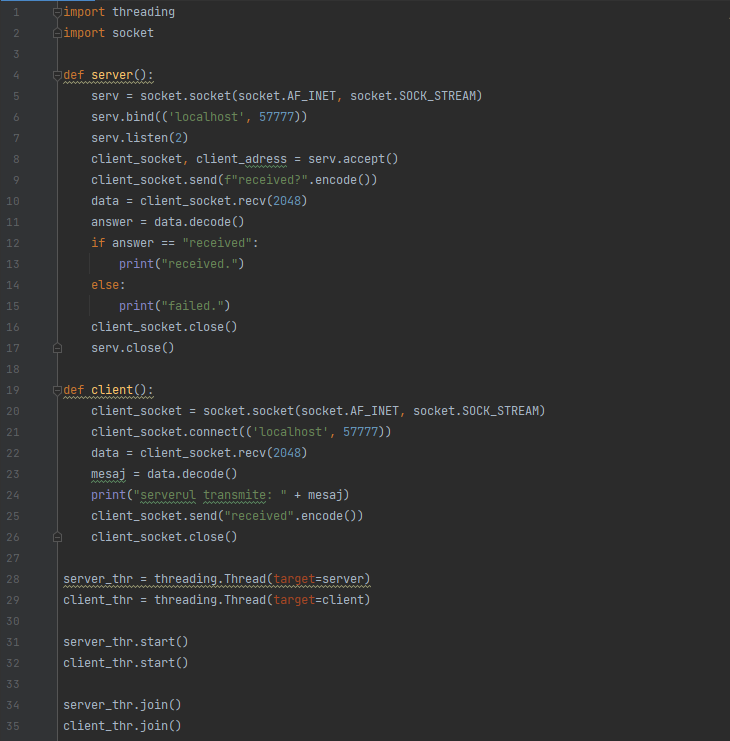
Receptie clinica stomatologica

1. Codul de baza
2. Programul de receptie
3. Scheletul

Am inceput prin a crea un simplu cod pentru a verifica conexiunea intre client si server:



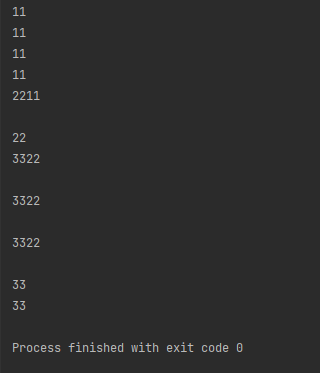
Am separat codul clientului si al serverului in doua functii separate pentru organizare si usurinta de apelare.

Serverul salveaza adresa cat si socketul clientului pentru a-i trimite un mesaj, in cazul meu “received?”. Acesta asteapta mesajul de la client, iar cand acesta ajunge, il testeaza. In situatia fericita in care a primit mesajul corect, se afiseaza afirmatia, altfel se printeaza “failed” pentru a da un semnal ca a fost o problema. La final serverul inchide conexiunea cu clientul, apoi se inchide pe el insusi.

Clientul primeste si salveaza mesajul primit de la server. Trimite serverului mesajul “received” si se inchide.

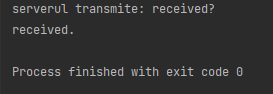
Am folosit bibloteca threading deoarece imi era dificil sa urmaresc codul intr-un singur program. Aceasta permite rularea functiilor server si client simultan, fiind astfel mai realistic. In exemplu\_thr am creat un obiect de tip Thread care imi permite rularea functiei target in tanden cu alta.

Pentru a vizualiza cum functioneaza in general threading-ul mi-am luat un mic exemplu:

Functiile nu ‘asteapta’ pot incepe executarea inainte ca o functie s-a isi fi terminat derulearea.

In final, codul schelet confirma conexiunea:



1. Programul de receptie

Modelul pe care doresc sa il urmez arata in felul urmator:

i) Robotul intreaba daca este o urgenta. In cazul afirmativ, robotul informeaza pacientul ca va fi contactat cat de devreme posibil de un operator si se inchide. In cazul negativ, linia de intrebari continua.

ii) Robotul intreaba la care sectie de stomatologie ar dori sa isi faca pacientul programare.

