**Sistem de file-sharing**

**Documentul de Proiectare Arhitecturala**

**Echipa:**

**Dogaru Constantin**

**Gherghina Cristian-Catalin**

**Macarie Roxana Georgiana**

**Manciu Stefan**

**Cuprins**

1. Introducere………………………………………………………………………………………………….3

2. Obiective de proiectare…………………………………………………………………………..…..4

3. Arhitectura propusa

3.1 Prezentare generala a arhitecturii sistemului………………………..………...5

3.2 Decompozitia in subsisteme si responsabilitatile fiecarui subsistem..6

3.3 Distributia subsistemelor pe platforme hardware/software…………....7

3.4 Managementul datelor persistente………………………………………………….8

3.5 Controlul accesului utilizatorilor la sistem.……………………………………….9

3.6 Fluxul global al controlului……………………………………………………………..10

3.7 Conditiile limita………………………………………………………………………………11

**1. Introducere**

**1.1 Scopul sistemului**

JTorrent este un program opensource (free), prin urmare orice utilizator este liber pentru a downloada codul-sursa si a-l modifica. Partajarea de fisiere nu este restrictionata sau limitata, dar este recomandat ca fisierele pe care le puneti in Sharing sa fie materiale care respecta drepturile de autor.  
 Sistemul va permite utilizatorilor sa puna in comun (share) mai multe fisiere. De asemenea va oferi posibilitatea de a cauta prin lista partajata de fisiere ale celorlalti utilizatori si de a descarca fisierele de interes.

File-sharing-ul ofera accesul la informatia dorita mult mai usor si rapid. Datorita acestui fapt se vor consuma mai putin timp si efort si in acest fel performanta altor taskuri va creste.

**1.2 Definitii, acronime**

**File-sharing** = sistem de partajere a resurselor intre utilizatori (clienti)

**Download** = descarcarea (copierea/clonarea) unui fisier de interes aflat pe un sistem al altui utilizator pe sistemul propriu. In urma actiunii de download cu succes, va exista o copie a acelui fisier.

**Upload** = urcarea (punerea in share) unui fisier in lista de fisiere disponibile pe server

**Search** = cautarea dupa anumite cuvinte cheie si returnarea unei liste cu cele mai bune rezultate

**1.3 Documente referinte**

* Documentul de Specificatie a Cerintelor
* Document de Planificare

**2. Obiective de proiectare**

Proiectul trebuie sa realizeze cu succes partajarea resurselor intre utilizatori. Astfel este necesara stabilirea unei conexiuni cu serverul si cel mai important este ca fiecare utilizator sa aiba acces la internet in momentul in care foloseste aplicatia.

In momentul in care utilizatorul porneste pentru prima data JTorrent, acesta va trebui sa isi creeze un cont. Astfel actiunea de **register** este necesara pentru o viitoare autentificare a clientului.

Autentificarea clientului nu necesita parola, intrucat se va mapa statia utilizatorului cu numele (username-ul) pe care acesta l-a ales in momentul inregistrarii.

Inainte de a putea realiza orice **download**, implementarea actiunii de **upload** este obligatorie. Aceasta trebuie sa se realizeze cu succes si serverul sa aiba o lista valida si actualizata cu toate fisierele partajate pana in acel moment.

Pentru a putea alege un fisier de interes din acea lista, este necesar ca serverul sa poate distribui tuturor clientilor interesati acea lista pe care o detine. Ulterior se va implementa actiunea de **search** care va rafina cautarea unui utilizator si ii va returna o lista doar cu posibilele fisiere de interes in loc de toata lista disponibila, navigarea printre acestea devenind astfel mult mai usoara.

In momentul in care un utilizator are o lista de fisiere partajate care sunt puse in comun (share), acesta va putea **downloada** fisierele pe care le doreste. Initial acesta poate downloada doar un fisier, ulterior va putea downloada mai multe fisiere in paralel.

Server-ul va retine o lista cu toate fisierele partajate pana in momentul acela, neavand nevoie ca la actiunile upload/download fisierele sa treaca “prin el”. Conexiunea va fi **Peer 2 Peer**.

**3. Arhitectura propusa**

**3.1 Prezentare generala a arhitecturii sistemului**

Arhitectura software va fi divizita in 2 subsisteme principale:

* **JTorrent** server care se va ocupa de realizarea conexiunii *client-client*(pentru actiunile de download si upload) si *client-server*(pentru actiunile de search si inregistrare username).

Folosind Command Pattern, toate cele 4 comenzi vor extinde aceeasi interfata Command si vor implementa actiunea execute asociata interfetei.

