

Programville – Aplicație pentru testarea cunoștințelor

-secțiunea educațional-

Site creat de: **David - Robert Bratosin – clasa a XII - a A**
Colegiul Național B. P. Hasdeu Buzău

Profesor coordonator:
Nicolae Olăroiu

Contents

1. Prezentarea generală a temei.....	3
2. Descrierea tehnologiilor folosite.....	4
Limbajul C#.....	4
Scurta prezentare a serverului de baze de date MSSql Server	5
3. Resursele necesare rulării proiectului.....	7
4. Realizarea aplicației	8
Structura bazei de date.....	8
Descrierea interfeței	12
Descrierea modulelor de cod utilizate	21
5. Extinderi posibile ale aplicației	24

1. Prezentarea generală a temei

Realizarea unei aplicații de teste grilă pentru testarea cunoștințelor din diverse domenii oferă utilizatorului posibilitatea de a își extinde cunoștințele și de a se perfecționa în domeniul pe care îl activează. Aplicația poate fi folosită de școli până la companii ce doresc să crească performanța angajaților săi. Fiecare test poate fi personalizat de către un cont de administrator și încărcate spre rezolvare. Statistica, disponibilă atât administratorilor cât și utilizatorilor, oferă în detaliu cunoștințele care trebuie aprofundate.

Acest proiect a fost realizat cu scop didactic. Programul conține, la momentul realizării, posibilitatea de a crea și edita teste, de a organiza acestea în funcție de materii. Pe baza rezolvărilor realizate de utilizatori, statistica este construită atât pentru fiecare utilizator cât și pentru toți ceilalți. Fiecare test este prezentat sub forma unui card informativ, de unde utilizatorul poate vedea atât descrierea , cât și imaginea de prezentare. Astfel, interfața devine prietenoasă și captează atenția utilizatorului.

2. Descrierea tehnologiilor folosite

Limbajul C#

C# este un limbaj de programare orientat – obiect ce permite dezvoltatorilor să realizeze o suită de aplicații sigure și robuste ce funcționează pe .NET Framework. Acesta poate fi folosit pentru a realiza aplicații pentru Windows, mobil și site-uri web.

Mai multe caracteristici C# ajută la construirea de aplicații robuste și durabile: colectarea gunoiului recuperează automat memoria ocupată de obiecte neutilizate; gestionarea excepțiilor oferă o abordare structurată și extensibilă pentru detectarea și recuperarea erorilor; iar designul limbajului de tip type-safe face imposibilă citirea din variabile neinițializate, indexarea matricelor dincolo de limitele lor sau efectuarea de exprimări de tip necontrolate.

.NET, cadrul (Framework) folosit de limbajul C# permite realizarea, distribuirea și rularea aplicațiilor-desktop Windows și aplicațiilor WEB. Tehnologia .NET pune laolaltă mai multe tehnologii(ASP, XML, OOP, SOAP, WDSL, UDDI) asigurând portabilitatea codului compilat între diferite calculatoare cu sistem Windows, cât și reutilizarea codului în programe.

Această componentă este livrată împreună cu sistemul de operare Windows. De fapt, .NET 2.0 vine cu Windows Server 2003 și se poate instala pe versiunile anterioare, până la Windows 98 inclusiv; .NET 3.0 vine instalat pe Windows Vista și poate fi instalat pe versiunile Windows XP și Windows Server 2003.

Platforma comună de executare a programelor este numită Common Language Runtime(CLR). Aceasta se ocupă cu managementul și execuția codului scris în limbaje specifice .NET. CLR furnizează de asemenea servicii importante, cum sunt managementul memoriei și tratarea excepțiilor.

Scurta 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

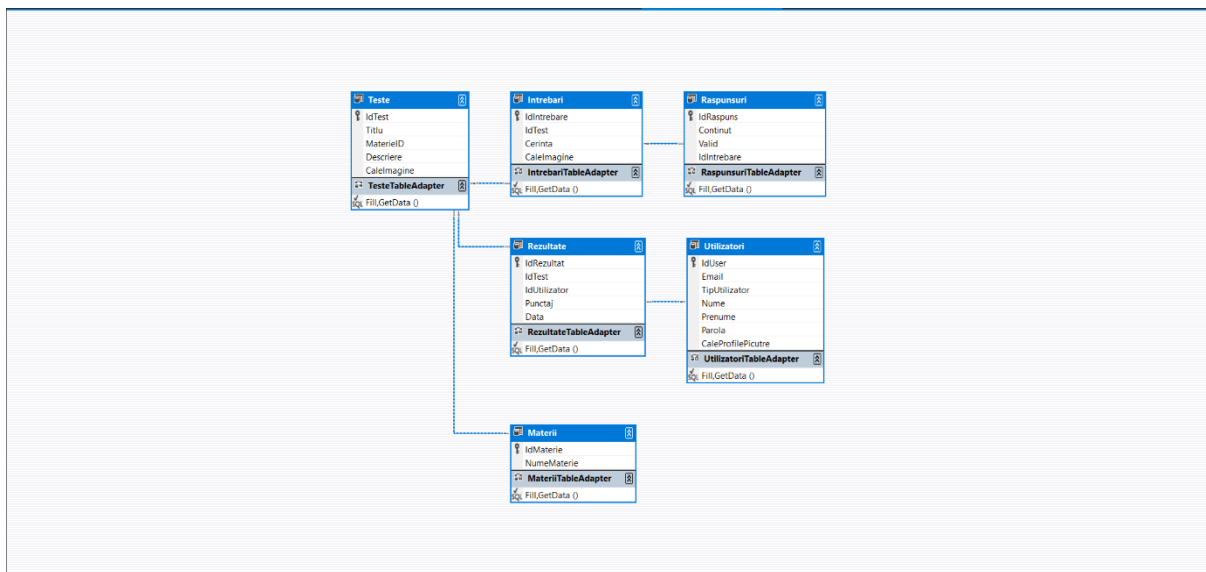
tabele cu rânduri și coloane. Fiecare coloană este de un anumit tip: SQL Server suportă atât tipurile primare, precum întreg, zecimal, caracter, dată calendaristică, dar și tipuri mai complexe precum text, date binare, date geometrice, date spațiale, XML, etc.

