

COLEGIUL NAȚIONAL

„MIHAI VITEAZUL”

ATESTAT LA INFORMATICĂ

chatbox

Profesor îndrumător:

STAN

MIHAELA-VERONICA

Elev:

VÎJIALĂ

TUDOR-GABRIEL

18 mai, 2015

1 Introducere

chat – s.f. conversație informală

to box – v. tranz. a închide, a separa (într-o cutie)

– Dicționar englez-român

Chatbox este un serviciu de mesagerie Web.

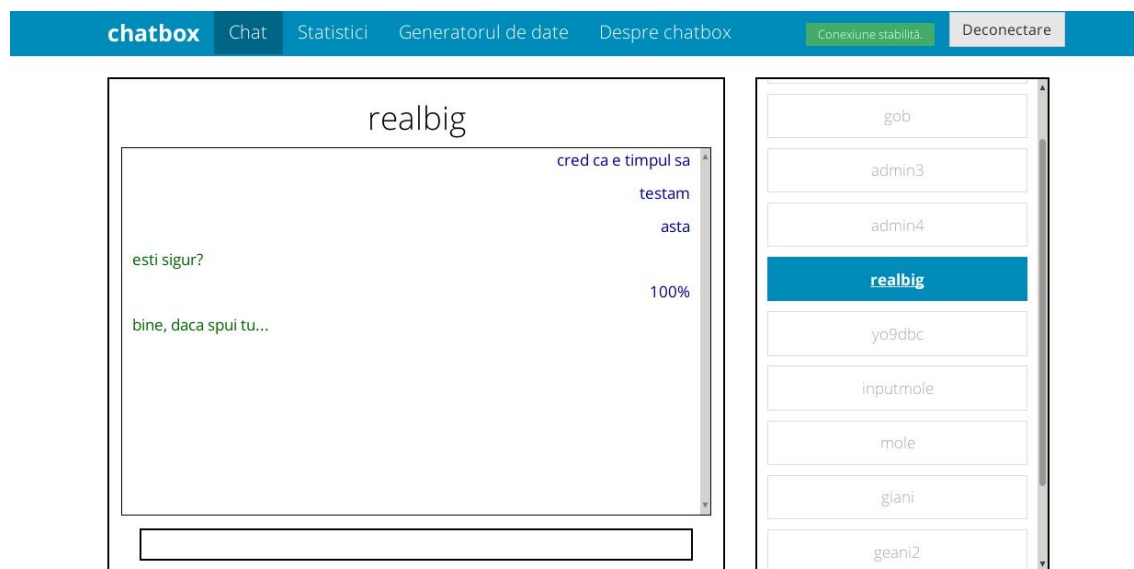
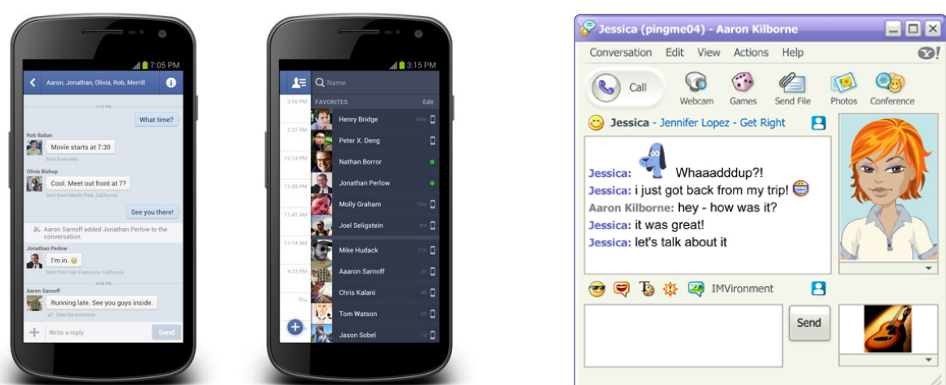


Figura 1: Interfața modului de mesagerie

Acest proiect a fost inspirat de serviciile de „mesagerie instantanee” precum *Y!Messenger*, *Google Talk* sau *Facebook Messenger*.



