**UNIVERSITATEA NAȚIONALĂ DE ȘTIINȚĂ ȘI TEHNOLOGIE POLITEHNICA BUCUREȘTI**

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

**Proiect**

**Programarea Interfețelor pentru Baze de Date**

Student: Cernat Alexandru Florin

Grupa: 431D

Profesor coordonator:

Ș.l. Dr. Ing. Pupezescu Valentin

București 2024

**Cuprins**

[1. Cerințe 3](#_Toc158264816)

[2. Tehnologii utilizate 3](#_Toc158264817)

[2.1. MySQL 3](#_Toc158264818)

[2.2. Java Server Pages (JSP) 3](#_Toc158264819)

[2.3. Apache Tomcat 4](#_Toc158264820)

[2.4. HTML 4](#_Toc158264821)

[3. Descrierea aplicației 5](#_Toc158264822)

[3.1. Baza de date 5](#_Toc158264823)

[3.2. Diagrama ERD 6](#_Toc158264824)

[3.3. Implementarea clasei JavaBean 7](#_Toc158264825)

[3.4. Implementarea JSP-urilor 11](#_Toc158264826)

[4. Concluzii ..............................................................................................................................14](#_Toc158264827)

5. Bibliografie ..........................................................................................................................14

## **Cerințe**

Creaţi două aplicaţii care să conţină o bază de date creată în sistemul de gestiune a bazelor de date MySql şi două interfeţe la aceasta (baza de date este comună).

La crearea interfeţelor se vor folosi două tehnologii (la alegere - ex.: JSP, Hibernate, JPA, .NET, Python, JavaFX, Spring Boot etc.). Baza de date va fi compusă din tabelele: Pupils şi Disciplines, cu asociere de tip M:N.

Interfeţele vor trebui să permită utilizatorului să execute următoarele operaţii pe toate tabele (inclusiv pe cele de legătura dacă aveti asocieri de tip M:N): vizualizare, adăugare, modificare şi ştergere de date. Vizualizarea tabelelor de legătură va presupune vizualizarea datelor referite din celelalte tabele.

Pentru tema primită, se aleg tehnologiile diferite JSP și Java Spring Boot. În prezentul document, se va face referire la tehnologia JSP.

## **Tehnologii utilizate**

### **2.1 MySQL**

MySQL este un sistem de gestionare a bazelor de date relaționale, utilizat frecvent în dezvoltarea aplicațiilor web și a altor soluții software. Acesta permite stocarea și administrarea eficientă a datelor, cum ar fi cele referitoare la angajați, produse, vânzări sau inventar. La fel ca alte sisteme de baze de date, precum Microsoft Access, MySQL oferă funcționalități esențiale pentru organizarea și interogarea datelor. Totuși, MySQL este preferat pentru proiecte de mare amploare datorită capacității sale de scalare și performanței ridicate, în timp ce Microsoft Access este mai potrivit pentru aplicații de dimensiuni mici și medii datorită ușurinței în utilizare.1]

### **2.2 Java Server Pages (JSP)**

JavaServer Pages (JSP) este o tehnologie care le permite dezvoltatorilor să creeze pagini web dinamice prin combinarea codului Java cu limbaje de marcare precum HTML sau XML. Paginile JSP includ atât elemente statice (HTML, XML), cât și componente dinamice scrise în Java, generând astfel conținut personalizat pe server înainte de a fi trimis către utilizator. Această metodă asigură portabilitate, scalabilitate și eficiență, fiind o alegere populară pentru dezvoltatorii care construiesc aplicații web complexe și adaptabile. [2]

**2.3 Apache Tomcat**

Apache Tomcat este un server web open-source și un container de servlet-uri Java dezvoltat de Apache Software Foundation. Acesta furnizează un mediu de rulare pentru aplicații web Java, permițând implementarea și gestionarea lor într-un mod eficient și scalabil.

Tomcat acționează ca un server web sau un mediu de execuție pentru aplicațiile Java, gestionând cererile HTTP de la clienți și facilitând executarea servlet-urilor și JSP-urilor pentru generarea răspunsurilor dinamice.[3]

**2.4 HTML**

HyperText Markup Language (HTML) este limbajul standard pentru crearea și structurarea paginilor web. Prin utilizarea unei varietăți de elemente și atribute, HTML permite dezvoltatorilor să definească structura conținutului, incluzând paragrafe, titluri, linkuri și imagini. Această structură semantică este esențială nu doar pentru prezentarea vizuală și organizarea conținutului în browser, ci și pentru accesibilitate și optimizarea pentru motoarele de căutare. HTML funcționează în strânsă legătură cu CSS pentru stilizare și JavaScript pentru funcționalități interactive, formând împreună fundamentul dezvoltării web moderne. Această combinație permite crearea de site-uri web interactive și atractive.[4]

