

Document de Proiectare Arhitecturală

Aplicație Timesheet Management

Revizie: 13 Noiembrie 2013

Nume echipă:

- Beta Team

Membri:

- Constantin Șerban-Rădoi 342C5
- Gabriel Ivănică 342C5
- Radu Iacob 342C5

Cuprins

Introducere

Scopul documentului

Scopul proiectului

Cazuri de utilizare

Administrator Aplicație

Specificarea diviziilor/departamentelor

Setare mecanism de audit și nivele de logare

Denumire șefi de departament/divizie și director

Angajat

Autentificare

Completarea unei fișe zilnice

Editare timesheet lunar

Trimitere timesheet spre aprobare

Șef de departament

Aprobă/Respinge timesheet subalterni

Adăugare/Ștergere subalterni

Adăugare proiecte și clienți

Vizualizare raport subalterni

Șef de divizie

Vizualizare rapoarte suplimentare

Director

Vizualizare rapoarte suplimentare

Descrierea arhitecturală pe nivele a aplicației

Tehnologii client-side

JavaScript

HTML

CSS

AngularJS (<http://angularjs.org/>)

Bootstrap.js (<http://twitter.github.io/bootstrap/>)

jQuery (<http://jquery.com/>)

Tehnologii server side

node.js (<http://nodejs.org/>)

socket.io (<http://socket.io/>)

Tehnologii bază de date

PostgreSQL (<http://www.postgresql.org/>)

Diagrame de clase

Diagrame de componente

Diagrame de activitate

Structura bazei de date

Diagrame de deployment

Prototipuri

Login page

Home page

Timesheets page

1. Introducere

1.1. Scopul documentului

Documentul își propune să prezinte pe scurt structura de ansamblu a proiectului, împreună cu o serie de decizii legate de modul în care este construit. În continuare, putem regăsi o suită de diagrame de utilizare, de componente și altele.

1.2. Scopul proiectului

Aplicația reprezintă un mod electronic al gestionării unui timesheet pentru o anumită companie. Aceasta include o gamă variată de utilizatori, atât angajați obișnuiți, ce au drepturi limitate de utilizare, cât și manageri, respectiv administratori de sistem.

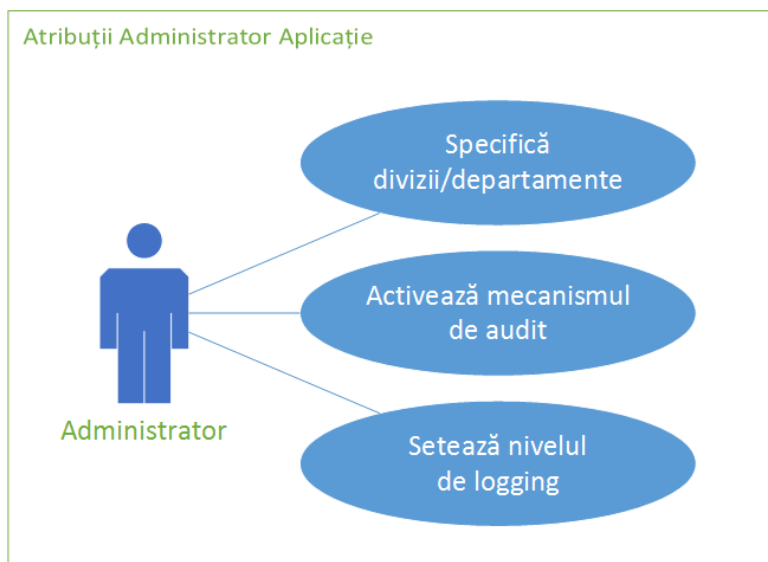
Prin acest sistem se dorește ca fiecare angajat să își poată înregistra activitățile pe care le desfășoară în cadrul firmei și posibilitatea urmăririi mai ușoare de către manageri a tot ceea ce se întâmplă.

2. Cazuri de utilizare

2.1. Administrator Aplicație

2.1.1. Specificarea diviziilor/departamentelor

1. Administratorul accesează pagina de configurare a companiei
2. O lista cu diviziile din cadrul companiei este afișată
3. Administratorul poate adăuga noi divizii sau șterge unele vechi
4. Selectând o divizie, se va afișa o listă cu departamentele din cadrul acelei divizii
5. Administratorul poate adăuga sau șterge departamente din cadrul unei divizii



2.1.2. Setare mecanism de audit și nivele de logare

1. Din interfața de configurare a companiei se selectează pagina de configurare audit
2. Se va afișa o listă cu rolurile din cadrul companiei și nivelele de logare pentru fiecare
3. Administratorul poate selecta nivelul de logare independent pentru fiecare rol

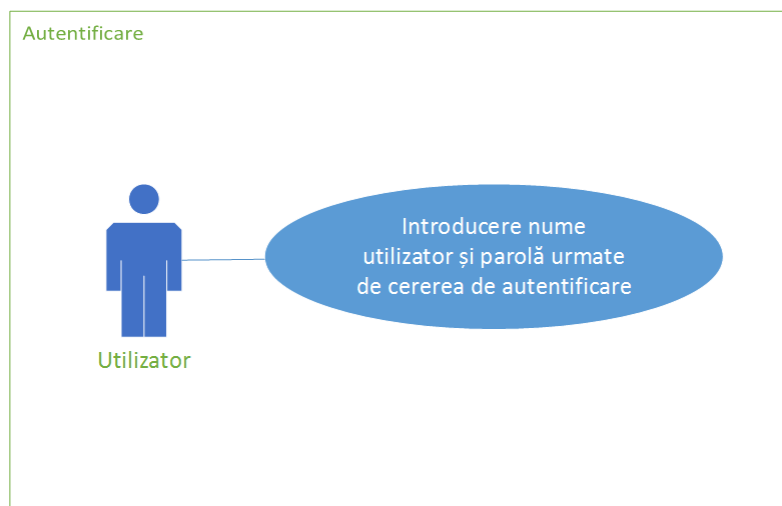
2.1.3. Denumire șefi de departament/divizie și director

1. Din interfața de configurare a companiei se selectează pagina de configurare a departamentelor/diviziilor
2. Se afișează lista cu departamente/divizii și șeful corespondent fiecăreia
3. Administratorul poate modifica șeful de divizie/departament selectând din listă un alt angajat