(a) Facebook Messenger

(b) Yahoo! Messenger

Figura 2: Servicii de mesagerie digitală

În prezent, **Chatbox** poate fi accesat la adresa **penultim.ddns.net/chatbox**.

2 Resurse logice

Resursele logice sunt componentele software ale calculatorului care au funcții de administrare a resurselor și a datelor.

În cazul proiectului **Chatbox**, acestea reprezintă mijlocul prin care pagina Web este programată, monitorizată și administrată.

2.1 Sistemul MySQL

MySQL¹ este cel mai folosit SGBD² open-source, la ora actuală. Produs inițial de compania suedeză MySQL AB și distribuit sub Licența Publică Generală GNU¹⁶, în prezent MySQL este dezvoltat de Corporația Oracle.

Ca instrument de management pentru bazele de date MySQL este folosită o aplicație PHP numită phpMyAdmin.

2.2 Limbajul PHP

Limbajul de programare PHP³ este folosit pe scară largă în dezvoltarea paginilor și aplicațiilor web.

Se folosește în principal înglobat în codul HTML, dar poate fi utilizat și pentru programarea aplicațiilor CLI (linie de comandă).

PHP este disponibil sub Licența PHP și Free Software Foundation îl consideră a fi un software liber¹⁶.

2.2.1 Librăria PDO

PHP Data Objects⁴ este o componentă PHP ce permite accesarea unor SGBD din programe PHP. **Chatbox** folosește în mod exclusiv componenta PDO pentru accesarea bazei de date.

2.2.2 Librăria Ratchet

Websocket⁵ este un protocol ce furnizează o conexiune full-duplex prin o legătură TCP. Tehnologia Websocket a fost dezvoltată odata cu inițiativa de inovare HTML5.

Folosind tehnologia Websocket, se pot trimite date în timp real între client și server. **Chatbox** folosește acest protocol pentru transmiterea instantanee a mesajelor și a datelor.

Ratchet⁶ este o librărie ce permite utilizarea protocolului Websocket în limbajul PHP.

2.3 Limbajul JavaScript

JavaScript (sau ECMAScript) este un limbaj de programare orientat pe obiecte⁷ ce rulează în browserele utilizatorilor.

Limbajul este binecunoscut pentru folosirea sa în construirea paginilor Web. În ciuda numelui și a unor similarități în sintaxă, între *JavaScript* și limbajul *Java* nu există nicio legătură.

2.3.1 Tehnica AJAX

O tehnică de construire a paginilor web tot mai întâlnită în ultimul timp este AJAX, abreviere de la „Asynchronous JavaScript and XML”. Această tehnică constă în executarea de cereri HTTP în fundal, fără a reîncărca toată pagina Web, și actualizarea numai anumitor porțiuni ale paginii.

Chatbox folosește tehnica AJAX pentru a încărca o multitudine de elemente, precum textul mesajelor primite sau lista de utilizatori activi.

2.3.2 Librăria jQuery

jQuery⁸ este o platformă de dezvoltare JavaScript, concepută pentru a ușura și îmbunătăți procese precum traversarea arborelui DOM⁹ în HTML, managementul evenimentelor, animații și cereri de tip AJAX.

2.3.3 Librăria chart.js

Chart.js¹⁰ este o librărie JavaScript ce permite afișarea unor grafice în mod dinamic. Graficele sunt redimensionate, la nevoie, după mărimea ecranului clientului.

Proiectul **Chatbox** folosește chart.js pentru afișarea rezultatelor numerice a unor interogări SQL și pentru vizualizarea de statistici.

2.3.4 Librăria Bootstrap

Bootstrap este cel mai popular framework de HTML, CSS, și JavaScript dedicat dezvoltării proiectelor Web.

Bootstrap permite aranjarea elementelor grafice într-un mod potrivit mărimii ecranului vizitatorului. Astfel, site-ul va fi afișat corect atât pe calculatoare Desktop, cât și pe tablete și telefoane.

2.4 Limbajul Python

Python este un limbaj de programare dinamic multi-paradigmă¹², creat în 1989 de către programatorul olandez Guido van Rossum¹³.

Python pune accentul pe curățenia și simplitatea codului, iar sintaxa sa le permite dezvoltatorilor să exprime unele idei programatice într-o manieră cât mai clară și mai concisă.

