**Assignment 1\_2 Description**

Descriere

Aplicatia e constituita dintr-o pagina de logare una de inregistrare una pentru utilizatori obisnuiti si una pentru administrator. Un utilizator obisnuit dupa logare/inregistrare poate sa vizualizeze date despre zborurile avioanelor: numarul zborului, orasul de plecare, orasul destinatie, ora de plecare si de sosire si tipul avionului care va efectua zborul. El poate deasemenea sa afleora locala cand un anumit avion va ajunge lao anumita destinatie. Pentru aceasta el trebuie sa introduca numarul zborului dorit si sistemuli va returna ora locala a destinatiei. Administratorul va fi logat intr-o sectiune separata a aplicatiei. El poate sa vizualizeze zborile, sa adauge un zbor, sa elimine un zbor sau sa modifice un zbor.

Concelptual architecture of distributed system

Aplicatia respecta arhitectura pe nivele, avand un nivel de Presentation constituit din paginile jsp, un nivel de Busines Layer reprezentat de Servlet-uri si Data Acces reprezentat de modelul DAO de conexiune la baza de date.



Fiecare Servlet se foloseste de metodele doGet() si doPost() pentru a transmite atribute ce le obtine din Data Acces spre paginile jsp unde le foloseste pentru a construi tabelul cu zboruri. Proiectul contine urmatoarele pachete:

* com,
* dao,
* entities,
* resources (director de resurse pentru hibernate),
* webapp,
* bootstap.

Tehnologii Folosite

Aplicatia este rulata cu un server Tomcat. Ea foloseste jsp-uri pentru a creea paginile html si clase java pentru Servlet-uri. Baza de date este pastrata in My SQL Server si schema folosita (flights) contine urmatoarele tabele:

* Flight
* Flight Number
* Airplane Type
* Departure City
* Departure Time
* Arrival City
* Arrival Time
* City
* Id
* City Name
* Latitude
* Longitude
* User
* User Id
* Username
* Password
* Role

Pentru conexiunea cu baza de date se foloseste Hibernate si adnotatii pentru a specifica relatile cu clasele din packetul „entities”. Aplicatia contine un document denumit pom.xml care contine urmatoarele dependente folosite:

* Junit
* Org.hibernate
* Javax.servlet
* Mysql
* Common-logging
* Jst1

Pentru o interfata mai prietenoasa s-a folosit libraria web Bootstrap care care extrage din packetul „bootstrap” urmatoarele surse:

./bootstrap/css/bootstrap.min.css

./bootstrap/js/bootstrap.min.js

http://code.jquery.com/jquery-1.11.1.min.js

Acestea document definesc clase css si javascript iar prin folosirea lor aplicatia are atat un nivel mai bun de ordonare a tagurilor cat si un design mai frumos

Database Design

Scema din baza de date „flights” contine 3 tabele care prin ORM-ul „Hibernate” sunt cuplate cu clasele din pachetul entities.



UML Deployment Diagram

Proiectiul cuprinde patru clase de care extind HttpServlet. Hash-link-urile lor sunt: /register, /user, /crud si /localhour.



Usage consideration

La rularea aplicatie se deschide pagina web denumita <http://localhost:8080>. Aceasta pagina de pornie incarca documentul „login.jsp” unde este afisat un form de inregistrare in care utilizatorul trebuie sa isi introduca username-ul si parola si sa apese login sau sa apese pe link-ul „Register” si sa isi creeze cont.Acest link redirectioneaza utilizatorul spre „register.jsp” unde intr-ul alt form se cere user-ul si de doua ori parola. Daca userul nu mai exista in baza de date si daca parolele introduse in cele doua campuri corespun iar numarul de caractere este mai mare decat 4 atunci o noua inregistrare va vi adaugat in baza de date si utilizatorul va fi directiont spre pagina de regular user „user.jsp”. Daca la logare un utilizator foloseste un cont de regular user acesta va fi redirectionat pe pagina de „user.jsp” iar daca foloseste un cont de administrator acesta va fi directionat spre pagina „admin.jsp”. Un utilizator obisnui va putea vizualiza data despre zboruri intr-o tabela generata cu informatie din baza de date. El poate deasemene sa afle la ce ora va ajunge un anumit zbor in orasul destinatie. Pentru asta el va introduce numarul zborului dorit si va apasa butonul „Local Time” iar un mesaj de tip alert va aparea si va contine data si ora sau un mesaj de eroare in cazul lipsei zborului. Administarorul va putea vizualiza tabela de zboruri si in prima linie a acesteia vor exista containere de input in care se pot introduce date despre un zbor si insera in tabela si baza de date. El mai poate modifica sau sterge datele despre un zbor. Tabela va fi automat redesenata. Delogare din aplicatie se face atat la administrator cat si la utilizator obisnuit prin butonul de „Logout” din partea dreapta sus a paginii. Dupa delogare aplicatia va afisa pagina de logare.