Company name

Description automatically generated

**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** 3](#_Toc97738829)

[**Assumption** 3](#_Toc97738830)

[**Program Design** 4](#_Toc97738831)

[Pseudocode 4](#_Toc97738832)

[Flowchart 61](#_Toc97738833)

[**Program Source Code** 87](#_Toc97738834)

[**Sample Input and Output** 107](#_Toc97738835)

[**Conclusion** 112](#_Toc97738836)

[**References** 112](#_Toc97738837)

# **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].center(20)+"|"+ppeInfo[line][3].center(20)+"|"+(ppeInfo[line][4]+"/"+ppeInfo[line][5]+"/"+ppeInfo[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**

## Diagram, engineering drawing Description automatically generatedFlowchart

Diagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated

Diagram

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram

Description automatically generated

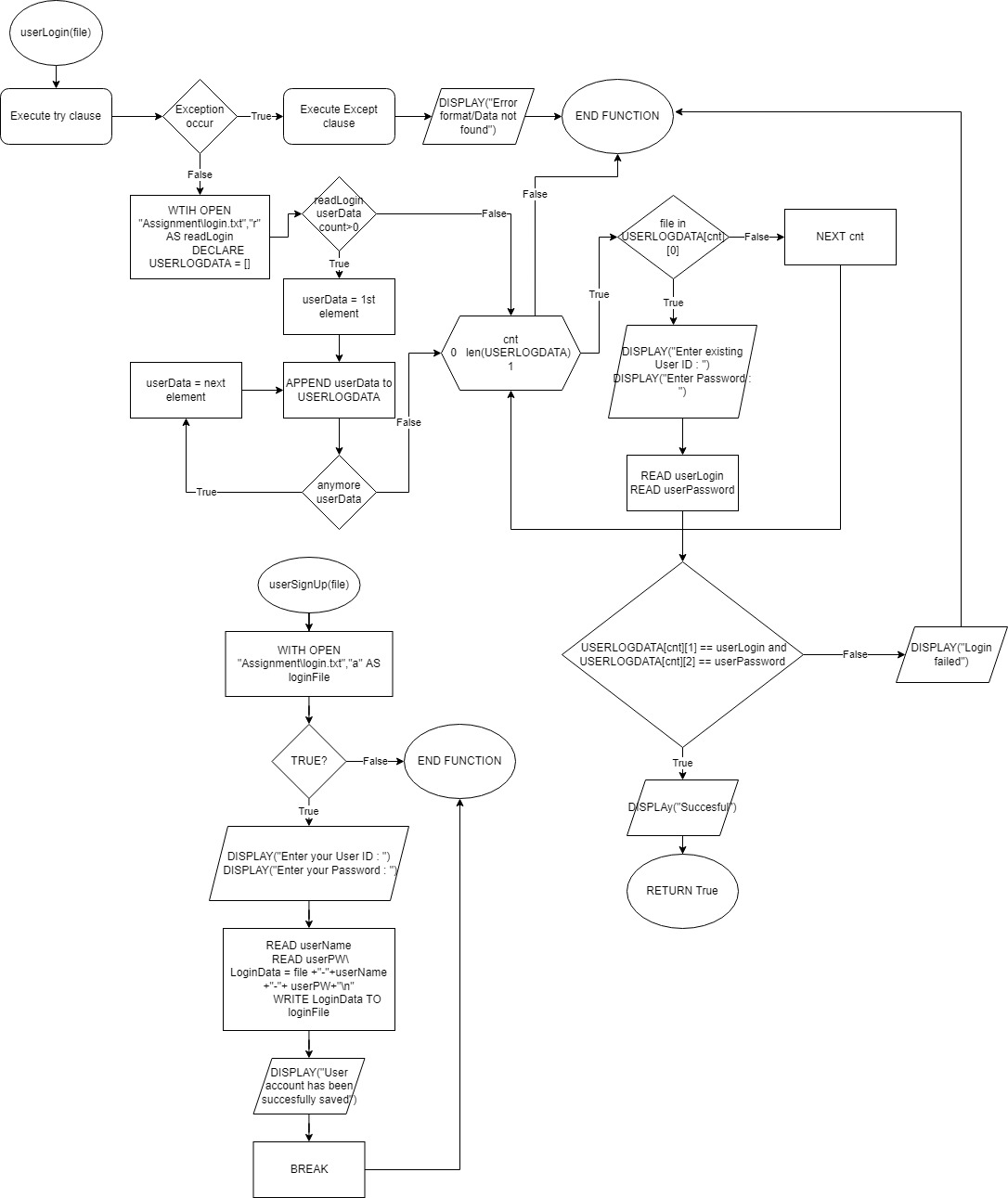
Diagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated



Diagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated

Diagram, engineering drawing

Description automatically generatedDiagram

Description automatically generated

Diagram

Description automatically generatedDiagram, engineering drawing

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated

Diagram

Description automatically generatedDiagram

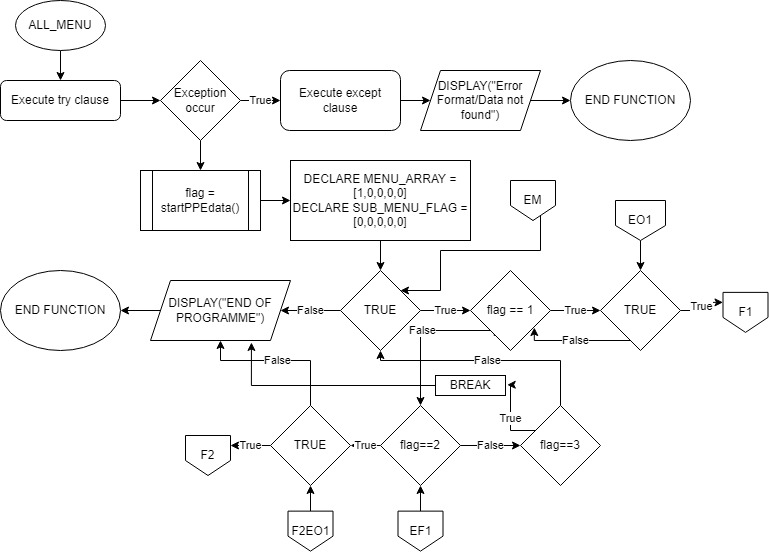
Description automatically generated

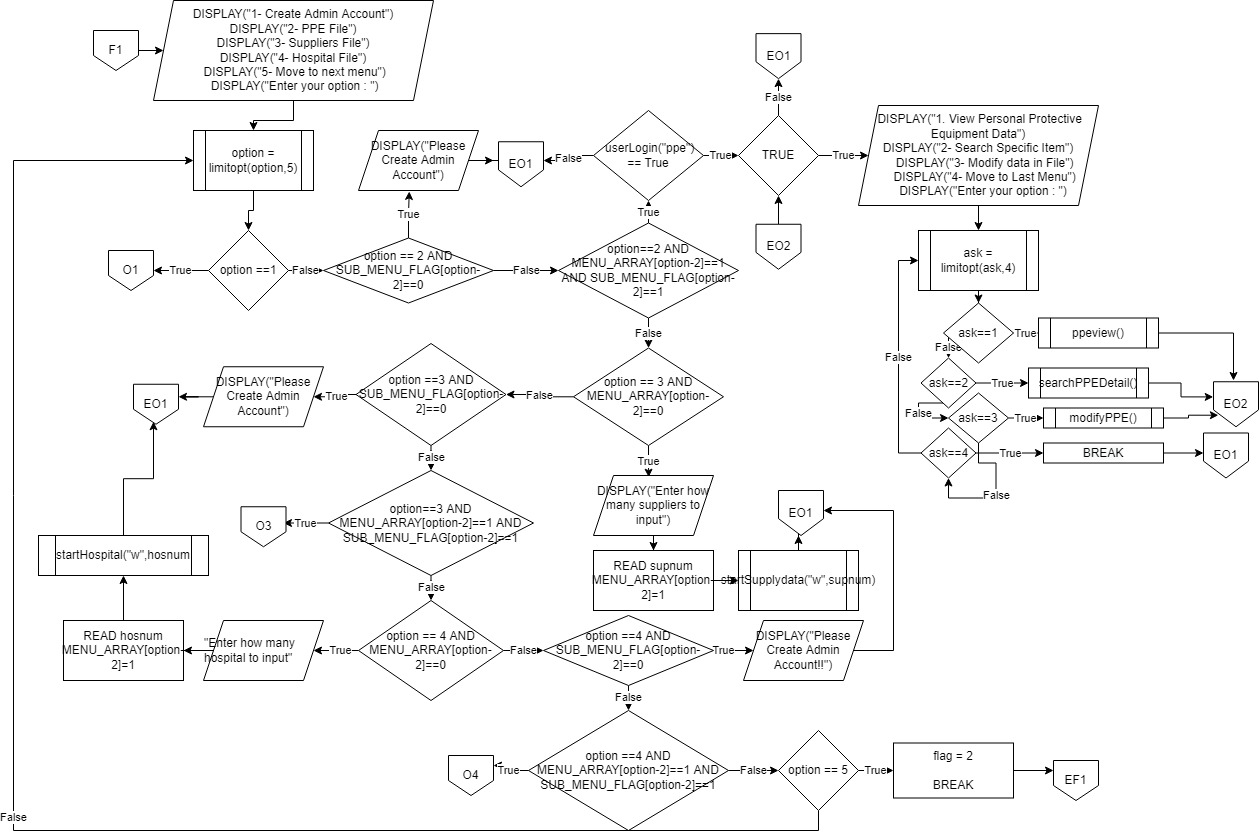
Diagram

Description automatically generatedDiagram

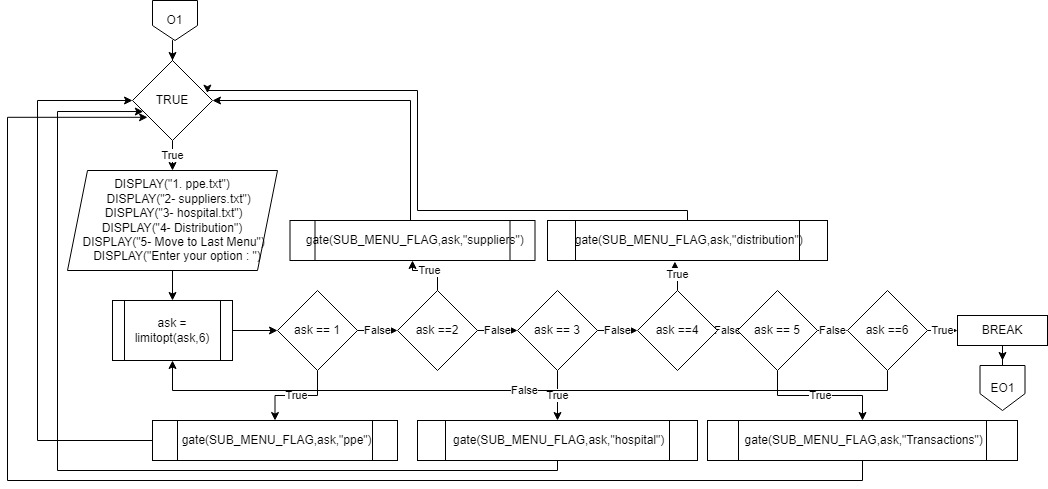
Description automatically generatedDiagram

Description automatically generated





Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

# **Program Source Code**



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.

Text

Description automatically generated

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.

Text

Description automatically generatedText

Description automatically generated

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.

Text

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

Figure 4 startHospital(mode,hosnum) function

Text

Description automatically generatedText

Description automatically generated

Text

Description automatically generatedThis 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.

Figure 5 StartDistribution(mode) function

Text

Description automatically generated

Text

Description automatically generated

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

Text

Description automatically generated

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.

Text

Description automatically generated

Figure 8 u\_code(ind) function

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

Text

Description automatically generated

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.

Text

Description automatically generated

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.

Text

Description automatically generated

Figure 11 viewInformation()

A screenshot of a computer

Description automatically generated with medium confidenceThis 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.

Figure ppeview() function

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

Text

Description automatically generated

Figure 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

Text

