

# Raport Tehnic pentru Aplicația TopMusic

Hamza Eduard-Miahil

Facultatea de Informatică, Universitatea Alexandru Ioan Cuza, Iași, România

## 1 Introducere

Acest raport își propune să prezinte funcționalitățile aplicației TopMusic precum și modalitatea prin care aceasta a fost implementată. În acest scop pe parcursul lucrării se vor aborda în secțiuni separate **tehnologiile utilizate**, **arhitectura aplicației** și **detalii de implementare**. De asemenea, lucrarea se încheie cu o secțiune de **concluzii** în care se vor prezenta opinii proprii cu privire la modalitățile de îmbunătățire a acestei aplicației.

### 1.1 Descrierea generală a aplicației

TopMusic este o aplicație de tipul client/server pentru managementul a unui top muzical conținând genuri diverse. Aplicația are următoarele funcționalități: înregistrarea utilizatorilor de mai multe tipuri: obișnuiți, administrator; logarea în sistem. Comenzile nu vor putea fi executate dacă utilizatorul nu este autentificat. Operații ce au fost implementate: adăugarea unei melodii la top, votarea unei melodii, afișarea topului general curent în funcție de numărul de voturi, afișarea topului pentru un anumit gen de muzică(ex: dance, rock, hip-hop, etc...). O melodie va avea un nume, o descriere, va aparține unuia sau mai multor genuri de muzică, și va avea asociat un link către videoclipul său pe youtube sau alte site-uri asemănătoare. Utilizatorii autentificați vor putea posta diferite comentarii asociate unei melodii. Administratorul va putea șterge o melodie din top și va putea restricționa opțiunea de vot a unui utilizator.

## 2 Tehnologii utilizate

### 2.1 Protocol de comunicare

TopMusic este o aplicație de tipul client/server. Comunicarea între client și server se realizează prin **socket-uri**. Protocolul de comunicare ales este **TCP** (Transmission Control Protocol). Acesta este un protocol orientat-conexiune, fără pierderi de informații, vizând oferirea calității maxime a serviciilor. Integrează mecanisme de stabilire și eliberare a conexiunii, de detectare și corectare a erorilor și de asigurare a ordinii pachetelor.

Acest protocol a fost ales întrucât este necesară o conexiune de încredere, iar timpul de transmitere al pachetelor în rețea nu este atât de crucial precum integritatea datelor transmise.

## 2.2 Stocare date

Pentru stocarea persistentă a datelor s-a folosit o **bază de date SQLite**, atât pentru stocarea datelor despre utilizatori, cât și pentru stocarea datelor despre melodii. Pentru comunicarea dintre baza de date și aplicație s-a utilizat interfața oficială SQLite (apeluri precum `sqlite3_exec()` și `sqlite3_prepare_v2()`). A fost aleasă această metodă de stocare din motive de scalabilitate și eficiență în comparație cu modul clasic de stocare a datelor în fișiere.

## 2.3 Interfața grafică

Pentru realizarea interfeței grafice s-a folosit **frameworkul QT**. Interfața grafică constă dintr-un ecran de logare (plus opțiuni de creare utilizator) și dintr-un "ecran" principal accesibil după logare. Ecranul principal conține pe mijloc o zonă (de tip scroll area) în care se poate viziona topul, pe laterale butoanele pentru operațiile implementate (adăugare melodie, afișare topuri personalizate, votare, adăugare comentariu, vezi info și logout). În partea stângă este disponibilă o casuță text în care pot fi vazute informații despre o melodie. Fiecare acțiune aplicată pe o melodie (votarea, arată info, sterge, votează, adaugă/vezi comentarii) se va face prin selectarea melodiei din listă și apăsarea butonului corespunzător. Butonul *Adauga melodie* duce la o nouă pagină cu câmpuri pentru introducerea de date despre melodie și selectarea prin check-boxuri a genurilor asociate. Butonul *Comentarii* duce la o nouă pagină cu lista tuturor comentariilor asociate unei melodii și posibilitatea de a adăuga propriul comentariu. De asemenea, dacă utilizatorul este de tip administrator, poate utiliza 2 noi opțiuni: restricționează drept vot, sterge melodie, opțiuni indisponibile pentru utilizatorul obișnuit. În cazul apăsării butonului *Gestionează drepturi de vot*, administratorului îi va apărea o pagină cu lista utilizatorilor non-admin cărora le va putea modifica dreptul de vot.

