Documentatie proiect "Kyte"

Aplicatie pentru comunicare audio, video, text

1.Descrierea proiectului

Kyte este o aplicatie care faciliteaza comunicarea text si audio-video intre doi sau mai multi utilizatori. Programul poate rula in mai multe instante, pe calculatoare diferite care sunt conectate la internet.

Fiecare utilizator detine o lista de adrese de contact pe care o poate consulta si modifica la pornirea programului, urmand sa aleaga un contact sau mai multi in vederea trimiterii unor invitatii si eventual a stabilirii unei convorbiri, respectiv, stabilirea unei conferinte.

2.Lista de cerinte ale aplicatiei

- comunicare audio, video, text
- fiabilitatea sesiunii de comunicare
- mentirea unei liste de contacte
- posibilitatea de a purta o conversatie (2 oameni) sau o conferinta (3 sau mai multi)
- Optional: transfer de fisiere, fereastra de talk-text, feedback vreme

3.Specificatii formale in Z

Serviciul de naming se ocupa cu gestionarea unei baze de date ce contine numele si ip-urile utilizatorilor, facilitand mentinerea unei liste de contacte usor de utilizat. Userul este memorat in baza de date cand acceseaza pentru prima data serviciul, iar pentru a adauga un contact in lista trebuie sa ii precizeze doar numele.

Urmatoarele doua specificatii formale in Z descriu serviciul:

[Ip]
[Name]

----------namingService----------

$data \subseteq Ip \times Name$
$nameContact? : Name$
$address! : Ip$
$\emptyset = \{ x : Ip \mid false \}$

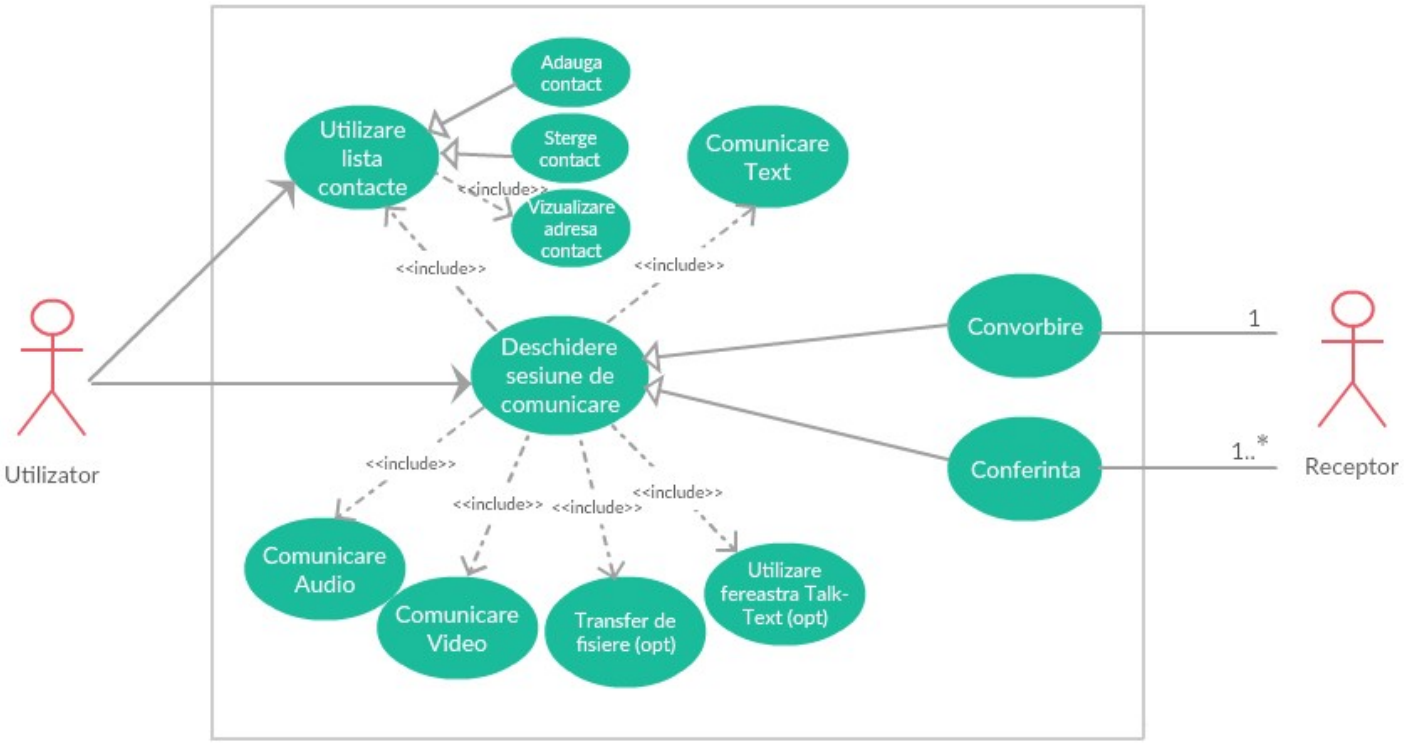
$(\exists ip \in Ip \bullet ((ip, nameContact?) \in data)) \Rightarrow address! = ip$
$(\forall ip \in Ip \bullet (\neg ((ip, nameContact?) \in data))) \Rightarrow address! \in \emptyset$

