Baze de Date

-Proiect-

Situația școlară in cadrul facultăților

Student: Brumă Sebastian

Grupa: 3131 B

An: 3

Cuprins

[Descrierea temei 3](#_Toc105422014)

[Mediu de dezvoltare 3](#_Toc105422015)

[Entitățile bazei de date 4](#_Toc105422016)

[Structura entităților/tabelelor 4](#_Toc105422017)

[Schema bazei de date 5](#_Toc105422018)

[Întrebările la care va răspunde baza de date proiectată 6](#_Toc105422019)

[Lista cu întrebări ale căror răspunsuri implică interogarea tabelelor 6](#_Toc105422020)

[Lista cu întrebări ale căror răspunsuri implică adăugarea, actualizarea și ștergerea datelor 6](#_Toc105422021)

[Scripturile utilizate pentru creearea tabelelor 7](#_Toc105422022)

[Schema interfetei pentru utilizator 9](#_Toc105422023)

[Manual de utilizare 9](#_Toc105422024)

[Concluzie 13](#_Toc105422025)

[Bibliografie 13](#_Toc105422026)

# Descrierea temei

Tema proiectului reprezintă implementarea unei aplicații ce gestioneaza situația școlară a studenților. Pentru implementare am decis ca am nevoie de următoarele entități:

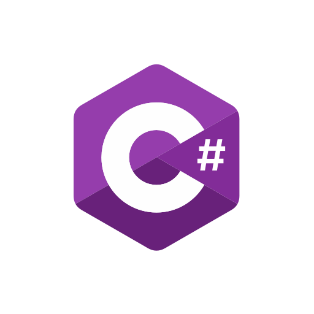
* Facultăți
* Program de studiu
* Studenți
* Materii
* Notele studentului

Pentru a putea înțelege mai bine aceste entități, ne putem imagina că un student din cadrul facultății X este in anul 2 si are materiile A, B și C. Un administrator/profesor va trebui să adauge notele studentului la cele 3 materii.

Pentru a stoca aceste informații cu ușurință într-un mediu digital, vom avea nevoie de o bază de date respectiv o aplicație ce procesează informațiile, la rândul său aplicația va avea nevoie de o interfață ușoară de utilizat de către un utilizator.

# Mediu de dezvoltare

Pentru dezvoltarea aplicațiilor de baze de date, tehnologia .NET oferă interfața ADO.NET (Access Data Object) ca parte integrantă a bibliotecii .NET Framework Class Library (FCL) care permite accesul la baze de date atât în aplicații locale cât şi în aplicații distribuite în Internet. Pentru accesul la bazele de date, ADO.NET folosește un modul software de interconectare numit driver/data provider.



# Entitățile bazei de date

1. Facultati

- în acest tabel se vor afla denumirile facultatilor create

1. Program de studiu

* un tabel in care sunt inregistrare toate programele de studii pentru facultatile create

1. Materii

* un tabel in care se gasesc toate materiile inregistrate

1. Studenti

* tabelul va contine informatiile despre studenti

1. Note

* Acest tabel va face legatura intre tabelul cu studenti si cel cu materii, pentru a putea stoca notele

# Structura entităților/tabelelor

1. Facultati

* IDFacultate (PK)
* Denumire

1. Program de studiu

* IdProgramStudiu (PK)
* Ciclu
* Specializare
* Durata
* IdFacultate (FK)

1. Studenti

* IDStudent (PK)
* Nume
* Prenume
* Email
* Telefon
* An
* IdProgramStudiu (FK)

