. Calculate total=total+INTEGER(i list[third content]}
   6.4.ENDIF
7. ENDFOR
8. Display (the total cost of {customer_name} is {total} RM)
9. Close file handler
10. END specific_customer_orders
```

Define specific_grocery( )function

```
1. Start specific_groceries function
2. Display (Enter the grocery name:)
3. Get grocery_name
4. Set file handler to open “groceries.txt” file for reading
5. FOR i IN “groceries.txt” file
   5.5.Set i=split i into list uing (“,”)
   5.6.IF grocery_name==i list[first content] THEN
       5.6.1. Display (grocery name is: {i list[first content]})
       5.6.2. Display( the grocery cost is {i list[second content]})
   5.7.Close handler file
6. END specific_grocery function
```

Define delete_product( ) function

```
1. Start delete_product function
2. Set repeat=1
3. DOWHILE repeat==1
   3.1.Set new_file to save into it
   3.2.read_groceries_file( ) function
   3.3.Display (enter the grocery’s number:)
   3.4.Get groceries_number
   3.5.Set file1 handler to open “groceries.txt” file for READING
   3.6.Set count=1
   3.7.Set status=False
   3.8.FOR i IN “groceries.txt” file
      3.8.1. Read lines in “groceries.txt” file
      3.8.2. IF string(count)!=groceries_number THEN
         3.8.2.1.Set new_file=new_file+i
         3.8.2.2.Calculate count=count+1
      3.8.3. ELSE
         3.8.3.1.Display(the product was deleted)
         3.8.3.2.Set statues=True
         3.8.3.3.Calculate count=count+1
      3.8.4. ENDIF
   3.9.ENDFOR
   3.10. IF statues==False THEN
      3.10.1. Display (your entry is wrong)
   3.11. ENDIF
   3.12. Close file1 handler
   3.13. Set file2 handler to open “groceries.txt” file for READING
   3.14. Write into “groceries.txt” file (new_file)
   3.15. Close file2 handler
   3.16. Display (Enter 1 to continue,  Enter 2 to exit)
   3.17. Get input
```

```
3.18.   IF input==”2” THEN
    3.18.1. Return True
3.19.   ENDIF
3.20.   END delete_product function
```

Define all_orders( ) function

```
1. Start all_orders function
2. Set file handler to open”orders.txt” file for READING
3. FOR i IN “orders.txt” file
   3.1.Read lines in “orders.txt” file
   3.2.Set i= split i to list using(“,”)
   3.3.Display (the customer {i list[first content]} orderd{i list[second content]})
4. Close file handler
5. END all_orders function
```

Define login_type( ) function

```
1. Start login_type function
2. Display (Main Page)
3. Display (Please enter one number of the next operation you want)
4. Display (Enter 1 to login to an existing account
                Enter 2 to login as an admin
                Enter 3 to continue as a guest
                Enter 4 to exit the grocery store)
5. Try:
   5.1.Display (Enter the number :)
   5.2.Read Choice (INTEGER only)
   5.3.Return choice
6. Except:
   6.1.Display(please type numbers only)
7. END login_type function
```

Define guest_menu( ) function

```
1. Start guest_menu function
2. Display (Guest Portal)
3. Display (Please enter one number of the next operation you want)
4. Display (Enter 1 to view Groceries detail
            Enter 2 to create an account
            Enter 3 to go back to the Main Page)
5. TRY:
   5.1.Display (Enter the number:)
   5.2.Read choice (INTEGER only)
   5.3.Return choice
6. EXCEPT:
   6.1.Display (please type numbers only)
7. END guest_menu function
```

Define admin_menu ( ) function

```
1.  Start admin_menu function
2.  Display (Admin Portal)
3.  Display (Please enter one number of the next operation you want)
4.  Display (Enter 1 to upload Groceries detail IN system
 Enter 2 to update/modify Groceries
 Enter 3 to view all uploaded Groceries
 Enter 4 to view all orders of customers
 Enter 5 to search order of specific customer
        Enter 6 to delete Groceries information
        Enter 7 to go back to the Main Page)
5.  TRY:
    5.1.Display (Enter the number:)
    5.2.Read choice (INTEGER only)
    5.3.Return choice
6.  EXCEPT:
    6.1.Display (please type numbers only)
7.  END admin_menu function
```

Define customer_menu ( ) function

```
1. Start customer_menu function
2. Display (Customer Portal)
3. Display (Please enter one number of the next operation you want)
4. Display (Enter 1 to view own order

            Enter 2 to View all Groceries

            Enter 3 to view personal information

            Enter 4 to Place order of Groceries

            Enter 5 to go back to the Main Page)

5. TRY:
    5.1.Display (Enter the number:)
    5.2.Read choice (INTEGER only)
    5.3.Return choice
6. EXCEPT:
    6.1.Display (please type numbers only)
7. END customer_menu function
```

Define welcome ( ) function

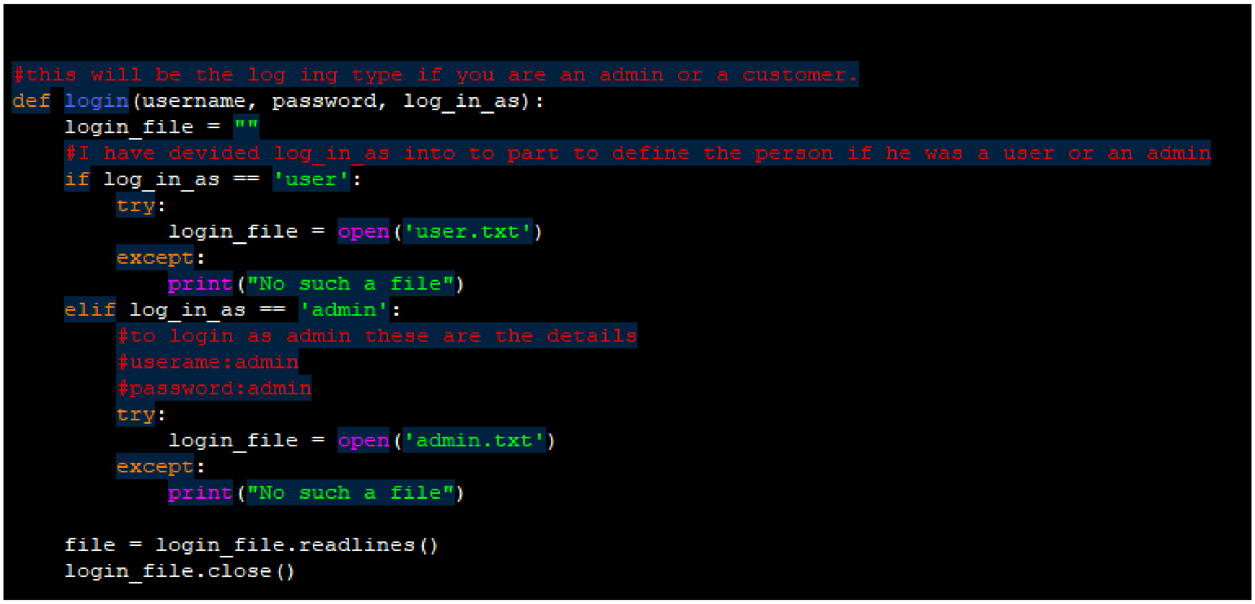
1. Start welcome Function
2. Display (welcome to FRESHCO Sdn Bhd Groceries)
3. END welcome function

Define main ( ) function

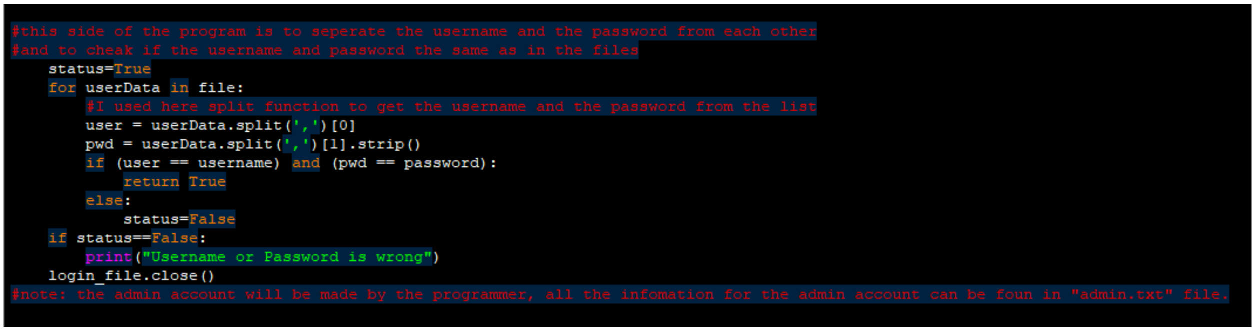
```
1.  Start main function
2.  welcome function
3.  DOWHILE True
    3.1.Set choice=login_type function
    3.2.IF choice==1 THEN
        3.2.1.  Display(enter your username)
        3.2.2.  Get username
        3.2.3.  Display (enter your password)
        3.2.4.  Get password
        3.2.5.  IF login(PAREMETER:username,password,"user") function THEN
            3.2.5.1.DOWHILE True
                3.2.5.1.1.  Set customer_choice=customer_menu function
                3.2.5.1.2.  IF customer_choice==1 THEN
                    3.2.5.1.2.1.View_order(PAREMETER: username) function
                3.2.5.1.3.  ELSEIF customer_choice==2
                3.2.5.1.4.  Read_groceries_file( ) function
                3.2.5.1.5.  ELSEIF customer_choice==3
                    3.2.5.1.5.1.Information_file(PAREMETER: username) function
                3.2.5.1.6.  ELSEIF customer_choice==4
                    3.2.5.1.6.1.Set order_status=order(PAREMETER:username) function
                    3.2.5.1.6.2.IF order_status==True THEN
                        3.2.5.1.6.2.1.  Display(payment has been done successfully)
                    3.2.5.1.6.3.ELSE
                        3.2.5.1.6.3.1.  Display(payment has not been done)
                    3.2.5.1.6.4.ENDIF
                3.2.5.1.7.  ELSEIF customer_choice==5
                    3.2.5.1.7.1.Break
                3.2.5.1.8.  ELSE
                    3.2.5.1.8.1.Display(your entry is wrong)
                    3.2.5.1.8.2.Continue
            3.2.5.1.9.   ENDIF
        3.2.6.   ELSE
            3.2.6.1.Continue
            3.2.6.2.ENDDO
    3.3.ELSEIF choice==2
        3.3.1.   Display (enter your username)
        3.3.2.   Get username
        3.3.3.   Display(enter your password)
        3.3.4.   Get password
        3.3.5.   IF login(PAREMETER:username,password,"admin") function THEN
            3.3.5.1.DOWHILE True
                3.3.5.1.1.   Set admin=admin_menu() function
                3.3.5.1.2.   IF admin_choice==1 THEN
                    3.3.5.1.2.1.Display (enter the grocery name:)
                    3.3.5.1.2.2.Get groceries_name
                    3.3.5.1.2.3.Display (enter the price:)
                    3.3.5.1.2.4.Get price
                    3.3.5.1.2.5.Append_groceries(PAREMETER:available_groceries) function
                3.3.5.1.3.   ELSEIF admin_choice==2
                    3.3.5.1.3.1.modify_groceries() function
                3.3.5.1.4.   ELSEIF admin_choice==3
                3.3.5.1.5.   Read_groceries_file( ) function
                3.3.5.1.6.   ELSEIF admin_choice==4
                    3.3.5.1.6.1.all_orders() function
                3.3.5.1.7.   ELSEIF admin_choice==5
                    3.3.5.1.7.1.specific_customer_orders() function
                3.3.5.1.8.   ELSEIF admin_choice==6
                    3.3.5.1.8.1.delete_product() function
                3.3.5.1.9.   ELSEIF admin_choice==7
                    3.3.5.1.9.1.Break
                3.3.5.1.10. ELSE
                    3.3.5.1.10.1.    Display(your entry is wrong)
                    3.3.5.1.10.2.    Continue
                3.3.5.1.11. ENDIF
            3.3.5.2.ENDDO
        3.3.6.   ELSE
            3.3.6.1.Continue
        3.3.7.   ENDIF
    3.4.ELSEIF choice==3
        3.4.1.   DOWHILE True
            3.4.1.1.Set geust_choice=guest_menu() function
            3.4.1.2.IF geust_choice==1 THEN
                3.4.1.2.1.   read_groceries_file( ) function
            3.4.1.3.ELSEIF geust_choice==2
                3.4.1.3.1.   Display (enter username:)
                3.4.1.3.2.   Get username
                3.4.1.3.3.   Display (enter password:)
                3.4.1.3.4.   Get password
                3.4.1.3.5.   Display(enter your gender:)
                3.4.1.3.6.   Get gender
                3.4.1.3.7.   Display (enter your date of birth:)
                3.4.1.3.8.   Get date_of_birth
                3.4.1.3.9.   Display(enter your Email)
                3.4.1.3.10. Get Email
                3.4.1.3.11. Display(enter your phone number:)
                3.4.1.3.12. Get phone_number
                3.4.1.3.13. Display(enter your address:)
                3.4.1.3.14. Get address
                3.4.1.3.15. IF new_account(PAREMETER:username,password,gender, date_of_birth,Email,phone_number,address) function THEN
                    3.4.1.3.15.1.    Display (Account created)
                3.4.1.3.16. ENDIF
            3.4.1.4.ELSEIF geust_choice==3
                3.4.1.4.1.   Break
            3.4.1.5.ELSE
                3.4.1.5.1.   Display(your entry is wrong)
                3.4.1.5.2.   Continue
            3.4.1.6.ENDIF
        3.4.2.   ENDDO
    3.5.ELSEIF choice==4
        3.5.1.   Break
    3.6.ELSE
        3.6.1.   Display(your entry is wrong)
        3.6.2.   Continue
    3.7.ENDIF
    4.   ENDDO
    5.   END main( ) function
```

Program source code and explanation

Login function

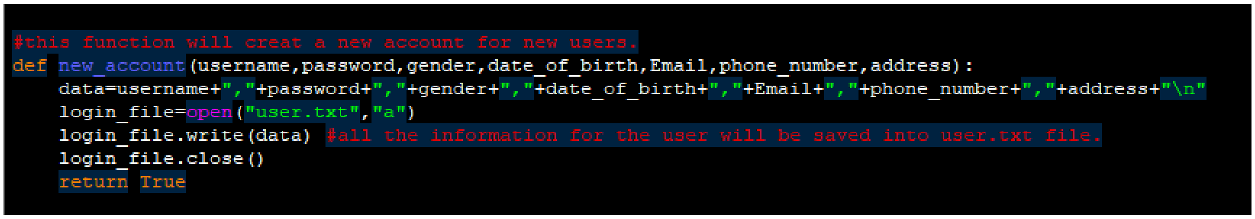