3. Resursele necesare rulării proiectului

- Sistem de operare: Windows 10
- Procesor: x86 sau x64
- RAM: 512 MB (minim), 1 GB (recomandat)
- Hard disk: 400MB necesari pentru instalare
- .NET Framework 4.7.2
- Microsoft SQL Server

4. Realizarea aplicației

Structura bazei de date



Pentru a realiza schema bazei de date și legăturile între tabele, folosim clasa „DataSet”, parte a arhitecturii .NET ce organizează baza de date în tabele ce pot fi modificate de către programator în timp real. Aceasta oferă programatorului clasa „TableAdapter” pentru fiecare tabel, obiect ce poate realiza conexiunea între baze de date și aplicație și permite modificarea tabelurilor. Astfel, putem insera, șterge, actualiza informațiile aflată în baza de date fără a mai fi necesar să scriem instrucțiunile SQL.

Observăm, de asemenea, și relațiile aflate între fiecare tabelă. De exemplu, fiecare utilizator poate avea o rezolvare a unui test, iar asocierea între informațiile rezolvării și utilizator se face prin coloana „IdUtilizator” din tabela „Rezultate” și coloana „IdUser” din tabela „Utilizatori”. În mod similar, tabela de „Răspunsuri” reține și punctajul obținut pentru fiecare test de către utilizator, prin asociația cu coloana „IdTest” din tabela „Teste” și coloana cu aceeași nume din tabela „Rezultate”.

Tabela „Utilizatori” reține datele despre fiecare utilizator ce folosește aplicația: numele, prenumele. Pentru determinarea tipului de cont (Utilizator sau Administrator) se folosește coloana „TipUtilizator” iar pentru imaginile de profil ale fiecăruia și parolă, se folosesc coloanele „Parola”, „CaleProfilePicutre”. Pentru a afișa imaginile de profil se reține calea relativă a acestora în directorul „profile_pictures”. Acestea sunt denumite în funcție de ID-ul fiecărui utilizator.

```
CREATE TABLE [dbo].[Utilizatori] (  
    [IdUser] INT IDENTITY (1, 1) NOT NULL,  
    [Email] NVARCHAR (MAX) NOT NULL,  
    [TipUtilizator] INT NOT NULL,  
    [Nume] NVARCHAR (MAX) NOT NULL,  
    [Prenume] NVARCHAR (MAX) NOT NULL,  
    [Parola] NVARCHAR (MAX) NOT NULL,  
    [CaleProfilePicutre] NVARCHAR (MAX) NULL,  
    PRIMARY KEY CLUSTERED ([IdUser] ASC)  
);
```

Tabela „Test” conține datele de prezentare pentru fiecare test ce poate fi rezolvat de către utilizator. Acestea sunt organizate în funcție de materie. Pentru aceasta, asociem fiecărei materii un ID pe care, ulterior, îl vom reține în această tabelă prin intermediul coloanei „MaterieID”. Asemănător imaginilor de profil ale utilizatorilor, coloana „CaleImagine” reține calea relativă către fiecare imagine de prezentare a testului. Acestea sunt denumite în funcție de ID-ul fiecărui test.

```

CREATE TABLE [dbo].[Teste] (
    [IdTest]          INT          IDENTITY (1, 1) NOT NULL,
    [Titlu]           NVARCHAR (MAX) NOT NULL,
    [MaterieID]       INT          NOT NULL,
    [Descriere]       NVARCHAR (MAX) NOT NULL,
    [CaleImagine]     NVARCHAR (MAX) NOT NULL,
    PRIMARY KEY CLUSTERED ([IdTest] ASC),
    CONSTRAINT [FkMaterie_Testes] FOREIGN KEY ([MaterieID]) REFERENCES [dbo].[Materii]
    ([IdMaterie])
);

```

Pentru afișarea întrebărilor din cadrul fiecărui test, acestea sunt reținute fiecare în tabela „Intrebari”. Pentru a asocia fiecare întrebare la testul corespunzător, ne folosim de coloana „IdTest” care reprezintă cheia primară a tabelului „Teste”. De asemenea, orice întrebare poate avea o imagine pentru a facilita explicarea întrebării utilizatorului. În mod similar, se reține calea relativă.

```

CREATE TABLE [dbo].[Intrebari] (
    [IdIntrebare] INT          IDENTITY (1, 1) NOT NULL,
    [IdTest]       INT          NOT NULL,
    [Cerinta]      NVARCHAR (MAX) NOT NULL,
    [CaleImagine]  NVARCHAR (MAX) NULL,
    PRIMARY KEY CLUSTERED ([IdIntrebare] ASC),
    CONSTRAINT [FK_Table_ToTable] FOREIGN KEY ([IdTest]) REFERENCES [dbo].[Teste]
    ([IdTest])
);

