**Proiect de Semestru**

la disciplina

Introducere in Baze de Date

**Platforma de studiu**

An academic: 2022-2023

**Studenti:**

Oltean Darius-Ionut

Rus Ionel

**Cuprins**

[1. Introducere 1](#_Toc124124630)

[Enunt 1](#_Toc124124631)

[2. Tipuri de acces 1](#_Toc124124632)

[1.1. Utilizator 1](#_Toc124124633)

[1.2. Student 1](#_Toc124124634)

[1.3. Profesor 1](#_Toc124124635)

[1.4. Administrator 1](#_Toc124124636)

[1.5. Super-Administrator 1](#_Toc124124637)

[3. Modelul de date 3](#_Toc124124638)

[a. Diagrama UML 3](#_Toc124124639)

[b. Tabele MySQL 3](#_Toc124124640)

[c. Functii MySQL 7](#_Toc124124641)

[d. Chei straine 9](#_Toc124124642)

[e. Triggere 10](#_Toc124124643)

[f. Inserari/Calluri 11](#_Toc124124644)

[4. Interfata Java 18](#_Toc124124645)

[5. Idei de adaugat pe viitor 18](#_Toc124124646)

# Introducere

Enunt***:*** Se doreste implementarea unui sistem informatic destinat gestiunii unei platforme de studiu. Aplicatia va folosi un sistem de gestiune pentru baze de date MySQL, iar interactiunea cu aceasta va fi realizata doar prin interfata grafica. Functionalitatile pe care le va oferi programul vizeaza operatii ce tin de gestiunea studentilor, profesorilor si administrarea operatiilor curente din cadrul unor programe de studiu.

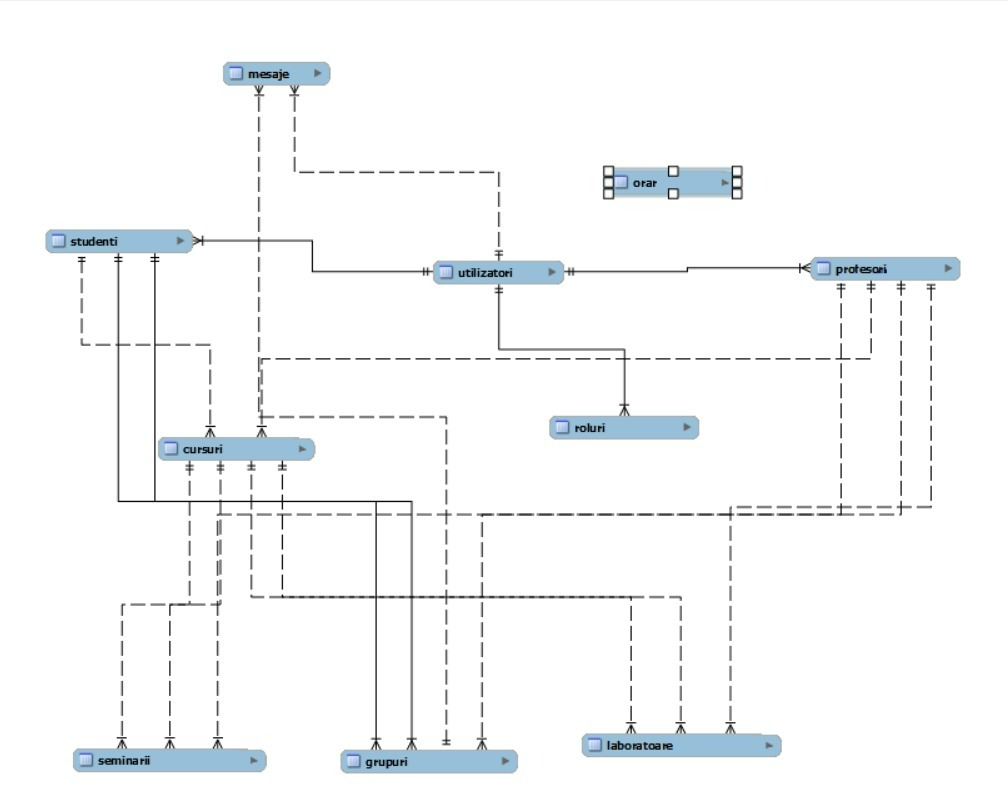
# Tipuri de acces

* 1. Utilizator***:*** Pentru fiecare tip de utilizator se vor reţine informații precum CNP, nume, prenume, adresa, număr de telefon, email, cont IBAN, numărul de contract. Fiecare utilizator își va putea vizualiza datele personale imediat după ce va accesa sistemul informatic, fără a avea însă posibilitatea de a le modifica. Totodată, programul trebuie să ofere și o funcționalitate pentru deautentificare, prin care se revine la fereastra care solicită datele de acces, astfel încât și un alt utilizator să îl poată folosi ulterior, fără a fi necesară repornirea sa.
  2. Student***:*** Pentru un utilizator de tip student se va retine si anul de studiu si numarul de ore pe care trebuie sa le sustina.
  3. Profesor***:*** Pentru un utilizator de tip profesor se vor retine si cursurile predate, numarul minim si numarul maxim de ore pe care le poate preda si departamentul din care face parte.
  4. Administrator***:*** Utilizatorul de tip administrator poate adăuga, modifica şi şterge informaţii în baza de date , informații legate de utilizatori. Administratorii pot sa caute utilizatorii dupa nume si ii pot filtra dupa tip, pot asigna profesorii la cursuri si pot face cautare dupa numele cursului. La cautarea unui curs se afiseaza si numele profesorilor de la acel curs si un buton care permite vizualizarea tuturor studentilor inscrisi la cursul respectiv.
  5. Super-Administrator***:*** Poate opera inclusiv asupra utilizatorilor de tip administrator.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Student | Profesor | Admin |
| Date Personale | da | da | da |
| Date despre alti utilizatori | nu | nu | da |
| Date despre cursuri | doar ale sale | doar ale sale | da |
| Alocare student la activitati | da | nu | da |
| Alocare profesori la cursuri | nu | nu | da |

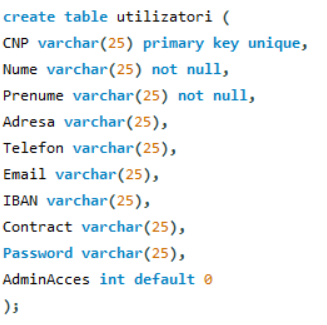
# Modelul de date

* 1. Diagrama UML:

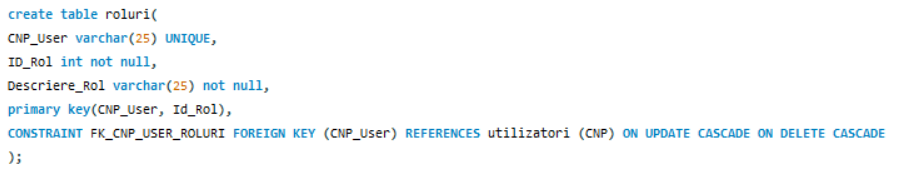


* 1. Tabele MySQL:

