UNIVERSITATEA „POLITEHNICA” DIN BUCUREȘTI

Facultatea de Electronică Telecomunicații și Tehnologia Informației

PROIECT P.I.B.D.

Programarea unei aplicații în tehnologie Hibernate

ce operează o bază de date în tehnologie MySQL

Tițu Alex-Costin

Grupa 433A

București 2022

Cuprins

1. Structura aplicației și informații generale ........... pag. 3-5
2. Descrierea aplicației ............................................ pag. 5-8
3. Interfața aplicației ............................................. pag. 9-10
4. Funcțiile de modificare și ștergere .................. pag. 11-14
5. Funcția de adăugare ....................................... pag. 15-16
6. Funcțiile de vizualizare și modificare profil ..... pag. 17-19
7. Concluzii ............................................................... pag. 20
8. Bibliografie ........................................................... pag. 21

Game Zone

1. Structura aplicației și informații generale

1.1 Cerința proiectului și tema alocată

Creați două aplicații care să conțină o bază de date creată în sistemul de gestiune a bazelor de date MySQL și două interfețe la aceasta (baza de date este comună). La crearea interfețelor se vor folosi două tehnologii (la alegere - ex.: PHP, JSP, Hibernate, JPA, .NET etc.).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Număr temă** | **Grupă** | **Nume, prenume** | **Tabela 1** | **Tip asociere** | **Tabela 2** |
| 26 | 433A | Tițu Alex-Costin | Jocuri | M:N | Jucători |

1.2 Descrierea tehnologiilor utilizate

În realizarea aplicației de baze de date am utilizat un **sistem de gestiune al bazelor de date relațional**, mai exact **MySQL**, împreună cu framework-ul **Hibernate** pentru a face legătura între interfața accesibilă utilizatorului și baza de date în sine.

Un **sistem de gestiune al bazelor de date** este un sistem computerizat de menținere a evidenței uneia numite activități, folosind baze de date**. O bază de date relațională** este compus dintr-o mulțime finită de relații ce sunt reprezentate sub formă de tabele. O relație este unică într-o bază de date și se definește prin intermediul atributelor sale (coloanele tabelei). Fiecare atribut reprezintă tipul de entitate sau asociere e care îl descrie relația respectivă. Acestea au o denumire și un domeniu de definiție (tip de dată). Atributele iau o singură valoare (scalare).[1]

Fiecare relație are o cheie primară (PK) drept atribut, cheie unică și ireductibilă ce diferențiază fiecare element al relației de toate celelalte. O cheie primară poate avea mai multe chei candidat (superchei ce pot fi folosite drept cheie primară). O supercheie este o submulțime de atribute ale unei relații care prezintă proprietatea de unicitate. Prin urmare, o cheie primară poate fi naturală ( atribut simplu sau compus al relației cu proprietățile de mai sus, în mod natural. Ex: CNP-ul) sau artificială (atribut, de obicei simplu, ce nu reprezintă o proprietate a tipului de entitate).[2]

Pentru a face legături între relații se utilizează o cheie străină (Foreign Keys), ce constă în introducerea cheii private a tabelei ce află în relație de 1 drept atribut în tabela ce se află în relație de M (one-to-many). Pentru a face o legătură many to many se utilizează o tabelă ce conține ambele chei private ale celor două tabele ce se află în această relație. Foarte important este ca atributul reprezentat de cheia străină și cheia primară pe care o descrie să aibă domenii compatibile (ex: ambele să fie unsigned BIGINT dacă PK este astfel).

Baza de date relațională este vizualizată și modificată prin interogări de tipul SELECT, INSERT, UPDATE și DELETE. Pentru a putea vizualiza într-un mod practic o relație de legătură dintr-o relaționare many-to-many se utiliează funcția de inner join pentru vizualizarea și a datelor referite.

