

|  |
| --- |
| Gestiunea unei platforme de studiu |

Cuprins

[Tema proiectului 3](#_Toc156516964)

[Proiectare conceptuala 4](#_Toc156516965)

[Soluția de transformare în relațional 5](#_Toc156516966)

[Descrierea bazei de date 6](#_Toc156516967)

[Argumentarea nivelului de normalizare 7](#_Toc156516968)

[Codul în SQL 8](#_Toc156516969)

[Operații CRUD 8](#_Toc156516970)

[Funcționalitatea aplicației 9](#_Toc156516971)

[Pachetul Configuration 9](#_Toc156516972)

[Pahetul Databasesc 9](#_Toc156516973)

[Pachetul Repository 10](#_Toc156516974)

[Pachetul Service 11](#_Toc156516975)

[Manual de utilizare 20](#_Toc156516976)

[Dezvoltări ulterioare 21](#_Toc156516977)

# TEMA pROIECTULUI

Aplicația concepută furnizează un sistem informatic pentru gestionarea unei platforme de studiu prin intermediul unei interfețe grafice. Funcționalitatea acesteia acoperă gestionarea studenților, a profesorilor și administrația operațiunilor curente în cadrul programelor de studiu.

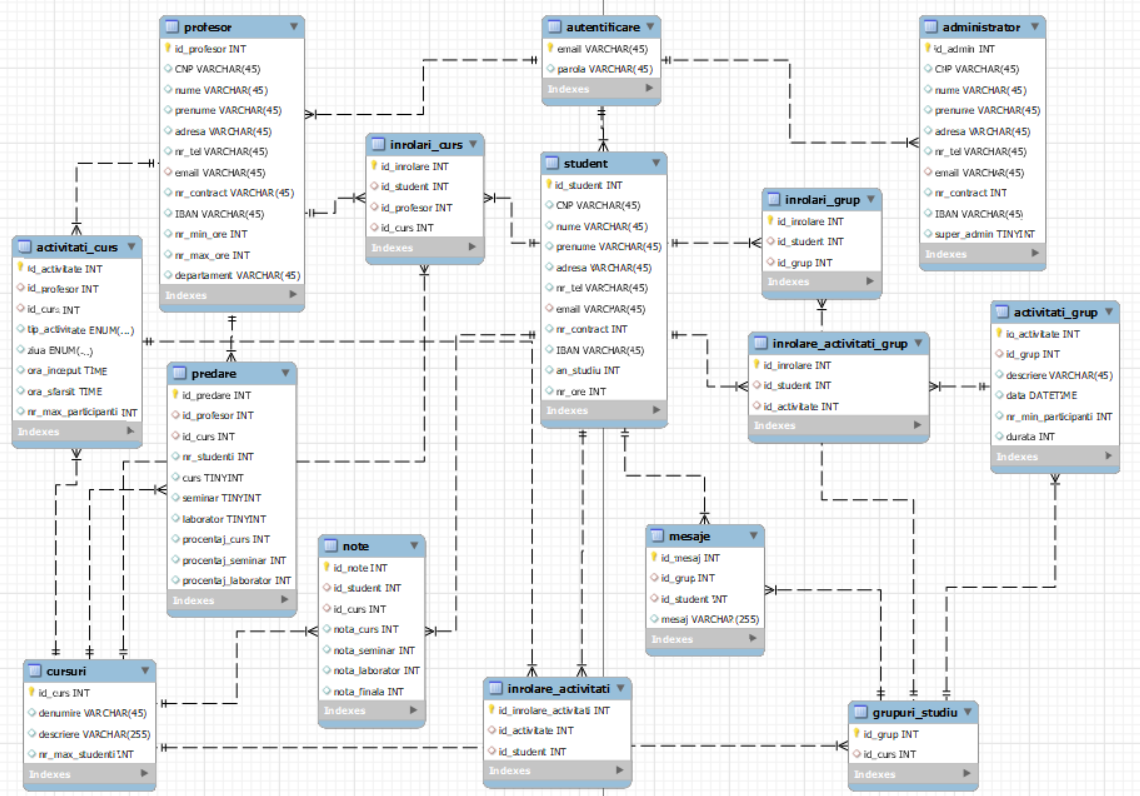
Utilizatorii, inclusiv studenți, profesori și administratori, accesează aplicația prin procesul de autentificare, folosind un email si o parolă. După autentificare, utilizatorii pot vizualiza datele personale, precum CNP, nume, prenume, adresă, număr de telefon, email, cont IBAN și număr de contract. De asemenea, ei au posibilitatea să se deautentifice. În cazul in care utilizatorul este nou pe această platformă, el are opțiunea de a-și creea un cont, în funcție de rolul pe care il are și furnizând datele necesare, amintite și mai sus.

După autentificare, studenții se pot înscrie sau renunța la cursuri și pot vizualiza notele lor. Aceștia pot alege activitățile la care doresc să participe și pot participa doar dacă mai sunt locuri disponibile sau nu există suprapuneri cu alte activități. Studenții au posibilitatea de a se înscrie în grupuri de studiu pentru o anumită materie. Aceștia pot vedea membrii grupului, lăsând mesaje.

Atât studenții cât și profesorii pot accesa o pagină cu toate activitățile la care sunt asignați sau înscriși. Fiecare materie are una sau mai multe activități didactice. Studenții se pot înscrie la cursuri și sunt evaluați prin note pentru fiecare activitate didactică. Media finală a studentului este calculată conform formulei stabilite de profesor. De asemenea, profesorii pot programa activitățile într-un calendar și pot accesa un catalog unde pot filtra studenții după cursuri și pot adăuga note.

Administratorii au atribuții extinse, putând adăuga, modifica și șterge informații despre utilizatori. Există și rolul de super-administrator, care are control exclusiv asupra utilizatorilor de tip administrator, având posibilitatea de a modifica datele acestora. Administratorii pot efectua căutări după nume și pot filtra utilizatorii după tip. Ei pot asigna profesorii la cursuri și pot căuta cursuri după nume. Când se caută un curs, se afișează numele profesorilor și un buton pentru vizualizarea tuturor studenților înscriși la acel curs.

