**FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE**

**DEPARTAMENTUL CALCULATOARE**

**Proiect**

**la disciplina**

**Baze de date**

**Aplicație pentru gestionarea facultății**

**An academic : 2022 – 2023**

**Bugnariu Dan**

**Zaharie Oana Denisa**

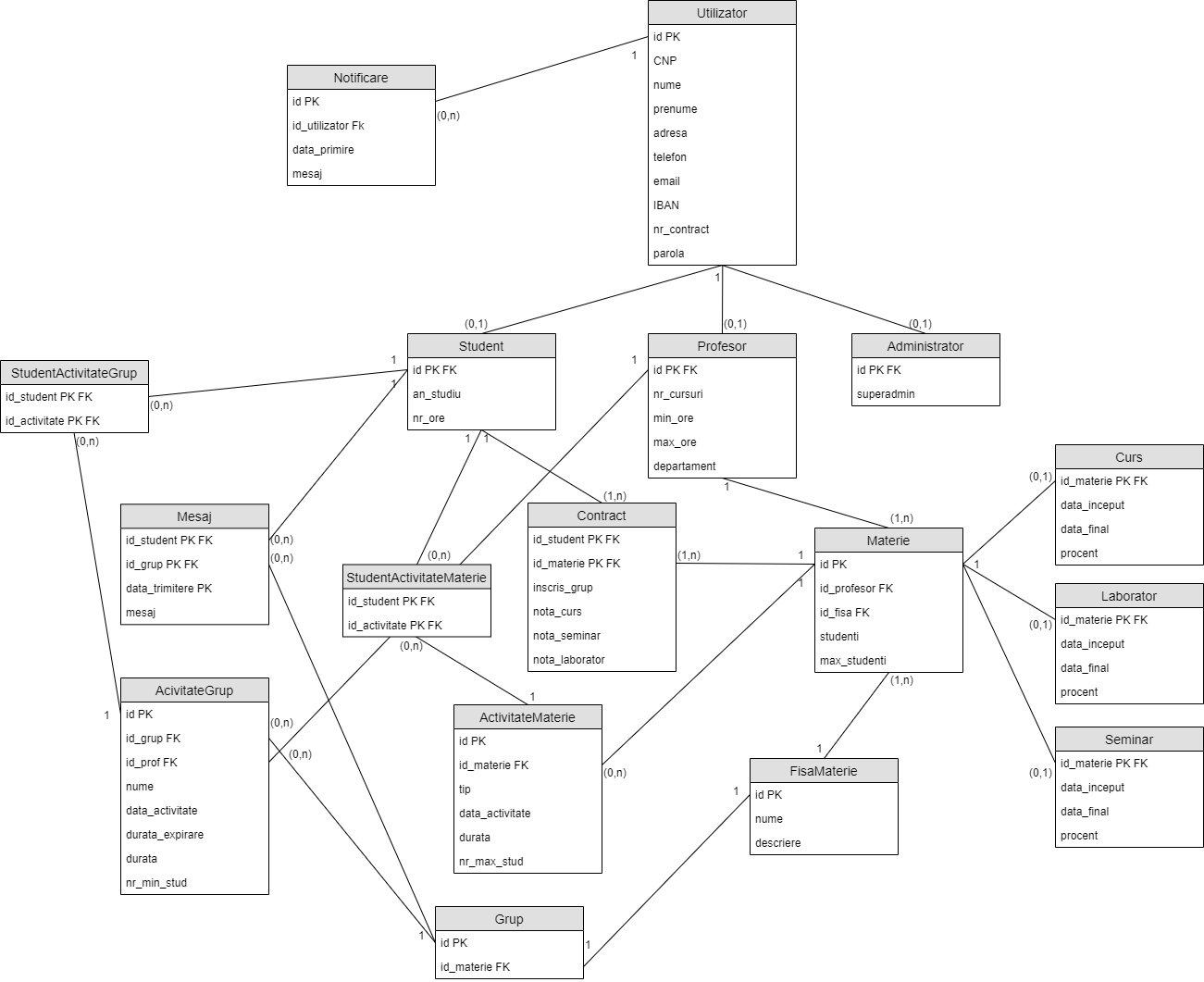
Cuprins:

