



江漢大學  
Jiangnan University

# 毕业论文(设计)

论文题目 物理论文网站的设计与实现

(英文) The design and implementation  
of the website of Physics

学 院 物理与信息工程学院

专 业 光信息科学与技术

姓 名 刘松林

学 号 122207102113

指导教师 柯旋

# 物理论文网站的设计与实现

学生：刘松林，物理与信息工程学院  
指导老师：柯旋，物理与信息工程学院

## 摘要

如今是网络时代，越来越多以互联网为基础的 WEB 应用出现在我们眼前，给我们的生活带来了许多的便捷。将互联网的 WEB 技术运用于论文管理工作中，将是一件非常有意义的工作。

本科毕业生在毕业论文环节中遇到的与导师沟通困难，论文选题、提交、评审和查阅过程繁琐，工作量大，交互性差，效率低下等问题。针对以上问题，本文就是利用 JAVA 技术和 MYSQL 数据库，在 JFinal 框架的基础上，搭建出了一个基于 WEB 的论文管理系统。

首先，本文深入探讨了论文管理系统目前的发展前景和实现方式，然后比较了 B/S(客户端/服务器)和 C/S(浏览器/服务器)两种模式各自的优缺点，总结了 B/S 模式的工作方式及优势，确立了本文的技术路线；在此基础上，深入调研了论文管理系统所需要的功能，设计出了符合需要的网站结构。然后通过编程实现了该系统，最后对系统进行了测试，测试结果说明，该系统功能完整实用，操作方便，能投入实际使用。

本文最后对系统在开发过程中遇到的问题和相应的解决方案进行了阐述，总结了本次研发中的心得体会。

**关键词：**B/S；WEB；Java；MYSQL；物理论文；管理系统

## Abstract

Today is the Internet era, more and more applications based WEB appear in front of us, to bring a lot of convenience to our life. It will be a very meaningful work to apply the WEB technology to the management of the paper.

Undergraduate graduates in the graduation thesis in the process of communication with the instructor difficulties, the paper topics, submission, review and inspection process cumbersome, heavy workload, poor interaction, low efficiency and other issues. For the above problems, this paper is based on JAVA technology and MYSQL database, based on the framework of JFinal, to build a WEB based paper management system.

First of all, this paper discusses the thesis management system for the current development prospects and realization ways, then compares the advantages and disadvantages of the two modes of B / S (client / server) and C / S (Browser / server) of the respective, summarizes the work and advantage of B / S mode, establish the technical route of this paper. On the basis of this, in-depth research of the thesis management system needs the function, designed in line with the needs of the structure of a WEB site. Then the system is realized by programming. Finally,

the system is tested. The test results show that the system is functional, practical and convenient, and can be put into practical use.

At the end of this paper, the problems encountered in the development process and the corresponding solutions are described, summed up the experience in this research and development.

Keyword: B/S;WEB; Java; thesis of physics ;management system

# 目录

第一章：绪论 .....	7
1.1 论文管理系统的背景 .....	7
1.2 研究意义 .....	7
1.2 研究现状 .....	8
第二章 系统需求分析 .....	10
3.1 功能性需求分析 .....	10
3.3 系统设计 .....	10
3.4 系统功能设计 .....	11
3.5 系统功能简介 .....	12
3.6 系统性能分析 .....	13
3.7 运行环境分析 .....	13
3.7.1 硬件需求 .....	13
3.7.2 软件需求 .....	13
3.8 本章小结 .....	13
第三章 系统的设计与实现 .....	14
4.1 系统结构与模块 .....	14
4.1.1 登录模块 .....	14
4.1.2 论文上传模块 .....	14
4.1.3 论题模块 .....	14
4.2 数据库设计 .....	14
4.2.1 逻辑设计 .....	14
4.2.2 物理结构设计 .....	15
4.2 项目文件结构 .....	17
4.3 主要功能分析与实现 .....	19
4.4.1 用户登录 .....	19
4.4.2 定时获取通知推送 .....	20
4.4.3 统计在线人数 .....	21
4.4.4 论文文件上传 .....	22
4.4.5 论文上传页面选项卡联动 .....	23

4.4.6 在线预览 PDF.....	24
4.4.7 教师评审 .....	26
4.4.8 论题的管理 .....	28
4.4.9 拦截器的配置 .....	28
4.4.5 第三方登录 .....	29
4.4 小结 .....	33
第四章 遇到问题和解决方案 .....	34
5.1 防止用户重复提交论文 .....	34
5.2 用户回话过期处理 .....	34
5.3 浏览器兼容性处理 .....	34
5.4 防止 SQL 注入攻击的处理方案 .....	35
第五章 总结 .....	36
参考文献 .....	37
致谢 .....	38

# 第一章：绪论

## 1.1 论文管理系统的背景

毕业论文设计是我系毕业生大学生涯中一个十分重要的组成部分，出色完成毕业设计的课题，是我们每个同学的期望。所以对毕业论文实行科学化的管理就显得非常有必要了。

当今的时代是网络时代，越来越多以互联网为基础的 WEB 应用出现在我们眼前，给我们的生活带来了许多的便捷，通过一根网线，我们就能与世界各地取得联系，第一时间获取到最新的信息，这就是我们所说的 WEB。如何通过网络将高校的毕业论文统一化管理，成为一个解决我校论文管理工作成为我校业务发展的一个新的课题。

目前我校的论文管理基本上是人工完成，学生通过见面会或者电话邮件的方式与导师取得联系，对于选题和论文上交，以及论文成绩统计等事项工作量大，效率低下等问题，都无法有效得到解决。

在此背景下，开发出一套论文管理系统就显得十分必要了，能有效解决我院毕业生论文管理的存在问题，显著提高论文工作进展的效率。本次研究是基于 java WEB 技术开展的，设计一个论文管理系统

## 1.2 研究意义

毕业设计是我们大学四年来非常重要的一个实践环节，是学生综合能力提高和展现的一种非常好的方式。本次研究是基于 Java WEB 技术开展的，设计一个论文管理系统，基本包含了我校毕业论文管理的业务需求。

系统包含了用户登录验证，学生论文上传，论文在线预览和下载，教师对论文进行评审，管理员可以对用户数据进行维护，具有修改和删除的权限。开发出此系统，能有效的减少我校毕业生在论文选题，论文提交和成绩统计等过程中的工作量。

同时本平台是一个基于 B/S 架构设计的，所以对用户电脑没有特殊要求，只需要接入互联网就可以完成论文的操作，真正做到了，足不出户，便可掌控全局的效果。

## 1.2 研究现状

从目前来看，很多大学已经将论文管理系统纳入教务管理的一部分，或者通过第三方网站来实现论文管理，或者通过教务系统通知来实现论文资料的发放工作。

从实现方式上来看，大体上分为 C/S（浏览器/服务器模式）架构和 B/S（客户端/服务器模式）架构。

传统的 C/S 模式，在前几年运用十分广泛，我们应该都非常熟悉，像我们每天都使用的 QQ 客户端就是最好的例子。运用 C/S 模式，客户端的处理能力得到了充分的发挥，因为大部分计算工作是在客户端完成，而且客户端响应比较快。但是，C/S 模式也存在一些缺点，首先每台机器上必须先安装客户端，才可以进行后续的使用，安装过程又会涉及到兼容性问题，而且客户端需要维护，如软件升级，客户端软件在升级过程中，要对每一台客户机上的客户端都进行一次升级操作，不仅花费人力物力，还十分繁琐，还有客户端在运行过程可能会受到病毒的侵扰，导致软件故障。

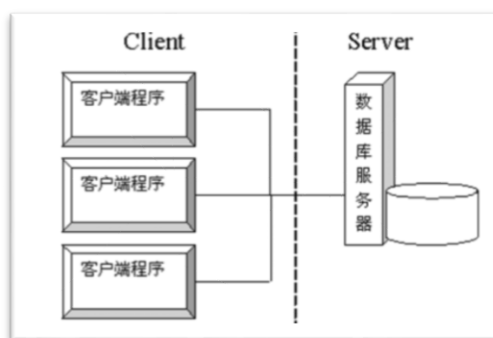


图 1-1: c/s 架构图

所以一种全新的架构就被提出来了，是一种和互联网紧密相关的模式，即 B/S 模式，我们每天都接触的浏览器成为客户端最主要的应用软件。这种模式将客户端进行了统一，将系统功能实现的核心部分放到了服务器上来完成，这也简化了客户端的开发，不必再为每种客户端都开发出一套客户端软件，客户机上只要安装一个浏览器，用户也只需要维护自己的浏览器，如常用的 Google Chrome 浏览器或 IE 浏览器[10]。

B/S 模式，也可以说属于一种三层架构。其主要利用了不断成熟的 WEB 技术和互联网的支撑，只通过浏览器就可以完全实现原来需要复杂客户端软件才能实现的强大功能，可以节省开发成本，是一种全新的软件构造技术，已经被越来越



