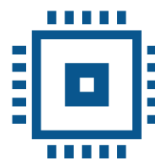




Universitatea *Transilvania* din Braşov
Facultatea de Inginerie Electrică şi Ştiinţa
Calculatoarelor
Departamentul Automatică şi Tehnologia
Informaţiei



PROIECT POO

Profesor îndrumător: Conferenţiar universitar ,
KRISTÁLY Dominic Mircea

Albert Alexandra, Gr. 1,
Chişu Alexandru, Gr. 1,
Ciangău Raluca, Gr. 1,
Grabovenco Bogdan, Gr. 2,
Dumitriu Andrei, Gr. 3,
AIA II

BRAŞOV, 2023

Cuprins

1. Titlul proiectului	2
1.1. Enunț	2
1.2. Rezolvare proiect	2
1.3. Rezultate	7

1. UniCatalog

1.1. Enunt

Să se realizeze o aplicație care să implementeze toate funcțiile necesare unui catalog pentru o facultate. Aplicația va oferi modulele:

1. Cicluri de învățământ
2. Programe de studii
3. Ani de studii
4. Planuri de învățământ / Discipline
5. Studenți
6. Grupe
7. Cadre didactice
8. Utilizatori
9. Cataloage / note

Aplicația poate conține orice alte module care sunt considerate a fi necesare.

1.2. Rezolvare Proiect

În secvență de cod 1 reprezintă butonul de autentificare ce conține logică autentificării. Practic se face comparația între conturile validate și datele introduse noi. În mod implicit, dacă avem aceleași valori, contul scris este corect și vom fi redirectionezi în meniul nostru (Menu.cs).

Secvență de cod 1

```
private void lblLogging_Click(object sender, EventArgs e)
{
    bool chk = false;
    try
    {
        SqlConnection con = new SqlConnection
            ("Data Source=DESKTOP-C6C06Q0;Initial
Catalog=CatalogVirtual;Integrated Security=True");
        con.Open();
        SqlCommand cmd = new SqlCommand("select Tip, [Nume
utilizator], [Parolă] " +
            "from Utilizatori where Tip=@a and [Nume utilizator]=@b
and " +
            "[Parolă]=@c", con);
        cmd.Parameters.AddWithValue("@a",
lbTip.GetItemText(lbTip.SelectedItem));
        cmd.Parameters.AddWithValue("@b", tbUsername.Text);
        cmd.Parameters.AddWithValue("@c", tbPassword.Text);
        //face ce face selectul din SSMS, adica returneaza un tabel
cu conditiile impuse

        SqlDataReader reader = cmd.ExecuteReader();//returneaza
fiecare coloana in parte
        if (reader.HasRows)//aici intra daca are minim un rand in
afisarea selectului
            chk = false;
        else
```

```
        chk = true;

        con.Close();
    }
    catch (Exception ex) { chk = true; MessageBox.Show(ex.Message);
}

    if (chk == false)
    {
        MessageBox.Show("Autentificare cu succes");
        this.Hide();
        Menu menu = new Menu();
        menu.ShowDialog();
    }
    else
    {
        MessageBox.Show("Parolă sau/și nume de utilizator
greșit/greșite");
        tbUsername.Text = ""; tbPassword.Text = "";
        lbTip.SelectedItem = "";
    }
}
```

Logica buton Facultate(Secvență de cod 2).

Secvență de cod 2

```
private void cicluri_Click(object sender, EventArgs e)
{
    //this.Controls.Add(list[2]); //de la 0 la 7
    spec.Hide();
    int i = btnCheck.Length;
    while (i > 0)
    {
        i--; btnCheck[i] = false; ;
    }
    btnCheck[0] = true;
    EnableDisable(false, false, false, true);

    cassette.Location = new Point(190, 70);
    this.Controls.Add(cassete);
    cassete.Show();
}
```

Logica buton Specializări (Secvență de cod 3).

