***Universitatea „POLITEHNICA” Bucuresti***

**FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII ȘI TEHNOLOGIA INFORMAȚIEI**

*Proiect*

***Programarea Interfețelor pentru Baze de Date***

***(JSP)***

*Mitrea Bogdan-Gabriel*

*Grupa 432Da*

*Prof. coordonator: Ș.l. Dr. Ing. Pupezescu Valentin*

*2023 – 2024*

**Cerință**

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 etc.).

**Introducere**

Am abordat tema gestionării magazinelor, a clienților și a tranzacțiilor. Am dezvoltat o interfață Web care interacționează cu o bază de date, stocând toate informațiile necesare într-un mod eficient. În această etapă a proiectului, am implementat funcționalități precum vizualizarea, adăugarea, modificarea și ștergerea datelor într-un mod elegant și intuitiv.

Un Sistem de Gestiune al Bazelor de Date (SGBD) reprezintă o aplicație sofisticată, responsabilă cu administrarea tuturor aspectelor legate de o bază de date.

Printre funcțiile esențiale ale unui SGBD se numără:

* facilitarea creării bazelor de date
* asigurarea securizată a stocării datelor
* traducerea algoritmilor în limbaj de programare
* furnizarea de comenzi detaliate pentru gestionarea datelor
* oferirea unei interfețe accesibile atât pentru programatori, cât și pentru utilizatorii neexperimentați.

În ceea ce privește infrastructura proiectului, s-a optat pentru MySQL Workbench 8.0, un sistem de gestionare a bazelor de date relaționale recunoscut, în principal, în contextul aplicațiilor online. Acesta se impune ca fiind cel mai popular SGBD open-source în prezent, reprezentând o componentă esențială a stivei LAMP (Linux, Apache, MySQL, PHP).

În prima parte a proiectului, am ales tehnologia JSP (Java Server Pages) pentru a implementa interfața cu baza de date. JSP este o unealtă simplă, dar puternică, utilizată pe partea de server pentru a genera conținut HTML dinamic. Această decizie a contribuit la crearea unei interfețe eficiente și ușor de utilizat pentru utilizatori.

**Implementare**

Implementarea bazei de date a fost realizată prin intermediul MySQL Workbench, constând în două entități principale, anume tabelele "clients" și "shops". Pentru a gestiona relația de tip M:N între acestea, a fost concepută o entitate intermediară denumită "transaction".

**Structura coloanelor tabelelor:**

***Tabela clients***

* idClient (tip INT, cheie primară, Not Null, Unsigned, Auto-Increment)
* Nume (tip VARCHAR(45))
* Prenume (tip VARCHAR(45))
* Email (tip VARCHAR(45))
* Phone (tip VARCHAR(20))

***Tabela shops***

* idShop (tip INT, cheie primară, Not Null, Unsigned, Auto-Increment)
* NumeMagazin (tip VARCHAR(45))
* Localitate (tip VARCHAR(45))
* ContactEmail (tip VARCHAR(45))
* ContactPhone (tip VARCHAR(20))

***Tabela transaction***

* idTranzactie (tip INT, cheie primară, Not Null, Unsigned, Auto-Increment)
* idShop (tip INT)
* idClient (tip INT)
* PurchaseDate (tip DATE)
* TotalSpent (tip FLOAT)

Datorită asocierii de tip M:N, coloanele idClient și idShop au fost designate ca chei străine (foreign keys) în cadrul tabelei "transaction". Cheile primare corespunzătoare fiecărei tabele au fost configurate cu proprietățile Not Null (NN), Auto-Increment (AI), și Unsigned (UN). Diagrama aferentă tabelelor și structurii acestora este reprezentată în Fig. 1.

A diagram of a blockchain

Description automatically generated

Fig. 1 - Diagrama EER a tabelelor și conținutului acestora

În vederea implementării interfeței între utilizator și baza de date, am optat pentru tehnologia Java Server Pages (JSP). Această tehnologie se distinge prin facilitarea dezvoltării paginilor web generate dinamic de către dezvoltatorii software, fundamentate pe limbajele HTML, XML sau alte tipuri de documente.

