

Colegiul Național „B.P. Hasdeu” Buzău

**Lucrare pentru obținerea atestatului profesional la
INFORMATICĂ**

Gestiunea datelor despre elevii unui liceu

**Profesor coordonator,
Olăroiu Nicolae**

**Realizator,
Sandu Victor Codrin
clasa a XII-a A**

mai 2021

Cuprins

1. Prezentarea generală a temei.....	3
2. Resurse hardware și software necesare	4
2.1. Scurtă descriere a limbajului C#.....	4
2.2. Scurtă descriere a programării orientate pe obiecte	6
2.3. Scurtă prezentare a serverului de baze de date MSSql Server	8
2.4. Resurse hardware necesare rulării proiectului.....	10
3. Realizarea aplicației.....	11
3.1. Structura bazei de date.....	11
3.2. Descrierea interfețelor	13
3.3. Descrierea modulelor de cod	18
4. Utilizarea aplicației	21
4.1. Utilizarea aplicației de către administrator.....	22
4.2. Utilizarea aplicației de către elev	26
5. Extinderi posibile ale aplicației.....	28
6. Bibliografie	29

1. Prezentarea generală a temei

Gestiunea datelor unui liceu reprezintă stocarea informațiilor referitoare la elevii liceului și accesarea acestora într-o manieră stabilită de programator. Stocarea datelor în format electronic este mult mai eficientă atât din punct de vedere al resurselor necesare, cât și din punct de vedere al vitezei cu care acestea pot să fie accesate. Datele despre elevi sunt atât cu caracter personal (precum numele, prenumele sau data nașterii), dar și informații referitoare la statutul școlar al elevilor.

Acest proiect a fost realizat cu scop didactic. Programul conține, la momentul actual, două interfețe ce afișează datele elevilor, le modifică sau permit introducerea acestora. Cele două interfețe sunt pentru administrator și pentru elev, fiecare fiind accesată în funcție de tipul de utilizator care este stocat în baza de date în tabela „Utilizatori”.

Prima interfață este cea afișată pentru utilizatorii de tip „Elev”, în care utilizatorul poate vedea notele elevului asociat contului, media acestuia la fiecare materie, media anuală a elevului și să modifice credențialele contului, fără posibilitatea de a modifica și statutul utilizatorului de la „Elev” la „Administrator”.

A doua interfață, cea afișată utilizatorilor de tip “Administrator”, atribuie utilizatorului control asupra informațiilor. Utilizatorul poate introduce datele unui elev, poate asocia elevilor introduși un cont de utilizator, poate nota elevii, poate modifica informațiile legate de aceștia sau anumite note existente și poate șterge din baza de date un elev sau un utilizator. Spre deosebire de utilizatorul de tip „Elev”, acesta poate modifica statutul unui utilizator de la „Elev” la „Administrator”.

2. Resurse hardware și software necesare

2.1. Scurtă descriere a limbajului C#

Una dintre cele mai mari probleme care a apărut odată cu dezvoltarea tehnologiei și introducerea unei varietăți din ce în ce mai mare de platforme și sisteme de operare diferite este portabilitatea codului. Limbajul C# a fost creat pentru a ajuta dezvoltatorii de aplicații să depășească această problemă. Cu toate că limbajul Java a fost primul care a oferit o soluție, acesta are câteva limitări. Una dintre acestea este interoperabilitatea limbajelor diferite sau programarea în limbaj mixt (posibilitatea codului scris într-un limbaj de a lucra în mod natural cu codul scris în alt limbaj). Interoperabilitatea limbajelor diferite este esențială la realizarea sistemelor software de dimensiuni mari.

Ca parte a ansamblului strategiei .NET, dezvoltată de Microsoft, în jurul anului 2000 a fost creat limbajul C#. C# este direct înrudit cu C, C++, dar și cu Java. Precursorul limbajului C# este C-ul. De la C, C# moștenește sintaxa, multe din cuvintele cheie și operatorii. De asemenea, C# construiește peste modelul de obiecte definit în C++. Relația dintre C# și Java este mai complexă. Java derivă la rândul său din C și C++. Ca și Java, C# a fost proiectat pentru a produce cod portabil. Limbajul C# nu derivă din Java. Relația dintre cele două provine din faptul că ele derivă din același strămoș, dar se deosebesc prin multe caracteristici importante.

Limbajul C# conține mai multe facilități inovatoare, dintre care cele mai importante se referă la suportul incorporat pentru componente software. C# dispune de facilități care implementează direct elementele care alcătuiesc componentele software, cum ar fi proprietățile, metodele și evenimentele. Poate cea mai importantă facilitate de care dispune C# este posibilitatea de a lucra într-un mediu cu limbaj mixt.

C# are o legătură deosebită cu mediul său de rulare, arhitectură .NET. Arhitectură .NET definește un mediu de programare care permite dezvoltarea și execuția aplicațiilor, indiferent de platformă. Aceasta permite programarea în limbaj mixt și oferă facilități de securitate și portabilitate a programelor. Deși Microsoft nu a implementat-o în totalitate decât pentru sistemele de operare Windows, arhitectura a fost concepută astfel încât să fie cross-platform. Astfel, ea poate fi implementată cu succes și pe alte sisteme de operare, păstrând compatibilitatea.

Legat de C#, arhitectură .NET definește două entități importante și anume biblioteca de clase .NET sau Framework Class Library (FCL) și motorul comun de programare sau Common Language Runtime (CLR).

C# nu are o bibliotecă de clase proprie, ci utilizează direct biblioteca de clase .NET. De exemplu, când se rulează un program care efectuează operații de intrare-ieșire, cum ar fi afișarea unui text pe ecran, se utilizează biblioteca de clase .NET.

Motorul comun de programare (CLR) se ocupă de execuția programelor C#. El asigură, de asemenea, programarea în limbaj mixt, securitatea și portabilitatea programelor. Atunci când este compilat un program C# sau un program în limbaj mixt, rezultatul compilării nu este un cod executabil. În locul acestuia, se produce un fișier care conține un tip de pseudocod numit limbaj intermediar sau pe scurt, IL (Intermediate Language). Acest fișier poate fi copiat în orice calculator care dispune de .NET CLR. Prin intermediul unui compilator denumit JIT (Just In Time), motorul comun de programare transformă codul intermediar în cod executabil. Procesul de conversie decurge astfel: atunci când un program .NET este executat, CLR activează compilatorul JIT. Compilatorul JIT convertește IL în cod executabil pe măsură ce fiecare parte a programului este necesară. În concluzie, orice program compilat până în format IL poate rula în orice mediu pentru care CLR este implementat, asigurându-se portabilitatea.

2.2. Scurtă descriere a programării orientate pe obiecte

Metodologiile de programare s-au modificat continuu de la apariția calculatoarelor pentru a ține pasul cu mărirea complexității programelor. Pentru primele calculatoare, programarea se făcea introducând instrucțiunile scrise în binar. Pe măsură ce programele au crescut, s-a inventat limbajul de asamblare, în care se puteau gestiona programe mai mari prin utilizarea unor reprezentări simbolice ale instrucțiunilor mașină. Cum programele continuau să crească, s-au introdus limbaje de nivel înalt, precum FORTRAN și COBOL, iar apoi s-a inventat programarea structurată.

POO a preluat cele mai bune idei de la programarea structurată, combinându-le cu concepte noi. A rezultat o modalitate diferită de a organiza un program. În fapt, un program poate fi organizat în două moduri: în jurul codului (mod de lucru descris de sintagma „codul acționează asupra datelor”, valabil în cazul programării structurate) sau în jurul datelor (abordare descrisă de sintagma „datele controlează accesul la cod”, valabilă în cazul programării orientate obiect).