## 

## **3. Descrierea aplicației**

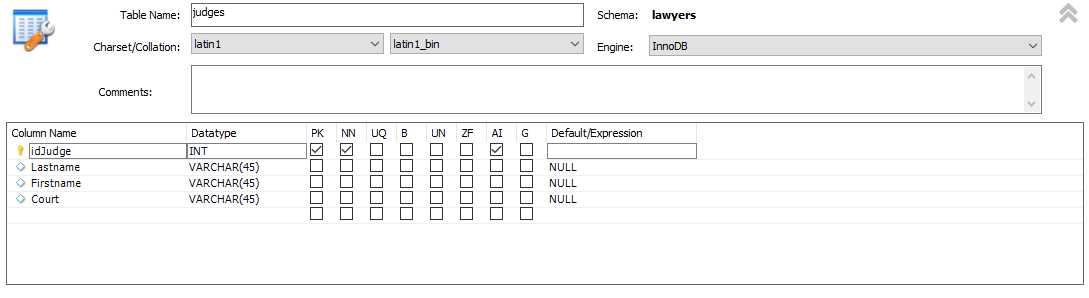
### **Baza de date**

Tema dată presupune asocierea M:N a două tabele, „Judges” și „Trials”.

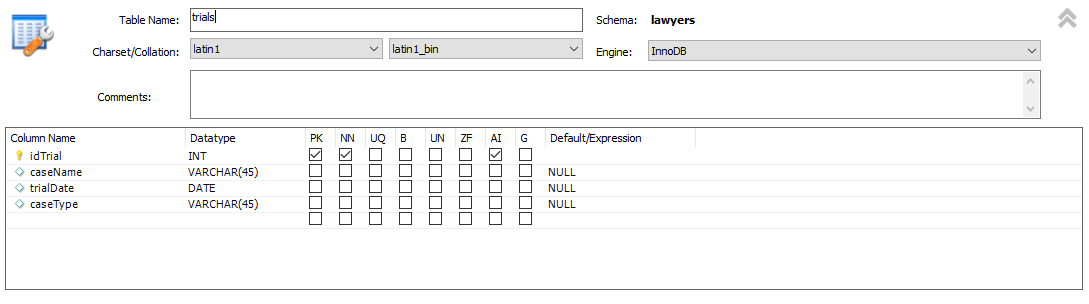
Asocierea M:N (Many-to-Many) se caracterizează prin faptul că fiecărui element dintr-o tabelă îi pot fi asociate mai multe elemente din cealaltă tabelă și invers.

În acest caz, unui „judge” îi pot fi asociate mai multe „trials”, la fel cum și unui „trial” îi pot fi asociați mai mulți judecători.

În cele ce urmează, se vor prezenta relațiile dintre aceste entități, împreună cu atributele specifice fiecăreia.  
**Tabel 1 - Judges:**

**

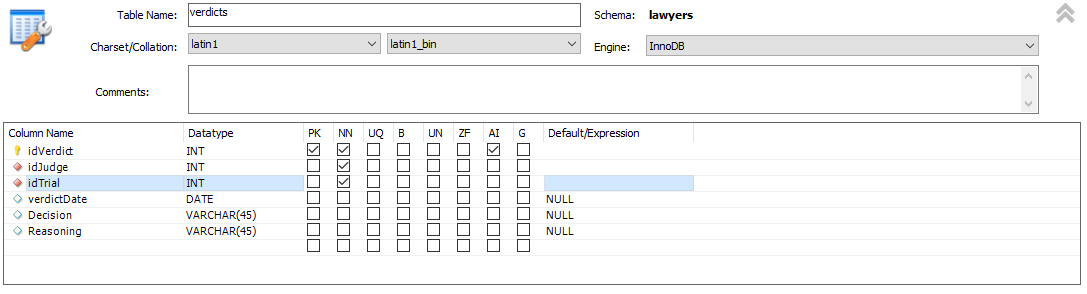
**Tabel 2 - Trials**

**

Pentru o asociere M:N este necesară crearea unei relații de legătură, în acest caz, „verdicts”, o a treia tabelă de joncțiune. Scopul acesteia este de a diviza asocierea many-to-many în două asocieri de tip 1:N.

În această nouă tabelă, atributele ce au fost selectate drept chei primare pentru tabelele de mai sus, vor deveni chei străine pentru tabela de legătură.