多的企业所选择。但是对于图形处理要求比较高的软件，B/S 也是爱莫能助。说到 B/S，就离不开 HTTP（超文本传输协议）了，这是互联网上运用最广泛的一种网络通信协议，我们从浏览器的地址栏就可以看出 http://就是表示与服务器通信的协议，通过协议我们就能通过网络传输和发布我们的 HTML 页面，即我们常说的网页。

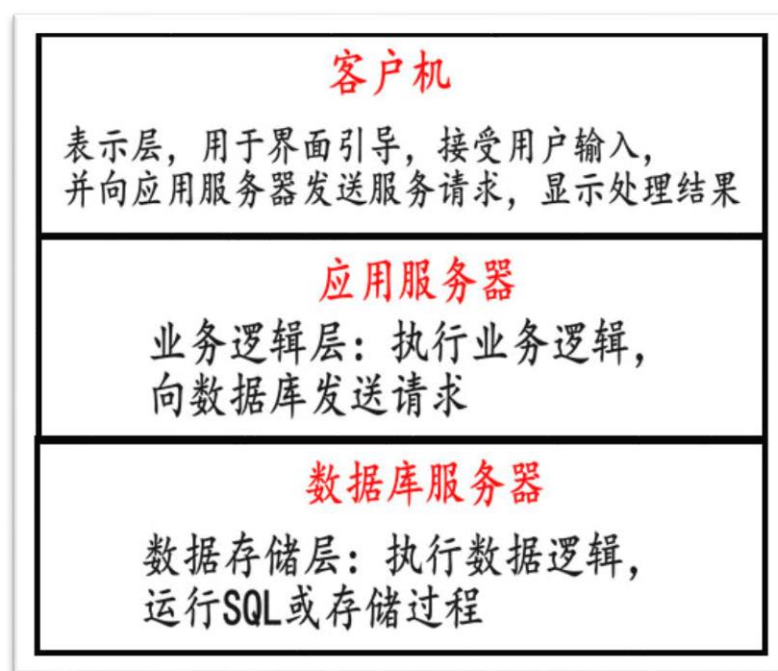


图 1-2：三层架构

三层架构[9]：

第一层是浏览器，即客户端，是用户与应用程序之间的交互的入口，处理极少本分业务逻辑，主要是表示层，用于用户页面的引导，接收用户的输入信息，向服务器发送请求，展现处理结果或者从服务器接收的数据。

第二层是 WEB 应用服务器，是业务逻辑层，可以接收用户的请求，获得用户的请求参数，根据请求参数执行相应的业务逻辑，包括数据的处理和数据的持久化操作。最后将处理结果返回给浏览器，呈现给用户。

第三层是数据库服务器，是数据持久层，负责存储数据，通过 SQL 或者存储过程可以对数据进行处理，执行数据逻辑。通常是通过应用服务器来发送 SQL 命令到数据库服务器来对获取数据或进行管理。

## 第二章 系统需求分析

### 3.1 功能性需求分析

本系统的最终用户为学生，教师和系统管理员。

主要实现以下功能点：用户登录，用户发布或者查看新闻通知，教师提交或者查看论文选题，学生根据选题提交论文资料（包括各个阶段的文档资料），用户查看自己论文的提交记录及成绩，用户在线预览论文资料，教师评阅学生论文，管理员维护论文资料。

系统的 UML 图如下：

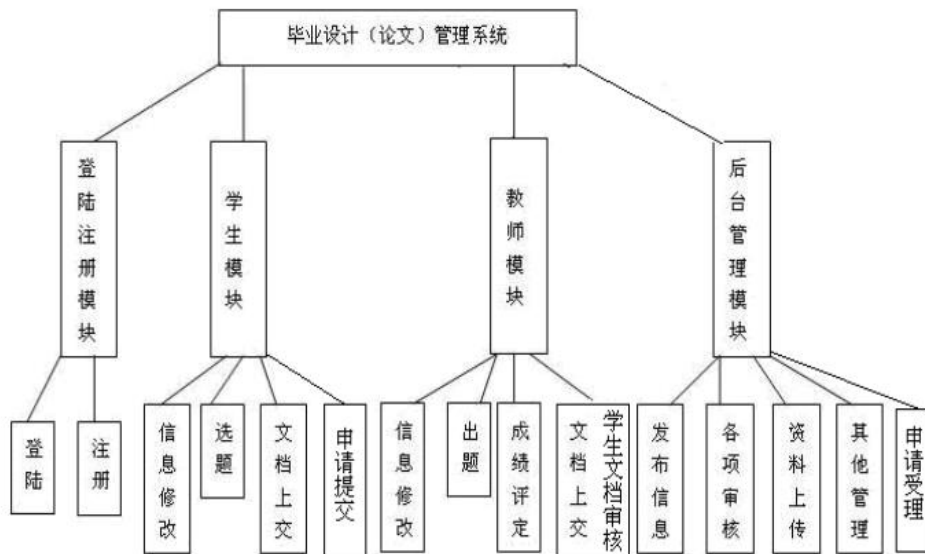


图 3-1：系统 UML 图

### 3.3 系统设计

系统是在 JFinal 框架的基础上搭建起来的，设计期间我参看了不少 JAVA WEB 开发方面的书籍[1]与期刊[4]，最后选定 JFinal 作为本系统的项目框架，一方面是想提升我学习新知识的能力，另一方面是该框架非常适合本项目的各项需求。JFinal 是一个三层系统, 用户界面层，业务操作层，数据访问层, 及对应 MVC 设计模式用的视图模型和控制器[12]。

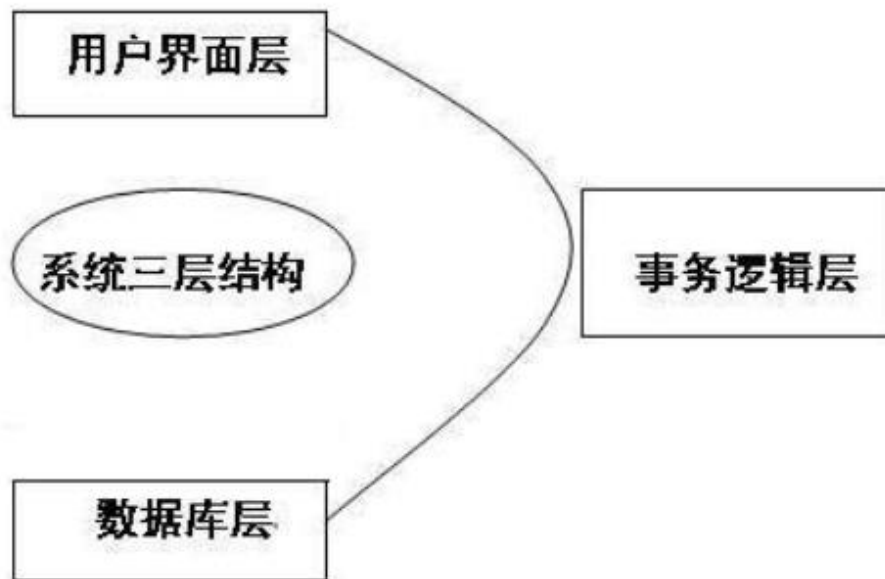


图 3-2 JFinal 三层系统

用户界面层,负责与用户进行交互,接收用户输入,展现数据的输出,并呈现给用户,起到视图的作用,业务操作层负责接收浏览器的请求,JFinal 中配置了一个全局过滤器,JFinal Filter 它会拦截所有 HTTP 请求,根据配置的地址映射将各个请求分配到相应的控制器进行处理,从请求中可以获取请求参数,控制器会调用数据访问层的方法,进行数据的持久化操作,最后控制器将结果返回给视图层。

### 3.4 系统功能设计

其工作流程为:学生用登录,查看自己论文上传的状态,如果没有上传,则可以进行论文上传和预览的操作,并且可以查看当前可以被选择的论题。教师登录系统可以发布论题,预览学生论文并审批学生论文。系统管理员可以对论文进行删除操作,前提是用户论文信息上传错误。

另外,系统还包括通知的功能,在登陆后的主页上就可以看到,通知会实时记录当前系统的动态,比如用户登录,用户上传,教师对论文进行了评审,或者也可以自由在输入框中发送消息,达到一个与用户互动的效果。

