INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT010-3-1-PYP

PYTHON PROGRAMMING

Jordan Lau Jing Hong TP064941

APD1F2106/APU1F2016 –
CE/ME/TE/PE/EEE/CS/CS(CYB)/SE/IS/IT/CS(DF)/MMT/CGD

HAND OUT DATE: **5TH JULY 2021**

HAND IN DATE: **6TH AUGUST 2021**

WEIGHTAGE: **100%**

INSTRUCTIONS TO CANDIDATES:

1. **Submit your assignment online in Moodle Folder 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 and Assumptions

Coronavirus or severe acute syndrome coronavirus 2 (SARS-CoV-2) is a deadly virus that has caused global pandemic. To overcome this deadly virus, public has to go through vaccination process in order to obtain immunity towards the virus. And to make the process easier, this program is designing to handle the process of vaccination. The function of the program includes new patient registration, search of patient record and vaccination status and statistical information on patients vaccinated. It is assumed that patient will register themselves into the system using this program first. After that, patient will either go receive their first dose directly or go home and come back to receive their vaccine when they are free. After they have received first dose, they will go to the staff and the time and date will be record in the system and second dose date will be calculated and shown if second dose is available, their vaccination status will also be update to completed dose 1 or all dose complete if there is only one dose needed for the vaccine they have chosen. The second dose process is similar, but it is just about update the patient's vaccination stage to complete all dose.