1. Materii

* IDMaterie (PK)
* Denumire
* An
* Semestru
* ProcentLaborator
* ProcentCurs

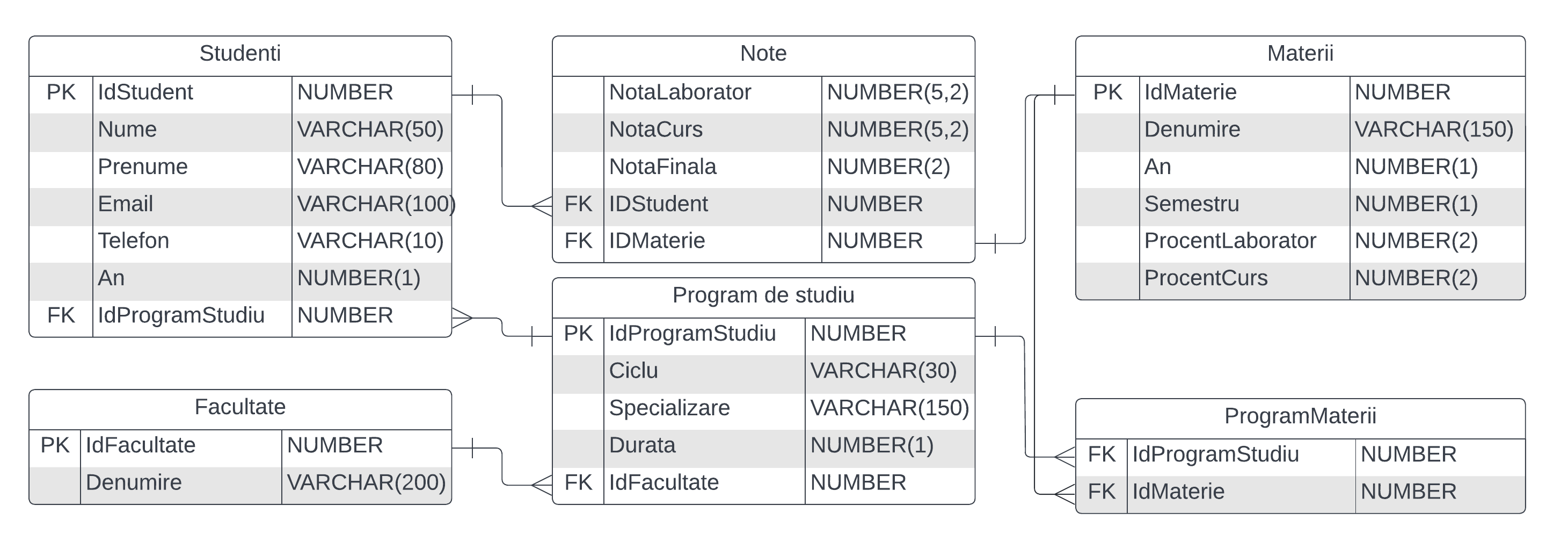
1. Note

* NotaLaborator
* NotaCurs
* NotaFinala
* IDStudent (FK)
* IDMaterie (FK)

1. Program-Materii – un tabel de legatura intre materii si programul de studiu

* IdProgramStudiu (FK)
* IdMaterie (FK)

# Schema bazei de date



# Întrebările la care va răspunde baza de date proiectată

## Lista cu întrebări ale căror răspunsuri implică interogarea tabelelor

1. Care sunt studenții de la specializarea X?
2. Câți studenți sunt in cadrul facultății X?
3. Ce materii a promovat studentul X?
4. Care sunt facultățile cu durata de X ani?
5. Care sunt materiile si notele studentului X?

## Lista cu întrebări ale căror răspunsuri implică adăugarea, actualizarea și ștergerea datelor

1. Cum se poate rezolva daca un profesor a greșit nota unui student?

R: Se modifică nota studentului respectiv.

1. Un student nou înscris la facultatea X trebuie înregistrat. Cum putem face asta?

R: Adăugăm în baza de date noul student.

1. Daca un student a obținut o nouă notă la materia X. Cum putem înregistra acest lucru?

R: Adăugăm in baza de date nota obținută a studentului.

1. Ce facem daca o facultate a fost desființată?

R: Ștergem datele despre facultatea respectivă.

1. Ce se întâmplă dacă un student își schimbă numele de familie?

R: Modificăm câmpul “Nume” in baza de date.

1. Cum înregistrăm o nouă facultate apărută?

R: Adăugăm facultatea in baza de date.

1. Cum se rezolvă o eroare in care o materie a fost înregistrată la alta facultate?

R: Ștergem înregistrarea privind materia respectiva.

1. Cum actualizăm dacă un program de studiu și-a mărit durata de studiu la X ani?

R: Modificăm durata la X ani in baza de date.

1. Cum procedăm dacă au apărut 2 materii la facultatea X?

R: Adăugăm noile materii.

1. Ce putem face dacă s-au schimbat procentele la o materie?

R: Modificam procentajele.

