

Documentatie proiect „Top Music”

Bujor Bogdan-Constantin

Facultatea de Informatica Iasi

1 Introducere

Proiectul are la baza realizarea unei aplicatii de tipul client/server pentru gestionarea unui top muzical. Topul muzical contine mai multe genuri muzicale (de exemplu:rock,clasica,dance, etc.). Fiecare melodie are un nume, o descriere, apartine unuia sau mai multor genuri de muzica si are asociat un link catre videoclipul de pe Youtube. Aplicatia pune la dispozitia utilizatorilor mai multe comenzi. Comenzile sunt: adaugarea unei melodii la top, votarea unei melodii sau melodiilor preferate, afisarea topului general pentru situatia curenta a voturilor sau afisarea topului pentru un anumit tip de muzica pentru situatia curenta a voturilor, adaugarea de comentarii, vizualizarea comentariilor sau a descrierilor. In plus, utilizatorii de tip administrator au in plus urmatoarele comenzi: stergerea unei melodii si restrictionarea votarii pentru un utilizator.

2 Tehnologii utilizate

2.1 Server TCP concurent

Am ales utilizarea acestui protocol de comunicare, deoarece ofera o singuranta mare fata de comunicarea UDP. Avantajele acestui protocol de comunicare sunt:

1. TCP garanteaza ca destinatarul (clientul) va primi pachetele (informatiile) cerute.
2. TCP se asigura ca informatiile nu au erori.
3. Informatiile trimise sunt urmarite si de aceea nu o sa avem pierderi de informatii sau informatii corupte in tranzit.

4. Daca se pierde informatii pe drum, serverul va cere din nou partea pierduta, alta varianta neexistand, fie are toata informatia, fie intrerupe conexiunea.

Pentru acest proiect, TCP este mai bun de folosit pentru ca in momentul in care clientul cere sau transmite o informatie (de exemplu: topul muzical, adaugarea unei melodii) ne dorim ca mesajul sa ajunga intreg si fara erori.

2.2 SQLITE/Baza de date

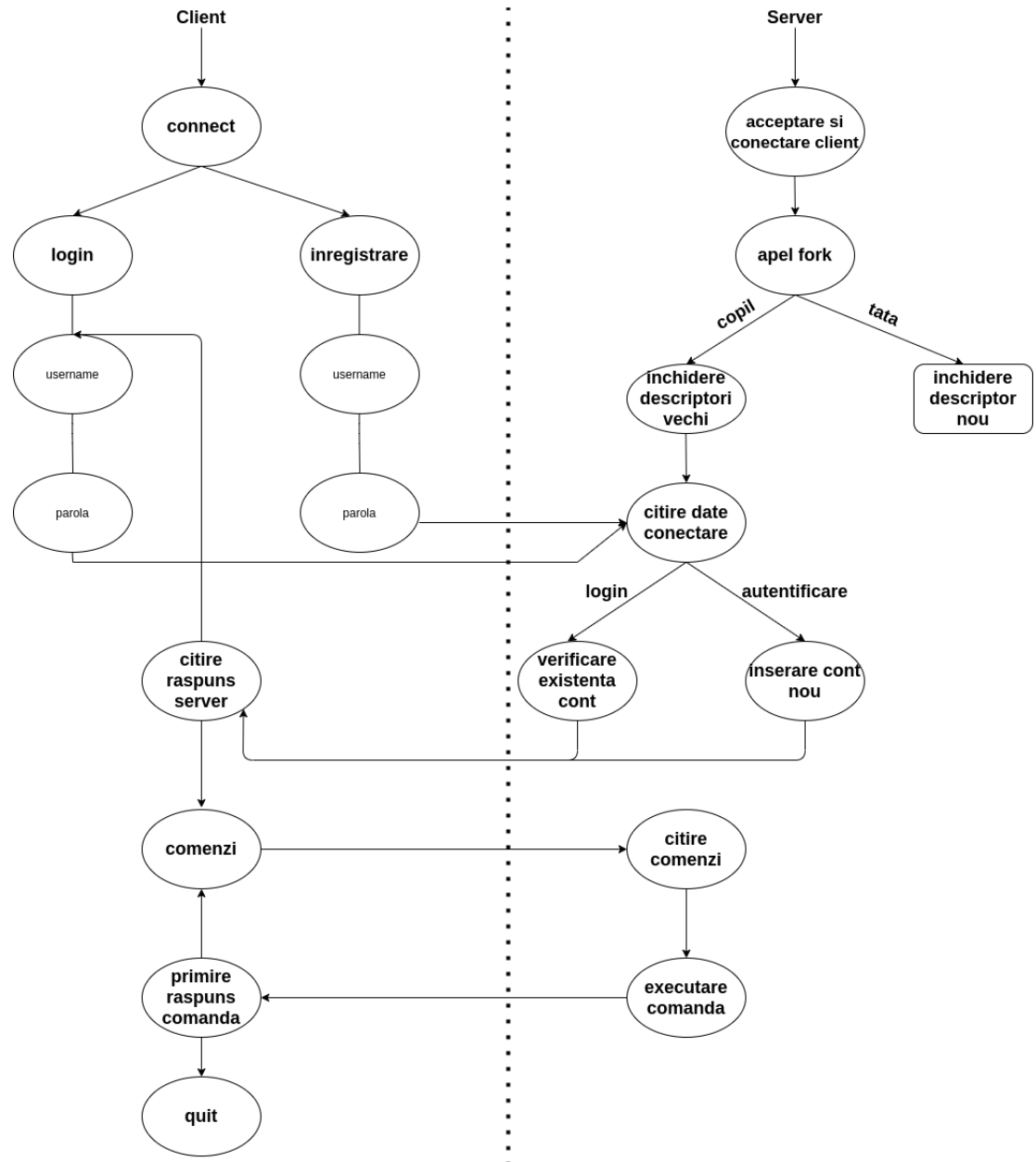
Am ales utilizarea unei baze de date. Avantajele folosirii unei baze de date sunt:

