

# Documentație Rețele de Calculatoare - ConsoleShopper

Artene Raluca-Ioana

Grupa X4

An universitar 2022-2023

Universitatea Alexandru Ioan Cuza Iași, Facultatea de Informatică

## 1 INTRODUCERE

Această fișă de raport are scopul de a pune în vedere detaliile referitoare la realizarea și funcționalitățile proiectului pe care doresc să îl implementez, și anume proiectul **ConsoleShopper**. Am ales acest proiect deoarece mi s-a parut interesant să încerc să implementez la un nivel minimal cum este gândit un magazin online.

Enunțul proiectului prezintă o aplicație client/server ce trebuie să ofere clienților funcționalitățile unui magazin online de produse. Astfel, comunicația va fi realizată prin intermediul socket-urilor în așa fel încât programul client să poată fi rulat de orice utilizator, iar programul server doar de autor, care va pune la dispoziția acestora diferite comenzi cum ar fi autentificarea, înregistrarea și alte comenzi ce stau la baza unui magazin online. De asemenea, programul server va putea să servească simultan mai multe cereri primite de la clienți.

## 2 TEHNOLOGII UTILIZATE

### 2.1 TCP

Ținând cont de faptul că se dorește deservirea mai multor clienți în același timp, pentru realizarea conexiunii între server și client se va folosi un server concurent TCP. Folosind acest tip de protocol avem garanția absolută că datele transferate rămân intacte și ajung în aceeași ordine în care au fost trimise. "TCP este un protocol folosit în general de aplicații care au nevoie de confirmare de primire a datelor. Efectuează o conectare virtuală full duplex între două puncte terminale, fiecare punct fiind definit de o adresă IP și de un port TCP. Acesta asigură livrarea ordonată a unui flux de octeți de la un program de pe un computer la alt program de pe un alt computer aflat în rețea." [1]

### 2.2 Multithreading

Pentru a servi mai mulți clienți simultan folosim tehnologia de multithreading, fiecare thread reprezentând un client unic. Folosim această metodă deoarece, spre

deosebire de procese, toate thread-urile accesează aceeași memorie(shared memory). Serverul TCP va crea pentru fiecare client în parte un thread, facilitând servirea simultană a celor conectați în rețea. Astfel, se va eficientiza execuția programelor, executând porțiuni distincte de cod în paralel în interiorul aceluiași proces. Utilizând thread-uri, serverul va putea accepta conectarea unui client, crearea unui thread pentru comunicație și continuarea primirii request-urilor de la ceilalți clienți.

### 2.3 UNIX domain sockets

Pentru a permite comunicarea între server și clienți folosim UNIX domain sockets deoarece acestea permit comunicarea în ambele direcții. Astfel fiecare proces poate să scrie și să citească de la același descriptor, fără necesitatea de a crea două canale de comunicare separate.

### 2.4 Fișiere text

Fișierele de tip .txt au fost utilizate datorită simplității lor. Sunt folosite în mai multe situații, credențialele de autentificare/înregistrare sunt salvate într-un fișier text denumit credentials.txt, precum și produsele disponibile organizate pe categorii în fișierul products.txt. Coșul de cumpărături al fiecărui client se salvează într-un fișier de tipul user.txt unde user este username-ul clientului deja autentificat.

## 3 ARHITECTURA APLICAȚIEI

### 3.1 Funcționalitatea aplicației

Așa cum am precizat anterior, comunicarea va fi realizată prin socket-uri și thread-uri. Astfel, sunt create două fișiere client.c și server.c, unde clientul are posibilitatea de a se conecta la server. Pentru început clientul trebuie să introducă comanda login sau register în funcție de ce are nevoie. Fără autentificarea corespunzătoare clientul nu va avea posibilitatea de a solicita alte tipuri de comenzi.

**Comanda REGISTER.** Clientul trebuie să aleagă un username și apoi o parolă cu care se va conecta la server. Acestea urmează apoi să fie stocate în fișierul credentials.txt pentru a putea fi folosite la următoarea conectare la server. După înregistrare, clientul rămâne autentificat în noul cont, unde are acces și la restul comenzilor existente.

**Comanda LOGIN.** Ca și la comanda descrisă mai sus, clientul trebuie să introducă username-ul și parola cu care s-a înregistrat prima dată. După autentificare va avea posibilitatea de a transmite alte comenzi către server pentru a executa acțiunile dorite.

**Comanda CATEGORIES.** Această comandă oferă clientului denumirile categoriilor existente în magazin. Serverul deschide fișierul `products.txt` și citește toate categoriile apoi le trimite către client. O linie din fișier trebuie să respecte următorul format `;;categorie: produs produs produs;;`.