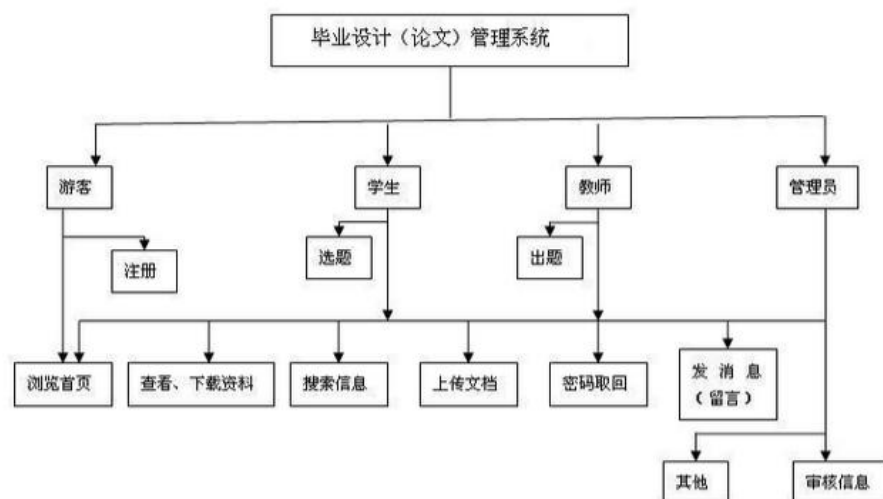


图 3-3 系统流程图

### 3.5 系统功能简介

本系统的主要功能主要是通过以下模块来实现：

1、系统登录模块：该模块负责处理不同角色的登录请求，并对用户权限进行分配

2、论文，论题模块主要包括以下：

(1) 论文，论题浏览（这里的信息包括网页通知、论文列表、论题选择、论文预览、论题浏览、相关论文、论题的发布等）：该模块负责查询数据，如论文上传者的信息，论文标题信息，论题的信息等。

(3) 论文，论题查询：该模块提供了论文，论题查询功能，输入论文上传人的姓名或者论题的名称，可以查询出符合查询条件的结果。

(4) 论文，论题管理：该模块负责论文数据，论题数据和用户数据的维护，管理员可以对所有信息进行维护，教师可以对论题进行维护，学生只有上传的功能，没有数据修改的权限。

3、论文上传模块：该模块负责学生用户的文档上传的功能。会将用户上传的论文资料归档到指定的目录位置，并记录文件位置到数据库。

4. 论题模块：该模块主要为教师和管理员角色提供论文论题的创建和修改功能，为学生提供论题的查询功能。

## 3.6 系统性能分析

系统设计应当考虑到稳定性, 并发性和系统的稳定性, 由于该系统使用的是 B/S 模式, 允许用户在 Internet 进行访问和操作, 访问是随时随地的, 故要求系统 24 小时不间断运行

并发性主要考虑学生在论文文档提交的过程中, 可能出现大量用户同时访问登录系统, 并同时提交的情况, 因此决定采用负载均衡的方法, 将访问请求平均分配到不同的服务器, 以降低系统的压力。

## 3.7 运行环境分析

### 3.7.1 硬件需求

- 应用服务器:

Window server 2003 双核 CPU, 内存 2G 以上, 硬盘 40G 以上  
接入公网, 拥有公网 IP

- 数据库服务器:

Window server 2003 双核 CPU, 内存 2G 以上, 硬盘 40G 以上  
接入公网, 拥有公网 IP

### 3.7.2 软件需求

- 数据库软件:

MYSQL-5.6.26-winx64 及以上版本

- 服务器软件:

Tomcat 6.0 及以上版本

JDK1.6 及以上版本

IE8 以上版本

## 3.8 本章小结

本章介绍了系统的需求, 包括系统在功能, 性能上面的需求, 以及要达到什么样的效果, 并给出了系统详细设计的框图, 为后面系统功能的实现提供相关依据。

## 第三章 系统的设计与实现

详细介绍了系统功能模块的实现过程，以及系统关键代码的展示。

### 4.1 系统结构与模块

#### 4.1.1 登录模块

此模块的主要控制器为 AdminController，负责处理用户的登录请求，Session 的创建的, 系统登出请求，获取通知消息，发送通知消息等功能。

#### 4.1.2 论文上传模块

此模块的主要控制器为 UploadController，负责处理论文上传的请求，并将论文存在项目指定的目录下，还包含了文件后缀名校验，文件大小限制等功能，防止用户上传对系统有危害的文件。

#### 4.1.3 论题模块

此模块的主要控制器为 TitleController，负责论文论题的增加，删除，修改，查询请求。

### 4.2 数据库设计

数据库系统为 MYSQL, 版本号为 5.6.26, 64 位, MYSQL 数据库作为开源数据库系统，拥有轻量级，功能强大，满足本系统的要求

数据库管理工具：Navicat for MYSQL 版本号 10.1.7

#### 4.2.1 逻辑设计

逻辑设计过程中参看了 MYSQL 权威指南[2]，对我设计出此结构提供了很大帮助。

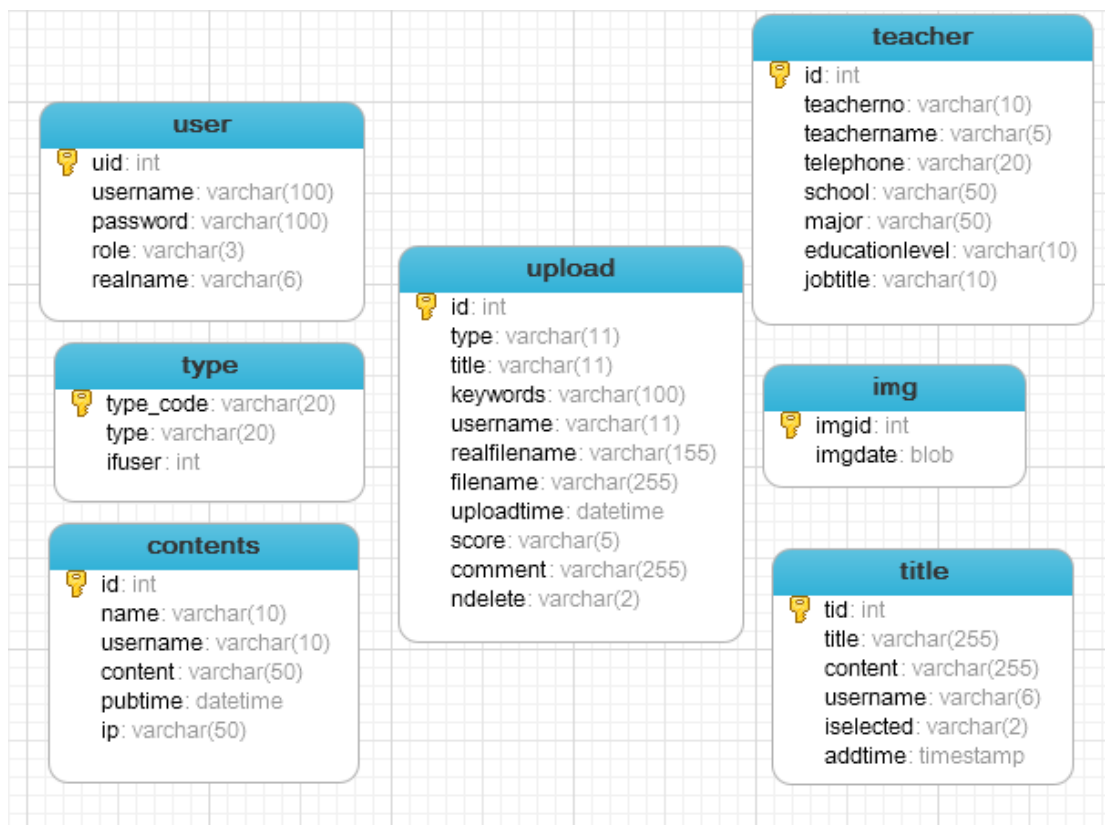


图 4-1 逻辑结构

#### 4.2.2 物理结构设计

数据库设计是程序设计中的重中之重，一个结构设计优良的数据库，应该表结构清晰，给部分各司其职，表之间关联得当。查询数据方面，通过表与表之间的关联，可以快速高效的查询到用户需要的数据，而且当数据量大的时候，其性能优势也能体现出来，不会轻易的数据表锁定的情况。

本系统的使用的表：

序号	表名	描述
1	user	记录用户名，密码及用户角色
2	Type	论文类型表
3	Contents	通知公告表
4	Upload	论文上传表
5	Teacher	教师表
6	mg	图片表
7	title	论题表
8	Role	角色表

数据表详细字段说明：

1.用户表（user）：

序号	字段名	字段类型	字段长度	描述
----	-----	------	------	----

1	uid	int	11	用户主键，唯一标识
2	username	varchar	10	用户名
3	password	varchar	10	用户密码
4	role	varchar	3	用户角色
5	realname	varchar	6	用户真实姓名

2. 论文类型表(Type)

序号	字段名	字段类型	字段长度	描述
1	type_code	varchar	20	类型编码，唯一标识
2	type	varchar	20	类型名
3	ifuser	int	2	是否启用

1. 通知记录表 (Contents)

序号	字段名	字段类型	字段长度	描述
1	id	int	11	通知主键，唯一标识
2	name	varchar	10	当前登录名
3	username	varchar	10	当前登录用户真实姓名
4	content	varchar	50	通知内容
5	pubtime	datetime	6	通知发布时间
6	ip	varchar	50	客户端 IP

2. 教师表 (Teacher)