# Proiectare conceptuala



# sOLUțIA DE TRANSFORMARE îN RELAțIONAL

Modelul creat implică 6 relații de tip Many-to-Many, implementate prin adăugarea de tabele intermediare, după cum urmează:

* Un profesor poate preda mai multe cursuri și un curs poate fi predat de mai mulți profesori. Între tabela ,,profesori” și tabela ,,cursuri” am adaugat tabela ,,predare” în care sunt stocate și detalii suplimentare precum numarul de studenți ai cursului, ce activități poate conține cursul (laborator, seminar) și procentajele pentru nota finala
* Un student poate participa la mai multe cursuri și mai multe cursuri pot găzdui mai mulți studenți. Între tabela ,,studenti” și tabela ,,cursuri” am adaugat tabela ,,inrolare\_curs” în care sunt stocate id-ul studentului, al cursului la care participă și al profesorului care predă
* Un student are mai multe note pentru fiecare curs la care este înrolat. Între tabela ,,studenti” și tabela ,,cursuri” am adaugat tabela ,,note” în care sunt stocate notele studentului la activitățile cursului și nota finală la acea materie.
* Un curs poate avea mai multe activități la care participă mai mulți studenți. Între tabela ,,activitati\_curs” și tabela ,,studenti” am adaugat tabela ,,inrolare\_activitati” în care se reține id-ul studentului și al activității la care participă.
* Un student se poate înscrie în mai multe grupuri de studiu și un grup de studiu poate avea mai mulți participanți. Între tabela ,,studenti” și tabela ,,grupuri\_studiu” am adaugat tabela ,,inrolare\_grup” în care sunt stocate id-ul studentului și al grupului din care face parte.
* Un grup de studiu poate organiza mai multe activități și mai mulți studenți pot participa la acestea. Între tabela ,,studenti” și tabela ,,activitati\_grup” am adaugat tabela ,,inrolare\_activitati\_grup” în care sunt stocate id-ul studentului și al activității la care iau parte.

# Descrierea bazei de date

În cadrul aplicației concepute am folosit 15 tabele pentru a reține întreg setul de date necesar unei funcționări optime a bazei de date.

* Autentificare (email, parola)- reține datele de logare ale fiecarui utilizator
* Administrator (id\_admin, CNP, nume, prenume, adresa, nr\_tel, email, nr\_contract, IBAN, super\_admin)- conține detaliile utilizatorilor care au rolul de administrator, respectiv super-administrator, dacă are “1” în câmpul super\_admin
* Student (id\_student, CNP, nume, prenume, adresa, nr\_tel, email, nr\_contract, IBAN, an\_studiu, nr\_ore)- conține detaliile utilizatorilor de tip student
* Profesor (id\_profesor, CNP, nume, prenume, adresa, nr\_tel, email, nr\_contract, IBAN, nr\_min\_ore, nr\_max\_ore, departament)- conține detaliile utilizatorilor care au rol de profesor
* Cursuri (id\_curs, denumire, descriere, nr\_max\_studenti)- conține detalii cu privier la cursuri
* Grupuri\_studiu (id\_grup, id\_cur)- conține grupurile de studiu ale studenților la un anumit curs
* Predare(id\_predare, id\_profesor, id\_curs, nr\_studenti, curs, seminar, laboraator, procentaj\_curs, procentaj\_seminar, procentaj\_laborator)- reține detaliile despre predarea unui anumit curs, de către un profer, care poate selecta procentajele notelor la diferite activități
* Inrolari\_curs(id\_inrolare, id\_student, id\_profesor, id\_curs)- reține înscrierile studenților la un anumit curs
* Activitati\_curs(id\_activitate, id\_profesor, tip\_activitate, ziua, ora\_inceput, ora\_sfarsit, nr\_max\_participanti)- reține detaliile de desfășurare a unei activități la un curs
* Inrolare\_activiatati (id\_inrolare, id\_activitate, id\_student)- reține înscrierile studenților la activități
* Note (id\_note, id\_student, id\_curs, nota\_curs, nota\_laborator, nota\_finala)- reține notele studenților la fiecare dintre activităților care se desfășoara la un anumit curs
* Inrolari\_grup (id\_inrolare, id\_student, id\_grup)- reține înscrierile studenților într-un anumit grup
* Activitati\_grup (id\_activitate, id\_grup, descriere, data, nr\_min\_participanti, durata)- reține detalii legate de activitățile unui anumit grup
* Inrolari\_activitati\_grup(id\_inrolare, id\_student, id\_activitate)- reține înscrierile studenților la o anumită activitate, dintr-un anumit grup
* Mesaje(id\_mesaj, id\_grup, id\_student, mesaj)- stochează mesajele dintre studenții înscriși într-un grup de studiu.

# Argumentarea nivelului de normalizare

Normalizarea bazei de date este un proces de proiectare al structurii relaționale, având ca obiectiv eliminarea redundanțelor și menținerea consistenței datelor.

Pentru baza noastră de date am abordat utilizarea cheilor primare și a cheilor străine, asigurând astfel integritatea referențială. De asemenea am creat o tabelă separate pentru Autentificare, ceea ce este considerat a fi o bună practică pentru gestionarea informațiilor de autentificare.

De asemenea, în cazul bazei de date create, există o serie de aspecte care îndeplinesc condițiile de normalizare:

1. Prima Formă Normală (1NF):

* Toate tabelele îndeplinesc această conditie, deoarece coloanele conțin valori atomice și nu există valori multiple în aceleași celule.

1. A Doua Formă Normală (2NF):

* Toate coloanele non-cheie depind complet de cheia primară.

1. A Treia Formă Normală (3NF):

* Informațiile sunt legate în mod direct prin chei-primare, eliminând astfel dependențele tranzitive între coloanele non-cheie.

# Codul în sql

Utilizatorii aplicației noastre se impart în patru tipuri: studenți, profesori, amdinistratori și super-administratori. Aceștia pot îndeplini diferite funcționalități, având rank-uri diferite. Cu toții au în comun posibilitatea de vizualizare a propriilor date personale, însă tipul de admin este singurul care poate modifica datele și de a șterge un utilizator de tip student sau profesor, în timp ce super-adminul are privilegii maxime și poate modifica și șterge și utilizatori de tip administrator.

Utilizatorul de tip student are funcționalități precum: înrolarea sau renunțarea la un curs; vizualizarea notelor de la un curs; vizualizarea, înscrierea sau renunțarea la activități din cadrul unui curs, precum si vizualizarea activităților la care este înscris; vizualizarea, înscrierea sau renunțarea într-un grup de studiu și la activități organizate de acel grup, precum și schimbul de mesaje între membrii.

Utilizatorul de tip profesor are funcționalități precum: vizualizare activități, programare activități, setare procenjtaje notă pentru diferitele activități ale cursului (laborator, seminar, curs), accesare catalog unde pot filtra studenții după curs și le pot adăuga note.

Utilizatorul de tip administrator are funcționalități precum: căutarea utilizatorilor dupa nume, filtrarea tuturor utilizatorilor după tip, căutarea unui curs și asignarea profesorilor la acesta, vizualizarea tuturor studenților înscriși la un anumit curs.

Toate aceste funcționalități au fost implementate folosind proceduri specifice, pe care ulterior le-am apelat in Java, pentru a putea lega utilizatorii aplicației de baza de date.

