

TopMusic (B)

Cozma Ilie-Catalin

5 Ianuarie 2021

1 Introducere

1.1 Enunțul problemei

Să se realizeze o aplicație de tipul client/server pentru managementul a unui top muzical continand genuri diverse. Aplicația va avea următoarele functionalitati: înregistrarea utilizatorilor de mai multe tipuri: obișnuiți, administrator; logarea în sistem. Comenzile nu vor putea fi executate dacă utilizatorul nu este autentificat. Operații ce trebuie implementate: adaugarea unei melodii la top, votarea unei melodii, afișarea topului general curent în funcție de numărul de voturi, afișarea topului pentru un anumit gen de muzica(ex: dance, rock, hip-hop, etc...). O melodie va avea un nume, o descriere, va aparține unuia sau mai multor genuri de muzica, și va avea asociat un link către videoclipul sau pe youtube sau alte site-uri asemănătoare. Utilizatorii autentificați vor putea posta diferite comentarii asociate unei melodii. Administratorul va putea șterge o melodie din top și va putea restricționa opțiunea de vot a unui utilizator.

1.2 Motivatia alegerii temei

Folosesc o aplicatie de acest tip (Spotify) si sunt curios cum ar arata o implementare.

2 Tehnologii utilizate

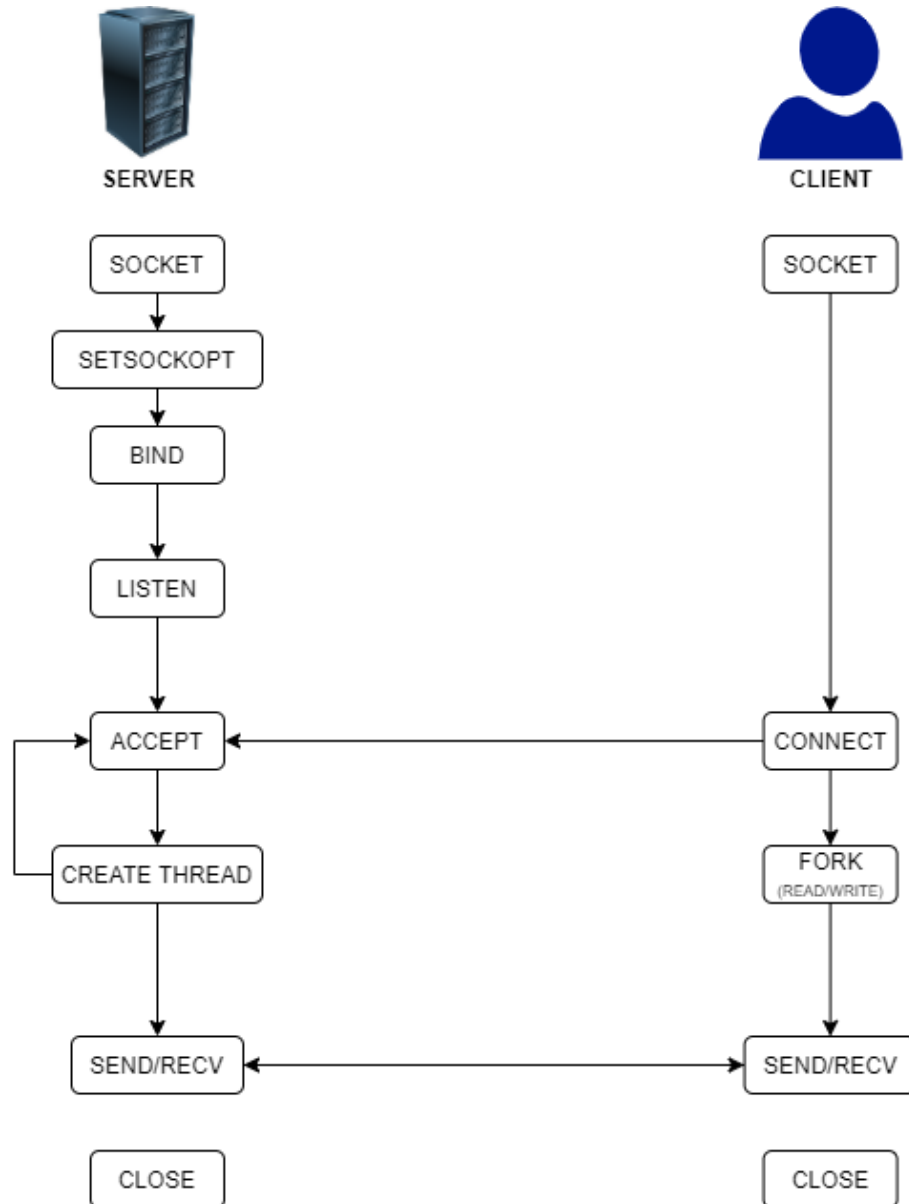
2.1 Limbajul de programare C

Am ales limbajul C deoarece este un limbaj de programare procedural, flexibil și versatil, care îți permite să ai control maxim prin comenzi minime.

2.2 Baza de date MYSQL

Baza de date MYSQL creata folosind MYSQL Workbench

2.3 TCP



Am creat un server iar comunicarea se face pe TCP acesta fiind concurrent. TCP este potrivit pentru aplicațiile care necesită o fiabilitate ridicată, iar timpul de transmisie este relativ mai puțin critic. Există o garanție absolută că datele transferate rămân intacte și ajung în aceeași ordine în care au fost trimise.

