



INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT010-3-1-PYP

PYTHON PROGRAMMING

WOON YAN QI VIVIAN – TP054626

**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 SEPTEMBER 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

1. Introduction.....	3
2. Assumptions.....	4
3. Design of the Program	
3.1 Pseudocode.....	6
3.2 Flowcharts.....	24
4. Program Source Code and Explanation.....	45
5. Screenshots of Sample Input/Output and Explanation.....	62
6. Conclusion.....	80

1 Introduction

Covid-19 is a lethal disease caused by a deadly virus which named as Coronavirus. Vaccination needs to be taken to prevent become a victim of this global pandemic. However, to increase the vaccination process efficiency, a vaccination application specifically for Covid-19. It allows new registration to new patient, vaccination administration, search patient record and vaccination status, and show statistical information on patients record.

To improve the work efficiency, this application is not only for staff, but also for patients by using login function. Of course, through login, some functions are restricted to the patient. Both staff and patient can access new registration, search patient record and vaccination status and show statistical information on patients record. The vaccination administration function is for staff only. By the way, patient can search their own vaccination information only, meanwhile, staff can proceed to every patient's record.

Apart from these functions mentioned above, some additional functions are proposed to provide a variety of help.

2 Assumptions

1. There are only two vaccination centre, named as VC1 and VC2. Patients can choose the vaccination centre while doing registration.
2. There are five types of vaccine provided, named as AF, BV, CZ, DM, EC. The restriction of age upon every type of vaccine is as the table below. While patient doing registration, the program will list out the type of vaccine which are suitable for the patients according to the age of patient.

Table 1: Details of Vaccines Administered in VC1 and VC2

Vaccine Code	Dosage Required	Interval Between Doses	Age Group
AF	2	2 weeks (or 14 days)	12 and above
BV	2	3 weeks (or 21 days)	18 and above
CZ	2	3 weeks (or 21 days)	12 - 45
DM	2	4 weeks (or 28 days)	12 and above
EC	1		18 and above

3. The date of dose 1 is three days after the registration, patients are noticed about this in advanced so they can do good time planning.
4. Malaysian need to use IC and non-Malaysian use passport number for registration. One IC/passport only can register for once.
5. Patients are not forced to come on the date given, but they are advised, and they have the obligation to do so. The date is for the vaccination centre to suitable preparation in advanced on the specific date. If patients simply don't follow the date given, the vaccine may be wasted. Also, if patient comes late for the dose 2, the effectiveness of the vaccination will decrease and this may caused irreversible lost since one IC/passport only can register once.
6. There is no maximum number of vaccination administration in a day, the vaccination centre will do suitable preparation according to the patient's registration.

7. Although the patient will get noticed on the dose 1 and dose 2 date after registration, the dose 2 date depends on the actual dose 1 date. Hence, if patient comes early or late for the dose 1, the dose 2 date will change accordingly.
8. Due to the meaning of dose date, patient is not allowed to change either the vaccination centre or vaccine code after registration. This will affect the preparation for vaccination. Thus, during registration, patient have many chances to confirm and change the registration information. Once the registration is done, patient can only change their contact number and email.
9. Users can exit the menu only when they are intend to, this program won't stop by error.
10. While checking patient record and vaccination status, the vaccination status will be prompt according to the cases as the table below.

Table 2: Patient Vaccination Status

Vaccine Code	Dosage Required	Status		
		Before Dose 1	After Dose 1	After Dose 2
AF	2	NEW	COMPLETED-D1	COMPLETED
BV	2			
CZ	2			
DM	2			
EC	1		COMPLETED	

11. The staff username and password to use for staff login cannot be registered to the program, it is keyed in manually in the stafflogin.txt file. The usernames are unique, the use of username depends on the counter. For example, for counter 1 in VC1, the username is SVC1000001. Hence, the situation of multiple persons using the same username at the same time shall not happen.

3 Design of the Program

3.1 Pseudocode

Main Logic

```

BEGIN
    IMPORT datetime
    DEFINE FUNCTION user_login, add_user, staff_login, patient_main_menu, staff_main_menu
    DEFINE LIST plogallrec, slogallrec, pallrec, vallrec, plogrec, ploginfo, fhlog, slogrec, sloginfo, fhslog, prec, pinfo, fhp, vrec, vinfo, fhv
    DEFINE choice, valid_user
    WITH OPEN (patientlogin.txt) in read mode as fhlog
        LOOP plogrec in fhlog
            ploginfo = plogrec.split(":")
            plogallrec.append(ploginfo)
            NEXT plogrec
        ENDLOOP
    ENDWITH
    WITH OPEN (stafflogin.txt) in read mode as fhslog
        LOOP slogrec in fhslog
            sloginfo = slogrec.split(":")
            slogallrec.append(sloginfo)
            NEXT slogrec
        ENDLOOP
    ENDWITH
    WHILE TRUE
        DISPLAY("1-Login as patient")
        DISPLAY("2-Sign up as patient")
        DISPLAY("3-Staff")
        DISPLAY("4-Exit")
        DISPLAY("Please select your identity:")
        READ choice
        IF choice=="1" THEN
            CALL FUNCTION user_login()
            valid_user=user_login(plogallrec)
            IF valid_user==1 THEN
                CALL FUNCTION patient_main_menu()
            ELSE THEN
                DISPLAY("LOGIN FAILED! Username or password wrong.")
            ENDIF
        ELSE IF choice=="2" THEN
            CALL FUNCTION add_user()
            plogallrec.append(add_user())
        ELSE IF choice=="3" THEN
            CALL FUNCTION staff_login()
            valid_user = staff_login(slogallrec)
            IF valid_user==1 THEN
                CALL FUNCTION staff_main_menu()
            ELSE THEN
                DISPLAY("LOGIN FAILED! Username or password wrong.")
            ENDIF
        ELSE IF choice=="4" THEN
            DISPLAY("Exited Program")
            break
        ELSE THEN
            DISPLAY("INVALID INPUT!!!!")
        ENDIF
    ENDWHILE
END

```

user_login(plogallrec)

```

BEGIN
    DEFINE FUNCTION user_login(plogallrec)
        DEFINE LIST plogallrec, i
        DEFINE username, password, flg
        DISPLAY("Enter username:")
        READ username
        DISPLAY("Enter password:")
        READ password
        flg=0
        LOOP i in range(len(plogallrec))
            IF username in plogallrec[i][0] AND password in plogallrec[i][1] THEN
                flg=1
                break
            ENDIF
            NEXT i
        ENDLOOP
        RETURN flg
    ENDDEFINE
END

```

add_user()

```

BEGIN
    DEFINE FUNCTION add_user()
        DEFINE LIST fhplog, newrec, plogallrec, i
        DEFINE username, password, flag
        WITH OPEN (patient.txt) in append mode as fhplog
            DISPLAY("Please enter username:")
            READ username
            DISPLAY("Please enter password:")
            READ password
            flag=0
            LOOP i in range(len(plogallrec))
                IF username in plogallrec[i][0] THEN
                    DISPLAY ("SIGN UP FAILD! Current username is used.")
                    flag=1
                    break
                ENDIF
                NEXT i
            ENDLOOP
            IF flag==0 THEN
                newrec.append(username)
                newrec.append(password)
                fhplog.write(":".join(newrec)+"\n")
            ENDIF
        ENDWITH
        RETURN newrec
    ENDDEFINE
END

```

staff_login(slogallrec)

```
BEGIN
    DEFINE FUNCTION staff_login(slogallrec)
        DEFINE LIST i, slogallrec
        DEFINE username, password, flg
        DISPLAY("Enter username:")
        READ username
        DISPLAY("Enter password:")
        READ password
        flg=0
        LOOP i in range(len(slogallrec))
            IF username in slogallrec[i][0] AND password in slogallrec[i][1] THEN
                flg=1
                break
            ENDIF
            NEXT i
        ENDLOOP
        RETURN flg
    ENDDEFINE
END
```

patient_main_menu()

```

BEGIN
    DEFINE FUNCTION patient_main_menu()
        DEFINE FUNCTION add_patient, get_new_id, modify_patient, modify_login, patient_check_status, search_patient_id, check_overall_status
        DEFINE LIST pallrec, vallrec, plogallrec, prec, pinfo, vrec, vinfo, plogrec, ploginfo, fhplog, fhp, fhv
        DEFINE choice, returns
        WITH OPEN (patient.txt) in read mode as fhp
            FOR prec in fhp
                pinfo = prec.split(":")
                pallrec.append(pinfo)
            ENDFOR
        ENDWITH
        WITH OPEN (patientlogin.txt) in read mode as fhplog
            FOR plogrec in fhplog
                ploginfo = plogrec.split(":")
                plogallrec.append(ploginfo)
            ENDFOR
        ENDWITH
        WITH OPEN (vaccination.txt) in read mode as fhv
            FOR vrec in fhv
                vinfo = vrec.split(":")
                vallrec.append(vinfo)
            ENDFOR
        ENDWITH
        WHILE TRUE
            DISPLAY ("1- Register:")
            DISPLAY ("2- Modify Registration:")
            DISPLAY ("3- Modify Login:")
            DISPLAY ("4- Check Vaccination Status:")
            DISPLAY ("5- Check Overall Vaccination Status:")
            DISPLAY ("6- Check Patient ID:")
            DISPLAY ("7- Exit.")
            DISPLAY ("Enter your choice to proceed to next step:")
            READ choice

            IF choice=="1" THEN
                CALL FUNCTION add_patient()
                returns=add_patient()
                IF returns!=[[],[]] THEN
                    pallrec.append(returns[0])
                    vallrec.append(returns[1])
                ENDIF
            ELSE IF choice=="2" THEN
                CALL FUNCTION modify_patient(pallrec)
                pallrec = modify_patient(pallrec)
            ELSE IF choice=="3" THEN
                CALL FUNCTION modify_login(plogallrec)
                plogallrec = modify_login(plogallrec)
            ELSE IF choice=="4" THEN
                CALL FUNCTION patient_check_status(vallrec)
            ELSE IF choice=="5" THEN
                CALL FUNCTION check_overall_status(vallrec)
            ELSE IF choice=="6" THEN
                CALL FUNCTION search_patient_id(pallrec)
            ELSE IF choice=="7" THEN
                break
            ELSE THEN
                DISPLAY ("INVALID OPTION!")
            ENDIF
        ENDWHILE
    ENDDEFINE
END

```

staff_main_menu()

```

BEGIN
    DEFINE staff_main_menu()
        DEFINE FUNCTION display_patient_rec, display_vac_rec, vaccination_administration, check_overall_status, check_patient_status,
                    search_patient_id, modify_patient, add_patient, get_new_id
        DEFINE LIST pallrec, vallrec, prec, pinfo, vrec, vinfo, fhp, fhv
        DEFINE choice, returns
        WITH OPEN (patient.txt) in read mode as fhp
            FOR prec in fhp
                pinfo = prec.split(":")
                pallrec.append(pinfo)
            ENDOFOR
        ENDWITH
        WITH OPEN (vaccination.txt) in read mode as fhv
            FOR vrec in fhv
                vinfo = vrec.split(":")
                vallrec.append(vinfo)
            ENDOFOR
        ENDWITH
        WHILE (True)
            DISPLAY ("1- Register:")
            DISPLAY ("2- Modify Patient's Registration:")
            DISPLAY ("3- Display Patient's Personal Information:")
            DISPLAY ("4- Display Patient's Vaccination Information:")
            DISPLAY ("5- Vaccination Administration:")
            DISPLAY ("6- Check Patient Vaccination Status:")
            DISPLAY ("7- Check Overall Vaccination Status:")
            DISPLAY ("8- Check Patient ID:")
            DISPLAY ("9- Exit.")
            DISPLAY ("Enter your choice to proceed to next step:")
            READ choice

            IF choice=="1" THEN
                CALL FUNCTION add_patient()
                returns=add_patient()
                IF returns!=([],[]) THEN
                    pallrec.append(returns[0])
                    vallrec.append(returns[1])
                ENDIF
            ELSE IF choice=="2" THEN
                CALL FUNCTION modify_patient(pallrec)
                pallrec = modify_patient(pallrec)
            ELSE IF choice=="3" THEN
                CALL FUNCTION display_patient_rec(pallrec)
            ELSE IF choice=="4" THEN
                CALL FUNCTION display_vac_rec(vallrec)
            ELSE IF choice=="5" THEN
                CALL FUNCTION vaccination_administration(vallrec)
            ELSE IF choice=="6" THEN
                CALL FUNCTION check_patient_status(vallrec)
            ELSE IF choice=="7" THEN
                CALL FUNCTION check_overall_status(vallrec)
            ELSE IF choice=="8" THEN
                CALL FUNCTION search_patient_id(pallrec)
            ELSE IF choice=="9" THEN
                break
            ELSE THEN
                DISPLAY ("INVALID OPTION!")
            ENDIF
        ENDWHILE
    ENDDEFINE
END

```

add_patient()

```

BEGIN
    DEFINE FUNCTION add_patient()
    DEFINE FUNCTION get_new_id()
    DEFINE LIST new_prec, new_vrec, pallrec, fh
    DEFINE name, bigflag, flag, icpp, malaysian, choice, i, vc_centre, age, vc_code, contact, email, today, d1, d2, didate, d2date, choose, pid_code, pid
    WHILE TRUE
        WHILE TRUE
            DISPLAY ("Enter your name as per IC or passport:")
            READ name
            name=name.upper()
            name = ''.join(i for i in name if not i.isdigit())
            DISPLAY ("Your name is:", name)
            DISPLAY ("Press 1 to continue:")
            READ choice
            IF choice=="1" THEN
                break
            ELSE THEN
                DISPLAY ("Please reenter your name.")
            ENDIF
        ENDWHILE

        bigflag=0
        WHILE bigflag==0:
            DISPLAY ("Are you Malaysian? YES/NO:")
            READ malaysian
            IF malaysian=="YES" OR malaysian=="yes"
                DISPLAY ("Enter your IC number: (ex: 123456-78-9999)")
                READ icpp
                TRY
                    IF icpp[6]=="-" AND icpp[9]=="-" AND len(icpp)==14 AND type(int(icpp[:6]))==int AND type(int(icpp[7:9]))==int AND type(int(icpp[10:]))==int THEN
                        flag=0
                        LOOP i in range (len(pallrec))
                            IF icpp in pallrec[i][5] THEN
                                DISPLAY ("This IC already registered!")
                                flag=1
                                break
                            ENDIF
                        NEXT i
                    ENDLOOP
                    IF flag==0 THEN
                        DISPLAY ("Your IC is:", icpp)
                        DISPLAY ("Press 1 to continue:")
                        READ choice
                        IF choice=="1" THEN
                            bigflag=1
                        ELSE THEN
                            DISPLAY ("Please reenter your IC.")
                            flag=1
                        ENDIF
                    ENDIF
                    ELSE THEN
                        DISPLAY ("INVALID INPUT!")
                    ENDIF
                EXCEPT
                    DISPLAY ("INVALID INPUT!")
                ENDTRY
            ELSE IF malaysian == "NO" OR malaysian == "no" THEN
                DISPLAY ("Enter your passport number:")
                READ icpp
                flag=0
                LOOP i in range (len(pallrec))
                    IF icpp in pallrec[i][5] THEN
                        DISPLAY ("This passport already registered!")
                        flag=1
                        break
                    ENDIF
                NEXT i
                ENDLOOP
                IF flag==0
                    DISPLAY ("Your passport is:", icpp)
                    DISPLAY ("Press 1 to continue:")
                    READ choice
                    IF choice=="1" THEN
                        bigflag=1
                    ELSE THEN
                        DISPLAY ("Please reenter your passport.")
                        flag=1
                    ENDIF
                ENDIF
                ELSE THEN
                    DISPLAY ("INVALID INPUT!")
                ENDIF
            ENDWHILE

            WHILE TRUE
                DISPLAY("Vaccine centre:")
                DISPLAY("1-VC1")
                DISPLAY("2-VC2")
                DISPLAY("Enter your choice:")
                READ vc_centre
                IF vc_centre == "1" THEN
                    vc_centre="VC1"
                    DISPLAY("Your vaccine centre is:"+vc_centre)
                    DISPLAY("Press 1 to continue:")
                    READ choice
                    IF choice=="1" THEN
                        break
                    ELSE THEN
                        DISPLAY ("Please reselect your name.")
                    ENDIF
                ELSE IF vc_centre == "2" THEN
                    vc_centre="VC2"
                    DISPLAY("Your vaccine centre is:", vc_centre)
                    DISPLAY("Press 1 to continue:")
                    READ choice
                    IF choice=="1" THEN
                        break
                    ELSE THEN
                        print("Please reselect your name.")
                    ENDIF
                ENDIF
            ENDWHILE
        ENDWHILE
    ENDIF
END

```

```

        ELSE THEN
            DISPLAY ("INVALID INPUT!")
        ENDIF
    ENDWHILE

    bigflag=0
    WHILE bigflag==0
        flag=0
        WHILE flag==0
            TRY
                DISPLAY ("Enter your age:")
                READ int(age)
                IF age>=12 AND age<18 THEN
                    WHILE TRUE
                        DISPLAY ("Available vaccine code for your age:")
                        DISPLAY ("1-AF")
                        DISPLAY ("2-CZ")
                        DISPLAY ("3-DM")
                        DISPLAY ("4-EC")
                        DISPLAY ("Enter your choice:")
                        READ vc_code
                        IF vc_code == "1" THEN
                            vc_code="AF"
                            flag=1
                            break
                        ELSE IF vc_code == "2" THEN
                            vc_code="CZ"
                            flag=1
                            break
                        ELSE IF vc_code == "3" THEN
                            vc_code="DM"
                            flag=1
                            break
                        ELSE THEN
                            DISPLAY ("INVALID INPUT!")
                        ENDIF
                    ENDWHILE
                ELSE IF age>=18 AND age<=45 THEN
                    WHILE TRUE
                        DISPLAY ("Available vaccine code for your age:")
                        DISPLAY ("1-AF")
                        DISPLAY ("2-BV")
                        DISPLAY ("3-CZ")
                        DISPLAY ("4-DM")
                        DISPLAY ("5-EC")
                        DISPLAY ("Enter your choice:")
                        READ vc_code
                        IF vc_code == "1" THEN
                            vc_code="AF"
                            flag=1
                            break
                        ELSE IF vc_code == "2" THEN
                            vc_code="BV"
                            flag=1
                            break
                        ELSE IF vc_code == "3" THEN
                            vc_code="CZ"
                            flag=1
                            break
                        ELSE IF vc_code == "4" THEN
                            vc_code="DM"
                            flag=1
                            break
                        ELSE IF vc_code == "5" THEN
                            vc_code="EC"
                            flag=1
                            break
                        ELSE THEN
                            DISPLAY ("INVALID INPUT!")
                        ENDIF
                    ENDWHILE
                ELSE IF age>=46 AND age<=200 THEN
                    WHILE TRUE
                        DISPLAY ("Available vaccine code for your age:")
                        DISPLAY ("1-AF")
                        DISPLAY ("2-BV")
                        DISPLAY ("3-DM")
                        DISPLAY ("4-EC")
                        DISPLAY ("Enter your choice:")
                        READ vc_code
                        IF vc_code == "1" THEN
                            vc_code="AF"
                            flag=1
                            break
                        ELSE IF vc_code == "2" THEN
                            vc_code="BV"
                            flag=1
                            break
                        ELSE IF vc_code == "3" THEN
                            vc_code="DM"
                            flag=1
                            break
                        ELSE IF vc_code == "4" THEN
                            vc_code="EC"
                            flag=1
                            break
                        ELSE THEN
                            print("INVALID INPUT!")
                        ENDIF
                    ENDWHILE
                ELSE THEN
                    DISPLAY ("Sorry, there's no vaccine available for you.")
                ENDIF
            ENDIF
        EXCEPT
            DISPLAY ("INVALID INPUT!!!!")
        ENDTRY
    ENDWHILE

    WHILE flag==1

```

```

DISPLAY ("Your age is:", str(age))
DISPLAY ("Your vaccine code is:", vc_code)
DISPLAY ("Press 1 to continue:")
READ choice
IF choice=="1" THEN
    bigflag=1
    break
ELSE THEN
    DISPLAY ("Please reselect your age and vaccine code.")
    flag=0
ENDIF
ENDWHILE
ENDWHILE

WHILE TRUE
    DISPLAY ("Enter your mobile number:(ex. 012-23456789)")
    READ contact
    TRY
        IF contact[:2]=="01" AND contact[3]=="-" AND type(int(contact[4:]))==int AND len(contact)==11 OR len(contact)==12 THEN
            DISPLAY ("Your contact is:", contact)
            DISPLAY ("Press 1 to continue:")
            READ choice
            IF choice=="1"THEN
                break
            ELSE THEN
                DISPLAY ("Please reenter your mobile number.")
            ELSE
                DISPLAY ("INVALID INPUT!")
            ENDIF
        EXCEPT
            DISPLAY ("INVALID INPUT!")
        ENDTRY
    ENDWHILE

WHILE TRUE
    DISPLAY ("Enter your e-mail address:")
    READ email
    flag=0
    cnt=0
    LOOP i in email
        IF i=="@" THEN
            flag=1
            break
        ELSE
            cnt=cnt+1
        ENDIF
    NEXT i
    ENDLOOP
    IF cnt!=0 AND flag==1 THEN
        LOOP i in email[cnt+2:-1]
            IF i=="," THEN
                flag=2
                break
            ENDIF
        ENDLOOP
        IF flag==2 THEN
            DISPLAY ("Your email is:", email)
            DISPLAY ("Press 1 to continue:")
            READ choice
            IF choice=="1" THEN
                break
            ELSE THEN
                DISPLAY ("Please reenter your email.")
            ENDIF
        ELSE THEN
            DISPLAY ("INVALID INPUT!")
        ENDIF
    ENDWHILE

today = datetime.datetime.today()
d1 = today + datetime.timedelta(days=3)
d2date = d1.strftime("%d/%m/%Y")
IF vc_code == "AF" THEN
    d2 = d1 + datetime.timedelta(days=14)
    d2date = d2.strftime("%d/%m/%Y")
ELSE IF vc_code == "BV" THEN
    d2 = d1 + datetime.timedelta(days=21)
    d2date = d2.strftime("%d/%m/%Y")
ELSE IF vc_code == "CZ" THEN
    d2 = d1 + datetime.timedelta(days=21)
    d2date = d2.strftime("%d/%m/%Y")
ELSE IF vc_code == "DM" THEN
    d2 = d1 + datetime.timedelta(days=28)
    d2date = d2.strftime("%d/%m/%Y")
ELSE IF vc_code == "EC" THEN
    d2date = "NONE"
ENDIF

DISPLAY ("Name      :", name)
DISPLAY ("Age      :", str(age))
DISPLAY ("Vaccine Centre:", vc_centre)
DISPLAY ("Vaccine Code :", vc_code)
DISPLAY ("Contact   :", contact)
DISPLAY ("Email     :", email)
DISPLAY ("Make sure your information are correct.")
DISPLAY ("Once registration done, only contact and email can be changed.")
DISPLAY ("Press 1 to continue. Press 2 to exit registration. Otherwise, restart the registration:")
READ choose
IF choose=="1" THEN
    break
ELSE IF choose=="2" THEN
    DISPLAY ("exiting to menu...")
    break
ELSE THEN
    DISPLAY ("RESTART REGISTRATION:")
ENDIF

