

**ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION**

PYTHON PROGRAMMING

CT108-3-1-PYP-LAB-6

INDIVIDUAL ASSIGNMENT

NAME: CHUA YI XUAN

STUDENT NUMBER: TP068394

INTAKE CODE: APD1F2203CS

ASSIGNMENT: PYTHON PROGRAMMING

LECTURER: MR. VIKNESH A/L RAMAMOORTHY

Table of Contents

Cover Page	1
Table of Contents	2
Introduction and Assumptions	3
Pseudocode and Flowcharts for Every Functions	4
Program source code and explanation	51
User Manual	74
I. Menu	74
II. New Customer	75
III. Registered Customer	79
IV. Admin	86
Conclusion	99
References	100

Introduction and Assumptions

The purpose of this assignment is to develop an online pharmacy management system for OCEAN Sdn Bhd. OCEAN decided to develop this online pharmacy management system because the lockdown in Malaysia due to the covid pandemic.

The admin username is admin and the password is admin.

The first customer username is Admin and the password is 12345678. The second customer username is Jayden and the password is 123456.

The medicine name in the system are Roxithromycin, Isotretinoin and Alendronate.

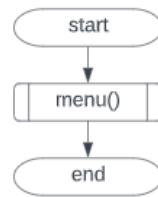
The order has 3 order which are Admin ordered 4 Alendronate, Admin ordered 4 Isotretinoin and Jayden ordered 8 Roxithromycin.

Pseudocode and Flowcharts for Every Functions

Pseudocode:

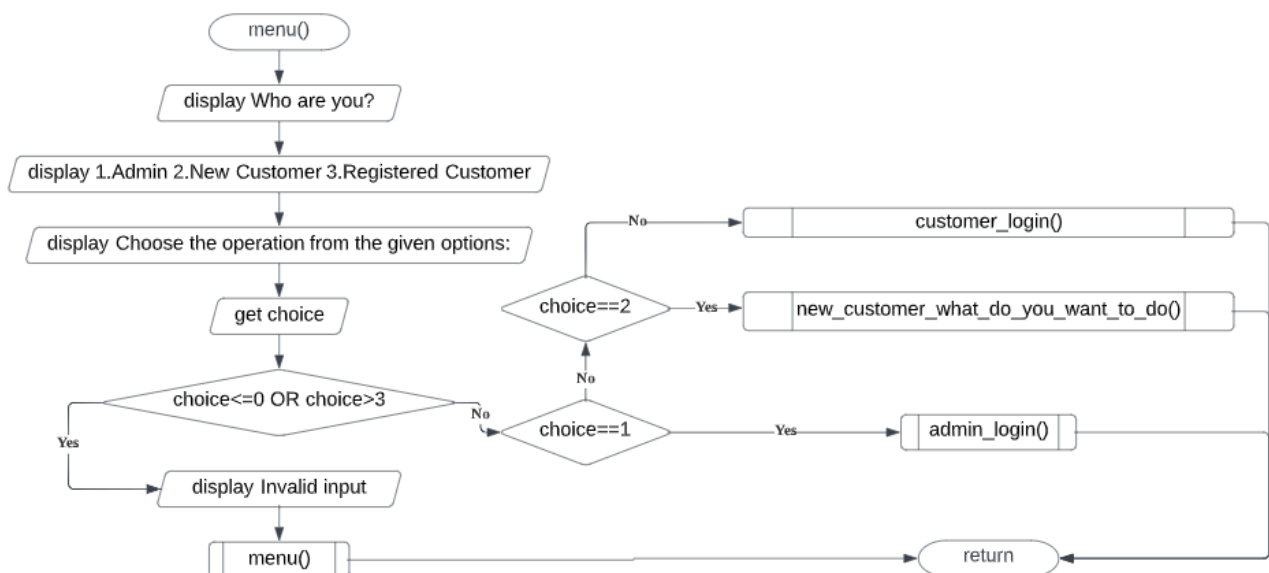
1. start
2. CALL menu()
3. end

Flowchart:



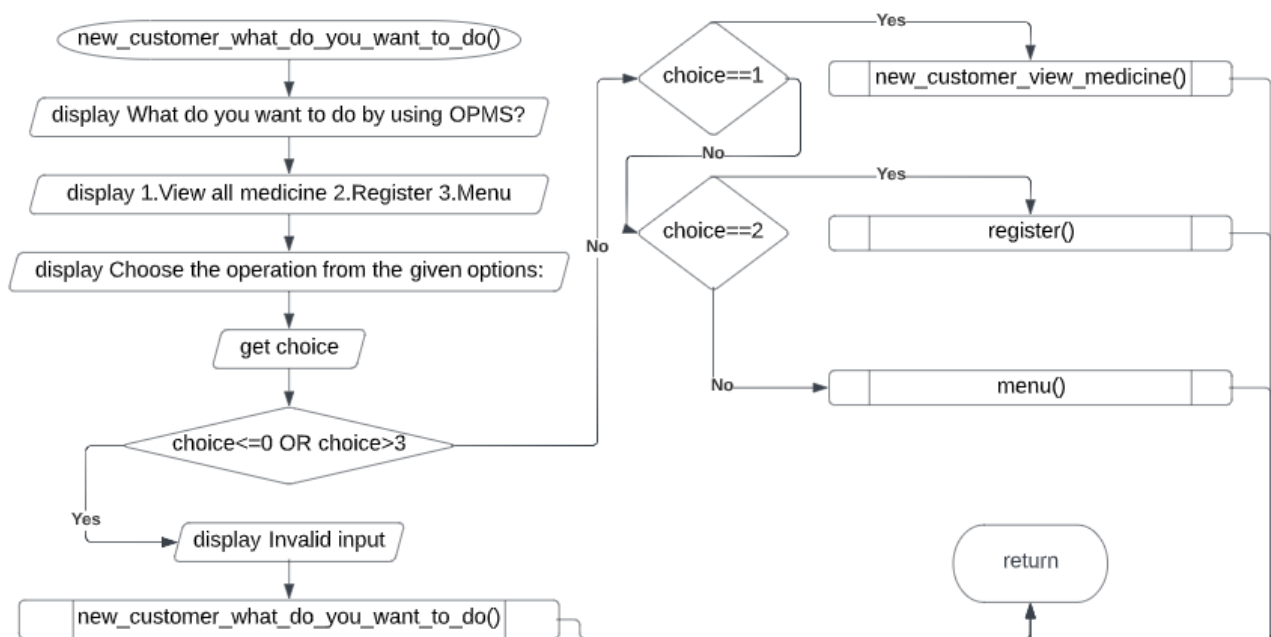
Function: menu()**Pseudocode:**

1. menu()
2. display Who are you?
3. display 1.Admin 2.New Customer 3.Registered Customer
4. display Choose the operation from the given options:
5. get choice
6.
 - 6.1 IF choice<=0 OR choice>3
 - 6.1.1 display Invalid input
 - 6.1.2 CALL menu()
 - 6.2 ELSEIF choice==1
 - 6.2.1 CALL admin_login()
 - 6.3 ELSEIF choice==2
 - 6.3.1 CALL new_customer_what_do_you_want_to_do()
 - 6.4 ELSE
 - 6.4.1 CALL customer_login()
 - 6.6 ENDIF
7. return

Flowchart:

Function: new_customer_what_do_you_want_to_do()**Pseudocode:**

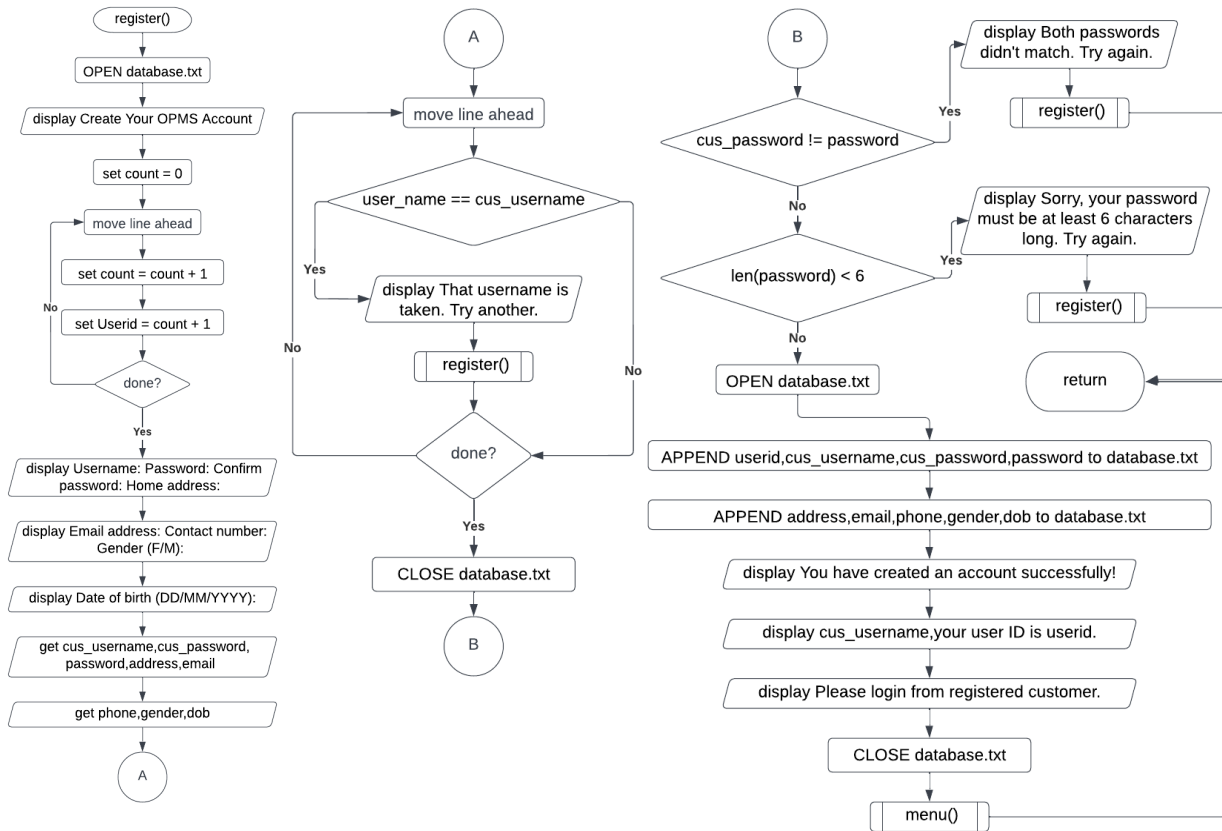
1. new_customer_what_do_you_want_to_do()
2. display What do you want to do by using OPMS?
3. display 1.View all medicine 2.Register 3.Menu
4. display Choose the operation from the given options:
5. get choice
6.
 - 6.1 IF choice<=0 OR choice>3
 - 6.1.1 display Invalid input
 - 6.1.2 CALL new_customer_what_do_you_want_to_do()
 - 6.2 ELSE
 - 6.2.1 IF choice==1
 - 6.2.1.1 CALL new_customer_view_medicine()
 - 6.2.2 ELSEIF choice==2
 - 6.2.2.1 CALL register()
 - 6.2.3 ELSE
 - 6.2.3.1 CALL menu()
 - 6.2.3 ENDIF
 - 6.3 ENDIF
7. return

Flowchart:

Function: register()**Pseudocode:**

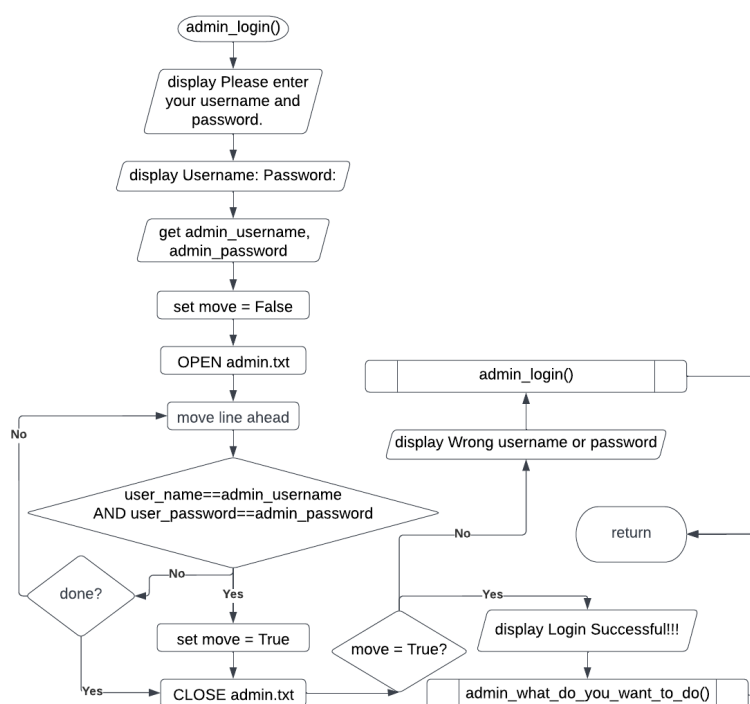
1. register()
2. OPEN database.txt
3. display Create Your OPMS Account
4. set count = 0
5. 5.1 FOR line IN database.txt
 - 5.1.1 set count = count + 1
 - 5.1.2 set Userid = count + 1
6. display Username: Password: Confirm password: Home address:
7. display Email address: Contact number: Gender (F/M):
8. display Date of birth (DD/MM/YYYY):
9. get cus_username,cus_password,password,address,email
10. get phone,gender,dob
11. 11.1 FOR line IN database.txt
 - 11.1.1 IF user_name == cus_username
 - 11.1.1.1 display That username is taken. Try another.
 - 11.1.1.2 CALL register()
 - 11.1.1.3 BREAK
 - 11.1.2 ELSE
 - 11.1.2.1 CONTINUE
 - 11.1.3 ENDIF
12. CLOSE database.txt
13. 13.1 IF cus_password != password
 - 13.1.1 display Both passwords didn't match. Try again.
 - 13.1.2 CALL register()
- 13.2 ELSEIF len(password) < 6
 - 13.2.1 display Sorry, your password must be at least 6 characters long. Try again.
 - 13.2.2 CALL register()
- 13.3 ELSE
 - 13.3.1 OPEN database.txt
 - 13.3.2 APPEND userid,cus_username,cus_password,password to database.txt
 - 13.3.3 APPEND address,email,phone,gender,dob to database.txt
 - 13.3.4 display You have created an account successfully!
 - 13.3.5 display cus_username,your user ID is userid.
 - 13.3.6 display Please login from registered customer.
 - 13.3.7 CLOSE database.txt
14. CALL menu()
15. return

Flowchart:



Function: admin_login()**Pseudocode:**

1. admin_login()
2. display Please enter your username and password.
3. display Username: Password:
4. get admin_username, admin_password
5. set move = False
6. OPEN admin.txt
7. 7.1 FOR line IN admin.txt
 - 7.1.1 IF user_name==admin_username AND user_password==admin_password
 - 7.1.2 set move = True
 - 7.1.3 BREAK
8. CLOSE admin.txt
9. 9.1 IF move = True
 - 9.1.1 display Login Successful!!!
 - 9.1.2 CALL admin_what_do_you_want_to_do()
- 9.2 ELSE
 - 9.2.1 display Wrong username or password
 - 9.2.2 CALL admin_login()
- 9.3 ENDIF
10. return

Flowchart:

Function: customer_login()**Pseudocode:**

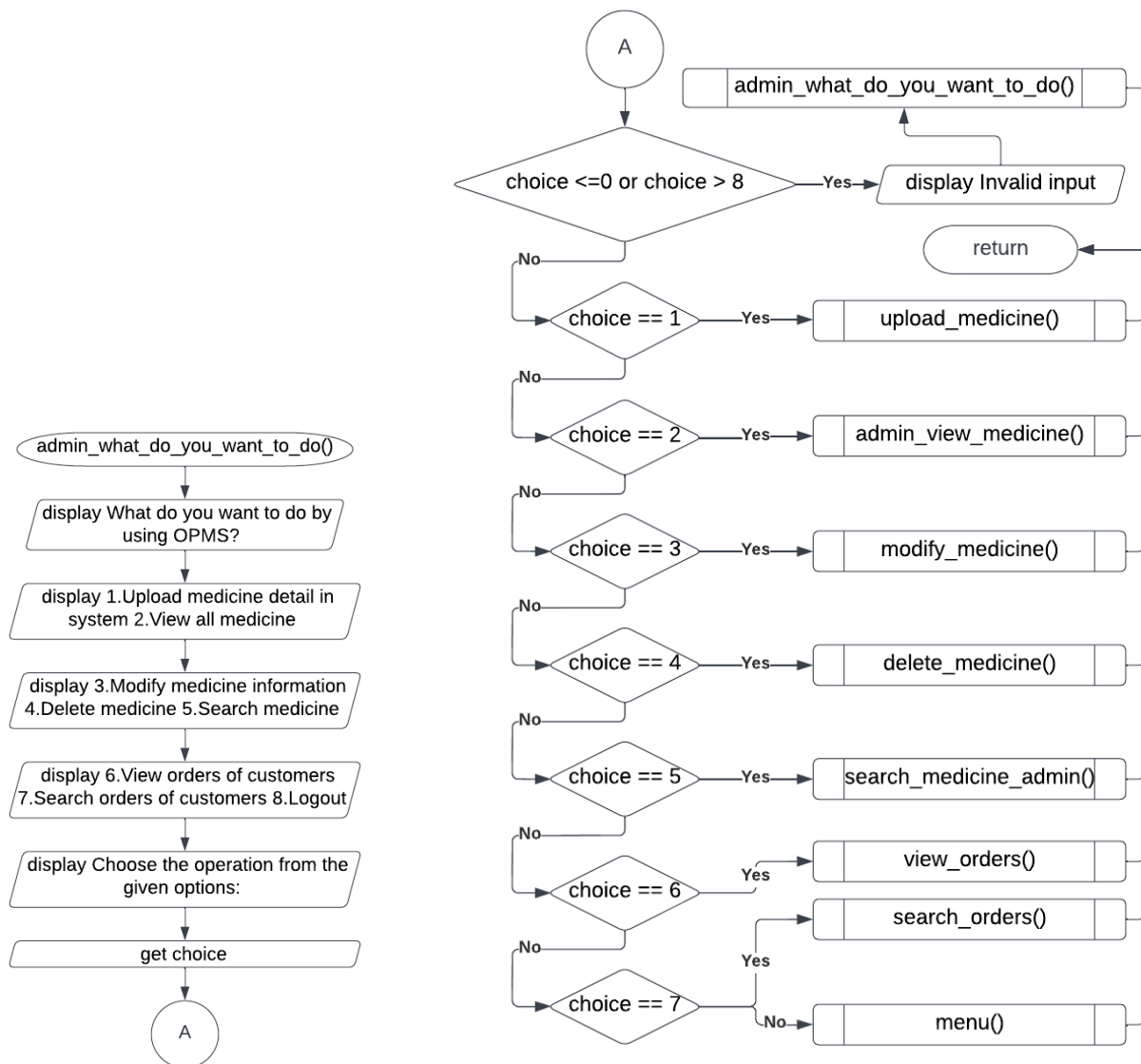
1. customer_login()
2. display Please enter your username and password.
3. display Username: Password:
4. get cus_username,cus_password
5. set move = False
6. OPEN database.txt
7. 7.1 FOR line IN database.txt
 - 7.1.1 IF user_name==cus_username AND user_password==cus_password
 - 7.1.2 set move = True
 - 7.1.3 BREAK
8. CLOSE database.txt
9. 9.1 IF move = True
 - 9.1.1 display Login Successful!!!
 - 9.1.2 CALL registered_customer_what_do_you_want_to_do()
- 9.2 ELSE
 - 9.2.1 display Wrong username or password
 - 9.2.2 CALL customer_login()
- 9.3 ENDIF
10. return

Flowchart:

Function: admin_what_do_you_want_to_do()**Pseudocode:**

```
1. admin_what_do_you_want_to_do()
2. display What do you want to do by using OPMS?
3. display 1.Upload medicine detail in system 2.View all medicine
4. display 3.Modify medicine information 4.Delete medicine 5.Search medicine
5. display 6.View orders of customers 7.Search orders of customers 8.Logout
6. display Choose the operation from the given options:
7. get choice
8.      8.1 IF choice <=0 or choice > 8
          8.1.1 display Invalid input
          8.1.2 CALL admin_what_do_you_want_to_do()
      8.2 ELSE
          8.2.1 IF choice == 1
              8.2.1.1 CALL upload_medicine()
          8.2.2 ELSEIF choice == 2
              8.2.2.1 CALL admin_view_medicine()
          8.2.3 ELSEIF choice == 3
              8.2.3.1 CALL modify_medicine()
          8.2.4 ELSEIF choice == 4
              8.2.4.1 CALL delete_medicine()
          8.2.5 ELSEIF choice == 5
              8.2.5.1 CALL search_medicine_admin()
          8.2.6 ELSEIF choice == 6
              8.2.6.1 CALL view_orders()
          8.2.7 ELSEIF choice == 7
              8.2.7.1 CALL search_orders()
      8.2.8 ELSE
          8.2.8.1 CALL menu()
      8.2.9 ENDIF
      8.3 ENDIF
9. return
```

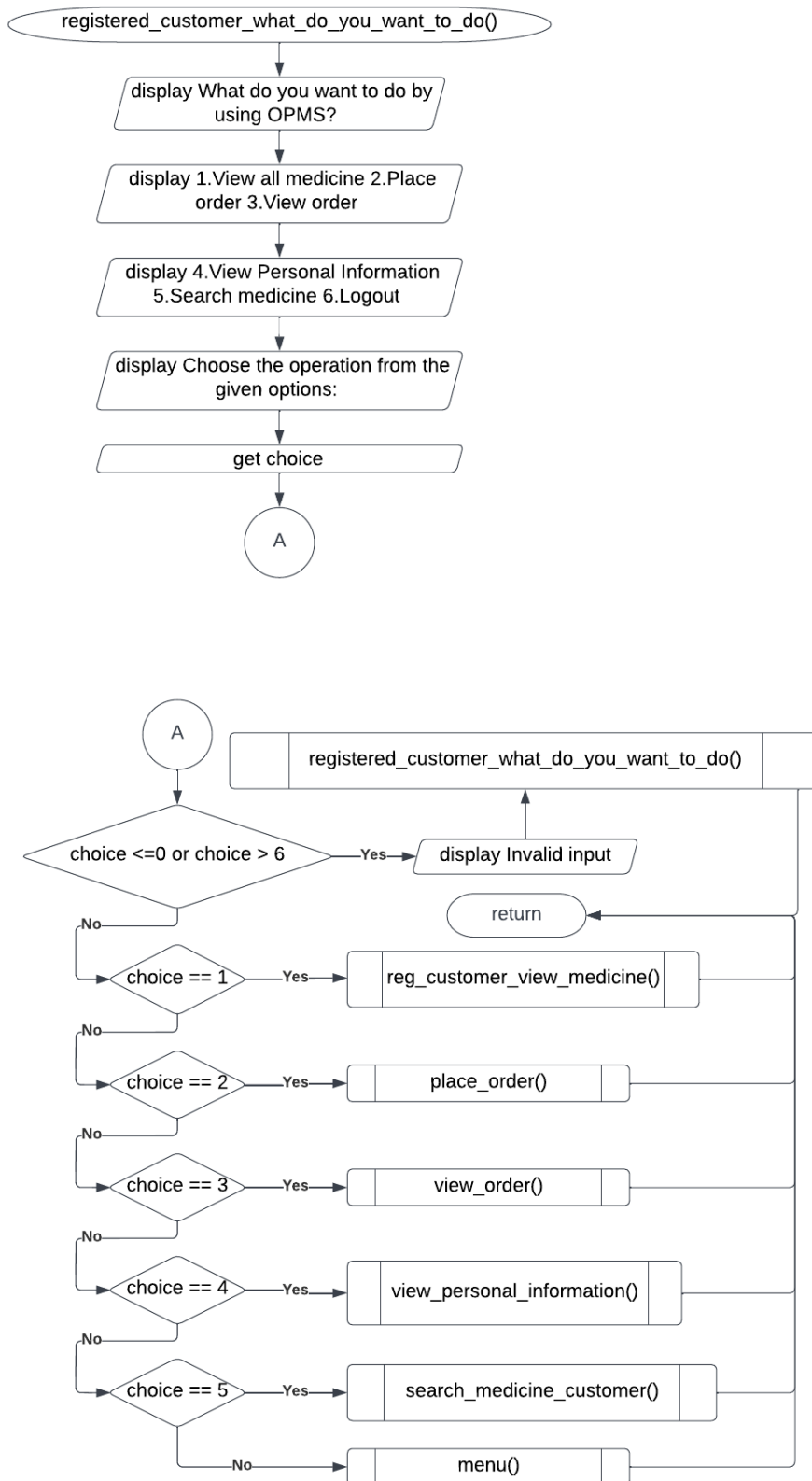
Flowchart:



