TopMusic

Ailenei Lucian Costin

Facultatea de Informatica, Universitatea “ Alexandru Ioan Cuza “, Iasi

[lucian.ailenei@info.uaic.ro](mailto:lucian.ailenei@info.uaic.ro)

**Abstract.** Lucrarea trateaza problema implementarii aplicatiei de tip client/server TopMusic, oferind detalii despre functionalitatile pe care aceasta le ofera si mecanismele care stau in spatele acestora. In cele ce urmeaza vor fi prezentate paragrafe referitoare la tehnologiile care sunt utilizate, arhitectura aplicatiei, detalii despre implementare, cateva concluzii si bibliografie. Se vor gasi deasemenea secvente relevante de cod si diagrame care duc la o intelegere mai buna a ideii de implementare. In urma parcurgerii lucrarii, utilizatorul va fi capabil sa foloseasca aplicatia intr-un mod cat mai eficient si distractiv.

**Keywords:** TopMusic, client, server, tcp/ip, qt

1. Introducere

TopMusic este o aplicatie in retea care foloseste paradigma client/server si presupune managementul unui top muzical ce contine genuri diverse.

Aplicatia inglobeaza mai multe functionalitati puse la dispozitia utilizatorului, functionalitati la nivel de administrare, vizualizare, actualizare, feedback, etc…

Atunci cand aplicatia este rulata, accesul la interfata este permis doar utilizatorilor care detin un cont. Prin urmare, aplicatia permite inregistrarea de noi conturi.

In momentul in care se face o noua inregistrare, se poate alege dintre doua tipuri de utilizator: utilizator administrator sau utilizator obisnuit. Utilizatorul administrator are in plus permisiuni de administrare fata de utilizatorul obisnuit.

Dupa ce autentificarea este facuta success, utilizatorul obisnuit are posibilitatea de a face urmatoarele operatiuni: poate sa adauge noi melodii la top ( fiecare melodie are un nume, un artist, un gen, o descriere si un link catre un videoclip aflat pe un site), poate sa voteze melodiile care sunt in top, poate adauga comentarii la melodiile din top, poate sa afiseze topul general al melodiilor care vor fi ordonate in functie de numarul de voturi sau poate sa afiseze topul melodiilor doar pentru anumite genuri muzicale.

Fiecare melodie care se afla in top are un nume, un artist, un gen, o descriere si un link catre un videoclip aflat pe un site.

Utilizatorul administrator poate la randul lui sa faca aceleasi operatiuni pe care utilizatorul obisnuit le are la dispozitie, insa, pe langa acestea, el este preocupat cu supravegherea activitatii celorlalti utilizatori. In momentul in care se adauga o noua melodie la top sau un nou gen, aceastea trebuie sa fie aprobate de catre unul din administratori, care are deasemenea posibilitatea de a sterge o melodie sau un gen din top. Acelasi lucru se intampla si pentru voturile si comentariile utilizatorilor. Totodata, administratorii au dreptul sa baneze ceilalti utilizatori in cazul in care nu au un comportament adecvat, adica pot sa le restrictioneze posibilitatea de a adauga melodii, de a vota sau de a lasa comentarii.

1. Tehnologii utilizate
   1. QT

Qt este un sistem inter-platformă de dezvoltare a programelor pentru calculator, folosit atât pentru crearea programelor cu [interfață grafică](http://ro.wikipedia.org/wiki/Interfa%C8%9B%C4%83_grafic%C4%83" \o "Interfață grafică) cât și pentru pentru programe fără interfață grafică, cum sunt serverele. Cele mai cunoscute utilizări ale Qt sunt [KDE](http://ro.wikipedia.org/wiki/KDE), browserul web [Opera](http://ro.wikipedia.org/wiki/Opera_(web_browser)" \o "Opera (web browser)),[Google Earth](http://ro.wikipedia.org/wiki/Google_Earth), [Skype](http://ro.wikipedia.org/wiki/Skype), [Qtopia](http://ro.wikipedia.org/w/index.php?title=Qtopia&action=edit&redlink=1" \o "Qtopia — pagină inexistentă). Qt este produs de firma [norvegiană](http://ro.wikipedia.org/wiki/Norvegia" \o "Norvegia) [Trolltech](http://ro.wikipedia.org/w/index.php?title=Trolltech&action=edit&redlink=1" \o "Trolltech — pagină inexistentă). (Qt)

Comunicarea dintre clienti ( utilizatori ) si server se face in felul urmator: Clientul executa o comanda care trimite o cerere catre server, serverul preia cererea si o proceseaza, efectueaza operatia corespunzatoare si intoarce catre client un raspuns.