2.2. Angajat

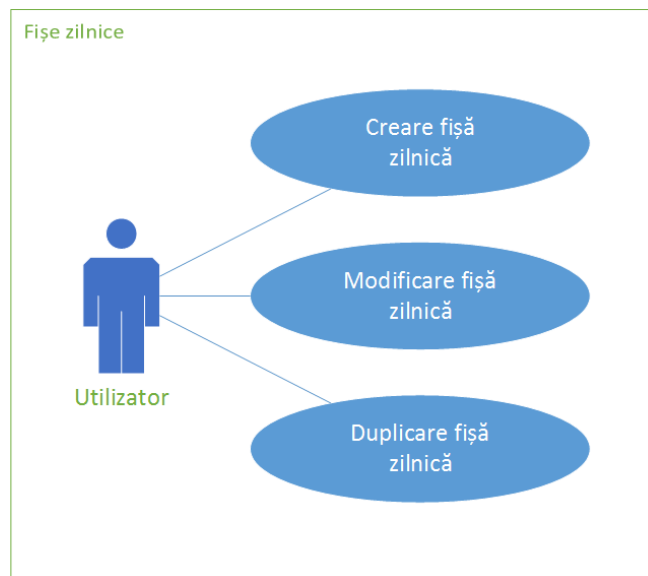
2.2.1. Autentificare

1. Fereastra inițială la deschiderea aplicației este fereastra de login
2. Utilizatorul introduce numele de utilizator/email-ul și parola și apasă butonul de logare
3. Datele de autentificare sunt trimise către server și se așteaptă un răspuns
 - a. Dacă autentificarea s-a realizat cu succes utilizatorul este redirecționat către pagina principală a aplicației
 - b. Dacă autentificarea nu s-a realizat cu succes, un mesaj de eroare este afișat



2.2.2. Completarea unei fișe zilnice

1. Creare fișei zilnice
 - a. Din pagina principală angajatul alege să creeze o nouă fișă zilnică
 - b. Se poate selecta o altă fișă zilnică de la care să se pornească
2. Angajatul poate adăuga sau șterge activități din cadrul fișei zilnice
3. După salvare, fișa zilnică trece în starea "SUBMITTED"

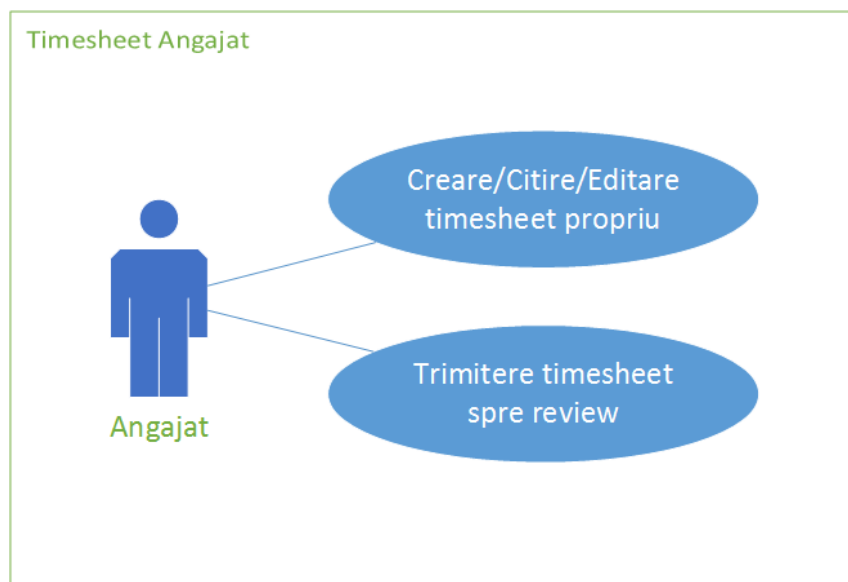


2.2.3. Editare timesheet lunar

1. Angajatul accesează meniul de timesheet lunar
2. Se va afișa o listă cu ultimele timesheet-uri lunare ale angajatului
3. Angajatul selectează un timesheet pentru editare
 - a. Timesheet-ul lunar curent
 - b. Un timesheet având status "REJECTED" ce va trece în starea "OPEN"
4. Angajatul poate adăuga un comentariu la timesheet

2.2.4. Trimitere timesheet spre aprobare

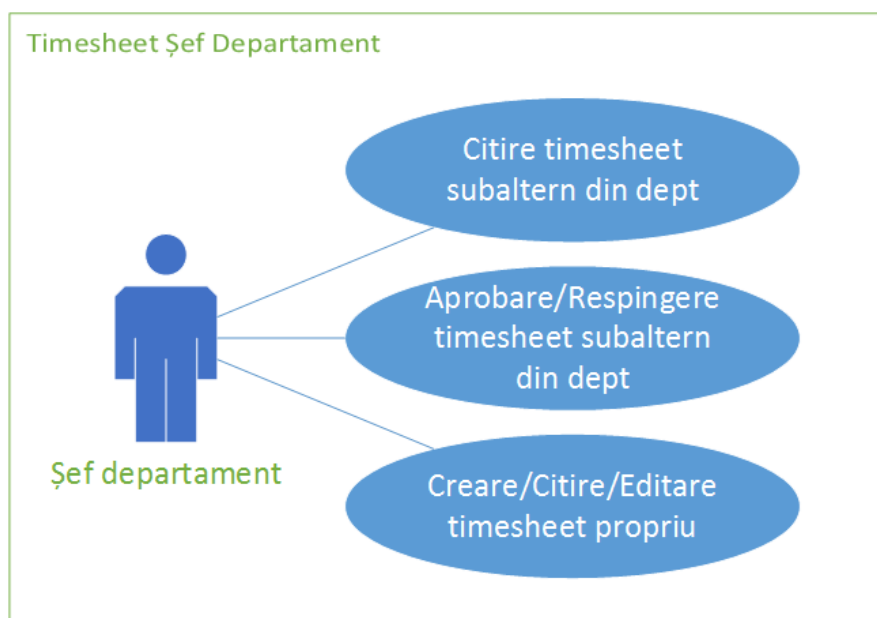
1. La sfârșitul lunii angajatul poate trimite un timesheet spre aprobare
2. Odată trimis, timesheet-ul trece în starea "SUBMITTED" și nu mai poate fi editat



2.3. Șef de departament

2.3.1. Aprobă/Respinge timesheet subalterni

1. Accesare pagină unde sunt listate timesheet-urile asociate fiecărui subaltern.
2. Utilizatorul poate să selecteze un timesheet din lista celor aflate în starea “SUBMITTED”.
3. În continuare utilizatorul poate să aprobe sau să respingă timesheet-ul selectat (caz în care acesta trece în starea “APPROVED”, respectiv “REJECTED”). În cazul respingerii, subalternul căruia îi aparține timesheet-ul este notificat.