```

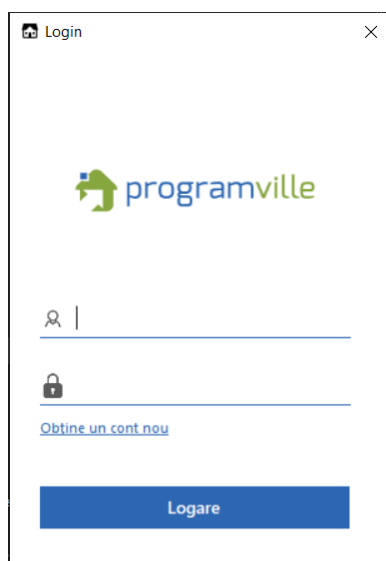
Asemănător asocierii întrebărilor cu testele corespunzătoare, fiecare răspuns este asociat unei întrebări. Astfel, prin intermediul tabelului „Raspunsuri”, se pot crea întrebări cu un număr variabil de răspunsuri.

```
CREATE TABLE [dbo].[Raspunsuri] (  
    [IdRaspuns] INT IDENTITY (1, 1) NOT NULL,  
    [Continut] NVARCHAR (MAX) NOT NULL,  
    [Valid] INT NOT NULL,  
    [IdIntrebare] INT NOT NULL,  
    PRIMARY KEY CLUSTERED ([IdRaspuns] ASC),  
    CONSTRAINT [FK_Raspunsuri_ToTable] FOREIGN KEY ([IdIntrebare]) REFERENCES  
    [dbo].[Intrebare] ([IdIntrebare])  
);
```

Într-un final, toate testele sunt grupate în funcție de materii. Fiecare materie are asociată un ID, pe baza căruia putem realiza o filtrare mult mai ușoară. Astfel, tabela „Materii” stochează fiecare materie și asociază acesteia un ID, necesar pentru identificarea fiecărui test.

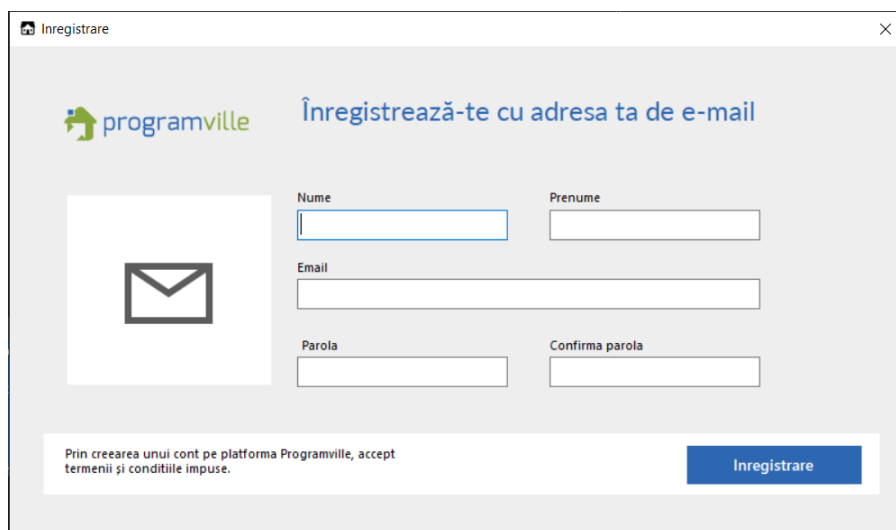
```
CREATE TABLE [dbo].[Materii] (  
    [IdMaterie] INT IDENTITY (1, 1) NOT NULL,  
    [NumeMaterie] NVARCHAR (MAX) NOT NULL,  
    PRIMARY KEY CLUSTERED ([IdMaterie] ASC)  
);
```

Descrierea interfeței



The screenshot shows a 'Login' window with the Programville logo at the top. Below the logo are two input fields: one for the username (indicated by a person icon) and one for the password (indicated by a lock icon). A blue button labeled 'Logare' is positioned below the password field. A link labeled 'Obține un cont nou' is located between the password field and the login button.

Panoul de autentificare. Cele 2 căsuțe de text (TextBox din arhitectura .NET) preiau adresa de e-mail și parola utilizatorului, verificând astfel identitatea acestuia. Pentru utilizatorii ce nu dețin un cont, hyperlink-ul aflat sub cele două căsuțe deschid formularul de înregistrare.

