**TopMusic**

Grițco Sorin

Facultatea de Informatica, Universitatea Alexandru Ioan Cuza, Iași

# Introducere

Am ales să implementez acest proiect cu scopul de a crea o aplicție server/client care să facă managementul unui top muzical, ce conține genuri diverse de muzică. Aplicația va putea fi accesată, doar de utilizatorii înregistrați în sistem, prin intermediul unei parole și login, ce vor fi memorate într-o bază de date. După autentificare, utilizatorii vor putea accesa aplicația prin intermediul logării.

În momentul în care logarea a reușit, utilizatorul va putea introduce la top o nouă melodie, va putea acorda o notă melodiilor, astfel schimbând media aritmetică a notelor și influențând asupra clasamentului melodiilor. Utilizatorul va putea să filtreze melodiile după genul muzical sau/și după note, la fel va putea accesa melodiile prin intermediul unui link pe youtube sau alte site-uri.

Utilizatorii vor putea să posteze comentarii, în secțiunea comentarii ale melodiei. După utilizarea aplicației, ei vor putea să se delogeze. Doar, administratorul va putea șterge melodiile din top și să restricționeze activitatea de votare a utilizatorilor obișnuiți.

# Tehnologii utilizate

Pentru implementarea acestui proiect voi utiliza modelul client/server TCP concurent, cu scopul ca mai mulți utilizatori să poată folosi aplicația în același timp (în mod concurent).

Datele despre utiliztori, login-ul și parola vor fi memorate într-o bază de date MySQL, login-ul va fi în clar(plaintext), iar parola va fi criptată prin intermediul funcției hash SHA-256, pentru a securiza utilizatorii de la o posibilă spargere a bazei de date.

Numele melodiilor vor fi memorate într-o bază de date MySQL, pe lângă numele acestora, vor exista în coloane diferite, genurile aferente, nota generală acestora și link-ul către youtube sau alte site-uri.

Comentariile și genurile muzicale vor fi memorate în tabele diferite, corespondența dintre melodii și genuri se va face pe baza cheilor străine, la fel și pentru comentariile introduse.

# Detalii de implementare

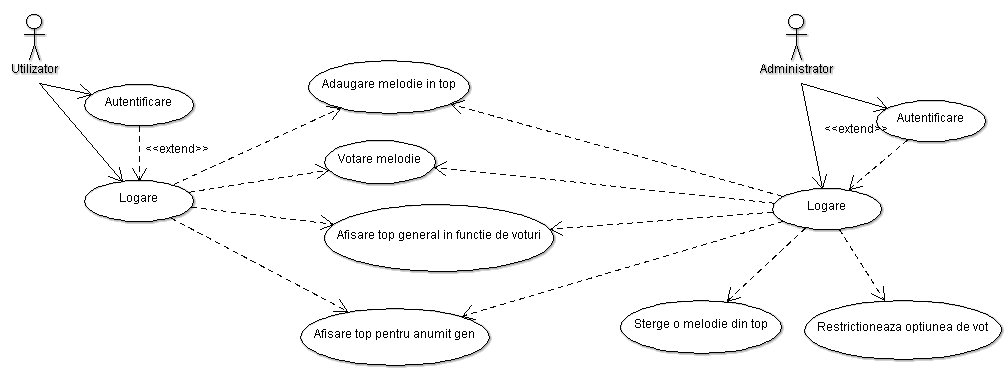
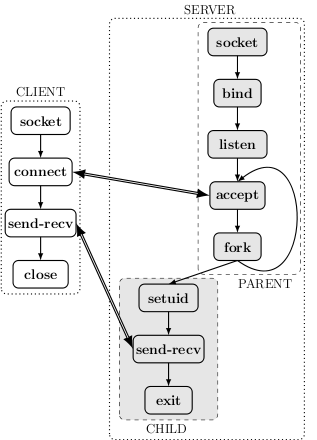


Diagrama Use Case

**Modelul TCP de implementat[1]:**



**Cod relevant:**

1. **Serverul TCP concurrent:**

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <errno.h>

#include <unistd.h>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <sys/wait.h>

/\*port utilizat\*/

#define PORT 2025

extern int errno;

int main(){

struct sockaddr\_in server; //structura folosita de server

struct sockaddr\_in from;

char msg[100]; //mesajul primit de la client

char msgrasp[100] = " "; //mesaj de raspuns pentru client

int sd; //descriptor de socket

/\*crearea unui socket\*/

if ((sd = socket(AF\_INET, SOCK\_STREAM, 0))==-1){

perror("[server] Eroare la socket().\n");

return errno;

}

/\*pregatirea structurilor de date\*/

bzero(&server, sizeof(server));

bzero(&from, sizeof(from));

