



GROUP ASSIGNMENT

CT108-3-1-PYP

PYTHON PROGRAMMING

APU1F2307CS (CYB)

HAND OUT DATE : 25 AUGUST 2023

HAND IN DATE : 30 OCTOBER 2023

WEIGHTAGE : 50%

GROUP MEMBERS:

STUDENT NAME	TP NUMBER
TIEW KHAI EN	TP069872
VIOLET SO WEN LIN	TP075001

Table of Contents

1.0 Introduction	4
1.1 Assumption.....	4
2.0 Design of Program	5
2.1 Pseudocode.....	5
2.2 Flowchart.....	28
login().....	28
admin_menu()	29
staff_menu()	31
add_user().....	33
delete_user()	34
search_user()	35
modify_user()	36
add_hospital().....	38
add_supplier().....	39
modify_supplier()	40
modify_hospital()	42
search_supplier()	44
search_hospital()	45
add_ppe()	46
update_ppe()	47
item_tracking()	49
item_track_1()	50
item_track_2()	51
item_track_3()	52
item_track_4.....	53
search_ppe()	54
MainLogic()	55
3.0 Program Source Code and Explanation	56
3.1 Receive Information	56
3.2 Output Information	57
3.3 Performing Arithmetic.....	58
3.4 Assigning Value to A Piece of Data	59

3.5 Repeat A Group of Actions.....	60
4.0 Screenshots of Sample Input/Output and Explanation	61
 4.1 Login Page.....	61
 4.2 Menu Page	62
 4.3 Add/Delete/Modify/Search User Page	63
 4.4 Update/Search Hospital Page.....	65
 4.5 Update/Search Supplier Page.....	66
 4.6 Update Personal Protective Equipment (PPE) Page	67
 4.7 Item Inventory Tracking Page	68
 4.8 Search Personal Protective Equipment (PPE) Transaction Page	70
5.0 Conclusion	71
6.0 References	72

1.0 Introduction

To track and distribute personal protective equipment (PPE), we created an inventory management system for the Department of Health. The software makes sure that PPE supply management is accurate, transparent, and efficient. An intuitive user interface makes it simple for administrators and employees to update, navigate, and check inventories—thereby averting any problems during unexpected spikes in demand.

1.1 Assumption

There are two different user types in the system: "Admin" and "Staff," each with different roles and levels of access. Security is ensured by a secure login process that uses pre-established passwords. Admins may update PPE records and manage users with a wealth of capabilities. With menu-driven interactions, the system is easy to use and capable of handling and fixing errors.

2.0 Design of Program

2.1 Pseudocode

```

START

DISPLAY blank line
DISPLAY line of 198 hyphens
OPEN file "title.txt" for reading and STORE AS tfile
READ content from tfile into content
DISPLAY content
CLOSE tfile
DISPLAY line of 198 hyphens
DISPLAY "Welcome to Personal Protective Equipment (PPE) Inventory Management System"

FUNCTION login():
    WHILE true:
        DISPLAY "Welcome to Login Page"
        ASK "Please enter your usertype(admin/staff):" and STORE in usertype
        ASK "Please enter the password:" and STORE in userpassword

        IF user type is "ADMIN":
            IF password is "admin123":
                DISPLAY "You are login to admin page"
                CALL admin_Menu()
            ELSE:
                DISPLAY "Wrong password, please try again later..."

        ELIF user type is "STAFF":
            IF password is "staff789":
                DISPLAY "You are login to staff page"
                CALL staff_menu()
            ELSE:
                DISPLAY "Wrong password, please try again later..."
                RECALL login()

    ELSE:
        DISPLAY "Invalid input, please try again..."
        RECALL login()

ENDFUNCTION
CALL login()

```

```

FUNCTION admin_Menu():
    DISPLAY "You are in the Menu Page"
    DISPLAY "Press 1 - Add new user"
    DISPLAY "Press 2 - Delete user"
    DISPLAY "Press 3 - Modify user"
    DISPLAY "Press 5 - Update Hospital Details"
    DISPLAY "Press 6 - Update Supplier Details"
    DISPLAY "Press 7 - Search Hospital"
    DISPLAY "Press 8 - Search Supplier"
    DISPLAY "Press 9 - Update PPE"
    DISPLAY "Press 10 - Item inventory tracking"
    DISPLAY "Press 11 - Search PPE transaction"
    DISPLAY "Press 12 - Logout"
    ASK "Please enter your option here:" and STORE in option
    IF option is "1":
        CALL add_user()
    ELIF option is "2":
        CALL delete_user()
    ELIF option is "3":
        CALL modify_user()
    ELIF option is "4":
        CALL search_user()
    ELIF option is "5":
        CALL modify_hospital()
    ELIF option is "6":
        CALL modify_supplier()
    ELIF option is "7":
        CALL search_hospital()
    ELIF option is "8":
        CALL search_supplier()
    ELIF option is "9":
        CALL update_ppe()
    ELIF option is "10":
        CALL item_tracking()
    ELIF option is "11":
        CALL search_item()
    ELIF option is "12":

```

```

CALL login()

ELSE:
    DISPLAY "Invalid Option"
    DISPLAY "Please enter a valid option"
    RECALL admin_Menu()

ENDFUNCTION

FUNCTION staff_menu():
    DISPLAY "You are now in the Menu page"
    DISPLAY "Press 1 - Update Hospital Details"
    DISPLAY "Press 2 - Update Supplier Details"
    DISPLAY "Press 3 - Search Hospital"
    DISPLAY "Press 4 - Search Supplier"
    DISPLAY "Press 5 - Update PPE"
    DISPLAY "Press 6 - Item inventory tracking"
    DISPLAY "Press 7 - Search PPE transaction"
    DISPLAY "Press 8 - Logout"

    ASK "Please enter your option here:" and STORE in option

    IF option is "1":
        CALL modify_hospital()
    ELIF option is "2":
        CALL modify_supplier()
    ELIF option is "3":
        CALL search_hospital()
    ELIF option is "4":
        CALL search_supplier()
    ELIF option is "5":
        CALL update_ppe()
    ELIF option is "6":
        CALL item_tracking()
    ELIF option is "7":
        CALL search_ppe()
    ELIF option is "8":
        CALL login()

    ELSE:
        DISPLAY "Invalid Option"
        DISPLAY "Please enter a valid option"

```

```

RECALL staff_menu()

ENDFUNCTION

FUNCTION add_user():
    DISPLAY "Do you want to add new user?"
    DISPLAY "Press Y - add new user"
    DISPLAY "Press N - return back to Menu"
    ASK "Please enter your option here:" and STORE in adduser
    IF option is "Y":
        OPEN "users.txt" for reading
        INITIALIZE empty list userlist

        WHILE true:
            ASK "Please enter the staff (IC/Passport):" and STORE in userid
            IF user ID not in userlist:
                BREAK
            ELSE:
                DISPLAY "Invalid Input"
                DISPLAY "ID already exists"

        WHILE true:
            ASK "Please enter the staff name:" and STORE in name
            BREAK

        WHILE true:
            ASK "Please enter your password(at least 8 characters):" and STORE in pw
            IF pw length < 8:
                DISPLAY "Invalid password, password at least 8 characters"
                ASK "Please enter to continue"
            ELSE:
                BREAK

        WHILE true:
            ASK "Please enter the usertype(admin or staff):" and STORE in type
            IF user type not in ["admin", "staff"]:
                DISPLAY "Invalid input, please enter admin or staff"
                ASK "Please enter to re-enter"

```

```

ELSE:
    BREAK

ELIF option is "N":
    CALL admin_Menu()

ELSE:
    DISPLAY "Invalid input"
    RECALL add_user()

OPEN "users.txt" in append mode as userfile
    WRITE formatted user details (userid, name, pw, type) to userfile
    CLOSE userfile
    DISPLAY "The new user -- [name], [userid], [type] added successfully."
    CALL admin_Menu()

ENDFUNCTION

FUNCTION delete_user():
    DISPLAY "Do you want to REMOVE USER?"
    DISPLAY "Press Y - delete user"
    DISPLAY "Press N - return back to Menu"
    ASK "Please enter your option here:" and STORE in option

    IF option is "Y":
        ASK "Please enter the User ID you would like to remove:" and STORE in user_id
        INITIALIZE empty list userlist

        OPEN "users.txt" for reading
        FOR each line in userfile:
            SPLIT line by spaces and APPEND to userlist
            CLOSE userfile

            SET flag to -1
            DISPLAY length of userlist

        FOR each user in userlist:
            IF User ID matches:

```

```

        SET flag to current user's index
        BREAK

IF flag is not -1:
    REMOVE user from userlist based on flag
    OPEN "users.txt" for writing
        WRITE updated userlist back to file
        CLOSE file
    DISPLAY "User removed successfully"

ELSE:
    DISPLAY "User not found"
    RECALL delete_user()

ELIF option is "N":
    CALL admin_Menu()

ELSE:
    DISPLAY "Invalid input"
    DISPLAY "Please enter Y/N ONLY"
    RECALL delete_user()

END FUNCTION

FUNCTION search_user():
    DISPLAY "Do you want to search user?"
    DISPLAY "Press Y - search user"
    DISPLAY "Press N - return back to Menu"
    ASK "Please enter your option here:" and STORE in searchuser
    IF searchuser option is "Y":
        ASK "Please enter the User ID you would like to search:"
        INITIALIZE empty list userlist
        OPEN "users.txt" for reading

        SET flag to False

        FOR each line in userfile:
            REMOVE trailing whitespace from line
            IF chooseuser is in line when split by spaces:

```

```

SET flag to True
DISPLAY line with separators and user details header
RECALL search_user()

IF flag is False:
    DISPLAY "No Staff record under this ID."
    RECALL search_user()

ELIF searchuser option is "N":
    CALL admin_Menu()

ELSE:
    DISPLAY "Invalid input"
    DISPLAY "Please enter Y/N ONLY"
    RECALL search_user()

ENDFUNCTION

FUNCTION modify_user():
    DISPLAY "Do you want to MODIFY USER?"
    DISPLAY "Press Y - modify user"
    DISPLAY "Press N - return back to Menu"
    ASK "Please enter your option here:" and STORE in option

    IF option is "Y":
        ASK "Please enter the User ID you would like to search:" and STORE in user_id
        INITIALIZE empty userlist

        OPEN "users.txt" for reading
        READ all users and APPEND to userlist
        CLOSE file

        SET flag to -1 (to indicate user not found)

        FOR each user in userlist:
            IF User ID matches:
                SET flag to current user's index
                BREAK

```

```

IF flag is not -1 (user found):
    REMOVE user from userlist based on flag
    OPEN "users.txt" for writing
    WRITE updated userlist back to file
    CLOSE file

ELSE:
    DISPLAY "User not found"
    RECALL modify_user()

OPEN "users.txt" for reading
INITIALIZE empty userlist

WHILE true:
    ASK "Please enter the user new (IC/Passport):" and STORE in userid
    IF new user ID not in userlist:
        BREAK
    ELSE:
        DISPLAY "Invalid Input"
        DISPLAY "ID already exists"

WHILE true:
    ASK "Please enter the user name:" and STORE in name
    BREAK

WHILE true:
    ASK "Please enter new password(at least 8 characters):" and STORE in pw
    IF password length < 8:
        DISPLAY "Invalid password, password at least 8 characters"
        ASK "Please enter to continue"
    ELSE:
        BREAK

WHILE true:
    ASK "Please enter the usertype(admin or staff):" and STORE in type
    IF user type not in ["admin", "staff"]:
        DISPLAY "Invalid input, please enter admin or staff"

```

```

ASK "Please enter to re-enter"
ELSE:
    BREAK

OPEN "users.txt" in append mode
    WRITE new user details to file
    CLOSE file
    DISPLAY success message
    CALL admin_Menu()

ELIF option is "N":
    CALL admin_Menu()

ELSE:
    DISPLAY "Invalid input"
    DISPLAY "Please enter Y/N ONLY"
    RECALL modify_user()
ENDFUNCTION

FUNCTION add_hospital():
    DISPLAY "Do you want to add hospital?"
    DISPLAY "Press Y - Add hospital"
    DISPLAY "Press N - return back to Menu"
    ASK "Please enter your option here:" and STORE in addhosp

    IF addhosp option is "Y":
        OPEN "hospital.txt" for reading
        INITIALIZE empty list hosplist

        WHILE true:
            ASK "Please enter the hospital code(H1/H2/H3):" and STORE in hospcode
            IF hospital code not in hosplist:
                BREAK
            ELSE:
                DISPLAY "Invalid Input"
                DISPLAY "ID already exists"

```

```

WHILE true:
    ASK "Please enter the hospital name:" and STORE in name
    BREAK

WHILE true:
    ASK "Please enter hospital contact number:" and STORE in pn
    BREAK

WHILE true:
    ASK "Please enter the name of person in charge:" and STORE in pic
    BREAK

ELIF addhosp option is "N":
    CALL staff_menu()

ELSE:
    DISPLAY "Invalid input"
    RECALL add_hospital()

OPEN "supliers.txt" in append mode
    WRITE formatted details (hospcode, name, pn, pic) to userfile separated by tabs
    CLOSE userfile
    DISPLAY "The new user -- [hospcode], [name], [pn], [pic] added successfully."
    CALL staff_menu()

ENDFUNCTION

FUNCTION add_supplier():
    DISPLAY "Do you want to add supplier?"
    DISPLAY "Press Y - add supplier"
    DISPLAY "Press N - return back to Menu"
    ASK "Please enter your option here:" and STOREE in addsupp
    IF addsupp is "Y":
        OPEN "supliers.txt" for reading as ufile
        INITIALIZE empty list supplest

        WHILE true:
            ASK "Please enter the supplier code (S1/S2/S3):" and STORE in suppcode

```

```

IF suppcode is not in supplist:
    BREAK
ELSE:
    DISPLAY "Invalid Input"
    DISPLAY "Code already exists"

WHILE true:
    ASK "Please enter the supplier name:" and STORE in name
    BREAK

WHILE true:
    ASK "Please enter supplier contact number:" and STORE in pn
    BREAK

WHILE true:
    ASK "Please enter the name of person in charge:" and STORE in pic
    BREAK

ELIF addsupp option is "N":
    CALL admin_Menu()

ELSE:
    DISPLAY "Invalid input"
    RECALL add_supplier()

OPEN "supliers.txt" in append mode as userfile
    WRITE formatted details (suppcode, name, pn, pic) to userfile
    CLOSE userfile
    DISPLAY "The new user -- "+suppcode+", "+name+", "+pn+", "+pic+" added successfully."
    CALL admin_Menu()

ENDFUNCTION

FUNCTION modify_hospital():
    DISPLAY "Do you want to Update Hospital Details?"
    DISPLAY "Press Y - Update hospital details"
    DISPLAY "Press N - return back to Menu"
    ASK "Please enter your option here:" and STORE in option

```

IF option is "Y":

ASK "Please enter the hospital code you like to update:" and STORE in hospcode
 INITIALIZE empty list hosplist

OPEN "hospital.txt" for reading as hospfile

READ lines from hospfile and STORE in hosplist as lists of strings

SET flag to -1

FOR each index cnt in range of hosplist:

IF hospcode matches the first element of hosplist at index cnt:

SET flag to cnt

BREAK

IF flag is not -1:

REMOVE hosplist entry at index flag

OPEN "hospital.txt" for writing as hosp_file

WRITE each entry of hosplist to hosp_file as formatted string

ELSE:

DISPLAY "Hospital not found."

RECALL modify_hospital()

OPEN "hospital.txt" for reading as ufile

INITIALIZE empty list hosplist

WHILE true:

ASK "Please enter the hospital code (H1/H2/H3):" and STORE in hospcode

IF hospcode is not in hosplist:

BREAK

ELSE:

DISPLAY "Invalid Input"

DISPLAY "Hospital already exists"

WHILE true:

ASK "Please enter the hospital name:" and STORE in name

BREAK

```

WHILE true:
    ASK "Please enter hospital contact number:" and STORE in pn
    BREAK

WHILE true:
    ASK "Please enter the name of person in charge:" and STORE in pic
    BREAK

OPEN "hospital.txt" for appending as userfile
    WRITE formatted details (hospcode, name, pn, pic) to userfile
    CLOSE userfile
    DISPLAY "The new user -- "+hospcode+", "+name+", "+pn+", "+pic+" added successfully."
    CALL staff_menu()

ELIF option is "N":
    CALL staff_menu()

ELSE:
    DISPLAY "Invalid input"
    DISPLAY "Please enter Y/N ONLY"
    RECALL modify_hospital()
ENDFUNCTION

FUNCTION modify_supplier():
    DISPLAY "Do you want to Update Supplier Details?"
    DISPLAY "Press Y - Update supplier details"
    DISPLAY "Press N - return back to Menu"
    ASK "Please enter your option here:" and STORE in option

    IF option is "Y":
        ASK "Please enter the supplier code you like to update:" and STORE in suppcode
        INITIALIZE empty list supplist

        OPEN "suppliers.txt" for reading as suppfile
        READ lines from suppfile and STORE in supplist as lists of strings

```

```

SET flag to -1
FOR each index cnt in range of supplist:
  IF suppcode matches the first element of supplist at index cnt:
    SET flag to cnt
    BREAK

  IF flag is not -1:
    REMOVE supplist entry at index flag
    OPEN "suppliers.txt" for writing as supp_file
    WRITE each entry of supplist to supp_file as formatted string
  ELSE:
    DISPLAY "Supplier not found."
    RECALL modify_supplier()

  OPEN "suppliers.txt" for reading as ufile
  INITIALIZE empty list supplist

  WHILE true:
    ASK "Please enter the supplier code (S1/S2/S3):" and STORE in suppcode
    IF suppcode is not in supplist:
      BREAK
    ELSE:
      DISPLAY "Invalid Input"
      DISPLAY "Code already exists"

  WHILE true:
    ASK "Please enter the supplier name:" and STORE in name
    BREAK

  WHILE true:
    ASK "Please enter supplier contact number:" and STORE in pn
    BREAK

  WHILE true:
    ASK "Please enter the name of person in charge:" and STORE in pic
    BREAK

