

Championship

Radu Damian

¹ Universitatea "Alexandru Ioan Cuza", Iași, România

² Facultatea de Informatică

Abstract. Acest raport constituie o documentație a proiectului *Championship*, care constă în crearea unui mediu virtual prin care utilizatorii pot crea sau/și participa la turnee destinate în mare parte gaming-ului(e.g. jocuri eSports) dar nu numai.

Se vor regăsi următoarele secțiuni: **Introducere** - se va prezenta pe scurt proiectul cu funcționalitățile aferente , **Tehnologii utilizate** - vor fi menționate tehnologiile utilizate în realizarea proiectului și motivația alegerii acestora, **Arhitectura aplicației** - vor fi prezentate conceptele de funcționare a aplicației, precum și o diagramă a aplicației, **Detalii de implementare** - vor fi prezentate mostre de cod relevante proiectului, precum și scenarii de utilizare, **Concluzii** - reflectări asupra modului în care ar putea fi îmbunătățită aplicația.

1 Introducere

Scopul acestei aplicații este de a administra diverse campionate. Prima funcționalitate necesară este **înregistrarea utilizatorilor**, iar următoarele funcționalități menționate vor fi restricționate de **secțiunea de logare**: înregistrarea unui campionat, specificarea jocului, a numărului de jucători, diferite reguli sau structuri de campionat, înregistrarea unui utilizator într-un campionat.

2 Tehnologii utilizate

Pentru implementarea comunicării între clienți și server s-a utilizat un server TCP concurent utilizând fork-uri. Motivarea acestei decizii este reprezentată de faptul că informațiile transmise între client și server nu trebuie să fie alterate, ele trebuie să-și păstreze caracterul veridic, deoarece natura aplicației impune ca informațiile să fie corecte pentru ca campionatele înregistrate să se desfășoare în mod optim. Pe de altă parte, funcționalitățile oferite de aplicație nu ar impune implementarea unui server cu caracteristici mai complexe.

Interfața grafică a fost implementată prin intermediul utilitarului Qt [7], iar datele au fost stocate utilizând fișiere XML. Procesarea acestora a fost realizată prin intermediul librăriilor oferite de Qt, mai concret QDomStreamReader(pentru citirea din fișiere XML), respectiv QDom(e.g. QDomElement, QDomDocument) pentru actualizarea acestor fișiere. [8] [9]

3 Arhitectura aplicației

Având în vedere funcționalitățile aplicației, acestea au fost numite în mod convențional în felul următor: create, login, recordChamp, setGame, setNrPlayers, setRules, setDraw, recordPlayer, reprogramMatch, quit.

Totuși, în timpul execuției aplicației, aceste funcționalități sunt "mascate" prin intermediul elementelor grafice sugestive(utilizatorul nu trebuie să scrie explicit în linia de comandă funcționalitatea dorită), făcând aplicația user-friendly.

Pe de altă parte, serverul va verifica prin intermediul informațiilor stocate dacă poate satisface cerințele clientului(e.g. înregistrarea unui utilizator cu un anumit nume, crearea unui campionat) și vor fi afișate mesaje corespunzătoare.

Arhitectura aplicației este următoarea:

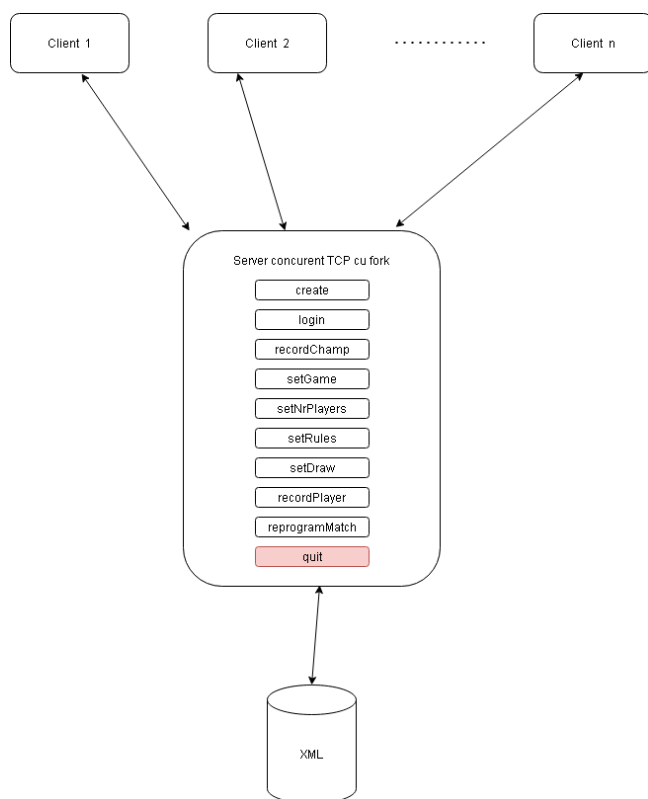


Figura 1. Diagrama aplicației

4 Detalii de implementare

În cele ce urmează vor fi ilustrate secvențe de cod(adaptate prin intermediul Qt) legate de modul în care se realizează conexiunea între server și clienți, dar și

comunicarea acestora, precum și o diagramă sugestivă cu scenariile de utilizare ale aplicației.

Client

```
QApplication a(argc, argv);
MainWindow w;
w.show();
int sd, port;
struct sockaddr_in server;
if (argc != 3)
{
    printf ("Sintaxa: %s <adresa_server> <port>\n", argv[0]);
    return -1;
}
port = atoi (argv[2]);
if ((sd = socket (AF_INET, SOCK_STREAM, 0)) == -1)
{
    perror ("Eroare la socket:");
    return -1;
}

server.sin_family = AF_INET;
server.sin_addr.s_addr = inet_addr(argv[1]);
server.sin_port = htons (port);

if (connect (sd, (struct sockaddr *) &server, sizeof (struct sockaddr)) == -1)
{
    perror ("[client]Eroare la connect:");
    return -1;
}
w.setSoc(sd);
return a.exec();
```

Figura 2. Modalitatea prin care un client se conectează la server.

Server

```
while (1) ///prima bucla while infinita ce mentine serverul activ
{
    socklen_t length = sizeof (from);
    pid_t pid;

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

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

    if (client < 0)
    {
        perror ("[server]Eroare la accept()");
        return -1;
    }
    pid=fork();
    if(pid==-1)
    {
        perror("[server]Eroare la fork()");
        return -1;
    }
    if(pid)
    {
        close (client);
        continue;
    }
    break;
}
```

Figura 3. Modalitatea prin care serverul acceptă clienți și crează procese prin apelul `fork()`

```

while(1) ///a doua bucla while infinita pentru a gestiona cerintele clientului
{

    bzero (msg, 100);
    printf ("[server]Asteptam mesajul...\n");
    fflush (stdout);
    int requestTournaments=0;

    if (read (client, msg, 100) <= 0)
    {
        perror ("[server]Eroare la citire de la client:");
        return -1;
    }

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

    if(strstr(msg,"[1]") ) //logare
    {
        bool canLogin=false;
        prelucrareLogin(username,password,msg);
        char usernameFile[100],passwordFile[100],idFile[100];
        std::string aux_username,aux_password,aux_id;
        qDebug()<<"clientul doreste sa se logheze"<<'\n';
        QFile file("useri.xml");
        if(!file.open(QFile::ReadOnly | QFile::Text))
        {
            qDebug() << "Cannot read file" << file.errorString();
            exit(0);
        }

        QDomStreamReader reader(&file);
        if (reader.readNextStartElement())

```

Figura 4. Modalitatea prin care serverul verifică inputul trimis de client

De menționat pentru această figură faptul că acest cod este mult mai larg(nu au putut fi cuprinse într-o poză toate request-urile primite de la client). În această mostră de cod serverul primește de la client un username și o parolă și verifică cu acestea prin intermediul fișierului *useri.xml* dacă respectivul client se poate loga.

