



**UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA**

FACULTATEA DE AUTOMATICĂ ȘI
CALCULATOARE

PROIECT

INFORMATICĂ INDUSTRIALĂ

Aplicație pentru administrarea cursurilor universitare

Team Leader: Ungur Gelu

Developers: Szenasi Peter Bogdan

Țugui Ioan Vlad

Tester: Muntean Tiberiu

Grupa: 30131

24 mai, 2021

Cuprins

1	Introducere	3
1.1	Cerințele proiectului	3
1.2	Soluții alese	3
1.3	Obiective & specificații	3
2	Arhitectura aplicației	4
2.1	Diagrama claselor	4
2.2	Diagrama <i>use-case</i>	5
2.3	Diagrama bazei de date	6
3	Implementarea aplicației	7
3.1	Pagina de login	7
3.2	Pagina de înregistrare	8
3.3	Pagina <i>dashboard</i>	9
3.4	Materii	11
3.4.1	Lista materiilor	11
3.4.2	Vizualizare materie	13
3.4.3	Editare materie	14
3.4.4	Creare materie	15
3.5	Cereri de revendicare	16
3.6	Facultăți	17
3.6.1	Lista facultăților	17
3.6.2	Vizualizare facultate	18
3.6.3	Editare facultate	19
3.6.4	Creare facultate	20
3.7	Domenii	21
3.7.1	Lista domeniilor	21
3.7.2	Vizualizare domeniu	22
3.7.3	Editare domeniu	23
3.7.4	Creare domeniu	23
3.8	Specializări	24
3.8.1	Lista specializărilor	24
3.8.2	Vizualizare specializare	25
3.8.3	Editare specializare	26
3.8.4	Creare domeniu	26
3.9	Utilizatori	27
3.9.1	Lista utilizatorilor	27
3.9.2	Vizualizare utilizator	28
3.9.3	Editare utilizator & parolă	29

3.9.4	Creare utilizator	30
3.10	Roluri	31
3.10.1	Lista rolurilor	31
3.10.2	Creare & editare rol	32
4	Testarea aplicației	33
5	Anexe (Codul sursă)	34

1 Introducere

1.1 Cerințele proiectului

Se cere o aplicație complexă proiectată multinivel, cu nivelul de date în format .xml sau bază de date (cel puțin 3 structuri de date XML sau 3 tabele), cu nivel logic implementat fie în interiorul programului, fie ca serviciu web și cu nivelul prezentare în format ASP.net sau Windows Form.

1.2 Soluții alese

Pentru acest proiect, echipa noastră a ales dezvoltarea unui server de tip Webservice, clientul fiind de tip Windows Form Application.

Ca mediu de programare, am folosit **Microsoft Visual Studio 2019**, iar pentru versionarea codului și colaborare am folosit **GitHub**.

1.3 Obiective & specificații

Tema proiectului constă într-o aplicație pentru administrarea cursurilor dintr-o universitate. Accesul în aplicație se face printr-o bază de date de utilizatori, fiecărui utilizator atribuindu-se un rol prin care îi este permis să acceseze anumite funcționalități din aplicație.

Având o bază de date existentă cu materii din cadrul unei universități (cu posibilitate de modificare/adăugare de materii), profesorii au posibilitatea de a-și crea un cont utilizând adresa de e-mail instituțională (pentru a asigura autenticitatea persoanei). După crearea contului, profesorul poate trimite o cerere de revendicare pentru una sau mai multe materii. Cererea este înaintată spre secretariat, iar după ce este aprobată profesorul poate edita informațiile aferente materiei.

Principalele roluri sunt:

- **Administrator:** are acces la orice funcționalitate din aplicație, poate modifica orice conținut din cadrul bazei de date;
- **Secretar:** poate aproba/respinge cererile de revendicare pentru materiile din cadrul facultății din care face parte și poate edita materiile respective;
- **Profesor:** poate trimite cereri de revendicare pentru o anumită materie și poate edita datele aferente materiilor revendicate;
- **Guest** (fără cont): poate căuta și accesa paginile specifice materiilor pentru a vizualiza date despre respectiva materie.

