**ONE PERFORMANCE INNOVATIVE TASK**

**“Beyond the Queue: Innovations to Reduce Medical Delays”**



In Partial Fulfillment of the Requirements in

Department of Information Technology

University of Science and Technology of Southern Philippines

Oroquieta City Campus

**BENZ CARORO**

**MATT NYL S. YTING**

**ALEXANDER M. DOMINGUEZ**

**KIMBERLY C. TAYTAYAN**

**BSIT 1D 3**

May 2025

**PROBLEM ANALYSIS REPORT**

**Introduction**

Good health is very important for everyone because it helps us live a happy and active life. Regular health checkups allow people to find out if they have any illnesses early and get treatment before the problem becomes serious. At Oroquieta City Health Center, many people come regularly for their checkups and medical care. This health center is an important place for the community because it helps keep people healthy.

However, many patients face problems when they visit the health center. One of the biggest problems is overcrowding. Too many people come at the same time, and the waiting area becomes very crowded. Because of this, patients have to wait for a long time before they can see a doctor or nurse. Long waiting times make people tired and frustrated. Some patients miss their meals, work, or other important activities because they have to wait in long lines.

The current queuing system at the health center is old and not effective. Patients get a number and wait in line, but there is no way to know how long the wait will be. This process makes the waiting experience hard for patients and also puts pressure on the health workers. The staff become busy managing the long queues, which affects their work and health.

This report will explain the problems caused by the current queuing system at Oroquieta City Health Center. It will also discuss how these problems affect patients, health workers, and the community. Finally, this report will suggest ways to improve the system to make waiting times shorter and the whole experience better for everyone.

**Community Problem Description**

1. The health center still uses an old queuing system. Patients receive a number and wait in line for their turn to get treated.
2. This queuing system is not efficient. Patients must stay at the health center for a long time while waiting. Waiting in line causes them to miss meals, work, and other daily tasks. It takes a lot of time and effort just to get medical help.
3. Patients cannot easily check their appointment schedule. Without a digital system, patients do not know exactly when they will be seen. They have to wait at the health center and cannot plan their other activities while waiting for their turn.

**Relevance and Impact Assessment**

The Oroquieta City Health Center in Misamis Occidental is an important place for public health. It offers regular health checkups that help detect diseases early and prevent serious illness. But overcrowding causes long waiting times and delays in medical care. This problem is common in many parts of the Philippines, especially where people have to pay a lot for health services or where health centers are hard to reach.

1. Patients Impact: Patients are affected because the manual queuing system with paper numbers causes overcrowding and long lines. This makes patients skip meals, miss other important activities, and sometimes return another day, doubling their travel costs.
2. Healthcare Workers Impact: Health workers have to deal with many patients at once. This causes overtime, missed meals, and less rest, which can harm their health and reduce their ability to work well.
3. Community and Economic Impact: Long waits at the health center cause patients to miss work or other productive activities. This loss of time affects the community economy and overall well-being.

**Conclusion and Recommendations**

Fixing the long waiting times and crowded lines at Oroquieta City Health Center needs real, practical changes. The following steps can help solve these problems and make the experience better for patients and staff:

1. Set up a digital queue management system: Use a computer-based system to organize patient appointments and queues. This will help patients know their schedule and reduce the need to wait in line for hours.

2. Make appointment scheduling clear and easy: Allow patients to book their checkups for specific times. This helps avoid too many people coming at once and makes waiting times shorter.

3. Train staff to use the new system: Teach health center staff how to use the digital queuing system so everything runs smoothly.

4. Inform and guide patients: Let patients know how the new system works and how it helps them save time. Give simple instructions so everyone can use it easily.

5. Check and improve the system regularly: Listen to feedback from patients and staff. Make changes as needed to keep the system working well and to fix any problems quickly.

By making these improvements, Oroquieta City Health Center can provide faster, more organized service. Patients will spend less time waiting and more time on other important things, while staff can work more efficiently and with less stress. This will help build trust and satisfaction among everyone who uses the health center.

**TECHNOLOGICAL SOLUTION PROPOSAL**

**SmartQHealth: A Python Based Patient Queue Management System**

**I. Overview of the Proposed Solution**

SmartQHealth Console is a Python based, menu driven application created to solve the problem of long waiting times and overcrowding at Oroquieta City Health Center. The system uses a simple computer program to help organize patient appointments and queues. Patients can register, receive a unique patient number, and get a scheduled time for their checkup. Staff can view and manage the queue, mark attendance, and keep digital records. This solution is designed for easy use by both patients and staff.

Key Features:

* Register patients and collect their details.
* Assign unique patient numbers and appointment times.
* Let patients check their schedule.
* Allow staff to view, process, and update the queue.
* Save and view patient records for future use.

**II. Algorithms and Data Structures Used**

SmartQHealth uses basic Python lists, dictionaries, functions, and data structures to manage the queue, patient data and appointments efficiently. The code is divided into modules for easy management.

