

Raport Tema 2 - TopMusic (B)

Paduraru Andra - Elena, Anul II, Grupa A4

1 Introducere

Aceasta fisa de raport are rolul de a prezenta detalii ce au legatura cu realizarea proiectului viitor pe care doresc sa il fac in limbajul C, si anume TopMusic (B).

Acesta este o aplicatie client-server, care are rolul de a lucra cu datele unui top muzical ce contine diverse genuri. Functionalitatile aplicatiei constituie din inregistrarea administratorilor si a utilizatorilor, logarea acestora si comenzile pe care le vor executa doar dupa ce se vor loga in aplicatie.

Adminul va avea dreptul de a sterge melodiile din top si de a restrictiona dreptul de vot al unui utilizator. Utilizatorul va putea sa voteze o melodie si deci sa o adauge in top, sa posteze comentarii la diverse melodii, sa vizualizeze topul general cat si topul pe genuri muzicale (dance, rock, hip-hop etc.). O melodie va avea asociat un nume, o descriere, va apartine unuia sau mai multor genuri de muzica si va avea asociat un link catre videoclip (pe YouTube sau alte site-uri).

2 Tehnologii utilizate

Pentru realizarea conexiunii server-client se va utiliza modelul TCP concurent (modelul orientat-conexiune), care permite posibilitatea conectarii mai multor clienti la un server. Cateva caracteristici ale conexiunii sunt: mecanisme de control al fluxului si de control al congestiei, asigurarea transmiterii datelor in ordine.

Serverul TCP va crea cate un proces copil pentru fiecare client in asa fel incat va exista posibilitatea servirii mai multor clienti in mod simultan. Acesta va avea rolul de a prelua datele de la client, de a le procesa si ulterior va transmite rezultatul inapoi la client.

Clientul va stabili un port pentru conexiune. Acesta va trebui sa preia datele introduse de utilizator si sa le transmita la server, iar dupa ce le primeste inapoi, le va afisa pe ecran.

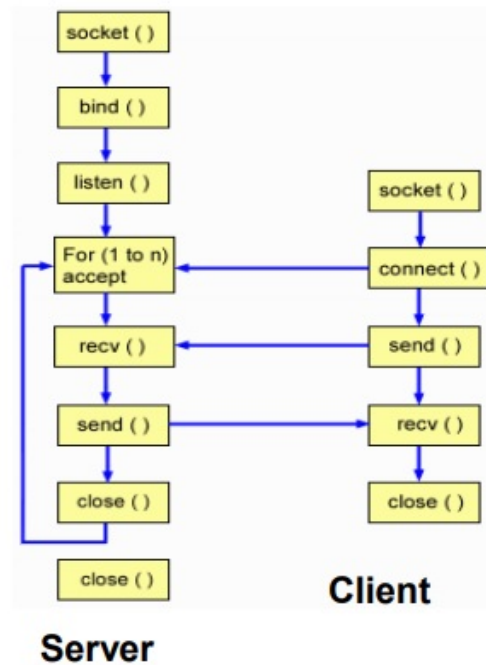
Functionalitatea se poate observa in diagrama din cadrul capitolului 3, in care este prezentat pe scurt cum functioneaza conexiunea server-client.