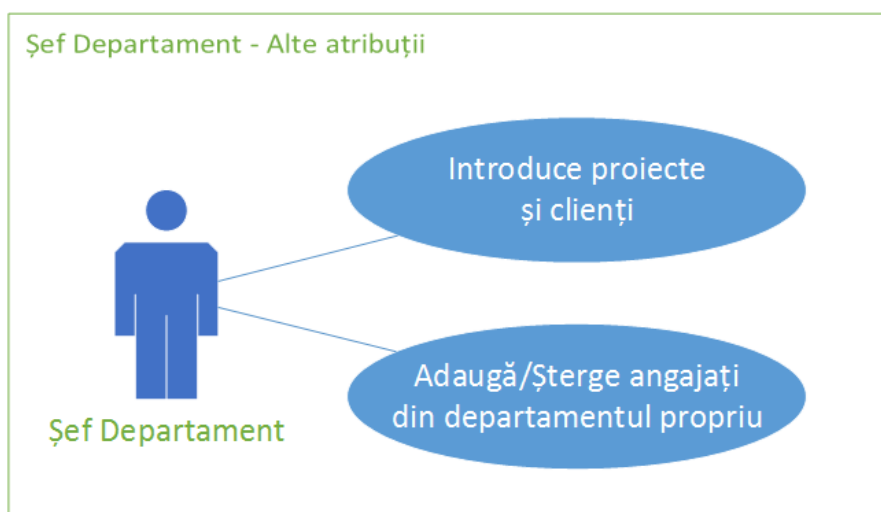


2.3.2. Adăugare/Ștergere subalterni

1. Utilizatorul accesează pagina dedicată managementului de resurse umane, unde sunt listați toți angajații din departament.
2. Utilizatorul poate introduce datele corespunzătoare unui angajat nou (sau aparținând altui departament) pentru adăugare în listă, respectiv poate selecta un angajat pentru a fi șters din listă.

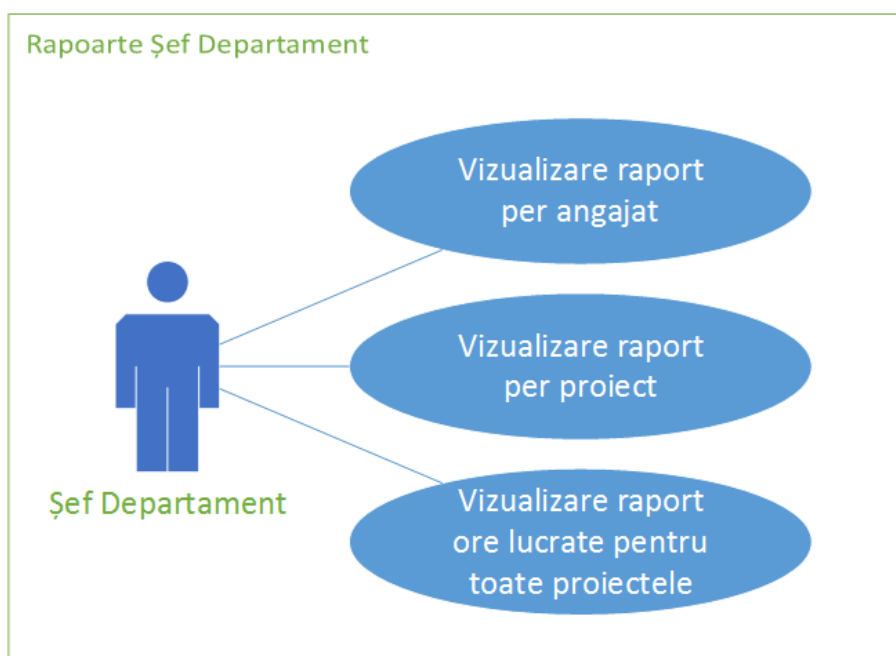
2.3.3. Adăugare proiecte și clienți

1. Utilizatorul introduce datele aferente unui nou proiect (nume/descriere) precum și clientul asociat, în pagina dedicată managementului proiectelor.



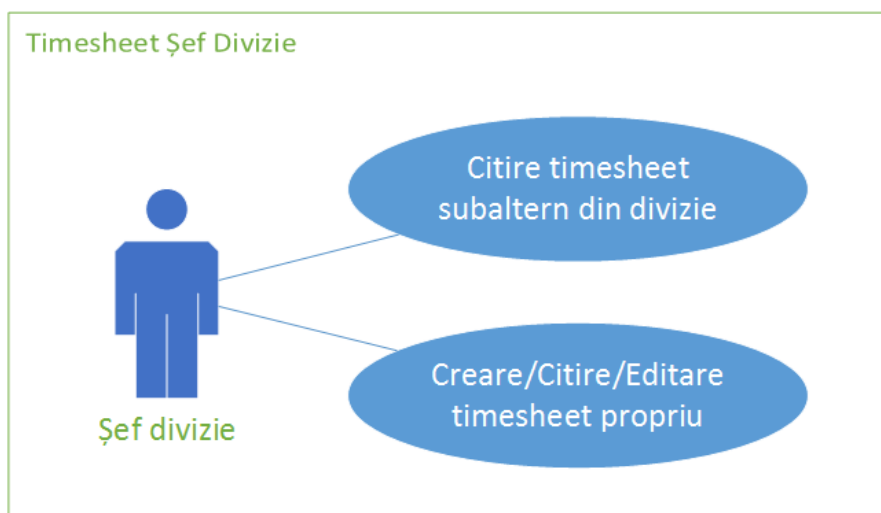
2.3.4. Vizualizare raport subalterni

1. Utilizatorul accesează o pagină dedicată care oferă posibilitatea realizării de rapoarte automate.
2. Selectare dintr-o lista tipul raportului dorit:
 - a. Numărul total de ore lucrate de angajat pentru fiecare proiect în parte. (este necesar ca utilizatorul să introducă ID-ul angajatului și intervalul dorit)
 - b. ID-urile persoanelor care au lucrat la un anumit proiect și numărul total de ore. (este necesar ca utilizatorul să introducă numele proiectului și perioada).
 - c. Numărul total de ore lucrate pentru fiecare proiect în parte (este necesar ca utilizatorul să limiteze rezultatul la o perioadă de timp).