```

```

OPEN "supliers.txt" for appending as userfile
    WRITE formatted details (suppcode, name, pn, pic) to userfile
    CLOSE userfile
    DISPLAY "The new user -- "+suppcode+", "+name+", "+pn+", "+pic+" added
            successfully."
    CALL staff_menu()

ELIF option is "N":
    CALL staff_menu()

ELSE:
    DISPLAY "Invalid input"
    DISPLAY "Please enter Y/N ONLY"
    RECALL modify_supplier()
ENDFUNCTION

FUNCTION search_supplier():
    DISPLAY "Do you want to search supplier?"
    DISPLAY "Press Y - search supplier"
    DISPLAY "Press N - return back to Menu"
    ASK "Please enter your option here:" and STORE in searchsupp

    IF searchsupp is "Y":
        ASK "Please enter the supplier code you would like to search:" and STORE in choosesupp
        INITIALIZE empty list supplist
        OPEN "supliers.txt" for reading as suppfile
        SET flag to False

        FOR each line in suppfile:
            SPLIT line by ":" and STORE in a list
            IF choosesupp is in the list:
                SET flag to True
                DISPLAY formatted supplier details
                CALL staff_menu()

        IF flag is False:
            DISPLAY "No Supplier record under this code."
            RECALL search_supplier()

```

```

ELIF searchsupp is "N":
    CALL staff_menu()

ELSE:
    DISPLAY "Invalid input"
    DISPLAY "Please enter Y/N ONLY"
    RECALL search_supplier()

ENDFUNCTION

FUNCTION search_hospital():
    DISPLAY "Do you want to search hospital?"
    DISPLAY "Press Y - search hospital"
    DISPLAY "Press N - return back to Menu"
    ASK "Please enter your option here:" and STORE in searchhosp

    IF searchhosp is "Y":
        ASK "Please enter the hospital code you would like to search:" and STORE in choosehosp
        INITIALIZE empty list hospList

        OPEN "hospital.txt" for reading as hospfile
        SET flag to False

        FOR each line in hospfile:
            SPLIT line by ":" and STORE in a list
            IF choosehosp is in the list:
                SET flag to True
                DISPLAY formatted hospital details
                CALL staff_menu()

        IF flag is False:
            DISPLAY "No Hospital record under this code."
            RECALL search_hospital()

    ELIF searchhosp is "N":
        CALL staff_menu()

```

```

ELSE:
    DISPLAY "Invalid input"
    DISPLAY "Please enter Y/N ONLY"
    RECALL search_hospital()

ENDFUNCTION

FUNCTION add_ppe():
    DISPLAY "Do you want to add PPE?"
    DISPLAY "Press Y - add ppe"
    DISPLAY "Press N - return back to Menu"
    ASK "Please enter your option here:" and STORE in addppe

    IF addppe is "Y":
        OPEN "ppe.txt" for reading as file
        INITIALIZE empty list ppelist

        WHILE true:
            ASK "Please enter the PPE code:" and STORE in ppecode
            IF ppecode is not in ppelist:
                BREAK
            ELSE:
                DISPLAY "Invalid Input"
                DISPLAY "ID already exists"

        WHILE true:
            ASK "Please enter the Supplier code:" and STORE in suppcode

        WHILE true:
            ASK "Please enter PPE stock:" and STORE in ppestock

    ELIF addppe is "N":
        CALL admin_Menu()

    ELSE:
        DISPLAY "Invalid input"
        CALL add_hospital()

```

```

OPEN "ppe.txt" for appending as ppefile
    WRITE formatted details (ppocode, suppcode, ppestock) to ppefile
    CLOSE ppefile
DISPLAY "PPE added successfully."
RECALL add_ppe()
ENDFUNCTION

FUNCTION update_ppe():
    DISPLAY "Do you want to UPDATE PPE STOCK?"
    DISPLAY "Press Y - Update PPE Stock"
    DISPLAY "Press N - return back to Menu"
    ASK "Please enter your option here:" and STORE in option

    IF option is "Y":
        DISPLAY "HC / FS / MS / GL / GW / SC"
        INITIALIZE empty list ppe_data

        OPEN "ppe.txt" for reading as ppefile
        FOR each line in ppefile:
            SPLIT line by ":" and STORE in a list
            ADD list to ppe_data
            CLOSE ppefile

        ASK "Please enter the PPE code to update the stock quantity:" and STORE in ppe_code
        ASK "Please enter relevant Supplier CODE(S1/S2/S3):" and STORE in ppe_supplier
        DISPLAY "R-Received, D-Distributed"
        ASK "Please enter transaction type(R/D):" and STORE in ppe_trans

        SET flag to -1
        FOR cnt from 0 to length of ppe_data:
            IF ppe_code is in ppe_data[cnt]:
                ASK "Please enter the original stock(in boxes):" and STORE in ppe_ori
                SET flag to cnt
                BREAK

        IF flag is not -1:
            REMOVE the item at index 'flag' from ppe_data

```

```

ASK "Please enter quantity needed for transaction(in boxes):" and STORE in ppe_enter
CONVERT ppe_enter to integer

IF ppe_trans is "R":
    DISPLAY "Supplier code:" and ppe_supplier

ELIF ppe_trans is "D":
    ASK "Please enter relevant Hospital CODE(H1/H2/H3):" and STORE in ppe_hosp
    IF ppe_ori is less than ppe_enter:
        DISPLAY "Quantity in stock is insufficient"
        RECALL update_ppe()

OPEN "ppe.txt" for writing as ppefile
FOR each item in ppe_data:
    WRITE item to ppefile
CLOSE ppefile

IF ppe_trans is "R":
    CALCULATE new stock as ppe_ori + ppe_enter
    APPEND new stock to "ppe.txt"

ELIF ppe_trans is "D":
    CALCULATE new stock as ppe_ori - ppe_enter
    APPEND new stock to "ppe.txt"

DISPLAY "Update Successful"

OPEN "transaction.txt" for appending as ppefile
GET current date and STORE in Date
GET current time and STORE in Time

IF ppe_trans is "R":
    SET ppe_r to ("+", ppe_enter)
    WRITE ppe_code, Date, Time, ppe_trans, ppe_supplier, ppe_r to ppefile

ELIF ppe_trans is "D":
    SET ppe_d to ("-", ppe_enter)

```

```

        WRITE ppe_code, Date, Time, ppe_trans, ppe_hosp, ppe_d to ppefile
        CLOSE ppefile
        DISPLAY "PPE transaction added successfully."
        CALL staff_menu()

    ELSE:
        DISPLAY "Invalid input, please try again later."
        RECALL update_ppe()

    ELIF option is "N":
        CALL staff_menu()

    ELSE:
        DISPLAY "Invalid input"
        RECALL update_ppe
ENDFUNCTION

FUNCTION item_tracking():
    DISPLAY "-----PPE TRACKING-----?"
    DISPLAY "Press 1 - Print ALL PPE stock"
    DISPLAY "Press 2 - Print SHORTAGE PPE"
    DISPLAY "Press 3 - Check PARTICULAR PPE stock"
    DISPLAY "Press 4 - Check PPE Transaction"
    DISPLAY "Press 5 - Return back to Menu"
    ASK "Please enter your option here:" and STORE in option

    IF option is "1":
        CALL item_track_1()
    ELIF option is "2":
        CALL item_track_2()
    ELIF option is "3":
        CALL item_track_3()
    ELIF option is "4":
        CALL item_track_4()
    ELIF option is "5":
        CALL staff_menu()
    ELSE:

```

```

        DISPLAY "Invalid input"
        DISPLAY "Please enter valid option"
        RECALL item_tracking()

ENDFUNCTION

FUNCTION item_track_1():

    DISPLAY "PRINTING PPE STOCK AVAILABLE....."

    OPEN "ppe.txt" for reading as ppe_file
    INITIALIZE empty list ppe_data
    FOR each line in ppe_file:
        SPLIT line by " " and STORE in a list
        ADD list to ppe_data
    CLOSE ppe_file

    DISPLAY line with 91 dashes
    DISPLAY "PPE Code : Supplier Code : Stock Available"
    DISPLAY line with 91 dashes
    DISPLAY ppe_data
    CALL item_tracking()

ENDFUNCTION

FUNCTION item_track_2():

    DISPLAY "PRINTING SHORTAGE PPE STOCK....."

    OPEN "ppe.txt" for reading as ppe_file
    INITIALIZE empty list ppe_data
    FOR each line in ppe_file:
        SPLIT line by ":" and STORE in a list
        ADD list to ppe_data
    CLOSE ppe_file

    FOR each item in ppe_data:
        IF stock of item is less than 25:
            DISPLAY shortage warning for the item
        ELSE:
            DISPLAY availability of the item

```

```

        CALL item_tracking()

ENDFUNCTION

FUNCTION item_track_3():
    DISPLAY "Checking Particular PPE STOCK....."
    ASK "Please enter the PPE code you would like to search:" and STORE in search

    OPEN "ppe.txt" for reading as file
        SET flag to False
        FOR each line in file:
            SPLIT line by ":" and STORE in a list
            IF search is in list:
                SET flag to True
                DISPLAY table headers
                DISPLAY list
                CLOSE file

            IF flag is False:
                DISPLAY "No PPE record under this code."
                RECALL item_track_3()

        CALL item_tracking()

ENDFUNCTION

FUNCTION item_track_4():
    OPEN "transaction.txt" for reading as transfile
        INITIALIZE empty list transaction_list
        FOR each content in transfile:
            SPLIT content by " " and STORE in sublist
            ADD sublist to transaction_list
        CLOSE transfile

        DISPLAY table headers
        FOR each transaction in transaction_list:
            DISPLAY transaction
            CALL item_tracking()

ENDFUNCTION

```

```

        CALL item_tracking()

ENDFUNCTION

FUNCTION item_track_3():
    DISPLAY "Checking Particular PPE STOCK....."
    ASK "Please enter the PPE code you would like to search:" and STORE in search

    OPEN "ppe.txt" for reading as file
        SET flag to False
        FOR each line in file:
            SPLIT line by ":" and STORE in a list
            IF search is in list:
                SET flag to True
                DISPLAY table headers
                DISPLAY list
                CLOSE file

            IF flag is False:
                DISPLAY "No PPE record under this code."
                RECALL item_track_3()

        CALL item_tracking()

ENDFUNCTION

FUNCTION item_track_4():
    OPEN "transaction.txt" for reading as transfile
        INITIALIZE empty list transaction_list
        FOR each content in transfile:
            SPLIT content by " " and STORE in sublist
            ADD sublist to transaction_list
        CLOSE transfile

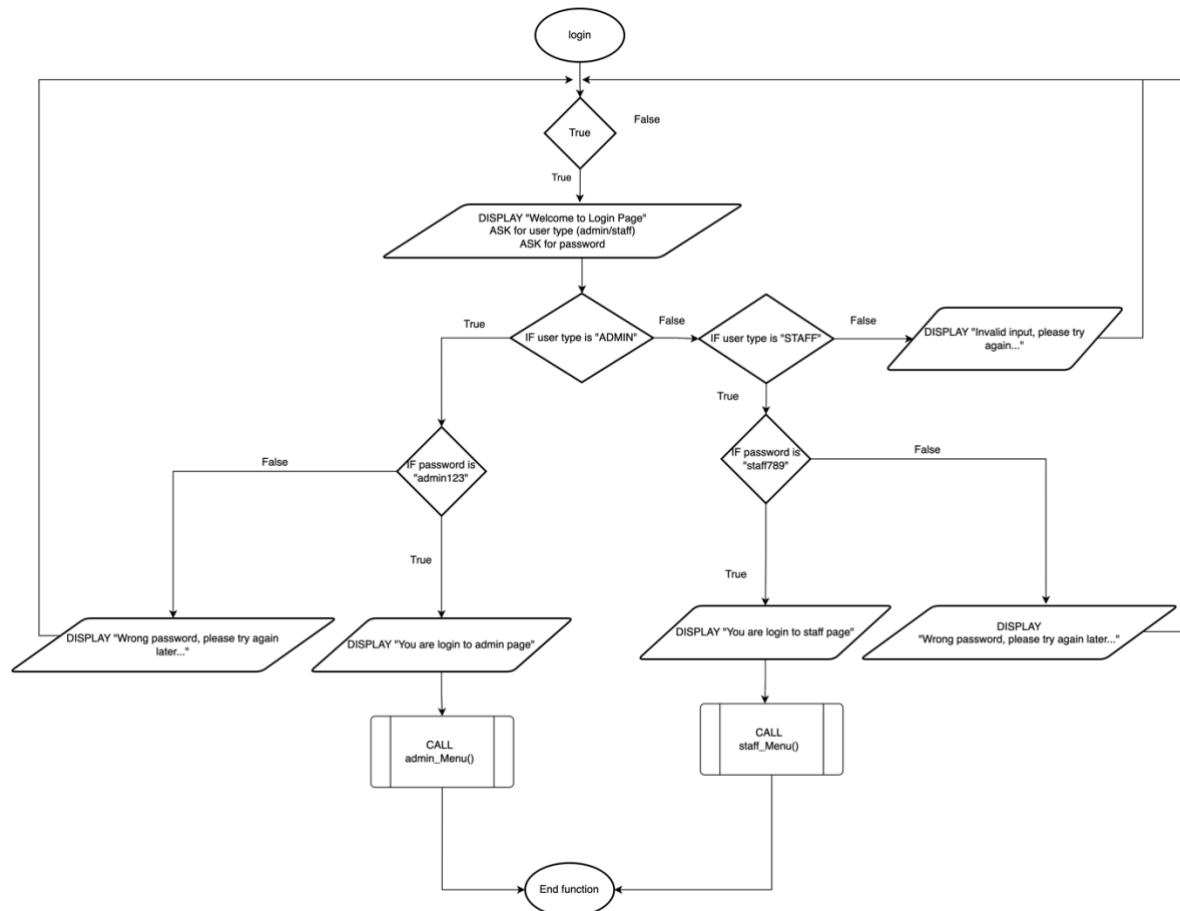
        DISPLAY table headers
        FOR each transaction in transaction_list:
            DISPLAY transaction
            CALL item_tracking()

ENDFUNCTION

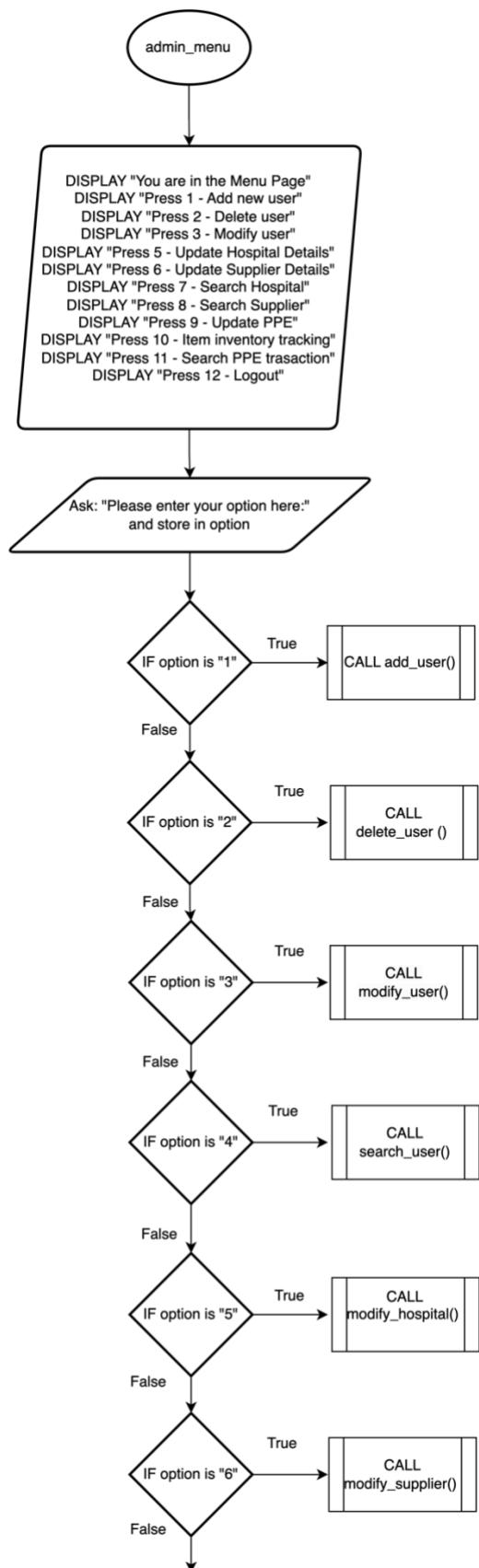
```

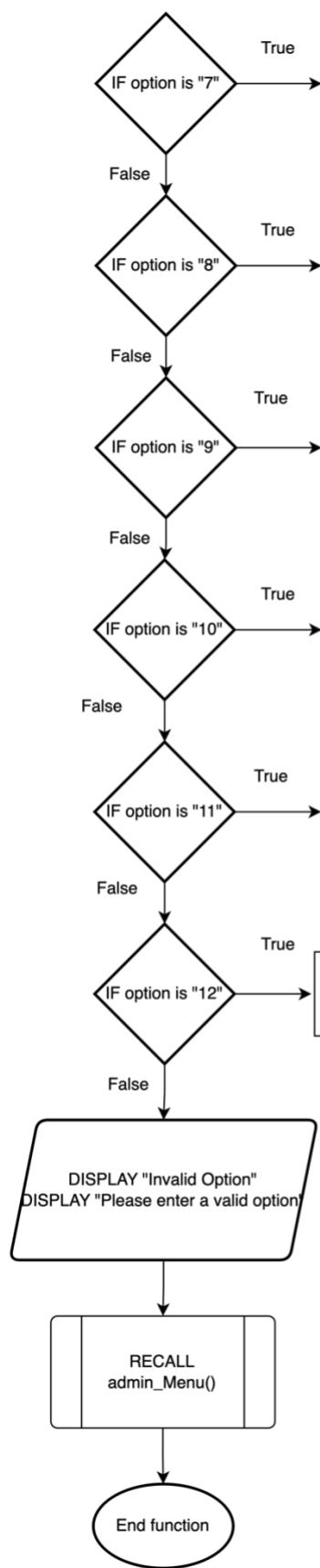
2.2 Flowchart

login()

