

**本 科 毕 业 设 计（论文）**



**题目: 学生作业批改系统**

**姓 名 赵瑞**

**学 院 计算机学院**

**专 业 网络工程**

**班 级 2016211311**

**学 号 2016211425**

**班内序号 16**

**指导教师 吴起凡**

**2020 年 5 月**

**北 京 邮 电 大 学**

**本科毕业设计（论文）任务书**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 计算机学院 | | 专业 | 网络工程 | | 班级 | 11班 |
| 学生姓名 | 赵瑞 | | 学号 | 2016211425 | | 班内序号 | 16 |
| 指导教师姓名 | 吴起凡 | | 所在单位 | 计算机学院 | | 职称 | 高工 |
| 设计(论文)题目 | （中文）学生作业批改系统 | | | | | | |
| （英文）Student Homework Correcting System | | | | | | |
| 题目分类 | 工程实践类☑ 研究设计类□ 理论分析类□ | | | | | | |
| 题目来源 | 题目是否来源于科研项目 是□ 否☑ | | | | | | |
| 科研项目名称： | | | | | | |
| 科研项目负责人： | | | | | | |
| 主要任务及目标：  在B/S架构下采用JavaWeb技术构建“学生作业批改系统”，为在校师生提供一个功能强大且便利简洁的平台，项目开发基于软件工程流程。目标：实现一系列的基础功能，包括管理员对成员、课程及年级班级的管理，教师布置、下载批改作业及打分统计，学生下载提交作业和分数查看，以及通用的个人资料、修改密码和退出登录。并在此基础上进行功能拓展和性能优化。 | | | | | | | |
| 主要内容：  主要采用JavaWeb技术，配套Tomacat、MySQL数据库和Navicat数据库管理工具等，于Eclipse上进行开发一个“学生作业批改系统”。需要深刻学习理解Java、jsp、servlet、MySQL、前端框架优化等相关知识。基于软件工程八大步骤（问题定义、可行性研究、需求分析、总体设计、详细设计、编码及单元测试、综合测试、软件维护）逐步构建，并做好相应记录。  主要功能思考：1.确保管理员的操作权限完整，不同操作间具有良好的逻辑关系 2.针对学生作业提交的不同格式进行相应处理 3.解决文件同时上传下载冲突及数据库等并发控制 4.建立目录结构为学生教师分角色存放每次作业，方便学生找回复习和教师期末统计 5.实现教师批量下载作业、作业批量上传及自动分类存放 6.个人资料界面绑定个人邮箱，若忘记密码可通过邮箱重置修改密码  先分单元测试再总体测试，详尽记录，项目完善完成后整理论文并准备答辩。 | | | | | | | |
| 主要参考文献：  [1]张海藩、牟永敏.软件工程导论.清华大学出版社.2013，（6）  [2] David Flanagan，美. JavaScript权威指南. 机械工业出版社2011,(6)  [3]郭欣.构建高性能Web站点. 电子工业出版社.2009  [4] Brian Goetz等.Java并发编程实战. 机械工业出版社.2012  [5]李兴华、王月清.Java Web开发实战经典. 清华大学出版社.2010  [6]JonathanChaffer、KarlSwedberg. jQuery基础教程.人民邮电出版社.2008  [7] Trevor Burnham. JavaScript异步编程. 人民邮电出版社.2013  [8] Craig Walls. Spring实战. Manning.2011 | | | | | | | |
| 进度安排：  第1~3周：需求分析，明确项目重点难点，完成《毕业设计任务书》及《开题报告》，搭建项目环境  第4~5周：完成网页基础模块构建，连接数据库，优先实现管理员登录  第5~8周：逐步实现各种基础功能，确保模块间连接稳定，记录相关问题  第9~11周：改进完善基础功能，对所有功能进行技术测试并记录  第12周~：完成本科毕业论文，准备答辩 | | | | | | | |
| 指导教师签字 | |  | | 日期 | 年 月 日 | | |

**北京邮电大学本科生毕业设计（论文）成绩评定表**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学生  姓名 | 赵瑞 | | | | | 所在学院 | 计算机学院 | | | | | | | | | | | | | | |
| 学号 | 2016211425 | | 专业 | | 网络工程 | | | 班级 | | | | 2016211311 | | | | | | | | | |
| 论文  题目 | （中文）学生作业批改系统 | | | | | | | | | | | | | | | |  | | | | |
| （英文）Student Homework Correcting System | | | | | | | | | | | | | | | |  | | | | |
| 指导教师 姓 名 | 吴起凡 | | 指导教师职称 | 高工 | | | | 指导教师单位 | | | | 计算机学院 | | | | | | | | | |
| 中期检查小组评分 | （满分10分）：宋体，五号 中期检查小组组长签字：宋体，五号 检查日期： | | | | | | | | | | | | | | | | | | | | |
| 指 导 教 师 评 分 | 评价内容 | 具体要求 | | | | | | 分值 | | | | | | | | | | 评分 | | | |
| 指导教师 | | | 复议 |
| 调研论证 | 能独立查阅文献和从事相关调研；能正确翻译外文资料；有收集、加工各种信息及获取新知识的能力和自学能力。 | | | | | | 5 | 4 | | 3.5 | | | 3 | 2 | | |  | | |  |
| 方案设计 | 能独立提出符合选题的可行性研究方案、实验方案、设计方案，独立进行实验（如安装、调试、操作）和研究方案论证。能够在设计环节中，考虑社会、健康、安全、法律、文化以及环境等因素；理解和评价工程实践对环境、社会可持续发展的影响。 | | | | | | 5 | 4 | | 3.5 | | | 3 | 2 | | |  | | |  |
| 能力水平 | 能综合运用所学知识和技能去分析与解决毕业设计（论文）过程中遇到的实际问题；能正确处理实验数据；能对课题进行理论分析，得出有价值的结论。 | | | | | | 5 | 4 | | 3.5 | | | 3 | 2 | | |  | | |  |
| 学习态度 | 认真、勤奋、努力、诚实、严格遵守纪律，按期饱满完成规定的任务。 | | | | | | 5 | 4 | | 3.5 | | | 3 | 2 | | |  | | |  |
| 设计（论文）水平 | 文题相符、综述简练完整，有见解；立论正确，论述充分，结论严谨合理；实验正确，分析处理科学；文字通顺，技术用语准确，设计（论文）有理论价值和应用价值。 | | | | | | 5 | 4 | | 3.5 | | | 3 | 2 | | |  | | |  |
| 文本规范 | 装订顺序正确，字体字号等与基本规范相符，符号统一，编号齐全，图表完备、整洁、正确。 | | | | | | 5 | 4 | | 3.5 | | | 3 | 2 | | |  | | |  |
| 指导教师评分合计（满分30分）：  评语：  宋体，五号，1.2倍行距，两端对其，首行缩进2字符  指导教师签字： 日期： 年 月 日 | | | | | | | | | | | | | | | | | | | | |
| 复议 | □是 □否 复议评分合计： 复议人签字： 复议日期：  复议有权限修改指导教师评分，选择复议后指导教师评分将由复议评分替换 | | | | | | | | | | | | | | | | | | | | |
| 本科生毕业设计（论文）答辩成绩评定标准 | | | | | | | | | | | | | | | | | | | | | |
| 答  辩  小  组  成  绩  评  定 | **评价内容** | **具体要求** | | | | | | **分值** | | | | | | | | | | | | **评分** | |
| 选题 | 符合专业培养目标，符合社会实际、结合工程实际，难易适度，体现新颖性、综合性。 | | | | | | 5 | | 4 | | | 3.5 | | | 3 | | | 2 |  | |
| 设计（论文）质量水平 | 全面完成任务书中规定的各项要求，文题相符，工作量饱满，写作规范，达到综合训练的要求，有理论成果和应用价值。能够在设计环节中，考虑社会、健康、安全、法律、文化以及环境等因素；理解和评价工程实践对环境、社会可持续发展的影响。 | | | | | | 20 | | 16 | | | 14 | | | 12 | | | 8 |  | |
| 答辩准备 | 准备充分；有简洁、清晰、美观的演示文稿；准时到场。 | | | | | | 5 | | 4 | | | 3.5 | | | 3 | | | 2 |  | |
| 内容陈述 | 语言表达简洁、流利、清楚、准确，思路清晰，重点突出，逻辑性强，概念清楚，论点正确；实验方法科学，分析归纳合理；结论严谨；表现出对毕业设计（论文）内容掌握透彻。 | | | | | | 20 | | 18 | | | 14 | | | 12 | | | 8 |  | |
| 回答问题 | 回答问题准确、有深度、有理论根据、基本概念清晰。 | | | | | | 10 | | 8 | | | 7 | | | 6 | | | 4 |  | |
| 答辩小组评分合计（满分60分） | | | | | | |  | | | | | | | | | | | | | |
| 意见：  宋体，五号，1.2倍行距，两端对其，首行缩进2字符  答辩小组组长签字： 年 月 日 | | | | | | | | | | | | | | | | | | | | |
| 答辩小组成员：宋体，五号 | | | | | | | | | | | | | | | | | | | | |
| 学 院 意 见 | 最终成绩：百分制 ； 五分制  院长签章： 学院盖章： 年 月 日 | | | | | | | | | | | | | | | | | | | | |
| 备注 |  | | | | | | | | | | | | | | | | | | | | |

注：1．毕业设计（论文）成绩由中期检查评分（满分10分）、指导教师评分/复议评分（满分30分）和答辩小组评分（满分60分）相加，得出百分制成绩，再按100-90分为“优”、89-80分为“良”、79-70分为“中”、69-60分为“及格”、60分以下为“不及格”的标准折合成五级分制成绩；

2． 此表原件一式三份，一份存入学生档案，一份装订到毕业论文中，一份交教务处存入档案馆。

**北 京 邮 电 大 学**

**本科毕业设计（论文）诚信声明**

本人声明所呈交的毕业设计（论文），题目《学生作业批改系统》是本人在指导教师的指导下，独立进行研究工作所取得的成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名： 日期：

**学生作业批改系统**

**摘 要**

该选题基于传统作业提交速度慢、效率低等弊端，旨在结合网络的高效和便利，为在校师生提供一个在线作业批改平台。以往的作业批改依赖老旧的流程：教师课堂布置作业-学生记录作业-学生按时完成-课间课代表收集-教师讲解批改-课代表分发-学生纠错保存-教师期末统计平时分，纵观这一流程，给人第一印象便是繁琐拖沓，很多过程都可能发生时间的浪费和作业的遗漏。于是该选题应运而生，于学校具有十分现实的意义，能够大幅度提高作业批改的效率，并在繁忙的期末提供便利。

本文主要基于完成该系统程序所做的记录和总结，包括课题相关内容、采用技术介绍、系统需求分析、系统总体设计、主要功能模块详细设计、功能测试改进和总结展望。对程序的过程实现和功能分析等具有完备的分析，是对整个项目从立项到完成的流程记录，其中内容可以根据需要在文中自行查看了解。

该项目采用Javaweb、Mysql、bootstrap、jquery、Ajax等开发技术，流程参照软件工程，于Eclipse上实现的基于B/S架构的网页项目。主要对“学生作业批改”这一核心内容进行模拟和功能实现，包括角色的功能权限、作业上传下载、数据库构建、布局UI等核心模块的尝试实现和测试修改。整体目标是：对系统功能结合实际分析讨论，在完成该系统的基础核心功能的前提下，尽可能做到使用便利、界面美观、高拓展性等额外考虑，超脱个人项目，向商业软件靠齐。

**关键词** 作业批改 Javaweb B/S架构 软件工程

**Student Homework Correcting System**

**ABSTRACT**

This topic is based on the disadvantages of slow speed and low efficiency of traditional homework submission. It aims to provide an online homework Correcting platform for teachers and students in school with the high efficiency and convenience of the network. In the past, homework grading relies on the old process: Teacher's classroom assignment, student's record of homework, student's timely completion, collection of representatives of class, teacher's explanation and correction, class representative's distribution, student's error correction and saving, and teacher's final statistics of normal score. Looking at this process, the first impression is that it is tedious and procrastinating, and many processes may lead to waste of time and omission of homework. Therefore, the topic came into being, which has a very practical significance in the school, can greatly improve the efficiency of homework correction, and provide convenience at the end of the busy period.

This paper is mainly based on the record and summary of the completion of the system program, including the relevant content of the subject, the introduction of technology, the analysis of system requirements, the overall design of the system, the detailed design of the main functional modules, the improvement of the function test and the summary of the outlook. It has a complete analysis on the process realization and function analysis of the program. It is a process record of the whole project from project initiation to completion. The content can be viewed and understood in the paper according to the needs.

The project uses Java Web, mysql, bootstrap, jQuery, Ajax and other development technologies, and processes reference software engineering to implement a web page project based on B / S architecture on eclipse. It mainly simulates and realizes the core content of "student's homework correction", including the function permission of role, upload and download of homework, database construction, layout UI and other core modules' attempt realization and test modification. The overall goal is to analyze and discuss the system functions in combination with the actual situation. On the premise of completing the basic core functions of the system, it is necessary to make use of the convenience, beautiful interface, high expansibility and other external considerations as much as possible, to get rid of personal projects and to rely on commercial software.

**KEY WORDS** job grading java Web b/ s architecture software engineering

**目 录**

[**第一章 引言 1**](#_Toc514855714)

[1.1 课题背景 1](#_Toc514855715)

[1.2 课题任务 1](#_Toc514855718)

[1.2.1 课题内容 1](#_Toc514855719)

[1.2.2 本人承担任务 1](#_Toc514855720)

[1.3 论文结构 2](#_Toc514855721)

[**第二章 相关技术介绍(可选) 3**](#_Toc514855722)

[2.1 Javaweb 3](#_Toc514855723)

[2.2 Tomcat 3](#_Toc514855725)

[2.3 MySQL 4](#_Toc514855726)

[2.4 eclipse 5](#_Toc514855727)

[2.5 本章小结 6](#_Toc514855728)

[**第三章 系统的需求分析 7**](#_Toc514855729)

[3.1系统功能需求分析 7](#_Toc514855732)

[3.1.1 管理员 8](#_Toc514855731)

[3.1.2 教师 10](#_Toc514855731)

[3.1.3 学生 10](#_Toc514855731)

[3.2 系统非功能需求分析 11](#_Toc514855733)

[3.3 本章小结 11](#_Toc514855735)

[**第四章 系统的总体设计 12**](#_Toc514855736)

[4.1 系统网络结构设计 12](#_Toc514855737)

[4.2 系统软件层次架构设计 12](#_Toc514855739)

[4.3 系统功能模块设计 14](#_Toc514855740)

[4.4 系统的数据库设计 15](#_Toc514855741)

[4.5 系统的界面设计 18](#_Toc514855742)

[4.6 本章小结 19](#_Toc514855743)

[**第五章 系统主要功能模块的详细设计与实现 21**](#_Toc514855744)

[5.1 管理模块的详细设计与实现 21](#_Toc514855745)

[5.2 作业模块的详细设计与实现 23](#_Toc514855747)

[5.3 通用模块的详细设计与实现 25](#_Toc514855748)

[5.4 额外模块的详细设计与实现 27](#_Toc514855748)

[5.5 本章小结 27](#_Toc514855750)

[**第6章 系统测试 28**](#_Toc514855751)

[6.1 系统测试方法介绍 28](#_Toc514855752)

[6.2 系统的功能测试 28](#_Toc514855753)

[6.3 系统的性能测试 30](#_Toc514855754)

[6.4 本章小结 31](#_Toc514855755)

[**第7章 结束语 33**](#_Toc514855756)

[7.1 论文工作总结 33](#_Toc514855757)

[7.2 问题和展望 33](#_Toc514855758)

[**参考文献**](#_Toc514855759)

[**致 谢**](#_Toc514855760)

[**附 录**](#_Toc514855761)

[**外 文 资 料**](#_Toc514855762)

[**外 文 译 文**](#_Toc514855763)

**开 题 报 告**

**中 期 检 查 表**

**教师指导毕业设计（论文）记录表**

# 引言

## 课题背景

时至今日，随着高校课业的多样化和计算机网络的高速发展，传统作业布置和纸质版提交愈发显得笨重繁琐，依靠课代表收集作业的模式不能确保每次作业收集的稳定性，大大降低了学生作业批改的效率，尤其是对于习惯了网络便利的当代大学生而言，日常生活中的作业保管和忘记作业时间等问题屡见不鲜，而网络的高效便利恰恰与之互补。

针对这种普遍存在的情况，构建一个“学生作业批改系统”线上平台，为学生提供便利的作业提交渠道和长久的作业保存，能够减轻学生不必要的过失（忘记提交时限、丢失作业等）；对于教师，能够省去传统通过线下布置作业并通过课代表收集统计的时间，在“学生作业批改系统”中实现一键批量下载和批量上传也能够大大节约教师时间。科技的发展应该更好的服务人们的生活工作，该系统应用于学校能更好地统计学生的平时成绩，在期末这种工作繁多的时候能够提供很大帮助，该系统的开发具有深远的现实意义，可谓百益而无一害。

## 1.2 课题任务

### 1.2.1 课题内容

采用Java Web技术，配套相应工具实现一个应用于学校的“学生作业批改系统”，为在校师生提供一个功能强大且便利简洁的平台，项目开发基于软件工程流程。基本内容包含管理员、教师和学生三个角色，并分模块实现各角色对应基础功能：包括管理员对成员、课程、年级及班级的管理，教师布置作业、下载批改作业及打分统计，学生下载提交作业和分数查看，以及通用的个人资料、修改密码和退出登录，并在此基础上进行功能拓展和性能优化。设计功能时需要充分考虑实际使用情况，结合现实问题全面分析，构建数据流图，力求通过分析得到各模块的细致划分；勇于在项目实现中尝试新技术，充分发挥创造力和想象力，做出让人眼前一亮的设计；具体实现中应当对比不同设计和不同代码实现的差别优劣，选取更加稳定高效的方案实现。

### 1.2.2 本人承担任务

作为该选题的实施者，我有义务对自己所选题目负责，要以神圣的态度对待本次毕设，它不仅仅是我毕业前的最后一次考核，也是对于自己一次试炼，一场既关乎技术也关乎为人行事的考验，所以本人应在完成系统基础功能的前提下，确保程序的合理性和完整性，优化各模块的逻辑功能，记录好程序开发流程，整理保存相关文件，汇总并精简的完成论文撰写工作，做到对程序的详尽分析测试，并备好相关资料以备答辩之需。

## 1.3 论文结构

由于该选题是基于工科实践类的程序设计，故论文中有较多程序设计相关知识，而这些知识是体系完备并且应用规范的。所以在论文中我会尽可能以自己的理解和语言，来分析解释程序中技术的应用和功能模块的实现，我会截取部分重要代码记录分析重要功能，部分代码放至附录以作参考。论文分为论文规范和程序设计两部分，整体结构如下：封面→任务书→北京邮电大学本科生毕业设计(论文)成绩评定表→诚信说明→中文摘要（含关键词）→外文摘要（含关键词）→目录→正文→参考文献→致谢→附录→外文资料→外文翻译→开题报告→中期检查表→教师指导毕业设计（论文）记录表-（上述为论文装订顺序，以下为论文内容顺序，且全部包含于论文正文部分中）-引言（课题背景及任务）-相关技术介绍-系统需求分析-系统总体设计-系统主要功能模块的详细设计与实现-系统测试-结束语（终结和展望）。

# 第二章 相关技术介绍(可选)

## 2.1 Javaweb

Javaweb指的是一系列服务于java网页开发的技术和工具，从属于JavaEE领域，即Java Enterprise Edition。初级可以用JSP（Java Server Pages）+servlet+Javabean来开发，大型网站一般使用框架和更为复杂的容器。

WEB应用程序指供浏览器访问的程序，通常也简称为web应用。例如有a.html 、b.html…..多个web资源，这多个web资源用于对外提供服务，此时应把这多个web资源放在一个目录中，以组成一个web应用（或web应用程序）  
　　一个web应用由多个静态web资源和动态web资源组成，如:html、css、js文件，Jsp文件、java程序、支持jar包、配置文件等等。  
　　Web应用开发好后，若想供外界访问，需要把web应用所在目录交给web服务器管理，这个过程称之为虚似目录的映射。

