UNIVERSITATEA „POLITEHNICA” din BUCUREȘTI

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

Proiect

Programarea Interfețelor pentru Baze de Date

(2)

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 unei aplicatii, pentru a facilitata construirea interfetei bazei de date.

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 celei de a doua părți a acestui proiect, am abordat crearea unei aplicații ce rulează pe sistemul de operare Windows, aplicație realizată prin intermediul platformei open-source JavaFX, ce este conectată la baza de date și realizează operații asupra sa.

# 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 platforma JavaFX, tehnologie ce facilitează crearea și livrarea de aplicații desktop și aplicații Web, ce pot rula pe o mare varietate de dispozitive.[[3]](#_bookmark0) Partea back-end a fost implementată cu ajutorul limbajului de programare Java, îmbinând partea de front-end concepută că ajutorul limbajului de marcare FXML și stilizată cu elemente puse la dispoziție de CSS (Cascading Style Sheets). Interfața grafică și interactivă a fost facilitată de instrumentul Scene Builder, ce ușurează proiectarea interfețelor grafice ale aplicațiilor JavaFX.[[5]](#_bookmark1)

A screenshot of a computer program

Description automatically generated

Fig. 2 – Diagrama UML aferentă clasei DBoperations din fișierul DBOperations.java

Fișierul *DBOperations.java*, ce reprezintă “motorul” care realizează operațiile pentru conectarea bazei de date și gestionarea tabelelor și a datelor, conține o singură clasă numită *DBOperations* 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 *modificaTabela*), dar și metode ce ne permit returnarea unei linii dintr-o tabelă folosind ID-ul acesteia.

Odată executată aplicația, utilizatorul va fi întâmpinat de fereastra principală în care se vor regăsi paginile (tab-urile) aferente vizualizării datelor dorite din baza de date. Totodată, tot în această fereastră se regăsesc butoanele corespunzătoare operațiilor ce se pot realiza asupra tabelelor. Inițial datele nu vor fi încărcate în tabelele pentru vizualizarea acestora, însă odată apăsate butoanele “Incarca Filme”, “Incarca Categorii”, respectiv “Incarca Metadata”, acestea vor fi încărcate și afișate.

A screenshot of a computer

Description automatically generated

Fig. 3 – Fereastra principală a aplicației, conținând tab-urile aferente tabelelor bazei de date

La fel ca la toate ferestrele destinate diverselor operații implementate, fereastra principală beneficiază de un “controller”, mai exact fișier *PIBDPROJController.java* ce controlează și gestionează comportamentul elementelor ferestrei descrise în fișierul *PIBDPROJ.fxml*. Metodele ce se regăsesc în controller fac posibilă legătura dintre interacțiunile uilizatorului, elementele grafice și operațiile implementate.

Codul pentru preluarea datelor din tabela dorită, regăsit în fișierul *DBoperations.java* este următorul:

**A screen shot of a computer code

Description automatically generated**

Această funcție este apelată din controllerele ferestrelor, datele putând fi ulterior afișate în tabelele și elementele grafice aferente ferestrelor.

Codul aferent legăturii dintre comportamentul unui buton de încărcare a datelor și preluării acestora din baza de date, cu ajutorul metodei menționate mai sus, aflat în fișierul *PIBDPROJController.java*, este umătorul:

A computer screen shot of text

Description automatically generated

Tabelul grafic în care sunt afișate datele ce se regăsesc în tabela din baza de date, permite selecția unei intrări ce se dorește a fi modificată sau ștearsă. Pentru aceste două operații, au fost adăugate două butoane ce implementează aceste operațiuni.

În cazul ștergerii unei intrări, va trebui selectat în tabelul rândul corespunzător intrării din baza de date ce se dorește a fi șters, apoi apăsat butonul “Sterge Filme”, “Sterge Categorie” sau “Sterge Metadata”, după caz.

Codul ce stă în spatele acestei operațiuni este următorul:

*DBOperations.java*:

**A screen shot of a computer program

Description automatically generated**

*ApplicationController.java*:

A screen shot of a computer

Description automatically generated

În cazul modificării unei intrări, va trebui selectat în tabelul rândul corespunzător intrării din baza de date ce se dorește a fi șters, apoi apăsat butonul “Modifica Filmul”, “Modifica Categorie” sau “Modifica Metadata”, după caz. Aceasta implementare nu exista momentan in codul aplicatiei.

Revenind la fereastra principală a aplicației, pe lângă operațiile de ștergere și modificare regăsim și butonul aferent operațiunii de adăugare. La fel ca în cazul modificării unei intrări, odată apăsat butonul “Adauga Filme”, “Adauga Categorie”, “Adauga Metadata” după caz, o nouă fereastră va fi deschisă pentru ca utilizatorul să selecteze și completeze în câmpurile corespunzătoare, datele noii intrări. Ferestrele de adăugare sunt descrise grafic în fișierele *AdaugaFilme.fxml*,*AdaugaCategorii.fxml,*respectiv *AdaugaMetadata.fxml* ce sunt legate la controllerul PIBDPROJController.java.

Un exemplu pentru operația de adăugare este prezentat în continuare.

A screenshot of a computer

Description automatically generated

Fig. 5 – Fereastra dedicată adăugării unui nou film

*AdaugaFilme.fxml*:

A screenshot of a computer program

Description automatically generated

*PIBDPROJController.java*(pentru adaugare/stergere de Metadata):

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A computer screen shot of text

Description automatically generated

A screen shot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

De amintit faptul că butoanele ce instanțiază, creează și deschid noile ferestre aferente operațiilor de adăugare și modificare apelează funcțiile implementate în *PIBDPROJController.java.*

Ca exemplu putem lua implementarea butonului “Adauga Metadata” căruia i-a fost scris codul:

A screen shot of a computer

Description automatically generated

# Concluzii

În această parte a proiectului, în care au fost aplicate conceptele programării orientate obiect si prin intermediul platformei JavaFx, am conceput o aplicație ce rulează pe sistemul de operare Windows, cu ajutorul căreia un administrator de date poate gestiona o baza de date ce stocheaza nenumarate filme cat si cateva din caracteristicile lor.

Utilizarea JavaFX și a instrumentului SceneBuilder a facilitat crearea unei interfețe grafice atractive, acestea reprezentând o soluție viabilă pentru dezvoltarea de aplicații de tip GUI (Graphic User Interface), cu un aspect profesional. Conectarea aplicației realizate la baza de date permite stocarea și accesarea informațiilor într-un mod organizat, eficient și intuitiv. Astfel că acest proiect demonstrează capacitatea de a combina tehnologii diferite pentru a crea aplicații robuste și scalabile într-o manieră elegantă, ce poate fi la îndemâna oricărui tip de utilizator.

***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/JavaFX>
4. \*\*\*, <https://fr.wikipedia.org/wiki/Scene_Builder>
5. \*\*\*,<https://stackoverflow.com>