Pentru menținerea integrității referențiale am ales ștergerea și actualizarea în cascadă atunci când un joc/jucător este eliminat/modificat din baza de date. Astfel, orice alte elemente ce conțin id-ul cheii străine identic cu id-ul cheii primare obiectului ce urmează să fie șters/modificat, va fi și acestea eliminat/modificat în baza de date.

Integritatea referențială este proprietatea bazei de date prin care orice cheie străină fie are o valoare care se regăsește printre valorile cheii candidate referite (străine), fie are valoarea NULL.[3]

Pentru realizarea interfeței dintre utilizator și bază de date s-a utilizat tehnologia Hibernate, o tehnologie de mapare obiect-relațională special concepută pentru Java. Funcționalitatea principală a framework-ului Hibernate este aceea că transpune sub formă de clase Java, tabele de date SQL și mapează tipuri de date Java cu tipuri de date SQL.[4] Astfel, Hibernate transpune componenta relațională a tehnologiei MySQL în programare orientată obiect. Maparea și datele de conectare la baza de date sunt concretizate prin intermediul unor fișiere .hbm.xml. S-au creat astfel 4 pachete (packages) pentru a putea structura aplicația:

* **pachetul Controller** ce deține toate clasele ce fac legătura dintre interfața utilizatorului și partea de back-end;
* **pachetul DAO** (Data Access Object) ce conține clase cu interfețe de legătură între pachetul Controller și implementările DAO;
* **pachetul DAOImpl** ce conține implementările obiectelor ce facilitează accesul la baza de date, acces făcut sub formă de sesiuni (sessions) pentru a nu permite transmiterea sau primirea eronată a datelor din baza de date;
* **pachetul pojo** (Plain old Java object) ce deține fișierele hibernate mapping xml (.hbm.xml) de mapare a relațiilor din baza de date la limbajul Java obiect-orientat și clasele Java aferente în care toate atributele sunt definite.

Programarea interfeței Web se face în fișiere JSP, de această dată sub framework-ul Hibernate. Pentru a implementa și rula fișiere JSP este necesar un server web compatibil, cu un servlet container, precum Apache Tomcat (în cazul meu).

Partea de design al interfeței aplicației (front-end) constă în elemente de HTML, CSS, Bootstrap, JavaScript.

Pentru realizarea interfeței site-ului m-am folosit și de aplicația **Bootstrap Studio** ce implementează framework-ul open-source **Bootstrap**, bazat pe template-uri bazate pe **CSS** și **JavaScript** pentru tipografie, formulare, butoane, navigație, și alte componente de interfațare.[5]

**Cascading Style Sheets (CSS)** este un limbaj style sheet (pagini stilizate) utilizat pentru descriere și formatarea prezentării (design-ul) unei pagini scrise într-un limbaj de tip markup (de marcare) precum **HTML**. [6]

**HyperText Markup Language**, sau **HTML** este limbajul markup standard pentru documentele concepute pentru a fi deschise într-un browser web. Scopul limbajului este mai degrabă prezentarea informațiilor decât descrierea semanticii documentului. În cadrul dezvoltării web de tip front-end, HTML este utilizat împreună cu CSS și JavaScript.[7]

2. Descrierea aplicației

Game Zone reprezintă un site ce permite înregistrarea datelor personale a mai multor utilizatori, datele jocurilor video pe care aceștia le joacă și profilul utilizatorilor în cadrul jocului respectiv. Acest lucru le permite utilizatorilor să își compare rezultatele și performanța atinsă față de mulți alți utilizatori ai platformei.

Toate informațiile sunt stocate într-o bază de date realizată în tehnologie MySQL denumită „titualex” de structura următoare:

A picture containing graphical user interface

Description automatically generated

**Fig-1.0 Diagrama logică a bazei de date generată în MySQL**

În cadrul tabelei jocuri se va stoca numele jocului, numărul de jucători, tipul jocului, data lansării și cine a produs jocul. Pe lângă aceste informații se vor mai stoca informațiile utile despre campionate: când va avea loc următorul, câte s-au ținut și cine este ultimul campion.

