

Offline Messenger

Smău Adrian-Constantin, anul 2, grupa B5
smauadrian@gmail.com

Facultatea de Informatică Iași

1 Introducere

O aplicație server/client care permite schimbul de mesaje între clienții conectați la server, însă implementează și trimiterea de mesaje către utilizatori ce nu sunt conectați, acestora din urmă apărându-le mesajele atunci când se vor conecta. Utilizatorii au posibilitatea să trimită un răspuns (reply) în mod specific la anumite mesaje primite. Aplicația oferă și istoricul conversațiilor pentru și cu fiecare utilizator în parte, care poate fi vizualizat sau șters. Toate aceste funcționalități sunt, însă, accesibile dacă utilizatorul se loghează cu un cont existent în baza de date, sau se înregistrează, fiind adăugat astfel în baza de date.

2 Tehnologiile utilizate

În cadrul implementării proiectului, am folosit tehnologia TCP. Mai precis, aplicația se bazează pe TCP concurent folosind thread-uri.

2.1 Modelul server-client

Modelul client-server este un mod de structurare al unei aplicații distribuite care ajută la partajarea de resurse dintre un provider (server) și un requester (client). Clientul este cel care inițiază conexiunile și de asemenea cel care se ocupă de request-uri. Serverul este cel care procesează cererile primite de la client, nu invers. Clientul comunică cu serverul folosindu-se de un protocol stabilit apropiat. Rolul unui server este de a se fixa (bind) pe un port specific (port folosit de client pentru a localiza serverul) și să asculte pentru noi conexiuni. Spre deosebire de client care este temporar și rulează doar când utilizatorul alege, serverul trebuie să ruleze în mod continuu (chiar dacă nu există clienți conectați) pentru a se asigura că eventualele servicii client se pot conecta.

2.2 Stiva TCP/IP (Transmission Control Protocol/Internet Protocol)

TCP/IP este o stivă de protocoale de comunicare folosită pentru a interconecta la distanță (via internet) dispozitive diferite. Deși este folosită în principal pentru a conecta dispozitive remote, stiva TCP/IP poate fi folosită și într-o rețea privată. Întreaga stivă de protocoale (reguli de comunicare) folosită de către

internet este cunoscută sub numele de TCP/IP. TCP/IP specifică modul în care datele sunt schimbate între doua endpoint-uri – este pus la dispoziție un model standard de comunicare end-to-end care specifică exact cum ar trebui împărțite pachetele, cum ar trebui adresate, transmise, routate și primite la destinație. Cele două protocoale prezente în numele stivei au roluri bine definite. TCP definește cum se pot crea canale de comunicare pentru a trimite date via internet (dispozitive aflate pe mașini diferite). De asemenea, se ocupă de modul în care mesajele sunt împărțite în mesaje mai mici, trimise în rețea și reasamblate în ordinea corectă la destinație. IP definește modul în care pachetele sunt adresate și împartite (routed) până la destinație și asigură că pachetele ajung la destinația corectă. TCP (Transmission Control Protocol) este protocolul pe care l-am abordat pentru comunicarea client-server. Un protocol, în contextul rețelelor de calculatoare, este un set de reguli și proceduri care guvernează modul în care datele sunt transmise. Scopul unui protocol este acela de a furniza un mod de comunicare pentru oricine din întregul world wide web, independent de locație, limbaj, hardware. Scopul protocolului TCP este acela de a controla transferul de date în așa fel încât acesta să fie de încredere. În contextul întregului internet, datele sunt trimise în pachete. Pachetele sunt bucăți de date transmise independent și sunt reasamblate odată ce ajung la destinație pentru a ajunge la forma originală. Aplicația având nevoie de un mod de comunicare între server și client cât mai eficient, am ales protocolul TCP deoarece este orientat conexiune, oferind mecanisme de control ale congestiei și ale fluxului, fiind în același timp sigur, în sensul în care garantează transmiterea corectă și în ordine a datelor. Implementarea cu ajutorul thread-urilor (câte un thread pentru fiecare client, în mod concurrent) asigură eficiență și ușurință în tratarea clienților de către server.

