INDIVIDUAL ASSIGNMENT

NICHOLAS SUN TP062907

TECHNOLOGY PARK MALAYSIA

CT010-3-1-PYP

PYTHON PROGRAMMING

APD1F2111/APU1F2111 – CE/ME/PE/EEE/TE/ CS/CS(CYB)/CS(DF)/CS(IS)/SE/IT/CGD/MMT

HAND OUT DATE: 28TH DECEMBER 2021

HAND IN DATE: 14TH MARCH 2022

WEIGHTAGE: 100%

INSTRUCTIONS TO CANDIDATES:

1. Submit your assignment online in MS Teams unless advised otherwise

2. Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld

3. Cases of plagiarism will be penalized

4. You must obtain at least 50% in each component to pass this module

Table of Contents

Introduction

Inventory management system is one of the popular and most needed system by most company or businesses. It is very convenient and efficient to own a system that can track the goods quantity in the storage. As a state department of health, they will receive tons of supply from different kind of suppliers and a lot of personal protective equipment that will be distributed to the different kind of hospital in the state. Therefore, the department of health will require an efficient inventory management system to control their inventory distribution and transaction between suppliers and hospitals.

Assumption

The inventory management system that is essential for the department of health items distribution will require a few features inside it. There are 6 main features that will be needed in the system. The first feature is to add the personal protective equipment itself into a text file. The second feature is to create a text file that will contain all the supplier details. The third feature will be creating a text file that contain the hospital details. The first 3 text file which are the ppe.txt, suppliers.txt, and hospital.txt will have 3 other features which is to view all the data inside the text file, to view only one type of data from all data, and to modify the specific detail of the data. The fourth feature will be to distribute the personal protective equipment to the hospital in the hospital text file and record the distribution in the distribution text file. The fifth feature will be recording the transaction when supplier supplied its goods into the department of health. Lastly, the feature to track all the items quantities in the storage and to notify the users of all the items that quantity is less than 25 boxes in the inventory. In addition, the program may have protection to prevent unauthorized access which by creating username and password for each file. The system will then prompt user for the registered username and password every time user wants to do something with the feature of the system.

Program Design

Pseudocode

```
IMPORT datetime

DEFINE startPPEdata()
    CALL genAuthcode()
    DISPLAY("Welcome to the initial inventory creation")
    DISPLAY("Please fill the required information with the correct format")
    TRY
        WITH OPEN "Assignment\ppe.txt","w" AS ppeFile
            DOWHILE True
                DISPLAY("Please enter the PPE item name : ")
                READ itemname
                DISPLAY("Please enter the item code : ")
                READ itemcode
                DISPLAY("Register / Enter Supplier Code : ")
                READ suppliercode
                DECLARE quantity as INTEGER
                quantity = 100
                DISPLAY("Please enter the year item inputted (YYYY) : ")
                READ YYYY
                DISPLAY("Please enter the month item inputted (MM) : ")
```

```
            READ MM
            MM = CALL monthLimit(MM)
            DISPLAY("Please enter the date item inputted (DD) : ")
            READ DD
            DD = CALL dayLimit(YYYY, MM, DD)
            ppeData = itemname+"-"+itemcode+"-"+suppliercode+"-"+STRING(quantity)+"-
"+STRING(YYYY)+"-"+STRING(MM)+"-"+STRING(DD)+"\n"
            WRITE ppeData to ppeFile
            DISPLAY("Press Enter key to continue or [q] to stop : ")
            READ cont
            IF cont WITH lower()=="q" THEN
               DISPLAY("PPE data has been recorded to ppe.txt file")
               BREAK
            ENDIF
          ENDDO
       ENDWITH
       RETURN 1
   EXCEPT
       DISPLAY("Error Format Detected, Please try again")
   ENDTRY
ENDDEFINE
```

```
DEFINE startSupplydata(mode,supnum)
    TRY
        WITH OPEN "Assignment\ppe.txt","r" AS readPPe
            DECLARE ppeList = []
            FOR EACH Data IN readPPe
                APPEND Data TO ppeList WITH strip() and split("-")
            ENDFOR
        ENDWITH
        WITH OPEN "Assignment\suppliers.txt",mode AS insupply
            LOOP i FROM 0 TO supnum STEP 1
                DECLARE checklist = []
                DISPLAY("Please enter registered supplier code / [q] to stop : ")
                READ supID
                IF supID WITH lower() =="q" THEN
                    BREAK
                ENDIF
                WITH OPEN "Assignent\suppliers.txt","r" AS readfh
                    FOR EACH data IN readfh
                        APPEND data TO checklist WITH strip() and split("-")
                    ENDFOR
```

```
DECLARE checkf=-1
LOOP cnt FROM 0 TO len(checklist) STEP 1
    IF supID IN checklist[cnt][1] THEN
        DISPLAY("This supplier code has been inputted")
        checkf=cnt
        BREAK
    ENDIF
    NEXT cnt
ENDLOOP
IF checkf==-1 THEN
    fl = -1
    LOOP cnt FROM 0 TO len(ppeList) STEP 1
        IF supID IN ppeList[cnt][2] THEN
            fl=cnt
            DISPLAY("Item Supplied : "+ ppeList[cnt][0])
            supItem = ppeList[cnt][0]
            DISPLAY("Enter supplier name : ")
            READ supName
            DISPLAY("Enter supplier address : ")
            READ supAddress
            supQuant = ppeList[cnt][3]
```

```
                    suppliers = supName+"-"+supID+"-"+supItem+"-"+supAddress+"-
"+STRING(supQuant)+"\n"
                    WRITE suppliers TO insupply
                    FLUSH insupply
                ENDIF
                NEXT cnt
            ENDLOOP
            IF fl ==-1 THEN
                DISPLAY("Supplier code not found")
            ENDIF
        ENDIF
    ENDWITH
    NEXT i
  ENDLOOP
  DISPLAY("suppliers has been recorded in suppliers.txt")
 ENDWITH
 EXCEPT
  DISPLAY("Error format / Data not found")
 ENDTRY
ENDDEFINE

DEFINE startHospital(mode,hosnum)
```

```
WITH OPEN "Assignment\ppe.txt","r" AS ppefh
    DECLARE list = []
    FOR EACH data IN ppefh
        APPEND data TO list WITH strip() and split("-")
    ENDFOR
ENDWITH


    DECLARE IDlist = []
    FOR EACH cnt IN LIST
        APPEND cnt[1] TO IDlist
    ENDFOR
TRY
    WITH OPEN "Assignment\hospital.txt",mode AS fh
        LOOP x FROM 0 TO hosnum STEP 1
            DISPLAY("Enter Hospital name or [q] to stop : ")
            READ hosName
            IF hosName WITH lower() == "q" THEN
                BREAK
            ELSE
                CALL u_code(0)
                hosID = CALL readID(0)
```

```
            DISPLAY("Enter hospital address : ")
            READ hosAdd
            DISPLAY("Enter hospital contact number : ")
            READ hosContact
            hosData = hosName +"-"+ hosID+"-"+ hosAdd +"-"+ hosContact
            LOOP i FROM 0 TO len(IDlist) STEP 1
               hosData = hosData +"-"+ IDlist[i]+"-"+"0"
               NEXT i
            ENDLOOP
            hosData = hosData +"\n"
            DISPLAY("Registered Hospital Code is : "+ hosID)
            WRITE hosData TO fh
         ENDIF
         DISPLAY("Hospitals has been recorded into hospital.txt")
         NEXT x
      ENDLOOP
   ENDWITH
 EXCEPT
   ("Error format / Data not found")
 ENDTRY
ENDDEFINE
```

```
DEFINE startDistribution(mode)
    TRY
        WITH OPEN "Assignment\ppe.txt" AS ppefh
            DECLARE ppeList = []
            FOR EACH elem IN ppefh:
                APPEND elem TO ppeList WITH strip() and split("-")
            ENDFOR
        ENDWITH

        WITH OPEN "Assignment\hospital.txt" AS hospitalfh
            DECLARE hospitalList = []
            FOR EACH elem IN hospitalfh
                APPEND elem TO hospitalList WITH strip() and split()
            ENDFOR
        ENDWITH

        WITH OPEN "Assignment\distribution.txt",mode AS distFh
            DOWHILE True
                DISPLAY("Enter the item code for distribution : ")
                READ itemID
```

```
itemflag = 1
LOOP line FROM 0 TO len(ppeList) STEP 1
    IF itemID IN ppeList[line][1] THEN
        itemflag = 2
        DISPLAY("This item has quantity of "+ppeList[line][3]+" boxes")
        DISPLAY("How many boxes to distribute ? : "))
        READ send_item
        DECLARE int_itemquantity AS INTEGER OF ppeList[line][3]
        IF send_item>int_itemquantity THEN
            DOWHILE True
                DISPLAY("Inssuficcient amount, Enter the correct amount : "))
                READ send_item
                IF send_item<=int_itemquantity THEN
                    BREAK
                ENDIF
            ENDO
        ENDIF
        DISPLAY("Enter the destination hospital code : ")
        READ hosID
        hospFlag = 1
        l2flag = -1
```

```
LOOP line FROM 0 TO len(hospitalList) STEP 1
    IF hospID IN hospitalList[line][1] THEN
        LOOP cnt FROM 4 TO len(hospitalList[line]) STEP 1
            IF itemID IN hospitalList[line][cnt] AND cnt%2==0 THEN
                DECLARE hosquant AS INTEGER (hospitalList[line][cnt+1])
                receive_quant = hosquant + send_item
                hospitalList[line][cnt+1] AS STRING (receive_quant)
                hospFlag = 2
                DISPLAY("This item has been distributed")
                CALL u_code(1)
                distID = CALL readID(1)
                DISPLAY("Distribution Code is : "+distID)
                distData = itemID+"-"+ STRING(send_item)+"-"+hospID+"-"+distID+"\n"
                WRITE distData TO distFh
                BREAK
            ENDIF
            NEXT cnt
        ENDLOOP
        LOOP cnt FROM 0 TO (len(ppeList)) STEP 1
            IF itemID IN ppeList[cnt][1] THEN
                BREAK
```

```
    ENDIF
DECLARE current_quantity AS INTEGER (ppeList[cnt][3])
new_quant = current_quantity - send_item
DECLARE ppeList[cnt][3] AS STRING (new_quant)
NEXT cnt
ENDLOOP


WITH OPEN "Assignment\ppe.txt","w" AS pfh
    LOOP cnt FROM 0 TO (len(ppeList) STEP 1
        join(ppeList[cnt]) with "-"+"\n" AS rec
        WRITE rec TO pfh
        NEXT cnt
    ENDLOOP
ENDWITH


WITH OPEN "Assignment\hospital.txt","w" AS hfh
    LOOP cnt FROM 0 TO (len(hospitalList)) STEP 1
        join(hospitalList[cnt]) with "-"+"\n" AS rec
        WRITE rec TO hfh
        NEXT cnt
    ENDLOOP
```

```
            ENDWITH


            DISPLAY("Enter [n] to exit this process : ")
            READ cont
            IF cont WITH lower() == "n" THEN
               DISPLAY("Distribution has been recorded to distribution.txt")
               l2flag =1
               BREAK
            ENDIF
         ENDIF
         NEXT line
      ENDLOOP
      IF l2flag == 1 THEN
         BREAK
      ENDIF
      IF hospFlag == 1 THEN
         DISPLAY("Hospital ID not found")
         BREAK
      ENDIF
   ENDIF
   NEXT line
```

```
            ENDLOOP
            IF itemflag == 1 THEN
               DISPLAY("Item not found")
               BREAK
            ENDIF
            IF l2flag==1 THEN
               BREAK
            ENDIF
         ENDDO
      ENDWITH
   EXCEPT
      DISPLAY("Error format/File doesn't Exist")
   ENDTRY
ENDDEFINE


DEFINE startTransactions()
   TRY
      IMPORT datetime
      WITH OPEN "Assignment\suppliers.txt","r" AS supfh
         DECLARE suplist = []
         FOR EACH data IN supfh
```

```
        APPEND data TO suplist WITH strip() and split("-")
    ENDFOR
ENDWITH
WITH OPEN "Assignment\ppe.txt" AS ppefh
    DECLARE ppelist = []
    FOR EACH data IN ppefh
        APPEND data to ppelist WITH strip() and split("-")
    ENDFOR
ENDWITH

WITH OPEN "Assignment\Transactions.txt","a" AS transfh
    DOWHILE True
        DISPLAY("Enter supplier ID : ")
        READ supplierID
        supflag = -1
        l1flag = -1
        LOOP cnt FROM 0 TO (len(suplist)) STEP 1
            IF supplierID IN suplist[cnt][1] THEN
                supflag = cnt
                sup_quant = INTEGER(suplist[cnt][4])
                item_name = suplist[cnt][2]
```

```
LOOP cnt_ppe FROM 0 TO (len(ppelist)) STEP 1
    IF item_name IN ppelist[cnt_ppe][0] THEN
        ItemID = ppelist[cnt_ppe][1]
        BREAK
    ENDIF
    NEXT cnt_ppe
ENDLOOP
DISPLAY("Item ID is :", ItemID)
LOOP cnt FROM 0 TO (len(ppelist)) STEP 1
    IF ItemID IN ppelist[cnt][1] THEN
        ppe_quant = INTEGER(ppelist[cnt][3])
        DISPLAY("Enter quantity of item added : ")
        READ receive_quantity
        new_sup_q = sup_quant + receive_quantity
        new_ppe_q = ppe_quant + receive_quantity
        suplist[cnt][4] = STRING(new_sup_q)
        ppelist[cnt][3] = STRING(new_ppe_q)
        CALL u_code(2)
        TransID = CALL readID(2)
        DISPLAY ("Transaction ID :", TransID)
        Date = FROM datetime CALL date.today()
```

```
                    trans = ItemID + "-"+supplierID+"-"+STRING(receive_quantity)+"-
"+STRING(Date)+"-"+TransID+"\n"
                    WRITE trans TO transfh
                    WITH OPEN "Assignment\ppe.txt","w" AS fh
                       LOOP cnt FROM 0 TO (len(ppelist)) STEP 1
                          join(ppelist[cnt]) with "-"+"\n" AS rec
                          WRITE rec TO fh
                          NEXT cnt
                       ENDLOOP
                    ENDWITH
                    WITH OPEN "Assignment\suppliers.txt","w" AS fh
                       LOOP cnt FROM 0 TO (len(suplist)) STEP 1
                          join(suplist[cnt]) with "-"+"\n" AS rec
                          WRITE rec TO fh
                          NEXT cnt
                       ENDLOOP
                    ENDWITH
                    DISPLAY("Do you wish to make another transactions? [n] to stop : ")
                    READ cont
                    IF cont WITH lower() == "n" THEN
                       DISPLAY("Transactions has been recorded in Transactions.txt")
                       l1flag = 1
```

```
                        BREAK
                    ENDIF
                NEXT cnt
            ENDLOOP
        NEXT cnt
    ENDLOOP
    IF l1flag ==1 THEN
        BREAK
    ENDIF
    IF supflag == -1 THEN
        DISPLAY("Supplier Code not found")
        BREAK
    ENDIF
  ENDDO
 EXCEPT
   DISPLAY("Error format/Data not found")
 ENDTRY
ENDDEFINE


DEFINE AddPPEdata()
 TRY
```

```
WITH OPEN "Assignment\ppe.txt","a" AS ppeFile
    DOWHILE True
        DISPLAY("Please enter the PPE item name : ")
        READ itemname
        DISPLAY("Please enter the item code : ")
        READ itemcode
        DISPLAY("Register / Enter Supplier Code : ")
        READ suppliercode
        DECLARE quantity as INTEGER
        quantity = 100
        DISPLAY("Please enter the year item inputted (YYYY) : ")
        READ YYYY
        DISPLAY("Please enter the month item inputted (MM) : ")
        READ MM
        MM = CALL monthLimit(MM)
        DISPLAY("Please enter the date item inputted (DD) : ")
        READ DD
        DD = CALL dayLimit(YYYY, MM, DD)
        ppeData = itemname+"-"+itemcode+"-"+suppliercode+"-"+STRING(quantity)+"-
"+STRING(YYYY)+"-"+STRING(MM)+"-"+STRING(DD)+"\n"
        WRITE ppeData TO ppeFile
        DISPLAY("Press Enter key to continue or [q] to stop : ")
```

```
            READ cont
            IF cont WITH lower()=="q" THEN
               DISPLAY("PPE data has been recorded to ppe.txt file")
               BREAK
            ENDIF
         ENDDO
      ENDWITH
   EXCEPT
      DISPLAY("Error format/Please Try again")
   ENDTRY
ENDDEFINE


DEFINE u_code(ind)
   DECLARE codeList = []
   WITH OPEN "Assignment\codegate.txt","r" AS algocode
      FOR EACH code IN algocode
         APPEND code to codeList WITH strip() and split("-")
      ENDFOR
      num = codeList[0][ind]
      numtemp = SLICE 6 words FROM num
      numonly = INTEGER(numtemp)
```

```
    DECLARE numList = []
    numonly+=1
    APPEND numonly TO numList
  ENDWITH
  DECLARE newcode=[]
  APPEND (codeList[0][ind][FIRST 6 WORDS]) TO newcode
  tempCode = newcode + (numList)
  genCode = ""
  LOOP i FROM 0 TO 2 STEP 1
    genCode = genCode+STRING(tempCode[i])
    NEXT i
  ENDLOOP
  codeList[0][ind] = genCode
  WITH OPEN "Assignment\codegate.txt","w" AS codewriter
    FOR EACH code IN codeList
      join(code) with "-"+"\n" AS cd
    ENDFOR
    WRITE cd TO codewriter
  ENDWITH
ENDDEFINE
```

```
DEFINE readID(ind)
    DECLARE codelist = []
    WITH OPEN "Assignment\codegate.txt","r" AS readsup
        FOR EACH code IN readsup
            APPEND code TO codelist WITH strip() and split("-")
        ENDFOR
    ENDWITH
    ID = codelist[0][ind]
    RETURN ID
ENDDEFINE

DEFINE genAuthcode()
    WITH OPEN "Assignment\codegate.txt","w" AS authFile
        auth = "HID9890"+"-"+"DID6560"+"-"+"TID7470"
        WRITE auth TO authFile
    ENDWITH
ENDDEFINE

DEFINE userSignUp(file)
    WITH OPEN "Assignment\login.txt","a" AS loginFile
        DOWHILE True
```

```
        DISPLAY("Enter your User ID : ")
        READ userName
        DISPLAY("Enter your Password : ")
        READ userPW
        LoginData = file +"-"+userName +"-"+ userPW+"\n"
        WRITE LoginData TO loginFile
        DISPLAY("User account has been succesfully saved")
        BREAK
    ENDDO
  ENDWITH
ENDDEFINE


DEFINE userLogin(file)
  TRY
    WTIH OPEN "Assignment\login.txt","r" AS readLogin
      DECLARE USERLOGDATA = []
      FOR EACH userData IN readLogin
        APPEND userData TO USERLOGDATA WITH strip() and split("-")
      ENDFOR
    ENDWITH
```

```
        LOOP cnt FROM 0 TO (len(USERLOGDATA)) STEP 1
            IF file IN USERLOGDATA[cnt][0] THEN
                DISPLAY("Enter existing User ID : ")
                READ userLogin
                DISPLAY("Enter Password : ")
                READ userPassword
                IF USERLOGDATA[cnt][1] == userLogin and USERLOGDATA[cnt][2] == userPassword
THEN
                    DISPLAy("Succesful")
                    RETURN True
                ELSE
                    DISPLAY("Login failed")
                ENDIF
            ENDIF
            NEXT cnt
        ENDLOOP
    EXCEPT
        DISPLAY("Error format/Data not found")
    ENDTRY
ENDDEFINE


DEFINE viewInformation()
```

```
DECLARE ppeInfo = []
DECLARE supInfo = []
DECLARE combInfo = []
WITH OPEN "Assignment\ppe.txt","r" AS readPPE
    FOR EACH info IN readPPE
        APPEND info TO ppeInfo WITH strip() and split("-")
    ENDFOR
ENDWITH
WITH OPEN "Assignment\suppliers.txt","r" AS readSuppliers
    FOR EACH info IN readSuppliers
        APPEND info TO supInfo WITH strip() and split("-")
    ENDFOR
ENDWITH
DECLARE allList = []
LOOP cnt FROM 0 TO (len(supInfo)) STEP 1
    combInfo = ppeInfo[cnt] + supInfo[cnt]
    APPEND combInfo TO allList
    NEXT cnt
ENDLOOP
LOOP x FROM 0 TO (len(allList)-1) STEP 1
    LOOP y FROM x+1 TO (len(allList)) STEP 1
```

```
            IF allList[x][0] > allList[y][0] THEN
                temp = allList[x]
                allList[x] = allList[y]
                allList[y] = temp
            ENDIF
            NEXT y
        ENDLOOP
        NEXT x
    ENDLOOP
    CALL headerView(allList)
ENDDEFINE


DEFINE view25()
    DECLARE allList = []
    WITH OPEN "Assignment\ppe.txt","r" AS readPPE
        FOR EACH info IN readPPE
            APPEND info TO allList WITH strip() and split("-")
        ENDFOR
    ENDWITH
    flag = -1
    LOOP cnt FROM 0 TO (len(allList)) STEP 1
```

```
        quant = INTEGER(allList[cnt][3])
        item_name = allList[cnt][0]
        item_ID = allList[cnt][1]
        IF quant < 25 THEN
            flag = cnt
            DISPLAY(item_name+" "+item_ID+" has quantity of "+STRING(quant)+" boxes left")
        ENDIF
        NEXT cnt
    ENDLOOP
    IF flag == -1 THEN
        DISPLAY("All item stock is above 25 boxes")
    ENDIF
ENDDEFINE

DEFINE ppeview()
    WITH OPEN "Assignment\ppe.txt" AS fh
        DECLARE ppe = []
        FOR EACH data IN fh
            APPEND data TO ppe WITH strip() and split("-")
        ENDFOR
        DISPLAY("="* 75)
```

```
    DISPLAY("Item Name".center(15)+"|"+"Item Code".center(20)+"|"+"Item
Quantity(Box)".center(20)+"|"+"Date Inputted".center(15))

    DISPLAY("="* 75)

    LOOP line FROM 0 TO (len(ppe)) STEP 1

DISPLAY(ppe[line][0].center(15)+"|"+ppe[line][1].center(20)+"|"+ppe[line][3].center(20)+"|"
+(ppe[line][4]+"/"+ppe[line][5]+"/"+ppe[line][6]).center(15))

        NEXT line

    ENDLOOP

  ENDWITH

ENDDEFINE


DEFINE supplyview()

  WITH OPEN "Assignment\suppliers.txt" AS fh

    DECLARE supplier = []

    FOR EACH data IN fh

      APPEND data TO supplier WITH strip() and split("-")

    ENDFOR

    DISPLAY("="* 95)

    DISPLAY("Supplier Name".center(15)+"|"+"Supplier Code".center(20)+"|"+"Item
Supplied".center(20)+"|"+"Supplier Address".center(20)+"|"+"Item quantity".center(15))

    DISPLAY("="* 95)

    LOOP line FROM 0 TO (len(supplier)) STEP 1
```

```
DISPLAY(supplier[line][0].center(15)+"|"+supplier[line][1].center(20)+"|"+supplier[line][2].center(20)+"|"+supplier[line][3].center(20)+"|"+supplier[line][4].center(15))
            NEXT line
        ENDLOOP
    ENDWITH
ENDDEFINE

DEFINE hospitalview()
    WITH OPEN "Assignment\hospital.txt") AS fh
        DECLARE hospital = []
        FOR EACH data IN fh
            APPEND data TO supplier WITH strip() and split("-")
        ENDFOR
        DISPLAY("="* 75)
        DISPLAY("Hospital Name".center(15)+"|"+"Hospital Code".center(20)+"|"+"Hospital Address".center(20)+"|"+"Contact Number".center(15))
        DISPLAY("="* 75)
        LOOP line FROM 0 TO (len(hospital)) STEP 1

DISPLAY(hospital[line][0].center(15)+"|"+hospital[line][1].center(20)+"|"+hospital[line][2].center(20)+"|"+hospital[line][3].center(15))
            NEXT line
```

```
        ENDLOOP
    ENDWITH
ENDDEFINE


DEFINE headerView(lists):
    DISPLAY("="* 136)
    DISPLAY("Item Name".center(15)+"|"+"Item Code".center(20)+"|"+"Item Quantity(Box)".center(20)+"|"+"Supplier Name".center(20)+"|"+"Supplier Code".center(20)+"|"+"Supplier Address".center(20)+"|"+"Date Inputted".center(15))
    DISPLAY("="* 136)
    LOOP line FROM 0 TO (len(lists)) STEP 1

DISPLAY(lists[line][0].center(15)+"|"+lists[line][1].center(20)+"|"+lists[line][3].center(20)+"|"+lists[line][7].center(20)+"|"+lists[line][8].center(20)+"|"+lists[line][10].center(20)+"|"+(lists[line][4]+"/"+lists[line][5]+"/"+lists[line][6]).center(15))
        NEXT line
    ENDLOOP
ENDDEFINE


DEFINE viewDistribution()
    WITH OPEN "Assignment\distribution.txt","r" AS disfh
        DECLARE dislist = []
        FOR EACH elem IN disfh
```

```
        APPEND elem TO dislist WITH strip() and split("-")
    ENDFOR
    DISPLAY("="* 77)
    DISPLAY("Item Code".center(15)+"|"+"Item Quantity(Box)".center(20)+"|"+"Hospital Code".center(20)+"|"+"Distribution Code".center(20))
    DISPLAY("="* 77)
    LOOP line FROM 0 TO (len(dislist)) STEP 1

DISPLAY(dislist[line][0].center(15)+"|"+dislist[line][1].center(20)+"|"+dislist[line][2].center(20)+"|"+dislist[line][3].center(20))
        NEXT line
    ENDLOOP
  ENDWITH
ENDDEFINE


DEFINE modifyPPE()
  TRY
    DECLARE data = []
    WITH OPEN "Assignment\ppe.txt","r" AS fh
      FOR EACH line IN fh
        elem = line WITH strip() and split("-")
        APPEND elem TO data
      ENDFOR
```

```
ENDWITH
DISPLAY("Please enter Item Code : ")
READ skey
flg = -1
LOOP cnt FROM 0 TO (len(data))
   IF skey IN data[cnt][1] THEN
      flg = cnt
      BREAK
   ENDIF
   NEXT cnt
ENDLOOP
IF flg == -1 THEN
   DISPLAY("Data not found")
ENDIF
IF flg != -1 THEN
   DISPLAY("1 -Item Name   : "+ data[flg][0])
   DISPLAY("2 -Item Code     : "+ data[flg][1])
   DISPLAY("3 -Supplier Code : "+ data[flg][2])
   DISPLAY("4 -Quantity : "+ data[flg][3])
   DISPLAY("5- Year : "+ data[flg][4])
   DISPLAY("6 -Month : "+ data[flg][5])
```

```
        DISPLAY("7 -Date : "+ data[flg][6])
        DISPLAY("Ënter the number to modify :"))
        READ ans
        DISPLAY("Ënter a new value: ")
        ans = CALL limitopt(ans,7)
        READ data[cnt][ans-1]
        DISPLAY("***Succesfully changed***")
        WITH OPEN "Assignment\ppe.txt","w" AS fh
          LOOP cnt FROM 0 TO (len(data)) STEP 1
            join(data[cnt]) with "-"+"\n" AS rec
            WRITE rec TO fh
            NEXT cnt
          ENDLOOP
        ENDWITH
      ENDIF
    EXCEPT
      DISPLAY("Error or File not found")
    ENDTRY
ENDDEFINE

DEFINE modifySup()
```

```
TRY
    DECLARE data = []
    WITH OPEN "Assignment\suppliers.txt","r" AS fh
        FOR EACH line IN fh
            elem = line WITH strip() and split("-")
            APPEND elem TO data
        ENDFOR
    ENDWITH
    DISPLAY("Please enter Supplier Code : ")
    READ skey
    flg = -1
    LOOP cnt FROM 0 TO (len(data))
        IF skey IN data[cnt][1] THEN
            flg = cnt
            BREAK
        ENDIF
        NEXT cnt
    ENDLOOP
    IF flg == -1 THEN
        DISPLAY("Data not found")
    ENDIF
```

```
    IF flg != -1 THEN
        DISPLAY("1 -Supplier Name : "+ data[flg][0])
        DISPLAY("2 -Supplier Code : "+ data[flg][1])
        DISPLAY("3 -Item Supplied : "+ data[flg][2])
        DISPLAY("4 -Supplier Address : "+ data[flg][3])
        DISPLAY("5- Quantity Supplied : "+ data[flg][4])
        DISPLAY("Ënter the number to modify :"))
        READ ans
        DISPLAY("Ënter a new value: ")
        ans = CALL limitopt(ans,7)
        READ data[cnt][ans-1]
        DISPLAY("***Succesfully changed***")
        WITH OPEN "Assignment\ppe.txt","w" AS fh
            LOOP cnt FROM 0 TO (len(data)) STEP 1
                join(data[cnt]) with "-"+"\n" AS rec
                WRITE rec TO fh
                NEXT cnt
            ENDLOOP
        ENDWITH
    ENDIF
EXCEPT
```

```
        DISPLAY("Error or File not found")
    ENDTRY
ENDDEFINE


DEFINE modifyHosp()
    TRY
        DECLARE data = []
        WITH OPEN "Assignment\hospital.txt","r" AS fh
            FOR EACH line IN fh
                elem = line WITH strip() and split("-")
                APPEND elem TO data
            ENDFOR
        ENDWITH
        DISPLAY("Please enter Supplier Code : ")
        READ skey
        flg = -1
        LOOP cnt FROM 0 TO (len(data))
            IF skey IN data[cnt][1] THEN
                flg = cnt
                BREAK
            ENDIF
```

```
    NEXT cnt
ENDLOOP
IF flg == -1 THEN
    DISPLAY("Data not found")
ENDIF
IF flg != -1 THEN
    DISPLAY("1 -Hospital Name : "+ data[flg][0])
    DISPLAY("2 -Hospital Code : "+ data[flg][1])
    DISPLAY("3 -Hospital Address : "+ data[flg][2])
    DISPLAY("4 -Contact Number : "+ data[flg][3])
    DISPLAY("Ënter the number to modify :"))
    ans = CALL limitopt(ans,4)
    READ ans
    DISPLAY("Ënter a new value: ")
    READ data[cnt][ans-1]
    DISPLAY("***Succesfully changed***")
    WITH OPEN "Assignment\hospital.txt","w" AS fh
        LOOP cnt FROM 0 TO (len(data)) STEP 1
            join(data[cnt]) with "-"+"\n" AS rec
            WRITE rec TO fh
            NEXT cnt
```

```
            ENDLOOP
        ENDWITH
    ENDIF
  EXCEPT
    DISPLAY("Error or File not found")
  ENDTRY
ENDDEFINE


DEFINE searchPPEDetail()
  TRY
    DECLARE ppeInfo = []
    WITH OPEN "Assignment\ppe.txt","r" AS readPPE
      FOR EACH info IN readPPE:
        APPEND info TO ppeInfo WITH strip() and split("-")
      ENDFOR
    ENDWITH
    DISPLAY("Please enter item Code : ")
    READ skey
    flag = -1
    LOOP line FROM 0 TO (len(ppeInfo)) STEP 1
      IF skey IN ppeInfo[line][1] THEN
```

```
            flag = line
            DISPLAY("="* 115)
            DISPLAY("Item        Name".center(15)+"|"+"Item        Code".center(20)+"|"+"Supplier
Code".center(20)+"|"+"Item Quantity(Box)".center(20)+"|"+"Date Inputted".center(15))
            DISPLAY("="* 115)

DISPLAY(ppeInfo[line][0].center(15)+"|"+ppeInfo[line][1].center(20)+"|"+ppeInfo[line][2].ce
nter(20)+"|"+ppeInfo[line][3].center(20)+"|"+(ppeInfo[line][4]+"/"+ppeInfo[line][5]+"/"+ppeIn
fo[line][6]).center(15))
            BREAK
         ENDIF
         NEXT line
      ENDLOOP
      IF flag == -1 THEN
         DISPLAY("Data not found")
      ENDIF
   EXCEPT
      DISPLAY("Error format/Data not exist")
   ENDTRY
ENDDEFINE

DEFINE searchSupDetail()
   TRY
```

```
DECLARE supInfo = []
WITH OPEN "Assignment\suppliers.txt","r" AS readSup
   FOR EACH info IN readSup
      APPEND info TO supInfo WITH strip() and split("-")
   ENDFOR
ENDWITH
DISPLAY("Please enter supplier Code : ")
READ skey
flag = -1
LOOP line FROM 0 TO (len(supInfo)) STEP 1
   IF skey IN supInfo[line][1] THEN
      flag = line
      DISPLAY("="* 116)
      DISPLAY("Supplier     Name".center(15)+"|"+"Supplier     Code".center(20)+"|"+"Item Supplied".center(20)+"|"+"Supplier Address".center(20)+"|"+"Quantity Supply".center(20))
      DISPLAY("="* 116)

DISPLAY(supInfo[line][0].center(15)+"|"+supInfo[line][1].center(20)+"|"+supInfo[line][2].center(20)+"|"+supInfo[line][3].center(20)+"|"+supInfo[line][4].center(15))
      BREAK
   ENDIF
   NEXT line
ENDLOOP
```

```
        IF flag == -1 THEN
            DISPLAY("Data not found")
        ENDIF
    EXCEPT
        DISPLAY("Error format/Data not exist")
    ENDTRY
ENDDEFINE


DEFINE searchHospDetail()
    TRY
        DECLARE hospeInfo = []
        WITH OPEN "Assignment\hospital.txt","r" AS readfh
            FOR EACH info IN readfh
                APPEND info TO supInfo WITH strip() and split("-")
            ENDFOR
        ENDWITH
        DISPLAY("Please enter Hospital Code : ")
        READ skey
        flag = -1
        LOOP line FROM 0 TO (len(hospeInfo)) STEP 1
            IF skey IN hospeInfo[line][1] THEN
```

```
            flag = line
            DISPLAY("="* 75)
            DISPLAY("Hospital  Name".center(15)+"|"+"Hospital  Code".center(20)+"|"+"Hospital Address".center(20)+"|"+"Contact Number".center(15))
            DISPLAY("="* 75)

DISPLAY(hospeInfo[line][0].center(15)+"|"+hospeInfo[line][1].center(20)+"|"+hospeInfo[line][2].center(20)+"|"+hospeInfo[line][3].center(15))
            BREAK
         ENDIF
         NEXT line
      ENDLOOP
      IF flag == -1 THEN
         DISPLAY("Data not found")
      ENDIF
   EXCEPT
      DISPLAY("Error format/Data not exist")
   ENDTRY
ENDDEFINE

DEFINE searchDistribution()
   TRY
      DECLARE dislist = []
```

```
    WITH OPEN "Assignment\distribution.txt","r" AS disfh
      FOR EACH info IN disfh
        APPEND info TO supInfo WITH strip() and split("-")
      ENDFOR
    ENDWITH
    DISPLAY("Please enter item Code : ")
    READ skey
    flag = -1
    DISPLAY("="* 77)
    DISPLAY("Item  Code".center(15)+"|"+"Item  Quantity(Box)".center(20)+"|"+"Hospital Code".center(20)+"|"+"Distribution Code".center(20))
    DISPLAY("="* 77)
    LOOP line FROM 0 TO len(dislist)
      IF skey IN dislist[line][0] THEN
        flag = line
        DOWHILE True

DISPLAY(dislist[line][0].center(15)+"|"+dislist[line][1].center(20)+"|"+dislist[line][2].center(20)+"|"+dislist[line][3].center(20))
          BREAK
        ENDDO
      ENDIF
      NEXT line
```

```
        ENDLOOP
        IF flag == -1 THEN
            DISPLAY("Data not found")
        ENDIF
    EXCEPT
        DISPLAY("Error format/Data not exist")
    ENDTRY
ENDDEFINE



DEFINE monthLimit(MM)
    IF MM<0 OR MM>12 THEN
        mFlag = -1
        DOWHILE mFlag == -1
            DISPLAY("Please enter the month in the proper format (MM) : ")
            READ MM
            IF MM>0 AND MM<=12 THEN
                mFlag=MM
                BREAK
            ENDIF
        ENDDO
```

```
    ENDIF
    RETURN MM
ENDDEFINE

DEFINE dayLimit(YYYY, MM, DD)
    IF MM == 4 or MM == 6 or MM == 9 or MM == 11 THEN
        IF DD<0 OR DD>30 THEN
            dFlag = -1
            DOWHILE dFlag == -1
                DISPLAY("Please enter the date in the proper format : ")
                READ DD
                IF DD>0 AND DD<=30 THEN
                    dFlag = DD
                    BREAK
                ENDIF
            ENDDO
        ENDIF

    ELSEIF YYYY%4==0 AND MM==2 THEN
        IF DD<0 OR DD>29 THEN
            d2Flag = -1
```

```
        DOWHILE d2Flag == -1
            DISPLAY("Please enter the date with the correct format : ")
            READ DD
            IF DD>0 AND DD<=29 THEN
                d2Flag == DD
                BREAK
            ENDIF
        ENDDO
    ENDIF
ELSEIF YYYY%4!= AND MM == 2 THEN
    IF DD<0 OR DD>28 THEN
        d3Flag = -1
        DOWHILE d3Flag == -1
            DISPLAY(input("Please enter the date with the correct format : ")
            IF DD>0 AND DD<=28 THEN
                d3Flag == DD
                BREAK
            ENDIF
        ENDO
    ENDIF
ELSEIF DD<0 OR DD>31 THEN
```

```
        d4Flag = -1
        DOWHILE d4Flag == -1
            DISPLAY("Please enter the date the correct format : "))
            IF DD>0 AND DD<=31 THEN
                d4Flag = DD
                BREAK
            ENDIF
        ENDDO
    ENDIF
    RETURN DD
ENDDEFINE


DEFINE limitopt(opt,upper)
    DOWHILE opt<1 OR opt>upper
        DISPLAY("Enter according to the option available : ")
        IF opt IN RANGE FROM 1 UNTIL upper+1 THEN
            BREAK
        ENDIF
    ENDDO
    RETURN opt
ENDDEFINE
```

```
DEFINE gate(arr,opt,file)
    IF arr[opt-1] == 0  THEN
        userSignUp(file)
        arr[opt-1] =1
    ELSE
        DISPLAY("*** Account Exist ***")
    ENDIF
ENDDEFINE


DEFINE ALL_MENU
    TRY
        flag = CALL startPPEdata()
        DECLARE MENU_ARRAY = [1,0,0,0,0]
        DECLARE SUB_MENU_FLAG = [0,0,0,0,0]
        DOWHILE True
            IF flag == 1 THEN
                DOWHILE True
                    DISPLAY("1- Create Admin Account")
                    DISPLAY("2- PPE File")
                    DISPLAY("3- Suppliers File")
```

```
DISPLAY("4- Hospital File")
DISPLAY("5- Move to next menu")
DISPLAY("Enter your option : ")
READ option
option = CALL limitopt(option,5)
IF option ==1 THEN
    DOWHILE True
        DISPLAY("1. ppe.txt")
        DISPLAY("2- suppliers.txt")
        DISPLAY("3- hospital.txt")
        DISPLAY("4- Distribution")
        DISPLAY("5- Transactions")
        DISPLAY("6- Move to Last Menu")
        DISPLAY("Enter your option : ")
        READ ask
        ask = CALL limitopt(ask,6)
        IF ask == 1 THEN
            CALL gate(SUB_MENU_FLAG,ask,"ppe")
        ELSEIF ask ==2 THEN
            CALL gate(SUB_MENU_FLAG,ask,"suppliers")
        ELSEIF ask== 3 THEN
```

```
            CALL gate(SUB_MENU_FLAG,ask,"hospital")
        ELSEIF ask == 4 THEN
            CALL gate(SUB_MENU_FLAG,ask,"distribution")
        ELSEIF ask == 5 THEN
            CALL gate(SUB_MENU_FLAG,ask,"Transactions")
        ELSEIF ask == 6 THEN
            BREAK
        ENDIF
    ENDDO
ELSEIF option == 2 AND SUB_MENU_FLAG[option-2]==0 THEN
    DISPLAY("Please Create Admin Account")
ELSEIF option==2 AND MENU_ARRAY[option-2]==1 AND SUB_MENU_FLAG[option-
2]==1 THEN
    IF CALL userLogin("ppe") == True THEN
        DOWHILE True
            DISPLAY("1. View Personal Protective Equipment Data")
            DISPLAY("2- Search Specific Item")
            DISPLAY("3- Modify data in File")
            DISPLAY("4- Move to Last Menu")
            DISPLAY("Enter your option : ")
            READ ask
            ask = CALL limitopt(ask,4)
```

```
                IF ask==1 THEN
                    CALL ppeview()
                ELSEIF ask==2 THEN
                    CALL searchPPEDetail()
                ELSEIF ask==3 THEN
                    CALL modifyPPE()
                ELSEIF ask == 4 THEN
                    BREAK
                ENDIF
            ENDDO
        ENDIF
    ELSEIF option == 3 AND MENU_ARRAY[option-2]==0 THEN
        DISPLAY("Enter how many suppliers to input")
        READ supnum
        CALL startSupplydata("w",supnum)
        MENU_ARRAY[option-2]=1
    ELSEIF option ==3 AND SUB_MENU_FLAG[option-2]==0 THEN
        DISPLAY("Please Create Admin Account")
    ELSEIF option==3 AND MENU_ARRAY[option-2]==1 AND SUB_MENU_FLAG[option-
2]==1 THEN
        IF CALL userLogin("suppliers") == True THEN
            DOWHILE True
```

```
            DISPLAY("1. View Suppliers Data")
            DISPLAY("2- Search Specific Supplier")
            DISPLAY("3- Modify data in File")
            DISPLAY("4- Move to Last Menu")
            DISPLAY("Enter your option : ")
            READ ask
            ask = CALL limitopt(ask,4)
            IF ask==1 THEN
               CALL supplyview()
            ELISEF ask==2 THEN
               CALL searchSupDetail()
            ELSEIF ask==3 THEN
               CALL modifySup()
            ELSEIF ask == 4 THEN
               BREAK
            ENDIF
         ENDDO
      ENDIF
   ELSEIF option == 4 AND MENU_ARRAY[option-2]==0 THEN
      DISPLAY("Enter how many hospital to input")
      READ hosnum
```

```
            CALL startHospital("w",hosnum)
            MENU_ARRAY[option-2]=1
        ELSEIF option ==4 AND SUB_MENU_FLAG[option-2]==0 THEN
            DISPLAY("Please Create Admin Account!!")
        ELSEIF option ==4 AND MENU_ARRAY[option-2]==1 AND
SUB_MENU_FLAG[option-2]==1 THEN
            IF CALL userLogin("hospital") == True THEN
                DOWHILE True
                    DISPLAY("1. View Hospital Data")
                    DISPLAY("2- Search Specific Hospital")
                    DISPLAY("3- Modify data in File")
                    DISPLAY("4- Move to Last Menu")
                    DISPLAY("Enter your option : ")
                    READ ask
                    ask = CALL limitopt(ask,4)
                    IF ask==1 THEN
                        CALL hospitalview()
                    ELSEIF ask==2 THEN
                        CALL searchHospDetail()
                    ELSEIF ask==3 THEN
                        CALL modifyHosp()
                    ELSEIF ask == 4 THEN
```

```
                    BREAK
                ENDIF
            ENDDO
        ENDIF
    ELSEIF option == 5 THEN
        flag = 2
        BREAK
    ENDIF
ENDDO
IF flag == 2 THEN
    DOWHILE True
        DISPLAY("1 - Register more PPE Item")
        DISPLAY("2 - Register more Supplier")
        DISPLAY("3 - Register more Hospital")
        DISPLAY("4 - Hospital Distribution")
        DISPLAY("5 - Supplier Transaction")
        DISPLAY("6 - Inventory Tracking")
        DISPLAY("7 - Distribution List")
        DISPLAY("8 - Move to Last Menu")
        DISPLAY("9 - End the Programme")
        DISPLAY("Enter your option : ")
```

