Ice Cloud

Introducere

Datorită creşterii informatiei digitale şi a necesitatii de a stoca informaţii personale uşor accesibile prin intermediul internetui am ales sa implementam un serviciu de cloud, denumit Ice Cloud. Principala functionalitate al acestuia este de a permite userului sa stocheze informaţii personale precum documente, imagini şi fisiere media.
Fiind totodata constienti de existenţa unor servicii similare, am dorit sa implementam o versiune proprie a ceea ce reprezintă un cloud aducând o nuanta proprie şi personală destinată momentan unui public restrans.

Aplicaţia ofera posibilitatea crearii unui cont personal şi securizat. Prin intermediul unui form în care utilizatorul îşi introduce datele personale precum nume, user name, parola, e-mail, întrebare de siguranţa în cazul pierderii parolei curente.
Odată creat contul curent utilizatorul are posibilitatea de a încarca date personale pe care le va putea accesa cu ajutorul accesări contului prin intermediul internetului.

Pentru stocarea informatiilor mentionate am ales o varianta open şi free source prin utilizarea şi configurarea unui server de linux, CentOS. Conectarea la baza de date personale se va face prin instalarea şi configurarea pe server a servicilor oferite de apache(versiunea 2.4.2) şi PHP5.
Comunicarea dinspre partea de interfata utilizator o vom face cu ajutorul limbajului PHP, prin intermediul interogarilor de tip SQL.

Interfata utilizator necesita utilizarea unor tehnologii de dezvoltare WEB precum javascript, AngularJS, CSS3 şi HTML5.
Cu ajutorul HTML se va crea scheletul propriu-zis al site-ului iar infrumusetarea acestuia se va face prin intermediul tehnologiilor oferite de CSS3.
Pentru optimizarea incarcarii paginilor vom folosi serviciile oferite de AngularJS pentru încărcarea numai a modulelor necesare, nemaifiind nevoie de încărcarea intregului continut HTML şi CSS astfel fiind îmbunătăţită comunicarea cu serverul.

Având în vedere ca dorim sa inbunatatim lucrul în echipa şi sa acţionăm prompt folosim o platforma online de gestionare a codului denumita GitHub.
Aceasta permite încărcarea codului urcat de fiecare dintre membrii echipei şi modelarea acestuia în mod dinamic astfel încât toţi sa aibă acces la el. Platforma permite tot odată munca pe ramuri destinate fiecarui contribuitor, astfel evitandu-se posibilitatea de deteriorare a unor versiuni precedente functionale.

Pe maşinile locale am instalat platforma de gestionare a codului GitHub şi am testat buna funcţionare a aceteia şi am creat un schelet pentru partea ce se refera la interacţiunea cu utilizatorul.

În cele ce urmează vom aprofunda cunostintele despre tehnologiile de dezvoltare WEB precum JavaScript, CSS, HTML şi AngularJS.

1. Analiza cerințelor și diagrame Use-Case

A. Analiza cerințelor

1. Problema
   Sistemul va permite unui utilizator să stocheze diferite documente necesare fără utilizarea unui dispozitiv fizic pentru portabilitate. Sistemul oferă totodată accesarea și descărcarea rapidă și ușoară a acestor documente fiind necesară doar o simplă conexiune la internet.

2. Domeniul
   Aplicația ca atatre reprezintă un serviciu de cloud care permite unui utilizator să încarce și să descarce fișiere personale cu ajutorul unui server astfel având acces în timp real la datele sale.

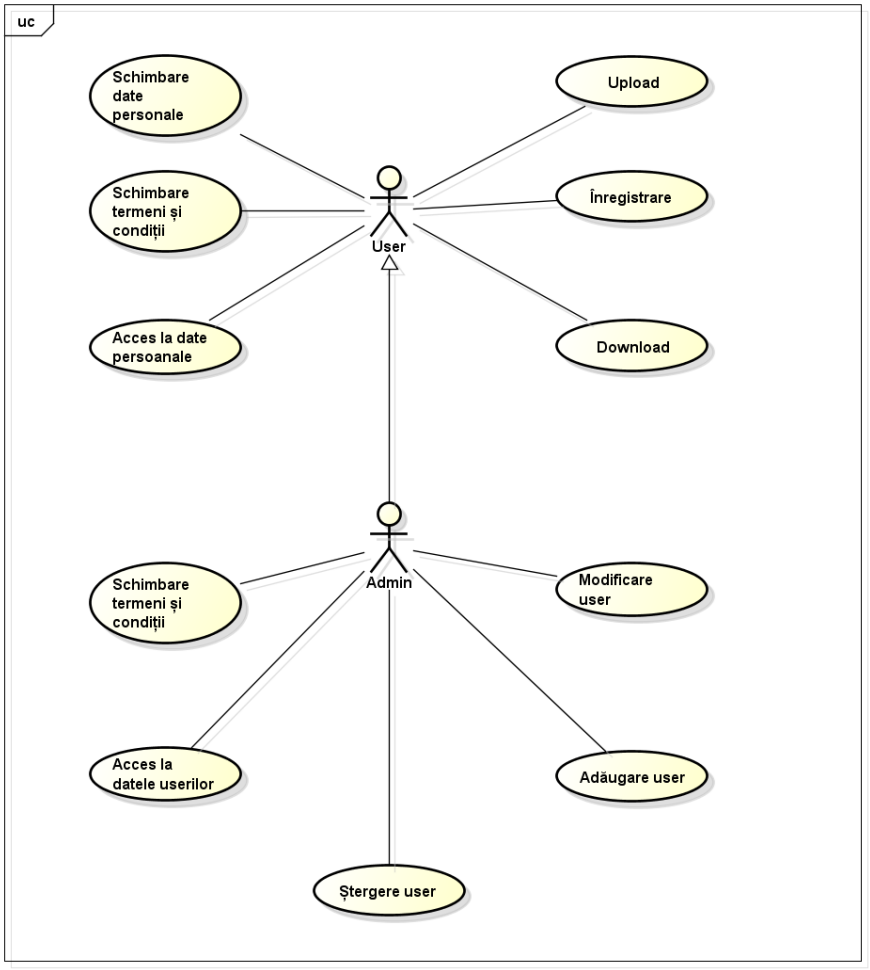
3. Cerințe funcționale și nefuncționale
   Cerințele funcționale sunt reprezentate de servicii furnizate de către utilizator sau de către alte sisteme. Un exemplu de funcționalitate descrisă poate fi chiar încărcarea unui nou fișier în baza de date a sistemului precum și descărcarea acestuia.

   Cerințele nonfuncționale reflectă anumite atibute de calitate precum timpul de răspuns al serverului, anonimitatea datelor și mentenabilitatea, sau diferite aspecte legate de platforma și tehnlogia de calcul.

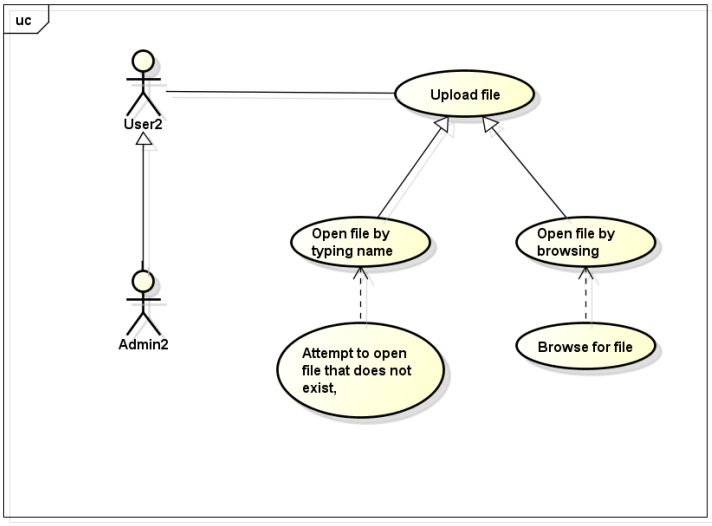
B. Diagrame Use-Case

Cei doi actori ai proiectului nostru sunt: utiliatorul și administratorul. Administratorul are dreptul de a face aceleași operații ca și utilizatorul, dar în plus are acces la datele tuturor utilizatorilor și drept de acces și modificare asupra informațiilor despre aceștia.

Precum se vede în diagramă, atât utilizatorul cât și administratorul pot să se inregistreze, să facă upload, download de fișiere, să creeze cont, să aibă acces la datele personale, dar să le poată și schimba și să vizualizeze termenii și condițiile. În plus, administratorul are dreptul de a schimba termenii și condițiile în caz că apare ceva nou, poate adăuga useri noi sau să șteargă dintre cei existenți sau să modifice date ale acestora. Administratorul mai are dreptul de a vizualiza datele userilor.



Luăm un exemplu de utilizare a apliației noastre: încărcarea unui fișier. Diagrama use-case corespunzătoare este următoarea:



Dacă un user vrea să încarce un fișier, va trebui să il deschidă, scriindu-i nume sau căutându-l printre cele existente.

2. Specificații și Testare

Sisteme de Operare.

Pentru implementarea proiectului ales am ales o abordare open și free software prin utilizarea și configurarea unui server de linux. Așadar sistemul de operare pe care se va desfasura întreaga activitate va fi unul linux based și anume **CentOS**.

Însă având în vedere faptul ca pentru mulți utilizatori de rând sistemele de operare **GNU/Linux** par greu de înțeles am dorit sa oferim și o varianta utilizatorilor de Windows. Așadar în cele ce urmează vom prezenta tool-urile și configurațiile necesare pentru ambele tipuri de sisteme de operare.