3 Arhitectura aplicației

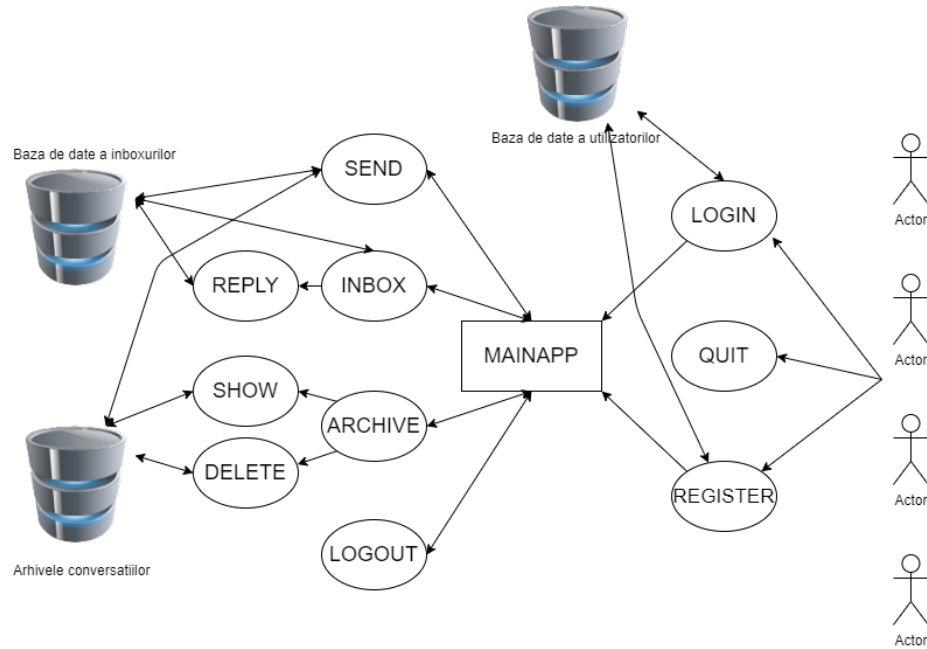


Fig. 1. Diagrama aplicației

Aplicația este structurată în două module : cel de login si cel ce presupune funcționalitățile aplicației în sine odată ce utilizatorul este conectat.

3.1 Moduri de stocare a datelor

Baza de date a utilizatorilor - Organizarea bazei de date a conturilor din sistem este următoarea : local, am definit struct-ul "User info" care reține local pentru fiecare utilizator din baza de date ID-ul, username-ul, parola, prenumele și numele. "Umplerea" acestui struct se face la inițializarea serverului, când din baza de date externă se importă datele fiecărui utilizator în parte și se trec în struct-ul menținut anterior. Baza de date externă constă într-un fișier text, în care fiecare rând corespunde unui user. Sintaxa fișierului este "ID";"USERNAME";"PAROLA";"PRENUME";"NUME", delimitatorul ";" fiind un element ce ne ajută atunci când importam local userii. Atunci când un utilizator nou se înregistrează, acesta este adăugat local dar este adăugat și în "database.txt", baza de date externă.

Statusul utilizatorilor - Statusul fiecarui utilizator este reținut local prin intermediul unui array de int-uri cu dimensiunile [countID][2], prima dimensiune fiind alocată pentru fiecare utilizator din sistem iar cea de-a doua poate lua valorile 0 sau 1 - însemnând 0 pentru OFFLINE și 1 pentru ONLINE. La inițializarea serverului, toate statusurile sunt OFFLINE. La autentificarea/înregistrarea unui user, statusul său este marcat drept ONLINE. La deconectare, este restabilită starea OFFLINE.

Inboxurile - Local, am implementat structul "inbox" care reține pentru fiecare utilizator din sistem ID-ul, numărul de mesaje din inbox, fiecare mesaj în parte precum și sender-ul fiecărui mesaj în parte. Extern, fiecare user din sistem are un inbox în fișierul "inboxes/", unde fiecare rând corespunde unui mesaj nou. Fiecare mesaj este prefixat cu "(n)", unde n este indexul său. La inițializarea serverului, se importă local toate inboxurile utilizatorilor (dacă nu există, se creează). La înregistrarea unui nou user, i se creează și un inbox extern, dar și intern.