```
#this will be the log ing type if you are an admin or a customer.
def login(username, password, log_in_as):
    login_file = ""
    #I have devided log_in_as into to part to define the person if he was a user or an admin
    if log_in_as == 'user':
        try:
            login_file = open('user.txt')
        except:
            print("No such a file")
    elif log_in_as == 'admin':
        #to login as admin these are the details
        #userame:admin
        #password:admin
        try:
            login_file = open('admin.txt')
        except:
            print("No such a file")

    file = login_file.readlines()
    login_file.close()
```



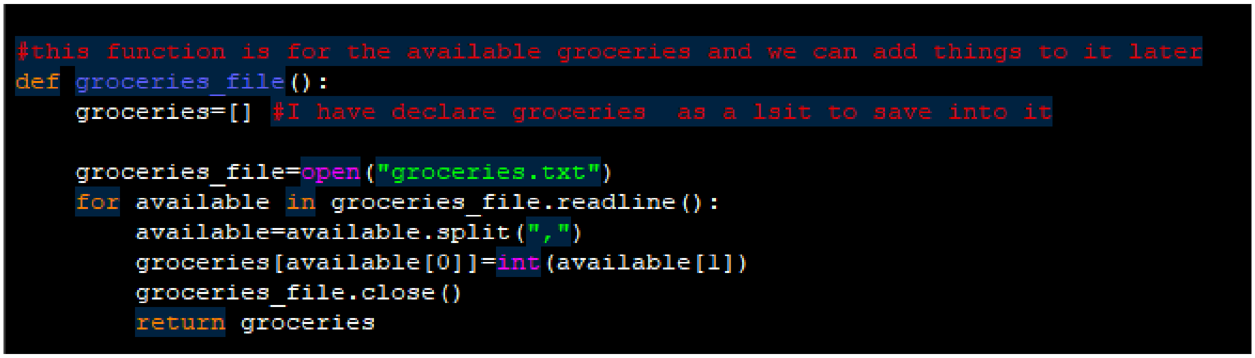
```
#this side of the program is to seperate the username and the password from each other
#and to cheak if the username and password the same as in the files
    status=True
    for userData in file:
        #I used here split function to get the username and the password from the list
        user = userData.split(',')[0]
        pwd = userData.split(',')[1].strip()
        if (user == username) and (pwd == password):
            return True
        else:
            status=False
    if status==False:
        print("Username or Password is wrong")
    login_file.close()
#note: the admin account will be made by the programmer, all the infomation for the admin account can be foun in "admin.txt" file.
```

New account function



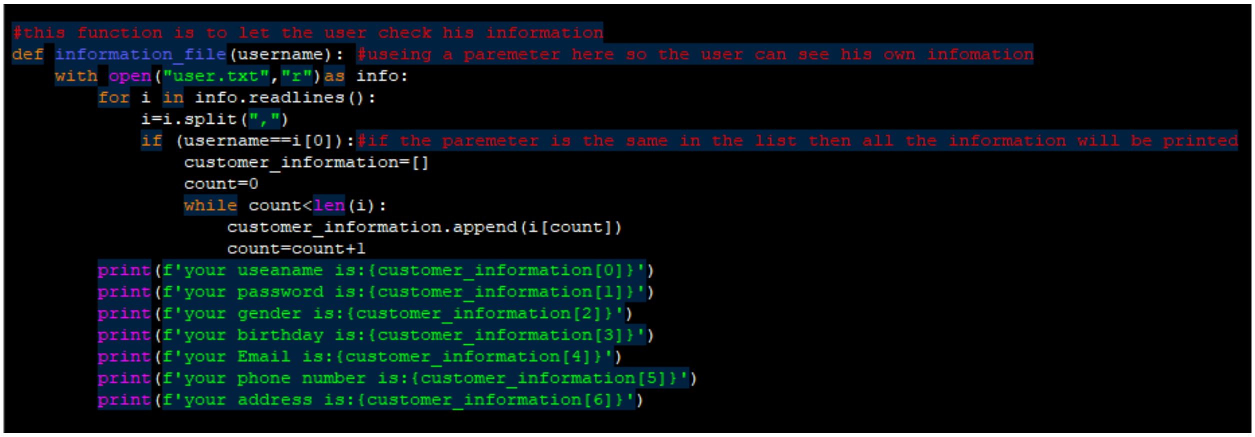
```
#this function will creat a new account for new users.
def new_account(username,password,gender,date_of_birth,Email,phone_number,address):
    data=username+","+password+","+gender+","+date_of_birth+","+Email+","+phone_number+","+address+"\n"
    login_file=open("user.txt","a")
    login_file.write(data) #all the information for the user will be saved into user.txt file.
    login_file.close()
    return True
```

Groceries file function



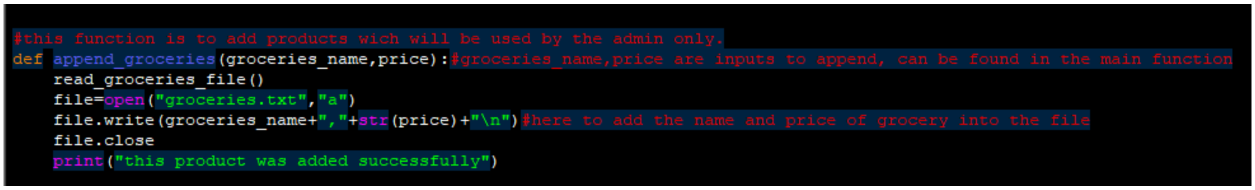
```
#this function is for the available groceries and we can add things to it later
def groceries_file():
    groceries=[] #I have declare groceries  as a lsit to save into it

    groceries_file=open("groceries.txt")
    for available in groceries_file.readline():
        available=available.split(",")
        groceries[available[0]]=int(available[1])
        groceries_file.close()
        return groceries
```

Information file function



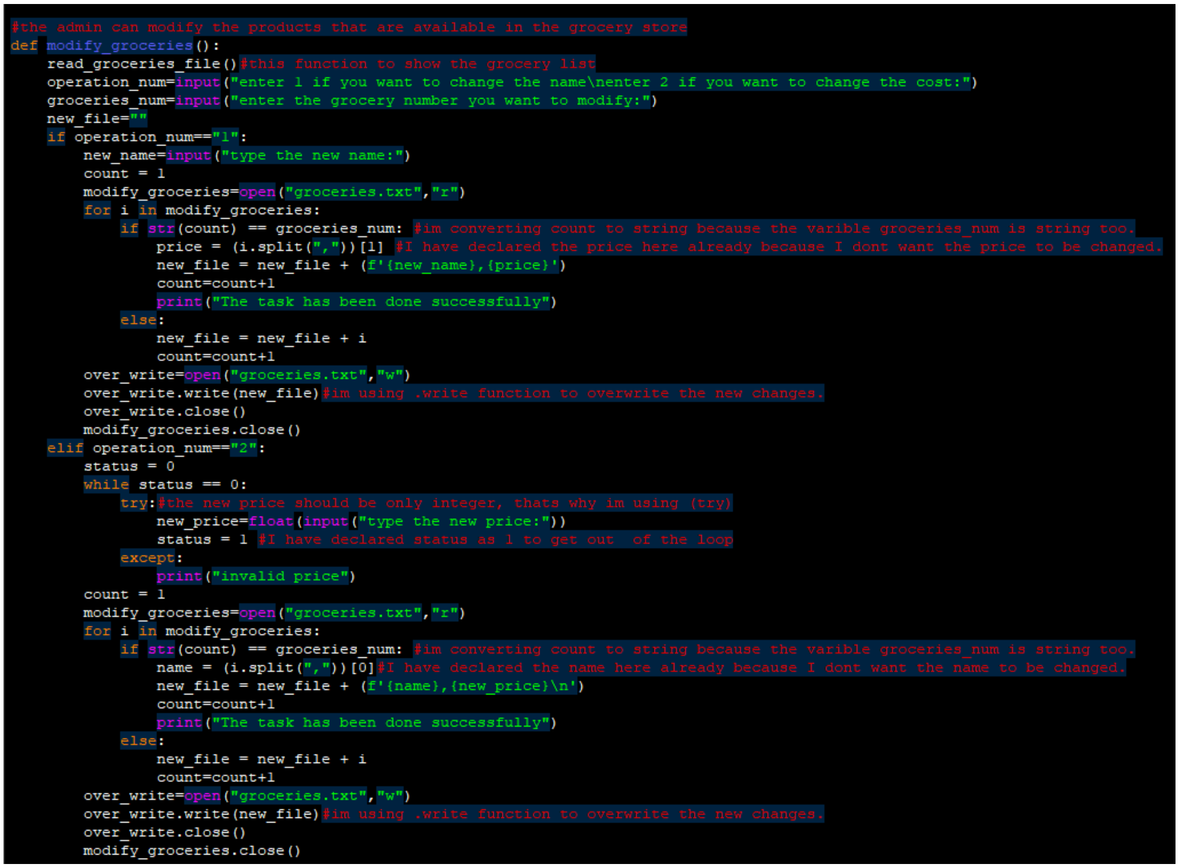
```
#this function is to let the user check his information
def information_file(username): #useing a paremeter here so the user can see his own infomation
    with open("user.txt","r")as info:
        for i in info.readlines():
            i=i.split(",")
            if (username==i[0]):#if the paremeter is the same in the list then all the information will be printed
                customer_information=[]
                count=0
                while count<len(i):
                    customer_information.append(i[count])
                    count=count+1
        print(f'your useaname is:{customer_information[0]}')
        print(f'your password is:{customer_information[1]}')
        print(f'your gender is:{customer_information[2]}')
        print(f'your birthday is:{customer_information[3]}')
        print(f'your Email is:{customer_information[4]}')
        print(f'your phone number is:{customer_information[5]}')
        print(f'your address is:{customer_information[6]}')
```

Append grocery’s function



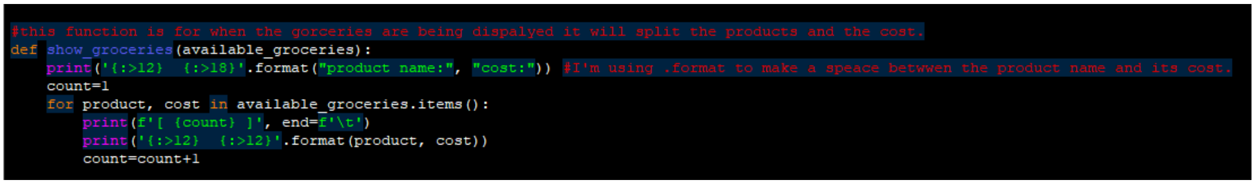
```
#this function is to add products wich will be used by the admin only.
def append_groceries(groceries_name,price):#groceries_name,price are inputs to append, can be found in the main function
    read_groceries_file()
    file=open("groceries.txt","a")
    file.write(groceries_name+","+str(price)+"\n")#here to add the name and price of grocery into the file
    file.close
    print("this product was added successfully")
```

Modify groceries



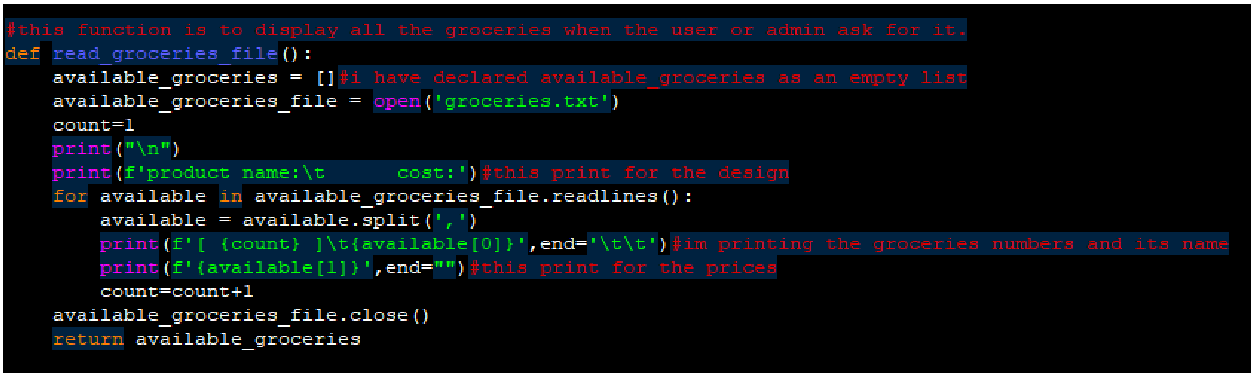
```
#the admin can modify the products that are available in the grocery store
def modify_groceries():
    read_groceries_file()#this function to show the grocery list
    operation_num=input("enter 1 if you want to change the name\nenter 2 if you want to change the cost:")
    groceries_num=input("enter the grocery number you want to modify:")
    new_file=""
    if operation_num=="1":
        new_name=input("type the new name:")
        count = 1
        modify_groceries=open("groceries.txt","r")
        for i in modify_groceries:
            if str(count) == groceries_num: #im converting count to string because the varible groceries_num is string too.
                price = (i.split(","))[1] #I have declared the price here already because I dont want the price to be changed.
                new_file = new_file + (f'{new_name},{price}')
                count=count+1
                print("The task has been done successfully")
            else:
                new_file = new_file + i
                count=count+1
        over_write=open("groceries.txt","w")
        over_write.write(new_file)#im using .write function to overwrite the new changes.
        over_write.close()
        modify_groceries.close()
    elif operation_num=="2":
        status = 0
        while status == 0:
            try:#the new price should be only integer, thats why im using (try)
                new_price=float(input("type the new price:"))
                status = 1 #I have declared status as 1 to get out  of the loop
            except:
                print("invalid price")
        count = 1
        modify_groceries=open("groceries.txt","r")
        for i in modify_groceries:
            if str(count) == groceries_num: #im converting count to string because the varible groceries_num is string too.
                name = (i.split(","))[0]#I have declared the name here already because I dont want the name to be changed.
                new_file = new_file + (f'{name},{new_price}\n')
                count=count+1
                print("The task has been done successfully")
            else:
                new_file = new_file + i
                count=count+1
        over_write=open("groceries.txt","w")
        over_write.write(new_file)#im using .write function to overwrite the new changes.
        over_write.close()
        modify_groceries.close()
```

Show groceries function



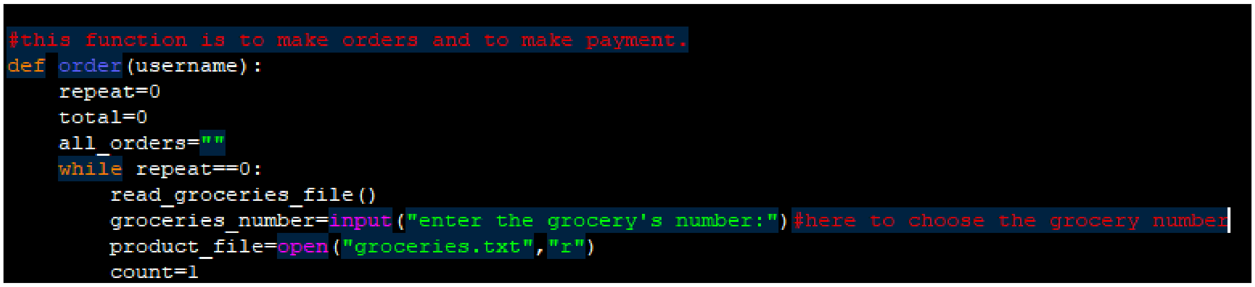
```
#this function is for when the gorceries are being dispalyed it will split the products and the cost.
def show_groceries(available_groceries):
    print('{:>12}  {:>18}'.format("product name:", "cost:")) #I'm using .format to make a speace betwwen the product name and its cost.
    count=1
    for product, cost in available_groceries.items():
        print(f'[ {count} ]', end=f'\t')
        print('{:>12}  {:>12}'.format(product, cost))
        count=count+1
```

Read groceries file function

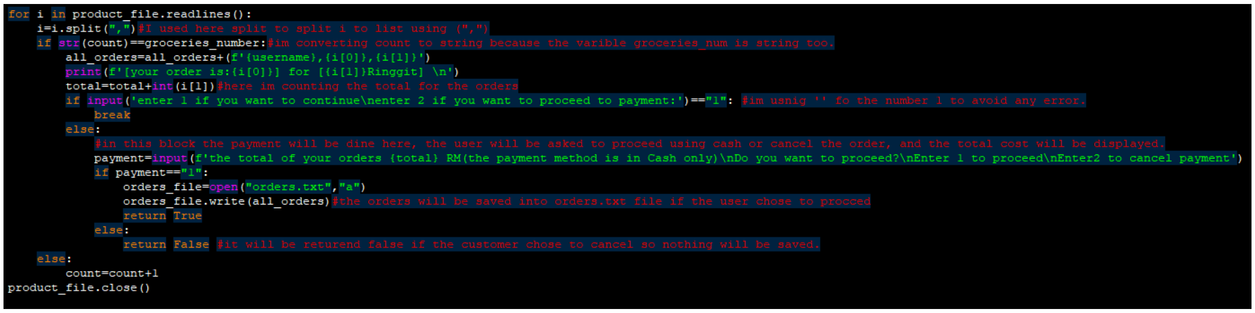


