UNIVERSITATEA „POLITEHNICA” din BUCUREȘTI

**Facultatea de Electronică, Telecomunicații și Tehnologia Informației**

Proiect

Programarea Interfețelor pentru Baze de Date

(1)

Student: Nicolae Stefan Prof. coordonator: Grupa 434A Ș.l. Dr. Ing. Pupezescu Valentin

București 2024

# 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.)

# Introducere

Drept tema propusa, s-a abordat gestionarea filmelor, a categoriilor acestora, dar si a elementor comune, a MetaDatei. Pentru proiectare, de data aceasta, s-a ales folosirea tehnologiei JSP(Jakarta Server Pages).

Un sistem de gestiune al bazelor de date (SGBD) este o aplicație care gestionează toate aspectele ce țin de o bază de date.

Un SGBD are următoarele funcții:

* Asigură crearea bazelor de date;
* Asigură salvarea în siguranță a datelor;
* Conține un set de instrucțiuni minime pentru transpunerea în limbaj de programare a unui algoritm elementar;
* Conține un set de comenzi detaliate pentru gestionarea datelor salvate în baza de date;
* Asigură o interfață ce poate fi utilizată atât de programatori, cât și de utilizatorii neexperimentați;
* Conține un set de instrucțiuni ce permite programatorului să creeze o interfață cu utilizatorul.

Pentru realizarea acestui proiect a fost utilizat MySQL Workbench 8.2, un sistem de gestiune a bazelor de date relaționale ce este utilizat în principal pentru aplicațiile online. În prezent este cel mai popular SGBD open-source, reprezentând o componentă cheie a stivei LAMP (Linux, Apache, MySQL, PHP).

În cadrul acestei prime părți a proiectului a fost implementată ca interfață pentru baza de date creată tehnologia JSP (Jakarta Server Pages), o simplă dar puternică unealtă utilizată pe partea de server pentru a genera conținut HTML dinamic.

# Implementare

Baza de date a fost realizată cu ajutorul MySQL Workbench, fiind alcătuită din două tabele: *filme* și *categorii*. Pentru a realiza o asociere între aceste două tabele de tip M:N, a fost creată o tabelă intermediară numită *metadata*.

Structura coloanelor tabelelor:

**Tabela** *filme*

* idfilme (tip BIGINT, cheie primară, Not Null, Auto-Increment)
* nume (tip VARCHAR(80))
* durata (tip VARCHAR(45))
* rating (tip VARCHAR(45))

**Tabela** *categorii*

* idcategorii(tip BIGINT, cheie primară, Not Null, Auto-Increment)
* nume (tip VARCHAR(45))
* popularitate (tip VARCHAR(45))
* tara (tip VARCHAR(45))

**Tabela** *metadata*

* idmetadata (tip BIGINT, cheie primară, Not Null, Unsigned, Auto-Increment)
* idfilme (tip BIGINT)
* idcategorii (tip BIGINT)
* an\_de\_lansare (tip BIGINT)
* limba (tip VARCHAR(45))

Datorită asocierii M:N, *idfilme* și *idcategorii* sunt setate ca chei străine (foreign keys) în cadrul tabelei *metadata*. Cheile primare corespunzătoare fiecărei tabele au fost setate cu proprietățile Not Null (NN), Auto-Increment (AI) și Unsigned (UN). Diagrama aferentă tabelelor și structurii acestora este reprezentată în Fig. 1.

A screenshot of a computer program

Description automatically generated

Fig. 1 – Diagrama EER a tabelelor și conținutului acestora

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

A screen shot of a computer program

Description automatically generated

Fig. 2 – Diagrama UML aferentă clasei JavaBean din fișierul JavaBean.java

Fișierul *JavaBean.java*, ce reprezintă “motorul” care realizează operațiile pentru conectarea bazei de date și gestionarea tabelelor și a datelor, conține o singură clasă numită *JavaBean* în care se regăsesc metodele implementate de care avem nevoie: vizualizare tabele (metodele *vedeTabela* și *vedeMetadata*), inserare de date în tabele (metodele *adaugaFilm*, *adaugaCategorie* și *adaugaMetadata*), ștergere date existente (metodele *stergeDateTabela* și *stergeTabela*), modificare date (metoda *modificareTabela*), dar și metode ce ne permit returnarea unei linii dintr-o tabelă folosind ID-ul acesteia.

Pagina principală a interfeței *index.html*, realizată exclusiv cu limbajul de marcare HTML și limbajul descriptiv CSS, ne oferă acces la panoul de control al aplicației noastre Web, prin care putem adăugate date noi, putem modifica datele existente sau le putem șterge din tabela pe care dorim să operăm.

