

# VirtualSoc - Documentatie

## Tema 2 Retele

Paduraru Dana - B3

Faculty of Computer Science UAIC

### 1 Introducere

VirtualSoc() este o aplicatie client/server ce simuleaza o retea de socializare formata din mai multi utilizatori. Acestia au posibilitatea de a crea postari, ce pot fi publice sau private, si de a vizualiza postarile altor utilizatori. Postarile private vor putea fi vizualizate de utilizatori adaugati in anumite liste de prieteni ale fiecarui user(prieteni, prieteni apropiati, colegi) Sistemul permite un mecanism de comunicare privata intre 2 sau mai multe persoane.

Aplicatia poate fi folosita, de exemplu, de un grup de persoane ce prezinta un interes comun: studentii si profesorii de la o facultate, elevii si profesorii de la o scoala, asociatii de voluntariat, club de sah, club de lectura etc.

### 2 Tehnologii utilizate

Codul este scris in limbajul **C++**. Am folosit **SQLite 3** pentru a crea si a manipula baza de date in care stocam atat userii cat si postarile lor, alegand SQLite fiindca este simplu de utilizat si de integrat in cod.

Am implementat un model server/client **TCP** concurent, pentru a avea posibilitatea de a trata mai multi client in acelasi timp. Am ales modelul TCP deoarece:

- suporta PPP (Point-to-Point Protocol) ⇒ cele doua rute vor putea fi conectate direct, fara a fi nevoie de un host intermediar sau alte dispozitive de retea.

- este orientat conexiune (mereu se va stabili mai intai o conexiune intre server si client)

- ofera siguranta, datele nu se vor pierde si se vor trimite in ordinea corecta.

- ofera mecanisme de control al fluxului si control al congestiei ⇒ Congestia apare cand numarul de pachete transmise prin retea este aproape de capacitatea de manipulare a retelei. Controlul acesteia presupune pastrarea numarului de pachete sub nivelul la care performanta scade.