## OPERAȚII CRUD

* **CREATE**- am conceput procedurile de ,,AdaugareAdministraor” , ,, AdaugareProfesor”, ,, AdaugareStudent”, care pot fi apelate de un nou utilizator dornic să folosească aplicația, dar și de ,,InrolareCurs”, ,,InrolareGrupStudiu”, ,,InrolareProfesorLaCurs”, ,,InrolareActivitateGrup”, ,,InrolareStudentLaActivitateCurs”
* **READ**- am conceput procedurile de ,, VizualizareDatePersonale”, ,,VizualizareUtilizatori ”, ,, VizualizareDetaliiCurs”, ,, VizualizareActivitatiStudent”, ,, VizualizareActivitatiGrupStudiu”, ,, VizualizareActivitatiGrupStudent ”, ,, VizualizareActivitatiCurs”, ,, AfisareNoteStudent”
* **UPDATE**- am conceput procedurile de ,, ActualizareDatePersonale”
* **DELETE**- am conceput procedurile de ,, RenuntareCurs”, ,, RenuntareGrupStudiu”, ,, RenuntareActivitateGrup”, ,, RenuntareActivitateCurs”, ,, StergereUtilizatorDupaEmail”,

# Funcționalitatea aplicației

Pentru a lucra cât se poate de eficient, am structurat codul de Java în mai multe pachete:

* + - Configuration- asigură legatura dintre baza de date și platforma de lucru in Java
    - Databases- conține clasele model pentru tabelele din baza de date
    - Repository- conține apelurile procedurilor din baza de date
    - Service- conține interfețele fiecarei file și funcționalități ale aplicației

## PACHETUL CONFIGURATION

Acest pachet conține clasa ConnectConfig care stabilește legătura dintre baza noastră de date din MySQL și aplicația IntelliJ, în care ne-am dezvoltat aplicația de gestionare a platformei de studio.

## PACHETUL DATABASES

În acest pachet se găsesc toate clasele-model utilizate in program:

* + - ActivitatiCurs- conține informații despre activitățile unui curs
    - ActivitatiGrup- conține informații despre activitățile unui grup
    - Administrator- conține date despre utilizatorul de tip Administrator
    - Autentificare- conține detaliile de autentificare
    - Cursuri- conține date despre cursuri
    - GrupStudiu- conține date despre grupurile de studiu
    - InrolariActivitati- conține înscrierile studenților la activități
    - InrolariCurs- conține înscrierile studenților la cursuri
    - InrolariGrup- conține înscrierile studenților în grupuri
    - Mesaje- reține schimbul de mesaje dintre studenții unui grup de studiu
    - Note- reține notele studenților la activitățile unui curs
    - Predare- conține legatura dintre un profesor și cursul pe care îl predă
    - Profesor- conține date despre utilizatorul de tip Profesor
    - Student- conține date despre utilizatorul de tip Student

## PACHETUL REPOSITORY

Acest pachet este foarte important, întrucât conține clasa UtilizatorRepository în care sunt create toate metodele necesare aplicației, care apelează procedurile din MySQL. În total, proiectul conține 37 de proceduri.

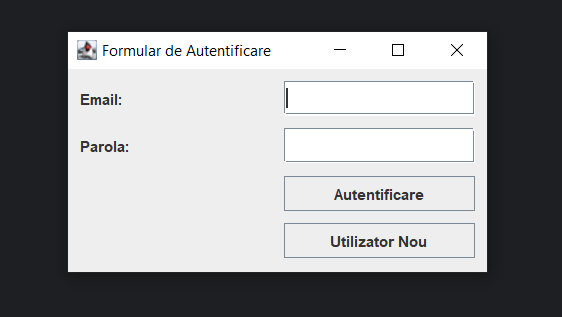
Câteva dintre aceste metode concepute sunt:

* + - *autentificare* apelează procedura *Autentificare*, cu doi parametrii (email, parola) care fac posibilă intrarea în contul utilizatorului
    - *adaugareStudent, adaugareProfesor, adaugareAdministrator*  apelează procedurile *AdaugareStudent*, *AdaugareProfesor* respectiv *AdaugareAdministrator*, având ca parametrii toate detaliile necesare profilului de Student, Profesor respectiv Administrator, și face posibilă adaugarea unui nou utilizator în baza de date
    - *vizualizareDatePersonale* apelează procedura *VizualizareDatePersonale* cu un singur parametru (email) care face posibilă afișarea datelor utilizatorului
    - *inrolareCurs, renuntareCurs* apelează procedurile *IrolareCurs* respectiv *RenuntareCurs* care fac posibilă înscrierea respectiv renunțarea unui curs
    - *calculareNotaFinala* apelează procedura *CalculareNotaFinala* care primește emailul studentului și numele cursului la care se calculează nota finală în funcție de procentajele fiecărei activități
    - *verificaTipUtilizator* apelează procedura VerificaTipUtilizator care primește emailul utilizatorului și returneaza 0 dacă este student, 1 dacă este profesor, 2 dacă este administrator sau 3 dacă este super-administrator.
    - *cautareUtilizator* apelează procedura *CautareUtilizator* care primește numele și prenumele unui utilizator și caută pe rând în bazele de date, aflând rolul acestuia
    - *stergereUtilizatorDupaEmail* apelează procedura *StergereUtilizatorDupaEmail* care primește ca parametrii emailul adminului sau super-adminului, care are posibilitatea de ștergere a utilizatorilor, și emailul persoanei care se dorește ștearsă
    - *inrolareProfesorLaCurs* apelează procedura *InrolareProfesorLaCurs* și face posibilă asignarea unui profesor la un curs, administratorul fiind în măsură să aleagă și activitățile posibile (curs, seminar, laborator)
    - *studentiCuCurs* apelează procedura S*tudentiCuCurs* primește ca parametru numele unui curs și face posibilă afișarea tuturor studenților înrolați la cursul respectiv

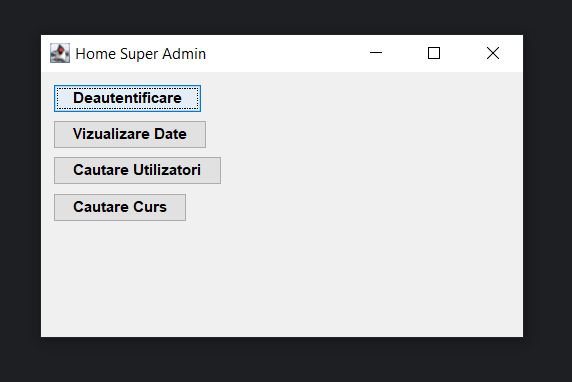
## PACHETUL SERVICE

În acest pachet se găsesc mai multe clase pentru ferestrele din interfața aplicației:

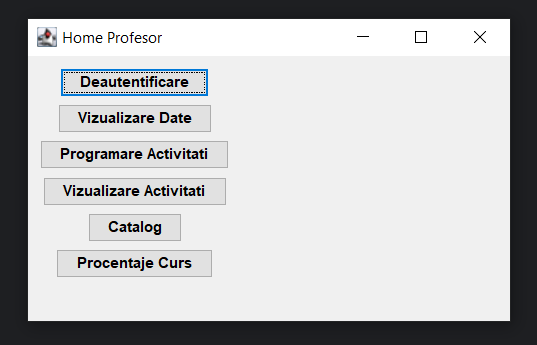
* + - AutentifInterface- acționează o pagină de autentificare care conține câmpuri text pentru introducerea email-ului și a parolei, dar și două butoane, pentru a intra în cont, respectiv pentru a creea un utilzator nou pe platformă



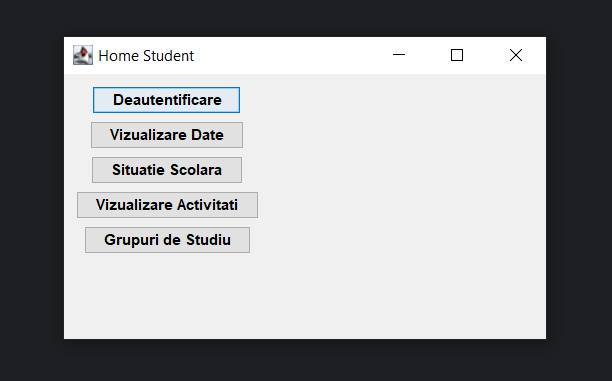
* + - HomeAdmin- este o pagină destinată doar administratorilor și prezintă funționalitățile specifice lor
    - HomeSuperAdmin- este o pagină destinată super-administratorilor și se aseamănă cu HomeAdmin, având câteva implementări suplimentare



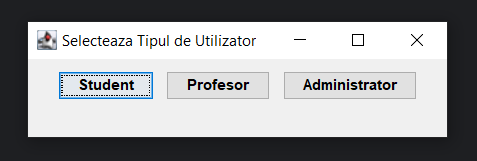
* + - HomeProf- este o pagină destinată doar profesorilor și prezintă funționalitățile specifice lor



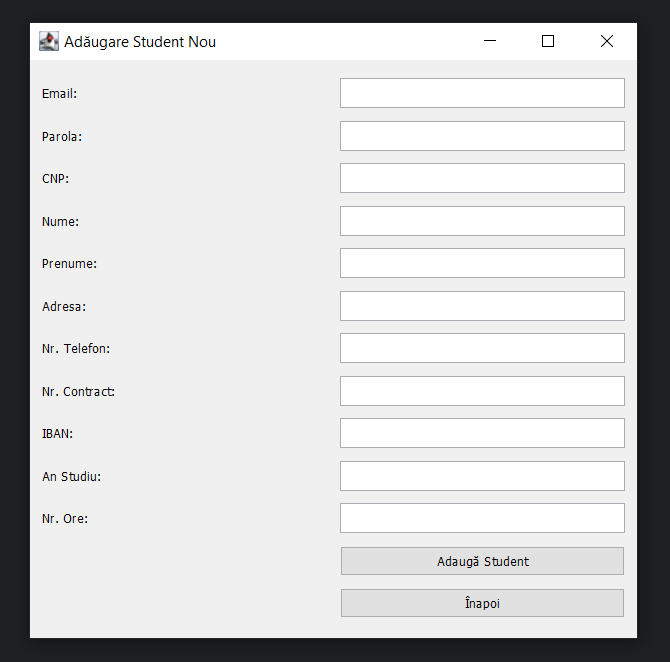
* + - HomeStudent- este o pagină destinată doar studenților și prezintă funționalitățile specifice lor



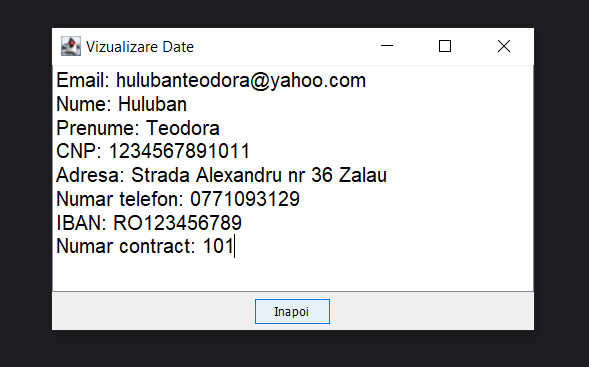
* + - TipUtilizator- este o pagină de selecție din care utilizatorul este pus să aleagă odată cu creearea unui nou cont pe platformă



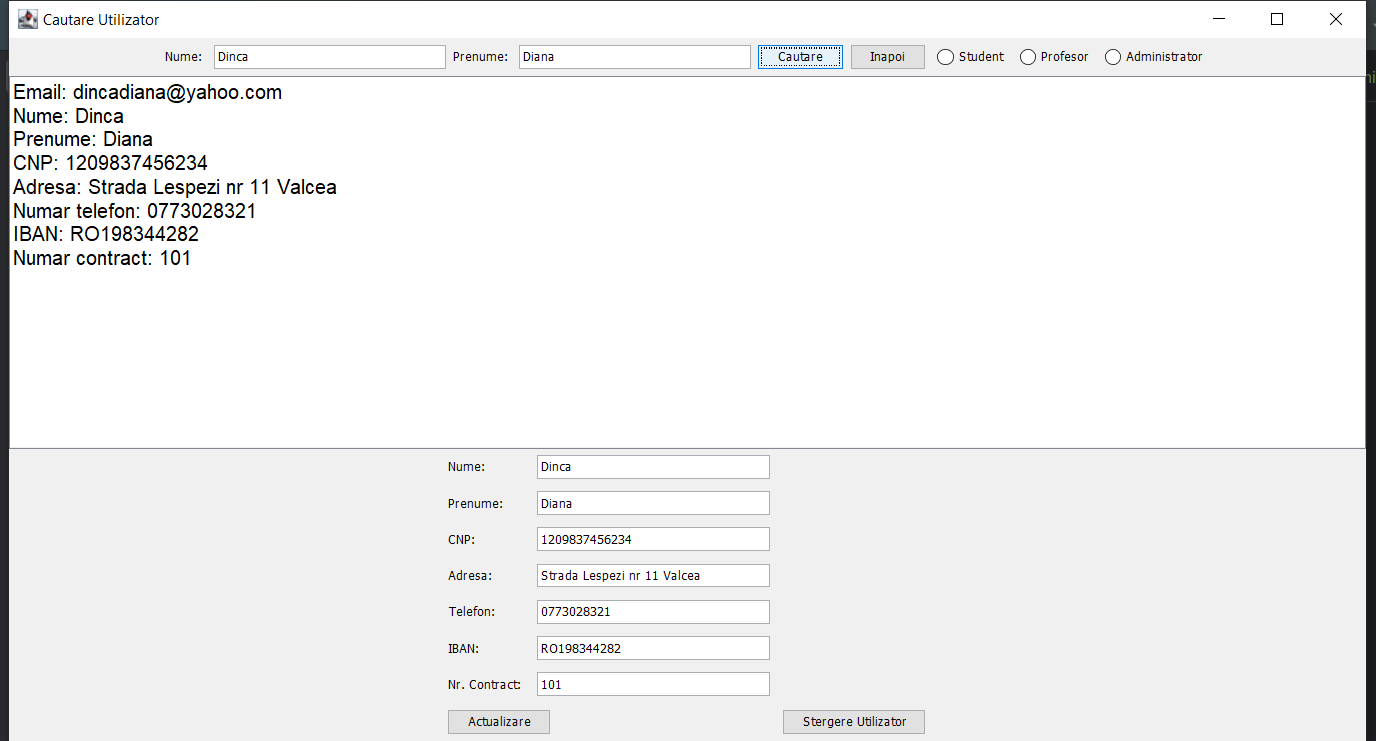
* + - NewAdmin- este o pagină care realizează introducerea datelor pentru un admin nou
    - NewProf- este o pagină care realizează introducerea datelor pentru un profesor nou
    - NewStudent- este o pagină care realizează introducerea datelor pentru un student nou

