

# Raport Tehnic - QuizzGame

Cazan Rares

January 15, 2024

## 1 Introducere

Proiectul QuizzGame este un joc de întrebări și răspunsuri implementat în limbajul de programare C++ utilizând socket-uri pentru comunicare între server și clienți. Scopul proiectului este de a coordona clienții, de a gestiona întrebările și răspunsurile, și de a menține o sincronizare între participanți. Acest raport oferă detalii privind implementarea proiectului.

## 2 Tehnologii Aplicate

Pentru implementare, am ales limbajul de programare C++ datorită eficienței sale și am utilizat biblioteca sockets pentru gestionarea comunicării între server și clienți. Am optat pentru comunicarea prin socket-uri TCP/IP pentru a asigura o transmitere eficientă a datelor între server și clienți.

### 2.1 Motivații și Tehnologii Specifice

- **TCP/IP:** Am ales TCP/IP pentru comunicarea între server și clienți datorită fiabilității sale în transmiterea datelor. Protocolul TCP asigură livrarea ordonată a pachetelor, ceea ce este esențial într-un joc interactiv cum este QuizzGame.
- **Bază de Date SQLite:** Pentru stocarea și gestionarea întrebărilor și răspunsurilor, am optat pentru o bază de date SQLite. Acest lucru oferă o soluție ușoară, eficientă și integrată pentru manipularea datelor, facilitând extensibilitatea jocului.

## 3 Structura Aplicației(Vizualizati figura 1)

Structura aplicației este divizată în două componente principale: server și client.

### 3.1 Server

- Înregistrarea clienților: La conectare, fiecare client este înregistrat și adăugat la o listă de clienți activi.
- Încărcarea întrebărilor: Serverul încarcă întrebările și variantele de răspuns dintr-o bază de date SQLite, respectând formatul specificat în cerințe.
- Coordonarea jocului: Serverul dirijează jocul, adresând întrebările clienților în ordinea înregistrării lor și colectând răspunsurile într-un mod sincronizat.
- Eliminarea clienților: Serverul gestionează situațiile în care un client părăsește jocul, eliminându-l din runde și reorganizând ordinea de adresare a întrebărilor.
- Delogarea clienților: Serverul gestionează cererile de delogare de la clienți, eliminându-i din lista de clienți activi.
- Terminarea jocului: Jocul se încheie când toate întrebările au fost parcurse, iar serverul anunță câștigătorul către toți clienții.

### 3.2 Client

- Răspuns la întrebări: Clientul primește întrebările de la server și furnizează răspunsurile într-un interval de timp prestabilit.
- Părăsirea jocului: Clientul poate părăsi jocul în orice moment, iar serverul va gestiona continuarea jocului fără el.
- Delogare: Clientul poate solicita delogarea, iar serverul îl va elimina din lista de clienți activi.

## 4 Aspecte de Implementare

- Gestionarea conexiunilor: Serverul este implementat în mod multithreading pentru a gestiona simultan mai mulți clienți. Se utilizează mecanisme de blocare pentru a asigura sincronizarea între thread-uri.
- Comunicarea prin socket-uri: Serverul și clienții comunică prin socket-uri TCP/IP, asigurând transmiterea eficientă a datelor între ele.
- Bază de date SQLite: Am implementat o bază de date SQLite pentru stocarea întrebărilor și răspunsurilor, facilitând gestionarea și extinderea jocului.

### Captarea si incarcarea informatiilor din baza de date

```
int CheckSQL(int err, const char *msg)
{
    if(err !=SQLITE_OK)
    {
        fprintf(stderr, "Error: %s (%d)\n", msg, err);
        exit(EXIT_FAILURE);
    }
    return err;
}

int callBack(void *data, int argc, char **argv, char **col)
{
    struct Question *intrebari = (struct Question *)data;

    intrebari[questionCT].id = atoi(argv[0]);
    snprintf(intrebari[questionCT].text,
    sizeof(intrebari[questionCT].text), "%s", argv[1]);
    snprintf(intrebari[questionCT].raspuns1,
    sizeof(intrebari[questionCT].raspuns1), "%s", argv[2]);
    snprintf(intrebari[questionCT].raspuns2,
    sizeof(intrebari[questionCT].raspuns2), "%s", argv[3]);
    snprintf(intrebari[questionCT].raspuns3,
    sizeof(intrebari[questionCT].raspuns3), "%s", argv[4]);
    intrebari[questionCT].raspunsCorect = atoi(argv[5]);
    questionCT++;

    return 0;
}

void LoadIntrebari()
{
    char *err;
    char *selectIntrebari = "SELECT * FROM intrebari;";
    questionCT = 0;

    CheckSQL(sqlite3_exec(db, selectIntrebari, callBack,
    intrebari, &err), "Failed to load
    questions from database");
}
```