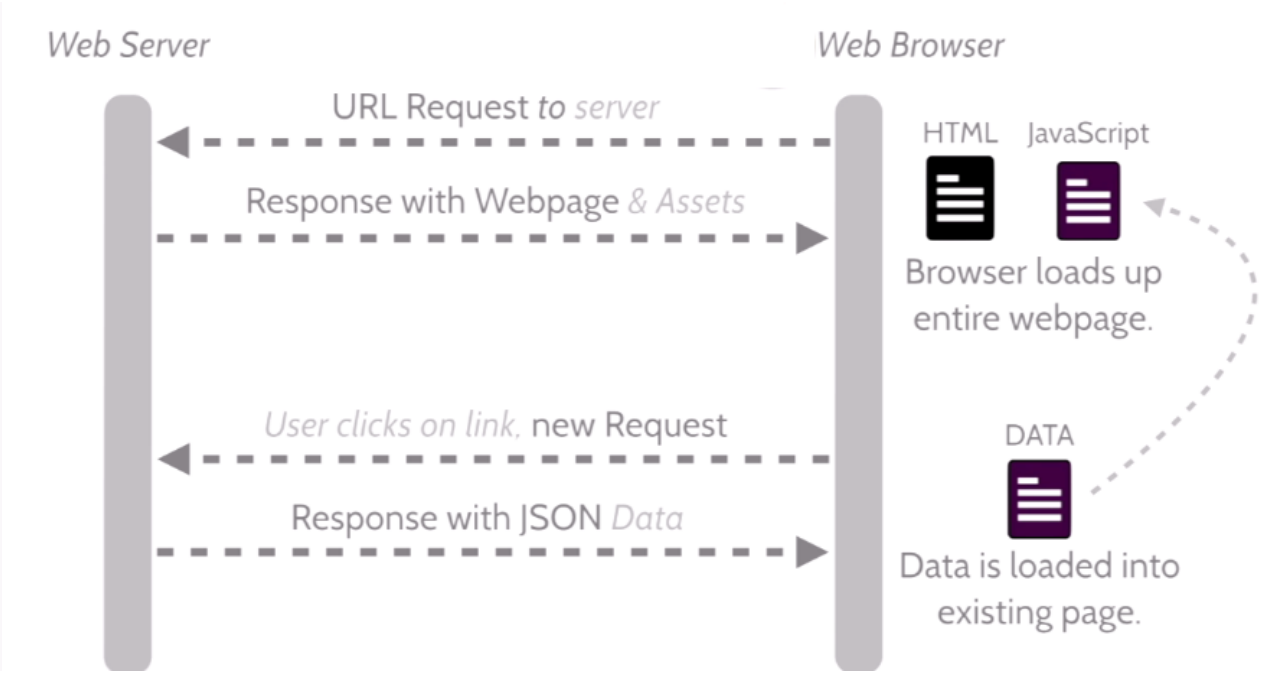
Limbaje de Programare.

Pentru crearea efectivă a platformei de cloud dorite vom avea nevoie în primul rând de o interfață utiliazator cu ajutorul careia se vor efectua diferitele operații oferite precum cea upload și cea download. Pentru aceasta vom utiliza câteva din cele mai importante limbaje de dezvoltare **WEB** și anume **HTML** și **CSS**. Cu ajutorul HTML vom crea practic scheletul interfeței fară ai oferi un aspect neapărat plăcut. Practic vom crea anumite tabele și containere pe care mai apoi le vom manipula. Infrumusețarea interfeței și crearea unui design atrăgător și plăcut se va face prin implementarea tehnologiilor oferite de CSS. Prin poziționarea și manipularea obiectelor și structuri create în HTML și crearea unui design care va răspunde la diferitele device-uri utilizare, desktop și mobile.

Cu toate acestea designul creat este unul static, însă aici intervine utilizarea tehnologiilor oferite de **JavaScript** care permit modelarea și crearea unor sabloane ce permit modificari dinamice în codul creat precedent cu ajutorul HTML și CSS.
În cele din urmă din motive de optimizare vom folosi diferite framework-uri ale JavaScript-ului precum **JQuery** și **AngularJS**.

Am ales utilizarea JQuerry datorită existenței unor funcții existente în librariile acestuia ce permit o manipulare mai ușoară a codului existent iar utilizarea framework-ului oferite de AngularJS pentru încărcarea unor bucăți de cod ce se vom modifica des.

Imbunatațirea adusă utilizării AngularJS se datorează faptului ca permite crearea unor **module** ce permit încărcarea de către server doar a unei parți din pagina afișată utilizatorului și nu a integului conținut, întreaga operație facându-se în mod transparent pentru utilizator, tot odată pastrând codul mult mai mentenabil.



Sursa: http://angularjs.org

Cele menționate mai sus sunt ilustrate în imaginea alaturată în care se arată efectiv comportarea încarcării conținutului la fiecare cerere nouă dată de către utilizator.

Pentru partea de comunicare a continutului paginii cu serverul și cu baza de date avem nevoie de tehnologiile oferite de PHP prin care vom crea un WEB Server și cu ajutorul caruia se vom face diferitele interogari în baza de date. Cu alte cuvinte PHP reprezintă liantul dintre serverul local, baza de date și interfața utilizator oferită prin intermediul paginii create, reprezintă un carauș al diferitelor cereri de la utilizator la baza de date.

Pentru crearea și gestionarea unei baze de date vom folosi tehnologiile oferite de bazele de date relationale, datorită faptului ca aplicația creată repreintă un prototip destinat pentru început unui grup restrans de utilizatori. Diferitele cereri din interfața grafică oferite utilizatorului se traduc ca fiind în spate diferite interogării ale bazei de date pe care le vom manipula cu bazele de date relaționale, **SQL**.

Configurare Hardware.

Windows.

Pentru pregatirea mașinii care rulează pe Windows nu este necesar decât prezenta unui browser de internet precum Mozilla Firefox și înstalarea utilitarului BitnamiWAMP care este un instaler open source ce conține pachete pentru instalarea și dezvoltare aplicatiilor WEB precum Apacche, PHP, phpMyAdmin și mySQL server.

Singura configurare necesara ceruta de instaler fiind cea pentru parola implicita a adminului.

Odată instalat se poate accesa exe-ul aplicației care v-a deschide o pagina în browser și va redirecta spre pagina de logare a adminului. După efectuarea logarii afisandu-se pagina de phpMyAdmin pentru gestionarea bazelor de date.

Linux.

Aici configurarea este puțin mai complicată însă odată efectuată aceasta nu mai este necesară instalarea nici unui utilitar separat.

Având în vedere ca CentOS este un sistem bazat pe pachetele celor de la Red Hat package managerul implicit este **yum.** Cu ajutorul acestuia vom instala pe server modulele necesare configurării serverului.

Vom rula următoarea comandă pentru a instala my-sql server local pe mașina personală.

yum -y install mysql mysql-server

Odată instalat cu succes vom seta ca acesta să fie încărcat în kernel odată cu pornirea sistemului prin comanda următoare.

chkconfig --levels 235 mysqld on
/etc/init.d/mysqld start

Odată cu setearea acestei comenzi vom configura serverul local de mySQL prin rularea în consolă a scriptului de configurare.

mysql_secure_installation

Odată rulat vor apărea un set de întrebări în consolă prin care vom seta anumite flaguri și opțiuni pentru serverul de mySQL.
Prima dată vom seta parola de root apoi vom sterge userii anonimi pentru a nu permite logarea la baza de date numai prin intermediul root, vom dezactiva conectarea de la distanța pentru a spori securitatea, vom rula testul implicit pentru testarea functionarii corecte a bazei de date care vine cu un test presetat și vom reincarca privilegiile tabelelor pentru a face ca orice modificare sa aibe loc imediat.

Acum că am terminat cu instalarea și configurarea serviciului de baze de date ne vom ocupa de serviciul oferit de către Apache. Vom instala ca și mai sus serviciul de apache necesar prin comanda.

yum -y install httpd

Iar mai apoi îl vom încarca și pe acesta odată cu pornirea sistemului prin comanda.

chkconfig --levels 235 httpd on

Odată setat vom porni servicul de apache prin următoare line de comandă.

/etc/init.d/httpd start

Pentru testarea functionării acestuia vom tasta în Browser adresa locală generata de server. Aceasta difera de la mașină la mașină.

Ex: http://192.168.0.100

Documentul implicit root Apache este **/var/www/html** pe CentOS și fișierul de configurare este **/etc/httpd/conf/httpd.conf**. Configurațiile suplimentare sunt stocate în directorul **/etc/httpd/conf.d/**.

Următorul pas necesar este instalarea serviciilor oferite de php. Vom instala php iar după aceea vom restarta serviciul de apache cu noile configurări făcute prin urmatoarele comenzi.

yum -y install php
/etc/init.d/httpd restart

Odată săvârșită instalarea vom testa funcșionarea corecta a acestor servicii cât și a conexiunii dintre php și serverul de apache.

Documentul rădăcină al site-ului este **/var/www /html**. Vom crea acum un fisier PHP mic (info.php), în acel director. Fișierul va afișa o mulțime de detalii utile despre instalarea nostra PHP, cum ar fi versiunea instalată PHP.

Vom crea fișierul dorit prin utilizare editorului implicit în CentOs și anume vi.

vi /var/www/html/info.php

Fișierul creat va trebui să conțină urmatoarele linii de cod.

```
<?php
phpinfo();
?>
```

Pentru a vedea corectitudinea si funcționarea vom accesa acum din browser output-ul fisierului accesat de pe server.

http://192.168.0.100/info.php

Pentru a obține sprijin MySQL în PHP, putem instala pachetul php mysql.

yum -y install php-mysql

În etapa următoare, voi instala unele module comune PHP care sunt cerute de sisteme CMS precum Wordpress, Joomla și Drupal.

yum -y install php-gd php-imap php-ldap php-odbc php-pear php-xml php-xmlrpc php-mbstring php-mcrypt php-mssql php-snmp php-soap php-tidy curl curl-devel

APC este un PHP opcode cacher open și free source pentru caching și pentru optimizarea de cod intermediar PHP. Este similar cu alte cachers PHP opcode, cum ar fi eAccelerator și Xcache.

yum -y install php-pecl-apc

Acum este necesara restartarea serviciului de apache pentru încărcarea noilor configurații.

/etc/init.d/httpd restart

Putem observa ca la încărcarea în browser a fisierului php **http://192.168.0.100/info.php** acesta oferă mult mai multe informații necesare ulterior.

