

**UNIVERSITATEA POLITEHNICA din BUCUREȘTI**  
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

**Proiect**

**TEHNOLOGII DE PROGRAMARE ÎN INTERNET**

**- Sistem ERP universitar - Evidența studenților -**

**Profesor Coordonator:**

Boicescu Laurențiu

**Studenți:**

Feșchiu Bianca Ana-Maria

Mărăcineanu Maria-Dimitriana

**Grupa: 433D**

București 2020

# Cuprins

1. Introducere
  - 1.1. Obiectivul lucrării
  - 1.2. Conținutul lucrării
  - 1.3. Contribuții și mod de lucru
2. Scurtă introducere teoretică
3. Contribuțiile practice ale studenților
  - 3.1. Specificare + analiză
  - 3.2. Proiectare
  - 3.3. Diagrama bazei de date
  - 3.4. Descrierea aplicației
  - 3.5. Diagrama UML
4. Concluzii
  - 4.1. Concluziile proiectului
  - 4.2. Perspective de dezvoltare ulterioară
5. Bibliografie

# 1.Introducere

## 1.1. Obiectivul lucrării

Obiectivul lucrării este reprezentat de crearea unui sistem de gestiune al studenților dintr-o universitate.

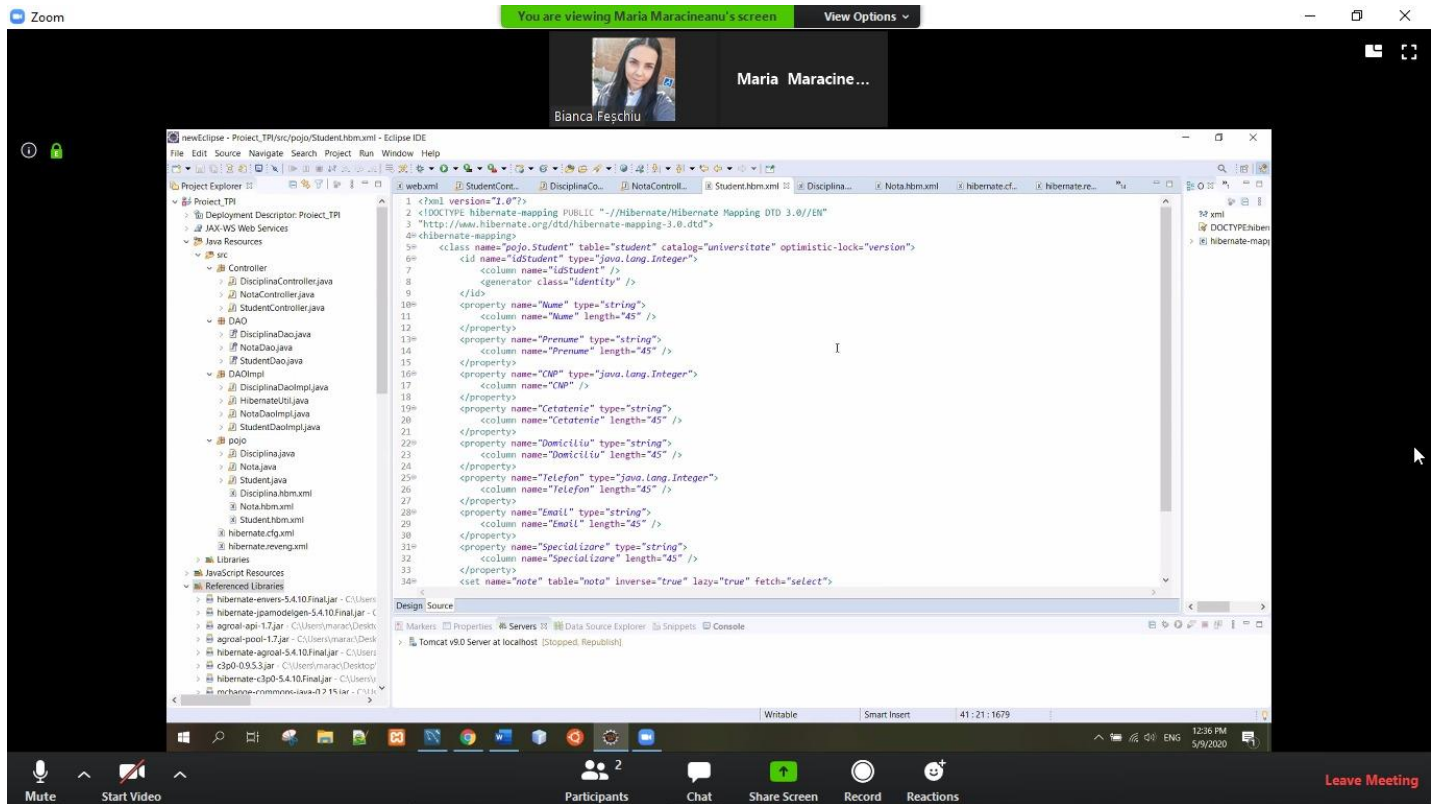
## 1.2. Conținutul lucrării

Aplicația permite unui administrator să execute operații pe tabele: vizualizare, adăugare, modificare și ștergere de date (studenți și date ale acestora precum nume, prenume, CNP, note, etc). Vizualizarea datelor tabelor de legătură (note) presupune vizualizarea datelor referite din celelalte tabele.