1. Introducere.............................................................................................................................................3
2. Schema UML.........................................................................................................................................3
3. Funcționalități utilizatori........................................................................................................................4
   1. Student............................................................................................................................................4
   2. Profesor...........................................................................................................................................4
   3. Administrator..................................................................................................................................4
   4. Super-administrator........................................................................................................................4
   5. Grup de studiu.................................................................................................................................5
4. Cod SQL.................................................................................................................................................5
   1. Proceduri.........................................................................................................................................5
   2. Triggere.........................................................................................................................................11
5. Entități și atributele lor.........................................................................................................................11
6. Interfața grafică....................................................................................................................................12
7. Limitări și dezvoltări ulterioare...........................................................................................................15

1. Introducere

Proiectul presupune realizarea unei aplicații pentru gestionarea unei platforme de studiu. Aceasta oferă o interfață grafică care are scopul de a ușura interancțiunea cu utilizatorii. Fiecare funcționalitate este creată astfel încât modul său de utilizare să fie foarte simplu de înțeles și folosit. Pentru implementare am folosit o bază de date în care reținem toate informațiile necesare unei bune funcționări.

Aplicația adună toate datele necesare studenților într-un singur loc, pentru a evita folosirea prea multor site-uri, oferindu-le și posibilitatea de a comunica între ei și de a se ajuta reciroc. Profesorii își pot organiza ușor și rapid cursurile predate și notele examenelor. Pentru administratori este un mod ușor de gestionare a activităților din cadrul facultății, a studenților și a profesorilor.

2. Schema UML

3. Funcționalități utilizatori

Toți utilizatorii au posibilitatea de a se deloga și de a-și vizuliza datele personale, fără a le putea modifica.

* 1. Student
  + Vizualizare notificari legate de grupurile de studiu
  + Vizualizarea și descărcarea orarului
  + Vizualizarea notelor
  + Înscrierea în grupuri de studiu
  + Înscrierea la cursri
  + Vizualizarea orarului cu posibilitatea de descărcare
  1. Profesor
  + Vizualizarea notificărilor
  + Vizualizarea și descărcarea orarului
  + Vizualizrea și descrcarea catalogului cu stdenții înscriși la cursuri si notele lor
  + Înscriere pentru predarea unui curs, cu alegere tipurilor de activități predate și procentul fiecăreia din nota finală
  + Creearea și programarea activităților (curs, seminar, laborator)
  + Notarea studenților
  1. Administrator
* Crearea conturilor pentru profesori și studenți
* Crearea materiilor care se vor putea preda
* Modificarea datelor utilizatorilor cu posibilitatea de ștergere a contului
* Vizualizare date ale altor utilizatori
  1. Super-administrator

Are aceleași funcționalităti ca administrtorul simplu, dar la cele cu interacțiune pe urilizatori poate opera și pe admini.

* 1. Grup de studiu
* Înscriere/Părăsire grup
* Creare de activități de studiu în grup
* Adăugare profesor la activitate
* Vizualizarea chat-ului (discuții dintre studenții înscriși în grup)
* Vizualizarea și înscrierea la activități deja existente

4. Cod SQL

1. Proceduri
2. Creare utilizatri

create procedure **createStudent**(in\_an\_studiu int,in\_cnp int,in\_nume varchar(45),in\_prenume varchar(45),in\_adresa varchar(200),in\_telefon varchar(10),in\_email varchar(45),in\_IBAN varchar(45),in\_nr\_contract int,in\_parola varchar(20))

begin

insert into utilizator(cnp,nume,prenume,adresa,telefon,email,IBAN,nr\_contract,parola) values (in\_cnp,in\_nume,in\_prenume,in\_adresa,in\_telefon,in\_email,in\_IBAN,in\_nr\_contract,in\_parola);

select @id:=id from utilizator p where p.email = in\_email;

insert into student (id,an\_studiu,nr\_ore) values (@id,in\_an\_studiu,0);