序号	字段名	字段类型	字段长度	描述
1	id	int	11	主键，唯一标识
2	teacherno	varchar	10	教师工号
3	teachername	varchar	5	教师姓名
4	telephone	varchar	20	电话
5	school	varchar	50	学校
6	major	varchar	50	主修
7	educationlevel	varchar	10	教育水平
8	jobtitle	varchar	10	职称

3. 图片表 (img)

序号	字段名	字段类型	字段长度	描述
1	imgid	int	11	照片主键，唯一标识
2	imgdate	Blob	0	图片数据，字节流

4. 论文表 (Upload)

序号	字段名	字段类型	字段长度	描述
1	id	int	11	主键，唯一标识
2	type	varchar	11	论文类型
3	title	varchar	11	论文标题
4	keywords	varchar	100	关键字
5	username	varchar	11	上传人
6	realfilename	varchar	155	文件真实名称
7	filename	varchar	255	文件保存路径
8	uploadtime	Datetime	0	上传时间
9	score	Varchar	5	评分



10	comment	Varchar	255	评审意见
11	ndelete	varchar	2	是否删除

5. 论题表(title)

序号	字段名	字段类型	字段长度	描述
1	tid	int	11	主键，唯一标识
2	title	varchar	255	论文标题
3	content	varchar	255	描述
4	username	varchar	6	出题人
5	isselected	varchar	2	是否被选

6. 角色表(role)

序号	字段名	字段类型	字段长度	描述
1	id	int	11	主键，唯一标识
2	role	varchar	10	角色

## 4.2 项目文件结构

项目采用 Git 进行版本控制, 项目仓库设置在了 Github, 目前时间上最大的开源项目仓库, 每次进行了项目修改就将代码对送到项目仓库进行托管, 下次如果在其他电脑上进行开发的话, 只需要从远程仓库克隆代码或者获取更新到本地仓库, 进行代码修改后, 再将进行的修改推送到远程仓库, 完成一次提交, 这样极其方便的进行代码的管理。

项目的目录组织结构严格按照模块进行划分, 文件的命名依据其在项目中行使的职责确定, 基本上通过文件名就能判断出文件的作用。

下面依次展现了前端页面的组织结构和 java 类文件的目录结构:

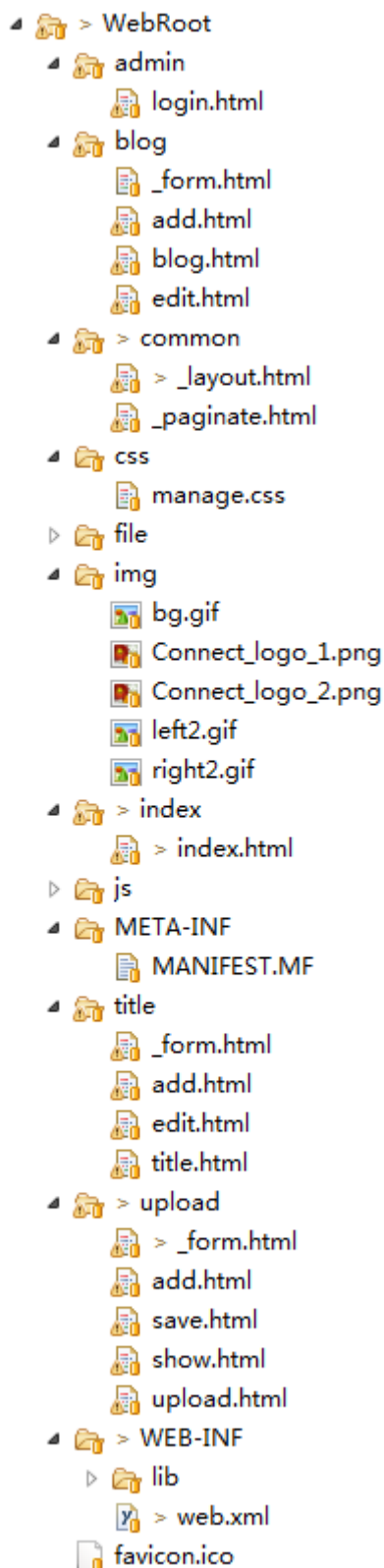


图 4-2 页面组织结构

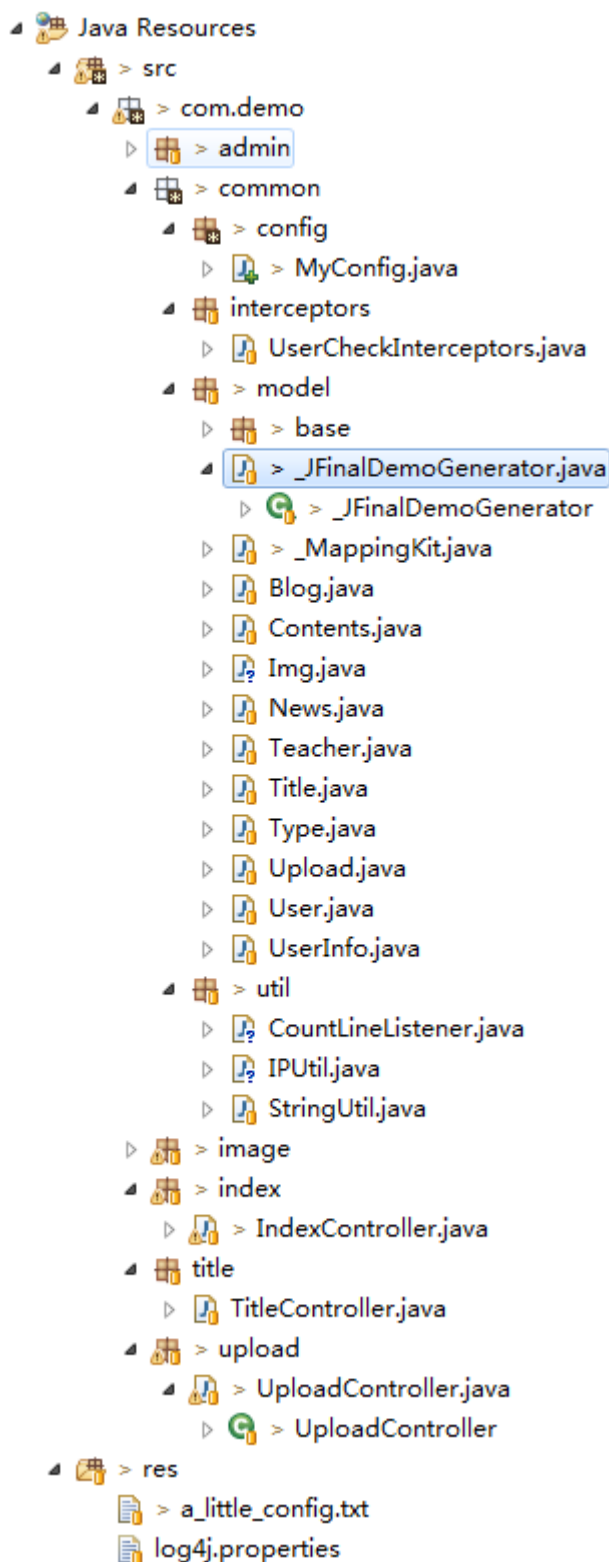


图 4-3 java 类目录结构

## 4.3 主要功能分析与实现

介绍了本系统主要的功能要点以及其实现方式,展现部分重要代码,由于采用了 JFinal 框架,所以大部分繁琐的数据库操作全部交给框架来完成,极大简化了项目的开发,最后额外介绍了第三方登录的功能。

### 4.4.1 用户登录

用户登录作为程序的入口,是本次设计中的重点,需要考虑的安全问题也比较多,用户登录验证的基本思路为,网页上用户输入的用户名和密码,点击提交后,后台控制器获取到用户输入,控制器访问数据库,执行一个用户名和密码配对的查询,如果查询到结果,证明该用户存在系统中,允许其登录,并记录用户登录的信息。

本系统出于安全考虑,对用户的密码进行了 MD5 加密,在获取到用户输入的密码后,先进行 MD5 加密,再将拿到的密文到数据库中进行比对,这样极大的提高了系统的安全性。

MD5 即消息摘要算法,几乎是不可逆的一种加密算法,是计算机领域广泛使用的一种广泛使用的散列函数,由于其具有任意长度的数据,算出的 MD5 值都是固定长度的,并且十分容易计算,而却哪怕对原数据只做一个字节的修改,其算出的 MD5 会发生非常大的变化,所以非常适合用于密码的保护[16]。即使数据库数据被人盗窃,拿到的密码也只是一串 MD5 加密后的字符串,很难逆向得到密码的原始

```
public void login() throws ClientProtocolException, Exception {
    String code = getPara("code");
    if (code == null || code.equals("")) {
        User user = getModel(User.class);

        String sql = "select * from user where username=" + "" + user.getUsername() + "" + " and password=" + ""
            + MD5.GetMD5Code(StringUtil.StringFilter(user.getPassword())) + "" + " and role=" + "" + user.getRole() + "";
        List<User> list = User.dao.find(sql);
        if (list.size() > 0) {
            String realname=Db.queryStr("select realname from user t where t.username= " + "" + user.getUsername() + "");
            Contents contents=new Contents();
            contents.setName(realname);
            contents.setContent("登录了系统");
            contents.setPubtime(new Date());
            contents.setUsername(user.getUsername());
            contents.setIp(IPUtil.getRemoteLoginUserIp(getRequest()));
            setSessionAttr("username", user.getUsername());
            setSessionAttr("realname", realname);
            setSessionAttr("role", user.getRole());
            contents.save();
            forwardAction("/index");
        } else {
            setAttr("error", "用户名或密码错误");
            render("/admin/login.html");
        }
    }
}
```