În tabela jucători se stochează informațiile personale ale jucătorilor precum numele, prenumele, email-ul, data nașterii, genul și naționalitatea.

Numele tabelelor sunt date sugestiv, iar relația dintre *jocuri și jucători* este many to many sau M:N (mai mulți jucători, pot juca mai multe jocuri). Prin urmare, pentru a realiza această legătură am utilizat tabela profil ce permite ca fiecare utilizator să aibă mai multe profiluri la mai multe jocuri și invers, în cazul jocurilor.

În tabela profil se păstrează id-ul jocului pe care jucătorul respectiv îl joacă și id-ul jucătorului (pentru relația M:N), data la care a fost creat profilul, gradul (rank-ul) în jocul respectiv, dacă deține un astfel de sistem jocul și numele de utilizator din joc. Pe lângă aceste informații, mai sunt memorate numărul de campionate la care a participat, numărul de medalii obținute și dacă este campion al ultimului campionat ținut.

Mai departe, voi prezenta o schiță a ideii de la care am plecat în implementarea bazei de date.

Diagram

Description automatically generated

**Fig-1.1 Schițarea bazei de date**

Au fost alese elemente ce sunt obligatorii de completat la fiecare intrare nouă în baza de date (element NOT NULL) pentru a menține un minim de informație oferit utilizatorilor despre jocul, jucătorul sau profilul respectiv. Acestea sunt notate cum rombul albastru plin în diagrama generată de aplicația MySQL.

Elementele notate cu romb roșu gol reprezintă o cheie străină (în tabela profil), iar celalalte elemente notate cu romb albastru gol reprezintă datele, fără vreo constrângere.

În dreapta fiecărui element, este scris și domeniul pe care este definit ca tip de dată.

**VARCHAR(45)** reprezintă un șir de lungime de maxim 45 de caractere. **BIGINT** reprezintă timpul de date întreg definit pe 8 octeți. **DATE** stochează formatul de dată YYYY-MM-DD (**Y**ear, **M**onth, **D**ay). **TINYINT** stochează doar valoarea 0 sau 1, precum un tip de date Boolean.

Toate cheile primare sunt definite ca BIGINT, NN (nenule), UN (fără semn) și AI (autoincrementare).

Cheile străine sunt definite doar ca fiind BIGINT și UN, pentru a coincide cu domeniul de definiție al cheilor primare pe care le reprezintă. Ele fiind chei străine ce trebuie să corespundă altor intrări în tabela jocuri sau jucători, nu trebuie autoincrementate sau definite ca nenule, pentru a se putea alege exact ce joc și jucător reprezintă profilul respectiv.

În cele ce urmează voi expune diagrama UML a aplicației realizată în tehnologia Hiberante.

Graphical user interface, application, Word

Description automatically generated

**Fig-1.2 Diagrama UML a proiectului realizat în tehnologie Hibernate**

3. Interfața aplicației

Interfața are ca pagină de pornire o scurtă prezentare a scopului site-ului împreună cu câteva imagini sugestive. Navigația se face fie din partea de sus, fie din partea de jos a paginii. Butoanele Games, Gamers, Profiles vor direcționa utilizatorul la vizualizarea datelor din tabele jocuri, jucători, respectiv profil. Butonul drop-down Register va permite utilizatorului să aleagă dintre meniurile de adăugare de joc, jucător sau profil nou. Prin apăsarea logo-ului din partea stângă, sus, în orice moment al utilizării aplicației, se revine la pagina principală.

A screenshot of a video game

Description automatically generated

**Fig-2.0 Navigația, textul și imaginile sugestive (partea superioară)**

A screenshot of a video game

Description automatically generated

**Fig-2.1 Navigația, textul și imaginile sugestive (partea superioară)**

În acest mod se va naviga pe tot cuprinsul site-ului: cu ajutorul celor două zone împreună cu alte butoane ajutătoare. Codurile de mai jos reprezintă elementele de navigație scrisă în tehnologiile front-end menționate.