Întregul proces poate fi enumerat în următoarele etape:

Server TCP -

1. Folosind `create ()`, cream socket-ul TCP.
2. Folosind `setsockopt` setam optiunile socket-ului
3. Folosind `bind ()`, facem bind la adresa serverului.
4. Folosind `Listen ()`, punem socket-ul serverului într-un mod pasiv, unde așteaptă clientul să se apropie de server pentru a realiza o conexiune.
5. Folosind `accept ()`, în acest moment, conexiunea este stabilită între client și server.
6. Creem un nou thread iar clientul și serverul sunt gata să transfere date.
7. Ne întoarcem la pasul 5.

Client TCP -

1. Cream socket-ul TCP.
2. Conectăm socket-ul clientului la server.

2.4 Multithreading

2.4.1 Ce este un thread?

Thread-ul este un flux de secvență unic în cadrul unui proces. Deoarece thread-urile au unele dintre proprietățile proceselor, ele sunt uneori numite procese lightweight,

2.4.2 Care sunt diferențele dintre procese și thread?

1. Thread-urile nu sunt independente unele de altele, cum sunt procesele.
2. Thread-urile partajează cu alte thread-uri secțiunea lor de cod, secțiunea de date și resursele de sistem de operare, cum ar fi fișierele deschise și semnale.

2.5 De ce Multithreading?

Thread-urile sunt o modalitate populară de a îmbunătăți aplicarea prin paralelism. De exemplu, în browser, mai multe tab-uri pot fi thread-uri diferite. Multithreading-ul folosește mai multe thread-uri, un thread pentru formatarea textului, un alt thread pentru procesarea intrărilor etc. Thread-urile funcționează mai rapid decât procesele din următoarele motive:

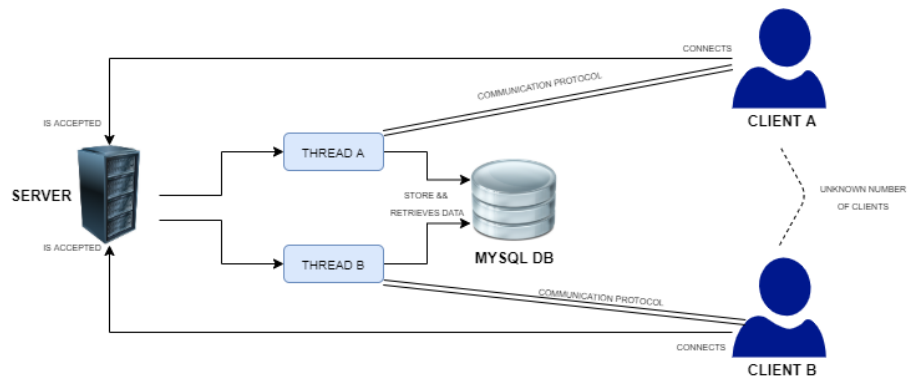
1. Crearea thread-urilor este mult mai rapidă.
2. Comutarea contextului între thread-uri este mult mai rapidă.