2 Arhitectura aplicației

2.1 Diagrama claselor

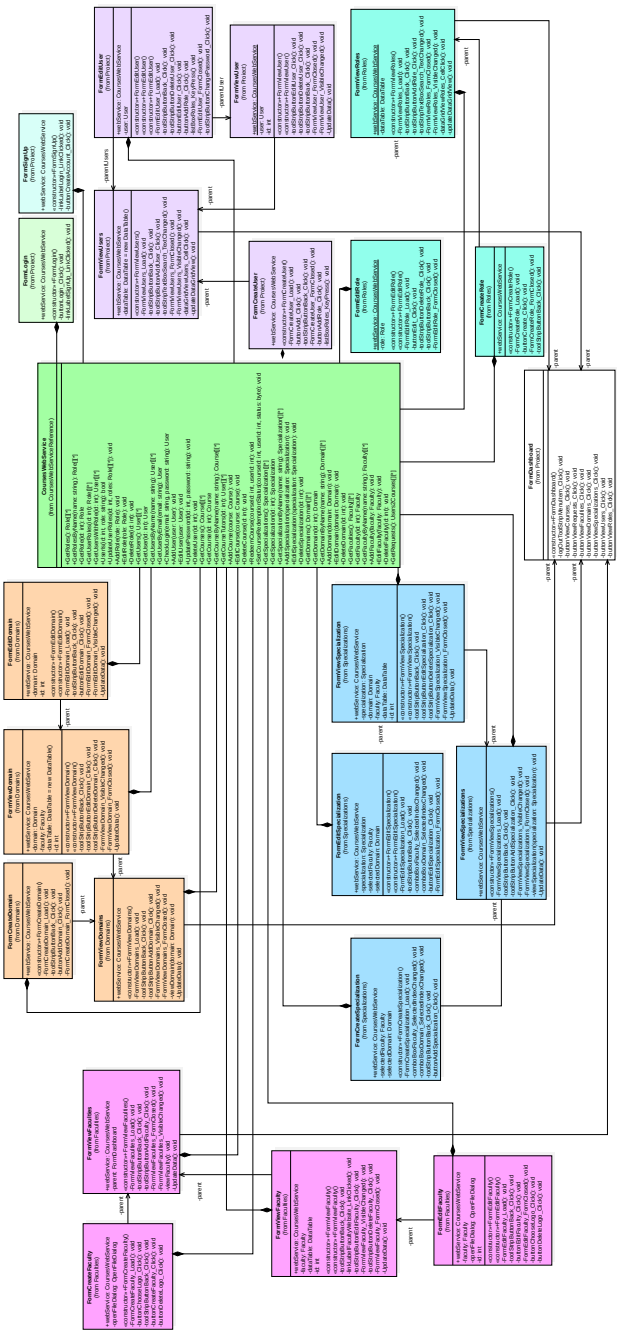


Figura 1: Diagrama claselor

2.2 Diagrama *use-case*

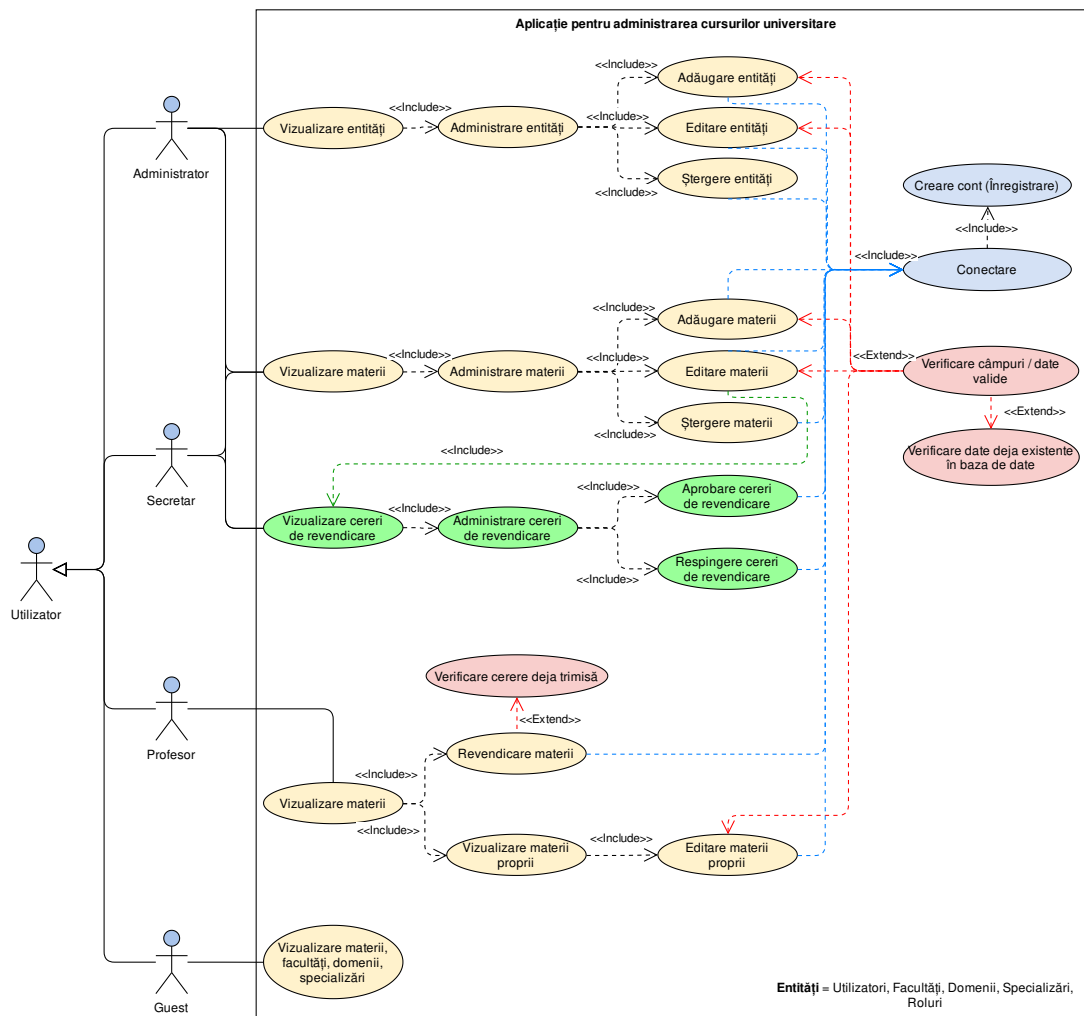


Figura 2: Diagrama *use-case* a aplicației

2.3 Diagrama bazei de date

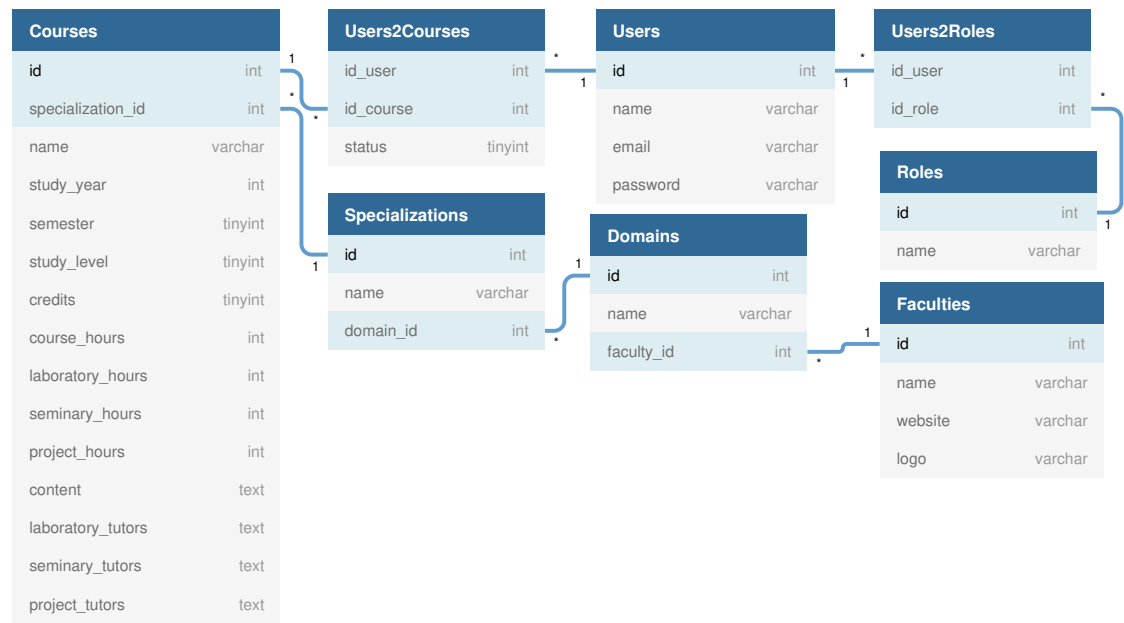


Figura 3: Diagrama bazei de date

3 Implementarea aplicației

3.1 Pagina de login

Pentru prima rulare aplicației, utilizatorul este întâmpinat de forma pentru **Conectare**. Aici, el are opțiunea de a se conecta cu o adresă de e-mail și parolă validă, existentă în baza de date.

Opțional, acesta poate bifa opțiunea *Rămâi conectat* pentru ca la următoarea rulare a aplicației, acesta să fie întâmpinat de forma pentru *Dashboard*, fără a mai fi nevoie de introducerea celor 2 tipuri de date pentru conectare. Pentru acest lucru, se salvează în setările interne ale aplicației dacă utilizatorul a bifat această opțiune, în acest caz salvându-se și *id*-ul lui în setări. Acest *id* se va prelua de fiecare dată când aplicația rulează pentru a realiza automat conectarea.

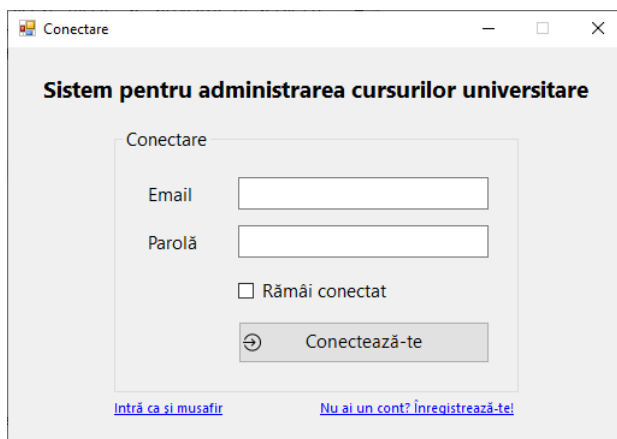


Figura 4: Forma pentru conectare

În cazul în care datele introduse sunt greșite, utilizatorul va fi întâmpinat de un mesaj de eroare.

Alternativ, dacă utilizatorul nu are un cont, poate da click pe un link care îi deschide o formă pentru crearea unui cont nou.

De asemenea, este posibilă conectarea fără un cont dând click pe **Intră ca și musafir**, dar vor fi disponibile doar vizualizările pentru materii, facultăți, domenii și specializări.

Metode din *web service* folosite:

1. **CheckLogin(email, password)**: verifică dacă combinația introdusă de email și parolă se regăsește în baza de date;

3.2 Pagina de înregistrare

Pentru a realiza înregistrarea unui utilizator nou în baza de date, acesta trebuie să completeze informații de bază precum **numele și prenumele**, **email-ul** și **parola**, cât și o **confirmare a parolei** pentru a evita introducerea unei parole greșite.

În ceea ce privește adresa de email, aceasta trebuie fie una **instituțională**, adică o adresă care să asigure autenticitatea profesorului care își creează contul. În cazul aplicației noastre, toate adresele de email trebuie să se termine în ***.utcluj.ro**, unde * este un subdomeniu.

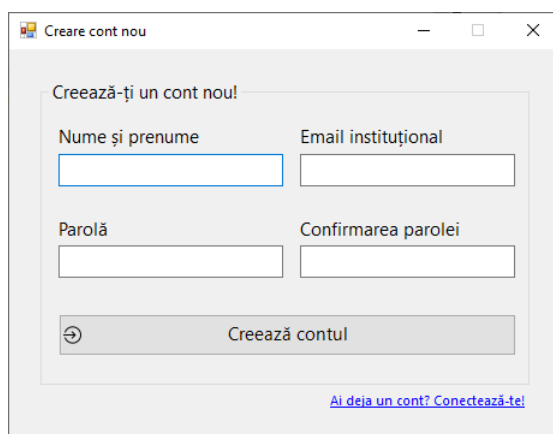


Figura 5: Forma pentru înregistrare

Odată creat un cont, datele sunt introduse în baza de date și pot fi folosite la conectare.

Metode din *web service* folosite:

1. **AddUser(user)**: introduce o nouă linie în baza de date;

Metode din *Utils* folosite:

1. **IsValidEmail(email)**: verifică dacă un email are o structură validă;
2. **EmailAlreadyTaken(email)**: verifică dacă un email există deja în baza de date;

3.3 Pagina *dashboard*

Această pagină poate fi considerată pagina principală a aplicației, de unde utilizatorii pot accesa toate modulele acesteia. În partea stânga, poate fi vizibil numele utilizatorului conectat, cât și email-ul acestuia.



Figura 6: Forma pentru *Dashboard*

Notă: Atât în figura de sus, cât și în figurile următoare, sunt vizibile toate acțiunile prezente în aplicație. În realitate, doar unele sunt accesibile de către utilizatori, în funcție de rolurile acestora.

Acțiuni disponibile:

1. **Vizualizare materii** (vizibilă tuturor utilizatorilor): deschide o formă care conține o listă cu toate materiile din baza de date și acțiunile aferente lor;
2. **Vizualizare cereri** (vizibilă utilizatorilor cu rolurile de **Administrator**, **Secretar** și **Profesor**): deschide o formă care conține o listă cu toate cererile de revendicare a materiilor, trimise de către profesori;
 - **Observație:** În cazul în care contul conectat are rolul de **Profesor**, în forma respectivă se vor afișa doar cererile acestuia, fără posibilitatea de a le aproba sau respinge.

3. **Vizualizare facultăți** (vizibilă tuturor utilizatorilor): deschide o formă care conține o listă cu toate facultățile din baza de date și acțiunile aferente lor;
4. **Vizualizare domenii** (vizibilă tuturor utilizatorilor): deschide o formă care conține o listă cu toate domeniile din baza de date și acțiunile aferente lor;
5. **Vizualizare specializări** (vizibilă tuturor utilizatorilor): deschide o formă care conține o listă cu toate specializările din baza de date și acțiunile aferente lor;
6. **Vizualizare facultăți** (vizibilă tuturor utilizatorilor): deschide o formă care conține o listă cu toate facultățile din baza de date și acțiunile aferente lor;
7. **Vizualizare utilizatori** (vizibilă utilizatorilor cu rolul de **Administrator**): deschide o formă care conține o listă cu toți utilizatorii din baza de date și acțiunile aferente lor;
8. **Vizualizare roluri** (vizibilă utilizatorilor cu rolul de **Administrator**): deschide o formă care conține o listă cu toate rolurile din baza de date și acțiunile aferente lor;

În fiecare dintre aceste forme există acțiuni care sunt, la rândul lor, disponibile unor utilizatori care au anumite roluri. Acestea vor fi detaliate în secțiunile dedicate în cele ce urmează.

Pe lângă butoanele menționate mai sus, există un control de tip **toolStripSplit-Button** care permite afișării opțiunii de *Log out (Deconectare)* odată cu apăsarea săgeții îndreptate în jos, cât și a unei opțiuni care permite editarea datelor pentru contul conectat (forma respectivă va fi detaliată ulterior în capitolul **Utilizatori**).

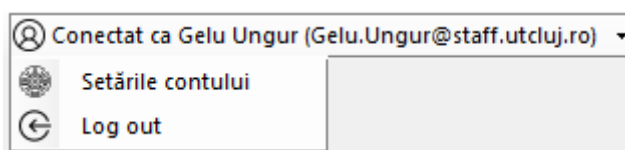


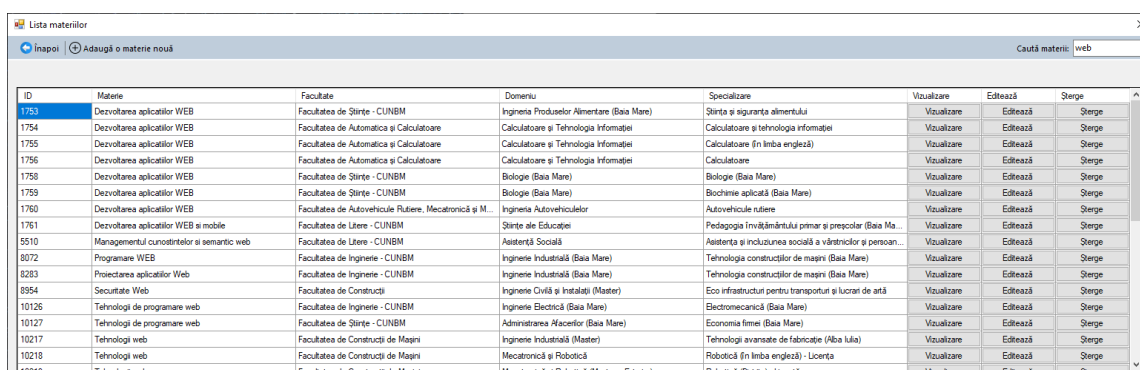
Figura 7: Opțiunile *Log out (Deconectare)* și *Setările contului*

Odată cu apăsarea acestui buton, utilizatorul conectat este deconectat, iar la următoarele rulări ale aplicației va fi nevoit să-și introducă email-ul și parola din nou în forma pentru conectare.

