

MINISTERUL EDUCAȚIEI



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICA SI CALCULATOARE

DEPARTAMENTUL CALCULATOARE

APLICATIE DESTINATA UNEI INSTITUTII DE STUDIU

Proiect la

Baze de Date

Ilovan Mara Gabriela

Iosif Marian Andrei

Grupa 30222

An de studiu 2022- 2023

CUPRINS

1. Introducere

- Introducere, Argumente, Scop și obiective specifice

2. Analiza Cerințelor Utilizatorilor (Specificațiile De Proiect)

- Ipoteze specifice domeniului ales pentru proiect (Cerințe, Constrângeri)
- Organizare Structurată (Tabelarea Cerințelor)

3. Modelul De Date Și Descrierea Acestuia

- Entitățile și attributele lor
- Diagrama EER/UUMP pentru modelul de date complet
- Procedurile utilizate

4. Detalii De Implementare

- Elemente de utilizare/instalare

5. Concluzii Și Dezvoltări Ulterioare

1. Introducere:

Proiectul realizat consta in dezvoltarea unei aplicații cu interfața grafica pentru gestiunea unei platforme de studiu. Aplicația poate fi folosita de mai multe tipuri de utilizatori, oferind funcționalități ce depind de drepturile pe care aceștia le dețin in sistem. Astfel, cu ajutorul aplicației studenții se pot înscrie la cursuri si grupuri de studiu, pot sa vizualizeze notele si orarul. Profesorii sunt distribuiți la cursuri de către administrator sau super-administrator, pot da note elevilor, sa își programeze orarul si sa îl vizualizeze. Aceștia pot atribui cate un procentaj pentru fiecare activitate(curs, seminar, laborator) din cadrul materiilor la care au fost asignați. De fiecare data când profesorul modifica o nota, nota finala se recalculează. Administratorul poate vedea datele despre orice tip de utilizator, dar si cursuri si poate șterge sau modifica datele unui student si profesor. Super-administratorul păstrează rolurile administratorului, însă acesta poate opera si cu administratorii.

Aceasta interfața a fost creata pentru a facilita comunicarea între studenți si profesor, dar si între studenți, existând grupuri de studiu. Astfel, studenții își vad mult mai ușor notele iar înscrierea la cursuri este simplificata.

Pentru întocmirea proiectului am utilizat MySQL Workbench 8.0 CE si IntelliJ. Pentru crearea bazei de date, popularea acesteia si crearea unor proceduri am utilizat MySQL Workbench, iar IntelliJ pentru realizarea conexiunii cu baza de date si interfața grafica a aplicației.

2. Analiza cerințelor utilizatorilor

Ipoteze specifice:

Aplicația este făcută pentru patru tipuri de utilizator: super-administrator, administrator, profesor si student. Fiecare utilizator primește cate un id (unic) la înregistrare, in baza de date sunt reținute informații despre aceștia precum: CNP, nume, prenume, număr de telefon, email (folosit pentru conectare), adresa, IBAN. Totodată, aceștia primesc si un număr de contract.

Materiile sunt predate de unul sau mai mulți profesori. Fiecare materie poate avea activitățile: curs, seminar, laborator, o descriere, o data de început și una de final. Fiecare tip de activitate din cadrul fiecărei materii are un număr maxim de participanți, un procentaj (adăugat de profesorul care predă acea activitate), o zi și ora la care are loc, dar și durata.

Studentii au posibilitatea de a se înscrie la una sau mai multe materii, fiind distribuit la fiecare dintre activitățile acestuia, dacă mai sunt locuri disponibile. Profesorii îi notează, iar media notelor obținute la activități este calculată automat la fiecare modificare a unei note de la materia respectivă. Totodată, profesorii își pot programa cursurile și pot specifica numărul maxim de studenți.

Administratorii pot căuta utilizatori, pot asigna profesorii la curs și pot modifica datele despre orice tip de utilizator. Super-administratorul are aceleași roluri, dar poate opera și cu utilizatorii de tip administrator.

Organizarea structurală a cerințelor:

În cadrul bazei de date avem 14 tabele, care stochează informații despre:

- Utilizatori: studenți, profesori, administratori și super-administratori
- Materiile din cadrul instituției de studiu, și date despre acestea
- Activitățile din cadrul unei materii și când au loc acestea
- Grupuri de studiu
- Activitățile din cadrul grupurilor de studiu

Interfața realizată ne permite interogarea bazei de date, astfel, cu ajutorul procedurilor realizate în SQL și limbajului de nivel înalt Java, având următoarele opțiuni:

- Înregistrare (adăugare)
- Modificare date (update)
- Ștergere
- Asignarea unui profesor la curs (inserare)
- Conectare
- Adăugare de note unui student (update)
- Adăugare procentaj și programarea activităților (update)
- Înscriere la curs (inserare)
- Creare grup de studiu (inserare)
- Vizualizare orar