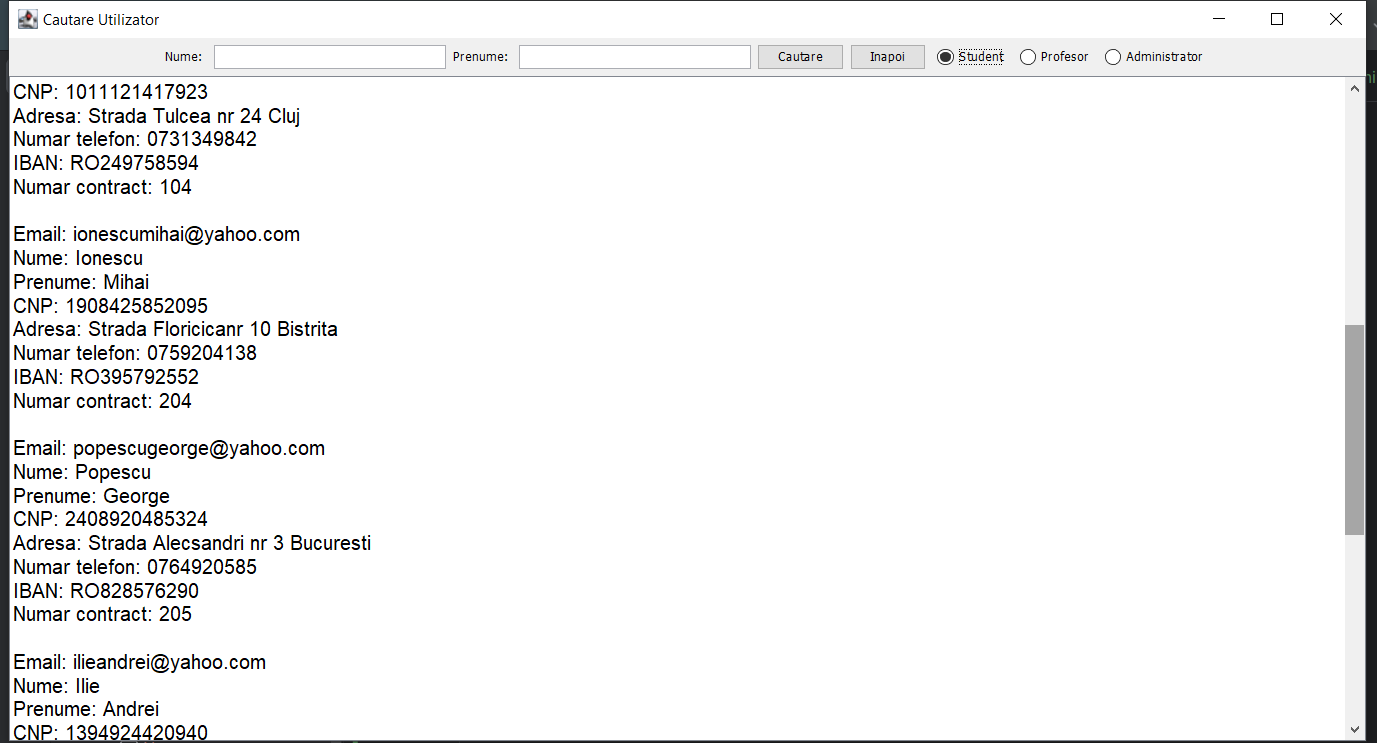


* + - VizualizareDate- este o pagină care se deschide atunci când un utilizator dorește să își acceseze datele personale

