Raport al proiectului TopMusic (B)

Gîrbea Dumitru-Andrei Grupa B1 anul II

January 14, 2020

1 Introducere

Scopul acestui raport este acela de a detalia încercarea mea de a realiza proiectul de tip B, TopMusic, pe care am să îl fac în limbajele C/C++. Acest proiect este o aplicație server-client ce se ocupa cu datele unui top muzical ce conține diferite genuri muzicale (ex: dance, pop, hip-hop, soul, folk, etc.).

Principalele funcționalități ale aplicației sunt: înregistrarea de "normal users" și administratori, logarea lor în aplicație, iar numai daca această acțiune a fost finalizată utilizatorii și adminii vor putea sa execute diverse comenzi pentru prelucrarea datelor din top sau introducerea unor date noi.

Așadar utilizatorii normali vor avea următoarele drepturi: de a adăuga melodii în top (împreună cu numele acesteia, o descriere, genul/genurile acesteia și un link către videoclipul melodiei de pe Youtube sau alte site-uri asemănătoare),pot posta diferite comentarii unei melodii, va vota melodii, va afișa topul general, topul pentru un anumit gen muzical sortat după numărul de voturi, vor putea descarca o melodie sau videoclipul acesteia sau vor putea sa o asculte; de asemenea administratorii vor putea să facă tot ceea ce fac utilizatorii normali însă pe lângă toate cele de mai sus ei vor putea să șteargă o melodie din top și vor restricționa dreptul la vot a unui utilizator normal.

2 Tehnologiile utilizate

Pentru realizarea conexiunii server-client am considerat că cel mai potrivit ar fi utilizarea modelului TCP (Transmission Control Protocol) în ciuda modelului UDP (User Datagram Protocol) care deși este mult mai rapid (fiind utilizat în special pentru apelurile telefonice, transmisiunile live, pentru jocurile online, etc.) acesta nu îmi asigură siguranța sosirii datelor la destinație, iar în cazul în care constatăm că pachetele noastre s-au pierdut ele nu mai pot

fi recuperate. Spre deosebire de UDP, protocolul TCP care este un protocol orientat spre conexiune garantează că destinatarul va primi pachetele în ordine, dacă acestea au ajuns la destinație destinatarul va trimite mesaje expeditorului spunând că a primit pachetele, dacă expeditorul nu primește un răspuns pozitiv va retrimite pachetele pentru a se asigura că destinatarul le-a primit, făcând acest lucru cât timp există o conexiune deschisă. De asemenea pachetele trimise cu TCP sunt verificate de erori pentru a nu deveni corupte obținând astfel fiabilitate. Așadar consider că integritatea datelor este vitală pentru corectitudinea topului meu muzical iar faptul că datele (de exemplu voturile) nu se vor pierde sunt sunt motivele ce m-au făcut să optez pentru modelul TCP.

De asemenea serverul va trebui sa permită conectarea mai multor clienți astfel pentru a face modelul nostru unul concurent ne vom folosi de primitiva fork(). Astfel serverul TCP concurent va crea câte un proces copil pentru fiecare client, acesta va trebui să preia datele de la client, să le proceseze și să le transmită la server și poate mai târziu să trimită rezultatul înapoi pentru a fi afișat clientului pe ecran.

3 Arhitectura aplicației

După cum am spus mai sus proiectul este o aplicație server-client ce servește la managementul unui top muzical, însă acest lucru presupune stocare de date. Am ales ca să stochez datele într-o bază de date MySQL deoarece așa aș putea manipula datele mai ușor. Vom crea o legatură între program și baza de date iar în funcție de comanda introdusă în terminal, se va returna răspunsul care va corespunde interogarii SQL asociată comenzii respective. Pentru a nu avea conflicte în baza de date vom asocia fiecărei melodii un SONG_ID unic; baza de date va cuprinde următoarele tabele:

- CLIENTS cu: LOGIN (numele user-ului), PASSWORD (parola), ADMIN (0-administrator, 1- normal user), RESTRICT(0 - are drept de votare, 1 - dreptul de votare a fost luat)
- SONGS cu: SONG_ID (un id unic fiecărei melodii - primary key), NAME (numele melodiei), DESCRIPTION (descrierea acesteia), URL (linkul spre videoclip), VOTES (numărul de voturi)
- GENRES cu: SONG_ID (id-ul melodiei), GENRE (genul melodiei)
- COMMENTS cu: SONG_ID (id-ul melodiei), COMMENT (comentariul), LOGIN_CLIENT (numele celui ce a pus comentariul)