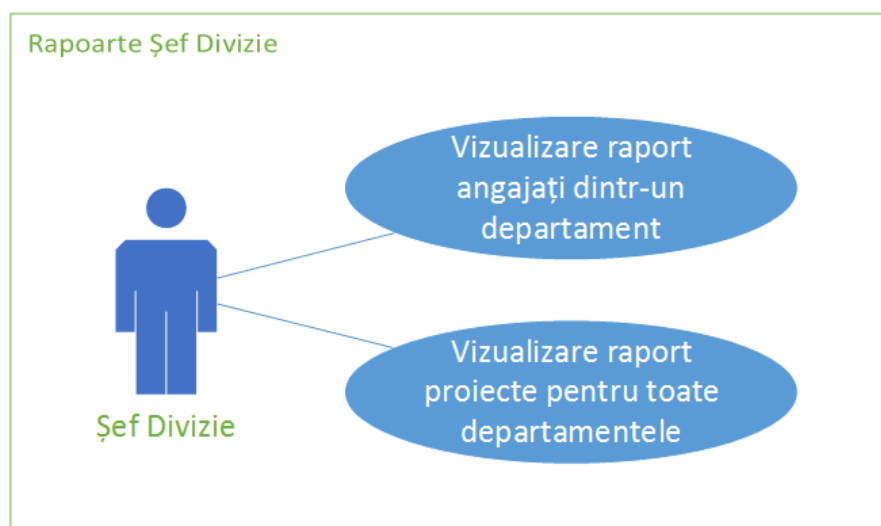
2.4. Șef de divizie

Șeful de divizie are disponibile aceleași opțiuni ca șefii de departament. În plus față de aceștia are posibilitatea vizualizării de rapoarte care agregă informații aferente departamentelor subordonate. De asemenea poate vizualiza orice timesheet pentru orice angajat din divizia asociata.



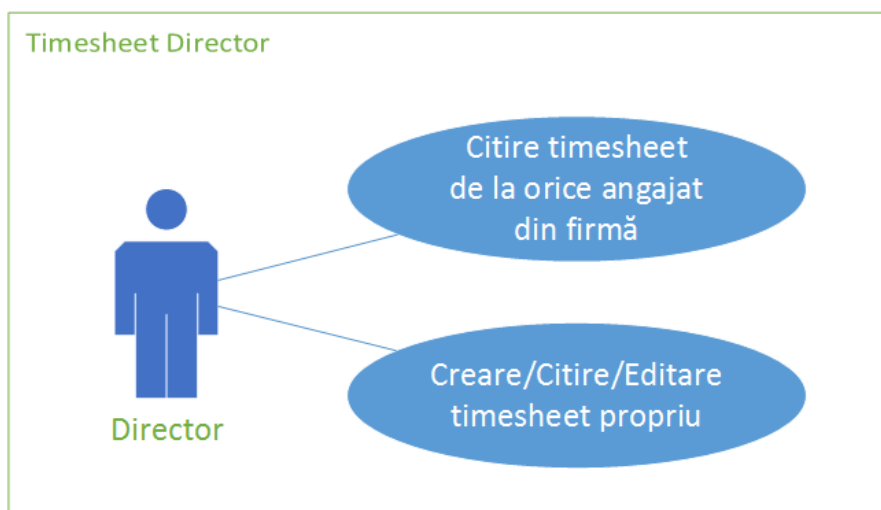
2.4.1. Vizualizare rapoarte suplimentare

1. Utilizatorul accesează o pagină dedicată care oferă posibilitatea vizualizării de rapoarte automate.
2. Selectare dintr-o listă a tipului raportului dorit:
 - a. Proiectele în lucru și numărul total de ore lucrate în cadrul respectivelor proiecte, pentru fiecare departament subordonat.(este necesar ca utilizatorul să limiteze rezultatul la o perioadă de timp)
 - b. Lista de angajați pentru fiecare departament subordonat.



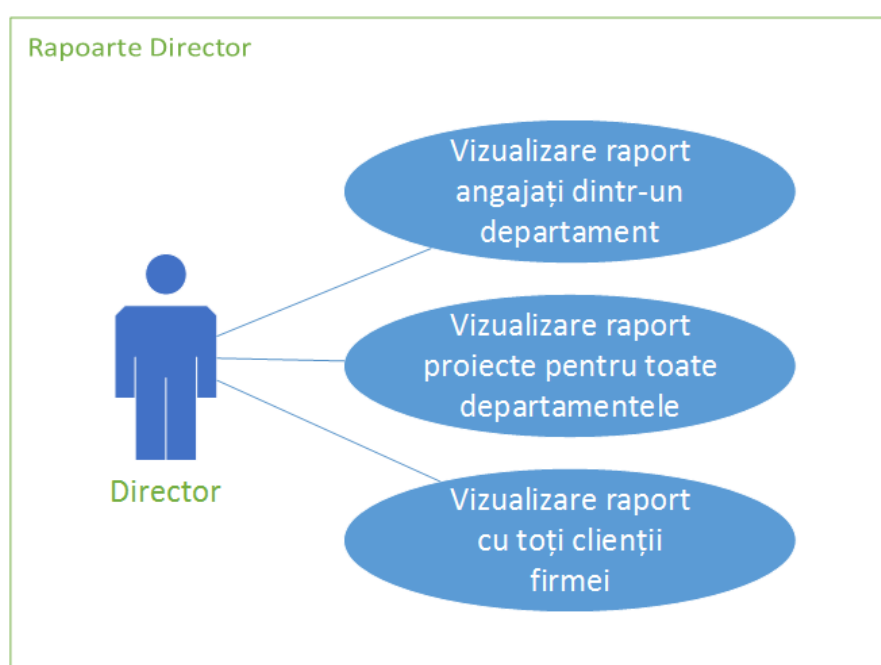
2.5. Director

Directorul are disponibile aceleași opțiuni ca șefii de divizie. În plus, are posibilitatea vizualizării de rapoarte care agregă informații aferente tuturor diviziilor din companie. De asemenea poate vizualiza orice timesheet pentru orice angajat din companie.