3.4 Materii

3.4.1 Lista materiilor

Această formă este afișată la apăsarea butonului *Vizualizare materii* din Dashboard. De aici, utilizatorii pot căuta anumite materii în funcție de un cuvânt sau o frază cu ajutorul `textBox`-ului din colțul din dreapta sus.



ID	Materie	Facultate	Domeniu	Specializare	Vizualizare	Editează	Șterge
1753	Dezvoltarea aplicațiilor WEB	Facultatea de Științe - CUNBIM	Ingineria Produselor Alimentare (Baia Mare)	Știința și siguranța alimentului	Vizualizare	Editează	Șterge
1754	Dezvoltarea aplicațiilor WEB	Facultatea de Automatică și Calculatoare	Calculatoare și Tehnologia Informației	Calculatoare și tehnologia informației	Vizualizare	Editează	Șterge
1755	Dezvoltarea aplicațiilor WEB	Facultatea de Automatică și Calculatoare	Calculatoare și Tehnologia Informației	Calculatoare (în limba engleză)	Vizualizare	Editează	Șterge
1756	Dezvoltarea aplicațiilor WEB	Facultatea de Automatică și Calculatoare	Calculatoare și Tehnologia Informației	Calculatoare	Vizualizare	Editează	Șterge
1758	Dezvoltarea aplicațiilor WEB	Facultatea de Științe - CUNBIM	Biologie (Baia Mare)	Biologie (Baia Mare)	Vizualizare	Editează	Șterge
1759	Dezvoltarea aplicațiilor WEB	Facultatea de Științe - CUNBIM	Biologie (Baia Mare)	Biologie aplicată (Baia Mare)	Vizualizare	Editează	Șterge
1760	Dezvoltarea aplicațiilor WEB	Facultatea de Autovehicule Rutiere, Mecanică și M...	Ingineria Autovehiculelor	Autovehicule rutiere	Vizualizare	Editează	Șterge
1761	Dezvoltarea aplicațiilor WEB și mobile	Facultatea de Litere - CUNBIM	Științe ale Educației	Pedagogia învățământului primar și preșcolar (Baia Ma...	Vizualizare	Editează	Șterge
5510	Managementul cunoștințelor și semantic web	Facultatea de Litere - CUNBIM	Asistență Socială	Asistența și incluziunea socială a vârstnicilor și persoan...	Vizualizare	Editează	Șterge
8072	Programare WEB	Facultatea de Inginerie - CUNBIM	Inginerie Industrială (Baia Mare)	Tehnologia construcțiilor de mașini (Baia Mare)	Vizualizare	Editează	Șterge
8283	Proiectarea aplicațiilor Web	Facultatea de Inginerie - CUNBIM	Inginerie Industrială (Baia Mare)	Tehnologia construcțiilor de mașini (Baia Mare)	Vizualizare	Editează	Șterge
8954	Securitate Web	Facultatea de Construcții	Inginerie Civilă și Instalații (Master)	Eco infrastructură pentru transporturi și lucrări de artă	Vizualizare	Editează	Șterge
10126	Tehnologii de programare web	Facultatea de Inginerie - CUNBIM	Inginerie Electrică (Baia Mare)	Electromecanică (Baia Mare)	Vizualizare	Editează	Șterge
10127	Tehnologii de programare web	Facultatea de Științe - CUNBIM	Administrarea Afacerilor (Baia Mare)	Economia firmei (Baia Mare)	Vizualizare	Editează	Șterge
10217	Tehnologii web	Facultatea de Construcții de Mașini	Inginerie Industrială (Master)	Tehnologii avansate de fabricație (Alba Iulia)	Vizualizare	Editează	Șterge
10218	Tehnologii web	Facultatea de Construcții de Mașini	Mecanică și Robotică	Robotică (în limba engleză) - Licență	Vizualizare	Editează	Șterge

Figura 8: Forma pentru *vizualizarea materiilor*

În mod implicit, la deschiderea acestei forme nu sunt aduse materii deoarece poate exista riscul ca aplicația să se blocheze în cazul în care este un număr mare de cursuri (baza de date folosită la acest proiect are 10 000+ materii). În plus, la căutare, se pune un *delay* de 3 secunde înainte ca Server-ul să primească un *request* de a aduce spre Client rezultatele căutării pentru a evita o posibilă supraîncărcare a Server-ului. Acestei condiții i se adaugă o lungime minimă a cuvântului de căutare de **3 caractere** (*request*-ul nu este făcut decât dacă termenul de căutare are mai mult de 3 caractere).

Acțiuni disponibile:

1. **Înapoi:** închide forma curentă și o redeschide pe cea precedentă;
2. **Adaugă o materie nouă** (doar pentru utilizatorii cu rolurile de **Administrator** și **Secretar**): deschide o formă folosită la introducerea unei noi materii în baza de date;
3. **Căutare:** permite căutarea în baza de date după numele materiei;
4. **Vizualizare:** deschide o formă care conține informațiile materiei respective (în mod *read-only*);
5. **Editare** (doar pentru utilizatorii cu rolurile de **Administrator**, **Secretar** sau **Profesor**): deschide o formă care permite editarea informațiilor materiei;

- **Notă:** deși butonul este afișat și profesorilor, aceștia vor putea să editeze detaliile aferente materiei doar dacă le aparține (adică, au depus o cerere de revendicare iar aceasta a fost acceptată).
6. **Ștergere** (doar pentru utilizatorii cu rolurile de **Administrator** și **Secretar**): șterge materia respectivă;
- Înainte de ștergerea propriu-zisă, utilizatorul primește un mesaj de confirmare și este înștiințat de efectele acțiunii sale:

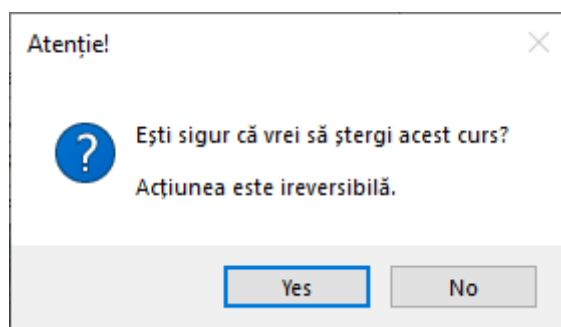


Figura 9: Mesaj de atenționare pentru ștergere

Metode din *web service* folosite:

1. **GetCoursesByName(searchTerm)**: returnează materiile care au în nume termenul de căutare;
2. **DeleteCourse(id)**: șterge materia cu *id*-ul specificat din baza de date;
3. **GetDomain(id)**: returnează domeniul cu *id*-ul specificat din baza de date;
4. **GetSpecialization(id)**: returnează specializarea cu *id*-ul specificat din baza de date;
5. **GetFaculty(id)**: returnează facultatea cu *id*-ul specificat din baza de date;

3.4.2 Vizualizare materie

În cadrul acestei forme, orice utilizator poate vedea informațiile cele mai importante legate de o anumită materie.

Vizualizare materie • Informatica aplicata

Înapoi Revendică materia Editează materia

Informații generale

Nume: Informatica aplicata
An studiu: Anul 3
Semestru: Semestrul 2
Nivel studiu: Licență
Credite: 3

Facultate: Facultatea de Automatică și Calculatoare
Domeniu: Ingineria Sistemelor
Specializare: Automatică și informatică aplicată

Conținutul cursului

1. Limbajul C#. Noțiuni de programare obiectuală
2. Mostenirea. Polimorfism. Clase abstracte. Interfete.
3. Delegați și evenimente
4. Programare multifilă
5. Lucrul cu fișiere. Serializare binară, xml.
6. Aplicații pentru lucrul cu baze de date.
7. Tranzacții
8. Migrarea bazelor de date
9. Servicii web
10. Arhitecturi multinivel. Stratul de date și logic.
11. Arhitecturi multinivel. Stratul prezentare

Curs

Ore curs: 75

Titulari: Honoriu Valean

Seminar

Ore seminar: 0

Titulari:

Laborator

Ore laborator: 42

Titulari: SL. dr. ing. Teodora Sanislav
As. dr. ing. Dan Gota
As. dr. ing. Adela Pop

Proiect

Ore proiect: 28

Titulari: SL. dr. ing. Teodora Sanislav
As. dr. ing. Dan Gota
As. dr. ing. Adela Pop

Figura 10: Forma pentru vizualizarea unei materii

Acțiuni disponibile:

1. **Înapoi**: închide forma curentă și o redeschide pe cea precedentă;
2. **Revendică materia** (doar pentru utilizatorii cu rolurile de **Profesor**): trimite o cerere de revendicare pentru materia respectivă;
 - **Notă**: În cazul în care există deja o cerere a utilizatorului respectiv pentru acea materie, butonul este ascuns.
3. **Editează materia** (doar pentru utilizatorii cu rolurile de **Administrator**, **Secretar** sau **Profesor**): permite editarea informațiilor despre materia respectivă;
 - **Notă**: În cazul profesorilor, butonul este vizibil doar dacă cererea lor de revendicare a fost acceptată.

Metode din *web service* folosite:

1. **GetCourse(id)**: returnează materia cu *id*-ul specificat;
2. **GetDomains()**: returnează o listă cu toate domeniile;

3. **GetSpecializations()**: returnează o listă cu toate specializările;
4. **GetFaculties()**: returnează o listă cu toate facultățile;
5. **GetCourseTutors(id)**: returnează o listă cu toți profesorii care au revendicat materia respectivă (titularii de curs);
6. **RedeemCourse(userID, courseID)**: adaugă o cerere de revendicare pentru materia respectivă;

3.4.3 Editare materie

În cadrul acestei forme, utilizatorii cu rolurile de **Administrator**, **Secretar** sau **Profesor** (cărora le-a fost acceptată cererea de revendicare) pot edita informațiile unei anumite materii.