```

```
ENDWHILE
IF choose=="1" THEN
    DISPLAY ("REGISTRATION SUCCESSFUL!")
    pid_code = "P"+vc_code
    pid = get_new_id(pid_code)
    DISPLAY ("Patient ID:", pid)
    DISPLAY ("Dose 1   ;", d1date)
    DISPLAY ("Dose 2   ;", d2date)
    DISPLAY ("REMINDER: Patient ID is important when having vaccination.")
    WITH OPEN (vaccination.txt) in append mode as fh
        new_vrec.append(pid)
        new_vrec.append(vc_centre)
        new_vrec.append(vc_code)
        new_vrec.append("D1")
        new_vrec.append(d1date)
        new_vrec.append("NO")
        new_vrec.append("N/A")
        new_vrec.append("N/A")
        new_vrec.append("N/A")
        new_vrec.append("NEW")
        new_vrec.append(icpp)
        fh.write(":".join(new_vrec)+"\n")
    ENDWITH
    WITH OPEN (patient.txt) in append mode as fh
        new_prec.append(name)
        new_prec.append(str(age))
        new_prec.append(pid)
        new_prec.append(vc_centre)
        new_prec.append(vc_code)
        new_prec.append(icpp)
        new_prec.append(contact)
        new_prec.append(email)
        fh.write(":".join(new_prec)+"\n")
    ENDWITH
ELSE THEN
    DISPLAY ("Registration Failed.")
ENDIF
RETURN new_prec, new_vrec
ENDDEFINE
END
```

```
get_new_id(eid)

BEGIN
    DEFINE FUNCTION get_new_id()
        DEFINE LIST fh, idlist, sep_idlist
        DEFINE eid, ind, nextid, newid
        WITH OPEN (id.txt) in read mode as fh
            idlist=fh.readline()
        ENDWITH
        IF eid == "PAF" THEN
            ind=0
        ELSE IF eid == "PBV" THEN
            ind=1
        ELSE IF eid == "PCZ" THEN
            ind=2
        ELSE IF eid == "PDM" THEN
            ind=3
        ELSE IF eid == "PEC" THEN
            ind=4
        ELSE
            ind=5
        ENDIF

        sep_idlist = idlist.split(":")
        nextid = sep_idlist[ind]
        newid = str(int(nextid[3:]))+1

        IF len(newid) == 1 THEN
            nextid = nextid[:3]+"00000"+newid
        ELSE IF len(newid) == 2 THEN
            nextid = nextid[:3]+"0000"+newid
        ELSE IF len(newid) == 3 THEN
            nextid = nextid[:3]+"000"+newid
        ELSE IF len(newid) == 4 THEN
            nextid = nextid[:3]+"00"+newid
        ELSE IF len(newid) == 5 THEN
            nextid = nextid[:3]+"0"+newid
        ELSE IF len(newid) == 6 THEN
            nextid = nextid[:3]+newid
        ENDIF

        sep_idlist[ind]=nextid
        idlist=":".join(sep_idlist)
        WITH OPEN (id.txt) in write mode as fh
            fh.write(idlist)
        ENDWITH

        RETURN nextid
    ENDDEFINE
END
```

modify_patient(pallrec)

```

BEGIN
  DEFINE FUNCTION modify_patient(pallrec)
    DEFINE LIST pallrec, fhp, prec
    DEFINE searchname, searchicpp, choice, contact, email, cnt
    DISPLAY ("Please enter name to search:")
    READ searchname
    DISPLAY ("Please enter IC or PASSPORT NO to search:")
    READ searchicpp
    cnt=0
    LOOP i in range (len(pallrec))
      IF searchname in pallrec[i][0] AND searchicpp in pallrec[i][5] THEN
        DISPLAY ("Please confirm this is you before make any changes:")
        DISPLAY ("Name : ", pallrec[i][0])
        DISPLAY ("Patient ID: ", pallrec[i][2])
        DISPLAY ("1-Contact ", pallrec[i][6])
        DISPLAY ("2-Email ", pallrec[i][7])
      WHILE TRUE
        DISPLAY ("Enter the no. of field to modify:")
        READ idx
      IF idx=="1" THEN
        WHILE TRUE
          DISPLAY ("Enter your mobile number:(ex. 012-23456789)")
          READ contact
          TRY
            IF contact[:2]=="01" AND contact[3]=="-" AND type(int(contact[4:]))==int AND len(contact)==11 OR len(contact)==12 THEN
              DISPLAY ("Your contact is:", contact)
              DISPLAY ("Press 1 to continue:")
              READ choice
              IF choice=="1" THEN
                pallrec[i][6]=contact
                break
              ELSE THEN
                DISPLAY ("Please reenter your mobile number.")
              ENDIF
            ELSE THEN
              DISPLAY ("INVALID INPUT!")
            ENDIF
          EXCEPT
            DISPLAY ("INVALID INPUT!")
          ENDTRY
        ENDWHILE
        break
      ELSE IF idx=="2" THEN
        WHILE TRUE
          DISPLAY ("Enter your e-mail address:")
          READ email
          flag=0
          cnt=0
          LOOP j in email
            IF j=="@" THEN
              flag=1
              break
            ELSE THEN
              cnt=cnt+1
            ENDIF
            NEXT j
          ENDLOOP
          IF cnt!=0 AND flag==1 THEN
            LOOP j in email[cnt+2:-1]
              IF j==". " THEN
                flag=2
                break
              ENDIF
            NEXT j
          ENDLOOP
          ENDIF
          IF flag==2 THEN
            DISPLAY ("Your email is:", email)
            DISPLAY ("Press 1 to continue:")
            READ choice
            IF choice=="1" THEN
              pallrec[i][7]=email
              break
            ELSE THEN
              DISPLAY ("Please reenter your email.")
            ENDIF
          ELSE THEN
            DISPLAY ("INVALID INPUT!")
          ENDIF
        ENDWHILE
        break
      ELSE THEN
        DISPLAY ("INVALID INPUT!")
      ENDIF
      ENDWHILE
      break
    ENDIF
    NEXT i
  ENDLOOP
  IF cnt==len(pallrec) THEN
    DISPLAY ("Record not found!")
  ENDIF
  WITH OPEN ('patient.txt') in write mode as fhp
    LOOP prec in pallrec
      fhp.write(":.".join(prec))
      NEXT prec
    ENDLOOP
  ENDWITH
  RETURN pallrec
ENDDEFINE
END

```

modify_login(plogallrec)

```

BEGIN
    DEFINE FUNCTION modify_login(plogallrec)
        DEFINE LIST plogallrec, fhplog, plogrec
        DEFINE searchname, searchpw, choice, newname, choice, newpw, cnt
        DISPLAY ("Please enter current username:")
        READ searchname
        cnt=0
        LOOP i in range (len(plogallrec))
            IF searchname in plogallrec[i][0] THEN
                DISPLAY ("Please enter current password:")
                READ searchpw
            IF searchpw in plogallrec[i][1] THEN
                DISPLAY ("Modify field:")
                DISPLAY ("1-username")
                DISPLAY ("2-password")
                WHILE TRUE
                    DISPLAY ("Please enter your choice:")
                    READ choice
                    IF choice == "1" THEN
                        WHILE TRUE
                            DISPLAY ("Please enter your new username:")
                            READ newname
                            DISPLAY ("Please confirm your username:")
                            READ confirm
                            IF newname == confirm THEN
                                flag=0
                                LOOP i in range(len(plogallrec))
                                    IF username in plogallrec[i][0] THEN
                                        DISPLAY ("SIGN UP FAILD! Current username is used.")
                                        flag=1
                                        break
                                    ENDIF
                                    NEXT i
                                ENDOLOOP
                                IF flag==0 THEN
                                    plogallrec[i][0]=newname
                                    break
                                ELSE THEN
                                    DISPLAY("Please reenter username.")
                                ENDIF
                            ELSE THEN
                                DISPLAY ("INPUT WRONG!")
                            ENDIF
                        ENDWHILE
                        break
                    ELSE IF choice == "2" THEN
                        WHILE TRUE
                            DISPLAY ("Please enter your new password:")
                            READ newpw
                            DISPLAY ("Please confirm your password:")
                            READ confirm
                            IF newpw == confirm THEN
                                plogallrec[i][1]=newpw
                                break
                            ELSE THEN
                                DISPLAY ("INPUT WRONG!")
                            ENDIF
                        ENDWHILE
                        break
                    ELSE THEN
                        DISPLAY ("INVALID INPUT!")
                    ENDIF
                ENDWHILE
                break
            ENDIF
        ELSE THEN
            cnt=cnt+1
        ENDIF
        NEXT i
    ENDOLOOP
    IF cnt==len(plogallrec) THEN
        DISPLAY ("Record not found!")
    ENDIF
    WITH OPEN (patientlogin.txt) as fhplog
        LOOP plogrec in plogallrec
            fhplog.write(":.".join(plogrec))
            NEXT plogrec
        ENDOLOOP
    ENDWITH
    RETURN plogallrec
ENDDEFINE
END

```

patient_check_status(vallrec)

```

BEGIN
    DEFINE FUNCTION patient_check_status(vallrec)
        DEFINE LIST vallrec
        DEFINE searchicpp, searchid, i,
        DISPLAY ("Registered IC/passport:")
        READ searchicpp
        DISPLAY ("Patient ID:")
        READ searchid
        LOOP i in range (len(vallrec))
            IF searchid in vallrec[i][0] AND searchicpp in vallrec[i][12] THEN
                DISPLAY ("PID      :", vallrec[i][0])
                DISPLAY ("CENTRE   :", vallrec[i][1])
                DISPLAY ("VACCINE  :", vallrec[i][2])
                DISPLAY ("D1 DATE   :", vallrec[i][4])
                DISPLAY ("D1 STATUS:", vallrec[i][5])
                DISPLAY ("D2 DATE   :", vallrec[i][8])
                DISPLAY ("D2 STATUS:", vallrec[i][9])
                DISPLAY ("VACCINE STATUS:", vallrec[i][11])
                break
            ENDIF
            NEXT i
        ENDOLOOP
    ENDDEFINE
END

```

check_patient_status(vallrec)

```

BEGIN
    DEFINE FUNCTION check_patient_status(vallrec)
        DEFINE LIST vallrec
        DEFINE search, i
        DISPLAY ("Patient ID:")
        READ search
        LOOP i in range (len(vallrec))
            IF search in vallrec[i][0] THEN
                DISPLAY ("PID      :", vallrec[i][0])
                DISPLAY ("CENTRE   :", vallrec[i][1])
                DISPLAY ("VACCINE  :", vallrec[i][2])
                DISPLAY ("D1 DATE   :", vallrec[i][4])
                DISPLAY ("D1 STATUS:", vallrec[i][5])
                DISPLAY ("D2 DATE   :", vallrec[i][8])
                DISPLAY ("D2 STATUS:", vallrec[i][9])
                DISPLAY ("VACCINE STATUS:", vallrec[i][11])
                break
            ENDIF
            NEXT i
        ENDOLOOP
    ENDDEFINE
END

```

search_patient_id(pallrec)

```

BEGIN
    DEFINE LIST pallrec
    DEFINE searchname, searchicpp
    DISPLAY ("Please enter name to search:")
    READ searchname
    DISPLAY ("Please enter IC or PASSPORT NO to search:")
    READ searchicpp
    LOOP i in range (len(pallrec))
        IF searchname in pallrec[i][0] AND searchicpp in pallrec[i][5] THEN
            DISPLAY ("Name:",pallrec[i][0])
            DISPLAY ("IC/passport:",pallrec[i][5])
            DISPLAY ("Patient ID:",pallrec[i][2])
            break
        ENDIF
        NEXT i
    ENDOLOOP
ENDDEFINE
END

```

display_patient_rec(pallrec)

```

BEGIN
    DEFINE FUNCTION display_patient_rec(pallrec)
        DEFINE LIST pallrec
        DEFINE i, cnt
        DISPLAY ("=*137")
        DISPLAY ("NO".center(10), "|", "NAME".center(30), "|", "AGE".center(4), "|", "PID".center(11), "|", "CENTRE".center(8), "|", "CODE".center(6),
                 "|", "IC/PP".center(16), "|", "CONTACT".center(14), "|", "EMAIL".center(30))
        DISPLAY ("=*137")
        cnt=1
        LOOP i in range (len(pallrec))
            DISPLAY (str(cnt).ljust(10), "|", pallrec[i][0].ljust(30), "|", pallrec[i][1].center(4), "|", pallrec[i][2].center(11), "|", pallrec[i][3].center(8),
                     "|", pallrec[i][4].center(6), "|", pallrec[i][5].center(16), "|", pallrec[i][6].ljust(14), "|", pallrec[i][7].ljust(30))
            cnt=cnt+1
        NEXT i
    ENDOLOOP
    DISPLAY ("=*137")
ENDDEFINE
END

```

display_vac_rec(vallrec)

```

BEGIN
    DEFINE FUNCTION display_vac_rec(vallrec)
        DEFINE LIST vallrec
        DEFINE i, cnt
        DISPLAY ("=*140")
        DISPLAY ("NO", "PID".center(11), "|", "CENTRE".center(8), "|", "CODE".center(6), "|", "D1".center(20), "|", "D1_STA".center(8),
                 "|", "D1_BATCH".center(12), "|", "D2".center(20), "|", "D2_STA".center(8), "|", "D2_BATCH".center(12), "|", "STATUS".center(16))
        DISPLAY ("=*140")
        cnt=1
        LOOP i in range (len(pallrec))
            DISPLAY (str(cnt).ljust(10), "|", vallrec[i][0].center(11), "|", vallrec[i][1].center(8), "|", vallrec[i][2].center(6), "|", vallrec[i][4].ljust(20),
                     "|", vallrec[i][5].ljust(8), "|", vallrec[i][6].ljust(12), "|", vallrec[i][8].ljust(20), "|", vallrec[i][9].ljust(8), "|",
                     vallrec[i][10].ljust(12), "|", vallrec[i][11].ljust(16))
            cnt=cnt+1
        NEXT i
    ENDOLOOP
    DISPLAY ("=*140")
ENDDEFINE
END

```

vaccination_administration(vallrec)

```

BEGIN
    DEFINE FUNCTION vaccination_administration(vallrec)
        DEFINE LIST vallrec, fhv, vrec
        DEFINE search, i, done, batch, choice, d1, dldate, d2, d2date
        DISPLAY ("Patient ID:")
        READ search
        LOOP i in range (len(vallrec))
            IF search in vallrec[i][0] THEN
                DISPLAY ("PID      :", vallrec[i][0])
                DISPLAY ("CENTRE   :", vallrec[i][1])
                DISPLAY ("VACCINE  :", vallrec[i][2])
                DISPLAY ("D1 DATE   :", vallrec[i][4])
                DISPLAY ("D1 STATUS:", vallrec[i][5])
                DISPLAY ("D2 DATE   :", vallrec[i][8])
                DISPLAY ("D2 STATUS:", vallrec[i][9])
                DISPLAY ("VACCINE STATUS:", vallrec[i][11])
            IF vallrec[i][5]=="NO" THEN
                DISPLAY ("D1 vaccination.")
                DISPLAY ("Enter 1 when D1 is done:")
                READ done
            IF done == "1" THEN
                WHILE TRUE
                    DISPLAY ("Enter vaccine batch number:")
                    READ batch
                    DISPLAY ("Vaccine batch number:", batch)
                    DISPLAY ("Press 1 to continue:")
                    READ choice
                    IF choice=="1" THEN
                        vallrec[i][6]=batch
                        break
                    ELSE THEN
                        DISPLAY ("Please reenter vaccine batch number.")
                    ENDIF
                ENDWHILE
            vallrec[i][5]="YES"
            d1 = datetime.datetime.today()
            dldate = d1.strftime("%d/%m/%Y, %H%M")
            vallrec[i][4]=dldate
            IF vallrec[i][2] == "AF" THEN
                d2 = d1 + datetime.timedelta(days=14)
                vallrec[i][8] = d2.strftime("%d/%m/%Y")
                DISPLAY ("D2 DATE:", vallrec[i][8])
                vallrec[i][11] = "COMPLETED-D1"
                DISPLAY ("STATUS:", vallrec[i][11])
            ELSE IF vallrec[i][2] == "BV" THEN
                d2 = d1 + datetime.timedelta(days=21)
                vallrec[i][8] = d2.strftime("%d/%m/%Y")
                DISPLAY ("D2 DATE:", vallrec[i][8])
                vallrec[i][11] = "COMPLETED-D1"
                DISPLAY ("STATUS:", vallrec[i][11])
            ELSE IF vallrec[i][2] == "CZ" THEN
                d2 = d1 + datetime.timedelta(days=21)
                vallrec[i][8] = d2.strftime("%d/%m/%Y")
                DISPLAY ("D2 DATE:", vallrec[i][8])
                vallrec[i][11] = "COMPLETED-D1"
            ENDIF
        ENDLOOP
    ENDFUNCTION

```

```
        DISPLAY ("STATUS:", vallrec[i][11])
    ELSE IF vallrec[i][2] == "DM" THEN
        d2 = d1 + datetime.timedelta(days=28)
        vallrec[i][8] = d2.strftime("%d/%m/%Y")
        DISPLAY ("D2 DATE:", vallrec[i][8])
        vallrec[i][11] = "COMPLETED-D1"
        DISPLAY ("STATUS:", vallrec[i][11])
    ELSE IF vallrec[i][2] == "EC" THEN
        vallrec[i][8] = "NONE"
        vallrec[i][9] = "NONE"
        vallrec[i][10] = "NONE"
        vallrec[i][11] = "COMPLETED"
        DISPLAY ("D2 DATE:", vallrec[i][8])
        DISPLAY ("STATUS:", vallrec[i][11])
    ENDIF
    ELSE THEN
        DISPLAY ("D1 vaccination failed. Exiting to menu...")
    ENDIF
    ELSE IF vallrec[i][5]=="YES" AND vallrec[i][9]=="NO" THEN
        DISPLAY ("D2 vaccination.")
        DISPLAY ("Enter 1 when D2 is done:")
        READ done
        IF done == "1" THEN
            WHILE TRUE
                DISPLAY ("Enter vaccine batch number:")
                READ batch
                DISPLAY ("Vaccine batch number:", batch)
                DISPLAY ("Press 1 to continue:")
                READ choice
                IF choice=="1" THEN
                    vallrec[i][10]=batch
                    break
                ELSE THEN
                    DISPLAY ("Please reenter vaccine batch number.")
                ENDIF
            ENDWHILE
            vallrec[i][9]="YES"
            vallrec[i][11]="COMPLETED"
            d2 = datetime.datetime.today()
            d2date = d2.strftime("%d/%m/%Y,%H%M")
            vallrec[i][8]=d2date
            DISPLAY ("STATUS:", vallrec[i][11])
        ELSE THEN
            DISPLAY ("D2 vaccination failed.Exiting to menu...")
        ENDIF
    ELSE THEN
        DISPLAY ("Vaccination Completed.")
    ENDIF
    break
ENDIF
NEXT i
ENDLOOP
WITH OPEN (vaccination.txt) in write mode as fhv
    LOOP vrec in vallrec
        fhv.write(":".join(vrec)+"\n")
    NEXT vrec
ENDLOOP
ENDWITH
RETURN vallrec
ENDDEFINE
END
```

check_overall_status(vallrec)

```

BEGIN
    DEEFINE FUNCTION check_overall_status(vallrec):
        DEFINE LIST vallrec
        DEFINE af, af1, af2, bv, bvl, bv2,, cz, cz1, cz2, dm, dm1, dm2, ec, ec1, ec2, vc1, vc2, vcld1, vcld1, vc2d1, vc2d2, total, i
        af=0
        af1=0
        af2=0
        bv=0
        bvl=0
        bv2=0
        cz=0
        cz1=0
        cz2=0
        dm=0
        dm1=0
        dm2=0
        ec=0
        ec1=0
        ec2=0
        vc1=0
        vc2=0
        vcld1=0
        vcld2=0
        vc2d1=0
        vc2d2=0
        total=len(vallrec)
    LOOP i in range(total)
        IF vallrec[i][2]==="AF" THEN
            af=af+1
            IF vallrec[i][11]==="COMPLETED" THEN
                af2=af2+1
            ELSE IF vallrec[i][5]==="YES" AND vallrec[i][9]==="NO" THEN
                af1=af1+1
            ENDIF
        ELSE IF vallrec[i][2]==="BV" THEN
            bv=bv+1
            IF vallrec[i][11]==="COMPLETED" THEN
                bv2=bv2+1
            ELSE IF vallrec[i][5]==="YES" AND vallrec[i][9]==="NO" THEN
                bvl=bvl+1
            ENDIF
        ELSE IF vallrec[i][2]==="CZ" THEN
            cz=cz+1
            IF vallrec[i][11]==="COMPLETED" THEN
                cz2=cz2+1
            ELSE IF vallrec[i][5]==="YES" AND vallrec[i][9]==="NO" THEN
                cz1=cz1+1
            ENDIF
        ELSE IF vallrec[i][2]==="DM" THEN
            dm=dm+1
            IF vallrec[i][11]==="COMPLETED" THEN
                dm2=dm2+1
            ELSE IF vallrec[i][5]==="YES" AND vallrec[i][9]==="NO" THEN
                dm1=dm1+1
            ENDIF
        ELSE IF vallrec[i][2]==="EC" THEN
            ec=ec+1
            IF vallrec[i][11]==="COMPLETED" THEN
                ec2=ec2+1
            ELSE IF vallrec[i][5]==="NO" THEN
                ec1=ec1+1
            ENDIF
        ENDIF
        NEXT i
    ENDLOOP
    LOOP i in range(total)
        IF vallrec[i][1]==="VC1" THEN
            vc1=vc1+1
            IF vallrec[i][11]==="COMPLETED-D1" THEN
                vcld1=vcld1+1
            ELSE IF vallrec[i][11]==="COMPLETED" THEN
                vcld2=vcld2+1
            ENDIF
        ELSE IF vallrec[i][1]==="VC2" THEN
            vc2=vc2+1
            IF vallrec[i][11]==="COMPLETED-D1" THEN
                vc2d1=vc2d1+1
            ELSE IF vallrec[i][11]==="COMPLETED" THEN
                vc2d2=vc2d2+1
            ENDIF
        ENDIF
        NEXT i
    ENDLOOP