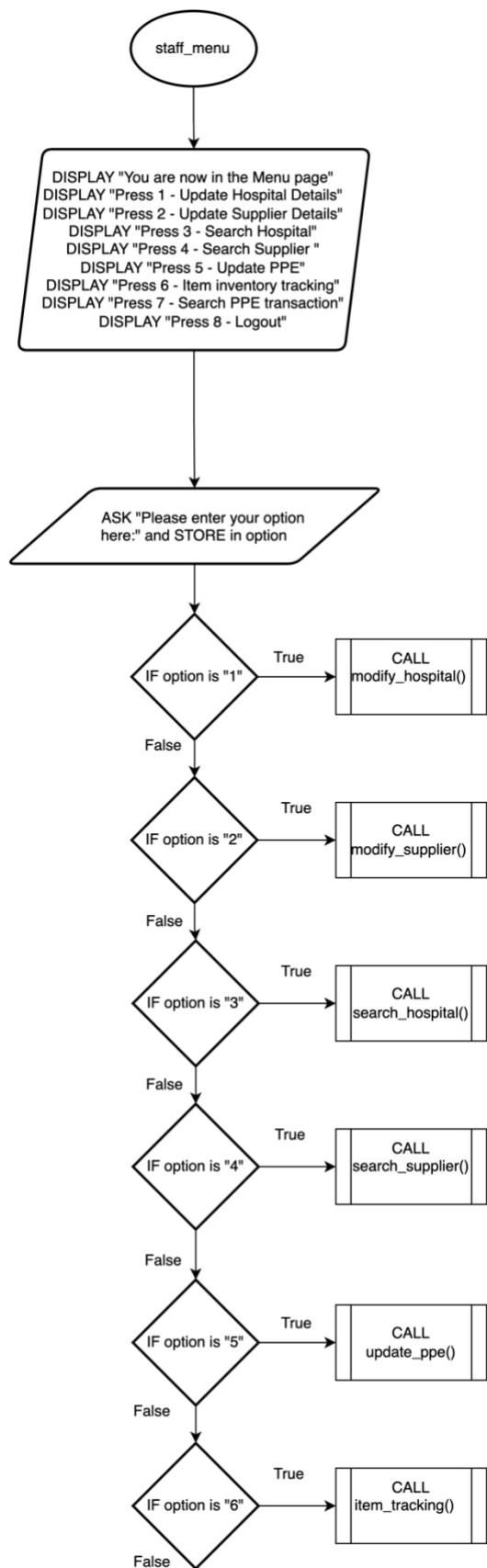


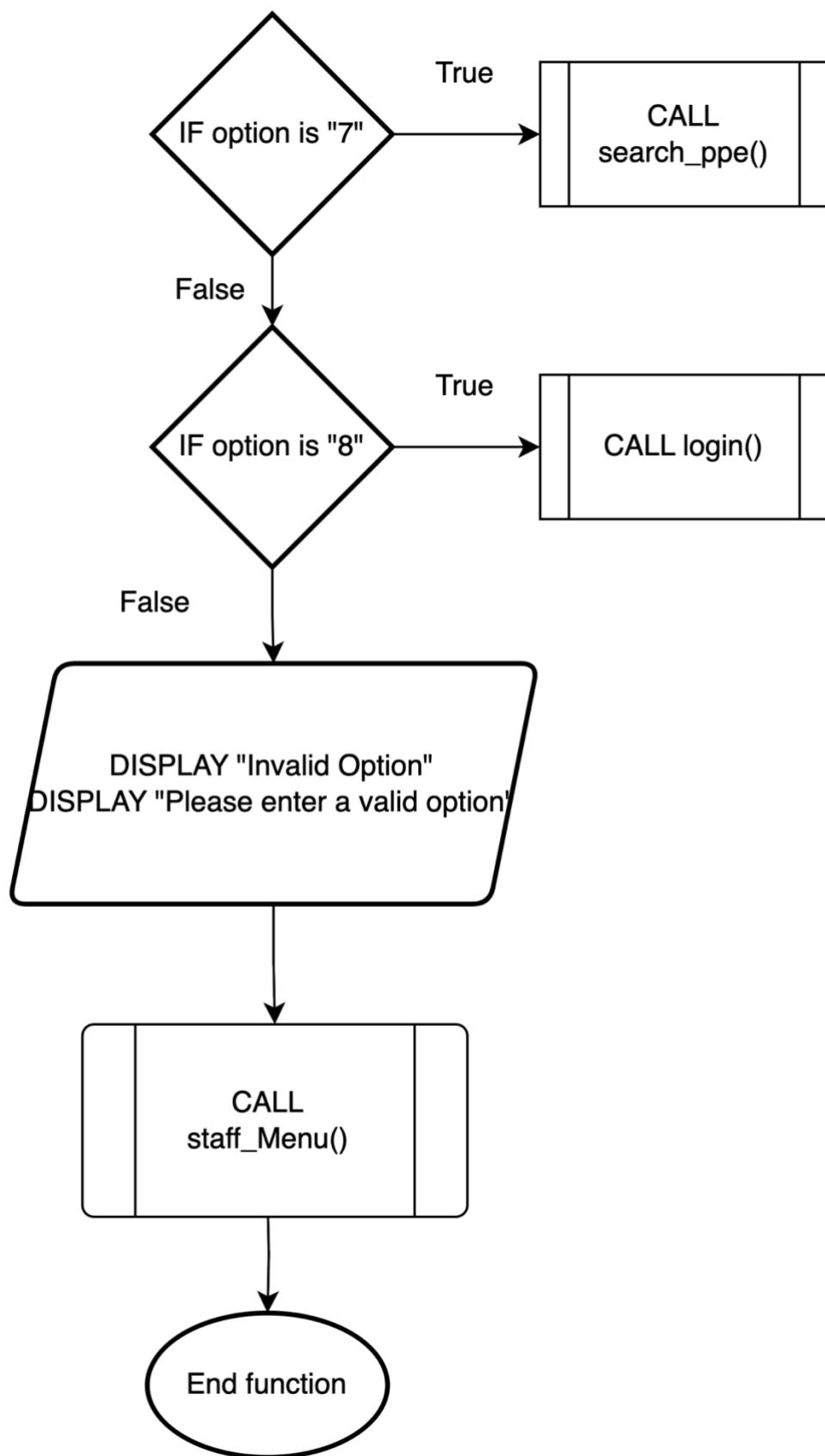
admin_menu()



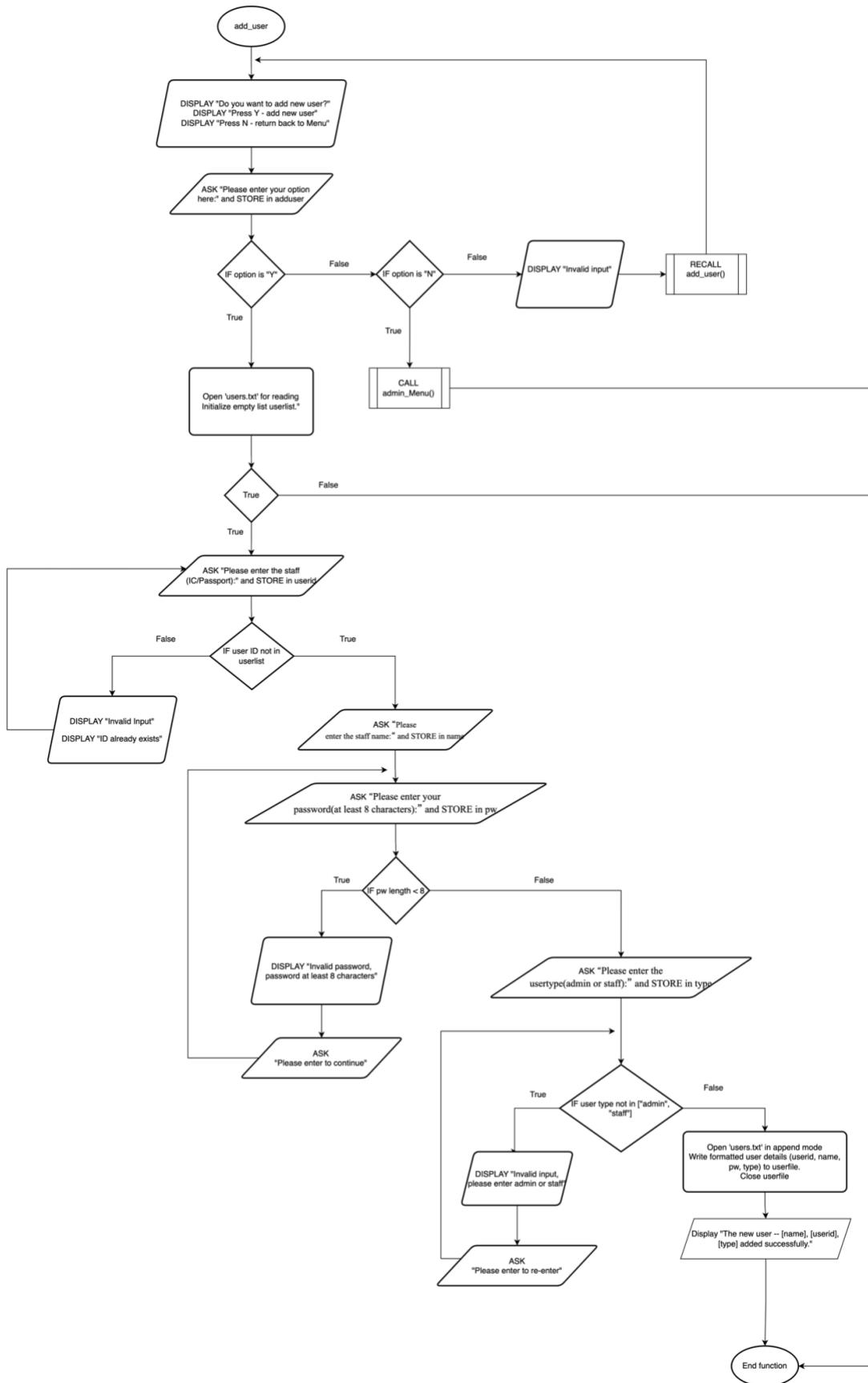


staff_menu()

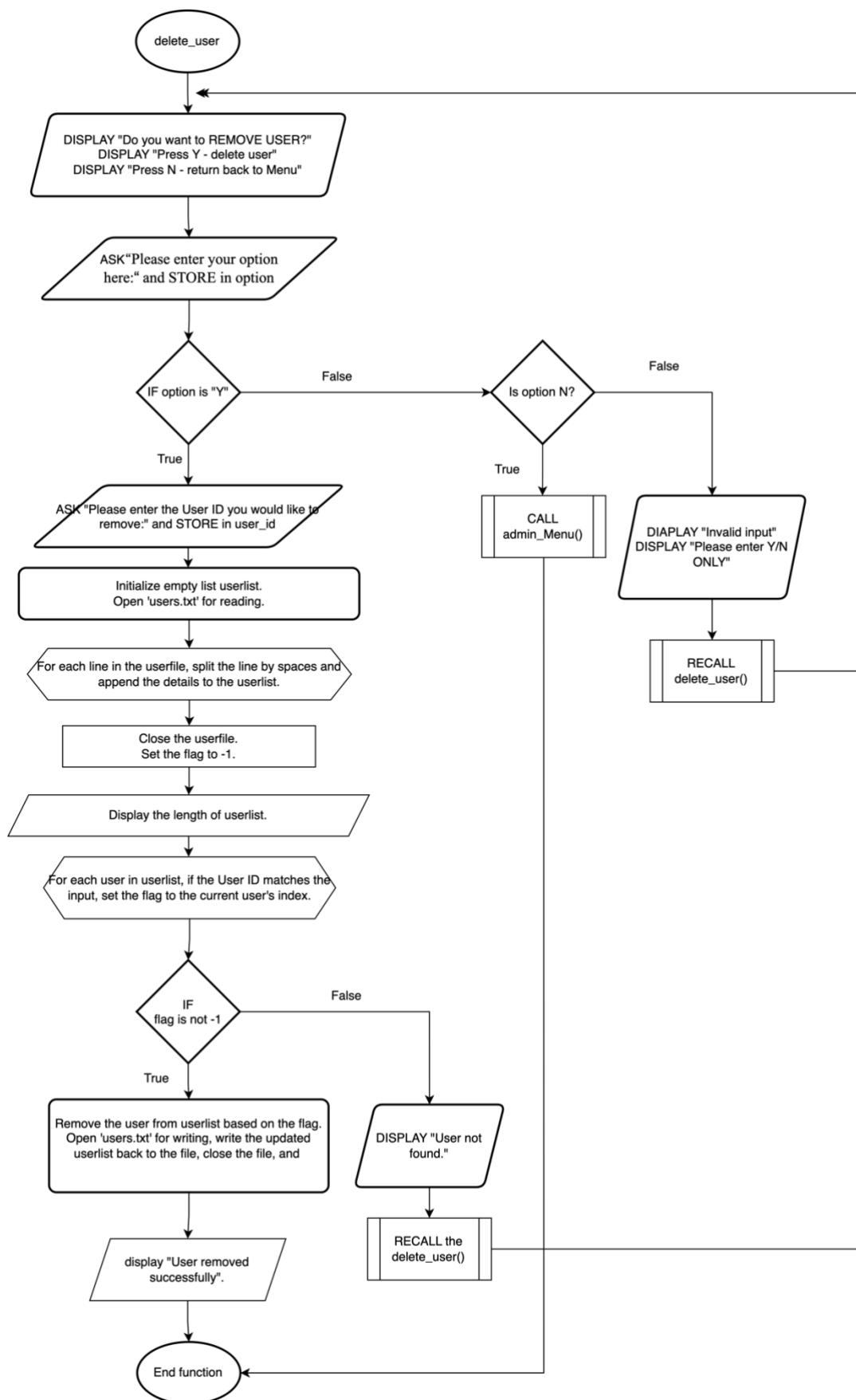




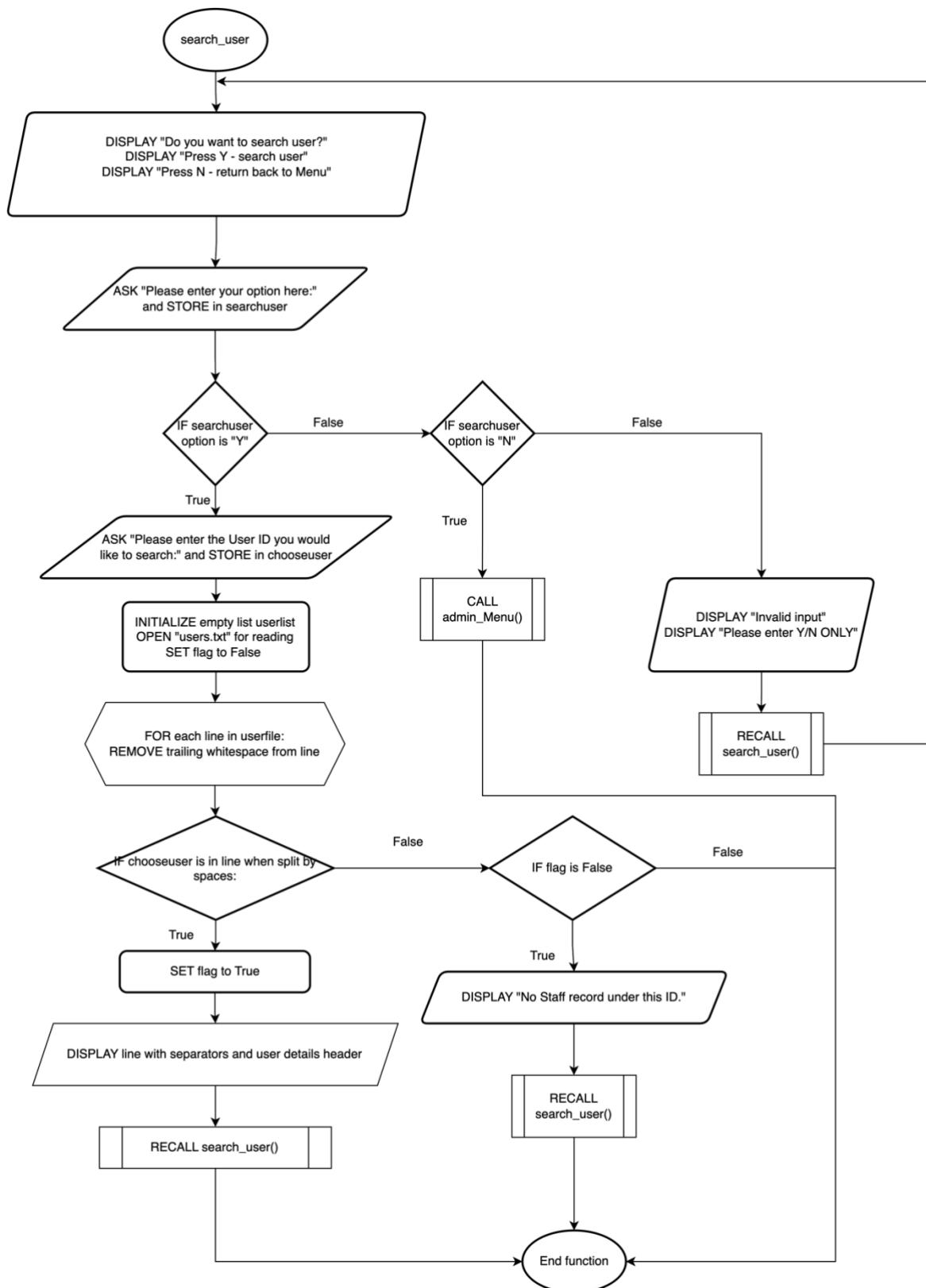
add_user()