图 4-4 登录关键代码

#### 4.4.2 定时获取通知推送

当用户登录后，首页会显示当前的系统通知，以及发生的系统事件，这个消息推送是由一个定时器完成，当用户打开首页后，这个定时器就开始工作，每隔 5 秒就会 Post 请求一次服务器，当然这次请求不是异步请求，不会刷新这个页面，所以用户察觉不到每次请求的发生，但是数据会实时更新[20]，当服务器接收到请求后就会去数据库中查询通知消息，并按照发布时间顺序排列，取前面最新的 5 条返回给页面，页面再通过循环展现数据。

并且提供用户输入的功能，实时更新用户输入。

```
public void addContent(){
    String content=StringUtil.StringFilter(getPara("content"));
    Contents contents=new Contents();
    contents.setContent(content);
    contents.setPubtime(new Date());
    contents.setUsername((String)getSessionAttr("username"));
    contents.setIp(IPUtil.getRemoteLoginUserIp(getRequest()));
    contents.setName((String)getSessionAttr("realname"));
    contents.save();
    renderJson();
}
public void getContents(){
    renderJson("contents",Contents.dao.find("select * from contents t order by pubtime desc limit 0,5"));
}
```

图 4-5 服务端代码实现

```

//用户发送消息
$("#speak").click(function(){
    var content=$("#content").val().trim();
    if(content=="||content==null){
        alert("请输入内容");
        return false;
    }
    //异步请求服务器，增加通知内容
    $.ajax({
        url:"${base}/admin/addContent",
        type:"post",
        data:{content:content},
        datatype:"json",
        cache:"false",
        //请求成功后的回调函数
        success:function(obj){
            $("#content").val("");
        }
    });
    //异步请求服务器，拉取最新通知数据
    function update(){
        $.ajax({
            url:"${base}/admin/getContents",
            type:"post",
            datatype:"json",
            cache:"false",
            //请求成功后的回调函数，设置展现数据
            success:function(obj){
                var viewHTML="";
                for(var i=0;i<obj.contents.length;i++){
                    var timeStr=getDateDiff(getDateTimeStamp(obj.contents[i].pubtime));
                    viewHTML=viewHTML+ "<li>["+obj.contents[i].username+"@"+obj.contents[i].ip+"~]"+
                        obj.contents[i].name+": "+
                        obj.contents[i].content+
                        '-----'+timeStr+"</li>";
                }
                $("#contents").html(viewHTML);
            }
        });
    }
    update();
    //定时器，每隔5秒执行update函数，拉取通知消息
    var t1 = window.setInterval(update,5000);
}

```

图 4-6 前端页面代码实现

#### 4.4.3 统计在线人数

统计在线人数是通过监听器来实现的，创建了名为 CountLineListener 的监听器，使其实现 HttpSessionListener 接口，重写 sessionCreated()，sessionDestroyed() 方法，并且在 WEB.xml 中配置该监听器，当用户登录时，会创建一个 session，在创建时会被 CountLineListener 监听到，并执行 sessionCreated() 方法，在此方法中实现用户在线人数增加的业务操作，当用户退出登录，session 会被销毁，执行 sessionDestroyed() 方法，在此处进行在线人数减少的业务操作[17]

```

<listener>
    <listener-class>com.demo.common.util.CountLineListener</listener-class>
</listener>

```

图 4-7 配置 session 监听器

```

public class CountLineListener implements HttpSessionListener{
    /**
     * 创建session时调用
     */
    public void sessionCreated(HttpSessionEvent event) {
        System.out.println("创建session.....");
        ServletContext context=event.getSession().getServletContext();
        Integer count=(Integer)context.getAttribute("count");
        if(count==null){
            count=new Integer(1);
        }else{
            int co = count.intValue( );
            count= new Integer(co+1);
        }
        System.out.println("当前用户人数: "+count);
        context.setAttribute("count", count);//保存人数
    }

    /**
     * 销毁session时调用
     */
    public void sessionDestroyed(HttpSessionEvent event) {
        System.out.println("销毁session.....");
        ServletContext context=event.getSession().getServletContext();
        Integer count=(Integer)context.getAttribute("count");
        int co=count.intValue();
        count=new Integer(co-1);
        context.setAttribute("count", count);
        System.out.println("当前用户人数: "+count);
    }
}

```

图 4-8 监听器关键代码

#### 4.4.4 论文文件上传

论文上传是用户将本地文件上传到服务端的一个过程，要实现此过程，首先要将前端页面的表单元素的 `enctype` 属性设置为“`multipart/form-data`”[18]，这样当用户提交时，在服务端就能获取到文件的字节流，然后将文件字节流输入到服务器上一个新建的文件中，并记录文件的路径信息，方便文件预览或下载时可以找到该文件。

文件上传时，在控制器中对文件类型进行了过滤，当服务端获取到上传文件名时，对文件名进行分割处理，获取到以“.”开头的文件扩展名，将此扩展名与允许接受的扩展名进行对比，如果包含在其中，则允许上传操作，否则取消上传并提示用户该类型文件不允许上传。

文件上传时，在控制器中对文件大小进行了限制，当服务端获取到上传文件字节流时，对文件字节流进行单位换算，变成常用的兆字节（MB）单位，判断文件大小是否在规定大小之内，如果是，则进行上传操作，否则终止上传，并提示用户文件大小超过规定大小。

```
Upload upload=new Upload();
Contents contents=new Contents();
contents.setName((String)getSessionAttr("realname"));
File source = file.getFile();
String fileName = file.getFileName();
upload.setRealfilename(fileName);
contents.setContent("上传了"+fileName);
contents.setPubtime(new Date());
contents.setUsername((String)getSessionAttr("username"));
contents.setIp(IPUtil.getRemoteLoginUserIp(getRequest()));
String extension = fileName.substring(fileName.lastIndexOf("."));
if(extension.equals(".pdf")){
String prefix= "file";
fileName = generateWord() + extension;
String type = getPara("type");
String title = getPara("title");
String keywords = getPara("keywords");

JSONObject json = new JSONObject();
try {
    FileInputStream fis = new FileInputStream(source);
    File targetDir = new File(PathKit.getWebRootPath() + "/" + prefix + "/u/" + path);
    if (!targetDir.exists()) {
        targetDir.mkdirs();
    }
    File target = new File(targetDir, fileName);
    if (!target.exists()) {
        target.createNewFile();
    }
    FileOutputStream fos = new FileOutputStream(target);
    byte[] bts = new byte[300];
    while (fis.read(bts, 0, 300) != -1) {
        fos.write(bts, 0, 300);
    }
    fos.close();
    fis.close();
}
```

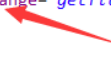
图 4-9 论文上传关键代码

#### 4.4.5 论文上传页面选项卡联动

在论文上传的表单页面，当用户点击了指导老师后，会自动在选择框中加载该指导老师对应的未被选择的论题。此处设计是由一个表单的 onChange 事件函数来实现，即当指导老师选项框的值发生改变时，将当前选择的指导老师的主键值通过异步请求发送到服务端，服务端通过该教师的主键值查找出该教师对应的论文信息，通过 json 格式对数据进行封装后传到页面，最后循环遍历此 json 数组，将数据赋值到对应的 html 标签上。所有的请求全部是通过 ajax 方式进行，

所以页面不会发生刷新操作。

```
<div>
  <label>指导老师</label>
  <select name="teacher" id="teacher" onchange="getTitle(this)" required="required">
</select>
</div>
</div>
```



监控值变化

图 4-10 前端页面代码

```
function getTitle(obj){
$.ajax({
  url:"${base}/upload/gettitle",
  type:"post",
  data:"id="+obj.value,
  dataType:"json",
  cache:false,
  success:function(obj){
    $("#title").html(" ");
    $.each(obj, function(index,title){ // 解析出data对应的Object数组
      $("#title").append('<option value="'+title.tid+' ">'+title.title+' </option>');
    });
  }
})
}
```

图 4-11 异步调用服务端方法

#### 4.4.6 在线预览 PDF

在线预览 pdf 是用户将论文资料上传到服务器后,可以重新从服务器调用论文资料,并在浏览器预览和下载的功能。首先用户在上传论文时,我们对论文的存放路径进行了记录,在用户预览时就可以通过这个路径找到论文,并提供在线预览。