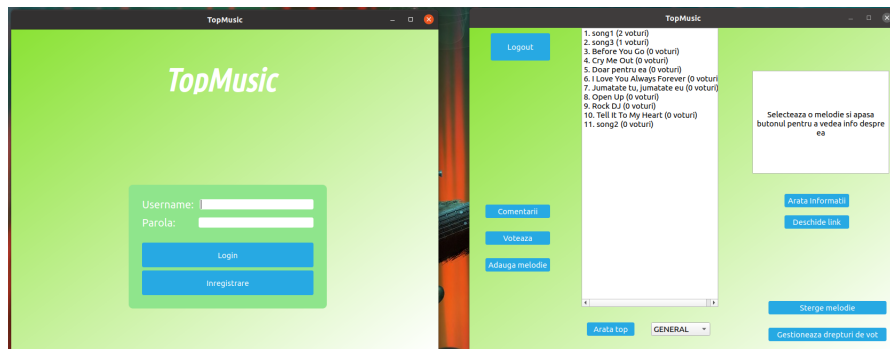
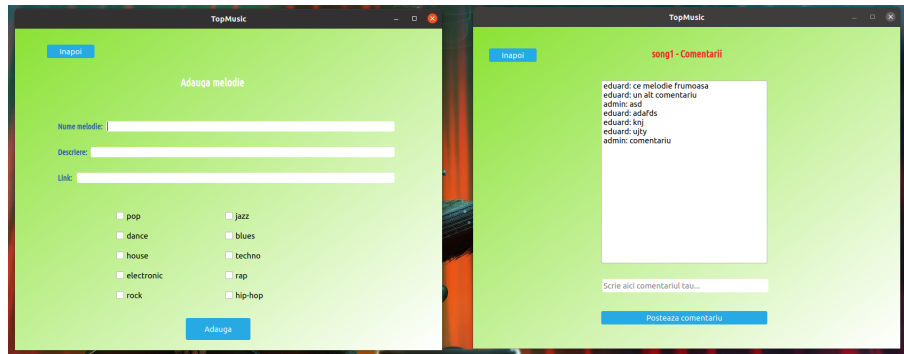
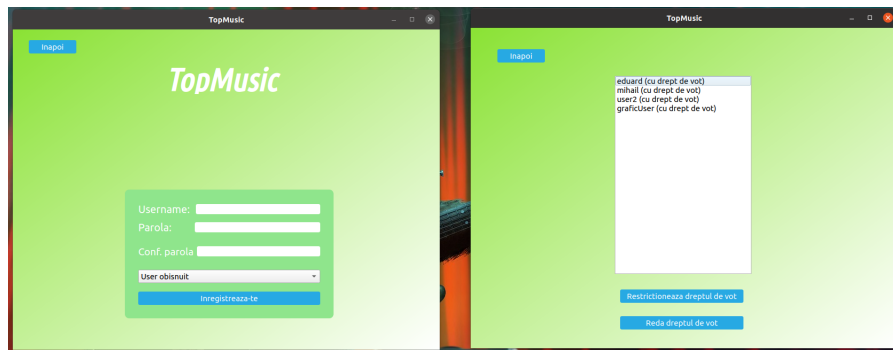


Figura 1. Interfața login și pagina principală



**Figura 2.** Interfata adăugare melodie și vizualizare/adaugare comentarii



**Figura 3.** Interfață înregistrare utilizator nou și restricționare drept vot

## 2.4 Limbaj de programare

Codul sursă al aplicației a fost scris în limbajul C/C++ și compilat cu ajutorul compilatorului GCC.

## 3 Arhitectura aplicației

### 3.1 Diagrama detaliată a aplicației

Aplicația este construită pe baza **modelului client/server concurent**.

**Clientul** are posibilitatea de a se loga în sistem, ca mai apoi să solicite serverului să realizeze diverse operații (votare, adăugare de comentarii,...).

**Serverul** este **implementat** într-o manieră **concurentă**, utilizând apelul de sistem **fork()** pentru a crea câte un proces fiu care să servească fiecare client

conectat. Serverul este cel care realizează operațiile propriu-zise la cererea clientului, dacă acesta are dreptul de a solicita acele operații (privilegii, utilizator logat/nelogat). De asemenea, tot serverul este cel care gestionează baza de date.

