Jason Bhatnagar

Professor Nguyen

CIS-30E

23 November 2022

CIS-30E Project Part 1 - Documentation

Project Information And Details

My project was **solving a problem** that a patient may face in scheduling and keeping track of their appointment, mainly getting lost in the process of choosing doctors and times. The **objective and purpose of the program** is streamlining and simplifying the process of creating an appointment with very clear instructions and options for each step. It removes any confusion someone may deal with as a result, avoiding possible time conflicts as one example. The **solution/algorithm (goes hand-in-hand)** involves a list of doctors being shown after the patient enters their desired date and chooses between scheduling an appointment with a doctor directly, or scheduling one by selecting a doctor based on their specialty first. Then, the doctors available have their schedules shown with what times they are available and what times are unavailable. This makes it extremely easy for the patient to find the right appointment time based on their personal schedule. Once the patient selects the time slot they want to book, a confirmation message is shown.

The algorithm uses **concurrency** as another tool to generate and print the list of appointment times at the same time. **Modules implemented** in the program were random, datetime, and threading. Code from *data.py* is brought as well. Random randomly generates the appointment schedule (available/occupied times), datetime is used for scheduling and checking doctor availability, threading is used for the main thread, and *data.py* brings in a list of doctors (dictionary), list of patients (dictionary), and a list of appointments (dictionary).

Limitations of the program include times being in military time instead of standard time, and the doctor section is not fully fleshed out to show the appointment for a certain doctor after scheduling one as a patient (I wanted to work on this but unfortunately time was cut short due to UCI's hard deadline and needing my grade earlier to send my transcript). **These limitations can be improved** by making the schedule show standard time and giving doctors more utility by allowing them to see more information on what is happening on the patient scheduling side.