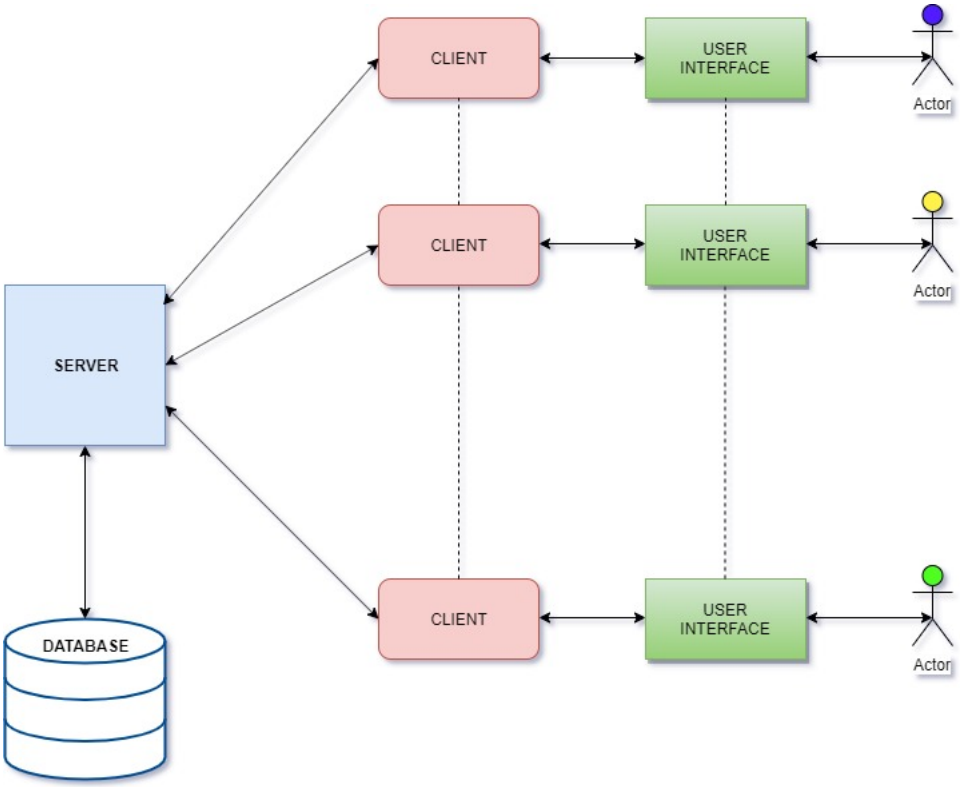

Figure 1: Diagrama proiectului

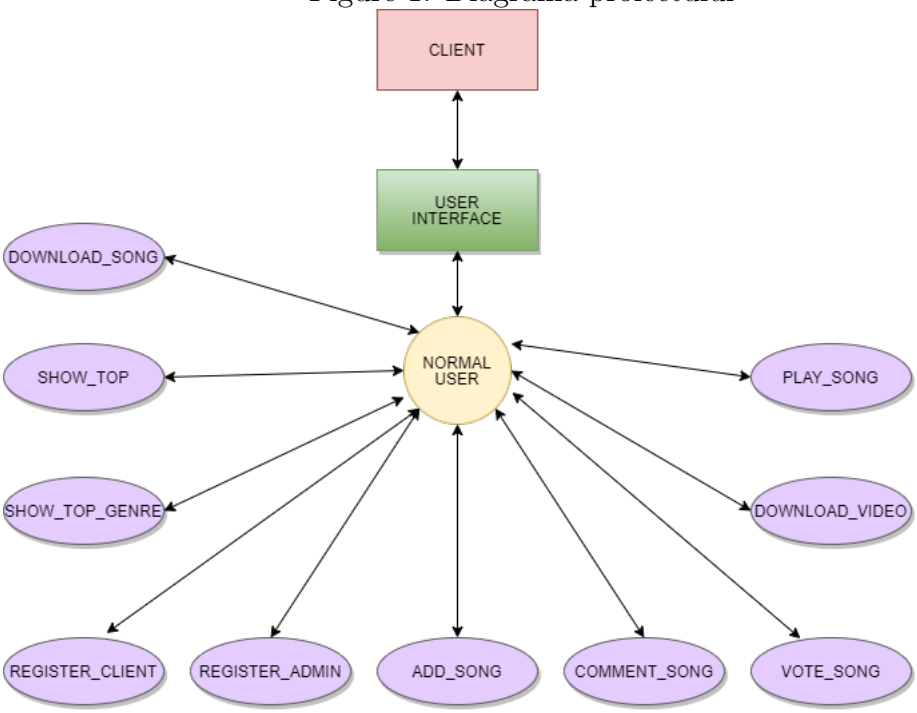

Figure 2: Funcțiile utilizatorului normal

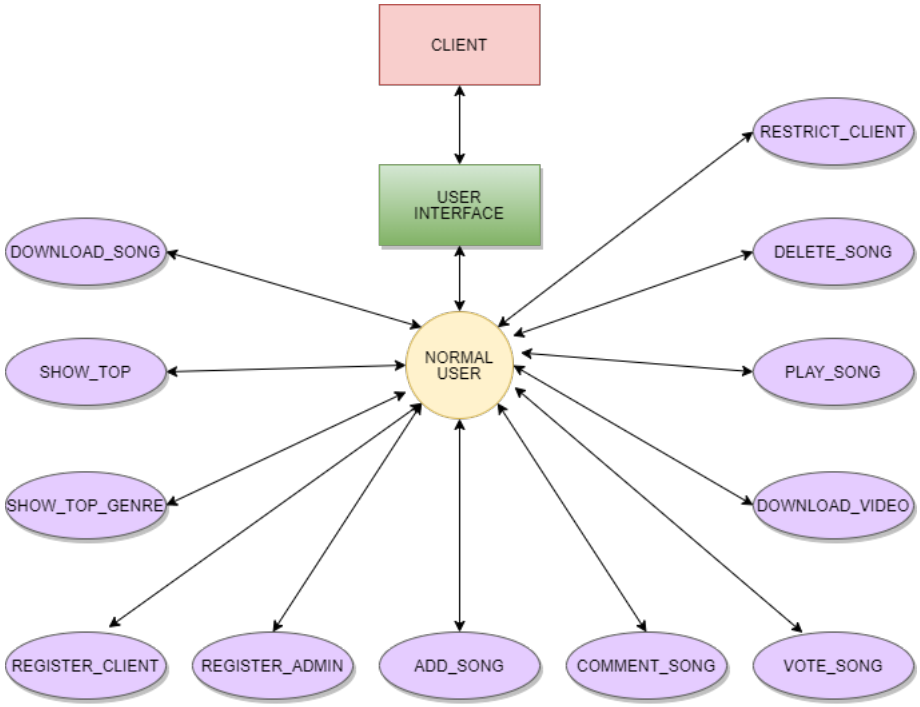

Figure 3: Funcțiile administratorului

4 Detalii de implementare

Vom implementa o funcție REGISTER pentru înregistrarea de ADMIN sau NORMAL USER care va avea trei câmpuri nume, parola și tipul de utilizator și o funcție LOGIN cu doua câmpuri pentru nume și parolă.
Fiecare NORMAL USER va putea executa următoarele funcții:

- funcția pentru: înregistrarea utilizatorilor normali: register_client <login> ¡password¿
- funcția pentru: înregistrarea administratorilor: register_admin <login> <password>
- adăugare de melodii: add_song <name> @ <url> @ <description> @ <genres> cu:
  - adăugarea de NUME
  - adăugarea unei DESCRIERI
  - GENUL în care se încadrează
  - un link către VIDEOCLIPUL melodiei

- funcția pentru votare: vote_song <id>
- funcția pentru afișarea topului: show_top
- funcția pentru afișarea topului în funcție de gen: show_genre_top <genre>
- funcția pentru adăugarea unui comentariu: comment_song <id> <comment>
- funcția pentru restricționarea accesului unui utilizator: restrict_client <login>
- funcția pentru descărcarea unei melodii din top: download_song <id>
- funcția pentru descărcarea videoclipului unei melodii din top: download_video <id>
- funcția pentru ascultarea unei melodii din top:play_song <id>

