

TopMusic Documentatie

Mihai Cristian-Laurentiu, Anul 2, Grupa B4

Universitatea Alexandru Ioan Cuza din Iasi

1 Introducere

TopMusic este o aplicatie de tipul client/server pentru managementul unui top muzical continand diverse genuri muzicale. Aceasta ofera utilizatorilor posibilitatea sa voteze o melodiile favorite, sa vizualizeze topul complet sau pe genuri muzicale al melodiilor si sa adauge comentarii acestora. Interfata aplicatiei este realizata in terminalul Linux, acesta nefind foarte prietenoasa cu utilizatorul dar oferind comenzi usor de folosit pentru facilitatile mentionate mai sus.

2 Tehnologii utilizate

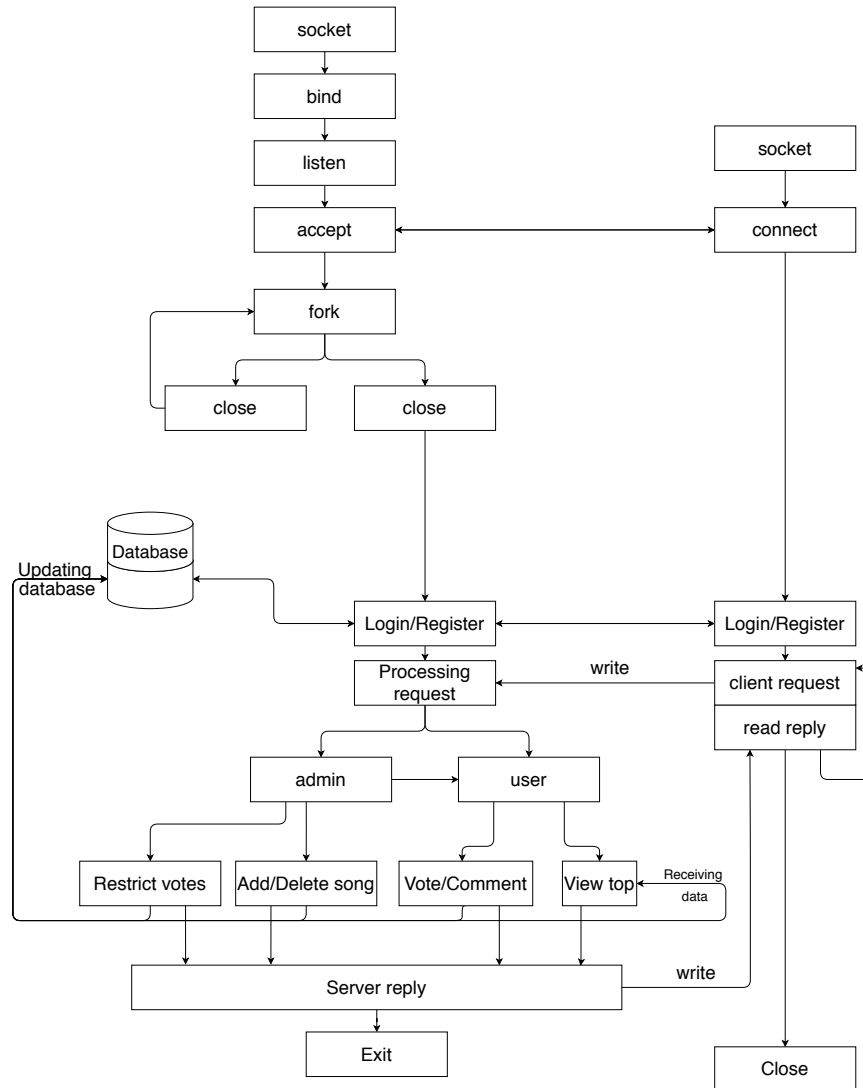
Aplicatia TopMusic foloseste Transmission Control Protocol (TCP) pentru a realiza comunicarea intre utilizator (client) si server. Acest protocol este folosit de obicei de aplicatii care au nevoie de confirmare de primire a datelor. Efectuează o conectare virtuală full duplex între două puncte terminale, fiecare punct fiind definit de către o adresă IP și de către un port TCP [1]. Pentru a nu limita posibilitatea de conectarea a clientilor aplicatia foloseste un modelul de implementare concurent al protocolului TCP.

