MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

Assignment 1.2

Web Application using Request-Reply

Ichim Daniel Alexandru
Gr 30645

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

Contents

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE
UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA
FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

1. Requirements

Design, implement and test a three-tiered distributed system to view and post flights for an airport. The system consists of the following tiers: Presentation, Business Layer and Data Access.

1.1. Functional Requirements

- Users log in. Users are redirected to the page corresponding to their role.
- Client role
  - A client can view on his/her page all the flights in a list or table.
  - A client can query for the local time of the flight arrival and departure cities computed based on cities geographical coordinates.
- Administrator role
  - The administrator can perform CRUD operations on flights (Create, Read, Update and Delete)
- Each flight consists of the following information: flight number, airplane type, departure city, departure date and hour, arrival city, arrival date and hour.
- Each city has associated its geographical coordinates: latitude and longitude.
- In order to display the local time, the geographical coordinates of the city are passed to an external web service which will return the actual time value.
- The simple users will not be able to enter the administrator page (e.g. by log-in and then copy-paste the admin URL to the browser)

1.2. Implementation technologies

- Use the following technologies: HTML, Java Servlets and Hibernate ORM

1.3. Non-functional requirements

- Security: use authentication in order to restrict users to access the administrator pages (cookies, session, etc.)

2. Conceptual architecture

Hibernate is a high-performance Object/Relational persistence and query service which is licensed under the open source GNU Lesser General Public License (LGPL) and is free to download. Hibernate not only takes care of the mapping from Java classes to database tables (and from Java data types to SQL data types), but also provides data query and retrieval facilities.

Java Servlets are programs that run on a Web or Application server and act as a middle layer between a request coming from a Web browser or other HTTP client and databases or applications on the HTTP server.