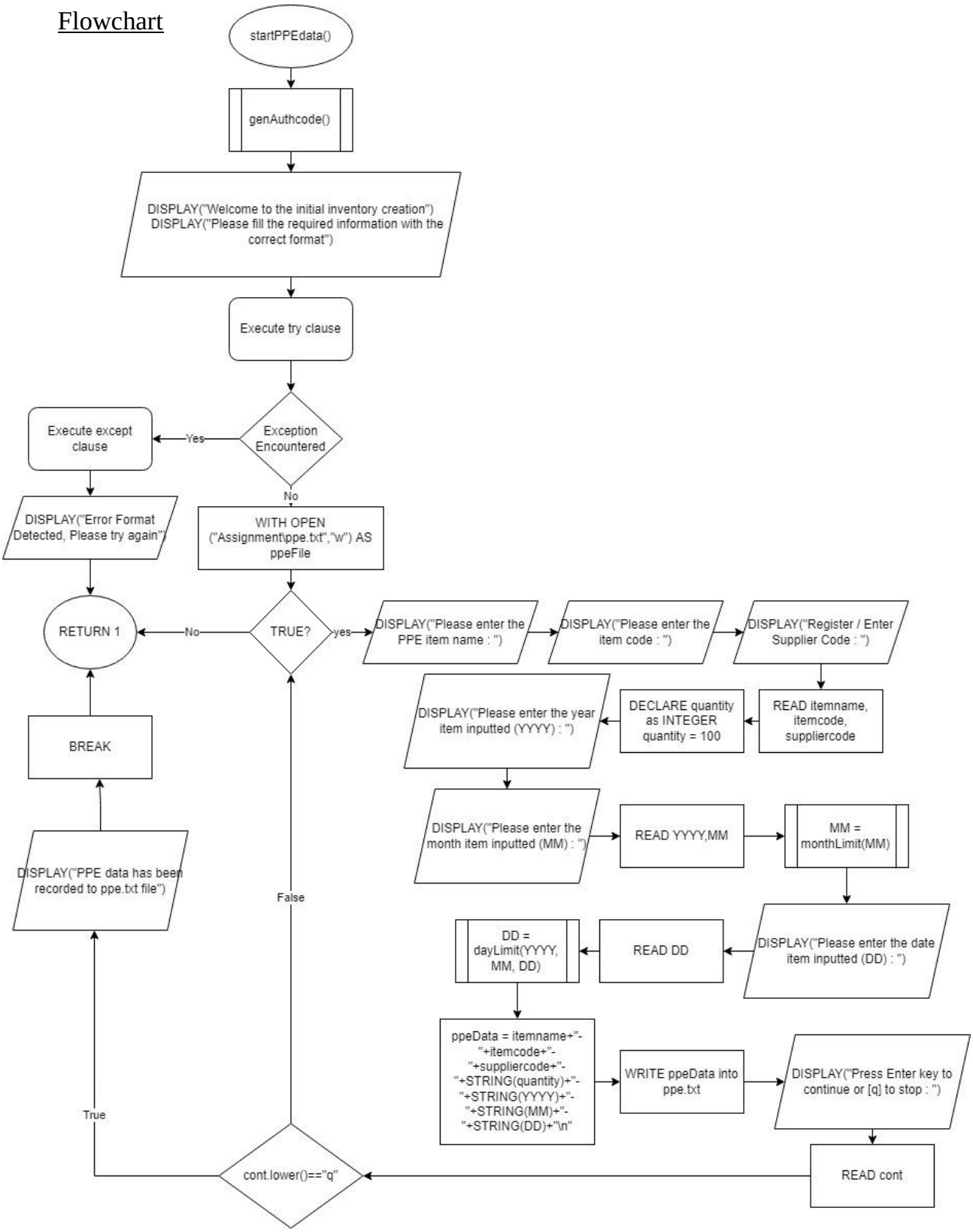
```
READ option
option = CALL limitopt(option,9)
IF option == 1 THEN
    CALL AddPPEdata()
ELSEIF option == 2 THEN
    DISPLAY("Enter how many suppliers to input : ")
    READ supnum
    CALL startSupplydata("a",supnum)
ELSEIF option == 3 THEN
    DISPLAY("Enter how many hospital to input : ")
    READ hosnum
    CALL startHospital("a",hosnum)
ELSEIF option == 4 THEN
    IF MENU_ARRAY[option-2]==1 THEN
        IF userLogin("distribution")==True THEN
            CALL startDistribution("a")
        ENDIF
    ELSE
        print("hospital.txt / Admin Account doesn't exist")
    ENDIF
ELSEIF option == 5 THEN
```

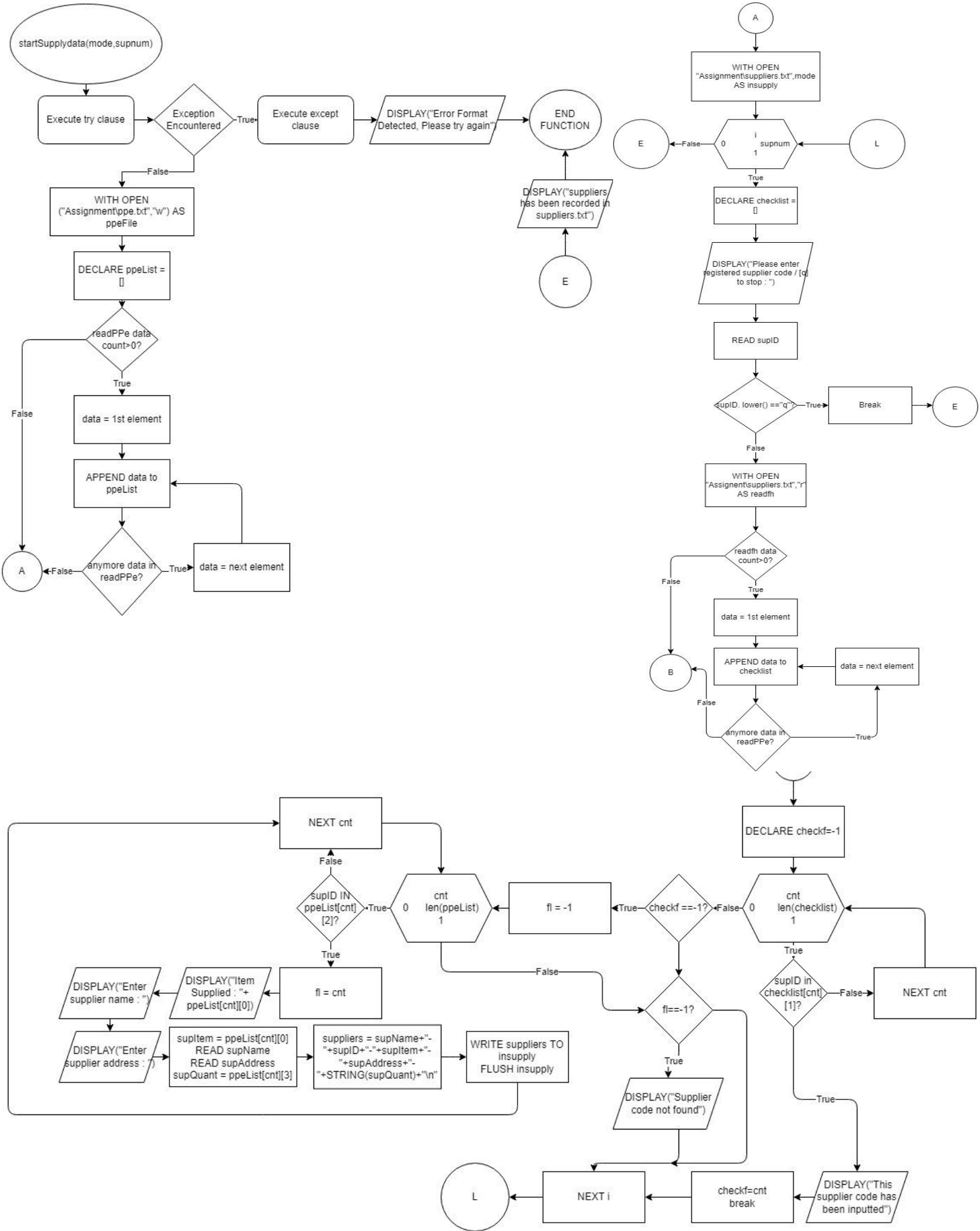
```
    IF MENU_ARRAY[1]==1 THEN
        IF userLogin("Transactions")==True THEN
            CALL startTransactions()
        ENDIF
    ELSE
        DISPLAY("suppliers.txt / Admin account does not exist")
    ENDIF
ELSEIF option == 6 THEN
    DOWHILE True
        DISPLAY("1. View quantity available for all item")
        DISPLAY("2. View all item < 25 boxes")
        DISPLAY("3. Move to Last Menu")
        DISPLAY("Enter your option : ")
        READ ask
        ask = CALL limitopt(ask,3)
        IF ask == 1 THEN
            CALL viewInformation()
        ELSEIF ask ==2 THEN
            CALL view25()
        ELSEIF ask ==3
            BREAK
```

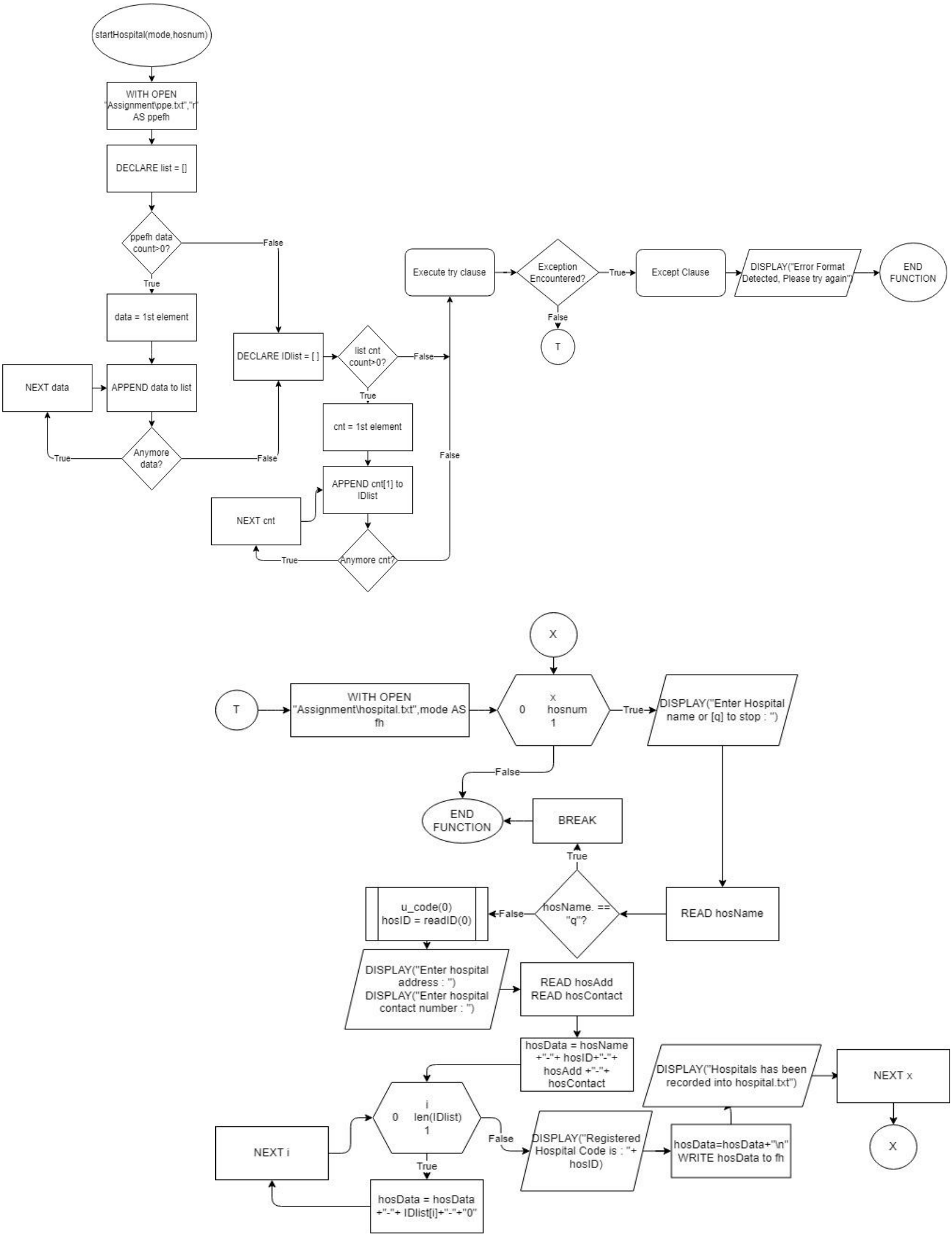
```
        ENDIF
    ENDDO
ELSEIF option==7 THEN
    DOWHILE True
        DISPLAY("1. View All Distribution")
        DISPLAY("2. Search for specific item distribution")
        DISPLAY("3. Move to Last Menu")
        DISPLAY("Enter your option : ")
        READ ask
        ask = CALL limitopt(ask,3)
        IF ask ==1 THEN
            CALL viewDistribution()
        ELSEIF ask ==2 THEN
            CALL searchDistribution()
        ELSEIF ask==3 THEN
            BREAK
        ENDIF
    ENDDO
ELSEIF option == 8 THEN
    flag = 1
    BREAK
```

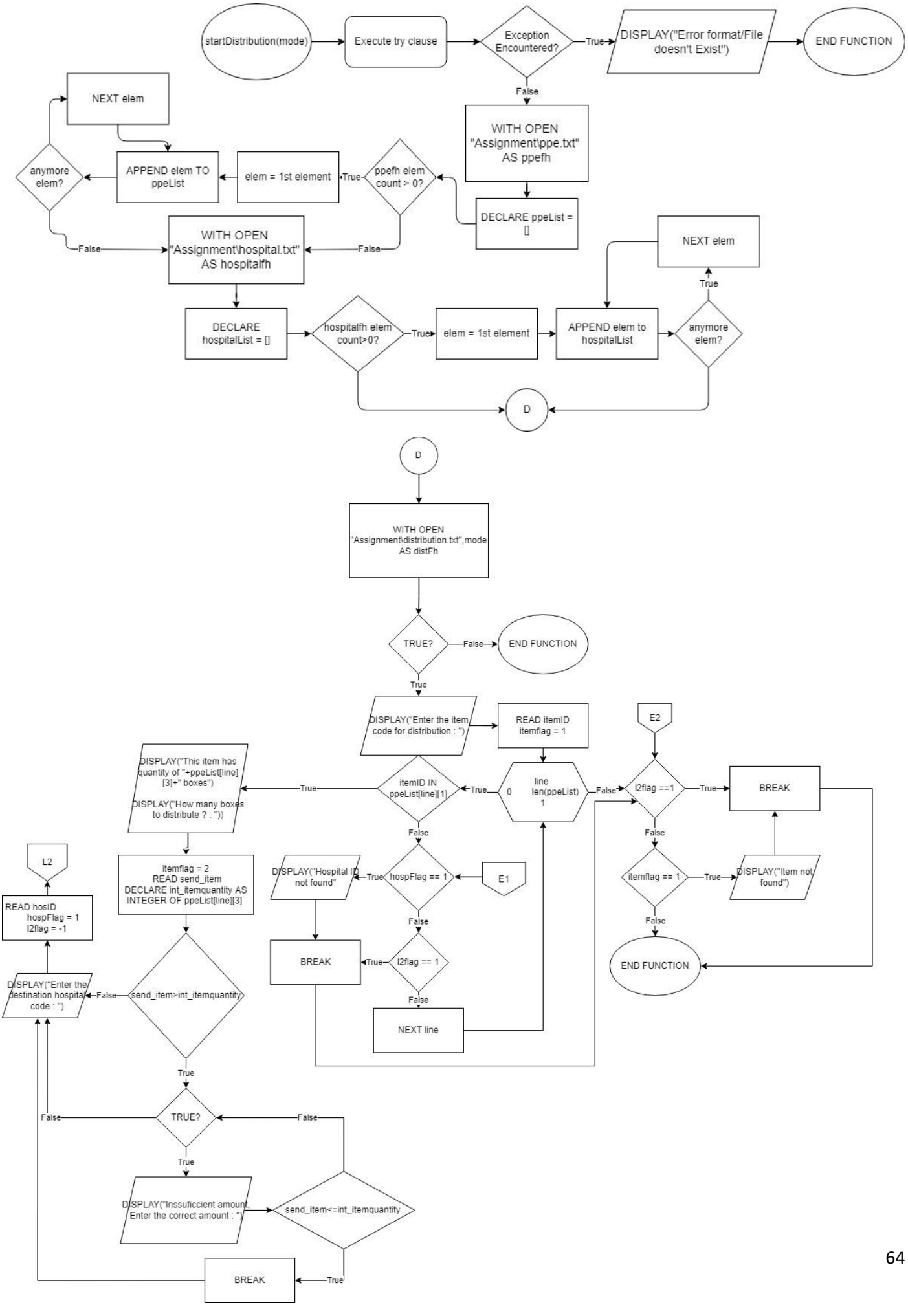
```
            ELSEIF option == 9 THEN
                flag = 3
                BREAK
            ENDIF
        ENDDO
    ELSEIF flag ==3 THEN
        BREAK
    ENDIF
  ENDDO
  DISPLAY("END OF PROGRAM")
 EXCEPT
  DISPLAY("Error Format/Data not found")
 ENDTRY
ENDDEFINE

BEGIN
 CALL ALL_MENU()
END
```