3. Thread-urile pot fi terminate cu ușurință.
4. Comunicarea dintre thread-uri este mai rapidă.

Din acest motiv thread-urile sunt mult mai utile decat procesele pentru aplicatia curenta.

3 Arhitectura Aplicatiei

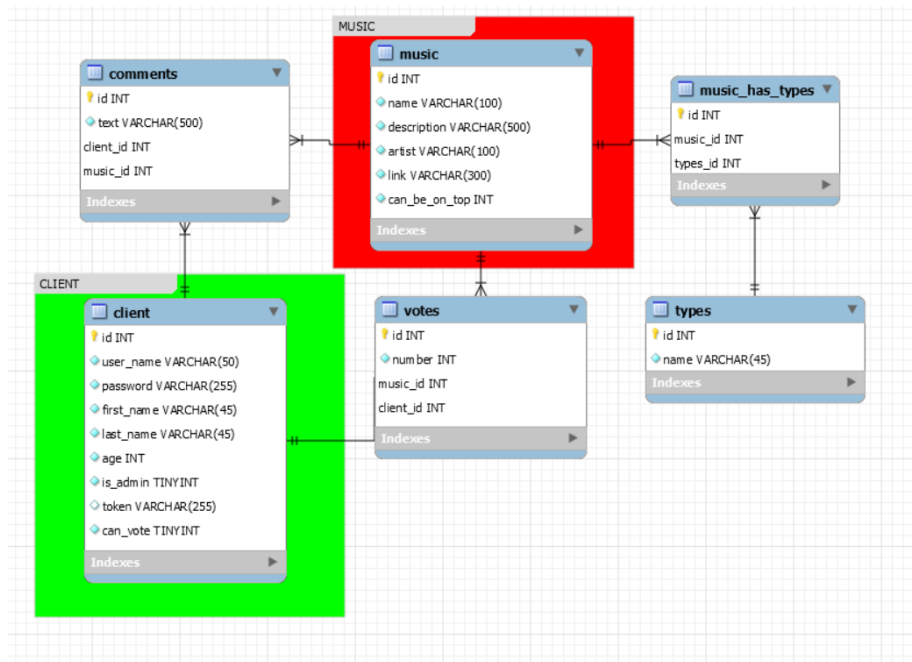
3.1 Diagrama aplicatiei



Clientul se conecteaza la baza de date iar pentru fiecare client va fi creat un thread nou.

Fiecare thread activ poate accesa aceeași baza de date pentru a prelua și a stoca date.