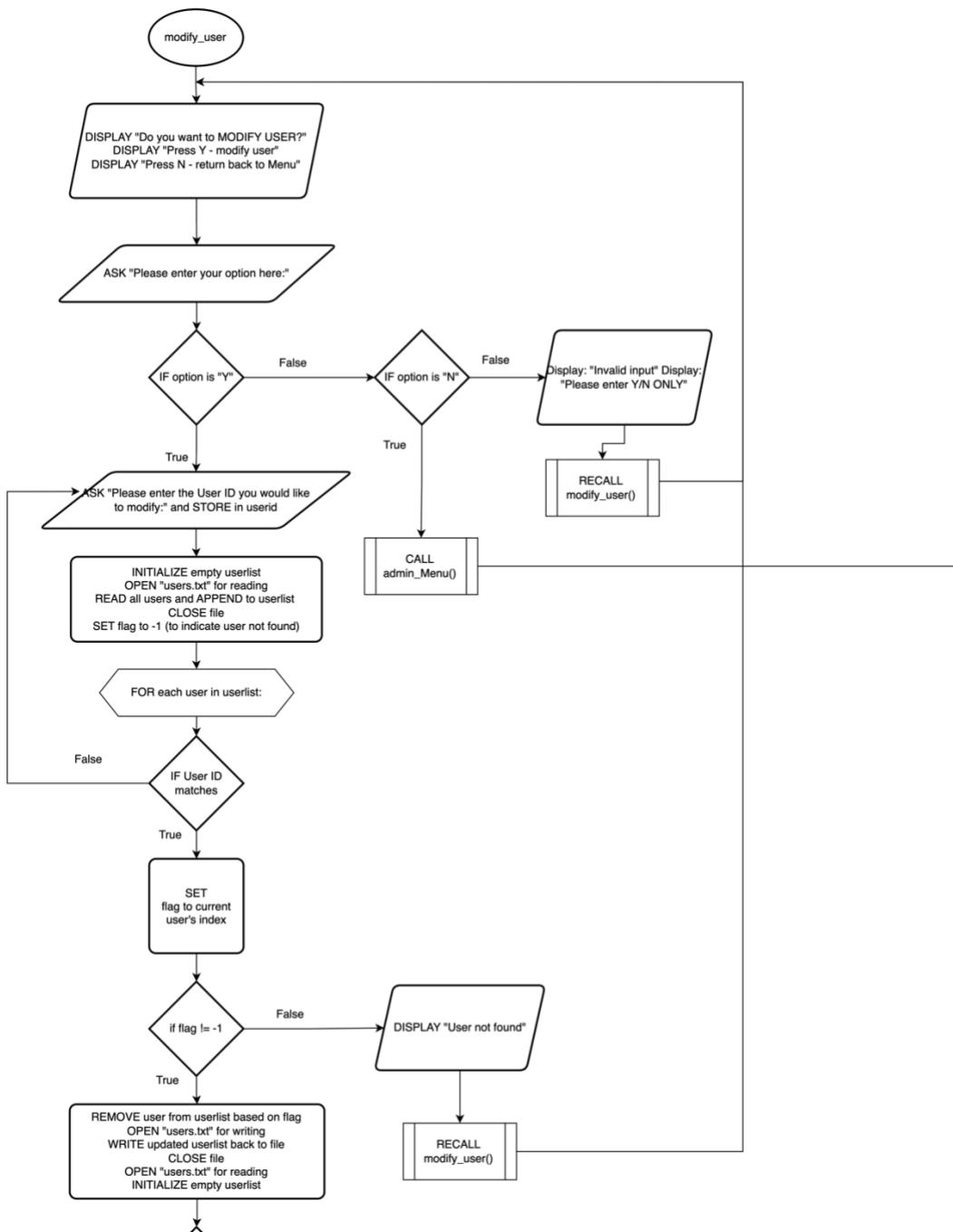
`delete_user()`

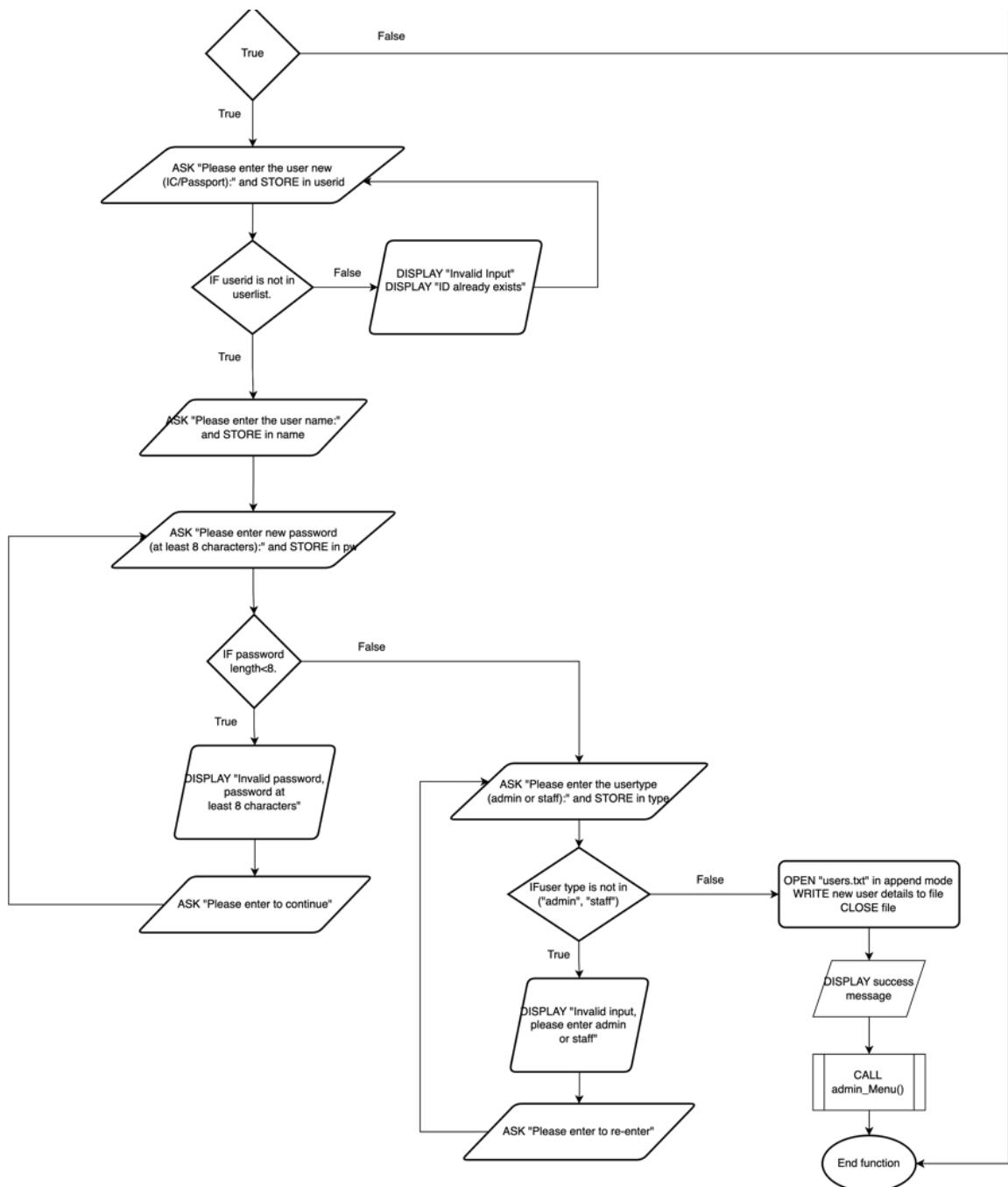


search_user()

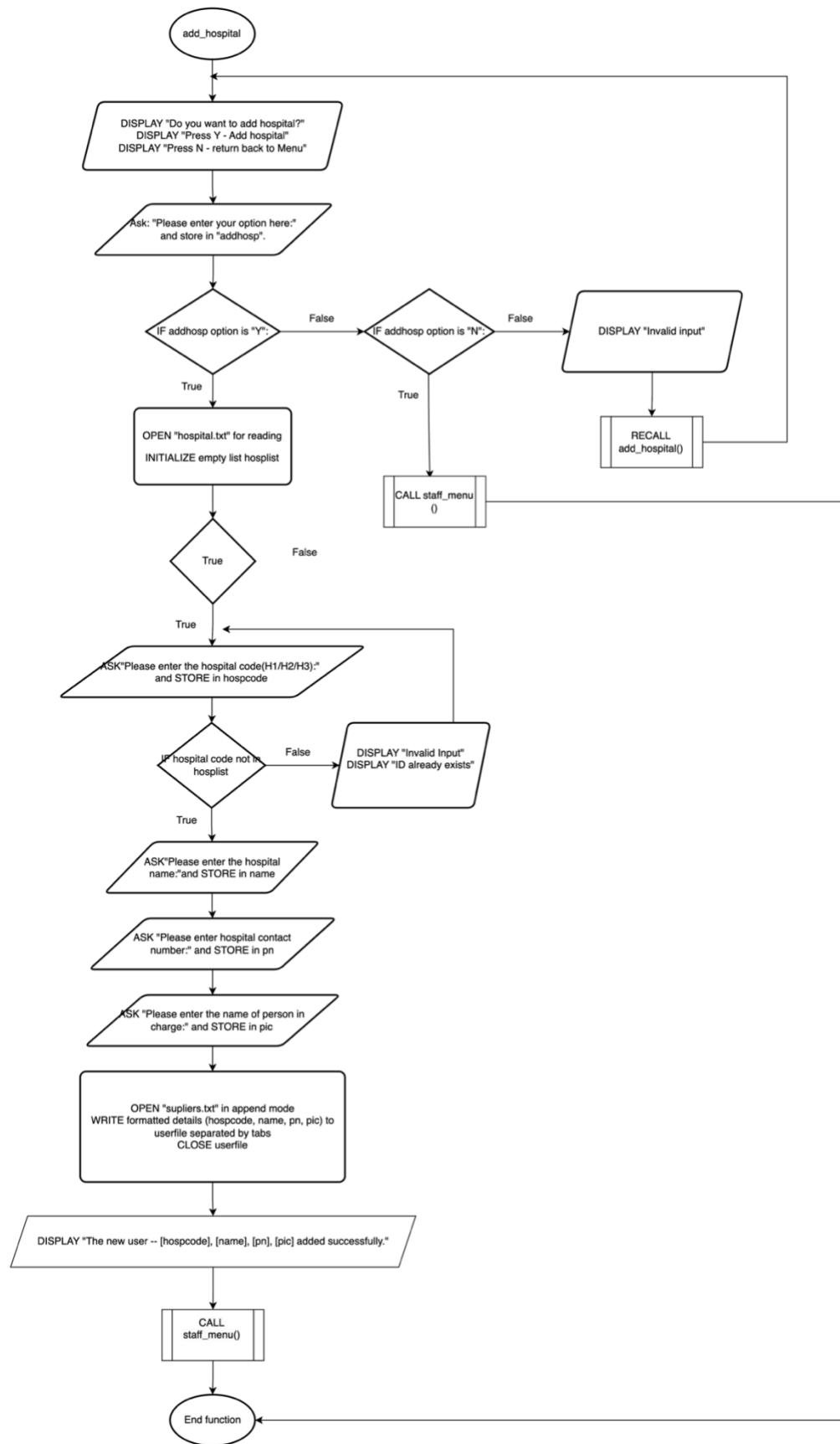


modify_user()

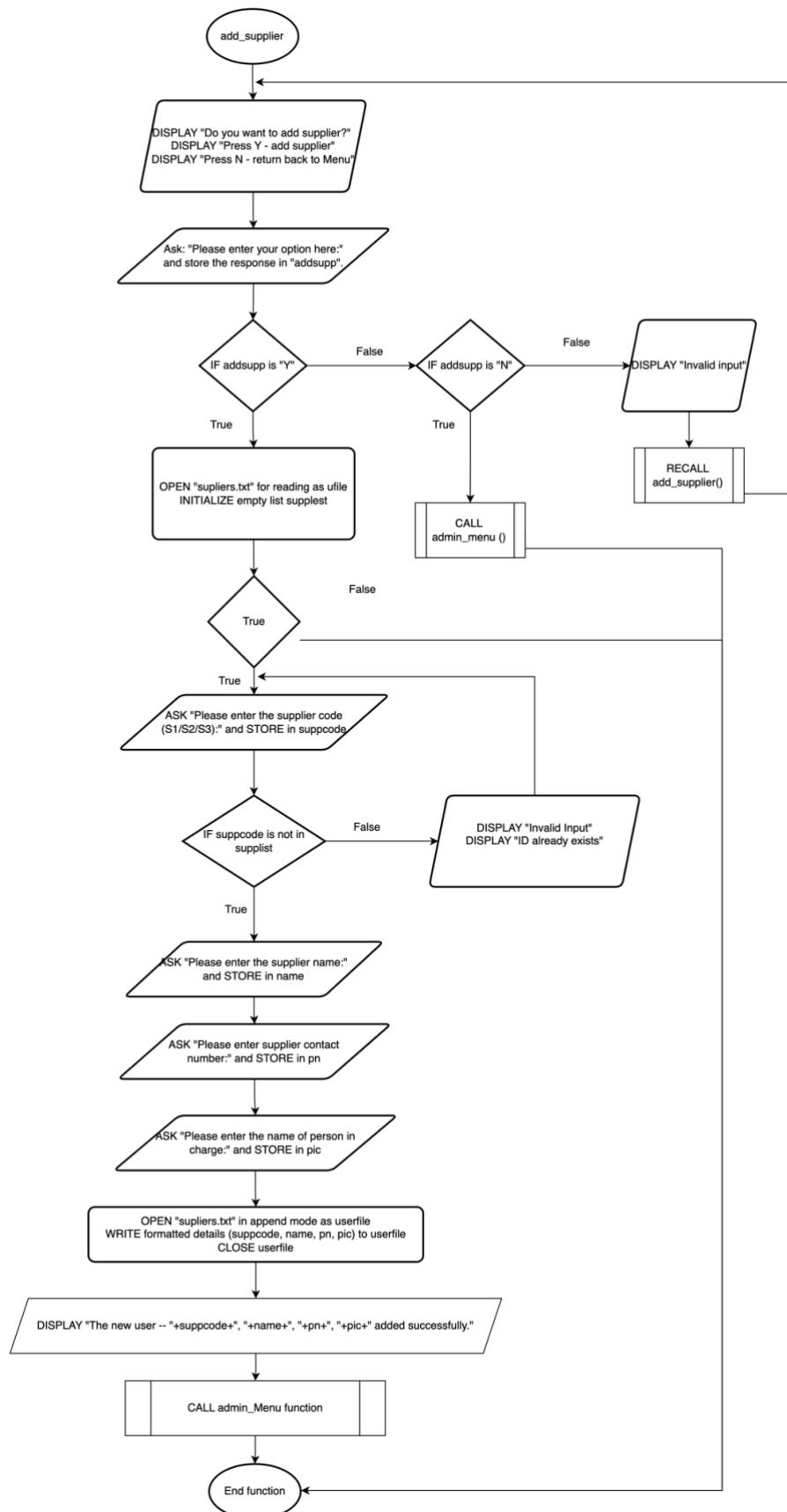




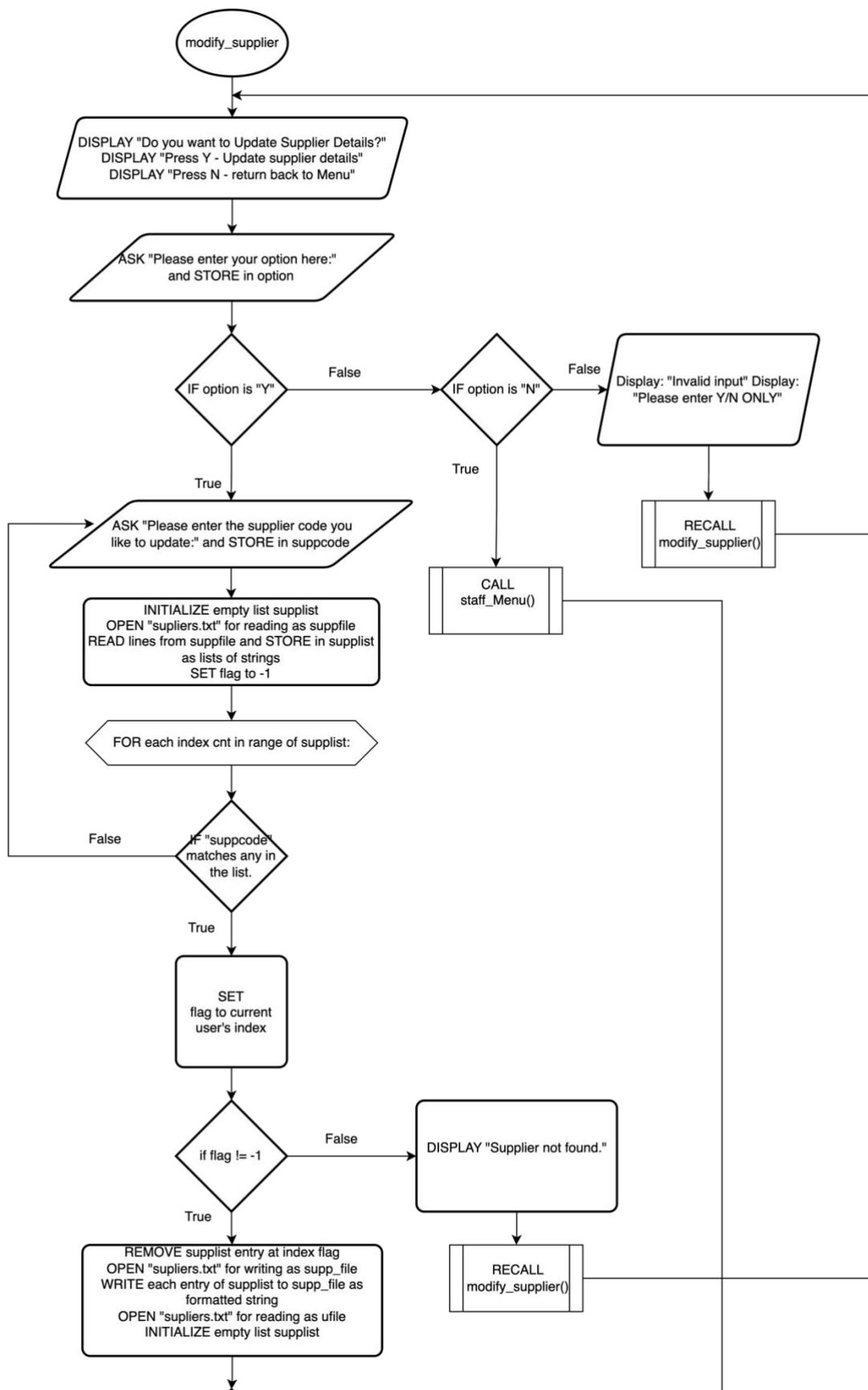
add_hospital()

