Sistem de file-sharing

- Specificația sistemului -

Versiune intermediară - 25.03.2013

TheGunn3rs:

Țucă Laurențiu - 333 CA

Șerban-Rădoi Constantin - 333 CA

Culcuș Mihaela-Nicoleta - 333 CC

Petrescu Rareș - 333 CC

Îndrumător:

Oana Maria Ferche

Cuprins

[1. INTRODUCERE 4](#_Toc352023821)

[1.1 Scopul documentului 4](#_Toc352023822)

[1.2 Definirea termenilor 4](#_Toc352023823)

[1.3 Rezumatul documentului 4](#_Toc352023824)

[2. DESCRIEREA GENERALĂ A PRODUSULUI 5](#_Toc352023825)

[2.1 Situația curentă 5](#_Toc352023826)

[2.2 Scopul produsului 5](#_Toc352023827)

[2.3 Descrierea utilizatorilor 5](#_Toc352023828)

[2.4 Trecere în revistă ale funcțiilor îndeplinite de sistem 6](#_Toc352023829)

[2.4.1 Serverul central care oferă urmatoarele funcționalități: 6](#_Toc352023830)

[2.4.2 Clientul (aplicația de bază) care oferă următoarele funcționalități: 6](#_Toc352023831)

[2.5 Constrângeri 6](#_Toc352023832)

[2.6 Beneficii 6](#_Toc352023833)

[3. SPECIFICAREA CERINȚELOR FUNCȚIONALE 6](#_Toc352023834)

[3.1 Abstract 6](#_Toc352023835)

[3.2 Cazuri de utilizare 7](#_Toc352023836)

[3.2.1 Conectarea la server 8](#_Toc352023837)

[3.2.2 Adăugare fișiere partajate 9](#_Toc352023838)

[3.2.3 Vizualizare utilizatori conectați 9](#_Toc352023839)

[3.2.4 Conectarea la utilizator și listarea de fișierelor 9](#_Toc352023840)

[3.2.5 Descărcare fișiere de la un utilizator 10](#_Toc352023841)

[3.2.6 Trimiterea de mesaje text 10](#_Toc352023842)

[3.2.7 Deconectarea de la server 11](#_Toc352023843)

[3.3 Alte cazuri de utilizare și extensibilitate (optional) 11](#_Toc352023844)

[4. ARHITECTURA SISTEMULUI 11](#_Toc352023845)

[4.1 Specificații hardware 11](#_Toc352023846)

[4.2 Specificații software 11](#_Toc352023847)

[5. MODELAREA DATELOR 11](#_Toc352023848)

[6. SPECIFICAREA CERINȚELOR NEFUNCȚIONALE 12](#_Toc352023849)

[6.1 Cerințe operaționale 12](#_Toc352023850)

[6.2 Cerințe de interfață 12](#_Toc352023851)

[6.2.1 Interfața utilizator 12](#_Toc352023852)

[6.2.2 Interfață comunicare 13](#_Toc352023853)

[6.2.3 Cerinte hardware 13](#_Toc352023854)

[6.2.4 Cerințe software 13](#_Toc352023855)

[6.2.5 Interfața login 14](#_Toc352023856)

[6.3 Cerințe de performanță 14](#_Toc352023857)

[6.4 Cerințe de calitate 14](#_Toc352023858)

[6.4.1 Portabilitatea 14](#_Toc352023859)

[6.4.2 Disponibilitatea: 14](#_Toc352023860)

[6.4.3 Adaptabilitatea: 15](#_Toc352023861)

[6.5 Cerințe de securitate 15](#_Toc352023862)

[6.5.1 Securitatea datelor: 15](#_Toc352023863)

[6.5.2 Securitatea în funcționare: 15](#_Toc352023864)

[6.6 Cerințe de planificare 15](#_Toc352023865)

# INTRODUCERE

## Scopul documentului

Acest document a fost creat pentru a reda o specificație căt mai realistă în ceea ce privește cerințele ș1 constrângerile de implementare ale unui sistem ce va permite o comunicație peer-to-peer între doi utilizatori legați la un server, în scopul descărcării de fișiere.

Documentul este destinat celor care vor implementa sistemul specificat în continuare.

