PROIECT - INGINERIA SISTEMELOR SOFTWARE

AN UNIVERSITAR 2019-2020

SEMESTRUL 2

Aplicatie

Monitorizare Angajati

**Nume si Prenume Student Coordonator:**

Mag Raul-Ciprian Prof. Sima Ioan

224/2

Informatica

# Prezentarea cerintei

Organizarea intr-o firma cu un numar mare de angajati devine problematica pe masura ce numarul de angajati creste si se ajunge la pierderi de timp greseli si ineficienta.

Aplicatia dezvoltata de noi, aduce in prim-plan un sistem de monitorizare complex a angajatilor unei companii prin legarea admin sistem-seful firmei -angajat la baza de date a firmei pentru a facilita procesul e atribuire a sarcinilor catre angajati

O firma si-a creat o infrastructura prin care seful monitorizeaza angajatii prezenti la lucru si le traseaza sarcini individuale. Firma are o aplicatie care ofera:

• o fereastra pentru sef, cu ajutorul careia seful vede lista angajatilor prezenti în firma, un element din lista precizând numele angajatului si ora la care s-a logat în sistem. De asemenea, seful poate transmite o sarcina unui angajat prezent astfel: selecteaza angajatul din lista, introduce o descriere a sarcinii si declanseaza un buton "transmite sarcina". Imediat dupa transmiterea unei sarcini, aceasta poate fi consultata de catre angajatul respectiv.

• câte o fereastra pentru fiecare angajat: Atunci când angajatul vine la serviciu, introduce ora sosirii si declanseaza un buton "prezent". Imediat dupa declansarea butonului, seful vede în lista lui ca angajatul este prezent. În continuare, cât timp angajatul sta la serviciu, el primeste si, în consecinta, vede în fereastra lui, sarcinile transmise de sef. La plecare, angajatul închide fereastra, moment în care seful este notificat de delogarea acestuia din sistem.

• o fereastra pentru a ministrator acesta are o lista ce contine angajatii firmei ce au drept de acces in aplicatie, el poate sa adauge un angajat nou pentru a oferii acces la aplicatie, poate sa stearga conturi si sa le modifice.

Angajatii isi pot recupera parola in cazul in care au uitat-o prin actionarea butonului specific in fereastra de logare pentru generarea unei parole noi

# FUNCTIONALITATILE

**Administrator:**

* Se logheaza cu un cont special conceput pentru el, care vine odata cu aplicatia
* El se ocupa cu managementul angajatilor din firma, adaugarea de angajati noi, adica sa le ofere acces la aplicatie, sa retraga accesul unei persoane ,sa ii modifice datele personale si de logare.
* El vede o lista cu toti angajatii inregistrati in aplicatie, si are 3 butonae, adaugare angajat nou, stergere angajat selectat, si modificare angajat selectat.
* La adaugare si modificare, sistemul afiseaza o fereastra noua unde introduce datele persoanei.

**Sef:**

* Sef ul are un cont special care vine cu aplicatia, el vede o lista cu angajatii prezenti in firma, o lista cu sarcinile trimise de el.
* Cand un angajat se logheaza in aplicatie, acesta va aparea in lista sefului.
* Poate sa transmita o sarcina noua, selectand un angajat, si apasand pe butonul de transmitere sarcina, dupa care el trebuie sa introduca un titlu si o descriere pentru sarcina.
* In tabelul cu sarcini ele sunt impartite in doua categorii, verde, sarcini rezolvate deja, si fundal galben, sarcini care nu sunt rezolvate inca.
* In tabelul de sarcini, in dreptul fiecarei sarcini sunt 3 butoane:butonul “i” arata sefului ce feedback a lasat angajatul pentru sarcina, un buton de delete is un buton de edit, unde seful poate sa modifice o sarcina trimisa deja.

**Angajat:**

* Angajatul se logheaza iar daca este o logare cu succes, i se cere sa introduca ora la care a venit la munca, dupa care poate sa vada ce sarcini are de indeplinit, si ce sarcini a indeplinit pana in prezent.
* Daca a uitat parola, are un buton de recuperare date pe fereastra de logare, el introduce username ul si apasa buonul, moment in care o sa ii apara o parola cu care se poate loga.
* Are o lista cu sarcinile sale, in dreptul fiecare sarcini are 3 butoane, butonul de info , “i”, in care vede descrierea sarcinii, buton de finalizare a sarcninii, si buton de adaugare de feedback.
* Cand se delogheaza sau inchide fereastra, el dispare din lista cu angajati a sefului.
* Cand adauga feedback, seful este notificat, iar el este notificat daca seful a sters o sarcina sau a modificat ceva la o sarcina.

Etapele proiectarii:

**Analiza cerintelor:**

* Am indentificat cerintele impuse, functionalitatiile(un admi adauga angajati in aplicatie, ii sterge, ii modifica, un sef vede angajtii prezenti in firma si le da sarcini, angajatii vad ce sarcini au si le indeplinesc) si constrangerile( seful vede doar angajatii care sunt prezenti, un angajat vede doar sarcinile lui).

**Analiza:**

* *Cerintele functionale:* Administratorul trebuie sa poata adauga angajati, sterge angajati si modifica, seful trebuie sa poata vedea angajatii prezenti in firma si sa trimita sarcini, angajatul trebuie sa isi poata vedea sarcinile sale, si sa le consemneze indeplinite.
* *Cerinte nefunctionale:* Aplicatia trebuie sa fie scrisa in Java.

Aplicatia trebuie sa fie usor de utilizat de orice tip de utilizator.

**Proiectarea sistemului:**

* Am folosit draw.io pentru a dezvolta diagrame pentru a ne usura munca. Am dezvoltat diagrama de utilizare, diagrama de stare, diagrama de activitate, diagrama de comunicare/colaborare, diagrama de arhitectura, diagrama bazei de date, diagrama de clase.

**Object Design:**

* Etapa de specificare a constrangerilor.

**Implementarea:**

* Am inceput prin a face baza de date, dupa interfetele si apoi functionalitatiile aplicatiei.

Tehnologii folosite:

**Baza de date:**

Am folosit PostgreSQL ca si baza de date, deoarece este foarte simplu de folosit, foarte rapida si suntem familiarizati cel mai bine cu pentru ca am folosit-o semestrul acesta mai des si si semestrul trecut.