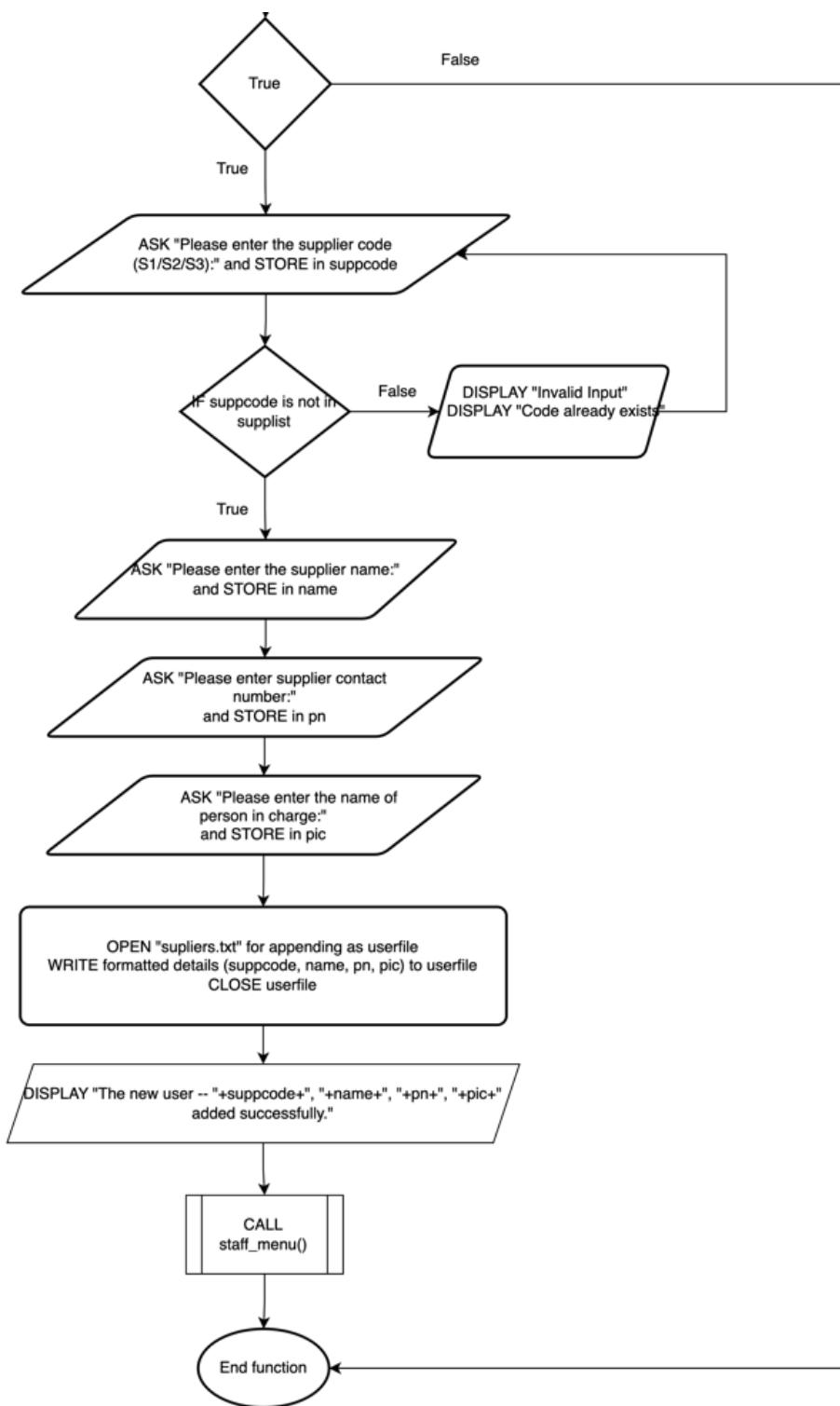


add_supplier()

