

Proiectarea arhitecturala

TotalShare

Agiali Taygun, 331CC

Dimcica Tudor, 331CC

Datcu Andrei, 331CC

Cuprins

1. INTRODUCERE.....	2
1.1 Obiective	2
1.2 Overview	2
2. TEHNOLOGII.....	2
2.1 C++	2
2.2 Qt.....	3
2.3 Socketi	4
2.4 Microsoft Visual Studio	5
2.5 Git	5
3. DESCRIERE GENERALA	6
3.1 Front end (Client)	6
3.2 Back end (Server)	7
4. EXPLICAREA FUNCTIONALITATILOR	7
4.1 Conectare	7
4.2 Transferul fisierelor	8
4.3 Interactiunea modulelor	8

1. Introducere

1.1 Obiective

Obiectivul acestui document este de a face o analiza a sistemului pentru implementarea aplicatiei de file-sharing.

Se va face o trecere in revista a tehnologiilor folosite si o descriere a modalitatii de realizare efectiva a viitorului sistem la nivel inalt.

Se vor prezenta modulele ce alcatuiesc sistemul, relatiile dintre acestea si cateva situatii de utilizare de reliefeaza caracteristicile importante ale aplicatiei.

1.2 Overview

Capitolul 1 reprezinta partea de descriere a documentului curent.

Capitolul 2 contine lista tehnologiilor si resurselor folosite, cu o mica descriere la fiecare, motivarea alegerii lor si explicarea contributiei acestora la aplicatie.

Capitolul 3 prezinta modulele folosite, iar capitolul 4 prezinta interactiunea dintre ele pentru indeplinirea anumitor functii.

2. Tehnologii

2.1 C++

C++ este un limbaj de programare general, compilat. El foloseste o multi-paradigma, suportand programare procedurala, programare orientata pe obiecte si abstractizare a datelor. A fost dezvoltat de Bjarne Stroustrup in anii 1980, pornind de la limbajul C deja existent si adaugandu-i diverse imbunatatiri.

Am ales acest limbaj tocmai pentru ca este unul puternic, care permite manipularea cu usurinta a datelor de dimensiuni mari, gestiunea memoriei si a comunicarii in retea.

Avantajele acestui limbaj pentru proiectul curent sunt:

- un API pentru lucrul cu socketi (prezentat mai jos)
- Standard Template Library, o colecție de containere și algoritmi folositori pentru lucrul cu multe date
- modalități de a lucra cu threaduri și de folosi facilități ale sistemului de operare Linux: creare procese, lucrul cu memorie mapată, etc.

De asemenea, toți membrii echipei sunt obișnuiți și experimentați cu C++, ceea ce înseamnă un workflow bun și o dezvoltare rapidă a proiectului.

Am ales C++ și datorită posibilității de a îmbina programarea procedurală cu cea orientată pe obiecte, permitându-ne să creăm o arhitectură bună a modelelor de lucru, și a aplicației client-server.

Referințe: [C++](#), [Standard Template Library](#)

2.2 QT

Qt este un sistem inter-platformă de dezvoltare a programelor pentru calculator, folosit atât pentru crearea programelor cu interfață grafică cât și pentru programe fără interfață grafică, cum sunt serverele. Cele mai cunoscute utilizări ale Qt sunt KDE, browserul web Opera, Google Earth, Skype, Qtopia. Qt este produs de firma norvegiană Trolltech.

Începând cu versiunea 4, Qt este disponibil pentru toate platformele, inclusiv Windows sub o licență duală, GPL și o licență proprietară. Acest fapt face posibilă dezvoltarea de aplicații Qt licențiate GPL și pentru alte sisteme, cel mai notabil efect fiind disponibilitatea KDE 4 pentru Windows și Mac OS X.

Pentru a crea o interfață grafică utilizând QT sunt necesari următorii pași:

- includerea header-ului qt (și ulterior linkarea cu biblioteca partajată)
- inițializarea modului, utilizând QApplication
- crearea unui QWidget pentru fereastra principală
- crearea de QWidget-uri pentru câmpurile dorite și aranjarea acestora în cadrul ferestrei
- setarea proprietății de vizibilitate
- rularea QApplication.exec

Am ales Qt pentru că este folosit pe scară largă și portat pe majoritatea

platformelor. Deasemenea, fiind un proiect open-source cu licență GPL ne este permis să utilizăm și integrăm în aplicație. Acesta este un avantaj major și pentru utilizatori, întrucât poate fi ușor obținut și instalat, fără a necesita achiziția de software suplimentar.

Utilizarea sistemului de file sharing va impune ca dependență instalarea pachetelor Qt pentru sistemul de operare folosit.

2.3 Socketi

Un socket este partea finală a comunicării inter-procese în rețele de calculatoare. În internet, fiecare calculator este recunoscut printr-o adresă IP, iar la nivelul acestuia, fiecare proces ce comunică în rețea este recunoscut printr-un port. Astfel, un socket este perechea (ip, port) ce definește o aplicație în internet.