Secvență de cod 3

```
private void btnSpec_Click(object sender, EventArgs e)
{
    cassette.Hide();
    studenti.Hide();
    cadre.Hide();
    int i = btnCheck.Length;
    while (i > 0)
    {
        i--; btnCheck[i] = false; ;
    }
    btnCheck[1] = true;
    EnableDisable(true, true, true, true);

    spec.Location = new Point(190, 70);
    this.Controls.Add(spec);
    spec.Show();
}
```

Logica buton Studenți (Secvență de cod 4).

Secvență de cod 4

```
private void button4_Click(object sender, EventArgs e)
{
    cassette.Hide();
    spec.Hide();
    cadre.Hide();
    int i = btnCheck.Length;
    while (i > 0)
    {
        i--; btnCheck[i] = false; ;
    }
    btnCheck[2] = true;
    EnableDisable(true, true, true, true);

    studenti.Location = new Point(190, 70);
    this.Controls.Add(studenti);
    studenti.Show();
}
```

Logica buton Profesori(Secvență de cod 5).

Secvență de cod 5

```
private void button3_Click(object sender, EventArgs e)
{
    cassette.Hide();
    spec.Hide();
    studenti.Hide();
    int i = btnCheck.Length;
    while (i > 0)
    {
        i--; btnCheck[i] = false; ;
    }
    btnCheck[3] = true;
    EnableDisable(true, true, true, true);

    cadre.Location = new Point(190, 70);
    this.Controls.Add(cadre);
    cadre.Show();
}
```

Exemplu de căutare în baza de date(Secvență de cod 6).

Secvență de cod 6

```
if (cassete.StrF() != "" && cassete.StrA() == "" && cassete.StrAn()
== "")
{
    SqlCommand cmd = new SqlCommand("select * from Facultate where
Facultate=@a and " + "[Ani de studiu - Licență]=@b", con);
    cmd.Parameters.AddWithValue("@a", cassete.StrF());
    cmd.Parameters.AddWithValue("@b", int.Parse(cassete.StrAn()));
    //cmd.ExecuteNonQuery();
    SqlDataAdapter adapter = new SqlDataAdapter(cmd);
    DataTable dt = new DataTable();
    adapter.Fill(dt);
    dgvMain.DataSource = dt; //preia afisarea
    con.Close();
}
```

Definirea tabelului Utilizatori in SQL(Secvență de cod 7).

Secvență de cod 7

```
create table Utilizatori
(
Tip nvarchar(20) not null,
[Nume utilizator] nvarchar(200),
[Parolă] nvarchar(200),
constraint c20 check (Tip='administrator' or Tip='secretar' or Tip='cadru
didactic'),
constraint pk20 primary key(Tip, [Nume utilizator], [Parolă])
);
```

Definirea tabelului Note in SQL(Secvență de cod 8).

Secvență de cod 8

```
create table Note
(
Nume nvarchar(50) not null,
Prenume nvarchar(60) not null,
[Număr matricol] nvarchar(10) not null,
Nota smallint not null,
[Disciplină] nvarchar(150) not null,
constraint fk100 foreign key([Disciplină]) references [DPM-DI Discipline
Licență] ([Disciplină]),
constraint fk97 foreign key(Nume, Prenume, [Număr matricol]) references
Student(Nume, Prenume, [Număr matricol])
);
```

Definirea tabelului Facultate in SQL(Secvență de cod 9).

Secvență de cod 9

```
create table Facultate
(
Denumire nvarchar(100) not null,
[Abreviere/Prescurtare] nvarchar(30) not null,
[Ani de studiu - Licență] smallint not null,
constraint pk primary key([Abreviere/Prescurtare]),
constraint c1 check ([Ani de studiu - Licență]>=3 and [Ani de studiu -
Licență]<=10)
);
```

Algoritm trigger pentru Studenti (Secvență de cod 10).