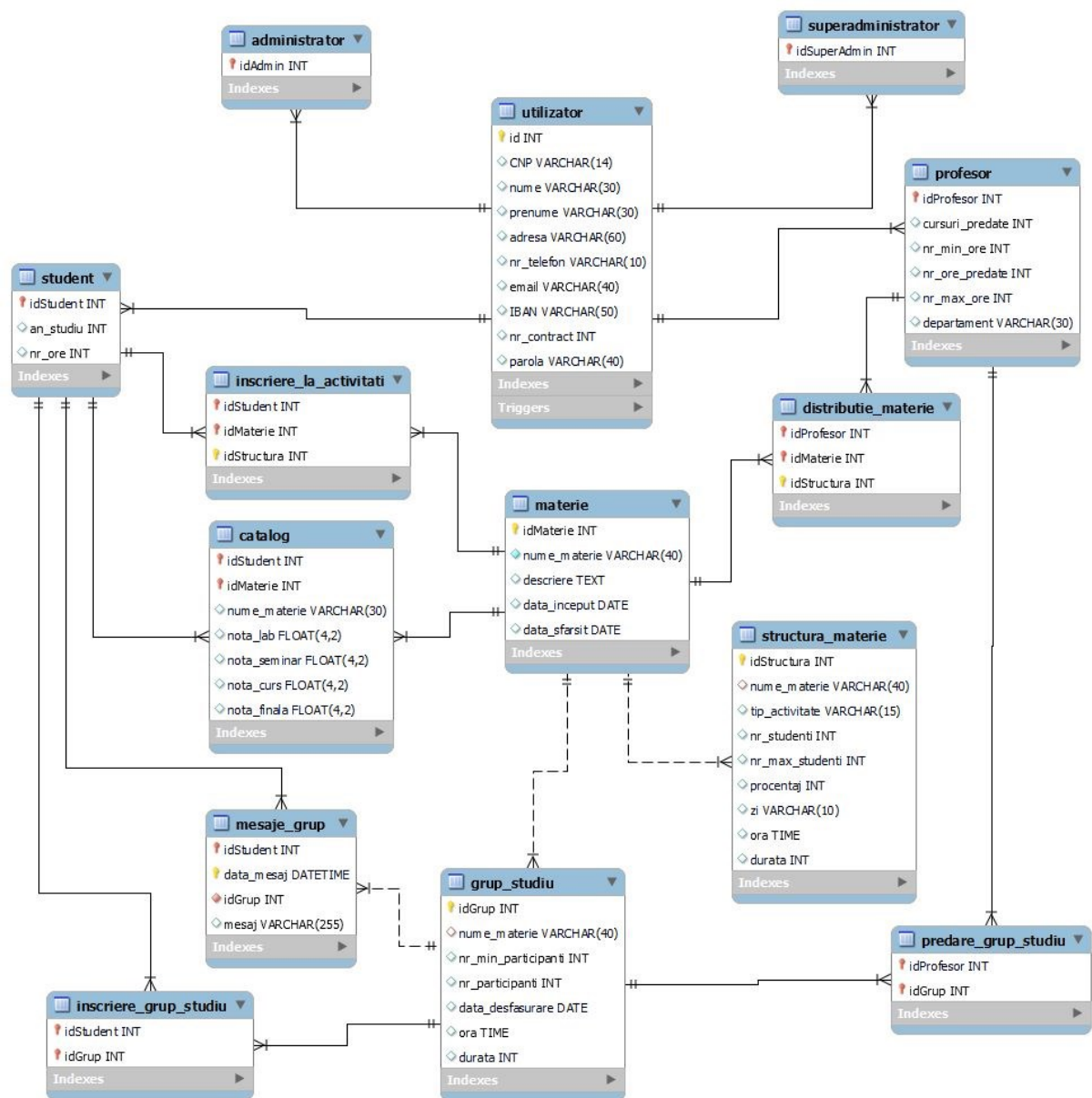
- Liste studenți cursuri/ grupuri de studiu