#### Trimiterea intrebarilor catre clienti

```
for(int i=0; i<questionCT; i++)
{
    this_thread::sleep_for(chrono::seconds(5));

    string questionMessage = "Intrebarea " +
        to_string(i + 1) + "): " + intrebari[i].text
        + "\n1. " +
        intrebari[i].raspuns1 + "\n2. " +
        intrebari[i].raspuns2 + "\n3. " +
        intrebari[i].raspuns3 + "\nRaspunsul tau: ";

    send(clientSocket, questionMessage.c_str(),
        questionMessage.length(), 0);
}
```

#### Primirea raspunsurilor de la client si actualizarea punctajelor

```
if(sscanf(buffer, "%d", &playerAnswer)==1 &&
playerAnswer>=1 && playerAnswer<=3)
{
    auto end = std::chrono::steady_clock::now();
    auto elapsed_time = std::chrono::duration_cast
<std::chrono::seconds>(end-start).count();

    if(elapsed_time <= 5)
    {
        if(playerAnswer==intrebari[i].raspunsCorect)
        {
            send(clientSocket, "Raspuns corect! Ai
castigat un punct!", strlen
("Raspuns corect! Ai castigat un punct!"),
0);
            players[playerNumber].punctaj
+=1;
            if(players[playerNumber].punctaj >
maxScore)
            {
                winner=playerNumber;
                maxScore=players[playerNumber].punctaj;
            }
        }
        else
        {
            send(clientSocket, "Raspuns gresit!",
strlen("Raspuns gresit!"), 0);
        }
    }
}
```

## 5 Concluzii

Proiectul QuizzGame oferă o soluție eficientă pentru implementarea unui joc de întrebări și răspunsuri, în concordanță cu cerințele propuse. Potențiale îmbunătățiri pot include adăugarea unui sistem de punctaj și extinderea tipurilor de întrebări.

## 6 Referințe Bibliografice

- Documentație SQLite: <https://www.sqlite.org/docs.html>

- Sockets Programming in C: <https://profs.info.uaic.ro/georgiana.calancea/>
- Desen diagrama: <https://app.diagrams.net/>

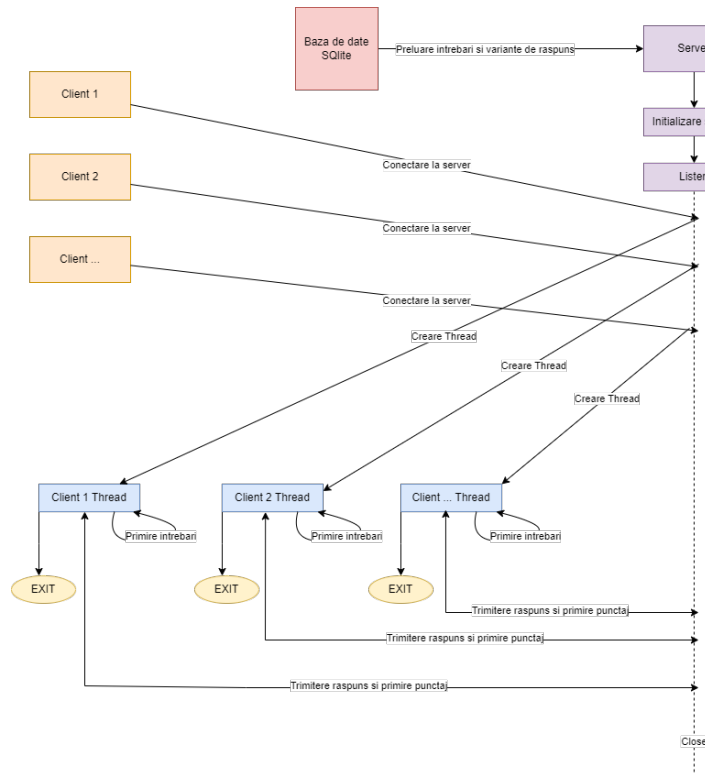


Figura 1