```
#this function is to display all the groceries when the user or admin ask for it.
def read_groceries_file():
    available_groceries = []#i have declared available_groceries as an empty list
    available_groceries_file = open('groceries.txt')
    count=1
    print("\n")
    print(f'product name:\t       cost:')#this print for the design
    for available in available_groceries_file.readlines():
        available = available.split(',')
        print(f'[ {count} ]\t{available[0]}',end='\t\t')#im printing the groceries numbers and its name
        print(f'{available[1]}',end="")#this print for the prices
        count=count+1
    available_groceries_file.close()
    return available_groceries
```

Order function

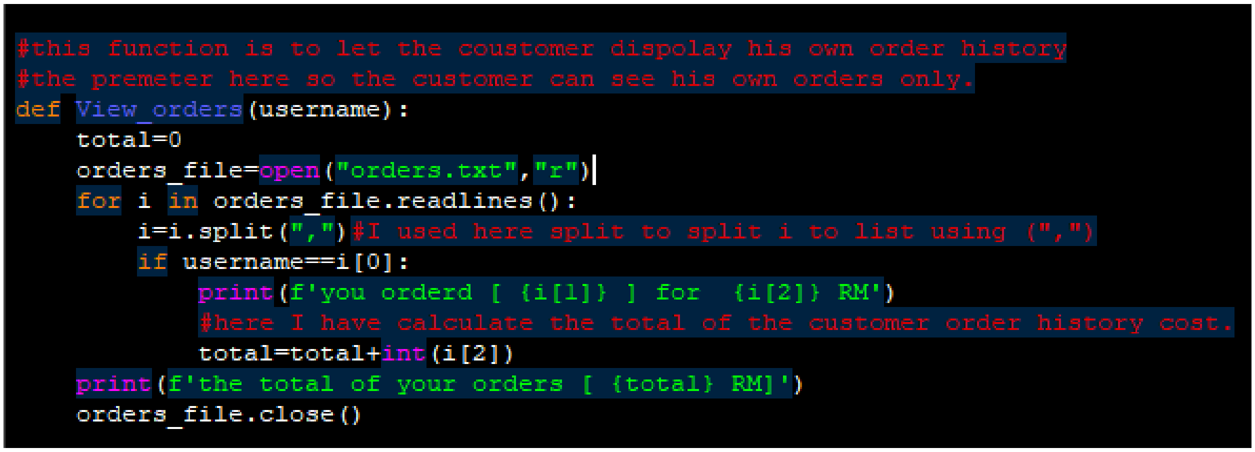


```
#this function is to make orders and to make payment.
def order(username):
    repeat=0
    total=0
    all_orders=""
    while repeat==0:
        read_groceries_file()
        groceries_number=input("enter the grocery's number:")#here to choose the grocery number
        product_file=open("groceries.txt","r")
        count=1
```



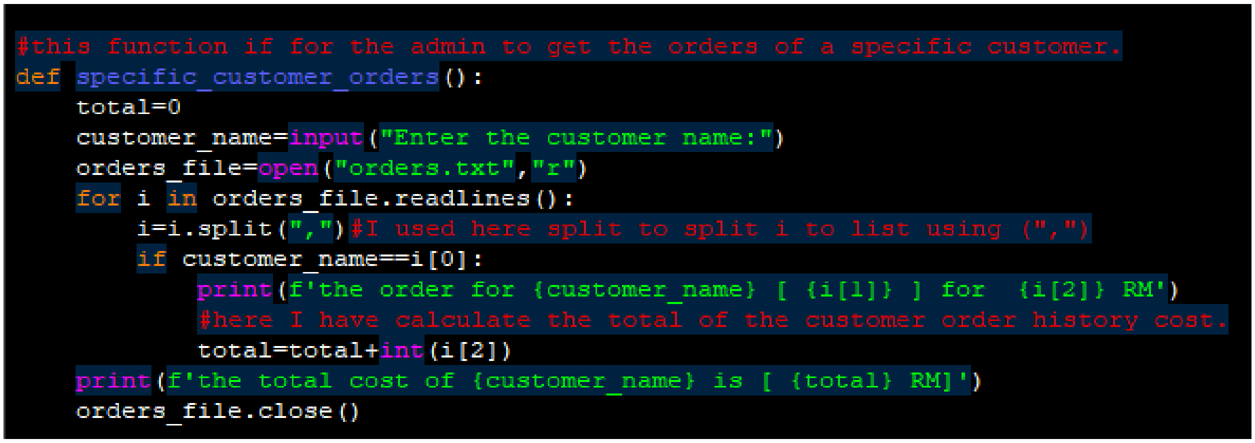
```
for i in product_file.readlines():
    i=i.split(",")#I used here split to split i to list using (",")
    if str(count)==groceries_number:#im converting count to string because the varible groceries_num is string too.
        all_orders=all_orders+(f'{username},{i[0]},{i[1]}')
        print(f'[your order is:{i[0]}] for [{i[1]}Ringgit] \n')
        total=total+int(i[1])#here im counting the total for the orders
        if input('enter 1 if you want to continue\nenter 2 if you want to proceed to payment:')=="1": #im usnig '' fo the number 1 to avoid any error.
            break
        else:
            #in this block the payment will be dine here, the user will be asked to proceed using cash or cancel the order, and the total cost will be displayed.
            payment=input(f'the total of your orders {total} RM(the payment method is in Cash only)\nDo you want to proceed?\nEnter 1 to proceed\nEnter2 to cancel payment')
            if payment=="1":
                orders_file=open("orders.txt","a")
                orders_file.write(all_orders)#the orders will be saved into orders.txt file if the user chose to procced
                return True
            else:
                return False #it will be returend false if the customer chose to cancel so nothing will be saved.
    else:
        count=count+1
product_file.close()
```

View orders function



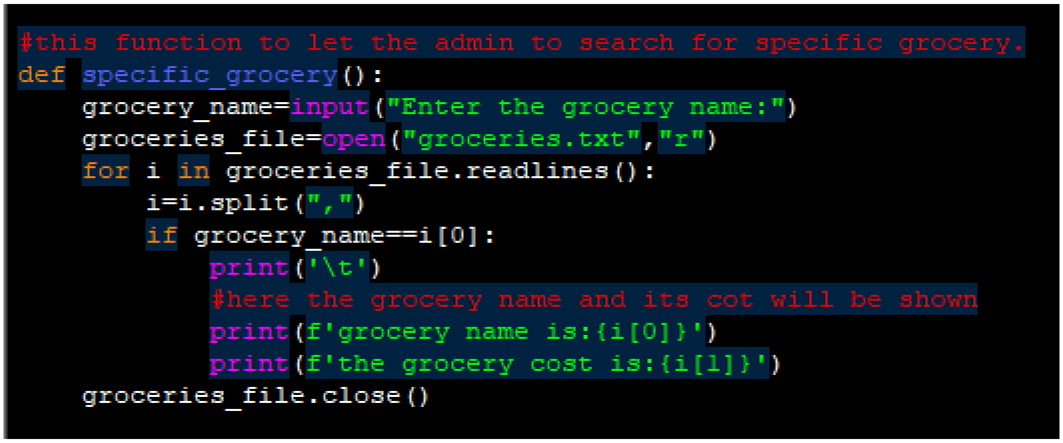
```
#this function is to let the coustomer dispolay his own order history
#the premeter here so the customer can see his own orders only.
def View_orders(username):
    total=0
    orders_file=open("orders.txt","r")
    for i in orders_file.readlines():
        i=i.split(",") #I used here split to split i to list using (",")
        if username==i[0]:
            print(f'you orderd [ {i[1]} ] for  {i[2]} RM')
            #here I have calculate the total of the customer order history cost.
            total=total+int(i[2])
    print(f'the total of your orders [ {total} RM]')
    orders_file.close()
```

Specific customer orders



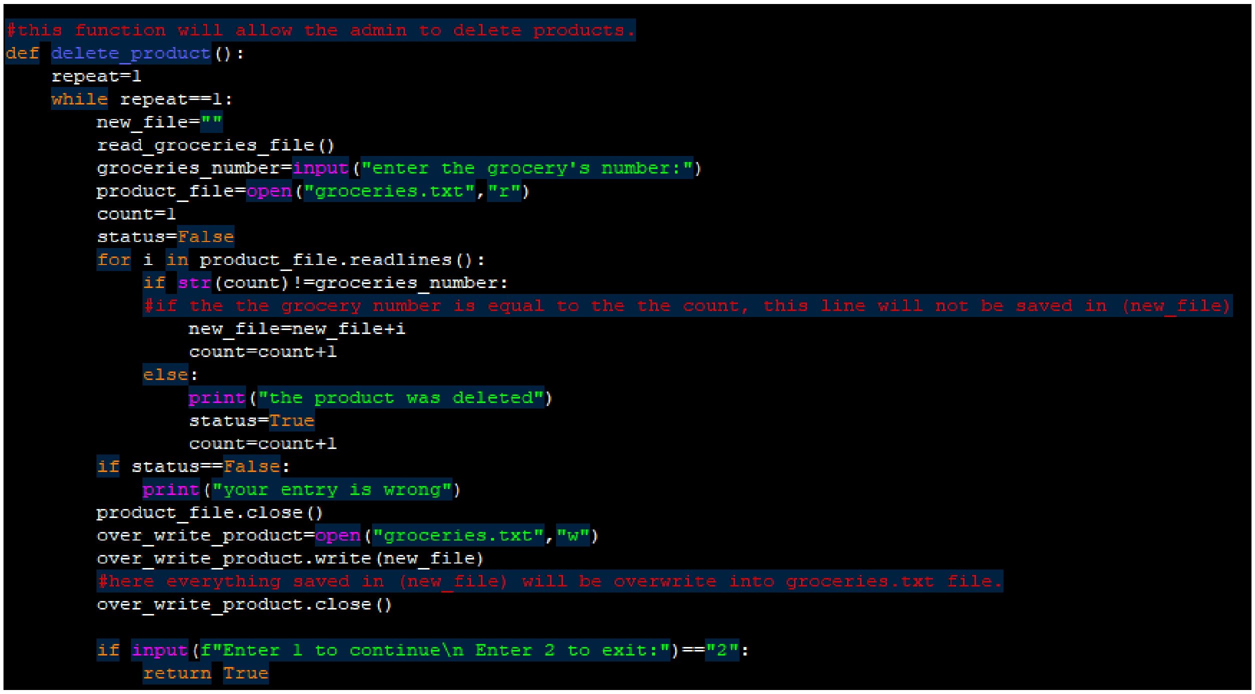
```
#this function if for the admin to get the orders of a specific customer.
def specific_customer_orders():
    total=0
    customer_name=input("Enter the customer name:")
    orders_file=open("orders.txt","r")
    for i in orders_file.readlines():
        i=i.split(",") #I used here split to split i to list using (",")
        if customer_name==i[0]:
            print(f'the order for {customer_name} [ {i[1]} ] for  {i[2]} RM')
            #here I have calculate the total of the customer order history cost.
            total=total+int(i[2])
    print(f'the total cost of {customer_name} is [ {total} RM]')
    orders_file.close()
```

Specific grocery function



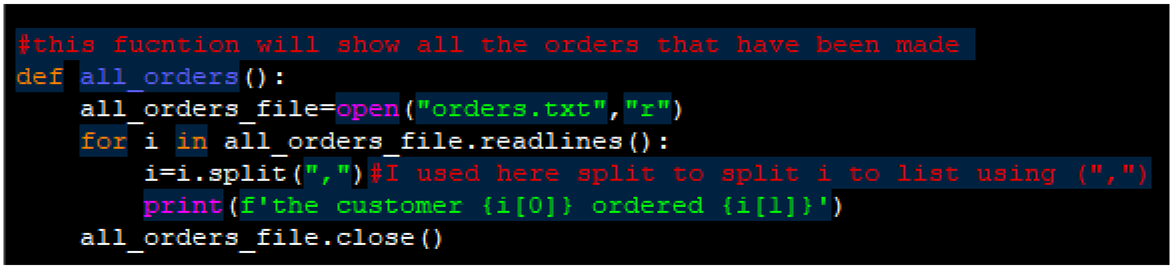
```
#this function to let the admin to search for specific grocery.
def specific_grocery():
    grocery_name=input("Enter the grocery name:")
    groceries_file=open("groceries.txt","r")
    for i in groceries_file.readlines():
        i=i.split(",")
        if grocery_name==i[0]:
            print('\t')
            #here the grocery name and its cot will be shown
            print(f'grocery name is:{i[0]}')
            print(f'the grocery cost is:{i[1]}')
    groceries_file.close()
```

Delete products function



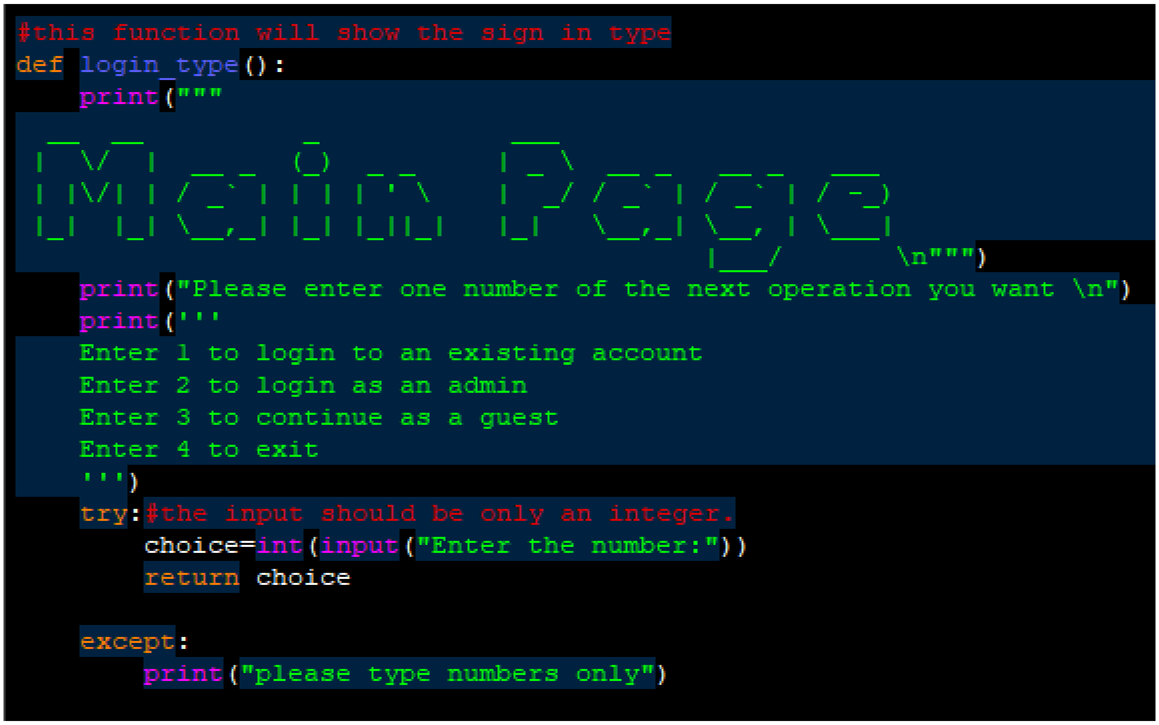
```
#this function will allow the admin to delete products.
def delete_product():
    repeat=1
    while repeat==1:
        new_file=""
        read_groceries_file()
        groceries_number=input("enter the grocery's number:")
        product_file=open("groceries.txt","r")
        count=1
        status=False
        for i in product_file.readlines():
            if str(count)!=groceries_number:
            #if the the grocery number is equal to the the count, this line will not be saved in (new_file)
                new_file=new_file+i
                count=count+1
            else:
                print("the product was deleted")
                status=True
                count=count+1
        if status==False:
            print("your entry is wrong")
        product_file.close()
        over_write_product=open("groceries.txt","w")
        over_write_product.write(new_file)
        #here everything saved in (new_file) will be overwrite into groceries.txt file.
        over_write_product.close()