**Mediul de dezvoltare:**

Am ales sa folosimIntelliJ IDEA Ultimate. L-am ales pentru usurinta codarii, auto-genereaza codul si pentru multitudinea de framework-uri care pot fi utilizate.

**Limbaj de programare:**

Limbajul de programare ales de mine, este Java pentru cateva motive simple. Are o comunitate mare de utilizatori care sprijina dezvoltarea lui, un mare ajutor la nelamurirele melepe care le-am intampinat pe parcursul proiectului. Este un limbaj nativ, si usor de gestionat. Functioneaza pe platformele de operare.

**ORM:**

Object / Relational Mapping (ORM) este o tehnica de programare ce face posibila accesarea și manipularea obiectelor fara ca programatorii sa fie interesati de sursa de date de unde provin aceste obiecte. Aceasta tehnica a aparut din nevoia de a depasi diferentele de paradigma dintre modelul orientat pe obiecte si modelul relational.

Am folosit Hibernate, pentru usurinta sa, fara a implementa manual baza de date.

**Proiectarea diagramelor:**

Am folosit draw.io la proiectarea diagramelor pentru usurinta si multitudinea de avantaje ale programului.

**Source control:**

Am folosit GitHub-ul pentru a urmarii schimbarile aplicatiei pe parcursul dezvoltarii ei. Am creeat cate un branch pentru fiecare iteratie.

**Distribuirea task-urilor:**

Am facut la inceput o impartire a functionalitatilor pe iteratii pentru a avea un plan de opertiune bine pus la punct.

**GUI:**

Am folosit JavaFX pentru usurinta de folosire si combinarea cu CSS-ul pentru un design mai placut. Scene Builder e foarte comod de implementat interfete.

**Comunicare Client-Server**:

Structura proiectului este organizata pe niveluri: Model, Persistence, Service, Client.

*Model:* contine declaratii de entitati pentru controlul bazei de date.

*Persistence:* folosim repository care permit interogarea bazei de date folosind Session Factory de la Hibernate.

*Service:* contine toate interfetele folosite in aplicatie, si toata partea de logica a aplicatiei.Aici vin toate semnalele trimise de interfetele utilizator, sunt procesate, si trimse raspunsuri inapoi la interfete.

*Client:* contine toate interfetele vizibile si controller-urile acestoare, care cer informatii la server.

# Scenariu de utilizare

**Administrator:**

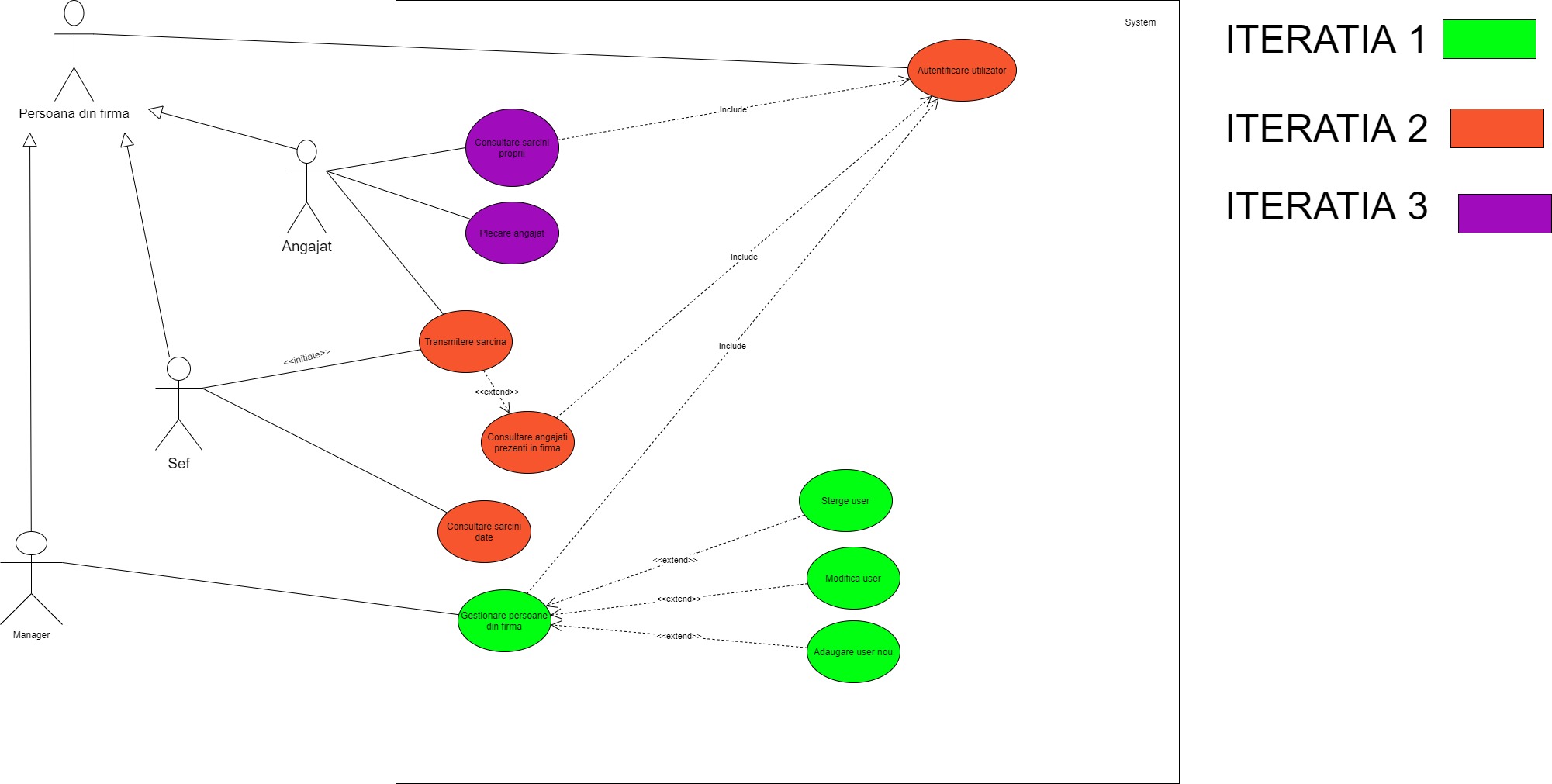
* Deschide aplicatia, se logheaza cu un cont special pentru el, are o lista angajatii inregistrati in aplicatie. Poate sa adauge un angajat nou, prin apasarea butonului de add, moment in care se deschide o fereastra noua in care introduce datele angajatului. Selecteaza un angajat si apasa pe butonul de stergere pentru a sterge un angajat din aplicatie, si butonul de modificare, la care ii apare din nou o fereastra cu datele angajatului selectat unde poate sa modifice datele despre el.