### 3 Arhitectura aplicatiei

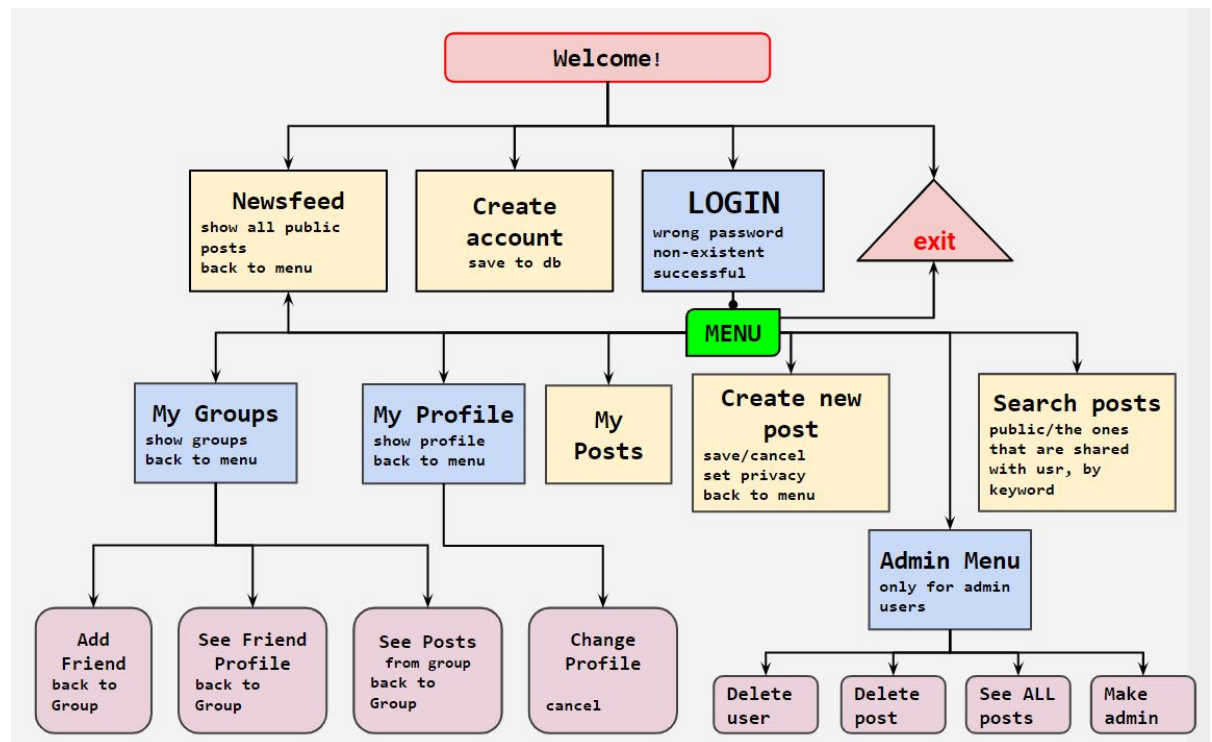


Fig. 1: Diagrama reprezentand functionalitatile aplicatiei

#### Welcome!

Start-ul aplicatiei, unde clientul poate alege intre a se loga, a-si crea cont, a vedea postarile publice fara sa fie autentificat si a iesi din aplicatie. Alegerile primite se vor transmite cu numere: 1 Login, 2 Create account etc.

#### Newsfeed

Lista cu toate postarile publice. Exista posibilitatea de intoarcere catre meniul principal, in acest caz in "Welcome!".

#### Create account

Clientul isi poate crea un cont: este nevoie de nume, prenume si parola, urmand ca celelalte date sa poata fi adaugate ulterior. Se va genera un ID de forma "nume.prenume", iar informatiile despre noul user se vor salva in baza de date. Daca exista cineva cu numele respectiv, id-ul generat va fi de forma "nume.prenume01", "nume.prenume02".

#### Login

Pentru logare se va folosi id-ul generat si parola. Se acopera si cazurile particulare: daca userul si-a gresit parola sau daca nu exista un cont cu acel id.

#### Exit

Iesirea din aplicatie, disponibila si in meniul principal, si in meniul de start.

## Menu

Reprezinta meniul unui user logat. Acesta poate sa-si vada profilul, sa vada grupurile in care se afla, newsfeed-ul, sau sa creeze o noua postare.

## Create new post

Postarea nou creata poate fi salvata, sau anulata (exista optiunea CANCEL-intoarcere catre meniul principal). Daca este salvata, userul va putea seta PRIVACY pentru postarea sa: publica/catre un anumit grup.

## My Profile

Se afiseaza datele despre utilizator, ce pot fi modificate. Exista optiunea de intoarcere catre meniul principal.

## My Groups

Fiecare utilizator are 3 grupuri, adica liste de prieteni:

1. Prieteni
2. Prieteni apropiati
3. Colegi

In My Groups, utilizatorul va putea vizualiza aceste liste de prieteni,profilul unui anumit prieten (cautandu-l dupa ID), postarile care s-au trimis catre acest grup, si de asemenea sa adauge un nou prieten. Adaugarea unui nou prieten se face prin intermediul id-ului sau.

## My Posts

Userul isi poate vedea toate postarile proprii indiferent de "privacy".

## Search Posts

Userul poate cauta o postare dupa un keyword. Postarile care pot aparea fie sunt publice, fie sunt trimise userului de alti useri (prieteni, colegi etc)

## Admin Menu

Userii care sunt administratori ai aplicatiei pot sterge postari (dupa ID), pot sterge useri, pot vedea toate postarile indiferent de "privacy"-ul setat si pot face alti useri admini.