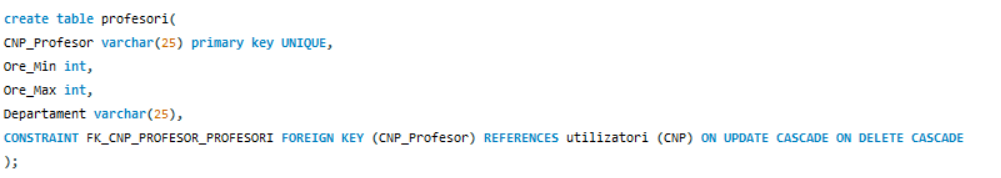
1. Utilizatori: tabela principala cu datele fiecarui utilizator;



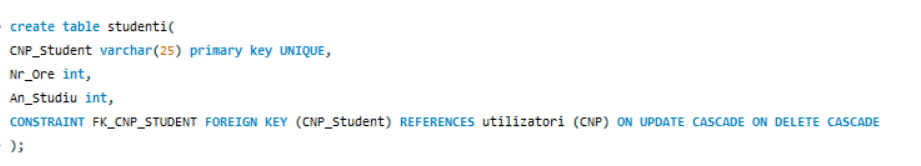
1. Roluri: tabela in care se stocheaza CNP-ul fiecarui utilizator/profesor/student cu rolul aferent cu ajutorul unui trigger;



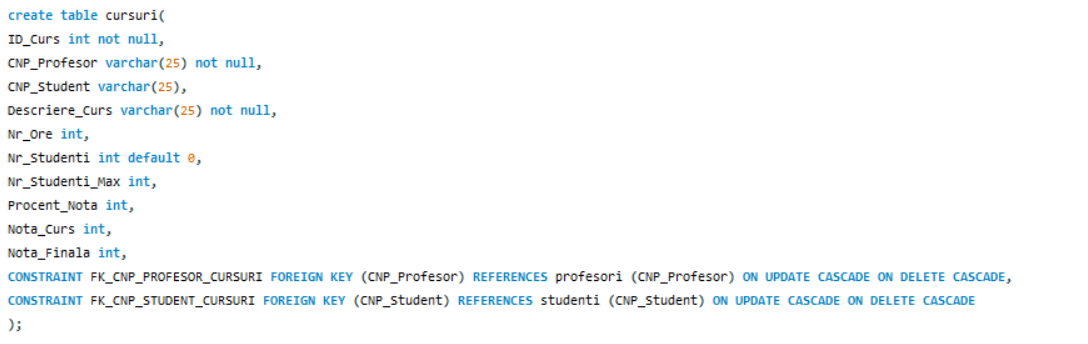
1. Profesori: tabela cu datele profesorilor;



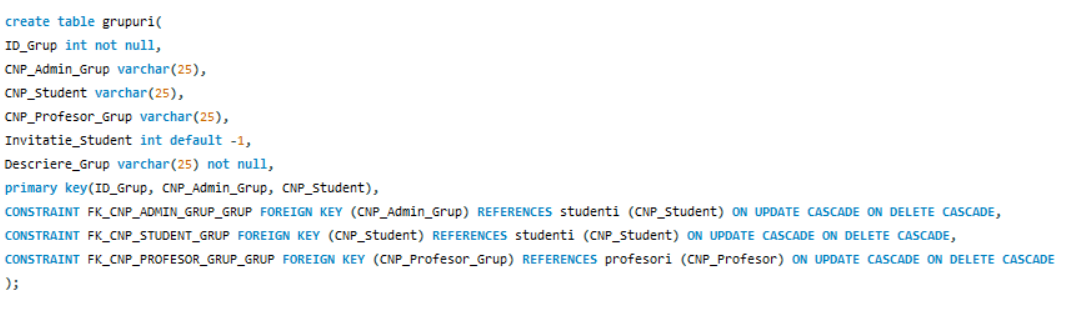
1. Studenti: tabela cu datele studentilor;



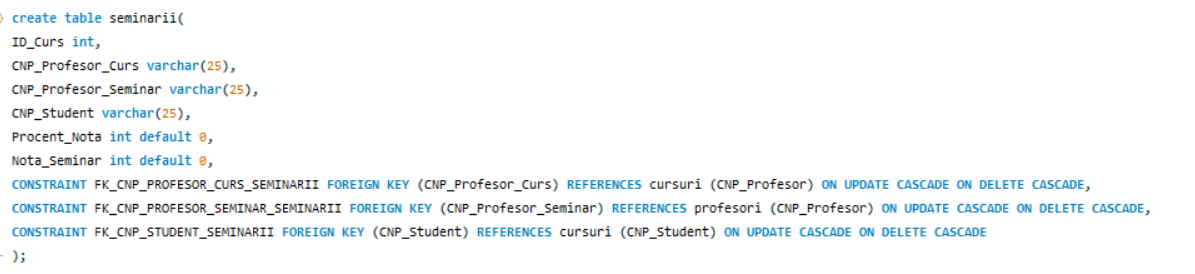
1. Cursuri: tabela in care sunt stocate toate cursurile cu studentii, respectiv notele si numarul maxim de studenti;



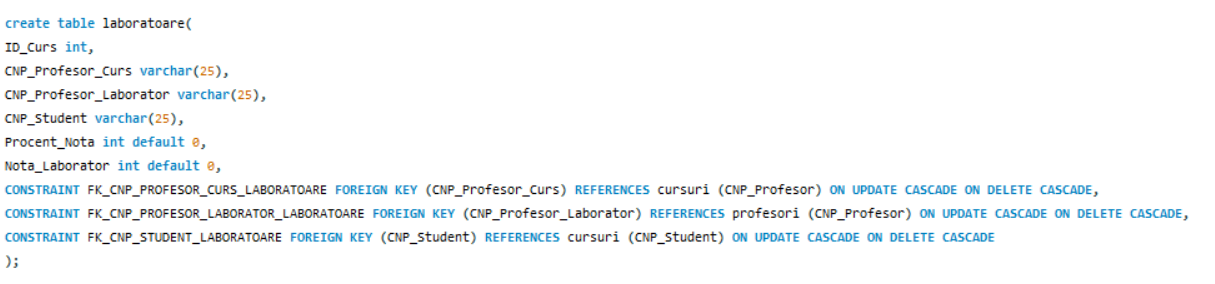
1. Grupuri: tabela pentru grupurile de studiu pentru studenti;



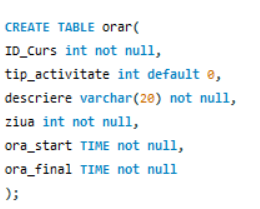
1. Seminarii: tabela legata de cursuri in cazul in care profesorul alege sa desfasoare si seminar;



