**Offline Messenger**

Spînu Vasilică-Ștefan

Facultatea de Informatica “Alexandru Ioan Cuza” Iasi, General Berthelot 16, 700483 - Iasi

SpinuNVasilicaStefan@gmail.com

1. Introducere

Am ales acest proiect deoarece, inca din momentul in care ne-au fost prezentate in cadrul cursului tehnologiile TCP/UDP pentru crearea de aplicatii tip client-server am fost interest in special de modelul TCP, datorita eficientiei timp pe care acesta o ofera.

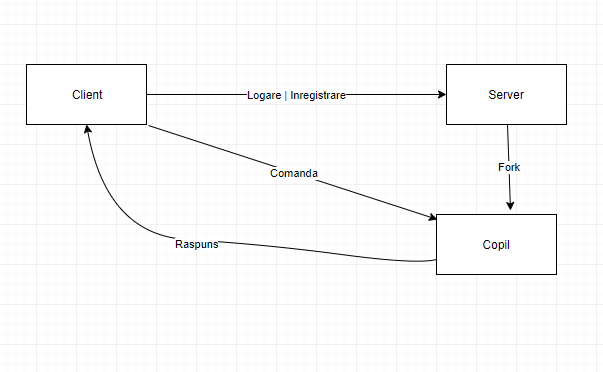
Totodata am fost foarte atras si de topicul aplicatiei, messenger-ul fiind una dintre aplicatiile cruciale ale zilelor noastre am considerat ca, crearea unei “copi” ar fi un bun exercitiu pentru a putea intelege concepte necesare viitoarelor proiecte, deoarece, majoritatea aplicatiilor online tind sa se bazeze foarte mult pe latura sociala a oamenilor.

1. Tehnologii utilizate

In implementarea proiectului voi folosi principiul si protocolul UDP, deoarece, schimbul de mesaje intre utilizatori trebuie sa fie cat mai rapid chiar si cu riscul ca unele mesaje sa nu ajunga la destinatar.Pentru a realiza aceasta comunicare, voi folosi socket-uri.Totodata, o sa implementez si o functie de hashing care v-a stoca alipi la input-ul reprezentat de parola, un salt arbitrar ales, va construi hash-ul si in functie de actiunea utilizatorului (logare/inregistrare) aceasta va stoca in baza de date hash-ul obtinut, sau il va compara cu datele curente pentru a verifica daca parola primta apartine unui cont existent.Numele de utilizator,emailul si hash-ul parolei vor fi stocate intr-o baza de date pentru o mai usoara gestiune a acestora.

1. Arhitectura aplicatiei

Serverul le va oferi clientilor ocazia sa se logheze pe un cont deja existent, sau sa creeze un nou cont. Dupa logare sau inregistrare, serverul notifica clientul in legatura cu mesajele noi primite si necitite, daca acestea exista. Clientul are posibilitatea de a executa una din urmatoarele comenzi: “Afisare useri online”, “Afisarea unei conversatii”,” Stergerea unei conversatii” sau “Trimiterea unui nou mesaj” catre un utilizator specificat. Atunci cand clientul alege comanda care afiseaza userii, serverul ii v-a trimite clientului o lista cu userii ordonati in ordinea logarii fara userul care a executat comanda. Atunci cand clientul alege comanda de afisare a unei conversatii, serverul va trimite tot continutul fisierului de la ultima stergere pana in present. Cand clientul alege stergerea unei conversatii, serverul va goli fisierul. Atunci cand clientul selecteaza trimiterea unui nou mesaj, acesta va fi stocat atat in conversatia clientului cat si in conversatia userului catre care se doreste trimiterea, iar,­ acesta va fi notificat imediat daca este online, sau atunci cand se logheaza.Notificarea v-a contine N



1. Detalii de implementare

In cadrul proiectului voi implementa urmatoarele functii:

char Hash ( pass ) – creaaza hash-ul unei parole

int ACC(gmail nickname, pass) – in cazul in care exista un cont stocat in baza de date cu specificatiile date ca si parametru, functia intoarce 1, altfel, 0

void Conv(s, fis , flag, client, leng) – trimite continutul fisierului la socket-ul clientului

void Onl(s, buff, client, leng) – trimita o lista cu toti userii online in acel moment clientului

