

Cuprins

| | |
|---------------------------------------|-----------|
| 1 Introducere | 2 |
| 1.1 Scopul documentui | 2 |
| 1.2 Definire termeni tehnici | 2 |
| 1.3 Rezumat document | 4 |
| 2 Descriere generală aplicație | 5 |
| 2.1 Scopul aplicației | 5 |
| 2.2 Rezumat funcționalități | 5 |
| 2.3 Constrângeri majore | 8 |
| 2.4 Beneficiu/Valoare | 9 |
| 3 Proiectare arhitecturală | 9 |
| 3.1 Diagrama arhitecturii pe nivele | 9 |
| 3.2 Precizarea tehnologiilor folosite | 10 |
| 4 Modelarea datelor | 13 |
| 4.1 Diagrama claselor | 13 |
| 4.2 Structura bazei de date | 14 |
| 5 Interfața grafică | 15 |

1 Introducere

1.1 Scopul documentui

Acest document are rolul de a descrie funcționalitatea și constrângerile unei aplicații web based de timesheet management .

Astfel, se dorește ca angajații unei companii să își poată organiza programul zilnic al activităților în cadrul unui timesheet lunar. Aplicația permite, printre altele vizualizarea detaliilor unui angajat, al unui proiect sau generarea unor rapoarte pe anumite perioade. Detaliile user case-urilor sunt prezentate la 2.2.

Acest document poate fi folosit și înțeles atât de utilizatorii aplicației și membrii echipei dezvoltatoare, cât și de posibili viitori dezvoltatori, codul sursă fiind disponibil open source.

1.2 Definire termeni tehnici

Administrator (admin) - utilizatorul care are anumite drepturi specifice unui „operator de sistem”: ce divizii există în companie, ce departamente sunt în cadrul unei divizii, ce joburi sunt specifice companiei, care sunt angajații din companie, configurează și activează mecanismul de audit.

API (Application Programming Interface) - denumirea în limba engleză a unei interfețe pentru programarea de aplicații. De obicei este vorba despre interfața dintre programele de aplicație și sistemul de operare, care stabilește în amănunt modul în care programele de aplicație pot accesa (apela) serviciile sistemului de operare sub care rulează.

Bază de date - modalitate de stocare a unor informații și date pe un suport extern (un dispozitiv de stocare), cu posibilitatea extinderii ușoare și a regăsirii rapide a acestora.

Bază de date back-end - bază de date care este accesată de către utilizatori indirect printr-o aplicație externă, mai degrabă decât de programare a aplicațiilor stocate în baza de date în sine sau prin manipulare nivel scăzut de date (de exemplu, prin comenzi SQL).

Cookie - text special, deseori codificat, trimis de un server unui navigator web și apoi trimis înapoi (nemodificat) de către navigator, de fiecare dată când accesează acel server. Cookie-urile sunt folosite pentru autentificare precum și pentru urmărirea comportamentului utilizatorilor; aplicații tipice sunt reținerea preferințelor utilizatorilor și implementarea sistemului de „coș de cumpărături”.

CSS (Cascading Style Sheets) - standard pentru formatarea elementelor unui document HTML. Stilurile se pot atașa elementelor HTML prin intermediul unor fișiere externe sau în cadrul documentului, prin elementul <style> și/sau atributul style. CSS se poate utiliza și pentru formatarea elementelor XHTML, XML și SVG.

Framework - ansamblu standardizat de concepte, practici și criterii pentru a se aplica asupra unui tip particular de problemă, ce servește ca referință pentru a propune și rezolva probleme cu conotații similare.

HTTP (Hypertext Transfer Protocol) - metoda cea mai des utilizată pentru accesarea informațiilor în Internet care sunt păstrate pe servere World Wide Web (WWW). Protocolul HTTP este un protocol de tip text, fiind protocolul "implicit" al WWW.

JavaScript - limbaj de programare orientat obiect bazat pe conceptul prototipurilor. Este folosit mai ales pentru introducerea unor funcționalități în paginile web, codul Javascript din aceste pagini fiind rulat de către browser.

Login - acțiunea prin care un utilizator introduce datele necesare pentru obținerea accesului la o aplicație web

MVC (Model-view-controller) - model arhitectural utilizat în [ingineria software](#). Succesul modelului se datorează izolării logicii de business față de considerentele interfeței cu utilizatorul, rezultând o aplicație unde aspectul vizual sau/și nivelele inferioare ale regulilor de business sunt mai ușor de modificat, fără a afecta alte nivele.

Rack - interfață minimală, modularizată și adaptabilă pentru dezvoltarea de aplicații web în Ruby. Prin împachetarea de cereri și răspunsuri HTTP în cel mai simplu mod

posibil, se unifică API-ul pentru servere de web, framework-uri, și software-ul dintre acestea (așa-numita middleware) într-un singur apel de metodă.

SOAP (Simple Object Access Protocol) - specificație de protocol pentru schimbul de informații structurate în punerea în aplicare a Web Services în rețele de calculatoare. Ea se bazează pe informații XML Set pentru formatul mesajului, și, de obicei, se bazează pe alte protocoale de aplicare strat, mai ales HTTP sau SMTP, pentru mesajul de negociere și de transmisie.

XML (eXtensible Markup Language) - meta-limbaj de marcare recomandat de Consorțiul Web pentru crearea de alte limbaje de marcare, cum ar fi XHTML, RDF, RSS, MathML, SVG, OWL etc.

1.3 Rezumat document

Capitolul 1 conține o descriere a documentului, conținutul, și scopul creării acestuia.

Capitolul 2 cuprinde descrierea, nu foarte amplă, a produsului prezentat în acest document: necesitatea aplicației, funcțiile sale, precum și beneficiile aduse în cadrul unei companii.

Capitolul 3 descrie arhitectura pe nivele a aplicației, pentru fiecare tier specificându-se tehnologia/framework-ul ales pentru implementarea sa.

Capitolul 4 prezintă modelarea datelor evidențiată prin diagrama claselor și prin detalierea structurii bazei de date, descriindu-se fiecare tabelă (câmpuri, chei primare/straine, constrângeri, etc.).

Capitolul 5 conține prezentarea interfeței grafice (GUI) prin prezentarea “Home page”-ului, a paginii “Timesheet”-ului, a paginii “Fișa zilnică și cea de Admin..

