

Universitatea “Dunarea de Jos” Galati

Facultatea de Automatica, Calculatoare, Inginerie Electrica si Electronica

Departamentul de Calculatoare si Tehnologia Informatiei

**Gestiunea elevilor și a notelor într-o școală generală**

Student: Beatrice Bulugu

Profesor: Dan Munteanu

2025

**1. Tema**

Gestiunea studenților și a notelor într-o școală generală

Proiectul reprezintă o aplicație de gestionare a informațiilor școlare. Acesta permite gestionarea elevilor, profesorilor, materiilor și notelor în cadrul unei instituții de învățământ. Aplicația include funcționalități pentru administrare, raportare și vizualizarea datelor educaționale. Este orientată către trei tipuri principale de utilizatori: **administrator**, **profesor** și **elev**.

**2. Analiza problemei**

Descrierea realității care se modelează:

În cadrul unei școli generale, gestionarea informațiilor despre studenți, note și materii este esențială pentru a asigura buna funcționare a procesului educațional. De obicei, fiecare elev este înscris la mai multe materii predate de profesori diferiți, iar performanța lui școlară este monitorizată prin note și alte evaluări pe parcursul semestrelor.

Entitățile principale care apar în acest sistem sunt:

* Elevii: fiecare elev este o persoană care frecventează cursurile școlii și este evaluată periodic în funcție de performanțele academice. În realitate, elevii sunt responsabili de participarea la cursuri, de învățarea materiei și de susținerea evaluărilor (teste, examene, proiecte). Atributele asociate unui elev pot include:
* Nume
* Prenume
* Data nașterii
* Clasa (de exemplu, clasa a V-a, clasa a VIII-a)
* Adresă
* Număr de telefon de contact
* CNP (cod numeric personal)
* Părinți/tutori legali
* Profesorii: sunt persoanele care predau materia și evaluează elevii. Ei sunt responsabili de organizarea lecțiilor, de corectarea lucrărilor și de notarea elevilor. Profesorii sunt repartizați pe diferite materii și clase. Atributele asociate unui profesor pot include:
* Nume
* Prenume
* Materia predată
* Grad didactic (de exemplu, debutant, titular, profesor gradul I)
* Adresa
* Telefon de contact
* Materiile: acestea reprezintă domenii de studiu predate în cadrul școlii. Fiecare materie este predată de un profesor și evaluată periodic. Exemple de materii sunt: Matematică, Limba Română, Geografie, Biologie etc. In cadrul unei scoli, un profesor poate avea dubla-specializare, deci poate preda mai multe materii, iar o materie poate fi predata de mai multi profesori(mai ales in scolile cu mai multi elevi). Atributele pentru o materie includ:
* Denumirea materiei
* Clasa căreia i se adresează (de exemplu, clasele V-VIII)
* Profesorul care predă material
* Numarul de ore pe saptamana
* Notele: sunt evaluările pe care elevii le primesc pentru performanțele lor la diferitele materii. Ele pot reflecta rezultatele obținute la teste, lucrări, examene, proiecte sau alte forme de evaluare. Notele sunt asociate unei materii și unui elev specific. Atributele unei note pot include:
* Valoarea notei (de la 1 la 10, în cazul sistemului de notare din România)
* Data evaluării
* Materia la care s-a primit nota
* Elevul care a fost evaluat
* Tipul evaluării (test, examen, proiect)
* Clasele: fiecare școală este organizată pe clase, iar elevii sunt distribuiți în funcție de vârstă și nivel de studii (de exemplu, clasa a V-a, clasa a VI-a etc.). Clasele sunt formate din grupuri de elevi care participă la aceleași cursuri. Atributele asociate unei clase includ:
* Numele clasei (de exemplu, a V-a A, a VI-a B)
* Dirigintele clasei (profesorul responsabil de gestionarea clasei)
* Lista elevilor care fac parte din clasă

**Modul de interacțiune:**

Elevii, profesorii și administrația școlii interacționează în moduri diferite. Profesorii interacționează cu elevii în cadrul lecțiilor, îi evaluează prin diferite teste și înregistrează rezultatele sub formă de note. Elevii, la rândul lor, participă la ore și la evaluări. Administrația școlii urmărește progresul elevilor prin intermediul rapoartelor și gestionează datele despre profesori, elevi și materii.

Acțiunile care se desfășoară în realitate:

Elevii:

* Participă la orele de curs.
* Susțin evaluări scrise sau orale (teste, examene).
* Obțin note în urma evaluărilor.
* Interacționează cu profesorii în cadrul orelor de curs.

Profesorii:

* Predau materia conform programului școlar.
* Evaluează performanțele elevilor prin teste, examene, proiecte.
* Înregistrează notele obținute de elevi.
* Pregătesc rapoarte pentru administrație sau pentru părinți.
* Organizează ședințe cu părinții și consultații pentru elevi.

Administrația școlii:

* Monitorizează activitatea didactică a profesorilor.
* Administrează informațiile legate de elevi (date personale, prezență).
* Monitorizează notele și rezultatele școlare ale elevilor.
* Coordonează relațiile cu părinții și organizează evenimente școlare (ședințe cu părinții, festivități etc.).

**3. Specificarea cerințelor detaliată**

Aplicația va avea trei tipuri principale de utilizatori: **administrator**, **profesor** și **elev**. Fiecare rol va avea acces la funcționalități specifice, detaliate mai jos.