Psuedocode

```
import datetime
from time import sleep, perf_counter
from threading import Thread
from data import lst_appointments, lst_doctors, lst_patients
import random
SUBROUTINE
view_appointments( namestr, dt THEN str, usertype ← 1 )  THEN
   lst ← [ ]
```

```
    IF usertype== 0  THEN
        FOR lap IN lst_appointments
            IF lap[ "dr_name"] == name AND str( lap[ "appointment"] ) . split( " ") [ 0 ] == dt THEN
                lst. append(lap)

        ENDFOR
    ELSE IF usertype== 1  THEN
        FOR lap IN lst_appointments
            IF lap[ "pt_name"] == name AND str( lap[ "appointment"] )  .split( " ") [ 0 ] == dt THEN
                lst. append(lap)

        ENDFOR
    ELSE
        OUTPUT "Please Enter proper user type\n0 for Doctor, and\n1 for patient"
        returnNone

    ENDIF
    returnlst

ENDSUBROUTINE
SUBROUTINE
view_specialty( )
    lst ← [ ]
    FOR ld IN lst_doctors
        IF ld[ "specialty"]  NOT inlst THEN
            lst. append( ld[ "specialty"] )

        ENDIF

    ENDFOR
    returnlst

ENDSUBROUTINE
SUBROUTINE
view_doctors_by_specialty( speciality)
    lst ← [ ]
    FOR ld IN lst_doctors
        IF ld[ "specialty"] == speciality THEN
            lst. append( ld)

        ENDIF

    ENDFOR
    returnlst

ENDSUBROUTINE
SUBROUTINE
show_all_doctors( )
    OUTPUT "Here is the List of Doctors in this Facility:"
    FOR ld IN lst_doctors
        OUTPUT ld"name"] )

    ENDFOR

ENDSUBROUTINE
SUBROUTINE
```

```
get_weekday_from_date( dtdatetime)  THEN
   day_no ← datetime. datetime. date( dt) . weekday( )
   IF day_no== 0  THEN
      return'Monday'
   ELSE IF day_no== 1  THEN
      return'Tuesday'
   ELSE IF day_no== 2  THEN
      return'Wednesday'
   ELSE IF day_no== 3  THEN
      return'Thursday'
   ELSE IF day_no== 4  THEN
      return'Friday'
   ELSE IF day_no== 5  THEN
      return'Saturday'
   ELSE IF day_no== 6  THEN
      return'Sunday'
   ELSE
      returnNone

   ENDIF

ENDSUBROUTINE
SUBROUTINE
check_doctor_availability( dr_name, dt)
   dte ← datetime. datetime. strptime( str(dt) , "%m-%d-%Y")
   day_asked ← get_weekday_from_date(dte)
   days_available ← [ ]
   time_available ← None
   time_slots_available ← [ ]
   FOR ld IN lst_doctors
      IF ld[ "name"] == dr_name THEN
         da_list ← ld[ "visit_days"]
         time_available ← ld[ "visit_hours"]
         FOR dal IN da_list
            days_available. append( dal)

         ENDIF

         ENDFOR

      ENDIF

   ENDFOR
   IF day_askedindays_available THEN
      lst_slots ← create_time_slot( time_available, dte)
      returnTrue, lst_slots

   ENDIF
   returnFalse, None

ENDSUBROUTINE
SUBROUTINE
create_time_slot( time_availablestr, dt THEN datetime)  THEN
   lst_slots ← [ ]
   lst_exact_slots ← [ ]
   s_time, e_time ← time_available. split( "-")
```

```
    yr ← dt. year
    mn ← dt. month
    dat ← dt. day
    starting_time ← datetime. datetime( yr, mn, dat, int( s_time) )
    ending_time ← datetime. datetime( yr, mn, dat, int( e_time) )
    OUTPUT "Starting time: "starting_time)
    OUTPUT "Ending time: "ending_time)
    lst_slots. append( starting_time)
    slot ← starting_time
    WHILE slot!= ending_time
       slot ← slot+ datetime. timedelta( minutes ← 30 )
       lst_slots. append( slot)

    ENDWHILE
    n ← len( lst_slots)
    rnd_pct ← round( random. uniform( 0.6 , 0.7 ) , 2 )
    m ← int( n* rnd_pct)
    rand_lst ← [ ]
    dx ← { }
    FOR j ← m TO :    rand_lst. append( random. randint( 0 , n) )

    ENDFOR
    FOR i IN lsinenumerate( lst_slots)
       IF iinrand_lst THEN
          dx[ i]  ← str(ls) + "|"+ "DOCTOR IS OCCUPIED"
       ELSE
          dx[ i]  ← str(ls) + "|"+ "AVAILABLE"

       ENDIF

    ENDFOR
    lst_exact_slots. append(dx)
    returnlst_exact_slots

ENDSUBROUTINE
SUBROUTINE
create_appointment( lst_slts, id_of_available_slot, dr_name, pat_name)

ENDSUBROUTINE
    latest_id ← - 1
    dx ← { }
    FOR lsap IN lst_appointments
       IF int( lsap[ "id"] ) > latest_id THEN
          latest_id ← int( lsap[ "id"] )

       ENDIF

    ENDFOR
    FOR ls IN lst_slts
       FOR k IN vinls. items( )
          IF int( str( k) . lstrip( ) . rstrip( ) . strip( ) ) == int( id_of_available_slot)  THEN
             original, status ← str( v) . split( "|")
             dt, tm ← original. split( " ")
             IF status== "AVAILABLE" THEN
                status ← "DOCTOR IS OCCUPIED"
                new_id ← latest_id+ 1
```

```
                    dx[ "id"]  ← new_id
                    dx[ "dr_name"]  ← dr_name
                    dx[ "pt_name"]  ← pat_name
                    dx[ "appointment"]  ← original+ "|"+ status
                    OUTPUT "Appointment Created Successfully:"
                    OUTPUT "-------------------------------------------------------------------"
                    OUTPUT "Patient {} to see {} on {} at {}"format( pat_name, dr_name, dt, tm) )
                    OUTPUT "-------------------------------------------------------------------"

                  ENDIF

                ENDIF

              ENDFOR

            ENDFOR
            IF len(dx) > 0  THEN
              lst_appointments. append( dx)

            ENDIF
          SUBROUTINE
          main( )
            WHILE True
              user_type ← int( USERINPUT "Enter 0 if your are Doctor\nEnter 1 if you are Patient\nEnter 9 to
Quit\n")
              IF user_type== 0  THEN
                OUTPUT "Welcome Doctor"
                dr_name ← USERINPUT "Please Enter Doctor's Name to view Appointments:\n"
                dt ← USERINPUT "Please Enter Date to view Appointments (MM-dd-YYYY):\n"
                lst ← view_appointments( dr_name, dt, usertype ← 0 )
                OUTPUT "Appointments Schedule for Dr. {} on {}"format( dr_name, dt) )
                IF len(lst) > 0  THEN
                  FOR l IN lst
                    FOR k IN vinl. items( )
                      OUTPUT "{:<2}\t{}"format( k, v) )

                    ENDFOR

                  ENDFOR
                ELSE
                  OUTPUT "Dr. {} has no scheduled appointments on {}"format( dr_name, dt) )

                ENDIF

                ENDIF
                break

              ENDIF
              IF user_type== 1  THEN
                OUTPUT "Welcome Patient"
                WHILE True
                  pt_name ← USERINPUT "Please Enter Patient's Name:\n"
                  dt ← USERINPUT "Enter Appointment Date (MM-dd-YYYY):\n"
                  pt_option ← int( USERINPUT "Please Enter 1 to Schedule an Appointment:\n")
                  IF pt_option== 1  THEN
                    OUTPUT "Scheduling an Appointment ..."
```

```
    appt_option ← int( USERINPUT "Enter 1 to Schedule Appointment using doctor's
name:\nEnter 2 to ""Schedule Appointment selecting doctor from specialty:\n") )
    IF appt_option== 1  THEN
      show_all_doctors( )
      dr_name ← USERINPUT "Enter Doctor's name:\n"
      is_doc_available, lst_of_slots ← check_doctor_availability( dr_name, dt)
      IF is_doc_available THEN
        OUTPUT "Here is Schedule for {} on {}"format( dr_name, dt) )
        FOR los IN lst_of_slots
          FOR k IN vinlos. items( )
            OUTPUT "{:<5} {}"format( k, v) )

          ENDFOR

        ENDFOR
        tm_slt_id ← USERINPUT "Select Id of Time Slot (e.g. 3):\n"
        create_appointment( lst_of_slots, tm_slt_id, dr_name, pt_name)
      ELSE
        OUTPUT "Doctor {} is not available on {}"format( dr_name, dt) )
    ELSE IF appt_option== 2  THEN
      splt_lst ← view_specialty( )
      OUTPUT "Specialties Available at this Facility:"
      FOR sl IN splt_lst
        OUTPUT sl

      ENDIF

      ENDFOR
      selected_specialty ← USERINPUT "Enter chosen Specialty:\n"
      splt_dr_lst ← view_doctors_by_specialty( selected_specialty)
      OUTPUT "Here is a list of doctors specializing in {}"format( selected_specialty) )
      FOR sdl IN splt_dr_lst
        days ← ""
        FOR dy IN sdl[ "visit_days"]
          days ← days+ " "+ dy

        ENDFOR
        OUTPUT "{:<25} {:<50} {:<15}"format( sdl[ "name"] , days, sdl[ "visit_hours"] ) )

      ENDFOR
      dr_name ← USERINPUT "Enter the name of Doctor you'd like to visit:\n"
      is_doc_available, lst_of_slots ← check_doctor_availability( dr_name, dt)
      IF is_doc_available THEN
        OUTPUT "Here is Schedule for {} on {}"format( dr_name, dt) )
        FOR los IN lst_of_slots
          FOR k IN vinlos. items( )
            OUTPUT "{:<5} {}"format( k, v) )

          ENDFOR

        ENDFOR
        tm_slt_id ← USERINPUT "Select Id of Time Slot (e.g. 3):\n"
        create_appointment( lst_of_slots, tm_slt_id, dr_name, pt_name)
      ELSE
        OUTPUT "Doctor {} is not available on {}"format( dr_name, dt) )
    ELSE
```

```
                OUTPUT "Invalid Option"

              ENDIF
              break
            ELSE
              OUTPUT "Invalid Input"

            ENDIF
            break

          ENDWHILE
          break

        ENDIF
        IF user_type== 9  THEN
          exit( 0 )

        ENDIF

      ENDWHILE

    ENDSUBROUTINE
    IF __name__== "__main__" THEN
      threads ← [ ]
      FOR n ← 3 TO :
        t ← Thread( target ← main( ) )
        threads. append( t)
        t. start( )

      ENDFOR

    ENDIF
```