在静态WEB程序中，客户端使用WEB浏览器（IE、chrome等）经过网络(Network)连接到服务器上，使用HTTP协议发起一个请求（Request），告诉服务器我现在需要得到哪个页面，所有的请求交给WEB服务器，之后WEB服务器根据用户的需要，从文件系统（存放了所有静态页面的磁盘）取出内容。之后通过WEB服务器返回给客户端，客户端接收到内容之后经过浏览器渲染解析，得到显示的效果。Web可以分为静态和动态。

静态WEB中存在以下几个缺点：

　　1、Web页面中的内容无法动态更新，所有的用户每时每刻看见的内容和最终效果都是一样的。

　　为了可以让静态的WEB的显示更加好看，可以加入了JavaScript以完成一些页面上的显示特效，但是这些特效都是在客户端上借助于浏览器展现给用户的，所以在服务器上本身并没有任何的变化。

　　实现静态WEB客户端动态效果的手段：

  JavaScript

  VBScript

　　在实际的开发中JavaScript使用得最多。

　　2、静态WEB无法连接数据库，无法实现和用户的交互。

　　使用数据库保存数据是现在大多数系统的选择，因为数据库中可以方便地管理数据，增删改查操作可以使用标准的SQL语句完成。

所以当今主流都是动态web，即具有交互性的web页面，既能够便利的实现交互，也能支持更多的技术。

## 2.2 Tomcat

Tomcat是当前流行的Web轻量级应用服务器，可以用来开发Javaweb项目的容器，且具有处理HTML页面的功能，能够运行JSP页面和Servlet，其结构如图2-1所示，

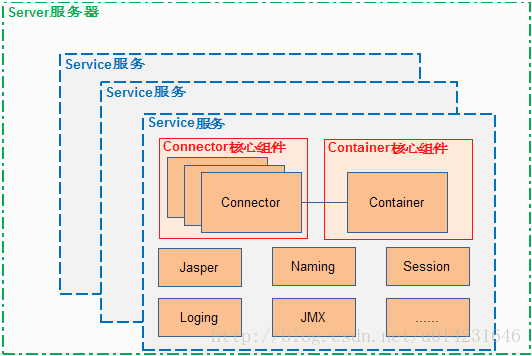


图2-1 Tomcat 结构

Tomcat主要功能便是实现网络交互传递功能，其中Server是一个顶级组件，代表一个Tomcat实例。Connector主要用来处理客户端传来的请求，每个Connector与一个TCP端口绑定，且Service中默认包含两个Connector。Container是容器的父接口，由四个自容器组件构成，分别是Engine、Host、Context、Wrapper。Engine 容器基础，它只定义了一些基本的关联关系。Host 是 Engine 的子容器，一个Host在 Engine 中代表一个虚拟主机，它的子容器通常是 Context，它除了关联子容器外，还有就是保存一个主机应该有的信息。Context代表Servlet的Context，它具备了Servlet运行的基本环境，Context最重要的功能就是管理它里面的Servlet实例Servlet实例在Context 中是以Wrapper出现的。Wrapper代表一个Servlet，它负责管理一个 Servlet，包括的 Servlet 的装载、初始化、执行以及资源回收。Wrapper 是最底层的容器，它没有子容器了，所以调用它的 addChild 将会报错。

## 2.3 MySQL

MySQL 是最流行的数据库之一，是一个免费开源的关系型数据库管理系统，但也不意味着该数据库是完全免费的。MySQL 由瑞典 MySQL AB 公司开发，目前属于 Oracle 公司。MySQL 适合中小型软件，饱受个人用户以及中小企业青睐。

MySQL 数据库管理系统具有很多的优势，下面总结了其中几种。

1）MySQL 是开放源代码的数据库

MySQL 是开放源代码的数据库，任何人都可以获取该数据库的源代码。这就使得任何人都可以修正 MySQL 的缺陷，并且任何人都能以任何目的来使用该数据库。MySQL 是一款可以自由使用的数据库。

2）MySQL 的跨平台性

MySQL 不仅可以在 Windows 系列的操作系统上运行，还可以在 UNIX、Linux 和 Mac OS 等操作系统上运行。因为很多网站都选择 UNIX、Linux 作为网站的服务器，所以 MySQL 的跨平台性保证了其在 Web 应用方面的优势。虽然微软公司的 SQL Server 数据库是一款很优秀的商业数据库，但是其只能在 Windows 系列的操作系统上运行。因此，MySQL 数据库的跨平台性是一个很大的优势。

3）价格优势

MySQL 数据库是一个自由软件，任何人都可以从 MySQL 的官方网站上下载该软件，这些社区版本的 MySQL 都是免费试用的，即使是需要付费的附加功能，其价格也是很便宜的。相对于 Oracle、DB2 和 SQL Server 这些价格昂贵的商业软件，MySQL 具有绝对的价格优势。

4）功能强大且使用方便

MySQL 是一个真正的多用户、 多线程 SQL 数据库服务器。它能够快速、有效和安全的处理大量的数据。相对于 Oracle 等数据库来说，MySQL 的使用是非常简单的。MySQL 主要目标是快速、健壮和易用。  
  
 MySQL 与常用的主流数据库 Oracle、SQL Server 相比，主要特点就是免费，并且在任何平台上都能使用，占用的空间相对较小。但是，MySQL 也有一些不足，比如对于大型项目来说，MySQL 的容量和安全性就略逊于 Oracle 数据库。

## 2.4 eclipse

Eclipse是一个基于Java的、开放源码的、可扩展的应用开发平台，它为编程人员提供了一流的Java集成开发环境（Integrated Development Environment，IDE）。在Eclipse的官方网站中提供了一个Java EE版的Eclipse IDE。应用Eclipse IDE for Java EE，既可以创建Java项目，也可以创建动态Web项目。

在Eclipse工作台的上方提供了菜单栏，该菜单栏包含了实现Eclipse各项功能的命令，并且与编辑器相关，即菜单栏中的菜单项与当前编辑器内打开的文件是关联的。

其良好的拓展性和强大的兼容性是其流行重要原因，它本身自带一个软件市场，许多需要额外工具的开发都可以得到满足，就连Github库它也能直接分享到网页指定url中去，其功能十分强大且上限极高。

Eclipse SDK（软件开发者包）是Eclipse Platform、JDT和PDE所生产的组件合并，它们可以一次下载。这些部分在一起提供了一个具有丰富特性的开发环境，允许开发者有效地建造可以无缝集成到Eclipse Platform中的工具。Eclipse SDK由Eclipse项目生产的工具和来自其它开放源代码的[第三方软件](https://baike.baidu.com/item/%E7%AC%AC%E4%B8%89%E6%96%B9%E8%BD%AF%E4%BB%B6)组合而成。Eclipse项目生产的软件以 GPL发布，第三方组件有各自自身的许可协议。

## 

## 2.5 本章小结

本章主要简述了部分该项目采用的技术和工具，是该程序开发的必备前提。对技术和工具的了解能够很大程度上帮助我们进行合理的程序开发。其实很多该程序使用的其他技术都没有在本章提及，比如jQuery和ajax，bootstrap框架，很多资源包拓展包等等。不是因为他们不重要，而是因为一章的内容有限，过多的内容会导致章节的主题得不到体现，并且会显得冗余，因为一项技术的介绍，并不单单是它的原理用途，更要加上自己的使用心得和自我理解，但这样还是会导致篇幅过多显得冗杂，于是我只能选取其中最基础而不是最重要的技术以作分析。在后续的需求分析和功能具体实现中，我有信心将所有技术体现在程序的设计中，所以此处没有放置过多的技术原理。

# 系统的需求分析

## 3.1 系统功能需求分析

功能需求分析是软件计划阶段的重要活动，是软件后续开发的基础保障，通过明确程序角色的目的，分析其实际效用，通过绘制一系列的uml图，按使用角色得出各模块功能，再对功能进行具体的需求分析。该阶段主要任务是通过分析系统的功能，指定需要“实现什么”，而不考虑“如何实现”。将功能分角色分析实现是一个好办法，能够优先从角色方面入手，解决一系列的问题，弊端是容易单独考虑某个职能而忽视模块之间的交互逻辑、数据库的约束等条件。不同角色也可能具有相同模块，如该作业批改系统中便是个人信息、退出登录、修改和找回密码等通用模块，其中也分为依附于UI界面的退出和有数据变动如修改信息这两类，前者可以在UI上单独解决，后者在不同角色中解决思路相近，但具体操作可能具有很大差异。通过绘制uml图初步分析需求，以后再根据进度修改对应部分。

整体功能需求见图3-1，角色间关系的类图如图3-2.

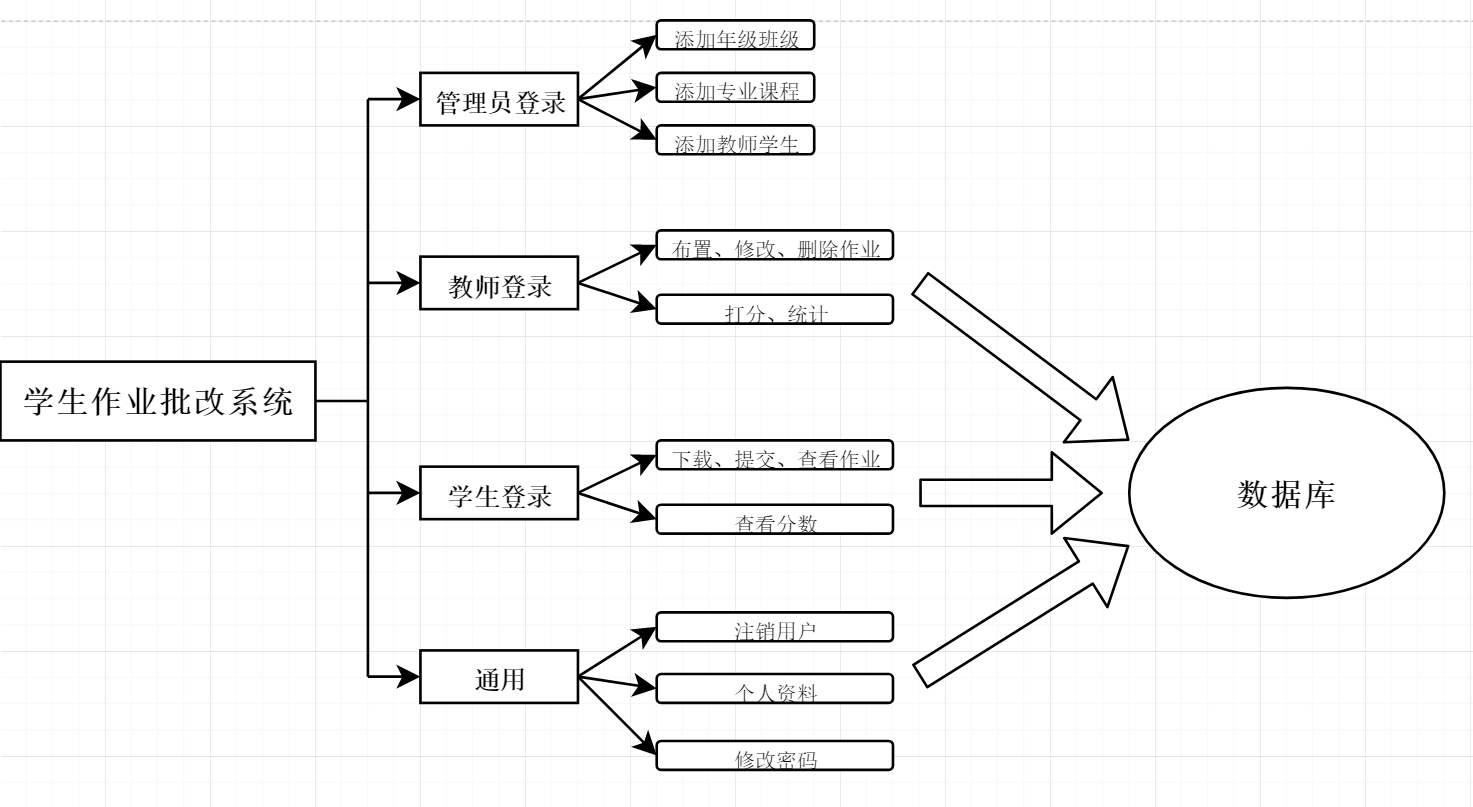


图3-1 整体需求分析

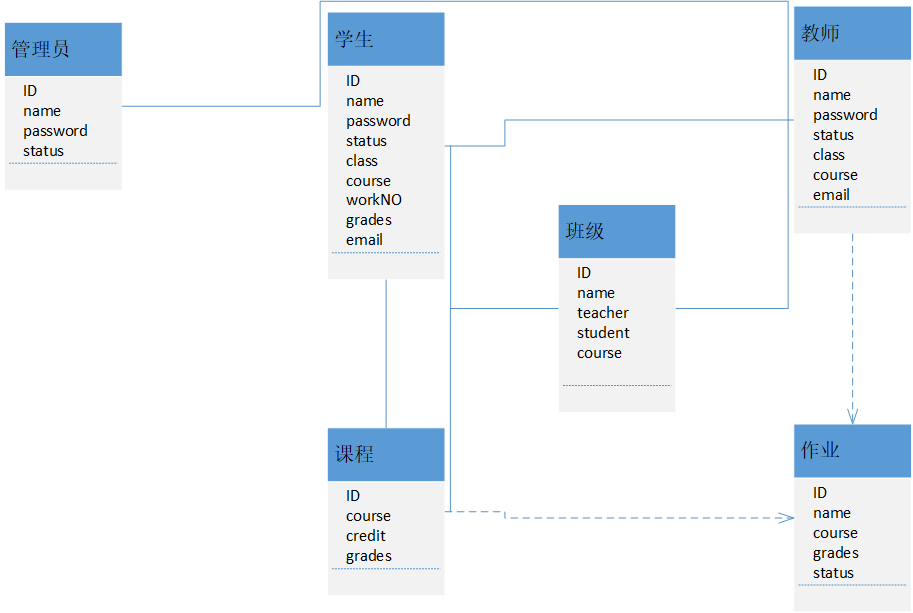


图3-2 类图

### 3.1.1 管理员

管理员角色是程序运行维护的首要操作对象，其职能应当包含对整个系统模块和其他角色信息的完全控制，其功能完全是整个程序运行的前提。管理员角色具体的功能包括：对用户（教师、学生）的管理（增删改查），对年级班级的管理（通过为学生添加学号班号实现）。结合实际考虑到学校不会有专门的人员进行管理，一般是选取某些特定教师作为管理员，所以管理员也要承担教师的作业功能，于是在设计上将拥有额外功能的教师设置为管理员，并且赋予管理员添加其他教师为管理员的权限。所以管理员也具有教师的职能：包括布置作业，下载批改，上传答案。以下为部分uml图。

管理员用例图如图3-3。用例图主要用由参与者和用例构成，参与者表示再系统交互中所扮演的角色，可以是成员，也可以是参与系统的元素变量；用例主要针对具体功能进行建模。

系统顺序图如图3-4。系统顺序图（SSD）主要应用于更为复杂的场景，包含模块间的数据交互和顺序流程。

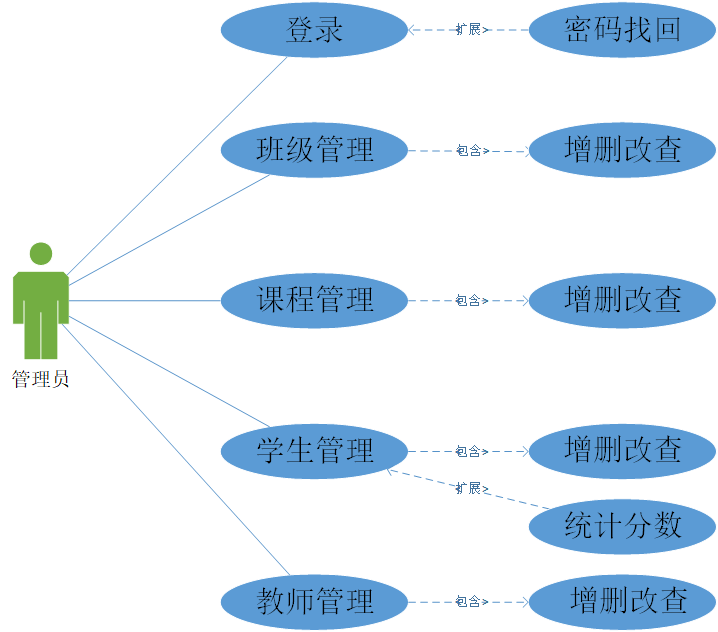


图3-3 管理员用例图



图3-4 管理员 SSD图

### 3.1.2 教师

教师的职能主要为布置作业、下载批改、查看编辑及打分统计。其中需要注意的是：一个教师可能执教多个班级，但是一般只会执教同一个年级，这一点在程序中实现的话需要考虑众多复杂情况，比如数据库中班级、教师、学生之间的约束，教师对同一科目不同班级的下载查看，也会影响学生作业的保存。虽然需求分析在前软件实现在后，但如果像这种可以预知的问题还是需要在需求分析中体现出思考。

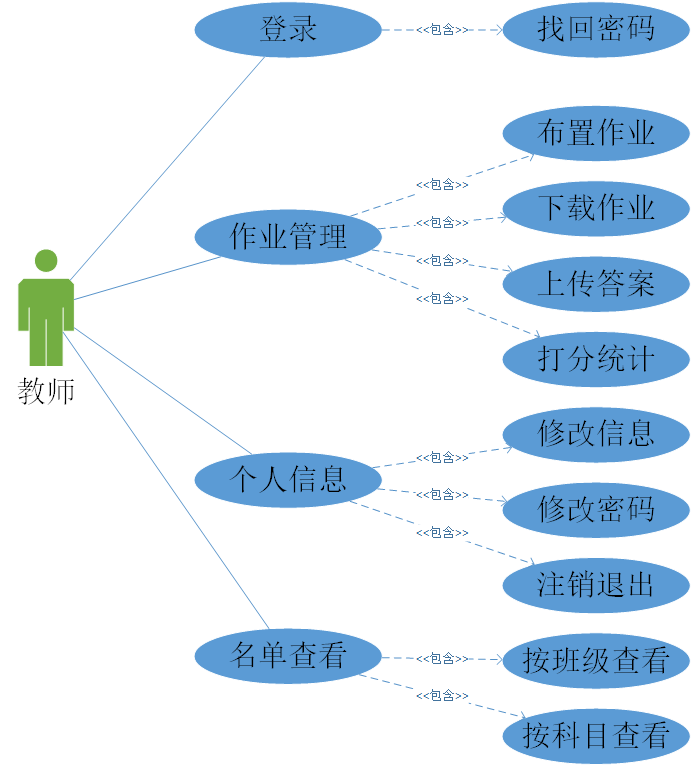
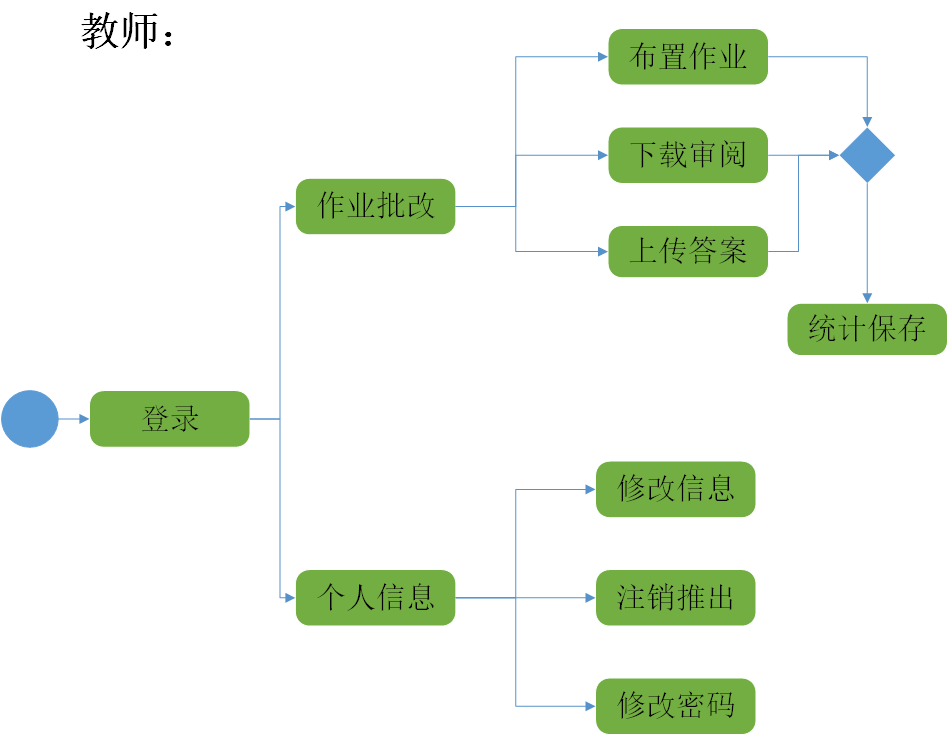