```

```
DISPLAY ("Number of registration:",total)
DISPLAY ("AF:",af)
DISPLAY ("BV:",bv)
DISPLAY ("CZ:",cz)
DISPLAY ("DM:",dm)
DISPLAY ("EC:",ec)

DISPLAY ("All dose completed:",af2+bv2+cz2+dm2+ec2)
DISPLAY ("AF:",af2)
DISPLAY ("BV:",bv2)
DISPLAY ("CZ:",cz2)
DISPLAY ("DM:",dm2)
DISPLAY ("EC:",ec2)

DISPLAY ("One more dose left:",afl+bvl+czl+dml+ecl)
DISPLAY ("AF:",afl)
DISPLAY ("BV:",bvl)
DISPLAY ("CZ:",czl)
DISPLAY ("DM:",dml)
DISPLAY ("EC:",ecl)

DISPLAY ("Number of patients vaccinated in VC1:",vc1)
DISPLAY ("Waiting for dose 2:",vc1d1)
DISPLAY ("Completed vaccination:",vc1d2)

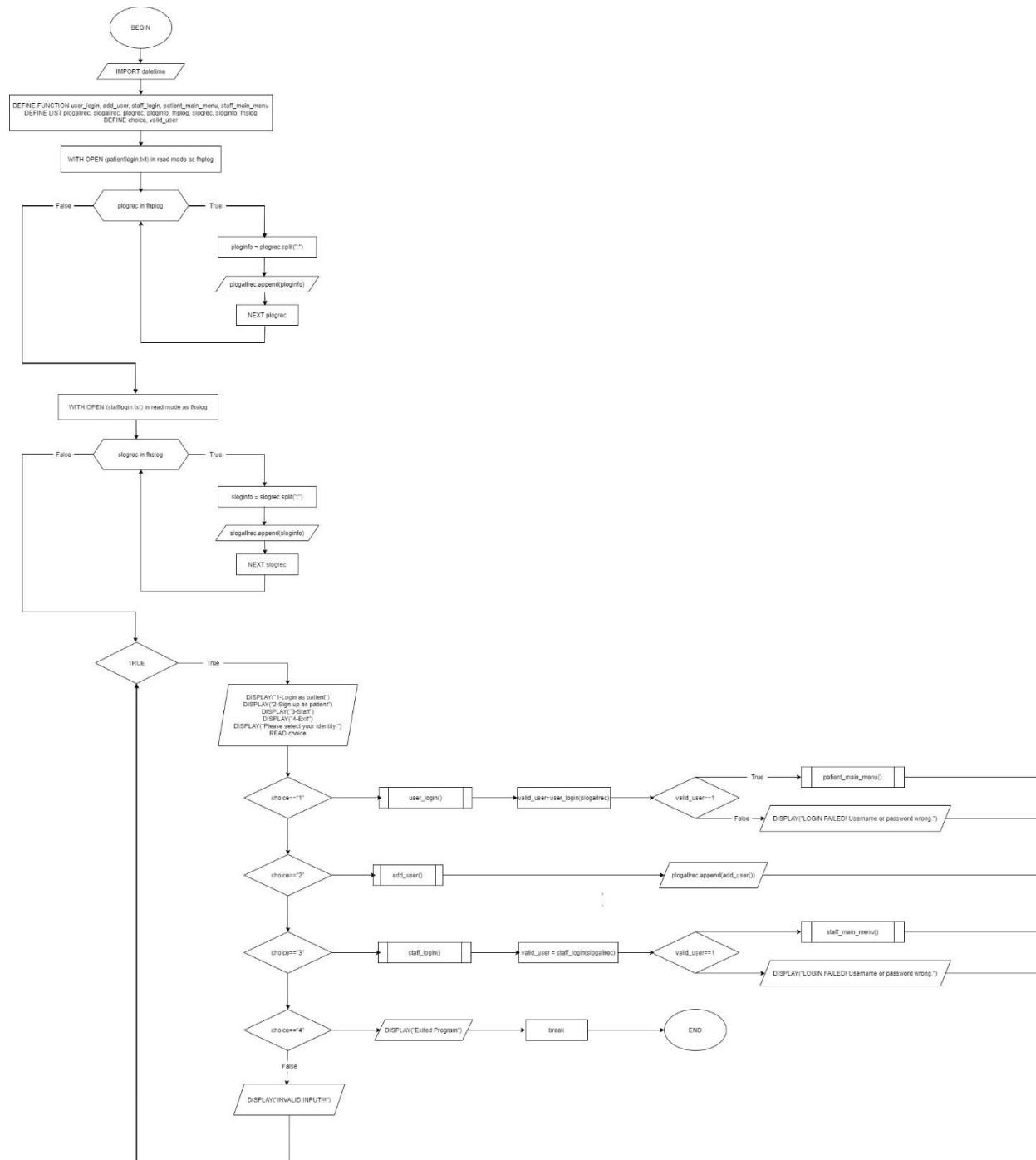
DISPLAY ("Number of patients vaccinated in VC2:",vc2)
DISPLAY ("Waiting for dose 2:",vc2d1)
DISPLAY ("Completed vaccination:",vc2d2)

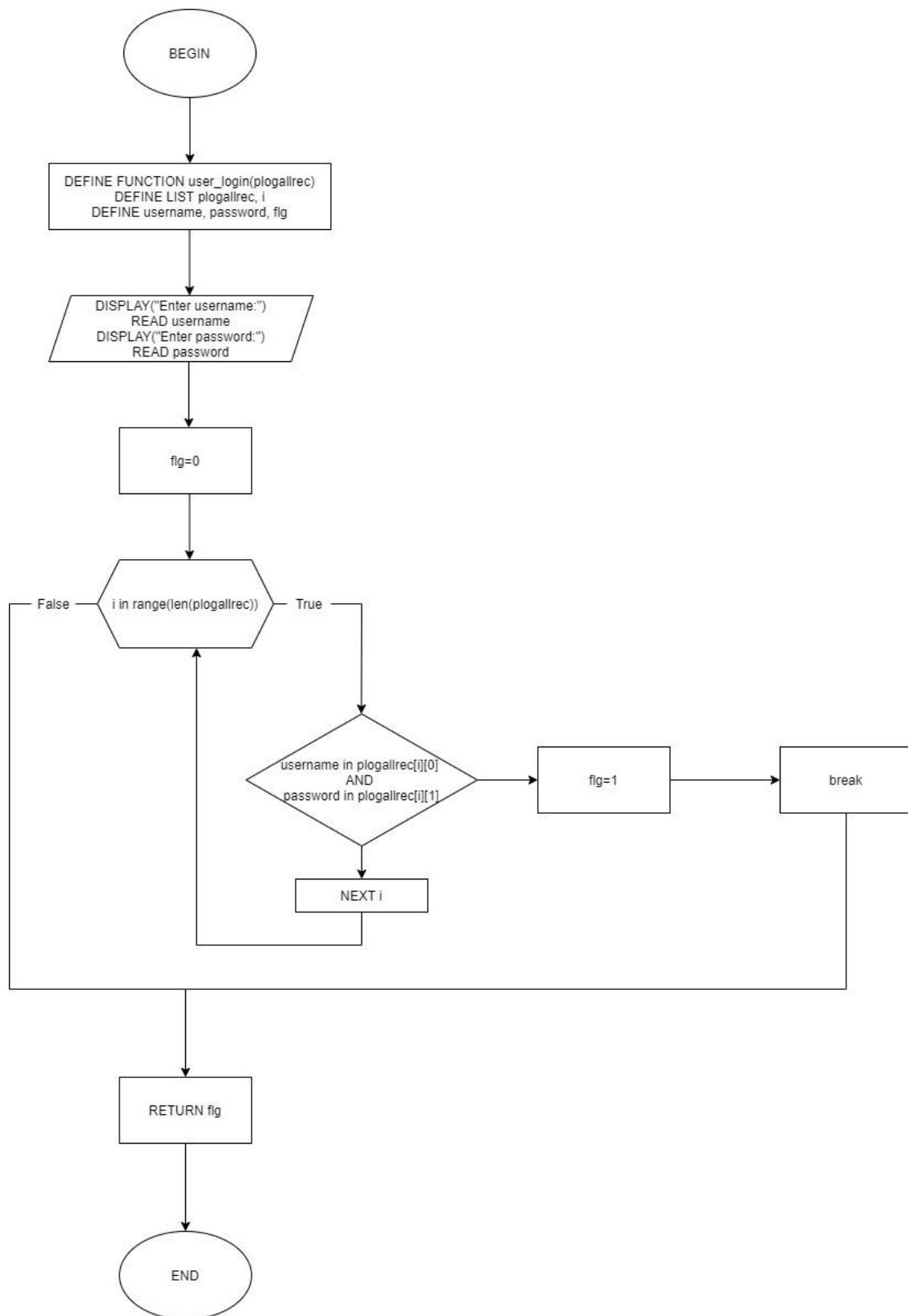
ENDDEFINE

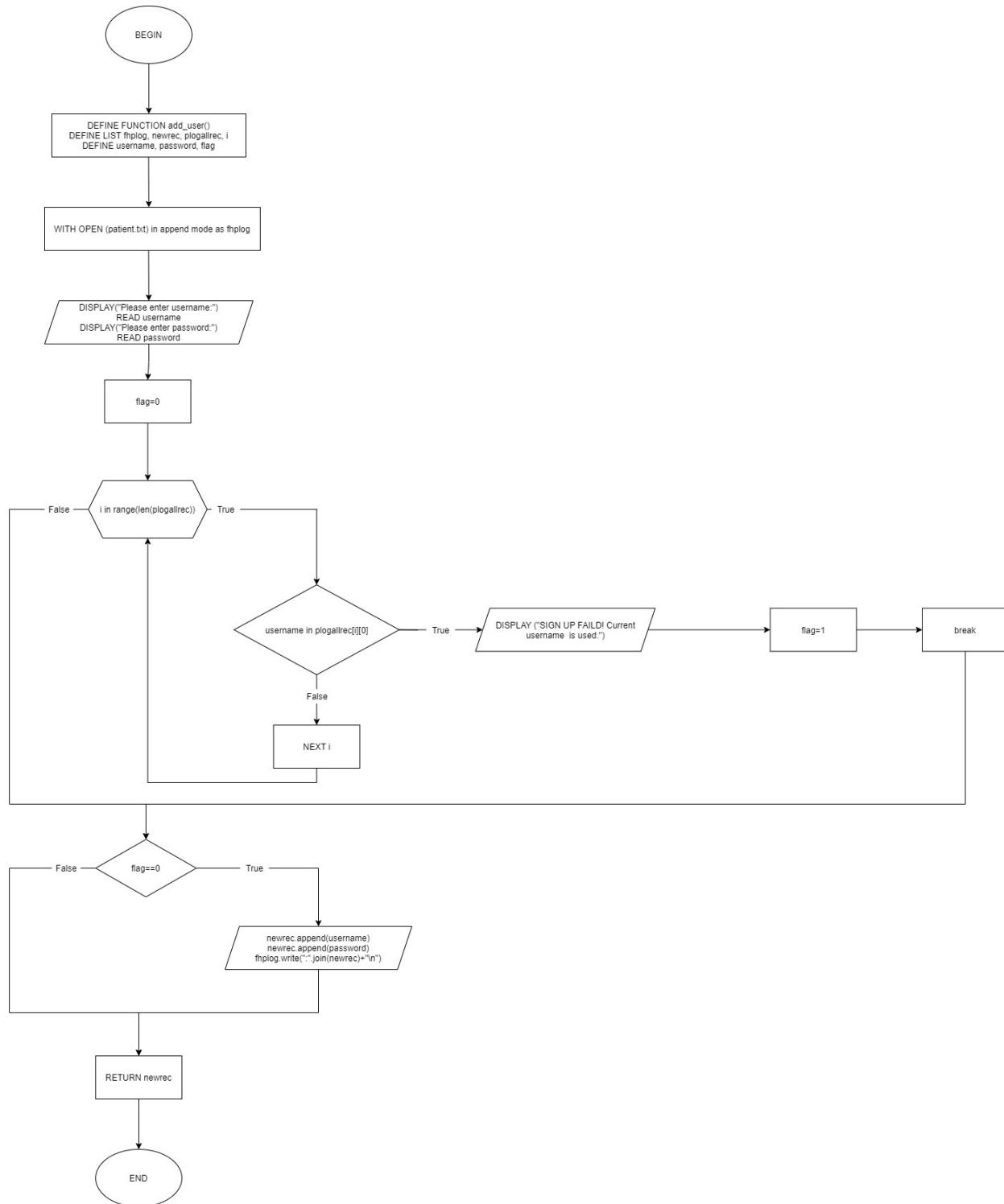
END
```