Protocolul TCP este cea mai potrivita alegere pentru acesta aplicatie deoarece ofera siguranta si asigura transmiterea in ordine a datelor [2], astfel se poate asigura o experienta placuta si completa utilizatorului si exista siguranta ca request-urile facute de catre acesta sunt receptionate de catre server, fiind apoi satisfacute si raspunsurile fiind trimis inapoi clientului in ordinea in care au fost facute.

La cazul general protocolul TCP presupune trei faze. In prima faza, trebuie stabilita conexiunea intre client si server. Dupa ce conexiunea este realizata, se poate face transferul de date, faza in care serverul satisface request-urile clientului. Dupa finalizarea transferului de date, conexiunea este terminata pentru eliberarea resursele hardware/software folosite in proces.

Modelul de implementare concurent al protocolului TCP asigura fiabilitate aplicatiei deoarece face posibila conectarea mai multor clienti simultan si, astfel, poate satisface request-urile fiecarui client independent de ceilalti.

3 Arhitectura aplicatiei



Aplicatia TopMusic este structurata in doua fisiere executabile, clientul si serverul. Clientul asigura interfața de conectare a utilizatorului, unde, după ce se realizează conexiunea cu serverul și autentificarea, acesta poate introduce

request-urile facute catre server. Serverul reprezinta punctul final unde un client (sau mai multi) se poate conecta. Serverul primeste request-ul clientului, il proceseaza si transmite inapoi clientului un raspuns. Procesarea cererilor se face prin extragerea sau inserarea de informatii din baza de date.

Dupa cum se poate observa din diagrama aplicatiei dupa autentificare, vor exista doua tipuri de utilizatori, administratorul si utilizatorul normal. Utilizatorul normal va putea sa voteze o melodie preferata sau sa adauge comentarii acestia sau sa vizualizeze topul curent al tuturor melodiilor sau topurile pe genuri muzicale. Administratorul, implicit va putea face toate operatiile pe care un utilizator normal le poate face, dar in plus poate adauga sau sterge melodii din baza de date (astfel si din topuri) si va putea restrictioana permisiunea de vot al unor utilizatori.

Toate aceste operatii mentionate anterior se realizeaza fie prin extragerea de informatii din baza de date fie prin inserrarea sau actualizarea lor. Aplicatia foloseste SQLite pentru a realiza interogari in baza de date. SQLite este o mică bibliotecă C care implementează un motor de baze de date SQL încapsulat, oferă posibilitatea de a-l introduce în diverse sisteme și necesită zero-configurare[4].

Primitivele de baza folosite in aplicatie sunt:

- `socket()` – creaza un nou punct terminal al conexiunii[3];
- `bind()` ataseaza o adresa locala la un socket[3];
- `listen()` permite unui socket sa accepte conexiuni[3];
- `accept()` blocheaza apelantul pina la sosirea unei cereri deconectare(utilizata de serverul TCP)[3];
- `connect()` tentativa (activa) de stabilire a conexiunii (folosita de clientul TCP)[3]; `file:///C:/Users/Laurentiu/Desktop/Webex.lnk`
- `send()` trimitere de date via socket[3], folosit pentru a trimite requesturile de la client la server si raspunsul catre client;
- `recv()` receptarea de date via socket[3], folosit de serverl pentru a receptiona request-urile clientului si de catre client pentru a primi raspunsul serverului;
- `close()` elibereaza conexiunea (inchide un socket)[3].

