目录

[一、项目背景 1](#_Toc346053887)

[二、理论基础 1](#_Toc346053888)

[2.1数据库技术 1](#_Toc346053889)

[2．2B/S架构的WEB程序设计技术 1](#_Toc346053890)

[三、需求分析 3](#_Toc346053891)

[3.1系统用例图 3](#_Toc346053892)

[3.2系统功能体系图 3](#_Toc346053893)

[3.3目标 4](#_Toc346053894)

[3.4 用户的特点 4](#_Toc346053895)

[3.5 基本功能 5](#_Toc346053896)

[3.6 对性能的规定 5](#_Toc346053897)

[3.7 限制条件 5](#_Toc346053898)

[3.8系统难点分析 6](#_Toc346053899)

[四、系统总体软件结构图 6](#_Toc346053900)

[4.1MVC架构图 6](#_Toc346053901)

[4.2系统包图 7](#_Toc346053902)

[五、详细设计 9](#_Toc346053903)

[5.1 E-R图 9](#_Toc346053904)

[5.2 关系模式 9](#_Toc346053905)

[5.3 软件包类图 12](#_Toc346053906)

[5.3.1Action包图 12](#_Toc346053907)

[5.3.2Bean包图 12](#_Toc346053908)

[5.3.3Controller包图 13](#_Toc346053909)

[5.3.4 Dao包图 14](#_Toc346053910)

[5.4 部分功能算法和代码 15](#_Toc346053911)

[5.4.1 文件上传的流程图 15](#_Toc346053912)

[5.4.2分页技术 17](#_Toc346053913)

[5.4.3小组删除识别 18](#_Toc346053914)

[5.4.4老师部分功能 18](#_Toc346053915)

[六、讨论和展望 21](#_Toc346053916)

[6.1系统优点 21](#_Toc346053917)

[6.2 系统缺点 22](#_Toc346053918)

[6.3 系统设计和开发过程中的收获 22](#_Toc346053919)

# 一、项目背景

随着互联网技术的迅猛发展，网络给人们带来了很多便利，比如人们借助于网络进行相互交流、相互通信、共享信息、文件的上传下载等。同时互联网的发展也推动了教育的发展和变革，现在不少高校都有自己的作业提交系统，由于各种原因我们川大还没有统一的使用作业提交系统。为了更好的利用校园网的软硬件资源，实现学生作业的无纸化管理，让学生提交作业、查询作业信息更为方便，让老师审核学生作业更加的快捷，我们决定基于B/S结构开发一个简单又实用的在线提交作业系统。

# 二、理论基础

## 2.1数据库技术

PostgreSQL 是一种对象-关系型数据库管理系统(ORDBMS),也是目前功能最强大、特性最丰富和最复杂的自由软件数据库系统.它起源于伯克利(BSD)的数据库研究计划,目前是最重要的开源数据库产品开发项目之一, 有着非常广泛的用户。

PostgreSQL 是唯一支持事务、子查询、多版本并行控制系统、数据完整性检查等特性的唯一的一种自由软件的数据库管理系统.能在多平台下---包括Linux、FreeBSD和Windows等---运行,并且支持多语言的开发。

我们采用PostgreSQL8.4作为我们的数据库系统。

## 2．2B/S架构的WEB程序设计技术

B/S结构（Browser/Server结构）即浏览器和服务器结构。它是随着Internet技术的发展，对C/S结构的一种变化或者改进的结构。在这种结构下，用户工作界面是通过网页浏览器来实现，极少部分事务逻辑在前端（Browser）实现，但是主要事务逻辑在服务器端（Server）实现，形成所谓三层3-tier结构。这样就大大简化了客户端电脑载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本（TCO）。以目前的技术看，局域网建立B/S结构的网络应用，并通过Internet/Intranet模式下数据库应用，相对易于把握、成本也是较低的。它是一次性到位的开发，能实现不同的人员，从不同的地点，以不同的接入方式（比如LAN, WAN, Internet/Intranet等）访问和操作共同的数据库；它能有效地保护数据平台和管理访问权限，服务器数据库也很安全 。B/S结构最大的优点就是可以在任何地方进行操作而不用安装任何专门的软件。只要有一台能上网的电脑就能使用，客户端零维护。系统的扩展非常容易，只要能上网，再由系统管理员分配一个用户名和密码，就可以使用了。甚至可以在线申请，通过公司内部的安全认证（如CA证书）后，不需要人的参与，系统可以自动分配给用户一个账号进入系统。