## Baza de date

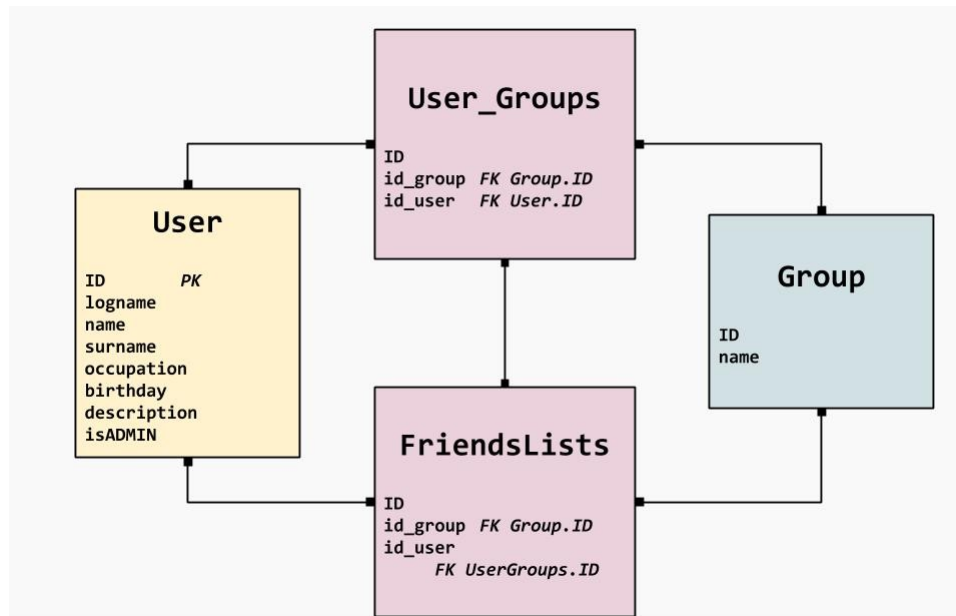


Fig. 2: Baza de date - 1

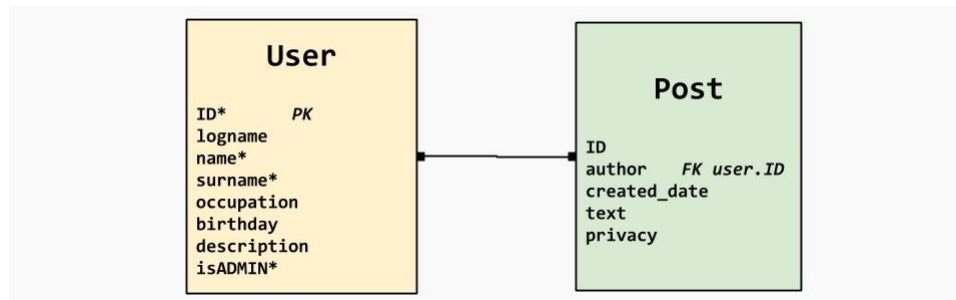


Fig. 3: Baza de date - 2

Vom crea o tabela **User** ce va contine informatiile referitoare la utilizator, fiind modalitatea prin care salvam utilizatorii ce isi creeaza un cont.

Tabela **Group** va contine cele 3 grupuri pe care le detine fiecare utilizator: Friends, Close Friends si Mates. Fiecare va avea un ID unic pe care il vom folosi in urmatoarea tabela creata.

**UserGroups** este utilizata pentru a pastra pentru fiecare utilizator id-uri pentru grupurile sale. Deci, fiecare user va avea pentru fiecare grup al sau cate un ID unic.

In tabela **FriendsLists**, vom pastra listele de prieteni pentru useri. De exemplu, o linie de forma 1 4 2 va insemna ca userul cu ID-ul 4 (din User - greseala in

diagrama, trebuia FK User.ID) apartine grupului cu ID-ul 2 (din UserGroups), unde id-ul 2 este, de exemplu, lista de prieteni apropiati pentru primul user inregistrat, rezultand ca userul cu id-ul 4 a fost adaugat de userul cu id-ul 1 la lista sa de prieteni apropiati.

In tabela **Post** se vor retine datele referitoare la postarile utilizatorilor, autorul (preluat din tabela User), textul postarii, data crearii si nivelul de "privacy" (public/catre un anumit grup). Poate fi folosita pentru a vedea toate postarile care apartin unui singur utilizator.