**Tabel 3 – Verdicts:**

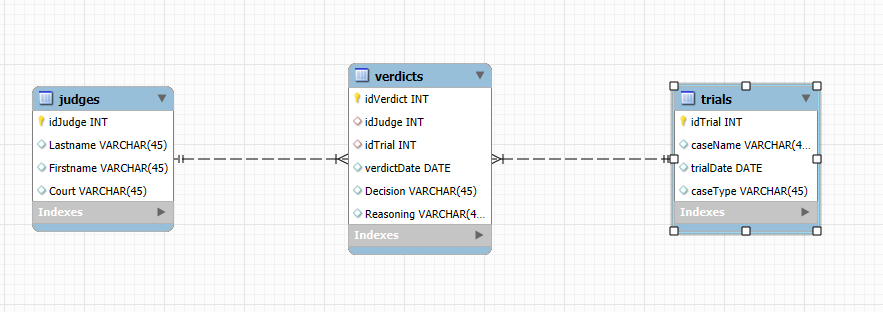


### **Diagrama ERD**

Diagrama ERD oferă o reprezentare vizuală compleza care iltrusrează entitățile, relațiile și contrângerile acestora, oferint o perspectivă detaliată a structurii și interconexiunilor din cadrul bazei de date.

Asocierile dintre tabele:

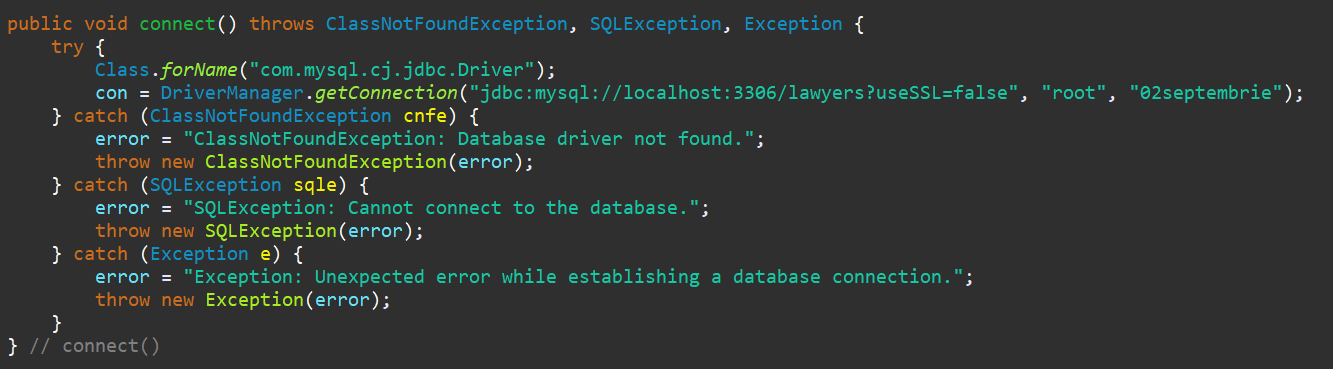
* Între „judges” și „trials” este o asociere M:N;
* Între „judges” și „verdicts” este o asociere 1:N;
* Între „trials” și „verdicts” este o asociere 1:N.

  
 **3.3 Implementarea clasei JavaBean**

Clasa JavaBean servește ca un strat de intermediere între aplicația Java și baza de date MySQL, oferind metode pentru conectarea la baza de date, manipularea datelor și gestionarea excepțiilor.

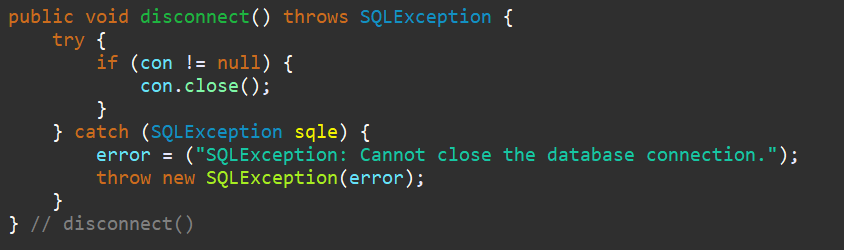
**Iniţializare şi conectare la baza de date**

Metodele connect(), connect(String bd) şi connect(String bd, String ip) sunt utilizate pentru a stabiliti conexiunea cu baza de date MySQL, folosind driver-ul JDBC și credențialele specificate.