Secvență de cod 10

```
create trigger trg1_Student on Student after insert, update as
begin
    declare @IsValid bit = 1;
    declare cursor_name cursor for
    select Facultate from inserted;
    open cursor_name;
    declare @Facultate nvarchar(100);

    fetch next from cursor_name INTO @Facultate;

    while @@FETCH_STATUS = 0
    begin
        if not exists (select 1 from Facultate where [Abreviere/Prescurtare]
= @Facultate)
        begin
            set @IsValid = 0;
            break;
        end;

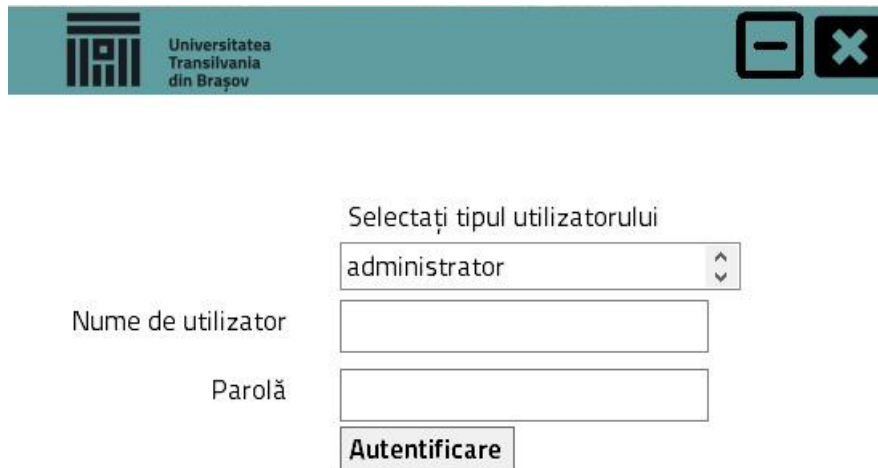
        fetch next from cursor_name INTO @Facultate;
    end;

    close cursor_name;
    deallocate cursor_name;

    if @IsValid = 0
    begin
        raiserror ('Invalid value for Facultate.', 16, 1);
        rollback transaction;
        return;
    end;
end;
```