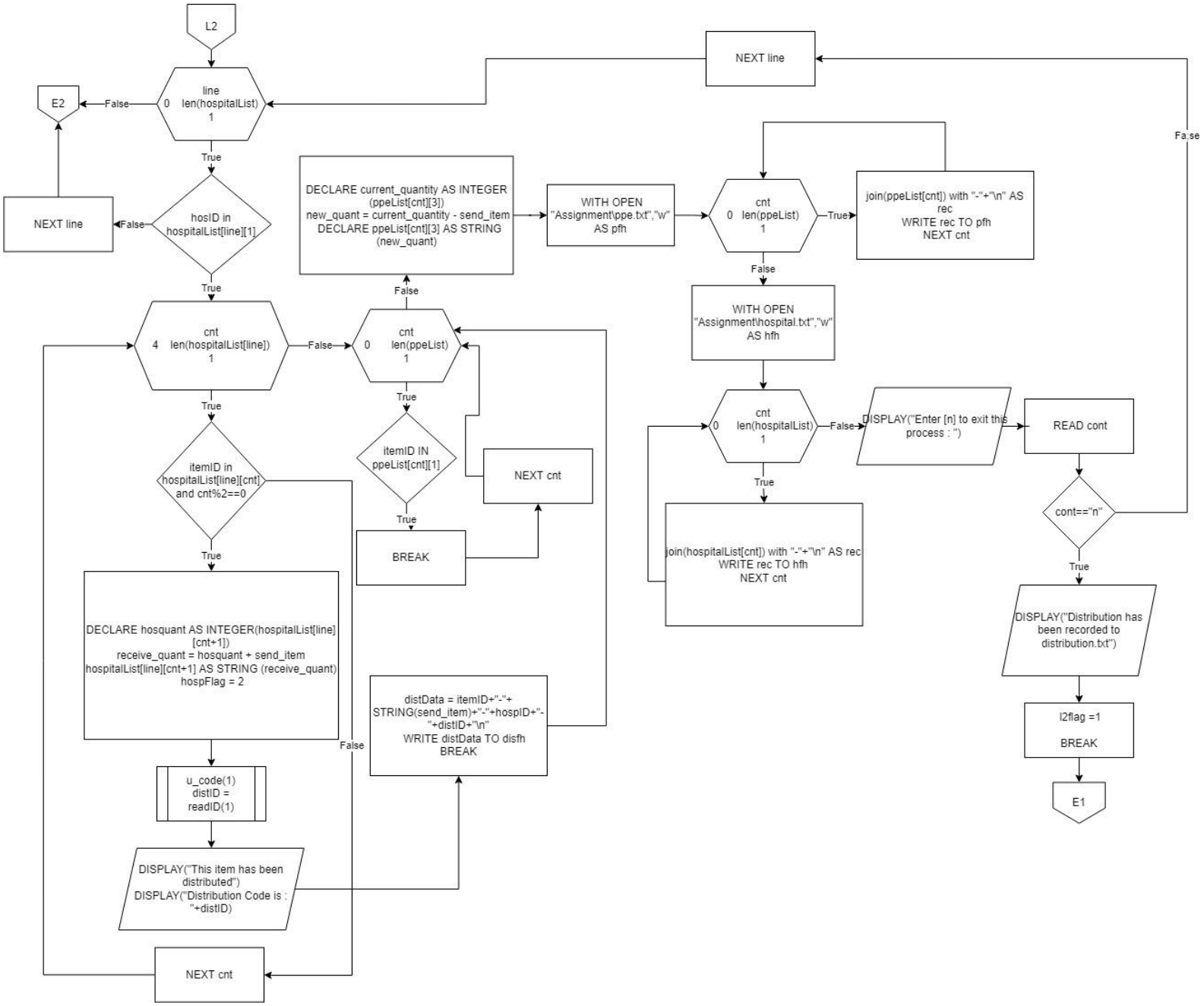
Flowchart

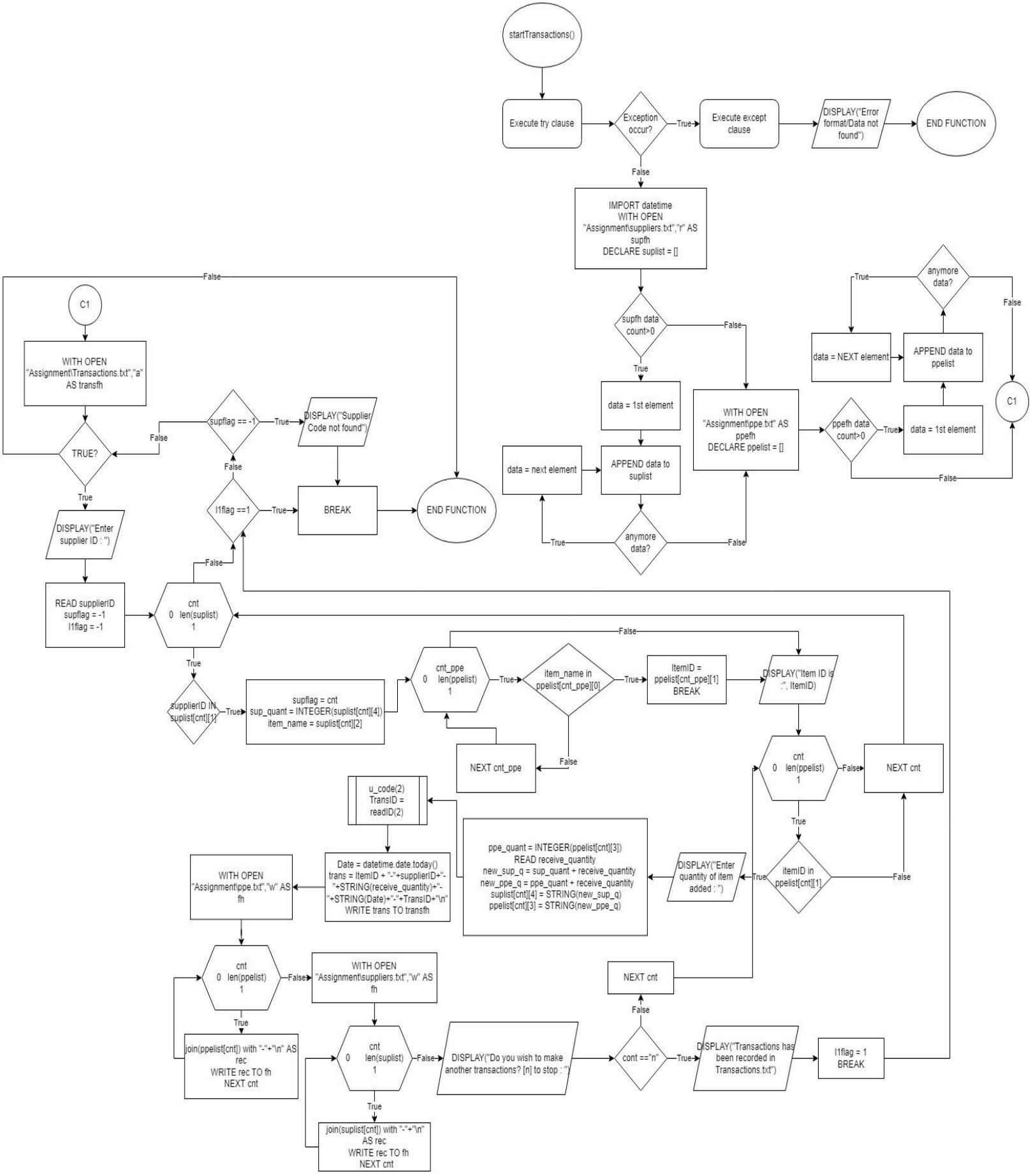

startPPEdata()
genAuthcode()
DISPLAY("Welcome to the initial inventory creation")
DISPLAY("Please fill the required information with the correct format")
Execute try clause
Exception Encountered
Yes
Execute except clause
DISPLAY("Error Format Detected, Please try again")
No
WITH OPEN ("Assignment\ppe.txt","w") AS ppeFile
TRUE?
No
RETURN 1
yes
DISPLAY("Please enter the PPE item name : ")
DISPLAY("Please enter the item code : ")
DISPLAY("Register / Enter Supplier Code : ")
READ itemname, itemcode, suppliercode
DECLARE quantity as INTEGER quantity = 100
DISPLAY("Please enter the year item inputted (YYYY) : ")
DISPLAY("Please enter the month item inputted (MM) : ")
READ YYYY,MM
MM = monthLimit(MM)
DISPLAY("Please enter the date item inputted (DD) : ")
READ DD
DD = dayLimit(YYYY, MM, DD)
ppeData = itemname+"-"+itemcode+"-"+suppliercode+"-"+STRING(quantity)+"-"+STRING(YYYY)+"-"+STRING(MM)+"-"+STRING(DD)+"\n"
WRITE ppeData into ppe.txt
DISPLAY("Press Enter key to continue or [q] to stop : ")
READ cont
cont.lower()=="q"
True
False
BREAK
DISPLAY("PPE data has been recorded to ppe.txt file")

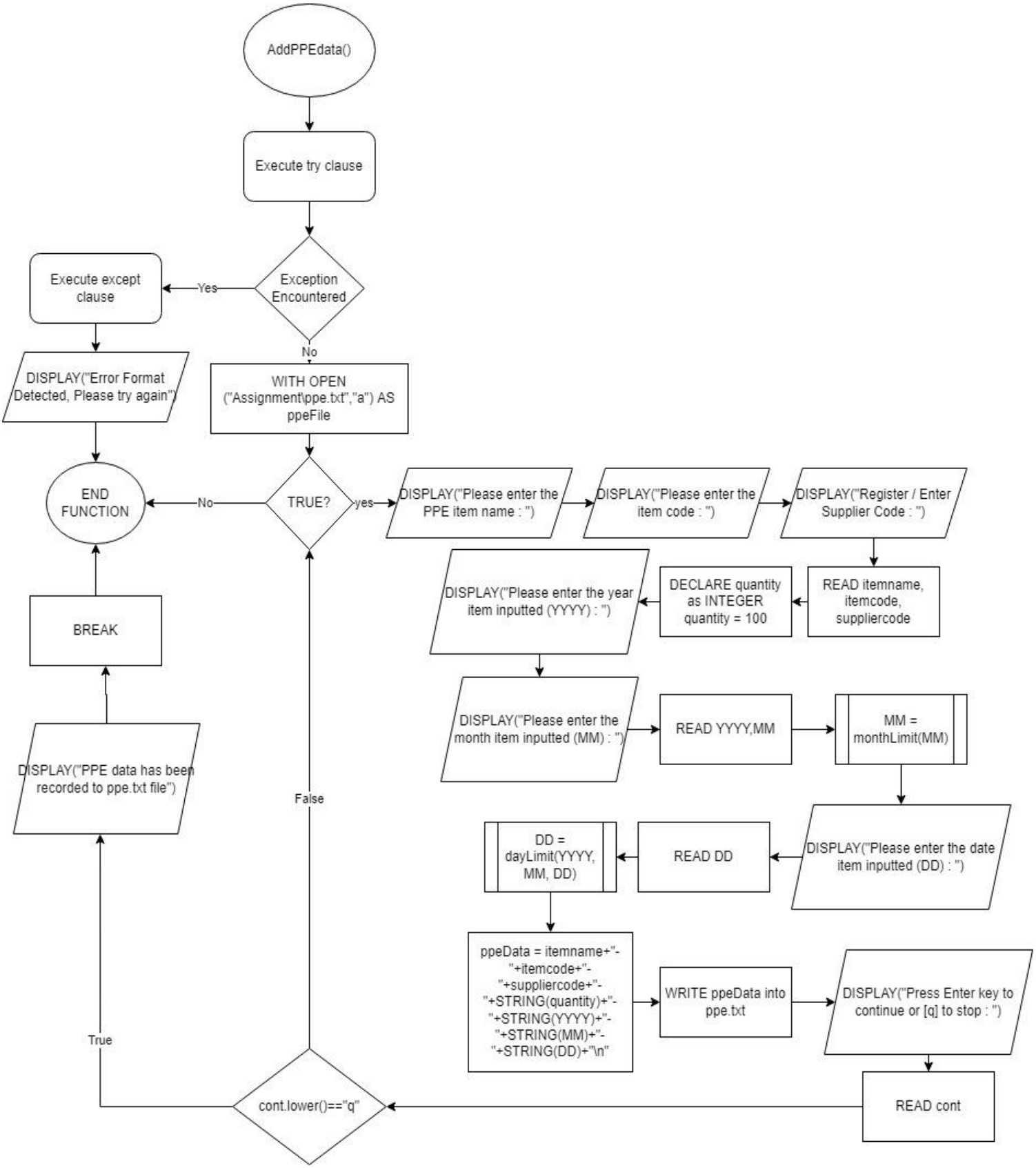

startSupplydata(mode,supnum)
Execute try clause
Exception Encountered
True
Execute except clause
DISPLAY("Error Format Detected, Please try again")
END FUNCTION
False
WITH OPEN ("Assignment\ppe.txt","w") AS ppeFile
DECLARE ppeList = []
readPPe data count>0?
True
False
data = 1st element
APPEND data to ppeList
anymore data in readPPe?
False
True
data = next element
A
DISPLAY("suppliers has been recorded in suppliers.txt")
E
A
WITH OPEN "Assignment\suppliers.txt",mode AS insupply
i
0
supnum
1
False
True
E
L
DECLARE checklist = []
DISPLAY("Please enter registered supplier code / [q] to stop : ")
READ supID
supID. lower() =="q"?
True
Break
E
False
WITH OPEN "Assignent\suppliers.txt","r" AS readfh
readfh data count>0?
False
True
data = 1st element
APPEND data to checklist
data = next element
B
anymore data in readPPe?
False
True
DECLARE checkf=-1
cnt
0
len(checklist)
1
True
False
checkf ==-1?
True
False
fl = -1
cnt
0
len(ppeList)
1
supID IN ppeList[cnt][2]?
True
False
NEXT cnt
supID in checklist[cnt][1]?
False
True
NEXT cnt
fl = cnt
DISPLAY("Item Supplied : "+ ppeList[cnt][0])
DISPLAY("Enter supplier name : ")
DISPLAY("Enter supplier address : ")
supItem = ppeList[cnt][0]
READ supName
READ supAddress
supQuant = ppeList[cnt][3]
suppliers = supName+"-"+supID+"-"+supItem+"-"+supAddress+"-"+STRING(supQuant)+"\n"
WRITE suppliers TO insupply
FLUSH insupply
fl==-1?
True
DISPLAY("Supplier code not found")
True
DISPLAY("This supplier code has been inputted")
checkf=cnt
break
NEXT i
L

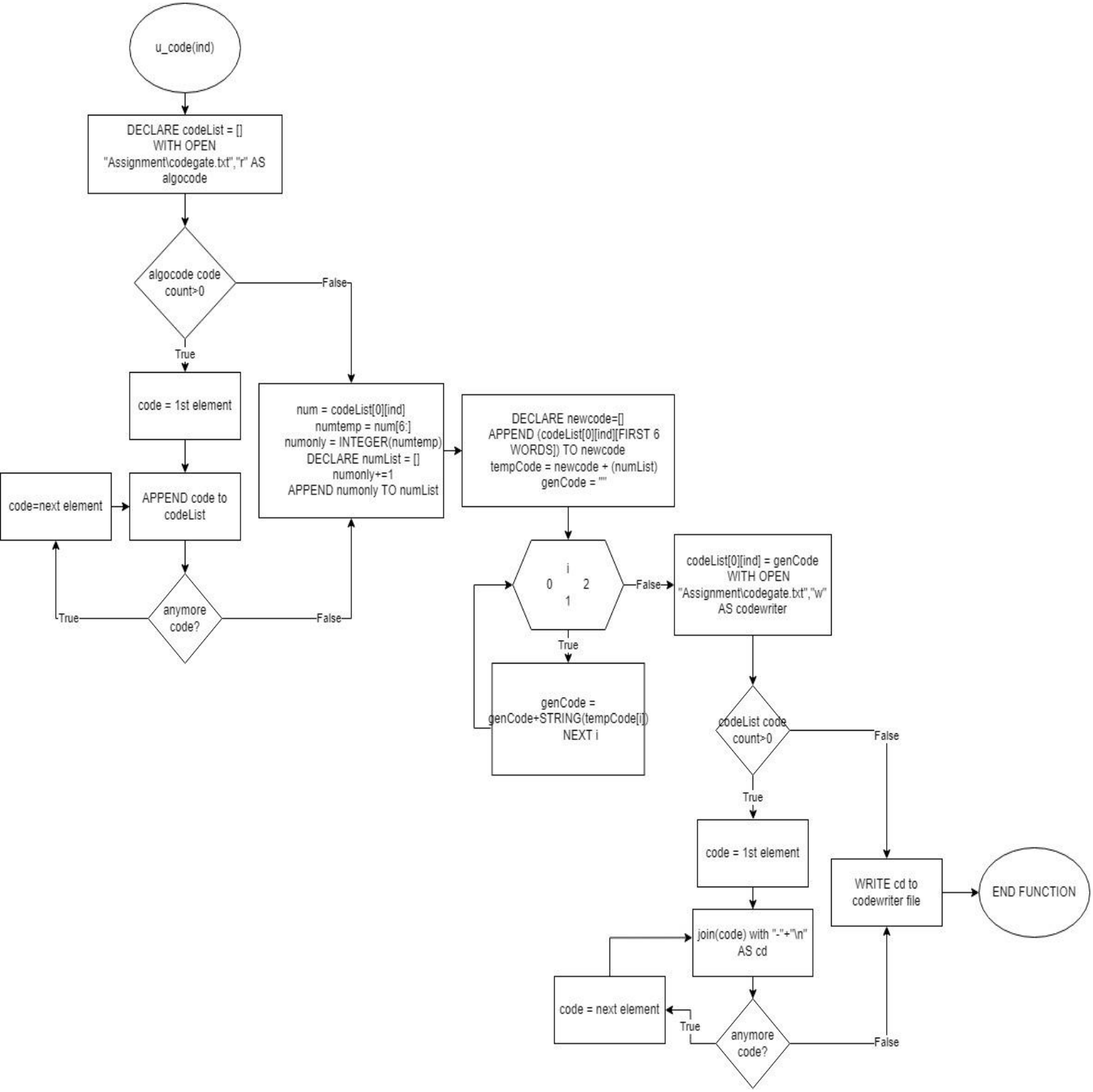

startHospital(mode,hosnum)
WITH OPEN "Assignment\ppe.txt","r" AS ppefh
DECLARE list = []
ppefh data count>0?
False
True
data = 1st element
NEXT data
APPEND data to list
Anymore data?
True
False
DECLARE IDlist = [ ]
list cnt count>0?
False
True
cnt = 1st element
APPEND cnt[1] to IDlist
NEXT cnt
Anymore cnt?
True
False
Execute try clause
Exception Encountered?
True
False
Except Clause
DISPLAY("Error Format Detected, Please try again")
END FUNCTION
T
X
T
WITH OPEN "Assignment\hospital.txt",mode AS fh
0 x hosnum 1
True
False
DISPLAY("Enter Hospital name or [q] to stop : ")
END FUNCTION
BREAK
True
READ hosName
hosName. == "q"?
False
u_code(0)
hosID = readID(0)
DISPLAY("Enter hospital address : ")
DISPLAY("Enter hospital contact number : ")
READ hosAdd
READ hosContact
hosData = hosName +"-"+ hosID+"-"+ hosAdd +"-"+ hosContact
0 i len(IDlist) 1
NEXT i
True
False
hosData = hosData +"-"+ IDlist[i]+"-"+"0"
DISPLAY("Registered Hospital Code is : "+ hosID)
hosData=hosData+"\n"
WRITE hosData to fh
DISPLAY("Hospitals has been recorded into hospital.txt")
NEXT x
X

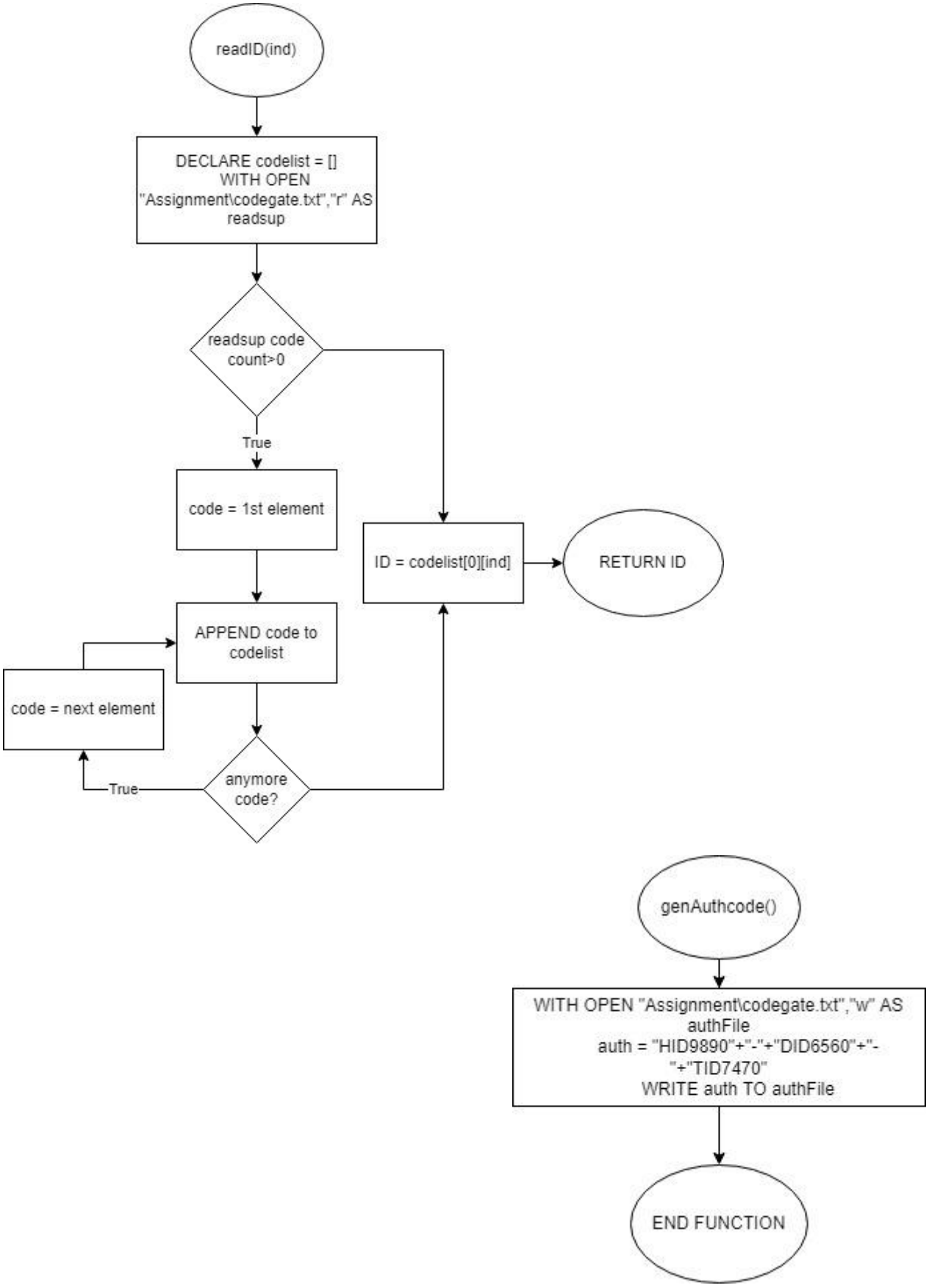

startDistribution(mode)
Execute try clause
Exception Encountered?
True
DISPLAY("Error format/File doesn't Exist")
END FUNCTION
False
WITH OPEN "Assignment\ppe.txt" AS ppefh
DECLARE ppeList = []
ppefh elem count > 0?
True
elem = 1st element
APPEND elem TO ppeList
anymore elem?
NEXT elem
False
WITH OPEN "Assignment\hospital.txt" AS hospitalfh
DECLARE hospitalList = []
hospitalfh elem count>0?
elem = 1st element
APPEND elem to hospitalList
anymore elem?
D
WITH OPEN "Assignment\distribution.txt",mode AS distFh
TRUE?
END FUNCTION
DISPLAY("Enter the item code for distribution : ")
READ itemID itemflag = 1
0 line len(ppeList) 1
itemID IN ppeList[line][1]
DISPLAY("This item has quantity of "+ppeList[line][3]+" boxes")
DISPLAY("How many boxes to distribute ? : "))
itemflag = 2
READ send_item
DECLARE int_itemquantity AS INTEGER OF ppeList[line][3]
send_item>int_itemquantity
DISPLAY("Enter the destination hospital code : ")
READ hosID
hospFlag = 1
l2flag = -1
L2
hospFlag == 1
DISPLAY("Hospital ID not found")
E1
l2flag == 1
BREAK
NEXT line
E2
l2flag ==1
itemflag == 1
DISPLAY("Item not found")
BREAK
END FUNCTION
TRUE?
DISPLAY("Inssuficcient amount. Enter the correct amount : ")
send_item<=int_itemquantity
BREAK

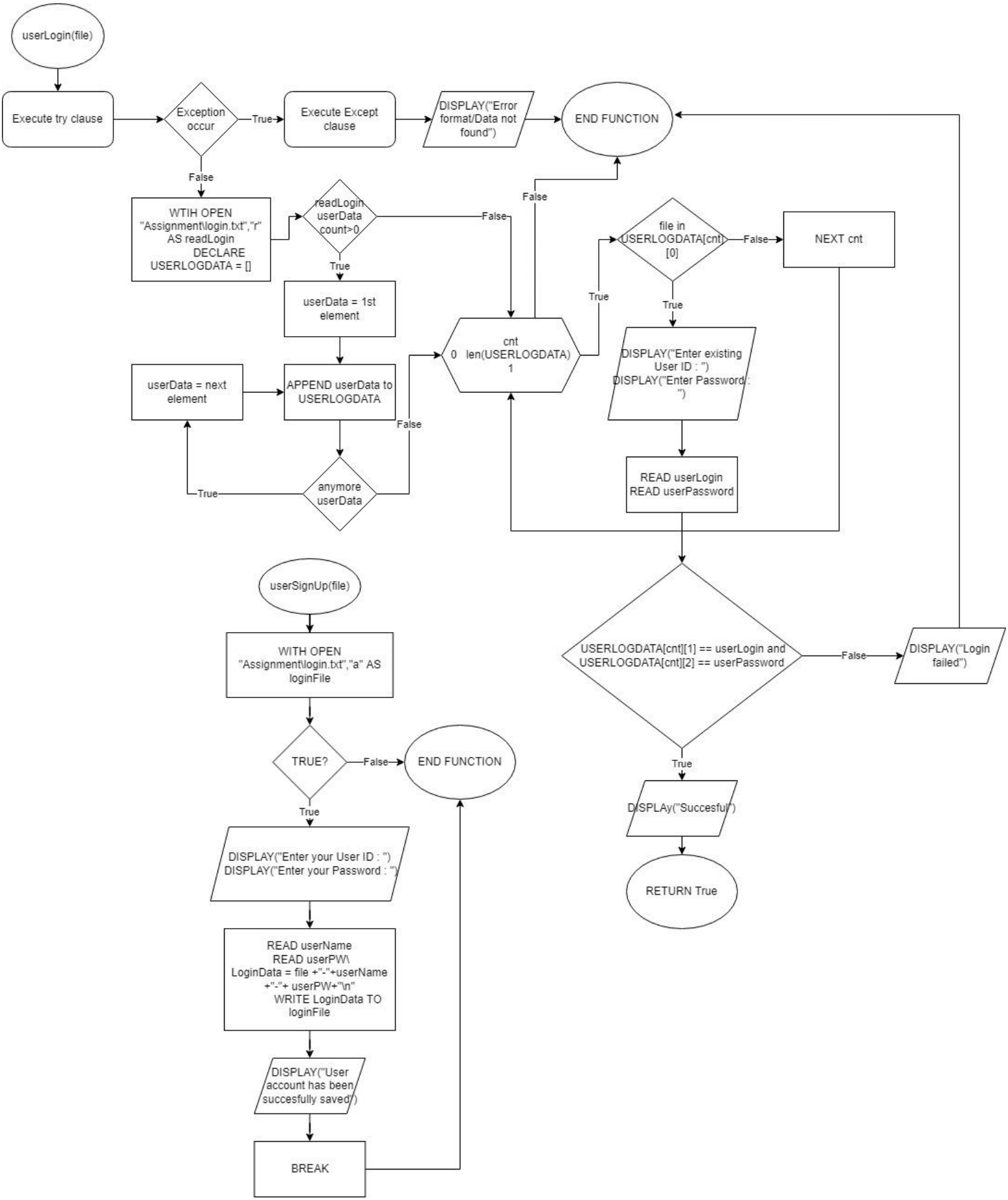

L2
line
0 len(hospitalList)
1
False
E2
True
hosID in hospitalList[line][1]
False
NEXT line
True
cnt
4 len(hospitalList[line])
1
False
True
itemID in hospitalList[line][cnt] and cnt%2==0
True
DECLARE hosquant AS INTEGER(hospitalList[line][cnt+1])
receive_quant = hosquant + send_item
hospitalList[line][cnt+1] AS STRING (receive_quant)
hospFlag = 2
u_code(1)
distID = readID(1)
DISPLAY("This item has been distributed")
DISPLAY("Distribution Code is : "+distID)
NEXT cnt
False
distData = itemID+"-"+ STRING(send_item)+"-"+hospID+"-"+distID+"\n"
WRITE distData TO disfh
BREAK
cnt
0 len(ppeList)
1
True
itemID IN ppeList[cnt][1]
True
BREAK
NEXT cnt
False
DECLARE current_quantity AS INTEGER (ppeList[cnt][3])
new_quant = current_quantity - send_item
DECLARE ppeList[cnt][3] AS STRING (new_quant)
WITH OPEN "Assignment\ppe.txt","w" AS pfh
cnt
0 len(ppeList)
1
True
join(ppeList[cnt]) with "-"+"\n" AS rec
WRITE rec TO pfh
NEXT cnt
False
WITH OPEN "Assignment\hospital.txt","w" AS hfh
cnt
0 len(hospitalList)
1
False
DISPLAY("Enter [n] to exit this process : ")
READ cont
cont=="n"
True
DISPLAY("Distribution has been recorded to distribution.txt")
l2flag =1
BREAK
E1
False
True
join(hospitalList[cnt]) with "-"+"\n" AS rec
WRITE rec TO hfh
NEXT cnt
NEXT line
False