end

create procedure **createProfesor**(in\_nr\_ore\_min int,in\_nr\_ore\_max int,in\_departament varchar(45),in\_cnp int,in\_nume varchar(45),in\_prenume varchar(45),in\_adresa varchar(200),in\_telefon varchar(10),in\_email varchar(45),in\_IBAN varchar(45),in\_nr\_contract int,in\_parola varchar(20))

begin

insert into utilizator(cnp,nume,prenume,adresa,telefon,email,IBAN,nr\_contract,parola) values (in\_cnp,in\_nume,in\_prenume,in\_adresa,in\_telefon,in\_email,in\_IBAN,in\_nr\_contract,in\_parola);

select @id:=id from utilizator p where p.email = in\_email;

insert into profesor (id,nr\_cursuri,min\_ore,max\_ore,departament) values (@id,0,in\_nr\_ore\_min,in\_nr\_ore\_max,in\_departament);

end

create procedure **createAdmin**(in\_cnp int,in\_nume varchar(45),in\_prenume varchar(45),in\_adresa varchar(200),in\_telefon varchar(10),in\_email varchar(45),in\_IBAN varchar(45),in\_nr\_contract int,in\_parola varchar(20))

begin

insert into utilizator(cnp,nume,prenume,adresa,telefon,email,IBAN,nr\_contract,parola) values (in\_cnp,in\_nume,in\_prenume,in\_adresa,in\_telefon,in\_email,in\_IBAN,in\_nr\_contract,in\_parola);

select @id:=id from utilizator p where p.email = in\_email;

insert into administrator(id, superadmin) value(@id, 0);

end

create procedure **createSuperAdmin**(in\_cnp int,in\_nume varchar(45),in\_prenume varchar(45),in\_adresa varchar(200),in\_telefon varchar(10),in\_email varchar(45),in\_IBAN varchar(45),in\_nr\_contract int,in\_parola varchar(20))

begin

insert into utilizator(cnp,nume,prenume,adresa,telefon,email,IBAN,nr\_contract,parola) values (in\_cnp,in\_nume,in\_prenume,in\_adresa,in\_telefon,in\_email,in\_IBAN,in\_nr\_contract,in\_parola);

select @id:=id from utilizator p where p.email = in\_email;

insert into administrator(id, superadmin) value(@id, 1);

end

1. Vizualizare date utilizatori

create procedure **viewDataStudent**(id int)

begin

select p.cnp, p.nume, p.prenume, p.adresa, p.telefon, p.email, p.IBAN, p.nr\_contract, s.an\_studiu, s.nr\_ore

from utilizator p, student s

where p.id = s.id and p.id = id;

end

create procedure **viewDataProfesor**(id int)

begin

select p.cnp, p.nume, p.prenume, p.adresa, p.telefon, p.email, p.IBAN, p.nr\_contract, pr.nr\_cursuri, pr.min\_ore, pr.max\_ore, pr.departament

from utilizator p, profesor pr

where p.id = pr.id and p.id = id;

end

create procedure **viewDataAdmin**(ida int)

begin

select p.cnp, p.nume, p.prenume, p.adresa, p.telefon, p.email, p.IBAN, p.nr\_contract, a.superadmin

from utilizator p, administrator a

where p.id = ida and p.id=a.id;

end

1. Vizualizarea cursurilor la care este înscris studentul și a notelor

create procedure **viewcursstudent**(id int)

begin

select f.nume, f.descriere, pe.nume, pe.prenume, c.nota\_curs, c.nota\_seminar, c.nota\_laborator

from student s

join contract c on s.id = c.id\_student and s.id = id

join materie m on c.id\_materie = m.id

join fisamaterie f on m.id\_fisa = f.id

join profesor p on m.id\_profesor = p.id

join utilizator pe on pe.id = p.id;

end

1. Vizualizarea cursurilor unui profesor

create procedure **viewcursprofesor**(id int)

begin

select f.nume, f.descriere, m.max\_studenti, m.studenti

from profesor p

join materie m on p.id = m.id\_profesor and p.id = id

join fisamaterie f on m.id\_fisa = f.id;

end

1. Înscrierea studentului la o materie la profesorul cu mai putini participanti

create procedure **alocarestudent**(id\_stud int,id\_mat int)