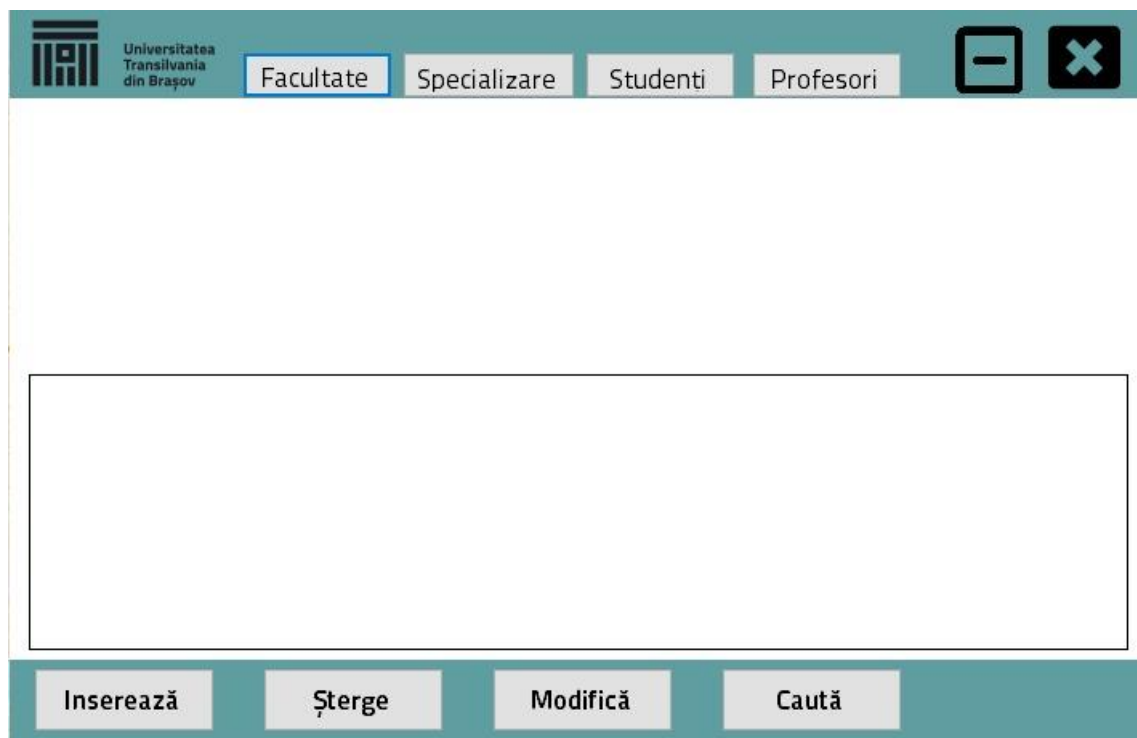

1.3. Manual de utlizare

Utilizatorul va accesa aplicație prin intermediul unei ferestre de logare. Acesta se poate autentifica ca administrator, secretar, cadru didactic fiecare în parte având acces la diferite comenzi.



The screenshot shows a login window with a teal header bar. On the left is the logo of Universitatea Transilvania din Brașov. On the right are window control buttons (minimize, maximize, close). The main area contains a dropdown menu labeled 'Selectați tipul utilizatorului' with 'administrator' selected. Below it are input fields for 'Nume de utilizator' and 'Parolă'. At the bottom is a button labeled 'Autentificare'.

Meniu de logare



The screenshot shows a main menu window with a teal header bar. On the left is the logo of Universitatea Transilvania din Brașov. On the right are window control buttons (minimize, maximize, close). Below the header is a row of buttons: 'Facultate' (highlighted), 'Specializare', 'Studenti', and 'Profesori'. The main area is a large empty rectangle. At the bottom is a row of buttons: 'Inserează', 'Șterge', 'Modifică', and 'Caută'.

Meniul principal

În acesta fereastră putem introduce o nouă facultate în baza de date, putem caută în baza de date o facultate deja existența , putem să modificăm să ștergem o anumită facultate.Când accesăm acest buton nu va fi nimic afișat , iar dacă apăsăm pe caută fără cerințe de selecție va selecta tot tabelul și îl va afișa.



Universitatea
Transilvania
din Brașov

Facultate Specializare Studenti Profesori

Facultate

Abreviere


Ani

	Denumire	Abreviere/Prescurt	Ani de studiu - Licență
▶	Facultatea de Al...	AT	4
	Facultatea de C...	Construcții	4
	Facultatea de D...	DMIL	4
	Facultatea de D...	DPM	4

Inserează Șterge Modifică Caută

Meniu Principal - Facultate

La fel că în fereastră Facultate, se pot adaugă , vizualiza sau caută diferiți studenți în funcție criteriile cerute.



Universitatea
Transilvania
din Brașov

Facultate Specializare Studenti Profesori

Nume

Prenume

Facultate


Specializare

	Număr matricol	Nume	Prenume	Inițială tată/tutore	CNP
▶	100M100	Ion	Popa	M	50201020139988
	100M101	Ionel	Popescu	L	50201020139988
	100M500	Ionelut	Plopu	M	50201020139988

Inserează Șterge Modifică Caută



Meniu Principal - Studenti

Aceeași funcționalitate ca în ferestrele precedente.



Universitatea
Transilvania
din Brașov

Facultate
 Specializare
 Studenti
 Profesori

Nume

Prenume

Post

	Nume	Prenume	Marcă Angajat	Titlu	Post
▶	Marius	Budăi	123ABC123F	dr. ing.	conferențiar uni...
	Dorin	Vasile	223ABC123F	dr. pr.	CDA
*					


Inseerează

Șterge

Modifică



Caută

Meniu Principal - Profesori



Universitatea
Transilvania
din Brașov

Facultate
 Specializare
 Studenti
 Profesori

Nume

Prenume

Post

	Nume	Prenume	Marcă Angajat	Titlu	Post
▶	Dorin	Vasile	223ABC123F	dr. pr.	CDA
*					

Inseerează

Șterge

Modifică

Caută

Meniu Principal - Profesori