

# Championship (B) - Raport Tema 2

Zamfir Adrian-Iulian  
Anul II Grupa A5

Decembrie 2020

## 1 Introducere

Asa cum apare si in titlu, raportul se bazeaza pe un proiect de tip B, anume "Championship".

Se cere implementarea unei aplicatii de tip client-server prin care sa se simuleze administrarea unei platforme de tip ESL, unde se pot crea campionate pentru diferite tipuri de jocuri video.

Vor exista doua tipuri de useri:

- useri de tip "**normal**", capabili sa obtina informatii despre orice campionat (disponibil), sa se inscrie intr-un capionat (disponibil) si ulterior sa primeasca un email prin care sunt instiintati daca au fost acceptati in campionat sau nu.
- useri de tip "**administratori**", capabili sa creeze un capionat, sa baneze un user normal, sa obtina un istoric al partidelor dintr-un campionat sau sa stearga un capionat ( care practic a fost anulat din diverse motive ).

## 2 Tehnologii utilizate

Se foloseste modelul TCP/IP concurent pentru conexiunea client-server.

Pentru fiecare client conectat, simultan, serverul creaza cate un thread, unde se va astepta comenzile date de client. Threadul trimite un raspuns inapoi clientului, in functie de comanda data si de tipul userului, astfel un user de tip "normal" nu poate accesa o comanda specifica unui client de tip "administrator" ( si nici invers ). Chiar daca in timpul servirii unui client, se conecteaza un alt client, cei doi nu se pot influenta reciproc datorita detasarii threadului.

Motivul central pentru care am ales sa utilizez un model TCP/IP este asigurarea integritatii datelor, ceea ce in modelul UDP nu este garantat. Cu toate ca un server de tip UDP ar fi mult mai rapid, integritatea datelor este mult mai prioritara fata de o simpla viteza de rulare si prelucrare.

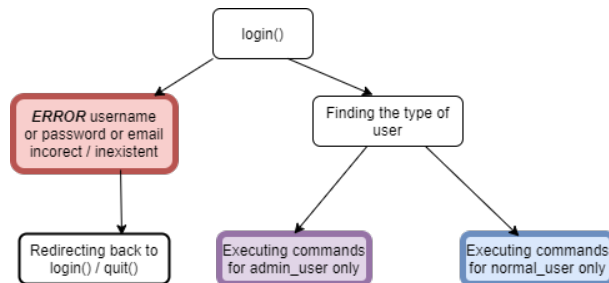
### 3 Arhitectura Aplicatiei

Se va folosi o legatura dintre programul in c si o baza de date realizata in SQLITE. Pentru a utiliza SQLITE este nevoie de a se descarca librariile aferente.

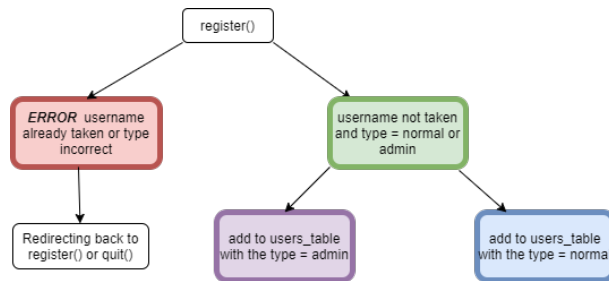
In mare parte, in SQL se vor stoca mai multe tabele in care se regasesc diverse informatii, precum in tabela "USERS" unde se regasesc Username, Parola si Emailul aferent, sau tabelul "CHAMPIONSHIP" unde se regasesc informatiile despre championate precum Nume-Championat, Data, Ora, etc...