## Definirea termenilor

**Sistem -** produsul actual;

**Actor -** rol pe care o entitate externă produsului îl joacă în relație cu sistemul;

**LAN -** Local Area Network;

**Server -** serverul central ce menține o bază de date dinamică cu IP-urile celor conectați la acesta;

**Client -** entitate/program ce se conectează la server în scopul de a afla adrese IP ale altor clienți conectați la server; conține printre altele și functiile de miniserver/miniclient, definite mai jos;

**Miniclient -** entitate client pentru simularea sistemului client/server în cadrul comunicației P2P; este cel care face cererea pentru a-i fi trimis un fisier și cel ce primește un raspuns la aceasta cerere;

**Miniserver -** entitate server ce primește cereri de la un miniclient și îi trimite lista de fișiere/fișierul cerut.

## Rezumatul documentului

Capitolul 2 conține o descriere generală a produsului, scopul acestuia corespunzător la nevoile actuale de pe piață, descrierea categoriilor de utilizatori care îl pot folosi și funcțiile pe care produsul le indeplinește.

Capitolul 3 conține specificarea cerințelor funcționale ale produsului.

Capitolul 4 conține câteva mici specificații ale arhitecturii sistemului.

Capitolul 5 conține aspecte privind modelarea datelor sistemului.

Capitolul 6 conține principalele cerințe nefuncționale.

# DESCRIEREA GENERALĂ A PRODUSULUI

## Situația curentă

Pe piața produselor software nu există în momentul de față un sistem multiplatformă pentru comunicarea de tip DC/DC++(Direct Connect) într-o rețea. De aceea considerăm că apariția unui astfel de sistem va fi binevenită, cu atat mai mult cu cât este și necesară.

Deși unele functionalități se pot regăsi și în alte programe, aplicația nu-și dorește înlocuirea unui alt produs deja existent pe piață.

## Scopul produsului

Produsul specificat în continuare va oferi pe lângă portabilitate, un sistem eficient, intuitiv de schimb de fișiere între utilizatori.

Fiecare client va distribui o listă cu fișiere din calculatorul său pe care dorește să le împartă cu ceilalți utilizatori. Un utilizator accesează lista de fișiere a oricărui alt client din sistem și descarcă orice dorește și îi este necesar din acea listă pusă la dispoziție.

Oamenii sunt încântați de sistemele de file-sharing, fiind o modalitate gratis, ușoară și rapidă de a găsi ceea ce cauți: filme, cărti, muzică, programe, documente etc.

Programul se poate realiza și utiliza fără niciun cost adițional, nefiind necesare echipamente hardware adiționale, iar software-ul folosit în crearea lui fiind disponibil în mod gratuit.

## Descrierea utilizatorilor

Ținta nu este o anumită piață de utilizatori, dintr-un anumit domeniu poate, ci toți oamenii, indiferent de activitatea desfășurată.

Clienții produsului sunt reprezentați de toți cei care au acces la o rețea , fie ea publică sau privată. Produsul nostru funcționeaza atât într-o rețea locală, cât și în Internet și va fi construit în așa fel încat să poată fi folosit ușor de orice persoană capabilă de a lucra la un nivel mediu cu un computer. Nu este necesar niciun echipament sofisticat, doar un calculator și acces la internet sau la o rețea locală. Nu sunt necesare nici cunostinte tehnice deosebite.

Exemplu de posibili utilizatori:

• studenți ce folosesc aplicația pentru a găsi documente necesare pentru proiectele de la facultate;

• ingineri ce folosesc aplicația pentru a găsi diverse cărți, programe necesare activității lor sau pur și simplu pentru muzică, filme etc;

• economist, profesor, șofer, contabil etc. (persoane din absolut orice domeniu).

Deși este accesibil tuturor, în urma unui studiu se estimează că majoritatea clienților vor fi totuși adolescenți, elevi, studenți, aceștia fiind cei care folosesc Internetul în cea mai mare măsură.

## Trecere în revistă ale funcțiilor îndeplinite de sistem

Există două parți independente ale sistemului, descrise în continuare:

### Serverul central care oferă urmatoarele funcționalități:

* interfață în linie de comandă;
* bază de date limitată ce conține informații de bază despre utilizatorii conectați la un anumit moment;
* blocare utilizator în cazul share-uirii de materiale interzise, folosire cuvinte obscene etc.;
* oferirea către un user a datelor de conectare la un alt user(IP/port).

### Clientul (aplicația de bază) care oferă următoarele funcționalități:

* interfața grafică user-friendly;
* miniclient (search, request, receive);
* miniserver (answer, send);
* salvare history;
* chat minimal.

## Constrângeri

Va exista o limită de utilizatori ce se vor putea conecta la un moment dat la serverul central.