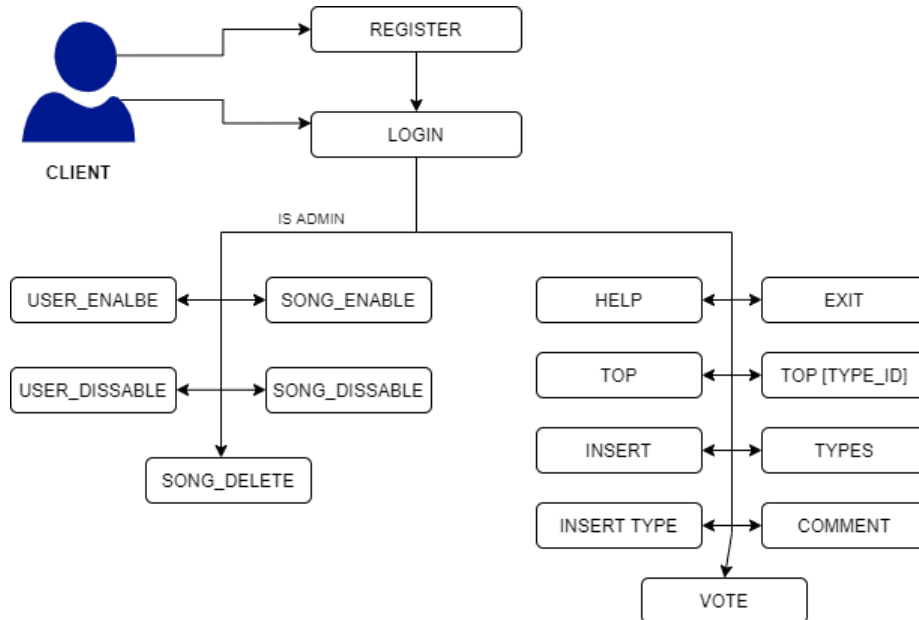
3.2 Baza de date



Relatia many to many este prezenta intre client-coments,client-votes,music-comments,music-votes,music-types.In tabela client avem un camp is_admin care ne va spune daca acel client poate folosi comenzile speciale iar in tabela music un camp can_be_on_top care ne spune daca o melodie poate sau nu aparea in topul muzical. Poate nu vrem sa o stergem ci doar sa nu apara.

4 Detalii de implementare

4.1 Scenarii de utilizare



Utilizatorii neinregistrati au posibilitatea sa isi faca un cont - prin alegerea unui username si a unei parole (register [username] [password])

Cei care sunt deja inregistrati se pot autentifica. (login [username] [password])

Dupa ce s-au autentificat, utilizatorii pot face urmatoarele actiuni:

1. exit (user-ul va fi deconectat de catre server)
2. Adauga o melodie noua (insert [name] [description] [artist] [link] [type-1] [type-2] ... [type-n]);
3. Afiseaza topul curent dupa voturi (top) sau dupa genul muzical (top [type_id]);

4. Afiseaza toate genurile prezente in baza de date (types)

5. Insereaza un nou gen (insert_type [name])

6. Voteaza o melodie (vote [song_id] [number]);

7. Comenteaza melodie (comment [song_id] [text]);

8. Vizualizeze comenzile disponibile (help)

Pe langa actiunile de mai sus, adminii vor putea si sa:

1. Restrictioneze dreptul unei melodii de a fi in top(song_dissable [song_id])

2. Adauga dreptul unei melodii de a fi in top(song_enable [song_id])

3. Stearga o melodie din baza de date (song_delete [song_id])

4. Restrictioneze dreptul de a vota unui utilizator (user_dissable [user_id])

5. Adauga dreptul de a vota unui utilizator (user_enable [user_id])

4.2 Trimiterea si primirea mesajelor

4.2.1 Trimiterea

```
20 void send_msg(int cl, char msgToSend[]){
21     int bytesMsgToSend = sizeof(char)*strlen(msgToSend);
22     char msg[MSG_LENGTH]; sprintf(msg, "%s", msgToSend);
23     printf("SENDING %s cu size %d\n", msgToSend, bytesMsgToSend);
24
25     // SENDING MESSAGE LENGTH BYTES
26     if (write (cl, &bytesMsgToSend, sizeof(int)) <= 0){
27         perror ("[SERVER]Eroare la write() bytes spre client.\n");
28         bzero(msgToSend, bytesMsgToSend);
29         return;
30     }
31
32     // SENDING THE MESSAGE
33     if (write (cl, &msg, bytesMsgToSend) <= 0){
34         perror ("[SERVER]Eroare la write() message spre client.\n");
35         bzero(msgToSend, bytesMsgToSend);
36         return;
37     }
38     bzero(msgToSend, bytesMsgToSend);
39     fflush(stdout);
40 }
41
```

