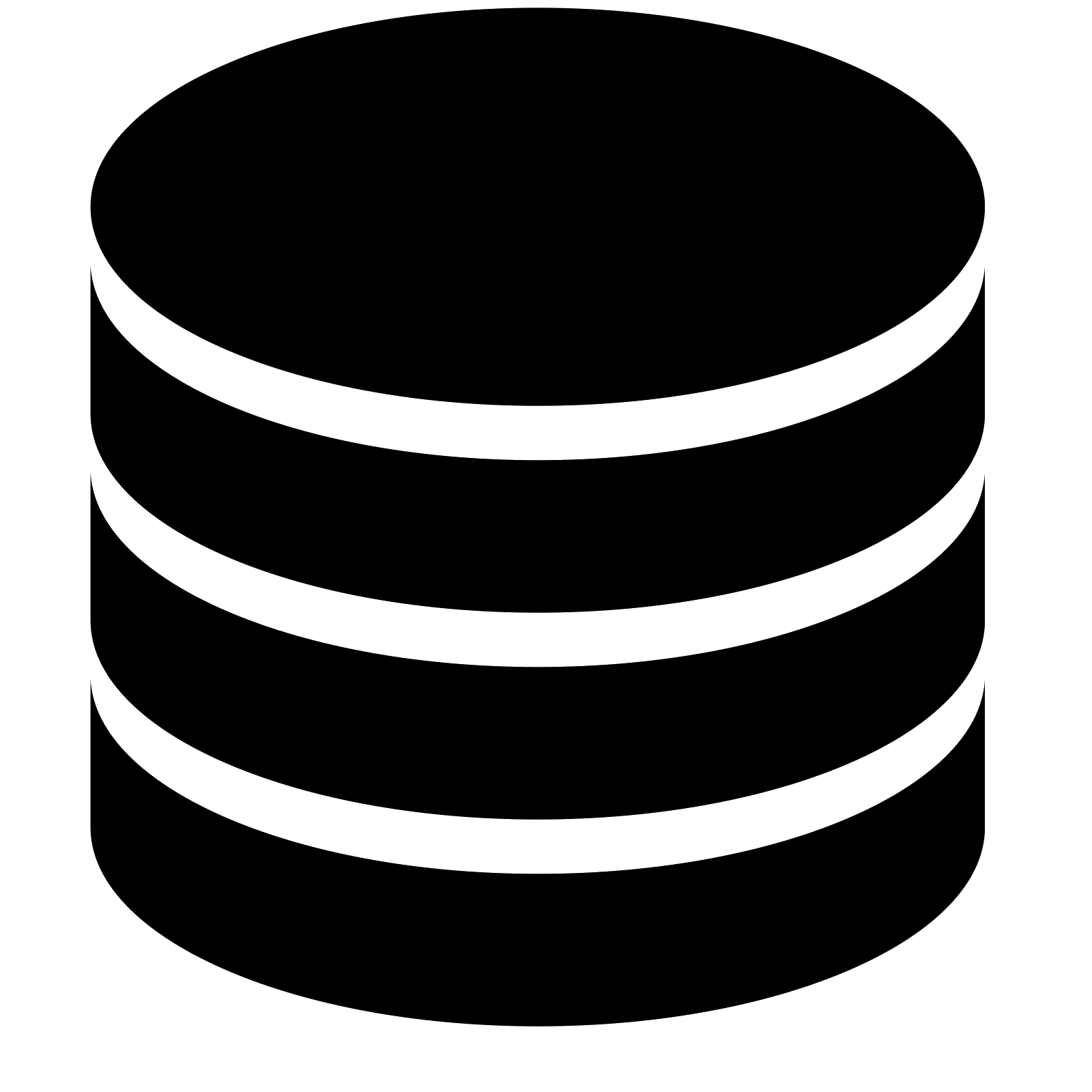
**PROIECT**

la disciplina

**BAZE DE DATE**

**Profesor coordonator:**  **Echipa de proiect:**

Cosmina Ivan Ioan-Florin Iftene

Ion-Robert Brînzoi

Mark-Otto Vicsi

**An academic: 2021-2022**

Cuprins

[1. Prezentarea proiectului 4](#_Toc92892236)