4 Detalii de implementare

4.1 Implementare

Aplicatia Topmusic functioneaza pe baza diagramei prezentate anterior. In continuare vom detalia o parte din particularitaile de implementare.

```

case 2:
    if (!logged_in)
    {
        char *err_msg = 0, *user = msg_received + 9, sql[100];
        bzero(sql, 100);
        sprintf(sql, "SELECT username FROM users WHERE username='%s';", user);
        bzero(query_result, 4000);
        return_code = sqlite3_exec(db, sql, callback1, query_result, &err_msg);
        if (return_code != SQLITE_OK)
        {
            sprintf(msg_sent, "SQL error: %s\n", err_msg);
            sqlite3_free(err_msg);
        }
        else if (strstr(query_result, user))
        {
            strcpy(msg_sent, RED "Username already exists! Try logging in ?" RESET);
        }
        else
        {
            bzero(sql, 100);
            sprintf(sql, "INSERT INTO users VALUES ((select max(uid) from users)+1,'%s',0, 1);", user);
            //username=username, type=0 (normal user), vote_permission=1
            bzero(query_result, 4000);
            return_code = sqlite3_exec(db, sql, callback1, query_result, &err_msg);
            if (return_code != SQLITE_OK)
            {
                sprintf(msg_sent, RED "SQL error: %s\n" RESET, err_msg);
                sqlite3_free(err_msg);
            }
            else
            {
                strcpy(msg_sent, GREEN "Registration completed! You can now log in" RESET);
            }
        }
    }
    else
    {
        strcpy(msg_sent, RED "!!! You are already logged in !!!" RESET);
    }
}

```

Inregistrarea utilizatorilor se face prin citirea mesajului de la client (in variabila "message"), si in cazul in care clientul doreste inregistrarea, se va cauta in baza de date username-ul dorit iar daca acesta nu exista se va crea un nou cont, care va fi adaugat in baza de date (cu username-ul dorit,tipul 0 - utilizator normal,permisiunea de vot 1), urmand ca apoi utilizatorul sa se conecteze folosind acel username.

Pentru inregistrarea unui administrator, crearea contului va fi facuta manual de un alt administrator pentru a putea permisiunile specifice. Acest lucru este necesar pentru ca un utilizator normal sa nu poata crea conturi cu permisiuni de administrator.

```

case 1:
    bzero(msg_sent, 4000);
    if (!logged_in)
    {
        char *err_msg = 0, sql[100], *username = msg_received + 6, *id, copy[100], *p;
        bzero(sql, 100);
        bzero(query_result, 4000);
        sprintf(sql, "SELECT uid,username,type,parm FROM users WHERE username='%s'", username);
        int return_code = sqlite3_exec(db, sql, callback1, query_result, &err_msg);

        if (strlen(query_result))
        {
            strcpy(copy, query_result);
            id = strtok(copy, " ");
            uid = atoi(id);
            p = strtok(NULL, "");
            if (return_code != SQLITE_OK)
            {
                sprintf(msg_sent, "SQL error: %s\n", err_msg);
                sqlite3_free(err_msg);
            }
            else
            {
                //succesful login
                strcpy(msg_sent, GREEN "Welcome! # RESET");
                logged_in = 1;
                //permisiuni de admin si de utilizator logat
                if (strstr(query_result, "1 1"))
                {
                    admin = 1;
                    perm = 1;
                }
                else if (strstr(query_result, "1 0"))
                {
                    admin = 1;
                    perm = 0;
                }
                else if (strstr(query_result, "0 1"))
                {
                    admin = 0;
                    perm = 1;
                }
            }
        }
    }
}

```

Autentificarea in aplicatie se face similar cu inregistrarea doar de acesta data nu mai este necesara creare unui cont si astfel este suficient cautarea in baza de date a username-ului folosit, iar daca acesta exista, utilizatorul este autentificat in aplicatie.

```

case 5:
    bzero(msg_sent, 4000);
    if (logged_in)
    {
        bzero(query_result, 4000);
        char sql[100], *err_msg, *song_name;
        song_name = msg_received + 5;
        if (perm == 1)
        {
            //User can vote
            bzero(query_result, 4000);
            bzero(sql, 100);
            sprintf(sql, "SELECT sid FROM songs WHERE name='%s'", song_name);
            return_code = sqlite3_exec(db, sql, callback1, query_result, &err_msg);
            if (return_code != SQLITE_OK)
            {
                sprintf(msg_sent, "SQL error: %s\n", err_msg);
                sqlite3_free(err_msg);
            }
            else
            {
                if (query_result[0])
                {
                    bzero(sql, 100);
                    int n = strlen(query_result);
                    query_result[n] = '\0';
                    sprintf(sql, "UPDATE songs SET nr_votes = nr_votes + 1 WHERE sid = '%s'", query_result);
                    bzero(query_result, 4000);
                    return_code = sqlite3_exec(db, sql, callback1, query_result, &err_msg);
                    if (return_code != SQLITE_OK)
                    {
                        sprintf(msg_sent, "SQL error: %s\n", err_msg);
                        sqlite3_free(err_msg);
                    }
                    else
                        strcpy(msg_sent, GREEN "Succesfully voted #\" RESET);
                }
                else
                {
                    strcpy(msg_sent, YELLOW "Non existing song\" RESET);
                }
            }
        }
        else
            strcpy(msg_sent, YELLOW "You don't have permission to vote!\" RESET);
    }