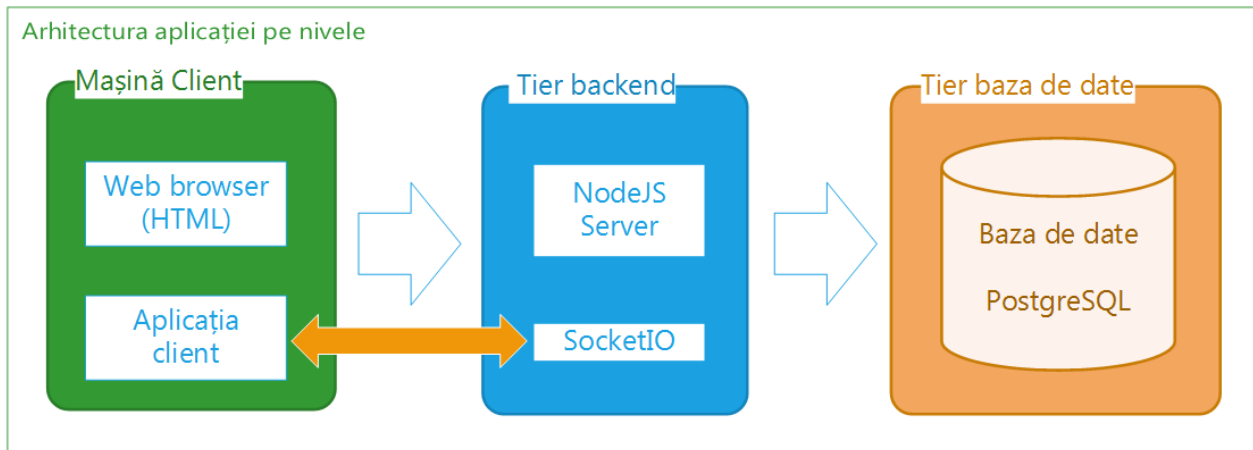


2.5.1. Vizualizare rapoarte suplimentare

1. Din pagina dedicată, directorul are posibilitatea de a vizualiza un raport care să cuprindă:
 - a. toți clienții firmei, proiectele lor, numărul de ore și procentul din timp dedicat de către angajați pentru proiectele respective (este necesar ca utilizatorul să limiteze rezultatul la o perioadă de timp)
 - b. angajații unui departament
 - c. lista de proiecte pentru toate departamentele



3. Descrierea arhitecturală pe nivele a aplicației



Aplicația este concepută după modelul client-server, având la nivelul clientului aplicația ce se accesează dintr-o pagină web, ce interacționează cu backend-ul prin intermediul socheților, care la rândul său interacționează cu o bază de date, unde sunt stocate informațiile despre utilizatori și alte date utile din aplicație.

3.1. Tehnologii client-side

3.1.1. JavaScript

JavaScript este un limbaj de scripting utilizat în mediul Web și suportat nativ de orice Browser. Fiind o aplicație web-based este utilizat JavaScript.

3.1.2. HTML

Limbaj hipertext de marcare interpretat de browser și utilizat pentru descrierea interfeței unei pagini web.

3.1.3. CSS

CSS este un limbaj utilizat pentru a stiliza și organiza elementele HTML în pagină.

3.1.4. AngularJS (<http://angularjs.org/>)

AngularJs este un framework client-side folosit pentru aplicații web single page. Oferă un sistem de templating și posibilitatea utilizării asemenea unui framework model-view-controller.



Am ales AngularJs deoarece oferă o serie de funcționalități necesare platformei noastre: templating, actualizare interfeței web prin utilizarea controlerelor și posibilitatea de a crea un single-page app.

3.1.5. Bootstrap.js (<http://twitter.github.io/bootstrap/>)

Bootstrap.js este un framework front-end folosit pentru dezvoltarea rapidă a paginilor web. Oferă un set extins de elemente pentru interfața cu utilizatorul, teme cât și posibilitatea utilizării unui model responsive design pentru realizarea de pagini cu model dinamic în funcție de rezoluția utilizată.



Am ales să folosim Bootstrap.js datorită modelului rapid de dezvoltare al interfețelor cu utilizatorul.

3.1.6. jQuery (<http://jquery.com/>)

jQuery este o bibliotecă JavaScript foarte puternică ce oferă posibilitatea de a scrie cod JavaScript cross-platform foarte eficient.



Am ales să utilizăm jQuery atât datorită faptului că oferă o implementare high-level a celor mai utilizate funcționalități JavaScript, simplificând foarte mult codul scris, cât și datorită necesității de compatibilitate cu majoritatea browserelor disponibile.

3.2. Tehnologii server size

3.2.1. node.js (<http://nodejs.org/>)



Node.js este o platformă construită pe baza motorului JavaScript V8 ce oferă posibilitatea de a crea servere web rapide, scalabile și eficiente pentru comunicații real-time. Am ales utilizarea platformei Node.js deoarece oferă toate aspectele necesare aplicației web: server JavaScript, comunicare real-time prin socketi.

3.2.2. socket.io (<http://socket.io/>)

O bibliotecă ce oferă implementarea de websockets pentru aplicațiile web ce necesită comunicare în timp real cu server. Utilizăm socket.io pentru comunicare real-time între aplicație și server.