In continuare se regasesc diagramele prin care se demonstreaza functionalitatea codului:

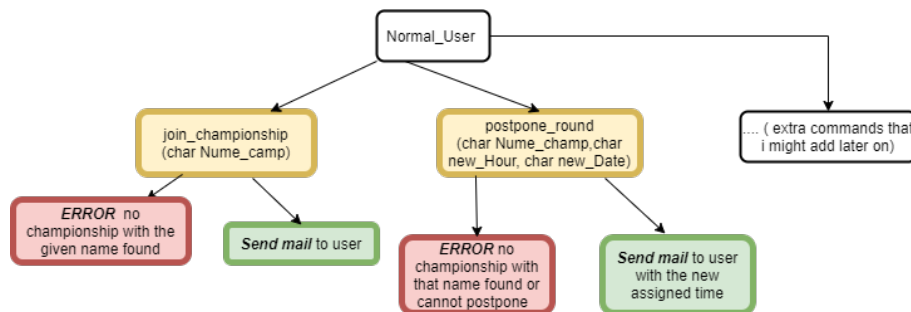
#### DIAGRAMA LOGIN



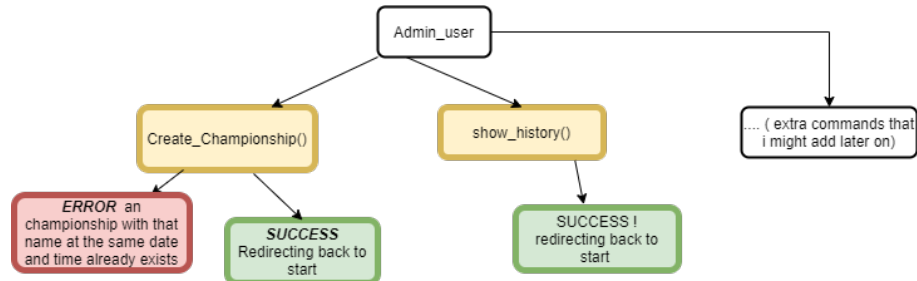
#### DIAGRAMA REGISTER



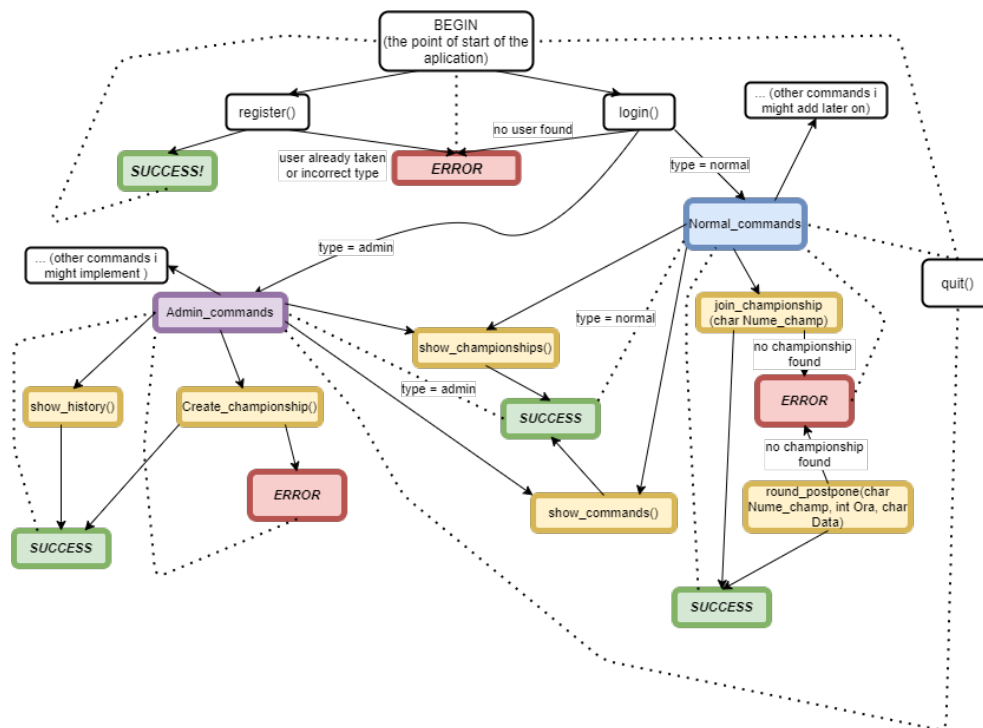
#### DIAGRAMA COMENZI UNICE PENTRU NORMAL USER