Generatorul de date folosit de **Chatbox** este implementat în Python 3.

2.5 Despre serverul Linux

Codul proiectului **Chatbox** rulează pe un calculator personal *HP Compaq 6005 Pro* care servește, printre altele, drept server Web. Sistemul are următoarele caracteristici:

- Procesor: AMD Athlon II X2 2.8GHz
- Memorie: 2GB RAM DDR3
- Hard Drive: 2TB, Samsung
- Placă de rețea: 100Mbps
- Sistem de operare: Debian 8

Pentru a dispune de o adresă permanentă a acestui server, am folosit serviciile de DNS dinamic ale companiei NoIP¹⁴.

2.5.1 Sistemul de operare Debian 8

Debian¹⁵ este un sistem de operare compus din software liber¹⁶, și o distribuție populară și foarte influentă între distribuțiile GNU/Linux.

Versiunea 8 a sistemului de operare este cunoscută în prezent sub numele de *testing*. Pachetele de software pentru versiunea *testing* sunt, după cum sugerează și numele, în curs de testare. Totuși, sunt suficient de stabile pentru modul în care este utilizat acest sistem.

2.5.2 Serverul HTTP Apache 2

Apache este un server HTTP open-source. Acesta reprezintă standardul în industria de Web Hosting, fiind cel mai folosit server HTTP, susținând 53.34% din site-urile Web¹⁷.

2.5.3 Sistemul de monitorizare daemontools

Daemontools¹⁸ este o colecție de unelte software folosite pentru controlul și monitorizarea serviciilor UNIX.

Daemontools este folosit de **Chatbox** pentru a monitoriza serverul de *websockets* și a-l reporni în eventualitatea unei erori.

2.6 Procesorul L^AT_EX

Acest document a fost compus folosind sistemul L^AT_EX care permite prepararea acestuia pentru tipărire în format electronic, cu ajutorul limbajului de programare T_EX.

3 Descrierea proiectului

Chatbox a fost conceput ca un mijloc de a învăța atât despre programarea Web în general, cât și despre sistemele de transmitere de date în timp real.

3.1 Baza de date

Sistemul **Chatbox** stochează în baza de date toate detaliile interacționării utilizatorilor cu acesta. Informațiile sunt atât funcționale (conținutul mesajelor trimise), cât și statistici (ce sistem de operare a fost folosit pentru autentificarea unui utilizator).

Structura tabelor este următoarea:

Tabel	Nume	Tip	Observații
utilizatori	id	bigint(20)	PK: identificator numeric
	nume	tinytext	numele utilizatorului
	hash	varchar(512)	hash-ul parolei ¹⁹
	data_ire	datetime	data înregistrării
	activ	tinyint(1)	1 - activ, 0 - inactiv
sesiuni	id_sesiune	bigint(20)	PK
	cheie_sesiune	varchar(512)	
	id_utilizator	bigint(20)	FK
	inceput	datetime	momentul de log-in
	sfarsit	datetime	momentul de log-out
	adresa_ip	tinytext	
	browser	tinytext	browserul folosit
	platforma	tinytext	SO folosit
mesaje	id_mesaj	bigint(20)	PK
	id_expeditor	bigint(20)	FK
	id_destinatar	bigint(20)	FK
	text	text	conținutul mesajului
	data	datetime	momentul expedierii
	citit	tinyint(1)	1 - citit, 0 - necitit

Figura 3: Structura bazei de date

3.2 Diagrama de secvență

Unified Modeling Language (prescurtat UML) este un limbaj standard pentru descrierea de modele și specificații pentru software²⁰.

Următoarea diagramă descrie interacțiunile dintre cele patru componente de bază ale sistemului: browserul utilizatorului, serverul Web, baza de date și serverul Web-socket.