Arhivele conversațiilor - Fiecare utilizator din sistem are o arhivă externă cu fiecare alt utilizator diferit de el însuși. Aceste conversații se află în fișierul "comms/" și se inițializează la pornirea serverului. Atunci când un nou user se înregistrează, îi este creată o conversație cu toți ceilalți utilizatori din sistem, precum și lor cu acest nou user.

Alte detalii - ID-ul fiecărui client se decide, cu scopul simplității, prin ordinea de înregistrare în sistem. O variabilă locală, countID, este incrementată cu fiecare nou user, și reprezintă numărul total de utilizatori din sistem. Am setat și o limită de utilizatori în sistem, care, dacă este atinsă, face ca înregistrările în sistem să nu mai fie posibile. De asemenea, există o limită a mesajelor din inbox. Dacă limita este atinsă, mesajele trimise nu se vor mai înregistra în inbox, ci doar în arhive.

3.2 Modulul Login/Register

La conectarea clientului la server, acestui îi sunt prezentate două mai multe posibile comenzi : login, register sau quit. "Quit" determina terminarea clientului și deconectarea, iar orice eroare de sintaxă este tratată prin afișarea unui mesaj de avertisment. Opțiunea "login" îi cere utilizatorului să își introducă prima oară username-ul. În server, se caută local existența acestui username. Dacă această căutarea este validată, userului i se cere parola. În caz contrar, clientul este deconectat și informat. În server, parola este verificată în corelație cu username-ul. Dacă parola corespunde username-ului introdus, utilizatorului i se oferă acces la aplicația în sine și îi este reținut ID-ul pentru viitoare operații. Dacă nu, este deconectat și avertizat. "Register" este ultima opțiune din acest modul. Clientului i se cere să introducă un nume de utilizator. Apoi, în server, se verifică

disponibilitatea acestui username. Dacă se află deja în baza de date, clientul este avertizat și deconectat. Dacă nu, mai departe, acestuia i se cere parola, prenumele și numele, în ordine. Parola are voie să fie nulă însă dacă fie prenumele, fie numele sunt vide, înregistrarea nu are loc, clientul este avertizat și deconectat.

3.3 Modulul Mainapp

Odată cu autentificarea cu succes a unui utilizator, acesta se consideră ONLINE și are acces la meniul principal. Aici, are mai multe opțiuni : send, inbox, archive și logout. Dacă are loc o eroare de sintaxă, un mesaj de avertisment este prezentat.

Opțiunea send - permite userului să trimită un mesaj oricărui alt user din sistem. La selectarea acestei comenzi, clientului îi este afișată o listă cu toți utilizatorii din sistem, precum și statusul lor - dacă sunt sau nu activi. Apoi, clientului i se cere să introducă ID-ul userului căruia dorește să îi trimită mesajul și mesajul în sine. Dacă nu există erori de sintaxă, un mesaj este inserat în inboxurile local și extern ale receiver-ului, precum și în conversațiile sender-receiver și receiver-sender. Mesajul nu are voie să fie nul, iar dacă receiver-ul nu este introdus corect, clientul este avertizat.

Opțiunea inbox - permite vizualizarea inboxului și reply-ul (dacă se dorește) la un mesaj anume dintre cele aflate în inbox. Mesajele sunt prefixate cu un indice, facilitând procesul de reply. Inboxul este automat golit (local și extern), întrucât a fost accesat și "văzut". Dacă se dorește un reply, clientul este notificat că trebuie să introducă indexul mesajului la care dorește să răspundă. Dacă nu dorește acest lucru, poate tasta comanda "back" care îl va duce înapoi la meniul principal. Orice greșală sintactică este semnalată. Apoi, dacă se dorește reply, clientului îi este cerut să introducă mesajul dorit. Dacă mesajul este nul, clientul este avertizat și mesajul nu este trimis. În caz contrar, mesajul este trimis în aceeași manieră în care este trimis la accesarea comenzii "send".