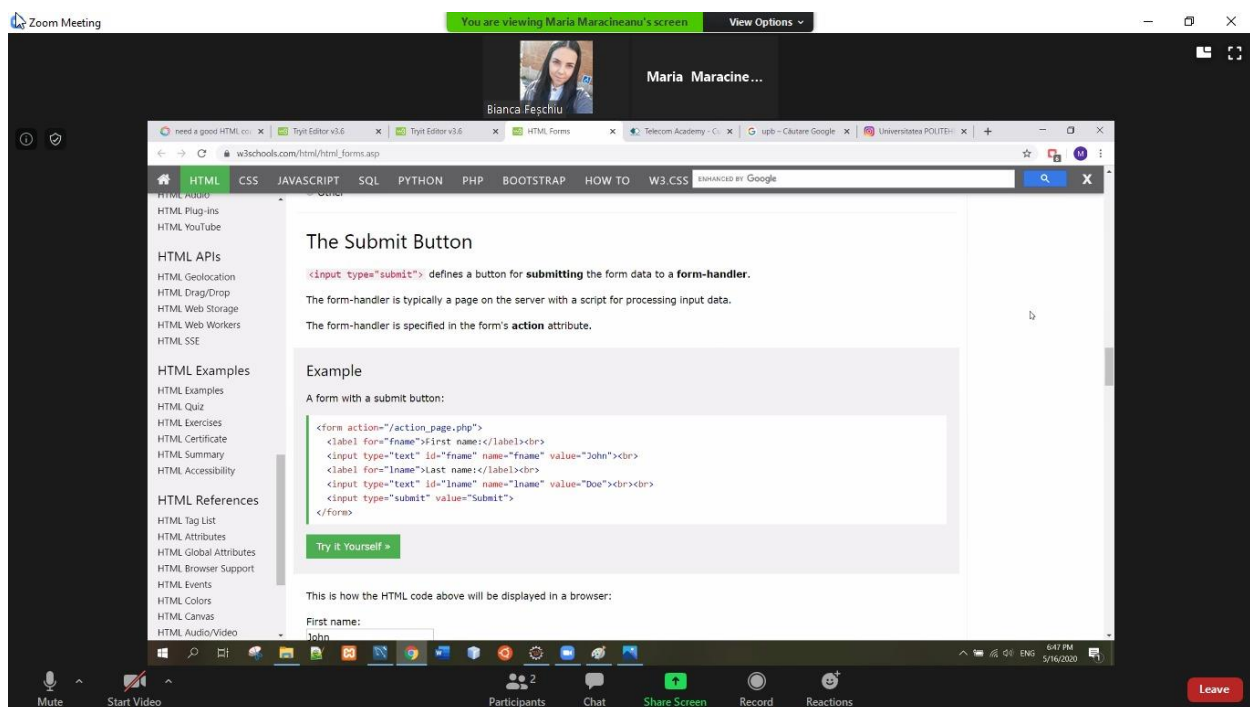
Lucrarea conține:

- Obiectivul lucrării
- O scurtă introducere teoretică
- Contribuții practice ale studenților:
  - Mod de lucru
  - Specificare, analiză, proiectare
  - Diagrama bazei de date
  - Descrierea aplicației
  - Diagrama UML
- Concluzii și perspective de dezvoltare

### 1.3. Contribuții și mod de lucru



La realizarea proiectului, am utilizat platforma Zoom, ce ne-a permis să lucrăm în același timp.



## 2. Scurtă introducere teoretică

### ❖ JAVA.

Java este un limbaj de programare orientat-obiect de nivel înalt. În cea mai mare parte, acest limbaj este utilizat pentru dezvoltarea aplicațiilor web, dar și pentru aplicațiile destinate telefoanelor sau programe adresate unor centre de date. Acesta este creat astfel încât să elimine cele mai multe erori de programare, fiind un limbaj foarte sigur. Java este disponibil pe cele mai multe dispozitive. Aici nu ne referim doar la computere, console, smartphone-uri, supercomputere, centre de data etc. Aproximativ 3 miliarde de aparate utilizează Java.

### ❖ TEHNOLOGIA JAVA SERVERPAGES.

O pagină JSP este un document text ce conține 2 tipuri de text: static (poate fi exprimat în orice tip de format bazat pe text) și conținutul JSP propriu-zis, adică cel dinamic. Această tehnologie permite accesarea mai ușoară a conținutului dinamic. JSP-ul este o extensie a tehnologiei Java servlet care este concepută pentru a permite o dezvoltare rapidă de pagini web dinamice care sunt ușor de întreținut.

