

Managementul resurselor unui aeroport

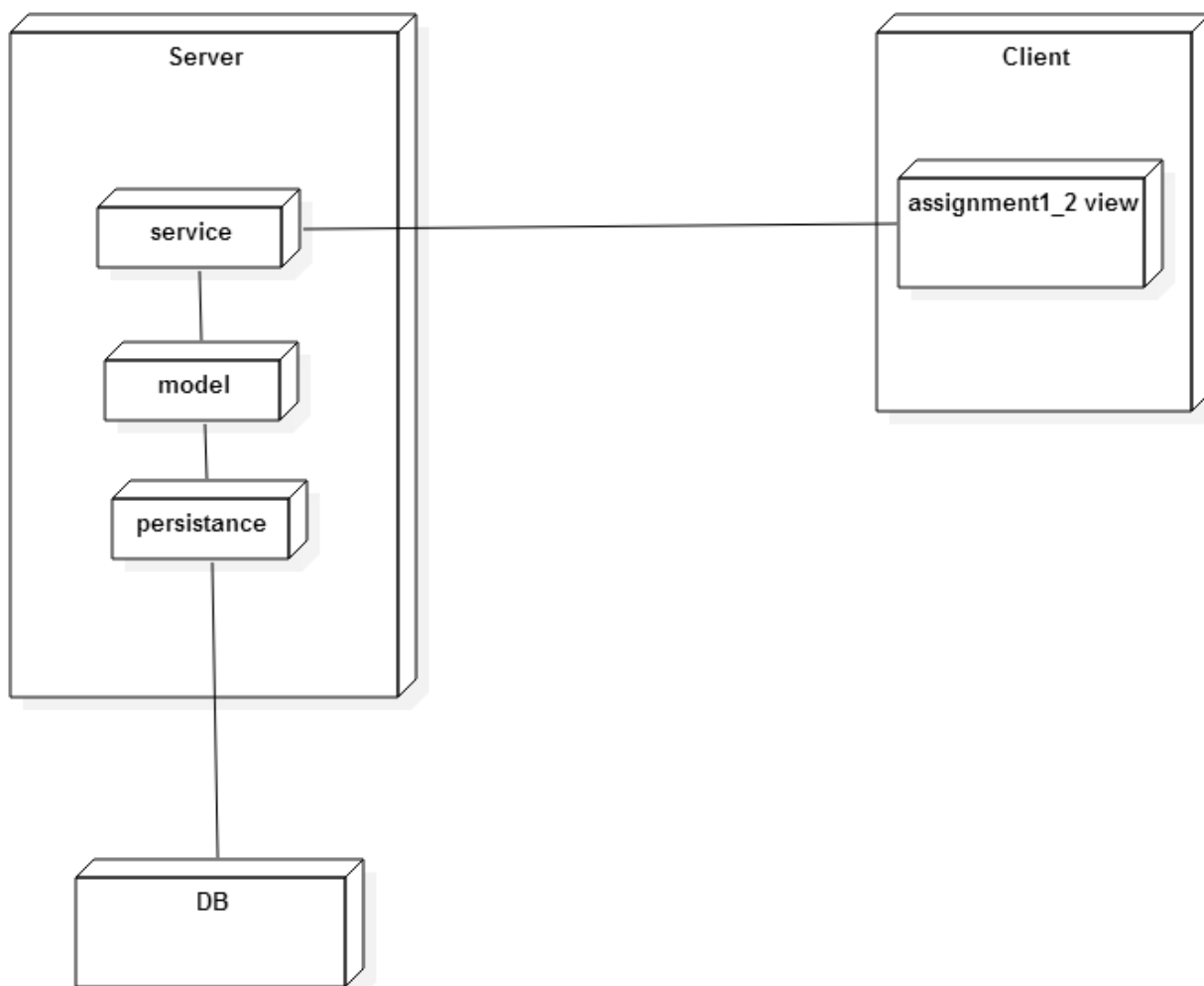
Tolas Ramona

Grupa:30243

1.Arhitectura conceptuala

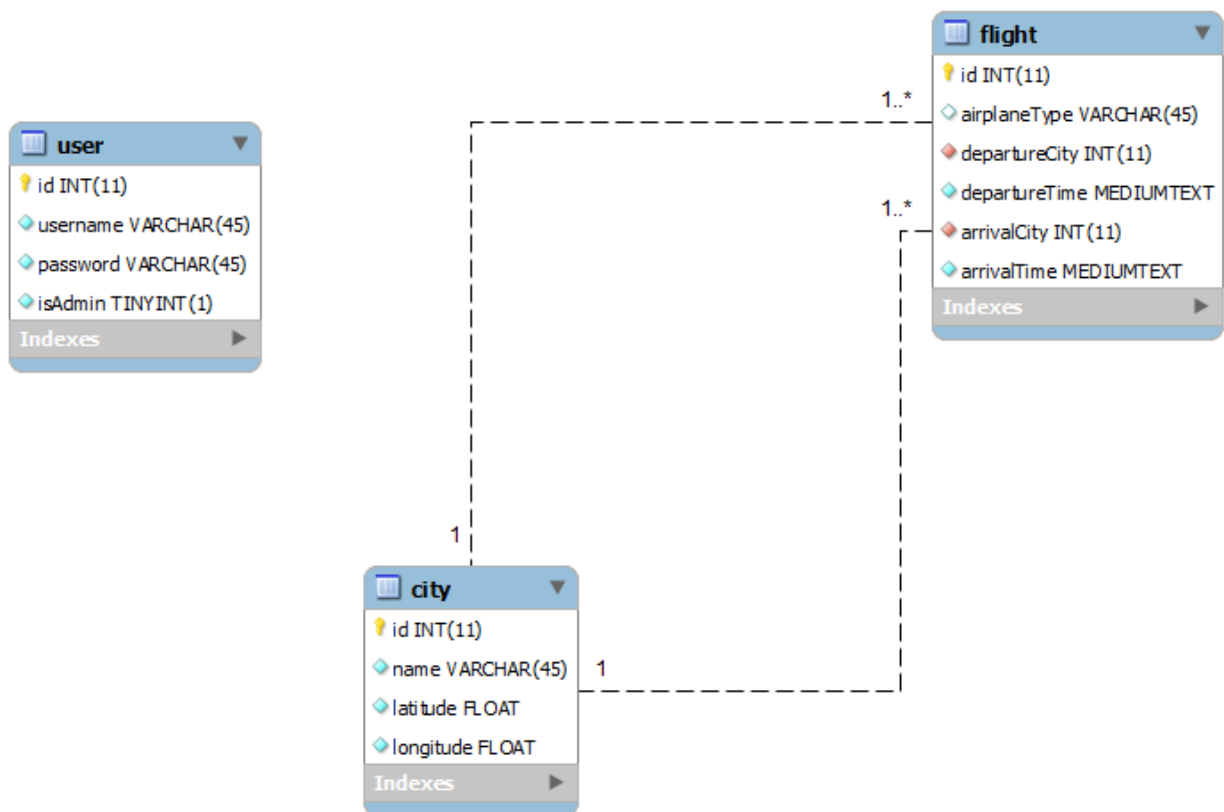
Proiectul are un design de tip layer. In ceea ce priveste design-ul arhitectural se distinge un cunoscut pattern : Client-Server.In partea de Client se afla partea de View a aplicatiei, implementata prin JSP. Tehnologiile folosite pentru a realiza view-ul sunt : librariile javascript(bootstrap, ajax, jQuery). Partea de servicii, server-ul este conceput in stilul layer, ingloband urmatoarele pachete:service,model si persistance. Pachetul service contine Servlet-urile aplicatiei, adica endpoint-urile care sunt accesate din partea de view. Servelt-urile folosesc pachetele model si persistance. Pachetul model contine modelele conceptuale ale enitatilor implicate in flow-ul aplicatiei : City,Flight,User. Partea de persistance este asigurata d epachetul cu acelasi nume care are clasele CityDao,FlightDao si UserDAo. Pentru implementarea acestor clase care au ca scop accesul la baza de date s-a folosit biblioteca java Hibernate care face accesul la baza de date mult mai usor.

Mai jos se poate vizualiza diagrama conceptuala a aplicatiei care afiseaza intr-o forma grafica cele prezentate anterior:



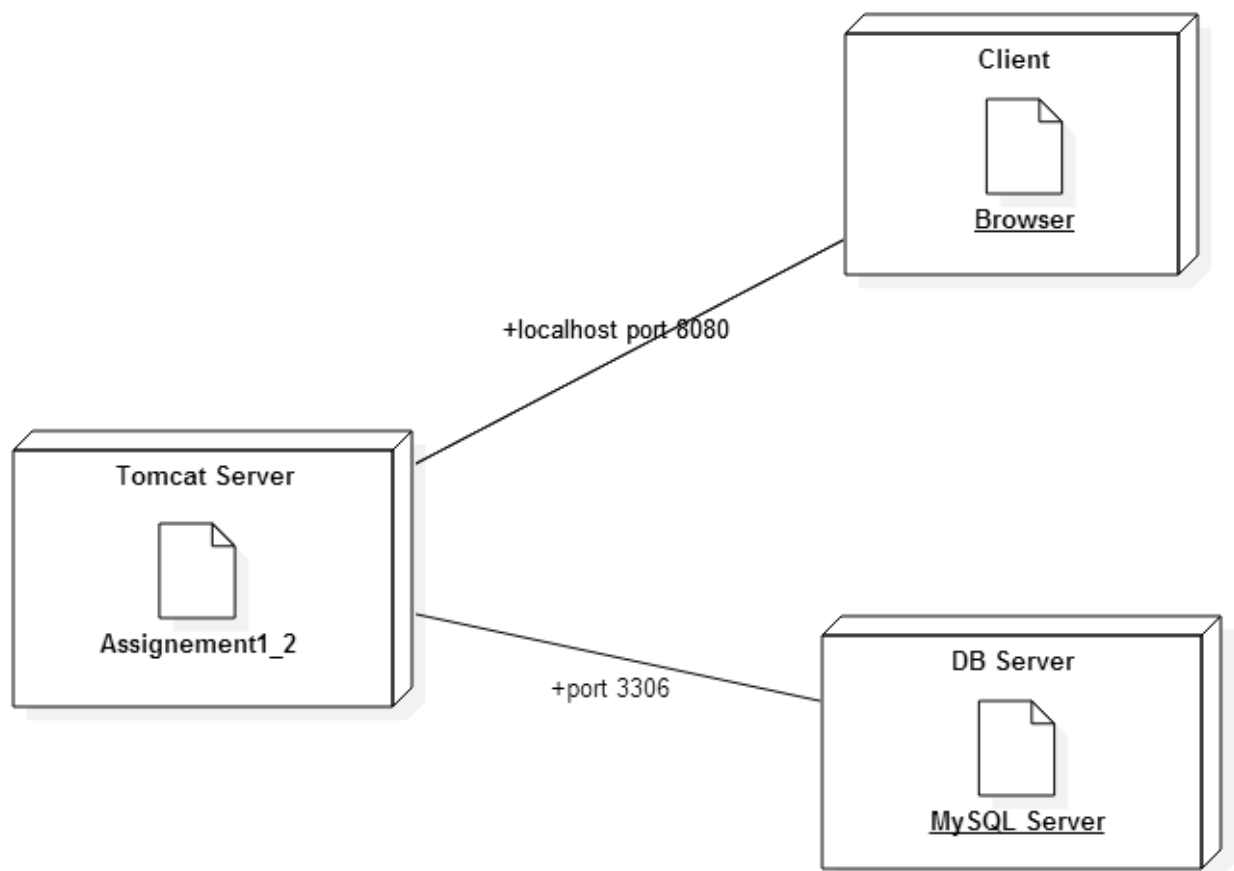
2.DB design

Pentru a asigura persistenta datelor , toate infomatiile au fost stochate in baza de date. S-a folosit serverul MySql. Baza de date are 3 tabele : City, Flight si User. Tabelul City si Flight sunt legate printr-o legatura de cheie straina, intrucat Flight are ca si atribut orasul d eplecare si orasul de sosire, iar in baza de date, acest lucru s-a implementat prin cheie straina d ela Flight la City, iar in tabelul Flight regasim id-ul orasului de plecare sau de sosire.Se regaseste mai jos diagrama bazei de date:



3. UML deployment diagram

Intrucat aplicatia este de tip Client-Server, exista 2 componente principale . Ambele sunt gazduite de server-ul Tomcat . Acestea comunica prin protocolul http, iar partea de server ofera o serie de servicii a caror descriere va fi mai jos, servicii pe care client-ul le cere prin intermediul metodelor POST si GET . Mai jos este prezentata diagrama UML de deployment unde se pot distinge componentele. Se observa ca pentru a comunica cu server-ul clientul foloseste domeniul localhost si portul 8080. Astfel request-ul ajunge la Tomcat care cauta in resursele disponibile si verifica daca exista resursa ceruta de client. O alta comunicare este cea intre server si serverul de baze de date. Acesta comunicare se realizeza prin intermediul portului 3306. Baza de date folosita este *assignment-flight* la care server-ul se conecteaza cu user-ul root ,si cu parola empty string. Acest lucru este important deoarece aceste configurari sunt in fiserele de configurare pentru Hibernate si asigura partea de persistenta a datelor.



4. Endpoint-urile serverului

URL
http://localhost:8080/assignment1_2/FindAllFlightsServlet
http://localhost:8080/assignment1_2/FindAllCities
http://localhost:8080/assignment1_2/InsertNewFlight
http://localhost:8080/assignment1_2/DeleteFlight
http://localhost:8080/assignment1_2/UpdateFlight
http://localhost:8080/assignment1_2/FindCityByName
http://localhost:8080/assignment1_2/LocaTimeServlet

5. Manual de utilizare

- Se descaraca arhiva cu aplicatia.
- Se ia din folderul target fisierul assignmen1_2.war
- Se pune in Tomcat, in foloder-ul web-app
- Se porneste Tomcat dand dublu click pe fisierul catalina-start din foloderul Tomcat
- Se deschide broser-ul si se introduce URL-ul http://localhost:8080/assignment1_2/