三层客户/服务器(B/S)体系结构



客户端仅仅运行用户界面,浏览器（Web browser）用于数据的呈现 瘦客户；应用服务器运行业务逻辑和数据处理,应用服务器 ＋ Web server ；数据库服务器提供企业数据的集成和管理。

# 三、需求分析

## 3.1系统用例图



## 3.2系统功能体系图

作业提交系统

注册

登录

学生用户登录

教师用户登录

多次上传作业

添加注册课程

下载

删除

作业

数据库管理

查看下载作业

注册课程

添加管理小组

查看学生和小组

发布作业

通用模板

修改密码

查看个人资料

注销系统

查看教师通知

统计作业信息

发布通知

## 3.3目标

完成一套网上提交作业系统，练习体验Web数据库应用开发，最终希望可以把源代码提供给学校使用，能够完成简单的作业管理任务，讲源代码公开，希望有更多的同学参与进来将这个系统修改和完善。

## 3.4 用户的特点

用户对象是川大老师和同学，基于同学和老师对计算机技术有一定了解，可以很快熟悉系统的使用方法。也正是这个原因，我们将尽可能简单的设计我们的系统，把更多的主动权交给用户，以此来满足用户的需求。管理员对计算机很了解，不过我们希望管理员只做很少一部分，即对部分老师和学生的不规范行为矫正，如老师的作业没有在一定时间内删除，学生毕业后没有注销等。

## 3.5 基本功能

1、不同用户登陆进入不同的用户界面

2、学生、教师的信息修改和密码修改

3、学生、教师的注册注销

4、学生、教师可以多次上传文件覆盖上一次作业

5、学生注册课程

6、学生添加小组和加入小组

7、组长管理组员

8、学生查看作业和下载作业

9、学生、小组上传提交作业

10、学生查看教师通知

11、教师添加注册课程

12、教师布置作业

13、教师发布关于作业通告

14、教师查看选课情况和学生和小组信息

15、教师查看和统计提交作业情况（人数、日期等）

16、教师下载学生和小组作业

## 3.6 对性能的规定

要求系统支持PostGreSQL数据库

## 3.7 限制条件

1. 学生不可以进行教师页面
2. 教师也不可以进入学生页面
3. 每个学生登录进入之后只能看到自己注册课程的作业，不要恶意注册其他没有选的课程
4. 每个教师登录只对自己发布的作业进行管理，不可以对别的教师发布的作业进行管理
5. 不同的用户使用修改密码都只能修改自己的密码，无法修改他人的
6. 不同的用户使用个人数据功能时，也是只可以看到自己的信息不可以看到他们的信息
7. 学生上传作业的类型，大小受系统约束，

## 3.8系统难点分析

1、数据库的设计，推广本系统的话就要减少防止数据库的三类异常，同时兼顾效率，要在两者之间取舍。

# 四、系统总体软件结构图

## 4.1MVC架构图



控制器起到不同层面间的组织作用，用于控制应用程序的流程。它处理事件并作出响应。“事件”包括用户的行为和数据模型上的改变。

视图层能够实现数据有目的的显示（理论上，这不是必需的）。在视图中一般没有程序上的逻辑。为了实现视图上的刷新功能，视图需要访问它监视的数据模型（Model），因此应该事先在被它监视的数据那里注册。

“数据模型”（Model）用于封装与应用程序的业务逻辑相关的数据以及对数据的处理方法。“模型”有对数据直接访问的权力，例如对数据库的访问。“模型”不依赖“视图”和“控制器”，也就是说，模型不关心它会被如何显示或是如何被操作。但是模型中数据的变化一般会通过一种刷新机制被公布。为了实现这种机制，那些用于监视此模型的视图必须事先在此模型上注册，从而，视图可以了解在数据模型上发生的改变。