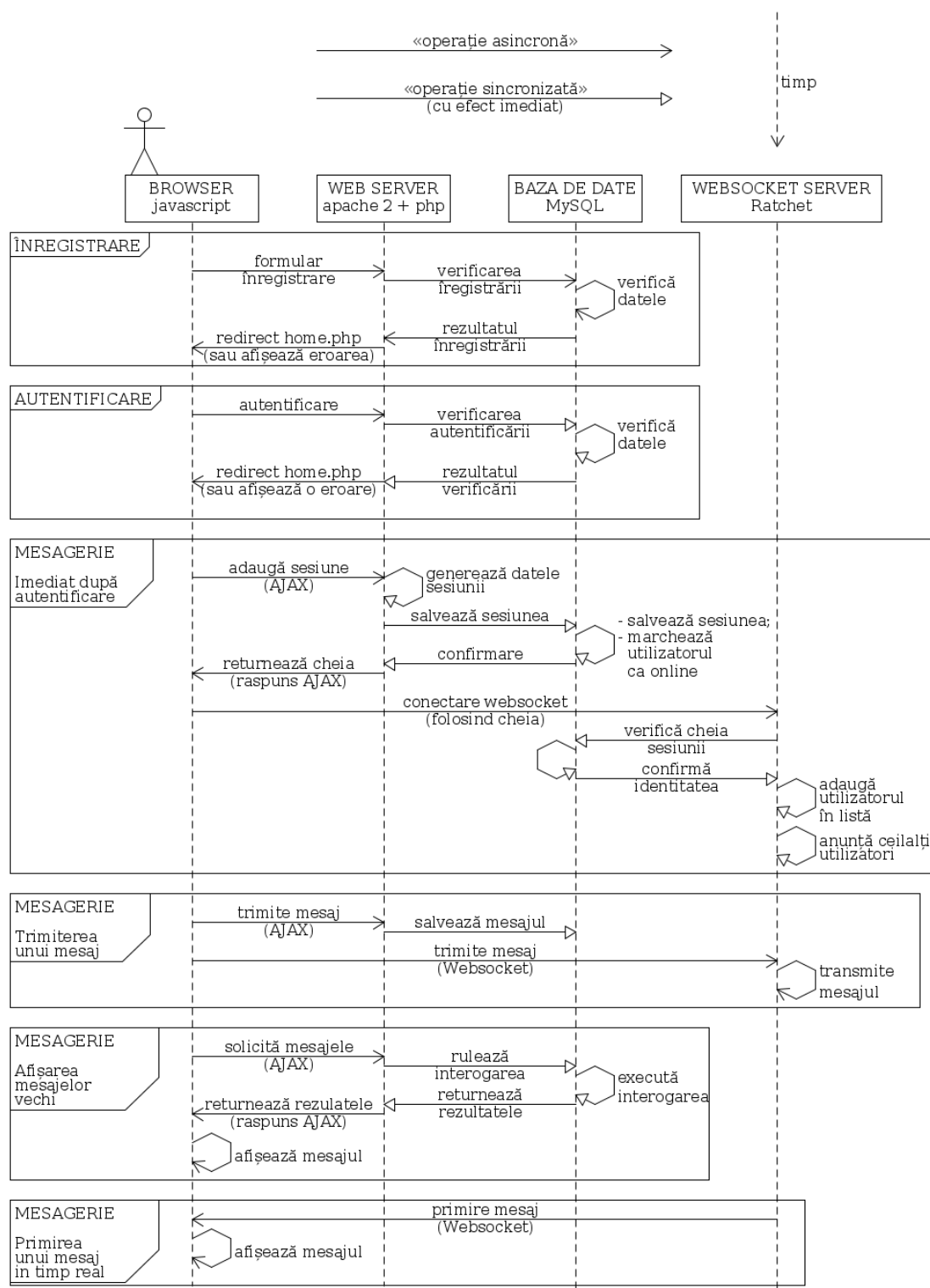


Figura 4: Diagrama de secvență pentru modulul de mesagerie

3.3 Sistemul de autentificare

Pentru a folosi serviciul de mesagerie, un utilizator trebuie să fie înregistrat și autentificat. Înregistrarea este liberă; singurele date necesare sunt numele de utilizator și o parolă.

Înregistrare

Autentificare

Figura 5: Formulare de înregistrare și autentificare

În cazul în care apar erori în procesul creării contului sau în procesul autentificării, utilizatorul va primi o înștiințare pe fundal roșu care explică situația.

Aceste verificări se fac pe server, în cate un script PHP, pentru verificarea autentificării, respectiv a înregistrării.