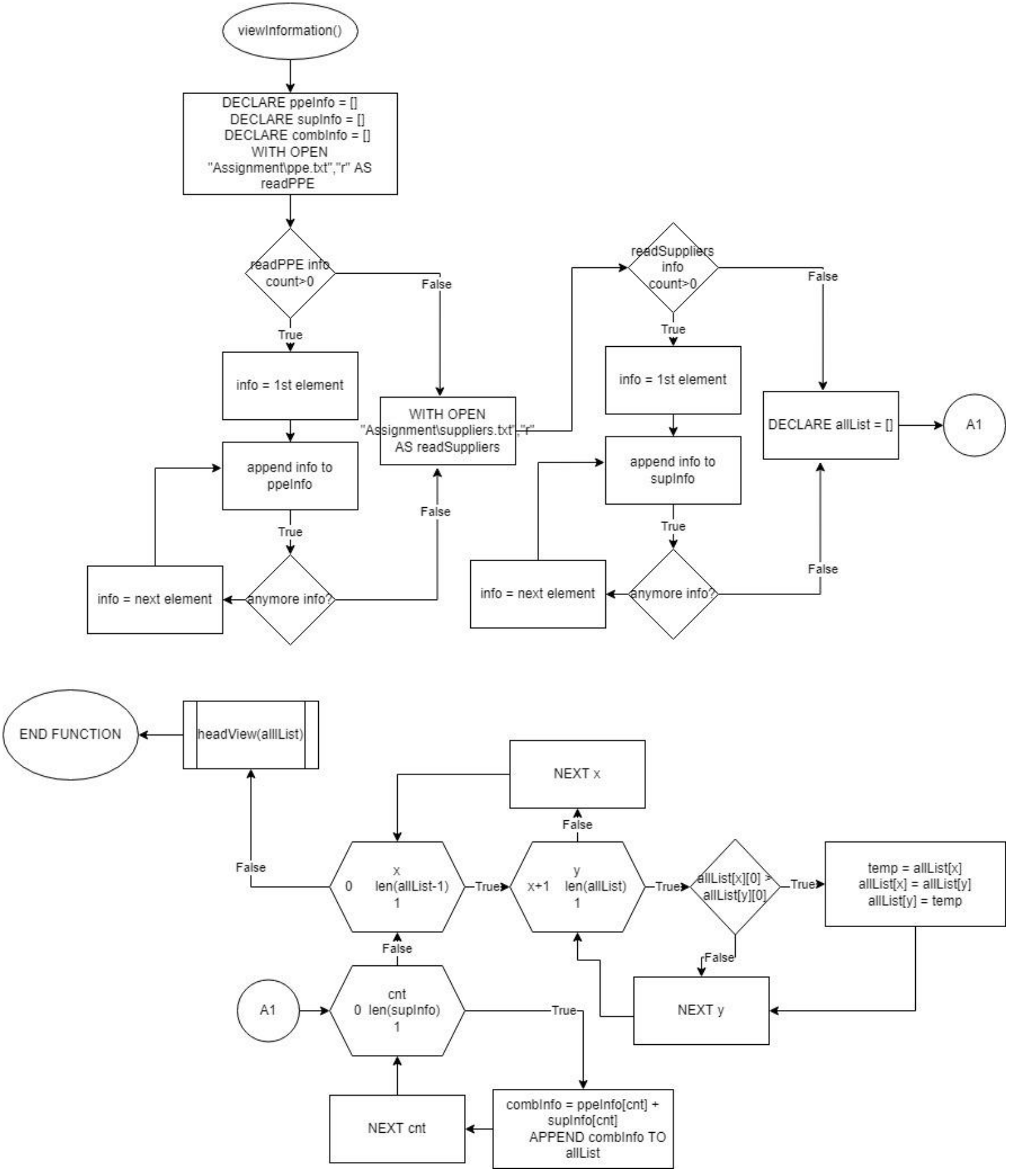

startTransactions()
Execute try clause
Exception occur?
True
Execute except clause
DISPLAY("Error format/Data not found")
END FUNCTION
False
IMPORT datetime
WITH OPEN "Assignment\suppliers.txt","r" AS supfh
DECLARE suplist = []
supfh data count>0
False
True
data = 1st element
APPEND data to suplist
data = next element
anymore data?
True
False
WITH OPEN "Assignment\ppe.txt" AS ppefh
DECLARE ppelist = []
ppefh data count>0
True
data = 1st element
APPEND data to ppelist
data = NEXT element
anymore data?
True
False
False
C1
WITH OPEN "Assignment\Transactions.txt","a" AS transfh
TRUE?
True
False
DISPLAY("Enter supplier ID : ")
READ supplierID
supflag = -1
l1flag = -1
supflag == -1
True
DISPLAY("Supplier Code not found")
False
l1flag ==1
True
BREAK
END FUNCTION
cnt
0 len(suplist)
1
True
False
supplierID in suplist[cnt][1]
True
supflag = cnt
sup_quant = INTEGER(suplist[cnt][4])
item_name = suplist[cnt][2]
cnt_ppe
0 len(ppelist)
1
True
False
item_name in ppelist[cnt_ppe][0]
True
ItemID = ppelist[cnt_ppe][1]
BREAK
DISPLAY("Item ID is :", ItemID)
NEXT cnt_ppe
False
cnt
0 len(ppelist)
1
False
NEXT cnt
True
itemID in ppelist[cnt][1]
True
False
DISPLAY("Enter quantity of item added : ")
ppe_quant = INTEGER(ppelist[cnt][3])
READ receive_quantity
new_sup_q = sup_quant + receive_quantity
new_ppe_q = ppe_quant + receive_quantity
suplist[cnt][4] = STRING(new_sup_q)
ppelist[cnt][3] = STRING(new_ppe_q)
u_code(2)
TransID = readID(2)
Date = datetime.date.today()
trans = ItemID + "-"+supplierID+"-"+STRING(receive_quantity)+"-"+STRING(Date)+"-"+TransID+"\n"
WRITE trans TO transfh
WITH OPEN "Assignment\ppe.txt","w" AS fh
cnt
0 len(ppelist)
1
False
True
join(ppelist[cnt]) with "-"+"\n" AS rec
WRITE rec TO fh
NEXT cnt
WITH OPEN "Assignment\suppliers.txt","w" AS fh
cnt
0 len(suplist)
1
False
True
join(suplist[cnt]) with "-"+"\n" AS rec
WRITE rec TO fh
NEXT cnt
DISPLAY("Do you wish to make another transactions? [n] to stop : ")
cont =="n"
True
False
NEXT cnt
DISPLAY("Transactions has been recorded in Transactions.txt")
l1flag = 1
BREAK

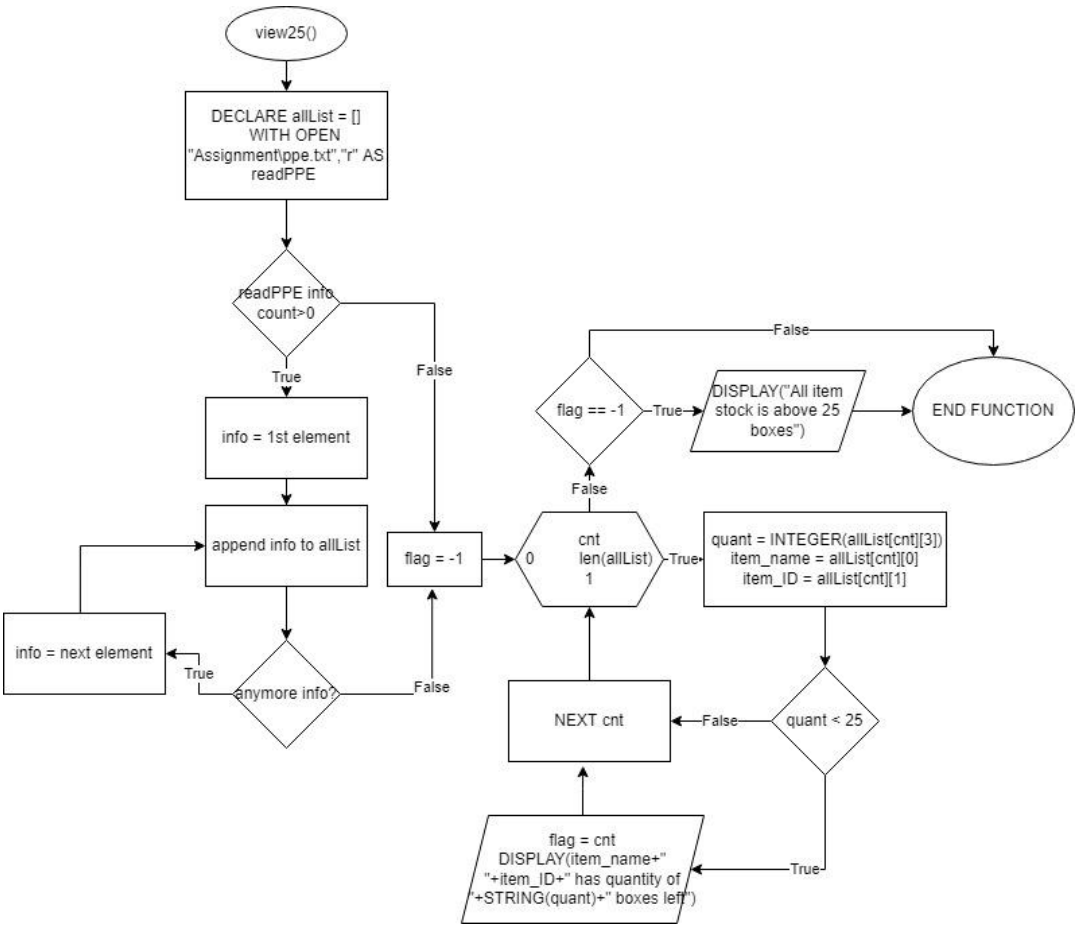

AddPPEdata()
Execute try clause
Exception Encountered
Yes
Execute except clause
DISPLAY("Error Format Detected, Please try again")
No
WITH OPEN ("Assignment\ppe.txt","a") AS ppeFile
TRUE?
No
END FUNCTION
yes
DISPLAY("Please enter the PPE item name : ")
DISPLAY("Please enter the item code : ")
DISPLAY("Register / Enter Supplier Code : ")
READ itemname, itemcode, suppliercode
DECLARE quantity as INTEGER quantity = 100
DISPLAY("Please enter the year item inputted (YYYY) : ")
DISPLAY("Please enter the month item inputted (MM) : ")
READ YYYY,MM
MM = monthLimit(MM)
DISPLAY("Please enter the date item inputted (DD) : ")
READ DD
DD = dayLimit(YYYY, MM, DD)
ppeData = itemname+"-"+itemcode+"-"+suppliercode+"-"+STRING(quantity)+"-"+STRING(YYYY)+"-"+STRING(MM)+"-"+STRING(DD)+"\n"
WRITE ppeData into ppe.txt
DISPLAY("Press Enter key to continue or [q] to stop : ")
READ cont
cont.lower()=="q"
True
False
BREAK
DISPLAY("PPE data has been recorded to ppe.txt file")

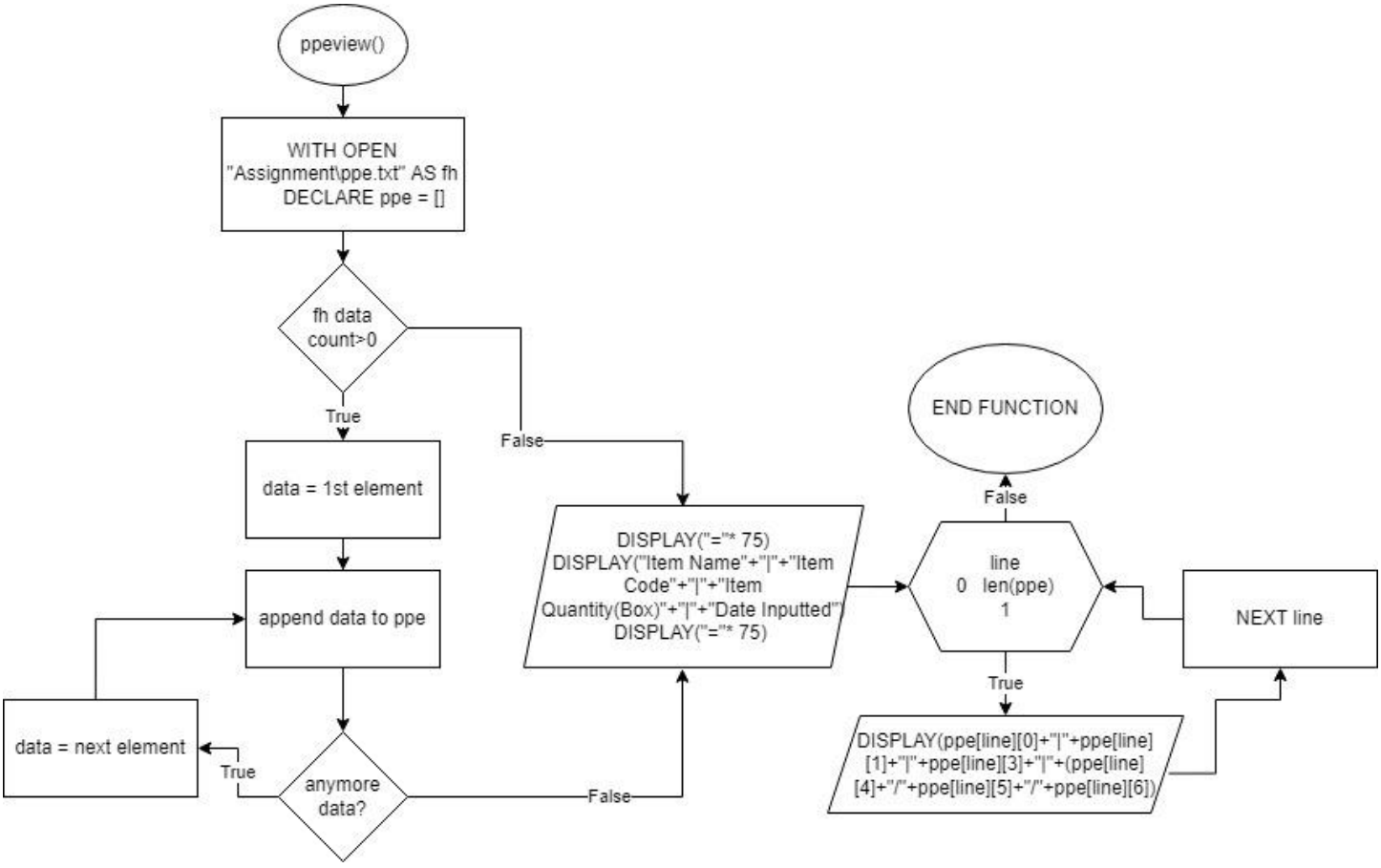

u_code(ind)
DECLARE codeList = []
WITH OPEN
"Assignment\codegate.txt","r" AS
algocode
algocode code
count>0
False
True
code = 1st element
APPEND code to
codeList
code=next element
anymore
code?
True
False
num = codeList[0][ind]
numtemp = num[6:]
numonly = INTEGER(numtemp)
DECLARE numList = []
numonly+=1
APPEND numonly TO numList
DECLARE newcode=[]
APPEND (codeList[0][ind][FIRST 6
WORDS]) TO newcode
tempCode = newcode + (numList)
genCode = ""
i
0 2
1
False
True
genCode =
genCode+STRING(tempCode[i])
NEXT i
codeList[0][ind] = genCode
WITH OPEN
"Assignment\codegate.txt","w"
AS codewriter
codeList code
count>0
False
True
code = 1st element
join(code) with "-"+"\n"
AS cd
code = next element
True
anymore
code?
False
WRITE cd to
codewriter file
END FUNCTION

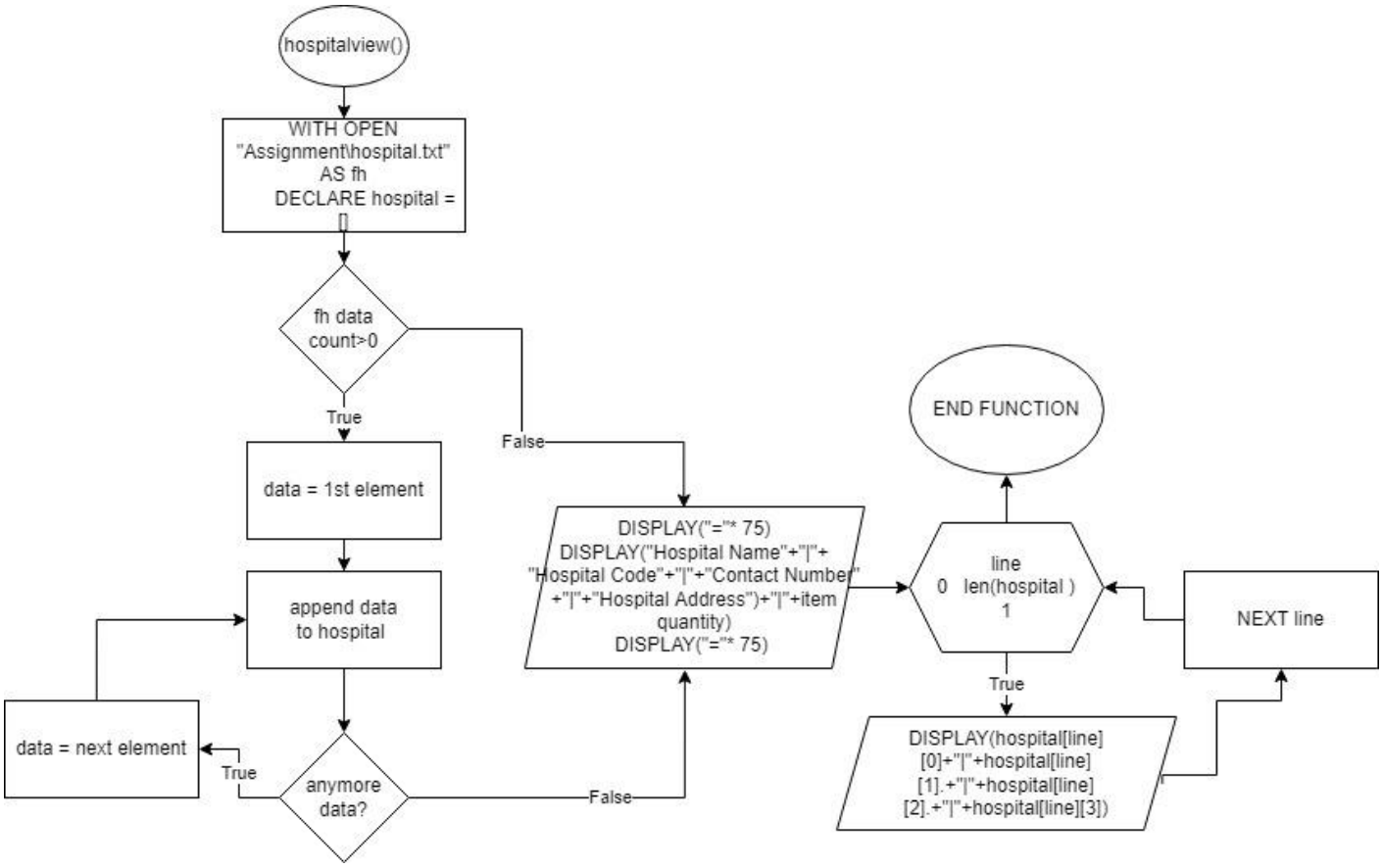

readID(ind)
DECLARE codelist = []
WITH OPEN "Assignment\codegate.txt","r" AS readsup
readsup code count>0
True
code = 1st element
APPEND code to codelist
code = next element
anymore code?
True
ID = codelist[0][ind]
RETURN ID
genAuthcode()
WITH OPEN "Assignment\codegate.txt","w" AS authFile
auth = "HID9890"+"-"+"DID6560"+"-"+"TID7470"
WRITE auth TO authFile
END FUNCTION

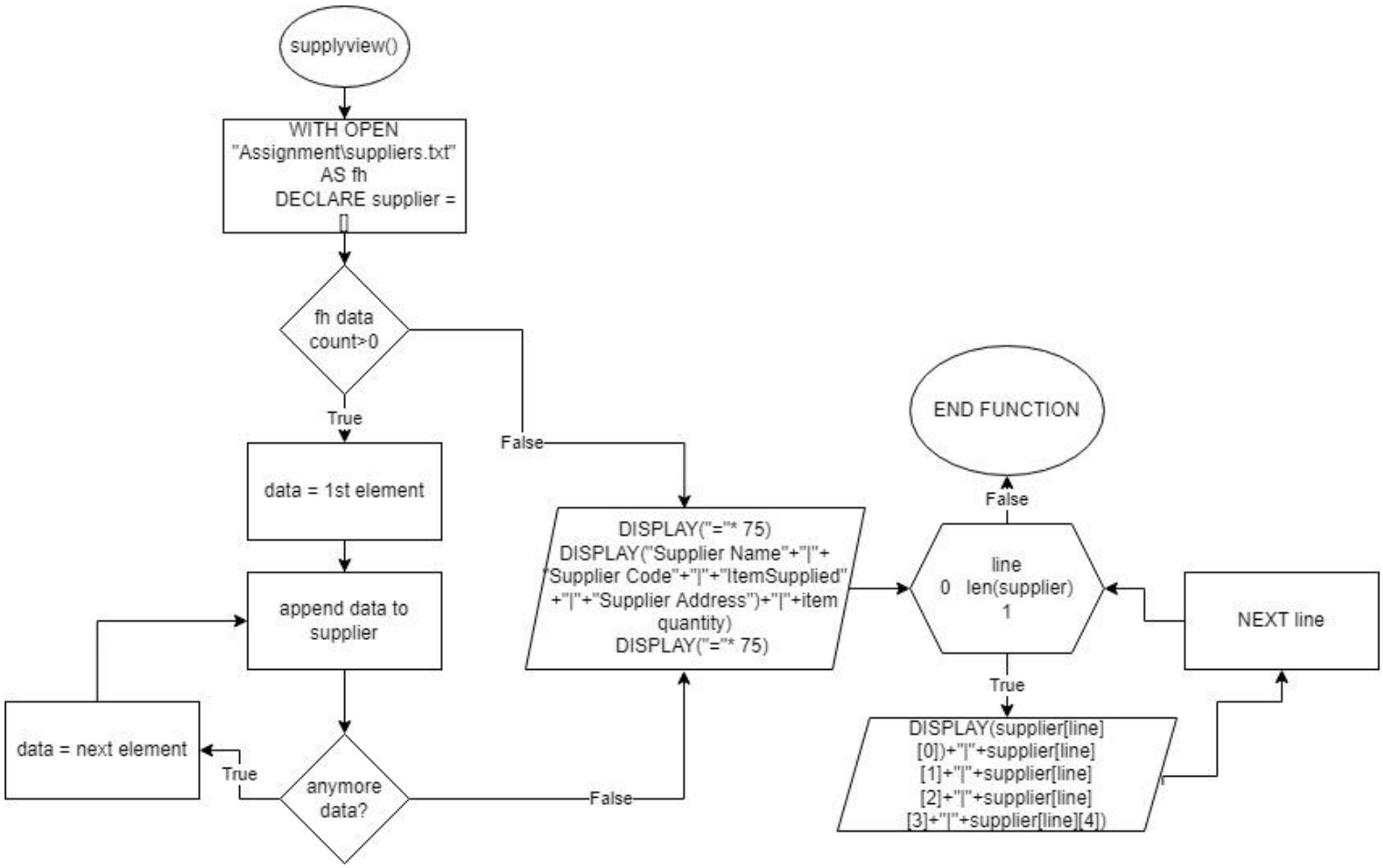

userLogin(file)
Execute try clause
Exception occur
True
Execute Except clause
DISPLAY("Error format/Data not found")
END FUNCTION
False
WTIH OPEN "Assignment\login.txt","r" AS readLogin
DECLARE USERLOGDATA = []
readLogin userData count>0
False
True
userData = 1st element
userData = next element
APPEND userData to USERLOGDATA
anymore userData
True
False
cnt
0 len(USERLOGDATA) 1
False
True
file in USERLOGDATA[cnt] [0]
False
NEXT cnt
True
DISPLAY("Enter existing User ID : ")
DISPLAY("Enter Password : ")
READ userLogin
READ userPassword
USERLOGDATA[cnt][1] == userLogin and USERLOGDATA[cnt][2] == userPassword
False
DISPLAY("Login failed")
True
DISPLAy("Succesful")
RETURN True
userSignUp(file)
WITH OPEN "Assignment\login.txt","a" AS loginFile
TRUE?
False
END FUNCTION
True
DISPLAY("Enter your User ID : ")
DISPLAY("Enter your Password : ")
READ userName
READ userPW\
LoginData = file +"-"+userName +"-"+ userPW+"\n"
WRITE LoginData TO loginFile
DISPLAY("User account has been succesfully saved")
BREAK

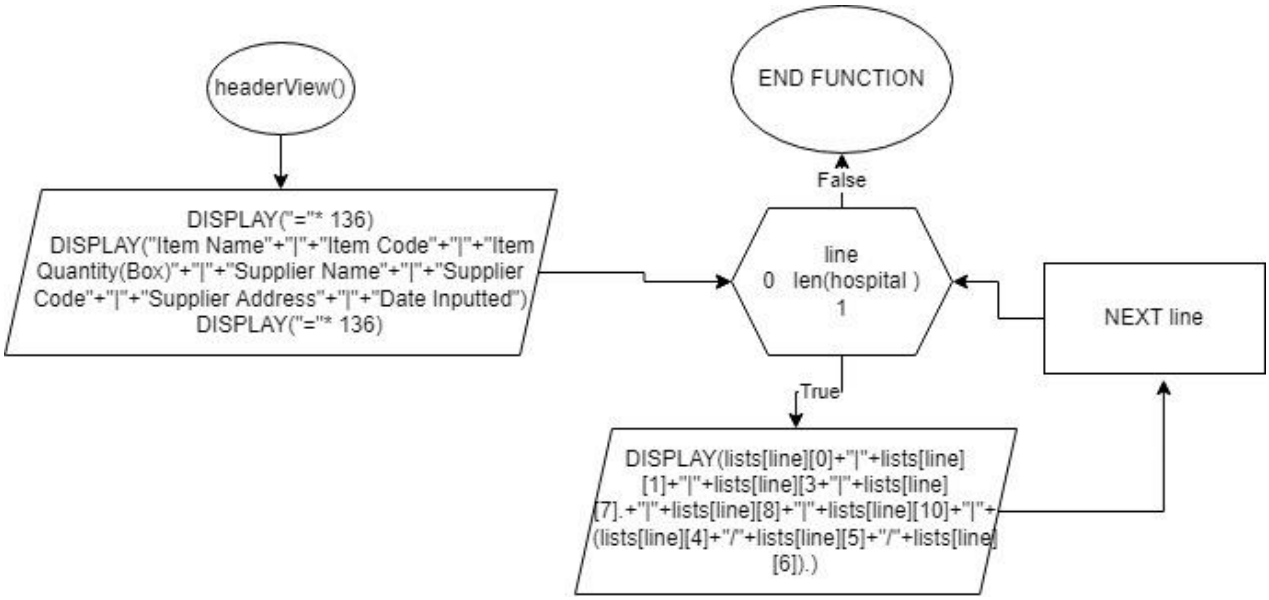

viewInformation()
DECLARE ppeInfo = []
DECLARE supInfo = []
DECLARE combInfo = []
WITH OPEN "Assignment\ppe.txt","r" AS readPPE
readPPE info count>0
True
False
info = 1st element
append info to ppeInfo
anymore info?
True
False
info = next element
WITH OPEN "Assignment\suppliers.txt","r" AS readSuppliers
readSuppliers info count>0
True
False
info = 1st element
append info to supInfo
anymore info?
True
False
info = next element
DECLARE allList = []
A1
END FUNCTION
headView(allList)
NEXT x
False
x
0 len(allList-1) 1
True
y
x+1 len(allList) 1
True
allList[x][0] > allList[y][0]
True
temp = allList[x]
allList[x] = allList[y]
allList[y] = temp
False
NEXT y
False
A1
cnt
0 len(supInfo) 1
True
False
combInfo = ppeInfo[cnt] + supInfo[cnt]
APPEND combInfo TO allList
NEXT cnt

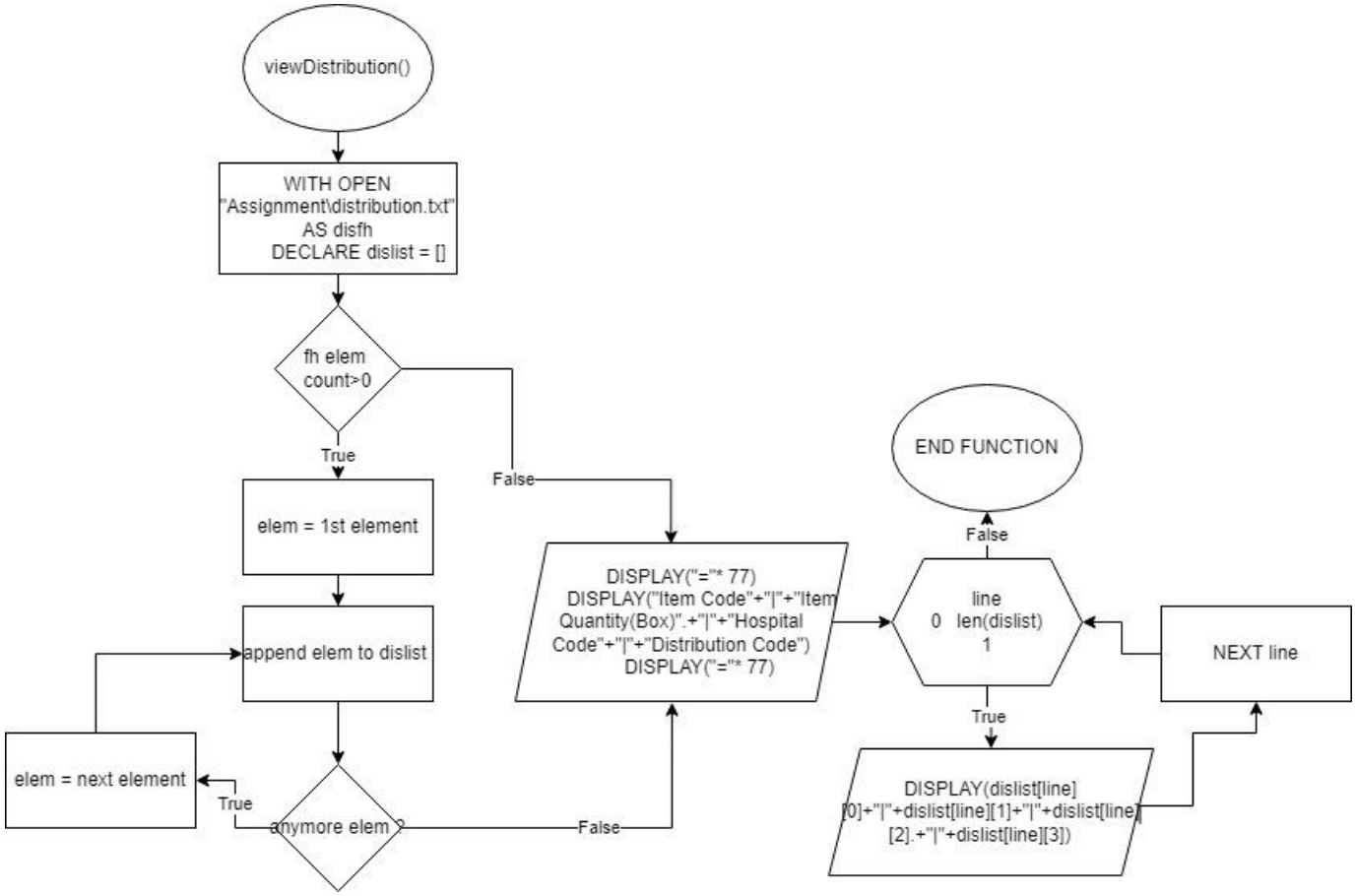

view25()
DECLARE allList = []
WITH OPEN "Assignment\ppe.txt","r" AS readPPE
readPPE info count>0
True
False
info = 1st element
append info to allList
info = next element
anymore info?
True
False
flag = -1
0 cnt len(allList) 1
True
False
flag == -1
True
False
DISPLAY("All item stock is above 25 boxes")
END FUNCTION
quant = INTEGER(allList[cnt][3])
item_name = allList[cnt][0]
item_ID = allList[cnt][1]
quant < 25
False
NEXT cnt
True
flag = cnt
DISPLAY(item_name+" "+item_ID+" has quantity of "+STRING(quant)+" boxes left")