----------login----------

data ⊆ Ip × Name
myName? : Name; myIp? : Ip
contactList! ⊆ Ip × Name

⌝ ((myIp?, myName?) ∈ data) ⇒ data = data ∪ (myIp?, myName?)
contactList! = data

4. Diagrama UML - Cazuri de utilizare



Descriere:

- Deschidere sesiune de comunicare
    - Utilizatorul va putea porni o sesiune de comunicare in care va putea specifica tipul ei si persoanele participante

- Convorbire

    - Utilizatorul poate opta ca sesiunea de comunicare sa se desfasoare doar cu un singur partener.

- Conferinta
    - Utilizatorul poate opta ca sesiunea de comunicare sa se desfasoare cu 2 sau mai multi parteneri.

- Comunicare Text
    - Utilizatorul poate trimite si primi mesaje in cadrul unei sesiuni de comunicare

- Comunicare Audio
    - Utilizatorul va putea activa/deactiva optiunea de comunicare audio in cadrul unei sesiuni de comunicare

- Comunicare Video
    - Utilizatorul va putea activa/deactiva optiunea de comunicare video in cadrul unei sesiuni de comunicare

- Transfer de fisiere
    - In cadrul acestei facilitati, utilizatorul poate face transfer de fisiere cu receptorii sai in cadrul unei sesiuni de comunicare

- Utilizare fereastra talk text
    - Utilizatorul poate deschide o fereastra text in care sa editeze, in timp ce partenerii sai pot urmari in timp real modificarile .