The screenshot shows a web application window titled 'Editare materie • Informatica aplicata'. It contains several form sections:

- Informații generale**: Fields for 'Nume' (Informatica aplicata), 'An studiu' (Anul 3), 'Semestrul' (Semestrul 2), 'Nivel studiu' (Licență), 'Credite' (3), 'Facultate' (Facultatea de Automatica și Calculatoare), 'Domeniu' (Ingineria Sistemelor), and 'Specializare' (Automatică și informatică aplicată).
- Curs**: 'Ore curs' (75) and 'Titulari' (Honoriu Valean).
- Seminar**: 'Ore seminar' (0) and 'Titulari' (empty field with a '+' button).
- Laborator**: 'Ore laborator' (42) and 'Titulari' (SL. dr. ing. Teodora Sanislav, As. dr. ing. Dan Gota, As. dr. ing. Adela Pop).
- Proiect**: 'Ore proiect' (28) and 'Titulari' (SL. dr. ing. Teodora Sanislav, As. dr. ing. Dan Gota, As. dr. ing. Adela Pop).
- Conținutul cursului**: A list of 11 topics related to C# programming and database applications.
- Salvează modificările**: A button at the bottom to save changes.

Figura 11: Forma pentru editarea unei materii

Modificările se fac prin intermediul unor controale de tip **textBox**, **comboBox**, **listBox** sau **richTextBox**. O utilizare mai aparte este prezentă la titularii activităților de Seminar, Laborator și Proiect, unde numele lor sunt introduse manual din **textBox**-ul de sub **listBox**-ul corespunzător. Adăugarea în **listBox** se face prin apăsarea butonului marcat cu un plus. Pentru ștergerea unui anumit *item* din listă, se dă click pe el și se apasă tasta **backspace**.

Salvarea titularilor în baza de date se face într-un câmp de tip **varchar**, numele fiind despărțite printr-un caracter special și mai apoi prelucrate în vederea afișării lor în Client și stocării lor în baza de date. Salvarea modificărilor se face cu metoda din *web service* numită **EditCourse(course)**.

3.4.4 Creare materie

În cadrul acestei forme, utilizatorii cu rolurile de **Administrator** sau **Secretar** pot adăuga o nouă materie.

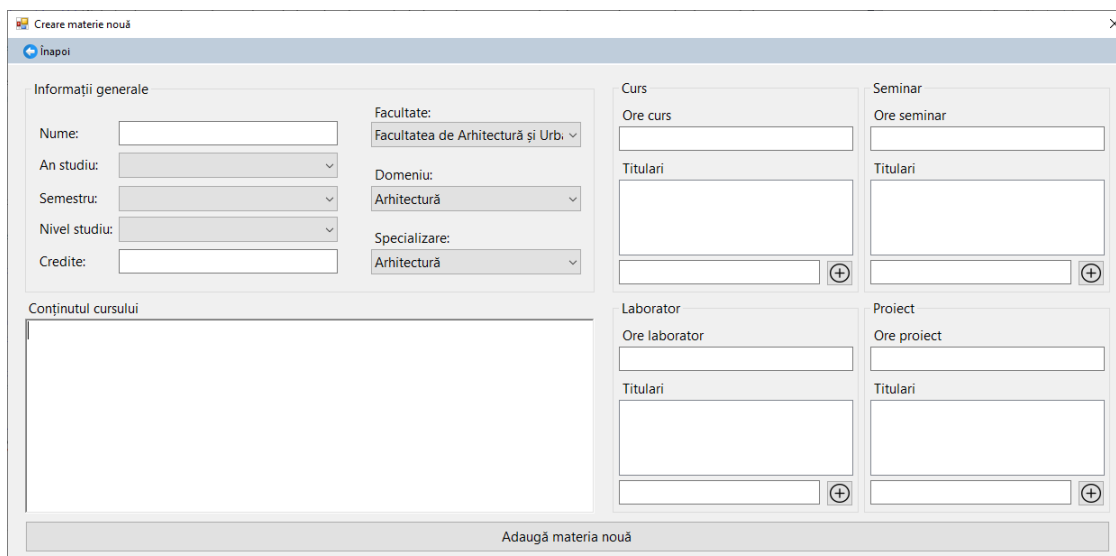
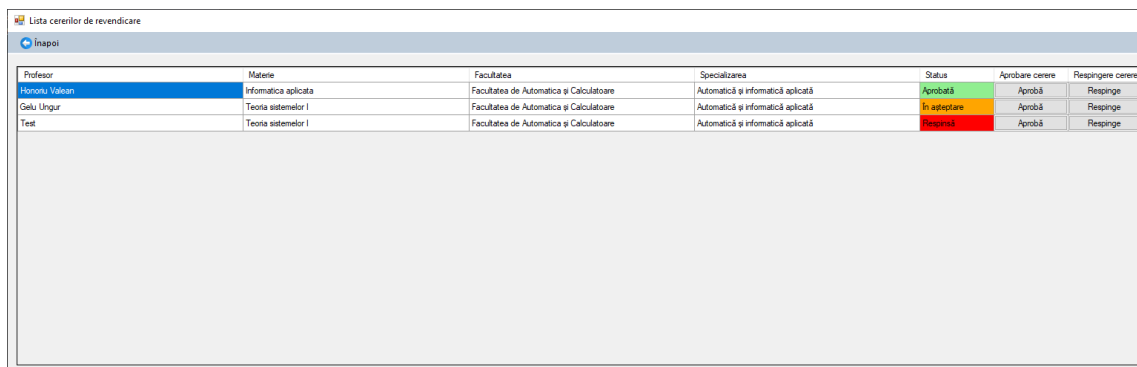


Figura 12: Forma pentru crearea unei materii

Funcționalitatea este identică cu cea pentru editarea unei materii, prezentată anterior. Salvarea modificărilor se face cu metoda din *web service* numită **Add-Course(course)**.

3.5 Cereri de revendicare

Această formă este vizibilă utilizatorilor care au rolurile de **Administrator**, **Secretar** sau **Profesor**. De aici, aceștia (în afară de Profesori) pot administra cererile de revendicare a materiilor care au fost realizate în cadrul aplicației.



Profesor	Materie	Facultate	Specializarea	Status	Aprobare cerere	Respingere cerere
Nikola Vitan	Informatica aplicata	Facultatea de Automatica și Calculatoare	Automatică și informatică aplicată	Aprobată	Aprobă	Respinge
Gelu Ungur	Teoria sistemelor I	Facultatea de Automatica și Calculatoare	Automatică și informatică aplicată	In asteptare	Aprobă	Respinge
Test	Teoria sistemelor I	Facultatea de Automatica și Calculatoare	Automatică și informatică aplicată	Respinsă	Aprobă	Respinge

Figura 13: Forma pentru *administrarea cererilor de revendicare*

Prin intermediul celor 2 butoane, **Aprobă** și **Respingere**, utilizatorii pot să ofere sau să respingă accesul profesorilor la editarea informațiilor pentru un anumit curs.

Metode din *web service* folosite:

1. **GetRequests()**: returnează o listă cu toate cererile de revendicare;
2. **SetCourseRedemptionStatus(courseId, status)**: modifică status-ul unei cereri de revendicare pentru un anumite curs (În așteptare, Aprobată sau Respinsă);

3.6 Facultăți

3.6.1 Lista facultăților

Această formă este afișată la apăsarea butonului *Vizualizare facultăți* din Dashboard. De aici, utilizatorii pot accesa informații despre toate facultățile introduse în baza de date a aplicației, dând click pe butonul corespunzător facultății dorite.

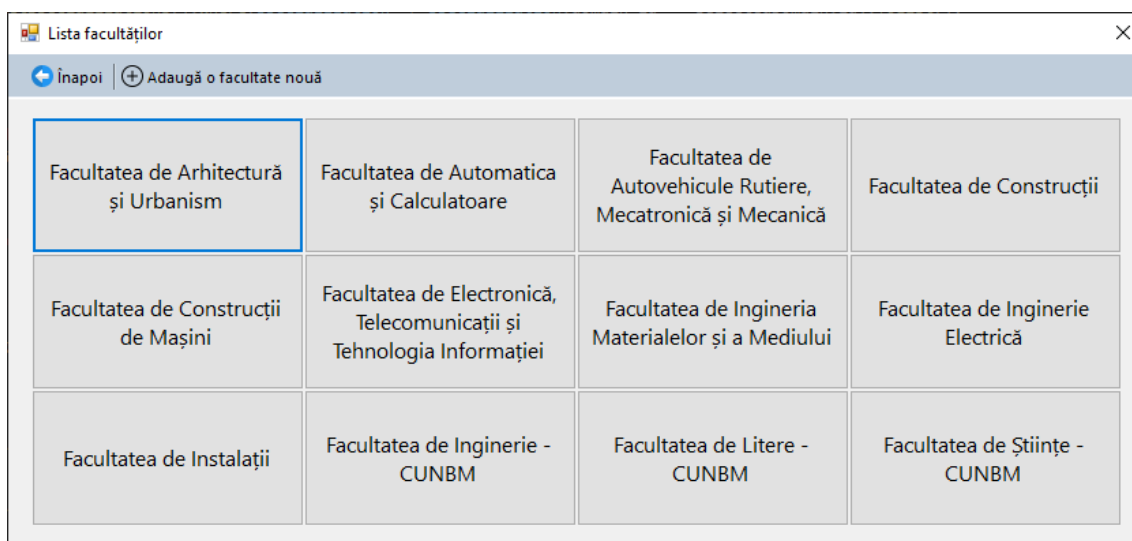


Figura 14: Forma pentru *vizualizarea facultăților*

Butoanele sunt generate în mod dinamic în funcție de datele primite din *web service*. Fiind plasate într-un control de tip **panel**, în momentul în care sunt prea multe butoane pe verticală, va apărea un scrollbar care permite navigarea pe verticală a listei cu facultăți.

Acțiuni disponibile:

1. **Înapoi**: închide forma curentă și o redeschide pe cea precedentă;
2. **Adaugă o facultate nouă** (doar pentru utilizatorii cu rolurile de **Administrator** și **Secretar**): deschide o formă folosită la introducerea unei noi facultăți în baza de date;

Metode din *web service* folosite:

1. **GetFaculties()**: returnează o listă cu toate facultățile din baza de date;

3.6.2 Vizualizare facultate

În cadrul acestei forme, orice utilizator poate vedea informațiile cele mai importante legate de o anumită facultate, precum și domeniile aferente ei.