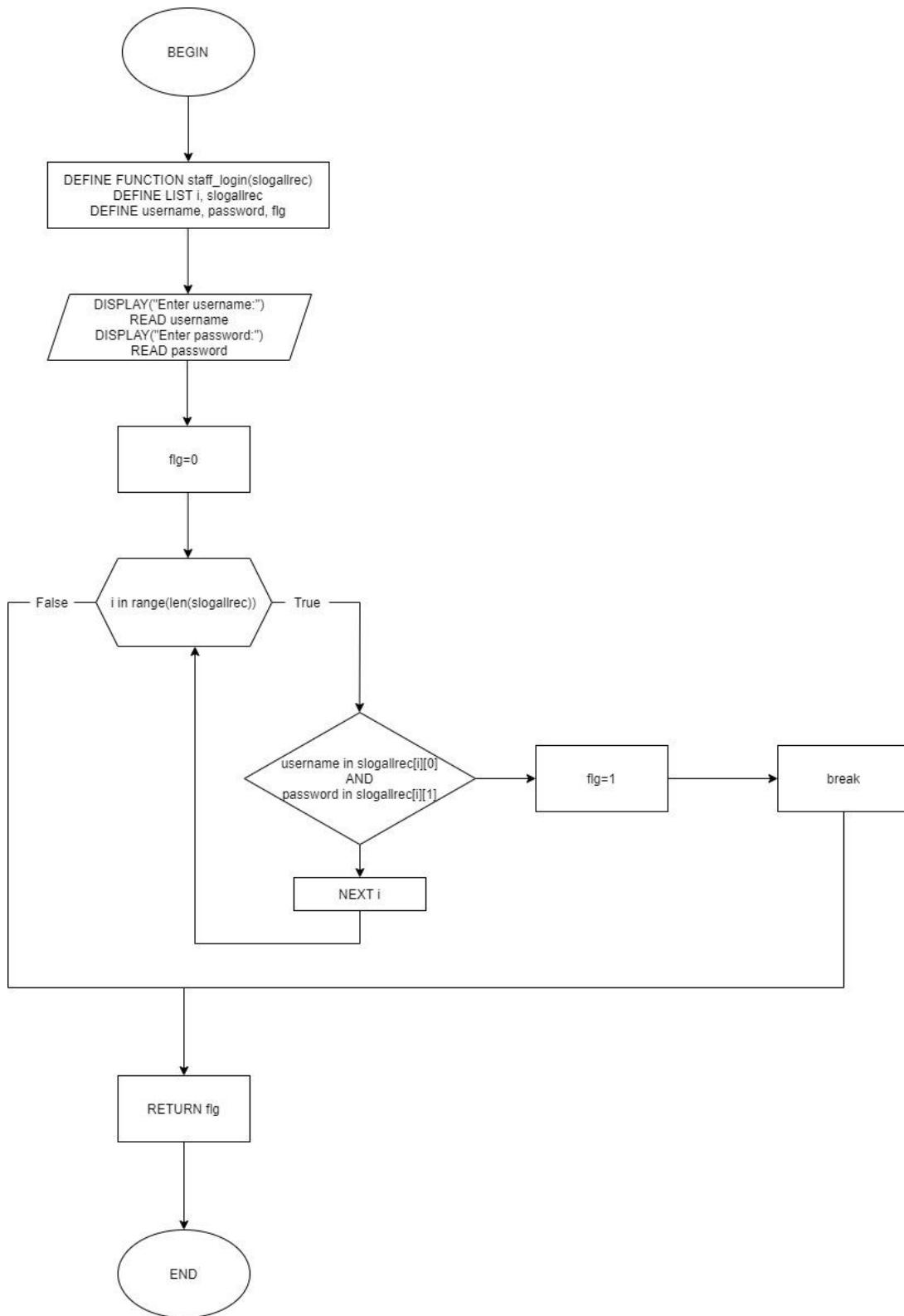
3.2 Flowchart

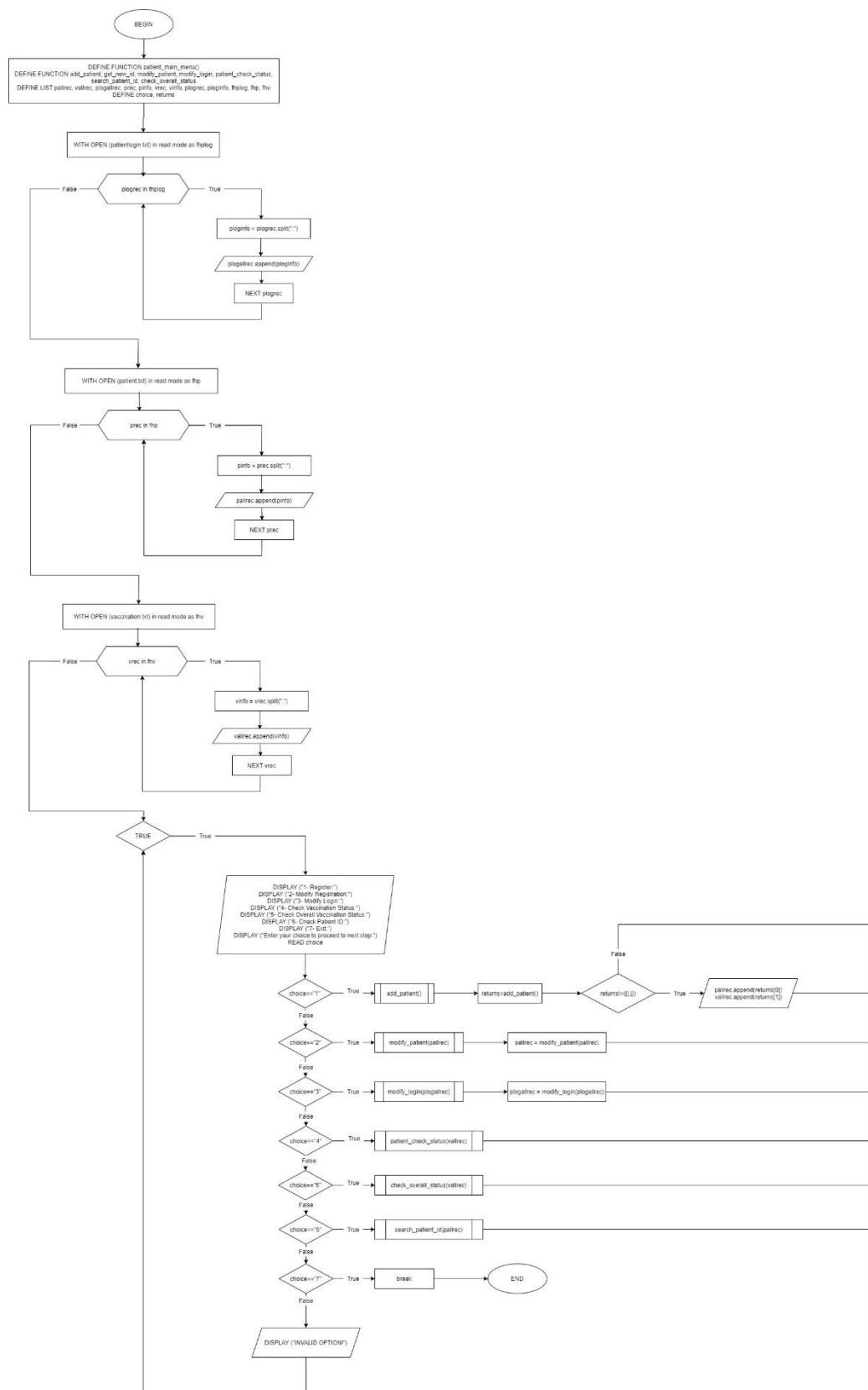
Main Logic

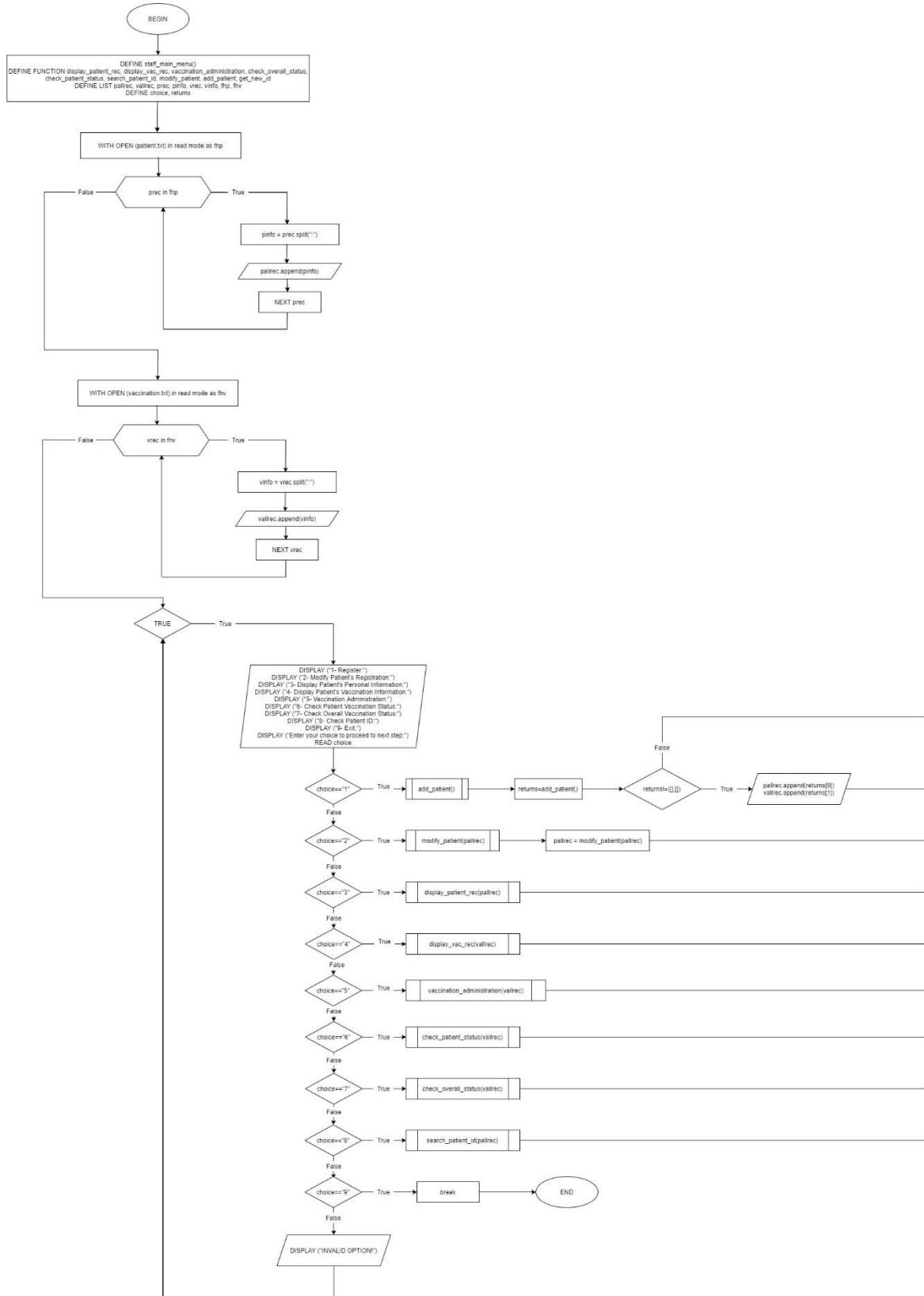


user_login(plogallrec)

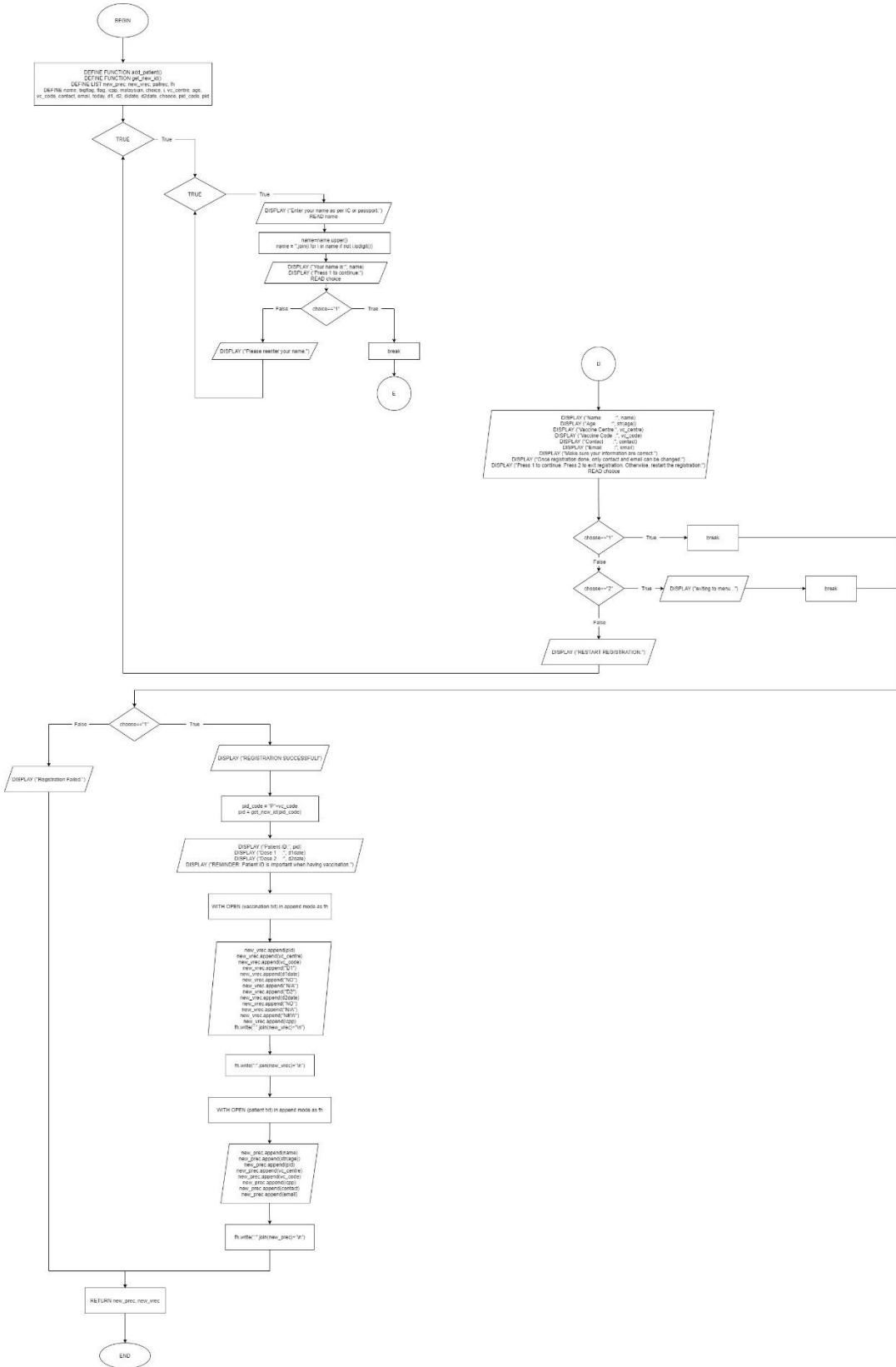
add_user()

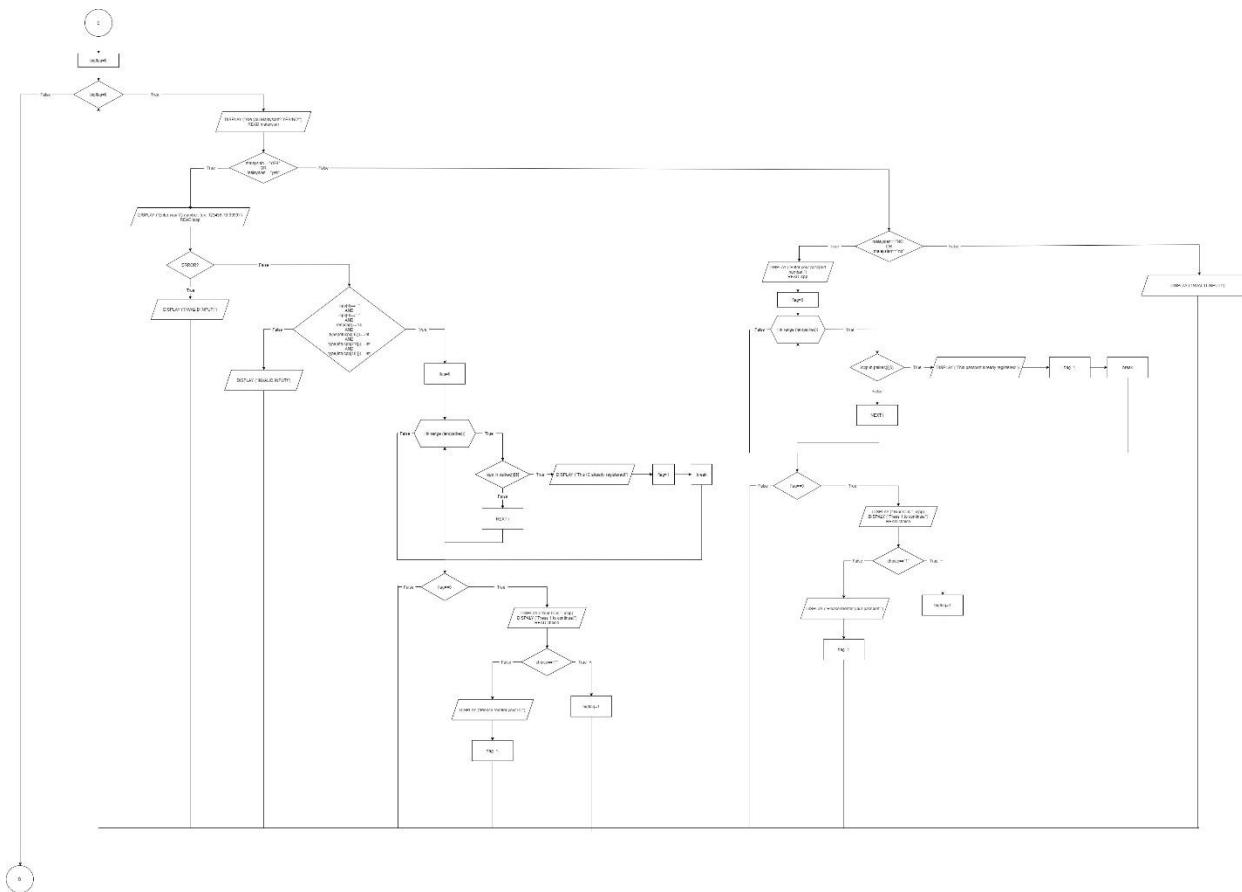
staff_login(slogallrec)

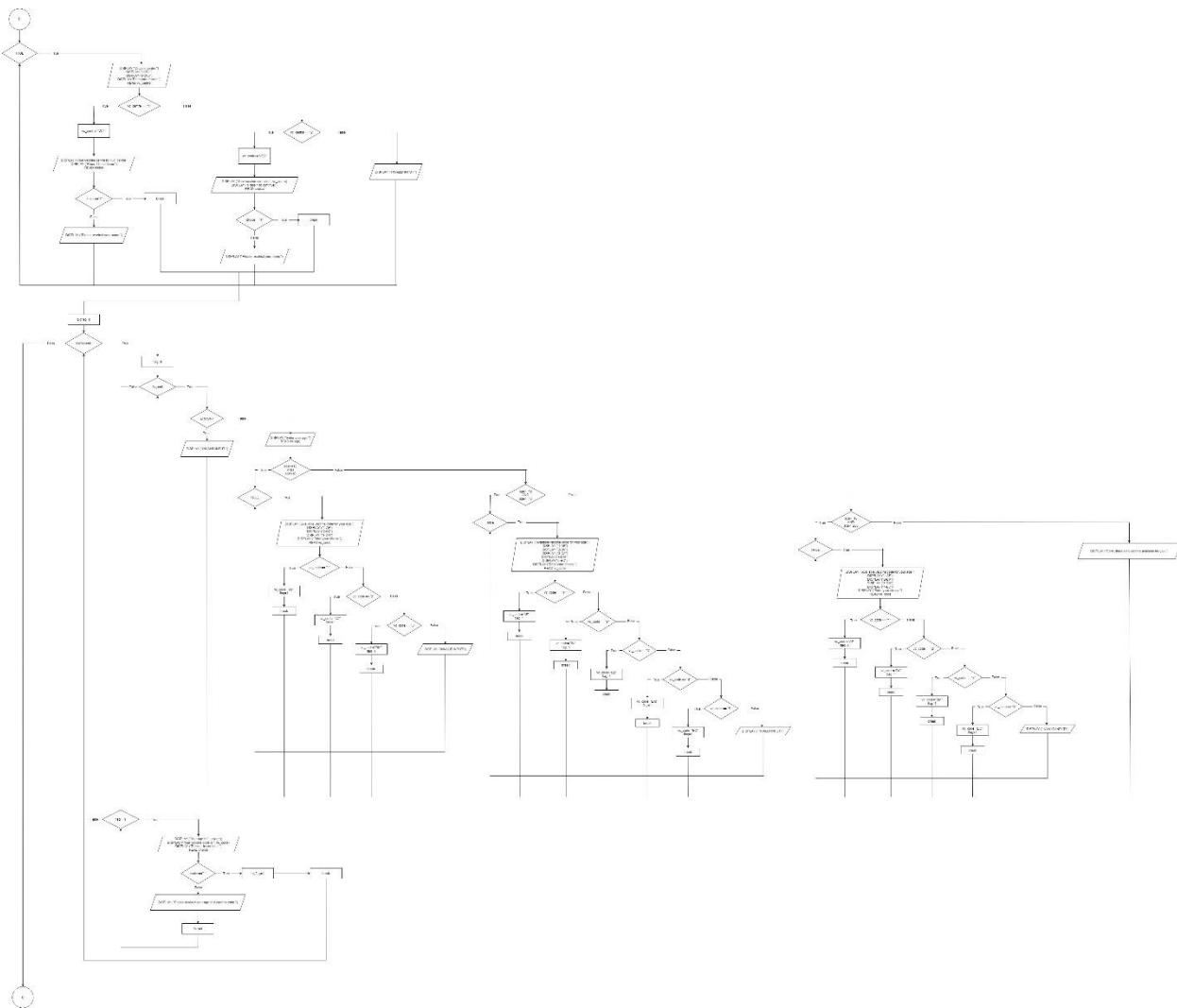
patient_main_menu

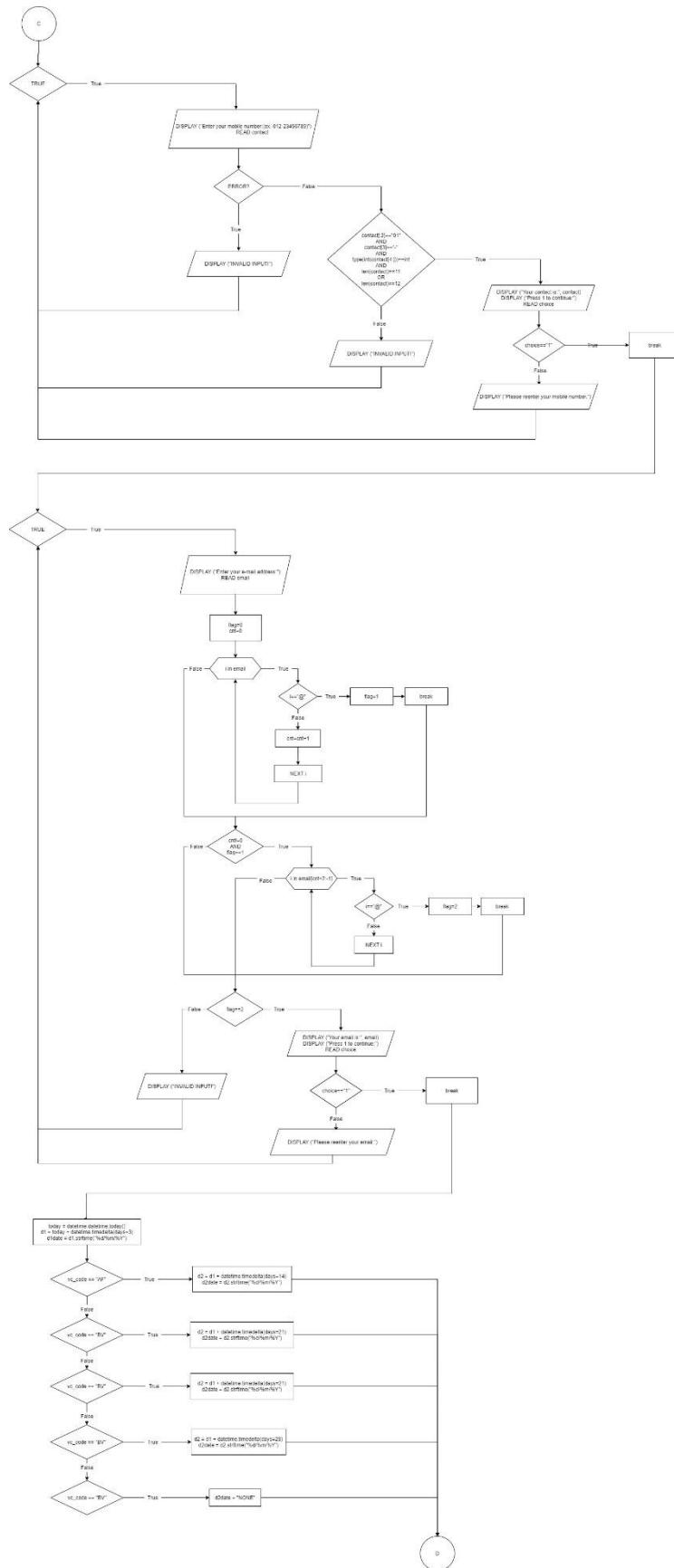
staff_main_menu()

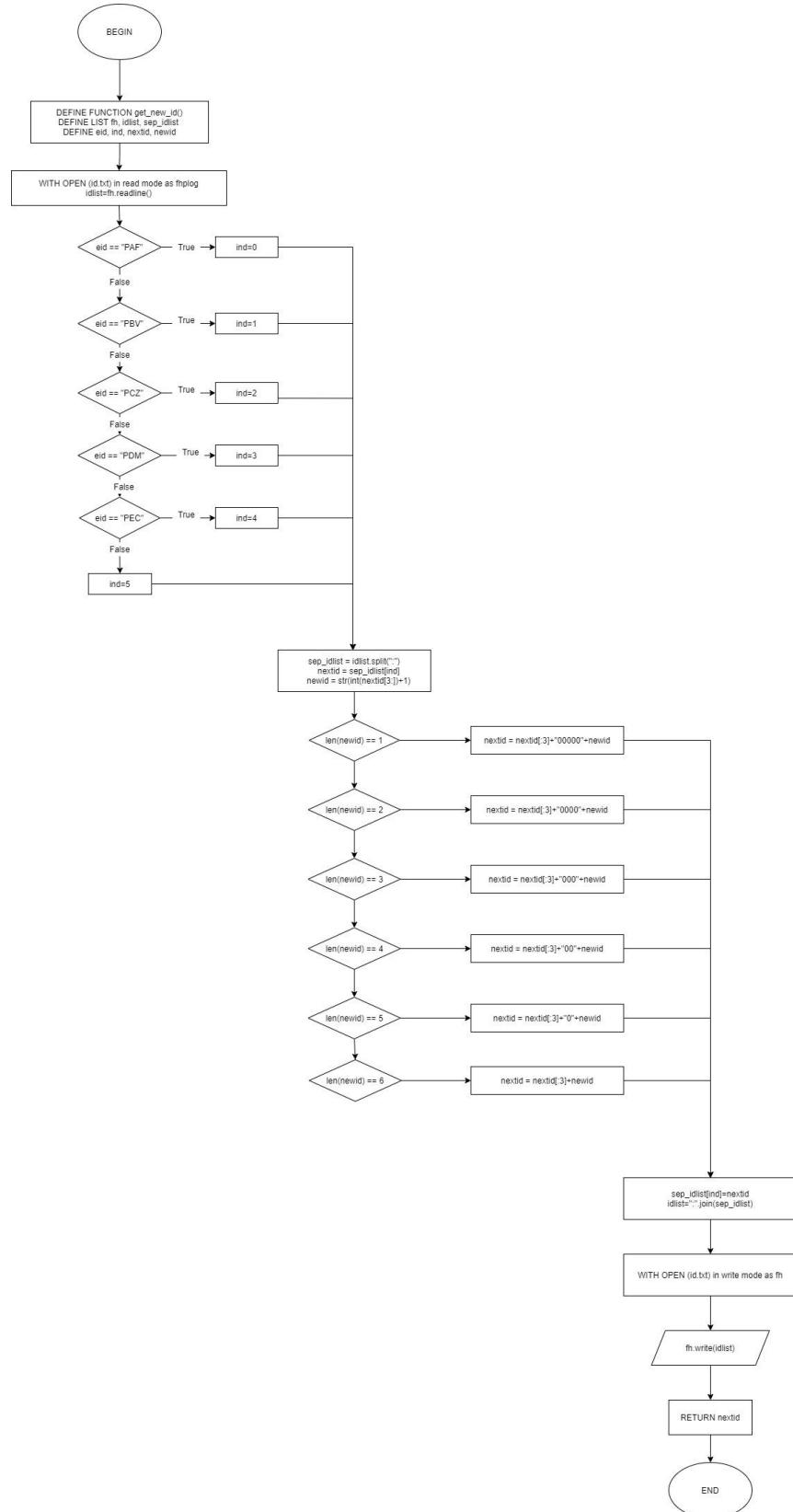
add_patient()