图3-5 教师活动图 图3-6 教师用例图

### 3.1.3 学生

学生的主要功能包含：下载、提交作业，查看分数、答案和自己保存的作业。

学生的功能大多依附于其他角色，在作业流程中属于被动的一方，所以在需求分析中更需着重于其所属模块和其他模块的交互接口，主从次序。

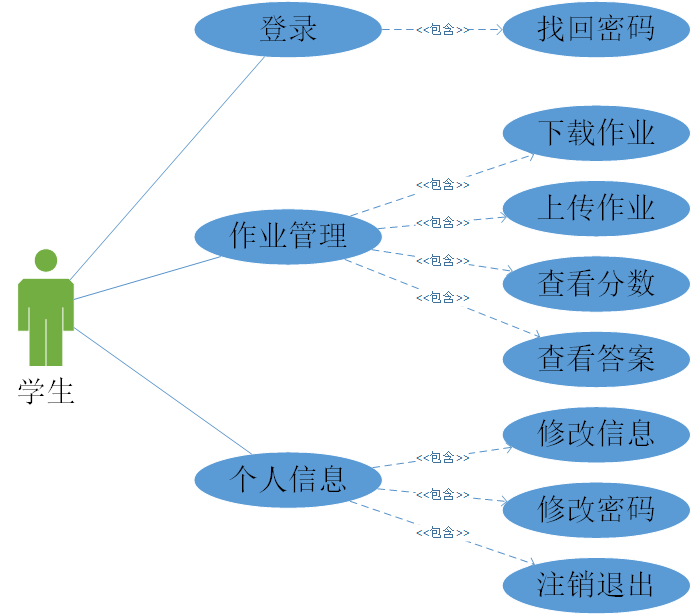
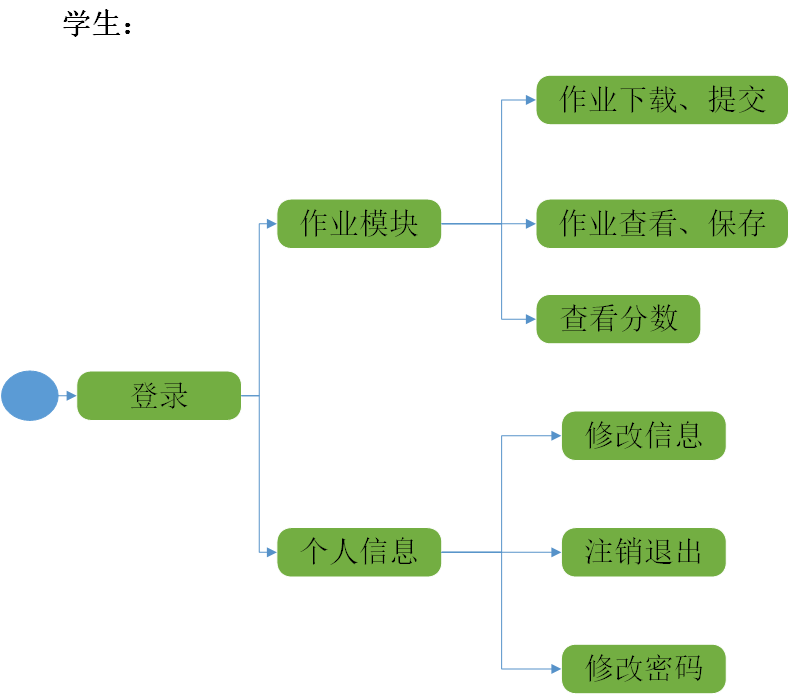


图3-7 学生活动图 图3-8 学生用例图

## 3.2 系统非功能需求分析

非功能需求更着重于整个系统的完整性、可靠性、拓展性等硬性要求。打个比方：一场体育比赛，其中运动员、裁判、比赛项目、比赛规则等类比于功能性需求，是构成这场比赛必不可少的条件；而场地、天气、观众、门票价格等类比于非功能需求，这些能够提高比赛的舒适度、观赏度，也就是软件的使用观感。非功能需求大体分为几个方面：性能、可靠性、可扩展性、互操作性、健壮性等，以下将具体针对该系统进行非功能需求分析。

观感需求：由于该系统属于经常使用的网页项目，所以其UI界面的构建需要满足观感良好、使用便利，比主要功能按钮应当显眼，各模块区分度要高，界面力求精简美观，由于是毕设项目，所以尽可能将所有功能体现在界面中，给人一种整体感。

性能：其中各模块主要涉及性能的有文件上传下载、文件存储和对数据库的操作。由于现阶段仍属于个人项目，角色占用的资源和网络比较少，对网络和服务器的负荷几乎可以不用考虑，所以当前性能主要体现在功能处理的逻辑、流畅分层次的代码结构和数据库构建的完整性上。

安全性：和上述类似， 现阶段对于安全性的考虑很难达到商业化软件对网络、硬件、更新和信息的安全考虑，更多侧重的还是角色用户的功能性完备，用户数据（个人性息和上传作业）的隔离良好，角色的权限逻辑无误等偏向功能性的考虑。

可维护性：可维护性强调一个模块的底层实现一定要留有余地，尽量简洁，为以后功能的拓展提供良好的底层条件。在具体实现中，如果该功能考虑到以后可能会有某方面的发展改进，最好预留出接口与数据空间，并记录备注，方面以后版本更新。对于一些容易发生报错的地方，要明确该处的实现方法与技术规格，也是为以后检修升级做好准备。

## 3.3 本章小结

本章的需求分析是程序开发的基础，好的需求分析能够减轻后续开发的许多负担，但是随着程序开发的进度，不同时期的要求或者实现的方法也不尽相同，这时需要及时调整，做到与进度匹配，因为程序的需求分析不管再全面再细致，终究是纸上谈兵，落到实处还是为了程序实现服务，所以二者相互制约相互改进，才是程序开发的合理之道。

# 第四章 系统的总体设计

## 4.1 系统整体架构设计

该系统是标准规格的网络项目，其网络部分的信息传输完全遵循HTTP协议，HTTP在服务器和客户端之间建立请求/响应模型实现交互和信息传输，简单快捷。该系统网络核心部分采用tomcat实现。

由于服务器架构因需求而异，所以以图4-1做参考模板。

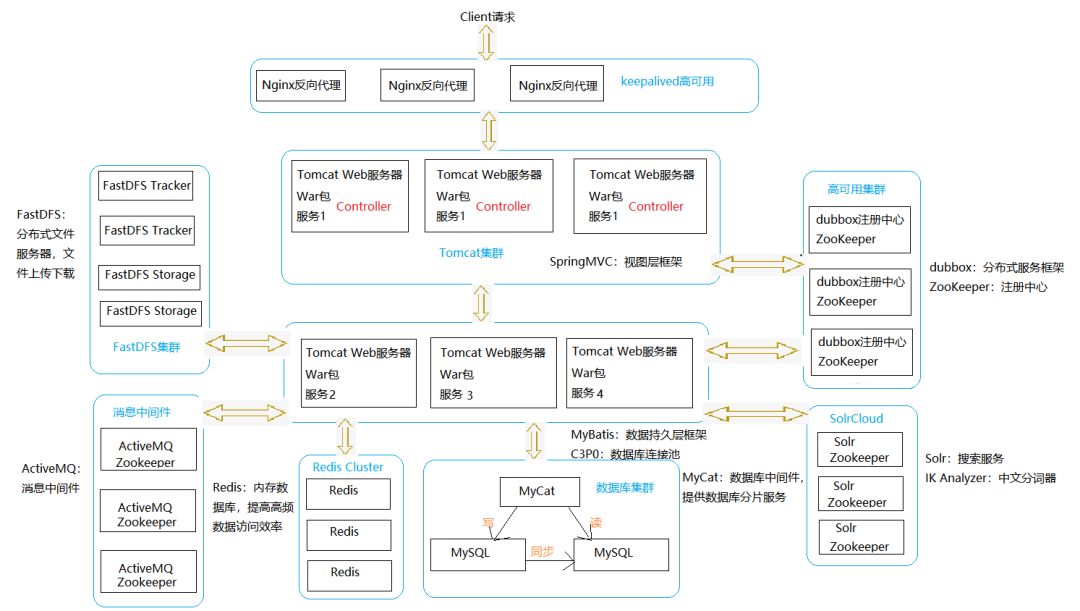


图4-1 整体架构参考

## 4.2 系统软件层次架构设计

该系统采用三层架构，其结构模式如图4-2。三层架构的三层也可以表示为界面层（User Interface layer）、业务逻辑层（Business Logic Layer）、数据访问层（Data access layer）。表示层顾名思义，主要是界面UI，用于用户交互操作。系统主要功能和业务逻辑都在业务逻辑层进行处理。数据访问层主要是针对数据库的操作，拥有数据处理能力。

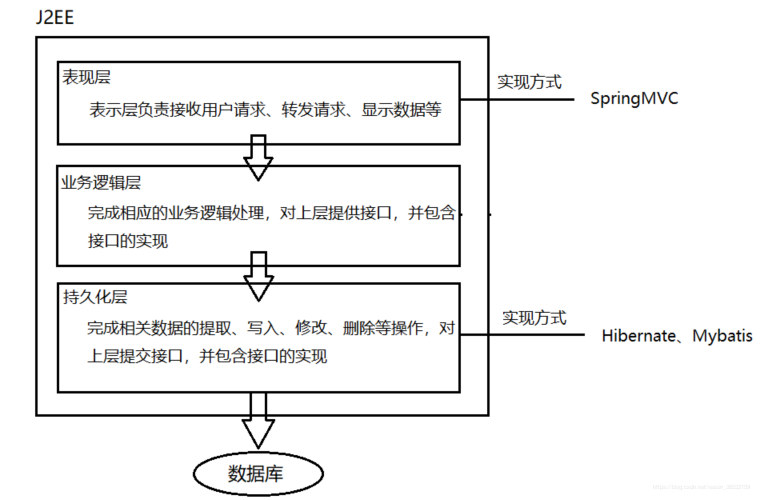


图4-2 三层架构

Spring MVC框架，全称Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写。其参考结构如图4-3.工作原理如图4-4

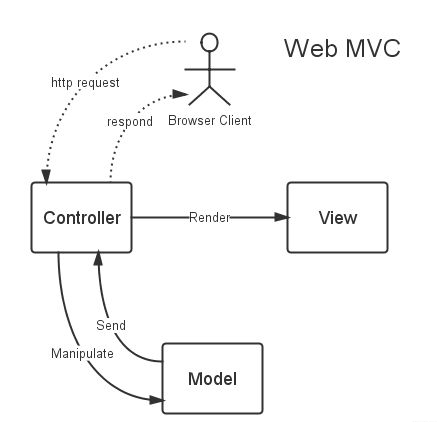


图4-3 MVC 结构

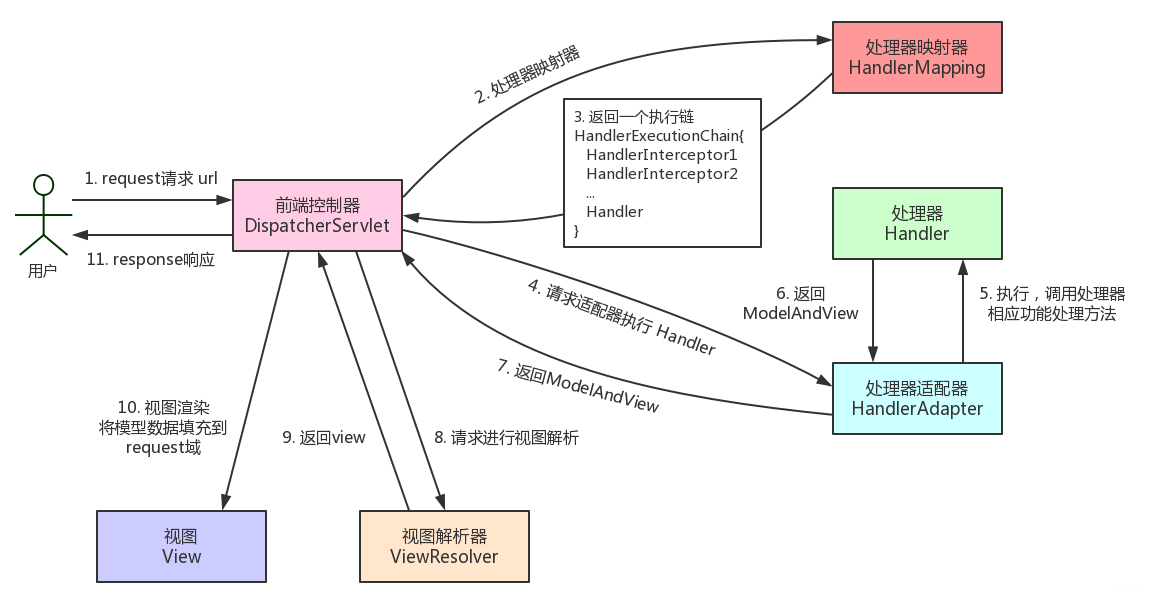


图4-4 MVC工作原理

## 4.3 系统功能模块设计

在需求分析中已经将角色功能分析完全，此处的系统功能模块设计主要是将所有功能分类整合，由于角色间的功能有所交叉重复，所以需要将角色功能进行分离整合，需要游离于角色之外，表现为单独的模块化，为后续程序开发提供功能性的区分隔离，该系统的具体体现如图4-5.

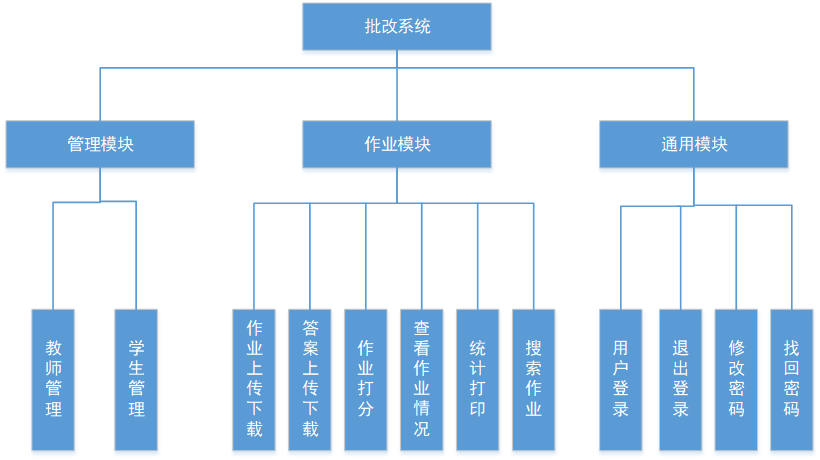


图4-5 系统功能模块

## 4.4 系统的数据库设计

数据库设计尽量做到精简完备，每个表的元素属性应当充分考虑所有情况，将表的属性对应到程序功能实现中去，也可以通过功能反推到数据库属性，比如我需要发送邮件找回密码，那么个人信息中就需要有邮件这一属性；再者作业需要打分，所以分数也是“作业“该表的属性之一。

数据库的设计还需遵循操作便利，提高访问效率，减少程序的负荷。

数据库的核心可以分为构建与操作，操作主要设计前后端交互，需要在程序中实现，数据库的构建需要合理建立表的数量和属性，设置好表之间的约束是其中关键。数据库表的约束可以分为五大约束，分别为：主键约束、唯一约束、检查约束、默认约束和外键约束。此处不再逐一解释分析，只分析使用了的约束。主键约束：唯一性的属性，能够完全区分该表下的所有成员。默认约束：初始化后的默认值，后续可修改，可以应用于初始密码等设计。外键约束：表与表之间元素的关系，可以进行并发控制。该系统运用外键约束在作业和学生作业的zyid（作业ID）中，比如该系统中的教师删除作业，那么学生作业也会同时删除，但这样设计会导致学生上传的作业会丢失，失去了为学生保存作业的初衷，此处仍有待考虑，但外键约束可以存在，其他外键约束详见E-R图。