2 Descriere generală aplicație

2.1 Scopul aplicației

Produsul prezentat în acest document dorește a simula pontajul, operația prin care se înregistrează prezența muncitorilor la locul de muncă, cantitatea produselor efectuate într-un anumit interval de timp sau timpul folosit pentru efectuarea unei lucrări, în vederea calculării salariului sau a plății.

2.2 Rezumat funcționalități

În cadrul aplicației vor fi vizibile următoarele operații:

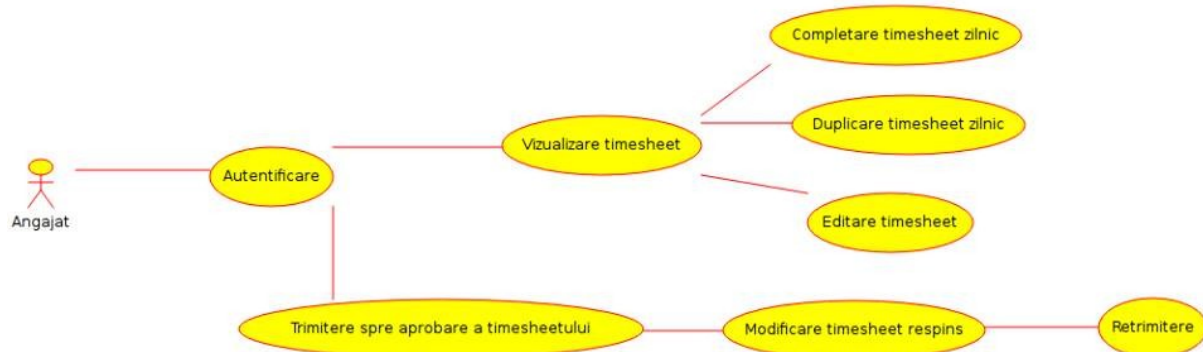
- pentru **admin**:



- introducerea/ștergerea diviziilor
- introducerea/ștergerea departamentelor
- introducerea/ștergerea angajaților
- configurare/activare mecanism audit

- pentru **angajat**:

- completarea fișei lucratoare de la 0 sau duplicarea unei fișe deja salvate în fiecare zi de servici prin introducerea activităților depuse, atât pentru cele 8 ore de lucru cât și pentru orele extra, dacă este cazul
- trimiterea timesheet-ului la sfârșitul unei luni calendaristice sau chiar și după, în urma primirii unei notificari conform căreia trebuie să-și trimită timesheet-ul
- editarea timesheet-ului făcând modificările necesare și retrimiterea acestuia șefului de departament în urma unei rejectari a timesheet-ului



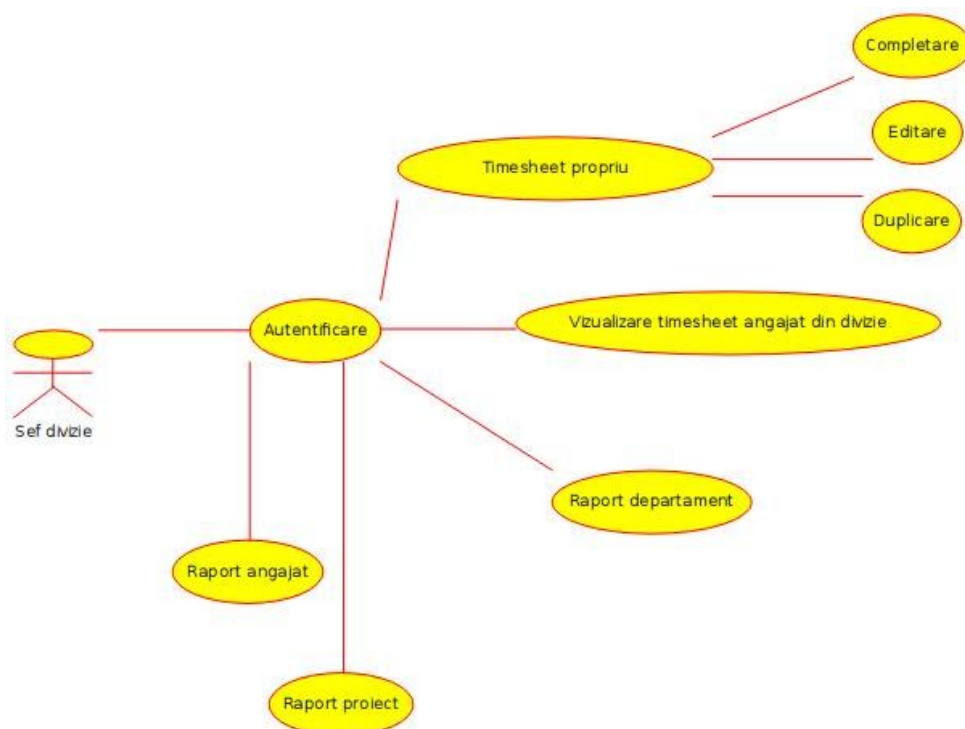
- pentru **șef de departament**:



- completarea fisei lucratoare de la 0 sau duplicarea unei fișe deja salvate în fiecare zi de servicii prin introducerea activităților depuse, atât pentru cele 8 ore de lucru cât și pentru orele extra, dacă este cazul
- aprobarea sau rejectarea, urmată de specificarea motivului, timesheet-ului unui angajat
- citirea timesheet-ului unui angajat din cadrul departamentului său
- introducerea proiectelor și a clienților pentru care lucrează angajații
- citirea rapoartelor cu detalii despre angajați și proiecte