## 4 Protocol

Protocolul cuprinde mai multe comenzi ce se incadreaza in urmatorul tipar:

- clientului ii este afisat un mesaj, ce va specifica optiunile valabile la momentul curent
- clientul trimite catre server optiunea aleasa, adica trimite un integer sau un text.
- serverul primeste de la client integer-ul trimis si il recunoaste drept una dintre optiunile posibile (1 Login, 2 Create account etc) sau salveaza textul (eventual in baza de date) pentru a-l prelucra.
- serverul va executa optiunea aleasa de client

### MENIU DE START

Clientului ii este afisat un meniu ce va contine optiuni de la 1 la 5. Se asteapta un mesaj din partea clientului de tipul integer, mai precis un numar de la 1 la 5. Daca nu se primeste un numar de la 1 la 5 se va afisa "Please enter one of the options above" si se va astepta din nou un mesaj.

**Funcția care verifica daca un mesaj se incadreaza in optiunile valabile este utilizata de fiecare data cand se asteapta o optiune aleasa de client.**

Optiunea (1) - **LOGIN**: Clientul va scrie id-ul de logare(char) (de forma nume.prenume) si parola (char), care vor fi trimise la server. Serverul va verifica daca contul exista si daca parola este corecta si va trimite un mesaj corespunzator inapoi: ID doesn't exist in the database/Password incorrect/Login successful. Daca logarea s-a facut cu succes, clientului i se va afisa urmatorul meniul.

**Clientul va comunica optiunile alese serverului prin numere, si va trimite mesaje de tip char atunci cand este nevoie sa trimita un ID, un text, etc.**

Optiunea (2) - **SEE PUBLIC POSTS**: Clientului i se vor afisa cele mai recente 50 de postari publice (un char mare creat preluand postarile din baza de date) pe care le poate vedea chiar daca nu este autentificat.

Optiunea(3) - **CREATE CONT**: Clientul va trimite mai multe siruri (char) catre server: nume, prenume, parola. Toate sunt verificate in server, astfel incat numele si prenumele sa fie scris cu initiala majuscula si restul literelor mici, iar parola va trebui sa aiba minim 3 caractere. Serverul va genera un ID de logare pentru client, de forma nume.prenume. Daca mai exista deja un user cu acelasi nume si acelasi prenume, se va genera un id nume.prenume1, nume.prenume2 etc. pentru ca acesta sa fie unic. ID-ul (char) este trimis la client si afisat, iar clientul il va folosi de acum incolo pentru a se loga.

Optiunea (4) - **FIND MORE...**: afiseaza clientului informatii cu privire la aplicatie

Optiunea (5) - **EXIT**: se inchide conexiunea cu clientul, se afiseaza un mesaj: "You have exited the application."

**Activitatile clientului sunt afisate in server. De exemplu: "Client 6 has created an account."**

#### **Dupa logare**

(1) **CREATE NEW POST**: Clientul poate scrie (sau copy/paste) o postare de maxim 800 de caractere. Dupa ce este trimisa, i se va selecta "privacy" (integer) pentru a sti carui grup i se adreseaza postarea. Nu in ultimul rand, se va salva sau se va da Cancel la postare.(adica clientul va trimite un integer, 1, respectiv 2 catre server). Clientului i se va afisa un mesaj corespunzator: "Your post has been saved", "No changes were made".

(2) **NEWSFEED**: Clientul va vedea cele mai recente 50 de postari publice.