1. Securitatea informatiilor.
2. Acces controlat la baza de date.
2. Stocarea, regasirea, actualizarea datelor mult mai simpla.
3. Control concurent (uneori la fisiere nu putem asigura in mod corect acest lucru).
4. Catalog (dictionare de date).

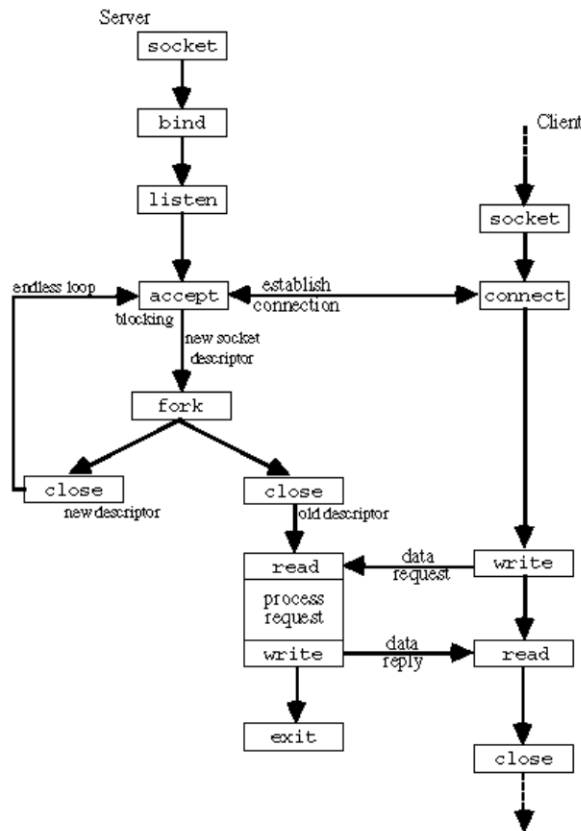
Pentru acest proiect, utilizarea unei baze de date este o varianta mult mai buna decat folosirea mai multor fisiere, deoarece in primul rand, find un protocol de comunicare concurent, vom dori mai multe informatii poate chiar in acelasi timp, iar cum am spus mai sus, baza de date se ocupa de controlul concurent, pe cand cu mai multe fisiere este mai greu sa facem acest lucru. Baza de date are toate informatiile la un loc si ordonate (aranjate), deci ofera un timp mai scurt pentru gasirea sau stocarea informatiilor.