**A. Core Features and Algorithms**

Patient Registration and Scheduling

* Patients enter their name, age, gender, birthdate, phone number, and address.
* The program checks if the information is correct (for example, age must be a number, phone number must be 11 digits).
* Each patient gets a unique patient number (using a random 4-digit code that is not repeated).
* The system assigns the next available appointment time from a list of time slots. If all slots for the day are full, the next day is used.
* All patient details are saved in lists and dictionaries.

B. Queue Management

* The system keeps lists of all patient numbers, schedules, and other details.
* A dictionary called pstatus keeps track of each patient’s status (Pending, Attended, or Unattended).
* Staff can view the queue, see patient details, and process (mark) patients as attended or unattended.

C. Schedule Lookup

* Patients can check their schedule by entering their patient number. The system shows their name, details, and appointment time.

D. Record Keeping

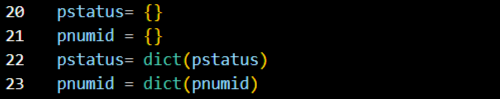
* All patient records are saved in a formatted text file (Patients.txt). This file can be viewed by staff to see the history of all patients.

**B. Key Data Structures:**

* Lists:

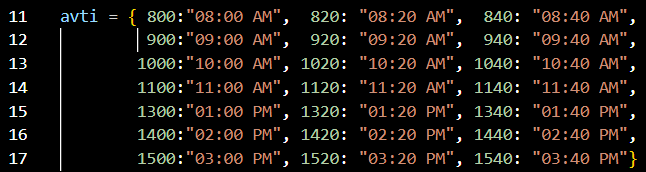
allpsc: All scheduled appointment times

allpnu: All patient numbers

Dictionaries: 

pstatus: Patient status (Pending, Attended, Unattended)

pnumid: Maps patient number to their info and schedule

3. Appointment Times:

* avti: Dictionary mapping numbers (like 800) to times (like "08:00 AM")

4. Patient Record File:

* PRfile: Store Patients Records

**III. Minimum Viable Product (MVP) Prototype of SmartQ Health.**

The application is divided into modules:

**A. TUI\_display.py:**

import time

import os

rst = "\033[0m"

cyc = "\033[96m"

grc = "\033[92m"

yec = "\033[93m"

rec = "\033[91m"

def adjust\_Y(nu\_to\_ad):

    for i in range(nu\_to\_ad):

        print()

def ma\_mo\_d():

    os.system('cls')

    adjust\_Y(4)

    print(" "\*70+cyc+"CITY HEALTH CENTER")

    print(" "\*63+cyc+"SmartQ Health Management System"+rst)

    print()

    print(" "\*57+"╔══════════════════════════════════════════╗")

    print(" "\*57+"║                "+cyc+"Main Menu"+rst+"                 ║")

    print(" "\*57+"╠══════════════════════════════════════════╣")

    print(" "\*57+"║ "+grc+"[1] Patient Mode "+rst+"                        ║")

    print(" "\*57+"║ "+grc+"[2] Admin Mode   "+rst+"                        ║")

    print(" "\*57+"║ "+grc+"[3] Exit         "+rst+"                        ║")

    print(" "\*57+"╚══════════════════════════════════════════╝")

def ac\_ad\_d():

    os.system('cls')

    adjust\_Y(10)

    print(" "\*57+"╔══════════════════════════════════════════╗")

    print(" "\*57+"║           "+yec+"Accessing Admin Mode"+rst+"           ║")

    print(" "\*57+"╚══════════════════════════════════════════╝")

def ex\_pr\_d():

    os.system('cls')

    adjust\_Y(10)

    print(" "\*57+"╔══════════════════════════════════════════╗")

    print(" "\*57+"║      "+yec+"Are you sure you want to exit"+rst+"       ║")

    print(" "\*57+"║                                          ║")

    print(" "\*57+"║         "+grc+"[1] No           "+rec+"[2] Yes"+rst+"         ║")

    print(" "\*57+"╚══════════════════════════════════════════╝")

def pa\_mo\_d():

    os.system('cls')

    adjust\_Y(4)

    print(" "\*70+cyc+"CITY HEALTH CENTER")

    print(" "\*63+"SmartQ Health Management System"+rst)

    print()

    print(" "\*57+"╔══════════════════════════════════════════╗")

    print(" "\*57+"║               "+cyc+"Patient Mode           "+rst+"    ║")

    print(" "\*57+"╠══════════════════════════════════════════╣")

    print(" "\*57+"║ "+grc+"[1] Add Patient to Appointment Queue "+rst+"    ║")

    print(" "\*57+"║ "+grc+"[2] View Patient Schedule            "+rst+"    ║")

    print(" "\*57+"║ "+grc+"[3] Back to Main Menu                "+rst+"    ║")

    print(" "\*57+"╚══════════════════════════════════════════╝")