# Scripturile utilizate pentru creearea tabelelor

Tabel Facultate:

CREATE SEQUENCE seq\_facultati\_brumas;

CREATE TABLE facultati\_brumas(

IdFacultate NUMBER PRIMARY KEY,

Denumire VARCHAR(200) NOT NULL UNIQUE

);

Tabel Program de studiu:

CREATE SEQUENCE seq\_programestudii\_brumas;

CREATE TABLE programestudii\_brumas(

IdProgramStudiu NUMBER PRIMARY KEY,

Ciclu VARCHAR(30) NOT NULL,

Specializare VARCHAR(150) NOT NULL,

Durata NUMBER(1) NOT NULL

CONSTRAINT ck\_programstudii\_durata\_brumas CHECK(Durata BETWEEN 1 AND 8),

IdFacultate NUMBER

CONSTRAINT nn\_programstudiu\_fac\_brumas not null

CONSTRAINT fk\_programstudiu\_fac\_brumas references facultati\_brumas(IdFacultate)

);

Tabel Materii:

CREATE SEQUENCE seq\_materii\_brumas;

CREATE TABLE materii\_brumas(

IdMaterie NUMBER PRIMARY KEY,

Denumire VARCHAR2(150) NOT NULL,

An NUMBER(1) NOT NULL

CONSTRAINT ck\_materii\_an\_brumas CHECK(An BETWEEN 1 AND 6),

Semestru NUMBER(1) NOT NULL

CONSTRAINT ck\_materii\_semestru\_brumas CHECK(Semestru BETWEEN 1 AND 2),

ProcentLaborator NUMBER(2) NOT NULL

CONSTRAINT ck\_materii\_procentlab\_brumas CHECK(ProcentLaborator BETWEEN 10 AND 90),

ProcentCurs NUMBER(2) NOT NULL

CONSTRAINT ck\_materii\_procentcurs\_brumas CHECK(ProcentCurs BETWEEN 10 AND 90),

CONSTRAINT ck\_materii\_procente\_brumas CHECK(ProcentLaborator + ProcentCurs = 100) );

Tabel Studenti:

CREATE SEQUENCE seq\_studenti\_brumas;

CREATE TABLE studenti\_brumas(

IdStudent NUMBER PRIMARY KEY,

Nume VARCHAR2(50) NOT NULL,

Prenume VARCHAR2(80) NOT NULL,

Email VARCHAR2(100) NOT NULL UNIQUE,

Telefon VARCHAR(10) NOT NULL UNIQUE,

An NUMBER(1) NOT NULL,

IdProgramStudiu NUMBER

CONSTRAINT nn\_stud\_programstudiu\_brumas not null

CONSTRAINT fk\_stud\_programstudiu\_brumas references programestudii\_brumas(IdProgramStudiu)

);

Tabel Note:

CREATE TABLE note\_brumas (

NotaLaborator NUMBER(5,2)

CONSTRAINT ck\_note\_notaLaborator\_brumas CHECK(NotaLaborator BETWEEN 1.00 AND 10.00),

NotaCurs NUMBER(5,2)

CONSTRAINT ck\_note\_notaCurs\_brumas CHECK(NotaCurs BETWEEN 1.00 AND 10.00),

NotaFinala NUMBER(2) DEFAULT -1

CONSTRAINT ck\_note\_notaFinala\_brumas CHECK(NotaFinala BETWEEN 1 AND 10),

IdStudent NUMBER

CONSTRAINT nn\_note\_studenti\_brumas not null

CONSTRAINT fk\_note\_studenti\_brumas references studenti\_brumas (IdStudent),

IdMaterie NUMBER

CONSTRAINT nn\_note\_materii\_brumas not null

CONSTRAINT fk\_note\_materii\_brumas references materii\_brumas (IdMaterie)

);

Tabel Legatura Program-Materii

CREATE TABLE programmaterii\_brumas(

IdProgramStudiu NUMBER

CONSTRAINT nn\_program\_programstud\_brumas not null

CONSTRAINT fk\_program\_programstud\_brumas references programestudii\_brumas (IdProgramStudiu),