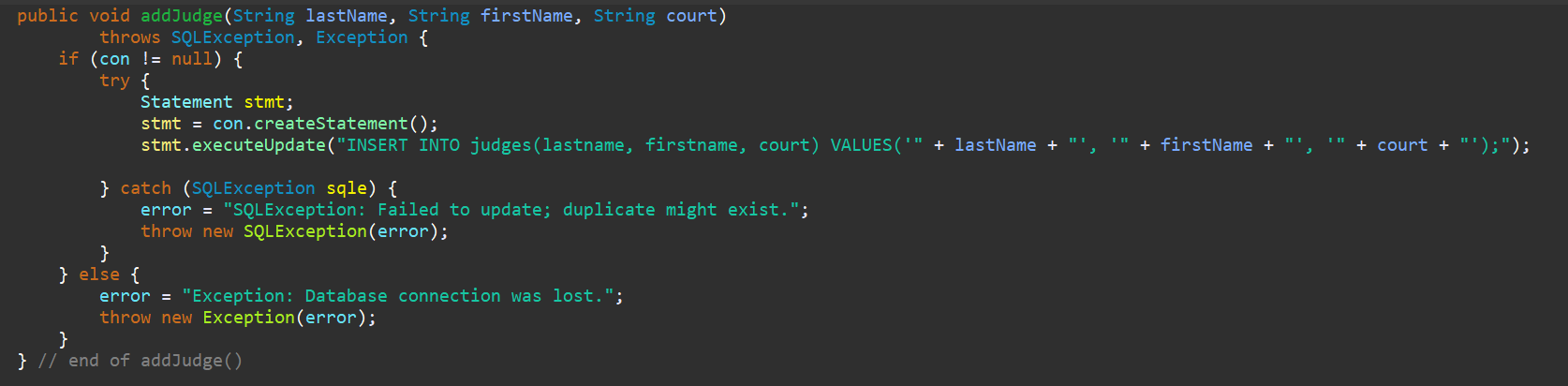
**Închiderea conexiunii**

Metoda disconnect() închide conexiunea cu baza de date, asigurând gestionarea corespunzătoare a resurselor.