Tehnologia JSP facilitează separarea interfeței de utilizator de generarea de conținut, care permite designerilor modificarea aspectului general al paginii fără a modifica conținutul dinamic care stă la baza acesteia. Tehnologia JSP utilizează etichete asemănătoare XML, denumite etichete personalizate, pentru a încapsula funcționalitatea și logica conținutului paginii. Un JSP este pur și simplu o pagină HTML care conține cod Java încorporat.

### ❖ HTML.

HyperText Markup Language este codul utilizat pentru a structura și afișa o pagină web și conținutul acesteia. El permite crearea de documente hipertext pentru paginile web. Un document HTML este divizat într-o secțiune antet (aici, între <HEAD> și </HEAD>) și o secțiune corp (aici, între < BODY> și </BODY>).

❖ MySQL.

O bază de date reprezintă un instrument pentru colectarea și organizarea informațiilor. Aceasta poate stoca informații despre persoane, produse, etc. Deși este folosit foarte des împreună cu limbajul de programare PHP, cu MySQL se pot construi aplicații în orice limbaj major.

❖ Hibernate ORM (sau simplu Hibernate).

Hibernate este un instrument de mapare obiect-relațional pentru limbajul de programare Java. Acesta oferă un cadru pentru maparea unui model de domeniu orientat pe obiecte într-o bază de date relațională. Hibernate gestionează problemele de nepotrivire a impedanței relaționale dintre obiecte prin înlocuirea acceselor directe, persistente ale bazelor de date cu funcții de gestionare a obiectelor la nivel înalt.

Hibernate este un software gratuit care este distribuit sub licența GNU Lesser General Public License 2.1.

Funcția principală a Hibernate este maparea de la clasele Java la tabelele bazei de date și maparea de la tipurile de date Java la tipurile de date SQL. Hibernate oferă, de asemenea, facilități de interogare și recuperare a datelor. Acesta generează apeluri SQL și îl scutește pe dezvoltator de manipularea manuală și conversia obiectelor din setul de rezultate.

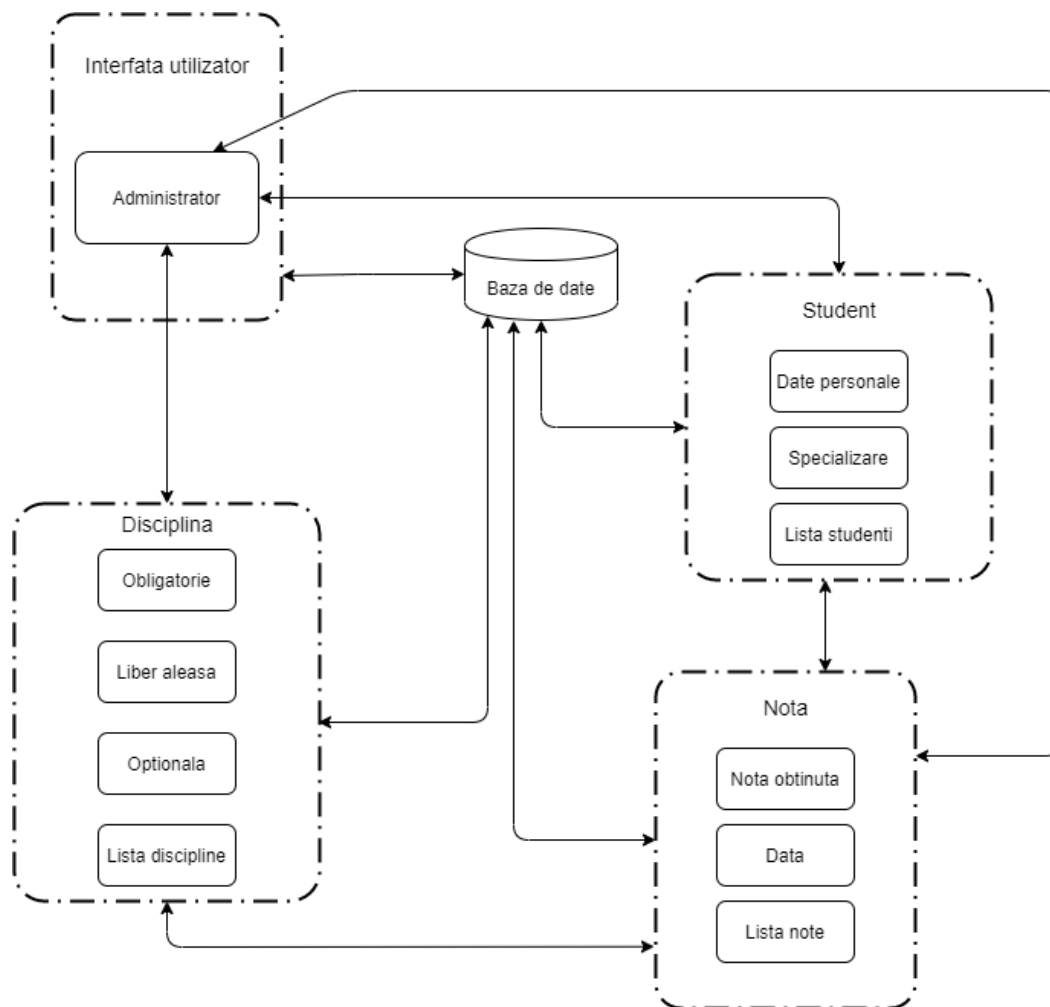
### 3. Contribuții practice ale studenților