最终的数据库构建由5个表构成，详见图4-6的用户、图4-7的学生、图4-8的班级、图4-9的作业和图4-10的学生作业。E-R图如图4-11所示。

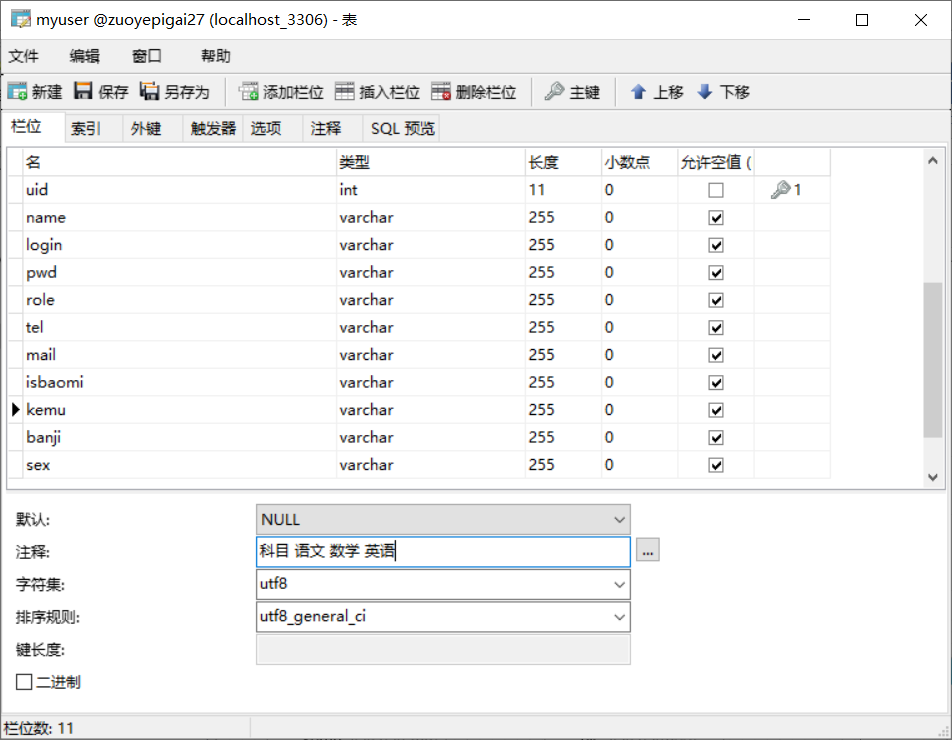


图4-6 用户

其中属性依次为：用户ID、用户姓名、用户登陆名、登陆密码、登录角色、电话、邮箱、电话是否对学生保密、执教科目、执教班级、用户性别

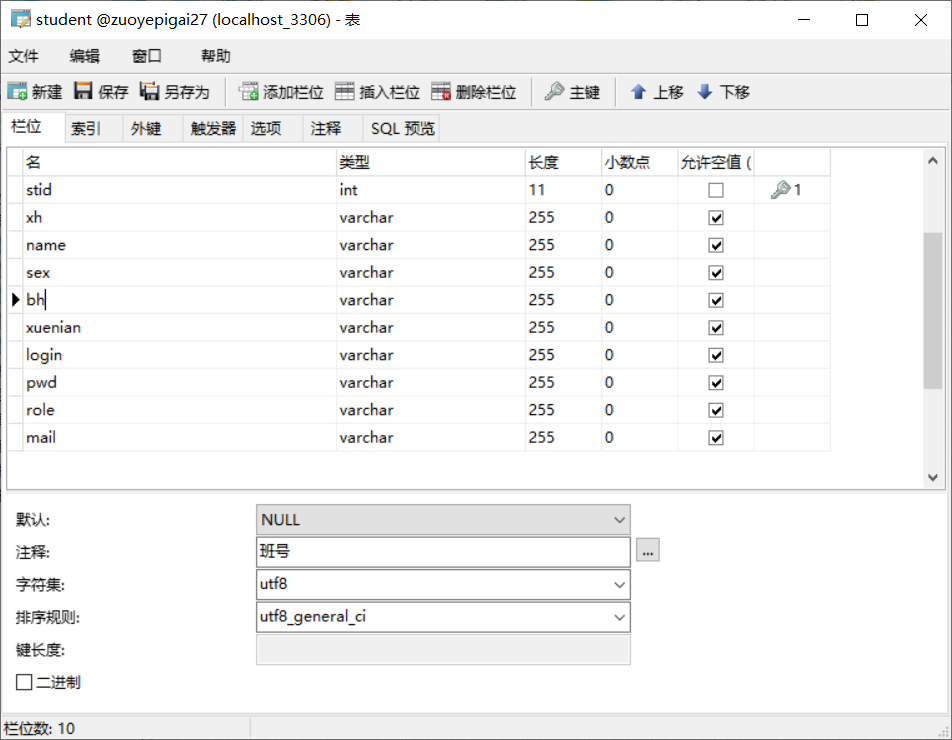


图4-7 学生

其中属性依次为：学生ID、学号、姓名、性别、班号、学年、登录名、密码、角色、邮箱

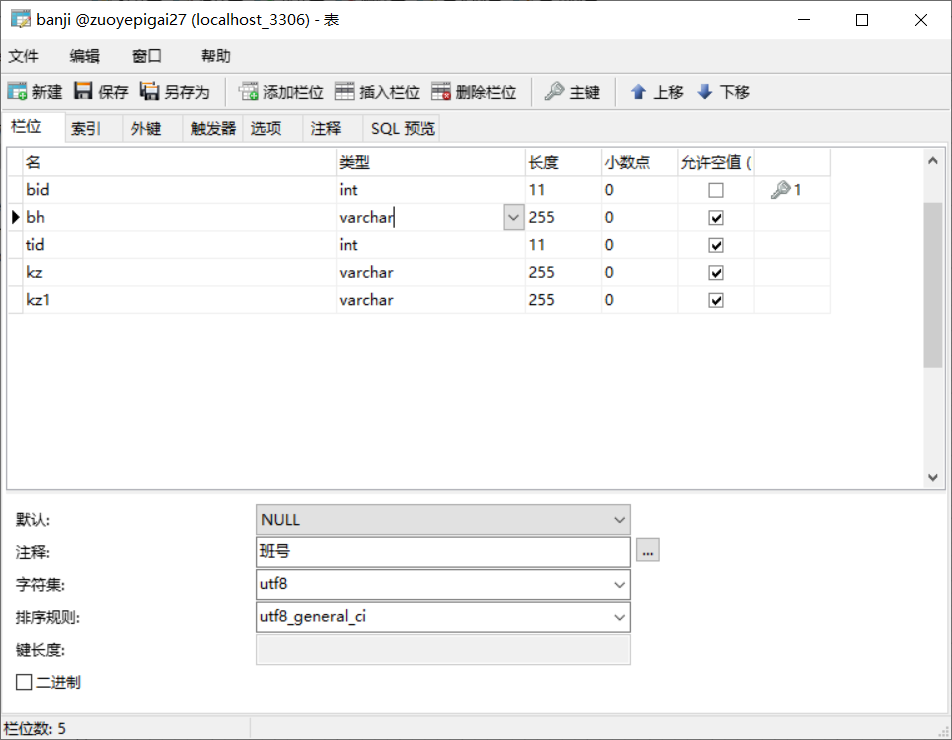


图4-8 班级

其中属性依次为：班级ID、班号、所属教师ID、预留项、拓展项

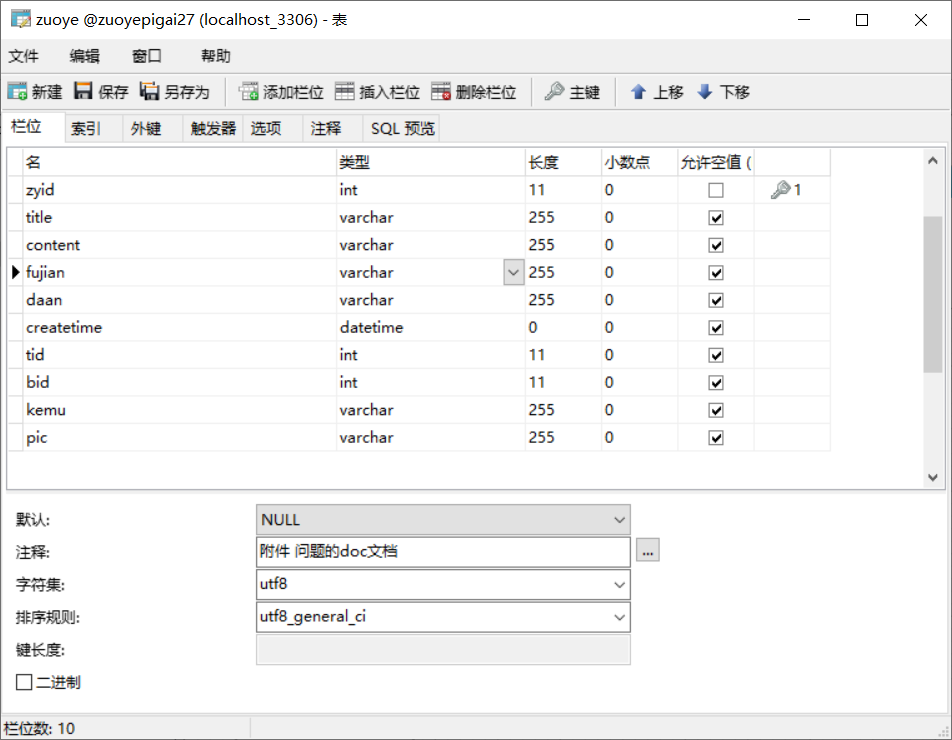


图4-9 作业

其中属性依次为：作业ID、作业标题、作业内容、作业附件、作业答案、作业创建时间、作业所属教师ID、作业所属班级ID、作业科目、作业截图

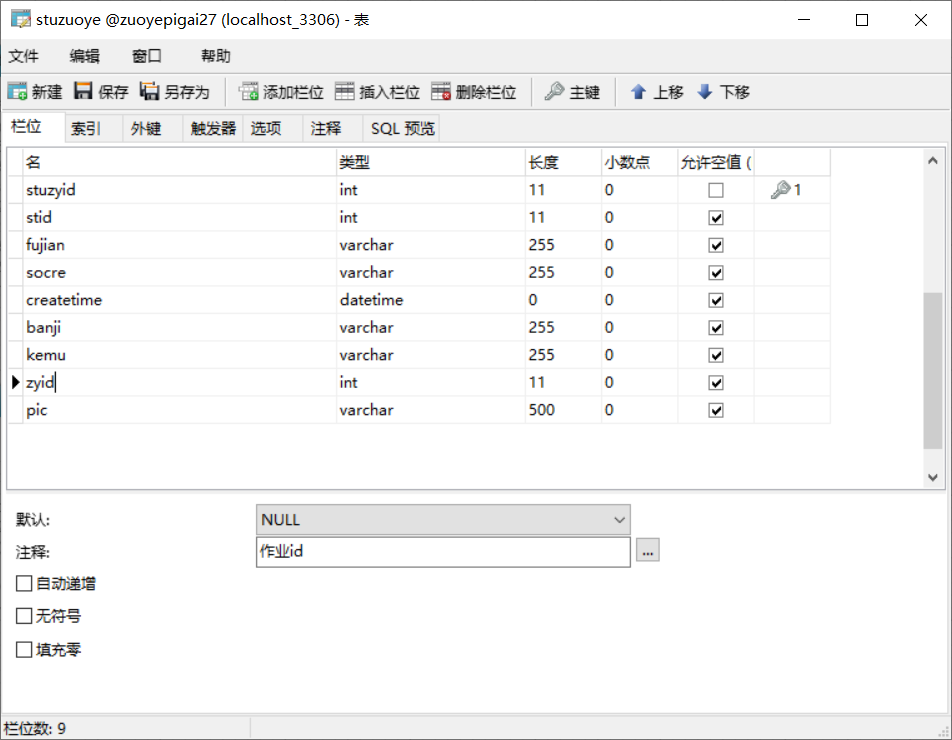


图4-10 学生作业

其中属性依次为：学生作业ID、学生ID、学生作业附件、学生作业分数、学生作业创建时间、学生作业所属班级、学生作业所属科目、学生作业所属作业ID、学生作业截图

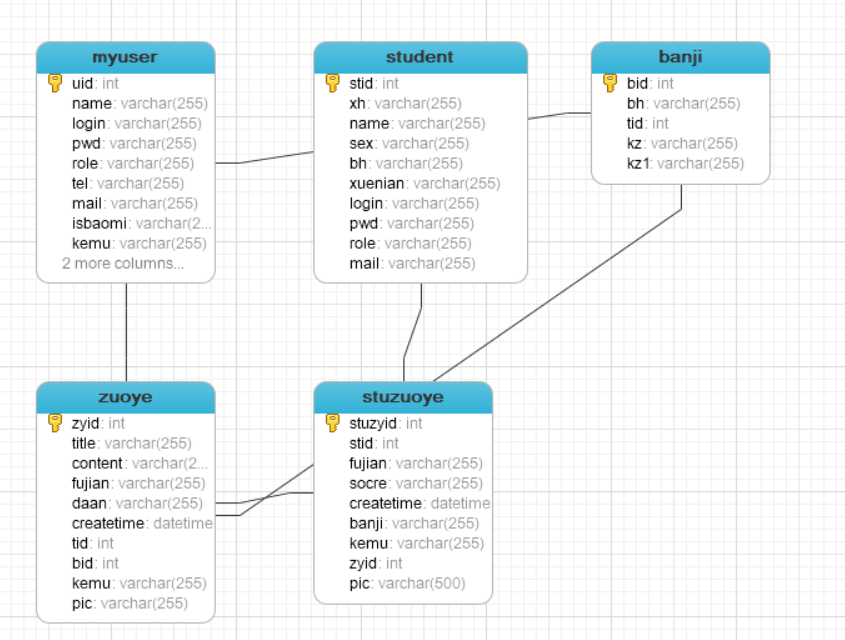


图4-11

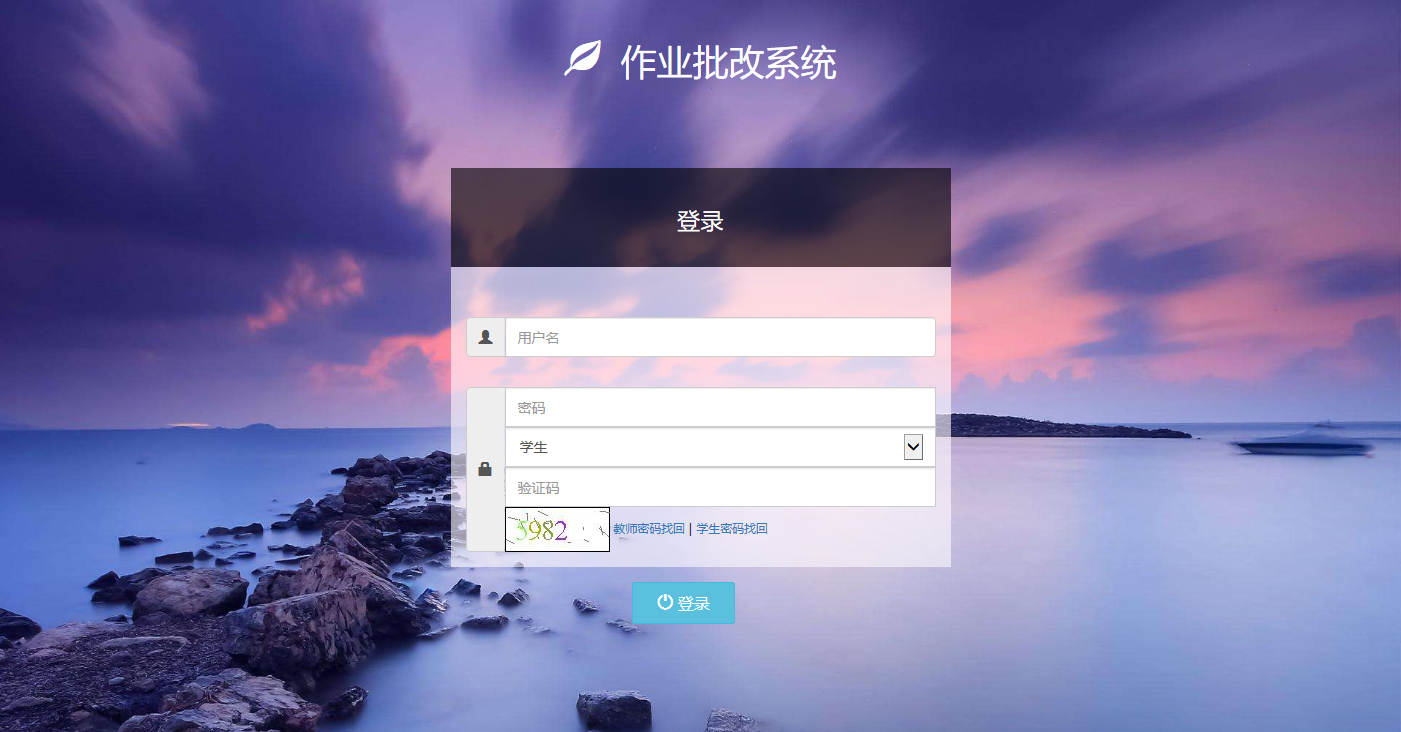
其中连线为外键约束。

## 4.5 系统的界面设计

界面的设计依赖web前端技术，构建体系完整的tml页面，其中按钮表格通过js修饰润色，也可以采用流行框架减少js工作量和使界面更加统一整洁。

界面的设计采用Bootstrap相关操作，主要涉及登陆界面、角色主使用界面和角色管理界面，由于页面涉及很多文件，并且使用该系统便可查看界面样式，所以以下将截取部分页面截图和前端代码进行界面设计的论文记录。

登陆界面、角色主使用界面和角色管理界面整体样式如图4-12、图4-13和图4-14。



4-12 登陆界面

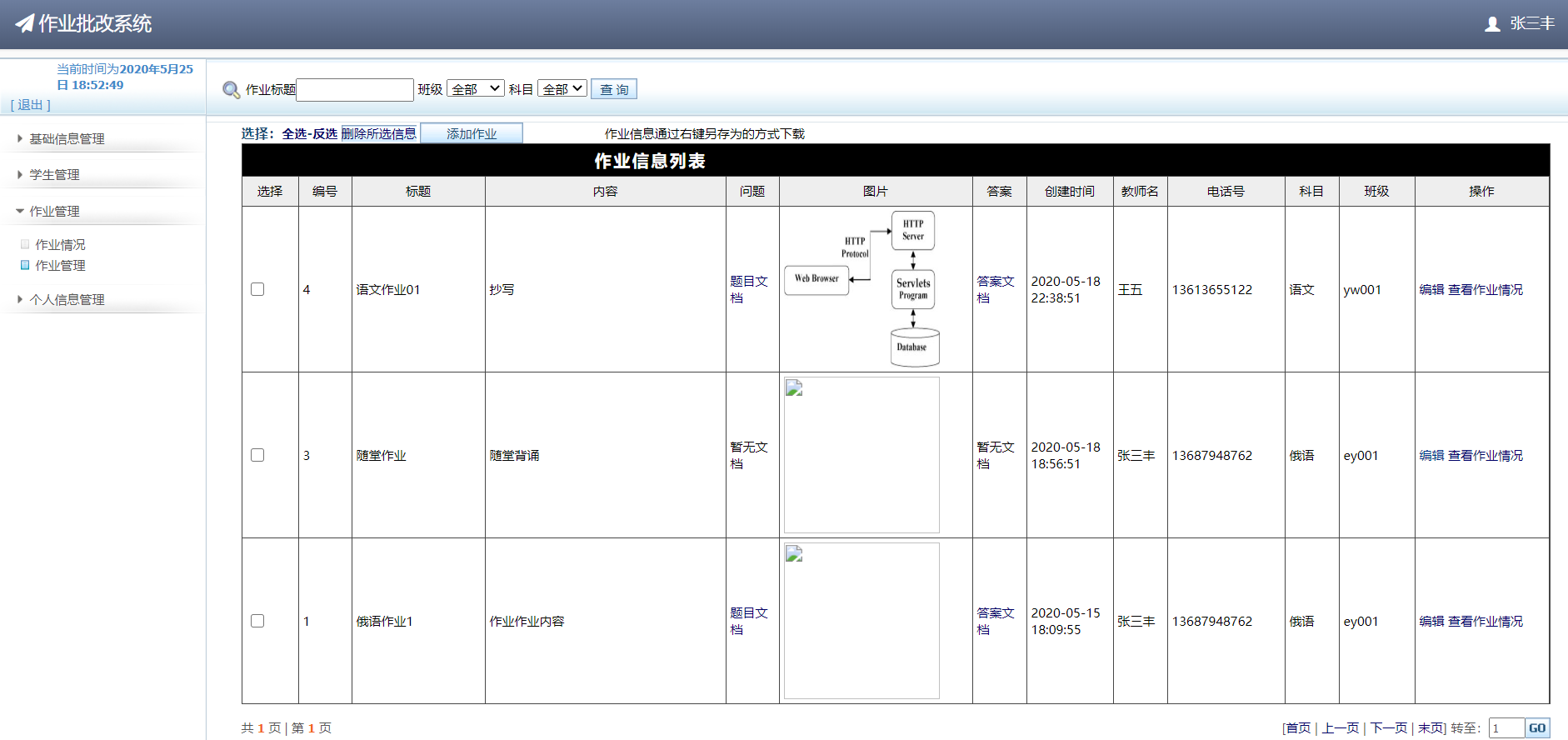


图4-13 角色主使用界面



图4-14角色管理界面

图4-15 jquery的使用 图4-16 点击切换验证码

## 4.6 本章小结

本章在需求分析之后将进度落实到系统整体设计，完成了数据库的设计、功能模块的分类和界面UI的框架样式。该部分具体完成的程序内容有数据库表的创建、约束和完善，也完成了界面的布局按钮和前端逻辑等。这些都是为了配合后续模块功能的实现，可以说是是整个项目的进阶基础，如果将程序开发比作修建一栋房子，那么需求分析便是设计图，本章的总体设计便是地基和墙围，模块功能便是水电房门等居家必备条件。在好的基础上进行功能实现能大大减少错误和工程量，并且后续额外功能或者功能改进都需要UI、数据库等方面的配套改进。

# 第五章 系统主要功能模块的详细设计与实现

## 5.1 管理模块的详细设计与实现

### 5.1.1 教师管理

教师管理主要由管理员对教师用户进行操作，具体操作包括：增删改查，教师信息修改，绑定教师班级。该系统除了邮件部分和退出部分等通用模块，其他大部分功能都是依赖于数据库的操作（访问、查询、修改、删除），其中便包括教师、学生、作业的数据库操作，于是都可以采用同一种设计思路来完成一系列功能不同但是实现方法类似的操作。考虑到都涉及对数据库的访问，所以构建以下模式：创建servlet下的.do方法，用来响应.jsp动态网页文件中的操作，具体由action实现，网页可以读取action操作，创建一个控制包，用来处理同一类功能发送的请求，其中函数声明该类问题的.Dao类型的实体，Dao为数据库的封装，其中包含对数据库访问操作的函数，由jdbc实现连接，再设置需要修改更新的数据为接口ID，在dao.xml中完成数据的更新和对数据库的修改。具体的程序运行过程如下。