Toate limbajele POO au patru caracteristici comune: încapsularea, polimorfismul, moștenirea și reutilizarea.

Încapsularea este un mecanism care combină codul și datele pe care le manipulează, menținând integritatea acestora față de interferența cu lumea exterioară. Când codul și datele sunt încapsulate, se creează un obiect. În cadrul unui obiect, codul și datele pot fi publice sau private. Codul și datele private sunt accesibile doar în cadrul aceluiași obiect, în timp ce codul și datele publice pot fi utilizate și din părți ale programului care există în afara acelui obiect. Unitatea fundamentală de încapsulare este clasa. Clasa specifică datele și codul care operează asupra datelor. O clasă definește forma unui obiect.

Polimorfismul este calitatea care permite unei interfețe să aibă acces la un grup generic de acțiuni. Termenul este derivat dintr-un cuvânt grecesc având semnificația “cu mai multe forme”. Spre exemplu, să presupunem că avem nevoie de o rutină care să returneze aria unei forme geometrice, care poate fi un triunghi, cerc sau trapez. Întrucât ariile celor trei forme se calculează diferit, rutina trebuie să fie adaptată la datele pe care le primește, astfel încât să distingă despre ce fel de formă este vorba și să returneze rezultatul corect.

Moștenirea este mecanismul prin care o clasă preia structura (datele membre) și comportamentul (metodele) unei alte clase, la care adaugă elemente specifice. Se definesc conceptele de clasă de bază (clasa de la care se preiau structura și comportamentul) și clasă derivată (clasa care preia structura și comportamentul). Avantajele oferite de acest mecanism sunt posibilitatea reutilizării codului și obținerea extensiei unei clase fără a fi necesară o recompilare a clasei inițiale.

Atunci când este creată o clasă, aceasta poate fi utilizată pentru a crea o multitudine de obiecte. Prin utilizarea moștenirii și a încapsulării, clasa amintită poate fi reutilizată. Nu mai este necesară testarea codul respectiv, ci doar utilizarea să corectă.

2.3. Scurtă prezentare a serverului de baze de date MSSql Server

Microsoft SQL Server este un sistem de gestionare a bazelor de date relaționale (RDBMS) produs de compania Microsoft. Limbajul de interogare este SQL, iar extensia procedurală este T-SQL.

Acesta suportă versiunea companiei Microsoft de SQL (Structured Query Language), cel mai răspândit limbaj de interogare pentru bazele de date. Este considerat un sistem de gestionare a bazelor de date pentru întreprinderi. El se poate aplica bazelor de date de dimensiuni foarte mari. Cu toate acestea, ultimele versiuni au venit cu mai multe tipuri de licențiere, de la cele free la unele avansate, astfel că în prezent se poate considera că SQL Server este potrivit pentru baze de date de orice dimensiune.

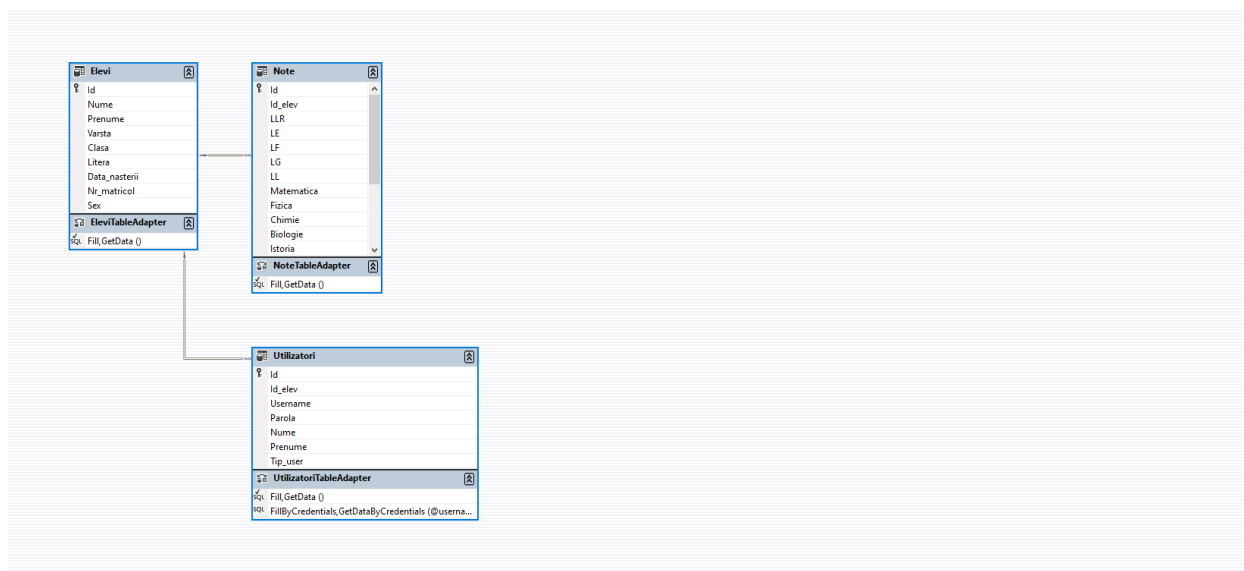
Codul de bază pentru Microsoft SQL Server își are originile în Sybase SQL Server și a reprezentat intrarea Microsoft pe piața bazelor de date relaționale, concurând cu Oracle, IBM și Sybase. Microsoft, Sybase și Ashton-Tate s-au unit pentru a crea și a scoate pe piață prima versiune numită SQL Server 4.2. Mai târziu Microsoft a negociat pentru drepturi de exclusivitate la toate versiunile de SQL Server concepute pentru sistemele de operare Microsoft. Sybase și-a schimbat ulterior numele în Adaptive Server Enterprise, pentru a evita confuzia cu Microsoft SQL Server.

SQL Server 7.0 a fost primul server de baze de date bazat pe o interfață grafică (GUI). O variantă de SQL Server 2000 a fost prima variantă comercială pentru arhitectura Intel. În anii următori s-au lansat și alte versiuni, ce au adus îmbunătățiri de performanță, unelte IDE și ETL, suport pentru date XML, geometrice sau spațiale, căutare în text, self-tuning, clustering (un cluster este un grup de 2 sau mai multe servere fizice, cu același nivel de acces la stocare, care au rolul de a asigura

resursele necesare stocării datelor din baza de date și care poartă denumirea de noduri) și HADR (high availability and disaster recovery).

Microsoft SQL Server folosește o variantă de SQL numită T-SQL, sau Transact-SQL, o implementare de SQL-92 (standardul ISO pentru SQL) cu unele extensii. T-SQL adaugă sintaxă adițională pentru procedurile stocate și pentru tranzacții. Standardele SQL necesită ACID: cele 4 condiții ce trebuie îndeplinite de orice tranzacție: atomicitate, consistență, izolare, durabilitate. MS SQL Server suportă ODBC (Open Database Connectivity).

Stocarea datelor se face pe modelul relațional, sub forma unei colecții de cu
tabele
rânduri



și coloane. Fiecare coloană este de un anumit tip: SQL Server suportă atât tipurile primare, precum întreg, zecimal, caracter, dată calendaristică, dar și tipuri mai complexe precum text, date binare, date geometrice, date spațiale, XML, etc.

