**Universitatea Politehnica din Bucuresti**

**Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei**

**Programarea interfetelor pentru baze de date**

**Proiect varianta 1 (JSP)**

**Student : Dumbraveanu Ionut-Andrei**

**Grupa : 433B**

**Profesor : Pupezescu Valentin**

**Cuprins**

I. Prezentare Tehnologii Utilizate in Dezvoltare .........................................

1. MySQL........................................................................................... 3

2. JSP(JavaServer Pages)………………………………………………3

II. Baza de date .......................................................................................... 4

III. Clasa JavaBean. Legatura intre interfata si baza de date .....................

1. Prezentare genereala JavaBean ................................................... 5

2. Functii si proceduri …………………………………………………….5

IV. Implementarea interfetelor web .............................................................

1. Pagina de start .............................................................................. 7

2. Vizualizare tabele .......................................................................... 8

3. Adaugare entitate in tabela............................................................. 8

4. Modificare entitate in tabela.......................................................... 13

5. Stergerea entitatii in tabela........................................................... 16

V. Bibliografie ............................................................................................ 17

**I. Prezentare Tehnologii Utilizate in Dezvoltare**

**1.MySQL**

Tehnologia folosita pentru crearea bazei de date: **MySql**.

**“MySQL** este un sistem de gestiune a bazelor de date relaționale, produs de compania suedeza MySQL AB și distribuit sub Licența Publică Generală GNU. Este cel mai popular SGBD open-source la ora actuală, fiind o componentă cheie a stivei LAMP (Linux, Apache, MySQL, PHP).” [[1]](#b1)

**2.JSP(JavaServer Pages)**

Pentru interfata am folosit **JSP(JavaServer Pages).**  In acest mod putem edita si vizualiza tabelele.

Tehnologia Java Server Pages (JSP) este cea mai populara metoda de a crea interfete Web pentru aplicatiile care ruleaza pe platforma Java, creata de Sun. Ea se bazeaza pe tehnologia numita Java Servlets fiind, de fapt, o completarea a acesteia in ideea crearii cat mai facile a paginilor Web dinamice.[1]

Punctul central al tehnologiei o reprezinta asa-numitele *pagini JSP* care sunt, practic, fisiere text care combina descrieri HTML cu cod Java. Paginile JSP sunt gestionate si accesibile prin intermediul unui **server de aplicatii**. Acesta primeste cereri venite prin HTTP de la un browser Web. Daca o cerere refera o pagina JSP, serverul prelucreaza local pagina respectiva si, in functie de continutul acesteia, genereaza dinamic o pagina HTML pe care o trimite, ca raspuns, browser-ului. Este important de retinut faptul ca toate prelucrarile legate de paginile JSP se fac pe partea de server, acestea nefiind niciodata transmise in forma originala catre client. In plus, trebuie retinut faptul ca serverul de aplicatii include si o masina virtuala Java in care ruleaza atat codul Java intalnit in paginile JSP cat si obiectele instantiate de acesta. Pentru cei interesati, facem precizarea ca procesul de prelucrare pe partea de server a paginilor JSP presupune, de fapt, crearea unor clase Java Servlet care urmeaza regulile scrise in pagina JSP si include codul Java din aceasta. Clasele astfel generate sunt apoi compilate si rulate in masina virtuala amintita. [2]

Un alt element important este ca orice aplicatie Web JSP trebuie sa fie **instalata *(deployed)*** in serverul de aplicatii inainte de a putea fi rulata. Instalarea presupune copierea paginilor JSP si a claselor Java folosite de acestea in locatii bine stabilite de catre serverul de aplicatie utilizat, eventual intr-o forma arhivata de tip .jar (Java Archive) sau .war (Web Archive). [[2]](#b2)

**II. Baza de date**

Am realizat o baza de date in MySQL WORKBENCH , ea continand 2 tabele : **hotel** si **camera**. Asocierea dintre ele este  de tipul M:N , ceea  ce presupune crearea unei tabele intermediare , pe care am numit-o **cazare**.

Tabela **hotel** contine urmatoarele coloane:

* **idhotel**, de tip BIGINT , cheie primara
* **nume**, de tip VARCHAR(45)
* **locatie**, de tip VARCHAR(45)
* **rating**, de tip INT
* **dataconstruire**, de tip VARCHAR(45)

Tabela **camera** contine urmatoarele coloane:

* **idcamera**, de tip BIGINT, cheie primara
* **numarcamera**, de tip BIGINT
* **numarpaturi**, de tip INT
* **balcon**, de tip VARCHAR(45)
* **bucatarie**, de tip VARCHAR(45)

Tabela **cazare** contine urmatoarele coloane:

* **idcazare**, de tip BIGINT, chieie primara
* **idhotel**, de tip BIGINT, cheie straina
* **idcamera**, de tip BIGINT, cheie straina
* **status**, de tip VARCHAR(45)