**Sef:**

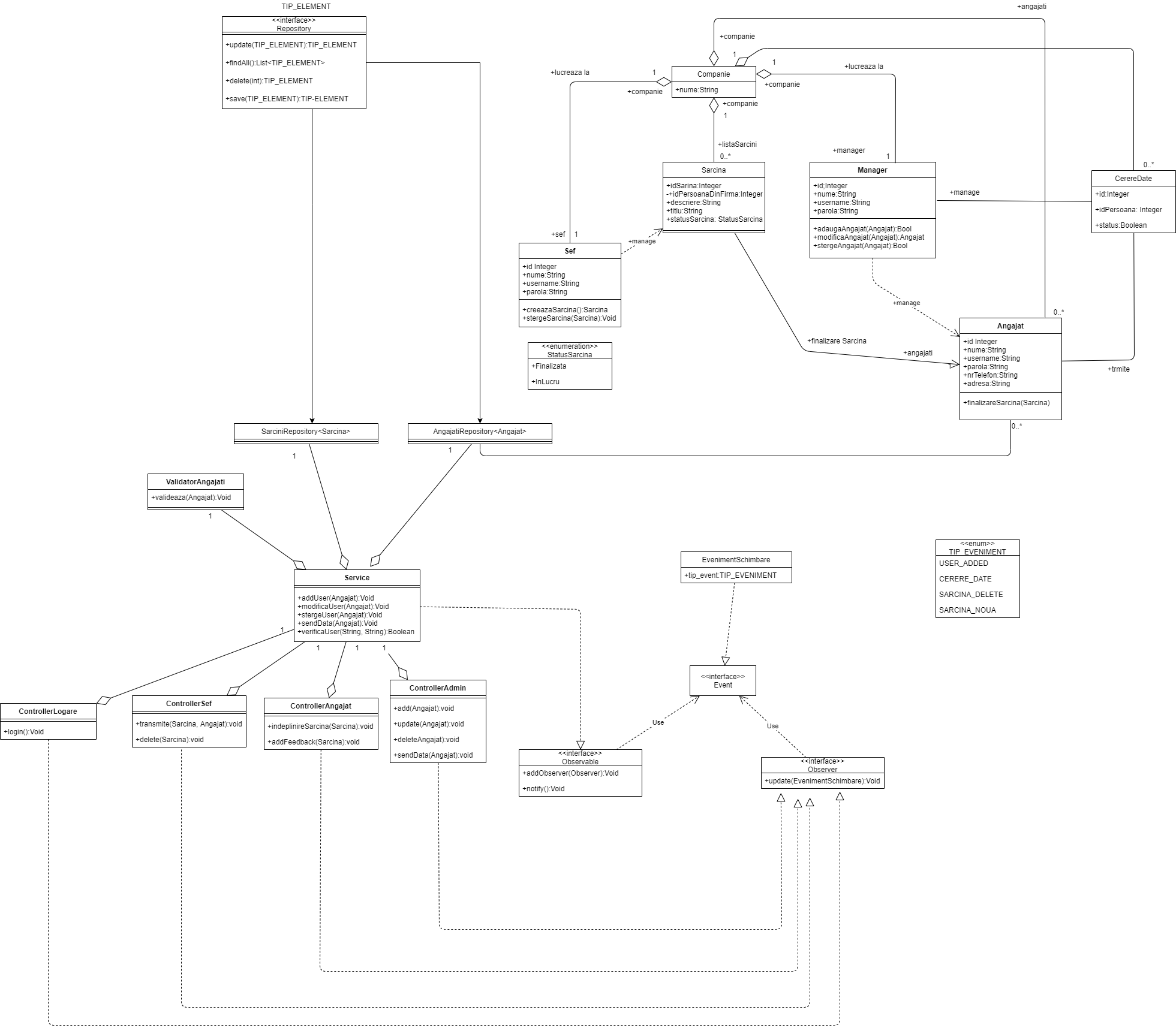
* La deschiderea aplicatiei, ii apare fereastra de login, unde introduce contul specific pentru sef, parola si apasa pe log in. Dupa care ii apare fereastra unde vede angajatii prezenti in firma. El poate sa transmita sarcini, pentru asta selecteaza un angajat, si apasa pe butonul de transmite sarcina, atunci ii apare o fereasta in care introduce un titlu si o descriere. Cand apasa pe submit srcina o sa ii apara in lista cu sarcini transmise. El poate sa stearga o sarcina, apasand pe butonul de delete in dreptul srcinii, sa vada ce feedback a aprimit sarcina aceea, sau sa o modifice, caz in care va fi notificat de asta si angajatul.

**Angajat:**

* La deschiderea aplicatiei, ii apare fereastra de login unde introduce username si parola, primite de la administrator. Daca a uitat parola, introduce doar username si apasa pe butonul recuperare date, atunci o sa ii apara o nou parola cu care se va putea loga. Dupa logarea cu succes o sa vada toate srcinile pentru el, cu fndal verde sarcini indelinite deja si cu fundal galben sarcini ce trebuie rezolvate. Pentru a indeplinii o sarcina apasa pe butonul cu bifa din dreptul ei, pentru a vedea descrierea apasa pe butonul cu ”i”, iar pentru a adauga un feedback apasa pe butonul cu “+”. Cand adauga un feedback seful va fi notificat. Pentru a parasi aplicatia inchide fereastra sau apasa pe log out, moment in care dispare din lista cu angajati a sefului.