**Afisand o postare, afisam: id-ul postarii din baza de date (pentru a putea fi gasita mai tarziu de catre administratori), data crearii, id-ul userului care a creat-o (nume.prenume), textul postarii. Nu se afiseaza privacy.**

(3) **MY PROFILE**: Clientului i se va afisa profilul sau, preluand informatiile din baza de date (nume, prenume) folosind id-ul cu care s-a logat. Pentru a personaliza profilul, au fost adaugate field-uri precum ocupatia, ziua de nastere, descriere. Clientul are posibilitatea de a-si modifica aceste field-uri folosind prima optiune (1) care apare, Change Profile. Va fi intrebat ce camp doreste sa modifice, si va trimite un numar pentru campul respectiv (de exemplu, 3 - Birthday). Clientul va scrie noua informatie pentru campul respectiv si va salva modificarile, sau le va anula (acelasi mecanism ca mai sus de Save/Cancel). Va primi un mesaj de feedback in functie de optiunea aleasa.

(4) **My Posts**: Daca alege aceasta optiune, clientului i se vor afisa cele mai recente 50 de postari (maxim) pe care le-a creat, indiferent de privacy.

(5) **MY GROUPS**: Mai intai, clientul va alege ce grup doreste sa vizualizeze (friends, close friends, mates, back to menu) (va trimite un integer). De exemplu, daca clientul alege (1) Friends, serverul va trimite clientului si vor fi afisate postarile create de alti utilizatori ce au adaugat clientul in lista lor de

"Friends". Tot aici, se alege dintre back to menu si optiunea See Friends. Daca se alege a doua optiune, va aparea o lista cu userii pe care clientul i-a adaugat ca prieteni si de asemenea mai mult optiuni ce vor fi din nou alese trimitand integeri. Prima optiune este "Add Friend", unde clientul poate trimite un ID catre server (nume.prenume) si sa adauge un nou prieten in lista sa. Este verificat in server daca ID-ul exista si daca este deja adaugat in lista respectiva. De asemenea, o alta optiune este "See Friend's Profile", unde este din nou asteptat un ID, ce va fi trimis la server si vor fi preluate datele profilului acelui user, in aceeași maniera ca la My Profile, doar ca aici nu va aparea optiunea de "Change Profile".

**Este de retinut faptul ca exista mereu o optiune de intoarcere la meniul principal.**

**SEARCH POST** Clientul va trimite catre servere un keyword (char) dupa care serverul va cauta in toate postarile publice, dar si in postarile care ii sunt vizibile clientului (de exemplu, o postare cu privacy "Mates" facuta de un user care a adaugat clientul in lista de "Mates"). Se vor afisa toate postarile ce contin keyword-ul respectiv.

**EXIT** Se inchide conexiunea cu clientul, se afiseaza mesajul : "You have exited the application".

**APP MANAGEMENT** Este o optiune care apare doar pentru userii ce au in baza de date field-ul "isAdmin" cu valoarea 1, deci cei care sunt administratorii aplicatiei. Vor avea la dispozitie 5 optiuni: prima, "See most recent 50 posts" va afisa cele mai recente 50 de postari indiferent de privacy-ul setat. A doua optiune "Delete post" va putea fi folosita pentru a sterge o postare ce nu se incadreaza in politicile aplicatiei. Clientul va trimite id-ul postarii pe care doreste sa o stearga, serverul o va cauta si postarea va fi din nou afisata pentru a se asigura ca este cea dorita. Clientul va salva modificarile si postarea va fi stearsa sau va anula modificarile si nimic nu va fi modificat.(Save/Cancel, se primeste mesaj de feedback). A treia optiune "Delete user" functioneaza in mod asemanator, doar ca de data aceasta clientul va trimite un char cu id-ul de logare al userului ce trebuie sters, si din nou, va salva sau va anula modificarile. A patra optiune "Make Admin" se aseamana cu a treia, dar userul nu va fi sters, ci field-ul in baza de date "isAdmin" pentru acest user va deveni 1. Ultima optiune este cea de Back to Menu.