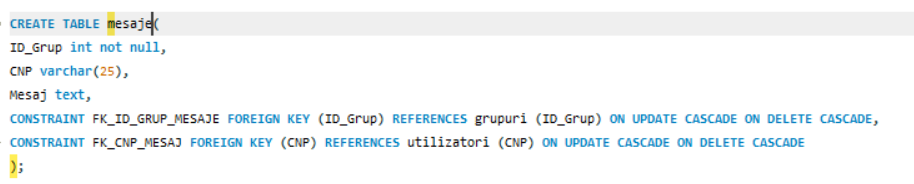
1. Laboratoare: tabela legata de cursuri in cazul in care profesorul alege sa desfasoare si laborator;



1. Orar: orarul general;



1. Mesaje: tabela in care este stocat mesajul trimis de fiecare student/profesor.



* 1. Chei straine:

Pentru fiecare tabela am ales chei straine pentru a avea aceleasi date in tabele si nu date straine.

a) Roluri – aici avem doar o cheie straina pentru CNP, cheie care este actualizata la fiecare sterge, inserare sau actualizare in tabela parinte, respectiv tabela utilizatori;

b) Profesori/Studenti – doua tabele aproximativ la fel, avand doar atribute diferite, dar aceeasi utilizare. Avem o cheie straina pentru CNP, cheie care este actualizata la fiecare sterge, inserare sau actualizare in tabela parinte, respectiv tabela utilizatori;

c) Cursuri - doua chei straine, una pentru CNP-ul studentului si cealalta pentru profesor, chei care sunt actualizate la fiecare sterge, inserare sau actualizare in tabela parinte, respectiv tabela studenti si profesori;

d) Grupuri - trei chei straine, una pentru CNP-ul studentului care este admin, a doua pentru CNP-ul studentului care face parte grupului si cealalta pentru profesorul grupului, chei care sunt actualizate la fiecare sterge, inserare sau actualizare in tabela parinte, respectiv tabela studenti, studenti si profesori;

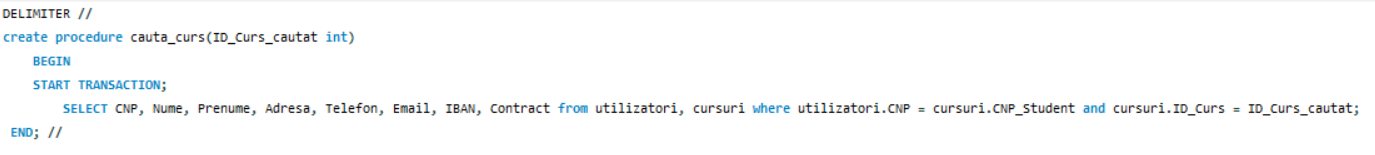
e) Seminarii - trei chei straine, una pentru CNP-ul profesorului de la curs, a doua pentru CNP-ul profesorului de la seminar si cealalta pentru studentii de la seminar, chei care sunt actualizate la fiecare sterge, inserare sau actualizare in tabela parinte, respectiv tabela studenti, studenti si profesori;

f) Laboratoare - trei chei straine, una pentru CNP-ul profesorului de la curs, a doua pentru CNP-ul profesorului de la laborator si cealalta pentru studentii de la laborator, chei care sunt actualizate la fiecare sterge, inserare sau actualizare in tabela parinte, respectiv tabela studenti, studenti si profesori;

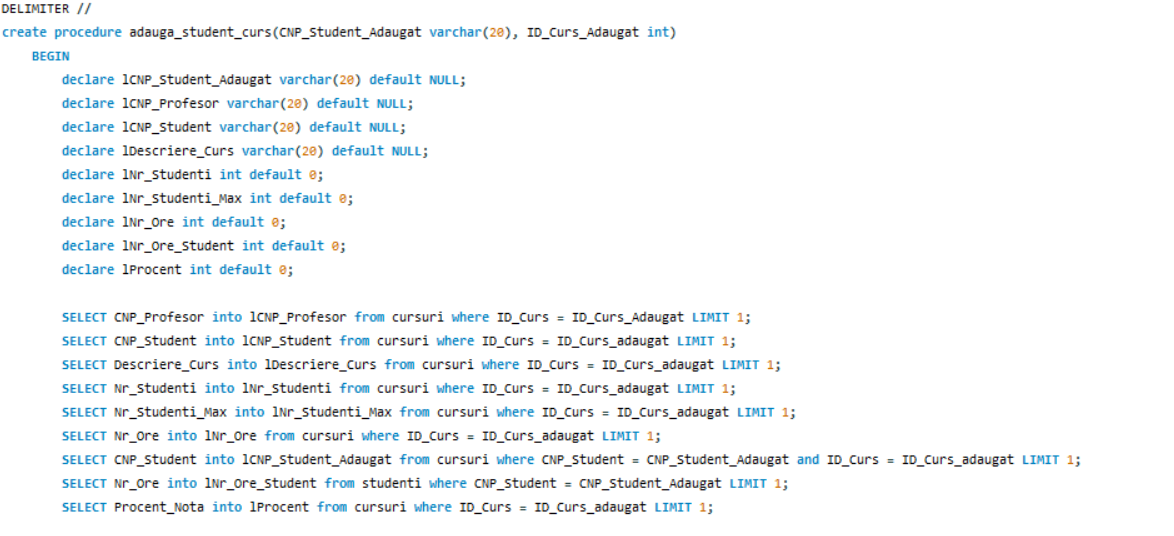
g) mesaje – doua chei straine, una pentru ID-ul grupului pe care a fost dat mesajul si cealalta pentru CNP-ul studentului care a dat mesajul.

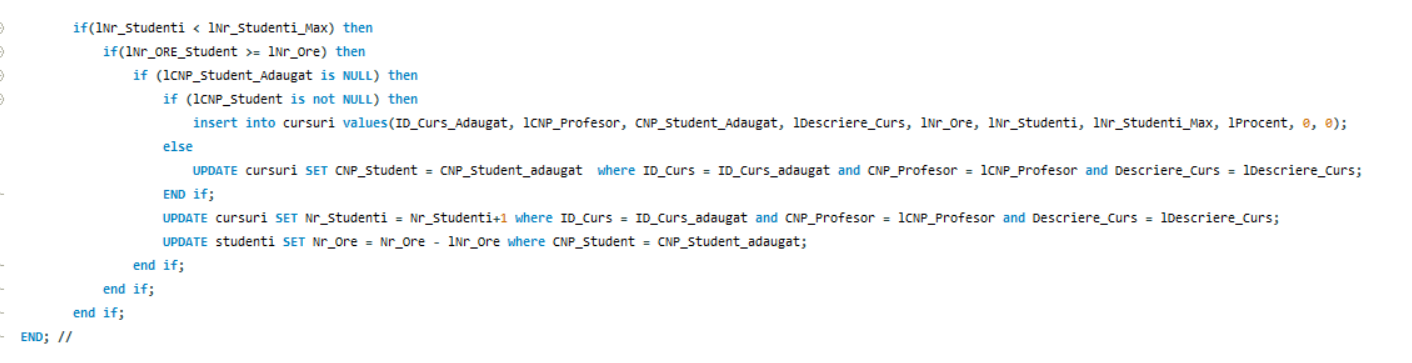
* 1. Functii MySQL:

1. cauta\_curs: functie cu ajutorul careia cauti un curs; ID\_Curs\_Cautat reprezinta ID-ul cursului pe care dorim sa-l cautam.