Pentru a ușura munca cu bazele de date vom instala **phpMyAdmin** care este o interfață web prin care se pot gestiona bazele de date MySQL.

În primul rând vom permite repositoriului RPMforge pe sistemul nostru CentOS deoarece phpMyAdmin nu este disponibil în arhivele oficiale CentOS.

rpm --import http://dag.wieers.com/rpm/packages/RPM-GPG-KEY.dag.txt

Pe sistemele x86_64:

yum -y instala http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el6.rf.x86_64.rpm

Pe sistemele i386:

yum -y instala http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el6.rf.i686.rpm

phpMyAdmin poate fi instalat acum, după cum urmează cu următoarea comandă.

yum -y install phpMyAdmin

Acum vom configura phpMyAdmin și vom schimba configurația Apache, astfel încât phpMyAdmin să permită conexiuni nu numai de la localhost.

vi /etc/httpd/conf.d/phpmyadmin.conf

```
#
#  Web application to manage MySQL
#

#<Directory "/usr/share/phpmyadmin">
#  Order Deny,Allow
#  Deny from all
#  Allow from 127.0.0.1
#</Directory>

Alias /phpmyadmin /usr/share/phpmyadmin
Alias /phpMyAdmin /usr/share/phpmyadmin
Alias /mysqladmin /usr/share/phpmyadmin
```

Și vom schimba autentificara phpmyadmin prin intermediul unui cookie către http.

vi /usr/share/phpmyadmin/config.inc.php

```
[...]
/* Authentication type */
$cfg['Servers'][$i]['auth_type'] = 'http';
[...]
```

Odată cu restartarea serviciului de apache si reincarcarea noilor setari făcute **/etc/init.d/httpd restart** putem face testul pentru verificarea functionării corecte a serviciilor phpmyadmin prin accesarea.

http://192.168.0.100/phpmyadmin/

Astfel încheindu-se întreaga instalare si configurare a pachetelor necesare.

3. Fundamente teoretice

Cloud computing

În ultima perioadă conexiunea la internet a utilizatorilor a devenit foarte răspândită, astfel încât aproate toate resursele disponibile se pot plasa și partaja pe internet, uneori chiar intre utilizatori complet independenți unii de alții: software (programele) și datele/informațiile sunt aduse din Internet pe calculatorul utilizatorului la cerere (on demand), ca și cum ar fi vorba de servicii publice banale precum apa sau energia electrică.

Ce înseamnă **cloud computing**? Este un concept relativ nou (2006-2007) din domeniul computerelor și informaticii și se referă la un ansamblu distribuit de servicii de calcul, aplicații, acces la informații și stocare de date, fără ca utilizatorul să aibă nevoie să cunoască configurația fizică a sistemelor care furnizează aceste servicii. În limba română încă nu există o traducere a acestui concept. Prin cloud computing se pot accesa la cerere resurse harware si software, prin intermediul internetului, iar aceste cereri se vor livra sub formă de servicii. De exemplu, o companie își poate pune la dispoziția altor companii anumite resurse precum infrastructură, servere, aplicații sub formă de servicii.

Există trei tipuri de servicii: **SaaS, adică Software as a Service**, prin care se închiriază licențe software pe bază de abonament; pentru acestea clienții nu trebuie să își facă probleme legate de instalarea, setarea și rularea aplicației deoarece distribuitorul se va ocupa de acest lucru. Exemple pentru acest tip sunt: aplicațiile de email: Gmail, Yahoo mail, sau cele de socializare, cum ar fi Facebook, Google+.Al doilea tip este **IaaS-Infrastructure as a Service**, care reprezintă închirierea de resurse hardware cum ar fi memorie și putere de calcul. Celălalt tip**, PAAS (Platform as a Service)** este unul dintre cele mai răspândite printre dezvoltatori deoarece clientul crează o aplicație sau un serviciu folosindu-se de librării sau instrumente primite de la furnizor. Totodată clientul controlează implementare software și setările de configurare. Furnizorul oferă servere, rețeaua și alte servicii de care are nevoie clientul.

Cloud computing reprezintă o nouă schimbare de paradigmă, poate la fel de importantă ca trecerea de la mainframes la conceptul de client-server (1980), deoarece astfel executarea aplicațiilor se va face online și nu pe stația de lucru proprie. Dacă interfața pusă la dispoziție de furnizor este una de bună calitate, atunci utilizatorul nu trebuie neaparat să fie expert în tehnologie și infrastructură, ceea ce, consider eu, este de o mare importanță.

Furnizorii tipici de **cloud computing** pun la dispoziție, de exemplu, aplicații comerciale standard iar utilizatorul are acces la acestea doar prin intermediul unui browser local, deoarece atât aplicația cât și datele proprii ale utilizatorului sunt găzduite în cloud, pe serverul furnizorului de servicii. Acestea fiind spuse, înseamnă că asigurarea confidențialității și a drepturilor de acces la date are un rol foarte important.

Folosirea serviciilor de cloud computing devine din ce in ce mai populara, in ultimii ani făcându-și apariția în domeniul afacerilor sintagma „If you are not in the cloud you are not going to be in business”.

Server

În realizarea proiectului nostru o parte importantă este serverul, iar în tehnologia informației acesta poate fi definit ca un program de aplicație care furnizează servicii altor aplicații (aplicații client) care se află pe același calculator sau pe calculatoare sau dispozive diferite. Aplicația de server așteaptă și conexiuni din partea aplicațiilor client. Mai putem să numim server calculatorul pe care rulează una sau mai multe asemenea aplicații.

Diferența majoră între computerele personale și servere nu este partea hardware ci partea de software. Pe servere rulează sisteme de operare care sunt special proiectate pentru acestea. De asemenea ele rulează aplicații special proiectate pentru procesele dorite.

Sistemul de operare Microsoft Windows este predominant în rândul computerelor personale, dar în lumea serverelor cele mai populare sistem de operare sunt FreeBSD, SunSolaris și GNU/Linux – care derivă și sunt asemănătoare cu sistemul de operare UNIX. UNIX a fost proiectat inițial pentru microcomputere și pentru servere, care au înlocuit treptat microcomputerele. UNIX a fost o alegere logică și eficientă ca sistem de operare pentru servere.

Linux

Linux este un sistem de operare care a fost creat de Linus Torvalds, la Universitatea din Helsinki, în Finlanda. Kernelul creat de Linus este inima tuturor sistemelor Linux, fiind dezvoltat şi distribuit sub **GNU** General Public License, iar codul său sursă este disponibil gratuit pentru oricine.

Acest kernel poate fi considerat forma de bază în jurul căreia un sistem de operare Linux este dezvoltat. Există sute de companii si organizatii care au lansat propriile lor versiuni de sisteme de operare bazate pe nucleul Linux. În afară de faptul că este distribuit gratuit, funcţionalitatea Linux, adaptabilitatea şi robusteţea, au făcut din Linux principala alternativa la sistemele de operare proprietare, cum ar fi, de exemplu, sistemul de operare al Microsoft, Windows.
In al doilea deceniu de existenţă, Linux a fost adoptat la nivel mondial în primul rând ca o platforma de server. Utilizarea sa ca si sistem de operare pentru calculatoarele persoanele este, de asemenea în creştere.

CentOS este o distributie de Linux bazată pe sursele puse la dispoziție sub licență liberă ale Red Hat Enterprise Linux, care prezintă compatibilitate pe ambele tipuri de arhitecturi x86-64 și x86. Funcționalitatea de bază a acestei platforme este destinată servicilor de tip free-computing și anume servere și mainframe-uri. Distribuția prezintă un nucleu monolitic, o interfață utilizator implicită **GNOME** sau **KDE** iar ca și metodă de gestionare a pachetelor se folosește package manager-ul implicit distribuțiilor de tip Red Hat și anume **YUM**.

Windows

Microsoft Windows sau **Windows** este o meta familie de sisteme de operare grafice dezvoltate, comercializate, și vândute de Microsoft. Se compune din mai multe familii de sisteme de operare, fiecare dintre care răspunde la un anumit sector al industriei de calcul precum WindowsNT, Windows Phone sau Windows Server.

Windows 8 este un sistem de operare ce face parte din familia Windows NT de sisteme de operare.Windows 8 a introdus schimbări majore la platforma și interfața cu utilizatorul a sistemului pentru a îmbunătăți experiența de utilizare pe tablete prin introducerea designului **Metro** și a noului tip de boot securizat **UEFI**.

HTML

HTML sau HyperText Markup Language este limbajul de marcare standard utilizat pentru a crea pagini Web.

HTML este scris sub forma de elemente HTML formate din tag-uri închise în paranteze unghiulare (cum ar fi <html>).

Un browser Web poate citi fișiere HTML și le poate compune în pagini Web vizibile sau sonore. Browserul nu afișează tag-uri HTML, dar le folosește pentru a interpreta conținutul paginii. HTML descrie structura unui site web semantic, împreună cu indicii de prezentare, ceea ce face un limbaj de marcare, mai degrabă decât un limbaj de programare.

Elemente HTML formează blocurile de baza la toate site-urile. HTML permite imagini și obiecte care trebuie încorporate și pot fi utilizate pentru a crea formulare interactive. Acesta oferă un mijloc de a crea documente structurate care denotă semantică structurala pentru text, cum ar fi titluri, paragrafe, liste, link-uri, citate și alte elemente. HTML poate încorpora și scripturi scrise în diverse limbaje, cum ar fi JavaScript sau CSS.

CSS

Cascading Style Sheets este folosit pentru a descrie aspectul și formatarea unui document scris într-un limbaj de marcare. Este cel mai des folosit pentru a schimba stilul de pagini web și interfete scrise în HTML și XHTML, limbajul poate fi aplicat la orice tip de document XML, inclusiv XML simplu, SVG și XUL. Împreună cu HTML și JavaScript, CSS este o tehnologie piatră de temelie folosită de cele mai multe site-uri web pentru a crea pagini web aranjate vizual într-un mod plăcut, interfețe de utilizator pentru aplicații web, și interfețe de utilizator pentru multe aplicații mobile.

