

Raport Proiect Retele de Calculatoare

Championship

Tema2

Constantinescu M.G. George-Gabriel
Grupa E3, Anul II

10 Decembrie 2020

1 Introducere

Scopul acestui raport este sa prezinte cat mai multe detalii in legatura cu proiectul pe care doresc sa il realizez in cadrul disciplinei Retele de calculatoare[1] din cadrul Facultatii de Informatica, Universitatea Alexandru Ioan Cuza Iasi. Am ales proiectul Championship (B) , un proiect de tipul B.[4]

Acest proiect doreste sa reprezinta o aplicatie de tipul client-server care simuleaza o platforma de tipul ESL(Electronic Sports League) , prin care se pot inregistra anumite campionate de jocuri electronice la care pot participa mai multe echipe pe diferite categorii de jocuri.

Functionalitatile majore ale acestei aplicatii sunt:

- inregistrarea utilizatorilor(atat cei obisnuiti cat si administratorii);
- utilizatorii pot vizualiza informatii despre campionate si se pot inregistra daca doresc;
- inregistrarea unui campionat;
- specificarea pentru fiecare campionat a jocului, numarul de jucatori, diferite reguli si structura campionatului(meciuri de tipul BO1 sau BO2 etc.);
- modul de extragere al partidelor;
- inregistrarea unei echipe sau unui jucator intr-un campionat, ulterior acesta fiind instiinat prin email ca a facut acest lucru si daca a fost acceptat sau nu;
- jucatorul va primi informatii aditionale via email despre partidele sale (ora, adversarul, s.a.m.d);
- jucatorul are posibilitatea de a reprograma o partida;
- administratorii pot vedea un istoric al partidelor jucate;

2 Motivatie

Motivatia pentru care a fost ales acest proiect este aceea ca sunt obisnuit sa interactionez cu o aplicatie de tipul acesta. In trecut, am participat la destul de multe campionate de Dota2 alaturi de o echipa internationala si am o idee generala asupra acestui tip de aplicatie.

Astfel, sunt "obisnuit" cu functionalitatile acestui tip de proiect: inregistrarea unei echipe, participarea la un anumit campionat, setarea datei si orei inceperii unei partide s.a.m.d .

3 Tehnologii utilizate in aplicatie

Pentru realizarea acestui proiect si a conexiunii dintre client si server voi folosi un protocol de comunicare de tipul TCP/IP concurent, astfel incat sa se poate conecta la server in acelasi timp cat mai multi utilizatori. [2][3]

Motivatie pentru care am ales acest tip de comunicare este ca TCP/IP ofera o viteza destul de buna pentru schimbul de informatii(este "reliable"), acest aspect fiind foarte important intr-o astfel de aplicatie deoarece daca ne-am pune in ipostaza jucatorilor, dorim sa obtinem informatii cat mai rapid in legatura cu anumite campionate in care ne-am inregistrat.

Odata ce un client se va conecta la serverul aplicatiei, pentru acesta se va crea un thread cu un id specific, unde serverul asteapta anumite comenzi specifice aplicatiei de la client, le prelucreaza si ofera inapoi un raspuns pe baza comenzii pe care a introdus-o utilizatorul.[2][3]

4 Arhitectura aplicatiei

Pentru a oferi o experienta de utilizare cat mai buna dar si o implementare cat mai eficienta, in proiect se va realiza o legatura foarte stransa intre o baza de date in SQLITE si programul in sine. In baze de date voi stoca mai multe tabele care vor contine urmatoarele:

- o tabela va contine username-ul utilizatorului, tipul de utilizator(daca este simplu jucator sau administrator), email-ul fiecaruia dar si o parola;
- o tabela va contine numele si data de incepere a fiecarui campionat, jocul asociat campionatului, numarul de jucatori, diferite reguli organizatorice si modul de extragere al partidelor;
- o tabela va contine echipele inscrise la anumite campionate;
- o tabela va contine istoricul partidelor jucate si scorul aferent;

In functie de alte aspecte ale aplicatiei, o sa se implementeze si alte tabele cu scopul de a face o experienta de utilizare cat mai placuta si eficienta (de exemplu: o tabela cu componenta echipelor s.a.m.d.).

Astfel, in functiei de fiecare comanda care va fi introdusa de fiecare utilizator in parte, se va returna raspunsul care corespunde interogarii specifice in limbajul SQL asociate comenzii introduse.