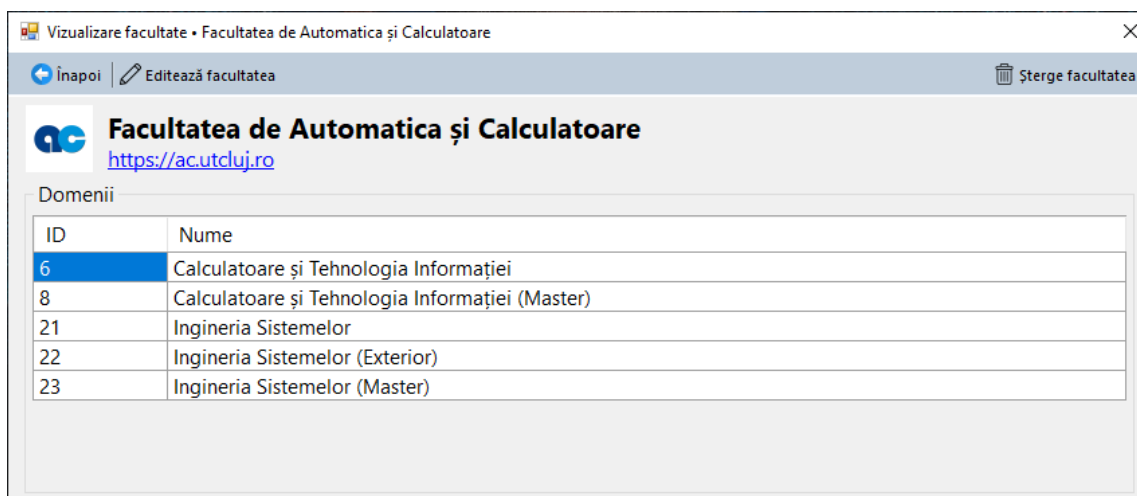


Figura 15: Forma pentru vizualizarea unei facultăți

Acțiuni disponibile:

1. **Înapoi**: închide forma curentă și o redeschide pe cea precedentă;
2. **Editează facultatea** (doar pentru utilizatorii cu rolurile de **Administrator** și **Secretar**): deschide o formă folosită la editarea informațiilor pentru o anumite facultate;
3. **Ștergere** (doar pentru utilizatorii cu rolurile de **Administrator** și **Secretar**): șterge facultatea respectivă;
 - Înainte de ștergerea propriu-zisă, utilizatorul primește un mesaj de confirmare și este înștiințat de efectele acțiunii sale:

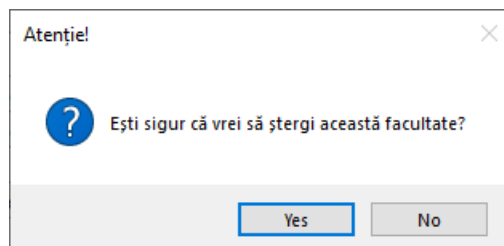


Figura 16: Mesaj de atenționare pentru ștergere

În plus, se poate da click pe link-ul facultății iar pagina respectivă se va deschide în browser-ul implicit al sistemului de operare.

Metode din *web service* folosite:

1. **GetFaculty(id)**: returnează facultatea cu *id*-ul specificat;
2. **GetDomains()**: returnează o listă cu toate domeniile;
3. **DeleteFaculty(id)**: șterge facultatea cu *id*-ul specificat din baza de date;

3.6.3 Editare facultate

În cadrul acestei forme, utilizatorii cu rolurile de **Administrator** sau **Secretar** pot edita informațiile unei anumite facultăți.

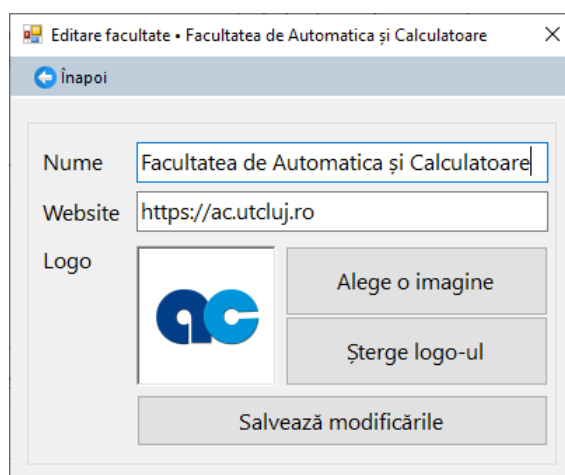
The image shows a web application window titled 'Editare facultate • Facultatea de Automatica și Calculatoare'. It features a 'Înapoi' (Back) button at the top left. The form contains three input fields: 'Nume' (Name) with the value 'Facultatea de Automatica și Calculatoare', 'Website' with the value 'https://ac.utcluj.ro', and 'Logo'. The 'Logo' section includes a preview of a blue circular logo with white letters 'ac', and two buttons: 'Alege o imagine' (Choose an image) and 'Șterge logo-ul' (Delete logo). At the bottom of the form is a large button labeled 'Salvează modificările' (Save changes).

Figura 17: Forma pentru editarea unei facultăți

Modificările se fac prin intermediul unor controale de tip **textBox**, iar încărcarea logo-ului se face prin intermediul unui **openFileDialog**, afișarea acestuia făcându-se într-un control de tip **pictureBox**. Corectitudinea adresei web se face cu ajutorul funcției din clasa **Utils: IsValidURL**.

O utilizare mai aparte este prezentă la încărcarea logo-ului: după ce butonul **Alege o imagine** este apăsat, se deschide un control de tip **openFileDialog** ce permite alegerea unui fișier de tip imagine (fișierele sunt filtrate astfel încât să fie imposibilă încărcarea unui alt tip de fișier care nu este o imagine). În momentul în care butonul **Salvează modificările** este acționat, imaginea este încărcată în folder-ul *Logos*, numele fișierului fiind stocat în baza de date.

Salvarea modificărilor se face cu metoda din *web service* **EditFaculty(faculty)**.

3.6.4 Creare facultate

În cadrul acestei forme, utilizatorii cu rolurile de **Administrator** sau **Secretar** pot adăuga o nouă facultate.

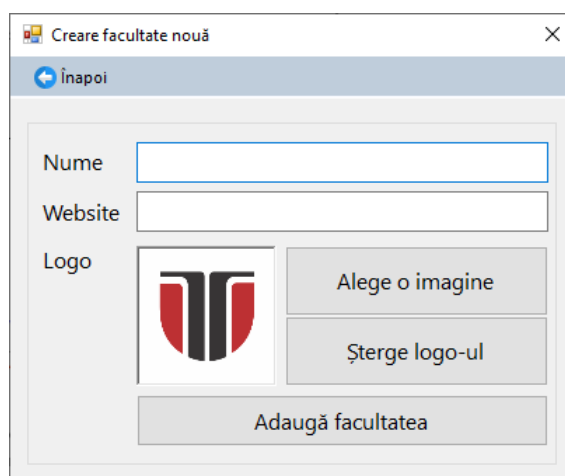


Figura 18: Forma pentru crearea unei facultăți

Atât în cazul creării, cât și în cazul editării unei facultăți, dacă nu se încarcă un logo, se va folosi imaginea implicit setată din setările aplicației (în cazul acesta, sigla UTCN).

Funcționalitatea este identică cu cea pentru editarea unei facultăți, prezentată anterior. Salvarea modificărilor se face cu metoda din *web service* numită **AddFaculty(faculty)**.

3.7 Domenii

3.7.1 Lista domeniilor

Această formă este afișată la apăsarea butonului *Vizualizare domenii* din Dashboard. De aici, utilizatorii pot accesa informații despre toate domeniile introduse în baza de date a aplicației, dând click pe butonul corespunzător domeniilor dorite.

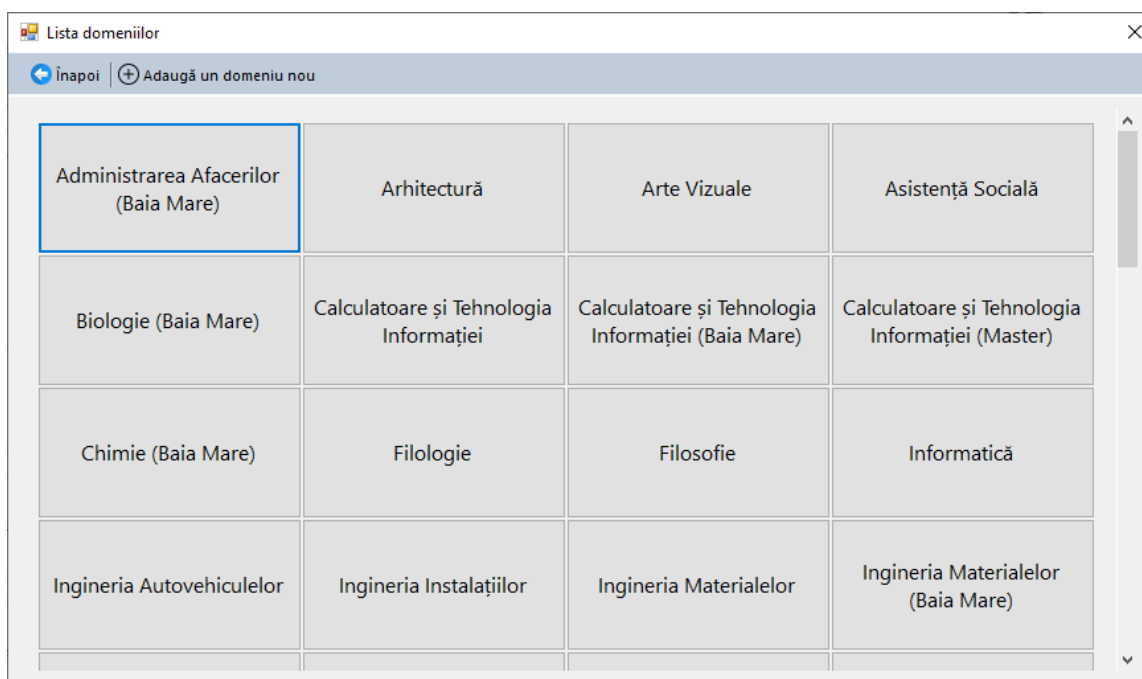


Figura 19: Forma pentru *vizualizarea domeniilor*

Butoanele sunt generate în mod dinamic în funcție de datele primite din *web service*. Fiind plasate într-un control de tip **panel**, în momentul în care sunt prea multe butoane pe verticală, va apărea un scrollbar care permite navigarea pe verticală a listei cu domenii.

Acțiuni disponibile:

1. **Înapoi**: închide forma curentă și o redeschide pe cea precedentă;
2. **Adaugă un domeniu nou** (doar pentru utilizatorii cu rolurile de **Administrator** și **Secretar**): deschide o formă folosită la introducerea unui nou domeniu în baza de date;

Metode din *web service* folosite:

1. **GetDomains ()**: returnează o listă cu toate domeniile din baza de date;

3.7.2 Vizualizare domeniu

În cadrul acestei forme, orice utilizator poate vedea informațiile cele mai importante legate de un anumit domeniu, precum și specializările aferente lui.