- pentru **șef de divizie:**

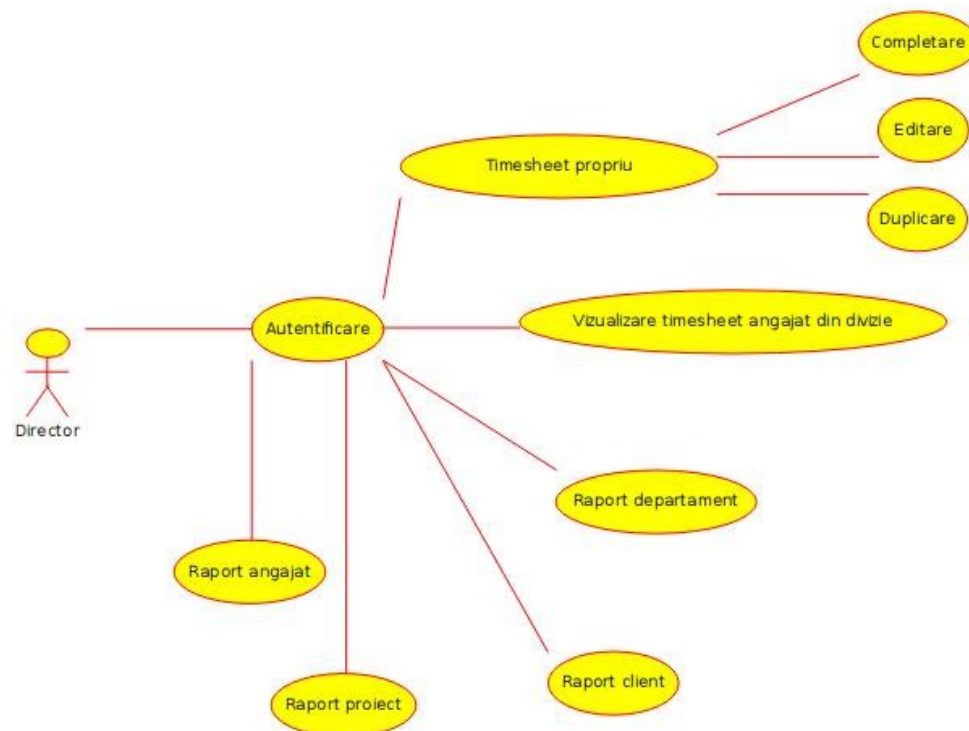
- completarea fisei lucratoare de la 0 sau duplicarea unei fișe deja salvate în fiecare zi de servicii prin introducerea activităților depuse, atât pentru cele 8 ore de lucru cât și pentru orele extra, dacă este cazul
- citirea timesheet-ului unui angajat din cadrul diviziei sale
- citirea rapoartelor cu detalii despre angajați, proiecte și departamente



- pentru **director:**

- completarea fisei lucratoare de la 0 sau duplicarea unei fișe deja salvate în fiecare zi de servicii prin introducerea activităților depuse, atât pentru cele 8 ore de lucru cât și pentru orele extra, dacă este cazul

- citirea timesheet-ului unui angajat din cadrul companiei
- citirea rapoartelor cu detalii despre angajați, departamente, proiecte și client



2.3 Constrângeri majore

Timesheet:

- creare: OPEN și editabil.
- trimitere: SUBMITTED și needitabil
- aprobat: APPROVED și needitabil
- respins: REJECTED (se poate trece în OPEN și editabil)

| USER | DREPTURI |
|--------------------|---|
| Angajat | - Crează, citește și modifică propriul timesheet |
| Șef de departament | - Crează, citește și modifică propriul timesheet - Citește, aprobă/rejectează timesheet-uri pentru angajații din departamentul pe care îl conduce. |
| Șef de divizie | - Crează, citește și modifică propriul timesheet - Citește orice timesheet pentru orice angajat din divizie |
| Director | - Crează, citește și modifică propriul timesheet |

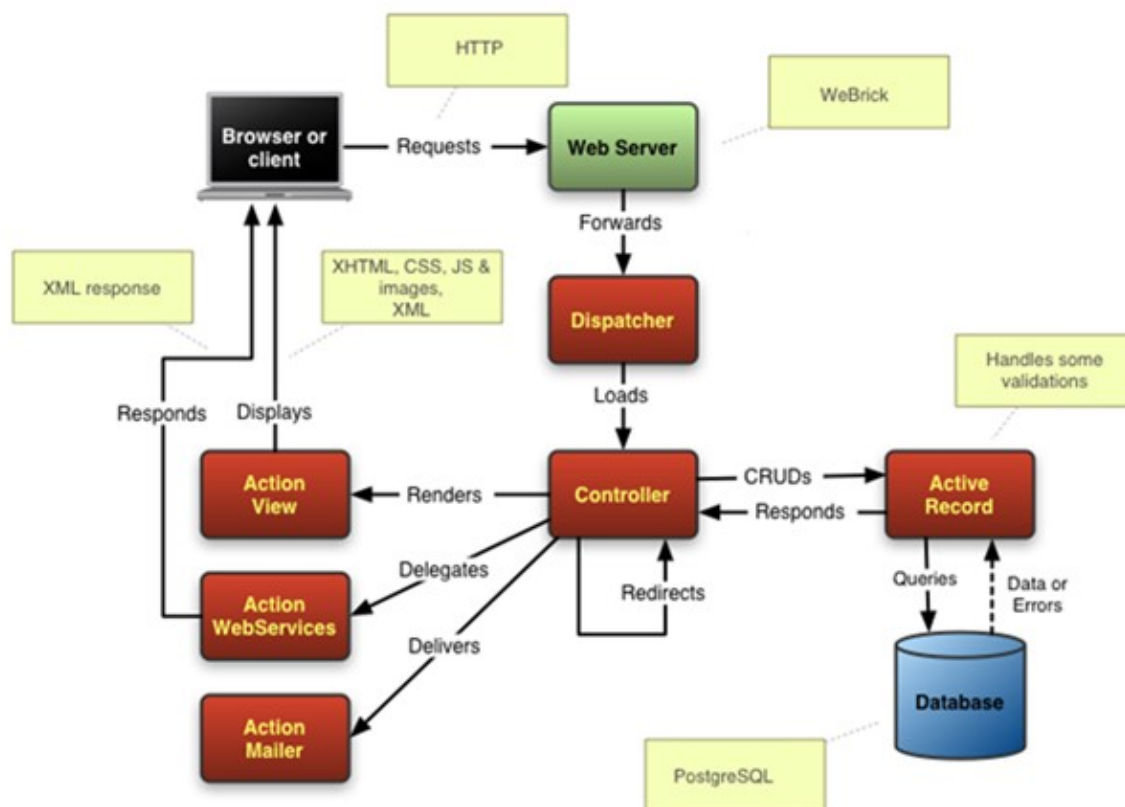
| |
|--|
| - Citește orice timesheet pentru orice angajat din firma |
|--|

2.4 Beneficiu/Valoare

Pe lângă buna organizare a activităților angajaților, aplicația conduce la o bună planificare a proiectelor implementate în cadrul companiei. Ierarhizarea drepturilor de vizualizare a anumitor rapoarte permite observarea și îmbunătățirea eficienței la nivel de proiect, angajat, departament, divizie și companie.

3 Proiectare arhitecturală

3.1 Diagrama arhitecturii pe nivele



3.2 Precizarea tehnologiilor folosite

Back-end-ul aplicației este realizat în framework-ul “Ruby on Rails” folosind limbajul de programare „Ruby”.

Front-end-ul sistemului folosește elemente de HTML5, CSS3 și JavaScript.