Function: registered_customer_what_do_you_want_to_do()**Pseudocode:**

```
1. registered_customer_what_do_you_want_to_do()
2. display What do you want to do by using OPMS?
3. display 1.View all medicine 2.Place order 3.View order
4. display 4.View Personal Information 5.Search medicine 6.Logout
5. display Choose the operation from the given options:
6. get choice
7. 7.1 IF choice <=0 or choice > 6
    7.1.1 display Invalid input
    7.1.2 CALL registered_customer_what_do_you_want_to_do()
    7.2 ELSE
        7.2.1 IF choice == 1
            7.2.1.1 CALL reg_customer_view_medicine()
        7.2.2 ELSEIF choice == 2
            7.2.2.1 CALL place_order()
        7.2.3 ELSEIF choice == 3
            7.2.3.1 CALL view_order()
        7.2.4 ELSEIF choice == 4
            7.2.4.1 CALL view_personal_information()
        7.2.5 ELSEIF choice == 5
            7.2.5.1 CALL search_medicine_customer()
        7.2.6 ELSE
            7.2.6.1 CALL menu()
        7.2.9 ENDIF
    7.3 ENDIF
8. return
```

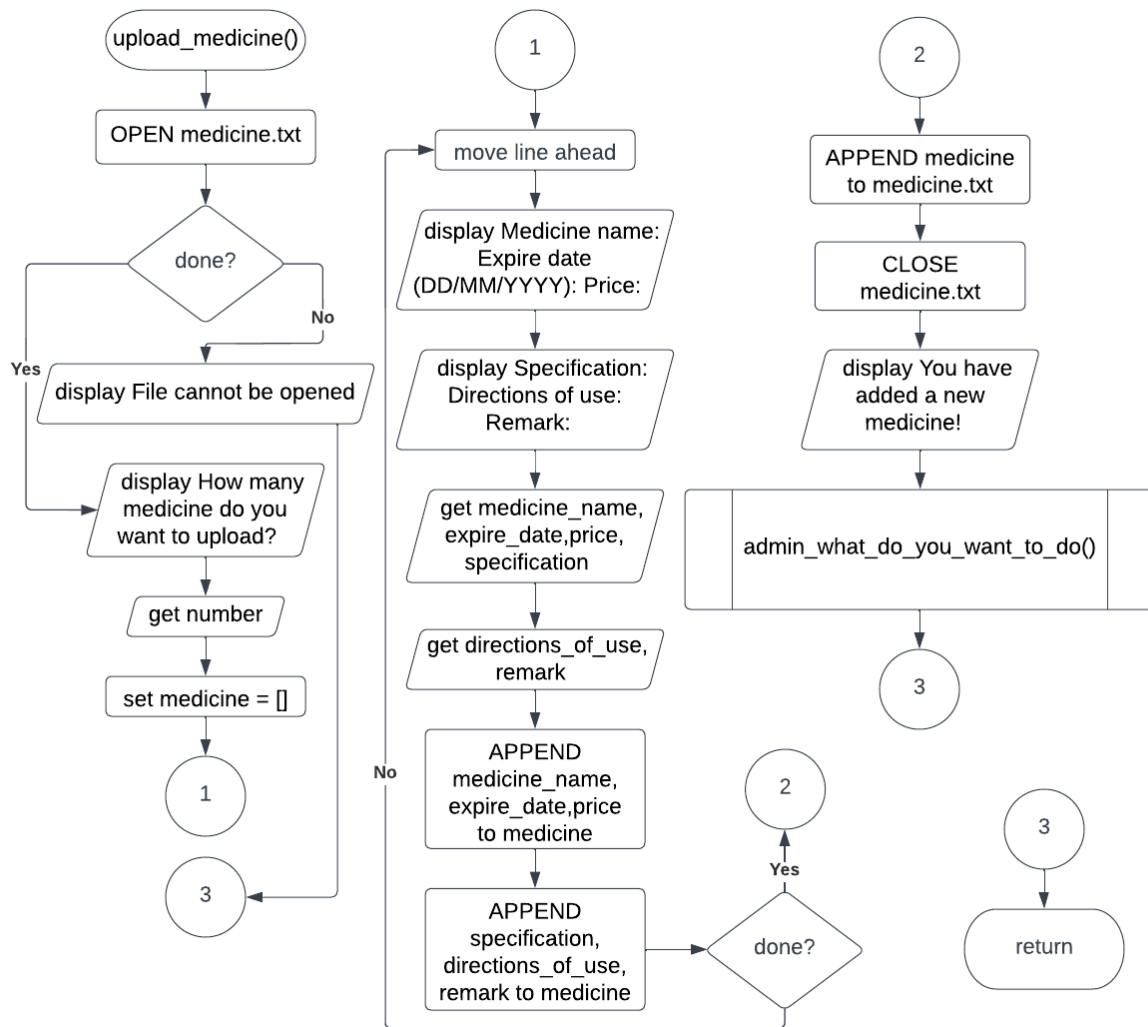
Flowchart:



Function: upload_medicine()**Pseudocode:**

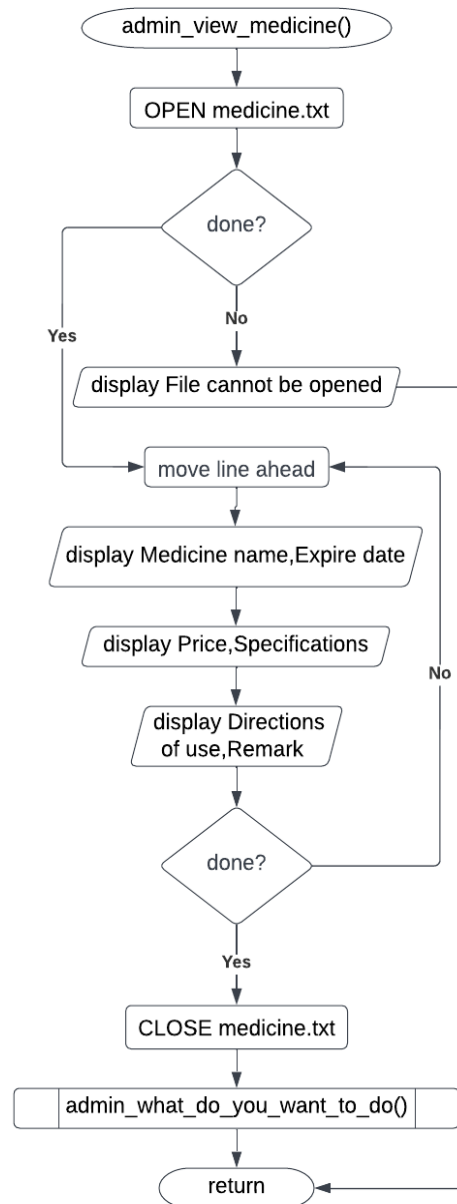
1. upload_medicine()
2. 2.1 TRY
 - 2.1.1 OPEN medicine.txt
 - 2.2 EXCEPT
 - 2.2.1 display File cannot be opened
3. display How many medicine do you want to upload?
4. get number
5. set medicine = []
6. 6.1 FOR line_ IN range(number)
 - 6.1.1 display Medicine name: Expire date (DD/MM/YYYY): Price:
 - 6.1.2 display Specification: Directions of use: Remark:
 - 6.1.3 get medicine_name,expire_date,price,specification
 - 6.1.4 get directions_of_use,remark
 - 6.1.5 APPEND medicine_name,expire_date,price to medicine
 - 6.1.6 APPEND specification,directions_of_use,remark to medicine
7. APPEND medicine to medicine.txt
8. CLOSE medicine.txt
9. display You have added a new medicine!
10. CALL admin_what_do_you_want_to_do()
11. return

Flowchart:



Function: admin_view_medicine()**Pseudocode:**

1. admin_view_medicine()
2. 2.1 TRY
 - 2.1.1 OPEN medicine.txt
- 2.2 EXCEPT
 - 2.2.1 display File cannot be opened
3. 3.1 FOR line IN medicine.txt
 - 3.1.1 display Medicine name,Expire date
 - 3.1.2 display Price,Specifications
 - 3.1.3 display Directions of use,Remark
4. CLOSE medicine.txt
5. CALL admin_what_do_you_want_to_do()
6. return

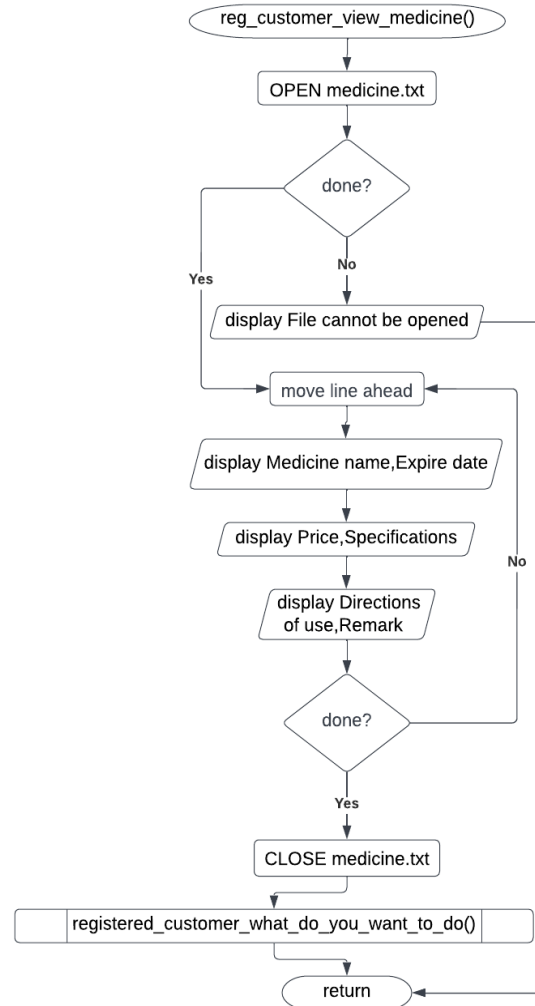
Flowchart:

Function: reg_customer_view_medicine()**Pseudocode:**

```

1. reg_customer_view_medicine()
2.     2.1 TRY
        2.1.1 OPEN medicine.txt
    2.2 EXCEPT
        2.2.1 display File cannot be opened
3.     3.1 FOR line IN medicine.txt
        3.1.1 display Medicine name,Expire date
        3.1.2 display Price,Specifications
        3.1.3 display Directions of use,Remark
4. CLOSE medicine.txt
5. CALL registered_customer_what_do_you_want_to_do()
6. return

```

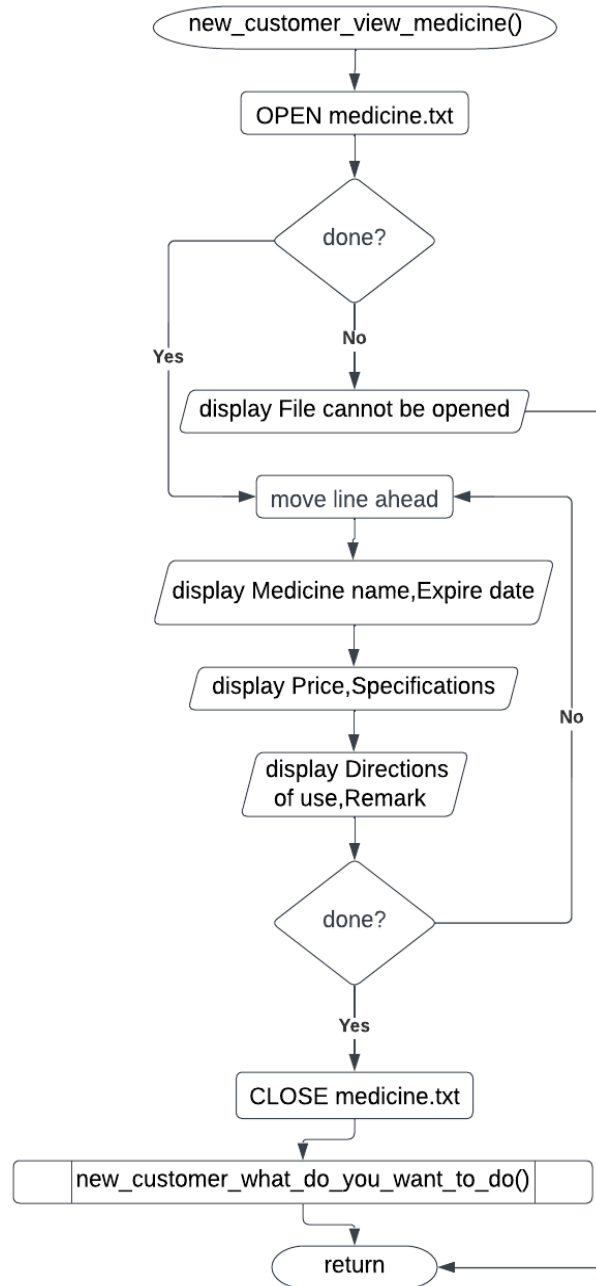
Flowchart:

Function: new_customer_view_medicine()**Pseudocode:**

```

1. new_customer_view_medicine()
2.     2.1 TRY
        2.1.1 OPEN medicine.txt
    2.2 EXCEPT
        2.2.1 display File cannot be opened
3.     3.1 FOR line IN medicine.txt
        3.1.1 display Medicine name,Expire date
        3.1.2 display Price,Specifications
        3.1.3 display Directions of use,Remark
4. CLOSE medicine.txt
5. CALL new_customer_what_do_you_want_to_do()
6. return

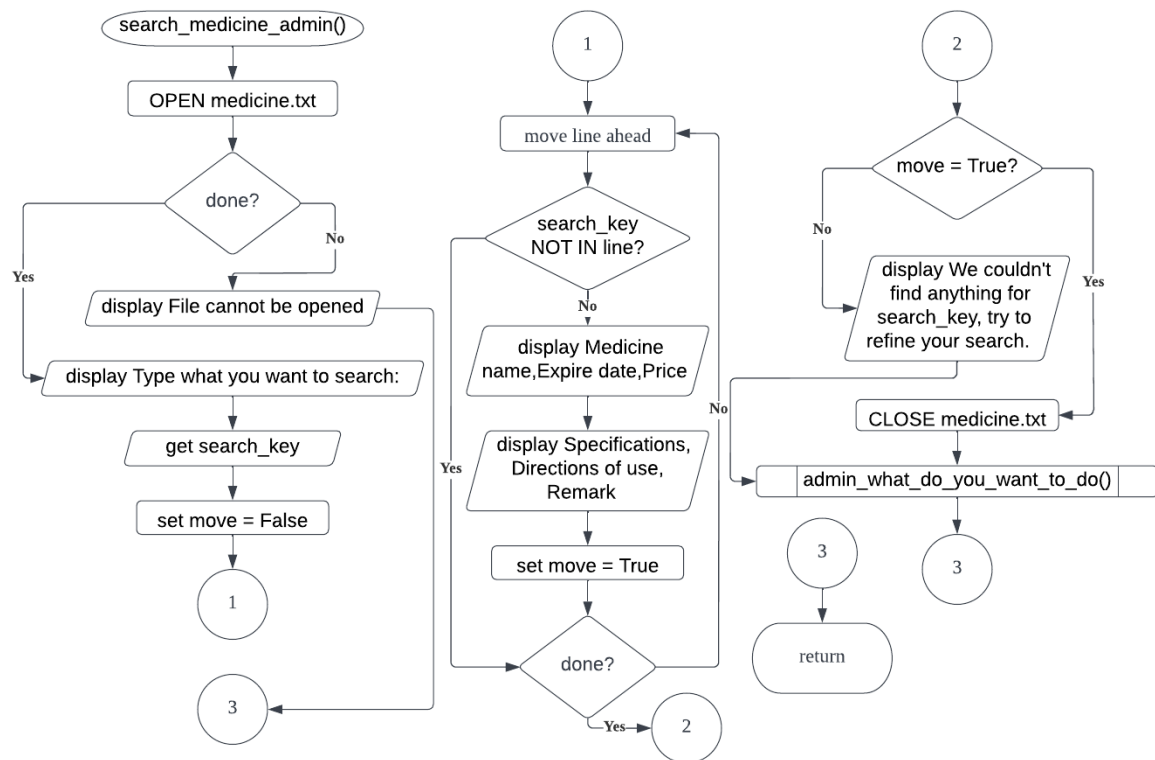
```

Flowchart:

Function: search_medicine_admin()**Pseudocode:**

1. search_medicine_admin()
2. 2.1 TRY
 - 2.1.1 OPEN medicine.txt
 - 2.2 EXCEPT
 - 2.2.1 display File cannot be opened
3. display Type what you want to search:
4. get search_key
5. set move = False
6. 6.1FOR line IN medicine.txt
 - 6.1.1 IF NOT search_key IN line
 - 6.1.1.1 CONTINUE
 - 6.1.2 ELSE
 - 6.1.2.1 display Medicine name,Expire date,Price
 - 6.1.2.2 display Specifications,Directions of use,Remark
 - 6.1.2.3 set move = True
 - 6.1.3 ENDIF
7. 7.1 IF move = True
 - 7.1.1 CONTINUE
 - 7.2 ELSE display We couldn't find anything for search_key, try to refine your search.
 - 7.3 ENDIF
8. CLOSE medicine.txt
9. CALL admin_what_do_you_want_to_do()
10. return

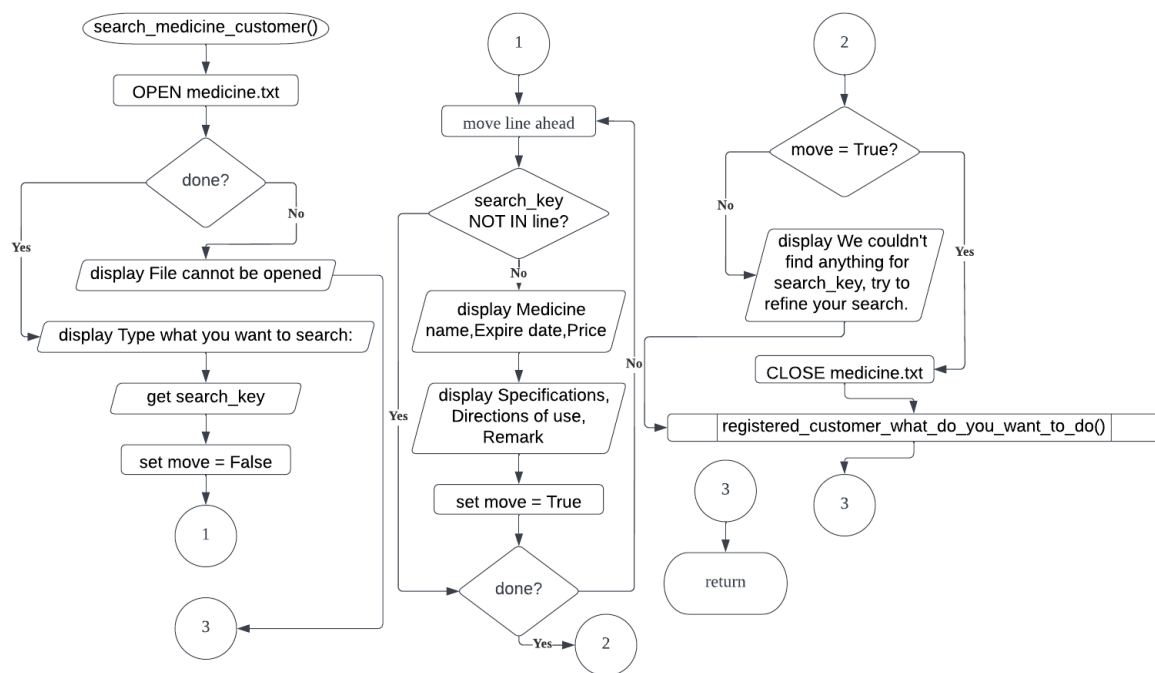
Flowchart:



Function: search_medicine_customer()**Pseudocode:**

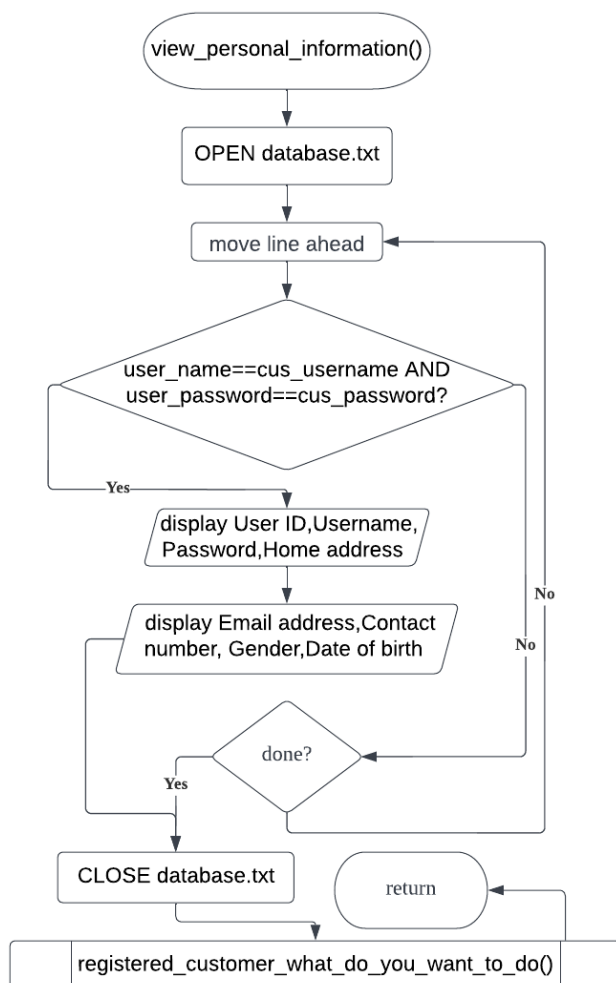
1. search_medicine_customer()
2. 2.1 TRY
 - 2.1.1 OPEN medicine.txt
 - 2.2 EXCEPT
 - 2.2.1 display File cannot be opened
3. display Type what you want to search:
4. get search_key
5. set move = False
6. 6.1FOR line IN medicine.txt
 - 6.1.1 IF NOT search_key IN line
 - 6.1.1.1 CONTINUE
 - 6.1.2 ELSE
 - 6.1.2.1 display Medicine name,Expire date,Price
 - 6.1.2.2 display Specifications,Directions of use,Remark
 - 6.1.2.3 set move = True
 - 6.1.3 ENDIF
7. 7.1 IF move = True
 - 7.1.1 CONTINUE
 - 7.2 ELSE display We couldn't find anything for search_key, try to refine your search.
 - 7.3 ENDIF
8. CLOSE medicine.txt
9. CALL registered_customer_what_do_you_want_to_do()
10. return

Flowchart:



Function: view_personal_information()**Pseudocode:**

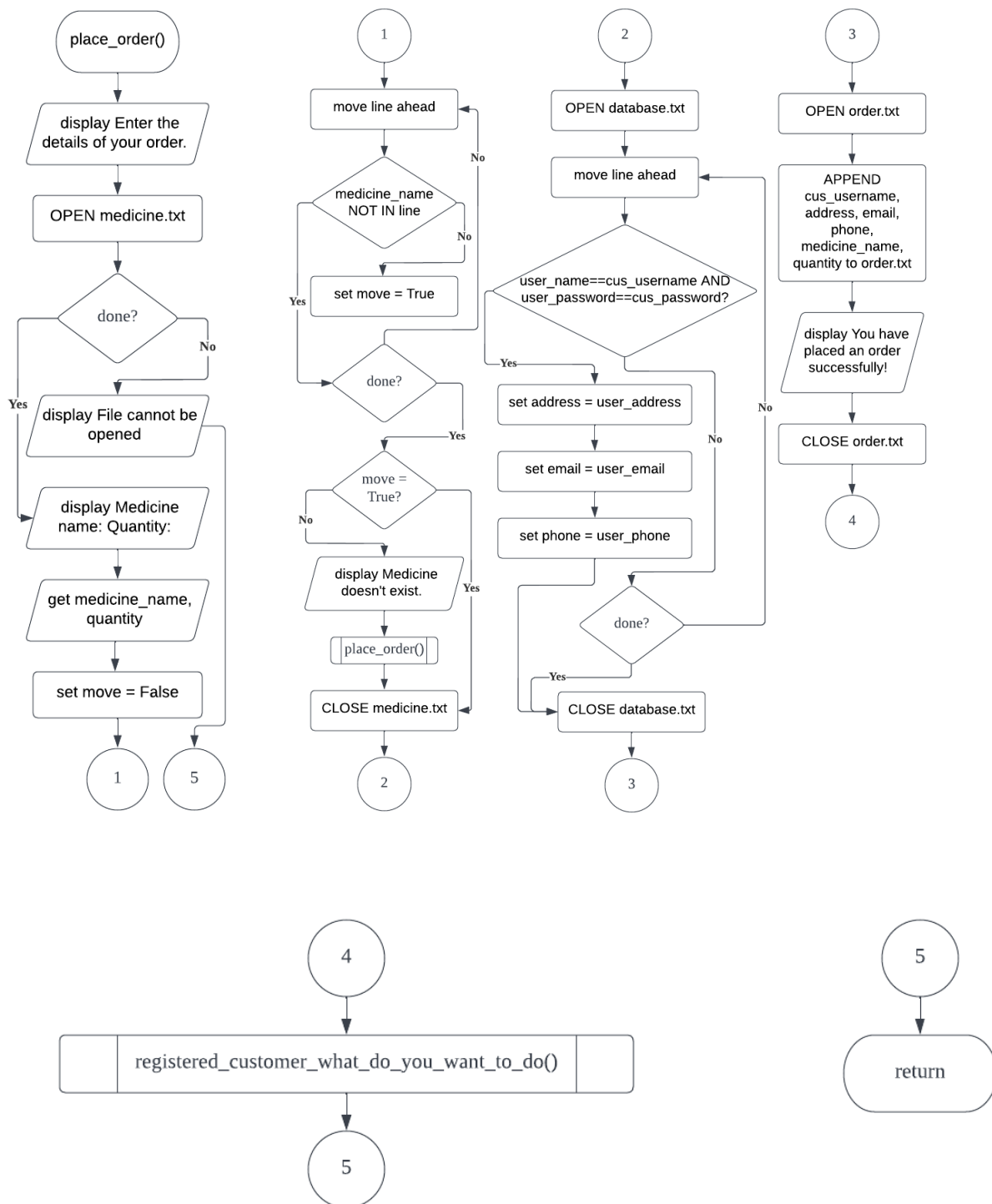
1. view_personal_information()
2. OPEN database.txt
3. 3.1 FOR line IN database.txt
 - 3.1.1 IF user_name==cus_username AND user_password==cus_password
 - 3.1.1.1 display User ID,Username,Password,Home address,
 - 3.1.1.2 display Email address,Contact number,Gender,Date of birth
 - 3.1.1.3 BREAK
 - 3.1.2 ELSE
 - 3.1.2.1 CONTINUE
- 3.1.3 ENDIF
4. CLOSE database.txt
5. CALL registered_customer_what_do_you_want_to_do()
6. return

Flowchart:

Function: place_order()**Pseudocode:**

```
1. place_order()
2. display Enter the details of your order.
3.     3.1 TRY
           3.1.1 OPEN medicine.txt
       3.2 EXCEPT
           3.2.1 display File cannot be opened
4. display Medicine name: Quantity:
5. get medicine_name,quantity
6. set move = False
7.     7.1 FOR line IN medicine.txt
           7.1.1 IF NOT medicine_name IN line
               7.1.1.1 CONTINUE
           7.1.2 ELSE
               7.1.2.1 set move = True
           7.1.3 ENDIF
8.     8.1 IF move = True
           8.1.1 CONTINUE
       8.2 ELSE
           8.2.1 display Medicine doesn't exist.
           8.2.2 CALL place_order()
       8.3 ENDIF
9. CLOSE medicine.txt
10. OPEN database.txt
11.     11.1 FOR line IN database.txt
           11.1.1 IF user_name==cus_username AND user_password==cus_password
               11.1.1.1 set address = user_address
               11.1.1.2 set email = user_email
               11.1.1.3 set phone = user_phone
               11.1.1.4 BREAK
           11.1.2 ELSE
               11.1.2.1 CONTINUE
           11.1.3 ENDIF
12. CLOSE database.txt
13. OPEN order.txt
14. APPEND cus_username,address,email,phone,medicine_name,quantity to order.txt
15. display You have placed an order successfully!
16. CLOSE order.txt
17. CALL registered_customer_what_do_you_want_to_do()
18. return
```

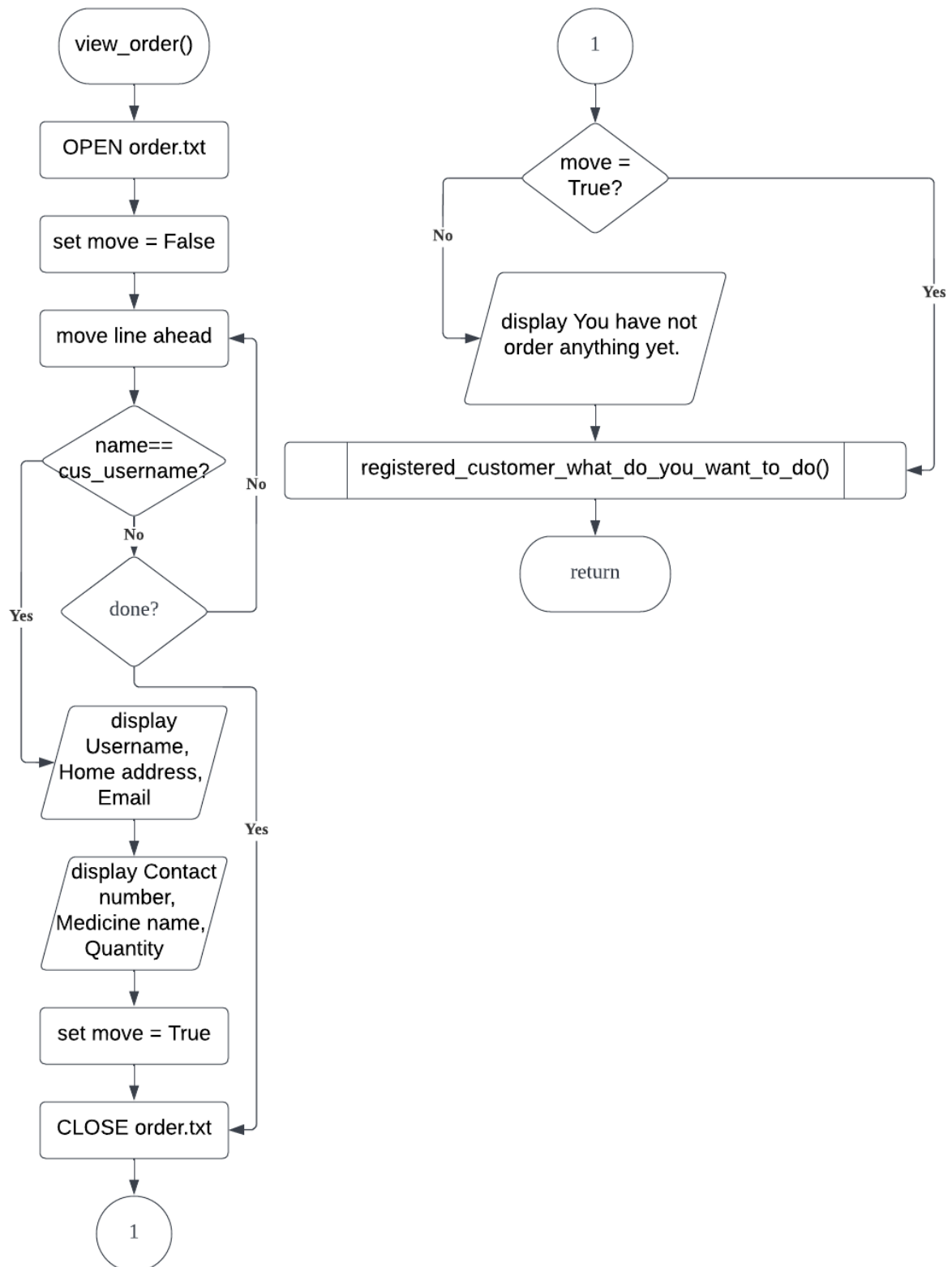
Flowchart:



Function: view_order()**Pseudocode:**

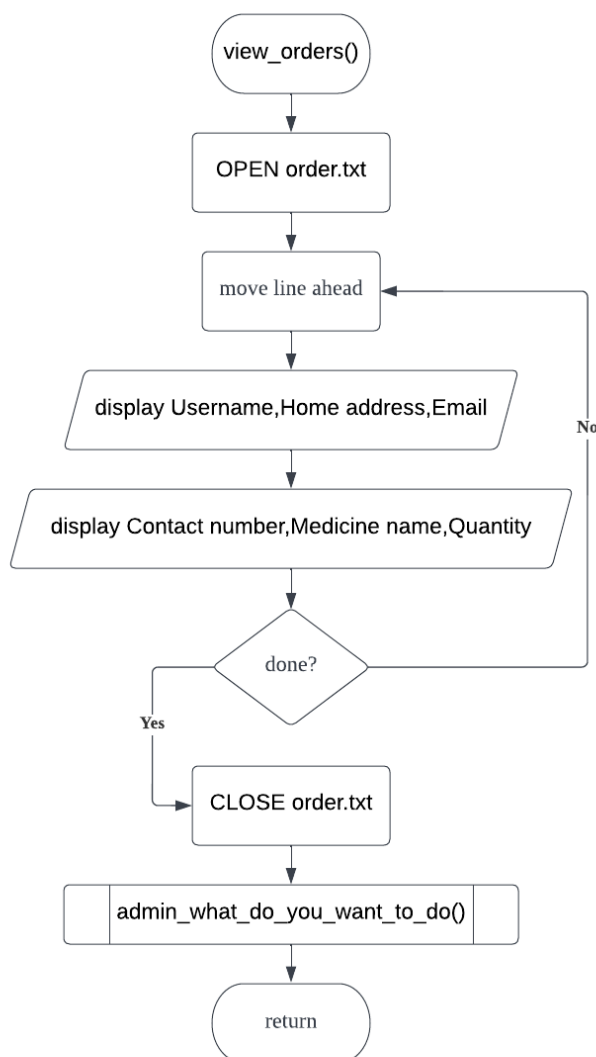
```
1. view_order()
2. OPEN order.txt
3. set move = False
4.     4.1 FOR line IN order.txt
           4.1.1 IF name==cus_username
                   4.1.1.1 display Username,Home address,Email
                   4.1.1.2 display Contact number,Medicine name,Quantity
                   4.1.1.3 set move=True
                   4.1.1.4 BREAK
           4.1.2 ELSE
                   4.1.2.1 CONTINUE
           4.1.3 ENDIF
5. CLOSE order.txt
6.     6.1 IF move = True
           6.1.1 CONTINUE
       6.2 ELSE
           6.2.1 display You have not order anything yet.
       6.3 ENDIF
7. CALL registered_customer_what_do_you_want_to_do()
8. return
```

Flowchart:



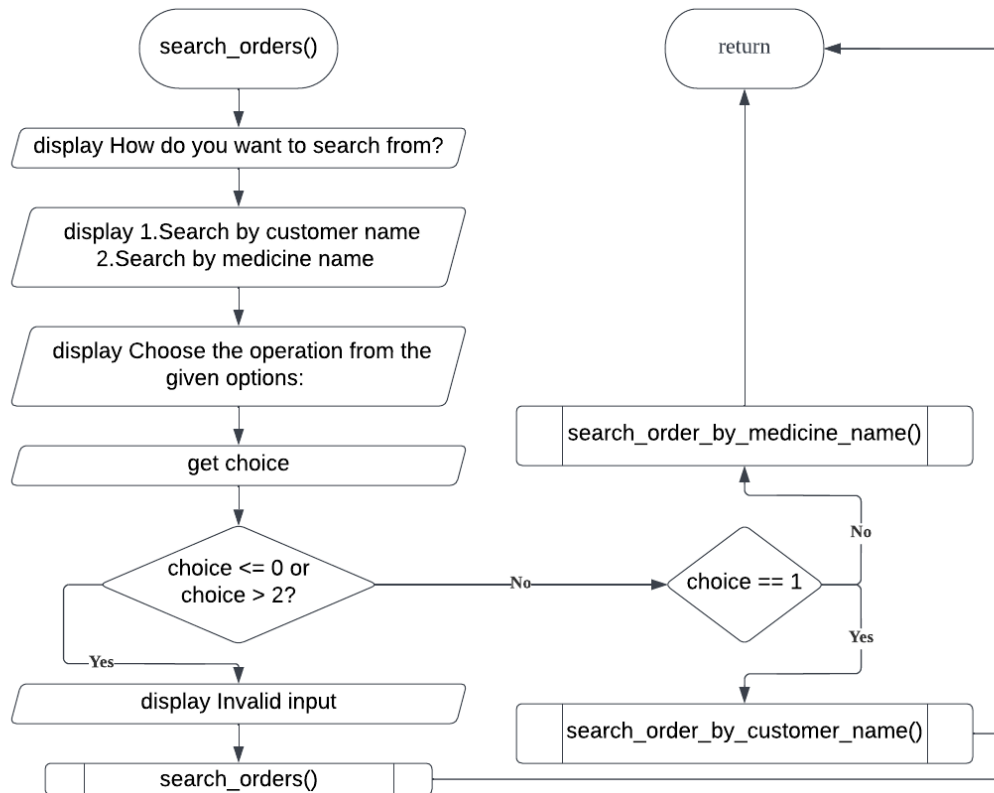
Function: view_orders()**Pseudocode:**

1. view_orders()
2. OPEN order.txt
3. 3.1 FOR line IN order.txt
 - 3.1.1 display Username,Home address,Email
 - 3.1.2 display Contact number,Medicine name,Quantity
4. CLOSE order.txt
5. CALL admin_what_do_you_want_to_do()
6. return

Flowchart:

Function: search_orders()**Pseudocode:**

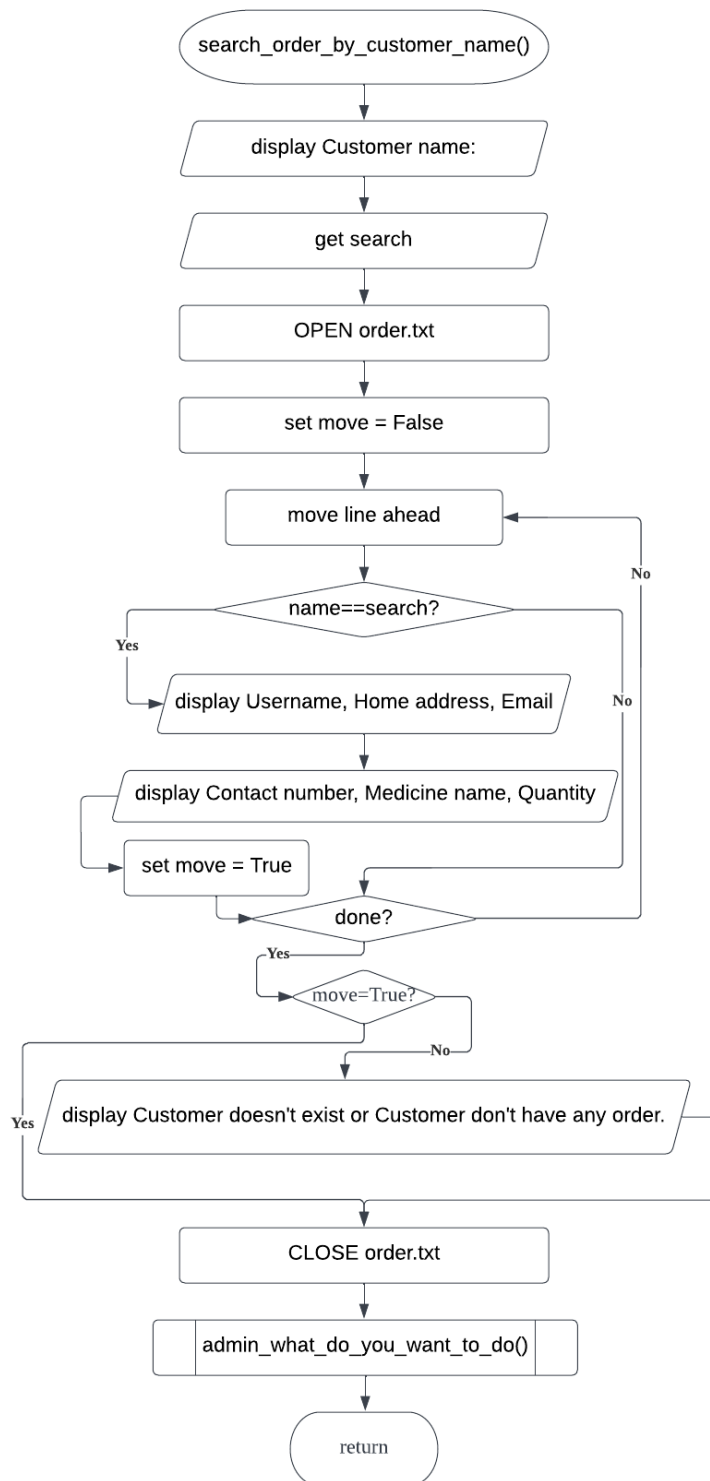
1. search_orders()
2. display How do you want to search from?
3. display 1.Search by customer name 2.Search by medicine name
4. display Choose the operation from the given options:
5. get choice
6.
 - 6.1 IF choice <= 0 or choice > 2
 - 6.1.1 display Invalid input
 - 6.1.2 CALL search_orders()
 - 6.2 ELSEIF choice == 1
 - 6.2.1 CALL search_order_by_customer_name()
 - 6.3 ELSE
 - 6.3.1 CALL search_order_by_medicine_name()
 - 6.4 ENDIF
7. return

Flowchart:

Function: search_order_by_customer_name()**Pseudocode:**

1. search_order_by_customer_name()
2. display Customer name:
3. get search
4. OPEN order.txt
5. set move = False
6. 6.1 FOR line IN order.txt
 - 6.1.1 IF name==search
 - 6.1.1.1 display Username,Home address,Email
 - 6.1.1.2 display Contact number,Medicine name,Quantity
 - 6.1.1.3 set move = True
 - 6.1.2 ELSE
 - 6.1.2.1 CONTINUE
 - 6.1.3 ENDIF
7. 7.1 IF move = True
 - 7.1.1 CONTINUE
- 7.2 ELSE
 - 7.2.1 display Customer doesn't exist or Customer don't have any order.
- 7.3 ENDIF
8. CLOSE order.txt
9. CALL admin_what_do_you_want_to_do()
10. return

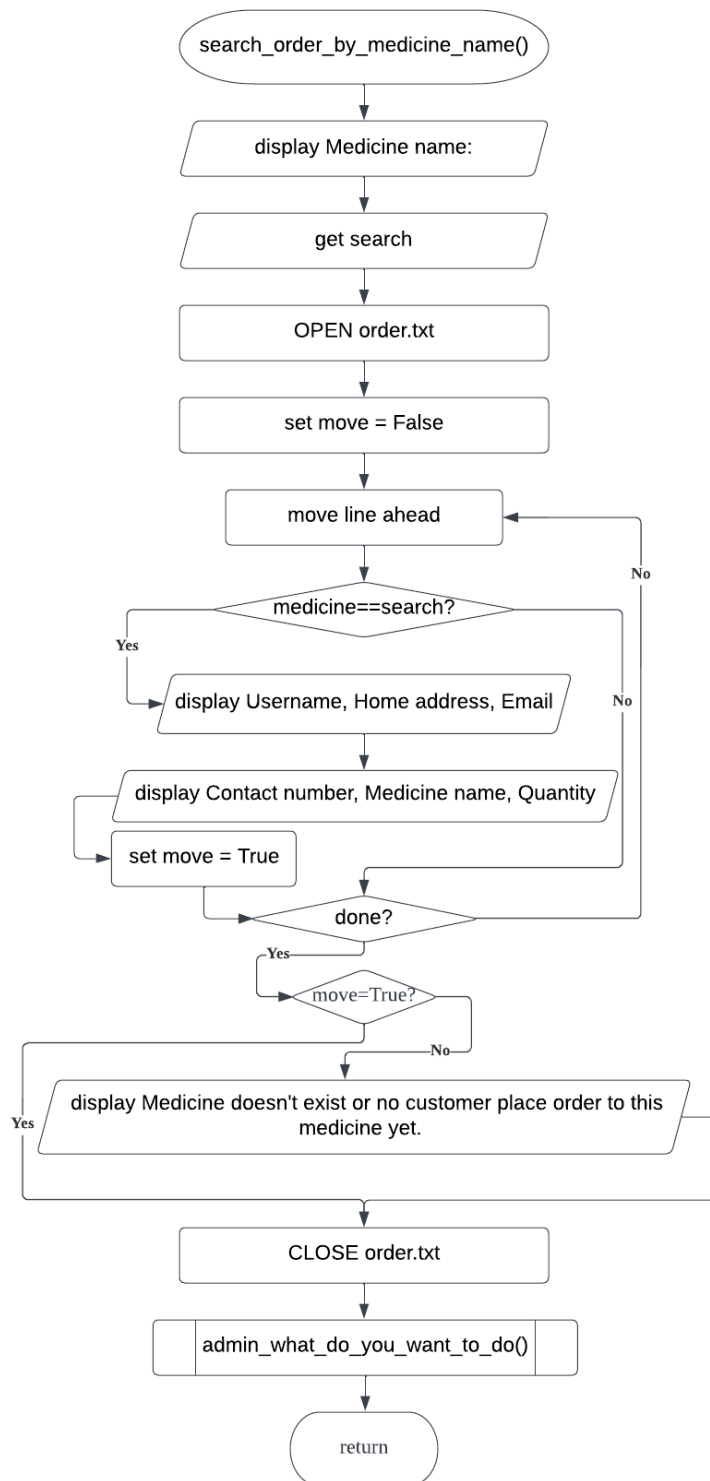
Flowchart:



Function: search_order_by_medicine_name()**Pseudocode:**

1. search_order_by_medicine_name()
2. display Medicine name:
3. get search
4. OPEN order.txt
5. set move = False
6. 6.1 FOR line IN order.txt
 - 6.1.1 IF medicine==search
 - 6.1.1.1 display Username,Home address,Email
 - 6.1.1.2 display Contact number,Medicine name,Quantity
 - 6.1.1.3 set move = True
 - 6.1.2 ELSE
 - 6.1.2.1 CONTINUE
 - 6.1.3 ENDIF
7. 7.1 IF move = True
 - 7.1.1 CONTINUE
- 7.2 ELSE
 - 7.2.1 display Medicine doesn't exist or no customer place order to this medicine yet.
- 7.3 ENDIF
8. CLOSE order.txt
9. CALL admin_what_do_you_want_to_do()
10. return