Pentru comanda de *download* flow-ul va fi in felul urmator: client-ul se conecteaza la server si il anunta ca vrea sa se conecteaza la alt client pentru a descarca un fisier de la acesta. Serverul care tine o evidenta a tuturor userilor conectati/online va cauta clientul cu pricina si va realiza conexiunea intre cei doi. Actiunea este similara pentru comanda de *upload.*

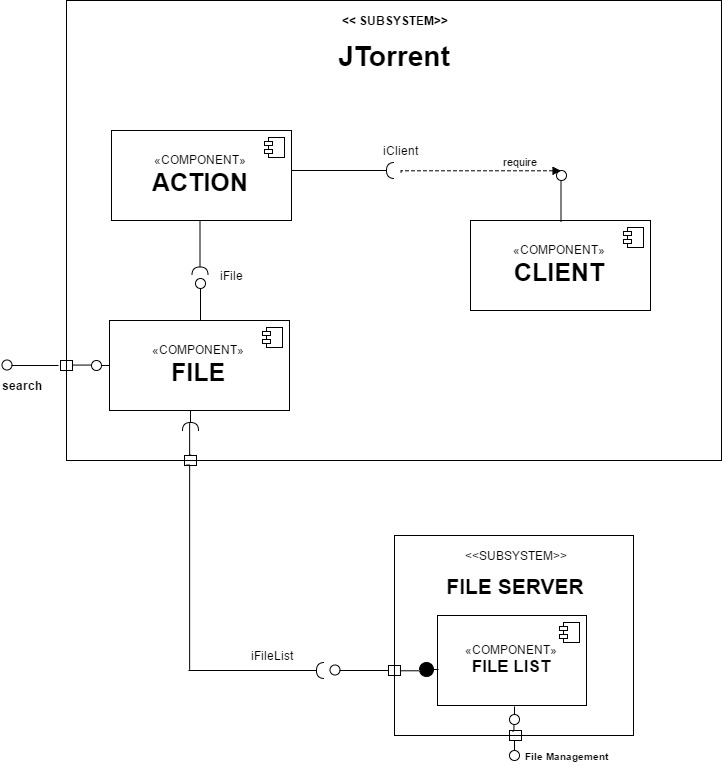
In cazul comenzii de *inregistrare user nou*, conexiunea va fi client-server deoarece clientul nu are nevoie decat de permisiunea serverului pentru a sti daca numele pe care incearca sa-l utilizeze este disponibil. In caz afirmativ, serverul ii intoarce un cod STATUS\_OK si operatiunea este realizata cu succes. In caz contrar serverul intoarce un STATUS\_NOT\_OK iar utilizatorul va fi rugat sa incerce un alt user name.

Pentru actiunea de *search*clientul ii cere serverului o lista care sa tina cont de niste criterii de cautare (nume, dimensiune, data partajare etc.). Serverul, mai departe, va face un query in baza de date si va extrage de acolo toate fisierele ce indeplinesc conditiile respective.

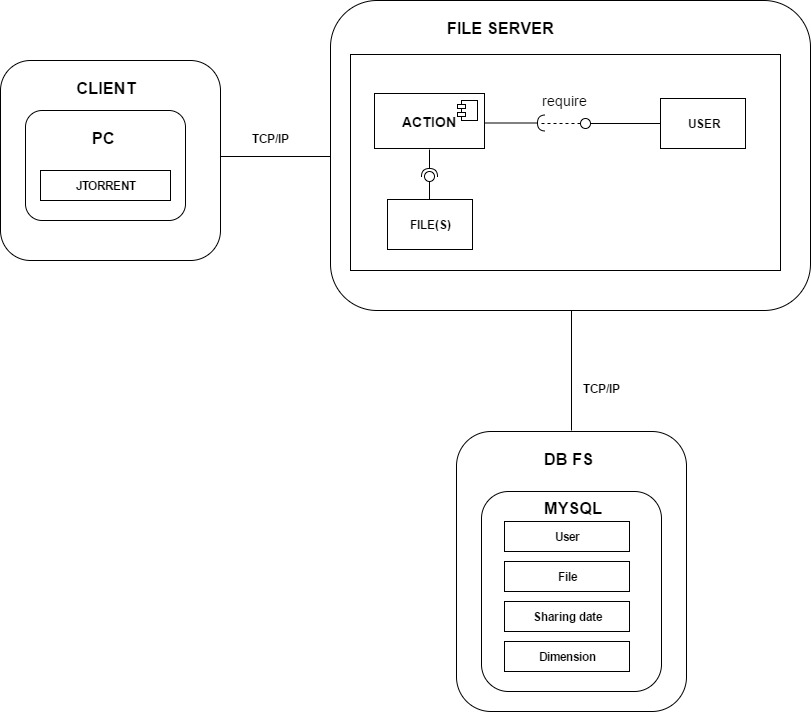
* **FileServer** cu conexiune la baza de date**.** Aceasta va dispune de 2 tabele principale. In prima vor fi salvati toti utilizatorii inregistrati in sistem si va contine trei coloane: id, nume si data inregistrare. Cea de-a doua tabela va pastra informatii privitoare la fisiere, informatii cum ar fi nume fisier, dimensiune, utilizatorul care a partajat fisierul, data la care a fost partajat etc.