Opțiunea archive - oferă posibilitatea de a vedea sau de a șterge o conversație cu un anumit alt user din sistem. La selectarea comenzii, clientului îi este prezentată o listă cu toți userii din baza de date, și poate introduce ID-ul oricăruia dintre aceștia (mai puțin al său) pentru a vedea/șterge conversația. Orice eroare de sintaxă este semnalată. La selectarea corectă a unui ID, clientul are opțiunile "show" sau "delete". "Show" îi arată istoricul conversațiilor cu userul respectiv, iar "delete" șterge conversația sa cu respectivul (însă celui user ales îi rămâne în continuare o copie a conversației).

Opțiunea logout - deloghează userul și îl marchează drept "offline".

Troubleshoot - sunt tratate și cazurile în care utilizatorul se deconectează subit în orice etapă a programului. Dacă este în mijlocul unei operații sau este în stand-by în meniul principal sau cel de login, serverul se asigură că nicio eroare nu are loc în bazele de date, și clientul este considerat deconectat.

4 Detalii de implementare

```

+++++
Bun venit! Pentru a continua, trebuie sa te autentifici. Daca nu ai cont, te pot
i inregistra. Sintaxa comenzilor este urmatoarea :
<login>, pentru a te loga in contul de utilizator deja existent
<register>, pentru a-ti crea un nou cont de utilizator
<quit>, pentru a iesi din program
+++++
login
Va rog introduceti numele dvs. de utilizator : adygamer
Va rog introduceti parola : 123
Login efectuat cu succes!

Vei fi redirectat la Main Menu!...

+++++
Bun venit pe platforma Offline Messenger, Adrian!
Sintaxa comenzilor este urmatoarea :
<send>, pentru a-i trimite unui utilizator un mesaj
<inbox>, pentru a verifica daca ai mesaje noi
<archive>, pentru a vedea/sterge istoricul conversatiilor cu alti utilizatori
<logout>, pentru a iesi din program
+++++

```

Fig. 2. Exemplu de login cu un cont existent în baza de date

```

+++++
Bun venit! Pentru a continua, trebuie sa te autentifici. Daca nu ai cont, te pot
i inregistra. Sintaxa comenzilor este urmatoarea :
<login>, pentru a te loga in contul de utilizator deja existent
<register>, pentru a-ti crea un nou cont de utilizator
<quit>, pentru a iesi din program
+++++
login
Va rog introduceti numele dvs. de utilizator : adygamer
Va rog introduceti parola : 234

Parola incorecta! Va rugam incercati din nou...
Disconnecting...

```

Fig. 3. Exemplu de login eșuat, întrucât am introdus parola greșită

```

+++++
Bun venit! Pentru a continua, trebuie sa te autentifici. Daca nu ai cont, te pot
i inregistra. Sintaxa comenzilor este urmatoarea :
<login>, pentru a te loga in contul de utilizator deja existent
<register>, pentru a-ti crea un nou cont de utilizator
<quit>, pentru a iesi din program
+++++
register
Incepe inregistrarea prin a-ti seta un nume de utilizator : andi22
Mai departe, va rog scrieti o parola usor de retinut cu care va veti loga in con
t : 234
Introduceti prenumele dvs. : Andrei
Introduceti numele dvs. : Popescu
Cont creat cu succes!

Vei fi redirectat la Main Menu!...

+++++
Bun venit pe platforma Offline Messenger, Andrei!
Sintaxa comenzilor este urmatoarea :
<send>, pentru a-i trimite unui utilizator un mesaj
<inbox>, pentru a verifica daca ai mesaje noi
<archive>, pentru a vedea/sterge istoricul conversatiilor cu alti utilizatori
<logout>, pentru a iesi din program

```

Fig. 4. Exemplu de register valid, astfel înregistrându-ne în baza de date

```

+++++
Bun venit! Pentru a continua, trebuie sa te autentifici. Daca nu ai cont, te pot
i inregistra. Sintaxa comenzilor este urmatoarea :
<login>, pentru a te loga in contul de utilizator deja existent
<register>, pentru a-ti crea un nou cont de utilizator
<quit>, pentru a iesi din program
+++++
register
Incepe inregistrarea prin a-ti seta un nume de utilizator : andi22
Numele de utilizator se afla deja in baza noastra de date sau a aparut o eroare!
Va rugam incercati din nou...
Disconnecting...