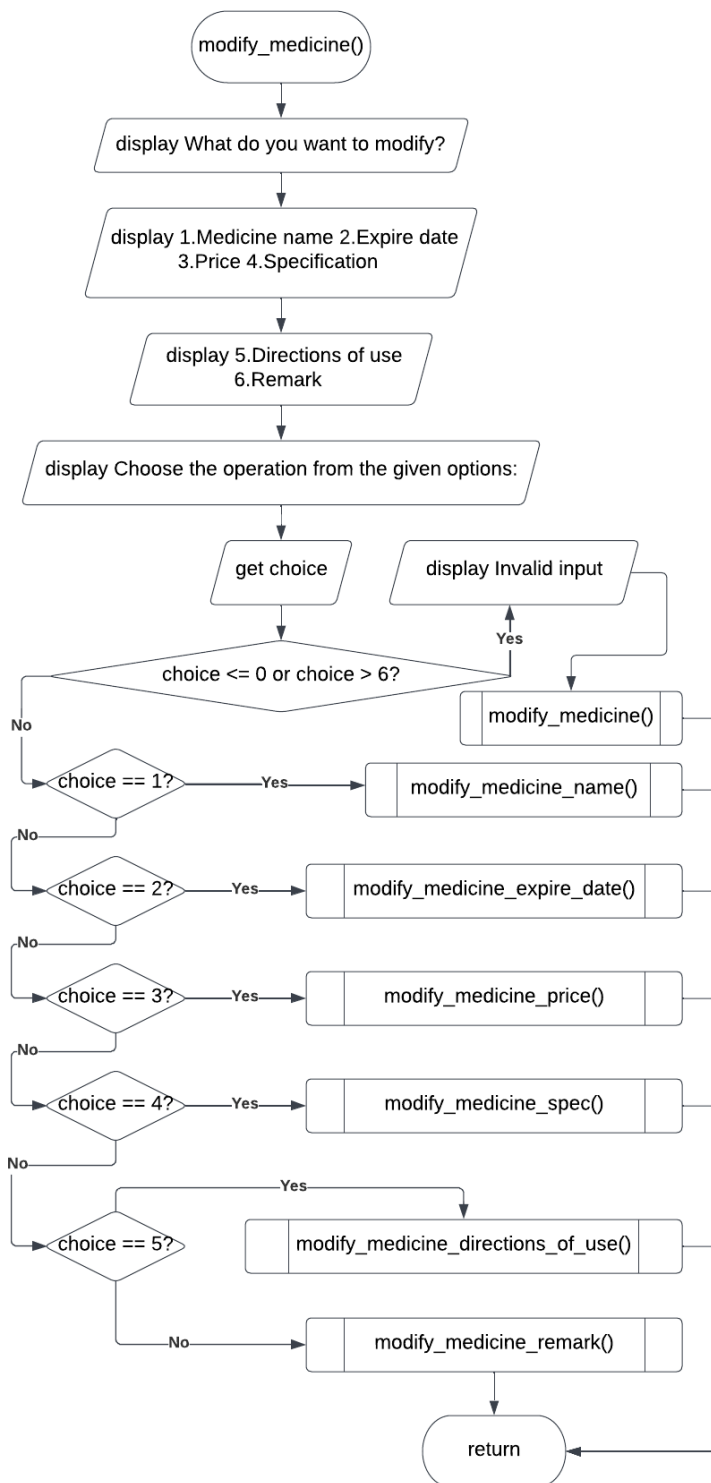
Flowchart:



Function: modify_medicine()**Pseudocode:**

1. modify_medicine()
2. display What do you want to modify?
3. display 1.Medicine name 2.Expire date 3.Price 4.Specification
4. display 5.Directions of use 6.Remark
5. display Choose the operation from the given options:
6. get choice
7.
 - 7.1 IF choice <= 0 or choice > 6
 - 7.1.1 display Invalid input
 - 7.1.2 CALL modify_medicine()
 - 7.2 ELSEIF choice == 1
 - 7.2.1 CALL modify_medicine_name()
 - 7.3 ELSEIF choice == 2
 - 7.3.1 CALL modify_medicine_expire_date()
 - 7.4 ELSEIF choice == 3
 - 7.4.1 CALL modify_medicine_price()
 - 7.5 ELSEIF choice == 4
 - 7.5.1 CALL modify_medicine_spec()
 - 7.6 ELSEIF choice == 5
 - 7.6.1 CALL modify_medicine_directions_of_use()
 - 7.7 ELSE
 - 7.7.1 CALL modify_medicine_remark()
 - 7.8 ENDIF
8. return

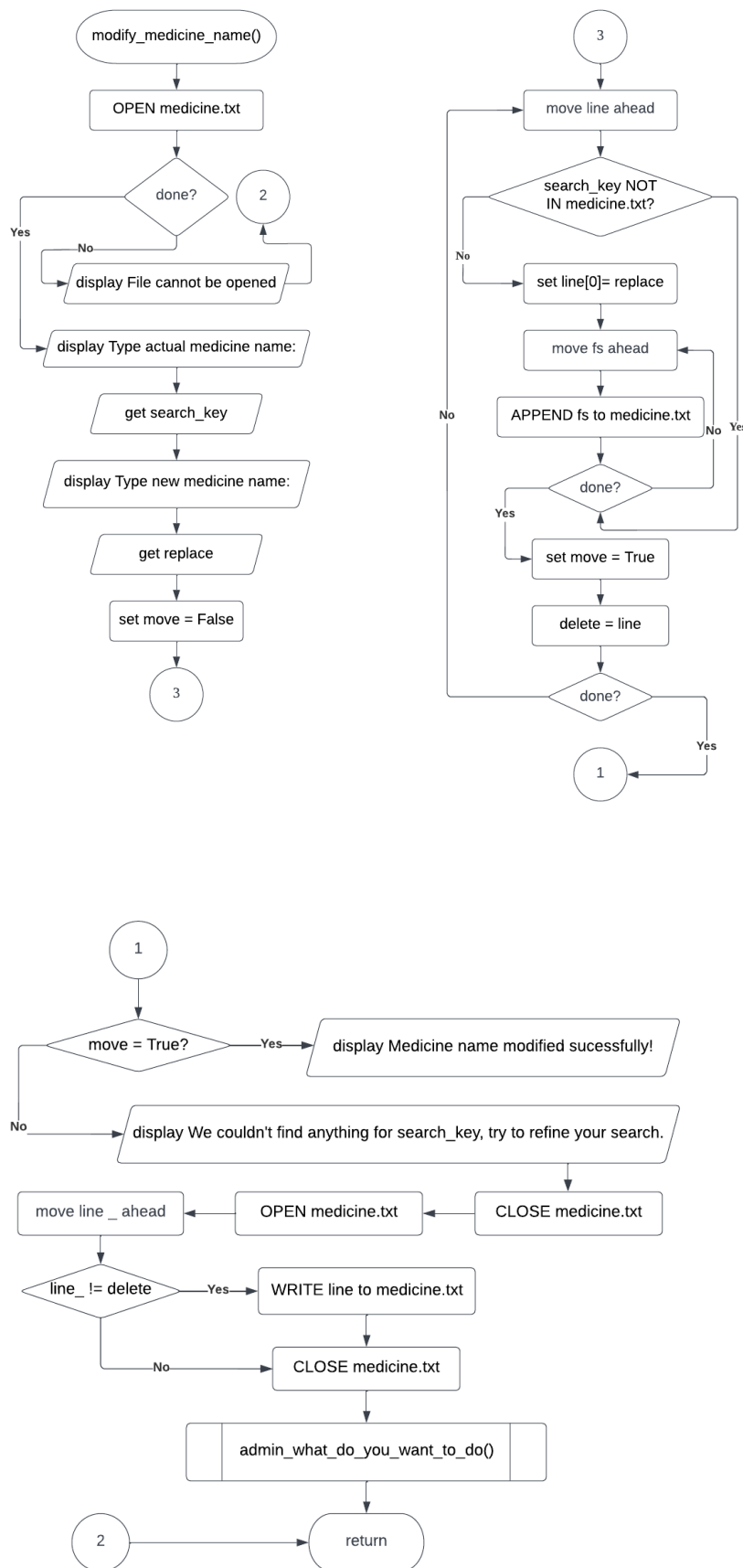
Flowchart:



Function: modify_medicine_name()**Pseudocode:**

1. modify_medicine_name()
2. 2.1 TRY
 - 2.1.1 OPEN medicine.txt
- 2.2 EXCEPT
 - 2.2.1 display File cannot be opened
3. display Type actual medicine name:
4. get search_key
5. display Type new medicine name:
6. get replace
7. set move = False
8. 8.1 FOR line IN medicine.txt
 - 8.1.1 IF NOT search_key in medicine.txt
 - 8.1.1.1 CONTINUE
 - 8.1.2 ELSE
 - 8.1.2.1 set line[0]= replace
 - 8.1.2.2 FOR fs IN line
 - 8.1.2.2.1 APPEND fs to medicine.txt
 - 8.1.2.3 set move = True
 - 8.1.2.4 set delete = line
 - 8.1.2.5 CONTINUE
 - 8.1.3 ENDIF
9. 9.1 IF move=True
 - 9.1.1 display Medicine name modified sucessfully!
- 9.2 ELSE
 - 9.2.1 display We couldn't find anything for search_key, try to refine your search.
- 9.3 ENDIF
10. CLOSE medicine.txt
11. OPEN medicine.txt
12. 12.1 FOR line_ IN medicine.txt
 - 12.1.1 IF line_ != delete
 - 12.1.1.1 WRITE line to medicine.txt
 - 12.1.2 ELSE
 - 12.1.2.1 CONTINUE
 - 12.1.3 ENDIF
13. CLOSE medicine.txt
14. CALL admin_what_do_you_want_to_do()
15. return

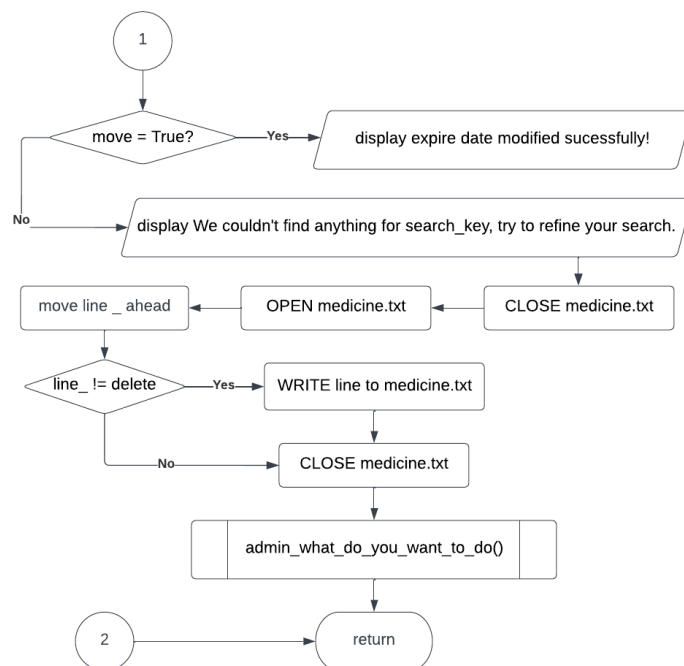
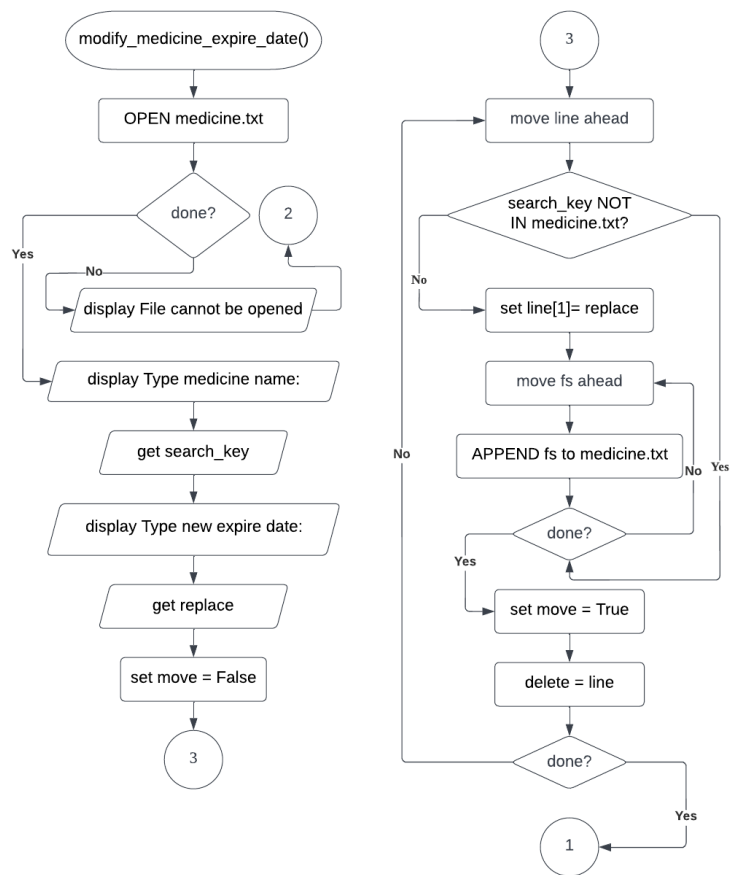
Flowchart:



Function: modify_medicine_expire_date()**Pseudocode:**

```
1. modify_medicine_expire_date()
2.     2.1 TRY
        2.1.1 OPEN medicine.txt
    2.2 EXCEPT
        2.2.1 display File cannot be opened
3. display Type medicine name:
4. get search_key
5. display Type new expire date:
6. get replace
7. set move = False
8.     8.1 FOR line IN medicine.txt
        8.1.1 IF NOT search_key in medicine.txt
            8.1.1.1 CONTINUE
        8.1.2 ELSE
            8.1.2.1 set line[1]= replace
            8.1.2.2 FOR fs IN line
                8.1.2.2.1 APPEND fs to medicine.txt
            8.1.2.3 set move = True
            8.1.2.4 set delete = line
            8.1.2.5 CONTINUE
        8.1.3 ENDIF
9.     9.1 IF move=True
        9.1.1 display expire date modified sucessfully!
    9.2 ELSE
        9.2.1 display We couldn't find anything for search_key, try to refine your search.
    9.3 ENDIF
10. CLOSE medicine.txt
11. OPEN medicine.txt
12.     12.1 FOR line_ IN medicine.txt
        12.1.1 IF line_ != delete
            12.1.1.1 WRITE line to medicine.txt
        12.1.2 ELSE
            12.1.2.1 CONTINUE
        12.1.3 ENDIF
13. CLOSE medicine.txt
14. CALL admin_what_do_you_want_to_do()
15. return
```

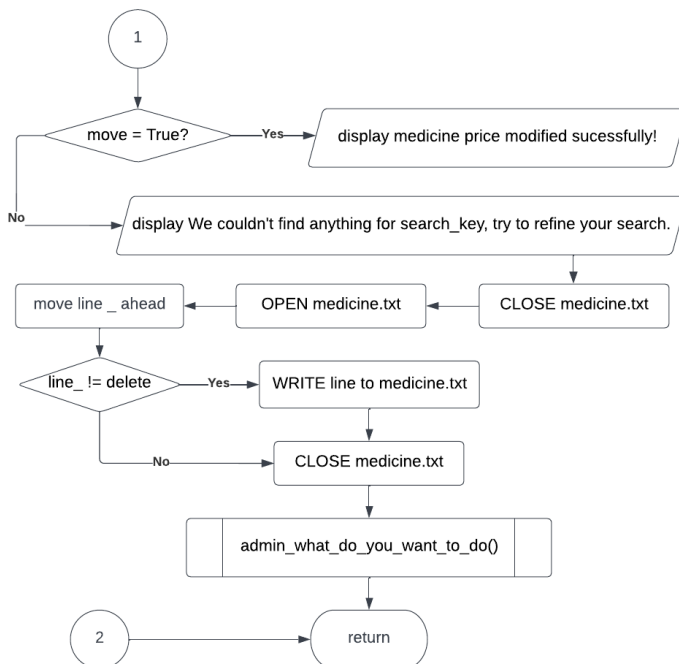
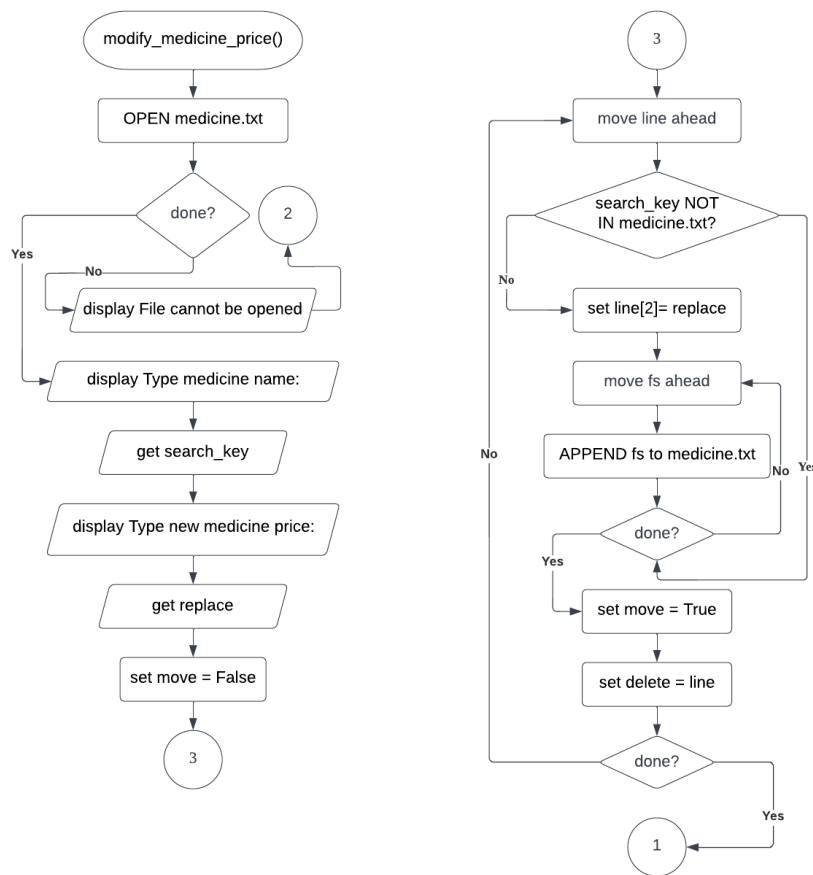
Flowchart:



Function: modify_medicine_price()**Pseudocode:**

1. modify_medicine_price()
2. 2.1 TRY
 - 2.1.1 OPEN medicine.txt
- 2.2 EXCEPT
 - 2.2.1 display File cannot be opened
3. display Type medicine name:
4. get search_key
5. display Type new medicine price:
6. get replace
7. set move = False
8. 8.1 FOR line IN medicine.txt
 - 8.1.1 IF NOT search_key in medicine.txt
 - 8.1.1.1 CONTINUE
 - 8.1.2 ELSE
 - 8.1.2.1 set line[2]= replace
 - 8.1.2.2 FOR fs IN line
 - 8.1.2.2.1 APPEND fs to medicine.txt
 - 8.1.2.3 set move = True
 - 8.1.2.4 set delete = line
 - 8.1.2.5 CONTINUE
 - 8.1.3 ENDIF
9. 9.1 IF move=True
 - 9.1.1 display medicine price modified sucessfully!
- 9.2 ELSE
 - 9.2.1 display We couldn't find anything for search_key, try to refine your search.
- 9.3 ENDIF
10. CLOSE medicine.txt
11. OPEN medicine.txt
12. 12.1 FOR line_ IN medicine.txt
 - 12.1.1 IF line_ != delete
 - 12.1.1.1 WRITE line to medicine.txt
 - 12.1.2 ELSE
 - 12.1.2.1 CONTINUE
 - 12.1.3 ENDIF
13. CLOSE medicine.txt
14. CALL admin_what_do_you_want_to_do()
15. return

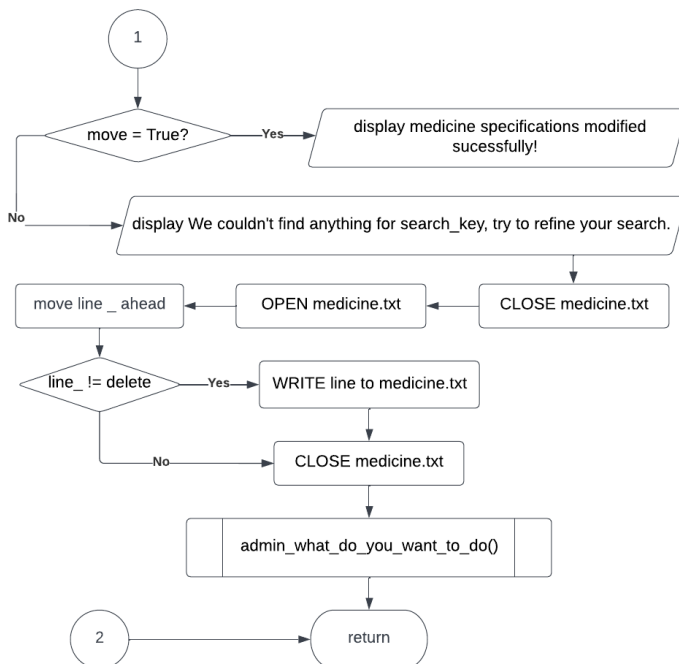
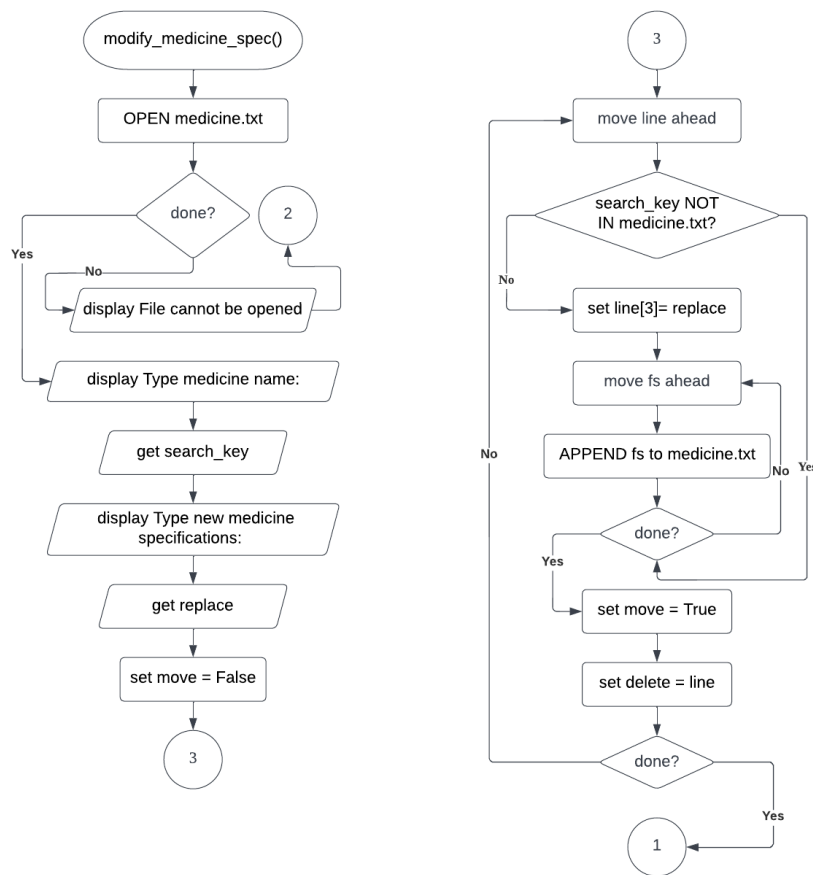
Flowchart:



Function: modify_medicine_spec()**Pseudocode:**

1. modify_medicine_spec()
2. 2.1 TRY
 - 2.1.1 OPEN medicine.txt
- 2.2 EXCEPT
 - 2.2.1 display File cannot be opened
3. display Type medicine name:
4. get search_key
5. display Type new medicine specifications:
6. get replace
7. set move = False
8. 8.1 FOR line IN medicine.txt
 - 8.1.1 IF NOT search_key in medicine.txt
 - 8.1.1.1 CONTINUE
 - 8.1.2 ELSE
 - 8.1.2.1 set line[3]= replace
 - 8.1.2.2 FOR fs IN line
 - 8.1.2.2.1 APPEND fs to medicine.txt
 - 8.1.2.3 set move = True
 - 8.1.2.4 set delete = line
 - 8.1.2.5 CONTINUE
 - 8.1.3 ENDIF
9. 9.1 IF move=True
 - 9.1.1 display medicine specifications modified sucessfully!
- 9.2 ELSE
 - 9.2.1 display We couldn't find anything for search_key, try to refine your search.
- 9.3 ENDIF
10. CLOSE medicine.txt
11. OPEN medicine.txt
12. 12.1 FOR line_ IN medicine.txt
 - 12.1.1 IF line_ != delete
 - 12.1.1.1 WRITE line to medicine.txt
 - 12.1.2 ELSE
 - 12.1.2.1 CONTINUE
 - 12.1.3 ENDIF
13. CLOSE medicine.txt
14. CALL admin_what_do_you_want_to_do()
15. return

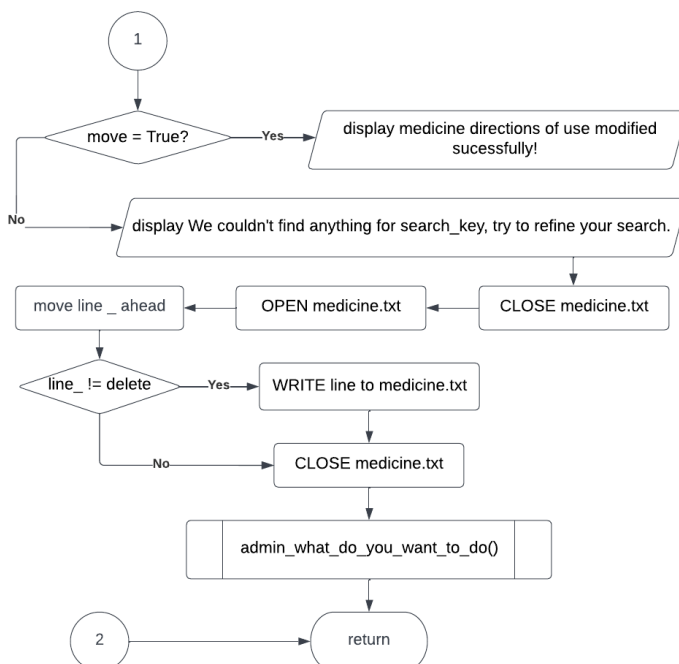
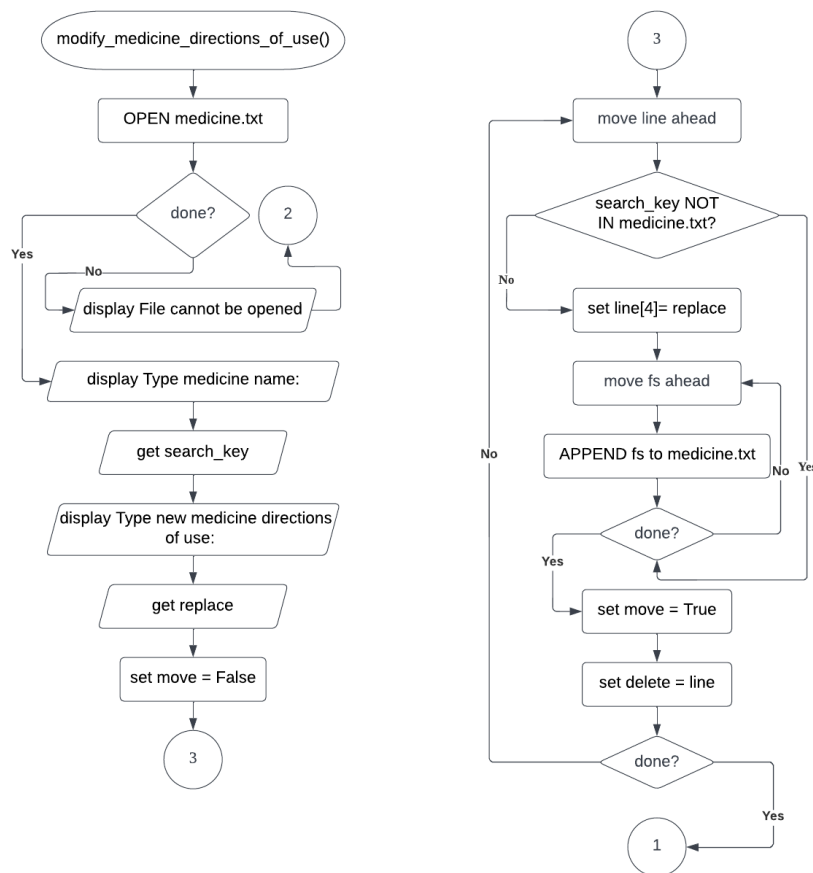
Flowchart:



Function: modify_medicine_directions_of_use()**Pseudocode:**

```
1. modify_medicine_directions_of_use()
2.     2.1 TRY
        2.1.1 OPEN medicine.txt
    2.2 EXCEPT
        2.2.1 display File cannot be opened
3. display Type medicine name:
4. get search_key
5. display Type new medicine directions of use:
6. get replace
7. set move = False
8.     8.1 FOR line IN medicine.txt
        8.1.1 IF NOT search_key in medicine.txt
            8.1.1.1 CONTINUE
        8.1.2 ELSE
            8.1.2.1 set line[4]= replace
            8.1.2.2 FOR fs IN line
                8.1.2.2.1 APPEND fs to medicine.txt
            8.1.2.3 set move = True
            8.1.2.4 set delete = line
            8.1.2.5 CONTINUE
        8.1.3 ENDIF
9.     9.1 IF move=True
        9.1.1 display medicine directions of use modified sucessfully!
    9.2 ELSE
        9.2.1 display We couldn't find anything for search_key, try to refine your search.
    9.3 ENDIF
10. CLOSE medicine.txt
11. OPEN medicine.txt
12.     12.1 FOR line_ IN medicine.txt
        12.1.1 IF line_ != delete
            12.1.1.1 WRITE line to medicine.txt
        12.1.2 ELSE
            12.1.2.1 CONTINUE
        12.1.3 ENDIF
13. CLOSE medicine.txt
14. CALL admin_what_do_you_want_to_do()
15. return
```

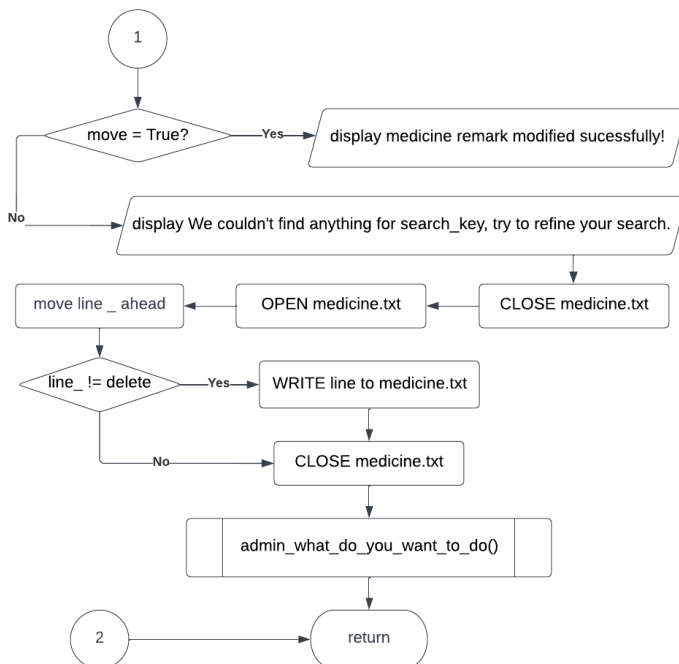
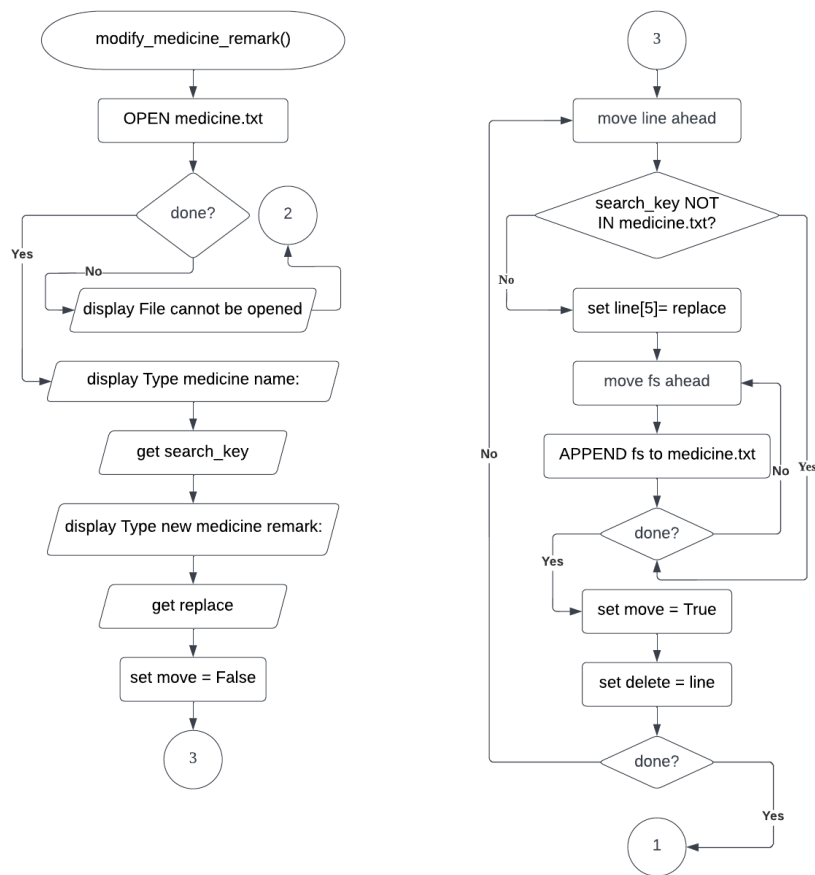
Flowchart:



Function: modify_medicine_remark()**Pseudocode:**

```
1. modify_medicine_remark()
2.     2.1 TRY
           2.1.1 OPEN medicine.txt
       2.2 EXCEPT
           2.2.1 display File cannot be opened
3. display Type medicine name:
4. get search_key
5. display Type new medicine remark:
6. get replace
7. set move = False
8.     8.1 FOR line IN medicine.txt
           8.1.1 IF NOT search_key in medicine.txt
               8.1.1.1 CONTINUE
           8.1.2 ELSE
               8.1.2.1 set line[5]= replace
               8.1.2.2 FOR fs IN line
                   8.1.2.2.1 APPEND fs to medicine.txt
               8.1.2.3 set move = True
               8.1.2.4 set delete = line
               8.1.2.5 CONTINUE
           8.1.3 ENDIF
9.     9.1 IF move=True
           9.1.1 display medicine remark modified sucessfully!
       9.2 ELSE
           9.2.1 display We couldn't find anything for search_key, try to refine your search.
       9.3 ENDIF
10. CLOSE medicine.txt
11. OPEN medicine.txt
12.     12.1 FOR line_ IN medicine.txt
           12.1.1 IF line_ != delete
               12.1.1.1 WRITE line to medicine.txt
           12.1.2 ELSE
               12.1.2.1 CONTINUE
           12.1.3 ENDIF
13. CLOSE medicine.txt
14. CALL admin_what_do_you_want_to_do()
15. return
```

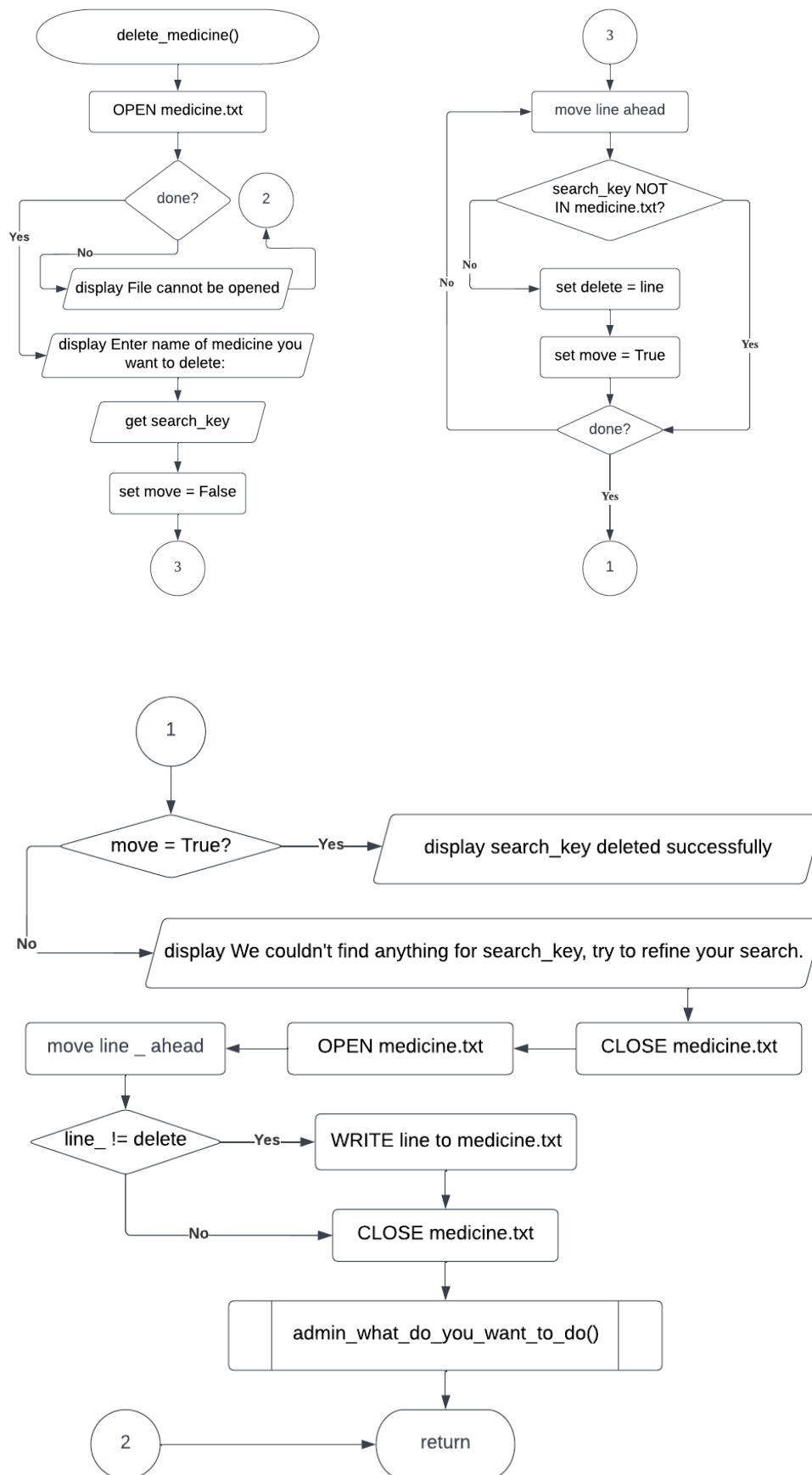
Flowchart:



Function: delete_medicine()**Pseudocode:**

```
1. delete_medicine()
2.     2.1 TRY
           2.1.1 OPEN medicine.txt
       2.2 EXCEPT
           2.2.1 display File cannot be opened
3. display Enter name of medicine you want to delete:
4. get search_key
5. set move = False
6.     6.1 FOR line IN medicine.txt
           6.1.1 IF NOT search_key in medicine.txt
               6.1.1.1 CONTINUE
           6.1.2 ELSE
               6.1.2.1 set delete = line
               6.1.2.2 set move = True
               6.1.2.3 BREAK
           6.1.3 ENDIF
9.     9.1 IF move=True
           9.1.1 display search_key deleted successfully
       9.2 ELSE
           9.2.1 display We couldn't find anything for search_key, try to refine your search.
       9.3 ENDIF
10. CLOSE medicine.txt
11. OPEN medicine.txt
12.     12.1 FOR line_ IN medicine.txt
           12.1.1 IF line_ != delete
               12.1.1.1 WRITE line to medicine.txt
           12.1.2 ELSE
               12.1.2.1 CONTINUE
           12.1.3 ENDIF
13. CLOSE medicine.txt
14. CALL admin_what_do_you_want_to_do()
15. return
```

Flowchart:



Program source code and explanation

```
956 menu()
```

The line 956 is to call the define function menu()

Function: menu()

```
938 ''' ----- '''
939 # This function is the menu to let the system know who is the user
940 def menu():
941     print('\n\nWho are you?')
942     print('\n\t1. Admin \n\t2. New Customer \n\t3. Registered Customer \n')
943     choice = int(input('\nChoose the operation from the given options: '))
944     # User can choose who are them here
945
946     if choice <= 0 or choice > 3:
947         print('\nInvalid input')
948         menu()
949     elif choice == 1:
950         admin_login()
951     elif choice == 2:
952         new_customer_what_do_you_want_to_do()
953     else:
954         customer_login()
955 ''' ----- '''
```

The line 941 is to prompt the user to think “Who they are?” and “What are they going to do by using OPMS?”. The line 942 is to provide several option to the user. The line 943 is to prompt the user to provide their choice to the system. Since that there are only three options for the user to choose, in line 946-948, if the user’s choice is lesser or equal to zero, or greater than three, the system will display “Invalid input” and call the function menu(). In line 949-954, the system will call the function admin_login() if the user’s choice is equal to 1 where the option is “Admin”, the system will call the function new_customer_what_do_you_want_to_do() if the user’s choice is equal to 2 where the option is “New Customer”, the system will call the function customer_login() when their choice is equal to 3 where the option is “Registered Customer”.

Function: new_customer_what_do_you_want_to_do()

```
918 '''
919 # This function is for new customer to choose what operation they want to do
920 def new_customer_what_do_you_want_to_do():
921     print('\nWhat do you want to do by using OPMS?')
922     print('\n\t1. View all medicine')
923     print('\t2. Register')
924     print('\t3. Menu')
925     choice = int(input('\nChoose the operation from the given options: '))
926     # Customer can choose what they want to do by using the system without logging in or to register an account
927
928     if choice <=0 or choice > 3:
929         print('Invalid input')
930         new_customer_what_do_you_want_to_do()
931     else:
932         if choice == 1:
933             new_customer_view_medicine()
934         elif choice == 2:
935             register()
936         else:
937             menu()
938 '''
```

The line 921 is to prompt the user to think “What are they going to do by using OPMS?”. The line 922-924 is to provide several option to the user. The line 925 is to prompt the user to provide their choice to the system. Since that there are only three options for the user to choose, in line 928-930, if the user’s choice is lesser or equal to zero, or greater than three, the system will display “Invalid input” and call the function new_customer_what_do_you_want_to_do(). In line 931-937, the system will call the function new_customer_view_medicine() if the user’s choice is equal to 1 where the option is “View all medicine”, the system will call the function register() if the user’s choice is equal to 2 where the option is “Register”, the system will call the function menu() when their choice is equal to 3 where the option is “Menu”.

Function: register()

```

867
868 # This function is for new customer to register themselves into the system
869 def register():
870     database = open('database.txt', 'r') # This is to open the database and so that the system can count how many cu
871
872     print('\nCreate Your OPMS Account \n')
873
874     count = 0
875     for line in database:
876         count = count + 1
877         Userid = count + 1
878         userid = str(Userid)
879
880     cus_username = str(input('\tUsername: '))
881     cus_password = str(input('\tPassword: '))
882     password = str(input('\tConfirm password: '))
883     address = str(input('\tHome address: '))
884     email = str(input('\tEmail address: '))
885     phone = str(input('\tContact number: '))
886     gender = str(input('\tGender (F/M): '))
887     dob = str(input('\tDate of birth (DD/MM/YYYY): '))
888
889     database = open('database.txt', 'r')
890     for line in database:
891         user_id, user_name, user_password, user_address, user_email, user_phone, user_gender, user_dob = line.split(' ')
892         if (user_name == cus_username):
893             print('\nThat username is taken. Try another.')
894             register()
895             break
896         else:
897             continue
898     database.close()
899     # This is to open the database and so that the system can check weather the current registering username had b
900
901     if cus_password != password:
902         print("\nBoth passwords didn't match. Try again.")
903         register()
904     elif len(password) < 6:
905         print('\nSorry, your password must be at least 6 characters long. Try again.')
906         register()
907     else:
908
909         database = open('database.txt', 'a')
910         database.write(userid + ' ' + cus_username + ' ' + password + ' ' + address + ' ' + email + ' ' + phone + ' ' + gender + ' ' + dob + '\n')
911         print('\nYou have created an account successfully!')
912         print('\n'+cus_username+', your user ID is', userid+'.')
913         print('\nPlease login from registered customer.')
914         database.close()
915         # These is to check weather the details given by the customer is valid
916
917         menu()
918

```

The line 870 is to open and allow the system to read the database text file. The line 872 is to prompt the user to create their own account below. The line 874-878 is to calculate the current registering user's user ID. It set the variable count equal to zero at the beginning, the count would be added one for each line in the database text file. Now the count is equal to the number of user recorded in the database text file, so the user ID is count plus one. The line 880-887 is to prompt the user to provide their details. The line 889 is to open and allow the system to read the database text file. The line 890-897 is to make sure that the username had not been registered before in the system. The line 898 is to close the database text file. The line 902-907 is to make sure the password and confirm password typed by the user is matched and also the password has at least 6 characters, if not, the system would call the function register(). The line 909-915 is to append the details provided by the user into the database text file and also prompt the user that they have created an account successfully.

Function: admin_login()

The line 847-849 is to prompt the user to enter the username and password.

The line 852 is to let the system know that the process of login had not been done. The line 853-859 is to let the system to open the admin text file and make sure that the username and password provided by the user is correct. The line 857 is to let the system

```

843 '''
844 # This function is for admin to login to the system
845 ## assumptions: username-admin, password-admin
846 def admin_login():
847     print('\nPlease enter your username and password.')
848     admin_username = str(input('\n\tUsername: '))
849     admin_password = str(input('\tPassword: '))
850     # Admin can provide their details here and login into the system.
851
852     move = False
853     file = open('admin.txt', 'r')
854     for line in file:
855         user_name, user_password = line.split(',')
856         if (user_name == admin_username and user_password == admin_password):
857             move = True
858             break
859     file.close()
860
861     if(move):
862         print('\nLogin Successful!!!')
863         admin_what_do_you_want_to_do()
864     else:
865         print('\nWrong username or password')
866         admin_login()
867 '''

```

know that the process of login had done. In line 861-863, if the process of login had done, it will show “Login Successful!!!” and call the function admin_what_do_you_want_to_do(). However, in line 864-866, if the process of login had not been done, it will show “Wrong username or password” and call the function admin_login().

Function: customer_login()

```

817 '''
818 # This function is for registered customer to login to the system
819 ## assumptions: username-Admin, password-12345678
820 def customer_login():
821     global cus_username
822     global cus_password
823     print('\nPlease enter your username and password.')
824     cus_username = str(input('\n\tUsername: '))
825     cus_password = str(input('\tPassword: '))
826     # Customer can provide their details here and login into the system.
827
828     move = False
829     file = open('database.txt', 'r')
830     for line in file:
831         user_id, user_name, user_password, user_address, user_email, user_phone, user_gender, user_dob = line.split(',')
832         if (user_name == cus_username and user_password == cus_password):
833             move = True
834             break
835     file.close()
836
837     if(move):
838         print('\nLogin Successful!!!')
839         registered_customer_what_do_you_want_to_do()
840     else:
841         print('\nWrong username or password')
842         customer_login()
843 '''

```

The line 821-822 is to share the variable to the whole Python code but not only the function customer_login(). The line 823-826 is to prompt the user to enter the username and password. The line 828 is to let the system know that the process of login had not been done. The line 829-835 is to let the system to open the database text file and make sure that the username and password provided by the user is correct. The line 833 is to let the system know that the process of login had done. In line 837-839, the process of login had done and it shows “Login Successful!!!” and call the function registered_customer_what_do_you_want_to_do(). However, in line 840-842, if the process of login had not been done, it will show “Wrong username or password” and call the function customer_login().