3. Modelul de date si descrierea acestuia:

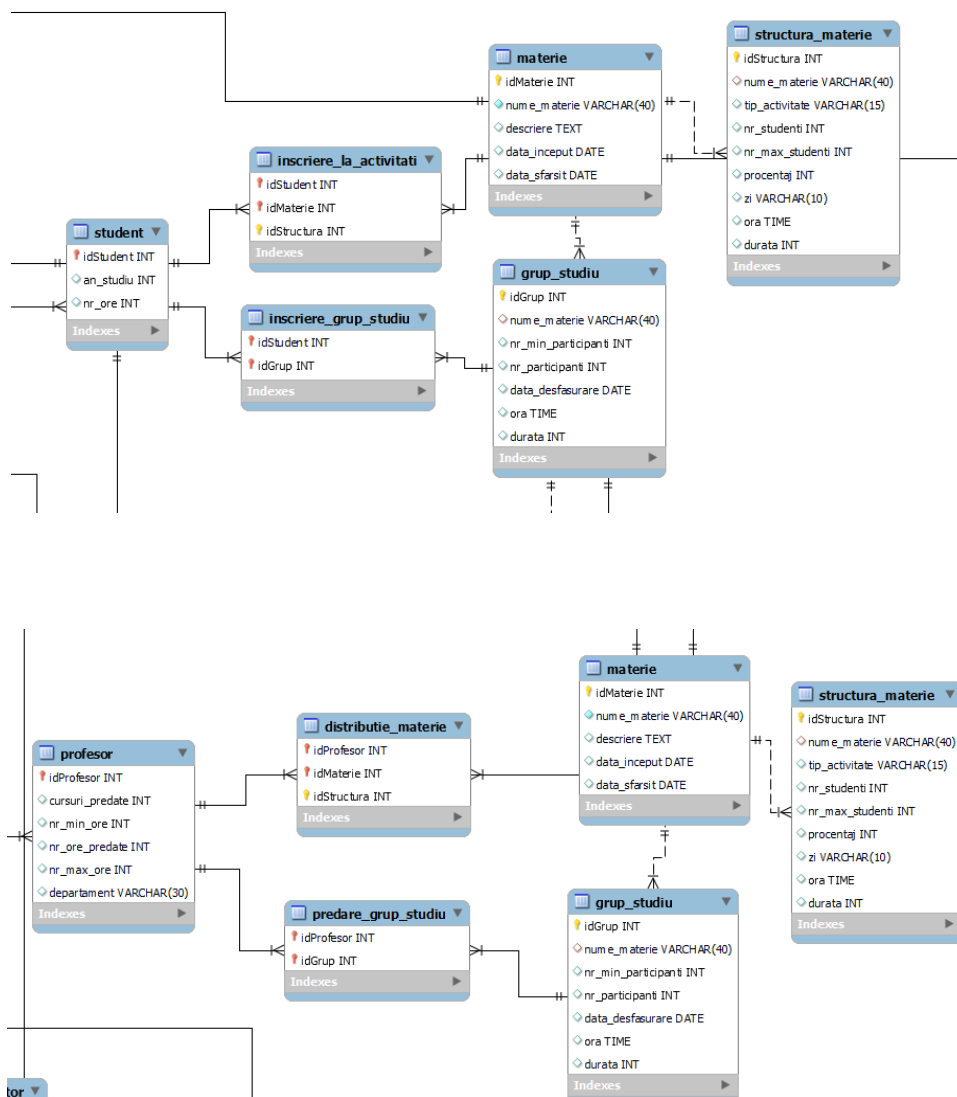
Tabelele din cadrul bazei de date stochează următoarele informații:

- Utilizator: id, CNP, nume, prenume, adresa, nr de telefon, email, IBAN, număr de contract si parola pentru contul in aplicație
- Profesor: id, cursuri predate, numărul minim de ore, numărul de ore predate, numărul maxim de ore si departamentul din care face parte
- Student: id, an de studiu si numărul de ore la care participa săptămânal
- Materie: id, numele materiei, descriere, data la care încep cursurile si data la care se încheie
- Activități (structura materiei): id, numele materiei, tip de activitate, numărul de studenți care participa la activitate, numărul maxim de participanți, procentaj, zi, ora, durata
- Catalog: id-ul studentului si al materiei la care s-a înscris, numele materiei, notele la acea disciplina pentru fiecare activitate si nota finala
- Grup de studiu: id, numele materiei, număr de participanți, numărul minim de participanți, data desfășurării, ora si durata
- Mesaje de grup: id-ul studentului, mesajul, data si id-ul grupului pentru care mesajul e valabil

Totodată, au mai fost nevoie de câteva tabele intermediare pentru realizarea legăturilor many to many.



Legăturile mai importante sunt cele many to many realizate prin intermediul tabelelor `distributie_materie` si `inscriere_la_activitati`:



Pentru interogarea bazei de date am construit mai multe proceduri, care ne ajuta sa operam cu baza de date din interfața grafica. De asemenea, am interogat baza de date si din mediul de dezvoltare Java, IntelliJ pentru a extrage informațiile dorite. Am folosit un trigger pentru generarea random a numărului de contract.