1. **Funcționalități pentru rolul Administrator**

Administratorul gestionează întreaga aplicație și are acces complet la date. Funcționalitățile pentru acest rol includ:

1. Gestionarea elevilor:
   * Adăugarea de elevi noi, completând informațiile: nume, prenume, CNP, clasă.
   * Ștergerea elevilor din baza de date, dacă aceștia nu au note asociate.
   * Actualizarea datelor despre elevi, cum ar fi schimbarea clasei.
2. Gestionarea profesorilor:
   * Adăugarea de profesori, completând informațiile personale (nume, prenume) și asociind materiile pe care le predau.
   * Ștergerea profesorilor, dacă aceștia nu au materii asociate.
   * Actualizarea datelor despre profesori.
3. Gestionarea materiilor:
   * Crearea unor materii noi, cu detalii despre profesori asociați.
   * Ștergerea materiilor, dacă acestea nu sunt legate de note.
   * Modificarea informațiilor despre materii.
4. Statistici și rapoarte:
   * Vizualizarea numărului total de elevi, grupat pe clase.
   * Generarea de grafice cu distribuția notelor pentru o materie sau clasă.
   * Exportarea datelor în format PDF sau XLS.
5. Vizualizarea datelor:
   * Accesarea listelor detaliate de elevi, profesori, clase și materii.

2. **Funcționalități pentru rolul Profesor**

Profesorii au un rol esențial în evaluarea elevilor. Aplicația va oferi următoarele funcționalități pentru acest rol:

1. Adăugarea notelor:
   * Profesorii pot acorda note elevilor la materiile pe care le predau.
   * Introducerea notei, cu validarea că nota se încadrează în intervalul 1-10.

3**. Funcționalități pentru rolul Elev**

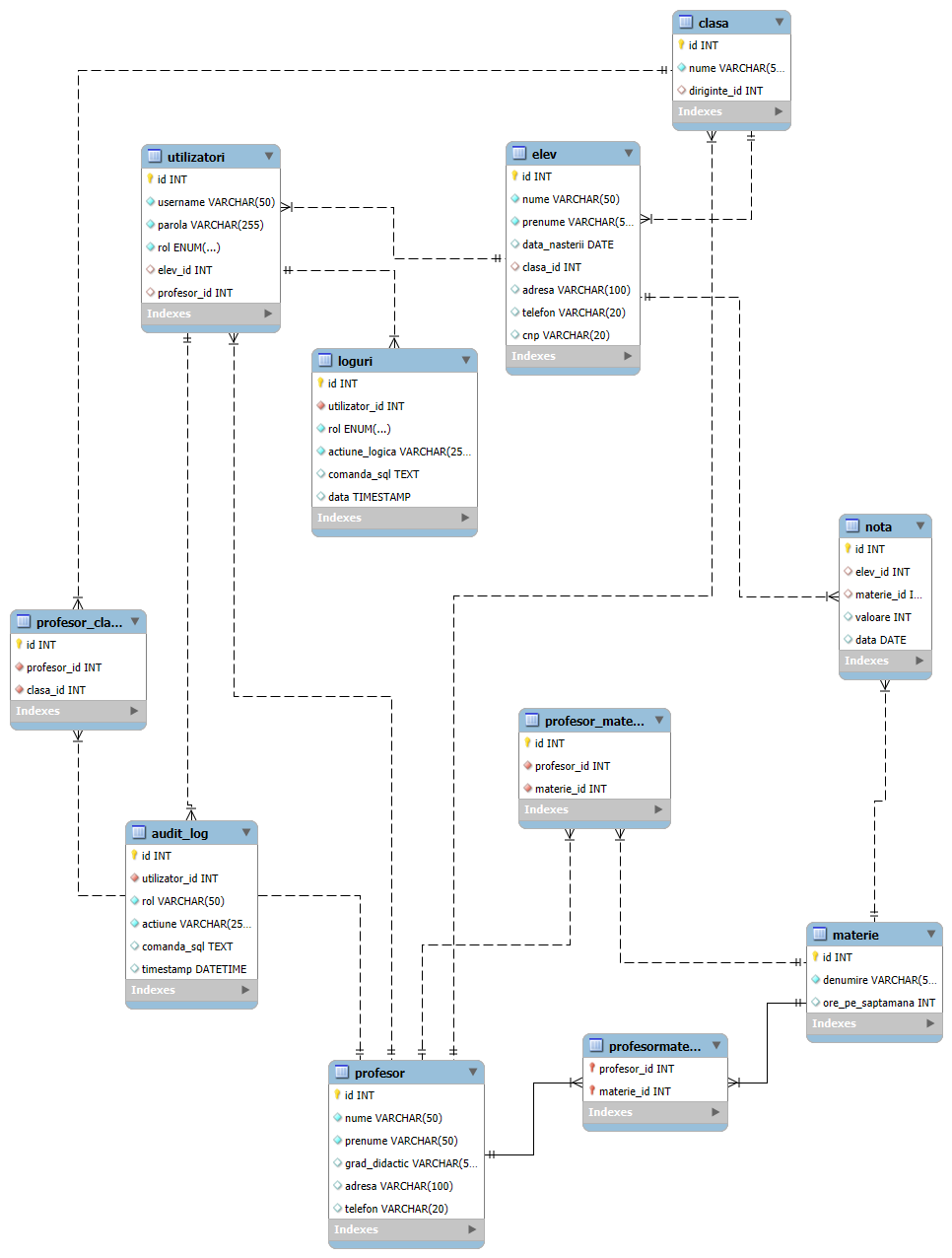
Elevii au acces limitat la aplicație, strict pentru a vizualiza informațiile personale. Funcționalitățile includ:

1. Vizualizarea notelor personale:
   * Elevii pot vedea toate notele obținute, grupate pe materii.

### **Alte detalii**

* **Autentificare și autorizare:**  
  Aplicația va avea un sistem de autentificare bazat pe nume de utilizator și parolă, care să permită accesul utilizatorilor la funcționalitățile specifice rolului lor.
* **Interfață utilizator:**  
  Designul aplicației va fi intuitiv, cu meniuri clare pentru fiecare funcționalitate, adaptate rolului utilizatorului.
* **Scalabilitate:**  
  Sistemul va permite adăugarea de noi clase, materii sau utilizatori fără a afecta performanța aplicației.

**4. Proiectarea la nivel conceptual (Diagrama entitate-relație)**



1. **Proiectarea la nivel logic**

**Tabela elev**

* Descriere: Reprezintă informațiile despre elevi.
* Atribute:
  + id: Cheie primară, identifică unic fiecare elev.
  + nume, prenume, cnp, data\_nasterii, adresa, telefon: Informații personale despre elev.
  + clasa\_id: Cheie externă către tabela clasa.
* Relații:
  + Fiecare elev este asociat unei clase (relație mulți-la-unu cu tabela clasa).
  + Legat de utilizatori prin câmpul elev\_id.

**Tabela clasa**

* Descriere: Reprezintă clasele școlare.
* Atribute:
  + id: Cheie primară, identifică unic fiecare clasă.
  + nume: Denumirea clasei (ex.: „10A”).
  + diriginte\_id: Cheie externă către tabela profesor (dirigintele clasei).
* Observații: Legătura cu tabela Profesori este de tip unu-la-unu (o clasă are un singur diriginte, iar un profesor poate fi diriginte la o singură clasă).

**Tabela utilizatori**

* Descriere: Reprezintă informațiile de autentificare și rolurile utilizatorilor sistemului.
* Atribute:
  + id: Cheie primară, identifică unic fiecare utilizator.
  + username, parola: Date pentru autentificare.
  + rol: Specifică tipul utilizatorului (e.g., elev, profesor, administrator).
  + elev\_id: Cheie externă către tabela elev (dacă utilizatorul este elev).
  + profesor\_id: Cheie externă către tabela profesor (dacă utilizatorul este profesor).
* Relații:
  + Legată de elev și profesor prin câmpurile elev\_id și profesor\_id.

**Tabela profesor**

* Descriere: Reprezintă informațiile despre profesori.
* Atribute:
  + id: Cheie primară, identifică unic fiecare profesor.
  + nume, prenume, grad\_didactic, adresa, telefon: Informații personale despre profesor.
* Relații:
  + Legată de utilizatori prin câmpul profesor\_id.
  + Legată de clasa prin tabela intermediară profesor\_clasa.
  + Legată de materie prin tabela intermediară profesor\_materie.

**Tabela materie**

* Descriere: Reprezintă materiile predate în școală.
* Atribute:
  + id: Cheie primară.
  + denumire: Numele materiei (ex.: „Matematică”).
  + ore\_pe\_saptamana: Numărul de ore săptămânale pentru această materie.
* Relații:
  + Legată de profesor prin tabela profesor\_materie.
  + Legată de nota prin câmpul materie\_id.

**Tabela nota**

* Descriere: Reprezintă notele primite de elevi.
* Atribute:
  + id: Cheie primară.
  + elev\_id: Cheie externă către tabela elev.
  + materie\_id: Cheie externă către tabela materie.
  + valoare: Nota efectivă primită de elev.
* Relații:
  + Legată de elev prin câmpul elev\_id.
  + Legată de materie prin câmpul materie\_id.

**Tabela loguri și audit\_log**

* Descriere: Reprezintă loguri ale acțiunilor utilizatorilor în sistem.
* Atribute principale:
  + utilizator\_id: Cheie externă către tabela utilizatori.
  + actiune, comanda\_sql: Înregistrează acțiunea efectuată și comanda SQL generată.
  + data, timestamp: Marca temporală pentru acțiuni.
* Relații:
  + Legată de tabela utilizatori pentru identificarea utilizatorului care a efectuat acțiunea.

### **Relațiile principale din baza de date**

1. **Elevi ↔ Clase ↔ Profesori**
   * Fiecare elev aparține unei clase (relație mulți-la-unu).
   * Fiecare clasă este gestionată de un diriginte (relație unu-la-unu cu tabela Profesori).
   * Profesorii pot preda la mai multe clase (prin tabela intermediară profesor\_clasa).
2. **Profesori ↔ Materii**
   * Un profesor poate preda mai multe materii (relație mulți-la-mulți prin tabela profesor\_materie). O materie poate fi predata de mai multi profesori
3. **Elevi ↔ Note ↔ Materii**
   * Fiecare notă este asociată unui elev și unei materii.
   * Relația este gestionată prin tabela nota.
4. **Utilizatori și integrarea lor cu alte tabele**
   * Utilizatorii pot fi elevi, profesori sau administrator/secretari legați prin câmpurile elev\_id, profesor\_id, admin\_id.
   * Orice acțiune efectuată de utilizatori este înregistrată în loguri și audit\_log.