Using Servlets, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

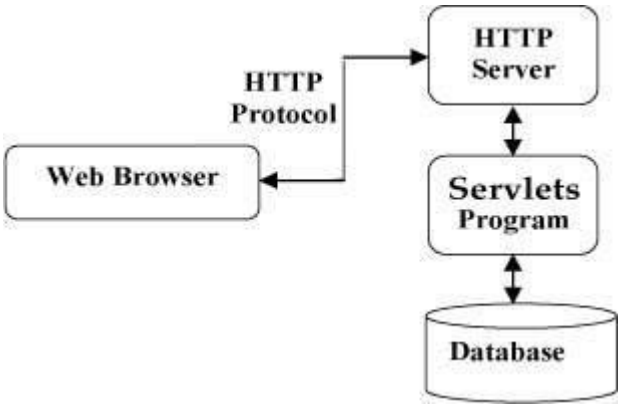


Figure 1 Servlets Architecture

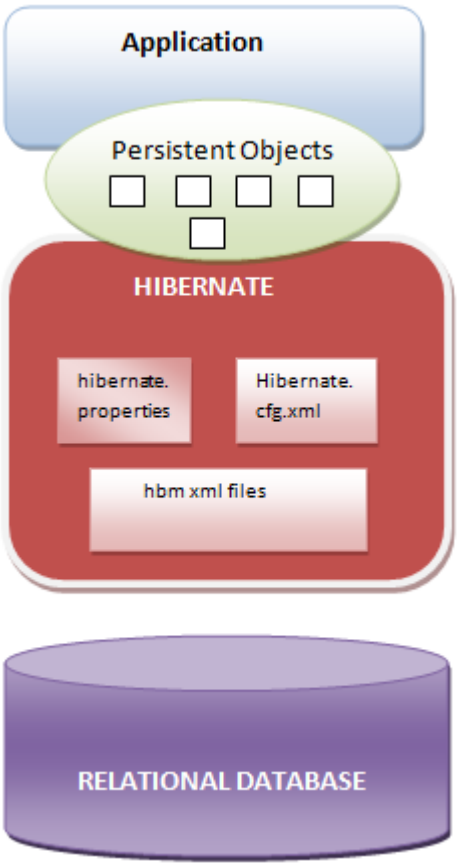
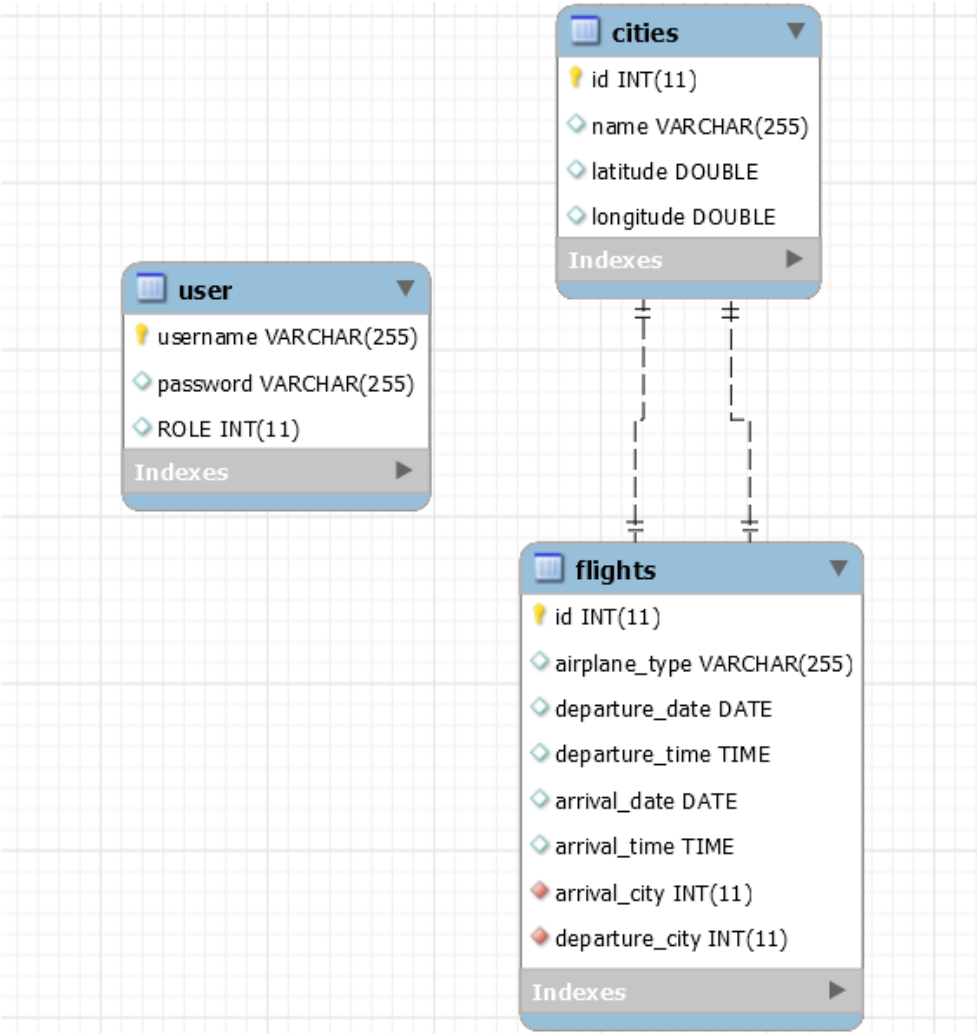