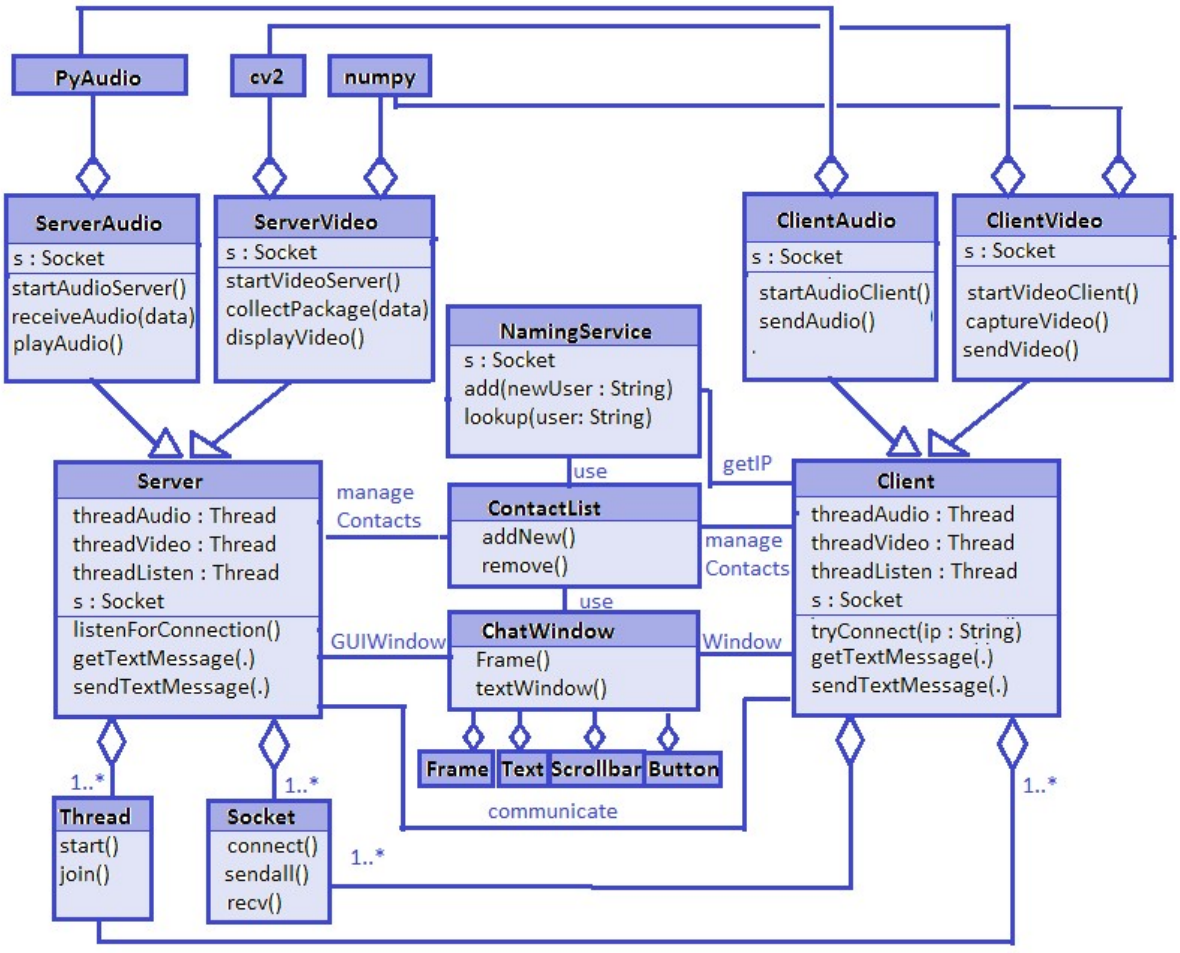
- Utilizare lista contacte
    - Permite vizualizarea contactelor precum si operatiile descrise mai jos

- Adauga contact
    - Permite adaugarea unui contact la lista de contacte

- Stergere contact
    - Permite stergerea unui contact din lista de contacte

- Vizualizare adresa contact
    - Permite vizualizarea informatiilor aferente unui contact

4. Diagrama UML de clase



Descriere:

Aplicatia creeaza o conexiune 'peer to peer' prin intermediul socket-urilor intre client si server. Socket-urile utilizeaza protocolul TCP , care este potrivit pentru transmiterea de date sincronizate. Cele trei tipuri de comunicare se realizeaza in threaduri separate. Asadar,

clasa Socket si clasa Thread sunt importate din librariile Python, socket si respectiv threading, pentru a realiza comunicarea.