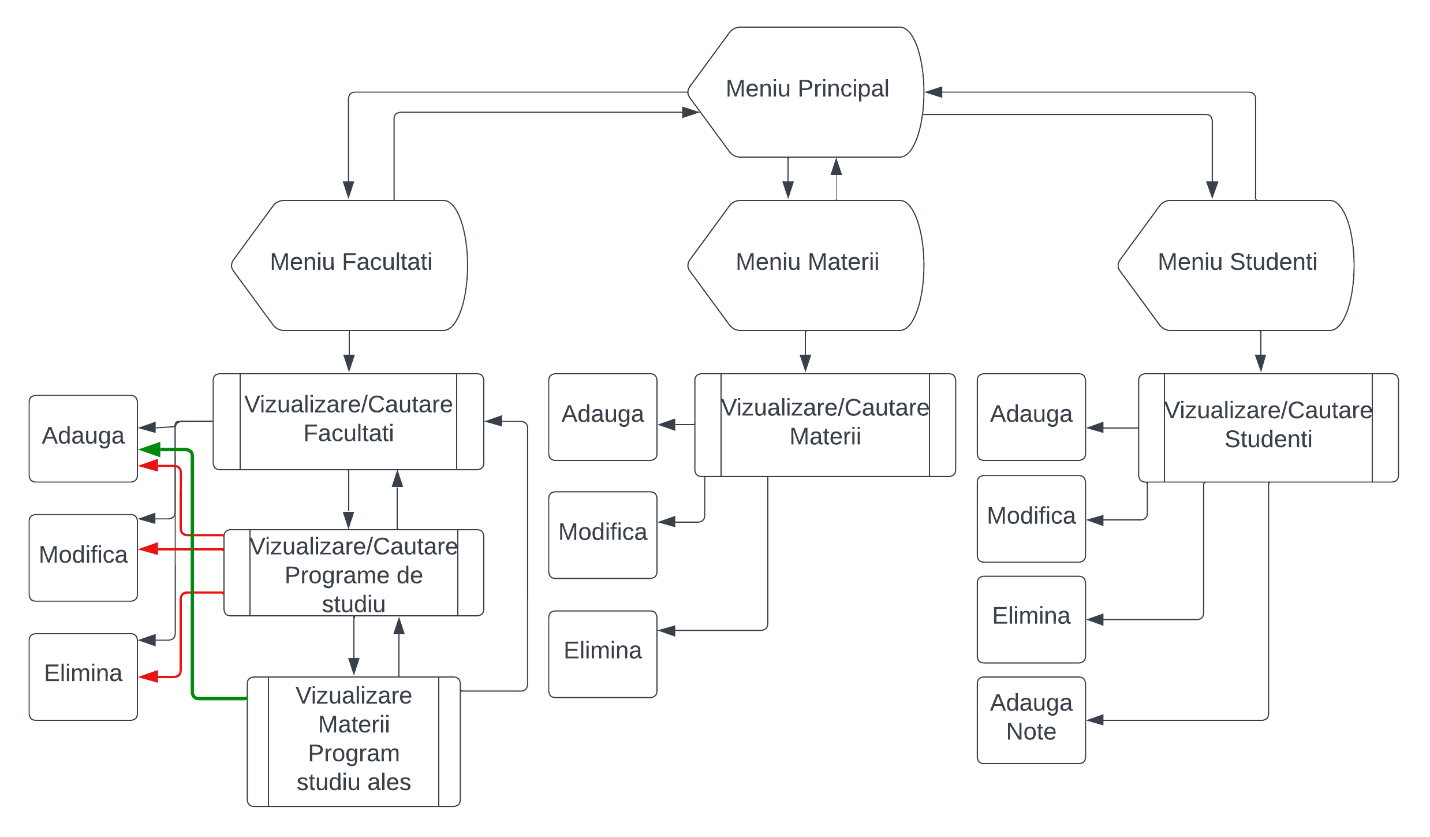
IdMaterie

CONSTRAINT nn\_programmat\_materii\_brumas not null

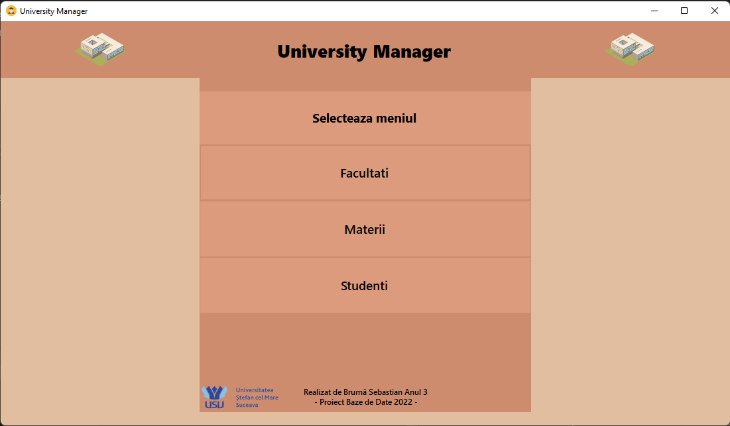
CONSTRAINT fk\_programmat\_materii\_brumas references materii\_brumas (IdMaterie)