Ruby

Versiunea: **2.0**

Website oficial: <http://www.ruby-lang.org/en/documentation/>

Ruby este un limbaj de programare reflectiv, dinamic și orientat pe obiecte.

Motivarea alegerii acestui limbaj de programare stă în faptul că este “frumos”, un limbaj “plin de artă”, și totuși, practic și accesibil. Filozofia din Ruby e ca fiecare bit de informație și de cod să fie considerat un obiect, căruia i se pot da propriile proprietăți și acțiuni. În realitate acest limbaj de programare de tip scripting care să fie mai puternic decât Perl, și mai orientat pe obiecte decât Python. Deasemenea Ruby este “highly portable”: este dezvoltat în principal pe GNU/Linux, dar rulează pe multe variante de UNIX, Mac OS X, Microsoft Windows, Windows Phone, Windows CE, sau pe un SBC de genul, Raspberry Pi.

Ruby, reprezintă în acest web-site, limbajul de programare ce realizează partea de back-end a acestuia, făcând practic posibilă comunicarea între user și aplicație.

Ruby on Rails (RoR/Rails)

Versiunea: **4.0**

Website oficial: <http://rubyonrails.org/>

Ruby on Rails este o platformă de dezvoltare web pentru limbajul de programare Ruby care facilitează construirea de aplicații web într-un mod mai accelerat.

Motivul alegerii acestui mediu de dezvoltare îl constituie faptul că Rails este un framework dedicat aplicațiilor web și include tot ceea ce e necesar pentru crearea unui web site bazat pe o bază de date back-end în conformitate cu șablonul Model-View-Control. O unelă utilă este "scheletarea" care poate construi automat unele din modelele și vizualizările necesare pentru a construi o pagină web simplă. Ruby include WEBrick, un server web simplu, și Rake, un build system. Împreună cu Ruby on Rails, aceste unelte creează un mediu simplu de dezvoltare.

Devise

Versiunea: **3.2**

Website oficial: <https://github.com/plataformatec/devise/>

Devise este o soluție flexibilă de autentificare pentru Ruby on Rails bazată pe Warden.

Acest gem a fost ales din mai multe motive, cum ar fi faptul că este bazat pe Rack, vine ca o soluție MVC peste motoarele Ruby, permite roluri (models) multiple înscrie în același moment și este creat cu o filozofie de modularizare: “use just what you really need”.

Rol în aplicație:

- **Database Authenticatable:** criptează și reține o parolă în baza de date pentru validarea autentificării unui user când acesta dorește să se log-eze. Această autentificare poate fi realizată atât prin metoda POST cât și prin HTTP Basic Authentication;
- **Confirmable:** trimite email-uri de confirmarea instrucțiunilor și verifică dacă un cont a fost deja confirmat în timpul Sign In-ului;
- **Recoverable:** resetază parola unui user și trimite email cu instrucțiunile de reset;
- **Registerable:** manipulează Sign Up-ul user-ilor printr-un process de înregistrare, permițându-le deasemenea să își editeze sau să renunțe la cont;
- **Timeoutable:** termină o sesiune în care nu a existat activitate pentru o anumită perioadă de timp;
- **Validatable:** asigură validarea email-ului și a parolei;

Cancan

Versiunea: 1.6

Website oficial: <https://github.com/ryanb/cancan>

Cancan e o bibliotecă care controlează accesul utilizatorilor la resurse.

Permișiunile sunt stocate într-o singură clasă, modelul “Activity”, fiind utilizate în modele, controllere, view-uri sau helpere.

RoleModel

Versiunea: 0.8

Website oficial: https://github.com/martinrehfeld/role_model

RoleModel e un gem pentru stocarea eficientă și declarativă a rolurilor utilizatorilor.

Rolurile sunt stocate în baza de date sub forma unor bitmask-uri și preluate de modelul User.

Twitter Bootstrap

Versiunea: 2.3.2

Website oficial: <http://getbootstrap.com>

Twitter Bootstrap este o colecție gratuită de instrumente pentru crearea de site-uri web și aplicații web. Acesta conține template-uri de proiectare bazate pe HTML și CSS pentru tipografie, forme, butoane, diagrame, navigație și alte componente de interfață, precum și extensii opționale JavaScript.

Am ales Bootstrap deoarece este un cadru de front-end elegant, intuitiv și puternic pentru dezvoltarea web mai rapidă și mai ușoară, fiind creat să arate bine și să se comporte decent atât pe ultimile browser-e desktop, cât și pe cele de pe tablete și smartphone-uri.

RABL (Ruby API Builder Language)

Versiunea: 0.9

Website oficial: <https://github.com/nesquena/rabl>

Este un gem pentru realizarea șabloanelor pentru diverse formate de schimbare de date. Pot fi utilizate în cadrul aceleiași aplicații formatele: JSON, XML, MessagePack, BSON sau PList.

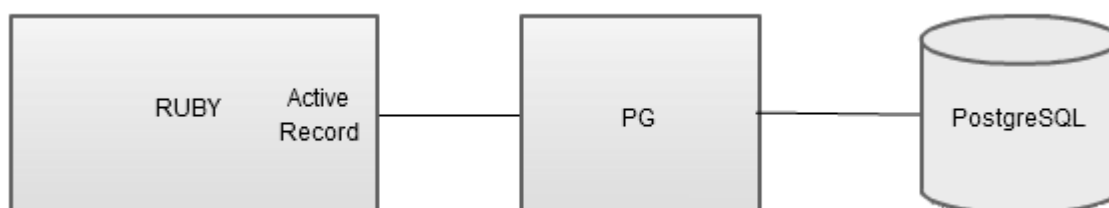
RubyGems - manager de pachete pentru limbajul de programare Ruby, care oferă un format standard pentru distribuirea de programe Ruby și biblioteci (într-un format de sine stătător numit gem - "bijuterie"), un instrument proiectat pentru a gestiona cu ușurință instalarea de gem-uri, și un server pentru distribuirea lor.

PostgreSQL

Versiunea: 9.3

Website oficial: <http://www.postgresql.org/>

În ceea ce privește persistența datelor, am utilizat o bază de date „PostgreSQL” la care am avut acces prin intermediul interfeței standard „Pg”, modul care înlocuiește vechiul modul „Postgres”.