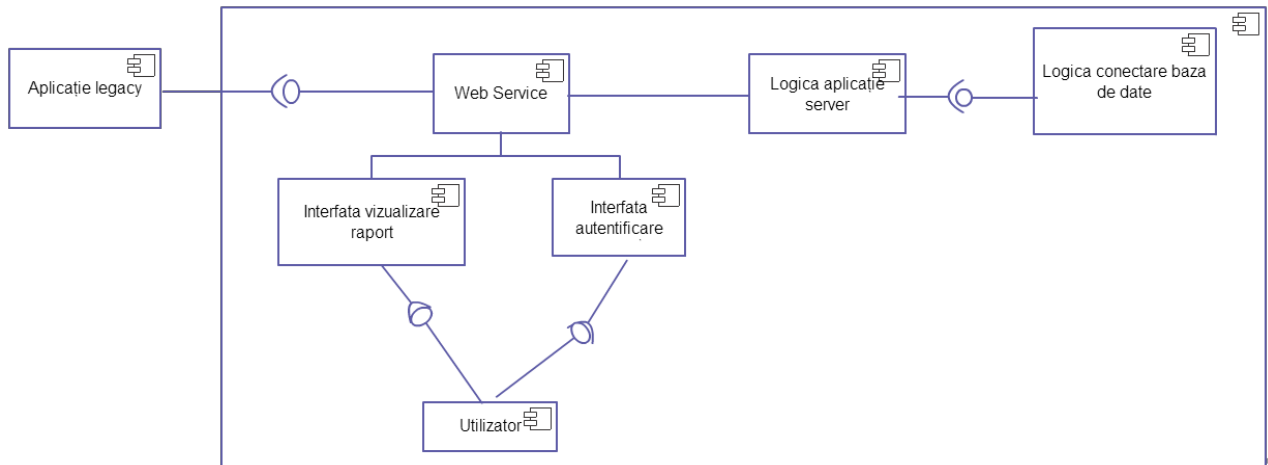
3.3. Tehnologii bază de date

3.3.1. PostgreSQL (<http://www.postgresql.org/>)

PostgreSQL este un sistem open-source de gestiune a bazelor de date relaționale ce oferă performanțe foarte bune.



3.4. Diagrame de componente



3.5. Diagrama de clase

JavaScript este un limbaj de programare ce nu ofera support pentru definirea claselor. Se pot defini insa obiecte cu support pentru single inheritance. Modelul de mostenire nu este asemanator cu limbajele Java / C / C++ ci un model de mostenire bazat pe prototyping.

Aplicatia este sturcturata pe obiecte astfel:

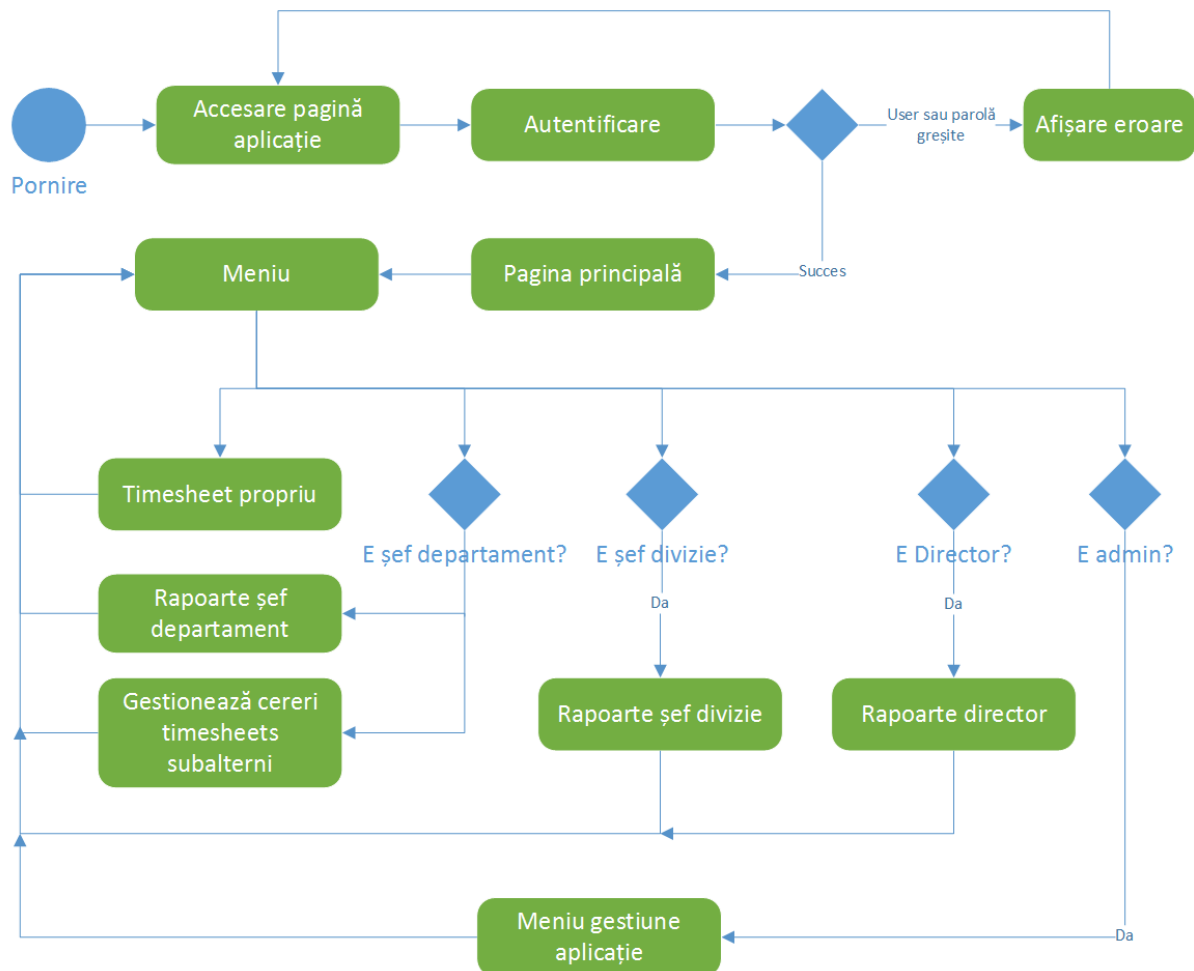
Pe partea de client side vom utiliza un obiect definirea controllere-lor ce se ocupa cu gestionarea actiunilor utilizatorului, trimiterea si receptionarea mesajelor de la server, parsarea si verificarea datelor introduse de utilizator dar si redirectionarea catre alte pagini din aplicatie.

Pe partea server side folosim un obiect ce mentine definirea metodelor de interactiune cu baza de date postgresQL. Pentru aceasta utilizand o biblioteca ORM din node.js ce permite definirea de metode getter/setter, posibilitatea de a crea join-uri cat si custom queries.