## 4.2系统包图



Controller包控制系统程序的流程，并对事物作出响应，其中的servlet接受JSP中提交的信息后，调用Module包中的Action类，然后Action再调用DaoImpl中的方法，DaoImpl完成了和数据库的连接，具有访问数据库的权利。这种结构，使得调用结构更加清晰，容易调试错误和以后对程序功能的扩展。JavaBean包定义了系统中的各种自定义数据结构。

# 五、详细设计

## 5.1 E-R图



## 5.2 关系模式

Student(student\_id,student\_name,student\_gender,grade,classes,student\_sdept,mail,password );

|  |  |  |  |
| --- | --- | --- | --- |
| Student Table | | | |
| 属性 | 类型 | 长度 | 约束条件 | |
| student\_id | varchar | 20 | 主码 | |
| student\_name | varchar | 20 | 不为空 | |
| student\_gender | varchar | 8 | ‘男’或‘女’ | |
| grade | varchar | 8 |  | |
| classes | varchar | 8 |  | |
| student\_sdept | varchar | 20 |  | |
| mail | varchar | 30 |  | |
| Password | varchar | 20 | 不为空 | |
|  |  |  |  | |

Teacher(teacher\_id,teacher\_name,teacher\_gender,mail,domain,password);

|  |  |  |  |
| --- | --- | --- | --- |
| Teacher Table | | | |
| 属性 | 类型 | 长度 | 约束条件 | |
| teacher\_id | varchar | 20 | 主码 | |
| teacher \_name | varchar | 20 | 不为空 | |
| teacher\_gender | varchar | 8 | ‘男’或‘女’ | |
| domain | varchar | 20 |  | |
| mail | varchar | 30 |  | |
| Password | varchar | 20 | 不为空 | |
|  |  |  |  | |

Homework(homework\_id,homework\_filename,message);

|  |  |  |  |
| --- | --- | --- | --- |
| Homework Table | | | |
| 属性 | 类型 | 长度 | 约束条件 | |
| homework\_id | varchar | 20 | 主码 | |
| homework\_filename | varchar | 20 | 不为空 | |
| message | varchar | 250 |  | |
| score | float |  |  | |
|  |  |  |  | |

Course(course\_id,course\_name,course\_sdept);

|  |  |  |  |
| --- | --- | --- | --- |
| Course Table | | | |
| 属性 | 类型 | 长度 | 约束条件 | |
| course\_id | varchar | 20 | 主码 | |
| course\_name | varchar | 20 | 不为空 | |
| course\_sdept | varchar | 20 |  | |
|  |  |  |  | |

Team(team\_id,team\_name,team\_monitorid);

|  |  |  |  |
| --- | --- | --- | --- |
| Team Table | | | |
| 属性 | 类型 | 长度 | 约束条件 | |
| team\_id | varchar | 20 | 主码 | |
| team\_name | varchar | 20 | 不为空 | |
| team\_monitorid | varchar | 20 |  | |
|  |  |  |  | |

Student\_team(student\_id,team\_id);(不完全遵守转换规则)

|  |  |  |  |
| --- | --- | --- | --- |
| Student\_team Table | | | |
| 属性 | 类型 | 长度 | 约束条件 | |
| team\_id | varchar | 20 | 外码 | |
| student\_id | varchar | 20 | 外码 | |
| Team\_monitorid | varchar | 20 |  | |
|  |  |  |  | |

Homework\_student(student\_id,homework\_id);(不完全遵守转换规则)

|  |  |  |  |
| --- | --- | --- | --- |
| Student\_team Table | | | |
| 属性 | 类型 | 长度 | 约束条件 | |
| homework\_id | varchar | 20 | 外码 | |
| student\_id | varchar | 20 | 外码 | |
| Course\_id | varchar | 20 |  | |
| submit\_date | Date |  | 不为空 | |
| Teacher\_id | varchar | 20 |  | |
|  |  |  |  | |