begin

set @ex := exists(select \* from (

select fm.id

from student s,contract c,materie m,fisamaterie fm

where s.id=id\_stud and c.id\_student=s.id and c.id\_materie=m.id and fm.id=m.id\_fisa) as mat

where mat.id=id\_mat);

if(@ex=0) then

select @id := id from materie where id\_fisa=id\_mat && studenti=( select min(studenti) from materie where id\_fisa=id\_mat and studenti<max\_studenti) limit 1;

insert into contract(id\_student,id\_materie,inscris\_grup,nota\_curs,nota\_seminar,nota\_laborator) values (id\_stud,@id,false,0,0,0);

end if;

end

1. Creeare unui curs de către profesor

create procedure **alocareprofesor**(id\_prof int,id\_mat int,nr\_max int,c boolean,s boolean,l boolean, pc int, ps int, pl int)

begin

set @ex := exists(select \* from materie where id\_profesor=id\_prof and id\_fisa=id\_mat);

if(@ex=0) then

insert into materie (id\_profesor, id\_fisa, max\_studenti, studenti) values (id\_prof, id\_mat, nr\_max, 0);

select @id := id from materie where id\_profesor = id\_prof and id\_fisa = id\_mat;

if(c) then insert into curs (id\_materie, procent) value (@id, pc);

end if;

if(l) then insert into laborator (id\_materie, procent) value (@id, pl);

end if;

if(s) then insert into seminar (id\_materie, procent) value (@id, ps);

end if;

end if;

end

1. Modificarea datelor unui utilizator (de către admin)

create procedure **modifyStudent**(in\_id int, in\_an\_studiu int,in\_cnp int,in\_nume varchar(45),in\_prenume varchar(45),in\_adresa varchar(200),in\_telefon varchar(10),in\_IBAN varchar(45))

begin

update utilizator

set cnp = in\_cnp, nume = in\_nume, prenume = in\_prenume, adresa = in\_adresa, telefon = in\_telefon, IBAN = in\_IBAN

where id = in\_id;

update student

set an\_studiu = in\_an\_studiu

where id = in\_id;

end

create procedure **modifyProfesor**(in\_id int, in\_nr\_ore\_min int,in\_nr\_ore\_max int,in\_departament varchar(45),in\_cnp int,in\_nume varchar(45),in\_prenume varchar(45),in\_adresa varchar(200),in\_telefon varchar(10),in\_IBAN varchar(45))

begin

update utilizator

set cnp = in\_cnp, nume = in\_nume, prenume = in\_prenume, adresa = in\_adresa, telefon = in\_telefon, IBAN = in\_IBAN

where id = in\_id;

update profesor

set departament = in\_departament, min\_ore = in\_nr\_ore\_min, max\_ore = in\_nr\_ore\_max

where id = in\_id;

end

create procedure **modifyAdmin**(in\_id int, in\_cnp int,in\_nume varchar(45),in\_prenume varchar(45),in\_adresa varchar(200),in\_telefon varchar(10),in\_IBAN varchar(45))

begin

update utilizator

set cnp = in\_cnp, nume = in\_nume, prenume = in\_prenume, adresa = in\_adresa, telefon = in\_telefon, IBAN = in\_IBAN

where id = in\_id;

end

1. Vizualizarea grupurilor în care este înscris un student

create procedure **getGrupuri**(id\_st int)

begin

select fm.nume,c.inscris\_grup,g.id

from student s,contract c,materie m,fisamaterie fm,grup g

where s.id=id\_st and c.id\_student=s.id and m.id=c.id\_materie and fm.id=m.id\_fisa and g.id\_materie=fm.id;

end

1. Adăugare mesaj în chat

create procedure trimiteMesaj(id\_st int,id\_gr int,data\_tr datetime,msj varchar(280))

begin

insert into mesaj(id\_student,id\_grup,data\_trimitere,mesaj) values(id\_st,id\_gr,data\_tr,msj);

end

1. Vizualizarea activităților din grupul în care este înscris un student

create procedure getActivitatiGrup(id\_st int,id\_gr int)

begin

SELECT ag.id,ag.nume,ag.data\_activitate,ag.durata,ag.data\_expirare,ag.nr\_min\_stud,exists(select \* from studentactivitategrup sag where sag.id\_activitate=ag.id and sag.id\_student=id\_st) as inscris,(select count(\*) from studentactivitategrup sag where sag.id\_activitate=ag.id) as nr\_studenti