!

• Autentificare eșuată

!

- Parola nu poate fi mai scurta de 6 caractere.
- Parolele nu coincid.
- Numele de utilizator are caractere interzise.

Figura 6: Posibile erori după completarea eronată a formularelor

Paginile accesate pentru autentificare și înregistrare sunt `autentificare.php` și `inregistrare.php`. Verificarea fiecărui procedeu este efectuată în programele `php/verifica_autentificare.php` și `php/verifica_inregistrare.php`.

Pentru exemplificare este atașată mai jos sursa fișierului responsabil de verificarea autentificării.


```
<?php
/* php/verifica_authenticare.php */
require_once('functii.php');
require_once("bd_functii.php");
session_start();
unset($_SESSION['ERORI']);

$mesaje_eroari = array();
$savem_eroare = false;

$nume = $_POST['nume'];
$parola = $_POST['parola'];

if ($nume == '') {
    $savem_eroare = true;
    $mesaje_eroari[] = "Numele de utilizator lipseste. ";
}
if ($parola == '') {
    $savem_eroare = true;
    $mesaje_eroari[] = "Parola lipseste. ";
}
if (preg_match('/\\W|_|_', $nume)) {
    $savem_eroare = true;
    $mesaje_eroari[] = "Numele de utilizator contine caractere interzise. ";
}
if ($savem_eroare == true) {
    $_SESSION['ERORI'] = $mesaje_eroari;
    redirect("../autenticare.php");
} else {
    unset($_SESSION['ERORI']);
}

conectare_baza_date();
$query = "SELECT * FROM utilizatori WHERE nume=?";
$fetch = interogare_bd($query, $nume);
$hash = $fetch['hash'];

if (password_verify($parola, $hash)) {
    // autentificare ok
    $_SESSION['ID'] = $fetch['id'];
    $_SESSION['NUME'] = $fetch['nume'];
    redirect("../home.php");
} else {
    // autentificare esuata
    $mesaje_eroari[] = "Autentificare esuata";
    $_SESSION['ERORI'] = $mesaje_eroari;
    redirect("../autenticare.php");
}
```

3.4 Interfață grafică și design

Pentru construirea interfeței grafice și a design-ului am folosit librăria Bootstrap¹¹ și aditii din partea site-ului Bootswatch²¹.

Chatbox abordează un stil minimalist, folosind doar culori elementare: albastru pentru interfață și meniu, roșu pentru erori. Culorile sunt alese folosind pachetul Yeti²² din colecția Bootswatch.

3.4.1 Funcția de remodelare

Bootstrap are la centrul său conceptul de „responsive web design”²³, adică modelarea componentelor vizuale pentru a „răspunde” cât mai bine la tipul de terminal folosit. Astfel, site-ul va avea altă structură dacă este afișat pe un Desktop PC, decât dacă este afișat pe un ecran de telefon.

Pentru exemplificare, comparați modul de afișare al modului de mesagerie, între răspunsul site-ului pentru un Desktop PC (pagina 1, Figura 1) și un ecran de telefon mobil (Figura 7).



Figura 7: Interfața modului de mesagerie, pe un terminal mobil

3.4.2 Bara de navigație

Bara albastră din partea de sus a ecranului este prezentă în fiecare pagină a site-ului **Chatbox** și se numește *bară de navigație*. Aceasta este folosită ca meniu pentru site, aici fiind afișate toate secțiunile de interes pe care utilizatorii le pot vizita.



Figura 8: Bara de navigație, afișată pe un sistem Desktop PC.

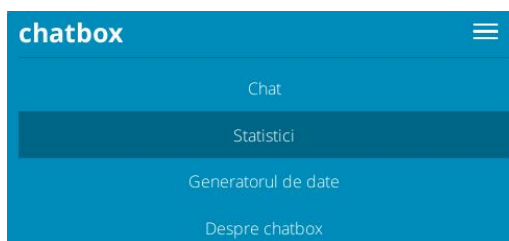


Figura 9: Bara de navigație, afișată pe un terminal mobil.