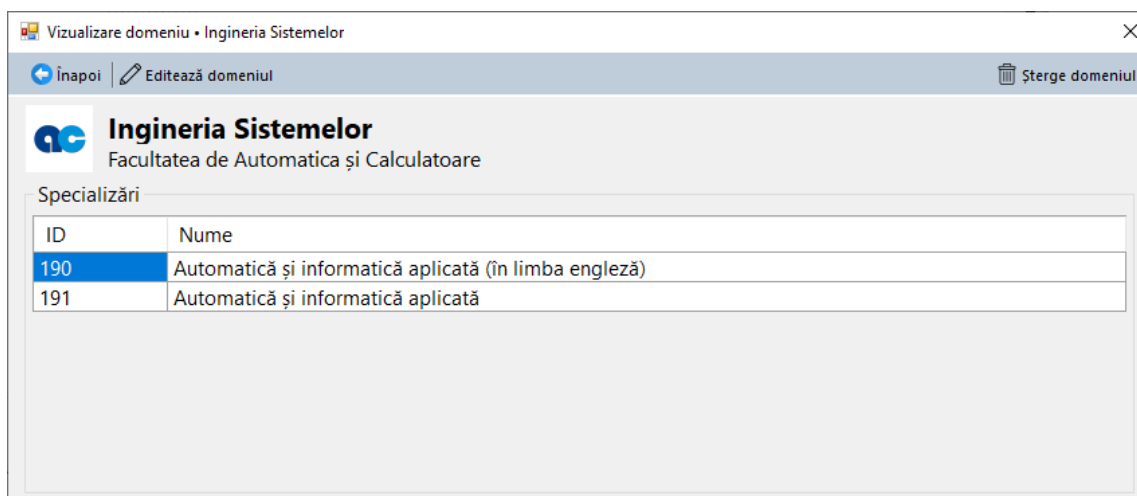


Figura 20: Forma pentru vizualizarea unui domeniu

Acțiuni disponibile:

1. **Înapoi**: închide forma curentă și o redeschide pe cea precedentă;
2. **Editează domeniul** (doar pentru utilizatorii cu rolurile de **Administrator** și **Secretar**): deschide o formă folosită la editarea informațiilor pentru un anumit domeniu;
3. **Ștergere** (doar pentru utilizatorii cu rolurile de **Administrator** și **Secretar**): șterge domeniul respectiv;
 - Înainte de ștergerea propriu-zisă, utilizatorul primește un mesaj de confirmare și este înștiințat de efectele acțiunii sale:

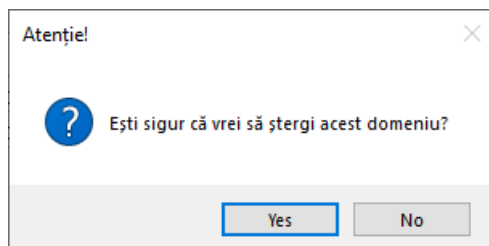


Figura 21: Mesaj de atenționare pentru ștergere

Metode din *web service* folosite:

1. **GetDomain(id)**: returnează domeniul cu *id*-ul specificat;
2. **GetSpecializations()**: returnează o listă cu toate specializările;
3. **DeleteDomain(id)**: șterge domeniul cu *id*-ul specificat;

3.7.3 Editare domeniu

În cadrul acestei forme, utilizatorii cu rolurile de **Administrator** sau **Secretar** pot edita informațiile unui anumit domeniu.

Figura 22: Forma pentru editarea unui domeniu

Modificările se fac prin intermediul unor controale de tip **textBox** și **comboBox**. Salvarea modificărilor se face cu metoda din *web service* **EditDomain(domain)**.

3.7.4 Creare domeniu

În cadrul acestei forme, utilizatorii cu rolurile de **Administrator** sau **Secretar** pot adăuga un domeniu nou.

Figura 23: Forma pentru crearea unui domeniu

Funcționalitatea este identică cu cea pentru editarea unui domeniu, prezentată anterior. Salvarea modificărilor se face cu metoda din *web service* numită **AddDomain(domain)**.

3.8 Specializări

3.8.1 Lista specializărilor

Această formă este afișată la apăsarea butonului *Vizualizare specializări* din Dashboard. De aici, utilizatorii pot accesa informații despre toate specializările introduse în baza de date a aplicației, dând click pe butonul corespunzător.

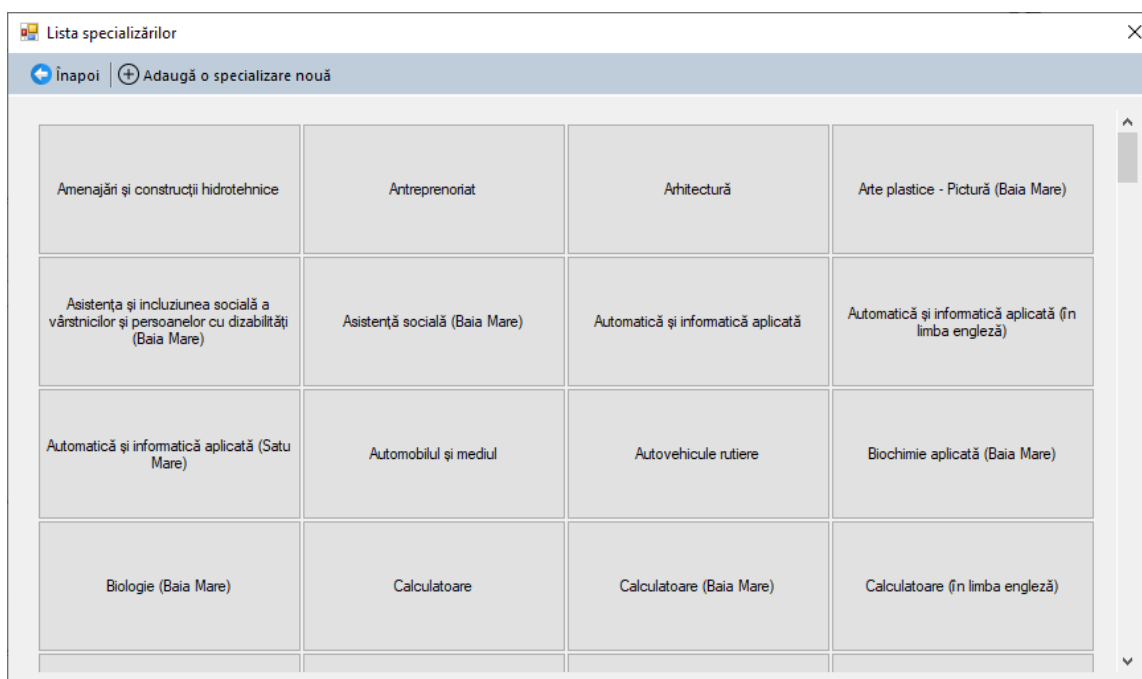


Figura 24: Forma pentru vizualizarea specializărilor

Butoanele sunt generate în mod dinamic în funcție de datele primite din *web service*. Fiind plasate într-un control de tip **panel**, în momentul în care sunt prea multe butoane pe verticală, va apărea un scrollbar care permite navigarea pe verticală a listei cu specializări.

Acțiuni disponibile:

1. **Înapoi**: închide forma curentă și o redeschide pe cea precedentă;
2. **Adaugă o specializare nouă** (doar pentru utilizatorii cu rolurile de **Administrator** și **Secretar**): deschide o formă folosită la introducerea unei noi specializări în baza de date;

Metode din *web service* folosite:

1. **GetSpecializations()**: returnează o listă cu toate specializările;

3.8.2 Vizualizare specializare

În cadrul acestei forme, orice utilizator poate vedea informațiile cele mai importante legate de o anumită specializare, precum și cursurile aferente lui.

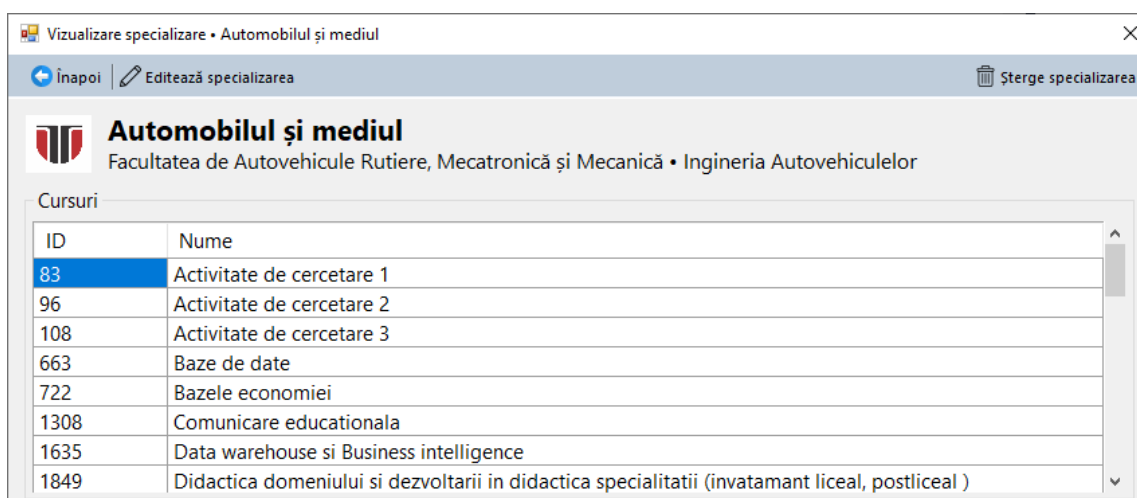


Figura 25: Forma pentru vizualizarea unei specializări

Acțiuni disponibile:

1. **Înapoi:** închide forma curentă și o redeschide pe cea precedentă;
2. **Editează specializarea** (doar pentru utilizatorii cu rolurile de **Administrator** și **Secretar**): deschide o formă folosită la editarea informațiilor pentru o anumită specializare;
3. **Ștergere** (doar pentru utilizatorii cu rolurile de **Administrator** și **Secretar**): șterge specializarea respectivă;
 - Înainte de ștergerea propriu-zisă, utilizatorul primește un mesaj de confirmare și este înștiințat de efectele acțiunii sale:

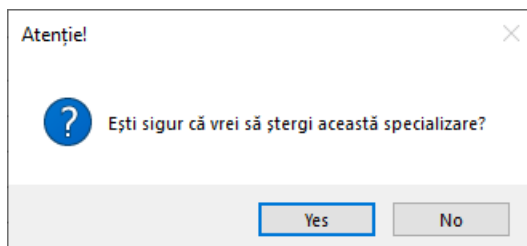


Figura 26: Mesaj de atenționare pentru ștergere

Metode din *web service* folosite:

1. **GetSpecialization(id)**: returnează domeniul cu *id*-ul specificat;
2. **GetCourses()**: returnează o listă cu toate specializările;
3. **DeleteSpecialization(id)**: șterge specializarea cu *id*-ul specificat;

3.8.3 Editare specializare

În cadrul acestei forme, utilizatorii cu rolurile de **Administrator** sau **Secretar** pot edita informațiile unei anumite specializări.