#### 3.1. Specificare + Analiză

- Administrator
  - Username
  - Parola
- Date personale – tabel student
  - Nume
  - Prenume
  - CNP
  - Cetățenie
  - Domiciliu
  - Telefon
  - E-mail
  - Specializare
- Disciplină – tabel disciplina
  - Denumire
  - Tip
  - Profesor
  - Puncte Credit
- Situație școlară – tabel nota
  - Nota obținută
  - Data
- Materii la alegere
  - Opționale
  - Obligatorii
  - Liber alese
- Opțiuni specializare
- Listă studenți
- Listă discipline
- Listă note

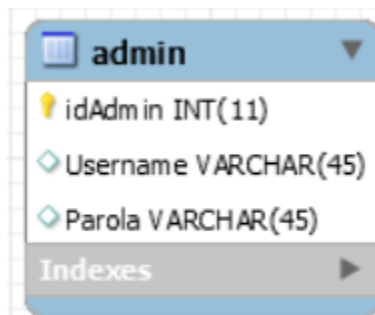
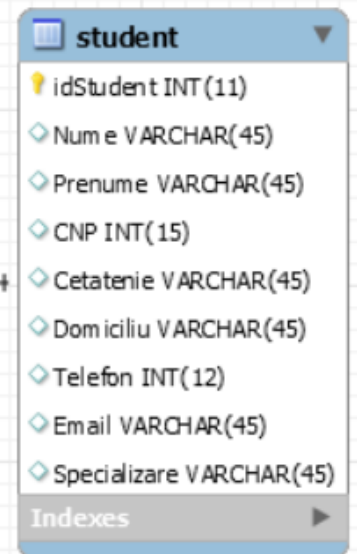
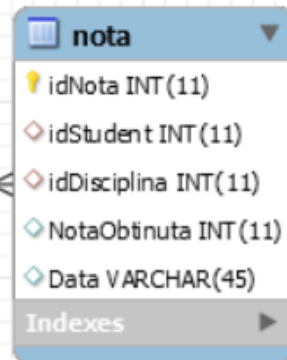
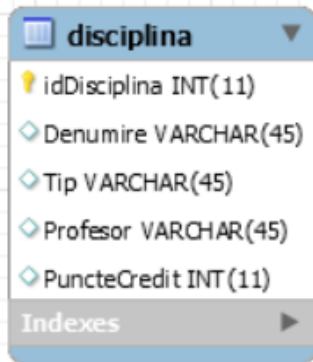
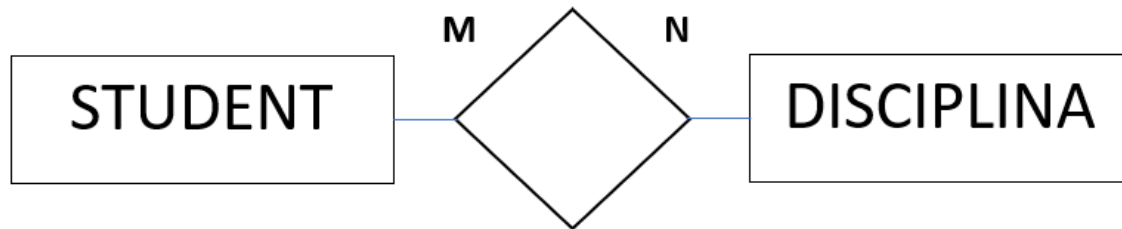
- Funcționalități:
  - Adăugare
  - Ștergere
  - Modificare

