**A circular logo with text and numbers

Description automatically generatedUNIVERSITATEA NAȚIONALĂ DE ȘTIINȚĂ ȘI TEHNOLOGIE POLITEHNICA BUCUREȘTI**

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Programarea Interfețelor pentru Baze de Date**

Student: Nistor Flaviu-Cristian

Grupa: 431D

Profesor coordonator

Ș.l. Dr. Ing. Pupezescu Valentin

București 2024

**Cuprins**

1. Cerințe ......................................................…………………….3
2. Tehnologii utilizate ............................................................ 3  
   2.1. Java Spring Boot ......................................................... 3  
   2.2. MySQL ....................................................................... 3  
   2.3. Apache Tomcat .......................................................... 4  
   2.4. HTML ......................................................................... 4  
   2.5. Thymeleaf .................................................................. 4
3. Descrierea aplicației ......................................................... 4  
   3.1. Baza de date .............................................................. 4  
   3.2. Diagrama ERD ............................................................ 5  
   3.3. Implementarea funcțiilor CRUD ……………..………......... 6  
   3.3.1. Conectarea la baza de date ...................................... 7  
   3.3.2. Manipularea datelor ................................................. 7  
   3.4. Implementarea interfețelor HTML ………………………….... 8  
   3.4.1. Introducerea datelor ................................................ 9  
   3.4.2. Prelucrarea datelor ................................................. 10  
   3.4.3. Ștergerea datelor .................................................... 11
4. Concluzii ......................................................................... 12
5. Bibliografie ...................................................................... 13

1.Cerinte

**Rezumat**—Scopul acestui proiect este dezvoltarea a două aplicații, fiecare conținând o bază de date gestionată prin MySQL, cu ambele interfețe accesând o bază de date comună. Baza de date va include două tabele, „Travellers” și „BranchCFR”, cu o relație de tip M:N între acestea. Interfețele vor fi implementate folosind două tehnologii diferite, precum JSP, Hibernate, JPA, .NET, Python etc. Funcționalitățile oferite de interfețe vor permite utilizatorilor să execute operații CRUD (Creare, Citire, Actualizare, Ștergere) asupra tuturor tabelelor. Operația de „vizualizare” va include, de asemenea, vizualizarea datelor referite din tabelele asociate.

În acest document, se va face referire exclusiv la tehnologia Spring Boot.

**Cuvinte cheie**—Sistem de management al bazelor de date, MySQL, operații CRUD, relație M:N, interfețe utilizator, Spring Boot.

2.Tehnologii utilizate

**2.1. Java Spring Boot**

Spring Boot este un cadru open-source pentru Java, utilizat în dezvoltarea aplicațiilor autonome de nivel producție, bazate pe platforma Spring. Acesta oferă un set de biblioteci care simplifică inițierea și gestionarea proiectelor. Spring Boot reprezintă o extensie a principiului „convenție asupra configurației” pentru platforma Spring, având ca scop reducerea complexității configurării aplicațiilor bazate pe Spring. Deși aplicația poate fi personalizată pentru a răspunde nevoilor specifice, proiectul inițial Spring Boot vine cu o configurație implicită optimizată, recomandând cea mai bună configurare pentru integrarea cu platforma Spring și bibliotecile externe selectate.1

**2.2. MySQL**

SQL este un limbaj utilizat de programatori pentru a crea, modifica și extrage date din baze de date relaționale, precum și pentru a controla accesul utilizatorilor la acestea. Pe lângă bazele de date relaționale și SQL, un RDBMS precum MySQL colaborează cu sistemul de operare pentru a implementa o bază de date relațională în sistemul de stocare al unui computer, gestionează utilizatorii, permite accesul prin rețea și facilitează testarea integrității bazei de date și crearea copiilor de siguranță.2

**2.3 Apache Tomcat**

Apache Tomcat (prescurtat „Tomcat”) este o implementare gratuită și open-source a tehnologiilor Jakarta Servlet, Jakarta Expression Language și WebSocket. Acesta oferă un mediu HTTP de server web „pur Java” în care codul Java poate rula. Prin urmare, Tomcat este un server de aplicații web Java, deși nu este un server complet de aplicații JEE.

Tomcat este dezvoltat și întreținut de o comunitate deschisă de dezvoltatori sub egida Apache Software Foundation, fiind lansat sub licența Apache License 2.0. 3

**2.4 HTML**

Hypertext Markup Language (HTML) este limbajul standard de marcare pentru documentele destinate a fi afișate într-un browser web. Acesta definește conținutul și structura informațiilor de pe web. De obicei, este completat de tehnologii precum Cascading Style Sheets (CSS) și limbaje de scripting, cum ar fi JavaScript, un limbaj de programare.