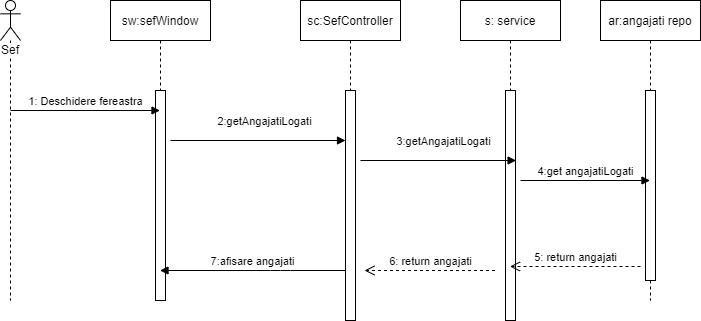
**Diagrama de clase**



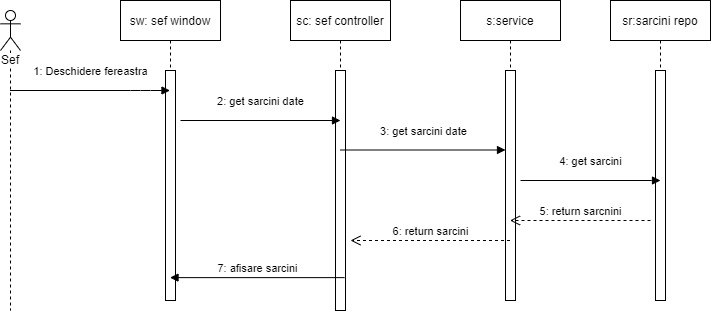
**Diagrama bazei de date**



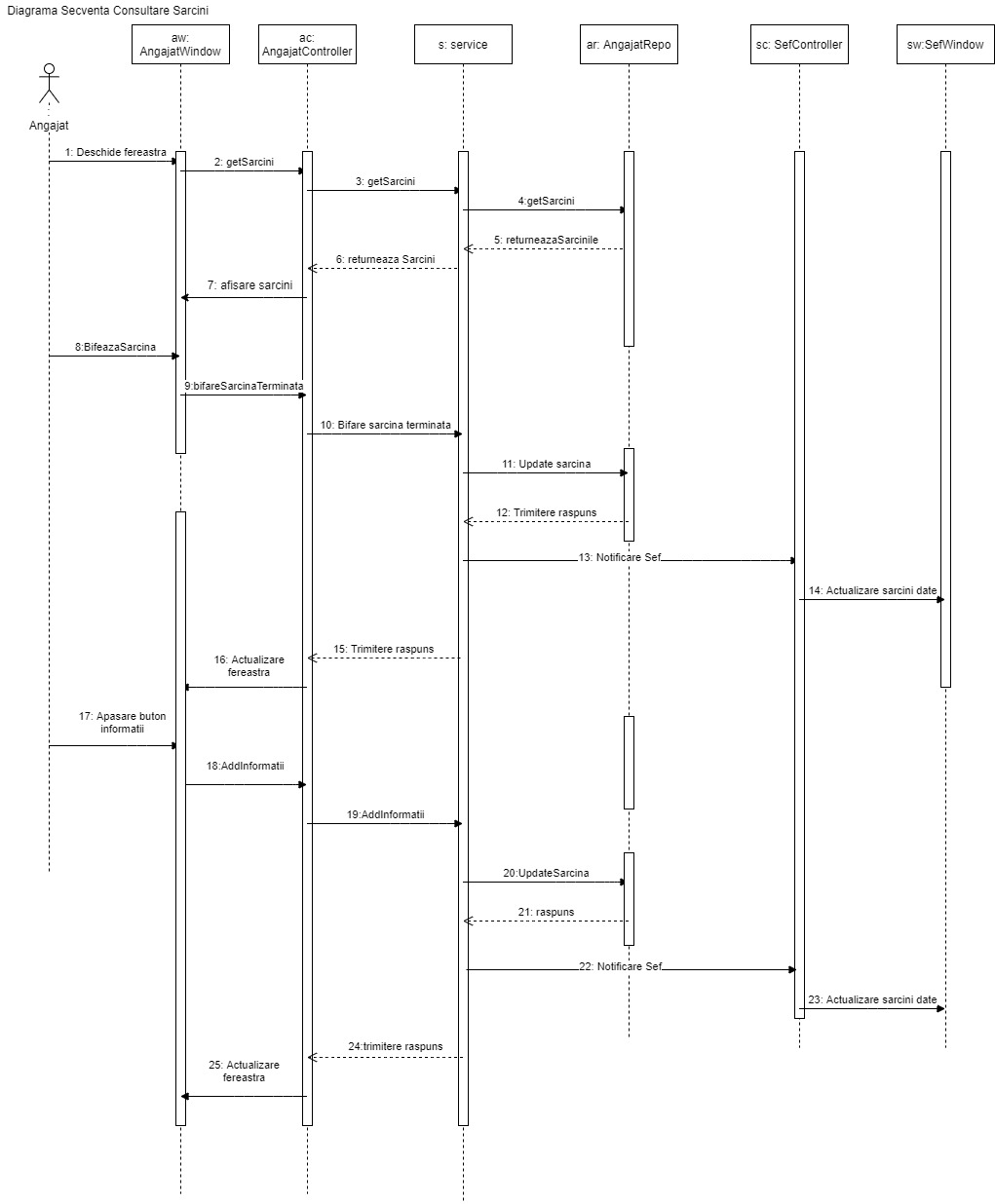
**Diagrama secventa consultare angajati**