教师的添加：web.xml中servlet实现.do方法，myuserAdd.jsp页面中调用myuserAdd.do，链接到src资源库中的MyUserController.java中，下列代码是该处具体实现 @RequestMapping("files/addMyuser.do")//添加信息

public String addMyuser(Myuser myuser ,HttpServletRequest request){

int result= md.insertSelective(myuser);

if(result>0){

request.setAttribute("message","<script>alert('操作成功!')</script>");

}，其中 md便是新建的MyuserDao，用来控制封装数据库操作，设置MyuserDao（封装）中的insertSelective作为ID，在MyuserDao.xml（专门处理用户操作）完成处理，其中包括Myuser.java中对用户的属性进行设置和初始化函数。界面展示如图5-1所示

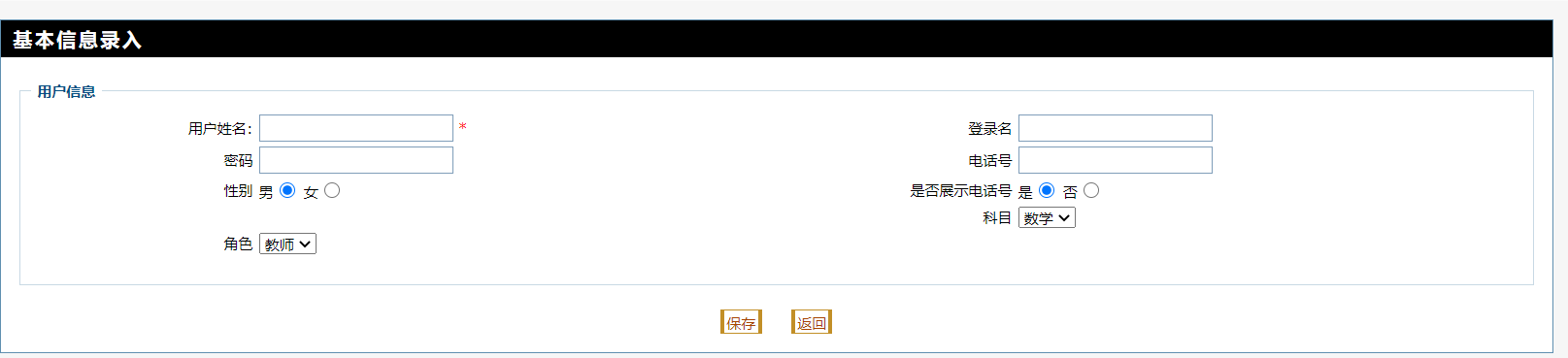


图5-1 添加教师

该流程实现了教师的添加功能，但是如果所有功能都单独实现的话，程序将会变得混乱不堪，所以还需要将同一类型的访问整理为同一种操作的整合，就像该处的教师添加是以上流程，而教师的删除虽然不是完全一致，但在类型上两者属于同种操作方式，于是将对教师的的操作尽可能的归于一类，减少文件数量，为能够共通使用或者共同传递的部分增加函数，来达到控制一类的操作。

这样应用下来，既能精简代码，也能优化程序结构，使其易于理解。比如该功能模块下的其他功能，教师信息的删除和修改，都可以归为教师数据的更新，同样在用户控制中访问更新函数：

MyuserQuery cq=**new** MyuserQuery();

cq.createCriteria().andUidEqualTo(myuser.getUid()) ;

**int** result= md.updateByExampleSelective(myuser, cq);

并通过.Dao完成数据库更新。

同理教师和班级的绑定也是如上述操作一致，是不过不是用户的部分而是班级管理部分，其他实现原理都一致，所以此处不再赘述。

### 5.1.1 学生管理

学生管理仍然是管理员的职着所在，其需要实现的功能和教师几乎一致，主要功能仍然是增删改查，所以它与上述教师的实现方法仍然相同。而该系统其实也正好将所有核心功能整合后分为了7大类类似功能，包括上述三种在内的教师用户控制、班级控制、学生控制、文件上传控制、作业控制和学生作业控制这7类，其他的诸如下载、邮件、查询和表格显示内容等部分由于功能单一，不需要进行整合操作，所以另作讨论，并会在后续具体实现中分析记录。

学生管理的效果图如图5-2和图5-3所示，



图5-2 学生添加



图5-3 学生管理

## 5.2 作业模块的详细设计与实现

### 5.2.1 作业上传下载

作业上传下载是该系统的核心功能之一，其主要技术涉及文件的上传和下载，由于作业布置也属于文件上传，应当归为一类操作进行设计，将教师布置作业和学生提交作业的页面设置为如图5-4和图5-5所示，

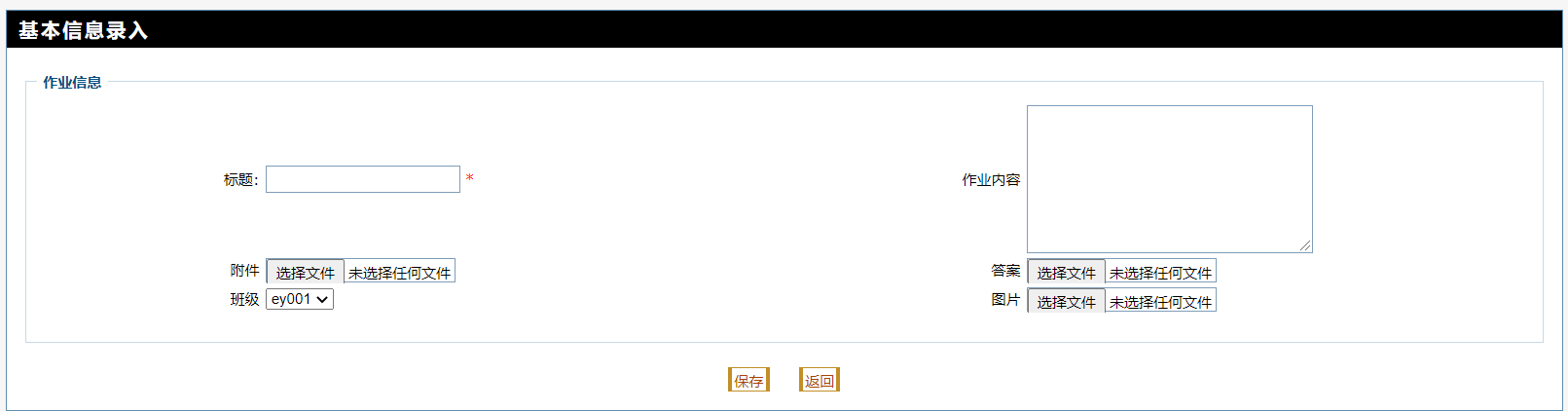


图5-4 作业布置

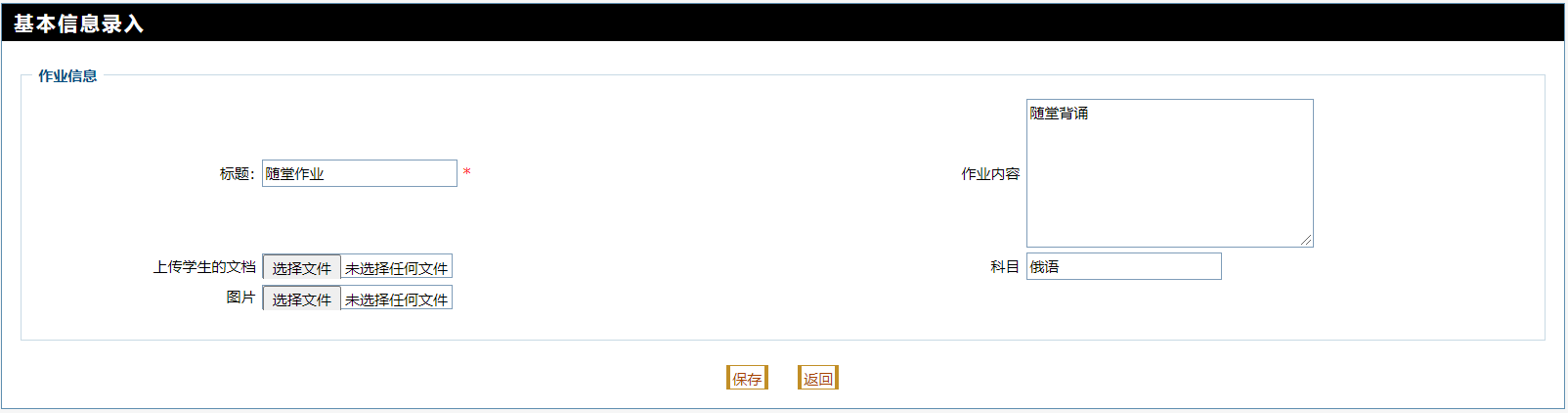


图5-5 作业提交

作业包含标题和作业内容两个文本框用于表述作业，剩下的部分都属于文件上传内容，文件类型支持图片、压缩包、文档等，其中右下角图片表示可以在网页预览。

上传的具体实现如下：如上小节所述，作业的管理也可以归为一类，虽然需要实现的功能不同，但是实现功能的步骤一致，都是为对应功能建造控制函数，再由控制函数链接Dao对数据库进行操作，此处文件上传下载也遵循该模块方法，所以该框架不需赘述，此处需要讨论分析的是文件上传下载的具体实现：

@RequestMapping("files/addZuoye.do")

public String addZuoye(Zuoye zuoye, @RequestParam("file") MultipartFile file ,@RequestParam("daanfile") MultipartFile daanfile,@RequestParam("tu") MultipartFile tu ,HttpServletRequest request)

此处仅给出文件上传函数名，完整函数详见附录1，该函数正常运行后，仍通过Dao对数据库完成操作。

作业的下载包括教师下载学生提交的作业和学生下载教师布置的作业，由于作业在服务器中是以文件存储，并且在服务器中有明确的链接（.fujian），该属性是依赖于一个list（将作业的所有元素整合归类为一个list，再对其中的属性访问操作），于是将作业设置为直接链接到附件，由浏览器实现对文件的另存为，完成下载的功能。

### 5.2.2 答案上传下载

答案的上传下载和作业的上传下载一致，只是角色只有教师，将学生上传作业下载答案和教师上传答案下载学生作业看成相反的情况，可以更好的理解它们之间的关系，其中具体的操作仍然和上小节一致，请类比理解，此处不再赘述。

### 5.2.3 作业打分

打分的实现存在于学生作业控制中，后续仍然通过Dao对数据库进行访问修改，其函数名如下，具体函数实现详见附录2.

@RequestMapping("files/showStuzuoyedetailsocre.do")//上分操作

public String shangfenshowStuzuoyedetail(Integer stuzyid,HttpServletRequest request)

### 5.2.4 查看作业情况

查看作业情况指的是教师在网页列表中查看学生的作业提交情况，包括下载学生作业，对学生作业打分，分类查看作业，搜索作业。由于搜索包含作业的搜索和用户的搜索，所以单起一小节进行模块构建，下载作业和为作业打分也在上一小节讨论分析过，所以此处仅讨论其他功能的实现，包括作业表格的显示和作业的分类查看。主要函数如下，实现效果如图5-6

作业的分类查看本质上是对数据的搜索，只不过将分类做成下拉菜单并链接教师对应的班级，此处需要对要查找的数据进行约束，否则可能出现不满足分类的情况。具体搜索的实现详见5.2.6的作业搜索功能。



图5-6 查看作业情况

### 5.2.5 统计打印

打印功能主要是对页面的打印并保存，对统计具有一定帮助，采用preview(oper)函数完成打印功能，详细代码请查看附录3。

### 5.2.6 搜索作业

由于作业在网页上是存储关联在一个构建好的list中，list包含作业的所有属性，于是搜索作业便是使list中的属性满足要求，核心函数如下。

作业<selectname="zyid" >

<option value="">全部</option>

<c:forEach items="${zylist }" var="z">

<option value="${z.zyid }" <c:if test="${z.zyid==zyid }"> selected </c:if>>${z.title }</option>

</c:forEach>

</select>

<input name="Submit" type="submit" class="right-button02" value="查 询" /> </td>

## 5.3 通用模块的详细设计与实现

### 5.3.1 用户登录

用户登录在我的设计中大概分为4部分：界面设计、登录拦截、验证码和密码找回。界面设计在上一大章已经分析过，密码找回详见5.2.4小节，验证码详见5.4小节额外模块的详细设计与实现。本小节着重讨论登录拦截的实现。

通过对登录按钮绑定函数，在函数中读取用户输入的数据，再和数据库中数据进行比较，实现登录拦截。

### 5.3.2 退出登录

用户的退出包含退出当前页面并清空密码，于是设置实现的方式为:直接跳转到登录界面，并清空个人数据。

### 5.3.3 修改密码

修改密码涉及到数据库的更新，所以仍然可以使用用户控制，只需在执行时添加一步判定条件，即输入的原密码正确。其中判别的函数check（）如下:

function check(){

var pwd1=document.getElementById("pwd1");

var oldpwd=document.getElementById("oldpwd");

if(pwd1.value!=oldpwd.value){

alert("原密码不正确");

return false;

}else{

document.getElementById("fom").submit();

}

### 5.3.4 找回密码

找回密码基于实际分析，我设置了两种不同的密码找回方式，分别用于教师和学生的密码找回。

结合实际情况并且减少功能逻辑，教师的手机号是对学生保密的，所以首先考虑采用手机号验证的方式找回教师密码，但是教师熟人会知道教师的密码，该方法具有弊端，又考虑到教师的登录名并非教师本名，所以最终决定采用教师验证登录名和手机号的方式进行密码找回，而登录名的修改需要管理员的操作。该方法的好处是教师不需要预留邮箱，且找回密码十分方便。实现效果如图5-7。

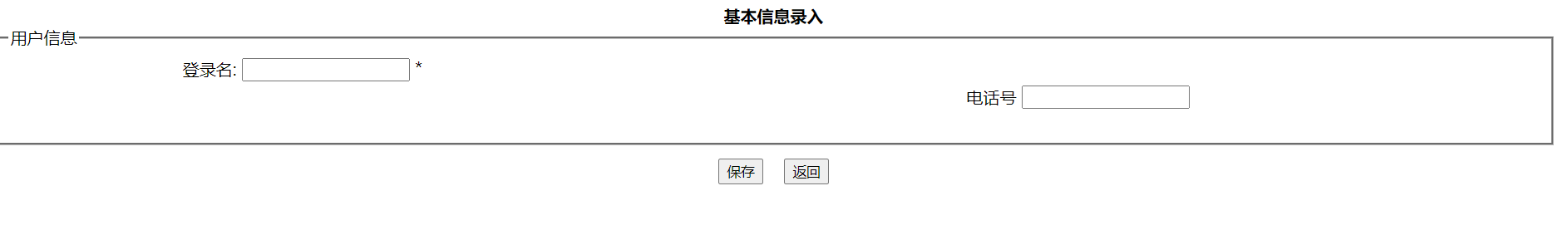


图-5.7 教师密码找回

学生的密码找回需要做额外的考虑，因为涉及到登录他人账号的隐私问题，于是此处设计成通过邮箱找回密码，邮箱由管理员为学生录入。具体实现方法是：学生输入正确的登录名和学号进行邮件密码重置，判定无误后，系统向该学号的邮箱发送随机密码并一并重置其登陆密码，由于只有该学号的邮箱会接收到密码，所以安全性得到了保障。邮件发送具体实现详见附录4，实现效果如图5-8

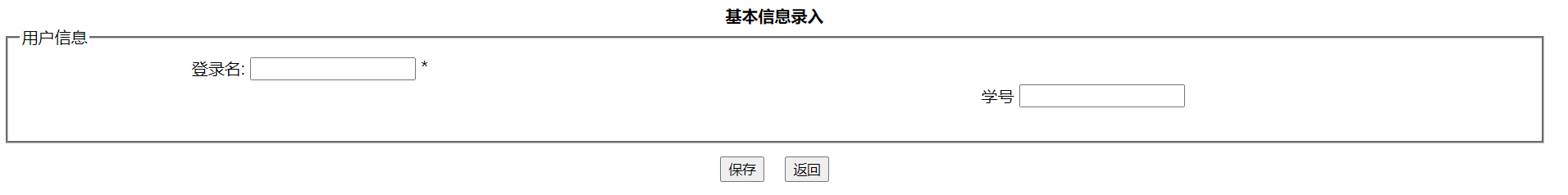


图5-8 学生密码找回

## 5.4 额外模块的详细设计与实现

额外功能不是程序必备的功能，但是对于程序的使用有很大的提升，此处主要探讨分析两类功能：验证码实现和网页表格的分页效果。

<img id="yzmimg" src="randomCode.do" onclick="changeVcode()" class="code" style="cursor:pointer;"/>

function changeVcode(){

$("#yzmimg").attr("src","randomCode.do?abc="+Math.random());

//document.getElementById("yzm").src="randomCode.do?abc="+Math.random();

//alert(document.getElementById("yzm").src);

}

上述为验证码和点击刷新验证码的实现。分页效果的代码详见附录5.

## 5.5 本章小结

本章的内容几乎涵盖了程序的所有功能方面，确保了程序对需求的实现。虽然不同模块间的数据少有交互，但是他们对数据库的操作都有类似的地方，于是根据实际情况构建了一系列的功能实现，跳出角色的思维局限，将相近功能归为一类有助于程序的逻辑明确，纠错改进也更加方便。

# 第6章 系统测试

## 6.1 系统测试方法介绍

系统测试是检验程序功能是否完备，检测程序逻辑是否正常等的必要条件，往往测试才能发现以前疏忽的问题，或者探讨出更高效的解决方法。以下是部分常用系统测试方法：

1、按测试对象分类

        白盒测试：软件底层代码功能实现，同时逻辑正确

        黑盒测试：测试软件外在功能是否可用（点点点）。

        灰盒测试：介于两者之间（接口测试）

2、按测试对象是否执行分类

        静态测试：测试对象不执行，侧文档

        动态测试：将软件运行在真实环境当中

3、按测试手段进行分类

        手工测试：由测试人员手动的对被测对象进行验证，优点，就是可以灵活的改变测试操作和环境

        自动化测试：两种形式：一种是自己写测试脚本，另外一种是通过第三方测试工具对被测对象进行测试。

              优点：高效率完成人不能做的测试。

              例如：同一时间，接口压力测试。

## 6.2 系统的功能测试

针对系统的功能逐一进行测试，力求做到测无可测，发现程序的bug和逻辑上的错误，对出错的地方及时进行改进和设计。由于该程序是供用户使用的网页项目，所以测试的重心应该放在用户的功能完全、体验良好上，以下将按照此思路对程序的功能依次展开测试。

1. 管理员对学生教师的增删改查

测试方法：通过管理员界面对教师、学生信息一次进行增删改查

测试结果：数据库符合增删改查的数据变化，并且信息界面能够反映信息的更新

1. 数据库关系约束测试

测试方法：删除与某项元素相关联的其他元素，或对其更新

测试结果：该元素受其他元素的变化影响，能够产生对应更新

1. 作业答案上传下载测试

测试方法：本地创建合理文件，进行上传下载

测试结果：上传下载功能完备无误

1. 教师打分测试

测试方法：教师为提交作业打分

测试结果：教师打分后学生能够查看分数

1. 答案查看测试

测试方法：教师待学生作业上传完毕后上传答案

测试结果：学生能够下载答案

1. 作业搜索测试

测试方法：设置不同搜索变量

测试结果：搜索结果符合预期，功能完备

1. 用户登录测试

测试方法：故意输错用户名、密码或者验证码

测试结果：程序能够对不同错误进行不同反应

1. 修改密码测试

测试方法：修改学生密码

测试结果：学生能够通过邮件接收新的随机密码

1. 找回密码测试

测试方法：教师输入用户名和手机号找回密码

测试结果：教师成功找回密码

1. 页面跳转测试

测试方法：点击不同按钮进行页面切换

测试结果：页面切换无误，能够正常跳转

1. 打印测试

测试方法：在学生作业情况页面进行打印

测试结果：能够正常打印并下载为图片

## 6.3 系统的性能测试

性能测试和功能仍有联系，好的功能逻辑能够减轻程序的额外负担，相当于增加了系统的性能。

性能在程序中可谓是处处都有体现，比如架构是否合理、数据库设计是否合理、代码是否存在性能问题、是否有不合理的内存使用、是否有不合理的线程同步操作、是否有不合理的资源竞争、算法是否已经是最优等。该系统主要应用于学校师生，面临着流量多负荷大，同时访问人数可能具有峰值等常见问题，所以对其性能的测试主要分为两点：其一为系统本身的数据处理能力，其二为对网络流量的应对能力。