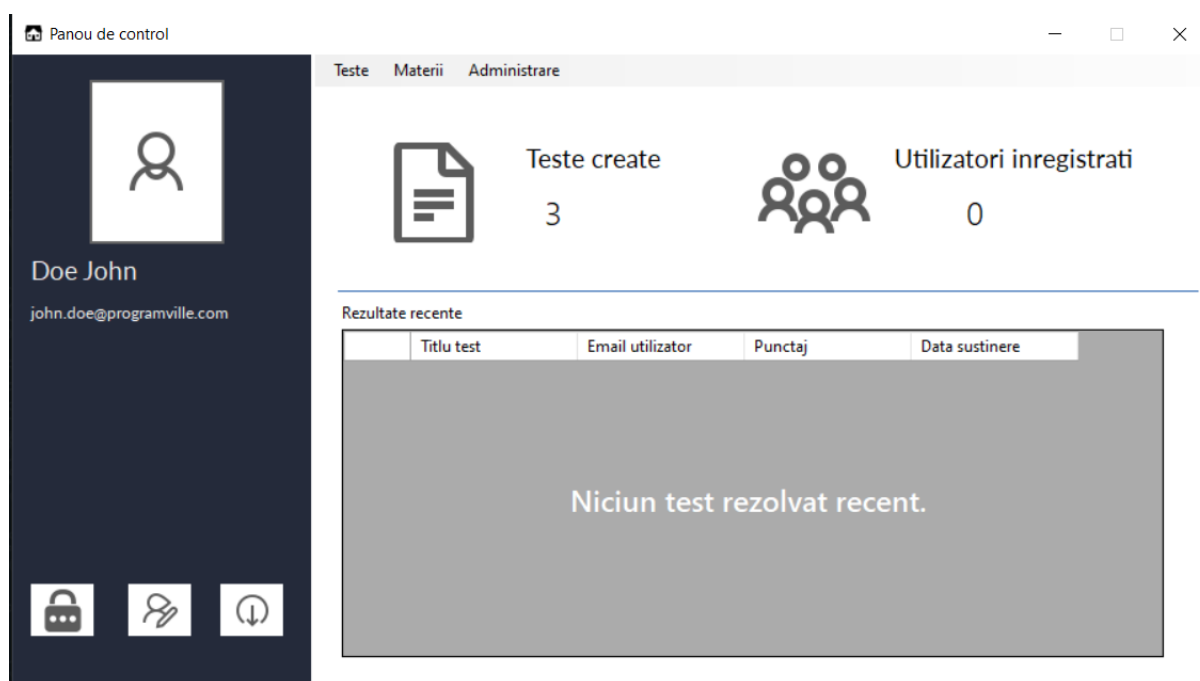


The screenshot shows an 'Înregistrare' (Registration) window. It features the Programville logo and the heading 'Înregistrează-te cu adresa ta de e-mail'. On the left is a large envelope icon. To the right are four input fields: 'Nume' (Name), 'Prenume' (First Name), 'Email', and 'Parola' (Password). The 'Parola' and 'Confirma parola' (Confirm password) fields are side-by-side. At the bottom left, there is a disclaimer: 'Prin crearea unui cont pe platforma Programville, accept termenii și condițiile impuse.' A blue button labeled 'Înregistrare' is at the bottom right.

Formularul de înregistrare preia datele esențiale identificării utilizatorului și parola pentru verificarea identității. Prin intermediul unui cont, fiecare utilizator are acces la testele publicate în aplicație și permite generarea

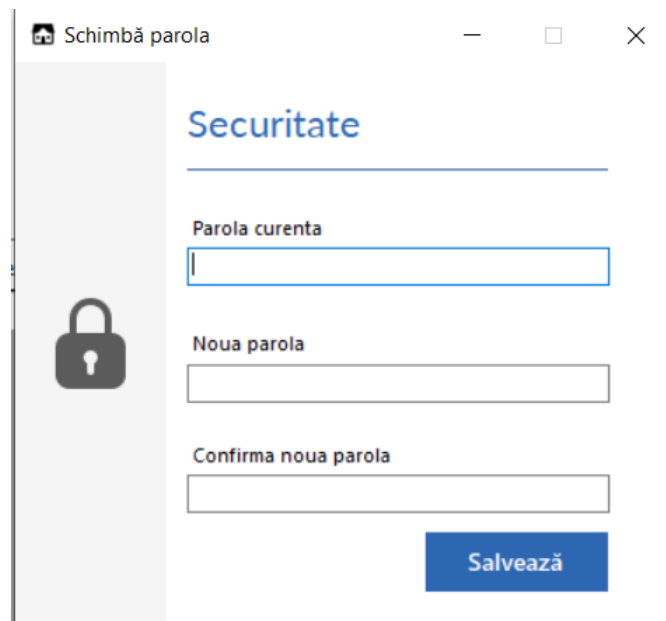
statisticilor pentru a observa performanța acestuia în funcție de fiecare test. Aceste informații se preiau în mod similar, prin intermediul căsuțelor de text.

După conectare, în funcție de tipul contului creat, se deschid meniurile corespunzătoare fiecăruia.



Interfața de administrare este împărțită prin intermediul unui „Split Container” în două panouri: lateral-stângă, ce deține informațiile și imaginea de profil asociate contului și cel lateral-dreapta, ce conține statistica și opțiunile de administrare ale platformei. Prin intermediul butoanelor, aflate în partea de jos, utilizatorul poate să schimbe parola, edita profilul său (schimbarea numelui, prenumelui și pozei de profil), iar panoul lateral-dreapta, meniul ce permit administrarea platformei și afișează statistica (câte teste au fost realizate, câți utilizatori sunt conectați și toate testele rezolvate recent”). Deconectarea se realizează prin intermediul celui de al treilea buton. Această acțiune mai poate fi realizată prin intermediul butonului de închidere a formularului. Înainte de orice acțiune importantă (deconectare, schimbarea parolei, profilului), aplicația

cere utilizatorului confirmarea. Pentru fiecare acțiune realizată, utilizatorul este informat.