A screenshot of a computer program

Description automatically generated

Fig. 2 – Fisierul JavaBean.java

**Fișierul JavaBean.java** funcționeaza ca un motor, fiind responsabil pentru efectuarea operațiilor de conectare la baza de date și gestionare a tabelelor și a datelor asociate. Acesta include o singură clasă denumită JavaBean, în care sunt implementate metodele necesare pentru diverse operațiuni, cum ar fi:

* Vizualizarea tabelelor (metodele vedeTabela și vedeContract).
* Inserarea de date în tabele (metodele adaugaShop, adaugaClient și adaugaTranzactie).
* Ștergerea datelor existente (metodele stergeDateTabela și stergeTabela).
* Modificarea datelor (metoda modificareTabela).
* Metode ce permit returnarea unei linii dintr-o tabelă utilizând ID-ul acesteia.

Această clasă JavaBean constituie nucleul funcțional al aplicației, furnizând setul de operațiuni esențiale pentru interacțiunea cu baza de date și manipularea datelor într-un mod eficient și organizat.

**Interfata**

Prima pagină a interfeței *index.html*, dezvoltată exclusiv folosind limbajul de marcare HTML și limbajul descriptiv CSS, oferă acces la panoul de control al aplicației web. Acest panou permite vizualizarea, adăugarea, modificarea și ștergerea datelor din tabela pe care dorim să lucrăm.

**A screenshot of a computer

Description automatically generated**

Fig. 3 – Pagina principala a interfetei

In aceasta pagina, utilizatorul beneficiază de posibilitatea de a accesa pagina corespunzătoare pentru vizualizarea, adaugarea și modificarea datelor asociate magazinelor, clienților și tranzactiilor, stocate în tabelele cu aceleași nume.

In cele ce urmeaza, funcționalitatea paginilor este similară în ceea ce privește operațiile legate de cele trei tabele: pentru a vizualiza si a adauga o nouă înregistrare în tabela dorită (Clienti, Magazine sau Tranzactii), utilizatorul va apasa pe unul din primele trei butoane. Pentru a modifica oricare set de date din fiecare tabela, utilizatorul navigheaza pe unul din ultimele trei butoane.

A screenshot of a computer

Description automatically generated

Fig. 4 – Tabela Clienti

In tabela „Clienti”, avem o optiune de a adauga un nou client, urmata de optiunea Home, care duce inapoi catre pagina principala. Folosindu-ne de casutele din stanga, putem selecta una sau mai multe linii, ca apoi sa apasam pe butonul „Sterge liniile marcate”, pentru a putea elimina acele intrari.

A screenshot of a login form

Description automatically generated

Fig. 5 – Adaugare clienti

In aceasta fereastra putem insera un nou client in tabela. Dupa ce am introdus campurile corespunzatoare, se apasa pe campul „Adauga clientul” pentru a confirma adaugarea. De asemenea, utilizatorul se poate intoarce inapoi la Home.

Codul implementat aferent operației de adăugare a unui client este următorul:

public void adaugaClient(String Nume, String Prenume, String Email, String Phone)

throws SQLException, Exception {

if (con != null) {

try {

// creaza un "prepared SQL statement"

Statement stmt;

stmt = con.createStatement();

stmt.executeUpdate("insert into clients(Nume, Prenume, Email, Phone) values('" + Nume

+ "' , '" + Prenume + "', '" + Email + "', '" + Phone + "');");

} catch (SQLException sqle) {

error = "ExceptieSQL: Reactualizare nereusita; este posibil sa existe duplicate.";

throw new SQLException(error);

}

} else {

error = "Exceptie: Conexiunea cu baza de date a fost pierduta.";

throw new Exception(error);

}

}

Implementarea codului pentru adăugarea unui nou magazin este identică, fiind diferite doar numărul parametrilor primiți de funcție și numele acestora.

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

A screenshot of a computer

Description automatically generated

Fig. 6 – Tabela Magazine

A screenshot of a computer

Description automatically generated

Fig. 7 – Adaugare magazine

Tabela Tranzactii face legatura intre cele doua tabele anterioare, si vine cu doua noi atribute: data in care a fost facuta tranzactia (PurchaseDate), si totalul tranzactiei respective (TotalSpent).

A screenshot of a computer

Description automatically generated

Fig. 8 – Tabela Tranzactii

A screenshot of a computer

Description automatically generated

Fig. 9 – Adaugare tranzactii

În cazul în care se doresc a fi aduse modificări asupra clientilor, magazinelor sau a contractelor, se accesează pagina corespunzătoare modificării datelor, prin accesarea butoanelor “Modifica Clienti”, “Modifica Magazine” sau “Modifica Tranzactii”.