Figure 2 Hibernate Architecture

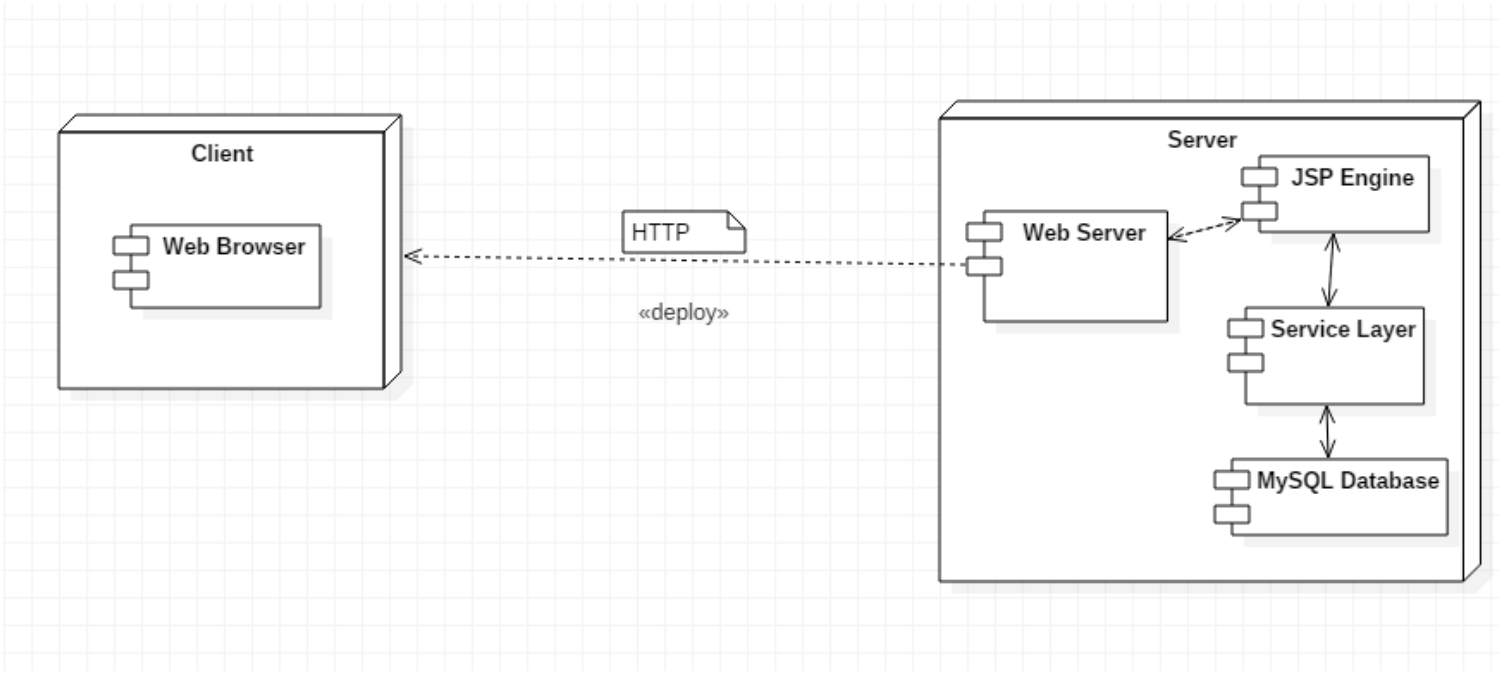
3. Database design

I designed the application using three tables for data persistence:

- **USERS** – information about the regular users, and administrators, required for login.
- **CITIES** – contains a list of cities, with their corresponding coordinates of latitude and longitude, required to compute the local time of the city.
- **FLIGHTS** – contains all the flights information required for storing, each flight has two references in the cities table for the departure and arrival city.



4. Deployment diagram



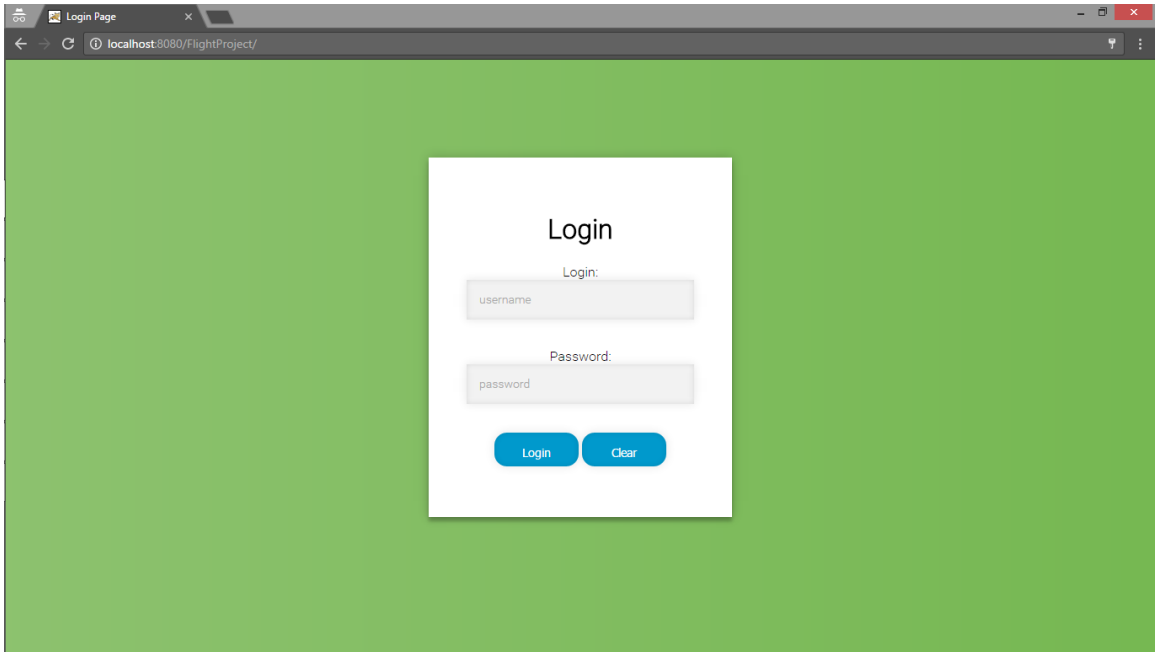
5. Build and execution considerations.