Function: admin_what_do_you_want_to_do()

```

781 '''
782 # This function is for admin to choose what operation they want to do
783 def admin_what_do_you_want_to_do():
784     print('\n\nWhat do you want to do by using OPMS?')
785     print('\n\t1. Upload medicine detail in system')
786     print('\n\t2. View all medicine')
787     print('\n\t3. Modify medicine information')
788     print('\n\t4. Delete medicine')
789     print('\n\t5. Search medicine')
790     print('\n\t6. View orders of customers')
791     print('\n\t7. Search orders of customers')
792     print('\n\t8. Logout')
793     choice = int(input('\nChoose the operation from the given options: '))
794     # Admin can choose what they want to do using this system
795
796     if choice <=0 or choice > 8:
797         print('Invalid input')
798         admin_what_do_you_want_to_do()
799     else:
800         if choice == 1:
801             upload_medicine()
802         elif choice == 2:
803             admin_view_medicine()
804         elif choice == 3:
805             modify_medicine()
806         elif choice == 4:
807             delete_medicine()
808         elif choice == 5:
809             search_medicine_admin()
810         elif choice == 6:
811             view_orders()
812         elif choice == 7:
813             search_orders()
814         else:
815             menu()
816
817 '''

```

The line 784 is to prompt the user to think “What are they going to do by using OPMS?”. The line 785-792 is to provide several option to the user. The line 793 is to prompt the user to provide their choice to the system. Since that there are only eight options for the user to choose, in line 796-798, if the user’s choice is lesser or equal to zero, or greater than eight, the system will display “Invalid input” and call the function `admin_what_do_you_want_to_do()`. In line 799-815, the system will call the function `upload_medicine()` if the user’s choice is equal to 1 where the option is “Upload medicine detail in system”, the function `admin_view_medicine()` if the user’s choice is equal to 2 where the option is “View all medicine”, the function `modify_medicine()` if the user’s choice is equal to 3 where the option is “Modify medicine information”, the function `delete_medicine()` if the user’s choice is equal to 4 where the option is “Delete medicine”, the function `search_medicine_admin()` if the user’s choice is equal to 5 where the option is “Search medicine”, the function `view_orders()` if the user’s choice is equal to 6 where the option is “View orders of customers”, the function `search_orders()` if the user’s choice is equal to 7 where the option is “Search orders of customers” and the function `menu()` if the user’s choice is equal to 8 where the option is “Logout”.

Function: registered_customer_what_do_you_want_to_do()

```

752 '''
753 # This function is for registered customer to choose what operation they want to do
754 def registered_customer_what_do_you_want_to_do():
755     print('\n\nWhat do you want to do by using OPMS?')
756     print('\n\t1. View all medicine')
757     print('\n\t2. Place order')
758     print('\n\t3. View order')
759     print('\n\t4. View Personal Information')
760     print('\n\t5. Search medicine')
761     print('\n\t6. Logout')
762     choice = int(input('\nChoose the operation from the given options: '))
763     # Customer can choose what they want to do using this system
764
765     if choice <=0 or choice > 6:
766         print('Invalid input')
767         registered_customer_what_do_you_want_to_do()
768     else:
769         if choice == 1:
770             reg_customer_view_medicine()
771         elif choice == 2:
772             place_order()
773         elif choice == 3:
774             view_order()
775         elif choice == 4:
776             view_personal_information()
777         elif choice == 5:
778             search_medicine_customer()
779         else:
780             menu()
781 '''

```

The line 755 is to prompt the user to think “What are they going to do by using OPMS?”. The line 756-761 is to provide several option to the user. The line 762 is to prompt the user to provide their choice to the system. Since that there are only six options for the user to choose, in line 765-767, if the user’s choice is lesser or equal to zero, or greater than six, the system will display “Invalid input” and call the function `registered_customer_what_do_you_want_to_do()`. In line 768-780, the system will call the function `reg_customer_view_medicine()` if the user’s choice is equal to 1 where the option is “View all medicine”, the function `place_order()` if the user’s choice is equal to 2 where the option is “Place order”, the function `view_order()` if the user’s choice is equal to 3 where the option is “View order”, the function `view_personal_information()` if the user’s choice is equal to 4 where the option is “View Personal Information”, the function `search_medicine_customer()` if the user’s choice is equal to 5 where the option is “Search medicine”, the function `menu()` if the user’s choice is equal to 6 where the option is “Logout”.

Function: upload_medicine()

The line 717-722 is to try to open and allow the system to read and append the medicine text file. If it fails, the system will display “File cannot be opened”. The line 724 is to prompt the user how many medicine they want to upload. The line 725-741 is to prompt the user to provide details of medicine, and append the details to a list which is the

```
713 # admin_what_do_you_want_to_do()
714 # This function is for admin to upload medicine into the system
715 ## assumptions: Roxithromycin & Isotretinoin & Alendronate
716 def upload_medicine():
717     try:
718         fileHandler = open('medicine.txt','a+')
719     except:
720         print ('File cannot be opened.')
721         exit()
722     # This is to open the medicine text file so that the admin can upload medicine to the system
723
724     number = int(input("\n\nHow many medicine do you want to upload?"))
725     medicine = []
726     for line in range(number):
727         med = []
728         medicine_name = input('\n\n\tMedicine name: ')
729         med.append(medicine_name)
730         expire_date = input('\tExpire date (DD/MM/YYYY): ')
731         med.append(expire_date)
732         price = input('\tPrice: ')
733         med.append(price)
734         specification = input('\tSpecification: ')
735         med.append(specification)
736         directions_of_use = input('\tDirections of use: ')
737         med.append(directions_of_use)
738         remark = input('\tRemark: ')
739         med.append(remark)
740         medicine.append(med)
741     # The admin can upload the details of the medicine here
742
743     for med in medicine:
744         for me in med:
745             fileHandler.write(me)
746             fileHandler.write(' \t')
747             fileHandler.write('\n')
748     fileHandler.close()
749
750     print ('\nYou have added a new medicine!')
751     admin_what_do_you_want_to_do()
```

variable med. The line 743-748 is to append medicine details to medicine text file. The system would display “You have added a new medicine!” and call the function `admin_what_do_you_want_to_do()` after the medicine been appended to medicine text file in line 750-751.

Function: admin view medicine()

The line 692-697 is to try to open and allow the system to read the medicine text file. If it fails, the system will display “File cannot be opened”. In line 699-708 allow the system to read line by line in medicine

```
688 # This function is for admin to view all the medicine in the system
689 ## assumptions: Roxithromycin & Isotretinoin & Alendronate
690 def admin_view_medicine():
691     try:
692         fileHandler = open('medicine.txt','r')
693     except:
694         print ('File cannot be opened.')
695         exit()
696     # This is to open the medicine text file so that the admin can view the details of all medicine
697 
698     for line in fileHandler:
699         line = line.rstrip()
700         name, expire_date, price, specifications, directions_of_use, remark = line.split()
701         print("\n\tMedicine name: ",name)
702         print("\t\tExpire date: ",expire_date)
703         print("\t\tPrice: ",price)
704         print("\t\tSpecifications: ",specifications)
705         print("\t\tDirections of use: ",directions_of_use)
706         print("\t\tRemark: ",remark)
707         print('\n')
708     fileHandler.close()
709     # This is to display the details of all medicine
710 
711     admin_what_do_you_want_to_do()
```

text file, split each line and display the details to the user. The medicine text file would be closed in line 709. After the details have been displayed to the user, the system will call the function `admin what do you want to do()`.

Function: reg_customer_view_medicine()

```

664 '''-----'''
665 # This function is for registered customer to view all the medicine in the system
666 def reg_customer_view_medicine():
667     try:
668         fileHandler = open('medicine.txt','r')
669     except:
670         print ('File cannot be opened:')
671         exit()
672     # This is to open the medicine text file so that the registered customer can view the details of all medicine
673
674     for line in fileHandler:
675         line = line.rstrip()
676         name, expire_date, price, specifications, directions_of_use, remark = line.split()
677         print("\n\tMedicine name: ", name)
678         print("\tExpire date: ", expire_date)
679         print("\tPrice: ", price)
680         print("\tSpecifications: ", specifications)
681         print("\tDirections of use: ", directions_of_use)
682         print("\tRemark: ", remark)
683         print('\n')
684     fileHandler.close()
685     # This is to display the details of all medicine
686
687     registered_customer_what_do_you_want_to_do()
688 '''-----'''

```

The line 667-672 is to try to open and allow the system to read the medicine text file. If it fails, the system will display “File cannot be opened”. In line 674-683 allow the system to read line by line in medicine text file, split each line and display the details to the user. The medicine text file would be closed in line 684. After the details have been displayed to the user, the system will call the function registered_customer_what_do_you_want_to_do().

Function: new_customer_view_medicine()

```

640 '''-----'''
641 # This function is for new customer to view all the medicine in the system
642 def new_customer_view_medicine():
643     try:
644         fileHandler = open('medicine.txt','r')
645     except:
646         print ('File cannot be opened:')
647         exit()
648     # This is to open the medicine text file so that the new customer can view the details of all medicine
649
650     for line in fileHandler:
651         line = line.rstrip()
652         name, expire_date, price, specifications, directions_of_use, remark = line.split()
653         print("\n\tMedicine name: ", name)
654         print("\tExpire date: ", expire_date)
655         print("\tPrice: ", price)
656         print("\tSpecifications: ", specifications)
657         print("\tDirections of use: ", directions_of_use)
658         print("\tRemark: ", remark)
659         print('\n')
660     fileHandler.close()
661     # This is to display the details of all medicine
662
663     new_customer_what_do_you_want_to_do()
664 '''-----'''

```

The line 642-648 is to try to open and allow the system to read the medicine text file. If it fails, the system will display “File cannot be opened”. In line 650-659 allow the system to read line by line in medicine text file, split each line and display the details to the user. The medicine text file would be closed in line 660. After the details have been displayed to the user, the system will call the function new_customer_what_do_you_want_to_do().

Function: search_medicine_admin()

```
603 '''  
604 # This function is for admin to search medicine in the system  
605 def search_medicine_admin():  
606     try:  
607         fileHandler = open('medicine.txt','r')  
608     except:  
609         print ('File cannot be opened:')  
610         exit()  
611     # This is to open the medicine text file so that the admin can search for the details of medicine  
612  
613     search_key = input('Type what you want to search: ')  
614     # Admin can type medicine details they wanted to search for  
615  
616     move = False  
617     for line in fileHandler:  
618         line = line.rstrip()  
619         if not search_key.lower() in line.lower():  
620             continue  
621         name, expire_date, price, specifications, directions_of_use, remark = line.split()  
622         print("\n\tMedicine name: ",name)  
623         print("\t\tExpire date: ",expire_date)  
624         print("\t\tPrice: ",price)  
625         print("\t\tSpecifications: ",specifications)  
626         print("\t\tDirections of use: ",directions_of_use)  
627         print("\t\tRemark: ",remark)  
628         print('\n')  
629         move = True  
630     # The system is searching for the medicine  
631  
632     if (move):  
633         print('')  
634     else:  
635         print("\nWe couldn't find anything for",search_key," , try to refine your search.")  
636  
637     fileHandler.close()  
638  
639     admin_what_do_you_want_to_do()  
640 '''
```

The line 606-611 is to try to open and allow the system to read the medicine text file. If it fails, the system will display “File cannot be opened”. The line 613-614 is to prompt the user to type the medicine they want to search. The line 616 is to let the system know that the process of searching medicine had not been done. The system will search for the line which have the medicine the user is searching in line 617-620. If the line has the medicine the user is searching, the system will split the line and display the details of the medicine in line 622-628. The line 629 is to let the system know that the medicine that the user is finding had been found. In line 632-633, the medicine had been found so it displays nothing while in line 634-635, the medicine had not been found and it will prompt the user that nothing had been found. After that, the system would close medicine text file in line 637 and call the function `admin_what_do_you_want_to_do()`.

Function: search_medicine_customer()

```

566 '''
567 # This function is for registered customer to search medicine in the system
568 def search_medicine_customer():
569     try:
570         fileHandler = open('medicine.txt', 'r')
571     except:
572         print('File cannot be opened:')
573         exit()
574     # This is to open the medicine text file so that the registered customer can search for the details of medicine
575
576     search_key = input('Type what you want to search: ')
577     # Registered customer can type medicine details they wanted to search for
578
579     move = False
580     for line in fileHandler:
581         line = line.rstrip()
582         if not search_key.lower() in line.lower():
583             continue
584         name, expire_date, price, specifications, directions_of_use, remark = line.split()
585         print("\n\tMedicine name: ", name)
586         print("\tExpire date: ", expire_date)
587         print("\tPrice: ", price)
588         print("\tSpecifications: ", specifications)
589         print("\tDirections of use: ", directions_of_use)
590         print("\tRemark: ", remark)
591         print('\n')
592         move = True
593     # The system is searching for the medicine
594
595     if (move):
596         print('')
597     else:
598         print("\nWe couldn't find anything for", search_key, ", try to refine your search.")
599
600     fileHandler.close()
601
602     registered_customer_what_do_you_want_to_do()
603 '''

```

The line 569-574 is to try to open and allow the system to read the medicine text file. If it fails, the system will display “File cannot be opened”. The line 576-577 is to prompt the user to type the medicine they want to search. The line 579 is to let the system know that the process of searching medicine had not been done. The system will search for the line which have the medicine the user is searching in line 580-583. If the line has the medicine the user is searching, the system will split the line and display the details of the medicine in line 584-591. The line 592 is to let the system know that the medicine that the user is finding had been found. In line 595-596, the medicine had been found so it displays nothing while in line 597-598, the medicine had not been found and it will prompt the user that nothing had been found. After that, the system would close medicine text file in line 600 and call the function registered_customer_what_do_you_want_to_do().

Function: view_personal_information()

```

541 '''
542 # This function is for registered customer to view all their personal information
543 def view_personal_information():
544     database = open('database.txt', 'r')
545     # This is to open the database text file so that the system can know the personal information of the customer
546
547     for line in database:
548         user_id, user_name, user_password, user_address, user_email, user_phone, user_gender, user_dob = line.split(',')
549         if (user_name==cus_username and user_password==cus_password):
550             print('\n\tUser ID: ', user_id)
551             print('\tUsername: ', user_name)
552             print('\tPassword: ', user_password)
553             print('\tHome address: ', user_address)
554             print('\tEmail address: ', user_email)
555             print('\tContact number: ', user_phone)
556             print('\tGender: ', user_gender)
557             print('\tDate of birth: ', user_dob)
558             break
559         else:
560             continue
561     # The details of the personal information would be displayed here
562     database.close()
563
564     registered_customer_what_do_you_want_to_do()
565 '''
566

```

The line 544 is to open and allow the system to read the database text file. The line 547-561 is to search and display details of the user. The line 563 is to close the medicine text file. The system would call the function registered_customer_what_do_you_want_to_do() in line 565.

Function: place_order()

```

487 '''
488 # This function is for registered customer to place an order
489 ## assumptions: Roxithromycin & Isotretinoin & Alendronate
490 def place_order():
491     print("\n\nEnter the details of your order.")
492
493     try:
494         fileHandler = open('medicine.txt', 'r')
495     except:
496         print('File cannot be opened:')
497         exit()
498     # This is to open the medicine text file so that the system can know what medicine is exist
499
500     medicine_name = str(input("\n\tMedicine name: "))
501     quantity = str(input("\tQuantity: "))
502
503     move = False
504     for line in fileHandler:
505         line = line.rstrip()
506         if not medicine_name.lower() in line.lower():
507             continue
508         move = True
509     # This is to search whether the medicine typed by the customer is exist in the system
510
511     if (move):
512         print('')
513     else:
514         print("\nMedicine doesn't exist.")
515         place_order()
516
517     fileHandler.close()
518
519     database = open('database.txt', 'r')
520     # This is to open the database text file so that the system can know the details of the customer
521
522     for line in database:
523         user_id, user_name, user_password, user_address, user_email, user_phone, user_gender, user_dob = line.split('!')
524         if (user_name==cus_username and user_password==cus_password):
525             address = user_address
526             email = user_email
527             phone = user_phone
528             break
529         else:
530             continue
531     # This is for the system to know the details of the customer
532
533     database.close()
534
535     order = open('order.txt', 'a') # This is to open the order text file so that the system can add a new order
536     order.write(cus_username + '!' + address + '!' + email + '!' + phone + '!' + medicine_name + '!' + quantity + '\n')
537     print('You have placed an order successfully!')
538     order.close()
539
540     registered_customer_what_do_you_want_to_do()
541 '''

```

The line 491 is to prompt the user to type the details of the order. The line 493-498 is to try to open and allow the system to read the medicine text file. If it fails, the system will display “File cannot be opened”. The line 500-501 is to let the user know what details they should provide. The line 503 is to let the system know that the medicine had not been found. The system will search for the line which have the medicine the user is searching in line 504-507. If the line has the medicine the user wishes to place an order, the line 508 will let the system know that the medicine had been found. The medicine text file would be close in line 517 and the database text file will open and allow the system to read in line 519. The system will search for the details of the user in line 522-531 and close the database text file in line 533. The line 535 is to open and allow the system to append the order text file. The system would append user’s order details to order text file in line 536 and let the user know that the order had been placed successfully in line 537. The order text file will be close in line 538 and the system will call the function registered_customer_what_do_you_want_to_do() in line 540.

Function: view_order()

```

458 '''
459 # This function is for registered customer to view their order
460 def view_order():
461     order = open('order.txt', 'r')
462     # This is to open the order text file so that the system can know what order is exist
463
464     move=False
465     for line in order:
466         name, address, email, phone, medicine, quantity = line.split('!')
467         if (name==cus_username):
468             print('\n\tUsername: ', name)
469             print('\tHome address: ', address)
470             print('\tEmail: ', email)
471             print('\tContact number: ', phone)
472             print('\tMedicine name: ', medicine)
473             print('\tQuantity: ', quantity)
474             move=True
475         else:
476             continue
477     # This is to show the order made by a specific customer
478
479     order.close()
480
481     if (move):
482         print('')
483     else:
484         print("\nYou have not order anything yet.")
485
486     registered_customer_what_do_you_want_to_do()
487 '''

```

The line 461 is to open and allow the system to read the order text file. The line 464 is to let the system know that the order has not been viewed. The system would find for the order and display it in line 465-477. The line 474 is to let the system know that the order had been viewed. The system would close the order text file in line 479. In line 481-484, if the order had been viewed, it will display nothing while if the order had not been viewed, the system would display “You have not order anything yet.”. After that, in line 486, the system will call the function registered_customer_what_do_you_want_to_do().

Function: view_orders()

```

439 '''
440 # This function is for admin to view all the order in the system
441 def view_orders():
442     order = open('order.txt', 'r')
443     # This is to open the order text file so that the system can know what order is exist
444
445     for line in order:
446         name, address, email, phone, medicine, quantity = line.split('!')
447         print('\n\tCustomer username: ', name)
448         print('\tHome address: ', address)
449         print('\tEmail: ', email)
450         print('\tContact number: ', phone)
451         print('\tMedicine name: ', medicine)
452         print('\tQuantity: ', quantity)
453     # This is to show all the orders made by customer
454
455     order.close()
456
457     admin_what_do_you_want_to_do()
458 '''

```

The line 442 is to open and allow the system to read the order text file. The line 445-453 is to display the details of all order line by line. The system would close the order text file in line 455 and call the function admin_what_do_you_want_to_do() in line 457.

Function: search_orders()

The line 427-428 is to prompt the user to choose an option from the option given and provide the choice to the system in line 429.

```

424 '''
425 # This function is for admin to choose how they want to search the order in the system
426 def search_orders():
427     print('How do you want to search from?')
428     print('\n\t1. Search by customer name \n\t2. Search by medicine name \n')
429     choice = int(input('Choose the operation from the given options: '))
430     # This is to let the admin to choose how they want to search the order.
431
432     if choice <= 0 or choice > 2:
433         print('Invalid input')
434         search_orders()
435     elif choice == 1:
436         search_order_by_customer_name()
437     else:
438         search_order_by_medicine_name()
439 '''

```

Since that there are only two options for the user to choose, in line 432-434, if the user's choice is lesser or equal to 0, or greater than 2, the system will display "Invalid input" and call the function search_orders(). In line 435-438, the system will call the function search_order_by_customer_name() if the user's choice is equal to 1 where the option is "Search by customer name" or the function search_order_by_medicine_name() if the user's choice is equal to 2 where the option is "Search by medicine name".

Function: search_order_by_customer_name()

The line 398 is to prompt the user to enter the customer name of the order they want to search. The line 399 is to open and allow the system to read the order text file. The

```

395 '''
396 # This function is for admin to search the order by customer name
397 def search_order_by_customer_name():
398     search = str(input('\n\tCustomer name: '))
399     order = open('order.txt', 'r') # This is to open the order text file so that the system can know what order is exist
400
401     move=False
402     for line in order:
403         name, address, email, phone, medicine, quantity = line.split(' ')
404         if name==search:
405             print('\n\tCustomer username: ', name)
406             print('\tHome address: ', address)
407             print('\tEmail: ', email)
408             print('\tContact number: ', phone)
409             print('\tMedicine name: ', medicine)
410             print('\tQuantity: ', quantity)
411             move = True
412         else:
413             continue
414     # This is to show the order made by a specific customer
415
416     if (move):
417         print('')
418     else:
419         print("\nCustomer doesn't exist or Customer don't have any order.")
420
421     order.close()
422
423     admin_what_do_you_want_to_do()
424 '''

```

line 401 is to let the system know that the process of searching the order had not been done. The system will search for the line with the customer username and display the details of the order when the line is found in line 402-414. The line 411 is to let the system know that the order had been found and displayed. In line 416-417, the order had been displayed so it displays nothing while in line 418-419, the order of the customer had not been found and it will prompt the user that nothing had been found. After that, the system would close medicine text file in line 421 and call the function admin_what_do_you_want_to_do() in line 423.

Function: search_order_by_medicine_name()

```

366 '''
367 # This function is for admin to search the order by medicine name
368 def search_order_by_medicine_name():
369     search = str(input('\n\tMedicine name: '))
370     order = open('order.txt','r')# This is to open the order text file so that the system can know what order is exist
371
372     move=False
373     for line in order:
374         name,address,email,phone,medicine,quantity = line.split(' ')
375         if medicine==search:
376             print('\n\tCustomer username: ',name)
377             print('\tHome address: ',address)
378             print('\tEmail: ',email)
379             print('\tContact number: ',phone)
380             print('\tMedicine name: ',medicine)
381             print('\tQuantity: ',quantity)
382             move = True
383         else:
384             continue
385     # This is to show the order with a specific medicine
386
387     if (move):
388         print('')
389     else:
390         print("\nMedicine doesn't exist or no customer place order to this medicine yet.")
391
392     order.close
393
394     admin_what_do_you_want_to_do()
395 '''

```

The line 369 is to prompt the user to enter the medicine name of the order they want to search. The line 370 is to open and allow the system to read the order text file. The line 372 is to let the system know that the process of searching the order had not been done. The system will search for the line with the medicine name and display the details of the order when the line is found in line 373-385. The line 382 is to let the system know that the order had been found and displayed. In line 387-388, the order had been displayed so it displays nothing while in line 389-390, the order of the medicine had not been found and it will prompt the user that nothing had been found. After that, the system would close medicine text file in line 392 and call the function admin_what_do_you_want_to_do() in line 394.

