

SwRTc – Proiect

Sistem *software* de comunicare distribuit, pentru conversatii sub forma de mesaje text

1. Specificatii generale

1.1. Cerinte generale

Sa se proiecteze si implementeze in limbajul de programare Java un sistem de comunicare distribuit, suport pentru conversatie textuala.

Proiectul va fi realizat de grupuri de 2-3 studenti.

Minimal, sistemul *software* pentru conversatie textuala (*chat*) va avea urmatoarele caracteristici:

- implementarea in limbajul de programare Java,
- interfata cu utilizatorii grafica (Swing).

Sistemul va putea fi:

- **fie o imbunatatire functionala** (adaugarea unui mod de autentificare, adaugarea de canale private, adaugarea unei interfete la server, etc.) **a sistemului dat ca exemplu** (care are o structura de tip client-server, este bazat pe *socket-uri* TCP si are o interfata grafica minimala) sau
- **fie o alternativa a sistemului dat ca exemplu** (de exemplu, avand o structura descentralizata, sau bazat pe *socket-uri* UDP).

Sunt admise si alte alternative (de exemplu, sistem de management al fisierelor la distanta), **la propunerea studentilor**, **daca sistemul utilizeaza socket-uri**. **Propunerea, discutarea si aprobarea alternativelor vor trebui finalizate pana in saptamana a 12 a semestrului.**

1.2. Continutul proiectului

Proiectul va contine (pentru fiecare student sau grup de studenti care are o tema distincta):

- **dosarul cu documentatia proiectului (minimum 12 pagini)**, incluzand:
 - titlul proiectului si autorii – cu indicarea rolului fiecaruia (prima pagina)
 - principalele caracteristici ale sistemului (client-server sau descentralizat, bazat pe TCP sau UDP, etc.) (pagina a doua)
 - documentarea solutiilor tehnice care nu apar in sistemul dat ca exemplu (mod de autentificare, canale private, interfata la server, etc.) **incluzand** diagrame UML (cateva pagini)
 - descrierea modului de utilizare al sistemului (cateva pagini)
 - **listiurile codurilor sursa** atasate ca **anexa la dosar**
 - **o discheta sau un CD** continand **proiectul in forma electronica** (intr-un plic atasat de dosar), adica:
 - documentatia si
 - codurile sursa.
-

1.3. Modul de sustinere

Sustinerea proiectului presupune:

- prezentarea principalelor caracteristici ale sistemului
- prezentarea rolului fiecarui autor la realizarea sistemului si a documentarii lui
- prezentarea executiei sistemului
- prezentarea solutiilor tehnice specifice
- raspunsuri la intrebari privind detalii ale proiectului (din documentatie si listinguri)
- predarea dosarului cu documentatia proiectului (avand atasata forma electronica)

Sustinerea va avea loc in ultimele 2 saptamani ale semestrului, sau in pre-ziua examenului (nu in ziua examenului!).

2. Exemplu de sistem *software* de comunicatie distribuit, pentru conversatii sub forma de mesaje text

2.1. Caracteristicile sistemului

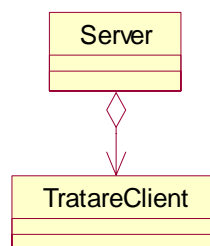
Sistemul *software* pentru conversatie textuala (*chat*) dat ca exemplu are urmatoarele caracteristici:

- implementarea in limbajul de programare Java,
- interfata cu utilizatorii grafica (Swing).
- structura de tip **client-server**,
- bazat pe **socketuri flux** (orientate spre **conexiune**, folosind protocolul **TCP**),

2.2. Modul de lucru

Modul de lucru este urmatorul:

1. Utilizatorul *lanseaza* componenta **client** a sistemului.
2. Clientul se *conecteaza* la server, *ofera* o interfata grafica **utilizatorului**, apoi *trimite* mesaje catre server, *preluate* de la **utilizator** prin interfata grafica.
3. Serverul *accepta* conexiunile si *creaza fire de executie* pentru tratarea clientilor.



4. Fiecare **fir de tratare** a unui client va *primi* mesaje de la **clientul** tratat si va *difuza* aceste mesaje catre toti **clientii**.

5. Fiecare **client** *preia* mesajele de la server si le *prezinta* **utilizatorului** in interfata grafica.

2.3. Detalii de proiectare

Diagrama de secventa a mesajelor schimbate, realizata in faza de analiza OO.