De asemenea, viteza de transmisie/recepție a fișierelor va fi mărginită superior de performanțele rețelei în care functionează sistemul creat de noi.

## Beneficii

Sistemul de față este dedicat transferului de fișiere în Internet sau într-o rețea locală printr-o comunicație peer-to-peer (punct la punct) între entități de tip client. Faptul că sistemul va fi dezvoltat in Java, deci va fi multiplatformă, va constitui un avantaj pentru alegerea ca sistem principal de file-sharing în companii, deoarece diferite departamente lucreaza în diferite sisteme de operare.

# SPECIFICAREA CERINȚELOR FUNCȚIONALE

## Abstract

Se dorește implementarea unei soluții care să permită conectarea mai multor utilizatori prin intermediul unui server cu scopul de a interschimba fișiere și alte informatii utile. De asemenea, se dorește o abstractizare a comunicării dintre interfața grafică și partea de backend a soluției.

Vor exista două părți a aplicației:

* Backend
* Frontend (Interfata grafica)

**Backend**-ul se va ocupa de partea nevăzută de utilizator, și va reprezenta nucleul aplicației. Aici se vor crea conexiunile dintre utilizatori și server, respectiv dintre utilizatori și alți utilizatori. Partea de backend constă și ea la rândul ei din două entități:

#### entitatea server care va fi ea însăși o aplicație independentă, în linie de comandă, eventual și cu o interfață limitată, care va aștepta conexiuni de la entitățile client. Scopul server-ului este de a facilita interconectarea dintre utilizatori. El va furniza utilizatorilor (clienților) informațiile necesare pentru ca aceștia să poată crea legături peer to peer. Practic server-ul este un intermediar dintre utilizatori.

#### entitatea client care va consta din aplicația principală, pusă la dispoziția utilizatorului. Din cadrul aplicației client, se vor putea face mai multe operații, precum:

* Conectare la un server (pot exista mai multe servere);
* Listare utilizatori conectați pe serverul curent;
* Cerere informații conectare către un anumit utilizator de la server;
* Trimitere mesaje către un anumit utilizator;
* Listare fișiere shared ale unui utilizator;
* Descărcare fișiere de la un utilizator;
* Editare lista fișiere shared proprie.

**Frontend**-ul se va ocupa de partea vizibilă utilizatorului, anume de modul în care vor fi afișate butoanele, fișierele și alte informații din aplicație. El constă într-o interfață grafică pentru operațiile descrise de entitatea client. Între cele două părți va exista o interfață de comunicare, astfel că partea de backend să funcționeze cu orice frontend care folosește interfața dată, putându-se crea cu ușurință mai multe moduri de vizualizare pentru aceleași date.

## Cazuri de utilizare

După cum am prezentat în abstract, vor exista mai multe scenarii de utilizare.

Un client se poate conecta la un server, vizualiza utilizatorii deja conectați la acel server, adăuga directoare partajate care să poată fi descărcate de alți utilizatori. De asemenea, se poate conecta la un anumit utilizator pentru vizualizare fișiere, descărcare fișiere, respectiv pornire o conversație de tip chat.

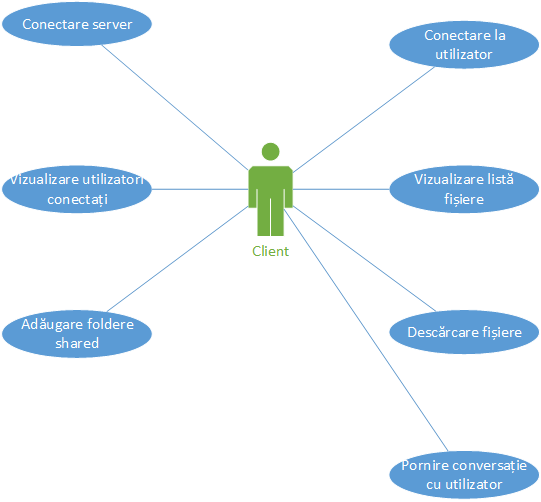


Figura 3.2.a - Usecase 1

Flow-ul descris anterior pentru utilizatori este cel din figura 3.2.2.