Fiecare ADMIN va putea executa toate funcțiile anterioare plus următoarele:

- funcția pentru ștergerea unei melodii: delete_song <id>
- funcția pentru restricționarea accesului unui utilizator: restrict_client <login>

```
char * register_new_admin(char *msg){
    if (client_online==0){
    int start_login=0;
    msg+='\0';
    char login[20],password[20];
    while (msg[start_login]!=' ')start_login++;
    int start_password=++start_login;
    while (msg[start_password]!=' '){
       login[start_password-start_login]=msg[start_password];
       start_password++;
    }
    login[start_password-start_login]='\0';
    int end=++start_password;
    while (msg[end]!='\0'){
       password[end-start_password]=msg[end];
       end++;
    }
    password[end-start_password-1]='\0';
    printf("login *%s* *%s* \n",login,password);

    char sqlQuery[200] = "SELECT login,password FROM clients WHERE login = '";
    strcat(sqlQuery,login);
    strcat(sqlQuery,"'");
    mysql_query(connectdb,sqlQuery);
    raspuns=mysql_store_result(connectdb);
    rowdb=mysql_fetch_row(raspuns);
    if (rowdb == NULL) {
        char insert[200]= "INSERT INTO clients(login,password,admin) VALUES ('";
        strcat(insert,login);
        strcat(insert,"','");
        strcat(insert,password);
        strcat(insert,"',1)");
        printf("%s\n",insert);
        mysql_query(connectdb,insert);
        client_online=2;
        return "Utilizator admin inregistrat cu succes";
    } else
    return "Login deja folosit!";
    } else return "Sunteti deja conectat";

}
```

Figure 4: Exemplu de cod

5 Concluzii

Consider că aplicația ar putea să fie îmbunătățită astfel: să punem o restricție astfel încât un utilizator să nu poată vota de mai multe ori aceași melodie, să putem șterge comentarii.

Implementarea unei interfețe grafice ar facilita anumite acțiuni și ar da un plus aplicației. M-am gândit că un player audio care ne-ar da posibilitatea de a asculta melodiile din top, ne-ar ajuta să ne decidem pe care melodie să votăm, de asemenea dacă am putea reda videoclipul melodiei în interfață am avea și un suport vizual care sa ne dea idei despre ce comentarii am putea să lăsăm la cântecul respectiv.

Pentru ca ideea de a avea un player audio să funcționeze trebuie să ne gândim unde am putea să stocăm melodiile, o primă modalitatea ar fi pe calculatorul personal însă ar fi destul de greu pentru altcineva să asculte melodiile noastre așa că utilizarea unui cloud ar fi soluția potrivită. O altă îmbunătățire ar putea fi crearea unei funcții care să descarce din cloud o melodie și să putem sa specificăm și locația unde sa stocăm melodia descărcată. Cât despre tehnologiile folosite un server UDP ar fi mult mai bun din punct de vedere al vitezei dacă considerăm implementarea îmbunătățirilor de mai sus existentă.

6 Bibliografie

- https://ro.topinfoweb.com/the-difference-between-tcp-and-udp-protocols-is
- https://ro.wikipedia.org/wiki/SQLite
- http://www.geekride.com/fork-forking-vs-threading-thread-linux-kernel
- https://www.smartdraw.com/use-case-diagram/
- https://about.draw.io/uml-use-case-diagrams-with-draw-io/
- https://en.wikipedia.org/wiki/Qt_(software)
- https://ro.wikipedia.org/wiki/SFML
- https://ro.sawakinome.com/articles/technology/tcp-vs-udp.html?fbclid=IwAR1R6mAINvm-L_VeNn9Fg9NKz-yWIw0zESq9Vc5mve7zrNVMUnu6rjennJ8

- https://www.audio.ro/stiatica/compararea-metodelor-video-de-tran sport-care-este-diferenta-dintre-tcp-si-udp/?fbclid=IwAR0jDyqF-foM60RSsSpp A-8tgkpftqzCwFTdxlatyQAmqJweZ6aHylVn7a0