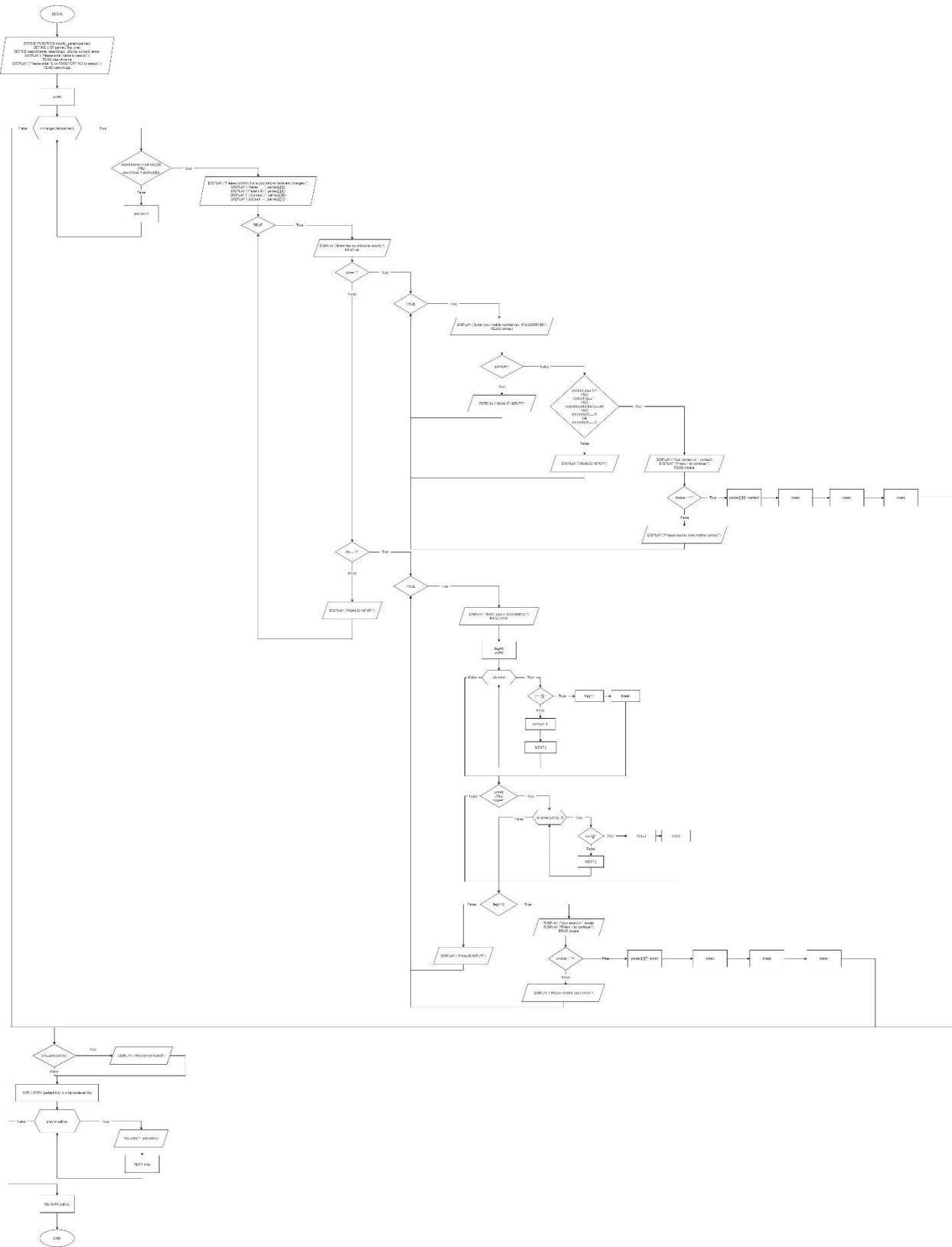


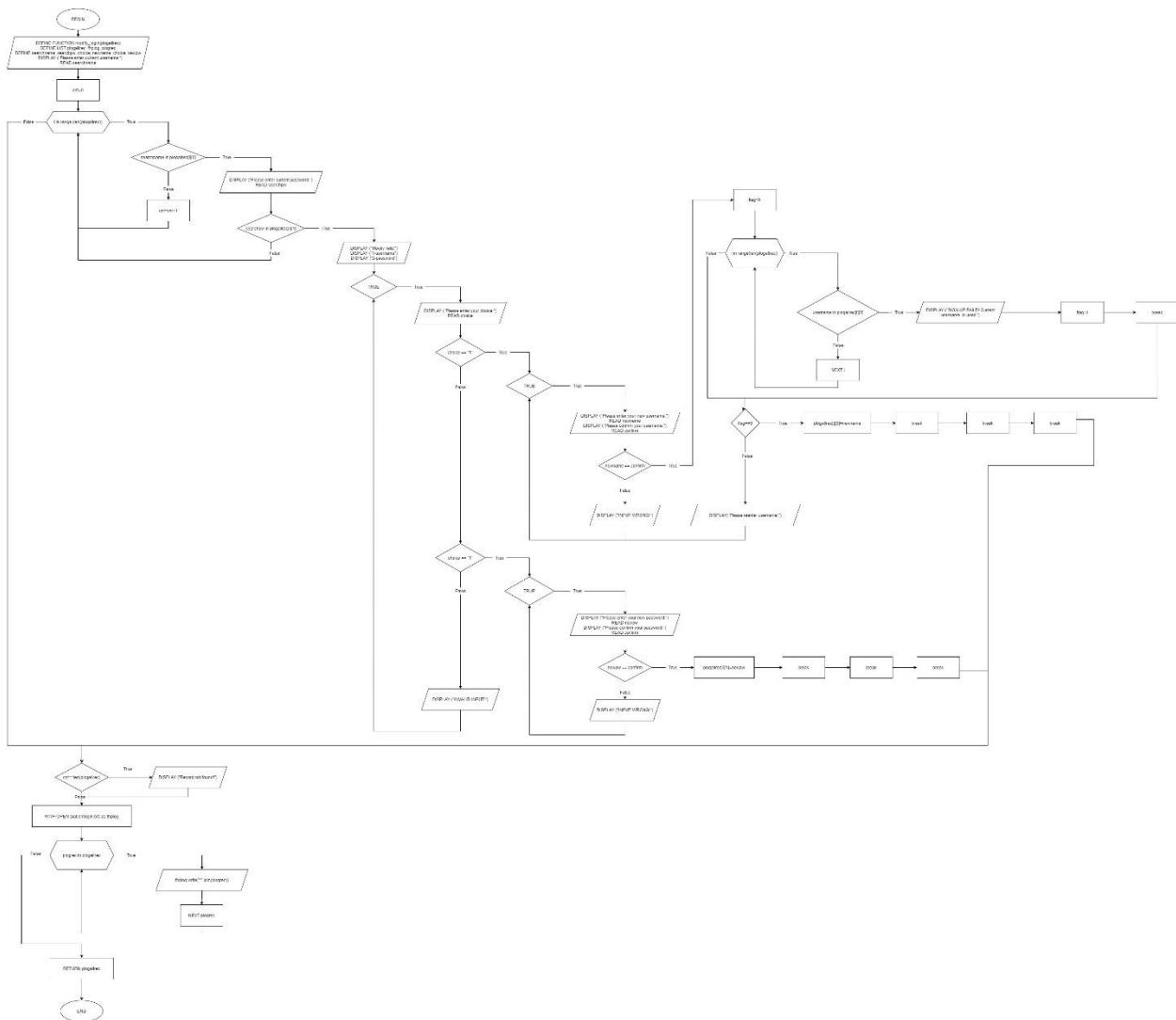


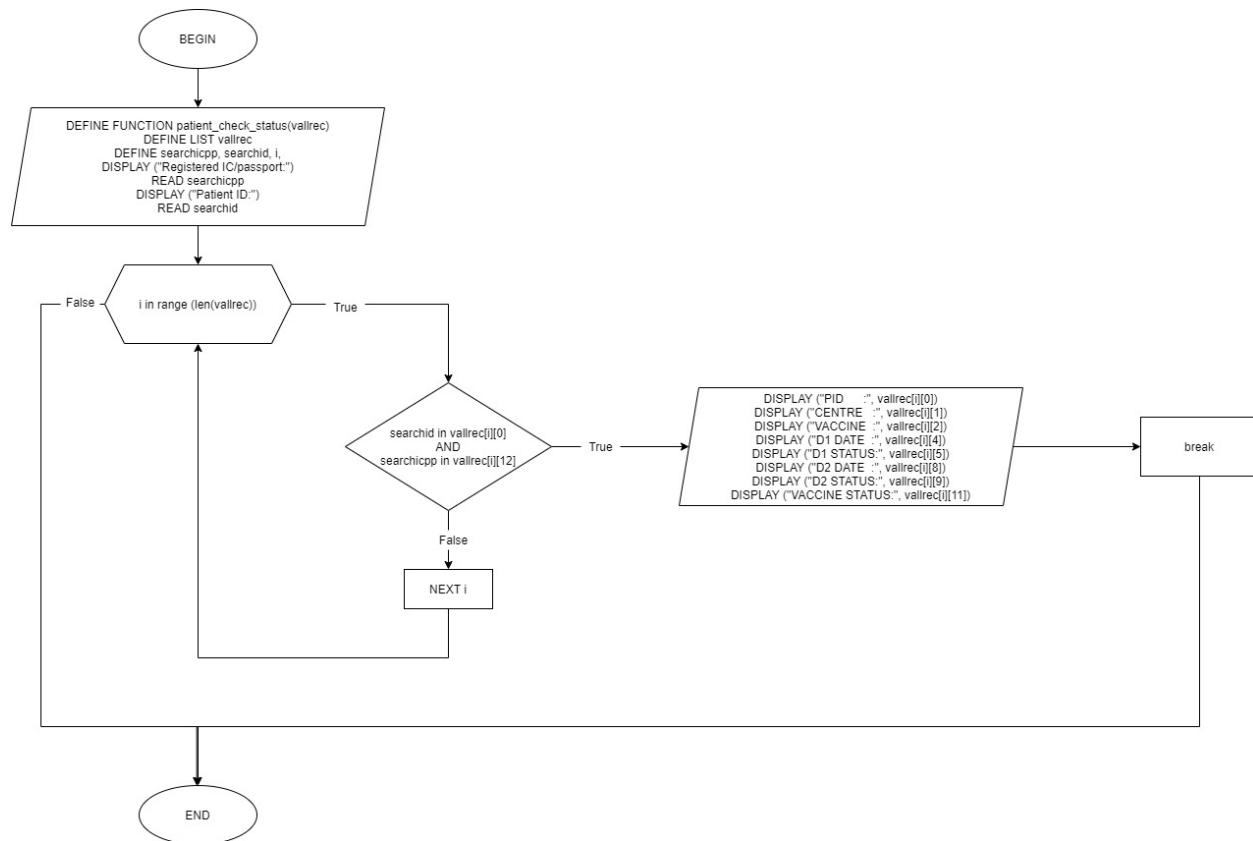


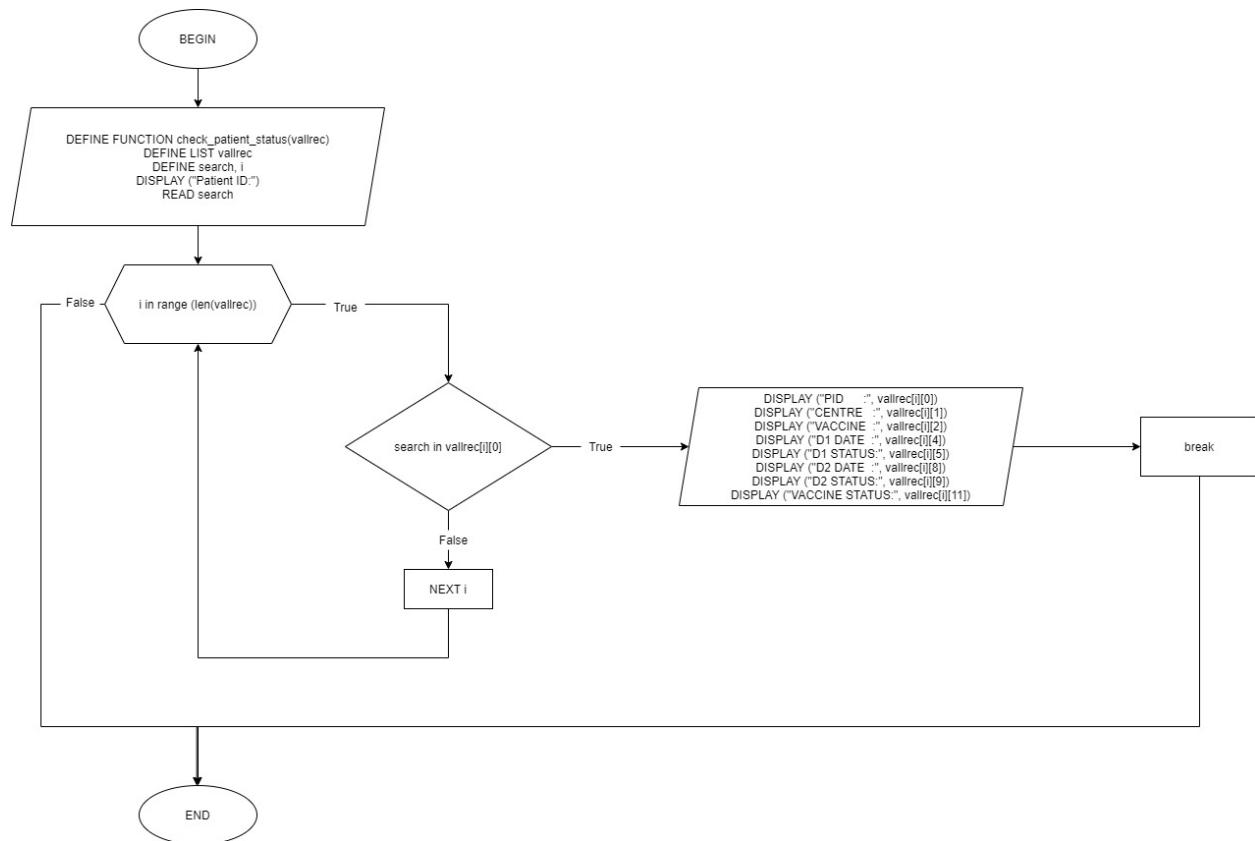


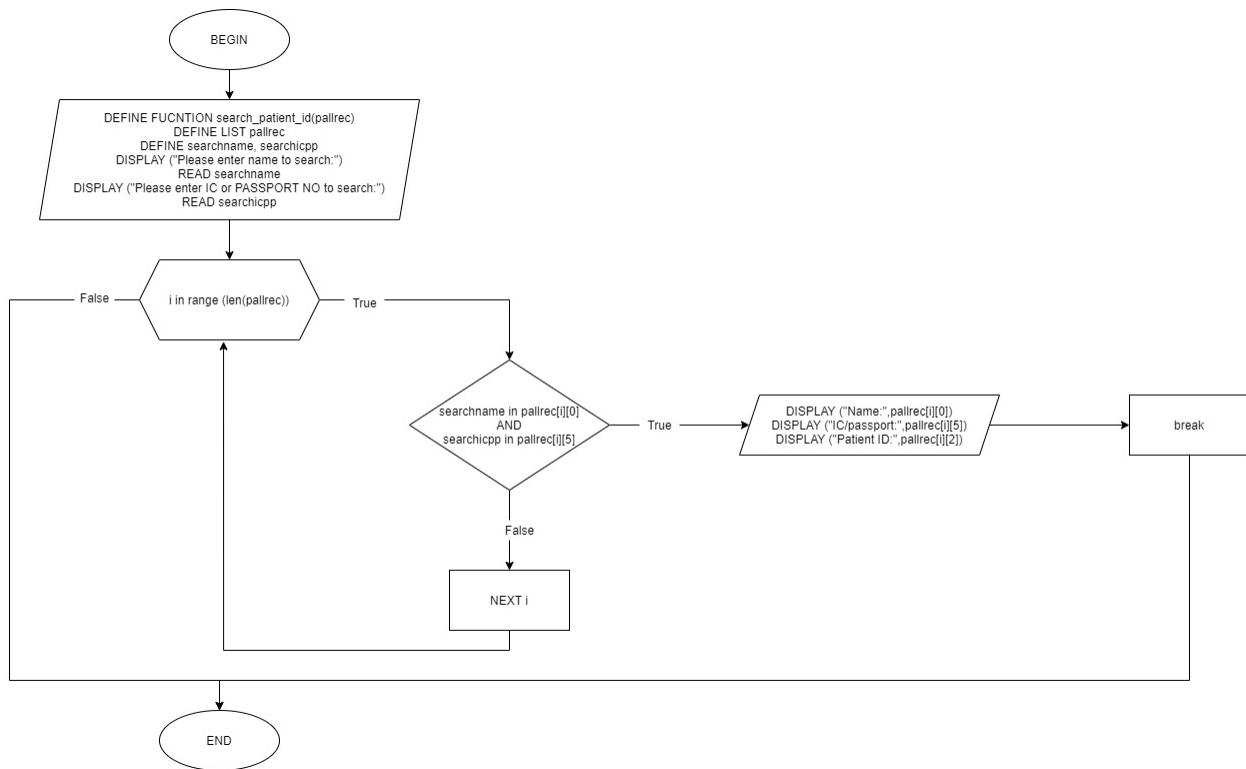
get_new_id(eid)

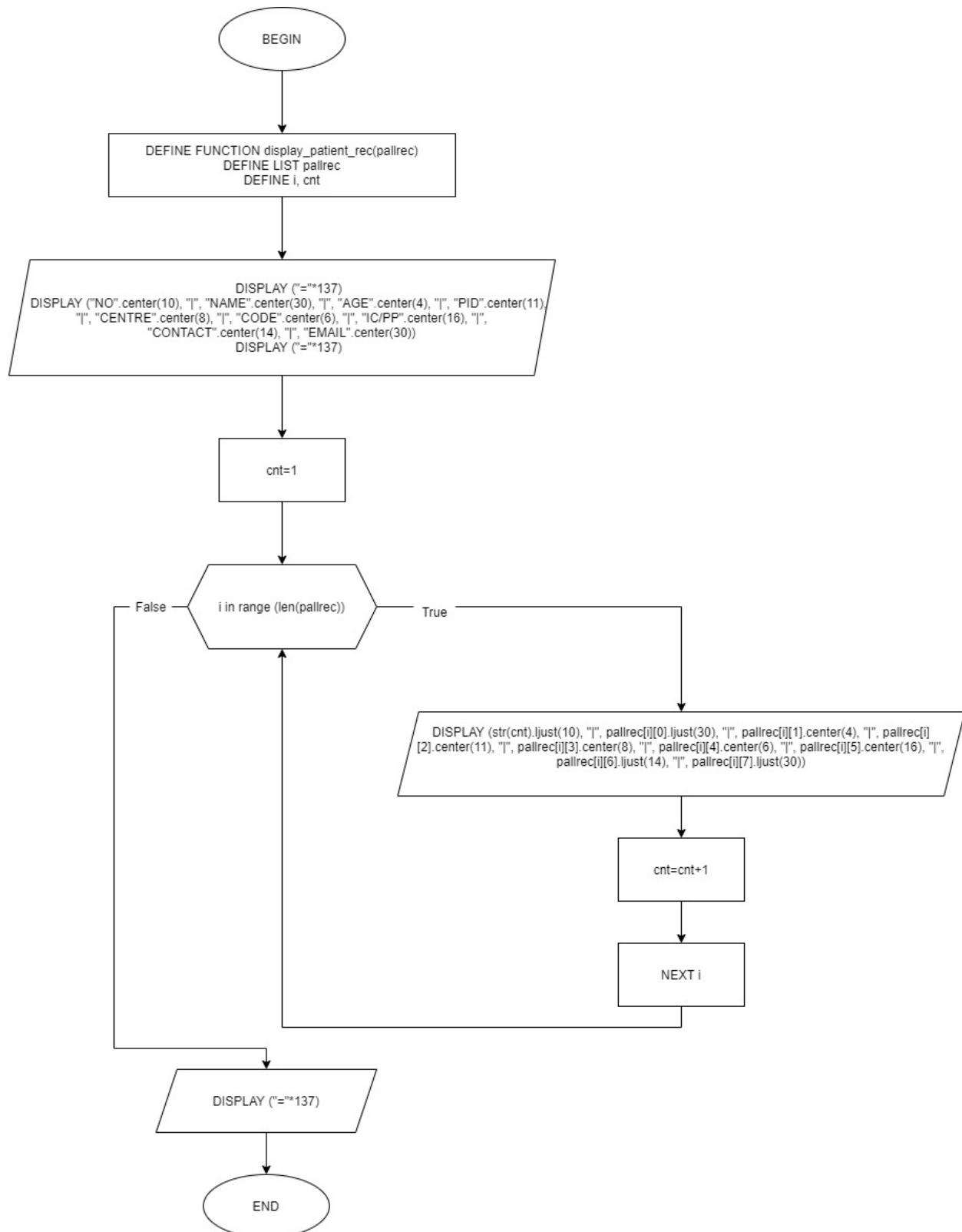
modify_patient(pallrec)