Figura 27: Forma pentru editarea unei specializări

Modificările se fac prin intermediul unor controale de tip **textBox** și **comboBox**.

Salvarea modificărilor se face cu metoda **EditSpecialization(specialization)**.

3.8.4 Creare domeniu

În cadrul acestei forme, utilizatorii cu rolurile de **Administrator** sau **Secretar** pot adăuga o nouă specializare.

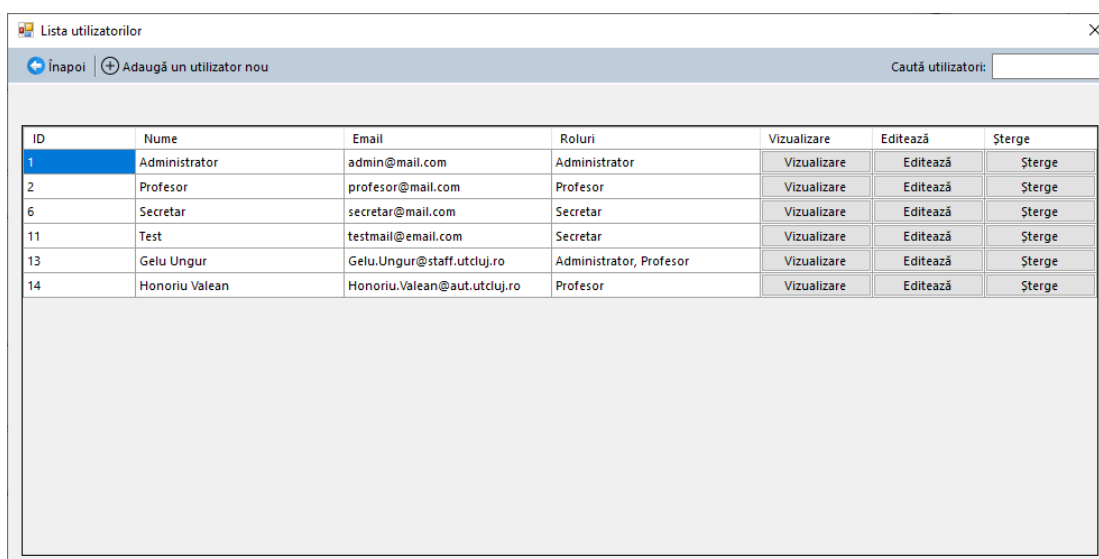
Figura 28: Forma pentru crearea unei specializări

Funcționalitatea este identică cu cea pentru editare. Salvarea modificărilor se face cu metoda din *web service* numită **AddSpecialization(specialization)**.

3.9 Utilizatori

3.9.1 Lista utilizatorilor

Această formă este afișată la apăsarea butonului *Vizualizare utilizatori* din Dashboard și este o secțiune accesibilă doar utilizatorilor care au rolul de **Administrator**. De asemenea, se pot căuta anumiți utilizatori în funcție de un cuvânt sau o frază cu ajutorul `textBox`-ului din colțul din dreapta sus, cu aceleași mențiuni în legătură cu performanța menționate anterior în capitolul **Materii**.



ID	Nume	Email	Roluri	Vizualizare	Editează	Șterge
1	Administrator	admin@mail.com	Administrator	Vizualizare	Editează	Șterge
2	Profesor	profesor@mail.com	Profesor	Vizualizare	Editează	Șterge
6	Secretar	secretar@mail.com	Secretar	Vizualizare	Editează	Șterge
11	Test	testmail@email.com	Secretar	Vizualizare	Editează	Șterge
13	Gelu Ungur	Gelu.Ungur@staff.utcluj.ro	Administrator, Profesor	Vizualizare	Editează	Șterge
14	Honoriu Valean	Honoriu.Valean@aut.utcluj.ro	Profesor	Vizualizare	Editează	Șterge

Figura 29: Forma pentru vizualizarea utilizatorilor

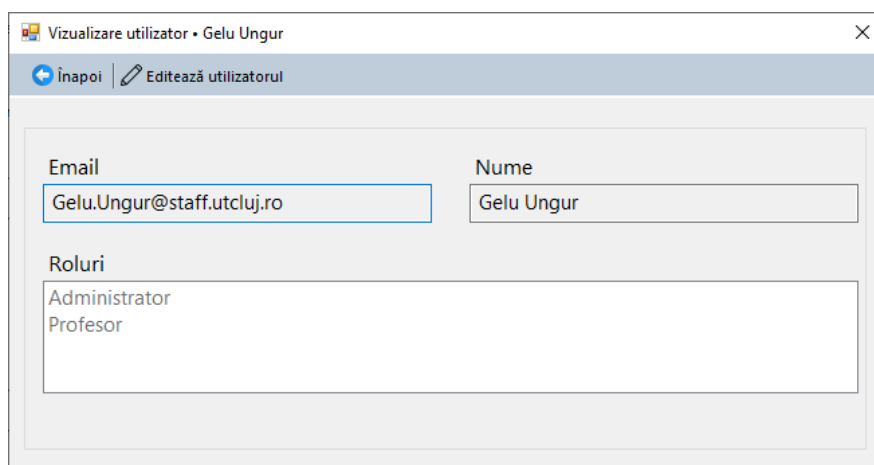
Acțiuni disponibile:

1. **Înapoi:** închide forma curentă și o redeschide pe cea precedentă;
2. **Adaugă un utilizator nou:** deschide o formă folosită la introducerea unui nou utilizator în baza de date;
3. **Căutare:** permite căutarea în baza de date după numele utilizatorului;
4. **Vizualizare:** deschide o formă care conține informațiile utilizatorului respectiv (în mod *read-only*);
5. **Editare:** deschide o formă care permite editarea informațiilor utilizatorului;
6. **Ștergere:** șterge utilizatorul respectiv;
 - Înainte de ștergerea propriu-zisă, utilizatorul primește un mesaj de confirmare și este înștiințat de efectele acțiunii sale.

Metode din *web service* folosite:

1. **GetUsers ()**: returnează o listă cu toți utilizatorii;
2. **GetUserRoles (id)**: returnează o listă cu toate rolurile unui anumit utilizator;
3. **GetUsersByName (searchTerm)**: returnează o listă cu toți utilizatorii care au în nume termenul căutării;

3.9.2 Vizualizare utilizator



The screenshot shows a web application window titled 'Vizualizare utilizator • Gelu Ungur'. Below the title bar, there are two buttons: 'Înapoi' (Back) and 'Editează utilizatorul' (Edit user). The main content area is divided into three sections: 'Email' with the value 'Gelu.Ungur@staff.utcluj.ro', 'Nume' (Name) with the value 'Gelu Ungur', and 'Roluri' (Roles) which is a list containing 'Administrator' and 'Profesor'.

Figura 30: Forma pentru vizualizarea unui utilizator

Acțiuni disponibile:

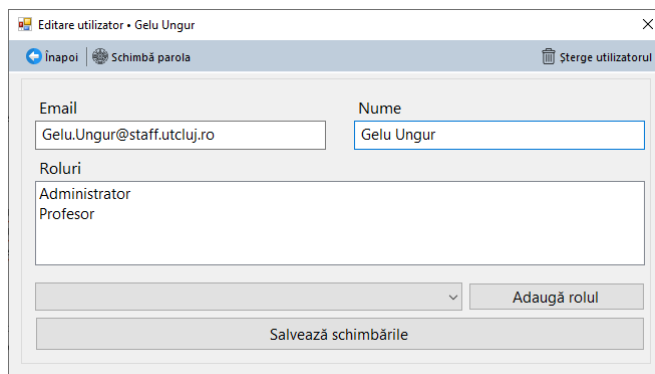
1. **Înapoi**: închide forma curentă și o redeschide pe cea precedentă;
2. **Editează utilizatorul**: deschide o formă folosită la editarea informațiilor pentru un anumit utilizator;

Metode din *web service* folosite:

1. **GetUser (id)**: returnează utilizatorul cu *id*-ul specificat;
2. **GetUserRoles (id)**: returnează o listă cu toate rolurile unui anumit utilizator;

3.9.3 Editare utilizator & parolă

Aceste forme sunt disponibile pentru toți utilizatorii, cu mici excepții în ceea ce privește editarea rolurilor (doar utilizatorii cu rolul de **Administrator** le pot modifica — pentru ceilalți utilizatori secțiunea este *read-only*).



The screenshot shows a web application window titled 'Editare utilizator • Gelu Ungur'. It has a navigation bar with buttons for 'Înapoi' (Back), 'Schimbă parola' (Change password), and 'Șterge utilizatorul' (Delete user). The main form contains two text input fields: 'Email' with the value 'Gelu.Ungur@staff.utcluj.ro' and 'Nume' (Name) with the value 'Gelu Ungur'. Below these is a 'Roluri' (Roles) section with a list box containing 'Administrator' and 'Profesor'. At the bottom of the roles section is a dropdown menu and a button labeled 'Adaugă rolul' (Add role). At the very bottom of the form is a button labeled 'Salvează schimbările' (Save changes).

Figura 31: Forma pentru editarea unui utilizator

Modificările se fac prin intermediul unor controale de tip **textBox**, **comboBox** și **listBox**. O utilizare mai aparte este prezentă la editarea rolurilor, care sunt alese din comboBox-ul din partea de jos a formei și adăugate în listBox-ul corespunzător, cele selectate fiind șterse din comboBox. Adăugarea în listBox se face prin apăsarea butonului **Adaugă rolul**. Pentru ștergerea unui anumit *rol* din listă, se dă click pe el și se apasă tasta **backspace**, acesta fiind adăugat din nou în comboBox.

Acțiuni disponibile:

1. **Înapoi**: închide forma curentă și o redeschide pe cea precedentă;
2. **Schimbă parola**: deschide o formă folosită la schimbarea parolei pentru un anumit utilizator;
3. **Ștergere utilizator**: șterge utilizatorul respectiv;

Dând click pe butonul **Schimbă parola**, se deschide o formă nouă unde este posibilă schimbarea parolei pentru utilizatorul respectiv.