        if input(f"Enter 1 to continue\n Enter 2 to exit:")=="2":
            return True
```

All orders function



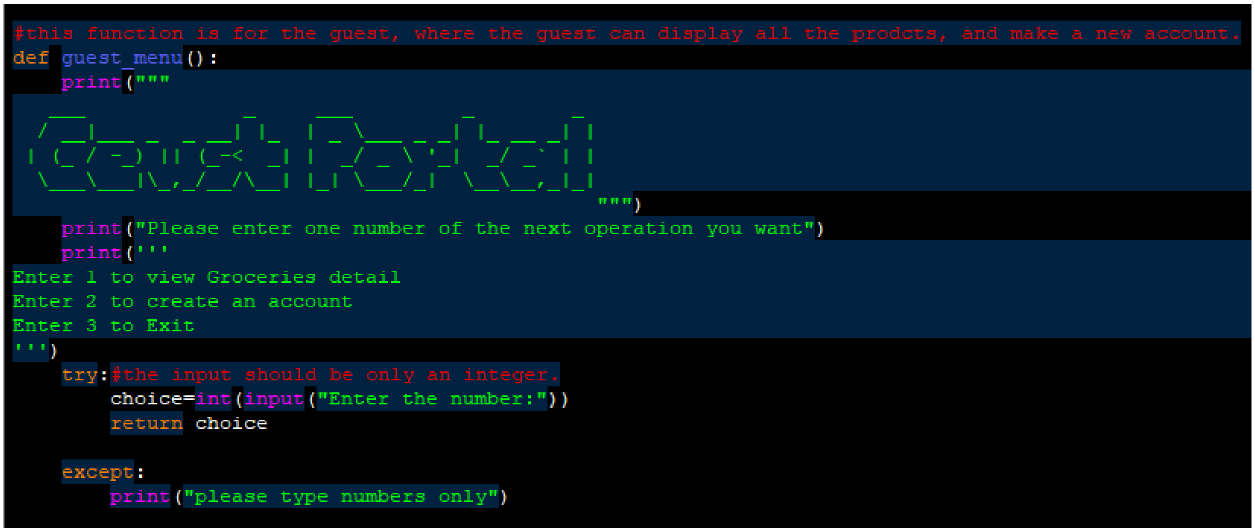
```
#this fucntion will show all the orders that have been made
def all_orders():
    all_orders_file=open("orders.txt","r")
    for i in all_orders_file.readlines():
        i=i.split(",")#I used here split to split i to list using (",")
        print(f'the customer {i[0]} ordered {i[1]}')
    all_orders_file.close()
```

Login type function



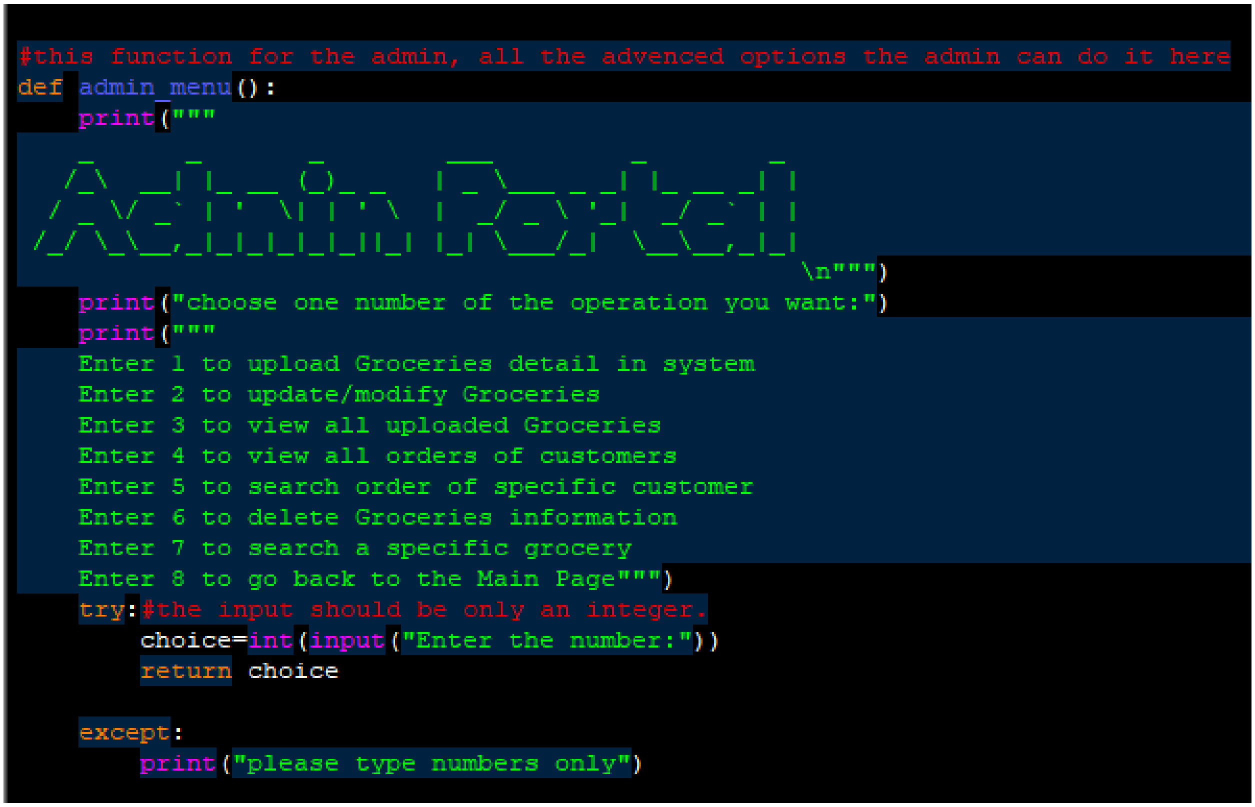
```
#this function will show the sign in type
def login_type():
    print("""

  __  __          _            ___
 |  \/  |  __ _  (_)  _ _     | _ \  __ _   __ _   ___
 | |\/| | / _` | | | | ' \    |  _/ / _` | / _` | / -_)
 |_|  |_| \__,_| |_| |_||_|   |_|   \__,_| \__, | \___|
                                           |___/       \n""")
    print("Please enter one number of the next operation you want \n")
    print('''
    Enter 1 to login to an existing account
    Enter 2 to login as an admin
    Enter 3 to continue as a guest
    Enter 4 to exit
    ''')
    try:#the input should be only an integer.
        choice=int(input("Enter the number:"))
        return choice

    except:
        print("please type numbers only")
```

Guest function



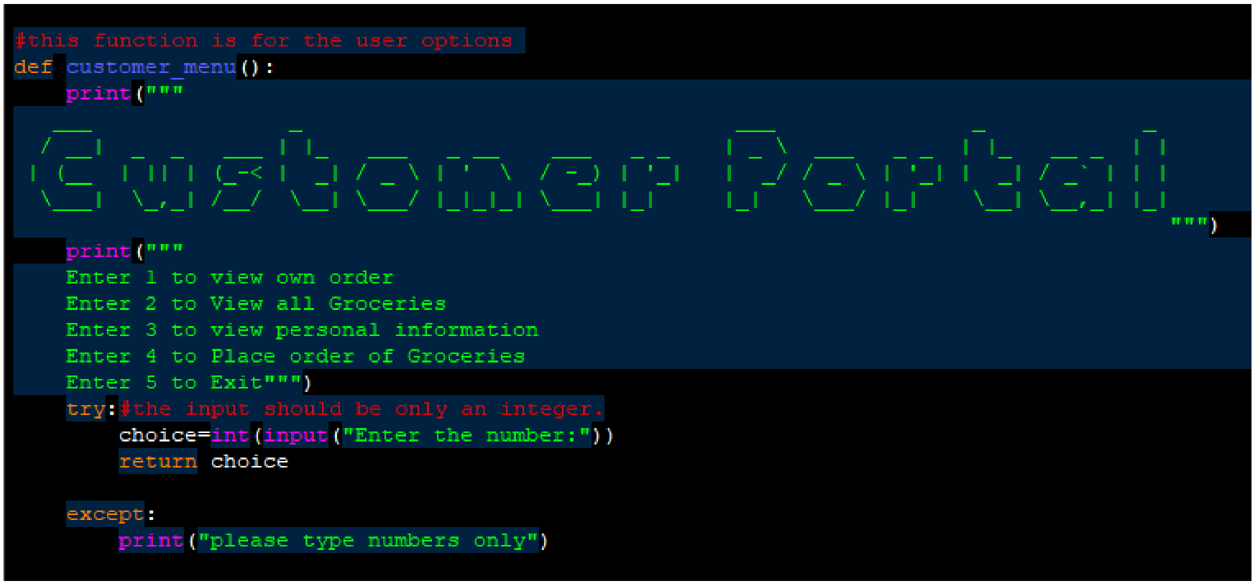
```
#this function is for the guest, where the guest can display all the prodcts, and make a new account.
def guest_menu():
    print("""

   ___               _     ___          _        _
  / __|___ _  _ __| |_  | _ \___ _ _| |_ __ _| |
 | (_ / -_) || (_-<  _| |  _/ _ \ '_|  _/ _` | |
  \___\___|\_,_/__/\__| |_|\___/_|  \__\__,_|_|
                                                """)
    print("Please enter one number of the next operation you want")
    print('''
Enter 1 to view Groceries detail
Enter 2 to create an account
Enter 3 to Exit
''')
    try:#the input should be only an integer.
        choice=int(input("Enter the number:"))
        return choice

    except:
        print("please type numbers only")
```

Admin function



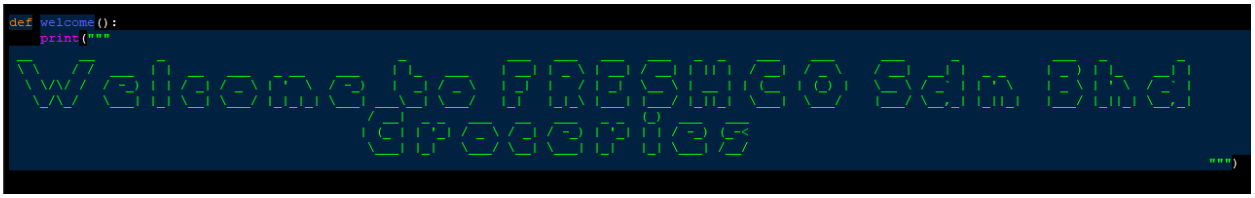
```
#this function for the admin, all the advenced options the admin can do it here
def admin_menu():
    print("""