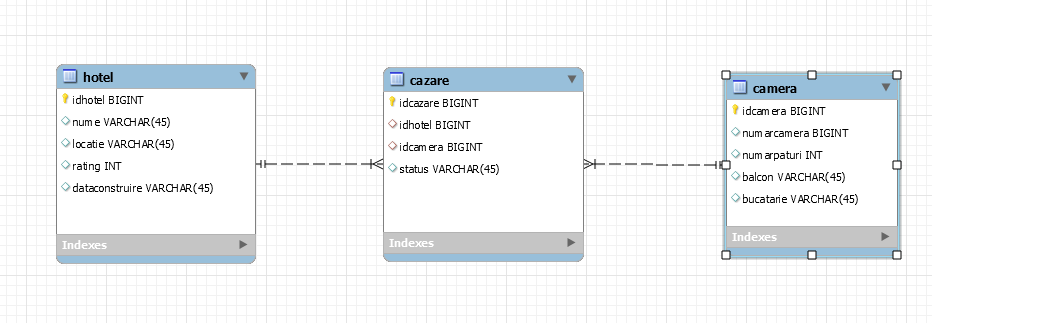
Datorita asocierii M:N ,  **idhotel** si **idcamera** sunt chei straine(FK) pentru tabela **cazare.** Cheile primare(PK) corespunzatoare fiecarei tabele au fost setate cu urmatoarele proprietati :  not null (NN) si auto-increment (AI). 

Figura 1: Diagrama Bazei de Date

**III. Clasa JavaBean. Legatura intre interfata si baza de date**

**1.Prezentare genereala JavaBean**

JavaBean este o clasa Java special construita pentru a putea obtine informatii din baza de date.

Principalele functionalitati ale acestei clase sunt:

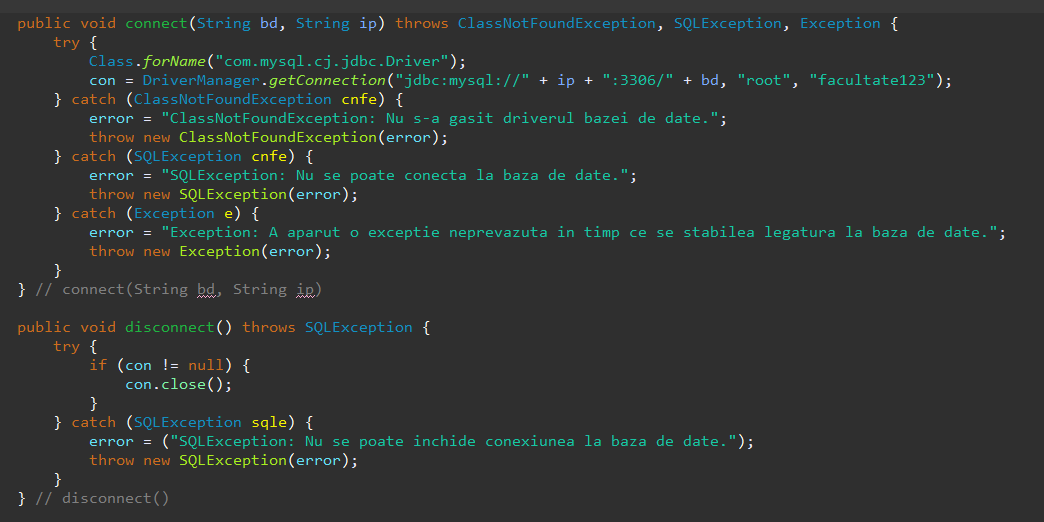
• Conectarea la baza de date

• Extragerea de rezultate din baza de date

• Introducerea informatiei in baza de date

**2.Functii si proceduri**

Procedurile connect si disconnect sunt folosite pentru a conecta/deconecta instanta din *JavaBean* la baza de date.

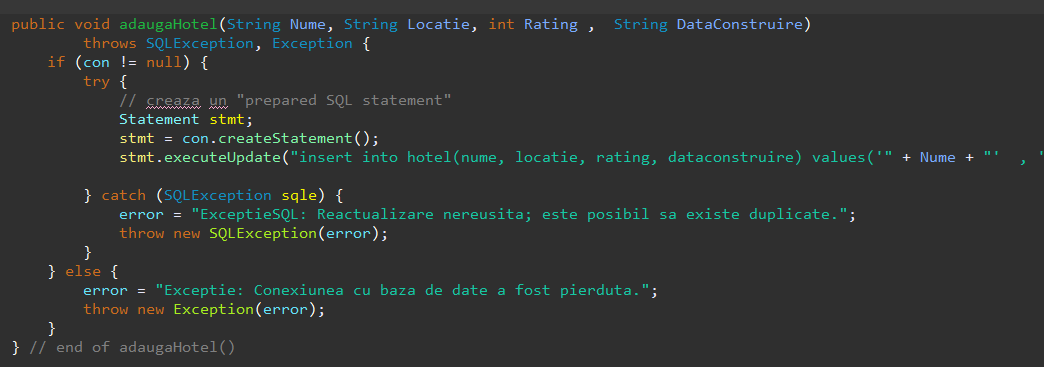


Figurile 2-3: functiile *connect* si *disconnect*

Toate functiile si procedurile din *JavaBean* se bazeaza pe atributul con de tip *Connection*.