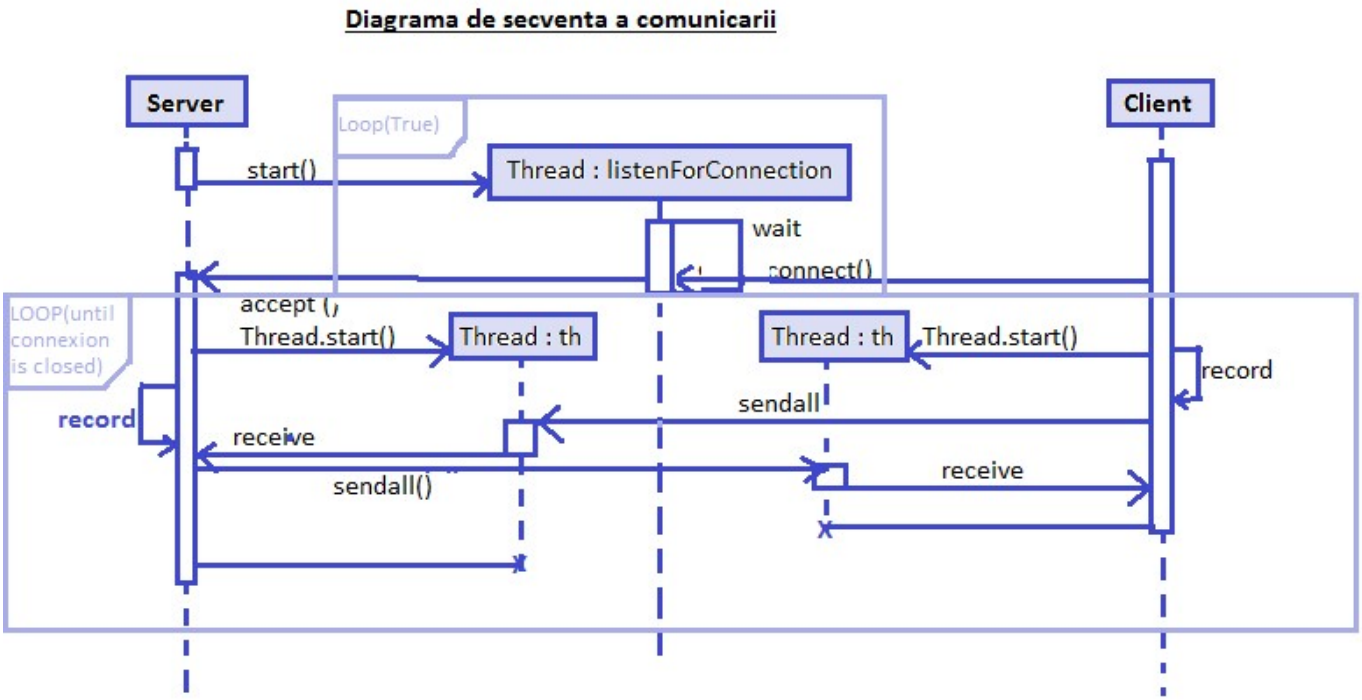
Pentru inregistrarea si difuzarea audio, utilizam clasa PyAudio, din libraria pyaudio.

Pentru accesarea camerei web, inregistrarea si afisarea video, utilizam clasa cv2 din libraria cv2, iar pentru prelucrarea imaginilor, in vederea transmiterii acestora prin socketuri, utilizam clasa numpy.

Clasa ChatWindow implementeaza interfata grafica, iar NamingService si ContactList gestioneaza lista de contacte si comunicarea cu baza de date.

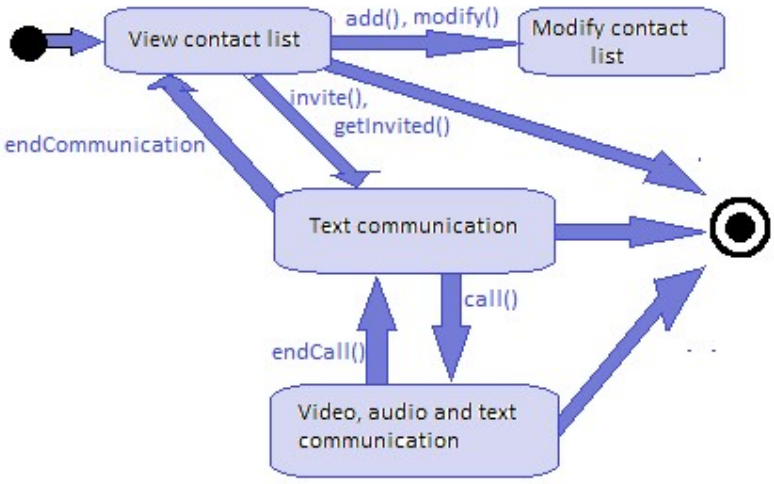
Conexiunile si comunicarea se realizeaza cu ajutorul claselor Server - Client, ServerAudio - ClientAudio, ServerVideo - ClientVideo.

5. Diagrama UML de secvente



6. Diagrama UML de stari

Diagrama de stari ale programului



7. Testarea Aplicatiei