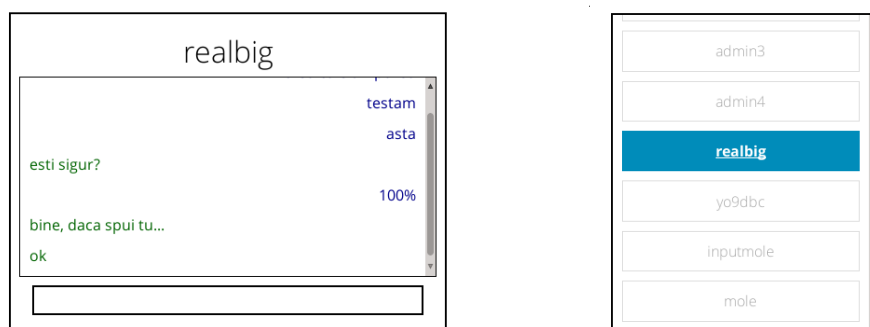
Pentru ecrane de rezoluție mică, precum cele de telefon sau tabletă, bara de navigație își schimbă forma și poate fi extinsă sau micșorată la nevoie, folosind butonul din dreapta-sus [Figura 9].

3.5 Sistemul de mesagerie

Funcționalitatea centrală a proiectului este sistemul de mesagerie în timp real. Acesta permite trimiterea de mesaje text către orice alt utilizator al **Chatbox**.

3.5.1 Interfața specifică

Ca să se păstreze stilul minimalist, interfața are un design simplu și funcțional. Ecranul este împărțit între o listă cu ceilalți utilizatori și o casuță unde apar mesajele din conversația curentă.



(a) Căsuța cu mesaje

(b) Lista de utilizatori

Figura 10: Interfața sistemului de mesagerie

Se poate schimba utilizatorul curent prin selectarea unui altul, folosind un simplu *click*. Utilizatorul activ este afișat în partea de sus a căsuței cu mesaje.



(a) Utilizator offline

(b) Mesaj nou

(c) Utilizator online

Figura 11: Elementele listei de utilizatori

Trimiterea unui mesaj se poate realiza prin tastarea acestuia în zona de trimitere și prin apăsarea tastei *enter*. Din acest motiv, am considerat că era inutil să se adauge și un buton care împlinește același rol. Alegerea a fost inspirată din design-ul chat-ului implementat de Facebook.

Mesajele se pot trimite doar dacă clientul este conectat la serverul de *websocket*, care are rolul de a primi și de a trimite mai departe conținutul mesajelor de la un utilizator la altul, în timp real. **Chatbox** anunță utilizatorii dacă sunt conectați printr-un element de afișaj.



Figura 12: Butoanele specifice interfeței de chat

3.5.2 Serverul de websockets

După cum a fost explicat mai sus, protocolul este folosit pentru a transmite mesajele de la expeditor la destinatar, într-un timp cât mai scurt. Acest procedeu se realizează, evident, doar dacă atât expeditorul cât și destinatarul sunt conectați la server.