Subsistemul FileServer va avea 2 clase principale: o clasa Connection ce se va ocupa de conectarea la baza de date si o clasa Repository ce va cuprinde query-urile pentru comanda search.

**3.2 Decompozitia in subsisteme si responsabilitatile fiecarui subsistem**



**3.3 Distributia subsistemelor pe platforme hardware/software**



**3.4 Managementul datelor persistente**

Pentru a salva configuratia sistemului de la un anumit moment, aplicatia JTorrent va folosi o baza de date in care va stoca utilizatorii si fisierele partajate de acestia.

Baza de date va fi implementata folosind SQL Server.

* **Schema conceptuala a bazei de date**



**3.5 Controlul accesului utilizatorilor la sistem**

Pentru a avea acces la serviciile oferite de aplicatie, utilizatorul trebuie sa aiba un username existent in sistem.

In momentul autentificarii serverul verifica daca username-ul introdus este existent in sistem. In caz afirmativ, acesta are acces la serviciile oferite de aplicatie: download, upload, search.

Un alt tip de utilizator este acela de administrator. Acesta poate sa gestioneze conturile deja existente, interzice accesul asupra unor fisiere sau sa modifice setarile sistemului.

**3.6 Fluxul global al controlului**

Actiunea de download din sistem va folosi o arhitectura de **Peer-2-Peer**, astfel fluxul controlului in fiecare subsistem va fi unul independent exceptand sincronizarile necesare pentru tratarea cererilor. In cazul in care apar modificari in baza de date, utilizatorii vor primi notificari.

De asemenea si celelalte actiuni vor fi independente, dar vor trebui mereu sincronizate cu baza de date. De exemplu actiunea de search va returna un rezultat corect doar in momentul in care server-ul dispune de o baza de date actualizata. De asemenea actiunea de upload modifica baza de date, astfel ea va trebui sincronizata inainte ca un alt user sa dea search.

**Download**

Utilizatorul va da dublu click pe fisierul dorit si este lansat urmatorul fir de executie: utilizatorul numarul 1 va face un request catre server pentru a initia o conexiune cu utilizatorul numarul 2. In cazul in care serverul a reusit sa initieze conexiunea, se va incepe descarcarea fisierului.

**Upload**

Utilizatorul deschide aplicatia si alege fisierul pe care doreste sa-l partajeze. Apasa pe butonul de upload si se initiaza conexiunea la server. Daca conexiunea a reusit si sunt validate toate conditiile de upload, atunci fisierul va fi incarcat pe server si se va returna un mesaj de succes

**Adaugare utilizator nou**

Primul pas este ca utilizatorul nou sa deschida aplicatia iar in acest fel se conecteaza la server. Isi alege un username iar daca acesta este unic, utilizatorul este inregistrat cu succes, altfel acesta este rugat sa introduca alt username. Dupa realizarea fluxului cu succes, administratorul are rolul de a introduce username-ul in sistem.

**Search**

Utilizatorul deschide aplicatia, se conecteaza la server prin intermediul username-ului si ii apare interfata grafica a aplicatiei cu toate functionalitatile. Acesta doreste sa caute un fisier si apasa pe optiunea de search. Poate sa caute un fisier dupa nume sau are la dispozitie optiunea de cautare avansata. Dupa ce a introdus parametrii pentru cautare ii este afisata o lista cu rezultatele gasite.

**3.7 Conditiile limita si tratarea lor (cazurile de utilizare limita)**

Conditiile limita:

* pornirea/oprirea serverului
  + La oprirea serverului se salveaza configuratia actuala a sistemului.
  + La pornirea acestuia, se incarca ultima configuratia valida a serverului.
* deschiderea/inchiderea conexiunii cu baza de date
  + In cazul in care conexiunea cu baza de date a esuat sau modificarea acesteia, actiunea nu se ia in considerare, iar sistemul ramane in acelasi stadiu ca inainte de modificare, garantand astfel functionarea in permanenta a aplicatiei.
* deconectarea utilizatorului in timpul actiunii de download
  + Utilizatorul va repeta procesul de download pana cand actiunea va fi finalizata cu succes