UNIVERSITATEA „POLITEHNICA” din BUCUREȘTI

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

**Proiect**

**Programarea Interfețelor pentru Baze de Date**

**(1)**

Catrinescu Teodora

431A

**Cerință**

Creați două aplicații care să conțină o bază de date creată în sistemul de gestiune al

bazelor de date MySQL și două interfețe pentru aceasta (baza de date este comună). La

crearea interfețelor se vor folosi două tehnologii la alegere (JSP, Hibernate, JPA, .NET,

PHP, Python+Django, Python+Flask etc.)

Am realizat o bază de date în MySQL, ea conținând 2 tabele: **poeti** și **carti**. Asocierea dintre ele este de tipul M:N (many-to-many), ceea ce presupune crearea unei tabele intermediare, pe care am numit-o **publicatii**.

Tabela **poeti** conține următoarele coloane:

●idpoet, de tip BIGINT, cheie primară

●nume, de tip VARCHAR

●prenume, de tip VARCHAR

●nationalitate, de tip VARCHAR

Tabela **carti** conține următoarele coloane:

●idcarte, de tip BIGINT, cheie primară

●titlu, de tip VARCHAR

●gen\_literar, de tip VARCHAR

●isbn, de tip VARCHAR

Tabela intermediară **publicatii** conține următoarele coloane:

●idpublicatie, de tip BIGINT, cheie primară

●idpoet, de tip BIGINT

●idcarte, de tip BIGINT

●data\_publicarii, de tip DATE

●editura, de tip VARCHAR

Datorită asocierii M:N, id\_poet și id\_carte sunt chei străine (FK) pentru tabela publicatii. Cheile primare (PK) corespunzătoare fiecărei tabele au fost setate cu următoarele proprietăți: not null (NN) și auto-increment (AI).

O cheie primară este formată dintr-una sau mai multe coloane ale căror valori identifică în mod unic toate liniile unei tabele.