FROM grup g,activitategrup ag

WHERE g.id=ag.id\_grup and g.id=id\_gr

ORDER BY ag.id;

end

1. Vizualizarea datelor despre o activitate de grup

create procedure getActivitateGrup(id\_st int,id\_act int)

begin

SELECT ag.id,ag.nume,ag.data\_activitate,ag.durata,ag.data\_expirare,ag.nr\_min\_stud,exists(select \* from studentactivitategrup sag where sag.id\_activitate=ag.id and sag.id\_student=id\_st) as inscris,(select count(\*) from studentactivitategrup sag where sag.id\_activitate=ag.id) as nr\_studenti

FROM activitategrup ag

WHERE ag.id=id\_act;

end



create procedure verificareActivitatiGrup()

begin

select ag.id,ag.nr\_min\_stud,(select count(\*) from studentactivitategrup sag where sag.id\_activitate=ag.id) as nr\_studenti,ag.data\_activitate,ag.data\_expirare,ag.durata,ag.nume,ag.id\_grup

from activitategrup ag;

end

1. Ștergerea unei acticități de grup

create procedure stergeActivitateGrup(id\_act int)

begin

select \* from studentactivitategrup where id\_activitate=id\_act;

delete from studentactivitategrup where id\_activitate=id\_act;

delete from activitategrup where id=id\_act;

end

1. Vizualizarea curusrilor pentru student/profesor

create procedure getCursruriStudent(id\_st int)

begin

select fm.nume,m.id

from student s, contract c,materie m,fisamaterie fm

where s.id=id\_st and c.id\_student=s.id and c.id\_materie=m.id and m.id\_fisa=fm.id;

end

create procedure getCursruriProfesor(id\_pr int)

begin

select fm.nume,m.id

from profesor p,materie m,fisamaterie fm

where p.id=id\_pr and m.id\_profesor=p.id and m.id\_fisa=fm.id;

end



create procedure getActivitatiCursuri(id\_ut int,id\_mat int)

begin

select am.id,am.tip,am.data\_activitate,am.durata,am.nr\_max\_stud,exists(select \* from studentactivitatematerie sam where sam.id\_activitate=am.id and sam.id\_student=id\_ut) as inscris,(select count(\*) from studentactivitatematerie sam where sam.id\_activitate=am.id) as nr\_studenti

from materie m,activitatematerie am

where m.id=id\_mat and am.id\_materie=m.id;

end



create procedure getActivitateCurs(id\_ut int,id\_act int)

begin

SELECT am.id,am.tip,am.data\_activitate,am.durata,am.nr\_max\_stud,exists(select \* from studentactivitatematerie sam where sam.id\_activitate=am.id and sam.id\_student=id\_ut) as inscris,(select count(\*) from studentactivitatematerie sam where sam.id\_activitate=am.id) as nr\_studenti

FROM activitatematerie am

WHERE am.id=id\_act;

end

1. Triggere
2. Actualizarea numărului de studentți îscriși la un curs

create trigger **addstudent** after insert

on contract for each row

begin

update materie set studenti=studenti+1 where id=new.id\_materie;

end

1. Creare unui grup de studiu pentru fiecare materie

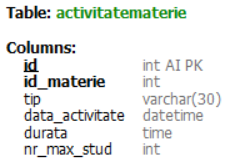
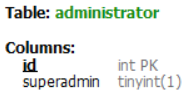
create trigger **addfisamaterie** after insert

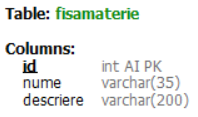
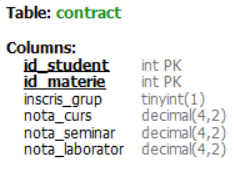
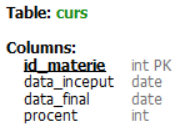
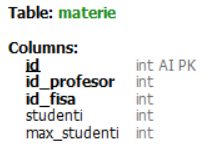
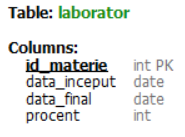
on fisamaterie for each row