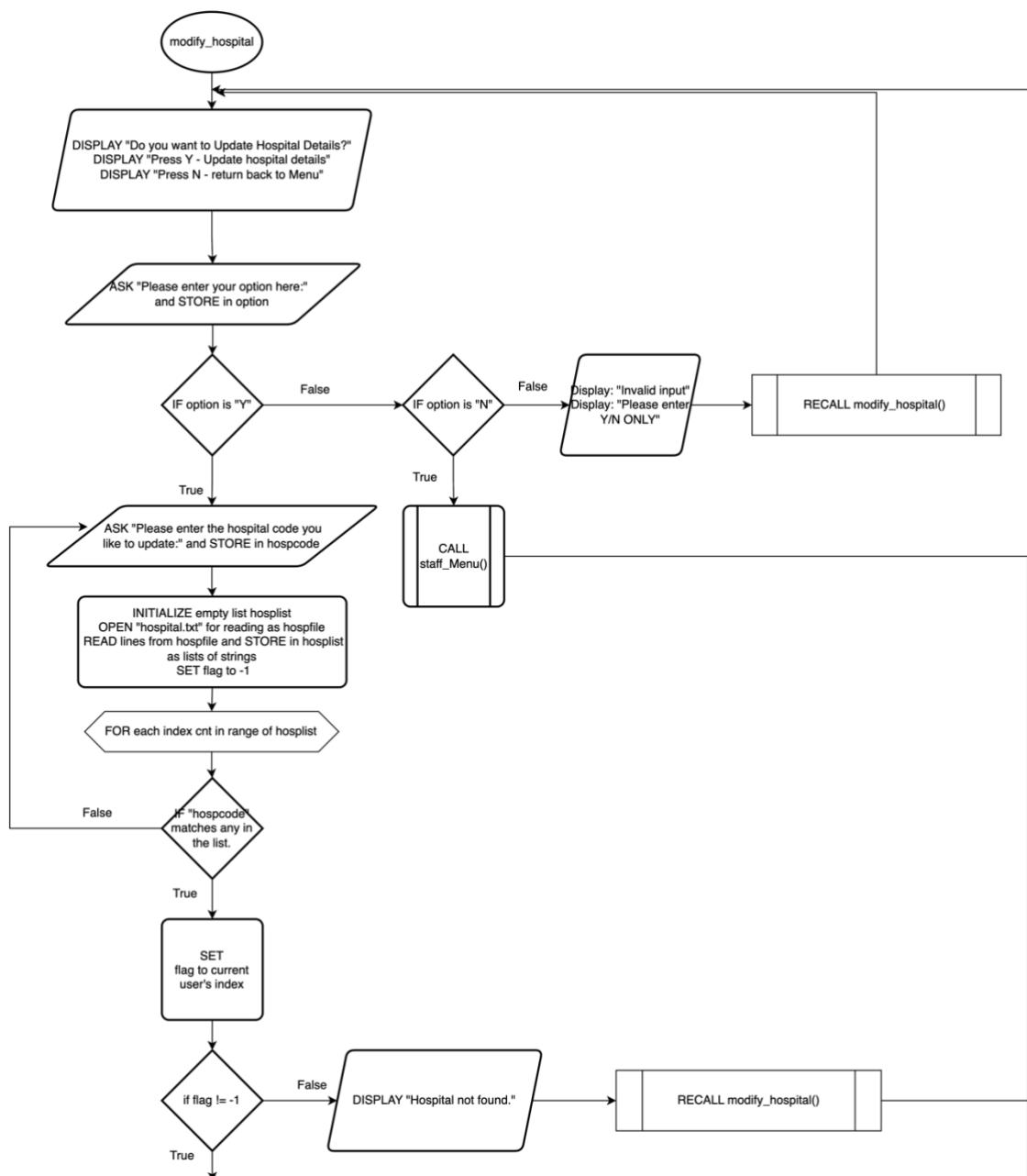


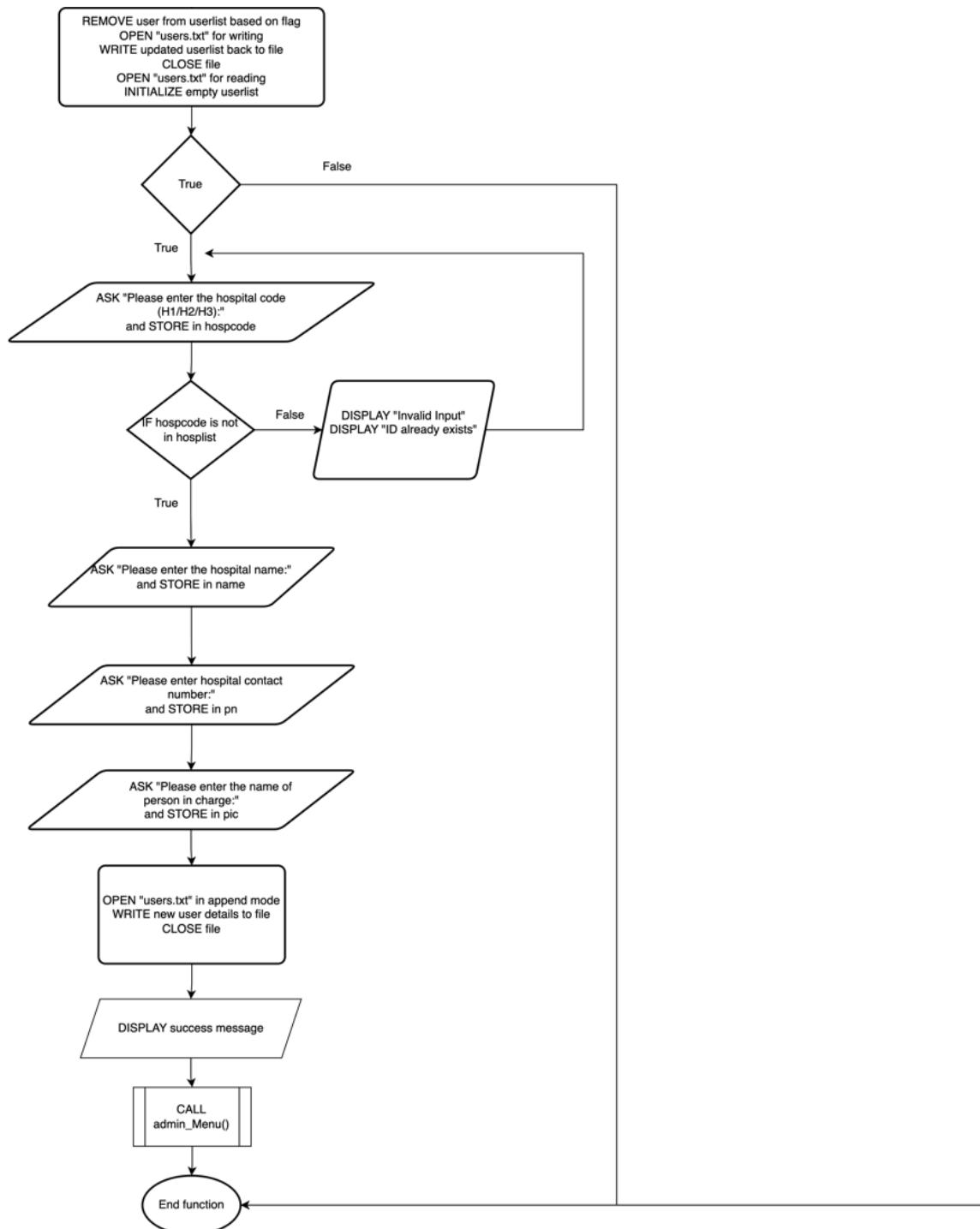
`modify_supplier()`



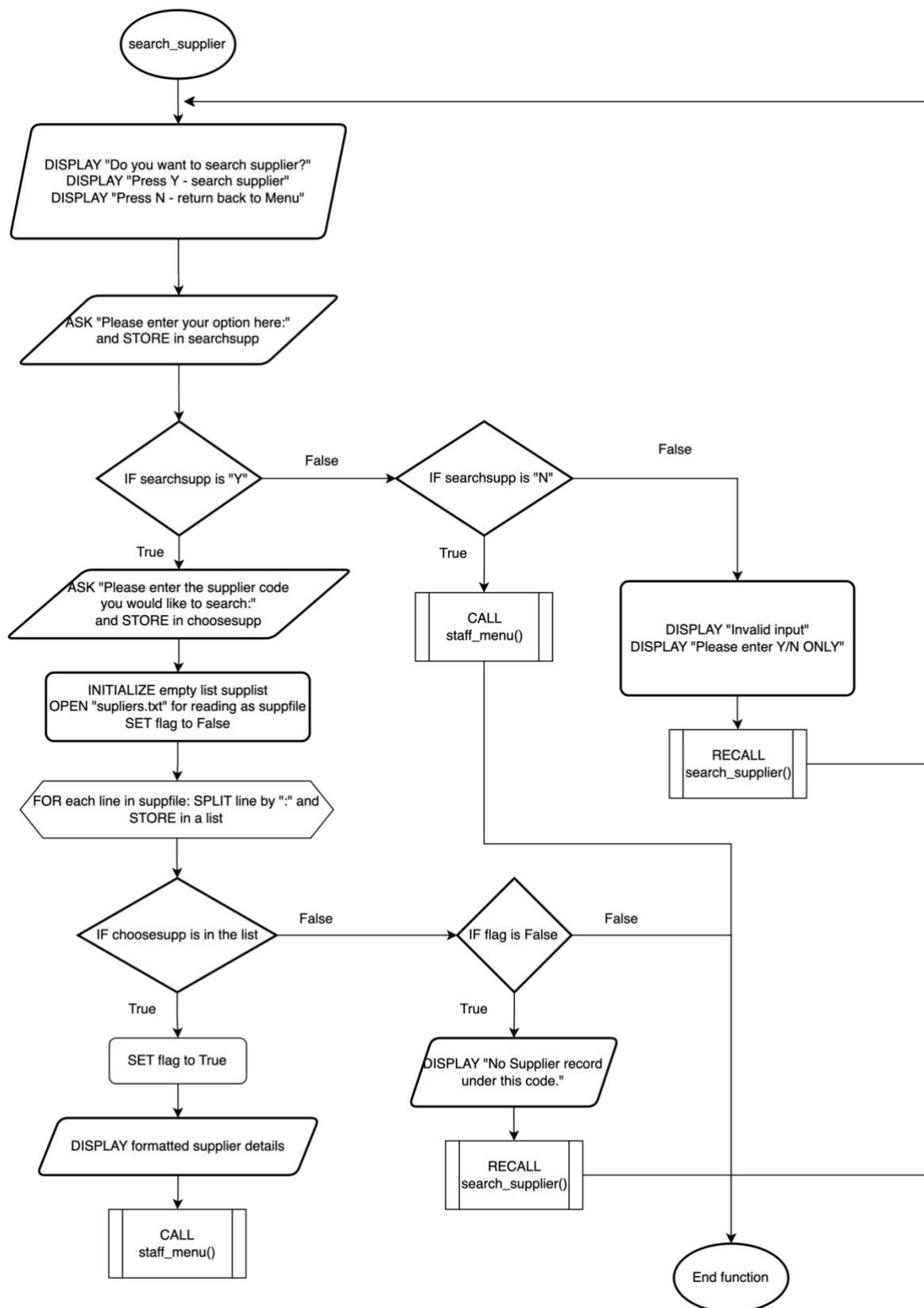


`modify_hospital()`

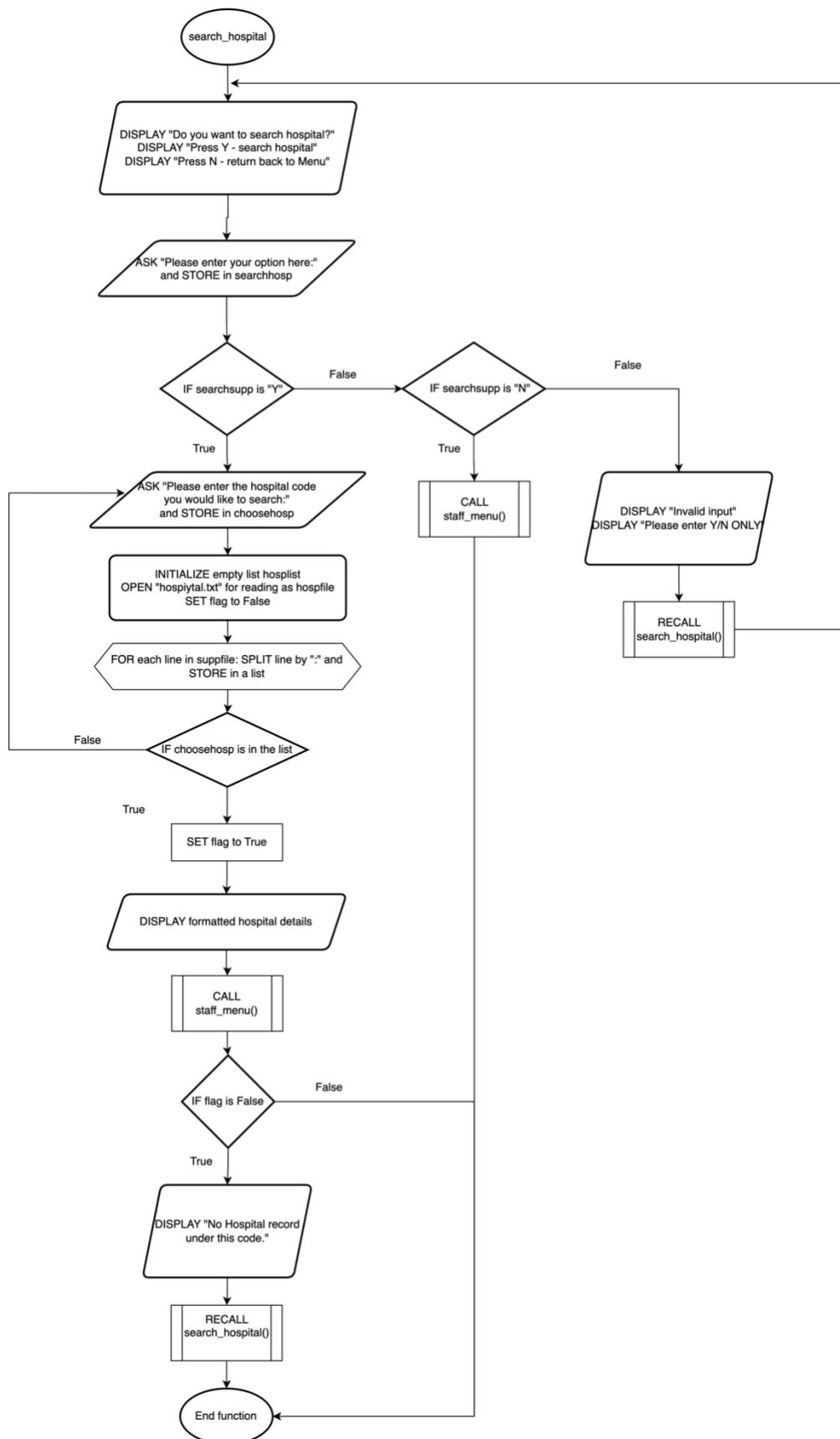




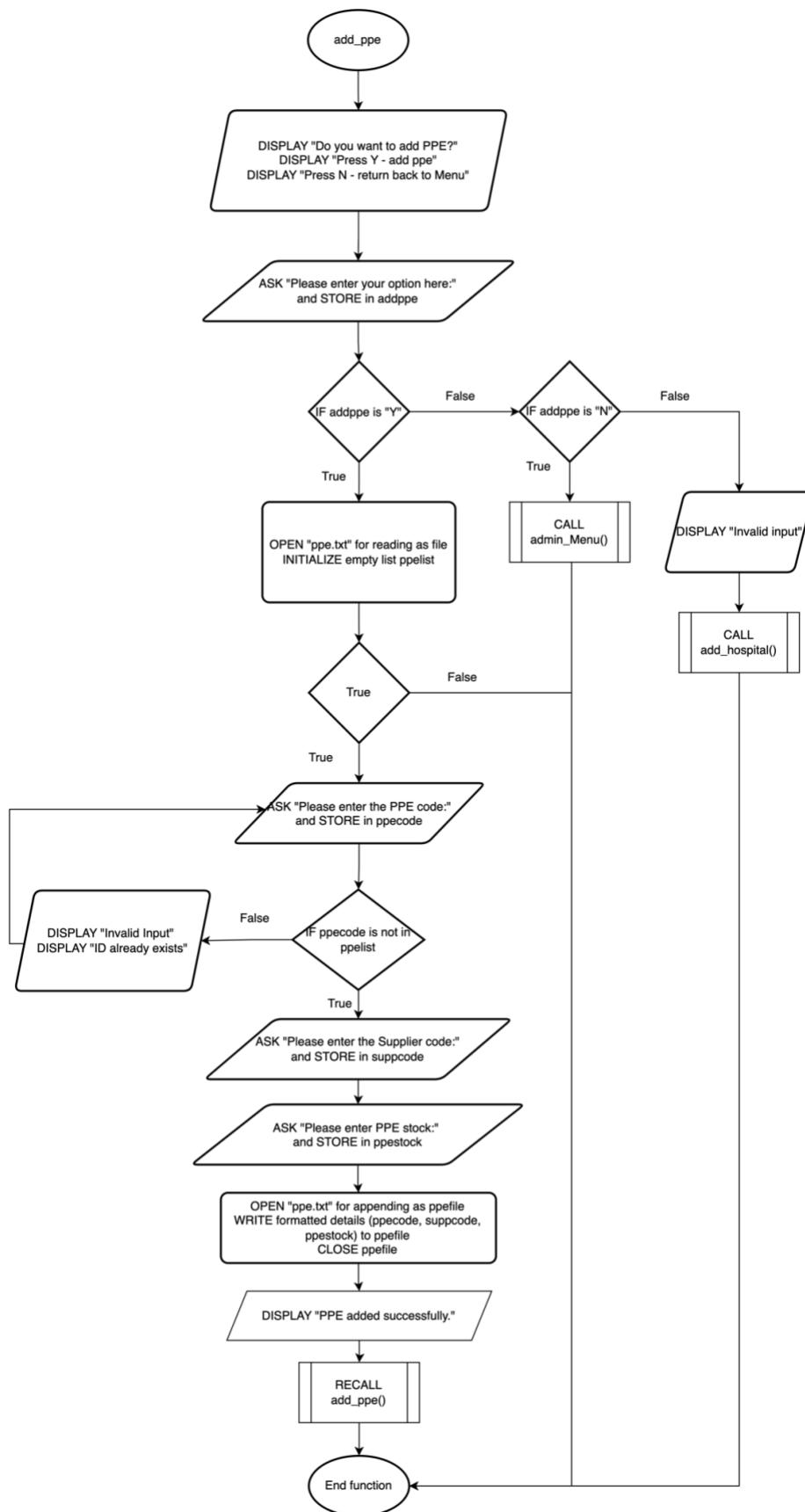
search_supplier()



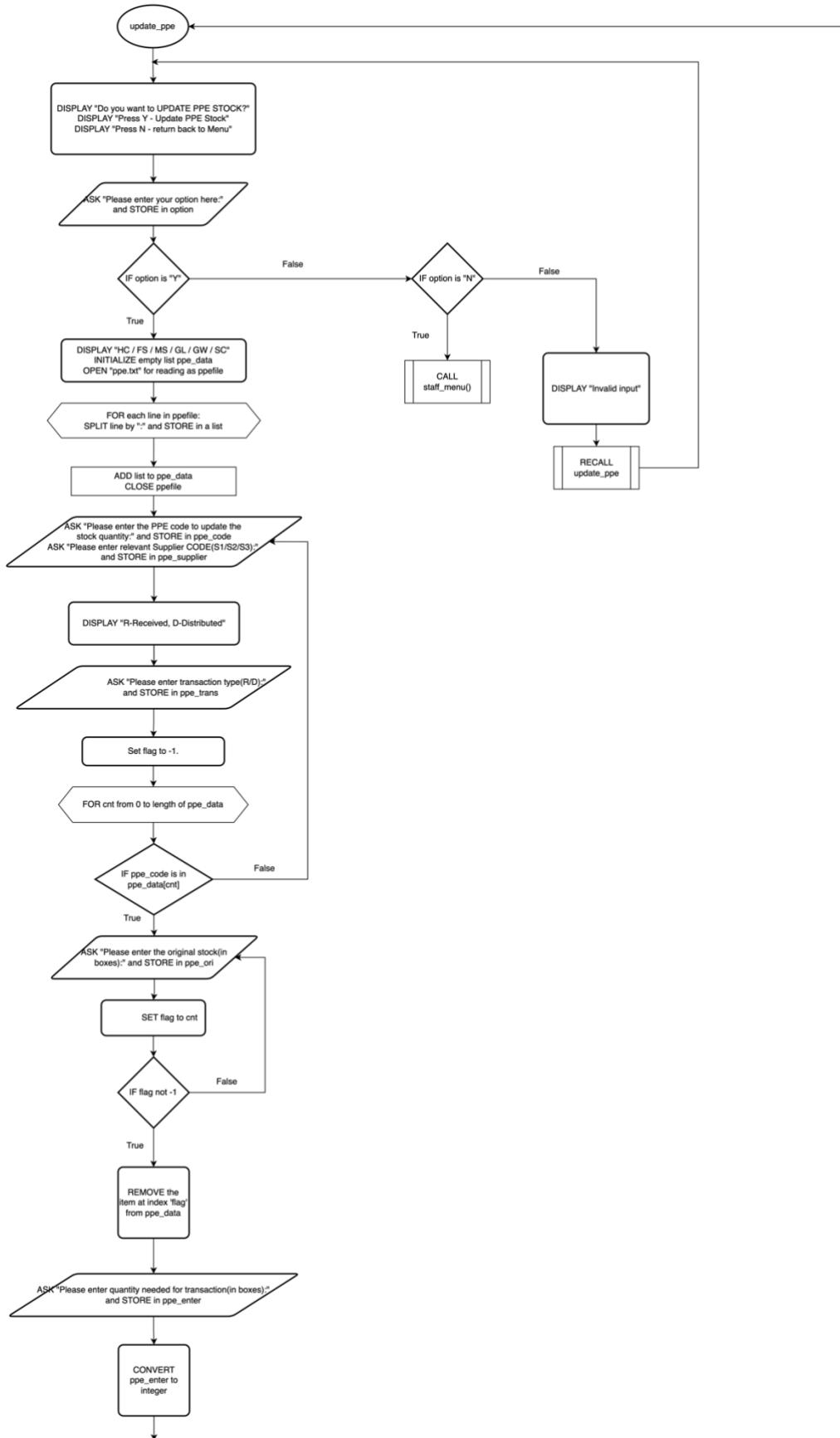
search_hospital()

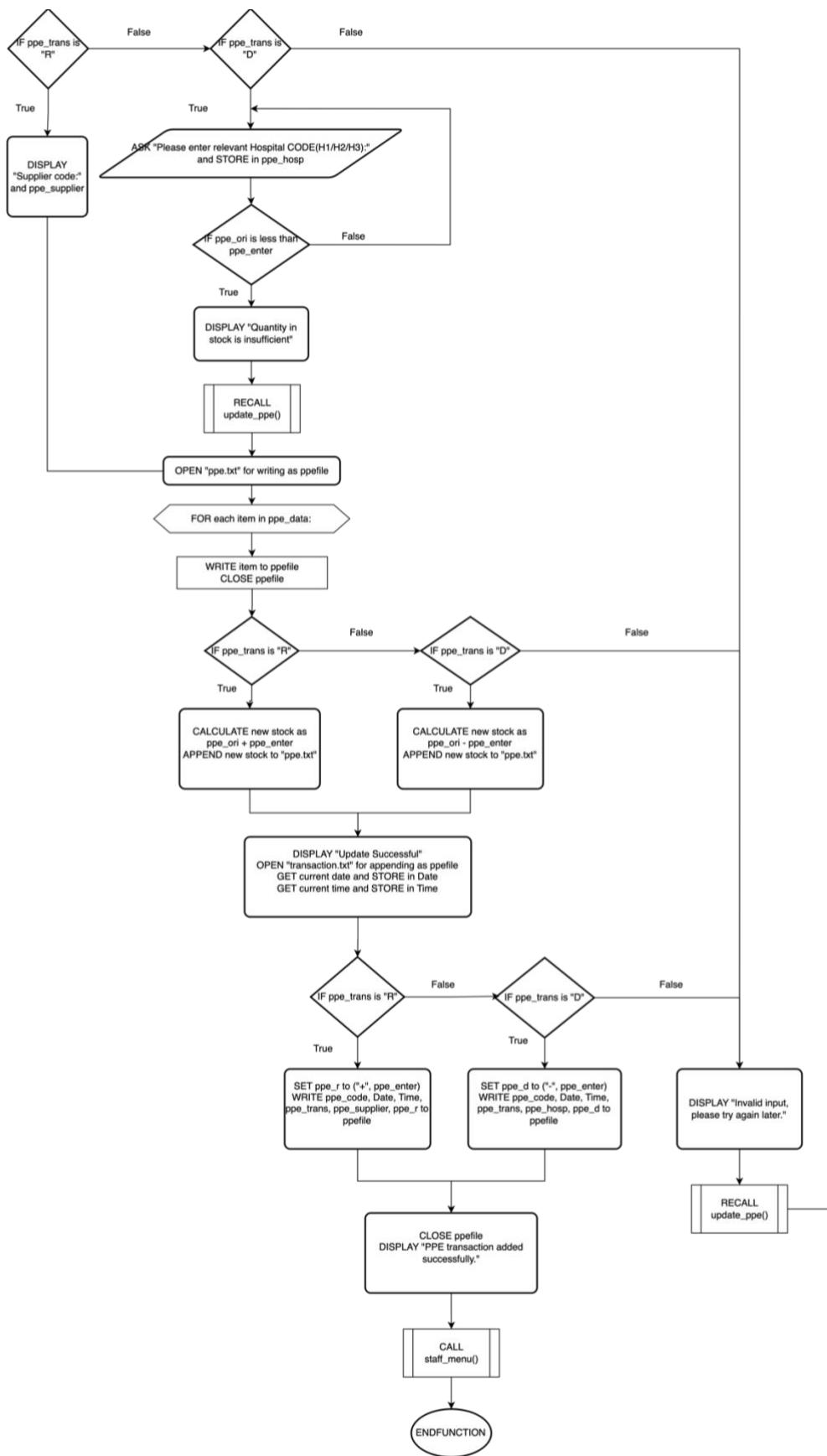


add_ppe()

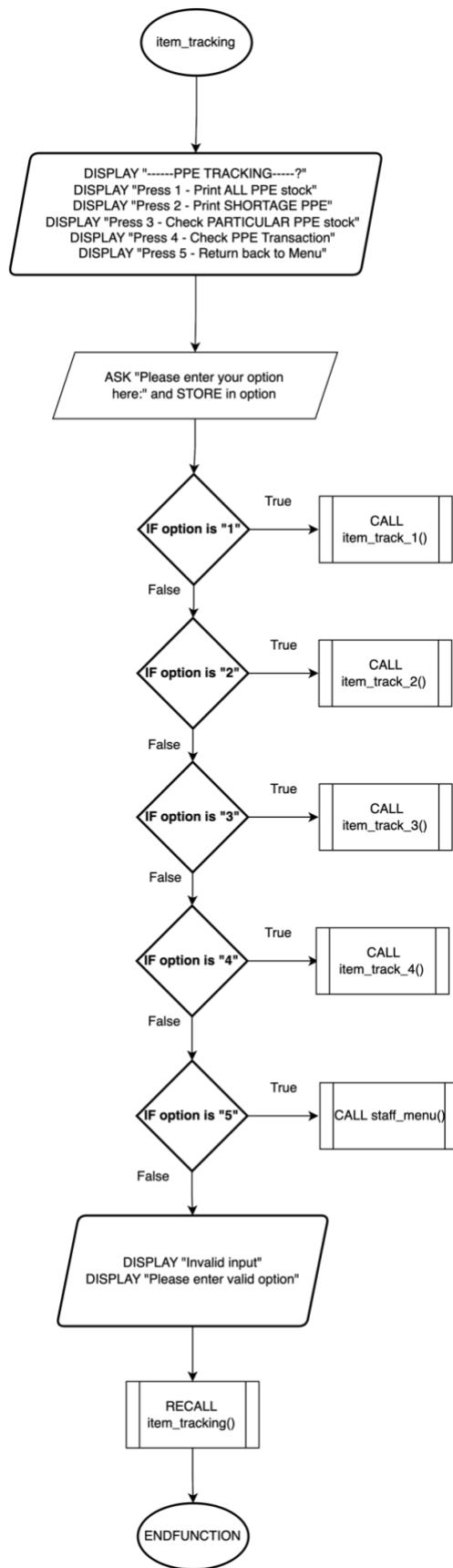


`update_ppe()`

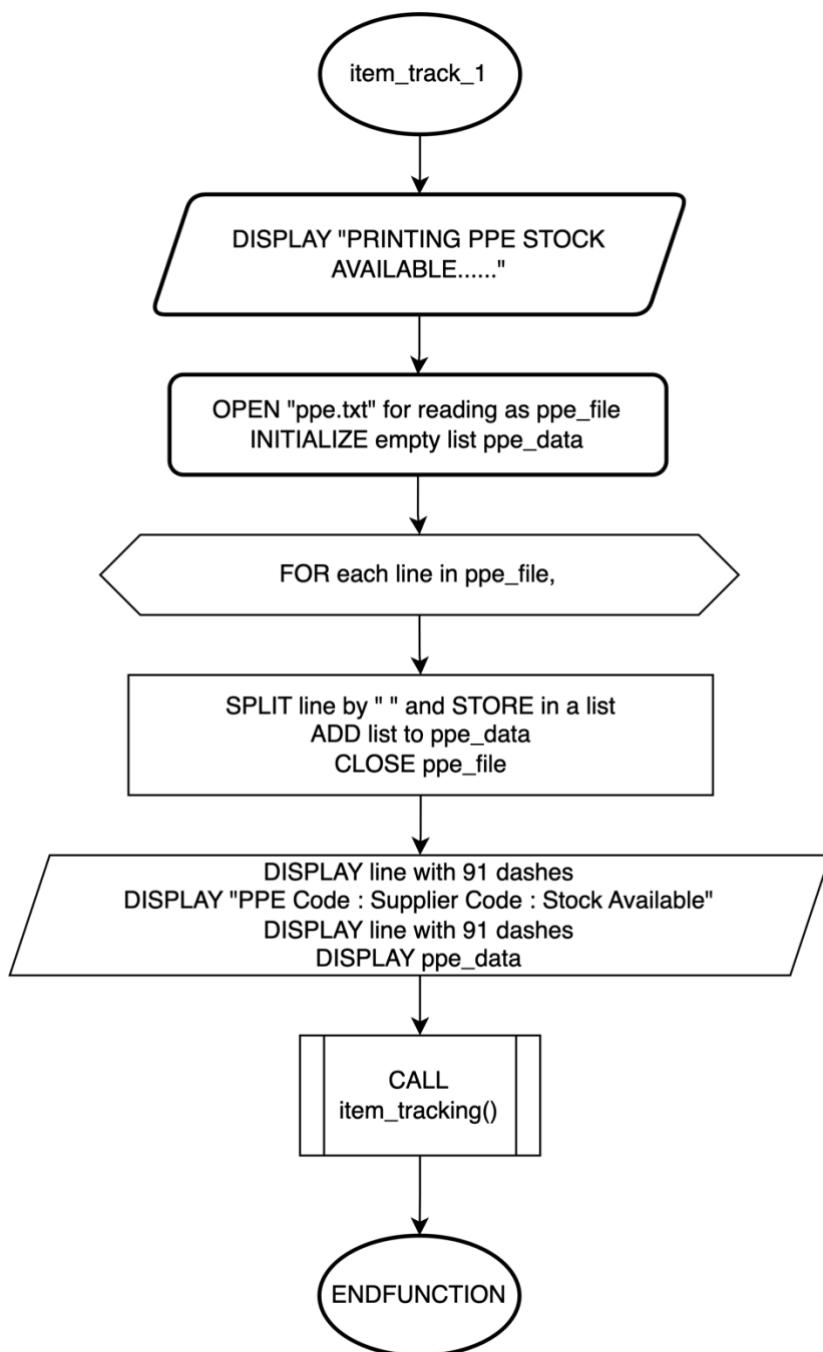




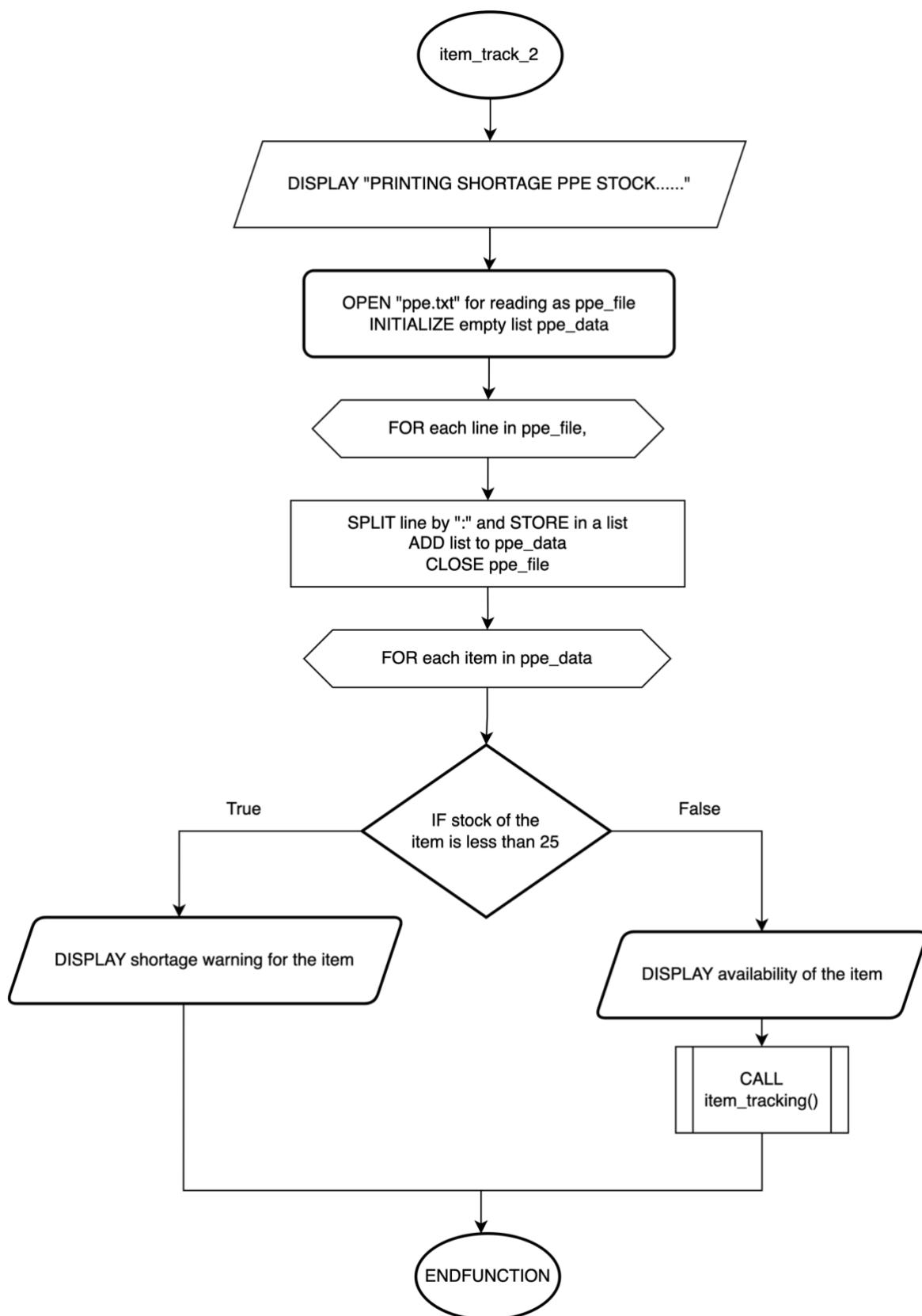
item_tracking()