[2. Modelul de date 5](#_Toc92892237)

[2.1. Tabele și atribute 5](#_Toc92892238)

[2.2. Diagrama EER 7](#_Toc92892239)

[2.3. Proceduri și funcții 8](#_Toc92892240)

[2.4. Triggere 10](#_Toc92892241)

[2.5. Evenimente 13](#_Toc92892242)

[2.6. Bonusuri 14](#_Toc92892243)

[3. Detalii de implementare 16](#_Toc92892244)

[4. Instrucțiuni de utilizare 18](#_Toc92892245)

[5. Concluzii 19](#_Toc92892246)

# Prezentarea proiectului

Proiectul nostru presupune dezvoltarea unei aplicații care lucrează cu baze de date, pentru gestiunea unei platforme de studiu. Prin realizarea unei aplicații cu interfață grafică, vom facilita interacțiunea utilizatorilor cu baza de date.

Aplicația va putea fi folosită de către 4 tipuri de utilizatori: studenți, profesori, administratori și super-administratori, pe baza unui proces de autentificare în care se vor folosi un nume de utilizator și o parolă. În funcție de tipul contului său, fiecare utilizator are acces la anumite date din baza de date și la diferite acțiuni care pot fi executate cu ajutorul interfeței grafice create.

Conturile de utilizatori pot fi create de către super-administrator sau de către administratori. Tot aceste 2 tipuri de conturi sunt și cele de pe care se pot adăuga cursuri, profesori, studenti, grupuri de studiu pentru studenți și se pot atribui profesori pentru fiecare curs existent.

De pe contul de tip profesor, se pot programa activități (curs, seminar, laborator, examen) pentru cursurile la care predă profesorul respectiv, precizându-se numărul minim de participanți, ziua și ora la care se va desfășura acea activitate, de când până când se repetă activitatea și ponderea pe care o are în nota finală. Din contul lui, profesorul va putea să vizualizeze notele studenților de la fiecare activitate, să adauge note noi, să vizualizeze programul săptămânal (pe zile și ore) și să îl descarce, și să își citească eventualele notificări (de exemplu, profesorul va primi o notificare când o activitate de grup la care participa nu se va mai desfășura).

Contul de tip student oferă utilizatorilor diverse opțiuni, precum înscrierea la un curs și la activitățile cursului respectiv, vizualizarea tuturor notelor, a notificărilor și vizualizarea și descărcarea orarului. În plus, studenții pot să se alăture grupurilor de studiu deja existente, unde pot invita alți studenți să se alăture grupului, pot programa activități, pentru care pot seta un număr minim de participanți și au opțiunea de a adăuga un profesor pentru activitatea respectivă și au posibilitatea de a trimite mesaje, care vor putea fi citite de către toți membrii grupului. Totodată, pe conturile lor, studenții vor primi sugestii de înscriere la activități ale cursurilor la care sunt deja înscriși.

Pentru crearea bazei de date, popularea inițială, dezvoltarea de proceduri, triggere și pentru generarea diagramei tabelelor am folosit MySQL Workbench 8.0.

Interfața grafică am realizat-o cu ajutorul Microsoft Visual Studio 2019, utilizând limbajul de programare C#. Limbajul C# este o componentă a pachetului Microsoft Visual Studio.NET care conține o bibliotecă modernă de clase numită Microsoft.NET Framework ce pune la dispoziția programatorului aproape toate instrumentele necesare pentru a realiza aplicații pentru internet, web si windows. Această colecție de clase conține biblioteca de clase WindowsForms care furnizează instrumente pentru realizare de aplicații tradiționale pentru sistemul de operare Windows.