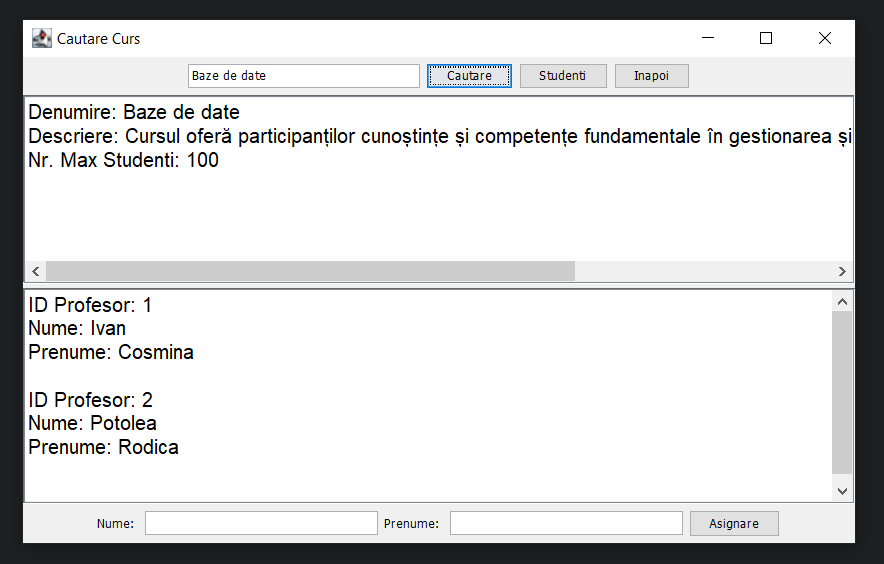


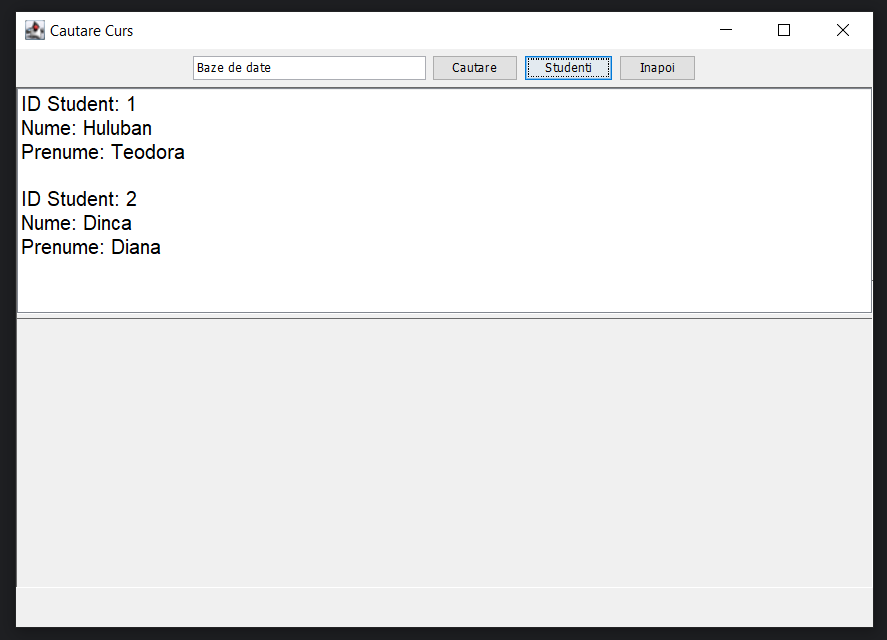
* + - CautareUtilizator- permite unui utilizator, de tip admin sau super-admin, să caute utilizatori după nume sau să filtreze întregii utilizatori ai aplicației după rolul acestor



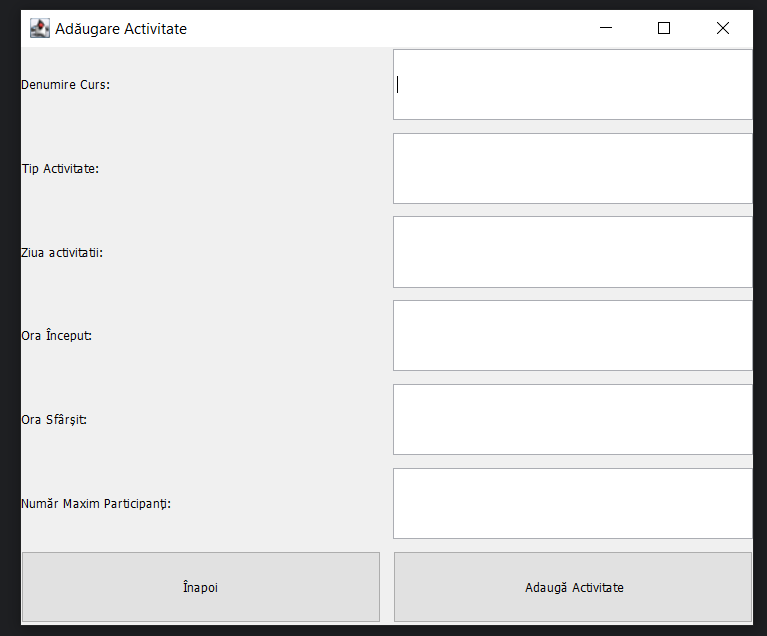


* + - CautareCurs- permite unui utilizator, de tip admin sau super-admin, să caute cursuri după nume, să asigneze profesori la cursul respectiv și să stabilească activitătile acestuia, iar butonul Studenti permite vizualizarea tuturor utilizatorilor de tip student