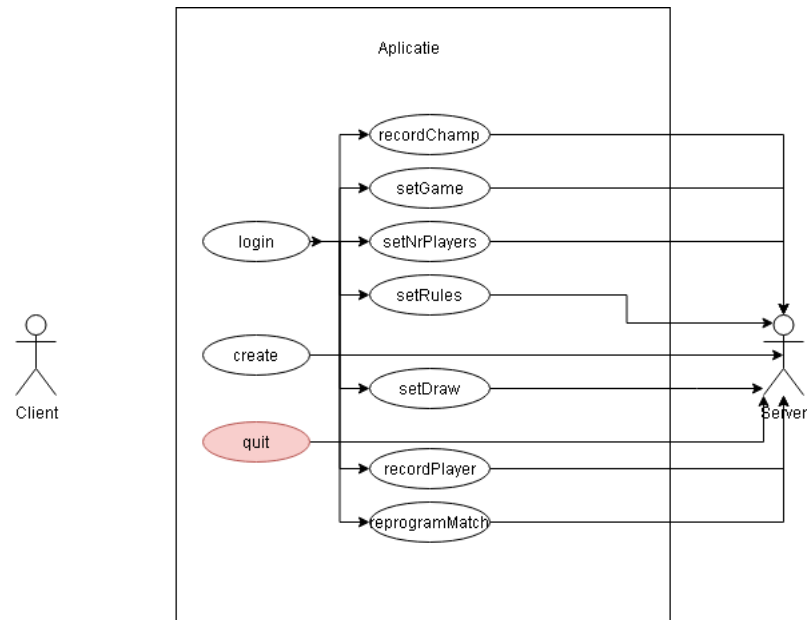


Figura 5. Scenarii de utilizare ale aplicației

5 Galerie

În această secțiune vor fi ilustrate câteva poze cu interfața grafică a aplicației.