   _      _       _        ___         _        _
  /_\  __| |_ __ (_)_ _   | _ \___ _ _| |_ __ _| |
 / _ \/ _` | '  \| | ' \  |  _/ _ \ '_|  _/ _` | |
/_/ \_\__,_|_|_|_|_|_||_| |_| \___/_|  \__\__,_|_|
                                                  \n""")
    print("choose one number of the operation you want:")
    print("""
    Enter 1 to upload Groceries detail in system
    Enter 2 to update/modify Groceries
    Enter 3 to view all uploaded Groceries
    Enter 4 to view all orders of customers
    Enter 5 to search order of specific customer
    Enter 6 to delete Groceries information
    Enter 7 to search a specific grocery
    Enter 8 to go back to the Main Page""")
    try:#the input should be only an integer.
        choice=int(input("Enter the number:"))
        return choice

    except:
        print("please type numbers only")
```

Customer function



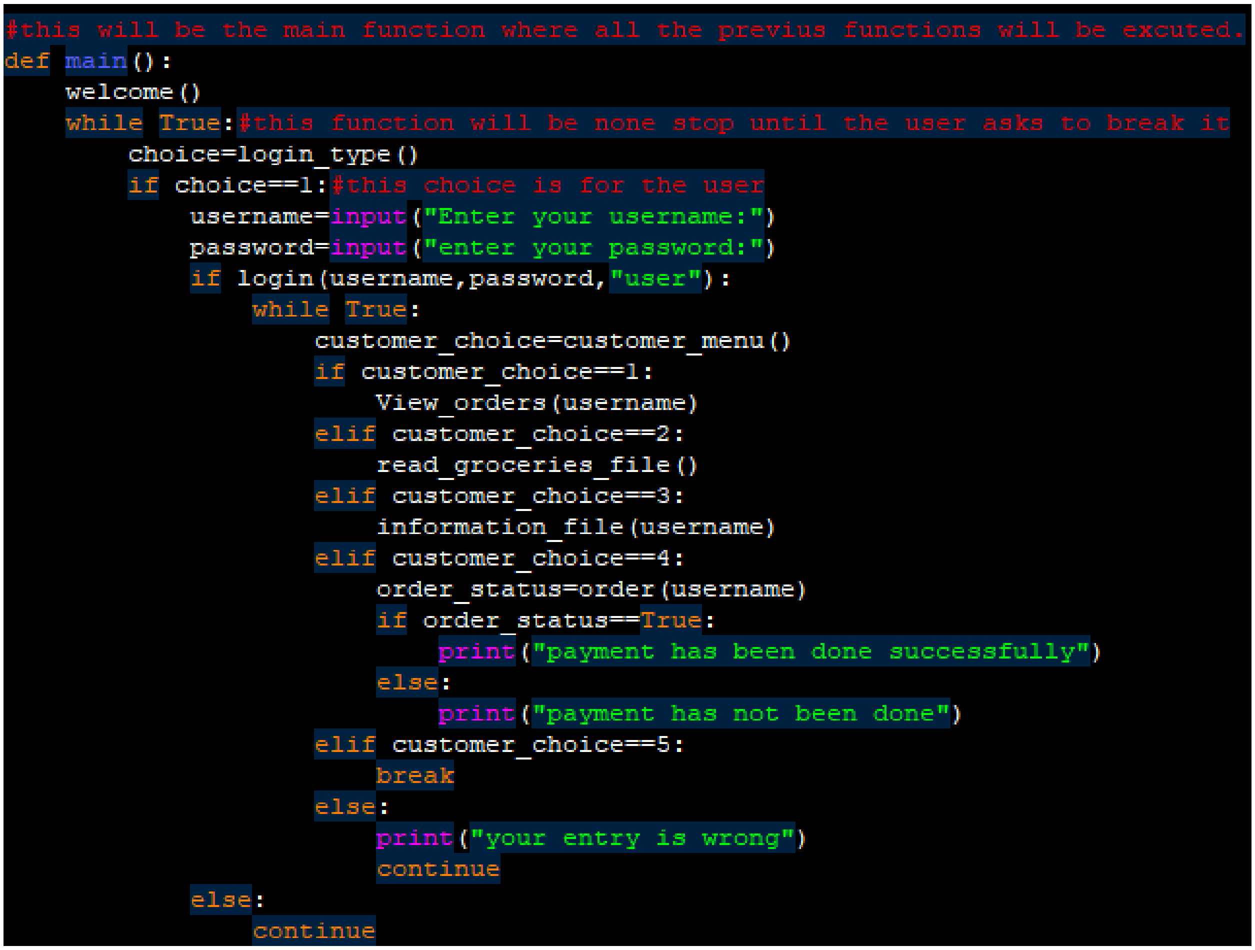
```
#this function is for the user options
def customer_menu():
    print("""
  ___          _                          ___         _        _
 / __| _  _  ___| |_  ___  _ __   ___  _ _   | _ \ ___  _ _ | |_  __ _ | |
| (__ | || |(_-<|  _|/ _ \| '  \ / -_)| '_|  |  _// _ \| '_||  _|/ _` || |
 \___| \_,_|/__/ \__|\___/|_|_|_|\___||_|    |_|  \___/|_|   \__|\__,_||_|
                                                                          """)
    print("""
    Enter 1 to view own order
    Enter 2 to View all Groceries
    Enter 3 to view personal information
    Enter 4 to Place order of Groceries
    Enter 5 to Exit""")
    try:#the input should be only an integer.
        choice=int(input("Enter the number:"))
        return choice

    except:
        print("please type numbers only")
```

Welcome function

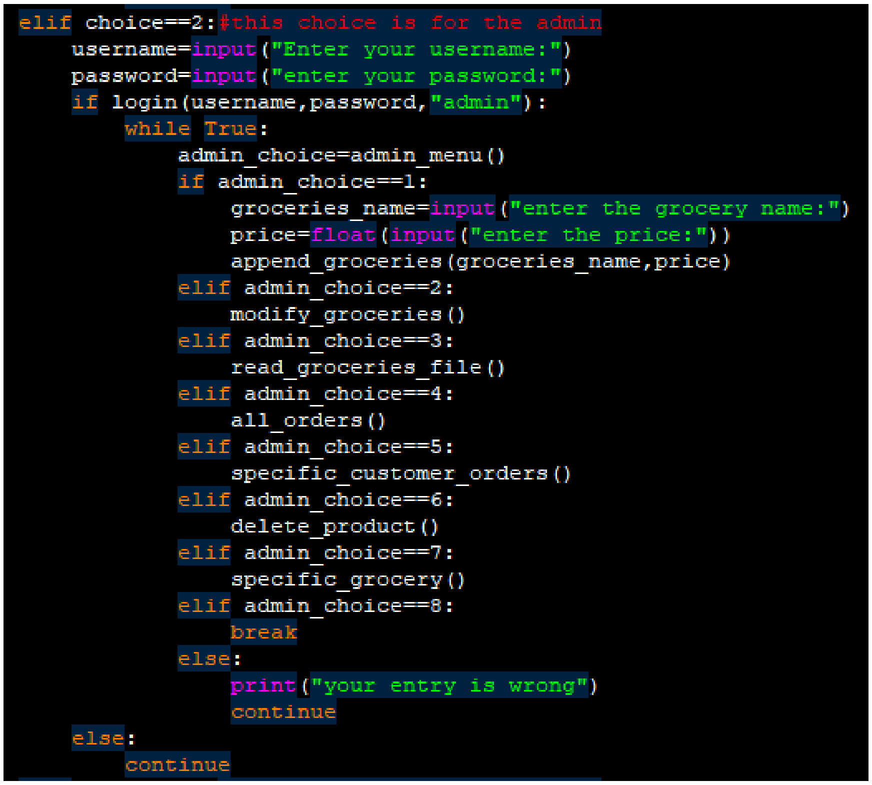


```
def welcome():
    print("""
__        __   _                            _          _____ ____  _____ ____  _   _  ____ ___    ____      _         ____  _         _ 
\ \      / /__| | ___ ___  _ __ ___   ___  | |_ ___   |  ___|  _ \| ____/ ___|| | | |/ ___/ _ \  / ___|  __| |_ __   | __ )| |__   __| |
 \ \ /\ / / _ \ |/ __/ _ \| '_ ` _ \ / _ \ | __/ _ \  | |_  | |_) |  _| \___ \| |_| | |  | | | | \___ \ / _` | '_ \  |  _ \| '_ \ / _` |
  \ V  V /  __/ | (_| (_) | | | | | |  __/ | || (_) | |  _| |  _ <| |___ ___) |  _  | |__| |_| |  ___) | (_| | | | | | |_) | | | | (_| |
   \_/\_/ \___|_|\___\___/|_| |_| |_|\___|__\__\___/  |_|   |_| \_\_____|____/|_| |_|\____\___/  |____/ \__,_|_| |_| |____/|_| |_|\__,_|
                                    / _` | '__/ _ \ / __/ _ \ '__| |/ _ \/ __|
                                   | (_| | | | (_) | (_|  __/ |  | |  __/\__ \
                                    \__, |_|  \___/ \___\___|_|  |_|\___||___/
                                                                                                                                     """)
```

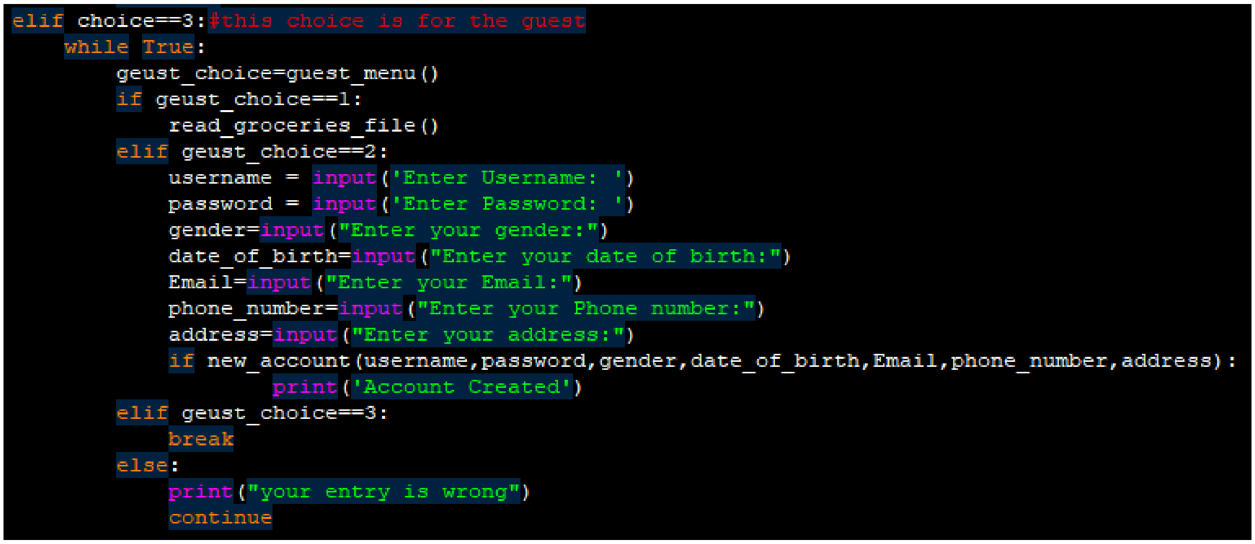
Main function



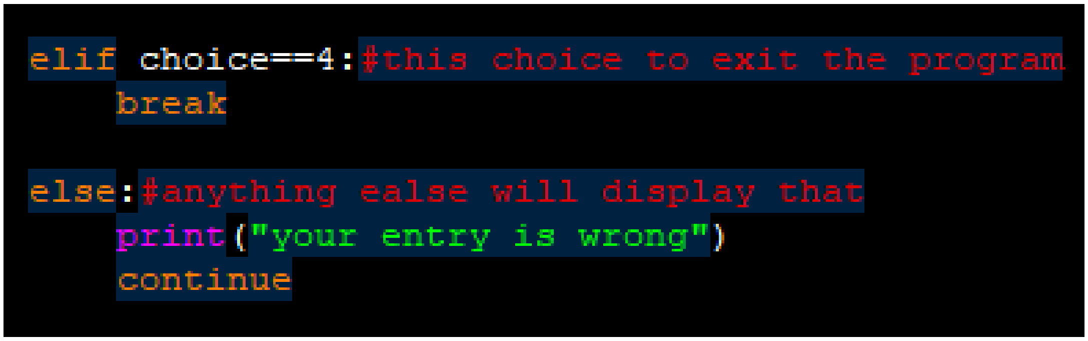
```
#this will be the main function where all the previus functions will be excuted.
def main():
    welcome()
    while True:#this function will be none stop until the user asks to break it
        choice=login_type()
        if choice==1:#this choice is for the user
            username=input("Enter your username:")
            password=input("enter your password:")
            if login(username,password,"user"):
                while True:
                    customer_choice=customer_menu()
                    if customer_choice==1:
                        View_orders(username)
                    elif customer_choice==2:
                        read_groceries_file()
                    elif customer_choice==3:
                        information_file(username)
                    elif customer_choice==4:
                        order_status=order(username)
                        if order_status==True:
                            print("payment has been done successfully")
                        else:
                            print("payment has not been done")
                    elif customer_choice==5:
                        break
                    else:
                        print("your entry is wrong")
                        continue
            else:
                continue
```



```
elif choice==2:#this choice is for the admin
    username=input("Enter your username:")
    password=input("enter your password:")
    if login(username,password,"admin"):
        while True:
            admin_choice=admin_menu()
            if admin_choice==1:
                groceries_name=input("enter the grocery name:")
                price=float(input("enter the price:"))
                append_groceries(groceries_name,price)
            elif admin_choice==2:
                modify_groceries()
            elif admin_choice==3:
                read_groceries_file()
            elif admin_choice==4:
                all_orders()
            elif admin_choice==5:
                specific_customer_orders()
            elif admin_choice==6:
                delete_product()
            elif admin_choice==7:
                specific_grocery()
            elif admin_choice==8:
                break
            else:
                print("your entry is wrong")
                continue
    else:
        continue
```



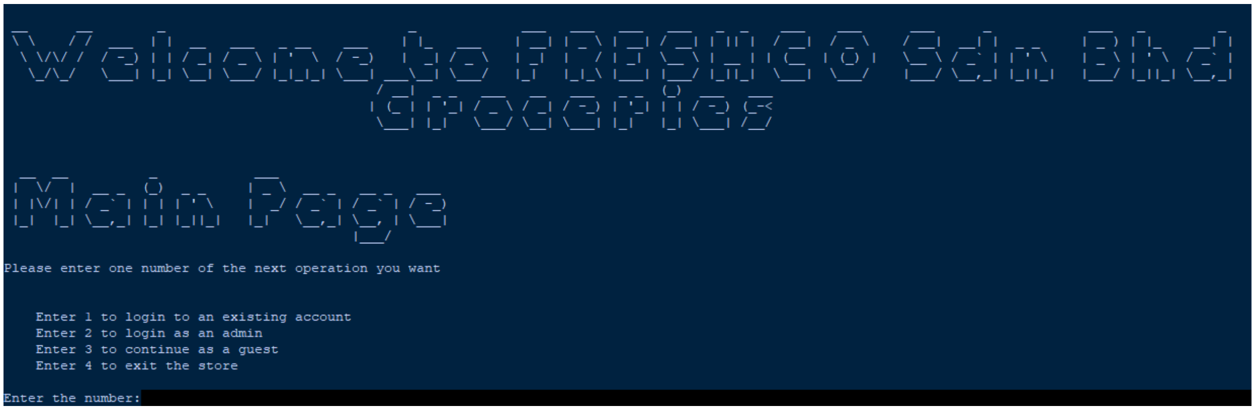
```
elif choice==3:#this choice is for the guest
    while True:
        geust_choice=guest_menu()
        if geust_choice==1:
            read_groceries_file()
        elif geust_choice==2:
            username = input('Enter Username: ')
            password = input('Enter Password: ')
            gender=input("Enter your gender:")
            date_of_birth=input("Enter your date of birth:")
            Email=input("Enter your Email:")
            phone_number=input("Enter your Phone number:")
            address=input("Enter your address:")
            if new_account(username,password,gender,date_of_birth,Email,phone_number,address):
                print('Account Created')
        elif geust_choice==3:
            break
        else:
            print("your entry is wrong")
            continue
```



