

# Raport Tehnic al Proiectului Virtual Soc

Boldeanu Razvan

## 1 Viziunea Proiectului

O retea sociala lipsita de erori si usor de folosit chiar si de utilizatorii noi. O retea care face comunicarea intre utilizatorii ei foarte intuitiva si care permite o multitudine de interactiuni posibile intre acestia.

Obiective:

- captivarea utilizatorilor prin comenzile simple, usor de retinut, si prin oferirea unei comenzi care afiseaza utilizatorului o lista cu toate comenzile posibile
- posibilitatea crearii a mai multor "chat-rooms". Acestea sunt camere in care sunt conectati 2 sau mai multi utilizatori, unde acestia pot comunica in timp real.
- optiunea postarii de stiri, care sa fie receptionate de o audienta specificata de utilizator
- posibilitatea crearii de postari publice/private, pentru a permite clientilor sa devina mai familiari si mai legati de aceasta aplicatie
- un sistem de imprietenire simplu, dar care permite diversitate utilizatorilor, fara de care niciunul dintre obiectivele de mai sus nu ar fi putut fi atinse

## 2 Tehnologii Aplicate

Am folosit un server TCP, iar pentru procesarea fiecarui client am folosit cate un thread separat. Am facut acest lucru pentru a face o separare clara intre clienti, dar si din cauza faptului ca aplicatia nu trimite input/output constant, cum ar fi apeluri vocale. De aceea am folosit un server TCP, iar nu unul UDP. De asemenea, un server concurent a fost vital pentru aceasta aplicatie, deoarece clientii trebuie sa fie tratati simultan de catre server, pentru o experienta cat mai autentica a unei retele sociale. De asemenea, in fiecare client am folosit doua thread-uri: unul pentru citire si altul pentru scriere.

### 3 Structura Aplicatiei

Aplicatia noastra foloseste o baza de date pentru stocarea marii majoritati de informatii. Utilizatorul foloseste, fara sa stie neaparat, aceasta baza de date la orice fel de comenzi, chiar si la un simplu login.

Cat despre tabelele folosite in baza de date, aplicatia foloseste urmatoarele tabele:

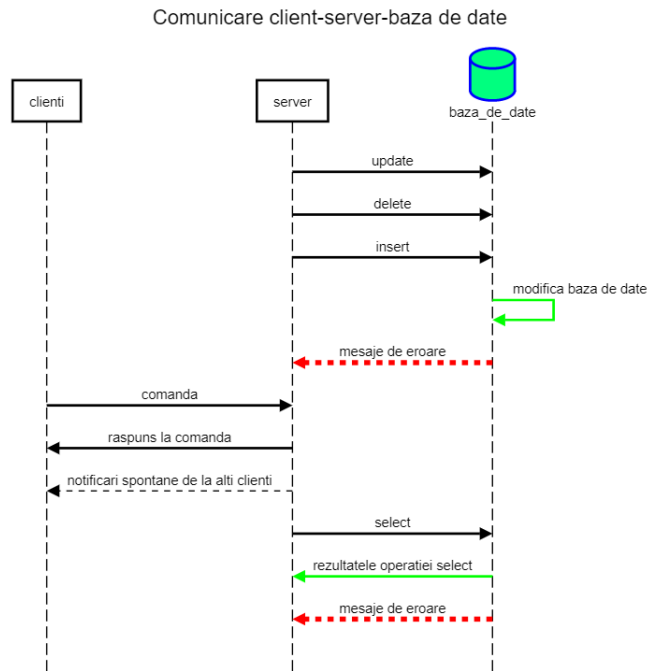
- users : informatii generale legate de fiecare utilizator
- posts : toate informatiile ce privesc postarile utilizatorilor
- friends : relatiile de prietenie dintre utilizatori, precum si tipul de prieteni (apropiati, normali)
- logs : utilizatorii logati la momentul actual
- friend\_requests : cererile de prietenie existente la momentul actual
- chats : informatiile despre utilizatorii care se afla in conversatii la momentul actual + in ce conversatii se afla
- rooms : toate camerele de chat existente (pot avea utilizatori active ce comunica in ele, dar pot avea si participanti offline)
- messages : toate mesajele din camerele de chat sunt stocate aici

Toate informatiile ce tin de utilizatori sunt stocate in baza de date a aplicatiei, iar cele de tin de socket-uri, in aplicatia pripiu-zisa.