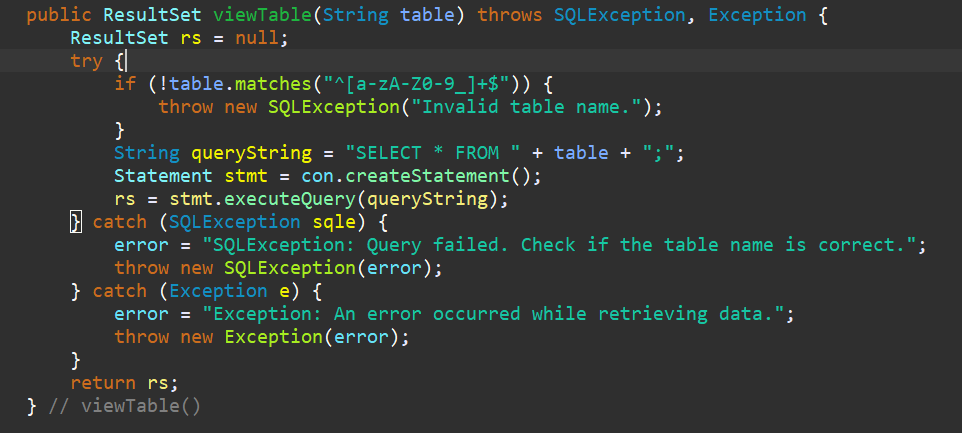


**Manipularea Datelor**

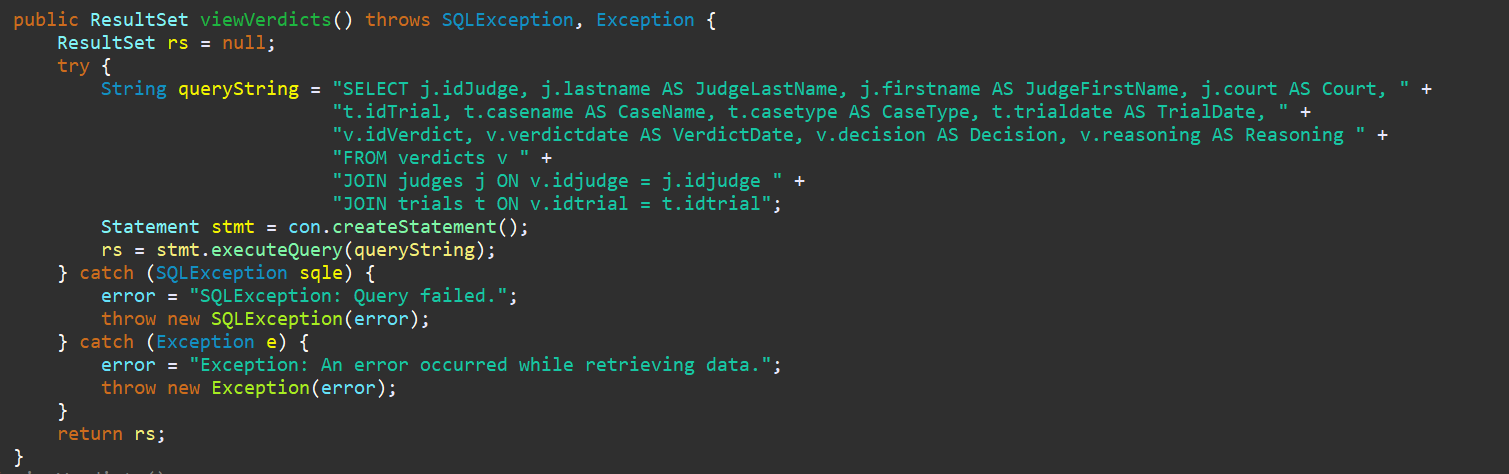
Metodele addJudge(), addTrial() și addVerdict() sunt folosite pentru a insera noi înregistrări în tabelele corespunzătoare.



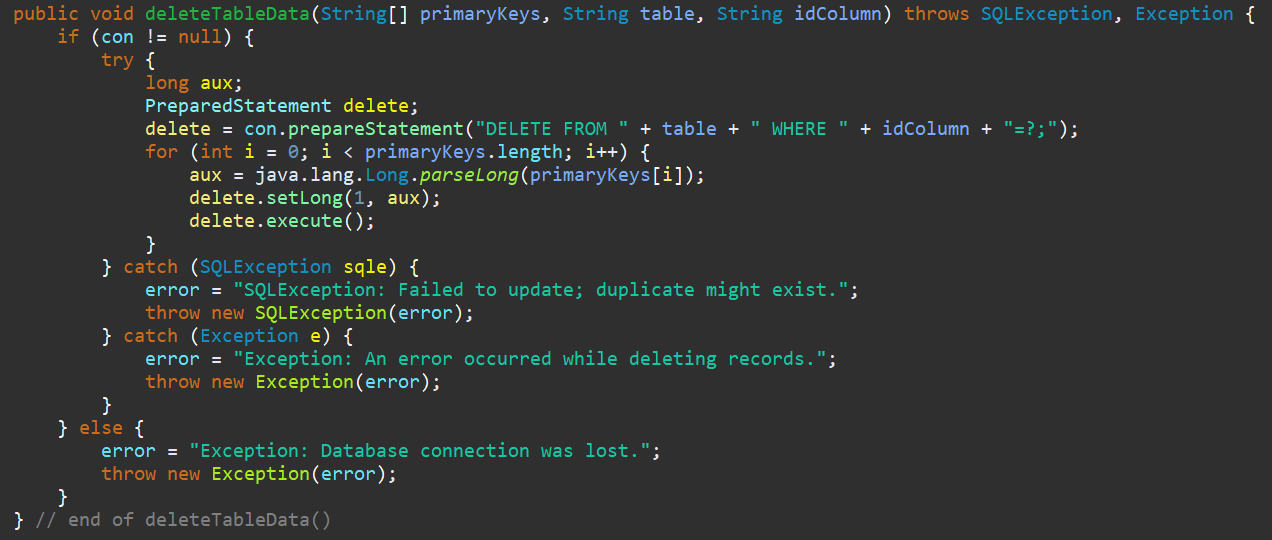
Metoda viewTable(String table) permite extragerea şi vizualizarea datelor din tabele.



Pentru tabela de legatura „Verdicts”, se realizează un inner-join între tabele.