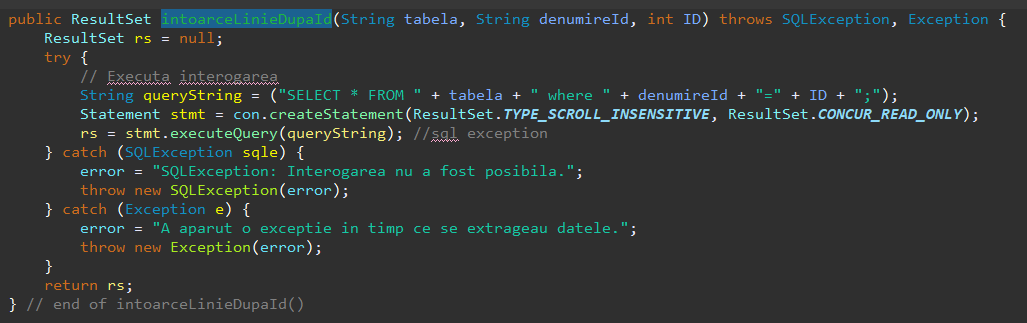
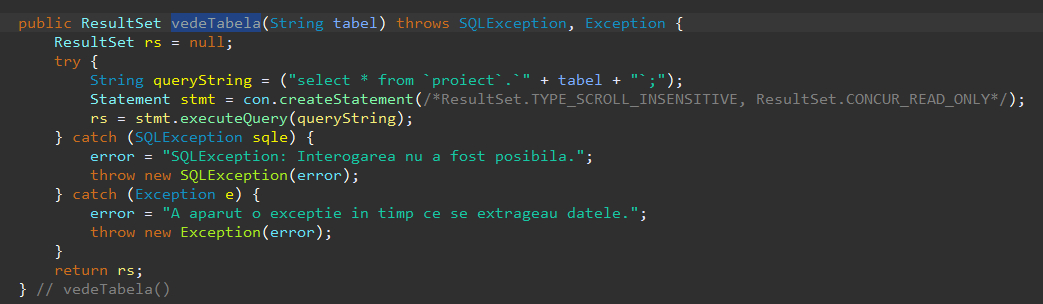
Daca conectarea nu este posibila, o exceptie este ridicata.

Pentru fiecare tabel in parte, exista functii de adaugare, modificare,vizualizare si stergere in tabela. Spre exemplu functia *adaugareAngajat* creeaza un statement pentru inserarea datelor de intrare in baza de date.

Figura 4: functia *adaugaAngajat*

Pentru a vizualiza datele majoritatea datelor, folosim functiile *vedeTabela* si *intoarceLinieDupaId*.

Functia *vedeTabela* returneaza un ResultSet care continue toate instantele din tabela selectata ca parametru al functiei.



Figurile 5-6: functiile *vedeTabela* si *intoarceLinieDupaId*

Functia *intoarceLinieDupaId* returneaza un ResultSet care continue toate elementele din tabela data ca parametru care satisfac conditiei:

\*tabela\*.\*denumireId\* == ID ,respectiv

\*tabela\*.\*denumireId1\* == ID1 si \*tabela\*.\*denumireId2\* == ID2

Pentru stergerea anumitor entry-uri din baza de date se foloseste procedura *stergeDataTabela*

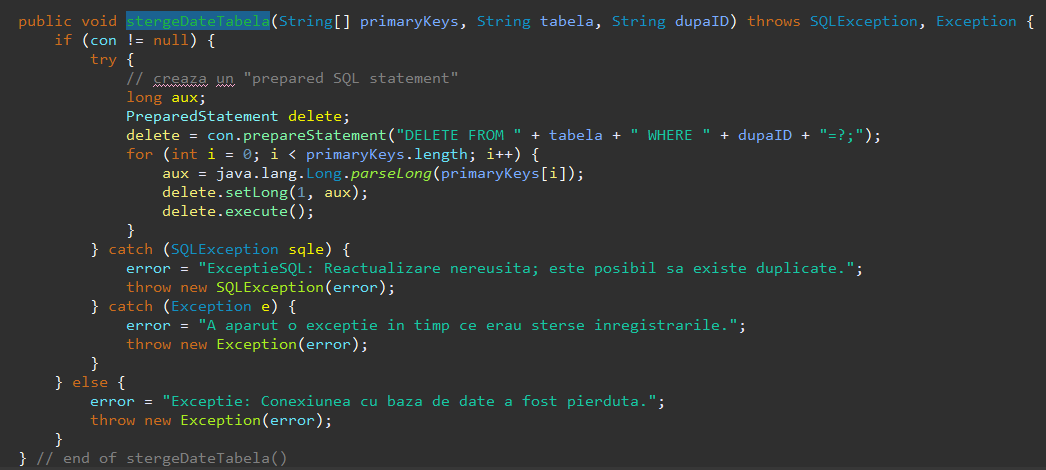


Figura 7: functia *stergeDateTabela*

**IV. Implementarea interfetelor web**

**1. Pagina de start**

Pentru a intra in pagina de start, este folosit urmatorul URL: http://localhost:8080/proiectPIBD/index.html

Pagina principala a interfetei (**index.html**) facuta prin aceasta tehnologie permite accesul spre paginile tabelelor si spre paginile din care putem modifica tabelele.



Figura 8: pagina principala a interfetei

Din pagina principala **index.html**, utilizatorul are posibilitatea de a selecta ce tebela doreste sa acceseze pentru a adauga o inregistrare, modifica si a sterge inregistrarile deja existente,pentru   fiecare din cele trei tabele din componenta bazei de date**,** accesand hyperlink-urile corespunzatoare. Accesul la vizualizarea tabelelor se va face prim intermediul barii de navigatie de sus.

