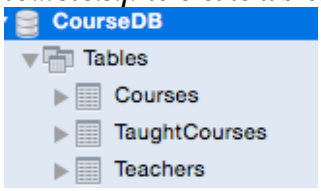
Test 1 JSTL, EL and Connection Pool

In this programming test, you are to create a very simple Web Application which allows students find a course by name or by teacher.

1. Data Source Configuration

Please create a database named as CourseDB in MySQL or any other relational database, then use the ***courses.sql*** to create tables and to insert sample data records.



As shown in the above figure, you will have three tables in your database, including COURSES, TEACHERS and TAUGHTCOURSES.

You should configure a Database Resource in your application server and access the dabatase in your Web Application.

2. Web Application Development

Your Application should be setup as follows:

1) ShowCourses

The ShowCourses Servlet will function as the student's primary interface to the Web Application. This controller will construct collections required to display course informations using the CourseList.jsp View.

You must make use of JavaBeans in your servlet when constructing the collections to be passed to the CourseList.jsp page.

2) CourseList.jsp

The CourseList.jsp will show a table listing courses. The table should contain the following columns:

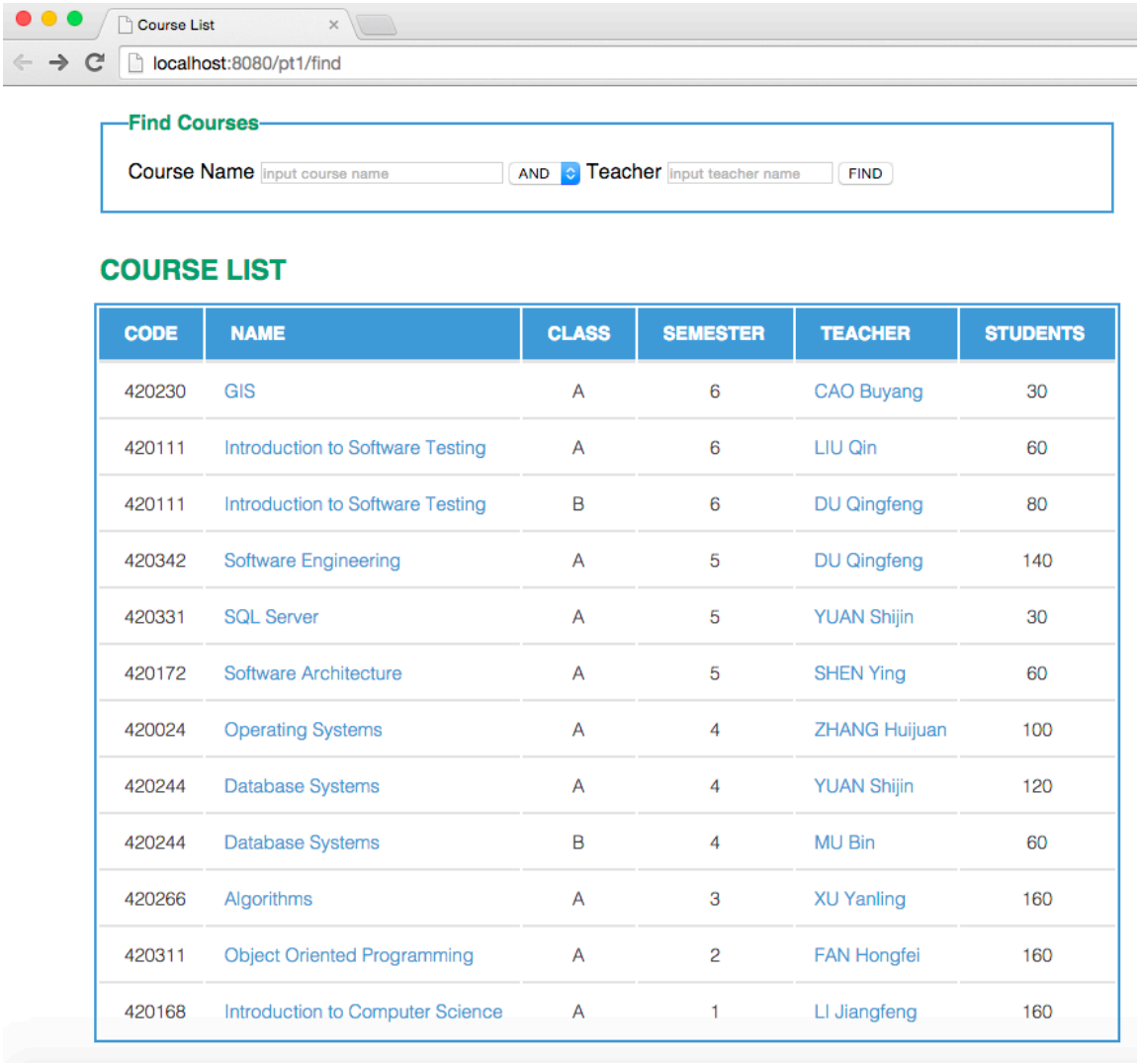
CODE, COURSE_NAME, TEACHER, SEMESTER, NUMBEROFSTUDENTS.