```
elif choice==4:#this choice to exit the program
    break

else:#anything ealse will display that
    print("your entry is wrong")
    continue
```

sample input/output

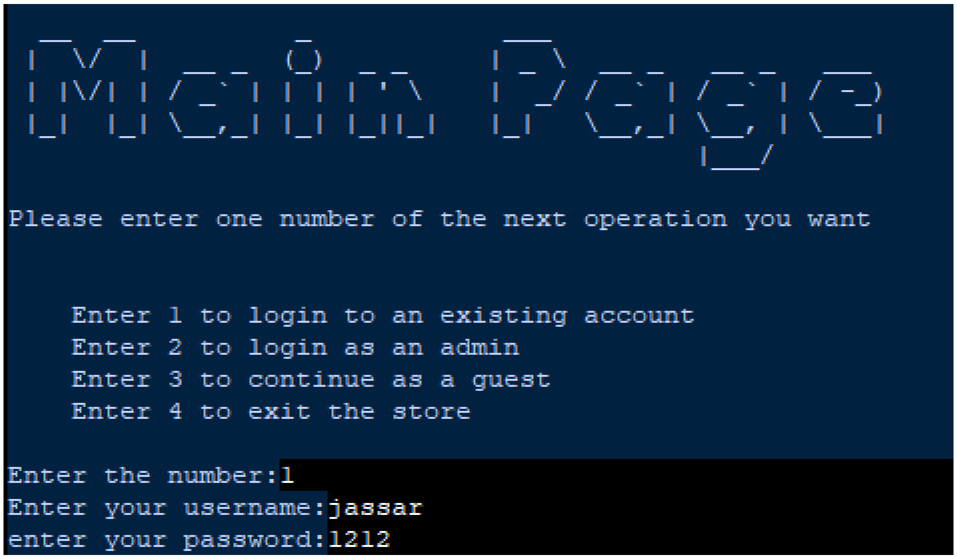
Main Page



```
Please enter one number of the next operation you want


    Enter 1 to login to an existing account
    Enter 2 to login as an admin
    Enter 3 to continue as a guest
    Enter 4 to exit the store

Enter the number:
```

This is the first page the user or the admin will see when they enter the grocery store, they will be asked to enter a number upon their needs.

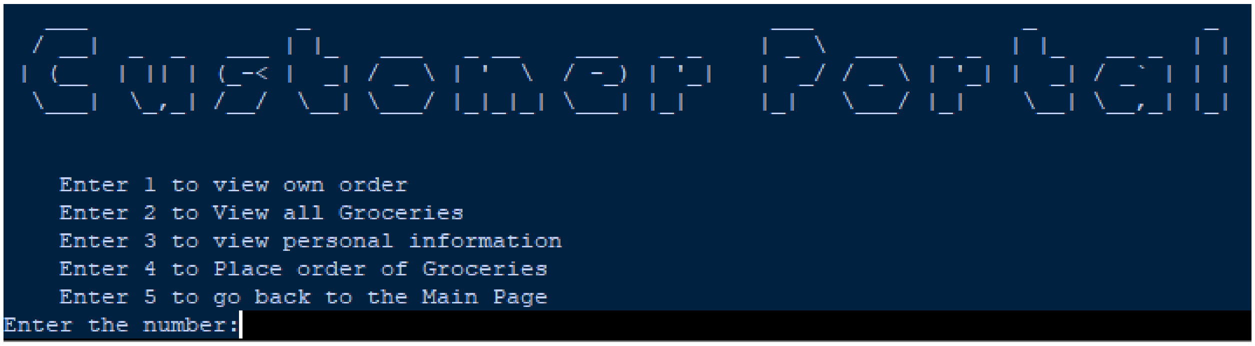
Customer login



```
 __  __       _         ____
|  \/  | __ _(_)_ __   |  _ \ __ _  __ _  ___
| |\/| |/ _` | | '_ \  | |_) / _` |/ _` |/ _ \
| |  | | (_| | | | | | |  __/ (_| | (_| |  __/
|_|  |_|\__,_|_|_| |_| |_|   \__,_|\__, |\___|
                                   |___/

Please enter one number of the next operation you want


    Enter 1 to login to an existing account
    Enter 2 to login as an admin
    Enter 3 to continue as a guest
    Enter 4 to exit the store

Enter the number:1
Enter your username:jassar
enter your password:1212
```

If it was a customer, the customer will enter number 1 to enter the login page, there will be asked to enter the username and the password. if the password or the username is wrong the system will not let this user enter.

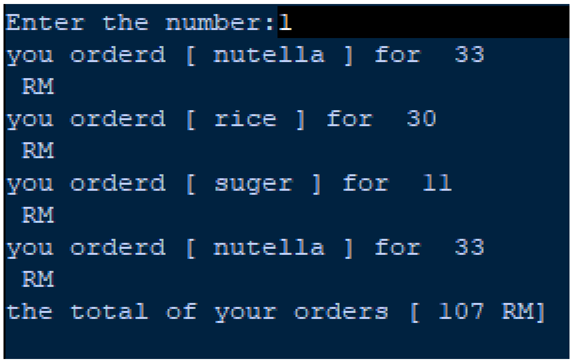
Customer page



```
    Enter 1 to view own order
    Enter 2 to View all Groceries
    Enter 3 to view personal information
    Enter 4 to Place order of Groceries
    Enter 5 to go back to the Main Page
Enter the number:
```

After the customer enters the username and password correctly this page we be displayed, and from here all the functions for the customer will be available, the customer will be able to view his own orders, view all the groceries, view personal information, place a new order, or to go back to the main page.

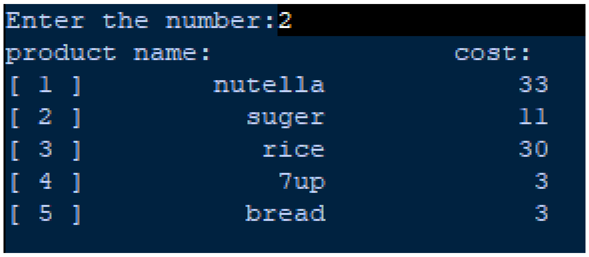
view own order



```
Enter the number:1
you orderd [ nutella ] for  33
 RM
you orderd [ rice ] for  30
 RM
you orderd [ suger ] for  11
 RM
you orderd [ nutella ] for  33
 RM
the total of your orders [ 107 RM]
```

If the customer chose 1, the customer will be able to see his order history and how much in total he has paid.

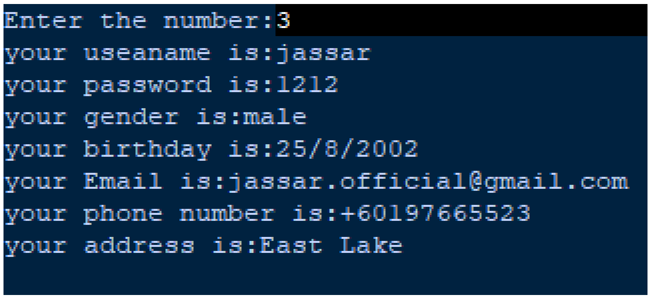
view all groceries



```
Enter the number:2
product name:                   cost:
[ 1 ]        nutella                33
[ 2 ]          suger                11
[ 3 ]           rice                30
[ 4 ]            7up                 3
[ 5 ]          bread                 3
```

When the customer enters 2, all the groceries that are available in the store will be shown with the numbering and cost for each product.

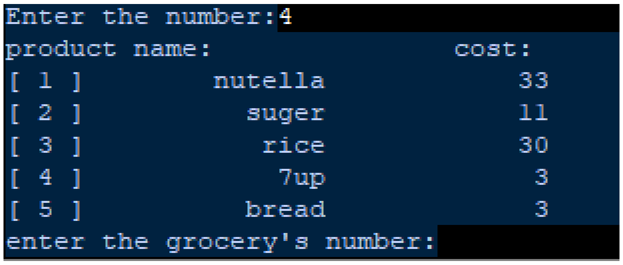
View personal information



```
Enter the number:3
your useaname is:jassar
your password is:1212
your gender is:male
your birthday is:25/8/2002
your Email is:jassar.official@gmail.com
your phone number is:+60197665523
your address is:East Lake
```

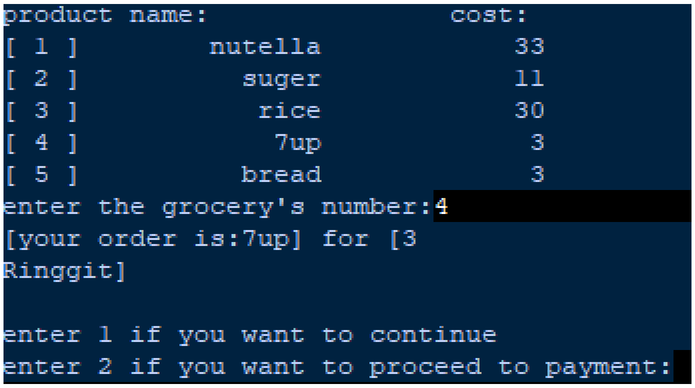
The third choice is to show all the details about the customer that are stored in the users' file, everything that the user entered when he created the account for the first time will be shown here.

Place an order of groceries



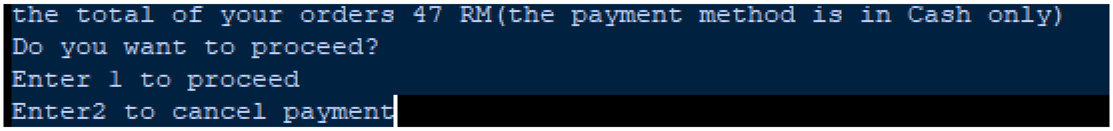
```
Enter the number:4
product name:               cost:
[ 1 ]        nutella            33
[ 2 ]          suger            11
[ 3 ]           rice            30
[ 4 ]            7up             3
[ 5 ]          bread             3
enter the grocery's number:
```

And the last function the customer can do is to place a new order, when the customer enters 4 the groceries will be shown so the customer can choose the product he wants.



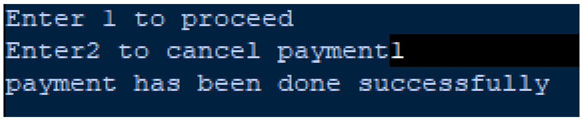
```
product name:                 cost:
[ 1 ]         nutella            33
[ 2 ]           suger            11
[ 3 ]            rice            30
[ 4 ]             7up             3
[ 5 ]           bread             3
enter the grocery's number:4
[your order is:7up] for [3
Ringgit]

enter 1 if you want to continue
enter 2 if you want to proceed to payment:
```

When the customer chooses an item two operations will be shown, one is to continue and add more items, and the second option is to proceed to payment.



```
the total of your orders 47 RM(the payment method is in Cash only)
Do you want to proceed?
Enter 1 to proceed
Enter2 to cancel payment
```

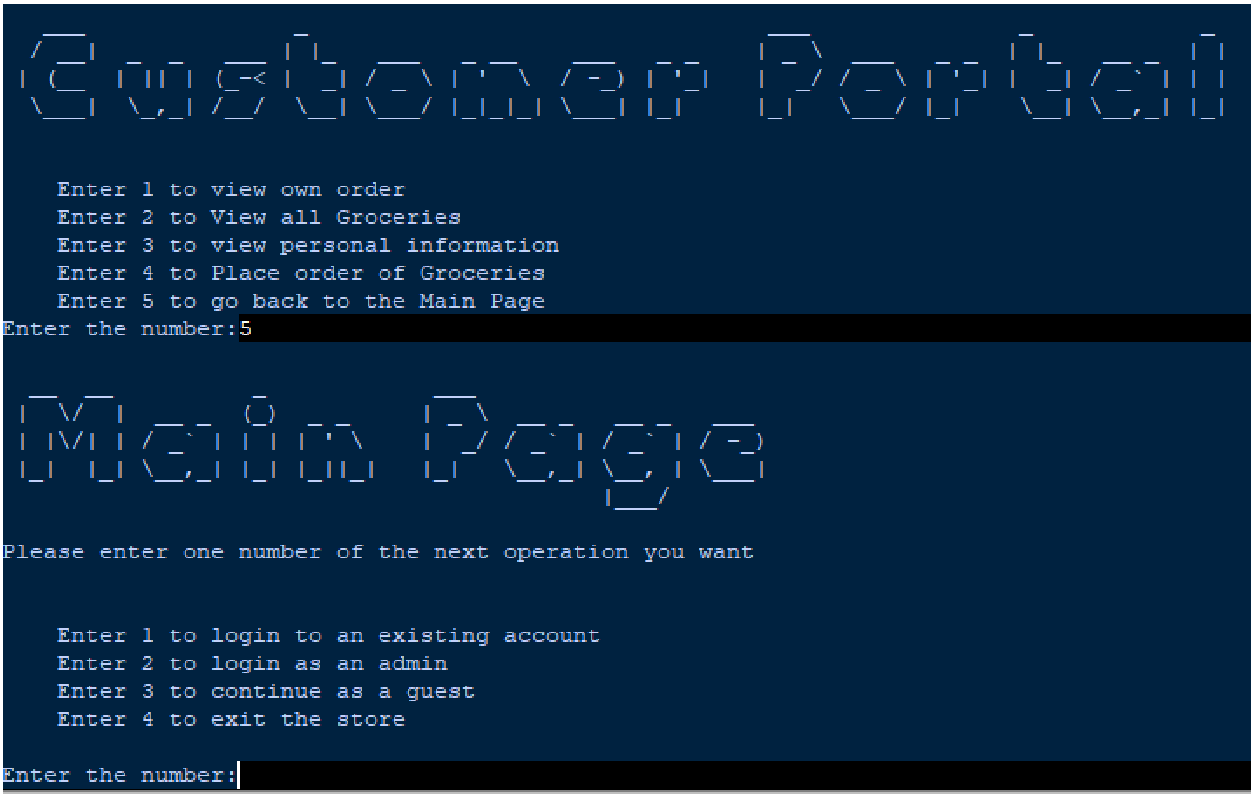
After the customer finishes ordering and wants to proceed to payment, the total of his orders will be shown, either to make the payment or to cancel and go back to the customer portal. The payment method is in cash only.