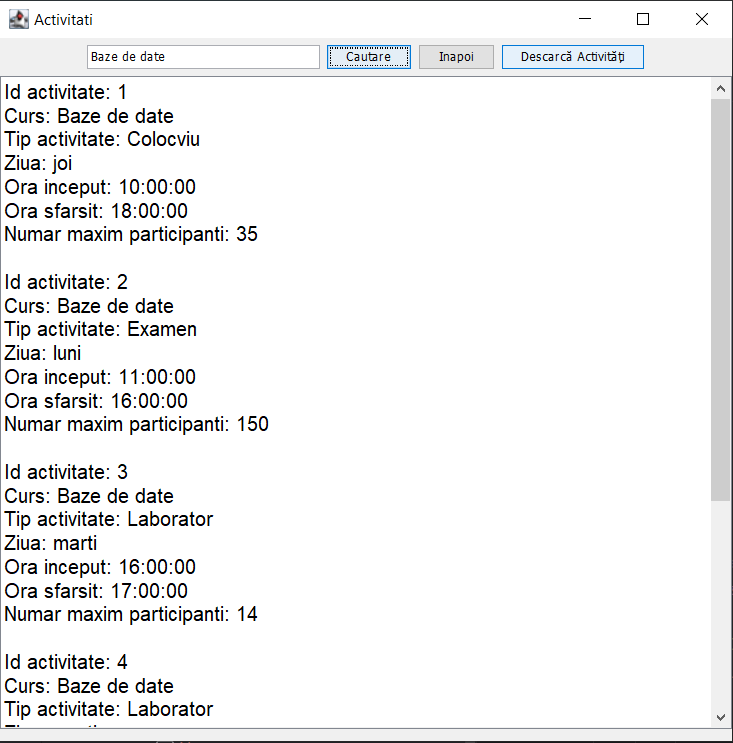




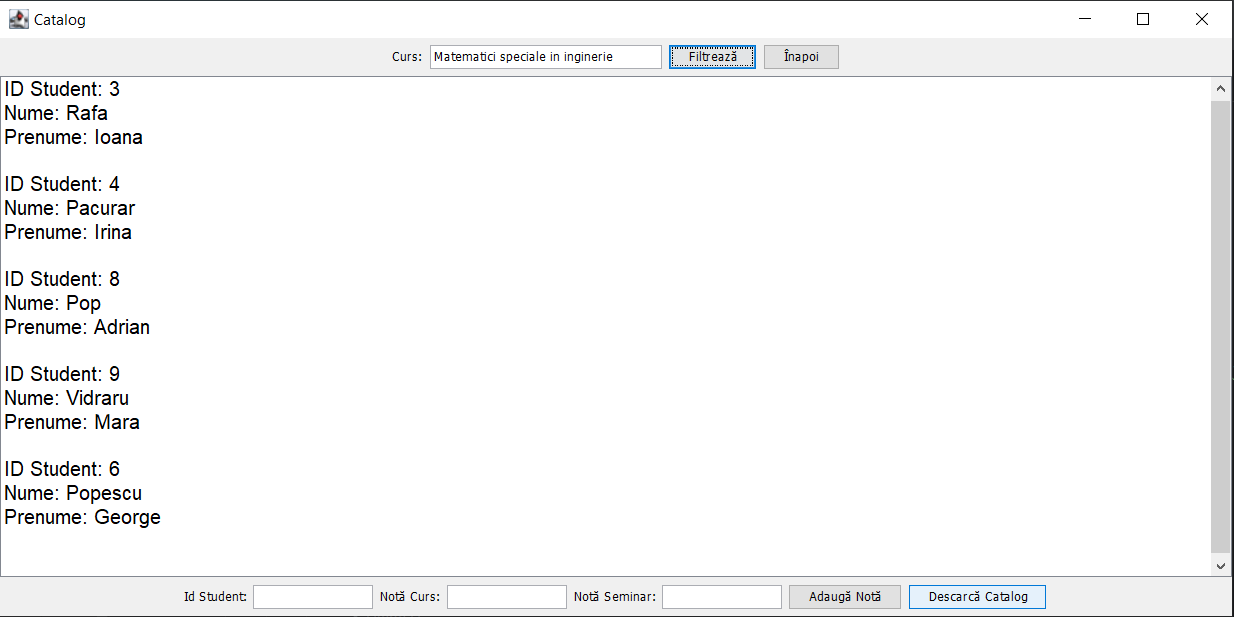
* + - Programare activități- permite unui profesor să introducă detaliile necesare pentru crearea unei activități la cursul său.



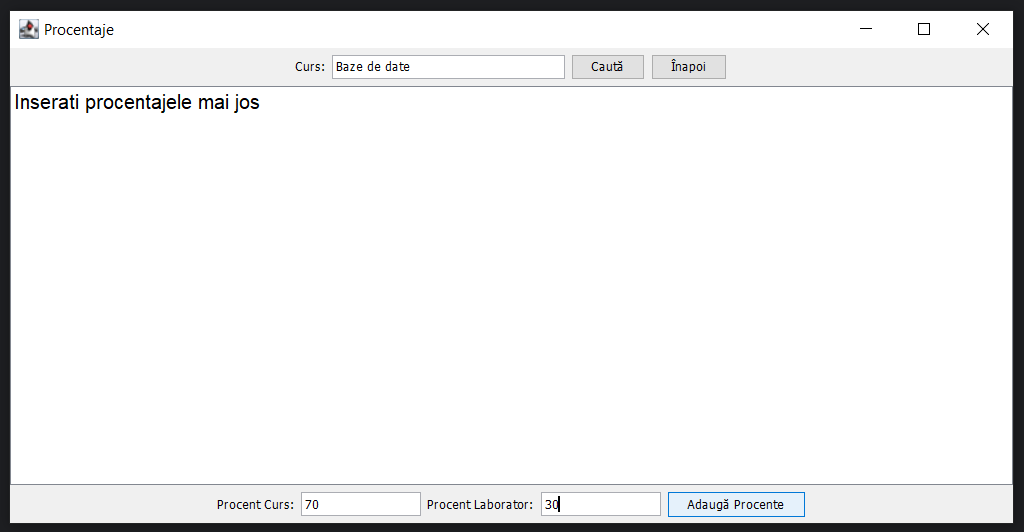
* + - ActivitățiProfsor- îi permite profesorului să țină evidența activităților sale și îi oferă posibilitatea descărcării listei sale de activități



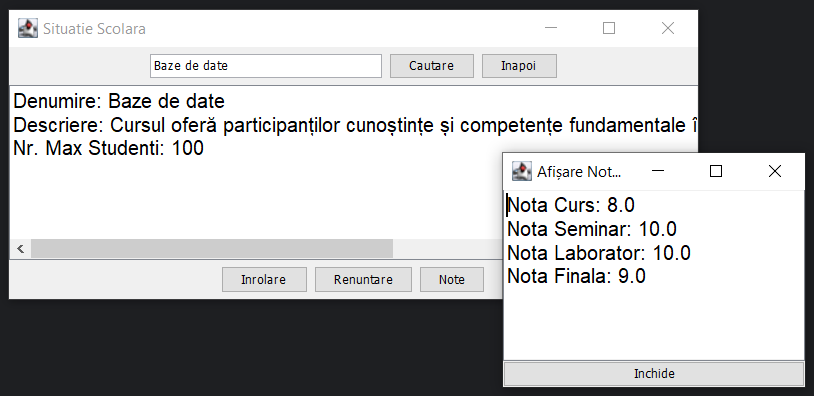
* + - Catalog- permite unui profesor să adauge note studenților înscriși la cursul său, dar și să descarce lista cu numele și id-ul studenților înrolați la acesta