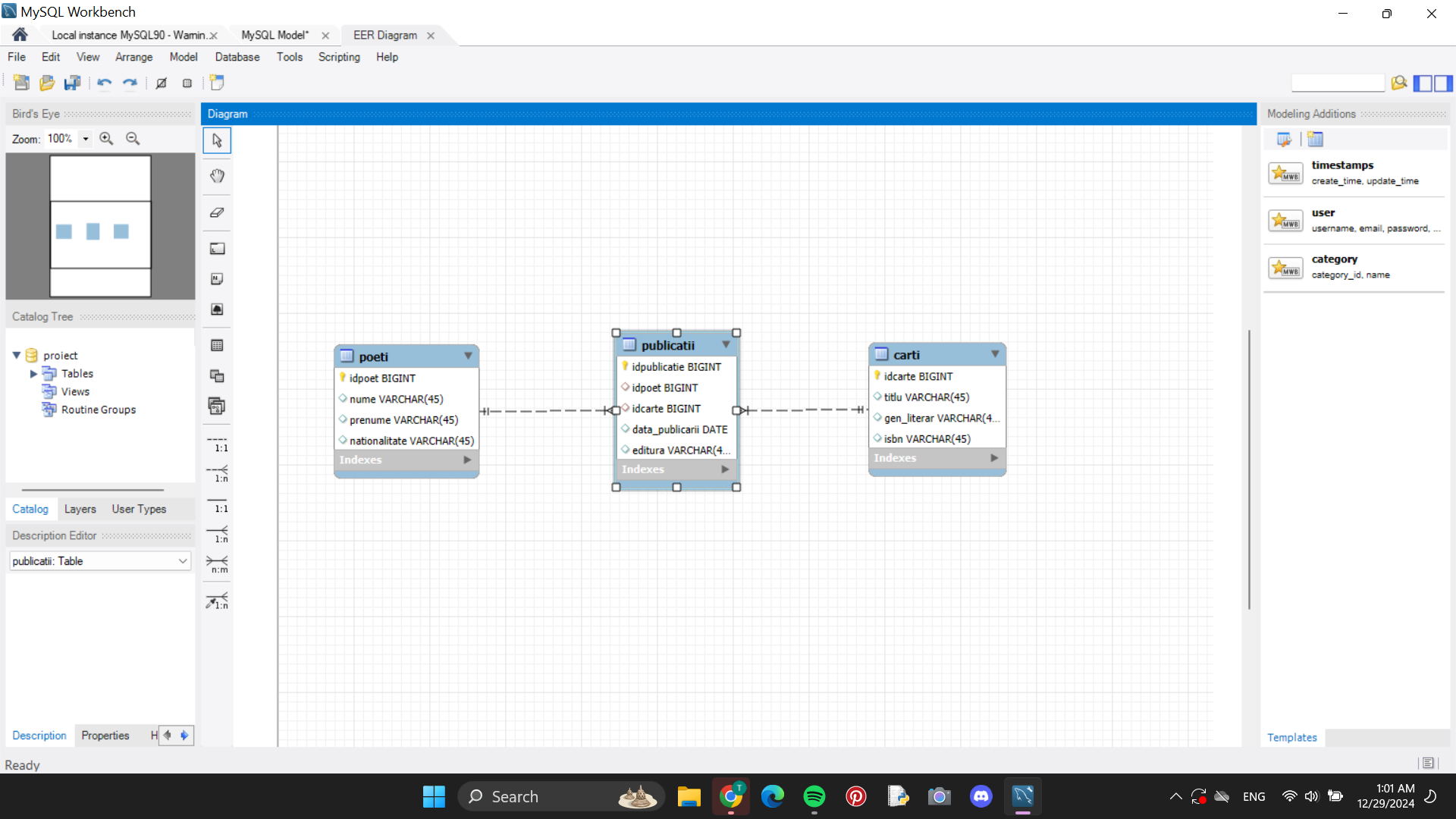


Fig. 1: diagrama asociata tabelelor

Pentru realizarea interfeței dintre utilizator și bază de date am ales tehnologia Java

Server Pages (JSP), tehnologie ce facilitează crearea paginilor Web de dezvoltatorii

software, pagini generate dinamic, bazate pe HTML, XML sau alte tipuri de documente.

Cu toate că JSP este similar cu PHP (Hypertext Preprocessor) și ASP (Active Server Pages),acesta utilizează limbajul de programare Java.

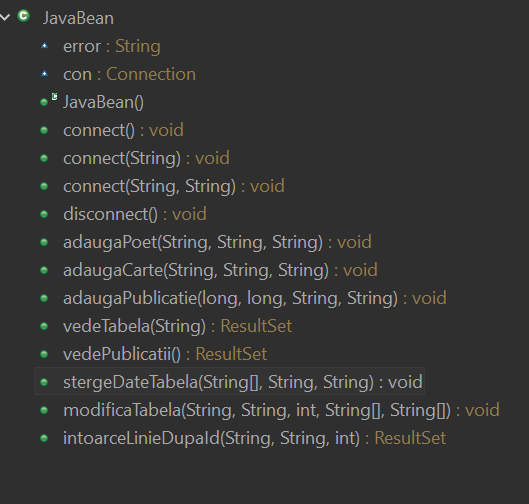


Fig. 2: functiile din JavaBean

Avem mai multe metode care ne permit sa facem toate operatiunile de care avem nevoie: vizualizare tabele (metodele vedeTabela si vedeDosar), inserare de date in tabele (metodele adaugaPoet, adaugaCarte si adaugaPublicatie), stergere date existente (stergeDateTabela), modificare date (modificaTabela) dar si metode ce ne permit intoarcerea unei linii folosind id-ul.

Pagina principala a interfetei (**index.html**) facuta in html permite accesul spre paginile tabelelor unde putem vizualiza datele, putem adauga date noi, putem modifica datele existente sau putem sterge ce date dorim.

Pagina principala a interfectei:



Fig. 3: pagina principala a interfetei

Din pagina principala, **index.html,** utilizatorul are posibilitatea de a intra pe 3 pagini aferente tabelelor “Poeti”, “Carti” si “Publicatii”,pe fiecare dintre aceste pagini poate vizualiza tabelele din baza de date, poate adauga, modifica si sterge date.In “Modifica Carti”, “Modifica Publicatii”, “Modifica Poeti” se pot edita linii si se pot adauga date noi.

Functionalitatea paginilor este similara pentru cele trei tabele, astfel : pentru a adauga o inregistrare noua in tabela (poeti, carti sau publicatii) utilizatorul trebuie sa apese pe butonul de modificare corespunzator. Se va deschide o pagina in care sunt afisate datele din tabelele corespunzatoare.