Design of the Program

Pseudocode:

```
DEFINE Pmain_menu()
    DOWHILE (True)
        DISPLAY ("1 - New patient registration")
        DISPLAY ("2 - Patient login")
        DISPLAY ("3 - Staff login")
        DISPLAY ("4 - Exit")
        DISPLAY ("Enter your choice: ")
        READ choice
        IF choice == 1 THEN
            CALL: new_patient_reg
        ELIF choice == 2 THEN
            CALL: patient_login
        ELIF choice == 3 THEN
            CALL: staff_login
        ELIF choice == 4 THEN
            EXIT ()
        ELSE
            DISPLAY ("Invalid option")
        ENDIF
    ENDDO
ENDDEFINE
```

```
DEFINE Smenu()
    DOWHILE (True)
        DISPLAY ("1 - Vaccine Administration")
        DISPLAY ("2 - Search Patient Record and Vaccination Status")
        DISPLAY ("3 - Statistical Information on Patients Vaccinated")
        DISPLAY ("4 - Exit")
        DISPLAY ("Enter your choice: ")
        READ choice
        IF choice == 1 THEN
            CALL: vc_admin
        ELIF choice == 2 THEN
            CALL: vc_admin
        ELIF choice == 3 THEN
            CALL: vc_statusview
        ELIF choice == 4 THEN
            EXIT (Pmain_menu)
        ENDIF
    ENDDO
ENDDEFINE



DEFINE new_patient_reg()
    fh = OPEN("patient.txt", "a")
```

```
DEFINE rec, rec2, flg
DOWHILE (True)
        DISPLAY ("Enter new patient name: ")
        READ name
        IF LENGTH(name) == 0 THEN
                DISPLAY ("Please input valid name.")
        ELSE
                BREAK
        ENDIF
ENDDO
DOWHILE (True)
        DISPLAY ("Enter new patient age: ")
        TRY
                READ int(age)
        EXCEPT
                DISPLAY ("Invalid input")
        ENDTRY
ENDDO
IF age<12 THEN
        DISPLAY("Sorry, no vaccine available for this age")
        EXIT (Pmain_menu())
ELIF age>=18 AND age<=45 THEN
        flg = 1
```

```
ELIF age>=12 AND age<=45 THEN
        flg = 2
ELIF age>=45 THEN
        flg = 3
ENDIF
IF flg == 1 THEN
        DISPLAY ("Vaccine available: ")
        DISPLAY ("="*65)
        DISPLAY ("Vaccine Code".center(14)+"|"+"Dosage
        Required".center(24)+"|"+"Interval Between Doses".center(24))
        DISPLAY ("="*65)
        DISPLAY ("AF".ljust(14)+"|"+"2".center(24)+"|"+"2 weeks (14 days)".center(24))
        DISPLAY ("BV".ljust(14)+"|"+"2".center(24)+"|"+"3 weeks (21 days)".center(24))
        DISPLAY ("CZ".ljust(14)+"|"+"2".center(24)+"|"+"3 weeks (21 days)".center(24))
        DISPLAY ("DM".ljust(14)+"|"+"2".center(24)+"|"+"4 weeks (28 days)".center(24))
        DISPLAY ("EC".ljust(14)+"|"+"1".center(24)+"|"+"-".center(24))
        DISPLAY ("="*65)
ELIF flg == 2 THEN
        DISPLAY ("Vaccine available: ")
        DISPLAY ("="*65)
        DISPLAY ("Vaccine Code".center(14)+"|"+"Dosage
        Required".center(24)+"|"+"Interval Between Doses".center(24))
        DISPLAY ("="*65)
```

```
        DISPLAY ("AF".ljust(14)+"|"+"2".center(24)+"|"+"2 weeks (14 days)".center(24))
        DISPLAY ("CZ".ljust(14)+"|"+"2".center(24)+"|"+"3 weeks (21 days)".center(24))
        DISPLAY ("DM".ljust(14)+"|"+"2".center(24)+"|"+"4 weeks (28 days)".center(24))
        DISPLAY ("="*65)
ELIF flg == 3 THEN
        DISPLAY ("Vaccine available: ")
        DISPLAY ("="*65)
        DISPLAY ("Vaccine Code".center(14)+"|"+"Dosage
        Required".center(24)+"|"+"Interval Between Doses".center(24))
        DISPLAY ("="*65)
        DISPLAY ("AF".ljust(14)+"|"+"2".center(24)+"|"+"2 weeks (14 days)".center(24))
        DISPLAY ("BV".ljust(14)+"|"+"2".center(24)+"|"+"3 weeks (21 days)".center(24))
        DISPLAY ("DM".ljust(14)+"|"+"2".center(24)+"|"+"4 weeks (28 days)".center(24))
        DISPLAY ("EC".ljust(14)+"|"+"1".center(24)+"|"+"-".center(24))
        DISPLAY ("="*65)
ENDIF
DOWHILE (True)
        DISPLAY ("Enter vaccine you wish to take: ")
        READ vc
        IF flg == 1 AND vc == AF OR BV OR CZ OR DM OR EC THEN
                BREAK
        ELIF flg == 2 AND vc == AF OR CZ OR DM THEN
                BREAK
```

```
        ELIF flg == 3 AND vc == AF OR BC OR DM OR EC THEN
                BREAK
        ELSE
                DISPLAY ("Invalid choice")
        ENDIF
ENDDO
DISPLAY ("Vaccine successfully register")
DOWHILE (True)
        DISPLAY ("Enter vaccine centre you wish to go(VC1 or VC2): ")
        READ vc_location
        IF vc_location == VC1 OR VC2 THEN
                DISPLAY ("Location successfully register")
                BREAK
        ELSE
                DISPLAY ("Invalid input")
        ENDIF
        IF vc_location == VC1 THEN
                p_id = gen_new_id("VCA")
        ELIF vc_location == VC2 THEN
                p_id = gen_new_id("VCB")
        ENDIF
ENDDO
DISPLAY ("Patient ID: ",p_id)
```

```
DOWHILE (True)
        DISPLAY ("Enter contact number: ")
        TRY
                READ int(p_number)
                BREAK
        EXCEPY
                DISPLAY ("Invalid input")
        ENDTRY
ENDDO
DOWHILE (True)
        DISPLAY ("Enter password for your account: ")
        READ p_pass
        DISPLAY ("Reenter your password: ")
        READ p_pass2
        IF p_pass == p_pass2 THEN
                DISPLAY ("Password successfully register")
                BREAK
        ELSE
                DISPLAY ("Password does not match first password")
        ENDIF
ENDDO
v_stage = 0
APPEND (p_id, p_pass, name, age, p_number, vc_location, vc) TO rec
```

```
        WRITE rec INTO patients.txt
        APPEND (p_id, vc, v_stage) TO rec2
        fh = OPEN("vaccination.txt", "a")
        WRITE rec2 INTO vaccination.txt
        ENDDO
ENDDEFINE



DEFINE gen_new_id(eid)
        fh = OPEN("id.txt", "r")
        rec = fh
        IF eid == VCA THEN
                ind = 0
        ELIF eif = VCB
                ind = 1
        ENDIF
        mylist = rec
        nextid = mylist[ind]
        IF LENGTH(newid) == 1 THEN
                nextid = nextid[:3] + "0000" + newid
        ELIF LENGTH(nextid) == 2 THEN
                nextid = nextid[:3] + "000" + newid
        ELIF LENGTH(nextid) == 3 THEN
```

```
            nextid = nextid[:3] + "00" + newid
        ELIF LENGTH(nextid) == 4 THEN
            nextid = nextid[:3] + "0" + newid
        ELIF LENGTH(nextid) == 5 THEN
            nextid = nextid[:3] + newid
        ENDIF
        mylist[ind] = nextid
        rec = mylist
        fh = OPEN("id.txt", "w")
        WRITE rec INTO id.txt
        RETURN (newid)
ENDDEFINE


DEFINE Pmod_date(PID)
        fh = OPEN("patient.txt", r")
        DEFINE allrec
        LOOP line IN fh
            mylist = fh
            APPEND mylist INTO allrec
        ENDLOOP
        DOWHILE
            DISPLAY ("1 - Patient Name")
            DISPLAY ("2 - Contact number")
            DISPLAY ("3 - Exit")
```

```
READ choice
IF choice == 1 THEN
        DISPLAY ("Current name: ", allrec[PID][2])
        DOWHILE (True)
                DISPLAY ("Enter new name: ")
                READ new_name
                IF LENGTH(new_name) == 0 THEN
                        DISPLAY ("Please input valid name.")
                ELSE
                        BREAK
                ENDIF
        ENDDO
        DOWHILE (True)
                DISPLAY ("Current name: ", allrec[PID][2])
                DISPLAY ("New name: ", new_name)
                DISPLAY ("Comfirm (Y/N): ")
                READ d_com
                IF d_com == Y THEN
                        allrec[PID][2] = new_name
                        fh = OPEN("patient.txt", "W")
                        WRITE allrec INTO patient.txt
                        BREAK
                ELIF d_com == N THEN
                        DISPLAY ("No change made")
                        BREAK
                ELSE
                        DISPLAY ("Invalid input")
                ENDIF
```

```
        ENDDO
ELIF choice == 2 THEN
        DOWHILE (True)
                DISPLAY ("Current contact number: ", allrec[PID][4])
                TRY
                        READ int(p_number)
                EXCEPT
                        DISPLAY ("Invalid input")
                ENDTRY
        ENDDO
        DOWHILE (True)
                DISPLAY ("Current contact number: ", allrec[PID][4])
                DISPLAY ("New contact number: ", p_number)
                DISPLAY ("Comfirm (Y/N): ")
                READ d_com
                IF d_com == Y/N THEN
                        allrec[PID][4] = str(p_number)
                        fh = OPEN("patient.txt", "w")
                        WRITE allrec INTO patient.txt
                        BREAK
                ELIF d_com == N THEN
                        DISPLAY ("No change made")
                        BREAK
                ELSE
                        DISPLAY ("Invalid input")
                ENDIF
        ENDDO
ELIF choice == 3 THEN
```

```
                BREAK
            ELSE
                DISPLAY ("Invalid input")
            ENDIF
        ENDWHILE
ENDDEFINE


DEFINE vc_statusview()
        fh = OPEN("vaccination.txt", "r")
        DEFINE allrec
        LOOP line IN fh
            mylist = fh
            APPEND mylist INTO allrec
        ENDLOOP
        no_of_rec = LENGTH(allrec)
        DEFINE af, bv, cz, dm, ec, d0, d1, d2
        LOOP i IN RANGE(no_of_rec)
            IF allrec[i][1] == AF THEN
                af = af + 1
            ELIF allrec[i][1] == BV THEN
                bv = bv + 1
            ELIF allrec[i][1] == CZ THEN
                cz = cz + 1
            ELIF allrec[i][1] == DM THEN
                dm = dm + 1
            ELIF allrec[i][1] == EC THEN
                ec = ec + 1
```

```
            ENDIF
            IF allrec[i][2] == 0 THEN
                d0 = d0 + 1
            ELIF allrec[i][2] == 1 THEN
                d1 = d1 + 1
            ELIF allrec[i][2] == 2 THEN
                d2 = d2 + 1
            ENDIF
        ENDLOOP
        DISPLAY ("Total number of patient: ", no_of_rec)
        DISPLAY ("Total number of patient who haven't receive any dose: ", d0)
        DISPLAY ("Total number of patient waiting for dose 2: ", d1)
        DISPLAY ("Total number of patient who completed vaccination: ", d2)
        DISPLAY ("Total number of patient choose AF: ", af)
        DISPLAY ("Total number of patient choose BV: ", bv)
        DISPLAY ("Total number of patient choose CZ: ", cz)
        DISPLAY ("Total number of patient choose DM: ", dm)
        DISPLAY ("Total number of patient choose EC: ", ec)
ENDDEFINE


DEFINE vc_admin()
        IMPORT date, datetime, timedelta FROM datetime
        fh = OPEN("patients.txt", "r")
        DEFINE allrec
        LOOP line IN fh
            mylist = fh
            APPEND mylist INTO allrec
```

```
ENDLOOP
fh2 = OPEN("vaccination.txt", "r")
DEFINE allrec2
LOOP line in fh2
        mylist2 = fh2
        APPEND mylist2 INTO allrec2
ENDLOOP
no_of_rec = LENGTH(allrec)
no_of_rec2 = LENGTH(allrec2)
DEFINE flg, flg2, flg3, flg4
DOWHILE (True)
        DISPLAY ("Enter patient ID to search (Type exit to exit): ")
        READ search
        LOOP i IN RANGE(no_of_rec)
                IF search == allrec[i][0] THEN
                j = i
                flg = 1
        ENDLOOP
        IF flg == 1 THEN
                LOOP k IN RANGE(no_of_rec2)
                IF search == allrec2[k][0]
                l = k
                ENDLOOP
                DOWHILE (True)
                        DISPLAY ("Patient ID: ", allrec[j][0])
                        DISPLAY ("Name: ", allrec[j][2])
                        DISPLAY ("Age: ", allrec[j][3])
                        DISPLAY ("Vaccine centre choosen: ", allrec[j][5])
```

```
DISPLAY ("Vaccine choosen ", allrec[j][6])
TRY
        DISPLAY ("Vaccine time: ",allrec2[l][3])
EXCEPT
        DISPLAY ("Patient haven't been vaccine")
ENDTRY
TRY
        DISPLAY ("Dose1 date: ",allrec2[l][4])
EXCEPT
        PASS
ENDTRY
TRY
        DISPLAY ("Dose2 date: ",allrec2[l][5])
EXCEPT
        PASS
ENDTRY
IF allrec2[l][1] == EC THEN
        flg2 = 1
ENDIF
IF allrec2[l][2] == 0 THEN
        DISPLAY ("Pending for dose")
        flg4 = 1
ELIF allrec[l][2] == 1 THEN
        DISPLAY ("Pending for second dose")
ELIF allrec[1][2] == 2 THEN
        DISPLAY ("All dose complete!")
        EXIT(Smenu())
ENDIF
```

```
DISPLAY ("1 - Sign up for vaccination")
DISPLAY ("2 - Exit")
DISPLAY ("Enter your choice: ")
READ choice
IF choice == 1 THEN
        DISPLAY ("Dose complete?")
        DOWHILE (True)
                DISPLAY ("Y/N: ")
                READ vc_status
                IF vc_status == Y THEN
                        flg3 = 1
                        IF flg2 == 0 THEN
                                allrec2[l][2] = str(int(allrec2[l][2])+1)
                        ELIF flg2 == 1 THEN
                                allrec2[l][2] = str(2)
                        BREAK
                        ENDIF
                ELIF vc_status == N THEN
                        DISPLAY ("No change is made")
                        flg3 = 0
                        BREAK
                ELSE
                        DISPLAY ("Invalid input")
                ENDIF
        ENDDO
        IF flg4 == 1 AND flg3 == 1 THEN
                vc = allrec[j][6]
                d_time = GET(now time FROM datetime)
```

```
                        d1_date = GET(now date FROM date)
                        IF vc == AF THEN
                            d2_date = d1_date + timedelta(days=14)
                        ELIF vc = BV OR CZ THEN
                            d2_date = d1_date + timedelta(days=21)
                        ELIF vc == DM THEN
                            d2_date = d1_date + timedelta(days=28)
                        ENDIF
                        TRY
                            APPEND d2_date INTO allrec2[l]
                        EXCEPT
                            PASS
                        ENDTRY
                    ENDIF
                    fh2 = OPEN("vaccination.txt", "w")
                    LOOP line IN allrec2
                        WRITE line INTO vc vaccination.txt
                    ENDLOOP
                    DISPLAY ("Vaccine status had been update")
                    EXIT(Smenu())
                ELIF choice == 2 THEN
                    EXIT(Smenu())
                ELSE
                    DISPLAY ("Invalid choice")
                ENDIF
            ENDDO
        ENDIF
        IF search == exit THEN
```

```
                EXIT(Smenu())
            ELSE
                DISPLAY ("Patient not found")
            ENDIF
        ENDDO
ENDDEFINE


DEFINE patient_login()
        DEFINE allrec, allrec2
        fh = OPEN("patients.txt", "r")
        LOOP line IN fh
            mylist = fh
            APPEND mylist INTO allrec
        ENDLOOP
        fh = OPEN("vaccination.txt", "r")
        LOOP line IN fh
            mylist2 = fh
            APPEND mylist2 INTO allrec2
        ENDLOOP
        no_of_user = LENGTH(allrec)
        no_of_user2 = LENGTH(allrec2)
        DOWHILE (True)
            flg_valid = 0
            DISPLAY ("Enter patient ID: ")
            READ user
            DISPLAY ("Enter password: ")
            READ password
```