The screenshot shows a window titled "Schimbă parola" (Change Password). On the left is a vertical grey bar with a black padlock icon. The main area has a blue header "Securitate" (Security). Below it are three input fields: "Parola curenta" (Current password), "Noua parola" (New password), and "Confirma noua parola" (Confirm new password). A blue "Salvează" (Save) button is at the bottom right.

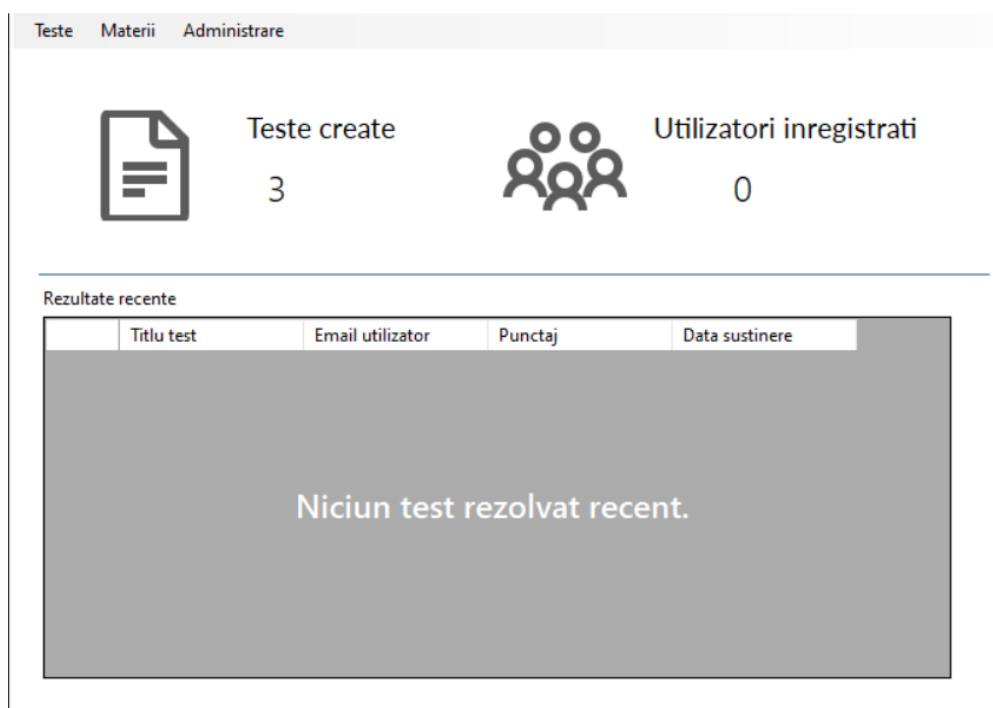
Pentru schimbarea parolei, fiecare utilizator confirmă identitatea prin reintroducerea parolei, prevenind astfel schimbările de către alți utilizatori.



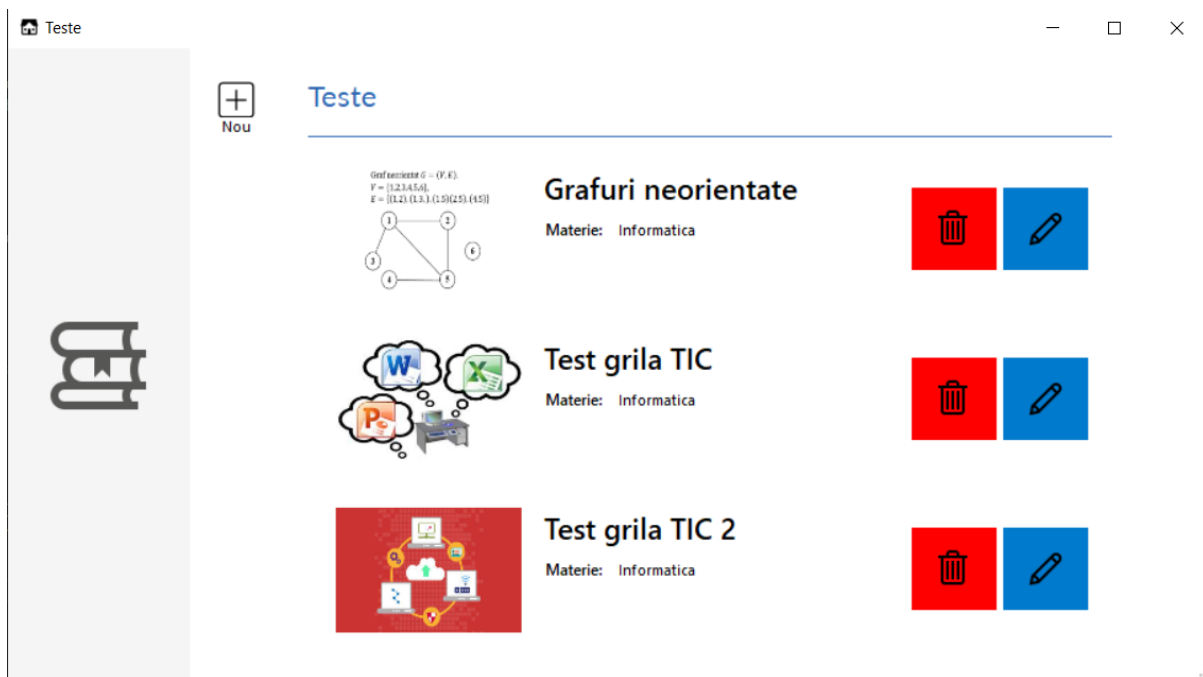
The screenshot shows a window titled "Editează profilul" (Edit Profile). On the left is a square placeholder for a profile picture with a person icon and a blue "Încarcă poză" (Upload photo) button below it. To the right are three input fields: "Nume" (Name) with "Doe", "Prenume" (First name) with "John", and "Email" with "john.doe@programville.com". At the bottom, there is a red warning message: "Din motive de securitate, schimbarea email-ului nu este posibilă!" (For security reasons, changing the email is not possible!). Next to it are two buttons: a red "Șterge cont" (Delete account) button and a blue "Salvează" (Save) button.