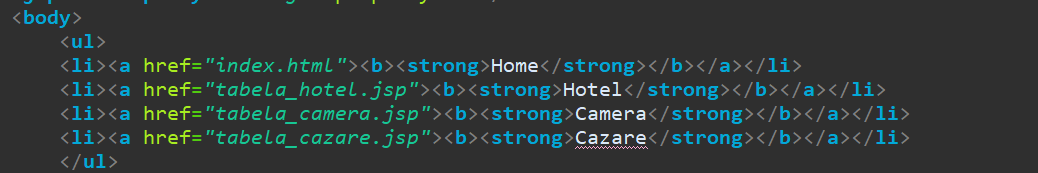


Figura 9:Codul utilizat pentru bara de navigatie

**2. Vizualizare tabele**

Pentru tabela **Hotel** vom da click pe “Hotel” din bara de navigatie. Similar vom proceda si pentru vizualizarea celorlalte doua tabele, **Camera** si **Cazare**. Pentu a ne intoarce la pagina principala putem apasa pe butonul “Home” din bara ne navigatie.



Figura 10:Vizualizare tabela **Hotel**

**3. Adaugare entitate in tabela**

Functionalitaea paginilor este similara pentru cele trei tabele, astfel : pentru a adauga o inregistrare noua in tabela (**hotel**, **camera** sau **cazare**) utilizatorul trebuie sa acceseze hyperlink-ul corespunzator. Se va deschide o pagina in care utilizatorul va completa campurile disponibile iar la sfarsit, prin apasarea butonului “**Adauga**”, datele vor fi preluate in tabela corespunzatoare.

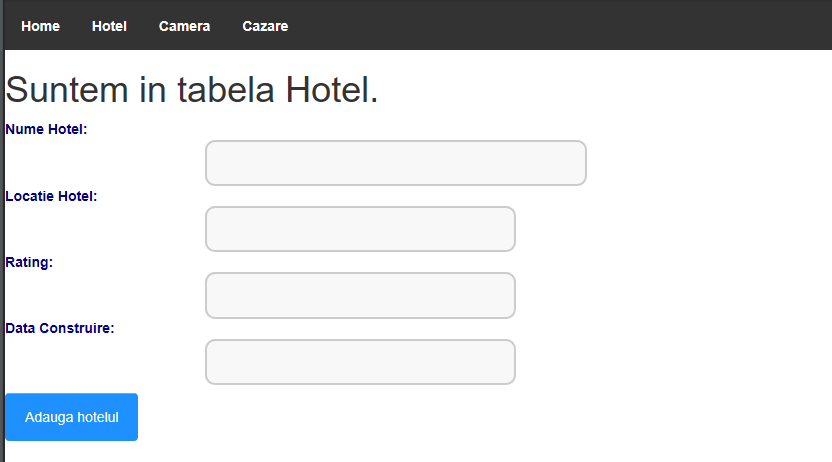


Figura 11: adaugare in tabela **Hotel**

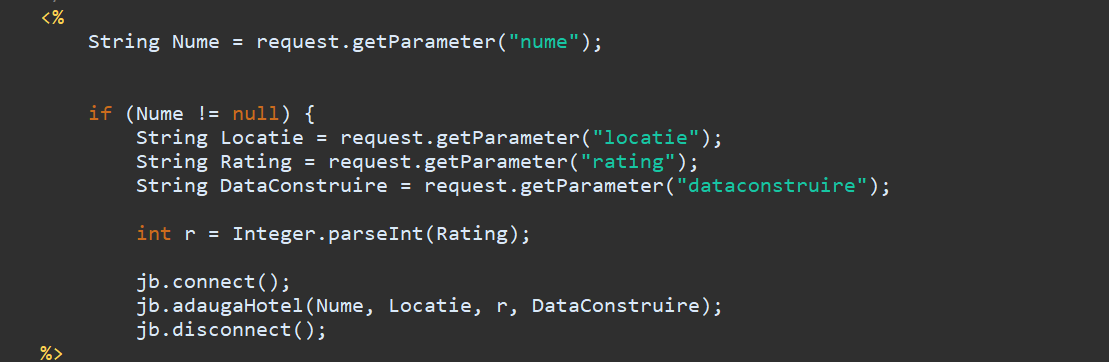


Figura 12: codul corespunzator  functiei adaugare

Dupa prelucrarea datelor va fi afisata o pagina cu mesagul care confirma acest lucru:”Datele au fost adaugate.”

De asemenea ne putem intoarce la tabela hotel prin butonul Back.