# Modelul de date

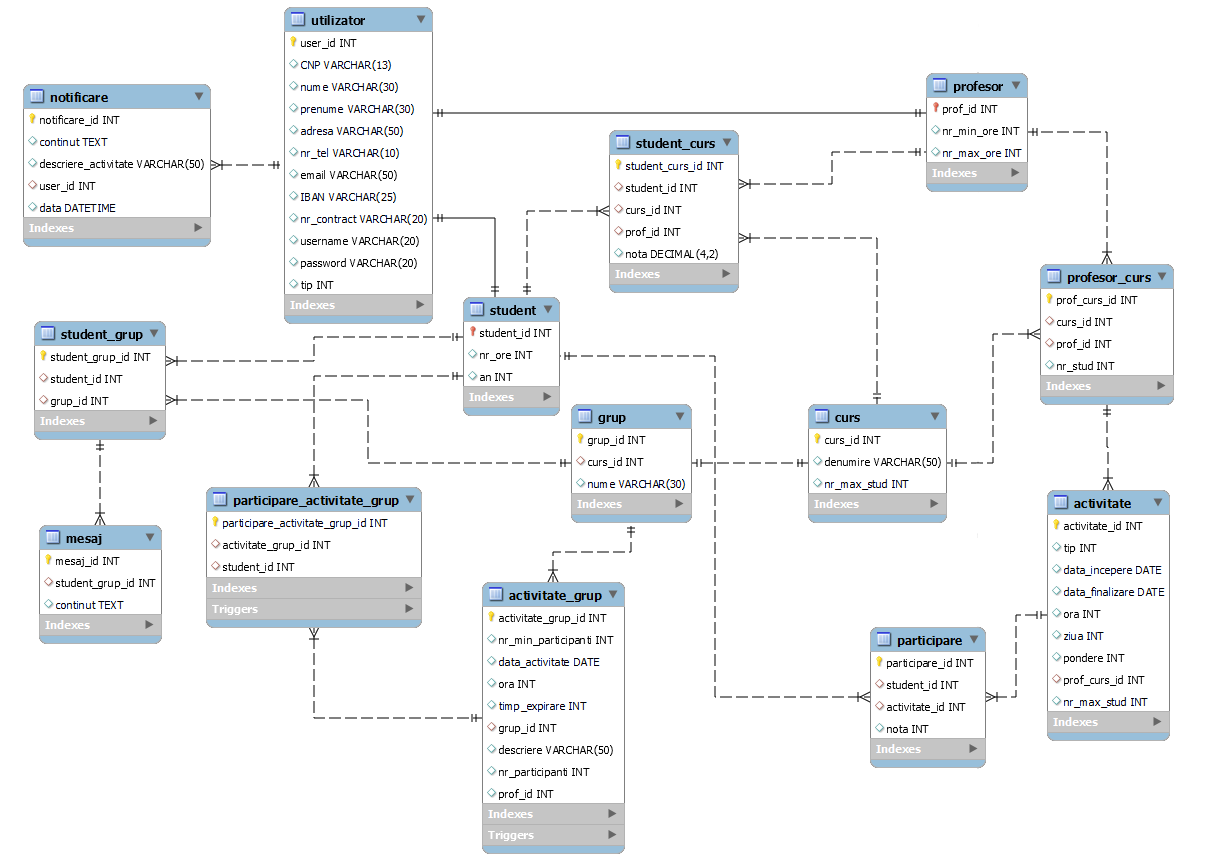
## Tabele și atribute

* **Utilizator** – cuprinde informații despre fiecare utilizator din baza de date, având un atribut suplimentar tip care denotă ce fel de utilizator este: 0 pentru student, 1 pentru profesor, 2 pentru administrator, 3 pentru super-administrator. Celelalte atribute sunt: user\_id, CNP, nume, prenume, adresă, număr de telefon, email, IBAN, număr de contract, nume de utilizator și parolă. Acesta are legături către tabele Student si Profesor.
* **Student** – cuprinde informații adiționale despre utilizatorii de tip student, și anume numărul de ore (nr\_ore) și anul de studiu (an). Este legat de tabelele Student\_curs si Student\_grup.
* **Profesor** – cuprinde informații suplimentare despre utilizatorii de tip profesor, adică numărul minim si maxim de ore (nr\_min, nr\_max). Are legătură cu tabelul Profesor\_curs.
* **Student\_curs** – rezolvă relația many-to-many dintre tabele Student si Curs. Atribute: student\_curs\_id, student\_id, curs\_id, prof\_id, nota (reprezintă nota finala la cursul respectiv).
* **Curs** – cuprinde totalitatea cursurilor, având un curs\_id, o denumire (denumire) si un număr maxim de studenți (nr\_max\_stud).
* **Profesor\_curs** – rezolvă relația many-to-many dintre tabele Profesor și Curs. Are următoarele atribute: prof\_curs\_id, curs\_id, prof\_id, nr\_studenti (reprezintă numărul curent de studenți pe care îl are un profesor la un anumit curs).
* **Activitate** – reține informații despre o activitate care poate fi programată de către un profesor care preda un curs. Atribute: activitate\_id, tip (0 = curs, 1 = seminar, 2 = laborator, 3 = colocviu, 4= examen), data\_incepere, data\_finalizare, ora, ziua, pondere, prof\_curs\_id, nr\_max\_stud (maximul numărului de studenți care pot fi înscriși la o activitate).
* **Participare** – are următoarele atribute: participare\_id, student\_id, activitate\_id, nota (nota unui student la o activitate).
* **Student\_grup** – rezolva relația many-to-many dintre tabele Student\_grup si Grup. Atribute: student\_grup\_id, student\_id, grup\_id.
* **Grup** – cuprinde totalitatea grupurilor de studiu. Atribute: grup\_id, curs\_id (cursul corespunzător pentru care a fost creat grupul), nume.
* **Activitate\_grup** – oferă informații despre o activitate care a fost programată pentru un grup. Atribute: activitate\_grup, descriere, nr\_participanti,