CSS este destinat în primul rând pentru a permite separarea conținutului documentului de prezentare de documente, inclusiv elemente cum ar fi aspectul, culorile, și fonturile. Această separare poate îmbunătăți accesibilitatea conținutului, poate oferi mai multă flexibilitate și control în specificarea caracteristicilor de prezentare și permite ca mai multe pagini HTML să aibă acces la formatare prin specificarea unui fișier **.css** separat.

JavaScript

JavaScript este un limbaj dinamic de programare.Este cel mai frecvent utilizat în browserele web deoarece permite scripturi client-side pentru a interacționa cu utilizatorul și comunica asincron. Este de asemenea folosit în rețea la programarea de tip de server cu cadre, cum ar fi Node.js, dezvoltarea de jocuri și crearea aplicații mobile și desktop.

JavaScript este clasificat ca un prototip bazat pe limbaj de scripting cu funcții de dactilografiere dinamic și de primă clasă. Acest amestec de caracteristici îl face un limbaj multi-paradigmă,ce suporta

diferite stiluri de programare precum cea orientata-obiect, imperativa și funcționala.

Sintaxa JavaScript este, de fapt derivat din C și permite, de asemenea, sa fie utilizat în medii care nu sunt bazate pe web, cum ar fi documentele PDF și widget-uri desktop.

Jquery

jQuery este cea mai răspândită biblioteca JavaScript concepută pentru a simplifica comunicarea client-side în limbajul de scripting.

Sintaxa jQuery este proiectata pentru a face mai ușoară navigarea un document și selectarea elementelor din DOM. jQuery prevede, de asemenea, capabilități pentru dezvoltatori cu scopul de a crea diverse plugin-uri pentru biblioteca de JavaScript. Acest lucru le permite dezvoltatorilor să creeze totodata efecte avansate de animație de nivel înalt.

AngularJs

AngularJS, denumit în mod obișnuit Angular este un framework open-source menținut de Google și o comunitate de dezvoltatori individuali și de anumite corporații ce își propune să abordeze multe dintre provocările cu care se confruntă în dezvoltarea de aplicații cu o singură pagină(single page applications). Scopul său este de a simplifica atât dezvoltarea cât și testarea unor astfel de cereri prin furnizarea unui cadru pentru client-side cu un model view-controller (MVC).

Acesta functionează prin interpretarea codului de HTML pe care mai apoi îl captează în așa numitele modele javascript cu care se va face modelarea ulterioara.

SQL

SQL (**Structured Query Language** - Limbaj Structurat de Interogare) este un limbaj de programare specific pentru manipularea datelor în sistemele de manipulare a bazelor de date relaționale (RDBMS), iar la origine este un limbaj bazat pe algebra relațională. Acesta are ca scop inserarea datelor, interogații, actualizare și ștergere, modificarea și crearea schemelor, precum și controlul accesului la date. A devenit un standard în domeniu, fiind cel mai popular limbaj utilizat pentru creearea, modificarea, regăsirea și manipularea datelor de către SGBD-urile (Sistemele de Gestiune a Bazelor de Date) relaționale.

SQL permite atât accesul la conținutul bazelor de date, cât și la structura acestora.

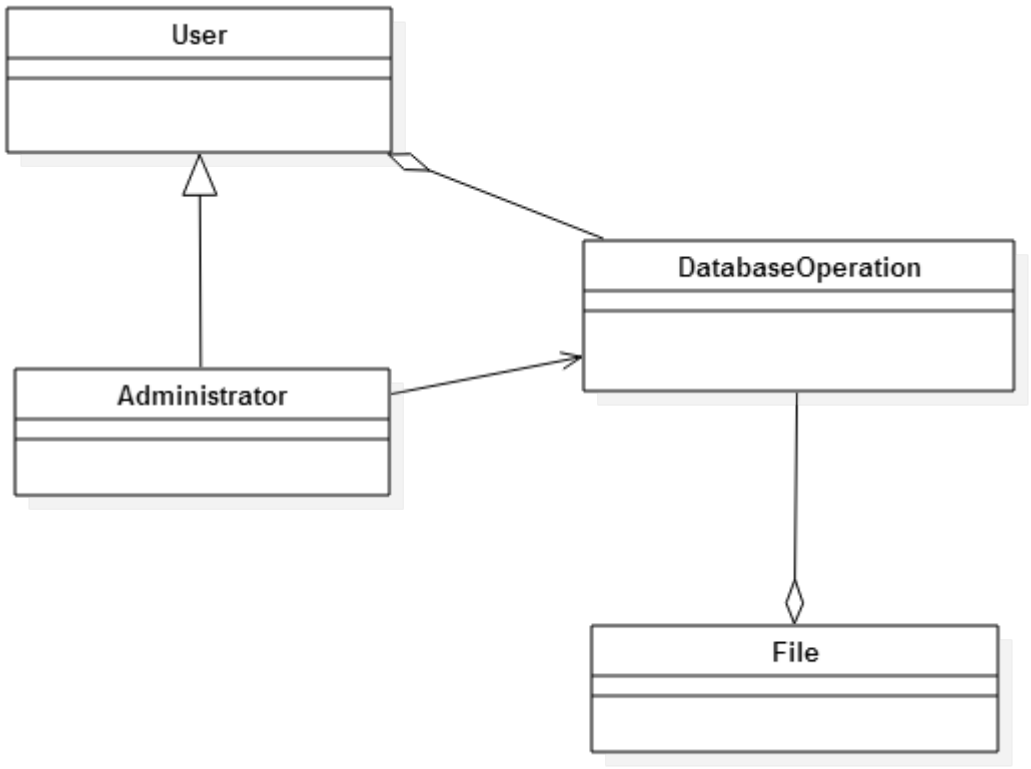
O bază de date reprezintă o modalita de a stoca informații și date pe un suport extern cu posibilitatea extinderii ușoare și a regăsirii rapide a acestora. Baza de date permite stocarea unor număr masiv de elemente, fiecare putând consta din cantități mari date ce pot fi accesate simultan prin internet de către mii de utilizatori. Baza de date asigură, de asemenea, disponibilitatea aplicației și a datelor permanentă. Bazele de date au un impact decisiv asupra modului de organizare și funcționare a unei aplicații.

Principale avantaje ale unei baze de date sunt controlul centralizat al datelor, viteza mare de actualizare și regăsire a informațiilor, sunt compacte și flexibile și redundanța scăzută a datelor memorate care se obține prin partajarea datelor între mai mulți utlizatori și aplicații, menținerea integrității datelor prin politica de securizare. O bază de date va urmări următoarele concepte: securitatea datelor, caracterul secret al datelor, confidențialitatea și integritatea datelor.

Pentru administrarea bazei de date se va folosi un sistem de gestiune al bazelor de date relațional, în care datele sunt memorate în tabele.

Dintre tipurile de date permise de baze de date, cel mai utilizat în cadrul proiectul este tipul de date LONGBLOB. LONGBLOB este un tip de date binare mari și este folosit pentru a stoca obiecte binare de mari dimensiuni (imagini, secvente audio sau video) sau pentru a reține texte de dimensiuni mai mari de 255 de caractere (dimensiunea maximă a tipului de date TEXT). Dimensiunea maximă a unui date LONGBLOB va fi de 4TB.

4. Diagrama de clase



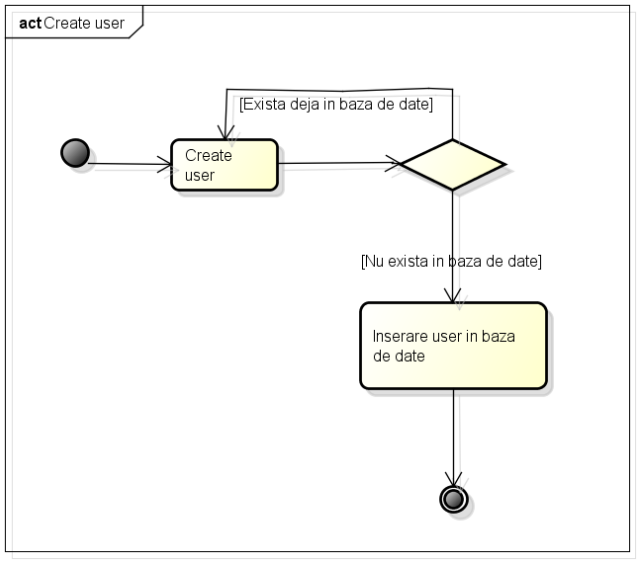
Cele patru obiecte necesare implementării serviciului de cloud vor fi userii, administratorii, baza de date și fișierele stocate de către useri. Pentru aceasta avem patru clase: clasa User ce va reprezenta un utilizator, clasa Admnistrator ce va moștenii de la user toate atributele și acțiunile unui user normal, dar va avea anumite responsabilități suplimentare, clasa DatabaseOperation ce va face conexiunea către baza de date și va face posibile toate operațiile necesare lucrului cu o bază de date precum inserări, interogări, actualizări sau ștergeri din baza de date și clasa File ce va modela un fișier.

Întrucât informațiile despre useri și fișiere vor fi reținute în baza de date, clasele User și File sunt în relație de agregare cu clasa DatabaseOperation. Administratorul va fi cel care va putea adăuga sau șterge useri, acesta se va afla în relație de asociere cu clasa DatabaseOperation.