La editarea profilului, utilizatorul poate încărca o imagine de profil local care ulterior va fi salvată de aplicație în directorul „profile_pictures”. Fiecare imagine este identificată prin ID-ul utilizatorului din baza de date. De asemenea,

din acest formular, utilizatorul își poate șterge contul, eliminând de asemenea orice informație reținută de acesta (rezultate testelor, adresa de e-mail).



Panoul lateral-dreapta, acoperă majoritatea funcțiilor importante ale interfeței de administrare. În partea de sus, sunt prezente 3 butoane: "Teste" - ce deschide formularul de editare și creare de teste, "Materii" – ce permite adăugarea și editarea materiilor ce permit clasificarea testelor, iar butonul "Administrare" permite unui administrator să șteargă un utilizator sau să transforme contul acestuia în administrator. În centrul panoului, sunt prezentate statistica platformei: numărul de teste create și disponibile spre rezolvare de către utilizatori și utilizatorii înregistrați. Tabela de rezultate recente, aflată în partea de jos permite vizualizarea rezultatelor recente înregistrate în platformă.

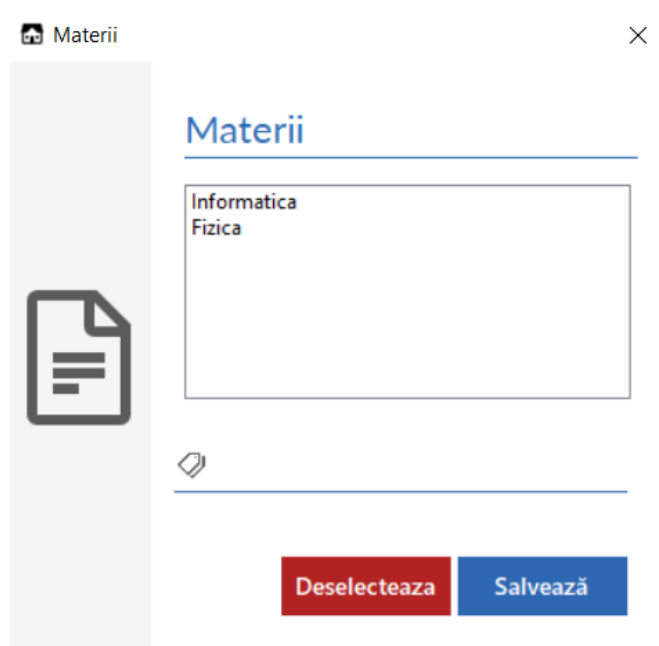


Acționarea butonului “Test” deschide interfața de editat fiecare test adăugat în platformă. Acestea sunt prezentate în centru sub forma unor carduri informative, ce dețin titlul și materia asociată fiecărui test. Acționarea butonului sub formă de creion, albastru, deschide formularul de editare al testului. Pentru adăugarea unui test, se poate acționa butonul din stânga sus denumit “Nou”. Butonul roșu permite ștergerea fiecărui test.

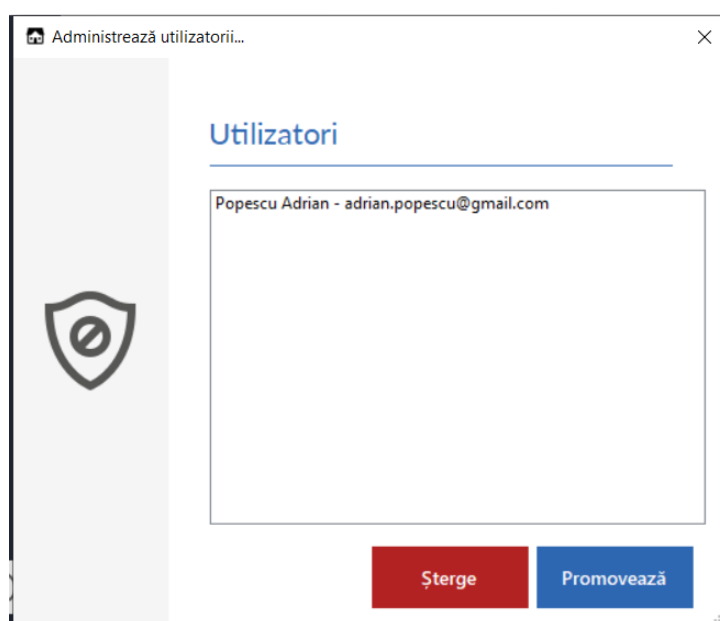
Acționarea butonului “Editare” sau “Nou”, deschide formularul de completare a informațiilor de prezentare a fiecărui test. În partea centrală se pot

adăuga titlul și descriere fiecărui test, pentru a facilita găsirea acestora de către utilizator. În partea dreaptă, se poate selecta materia din care face parte testul, iar mai jos se poate adăuga o imagine de prezentare a fiecărui test. Acționarea butonului continuă permite adăugarea de întrebări sau editarea acestora.