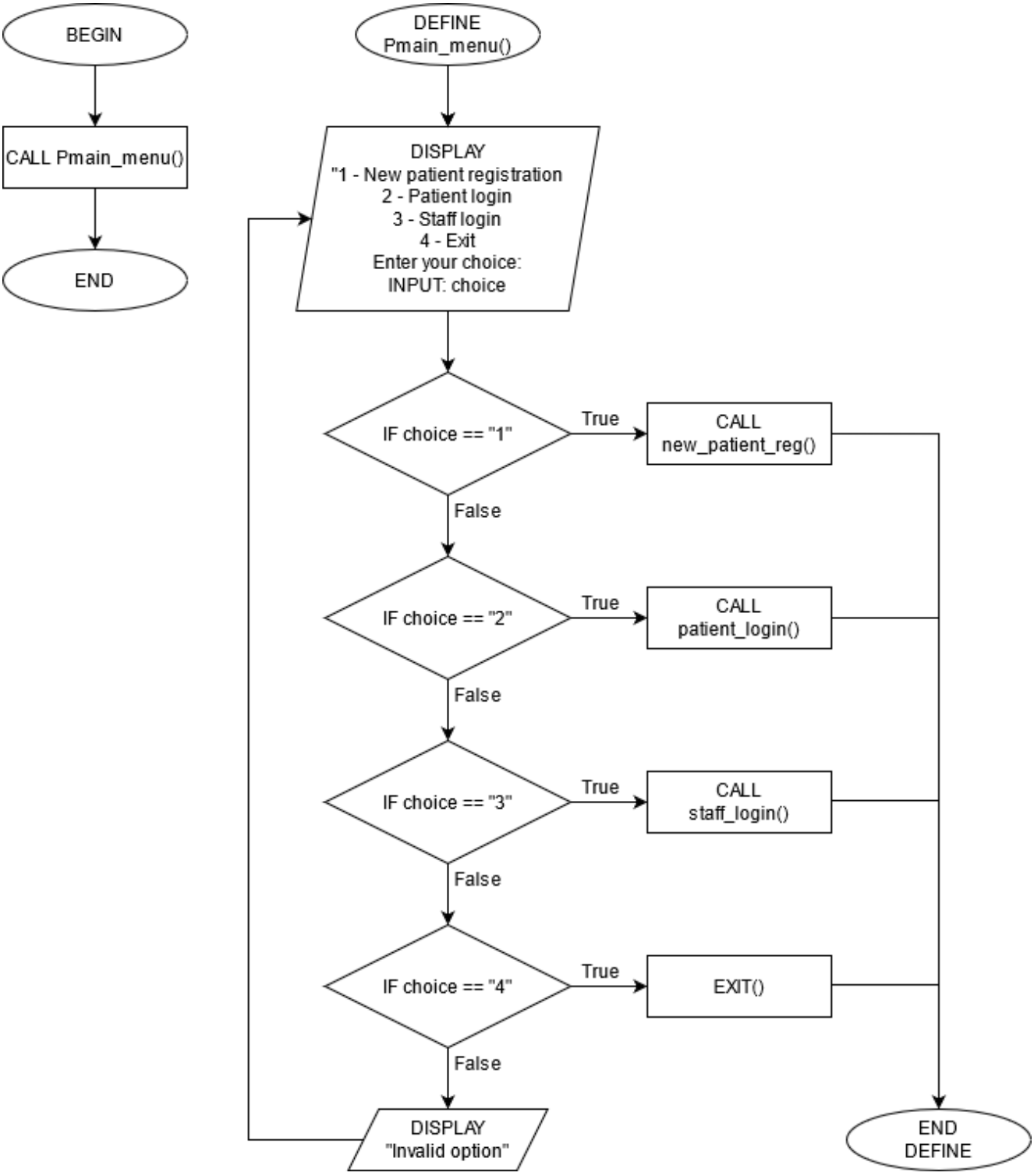
```
LOOP i IN RANGE(no_of_user)
        IF user == allrec[i][0] AND password == allrec[i][1] THEN
                flg_valid = 1
                j = 1
        ENDIF
ENDLOOP
IF flg_valid == 1 THEN
        LOOP k IN RANGE(no_of_user2)
                IF user == allrec2[k][0] THEN
                        l = k
                ELSE
                        DISPLAY ("Patient not found in vaccination.txt.")
                        DISPLAY ("Please contact staff.")
                        EXIT(Pmain_menu())
                ENDIF
        ENDLOOP
        DOWHILE (True)
                DISPLAY ("1 - Modify data")
                DISPLAY ("2 - View current status")
                DISPLAY ("3 - Exit (To refresh current data)")
                DISPLAY ("Enter your choice: ")
                READ choice
                IF choice == 1 THEN
                        CALL: Pmod_data(j)
                ELIF chice == 2 THEN
                        DISPLAY ("Patient ID: ", allrec[j][0])
                        DISPLAY ("Name: ", allrec[j][2])
                        DISPLAY ("Vaccine centre choosen: ", allrec[j][5])
```

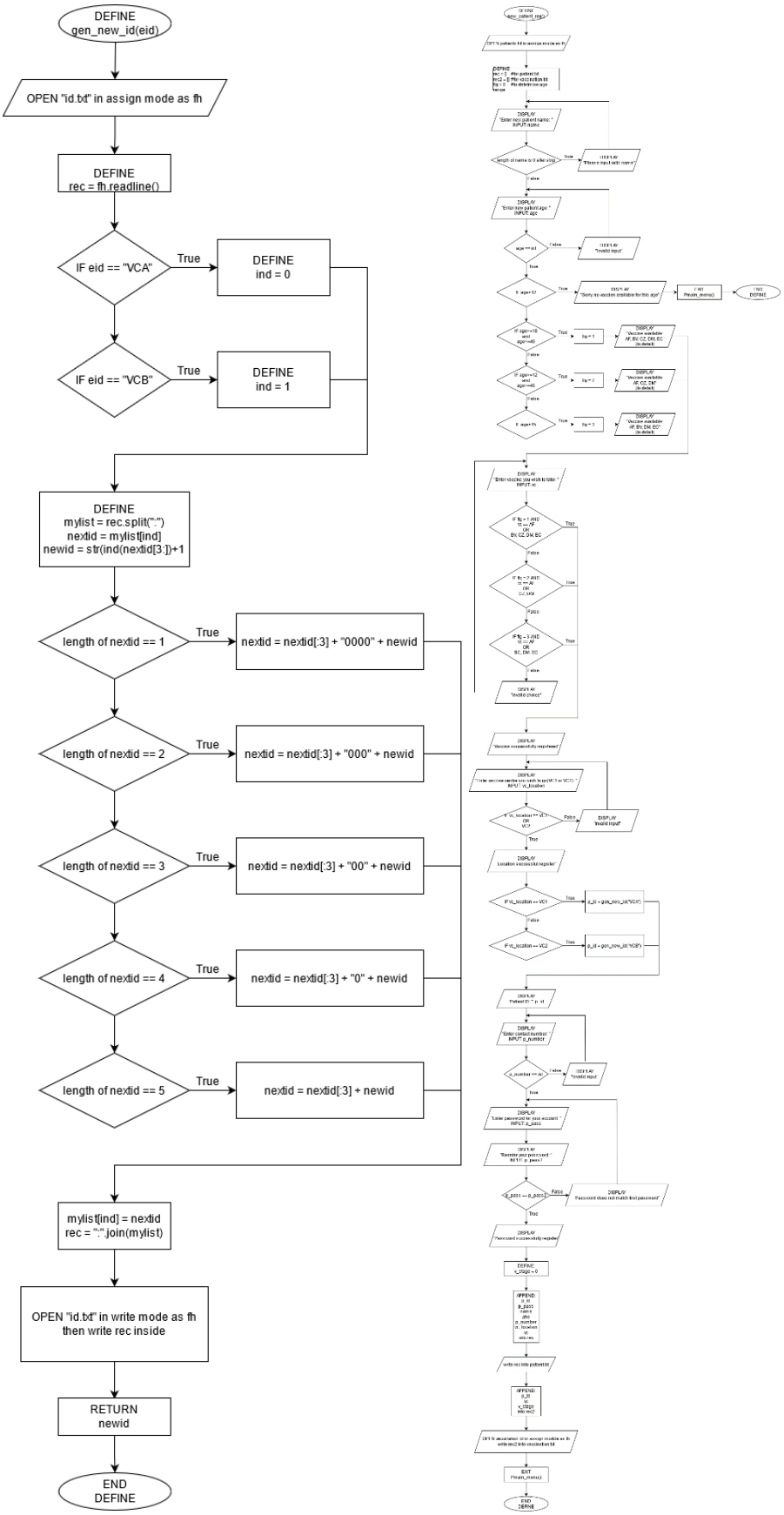
```
            DISPLAY ("Vaccine choosen: ", allrec[j][6])
            IF allrec2[l][2] == 0 THEN
                DISPLAY ("Vaccine status: Pending for dose")
            ELIF allrec2[l][2] == 1 THEN
                DISPLAY ("Vaccine status: Dose 1 complete, pending
                for dose 2")
            ELIF allrec2[l][2] == 2 THEN
                DISPLAY ("Vaccine status: All dose complete.")
            ENDIF
            TRY
                DISPLAY ("Vaccine time: ",allrec2[l][3])
            EXCEPT
                PASS
            ENDTRY
            TRY
                DISPLAY ("Dose1 date: ",allrec2[l][4])
            EXCEPT
                DISPLAY ("Patient haven't been vaccine")
            ENDTRY
            TRY
                DISPLAY ("Dose2 date: ",allrec2[l][5])
            EXCEPT
                PASS
            ENDTRY
    ELIF chice == 3 THEN
        EXIT(Pmain_menu())
    ELSE
        DISPLAY ("Invalid choice")
    ENDIF
```