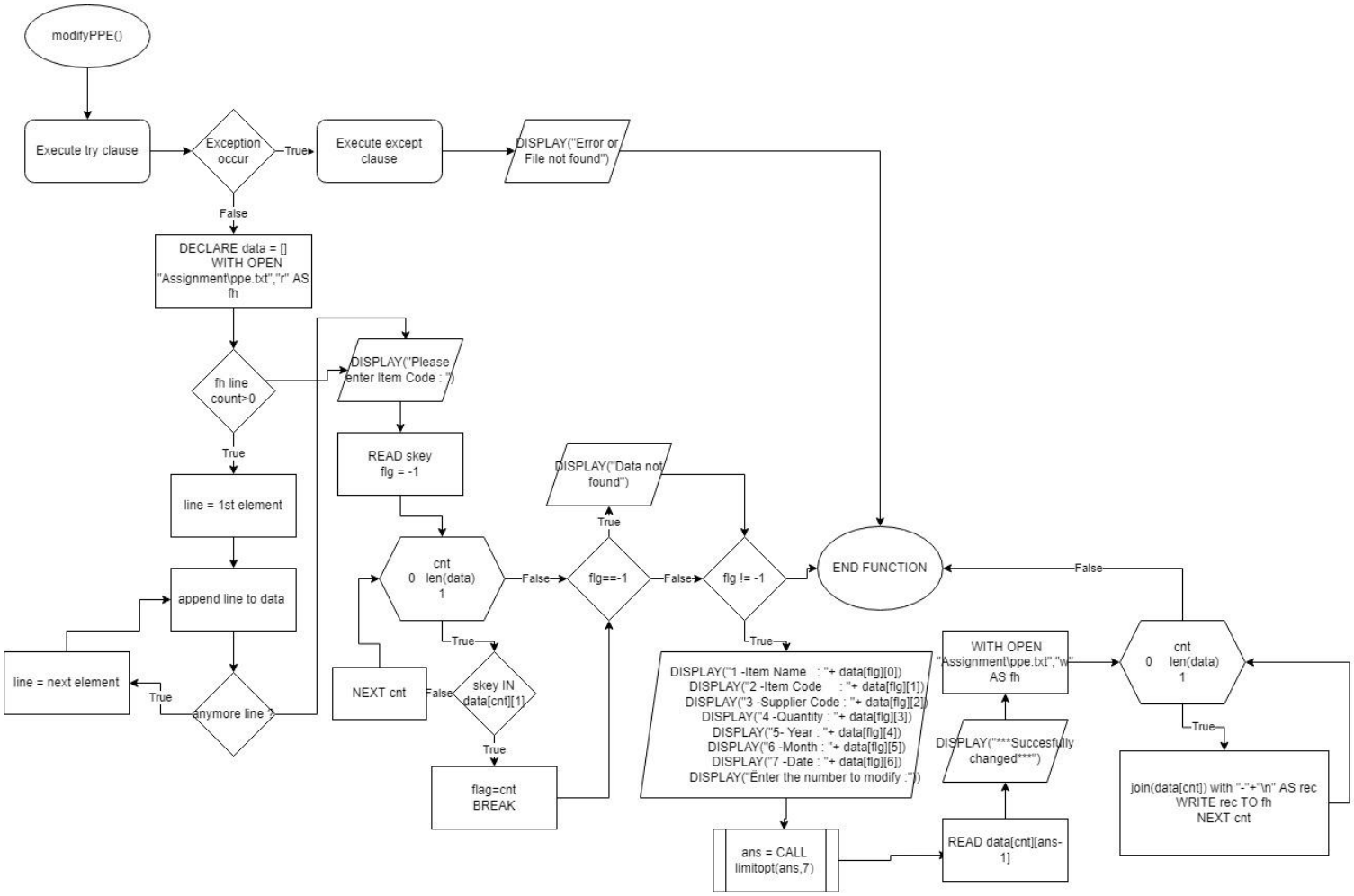

ppeview()
WITH OPEN "Assignment\ppe.txt" AS fh
DECLARE ppe = []
fh data count>0
True
False
data = 1st element
append data to ppe
data = next element
anymore data?
True
False
DISPLAY("="* 75)
DISPLAY("Item Name"+"|"+"Item Code"+"|"+"Item Quantity(Box)"+"|"+"Date Inputted")
DISPLAY("="* 75)
line 0 len(ppe) 1
False
True
END FUNCTION
NEXT line
DISPLAY(ppe[line][0]+"|"+ppe[line][1]+"|"+ppe[line][3]+"|"+(ppe[line][4]+"/"+ppe[line][5]+"/"+ppe[line][6])

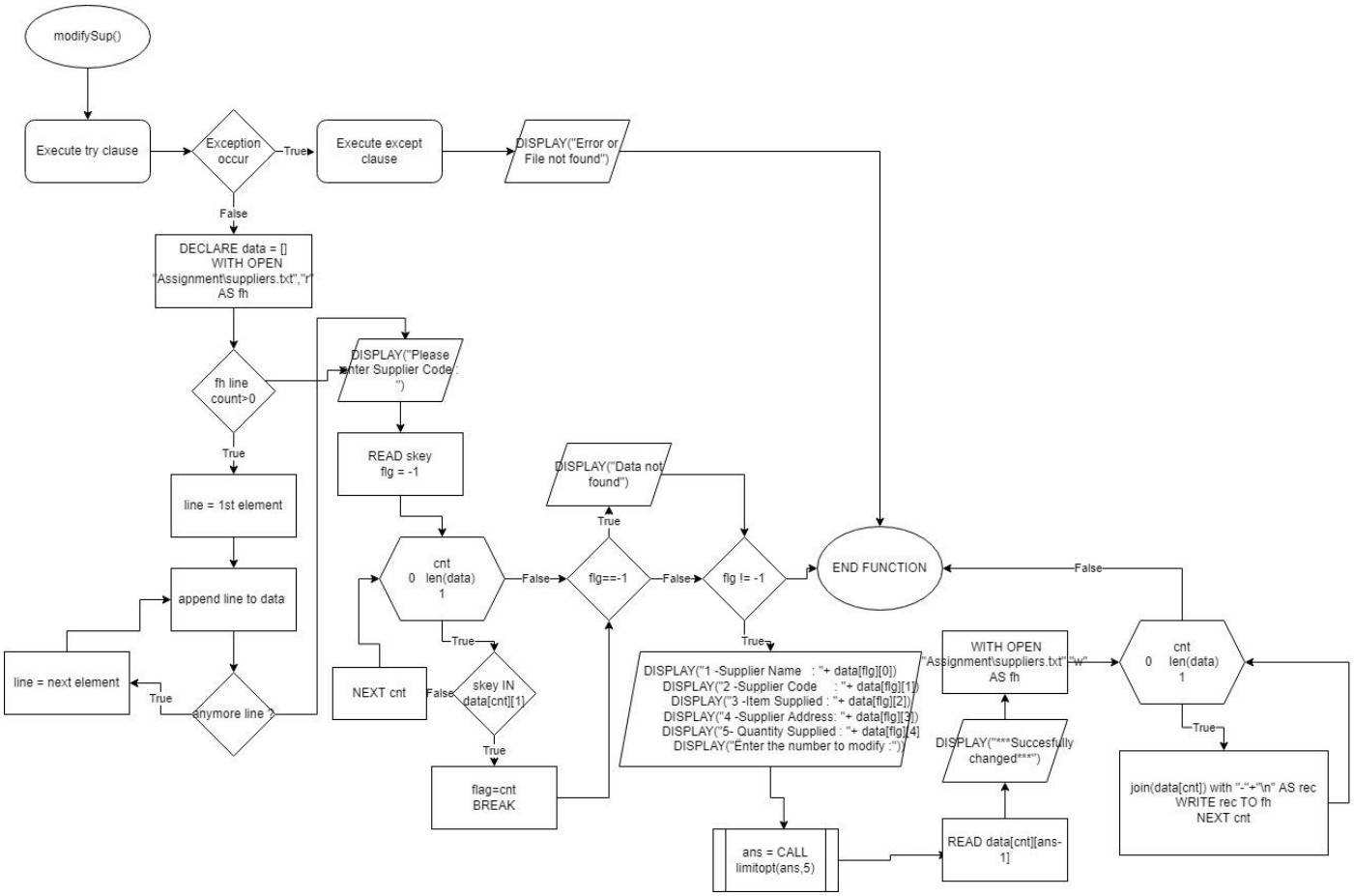

hospitalview()
WITH OPEN "Assignment\hospital.txt" AS fh
DECLARE hospital = []
fh data count>0
True
False
data = 1st element
append data to hospital
data = next element
True
anymore data?
False
DISPLAY("="* 75)
DISPLAY("Hospital Name"+"|"+ "Hospital Code"+"|"+"Contact Number" +"|"+"Hospital Address")+"|"+item quantity)
DISPLAY("="* 75)
END FUNCTION
line
0 len(hospital) 1
NEXT line
True
DISPLAY(hospital[line] [0]+"|"+hospital[line] [1].+"|"+hospital[line] [2].+"|"+hospital[line][3])

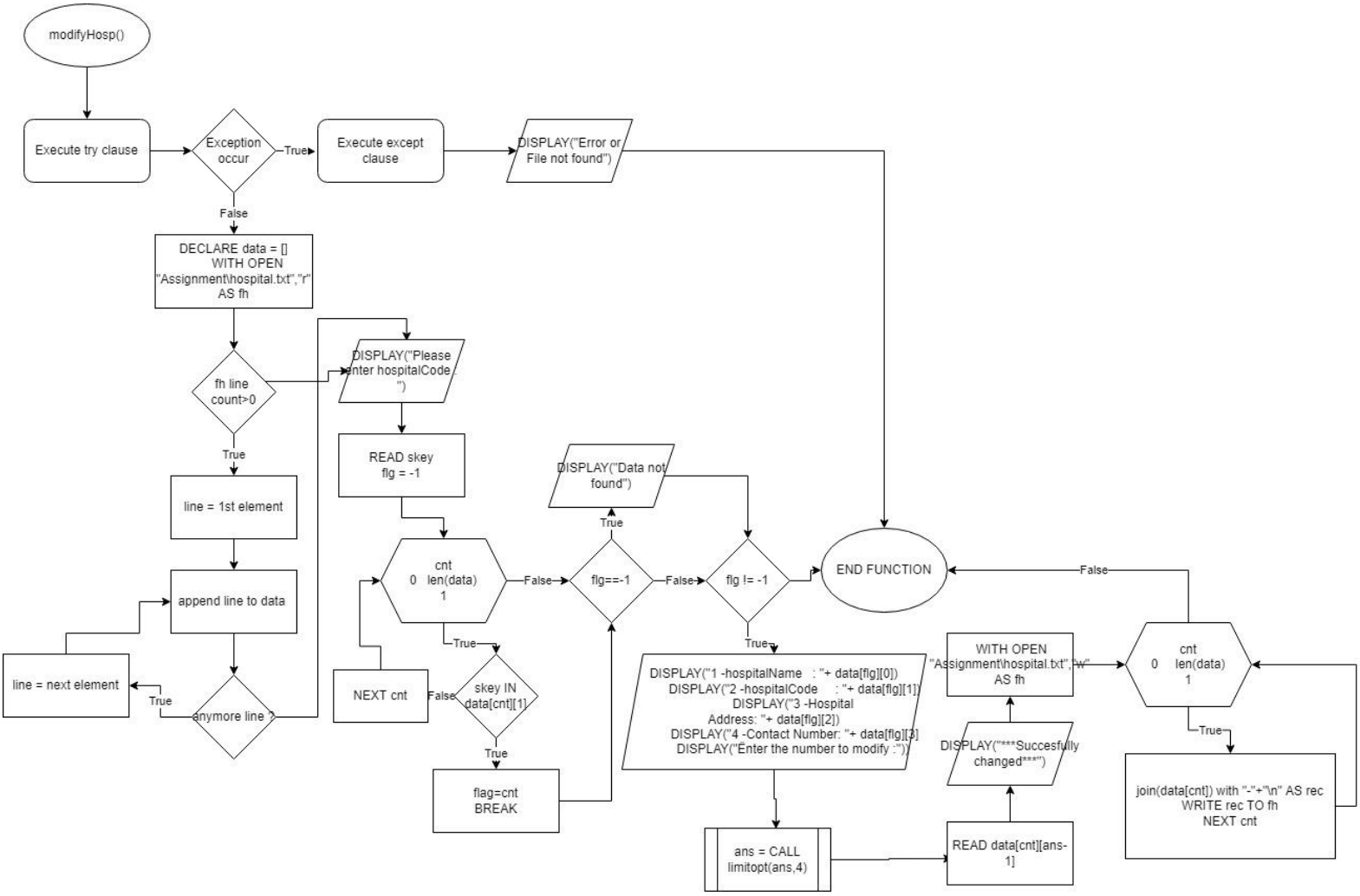

supplyview()
WITH OPEN "Assignment\suppliers.txt" AS fh
DECLARE supplier = []
fh data count>0
True
False
data = 1st element
append data to supplier
data = next element
True
anymore data?
False
DISPLAY("="* 75)
DISPLAY("Supplier Name"+"|"+ "Supplier Code"+"|"+"ItemSupplied" +"|"+"Supplier Address")+"|"+item quantity)
DISPLAY("="* 75)
END FUNCTION
False
line
0 len(supplier) 1
NEXT line
True
DISPLAY(supplier[line] [0])+"|"+supplier[line] [1]+"|"+supplier[line] [2]+"|"+supplier[line] [3]+"|"+supplier[line][4])

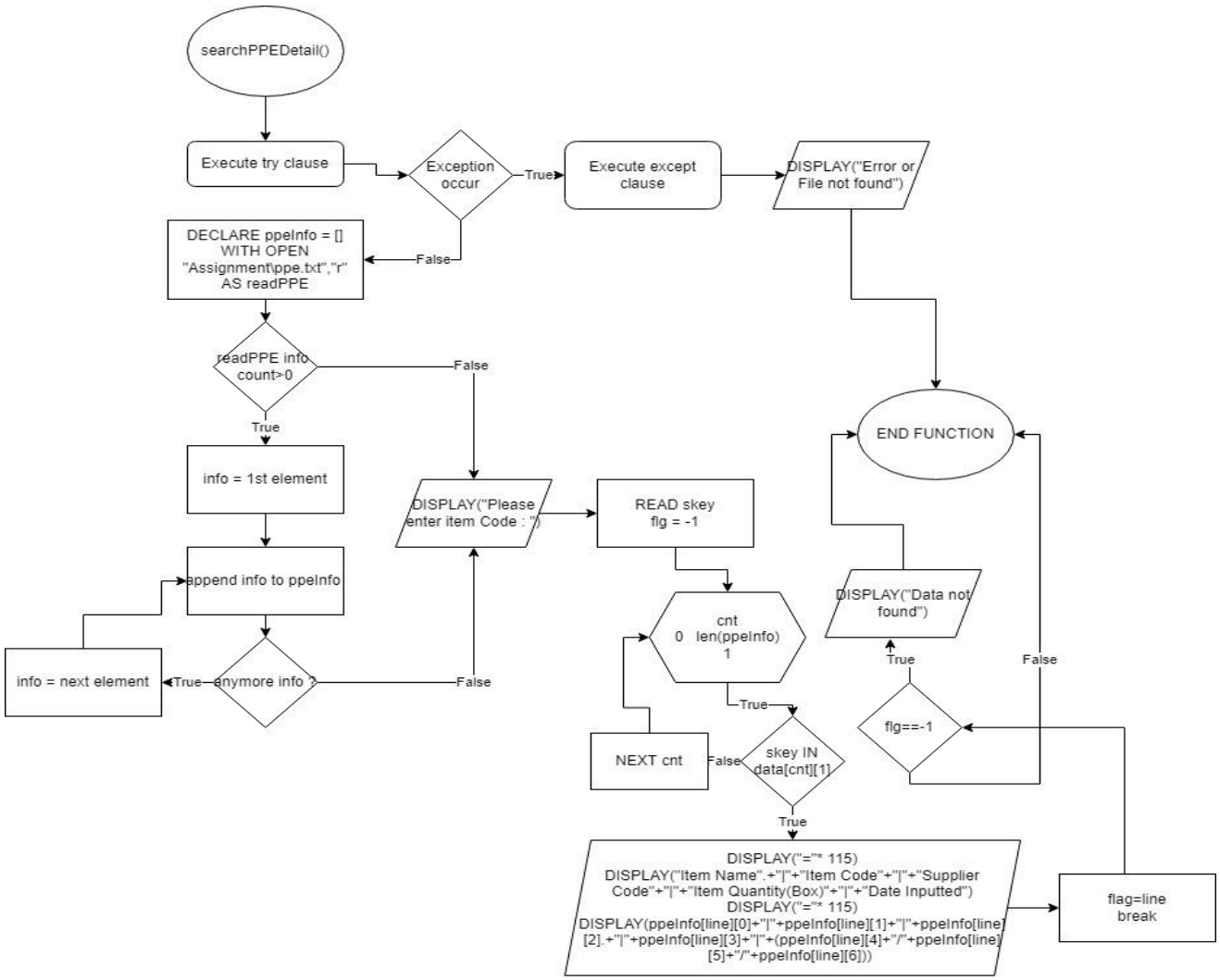

headerView()
DISPLAY("="* 136)
DISPLAY("Item Name"+"|"+"Item Code"+"|"+"Item Quantity(Box)"+"|"+"Supplier Name"+"|"+"Supplier Code"+"|"+"Supplier Address"+"|"+"Date Inputted")
DISPLAY("="* 136)
line
0 len(hospital)
1
False
END FUNCTION
True
NEXT line
DISPLAY(lists[line][0]+"|"+lists[line][1]+"|"+lists[line][3+"|"+lists[line][7].+"|"+lists[line][8]+"|"+lists[line][10]+"|"+(lists[line][4]+"/"+lists[line][5]+"/"+lists[line][6]).)

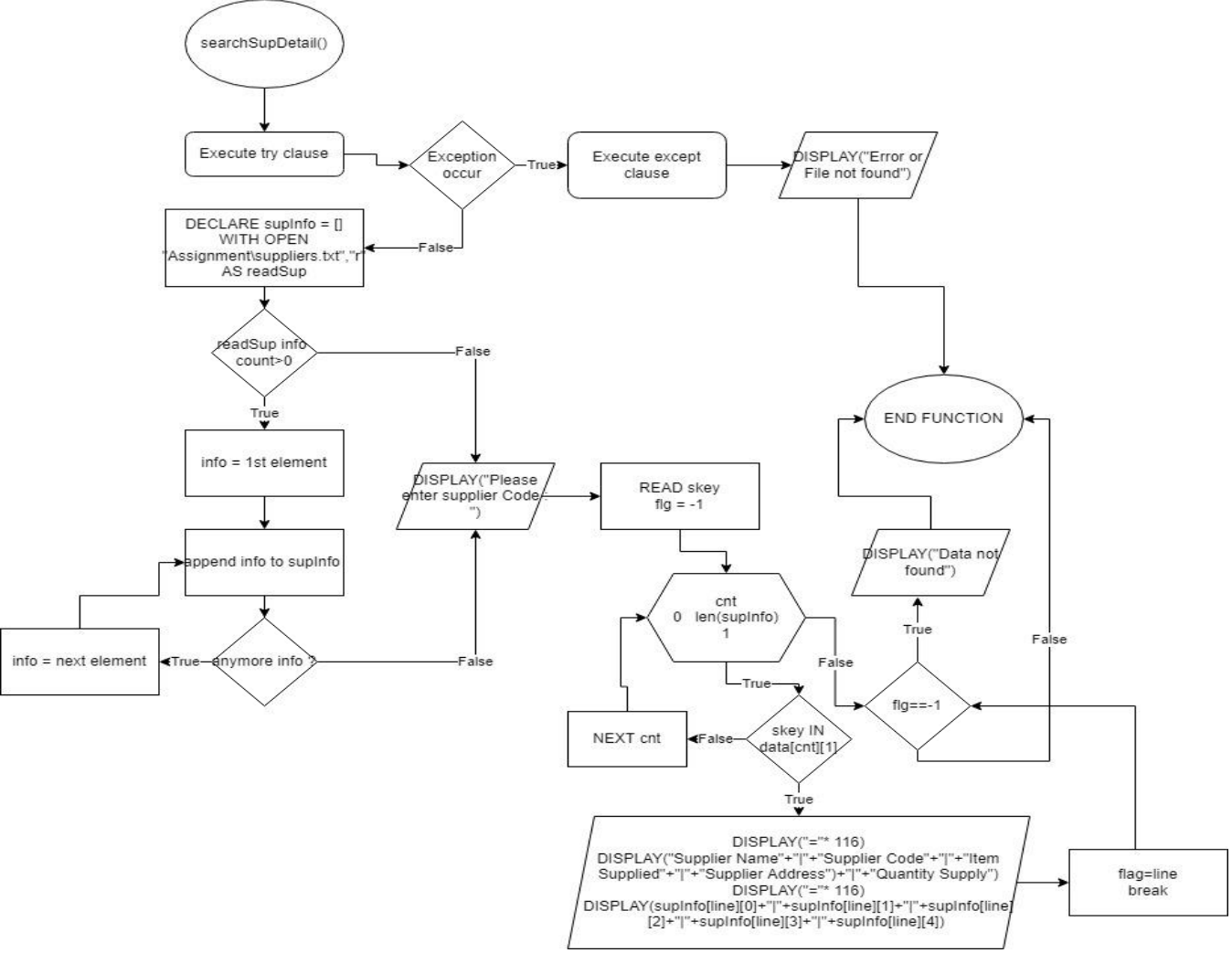

viewDistribution()
WITH OPEN "Assignment\distribution.txt" AS disfh
DECLARE dislist = []
fh elem count>0
True
False
elem = 1st element
append elem to dislist
elem = next element
True
anymore elem ?
False
DISPLAY("="* 77)
DISPLAY("Item Code"+"|"+"Item Quantity(Box)".+"|"+"Hospital Code"+"|"+"Distribution Code")
DISPLAY("="* 77)
END FUNCTION
False
line
0 len(dislist)
1
True
NEXT line
DISPLAY(dislist[line][0]+"|"+dislist[line][1]+"|"+dislist[line][2].+"|"+dislist[line][3])

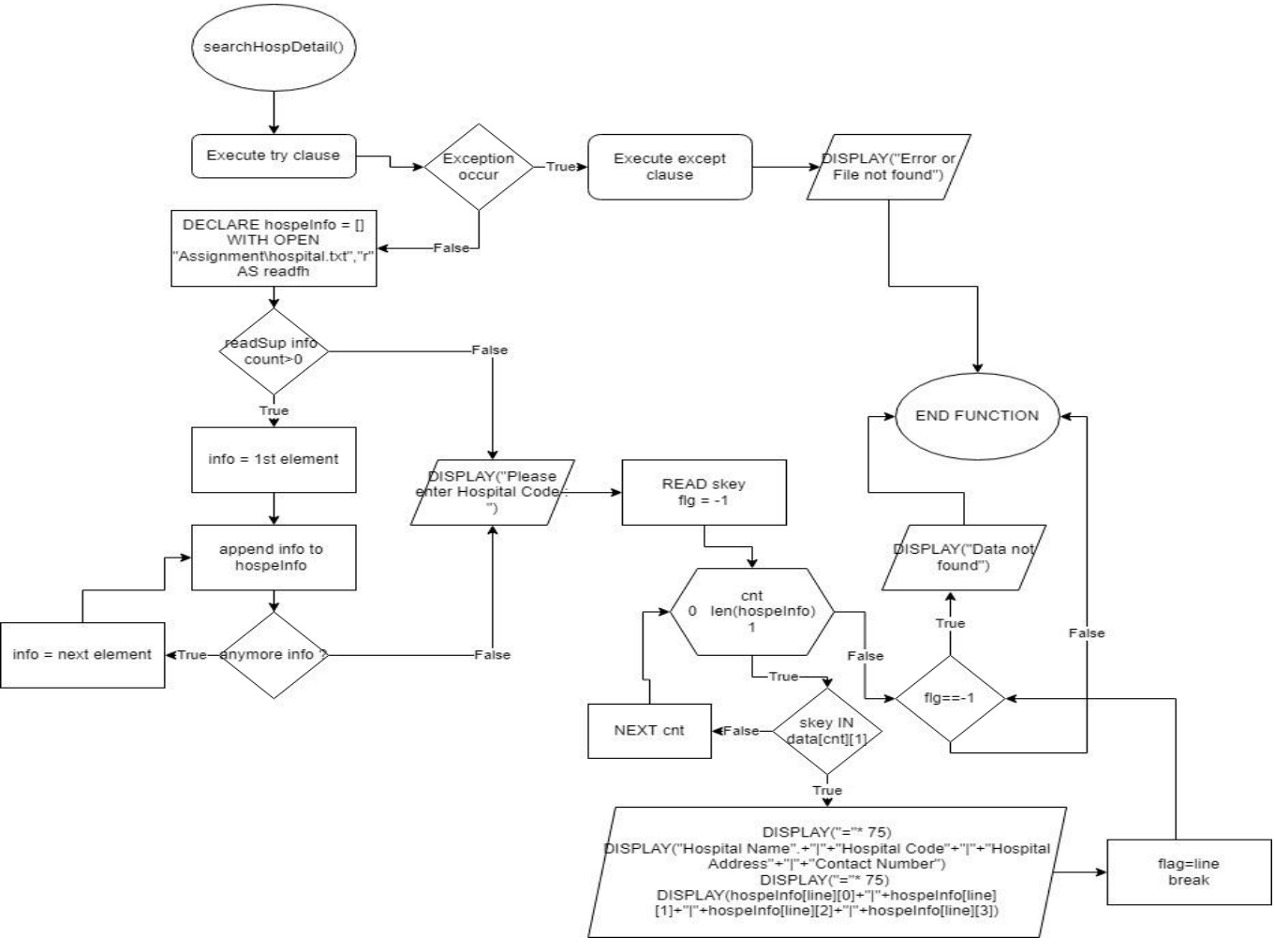

modifyPPE()
Execute try clause
Exception occur
True
Execute except clause
DISPLAY("Error or File not found")
False
DECLARE data = [] WITH OPEN "Assignment\ppe.txt","r" AS fh
fh line count>0
DISPLAY("Please enter Item Code : ")
True
line = 1st element
READ skey flg = -1
append line to data
cnt 0 len(data) 1
False
flg==-1
False
flg != -1
END FUNCTION
False
DISPLAY("Data not found")
True
line = next element
True
anymore line ?
NEXT cnt
False
skey IN data[cnt][1]
True
flag=cnt BREAK
True
DISPLAY("1 -Item Name : "+ data[flg][0])
DISPLAY("2 -Item Code : "+ data[flg][1])
DISPLAY("3 -Supplier Code : "+ data[flg][2])
DISPLAY("4 -Quantity : "+ data[flg][3])
DISPLAY("5- Year : "+ data[flg][4])
DISPLAY("6 -Month : "+ data[flg][5])
DISPLAY("7 -Date : "+ data[flg][6])
DISPLAY("Enter the number to modify :"))
ans = CALL limitopt(ans,7)
READ data[cnt][ans-1]
DISPLAY("***Succesfully changed***")
WITH OPEN "Assignment\ppe.txt","w" AS fh
cnt 0 len(data) 1
True
join(data[cnt]) with "-"+"\n" AS rec
WRITE rec TO fh
NEXT cnt

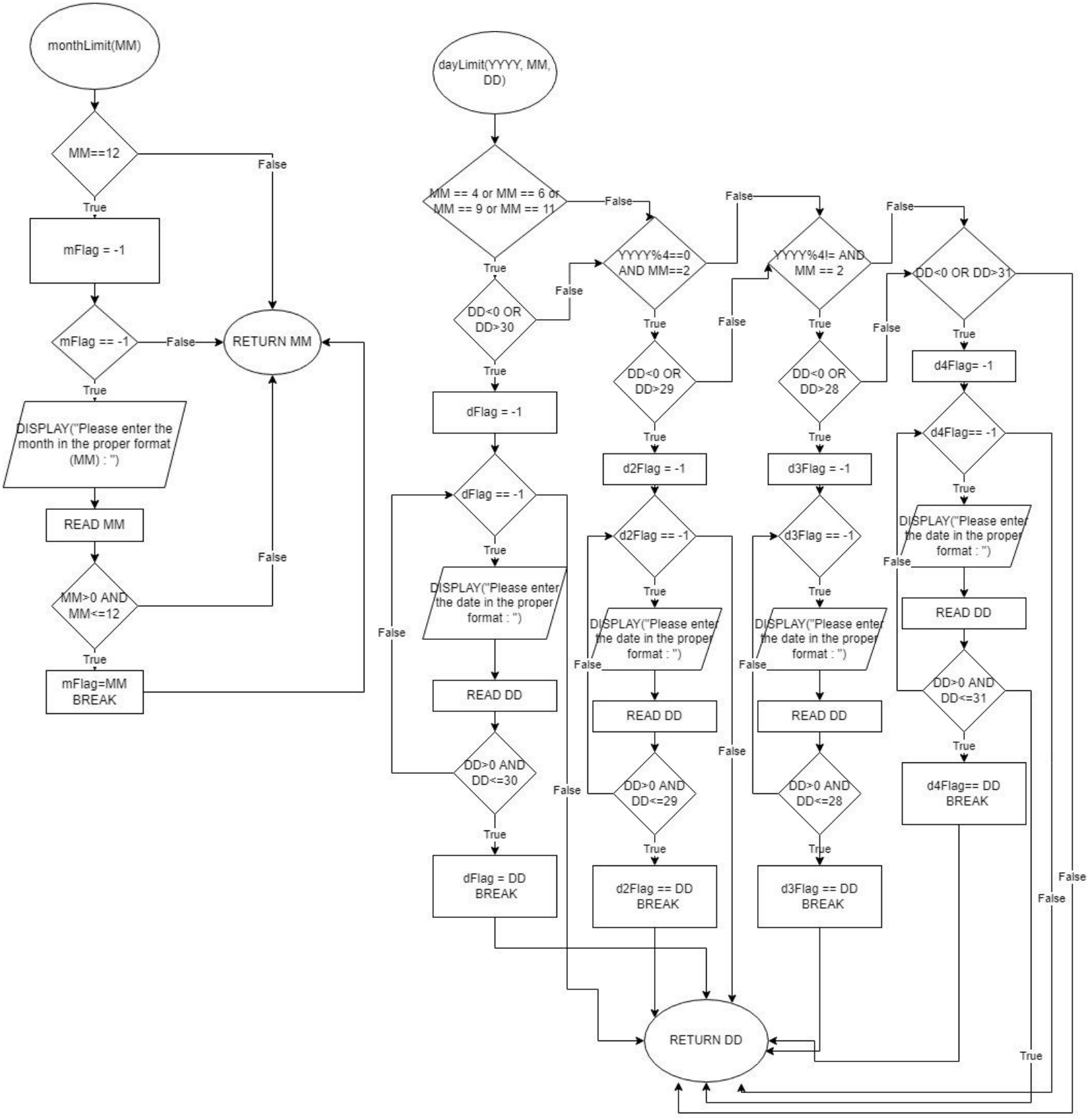

modifySup()
Execute try clause
Exception occur
True
Execute except clause
DISPLAY("Error or File not found")
False
DECLARE data = [] WITH OPEN "Assignment\suppliers.txt","r" AS fh
fh line count>0
DISPLAY("Please enter Supplier Code : ")
True
line = 1st element
READ skey flg = -1
append line to data
cnt 0 len(data) 1
False
flg==-1
False
flg != -1
END FUNCTION
False
DISPLAY("Data not found")
True
line = next element
True
anymore line ?
NEXT cnt
False
skey IN data[cnt][1]
True
flag=cnt BREAK
True
DISPLAY("1 -Supplier Name : "+ data[flg][0])
DISPLAY("2 -Supplier Code : "+ data[flg][1])
DISPLAY("3 -Item Supplied : "+ data[flg][2])
DISPLAY("4 -Supplier Address: "+ data[flg][3])
DISPLAY("5- Quantity Supplied : "+ data[flg][4]
DISPLAY("Enter the number to modify :"))
ans = CALL limitopt(ans,5)
READ data[cnt][ans-1]
DISPLAY("***Succesfully changed***")
WITH OPEN "Assignment\suppliers.txt","w" AS fh
cnt 0 len(data) 1
True
join(data[cnt]) with "-"+"\n" AS rec
WRITE rec TO fh
NEXT cnt