2.4. Resurse hardware necesare rulării proiectului

Intel Core i5-3470 CPU @ 3.2GHz

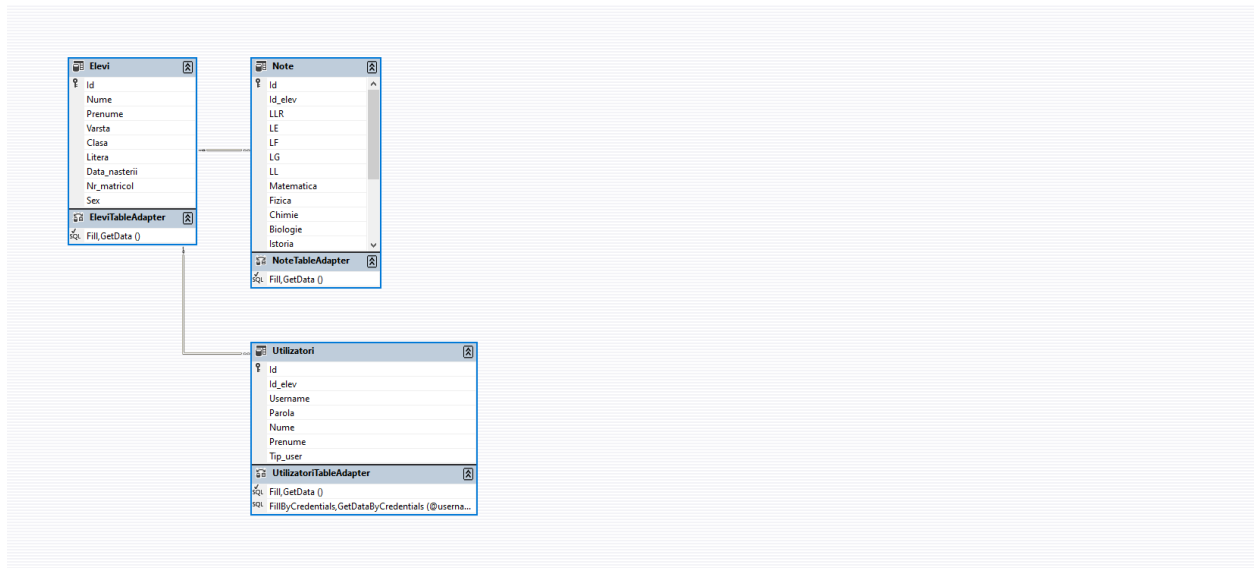
8.00 GB RAM

Nvidia GeForce GTX 860M

Stocare 512 GB SSD

3. Realizarea aplicației

3.1. Structura bazei de date



Name	Data Type	Allow Nulls	Default
Id	int	<input checked="" type="checkbox"/>	
Nume	varchar(50)	<input checked="" type="checkbox"/>	
Prenume	varchar(50)	<input checked="" type="checkbox"/>	
Varsta	int	<input checked="" type="checkbox"/>	
Clasa	int	<input checked="" type="checkbox"/>	
Litera	char(1)	<input checked="" type="checkbox"/>	
Data_nasterii	date	<input checked="" type="checkbox"/>	
Nr_matricol	varchar(10)	<input checked="" type="checkbox"/>	
Sex	int	<input checked="" type="checkbox"/>	

Keys (1)
<unnamed> (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

```
1 CREATE TABLE [dbo].[Elevi] (  
2     [Id] INT IDENTITY (1, 1) NOT NULL,  
3     [Nume] VARCHAR (50) NOT NULL,  
4     [Prenume] VARCHAR (50) NOT NULL,  
5     [Varsta] INT NOT NULL,  
6     [Clasa] INT NOT NULL,  
7     [Litera] CHAR (1) NOT NULL,  
8     [Data_nasterii] DATE NOT NULL,  
9     [Nr_matricol] VARCHAR (10) NOT NULL,  
10    [Sex] INT NOT NULL,  
11    PRIMARY KEY CLUSTERED ([Id] ASC)  
12 );
```

Tabela „Elevi” reține datele referitoare la elevi, atât cele cu caracter personal (Nume, Prenume, Vârsta, Data nașterii, Sexul), cât și cele necesare școlii (Clasa, Litera, Numărul matricol), iar coloana „Id” se generează automat la adăugarea unui rând nou.

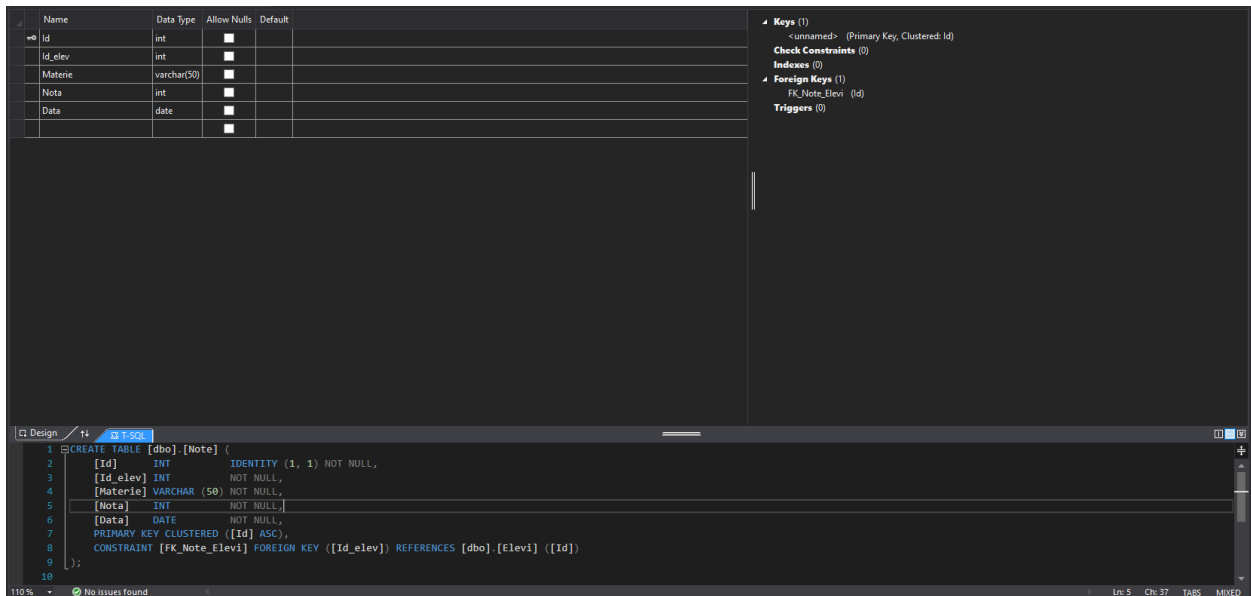


Tabela „Note” reține prin intermediul cheii străine „Id_elev” notele asociate unui elev din tabela „Elevi”, reținând atât valoarea notei, materia la care este pusă, cât și data la care este pusă.