### 3.2. Proiectare





### 3.3. Diagrama bazei de date

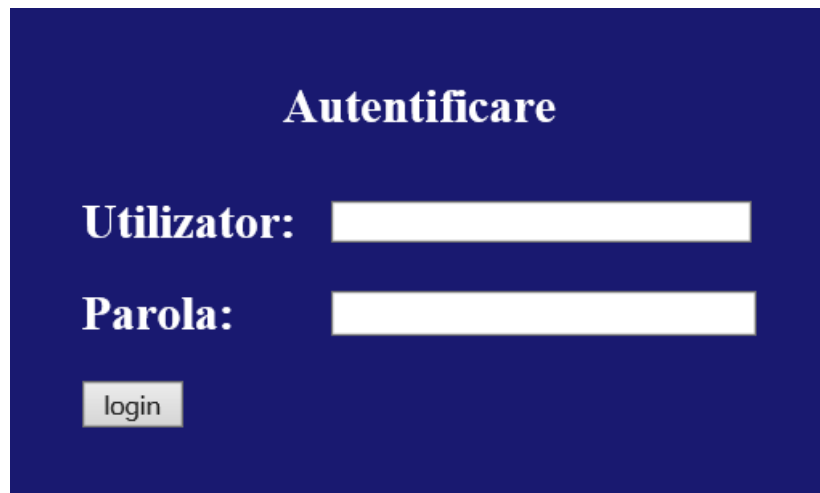


### 3.4. Descrierea aplicației

Aplicația conține o bază de date creată în sistemul de gestiune a bazelor de date MySql și o interfață creată folosind tehnologia Hibernate.

Prima pagină – Autentificare

În această pagină (login.jsp) se loghează administratorul ce gestionează datele.

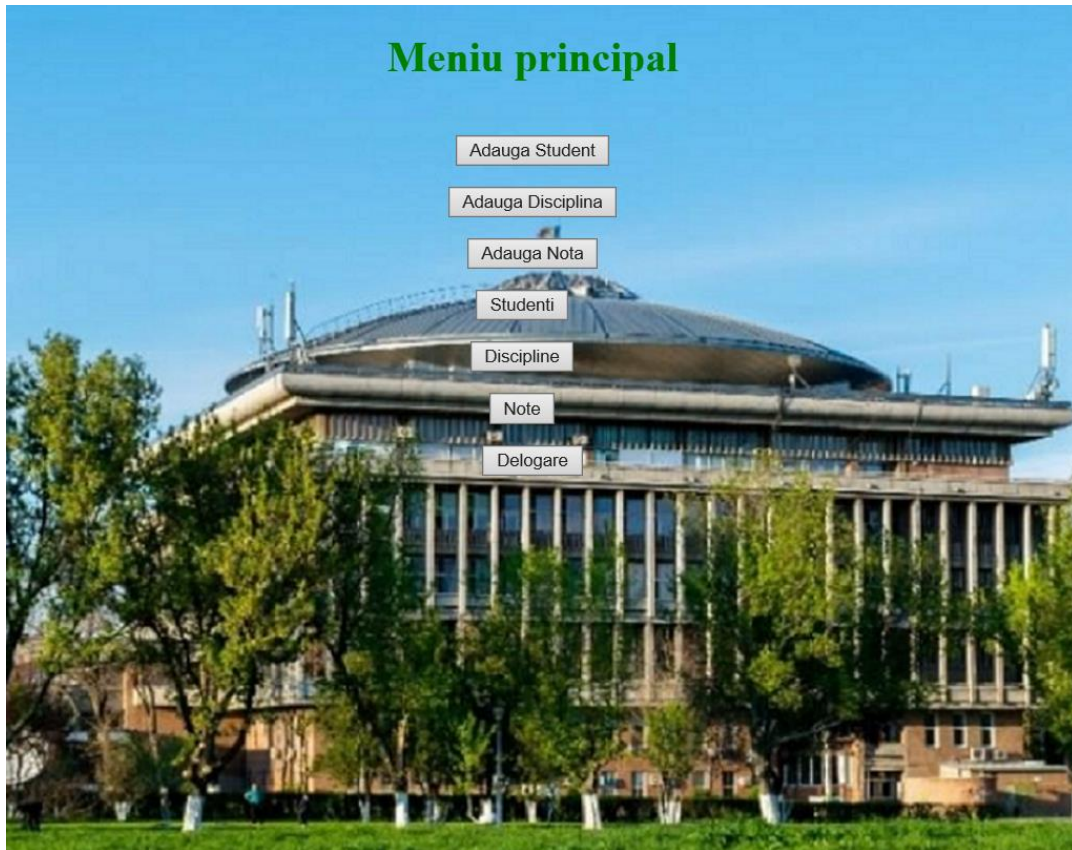


home.jsp

```
String driverName = "com.mysql.cj.jdbc.Driver";  
String url = "jdbc:mysql://localhost:3306/administrator";  
String user = "root";  
String dbpsw = "newrootpassword";  
String sql = "select * from admin where Username=? and Parola=?";  
String name = request.getParameter("Username");  
String password = request.getParameter("Parola");
```

În cazul în care datele introduse sunt incorecte, aplicația afisează o eroare la autentificare (pagina error.jsp).

După logare, prin apăsarea butonului “login”, se trece la pagina următoare: Meniu.html.



- Prin apăsarea butonului “Adauga Student”, se trece la pagina adauga\_Student.jsp.



