

# **CIS 4301 – Information and Database Systems I**

## **Term Project**

### **Online Airline Reservation System**

In this project, you will design and implement a simple interactive airline reservation system similar to orbitz ([www.orbitz.com](http://www.orbitz.com)). The system is a typical client-server application with a web-based front-end for the user and a database backend. This document provides a high-level description of the project and outlines the deliverables expected of your team. Given the imprecise nature of this description, it is the responsibility of your team to come up with a proper specification. It is crucial that you have a good solid design before you start coding.

The purpose of the online system is to create convenient and easy-to-use online system for passengers, trying to buy airline tickets. The system is based on a relational database with its flight management and reservation functions. We will have a database supporting dozens of major cities around the world as well as hundreds of flights by various airline companies. Above all, we hope to provide a comfortable user experience along with the best pricing available.

**Project is worth 30% of your final grade. Each team can have maximum of two members.**

#### **Requirements**

Users of the system should be able to retrieve flight information between two given cities with the given dates/times of travel from the database. A route from city A to city B is a sequence of connecting flights from A to B such that: a) there are at most two connecting stops, excluding the starting city and destination city of the trip, b) the connecting time is between one to two hours. The system will support two types of user privileges, Customer and Employee. Customers will have access to customer functions, and the employees will have access to both customer and flight management functions.

The customer should be able to do the following functions:

- Make a new reservation.
  - One-way
  - Round-Trip
  - Multi-city
  - Flexible Date/time
  - Confirmation.
- Cancel an existing reservation.
- View his itinerary.

The Employee should have following management functionalities:

- Customer functions.

- Reporting
  - Get all customers who have seats reserved on a given flight.
  - Get all flights for a given airport.
  - View flight rooster, schedule.
  - Get all flights whose arrival and departure times are on time/delayed.
  - Calculate total sales for a given flight.
- Administrative
  - Add/Delete a flight
  - Add a new airport
  - Update fare for flights.
  - Add a new flight leg instance.
  - Update departure/arrival times for flight leg instances.

Each flight has a limited number of available seats. There are number of flights that go from/to different cities at different dates and time.

### **Part 1: Database Design Document (Due Date October 25, 2007)**

This will include a precise description of the project as you understood it, the complete ER diagrams outlining the database schema, and the tools and technologies that you intend to use for the project.

Convert the E/R diagram to a relational schema. Indicate primary keys, foreign keys, triggers and any other constraints you may have. Clearly specify any assumptions you make and your rationale. Implement the SQL database schema for the design using CREATE TABLE commands. Have enough cities, flights and flight legs to test the above functionalities. Assume suitable data types for the attributes. You can use the CISE Oracle server or you can get the student version from Oracle website.

To Submit:

- E/R model and relation schema of your database.
- SQL queries used to create the tables.

### **Part 2: JDBC Connection and User Interface. (Due December 5, 2007)**

The main goal of this part is to connect user to the database, access tables, retrieve data into the application, and make updates to the database. The web interface will use JDBC connectivity. The interface should support the functionalities given in the requirements. There are many ways to implement a web-based database application, and one common way is using *JSP* and *Java Beans* (it's a nice thing to have on a resume, which is why we're going to use it). Each of the following four tools plays an important part in implementing a web-based application:

- *HTML* is the language that a web browser such as MS Internet Explorer or Mozilla understands. If you have never seen the inside of an HTML file, then you might want to begin by doing a little web surfing on how HTML works.

- A *JSP* web page is a web page with Java code embedded inside of it. Rather than containing only simple HTML, a JSP web page also has one or more sections that contain the output of a call to the method of a Java class. This allows a JSP web page to be dynamic (for example, it can output the result of a database query).
- A *Java Bean* is a little java class that is associated with a "control" on a web page. A "control" is an object like a button or a text box that can take input. The Java Bean is invoked when someone clicks on a control; whatever else it does, it typically outputs some HTML that tells the user the result of his or her action.
- *JDBC* is the way Java Bean can talk to the database.

You are free to choose the tools for this project (i.e PHP/ Perl/ ASP/ Servlets/ Oracle/ MySQL etc.). The website need not be fancy, just functional. Please refer to the following web page for the tools supported by the CISE server.

Check CISE web page for supported languages and drivers:

<http://www.cise.ufl.edu/help/database/oracle.shtml>

**Final Project Demo:** You will come up with a summary report which outlines your learning experience in the project and identify changes/modifications to the design. Include the contributions of both team members. They will meet with the TAs at a specific time (TBA) for a demo of the project.

## **Help**

If you have any questions, email the TAs. Emailed questions might be included in a project FAQ (available on the class website soon).