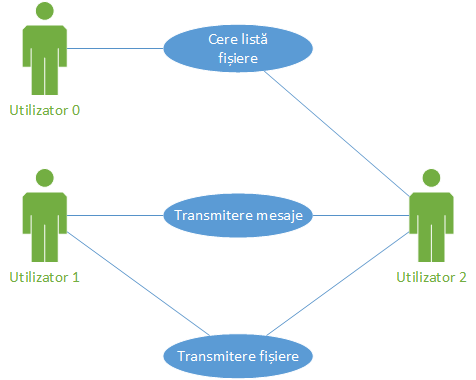


Figura 3.2.b - Usecase 2

### Conectarea la server

Primul lucru pe care trebuie să îl facă un utilizator este să se conecteze la un server.

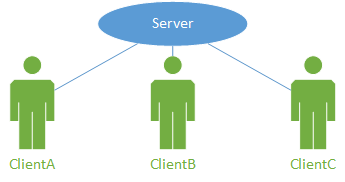


Figura 3.2.1 Conectare la server

Când un utilizator dorește să se conecteze la un Server, acesta trebuie să aibă un Nickname unic. În cazul în care pe server mai există un utilizator cu același Nickname, utilizatorul este avertizat că nu se poate conecta, și este sfătuit să încerce cu alt nume.

### Adăugare fișiere partajate

Pentru ca un utilizator (utilizator 1) să poată descărca fișiere de la un alt utilizator (utilizator 2), trebuie mai întâi ca utilizatorul 2 să partajeze aceste fișiere, ca ele să fie vizibile utilizatorului 1. Operația de adăugare fișiere partajate poate fi făcută oricând, dintr-un meniu corespunzător din aplicația Client.

### Vizualizare utilizatori conectați

Un utilizator îi poate cere server-ului oricând să îi afișeze o listă cu toți utilizatorii conectați la momentul respectiv.

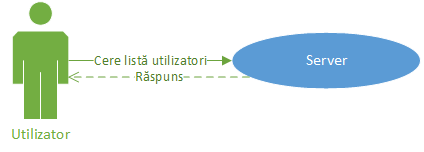


Figura 3.2.3 Listare utilizatori

### Conectarea la utilizator și listarea de fișierelor

În momentul în care un utilizator (utilizator 1) dorește să vadă ce fișiere partajate are un alt utilizator (utilizator 2), acesta se conectează la utilizatorul 2 și îi trimite acestuia o cerere corespunzătoare. Dacă nu există erori pe parcursul conectării, utilizatorul 2 îi răspunde utilizatorului 1, care va primi lista de fișiere.

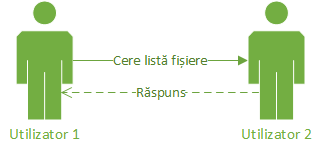


Figura 3.2.4 Listare fișiere

### Descărcare fișiere de la un utilizator

Odată ce a vizualizat lista de fișiere partajate, utilizatorul 1 poate să descarce de la utilizatorul 2 unul sau mai multe fișiere și să e salveze local într-un anumit director.

Există posibilitatea ca utilizatorul 2 să se deconecteze, sau să șteargă o parte din fișierele partajate, astfel că dacă în timpul încercării de descărcare se întâmplă un astfel de lucru, utilizatorul 1 este avertizat corespunzător, iar operația în curs eșuează. De asemenea, se poate și ca utilizatorul 1 să piardă legătura, caz în care se păstrează local doar fișierele descărcate complet.

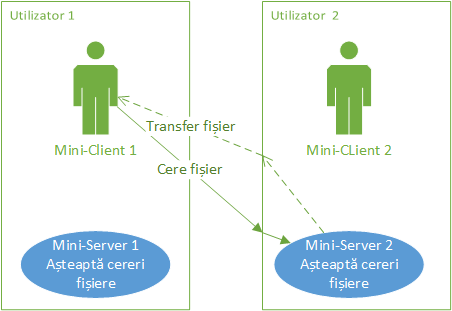


Figura 3.2.5 Listare fișiere

### Trimiterea de mesaje text

Pentru trimiterea de mesaje text, se va realiza o conexiune similară cu cea a transmiterii fișierelor. Utilizatorul 1 trimite mesaj la utilizatorul 2. Utilizatorul 2 are o entitate numită mini-server care așteaptă conexiuni de la orice utilizator, prin intermediul căreia el poate primi fie cereri, fie mesaje. Dacă unul din utilizatori pierde legătura, mesajul nu este transmis.

### Deconectarea de la server

Acest lucru se poate realiza fie la dorința utilizatorului, caz în care serverul își va modifica imediat lista de utilizatori conectați, iar toți utilizatorii care erau conectați la utilizatorul respectiv vor închide și ei legătura.