Tabela poeti:

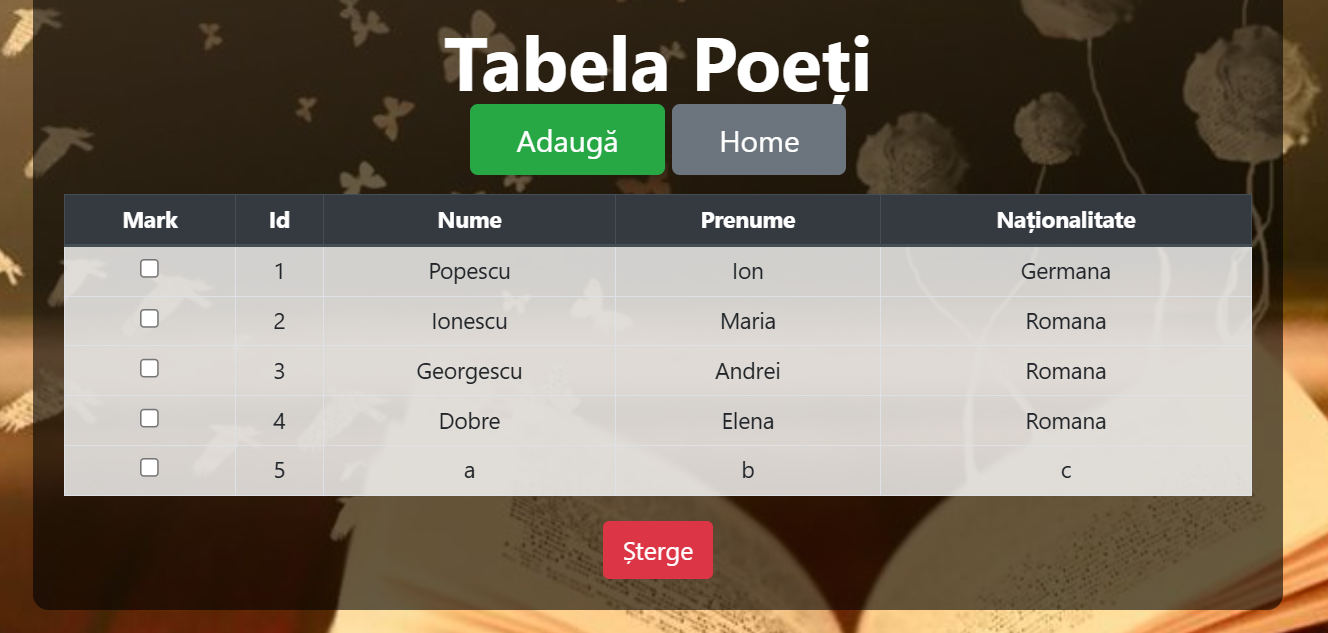


Fig.4: Tabela Poeti

Daca se doreste adaugarea unui nou poet, utilizatorul va da click pe adauga si se va deschide o pagina cu un formular unde utilizatorul poate complete datele necesare:



Fig.5: Adaugare in tabela poet

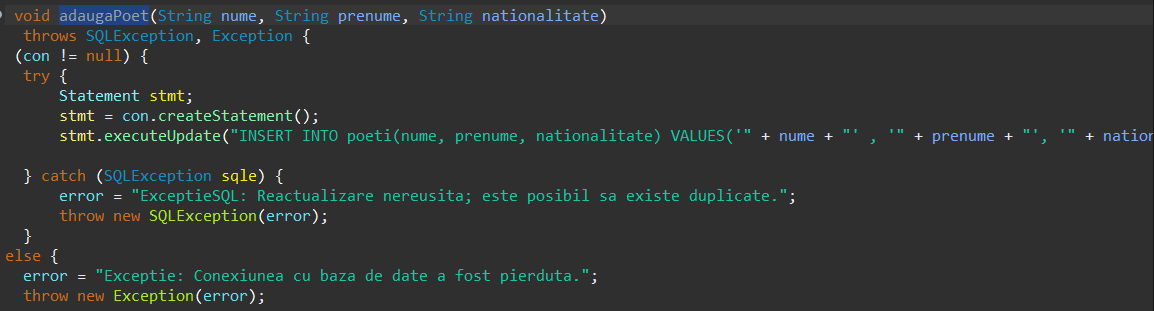


Fig. 6: codul corespunzator functiei adaugare

Se face un insert in tabela poeti cu toate campurile necesare (nume, prenume, nationalitate).