Librăria *Ratchet* este folosită pentru a implementa acest server, ce rulează la adresa paginii **Chatbox**, pe portul 8091.

```
<?php
/* php/websocket/ChatListener.php */

class ChatListener implements MessageComponentInterface {
    // Lista cu clienti activi
    protected $clienti;
    protected $legaturi = array(0 => null);

    public function __construct() {
        $this->clienti = new \SplObjectStorage;
        echo 'ChatListener initializat!\n';
    }

    // Trimite mesajul $mesaj de la expeditor.
    function trimite_mesaj($expeditor, $mesaj) {
        $trimis = false;
        $mesaj['id_expeditor'] = $this->legaturi[$expeditor->resourceId]['id'];
        foreach ($this->clienti as $client) {
            if (/* ... */ ) {
                $client->send(json_encode($mesaj));
                $trimis = true;
            }
        }
        return $trimis;
    }

    // Initializeaza o sesiune folosind cheia primita.
    function initializeaza_sesiune($utilizator, $cheie) {
        $transmisie = array(/* ... */);
        $transmisie_text = json_encode($transmisie);
        foreach ($this->clienti as $client) {
            if ($utilizator !== $client) {
                $client->send($transmisie_text);
            }
        }
        $q = "UPDATE utilizatori SET activ = 1 WHERE id = ?";
        inserare_bd($q, $id); deconectare_baza_date();
    }

    // Inchide o sesiune si seteaza utilizatorul ca fiind inactiv.
    function sfarseste_sesiune($conexiune, $id, $cheie) {
        $transmisie = array(/* ... */);
        $transmisie_text = json_encode($transmisie);
        foreach ($this->clienti as $client) {
            if ($conexiune !== $client) {
                $client->send($transmisie_text);
            }
        }
        inserare_bd("UPDATE utilizatori SET activ = 0 WHERE id = ?", $id);
        inserare_bd("UPDATE sesiuni SET sfarsit = ? WHERE cheie_sesiune = ?",
            array(\acum(), $cheie));
        deconectare_baza_date();
    }
}
```

```
// Rulat in momentul in care un utilizator se
// conecteaza la server.
public function onOpen(ConnectionInterface $conexiune) {
    $this->clienti->attach($conexiune);
}

// Rulat in momentul in care un utilizator trimite un mesaj
// catre server. Acest mesaj poate fi de doua tipuri
// (trimiteri de mesaj text, sau operatiune).
public function onMessage(ConnectionInterface $expeditor, $mesaj) {
    $resId = $expeditor->resourceId;
    $date = json_decode($mesaj, true);

    switch ($date['operatie']) {
        case 'trimiteri':
            $this->trimite_mesaj($expeditor, $date);
            break;
        case 'initializare':
            $this->initializeaza_sesiune(
                $expeditor, $date['cheie']);
            break;
    }
}

// Rulat in momentul in care un utilizator se
// deconecteaza de la server.
public function onClose(ConnectionInterface $conexiune) {
    $this->clienti->detach($conexiune);

    $id = $this->legaturi[$conexiune->resourceId]['id'];
    $nume = $this->legaturi[$conexiune->resourceId]['nume'];
    $cheie = $this->legaturi[$conexiune->resourceId]['cheie'];

    $this->sfarseste_sesiune($conexiune, $id, $cheie);
    unset($this->legaturi[$conexiune->resourceId]);
}

// Rulat daca apar erori.
public function onError(ConnectionInterface $conexiune, \Exception $e) {
    $conexiune->close();
}
}
?>

<?php
/* php/websocket/server.php */
use Ratchet\Server\IoServer;
use Ratchet\Http\HttpServer;
use Ratchet\WebSocket\WsServer;
use Chatbox\ChatListener;

$server = IoServer::factory(new HttpServer(
    new WsServer(new ChatListener()),
    8091);
$server->run();
```

3.6 Generarea și vizualizarea datelor

Pentru a verifica corectitudinea și eficiența algoritmilor folosiți de **Chatbox**, am realizat un sistem ce generează date și construiește grafice prin care ele sunt vizualizate.

3.6.1 Generatorul de date

Generatorul de date este un subprogram !!!!!

3.6.2 Vizualizarea datelor prin grafice

Vizualizarea datelor joacă un rol foarte important în înțelegerea lor.

Sistemul creează câte o vizualizare a datelor numerice rezultate dintr-o interogare SQL. Procedul este inovativ prin faptul că se poate alipi unei infrastructuri deja existente pentru a genera grafice cu datele prezente.

Întreaga analiză a bazelor de date poate fi găsită la adresa:

[penultim.ddns.net/
chatbox/statistici.php](http://penultim.ddns.net/chatbox/statistici.php).