## DIAGRAMA COMENZI UNICE PENTRU ADMIN USER



## DIAGRAMA INTREGULUI PROGRAM



## 4 Detalii de implementare

Precum am specificat mai sus, ma folosesc de o baza de date din SQL pentru memorarea informatiilor, deci, fiecare comanda va apela o interogare specifica din SQL.

#### 4.1 Comanda "login"

Permite utilizatorului sa se conecteze cu un cont deja creat de el. Daca username / parola / email-ul sunt introduse gresit, se afiseaza un mesaj de eroare apoi clientul este redirectionat inapoi la inceput, pentru a putea sa incerce register(), login() (din nou) sau sa iasa complet din program folosind quit().

#### 4.2 Comanda "register"

Permite utilizatorului inregistrarea unui nou cont, introducand un username, o parola, si un email. Daca se dovedeste ca email-ul sau username-ul este deja inregistrate, clientul va fi redirectionat catre inceput, unde poate sa incerce login(), register() (din nou) sau sa iasa complet din program folosind quit().

#### 4.3 Comanda "show commands"

Permite utilizatorului sa vada lista intreaga de comenzi posibile ( se vor vedea toate comenzile inclusiv cele specifice administratorului sau userului normal ). Incercarea de a introduce o comanda inexistentă sau o comanda care nu este specifica tipului de user logat, va rezulta in afisarea unei erori, apoi redirectionarea catre inceput unde ii este data sansa de a introduce o alta comanda.

#### 4.4 Comanda "show championships"

Este destul de evident, permite vizualizarea informatiilor tuturor campionatelor precum numele campionatului, tipul de joc , regulile specifice etc... . Este imposibil sa dea o eroare aici, chiar daca nu exista campionate, programul afiseaza simplu nimic.

#### 4.5 Comenzile "join championship" sau "postpone"

Comanda "join championship" permite inregistrarea la un campionat ales de client. Evident, daca se inceara aplicarea la un campionat inexistent, se va afisa eroare apoi se redirectioneaza inapoi la locul de unde poate incerca noi comenzi sau la parasirea programului.

Similar, comanda "postpone" permite schimbarea orei si a datei unui campionat. Daca se incearca postpone-ul unui campionat inexistent sau postpone-ul nu a fost autorizat se va afisa un mesaj de eroare specific apoi se redirectioneaza inapoi la introducerea unei noi comenzi sau parasirea programului.

**COMENZILE ACESTEA SUNT SPECIFICE UTILIZATORULUI NORMAL**

## 4.6 Comenzile "show history" sau "Create championship"

Comanda "show history" permite vizualizarea scorurilor tuturor partidelor campionatelor. Este practic imposibil sa aiba vreo eroare aici, chiar daca nu a avut loc niciun campionat pana acum, programul practic va afisa nimic.

Comanda "Create championship" permite crearea unui nou campionat. Daca se incearca crearea unui campionat cu un acelasi nume, data si ora fata de un alt campionat deja creat, programul afiseaza o eroare apoi redirectioneaza inapoi la introducerea unei noi comenzi sau la oprirea programului.

**COMENZILE ACESTEA SUNT SPECIFICE ADMINISTRATORULUI**

## 5 Concluzii

Exista multe posibilitati de a optimiza rulara serverului, precum alegerea unui model UDP peste modelul TCP, dar cu riscul de a pierde informatii. Printre alte optimizari s-ar gasi implemetarea mai eficienta a functiei "join championship" sau adaugarea unor functii suplimentare care sa ajute la optimizarea rezultatelor.

## 6 Bibliografie

---

PLM

---