Codul ( mecanismul ) care sta in spatele functionarii aplicatiei TopMusic este dezvoltat in limbajul C++ sub Linux, utilizand platforma QT.

* 1. Interfata Socket

Interfata socket reprezinta o facilitate generala, independent de arhitectura hardware, de protocol si de tipul de transmisiune a datelor, pentru facilitarea comunicarii intre procese aflate pe masini diferite, conetate in retea.

Interfata socket suporta diferite protocoale de comunicatie. Pentru comunicatii intre procese de pe acelasi computer sau de pe computere diferite conectate la internet este utilizat protocolul domeniului Internet, folosind TCP/IP ( desigur, sunt acceptate si altele ).

Asadar, un socket poate avea tipuri diferite si poate fi asociat cu unul sau mai multe procese, existand in cadrul unui domeniu de comunicatie. Datele pot fi schimbate numai intre socket-uri apartinand aceluiasi domeniu de comunicatie.

(Buraga & Gabriel, 2001)

* 1. TCP/IP

Comunicarea dintre clienti ( utilizatori ) si server se realizeaza prin intermediul socket-urilor, utilizand protocolul TCP (stream-uri de octeti) din familia de protocoale TCP/IP utilizate pentru Internet. TCP ( Transmission Control Protocol ) este un protocol orientat-conexiune care ofera posibilitatea de a realiza comunicatii full-duplex ( bi-directionale ) sigure.

Fiecare utilizator ( client ) este definit de catre un IP si un PORT TCP.

Se foloseste un server concurent creat prin fork(), adica in momentul in care un client se conecteaza la server, procesul parinte ( serverul ) creeaza un proces copil dedicat care se va ocupa de tratarea cererilor clientului.

Se va folosi un server concurent, iar fiecare client (utilizator) va avea asociat un thread ( fir de executie ) care partajează zona de cod, zona de date şi descriptorii, având propria stivă şi propriile valori ale regiştrilor. Prin folosirea firelor de executie, costurile crearii si managementul unui numar foarte mare de clienti vor fi minime.

* 1. SQLite

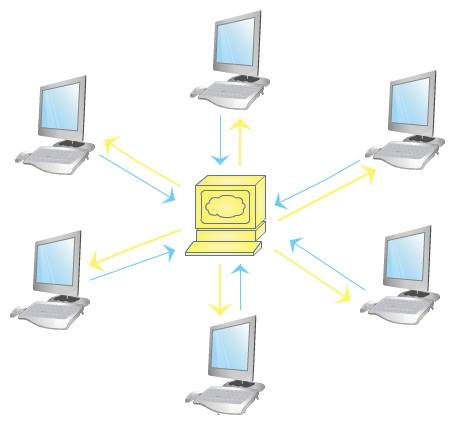
SQLite este o bibliotecă dezvoltata in limbajul de programare C standard care implementează un motor [de baze de date](http://ro.wikipedia.org/wiki/Baz%C4%83_de_date) SQL încapsulat, oferind posibilitatea de a-l introduce în diverse sisteme.

Aceasta biblioteca este folosita pentru stocarea datelor referitoare la melodii, utilizatori, voturi, comentarii, etc…

Baza de date este reactualizata la fiecare operatiune de modificare survenita in cadrul utilizarii aplicatiei.

In momentul in care aplicatia este inchisa, baza de date este salvata, iar atunci cand este repornita, datele sunt reincarcate si prelucrate de catre program in vederea afisarii informatiilor corespunzatoare .

1. Arhitectura aplicatiei
   1. Schema aplicatiei

**Fig.**  

In figura 1 este prezentata schema aplicatiei Top Music, pentru a intelege mai bine principiul pe baza caruia aceasta functioneaza.