Tabelele intermediare, precum **profesor\_clasa** și **profesor\_materie**, sunt utilizate pentru a gestiona relațiile **mulți-la-mulți** dintre entitățile din baza de date. Aceste relații nu pot fi reprezentate direct într-o bază de date relațională fără o tabelă intermediară.

### **Tabela** profesor\_clasa

#### **Rol:**

* Gestionează relația dintre **profesori** și **clase**.
* Într-o școală, un profesor poate preda la mai multe clase, iar o clasă poate avea mai mulți profesori care predau diferite materii. Această relație complexă este gestionată prin tabela intermediară profesor\_clasa.

#### **Structura:**

* id: Cheia primară a tabelei.
* profesor\_id: Cheie externă care referă un profesor din tabela profesor.
* clasa\_id: Cheie externă care referă o clasă din tabela clasa.

#### **Cum gestionează relația mulți-la-mulți:**

* Fiecare rând din profesor\_clasa indică o asociere între un profesor și o clasă.  
  De exemplu:
  + Profesorul cu profesor\_id = 1 predă la clasa cu clasa\_id = 10.
  + Profesorul cu profesor\_id = 2 predă la clasele cu clasa\_id = 10 și clasa\_id = 11.
* Dacă o clasă are mai mulți profesori, tabela va avea mai multe rânduri care leagă acea clasă de diferiți profesori.

#### **Exemplu de utilizare:**

Să presupunem că dorim să aflăm ce profesori predau la clasa "10A". Interogarea bazei de date va implica:

* Se caută în profesor\_clasa toate intrările unde clasa\_id corespunde clasei "10A".
* Se extrag valorile din câmpul profesor\_id, care se pot folosi pentru a identifica profesorii în tabela profesor.

### **Tabela** profesor\_materie

#### **Rol:**

* Gestionează relația dintre **profesori** și **materii**.
* Într-o școală, un profesor poate preda mai multe materii (ex.: Matematică și Informatică), iar o materie poate fi predată de mai mulți profesori (ex.: diferiți profesori predau Matematică în clase diferite). Această relație este gestionată prin tabela intermediară profesor\_materie.

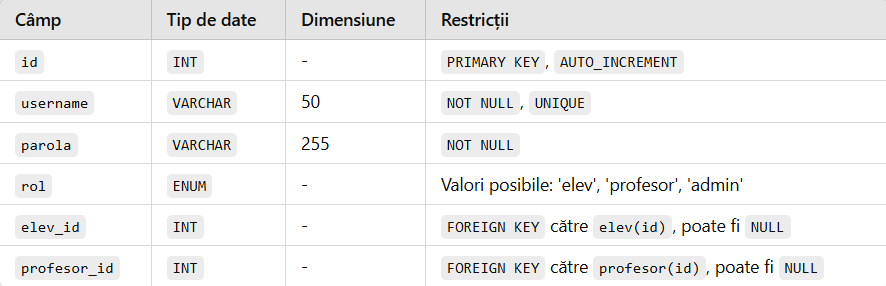
#### **Structura:**

* id: Cheia primară a tabelei.
* profesor\_id: Cheie externă care referă un profesor din tabela profesor.
* materie\_id: Cheie externă care referă o materie din tabela materie.

1. **Proiectarea la nivel fizic**

#### ****Tabela**** utilizatori

* **Scop:** Gestionează informațiile de autentificare și rolurile utilizatorilor.
* **Structură fizică:**

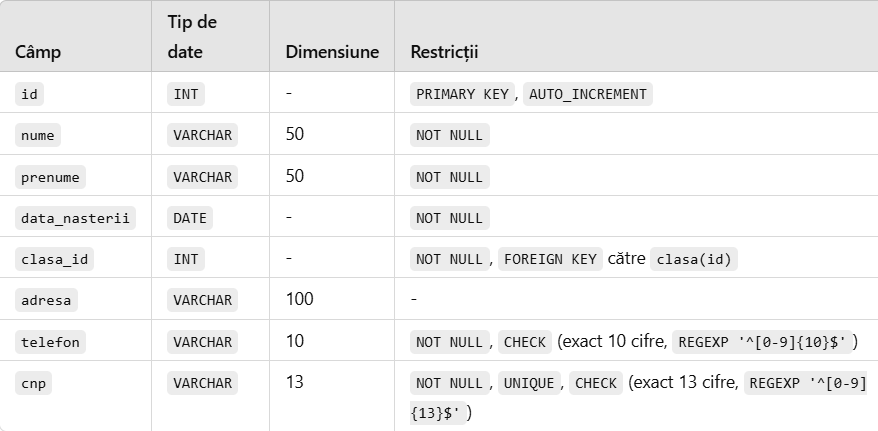


Observații:

* username este unic pentru a preveni conturi duplicate.
* parola are o dimensiune mare pentru a stoca parole criptate (hash-uri).
* rol definește tipul utilizatorului și permite validarea valorilor posibile.