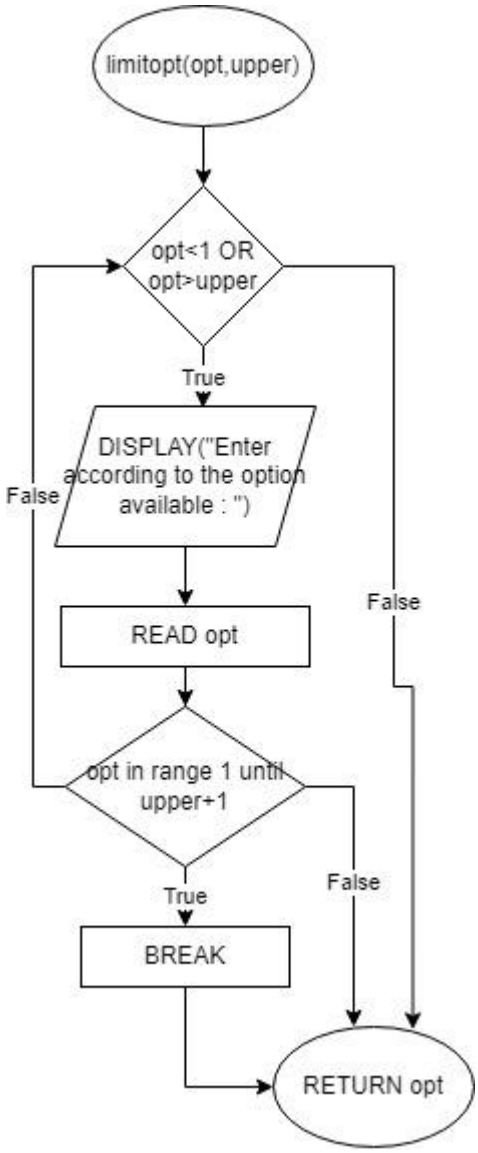

modifyHosp()
Execute try clause
Exception occur
True
Execute except clause
DISPLAY("Error or File not found")
False
DECLARE data = [] WITH OPEN "Assignment\hospital.txt","r" AS fh
fh line count>0
True
line = 1st element
append line to data
line = next element
True
anymore line ?
DISPLAY("Please enter hospitalCode ")
READ skey
flg = -1
cnt
0 len(data)
1
False
NEXT cnt
skey IN data[cnt][1]
True
flag=cnt
BREAK
flg==-1
True
DISPLAY("Data not found")
flg != -1
True
END FUNCTION
False
DISPLAY("1 -hospitalName : "+ data[flg][0])
DISPLAY("2 -hospitalCode : "+ data[flg][1])
DISPLAY("3 -Hospital Address: "+ data[flg][2])
DISPLAY("4 -Contact Number: "+ data[flg][3]
DISPLAY("Enter the number to modify :"))
ans = CALL limitopt(ans,4)
READ data[cnt][ans-1]
DISPLAY("****Succesfully changed****")
WITH OPEN "Assignment\hospital.txt","w" AS fh
cnt
0 len(data)
1
True
join(data[cnt]) with "-"+"\n" AS rec
WRITE rec TO fh
NEXT cnt

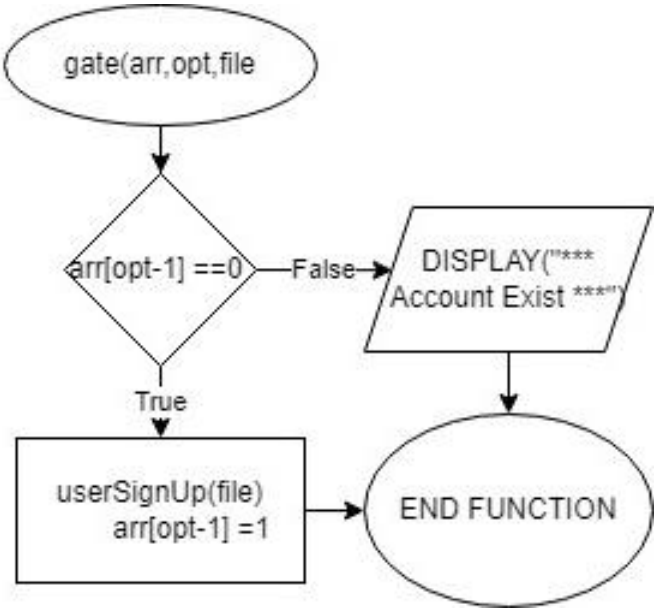

searchPPEDetail()
Execute try clause
Exception occur
True
Execute except clause
DISPLAY("Error or File not found")
False
DECLARE ppeInfo = [] WITH OPEN "Assignment\ppe.txt","r" AS readPPE
readPPE info count>0
False
True
info = 1st element
append info to ppeInfo
info = next element
True
anymore info ?
False
DISPLAY("Please enter item Code : ")
READ skey
flg = -1
cnt
0 len(ppeInfo)
1
True
NEXT cnt
False
skey IN data[cnt][1]
True
DISPLAY("Data not found")
True
flg==-1
False
END FUNCTION
DISPLAY("="* 115)
DISPLAY("Item Name".+"|"+"Item Code"+"|"+"Supplier Code"+"|"+"Item Quantity(Box)"+"|"+"Date Inputted")
DISPLAY("="* 115)
DISPLAY(ppeInfo[line][0]+"|"+ppeInfo[line][1]+"|"+ppeInfo[line][2].+"|"+ppeInfo[line][3]+"|"+(ppeInfo[line][4]+"/"+ppeInfo[line][5]+"/"+ppeInfo[line][6]))
flag=line break

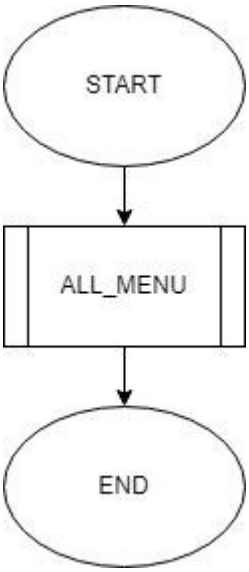

searchSupDetail()
Execute try clause
Exception occur
True
Execute except clause
DISPLAY("Error or File not found")
False
DECLARE supInfo = [] WITH OPEN "Assignment\suppliers.txt","r" AS readSup
readSup info count>0
False
True
info = 1st element
append info to supInfo
info = next element
True
anymore info ?
False
DISPLAY("Please enter supplier Code: ")
READ skey flg = -1
cnt 0 len(supInfo) 1
True
False
skey IN data[cnt][1]
False
NEXT cnt
True
flg==-1
True
False
DISPLAY("Data not found")
END FUNCTION
DISPLAY("="* 116)
DISPLAY("Supplier Name"+"|"+"Supplier Code"+"|"+"Item Supplied"+"|"+"Supplier Address")+"|"+"Quantity Supply")
DISPLAY("="* 116)
DISPLAY(supInfo[line][0]+"|"+supInfo[line][1]+"|"+supInfo[line][2]+"|"+supInfo[line][3]+"|"+supInfo[line][4])
flag=line break

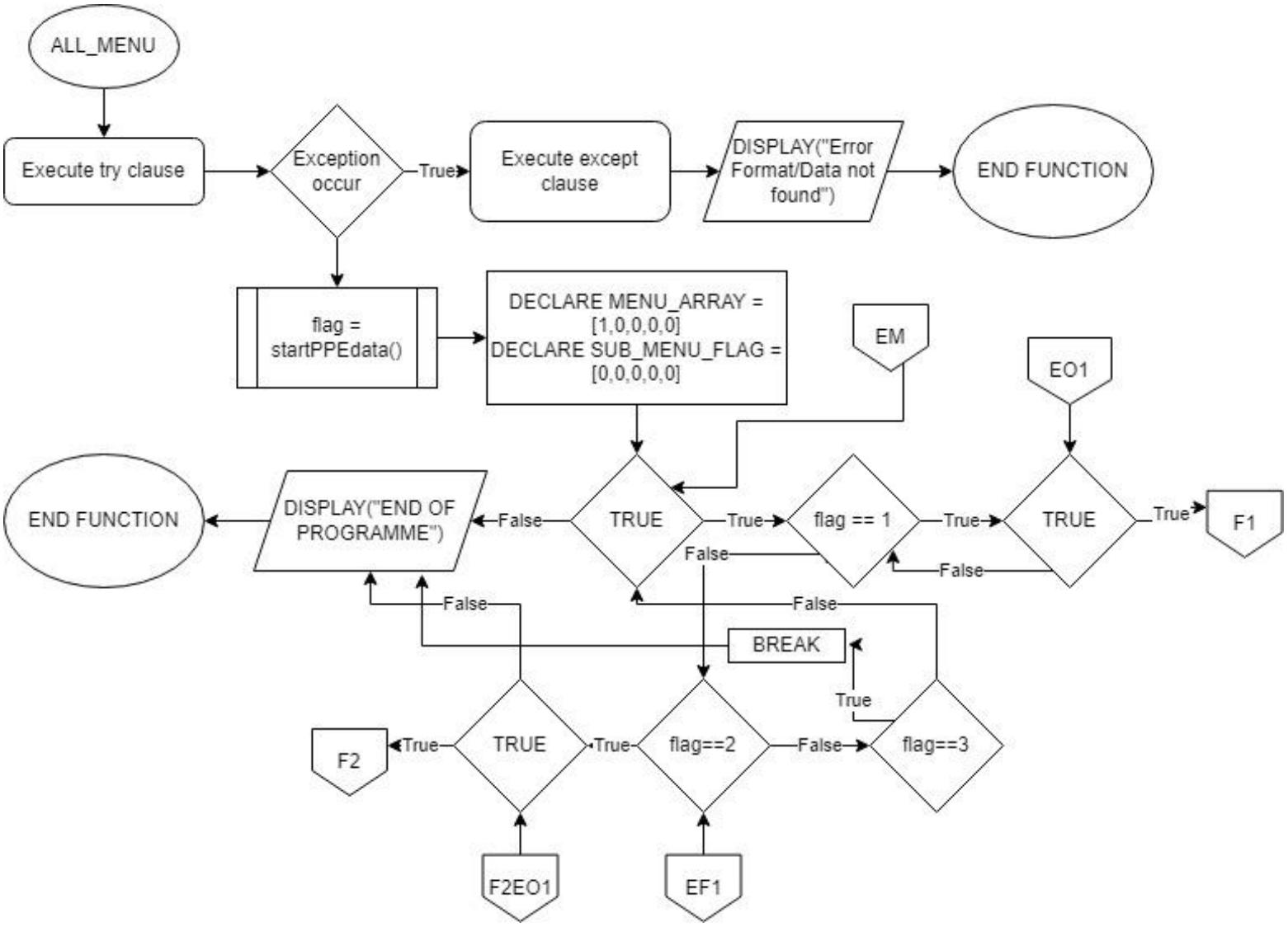

searchHospDetail()
Execute try clause
Exception occur
True
Execute except clause
DISPLAY("Error or File not found")
False
DECLARE hospeInfo = [] WITH OPEN "Assignment\hospital.txt","r" AS readfh
readSup info count>0
False
True
info = 1st element
append info to hospeInfo
info = next element
True
anymore info ?
False
DISPLAY("Please enter Hospital Code: ")
READ skey flg = -1
cnt 0 len(hospeInfo) 1
True
False
skey IN data[cnt][1]
False
NEXT cnt
True
flg==-1
True
False
DISPLAY("Data not found")
END FUNCTION
DISPLAY("="* 75)
DISPLAY("Hospital Name".+"|"+"Hospital Code"+"|"+"Hospital Address"+"|"+"Contact Number")
DISPLAY("="* 75)
DISPLAY(hospeInfo[line][0]+"|"+hospeInfo[line][1]+"|"+hospeInfo[line][2]+"|"+hospeInfo[line][3])
flag=line break

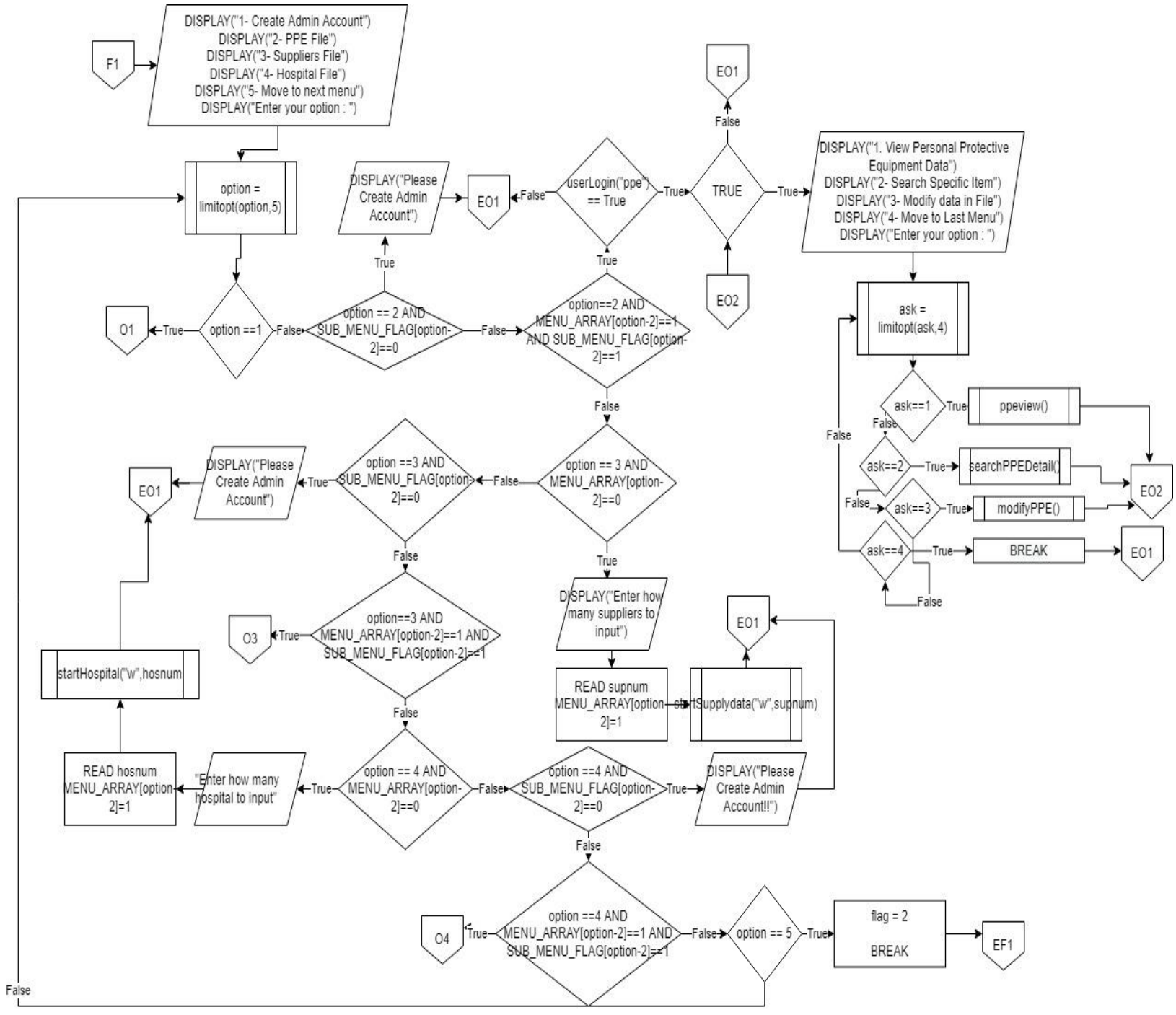

monthLimit(MM)
MM==12
False
True
mFlag = -1
mFlag == -1
False
RETURN MM
True
DISPLAY("Please enter the month in the proper format (MM) : ")
READ MM
MM>0 AND MM<=12
False
True
mFlag=MM
BREAK
dayLimit(YYYY, MM, DD)
MM == 4 or MM == 6 or MM == 9 or MM == 11
False
True
DD<0 OR DD>30
False
True
dFlag = -1
dFlag == -1
True
DISPLAY("Please enter the date in the proper format : ")
READ DD
DD>0 AND DD<=30
False
True
dFlag = DD
BREAK
YYYY%4==0 AND MM==2
False
True
DD<0 OR DD>29
False
True
d2Flag = -1
d2Flag == -1
True
DISPLAY("Please enter the date in the proper format : ")
READ DD
DD>0 AND DD<=29
False
True
d2Flag == DD
BREAK
YYYY%4!= AND MM == 2
False
True
DD<0 OR DD>28
False
True
d3Flag = -1
d3Flag == -1
True
DISPLAY("Please enter the date in the proper format : ")
READ DD
DD>0 AND DD<=28
False
True
d3Flag == DD
BREAK
DD<0 OR DD>31
False
True
d4Flag= -1
d4Flag== -1
True
DISPLAY("Please enter the date in the proper format : ")
READ DD
DD>0 AND DD<=31
False
True
d4Flag== DD
BREAK
RETURN DD
False
True

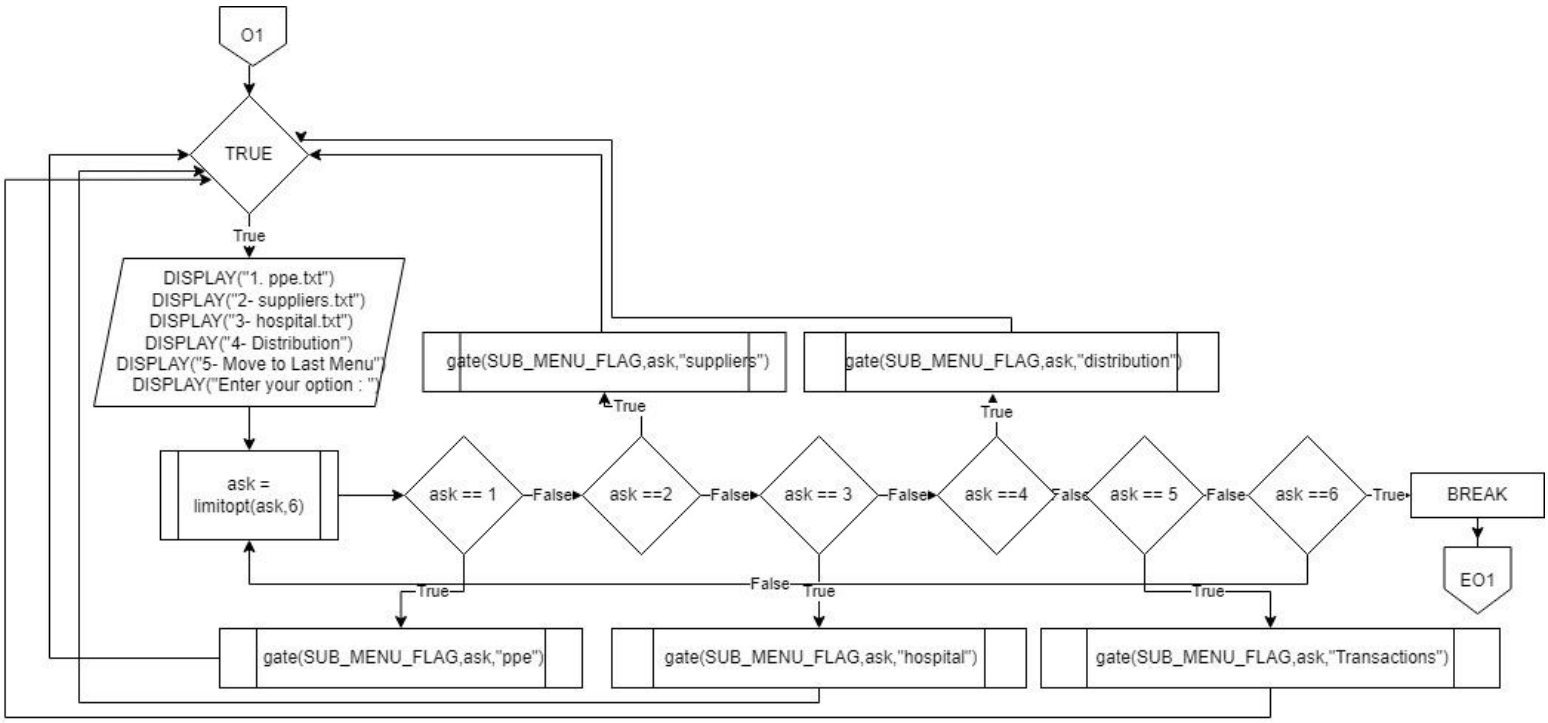

limitopt(opt,upper)
opt<1 OR opt>upper
True
DISPLAY("Enter according to the option available : ")
READ opt
opt in range 1 until upper+1
True
BREAK
False
False
False
RETURN opt

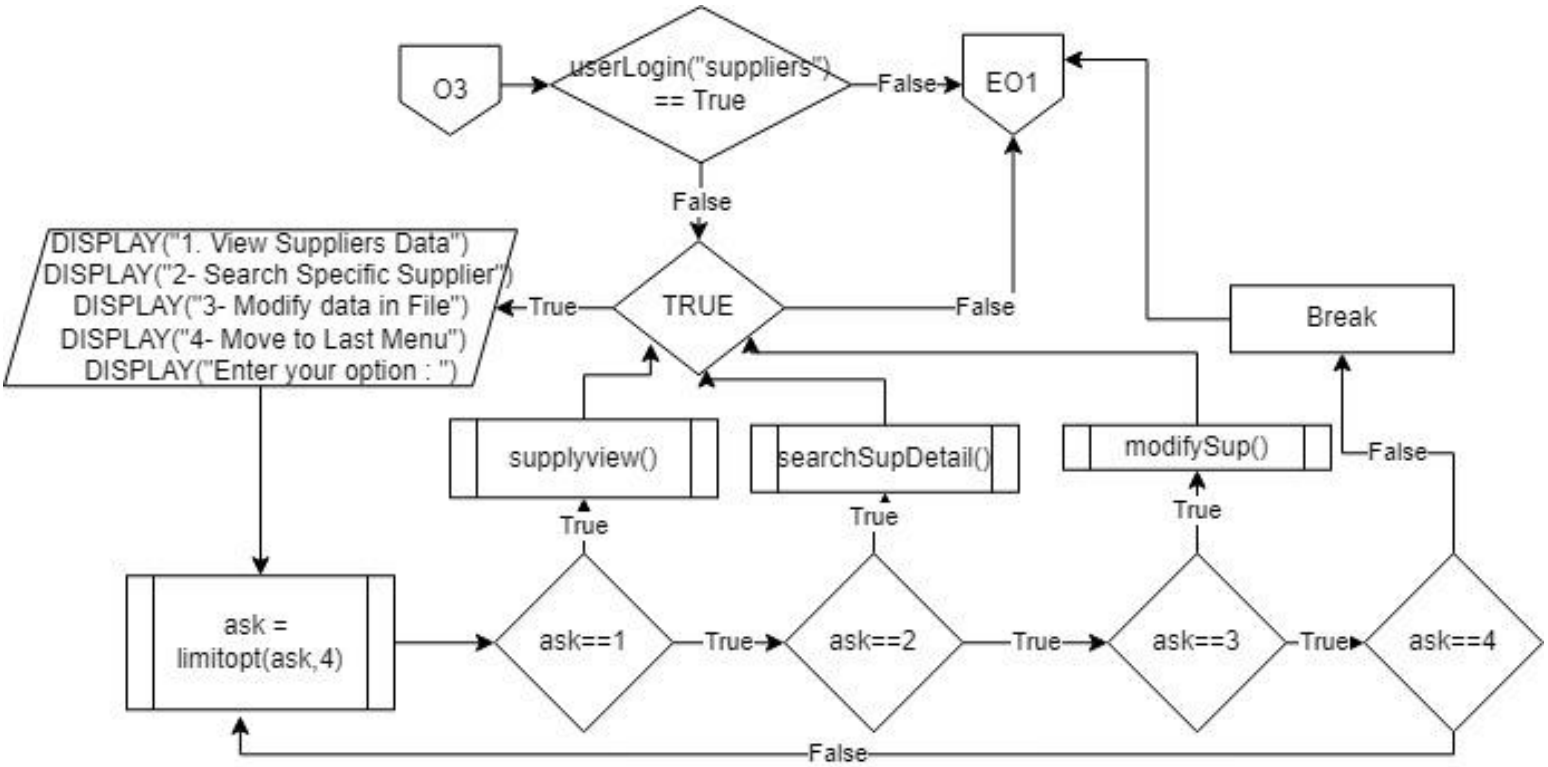

gate(arr,opt,file
arr[opt-1] ==0
False
DISPLAY("*** Account Exist ***")
True
userSignUp(file)
arr[opt-1] =1
END FUNCTION

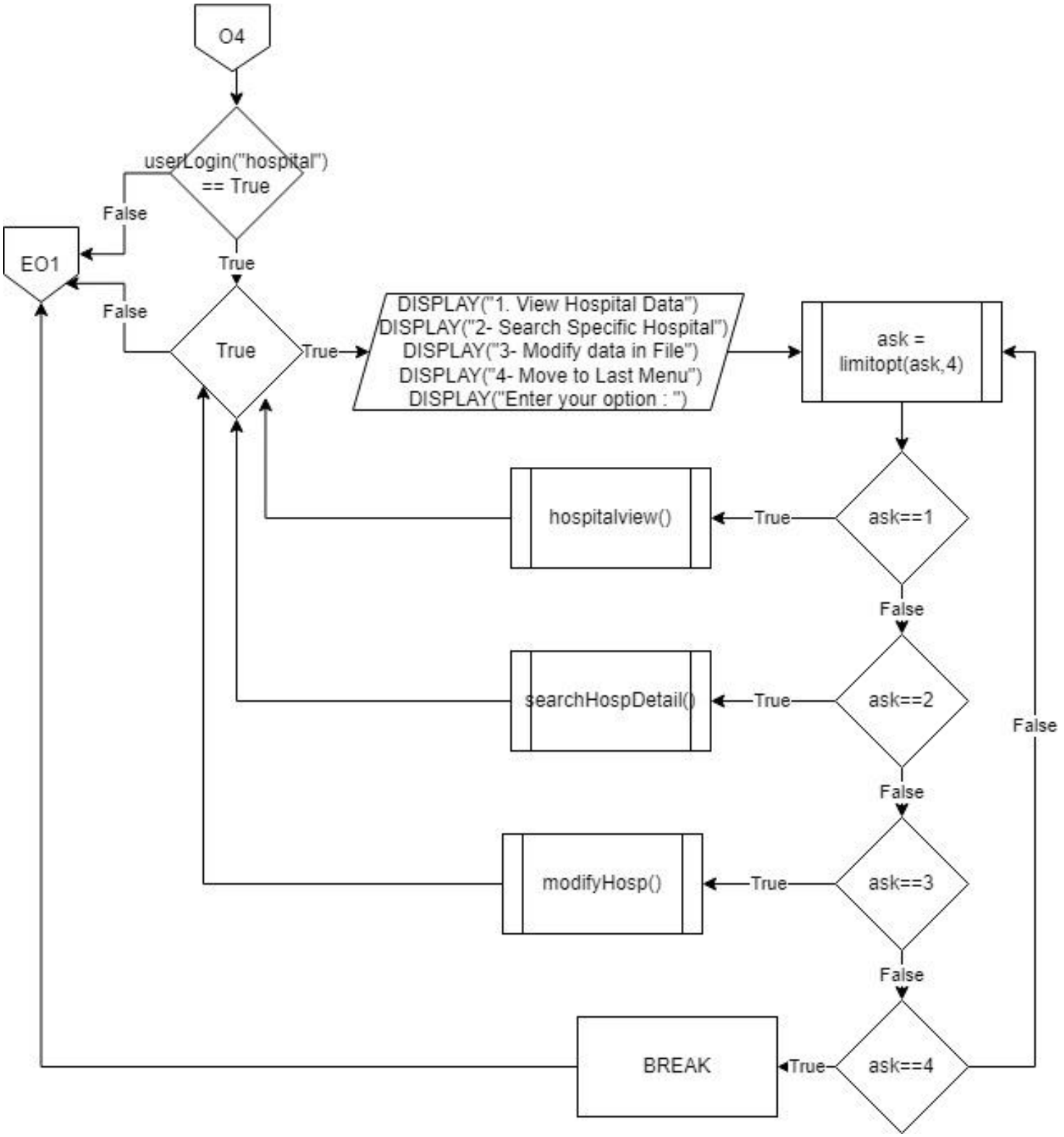

START
ALL_MENU
END

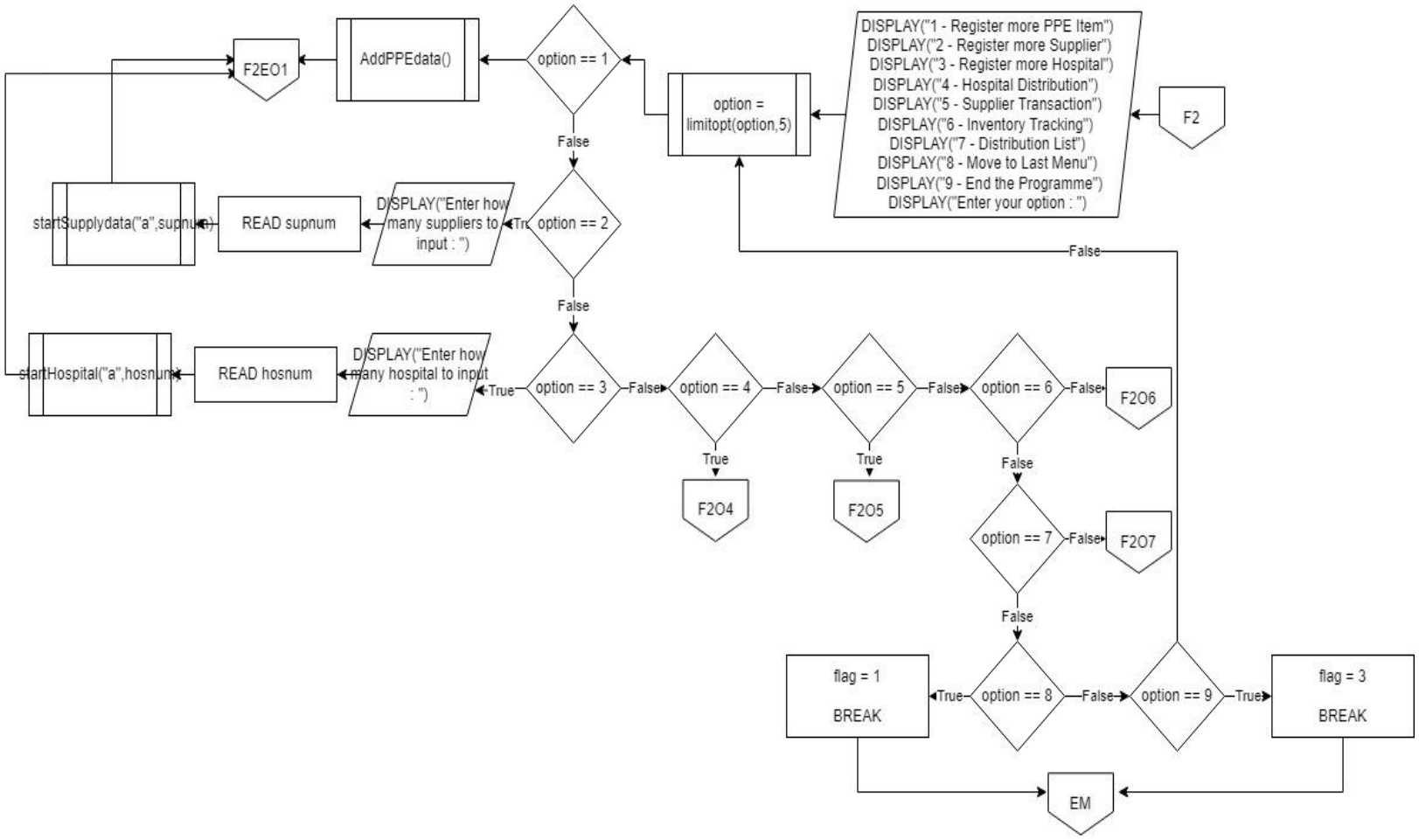

ALL_MENU
Execute try clause
Exception occur
True
Execute except clause
DISPLAY("Error Format/Data not found")
END FUNCTION
flag = startPPEdata()
DECLARE MENU_ARRAY = [1,0,0,0,0]
DECLARE SUB_MENU_FLAG = [0,0,0,0,0]
EM
EO1
END FUNCTION
DISPLAY("END OF PROGRAMME")
False
TRUE
True
flag == 1
True
TRUE
True
F1
False
False
False
False
BREAK
True
F2
True
TRUE
True
flag==2
False
flag==3
F2EO1
EF1