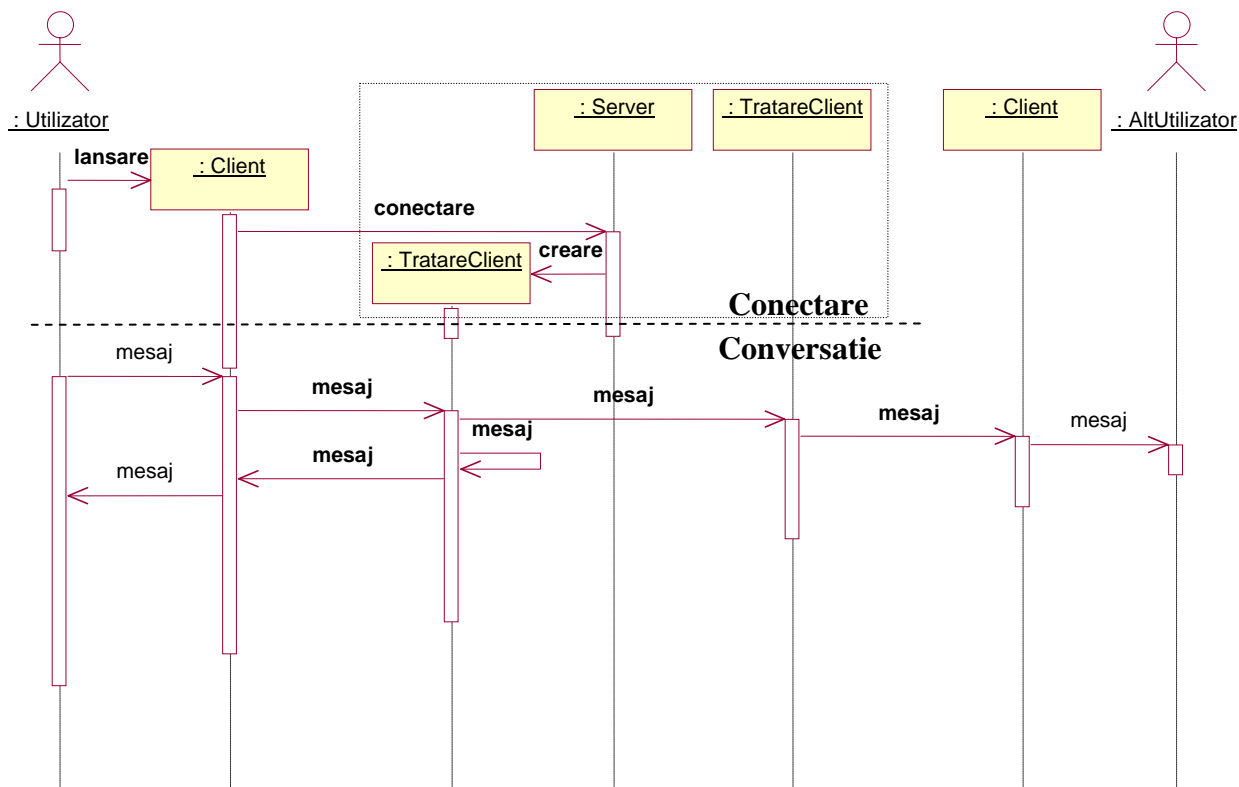


Diagrama de secventa a mesajelor schimbate, realizata in faza de proiectare OO.

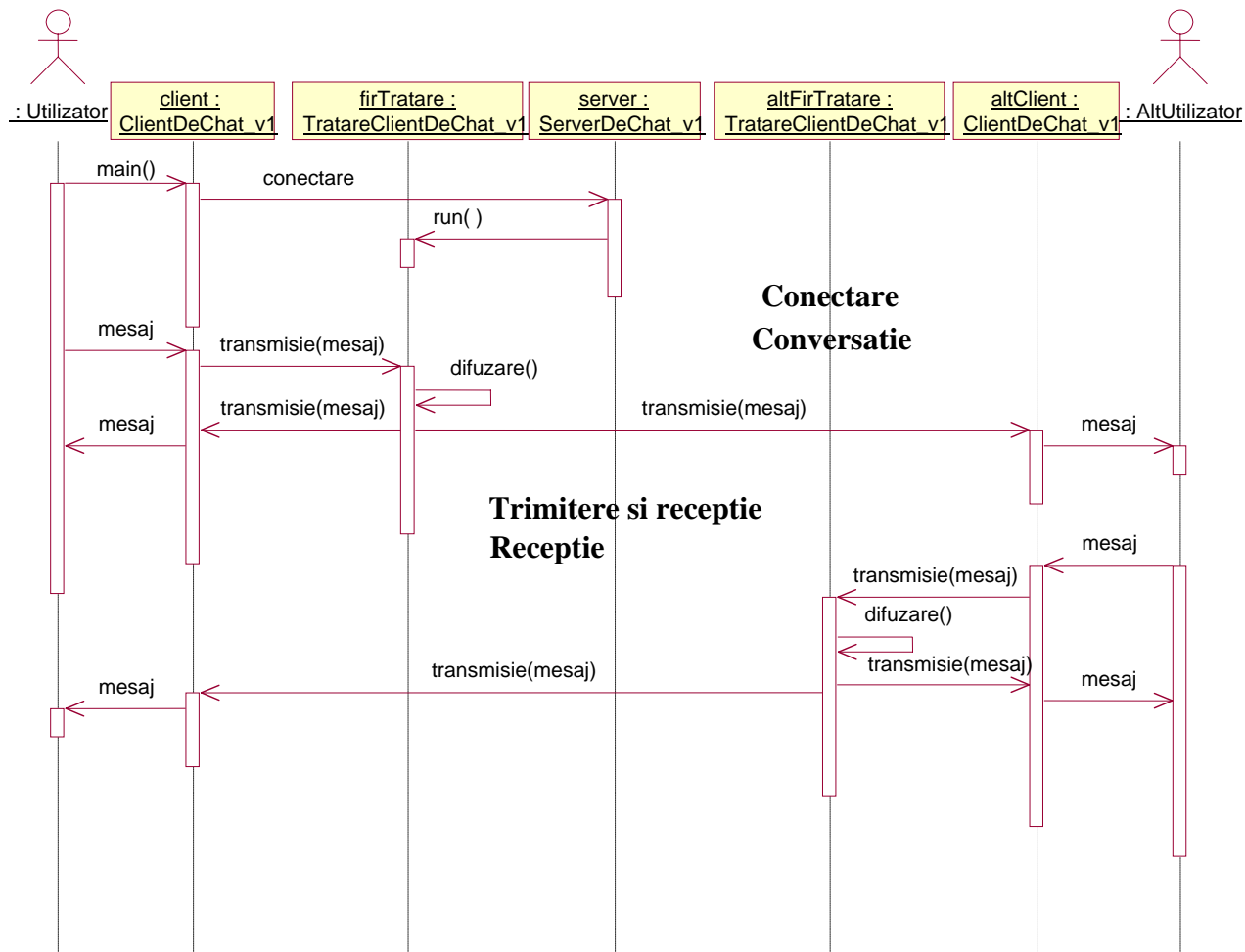


Diagrama de secventa a mesajelor schimbate la client, care detaliaza comportamentul subsistemului la nivel de apeluri de metode Java:

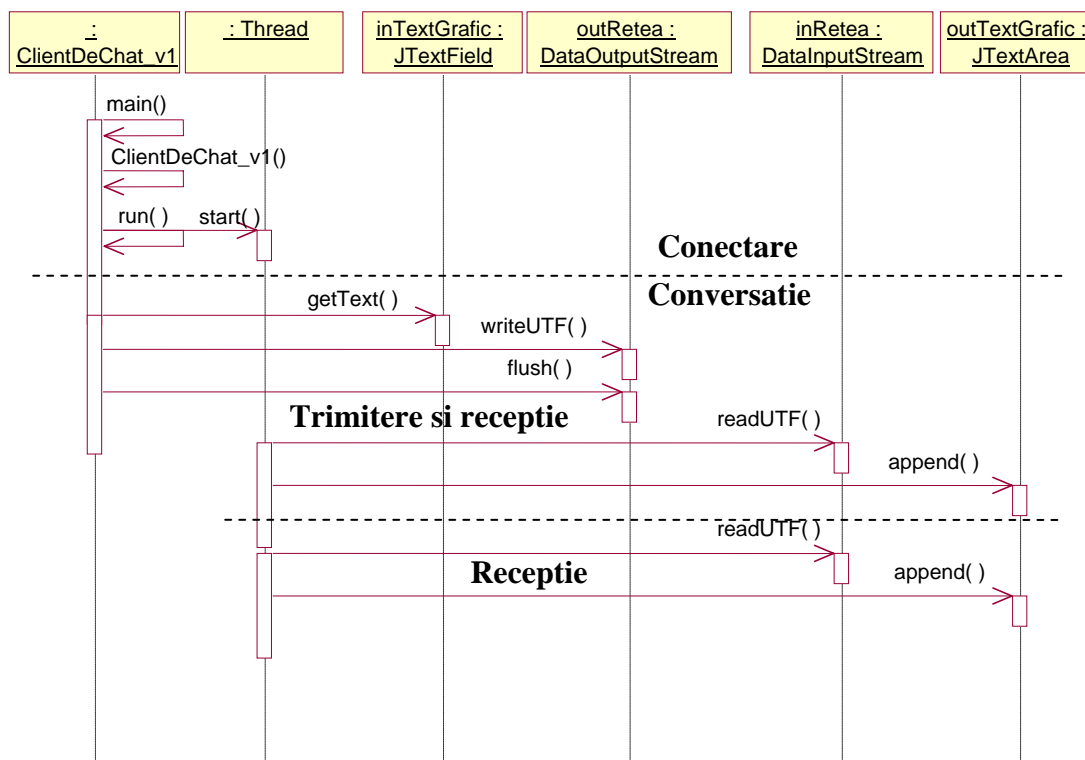
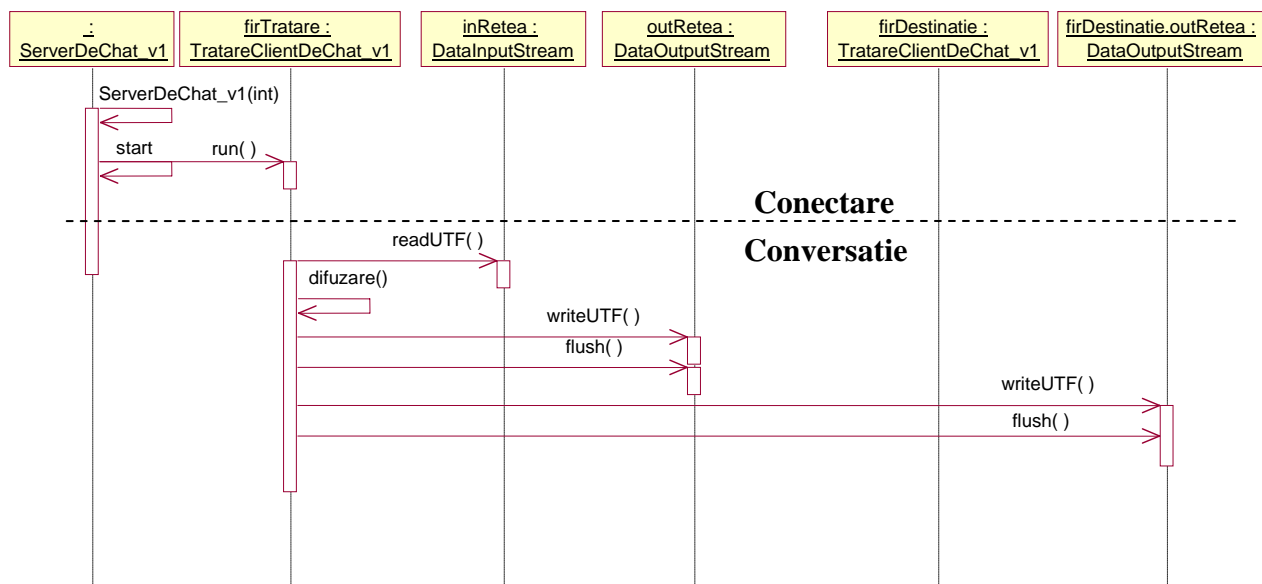


Diagrama de secventa a mesajelor schimbate la server, care detaliaza comportamentul subsistemului la nivel de apeluri de metode Java:



Sistemul utilizeaza urmatoarele biblioteci Java:

- biblioteca standard Java pentru comunicatii la nivel de *socket-uri* (*java.net*)
- biblioteca extensie standard Java pentru interfețe grafice avansate (*javax.swing*) si bibliotecile grafice asociate (*java.awt* si *java.awt.event*)
- biblioteca standard Java pentru fluxuri de intrare-iesire (*java.io*)
- biblioteca standard Java pentru clase utilitare (*java.util*)

Diagrama de clase a subsistemului client este urmatoarea:

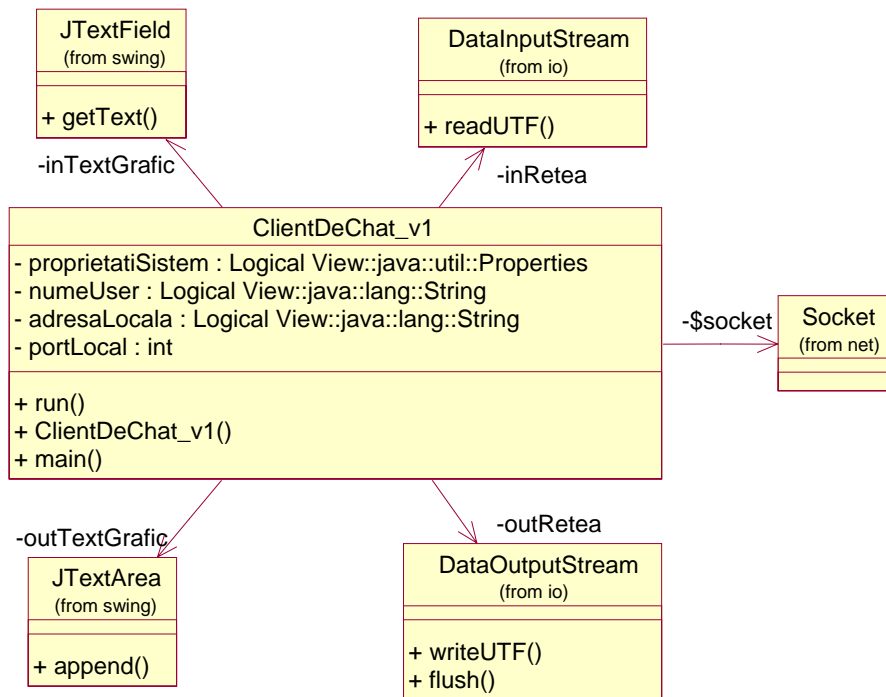
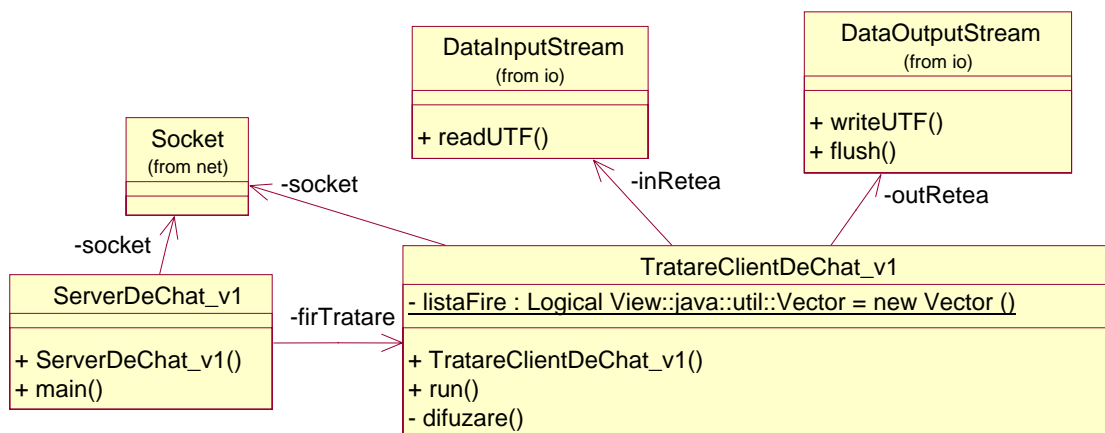
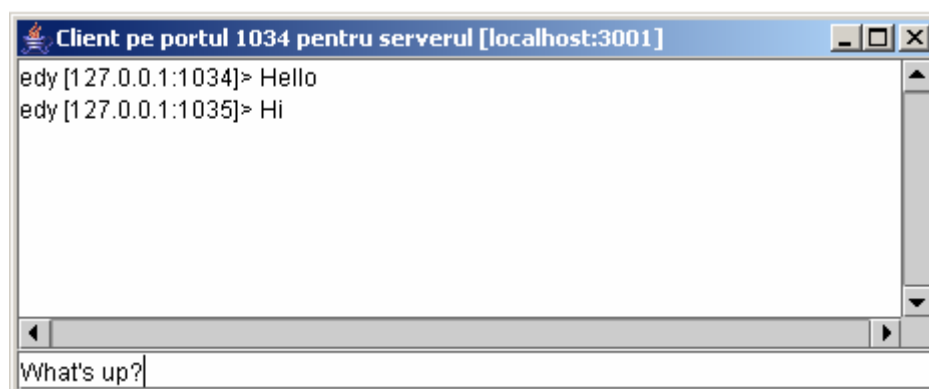


Diagrama de clase a subsistemului server este urmatoarea:



Interfata grafica, minimala, este formata dintr-o intrare de text (sub forma unei linii) pentru editarea mesajelor de trimis, si o zona grafica de text (multilinie si cu posibilitati de defilare) pentru prezentarea mesajelor receptionate de la server.



[Aici gasiti mai multe detalii cu privire la modul in care s-a ajuns de la specificarea cerintelor initiale la aceasta forma a proiectarii si implementarii \(2006_SwRTc_Proiect_Faza1_v01.htm\).](#)

3. Exemplu de imbunatatire a unui sistem *software*

3.1. Noile caracteristici ale sistemului

Presupunem ca s-a decis ca in iteratia urmatoare sa fie realizate urmatoarele imbunatatiri:

- posibilitatea oferita utilizatorilor de a-si alege un nume de utilizator (*nickname*)
- actualizarea automata a barei de defilare a zonei de text pentru ca ultimele mesaje primite sa fie vizibile
- difuzarea de catre server a numelor de utilizator catre toti utilizatorii si informarea de catre client a utilizatorilor despre modificarile in componenta grupului de utilizatori

Raman nerezolvate anumite posibile cerinte ale utilizatorilor, cum ar fi

- posibilitatea crearii mai multor “camere” sau grupuri pe un server
- posibilitatea crearii unor canale de comunicatie private
- etc (lista va fi largita prin discutii la laborator/proiect).