Astfel am folosit următoarele proceduri:

create procedure inserare_utilizator(CNPp varchar(14), numep varchar(30), prenumep varchar(30), adresap varchar(60), nrtelefon varchar(10), emailp varchar(40), ibanp varchar(50), parolap varchar(40))

- Pentru inserare de utilizatori, folosită la popularea bazei de date si la înregistrare in aplicație.

create procedure inserare_student(idstudent int, an_studiup int)

- Pentru inserarea studenților, folosita la popularea bazei de date si la înregistrare in aplicație.

create procedure inserare_administrator(idAdmin int)

- Pentru inserarea administratorilor, folosita la popularea bazei de date si la înregistrare in aplicație.

create procedure inserare_superadministrator(idAdmin int)

- Pentru inserarea super-administratorilor, folosita la popularea bazei de date si la înregistrare in aplicație.

create procedure inserare_profesor(idprof int, departament varchar(30))

- Pentru inserarea profesorilor, folosita la popularea bazei de date si la înregistrare in aplicație.

create procedure asignare_profesor(idprof int, idmaterie int, curs boolean, seminar boolean, lab boolean, nume varchar(40))

- Pentru asignarea profesorilor la curs de carte administrator

create procedure inserare_materie(numa_materie VARCHAR(40), descriere text, data_inceput date, data_sfarsit date)

- Pentru inserarea unei materii in baza de date, pentru popularea bazei de date

create procedure update_utilizator(idp int, CNPp varchar(14), numep varchar(30), prenumep varchar(30), adresap varchar(60), nrtelefonp varchar(10), emailp varchar(40), ibanp varchar(50), nr_contractp int)

- Pentru modificarea datelor, in cazul in care administratorul dorește sa schimbe datele

create procedure update_student(idp int, an_studiup int)

- Pentru modificarea datelor unui student

create procedure update_profesor(idp int, departamentp varchar(30), nr_min_orep int, nr_max_orep int)

- Pentru modificarea datelor unui profesor

create procedure updatate_structura_materie(idProfp int, idStructp int, procentajp int, nrmaxp int, zip varchar(10), orap time, duratap int)

- Pentru modificarea datelor din cadrul unei activități în cazul în care profesorul programează activitatea sau adaugă procentaj

create procedure inscriere_studenti(idStudentp int, idMateriep int, idStructurap int)

- Pentru înscrierea studenților la cursuri

create procedure notafinala(idStudentp int, idMateriep int, procentajcus int, procentajsem int, procentajlab int)

- Pentru calcularea notei finale, în momentul introducerii de note în catalog

create procedure renuntare_studenti(idStudentp int, idMateriep int)

- Pentru studenții care doresc să renunțe la un curs

create procedure reactualizare_structura(idStudentp int, idStructurap int)

- Folosita în cazul în care un student renunță la curs

create procedure creare_activitate(idStudentp int, nume_materiep varchar(40), nr_min_p int, datap date, orap time, duratap int)

- Crearea unei activități de grup de către un student

create procedure inscriere_activitate_grup(idStudentp int, nume_materiep varchar(40))

- Pentru studenții care doresc să participe la activități de grup create de alți studenți

create procedure renuntare_activitate_grup(idStudentp int, nume_materiep varchar(40))

- Folosita la renunțarea la activități de grup

create procedure predare_grup(idProfesorp int, nume_materiep varchar(40))

- Pentru asignare profesor la grup de studiu

4. Detalii de implementare

Nicio aplicație nu si-ar putea arata funcționalitatea fără o interfață grafică care sa faciliteze accesul utilizatorului la datele si rolul acestuia. Astfel, proiectul nostru implementează o interfață grafică realizată în Java care operează cu o baza de date realizată în MySQL Workbench.

Pentru realizarea proiectului am avut nevoie de conectarea din mediul de lucru Java la baza de date, acest fapt a fost realizat cu ajutorul conectorului pus la dispoziție de către MySQL. Astfel am reușit să implementăm o platformă de studiu cu o mulțime de funcționalități, care diferă în funcție de tipul de utilizator. Profesorii își pot programa activitățile, pot introduce note în cataloage și își pot vizualiza programul, studenții se înscriu la cursuri și grupuri de studiu și își vad orarul, administratorii le pot edita datele.

Toate aceste funcționalități vor fi integrate în cadrul sistemului informatic, sub forma unor meniuri din care poți decide ce rol vrei să folosești, în funcție de ce tip de utilizator este conectat la momentul de timp respectiv.

Instrucțiuni de utilizare:

Când rulăm codul se va deschide o pagină de logare unde avem opțiunea de creare a unui cont sau de logare.