#Add patient to Queue

def bo\_ap\_d():

    os.system('cls')

    print(" "\*43+"╔════════════════════════════════════════════════════════════════════════╗")

    print(" "\*43+"║                            "+cyc+"Book Appointment"+rst+"                            ║")

    print(" "\*43+"╚════════════════════════════════════════════════════════════════════════╝")

def re\_in\_d():

    os.system('cls')

    print(" "\*43+"╔════════════════════════════════════════════════════════════════════════╗")

    print(" "\*43+"║                       "+cyc+"Review Patient Information"+rst+"                       ║")

    print(" "\*43+"╠════════════════════════════════════════════════════════════════════════╣")

def ces\_d():

    print(" "\*43+"╠════════════════════════════════════════════════════════════════════════╣")

    print(" "\*43+"║    "+rec+"[1] Cancel Booking    "+yec+"[2] Edit Details    "+grc+"[3] Submit Appointment    "+rst+"║")

    print(" "\*43+"╚════════════════════════════════════════════════════════════════════════╝")

def ed\_in\_d():

    os.system('cls')

    print(" "\*43+"╔════════════════════════════════════════════════════════════════════════╗")

    print(" "\*43+"║                            "+cyc+"Edit Information"+rst+"                            ║")

    print(" "\*43+"╠════════════════════════════════════════════════════════════════════════╣")

def en\_\_ed\_in\_d():

    print(" "\*43+"╠════════════════════════════════════════════════════════════════════════╣")

    print(" "\*43+"╚════════════════════════════════════════════════════════════════════════╝")

def pa\_bo\_su\_d():

    print(" "\*43+"╔════════════════════════════════════════════════════════════════════════╗")

    print(" "\*43+"║                       "+grc+"Patient Booked Successfully"+rst+"                      ║")

    print(" "\*43+"╠════════════════════════════════════════════════════════════════════════╣")

def vi\_pa\_sc\_d():

    print(" "\*43+"╔════════════════════════════════════════════════════════════════════════╗")

    print(" "\*43+"║                          "+cyc+"View Patient Schedule"+rst+"                         ║")

    print(" "\*43+"╚════════════════════════════════════════════════════════════════════════╝")

def ad\_mo\_d():

    os.system('cls')

    adjust\_Y(4)

    print(" "\*70+cyc+"CITY HEALTH CENTER")

    print(" "\*63+"SmartQ Health Management System"+rst)

    print()

    print(" "\*57+"╔══════════════════════════════════════════╗")

    print(" "\*57+"║                "+cyc+"Admin Mode"+rst+"                ║")

    print(" "\*57+"╠══════════════════════════════════════════╣")

    print(" "\*57+"║ "+grc+"[1] View Patient Queue   "+rst+"                ║")

    print(" "\*57+"║ "+grc+"[2] Process Next Patient "+rst+"                ║")

    print(" "\*57+"║ "+grc+"[3] View All Patients    "+rst+"                ║")

    print(" "\*57+"║ "+grc+"[4] View Patient Records "+rst+"                ║")

    print(" "\*57+"║ "+grc+"[5] Back to Main Menu    "+rst+"                ║")

    print(" "\*57+"╚══════════════════════════════════════════╝")

# feedback notif

def pa\_nu\_no\_fo\_d():

    print(" "\*43+"┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┓")

    print(" "\*43+"┃                        "+rec+"Patient number not found"+rst+"                        ┃")

    print(" "\*43+"┗━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┛")

    time.sleep(1.25)

def cu\_no\_ne\_pa\_d():

    os.system('cls')

    adjust\_Y(10)

    print(" "\*43+"┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┓")

    print(" "\*43+"┃                       "+yec+"No Next Patient Available"+rst+"                        ┃")

    print(" "\*43+"┗━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┛")

    input(" "\*44+cyc+"Enter to proceed")

def num\_only\_s():

    print(" "\*57+"┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┓")

    print(" "\*57+"┃        "+yec+"Please enter numbers only."+rst+"        ┃")

    print(" "\*57+"┗━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┛")

    time.sleep(1.25)

def num\_only\_l():

    print(" "\*43+"┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┓")

    print(" "\*43+"┃                       "+yec+"Please enter numbers only."+rst+"                       ┃")

    print(" "\*43+"┗━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┛")

    time.sleep(1.25)

def invld\_ch\_s():

    print(" "\*57+"┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┓")

    print(" "\*57+"┃              "+rec+"Invalid choice"+rst+"              ┃")

    print(" "\*57+"┗━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┛")

    time.sleep(1.25)

def invld\_ch\_l():

    print(" "\*43+"┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┓")

    print(" "\*43+"┃                             "+rec+"Invalid Choice"+rst+"                             ┃")

    print(" "\*43+"┗━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┛")

    time.sleep(1.25)

def in\_co\_d():

    os.system('cls')

    adjust\_Y(10)

    print(" "\*57+"┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┓")

    print(" "\*57+"┃               "+rec+"Invalid Code"+rst+"               ┃")

    print(" "\*57+"┗━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┛")

    time.sleep(1.25)