nr\_min\_participanti, data\_activitate, ora, timp\_expirare, grup\_id, descriere, nr\_participanti, prof\_id.

* **Participare\_activitate\_grup**: are atributele participare\_activitate\_grup\_id, activitate\_grup\_id, student\_id.
* **Mesaj** – stochează mesajele care au fost trimise de un student într-un grup din care face parte. Atribute: mesaj\_id, student\_grup\_id, continut.
* **Notificare** – cuprinde mesaje care alertează studentul sau profesorul de anularea unei activități de grup care nu a atins numărul minim de participanți. De asemenea, poate cuprinde mesaje de sugestie pentru înscrierea unui student la un grup de studiu corespunzător pentru unul dintre cursurile la care participă. Atribute: notificare\_id, continut, descriere\_activitate, user\_id, date.

## Diagrama EER



## Proceduri și funcții

In realizarea proiectului am utilizat următoarele proceduri și funcții:

* **creeare\_utilizator()** – are rolul de a realiza inserarea unui utilizator nou de către administratori sau super-administrator. Variabila ”tip” indică dacă utilizatorul este student sau profesor. Variabilele ”var1” și ”var2” diferă în cele două cazuri: în cazul studentului se referă la numărul de ore, respectiv anul de studiu, iar în cazul profesorului, la numărul minim și maxim de ore.
* **stergere\_utilizator()** – șterge un utilizator identificat după user\_id.
* **student\_inscriere\_curs()** – selectează din profesorii care predau cursul dat ca și parametru pe cel care are numărul minim de studenți și îl asignează pe student la cursul predat de acesta.
* **prof\_asign\_act()** – oferă profesorului funcția de a programa o activitate pentru unul dintre cursurile predate de acesta. Procedura caută ID-ul unui curs după denumire și inserează o nouă activitate conform parametrilor care îi sunt furnizați.
* **auto\_inserare()** – în momentul în care un student programează o activitate în cadrul unui grup, această procedură va fi apelată prin intermediul aplicației. Se selectează activitatea respectivă și apoi este înscris studentul printr-un insert.
* **inserare\_mesaj()** - are rolul de a permite participantului unui grup, identificat prin ID să scrie un mesaj în grupul respectiv, care va fi salvat în tabelul mesaj.
* **prof\_nota\_act()** – permite utilizatorului de tip profesor să introducă sau să actualizeze o notă deja introdusă pentru o activitate (curs, seminar, laborator, colocviu, examen). Funcția modifică apoi nota finală conform noilor date.