The image displays two overlapping windows from a Java application. The foreground window, titled 'Pagina de inregistrare', contains a 'Register' form with the following fields: Nume, Prenume, CNP, Adresa, Numar de telefon, Cont IBAN, E-Mail, and Parola. Below these fields are radio buttons for 'Student', 'Profesor' (which is selected), and 'Administra...', followed by a 'Departament' field. A 'Sign In' button is at the bottom right of this window. The background window, titled 'Pagina de logare', contains a 'Login page' form with 'E-mail' and 'Parola' fields, a 'Login' button, and a 'Sign in' button with a link 'First time? Create an account'. A 'BACK' button is located at the bottom center of the 'Pagina de inregistrare' window.

Register

Nume:

Prenume:

CNP:

Adresa:

Numar de telefon:

Cont IBAN:

E-Mail:

Parola:

☐ Student ☐ Profesor ☒ Administra...

Astfel, dacă apăsăm pe butonul de Sign In ar trebui să ne redirecționeze către pagina de înregistrare, în care viitorul utilizator ar trebui să își introducă datele. Această pagină are 3 butoane care redeschid alte câmpuri, în cazul în care sunteți: student, profesor sau administrator.

Dacă folosim un cont de administrator pentru logare, ar trebui să ne ducă la pagina acestuia:

Aceasta reprezintă meniul din care putem alege ce dorim să facem ca administrator. Astfel, dacă apăsăm pe butonul de date personale o să apară date despre aparținătorul contului.

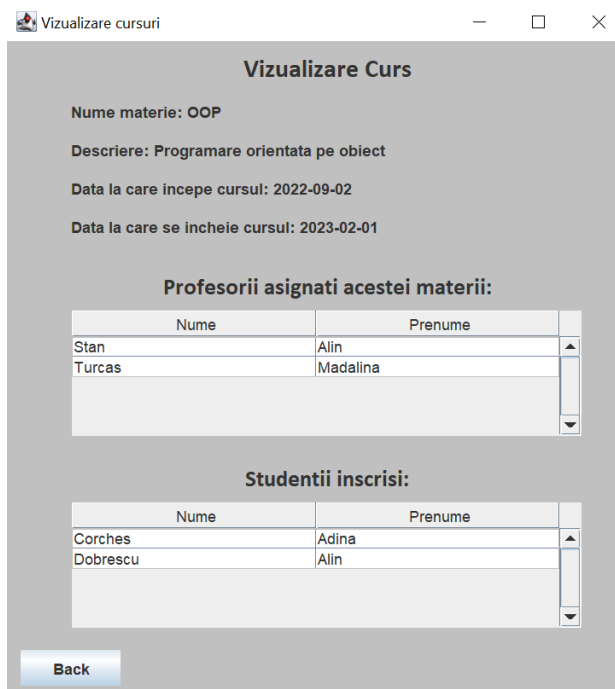
Administrator Page

Cautare cursuri

Materie

Introduceti materia:

Aici putem căuta un curs si ne vor apărea date despre acesta precum profesorii asignați materiei si studenții înscriși la aceasta.



Vizualizare Curs

Nume materie: OOP

Descriere: Programare orientata pe obiect

Data la care incepe cursul: 2022-09-02

Data la care se incheie cursul: 2023-02-01

Profesorii asignati acestei materii:

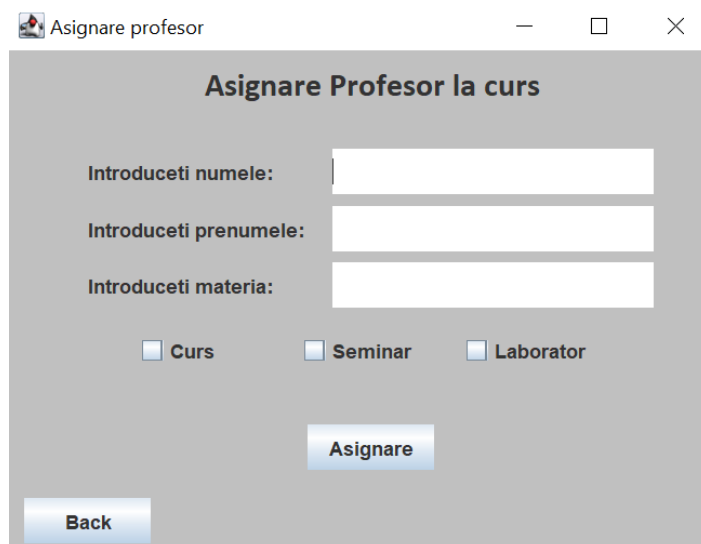
Nume	Prenume
Stan	Alin
Turcas	Madalina

Studentii inscristi:

Nume	Prenume
Corches	Adina
Dobrescu	Alin

Back

Daca apăsăm pe butonul vizualizare utilizatori o sa se deschidă o pagina asemănătoare cu cea de la cursuri in care putem caută studenți, căutându-i aflăm datele acestora. Butonul de modificare date ne permite modificarea datelor oricărui tip de utilizator. Iar, in final, butonul pentru asignare la curs a profesorilor care cu ajutorul numelui si prenumelui, dar si a numelui cursului la care vrem sa îl punem responsabil, este asignat.