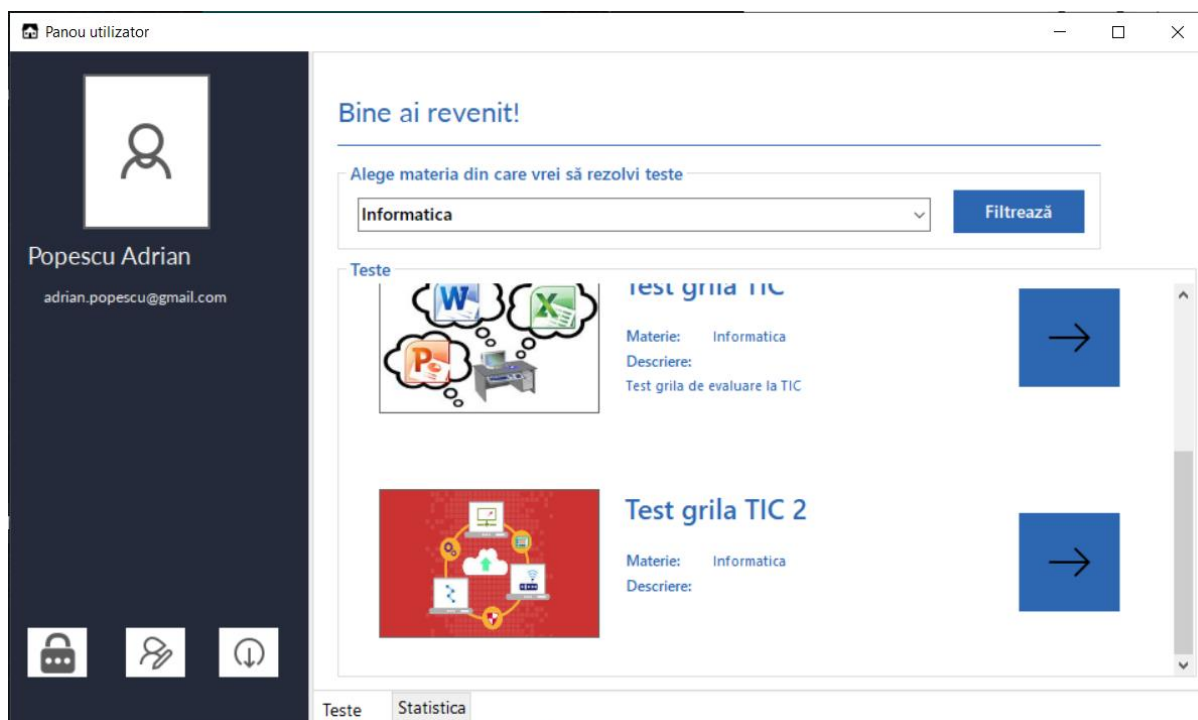
Formularul de editare a întrebărilor, permite vizualizarea și editarea fiecărei întrebări. Administratorul poate adăuga și șterge răspunsuri, poate realiza noi întrebări prin intermediul butonului “Nou” și poate încărca o imagine sugestivă fiecărei cerințe prin butonul “Încărcare”. Pentru salvarea fiecărei modificări realizate, utilizatorul trebuie să acționeze butonul “Salvare”, minimizând riscul unei erori umane. După finalizarea tuturor modificărilor, utilizatorul este obligat să apese butonul “Publică” confirmând astfel toate schimbările.



Acționarea butonului “Materii” deschide formularul prin intermediul căruia administratorul poate adăuga materii noi, facilitând organizarea testelor.



Administrarea utilizatorilor se poate realiza prin intermediul butonului “Administrare”. Aici, administratorul poate vedea fiecare utilizator. Asupra fiecăruia se pot realiza două acțiuni: ștergerea contului și promovarea acestuia la rolul de administrator.



În mod similar, meniul de utilizatori este împărțit în două panouri: în stângă se prezintă informațiile contului și imaginea de profil asociată, având în partea de jos acțiunile de editare ale contului, iar în dreapta se prezintă interfața de utilizator cu cele două opțiuni: vizualizarea și filtrarea testelor și statistica acestuia.



Fiecare test este prezentat sub forma unui card. Utilizatorul poate observa informațiile de prezentare ale fiecărui test. Acționarea butonului de tip săgeată, deschide formularul de rezolvare a testului.

Rezolvarea fiecărui test se realizează prin formularul prezentat mai sus. În partea de sus este prezentată cerința utilizatorului și variantele de răspuns adăugate de administrator. Pentru a confirma răspunsul, utilizatorul apasă butonul „Trimite răspunsul”. În momentul în care toate întrebările au fost rezolvate utilizatorul este informat de punctajul acestuia și rezultatul este disponibil în panoul „Statistici”.

The screenshot shows a web application window titled "Rezolva test". On the left is a vertical sidebar with a question mark icon. The main content area has a header "Grafuri neorientate" and "Întrebarea 1 din 7". Below this is a section labeled "Cerință" containing the question: "Care este numrul grafurilor parțiale ale unui graf neorientat cu n varfuri și m muchii?". There are four radio button options: $n!$, 2^n , $m!$, and 2^m . At the bottom left are left and right navigation arrows, and at the bottom right is a blue button labeled "Trimite răspunsul".