- By default, the CourseList.jsp will show all the courses available ordered by semester.
- It should also allow the user to find course by name or by teacher name.
- All teacher names and course names shown in the CourseList.jsp will link to ShowDetails servlet to retrive detailed information of the course or the teacher.

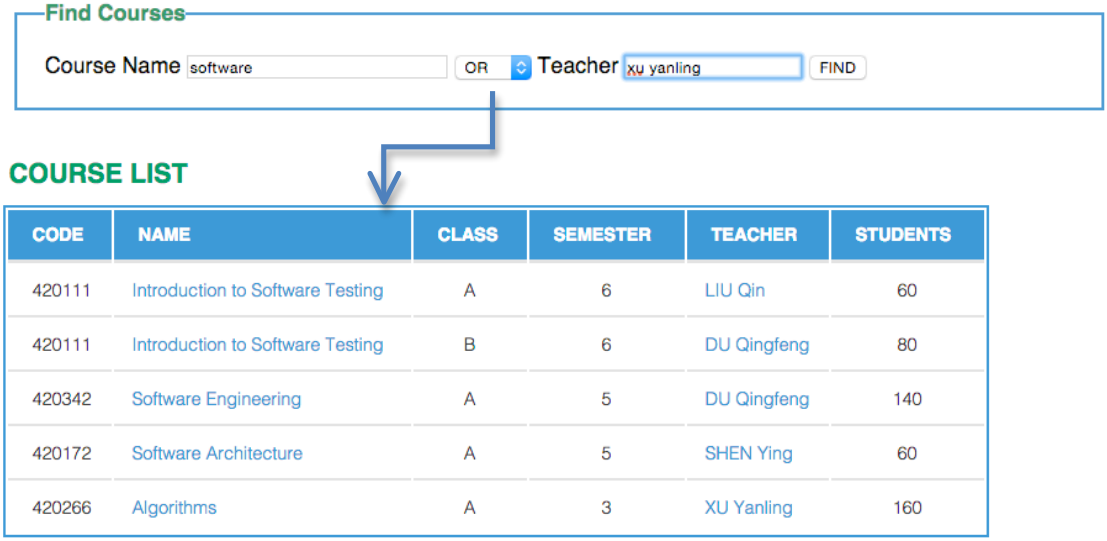
The CourseList.jsp page should not be directly accessible via the client's browser. The 'ShowCourses' should forward the request and response as well as data collections.

You are only allowed to use Beans, JSTL and EL to display data.