A computer screen capture

Description automatically generated with medium confidence

**Fig-2.2 Navigația din partea superioară (nav) a paginii, textul și imaginile sugestive**

A screenshot of a computer

Description automatically generated with medium confidence

**Fig-2.3 Navigația din partea inferioară (footer) a paginii împreună cu textul sugestiv**

1. Funcțiile de modificare și ștergere

Mai departe voi prezenta afișarea jocurilor înregistrate pe platformă, împreună cu modificarea și ștergerea de jocuri din baza de date. Similar, va funcționa și în cazul jucătorilor, fiind aduse doar mici modificări de interfață. Aceste operațiuni sunt SELECT, UPDATE și DELETE în MySQL.

Graphical user interface

Description automatically generated

**Fig-3.0 Afișarea datelor din tabela jocuri, împreună cu comenzile de ștergere/modificare**

Pentru a modifica o linie trebuie mai întâi bifată opțiunea de modificare (lucru care dezactivează automat opțiunea de ștergere) și completate toate câmpurile după preferințe, apoi trebuie apăsat butonul „Modifica Jocul”.

Pentru a șterge o linie trebuie mai întâi bifată opțiunea de Ștergere (lucru care dezactivează automat opțiunea de modificare și căsuțele de editare), selectat ID-ul jocului ce se dorește a fi șters și apăsat butonul „Sterge jocul” pentru a șterge jocul ce a fost selectată în drop-down.

Text

Description automatically generated

**Fig-3.1 Codul aferent funcției de ascundere a butoanelor**

În cazul modificărilor liniilor din tabel, trebuie completate toate câmpurile cu \* și trebuie ales ID-ul jocului ce se dorește a fi modificat.

Dacă nu este selectat niciun joc (atât la modificare, cât și la ștergere), la apăsarea butonului, utilizatorul este redirecționat la o pagină unde este avertizat că s-a întâmpinat o excepție.

Utilizatorul este nevoit să introducă în căsuțe și datele deja existente (deja înregistrate) despre joc în baza de date la modificarea acestuia.

Campionul actual se poate selecta doar dintre utilizatorii înregistrați în platformă. Aceștia vor fi înregistrați în tabelă cu numele și prenumele. În cazul în care există linii necompletate (blank), site-ul va pune „Unknown”, în cazul textelor, numerele și datele sunt obligatorii. (în cazul tuturor paginilor de modificare și adăugare de date se procedează la fel)

Atât în cazul ștergerii cât și în cazul modificării liniilor din tabel, utilizatorii vor fi redirecționați la pagina aferentă de adăugare de date tabelei pe care tocmai au operat-o.

Important de menționat este faptul că toate căsuțele utilizate pentru adăugarea sau modificarea datelor din tabele prezintă un Placeholder pentru a se ști exact ce trebuie completat în căsuță. Iar în cazurile în care nu pare destul de clară informația am adăugat și un tool-tip cu exemple sau formatul datelor ce trebuie introduse.

Mai jos va fi prezentat codul aferent operațiunilor de afișare, modificare și ștergere pentru tabela jocuri. (aproape identic fiind și la jucători)

Text

Description automatically generated

**Fig-3.2 Aducerea de informații din relația jucători**

**pentru a selecta campionul numai din lista de jucători existenți**

A computer screen capture

Description automatically generated with medium confidence

**Fig-3.3.1 Ștergerea și modificarea de elemente ale relației jocuri (Interfață)**

Text

Description automatically generated

**Fig-3.3.2 Ștergerea/Modificarea de elemente din relația jocuri (Back-end)**

Text

Description automatically generated

**Fig-3.4.1 Afișarea de elemente din relația jocuri (Interfață)**

Text

Description automatically generated

**Fig-3.4.2 Afișarea de elemente din relația jocuri (Back-end)**

Text

Description automatically generated

**Fig-3.5 Accesul la baza de date (Back-end)**

Important de observat aici faptul că toate accesările bazei de date se fac în sesiuni pentru a nu apărea erori în transmiterea sau primirea datelor. În cadrul unei sesiuni se așteaptă până la recepționarea sau trimiterea tuturor datelor de la/către baza de date, iar apoi se execută operațiunea de salvare (commit)

1. Funcția de adăugare

Mai departe voi prezenta funcționalitatea de **adăugare** de noi elemente la baza de date, mai exact de adăugare a unui jucător nou. (similar și în cazul jocului)

Din cele două drop-down-uri se pot alege genul (masculin, feminin sau altul) și țara (dintr-o listă ce conține toate țările).