PG

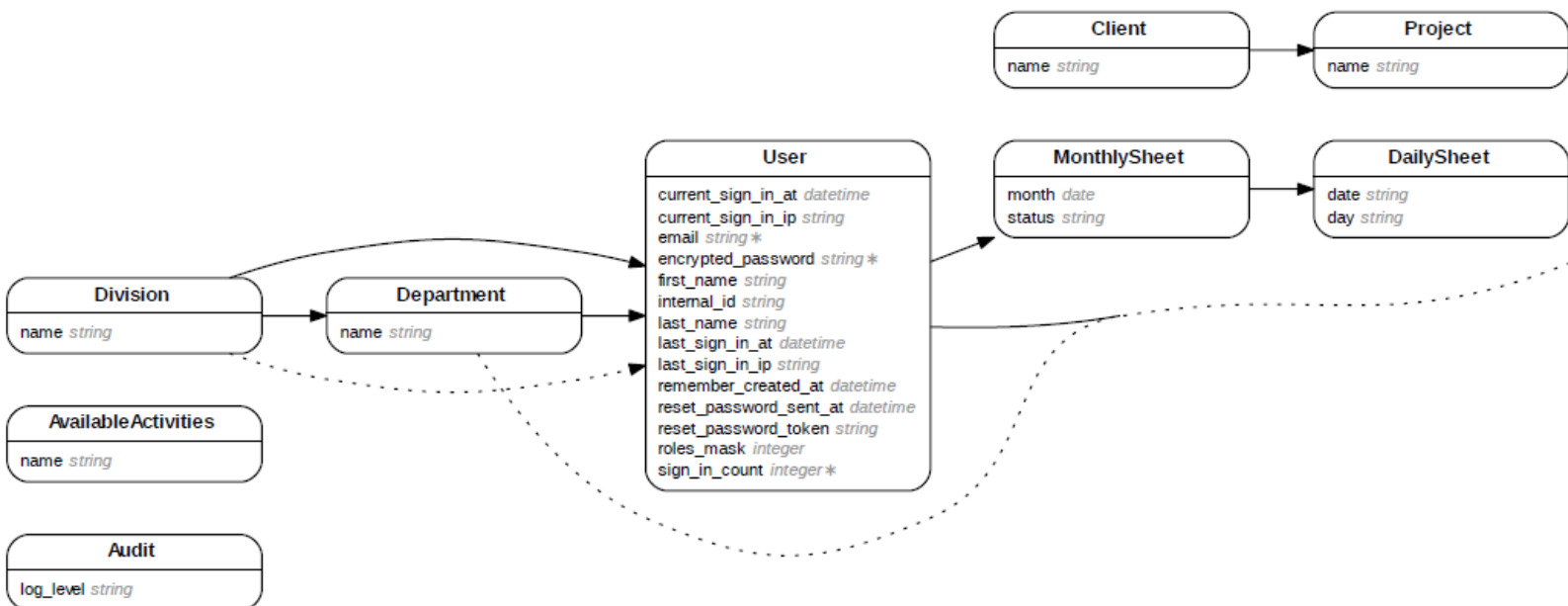
Versiunea: 0.17

Website oficial: <https://bitbucket.org/ged/ruby-pg/wiki/Home>

PG e interfața de comunicare cu PostgreSQL RDBMS.

4 Modelarea datelor

4.1 Diagrama claselor



Model ce stochează datele de autentificare prin coloanele email, encrypted_password, masca rolului utilizatorului prin atributul roles_mask cât și informații despre angajat cum ar fi numele și prenumele (first_name, last_name), identificatorul intern din cadrul firmei (internal_id).

Modelul mai stochează și departamentul și divizia în care se află angajatul.

Relațiile Department - User și Division - User sunt 1 departament/ divizie la mai mulți angajați

Division

Stochează divizia prin numele și idul acesteia.

Între Division și User se stabilește o relație de unu la mai mulți prin intermediul Department.

Department

Stochează departamentul. Atributele: id, name.

AvailableActivities

Stochează activitățile pe care le poate desfășura un angajat.

Audit

Clasă Singleton, tabel cu o singură coloană. Stabilește nivelul la care se poate face auditul.

De exemplu se poate scrie într-un jurnal doar activitățile în cadrul aplicației făcute de șefii de departamente.

Client

Tabelă ce reține clienții.

Project

Tabelă în care se stochează proiectele.

Există o relație de unul la mai mulți între Client și Project.

Task

Stochează activitatea pontată de angajat cu un număr de ore și o descriere.

Relații se găsesc între Task și User, mai multe activități aparțin doar unui singur angajat, dar și între Project sau Client cu Task, un client sau proiect are unul sau mai multe task-uri.

DailySheet

Este fișa zilnică pontată de utilizator.

MonthlySheet

Fișa lunară pontată de utilizator.

Atributul status stochează una din valorile OPEN, SUBMITTED, APPROVED sau REJECTED.

5 Interfața grafică

La logare, utilizatorul va fi trimis în pagina „Contul meu”, care are rolul de pagină principală. Aceasta pagina conține informațiile despre angajatul logat.