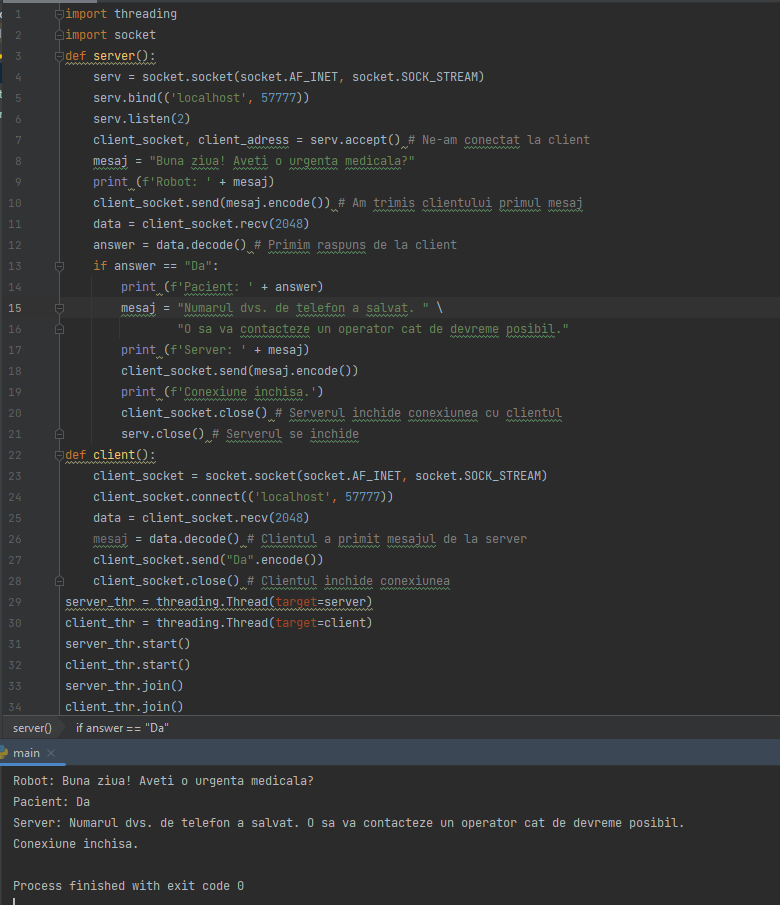
iii) Robotul cere numarul de telefon al pacientului si il informeaza datele vor fi transmise unui operator care va reveni cu optiuni pentru programare.

Raspunsurile clientului ar fi introduse prin functia input(), dar pentru usurinta de vizualizare am luat 2 cazuri.

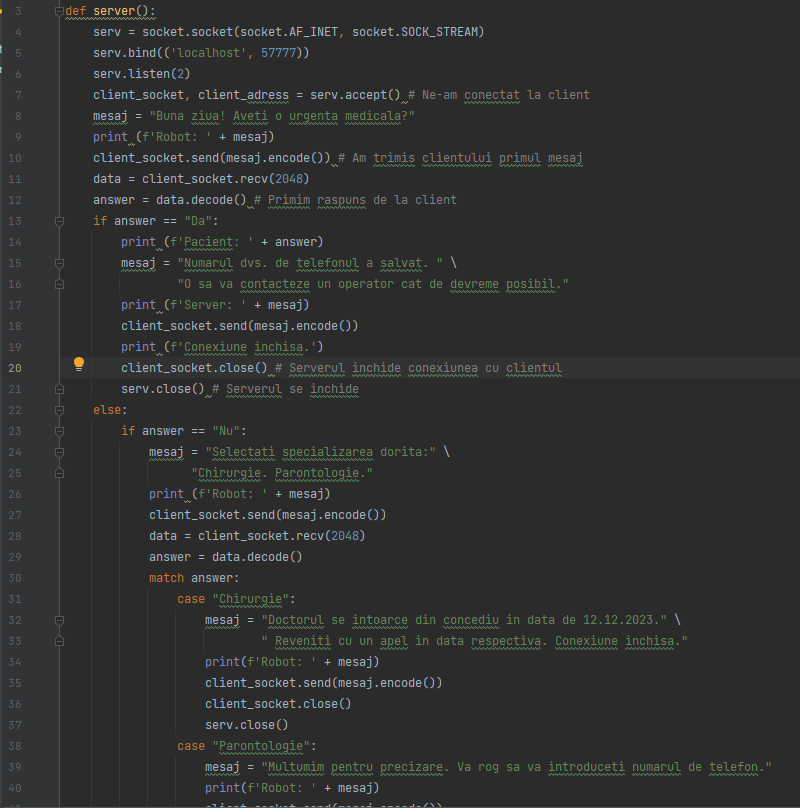
Observatii:

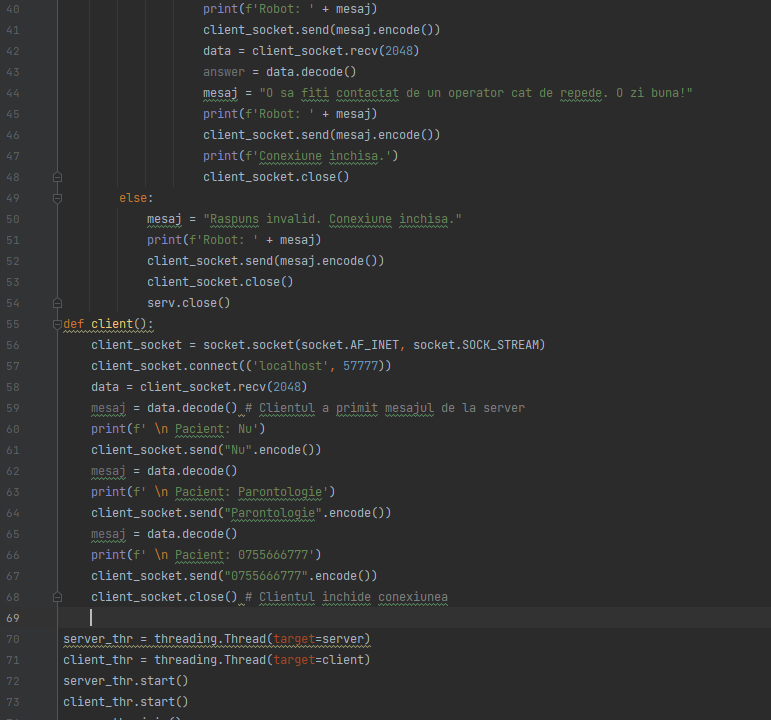
* Cazul urgentei a fost scurtat.
* Am printat conversatia dintre pacient si robot pentru a demonstra ca functioneaza.

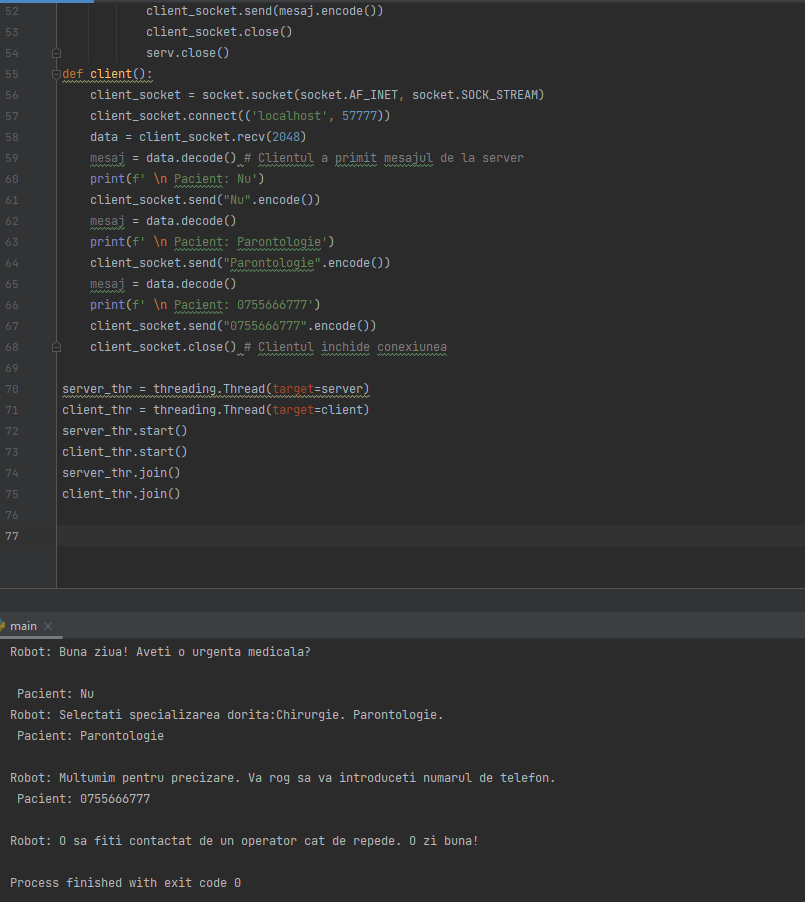
Cazul urgentei:



Cazul uzual:







1. Documentatie
2. Threading: <https://docs.python.org/3/library/threading.html>
3. Basics: <https://www.cs.ubbcluj.ro/~radu.dragos/examples/#Socket_Programming>
4. Sockets: <https://docs.python.org/3/library/socket.html>