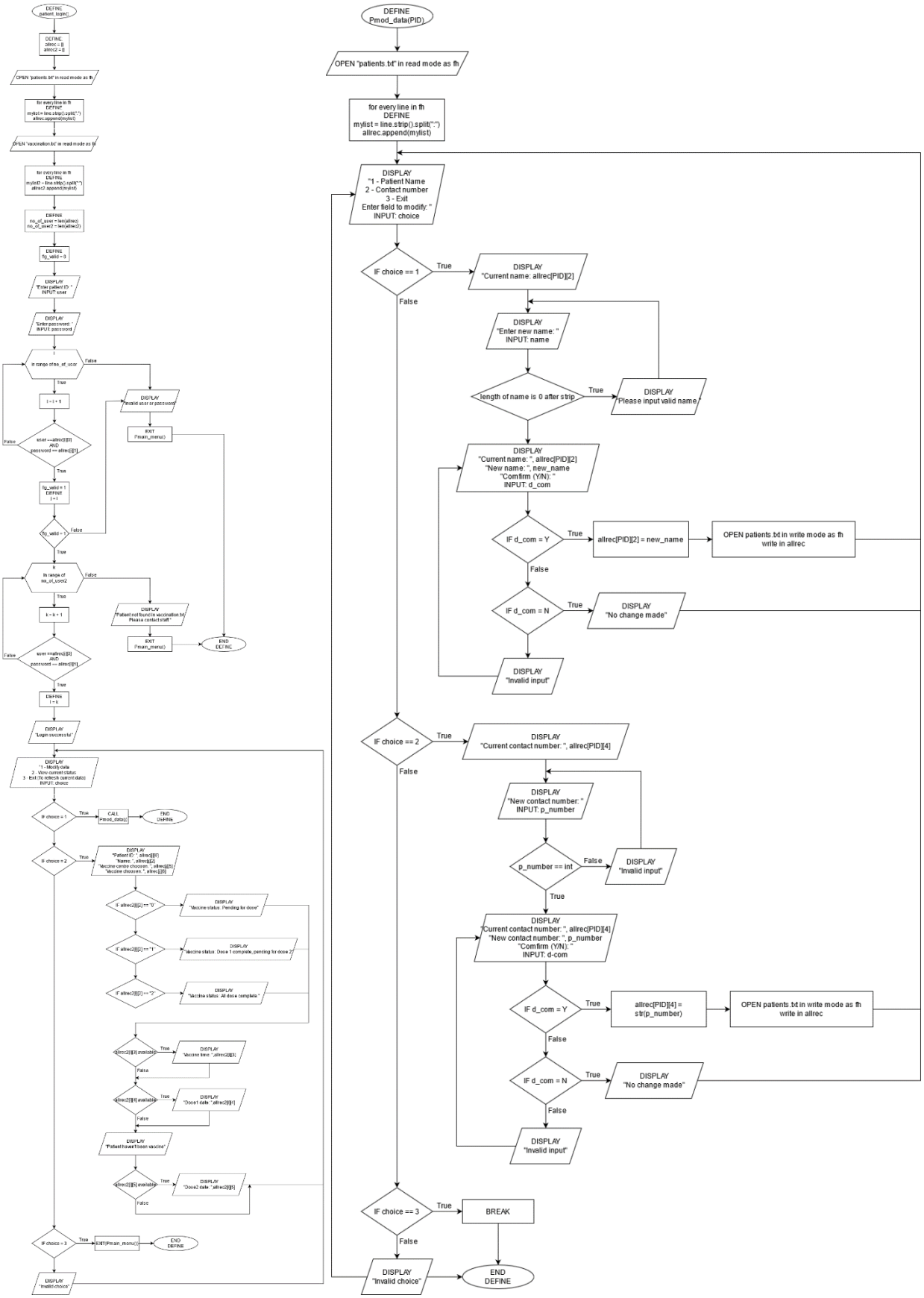
```
            ELSE
                DISPLAY ("Invalid user or password")
                EXIT(Pmain_menu())
            ENDIF
        ENDDO
ENDDEFINE


DEFINE staff_login()
        DEFINE allrec
        fh = OPEN("staff.txt", "r")
        LOOP line IN fh
            mylist = fh
            APPEND mylist INTO allrec
        ENDLOOP
        no_of_user = LENGTH(allrec)
        DOWHILE (True)
            flg_valid = 0
            DISPLAY ("Enter username: ")
            READ user
            DISPLAY ("Enter password: ")
            READ password
            LOOP i IN RANGE(no_of_user)
                IF user == allrec[i][1] AND password == allrec[i][2] THEN
                    flg_valid = 1
                ENDIF
            ENDLOOP
            IF flg_valid == 1 THEN
```

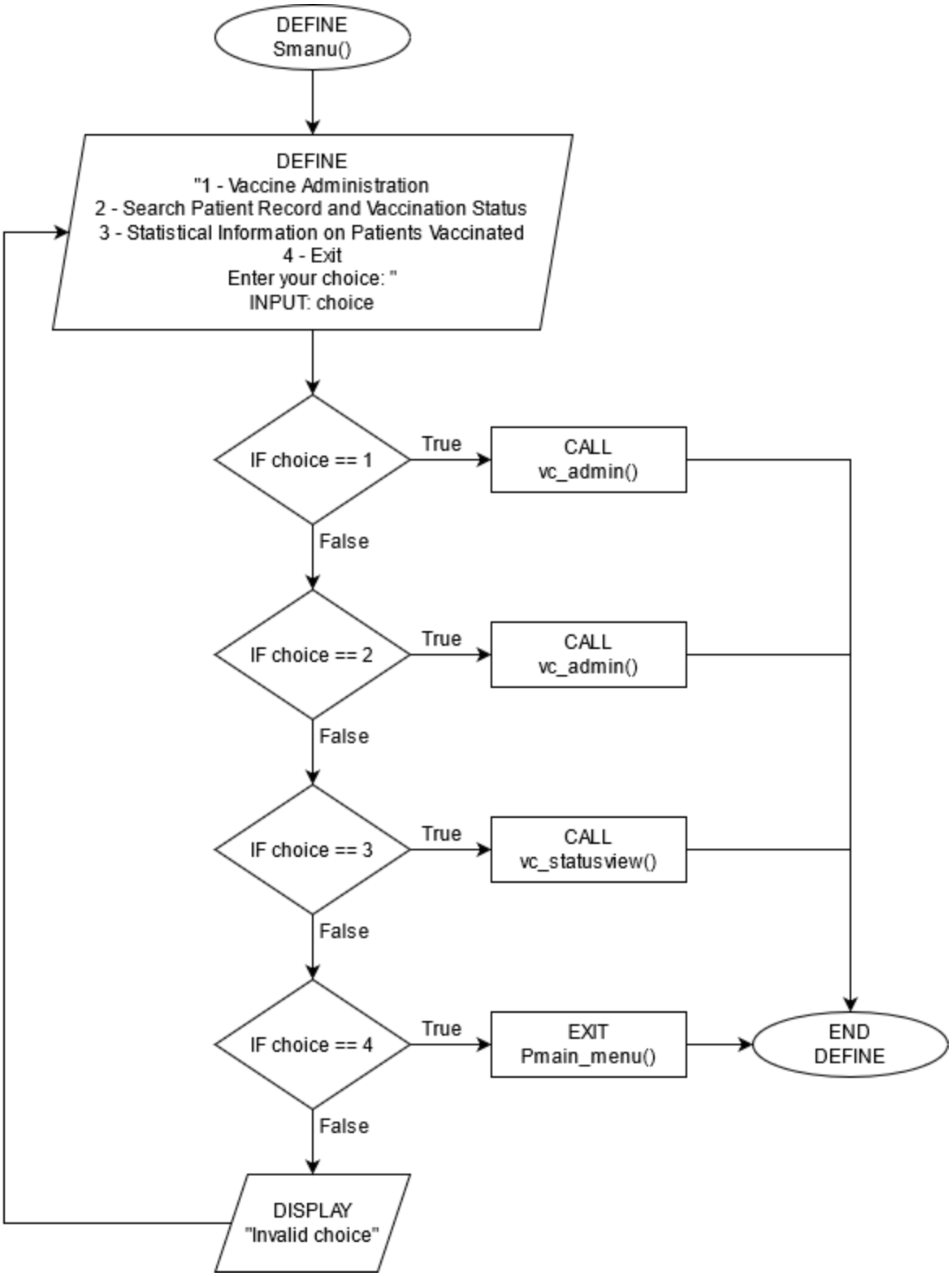
```
                DISPLAY ("Login successful")
                CAll: Smenu()
        ELSE
                DISPLAY ("Invalid user or password")
                EXIT(Pmain_menu())
ENDDEFINE


BEGIN
    CALL: Pmain_menu()
END
```