Homework\_team(student\_id,homework\_id);(不完全遵守转换规则)

|  |  |  |  |
| --- | --- | --- | --- |
| Student\_team Table | | | |
| 属性 | 类型 | 长度 | 约束条件 | |
| homework\_id | varchar | 20 | 外码 | |
| team\_id | varchar | 20 | 外码 | |
| Course\_id | varchar | 20 |  | |
| submit\_date | Date |  | 不为空 | |
| Teacher\_id | varchar | 20 |  | |
|  |  |  |  | |

Teacher\_course(teacher\_id,course\_id);

|  |  |  |  |
| --- | --- | --- | --- |
| Student\_team Table | | | |
| 属性 | 类型 | 长度 | 约束条件 | |
| teacher\_id | varchar | 20 | 外码 | |
| course\_id | varchar | 20 | 外码 | |
|  |  |  |  | |

Teacher\_course\_homeowrk(teacher\_id,course\_id);

|  |  |  |  |
| --- | --- | --- | --- |
| Student\_team Table | | | |
| 属性 | 类型 | 长度 | 约束条件 | |
| teacher\_id | varchar | 20 | 外码 | |
| course\_id | varchar | 20 | 外码 | |
| homework\_id | varchar | 20 | 外码 | |
| homework\_date | Date |  | 不为空 | |
|  |  |  |  | |

## 5.3 软件包类图

### 5.3.1 Action包图

本包属于Module，Action包接受Controller包中servlet的调用，对其隐藏细节而完成系统的各个功能，其中包括了上传作业（下载作业不作为一个类）、添加删除学生、查找课程和作业等类。