Function: modify_medicine()

```

343 '''
344 # This function is for admin to choose how they want to modify the medicine in the system
345 def modify_medicine():
346     print('\n\nWhat do you want to modify?')
347     print('\n\t1. Medicine name \n\t2. Expire date \n\t3. Price \n\t4. Specification \n\t5. Directions of use \n\t6. Remark \n')
348     choice = int(input('\nChoose the operation from the given options: '))
349     # This is to let the admin to choose how they want to modify the medicine.
350
351     if choice <= 0 or choice > 6:
352         print('Invalid input')
353         modify_medicine()
354     elif choice == 1:
355         modify_medicine_name()
356     elif choice == 2:
357         modify_medicine_expire_date()
358     elif choice == 3:
359         modify_medicine_price()
360     elif choice == 4:
361         modify_medicine_spec()
362     elif choice == 5:
363         modify_medicine_directions_of_use()
364     else:
365         modify_medicine_remark()
366 '''

```

The line 346-347 is to prompt the user to choose an option from the option given and provide the choice to the system in line 348. Since that there are only six options for the user to choose, in line 351-353, if the user's choice is lesser or equal to 0, or greater than 6, the system will display "Invalid input" and call the function `modify_medicine()`. In line 354-365, the system will call the function `modify_medicine_name()` if the user's choice is equal to 1 where the option is "Medicine name", the function `modify_medicine_expire_date()` if the user's choice is equal to 2 where the option is "Expire date", the function `modify_medicine_price()` if the user's choice is equal to 3 where the option is "Price", the function `modify_medicine_spec()` if the user's choice is equal to 4 where the option is "Specification", the function `modify_medicine_directions_of_use()` if the user's choice is equal to 5 where the option is "Directions of use" or the function `modify_medicine_remark()` if the user's choice is equal to 6 where the option is "Remark".

Function: modify_medicine_name()

```

294 '''
295 # This function is for admin to modify the medicine name
296 def modify_medicine_name():
297     try:
298         fileHandler = open('medicine.txt', 'r+')
299     except:
300         print('File cannot be opened:')
301         exit()
302     # This is to open the medicine text file so that the system can know what medicine is exist and also let the system to add details
303
304     search_key = input('Type actual medicine name: ')
305     replace = input('Type new medicine name: ')
306     # This is to let the admin to type the actual and new medicine name
307
308     move = False
309     for line in fileHandler:
310         line = line.rstrip()
311         if not search_key.lower() in line.lower(): # This is to search whether the medicine exist
312             continue
313         lines=line.split()
314         lines[0]= replace # This is to enter the new medicine name
315         for fs in lines:
316             fileHandler.write(fs)
317             fileHandler.write(' ')
318             fileHandler.write('\n')
319         move = True
320         delete = line + "\t" # This is to read the line which is going to be deleted
321         continue
322
323     if (move):
324         print("\n", 'Medicine name modified successfully!')
325     else:
326         print("\nWe couldn't find anything for", search_key, ", try to refine your search.")
327
328     fileHandler.close()
299 '''
330
331 old_file = open("medicine.txt", "r")
332 lines = old_file.readlines()
333 old_file.close()
334 # This is to read the details exist in medicine text file
335
336 new_file = open("medicine.txt", "w")
337 for line in lines:
338     if line.strip("\n") != delete:
339         new_file.write(line_)
340 new_file.close()
341 # This is to enter the details of the medicine back into the medicine text file but without the medicine with the old name
342
343 admin_what_do_you_want_to_do()
344 '''

```

The line 297-302 is to try to open and allow the system to read and write the medicine text file. If it fails, the system will display “File cannot be opened”. The line 304-306 is to prompt the user to enter the actual medicine name and the new medicine name. The line 308 is to let the system know that the medicine name had not been modified. The system will search for the line which have the medicine the user wanted to modify in line 309-312. If the line has the medicine the user wanted to modify, the system would read the details of the medicine and append a new medicine in medicine text file with the newest medicine details in line 313-318. The line 319 is to let the system know that the medicine name had been modified and line 320 is to let the system know that the details needed to be deleted. In line 323-324, the system will display “Medicine name modified successfully!” when the medicine name had been found while in line 325-326, the system will let the user know that the medicine cannot be found. The system will close medicine text file in line 328. In line 330-331, the system will open and read the medicine text file. It save the content in the variable lines. The system will close medicine text file in line 332. In line 335-338, the system will open and write the medicine text file. It rewrite the medicine text file without adding the line which have to be deleted. The system will close medicine text file in line 339 and call the function admin_what_do_you_want_to_do() in line 342.

Function: modify_medicine_expire_date()

```

245 '''
246 # This function is for admin to modify the medicine expire date
247 def modify_medicine_expire_date():
248     try:
249         fileHandler = open('medicine.txt', 'r+')
250     except:
251         print('File cannot be opened:')
252         exit()
253     # This is to open the medicine text file so that the system can know what medicine is exist and also let the system to add details
254
255     search_key = input('Type medicine name: ')
256     replace = input("Type new expire date: ")
257     # This is to let the admin to type the medicine name and the new expire date
258
259     move = False
260     for line in fileHandler:
261         line = line.rstrip()
262         if not search_key.lower() in line.lower(): # This is to search whether the medicine exist
263             continue
264         lines=line.split()
265         lines[1]= replace # This is to modify the expire date
266         for fs in lines:
267             fileHandler.write(fs)
268             fileHandler.write(' ')
269             fileHandler.write('\n')
270         move = True
271         delete = line + "\t" # This is to read the line which is going to be deleted
272         continue
273
274     if (move):
275         print("\n", 'Expire date modified successfully!')
276     else:
277         print("\nWe couldn't find anything for", search_key, ", try to refine your search.")
278
279     fileHandler.close()
280
281 old_file = open("medicine.txt", "r")
282 lines = old_file.readlines()
283 old_file.close()
284 # This is to read the details exist in medicine text file
285
286 new_file = open("medicine.txt", "w")
287 for line_ in lines:
288     if line_.strip("\n") != delete:
289         new_file.write(line_)
290 new_file.close()
291 # This is to enter the details of the medicine back into the medicine text file but without the medicine with the old expire date
292
293 admin_what_do_you_want_to_do()
294 '''

```

The line 248-253 is to try to open and allow the system to read and write the medicine text file. If it fails, the system will display “File cannot be opened”. The line 255-257 is to prompt the user to enter the medicine name and the new expire date. The line 259 is to let the system know that the expire date had not been modified. The system will search for the line which have the medicine the user wanted to modify in line 260-263. If the line has the medicine the user wanted to modify, the system would read the details of the medicine and append a new medicine in medicine text file with the newest medicine details in line 264-269. The line 270 is to let the system know that the medicine expire date had been modified and line 271 is to let the system know that the details needed to be deleted. In line 274-275, the system will display “Expire date modified successfully!” when the medicine name had been found while in line 276-277, the system will let the user know that the medicine cannot be found. The system will close medicine text file in line 279. In line 281-282, the system will open and read the medicine text file. It save the content in the variable lines. The system will close medicine text file in line 283. In line 286-289, the system will open and write the medicine text file. It rewrite the medicine text file without adding the line which have to be deleted. The system will close medicine text file in line 290 and call the function admin_what_do_you_want_to_do() in line 293.

Function: modify_medicine_price()

```

196
197 # This function is for admin to modify the medicine price
198 def modify_medicine_price():
199     try:
200         fileHandler = open('medicine.txt', 'r+')
201     except:
202         print('File cannot be opened:')
203         exit()
204     # This is to open the medicine text file so that the system can know what medicine is exist and also let the system to add details
205
206     search_key = input('Type medicine name: ')
207     replace = input("Type new medicine price: ")
208     # This is to let the admin to type the medicine name and the new medicine price
209
210     move = False
211     for line in fileHandler:
212         line = line.rstrip()
213         if not search_key.lower() in line.lower(): # This is to search whether the medicine exist
214             continue
215         lines=line.split()
216         lines[2]= replace # This is to modify the medicine price
217         for fs in lines:
218             fileHandler.write(fs)
219             fileHandler.write('\t')
220             fileHandler.write('\n')
221         move = True
222         delete = line + "\t" # This is to read the line which is going to be deleted
223         continue
224
225     if (move):
226         print("\n", 'Medicine price modified sucessfully!')
227     else:
228         print("\nWe couldn't find anything for", search_key, ", try to refine your search.")
229
230     fileHandler.close()
231
232
233 old_file = open("medicine.txt", "r")
234 lines = old_file.readlines()
235 old_file.close()
236 # This is to read the details exist in medicine text file
237
238 new_file = open("medicine.txt", "w")
239 for line_ in lines:
240     if line_.strip("\n") != delete:
241         new_file.write(line_)
242     # This is to enter the details of the medicine back into the medicine text file but without the medicine with the old price
243
244 admin_what_do_you_want_to_do()
245

```

The line 199-204 is to try to open and allow the system to read and write the medicine text file. If it fails, the system will display “File cannot be opened”. The line 206-208 is to prompt the user to enter the medicine name and the new medicine price. The line 210 is to let the system know that the medicine price had not been modified. The system will search for the line which have the medicine the user wanted to modify in line 211-214. If the line has the medicine the user wanted to modify, the system would read the details of the medicine and append a new medicine in medicine text file with the newest medicine details in line 215-220. The line 221 is to let the system know that the medicine price had been modified and line 222 is to let the system know that the details needed to be deleted. In line 225-226, the system will display “Medicine price modified successfully!” when the medicine name had been found while in line 227-228, the system will let the user know that the medicine cannot be found. The system will close medicine text file in line 230. In line 232-233, the system will open and read the medicine text file. It save the content in the variable lines. The system will close medicine text file in line 234. In line 237-240, the system will open and write the medicine text file. It rewrite the medicine text file without adding the line which have to be deleted. The system will close medicine text file in line 241 and call the function admin_what_do_you_want_to_do() in line 244.

Function: modify_medicine_spec()

```

147 '''
148 # This function is for admin to modify the medicine specifications
149 def modify_medicine_spec():
150     try:
151         fileHandler = open('medicine.txt', 'r+')
152     except:
153         print('File cannot be opened:')
154         exit()
155     # This is to open the medicine text file so that the system can know what medicine is exist and also let the system to add details
156
157     search_key = input('Type medicine name: ')
158     replace = input("Type new medicine specifications: ")
159     # This is to let the admin to type the medicine name and the new specifications
160
161     move = False
162     for line in fileHandler:
163         line = line.rstrip()
164         if not search_key.lower() in line.lower(): # This is to search whether the medicine exist
165             continue
166         lines=line.split()
167         lines[3]= replace # This is to modify the specifications
168         for fs in lines:
169             fileHandler.write(fs)
170             fileHandler.write('\t')
171             fileHandler.write('\n')
172         move = True
173         delete = line + "\t" # This is to read the line which is going to be deleted
174         continue
175
176     if (move):
177         print("\n", 'Medicine specifications modified sucessfully!')
178     else:
179         print("\nWe couldn't find anything for", search_key, ", try to refine your search.")
180
181     fileHandler.close()
182
183 old_file = open("medicine.txt", "r")
184 lines = old_file.readlines()
185 old_file.close()
186 # This is to read the details exist in medicine text file
187
188 new_file = open("medicine.txt", "w")
189 for line in lines:
190     if line.strip("\n") != delete:
191         new_file.write(line)
192 new_file.close()
193 # This is to enter the details of the medicine back into the medicine text file but without the medicine with the old specifications
194
195 admin_what_do_you_want_to_do()
196 '''

```

The line 150-155 is to try to open and allow the system to read and write the medicine text file. If it fails, the system will display “File cannot be opened”. The line 157-159 is to prompt the user to enter the medicine name and the new medicine specifications. The line 161 is to let the system know that the medicine specifications had not been modified. The system will search for the line which have the medicine the user wanted to modify in line 162-165. If the line has the medicine the user wanted to modify, the system would read the details of the medicine and append a new medicine in medicine text file with the newest medicine details in line 166-171. The line 172 is to let the system know that the medicine price had been modified and line 173 is to let the system know that the details needed to be deleted. In line 176-177, the system will display “Medicine specifications modified sucessfully!” when the medicine name had been found while in line 178-179, the system will let the user know that the medicine cannot be found. The system will close medicine text file in line 181. In line 183-184, the system will open and read the medicine text file. It save the content in the variable lines. The system will close medicine text file in line 185. In line 188-191, the system will open and write the medicine text file. It rewrite the medicine text file without adding the line which have to be deleted. The system will close medicine text file in line 192 and call the function admin_what_do_you_want_to_do() in line 195.

Function: modify medicine directions of use()

```

98 """
99 # This function is for admin to modify the medicine direction of use
100 def modify_medicine_directions_of_use():
101     try:
102         fileHandler = open('medicine.txt', 'r+')
103     except:
104         print ('File cannot be opened:')
105         exit()
106     # This is to open the medicine text file so that the system can know what medicine is exist and also let the system to add details
107
108     search_key = input('Type medicine name: ')
109     replace = input ("Type new medicine directions of use: ")
110     # This is to let the admin to type the medicine name and the new directions of use
111
112     move = False
113     for line in fileHandler:
114         line = line.rstrip()
115         if not search_key.lower() in line.lower(): # This is to search whether the medicine exist
116             continue
117         lines=line.split()
118         lines[4]= replace # This is to modify the directions of use
119         for fs in lines:
120             fileHandler.write(fs)
121             fileHandler.write('\t')
122             fileHandler.write('\n')
123         move = True
124         delete = line + "\t" # This is to read the line which is going to be deleted
125         continue
126
127     if (move):
128         print("\n", 'Medicine directions of use modified sucessfully!')
129     else:
130         print("\nWe couldn't find anything for", search_key, ", try to refine your search.")
131
132     fileHandler.close()
133
134
135 old_file = open("medicine.txt", "r")
136 lines = old_file.readlines()
137 old_file.close()
138 # This is to read the details exist in medicine text file
139
140 new_file = open("medicine.txt", "w")
141 for line_ in lines:
142     if line_.strip("\n") != delete:
143         new_file.write(line_)
144 new_file.close()
145 # This is to enter the details of the medicine back into the medicine text file but without the medicine with the old direction of
146
147 ... admin_what_do_you_want_to_do()
148 """

```

The line 101-106 is to try to open and allow the system to read and write the medicine text file. If it fails, the system will display “File cannot be opened”. The line 108-109 is to prompt the user to enter the medicine name and the new medicine directions of use. The line 112 is to let the system know that the medicine directions of use had not been modified. The system will search for the line which have the medicine the user wanted to modify in line 113-116. If the line has the medicine the user wanted to modify, the system would read the details of the medicine and append a new medicine in medicine text file with the newest medicine details in line 117-122. The line 123 is to let the system know that the medicine directions of use had been modified and line 124 is to let the system know that the details needed to be deleted. In line 127-128, the system will display “Medicine directions of use modified sucessfully!” when the medicine name had been found while in line 129-130, the system will let the user know that the medicine cannot be found. The system will close medicine text file in line 132. In line 134-135, the system will open and read the medicine text file. It save the content in the variable lines. The system will close medicine text file in line 136. In line 139-142, the system will open and write the medicine text file. It rewrite the medicine text file without adding the line which have to be deleted. The system will close medicine text file in line 143 and call the function admin_what_do_you_want_to_do() in line 146.

Function: modify_medicine_remark()

```

49 '''
50 # This function is for admin to modify the medicine remark
51 def modify_medicine_remark():
52     try:
53         fileHandler = open('medicine.txt', 'r+')
54     except:
55         print ('File cannot be opened:')
56         exit()
57     # This is to open the medicine text file so that the system can know what medicine is exist and also let the system to add details
58
59     search_key = input('Type medicine name: ')
60     replace = input ("Type new medicine remark: ")
61     # This is to let the admin to type the medicine name and the new remark
62
63     move = False
64     for line in fileHandler:
65         line = line.rstrip()
66         if not search_key.lower() in line.lower(): # This is to search whether the medicine exist
67             continue
68         lines=line.split()
69         lines[5]= replace # This is to modify the remark
70         for fs in lines:
71             fileHandler.write(fs)
72             fileHandler.write('\t')
73             fileHandler.write('\n')
74         move = True
75         delete = line + "\t" # This is to read the line which is going to be deleted
76         continue
77
78     if (move):
79         print("\n", 'Medicine remark modified sucessfully!')
80     else:
81         print("\nWe couldn't find anything for", search_key, ", try to refine your search.")
82
83     fileHandler.close()
84
85     old_file = open("medicine.txt", "r")
86     lines = old_file.readlines()
87     old_file.close()
88     # This is to read the details exist in medicine text file
89
90     new_file = open("medicine.txt", "w")
91     for line_ in lines:
92         if line_.strip("\n") != delete:
93             new_file.write(line_)
94     new_file.close()
95     # This is to enter the details of the medicine back into the medicine text file but without the medicine with the old remark
96
97     admin_what_do_you_want_to_do()
98 '''

```

The line 52-57 is to try to open and allow the system to read and write the medicine text file. If it fails, the system will display “File cannot be opened”. The line 59-60 is to prompt the user to enter the medicine name and the new medicine remark. The line 63 is to let the system know that the medicine remark had not been modified. The system will search for the line which have the medicine the user wanted to modify in line 64-67. If the line has the medicine the user wanted to modify, the system would read the details of the medicine and append a new medicine in medicine text file with the newest medicine details in line 68-73. The line 74 is to let the system know that the medicine remark had been modified and line 75 is to let the system know that the details needed to be deleted. In line 78-79, the system will display “Medicine remark modified sucessfully!” when the medicine name had been found while in line 80-81, the system will let the user know that the medicine cannot be found. The system will close medicine text file in line 83. In line 85-86, the system will open and read the medicine text file. It save the content in the variable lines. The system will close medicine text file in line 87. In line 90-93 the system will open and write the medicine text file. It rewrite the medicine text file without adding the line which have to be deleted. The system will close medicine text file in line 94 and call the function admin_what_do_you_want_to_do() in line 97.

Function: delete_medicine()

```

6 '''
7 # This function is for admin to delete the medicine
8 def delete_medicine():
9     try:
10         fileHandler = open('medicine.txt', 'r')
11     except:
12         print('File cannot be opened:')
13         exit()
14     # This is to open the medicine text file so that the system can know what medicine is exist
15
16     search_key = input('\n\tEnter name of medicine you want to delete: ')
17     # This is to let the admin to type the medicine name which they wanted to be delete
18
19     move = False
20     for line in fileHandler:
21         line = line.rstrip()
22         if not search_key.lower() in line.lower(): # This is to search for the medicine
23             continue
24         delete = line + "\t"
25         move = True
26         break
27     # This is to read the line which is going to be deleted
28
29     if (move):
30         print("\n", search_key, 'deleted successfully.')
31     else:
32         print("\nWe couldn't find anything for", search_key, ", try to refine your search.")
33
34     fileHandler.close()
35
36     a_file = open("medicine.txt", "r")
37     lines = a_file.readlines()
38     a_file.close()
39     # This is to read the details exist in medicine text file
40
41     new_file = open("medicine.txt", "w")
42     for line_ in lines:
43         if line_.strip("\n") != delete:
44             new_file.write(line_)
45     new_file.close()
46     # This is to enter the details of the medicine back into the medicine text file but without the deleted medicine
47
48     admin_what_do_you_want_to_do()
49 '''

```

The line 9-14 is to try to open and allow the system to read and write the medicine text file. If it fails, the system will display “File cannot be opened”. The line 16-17 is to prompt the user to enter the medicine name want to be deleted. The line 19 is to let the system know that the medicine had not been deleted. The system will search for the line which have the medicine the user wanted to delete in line 20-23. If the line has the medicine the user wanted to delete, the system would read the details of the medicine in line 24. The line 25 is to let the system know that the medicine name can be found. In line 29-30, the system will prompt the user that the medicine had been deleted when the medicine name can be found while in line 31-32, the system will let the user know that the medicine cannot be found. The system will close medicine text file in line 34. In line 36-37, the system will open and read the medicine text file. It save the content in the variable lines. The system will close medicine text file in line 38. In line 41-44 the system will open and write the medicine text file. It rewrite the medicine text file without adding the line which have to be deleted. The system will close medicine text file in line 45 and call the function `admin_what_do_you_want_to_do()` in line 48.

User Manual

I. Menu

When a user just on the Online Pharmacy Management System (OPMS), he or she can see a menu as below.

```
Who are you?  
    1. Admin  
    2. New Customer  
    3. Registered Customer  
  
Choose the operation from the given options:
```

Figure 1: Menu

The admin can choose the option 1 to login to the system and make changes to the system. The customer who are using OPMS for the first time or doesn't have an account can choose the option 2. Option 3 is for customer who have an account and can login to the system and use the system.

II. New Customer

```
Who are you?
1. Admin
2. New Customer
3. Registered Customer

Choose the operation from the given options: 2
What do you want to do by using OPMS?
1. View all medicine
2. Register
3. Menu

Choose the operation from the given options:
```

Figure 2: New Customer Menu

If the user is using the system for the first time or doesn't have an account, they can choose the option 2 in the menu. After choosing the option 2, the new customer can choose what they want to do by using OPMS. They can choose the option 1 to view all medicine, or the option 2 to register a new account, and also the option 3 to go back to the menu.

View all medicine

```

What do you want to do by using OPMS?

1. View all medicine
2. Register
3. Menu

Choose the operation from the given options: 1

Medicine name: Roxithromycin
Expire date: 31/12/2022
Price: RM_320
Specifications: an_antibiotic_that_is_used_to_treat_infections
Directions of use: take_1_tablets_2_times_daily_before_food
Remark: controlled_medicine

Medicine name: Isotretinoin
Expire date: 31/12/2023
Price: RM_32
Specifications: used_to_treat_severe_acne
Directions of use: take_1_tablets_1_times_daily_before_food
Remark: controlled_medicine

Medicine name: Alendronate
Expire date: 31/12/2024
Price: RM_3
Specifications: used_to_treat_and_prevent_osteoporosis
Directions of use: It_should_be_taken_as_soon_as_you_get_out_of_bed_in_the_morning_and_at_least_30_minutes_before_any_food
Remark: no_remark

What do you want to do by using OPMS?

1. View all medicine
2. Register
3. Menu

Choose the operation from the given options: |

```

Figure 3: New Customer View All Medicine

If the customer chooses the option 1 that is “View all medicine”, all the medicine recorded in the system will be shown to the customer. The system would show the medicine and its details which were recorded inside the medicine text file.

```

Roxithromycin 31/12/2022 RM_320 an_antibiotic_that_is_used_to_treat_infections take_1_tablets_2_times_daily_before_food controlled_medicine
Isotretinoin 31/12/2023 RM_32 used_to_treat_severe_acne take_1_tablets_1_times_daily_before_food controlled_medicine
Alendronate 31/12/2024 RM_3 used_to_treat_and_prevent_osteoporosis It_should_be_taken_as_soon_as_you_get_out_of_bed_in_the_morning_and_at_least_30_minutes_before_any_food no_remark

```

Figure 4: Medicine Text File

After showing the medicine and also the details of the medicine, the system will go back to allow the user to choose what they want to do next.

