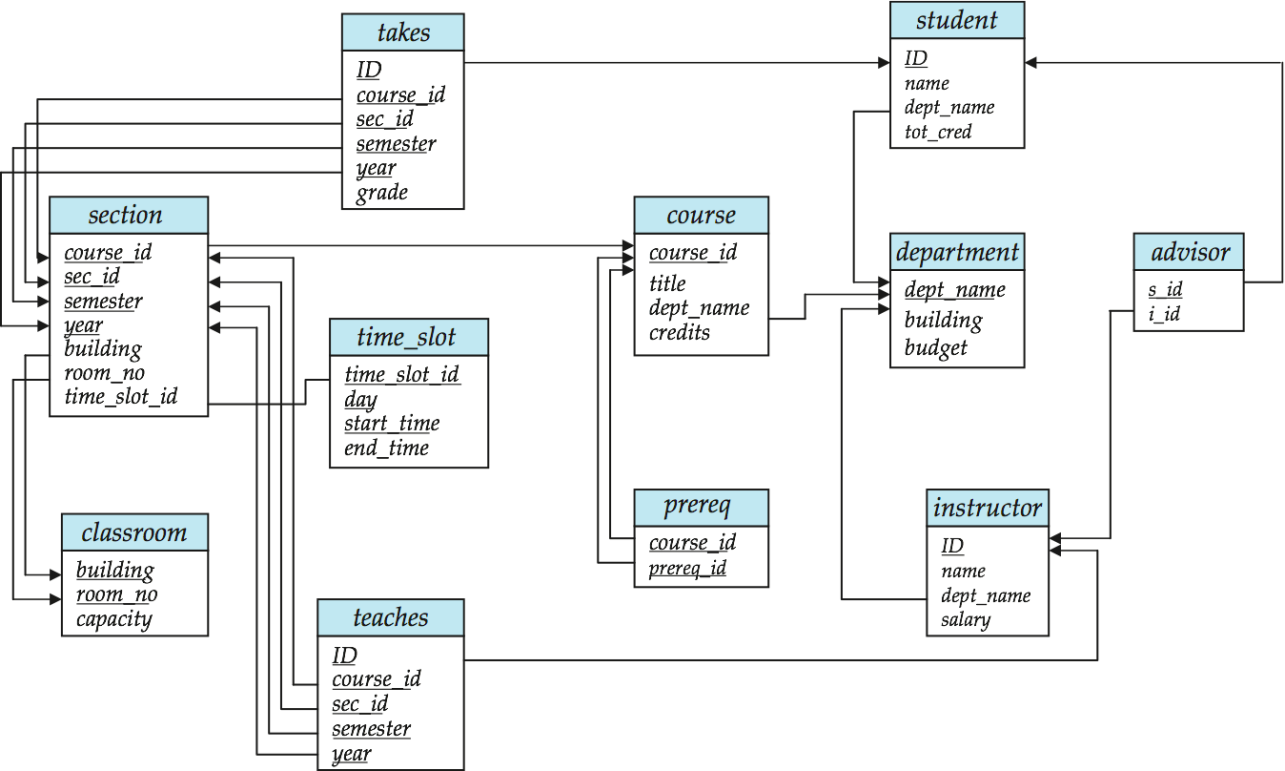
为保持后端代码风格统一，这里给出一个开发的建议思路

以开发功能“列举当前登录的老师的上课表(teaches)”举例：

1.开发teacherWelcome.html的功能。其中的动作listTeacherCourse未定义：

<a href="listTeacherCourse">查看课表</a><br>

这个功能想要展示登录的这个teacher的teaches表项（请参考数据库的关系图:

）

2.在web.xml中添加

-<servlet>

<servlet-name>TeacherCourseListServlet</servlet-name>

<servlet-class>servlet.TeacherCourseListServlet</servlet-class>

</servlet>

-<servlet-mapping>

<servlet-name>TeacherCourseListServlet</servlet-name>

<url-pattern>/listTeacherCourse</url-pattern>

</servlet-mapping>

定义动作listTeacherCourse的处理类是TeacherCourseListServlet.java

3.在sevlet包下新建TeacherCourseListServlet.java（代码见4），通过request.getSession().getAttribute("teaID");可以获取当前老师的ID。

（请注意这个请求R的状态：通过teacherLogin.html递交数据，负责处理的sevlet是TeacherLogin.java，请看TeacherLogin.java，里面有一个request.getSession().setAttribute("teaID", teacherID)。意味这这个请求被封装了输入的teacherID属性，可以通过teaID获取它）

在到达下一个页面teacherCourseList.jsp之前，这个请求R需要封装这个teacher对应的teaches信息，我们可以把它作为一个列表。

为了获取teaches列表的信息，需在中间逻辑层新建逻辑链接数据库，我们新建TeacherCourseDAO类。这个类和StudentCourseDAO类似，为实现指定功能至少需要提供返回一个列表(List)的方法，这个列表里面需要保存一系列对象，这种对象应该能够封装所有

关于teaches这个表的信息，我们把它叫做TeachesItem对象：我们新建一个TeachesItem类。

这样，我们就**建好了三个类（bean.TeachesITem、dao.TeacherCourseDAO、sevlet.TeacherCourseListServlet）**，但是都是空的。

4.根据以上思路，我们按照标准开发java从下而上的方式进行：

A.参考TakesItem.java和数据库关系图**填充TeachesItem.java类的属性和get set方法**

B.参考StudentCourseDAO**在TeacherCourseDAO链接数据库，从数据库获取对应的teacherI的其他的表项。填充到list方法中返回，运行成功后视情况在前后端增加动作选项和方法**

C.参考StudentCourseListSevlet.java，**根据3如下填充TeacherCourseListServlet.ja**va:

package servlet;

import java.io.IOException;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import bean.TeachesItem;

import dao.TeacherCourseDAO;

public class TeacherCourseListServlet extends HttpServlet {

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String teaID = (String) request.getSession().getAttribute("teaID");

List<TeachesItem> teachesList = new TeacherCourseDAO().list(teaID);

request.setAttribute("teachesList", teachesList);//请求封装teachesList的信息 可以通过引用jsp中${teachesList}获取它(注意第一项双引号里面是key，第二项是列表对象)（请参考studentCourseList.jsp）

request.getRequestDispatcher("teacherCourseList.jsp").forward(request, response);

}

}

5.接4，这个封装有teachesList的请求被带到了teacherCourseList.jsp，我们可以**如下新建teacherCourseList.jsp**：

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8" import="java.util.\*"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<title>教师课表信息</title>

<table align='center' border='1' cellspacing='0'>

<tr>

<td>teacherID</td>

<td>course\_id</td>

<td>sec\_id</td>

<td>semester</td>

<td>year</td>

</tr>

<c:forEach items="${teachesList}" var="teachesItem" varStatus="st">

<tr>

**<td>${teachesItem.ID}</td>**

<!--

...

根据TeachesItem中属性名填充（已经给出ID示例（加粗））

...

-->

</tr>

</c:forEach>

</table>

<div align='center'><a href="teacherWelcome.html">返回主界面</a></div>

至此完成。

点击教师界面查看课表，sevlet向DAO请求列表，封装到request之后jsp解引用这个请求里面的表项，可以显示这个teaID对应的teaches表项。