The steps to perform, in order to run the application are the following:

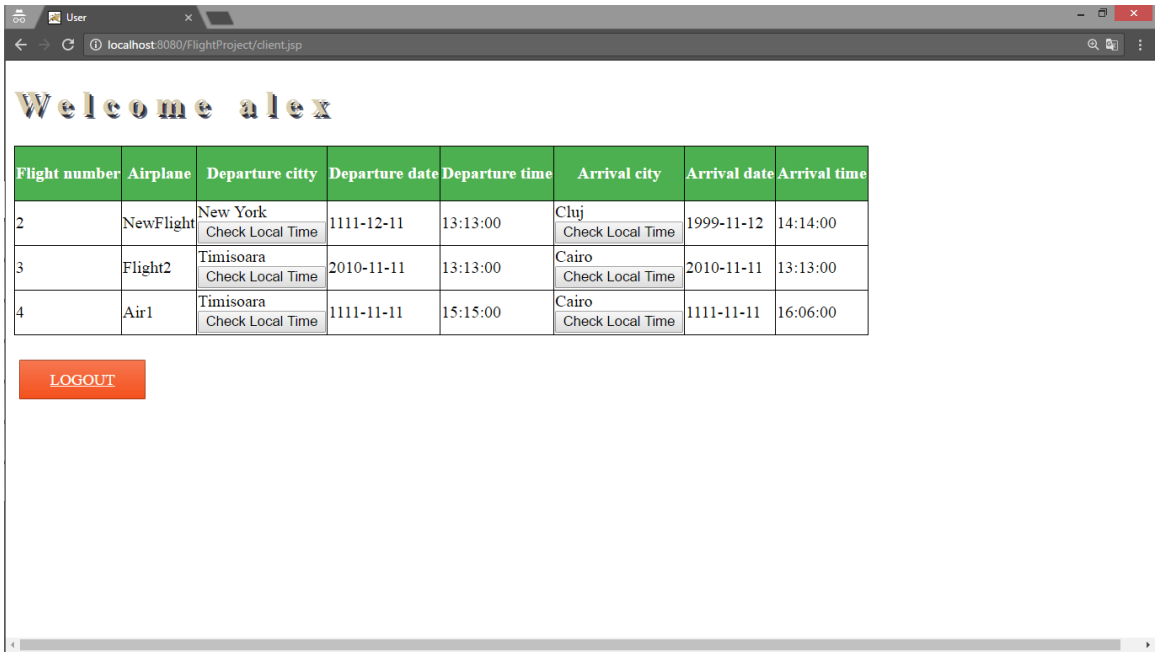
- Setup Java (preferably JDK8) on your machine (http://www.coned.utcluj.ro/~salomie/DS_Lic/LabAndProject/DS_Lab_Resources.pdf)
- Download Apache Tomcat from http://tomcat.apache.org (Tomcat 8.0.38 or later)
- Install MySQL Workbench to monitor the database http://dev.mysql.com/downloads/workbench/
- Import the project as Maven project into Eclipse/Intellij (preferably Eclipse)

Steps to run the program in Eclipse:

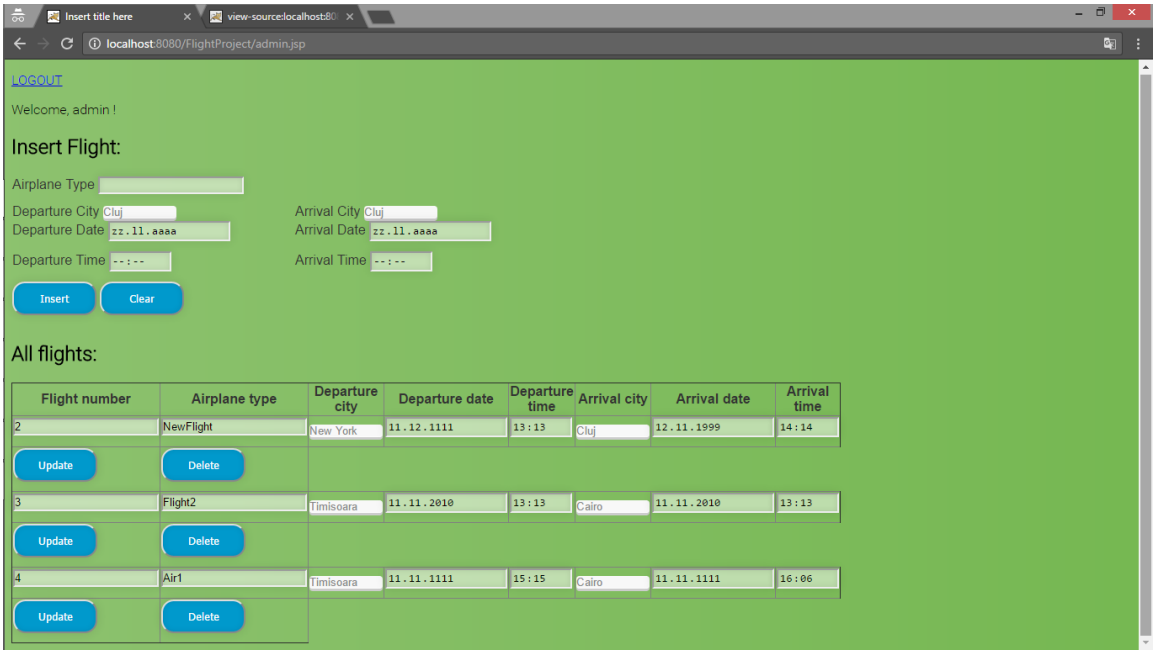
After completing all the steps above, you can run the application from Eclipse, right click on project and run on server (make sure you select Tomcat as server), the application can be accessed from the following URL in your Web Browser or within the IDE at http://localhost:8080/flightproject/ or http://localhost:8080/flightproject/index.jsp



The regular user login page after providing the appropriate credentials, is client.jsp. You are redirected after entering correct credentials to the client.jsp or shown an error otherwise.



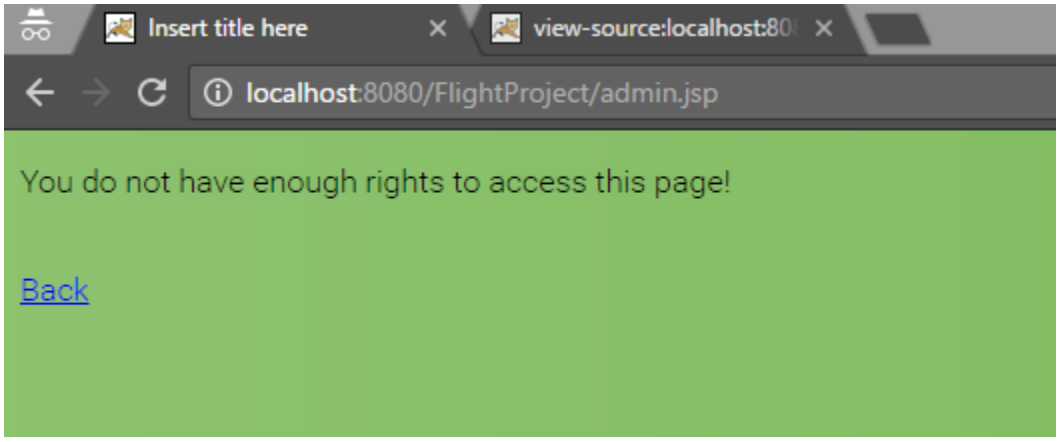
If you will log in as an admin you will be redirected to admin.jsp page.



After you logged in as an admin, you can insert a new flight by using the INSERT FLIGHT form, on the top of the page. Below this form it is shown a table with all the flights listed in the database, the flights fields can be edited so they and the using the update button, the flight will be updated. For deleting an flight, the user has to press the DELETE button under the corresponding flight.

Security:

- You can't access the admin or client pages without logging in, you will be redirected to the login page if you try to access them by copy pasting the URL.
- As client, you can't access the admin page, you will be redirected to the following page.


Insert title here
view-source:localhost:80
localhost:8080/FlightProject/admin.jsp
You do not have enough rights to access this page!
Back