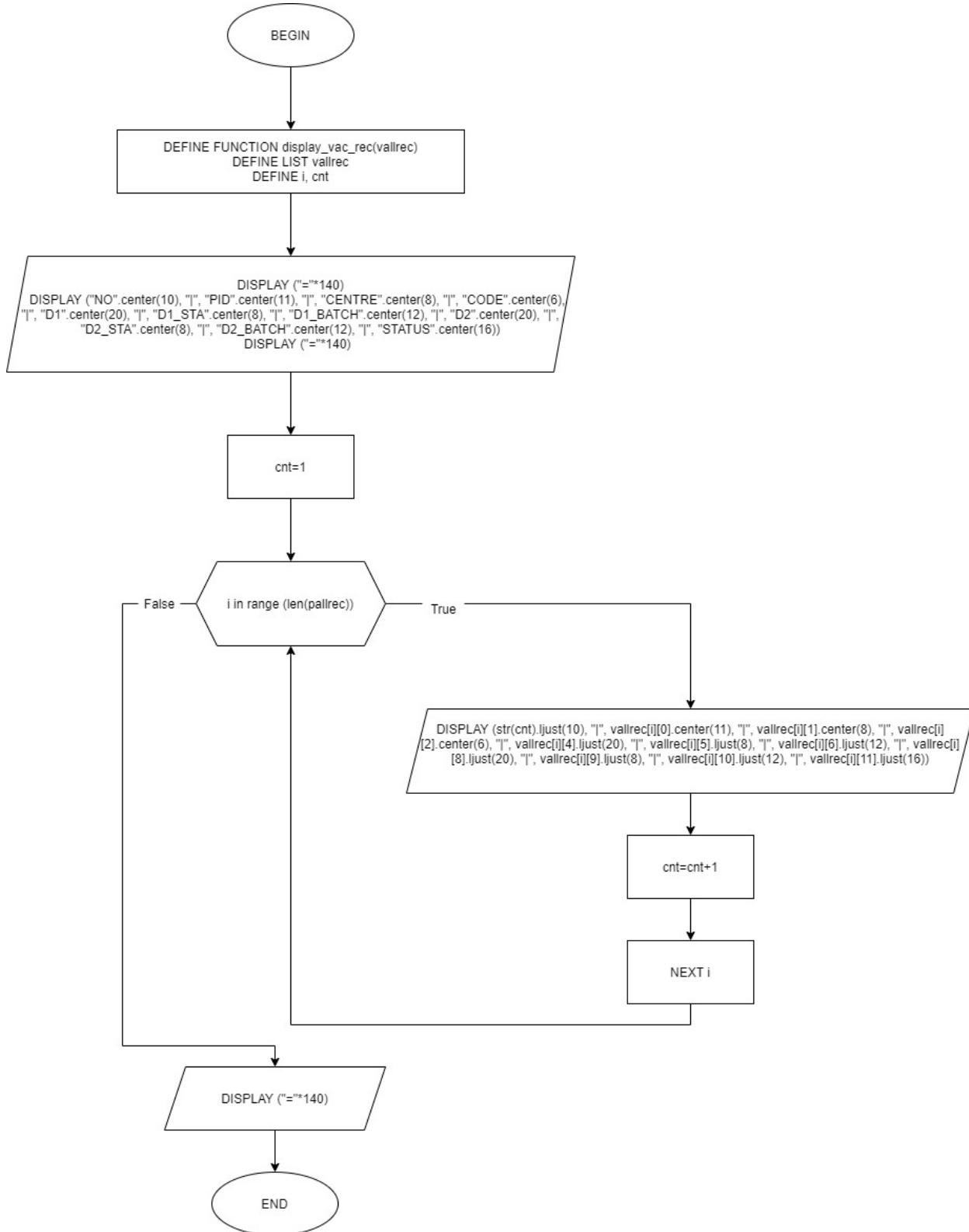
modify_login(plogallrec)

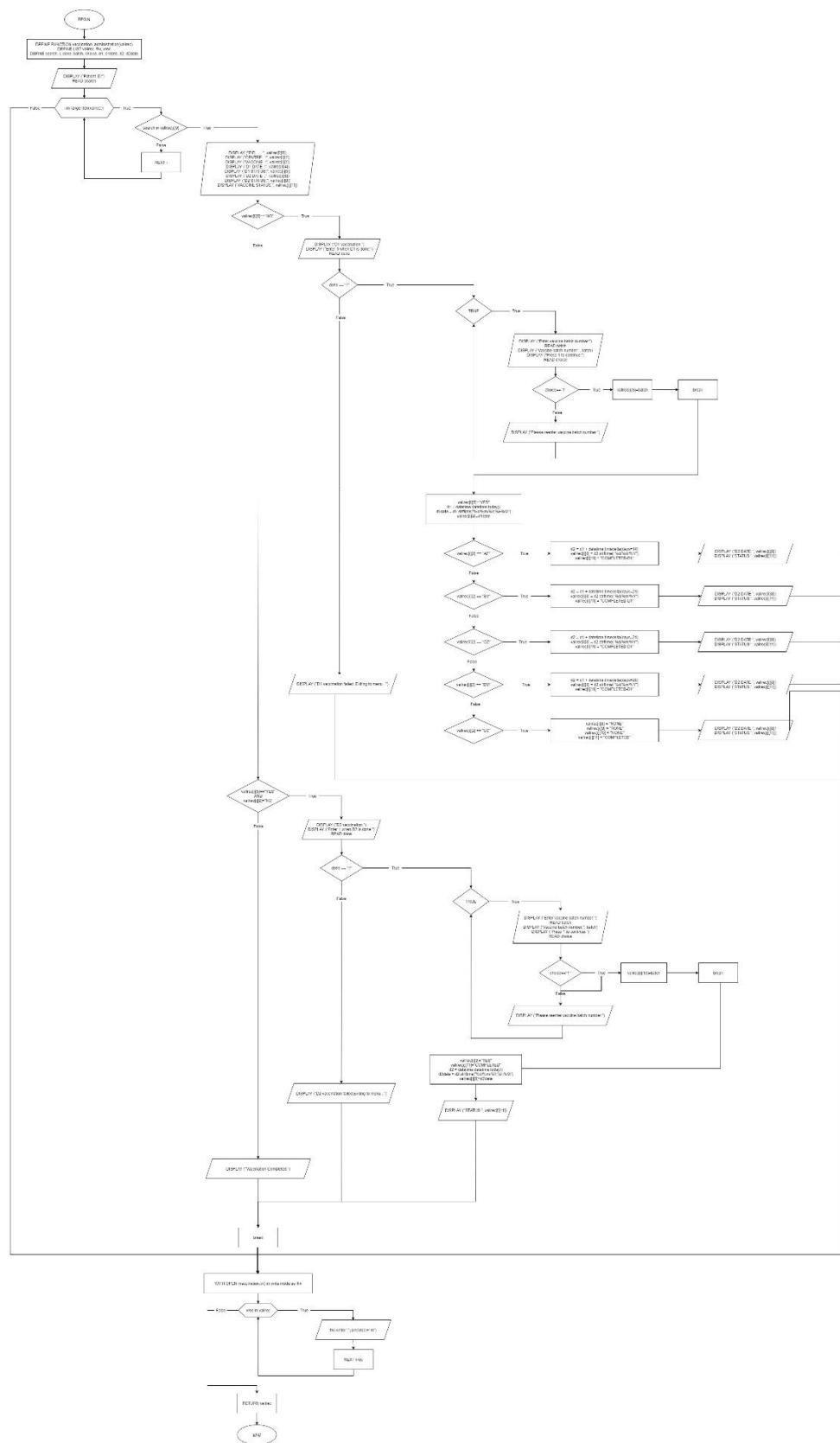
patient_check_status(vallrec)

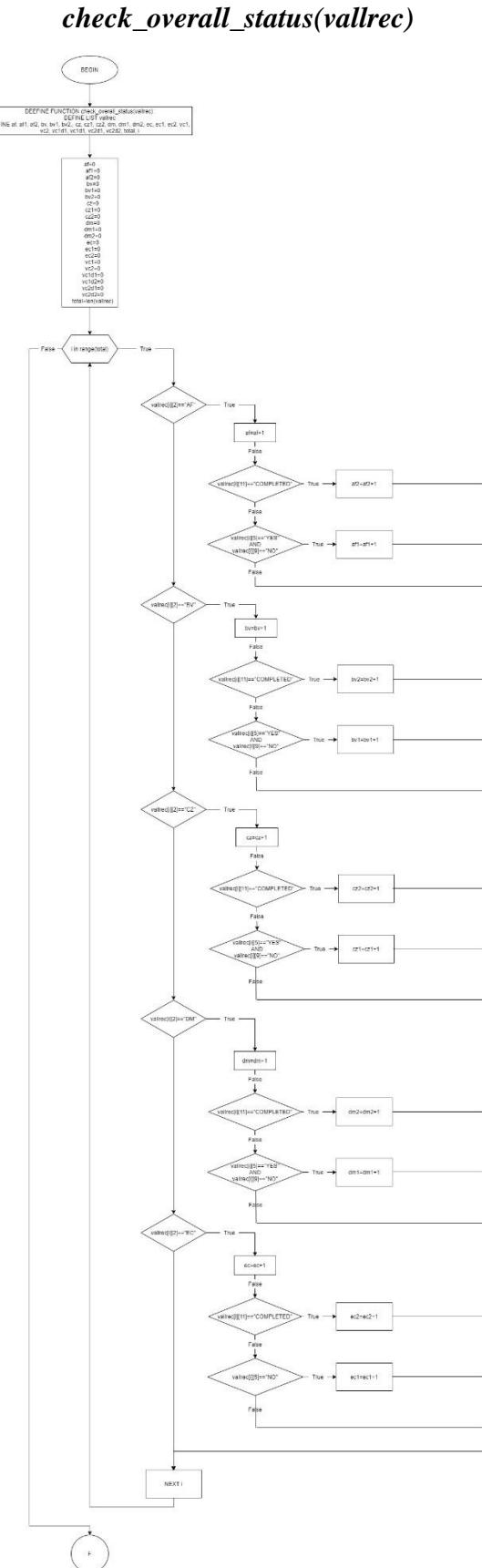
check_patient_status(vallrec)

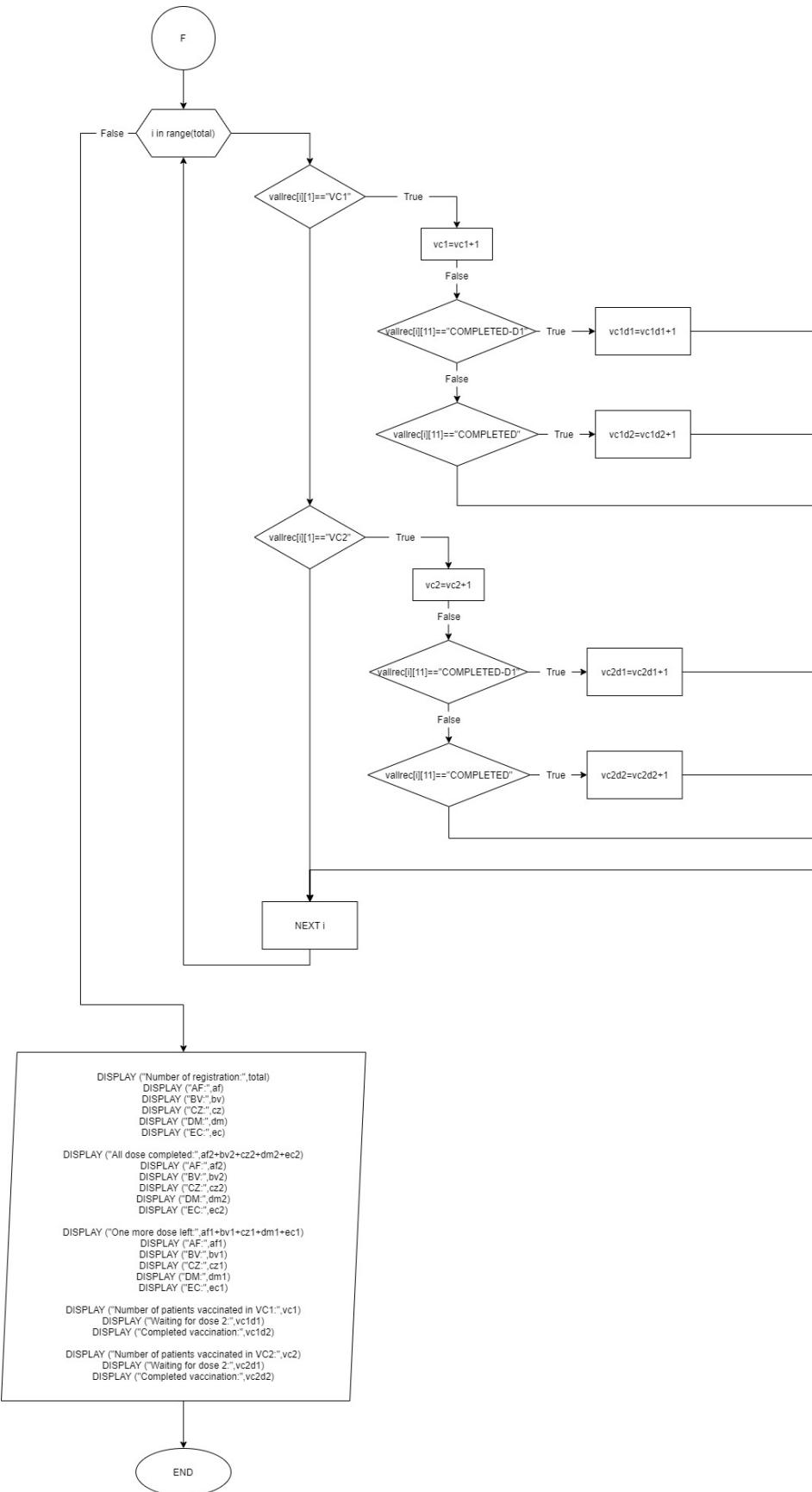
search_patient_id(pallrec)

display_patient_rec(pallrec)

display_vac_rec(vallrec)

vaccination_administration(vallrec)





4 Program Source Code and Explanation

Variables and User Input

Just as the mathematics term, a variable is a container containing value which can be changed anytime. Variable has many types, but in this program, only a few will be covered, which is int, sting, and float. If a variable contains multiple values, the variable is a list.

If user want to display something, user should use print() statement, the code can be written as the example below.

```
print("Available vaccine code for your age:")
print("1-AF")
print("2-BV")
print("3-DM")
print("4-EC")
```

To receive input from user after displaying something and store the value in a variable, the code can be written as below.

```
username = input ("Enter username:")
password = input ("Enter password:")
```

When the type of input is not clarified, the default input type will be string. In this case, the variable name is “username” and “password”, their value is string. Whatever user typed as an input, the value will be string and the value will always be in the quotation marks (“value”). However, in the screenshot above, not only the value of input is string, the “Enter username:” itself is a string too, but it is not a variable.

```
vc_code = input("Enter your choice:")
if vc_code == "1":
    vc_code="AF"
    flag=1
    break
```

To be simple, whatever in a pair of quotation mark (“ ”) is a string. String is like a picture. Whatever user typed for the input, the value of the variable will be exactly the picture of the input. As a “picture”, string cannot do calculation, but it can be split by some operation. In this case, the variable name is “vc_code”. From the example above, we can clearly understand

the definition of variable. At first, the value of vc_code is “1” but it changed to “AF” next. The value of a variable can be changed anytime.

Another important issue to mention is, the “1” in “vc_code==”1”” and “flag=1” is different. “1” is a string because it was quoted, 1 is an integer because it is not quoted.

```
if age>=12 and age<18:
```

If the input is defined as int(), the variable must be integer, otherwise an error will occurred. Different to string, int can do computation and operation which won’t involve decimals. However, if a variable involved in mathematical operation which contains decimals, the variable must be clarified as a float. For example, if a variable x=1, then float(x)=1.0.

Basically, it is easy to tell what are the type of variable if we can see the value. If the value is quoted, it is string; if the value is numeric and have decimals, it is float; otherwise, it is int.

```
age = int(input("Enter your age:"))
```

A string which the value between the quotations satisfy the definition of int, can be truned into int by using int(variable_name). The code above is actually a combination of two codes, which is:

```
age=input("Enter your age.")
```

```
age=int(age)
```

Similarly, if the condition is satisfied, a string can be turned into float by using float(variable_name). In fact, the initial value of the input is always a string, but it can be turned into int or float. Similarly, variables can always change the type easily by using str(), int() and float() operator.

```
type(int(contact[4:]))==int
```

However, we can know the type of the variable by using type(variable_name). The result will be int, string or float.

Numeric Operation

In python, there are some numeric operation symbol can be used to do mathematics or conditions validation. The justification is given as the table below.

Numeric Operation	Explanation	Examples
+	Normal addition	<code>cnt=cnt+1</code>
-	Normal Subtraction	<code>>>> 2-3 -1</code>
*	Normal Multiplication	<code>>>> 2*3 6</code>
**	Power (also known as ^ generally)	<code>>>> 3**2 9</code>
/	Full Division (with decimals)	<code>>>> 3/2 1.5</code>
//	Quotient/ Gauss Example: $5/2=2$ remainder 1 To get the result of 2, use $5//2=2$	<code>>>> 5//2 2 >>> 9//4 2 >>> 9//2 4</code>
%	Modular/ Remainder Example: $5/2=2$ remainder 1 To get the result of 1, use $5\%2=1$	<code>>>> 5%2 1 >>> 9%4 1 >>> 7%4 3</code>
=	Normal Equal Operator	<code>af=0 af1=0</code>
==	Absolutely Equal Not only equal in mathematics, but also equal in type. Usually used as a condition of if and while operation. Example: $1=1.0$ is true $1==1.0$ is false because 1 is int and 1.0 is float	<code>while flag==1: if flag==2 if eid == "PAF"</code>
!=	Not Equal	<code>cnt!=0 returns!=([], [])</code>

String Operation

As the explanation in “Variables and User Input” part, a string can be split into several parts by many ways. Before explaining the methods to split a string, len() function should be discussed first.

len()

len(contact)==11

The len() function is to calculate the length of a string or the list. If it is used on a string, len(string_name) will be the number of characters; if it is used on a list, len(list_name) will be the number of elements. In this case, contact is just a string. If a contact number is 012-3456789, the number of character in the contact is 11, so len(contact)=11.

[index_value]

An index value indicates the place of a character in a string. The index number for the first character is 0, so the index number of the last character is len(string)-1. Hence, to indicate the first character in the string, we use string_name[0].

nextid[3:]

To indicate multiple characters in a range, the method is [a:b]. Leaving blank for a and b indicates “from the beginning” and “till the end” respectively. Thus, for the example above, [3:] means the forth character till the end. If nextid= “PEC000002”, then nextid[3:] will be “000002”. This is the first way to slice a string.

string_name.split()

The split() function is to split the string based on something. If inside the bracket is empty, it means split the string by space.

```
sep_idlist = idlist.split(":")
```

In this case, it means split the string by ":".

```
PAF000005:PBV000002:PCZ000002:PDM000000:PEC000002
```

The code above is to split the id record in id.txt. If idlist=" PAF000005:PBV000002". After splitting idlist with ":" as a knife, idlist becomes "PAF000005", "PBV000002". When a variable contains multiple value, the variable will be list. The function of list will be discussed later. However, now idlist.split(":")=[“PAF000005”, “PBV000002”]. The code above saved the result of splitting as a new list which named as sep_idlist. This is the second method to split a string.

for loop

```
for j in email:  
    if j=="@":  
        flag=1
```

For loop split strings like the example given. If email="email", the first loop of j will be "e", the second loop of j will be "m", and so on. The code above is to test the appearance of "@" in the value of email. This is the third method to split a string.

join()

The string can be split, of course it can be joined too.

```
idlist=":".join(sep_idlist)
```

The code above in English, means join sep_idlist by using ":" and save it as idlist.