**B. allvariables.py:**

import os

import time

from TUI\_display import\*

from random import randint

from datetime import datetime, timedelta

rst = "\033[0m"

cyc = "\033[96m"

grc = "\033[92m"

yec = "\033[93m"

rec = "\033[91m"

rdate = datetime.today()

t24 = int(rdate.strftime("%H%M"))

inquire = rdate.strftime("%Y-%m-%d %I:%M %p")

t = 0

avti = { 800:"08:00 AM",  820: "08:20 AM",  840: "08:40 AM",

         900:"09:00 AM",  920: "09:20 AM",  940: "09:40 AM",

        1000:"10:00 AM", 1020: "10:20 AM", 1040: "10:40 AM",

        1100:"11:00 AM", 1120: "11:20 AM", 1140: "11:40 AM",

        1300:"01:00 PM", 1320: "01:20 PM", 1340: "01:40 PM",

        1400:"02:00 PM", 1420: "02:20 PM", 1440: "02:40 PM",

        1500:"03:00 PM", 1520: "03:20 PM", 1540: "03:40 PM"}

otime = avti.keys()

otime = tuple(otime)

pstatus= {}

pnumid = {}

pstatus= dict(pstatus)

pnumid = dict(pnumid)

allpsc  =[]

allpnu = []

PRfile = "Patients.txt"

**C. def\_of\_option.py:**

from allvariables import \*

from TUI\_display import \*

def create\_fh():

    if os.path.exists(PRfile):

        pass

    else:

        with open(PRfile,"w", encoding='utf-8') as file:

            file.write("\n╔"+("═"\*190)+"╗")

            file.write("\n║"+(" "\*87)+"Patients Records"+(" "\*87+"║"))

            file.write("\n╠"+("═"\*21)+"╤"+("═"\*36)+"╤"+("═"\*5)+"╤"+("═"\*8)+"╤"+("═"\*13)+"╤"+("═"\*14)+"╤"+("═"\*51)+"╤"+("═"\*22)+"╤"+("═"\*12)+"╣")

            file.write("\n║     Visit Date      │"+(" "\*16)+"Name"+(" "\*16)+"│ Age │ Gender │  Birthdate  │ Phone Number │"+(" "\*22)+"Address"+(" "\*22)+"│ Health Check Schedle │   Status   ║")

            file.write("\n╟"+("─"\*21)+"┼"+("─"\*36)+"┼"+("─"\*5)+"┼"+("─"\*8)+"┼"+("─"\*13)+"┼"+("─"\*14)+"┼"+("─"\*51)+"┼"+("─"\*22)+"┼"+("─"\*12)+"╢")

def in\_info(question):

    while True:

        res = input(question).title()

        if res != "":

            if question == " "\*44+"Name"+(" "\*21)+": " or question == " "\*44+"Address"+(" "\*18)+": ":

                if res.isdigit():

                    if question == " "\*44+"Name"+(" "\*21)+": ":

                        print(" "\*44+yec+"Name cannot be numbers only."+rst)

                    elif question == " "\*44+"Address"+(" "\*18)+": ":

                        print(" "\*44+yec+"Address cannot be numbers only."+rst)

                else:

                    break

            elif question == " "\*44+"Age"+(" "\*22)+": ":

                if res.isdigit():

                    if int(res) > 130:

                        print(" "\*44+rec+"Invalid age"+rst)

                    else:

                        res == res

                        break

                else:

                    print(" "\*44+yec+"Please enter numbers only."+rst)

            elif question == " "\*44+"Phone Number (11 digits) : ":

                if res.isdigit():

                    if len(res) != 11:

                        print(" "\*44+rec+"Invalid number"+rst)

                    else:

                        res == res

                        break

                else:

                    print(" "\*44+yec+"Please enter numbers only."+rst)

            elif question == " "\*44+"Gender (M/F)"+(" "\*13)+": ":

                if res.isalpha():

                    if res == "Male" or res == "Female" or res == "M" or res == "F":

                        if res == "M":

                            res = "Male"

                            break

                        elif res == "F":

                            res = "Female"

                            break

                        break

                    else:

                        print(" "\*44+rec+"Invalid Gender"+rst)

                else:

                    print(" "\*44+yec+"Please enter letters only."+rst)

            elif question == " "\*44+"Birthdate (YYYY-MM-DD)   : ":

                if len(res)== 10 and res[0:4].isdigit() and int(res[0:4]) > 1900  and int(res[0:4]) < 2025 and (res[4:5] == " "or res[4:5] == '/'or res[4:5] =='-') and res[5:7].isdigit() and int(res[5:7]) >= 1 and int(res[5:7])  <= 12 and (res[4:5] == " "or res[4:5] == '/'or res[4:5] =='-') and res[8:10].isdigit() and int(res[8:10]) >= 1 and int(res[8:10])  <= 31:

                    res = res[0:4]+"-"+res[5:7]+"-"+res[8:10]

                    break

                else:

                    print(" "\*44+rec+"Invalid Date"+rst)

        elif res == "":

            print(" "\*44+yec+"This field cannot be empty."+rst)

    return res

def ge\_pnu():

    pnu = ""

    while True:

        for i in range(4):

            rnum=str(randint(0,9))

            pnu+=rnum

        if pnu in allpnu:

            continue

        else:

            break

    return pnu

def ge\_psc():

    global rdate

    global t

    global t24

    if len(allpsc) == 0:

        while t < len(otime) and otime[t] < t24:

            t += 1

        if t >= len(otime):

            rdate = rdate + timedelta(days=1)

            t = 0

        sch = rdate.strftime("%Y-%m-%d ") + avti[otime[t]]

        t += 1

        return sch

    elif len(allpsc) != 0:

        lst = allpsc[-1][11:]

        lst\_key = None

        for k, v in avti.items():

            if v == lst:

                lst\_key = k

                break

        if lst\_key is not None and otime.index(lst\_key) < len(otime) - 1:

            t = otime.index(lst\_key) + 1

            sch = rdate.strftime("%Y-%m-%d ") + avti[otime[t]]

            t += 1

            return sch

        else:

            rdate = rdate + timedelta(days=1)

            t = 0

            sch = rdate.strftime("%Y-%m-%d ") + avti[otime[t]]

            t += 1

            return sch

def file\_handling():

    s = 0

    space = [20, 35, 4, 7, 12, 13, 50,21, 11]

    with open(PRfile, "a",encoding='utf-8') as file:

        for r in allpnu:

            for c in pstatus.get(r):

                if space[s] == 20:

                    file.write("\n║ "+c+(" "\*(space[s]-len(c))))

                elif space[s] == 11:

                    file.write("│ "+c+(" "\*(space[s]-len(c)))+"║")

                else:

                    file.write("│ "+c+(" "\*(space[s]-len(c))))

                s+=1

            s=0

**D. SmartQ.py:**

from allvariables import \*

from def\_of\_option import \*

create\_fh()

def mode():

    os.system('cls')

    while True:

        ma\_mo\_d()

        try:

            ch = int(input(" "\*59+cyc+"Enter number of your Choice: "+rst))

            if ch == 1:

                os.system('cls')

                patient\_mode()

                os.system('cls')

            elif ch == 2:

                ac\_ad\_d()

                ach=input(" "\*58+cyc+"Please Enter Administration CODE : "+rst)

                if ach == "hello world":

                    os.system('cls')

                    admin\_mode()

                    os.system('cls')

                else:

                    in\_co\_d()

            elif ch == 3:

                while True:

                    ex\_pr\_d()

                    try:

                        ch1 = int(input(" "\*59+cyc+"Enter the number of your Choice: "+rst))

                        if ch1 == 1:

                            break

                        elif ch1 == 2:

                            break

                        else:

                            invld\_ch\_s()

                            continue

                    except ValueError:

                        num\_only\_s()

                if ch1 == 1:

                    continue

                elif ch1 == 2:

                    file\_handling()

                    break

            else:

                invld\_ch\_s()

        except ValueError:

            num\_only\_s()

def patient\_mode():

    while True:

        pa\_mo\_d()

        try:

            ch = int(input(" "\*59+cyc+"Enter number of your Choice: "+rst))

            if ch == 1:

                os.system('cls')

                add\_patient\_to\_queue()

                os.system('cls')

            elif ch == 2:

                os.system('cls')

                view\_patient\_schedule()

                os.system('cls')

            elif ch == 3:

                break

            else:

                invld\_ch\_s()

        except ValueError:

            num\_only\_s()

    return

def add\_patient\_to\_queue():

    i=0

    bo\_ap\_d()

    inna = in\_info(" "\*44+"Name"+(" "\*21)+": ")

    inag = in\_info(" "\*44+"Age"+(" "\*22)+": ")

    inge = in\_info(" "\*44+"Gender (M/F)"+(" "\*13)+": ")

    inbi = in\_info(" "\*44+"Birthdate (YYYY-MM-DD)   : ")

    ince = in\_info(" "\*44+"Phone Number (11 digits) : ")

    inad = in\_info(" "\*44+"Address"+(" "\*18)+": ")

    p\_info = [inna,inag,inge,inbi,ince,inad]

    label = ["Name","Age","Gender","Birthdate","Phone Num","Address"]

    while True:

        re\_in\_d()

        for e\_info in p\_info:

            print(" "\*43+"║ "+label[i]+(" "\*(11-len(label[i])))+": "+e\_info+(" "\*(58-len(e\_info)))+"║")

            i+=1

        i=0

        ces\_d()

        try:

            ch = int(input(" "\*44+cyc+"Enter the number of your choice: "+rst))

            if ch == 1:

                return

            elif ch == 2:

                while True:

                    ed\_in\_d()

                    for e\_info in p\_info:

                        print(" "\*43+f"║ [{i+1}] "+label[i]+(" "\*(11-len(label[i])))+": "+e\_info+(" "\*(54-len(e\_info)))+"║")

                        i+=1

                    i=0

                    en\_\_ed\_in\_d()

                    try:

                        ech = int(input(" "\*44+cyc+"Enter the number you want to edit: "+rst))

                        if ech == 1:

                            inna = in\_info(" "\*44+"Name"+(" "\*21)+": ")

                        elif ech == 2:

                            inag = in\_info(" "\*44+"Age"+(" "\*22)+": ")

                        elif ech == 3:

                            inge = in\_info(" "\*44+"Gender (M/F)"+(" "\*13)+": ")

                        elif ech == 4:

                            inbi = in\_info(" "\*44+"Birthdate (YYYY-MM-DD)   : ")

                        elif ech == 5:

                            ince = in\_info(" "\*44+"Phone Number (11 digits) : ")

                        elif ech == 6:

                            inad = in\_info(" "\*44+"Address"+(" "\*18)+": ")

                        else:

                            invld\_ch\_l()

                            continue

                        p\_info = [inna,inag,inge,inbi,ince,inad]

                        break

                    except ValueError:

                        num\_only\_l()

            elif ch == 3:

                break

            else:

                invld\_ch\_l()

        except ValueError:

            num\_only\_l()

    gepn = ge\_pnu()

    geps = ge\_psc()

    pnumid.update({gepn:[inna,inag,inge,inbi,ince,inad,geps]})

    pstatus.update({gepn:[inquire,inna,inag,inge,inbi,ince,inad,geps,"Pending"]})

    allpnu.append(gepn)

    allpsc.append(geps)

    pa\_bo\_su\_d()

    print(" "\*43+"║ "+cyc+"Save your Patient Number : "+yec+f"{gepn}"+(" "\*40)+rst+"║")

    print(" "\*43+"║ "+cyc+"Your Schedule            : "+yec+f"{geps}"+(" "\*25)+rst+"║")

    print(" "\*43+"╚"+("═"\*72)+"╝")

    input(" "\*44+cyc+"Press Enter to Proceed"+rst)

    os.system('cls')

def view\_patient\_schedule():

    label = ["Name","Age","Gender","Birthdate","Phone Num","Address","Schedule"]

    l = 0

    vi\_pa\_sc\_d()

    entry = input(" "\*44+cyc+"Enter Your Patient Number: "+rst)

    if entry in allpnu:

            print(" "\*43+"╔"+("═"\*72)+"╗")

            print(" "\*43+f"║ Patient Number: {entry}"+(" "\*51)+"║")

            print(" "\*43+"╠"+("═"\*72)+"╣")

            for info in pnumid.get(entry):

                if l == 6:

                    print(" "\*43+"╠"+("═"\*72)+"╣")

                    print(" "\*43+"║ "+cyc+label[l]+(" "\*(10-len(label[l])))+": "+yec+info+(" "\*(59-len(info)))+rst+"║")

                    print(" "\*43+"╚"+("═"\*72)+"╝")

                else:

                    print(" "\*43+"║ "+label[l]+(" "\*(10-len(label[l])))+": "+info+(" "\*(59-len(info)))+"║")

                l+=1

            l=0

    else:

        pa\_nu\_no\_fo\_d()

    input(" "\*44+cyc+"Press Enter to proceed"+rst)

def admin\_mode():

    while True:

        ad\_mo\_d()

        try:

            ch = int(input(" "\*59+cyc+"Enter number of your Choice: "+rst))

            if ch == 1:

                os.system('cls')

                view\_patient\_queue()

                os.system('cls')

            elif ch == 2:

                os.system('cls')

                process\_next\_patient()

                os.system('cls')

            elif ch == 3:

                os.system('cls')

                view\_all\_patients()

                os.system('cls')

            elif ch == 4:

                os.system('cls')

                view\_patients\_records()

                os.system('cls')

            elif ch == 5:

                break

            else:

                invld\_ch\_s()

        except ValueError:

            num\_only\_s()

    return

def view\_patient\_queue():

    i=0

    print(" "\*40+"╔"+("═"\*80)+"╗")

    print(" "\*40+f"║                                 "+cyc+"Patient Queue"+rst+"                                  ║")

    print(" "\*40+"╠"+("═"\*5)+"╤"+("═"\*11)+"╤"+("═"\*40)+"╤"+("═"\*21)+"╣")

    print(" "\*40+f"║ No. │ Patient # │"+(" "\*18)+"Name"+(" "\*18)+"│      Schedule       ║")

    print(" "\*40+"╟"+("─"\*5)+"┼"+("─"\*11)+"┼"+("─"\*40)+"┼"+("─"\*21)+"╢")