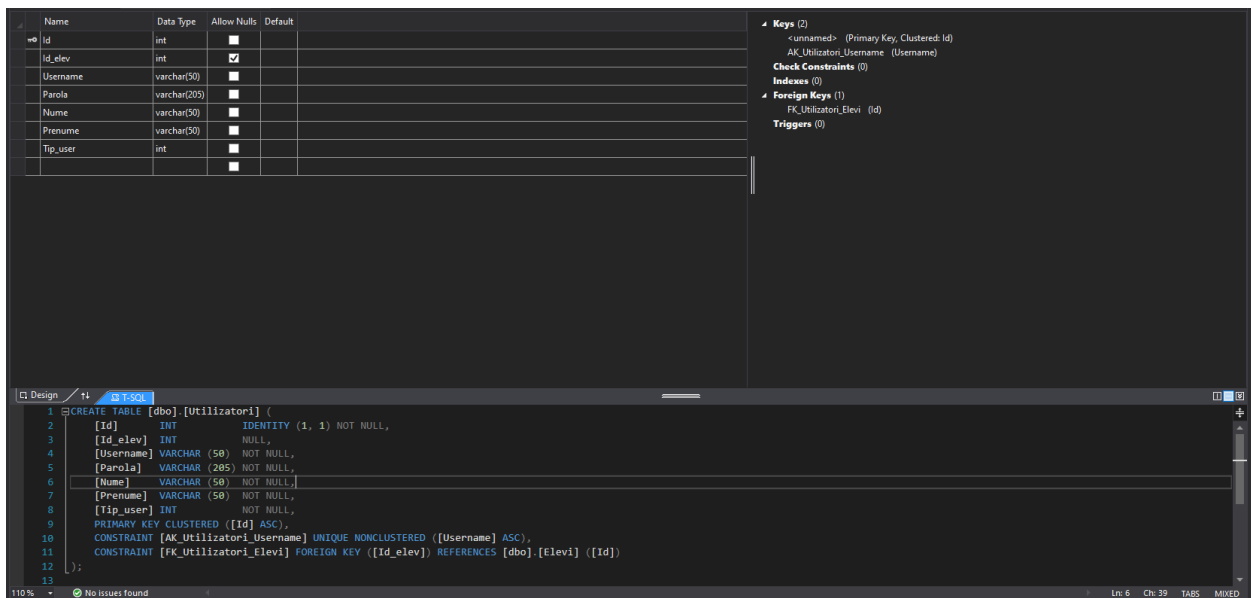
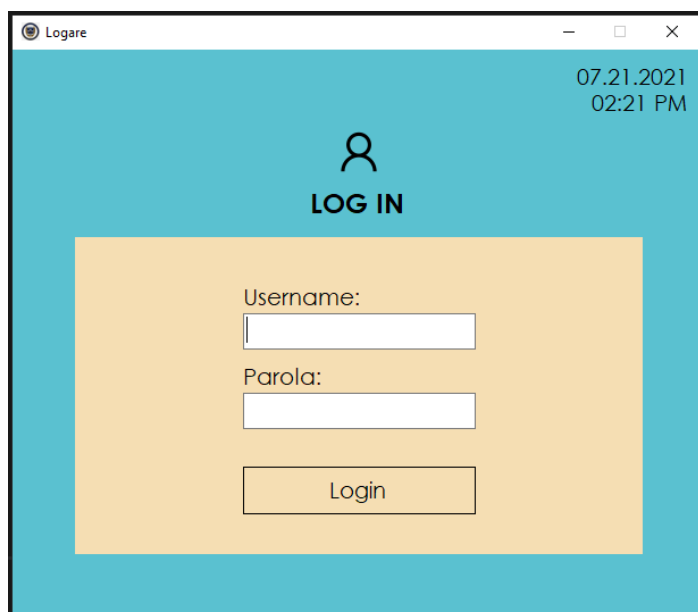
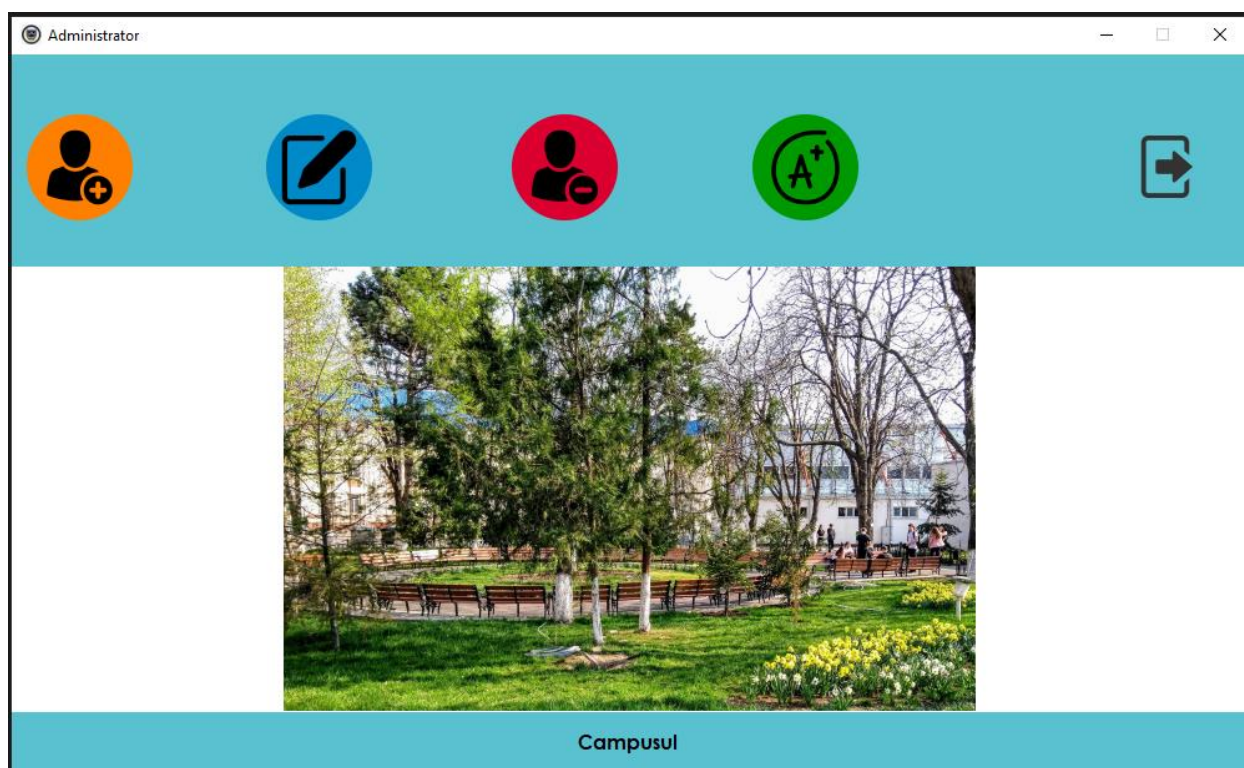


Tabela „Utilizatori” reține date despre utilizatori, precum username-ul și parola, numele și prenumele utilizatorului, dar și dacă acesta este sau nu administrator (2 – administrator, 1 – elev). În cazul în care nu este administrator, i se va asocia id-ul, numele și prenumele unui elev. Astfel coloana „Id_elev” este, ca în cazul tebelei „Note”, o cheie străină, ce leagă datele despre utilizator de datele unui elev, în cazul în care utilizatorul nu este administrator.

3.2. Descrierea interfețelor



Interfața de logare este prima care apare la pornirea programului, aici având data și ora, reactualizare în timp real, două casete text (pentru introducerea username-ului și pentru introducerea parolei) și butonul de login, la apăsarea căruia este verificat dacă username-ul și parola introduse sunt corecte și tipul de utilizator asociat acestora.



Interfața pentru utilizatorii de tip „Administrator” pune la dispoziție utilizatorului 4 butoane principale pentru: Adăugare (portocaliu), Modificare (Albastru), Ștergere (roșu) și Notare (verde) și încă un buton pentru ieșirea din aplicație. Deoarece butoanele nu au denumirea la vedere, în cazul în care cursorul

rămâne pe iconiță, în dreptul cursorului se va afișa denumirea acestuia. La apăsarea butonului pentru adăugare sub acesta se vor afișa alte două butoane, fiecare lansând altă interfață, la fel și în cazul butoanelor de modificare și ștergere. În interfață mai este și un slideshow de imagini cu locuri sau proiecte din liceu, slideshow-ul fiind însoțit de câte o denumire reprezentativă pentru fiecare imagine.