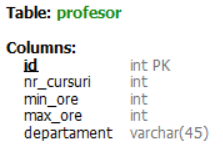
begin

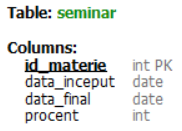
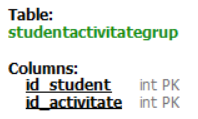
insert grup(id\_materie) values (new.id);

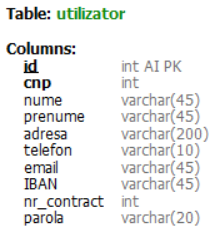
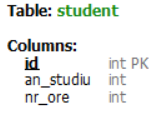
end

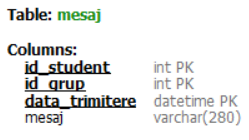
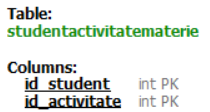
5. Entități și atributele lor







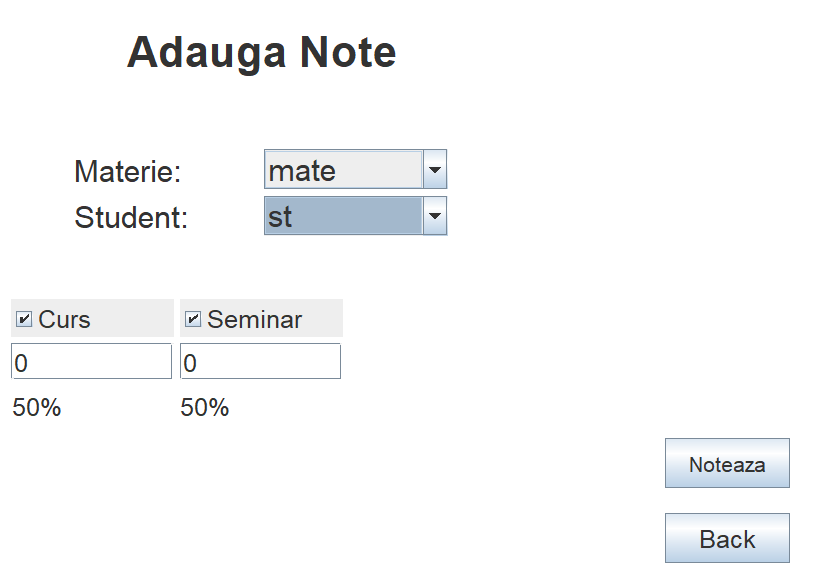




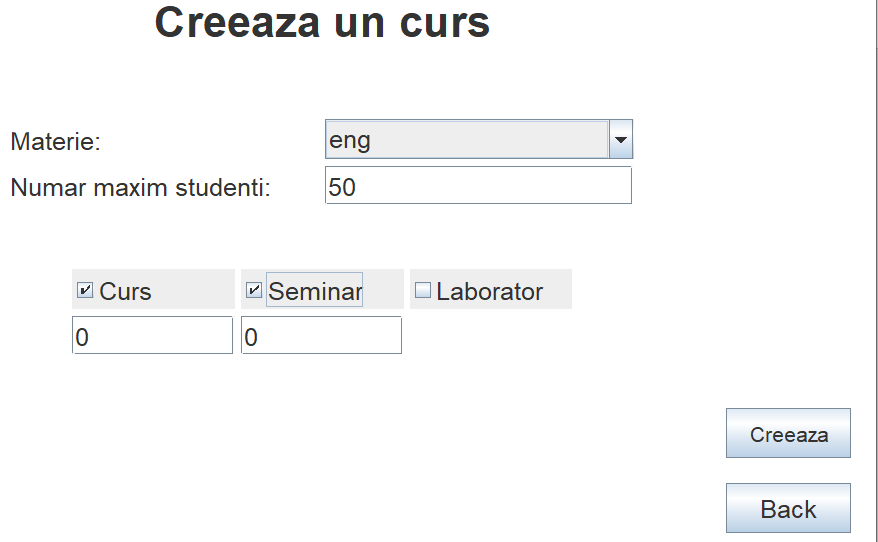
6. Interfața grafică

1. Pagina de Log in

La introducerea greșită a email-ului sau a parolei apare un mesaj de atenționare. Parola se poate face vizibilă / invizibilă prin apăsarea inconiței alăturate