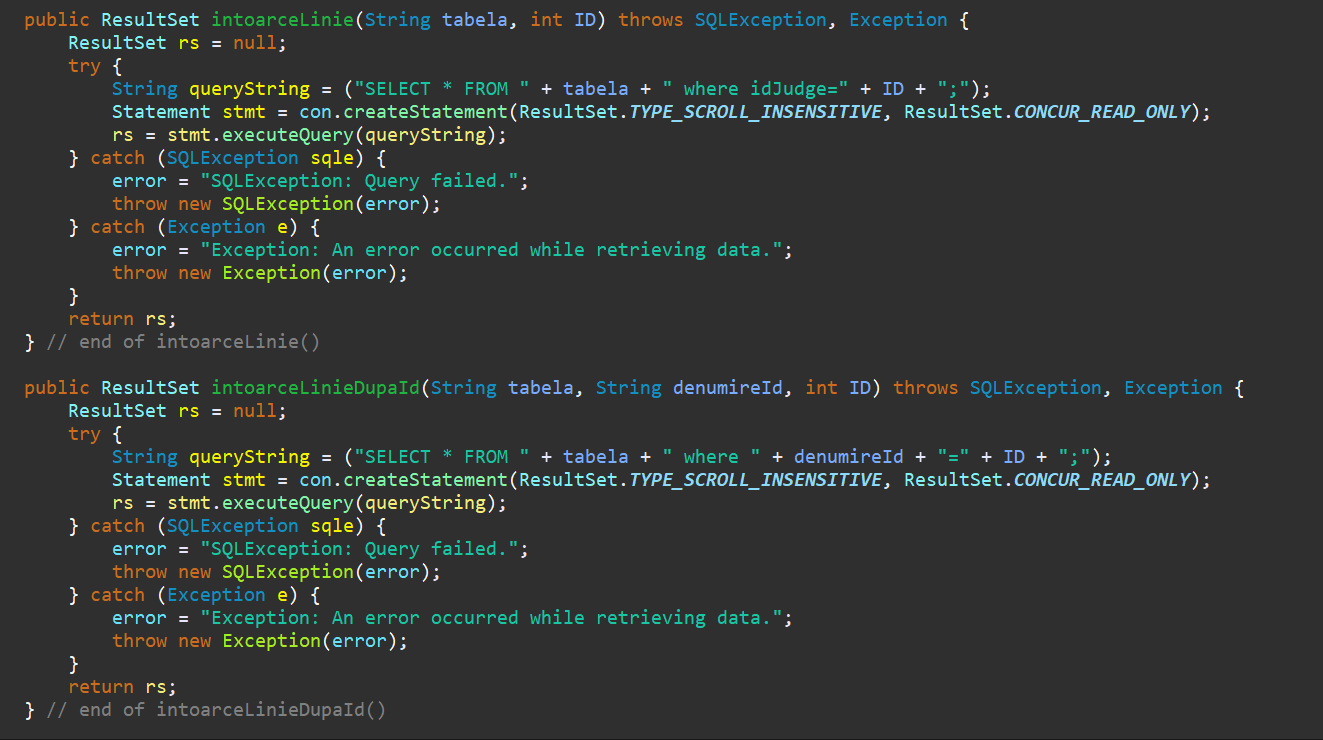
Metoda deleteTableData() este utilizată pentru a şterge înregistrări.



Metoda modificaTabela() permite actualizarea înregistrărilor dintr-o tabelă specificată, iar intoareLinie() și intoarceLinieDupaId() sunt folosite pentru a extrage rânduri specifice din tabele.