```
Enter 1 to proceed
Enter2 to cancel payment1
payment has been done successfully
```

If the customer wants to continue with the payment this message will appear that the payment has been done successfully and will be back to the customer portal.

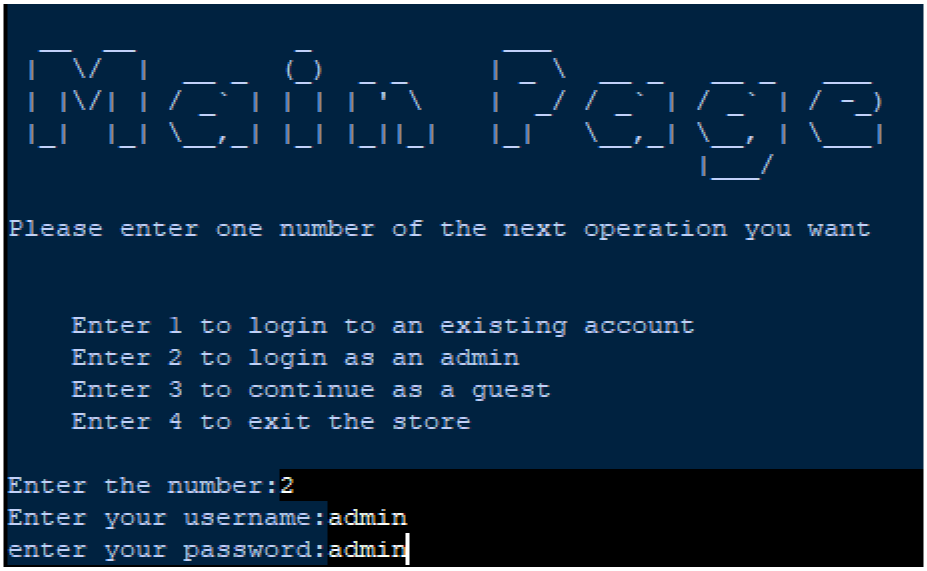
Exit the customer portal



```
  ___          _                              ___            _         _
 / __|  _  _  ___ | |_   ___  _ __   ___  _ _   | _ \  ___  _ _ | |_   __ _ | |
| (__  | || | (_-< |  _| / _ \ | '  \  / -_) | '_|  |  _/ / _ \ | '_| |  _| / _` | | |
 \___| \_,_| /__/  \__| \___/ |_|_|_| \___| |_|    |_|   \___/ |_|    \__| \__,_| |_|


    Enter 1 to view own order
    Enter 2 to View all Groceries
    Enter 3 to view personal information
    Enter 4 to Place order of Groceries
    Enter 5 to go back to the Main Page
Enter the number:5

  __  __          _          ___
 |  \/  |  __ _  (_)  _ _    | _ \  __ _   __ _   ___
 | |\/| | / _` | | | | ' \   |  _/ / _` | / _` | / -_)
 |_|  |_| \__,_| |_| |_||_|  |_|   \__,_| \__, | \___|
                                          |___/

Please enter one number of the next operation you want


    Enter 1 to login to an existing account
    Enter 2 to login as an admin
    Enter 3 to continue as a guest
    Enter 4 to exit the store

Enter the number:
```

To exit the customer page and go back to the main page the customer should enter number 5.

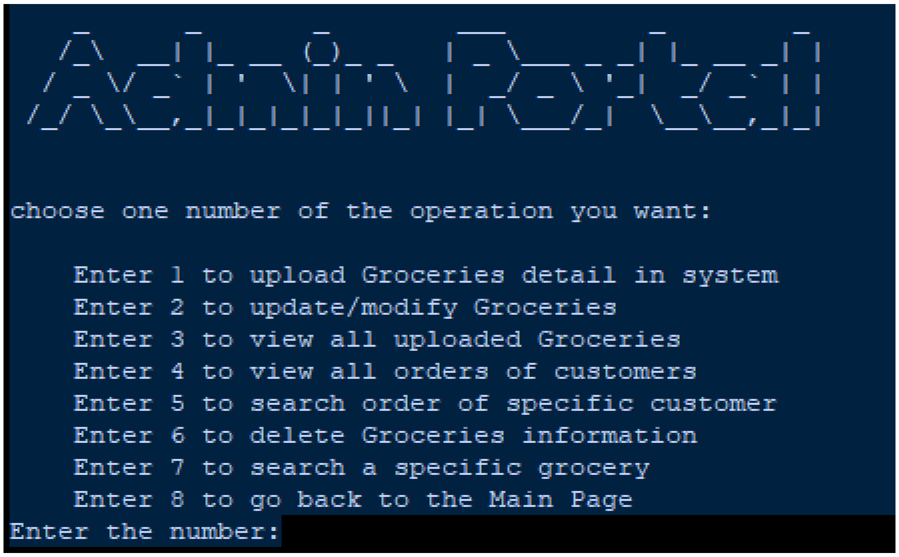
Admin Login



```
  __  __        _        ___
 |  \/  | __ _ (_) _ _  | _ \ __ _  __ _  ___
 | |\/| |/ _` || || ' \ |  _// _` |/ _` |/ -_)
 |_|  |_|\__,_||_||_||_||_|  \__,_|\__, |\___|
                                   |___/

Please enter one number of the next operation you want


    Enter 1 to login to an existing account
    Enter 2 to login as an admin
    Enter 3 to continue as a guest
    Enter 4 to exit the store

Enter the number:2
Enter your username:admin
enter your password:admin
```

If it was an admin who wants to log in, he will choose the second option, after that he will be asked for his username and password, which is admin, admin. Other than that, the system will not give access to enter the admin portal.

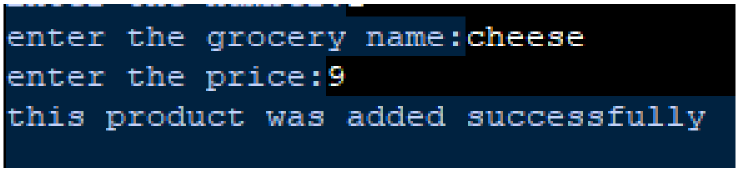
Admin Page



```
    _      _       _         ___         _        _
   /_\  __| |_ __ (_)_ _    | _ \___ _ _| |_ __ _| |
  / _ \/ _` | '  \| | ' \   |  _/ _ \ '_|  _/ _` | |
 /_/ \_\__,_|_|_|_|_|_||_|  |_| \___/_|  \__\__,_|_|


choose one number of the operation you want:

    Enter 1 to upload Groceries detail in system
    Enter 2 to update/modify Groceries
    Enter 3 to view all uploaded Groceries
    Enter 4 to view all orders of customers
    Enter 5 to search order of specific customer
    Enter 6 to delete Groceries information
    Enter 7 to search a specific grocery
    Enter 8 to go back to the Main Page
Enter the number:
```

When the admin enters, he will be able to use a number of functions like uploading a new grocery, modifying a grocery, viewing all uploaded groceries, viewing all orders of customers, viewing a specific customer order, deleting a grocery, and finally exiting the admin portal.

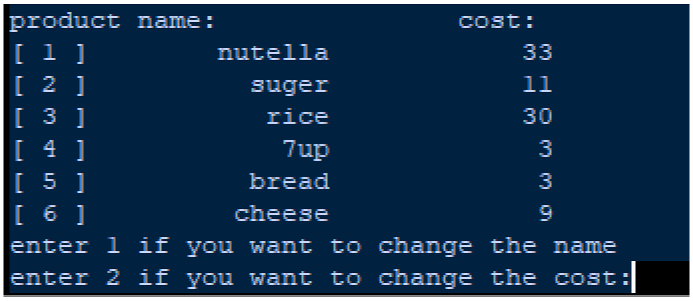
Upload a grocery



```
enter the grocery name:cheese
enter the price:9
this product was added successfully
```

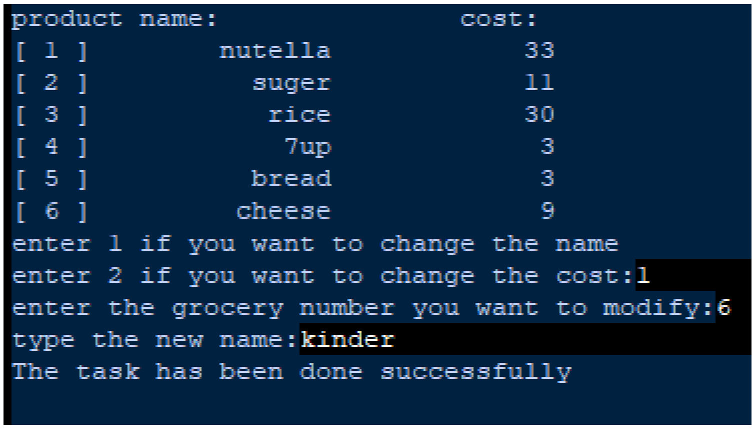
As shown if the admin chooses to upload a grocery, he will be asked to enter the grocery name and its price, after that the grocery will be stored in the groceries file.

Modify a grocery



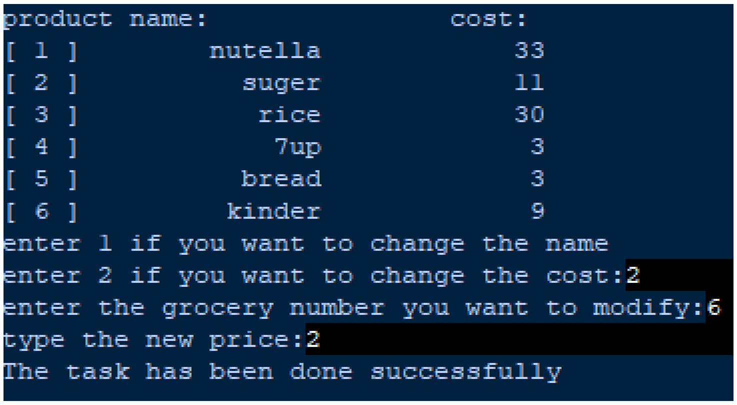
```
product name:                    cost:
[ 1 ]          nutella              33
[ 2 ]            suger              11
[ 3 ]             rice              30
[ 4 ]              7up               3
[ 5 ]            bread               3
[ 6 ]           cheese               9
enter 1 if you want to change the name
enter 2 if you want to change the cost:
```

If the admin wants to modify a grocery he can enter number 2, the groceries menu will be shown and the admin can choose to modify the name of the price of a grocery.



```
product name:                    cost:
[ 1 ]          nutella              33
[ 2 ]            suger              11
[ 3 ]             rice              30
[ 4 ]              7up               3
[ 5 ]            bread               3
[ 6 ]           cheese               9
enter 1 if you want to change the name
enter 2 if you want to change the cost:1
enter the grocery number you want to modify:6
type the new name:kinder
The task has been done successfully
```

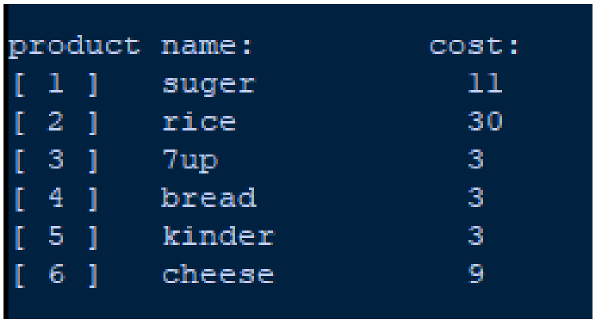
If the admin chose to change the name, he will be asked to choose the grocery number, after that he can modify the name.



```
product name:                cost:
[ 1 ]        nutella            33
[ 2 ]          suger            11
[ 3 ]           rice            30
[ 4 ]            7up             3
[ 5 ]          bread             3
[ 6 ]         kinder             9
enter 1 if you want to change the name
enter 2 if you want to change the cost:2
enter the grocery number you want to modify:6
type the new price:2
The task has been done successfully
```

And if the admin wants to change the price, he will be asked to enter number 2 to change the price, then the grocery number, and finally he can change the price of the grocery.

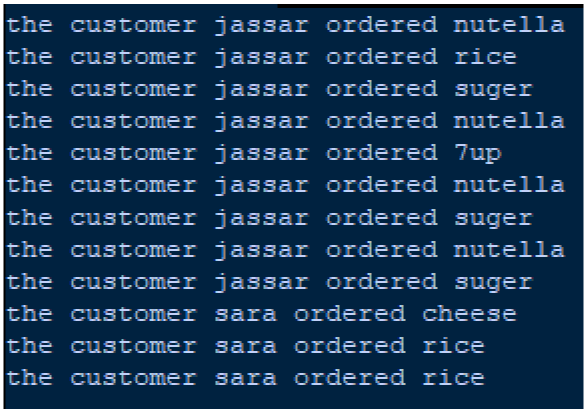
View all groceris



```
product name:          cost:
[ 1 ]    suger            11
[ 2 ]    rice             30
[ 3 ]    7up              3
[ 4 ]    bread            3
[ 5 ]    kinder           3
[ 6 ]    cheese           9
```

If the admin enters 3 all the groceries that are available will be displayed.

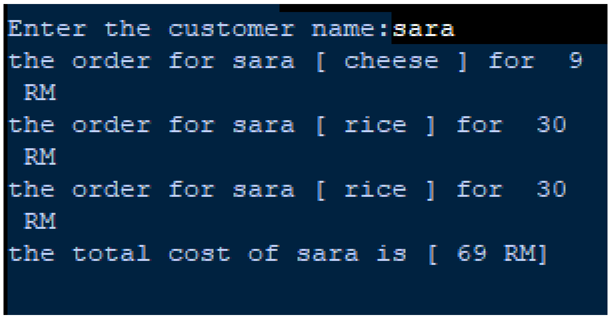
View all customers orders



```
the customer jassar ordered nutella
the customer jassar ordered rice
the customer jassar ordered suger
the customer jassar ordered nutella
the customer jassar ordered 7up
the customer jassar ordered nutella
the customer jassar ordered suger
the customer jassar ordered nutella
the customer jassar ordered suger
the customer sara ordered cheese
the customer sara ordered rice
the customer sara ordered rice
```

The forth option is to display all the orders that have been done by all the customers.

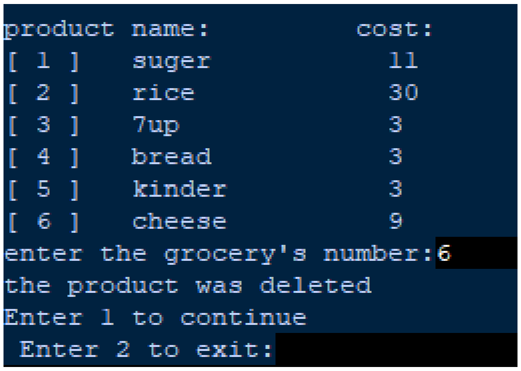
View a specific customer order