**Tabela** elev

* Scop: Stochează datele personale ale elevilor.
* Structură fizică:

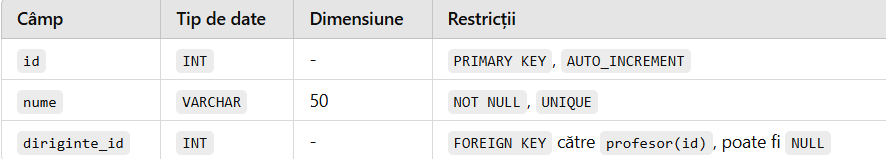


Observații:

* cnp este unic pentru a identifica elevii în mod distinct.
* clasa\_id leagă fiecare elev de o clasă.

#### ****Tabela**** clasa

* **Scop:** Reprezintă informații despre clasele școlare.
* **Structură fizică:**

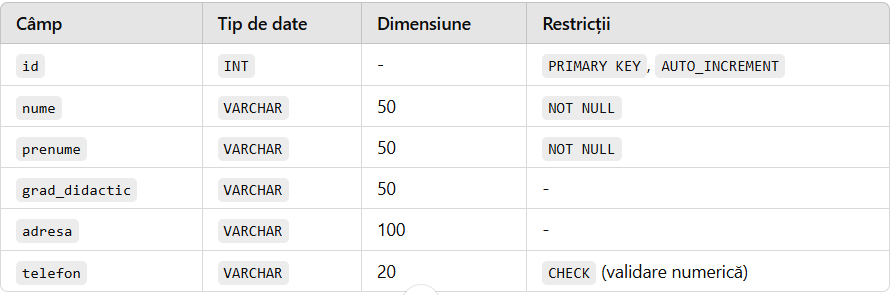


Observații:

* diriginte\_id indică profesorul care este dirigintele clasei (relație unu-la-unu).

#### ****Tabela**** profesor

* **Scop:** Stochează informații despre profesori.
* **Structură fizică:**



Câteva exemple de comenzi SQL utilizate pentru a crea aceste tabele:

CREATE TABLE utilizatori (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL UNIQUE,

parola VARCHAR(255) NOT NULL,

rol ENUM('elev', 'profesor', 'admin'),

elev\_id INT,

profesor\_id INT,

FOREIGN KEY (elev\_id) REFERENCES elev(id),

FOREIGN KEY (profesor\_id) REFERENCES profesor(id)

);

CREATE TABLE loguri (

id INT AUTO\_INCREMENT PRIMARY KEY,

utilizator\_id INT NOT NULL,

rol ENUM('elev', 'profesor', 'secretar') NOT NULL,

actiune\_logica VARCHAR(255) NOT NULL,

comanda\_sql TEXT,

data TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (utilizator\_id) REFERENCES utilizatori(id)

);

CREATE TABLE profesor\_materie (

id INT AUTO\_INCREMENT PRIMARY KEY,

profesor\_id INT NOT NULL,

materie\_id INT NOT NULL,

FOREIGN KEY (profesor\_id) REFERENCES profesor(id),

FOREIGN KEY (materie\_id) REFERENCES materie(id)

);

### **Limbaje și tehnologii utilizate**

1. **Java**  
   Java a fost utilizat ca limbaj principal pentru implementarea logicii aplicației. Acesta a fost folosit pentru gestionarea fluxurilor de lucru, procesarea datelor și interacțiunea cu baza de date.
2. **MySQL**  
   Aplicația utilizează MySQL ca sistem de gestionare a bazei de date relaționale. Acesta asigură stocarea, interogarea și manipularea eficientă a datelor, fiind integrat ușor cu Java prin intermediul driverului JDBC. Structura bazei de date a fost proiectată pentru a respecta principiile normalizării, evitând redundanțele și asigurând consistența datelor.
3. **Apache POI**  
   Apache POI este utilizat pentru generarea fișierelor Excel (format .xls sau .xlsx). Aceste fișiere permit exportul datelor într-un format ușor de utilizat de către utilizator. De exemplu, aplicația poate genera rapoarte cu toti elevii dintr-o anumita clasa.
4. **PDFBox**  
   Pentru generarea rapoartelor PDF, s-a utilizat librăria PDFBox. Aceasta oferă o soluție robustă pentru crearea documentelor PDF, utilizată pentru generarea de rapoarte oficiale, cum ar fi rezultatele elevilor sau situațiile academice.
5. **Maven**  
   Maven a fost folosit pentru gestionarea dependințelor proiectului. Acesta simplifică integrarea librăriilor externe, precum Apache POI și PDFBox, și automatizează procesul de construire al aplicației. De asemenea, Maven asigură o structură standardizată a proiectului și permite gestionarea eficientă a versiunilor.

### **Validarea datelor**

Pentru a garanta integritatea și calitatea datelor gestionate de aplicație, s-au implementat următoarele mecanisme de validare:

1. **Validarea CNP-ului**
   * **Reguli de validare**:
     + CNP-ul trebuie să conțină exact 13 caractere numerice.
     + Formatul este verificat utilizând expresii regulate CONSTRAINT `chk\_cnp` CHECK ((char\_length(`cnp`) = 13))

 Implementare:

* Validarea este realizată atât la nivel de aplicație, prin logica Java, cât și la nivel de bază de date, prin utilizarea constrângerii CHECK. Aceasta se asigură că datele introduse respectă formatul dorit.