Interfețele de adăugare a unui elev nou sau a unui user nou sunt asemănătoare, având în componență câmpuri ce trebuie completate și care corespund coloanelor din tabelele din baza de date. Interfețele conțin obiecte precum TextBox-uri, ComboBox-uri și RadioButton-uri.

În interfața pentru adăugarea unui user dacă este apăsat butonul „Da” apare un ComboBox, ce este populat cu numele și clasa fiecărui elev din baza de date a școlii, iar la tastarea unor caractere, dacă acestea sunt regăsite în numele de familie

ComboBoxul este repopulat cu rezultatele găsite.

Interfețele de modificare sunt asemănătoare celor de adăugare/notare. În cazul interfeței de modificare a unui elev câmpurile sunt prepopulate cu informațiile actuale din baza de date, iar la apăsarea butonului „Modifică”

Modificare_Nota

Clasă

Materie

Litera

Nota veche

Elev

Noua notă

Modifică

informația din baza de date se modifică. În cazul modificării unei note trebuie completate câteva câmpuri, iar după completarea acestora ComboBox-ul „Nota veche” se va popula cu notele existente în baza de date, ce corespund câmpurilor completate.

După completarea corectă a unui câmp următorul poate să fie completat, până atunci acesta rămâne inactiv.

Modificare User

Informații generale

Nume

Sandu

Prenume

Victor Codrin

Tip user:

☒ Elev

☐ Admin

Credențiale

Username

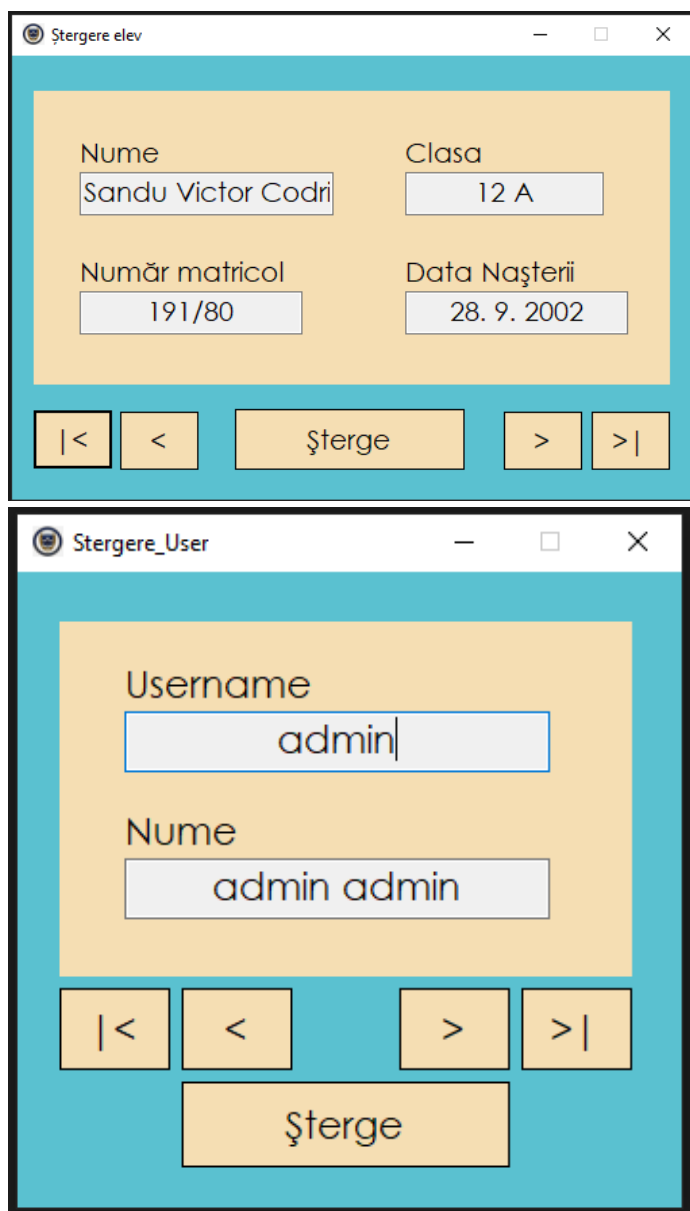
sandu_codrin

Parola nouă

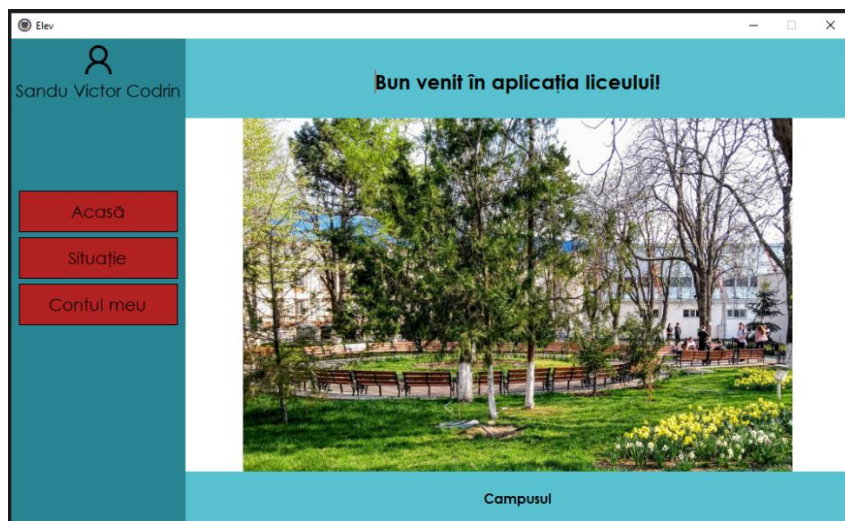
Confirmare parolă

<| < Modifică > >|

Interfața de modificare a unui utilizator permite modificarea numelui, prenumelui, în cazul modificării tipului de user, acesta poate fi modificat din elev în admin, dar nu și invers, pot să mai fie modificate și username-ul, dar și parola. Butoanele din colțul din dreapta jos sunt folosite pentru navigarea prin utilizatorii existenți în baza de date.



Interfețele pentru ștergere conțin mai puține TextBox-uri, acestea fiind inactive și populate cu datele stocate pe câte un rând din tabelele asociate din baza de date. În cazul interfeței de ștergere a unui elev câmpurile sunt populate cu numele și data nașterii, acestea fiind informații cu caracter personal și numărul matricol și clasa din care face parte elevul. În cazul interfeței de ștergere a unui utilizator câmpurile sunt completate cu username-ul și cu numele utilizatorului. În ambele interfețe la apăsarea butonului de ștergere utilizatorul este obligat să confirme sau să infirme selecția făcută, pentru a continua procedura de ștergere.



Interfața pentru utilizatorii elevi este la prima vedere asemănătoare cu cea a utilizatorilor de tip „Administrator”. Conține același slideshow și are cele trei opțiuni.

Butonul „Acasă” prezintă panel-ul care este afișat și la deschiderea interfeței.

Materie	Nota	Medie
Limba franceză	Nu au fost puse note	Medie:
Limba germană	Nu au fost puse note	Medie:
Limba latină	Nu au fost puse note	Medie:
Educație muzicală	Nu au fost puse note	Medie:
Educație plastică	Nu au fost puse note	Medie:
TIC	10	Medie: 10
		Media generală 8.67

La apăsarea butonului „Situație” este afișat panel-ul din imagine, care conține câte un GroupBox pentru fiecare materie, ce conține notele elevului la respectiva materie, media lor, iar în josul panel-ului media generală, calculată cu două zecimale.