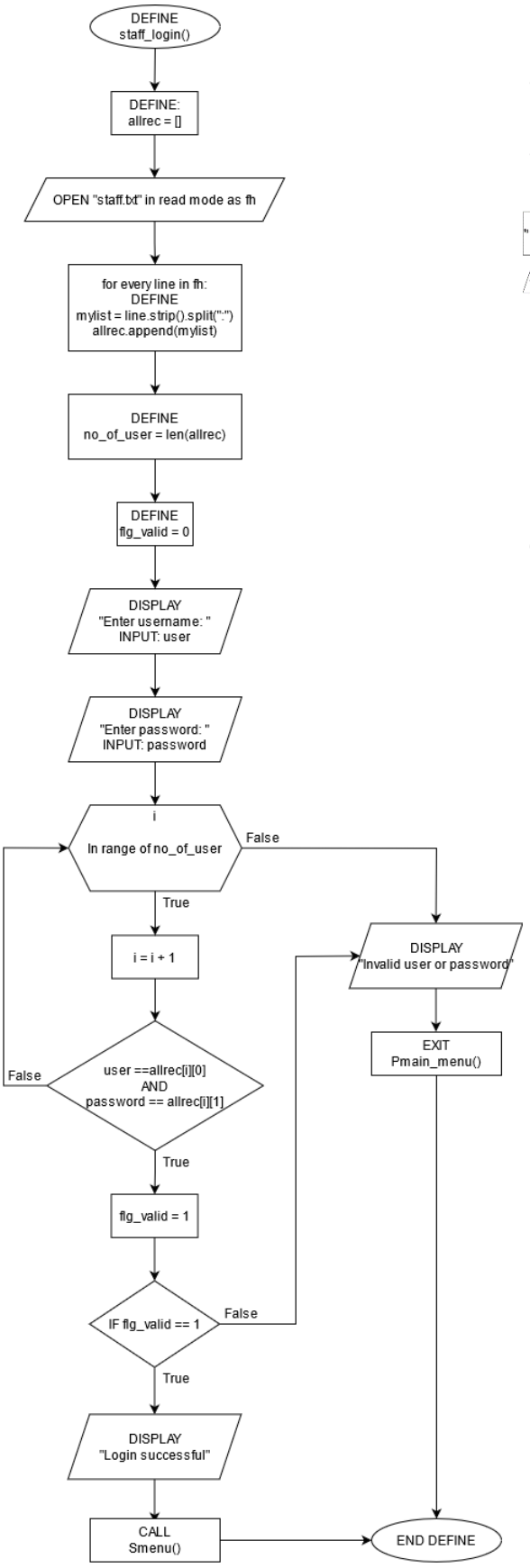
Flowchart:

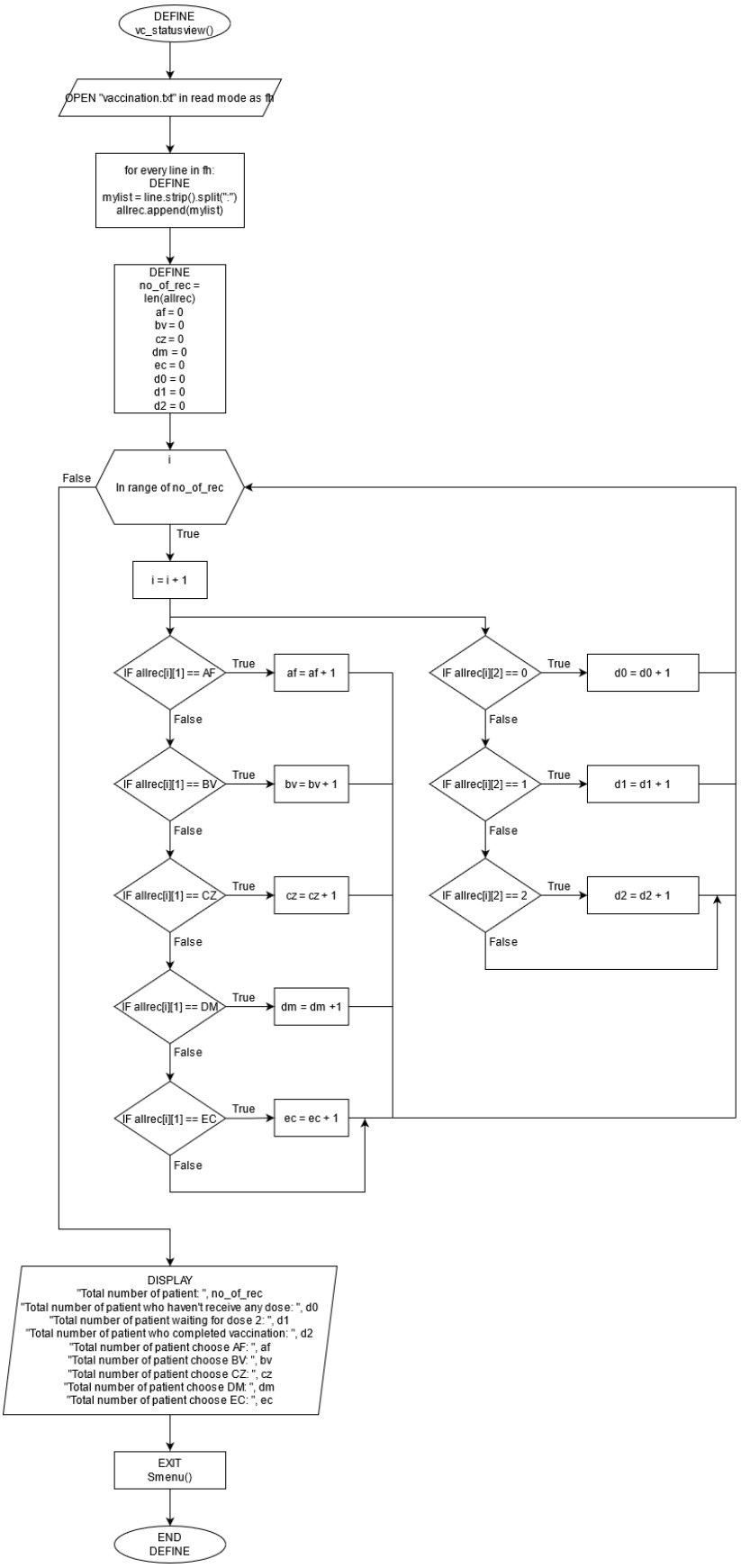

BEGIN
CALL Pmain_menu()
END
DEFINE
Pmain_menu()
DISPLAY
"1 - New patient registration
2 - Patient login
3 - Staff login
4 - Exit
Enter your choice:
INPUT: choice
IF choice == "1"
True
CALL
new_patient_reg()
False
IF choice == "2"
True
CALL
patient_login()
False
IF choice == "3"
True
CALL
staff_login()
False
IF choice == "4"
True
EXIT()
False
DISPLAY
"Invalid option"
END
DEFINE


DEFINE
gen_new_id(eid)
OPEN "id.txt" in assign mode as fh
DEFINE
rec = fh.readline()
IF eid == "VCA"
True
DEFINE
ind = 0
IF eid == "VCB"
True
DEFINE
ind = 1
DEFINE
mylist = rec.split("-")
nextid = mylist[ind]
newid = str(ind(nextid[3:])+1
length of nextid == 1
True
nextid = nextid[:3] + "0000" + newid
length of nextid == 2
True
nextid = nextid[:3] + "000" + newid
length of nextid == 3
True
nextid = nextid[:3] + "00" + newid
length of nextid == 4
True
nextid = nextid[:3] + "0" + newid
length of nextid == 5
True
nextid = nextid[:3] + newid
mylist[ind] = nextid
rec = "-".join(mylist)
OPEN "id.txt" in write mode as fh
then write rec inside
RETURN
newid
END
DEFINE