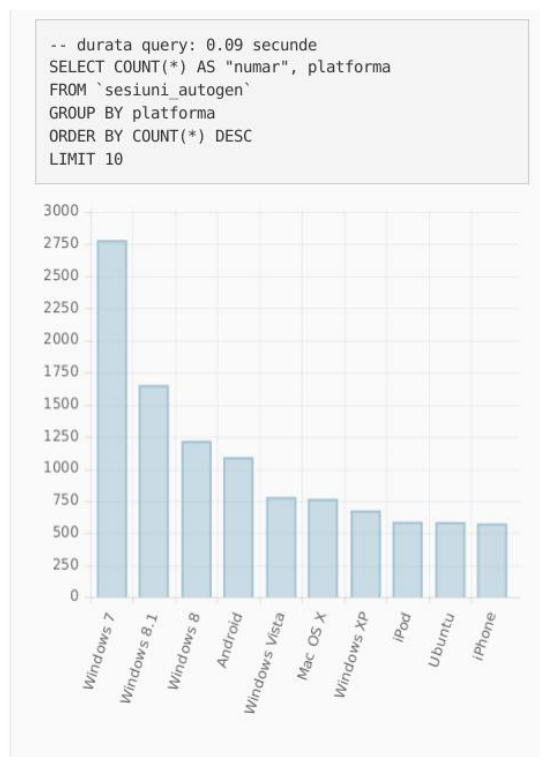


Figura 13: Grafic rezultat dintr-o interogare SQL

4 Bibliografie

- [1] MySQL
<http://www.mysql.com/about/>
- [2] Système
de
tion
Base
Données
http://ro.wikipedia.org/wiki/Sistem_de_gestiune_a_bazelor_de_date
- [3] PHP Hypertext Processor
<http://php.net/manual/en/intro-what-is.php>
- [4] PDO: PHP Data Objects
<http://php.net/manual/en/intro.pdo.php>
- [5] Websocket Protocol
<https://www.websocket.org/>
- [6] Ratchet: a Websocket Library
<http://socketo.me/>
- [7] Objects in Javascript
http://www.w3.org/community/webed/wiki/Objects_in_JavaScript
- [8] jQuery: The Write Less, Do More, JavaScript Library
<https://jquery.com/>
- [9] Document Object Model
<http://www.w3.org/DOM/>
- [10] Chart.js: Open source HTML5 Charts
<http://www.chartjs.org/>
- [11] Bootstrap – Designed for everyone, everywhere
<http://getbootstrap.com/>
- [12] Python
<https://www.python.org/about/>
- [13] Python, articol Wikipedia
<http://ro.wikipedia.org/wiki/Python>

- [14] No-IP: a Dynamic DNS company
<http://www.noip.com/>
- [15] Debian GNU/Linux
<https://www.debian.org/intro/about>
- [16] Software liber / Free Software
<https://www.fsf.org/about/what-is-free-software>
- [17] June 2013 Web Server Survey
<http://news.netcraft.com/archives/2013/06/06/june-2013-web-server-survey-3.html>
- [18] daemontools
<http://cr.yp.to/daemontools.html>
- [19] PHP5 PASSWORD_HASH
<http://php.net/manual/en/function.password-hash.php>
- [20] Unified Modeling Language
<http://www.uml.org/>
- [21] Bootswatch: Free themes for Bootstrap
<https://bootswatch.com/>
- [22] Bootswatch Yeti Theme: A friendly foundation
<https://bootswatch.com/yeti/>
- [23] Responsive Web Design
<http://alistapart.com/article/responsive-web-design>
http://en.wikipedia.org/wiki/Responsive_web_design

Cuprins

1	Introducere	1
2	Resurse logice	2
2.1	Sistemul MySQL	2
2.2	Limbajul PHP	2
2.2.1	Librăria PDO	2
2.2.2	Librăria Ratchet	2
2.3	Limbajul JavaScript	2
2.3.1	Tehnica AJAX	3
2.3.2	Librăria jQuery	3
2.3.3	Librăria chart.js	3
2.3.4	Librăria Bootstrap	3
2.4	Limbajul Python	3
2.5	Despre serverul Linux	4
2.5.1	Sistemul de operare Debian 8	4
2.5.2	Serverul HTTP Apache 2	4
2.5.3	Sistemul de monitorizare daemontools	4
2.6	Procesorul L ^A T _E X	4
3	Descrierea proiectului	5
3.1	Baza de date	5
3.2	Diagrama de secvență	5
3.3	Sistemul de autentificare	7
3.4	Interfață grafică și design	9
3.4.1	Funcția de remodelare	9
3.4.2	Bara de navigație	9
3.5	Sistemul de mesagerie	10
3.5.1	Interfața specifică	10
3.5.2	Serverul de websockets	11
3.6	Generarea și vizualizarea datelor	13
3.6.1	Generatorul de date	13
3.6.2	Vizualizarea datelor prin grafice	13
4	Bibliografie	14