Text

Description automatically generated

* **pondere\_locala()** – are rolul de a calcula ponderea cumulată a activităților programate de un profesor la un anumit curs. Ponderea este returnată și prin intermediul aplicației se adună cu ponderea unei noi activități pe care vrea profesorul să o programeze. Dacă această sumă depășește pragul de o sută, utilizatorul va primi o eroare prin care va fi informat de valoarea depășită și nu i se va permite introducerea activității.

Graphical user interface, text, application, email

Description automatically generated

## Triggere

* Graphical user interface, text, application

  Description automatically generated**increment\_nr\_part** –după momentul inserării unei tuple în tabelul participare\_activitate\_grup trigger-ul se declanșează și în funcție de activitate\_grup\_id merge în tabelul activitate\_grup și incrementează numărul de participanți pentru activitatea respectivă.
* **stergere\_grup** – după efectuarea unei ștergeri în tabelul student\_grup, se verifică dacă acesta mai are tuple. În caz contrar va avea loc ștergerea grupului respectiv.

**Graphical user interface, text

Description automatically generated**

* **increment\_stud\_prof\_curs** – după înscrierea unui student la curs, trigger-ul se va declanșa și va incrementa numărul de studenți pe care îi are un profesor la un curs, care se regăsește în tabelul Profesor\_curs

Text

Description automatically generated

* Graphical user interface, text, application

  Description automatically generated**stergere\_act** – înainte de ștergerea din tabelul activitate\_grup, verifică ce fel de ștergere a avut loc. Dacă se șterge o activitate programată în viitor din cauză faptului că nu a fost atins numărul minim de participanți, se va insera câte un mesaj în legătură cu anularea activității pentru fiecare student care s-a înscris și pentru profesorul asignat activității, dacă este cazul.

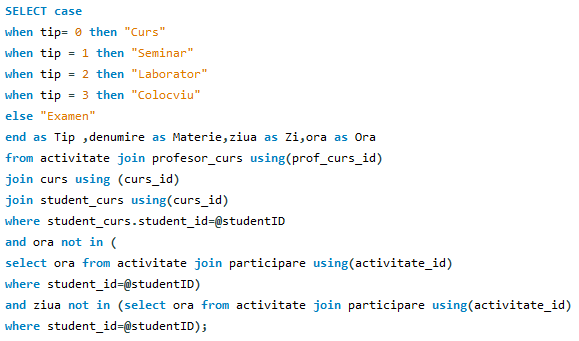
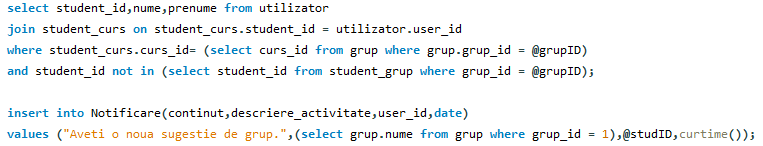
## Evenimente

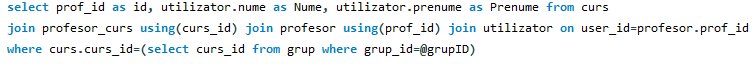
* **decrease** – acest eveniment are rolul de a decrementa o dată la un minut timpul de expirare pentru fiecare activitate creată în cadrul unui grup de studiu. Dacă timpul de expirare ajunge la zero și numărul maxim de participanți nu este atins, activitatea va fi ștearsă. Dacă se atinge acest număr, timpul de expirare va rămâne zero, iar activitatea va fi ștearsă numai în ziua următoare desfășurării acesteia.