1.  Pagina de notare

Profesorul poate selecta materia și studentul înscris. După selectarea materiei apar activitățile care există și, prin bifarea lor, poate introduce nota, vizualizând totodată și procentul ei din nota finală

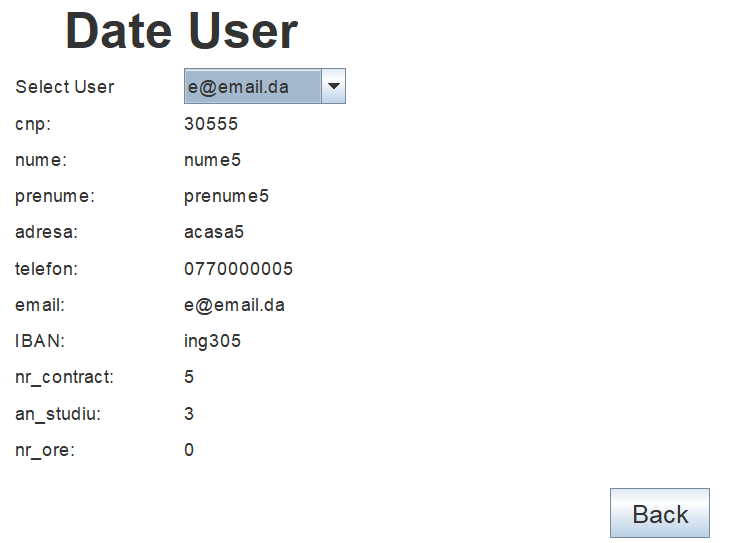


1. Crearea unui curs

Profesorul poate să aleagă materia pe care o va preda și numărul maxim de studenți ce se pot înscrie. Are opțiuni de selectare a activităților din cadrul cursului (curs, seminar, laborator), iar pentru cele alese se introduce procentul din nota finală

1.  Modificarea datelor utilizatorilor

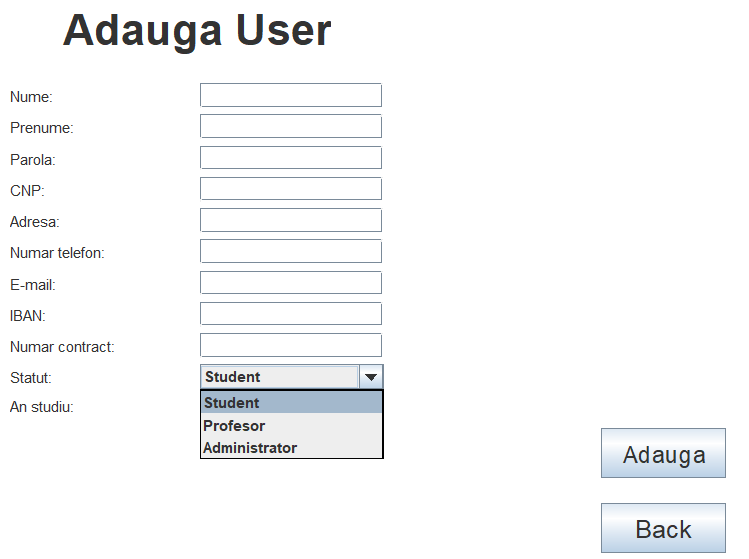
Un administrator poate alege utilizatorul pentru care va modifica datele prin adresa sa de email. Tot aici poate alege să șteargă un utilizator.