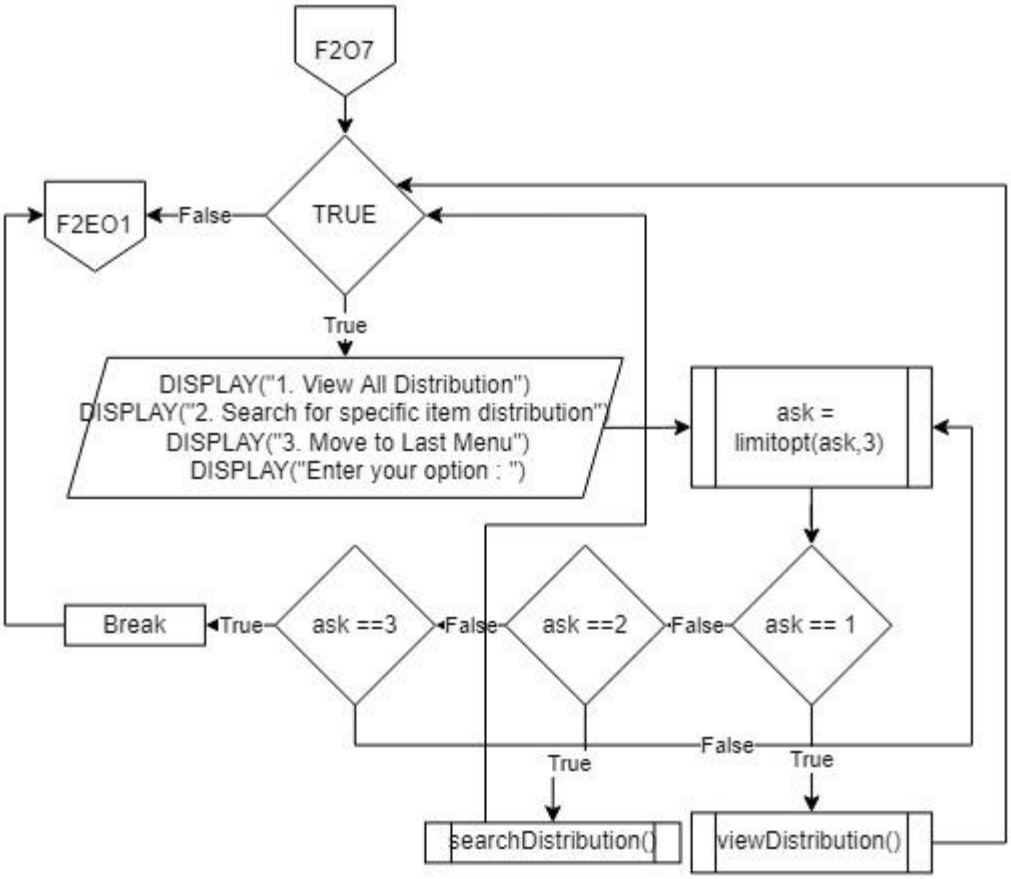

F1
DISPLAY("1- Create Admin Account")
DISPLAY("2- PPE File")
DISPLAY("3- Suppliers File")
DISPLAY("4- Hospital File")
DISPLAY("5- Move to next menu")
DISPLAY("Enter your option : ")
option = limitopt(option,5)
O1
option ==1
True
False
option == 2 AND SUB_MENU_FLAG[option-2]==0
True
DISPLAY("Please Create Admin Account")
EO1
False
userLogin("ppe") == True
True
option==2 AND MENU_ARRAY[option-2]==1 AND SUB_MENU_FLAG[option-2]==1
True
TRUE
True
EO1
False
EO2
DISPLAY("1. View Personal Protective Equipment Data")
DISPLAY("2- Search Specific Item")
DISPLAY("3- Modify data in File")
DISPLAY("4- Move to Last Menu")
DISPLAY("Enter your option : ")
ask = limitopt(ask,4)
ask==1
True
ppeview()
False
ask==2
True
searchPPEDetail()
EO2
False
ask==3
True
modifyPPE()
EO2
False
ask==4
True
BREAK
EO1
False
False
option == 3 AND MENU_ARRAY[option-2]==0
True
DISPLAY("Enter how many suppliers to input")
READ supnum MENU_ARRAY[option-2]=1
startSupplydata("w",supnum)
EO1
False
option ==3 AND SUB_MENU_FLAG[option-2]==0
True
DISPLAY("Please Create Admin Account")
EO1
False
O3
option==3 AND MENU_ARRAY[option-2]==1 AND SUB_MENU_FLAG[option-2]==1
True
startHospital("w",hosnum
False
option == 4 AND MENU_ARRAY[option-2]==0
True
"Enter how many hospital to input"
READ hosnum MENU_ARRAY[option-2]=1
False
option ==4 AND SUB_MENU_FLAG[option-2]==0
True
DISPLAY("Please Create Admin Account!!")
False
O4
option ==4 AND MENU_ARRAY[option-2]==1 AND SUB_MENU_FLAG[option-2]==1
True
False
option == 5
True
flag = 2
BREAK
EF1
False

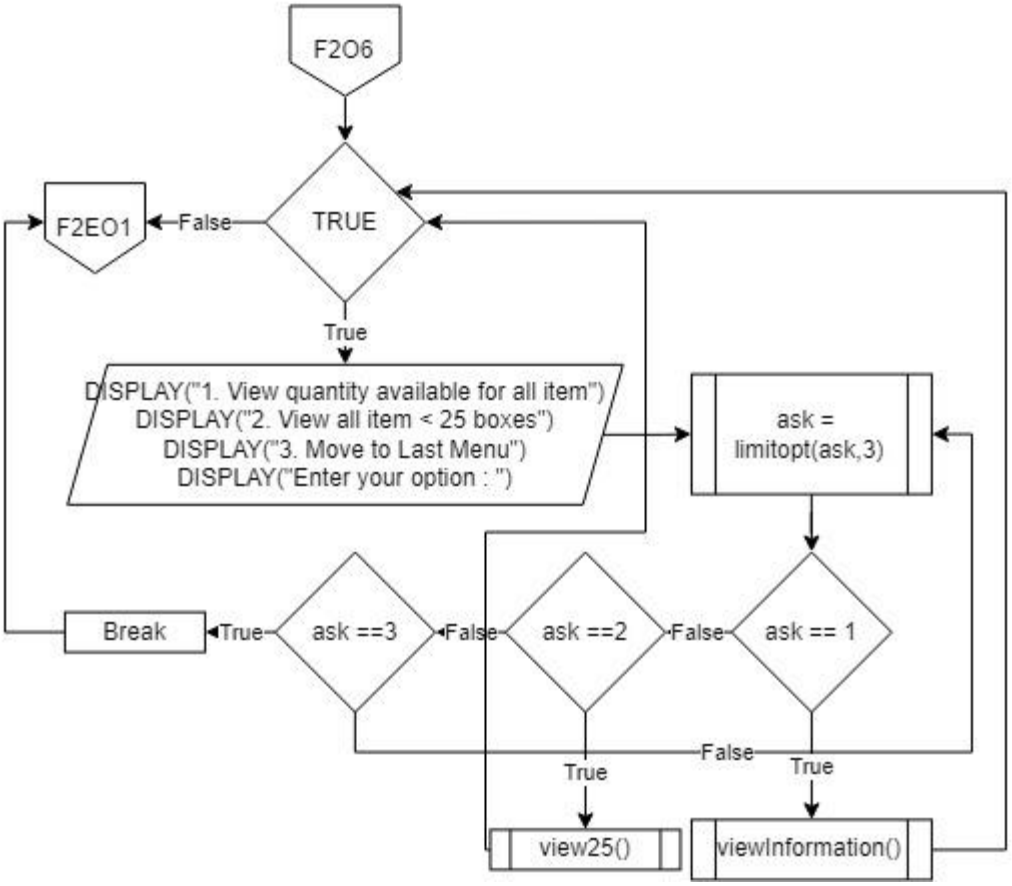

O1
TRUE
True
DISPLAY("1. ppe.txt")
DISPLAY("2- suppliers.txt")
DISPLAY("3- hospital.txt")
DISPLAY("4- Distribution")
DISPLAY("5- Move to Last Menu")
DISPLAY("Enter your option : ")
gate(SUB_MENU_FLAG,ask,"suppliers")
gate(SUB_MENU_FLAG,ask,"distribution")
ask = limitopt(ask,6)
ask == 1
ask ==2
ask == 3
ask ==4
ask == 5
ask ==6
BREAK
EO1
False
True
gate(SUB_MENU_FLAG,ask,"ppe")
gate(SUB_MENU_FLAG,ask,"hospital")
gate(SUB_MENU_FLAG,ask,"Transactions")

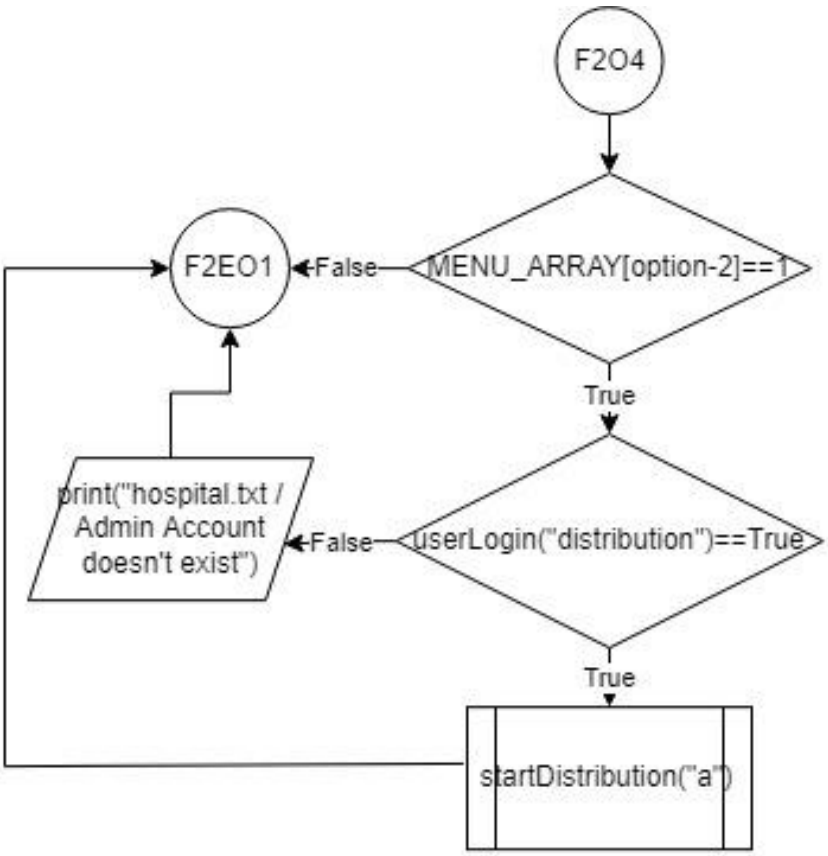

O3
userLogin("suppliers") == True
False
EO1
DISPLAY("1. View Suppliers Data")
DISPLAY("2- Search Specific Supplier")
DISPLAY("3- Modify data in File")
DISPLAY("4- Move to Last Menu")
DISPLAY("Enter your option : ")
TRUE
Break
supplyview()
searchSupDetail()
modifySup()
ask = limitopt(ask,4)
ask==1
ask==2
ask==3
ask==4

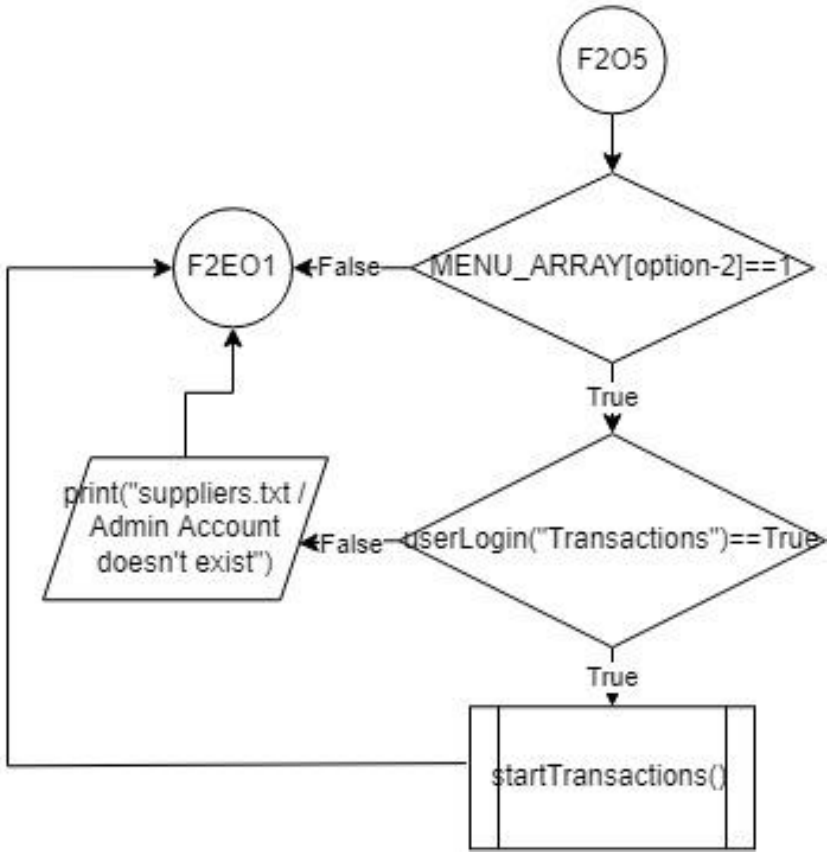

O4
userLogin("hospital") == True
False
True
EO1
False
True
True
DISPLAY("1. View Hospital Data")
DISPLAY("2- Search Specific Hospital")
DISPLAY("3- Modify data in File")
DISPLAY("4- Move to Last Menu")
DISPLAY("Enter your option : ")
ask = limitopt(ask,4)
ask==1
True
hospitalview()
False
ask==2
True
searchHospDetail()
False
ask==3
True
modifyHosp()
False
ask==4
True
BREAK
False


F2
DISPLAY("1 - Register more PPE Item")
DISPLAY("2 - Register more Supplier")
DISPLAY("3 - Register more Hospital")
DISPLAY("4 - Hospital Distribution")
DISPLAY("5 - Supplier Transaction")
DISPLAY("6 - Inventory Tracking")
DISPLAY("7 - Distribution List")
DISPLAY("8 - Move to Last Menu")
DISPLAY("9 - End the Programme")
DISPLAY("Enter your option : ")
option = limitopt(option,5)
option == 1
AddPPEdata()
F2EO1
False
option == 2
DISPLAY("Enter how many suppliers to input : ")
READ supnum
startSupplydata("a",supnum)
False
option == 3
DISPLAY("Enter how many hospital to input : ")
READ hosnum
startHospital("a",hosnum)
True
option == 4
F2O4
True
option == 5
F2O5
True
option == 6
F2O6
False
option == 7
F2O7
False
option == 8
flag = 1
BREAK
True
option == 9
flag = 3
BREAK
True
EM
False
False
False
False
False
False
False


F2O7
TRUE
False
F2EO1
True
DISPLAY("1. View All Distribution")
DISPLAY("2. Search for specific item distribution")
DISPLAY("3. Move to Last Menu")
DISPLAY("Enter your option : ")
ask = limitopt(ask,3)
Break
True
ask ==3
False
ask ==2
False
ask == 1
False
True
True
searchDistribution()
viewDistribution()


F2O6
TRUE
False
F2EO1
True
DISPLAY("1. View quantity available for all item")
DISPLAY("2. View all item < 25 boxes")
DISPLAY("3. Move to Last Menu")
DISPLAY("Enter your option : ")
ask = limitopt(ask,3)
Break
True
ask ==3
False
ask ==2
False
ask == 1
False
True
True
view25()
viewInformation()


F2O4
F2EO1
False
MENU_ARRAY[option-2]==1
True
print("hospital.txt / Admin Account doesn't exist")
False
userLogin("distribution")==True
True
startDistribution("a")


F2O5
F2EO1
False
MENU_ARRAY[option-2]==1
True
print("suppliers.txt / Admin Account doesn't exist")
False
userLogin("Transactions")==True
True
startTransactions()

Program Source Code

```
import datetime
```

Figure 1 import datetime

This module is called on the first line of the code to import the module that show the current day, month, year, and time with good accuracy and validation.

```
def startPPEdata():
    genAuthcode()
    print("Welcome to the initial inventory creation")
    print("Please fill the required information with the correct format")
    try:
        with open("Assignment\ppe.txt","w") as ppeFile:
            # Keep asking for input until user wish to discontinue
            while True:
                itemname = input("Please enter the PPE item name : ")
                itemcode = input("Please enter the item code : ")
                suppliercode = input("Register / Enter Supplier Code : ")
                quantity = int(100)
                YYYY = int(input("Please enter the year item inputted (YYYY) : "))
                MM = int(input("Please enter the month item inputted (MM) : "))
                # to limit month from 1 to 12 only
                MM = monthLimit(MM)
                DD = int(input("Please enter the date item inputted (DD) : "))
                # setting limitation for days in every month
                DD = dayLimit(YYYY, MM, DD)
                ppeData = itemname+"-"+itemcode+"-"+suppliercode+"-"+str(quantity)+"-"+str(YYYY)+"-"+str(MM)+"-"+str(DD)+"\n"
                ppeFile.write(ppeData)
                cont = input("Press Enter key to continue or [q] to stop : ")
                if cont.lower()=="q":
                    print("PPE data has been recorded to ppe.txt file")
                    break
        return 1 # this value will be given to flag on ALL_MENU()
    except:
        print("Error Format Detected, Please try again")
```

Figure 2 startPPEdata() function

The startPPEdata() function contain code that will firstly call another function which is genAuthcode() to generate initial code for hospital, distribution, and transactions. The rest of the code purpose is to prompt necessary detail from the users and save it to a file called ppe.txt.

```
def startSupplydata(mode,supnum):
    try:
        with open("Assignment\ppe.txt","r") as readPPe:
            ppeList = []
            for data in readPPe:
                ppeList.append(data.strip().split("-"))
        # LIST FOR CHECKING WHETHER THE SUPPLY ID HAS ALREADY EXISTED
        with open("Assignment\suppliers.txt",mode) as insupply:
            for i in range(supnum):
                checklist = []
                supID = input("Please enter registered supplier code / [q] to stop : ")
                if supID.lower()=="q":
                    break

                with open("Assignment\suppliers.txt","r") as readfh:
                    for data in readfh:
                        checklist.append(data.strip().split("-"))

                    checkf=-1
                    for cnt in range(len(checklist)):
                        if supID in checklist[cnt][1]: # IF SUPPLY ID EXISTED
                            print("This supplier code has been inputted")
                            checkf=cnt
                            break
                    if checkf==-1: # IF NOT EXISTED INPUT THE SUPPLY ID INTO suppliers.txt
                        fl = -1
                        for cnt in range(len(ppeList)):
                            if supID in ppeList[cnt][2]:
                                fl=cnt
                                print("Item Supplied : "+ ppeList[cnt][0])
                                supItem = ppeList[cnt][0]
                                supName = input("Enter supplier name : ")
                                supAddress = input("Enter supplier address : ")
                                supQuant = ppeList[cnt][3]
                                suppliers = supName+"-"+supID+"-"+supItem+"-"+supAddress+"-"+str(supQuant)+"\n"
                                insupply.write(suppliers)
                                insupply.flush() # TO REMOVE BUFFER TIME
                        if fl==-1:
                            print("Supplier code not found")
            print("suppliers has been recorded in suppliers.txt")
    except:
        print("Error format / Data not found")
```

Figure 3 startSupplydata(mode,supnum) function

This function will check for the existence of the supplier code that user will input by comparing it to the data inside ppe.txt and suppliers.txt to prevent duplication of data. When clarification completed, users will be able to input all the detail about the suppliers that will be saved into the suppliers.txt file.

```
def startHospital(mode,hosnum):
    with open("Assignment\ppe.txt","r") as ppefh:
        list = []
        for data in ppefh:
            list.append(data.strip().split("-"))
        IDlist = []
        for cnt in list:
            IDlist.append(cnt[1])
    try:
        with open("Assignment\hospital.txt",mode) as fh:
            for x in range(hosnum):
                hosName = input("Enter Hospital name or [q] to stop : ")
                if hosName.lower() == "q":
                    break
                else:
                    u_code(0)
                    hosID = readID(0)
                    hosAdd = input("Enter hospital address : ")
                    hosContact = input("Enter hospital contact number : ")
                    hosData = hosName +"-"+ hosID+"-"+ hosAdd +"-"+ hosContact
                    for i in range(len(IDlist)):
                        # TO LIST OUT ALL THE PPE WITH 0 AS ITS INITIAL QUANTITY OWN BY EACH HOSPITAL
                        hosData = hosData +"-"+ IDlist[i]+"-"+"0"
                    hosData = hosData +"\n"
                    print("Registered Hospital Code is : "+ hosID)
                    fh.write(hosData)
                print("Hospitals has been recorded into hospital.txt")
    except:
        ("Error format / Data not found")
```

Figure 4 startHospital(mode,hosnum) function

This function will prompt user for the details about hospitals that need registration. These details will later record into a file called hospital.txt.

```
def startDistribution(mode):
    try:
        with open("Assignment\ppe.txt") as ppefh:
            ppeList = []
            for elem in ppefh:
                ppeList.append(elem.strip().split("-"))

        with open("Assignment\hospital.txt") as hospitalfh:
            hospitalList = []
            for elem in hospitalfh:
                hospitalList.append(elem.strip().split("-"))

        with open ("Assignment\distribution.txt",mode) as distFh:
            while True:
                itemID = input("Enter the item code for distribution : ")
                itemflag = 1
                for line in range(len(ppeList)):
                    if itemID in ppeList[line][1]:
                        itemflag = 2
                        print("This item has quantity of "+ppeList[line][3]+" boxes")
                        send_item = int(input("How many boxes to distribute ? : "))
                        int_itemquantity = int(ppeList[line][3])
                        if send_item>int_itemquantity:
                            while True:
                                send_item = int(input("Inssuficcient amount, Enter the correct amount : "))
                                if send_item<=int_itemquantity:
                                    break
                        hospID = input("Enter the destination hospital code : ")
                        hospFlag = 1
                        l2flag = -1
                        for line in range(len(hospitalList)):
                            if hospID in hospitalList[line][1]:
                                for cnt in range(4,len(hospitalList[line])):
                                    # Adding to Hospital File
                                    if itemID in hospitalList[line][cnt] and cnt%2==0:
                                        hosquant = int(hospitalList[line][cnt+1])
                                        receive_quant = hosquant + send_item
                                        hospitalList[line][cnt+1] = str(receive_quant)
                                        hospFlag = 2
                                        print("This item has been distributed")
                                        u_code(1)
                                        distID = readID(1)
                                        print("Distribution Code is : "+distID)
                                        distData = itemID+"-"+str(send_item)+"-"+hospID+"-"+distID+"\n"
                                        distFh.write(distData)
                                        break

                                # Subtracting from PPE File
                                for cnt in range(len(ppeList)):
                                    if itemID in ppeList[cnt][1]:
                                        break
                                current_quantity = int(ppeList[cnt][3])
                                new_quant = current_quantity - send_item
                                ppeList[cnt][3] = str(new_quant)

                                with open("Assignment\ppe.txt","w") as pfh:
                                    for cnt in range(len(ppeList)):
                                        rec = "-".join(ppeList[cnt])+"\n"
                                        pfh.write(rec)

                                with open("Assignment\hospital.txt","w") as hfh:
                                    for cnt in range(len(hospitalList)):
                                        rec = "-".join(hospitalList[cnt])+"\n"
                                        hfh.write(rec)

                                cont = input("Enter [n] to exit this process : ")
                                if cont.lower() == "n":
                                    print("Distribution has been recorded to distribution.txt")
                                    l2flag =1
                                    break # break inner for loop
                            if l2flag == 1: # flag to break outer for loop
                                break
```

```
                        if hospFlag == 1:
                            print("Hospital ID not found")
                            break
                    if itemflag == 1:
                        print("Item not found")
                        break
                    if l2flag == 1:
                        break # TO BREAK WHILE LOOP
        except:
            print("Error format/File doesn't exist")
```

Figure 5 StartDistribution(mode) function

This function will firstly check and compare the data in 2 files which are ppe.txt and hospital.txt. The program will then prompt user for the distribution details which required users to fill the data according to the validation given. Program will then write the data given by the users and record it to distribution.txt file. Additionally, the quantity number of items in ppe.txt and hospital.txt has change accordingly to the action done in this section.

```
def startTransactions():
    try:
        with open("Assignment\suppliers.txt","r") as supfh:
            suplist = []
            for data in supfh:
                suplist.append(data.strip().split("-"))

        with open("Assignment\ppe.txt") as ppefh:
            ppelist = []
            for data in ppefh:
                ppelist.append(data.strip().split("-"))

        with open("Assignment\Transactions.txt","a") as transfh:
            while True:
                supplierID = input("Enter supplier ID : ")
                supflag = -1
                l1flag = -1
                for cnt in range(len(suplist)):
                    if supplierID in suplist[cnt][1]:
                        supflag = cnt
                        sup_quant = int(suplist[cnt][4])
                        item_name = suplist[cnt][2]
                        for cnt_ppe in range(len(ppelist)):
                            if item_name in ppelist[cnt_ppe][0]:
                                ItemID = ppelist[cnt_ppe][1]
                                break
                        print("Item ID is :", ItemID)
                        for cnt in range(len(ppelist)):
                            if ItemID in ppelist[cnt][1]:
                                ppe_quant = int(ppelist[cnt][3])
                                receive_quantity = int(input("Enter quantity of item added : "))
                                new_sup_q = sup_quant + receive_quantity
                                new_ppe_q = ppe_quant + receive_quantity
                                suplist[cnt][4] = str(new_sup_q)
                                ppelist[cnt][3] = str(new_ppe_q)
                                u_code(2)
                                TransID = readID(2)
```

```
                        print("Transaction ID :", TransID)
                        Date = datetime.date.today()
                        trans = ItemID + "-"+supplierID+"-"+str(receive_quantity)+"-"+str(Date)+"-"+TransID+"\n"
                        transfh.write(trans)

                        with open("Assignment\ppe.txt","w") as fh:
                            for cnt in range(len(ppelist)):
                                rec = "-".join(ppelist[cnt])+"\n"
                                fh.write(rec)
                        with open("Assignment\suppliers.txt","w") as fh:
                            for cnt in range(len(suplist)):
                                rec = "-".join(suplist[cnt])+"\n"
                                fh.write(rec)

                        cont = input("Do you wish to make another transactions? [n] to stop : ")
                        if cont.lower() == "n":
                            print("Transactions has been recorded in Transactions.txt")
                            l1flag = 1
                            break
                if l1flag ==1:
                    break
            if l1flag ==1:
                break
            if supflag == -1:
                print("Supplier Code not found")
                break
except:
    print("Error format/Data not found")
```