);

# Schema interfetei pentru utilizator



# Manual de utilizare

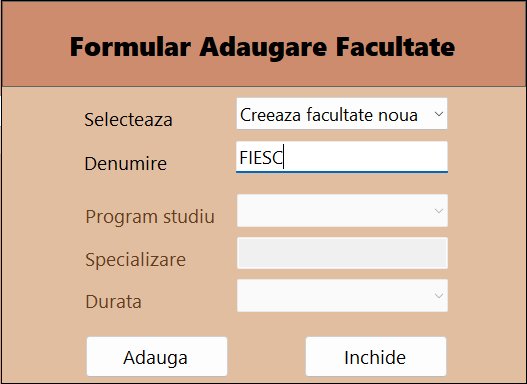
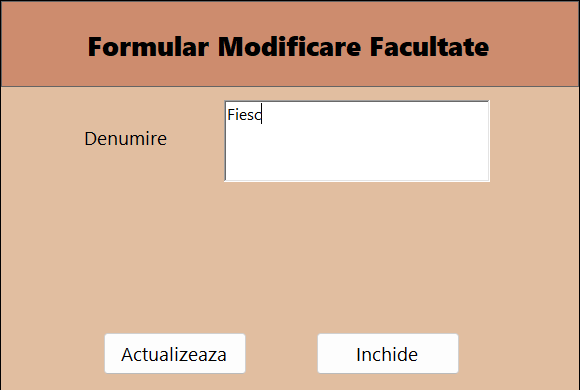


Prima fereastra a programului reprezinta meniul principal unde alegem un meniu din cele 3 afisate.

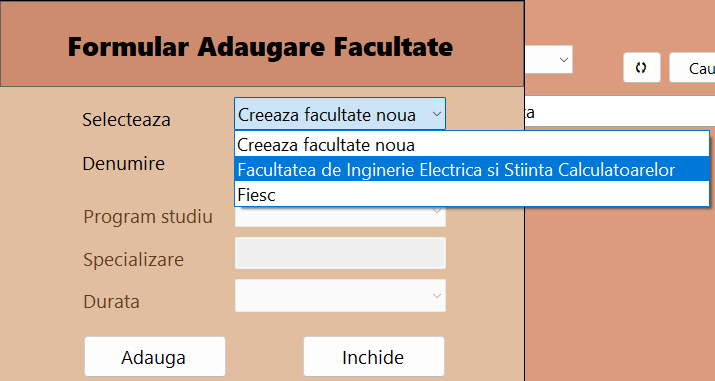


Dupa accesarea meniului „Facultati” se va deschide urmatoarea pagina, aici putem cauta, adauga, modifica sau elimina o facultate. Daca apasam „Vizualizeaza Programe” putem sa vedem toate programele de studiu pe care le are facultatea selectata.

Fereastra de adaugare Fereastra de modificare



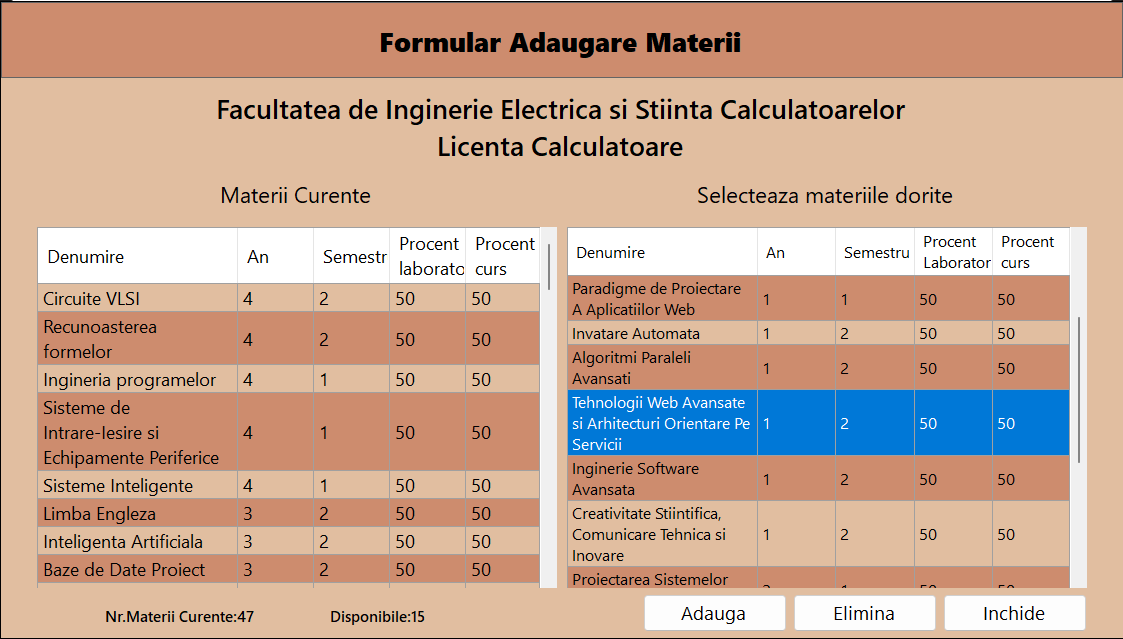
Asemanator tabelului cu facultati, putem cauta, adauga, modifica sau elimina un program de studiu.