5. Diagrama de interacțiune: secvență, colaborare, activitate

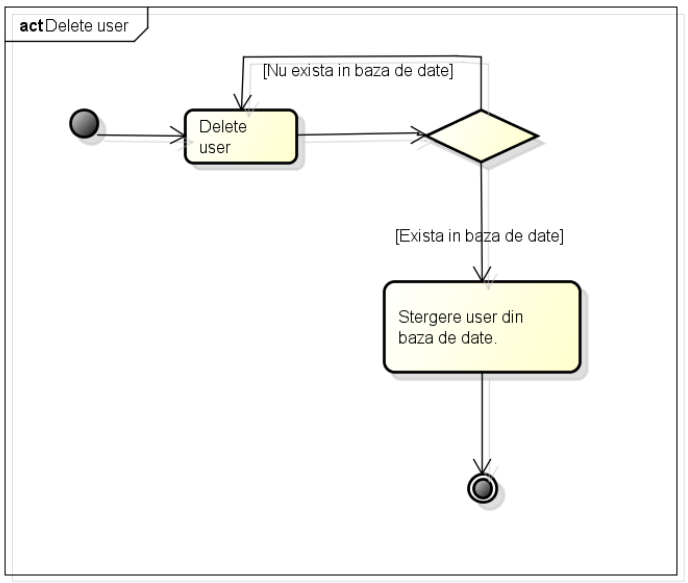
A. Diagrama de activitate

În cazul creării unui utilizator nou avem următoarea diagramă:

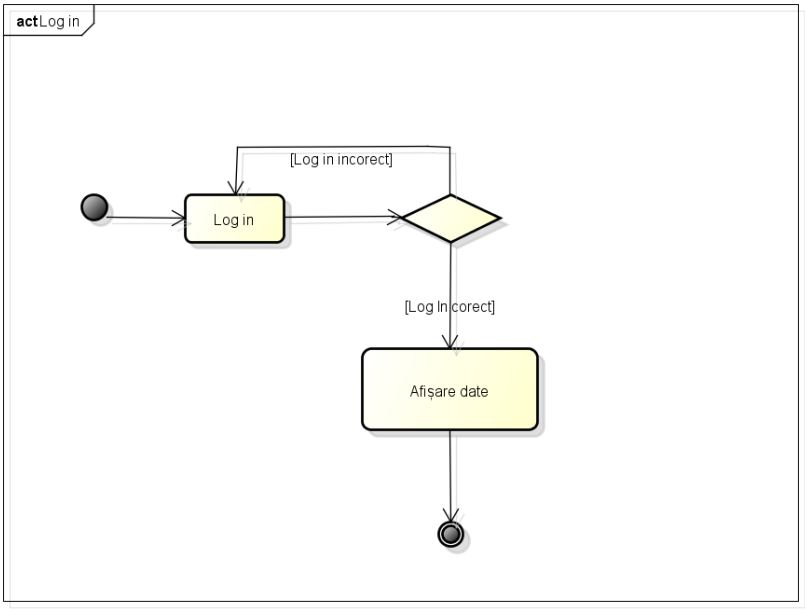


Se introduc datele noului user, iar apoi se verifică dacă acesta există în baza de date. În caz afirmativ, se revine în starea dinainte, iar în caz contrar se inserează datele utilizatorului nou în baza de date și astfel e finalizată operația.

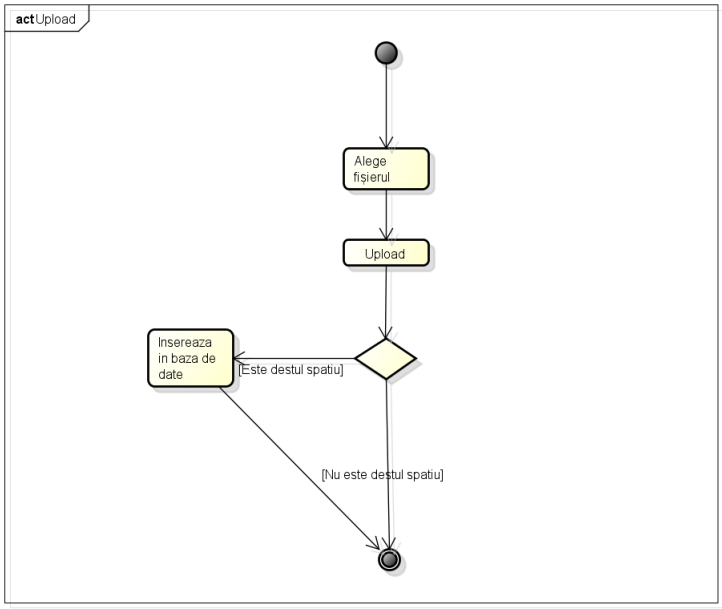
Asemănătoare este și operația de ștergere a unui user. Se introduce numele userului și se verifică existența acestuia în baza de date. Dacă există, acesta va fi șters, altfel se revine în starea inițială.



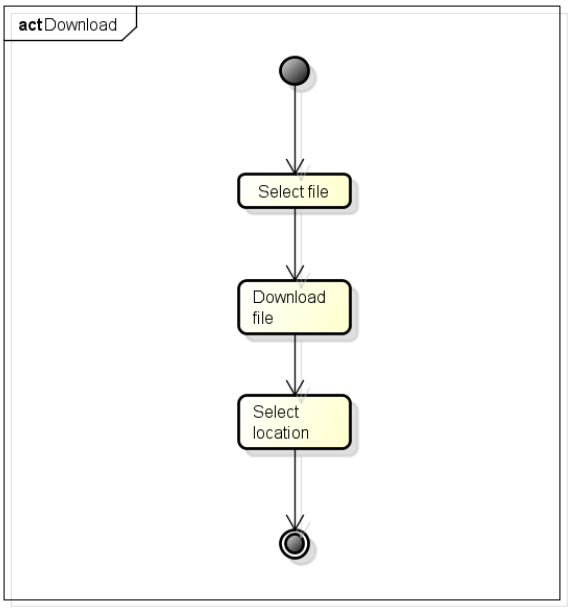
Pentru Log In, utilizatorul va trebui să își introducă numele pe care și l-a ales sau adresa de email și parola. Dacă aceste date sunt valide, atunci se va putea autentifica și i se vor afișa datele, altfel el va trebui să reîncerce să introducă date valide.



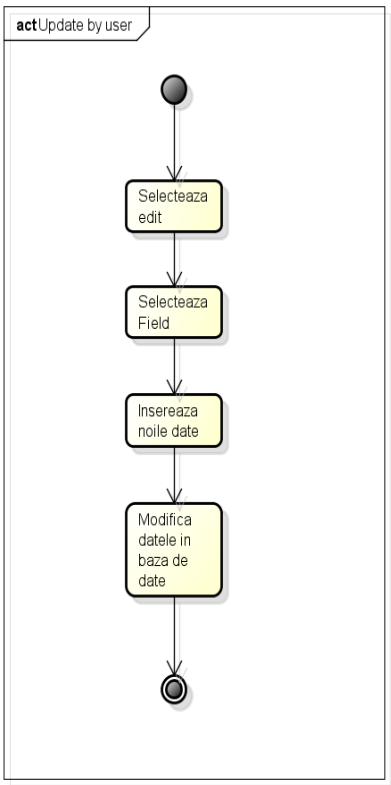
Pentru ca utilizatorul să încarce un fișier, el trebuie să aleagă fișierul dorit, să dea click pe butonul de upload, iar apoi se va verifica dacă există destul spațiu pentru ca fișierul să poată fi încărcat. Dacă este spațiu destul, fișierul va fi înregistrat în baza de date și se va finaliza operația. În caz contrat, operația se finalizează, iar utilizatorul va fi anunțat că fișierul nu a putut fi încărcat din cauza lipsei de spațiu.

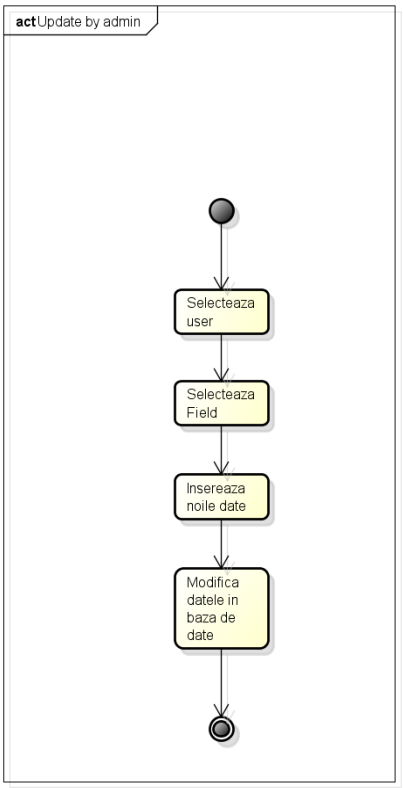


Dacă utilizatorul vrea să descarce un fișier, el trebuie doar să îl selecteze, să dea click pe butonul de download, să aleagă locul unde va fi salvat pe dispozitivul său, iar fișierul va fi descărcat.



În cazul în care administratorul vrea să modifice informații ale unui user, el va trebui să selecteze userul dorit, apoi câmpul pe care dorește să îl modifice, să introducă noile date, care la rândul lor vor fi modificare în baza de date. Astfel, operația este finalizată. Pentru ca un user să își modifice datele personale, operația este asemănătoare cu cea anterioară, diferența fiind faptul că el trebuie să apese buntonul de edit specific lui.