În cazul unei întreruperi de conexiune (accidentală, sau de altă natură), conexiunile curente vor rămâne deschise un timp prestabilit (de regulă câteva zeci de secunde). Dacă utilizatorul încearcă să se reconecteze 1la server înainte de expirarea timeout-ului, este avertizat că nu va putea să se conecteze și să încerce puțin mai târziu.

## Alte cazuri de utilizare și extensibilitate (optional)

Aplicația va putea fi extinsă pentru alte cazuri de utilizare, precum:

* Configurare și salvare setări utilizator în mod persistent;
* Criptarea informațiilor transmise;
* Limitarea vitezelor de upload/download în mod dinamic;
* Limitarea numărului maxim de conexiuni simultane.

# ARHITECTURA SISTEMULUI

## Specificații hardware

Ținând cont că proiectul va fi realizat exclusiv în Java , cerințele hardware sunt aceleași precum cele pentru rularea mașinii virtuale Java pe un sistem desktop/server. Mai multe detalii pot fi accesate pe site-ul oficial Java:

<http://www.java.com/en/download/help/sysreq.xml>

## Specificații software

Sistemul va consta din două aplicații distincte și o interfață de comunicație client-server. Aplicația server va rula (recomandabil) pe un sistem server dedicat, însa aceasta poate funcționa fără probleme și dacă este folosită ad-hoc. Aplicația client poate rula pe orice sistem desktop ce are instalat un sistem de operare compatibil cu cerințele din linkul de mai sus.

Baza de date din aplicația server este minimală, deoarece numărul userilor conectați este limitat de o valoare de 10000 de useri conectați simultan .

# MODELAREA DATELOR

Pe server va exista după cum este descris mai sus o bază de date dinamică ce reține utilizatorii conectați la un moment dat. Utilizatorii nu trebuie să își faca un cont pe server pentru a se putea conecta la acesta. Singura constrângere este data de numărul utilizatorilor conectați simultan.

Structura “User” din baza de date a serverului va conține următoarele câmpuri:

* Username
* IP
* port

Structura “Request to server” va conține doar username-ul utilizatorului conectat la server.

Structura “Answer from server” va conține înregistrarea utilizator luată din baza de date. Daca utilizatorul s-a deconectat între timp, structura va conține un mesaj de eroare.

Structura “RequestFile” corespunzătoare entității client conține următoarele câmpuri:

* FilePath
* PacketNumber [OPTIONAL] (în cazul implementării similare torrentelor) .
* Structura “ChatMessage” corespunzatoare entității client conține câmpurile:
* Username + “> “;
* Mesaj.

# SPECIFICAREA CERINȚELOR NEFUNCȚIONALE

## Cerințe operaționale

În cadrul aplicației client, interfața utilizator va fi intuitivă, user-friendly cu aspect similar cu alte aplicații dedicate aceluiași scop.

Aplicația server va conține un mini-shell de comenzi de tip restart, start, stop, ban <nickname>, update etc.

Datele nu vor fi stocate pe server; fiind un model peer to peer se transferă direct de la un client la altul, însă la înregistrare este necesar să se share-uiască o listă de fișiere.

Dacă sunt prea mulți utilizatori logați, aceasta va duce la viteze mult mai scăzute de descărcare, și la o încetinire a sistemului.

## Cerințe de interfață

### Interfața utilizator

Server – exista o interfata in linie de comanda, pentru diferite probleme de administrare, update, insa majoritatea operatiilor vor fi automate, inclusiv banarea utilizatorilor, asadar cerintele sunt minimala in ceea ce priveste acest aspect.

Client – interfata cu acesta este grafica (GUI). Este nevoie bineinteles de un monitor si de performante hardware adecvate rularii aplicatiei client.

#### Interfața se va împărți astfel:

##### într-un panou se afișează utilizatorii și dimensiunea listei lor de fișiere;

##### un panou cu descărcările active (va fi gol în cazul în care nu se descarcă nimic);

##### un panou cu o listă de fișiere (poate fi lista unui client, lista returnată de căutarea unei melodii de exemplu etc);

##### vom avea și posibilitatea de a face căutare după un anumit fișier dorit;

##### existența unui mini-chat pentru interacționarea cu utilizatorii;

#### Vom avea urmatoarele meniuri:

File:

- Settings (schimbare nickname, setare a directorului în care fișierele o să fie descarcate, share files)

- Search

- Exit

View:

- Finished downloads

- Finished uploads

Window: [Optional]

- Cascade

- Horizontal Tile

- Minimize all

Help:

- About

- Update

- Report bug

### Interfață comunicare

Comunicarea client-server va fi de același tip, iar comunicația client-client va fi de tipul peer-to-peer cu protocoalele aferente celor două tipuri de comunicații.