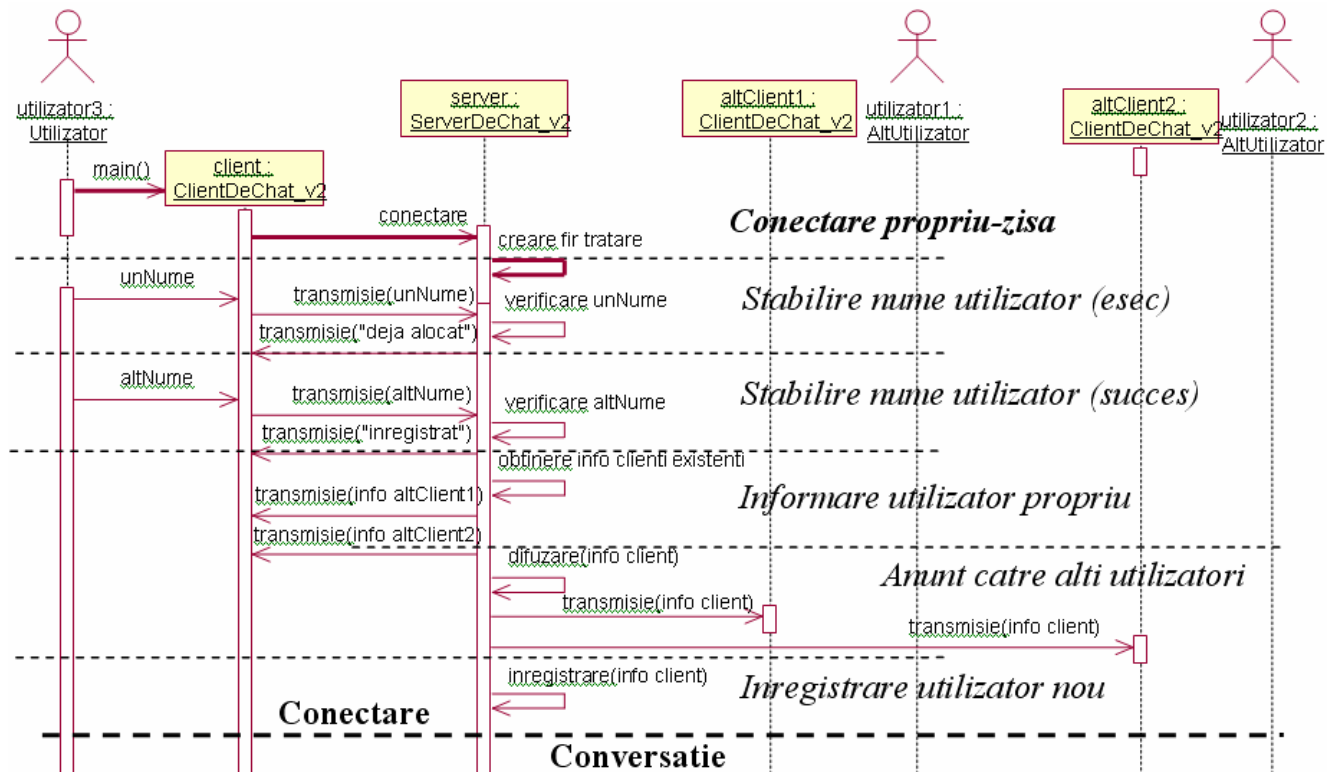
3.2. Modul de lucru al sistemului imbunatatit

Modul de lucru este urmatorul:

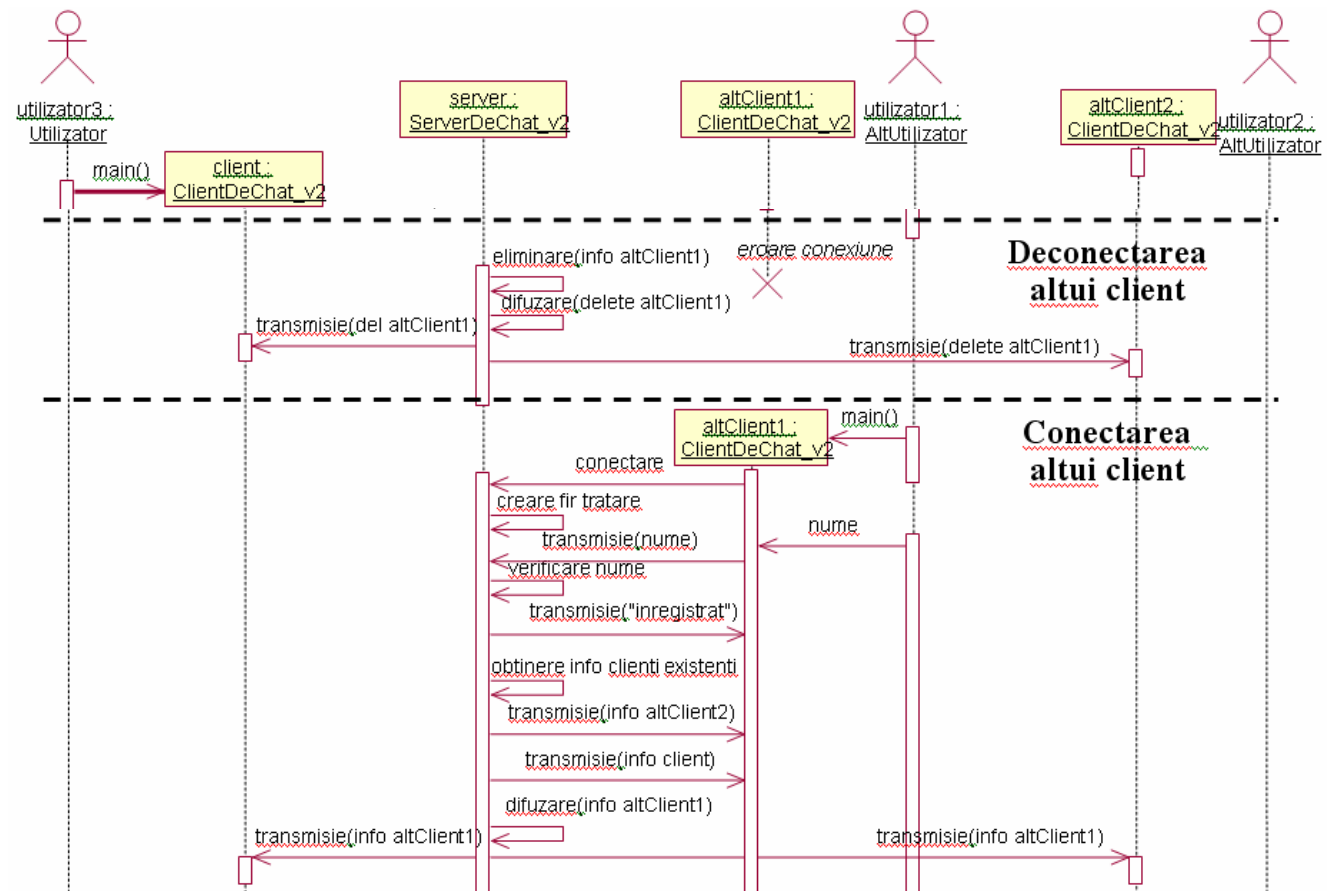
1. **Utilizatorul lanseaza** componenta **client** a sistemului.
 2. **Clientul se conecteaza** la server. **Serverul accepta** conexiunea si **creaza un nou fir de executie** pentru tratarea clientului.
 3. **Clientul obtine** un nume de la **utilizator** (*nickname*) si **il trimite** la server.
 4. **In cazul in care numele de utilizator nu este deja alocat, serverul confirma** clientului acceptarea acestui nume. **Altfel, serverul cere** un nou nume **clientului**, iar **clientul cere** un nou nume **utilizatorului** (si se revine la pasul 3).
 5. **Serverul trimite** noului **utilizator** lista utilizatorilor deja existenti, iar **utilizatorilor deja existenti** numele noului utilizator.
 6. **Noul client ofera** o interfata grafica **utilizatorului**, apoi **trimite** mesaje catre server, **preluate** de la **utilizator** prin interfata grafica.
 7. **Fiecare fir de tratare** a unui client **primeste** mesaje de la **clientul** tratat si **difuzeaza** aceste mesaje catre toti **clientii**.
 8. **Fiecare client preia** mesajele de la server si le **prezinta utilizatorului** in interfata grafica.
 9. **In cazul parasirii comunicatiei de catre un utilizator, serverul trimite** celorlalti utilizatori numele utilizatorului respectiv.
-