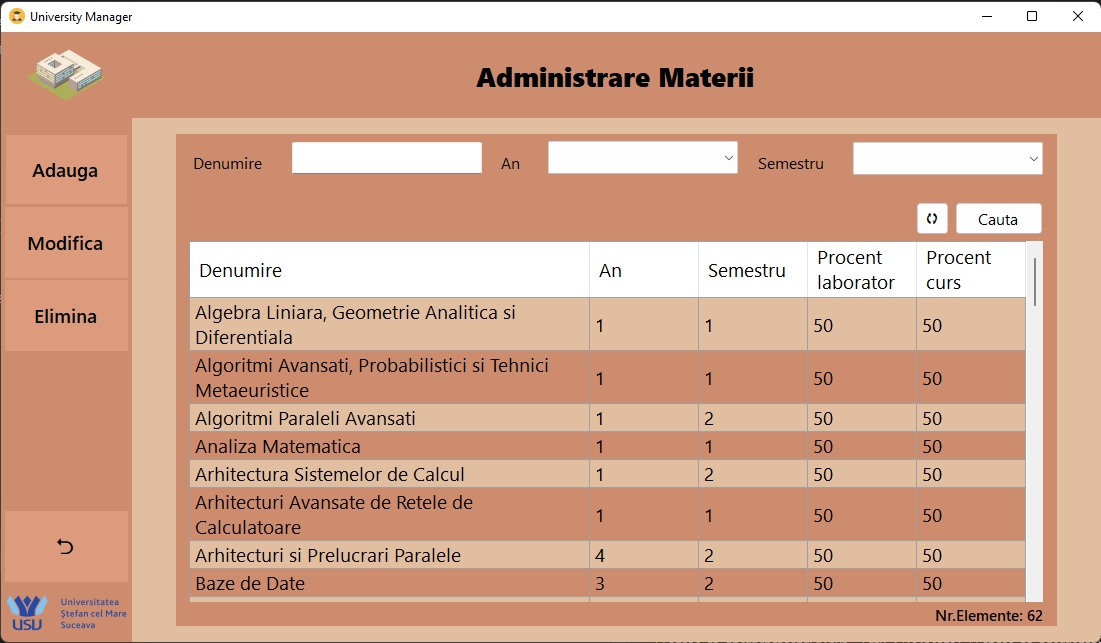
Pentru a adauga un program de studiu vom alege facultatea unde dorim sa inregistram noul program si se vor completa datele necesare.



Apasarea butonului „Vizualizeaza Materii” va deschide tabelul in care gasim materiile ce sunt la programul de studiu ales din tabel.

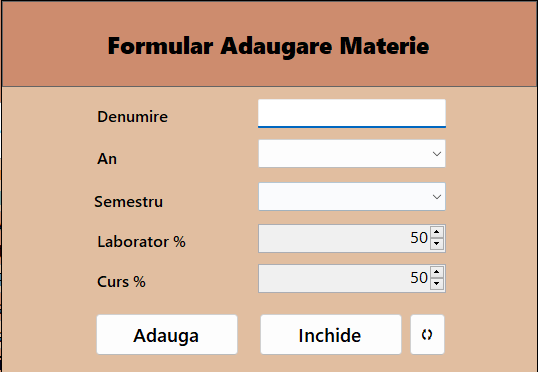
Aici putem doar sa adaugam/eliminam materii.

