DISTRIBUTED SYSTEMS

Assignment 1

Request - Reply

Communication

Paradigm

Crisan Adrian

30443

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

o A client can view on his/her page all the flights in a list or table.

o A client can query for the local time of the flight arrival and departure cities computed based on cities geographical coordinates.

Administrator role

o 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 (e.g. <http://new.earthtools.org/webservices.htm>) 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:

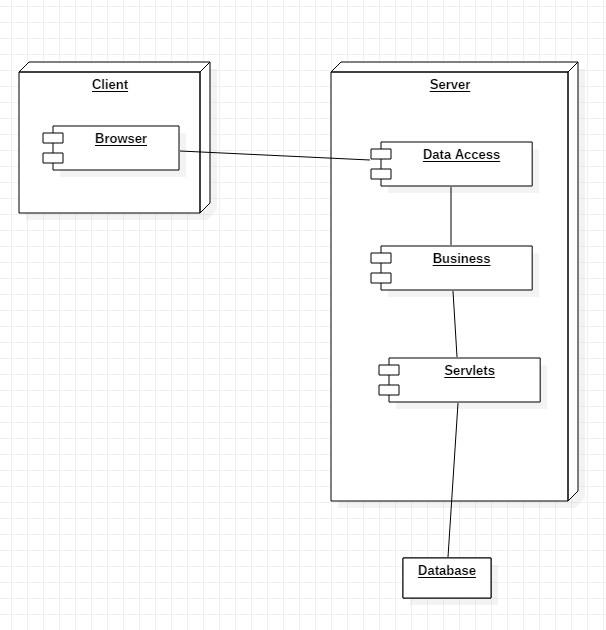
Use the following technologies: HTML, Java Servlets and Hibernate ORM.

1.3. Non-functional requirements:

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

1. **Conceptual architecture of the distributed system**

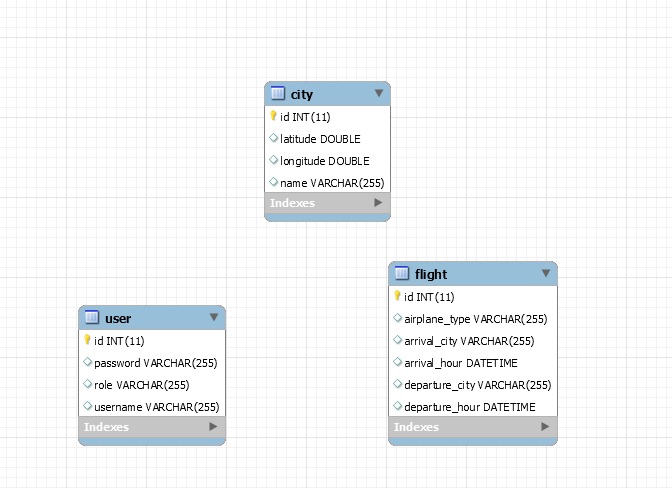
The architecture used in this application is a distributed one, based on the client-server architecture, in which the client role is played by the web browser one uses to connect to the application. This implies that the application is deployed to a server. For the server component, a 3-tier design architecture was used, to separate the responsibilities of the classes. The first tier, the “presentation” layer, or “servlets” as it is in the application, was used to instantiate and be a container for the servlet objects. The second and third tiers, “business” and “data access” are represented by the “dao” and “beans” packages.



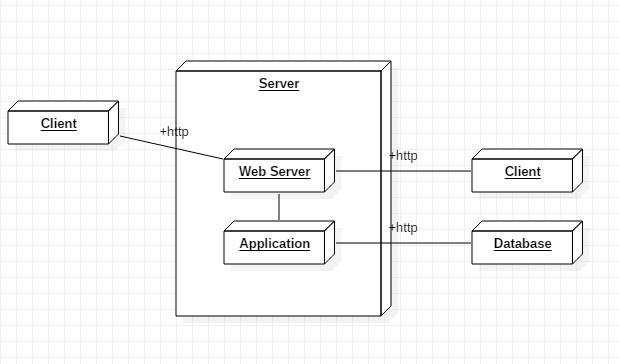
1. **Database design**

For this application a database containing three tables was used. A table “user” which contains the users of the application with their data and their corresponding role of USER or ADMIN. A table “flights” which contains data about the flights such as destination and arrival city. The third table “city” is used to store data about the cities.

The database was implemented in MySQL, and the connection from the application to the database was used using Hibernate ORM.



1. **UML Deployment Diagram**



1. **Readme and execution considerations**

The application was developed in Eclipse Photon IDE, being a Maven project. All the dependencies are found in the pom.xml file and the external libraries are automatically imported by Maven.

To configure the project properly, the Maven project must be created using the web artifact and Tomcat must be installed in order to run the server. The application can be runned by using the Tomcat server option to run the server and will run on a port on localhost, and then accessing the link <http://localhost:8080/assignment12/login.html> from any browser.