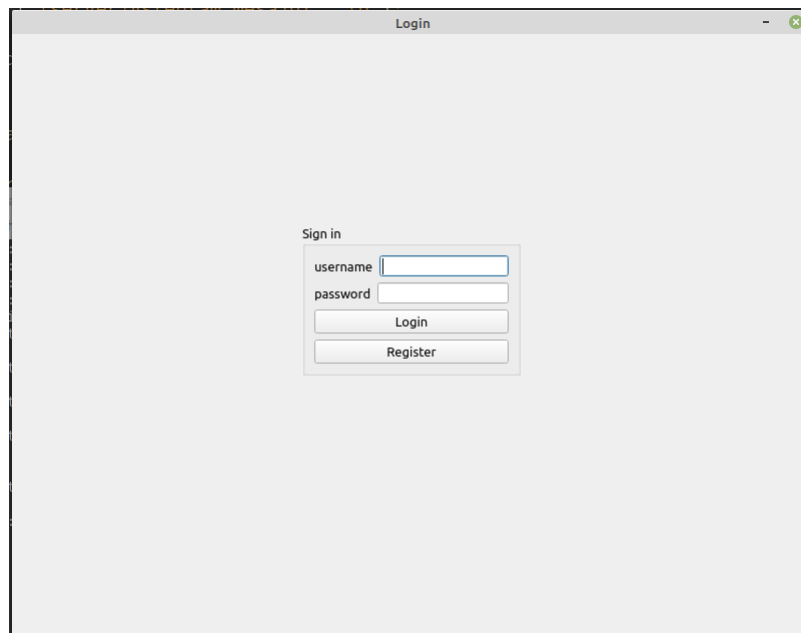


Figura 6. Fereastra de login

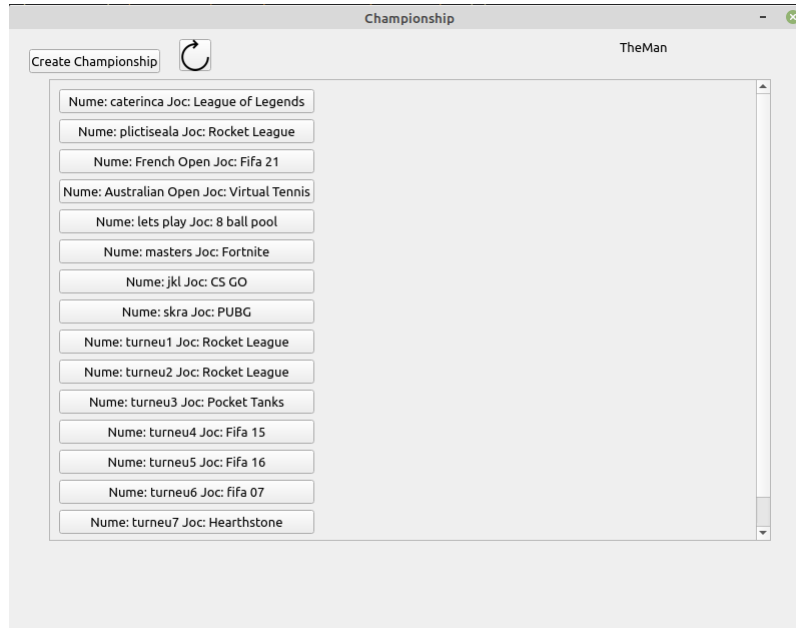


Figura 7. Lista de turnee

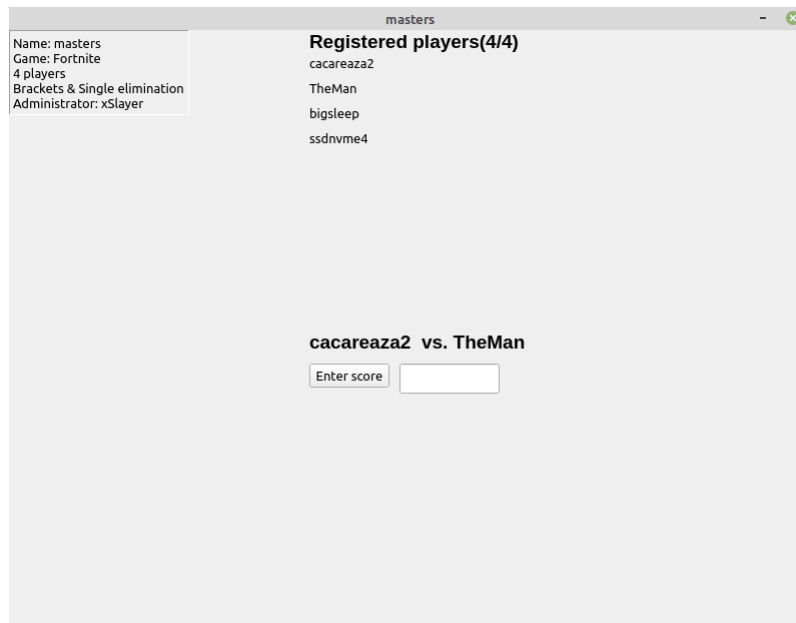


Figura 8. Desfasurarea unui turneu

6 Concluzii

În primul rând, interfața grafică ar putea fi îmbunătățită substanțial, în stadiul actual este relativ rudimentară (un exemplu de îmbunătățire ar fi afișarea bracket-urilor unui turneu).

Pe de altă parte, aplicația ar beneficia de implementarea unor funcții adiționale, precum stabilirea unor statistici pentru fiecare client (e.g. numărul de meciuri pierdute/câștigate), posibilitatea de a se retrage de la un turneu sau resetarea turneului. Ideea esențială aici este aceea că aplicația poate fi dezvoltată prin nemăsurate moduri, în funcție de complexitatea dorită.

În ceea ce privește stocarea datelor, un mod pentru a realiza acest lucru ar fi utilizarea unui sistem de gestionare de baze de date (e.g. SQLite).

References

1. Exemple de scenarii de utilizare
<https://medium.com/@warren2lynch/use-case-learn-by-examples-5a63b67fa64d>
2. Tool pentru realizarea diagramelor
<https://app.diagrams.net/>
3. Pagina cursului - Retele de calculatoare
<https://profs.info.uaic.ro/~computernetworks/cursullaboratorul.php>
4. Pagina laboratorului - Retele de calculatoare
<https://sites.google.com/view/fii-rc/home>
5. Redactarea unui raport în format LNCS
<https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines/>
6. Despre servere TCP, UDP, SCTP
<https://notes.shichao.io/unp/ch2/>
7. Pagina principală Qt
<https://www.qt.io/>
8. Documentație oficială QDomStreamReader
<https://doc.qt.io/qt-5/qxmlstreamreader.html>
9. Documentație oficială QDomDocument
<https://doc.qt.io/qt-5/qdomdocument.html>
10. Tutorial introductiv Qt cu C++ pe YouTube
<https://www.youtube.com/playlist?list=PLS1QulWo1RIZiBcTr5urECberTITj7gjA>
11. Compilare și executare proiect Qt la linia de comandă în Ubuntu
<https://vitux.com/compiling-your-first-qt-program-in-ubuntu/>