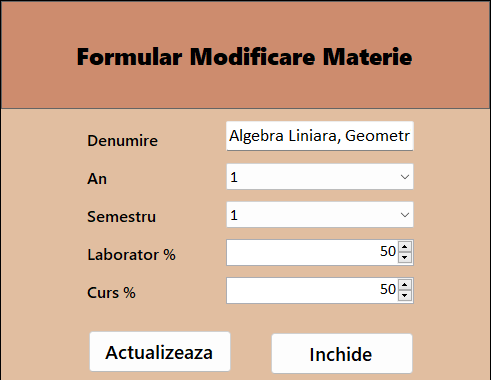


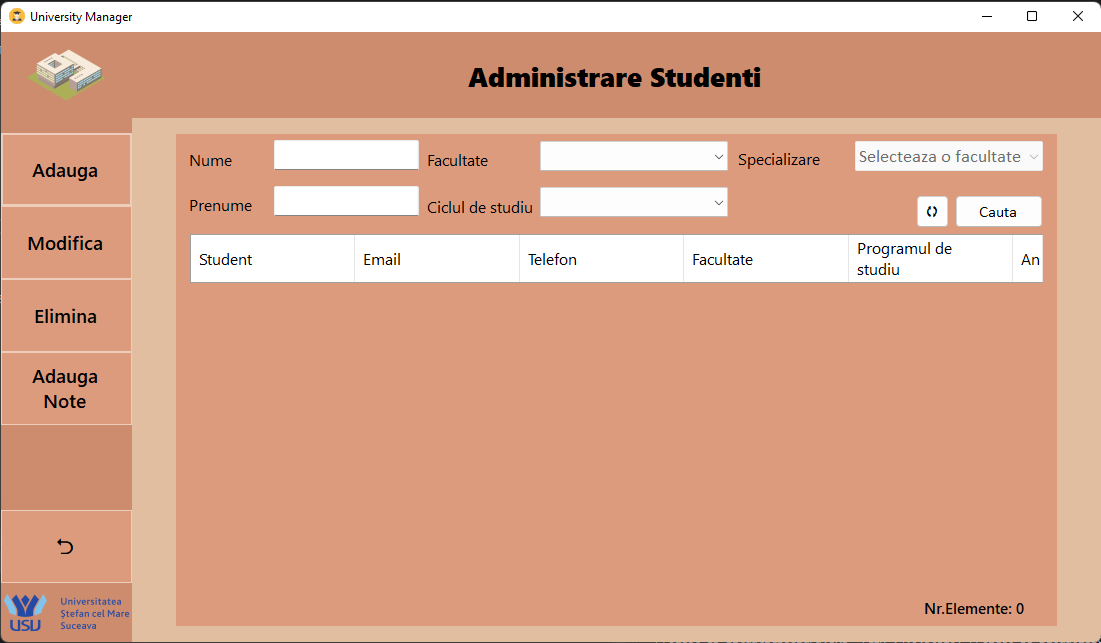
Din meniul principal vom alege acum „Materii”.

Aici va fi afisat tabelul cu toate materiile existente in baza de date.

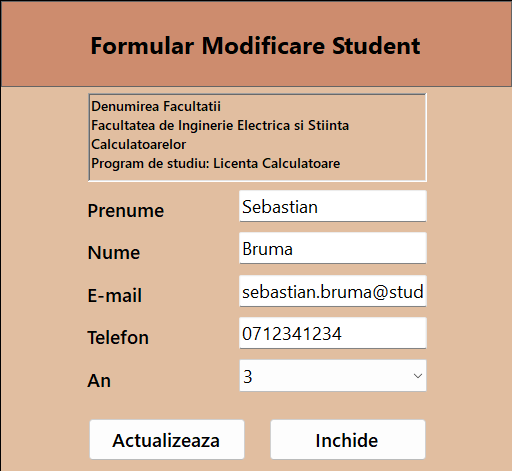
Putem cauta, adauga, modifica sau elimina.