Graphical user interface, text, application, email

Description automatically generated

## Bonusuri

1. Sugestiile pe care le primesc studenții de a se înscrie la activități ale cursurilor la care sunt înrolați:
2. Sugestiile de participanți pentru un grup de studiu, care are ca și subiect un curs la care sunt înrolați și trimiterea invitațiilor:
3. Pentru adăugarea unui profesor la o activitate de grup, studentul care creează activitatea are posibilitatea de a bifa o căsuță dacă dorește adăugarea unui profesor, moment în care îi va apărea o listă cu toți profesorii care predau materia pentru care e creeat grupul și va avea posibilitatea de a selecta un profesor din listă.



1. Sugerarea ferestrelor pentru activitățile grupurilor de studiu:

Pentru fiecare student care face parte dintr-un grup se selecteaza ora și ziua din saptamana curentă ale tuturor activităților sale. Vom avea un tabel auxiliar (freq\_search) care va avea atributele: frecvență, oră și zi. Se parcurge apoi tabelul cu activitățile studenților cu ajutorul unui cursor și de fiecare dată când se găsește o activitate, se incrementeaza cu 1 frecvența in tabelul auxiliar pentru ora si ziua corespunzatoare acelei activitati. La final, se ordoneaza tabelul crescator dupa frecvență, și se sugerează prima tuplă ca și fereastra potrivita pentru desfasurarea unei noi activitati.

# Detalii de implementare

Pentru realizarea aplicației am avut nevoie de 25 de clase, fiecare fiind corespunzătoare unei ferestre din interfața grafică:

* **LoginForm**: aceasta este fereastra de logare, cea care apare atunci când este deschisă aplicația;
* **MainFormSuperAdmin**: fereastra care apare dacă contul cu care s-a realizat logarea este de tip super-administrator;
* **MainFormAdmin**;
* **MainFormStudent**;
* **MainFormProfesor**;
* **AdaugaUserForm**: fereastra care se deschide atunci când administratorul sau super-administratorul crează un cont nou;
* **AdaugaCursForm**: fereastra care se deschide atunci când administratorul sau super-administratorul crează un curs nou;
* **AdaugaGrupForm**: fereastra care se deschide atunci când administratorul sau super-administratorul crează un grup nou;
* **ActivitateNouaForm**: fereastra care se deschide când un student programează o activitate pentru un grup din care face parte;
* **ActivitateGRUPForm**: fereastra din care studentul poate vizualuza activitățile unui grup din care face parte;
* **AdminShowForm**: fereastra din care administratorii/super-administratorii pot vizualiza cursurile, studentii, profesorii sau grupurile de studiu, în funcție de butonul pe care îl apasă din MainFormAdmin;
* **AsignProfCursForm**: fereastra din care administratorul atribuie un profesor pe care îl selectează din fereastra anterioară, unui anumit curs;
* **CatalogForm**: fereastra din care profesorul selectează materia și activitatea pentru care dorește să vadă notele;
* **VizCatalogForm**: fereastra în care profesorul poate vedea notele studenților. Aceasta este folosită și pentru afișarea notelor unui student (din contul de student);
* **DetaliiContForm**: fereastră accesibilă din toate conturile de utilizatori, în care apar anumite date personale ale utilizatorului (nume, prenume, adresă, numar de telefon, email, numarul contractului)
* **DetaliiGrupForm**: fereastra specifică unui grup, din care se pot accesa date precum membrii, programul și mesajele.
* **GrupNouForm**: fereastră accesibilă din conturile de tip student, din care aceștia se pot alătura unui nou grup;
* **GrupuriShowForm**: fereastră în care studenții pot vizualiza grupurile din care fac, pot accesa detaliile unui anumit grup, pot părăsi un grup și pot accesa ferestra pentru a se alătura unui nou grup;
* **InscriereStudentCurs**: fereastra din care un student se înscrie la un curs și la activități specifice acestuia;
* **MembriiShowForm**: fereastra în care se pot vedea membrii grupului din care a fost accesată;
* **MesajeForm**: fereastra în care membrii unui anumit grup pot vizualiza mesajele grupului respectiv și pot accesa fereastra pentru a lăsa un nou mesaj;
* **MesajNouForm**: fereastra din care studenții dintr-un grup pot trimite mesaje în grupul respectiv;
* **ProgrameazăForm**: fereastra din care un profesor programează o avtivitate pentru o materie la care predă
* **ProgrameProfForm**: fereastra în care profesorii și studenții își pot vizualiza și descărca programul săpămânal pentru facultate
* **SugestiiForm**: fereastră în care studenții pot vedea sugestii de înscriere la activități (dacă este accesată din MainFormStudent) sau sugestii de invitații pentru grup (dacă este accesată din MembriiShowForm).