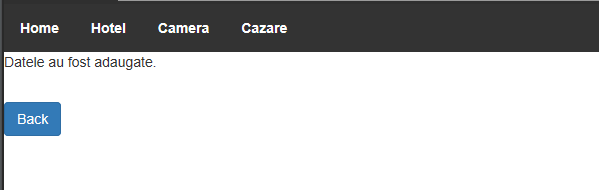


Figura 13: confirmarea adaugarii unui hotel

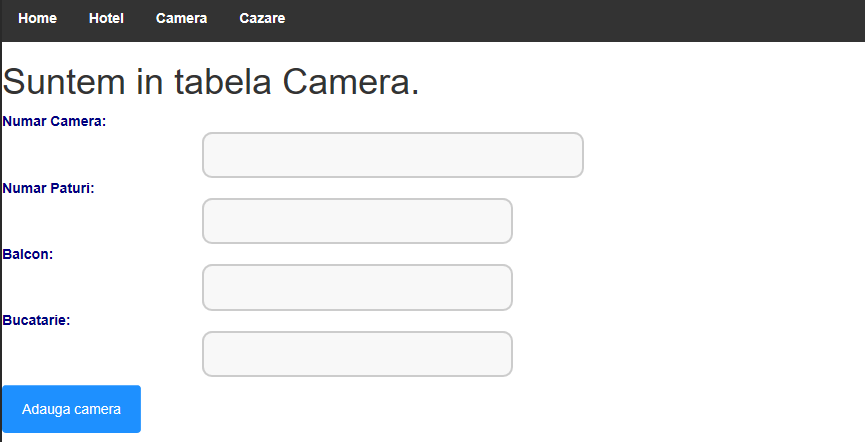


Figura 14: adaugare in tabela Camera

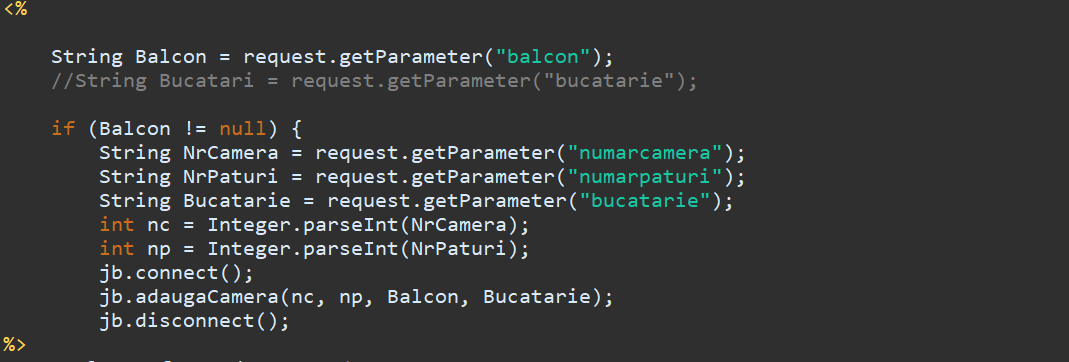


Figura 15: codul corespunzator  functiei adaugare

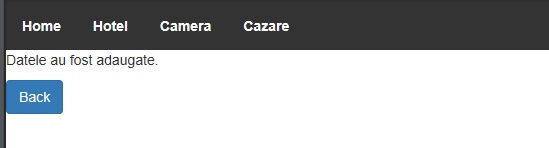


Figura 16: confirmarea adaugarii unei camere

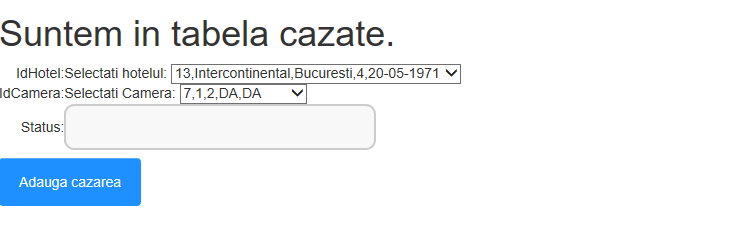


Figura 17: adaugare in tabela Cazare

Pentru adaugarea unor noi campuri in tabela Cazare se va selecta mai intai hotelul si camera dupa care se va introduce statusul camerei din hotelul respectiv.

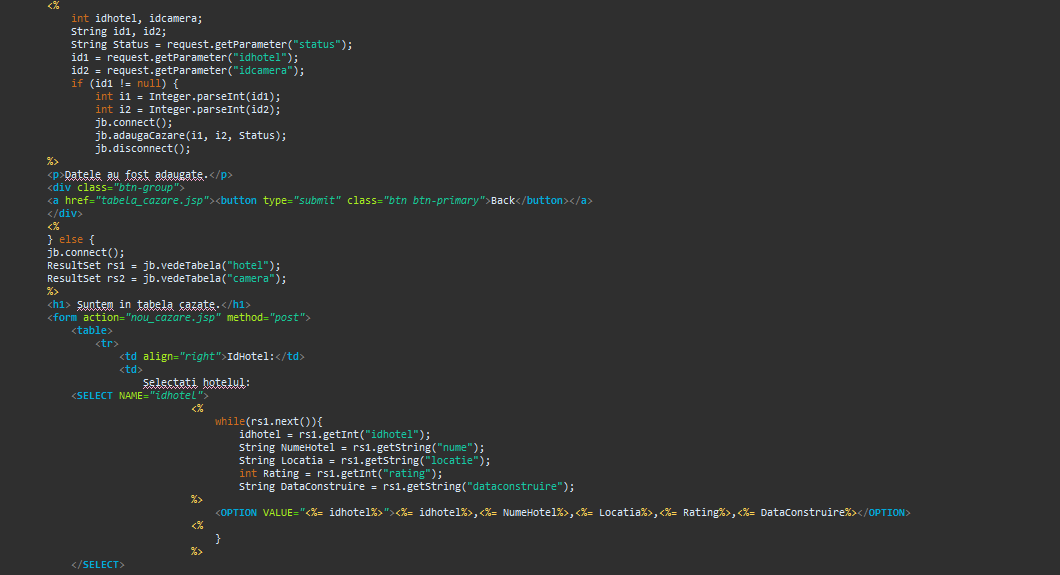


Figura 18:codul functiei de adaugare in tabela Cazre + codul prin care selectam hotelul