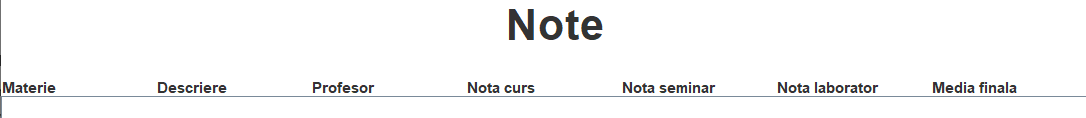


1. Vizualizarea datelor utilizatorilor

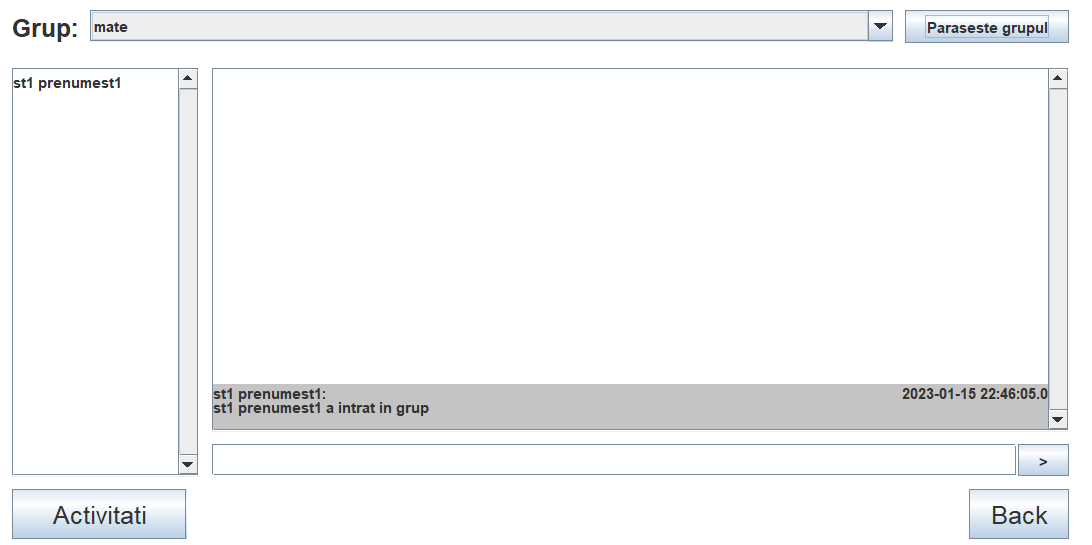
Administratorii pot vizualiza datele altor utilizatori folosind adresa lor de email.

1. Creare conturi

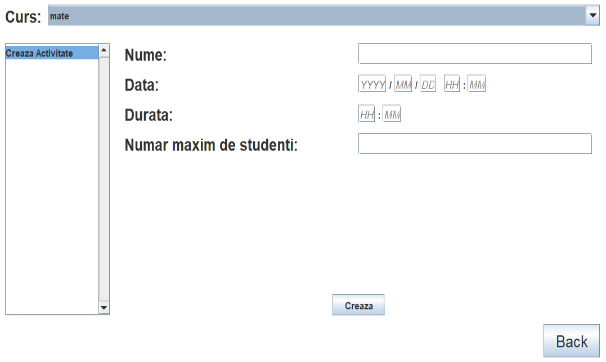
Un administrator poate adauga alți utilizatori. El le introduce datele și selectează tipul de utilizator dorit. Pentru fiecare tip se introduc și datele specifice.

1.  Vizualizarea notelor

Un student își poate vizualiza notele, atât pe activități cât și media finală, și profesorul la toate materiile.

1.  Grup de studiu

Un student se poate înscrie într-un grup de studiu la o materie. După înscriere, el poate vizualiza lista de studenți înscriși cu chat-ul de discuții și activitățile deja create, având totodată posibilitatea de a crea noi activități. El poate și ieși din grup și reintra mai tărziu, lucruri semnalate în chat. Studentul primește și o sugestie de a invita pe cineva în grup, prin trimiterea unei notificări.

1.  Creearea activităților de curs

Un profesor poate crea activități de tip curs, seminar sau laborato la care studenții se pot înscrie.

7. Limitări și dezvoltări ulterioare

* Pentru o comunicare mai ușoară, s-ar putea adăuga posibilitatea de transmitere a mesajelor către orice student sau elev, nu doar in cadrul grupului de studiu.
* Se pot adăuga convorbiri online, pentru a se putea desfășura ore cu profesorul sau întâniri între studenți
* O pagină unde profesorii să adauge materialele de studiu
* O pagină pentru adăugarea și încărcarea temelor