

# Retele de calculatoare - Top music

Ciprian Ursulean, grupa X2

December 26, 2020

## 1 Introducere

Proiectul Top music este un proiect foarte practic si intuitiv, acest tip de proiect fiind intr-o forma sau alta folosit de fiecare dintre noi in viata de zi cu zi (aplicatii precum Spotify, SoundCloud, etc). Utilizatorii acestui proiect vor putea sa isi exprime parerile si preferintele muzicale prin voturi si comentarii, vor putea sa vizioneze cele mai apreciate melodii ale momentului si de asemenea vor putea adauga propriile melodii in topuri. Aplicatia va avea doua tipuri de utilizatori, utilizatorul obisnuit, care va putea vizualiza, comenta, vota si adauga melodii, si utilizatorul administrator, care va avea toate functionalitatile unui utilizator obisnuit, plus capacitati de stergere melodii, restrictionare drepturi de votare si de comentare a altor utilizatori. Fiecare utilizator va putea vota o melodie o singura data pentru a preveni spamarea cu voturi si pentru a nu modifica construirea topurilor pe genuri sau a topului general, iar un utilizator cu drepturi de a adauga comentarii va putea adauga oricate comentarii doreste pentru o melodie.

## 2 Tehnologii utilizate

Pe parcursul realizarii acestui proiect am folosit o suita variata de tehnologii, fiecare cu o importanta semnificativa in modul de functionare corect si eficient al aplicatiei. Am decis sa folosesc un server concurent de tip TCP, datorita avantajelor oferite de catre TCP in detrimentul celor oferite de catre un server UDP:

- TCP este orientat conexiune, ceea ce inseamna ca este necesara o conectare la server in vederea realizarii comunicarii
- TCP este foarte sigur si garanteaza transmiterea datelor in siguranta si in ordinea in care acestea sunt trimise
- TCP ne ofera error checking la operatiile de transferuri de date

Utilizand TCP, avem certitudinea ca aplicatia va functiona corect in toate scenariile sale de utilizare, la autentificare de exemplu, triminand numele de

utilizator si parola, serverul le va primit exact in aceasta ordine si utilizatorul se va autentifica cu succes, de asemenea, in cazurile in care vor aparea anumite probleme legate de comunicare, capabilitatile de error checking ale TCP ului vor fi extrem de utile. Aplicatia foloseste un server concurent, care serveste clientii simultan, folosind treaduri multiple, fiecarui client fiind-ui asociat cate un tread. Utilizarea unui sistem de gestiune al bazelor de date a fost, evident, necesara, deoarece proiectul foloseste date din abundenta si o comunicare la ambele parti, ceea ce inseamna operatii dese cu date, operatii care sunt eficiente doar in cazul in care folosim o baza de date. Pe partea de stocare a datelor am utilizat tehnologia SQLITE, datorita anumitelor aspecte:

- SQLITE stocheaza datele in maniera eficienta intr-un fisier, nefiind necesar un server specializat pentru acest lucru. (MySQL, Oracle, Postgresql).
- Este extrem de eficient din punct de vedere al resurselor folosite, fiind numai bun pentru aplicatii de tip embedded sau aplicatii lightweight.
- Este foarte intuitiv si usor de folosit, chiar si in limbajul C, care nu este unul foarte prietenos cu programatorii.

Nu am mentionat limbajul C si platforma Linux/Unix datorita faptului ca era specificata in cerintele problemei folosirea acestora in mod obligatoriu, si nu au fost alte limbaje si platforme alternative cu care sa fac comparatie.

### 3 Arhitectura aplicatiei

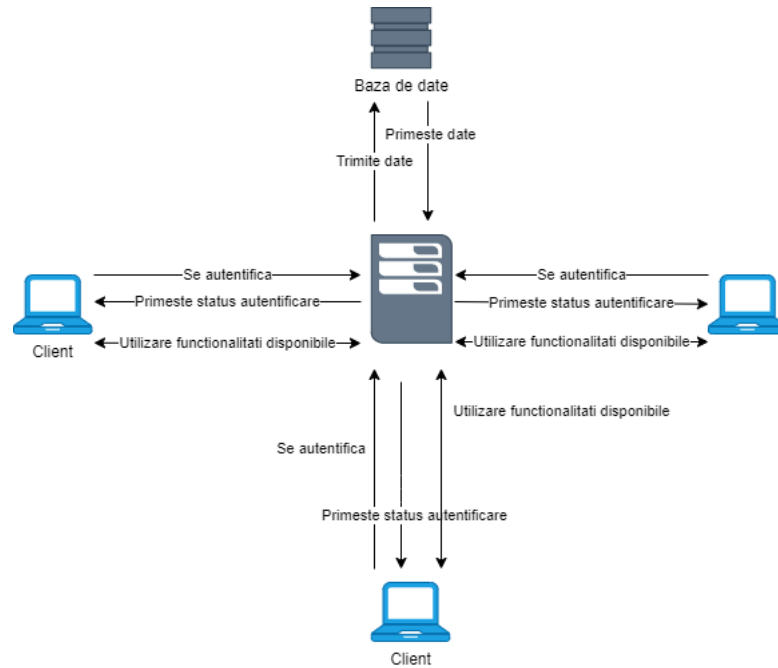


Figura 1: Diagrama client - server a aplicatiei Top music. De retinut ca serverul suporta  $n$  clienti conectati, dar pentru simplitate am adaugat un numar de 3 clienti conectati.