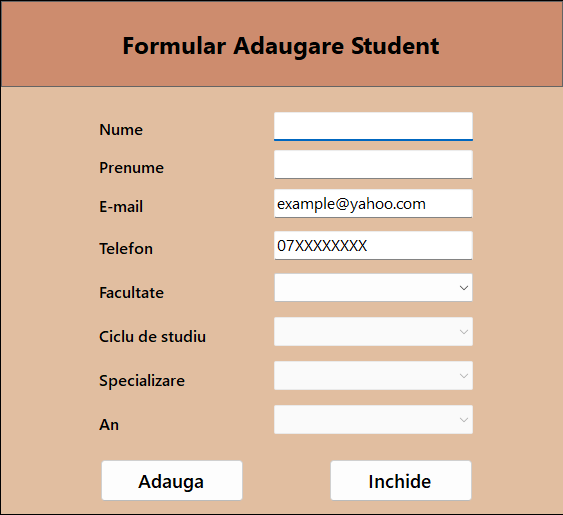
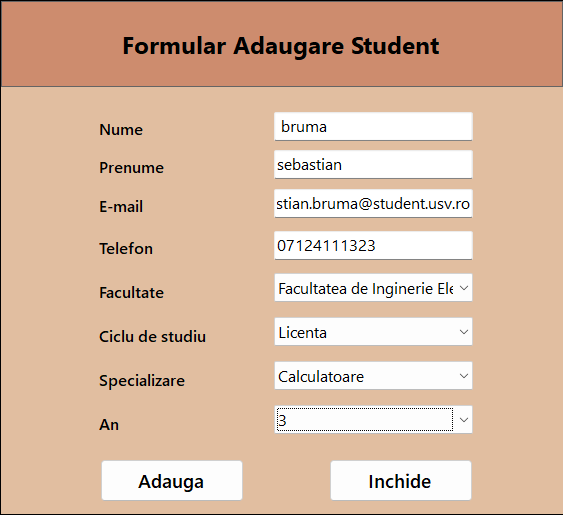
Fereastra de adaugare Fereastra de modificare

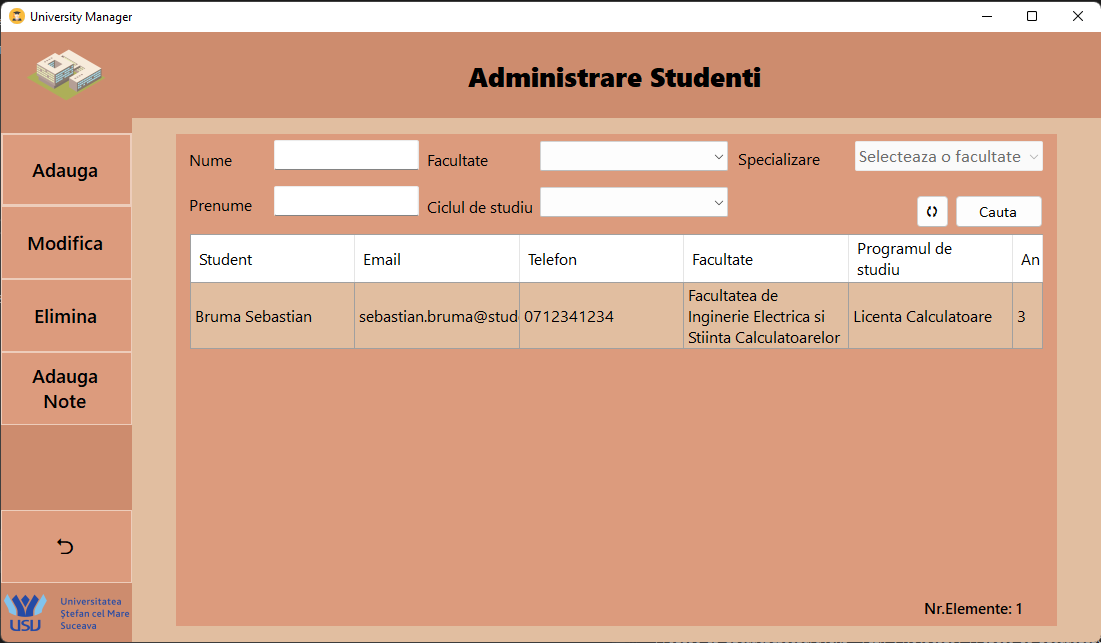


Din meniul principal vom alege „Studenti”.

In acest meniul putem cauta, adauga, modifica, elimina un student sau putem sa ii atribuim note.

Fereastra de adaugare Fereastra de modificare





Fereastra de adaugare note

# Concluzie

In urma implementarii proiectului am invatat sa folosec o baza de date de tip oracle, sa folosec libraria LINQ din C# si sa structurez un proiect.

Proiectul poate fi gasit si pe:

<https://github.com/HipersS1/Proiect_BD_SituatieScolara>

# Bibliografie

<https://stackoverflow.com/>

<https://www.lucidchart.com/pages/>