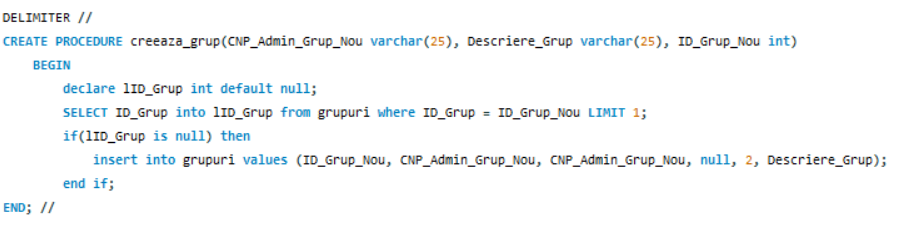


1. adauga\_student\_curs: functie de adaugare a unui student la curs; CNP\_Student\_Adaugat reprezinta CNP-ul studentului pe care dorim sa-l adaugam, iar ID\_Curs\_Adaugat reprezinta ID-ul cursului la care dorim sa adaugam studentul; Initial am verificat daca cursul respectiv mai are locuri disponibile, dupa daca studentul mai are ore disponibile, ulterior daca studentul este deja adaugat si dupa daca acel curs mai are studenti adaugati atunci o sa trebuiasca sa il inseram daca nu sa actualizam CNP-ul studentului.

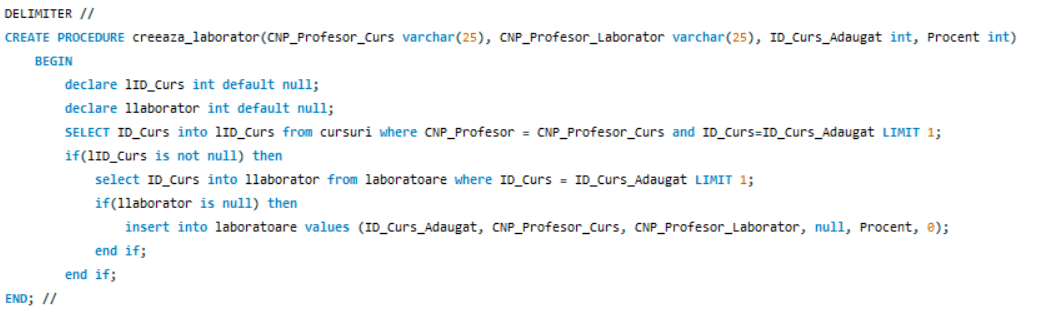




1. creeaza\_grup - functie cu ajutorul careia cream un grup de studiu; CNP\_Admin\_Grup\_Nou reprezinta CNP-ul studentului care creeaza grupul; Descriere\_Grup reprezinta descrierea grupului, respectiv activitatea pentru care este creat, iar ID\_Grup\_Nou reprezinta ID-ul unic al grupului.

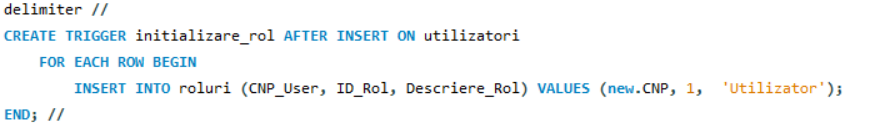


1. creeaza\_laborator – functie cu ajutorul careia cream laborator pentru un curs; CNP\_Profesor\_Curs reprezinta CNP-ul profesorului de la curs, CNP\_Profesor\_Laborator reprezinta CNP-ul profesorului de la laborator, ID\_Curs\_Adaugat ID-ul cursului la care dorim sa adaugam laboratorul, Procent reprezinta procentul pentru nota de la laborator.

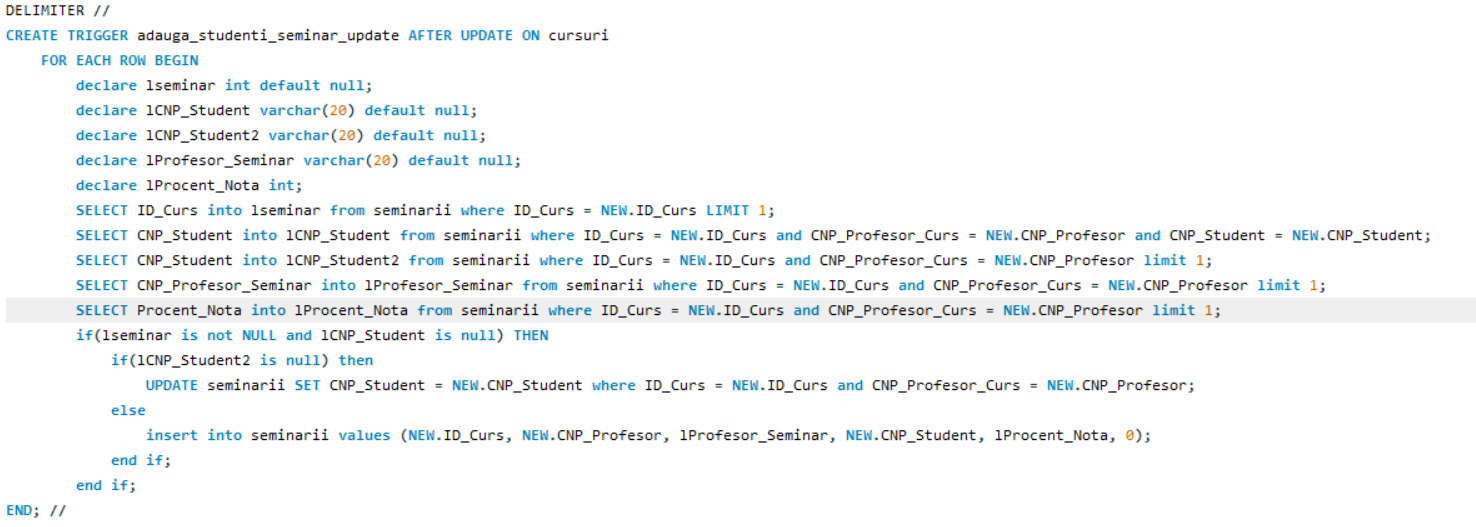


* 1. Triggere:

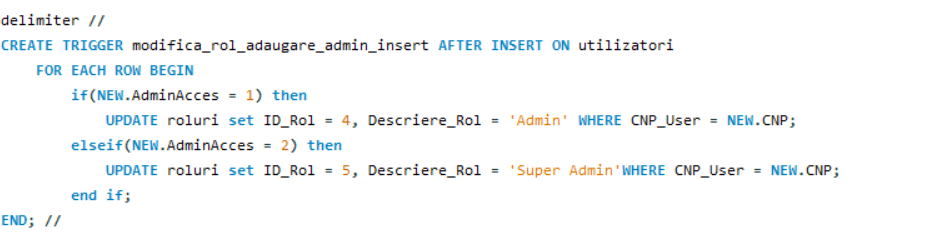
1. initializare\_rol – folosit la actualizarea tabelei roluri la fiecare inregistrare a unui utilizator nou;



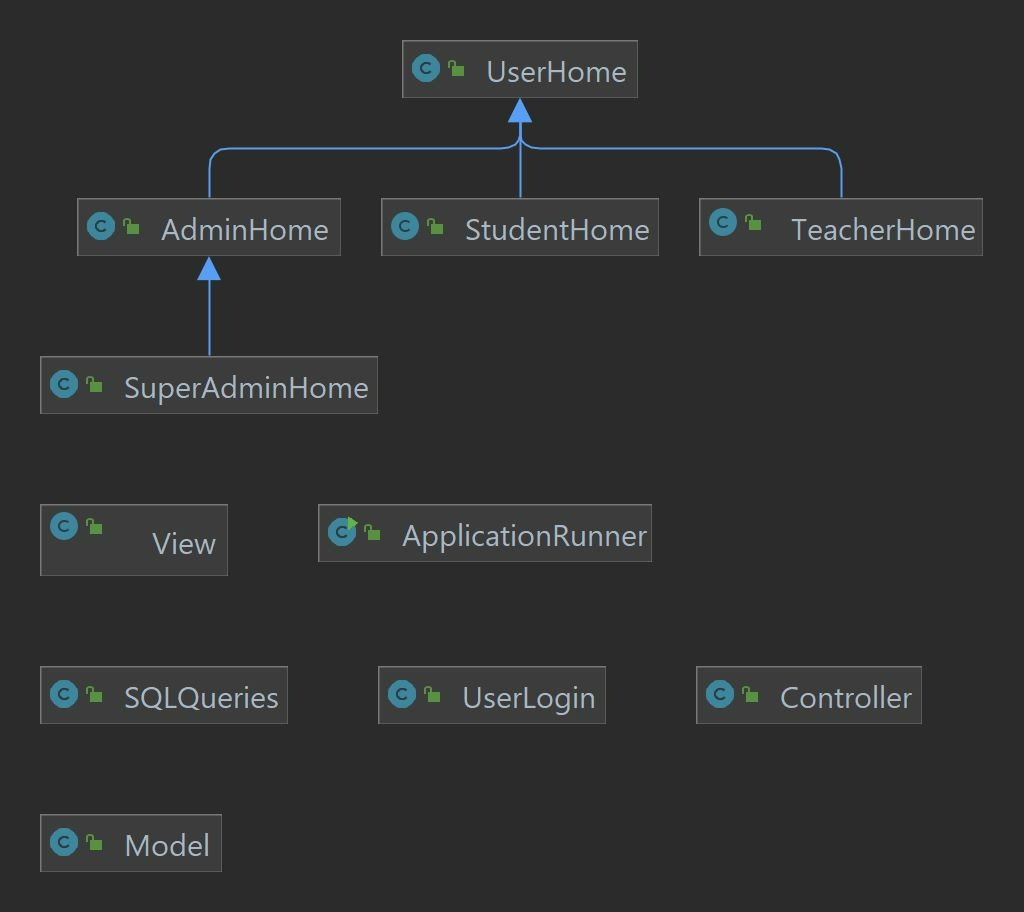
1. adauga\_studenti\_seminar\_update - folosit la actualizarea tabelei seminarii in momentul in care se modifica tabela de cursuri; Verificam daca exista seminar si daca daca studentul exista la respectivul curs, iar daca campul de CNP din seminarii este gol atunci o sa fie inlocuit cu noul CNP, daca nu o sa se insereze o linie noua; Acelasi procedeu este folosit si pentru laboratoare;



1. modifica\_rol\_adaugare\_admin\_insert – folosit la actualizarea tabelei de roluri in momentul in care modificam accesul unui utilizator la admin/super-admin; Verificam daca accesul este 1/2 si actualizam in functie de asta.