Instructiunea este inconjurata de un block try-catch pentru ca exista posibilitatea sa apara erori, de exemplu sa nu mai existe stocare pe server pentru a adauga in baza de date sau sa existe duplicate, sa existe alte erori SQL, etc



Fig. 7: Tabela Carti

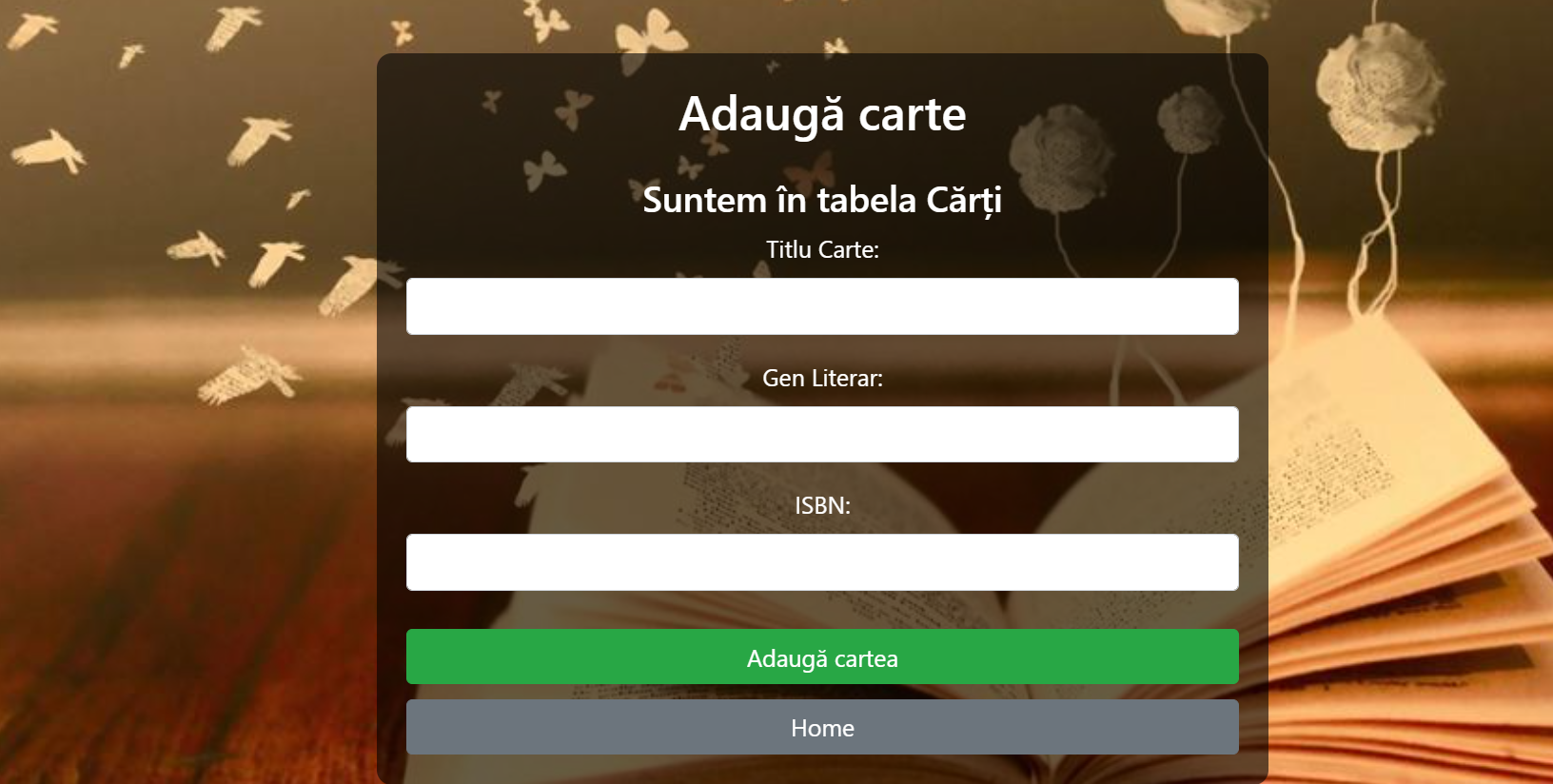


Fig. 8: Adaugare in tabela carti

Adaugarea in tabela carti este similara:

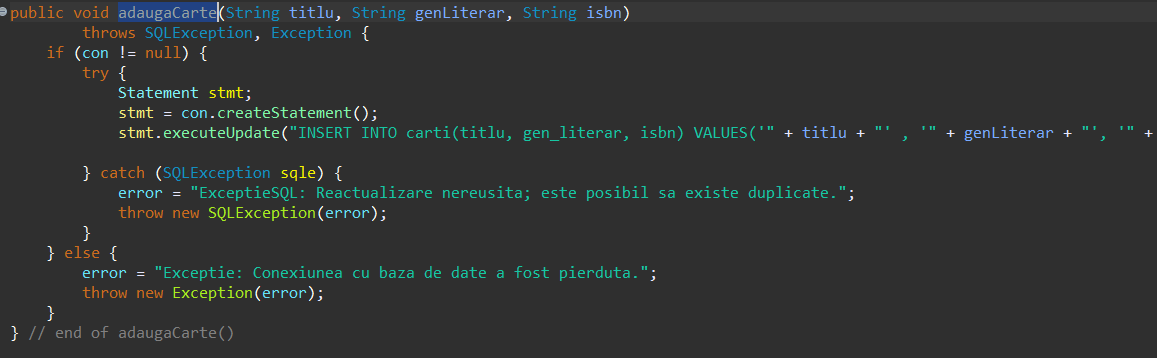


Fig. 9: codul corespunzator functiei adaugare

Se face un insert in tabela carti cu toate campurile necesare (titlu, gen literar,isbn).