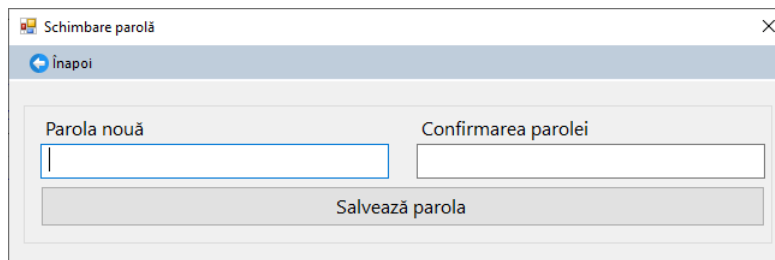


Figura 32: Forma pentru editarea parolei unui utilizator

Metode din *web service* folosite:

1. **GetUser(id)**: returnează utilizatorul cu *id*-ul specificat;
2. **GetUserRoles(id)**: returnează o listă cu toate rolurile unui anumit utilizator;
3. **DeleteUser(id)**: șterge utilizatorul cu *id*-ul specificat;
4. **EditUser(user)**: salvează modificările unui utilizator în baza de date;
5. **UpdatePassword(id, newPassword)**: editează parola unui utilizator în baza de date;

3.9.4 Creare utilizator

În cadrul acestei forme, utilizatorii cu rolurile de **Administrator** pot adăuga un nou utilizator.

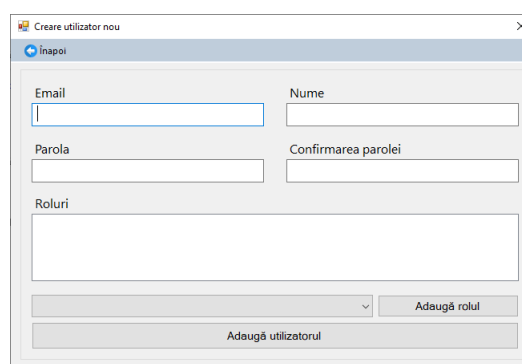


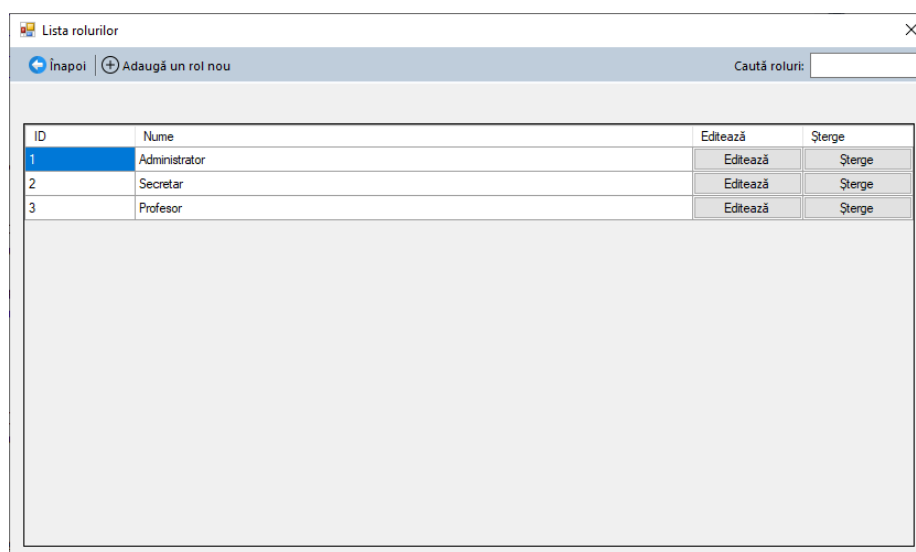
Figura 33: Forma pentru crearea unui utilizator

Funcționalitatea este identică cu cea pentru editarea unui utilizator, prezentată anterior. Salvarea modificărilor se face cu metoda din *web service* numită **AddUser(user)**.

3.10 Roluri

3.10.1 Lista rolurilor

Această formă este afișată la apăsarea butonului *Vizualizare roluri* din Dashboard și este o secțiune accesibilă doar utilizatorilor care au rolul de **Administrator**. De asemenea, se pot căuta anumite roluri în funcție de un cuvânt sau o frază cu ajutorul `textBox`-ului din colțul din dreapta sus, cu aceleași mențiuni în legătură cu performanța menționate anterior în capitolul **Materii**.



ID	Nume	Editează	Șterge
1	Administrator	Editează	Șterge
2	Secretar	Editează	Șterge
3	Profesor	Editează	Șterge

Figura 34: Forma pentru *vizualizarea rolurilor*

Acțiuni disponibile:

1. **Înapoi:** închide forma curentă și o redeschide pe cea precedentă;
2. **Adaugă un rol nou:** deschide o formă folosită la introducerea unui nou rol în baza de date;
3. **Căutare:** permite căutarea în baza de date după numele rolului;
4. **Editare:** deschide o formă care permite editarea informațiilor rolului;
5. **Ștergere:** șterge rolul respectiv;
 - Înainte de ștergerea propriu-zisă, utilizatorul primește un mesaj de confirmare și este înștiințat de efectele acțiunii sale.
 - Nu se poate șterge un rol dacă este atașat la cel puțin un utilizator. În acest caz, utilizatorul este înștiințat printr-un `messageBox`.

Metode din *web service* folosite:

1. **GetRoles()**: returnează o listă cu toți utilizatorii;
2. **GetRolesByName(searchTerm)**: returnează o listă cu toate rolurile care au în nume termenul căutării;

3.10.2 Creare & editare rol

Aceste forme sunt disponibile doar pentru utilizatorii care au rolul de **Administrator** și permit adăugarea, respectiv editarea unui anumit rol.

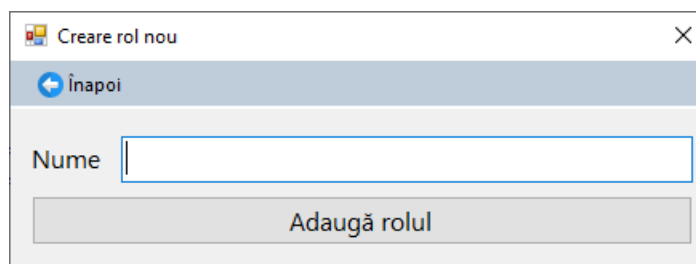
A screenshot of a Windows-style dialog box titled "Creare rol nou" (Create new role). It has a close button (X) in the top right corner. Below the title bar is a blue header bar with a back arrow icon and the text "Înapoi" (Back). The main area contains a text input field labeled "Nume" (Name) which is currently empty. Below the input field is a large grey button labeled "Adaugă rolul" (Add role).

Figura 35: Forma pentru adăugarea unui utilizator

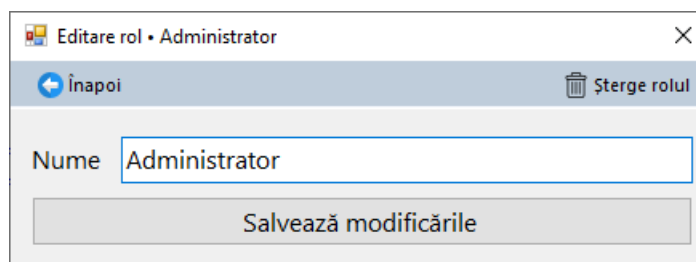
A screenshot of a Windows-style dialog box titled "Editare rol • Administrator" (Edit role • Administrator). It has a close button (X) in the top right corner. Below the title bar is a blue header bar with a back arrow icon and the text "Înapoi" (Back), and a trash can icon with the text "Șterge rolul" (Delete role). The main area contains a text input field labeled "Nume" (Name) which contains the text "Administrator". Below the input field is a large grey button labeled "Salvează modificările" (Save changes).

Figura 36: Forma pentru editarea unui rol

Modificările se fac prin intermediul unor controale de tip **textBox**, salvarea în baza de date făcându-se cu metodele din *web service* denumite **AddRole(role)**, respectiv **EditRole(role)**.

Acțiuni disponibile:

1. **Înapoi**: închide forma curentă și o redeschide pe cea precedentă;
2. **Șterge rolul**: șterge rolul respectiv;

4 Testarea aplicației

În urma testării aplicației, atât în timpul dezvoltării ei, cât și la final, am observat mici probleme la care am găsit soluții:

1. **Posibilitatea de trimitere spre baza de date a datelor nule:** în cazul unor anumite forme, se puteau salva anumite date chiar dacă controalele aferente câmpurilor lor erau goale. Această problemă ducea uneori la erori din SQL, sau pur și simplu unele câmpuri deveneau goale în baza de date.
 - **Soluția:** Am adăugat condiții în cod astfel încât *request*-ul către Server să se facă doar după ce toate câmpurile obligatorii au fost completate. În vederea atenționării utilizatorului, acesta va primi un MessageBox în acest sens:

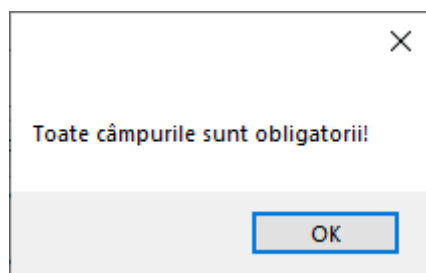


Figura 37: Mesaj pentru utilizator atunci când nu completează toate câmpurile

2. **Erori la constrângerile din baza de date:** la ștergerea unui obiect din baza de date, în cazul în care *id*-ul obiectului era referențiat într-o altă tabelă, serverul de SQL dădea o eroare în această privință (de exemplu, nu se putea ștergea un utilizator dacă acesta avea un curs sau un rol atașat).
 - **Soluția:** Am adăugat condiții în cod astfel încât să nu fie posibilă ștergerea unui anumit obiect dacă *id*-ul acestuia este referențiat într-o altă tabelă. În vederea atenționării utilizatorului, acesta va primi un MessageBox în acest sens:

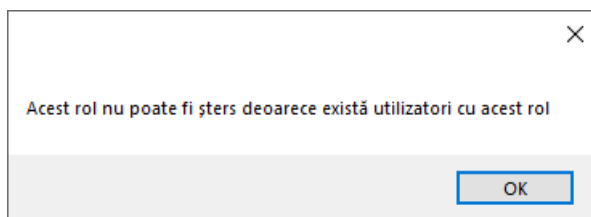


Figura 38: Mesaj pentru utilizator atunci când se violează o constrângere

5 Anexe (Codul sursă)