3 Arhitectura aplicatiei

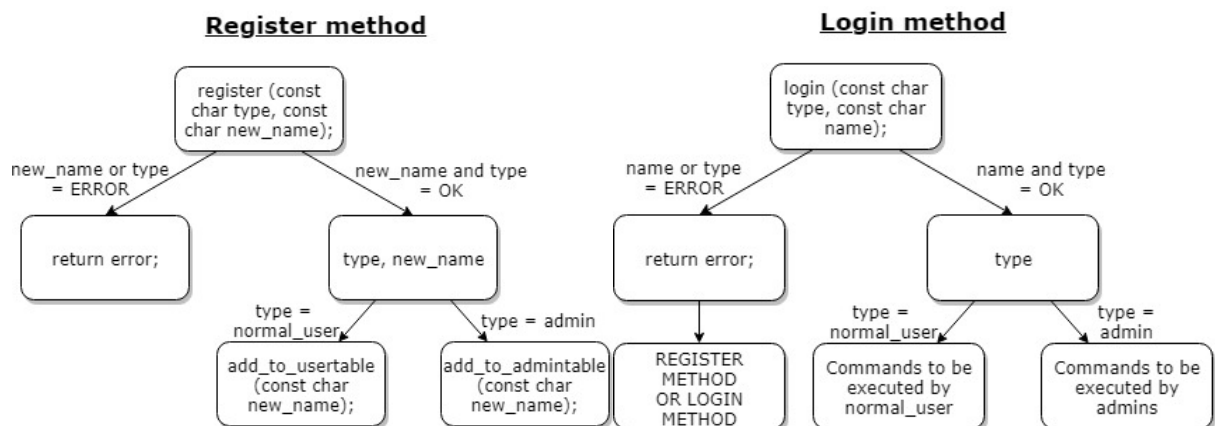
Se va utiliza legatura dintre o baza de date SQLITE si programul in C. Baza de date va contine mai multe tabele, cel putin urmatoarele: tabela cu melodii si detaliile despre melodii, tabela cu utilizatori, cea cu administratori, tabele pentru fiecare gen in parte, tabela de comentarii. In functie de necesitate se vor putea adauga si alte tabele ulterior.

Pentru a putea lega o baza de date la programul C, se va instala baza de date compatibila cu versiunea de Linux utilizata, se vor instala librariile necesare si in cadrul programului C se va crea legatura dintre cele doua.

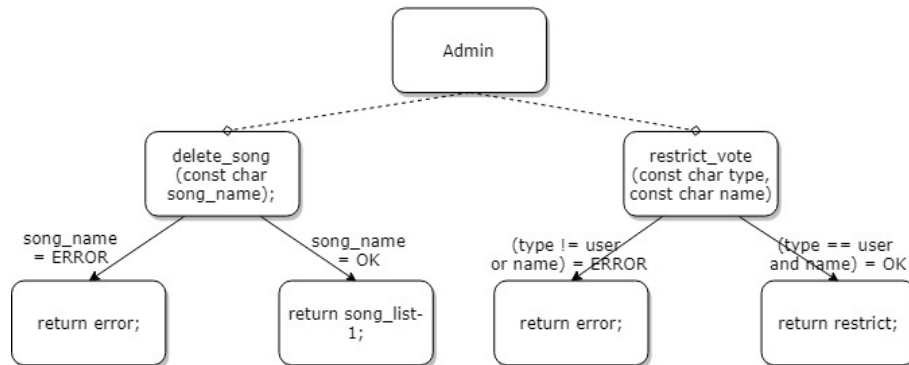
In functie de comanda introdusa in terminal, se va returna raspunsul care va corespunde interogarii SQL asociata acelei comenzi. Pe urmatoarele pagini se afla diagramele ce arata cum va functiona codul in C odata ce aplicatia este pornita:



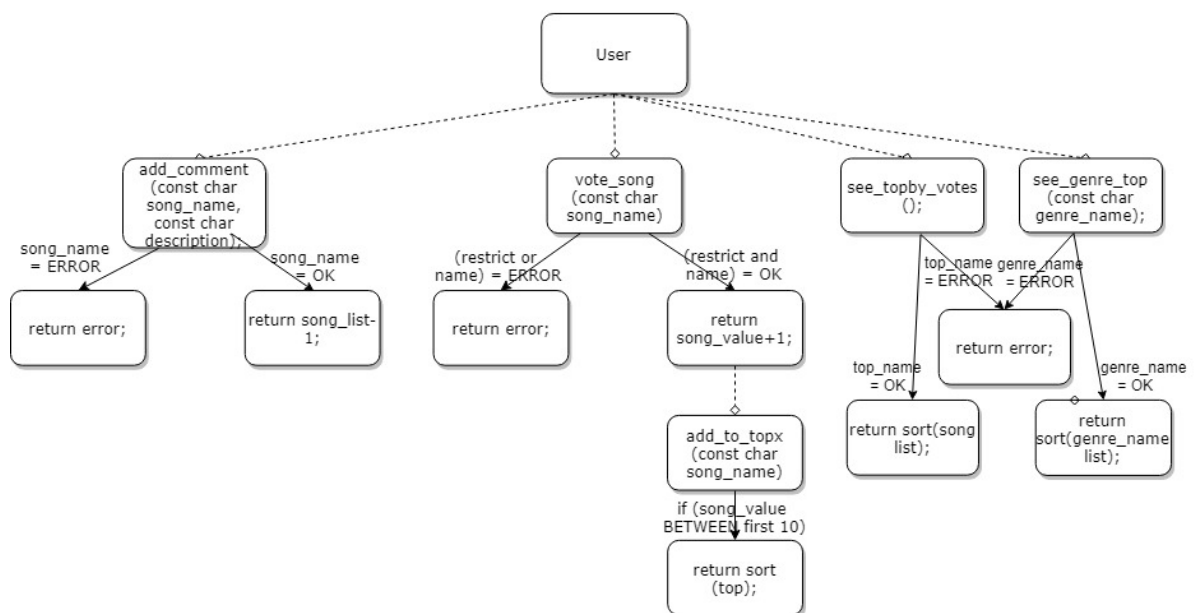
Model server/client TCP



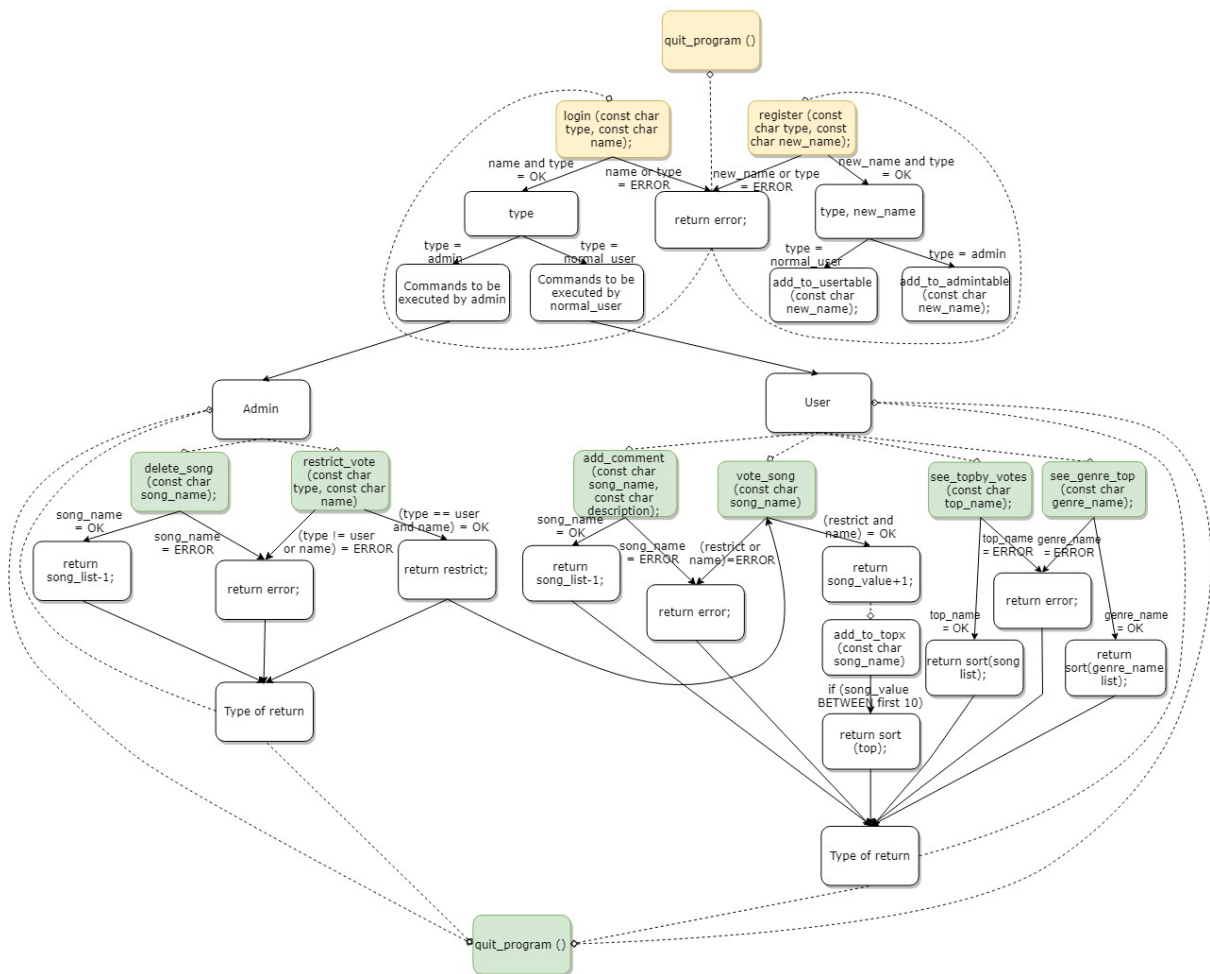
Admin's commands method



User's commands method



Program functions



4 Detalii implementare

Pentru ca aplicatia sa functioneze in modul dorit, functiile implementate vor trebui sa respecte ultima diagrama.

Funcția de înregistrare are rolul de a înregistra un admin sau un user, în funcție de tip și numele dorit, care se va memora în baza de date. Dacă tipul este gresit sau numele există deja în tabela SQL, se va afișa un mesaj de eroare și se va cere o nouă încercare sau dacă este dorită, ieșirea din program. Încercarea poate fi de înregistrare sau logare în cazul în care persoana în cauză are deja un cont.

Funcția de logare funcționează pe aceleași principii. Odată logat cu succes, în funcție de tip (administrator sau utilizator) se vor putea realiza diverse comenzi.

Adminul va avea dreptul de a ieși din program, șterge o melodie în funcție de numele ei sau de a restricționa dreptul unui user de a vota. Dacă numele melodiei este gresit sau dacă numele și tipul utilizatorului nu sunt corecte, se va afișa o eroare. Adminul va putea ulterior rescrie comanda.

În cadrul funcției de restricționare, se va seta un parametru special, care, în funcție de numele utilizatorului, va restricționa dreptul de votare al acestuia.

Utilizatorul poate ieși din program, poate adăuga un comentariu la o melodie în funcție de numele melodiei, poate să o voteze și vizualiza topurile. Dacă se produce vreo eroare, se va afișa un mesaj iar user-ul va avea dreptul de a rescrie comanda.

Voturile la melodii vor fi inițializate la începutul programului cu 0. Într-o coloană numită "Restrictionat" din baza de date în dreptul user-ului, prin valoarea 1 se va înțelege că userul are voie să voteze și prin valoarea -1, că nu are dreptul de a vota nicio melodie. Legătura între tabele se va realiza prin Join-uri. Pentru votare, se va verifica întâi dacă coloana "Restrictionat" are valoare pozitivă (1) și apoi se va actualiza votul pentru melodie prin incrementarea lui cu 1 și vor putea fi actualizate și listele de topuri din cadrul bazei de date.

Funcția de ieșire din program va putea fi accesată oricând și va deloga orice client care a cerut acest lucru și are sesiune activă.