```

Pentru a vota o anumita melodie utilizatorul trebuie sa stie numele acesteia (lucru posibil prin afisarea topului muzical), urmand ca aceasta sa fie cautata in baza de date iar daca exista, numarul de voturi este incrementat cu 1.

4.2 Scenarii de utilizare

Intr-un scenariu ideal utilizatorul urmeaza toti pasii corect si respecta formatul comenzilor. In practica apar situatii cand utilizatorul nu respecta formatul aplicatiei sau greseste la introducerea unor comenzi, cazuri care daca nu sunt tratate pot duce la functionarea defectuoasa a aplicatiei. In continuare vom expune un scenariu de utilizare normal.

Dupa deschiderea clientului, utilizatorului i se cere sa se autentifice in aplicatie (sau sa se inregistreze daca nu are un cont). Dupa autentificare apar doua situatii posibile:

Autentificarea ca utilizator presupune accesul la comezile de baza ale aplicatiei. Utilizatorul poate folosi comenzile "Vote song", "Comment song [text]" si "view top [genre]" care ii permit fie sa voteze sau sa adauge un comentariu

la o melodie fie sa vada intregul top sau topurile pe genuri ale melodiilor. La procesarea primelor doua comenzi se actualizeaza baza de date, adaugandu-se un vot sau un comentariu pentru o anumita melodie. Pentru cea din urma, se face o interogare in baza de date pentru a extrage topul dorit de utilizator.

Autentificarea ca administrator presupune anumite privilegii pe care un utilizator normal nu le are. Administratorul poate introduce oricare din comenzile unui utilizator normal, dar mai mult, poate restrictiona permisiunea de vot a unui utilizator prin comanda "restrict `username`" (sau poate permite acest lucru, "permit `username`"), iar dupa procesarea acestei comenzi un utilizator nu va mai putea vota, primind mesajul "Permission denied!". De asemenea administratorul poate actualiza topul muzical prin comanda "Add/Delete song" care va actualiza baza de date in mod corespunzator.

Dupa ce utilizatorul sau administratorul termina de folosit aplicatia, acestia pot folosi comanda "quit" care va termina conexiunea clientului cu serverul sau comanda "logout" care va permite reautentificarea in aplicatie cu un alt cont.

O situatie neprevazuta este deconectarea neasteptata a unui client, fapt care nu va influenta negativ aplicatia deoarece utilizatorul poate reporni clientul si in cazul in care este posibila crearea unui conexiuni intre client si server (daca nu exista probleme tehnice), acesta se poate autentifica din nou in aplicatie

5 Concluzii

Prima imbunatatire majora care ar putea fi adusa aplicatiei TopMusic este proiectarea si implementarea unui interfete grafice prietenoasa cu utilizatorul, pentru a utiliza mai usor functionalitatile oferite de aplicatie.

O alta imbunatarie care ar putea fi adusa aplicatiei, este restrictionarea la un singur vot de utilizator pentru fiecare melodie, deoarece in modalitatea de votare curenta, nu exista nicio restrictie asupra numarului de voturi pe melodie, astfel un utilizator poate vota de mai multe ori aceeaasi melodie. Acest lucru se poate realiza prin introducerea unei noi tabele care sa contina username-ul, id-ul melodiei si inca o colona "votat", astfel daca aceasta are valoarea 1 se stie ca un utilizator a votat deja o anumita melodie si nu o va mai putea face o data.

Referinte

1. Transmission Control Protocol, https://ro.wikipedia.org/wiki/Transmission_Control_Protocol
2. Alboaie Lenuta, Programarea In Retea (III), Curs 6 Retele de calculatoare, 11, (2020)
3. Alboaie Lenuta, Programarea In Retea (I), Curs 5 Retele de calculatoare, 14, (2020)
4. SQLite, <https://ro.wikipedia.org/wiki/SQLite>