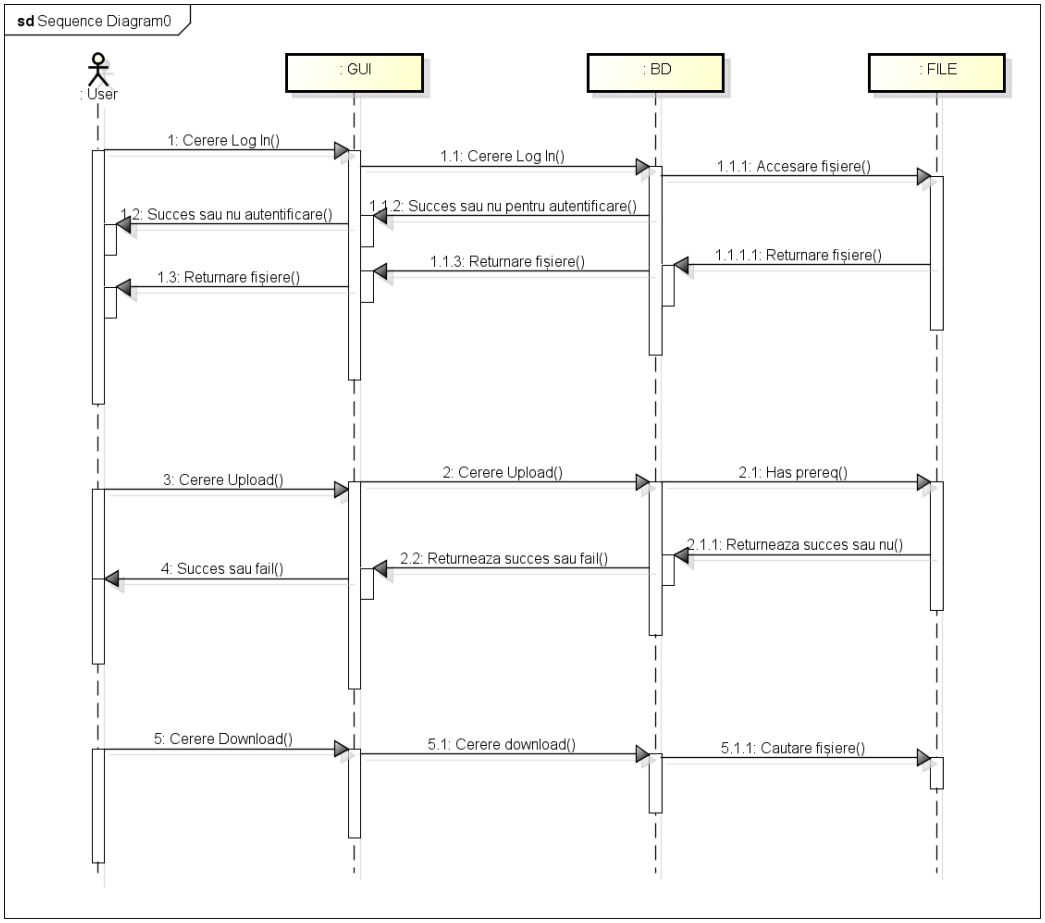


B. Diagrama de secvență

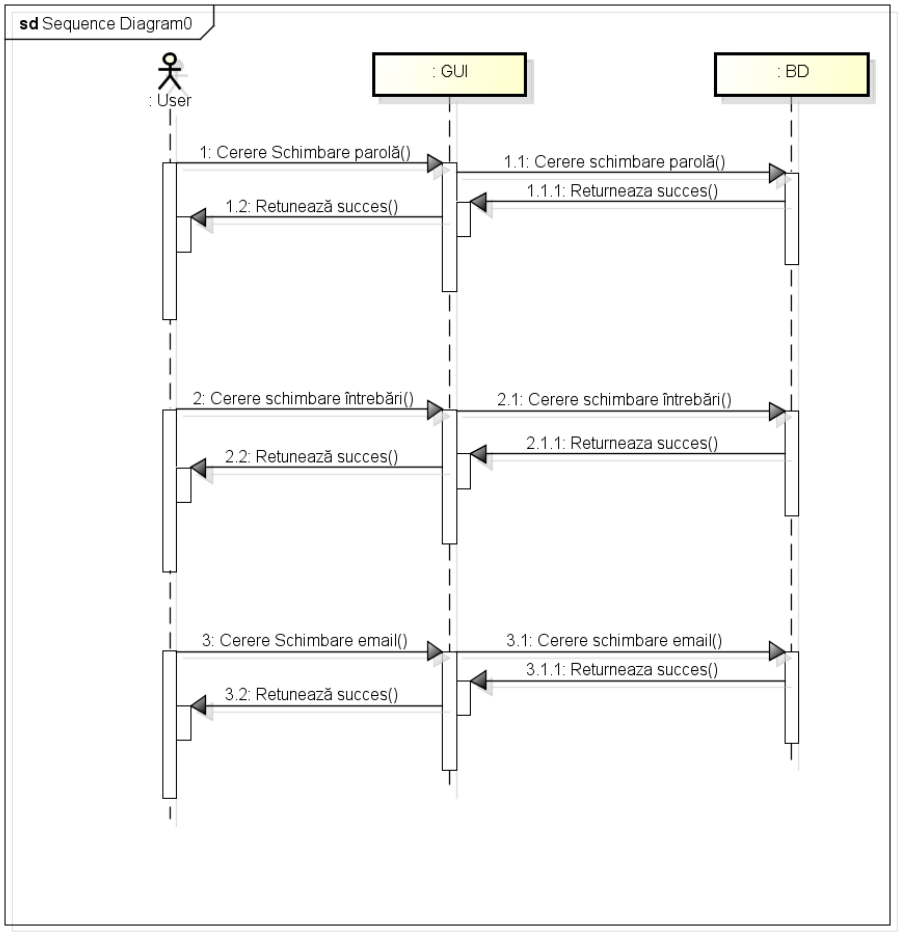
Dacă un utilizator va vrea să facă log in, el va face o cerere de log in partea de interfață, cerere care va ajunge la baza de date. De acolo va primi un răspuns dacă înregistrarea s-a făcut sau nu cu succes, iar în caz de succes se vor putea accesa fișierele.

În caz că utilizatorul va vrea să încarce un fișier, se va trimite un mesaj bazei de date, iar dacă operația s-a făcut cu succes, adică dacă dimensiunea disponibilă este mai mare decât dimensiunea fișierului, sau nu, userul va primi un mesaj.

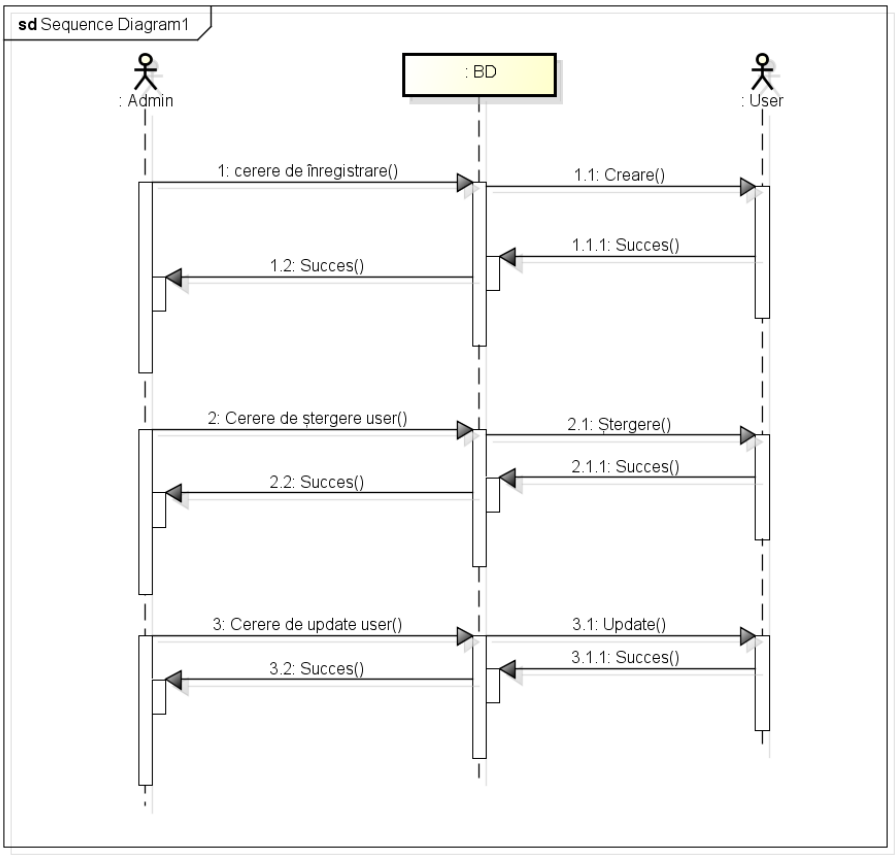
Pentru download, prin intermediul interfeței cu utilizatorul, acesta trimite o cerere către baza de date, iar dacă fișierul există, el va fi returnat utilizatorului.



În caz că utilizatorul vrea să își modifice parola sau întrebările secrete și implicit răspunsurile sau adresa de email sau orice altă informație despre el, prin intermediul interfeței se trimite către baza de date un mesaj, iar după efectuarea operației se primește un mesaj de succes.



Legătura dintre administrator și user se face prin baza de date. Administratorul poate să înregistreze utilizatori noi, să șteargă utilizatori existenți sau să modifice informații despre anumiți utilizatori. La finalizarea fiecărei operații menționate anterior, el va primi un mesaj cum că aceasta a fost realizată cu succces.