1. Validarea numărului de telefon

* Reguli de validare:
  + Numărul de telefon trebuie să conțină exact 10 cifre.
  + Este utilizată o expresie regulată similară CONSTRAINT `chk\_telefon` CHECK ((char\_length(`telefon`) = 10))\n)

**Verificarea integrității relațiilor**

* Reguli de integritate:
  + O clasă nu poate fi ștearsă dacă are elevi asociați.
  + O materie nu poate fi ștearsă dacă există note asociate.
* Implementare:
  + La nivel de bază de date, s-au utilizat constrângerile de cheie externă (FOREIGN KEY) cu opțiuni precum ON DELETE RESTRICT pentru a preveni ștergerile care ar duce la inconsistență.

ALTER TABLE elev

ADD CONSTRAINT fk\_clasa

FOREIGN KEY (clasa\_id) REFERENCES clasa(id)

ON DELETE RESTRICT;

# **Descrierea aplicației și manualul de utilizare**

Aplicația este un sistem de gestionare a activităților școlare, care permite utilizatorilor să administreze elevi, profesori, clase, note și alte informații asociate. Interfața este intuitivă și oferă funcționalități specifice fiecărui rol definit în aplicație.

## **1. Roluri disponibile în aplicație**

Aplicația este structurată pe baza următoarelor roluri, fiecare cu funcționalități distincte:

* **Administrator**: Gestionarea utilizatorilor, profesorilor, claselor, materiilor.
* **Profesor**: Gestionarea notelor elevilor.
* **Elev**: Vizualizarea notelor și a informațiilor personale.

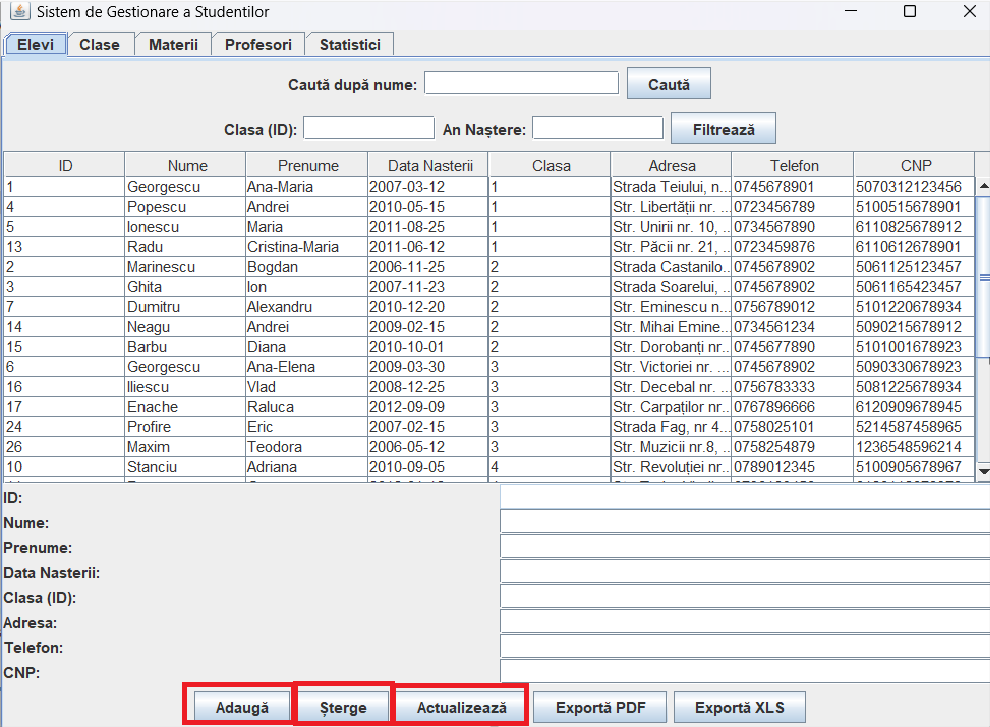
## **2. Funcționalități detaliate pe fiecare rol**

### **Rolul Administrator**

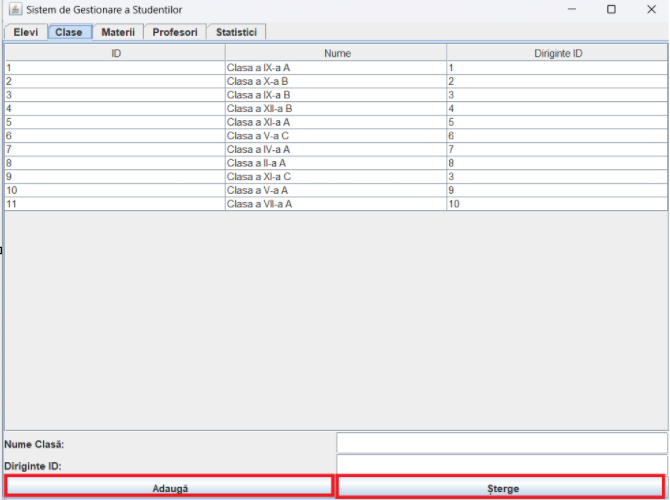
Administratorul are acces complet la toate funcționalitățile aplicației.