### *Adauga student*

Nume:	<input type="text"/>
Prenume:	<input type="text"/>
CNP:	<input type="text"/>
Cetatenie:	<input type="text"/>
Domiciliu:	<input type="text"/>
Telefon:	<input type="text"/>
Email:	<input type="text"/>
Specializare:	<input type="text" value="MON/ELA/RST/TST"/>
<input type="button" value="Adauga"/>	

Vezi studentii:

[Meniu principal](#)



Se completează câmpurile corespunzătoare datelor studentului. Prin apăsarea butonului “Adauga”, se apelează funcția doGet() – adaugaStudent din clasa StudentController, din pachetul Controller:

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    if (request.getParameter("adaugaStudent") != null) {
        String Nume = request.getParameter("Nume");
        String Prenume = request.getParameter("Prenume");
        String CNP = request.getParameter("CNP");
        String Cetatenie = request.getParameter("Cetatenie");
        String Domiciliu = request.getParameter("Domiciliu");
        String Telefon = request.getParameter("Telefon");
        String Email = request.getParameter("Email");
        String Specializare = request.getParameter("Specializare");
        student.setNume(Nume);
        student.setPrenume(Prenume);
        student.setCNP(Integer.parseInt(CNP));
        student.setCetatenie(Cetatenie);
        student.setDomiciliu(Domiciliu);
        student.setTelefon(Integer.parseInt(Telefon));
        student.setEmail(Email);
        student.setSpecializare(Specializare);
        studentDaoImpl.adaugaStudent(student);
        RequestDispatcher rd = request.getRequestDispatcher
            ("adauga_Student.jsp");
        rd.forward(request, response);
    }
}
```

- Similar pentru Disciplina și Nota:

Prin apăsarea butoanelor “Adauga Disciplina”, respectiv “Adauga Nota”, se vor deschide paginile adauga\_Disciplina.jsp, respectiv adauga\_Nota.jsp, având funcționalități asemănătoare:

### *Adauga disciplina*

Denumire:	<input type="text" value="Denumire"/>
Tip:	<input type="text" value="obligatorie/optionala/facultativa"/>
Profesor:	<input type="text" value="Profesor"/>
Puncte Credit:	<input type="text" value="Puncte credit"/>
<input type="button" value="Adauga"/>	

Vezi disciplinele:

[Meniu principal](#)

### *Adauga nota*

Student:	<input type="text" value="7, Popescu, Ion, 123456, romana, Bucuresti, 729678543, popescu.ion@yahoo.com, RST"/>
Disciplina:	<input type="text" value="4, TTI, obligatorie, Daniela Faur, 5"/>
Nota Obtinuta :	<input type="text" value="nota"/>
Data:	<input type="text" value="data"/>
<input type="button" value="Adauga"/>	

Vezi notele:

[Meniu principal](#)

Pentru cele 3 pagini, butonul “Afiseaza/Modifica” va redirectiona utilizatorul către paginile Studenti.jsp, Discipline.jsp, Note.jsp, unde sunt afișate listele cu studenți, respectiv discipline și note.



### Lista studenți

IdStudent	Nume	Prenume	CNP	Cetatenie	Domiciliu	Telefon	Email	Specializare	Modificari
7	Popescu	Ion	123456789	romana	Bucuresti	729645789	popescu.ion@yahoo.com	RST	<input type="text"/> Nume <input type="text"/> Prenume <input type="text"/> CNP <input type="text"/> Cetatenie <input type="text"/> Domiciliu <input type="text"/> Telefon <input type="text"/> Email <input type="text"/> Specializare <input type="button" value="Modifica"/> <input type="button" value="Sterge"/>
8	Ionescu	Elena	234567213	romana	Cluj	721678045	elena.ionescu@gmail.com	MON	<input type="text"/> Nume <input type="text"/> Prenume <input type="text"/> CNP <input type="text"/> Cetatenie <input type="text"/> Domiciliu <input type="text"/> Telefon <input type="text"/> Email <input type="text"/> Specializare <input type="button" value="Modifica"/> <input type="button" value="Sterge"/>
9	Dinca	Alexandru	156773213	romana	Iasi	745893710	alex.dinca12@yahoo.com	ELA	<input type="text"/> Nume <input type="text"/> Prenume <input type="text"/> CNP <input type="text"/> Cetatenie <input type="text"/> Domiciliu <input type="text"/> Telefon <input type="text"/> Email <input type="text"/> Specializare <input type="button" value="Modifica"/> <input type="button" value="Sterge"/>



### Lista discipline

IdDisciplina	Denumire	Tip	Profesor	Puncte Credit	Modificari
4	TTI	obligatorie	Daniela Faur	5 puncte	<input type="text"/> Denumire <input type="text"/> Tip <input type="text"/> Profesor <input type="text"/> Puncte credit <input type="button" value="Modifica"/> <input type="button" value="Sterge"/>
5	Microunde	obligatorie	Teodor Petrescu	5 puncte	<input type="text"/> Denumire <input type="text"/> Tip <input type="text"/> Profesor <input type="text"/> Puncte credit <input type="button" value="Modifica"/> <input type="button" value="Sterge"/>
6	CIA	obligatorie	Cosmin Popa	5 puncte	<input type="text"/> Denumire <input type="text"/> Tip <input type="text"/> Profesor <input type="text"/> Puncte credit <input type="button" value="Modifica"/> <input type="button" value="Sterge"/>
7	IEM	obligatorie	Alexandru Vulpe	4 puncte	<input type="text"/> Denumire <input type="text"/> Tip <input type="text"/> Profesor <input type="text"/> Puncte credit <input type="button" value="Modifica"/> <input type="button" value="Sterge"/>

În această pagină, se pot șterge date din tabel (buton Sterge) sau se pot modifica prin completarea spațiilor și apăsarea butonului Modifica.

Funcția apelată este doPost() din StudentController, respectiv DisciplinaController si NotaController:

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    if (request.getParameter("afiseazaStudent") != null) {
        List<Student> listaStudent = new ArrayList();
        listaStudent = studentDaoImpl.afiseazaStudent();
        request.setAttribute("listaStudent", listaStudent);
        RequestDispatcher rd = request.getRequestDispatcher("Studenti.jsp");
        rd.forward(request, response);
    }

    if (request.getParameter("modificaStudent") != null) {
        int id1 = Integer.parseInt(request.getParameter("idStudent"));
        String Nume = request.getParameter("Nume");
        String Prenume = request.getParameter("Prenume");
        String CNP = request.getParameter("CNP");
        String Cetatenie = request.getParameter("Cetatenie");
        String Domiciliu = request.getParameter("Domiciliu");
        String Telefon = request.getParameter("Telefon");
        String Email = request.getParameter("Email");
        String Specializare = request.getParameter("Specializare");
        studentDaoImpl.modificaStudent(id1, Nume, Prenume,
        Integer.parseInt(CNP), Cetatenie, Domiciliu, Integer.parseInt(Telefon), Email,
        Specializare);
        RequestDispatcher rd =
        request.getRequestDispatcher("adauga_Student.jsp");
        rd.forward(request, response);
    }

    if (request.getParameter("stergeStudent") != null) {
        int id2 = Integer.parseInt(request.getParameter("idStudent"));
        student.setIdStudent(id2);
        studentDaoImpl.stergeStudent(student);
        RequestDispatcher rd =
        request.getRequestDispatcher("adauga_Student.jsp");
        rd.forward(request, response);
    }
}
```