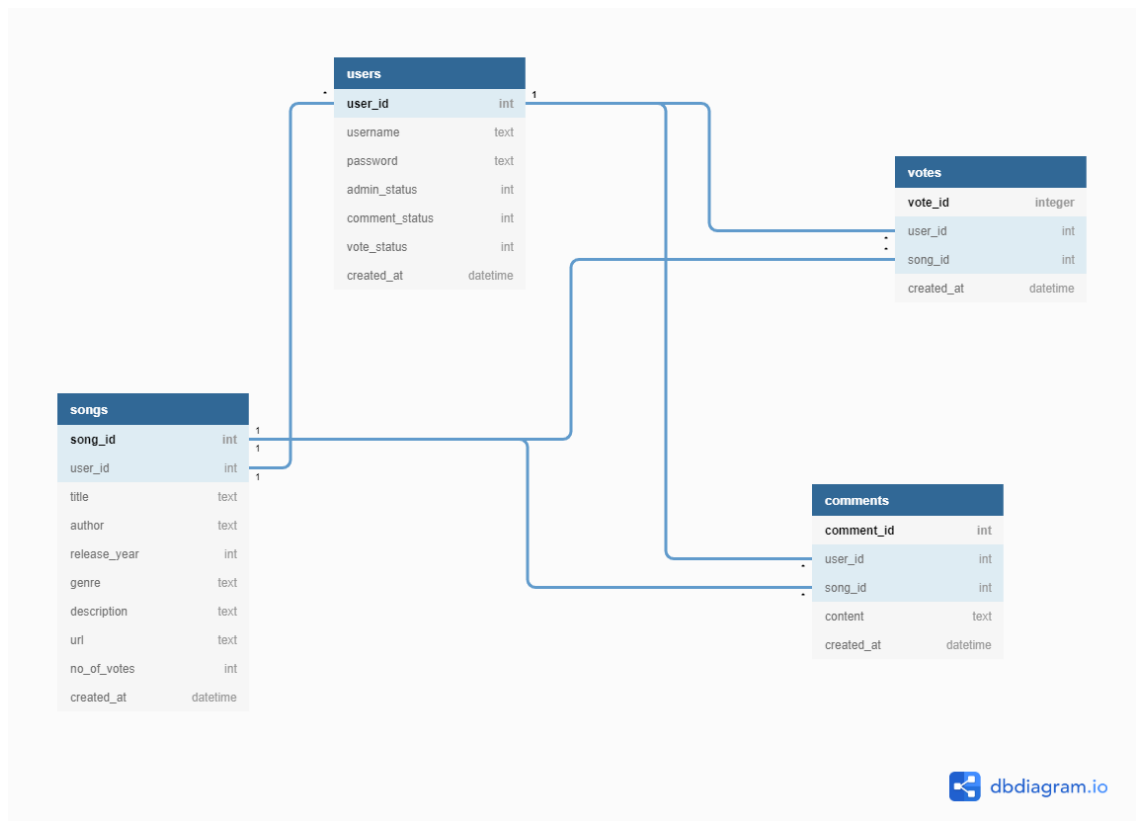


Figura 2: Diagrama bazei de date a aplicatiei Top music

## 4 Detalii de implementare

Pentru a se putea bucura de functionalitatile aplicatiei Top music, utilizatorii trebuie sa detina un cont, si fie se autentifica cand pornesc aplicatia, fie isi creaza unul si se autentifica ulterior.

In imaginea de mai jos observam ca un cont de utilizator este necesar pentru a utiliza aplicatia, fie unul existent fie unul nou.

```

138     printf("Pentru creare cont introduceti `1`\n");
139     printf("Pentru logare introduceti `2`\n");
140     printf("Introduceti optiunea: ");
141     scanf("%d", &optiune_votare);
142     while (optiune_votare < 1 || optiune_votare > 2) {
143         printf("Optiune incorecta, incercati din nou: ");
144         scanf("%d", &optiune_votare);
145     }

```

Dupa ce trimitem numele de utilizator si parola la server, acesta va cauta in baza de date, sau va insera in baza de date numele de utilizator si parola, dupa caz.