下列是部分主要性能元素：

吞吐量（TPS, QPS）：简单来说就是每秒钟完成的事务次数或者查询次数。通常吞吐量越大表明系统单位时间能处理的请求数越多，也就是性能越高，所以通常希望TPS越高越好

响应时间：即从请求发出去到收到系统返回的时间。响应时间一般不取平均值，而是要去掉不稳定的值之后再取均值，比如常用的90%响应时间，指的就是去掉了10%不稳定的响应时间之后，剩下90%的稳定的响应时间的均值。从聚类的观点看，其实就是去掉离群点。

错误率：即错误请求数与总请求数之比。不能处理的事务请求一般也会算作错误率，随着压力增加，有可能出现处理请求处理不过来的情况，这时错误率也会不断上升。

上述三者有极大的关联，任何孤立的数据都不能说明问题。典型的关系是，吞吐量增加时，响应延迟有可能增加，错误率也有可能增加。因此，单拿出一个很大的TPS并不能说明系统的性能问题。

**性能调优的思路**

一般情况，调优需要有个前提条件，即无论是用线上的真实流水还是线下的压力测试让问题扩大化，明显化。

根据这些比较明显的现象去初判问题，收集证据去验证初判结果成立，然后分析现象产生的原因，并尝试解决问题

## 1.性能摸底测试

## 对于新上的系统或者是有过较大代码改动的系统来说，做一次摸底测试还是很有必要的。一般来说，期望摸底的测试是一次对单机的压力测试。压力测试可以帮你大概搞清楚系统的极限TPS是多少，在压力上来时有没有暴露一些错误或者问题，系统大致的资源占用情况是什么，系统可能的性能瓶颈在哪。

## 如果进行一次摸底测试，分析其配置和结果。假如用12000并发用户对10台机器压测的结果，可以估算，TPS到7w多，平均响应时间为82ms，错误率在2.5%。

## 进行更深的分析的话，首先，TPS在后期迅速下落，实际上已经支撑不了如此大的并发量，即进入崩溃区，这里有几个可能，一是系统根本承受不了如此大的并发量，二是系统中间有问题导致TPS下跌。其次，随着时间增长，错误率显著增加，说明系统已经处理不了如此多的请求。结合前面两点以及相对平稳的平均响应时间，大致可以推断系统没法承受如此大的并发。

## 对于应用的特点，也要在这时候分析出来，即应用可能占用的资源。比如是CPU密集型应用还是IO密集型应用（还可以细分为是磁盘密集还是网络 ）

## 2.定义性能优化的目标

## 性能的优劣可以说是因设备环境而异，若用户对性能的需求比较低，那么低性能也可能满足，反之亦然。所以具体的性能分析一定要结合环境和用户需求，制定好规范的标准。

## 性能优化的目标一般是吞吐量达到多少，90%响应时间小于多少，错误率小于多少。同时还需要关注其他的性能指标，cpu使用情况，内存使用情况，磁盘使用情况，带宽使用情况等。对于摸底测试已经发现问题的，可以针对该问题专门优化，比如负载较高，cpu消耗过大，则目标可能是TPS，响应时间以及错误率不变的情况下降低CPU负载。或者内存增长过快，gc较为频繁，则目标可能是找出可能的内存泄露，或者进行相关的jvm内存调优。总之，目标可以比较灵活调整，但一定要明确。

由于该系统还未投入使用，现今都是在本机服务器上运行，所以对于网络的测试无能为力，对本机的程序性能能够从页面的跳转速度和数据处理速度上可见一二。这些可以结合上一小节的功能测试，基本可以确定没有问题。

## 6.4 本章小结

人们认为能够发现程序缺陷的测试是成功的测试，测试的根本目的就是为了发现尽可能多地缺陷。然而不幸的是，这种对软件测试过分单一的阐述和解释会带来两个原则性的问题。

首先，尽可能早的发现尽可能多的bug，会使软件测试成为一个数字游戏。大量的bug数量的统计会意味着软件测试的工作做的特好？大量的bug数量并不一定意味着测试的结果是最重要的关键问题被越早被发现, 另一个潜在的方面，简单的尽可能早的发现尽可能多的bug将导致貌似bug统计数量的爆炸，这是因为许多虚报或者重复的bug也被统计在内了。缺陷表现在许多方面。如果一个测试这部花费时间对导致bug的原因作认真的调查研究，那就有可能导致对同一个错误根源引起的若干个bug作若干个bug报告。不幸的是，许多测试人员（不一定是新手）经常坚信他们越早发现越多的bug可以改善软件质量。请记住，我们并不能测试软件质量！

其次, 当测试工程师集中精力寻找更多的错误，他们往往跳过一些不容易发现错误的地方或者想当然认为一些地方没有错误，从而使软件测试覆盖率降低。有证据表明，许多测试人员由于太过专注于发现重大或者重要的错误，往往忽略过一些极易发现错误的所谓简单地方。比如，在测试边界条件的时候，测试人员会简单的在边界条件有效值范围内指定最小值、最大值和中间值来做测试，如果通过则认为没有问题；但这样则错过了超出边界条件的无效值的验证。比如，最小值减一（Min-1）和最大值加一（Max+1），这恰恰是最容易出现错误的地方。

测试是程序面世必不可少的一环，可谓是离成功最近的一步，本章着重针对功能一次进行了详尽的测试，已经确保所有功能完整且逻辑正确合理，能够应对用户的各种需求。

# 第7章 结束语

## 7.1 论文工作总结

本文分析了“学生作业批改系统”的需求，针对该项目需要解决的实际问题，进行了充分的探讨和分析理解，讨论了各种情况下角色功能的具体实现，通过进行的测试对程序功能进行了改进和取舍，在进度满足时间安排的情况下，同时推动论文的记录和进展，最终编码完成了该“学生作业批改系统”，使其能够应用于实际教学生活中，完成了个人任务。

从一开始的毕设选题到现在为答辩准备，这段时间借由该选题我真的学到了很多，刚开始的时候只仗着自己有过相关开发的经验，所以贸然选下该题目，其实后续的功能实现和问题实在是困扰了我许久，不过还好最终都能有所解决。刚开始许多地方的技术都有所欠缺，需要恶补，也有看了这个忘了那个的情况发生，因为空在纸上学习技术，没有经过练手终究没有理解那么透彻，这样造成的后果便是在一开始的编码中，经常出现一些小错误，比如什么拼写错误，少打了括号或者其他符号，这些还好，那些对程序有错却检查不出来的更令人沮丧。但不可否认的是，经过这次毕设的锻炼，我的技术得到了实打实的进步，虽然离老练的程序员那种高效的开发还有很大距离，但我真的学到了解决问题的方法，那便是多查多问多试，尽管这个过程称得上枯燥乏味，也让人心烦意乱，但在编程的世界里，这便是最好的方法。程序之外便是论文的撰写工作，此前我最大的报告文字量也未超过5000字，但论文的撰写不仅要求字数，要求低重复率，还要求对所有内容的科学分析，这一点是平时报告所没有的，而且论文的格式规范要求十分严格，这也是一种对严谨的科学态度的尊重。总而言之，通过这次毕设，我成长了很多，是技术上的，也是心灵上的。

## 7.2 问题和展望

尽管该系统的功能已经足够完善，能够满足用户的所有需求，但是细节之处和某些功能仍然与实际要求有不少差距，问题能催人向前，给人以更好的展望。

仍未实现的功能：新教师和新学生自己注册的功能实现，教师批量下载学生作业，教师界面显示所有作业（能够通过分类查看显示自己班级，有利有弊），由于该系统未实际投入使用，所以并发控制也未曾考虑。分析上述问题，主要是一个应用环境的问题，我的软件是做出来给一群人用的，可我的设备不支持当前的活动，于是我便手足无措，这也是个人开发的弊端之一。

展望：后续如果要应用于学校实际生活教学，首先需要完善上述功能，然后需要移植到商业服务器，并且需要对界面进行优化。此外还需要考虑当前无法应用的功能，比如数据安全，比如网络负荷，学生作业保存是否需要设置时限，教师删除作业后学生还能否从网页上下载自己的作业等等一系列问题......问题多不可怕，可怕的是找不到解决的办法，一是个人对技术的理解运用还很低，二是对问题的应变很差，所以与其说对该项目的展望，不如说是我对我自己的展望，便是不断精进，成就自我。

# 参考文献

[1]张海藩、牟永敏.软件工程导论.清华大学出版社.2013，（6）

[2] David Flanagan，美. JavaScript权威指南. 机械工业出版社2011,(6)

[3]郭欣.构建高性能Web站点. 电子工业出版社.2009

[4] Brian Goetz等.Java并发编程实战. 机械工业出版社.2012

[5]李兴华、王月清.Java Web开发实战经典. 清华大学出版社.2010

[6]JonathanChaffer、KarlSwedberg. jQuery基础教程.人民邮电出版社.2008

[7] Trevor Burnham. JavaScript异步编程. 人民邮电出版社.2013

[8] Craig Walls. Spring实战. Manning.2011

[9] 苏航.接口的意义及在Javaweb三层架构中的作用分析[J]. 西部皮革，2016,（24）

[10] 李鹏博，于立婷，王天琪.Javaweb软件框架技术探析[J]. 通讯世界，2016,（14）

[11] 陈涛，黄艳峰.Javaweb开发中文件上传方法研究与实现[J]. 电脑知识与技术，2016

[12] 张圣筛，陈家琪.基于Javaweb的高校学生考勤管理系统设计与实现[J]. 软件导刊，2015,（12）

# 致 谢

路漫漫其修远兮，吾将上下而求索。四年求学生涯，由毕设画上一个句点，我由衷的感谢学校对我的栽培，日后成才定当不忘校恩！

万千不舍情，倒是离别日。与吴老师相处虽短，那句句的教诲，字字的传授，都让我受益匪浅。十分感谢吴老师的信任，和在毕设上给予我的帮助，再次，我谨向吴老师献上十二分的谢意！

# 附 录

附录1：

@RequestMapping("files/addZuoye.do")//添加信息

public String addZuoye(Zuoye zuoye, @RequestParam("file") MultipartFile file ,@RequestParam("daanfile") MultipartFile daanfile,@RequestParam("tu") MultipartFile tu ,HttpServletRequest request){

Myuser m=(Myuser)request.getSession().getAttribute("dangqianyonghu");

if(m==null){

request.setAttribute("msg","<script>alert('登录超时请重新登录!')</script>");

return "../login.jsp";

}

zuoye.setTid(m.getUid());//当前登录的教师可以留作业

zuoye.setCreatetime(new Date());

zuoye.setKemu(m.getKemu());

String path= request.getSession().getServletContext().getRealPath("/upload");//获取 文件夹的路径

System.out.println("path:::"+path);

int x=(int)(Math.random()\*999);

if(!file.isEmpty()){

String fname= file.getOriginalFilename();//获取上传文件名称 并且以当前用户编号+项目编号做区分

if(fname.lastIndexOf("\\")!=-1){

fname=fname.substring(fname.lastIndexOf("\\")+1);

}

System.out.println("fname"+x+fname);

File f=new File(path+"\\"+x+fname);//将文件上传到 对应的位置

try {

file.transferTo(f);

}catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

zuoye.setFujian("../upload/"+x+fname);

}else{

zuoye.setFujian(null);

}

if(!daanfile.isEmpty()){

String fname= daanfile.getOriginalFilename();//获取上传文件名称 并且以当前用户编号+项目编号做区分

if(fname.lastIndexOf("\\")!=-1){

fname=fname.substring(fname.lastIndexOf("\\")+1);

}

System.out.println("fname"+x+fname);

File f=new File(path+"\\"+x+fname);//将文件上传到 对应的位置

try {

daanfile.transferTo(f);

}catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

zuoye.setDaan("../upload/"+x+fname);

}else{

zuoye.setDaan(null);

}

if(!tu.isEmpty()){

String fname= tu.getOriginalFilename();//获取上传文件名称 并且以当前用户编号+项目编号做区分

if(fname.lastIndexOf("\\")!=-1){

fname=fname.substring(fname.lastIndexOf("\\")+1);

}

System.out.println("fname"+x+fname);

File f=new File(path+"\\"+x+fname);//将文件上传到 对应的位置

try {

tu.transferTo(f);

}catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

zuoye.setPic("../upload/"+x+fname);

}else{

zuoye.setPic(null);

}

int result= zd.insertSelective(zuoye);

if(result>0){

request.setAttribute("msg","<script>alert('操作成功!')</script>");

}else{

request.setAttribute("msg","<script>alert('操作失败!')</script>");

}

return "redirect:showZuoye.do";

}

@RequestMapping("files/toAddZuoye.do")//添加信息

public String toAddZuoye(Zuoye course ,HttpServletRequest request){

Myuser m=(Myuser) request.getSession().getAttribute("dangqianyonghu");

if(m==null){

request.setAttribute("message", "<script>alert('登录超时请重新登录')</script>");

return "../login.jsp";

}

//只能查询当前教师的班级信息

BanjiQuery bq=new BanjiQuery();

bq.createCriteria().andTidEqualTo(m.getUid());

List blist= bd.selectByExample(bq);

request.setAttribute("blist", blist); //班级列表

return "zuoyeAdd.jsp";

}

@RequestMapping("files/delZuoye.do") //删除信息

public String delZuoye(Integer[] ids ,HttpServletRequest request){

if(ids!=null&&ids.length>0){

ZuoyeQuery gq=new ZuoyeQuery();

gq.createCriteria().andZyidIn(Arrays.asList(ids));

zd.deleteByExample(gq);

}

return "redirect:showZuoye.do";

}

@RequestMapping("files/showZuoyedetail.do")

public String showOneZuoye(Integer zyid,HttpServletRequest request){

Zuoye g= zd.selectByPrimaryKey(zyid);

request.setAttribute("g", g);

return "zuoyeUpdate.jsp";

}

@RequestMapping("files/updateZuoye.do")

public String updateZuoye(Zuoye zuoye, @RequestParam("file") MultipartFile file,@RequestParam("daanfile") MultipartFile daanfile ,@RequestParam("tu") MultipartFile tu,HttpServletRequest request){

Myuser m=(Myuser)request.getSession().getAttribute("dangqianyonghu");

if(m==null){

request.setAttribute("msg","<script>alert('登录超时请重新登录!')</script>");

return "../login.jsp";

}

zuoye.setTid(m.getUid());

String path= request.getSession().getServletContext().getRealPath("/upload");//获取 文件夹的路径

System.out.println("path:::"+path);

int x=(int)(Math.random()\*999);

if(!file.isEmpty()){

String fname= file.getOriginalFilename();//获取上传文件名称 并且以当前用户编号+项目编号做区分

if(fname.lastIndexOf("\\")!=-1){

fname=fname.substring(fname.lastIndexOf("\\")+1);

}

System.out.println("fname"+x+fname);

File f=new File(path+"\\"+x+fname);//将文件上传到 对应的位置

try {

file.transferTo(f);

}catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

zuoye.setFujian("../upload/"+x+fname);

}else{

zuoye.setFujian(null);

}

if(!daanfile.isEmpty()){

String fname= daanfile.getOriginalFilename();//获取上传文件名称 并且以当前用户编号+项目编号做区分

if(fname.lastIndexOf("\\")!=-1){

fname=fname.substring(fname.lastIndexOf("\\")+1);

}

System.out.println("fname"+x+fname);

File f=new File(path+"\\"+x+fname);//将文件上传到 对应的位置

try {

daanfile.transferTo(f);

}catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

zuoye.setDaan("../upload/"+x+fname);

}else{

zuoye.setDaan(null);

}

if(!tu.isEmpty()){

String fname= tu.getOriginalFilename();//获取上传文件名称 并且以当前用户编号+项目编号做区分

if(fname.lastIndexOf("\\")!=-1){

fname=fname.substring(fname.lastIndexOf("\\")+1);

}

System.out.println("fname"+x+fname);

File f=new File(path+"\\"+x+fname);//将文件上传到 对应的位置

try {

tu.transferTo(f);

}catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

zuoye.setPic("../upload/"+x+fname);

}else{

zuoye.setPic(null);

}

ZuoyeQuery cq=new ZuoyeQuery();

cq.createCriteria().andZyidEqualTo(zuoye.getZyid());

int result= zd.updateByExampleSelective(zuoye, cq);

if(result>0){

request.setAttribute("msg","<script>alert('操作成功!')</script>");

}else{

request.setAttribute("msg","<script>alert('操作失败!')</script>");

}

return "redirect:showZuoye.do";

}

附录2：

@RequestMapping("files/showStuzuoyedetailsocre.do")//上分操作

public String shangfenshowStuzuoyedetail(Integer stuzyid,HttpServletRequest request){

Stuzuoye g= zd.selectByPrimaryKey(stuzyid);

request.setAttribute("g", g);

return "stuzuoyeUpdatescore123.jsp";

}

<update id="updateByPrimaryKey" parameterType="com.pojo.Stuzuoye" >

update stuzuoye

set stid = #{stid,jdbcType=INTEGER},

fujian = #{fujian,jdbcType=VARCHAR},

socre = #{socre,jdbcType=VARCHAR},

createtime = #{createtime,jdbcType=TIMESTAMP},

banji = #{banji,jdbcType=VARCHAR},

kemu = #{kemu,jdbcType=VARCHAR},

zyid = #{zyid,jdbcType=INTEGER},

pic = #{pic,jdbcType=VARCHAR}

where stuzyid = #{stuzyid,jdbcType=INTEGER}

</update>

附录3：

function preview(oper) {

if (oper < 10){

bdhtml=window.document.body.innerHTML;//获取当前页的html代码

sprnstr="<!--startprint"+oper+"-->";//设置打印开始区域

eprnstr="<!--endprint"+oper+"-->";//设置打印结束区域

prnhtml=bdhtml.substring(bdhtml.indexOf(sprnstr)+18); //从开始代码向后取html

prnhtml=prnhtml.substring(0,prnhtml.indexOf(eprnstr));//从结束代码向前取html

window.document.body.innerHTML=prnhtml;

window.print();

window.document.body.innerHTML=bdhtml;

}

else{

window.print();

}

附录4：

public class MailUtil {

/\*\*

\* SMTP服务器配置及邮件发送

\* @throws Exception

\*/

public void sendMail(MailConfig mc) throws Exception {

try{

// 连接邮件服务器的参数配置

Properties props = new Properties();

// 设置用户的认证方式

props.setProperty("mail.smtp.auth", "true");

// 设置传输协议(JavaMail规范要求)

props.setProperty("mail.transport.protocol","smtp");

// 设置发件人的SMTP服务器地址

props.setProperty("mail.smtp.host", "smtp.qq.com");

// 设置SMTP服务器端口 一般填写：25

props.setProperty("mail.smtp.port","25");

// 创建定义整个应用程序所需的环境信息的 Session 对象

Session session = Session.getInstance(props);

// 设置调试信息在控制台打印出来

session.setDebug(true);

// 创建邮件的实例对象

Message msg = getMimeMessage(session,mc);

// 根据session对象获取邮件传输对象Transport

Transport transport = session.getTransport();

// 设置发件人的账户名和密码

if(mc.getSenderAccount()==null || mc.getSenderAccount().isEmpty()){

throw new Exception("发件人账户为空");

}

if(mc.getSenderPassword()==null || mc.getSenderPassword().isEmpty()){

throw new Exception("发件人密码为空");

}

transport.connect(mc.getSenderAccount(), mc.getSenderPassword());

// 发送邮件，并发送到所有收件人地址，message.getAllRecipients() 获取到的是在创建邮件对象时添加的所有收件人, 抄送人, 密送人

transport.sendMessage(msg,msg.getAllRecipients());