Sunt atasate fisierele cpp pentru server si client si makefile-ul, iar mai jos sunt niste snippete de cod pentru trimiterea/primirea mesajelor:

*serverconcurrent.cpp*

```
// S-a realizat conexiunea, se astepta mesajul
bzero (m, 100);
printf ("[server]Asteptam mesajul...\n");
fflush (stdout);
```

```

        // Citirea mesajului
        if (read (client, m, 100) <= 0)
        {
            perror ("[server]ERR la read() de la client.\n");
            close (client);          // inchidem conexiunea
            continue;                // continuam sa ascultam
        }
        printf ("[server]RECEIVED: %s\n", m);

        // Pregatim mesajul de raspuns pentru client
        bzero(m_toc1,100);
        strcat(m_toc1,"Serverul va executa comanda 1, functia de login.\n ");
        printf("[server]SENT: %s\n",m_toc1);

        // Returnam mesajul clientului
        if (write (client, m_toc1, 100) <= 0)
        {
            perror ("[server]ERR la write() catre client.\n");
            continue;
        }
        close (client);

```

*client.cpp*

```

// Read message
bzero (m_tosv, 100);
printf ("[client>Welcome to VirtualSoc! Choose one of the following:\n 1 Login\n 2 See al
fflush (stdout);
read (0, m_tosv, 100); // CITESTE DE LA TASTATURA

// SEND TO server
if (write (sd, m_tosv, 100) <= 0)
{
    perror ("[client]ERR write() spre server.\n");
    return errno;
}

// Citirea raspunsului dat de server
    (apel blocant pina cind serverul raspunde)
if (read (sd, m_tosv, 100) < 0)
{
    perror ("[client]ERR read() de la server.\n");
    return errno;
}

```



```
printf ("[client]RECEIVED: %s\n", m_tosv);  
close (sd);
```

```
dana@dana-Ubuntu18:~/pro$ ./server.exe  
[server]Asteptam la portul 2025...  
[server]Asteptam la portul 2025...  
[server]Asteptam mesajul...  
[server]RECEIVED: 1  
  
[server]SENT: Serverul va executa comanda 1, functia de login.
```

Fig. 4: Server

```
dana@dana-Ubuntu18:~/pro$ ./client.exe 127.0.0.1 2025  
[client>Welcome to VirtualSoc! Choose one of the following:  
1 Login  
2 See all public posts  
3 Create new account  
4 Find out more about the application  
5 Exit  
1  
[client]RECEIVED: Serverul va executa comanda 1, functia de login.  
dana@dana-Ubuntu18:~/pro$ █
```

Fig. 5: Client

## SCENARIU DE UTILIZARE

Daca serverul nu este pornit, clientului i se va refuza conexiunea.

Daca clientul inchide brusc conexiunea, server-ul nu va fi afectat si nici datele stocate in baza de date nu vor fi alterate.

Daca serverul inchide brusc conexiunea din cauza unei probleme, se va inchide conexiunea cu clientul.

## 5 Concluzii. Imbunatatiri posibile

Serverul poate fi utilizat pentru un grup de persoane cu un interes comun, ce doresc sa comunice intr-un loc sigur si printr-un mod simplu. Daca serverul nu este foarte solicitat, varianta traditionala de server concurent (un fork() per client) este utilizabila. Exista si alte variante mai eficiente din punct de vedere al timpului de a crea serverul concurent (threads, de exemplu).

Functionalitati in plus adaugate:

- search post: cauta in postari dupa un keyword specificat de client
- my posts: clientul poate sa-si vada propriile postari
- change profile: clientul poate sa-si modifice datele din profil

- see friend's profile: clientul poate sa vizualizeze profilul altor utilizatori

Urmatoarele pot reprezenta alte functionalitati posibile pentru aplicatie:

- posibilitatea de a adauga propriile grupuri custom, pe langa cele 3 deja definite,
- "forgot password",
- un sistem de like-uri si/sau comentarii la postari,
- hashtag-uri la postari pentru a urmari anumite subiecte,
- un sistem de trimitere de mesaje private catre anumiți utilizatori,
- upload-ul de fisiere intr-o postare si downloadarea fisierului de utilizatori.
- interfata grafica

## 6 Bibliografie

Wikipedia - Transmission Control Protocol  
Controlul congestiei  
Curs Retele  
Laborator Retele