```

Fig. 5. Încercare eşuată de înregistrare, username-ul fiind deja în uz

```

+++++
Bun venit pe platforma Offline Messenger, Adrian!
Sintaxa comenzilor este urmatoarea :
<send>, pentru a-i trimite unui utilizator un mesaj
<inbox>, pentru a verifica daca ai mesaje noi
<archive>, pentru a vedea/sterge istoricul conversatiilor cu alti utilizatori
<logout>, pentru a iesi din program
+++++
send
Aceasta este lista utilizatorilor din baza de date:
1 - adygamer - ACTIVE
2 - vick23 - OFFLINE
3 - geo44 - OFFLINE
4 - andi22 - OFFLINE
Alegeti utilizatorul caruia doriti sa ii scrieti un mesaj prin a-i scrie ID-ul m
ai jos!
4
Acum, scrieti mesajul pe care doriti sa il trimiteti!
Ce faci?

Mesaj trimit cu succes! Veti fi redirectionat la meniul principal!..

```

Fig. 6. Exemplu de trimitere a unui mesaj

```

+++++
Bun venit pe platforma Offline Messenger, Andrei!
Sintaxa comenzilor este urmatoarea :
<send>, pentru a-i trimite unui utilizator un mesaj
<inbox>, pentru a verifica daca ai mesaje noi
<archive>, pentru a vedea/sterge istoricul conversatiilor cu alti utilizatori
<logout>, pentru a iesi din program
+++++
inbox

Aveti (1) mesaje noi :

(1)From adygamer : Ce faci?

Introduceti indexul mesajului la care doriti sa raspundeti in cazul in care dor
ti sa faceti acest lucru, sau comanda <back> daca doriti sa fiti redirectionati
in meniul principal. Inboxul dumneavoastra va fi golit!
1
Va rog introduceti mesajul cu care vreti sa raspundeti!...
Bine,tu?

Mesaj trimis cu succes! Veti fi redirectionat la meniul principal!..
+++++

```

Fig. 7. Exemplu de folosire a funcției inbox, primind mesajul trimis în figura anterioară și oferindu-i un răspuns


```

archive
Aceasta este lista utilizatorilor din baza de date:
1 - adygamer - OFFLINE
2 - vick23 - OFFLINE
3 - geo44 - OFFLINE
4 - andi22 - ACTIVE
Alegeti utilizatorul cu care doriti sa vedeti arhiva prin a-i scrie ID-ul mai jo
s!
1
Daca doriti sa vizualizati conversatia cu utilizatorul mentionat, introduceti co
manda <show>. Daca doriti sa stergeti aceasta conversatie, introduceti comanda <
delete>
show

Va fi afisata arhiva convorbirilor cu utilizatorul cu ID-ul : 1

[Adrian Smau] - Ce faci?
[Andrei Popescu] - Bine,tu?
END-OF-CONVERSATION

Veti fi redirectionat catre meniul principal!...
+++++
```