Descrierea modulelor de cod utilizate

Pentru a facilita conexiunea dintre baza de date și datele afișate în cadrul aplicației, fiecare înregistrare din tabela corespunzătoare este stocată sub forma unui obiect. Pentru conexiunea dintre baza de date și administrarea datelor din aceasta, a fost utilizată structura de date DataSet și obiectul TableAdapter, ce facilitează conexiunea dintre baza de date și programator. Astfel orice modificare asupra unui obiect se reflectă în baza de date, instrucțiunile SQL fiind generate de TableAdapter.

```
public Materie(int _idMaterie)
{
    MateriiTableAdapter materiiAdapter = new MateriiTableAdapter();
    DataTable materii = materiiAdapter.GetData();
    DataView viewMaterii = materii.DefaultView;

    viewMaterii.RowFilter = String.Format("IdMaterie = {0}", _idMateri
e);

    if (viewMaterii.Count == 1)
    {
        this.nume = Convert.ToString(viewMaterii[0]["NumeMaterie"]);
        this.idMaterie = Convert.ToInt32(viewMaterii[0]["IdMaterie"]);
    }
    else
        throw new Exception("Materia nu a putut fi initializata dupa I
D!");
}
```

Un exemplu este reprezentat de către extragerea unei materii din baza de date. Constructorul pentru materii preia ID-ul acesteia și extrage din tabela orice informație asociată cu aceasta. Dacă materia nu este găsită, automat raportăm eroarea programatorului ce folosește clasa respectivă.

```
public string Nume
{
    get { return nume; }
    set
    {
        this.nume = value;

        UtilizatoriTableAdapter utilizatoriTableAdapter = new UtilizatoriTableAdapter();

        Programville.MainDatabaseDataSet.UtilizatoriDataTable users = utilizatoriTableAdapter.GetData();

        users.Rows.Find(this.idUser)["Nume"] = Convert.ToString(nume);

        utilizatoriTableAdapter.Update(users);}
}
```

Asupra fiecărei înregistrări din baza de date, aceasta fiind stocată sub forma unui obiect, se pot folosi getteri și setteri din C#. Aceștia salvează local modificarea dar și actualizează coloana înregistrării corespunzătoare.

```
private void TestPreview_Load(object sender, EventArgs e)
{
    lbTitluTest.Text = crTest.Titlu;

    lbMaterie.Text = crTest.MaterieTest.Nume;

    lbDescriere.Text = crTest.Descriere;

    if(!String.IsNullOrEmpty(crTest.CaleImagine))

        picPreviewTest.Image = Image.FromFile(Application.StartupPath + crTest.CaleImagine);
}
```

Astfel, se pot afișa testele sub forma unei liste folosind controlul `FlowLayoutPanel` ce permite așezarea elementelor de sus în jos. Astfel, utilizatorul poate avea o previzualizare a fiecărui test: descrierea îi oferă un scurt rezumat asupra la ce va rezolva și imaginea de prezentare are scop de a capta atenția.

```
public void Delete()
{
    foreach (Intrebare intrebare in intrebari)
        intrebare.Delete();

    intrebari.Clear();

    TesteTableAdapter testeTableAdapter = new TesteTableAdapter();
    Programville.MainDatabaseDataSet.TesteDataTable teste = testeTable
Adapter.GetData();

    teste.Rows.Remove(teste.FindByIdTest(this.idTest));

    testeTableAdapter.Delete(this.idTest, this.materieTest.IdMaterie);
}
```

Fiecare înregistrare ștersă, îi se șterg și valorile asociate. De exemplu, dacă un test este șters de un administrator, întrebările și răspunsurile sunt automat eliminate din tabelele unde sunt salvate.

5. Extinderi posibile ale aplicației

Din punct de vedere al securității aplicației, se poate realiza un sistem de tip CAPTCHA („Completely Automated Public Turing test to tell Computers and Humans Apart”), prin intermediul căruia se poate determina dacă utilizatorul este o persoană sau un program de calculator. Astfel, se pot preveni încercările de atac tip brute-force, metodă prin care se generează pe baza unui model parole, astfel aflându-se parola utilizatorului ce este atacat.

De asemenea, realizarea unui sistem de criptare a parolei poate împiedica încercarea aflării parolei unui utilizator de către un potențial atacator, prin intermediul algoritmului SHA-256. Astfel, parola devine complexă, iar oricine are acces la baza de date nu va putea afla parola pentru a accesa un cont din aplicație.

În ceea ce privește statistica, se pot genera grafice prin intermediul cărora utilizatorul își poate analiza performanța sa în funcție de fiecare materie. Administratorii vor putea, de asemenea, să observe performanța fiecărui test realizat de aceștia și să ofere teste noi alternative pentru a aprofunda cunoștințele lipsă.

Socializarea reprezintă un factor important în dezvoltarea cunoștințelor, având posibilitatea de a cunoaște de la alți utilizatori, metode diferite de lucru și soluții ingenioase prezentate la problemele din teste. Pentru a facilita acest lucru, se poate realiza o funcție de mesagerie în cadrul aplicației pentru comunicarea eficientă între fiecare utilizator.

De asemenea, pentru a facilita aprofundarea cunoștințelor, se poate implementa o funcție distribuie de materiale de către administrator pentru fiecare materie. Utilizatorul le poate vizualiza în cadrul aplicației și descoperi noi cunoștințe care să îi îmbunătățească performanța sa.