#### **Funcționalități disponibile**:

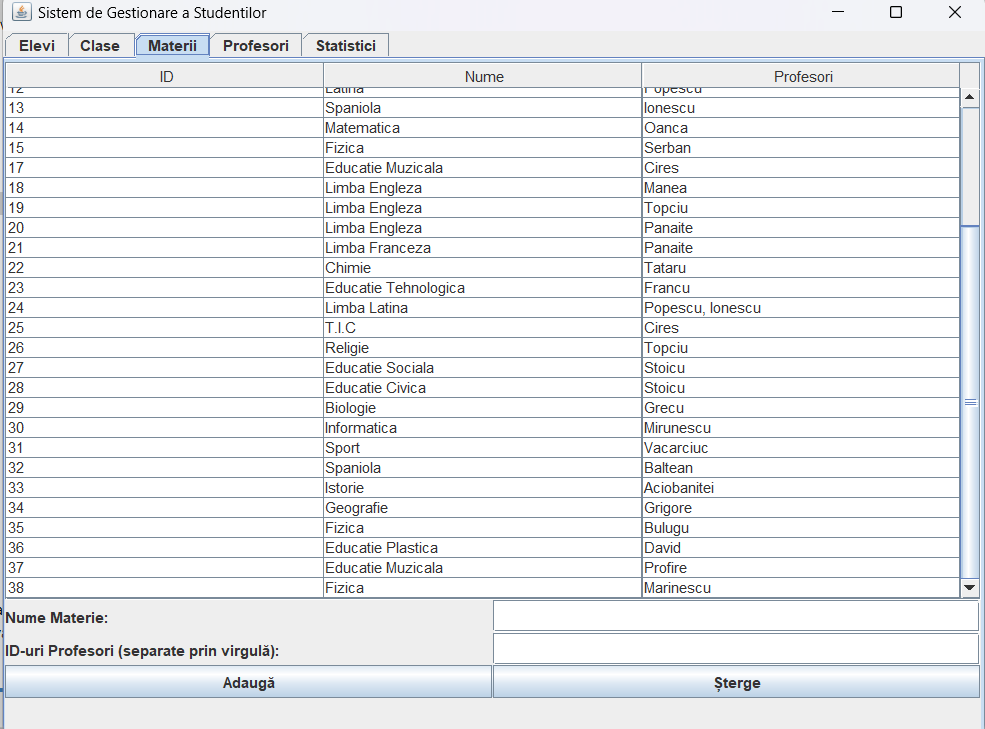
1. **Gestionarea elevilor:**
   * Adăugarea unei nou elev
   * Exmatricularea unui elev
   * Actualizarea unui elev



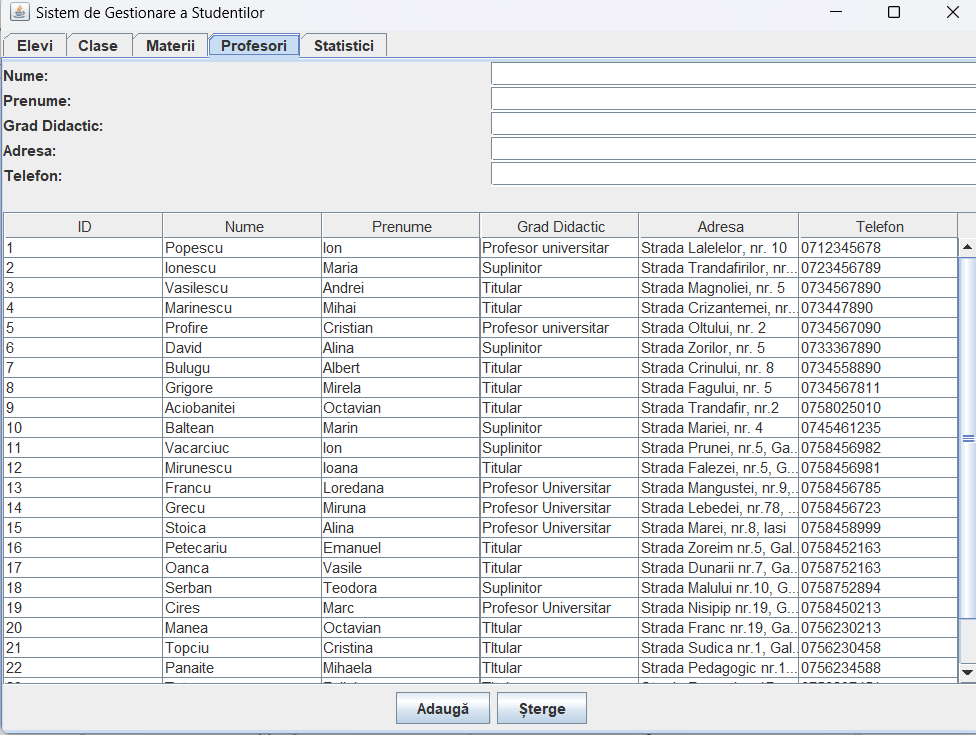
* **De asemenea, administratorul poate sa isi usureze munca, avand un camp de cautare si filtrare in susul paginii. Filtrarea se face in functie de clasa in care se afla elevul sau anul nasterii (pot fi ambele filter applicate simultan)**
* **In fereastra “clase”, administratorul poate adauga in sistem o noua clasa, cu dirigintele afferent ei. De asemenea, poate sterge o clasa(doar in cazul in care aceasta nu are elevi asociati)**