// 关闭邮件连接

transport.close();

}catch(Exception e){

e.printStackTrace();

}

}

/\*\*

\* 获取邮件的实例对象

\* @param session

\* @return MimeMessage

\* @throws MessagingException

\* @throws AddressException

\*/

private MimeMessage getMimeMessage(Session session,MailConfig mc) throws Exception{

// 创建一封邮件的实例对象

MimeMessage msg = new MimeMessage(session);

// 设置发件人地址

if (mc.getSenderAddress()!=null && !mc.getSenderAddress().isEmpty()){

msg.setFrom(new InternetAddress(mc.getSenderAddress()));

}else{

throw new Exception("发件人地址为空");

}

/\*\*

\* 设置收件人地址（可以增加多个收件人、抄送、密送），即下面这一行代码书写多行

\* MimeMessage.RecipientType.TO:发送

\* MimeMessage.RecipientType.CC：抄送

\* MimeMessage.RecipientType.BCC：密送

\*/

if (mc.getRecipientAddresses()!=null && !mc.getRecipientAddresses().isEmpty()){

InternetAddress[] recipients = new InternetAddress().parse(mc.getRecipientAddresses());

msg.setRecipients(MimeMessage.RecipientType.TO, recipients);

}else{

throw new Exception("收件人地址为空");

}

if (mc.getCopyToAddresses()!=null && !mc.getCopyToAddresses().isEmpty()){

InternetAddress[] copyTos = new InternetAddress().parse(mc.getCopyToAddresses());

msg.setRecipients(MimeMessage.RecipientType.CC,copyTos);

}

// 设置邮件主题

if (mc.getSubject()!=null && !mc.getSubject().isEmpty()){

msg.setSubject(mc.getSubject(),"UTF-8");

}else{

throw new Exception("邮件主题为空");

}

//设置邮件正文

// 设置（文本+图片）和 附件 的关系（合成一个大的混合"节点" / Multipart ）

MimeMultipart mm = new MimeMultipart();

// 混合关系

mm.setSubType("mixed");

// 创建附件"节点"

MimeBodyPart body = new MimeBodyPart();

if (mc.getContent()!=null && !mc.getContent().isEmpty()){

body.setContent(mc.getContent(), "text/html;charset=UTF-8");

}else{

throw new Exception("邮件内容为空");

}

mm.addBodyPart(body); // 如果有多个附件，可以创建多个多次添加

if(mc.getAttachmentPaths()!=null && mc.getAttachmentPaths().length>0){

for (String attachmentPath: mc.getAttachmentPaths()) {

// 创建附件"节点"

MimeBodyPart attachment = new MimeBodyPart();

// 读取本地文件

DataHandler dh = new DataHandler(new FileDataSource(attachmentPath));

// 将附件数据添加到"节点"

attachment.setDataHandler(dh);

// 设置附件的文件名（需要编码）

attachment.setFileName(MimeUtility.encodeText(dh.getName()));

mm.addBodyPart(attachment); // 如果有多个附件，可以创建多个多次添加

}

}

// 设置整个邮件的关系（将最终的混合"节点"作为邮件的内容添加到邮件对象）

msg.setContent(mm);

// 设置邮件的发送时间,默认立即发送

msg.setSentDate(new Date());

msg.saveChanges();

return msg;

}

}

附录5：

<!-- 分页部分开始 -->

<table width="95%" border="0" align="center" cellpadding="0" cellspacing="0">

<tr>

<td height="6"><img src="images/spacer.gif" width="1" height="1" /></td>

</tr>

<tr>

<td height="33"><table width="100%" border="0" align="center" cellpadding="0" cellspacing="0" class="right-font08">

<tr>

<td width="50%">共 <span class="right-text09">${pager.totalPage }</span> 页 | 第 <span class="right-text09">${pager.currentPage }</span> 页</td>

<td width="49%" align="right">[<a href="javascript:void(0)" onclick="fenye(1)" class="right-font08">首页</a> | <a href="javascript:void(0)" onclick="fenye(${pager.prePage})" class="right-font08">上一页</a> | <a href="javascript:void(0)" onclick="fenye(${pager.nextPage})" class="right-font08">下一页</a> | <a href="javascript:void(0)" onclick="fenye(${pager.totalPage})" class="right-font08">末页</a>] 转至：</td>

<td width="1%"><table width="20" border="0" cellspacing="0" cellpadding="0">

<tr>

<td width="1%"><input id="currentPage" name="currentPage" value="${pager.currentPage }" type="text" class="right-textfield03" size="1" /></td>

<td width="87%"><input name="Submit23222" type="submit" class="right-button06" value=" " />

</td>

</tr>

</table></td>

</tr>

</table></td>

</tr>

</table>

<script>

function fenye(p){

var pp= document.getElementById("currentPage");

pp.value=p;

if(pp.value>0&&pp.value!=""&&pp.value<=${pager.totalPage }){

document.forms[0].submit();

}else{

alert("跳转页数不正确");

pp.focus();

}

}

</script>

# <!-- 分页部分结束 -->

# 外 文 资 料

**Model-Driven Development of Web Applications with UWA, MVC and JavaServer Faces**

**Abstract**. This paper presents a model-driven approach to the development of web applications based on the Ubiquitous Web Application (UWA) design framework, the Model-View-Controller (MVC) architectural pattern and the JavaServer Faces technology. The approach combines a complete and robust methodology for the user-centered conceptual design of web applications with the MVC metaphor, which improves separation of business logic and data presentation. The proposed approach, by carrying the advantages of ModelDriven Development (MDD) and user-centered design, produces Web applications which are of high quality from the user's point of view and easier to maintain and evolve. Keywords: Web Engineering, Model-Driven Development, Model Transformation, Model Driven Architecture, UWA, UML, MVC, JavaServer Faces.

1．Introduction

Web applications design methodologies available in the literature can be classified into those which focus on “what” the application is required to do (conceptual design in the problem domain) and those which focus on “how” the application can satisfy its requirements and implement the “what” (logical design in the domain of solutions). UWA [38], OOHDM [36], OOWS [32] and OO-H [9] pertain to the first category; Conallen’s proposal [13] belongs to the second one; UWE [19][22] and WebML [10][11][12] can be considered hybrid methodologies, as they cover both conceptual and logical design. Conceptual design methodologies abstract from implementation details and offer an overall view of the system from the view point of the users. Conceptual modeling is the right starting point to implement complex systems. However, the big distance between the conceptual model of a web application and its implementation makes the use of conceptual design methodologies insufficient for the development of a web application. If intermediate design levels for translating conceptual specification into implementation design are not provided, then the implementation activities of a web application may proceed independently from conceptual design, which will thus be under-exploited or even a waste of effort.

The trend of well-known design methodologies to evolve towards Model-Driven Development (MDD) shows the need for integrated approaches that support the whole web application lifecycle. Methodologies that have recently moved in this direction include OO-H [3], OOWS [3], UWE [18][20] and OOHDM [35], discussed more in detail in the related work section. This paper presents a model-driven approach to developing web applications based on the Ubiquitous Web Application (UWA) conceptual design methodology and the Model-View-Controller architectural design pattern. Compared to others, the UWA design framework, with its methodology and models, is particularly suited for designing web applications which are intended to be ubiquitous (accessible by different user types in different usage contexts and with different goals) and usercentered. By combining the characteristics of UWA with the advantages of the MVC architecture and the MDD paradigm, the resulting approach is particularly suited for developing ubiquitous web applications in a user-centered perspective as well as for supporting their maintenance and evolution.

UWA models are made at the right abstraction level to be used with application stakeholders and, at same time, can be detailed enough (design in the small) to provide useful information to the application developers. The methodology is also supported by a number of tools for drawing up UWA models and generating application design documentation. Nevertheless, since UWA models are independent from implementation details, they do not specify aspects such as the architecture, the software components, nor the database that the developer should implement to realize the designed application. By creating an intermediate design model to represent the application being implemented and a set of heuristics to map the UWA conceptual models onto it, we lay the basis for a model-driven approach to developing UWA-based web applications. The intermediate design model (referred to as the logical model in the following) is based on standard UML diagrams and adopts the Model-View-Controller (MVC) architectural design pattern [8].

The main contributions of the paper are: 1. The definition of a logical model that describes the application code to be developed and maps it to the application requirements; 2. The definition of a set of heuristics to translate UWA conceptual models to the new logical model; 3. The definition of an MDD approach based on UWA to support the entire web application lifecycle. The remainder of the paper is organized as follows. The next section discusses the background and motivations of the work. Section 3 presents the proposed UWA-based MDD approach and an example of its application. Related work is presented in Section 4 while Section 5 concludes the paper and provides an overview of future work

2 .Background and Motivations

2.1 A General Framework for the Development of Web Applications

Despite the differences between the various methodologies for engineering web applications found in the literature, a set of common concerns and development steps can be identified [37]. Fig. 1 graphically represents these common features. Basically any design methodology models web applications at three construction levels (content, navigation and presentation) and addresses a number of aspects spanning from structure to behavior. Aspects are orthogonal to all construction levels. Structure applies to content (how contents are structured) as well as to navigation (hypertext structures) and presentation (e.g., page organization). Similarly, behavior (e.g., business rules and user operations) can be associated with content, navigation and presentation.

As represented in Fig. 1, the development process starts with requirement specification and analysis and proceeds through a number of design steps to the implementation phase. Design steps include: 1. Conceptual design, focused on describing the problem domain and what the system is expected to do, independently of any technological detail. 2. Logical design, focused on the operation of the system while hiding implementation details specific to a particular platform. 3. Physical design, which adapts the logical model of the application to obtain detailed specifications for implementation with the chosen platform

2.2 Model-Driven Development

Though the use of design methodologies is not yet a common practice in the field of web engineering, it is known that a model-based approach provides a better alternative to the ad-hoc development of web applications and its inherent problems The need for systematic approaches to be adopted when developing complex systems and for design prior to implementation is now widely accepted. Model-Driven Development (MDD) [5] sees software development as a process whereby a high-level abstract model is successively translated into increasingly more detailed models, in such a way that eventually one of the models can be directly executed by some platform [3]. The MDD approach not only advocates the use of models (such as those resulting from the design steps described in the previous section) for the development of software, but also emphasizes the need for transformations in all phases of development, from system specification to implementation and testing [23]. Transformations from one model to the next create a chain which enables the automated implementation of a system starting from requirements.

The possibly most well-known initiative of MDD is the Model Driven Architecture (MDA) proposal [25] defined by the Object Management Group (OMG). The central idea of MDA is to separate platform-independent design from platform-specific implementation of applications, delaying the dependence on specific technologies for as long as possible. Therefore, MDA advocates the construction of platformindependent models (PIM) and the support of model transformations. The model that is directly executed by a platform (PSM) which satisfies all requirements, including non-functional ones, is also called “code”, and is usually the last model in the refinement chain. The development of web applications is a specific domain in which MDD can be successfully applied, due to the web-specific separation of concerns: content, navigation, presentation and customization.

2.3 UWA

The Ubiquitous Web Applications design framework (UWA) [38] provides a methodology and a set of models and tools for user-centered conceptual design of ubiquitous web applications. Mapping UWA models onto the general web application development framework depicted in Fig. 1, produces the diagram shown in Fig. 2. UWA covers the phases of requirement specification and analysis and conceptual design. Requirement elicitation is the activity devoted by UWA for the identification of the application’s stakeholders, their goals and, through a refinement process, the resulting requirements for the following design of the application. Requirements are classified into content, structure of content, access to content, presentation, and behavior (system and user operations).

Following the requirement elicitation phase, the conceptual user-centered design of the application is developed. The UWA Information Model, Navigation Model and Publishing Model cover the three levels of design represented on the vertical axis in Fig. 1, respectively. Each of these models comprises the structure aspect (e.g., the Information Model gives information on the application content, their structure, the structures for accessing content, etc.). The behavior aspect is addressed by two models: the Transaction Model and the Operation Model, the former addressing the business process design and the latter the design of more elementary user functionalities.

Finally, the UWA Customization Design activity, defines the customization rules that may apply to any of the UWA design models, thus providing the designed application with the ability to adapt to different usage contexts.

The UWA methodology enables separation of design concerns by devoting a design activity and a resulting model to each of the levels and aspects which characterize data- and operation-intensive web applications [34][14]. User-centered design ability (the perspective in developing the different UWA models is that of the different user types of the application) and context awareness for the resulting applications makes UWA one of the most valuable conceptual design methodologies for web applications. The availability of a MOF-Complaint metamodel for UWA [4] makes it possible to specify the semantics associated with each modeling concept, the valid configurations and the constraints that apply. At same time, being the UWA metamodel MOF-compliant it is possible to create modeling tools that generate models that can be easily exchanged, imported in different design tools, rendered into different formats, transformed, and used to generate application code

3 .A UWA-Based MDD Approach

3.1 Process Overview

The proposed UWA-based MDD approach to the development of web applications uses a series of models and successive model transformations to progressively refine and enrich the application requirements so as to obtain the application source code. The different models are also the basis for updated project documentation, at different levels of abstraction

Fig. 3 depicts the overall development process and the generated models. The process begins with UWA Requirements Elicitation, to identify the stakeholders of the application and their goals, from which the application requirements are derived. Stakeholders, goals, sub-goals and requirements are represented by means of stereotyped UML use-case diagrams.

The conceptual design is carried out using the UWA methodology and the conceptual model of the application produced. This model includes the Information Model, the Transaction Model, the Navigation Model, the Operation Model, the Publishing Model and Customization Model. The conceptual design phase is followed by logical design. The UWA conceptual models are transformed, by means of suitable translation rules, into a logical model that is closer to the specifics of implementation but still platform-independent. The adoption of standard UML diagrams and of the MVC architecture for the resulting application gives the name “UML-MVC” to our proposed logical model. The final phase of the MDD process consists of specializing the UML-MVC logical model to the specific platform chosen for implementation, obtaining a platform-specific model. Any implementation of the MVC architectural design pattern is suitable as the destination platform for the UML-MVC logical model defined in the previous step. In particular, we detail the case of the JavaServer Faces (JSF) technology [15][17] which is the technology we chose for experimenting our approach with an e-commerce web application.

As emphasized by Fig. 3, the proposed process, in addition to be model-driven, adheres to MDA. The UWA Requirement Elicitation model can, indeed, be considered a CIM as it focuses on functionalities required for the application. The UWA conceptual model and the UML-MVC logical model are PIMs, as they do not imply any specific technology to be used for the implementation. Finally the JSF model is a PSM as it is the specialization of the UML-MVC PIM for the JavaServer Faces technology. In the following of this section we describe the UML-MVC logical model, the transformation rules to obtain this model from the UWA conceptual models, and the guidelines that can be used to map the UML-MVC model onto the JavaServer Faces implementation framework

3.2 The UML-MVC Logical Model

The logical model we elaborated consists of stereotyped standard UML diagrams and is intended to model the software components to be developed for each of the three layers of the MVC architecture [8] as well as the relationships among them. Specifically, the model is structured into the following UML diagrams: 1. Model Class Diagram (MCD): an UML class diagram modeling the classes that participate in the Model layer and implement the application business logic and data persistency. 2. View Class Diagram (VCD): an UML class diagram representing the client and server pages in the View layer. These pages have the responsibility of presenting data and content to the user and enabling user interaction with the system. This diagram also models: a. Classes making part of the Controller layer; b. Associations between user interactions (e.g., the “Submit” of a form of a View page) and methods of classes of the Model that are in charge to serve them; c. Associations between attributes of classes of the View (which correspond to data provided/requested by the application) to methods of classes of the Model that manage them. d. Navigation links between pages of the View. 3. One or more UML Sequence Diagram describing the interactions between the various components of the system and their state transitions during the execution of complex user activities and web transactions. In addition, this diagram is used to describe the navigation steps between pages of the View that are associated with the different return values of the execution of the methods of Model classes.

The resulting overall model is independent of the specific technologies chosen for the implementation of the application, thus it is a PIM in the MDA architecture, but it is already sufficiently detailed to guide the application's development team. The model can be used to create the application with any implementation technology based on the MVC pattern. On an experimental basis, we used the Java ServerFaces framework.

3.3 Mapping UWA Conceptual Models onto UML-MVC Logical Model

The transformation of UWA conceptual models into UML-MVC logical models is accomplished by means of a set of mapping heuristics which create appropriate correspondences between the UWA models and modeling concepts, with the components of the MVC architecture and the elements of the logical model. Table 1 summarizes these heuristics: the first column lists the UWA conceptual models; the second lists the main modeling concepts included in each of the UWA models; the third column reports the layers of the MVC architecture onto which the considered UWA modeling concept is mapped; finally, the fourth column reports the specific elements of the logical model that originate from the UWA concept. Elements of the UML-MVC logical model we defined include: (i) Classes, Class Attributes and Class Methods of the MCD; (ii) Client and Server Page Classes of the VCD; (iii) Controller Classes mapping the user interactions with pages of the VCD onto methods of classes on the MCD; (iv) Associations between Classes of the MCD, (v) Associations between attributes of Classes of the VCD with methods of MCD classes; (vi) Navigation Links between Pages of the VCD; (vii) Sequence diagrams representing the workflow of a transaction and navigation rules to be implemented by the Controller.

Broadly speaking, the UWA Information and Transaction Models merge into the MCD, while the UWA Navigation and Publishing Models merge into the VCD. Associations between attributes and user interaction elements of the View pages with methods of Model classes originate from the Navigation Model and, indirectly, from the Transaction Model. In addition, from the UWA Transaction Model the UML Sequence Diagrams are also created.

3.4 Mapping UML-MVC Logical Model onto JavaServer Faces Platform Specific Model

The JSF technology is a Java implementation of the MVC architectural design pattern which simplifies the building of user interfaces for web applications by assembling reusable UI components in a page, connecting these components to an application data source; and wiring client-generated events to server-side event handlers [17]. As synthesized by bottom part of Fig. 3, the JSF architecture is a specialization of the MVC architecture in which the Model component is realized by means of Java business objects, the View component is made up of JavaServer Pages (JSP) in which custom tag libraries are used for expressing the JSF user interface components, and the Controller is implemented by a Servlet named FacesServlet.

To map the UML-MVC logical models onto the JSF software components the following guidelines can be used: 1. Classes of the MCD are mapped onto Java Business Objects, such as JavaBeans (JB) or Enterprise JavaBeans (EJB). 2. Classes of the VCD are implemented by means of Java Server Pages including the JSF user interface components specified by means of the JSF tags. 3. Associations between pages of the VCD with methods of classes in the MCD are used to define associations between presentation JSF components included in the Java Server Pages and attributes and methods of the Java Business Objects. 4. Associations between Controller classes and classes of the MCD in the VCD, together with information derived from the Sequence Diagrams on navigation rules associated with the different return values of the execution of a method of a Model class are used to define the “faces-config.xml” configuration file for the FacesServlet Controller.

3.5 Costs/Benefits of the Approach

The introduction of an additional design phase in the development process of a web application cause the process to lengthen and complicate, and more effort to be required if supporting tools towards MDD are not provided. On the other hand, no matter of if tools for the automatic transformation of conceptual models into logical models are available, providing developers with models which are closer to the implementation simplifies implementation choices, reduces coding time and helps in producing higher quality software. In fact, having a model which from one hand is directly linked to the conceptual model and from the other is very close to the implementation details helps the development of applications which exhibit: • greater internal consistency, as they fully satisfy the requirements of conceptual design; • greater usability, as they are more able to satisfy the expectations of the users; • greater maintainability, as design and requirements, since the impact on the code of any changes to the model (or indeed the impact on the model of any changes to the code) can be traced. The proposed approach, by defining the UML-MVC logical model, makes it possible to establish a correspondence between the UWA conceptual design of a web application and its implementation. Maintenance and evolution operations become easier, requirements traceability is made possible as well as alignment between software and documentation during the entire application lifecycle.