DEFINE patient_login()
DEFINE allrec = [] allrec2 = []
OPEN "patients.txt" in read mode as fh
for every line in fh DEFINE mylist = line.strip().split(",") allrec.append(mylist)
OPEN "vaccination.txt" in read mode as fh
for every line in fh DEFINE mylist2 = line.strip().split(",") allrec2.append(mylist)
DEFINE no_of_user = len(allrec) no_of_user2 = len(allrec2)
DEFINE fg_valid = 0
DISPLAY "Enter patient ID: " INPUT: user
DISPLAY "Enter password: " INPUT: password
i
In range of no_of_user
False
True
i = i + 1
user == allrec[i][0] AND password == allrec[i][1]
DISPLAY "Invalid user or password"
EXIT Pmain_menu()
fg_valid = 1 DEFINE I = i
fg_valid = 1
False
True
In range of no_of_user2
k = k + 1
DISPLAY "Patient not found in vaccination list Please contact staff."
EXIT Pmain_menu()
END DEFINE
user == allrec2[k][0] AND password == allrec2[k][1]
DEFINE I = k
DISPLAY "Login success!\n"
DISPLAY "1 - Modify data 2 - View current status 3 - Exit (To refresh current data) INPUT: choice
IF choice = 1
CALL Pmod_data(I)
END DEFINE
IF choice = 2
DISPLAY "Patient ID: ", allrec[I][0] "Name: ", allrec[I][2] "Vaccine centre choosen: ", allrec[I][5] "Vaccine choosen: ", allrec[I][6]
IF allrec2[I][2] == "0"
DISPLAY "Vaccine status: Pending for dose"
IF allrec2[I][2] == "1"
DISPLAY "Vaccine status: Dose 1 complete, pending for dose 2"
IF allrec2[I][2] == "2"
DISPLAY "Vaccine status: All dose complete "
allrec2[I][3] available
DISPLAY "Vaccine time: ", allrec2[I][3]
allrec2[I][4] available
DISPLAY "Dose1 date: ", allrec2[I][4]
DISPLAY "Patient haven't been vaccine"
allrec2[I][5] available
DISPLAY "Dose2 date: ", allrec2[I][5]
IF choice = 3
EXIT(Pmain_menu())
END DEFINE
DISPLAY "Invalid choice"
DEFINE Pmod_data(PID)
OPEN "patients.txt" in read mode as fh
for every line in fh DEFINE mylist = line.strip().split(",") allrec.append(mylist)
DISPLAY "1 - Patient Name 2 - Contact number 3 - Exit Enter field to modify: " INPUT: choice
IF choice == 1
DISPLAY "Current name: allrec[PID][2]
DISPLAY "Enter new name: " INPUT: name
length of name is 0 after strip
DISPLAY "Please input valid name"
DISPLAY "Current name: ", allrec[PID][2] "New name: ", new_name "Comfirm (Y/N): " INPUT: d_com
IF d_com = Y
allrec[PID][2] = new_name
OPEN patients.txt in write mode as fh write in allrec
IF d_com = N
DISPLAY "No change made"
DISPLAY "Invalid input"
IF choice == 2
DISPLAY "Current contact number: ", allrec[PID][4]
DISPLAY "New contact number: " INPUT: p_number
p_number == int
DISPLAY "Invalid input"
DISPLAY "Current contact number: ", allrec[PID][4] "New contact number: ", p_number "Comfirm (Y/N): " INPUT: d-com
IF d_com = Y
allrec[PID][4] = str(p_number)
OPEN patients.txt in write mode as fh write in allrec
IF d_com = N
DISPLAY "No change made"
DISPLAY "Invalid input"
IF choice == 3
BREAK
DISPLAY "Invalid choice"
END DEFINE
True
False


DEFINE
Smanu()
DEFINE
"1 - Vaccine Administration
2 - Search Patient Record and Vaccination Status
3 - Statistical Information on Patients Vaccinated
4 - Exit
Enter your choice: "
INPUT: choice
IF choice == 1
True
CALL
vc_admin()
False
IF choice == 2
True
CALL
vc_admin()
False
IF choice == 3
True
CALL
vc_statusview()
False
IF choice == 4
True
EXIT
Pmain_menu()
END
DEFINE
False
DISPLAY
"Invalid choice"


DEFINE
staff_login()
DEFINE:
allrec = []
OPEN "staff.txt" in read mode as fh
for every line in fh:
DEFINE
mylist = line.strip().split("-")
allrec.append(mylist)
DEFINE
no_of_user = len(allrec)
DEFINE
flg_valid = 0
DISPLAY
"Enter username: "
INPUT: user
DISPLAY
"Enter password: "
INPUT: password
i
In range of no_of_user
False
True
i = i + 1
user ==allrec[i][0]
AND
password == allrec[i][1]
False
True
flg_valid = 1
IF flg_valid == 1
False
True
DISPLAY
"Login successful"
CALL
Smenu()
DISPLAY
"Invalid user or password"
EXIT
Pmain_menu()
END DEFINE


DEFINE
vc_statusview()
OPEN "vaccination.txt" in read mode as fh
for every line in fh:
DEFINE
mylist = line.strip().split(",")
allrec.append(mylist)
DEFINE
no_of_rec =
len(allrec)
af = 0
bv = 0
cz = 0
dm = 0
ec = 0
d0 = 0
d1 = 0
d2 = 0
i
In range of no_of_rec
False
True
i = i + 1
IF allrec[i][1] == AF
True
af = af + 1
False
IF allrec[i][1] == BV
True
bv = bv + 1
False
IF allrec[i][1] == CZ
True
cz = cz + 1
False
IF allrec[i][1] == DM
True
dm = dm +1
False
IF allrec[i][1] == EC
True
ec = ec + 1
False
IF allrec[i][2] == 0
True
d0 = d0 + 1
False
IF allrec[i][2] == 1
True
d1 = d1 + 1
False
IF allrec[i][2] == 2
True
d2 = d2 + 1
False
DISPLAY
"Total number of patient: ", no_of_rec
"Total number of patient who haven't receive any dose: ", d0
"Total number of patient waiting for dose 2: ", d1
"Total number of patient who completed vaccination: ", d2
"Total number of patient choose AF: ", af
"Total number of patient choose BV: ", bv
"Total number of patient choose CZ: ", cz
"Total number of patient choose DM: ", dm
"Total number of patient choose EC: ", ec
EXIT
Smenu()
END
DEFINE

Program Source Code and Explanation

To make explanation easier, picture of part of the program is include

Variables: A reserved memory location to store values.

```
flg = 0
```

An integer value of 0 is assign to flg in this example.

Numeric Operation: Add, subtract, multiply and divide of number.

```
newid = str(int(nextid[3:])+1)
```

In the example shown, the program is changing value in the list into integer and add 1 to the value then change it back to string. After all that, the value will be assign to variable, newid.

String Operation: Modification of further use of string variable such as assignment, concatenate, comparison, repetition and so on.

```
p_pass = input("Enter password for your account: ")
p_pass2 = input("Reenter your password: ")
if p_pass == p_pass2:
```

Password and reenter password is request from the user and store as string and assign to p_pass and p_pass2. After that the program will compare first password with the reenter password and do something according to the outcome.

If/Else Conditional: Evaluates whether that an expression is true or false and execute something if the condition is true or false.

```
if age<12:
    print("Sorry, no vaccine available for this age")
    exit(Pmain_menu())
elif age>=18 and age<=45:
    flg = 1
elif age>=12 and age<=45:
    flg = 2
elif age>45:
    flg = 3
```

In this example, if and else if is used where if age enter by user is smaller than 12, the program will print something and exit to main menu. Else if age entered is more than or equal to 18 and smaller or equal to 45, the program will assign flg as 1.

Try/Except Conditional: Allow test run of code where if error is encounter in try block, code in except block will then be execute.

```
while True:
    try:
        age = int(input("Enter new patient age: "))
        break
    except:
        print("Invalid input")
```

In this example, the program will try to change input given by user into integer. So if input given is not integer, the program will encounter error and display invalid input. But because of the while True in the first line, the program will ask user to input age again until integer is input.

While Loop: Allow repeatedly execution of a block of statement if condition is satisfied.

```
while True:
    print("\n\n")
    name = input("Enter new patient name: ")
    if len(name.strip()) == 0:
        print("Please input valid name.")
    else:
        break
```

Program will keep asking user for patient name if length of value input is equal to 0 when space is removed. For example, if user does not enter anything and press enter, the program will display "Please input valid name." and ask the user to input patient name again. But if user enter patient name, the program will exit the loop and continue to other code.

For Loop: Allow looping through an iterable object such as list, set and perform same action for each entry.

```
user = input("Enter username: ")
password = input("Enter password: ")
for i in range(no_of_user):
    if user == allrec[i][1] and password == allrec[i][2]:
        flg valid = 1
```

Program will loop within range of no_of_user start with assigning i with zero and run through the code below and in this case, compare if user and password enter is equal to value in allrec row i column 1 and 2. And if it is equal, the program will break, but if it does not match, the program will continue to the next value for i, where in this case 2 until the number is bigger than no_of_user then the for loop will stop.