3 Arhitectura aplicatiei

Mai jos este o diagrama care prezinta ideea principala a proiectului.



Ca si canal de comunicare intre server si client am folosit socketuri. Pentru asigurarea comunicarii interproces prin intermediul socketurilor o optiune o constituie primitiva `socketpair()`. Aceasta nu reprezinta altceva decat crearea unei perechi de socketuri conectate ce vor fi utilizate ulterior in comunicare asemeni cu descriptorii capetelor unui pipe intern. Diferenta majora o constituie modul de comunicare ce in acest caz e bidirectional un socket fiind posibil de utilizat atat pentru transmiterea cat si primirea datelor. Concurenta este asigurata de apelul `fork()`. Pentru fiecare client nou conectat, apelul `fork` creeaza un copil (un nou proces) care va servi fiecare client in parte, fara a ii pune sa astepte. Dupa ce clientul a terminat si paraseste aplicatia, procesul copil este ucis. Pentru a asigura si accesul concurent,corect si sigur la informatii am folosit si o baza de data relationala folosind „sqlite3”. Mai jos este o schema pentru serverul TCP concurent folosit.



4 Detalii de implementare

4.1 Cod relevant

```
sqlite> CREATE TABLE USERS (  
...> NAME_USER VARCHAR (100) NOT NULL,  
...> PAROLA VARCHAR(100) NOT NULL,  
...> ADMIN VARCHAR(5) NOT NULL,  
...> VOT BIT NOT NULL );
```

Mai sus am creat in baza de date „proiect.db” tabela cu numele „USERS”. Am facut aceasta tabela, deoarece am considerat ca este nevoie sa avem toti utilizatorii si informatiile lor tinute la un loc pentru a face cautarea mult mai rapida. Fiecare utilizator are un cont cu un username si o parola. In acelasi timp utilizatorul poate avea rol de administrator sau de utilizator obisnuit si poate avea voie sau nu sa voteze.

```
sqlite> CREATE TABLE MUZICA (  
...> NUME_MELODIE VARCHAR(100) NOT NULL,  
...> TIP_MUZICA VARCHAR(100) NOT NULL,  
...> VOTURI INT NOT NULL,  
...> LINK VARCHAR(100) NOT NULL,  
...> DESCRIERE VARCHAR(100) NOT NULL );
```

Mai sus am creat in baza de date „proiect.db” tabela cu numele „MUZICA”. Am facut aceasta tabela, pentru ca am vrut sa am toate melodiile la un loc. Acest lucru imi ofera un management mai bun a tuturor melodiilor, iar cautarea devine mult mai rapida. Fiecare melodie are un nume, unul sau mai multe genuri muzicale, are un anumit numar de voturi, un link catre youtube si o descriere.

```
sqlite> CREATE TABLE COMENTARII (
...> NAME_USER VARCHAR(100) NOT NULL,
...> NUME_MELODIE VARCHAR(100) NOT NULL,
...> COMENTARIU VARCHAR(100) NOT NULL );
```