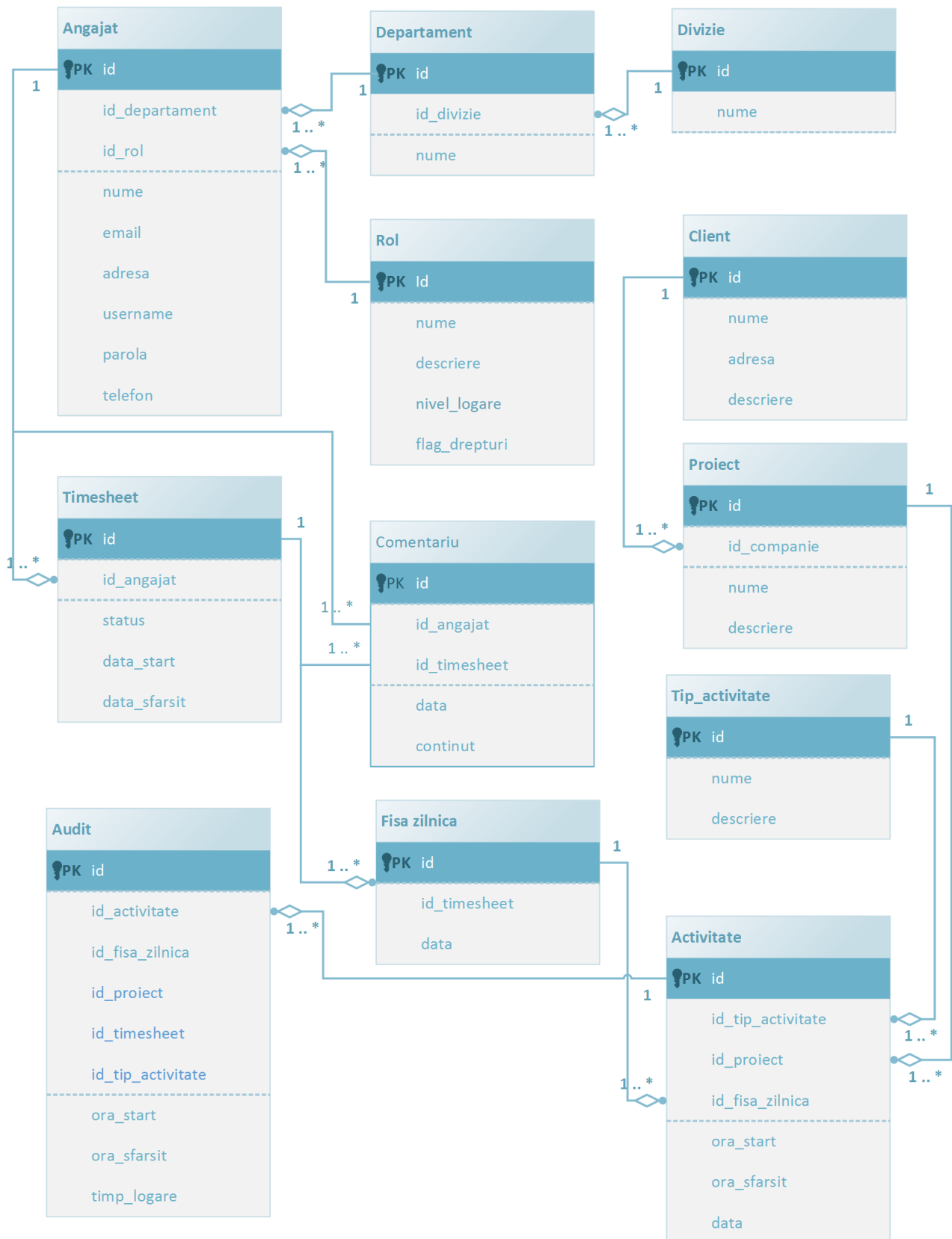
3.6. Diagrame de activitate

În această diagramă este prezentat modul de interacțiune cu aplicația, pornind de la pagina de autentificare, de unde utilizatorul poate ajunge într-un meniu principal dacă s-a autentificat cu succes, respectiv i se afișează o eroare în cazul în care autentificarea a eșuat și este întors înapoi la primul pas.

Mai departe, meniul principal este afișat în funcție de ce tip de utilizator este autentificat în pagină, orice utilizator putând accesa pagina pentru gestionare a timesheet-ului său, însă doar șefii pot accesa paginile aferente generării de rapoarte, respectiv aprobării timesheet-urilor subalternilor. Administratorul aplicației este singurul ce poate intra în meniul de gestiune a aplicației. Din fiecare dintre meniuri, se poate reveni în meniul principal.



4. Structura bazei de date



Baza de date este construită în PostgreSQL și conține o serie de tabele, ce au fost reprezentate în diagrama de mai sus. Principala tabelă este reprezentată de entitatea Angajat. Acesta face parte dintr-un anumit Departament, care la rândul său face parte dintr-o Divizie.

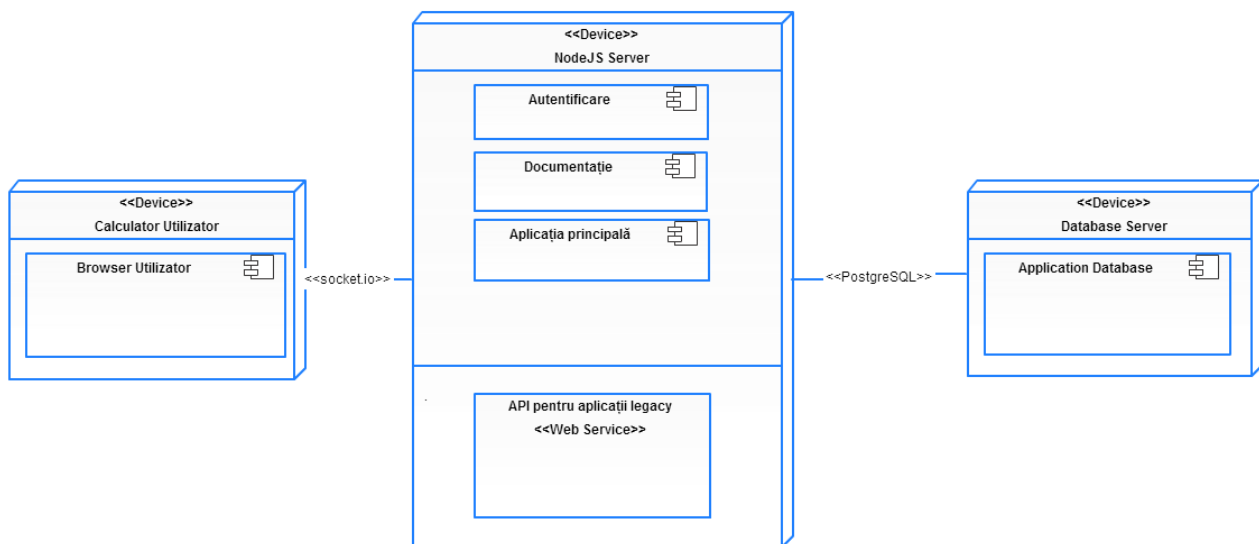
De asemenea, un angajat are un Rol, care înglobează anumite drepturi. În tabela cu Roluri se memorează astfel dacă un angajat este Șef de Departament, sau Șef de Divizie, sau Director, sau Administrator, căruia i se dau anumite drepturi, prin intermediul câmpului flag_drepturi. Aceste drepturi constau în posibilități de a crea anumite rapoarte, sau de a administra cereri de aprobare timesheet, etc, pe care le au doar anumite categorii de utilizatori.