Asignare Profesor la curs

Introduceti numele:

Introduceti prenumele:

Introduceti materia:

☐ Curs ☐ Seminar ☐ Laborator

Asignare

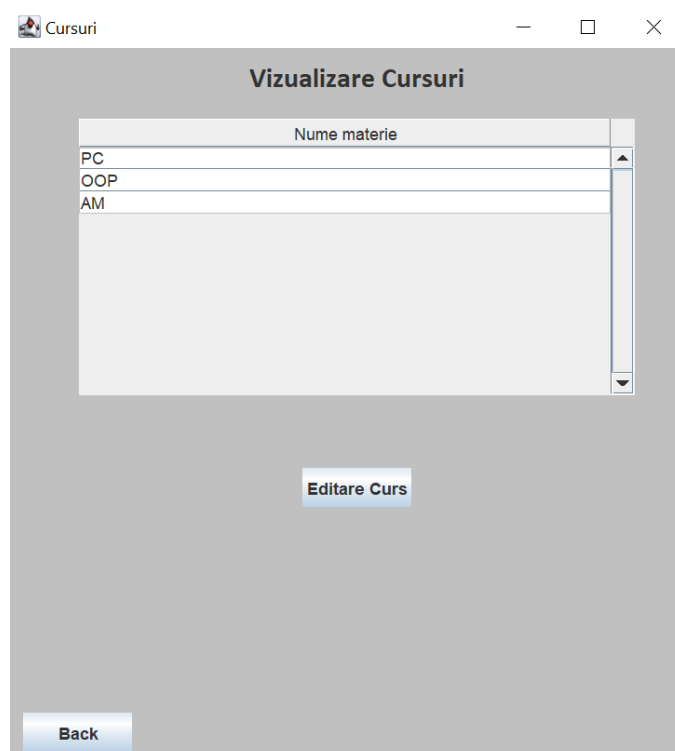
Back

Daca ne conectăm ca un super-administrator, avem același meniu, ce diferă este faptul ca putem modifica si datele administratorilor.


În schimb, dacă ne conectăm ca profesor pagina de început va avea următorul meniu:



Butonul de date personale, ca în cazul administratorului, ne va redirecționa la pagina care ne permite să ne vedem datele. Dacă apăsăm pe vizualizare cursuri avem opțiunea de a vedea cursurile la care am fost asignați să predăm, iar dacă butonul de editare din cadrul acestei pagini este apăsător și a fost selectat un curs vom putea programa activitățile și adăuga procentajele lor.



Daca vrem sa accesam catalogul online, se va deschide o pagina cu o lista de studenții cu activitatea si cursul la care participa, iar daca selectam care un rând din tabel vom putea adăuga nota la activitatea respectiva, la studentul respectiv.

 Cursuri

Vizualizare Studenti

id	Nume	Prenume	Materie	Tip activitate
11	Banciu	Matei	PC	laborator
12	Corches	Adina	PC	laborator
12	Corches	Adina	OOP	seminar
13	Dobrescu	Alin	OOP	seminar
12	Corches	Adina	OOP	laborator
13	Dobrescu	Alin	OOP	laborator
13	Dobrescu	Alin	AM	curs
13	Dobrescu	Alin	AM	seminar
13	Dobrescu	Alin	AM	laborator

Editare Note

Back

Daca dorim sa ne vedem orarul apăsăm pe orar si vom vedea:

Orar

Orar Activitati

Ora	Luni	Marti	Miercuri	Joi	Vineri
08:00		OOP / laborator		AM / laborator	
10:00	PC / laborator	AM / seminar			
12:00	OOP / seminar				AM / curs
14:00					
16:00					
18:00					

Back

Iar daca vrem sa vedem ce activități avem in ziua in care accesam pagina vom vedea:

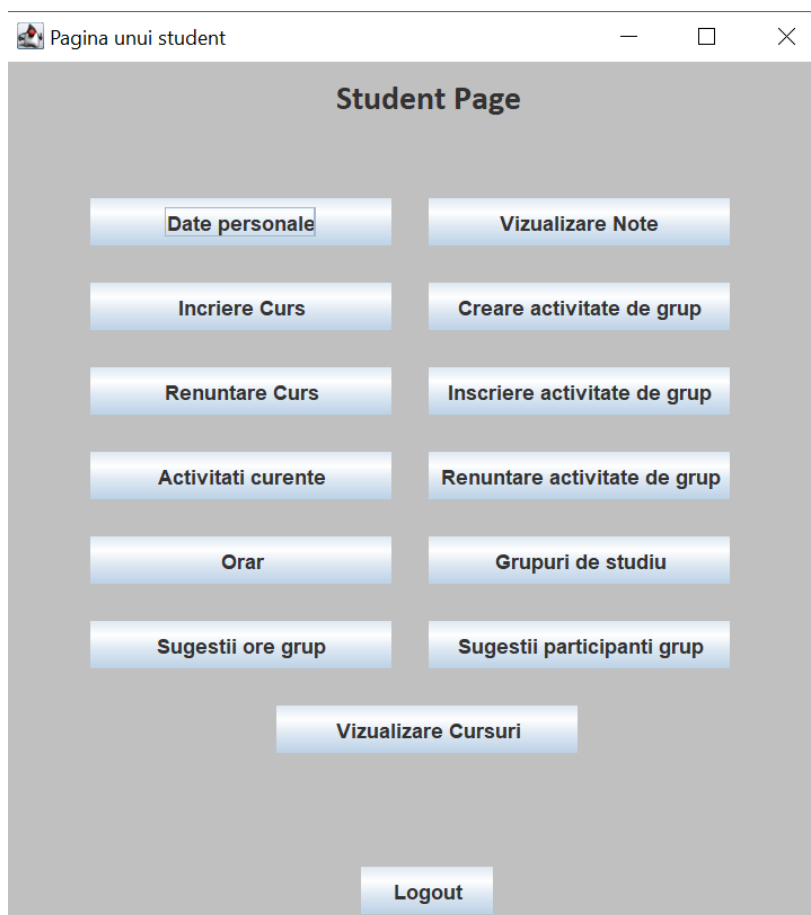
Activitatile de astazi

Activitatile de azi:

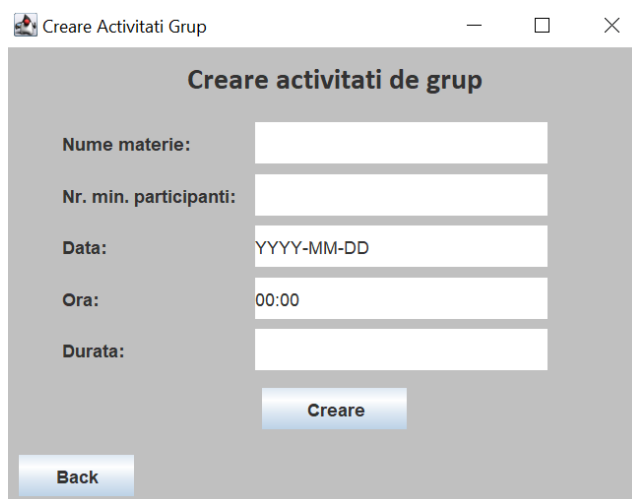
Ore	Astazi
08:00	
10:00	PC / laborator
12:00	OOP / seminar
14:00	
16:00	
18:00	

Back

Daca ne conectam cu contul unui student se va deschide următorul meniu din care vom decide ce funcționalități dorim sa utilizam.

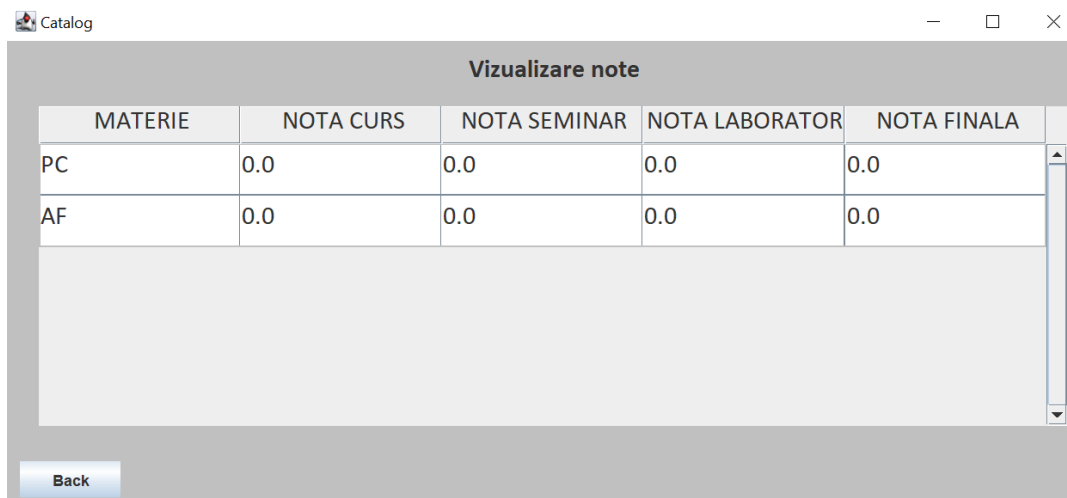


Funcțiile de date personale, orar, activități curente sunt similare cu opțiunile profesorului. Studentul are opțiunea de creare a unui grup de studiu, la o materie la care este înscris. Daca apasă pe butonul Creare activitate grup, acesta creează un grup la care se pot înscrie alți studenți, înscriși la rândul lor la acea materie.