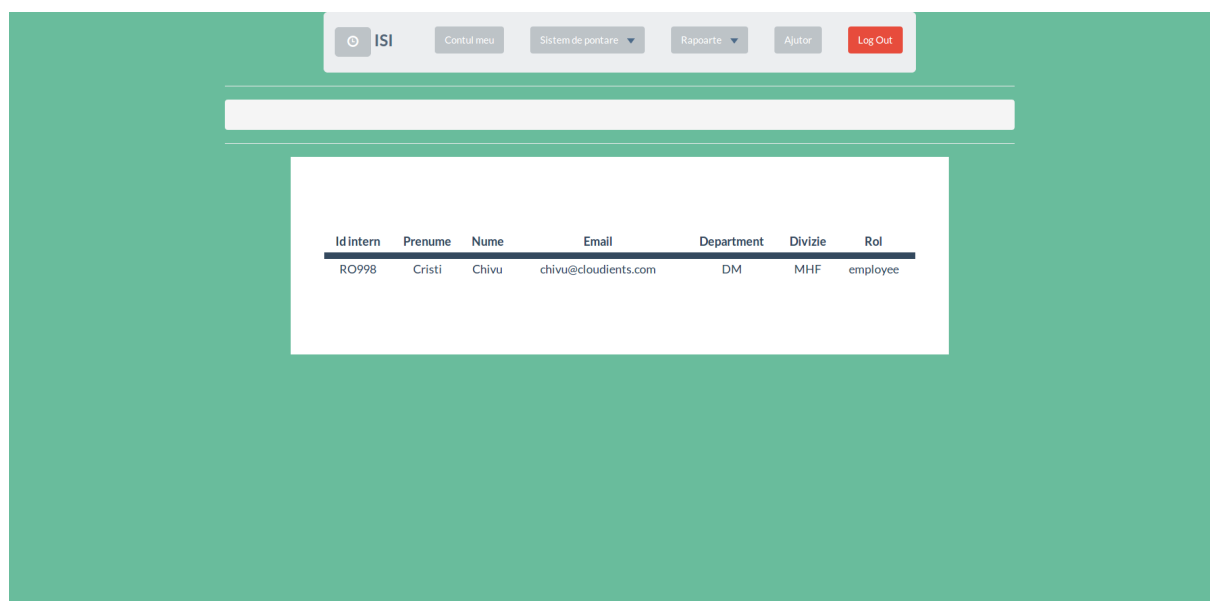


Fig.1 Home page

Acțiunile corespunzătoare timesheet-ului sunt prezente în meniul cascada ivit la apăsarea butonului „Sistem de pontare”.

La completarea unui timesheet, utilizatorul alege ziua pe care o va modifica și apasă butonul „Completează” care îl va redirecța în pagina corespunzătoare zilei alese.

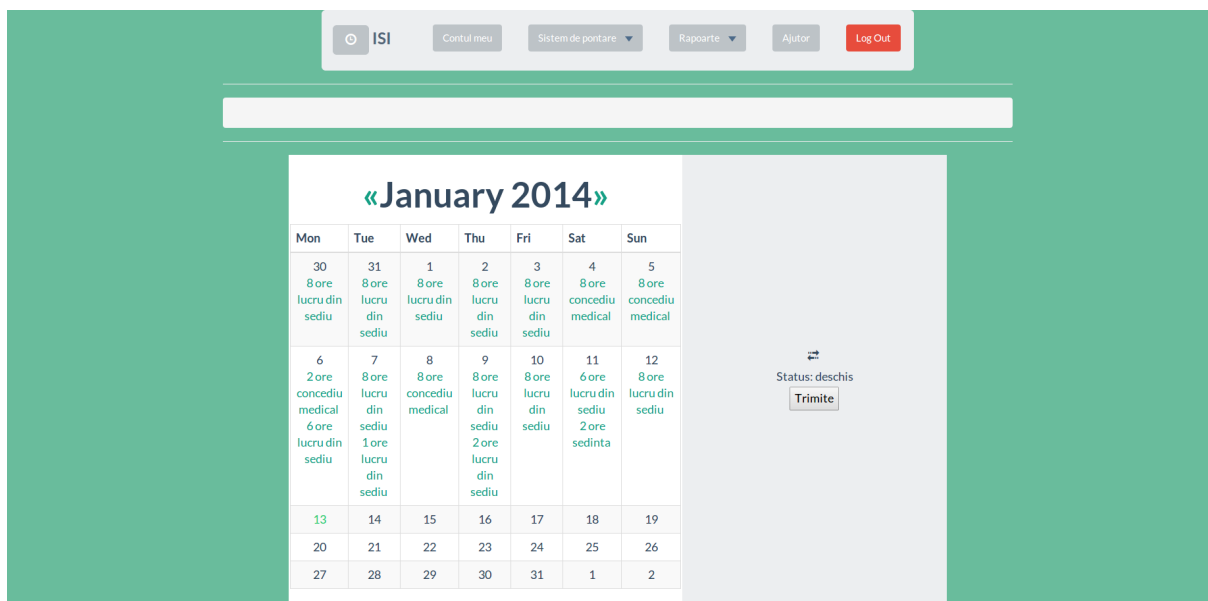


Fig.2 Timesheet angajat

În această pagina, utilizatorul adaugă o activitate și completează informațiile specifice (proiect, client, ore). Are posibilitatea să adauge și ore suplimentare, care vor fi tratate diferit.

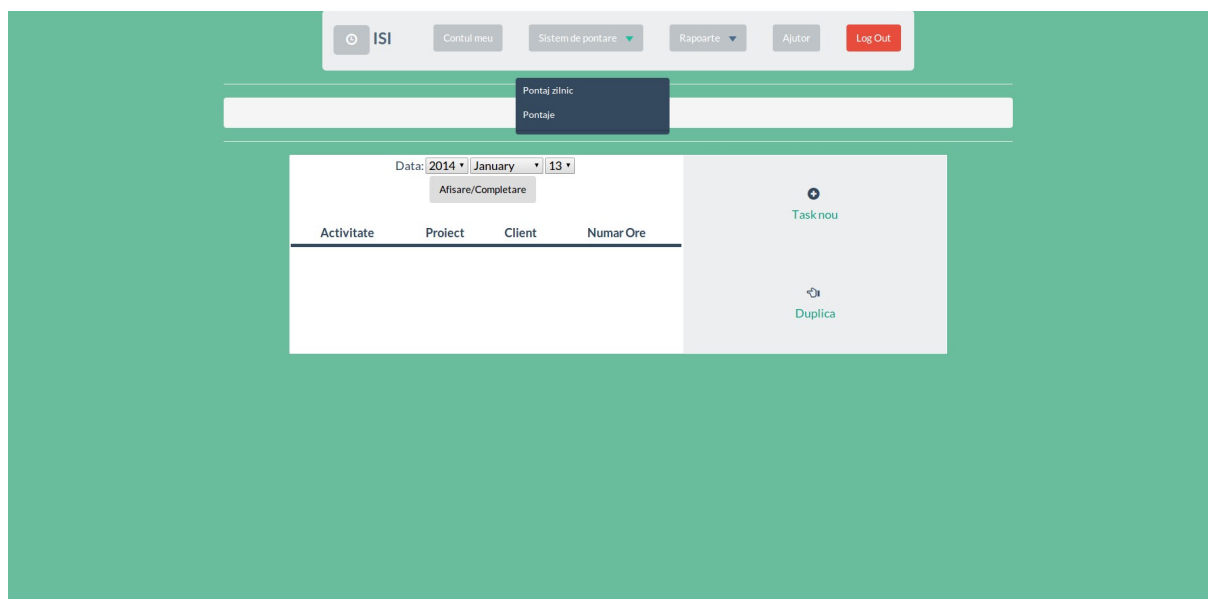
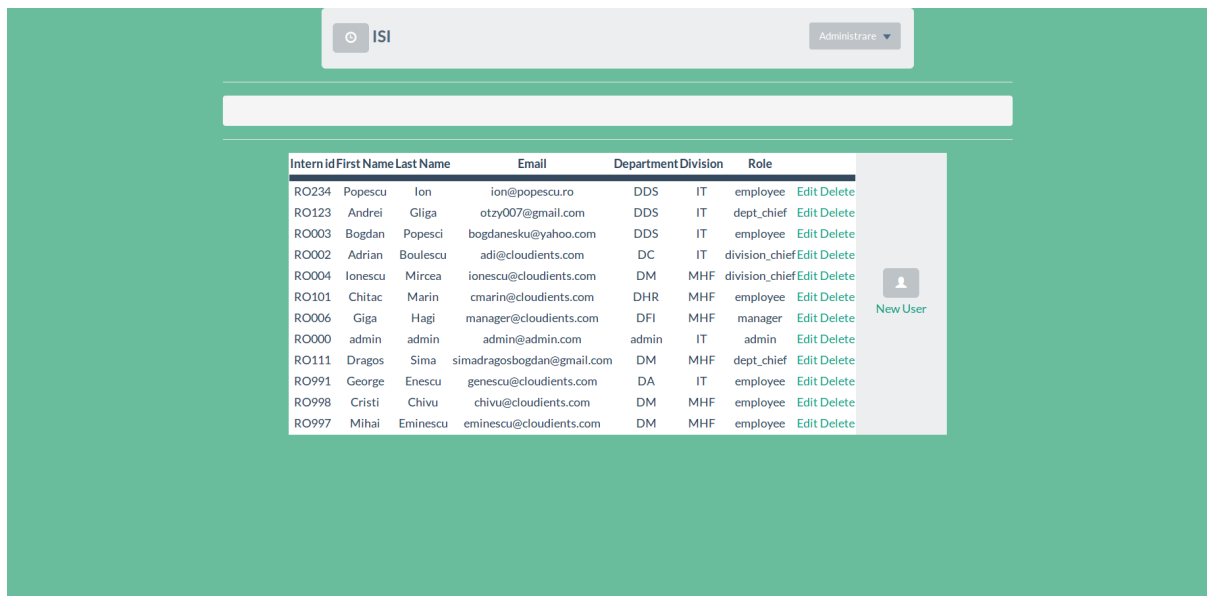