Rata de transfer în condiții de utilizare optimă va fi în medie de 1Mb/s.

### Cerinte hardware

Pentru rularea aplicației este necesar doar un calculator și o conexiune la Internet sau o rețea locală. Pentru o rulare optimă este indicată existența unei memorii RAM de minim 1GB.

### Cerințe software

Nu este necesară instalarea unor alte programe în afara sistemului de operare și a mașinii virtuale Java Runtime Environment (JRE).

### Interfața login

Pentru a folosi sistemul de fișiere, fiecare utilizator trebuie să se autentifice folosind un user și o parolă (autentificare clasica). În cazul în care sistemul recunoaște userul, acesta are acces la sistem. Altfel, sistemul nu recunoaște userul, deoarece acesta nu este încă înregistrat, se afișeaza un mesaj de eroare și operația de logare ia sfârșit.

## Cerințe de performanță

Server – datele vor fi reprezentate ca un HashMap cu înregistrări de tipul <UserName,IP+port> pentru accesarea lor eficientă.

Clientul va avea o listă cu fișiere/foldere partajate. Pentru ca un alt utilizator să facă o căutare eficientă prin ele, acestea se vor memora într-un TreeMap. Când sosește o cerere de livrare a unui fișier, prin intermediul cheii reprezentată de SharedPathName, se va accesa valoare Path, ce conține calea către fișierul ce va fi trimis.

Cea mai costisitoare operație va fi evident cea de search. Aceasta are în spate trimiterea către server a unei cereri de tip “Request All”, apoi căutari prin toate fișierele partajate de toți userii a cuvântului cheie căutat. Cu toate acestea, operațiile ar trebui să se facă rapid, datorită structurilor de date în care sunt memorate aceste informații.

## Cerințe de calitate

### Portabilitatea

Aplicația va fi portabilă pe orice sistem de operare (Windows, Linux, Mac OS), însă nu și pe terminalele mobile sau tablete. Există posibilitatea extinderii pe viitor și pe platforma Android, fiind bazat tot pe Java.

### Disponibilitatea:

Sistemul de file-sharing va fi adaptabil la cerințele clienților. Se dorește ca acesta să fie cât mai mult pe placul lor și se vor face sondaje regulate pentru a observa modificările dorite de aceștia.

Se intenționează ca sistemul să suporte până la 500 de utilizatori conectați în același timp, în orice zi a săptămânii, la orice oră. Nu se va permite accesul mai multor clienți decât numărul maxim prevăzut, pentru a nu afecta comportamentul aplicației. Numărul de utilizatori simultani până la care funcționează sistemul este cu mult peste numarul așteptat de noi.

Sistemul va deveni indisponibil doar pentru clienții care nu au acces la Internet sau la rețeaua locală din care face parte serverul.

În momentul în care aplicația nu va putea fi disponibilă din diverse motive tehnice, utilizatorii vor fi anunțați dinainte de intervalul de indisponibilitate.

### Adaptabilitatea:

[Opțional] Se vor putea adăuga noi comenzi, în funcție de preferințele clienților, fără afectarea celor vechi; nu o să se retesteze răspunsul sistemului la vechile comenzi și astfel clienții nu o să fie afectați de schimbări. Aceștia doar pot profita sau nu de comenzile în plus, conceptul de bază rămânând același.

## Cerințe de securitate

### Securitatea datelor:

Serverul, ca și clientul de asemenea, pot avea în spate doar metodele de securitate specifice sistemului de operare/firewall-ului sau antivirusului folosit. Aflarea IP-ului unui user de către un posibil “răufăcător” nu îl afecteaza cu nimic dacă PC-ul propriu are un sistem de securitate bine pus la punct(antivirus/firewall etc.). Se recomandă bineînțeles nepartajarea de informații personale, fișiere de sistem, etc.

### Securitatea în funcționare:

În cazul căderii Internetului, a unei pene de curent sau a oricărui eveniment de întrerupere a activității desfășurate de utilizator pe sistem, datele ce erau în curs de procesare nu se vor pierde.

## Cerințe de planificare

Perioada de dezvoltare: 1 martie 2013 – 1 mai 2013.