Register

```

What do you want to do by using OPMS?

    1. View all medicine
    2. Register
    3. Menu

Choose the operation from the given options: 2

Create Your OPMS Account

    Username: Kathy Sierra
    Password: 123456
    Confirm password: 123456
    Home address: 14, Jalan Merdeka, Bandar Hilir, 75000 Melaka
    Email address: kathysierra@gmail.com
    Contact number: 012-609 6790
    Gender (F/M): F
    Date of birth (DD/MM/YYYY): 05/07/2003

You have created an account successfully!

Kathy Sierra, your user ID is 5.

Please login from registered customer.

Who are you?

    1. Admin
    2. New Customer
    3. Registered Customer

Choose the operation from the given options:

```

Figure 5: New Customer Register

If the customer chooses the option 2 that is “Register”, the customer can register a new account for themselves. They have to provide a username (e.g. Kathy Sierra), a password with at least six characters (e.g. 123456), a confirm password which is totally the same with the password, home address (e.g. 14, Jalan Merdeka, Bandar Hilir, 75000 Melaka.) where OCEAN Sdn Bhd can send medicine to customer after customer place an order, email address (e.g. kathysierra@gmail.com), contact number (e.g. 012-609 6790), gender (e.g. F) and also date of birth (05/07/2003). If the account has been created successfully, it will show “You have created an account successfully!” and customer can get their user ID. The details of the customer will also be recorded into database text file.

```

1!Admin!12345678!OCEAN Sdn Bhd!opms@ocean.com.my!012-345 6789!F!01/01/2020
2!Jayden!123456!1!1!1!F!1
3!test!123456!123!123!123!123!123
4!123!123456!321!321!321!231!312
5!Kathy Sierra!123456!14, Jalan Merdeka, Bandar Hilir, 75000 Melaka!kathysierra@gmail.com!012-609 6790!F!05/07/2003

```

Figure 6: Database Text File

After the process of registration had been done, the system will go back to the menu and the customer can login to the system by choosing the option 3 which is registered customer.

Menu

```
What do you want to do by using OPMS?  
    1. View all medicine  
    2. Register  
    3. Menu  
Choose the operation from the given options: 3  
  
Who are you?  
    1. Admin  
    2. New Customer  
    3. Registered Customer  
Choose the operation from the given options: |
```

Figure 7: New Customer Menu

If the customer chooses the option 3 that is Menu, the system will go back to the menu.

III. Registered Customer

```
Who are you?
    1. Admin
    2. New Customer
    3. Registered Customer

Choose the operation from the given options: 3
Please enter your username and password.
    Username: Kathy Sierra
    Password: 123456

Login Successful!!!

What do you want to do by using OPMS?
    1. View all medicine
    2. Place order
    3. View order
    4. View Personal Information
    5. Search medicine
    6. Logout

Choose the operation from the given options:
```

Figure 8: Registered Customer Menu

If the user already had an account, they can choose the option 3 in the menu. After choosing the option 3, the registered customer has to login to the system with the username and password they registered. They can choose what they want to do by using OPMS. They can choose the option 1 to view all medicine, the option 2 to place an order, the option 3 to view the order they made before, the option 4 to view their personal information, the option 5 to search medicine in the system and also the option 6 to logout of their account.

View all medicine

```

Choose the operation from the given options: 1

Medicine name: Roxithromycin
Expire date: 31/12/2022
Price: RM_320
Specifications: an_antibiotic_that_is_used_to_treat_infections
Directions of use: take_1_tablets_2_times_daily_before_food
Remark: controlled_medicine

Medicine name: Isotretinoin
Expire date: 31/12/2023
Price: RM_32
Specifications: used_to_treat_severe_acne
Directions of use: take_1_tablets_1_times_daily_before_food
Remark: controlled_medicine

Medicine name: Alendronate
Expire date: 31/12/2024
Price: RM_3
Specifications: used_to_treat_and_prevent_osteoporosis
Directions of use: It_should_be_taken_as_soon_as_you_get_out_of_bed_in_the_morning_and_at_least_30_minutes_before_any_food
Remark: no_remark

What do you want to do by using OPMS?

1. View all medicine
2. Place order
3. View order
4. View Personal Information
5. Search medicine
6. Logout

Choose the operation from the given options:

```

Figure 9: Registered Customer View Medicine

If the customer chooses the option 1 that is “View all medicine”, all the medicine recorded in the system will be shown to the customer. The system would show the medicine and its details which were recorded inside the medicine text file.

```

Roxithromycin 31/12/2022 RM_320 an_antibiotic_that_is_used_to_treat_infections take_1_tablets_2_times_daily_before_food controlled_medicine
Isotretinoin 31/12/2023 RM_32 used_to_treat_severe_acne take_1_tablets_1_times_daily_before_food controlled_medicine
Alendronate 31/12/2024 RM_3 used_to_treat_and_prevent_osteoporosis It_should_be_taken_as_soon_as_you_get_out_of_bed_in_the_morning_and_at_least_30_minutes_before_any_food no_remark

```

Figure 10: Medicine Text File

After showing the medicine and also the details of the medicine, the system will go back to allow the user to choose what they want to do next.

Place order

```
What do you want to do by using OPMS?
1. View all medicine
2. Place order
3. View order
4. View Personal Information
5. Search medicine
6. Logout
Choose the operation from the given options: 2
Enter the details of your order.
Medicine name: Alendronate
Quantity: 5
You have placed an order successfully!
```

Figure 11: Registered Customer Place Order

If the customer chooses the option 2 that is “Place order”, the customer can order medicine here. After choosing the option 2, customer will need to provide the medicine name (e.g. Alendronate) that exist in the system and also the quantity (e.g. 5) of the medicine. The order made by the customer will be saved in order text file.

```
Admin!OCEAN Sdn Bhd!opms@ocean.com.my!012-345 6789!Alendronate!4
Admin!OCEAN Sdn Bhd!opms@ocean.com.my!012-345 6789!Isotretinoin!4
Jayden!OCEAN Sdn Bhd!opms@ocean.com.my!012-345 6789!Roxithromycin!8
Admin!OCEAN Sdn Bhd!opms@ocean.com.my!012-345 6789!Alendronate!3
Kathy Sierra!14, Jalan Merdeka, Bandar Hilir, 75000 Melaka!kathysierra@gmail.com!012-609 6790!Alendronate!5
```

Figure 12: Order Text File

After the order had been saved, the system will go back to allow the customer to choose what they want to do next.

View order

```
What do you want to do by using OPMS?

1. View all medicine
2. Place order
3. View order
4. View Personal Information
5. Search medicine
6. Logout

Choose the operation from the given options: 3

Username: Kathy Sierra
Home address: 14, Jalan Merdeka, Bandar Hilir, 75000 Melaka
Email: kathysierra@gmail.com
Contact number: 012-609 6790
Medicine name: Alendronate
Quantity: 5
```

Figure 13: Registered Customer View Order

If the customer chooses the option 3 that is “View order”, the customer can see what order have they ordered before. The system would show the username, home address, email and contact number of the customer. The medicine name and quantity of the medicine which had been ordered will also be shown.

View Personal Information

```
What do you want to do by using OPMS?

1. View all medicine
2. Place order
3. View order
4. View Personal Information
5. Search medicine
6. Logout

Choose the operation from the given options: 4

User ID: 5
Username: Kathy Sierra
Password: 123456
Home address: 14, Jalan Merdeka, Bandar Hilir, 75000 Melaka
Email address: kathysierra@gmail.com
Contact number: 012-609 6790
Gender: F
Date of birth: 05/07/2003
```

Figure 14: Registered Customer View Personal Information

If the customer chooses the option 4 that is “View Personal Information”, the customer can see the details of themselves which were recorded in the system. The system would show the user ID, username, password, home address, email address, contact number, gender and date of birth of the customer.

Search medicine

```
What do you want to do by using OPMS?

1. View all medicine
2. Place order
3. View order
4. View Personal Information
5. Search medicine
6. Logout

Choose the operation from the given options: 5
Type what you want to search: Alendronate

Medicine name: Alendronate
Expire date: 31/12/2024
Price: RM 3
Specifications: used to treat & prevent osteoporosis
Directions of use: It should be taken as soon as you get out of bed in the morning and at least 30 minutes before any food
Remark: no remark
```

Figure 15: Registered Customer Search Medicine Through Medicine Name

If the customer chooses the option 5 that is “Search medicine”, the customer will need to type the name of the medicine and they will can see the details of the medicine. The system would show the medicine name, expire date, price, specifications, directions of use and remark of the medicine. If the customer doesn’t know the name of the medicine, they can also type the keyword and the system will also provide the details of the medicine.

```
What do you want to do by using OPMS?

1. View all medicine
2. Place order
3. View order
4. View Personal Information
5. Search medicine
6. Logout

Choose the operation from the given options: 5
Type what you want to search: antibiotic

Medicine name: Roxithromycin
Expire date: 31/12/2022
Price: RM 320
Specifications: an antibiotic that is used to treat infections
Directions of use: take 1 tablets 2 times daily before food
Remark: controlled medicine
```

Figure 16: Registered Customer Search Medicine Through Keyword

Logout

```
What do you want to do by using OPMS?
1. View all medicine
2. Place order
3. View order
4. View Personal Information
5. Search medicine
6. Logout
Choose the operation from the given options: 6

Who are you?
1. Admin
2. New Customer
3. Registered Customer
Choose the operation from the given options:
```

Figure 17: Registered Customer Logout

If the customer chooses the option 6 that is Logout, the system will logout and go back to the menu.

IV. Admin

```
Who are you?
1. Admin
2. New Customer
3. Registered Customer

Choose the operation from the given options: 1
Please enter your username and password.

Username: admin
Password: admin

Login Successful!!!

What do you want to do by using OPMS?
1. Upload medicine detail in system
2. View all medicine
3. Modify medicine information
4. Delete medicine
5. Search medicine
6. View orders of customers
7. Search orders of customers
8. Logout

Choose the operation from the given options:
```

Figure 18: Admin Menu

If the admin of OPMS wanted to make any changes or wanted to use this system, they can choose the option 1 in the menu. After choosing the option 1, the admin has to login to the system with the username and password (e.g. admin, admin). They can choose what they want to do by using OPMS. They can choose the option 1 to upload medicine detail in system, the option 2 to view all medicine, the option 3 to modify medicine information, the option 4 to delete medicine, the option 5 to search medicine in the system, the option 6 to view all orders of customers, the option 7 to search orders of customer and also the option 8 to logout of their account.

Upload medicine detail in the system

```

What do you want to do by using OPMS?

1. Upload medicine detail in system
2. View all medicine
3. Modify medicine information
4. Delete medicine
5. Search medicine
6. View orders of customers
7. Search orders of customers
8. Logout

Choose the operation from the given options: 1

How many medicine do you want to upload?1

Medicine name: Abacavir
Expire date (DD/MM/YYYY): 05/07/2022
Price: RM_11.59
Specification: used_to_prevent_and_treat_HIV/AIDS
Directions of use: taken_once_or_twice_daily_with_or_without_food
Remark: controlled_medicine

You have added a new medicine!

```

Figure 19: Admin Upload Medicine Details

If the admin chooses the option 1 that is “Upload medicine detail in system”, the admin can add new medicine here. After choosing the option 1, admin will need to let the system know how many medicines does the admin wanted to upload (e.g. 1). After that, admin will need to provide the details of the medicine. The details included medicine name (e.g. Abacavir), expire date (e.g. 05/07/2022), price (e.g. RM 11.59), specification (e.g. used to prevent and treat HIV/AIDS), directions of use (e.g. taken once or twice daily with or without food) and remark (e.g. controlled medicine). The order made by the customer will be saved in medicine text file.

Roxithromycin	31/12/2022	RM_320	an_antibiotic_that_is_used_to_treat_infections	take_1_tablets_2_times_daily_before_food	controlled_medicine
Isotretinoin	31/12/2023	RM_32	used_to_treat_severe_acne	take_1_tablets_1_times_daily_before_food	controlled_medicine
Alendronate	31/12/2024	RM_3	used_to_treat_prevent_osteoporosis	It_should_be_taken_as_soon_as_you_get_out_of_bed_in_the_morning_and_at_least	
Abacavir	05/07/2022	RM_11.59	used_to_prevent_and_treat_HIV/AIDS	taken_once_or_twice_daily_with_or_without_food	controlled_medicine

Figure 20: Medicine Text File

After the medicine had been saved, the system will go back to allow the admin to choose what they want to do next.

View all medicine

```
Choose the operation from the given options: 2

Medicine name: Roxithromycin
Expire date: 31/12/2022
Price: RM 320
Specifications: an antibiotic that is used to treat infections
Directions of use: take 1 tablets 2 times daily before food
Remark: controlled medicine

Medicine name: Isotretinoin
Expire date: 31/12/2023
Price: RM 32
Specifications: used to treat severe acne
Directions of use: take 1 tablets 1 times daily before food
Remark: controlled medicine

Medicine name: Alendronate
Expire date: 31/12/2024
Price: RM 3
Specifications: used to treat & prevent osteoporosis
Directions of use: It should be taken as soon as you get out of bed in the morning and at least 30 minutes before any food
Remark: no remark

Medicine name: Amoxicillin
Expire date: 05/07/2023
Price: RM 11.30
Specifications: used to treat a wide variety of bacterial infections
Directions of use: take it 3 times a day
Remark: no remark
```

Figure 21: Admin View Medicine

If the admin chooses the option 2 that is “View all medicine”, all the medicine recorded in the system will be shown to the admin. The system would show the medicine and its details which were recorded inside the medicine text file. After showing all the medicines and also the details of the medicines, the system will go back to allow admin to choose what they want to do next.

Modify medicine information

```
What do you want to do by using OPMS?

1. Upload medicine detail in system
2. View all medicine
3. Modify medicine information
4. Delete medicine
5. Search medicine
6. View orders of customers
7. Search orders of customers
8. Logout

Choose the operation from the given options: 3

What do you want to modify?

1. Medicine name
2. Expire date
3. Price
4. Specification
5. Directions of use
6. Remark

Choose the operation from the given options: 1
Type actual medicine name: Abacavir
Type new medicine name: Amoxicillin

Medicine name modified sucessfully!
```

Figure 22: Admin Modify Medicine Information (Medicine name)

If the admin chooses the option 3 that is “Modify medicine information”, the admin will next need to choose what they want to modify. They can type 1 to modify the medicine name, type 2 to modify the expire date, type 3 to modify the price, type 4 to modify the specification, type 5 to modify the directions of use or type 6 to modify the remark of the medicine.

If the admin type 1, they will need to type the actual medicine name and also the new medicine name to modify the medicine name. The system would show “Medicine name modified successfully!” after the medicine name had been modified successfully in medicine text file. After that, the system will go back to allow admin to choose what they want to do next.

```
Choose the operation from the given options: 3

What do you want to modify?

    1. Medicine name
    2. Expire date
    3. Price
    4. Specification
    5. Directions of use
    6. Remark

Choose the operation from the given options: 2
Type medicine name: Amoxicillin
Type new expire date: 05/07/2023

Expire date modified sucessfully!
```

Figure 23: Admin Modify Medicine Information (Expire Date)

If the admin type 2 to modify the expire date, they will need to type the medicine name and also the new expire date. The system would show “Expire date modified successfully!” after the expire date had been modified successfully in medicine text file. After that, the system will go back to allow admin to choose what they want to do next.

```
Choose the operation from the given options: 3

What do you want to modify?

    1. Medicine name
    2. Expire date
    3. Price
    4. Specification
    5. Directions of use
    6. Remark

Choose the operation from the given options: 3
Type medicine name: Amoxicillin
Type new medicine price: RM 11.30
```

Figure 24: Admin Modify Medicine Information (Price)

If the admin type 3 to modify the price, they will need to type the medicine name and also the new price. The system would show “Medicine price modified successfully!” after the price had been modified successfully in medicine text file. After that, the system will go back to allow admin to choose what they want to do next.

```
Choose the operation from the given options: 3

What do you want to modify?

    1. Medicine name
    2. Expire date
    3. Price
    4. Specification
    5. Directions of use
    6. Remark

Choose the operation from the given options: 4
Type medicine name: Amoxicillin
Type new medicine specifications: used_to_treat_a_wide_variety_of_bacterial_infections

Medicine specifications modified sucessfully!
```

Figure 25: Admin Modify Medicine Information (Specification)

If the admin type 4 to modify the specification, they will need to type the medicine name and also the new specification. The system would show “Medicine specification modified successfully!” after the specification had been modified successfully in medicine text file. After that, the system will go back to allow admin to choose what they want to do next.

```
What do you want to modify?

    1. Medicine name
    2. Expire date
    3. Price
    4. Specification
    5. Directions of use
    6. Remark

Choose the operation from the given options: 5
Type medicine name: Amoxicillin
Type new medicine directions of use: take_it_3_times_a_day

Medicine directions of use modified sucessfully!
```

Figure 26: Admin Modify Medicine Information (Directions of Use)

If the admin type 5 to modify the directions of use, they will need to type the medicine name and also the new directions of use. The system would show “Medicine directions of use modified successfully!” after the directions of use had been modified successfully in medicine text file. After that, the system will go back to allow admin to choose what they want to do next.

```
Choose the operation from the given options: 3

What do you want to modify?

    1. Medicine name
    2. Expire date
    3. Price
    4. Specification
    5. Directions of use
    6. Remark

Choose the operation from the given options: 6
Type medicine name: Amoxicillin
Type new medicine remark: no_remark

Medicine remark modified sucessfully!
```

Figure 27: Admin Modify Medicine Information (Remark)

If the admin type 6 to modify the remark, they will need to type the medicine name and also the new remark. The system would show “Medicine remark modified successfully!” after the remark had been modified successfully in medicine text file. After that, the system will go back to allow admin to choose what they want to do next.

Delete medicine

```
What do you want to do by using OPMS?  
1. Upload medicine detail in system  
2. View all medicine  
3. Modify medicine information  
4. Delete medicine  
5. Search medicine  
6. View orders of customers  
7. Search orders of customers  
8. Logout  
Choose the operation from the given options: 4  
Enter name of medicine you want to delete: Abacavir  
Abacavir deleted successfully.
```

Figure 28: Admin Delete Medicine

If the admin chooses the option 4 that is “Delete medicine”, the admin will need to type the name or keyword of the medicine and the medicine will be deleted from medicine text file. The system will not show the medicine to the customer after the medicine had been deleted by admin.

Search medicine

```
What do you want to do by using OPMS?

1. Upload medicine detail in system
2. View all medicine
3. Modify medicine information
4. Delete medicine
5. Search medicine
6. View orders of customers
7. Search orders of customers
8. Logout

Choose the operation from the given options: 5
Type what you want to search: alendronate

Medicine name: Alendronate
Expire date: 31/12/2024
Price: RM 3
Specifications: used to treat & prevent osteoporosis
Directions of use: It should be taken as soon as you get out of bed in the morning and at least 30 minutes before any food
Remark: no remark
```

Figure 29: Admin Search Medicine Through Medicine Name

If the admin chooses the option 5 that is “Search medicine”, the admin will need to type the name of the medicine and they will can see the details of the medicine. The system would show the medicine name, expire date, price, specifications, directions of use and remark of the medicine. If the admin doesn’t know the name of the medicine, they can also type the keyword and the system will also provide the details of the medicine.

```
What do you want to do by using OPMS?

1. Upload medicine detail in system
2. View all medicine
3. Modify medicine information
4. Delete medicine
5. Search medicine
6. View orders of customers
7. Search orders of customers
8. Logout

Choose the operation from the given options: 5
Type what you want to search: anti

Medicine name: Roxithromycin
Expire date: 31/12/2022
Price: RM 320
Specifications: an antibiotic that is used to treat infections
Directions of use: take 1 tablets 2 times daily before food
Remark: controlled medicine
```

Figure 30: Admin Search Medicine Through Keyword

View orders of customer

```
Choose the operation from the given options: 6

Customer username: Admin
Home address: OCEAN Sdn Bhd
Email: opms@ocean.com.my
Contact number: 012-345 6789
Medicine name: Alendronate
Quantity: 4

Customer username: Admin
Home address: OCEAN Sdn Bhd
Email: opms@ocean.com.my
Contact number: 012-345 6789
Medicine name: Isotretinoin
Quantity: 4

Customer username: Jayden
Home address: OCEAN Sdn Bhd
Email: opms@ocean.com.my
Contact number: 012-345 6789
Medicine name: Roxithromycin
Quantity: 8

Customer username: Admin
Home address: OCEAN Sdn Bhd
Email: opms@ocean.com.my
Contact number: 012-345 6789
Medicine name: Alendronate
Quantity: 3

Customer username: Kathy Sierra
Home address: 14, Jalan Merdeka, Bandar Hilir, 75000 Melaka
Email: kathysierra@gmail.com
Contact number: 012-609 6790
Medicine name: Alendronate
Quantity: 5
```

Figure 31: Admin View Orders of Customer

If the admin chooses the option 6 that is “View orders of customer”, the admin can see all the order made by customer. The system would show the username, home address, email and contact number of all the customer. The medicine name and quantity of the medicine which had been ordered by the customer will also be shown.

Search orders of customer

```
What do you want to do by using OPMS?

1. Upload medicine detail in system
2. View all medicine
3. Modify medicine information
4. Delete medicine
5. Search medicine
6. View orders of customers
7. Search orders of customers
8. Logout

Choose the operation from the given options: 7
How do you want to search from?

1. Search by customer name
2. Search by medicine name

Choose the operation from the given options: 1

Customer name: Kathy Sierra

Customer username: Kathy Sierra
Home address: 14, Jalan Merdeka, Bandar Hilir, 75000 Melaka
Email: kathysierra@gmail.com
Contact number: 012-609 6790
Medicine name: Alendronate
Quantity: 5
```

Figure 32: Admin Search Orders of Customers (Customer Name)

If the admin chooses the option 7 that is “Search orders of customer”, the admin will next need to choose whether they want to search by customer name or medicine name. They can type 1 to search by customer name or 2 to search by medicine name. If the admin type 1, they will need to type the customer name and they can see all the order made by this customer. The system would show the username, home address, email and contact number of the customer. The medicine name and quantity of the medicine which had been ordered by the customer will also be shown.


```
Choose the operation from the given options: 7
How do you want to search from?

    1. Search by customer name
    2. Search by medicine name

Choose the operation from the given options: 2

Medicine name: Alendronate

Customer username: Admin
Home address: OCEAN Sdn Bhd
Email: opms@ocean.com.my
Contact number: 012-345 6789
Medicine name: Alendronate
Quantity: 4

Customer username: Admin
Home address: OCEAN Sdn Bhd
Email: opms@ocean.com.my
Contact number: 012-345 6789
Medicine name: Alendronate
Quantity: 3

Customer username: Kathy Sierra
Home address: 14, Jalan Merdeka, Bandar Hilir, 75000 Melaka
Email: kathysierra@gmail.com
Contact number: 012-609 6790
Medicine name: Alendronate
Quantity: 5
```

Figure 33: Admin Search Orders of Customers (Medicine Name)

If the admin type 2 to search by medicine name, they will need to type the medicine name and they can see all the order made to this medicine by the customer. The system would show the username, home address, email and contact number of the customer. The medicine name and quantity of the medicine which had been ordered by the customer will also be shown.

Logout

```
What do you want to do by using OPMS?
1. Upload medicine detail in system
2. View all medicine
3. Modify medicine information
4. Delete medicine
5. Search medicine
6. View orders of customers
7. Search orders of customers
8. Logout
Choose the operation from the given options: 8

Who are you?
1. Admin
2. New Customer
3. Registered Customer
Choose the operation from the given options:
```

Figure 34: Admin Logout

If the admin chooses the option 8 that is “Logout”, the system will logout and go back to the menu.

Conclusion

In conclusion, some research and investigation have been done throughout this entire report. Many knowledges about Python had been gained while doing this assignment. This assignment had been finally completed. I am very happy. I am very grateful to Mr. Viknesh as he was very patience when we ask him question and he also teaches us in a very interesting way.

References

- How to delete a line from a file in Python.* (2022, March 7). Retrieved from ADAM SMITH:
<https://www.adamsmith.haus/python/answers/how-to-delete-a-line-from-a-file-in-python>
- Python - Global Variables.* (2020, February 4). Retrieved from W3Schools:
https://www.w3schools.com/python/python_variables_global.asp
- Python String split() Method.* (2018, June 20). Retrieved from W3Schools:
https://www.w3schools.com/python/ref_string_split.asp
- techWithId. (2021, May 19). *Python Login System: Using a text file (Beginners Project)*. Retrieved from
YouTube:
https://www.youtube.com/watch?v=dR_cDapPWY&ab_channel=techWithId
- vic, c. w. (2020, June 17). *python login system tutorial (For beginners) Python Tutorial*. Retrieved from
YouTube:
https://www.youtube.com/watch?v=1e9okb_gVXc&ab_channel=codewithvic