List: An ordered and mutable container.

```
mylist = ["a", "b", "C", "1", "2", "3"]
print(mylist)
```

```
['a', 'b', 'C', '1', '2', '3']
```

a, b, c, 1, 2, 3 is assign into mylist as string as list can only store string. With the command print(mylist), the program display the mylist.

Function: A named section of code that performs a specific task.

```
def Pmain_menu():
    while True:
        print("\n\n")
        print("1 - New patient registration")
        print("2 - Patient login")
        print("3 - Staff login")
        print("4 - Exit")
        choice = input("Enter your choice: ")
        if choice == "1":
            new_patient_reg()
        elif choice == "2":
            patient_login()
        elif choice == "3":
            staff_login()
        elif choice == "4":
            exit()
        else:
            print("\n\nInvalid option")
```

This is a function that is called Pmain_menu. This is a function that is used to show menu and ask user choice and bring user to other function.

Files: Information or data that stays in computer storage devices.

```
open ("patient.txt", "r")
open ("patient.txt", "a")
open ("patient.txt", "w")
```

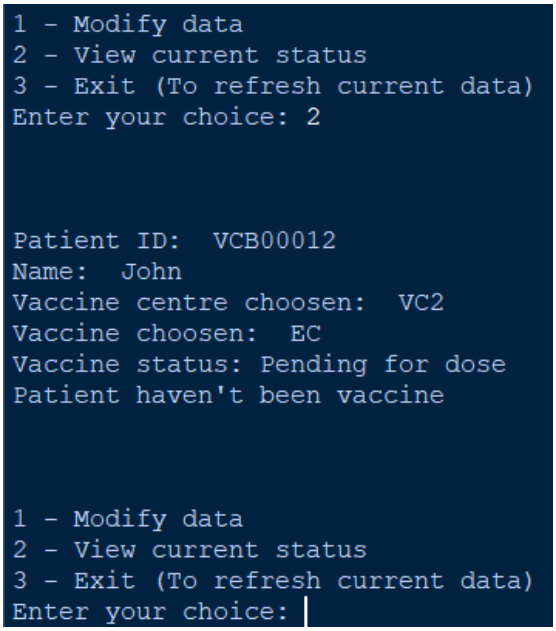
We can open files using command open (“FILES NAME”, “MODE”) where FILES NAME is the name of the file that wish to be open, in this example, patient.txt. And there is three basic mode, r,

read, a, assign, w, write where read mode can only read but cannot change or modify the file, assign mode can add on value to the list and write mode will overlap the value store in the file.

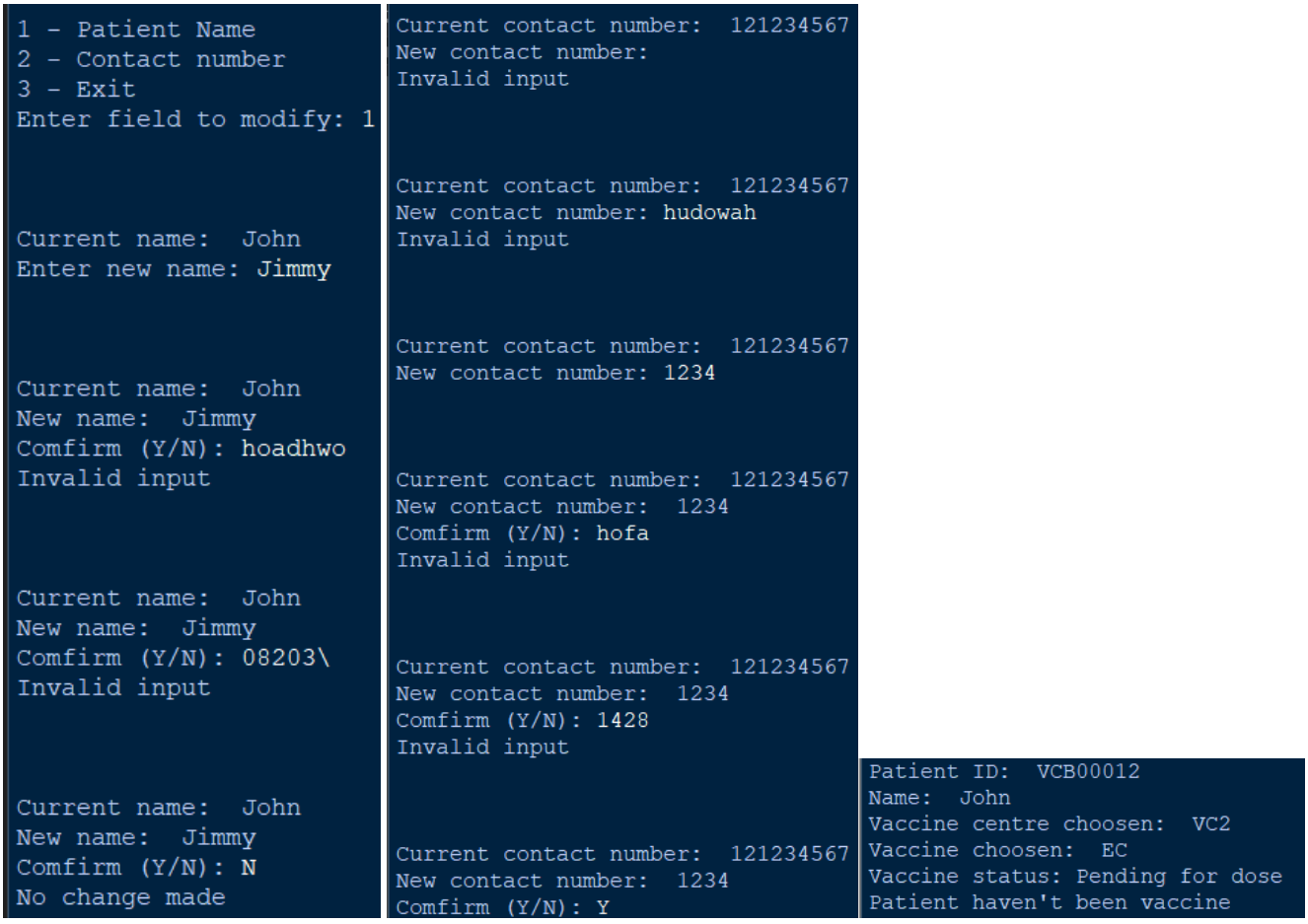
Screenshots of Sample Input / Output and Explanation

```
1 - New patient registration
2 - Patient login
3 - Staff login
4 - Exit
Enter your choice: 1



Enter new patient name: John
Enter new patient age: oidhas
Invalid input
Enter new patient age: 80
Vaccine available:
================================================================
 Vaccine Code |    Dosage Required     | Interval Between Doses
================================================================
AF            |           2            |   2 weeks (14 days)
BV            |           2            |   3 weeks (21 days)
DM            |           2            |   4 weeks (28 days)
EC            |           1            |           -
================================================================
Enter vaccine you wish to take: adhuha
Invalid choice
Enter vaccine you wish to take: 8124719
Invalid choice
Enter vaccine you wish to take: EC
Vaccine successfully register


Enter vaccine centre you wish to go(VC1 or VC2): sojf
Invalid input
Enter vaccine centre you wish to go(VC1 or VC2): 18y49
Invalid input
Enter vaccine centre you wish to go(VC1 or VC2): VC2
Location successfully register
Patient ID:  VCB00012
Enter contact number:
Invalid input
Enter contact number: dowajid
Invalid input
Enter contact number: 0121234567
Enter password for your account: 123
Reenter your password: 123
Password successfully register
```

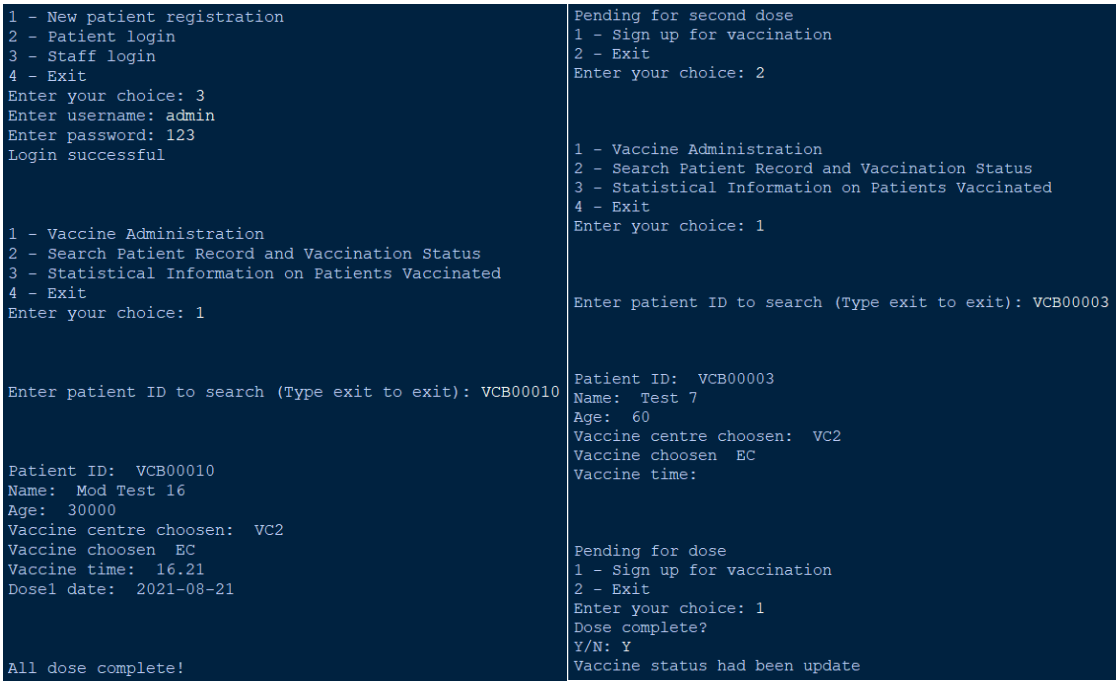
This is an example of the process of new patient registration. The program will ask user to input their name, age and output vaccine available according to patient's age. After that, program output patient ID and ask for patient to input their contact number and password for their account. While there is no restriction for name that patient can input but there is restriction for age, vaccine chosen, vaccine centre chosen and contact number where only number is allowed to be input in age and phone number, available vaccine type is able to be input into vaccine chosen and available vaccine centre is able to be input.