Mai multi utilizatori ( clienti ), se pot conecta simultan la server. Acestia utilizeaza functionalitatile pe care aplicatia le ofera si executa niste comezi prin intermediul interfetei consola, facandu-se o cerere de procesare catre server ( sagetile albastre ).

Serverul preia cererile facute de catre clienti, le proceseaza si returneaza un raspuns catre clienti ( sagetile galbene ). Interfata consola, preia raspunsul de la server si se modifica corespunzator.

* 1. Clase – Obiecte si Metode

Functionalitatile aplicatiei Top Music sunt implementate sub limbajul C++, folosindu-se paradigma de programare OOP ( Object Oriented Programming ). In server, este definita o clasa Top, care inglobeaza functionalitati ale aplicatiei ce comunica in mod direct cu baza de date in care sunt stocate datele referitoare la utilizatori, melodii, comentarii, executand interogari care folosesc la afisarea datelor sau la modificarea acestora, in functie de cererea provenita de la client. Practic, aplicatia functioneaza in felul urmator: in client si server avem definite o serie de functii care stabilesc comunicarea intre cele doua instante ( schimbul de mesaje ); in momentul in care un client se conecteaza la aplicatie, acesta apeleaza o functie prin intermediul interfetei consola care trimite un mesaj catre server; serverul, recunoaste tipul de mesaj primit si apeleaza functia corespunzatoare pentru citirea acestuia care contine deasemenea una din metodele clasei Top ce fac legatura cu baza de date, metoda care va fi apelata utilizand anumiti parametri din mesajul provenit de la client; aceasta metoda returneaza mai apoi un raspuns care este trimis catre client de catre functia corespunzatoare din server.

1. Detalii de implementare

La sectiunea 3.2 Clase – Obiecte si metode de la capitolul anterior explicam cum functioneaza practic legatura dintre client si server. Vom vedea in cele ce urmeaza un exemplu concret de implementare a unei astfel de functii, mai exact, functia de inregistrare a unui nou utilizator.

Prima functie asupra careia facem o analiza este functia din client, cea care trimite o cerere de adaugare a unui nou utilizator in baza de date si asteapta un raspuns de confirmare de la server.

void register\_function(){

// variabilele de stocare a datelor introduse de la tastatura

int register\_option;

char username[100];

char password[100];

bool s; //variabila de primire a raspunsului de la server

// mesajele pe care le afisam pentru ca utilizatorul sa stie ce sa introduca de la tastatura

QString choose\_account="Alegeti tipul de utilizator: \n [1] Admin \n [2] Utilizator obisnuit ";

QString enter\_username="Introduceti numele de utilizator: ";

QString enter\_password="Introduceti parola: ";

// citirea de la tastatura a datelor si trimiterea lor catre server

// 1 – trimiterea catre server a tipului de utilizator

qDebug() << qPrintable(choose\_account);

scanf("%d",&register\_option);

if(write(sd,&register\_option,sizeof(int))<=0)

perror("[client]Eroare la write(). \n");

// 2 – trimiterea catre server a numelui de utilizator

qDebug() << qPrintable(enter\_username);

scanf("%s",&username);

QString qusername=username;

int usernamelength=qusername.size();

if(write(sd,&usernamelength,sizeof(int))<=0)

perror("[client]Eroare la trimiterea dimnesiunii username-ului catre server. \n");

if(write(sd,username,usernamelength)<=0)

perror("[client]Eroare la trimiterea username-ului catre server.");

// 3 – trimiterea catre server a parolei

qDebug() << qPrintable(enter\_password);

scanf("%s",&password);

QString qpassword=password;

int passwordlength=qpassword.size();

if(write(sd,&passwordlength,sizeof(int))<=0){

perror("[client]Eroare la trimiterea dimnesiunii parolei catre server. \n");

}

if(write(sd,password,passwordlength)<=0){

perror("[client]Eroare la trimiterea parolei catre server.");

}

// primirea raspunsului de la server

if(read(sd,&s,sizeof(bool))<=0)

perror("[client]Eroare la primirea raspunsului signup. \n");

if(s==0)

qDebug()<<"Inregistrare esuata. Numele de utilizator exista deja in baza de date. \n";

if(s==1)

qDebug()<<"Inregistrare reusita. \n";