O situatie concreta: un jucator cu username-ul X doreste sa se conecteze la aplicatie. Acesta cand va utiliza comanda de login, aplicatia va cauta in baza de date, in tabela cu toti utilizatori inregistrati numele si parola acestuia. Daca le gaseste, il va lasa sa foloseasca si celelalte comenzi; Daca nu, va returna un mesaj de tipul: "username sau parola gresite / cont neinregistrat".

Mai jos se pot vedea mai multe diagrame in care se prezinta arhitectura aplicatiei.

Diagrama pentru inregistrare

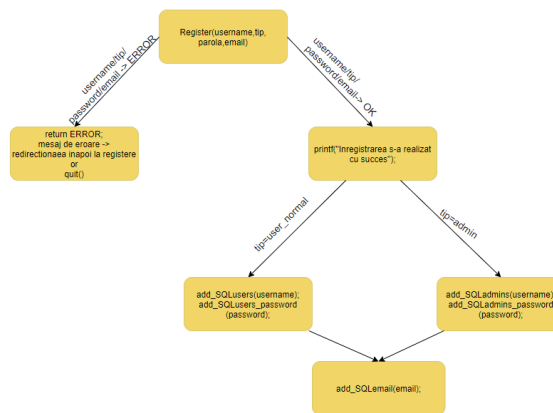


Diagrama pentru logare

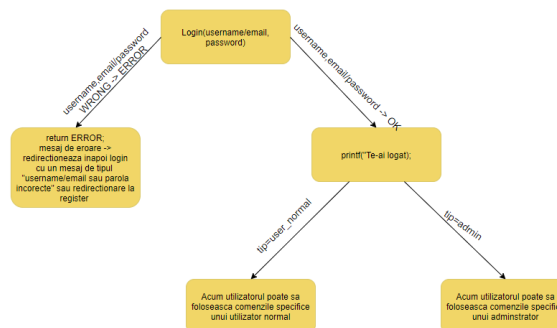


Diagrama comenzilor utilizatorului obisnuit

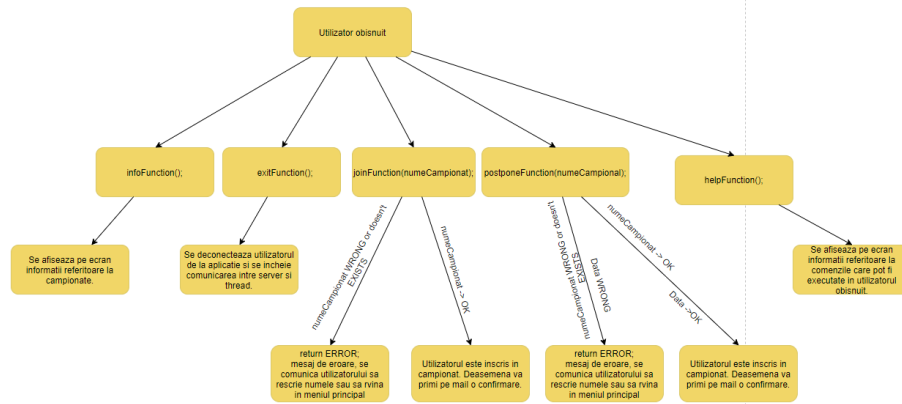
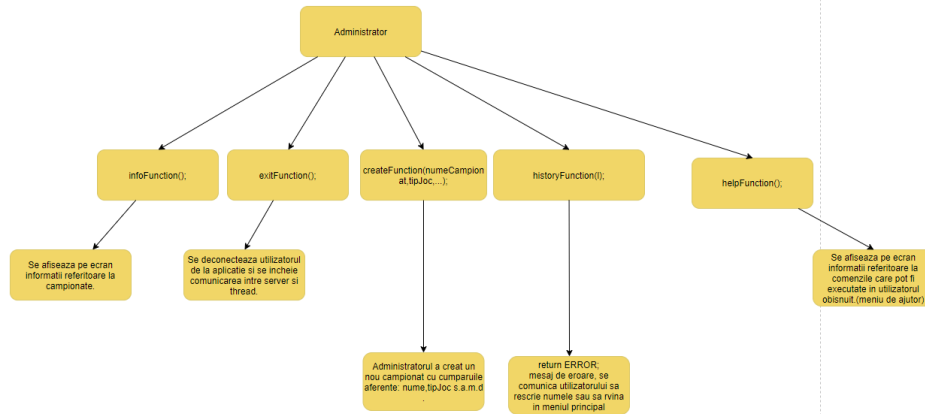
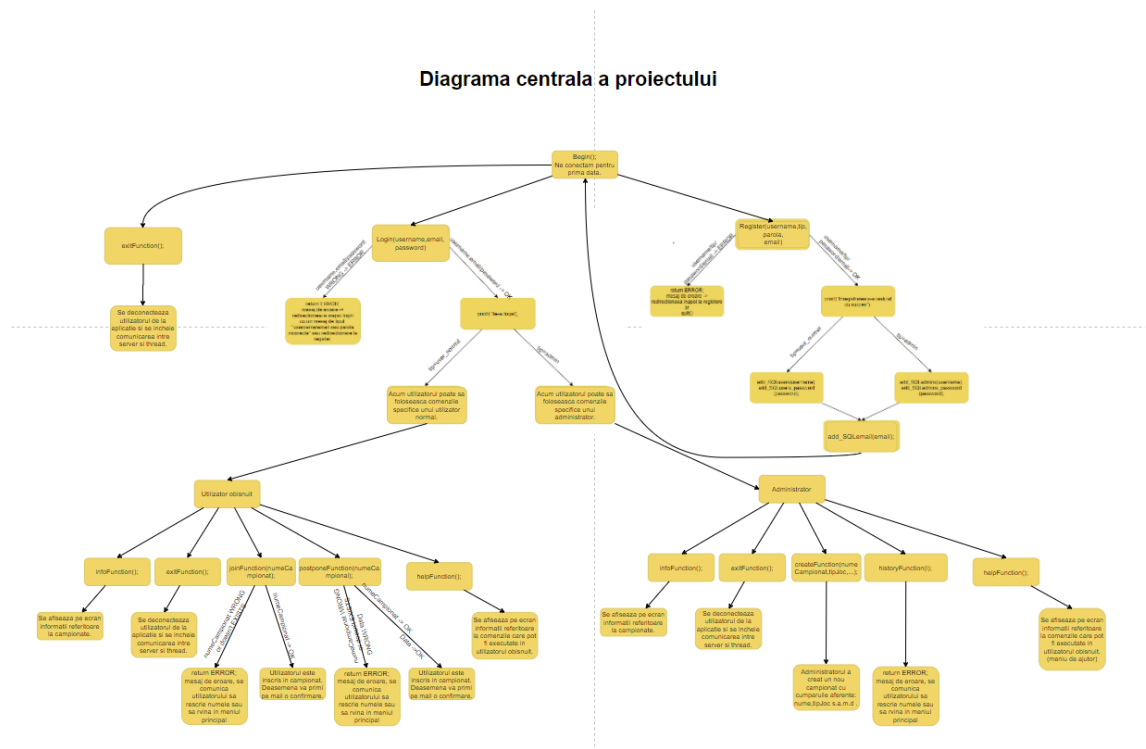


Diagrama comenzilor administratorilor





Observatie: Aceste diagrame nu reprezinta varianta finala a proiectului, ulterior voi adauga si alte functionalitati aplicatiei.

5 Detalii de implementare

Detaliile de implementare sunt intr-o stransa legatura cu diagramele de mai sus.

Pentru stocarea informatiilor am construit in SQLITE o baza de date formata din mai multe tabele, fiecare pentru o categorie specifica: utilizatori existenti, istoric al meciurilor, istoric al campionatelelor, s.a.m.d .

5.1 Functia de inregistrare

Scopul functiei de inregistrare este de a inregistra un utilizator in cadrul aplicatiei, fie ca acesta este un utilizator obisnuit sau un administrator in functie de anumiti parametrii amintiti in diagramele de mai sus: username, tipul de utilizator, parola si email-ul. Ulterior aceste informatii vor fi stocate in tabela SQL numita inregistrareSQL. In caz ca username-ul/email-ul se regasesc deja in tabela sau tipul de utilizator este unul invalid, programul afiseaza un mesaj de eroare, care il indruma pe utilizator in 2 directii: ori sa se inregistreze cu alte date ori/sa se logheze cu datele deja existente sau sa iasa din aplicatie.