****

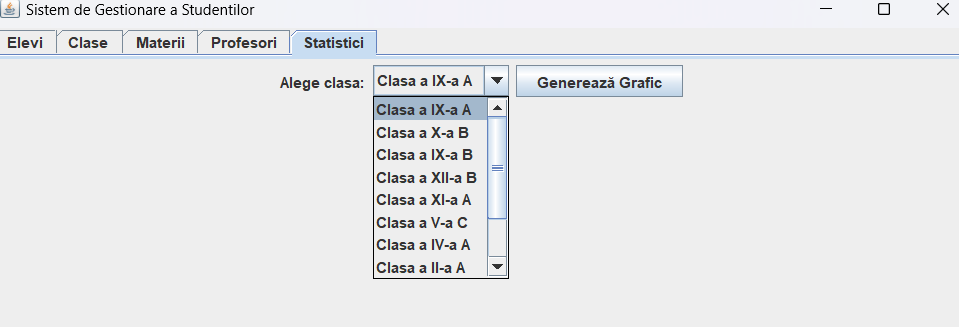
* **In continuare, administratorul poate adauga sau sterge o materie si profesorul asociat acestei materii, apasand butonul “adauga” sau “sterge”**

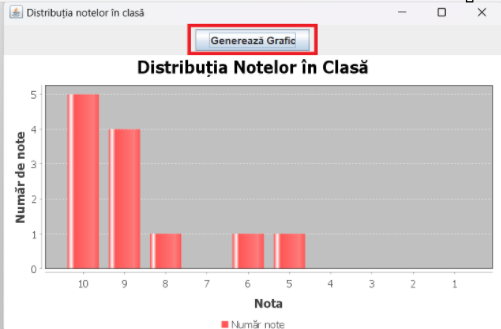
****

* **Poate adauga sau sterge un profesor din sistem**

****

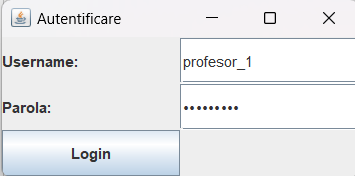
* **Poate vizualiza niste statistici, un grafic care, in functie de clasa selectata iti va arata notele elevilor din clasa respectiva**

****

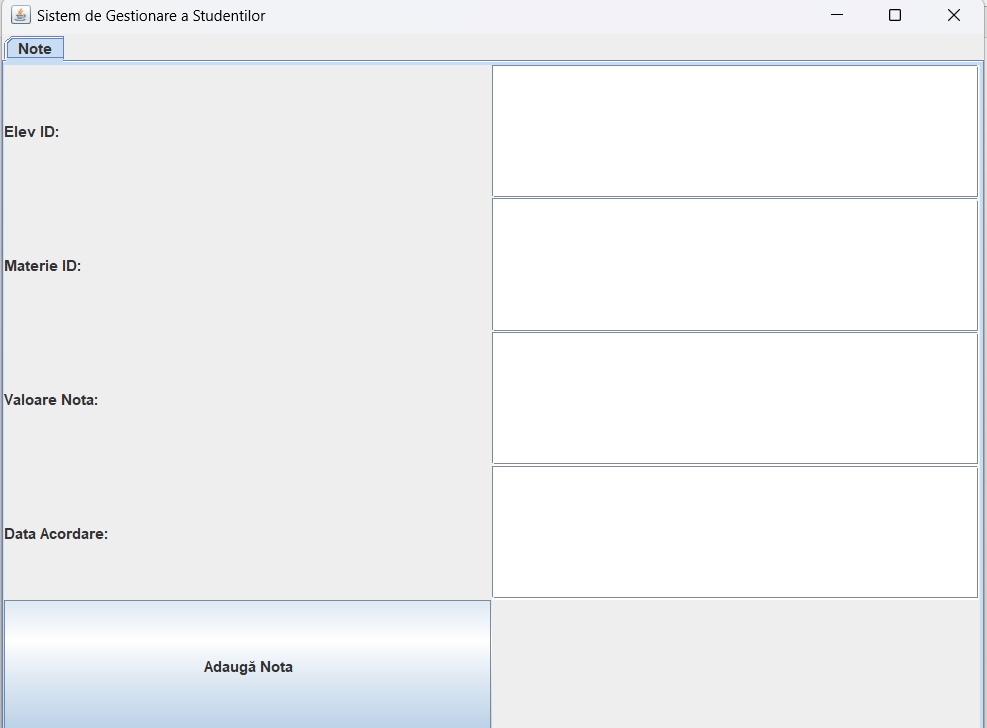
****

### **Rolul Profesor**

* Acesta trebuie sa se logheze cu id-ul lui si sa introduca parola personala

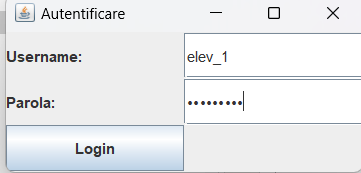


* I se va deschide o fereastra in care poate adauga nota unui elev in baza informatiilor pe care le furnizeaza



**Rolul elev**

* La randul lui, elevul trebuie sa se logheze cu id-ul lui si parola personala



* I se va deschide o fereastra in care isi poate vizualiza notele la fiecare materie in parte si data la care aceasta a fost acordata