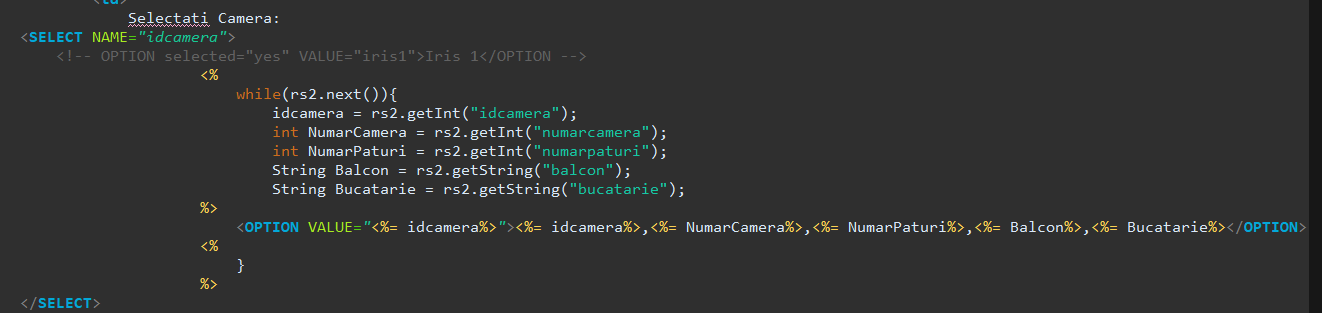


Figura 19: codul prin care selectam camera

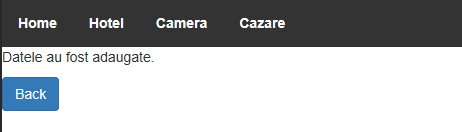


Figura 20: confirmarea adaugarii datelor

**4. Modificare tabela**

Pentru modificarea unor campuri , se apasa pe butonul “**Modifica**“ , unde ne va redirectiona spre pagina de modificare unde vom putea selecta linia dorita, apoi se va apasa din nou butonul “Modifica linia“. Se va deschide o pagina noua, asemanatoare cu cea de adaugare insa vor ramane datele initiale, acestea urmand a fii modificate.

De asemenea avem si butonul “Back” care ne va trimite inapoi.

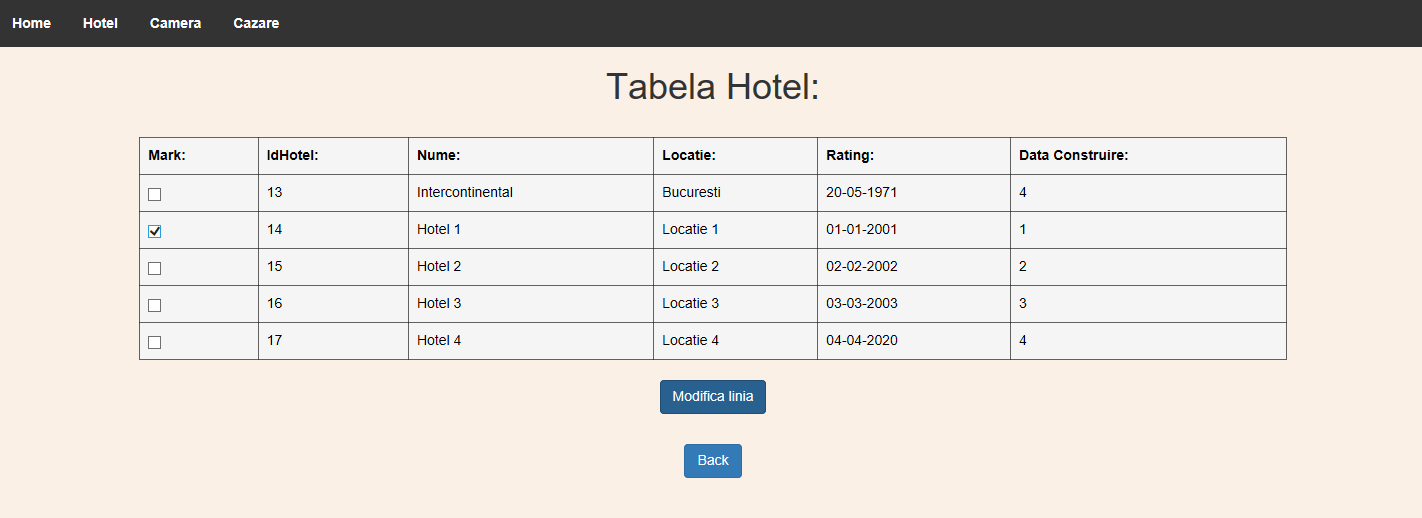


Figura 21: selectam linia pe care o dorim sa o modificam din tabela hotel



Figura 22: Modificare campuri



Figura 23: confirmarea modificarii datelor

Pentru fiecare tabela se va afisa confirmarea modificarii datelor si vom avea un buton care ne va trimite inapoi la tabela respectiva.