### 5.3.2 Bean包图

本包属于Module，对Action包和Dao包中的数据结构进行自定义，其中包括了学生信息、教师信息和课程信息等类。



### 5.3.3 Controller包图

本包属于Controller，对用户事件作出反应，用于处理浏览器中的表单和返回查询结果，通过本包调用Action包完成查询。



### 5.3.4 Dao包图

接收Action的调用，访问数据库完成一次数据库操作。



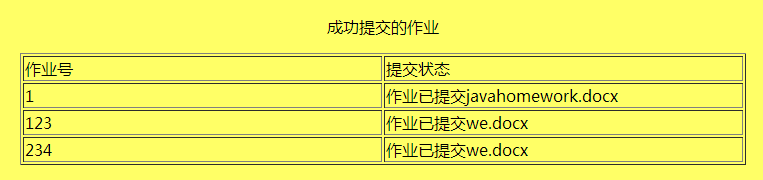
## 5.4 部分功能算法和代码

### 5.4.1 文件上传的流程图



效果截图：



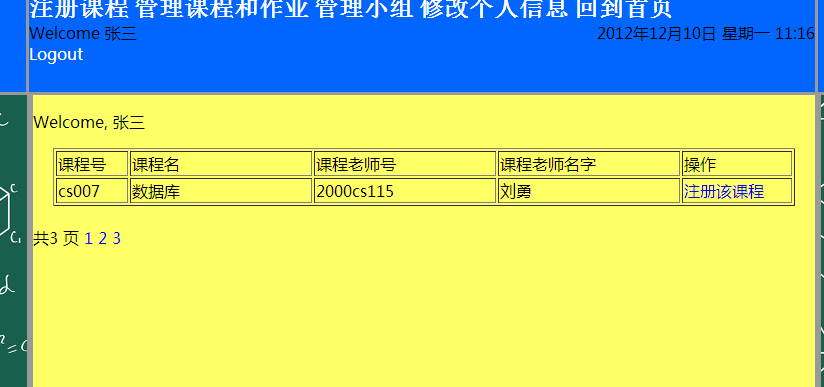


### 5.4.2分页技术

部分代码：

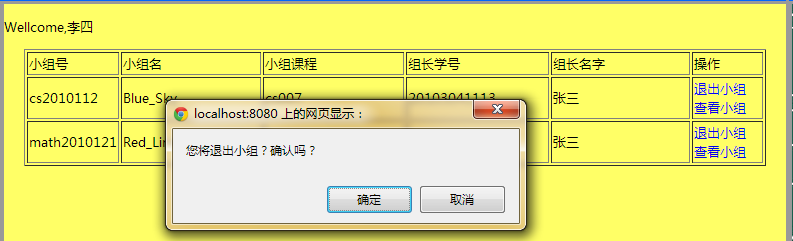
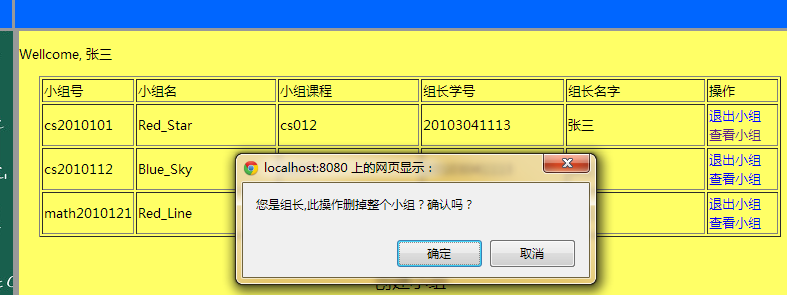
String sql = "select course.course\_id,course.course\_name,course.teacher\_id,course.teacher\_name from course left outer join student\_course on student\_course.student\_id=? and course.course\_id=student\_course.course\_idand course.teacher\_id=student\_course.teacher\_id where student\_course.student\_id !=? or student\_course.student\_id is null order by course.course\_id limit 1 offset ?";

效果图：(为展示效果我们将limit值设为 1，每页显示一项)