Browserele web primesc documentele HTML de la un server web sau din stocarea locală și le redau sub formă de pagini web multimedia. HTML descrie structura unei pagini web din punct de vedere semantic și, inițial, includea indicații pentru aspectul acesteia. 4

**2.5 Thymeleaf**

Thymeleaf este un motor de template Java foarte popular, folosit pentru a genera HTML, XML și alte formate de documente în cadrul aplicațiilor web. Este utilizat adesea împreună cu framework-uri precum Spring (în special Spring Boot) pentru a crea interfețe de utilizator (UI) dinamice și pentru a separa logica de prezentare de logica de business.

3. Descrierea aplicatiei

**3.1 Baza de date**

Tema cerută este realizarea unei asocieri M:N între două tabele, „Travellers” și „BranchCFR”. Asocierea de tip M:N (Mulți-la-Mulți) se caracterizează prin faptul că fiecărui element dintr-o tabelă îi pot fi asociate mai multe elemente din cealaltă tabelă și, la rândul lor, aceste elemente pot fi legate de mai multe elemente din prima tabelă. De exemplu, într-un scenariu similar, un călător poate cumpăra mai multe călătorii de la o sucursală CFR, iar o sucursală CFR poate vinde bilete mai multor clienți. În continuare, vor fi prezentate relațiile dintre tabele și atributele specifice acestora.

A screenshot of a test

Description automatically generated

Figura 1 Tabela Travellers

A screenshot of a computer

Description automatically generated

Figura 2 Tabela BranchCFR

Pentru o asociere M:N, este necesară crearea unei tabele de legătură, care reprezintă o tabelă de joncțiune. Scopul acesteia este de a transforma asocierea many-to-many în două asocieri de tip 1:N. În această tabelă, cheia primară va consta din cheile străine ale celor două tabele, stabilind astfel o legătură între ele.

A screenshot of a test

Description automatically generated

Figura 3 Tabela intermediara Journey

**3.2 Diagrama ERD**

Diagrama ERD oferă o reprezentare vizuală detaliată, care ilustrează entitățile, relațiile dintre ele și constrângerile asociate, oferind o perspectivă clară asupra structurii și interconexiunilor din cadrul bazei de date.  
Asocierile dintre tabele sunt următoarele:

* Între „travellers” și „branchcfr” există o asociere M:N;
* Între „travellers” și „journey” există o asociere 1:N;
* Între „branchcfr” și „journey” există o asociere 1:N

A screenshot of a computer screen

Description automatically generatedFigura 4 Diagrama ERD

**3.3 Implementarea functiilor CRUD**

A diagram of a journey

Description automatically generated

Figura 5 Diagrama UML

**3.3.1 Conectarea la baza de date**

Fișierul furnizat conține configurația unei aplicații Java Spring Boot care se conectează la o bază de date MySQL, utilizând Hibernate pentru maparea obiect-relațională (ORM). Configurația include numele aplicației, detalii privind conexiunea la baza de date, cum ar fi URL-ul JDBC, utilizatorul și parola și se specifică dialectul Hibernate pentru MySQL (MySQLDialect), astfel configurează Hibernate încât să se poata actualiza automat schema bazei de date la fiecare pornire a aplicației.

A screenshot of a computer

Description automatically generated

Figura 6 File application.properties

**3.3.2 Manipularea datelor**

Pentru fiecare dintre cele 3 tabele (Travellers, Journey, BranchCFR) sunt necesare să cream fișiere Java conform diagramei UML (Figura 5). Astfel, în fișierele de tip Controller sunt create metode de tip Create, cu rolul de a insera noi înregistrări în tabele (Figura 7), List, cu rolul de a vizualiza datele dintr-o tabelă (Figura 8), Edit, cu rolul de a edita datele existente (Figura 9) și Delete, cu rolul de a șterge datele existente (Figura 10).

A screen shot of a computer code

Description automatically generated

Figura 7 createTravellersForm()

A screen shot of a computer code

Description automatically generated

Figura 8 listTravellers()

A screen shot of a computer code

Description automatically generated

Figura 9 editJourneyForm()

A screen shot of a computer code

Description automatically generated

Figura 10 deleteBranchCFR()

**3.4 Implementarea interfetelor HTML**

A screenshot of a computer program

Description automatically generated

Figura 11 travellers.html

Fișierul travellers.html conține interfața tabelei Travellers (Figura 13). În aceasta au fost adăugate butoanele Add, Change și Delete, cu scopul de a trimite la funcțiile CRUD, precum și butoanele de comutare către celelalte interfețe (BranchCFR, Journey). Suplimentar, am adăugat o funcție ceas (Figura 12) pentru a păstra tematica feroviară a proiectului. De asemenea, în fișierul static am introdus imaginea cfrcalatori.jpeg, care se poate regăsi la nivelul tuturor interfețelor.