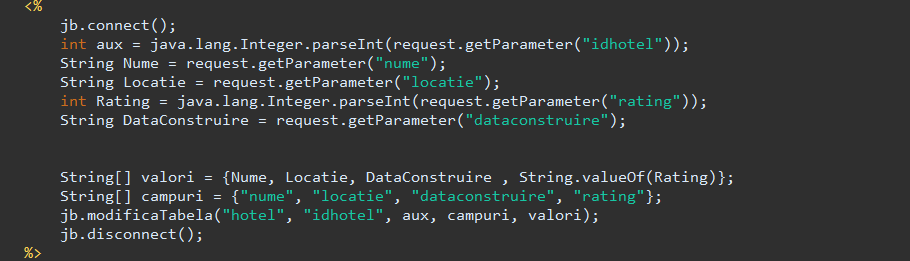


Figura 24: codul pentru modificare in tabela hotel

Procedam la fel pentru cele 2 tabele.

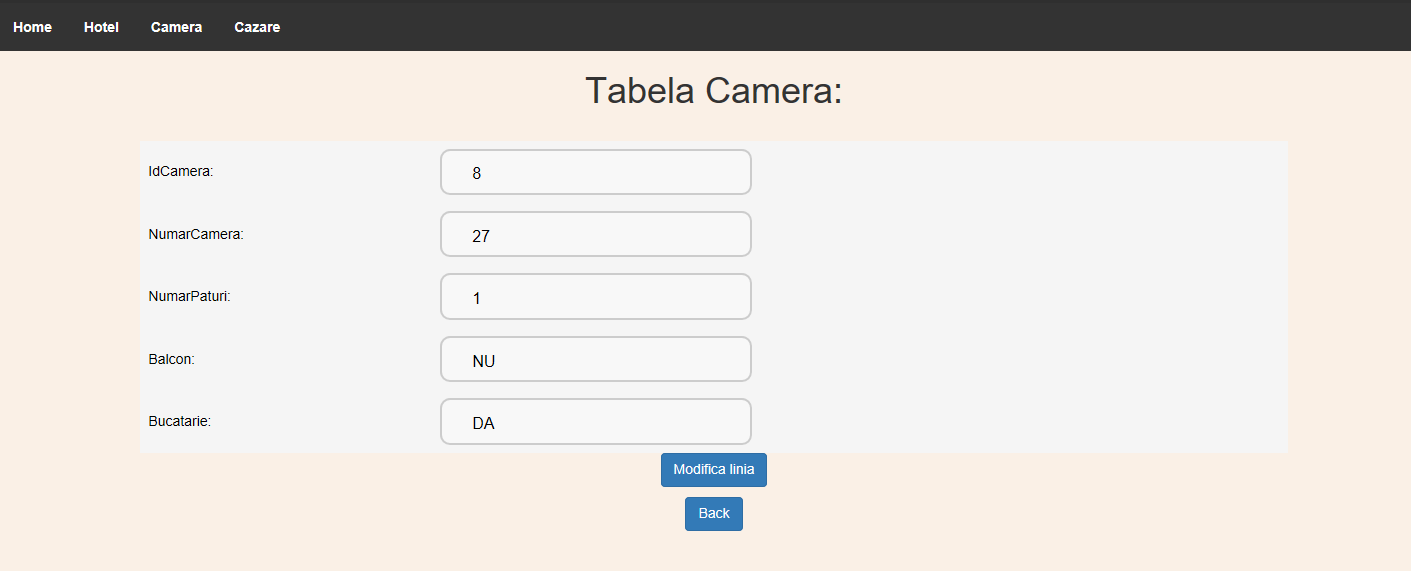


Figura 25: Modificare tabela camera



Figura 26: Confirmare modificare tabela camera

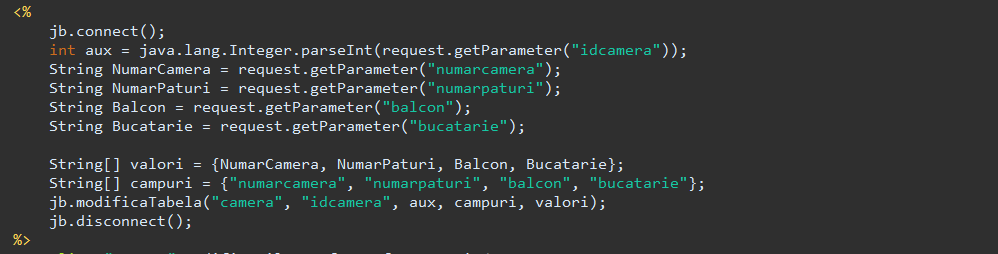


Figura 27: codul pentru modificare in tabela camera



Figura 28: Modificare tabela cazare

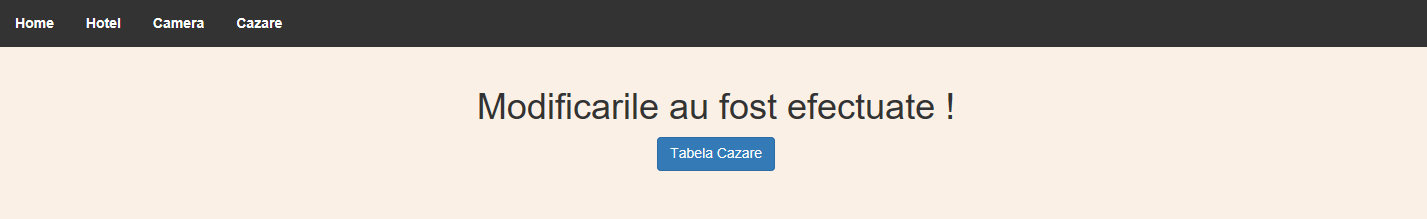


Figura 29: Confirmare modificare tabela cazare

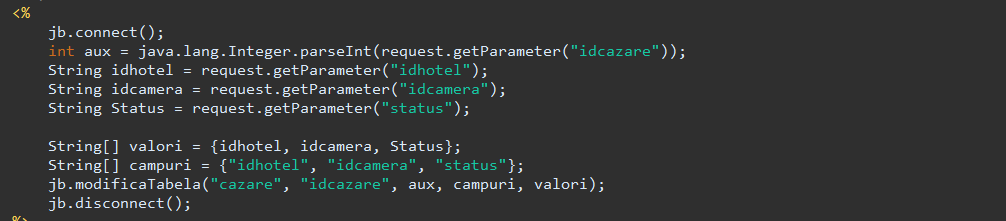


Figura 30: codul pentru modificare in tabela cazare

**5. Stergerea entitatii in tabela**

Pentru a sterge campuri din tabela se apasa pe butonul “**Sterge**” , apoi se selecteaza linia dorita.

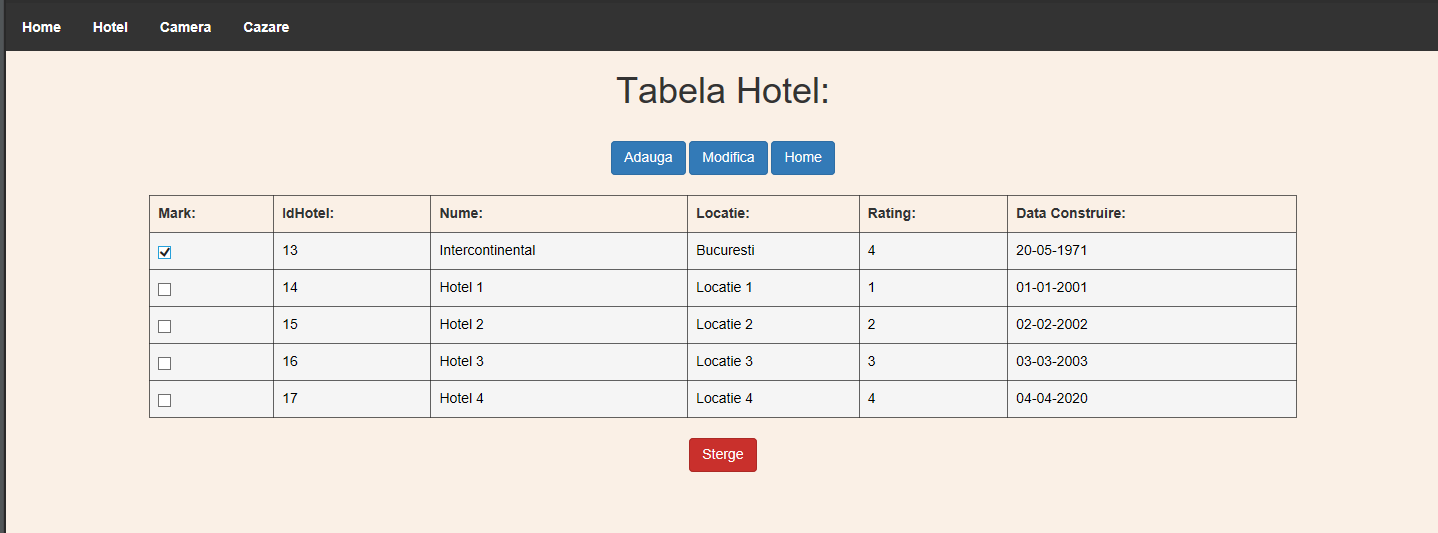


Figura 31:Selectam linia(sau liniile) pe care dori sa o stergem.