În continuare, un angajat are unul sau mai multe timesheet-uri, la care poate primi sau da comentarii. Fiecare Timesheet înglobează o serie de Fișe zilnice, care la rândul lor sunt compuse din mai multe activități. Activitățile sunt de un anumit tip (predefinit în tabela Tip_activitate) și fac parte din cadrul unui proiect, care este asociat unui client (Companie).

Tabela de activități este replicată în tabela de Audit, ce asigură partea de logare.

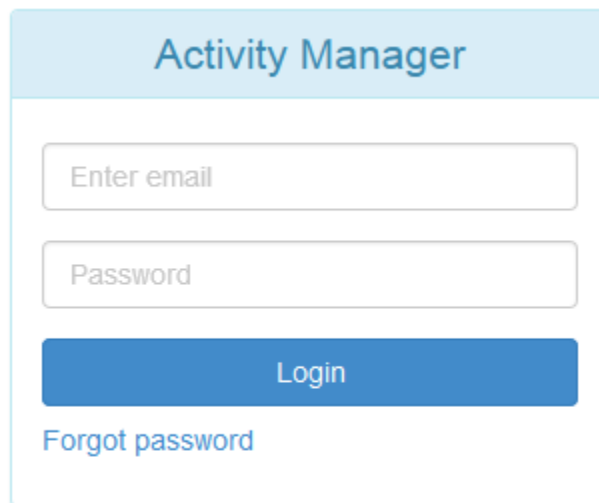
5. Diagrame de deployment

Deployment Diagram pentru aplicație Timesheet Management



6. Prototipuri

6.1. Login page



The login form is titled "Activity Manager" in a light blue header. It contains two input fields: "Enter email" and "Password". Below these is a blue "Login" button and a link "Forgot password" in blue text.

Activity Manager

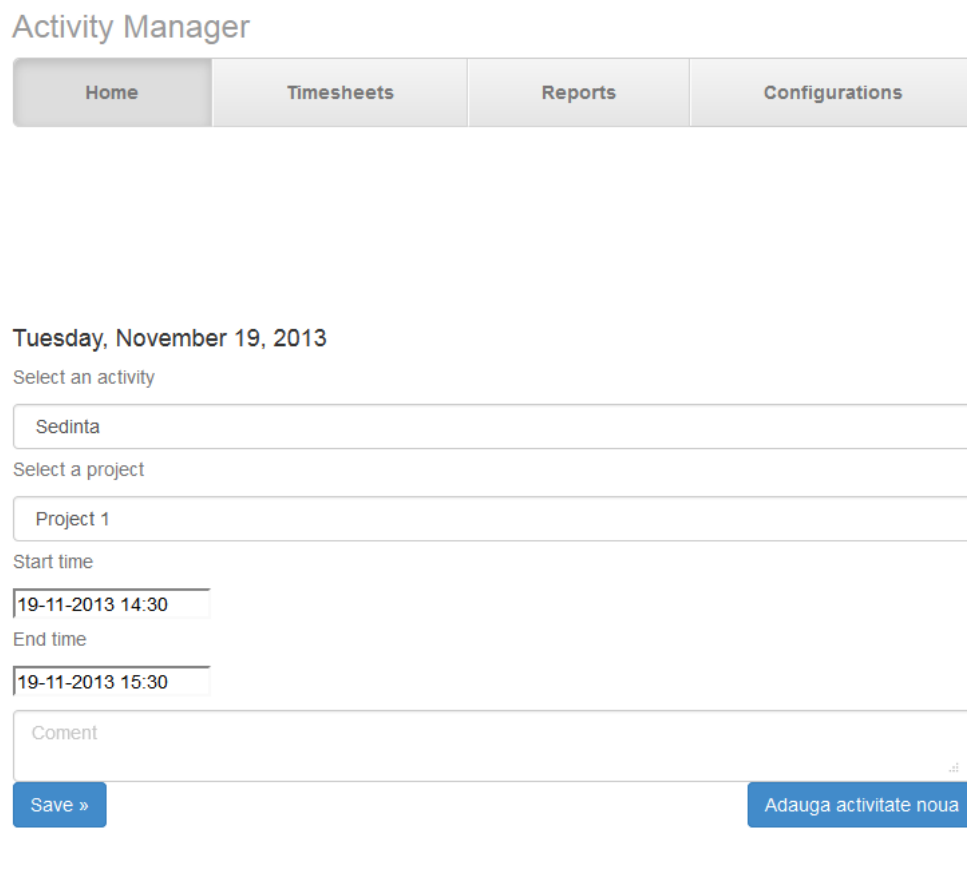
Enter email

Password

Login

[Forgot password](#)

6.2. Home page



The home page has a header "Activity Manager" and a navigation bar with four tabs: "Home", "Timesheets", "Reports", and "Configurations". The "Home" tab is active. Below the navigation bar, the date "Tuesday, November 19, 2013" is displayed. The main form area includes a "Select an activity" dropdown with "Sedinta" selected, a "Select a project" dropdown with "Project 1" selected, a "Start time" field with "19-11-2013 14:30", an "End time" field with "19-11-2013 15:30", and a "Coment" text area. At the bottom, there are two buttons: "Save »" and "Adauga activitate noua".

Activity Manager

Home Timesheets Reports Configurations

Tuesday, November 19, 2013

Select an activity

Sedinta

Select a project

Project 1

Start time

19-11-2013 14:30

End time

19-11-2013 15:30

Coment

Save »

Adauga activitate noua

6.3. Configuration page

Activity Manager

Home	Timesheets	Reports	Configurations
------	------------	---------	----------------

Company organization

Departments

Add new department

Delete selected department

Sef de departament

Divisions

Add new division

Delete selected division

Sef de divizie

Change log level

User role

Log level

© Company 2013