# Instrucțiuni de utilizare

Mai întâi, trebuie să vă asigurați ce dețineți un server MySQL pe care este implementată baza de date a proiectului, cu toate triggerele, procedurile, și evenimentele ei. Se deschide apoi ”Pro1.sln”, se selectează ”LoginForm.cs”, se apasă tasta F7 pentru a deschide fișierul de cod al clasei respective și apoi, pe linia 44 din acel fișier se modifică datele corespunzătoare uid și pwd cu numele de utilizator și parola serverului deținut, pentru a realiza conexiunea la baza de date.

Pentru a deschide aplicația, cu ”Pro1.sln” deschis, se apasă combinația de taste ”ctrl+F5”, sau se deschide direct executabilul accesând ”\Pro1\bin\Debug\netcoreapp3.1” și selectând apoi ”Pro1.exe”. Aplicația se va deschide cu pagina de logare, unde trebuie introduse numele de utilizator si parola pentru contul pe care doriți să îl accesați. Pentru început, vă puteți loga cu contul de super-administrator, folosind numele de utilizator ”superadmin” și parola ”superadmin”. Din acest cont, veți avea acces la datele tuturor celorlalte conturi deja înregistrate în baza de date și le veți putea folosi pentru a încerca conturile de administrator, profesor sau student.

Pe fiecare fereastră din aplicație veți vedea butoane denumite sugestiv, care vă pot ajuta să alegeți ceea ce doriți să faceți.

Dacă se realizează orice fel de modificare asupra bazei de date prin intermediul unor ferestre care afișau conținut din bază, este necesară apăsarea butonului ”Inapoi” si apoi redeschiderea acelei ferestre, pentru ca aplicația să reîmprospăteze datele și a putea, astfel, vizualiza noile modificări.

Pentru a închide aplicația este recomandată apăsarea butonului ”Inapoi” până la revenirea în fereastra principală a contului cu care s-a realizat logarea, de unde se apasă butonul ”Ieșire”.

# Concluzii

Proiectul nostru oferă o metodă eficientă pentru gestionarea unei platforme de studiu. Prin funcționalitățile oferite de către aplicație, fiecare tip de utilizator poate realiza cu ușurință acțiuni specifice pentru contul cu care s-a logat.

În viitor, aplicația ar putea fi îmbunătățită prin adăugarea opțiunii de creeare de teste de către profesori, teste pe care studenții să le poată susține prin intermediul aplicației și care să se poată corecta automat (dacă sunt de tip grilă), iar nota să fie introdusă imediat după finalizare în catalog. Totodată, ar mai putea fi introdusă opțiunea de a trimite mesaje către oricare alt utilizator al platformei (indiferent că este vorba de student, profesor sau administrator).

În acest moment, baza de date folosită este una locală, astfel că aplicația poate fi folosită doar de pe un dispozitiv care conține și este conectat la baza de date.