If/ Else Conditional

If conditional is used when an action is done only if it meets some certain condition. If there are many kinds of action regarding to many kinds of conditions, then the first condition will be “if” and the last condition can be but not must be “else”, the middle ones will be “elif”.

```
if username in plogallrec[i][0]:  
    print("\nSIGN UP FAILED! Current username is used.\n")  
    flag=1
```

If statement can be built too eventhough there is only one condition.

```
-----  
print("1- Register:")  
print("2- Modify Patient's Registration:")  
print("3- Display All Patients Personal Information:")  
print("4- Display All Patients Vaccination Status:")  
print("5- Vaccination Administration:")  
print("6- Check Patient Vaccination Status:")  
print("7- Check Overall Vaccination Statistics:")  
print("8- Check Patient ID:")  
print("9- Exit.")  
choice = input("Enter your choice to proceed to next step:")  
print()  
if choice == "1":  
    returns=add_patient()  
    if returns!=([], []):  
        pallrec.append(returns[0])  
        vallrec.append(returns[1])  
elif choice == "2":  
    pallrec = modify_patient(pallrec)  
elif choice == "3":  
    display_patient_rec(pallrec)  
elif choice == "4":  
    display_vac_rec(vallrec)  
elif choice == "5":  
    vaccination_administration(vallrec)  
elif choice == "6":  
    check_patient_status(vallrec)  
elif choice == "7":  
    check_overall_status(vallrec)  
elif choice == "8":  
    search_patient_id(pallrec)  
elif choice == "9":  
    break  
else:  
    print("\n\n INVALID OPTION!")
```

This is the staff main menu of the program. Users are suppose to input 1-9 to proceed with corresponding function. In case for some users who are naughty or accidentally keyed in an invalid value, else is used to indicate the input is wrong.

```

today = datetime.datetime.today()
d1 = today + datetime.timedelta(days=3)
d1date = d1.strftime("%d/%m/%Y")
if vc_code == "AF":
    d2 = d1 + datetime.timedelta(days=14)
    d2date = d2.strftime("%d/%m/%Y")
elif vc_code == "BV":
    d2 = d1 + datetime.timedelta(days=21)
    d2date = d2.strftime("%d/%m/%Y")
elif vc_code == "CZ":
    d2 = d1 + datetime.timedelta(days=21)
    d2date = d2.strftime("%d/%m/%Y")
elif vc_code == "DM":
    d2 = d1 + datetime.timedelta(days=28)
    d2date = d2.strftime("%d/%m/%Y")
elif vc_code == "EC":
    d2date = "NONE"

```

The time interval between first and second dose of every vaccine is different, so if condition is used to assign date for every different vaccine.

```

if username in plogallrec[i][0] and password in plogallrec[i][1]:
    flg = 1

```

If condition can be very creative, the conditions can not only be a value, it can also be an appearance or boolean.

```

if flag==0:
    print("Your passport is:"+icpp)
    choice= input("Press 1 to continue:")
    if choice=="1":
        print()
        bigflag=1
    else:
        print("Please reenter your passport.")
        print()
        flag=1

```

If there is a if statement under an if statement, it is called as nested decisions.

```

if icpp[6]=="-" and icpp[9]=="-" and len(icpp)==14 and type(int(icpp[:6]))==int and type(int(icpp[7:9]))==int and type(int(icpp[10:]))==int:

```

To prevent too many layers of nested decision, “and” and “or” can be used to decrease the line numbers and increase the speed of the program. However, the priority of “or” is higher than “and”.

Try/ Except Conditional

Try and except conditional is used when an error might occur in execution mode. For example, list index out of list, invalid literal for int() and so on. Basically, error “invalid literal for int()” is a very common error while running code. This will normally happen when a input variable is defined as int but the the input could not be tranformed into int.

Try and except must be used as a pair. A code which may have error occurred while executing should be placed under try. The program will run the code under try first, if any error occurred, then run the except code, so that the program won’t be forced killed by error.

```
contact = input("Enter your mobile number:(ex. 012-23456789)")
try:
    if contact[:2]=="01" and contact[3]=="-" and type(int(contact[4:]))==int and len(contact)==11 or len(contact)==12:
        print("Your contact is:"+contact)
        choice= input("Press 1 to continue:")
        if choice=="1":
            pallrec[i][6]=contact
            print()
            break
        else:
            print("Please reenter your mobile number.")
            print()
    else:
        print("INVALID INPUT!")
        print()
except:
    print("INVALID INPUT!")
    print()
```

The errors which might happen here is list index out of list for couldn’t find a pallrec[i][6] or invalid literal for int() for couldn’t change contact[4:] to int. Hence, this dangerous code should be placed under try. Once error occur, run except and print “Invalid Input”. Without try and except coding, when the error occurs, the code will be forced stop. Thus, a good program should include try and except to ensure the program is recyclable.

```
icpp = input("Enter your IC number: (ex: 123456-78-9999)")
try:
    if icpp[6]=="-" and icpp[9]=="-" and len(icpp)==14 and type(int(icpp[6]))==int and type(int(icpp[7:9]))==int and type(int(icpp[10:]))==int:
        flag=0
        for i in range (len(pallrec)):
            if icpp in pallrec[i][5]:
                print("This IC already registered!")
                print()
                flag=1
                break
        if flag==0:
            print("Your IC is:"+icpp)
            choice= input("Press 1 to continue:")
            if choice=="1":
                print()
                bigflag=1
            else:
                print("Please reenter your IC.")
                print()
                flag=1
        else:
            print("INVALID INPUT!")
            print()
    except:
        print("INVALID INPUT!")
        print()
```

This is another similar example for try and except conditional.

While Loop

Similar to if and else conditional, while conditional is used when an action is done repeatedly if it meets some certain condition. In other words, as long as it satisfies the condition, the action will be done repeatedly until it does not satisfy anymore. By using break, it can exit the loop before the loop ends to increase the efficiency of the program.

Different to while(condition), while True is a special while loop. It means repeat forever without any condition. The only way to get out from a while True loop is break, because it does not have a condition to quit same as does not have a condition to start loop.

```
while True:  
    print("Available vaccine code for your age:")  
    print("1-AF")  
    print("2-BV")  
    print("3-CZ")  
    print("4-DM")  
    print("5-EC")  
    vc_code = input("Enter your choice:")  
    if vc_code == "1":  
        vc_code="AF"  
        flag=1  
        break  
    elif vc_code == "2":  
        vc_code="BV"  
        flag=1  
        break  
    elif vc_code == "3":  
        vc_code="CZ"  
        flag=1  
        break  
    elif vc_code == "4":  
        vc_code="DM"  
        flag=1  
        break  
    elif vc_code == "5":  
        vc_code="EC"  
        flag=1  
        break  
    else:  
        print("INVALID INPUT!")  
        print()
```

This is an example of while True, if users typed the valid input, it will leads to break and exit the loop. Otherwise, invalid input leads to else condition which has no break inside. After running the code in the else, it will automatically prompt the available vaccine code to choose again, until user finally insert a valid input.

For Loop

For loop is also a looping function. The difference between while and for is, while loop is more like doing a loop under a certain condition, but for loop is more like doing a loop for a range of something.

```
for j in email:  
    if j=="@":  
        flag=1
```

As the example above mentioned in the string operation, looping characters in a string is one of the function of for loop.

Basically, the format of for loop is “for elements in a_range”. The range can be numeric or string or lists.

```
for i in range (len(pallrec)):
```

As the explanation of len() in string operation mentioned, the result of len() is definitely an integer. In this case, pallrec is a list of lists, so len(pallrec) will be the number of lists in it. Suppose len(pallrec)=5, then “for i in range(len(pallrec))” will be “for i in range(5)”. For numeric range, the result will be 0 up to but not including 5.

Numeric range can be tricky. To have a result from 0 to 4, it can be format such as “for i in range(5)”, “for i in range(0, 5)”, or “for i in range(0,5,1)”. All these three method will lead to the result of 0-4. For i in range(0, 5) also means from 0 up to but not including 5. Meanwhile, for i in range(0,5,1) means from 0 up to but not including 5 with a interval of 1.

In conclusion, for i in range(a) means from 0 up to but not including a; for i in range(a,b) means that from a up to b but not including b; for i in range(a,b,x) means from a up to b but not including b, with an interval of x. To be more crystal clear, here’s another example of for i in range(a,b,x). Assume for i in range(5,15,2), the result will be 5, 7, 9, 11, 13 only. 15 is not included because it is “up to but not including 15”.

```
for prec in fhp:  
    pinfo = prec.strip().split(":")
```

However, after discussing the string range and numeric range, the next will be the list range. If the range is a list, then the result will be the elements of list. Also, if the range is a lists in list, then the results will be the lists. In this case, fhp is a lists of list, hence prec are the lists in fhp. At the second line, every prec is split by ":" and stored as pinfo.

```
cnt=1
for i in range (len(vallrec)):
    print(str(cnt).ljust(10)+"|"+vallrec[i][0].center(11)+"|"+vallrec[i][1].center(8)+"|"+vallrec[i][2].center(6)+"|"+vallrec[i][4].ljust(20))
    cnt=cnt+1
```

This is to show elements by looping row by row of the lists in list.

```
for i in range (len(vallrec)):
    if search in vallrec[i][0]:
        print("PID :"+vallrec[i][0])
        print("CENTRE :"+vallrec[i][1])
        print("VACCINE :"+vallrec[i][2])
        print("D1 DATE :"+vallrec[i][4])
        print("D1 STATUS:"+vallrec[i][5])
        print("D2 DATE :"+vallrec[i][8])
        print("D2 STATUS:"+vallrec[i][9])
        print("VACCINE STATUS:"+vallrec[i][11])
        print()
        break
```

In the first sentence, the program gets the number od rows. The second sentence means search data in column 0 of every row. If the data is found, display the data's in the same row. If the data is not found in this row, proceed to the next row which is indicated by i.

List

If a variable contains multiple values, the variable is a list. Similar to the string operation. The index number indicates the place of an element in the list. By using append, program can add new elements into the list.

```
new_vrec.append(pid)
new_vrec.append(vc_centre)
new_vrec.append(vc_code)
new_vrec.append("D1")
new_vrec.append(d1date)
new_vrec.append("NO")
new_vrec.append("N/A")
new_vrec.append("D2")
new_vrec.append(d2date)
new_vrec.append("NO")
new_vrec.append("N/A")
new_vrec.append("NEW")
new_vrec.append(icpp)
```

The variables in the bracket are the elements which need to be append into the list. In this case, the name of list is new_vrec. The sequence of appending elements will decide the index number of every elements in the list.

```
new_prec.append(name)
new_prec.append(str(age))
new_prec.append(pid)
new_prec.append(vc_centre)
new_prec.append(vc_code)
new_prec.append(icpp)
new_prec.append(contact)
new_prec.append(email)
```

The special thing is, only string can be append into a list. Vice versa, the elements in the list are all string. For the second example, after appending,

new_prec=[“vivian”, “21”, “tp”, “vc1”, “af”, “icpp”, “0123456789”, 123@abc.com] now. The index number of the first element will be 0 and the last element will be len(new_prec)-1=7. It is same as the string operation.

For lists in list, to mention the location of an element, we need two index number to indicate row number and column number, which is list[i][j].

```
|VIVIAN WOON:22:PAF000001:VC1:AF:001226-10-1854:013-3986733:viviwoon@gamil.com  
VIVIAN TAN:23:PAF000003:VC1:AF:001226-10-0000:013-3876733:vivianwoon@asd.com  
woon:24:PAF000004:VC1:AF:001226-10-2222:011-1111111:woon@123.com  
YAN QI:33:PCZ000001:VC2:CZ:001226-10-3333:013-3986733:vvvv@woon.com  
TESTNAME:55:PEC000001:VC1:EC:001226-10-5555:014-4444444:TEST@TEST.COM  
TESTNAME:55:PEC000002:VC2:EC:001226-10-4444:014-4444444:TEST@TEST.COM  
TESTNAME:55:PAF000005:VC2:AF:001226-10-6666:015-5555555:TEST@TEST3.COM  
TESTTEST:33:PCZ000002:VC1:CZ:222222-22-2222:013-3333333:TESTTEST@TEST.COM  
TESTFINAL:44:PBV000002:VC2:BV:001122-10-4444:014-4444444:TESTFINAL@TEST.COM
```

```
with open("patient.txt", "r") as fhp:  
    for prec in fhp:  
        pinfo = prec.strip().split(":")  
        pallrec.append(pinfo)
```

For example, after a record in file went through the code above, every single elements can be found in pallrec. Then, pallrec[1][0] will be “VIVIAN TAN” and pallrec[8][1] will be “44”.

Function

```
def staff_main_menu():
```

When a program requires a reusable code, function need to be used. A function takes arguments as input and do something with the arguments, then return the result to where the function was called. To define a function, the reserved word is def.

For the example given above, the function name is staff_main_menu(). This is quite familiar because int() and float() has the same structure. In fact, int() and float() is really a built-in functions, so as type(), input() and so on.

To understand what is a function, the terms of arguments and parameters should be discussed first. While calling a function, input which is put inside the bracket is arguments. In the function, the variable which is used to allow the code to access the arguments is a parameters.

```
pid = get_new_id(pid_code)
```

For example, now we call the function get_new_id function with an arguments called pid_code.

```
def get_new_id(eid):
    with open("id.txt", "r") as fh:
        idlist=fh.readlines()
        if eid == "PAF":
            ind=0
        elif eid == "PBV":
            ind=1
        elif eid == "PCZ":
            ind=2
        elif eid == "PDM":
            ind=3
        elif eid == "PEC":
            ind=4
        else:
            ind=5
    sep_idlist = idlist.split(":")
    nextid = sep_idlist[ind]
    newid = str(int(nextid[3:])+1)
    if len(newid) == 1:
        nextid = nextid[:3] + "0000" + newid
    elif len(newid) == 2:
        nextid = nextid[:3] + "000" + newid
    elif len(newid) == 3:
        nextid = nextid[:3] + "00" + newid
    elif len(newid) == 4:
        nextid = nextid[:3] + "0" + newid
    elif len(newid) == 5:
        nextid = nextid[:3] + "0" + newid
    elif len(newid) == 6:
        nextid = nextid[:3] + newid
    sep_idlist[ind]=nextid
    idlist=":".join(sep_idlist)
    with open("id.txt", "w") as fh:
        fh.write(idlist)
    return nextid
```

Where we define the function, the variable in the bracket is called parameters. Hence, the parameters is eid.

Function:

```
return new_prec, new_vrec
```

Main logic:

```
returns=add_patient()
if returns!=([], []):
    pallrec.append(returns[0])
    vallrec.append(returns[1])
```

Return means that accepting arguments and do computation in the function and return a value. To be simple, it is the result of running the function and this result will be sent back to where the function is called. The parameters and arguments can be multiple but the number must be same. The return value also can be more than one and the result will be a list which contains every return value. The example above is the return value of add_patient(). Both of the return values are stored in the list (which is lists in list in this case). To append the right result accordingly, the list operation is used again.

Files

A file can be opened in python program in two ways. The first way is `fh=open("filename.txt", "r")`, the second way is with `open ("filename.txt", "r")` as `fh`. The first way requires to close the file by `fh.close()` at the end of the function, but the second way doesn't need.

A file can be opened in three modes, which are reading, writing, and append mode. The mode code is “r”, “w”, and “a” respectively. Reading mode only allows user to read but not editing; writing mode allows user to modify the current records; append mode allows user to add new records.

Basically, there are unspoken format for all three modes.

```
| plogallrec=[]
| with open("patientlogin.txt", "r") as fhplog:
|     for plogrec in fhplog:
|         ploginfo = plogrec.split(":")
|         plogallrec.append(ploginfo)
```

For reading mode, the format is as the code below. First of all, define a new list as list A. Open the file in reading mode as file handle. For every line in the file handle, split the line with something as a knife and store the splited line in list B. Then, append list B into list A. Repeat the same procedure for every line, until the line ends. This is all about copying the content of a file into a list which is list A. This is not straight away reading from the file.

```
pallrec[i][7]=email
with open("patient.txt", "w") as fhp:
    for prec in pallrec:
        fhp.write(":".join(prec))
```

For writing mode, open the file in reading mode first to do modification on the list A, which is the copy of the file. After doing modification by changing the value in specific location

in list A by the code below, then finally open the file in writing mode as file handle. For every line in list A, rewrite the file handle by joining the list B with a glue (in this case the glue is “:”).

```
with open("patient.txt", "a") as fh:  
    new_prec.append(name)  
    new_prec.append(str(age))  
    new_prec.append(pid)  
    new_prec.append(vc_centre)  
    new_prec.append(vc_code)  
    new_prec.append(icpp)  
    new_prec.append(contact)  
    new_prec.append(email)  
    fh.write(":".join(new_prec)+"\n")
```

For appending mode, no need to open the file in reading mode in advanced. First of all, define a new list. After collecting the datas which need to append into a new line in the file, open the file in append mode as file handle. Then, append all the datas into the new list. Finally, overwrite the file handle by joining the new list datas with glue (in this case the glue is “:”), and insert a new line in the end.

```
for i in range(len(pallrec)):  
    if searchname in pallrec[i][0] and searchicpp in pallrec[i][5]:  
        print("Please confirm this is you before make any changes:")  
        print("Name      :", pallrec[i][0])  
        print("Patient ID:", pallrec[i][2])  
        print("1-Contact -", pallrec[i][6])  
        print("2-Email   -", pallrec[i][7])
```

To serch data, use the concept of opening the file in read mode. Since the contents are copied into a lists in list, then we can find data using lists operation, as the code below.

Run the first i in $\text{range}(\text{len}(\text{pallrec}))$, it is 0, so for the first loop, $i=0$. If wanted data is founded in column 0 and 5 of row 0, then print the following data by using $i=0$. Else, run the next i in the range, which is $i=1$, and repeat the same procedure for every i in $\text{range}(\text{len}(\text{pallrec}))$.

5 Screenshots of Sample Input/Output and Explanation

Login Menu

```
1-Login as patient  
2-Sign up as patient  
3-Staff  
4-Exit  
Please select your identity:|
```

Once the program is started, user will need to do login in this interface, which is named as login menu.

```
1-Login as patient  
2-Sign up as patient  
3-Staff  
4-Exit  
Please select your identity:2
```

```
Please enter username:testing999  
Please enter password:999
```

```
1-Login as patient  
2-Sign up as patient  
3-Staff  
4-Exit  
Please select your identity:1
```

Before login, first sign up as a patient user. After the registration successful, it will automatically return to the login menu. Now, login as a patient by using the username and password which is just registered.

```
1-Login as patient  
2-Sign up as patient  
3-Staff  
4-Exit  
Please select your identity:1
```

```
Enter username:testing999  
Enter password:999
```

```
1- Register:  
2- Modify Registration:  
3- Modify Login:  
4- Check Personal Vaccination Status:  
5- Check Overall Vaccination Statistics:  
6- Check Patient ID:  
7- Exit.  
Enter your choice to proceed to next step:|
```

If the menu changed, means the login is successful.

1- Register:
2- Modify Registration:
3- Modify Login:
4- Check Personal Vaccination Status:
5- Check Overall Vaccination Statistics:
6- Check Patient ID:
7- Exit.

Enter your choice to proceed to next step:7

1-Login as patient
2-Sign up as patient
3-Staff
4-Exit
Please select your identity:2

Please enter username:testing999
Please enter password:999

SIGN UP FAILD! Current username is used.

1-Login as patient
2-Sign up as patient
3-Staff
4-Exit
Please select your identity:|

Choosing 7 allows to return to the login menu. Sign up as patient using the existing username and password, it will show sign up failed and return back to the menu.

Patient Menu / patient_main_menu()

- 1- Register:
- 2- Modify Registration:
- 3- Modify Login:
- 4- Check Personal Vaccination Status:
- 5- Check Overall Vaccination Statistics:
- 6- Check Patient ID:
- 7- Exit.

Enter your choice to proceed to next step:8

INVALID OPTION!

- 1- Register:
- 2- Modify Registration:
- 3- Modify Login:
- 4- Check Personal Vaccination Status:
- 5- Check Overall Vaccination Statistics:
- 6- Check Patient ID:
- 7- Exit.

Enter your choice to proceed to next step:|

Invalid input will not stop the program.

Registration / add_patient()

```
1- Register:  
2- Modify Registration:  
3- Modify Login:  
4- Check Personal Vaccination Status:  
5- Check Overall Vaccination Statistics:  
6- Check Patient ID:  
7- Exit.
```

Enter your choice to proceed to next step:1

Enter your name as per IC or passport:KHOR YOU QI

Your name is:KHOR YOU QI

Press 1 to continue:1

Are you Malaysian? YES/NO:YES

Enter your IC number: (ex: 123456-78-9999)001226-10-1919

Your IC is:001226-10-1919

Press 1 to continue:1

Vaccine centre:

1-VC1

2-VC2

Enter your choice:1

Your vaccine centre is:VC1

Press 1 to continue:1

Enter your age:21

Available vaccine code for your age:

1-AF

2-BV

3-CZ

4-DM

5-EC

Enter your choice:1

Your age is:21

Your vaccine code is:AF

Press 1 to continue:1

Enter your mobile number: (ex. 012-23456789)013-3986733

Your contact is:013-3986733

Press 1 to continue:1

Enter your e-mail address:khor@gmail.com

Your email is:khor@gmail.com

Press 1 to continue:1

Name	:KHOR YOU QI
Age	:21
Vaccine Centre	:VC1
Vaccine Code	:AF
Contact	:013-3986733
Email	:khor@gmail.com

Make sure your information are correct.

Once registration done, only contact and email can be changed.

Press 1 to continue. Press 2 to exit registration. Otherwise, restart the registration:1

REGISTRATION SUCCESSFUL!

Patient ID:PAF000006

Dose 1 :09/09/2021

Dose 2 :23/09/2021

REMINDER: Patient ID is important when having vaccination.

1- Register:

2- Modify Registration:

3- Modify Login:

4- Check Personal Vaccination Status:

5- Check Overall Vaccination Statistics:

6- Check Patient ID:

7- Exit.

Enter your choice to proceed to next step:1

This is the full flow of a successful patient registration. If the registration is successful, the patient id and dose dates will prompt immediately. The date of dose 1 will be three days after the registration.

1- Register:

2- Modify Registration:

3- Modify Login:

4- Check Personal Vaccination Status:

5- Check Overall Vaccination Statistics:

6- Check Patient ID:

7- Exit.