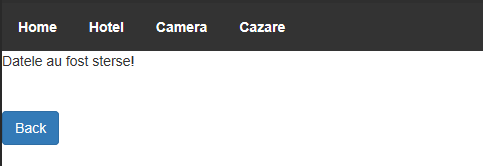


Figura 32: confirmarea stergerii datelor

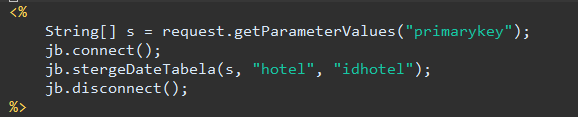


Figura 31 : stergere din tabela angajati

Similar se procedeaza pentru a sterge datele si pentru cele doua tabele, Camera si Cazare.

Pentru imbunatatirea aspectului paginilor html am folosit **CSS,** design-urile fiind procurate de pe site-ul celor de la w3cdhools.[[3]](#b3)

De asemenea pentru erori am apelat la site-uri precum: stackoverflow[[4]](#b4), youtube[[5]](#b5), google[[6]](#b6).

**V.Bibliografie**

**[1] :** https://bursasite.ro/ce-este-mysql/

**[2] :** http://labs.cs.upt.ro/labs/sprc/html/jsp.html

**[3] :** https://www.w3schools.com/

**[4] :** https://www.stackoverflow.com/

**[5] :** https://www.youtube.com/

**[6] :** https://www.google.com/