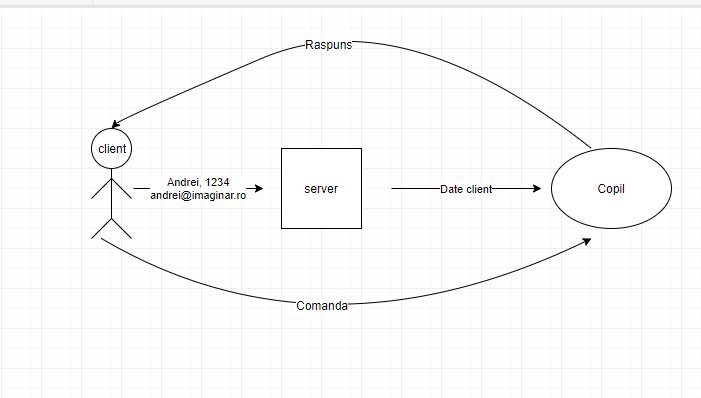
void Del(s, client1, clinet2, clientlog, leng) – sterge conversatia dintre clientul1 si clientul 2, aceasta modificare v-a aparea doar in fisierele clientului 1, daca este realizata cu success, se trimite mesajul: “Stergere efectuata cu succes”. Daca nu exista nici o conversatie intre clientul1 si clientul2 se afiseaza mesajul : “Nu aveti nici o conversatie cu aceasta persoana”.

Int date\_v(nickname, gmail) – aceasta functie verifica daca numele si email-ul folosit pentru crearea unui cont nou nu sunt deja folosite in cadrul altor inregistrari.

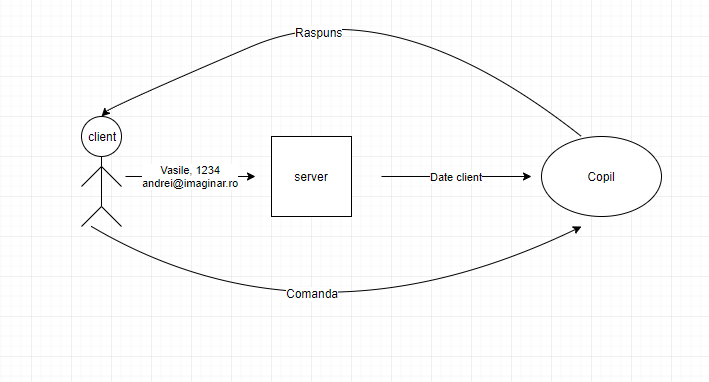
In procesul copil, destinat comunicarii cu clientul, ramanem intr-o bucla infinita care este oprita atunci doar cand, clientul fie introduce un set de date tip (nickname, email, parola) corect, fie alege optiunea de a se inregistra, fie alege sa inchida aplicatia. Atat timp cat ne aflam in aceasta bucla, nici o alta comanda nu este valida.

Sa presupunem ca in baza noastra de date singura inregistrare este: Andrei, 1234 , andrei@imaginar.com

In cadrul acestei diagrame, datele de logare fiind corecte, se foloseste functia de verificare, se creeaza hash-ul, se compara cu inregistrarile actuale din baza de data iar utilizatorul v-a putea folosi aplicatia.



Dar in acest caz, clientul introducand un set de date gresit, v-a ramane in bucla infinita mai sus mentionata, neputand sa foloseasca alta comanda.



1. Concluzii

Din punctul meu de vedere, proiectul ar putea fi imbunatatit semnificativ prin adaugarea unei interfate grafice, pentru a face experienta user-ului mai placuta, adaugarea unei optiuni de a trimite fisiere intre utilizatori si implementarea unei optiuni care sa permita crearea de grupuri in care 3 sau mai multi client sa discute.

1. Bibliografie
2. <http://www.masterraghu.com/subjects/np/introduction/unix_network_programming_v1.3/ch22lev1sec7.html>
3. https://www.quora.com/How-do-I-create-a-concurrent-server-and-concurrent-client-with-a-UDP-socket