A screen shot of a computer code

Description automatically generated

Figura 12 updateClock()

A screenshot of a computer

Description automatically generated

Figura 13 Interfata Travellers

**3.4.1 Introducerea datelor**

Din tabelul Travellers/Journey/BranchCFR, în urma apăsării butonului Add, utilizatorul este redirecționat către meniul de Create (Figura 14). Acesta permite introducerea datelor în timp real, care vor fi adăugate ulterior în tabelul principal. O posibilă îmbunătățire a codului ar fi implementarea unui mecanism de tratare a excepțiilor, astfel încât să se prevină introducerea datelor într-un format incorect (de exemplu, caractere în loc de valori numerice).

A screenshot of a computer

Description automatically generated

Figura 14 Meniul Create

A computer code on a black background

Description automatically generated

Figura 15 create-branchcfr.html

**3.4.2 Prelucrarea datelor**

Din tabelul Travellers/Journey/BranchCFR, în urma apăsării butonului Change/Modify, utilizatorul este redirecționat către meniul de Create (Figura 16). Acesta permite modificarea datelor în timp real, care vor fi schimbate ulterior în tabelul principal.

A screenshot of a notebook

Description automatically generated

Figura 16 Meniul Modify

A screen shot of a computer code

Description automatically generated

Figura 17 edit\_journey.html

**3.4.3 Stergerea datelor**

Din tabelul Travellers/Journey/BranchCFR, în urma apăsării butonului Delete, utilizatorul prin intermediul functiei Delete (Figura 18) sterge in timp real datele din tabele. Spre exemplu stergerea unui element din tabelele principale conduce la stergerea automata a unui element din tabela intermediara daca si numai daca aceasta contine o cheie straina a elementului sters.

A computer screen with blue text

Description automatically generated with medium confidence

Figura 18 Functie Delete

4. Concluzii

Proiectul prezentat demonstrează implementarea unei aplicații utilizând Java Spring Boot pentru gestionarea unei baze de date relaționale MySQL. Prin crearea și manipularea tabelelor *Travellers*, *BranchCFR* și *Journey*, s-a evidențiat importanța relațiilor între entități și a operațiilor CRUD necesare pentru administrarea eficientă a datelor.

Diagrama ERD și diagrama UML au fost utilizate pentru a clarifica structura și logica aplicației, facilitând astfel dezvoltarea unei soluții robuste și scalabile. Integrarea framework-ului Spring Boot a permis simplificarea gestionării conexiunii la baza de date, iar utilizarea Thymeleaf pentru interfața grafică a oferit o modalitate intuitivă de interacțiune cu utilizatorul.

Aplicația reușește să îndeplinească cerințele proiectului, permițând utilizatorilor să efectueze operațiuni de creare, citire, actualizare și ștergere a datelor, asigurând în același timp integritatea acestora. Totodată, prin utilizarea unui server Apache Tomcat, s-a asigurat rularea eficientă a aplicației în mediu web.

O posibilă îmbunătățire a proiectului ar putea consta în implementarea unui mecanism avansat de validare și tratare a erorilor pentru a preveni introducerea datelor invalide. De asemenea, integrarea unor funcționalități suplimentare, precum autentificarea utilizatorilor sau generarea de rapoarte, ar putea spori utilitatea aplicației.

În concluzie, acest proiect demonstrează aplicabilitatea și eficiența tehnologiilor utilizate în dezvoltarea unei aplicații web bazate pe baze de date relaționale, oferind o platformă stabilă și ușor de extins pentru gestionarea informațiilor despre călători și sucursale CFR.

5.Bibliografie

“Java Spring Boot.pdf” - Ș.l. Dr. Ing. Pupezescu Valentin

"Spring Tool Suite Documentation", Spring.io. [Online]. Available:

https://spring.io/tools. [Accesat: Dec. 6, 2024].

**Spring Framework Documentation**:

"Spring Framework Documentation", Spring.io. [Online]. Available:

https://docs.spring.io/spring-framework/docs/current/reference/html/. [Accesat: Dec.

6, 2024].

**Eclipse IDE Documentation**:

"Eclipse IDE Documentation", Eclipse Foundation. [Online]. Available:

https://www.eclipse.org/documentation/. [Accesat: Dec. 6, 2024].

**Spring Boot Documentation**:

"Spring Boot Documentation", Spring.io. [Online]. Available:

https://docs.spring.io/spring-boot/docs/current/reference/html/. [Accesat: Dec. 6,

2024].

**Spring Cloud Documentation**:

"Spring Cloud Documentation", Spring.io. [Online]. Available:

https://spring.io/projects/spring-cloud. [Accesat: Dec. 6, 2024].

Joshua Bloch, **EZective Java**, 3rd ed., Addison-Wesley Professional, 2018.

Craig Walls, **Spring in Action**, 5th ed., Manning Publications, 2018.

1https://en.wikipedia.org/wiki/Spring\_Boot

2https://en.wikipedia.org/wiki/MySQL

3https://en.wikipedia.org/wiki/HTML