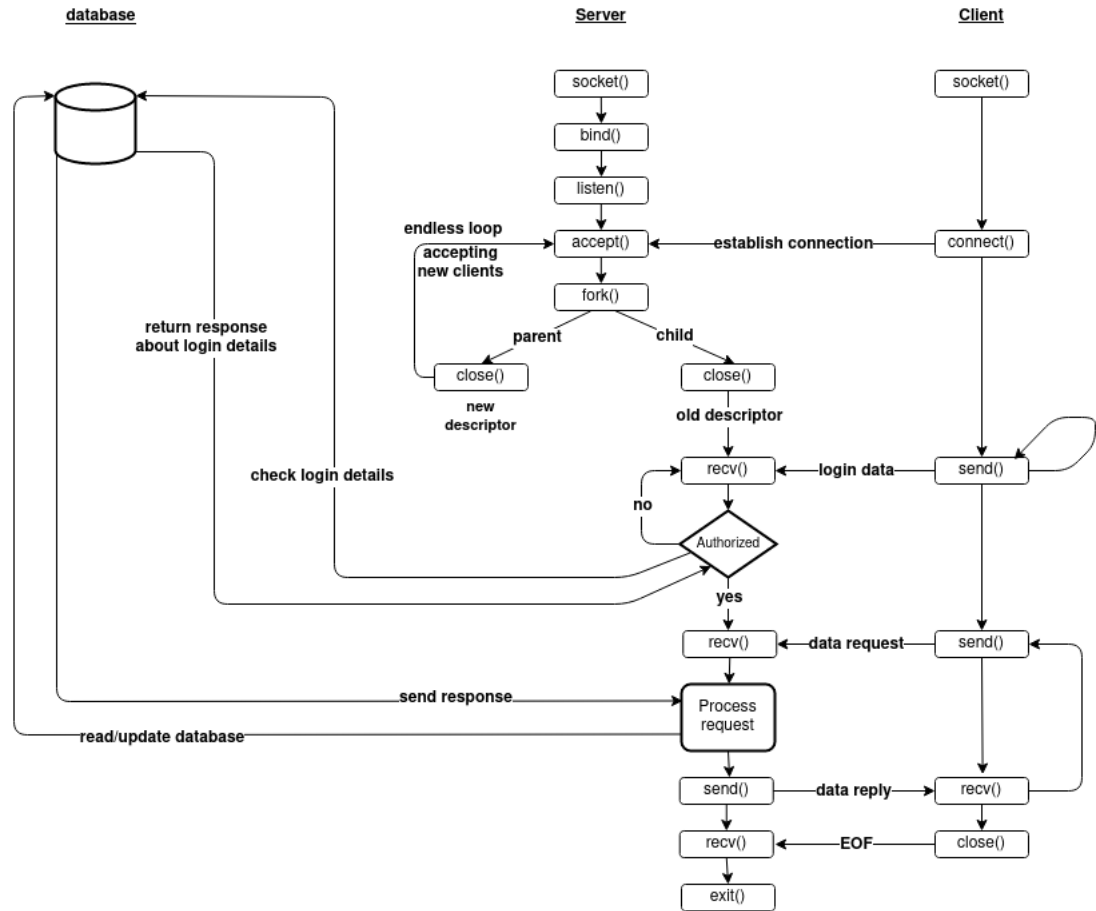


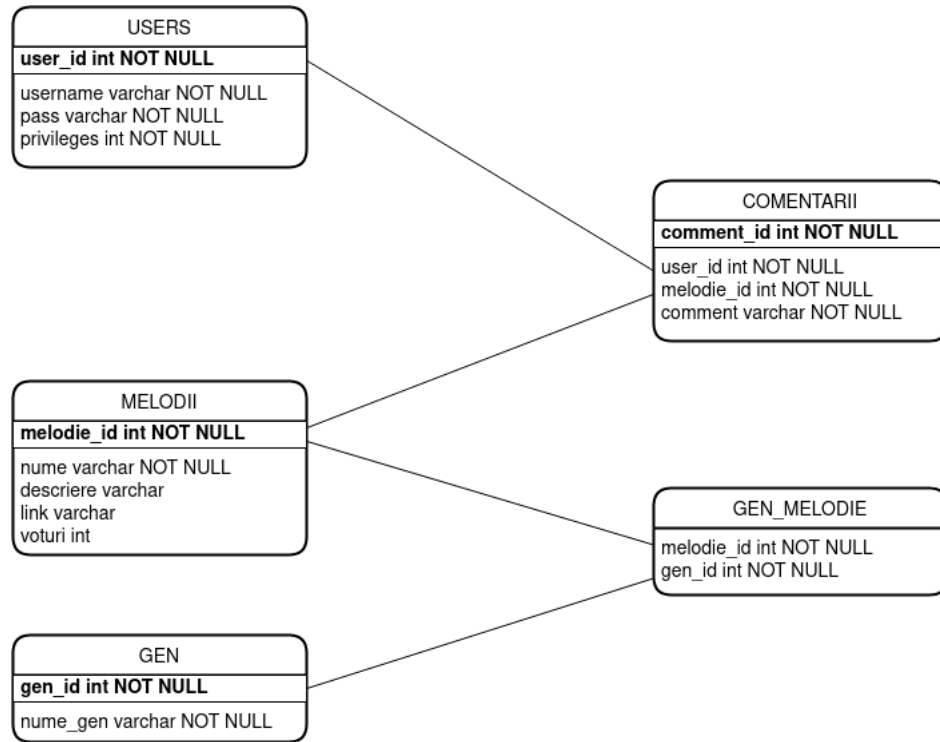
Figura 4. Diagrama detaliată a aplicației