```
Enter the customer name:sara
the order for sara [ cheese ] for  9
 RM
the order for sara [ rice ] for  30
 RM
the order for sara [ rice ] for  30
 RM
the total cost of sara is [ 69 RM]
```

The faith choice is to display a specific customer order and the admin can also see the total cost of customer orders.

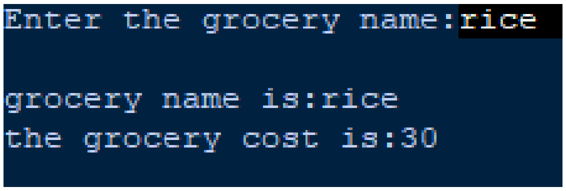
Delete a grocery



```
product name:         cost:
[ 1 ]   suger           11
[ 2 ]   rice            30
[ 3 ]   7up             3
[ 4 ]   bread           3
[ 5 ]   kinder          3
[ 6 ]   cheese          9
enter the grocery's number:6
the product was deleted
Enter 1 to continue
 Enter 2 to exit:
```

This function is to delete a grocery, after choosing the grocery number the admin can either continue deleting or to exit this function.

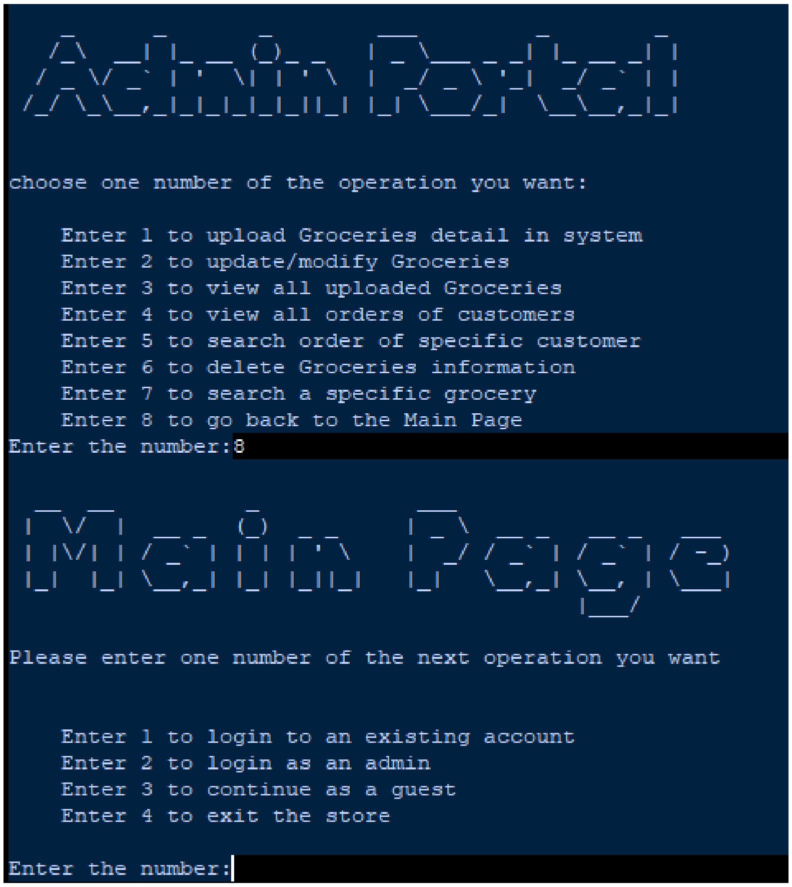
search a specific grocery



```
Enter the grocery name:rice

grocery name is:rice
the grocery cost is:30
```

the seventh function is to search for a specific grocery, in that case after entering the grocery name its details will be shown.

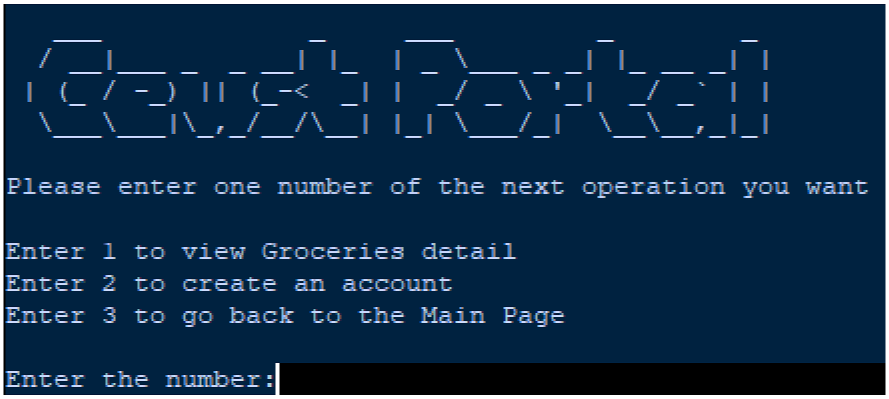
Exit the admin portal



```
   _      _       _        ___         _        _
  /_\  __| |_ __ (_)_ _   | _ \___ _ _| |_ __ _| |
 / _ \/ _` | '  \| | ' \  |  _/ _ \ '_|  _/ _` | |
/_/ \_\__,_|_|_|_|_|_||_| |_| \___/_|  \__\__,_|_|


choose one number of the operation you want:

    Enter 1 to upload Groceries detail in system
    Enter 2 to update/modify Groceries
    Enter 3 to view all uploaded Groceries
    Enter 4 to view all orders of customers
    Enter 5 to search order of specific customer
    Enter 6 to delete Groceries information
    Enter 7 to search a specific grocery
    Enter 8 to go back to the Main Page
Enter the number:8

 __  __      _        ___
|  \/  |__ _(_)_ _   | _ \__ _ __ _ ___
| |\/| / _` | | ' \  |  _/ _` / _` / -_)
|_|  |_\__,_|_|_||_| |_| \__,_\__, \___|
                              |___/

Please enter one number of the next operation you want


    Enter 1 to login to an existing account
    Enter 2 to login as an admin
    Enter 3 to continue as a guest
    Enter 4 to exit the store

Enter the number:
```

And finally, the Eighth option is to go back to the main page.

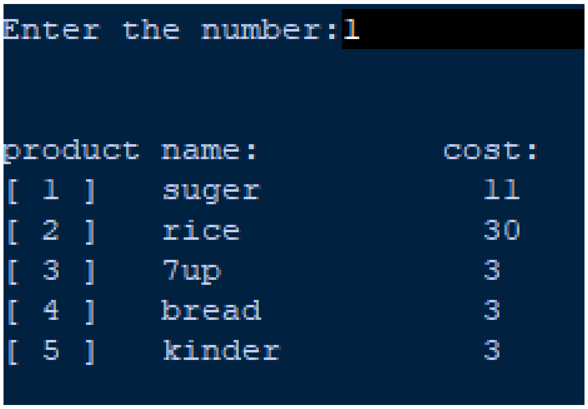
Guest portal



```
  ___                _     ___         _        _
 / __|___ _  _ ___| |_  | _ \___ _ _| |_ __ _| |
| (_ / -_) || (_-<  _| |  _/ _ \ '_|  _/ _` | |
 \___\___|\_,_/__/\__| |_| \___/_|  \__\__,_|_|

Please enter one number of the next operation you want

Enter 1 to view Groceries detail
Enter 2 to create an account
Enter 3 to go back to the Main Page

Enter the number:
```

The third option in the main page is for guests, users that have no account can enter this page to either display the groceries and have a look, or to create a new account.

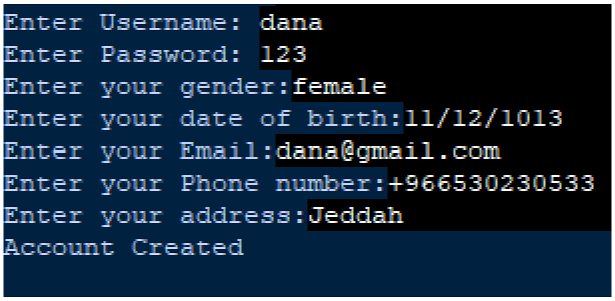
View all groceries



```
Enter the number:1


product name:          cost:
[ 1 ]   suger            11
[ 2 ]   rice             30
[ 3 ]   7up              3
[ 4 ]   bread            3
[ 5 ]   kinder           3
```

The first option is to display the menu.

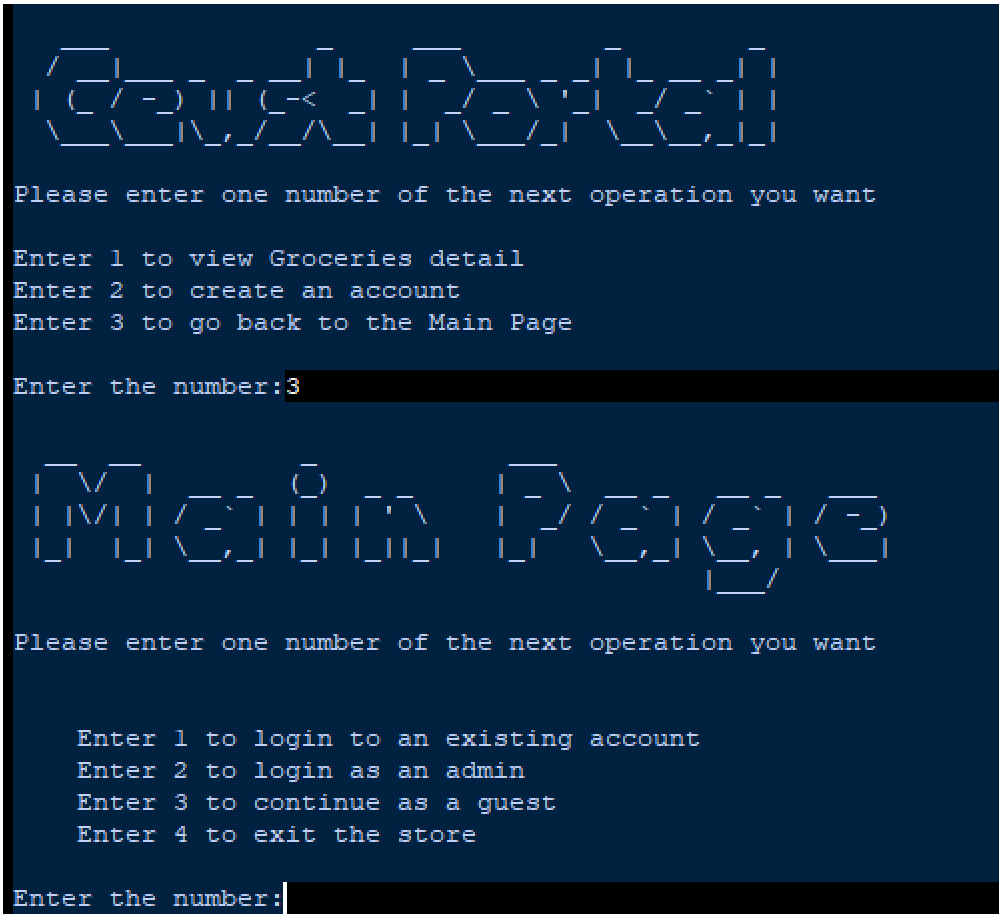
Create an account



```
Enter Username: dana
Enter Password: 123
Enter your gender:female
Enter your date of birth:11/12/1013
Enter your Email:dana@gmail.com
Enter your Phone number:+966530230533
Enter your address:Jeddah
Account Created
```

And the second option is to create a new account, here the user will be asked to fill all the needed information, after that this user can log in as a customer.

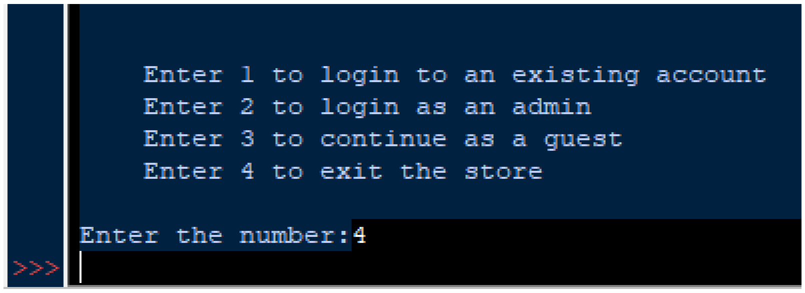
Exit the guest portal



```
  ___              _     ___         _        _ 
 / __|_  _ ___ __| |_  | _ \___ _ _| |_ __ _| |
| (_ | || / -_|_-<  _| |  _/ _ \ '_|  _/ _` | |
 \___|\_,_\___/__/\__| |_| \___/_|  \__\__,_|_|

Please enter one number of the next operation you want

Enter 1 to view Groceries detail
Enter 2 to create an account
Enter 3 to go back to the Main Page

Enter the number:3

 __  __       _        ___               
|  \/  |__ _ (_)_ _   | _ \__ _ __ _ ___ 
| |\/| / _` || | ' \  |  _/ _` / _` / -_)
|_|  |_\__,_||_|_||_| |_| \__,_\__, \___|
                               |___/     

Please enter one number of the next operation you want


    Enter 1 to login to an existing account
    Enter 2 to login as an admin
    Enter 3 to continue as a guest
    Enter 4 to exit the store

Enter the number:
```

The last function in guest menu is to go back to the main page.

Exit the store



```
    Enter 1 to login to an existing account
    Enter 2 to login as an admin
    Enter 3 to continue as a guest
    Enter 4 to exit the store

Enter the number:4
>>>
```

And lastly is the to exit the store by entering number 4.

Conclusion

In conclude, this program was for FRESHCO Sdn Bhd company, they haired me as a programmer to build a program for their new grocery store, I have used Python as the program language.

I have fulfilled everything that was required from me as a programmer, to bullied a program contains conditional statement as IF ELSEIF EELSE, and irritation to keep the program working as long as the user wants too, the loop will be working until the user BREAK it, and to extract the contents from the text files as list, for that I have used SPLIT function to get the content I want and I was using (,) to Split between the contents. All the program was divided into functions, each function will do a specific order that it's programed to do, and the last function which is the main function where all the functions have been executed there

When I was programming, I ran across a number of problems and errors, which prompted me to look for and read up on potential solutions, which ultimately assisted me in improving my skill in the Python programming language.

References

Denisewitsch, V. (2021, January 31). *8 Major Advantages of Online Grocery Shopping*. Retrieved from Food Runner: https://foodrunner.ca/blog/8-major-advantages-of-online-grocery-shopping