Tabela de legatura publicatii:

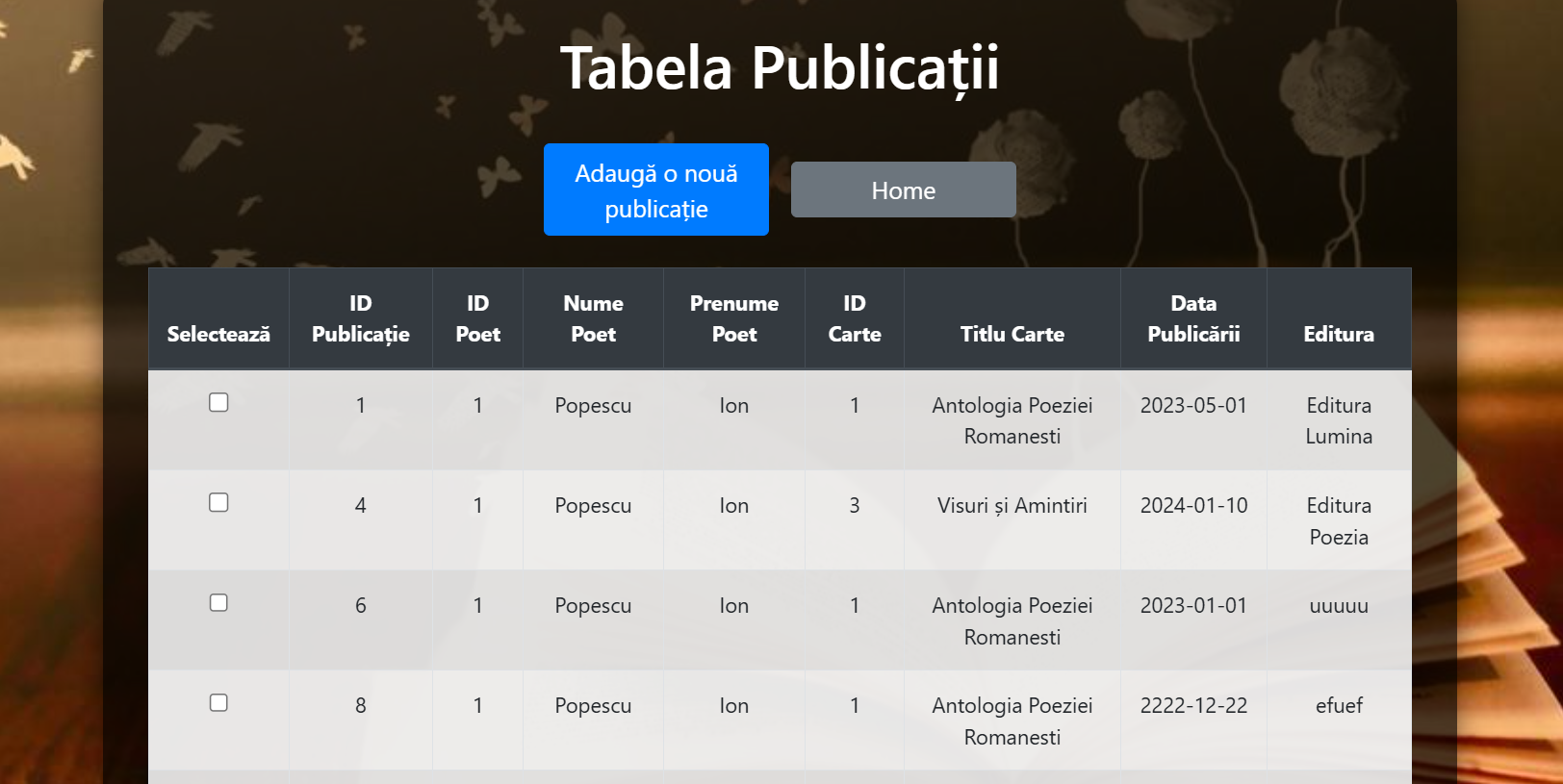


Fig. 10: Tabela Publicatii

Adaugarea in tabela publicatii este similara, dar difera putin datorita faptului ca tabela publicatii este tabela de legatura si trebuie sa selectam poetul si cartea aferenti unui publicatii dintr-o lista deja definita:



Fig. 11: Adaugare in tabela Publicatii

Codul aferent operatiunii de adaugare :

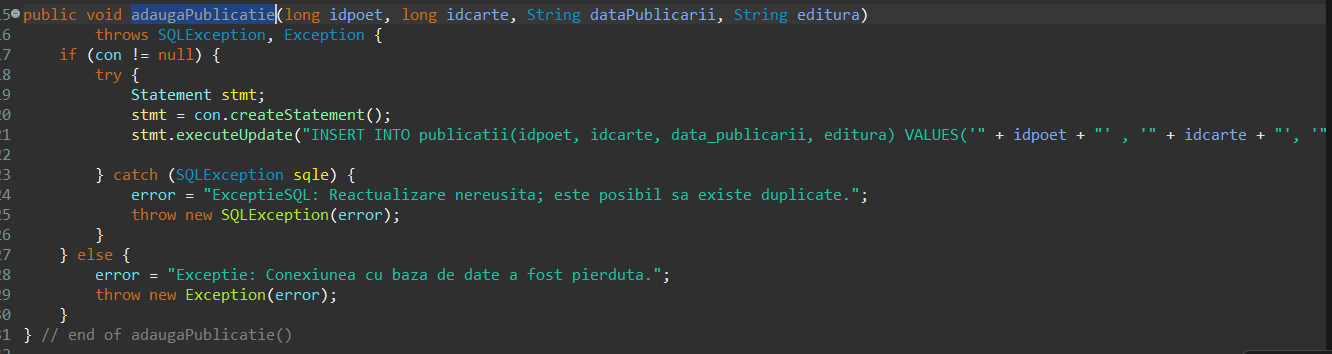


Fig. 12: codul corespunzator functiei adaugare

Similar cu tabela poeti si tabela carti se face tot un insert in baza de date cu campurile idpoet, idcarte, data\_publicarii, editura.

Dupa preluarea datelor va fi afisata o pagina cu mesajul care confirma adaugarea noii inregistrari : **“Datele au fost adaugate cu success ”** si 2 butoane, unul care trimite utilizatorul la pagina de start, altul care trimite utilizatorul la tabela respective unde a fost facuta adaugarea



Fig. 13: Confirmare de adaugare a datelor

Pentru a sterge campuri din tabele selectam linia dorita si apasam pe butonul rosu de stergere (“sterge publicatia”) :



Fig.14: Tabela publicatie cu butonul de stergere

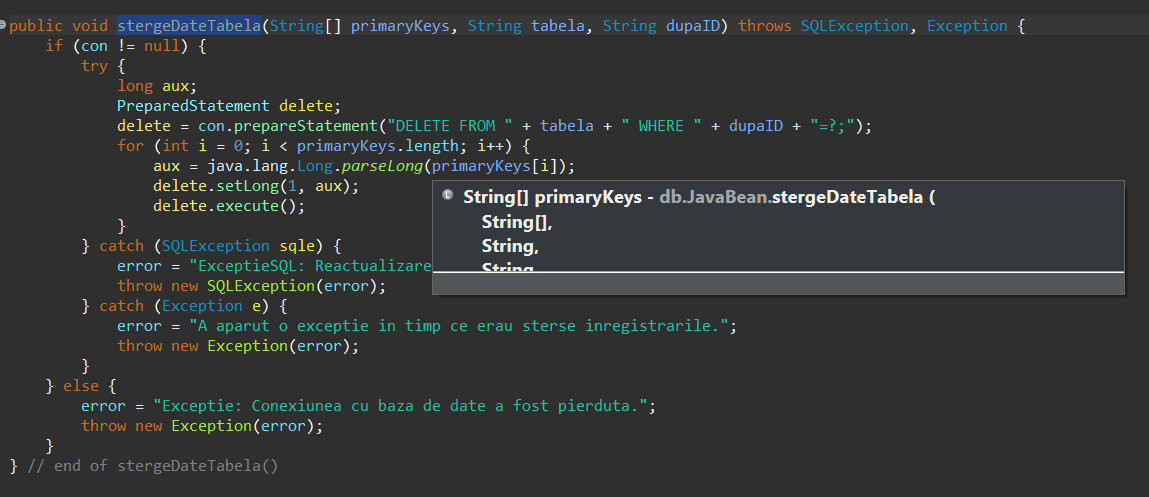


Fig. 15: codul corespunzator functiei de stergere

Dupa cum se poate vedea, se face un DELETE din tabela data, indentificand linia pe care dorim sa o stergem dupa id-ul dat.