    for p in allpnu:

        print(" "\*40+f"║  {i+1}"+(" "\*(3-len(str(i))))+ "│"+(" "\*3)+allpnu[i]+(" "\*4)+"│"+" "+pstatus.get(p)[1]+(" "\*(39-len(pstatus.get(p)[1])))+"│ "+pstatus.get(p)[7]+" ║")

        i+=1

    print(" "\*40+"╚"+("═"\*5)+"╧"+("═"\*11)+"╧"+("═"\*40)+"╧"+("═"\*21)+"╝")

    input(" "\*43+cyc+"Press Enter to Proceed"+rst)

def view\_all\_patients():

    s=0

    space = [20, 35, 4, 7, 12, 13, 50,21, 11]

    print("╔"+("═"\*190)+"╗")

    print("║"+(" "\*86)+"View All Patients"+(" "\*87+"║"))

    print("╠"+("═"\*21)+"╤"+("═"\*36)+"╤"+("═"\*5)+"╤"+("═"\*8)+"╤"+("═"\*13)+"╤"+("═"\*14)+"╤"+("═"\*51)+"╤"+("═"\*22)+"╤"+("═"\*12)+"╣")

    print("║     Visit Date      │"+(" "\*16)+"Name"+(" "\*16)+"│ Age │ Gender │  Birthdate  │ Phone Number │"+(" "\*22)+"Address"+(" "\*22)+"│ Health Check Schedle │   Status   ║")

    print("╟"+("─"\*21)+"┼"+("─"\*36)+"┼"+("─"\*5)+"┼"+("─"\*8)+"┼"+("─"\*13)+"┼"+("─"\*14)+"┼"+("─"\*51)+"┼"+("─"\*22)+"┼"+("─"\*12)+"╢")

    for r in pstatus.values():

        for c in r:

            if space[s] == 11:

                print("│ "+c+(" "\*(space[s]-len(c)))+"║",end="")

            elif space[s] == 20:

                print("║ "+c+(" "\*(space[s]-len(c))),end="")

            else:

             print("│ "+c+(" "\*(space[s]-len(c))),end="")

            s+=1

        s=0

        print()

    print("╚"+("═"\*21)+"╧"+("═"\*36)+"╧"+("═"\*5)+"╧"+("═"\*8)+"╧"+("═"\*13)+"╧"+("═"\*14)+"╧"+("═"\*51)+"╧"+("═"\*22)+"╧"+("═"\*12)+"╝")

    input(cyc+"  Press Enter to Proceed"+rst)

def process\_next\_patient():

    s = 0

    space = [20, 35, 4, 7, 12, 13, 50,21, 11]

    i = 0

    label = ["Name","Age","Gender","Birthdate","Phone Num","Address","Schedule"]

    while True:

        if len(allpnu) != 0:

            while True:

                os.system('cls')

                print(" "\*43+"╔"+("═"\*72)+"╗")

                print(" "\*43+f"║ "+cyc+"Patient Number: "+yec+f"{allpnu[0]}"+(" "\*51)+rst+"║")

                print(" "\*43+"╚"+("═"\*72)+"╝")

                for info in pnumid.get(allpnu[0]):

                    print(" "\*43+"│ "+label[i]+(" "\*(11-len(label[i])))+": "+info+(" "\*(58-len(info)))+"│")

                    i+=1

                i=0

                print(" "\*43+"╔"+("═"\*72)+"╗")

                print(" "\*43+"║         "+rec+"[1] Exit          "+yec+"[2] Unattended          "+grc+"[3] Attended         "+rst+"║")

                print(" "\*43+"╚"+("═"\*72)+"╝")

                try:

                    ch=int(input(" "\*43+cyc+"  Enter your choice: "+rst))

                    if ch == 1:

                        break

                    elif ch == 2:

                        pstatus.get(allpnu[0])[8] = "Unattended"

                        with open(PRfile, "a",encoding='utf-8') as file:

                            for info0 in pstatus.get(allpnu[0]):

                                if space[s] == 20:

                                    file.write("\n║ "+info0+(" "\*(space[s]-len(info0))))

                                elif space[s] == 11:

                                    file.write("│ "+info0+(" "\*(space[s]-len(info0)))+"║")

                                else:

                                    file.write("│ "+info0+(" "\*(space[s]-len(info0))))

                                s+=1

                            s=0

                        allpnu.remove(allpnu[0])

                        break

                    elif ch == 3:

                        pstatus.get(allpnu[0])[8] = "Attended"

                        with open(PRfile, "a",encoding='utf-8') as file:

                            for info0 in pstatus.get(allpnu[0]):

                                if space[s] == 20:

                                    file.write("\n║ "+info0+(" "\*(space[s]-len(info0))))

                                elif space[s] == 11:

                                    file.write("│ "+info0+(" "\*(space[s]-len(info0)))+"║")

                                else:

                                    file.write("│ "+info0+(" "\*(space[s]-len(info0))))

                                s+=1

                            s=0

                        allpnu.remove(allpnu[0])

                        break

                    else:

                        invld\_ch\_l()

                        continue

                except ValueError:

                    num\_only\_l()

                    continue

            if ch == 1:

                break

        else:

            cu\_no\_ne\_pa\_d()

            break

def view\_patients\_records():

    with open(PRfile,"r",encoding="utf-8") as file:

        info = file.read()

        print(info)

    print("╚"+("═"\*21)+"╧"+("═"\*36)+"╧"+("═"\*5)+"╧"+("═"\*8)+"╧"+("═"\*13)+"╧"+("═"\*14)+"╧"+("═"\*51)+"╧"+("═"\*22)+"╧"+("═"\*12)+"╝")

    input(cyc+"  Press Enter to Proceed"+rst)

if \_\_name\_\_ == '\_\_main\_\_':

    mode()

**I I I.** Implementation Feasibility**:**

A. Technical Feasibility:

* Programming Language: Python 3.12
* Works on: Any computer with Python (Windows, Mac, Linux)
* No internet needed
* Data Storage: Simple text file, no database needed
* No extra libraries required

B. Educational and Operational Feasibility:

* Users: Health center staff and patients
* Easy to Use: Simple menus and clear prompts
* Training: Only basic computer skills needed

C. Limitations:

* Only one user at a time (no network use)
* Manual data entry (no automatic notifications)
* Text-only interface (no graphics)

IV. Conclusion

SmartQHealth is a practical and easy-to-use digital solution for managing patient queues and appointments at Oroquieta City Health Center. It helps reduce waiting times, keeps things organized, and makes work easier for staff. The program is based on your actual code and can be improved in the future if needed. This system is a good example of how simple technology can solve real problems in the community.

**STS IMPACT ASSESSMENT REPORT**

**Introduction**

As students in higher education, we embrace digital transformation. There is a growing need for efficient and inclusive systems that support administrative and academic operations. At Oroquieta City Health Center, manual managing in queuing to receive medical attention takes a lot of time, some patients also file a leave of absence on their job that can result in negative impact on the community and economy. This report assesses the implementation of the SmartQHealth System(SQHS), and proposes a technological solution from an STS perspective.

**Societal, Ethical, and Environmental Implications**

I. Societal Implications

1. Fair Access to Medical Care. By reducing the crowd, the public healthcare facilities can further provide medical attention to patients, especially for frail individuals. By ensuring people are treated on time, they are more likely not to skip their basic needs.
2. Enhanced Well-Being of Patients. Patients are going to be treated right away. It minimizes the crowd and queue time. This is because of how efficient the system will be. This also enhances public health outcomes, reducing stress and encouraging more frequent use of healthcare services.
3. Enhanced Confidence in Public Health Systems. Reliable, patient-centered healthcare services promote stronger cooperation, better medical devotion, and community involvement, all of which help to build public confidence in government institutions.

II. Ethical Implications

1. Data Privacy and Security: The proposed system must comply with the Data Privacy Act of 2012. It is essential that all collected information such as names, address, birthdate, and log book records. It is securely stored and accessed only by authorized personnel.
2. Fair Use and Transparency. The system is a first-come, first-served queueing system. The system is based on valid and justifiable reasons. The system efficiently manages the flow of requests, ensuring that each individual is attended to in the order they arrive. This approach minimizes wait times and fosters a sense of fairness among users, as everyone has an equal opportunity to receive service without favoritism.
3. Accountability and Responsibility. SmartQHealth tracks schedules ensuring transparency. This empowers patients to engage in their care while promoting fairness for both professionals and patients. Together, this fosters an ethical and trustworthy healthcare system.

III. Environmental Implications

1. Paperless Operations. Transitioning to digital documentation (Numbering) drastically reduces the use of printed materials. This promotes environmentally friendly practices and advances institutional sustainability goals.

**Justification of the Technological Solution’s Benefits**

The implementation of a SmartQHealth system aligns with both institutional modernization goals and national efforts toward digital transformation. The benefits include:

* Increased efficiency through automation of repetitive tasks.
* Greater accuracy and fewer errors compared to manual processes.
* Enhanced public experience through timely queue time.
* Improved staff productivity by catering patients.
* Alignment with ethical standards by ensuring transparency and data security.

Ultimately, SmartQHealth system is not just a technological upgrade, but a strategic intervention that advances the Health Center’s vision of community collaboration, and sustainability.

**Conclusion**

From an STS perspective, Oroquieta City Health Center transformed the queueing problem into fast and efficient medical care. Healthcare access can be improved by reducing overcrowded and improving patient flow. Efficient systems like digital queueing, better staffing, and to optimize resource allocation can foster a more understanding and just environment. This solution goes beyond logistical improvements, timeliness, and inclusivity in public health services. It supports SDG 3's vision of good health and well-being for all.