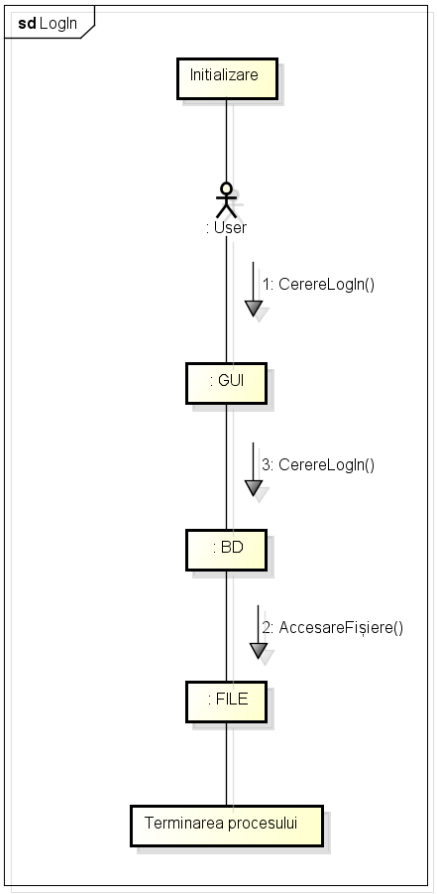


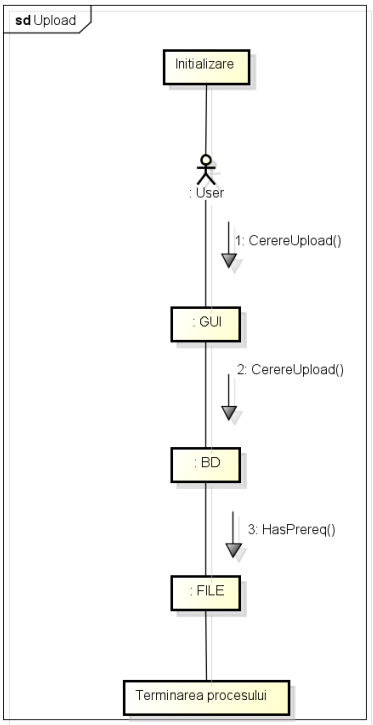
C. Diagrama de colaborare

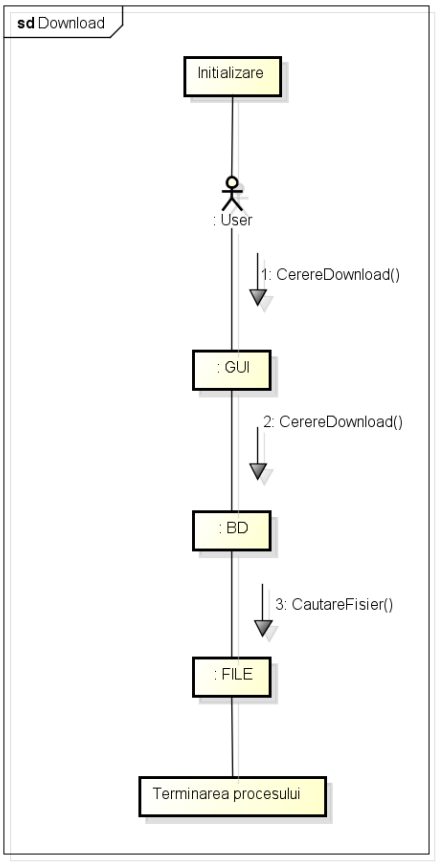
Diagrama de colaborare este derivată din cea de secvență. Explicațiile diagramelor au fost date deja în subpunctul anterior.

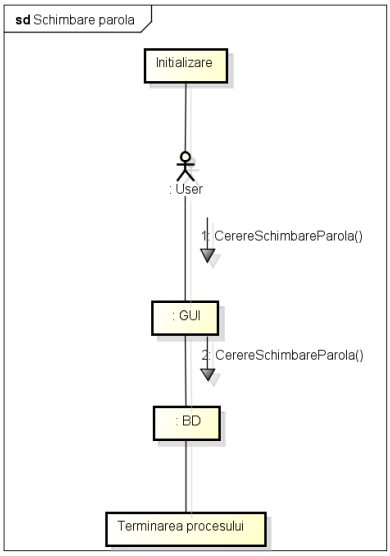
În cazul operațiilor pentru schimbare parolă, schimbare intrebări secrete și a emailului, diagramele sunt la fel, doar mesajul diferă. De aceea, am ales să dau numai un exemplu, pentru a nu încărca documentația cu același tip de diagrame.

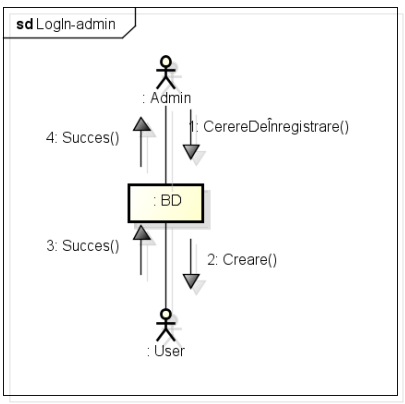
La fel și in cazul operațiilor de înregistrare utilizator de către administrator, ștergere utilizator de către administrator și modificarea informațiilor unui utilizator de către administrator. Un exemplu de e deajuns pentru a arăta cominicarea.


sd LogIn
Initializare
: User
1: CereregLogIn()
: GUI
3: CereregLogIn()
: BD
2: AccesareFișiere()
: FILE
Terminarea procesului


sd Upload
Initializare
: User
1: CereregUpload()
: GUI
2: CereregUpload()
: BD
3: HasPrereq()
: FILE
Terminarea procesului

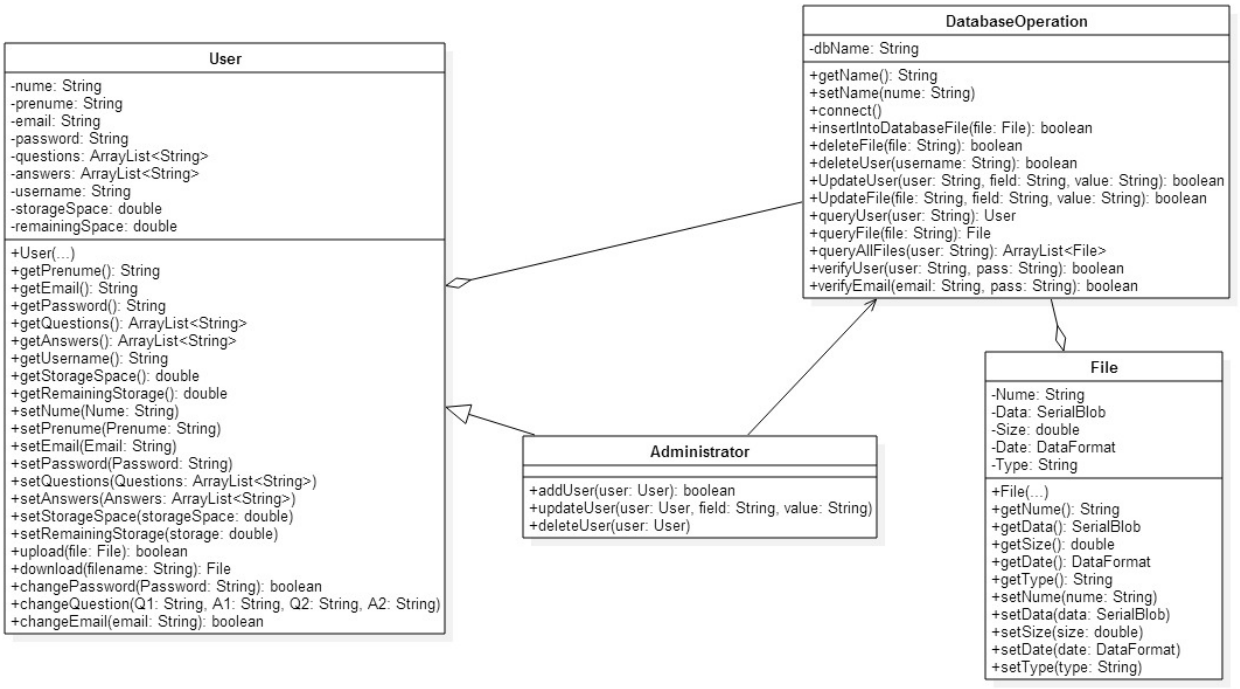

sd Download
Initializare
: User
1: CereregDownload()
: GUI
2: CereregDownload()
: BD
3: CautareFisier()
: FILE
Terminarea procesului


sd Schimbare parola
Initializare
: User
1: CereregSchimbareParola()
: GUI
2: CereregSchimbareParola()
: BD
Terminarea procesului


sd LogIn-admin
: Admin
4: Succes()
1: CereregDeînregistrare()
: BD
3: Succes()
2: Creare()
: User

6. Detalii de Implementare

1. Diagrama de clase – detaliere



Fiecare clasă va modela un obiect și avea anumite atribute și acțiuni specifice. Clasa user va conține toate informațiile ce trebuie reținute despre un utilizator precum nume, prenume, email, password, întrebări pentru recuperarea parolei și răspunsurile corespunzătoare, username (care va fi unic în baza de date), storageSpace ce va reține spațiul total disponil utilizatorului și remainingSpace ce va reține spațiul ce ia rămas utilizatorului. Toate atributele vor fi de tip private și vor avea metode de set și get corespunzătoare. Clasa va avea și un constructor pentru crearea de noi useri. Printre acțiunile pe care le va putea face userul se numără upload – utilizatorul va putea salva un fișier în spațiul său personal, download – utilizatorul va putea descărca un fișier dintre cele stocate, changePassword – schimbarea parolei, changeQuestion – schimbarea uneia dintre întrebările pentru recuperarea parolei și changeEmail – schimbarea adresei de email aferente contului.

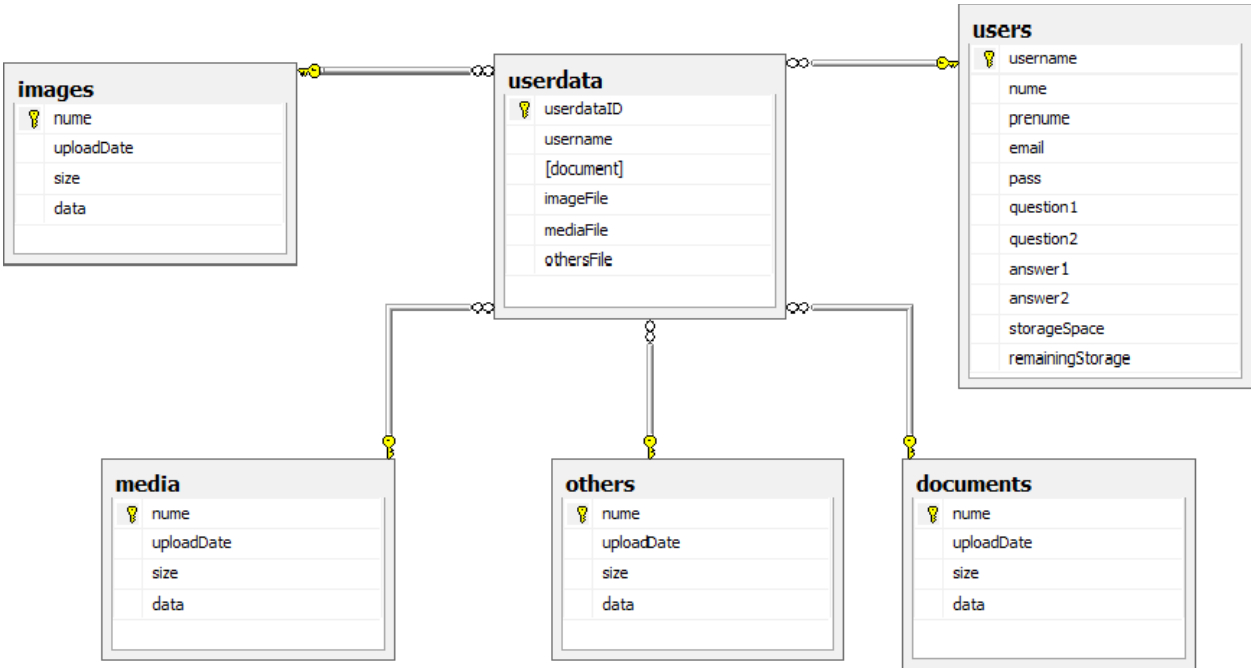
Clasa administrator va moșteni toate atributele și metodele clasei User și va avea în plus alte 3 clase cu care va putea administra userii din sistem: addUser – adăugarea unui nou utilizator în baza de date, updateUser – actualizarea informațiilor despre un anumit user și deleteUser – ștergerea unui user din sistem.