Prima oara se trimite un int reprezentand lungimea mesajului. Abia dupa se va trimite mesajul propriu zis

4.3 Primirea

```
int bytesReceived;
char msgReceived[MSG_LENGTH];
char receiveMsg[MSG_LENGTH];
while(read (sd, &bytesReceived, sizeof(int)) > 0){
if (read(sd, &msgReceived, bytesReceived) <= 0){
    perror ("[CLIENT]Eroare la read() de la server.\n");
    return errno;
}
```

Se va citi mai intai un numar reprezentand lungimea (numarul de bytes) mesajului ce va fi primit. Dupa aceea se va citi continutul mesajului.

4.4 Autentificarea

4.4.1 Functia din server

```
42 void login(int cl,char* username,char* password,int* isLogged, int *isAdmin){
43     char msgToSend[MSG_LENGTH];
44     printf("processing [username]%s [pass]%s\n",username,password);
45     if( username == NULL || password == NULL){
46         strcpy(msgToSend,"Please enter an username and password !");
47     } else {
48         int user_id = login_command(username,password,isAdmin);
49         if(user_id != 0){
50             strcpy(msgToSend,"Login succesfully!");
51             *(isLogged) = user_id;
52         } else {
53             strcpy(msgToSend,"User doesn't exist!");
54         }
55     }
56     send_msg(cl,msgToSend);
57 }
```

Se va cere fiecarui client nou sa introduca un username si o parola , se va verifica daca user-ul exista in baza de date si va fi setat parametrul isLogged cu id-ul gasit pentru datele introduse.

4.4.2 Functia din database handler

```
358 int login_command(char* username, char* password, int *isAdmin){
359     char s[QUERY_LENGTH];
360     sprintf(s,"SELECT * FROM client WHERE user_name=\"%s\" AND password=\"%s\";",username,password);
361     if (mysql_query(conn, s)) {
362         fprintf(stderr, "%s\n", mysql_error(conn));
363         exit(1);
364     }
365
366     MYSQL_RES *result = mysql_store_result(conn);
367     if(mysql_num_rows(result) == 0){
368         return 0;
369     }
370
371     MYSQL_ROW row;
372     row = mysql_fetch_row(result);
373     mysql_free_result(result);
374     *(isAdmin) = atoi(row[6]);
375     return atoi(row[0]);
376 }
377
```

Pentru fiecare client conectat , verificam datele trimise de catre acesta si returnam daca ele exista in baza de date id-ul randului din tabela. Tot pentru fiecare client , setam parametrul is-admin din baza de date , pentru a vedea daca ii vom oferi accesul la comenzile speciale.

5 Concluzii

Aplicatia TopMusic gestioneaza un top muzical si ar putea fi imbunatatita prin implementarea unei interfete grafice.

6 Bibliografie

GITHUB proiect: <https://github.com/CozmaCatalin/TopMusic>

MYSQL connection:

<https://technomanor.wordpress.com/2012/07/01/connect-c-with-mysql-database>

SERVER-CLIENT:

<https://profs.info.uaic.ro/computernetworks/files/NetEx/S12>

GENERATE DATABASE:

Mysql WorkBench

HOW TO COMPILE:

cd client && gcc client.c -o client.exe -lpthread mysql_config -cflags -libs

cd server && gcc server.c -o server.exe -lpthread mysql_config -cflags -libs

HELPERS:

<https://stackoverflow.com/questions/4513316/split-string-in-c-every-white-space>

<https://ramonnastase.ro/blog/ce-este-protocolul-tcp-si-cum-functioneaza/>

<https://www.geeksforgeeks.org/multithreading-c-2/>

<https://www.tutorialspoint.com/multithreading-in-c>

[https://ro.wikipedia.org/wiki/C_\(limbaj_de_programare\)](https://ro.wikipedia.org/wiki/C_(limbaj_de_programare))