**A screenshot of a computer

Description automatically generated**

Fig. 10 – Modifica Clienti

Similar cu tabela clientilor, aceasta fereastra nu aduce decat posibilitatea modificarii clientilor, prin butonul de jos, doar dupa ce s-a selectat una din linii de modificat.

La fel vor arata si celelalte doua taburi aferente modificarii magazinelor si a tranzactiilor:

A screenshot of a computer

Description automatically generated

Fig. 11 – Modifica Magazine

A screenshot of a computer

Description automatically generated

Fig. 12 – Modifica Tranzactii

Am lucrat si la aspectul paginii, micsorand marimea fontului astfel incat tabelul sa incapa pe toata pagina.

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

public void modificaTabela(String tabela, String IDTabela, int ID, String[] campuri, String[]

valori) throws SQLException, Exception {

String update = "update " + tabela + " set ";

String temp = "";

if (con != null) {

try {

for (int i = 0; i < campuri.length; i++) {

if (i != (campuri.length - 1)) {

temp = temp + campuri[i] + "='" + valori[i] + "', ";

} else {

temp = temp + campuri[i] + "='" + valori[i] + "' where " + IDTabela + " = '" + ID + "';";

}

}

update = update + temp;

// creaza un "prepared SQL statement"

Statement stmt;

stmt = con.createStatement();

stmt.executeUpdate(update);

} catch (SQLException sqle) {

error = "ExceptieSQL: Reactualizare nereusita; este posibil sa existe duplicate.";

throw new SQLException(error);

}

} else {

error = "Exceptie: Conexiunea cu baza de date a fost pierduta.";

throw new Exception(error);

}

}

Dupa ce s-a selectat linia care se vrea a fi modificata, intram in fereasta urmatoare, cu modificarea propriu-zisa, unde utilizatorul poate schimba datele liniei selectate in meniul anterior.

**A screenshot of a computer

Description automatically generated**

Fig. 13 – Meniu modificare clienti

Dupa ce s-au efectuat modificarile cu succes, se primeste un mesaj de confirmare:

A screenshot of a computer

Description automatically generated

Fig. 14 – Confirmare modificare client

Similare sunt si ferestrele de modificari si pentru celelalte doua tabele.

**Concluzii**

Cursul de *Programarea Interfețelor pentru Baze de Date* a contribuit semnificativ la extinderea perspectivelor asupra conceptului de baze de date și la dobândirea unei înțelegeri solide a conceptelor și tehnicilor esențiale pentru gestionarea și utilizarea eficientă a acestora. Am avut oportunitatea de a învăța cum să creăm și să administrăm o bază de date folosind Sistemul de Gestiune a Bazelor de Date (SGBD) MySQL, precum și modalitățile de conectare la un website utilizând tehnologia Java Server Pages(JSP).

Prin intermediul interfeței dezvoltate, un administrator de date dispune de capacitatea de a gestiona fluxul clienților și comportamentul acestora în ceea ce privește acțiunile permise asupra contului lor de economii. Interfața a fost concepută într-un mod intuitiv și clar, facilitând astfel operațiunile și pentru persoanele mai puțin familiarizate cu manipularea datelor propuse pentru stocare.

În ceea ce privește tehnologiile MySQL și JSP, acestea au demonstrat a fi instrumente eficiente pentru crearea unei astfel de interfețe într-un mod elegant, oferind avantajul unei implementări ușoare și precise.

**Bibliografie**

[1] Prof. Pupezescu V., Curs de Programarea Interfețelor pentru Baze de Date 2023-2024

[2] \*\*\*, <https://www.w3schools.com/> , accesat la data de 01.02.2024

[3] \*\*\*, <https://www.educba.com/jsp-redirect> , accesat la data de 01.02.2024

[4] \*\*\*, <https://www.smartdraw.com/uml-diagram/> , accesat la data de 02.02.2024

[5] \*\*\*, <https://stackoverflow.com/questions/9421234/cannot-connect-to-mysql-database-from-eclipse-datasource-explorer> , accesat la data de 02.02.2024

[6] \*\*\*, <https://stackoverflow.com/questions/4193905/warnings-of-valid-html5-attributes-in-eclipse> , accesat la data de 02.02.2024

[7] \*\*\*, <https://stackoverflow.com/questions/1450991/how-to-do-an-integer-parseint-for-a-decimal-number> , accesat la data de 02.02.2024