Informații generale

Nume
Sandu Victor Codrin

Clasa
12 A

Număr matricol
191/80

Modificare credențiale

Modificare credențiale

Username

Parola actuală

Parolă nouă

Confirmare parolă

Când butonul „Contul meu” este apăsat panel-ul afișat mai sus apare și prezintă modalitatea de a modifica username-ul și parola utilizatorului conectat în acel moment, prezentând și numele, clasa și numărul matricol. Pentru modificarea credențialelor trebuie apăsat pe label-ul din dreapta jos din GroupBox-ul cu titlul „Informații generale”.

3.3. Descrierea modulelor de cod

Variabilele și liniile de cod ce sunt folosite pentru realizarea conectării la baza de date, în cazul preluării tuturor datelor dintr-un tabel și posibilitatea navigării între acestea sunt următoarele:

```
SqlConnection conn;
SqlDataAdapter da;
DataSet ds;
DataRow dr;
int rowNumber, currentRow;

string query = "SELECT e.* FROM Elevi e";
conn = new SqlConnection(Properties.Settings.Default.LiceuConnectionString);
da = new SqlDataAdapter(query, conn);
ds = new DataSet("Elevi");
da.Fill(ds, "Elevi");
currentRow = 0;
rowNumber = ds.Tables[0].Rows.Count;
```

În cazul algoritmului pentru modificarea unui elev, la apăsarea unui buton acesta modifică valoarea variabilei `currentRow` și apelează funcția `navigate()`, care populează câmpurile din interfață și care are următoarele linii de cod:

```
private void navigate()
{
    dr = ds.Tables[0].Rows[currentRow];

    numeTbModif.Text = dr.ItemArray[1].ToString();
    prenumeTbModif.Text = dr.ItemArray[2].ToString();
    varstaLABEL.Text = dr.ItemArray[3].ToString();
    clasaCB.Text = dr.ItemArray[4].ToString();
    literaCB.Text = dr.ItemArray[5].ToString();
    dataNasteriiDTPModif.Text = dr.ItemArray[6].ToString();
    nr_matricolTbMdif.Text = dr.ItemArray[7].ToString();
    if (dr.ItemArray[8].ToString() == "1")
    {
        mRbModif.Checked = true;
        fRbModif.Checked = false;
    }
    else
    {
        if (dr.ItemArray[8].ToString() == "2")
        {
            fRbModif.Checked = true;
            mRbModif.Checked = false;
        }
    }
}
```

La apăsarea butoanelor precum „Modificare” sau „Adăugare” sunt verificate campurile din interfață, iar în cazul în care unul nu este completat funcția `Completate()` de tip `bool` returnează false. Codul pentru funcție este:

```
private bool Completate()
{
    if(infGenNumeTB.Text.Length == 0)
    {
        label4.Visible = true;
        return false;
    }
    else
        if(infGenPrenumeTB.Text.Length == 0)
        {
            label7.Visible = true;
            return false;
        }
    else
        if(credUserTB.Text.Length == 0)
        {
            label8.Visible = true;
            return false;
        }
    else
        if(credParolaNouaTB.Text != credConfParolaTB.Text)
        {
            label30.Visible = true;
            return false;
        }
    return true;
}
```

Codul pentru apăsarea butonului „Modificare” este foarte asemănător cu cel de adăugare, diferența dintre ele fiind făcută la nivelul interogării SQL:

```
private void ModifBTN_Click(object sender, EventArgs e)
{
    if (Completate())
    {
        int GEN = 0;
        if (fRbModif.Checked == true)
            GEN = 2;
        else
            if (mRbModif.Checked == true)
                GEN = 1;
        string query = "UPDATE Elevi SET Nume = '" + numeTbModif.Text +
            "', Prenume = '" + prenumeTbModif.Text + "', Varsta = " + varstaLABEL.Text + ",
            Clasa = " + clasaCB.Text + ", Litera = '" + literaCB.Text + "', Data_nasterii =
            '" + dataNasteriiDTPModif.Value.Date + "', Nr_matricol = '" +
            nr_matricolTbMdif.Text + "', Sex = " + GEN + " WHERE Id = " +
            ds.Tables[0].Rows[currentRow].ItemArray[0].ToString() + ";";

        if (conn == null)
            conn = new
            SqlConnection(Properties.Settings.Default.LiceuConnectionString);
    }
}
```

```

        else
            if (conn.State == ConnectionState.Closed)
                conn.Open();
            SqlCommand comanda = new SqlCommand(query, conn);
            int rezultat = comanda.ExecuteNonQuery();
            if (rezultat > 0)
            {
                MessageBox.Show("Modificare cu success !", "ATENTIE!",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
                Modificare_Load(sender, e);
            }
            else
            {
                MessageBox.Show("Eroare la modificarea datelor !",
                "ATENTIE!", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }
        }
    }
}

```

Pentru a fi afișat lângă cursor un mic mesaj, în care să scrie ce reprezintă obiectul peste care este pus cursorul se folosesc următoarele linii de cod:

```

ToolTip toolTip1 = new ToolTip();

toolTip1.AutoPopDelay = 5000;
toolTip1.InitialDelay = 1000;
toolTip1.ReshowDelay = 500;

toolTip1.SetToolTip(this.pictureBox1, "Adăugare");
toolTip1.SetToolTip(this.pictureBox2, "Modificare");
toolTip1.SetToolTip(this.pictureBox5, "Ștergere");
toolTip1.SetToolTip(this.pictureBox4, "Notare");
toolTip1.SetToolTip(this.exitBtn, "Ieșire");

```

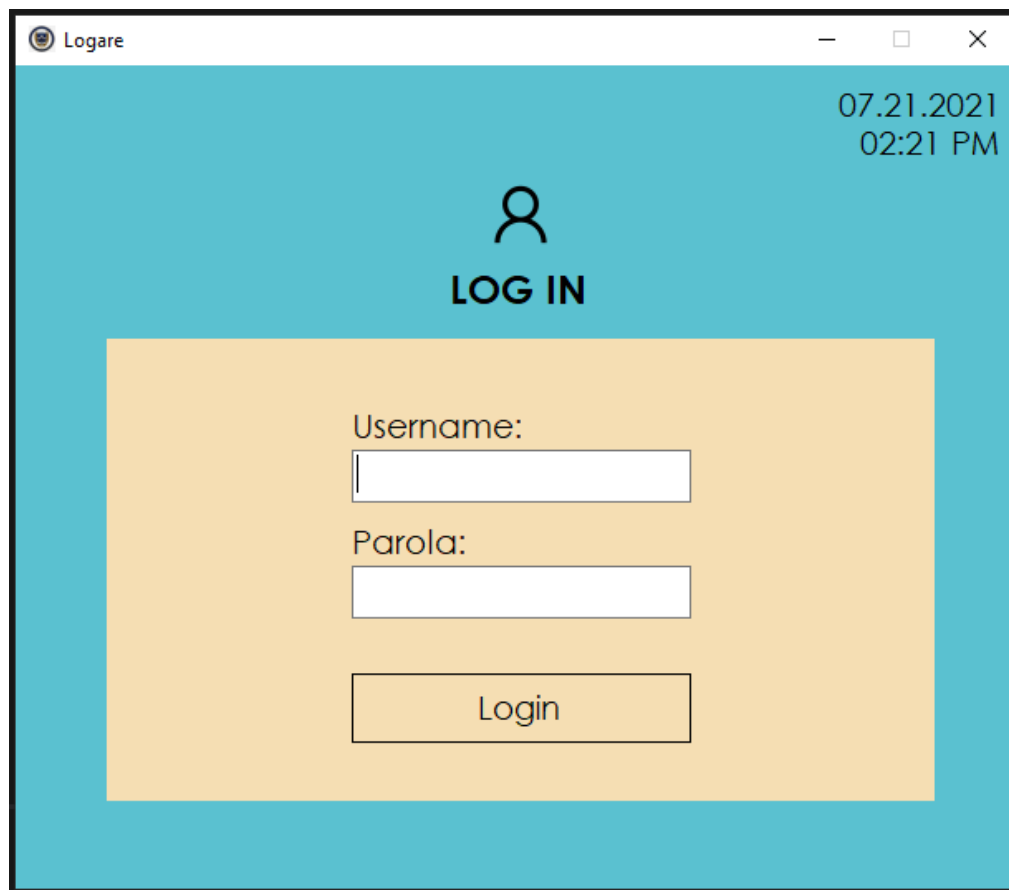
În cazul slideshow-ului de imagini, funcția LoadNextImage() asigură afișarea.

```

private void LoadNextImage()
{
    if (imageNumber == 7)
        imageNumber = 1;
    slidePic.ImageLocation = string.Format(@"images\{0}.jpg", imageNumber);
    imageNumber++;
    if (imageNumber == 2)
        imgTxt.Text = "Campusul";
    if (imageNumber == 3)
        imgTxt.Text = "Clubul treivirgulăpaispe";
    if (imageNumber == 4)
        imgTxt.Text = "Clubul de dezbateri";
    if (imageNumber == 5)
        imgTxt.Text = "Shin Dong-hyuk la Hasdeu";
    if (imageNumber == 6)
        imgTxt.Text = "Spectacolul Almost, Maine";
    if (imageNumber == 7)
        imgTxt.Text = "Concursul Hasdeu Open Tennis";
}