Fig.3 Completare timesheet

Pagina administratorului permite adăugarea, modificarea de angajați, departamente și divizii, precum și configurarea și activarea mecanismului de AUDIT.



| Intern Id | First Name | Last Name | Email | Department | Division | Role | |
|-----------|------------|-----------|----------------------------|------------|----------|----------------|-------------|
| RO234 | Popescu | Ion | ion@popescu.ro | DD5 | IT | employee | Edit Delete |
| RO123 | Andrei | Gliga | otzy007@gmail.com | DD5 | IT | dept_chief | Edit Delete |
| RO003 | Bogdan | Popesci | bogdanesku@yahoo.com | DD5 | IT | employee | Edit Delete |
| RO002 | Adrian | Boulescu | adi@cloudients.com | DC | IT | division_chief | Edit Delete |
| RO004 | Ionescu | Mircea | ionescu@cloudients.com | DM | MHF | division_chief | Edit Delete |
| RO101 | Chitac | Marin | cmarin@cloudients.com | DHR | MHF | employee | Edit Delete |
| RO006 | Giga | Hagi | manager@cloudients.com | DFI | MHF | manager | Edit Delete |
| RO000 | admin | admin | admin@admin.com | admin | IT | admin | Edit Delete |
| RO111 | Dragos | Sima | simadragosbogdan@gmail.com | DM | MHF | dept_chief | Edit Delete |
| RO991 | George | Enescu | genescu@cloudients.com | DA | IT | employee | Edit Delete |
| RO998 | Cristi | Chivu | chivu@cloudients.com | DM | MHF | employee | Edit Delete |
| RO997 | Mihai | Eminescu | eminescu@cloudients.com | DM | MHF | employee | Edit Delete |