De asemenea, merita sa mentionam care sunt comenzile pe care le suporta aplicatia noastra:

- .help (va afisa toate comenzile de mai jos, precum si ce fac)
- register <Nume> <Parola> <Confirmare Parola>
- login <Nume> <Parola>
- logout
- get\_admin <Parola\_ca\_sa\_fii\_admin>
- rm\_admin
- set\_public\_profile (singura diferenta dintre un profil public si unul privat este ca, pentru profilul privat, un utilizator care nu este prieten cu altul nu poate sa ii vada nici macar postarile publice ale acestuia)
- set\_private\_profile

- set\_public\_post <Numele postarii> (in general, postarile private pot fi vizionate doar de catre prieteni, iar cele publice de toata lumea)
- set\_private\_post <Numele postarii>
  
- add\_friend <Persoana>
- rm\_friend <Persoana>
- add\_close\_friend <Persoana>
- rm\_close\_friend <Persoana>
  
- make\_post <Title> <Content> <0/1 (privat/public)>
- view\_my\_posts
- view\_posts <Persoana>
  
- chat\_with <Persoana> <Persoana> ... <Persoana>
- leave (scoate utilizatorul dintr-o conversatie)
  
- make\_news <Title> <Content> <0/1/2 (pt toti / pt prieteni / pt prieteni apropiati)> (stirile apar ca o notificare spontana in terminalele tuturor persoanelor catre care este adresata)
  
- rm\_user <Persoana> (Poate fi folosita doar de utilizatorii administratori)
- exit (scoate clientul din aplicatie)
- exit1 - inchide serverul. Poate produce erori neasteptate, deci nu ar trebui folosita decat de persoanele care cunosc codul foarte bine



## 4 Aspecte de Implementare

Pentru ca utilizatorii sa poata primi si trimite mesaje constant si spontan, partea de client a aplicatiei contine doua thread-uri: thread-ul principal, unde clientul trimite informatiile catre server, si thread-ul secundar, care se ocupa cu citirea constanta de informatii de la server. Am implementat astfel pentru a trata cazul in care clientul primeste o stire/un mesaj de la un alt utilizator (acestea fiind mesaje spontane, care nu vin neaparat ca un raspuns la o comanda a clientului).

Pentru a implementa baza de date, am folosit SQLITE (un tutorial poate fi gasit la link-ul [1] din bibliografie), deoarece este optim pentru acest tip de aplicatie (in cod C), si este usor de implementat.

Probabil cea mai inovativa parte a proiectului este partea de chat-rooms. In sensul ca, in prima versiune a proiectului, in momentul in care un client cerea sa vorbeasca cu alti utilizatori, aplicatia genera un chat-room special pentru ei. Doar utilizatorii din acest room isi vor vedea mesajele unii altora. Aceasta idee oferea o implementare foarte simpla si sigura a comunicarii intre clienti. Iar, de fiecare data cand un client trimitea comenzi catre server, daca acesta facea parte dintr-un chat-room, informatiile trimise de client urmau sa fie tratate ca mesaje (cu exceptia

comenzii leave si exit).

Pe langa aceste beneficii, in ultima versiune a aplicatiei, conversatiile clientilor (mesajele propriu-zise) sunt memorate in baza de date. Asta inseamna ca un utilizator poate avea tot timpul acces la conversatiile din care a facut parte si nu trebuie sa faca parte constant din conversatia propriu-zisa. De asemenea, comunicarea dintre utilizatori a devenit mult mai usoara si intuitiva, tocmai deoarece aplicatia se aseamana mult mai mult cu retelele sociale folosite in viata de zi cu zi.

## 5 Concluzii

O imbunatatire drastica a aplicatiei ar fi folosirea mai multor functii, deoarece, in versiunea curenta, multe parti din cod se repeta de destul de multe ori, ceea ce poate face codul mai dificil de modificat si mai greu de citit (programul are aprox 3000 de linii de cod).

De asemenea, o idee buna ar fi folosirea a mai putine variabile globale si optimizarea bazei de date, in sensul ca unele tabele ar putea fi eliminate, iar informatiile din acestea ar putea fi stocate in alte tabele deja existente. Acest lucru ar putea ajuta considerabil la citirea datelor.

## 6 Referinte Bibliografice

[1]Tutorialspoint : SQLITE - C/C++:

[https://www.tutorialspoint.com/sqlite/sqlite\\_c\\_cpp.htm](https://www.tutorialspoint.com/sqlite/sqlite_c_cpp.htm)