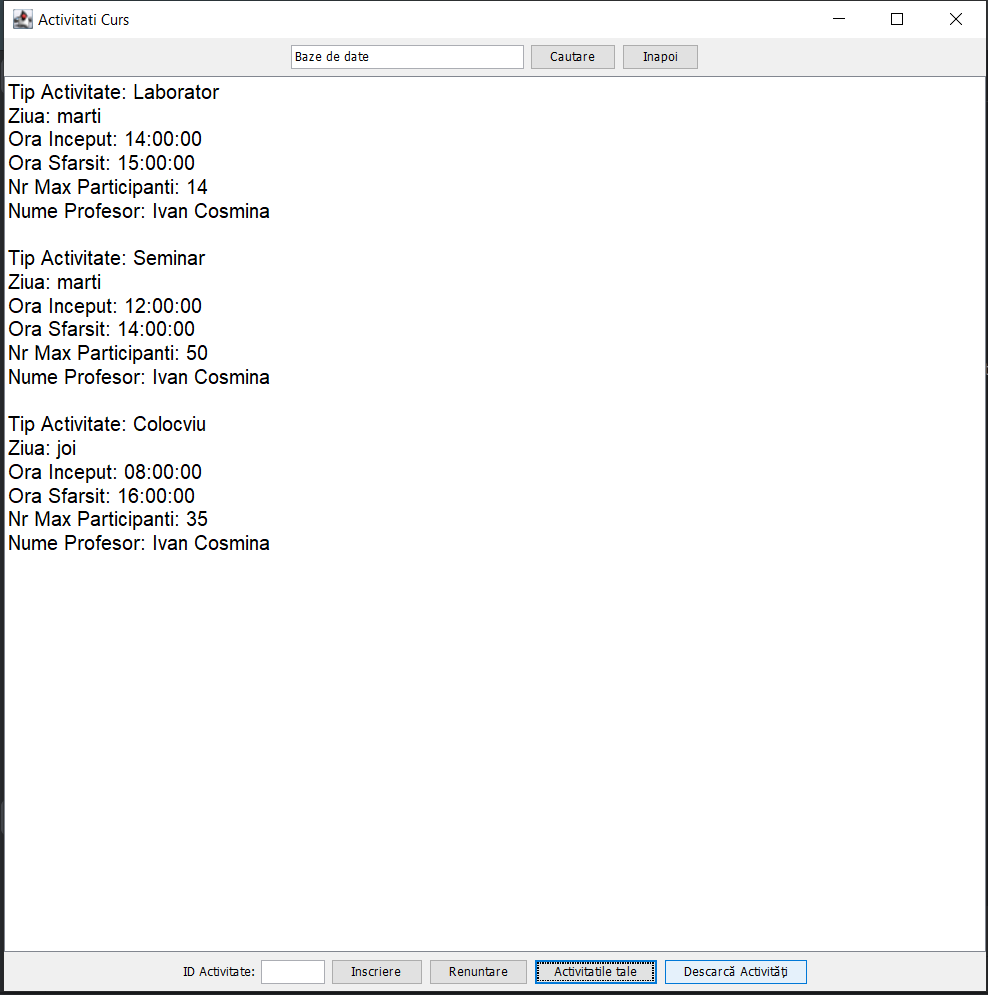
* + - Procentaje curs- permite profesorului să stabilească cât valorează fiecare activitate din cadrul cursului său (curs, laborator, seminar)



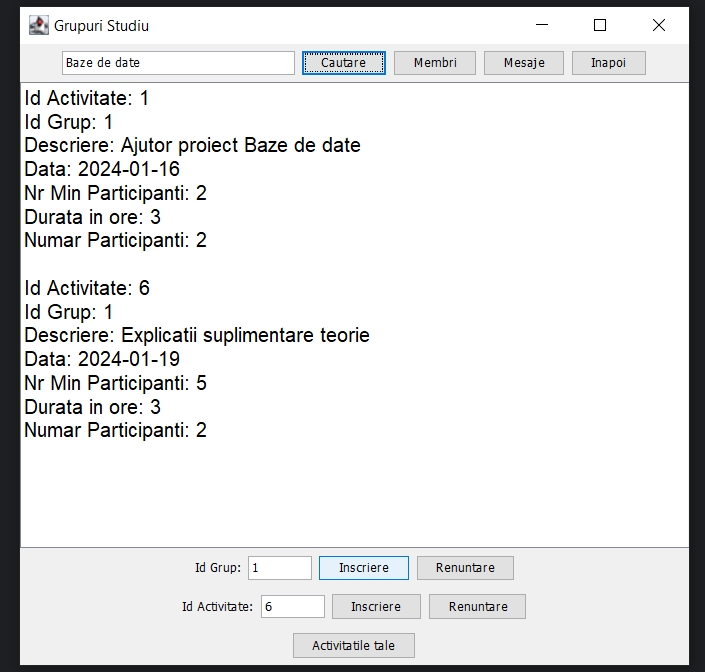
* + - Situatie scolara- ii permite studentului să se înroleze sau să renunțe la un curs, dar și să își vadă notele la activități și nota finală



* + - Vizualizare Activitati- îi oferă studentului posibilitatea de a se înscrie sau de a renunța la o activitate, dar și posibilitatea de a vedea activitățile la care este deja înscris, acesta având la dispoziție și un buton funcțional pentru descărcarea acestora



* + - Grupuri de Studiu- îi permite studentului să ia partes au să renunțe la un grup de studiu și la activitățile organizate, dar are și posibilitatea de a-și vedea activitățile la care este deja înscris



# MANUAL DE UTILIZARE

La pornirea aplicației se deschide fereastra de autentificare. Daca avem deja un cont, completăm câmpurile de email și parolă, altfel apăsăm butonul de “Utilizator Nou” care ne va duce într-o pagină de completare a datelor personale, specifice tipului de utilizator ales anterior: “Student”, “Profesor” sau “Administrator”. După ce a fost creat un cont, se deschide pagina Home.

În pagina Home, utilizatorul are de ales dintre mai multe funcționalități, printr-o simplă apăsare de buton:

1. Vizualizare date, care permite utilizatorului să vadă datele cu caracter personal
2. Situație școlară care permite studentului să caute, să se înroleze sau să renunțe la cursuri, să vadă notele obținute în cadrul cursului, al seminarului, respective al laboratorului și nota finală
3. Vizualizare activități, care permite studentului să caute, să se înscrie sau să renunțe la activitățile din cadrul unui curs
4. Grupuri studiu, care permit studentului să să caute, să se înscrie sau să renunțe la un grup de studiu, dar și să comunice cu alți studenți din cadrul grupului
5. Deautentificare, care permite utilizatorului să iasă din cont și să parească aplicația, acesta revenind la pagina de autentificare

Pe lângă aceste funcționalități, aplicația prezintă si butoane de “Înapoi”, care trimit utilizatorul la pagina anterioară.

# DEzvoltări ULteRioare

Aplicția noastră oferă o varietate de funcționalități și un design simplu și intuitiv. Utilizarea este usoară, cu butoane sugestive, fiind perfecta pentru utilizarea într-un cadru de învățământ. Pentru a îmbunătăți performanța aplicației și a permite accesul simultan al mai multor utilizatori în timp real, ar fi nevoie de gazduirea bazei de date pe un server.

De asemenea, există posibilitatea de a adăuga noi funcționalități pentru a îmbogăți experiența utilizatorilor. Acestea includ capacitatea de a trimite imagini în grup, organizarea întâlnirilor sub formă de video sau audio, personalizarea interfeței grafice, criptarea parolelor și a mesajelor, precum și opțiunea de recuperare a parolei prin intermediul email-ului.

În continuarea optimizărilor și îmbunătățirilor propuse pentru aplicație, se pot considera și integrarea unui sistem de notificări în timp real, pentru a ține utilizatorii la curent cu evenimentele importante, cum ar fi actualizări de cursuri, de note, mesaje primite sau schimbări în programul academic. O altă idee pentru îmbunătățirea serviciului ar putea fi integrarea a altor platforme și instrumente educaționale populare, precum biblioteci online, resurse multimedia, sau chiar platforme de e-learning, pentru a extinde gama de materiale disponibile.