```

4. Utilizarea aplicației

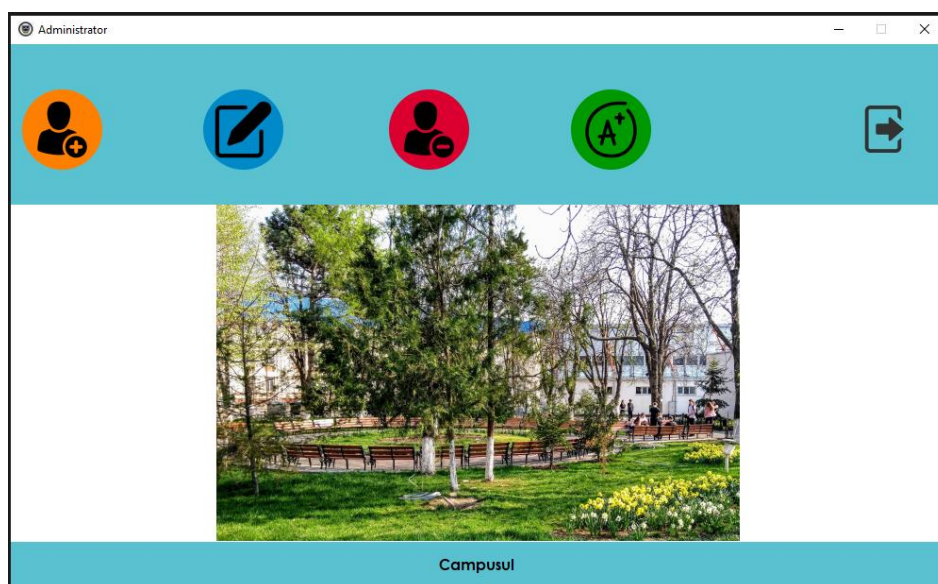


The screenshot shows a Windows-style application window titled "Logare". The window has a light blue background. In the top right corner, the date and time "07.21.2021 02:21 PM" are displayed. In the center, there is a black icon of a person and the text "LOG IN" below it. Below this, there is a white rectangular panel with a yellow background. Inside this panel, there are two text boxes: the first is labeled "Username:" and the second is labeled "Parola:". Below these text boxes is a yellow button with the text "Login".

La pornirea aplicației este deschisă interfața de logare, unde trebuie introduse datele contului unui utilizator. În cazul în care este apăsat butonul „Login” fără ca cel puțin unul din câmpuri să fie completat în marginea din dreapta jos a fiecărui TextBox apare un mesaj specific. Dacă utilizatorul introduce greșit credențialele, atunci pe ecran apare un MessageBox ce îl informează despre acest lucru, și TextBox-urile își golesc conținutul, acesta trebuind să ia de la capăt procedeul.

În colțul din dreapta sus al interfeței sunt afișate data și ora, iar în cazul apăsării pe iconița de utilizator sau pe label-ul de sub aceasta, panel-ul ce conține TextBox-urile unde trebuie introduse datele dispare și iconița, alături de label se mută în centrul interfeței. Dacă sunt apăstate iar, revin la poziția inițială și reappare panel-ul.

4.1. Utilizarea aplicației de către administrator



Interfața de mai sus prezintă cele patru mari acțiuni ce pot fi făcute de utilizator, prezentate și în capitolul „Descrierea interfețelor”, butonul de ieșire din aplicație, care și închide aplicația și slideshow-ul de imagini, însoțite de o denumire.

Pentru adăugarea unui elev trebuie apăsat butonul de adăugare și mai apoi cel pentru elev, fiind astfel ascunsă interfața de mai sus și afișată „Adăugare elev”.

În interfața de adăugare a unui elev trebuie completate câmpurile, acestea fiind obligatorii. Dacă este apăsat butonul „Adaugă”, fără ca toate câmpurile să fie completate, în dreptul câmpului apare un mic mesaj scris cu roșu, care anunță utilizatorul că acel câmp nu este completat sau că este completat eronat. Câmpurile care pot să fie completate în mod eronat sunt ComboBox-urile pentru clasă și literă, utilizatorul fiind nevoit să aleagă din opțiunile prestabilite.

În interfața de adăugare a unui utilizator nou trebuie completat prima dată câmpul referitor la tipul de utilizator. Dacă este apăsat butonul „Da”, atunci lângă acesta este afișat un ComboBox populat după cum este scris în capitolul „Descrierea interfețelor”. După

alegerea unui elev din listă, câmpurile „Nume” și „Prenume”, alături de „Tip de utilizator” sunt completate automat și nu pot să fie modificate. În cazul apăsării butonului „Nu” câmpurile „Nume” și „Prenume” sunt goale și trebuie completate.

La apăsarea butonului de adăugare dacă sunt câmpuri completate incorect sau necompletate se va afișa, ca și în cazul celorlalte interfețe, mesaje scrise cu roșu în dreptul câmpurilor.

Spre deosebire de restul interfețelor, cea de modificare a unei note prezintă inițial un singur câmp posibil de completat și nu permite completarea altuia până ce acela nu a fost completat în mod corect, dictând astfel ordinea completării câmpurilor.

Mesajele scrise cu roșu apar și în această interfață, dar doar la completare eronată a unui câmp. Această interfață este diferită de restul, deoarece în funcție de alegerile făcute la câmpurile anterioare, ComboBox-urile sunt populate diferit. În urma apăsării butonului „Modifică”, dacă toate câmpurile au fost completate corect, pe ecran apare un mesaj referitor la reușita modificării, apoi interfața revine la forma inițială, făcând astfel posibilă modificarea altei note.

În interfața de modificare a unui utilizator câmpurile din GroupBox-ul „Informații generale” sunt completate automat la navigarea prin datele despre utilizatori, cu ajutorul butoanelor „<|”, „<”, „>” și „>|”, dar și câmpul „Username”. La apăsarea „Modificare” este modificat utilizatorul, fiind memorat cu datele existente în acel moment în fiecare câmp. Dacă în câmpul „Parola nouă” nu sunt existente caractere, atunci parola rămâne neschimbată, în timp ce, dacă acolo există caractere, atunci parola se modifică în cea introdusă în câmpul „Parola nouă”, care trebuie să coincidă cu cea din câmpul „Confirmare parolă”, pentru a putea să fie memorată. La câmpurile referitoare la parolă avertismentul scris cu roșu apare doar dacă parolele din cele două câmpuri nu coincid. La restul câmpurilor apar mesajele doar în cazul necompletării.