Un API de socketi este o interfață de programare, de obicei prevăzută de sistemul de operare, care permite unei aplicații să folosească în mod direct socketii de rețea. API-ul de socketi pentru C++, pe lângă comenzile obișnuite de comunicație, suportă și niste rutine de gestionare a unui număr mare de conexiuni la un moment dat, pentru a asigura o performanță bună a întregii aplicații.

Din această cauză, socketii au fost cea mai bună alegere pentru comunicarea între clienți și server.

Clienții, pe rând, se vor conecta la server, care va avea datoria de a le accepta conexiunile, și vor iniția un flux de comunicație cu acesta.

Serverul va folosi un socket pentru a primi date/cereri de la clienți - el va trebui să se ocupe de gestionarea tuturor cererilor și să păstreze un echilibru în comunicarea cu acestea, pentru ca niciunul să nu fie dezavantajat.

Gestiunea conexiunilor se va face asincron, folosind event-loopul creat de frameworkul Qt. Astfel la fiecare nouă conexiune/cerere ajunsa la server se va crea un semnal care va fi tratat prin lansarea unei metode într-un alt thread care va rezolva cererea. Astfel, serverul va putea prelua cereri de la mai mulți clienți în paralel.

2.4 Microsoft Visual Studio

Visual Studio include un set complet de instrumente de dezvoltare pentru generarea de aplicații ASP.NET, Servicii Web XML, aplicații desktop și aplicații mobile. Visual Basic, Visual C++, Visual C# și Visual J# toate folosesc același mediu de dezvoltare integrat (IDE) care le permite partajarea instrumentelor și facilitează crearea de soluții folosind mai multe limbaje de programare. Aceste limbaje permit să beneficieze de caracteristicile .NET Framework care oferă acces la tehnologii cheie care simplifică dezvoltarea de aplicații web ASP și XML Web Services cu Visual Web Developer.

Am ales să folosim Visual Studio pentru că facilitează foarte mult dezvoltarea proiectului. Una dintre cele mai importante facilități oferite de Eclipse este cea de autocomplete. Aceasta este foarte bine dezvoltată, sugestiile date fiind foarte exacte și potrivite contextului.

2.5 Git

Git este un sistem de versionare al codului sursă distribuit care pune accent pe viteză. Inițial, a fost dezvoltat de Linus Torvalds pentru Linux Kernel, ulterior fiind adoptat în mai multe proiecte. Git este open source.

Câteva dintre avantajele majore ale git sunt:

- este distribuit și scalabil
- lucrează foarte bine cu proiecte de dimensiune mare
- ușor de folosit pentru proiecte puternic ramificate, care au mai multe branch-uri implementează multe strategii de rezolvare a conflictelor
- autentificare securizată, peste HTTPS și SSH

Câteva dintre operațiile întâlnite frecvent în cadrul unui repo git:

- `git init --bare` crează repo-ul inițial.
- `git clone <REPO>` clonează un repo deja existent, copiind sursele în directorul curent
- `git add <FILES>` adaugă fișiere (sau modificări ale fișierelor existente) pentru a fi salvate în repo
- `git commit` crează un commit cu toate fișierele ce au fost adăugate, este necesar un commit message
- `git push` uploadează commit-urile curente în repo-ul remote

`git pull` sincronizează repo-ul local cu remote-ul, downloadează commit-uri.

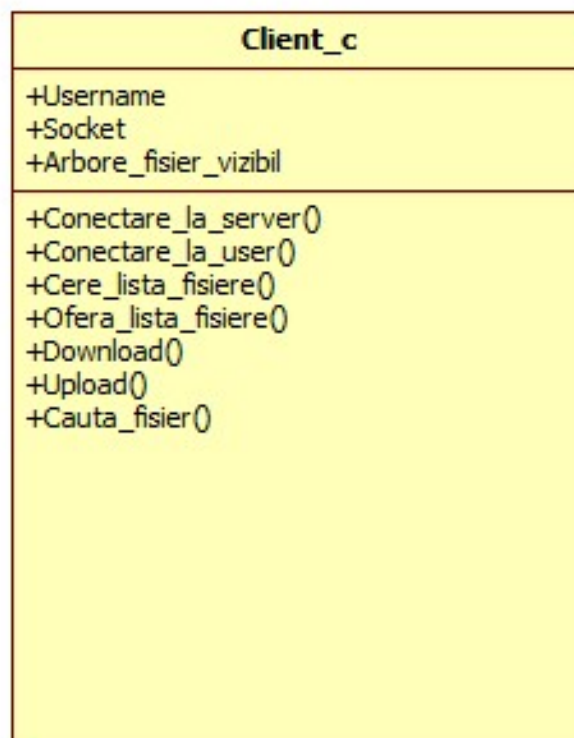
3 Descriere generala

Implementarea se bazează pe existența a două module principale:
Front-End (Client) și Back-End (Server).