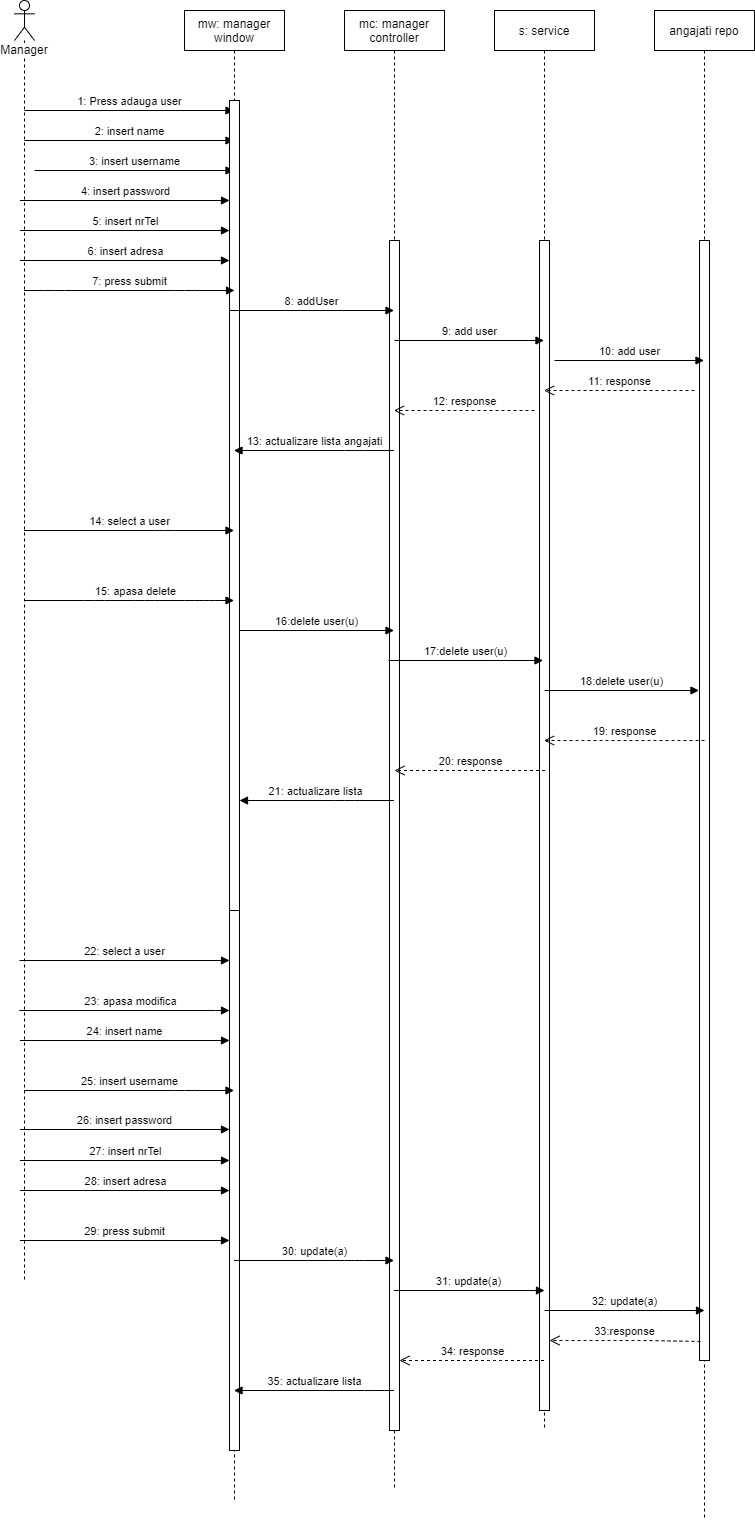
**Diagrama secventa consultare sarcini date**



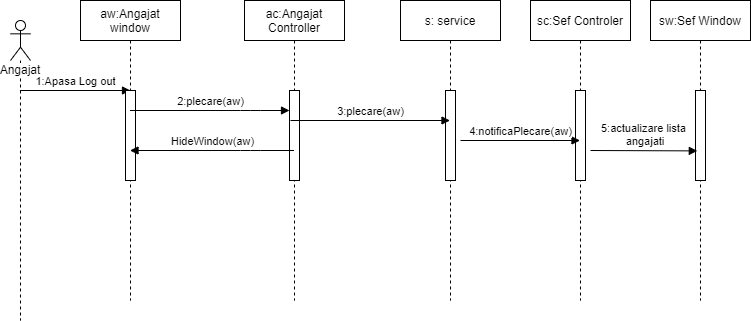
**Diagrama secventa Consultare Sarcini**



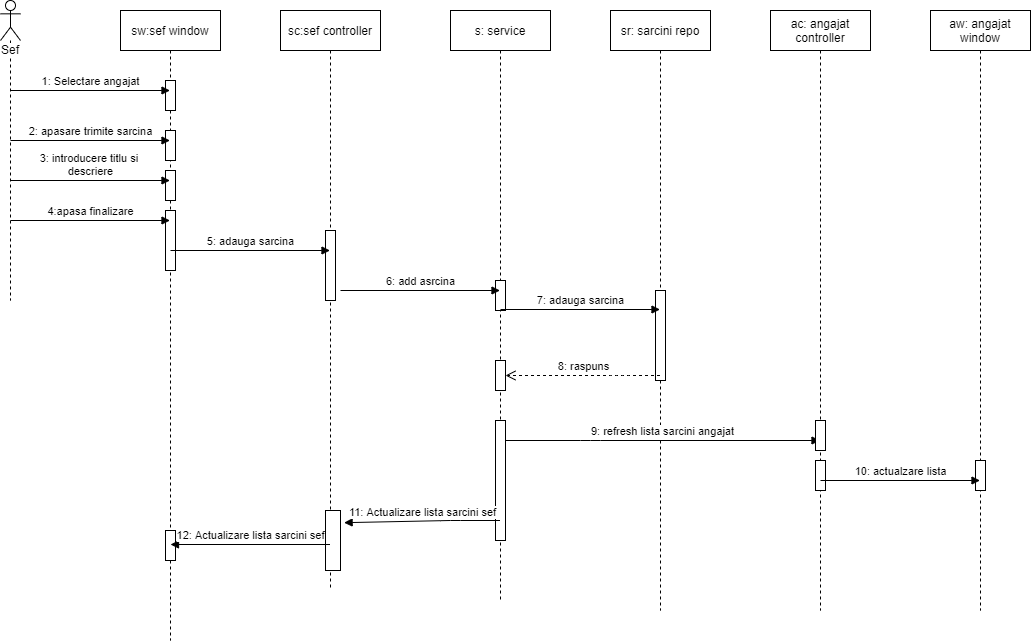
**Diagrama secventa gestionare angajati**