A screenshot of a computer program

Description automatically generated



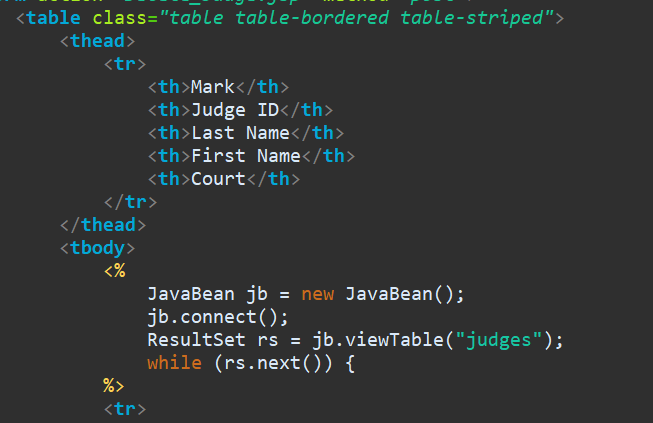
**Gestionarea excepţiilor**

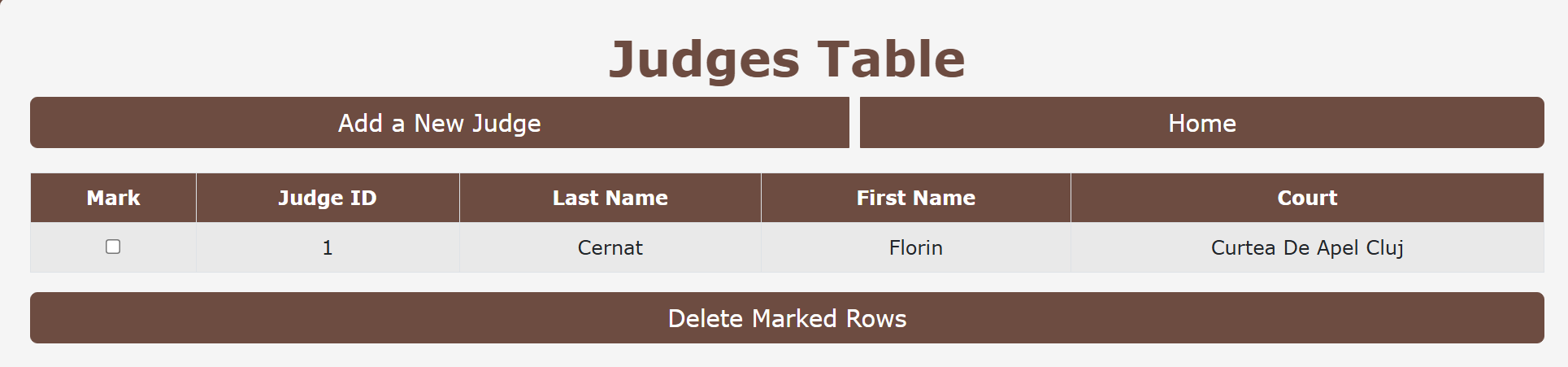
Fiecare metodă de conectare și manipulare a datelor este concepută pentru a gestiona excepții specifice, cum ar fi ClassNotFoundException, SQLEception și Exception. Aceasta asigură că erorile sunt tratate corespunzător și că aplicația poate răspunde adecvat la problemele de conectivitate sau la alte probleme care pot apărea.

**3.4 Implementarea JSP-urilor**

**Afişarea datelor**

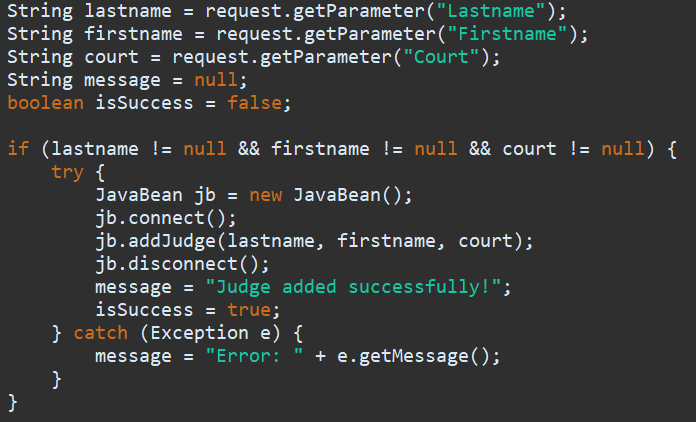
Următorul fragment de cod JSP este folosit pentru a afișa și interacționa cu datele referitoare la ”judges”, permițând utilizatorilor să selecteze și să execute acțiuni pe date specifice elevului.