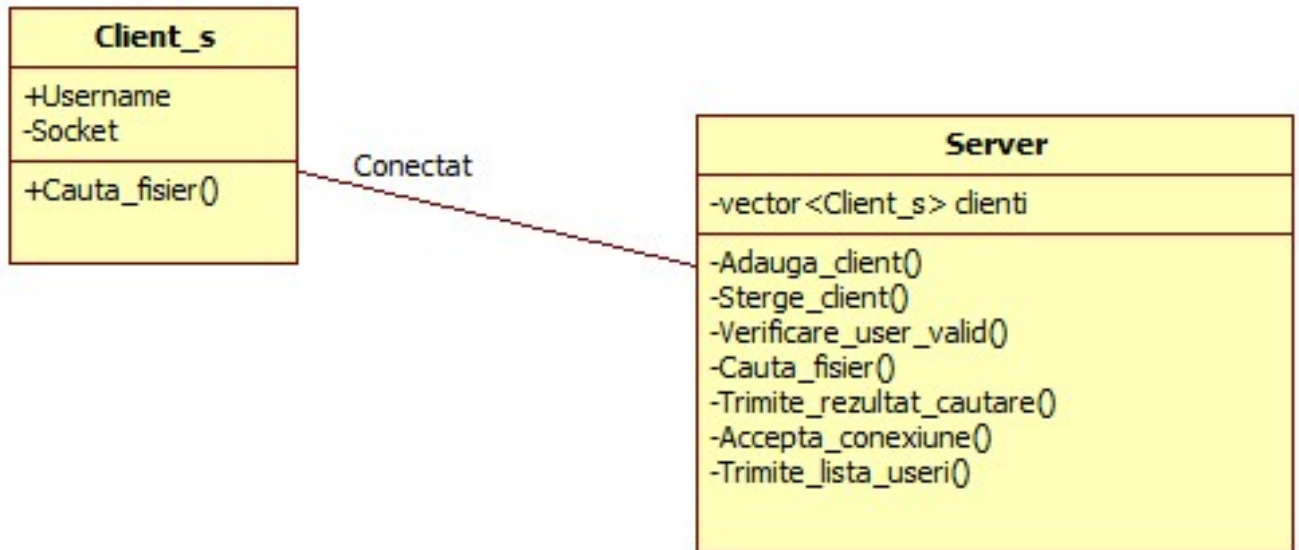
Funcționalitatea lor trebuie să fie conceptual diferită.

- modulul de Front-End trebuie să permită o clientului o comunicare simplă și cât mai intuitivă. De asemenea trebuie să prevină orice utilizare incorectă a utilizatorului. Acest modul trebuie să producă (și să afișeze) mesaje de eroare cât mai intuitive (eventual cu exemple) pentru a ghida userul către utilizarea corectă a sistemului.
- modulul de Back-End este responsabil cu gestiunea utilizatorilor. De asemenea acest modul este responsabil cu multiplexarea datelor astfel încât, în momentul în care există mai mulți clienți ce interoghează serverul, caută fișiere, etc. aceștia să aibă senzația că răspunsul la cereri se transmite simultan (și în egală măsură). De asemenea acesta trebuie să implementeze mecanisme de securitate pentru protecția datelor utilizatorilor și a fișierelor acestora.

3.1 Front-end (Client)



3.2 Back-end (Server)



4 Explicarea functionalitatilor

4.1 Conectare

Pentru a se conecta si a vizualiza fisierele altor utilizator, un utilizator are doua variante:

4.1.1 Conectare in mod public

Daca utilizatorul doreste sa vada lista celorlalti utilizatori din sistem, atunci se poate conecta la server, alegandu-si un username. Daca actiunea se va termina cu success, atunci clientul poate cauta si vizualiza fisierele tuturor celorlalti utilizatori conectati in modul public. In caz contrar, va primi un mesaj sugestiv de eroare(Ex: "username deja existent", "serverul nu raspunde", etc.)

4.1.2 Conectare in mod privat

Daca utilizatorul doreste sa vada fisierele unui alt utilizator si

cunoaste ip-ul acestuia atunci se poate conecta doar la acest utilizator in mod privat, fara a se conecta la server si fara a fi vizibil altor utilizatori.

4.2 Transferul fisierelor

Transferul de fisiere se va realiza doar intre doi client, indiferent daca acestia sunt conectati modul privat sau public. Un client, va cere, dupa vizualizarea listei de fisiere a celui alt utilizator unul sau mai multe fisiere spre descarcare. Acesta le va trimite pe rand.

Erorile care pot aparea se pot datora unor conexiuni defectuoase sau unor coliziuni, erori de trimitere. Aceste cazuri de eroare sunt tratate in mare de protocolul TCP si de asemenea se verifica un cod de checksum (MD5SUM) pentru corectitudinea fisierelor in momentul finalizarii transferului. In cazul in care datele se pierd, si eroarea nu poate fi tratata “on the run” se reinitializeaza transferul fisierului respectiv. Cu toate astea, se incearca evitarea acestui caz prin impartire in blocuri de dimensiune fixa si verificarea a “portiunii” din mesaj. De asemenea se garanteaza trimiterea in ordine a pachetelor astfel incat, la destinatie se realizeaza o simpla concatenare a mesajelor, neretinandu-se intern un numar de ordine a pachetului in cadrul fisierului trimis.

4.3 Interactiune module