### 3.2 Diagramă bază de date

Baza de date folosită pentru stocarea informațiilor este formată din 5 tabele: **USERS**, **MELODII**, **GEN**, **COMENTARII**, **GEN\_MELODIE**.

Tabela **USERS** stochează date despre utilizator necesare logării în sistem și efectuării operațiilor. Tabela **MELODII** reține numele, descrierea, link-ul și numărul de voturi ale fiecărei melodii. Pe baza *user\_id* și a *melodie\_id* se poate face legătura între melodie, user și comentariile postate de acesta din tabela

**COMENTARII.** Pe baza *melodie\_id* și *gen\_id* se poate asocia fiecărei melodii unul sau mai multe genuri din tabela **GEN\_MELODIE**.



**Figura 5.** Diagrama bazei de date

## 4 Detalii de implementare

### 4.1 Documentare cod

Aplicația construită pe modelul client/server folosește protocolul TCP pentru comunicarea prin socketuri. Serverul este implementat într-un mod concurent folosind apelul de sistem **fork()** pentru a crea câte un fiu pentru fiecare client, în rest schema de comunicare cu clientul fiind ce clasică pentru programarea cu socket-uri (schema prezentată și în Fig. 1). Serverul are rolul de a executa operațiile cerute de clienți și de a comunica cu baza de date. În acest scop s-a încercat o proiectare de tip OPP cu headere și surse în fișiere diferite. Există 3 clase: *auth* (variabile și metode pentru autentificare/creare user, *song* (variabile și metode pentru gestionarea melodiilor și a informațiilor asociate și *DbComm* (implementare de comenzi pentru comunicarea cu baza de date) și un fișier principal care gestionează aceste metode.

De cealaltă parte clientului îi este solicitat să se autentifice, iar după aceea poate solicita serverului executarea operațiunilor amintite în subsecțiunea următoare.

## 4.2 Scenarii de utilizare

În primă fază utilizatorul este rugat să se autentifice, în caz contrar nu are dreptul de a efectua nicio operație. Dacă nu are cont, i se oferă posibilitatea de a-și crea unul.

Dupa autentificare, **utilizatorii obișnuiți** pot adăuga melodii la top, pot vota o melodie sau pot adăuga un comentariu pentru o melodie. De asemenea, pot solicita afișarea topului curent, fie cel general, fie unul pentru un gen muzical specific.

**Utilizatorii cu drept de administrator**, pot efectua toate operațiunile permise unui utilizator obișnuit (de unde si "generalization link" folosit în Fig.3) și în plus pot șterge melodii din top sau restricționa dreptul de vot al unui utilizator.