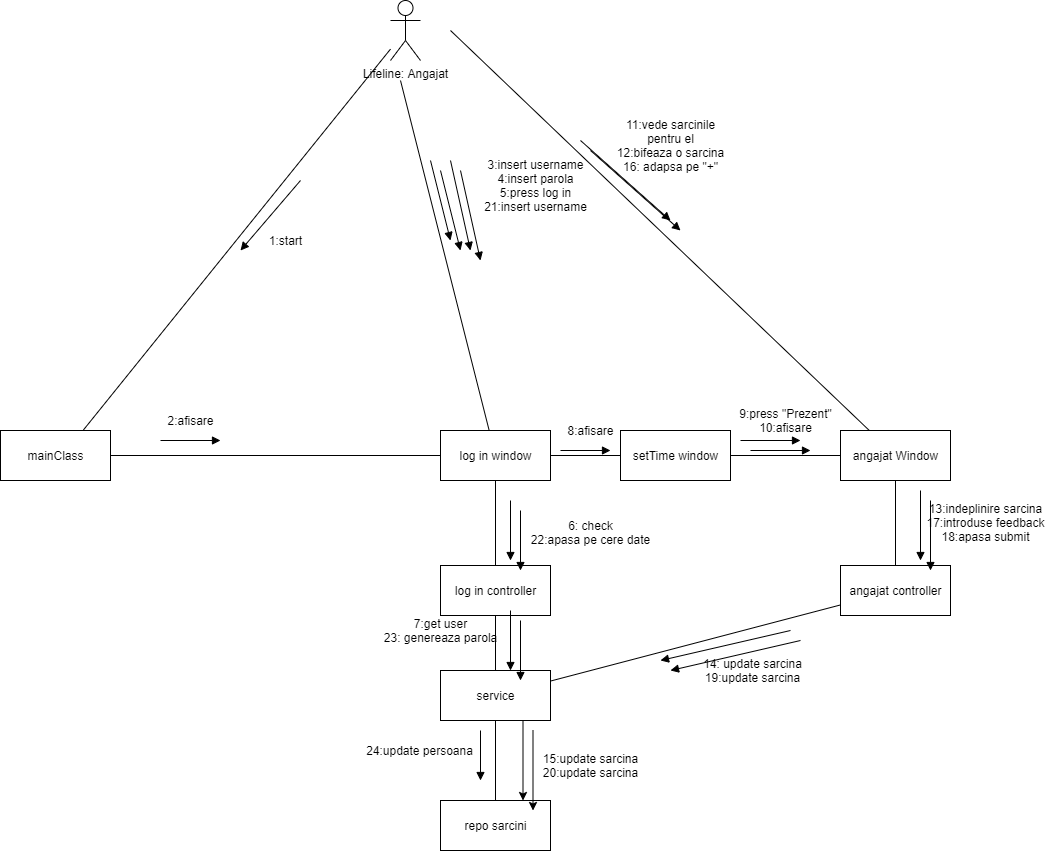
**Diagrama secventa plecare angajat**



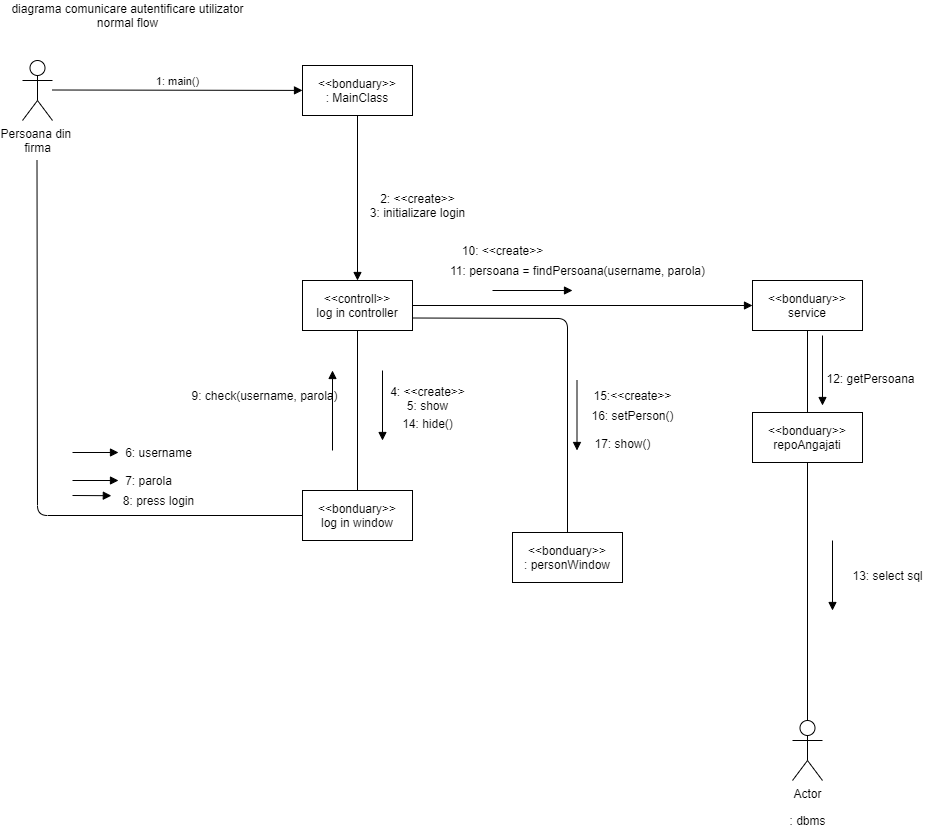
**Diagrama secventa transmite sarcina**



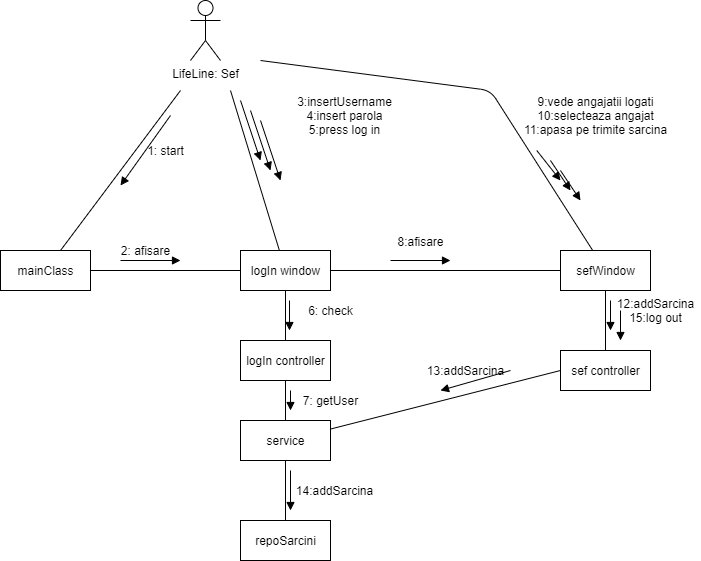
**Diagrama comunicare angajat**



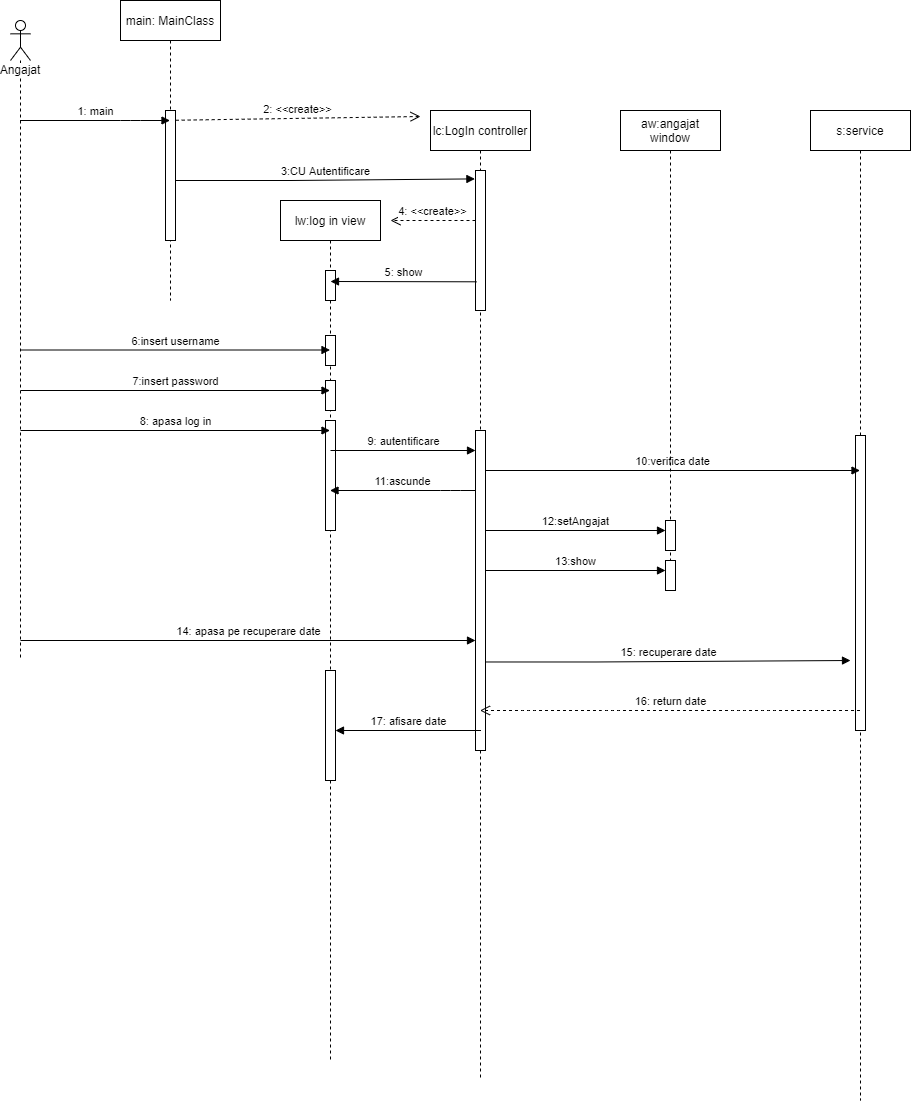
**Diagrama comunicare autentificare**



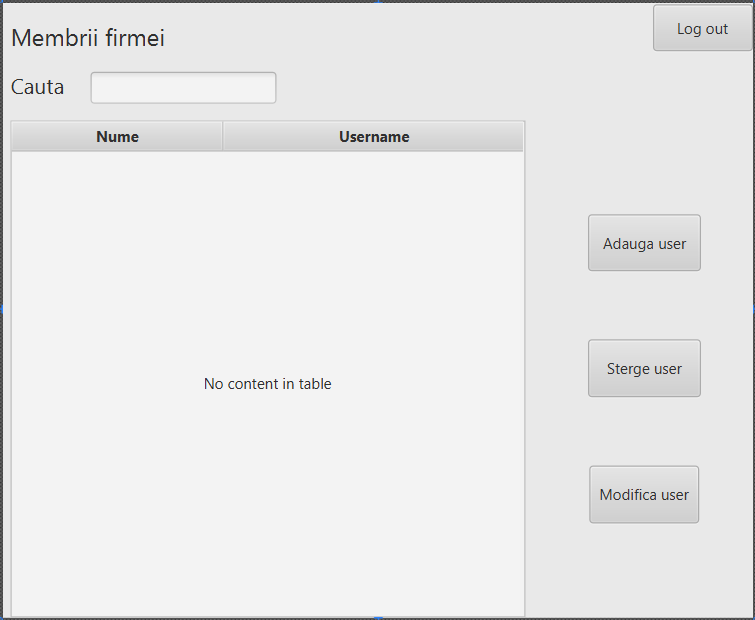