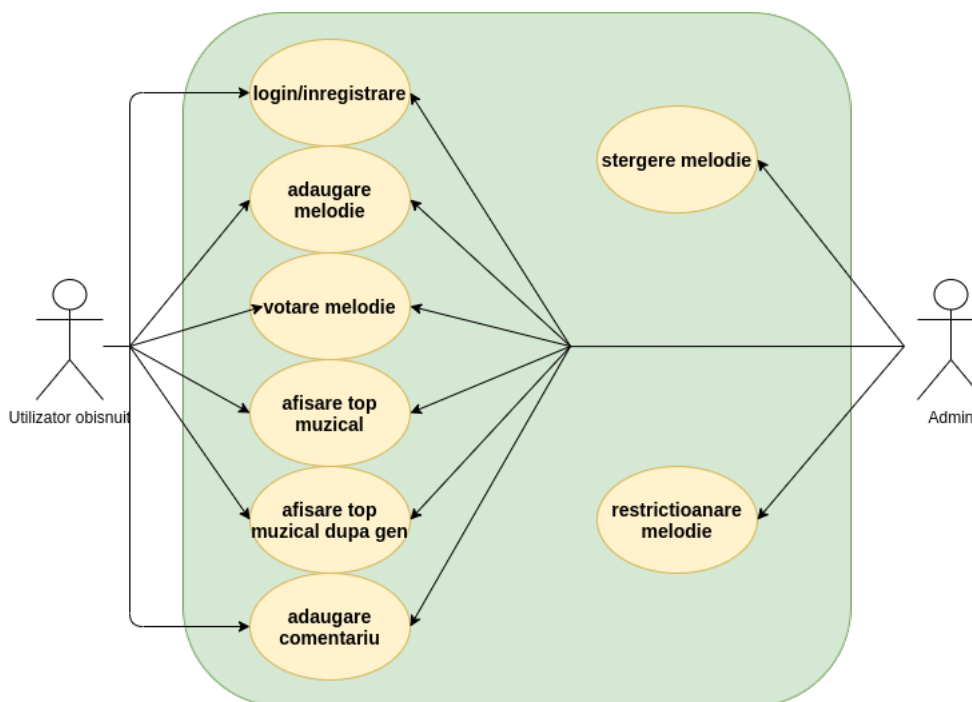
Mai sus am creat in baza de date „proiectrc.db” tabela cu numele „COMENTARII”. Am facut aceasta tabela, deoarece o melodie poate avea mai multe comentarii de la mai multi utilizatori si este mai bine sa le avem pe toate la un loc, inregistrand comentariul pus de un anumit utilizator la o anumita melodie.

```
int callbacklogin(void *data,int argc, char **argv, char **NameCol)
{
    int* x=(int*)data;
    for(int i = 0; i<argc; i++){
        if(argv[i]!=NULL)*x=1;
    }
    return 0;
}

strcpy(sql,"SELECT * FROM USERS WHERE NAME_USER='");
strcat(sql,username);
strcat(sql,"' AND PAROLA='");
strcat(sql,parola);
strcat(sql,"'");
rc=sqlite3_exec(baza_date,sql,callbacklogin,&ok,&errorMsg);
if( rc != SQLITE_OK ){
    fprintf(stderr, "SQL error: %s\n", errorMsg);
    sqlite3_free(errorMsg);
}
```

Mai sus am creat functia callbacklogin (apelata in sqlite3_exec) care verifica daca in baza de date „proiectrc.db” exista username-ul si parola date de client. Daca am gasit datele, la adresa unde se aflat „data” schimb valoarea in 1.

4.2 Scenarii de utilizare



5 Concluzii

Solutia propusa de mine, ar putea fi imbunatatita prin urmatoarele aspecte:

1. Criptarea parolei

Acest lucru va face ca doar clientul sa isi stie parola si sa nu poata fi vazuta de administratorul programului.

2. Grafica

Putem face o interfata grafica pentru client.

6 Bibliografie

1. [https://profs.info.uaic.ro/vcosmin/pagini/resurse_bd/cursuri_bd/Curs%201%20-%20Baze%20de%20date%20\(Introductura\).pdf](https://profs.info.uaic.ro/vcosmin/pagini/resurse_bd/cursuri_bd/Curs%201%20-%20Baze%20de%20date%20(Introductura).pdf)

2. <https://profs.info.uaic.ro/eonica/rc/lab04.html>

3. http://stst.elia.pub.ro/news/RC/Teme_RC_IVA_2011_12/Lecu%20tica%20vidrascu%204
(pg. 5)

4. <https://drive.google.com/file/d/1Zso1PTE6RBATLZ6MzaXfUwFcpEncYL8R/view>