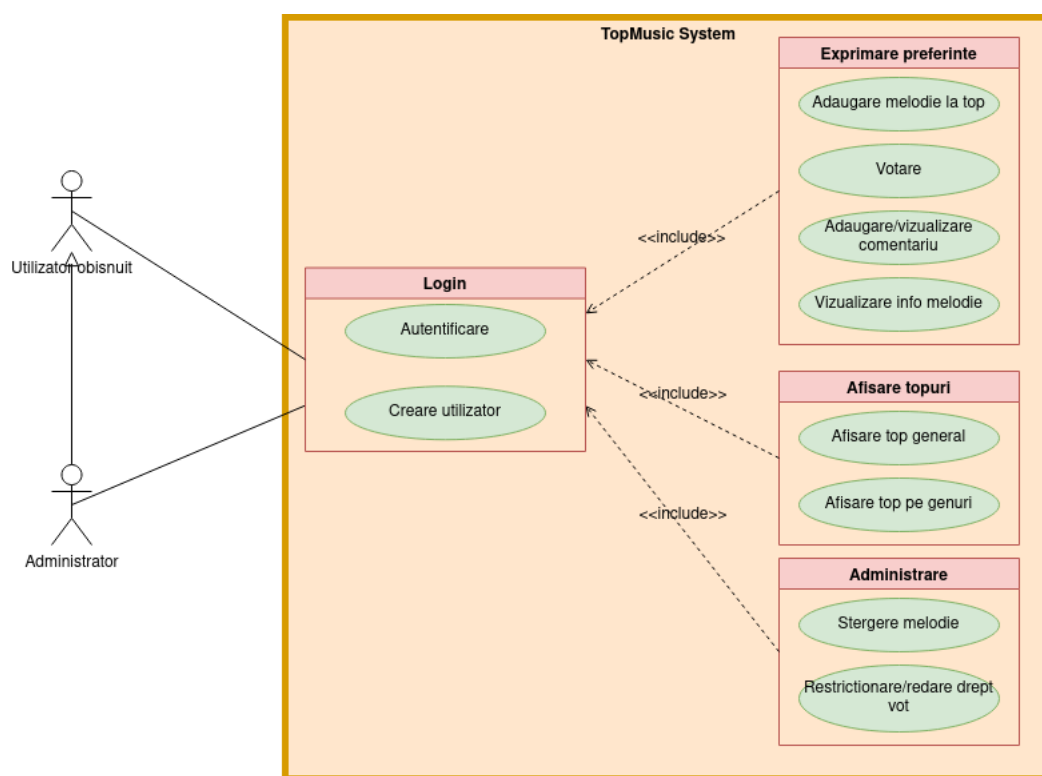


Figura 6. Diagrama use case

## 5 Concluzii

În ceea ce privește **ideile de îmbunătățire generale**, ar fi utile următoarele: adăugarea posibilității de creare și partajare de playlist-uri, modul de recuperare/resetare a parolei, integrarea unui player YouTube pentru ca videoclipurile de la link-urile asociate fiecărei melodii să poată fi vizionate direct în aplicație. Câteva **idei de îmbunătățire a implementării** ar fi: implementarea serverului concurent într-un mod mai eficient decât apelurile **fork()** (thread-uri, select()) și eficientizarea structurii bazei de date.

## Bibliografie

1. Alboaie A. și Panu A. (2020). Computer Networks - Faculty of Computer Science [Online]. Disponibil la <https://profs.info.uaic.ro/computer-networks/cursullaboratorul.php> (Acesat la 10 dec 2020).
2. Vasilache S. (2020). Retele de calculatoare [Online]. Disponibil la <https://sites.google.com/view/fii-rc/home> (Acesat la 13 dec 2020).
3. Mendeley (2017). How To Cite a Website [Online]. Disponibil la <https://www.mendeley.com/guides/web-citation-guide> (Acesat la 13 dec 2020).
4. Mendeley (2017). Harvard Format Citation Guide [Online]. Disponibil la <https://www.mendeley.com/guides/harvard-citation-guide> (Acesat la 13 dec 2020).
5. Springer (2017). Information for Authors of Springer Computer Science Proceedings [Online]. Disponibil la <https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines> (Acesat la 13 dec 2020).
6. Lynch W. (2019). Use Case — Learn By Examples [Online]. Disponibil la <https://medium.com/@warren2lynch/use-case-learn-by-examples-5a63b67fa64d> (Acesat la 13 dec 2020).
7. GeeksforGeeks (2019). TCP Server-Client implementation in C [Online]. Disponibil la <https://www.geeksforgeeks.org/tcp-server-client-implementation-in-c/> (Acesat la 13 dec 2020).
8. Wikipedia (2020). Transmission Control Protocol [Online]. Disponibil la [https://en.wikipedia.org/wiki/Transmission\\_Control\\_Protocol](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) (Acesat la 12 dec 2020).
9. Wikipedia (2020). Applications architecture [Online]. Disponibil la [https://en.wikipedia.org/wiki/Applications\\_architecture](https://en.wikipedia.org/wiki/Applications_architecture) (Acesat la 12 dec 2020).
10. Wikipedia (2020). Software architecture [Online]. Disponibil la [https://en.wikipedia.org/wiki/Software\\_architecture](https://en.wikipedia.org/wiki/Software_architecture) (Acesat la 12 dec 2020).