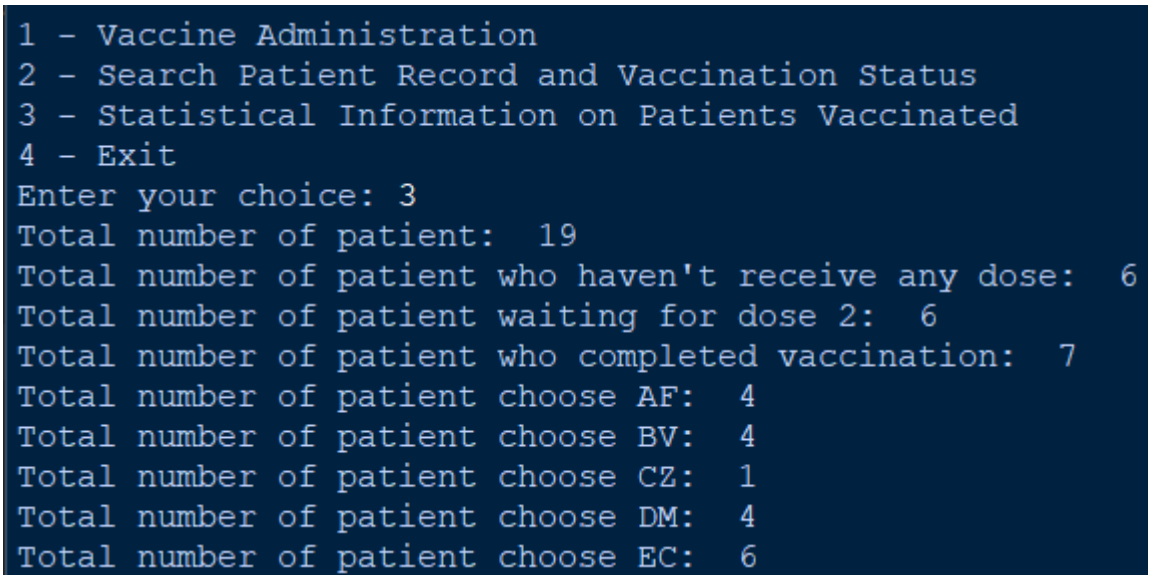
This is the process for user to login to their account in order to function such as modify data and view their current status.



This is the process of modifying data where invalid input such as empty and alphabet is not allowed when modifying phone number. Patient ID, name, vaccine centre chosen, vaccine status will been shown when user choose to view current status.



This is the process of vaccine administration or search patient record and vaccination status. The program will ask user to input patient ID in order to search for patient that need to be update. After that, depend on patient's vaccine stage display patient data and give option for user to update patient vaccine stage and calculate second dose date if available.



This is the process of view statistical information on patients vaccinated. Where the system will output statistical information on patients vaccinated.

All the test data:

VCA00001:123:Test 1:12:123:VC1:AF
VCB00001:123:Test 2:18:123:VC2:BV
VCA00002:123:Test 3:20:123:VC1:DM
VCA00003:123:Test 4:30:123:VC1:CZ
VCB00002:123:Test 5:40:123:VC2:BV
VCA00004:123:Test 6:50:123:VC1:AF
VCB00003:123:Test 7:60:123:VC2:EC
VCB00004:123:Test 8:60:123:VC2:DM
VCB00005:123:Test 9:70:123:VC2:EC
VCA00005:1231:Test 10:80:123:VC1:BV
VCB00006:123:Test 11:90:123:VC2:DM
VCA00006:123:Test 12:100:123:VC1:AF
VCB00007:123:Test 13:110:123:VC2:AF
VCB00008:123:Test 14:120:123:VC2:DM
VCB00009:123:Test 15:130:123:VC2:EC
VCB00010:123:Mod Test 16:30000:4567:VC2:EC
VCA00007:12:Test 17:36:1234:VC1:EC
VCA00008:123:John:47:121234567:VC1:BV
VCB00012:123:John:80:1234:VC2:EC

VCA00001:AF:1:16.11:2021-08-21:2021-09-04
VCB00001:BV:1:16.11:2021-08-21:2021-09-11
VCA00002:DM:2:11.26:2021-08-22:2021-09-19
VCA00003:CZ:2:11.26:2021-08-22:2021-09-12
VCB00002:BV:1:11.30:2021-08-22:2021-09-12
VCA00004:AF:1:23.00:2021-08-20:2021-09-03
VCB00003:EC:2::23.56:2021-08-24
VCB00004:DM:1:16.20:2021-08-21:2021-09-18
VCB00005:EC:0
VCA00005:BV:1:11.31:2021-08-22:2021-09-12
VCB00006:DM:2:11.31:2021-08-22:2021-09-19
VCA00006:AF:0
VCB00007:AF:0
VCB00008:DM:0
VCB00009:EC:2:16.20:2021-08-21
VCB00010:EC:2:16.21:2021-08-21
VCA00007:EC:2:11.15:2021-08-22
VCA00008:BV:0
VCB00012:EC:0

Left: patients.txt

Right: vaccination.txt

In the left list is list contain in patients.txt where starting from left to right are patient id, password, name, age, phone number, vaccine centre chosen and vaccine chosen. And in the right list is list contain in vaccination.txt where from left to right are patient id, vaccine chosen, vaccination stage, dose time, first dose date and second dose date. For vaccination stage 0 means patient haven't take any dose, 1 means patient have take first dose, 2 means patient has complete all dose.

Conclusion

This whole system is designed to aid the process of vaccination. And throughout this module, I've learned how to code using python from zero as I have no coding experience in python before. This is a fun class and a class full of knowledge as knowing how to code in python would help me a lot in my future career. Thanks for mister Usman for teaching me in this module, it benefits me so much.

References:

- MuraliKrishna (2011, July 29). *Adding days to a date in Python*. Retrieved August 25, 2021, from https://stackoverflow.com/questions/6871016/adding-days-to-a-date-in-python
- Retrived August 27, 2021, from http://www.aees.gov.in/htmldocs/downloads/XI_Class_Content_Computer_Science/8-Handout.pdf
- Process (October 5, 2020). *Progress Information Hub.* Retrieved August 27, 2021, from https://docs.progress.com/zh-CN/bundle/datadirect-connect-odbc-71/page/Numeric-Operators_2.html
- Priya Pedamkar (April 5, 2021) *String Operators in Python.* Retrieved August 27, 2021, from https://www.educba.com/string-operators-in-python/
- James Gallagher (October 22, 2020). *If else Python Statements: A Step-By-Step Guide*. Retrieved August 27, 2021, from https://www.educba.com/string-operators-in-python/
- James Gallagher (December 2, 2020). *Python Try Except: A Step-By-Step Guide.* Retrieved August 27, 2021, from https://careerkarma.com/blog/python-try-except/
- GeeksforGeeks (August 25, 2021). *Python While Loop*. Retrieved August 27, 2021, from https://www.geeksforgeeks.org/python-while-loop/
- Charlie Custer (May 30, 2019). *The Basics of Python For Loops A Tutorial.* Retrieved August 27, 2021, from https://www.dataquest.io/blog/python-for-loop-tutorial/
- Amanda Iglesias Moreno (April 21, 2020). *15 things you should know about List in Python.* Retrieved August 27, 2021, from https://towardsdatascience.com/15-things-you-should-know-about-lists-in-python-c566792eca98

- Sadrach Pierre (September 5, 2020). *Function Definition in Python.* Retrieved August 27, 2021, from https://towardsdatascience.com/function-definition-in-python-bae11c29f4cd
- Retrieved August 27, 2021, from https://pymbook.readthedocs.io/en/latest/file.html