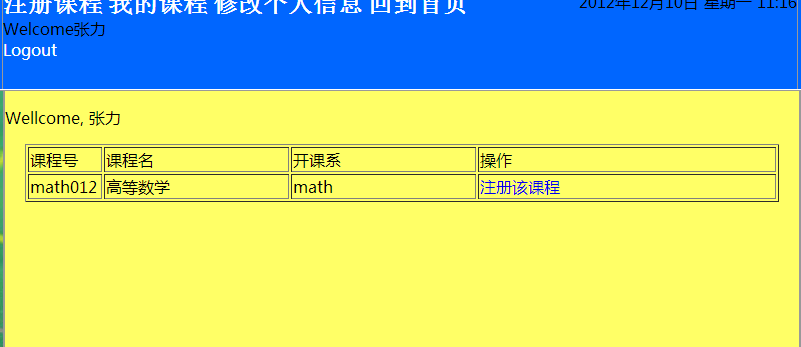
### 5.4.3小组删除识别

效果图小组组长进行此操作将会删除这个小组，而小组成员操作知识退出小组。



### 5.4.4老师部分功能

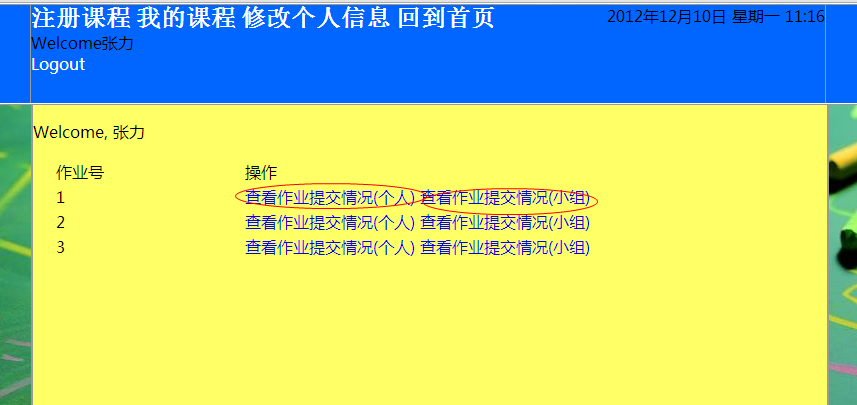
老师注册课程



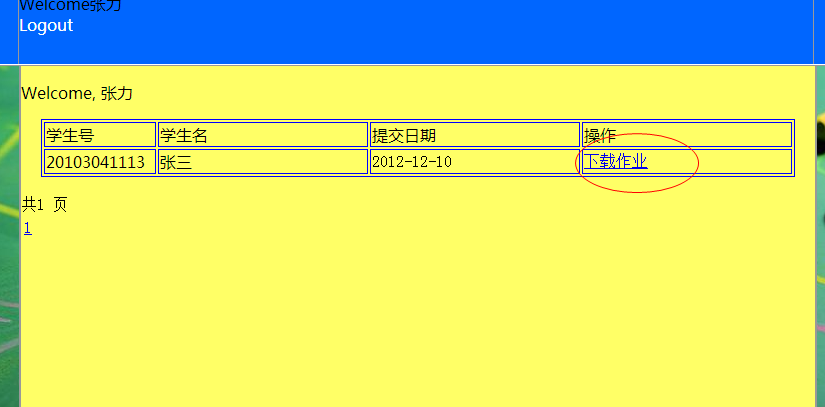
老师端的主界面：包含了主要功能，后期可以扩展



查看作业提交情况，分别以个人和小组来分类



查看学生个人作业，并且下载学生作业。



以小组为单位查看作业并下载



布置课程作业，可以覆盖前次上传。



# 六、讨论和展望

## 6.1系统优点

这是我们小组成员第一次合作，也是第一次做关于Jsp和数据库的应用系统，系统的基本功能已经实现，并且我们为以后的进一步开发业预留了接口和余地。在保证基本功能的同时我们也尽力为用户提供良好界面交互，也尽可能的接近可以应用的系统，为此我们添加了分页、重复上传和删除时识别权限等功能。

## 6.2 系统缺点

同样由于第一做此类系统，并且时间有点紧，所以系统还有很大缺点。功能不完善，我们在当初考虑要设计管理员，但是由于时间紧迫我们没有实现这方面的功能。教师和学生端的一些功能也有待进一步补充，教师端的发送群邮件没有完成。

还有就是系统界面，这是我们最后提交系统给助教老师检查前的最大担心，也是我们最大的遗憾，没有在最后做出一个交互良好的界面。

我们小组对于自己系统距离可以投入使用的差距很清楚，并且经过前面开发的经验和教训，如果有机会有时间的的话，可以根据上诉缺点一一改正，并且可以做到真正使用。

## 6.3 系统设计和开发过程中的收获

通过这次课程实践，我们对数据库的设计方法有了清晰的认识，即需求分析、概念设计、逻辑设计，对设计ER图和ER向关系模式的转换，这个过程中我们接受了书上的观点，为了提高代码效率和代码重用率我们允许了一些冗余。

通过系统开发，我们对SQL语句的应用更加熟练，在进行分页过程中，总是遇到想不到的问题，然后我们将Servlet之间的逻辑关系画出来，一边编写代码，一边解释代码意思，这样总算做出了“健壮”的分页。在学习查询优化之后，我们又试图将一些SQL语句进行优化，不过不能直观的对比两种方式的效果。此外还获得了网页开发和软件工程方面的一些知识和经验，比如CSS。

这次是我们小组成员的第一次合作，讨论决定这个项目后，我们在网上找资料、找视频，在图书馆看书学习，但效果很不好，写过一些代码，但随着后来任务的增多和小组成员的分工，开发进度越来越慢，后来由决定放弃，原来的开发模式，参考老师提供的项目代码和学习PPT后，我们决定按照MVC的模式从新开始工作。之后开发进度逐渐顺利起来，小组分工慢慢协调起来。整个项目坐下来，也从中体会了很多技术之外的东西，小组之间的交流和协调对于软件开发也是举足轻重。

最后，要感谢数据库老师对我们不耐其烦的指导，如果没有老师帮助我们不可能顺利开发。还要感谢助教对我们系统的检查以及对我们提出的建议和批评，让我们认识到我们的缺点。

我们将尽力推出更好的版本。