/\*umplem structura folosita de server\*/

/\*stabilirea familiei de socket-uri\*/

server.sin\_family = AF\_INET;

/\*acceptam orice adresa\*/

server.sin\_addr.s\_addr = htonl(INADDR\_ANY);

/\*utilizam un port utilizator\*/

server.sin\_port = htons(PORT);

/\*atasam socketul\*/

if (bind(sd, (struct sockaddr \*)&server, sizeof(struct sockaddr))==-1){

perror("[server]Eroare la bind().\n");

return errno;

}

if(listen(sd, 5)==-1){

perror("[server]Eroare la listen().\n");

return errno;

}

while(1){

int client;

int child\_pid;

int length = sizeof(from);

printf("[server] Asteptam la portul %d...\n", PORT);

fflush(stdout);

/\*acceptam un client (stare blocanta pana la realizarea conexiunii)\*/

client = accept(sd, (struct sockaddr \*) &from, &length);

/\*eroare la acceptarea conexiunii de la un client\*/

if((child\_pid = fork())<0){

perror("Eroare la fork.");

}

else if(child\_pid==0){ //suntem in copil

close(sd);

if(client < 0){

perror("[server] Eroare la accept().\n");

continue;

}

/\*s-a realizat conexiunea, se asteapta mesajul\*/

bzero(msg, 100);

printf("[server] Asteptam mesajul...\n");

fflush(stdout);

if(read (client, msg, 100)<=0){

perror("[server]Eroare la read() de la client.\n");

close(client); /\*Inchidem conexiunea cu clientul\*/

continue;

}

printf("[server]Mesajul a fost receptionat...%s\n", msg);

/\*pregatim mesajul de raspuns\*/

bzero(msgrasp, 100);

strcat(msgrasp, "Hello ");

strcat(msgrasp, msg);

printf("[server]Trimitem mesaju inapoi...%s\n", msgrasp);

/\*returnam mesajul clientului\*/

if(write(client, msgrasp, 100)<=0){

perror("[server]Eroare la write() catre client.\n");

continue; /\*contiunuam sa ascultam\*/

}

else

printf("[server] Mesajul a fost transmis cu succes.\n");

close(client);

exit(1);

}

wait(NULL); //parinte

}

}

1. **Conectarea la o bază de date MySQL [3]**

/\* Simple C program that connects to MySQL Database server\*/

**#include** <mysql.h>

**#include** <stdio.h>

**main**() {

MYSQL \*conn;

MYSQL\_RES \*res;

MYSQL\_ROW row;

char \*server = "localhost";

char \*user = "root";

char \*password = "PASSWORD"; /\* set me first \*/

char \*database = "mysql";

conn = **mysql\_init**(NULL);

/\* Connect to database \*/

**if** (!**mysql\_real\_connect**(conn, server,

user, password, database, 0, NULL, 0)) {

**fprintf**(stderr, "%s\n", **mysql\_error**(conn));

**exit**(1);

}

/\* send SQL query \*/

**if** (**mysql\_query**(conn, "show tables")) {

**fprintf**(stderr, "%s\n", **mysql\_error**(conn));

**exit**(1);

}

res = **mysql\_use\_result**(conn);

/\* output table name \*/

**printf**("MySQL Tables in mysql database:\n");

**while** ((row = **mysql\_fetch\_row**(res)) != NULL)

**printf**("%s \n", row[0]);

/\* close connection \*/

**mysql\_free\_result**(res);

**mysql\_close**(conn);

}

1. **Criptarea cu SHA256[3]:**

Requires OpenSSL, compile flag: -lssl –lcrypto

#include <stdio.h>  
#include <string.h>  
#include <openssl/sha.h>  
   
int main (void) {  
 const char \*s = "Rosetta code";  
 unsigned char \*d = SHA256(s, [strlen](http://www.opengroup.org/onlinepubs/009695399/functions/strlen.html)(s), 0);  
   
 int i;  
 for (i = 0; i < SHA256\_DIGEST\_LENGTH; i++)  
 [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("%02x", d[i]);  
 [putchar](http://www.opengroup.org/onlinepubs/009695399/functions/putchar.html)('\n');  
   
 return 0;  
}

# Concluzii

Pentru a îmbunătăți soluția propusă, aș putea să utilizez o librărie pentru grafică pentru a impelementa o interfață prietenoasă pentru utilizator.

# 5 Bibiliografie

1. <https://www.usenix.org/legacy/events/sec08/tech/full_papers/radhakrishnan/radhakrishnan_html/node6.html>
2. <https://www.cyberciti.biz/tips/linux-unix-connect-mysql-c-api-program.html>
3. https://rosettacode.org/wiki/SHA-256#C