New User

Fig.4 Admin page

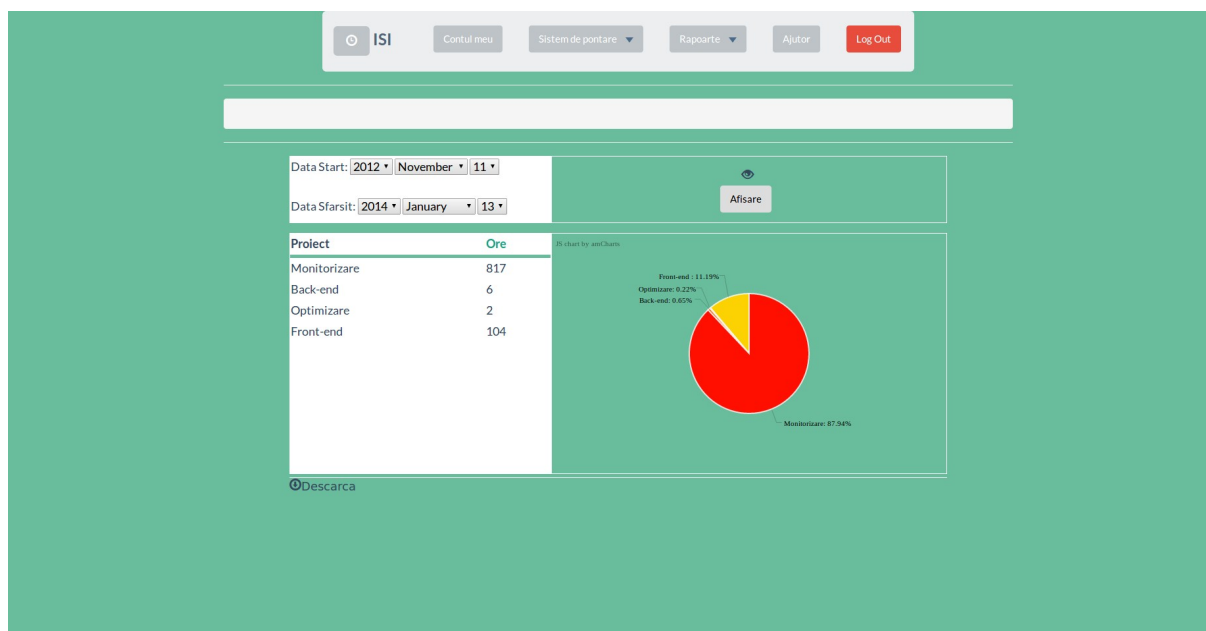


Fig.5 Raport 1

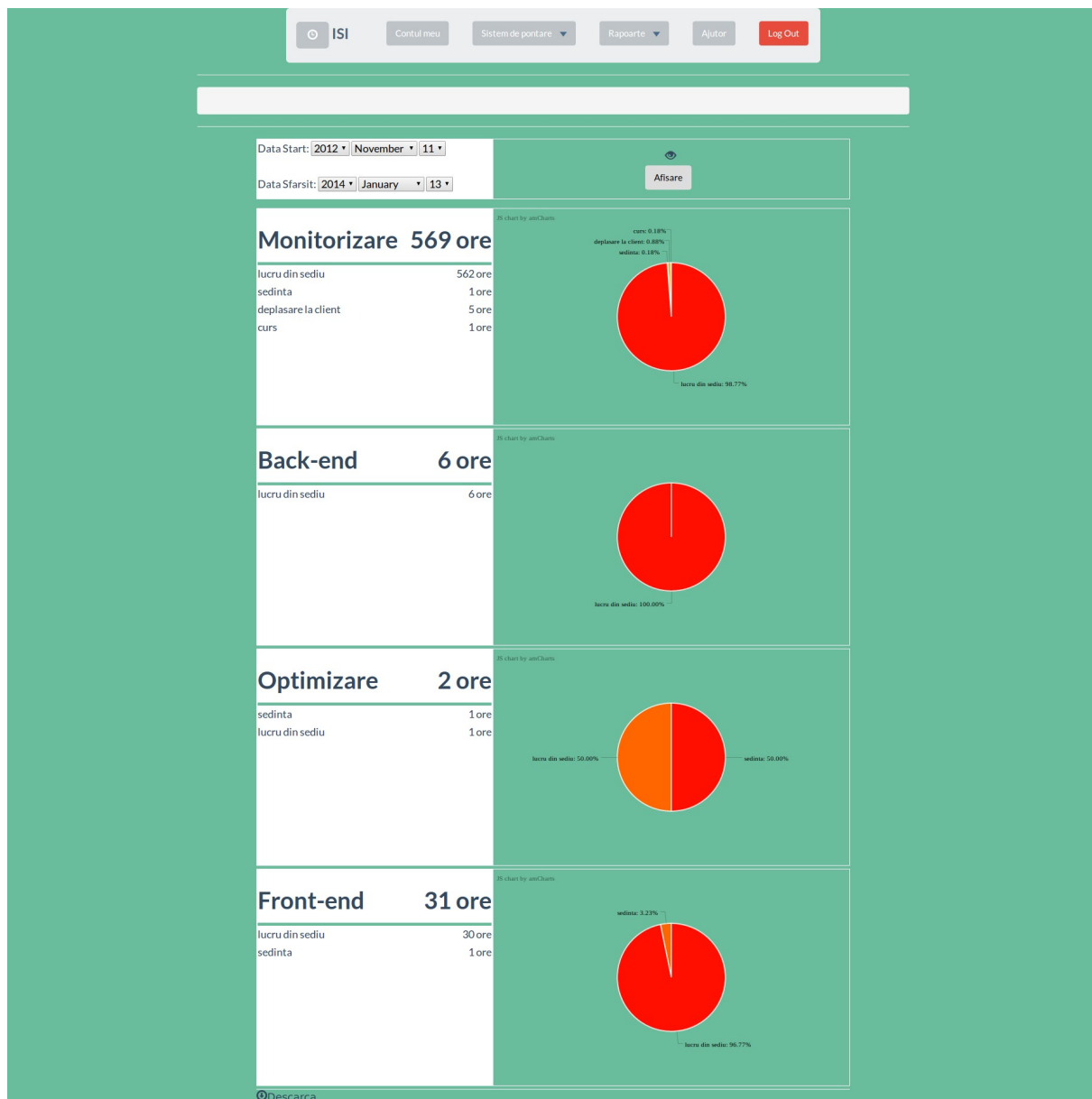


Fig.6 Rapoarte 2

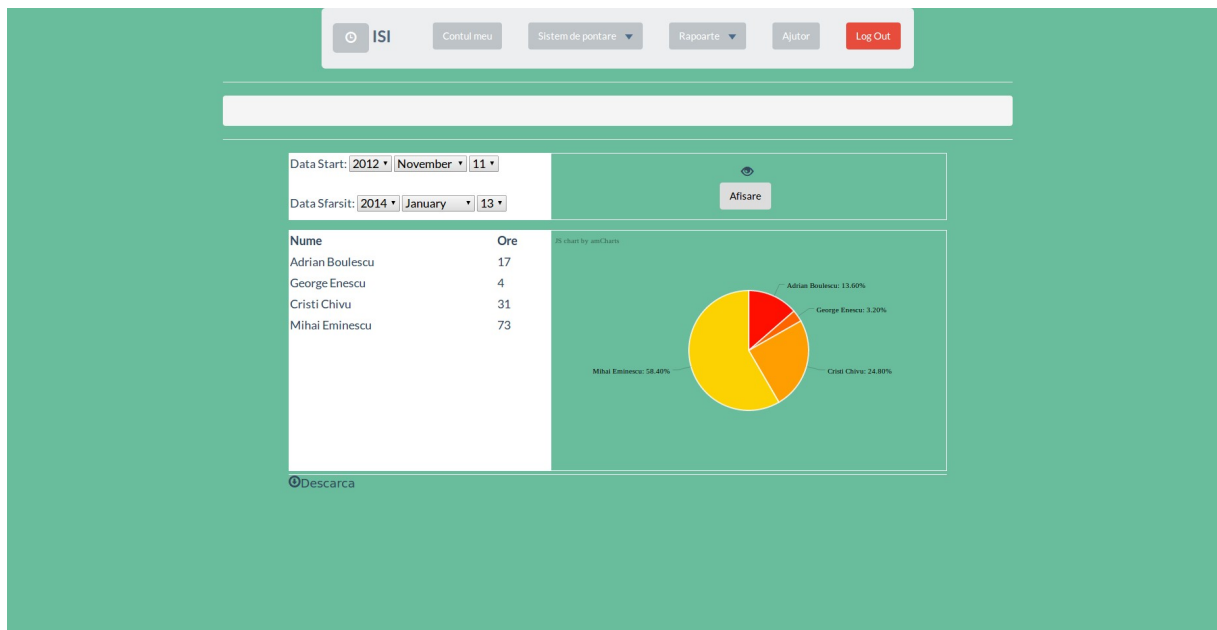


Fig.7 Raport 2