预览时我使用了一个 PDFObject 的 JavaScript 库,使用其提供的函数来在线预览 pdf 文件,所以在页面上我们需要引入该 JavaScript 库。

```
<script src="${base}/js/pdfobject.js" type="text/javascript"></script>
```

图 4-12 引入 JavaScript 库

然后在页面新建一个 div 用于加载 pdf 的容器,命名 id 为 pdf,其中 id 为 results 的元素用于显示页面加载成功提示或页面加载失败时的提示

```
<div class="table_box">
  <div id="results" class="hidden"></div>
  <div id="pdf"></div>
</div>
```

图 4-13 标签定义

并定义 pdf 容器的样式,



```

.pdfobject-container {
    width: 100%;
    max-width: 1000px;
    height: 900px;
    margin: 2em 0;
}
.pdfobject { border: solid 1px #666; }
#results { padding: 1rem; }
.hidden { display: none; }
.success { color: #4F8A10; background-color: #DFF2BF; }
.fail { color: #D8000C; background-color: #FFBABA; }

```

图 4-14 样式定义

然后编写 JavaScript 脚本，调用相关 API 来加载，传入需要加载的 pdf 文件路径和加载 pdf 文件的容器的 id 值

```

//页面加载完成是调用
window.onload = function() {
    //定义配置项
    var options = {
        pdfOpenParams: {
            navpanes: 0,
            toolbar: 0,
            statusbar: 0,
            view: "FitV"
        }
    };
    //新建一个PDFObject对象,传入文件路径,加载容器,配置项
    var myPDF = PDFObject.embed("${base}${upload.filename}", "#pdf", options);
    var el = document.querySelector("#results");
    el.setAttribute("class", (myPDF) ? "success" : "fail");
    //加载提示
    el.innerHTML = (myPDF) ? "成功加载预览" : "噢，文件似乎打不开啊";
};

```

图 4-15 加载 pdf 的脚本



图 4-16 页面显示效果

#### 4.4.7 教师评审

教师可以对学生上传的论文进行评审，一个评审过程包括论文成绩的判定，分为优，良，中，差四个等级，以及评审意见的填写，每篇论文只允许一次评审。

在加载论文预览的页面时，我将该论文是否评审的标识传到页面，通过该标识在判断评审按钮的开放权限。

审阅意见：

```
<#if upload.comment ??>
  ${upload.comment!''}
<#else>
  <#if role!="1">
    <a href="#Comentmodal" id="modaltrigger">未审批</a>
  </#if>
</#if>
```

图 4-17 页面判断

加载评审框是我采用了模态框的加载方式，以提高用户体验。

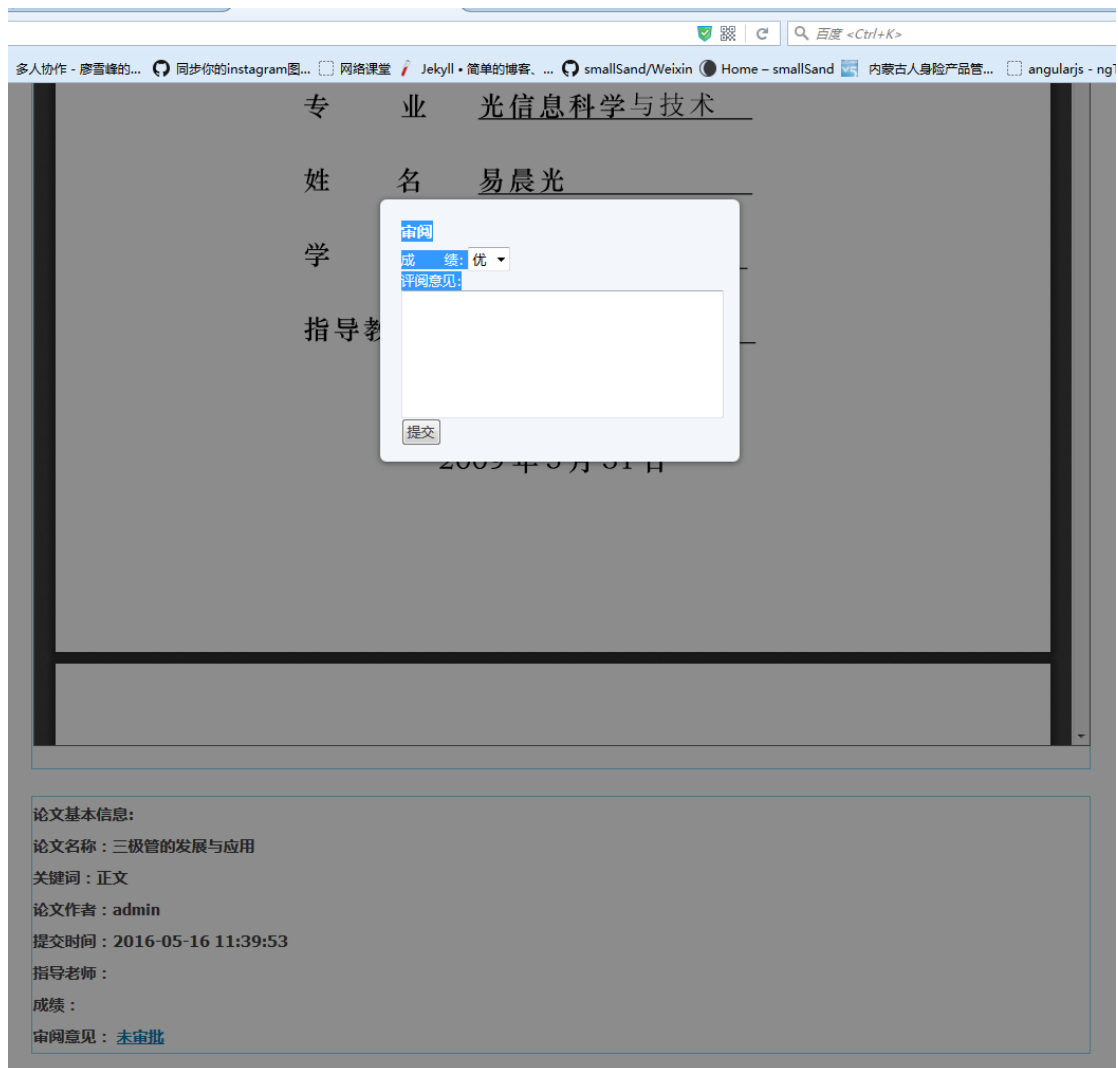


图 4-18 效果图

该模态框需要加载两个 JavaScript 库

```
<script src="{base}/js/jquery.leanModal.min.js" type="text/javascript"></script>
<script src="{base}/js/jquery-1.4.4.min.js" type="text/javascript" ></script>
```

图 4-19 需要的 JavaScript 库

加载过程：首先在页面上添加评审框的页面代码，并设置其不显示，即 display 属性为 none

```
<div class="modal" id="ComentModal" style="display: none;">
<h1>审阅</h1>
<form id="dataform" name="dataform" method="post"
    action="{base}/upload/approve">
    <input type="hidden" name="id" value="{upload.id}" /> <label>成&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~绩:
    </label> <select name="score">
        <option value="优">优</option>
        <option value="良">良</option>
        <option value="中">中</option>
        <option value="差">差</option>
    </select><br> <label>审阅意见:</label>
    <textarea rows="6" cols="45" name="comment" id="comment" style="resize: none" required maxlength="100"></textarea>
    <br> <input type="submit" name="submitbtn" id="submitbtn" value="提交">
</form>
</div>
```

图 4-20 模态框页面代码

为触发模态框的按钮增加 一个 name 属性值和 id 属性值，如图 4-14 所示  
编写触发事件

```
$(function() {
    $('#Comentmodal').submit(function(e) {
        return true;
    });
    $('#modaltrigger').leanModal({
        top : 110,
        overlay : 0.45,
        closeButton : ".hidemodal"
    });
});
```

图 4-21 模态框脚本

4.4.8 论题的管理

论题管理包括学生查看论题，教师创建修改论题，管理员管理论题的功能，创建的论题会在上传论文的论题选择下拉框中进行加载，并且选择过的论题会显示为已选择。

id	标题	描述	出题人	状态	操作
1	碳纳米管对微波吸收的应用和发展	碳纳米管对微波吸收的应用和发展	周迪	未选择	
2	马赫-曾德尔干涉实验研究	马赫-曾德尔干涉实验研究	乔燕	已选择	
3	新能源及新能源汽车的发展	新能源及新能源汽车的发展	李星	已选择	
4	基于复杂空间的演化博弈	基于复杂空间的演化博弈	朱小龙	已选择	
5	ZnO纳米线/P-GaN薄膜异质结紫外LED的研究进展	ZnO纳米线/P-GaN薄膜异质结紫外LED的研究进展	涂亚芳	已选择	
6	聚芴结构与光电性质的第一性原理研究	聚芴结构与光电性质的第一性原理研究	郑广	已选择	