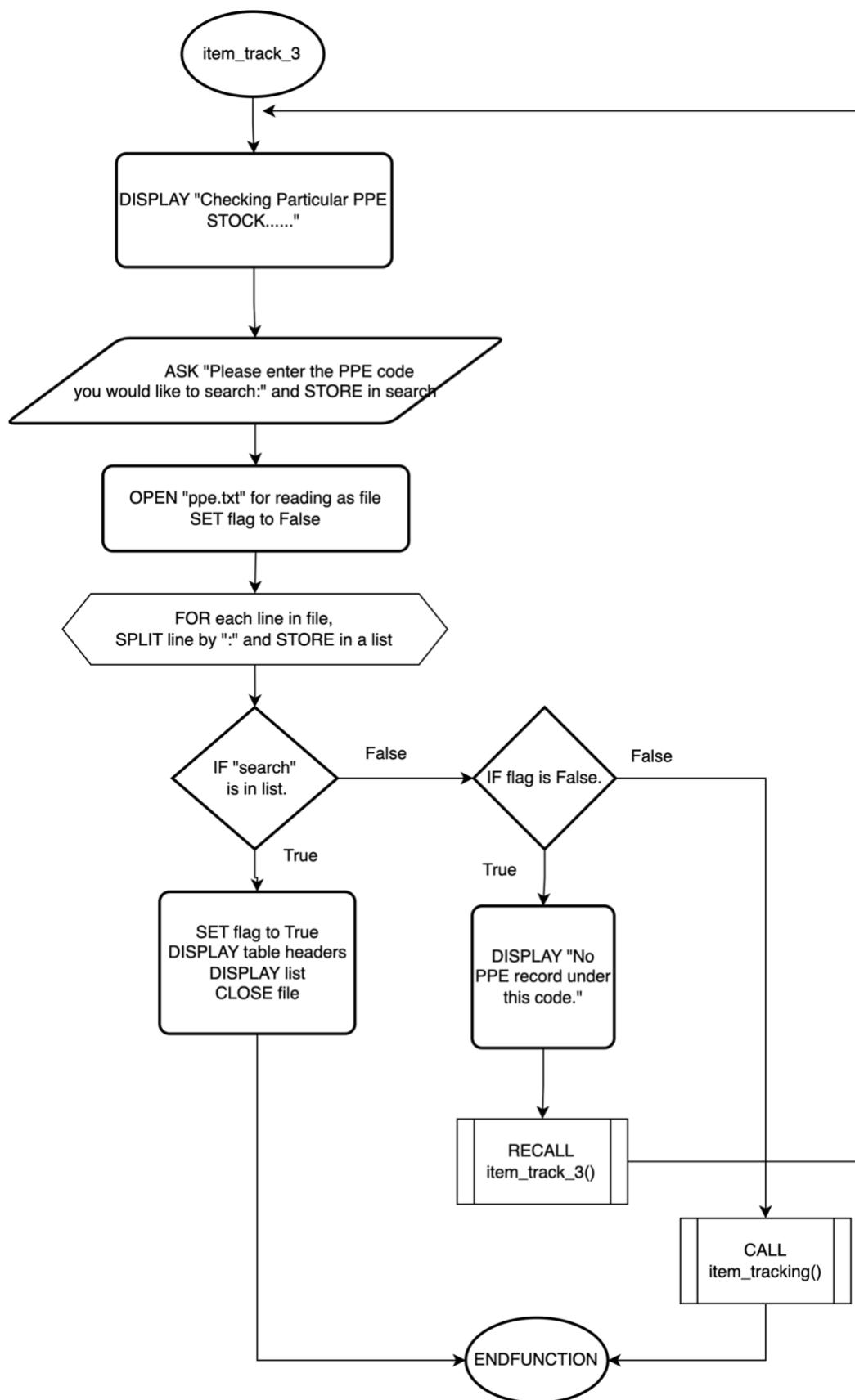
item_track_1()



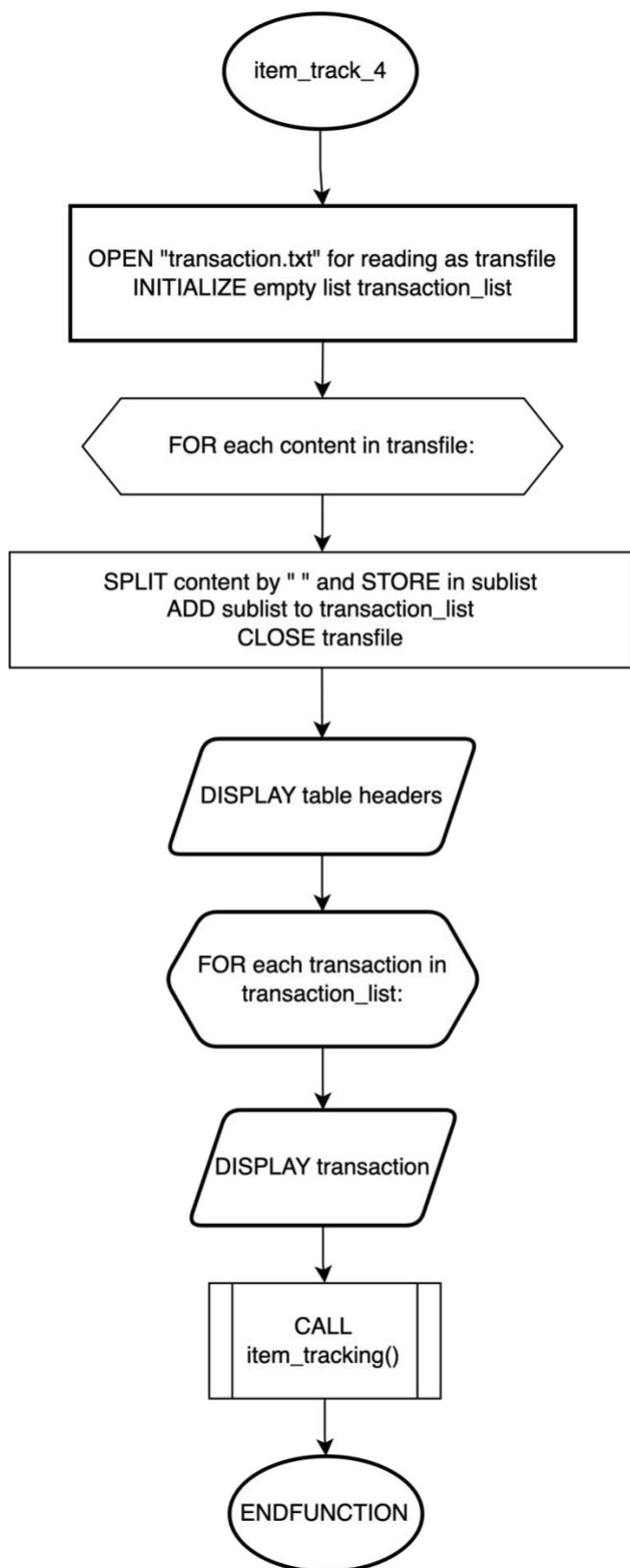
item_track_2()



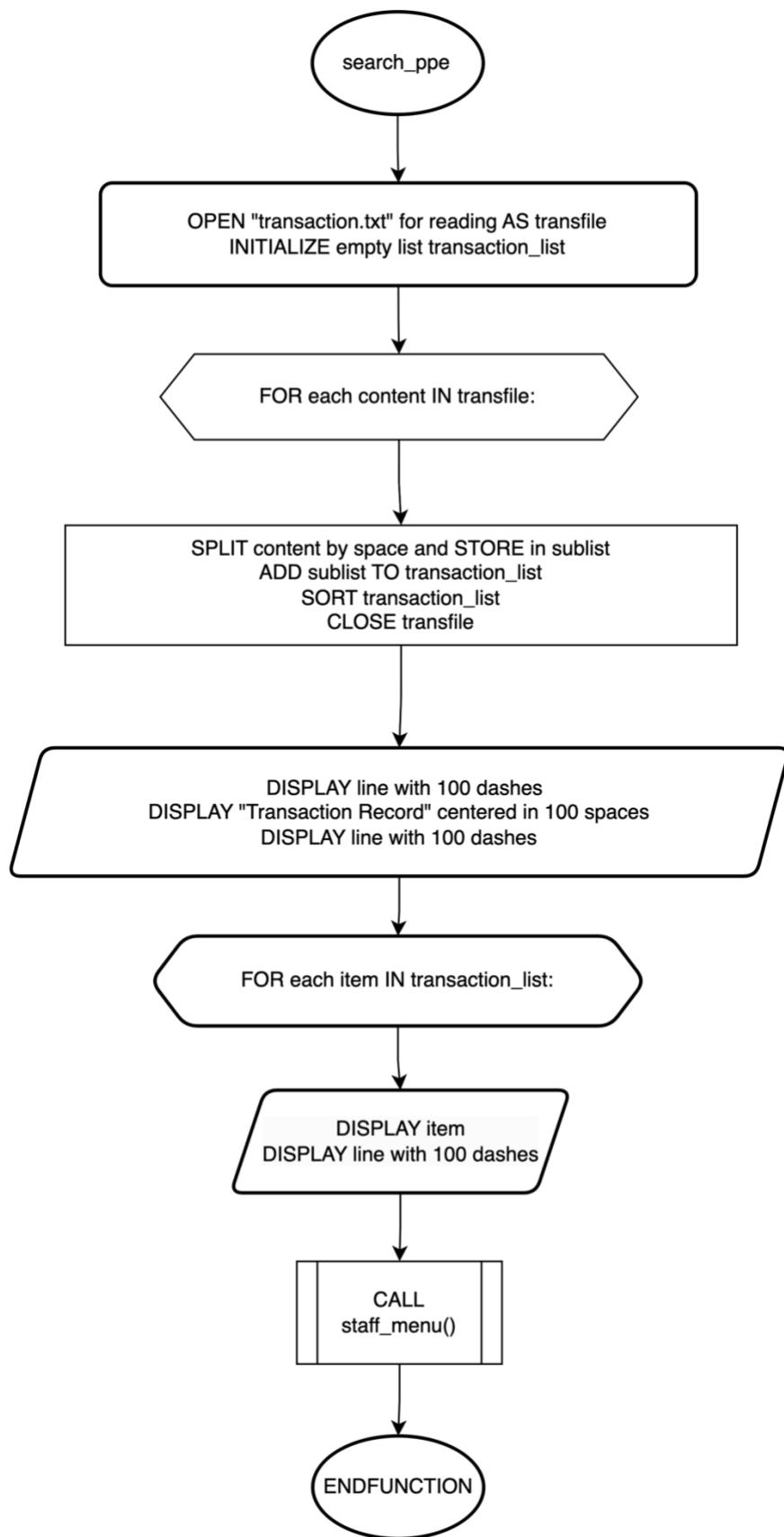
item_track_3()



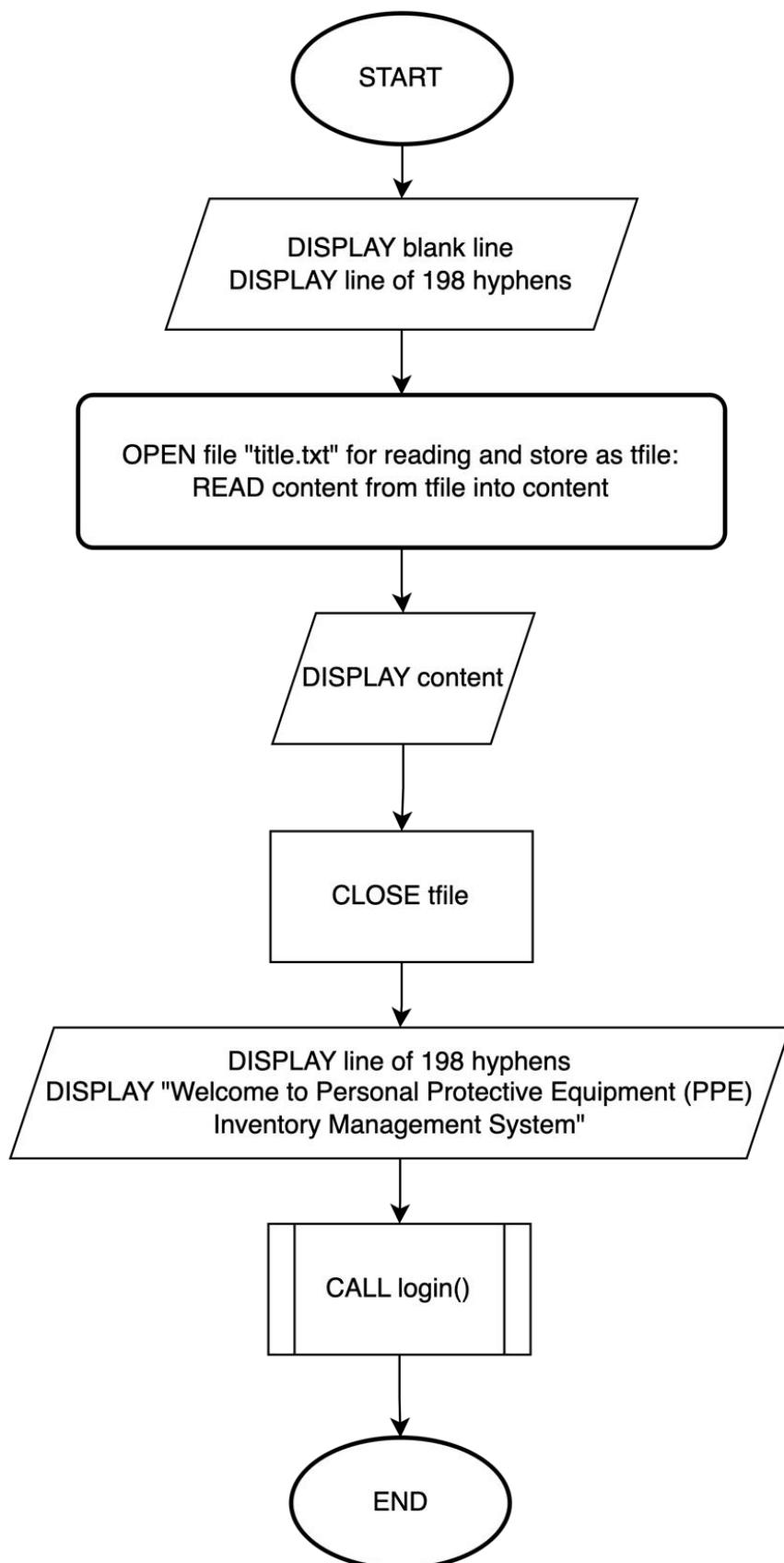
item_track_4



search_ppe()



MainLogic()



3.0 Program Source Code and Explanation

3.1 Receive Information

```
#----Login----  
def login():  
    while True:  
        print("\nWelcome to Login Page")  
        usertype = input("\nPlease enter your usertype(admin/staff):")  
        userpassword = input("\nPlease enter the password:")  
        if usertype.upper() == ("ADMIN"):  
            if userpassword == ("admin123"):  
                print("You are login to admin page")  
  
                admin_Menu()  
  
            else:  
                print("\nWrong password, please try again later...")  
  
        elif usertype.upper() == ("STAFF"):  
            if userpassword == ("staff789"):  
                print("You are login to staff page")  
                staff_menu()  
            else:  
                print("\nWrong password, please try again later...")  
                login()  
  
        else:  
            print("\nInvalid input, please try again...")  
            login()
```

Figure 4.1.1

This function receives information when the user inputs the user type. The system will follow the information input from the user and drive it to the respective menu.

3.2 Output Information

```
suppfile = open("suppliers.txt","r")
flag = False

for line in suppfile:
    line = line.rstrip()
    if choosesupp in line.split(" "):
        flag = True
        print("=" * 100)
        print("Supplier code" + " : " + "Name" + " : " + "Phone Number" + " : " + "Person In Charge")
        print("=" * 100)
        print(line)
        print('\n')
        staff_menu()
```

Figure 4.2.1

```
Please enter the supplier code you would like to search:S1
=====
Suplier code : Name : Phone Number : Person In Charge
=====
S1 : Supplier JB : 0159874632 : Mr. Jayden
```

Figure 4.2.2

This function outputs information when the system uses the `print` or similar functions. The system will display information based on the user's input and provide results.