5.2 Functia de logare

Scopul functiei de login este de a i se permite unui utilizator care este deja inregistrat in baza de date sa se conecteze la aplicatie cu contul sau. In caz ca username-ul/email-ul sau parola sunt incorecte, programul afiseaza un mesaj de eroare, care il indruma pe utilizator sa reintroduca din nou datele de conectare cu atentie, ori sa isi creeze un cont in caz ca nu are unul sau sa iasa din aplicatie.

Odata ce utilizatorul s-a conectat cu contul sau la aplicatie, in acest moment ii este permis sa foloseasca si restul comenzilor in functie de tipul e utilizator(utilizator obisnuit sau admin).

5.3 Functia de inregistrare a unui campionat

Scopul acestei comenzi este de a crea un nou campionat la care pot participa utilizatori obisnuiti. Aceasta comanda poate fi utilizata doar de administratori, ei avand drepturi pentru acest lucru. Odata cu apelul functiei, administratorul trebuie deasemenea sa precizeze si detalii referitoare la acest campionat: specificarea jocului, a numarului de jucatori, diferite reguli, structura campionatului s.a.m.d. .

5.4 Functia de inregistrare a unui utilizator obisnuit/a unei echipe intr-un campionat

Scopul acestei comenzi este de a inregistra un utilizator obisnuit intr-un campionat deja existent. Aceasta comanda poate fi utilizata doar de utilizatori obisnuiti pentru a se inscrie pe ei insasi. Odata ce este inscris, utilizatorul va fi instiintat via email daca a fost acceptat pentru acest campionat si va primi informatii aditionale despre campionatul in care a fost inscris. In caz ca numele campionatului este introdus gresit sau campionatul nu exista, programul va afisa un mesaj de eroare ce indruma utilizatorul sa introduca din nou numele campionatului sau sa revina in meniul principal.

5.5 Functia de reprogramare a unei sesiuni de joc

Scopul acestei comenzi este de a reprograma o partida de joc in caz ca utilizatorul nu poate participa la aceasta. Aceasta comanda poate fi folosita doar de utilizatori obisnuiti si astfel le este permis sa amane o anumita partida. In caz ca numele campionatului in care are loc partida este scris gresit sau nu exista, programul va afisa un mesaj de eroare ce indruma utilizatorul sa introduca din nou numele campionatului sau sa revina in meniul principal. In caz ca noua data la care este programata partida este neconforma, deasemenea programul va afisa un mesaj de eroare ce indruma utilizatorul sa introduca din nou datele pentru reprogramare sau sa revina in meniul principal.

5.6 Functia de afisare a unui istoric al partidelor

Scopul acestei comenzi este de a se afisa pe ecran un istoric al partidelor deja terminate intre jucatorii obisnuiti. Aceasta comanda poate fi folosita doar de administratori.

5.7 Functia de afisare a informatiilor despre campionate recente

Scopul acestei comenzi este de a se afisa informatii relevante despre anumite campionate: numele campionatului, tipul de joc, diferite reguli, structura campionatului etc. . Aceasta comanda poate fi utilizata atat de utilizatorii obisnuiti cat si de administratori. Ulterior, utilizatorii obisnuiti care sunt interesati de un anumit campionat, se pot inscrie si pot participa la acesta.

5.8 Functia de afisare a meniului de ajutor

Scopul acestei comenzi este de a afisa informatii referitoare la comenzile care pot fi introduse in functie de tipul de utilizator. Aceasta comanda poate fi utilizata atat de utilizatorii obisnuiti cat si de administratori. Aceasta comanda poate fi utilizata doar in momentul in care utilizatorul se afla in meniul principal.

5.9 Functia de iesire din aplicatie

Scopul acestei comenzi este de a deconecta utilizatorul de la aplicatie, astfel se simuleaza o comanda de tipul "logout". Aceasta comanda poate fi utilizata atat de utilizatorii obisnuiti cat si de administratori. In momentul in care aceasta comanda este utilizata, conexiunea dintre serverul aplicatiei si thread-ul specific fiecarui client este incetata.