In mod similar se procedeaza pentru campurile “poeti” si “carti” , iar la final va aparea pe ecran mesajul de confirmare a efectuarii operatiei si butonul de unde se poate accesa home.

Pentru modificarea unor campuri , se apasa in pagina principala pe una din tabele (“Modifica Carti”, “Modifica Publicatii”, “Modifica Poeti”), apoi ne va aparea o pagina unde putem sa selectam ce dorim sa modificam:

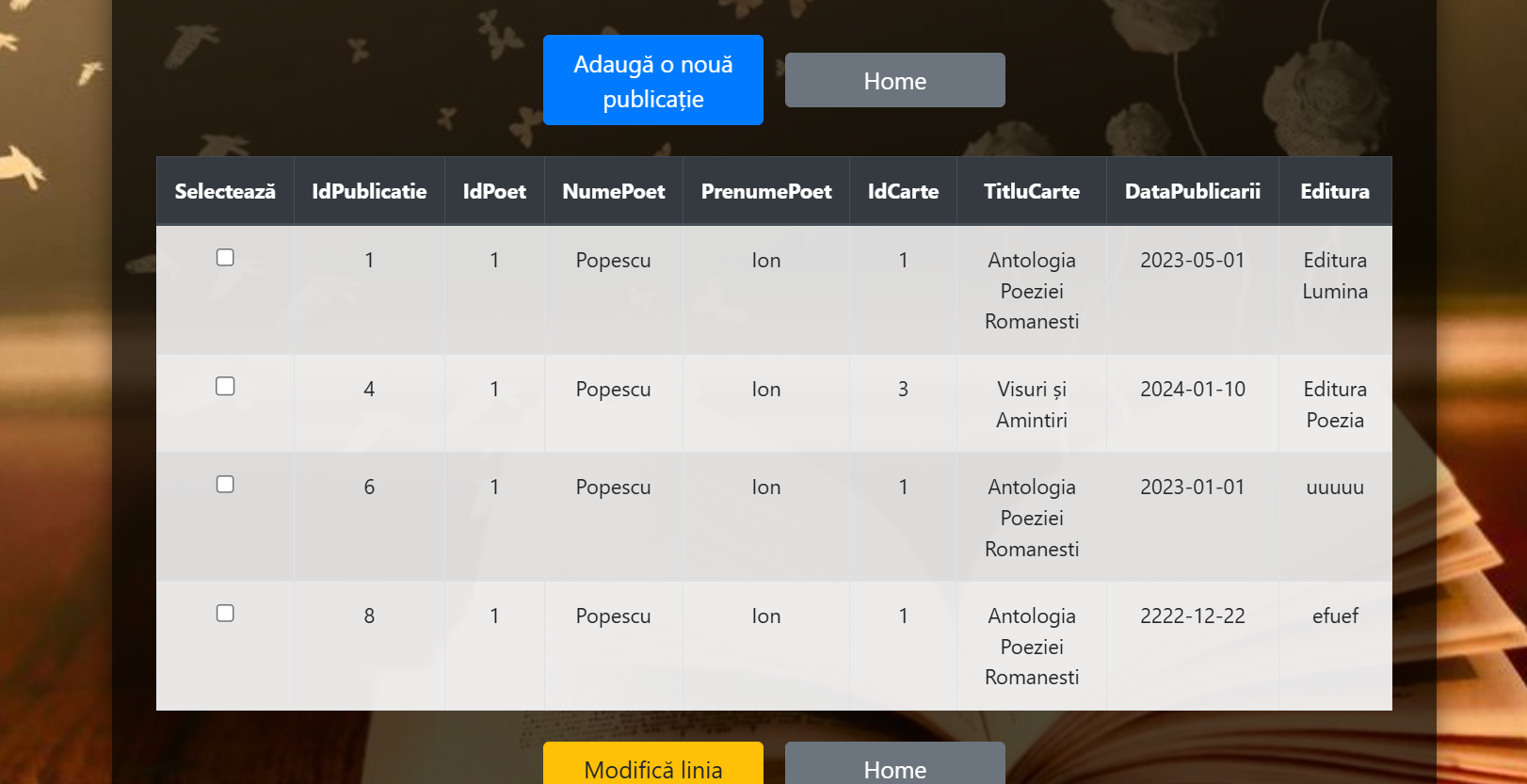


Fig.16: Pagina de modificare a publicatiei

Dupa selectarea liniei si apasare butonului Modificara linia in acest caz utilizatorii sunt redirectionati pe o pagina unde pot modifica datele publicatiei. Similar si pentru celelalte tabele.