Enter your choice to proceed to next step:1

Enter your name as per IC or passport:tan qwee choo

Your name is:TAN QWEE CHOO

The program will delete all the numbers and change all the alphabets to uppercase automatically.

```
Enter your age:77
Available vaccine code for your age:
1-AF
2-BV
3-DM
4-EC
Enter your choice:4
Your age is:77
Your vaccine code is:EC
Press 1 to continue:2
Please reselect your age and vaccine code.
```

Enter your age:

Users have many chances to do check and changes for their information because once the registration is successful, patients can only change their contact number and email.

```
Enter your e-mail address:tan@yahoowoelrf
INVALID INPUT!
```

```
Enter your e-mail address:woiyefg
INVALID INPUT!
```

```
Enter your e-mail address:@wqerf
INVALID INPUT!
```

```
Enter your e-mail address:@yahoo. com
INVALID INPUT!
```

```
Enter your e-mail address:tan@. com
INVALID INPUT!
```

```
Enter your e-mail address:tan@yahooocom
INVALID INPUT!
```

```
Enter your e-mail address:tan@yahoo.
INVALID INPUT!
```

```
Enter your e-mail address:tan@yahoo. com
Your email is:tan@yahoo. com
Press 1 to continue:1
```

The validation of email is designed strictly.

```
Enter your mobile number:(ex. 012-23456789)012-933-8900
INVALID INPUT!
```

```
Enter your mobile number:(ex. 012-23456789)012-9338900
Your contact is:012-9338900
Press 1 to continue:|
```

The validation for contact number is strict too.

Are you Malaysian? YES/NO:yes
Enter your IC number: (ex: 123456-78-9999) 001226-10-1854
This IC already registered!

Are you Malaysian? YES/NO:|

One patient only can register once by using IC or passport number.

Name :TAN QWEE CHOO
Age :77
Vaccine Centre:VC2
Vaccine Code :EC
Contact :011-56817234
Email :tan@yahoo.com

Make sure your information are correct.
Once registration done, only contact and email can be changed.
Press 1 to continue. Press 2 to exit registration. Otherwise, restart the registration:2
exiting to menu...
Registration Failed.

1- Register:
2- Modify Registration:
3- Modify Login:
4- Check Personal Vaccination Status:
5- Check Overall Vaccination Statistics:
6- Check Patient ID:
7- Exit.
Enter your choice to proceed to next step:|

Users have a lot of chances to verify their information, so that nobody who has registered can have excuse to change their personal information.

Modify Registration / modify_patient()

```
1- Register:  
2- Modify Registration:  
3- Modify Login:  
4- Check Personal Vaccination Status:  
5- Check Overall Vaccination Statistics:  
6- Check Patient ID:  
7- Exit.
```

Enter your choice to proceed to next step:2

```
Please enter name to search:KHOR YOU QI  
Please enter IC or PASSPORT NO to search:001226-10-1919  
Please confirm this is you before make any changes:  
Name : KHOR YOU QI  
Patient ID: PAF000006  
1-Contact - 013-3986733  
2-Email - khor@gmail.com  
Enter the no of field to modify:1  
Enter your mobile number:(ex. 012-23456789)012-3986733  
Your contact is:012-3986733  
Press 1 to continue:1
```

```
1- Register:  
2- Modify Registration:  
3- Modify Login:  
4- Check Personal Vaccination Status:  
5- Check Overall Vaccination Statistics:  
6- Check Patient ID:  
7- Exit.
```

Enter your choice to proceed to next step:2

```
Please enter name to search:KHOR YOU QI  
Please enter IC or PASSPORT NO to search:001226-10-1919  
Please confirm this is you before make any changes:  
Name : KHOR YOU QI  
Patient ID: PAF000006  
1-Contact - 012-3986733  
2-Email - khor@gmail.com  
Enter the no of field to modify:
```

Users use modify registration to modify personal contacts. The modification is successful.

```
1- Register:  
2- Modify Registration:  
3- Modify Login:  
4- Check Personal Vaccination Status:  
5- Check Overall Vaccination Statistics:  
6- Check Patient ID:  
7- Exit.  
Enter your choice to proceed to next step:2
```

```
Please enter name to search:aeiolwrfghu  
Please enter IC or PASSPORT NO to search:weif  
Record not found!
```

```
1- Register:  
2- Modify Registration:  
3- Modify Login:  
4- Check Personal Vaccination Status:  
5- Check Overall Vaccination Statistics:  
6- Check Patient ID:  
7- Exit.  
Enter your choice to proceed to next step:
```

If users inserted wrong record, it would return to the menu automatically and display that the record is not found.

Modify Login / modify_login()

```
1- Register:  
2- Modify Registration:  
3- Modify Login:  
4- Check Personal Vaccination Status:  
5- Check Overall Vaccination Statistics:  
6- Check Patient ID:  
7- Exit.  
Enter your choice to proceed to next step:3  
  
Please enter current username:testing999  
Please enter current password:999  
Modify field:  
1-username  
2-password  
Please enter your choice:1  
Please enter your new username:testing888  
Please confirm your username:testing888  
  
1- Register:  
2- Modify Registration:  
3- Modify Login:  
4- Check Personal Vaccination Status:  
5- Check Overall Vaccination Statistics:  
6- Check Patient ID:  
7- Exit.  
Enter your choice to proceed to next step:
```

If it returns to the menu automatically, the modification is successful. To check the validation, we exit to the login menu and login again.

```
1-Login as patient  
2-Sign up as patient  
3-Staff  
4-Exit  
Please select your identity:1  
  
Enter username:testing999  
Enter password:999  
  
LOGIN FAILED! Username or password wrong.  
  
1-Login as patient  
2-Sign up as patient  
3-Staff  
4-Exit  
Please select your identity:1  
  
Enter username:testing888  
Enter password:999  
  
1- Register:  
2- Modify Registration:  
3- Modify Login:  
4- Check Personal Vaccination Status:  
5- Check Overall Vaccination Statistics:  
6- Check Patient ID:  
7- Exit.  
Enter your choice to proceed to next step:
```

The modification is absolutely successful.

Check Personal Vaccination Status / patient_check_status()

Check Patient ID / search_patient_id()

- 1- Register:
- 2- Modify Registration:
- 3- Modify Login:
- 4- Check Personal Vaccination Status:
- 5- Check Overall Vaccination Statistics:
- 6- Check Patient ID:
- 7- Exit.

Enter your choice to proceed to next step:4

Registered IC/passport:001226-10-1919
Patient ID:euowirth

- 1- Register:
- 2- Modify Registration:
- 3- Modify Login:
- 4- Check Personal Vaccination Status:
- 5- Check Overall Vaccination Statistics:
- 6- Check Patient ID:
- 7- Exit.

Enter your choice to proceed to next step:|

To check personal vaccination status, patient need to key in the patient code to access.
Normally, nobody will remember the patient ID. Hence, “check patient ID” function is proposed.

- 1- Register:
- 2- Modify Registration:
- 3- Modify Login:
- 4- Check Personal Vaccination Status:
- 5- Check Overall Vaccination Statistics:
- 6- Check Patient ID:
- 7- Exit.

Enter your choice to proceed to next step:6

Please enter name to search:KHOR YOU QI
Please enter IC or PASSPORT NO to search:001226-10-1919
Name: KHOR YOU QI
IC/passport: 001226-10-1919
Patient ID: PAF000006

After having the patient ID, user can proceed to the check personal status again.

- 1- Register:
- 2- Modify Registration:
- 3- Modify Login:
- 4- Check Personal Vaccination Status:
- 5- Check Overall Vaccination Statistics:
- 6- Check Patient ID:
- 7- Exit.

Enter your choice to proceed to next step:4

Registered IC/passport:001226-10-1919

Patient ID:PAF000006

PID :PAF000006

CENTRE :VC1

VACCINE :AF

D1 DATE :09/09/2021

D1 STATUS:NO

D2 DATE :23/09/2021

D2 STATUS:NO

VACCINE STATUS:NEW

Users can check their status and dose dates here.

Check Overall Vaccination Statistics / check_overall_status()

- 1- Register:
 - 2- Modify Registration:
 - 3- Modify Login:
 - 4- Check Personal Vaccination Status:
 - 5- Check Overall Vaccination Statistics:
 - 6- Check Patient ID:
 - 7- Exit.
- Enter your choice to proceed to next step:5

Number of registration: 10

AF: 5
BV: 1
CZ: 2
DM: 0
EC: 2

All dose completed: 2
AF: 2
BV: 0
CZ: 0
DM: 0
EC: 0

One more dose left: 3
AF: 0
BV: 0
CZ: 1
DM: 0
EC: 2

Number of patients vaccinated in VC1: 6
Waiting for dose 2: 0
Completed vaccination: 2

Number of patients vaccinated in VC2: 4
Waiting for dose 2: 1
Completed vaccination: 0

Users can check the statistics of vaccination using this function.

Staff Menu / staff_main_menu()

```
1-Login as patient  
2-Sign up as patient  
3-Staff  
4-Exit  
Please select your identity:3
```

```
Enter username:SVC1000001
```

```
Enter password:VC1STAFF
```

```
1- Register:  
2- Modify Patient's Registration:  
3- Display All Patients Personal Information:  
4- Display All Patients Vaccination Status:  
5- Vaccination Administration:  
6- Check Patient Vaccination Status:  
7- Check Overall Vaccination Statistics:  
8- Check Patient ID:  
9- Exit.
```

```
Enter your choice to proceed to next step:
```

Staff's username and password are given manually as the assumptions. Every counter uses different staff account respectively. The register, modify patient's registration, check overall vaccination status, and check patient id function are same to the patient site.

Display All Patients Personal Information / display_patient_rec()

1- Register:
 2- Modify Patient's Registration:
 3- Display All Patients Personal Information:
 4- Display All Patients Vaccination Status:
 5- Vaccination Administration:
 6- Check Patient Vaccination Status:
 7- Check Overall Vaccination Statistics:
 8- Check Patient ID:
 9- Exit.
 Enter your choice to proceed to next step:3

NO	NAME	AGE	PID	CENTRE	CODE	IC/PP	CONTACT	EMAIL
1	VIVIAN WOON	22	PAF000001	VC1	AF	001226-10-1854	013-3986733	viviwoon@gamil.com
2	VIVIAN TAN	23	PAF000003	VC1	AF	001226-10-0000	013-3876733	vivianwoon@asd.com
3	WOON	24	PAF000004	VC1	AF	001226-10-2222	011-1111111	woon@123.com
4	YAN QI	33	PCZ000001	VC2	CZ	001226-10-3333	013-3986733	vvvv@woon.com
5	TESTNAME	55	PEC000001	VC1	EC	001226-10-5555	014-4444444	TEST@TEST.COM
6	TESTNAME	55	PEC000002	VC2	EC	001226-10-4444	014-4444444	TEST@TEST.COM
7	TESTNAME	55	PAF000005	VC2	AF	001226-10-6666	015-5555555	TEST@TEST3.COM
8	TESTTEST	33	PCZ000002	VC1	CZ	222222-22-2222	013-3333333	TESTTEST@TEST.COM
9	TESTFINAL	44	PBV000002	VC2	BV	001122-10-4444	014-4444444	TESTFINAL@TEST.COM
10	KHOR YOU QI	21	PAF000006	VC1	AF	001226-10-1919	012-3986733	khor@gmail.com

The latest record is record number 10. The contact modified just now is the latest changes too.

Display All Patients Vaccination Status / display_vac_rec()

1- Register:
 2- Modify Patient's Registration:
 3- Display All Patients Personal Information:
 4- Display All Patients Vaccination Status:
 5- Vaccination Administration:
 6- Check Patient Vaccination Status:
 7- Check Overall Vaccination Statistics:
 8- Check Patient ID:
 9- Exit.
 Enter your choice to proceed to next step:4

NO	PID	CENTRE	CODE	D1	D1_STA	D1_BATCH	D2	D2_STA	D2_BATCH	STATUS
1	PAF000001	VC1	AF	26/08/2021	YES	ABCD	26/08/2021, 0030	YES	ABCDE	COMPLETED
2	PAF000003	VC1	AF	29/08/2021	NO	N/A	12/09/2021	NO	N/A	NEW
3	PAF000004	VC1	AF	28/08/2021, 0321	YES	ABCDE	28/08/2021, 0324	YES	ABCDE	COMPLETED
4	PCZ000001	VC2	CZ	28/08/2021, 0327	YES	NEWNEW	18/09/2021	NO	N/A	COMPLETED-D1
5	PEC000001	VC1	EC	31/08/2021	NO	N/A	NONE	NO	N/A	NEW
6	PEC000002	VC2	EC	31/08/2021	NO	N/A	NONE	NO	N/A	NEW
7	PAF000005	VC2	AF	31/08/2021	NO	N/A	14/09/2021	NO	N/A	NEW
8	PCZ000002	VC1	CZ	31/08/2021	NO	N/A	21/09/2021	NO	N/A	NEW
9	PBV000002	VC2	BV	31/08/2021	NO	N/A	21/09/2021	NO	N/A	NEW
10	PAF000006	VC1	AF	09/09/2021	NO	N/A	23/09/2021	NO	N/A	NEW

For those who haven't receive any dose, the status will be "NEW"; for those who taken dose 1 only, excluding patients who chose EC, the status is "COMPLETED-D1"; for those who completed every dose, the status is "COMPLETED".

Vaccination Administration / vaccination_administration()

```

1- Register:
2- Modify Patient's Registration:
3- Display All Patients Personal Information:
4- Display All Patients Vaccination Status:
5- Vaccination Administration:
6- Check Patient Vaccination Status:
7- Check Overall Vaccination Statistics:
8- Check Patient ID:
9- Exit.

Enter your choice to proceed to next step:5

Patient ID:PAF000006
PID      :PAF000006
CENTRE   :VC1
VACCINE  :AF
D1 DATE   :09/09/2021
D1 STATUS:NO
D2 DATE   :23/09/2021
D2 STATUS:NO
VACCINE STATUS:NEW
D1 vaccination.
Enter 1 when D1 is done:1
Enter vaccine batch number:AF20210906-C
Vaccine batch number:AF20210906-C
Press 1 to continue:1
D2 DATE:20/09/2021
STATUS:COMPLETED-D1

```

The system will automatically show the dose number according to the patient's record. The initial dose 1 date is 9/9/2021 and the dose 2 date is 23/9/2021. After completing the dose 1, the date of dose 1 will change to todays date, and the date of dose 2 is recalculated according to the time interval for each dose. The dose 2 date and updated status are shown too.

```

1- Register:
2- Modify Patient's Registration:
3- Display All Patients Personal Information:
4- Display All Patients Vaccination Status:
5- Vaccination Administration:
6- Check Patient Vaccination Status:
7- Check Overall Vaccination Statistics:
8- Check Patient ID:
9- Exit.

Enter your choice to proceed to next step:4

```

NO	PID	CENTRE	CODE	D1	D1_STA	D1_BATCH	D2	D2_STA	D2_BATCH	STATUS
1	PAF000001	VC1	AF	26/08/2021	YES	ABCD	26/08/2021, 0030	YES	ABCDE	COMPLETED
2	PAF000003	VC1	AF	29/08/2021	NO	N/A	12/09/2021	NO	N/A	NEW
3	PAF000004	VC1	AF	28/08/2021, 0321	YES	ABCDE	28/08/2021, 0324	YES	ABCDE	COMPLETED
4	PCZ000001	VC2	CZ	28/08/2021, 0327	YES	NEWNEW	18/09/2021	NO	N/A	COMPLETED-D1
5	PEC000001	VC1	EC	31/08/2021	NO	N/A	NONE	NO	N/A	NEW
6	PEC000002	VC2	EC	31/08/2021	NO	N/A	NONE	NO	N/A	NEW
7	PAF000005	VC2	AF	31/08/2021	NO	N/A	14/09/2021	NO	N/A	NEW
8	PCZ000002	VC1	CZ	31/08/2021	NO	N/A	21/09/2021	NO	N/A	NEW
9	PBV000002	VC2	BV	31/08/2021	NO	N/A	21/09/2021	NO	N/A	NEW
10	PAF000006	VC1	AF	06/09/2021, 1921	YES	AF20210906-C	20/09/2021	NO	N/A	COMPLETED-D1

Check patient vaccination status again right after the vaccine administration, the record had changed.

```

1- Register:
2- Modify Patient's Registration:
3- Display All Patients Personal Information:
4- Display All Patients Vaccination Status:
5- Vaccination Administration:
6- Check Patient Vaccination Status:
7- Check Overall Vaccination Statistics:
8- Check Patient ID:
9- Exit.
Enter your choice to proceed to next step:5

```

```

Patient ID:PAF000006
PID      :PAF000006
CENTRE   :VC1
VACCINE  :AF
D1 DATE  :06/09/2021, 1921
D1 STATUS:YES
D2 DATE  :20/09/2021
D2 STATUS:NO
VACCINE STATUS:COMPLETED-D1
D2 vaccination.
Enter 1 when D2 is done:1
Enter vaccine batch number:AF20210906-D
Vaccine batch number:AF20210906-D
Press 1 to continue:1
STATUS:COMPLETED

```

Do dose 2 vaccination for the same patient again, the system knows that this patient is having dose 2. After the dose 2 is done, the status changed to “COMPLETED”.

```

1- Register:
2- Modify Patient's Registration:
3- Display All Patients Personal Information:
4- Display All Patients Vaccination Status:
5- Vaccination Administration:
6- Check Patient Vaccination Status:
7- Check Overall Vaccination Statistics:
8- Check Patient ID:
9- Exit.
Enter your choice to proceed to next step:4

```

NO	PID	CENTRE	CODE	D1	D1_STA	D1_BATCH	D2	D2_STA	D2_BATCH	STATUS
1	PAF000001	VC1	AF	26/08/2021	YES	ABCD	26/08/2021, 0030	YES	ABCDE	COMPLETED
2	PAF000003	VC1	AF	29/08/2021	NO	N/A	12/09/2021	NO	N/A	NEW
3	PAF000004	VC1	AF	28/08/2021, 0321	YES	ABCDE	28/08/2021, 0324	YES	ABCDE	COMPLETED
4	PCZ000001	VC2	CZ	28/08/2021, 0327	YES	NEWNEW	18/09/2021	NO	N/A	COMPLETED-D1
5	PEC000001	VC1	EC	31/08/2021	NO	N/A	NONE	NO	N/A	NEW
6	PEC000002	VC2	EC	31/08/2021	NO	N/A	NONE	NO	N/A	NEW
7	PAF000005	VC2	AF	31/08/2021	NO	N/A	14/09/2021	NO	N/A	NEW
8	PCZ000002	VC1	CZ	31/08/2021	NO	N/A	21/09/2021	NO	N/A	NEW
9	PBV000002	VC2	BV	31/08/2021	NO	N/A	21/09/2021	NO	N/A	NEW
10	PAF000006	VC1	AF	06/09/2021, 1921	YES	AF20210906-C	06/09/2021, 1930	YES	AF20210906-D	COMPLETED

Check record right after the vaccination again, the record is already changed.

```
1- Register:  
2- Modify Patient's Registration:  
3- Display All Patients Personal Information:  
4- Display All Patients Vaccination Status:  
5- Vaccination Administration:  
6- Check Patient Vaccination Status:  
7- Check Overall Vaccination Statistics:  
8- Check Patient ID:  
9- Exit.  
Enter your choice to proceed to next step:5
```

```
Patient ID:PAF000006  
PID      :PAF000006  
CENTRE   :VC1  
VACCINE  :AF  
D1 DATE  :06/09/2021, 1921  
D1 STATUS:YES  
D2 DATE  :06/09/2021, 1930  
D2 STATUS:YES  
VACCINE STATUS:COMPLETED  
Vaccination Completed.
```

For those who already done every dose, they cannot do vaccination again.

Check Patient Vaccination Status / check_patient_status()

- 1- Register:
 - 2- Modify Patient's Registration:
 - 3- Display All Patients Personal Information:
 - 4- Display All Patients Vaccination Status:
 - 5- Vaccination Administration:
 - 6- Check Patient Vaccination Status:
 - 7- Check Overall Vaccination Statistics:
 - 8- Check Patient ID:
 - 9- Exit.
- Enter your choice to proceed to next step:6

```
Patient ID:PAF000006
PID      :PAF000006
CENTRE   :VC1
VACCINE  :AF
D1 DATE  :06/09/2021, 1921
D1 STATUS:YES
D2 DATE  :06/09/2021, 1930
D2 STATUS:YES
VACCINE STATUS:COMPLETED
```

The check patient vaccination status function in staff site will only prompt for patient ID, but this function at patient site will prompt for IC or passport too because patient should see their status only.

6 Conclusion

I learnt a lot from this module. Teacher really did a great job on teaching the basis. I am not a complete newbie in programming. I've learnt C++ for one year and C# for half year, but I am confidence to say I really don't know what I learn form that 18 months. The class looks cool, the assignment questions are cool, the competition questions are cool. Teacher never teach us proper basis. On the contrary, teacher really emphasize the importance of basis here, and I finally know what I am learning and get the feel of success from programming. Although what we have learnt is just a very minor part in python, but using these simple command to solve big problem really makes me happy. I really appreciate the effort teacher put in class, although I seldom have questions to ask, but I really I gain motive from seeing teacher did your best to teach, I was moved. I am looking forward on teacher's intermediate classes.

By the way, I understand that this module is for basic only, but the restriction of not to use out of topic things makes me confuse sometimes. What are the exactly things which is out of topic? Many times I feel want to use something I saw which is very cool in this assignment, but the restrictions hold me back. Also, I hope teacher can teach additional things or at least guide us that which part of python we should learn according to our courses, because it seems like we won't have anymore python class in the future but python programming is quite important to our major.

Finally, I really want to say a great thank you to teacher. Without you, I might won't have the passion for programming anymore. Also, the delay for assignment is really appreciated. Thank you teacher!