Description automatically generated

Figure supplyview() function

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

Text

Description automatically generatedText

Description automatically generatedThis function allows users to directly view the details that recorded in the hospital.txt fileText

Description automatically generated

Figure modifySup()

Figure modifyPPE() function

Figure hospitalview() function

Text

Description automatically generated

Figure 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.

Text

Description automatically generatedText

Description automatically generated

Figure 20 searchPPEdetail()

Figure searchHospitalDetail()

Text

Description automatically generatedText

Description automatically generatedFigure 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.

Figure searchDistribution()

Figure searchSupDetail() function

Text

Description automatically generatedText

Description automatically generated

Figure 24 dayLimit(YYYY,DD,MM) function

Figure montLimit(MM) function

Text

Description automatically generatedFigure 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.

Figure 25 gate() function

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

Text

Description automatically generated

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.

Text

Description automatically generatedText

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Figure 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.

Text

Description automatically generated

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** |
| Text  Description automatically generated | Text  Description automatically generated |

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** |
| Text  Description automatically generated |  |

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

|  |  |
| --- | --- |
| **SAMPLE INPUT** | **SAMPLE OUTPUT** |
|  |  |

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** |
|  |  |

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

|  |  |
| --- | --- |
| **SAMPLE INPUT** | **SAMPLE OUTPUT** |
|  |  |

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** |
|  |  |

|  |  |
| --- | --- |
| **SAMPLE INPUT** | **SAMPLE OUTPUT** |
|  |  |

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.

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