Fig. 17: Pagina de modificare a unei linii din publicatie

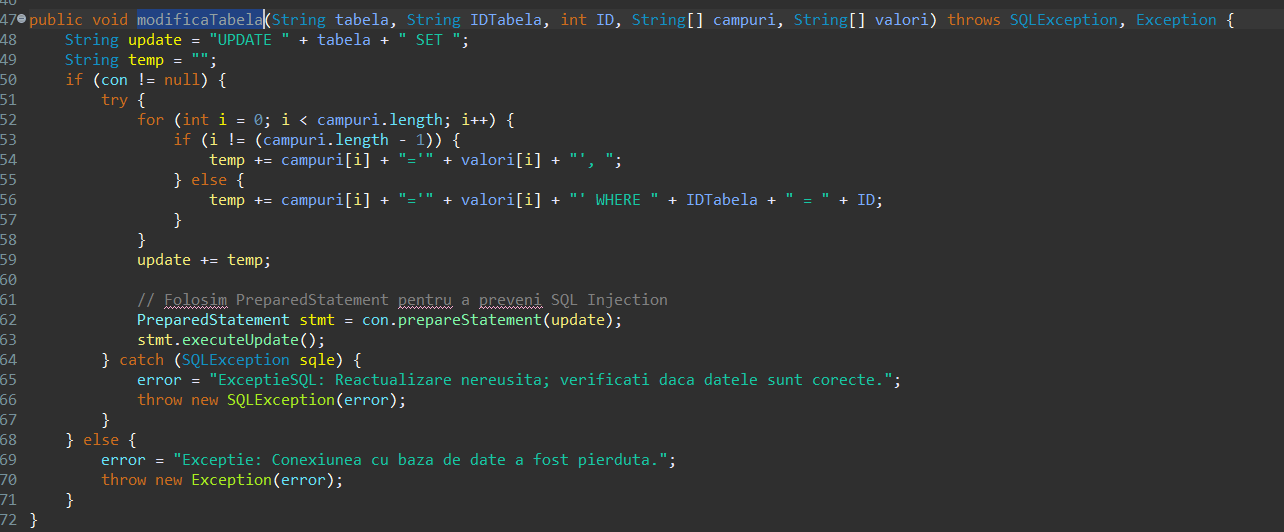


Fig. 18: Codul pentru modificare din tabele:

Metoda de modificare poate fi apelata pentru oricare din tabele. Variabila de tip string tabela parametrizeaza tabela, la fel si vectorii de tip string campuri si valori, prin acestia se pot modifica un numar variabil de campuri in functie de tabela data. Pentru asta se foloseste un for care parcurge tot vectorul cu campuri si in interiorul acestei bucle vom concatena campurile si valorile ce se doresc a fi modificate intr-un string cu numele temp. Acest string va fi folosit la instructiunea SQL de update pentru efectuarea modificarilor.

În concluzie, cursul ne-a învățat să putem crea o interfață prietenoasă care permite gestionarea datelor dintr-o bază de date MySQL. Folosind interfața creată, o instituție, precum o editură sau o bibliotecă, poate să gestioneze ușor informațiile despre autori și cărți. Acesta poate include adăugarea de noi autori, titluri de cărți și publicații, fără a necesita un angajat specializat pentru administrarea datelor. Oricare angajat existent poate să efectueze aceste operații prin intermediul interfeței.

Mai mult decât atât, datorită tehnologiilor JSP și MySQL, aplicația poate fi implementată pe un server, ceea ce permite accesul la aplicație din mai multe locații, de pe tot globul, atâta timp cât există o conexiune la internet. Astfel, nu mai este necesar ca un administrator de baze de date să se ocupe de gestionarea bazei de date folosind un program de management precum PhpMyAdmin, unde este necesar un nivel de cunoștințe SQL pentru a efectua operațiunile necesare.

Acest tip de bază de date va permite gestionarea eficientă a publicațiilor și relațiile între poeți și cărțile lor. Interfața va permite unui utilizator să adauge noi poeți, cărți și publicații, iar toate aceste date vor fi gestionate cu ușurință de orice angajat, fără a fi necesare cunoștințe tehnice avansate. Astfel, proiectul meu poate fi implementat și utilizat de orice persoană care are nevoie să administreze informațiile din această bază de date.

Bibliografie:

1. Curs de Programare a Interfetelor pentru Baze de Date (V. Pupezescu)
2. https://build.layoutit.com/