Program #4: Database-driven Web Application

Shuo Yang
Overview

• Given an application domain (see handout), design a relational data model for it and implement it in Oracle.

• Then build a web-based application on top of your database to support some basic information management tasks.
Three-tier Architecture

- **Web front-end**
  - web pages
- **Application server**
  - runs Tomcat web server
  - handles main business logic
- **Database back-end**
  - runs Oracle DBMS
  - a set of tables

Diagram:

- Web front-end
  - HTTP request
  - HTTP response
- Application server (Tomcat web-server)
- Database back-end
  - JDBC communication
DB Design

1. Requirement Analysis
2. Conceptual DB Design (ER model)
   - Satisfy all the requirements?
     - Y: Logical DB Design (relational model)
       - Normalization Analysis
         - Is this in 3NF?
           - Y: Implementation

CS460 Spring16, Program #4

Demo of a Simple JSP Application

Please excuse the simple appearance of this page, just click the button

list all employees
# Demo Application

## All Employees

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>EMPNAME</th>
<th>EMPSALARY</th>
<th>DEPARTMENT</th>
<th>BOSSNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice</td>
<td>75000</td>
<td>Management</td>
<td>null</td>
</tr>
<tr>
<td>2</td>
<td>Ned</td>
<td>45000</td>
<td>Marketing</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Andrew</td>
<td>25000</td>
<td>Marketing</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Clare</td>
<td>22000</td>
<td>Marketing</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Todd</td>
<td>38000</td>
<td>Accounting</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Nancy</td>
<td>22000</td>
<td>Accounting</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>Brier</td>
<td>43000</td>
<td>Purchasing</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>Sarah</td>
<td>56000</td>
<td>Purchasing</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>Sophie</td>
<td>35000</td>
<td>Personnel</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>Sanjay</td>
<td>15000</td>
<td>Navigation</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>Rita</td>
<td>15000</td>
<td>Books</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>Gigi</td>
<td>16000</td>
<td>Clothes</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>Maggie</td>
<td>16000</td>
<td>Clothes</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>Paul</td>
<td>11000</td>
<td>Equipment</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>James</td>
<td>15000</td>
<td>Equipment</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>Pat</td>
<td>15000</td>
<td>Furniture</td>
<td>3</td>
</tr>
<tr>
<td>17</td>
<td>Mark</td>
<td>15000</td>
<td>Recreation</td>
<td>3</td>
</tr>
</tbody>
</table>
How it works

7. Demo

An architectural diagram is given in Figure 1 to illustrate the components in the demo application and the connections between these components and the functionalities of them.

7.1. If you already started the tomcat server, now please shutdown.

```
$TOMCAT_HOME/bin/shutdown.sh
```

We will provide a very simple example that demonstrates how JS works. Generally, all the applications (including the welcome page) are placed under "$TOMCAT_HOME/webapps/ROOT," in order to replace the default pages with your application, you can simply rename the existing "ROOT" dir and create a new empty "ROOT" dir.

```
mv $TOMCAT_HOME/webapps/ROOT $TOMCAT_HOME/webapps/old_root
mkdir $TOMCAT_HOME/webapps/ROOT
```

To run the demo, follow the next four steps.

7.2. Copy the provided example, which is located at "$TOMCAT_HOME/webapps/460demo," to "$TOMCAT_HOME/webapps/ROOT."

```
cp -r $TOMCAT_HOME/webapps/460demo/* $TOMCAT_HOME/webapps/ROOT/
```

7.3. Launch the Application Server.

```
tomcat.sh start
```

7.4. Start the Web Front-End (WFE).

```
sh "webstart.sh"
```

To see the demo in action, go to "http://localhost:8080/460demo/index.html" in your browser.
First thing first

- Run the demo, make sure it works
- Form a team and report to TA
- Start reading the application domain and think about DB design