**Comanda SHOW PRODUCTS.** Trimițând această comandă către server clientul este apoi rugat să introducă numele categoriei pentru care dorește să vizualizeze produsele existente. Odată introdusă categoria dorită serverul caută în fișierul `products.txt` categoria apoi dacă este găsită trimite înapoi la client informațiile. În cazul în care categoria este inexistentă serverul trimite înapoi un mesaj anunțând clientul că ceea ce a introdus nu se numără printre categoriile existente.

**Comanda ADD TO CART.** După trimiterea comenzii `add to cart` clientul este rugat să introducă produsul dorit. Serverul urmează să creeze (în cazul în care nu există deja) sau să completeze fișierul de forma `username.txt` adăugând astfel produsul dorit.

**Comanda SHOW CART.** Comanda afișează clientului produsele pe care le-a adăugat în coșul de cumpărături anterior.

**Comanda BUY CART.** Comanda `buy cart` trimite un mesaj de confirmare clientului anunțând că au fost trimise cumpărăturile efectuate, iar serverul șterge coșul de cumpărături.

**Comanda EXIT.** Comanda `exit` deconectează serverul de la client, închizând comunicarea dintre cele două, dar nu înainte de a șterge automat coșul de cumpărături. Odată deconectat clientul nu mai are acces la vechiul coș.

### 3.2 Diagrama aplicației

Diagrama de mai jos reprezintă faptul că după ce un client introduce o comandă de autentificare sau înregistrare, aceasta va fi redirecționată către server și mai apoi către fișierul text în care se salvează toate credențialele cu scopul analizării acestuia pentru momentul în care contul va fi creat/verificat la următoarea conectare.

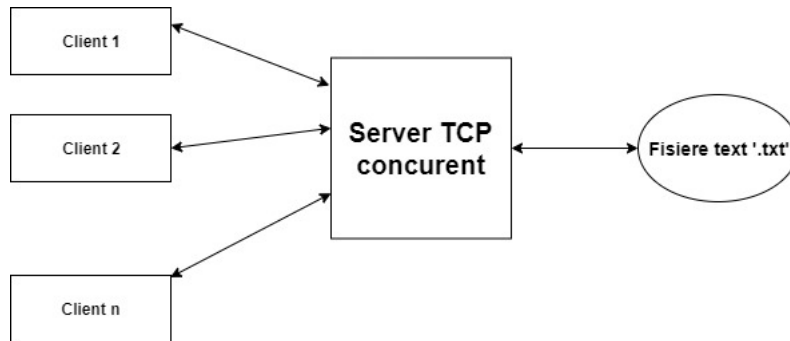


Fig. 1. Diagrama ConsoleShopper

## 4 CONCLUZII

Această aplicație ar mai putea fi îmbunătățită în stocării datelor, securității și de asemenea ar putea fi implementată și grafica programului. De asemenea ar mai putea fi implementate câteva comenzi ce ar putea face aplicația mai complexă și mai interesantă de utilizat pentru clienți.

**Stocarea datelor.** La momentul actual, aplicația utilizează fișierele text pentru stocarea diverselor date precum username-ul, parolele, coșurile de cumpărături ale clienșilor, produsele oferite de magazin și categoriile în care sunt organizate acestea. Pentru îmbunătățire poate fi folosită o bază de date ce ar oferi o notă mai elegantă implementării.

**Securitatea** La momentul actual, așa cum am menționat anterior parolele sunt de asemenea stocate în fișiere text. Un mare plus ar fi să fie implementată o altă metodă mai sigură pentru accesarea conturilor.

**Grafica.** Pentru a face aplicația ușor de utilizat pentru orice client putem crea o interfață grafică intuitivă.

**Alte comenzi.** Alte comenzi sau o modificare a celor actuale ar aduce cu siguranță un plus aplicației. Spre exemplu ar putea fi adăugată o metodă de a adăuga produsele la favorite fără a fi cumpărate. Un alt tip folositor de comandă ar putea fi una de tip delete product unde utilizatorul ar putea șterge produsele pe care se decide să nu le mai cumpere din coșul de cumpărături.

## References

1. [https://ro.wikipedia.org/wiki/Transmission\\_Control\\_Protocol](https://ro.wikipedia.org/wiki/Transmission_Control_Protocol)
2. <https://profs.info.uaic.ro/computernetworks/cursulaboratorul.php>
3. <https://www.diagrameditor.com/>