Fig. 8. Exemplu de folosire a funcției archive pentru a vizualiza o conversație

Prin intermediul unui use-case diagram, voi exemplifica ce se întâmplă în cadrul comunicării client/server într-un scenariu relevant funcționalității aplicației : clientul 1 se loghează în sistem. Alegând operațiunea login, clientul îi trimite serverului indicele 1, care este interpretat drept alegerea opțiunii de login. Mai departe, acesta își introduce username-ul, pe care serverul îl primește și verifica în baza de date locală (care corespunde cu cea externă) existența username-ului. Dacă este găsit, îi trimite userului ID-ul clientului cu username-ul respectiv. În caz contrar, îi trimite -1. Dacă feedback-ul de la server este diferit de -1, clientul preia din stdin parola pe care o trimite serverului. În urma unui check, clientului îi este trimis încă un feedback, 1 dacă parola este validă, -1 în caz contrar. Clientul este logat. La intrarea în mainapp, clientului îi este trimis ID-ul său, precum și prenumele userului său, pe care le reține. Alegând opțiunea send, clientul 1 trimite serverului un index corespunzător acestei operații. Acum serverul prelucrează fiecare utilizator din sistem și îi trimite, linie cu linie, clientului, care îi afișează (dar nu înainte să îi trimită numărul de utilizatori din sistem, pentru ca acest client să știe cât să citească). Clientul citește din stdin un indice care reprezintă ID-ul unui user pe care îl trimite serverului. Apoi, citește din stdin un mesaj care se dorește a fi trimis pe care îl trimite serverului. Apoi, serverul adaugă mesajul în : inboxul local, incrementând numărul de mesaje din inboxul receiverului și reținând senderul, în inboxul extern, precum și în arhivele conversațiilor dintre cei doi. Operațiunea a luat sfârșit. Cel de-al doilea user se loghează, apoi, în aceeași manieră. Alege comanda inbox, clientul trimițând serverului indicele corespunzător. Serverul trimite apoi, linie cu linie (dar nu înainte să trimită numărul de mesaje din inbox pentru ca clientul să știe cât să citească) mesajele din inboxul userului. Clientul 2 se decide să răspundă la mesajul clientului 1, și

astfel tastează indicele corespunzător. Acest indice al mesajului se trimite către server, apoi primind și mesajul cu care se dorește a se răspunde. Acum, serverul înregistrează mesajul în arhivele conversațiilor, în inboxurile local și extern ai clientului 1, și golește inboxurile local și extern ale clientului 2. Operațiunea a luat sfârșit.

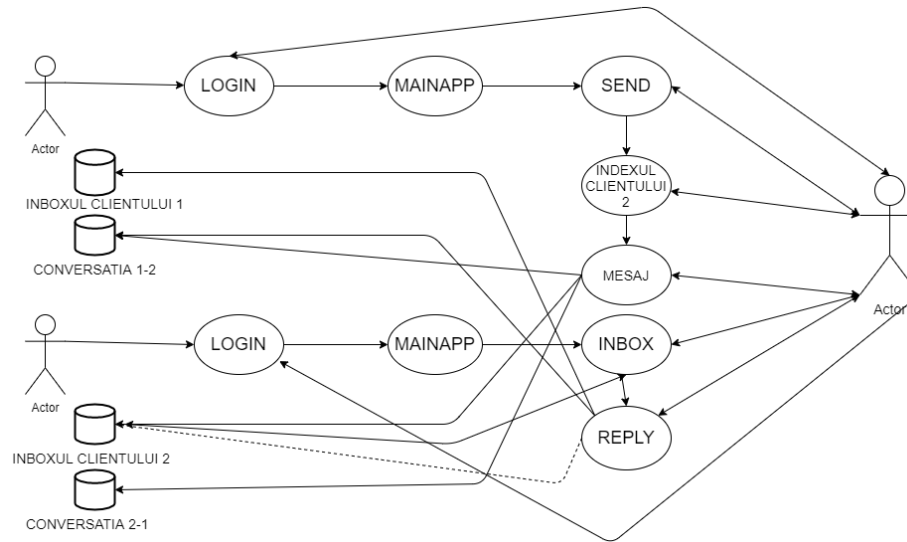


Fig. 9. Exemplul descris mai sus. Serverul este actorul din dreapta, clientul 1 este actorul stânga sus, iar clientul 2 este actorul stânga jos

5 Concluzii

Consider că acest proiect m-a ajutat să aprofundez noțiunile propuse atât în cadrul laboratoarelor, cât și în cadrul cursurilor. Unele lucruri ce puteau fi implementate pentru a îmbunătăți aplicația ar fi: conturi cu diverse drepturi (admin/member), trimiterea aceluiași mesaj mai multor clienți odată, implementarea unei liste de prieteni, etc.

6 Bibliografie

Documentație pentru partea de implementare a clientului și serverului
cu **thread-uri/detalii teoretice** - <https://profs.info.uaic.ro/~computernetworks/cursullaboratorul.php>

Tool pentru realizarea diagramelor - <https://app.diagrams.net/>

Platforma folosită pentru unele probleme întâmpinate la implementarea aplicației - <https://stackoverflow.com/>

Diferite aspecte practice/teoretice - <https://profs.info.uaic.ro/~ioana/>