A screen shot of a computer

Description automatically generated

Fig. 3 – Pagina principală a interfeței

După cum se poate observa, utilizatorul are posibilitatea de a accesa pagina aferentă vizualizării și gestionării datelor pentru filme, categorii și metadata, stocate în tabelele cu aceleași nume.

Funcționalitatea paginilor este similară pentru operațiile aferente celor trei tabele: pentru a adăuga o nouă înregistrare în tabela dorită (*filme*, *categorii* sau *metadata*).

O nouă pagină se va deschide, ce va afișa datele din tabela dorită.

A screenshot of a computer screen

Description automatically generated

Fig. 4 – Pagina destinată vizualizării datelor filmelor (tabela *filme*)

Dacă se dorește adăugarea unei noi bănci, utilizatorul va apăsa pe butonul

“Adauga un nou film” îl va redirecționa către pagina ce conține formularul ce necesită a fi completat pentru noua intrare.

A white background with black lines

Description automatically generated

Fig. 5 – Pagina destinată adăugării unui nou film

Codul implementat aferent operației de adăugare a băncii este următorul:

A screen shot of a computer program

Description automatically generated

Implementarea codului pentru adăugarea unei noi categorii este aproximativ identică, fiind diferite doar numărul parametrilor primiți de funcție și numele acestora.

Pentru evitarea neprevăzutelor erori ce pot să apară, ca de ex. epuizarea spațiului de stocare pe serverul gazdă, instrucțiunile au fost înconjurate cu block-uri de tip try-catch, ce pot “arunca” excepții pentru o depanare mai ușoară.

Paginile destinate vizualizării metadatei sunt similare celei de adăugare a filmelor, existând mici diferențe în funcție de numărul și tipul datelor fiecărei entități.

A screenshot of a computer

Description automatically generated

Fig. 6 –Pagina destinată vizualizării datelor metadatei (tabela *metadata*)

A screenshot of a computer

Description automatically generated

Fig. 7 – Pagina destinată adăugării metadatei

Codul implementat aferent operațiilor de modificare, indiferent de ce se dorește a fi modificat, este următorul:

A screen shot of a computer program

Description automatically generated

În cazul modificării realizate cu succes a datelor, va fi notificat printr-un mesaj de confirmare. Utilizatorul va fi redirecționat automat către pagina de vizualizare a datelor.

A screenshot of a computer

Description automatically generated

Fig. 13 – Mesaj de confirmare pentru operația de modificare realizată cu succes

În schimb, dacă utilizatorul dorește să șteargă date, acesta va trebui să bifeze una sau mai multe intrări prin selectarea checkbox-urilor intrărilor ce se doresc a fi șterse, urmând apoi să apese butonul “Sterge Liniile Marcate”, după caz.

Dacă operația de ștergere a fost realizată cu succes, utilizatorul va fi redirecționat către pagina de vizualizare a datelor, fără a mai primi un mesaj de confirmare.

Codul ce rulează pentru operațiile de ștergere este următorul:

A screen shot of a computer code

Description automatically generated

Metoda implementată mai sus pentru ștergerea datelor este apelată pentru oricare dintre tabele. Variabila și vectorii de tip String parametrizează tabela și câmpurile, putându-se astfel modifica un număr variabil de valori în funcție de tabela dată ca parametru.

# Concluzii

Cursul de *Programarea Interfețelor pentru Baze de Date* ne-a oferit o percepție mai amplă asupra conceptului de baze de date și o înțelegere solidă a conceptelor și tehnicilor necesare pentru a gestiona și utiliza bazele de date eficient. Am avut oportunitatea de a învăța cum putem crea și administra o bază de date folosind SGBD-ul MySQL, precum și modul prin care o putem conecta la un website folosind tehnologia Jakarta Server Pages.

Prin interfața creată, un administrator de date poate gestiona baza de date a filmelor cat si a catorva din caracteristicile lor. Interfața a fost creată pentru a fi cât mai simpla și curată dar cu posibilitati de remodelare,aceasta avand o scalabilitate in functie de nevoi, oferind astfel posibilitatea și unei persoane mai puțin instruite de a opera datele ce se propun a fi stocate.

În ceea ce privește tehnologiile MySQL și JSP, putem spune că reprezintă un mijloc prin care se poate crea o astfel de interfață într-o manieră elegantă, avânt totodată și avantajul unei implementări ușoare și exacte.

***Bibliografie***

1. Prof. Pupezescu V., *Curs de Programarea Interfețelor pentru Baze de Date 2023-2024*
2. \*\*\*, <https://ro.wikipedia.org/wiki/MySQL>
3. \*\*\*, <https://en.wikipedia.org/wiki/Jakarta_Server_Pages>