Example of program simulation/sequence

Providing User Inputs

1. First you will be asked to Enter 0 if you are a doctor or 1 if you are a patient
2. Say you enter 1 (being a patient)
3. You will be asked to Enter Patient's Name
4. Say you enter ‘XYZ’
5. You will be asked to Enter Appointment Date
6. Say you enter 11-04-2022
7. Then you will be prompted to Enter 1 to Schedule Appointment ... enter 1
8. Then you will be asked to Enter 1 if you know doctor you are visiting or 2 if you want to select the doctor from his/her specialty
9. Say you enter 1 - you will be presented with list of All Doctors
10. Enter the name of a doctor (must match exactly as shown)
11. Say you enter ‘Dr. Erica’
12. You will be presented with her available times on that date
13. And you will be asked to enter the ID# of the slot available (leftmost column)
14. Say you enter ‘3’

15. Your appointment will be scheduled and a message displayed

Doctor Views Appointment
1. Start the Program - You get following options
* Enter 0 if you're a Doctor
* Enter 1 if you’re a Patient
* Enter 9 to Quit

You Select '0', you get following prompt

* Welcome Doctor
* Please Enter Doctor's Name to view Appointments:
* You enter: ‘Dr. Nikki Moreno’
* Please Enter Date to view Appointments (MM-DD-YYYY):
* You enter: ‘11-05-2022’
* Appointments Schedule for Dr. Dr. Nikki Moreno on 11-05-2022
Dr. Dr. Nikki Moreno has no scheduled appointments on 11-05-2022