Figure 6 startTransactions() function

The startTransactions function also checking for the availability of data in 2 files which are ppe.txt and suppliers.txt. This function will then ask for the supplier ID as the indicator to start the transactions between suppliers and users. To record the transactions, program will prompt users for the detail of the transactions and write these transactions into the file called transactions.txt

```
def AddPPEdata():
    try:
        print("Please fill the required information with the correct format")
        with open("Assignment\ppe.txt","a") as ppeFile:
            # Keep asking for input until user wish to discontinue
            while True:
                itemname = input("Please enter the PPE item name : ")
                itemcode = input("Please enter the item code : ")
                suppliercode = input("Register / Enter Supplier Code : ")
                quantity = int(100)
                YYYY = int(input("Please enter the year item inputted (YYYY) : "))
                MM = int(input("Please enter the month item inputted (MM) : "))
                # to limit month from 1 to 12 only
                MM = monthLimit(MM)
                DD = int(input("Please enter the date item inputted (DD) : "))
                # setting limitation for days in every month
                DD = dayLimit(YYYY, MM, DD)
                ppeData = itemname+"-"+itemcode+"-"+suppliercode+"-"+str(quantity)+"-"+str(YYYY)+"-"+str(MM)+"-"+str(DD)+"\n"
                ppeFile.write(ppeData)
                cont = input("Press Enter key to continue or [q] to stop : ")
                if cont.lower()=="q":
                    print("PPE data has been recorded to ppe.txt file")
                    break
    except:
        print("Error format/Please Try again")
```

Figure 7 AdddPPEdata()

This function acts the same as the startPPEdata() function however, this function does not rewrite the whole content of the file instead it just adding further details into new lines in the files.

```
def u_code(ind):
    codeList = []
    with open("Assignment\codegate.txt","r") as algocode:
        for code in algocode:
            codeList.append(code.strip().split("-"))
        num = codeList[0][ind]
        numtemp = num[6:]
        numonly = int(numtemp)
        numList = []
        numonly+=1
        numList.append(numonly)
    newcode=[]
    newcode.append(codeList[0][ind][0:6]) # HID989[2]-DID656[3]-TID747[0]
    tempCode = newcode + (numList)
    # newcode = first 6 string  # numList = last digit
    genCode = ""
    for i in range(2):
        genCode = genCode+str(tempCode[i])
    codeList[0][ind] = genCode
    with open("Assignment\codegate.txt","w") as codewriter:
        for code in codeList:
            cd = "-".join(code)
        codewriter.write(cd)
```

Figure 8 u_code(ind) function

This function will change the code of hospitals, distributions, and transactions for each time this function called

```
#READ THE GENERATED CODE
def readID(ind):
    codelist = []
    with open("Assignment\codegate.txt","r") as readsup:
        for code in readsup:
            codelist.append(code.strip().split("-"))
    ID = codelist[0][ind]
    return ID
```

Figure 9 readID(ind) function

This function will read the code from the login.txt file and return it back to the other function that need it data to be written in the respective file.

```
#TO CHECK WHETHER THE ACCOUNT IS REGISTERED OR NOT
def userLogin(file):
    try:
        with open("Assignment\login.txt","r") as readLogin:
            USERLOGDATA = []
            for userData in readLogin:
                USERLOGDATA.append(userData.strip().split("-"))

        for cnt in range (len(USERLOGDATA)):
            if file in USERLOGDATA[cnt][0]:
                userLogin = input("Enter existing User ID : ")
                userPassword = input("Enter Password : ")
                if USERLOGDATA[cnt][1] == userLogin and USERLOGDATA[cnt][2] == userPassword:
                    print("Succesful")
                    return True
                else:
                    print("Login failed")
    except:
        print("Error format/Data not found")
```

Figure 10 userLogin(file) function

This function will check whether the username and password of users existed in the database of login.txt. If it does exist and matched the data inside login.txt, this function will return True to the other function that called it.

```
#FUNCTION TO TRACK AVAILABLE QUANTITY OF ITEMS
def viewInformation():
    ppeInfo = []
    supInfo = []
    combInfo = []
    with open("Assignment\ppe.txt","r") as readPPE:
        for info in readPPE:
            ppeInfo.append(info.strip().split("-"))
    with open("Assignment\suppliers.txt","r") as readSuppliers:
        for info in readSuppliers:
            supInfo.append(info.strip().split("-"))
    allList = []
    for cnt in range(len(supInfo)):
        combInfo = ppeInfo[cnt] + supInfo[cnt]
        allList.append(combInfo)
    # Sorting Ascending
    for x in range(len(allList)-1):
        for y in range(x+1,len(allList)):
            if allList[x][0] > allList[y][0]:
                temp = allList[x]
                allList[x] = allList[y]
                allList[y] = temp
    headerView(allList)
```

Figure 11 viewInformation()

This function will allow users to view directly all the valid items and suppliers' details in one box with sorted ascendingly according to its item code.

```
def ppeview():
    with open("Assignment\ppe.txt") as fh:
        ppe = []
        for data in fh:
            ppe.append(data.strip().split("-"))

        print("="* 75)
        print("Item Name".center(15)+"|"+"Item Code".center(20)+"|"+"Item Qua
        print("="* 75)
        for line in range(len(ppe)):
            print(ppe[line][0].center(15)+"|"+ppe[line][1].center(20)+"|"+ppe
```

Figure 12 ppeview() function

This function allows users to directly view the details that recorded in the ppe.txt file

```
def view25():
    allList = []
    with open("Assignment\ppe.txt","r") as readPPE:
        for info in readPPE:
            allList.append(info.strip().split("-"))

    flag = -1
    for cnt in range(len(allList)):
        quant = int(allList[cnt][3])
        item_name = allList[cnt][0]
        item_ID = allList[cnt][1]
        if quant < 25:
            flag = cnt
            print(item_name+" "+item_ID+" has quantity of "+str(quant)+" boxes left")
    if flag == -1:
        print("All item stock is above 25 boxes")
```

Figure 13 view25() function

This function allows users to view all the items that have quantity less than 25 boxes and notify the users of its current quantity

```
def supplyview():
    with open("Assignment\suppliers.txt") as fh:
        supplier = []
        for data in fh:
            supplier.append(data.strip().split("-"))
        print("="* 95)
        print("Supplier Name".center(15)+"|"+"Supplier Code".center(20)+"|"+'
        print("="* 95)
        for line in range(len(supplier)):
            print(supplier[line][0].center(15)+"|"+supplier[line][1].center(
```

Figure 14 supplyview() function

This function allows users to directly view the details that recorded in the suppliers.txt file

```
def hospitalview():
    with open("Assignment\hospital.txt") as fh:
        hospital = []
        for data in fh:
            hospital.append(data.strip().split("-"))
        print("="* 75)
        print("Hospital Name".center(15)+"|"+"Hospital Code".center(20)+"|"+
        print("="* 75)
        for line in range(len(hospital)):
            print(hospital[line][0].center(15)+"|"+hospital[line][1].center(
```

Figure 17 hospitalview() function

This function allows users to directly view the details that recorded in the hospital.txt file

```
def modifyPPE():
    try:
        data = []
        with open("Assignment\ppe.txt","r") as fh:
            for line in fh:
                elem = line.strip().split("-")
                data.append(elem)
        skey = input("Please enter Item Code : ")
        flg = -1
        for cnt in range(len(data)):
            if skey in data[cnt][1]:
                flg = cnt
                break
        if flg == -1:
            print("Data not found")
        if flg != -1:
            print("1 -Item Name   : "+ data[flg][0])
            print("2 -Item Code     : "+ data[flg][1])
            print("3 -Supplier Code : "+ data[flg][2])
            print("4 -Quantity : "+ data[flg][3])
            print("5- Year : "+ data[flg][4])
            print("6 -Month : "+ data[flg][5])
            print("7 -Date : "+ data[flg][6])
            ans = int(input("Enter the number to modify :"))
            ans = limitopt(ans,7)
            data[cnt][ans-1] = input("Enter a new value: ")
            print("***Succesfully changed***")
            with open("Assignment\ppe.txt","w") as fh:
                for cnt in range(len(data)):
                    rec = "-".join(data[cnt])+"\n"
                    fh.write(rec)
    except:
        print("Error or File not found")
```

Figure 16 modifyPPE() function

```
def modifySup():
    try:
        data = []
        with open("Assignment\suppliers.txt","r") as fh:
            for line in fh:
                rec = line.strip().split("-")
                data.append(rec)
        skey = input("Please enter Supplier Code : ")
        flg = -1
        for cnt in range(len(data)):
            if skey in data[cnt][1]:
                flg = cnt
                break
        if flg == -1:
            print("Data not found")
        if flg != -1:
            print("1 -Supplier Name : "+ data[flg][0])
            print("2 -Supplier Code : "+ data[flg][1])
            print("3 -Item Supplied : "+ data[flg][2])
            print("4 -Supplier Address : "+ data[flg][3])
            print("5 -Quantity Supplied : "+ data[flg][4])
            ans = int(input("Enter the number to modify :"))
            ans = limitopt(ans,5)
            data[cnt][ans-1] = input("Enter a new value: ")
            print("***Succesfully changed***")
            with open("Assignment\suppliers.txt","w") as fh:
                for cnt in range(len(data)):
                    rec = "-".join(data[cnt])+"\n"
                    fh.write(rec)
    except:
        print("Error or File not found")
```

Figure 15 modifySup()

```
def modifyHosp():
    try:
        data = []
        with open("Assignment\hospital.txt","r") as fh:
            for line in fh:
                rec = line.strip().split("-")
                data.append(rec)
        skey = input("Please enter Supplier Code : ")
        flg = -1
        for cnt in range(len(data)):
            if skey in data[cnt][1]:
                flg = cnt
                break
        if flg == -1:
            print("Data not found")
        if flg != -1:
            print("1 -Hospital Name : "+ data[flg][0])
            print("2 -Hospital Code : "+ data[flg][1])
            print("3 -Hospital Address : "+ data[flg][2])
            print("4 -Contact Number : "+ data[flg][3])
            ans = int(input("Enter the number to modify :"))
            ans = limitopt(ans,4)
            data[cnt][ans-1] = input("Enter a new value: ")
            print("***Succesfully changed***")
            with open("Assignment\hospital.txt","w") as fh:
                for cnt in range(len(data)):
                    rec = "-".join(data[cnt])+"\n"
                    fh.write(rec)
    except:
        ("Error or File not found")
```

Figure 18 modifyHosp() function

Figure 15, figure 16 and figure 18 functions allow users to modify the data inside the ppe.txt, suppliers.txt, and hospital.txt file by selecting which data to modify and entering the new value into it.

```
def searchHospDetail():
    try:
        hospeInfo = []
        with open("Assignment\hospital.txt","r") as readfh:
            for info in readfh:
                hospeInfo.append(info.strip().split("-"))
        skey = input("Please enter Hospital Code : ")
        flag = -1
        for line in range (len(hospeInfo)):
            if skey in hospeInfo[line][1]:
                flag = line
                print("="* 75)
                print("Hospital Name".center(15)+"|"+"Hospital Code".center(
                print("="* 75)
                print(hospeInfo[line][0].center(15)+"|"+hospeInfo[line][1].c
                break
        if flag == -1:
            print("Data not found")
    except:
        print("Error format/Data not exist")
```

Figure 19 searchHospitalDetail()

```
def searchPPEDetail():
    try:
        ppeInfo = []
        with open("Assignment\ppe.txt","r") as readPPE:
            for info in readPPE:
                ppeInfo.append(info.strip().split("-"))
        skey = input("Please enter item Code : ")
        flag = -1
        for line in range (len(ppeInfo)):
            if skey in ppeInfo[line][1]:
                flag = line
                print("="* 115)
                print("Item Name".center(15)+"|"+"Item Code".center(20)+"|"+
                print("="* 115)
                print(ppeInfo[line][0].center(15)+"|"+ppeInfo[line][1].cente
                break
        if flag == -1:
            print("Data not found")
    except:
        print("Error format/Data not exist")
```

Figure 20 searchPPEdetail()

```
def searchSupDetail():
    try:
        supInfo = []
        with open("Assignment\suppliers.txt","r") as readSup:
            for info in readSup:
                supInfo.append(info.strip().split("-"))
        skey = input("Please enter supplier Code : ")
        flag = -1
        for line in range (len(supInfo)):
            if skey in supInfo[line][1]:
                flag = line
                print("="* 116)
                print("Supplier Name".center(15)+"|"+"Supplier Code".center(
                print("="* 116)
                print(supInfo[line][0].center(15)+"|"+supInfo[line][1].cente
                break
        if flag == -1:
            print("Data not found")
    except:
        print("Error format/Data not exist")
```

Figure 21 searchSupDetail() function

```
def searchDistribution():
    with open("Assignment\distribution.txt","r") as disfh:
        dislist = []
        for elem in disfh:
            dislist.append(elem.strip().split("-"))

        skey = input("Please enter item Code : ")
        flag = -1
        print("="* 77)
        print("Item Code".center(15)+"|"+"Item Quantity(Box)".center(20)+"|"+
        print("="* 77)
        for line in range (len(dislist)):
            if skey in dislist[line][0]:
                flag = line
                while True:
                    print(dislist[line][0].center(15)+"|"+dislist[line][1].ce
                    break
        if flag == -1:
            print("Data not found")
```

Figure 20 searchDistribution()

Figure 19 – 22 show the functions for searching specific data details in the respective file. Users will tell the program for the code and program will check whether the code existed or not.

```
def monthLimit(MM):
    if MM<0 or MM>12:
        mFlag = -1
        while mFlag == -1:
            MM = int(input("Please enter the month in the proper format (MM)
            if MM>0 and MM<=12:
                mFlag=MM
                break
    return MM
```

Figure 22 montLimit(MM) function

```
def dayLimit(YYYY, MM, DD):
    if MM == 4 or MM == 6 or MM == 9 or MM == 11:
        if DD<0 or DD>30:
            dFlag = -1
            while dFlag == -1:
                DD=int(input("Please enter the date in the proper format : ")
                if DD>0 and DD<=30:
                    dFlag = DD
                    break
    elif YYYY%4==0 and MM==2:
        if DD<0 or DD>29:
            d2Flag = -1
            while d2Flag == -1:
                DD=int(input("Please enter the date with the correct format :
                if DD>0 and DD<=29:
                    d2Flag == DD
                    break
    elif YYYY%4!=0 and MM == 2:
        if DD<0 or DD>28:
            d3Flag = -1
            while d3Flag == -1:
                DD=int(input("Please enter the date with the correct format :
                if DD>0 and DD<=28:
                    d3Flag == DD
                    break
    elif DD<0 or DD>31:
        d4Flag = -1
        while d4Flag == -1:
            DD=int(input("Please enter the date the correct format : "))
            if DD>0 and DD<=31:
                d4Flag = DD
                break
    return DD
```

Figure 24 dayLimit(YYYY,DD,MM) function

Figure 23 and Figure 24 functions will limit the day and months for users when users entered the date in the wrong format. This will fix the users answer by asking for the correct answer repeatedly.

```
def gate(arr,opt,file):
    if arr[opt-1] == 0:
        userSignUp(file)
        arr[opt-1] =1
    else:
        print("*** Account Exist ***")
```

Figure 25 gate() function

This function will change the value in the list of certain to act as the flag of some conditional situation.

```
def limitopt(opt,upper):
    while opt<1 or opt>upper:
        opt = int(input("Enter according to the option available : "))
        if opt in range(1,upper+1):
            break
    return opt
```

Figure 26 limitopt() function

This function limit user's choice from 0 up to the limit that is given according to the situation the function was called.

```
def ALL_MENU():
    try:
        # AFTER INITIAL CREATION FUNC WILL RETURN 1 to flag
        flag = startPPEdata()
    ############     FLAG FOR MENU
        MENU_ARRAY = [1,0,0,0,0]
    ############### PPE,SUP,HOS,DIS,TRAN # WHEN ADMIN CREATED FOR EACH WILL T
        SUB_MENU_FLAG = [0,0,0,0,0]
        while True:
            if flag == 1:
                while True:
                #MENU 1
                    print("1 - Create Admin Account")
                    print("2 - PPE File")
                    print("3 - Suppliers File")
                    print("4 - Hospital File")
                    print("5 - Move to Next Menu")
                    option =int(input("Enter your option : "))
                    option = limitopt(option,5)
                    if option == 1:
                    #SUBMENU1-1
                        while True:
                            print("1. ppe.txt")
                            print("2. suppliers.txt")
                            print("3. hospital.txt")
                            print("4. Distribution")
                            print("5. Transactions")
                            print("6. Move to Last Menu")
                            ask = int(input("Enter your option : "))
                            ask = limitopt(ask,6)
                            if ask == 1:
                                gate(SUB_MENU_FLAG,ask,"ppe")
                            elif ask ==2:
                                gate(SUB_MENU_FLAG,ask,"suppliers")
                            elif ask==3:
                                gate(SUB_MENU_FLAG,ask,"hospital")
                            elif ask==4:
                                gate(SUB_MENU_FLAG,ask,"distribution")
                            elif ask==5:
                                gate(SUB_MENU_FLAG,ask,"Transactions")
                            elif ask==6:
                                break
                    elif option == 2 and SUB_MENU_FLAG[option-2]==0:
                        print("Please Create Admin account!!")

                    elif option == 2 and MENU_ARRAY[option-2]==1 and SUB_MENU
                        if userLogin("ppe")==True:
                            while True:
                                print("1. View Personal Protective Equipment
                                print("2. Search Specific Item")
                                print("3. Modify data in File")
                                print("4. Move to Last Menu")
                                ask = int(input("Enter your option : "))
                                ask = limitopt(ask,4)
                                if ask == 1:
                                    ppeview()
                                elif ask ==2:
                                    searchPPEDetail()
```

```
                    elif ask==4:
                        break

            elif option == 4 and MENU_ARRAY[option-2]==0:
                hosnum=int(input("Enter how many hospital to input :
                startHospital("w",hosnum)
                MENU_ARRAY[option-2]=1

            elif option ==4 and SUB_MENU_FLAG[option-2]==0:
                # WHEN ADMIN ACCOUNT DOES NOT EXISTED
                print("Please Create Admin Account!!")

            elif option ==4 and MENU_ARRAY[option-2]==1 and SUB_MENU
                if userLogin("hospital")==True:
                # SUBMENU1-4
                    while True:
                        print("1. View Hospital Data")
                        print("2. Search Specific Hospital")
                        print("3. Modify data in File")
                        print("4. Move to Last Menu")
                        ask = int(input("Enter your option : "))
                        ask = limitopt(ask,4)
                        if ask ==1:
                            hospitalview()
                        elif ask==2:
                            searchHospDetail()
                        elif ask==3:
                            modifyHosp()
                        elif ask==4:
                            break
        #MOVE TO MENU2
            elif option == 5:
                flag =2
                break

    elif flag == 2:
        while True:
        #MENU 2
            print("1 - Register more PPE Item")
            print("2 - Register more Supplier")
            print("3 - Register more Hospital")
            print("4 - Hospital Distribution") #DISTRIBUTING ITEM TO
            print("5 - Supplier Transaction") #UPDATE QUANTITY WHEN
            print("6 - Inventory Tracking") #TOTAL ITEM & RECORD OF
            print("7 - Distribution List")
            print("8 - Move to Last Menu")
```

```
            print("7 - Distribution List")
            print("8 - Move to Last Menu")
            print("9 - End the Programme")
            option =int(input("Enter your option : "))
            option = limitopt(option,9)
            if option == 1:
                AddPPEdata() # ADDING MORE DATA INTO ppe.txt
            elif option == 2:
                supnum = int(input("Enter how many suppliers to input
                startSupplydata("a",supnum) # ADDING MORE DATA into s
            elif option == 3:
                hosnum=int(input("Enter how many hospital to input :
                startHospital("a",hosnum) # ADDING MORE DATA into hos
            elif option == 4:
                if MENU_ARRAY[option-2]==1:
                    if userLogin("distribution")==True:
                        startDistribution("a")
                else:
                    print("hospital.txt / Admin account does not exis
            elif option == 5:
                if MENU_ARRAY[1]==1:
                    if userLogin("Transactions")==True:
                        startTransactions()
                else:
                    print("suppliers.txt / Admin account does not exi
            elif option == 6:
            #SUBMENU2-6
                while True:
                    print("1. View quantity available for all item")
                    print("2. View all item < 25 boxes")
                    print("3. Move to Last Menu")
                    ask = int(input("Enter your option : "))
                    ask = limitopt(ask,3)
                    if ask == 1:
                        viewInformation()
                    elif ask ==2:
                        view25()
                    elif ask ==3:
                        break
            elif option==7:
            #SUBENU2-7
                while True:
                    print("1. View All Distribution")
                    print("2. Search for specific item distribution")
                    print("3. Move to Last Menu")
                    ask = int(input("Enter your option : "))
                    ask = limitopt(ask,3)
                    if ask ==1:
                        viewDistribution()
```

```
                elif option==7:
                #SUBENU2-7
                    while True:
                        print("1. View All Distribution")
                        print("2. Search for specific item distribution")
                        print("3. Move to Last Menu")
                        ask = int(input("Enter your option : "))
                        ask = limitopt(ask,3)
                        if ask ==1:
                            viewDistribution()
                        elif ask ==2:
                            searchDistribution()
                        elif ask==3:
                            break
                elif option == 8:
                    flag = 1
                    break
                elif option == 9:
                    flag = 3
                    break
            elif flag == 3:
                break

    print("End of programme")
except:
    print("Error format/Data not found")
```

Figure 23 ALL_MENU() function

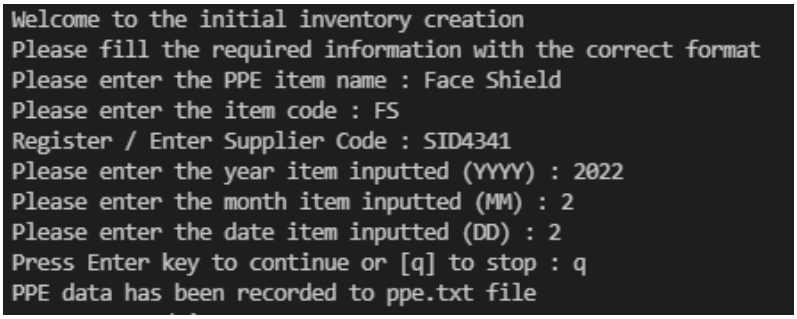
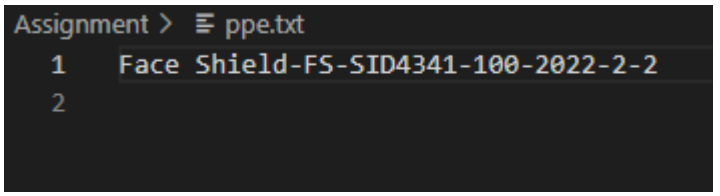
This function will function as the menu for users to click on. Users will be able to run the other functions through the given options in the ALL_MENU function.

```
# MAINLOGIC
ALL_MENU()
# END
```

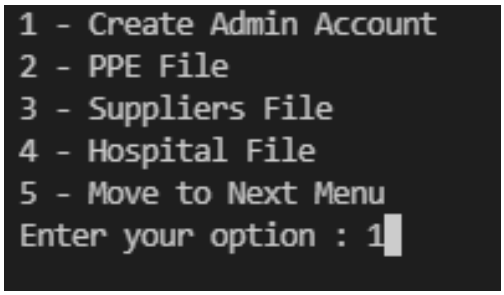
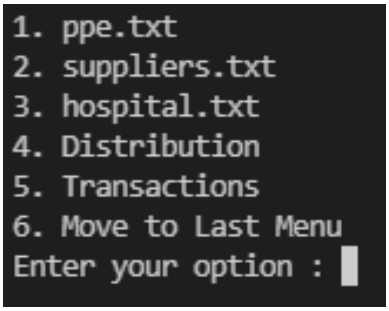
Figure 28 MAIN LOGIC

This is the main logic that called the ALL_MENU() function that will call the other functions.

Sample Input and Output

SAMPLE INPUT	SAMPLE OUTPUT
	

At the start of the program, Initial inventory creation will start by the startPPEdata() function which will ask the users to fill the required information that will recorded in a file called ppe.txt

SAMPLE INPUT	SAMPLE OUTPUT
	

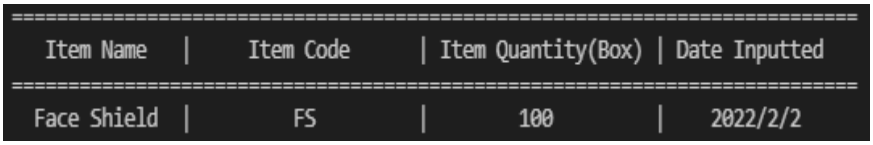
Users may enter their options from one up to five . if users input 1 into the option, another menu will pop up as the output

SAMPLE INPUT	SAMPLE OUTPUT
1. ppe.txt 2. suppliers.txt 3. hospital.txt 4. Distribution 5. Transactions 6. Move to Last Menu Enter your option : 1 Enter your User ID : PPEuser Enter your Password : PPEpass User account has been succesfully saved	Assignment > login.txt 1 ppe-PPEuser-PPEpass 2

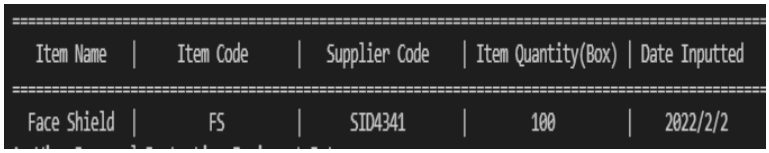
User will enter their preferred User ID and Password that will act as the key to enter another feature

SAMPLE INPUT	SAMPLE OUTPUT
1 - Create Admin Account 2 - PPE File 3 - Suppliers File 4 - Hospital File 5 - Move to Next Menu Enter your option : 2 Enter existing User ID : PPEuser Enter Password : PPEpass Succesful	1. View Personal Protective Equipment Data 2. Search Specific Item 3. Modify data in File 4. Move to Last Menu Enter your option :

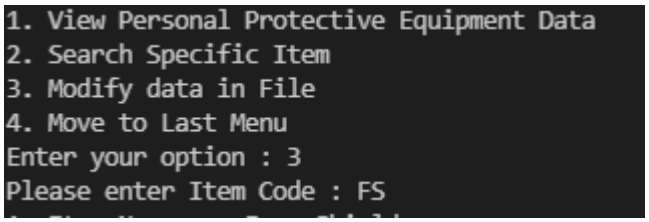
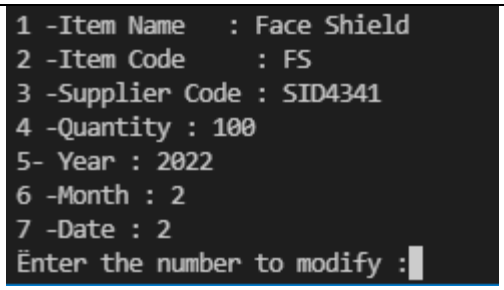
If user enter 2 or 3 or 4, the program will start the respective file creation. If file is already existed, users may get another menu that will ask user about its choices.

SAMPLE INPUT	SAMPLE OUTPUT
1. View Personal Protective Equipment Data 2. Search Specific Item 3. Modify data in File 4. Move to Last Menu Enter your option : 1	 Item Name | Item Code | Item Quantity(Box) | Date Inputted Face Shield | FS | 100 | 2022/2/2

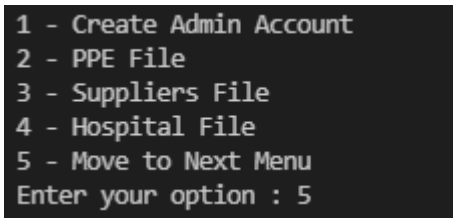
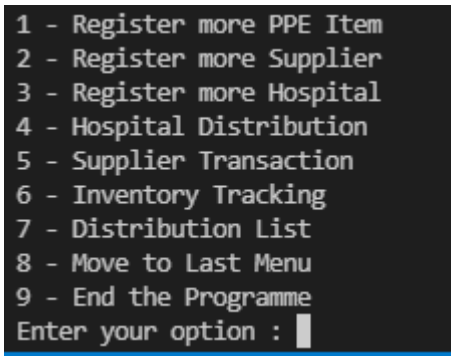
If user entered one into the option, users will be able to view all the data inside the file

SAMPLE INPUT	SAMPLE OUTPUT
1. View Personal Protective Equipment Data 2. Search Specific Item 3. Modify data in File 4. Move to Last Menu Enter your option : 2 Please enter item Code : FS	 Item Name | Item Code | Supplier Code | Item Quantity(Box) | Date Inputted Face Shield | FS | SID4341 | 100 | 2022/2/2

If two was inputted inside the option, program will ask for item code and show the data for that specific item code only.

SAMPLE INPUT	SAMPLE OUTPUT
	

If users inputted three into the option, users will be able to modify the data inside the file by choosing which number to modify and entering its new value.

SAMPLE INPUT	SAMPLE OUTPUT
	

When program receive five as the input for option, the next menu will be in display.

SAMPLE INPUT	SAMPLE OUTPUT
1 - Register more PPE Item 2 - Register more Supplier 3 - Register more Hospital 4 - Hospital Distribution 5 - Supplier Transaction 6 - Inventory Tracking 7 - Distribution List 8 - Move to Last Menu 9 - End the Programme Enter your option : 5 Enter existing User ID : tran Enter Password : tran Succesful Enter supplier ID : SID4341 Item ID is : FS Enter quantity of item added : 15 Transaction ID : TID7471 Do you wish to make another transactions? [n] to stop :	Assignment > Transactions.txt 1 FS-SID4341-15-2022-03-09-TID7471 2 Assignment > suppliers.txt 1 sup-SID4341-Face Shield-sup-115 2 Assignment > ppe.txt 1 Face Shield-FS-SID4341-115-2002-2-2 2

When user want to do transactions with suppliers, user will need to input its user id and password. Afterwards, users need to input the required information properly and when its completed, the quantity in other files will change accordingly.

SAMPLE INPUT	SAMPLE OUTPUT
1 - Register more PPE Item 2 - Register more Supplier 3 - Register more Hospital 4 - Hospital Distribution 5 - Supplier Transaction 6 - Inventory Tracking 7 - Distribution List 8 - Move to Last Menu 9 - End the Programme Enter your option : 9	Enter your option : 9 End of programme

Lastly, if nine inputted into the option, the program will show that it has ended and will not execute anymore code to the users.

Conclusion

In conclusion, creating this inventory management system with python language has been a splendid example of understanding deeper on modular programming technique and on its language itself. This system has showed the capabilities of python language that adapt well in any kinds of situations. This inventory management system may not be perfect for the best use in business situation, but it still able to operate like the best inventory management system example that is made by the professional. Lastly, this system can still improve to gain better features and design for the user best experience while using it.

References

1.) *Flowchart symbols - a complete guide.* Zen Flowchart. (n.d.). Retrieved March 9, 2022, from https://www.zenflowchart.com/flowchart-symbols#:~:text=Off%2Dpage%20Connector%3A%20An%20off,target%20is%20on%20another%20page.&text=11.,process%20block%20is%20usually%20dashed.

2.) *Python datetime*. Python Dates. (n.d.). Retrieved March 9, 2022, from https://www.w3schools.com/python/python_datetime.asp

3.) Bose, S. (2021, February 2). *Coding standards and best practices to follow*. BrowserStack. Retrieved March 1, 2022, from https://www.browserstack.com/guide/coding-standards-best-practices