Daca studentul se înscrie la activități, ulterior aceste activități vor apărea si in orar.

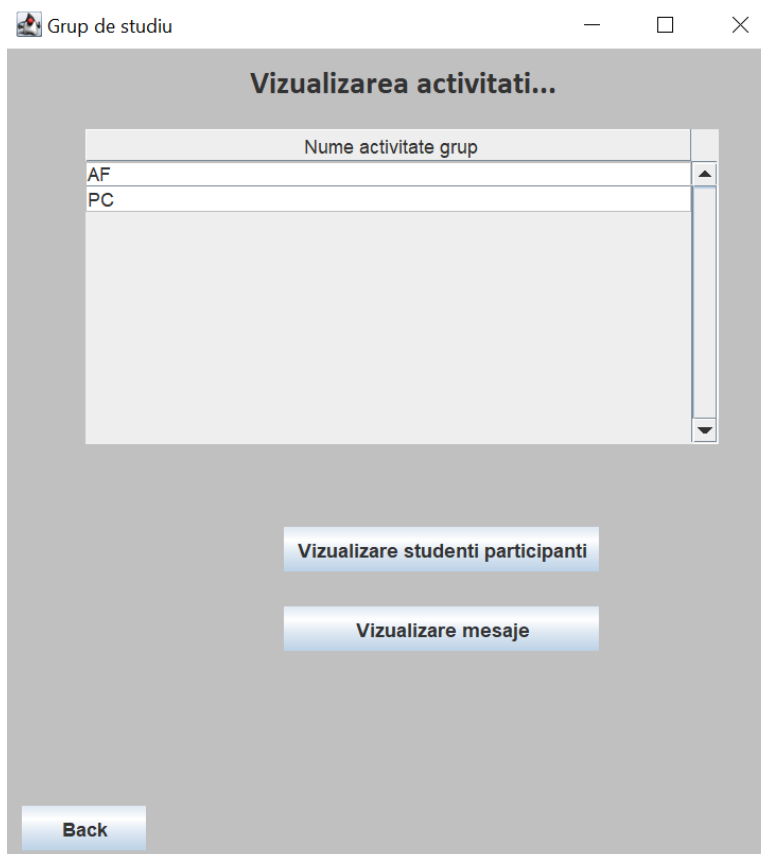
Acesta poate sa își vizualizeze notele:



MATERIE	NOTA CURS	NOTA SEMINAR	NOTA LABORATOR	NOTA FINALA
PC	0.0	0.0	0.0	0.0
AF	0.0	0.0	0.0	0.0

Back

Daca dorește sa vadă grupurile la care este înscris se va deschide pagina:



Nume activitate grup

AF

PC

Vizualizare studenti participanti

Vizualizare mesaje

Back

Selectând un curs, in aceasta pagina vom avea doua opțiuni: vizualizarea studenților participanți la curs si vizualizarea mesajelor lăsate de studenți pe acel grup de studiu.

Grup de studiu

Vizualizare Participanti

Nume	Prenume
Banciu	Matei
Corches	Adina

Back

Grup de studiu

Vizualizare Mesaje

Nume	Prenume	Mesaj	Data mesaj
Banciu	Matei	Salut! Ne vedem cu 10 minute inainte de inceprea activitatii?	2023-01-19 11:47:13
Corches	Adina	Sigur!	2023-01-19 11:48:46

Back

Trimitere mesaj

5. Concluzii si dezvoltări ulterioare

In concluzie, aplicația simulează gestiunea unei platforme de studiu, iar pe viitor se pot adăuga mai multe funcționalități precum: studentul ar putea avea o noua opțiune care sa îl ajute sa își tina evidenta la fiecare materie: astfel acesta poate adăuga sarcini care urmeza sa le facă si sa le bifeze după ce a terminat cu acestea, aceștia ar putea avea opțiunea de evaluare a cadrelor didactice, iar daca profesorul obține o medie mai mica de 5 ar putea fi scos din sistem. Totodată, studenții ar putea avea posibilitatea programării sesiunii, iar ulterior sa le apară, atât lor, cat si profesorilor un calendar ce va cuprinde examenele, data si ora la care urmează sa aibă loc.

Utilizatorii ar putea avea acces la informații privind universitatea. Astfel, la logare aplicația ar putea avea varianta de oaspete care permite vizualizarea de date despre universitate, cursurile care exista aici si câteva informații despre ele.