图 4-22 系统界面

4.4.9 拦截器的配置

本项目配置了一个全局拦截器，用于当用户会话过期时，如果继续访问会阻止访问，并跳转到用户登录页面，但是不会对例外的 URL 请求进行拦截

要实现一个拦截器，首先要实现 Interceptor 接口，并重写 intercept 方法，在此方法中实现控制跳转的业务逻辑

```

public class UserCheckInterceptors implements Interceptor{
    private static final Set<String> excludedActionKeys = new HashSet<String>();

    public static void addExcludedActionKey(String actionKey) {
        excludedActionKeys.add(actionKey);
    }

    @Override
    public void intercept(Invocation inv) {
        //添加拦截例外
        addExcludedActionKey("/admin/login");
        addExcludedActionKey("/admin/userinfo");
        addExcludedActionKey("/admin/logout");
        addExcludedActionKey("/");
        Controller controller=inv.getController();
        if (!excludedActionKeys.contains(inv.getActionKey())) {

            if( controller.getSessionAttr("username")==null){
                controller.setAttr("error", "回话已结束");
                controller.render("/admin/login.html");
            }else{
                inv.invoke();
            }
        }else{
            inv.invoke();
        }
    }
}

```

图 4-23 拦截器实现代码

#### 4.4.5 第三方登录

第三方登录是现在网站常用的功能之一，通过第三方登录，可以实现一个账号在多个网站共同使用的好处，快速登入系统，并且免去了用户注册的一系列繁琐的功能。

第三方登录是属于额外的开发功能，也让我学习了一下实现第三方账号登录的基本流程和开发过程。

这里的第三方是指通过腾讯的 QQ 号登入到此系统中，其中包括了一系列的认证过程，当认证通过，获取到授权码，通过授权码就可以获取到用户在腾讯的基本账号信息[13]。



图 4-24 基本流程图

开发前的准备：在自己网站的首页放置第三方登录的入口

**用户登录：**

用户名:

密 码:

角 色: 学生

其他账号登录:  

图 4-25 登录入口

然后将自己的网站提交给 QQ 互联审核，通过审核后或得到自己的 APPID 和 APP KEY

QQ 互联的第三方登录采用的是 OAuth2.0 认证，又称 WEB Server Flow，适用于需要从 WEB server 访问的应用[19]。其基本流程如下：

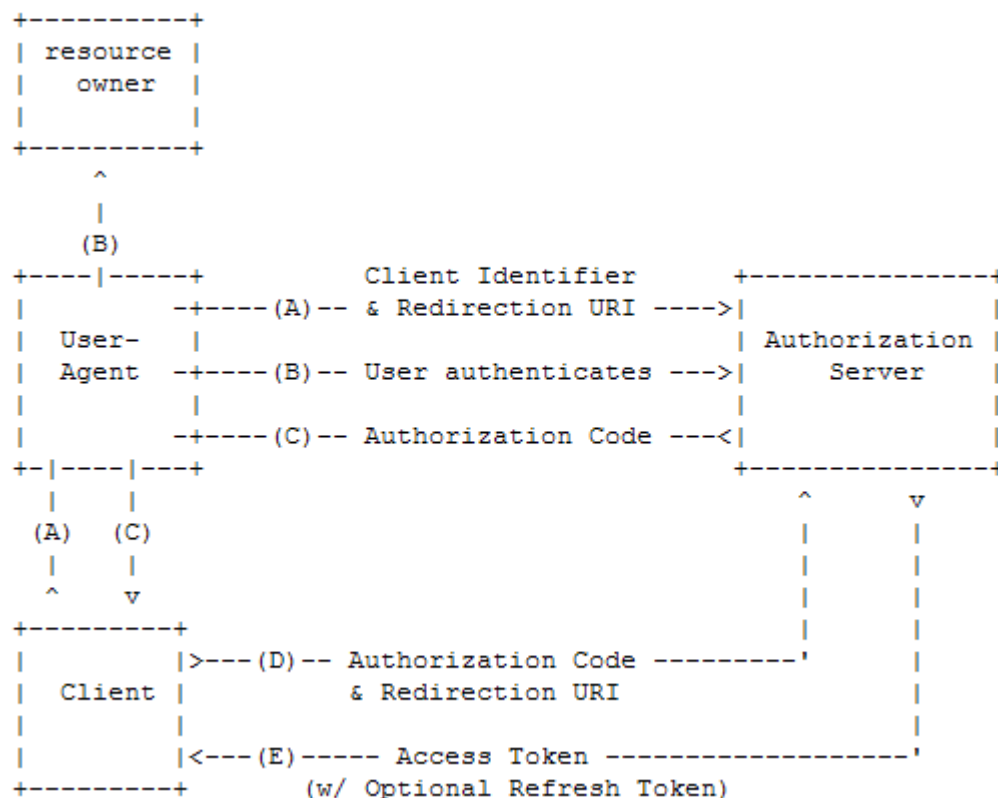


图 4-26 OAuth2.0 流程

在此项目中，我们只需要做两步操作：

1. 首先获取到授权码 Authorization Code

通过 GET 请求 <https://graph.qq.com/oauth2.0/authorize> 这个地址，请求参数传入第一步获取到的 APPID 和 APPKEY，以及回调地址和 Client 状态值

请求成功后会跳转到回调地址,并获得一个 code 值,即 Authorization Code

## 2. 通过授权码获取授权令牌 Access Token

通过 GET 请求 `https://graph.qq.com/oauth2.0/token`, 请求参数为授权类型, APPID, APPKEY, 第一步获取到的 code 值, 以及第一步中的授权码

请求成功后会获取到授权令牌及其有效期(默认三个月)

## 3. 获取到用户的 OPENID

通过 GET 请求 `https://graph.qq.com/oauth2.0/me`, 传入上一步获得的 Access Token,

请求成功后, 获得用户的 OPENID

## 4. 调用获取用户资料的 API

调用相关的 API 需要 GET 请求 `https://graph.qq.com`, 加上相应的 API 接口方法, 传入上面获得的 OPENID, Access Token 和开发者的 APPID, 就可以获得用户的相关信息, 本项目目前只使用了 `get_user_info` 的 API 接口, 具体请求示例为:

```
https://graph.qq.com/user/get_user_info?  
access_token=*****&  
oauth_consumer_key=12345&  
openid=*****
```

图 4-27 获取用户信息请求方法

响应获取的数据为

```

{
  "ret":0,
  "msg":"",
  "nickname":"Peter",
  "figureurl":"http://qzapp.qlogo.cn/qzapp/111111/942FEA70050EEAFBD4DCE2C1FC775E56/30",
  "figureurl_1":"http://qzapp.qlogo.cn/qzapp/111111/942FEA70050EEAFBD4DCE2C1FC775E56/50",
  "figureurl_2":"http://qzapp.qlogo.cn/qzapp/111111/942FEA70050EEAFBD4DCE2C1FC775E56/100",
  "figureurl_qq_1":"http://q.qlogo.cn/qqapp/100312990/DE1931D5330620DBD07FB4A5422917B6/40",
  "figureurl_qq_2":"http://q.qlogo.cn/qqapp/100312990/DE1931D5330620DBD07FB4A5422917B6/100",
  ,
  "gender":"男",
  "is_yellow_vip":"1",
  "vip":"1",
  "yellow_vip_level":"7",
  "level":"7",
  "is_yellow_year_vip":"1"
}

```

图 4-28 返回的用户数据

响应数据格式为 JSON, 使用反序列化工具将其转为字符串。

以上的全部过程全部使用 JAVA 完成, 关键代码如下:

```

CloseableHttpClient httpClient = HttpClients.createDefault();
URI uri = new URIBuilder().setScheme("https").setHost("graph.qq.com").setPath("/oauth2.0/token")
    .setParameter("grant_type", "authorization_code").setParameter("client_id", "101307810")
    .setParameter("client_secret", "ca3540429db64f88301469061d2d98b8").setParameter("code", code)
    .setParameter("redirect_uri", "http://lwgl.oschina.mopaasapp.com/admin/login").build();
HttpGet httpGet = new HttpGet(uri);
CloseableHttpResponse response = httpClient.execute(httpGet);
// access_token=YOUR_ACCESS_TOKEN&expires_in=3600
try {
    if (response.getStatusLine().getStatusCode() == 200) {
        HttpEntity entity = response.getEntity();
        String result = EntityUtils.toString(entity);
        String access_token = result.substring(13, 45);
        HttpGet httpGet1 = new HttpGet("https://graph.qq.com/oauth2.0/me?access_token=" + access_token);
        CloseableHttpResponse response1 = httpClient.execute(httpGet1);
        String result1 = EntityUtils.toString(response1.getEntity());
        result1 = result1.substring(10, result1.length() - 3);
        JSONObject json = JSONObject.fromObject(result1);
        String openid = json.getString("openid");

        URI uri1 = new URIBuilder().setScheme("https").setHost("graph.qq.com")
            .setPath("/user/get_user_info").setParameter("access_token", access_token)
            .setParameter("oauth_consumer_key", "101307810").setParameter("openid", openid).build();
        CloseableHttpResponse response2 = httpClient.execute(new HttpGet(uri1));
        String result2 = EntityUtils.toString(response2.getEntity());
        JSONObject jsonObj = JSONObject.fromObject(result2);
        String ret = jsonObj.getString("ret");
    }
}