Pentru note, afișarea este diferită:

### *Lista note*

Nume	Prenume	CNP	Cetatenie	Domiciliu	Telefon	Email	Specializare	Denumire	Tip	Profesor	PuncteCredit	NotaObtinuta	Data
Popescu	Ion	123456788	romana	Bucuresti	729645789	popescu.ion@yahoo.com	RST	TTI	obligatorie	Daniela Faur	5	8	20.10.2019
Popescu	Ion	123456788	romana	Bucuresti	729645789	popescu.ion@yahoo.com	RST	TTI	obligatorie	Daniela Faur	5	5	15.09.2019
Ionescu	Elena	234563213	romana	Cluj	721678045	elena.ionescu@gmail.com	MON	Microunde	obligatorie	Teodor Petrescu	5	5	13.10.2019
Georgescu	Mirela	256478954	romana	Constanta	720986345	mirela.georgescu@yahoo.com	TST	PAI	optionala	Valentin Pupezescu	3	10	9.01.2020

Modifica: ☐ Sterge: ☐

idNota: 4 ▼

idStudent: 7, Popescu, Ion, 123456788, romana, Bucuresti, 729645789, popescu.ion@yahoo.com, RST ▼

idDisciplina: 4, TTI, obligatorie, Daniela Faur, 5 ▼

Modifica Nota:

Modifica Data:

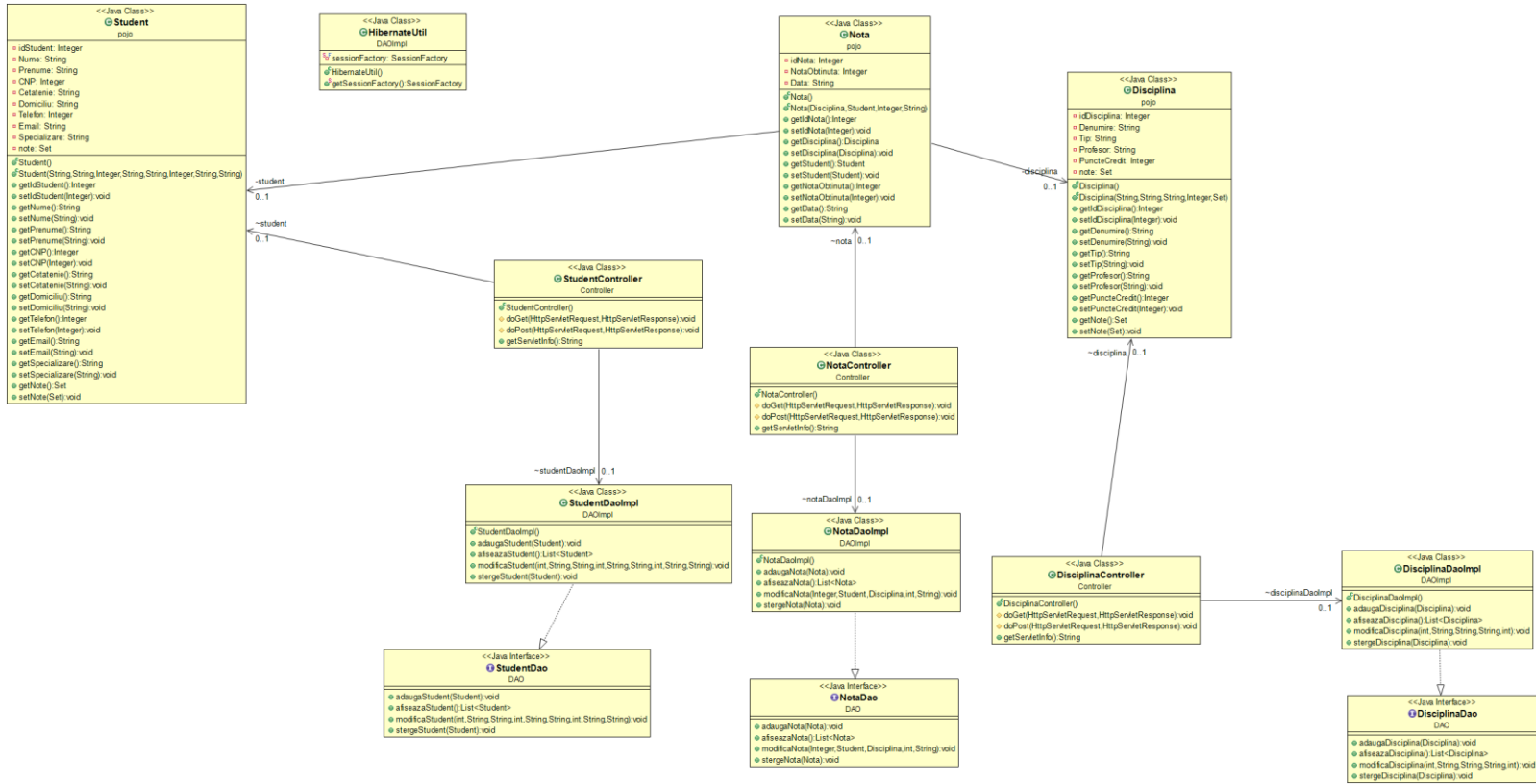
### Meniu principal

Se va bifa Modifică sau Șterge, iar în urma acestei acțiuni va apărea un buton pentru modificare, respectiv ștergere. Se va selecta studentul și disciplina unde se dorește modificarea, precum și id-ul notei.

- Prin apăsarea butoanelor “Studenti”, “Discipline”, “Note” (din meniul principal), se trece la paginile Studenti.jsp, Discipline.jsp, Note.jsp, corespunzătoare listelor cu studenți, discipline, respectiv note.
- Butonul “Delogare” trimite utilizatorul înapoi la pagina de autentificare.

Referința “Meniu principal” se regăsește în toate paginile și trimite utilizatorul la pagina Meniu.html.

### 3.5. Diagrama UML





## 4. Concluzii

### 4.1. Concluziile proiectului

Crearea unei interfețe în tehnologia Hibernate facilitează lucrul cu bazele de date, o pagină web fiind ușor de accesat și de utilizat de către orice utilizator, având un aspect mai familiar.

### 4.2. Perspective de dezvoltare ulterioară

- Îmbunătățirea aspectului interfeței. De exemplu pozele de pe fundal se vor adapta dimensiunii ecranului pe care este rulată aplicația.
- Studenții vor fi afișați în ordine alfabetică și în funcție de specializare. De asemenea, disciplinele vor fi afișate în funcție de tipul lor (obligatorii/ opționale/ facultative).
- Notele vor fi afișate pentru fiecare student în parte.
- Vor fi create pagini JSP pentru:
  - Afișarea studenților în funcție de medie
  - Selectarea specializării
  - Afișarea studenților bursieri
  - Afișarea studenților restanțieri. În această pagină va fi ținută și evidența plăților restanțelor.
- În urma introducerii datelor, vor apărea pe ecran ferestre de dialog conținând mesaje de confirmare precum “Studentul a fost adăugat”. De asemenea, se pot crea pagini separate pentru afișarea acestor mesaje.

# Bibliografie

Wikipedia Hibernate(Framework): [https://en.wikipedia.org/wiki/Hibernate\\_\(framework\)](https://en.wikipedia.org/wiki/Hibernate_(framework))  
<https://hibernate.org/orm/what-is-an-orm/>  
<http://discipline.elcom.pub.ro/tpi/>