## **Diagrama Comunicare Sef**



**Diagramna secventa autentificare**

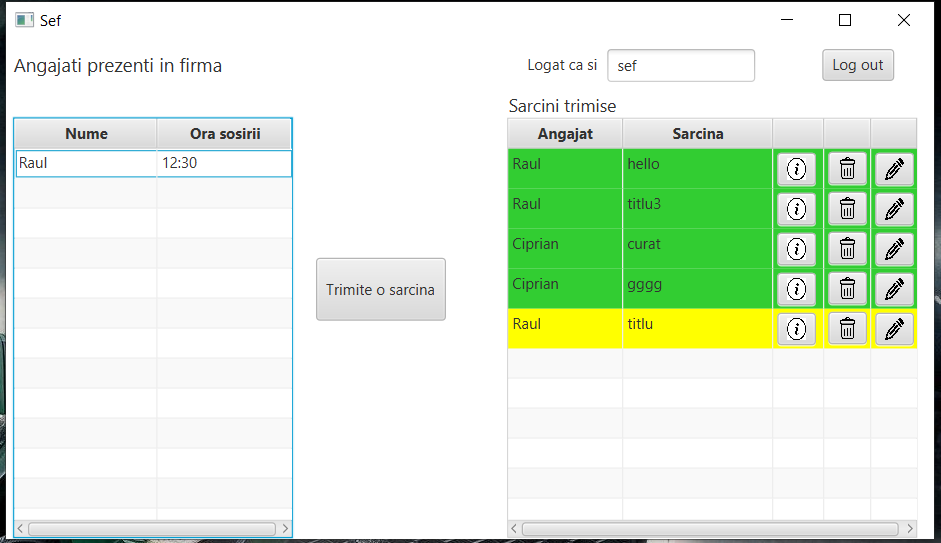


**Interfata administratorului:**



Administratorul poate sa caute un angajat dupa nume folosind bara de search, vede in lista angajatii cu nume si username si are cele 3 butoane pentru operatii specifice.