5.10 Observatii

Pe langa comenzile specificate mai sus, pe parcursul implementarii acestui proiect se vor mai adauga functii suplimentare care sa eficientizeze experienta de utilizare a acestei aplicatii. Ca de exemplu, in locul inregistrarii unui jucator la un campionat, sa se permita inregistrarea unei echipe cu mai multi jucatori, iar pentru fiecare dintre acestia sa se afiseze informatii referitoare la strategii de joc preferate dar si un rating de tipul Win-Lose, astfel putand sa se realizeze o ierarhie a jucatorilor/echipelor in functie de meciurile pierdute si castigate.

Ca exemplu, in poza de mai jos este exemplificat modul cum ar trebui sa functioneze comanda de login:

```
1  interogareSQL_nickname=
2      "SELECT username FROM users
3          WHERE username=nickname";
4  interogareSQL_password=
5      "SELECT username FROM users
6          WHERE username=nickname";
7  if(nickname=interogareSQL_nickname) then
8      if(password=interogareSQL_password) then
9          printf("Felicitari! Ati fost logat."); //de aici se pot utiliza
10     else                                     //comenzile specifice utilizatorului
11         printf("Ne pare rau. Nickname sau parola gresita.");
12         //redirectare la login/register sau iesire din aplicatie
```

6 Concluzii

La nivelul aplicatiei, se pot realiza anumite optimizari astfel incat experienta de utilizare a aplicatiei, atat in ipostaza unui utilizator obisnuit(jucator) cat si din cea a unui administrator sa fie una mai placuta.

De exemplu, in cazul repogramarii unei partide, nu doar jucatorul care doreste sa o reprogrameze trebuie sa fie de acord. Stim ca intr-un meci de tipul acesta, exista cel putin un adversar iar noi nu stim daca acesta va fi de acord cu mutarea datei partidei.

Astfel, o posibila imbunatatire ar putea fi constituita dintr-un sistem de checking astfel: jucatorul/echipa care doreste sa reprogrameze un meci va cere acest lucru unui administrator cu o noua data a partidei.

Administratorul are datoria de a transmite adversarilor acest lucru si datele legate de noua zi in care se va desfasura partida. Acestia pot decide daca sunt de acord sau nu cu modificarea. In caz ca sunt de acord, partida se va reprograma cu noua data. In caz ca nu, adeversarii propun o noua data care va fi transmisa prin administrator la jucatorul/echipa initiala.

In caz ca nici in acest moment nu se poate realiza o amanare a partidei, o fereastra de chat se va deschide intre cei doi jucatori/capitani de echipa a celor doua echipe in care vor discuta despre o noua data a disputarii meciului. In caz ca nu se poate realiza acest lucru, administratorul va oferi castig de cauza echipei adverse si meciul va fi declarat anulat prin neparticiparea primei echipe.

O imbunatatire la nivelul inregistrarii administratorilor este ca inregistrarea acestora sa nu se poate face de catre un utilizator obisnuit, ci de un administrator care este deja conectat la serverul aplicatiei. Astfel putem evita acel caz atunci cand un utilizator obisnuit ar putea produce haos in aplicatia noastra si ar putea corupe informatii referitoare la scorurile partidelor sau la campionate.

O alta imbunatatire majora care ar putea duce la sporirea functionalitatii acestei aplicatii este ca atat administratorii cat si ceilalti jucatori din acel campionat sa fie notificati via email atunci cand o partida este incheiata. In mail sa se

regaseasca numele jucatorilor/echipelor care au participat la meci, campionatul cu toate detaliile aferente si scorul partidei.

O imbunatatire la nivelul aplicatiei o poate constitui si schimbarea protocolului de transmitere a informatiilor. In loc e TCP/IP, pentru o viteza mai buna se poate folosi UDP/IP dar in acest caz stim ca retransmiterea pachetelor de date pierdute nu este posibila ca la TCP/IP.[6]

7 Bibliografie

- [1]Site-ul disciplinei
- [2]Programul de plecare pentru server.c
- [3]Programul de plecare pentru client.c
- [4]Lista proiectelor
- [5]Platforma ESL - Electronic Sports League
- [6]TCP vs. UDP
- [7]SQLite Website
- [8]How to install SQLite - fatal error
- [9]Tutorial SQLite
- [10]SQLite C Tutorial
- [11]How to design the callback function regarding SQLite communication with the server + others things
- [11]Site-ul laboratorului BD
- [12]Site for drawing diagrams
- [12]How to color lines in linux terminal