Clasa DatabaseOperation va avea ca atribut dbName – numele bazei de date și va fi responsabilă cu crearea unei conexiuni cu baza de date prin metoda connect. Ea va face posibile toate operațile necesare lucrului cu baza de date: inserare, prin metodele insertIntoDatabaseFile și insertIntoDatabaseUser, ștergere, prin metodele deleteFile și deleteUser, actualizare, prin metodele UpdateUser și UpdateFile și interogare, prin metodele queryFile, queryAllFiles și queryUser. Întrucât

numele va fi privat, clasa va avea nevoie de o metodă get și una set pentru a avea acces la acesta. Clasa va mai avea responsabilitatea de a verifica existența unui user în sistem la logare prin cele două metode de verificare: verifyUser și verifyEmail.

Clasa File va modela fișierele stocate în baza de date. Informațiile ce vor fi reținute despre fișiere sub forma de atribute private sunt: nume, data (fișierul sub formă binară), size (dimensiunea fișierului – reținută pentru calcularea spațiului de stocare disponibil), date (data la care fișierul a fost adăugat în sistem) și type (tipul fișierului – document, imagine sau fișier media). Întrucât atributele vor fi private, clasa va avea metode corespunzătoare de get și set pentru fiecare atribut, precum și un constructor pentru crearea unui nou fișier.

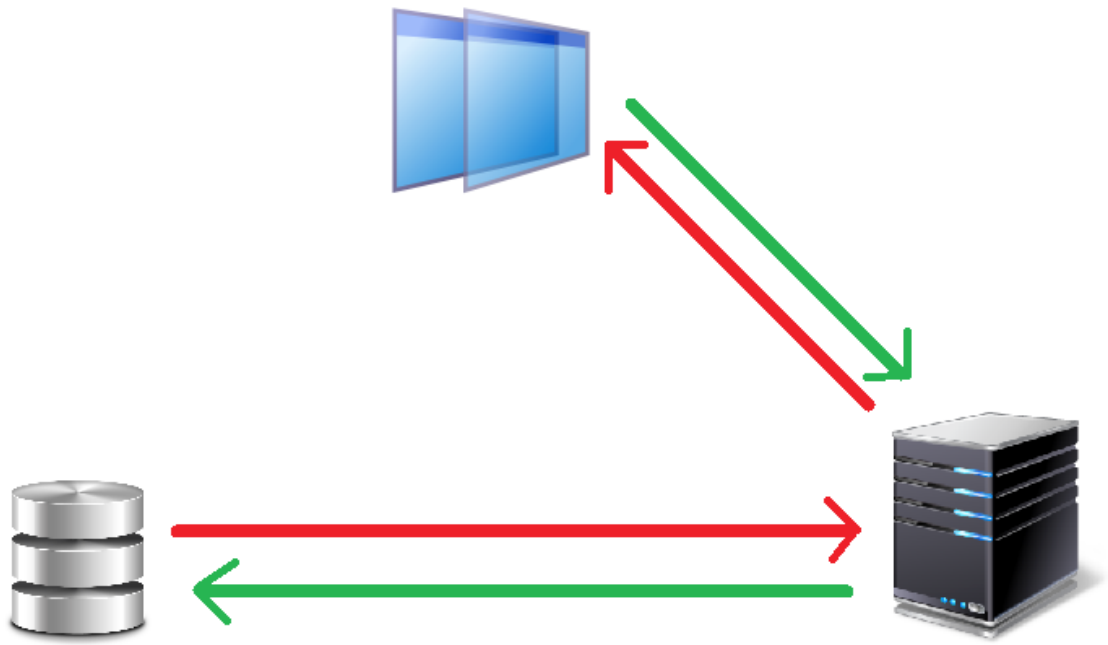
Baza de date va avea tabele corespunzătoare pentru fiecare tip de fișiere, precum și o tabelă pentru reținerea datelor despre utilizatori.



Tabelele bazei de date sunt: users – va reține toate informațiile menționate anterior despre utilizatori, documents, images, media și others ce vor reține informațiile despre fișiere și o tabelă ce va face legătura între utilizatori și fișierele sale numită userdata.

2. Diagrama de Module și Componente

Principalele componente ce pot fi regăsite în implementarea proiectului sunt constituite din **Server**, **Baza De Date** și **Interfata Grafică** (GUI) fiecare având roluri importante bine stabilite.

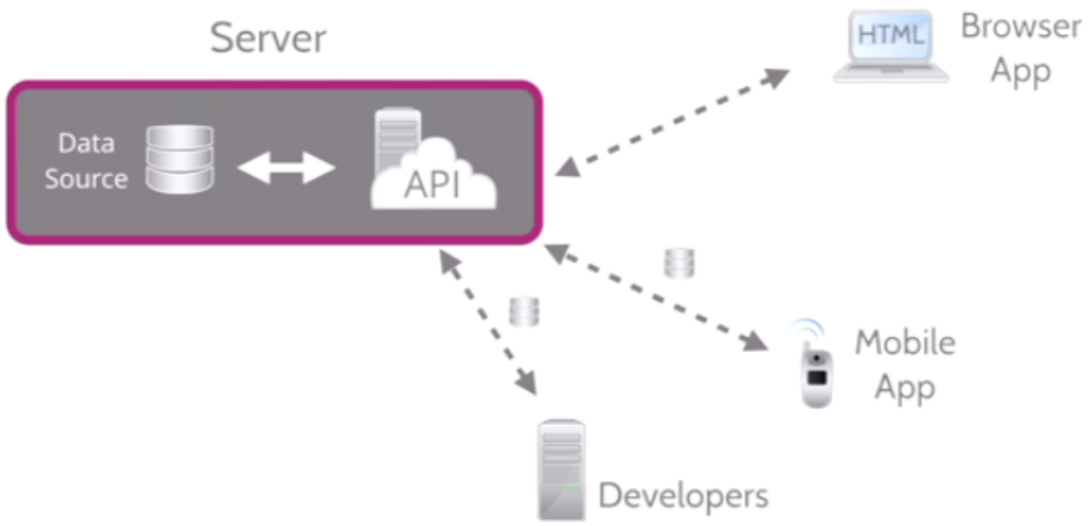


Interfata Grafică reprezintă cel mai aporopiat layer de user cu care acesta are interacțiune directă și prin intermediul căruia va transmite diferite solicitări către server și baza de date în scopul preluării unor informații solicitate necesare.

Server-ul are ca scop principal furnizarea de servicii și operearea continuă în rețeaua sa așteptând diferite solicitări din partea altor calculatoare din rețea sau din partea anumitor aplicații pe care apoi să le proceseze.

Baza de Date reprezintă o modalitate de stocare a unor informații și date cu posibilitatea extinderii ușoare și a regăsirii rapide a acestora. Totalitatea datelor sunt memorate în interiorul unor tabele între care există relații cu ajutorul cărora se vor face diferite interogări și legături dintre acestea.

După cum se poate observa din schema de mai sus există două tipuri de request-uri, cele de intrare și cele de ieșire, verde respectiv roșu.
Prin intermediul Interfeței Grafice user-ul sau adminul crează o anumită cerere(request) asupra unei anumite date. Această cerere este procesată de către server și transmisă mai departe bazei de date care va procesa cererea și va returna datele cerute cu ajutorul interogăriilor. Odată procesate datele se vor returna către server care mai apoi le va transmite interfeței utilizator. Astfel fiind posibilp afișarea și procesarea în timp real al datelor și cererilor solicitate.



Sursa: http:// angularjs.org

Într-un sens mai larg putem extinde diagrama principală și a vedea Server-ul ca fiind un BlackBox pentru un utilizator obișnuit care doar trimite anumite cereri către acesta.
În interiorul acestei cutii negre server-ul preia cerere și o transmite mai departe bazei de date care va întoarce rezultatul cerut, mai apoi server-ul oferind răspunsul diferiților utilizatori.
Întreaga comunicare se face prin intermediul unei interfețe (API) care oferă o modelare mai simplistă pentru diferitele cereri efectuate. Utilizatori necunoscand implementarea internă.
Se poate obseva ca cereriile pot veni de la mai multe tipuri de utilizatori, aplicații sau platforme precum cele mobile sau Desktop.

Biografie

http://softparcauto.ro/cfts-soft-parc-auto-online/ce-este-cloud-computing-ul/

http://ro.wikipedia.org/wiki/Cloud_computing

http://stackoverflow.com/questions/16820336/what-is-saas-paas-and-iaas-with-examples

http://en.wikipedia.org/wiki/Platform_as_a_service

http://angularjs.org

http://www.linuxfoundation.org