A screenshot of a video game

Description automatically generated

**Fig-4.0 Interfața de adăugare a unui jucător nou**

Text

Description automatically generated

**Fig-4.1.1 Adăugarea unui nou jucător**

A screenshot of a computer

Description automatically generated with medium confidence

**Fig-4.1.2 Adăugarea unui nou jucător**

Text

Description automatically generated

**Fig-4.2 Trimiterea comenzilor de adăugare de jucători**

1. Funcțiile de vizualizare și modificare profil

Vizualizarea profilurilor înregistrate de utilizatori, alături de datele acestora și datele jocului pe care îl joacă se face cu ajutorul operației de inner join în baza de date MySQL.

Coloana campion, ce reprezintă dacă utilizatorul este ultimul câștigător al ultimului campionat ținut pentru jocul respectiv este reprezentat de o valoarea Boolean (Adevărat = 1 iar Fals = 0). Prin urmare, 1 va reprezenta că este campion, iar 0 că nu este campion.

A screenshot of a computer

Description automatically generated with medium confidence

**Fig-5.0 Vizualizarea Profilurilor înregistrare**

Text

Description automatically generated

Text

Description automatically generated

**Fig-5.1.1 Importarea claselor utile și preluarea informațiilor spre vizualizarea Profilurilor înregistrare**

Text

Description automatically generated

Text

Description automatically generated with medium confidence

**Fig-5.1.2 Modificarea/Ștergerea unu Profil (Interfață)**

Text

Description automatically generated

**Fig-5.1.3 Afișarea obiectelor conținute de relația Profil (Interfață)**

Text

Description automatically generated

**Fig-5.1.4 Funcțiile de modificare și ștergere ale unui Profil**

1. Concluzii

Realizarea unei aplicații web în tehnologie Hibernate reprezintă o alternativă foarte bună dacă se dorește realizarea back-end-ului în limbaj Java. Acest framework permite maparea tabelelor aferente bazei de date concepută în tehnologie MySQL, în clase Java, pentru o structură mai intuitivă programatorului. Astfel se reduce numărul de pagini JSP, iar funcționalitatea aplicației se va axa pe lucru cu clase Java. În același timp, în cazul proiectelor de scară mare, monitorizarea fiecărei funcții a aplicației se va face mult mai ușor cu ajutorul aceste tehnologii, fiind posibilă și o împărțire a task-urilor proiectului, la mai mulți programatori. Fiecare va putea avea de făcut câte o clasă (funcționalitate).

MySQL este un SGBD intuitiv cum multiple funcționalități, ce poate fi utilizat atât în proiecte de scară mică, dar și mare.

1. Bibliografie
2. *Prezentare Curs 2 „Baze de date relaţionale” -* S.l. Dr. Ing. Valentin Pupezescu – anul universitar 2021-2022
3. *Prezentare Curs 2 „Baze de date relaţionale” -* S.l. Dr. Ing. Valentin Pupezescu – anul universitar 2021-2022
4. *Prezentare Curs 2 „Baze de date relaţionale” -* S.l. Dr. Ing. Valentin Pupezescu – anul universitar 2021-2022
5. *https://en.wikipedia.org/wiki/Hibernate\_(framework)*
6. *https://en.wikipedia.org/wiki/Bootstrap\_(front-end\_framework)*
7. *https://en.wikipedia.org/wiki/CSS*
8. *https://en.wikipedia.org/wiki/HTML*