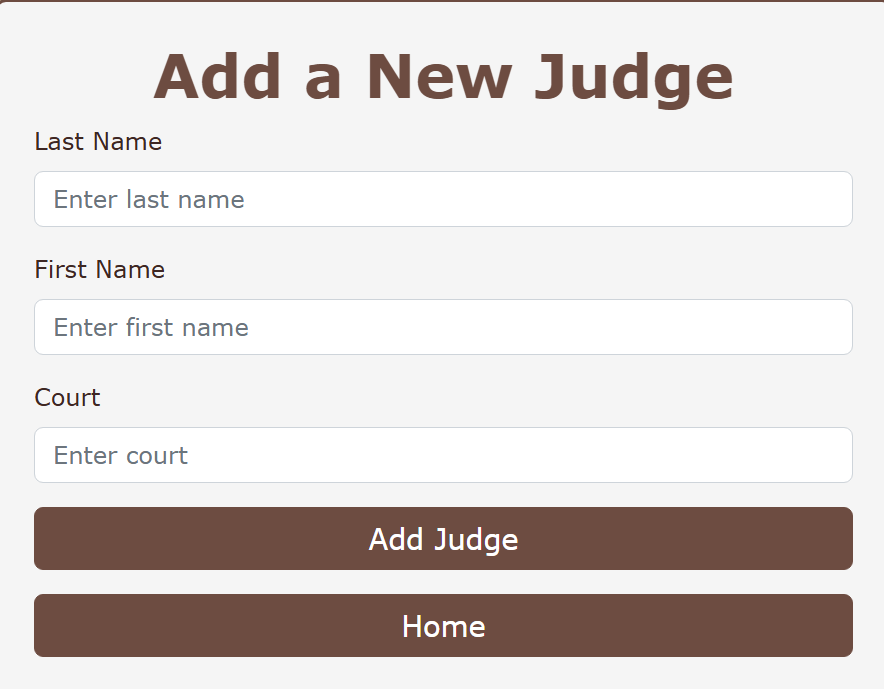


**Adăugarea datelor**

Adăugarea se realizează prin apăsarea butonului „Add” din pagina web. Acest fragment de cod JSP se ocupă cu preluarea parametrilor dintr-o cerere HTTP și utlizarea acestor parametri pentru a adăuga un nou client într-o bază de date.



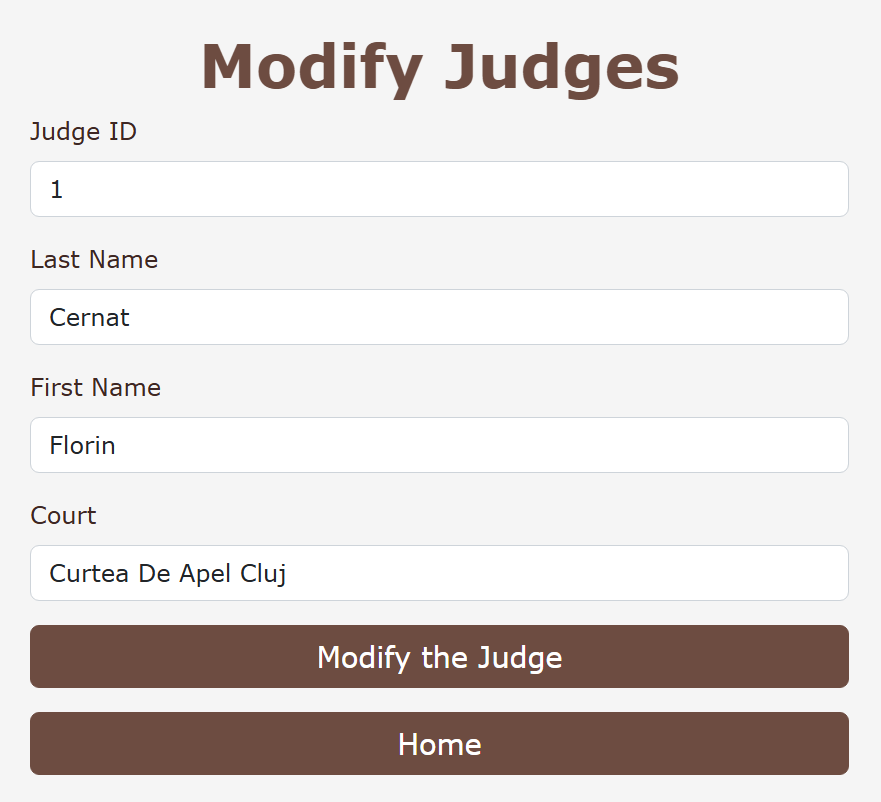




**Modificarea datelor**

Modificarea datelor se realizează prin selectarea checkbox-ului și apăsarea butonului de „Modify” din pagină. Acest fragment de cod JSP se ocupă cu preluarea parametrilor dintr-o cerere HTTP și utilizarea acestor parametri pentru a modifica clientul ales de utilizator din bază de date.





**Ștergerea datelor**

Ștergerea datelor se realizează prin selectarea checkbox-ului și apăsarea butonului de „Delete Marked Rows” din pagină.



**4. Concluzii**

Aplicația utilizează operațiunile de bază CRUD (creare, citire, actualizare, ștergere) pentru a oferi o modalitate simplă și eficientă de a administra clienții, tipurile de asigurări și contractele. Această funcționalitate o face o soluție practică și adaptată nevoilor din domeniul asigurărilor, ajutând la o gestionare optimă a datelor.  
Prin designul său simplu și intuitiv, aplicația permite utilizatorilor să acceseze și să organizeze rapid informațiile necesare, devenind un instrument indispensabil pentru agenții de asigurări.

**5. Bibliografie**

1. MySQL (https://www.mysql.com/)

2. Oracle Java Server Pages Technology (https://www.oracle.com/java/technologies/jspt.html)

3. Tomcat (https://tomcat.apache.org/)

4. FreeCodeCamp (https://www.freecodecamp.org/news/html-css-and-javascript-explained-for-beginners)