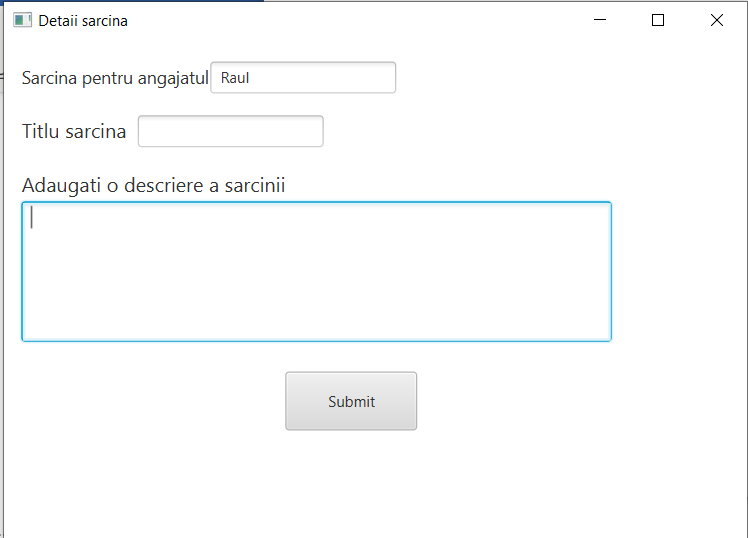
**Interfata sef:**



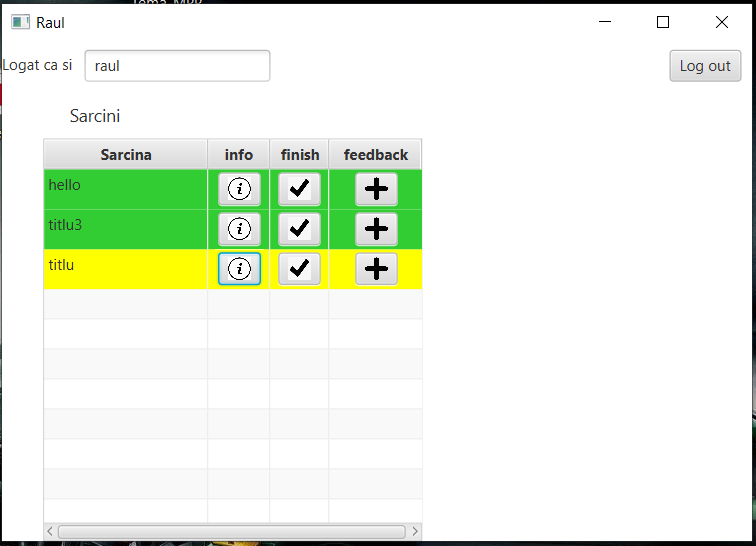
In stanga sunt angajati prezenti in momentul de fata in firma.Seful selecteaza unul din ei si apasa butonul trimite sarcina, dupa care completeaza cu titlu si descriere fereastra ce apare si apasa submit.

In dreapta sunt sarcinile trimise pentru toti angajatii. Pe butonul i se verifica feedback ul de la angajat.

**Interfata in care se dau detalii despre sarcina:**



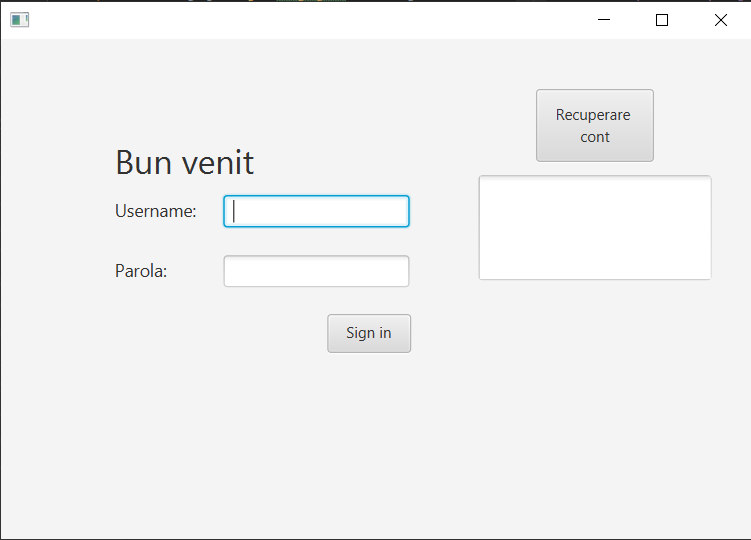
**Interfata pentru angajat:**



Sus in stanga apare numele angajatului care s a logat.

In lista apar sarcinile proprii. Pentru a completa o sarcina apasa pe bifa. Pentru a vedea descrierea sarcinii se apasa pe i. Pentruadauga feedback se apasa pe +.

**Interfata Login:**



Dupa introducerea username si parola si apasarea butonului sign in apare fereastra corespunzatoare persoanei care s a logat. Daca introduce username si apasa pe recuperare cont, se afiseaza o parola cu care si poate loga.

**Bibliografie:**

<http://www.cs.ubbcluj.ro/~vladi/Teaching/Didactic/ISS%202019-2020/info%20romana/curs/>

<http://www.cs.ubbcluj.ro/~vladi/Teaching/Didactic/ISS%202019-2020/info%20romana/seminar/>

<https://docs.oracle.com/javase/7/docs/api/>

<https://docs.jboss.org/hibernate/orm/3.2/api/>