3.3. Detalii de proiectare

Varianta de nivel inalt a diagramei de secventa pentru scenariul conectarii unui nou client.



Varianta de nivel inalt a diagramei de secventa pentru scenariul deconectarii unui client.



Varianta de nivel inalt a diagramei de secventa pentru scenariul conversatiei intre clienti.

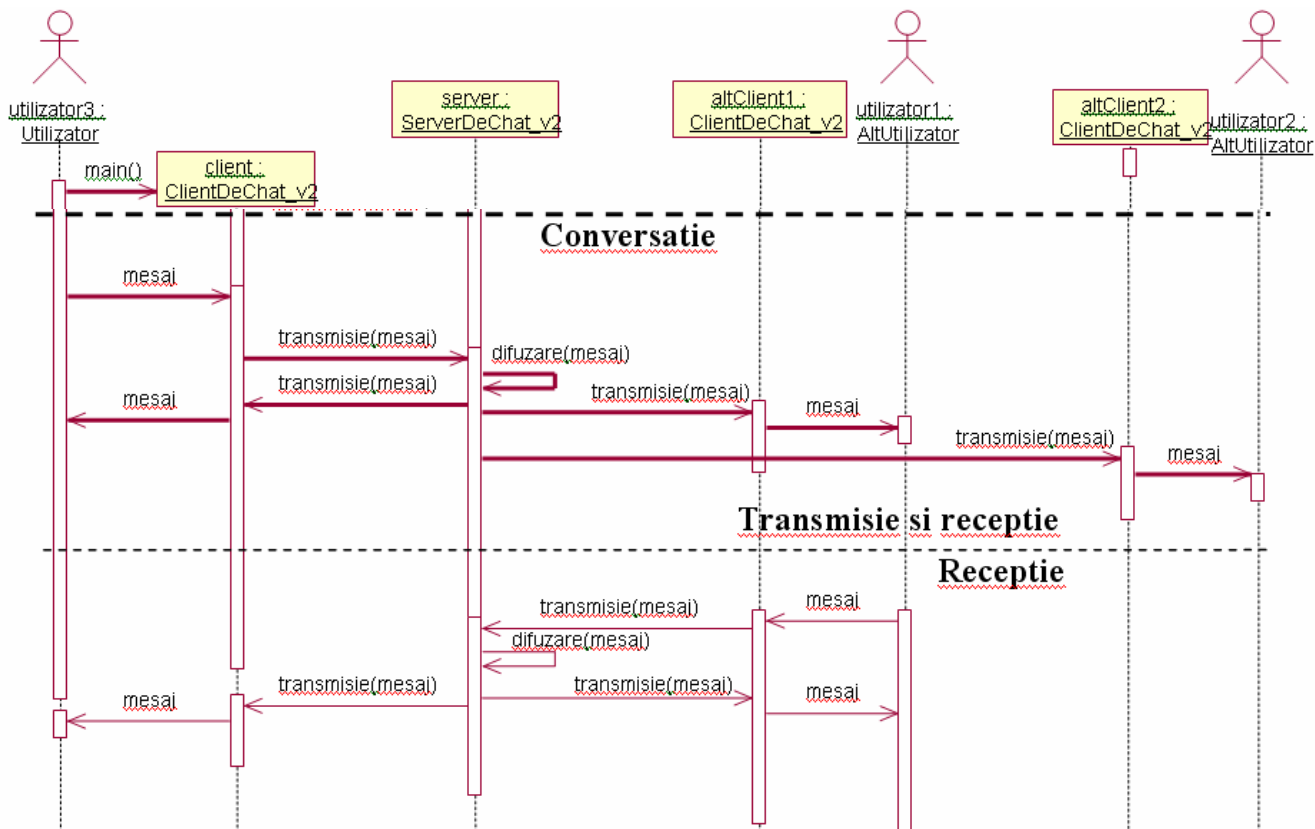


Diagrama de clase actualizata a clientului este urmatoarea:

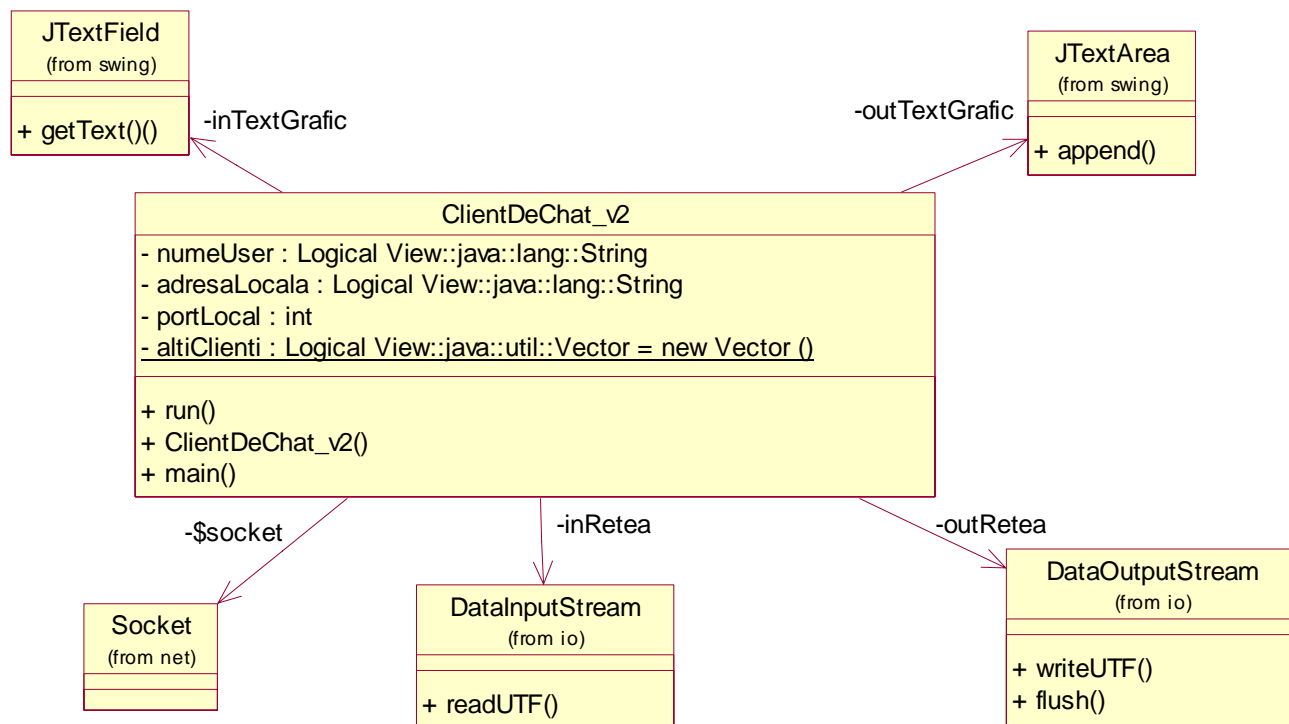
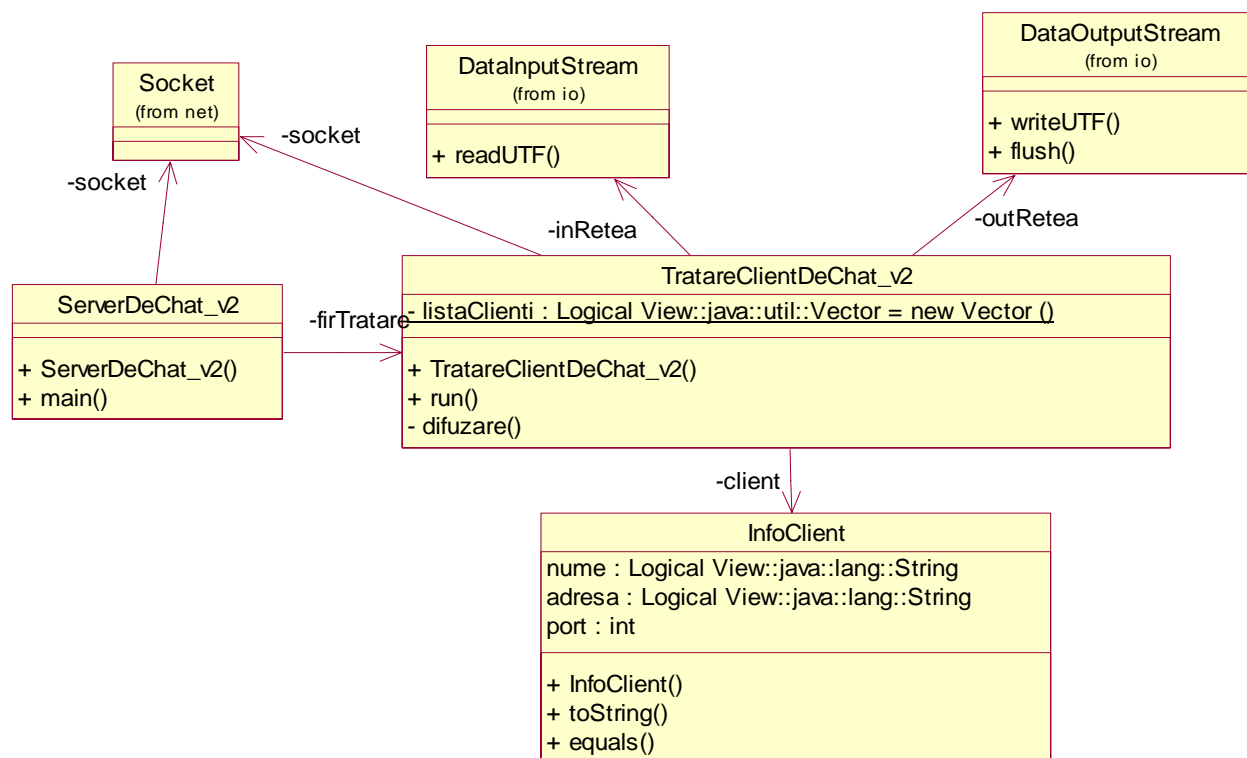
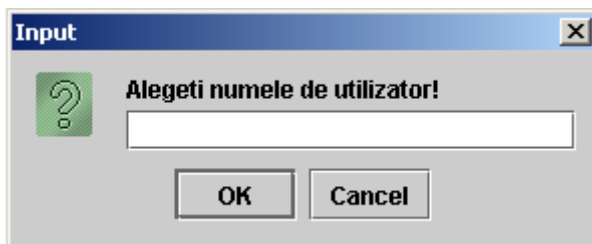


