# Digital Dashboard pentru restaurante

Chisca Serban 3B2

No Institute Given

Abstract. Acest raport prezintă proiectarea și implementarea unui sistem de gestionare a meniurilor unui restaurant utilizând socket-uri TCP. Sistemul suportă funcționalități precum accesarea și modificarea elementelor din meniu, asigurând în același timp un control securizat al accesului pentru sarcinile manageriale. Aspectele tehnice, arhitectura și detaliile implementării proiectului sunt discutate în cadrul acestui raport.

#### 1 Introducere

Proiectul implementat are ca scop crearea unui sistem pentru gestionarea meniului unui restaurant utilizând comunicații de tip TCP. Obiectivele principale sunt:

- Oferirea unui mod simplu pentru utilizatori de a consulta meniurile disponibile.
- Permisiuni exclusive pentru manager pentru a modifica meniurile.
- Stabilirea unei conexiuni fiabile între client și server.

## 2 Tehnologii Aplicate

Proiectul utilizează următoarele tehnologii:

- TCP Sockets: Protocolul TCP asigură o comunicare sigură și fiabilă între client și server, fiind ales datorită nevoii de consistență în transmiterea comenzilor și a răspunsurilor.
- Pthread: Folosirea thread-urilor permite serverului să gestioneze simultan mai mulți clienți, maximizând performanța.
- Limbajul C: Alegerea limbajului C a fost motivată de eficiența acestuia în gestionarea operațiunilor de rețea la nivel scăzut.

### 3 Structura Aplicației

Aplicația este structurată într-un model client-server:

- Serverul: Acceptă conexiuni de la mai mulți clienți și gestionează comenzile acestora.
- Clientul: Permite utilizatorilor să trimită comenzi și să primească răspunsuri de la server.

Diagrama detaliată a aplicației este prezentată în Fig. 1.

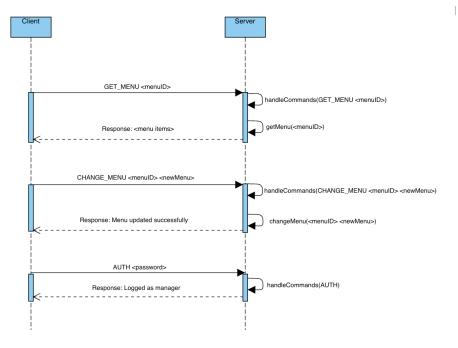


Fig. 1. Diagrama aplicației.

## 4 Aspecte de Implementare

Codul serverului utilizează socket-uri pentru a asculta conexiuni și pthread-uri pentru a gestiona mai mulți clienți. Exemplele sunt detaliate mai jos.

### 4.1 Protocolul Aplicației

Protocolul implementat include următoarele comenzi:

- GET\_MENU <număr\_meniu>: Returnează meniul specificat.
- CHANGE\_MENU <număr\_meniu> <conținut\_nou>: Modifică meniul, accesibil doar managerului după autentificare.
- AUTH <parolă>: Autentifică managerul.

### 4.2 Threaduri pentru Gestionarea Conexiunilor

Pentru a gestiona mai mulți clienți simultan, serverul utilizează thread-uri. Fiecare client conectat are un thread dedicat pentru a procesa cererile acestuia. Implementarea este următoarea:

Listing 1.1. Crearea thread-urilor pentru clienți

Funcția pthread\_create creează un thread nou pentru fiecare client conectat. Fiecare thread rulează funcția handleClientRequest, care gestionează cererile clientului. Resursa este eliberată automat folosind pthread\_detach după terminarea thread-ului.

#### 4.3 Scenariu Real de Utilizare

Un client trimite comanda GET\_MENU 1, iar serverul răspunde cu:

```
Menu 1: Pizza Vegana, Pizza Pui, Pizza Peperoni
```

Managerul folosește comanda  ${\tt AUTH}\,$ admin<br/>1234 pentru a se autentifica și ulterior modifică meniul:

CHANGE\_MENU 1 Pizza Margherita, Pizza Quattro Formaggi

### 5 Concluzii

Sistemul implementat funcționează corespunzător, însă există îmbunătățiri posibile, cum ar fi:

- Adăugarea suportului pentru salvarea persistentă a meniurilor.
- Extinderea funcționalității pentru a include statistici despre comenzile clienților.
- Implementarea criptării pentru o securitate sporită.