}

Cea de-a doua functie este functia din server corespunzatoare primirii cererii de inregistrare a unui nou utilizator, trimisa de catre client

void register\_function(){

// variabilele in care stocam datele primite de la client

//receive user

QString enter\_username="Introduceti numele de utilizator: ";

char stock\_username[100];

int username\_length;

//receive account type

int register\_option;

//receive password

QString enter\_password="Introduceti parola: ";

char stock\_password[100];

int password\_length;

if (read (client, &register\_option, sizeof(int) )<=0){

perror("[server]Eroare la primirea tipului de utilizator. \n");

}

else

printf ("[server]Tipul de utilizator a fost primita.\n");

// primirea numelui de utilizator

bzero(stock\_username,100);

if (read (client, &username\_length, sizeof(int) )<=0){

perror("[server]Eroare la primirea dimensiunii user-ului. \n");

}

else

printf ("[server]Dimensiunea user-ului a fost primita.\n");

if (read (client, &stock\_username, username\_length)<=0){

perror("[server]Eroare la primirea user-ului. \n");

}

else

printf ("[server]User-ul a fost primit.\n");

// primirea parolei

bzero(stock\_password,100);

if (read (client, &password\_length, sizeof(int) )<=0){

perror("[server]Eroare la primirea dimensiunii parolei. \n");

}

else

printf ("[server]Dimensiunea parolei a fost primita.\n");

if (read (client, &stock\_password, password\_length)<=0){

perror("[server]Eroare la primirea parolei. \n");

}

else

printf ("[server]Parola a fost primita.\n");

bool signupreturn;

QString quser=stock\_username;

QString qpass=stock\_password;

// apelarea functiei din clasa Top care prelucreaza baza de date – stocam codul pe // care il returneaza intr-o variabila care este trimisa mai apoi catre client

signupreturn=Top\_Music.sign\_up(register\_option,quser,qpass);

if(write(client,&signupreturn,sizeof(bool))<=0){

perror("Eroare la trimiterea raspunsului signup. \n");

}

}

Cea de-a treia functiei este functia din server care comunica cu baza de date si introduce datele primite de la client, functie apelata in cadrul celei de mai sus. Aceasta functie returneaza un cod, care este trimis catre client prin intermediul functiei anterioare, clientul stiind sa interpreteze raspunsul pe care l-a primit.

// functia are 3 parametri – numele de utilizator, parola si tipul de utilizator

bool Top::sign\_up(int account\_type, QString user\_name, QString password){

// crearea unei interogari pentru baza de date – verificam daca user-ul nu

// exista deja

QSqlQuery check\_new\_user;

check\_new\_user.exec("SELECT username FROM accounts");

while(check\_new\_user.next()){

if(user\_name==check\_new\_user.value(0))

return 0;

}

// introducerea datelor in baza de date

QSqlQuery query;

query.prepare("INSERT INTO accounts (restriction, username, password, type) "

"VALUES (?, ?, ?, ?)");

query.addBindValue(0);

query.addBindValue(user\_name);

query.addBindValue(password);

query.addBindValue(account\_type);

query.exec();

return 1;

}

1. Concluzii

Aplicatia TopMusic inglobeaza toate functionalitatile cerute de catre cerinta problemei.

Utilizarea platofrmei de dezvoltare QT, a limbajului C++ si a bibliotecii SQLite utilizata pentru crearea bazei date au usurat in foarte mare masura implementarea functionalitatilor pe care aplicatia le ofera.

Utilizarea metodei de creare a proceselor concurente prin fork, poate fi un mecanism destul de costisitor in momentul in care se produce o aglomerare de date, de aceea, o utilizare a unui mechanism bazat pe fire de executie ar fi fost mult mai eficienta, deoarece costurile crearii si managementul unui numar foarte mare de clienti ar fi fost mult mai mici.

Deasemenea, crearea unei interfete grafice ( posibila utilizand platforma QT ) ar fi oferit utilizatorilor o interactiune mult mai placuta cu aplicatia.

1. Bibliografie

Buraga, S., & Gabriel, C. (2001). Iasi: POLIROM.

*Qt*. (n.d.). Retrieved from Wikipedia: http://ro.wikipedia.org/wiki/Qt

*SQL*. (n.d.). Retrieved from Wikipedia: http://ro.wikipedia.org/wiki/SQL