Mai jos se află câteva imagini care prezintă pe scurt cum ar trebui să funcționeze câteva dintre funcțiile programului:

```
1 //partea de register
2 if (new_name != (SELECT name FROM users WHERE name = new_name) && strcmp(type, "user")
   == 0)
3     add_to_usertable (new_name);
4 else if (new_name != (SELECT name FROM admins WHERE name = new_name) && strcmp(type, "
   admin") == 0)
5     add_to_admintable (new_name);
6 else
7 {
8     printf ("%s, "Incorrect type or name already exists. If you already have an account,
   try to log in.\n");
9 }
```

```
1 //partea de login
2 if ( (name == (SELECT name FROM users WHERE name = name) && strcmp(type, "user") == 0)
3     //functions to be executed by users
4 else if (name == (SELECT name FROM admins WHERE name = name) && strcmp(type, "admin")
   == 0)
5     //functions to be executed by users
6 else
7 {
8     printf ("%s, "Incorrect type or name already exists.\n");
9 }
```

```
1 //partea de votare (admin)
2 result = ("SELECT nume FROM users WHERE nume = '%s'", name);
3 if (type != "user" || result != name)
4     printf ("%s, "Incorect user type or name does not exist!\n");
5 else
6     restrict_vote (type, name);
7
8 restrict_vote (const char* type, const char* name)
9 {
10     result = ("SELECT nume FROM users WHERE nume = '%s'", name);
```

```

11 UPDATE users SET restrict = -1 WHERE name = result;
12 printf ("%s, "Table modified.\n");
13 }

1 //partea de votare melodie (user)
2 result = (SELECT name FROM song WHERE name = song_name);
3 if (result != song_name)
4     printf ("%s, "You typed in the wrong name!\n")
5 else
6 {
7     if ((SELECT restrict FROM users WHERE name = actual_user) == 1)
8         vote_song (song_name);
9     else
10         printf ("%s, "You are restricted from voting any song by an admin!\n")
11 }
12
13 vote_song (const char* song_name)
14 {
15     UPDATE song SET vote = (SELECT vote FROM song WHERE name = song_name)+1 WHERE name =
        song_name;
16 }

```

5 Concluzii

Solutia data ar putea fi imbunatatita prin utilizarea unor chei primare in baza de date, pentru a lega tabelele intre ele. Sintaxele SQL complicate ar fi astfel evitate. In solutia propusa se vor folosi sintaxe pentru join-uri de tabele. De asemenea, o alta imbunatatire ar putea fi efectuata la votare, in asa fel incat un user sa nu poata vota o melodie de mai multe ori si daca exista mai multe melodii cu acelasi numar de voturi ce se pot afla in top, sa existe o departajare a acestora. De asemenea, o varianta mai buna pentru inregistrarea adminilor ar fi ca aceasta sa se poata realiza doar daca un admin este deja inregistrat si logat, pentru a evita riscul ca un user obisnuit sa devina admin.

In privinta tehnologiilor utilizate, un server UDP ar fi mai bun din punctul de vedere al vitezei, fata de TCP. Acesta ar fi mai performant dar nu garanteaza integritatea datelor.

6 Bibliografie

Site-uri utilizate:

<https://www.youtube.com/watch?v=n5Ed3DA2sbE>
<https://www.tutorialspoint.com/sqlite/sqliteccpp.htm>
<https://www.sqlite.org/index.html>
<http://zetcode.com/db/sqlitec/>
<https://gist.github.com/jsok/2936764>
<http://www.wassen.net/sqlite-c.html>
<https://packages.ubuntu.com/search?keywords=sqlite3>
<https://linuxhint.com/installsqlitebrowserubuntu1804/>
<https://askubuntu.com/questions/340603/sqlite-installation>
<https://www.thoughtco.com/creating-populating-running-database-sql-query-958233>
<https://stackoverflow.com/questions/28602168/mysql-insert-statement-in-c>
<https://profs.info.uaic.ro/computernetworks/>
<https://sites.google.com/view/fii-rc>
<https://www.ibm.com/support/knowledgecenter/en/SSLTBW2.1.0/>
<https://www.cs.dartmouth.edu/campbell/cs50/socketprogramming.html>
<https://www.w3schools.com/sql/sqlalter.asp>
<https://stackoverflow.com/questions/3024546/change-one-cells-data-in-mysql>
<https://stackoverflow.com/questions/4361774/mysql-update-multiple-tables-with-one-query>
<http://www.mysqltutorial.org/mysql-update-data.aspx>
<http://www.mysqltutorial.org/mysql-update-join/>