Below is an example of the page layout of CourseList.jsp

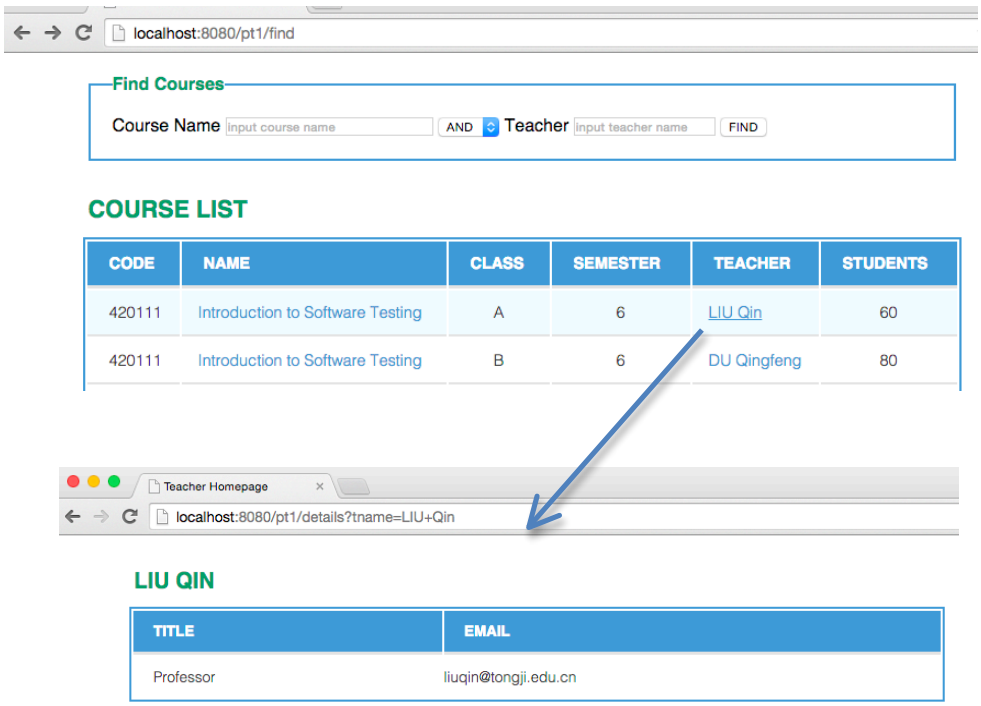


Find courses by name and/or teacher:



3) ShowDetails

The ShowDetails Servlet can retrive the teacher information or the course description from the database; and then use Teacher.jsp or Course.jsp to show the detailed.



4) CourseBean and TeacherBean

Please design the JavaBeans by yourself.

Notes:

- In order to save time, you can use ShowCourses.html given in the attached files as a template to construct your ShowCourses.jsp ; FindCourse.html given in the attached files can be included in your ShowCourses.jsp.
- A separate CSS file is also given, you may link the CSS in your jsp

```
<link href="<c:url value="/css/pagestyle.css"/> " rel="stylesheet">
```

- You may configure URL patterns for your servlets and JSPs to provide a more convinient access path.
- You needn't handle multiple pages output.
- Some SQL statetments might be used in this application for your reference:

```
SELECT TC.COURSE, TC.CLASSID, C.NAME, T.NAME, TC.SEMESTER, TC.NUMBEROFSTUDENTS
FROM TAUGHTCOURSES AS TC, COURSES AS C, TEACHERS AS T
WHERE TC.COURSE=C.CODE AND TC.TID=T.ID
ORDER BY TC.SEMESTER DESC;
```

```
SELECT TC.COURSE, TC.CLASSID, C.NAME, T.NAME, TC.SEMESTER, TC.NUMBEROFSTUDENTS
FROM TAUGHTCOURSES AS TC, COURSES AS C, TEACHERS AS T
WHERE TC.COURSE=C.CODE AND TC.TID=T.ID
        AND (C.NAME LIKE "%Software%" or T.NAME = "XU YANLING")
ORDER BY TC.SEMESTER DESC;
```

Hint: Please set the parameters dynamically.

- To add a hyperlink, you can use standard HTML markup <a href=””/a>

```
<td><a href="details?code=${course.code}" class="link"><c:out value="${course.name}"/></a></td>
```

- To avoid problems when passing paramter with special character, you may use <c:url> tag to encode your parameter.

```
<c:url value="details" var="teacherURL">
  <c:param name="tname" value="${course.teacher}"></c:param>
</c:url>
<td>
  <a href="<c:out value="${teacherURL}"/>" class="link">
  <c:out value="${course.teacher}"/>
  </a>
</td>
```

/********** THE END**********/