In imaginile de mai jos se observa cum este cautat in baza de date un utilizator si cum ii sunt preluate datele relevante, in cazul in care acesta exista.

```

215 void autentificare_utilizator(sqlite3* db, char* nume_utilizator, char* parola) {
216     char sql_query[512] = "SELECT user_id, admin_status, comment_status, vote_status FROM users WHERE username = '";
217     char* mesaj_eroare;
218
219     strcat(sql_query, nume_utilizator);
220     strcat(sql_query, "'");
221     strcat(sql_query, " AND password = '");
222     strcat(sql_query, parola);
223     strcat(sql_query, "'");
224
225     int db_descriptor = sqlite3_exec(db, sql_query, login_callback, 0, &mesaj_eroare);
226 }

```

```

77 static int login_callback(void *NotUsed, int argc, char **argv, char **azColName) {
78     user_id = atoi(argv[0]);
79     admin_status = atoi(argv[1]);
80     comment_status = atoi(argv[2]);
81     vote_status = atoi(argv[3]);
82     return 0;
83 }

```

In imaginea de mai jos este prezentata, intr-o forma limitata, din considerente de spatiu, implementarea functiei de creare cont nou. La crearea unui cont nou se va verifica daca este disponibil numele de utilizator dorit.

```

191     if (already_exists_user == 1) {
192         already_exists_user = -1;
193         return -1; // utilizatorul exista deja
194     }
195     else {
196         strcpy(sql_query, "INSERT INTO users (username, password, admin_status, comment_status, vote_status, created_at) VALUES ('");
197         strcat(sql_query, nume_utilizator);
198         strcat(sql_query, "','");
199         strcat(sql_query, "");
200         strcat(sql_query, parola);
201         strcat(sql_query, "','");
202         strcat(sql_query, " 0, 1, 1, DATE('NOW'))");
203         //printf("%s\n", sql_query);
204         //fflush(stdout);
205         select_status = sqlite3_exec(db, sql_query, 0, 0, &mesaj_eroare);
206         if (select_status != SQLITE_OK) {
207             printf("Eroare la cautarea utilizatorului - %s!", mesaj_eroare);
208             fflush(stdout);
209             sqlite3_free(mesaj_eroare);
210         }
211     }

```

Dupa ce am creat un cont cu succes sau dupa ce ne-am autentificat cu un cont deja existent, in functie de tipul de cont (simplu sau administrator) vom primi un meniu cu diverse optiuni.

```

29 void afisare_meniu_basic() {
30     printf("* Pentru adaugarea unei melodii la top introduceti `1`\n");
31     printf("* Pentru votarea unei melodii introduceti `2`\n");
32     printf("* Pentru afisarea topului general introduceti `3`\n");
33     printf("* Pentru afisarea topului pe genuri introduceti `4`\n");
34     printf("* Pentru adaugarea unui comentariu introduceti `5`\n");
35     printf("* Pentru vizualizarea comentariilor unei melodii introduceti `6`\n");
36 }
37
38 void afisare_meniu_admin() {
39     afisare_meniu_basic();
40     printf("* Pentru stergerea unei melodii introduceti `7`\n");
41     printf("* Pentru restrictionarea la vot a unui utilizator introduceti `8`\n");
42     printf("* Pentru restrictionarea de a comenta a unui utilizator introduceti `9`\n");
43 }

```

In functie de dorinta utilizatorului, acesta va introduce o optiune si va putea folosi aplicatia dupa bunul plac. De asemenea, un utilizator obisnuit nu va avea acces la comenzile destinate administratorilor. In imaginile de mai jos este prezentata partea centrala a proiectului top music, o buna parte din optiuni realizanduse prin anumite modificari ale functiilor urmatoare:

```

337 void afisare_lista_generala_melodii() {
338     char sql_query[128] = "SELECT title, url FROM songs";
339     char* mesaj_eroare;
340
341     int select_status = sqlite3_exec(db, sql_query, afisare_lista_melodii_callback, 0, &mesaj_eroare);
342     if (select_status != SQLITE_OK) {
343         printf("Eroare la selectarea melodiilor pentru comentare - %s!", mesaj_eroare);
344         fflush(stdout);
345         sqlite3_free(mesaj_eroare);
346     }
347 }

```

```

131 static int afisare_lista_melodii_callback(void *NotUsed, int argc, char** argv, char** azColName) {
132     char formatare_piesa[512];
133     strcpy(formatare_piesa, "Titlu: ");
134     strcat(formatare_piesa, argv[0]);
135     strcat(formatare_piesa, "\nURL: ");
136     strcat(formatare_piesa, argv[1]);
137
138     strcpy(lista_melodii[nr_melodii++], formatare_piesa);
139     return 0;
140 }

```

## 5 Concluzii

În concluzie, proiectul Top music mi se pare unul dintre cele mai interesante proiecte propuse la materia Rețele de calculatoare, acesta având un puternic impact practic. Top music ar mai putea fi îmbunătățit prin adăugarea unei interfețe grafice atractive și intuitive, care să fie mai prietenoasă cu utilizatorii decât interfața minimală de la linia de comandă. De asemenea, ar fi binevenite și niste funcționalități precum descărcare melodii pe calculatorul utilizatorului precum și generarea de playlist-uri în funcție de anumite criterii.

## 6 Bibliografie

<https://profs.info.uaic.ro/computernetworks/cursullaboratorul.php>  
<https://www.youtube.com/?hl=ro&gl=RO>  
<https://www.sqlite.org/cintro.html>  
<https://www.geeksforgeeks.org/sql-using-c-c-and-sqlite/>  
<http://zetcode.com/db/sqlite/>  
<https://stackoverflow.com/questions/28969543/fatal-error-sqlite3-h-no-such-file-or-directory/31764947>  
<https://app.diagrams.net/>  
<https://dbdiagram.io/home>