3.3 Performing Arithmetic

```
with open("ppe.txt", "w") as ppefile:
    for cnt in ppe_data:
        ppefile.write(":".join(cnt)+"\n")

    if ppe_trans == ("R"):
        ppe_rstock = (int(ppe_ori) + int(ppe_enter))
        ppefile.write("{0}:{1}:{2}".format(ppe_code, ppe_supplier, ppe_rstock))

    elif ppe_trans == ("D"):
        ppe_dstock = (int(ppe_ori) - int(ppe_enter))
        ppefile.write("{0}:{1}:{2}".format(ppe_code, ppe_supplier, ppe_dstock))

    print("\nUpdate Successful")
```

Figure 4.3.1

```
PPE Code : Supplier Code : Stock Available
[['FS:S3:180'], ['GL:S2:15'], ['GW:S1:100'], ['HC:S1:20'], ['MS:S2:170'], ['SC:S3:150']]
```

Figure 4.3.2

This function conducts calculations when the user uses numerical inputs. The system will process the input data, perform the arithmetic operations, and provide results.

3.4 Assigning Value to A Piece of Data

```
for i in range(0, len(ppe_data)):
    if int(ppe_data[i][2]) < 25:
        print("The Stock of", str(ppe_data[i][0]), "is only left", str(ppe_data[i][2]), "boxes and it's less than 25 boxes.")
        print("Please increase the stock available.")
    else:
        print(str(ppe_data[i][0]), "are available.")
```

Figure 4.4.1

```
GW are available.
SC are available.
The Stock of HC is only left 20 boxes and it's less than 25 boxes.
Please increase the stock available.
FS are available.
MS are available.
The Stock of GL is only left 15 boxes and it's less than 25 boxes.
Please increase the stock available.
```

Figure 4.4.2

This function assigns a specific value when the user uses an input for a data variable. The system processes the input data and allocates it to the designated data variable.

3.5 Repeat A Group of Actions

```
while True:  
    hospcode=input("\nPlease enter the hospital code(H1/H2/H3):")  
    if(hospcode not in hosplist):  
        break  
    else:  
        print("\nInvalid Input")  
        print("\nID already exists")  
  
while True:  
    name=input("\nPlease enter the hospital name:")  
    break  
  
while True:  
    pn=input("\nPlease enter hospital contact number:")  
    break  
  
while True:  
    pic=input("\nPlease enter the name of person in charge:")  
    break
```

Figure 4.5.1

This function initiates a loop when the user inputs or meets specific conditions. The system will repeatedly execute a set of tasks until the conditions are met or the user exits.

4.0 Screenshots of Sample Input/Output and Explanation

4.1 Login Page



Figure 4.1.1 Login to Admin



Figure 4.1.2 Login to Staff

Display a login page of the system. Users need to enter the correct user type and password, to be granted access to respective interfaces.

4.2 Menu Page

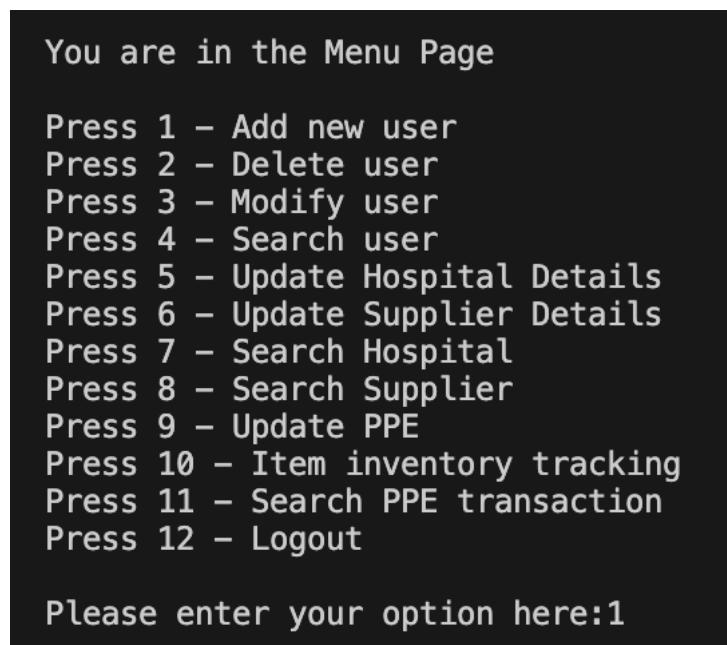


Figure 4.2.1 Admin Menu

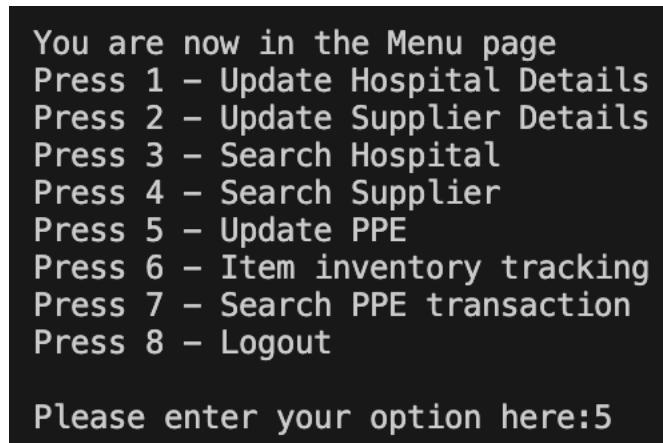


Figure 4.2.2 Staff Menu

Display a menu with various administrative options for users. When the user presses a specified number, the corresponding function will be executed.

4.3 Add/Delete/Modify/Search User Page

```
Do you want to add new user?  
Press Y – add new user  
Press N – return back to Menu  
  
Please enter your option here:y  
  
Please enter the staff (IC/Passport):059875063598  
  
Please enter the staff name:riki  
  
Please enter your password(at least 8 characters):plokjnhb  
  
Please enter the usertype(admin or staff):staff  
  
The new user -- riki, 059875063598, staff added successfully.
```

Figure 4.3.1 Add User

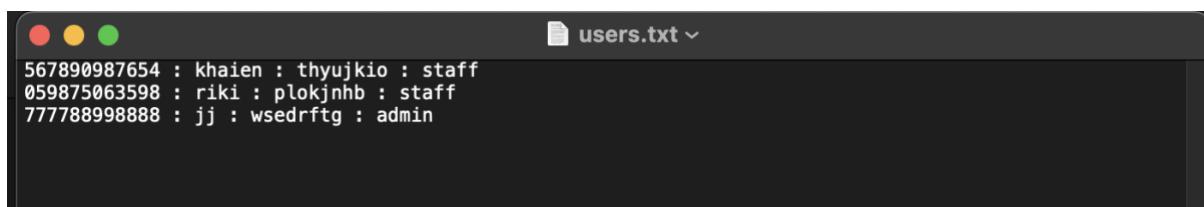


Figure 4.3.2 User Text File

```
Do you want to REMOVE USER?  
Press Y – delete user  
Press N – return back to Menu  
  
Please enter your option here:y  
  
Please enter the User ID you would like to remove:333333445555  
4  
  
User: 333333445555 removed successfully.
```

Figure 4.3.3 Delete User

```
Do you want to MODIFY USER?  
Press Y - modify user  
Press N - return back to Menu  
  
Please enter your option here:y  
  
Please enter the User ID you would like to modify:098908233456  
  
Please enter the user new (IC/Passport):777788998888  
  
Please enter the user name:jj  
  
Please enter new password(at least 8 characters):wsedrfhg  
  
Please enter the usertype(admin or staff):admin  
The new user -- jj, 777788998888, admin modify successfully.
```

Figure 4.3.4 Modify User

```
Do you want to search user?  
Press Y - search user  
Press N - return back to Menu  
  
Please enter your option here:  
  
Please enter the User ID you would like to search:567890987654  
  
User ID : Username : Password : User Type  
567890987654 : khaien : thyujkio : staff
```

Figure 4.3.5 Search User

Display user management options for admin. By selecting a specific option, the admin can add a new user, remove a user, modify a user, and search user.

4.4 Update/Search Hospital Page

```

Do you want to Update Hospital Details?
Press Y - Update hospital details
Press N - return back to Menu

Please enter your option here:y

Please enter the hospital code you like to update:h3

Please enter the hospital code(H1/H2/H3):h1

Please enter the hospital name:Hospital MY

Please enter hospital contact number:0152369874

Please enter the name of person in charge:Mr. Jayden
The new user -- H1, Hospital MY, 0152369874, Mr. Jayden added successfully.

```

Figure 4.4.1 Update Hospital details

```

Do you want to search hospital?
Press Y - search hospital
Press N - return back to Menu

Please enter your option here:y

Please enter the hospital code you would like to search:H2
=====
Hospital code : Name : Phone Number : Person In Charge
=====
H2 : Hospital KL : 0197856789 : Ms.Violet
=====
```

Figure 4.4.2 Search Hospital

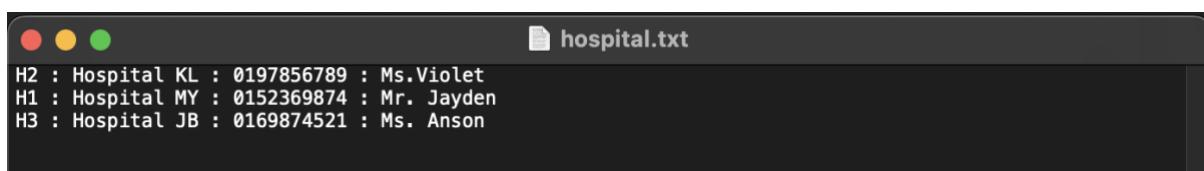


Figure 4.4.3 Hospital Text File

Display hospital management options for admin and staff. By selecting a specific option, admin and staff can add updated hospital details and search hospitals.

4.5 Update/Search Supplier Page

```

Do you want to Update Supplier Details?
Press Y - Update supplier details
Press N - return back to Menu

Please enter your option here:y

Please enter the supplier code you like to update:S1

Please enter the supplier code(S1/S2/S3):S1

Please enter the supplier name:Supplier KL

Please enter supplier contact number:0157896412

Please enter the name of person in charge:Mr. Usman
The new user -- S1, Supplier KL, 0157896412, Mr. Usman added successfully.

```

Figure 4.5.1 Update Supplier details

```

Do you want to search supplier?
Press Y - search supplier
Press N - return back to Menu

Please enter your option here:y

Please enter the supplier code you would like to search:S1
=====
Suplier code : Name : Phone Number : Person In Charge
=====
S1 : Supplier JB : 0159874632 : Mr. Jayden
=====


```

Figure 4.5.2 Search Supplier

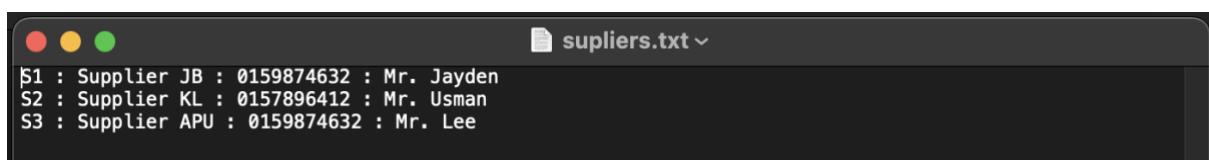


Figure 4.5.3 Supplier Text File

Display supplier management options for admin and staff. By selecting a specific option, admin and staff can add updated supplier details and search suppliers.

4.6 Update Personal Protective Equipment (PPE) Page

```

Do you want to UPDATE PPE STOCK?
Press Y - Update PPE Stock
Press N - return back to Menu

Please enter your option here:y
HC / FS / MS / GL / GW / SC

[['GL', 'S2', '100'], ['GW', 'S1', '100'], ['SC', 'S3', '150'], ['HC', 'S1', '20'], ['FS', 'S3', '180'], ['MS', 'S2', '170']]

Please enter the PPE code to update the stock quantity:gl

Please enter relevant Supplier CODE(S1/S2/S3):S2

R=Received, D=Distributed

Please enter transaction type(R/D):d

Please enter the original stock(in boxes):100

Please enter quantity needed for transaction(in boxes):85

Please enter relevant Hospital CODE(H1/H2/H3):H2

Update Successful

PPE transaction added successfully.

```

Figure 4.6.1 Distributed to Hospital

```

Do you want to UPDATE PPE STOCK?
Press Y - Update PPE Stock
Press N - return back to Menu

Please enter your option here:y
HC / FS / MS / GL / GW / SC

[['GW', 'S1', '100'], ['SC', 'S3', '150'], ['HC', 'S1', '20'], ['FS', 'S3', '180'], ['MS', 'S2', '170'], ['GL', 'S2', '15']]

Please enter the PPE code to update the stock quantity:GW

Please enter relevant Supplier CODE(S1/S2/S3):S1

R=Received, D=Distributed

Please enter transaction type(R/D):R

Please enter the original stock(in boxes):100

Please enter quantity needed for transaction(in boxes):20

Supplier code:S1

Update Successful

```

Figure 4.6.2 Received from Supplier

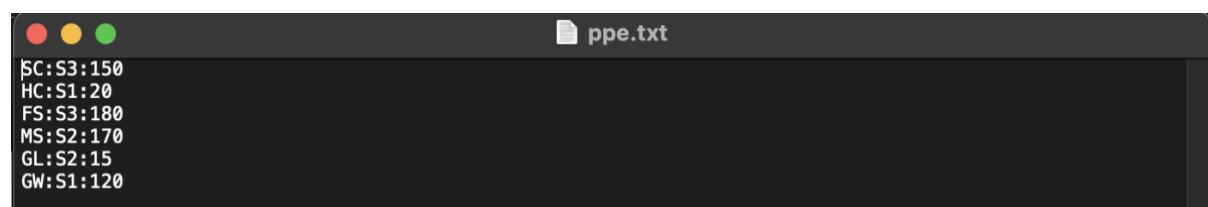


Figure 4.6.3 PPE Text File

Display PPE stock management options for users. Users can update PPE stock by entering the relevant PPE code. At the same time, the user can choose the type of transaction to enter the specific information.

4.7 Item Inventory Tracking Page

```
-----PPE TRACKING-----
Press 1 - Print ALL PPE stock
Press 2 - Print SHORTAGE PPE
Press 3 - Check PARTICULAR PPE stock
Press 4 - Check PPE Transaction
Press 5 - Return back to Menu

Please enter your option here:1

PRINTING PPE STOCK AVAILABLE.....
```

```
PPE Code : Supplier Code : Stock Available
[['FS:S3:180'], ['GL:S2:15'], ['GW:S1:100'], ['HC:S1:20'], ['MS:S2:170'], ['SC:S3:150']]
```

Figure 4.7.1 All PPE Stock

```
Please enter your option here:2

PRINTING SHORTAGE PPE STOCK.....
```

```
GW are available.
SC are available.
The Stock of HC is only left 20 boxes and it's less than 25 boxes.
Please increase the stock available.
FS are available.
MS are available.
The Stock of GL is only left 15 boxes and it's less than 25 boxes.
Please increase the stock available.
```

Figure 4.7.2 Shortage PPE Stock

```
Please enter your option here:3
```

```
Checking Particular PPE STOCK.....
```

```
Please enter the PPE code you would like to search:GW
```

```
PPE code : Supplier Code : Stock
```

```
['GW', 'S1', '100']
```

Figure 4.7.3 Particular PPE Stock

```
Please enter your option here:4
-----
                                         Transaction Record
-----
[{"SC:2023-10-23:19:19:44.754131:R:S3:( '+', , '50')"]
[{"HC:2023-10-23:19:20:19.564174:D:H3:( '-', , '80')"]
[{"FS:2023-10-25:15:18:10.094057:R:S3:( '+', , '80')"]
[{"MS:2023-10-27:16:09:06.675249:R:S2:( '+', , '70')"]
[{"GL:2023-10-27:23:14:53.819238:D:H2:( '-', , '85')"]
```

Figure 4.7.4 PPE Transaction

Display PPE stock inventory tracking options for users. By selecting a specific option, users can check the stock availability, shortage stock for PPE and the transaction for all the PPE items.

4.8 Search Personal Protective Equipment (PPE) Transaction Page

```
You are now in the Menu page
Press 1 - Update Hospital Details
Press 2 - Update Supplier Details
Press 3 - Search Hospital
Press 4 - Search Supplier
Press 5 - Update PPE
Press 6 - Item inventory tracking
Press 7 - Search PPE transaction
Press 8 - Logout

Please enter your option here:7
-----
                                         Transaction Record
-----
[{"FS:2023-10-25:15:18:10.094057:R:S3:( '+', , '80')"]
[{"GL:2023-10-27:23:14:53.819238:D:H2:( '+', , '85')"]
[{"HC:2023-10-23:19:20:19.564174:D:H3:( '+', , '80')"]
[{"MS:2023-10-27:16:09:06.675249:R:S2:( '+', , '70')"]
[{"SC:2023-10-23:19:19:44.754131:R:S3:( '+', , '50')"]
```

Figure 4.8.1 Search PPE According to The Alphabet

Display search PPE transaction options for admin and staff. By selecting the specific option, admin and staff can search the PPE transaction record. Due to the PPE sort according to the alphabet, users can search the relevant PPE code by grouping more easily.

5.0 Conclusion

The Personal Protective Equipment (PPE) Inventory Management System is a testament to the importance of streamlined and efficient inventory management, especially in the world of healthcare. With robust user roles, straightforward interfaces, and a secure authentication system baked into the platform, it ensures that PPEs are tracked from suppliers to hospitals with ease. More than just making administrative work easy, this system also guarantees that healthcare professionals have protective gear when they need it. By using technology in a smart way, it already makes great strides in management.

6.0 References

Python File Handling. . (2023). Retrieved from W3Schools:

https://www.w3schools.com/python/python_file_handling.asp

Python File Methods . . (2023). Retrieved from W3Schools:

https://www.w3schools.com/python/python_ref_file.asp

Python List/Array Methods . . (2023). Retrieved from W3Schools:

https://www.w3schools.com/python/python_ref_list.asp

Python String Methods. . . (2023). Retrieved from W3Schools:

https://www.w3schools.com/python/python_ref_string.asp