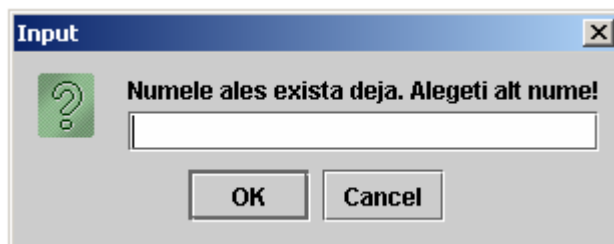
Diagrama de clase actualizata a serverului este urmatoarea:



Interfata grafica prezentata initial unui nou utilizator, pentru alegerea numelui in vederea conectarii, este o fereastra de dialog:



In cazul in care utilizatorul propune un nume care exista deja in lista serverului, acesta respinge inregistrarea iar clientul prezinta urmatoarea fereastra de dialog:

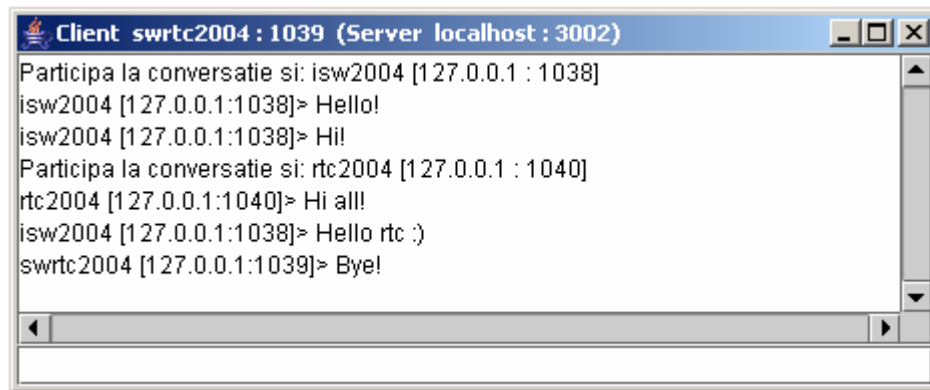


Presupunem urmatorul scenariu:

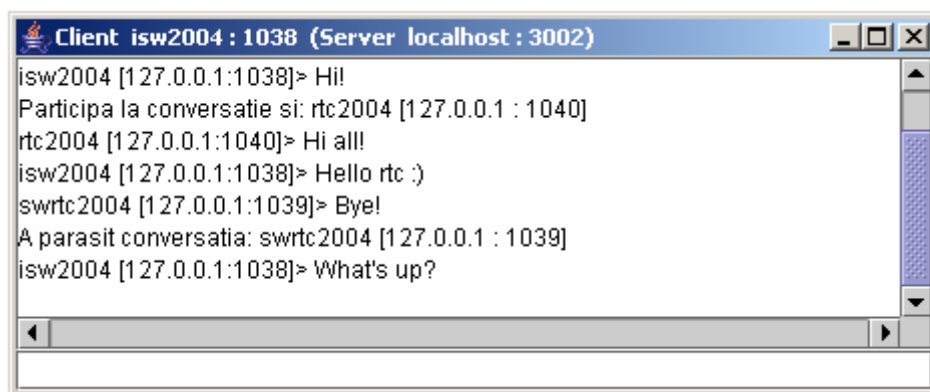
- **primul utilizator isi alege numele isw2004,**
- **al doilea utilizator isi alege numele swrtc2004** (moment in care ambii utilizatori sunt informati, unul de existenta celuilalt),
- **cei doi utilizatori schimba cateva mesaje,**
- **se conecteaza al treilea utilizator,** care isi alege numele **rtc2004** (moment in care primii doi utilizatori sunt informati de existenta celui de-al treilea, iar acesta de existenta celorlalti doi),
- **cei trei utilizatori schimba cateva mesaje,**
- **utilizatorul cu numele swrtc2004 paraseste conversatia** (moment in care ceilalti doi utilizatori sunt informati).

Continutul interfetelor grafice ale celor trei utilizatori pentru acest scenariu va fi urmatorul:

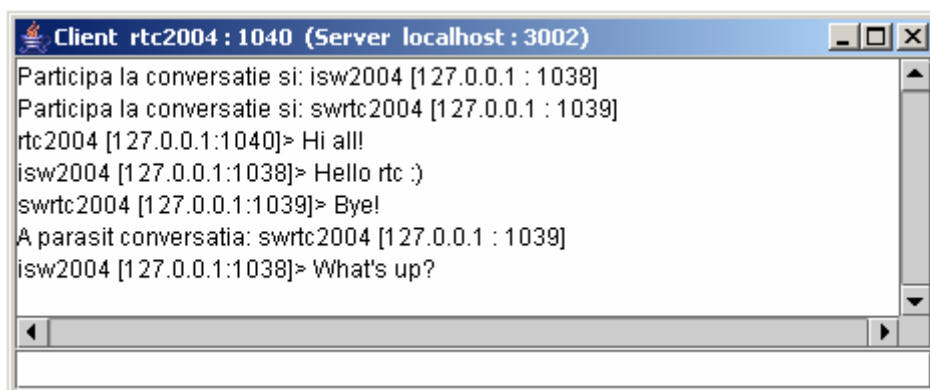
- pentru utilizatorul cu numele **swrtc2004**:



- pentru utilizatorul cu numele **isw2004**:



- pentru utilizatorul cu numele **rtc2004**:



[Aici gasiti mai multe detalii cu privire la modul in care s-a ajuns de la forma initiala la forma imbunatatita a proiectarii si implementarii](#) (2006_SwRTc_Proiect_Faza2_v01.htm).

4. Cum vor fi folosite exemplele de mai sus?

In cazul in care in cadrul proiectului s-a decis sa se realizeze inca o iteratie de imbunatatire functionala a sistemului dat ca exemplu (care sa porneasca de la varianta imbunatatita de mai sus), exemplele anterioare vor fi folosite ca model pentru evolutia proiectarii si implementarii unui sistem de-a lungul unei iteratii.

In cazul in care in cadrul proiectului s-a decis sa se realizeze o alternativa a sistemului dat ca exemplu, varianta imbunatatita de mai sus va fi folosita ca punct de start.