# Diagrama de clase Java

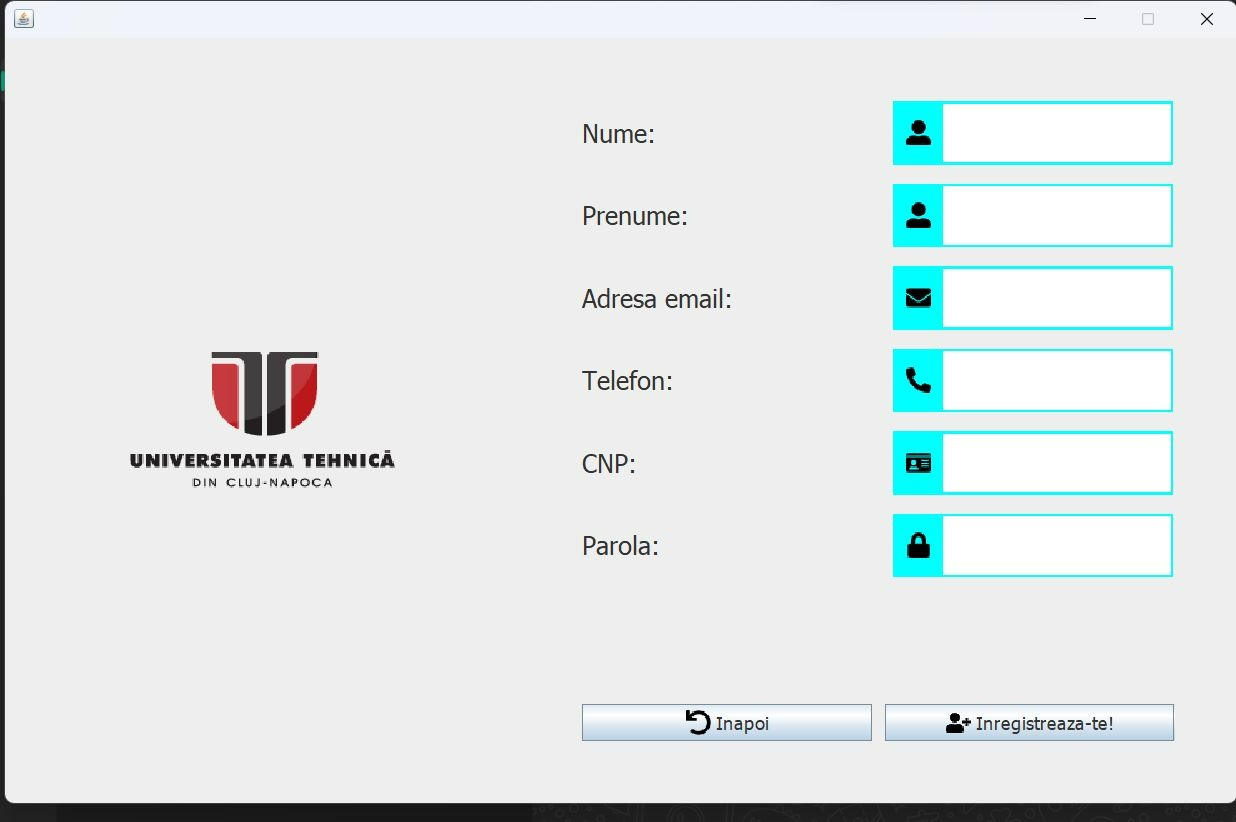


# Interfata Java

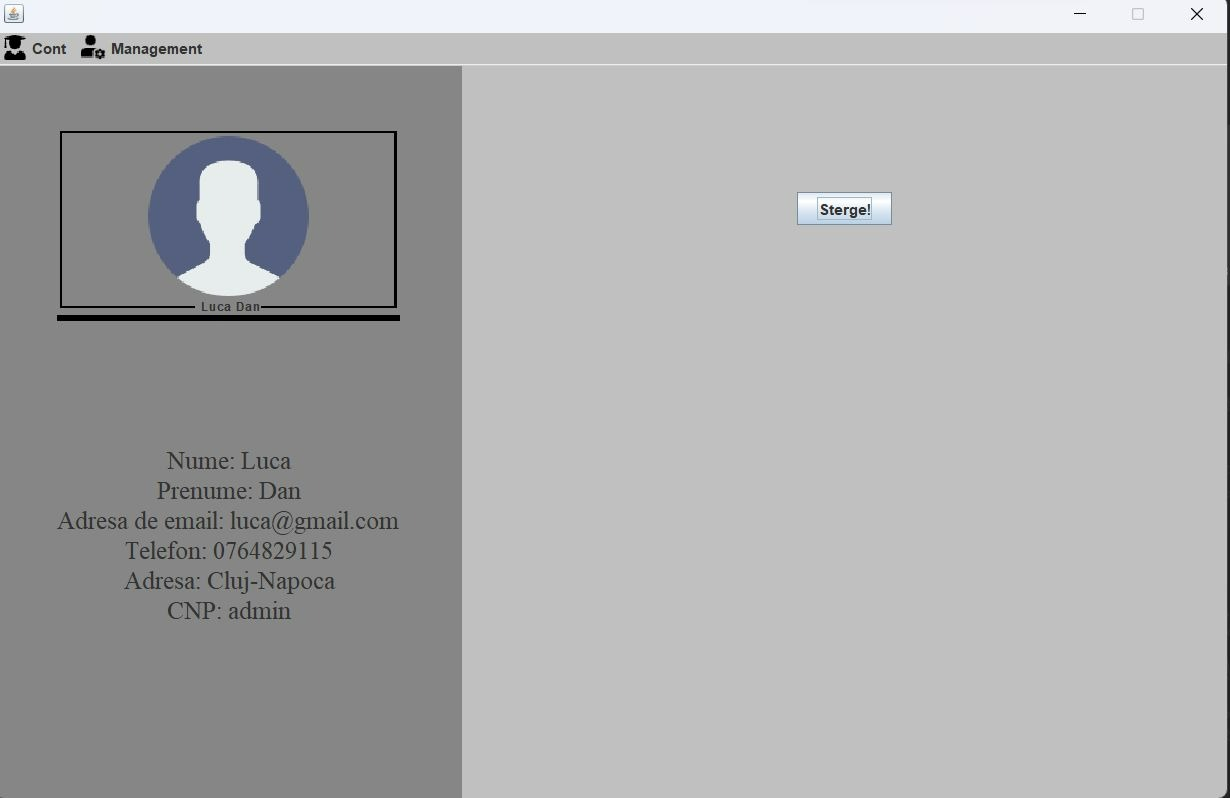
Logare:



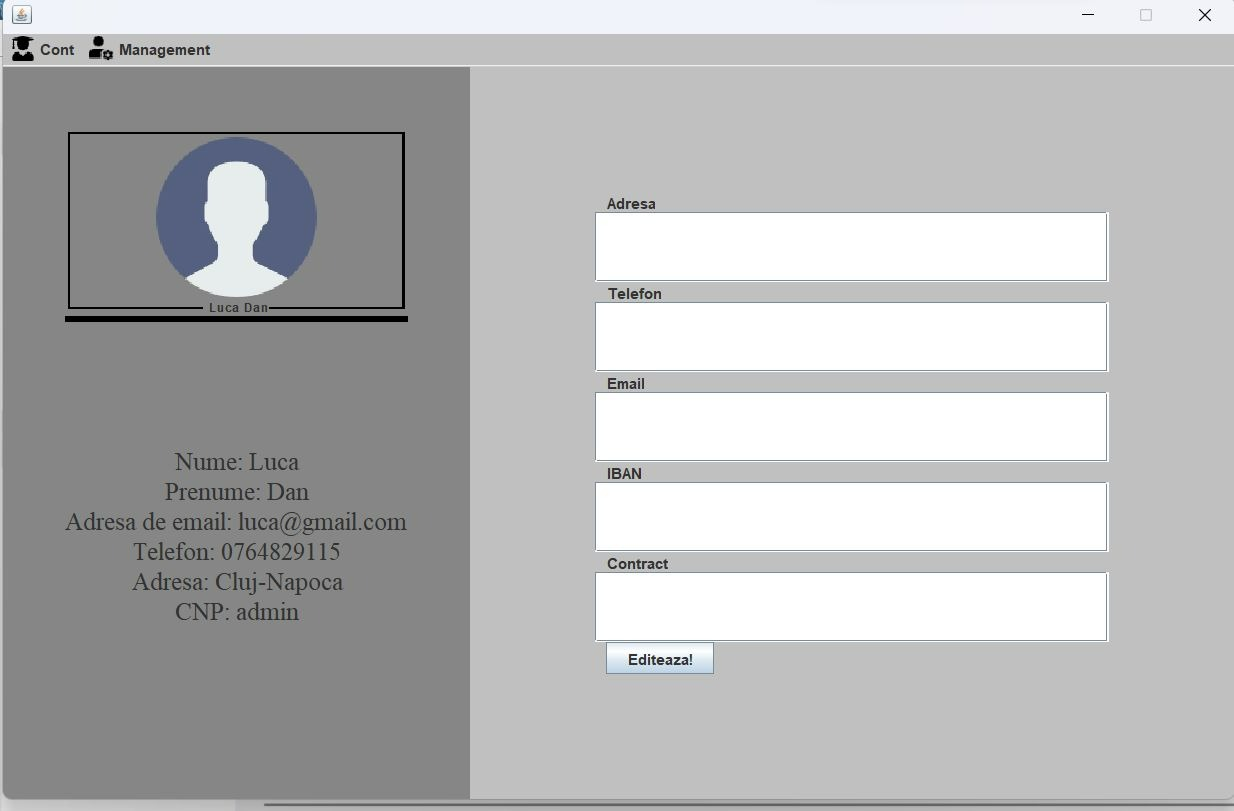
**Inregistrare:**



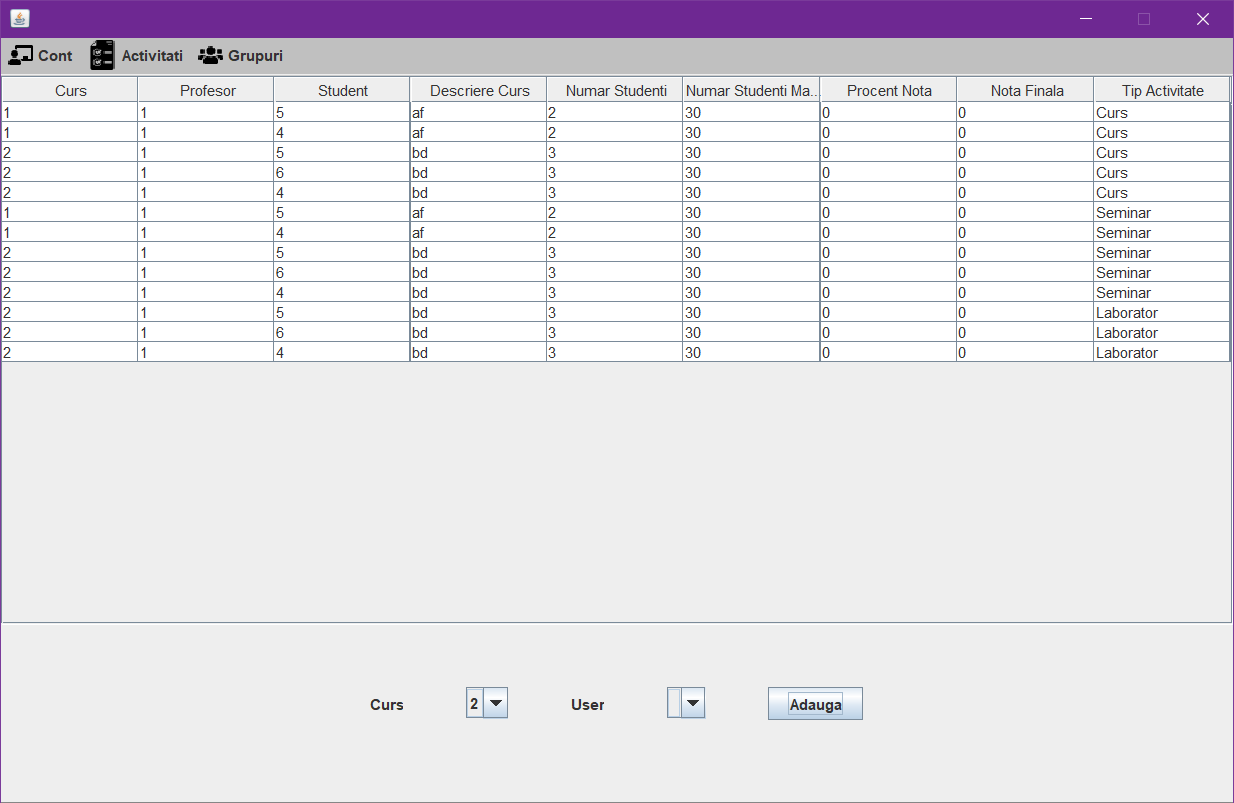
**Profilul meu:**



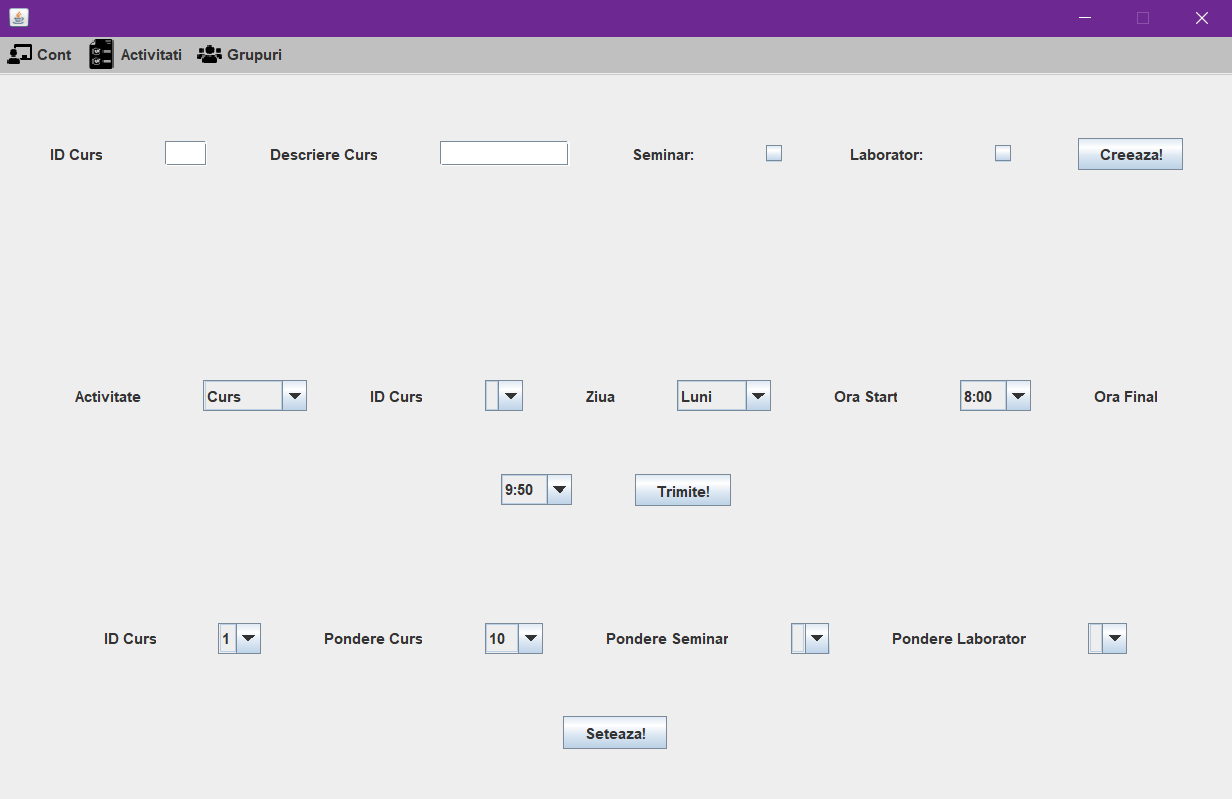
**Editeaza Cont:**



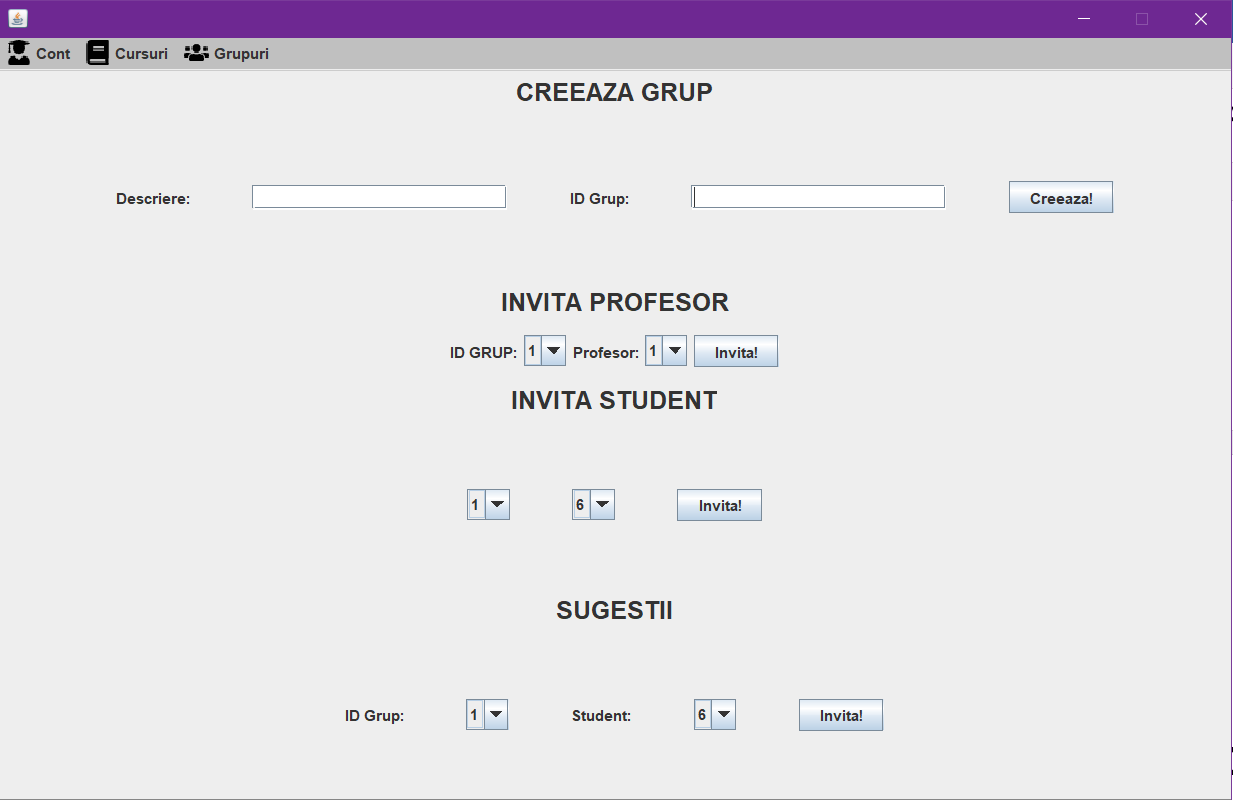
**Cursurile mele(Profesor):**



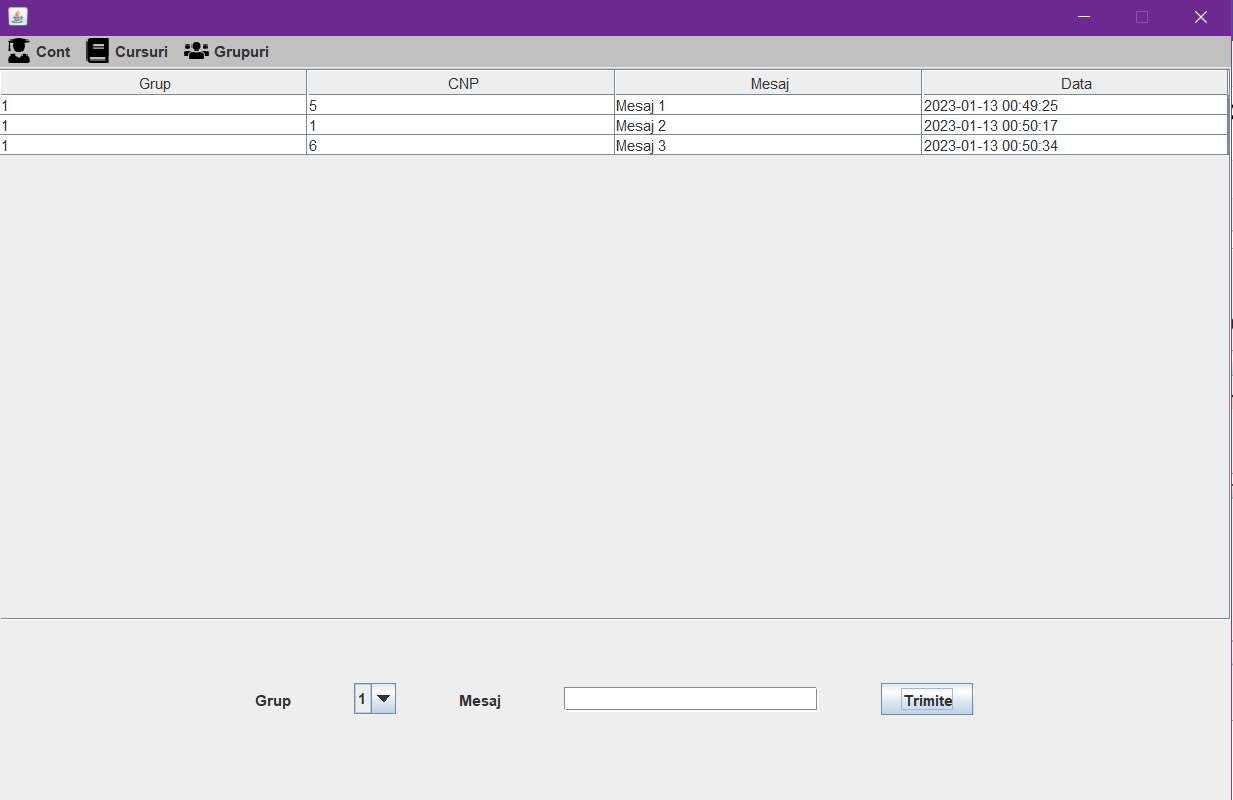
**Programare activitati(Profesor):**



**Creare Grup(Student):**



**Chat Grup(Student):**



# Idei de adaugat pe viitor

Am putea adauga o noua functie prin care sa-ti alegi un avatar, fie sa ai un anumit set de avatare, fie sa iti urci propria imagine. De asemenea s-ar putea adauga un sistem de bonusuri pentru nota finala la examen cu anumite taskuri pe care le primesti direct in interfata de catre profesor. Posibilitate de acceptare invitatie in grup.