În interfața de modificare a unui elev, la fel ca și în cea de modificare a unui user sunt folosite butoanele pentru navigarea prin datele stocate, de această dată fiind completate toate câmpurile. La apăsarea butonului „Modifică” sunt memorate datele în forma în care se afla în câmpurile specifice, doar dacă aceste câmpuri sunt completate corect, în caz contrar, în dreptul câmpurilor completate incorect, vor apărea mesaje care să anunțe acest lucru, mesaje ce dispar odată ce câmpurile sunt corect completate.

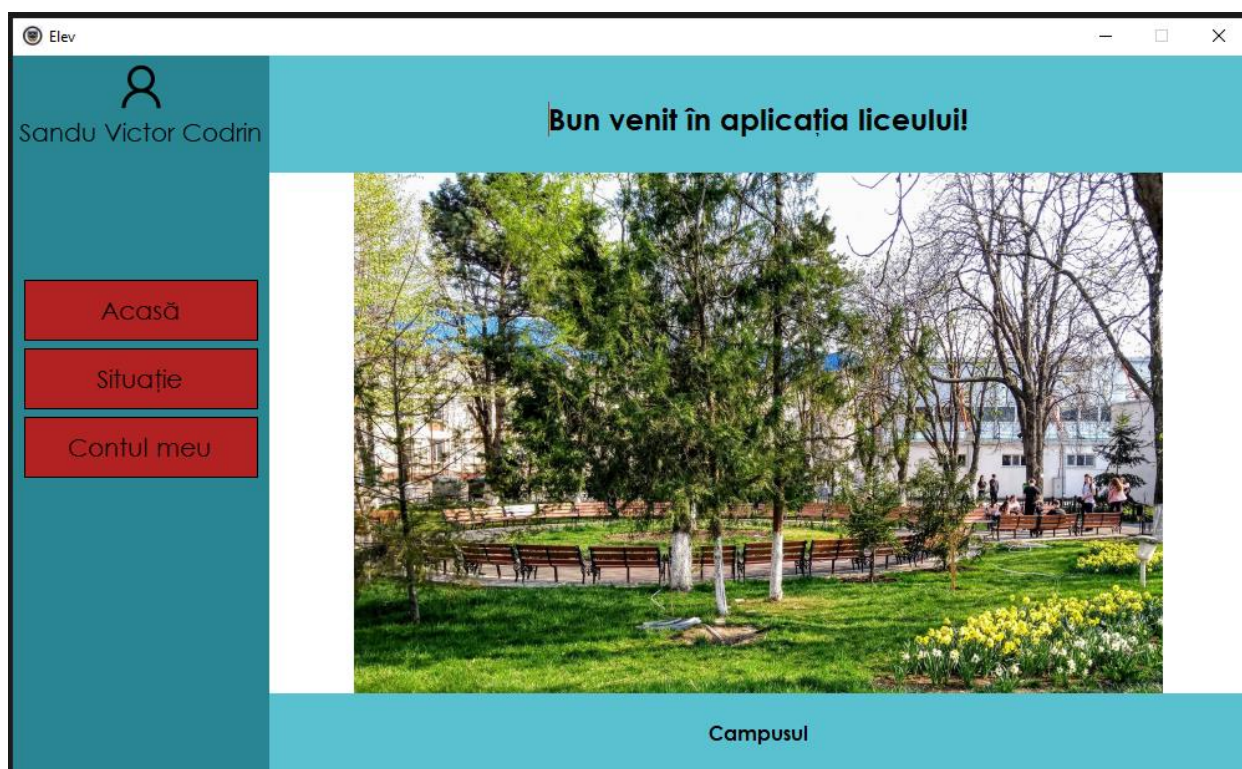
Ștergerea unui elev se face prin intermediul interfeței „Ștergere elev”, interfață ce prezintă patru câmpuri ce afișează informații referitoare la câte un elev. Navigarea pentru aceste informații se face la fel ca în cazul interfețelor pentru modificare, prin

butoanele din josul interfeței. Câmpurile din interfață nu pot fi modificate, având doar rolul de a afișa informația. La apăsarea butonului „Șterge” utilizatorului îi apare pe ecran un MessageBox pentru confirmare, iar în cazul confirmării mai apare un MessageBox pentru a anunța dacă ștergerea a reușit. Odată cu ștergerea unui elev, dacă acesta avea cont de utilizator sau note puse acestea dispar din baza de date.

Interfața de ștergere a unui utilizator funcționează după exact același principiu cu cea de ștergere a unui elev, având două câmpuri ce afișează username-ul și numele utilizatorului.

Interfața de notare este asemănătoare celei de modificare a unei note, fiecare câmp ce trebuie completat fiind utilizabil abia după completarea corectă a celor anterioare lui.

4.2. Utilizarea aplicației de către elev



Interfața afișată în cazul în care utilizatorul este elev conține același slideshow cu cel afișat în interfața pentru administrator, trei butoane pentru a schimba ceea ce este afișat în partea dreaptă. În colțul din stânga sus este afișat numele utilizatorului.



Odată cu apăsarea butonului „Situatie” în dreapta interfeței sunt afișate denumirile materiilor, notele, mediile și media generală, toate acestea neputând fi modificate în niciun fel.

Elev

Sandu Victor Codrin

Acasă

Situatie

Contul meu

Informații generale

Nume
Sandu Victor Codrin

Clasa
12 A

Număr matricol
191/80

Modificare credențiale

Modificare credențiale

Username

Parola actuală

Parolă nouă

Confirmare parolă

Când este apăsat butonul „Contul meu” partea dreaptă este din nou shimbată și permite vizualizarea informațiilor asociate contului. Interfața permite și modificarea credențialelor. Pentru a fi modificate, trebuie apăsat pe textul din colțul din dreapta jos al GroupBox-ului cu titlul „Informații generale”. După apăsare, câmpul „Username” este completa automat.

5. Extinderi posibile ale aplicației

Aplicația poate fi dezvoltată pe viitor pe mai multe planuri, atât prin îmbunătățirea celor existente, cât și prin adăugarea unor funcții noi.

În ceea ce privește funcțiile existente, primul pas ar fi îmbunătățirea logicii din spatele modificărilor credențialelor, atât modificarea din contul de administrator, cât și cea din contul de elev. Tot în cazul modificărilor posibile pentru interfețele existente ar fi și îmbunătățirea în ceea ce privește Frontend-ul.

În privința adăugării unor funcții noi, se poate introduce un nou tip de utilizator, tipul „Profesor”, care să aibă acces doar la interfețele de notare și modificare a unei note. O altă funcție ce poate fi atribuită programului este o modalitate de a conversa între utilizatori, atât de același fel, cât și de tipuri diferite.

6. Bibliografie

1. Crișan Daniela Alexandra, Programarea aplicațiilor folosind limbajul C# și platforma .NET, Ed Pro-Universitaria (2015);
2. Constantin Gălățan și Susana Gălățan, Curs de C# Programarea în Visual C# 2008;
3. Jack Purdum, Beginning Object-Oriented Programming with C#;
4. Programarea orientată pe obiecte și programarea vizuală, Microsoft.
5. <https://stackoverflow.com/>;
6. <https://ro.wikipedia.org/wiki/MySQL>;
7. https://ro.wikipedia.org/wiki/Programare_orientat%C4%83_pe_obiecte;