```

图 4-29 QQ 互联关键代码



分组名称	适用平台	API接口	功能说明	备注
访问用户资料	网站、移动	get_user_info	获取登录用户的昵称、头像、性别	
访问用户QQ会员信息	网站、移动	get_vip_info	获取QQ会员的基本信息	需要申请
		get_vip_rich_info	获取QQ会员的高级信息	
访问我的空间相册	网站、移动	list_album	获取用户QQ空间相册列表	需要申请
		upload_pic	上传一张照片到QQ空间相册	
		add_album	在用户的空间相册里，创建一个新的个人相册	
		list_photo	获取用户QQ空间相册中的照片列表	
访问我的腾讯微博资料	网站、移动	get_info	获取登录用户在腾讯微博详细资料	
分享内容到我的腾讯微博	网站、移动	add_t	发表一条微博	
		del_t	删除一条微博	
		add_pic_t	发表一条带图片的微博	
		get_repost_list	获取单条微博的转发或点评列表	
获得我的微博好友信息	网站、移动	get_other_info	获取他人微博资料	
		get_fanslist	我的微博粉丝列表	
		get_idollist	我的微博偶像列表	
		add_idol	收听某个微博用户	
		del_idol	取消收听某个微博用户	
访问我的财付通信息	网站、移动	get_tenpay_addr	在这个网站上将展现您财付通登记的收货地址	需要申请

图 4-30 附录其他获取用户资料的 API[13]

## 4.4 小结

本章主要介绍了项目开发过程中的一些重要技术点和重要功能实现的过程，采用流程图和代码展现的方式进行了详细的阐述，对上一章的基本需求使用 JAVA, HTML, JavaScript 进行了业务的实现。

## 第四章 遇到问题和解决方案

### 5.1 防止用户重复提交论文

首先在提交按钮上我做了限制，即当用户登录到系统中来的时候，会先对用户的一个提交次数进行查询，通过一条简单的 SQL 查询语句:select count(\*) from upload where username=?and nodelete=0 即可实现 nodelete=0 表示没有删除,如果记录被管理员删除，那么肯定是允许用户是可以再次提交的，这种情况一般发生在自己论文信息上传有误，联系管理员进行删除操作，但是删除操作并不是物理删除，只是做了一个标识，这样做的好处就是便于统计论文出错率。

如果查询出来的数量=1 那么证明该学生已经提交过论文了，所以系统会对提交论文的按钮进行一个隐藏。

其次，还存在用户提交了论文，但是接着又刷新页面，这样表单数据又会提交到服务器，进行第二次上传操作，虽然浏览器在此时会提示用户可能会重复提交表单数据，是否继续？但是为了保险起见，我在服务端还是做了重复检验的操作，同样也是执行上面的 SQL 语句，如果值大于 1 就不进行数据插入的操作，直接返回给页面一个提示信息：请勿重复提交。

### 5.2 用户回话过期处理

当用户长时间没有操作页面时，页面的回话会过期，此时为了系统的安全性，应该在用户回话结束时禁止用户对系统的操作，通过网上查看各种资料 and 问题的解决方案，我采用了拦截器的方式，对每一次用户的请求进行了拦截，并检查用户当前回话的状态，一旦发现用户会话不存在了就让用户访问的页面重定向到系统登录页面，并提示用户回话结束。

### 5.3 浏览器兼容性处理

本次研究中遇到的兼容性问题就是在进行 PDF 预览时，之前使用的插件要求浏览器 Flash 的支持，所以在火狐这种不支持 Flash 的浏览器中无法正常预览 PDF，后来通过网上查询资料，我找到了一些兼容性更好的 PDF 预览插件，摆脱了对浏览器 Flash 的要求，并成功在页面加载。

## 5.4 防止 SQL 注入攻击的处理方案

我们常说的 SQL 注入攻击，就是有些用户把 SQL 命令插入到 WEB 表单提交或输入域名或页面请求的查询字符串发送到服务器，达到欺骗服务器的目的，最终使服务器执行恶意 SQL 命令的过程[14]。

在本项目中我主要是用了两种方式防止 SQL 的注入，首先我对用户表的密码字段进行了 MD5 加密，即使用户信息泄露也保证了用户密码的安全，当然这是最坏的打算。

然后我对 SQL 查询中的参数进行了字符串过滤的处理，对于那些恶意拼凑查询字符串的参数，会直接被替换。

## 第五章 总结

虽然我是物理系的一名学生，做出的课题确实计算机学院学生应该研究的课题，心里还是有点愧对我所学的专业。但是，我也不会放弃对物理领域研究的追求，将自己所学的计算机知识运用到物理上，说不定还能产生意想不到的结果呢。

再来说说我的项目，项目总算是告一段落，这两个月的坚持努力还是没有白费，先不说自己的项目做得如何，至少让人看到了自己的研究成果，做出了点实实在在的东西，还是很令人欣慰的。期间也遇到了不少问题，借助互联网的帮助，我努力寻求问题的解决方案，最终还是解决了不少问题，我不禁感叹真是一张大网啊，我已沉浸在解决各种问题的快乐中，互联网的作用确实不容小视。但是有时候自己独立思考解决问题的也很重要，太过依赖也会让自己的思维产生惰性，凡是遇到问题都不愿意思考，直接上网查。有时候一个问题想不通彻夜难眠，一次偶然的机会，我受到生活中一点小事的启发，突然茅塞顿开，问题就迎刃而解了，都要高兴得几晚睡不着觉呢。

还要感谢我的论文老师，柯老师，当时给了我这个能激发我学习动力的具有挑战性的课题，虽然是按照任务书要求如期完成了，但是离他所说的“能真正实用”还是有点差距，如果以后有要求，我依然会继续完善此系统，到达他所说的终极目的。

## 参考文献

- [1] 张峰/李慧丽 《Java WEB 2.0 架构开发与项目实战》清华大学出版社，2009
- [2] 科夫勒 《MYSQL 5 权威指南 第3版》人民邮电出版社，2006-12
- [3] Bruce Eckel 《Thinking in Java》机械工业出版社，2006-2-20
- [4] 巴沙姆等著 《深入浅出 Servlet & JSP》南京东南大学出版社，2006-9
- [5] 詹波 《JFinal 开发手册》，2012
- [6] IBM developerWorks
- [7] 于守良 《关于 Java 面向对象程序设计课程的思考》，2013(15):117-118
- [8] 朱岩 《可视化数据挖掘技术的设计与应用》，2013(A33):254-254
- [9] 王进 《B/S 模式下的三层架构》，2011
- [10] 李云云 《康向荣浅析 B/S 和 C/S 体系结构》。2011
- [11] 张淼波 《基于网络的软件体系结构应用浅析》，2014(5):28-28
- [12] 孟勋 《MVC 与三层结构技术的应用研究》，2013(9):23-25
- [13] QQ 互联文档资料 <http://wiki.connect.qq.com>
- [14] 张卓 《SQL 注入攻击技术及防范措施研究》，2007
- [15] 张裔智;赵毅;汤小斌 《MD5 算法研究》，2008
- [16] 张浩华 《基于 JAVA 技术的 MD5 加密算法的设计与实现》，2009, 27(1):75-77
- [17] 柳丽娜 《浅谈 Session 机制与 Cookie 机制》，2008(15):28-29
- [18] 林天华 《基于 WEB 的大文件上传技术研究》，2007, 25(16):30-33
- [19] 时子庆;刘金兰 《基于 OAuth2.0 的认证授权技术》，2012, 21(3):260-264
- [20] 屈展;李婵;《JSON 在 Ajax 数据交换中的应用研究》，2011, 26(1):95-98

## 致谢

我在写作过程中得到了很多人的帮助，谢谢支持我的同学和老师，在我想要放弃的时候，他们给我鼓励，让我继续完成了剩余的任务。

然后我要衷心的感谢的论文指导老师柯旋教授。

他在论文选题和任务书上给与了巨大的帮助，我能选择这个论题进行研究，他的努力有很大的关系。

感谢江汉大学四年来对我的辛苦培育，让我在大学四年没有荒废，学到不少东西，对人生的规划有了新的见解。感谢物理与信息工程学院为我提供了良好的学习环境和氛围，感谢学院领导和老师们四年来对我无微不至的关怀和指导，让我在四年里不断成长和进步。在此，我还要感谢四年的同窗舍友，给了我莫大的支持与鼓励，感谢你们。

# 承 诺 书

本人的毕业论文（设计）无抄袭、剽窃现象。本人熟知学校对毕业论文（设计）抄袭、剽窃现象按作弊处理，对已毕业的学生，学校将追回毕业证和学位证书。如本人毕业论文（设计）有以上违纪现象，所造成的知识产权等纠纷，一切后果由本人承担。

承诺人： 刘松林

2016 年 5 月 20 日