# 外 文 译 文

**基于UWA、MVC和JavaServer的Web应用模型驱动开发**

**摘要** 基于广泛存在的存在的web应用（UWA）设计框架、模型-视图-控制器（MVC）体系结构模式和JavaServer-Faces技术，提出了一种基于模型驱动的web应用开发方法。该方法将以用户为中心的web应用概念设计的完整和健壮的方法与MVC内核结合起来，改进了业务逻辑和数据表示的提交。该方法充分利用了模型驱动开发（MDD）和以用户为中心设计的优点，从用户的角度出发，生成了高质量、易于维护和发展的Web应用程序。

**关键词** Web工程，模型驱动开发，模型转换，模型驱动架构，UWA，UML，MVC，JavaServer Faces。

1．简介

文献中的Web应用程序设计方法可以分为关注应用程序需要做什么的方法（问题领域的概念设计）和关注应用程序如何满足其需求并实现什么的方法（问题领域的逻辑设计）解决方案）。UWA 、OOHDM、OOWS和OO-H属于第一类；Conallen的提案属于第二类；UWE[和WebML可以被视为混合方法论，因为它们涵盖了概念设计和逻辑设计。概念设计方法从具体细节中抽象出来，并从用户的角度提供系统的总体视图。概念建模是实现复杂系统的正确起点。然而，web应用程序的概念模型与其实现之间的巨大距离使得概念设计方法的使用不足以开发web应用程序。如果没有提供用于将概念规范转换为实现设计的中间设计模式，则web应用程序的实现活动可能独立于概念设计进行，因此概念设计将开发不足，甚至是做无用功。

广为流传的设计方法向模型驱动开发（MDD）发展的趋势表明，需要支持整个web应用程序生命周期的集成方法。最近朝这个方向发展的方法包括OO-H、OOWS、UWE和OOHDM，在相关的工作部分中进行了更详细的讨论。基于广泛存在的web应用（UWA）概念设计方法和模型-视图-控制器体系结构设计模式，提出了一种基于模型驱动的web应用开发方法。与其他设计框架相比，UWA设计框架及其方法和模型特别适合于设计广泛接触的（不同用户类型在不同的使用环境和目标下可访问）和以用户为中心的web应用程序。通过将UWA的特性与MVC体系结构和MDD范式的优点结合起来，所得到的方法特别适合于以用户为中心的视角开发无处不在的web应用程序，以及支持它们的维护和发展。

UWA模型是在正确的抽象级别上建立的，可以与应用程序角色一起使用，同时，可以足够详细（小部分设计），以便为应用程序开发人员提供有用的信息。该方法也得到了一些用于绘制UWA模型和生成应用程序设计文档的工具的支持。然而，由于UWA模型独立于实现细节，因此它们没有指定开发人员为实现所设计的应用程序而应该实现的架构、软件组件和数据库等方面。通过创建一个表示正在实现的应用程序的中间设计模型和一组将UWA概念模型映射到其中的启发式方法，我们为基于UWA的web应用程序开发的模型驱动方法奠定了基础。中间设计模型（以下简称逻辑模型）基于标准UML图，采用模型-视图-控制器（MVC）架构设计模式。

本文的主要贡献有：1。描述要开发的应用程序代码并将其映射到应用程序需求的逻辑模型的定义。定义一组启发式算法，将UWA概念模型转换为新的逻辑模型。基于UWA的MDD方法的定义，以支持整个web应用程序生命周期。论文的其余部分安排如下。下一节将讨论这项工作的背景和动机。第三节介绍了基于UWA的MDD方法及其应用实例。第4节介绍了相关工作，第5节总结了论文，并对未来的工作进行了概述。

2．背景和动机

2.1开发Web应用程序的通用框架

尽管在文献中发现了用于工程web应用程序的各种方法之间的差异，但是可以确定一组共同的关注点和开发步骤。图1以图形方式表示这些共同特征。基本上，任何设计方法都是在三个构造级别（内容、导航和表示）上对web应用程序建模，并处理从结构到行为的许多方面。各方面与所有施工水平都是正交的。结构适用于内容（内容的结构）以及导航（超文本结构）和表示（例如，页面组织）。类似地，行为（例如，业务规则和用户操作）可以与内容、导航和表示相关联。

如图1所示，开发过程从需求说明和分析开始，并通过许多设计步骤进行到实现阶段。设计步骤包括：1。概念设计，专注于描述问题领域和系统预期做什么，独立于任何技术细节。2。逻辑设计，侧重于系统的操作，同时隐藏特定于特定平台的实现细节。三。物理设计，它调整应用程序的逻辑模型，以获得使用所选平台实现的详细规范。

2.2模型驱动开发

虽然使用设计方法在web工程领域还不是一个普遍的实践方法，但是基于模型的方法提供了一个更好的替代web应用程序的特别开发，及其固有的问题当开发复杂的系统时需要采用系统的方法，并且在开发之前需要进行设计，该种实施已被广泛接受。模型驱动开发（Model Driven Development，MDD）将软件开发视为一个过程，在这个过程中，一个高级抽象模型被连续地转换成越来越详细的模型，最终某个平台可以直接执行其中一个模型。MDD方法不仅提倡使用模型（如前一节中描述的设计步骤所产生的模型）来开发软件，而且还强调在开发的所有阶段，从系统规范到实现和测试，都需要进行转换。从一个模型到下一个模型的转换创建了一个链，允许从需求开始自动实现系统。

MDD最著名的创举可能是由对象管理组（OMG）定义的模型驱动架构（Model-Driven Architecture，MDA）。MDA的核心思想是将独立于平台的设计与特定于平台的应用程序实现分离开来，尽可能地推迟对特定技术的依赖。因此，MDA提倡平台无关模型（platformindependent models，PIM）的构建和模型转换的支持。由平台（PSM）直接执行的模型满足所有需求，包括非功能性需求，也称为“代码”，通常是细化链中的最后一个模型。web应用程序的开发是一个特定的领域，由于特定于web的关注点（内容、导航、表示和定制）的分离，MDD可以在其中成功应用。

2.3 UWA

广泛存在的Web应用程序设计框架（UWA）为以用户为中心的概念设计提供了一种方法论和一套模型和工具。将UWA模型映射到图1所示的一般web应用开发框架上，产生图2所示的图。UWA包括需求说明、分析和概念设计阶段。需求获取是UWA专门用于识别应用程序的涉众、他们的目标，以及通过一个改进过程，为应用程序的以下设计产生的需求的活动。需求分为内容、内容结构、内容访问、表示和行为（系统和用户操作）。

在需求获取阶段之后，开发了以用户为中心的概念性应用程序设计。UWA信息模型、导航模型和发布模型分别涵盖图1中垂直轴上表示的三个设计层次。这些模型中的每一个都包括结构方面（例如，信息模型给出关于应用程序内容、它们的结构、用于访问内容的结构等的信息）。行为方面由两个模型来处理：事务模型和操作模型，前者处理业务流程设计，后者处理更基本的用户功能设计。

最后，UWA定制设计活动定义可应用于任何UWA设计模型的定制规则，从而为设计的应用程序提供适应不同使用上下文的能力。

UWA方法通过将设计活动和结果模型用于描述数据和操作密集型web应用程序的每个层次和方面，实现了设计关注点的分离。以用户为中心的设计能力（开发不同UWA模型的视角是应用程序的不同用户类型）和对结果应用程序的上下文感知使得UWA成为web应用程序最有价值的概念设计方法之一。UWA的MOF投诉元模型的可用性使得可以指定与每个建模概念相关联的语义、有效配置和应用的约束。同时，由于符合UWA元模型MOF，可以创建建模工具来生成模型，这些模型可以很容易地交换、导入到不同的设计工具中、呈现为不同的格式、转换并用于生成应用程序代码。

3.基于UWA的MDD方法

3.1 过程概述

提出的基于UWA的MDD方法通过一系列的模型和连续的模型转换来逐步细化和丰富应用需求，从而获得应用源代码。不同的模型也是在不同抽象层次上更新项目文档的基础

图3描绘了总体开发过程和生成的模型。该过程从UWA需求获取开始，以确定应用程序的涉众及其目标，从中派生应用程序需求。目标、子目标和需求都是通过原型化的UML用例图来表示的。

概念设计是使用UWA方法和产生的应用程序的概念模型进行的。该模型包括信息模型、事务模型、导航模型、操作模型、发布模型和定制模型。概念设计阶段之后是逻辑设计阶段。UWA概念模型通过合适的转换规则被转换成更接近实现细节但仍然独立于平台的逻辑模型。采用标准的UML图和MVC架构作为最终的应用程序为我们提出的逻辑模型命名为“UML-MVC”。MDD过程的最后阶段包括将UML-MVC逻辑模型专门化到选择用于实现的特定平台，获得特定于平台的模型。MVC架构设计模式的任何实现都适合作为上一步定义的UML-MVC逻辑模型的目标平台。特别是，我们详细介绍了JavaServerFaces（JSF）技术的情况[15][17]，这是我们选择的用于在电子商务web应用程序中试验我们的方法的技术。

如图3所强调的，除了模型驱动之外，所提出的过程还遵循MDA。UWA需求获取模型实际上可以被视为CIM，因为它关注应用程序所需的功能。UWA概念模型和UML-MVC逻辑模型是PIMs，因为它们并不意味着要用于实现的任何特定技术。最后，JSF模型是一个PSM，因为它是用于JavaServer Faces技术的UML-MVC PIM的专门化。在本节的下面，我们将描述UML-MVC逻辑模型，从UWA概念模型获得该模型的转换规则，以及将UML-MVC模型映射到JavaServerFaces实现框架的指导原则。

3.2 UML-MVC逻辑模型

我们所阐述的逻辑模型由标准的原型化UML图组成，旨在为MVC体系结构的三个层中的每一层开发的软件组件以及它们之间的关系建模。具体来说，该模型被构造成以下UML图：1。模型类图（MCD）：一个UML类图，它对参与模型层的类进行建模，并实现应用程序业务逻辑和数据持久性。2。视图类图（VCD）：表示视图层中的客户机和服务器页面的UML类图。这些页面负责向用户呈现数据和内容，并允许用户与系统交互。该图还模拟了：a.构成控制器层一部分的类；b.用户交互（例如，视图页面的“提交”）与负责为其服务的模型类的方法之间的关联；c、 视图类的属性（对应于应用程序提供/请求的数据）与管理它们的模型类的方法之间的关联。d、 视图页面之间的导航链接。三。一个或多个UML序列图，描述在执行复杂的用户活动和web事务期间系统各组件之间的交互及其状态转换。此外，此图还用于描述与执行模型类方法的不同返回值相关联的视图页面之间的导航步骤。

最终的总体模型独立于为实现应用程序而选择的特定技术，因此它是MDA体系结构中的PIM，但是它已经足够详细，可以指导应用程序的开发团队。该模型可用于使用基于MVC模式的任何实现技术创建应用程序。在实验的基础上，我们使用了Java ServerFaces框架。

3.3 将UWA概念模型映射到UML-MVC逻辑模型

UWA概念模型到UML-MVC逻辑模型的转换是通过一组映射启发来完成的，这些映射启发在UWA模型和建模概念之间创建适当的对应关系，其中包含MVC体系结构的组件和逻辑模型的元素。表1总结了这些启发式方法：第一列列出了UWA概念模型；第二列列出了每个UWA模型中包含的主要建模概念；第三列报告了MVC体系结构的各个层，将考虑的UWA建模概念映射到这些层上；最后，第四列报告了起源于UWA概念的逻辑模型。我们定义的UML-MVC逻辑模型的元素包括：（i）MCD的类、类属性和类方法；（ii）VCD的客户端和服务器页类；（iii）控制器类将用户与VCD页的交互映射到MCD的类方法；（i v）MCD的类之间的关联；（v）MCD的类之间的关联使用MCD类方法的VCD类的属性；（vi）VCD页面之间的导航链接；（vii）表示事务工作流的序列图和由控制器实现的导航规则。

广义上讲，UWA信息和事务模型合并到MCD中，而UWA导航和发布模型合并到VCD中。视图页面的属性和用户交互元素与模型类方法之间的关联源于导航模型，间接地源于事务模型。此外，从UWA事务模型中还创建了UML序列图。

3.4 将UML-MVC逻辑模型映射到JavaServerFaces平台特定模型

JSF技术是MVC架构设计模式的Java实现，通过在页面中组装可重用的UI组件，将这些组件连接到应用程序数据源，并将客户端生成的事件连接到服务器端事件处理程序，简化了web应用程序用户界面的构建。如图3底部所示，JSF架构是MVC架构的一个特殊化，其中模型组件是通过Java业务对象实现的，视图组件是由JavaServer Pages（JSP）组成的，其中自定义标记库用于表示JSF用户界面组件，控制器是由名为FacesServlet的Servlet实现。

要将UML-MVC逻辑模型映射到JSF软件组件上，可以使用以下准则：1。MCD的类被映射到Java业务对象上，如JavaBeans（JB）或EnterpriseJavaBeans（EJB）。2。VCD的类是通过Java服务器页面实现的，包括通过JSF标记指定的JSF用户界面组件。三。VCD页面与MCD中类方法之间的关联用于定义Java服务器页面中包含的表示JSF组件与Java业务对象的属性和方法之间的关联。四。控制器类和VCD中MCD类之间的关联，以及从导航规则序列图中获得的与执行模型类方法的不同返回值相关的信息，用于定义“面-配置.xml“FacesServlet控制器的配置文件。

3.5 方法的成本/收益

在web应用程序的开发过程中引入一个额外的设计阶段会导致开发过程的延长和复杂化，如果没有提供MDD的支持工具，则需要付出更多的努力。另一方面，无论是否有将概念模型自动转换为逻辑模型的工具，为开发人员提供更接近实现的模型都可以简化实现选择，减少编码时间，并有助于生成更高质量的软件。事实上，拥有一个一方面与概念模型直接相关，另一方面又与实现细节非常接近的模型，有助于开发以下应用程序：•当它们完全满足概念设计的要求时，具有更大的内部一致性；•更大的可用性，因为它们更能满足用户的期望；•更大的可维护性，作为设计和需求，因为对模型的任何更改对代码的影响（或者对代码的任何更改对模型的影响）都可以跟踪。该方法通过定义UML-MVC逻辑模型，使web应用程序的UWA概念设计与实现之间建立对应关系成为可能。维护和演进操作变得更加容易，需求跟踪成为可能，并且在整个应用程序生命周期中软件和文档之间保持一致。

**北 京 邮 电 大 学**

**本科毕业设计（论文）开题报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 计算机学院 | 专业 | | 网络工程 | | 班级 | 11班 |
| 学生姓名 | 赵瑞 | 学号 | | 2016211425 | | 班内序号 | 16 |
| 指导教师姓名 | 吴起凡 | 所在单位 | | 计算机学院 | | 职称 | 高工 |
| 设计（论文）题目 | （中文）学生作业批改系统 | | | | | | |
| （英文）Student Homework Correcting System | | | | | | |
| 毕业设计（论文）开题报告内容：  1.课题背景及意义：  时至今日，随着高校课业的多样化和计算机网络的高速发展，传统作业布置和纸质版提交愈发显得笨重繁琐，依靠课代表收集作业的模式不能确保每次作业收集的稳定性，大大降低了学生作业批改的效率，尤其是对于习惯了网络便利的当代大学生而言，日常生活中的作业保管和忘记作业时间等问题屡见不鲜，而网络的高效便利恰恰与之互补。  针对这种普遍存在的情况，构建一个“学生作业批改系统”线上平台，为学生提供便利的作业提交渠道和长久的作业保存，能够减轻学生不必要的过失（忘记提交时限、丢失作业等）；对于教师，能够省去传统通过线下布置作业并通过课代表收集统计的时间，在“学生作业批改系统”中实现一键批量下载和批量上传也能够大大节约教师时间。科技的发展应该更好的服务人们的生活工作，该系统应用于学校能更好地统计学生的平时成绩，在期末这种工作繁多的时候能够提供很大帮助，该系统的开发具有深远的现实意义，可谓百益而无一害。  2.基本内容和主要问题：  2.1基本内容  采用Java Web技术，配套相应工具实现一个应用于学校的“学生作业批改系统”，为在校师生提供一个功能强大且便利简洁的平台，项目开发基于软件工程流程。基本内容包含管理员、教师和学生三个角色，并分模块实现各角色对应基础功能：包括管理员对成员、课程、年级及班级的管理，教师布置作业、下载批改作业及打分统计，学生下载提交作业和分数查看，以及通用的个人资料、修改密码和退出登录，并在此基础上进行功能拓展和性能优化。设计功能时需要充分考虑实际使用情况，结合现实问题全面分析，构建数据流图，力求通过分析得到各模块的细致划分；勇于在项目实现中尝试新技术，充分发挥创造力和想象力，做出让人眼前一亮的设计；具体实现中应当对比不同设计和不同代码实现的差别优劣，选取更加稳定高效的方案实现。  2.2主要问题  1.确保各项基础功能完整且稳定，能够应对各种情况的测试  2.确保管理员的操作权限完整，不同操作间具有良好的逻辑关系  3.针对学生作业提交的不同格式进行相应处理，pdf、文档等能直接打开  4.解决文件同时上传下载冲突，解决数据库等的并发控制  5.建立目录结构为学生教师分角色存放每次作业，方便学生找回复习和教师期末统计  6.实现教师批量下载作业、作业批量上传及自动分类存放  7.个人资料界面绑定个人邮箱，若忘记密码可通过邮箱重置修改密码  8.先分单元测试再总体测试，做到全面细致，并思考改进空间  3.研究方法及措施  在B/S架构下，主要采用JavaWeb技术，配套Tomacat、MySQL数据库和Navicat数据库管理工具等，于Eclipse上进行开发一个“学生作业批改系统”。需要深刻学习理解Java、jsp、servlet、MySQL、前端框架优化等相关知识。基于软件工程八大步骤（问题定义、可行性研究、需求分析、总体设计、详细设计、编码及单元测试、综合测试、软件维护）逐步构建，并做好相应记录。  B/S结构是一种网络结构模式。这种模式统一客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。  Java Web，是用Java技术来解决相关web互联网领域的技术总和 ，servlet，jsp和第三方框架等属于Java在服务器端的应用，能够简化工程量，优化系统结构。  MySQL是一个[关系型数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/696511)，具有体积小、速度快、总体拥有成本低的特点，十分适合中小项目开发使用。  Eclipse 是一个开放[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81/3969)的、基于[Java](https://baike.baidu.com/item/Java/85979)的可扩展开发平台，功能强大，可拓展性良好，集成许多环境配置，是Java项目开发的不二之选。  软件工程是一门研究用工程化方法构建和维护有效的、实用的和高质量的软件的学科。尤其注重工程开发中的步骤流程、功能间的逻辑关系和模块间的数据流通。  4.进度安排  第1~3周：需求分析，明确项目重点难点，完成《毕业设计任务书》及《开题报告》，搭建项目环境  第4~5周：完成网页基础模块构建，连接数据库，优先实现管理员登录  第5~8周：逐步实现各种基础功能，确保模块间连接稳定，记录相关问题  第9~11周：改进完善基础功能，对所有功能进行技术测试并记录  第12周~：提交软件及相关文档，整理完成本科毕业论文，准备答辩  5.主要参考文献  [1]张海藩、牟永敏.软件工程导论.清华大学出版社.2013，（6）  [2] David Flanagan，美. JavaScript权威指南. 机械工业出版社2011,(6)  [3]郭欣.构建高性能Web站点. 电子工业出版社.2009  [4] Brian Goetz等.Java并发编程实战. 机械工业出版社.2012  [5]李兴华、王月清.Java Web开发实战经典. 清华大学出版社.2010  [6]JonathanChaffer、KarlSwedberg. jQuery基础教程.人民邮电出版社.2008  [7] Trevor Burnham. JavaScript异步编程. 人民邮电出版社.2013  [8] Craig Walls. Spring实战. Manning.2011 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 年 月 日 | | |

**北 京 邮 电 大 学**

**本科毕业设计（论文）中期进展情况检查表**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | | | 计算机学院 | 专业 | | 网络工程 | | 班级 | 11班 |
| 学生姓名 | | | 赵瑞 | 学号 | | 2016211425 | | 班内序号 | 16 |
| 指导教师姓名 | | | 吴起凡 | 所在单位 | | 计算机学院 | | 职称 | 高工 |
| 设计（论文）题目 | | | （中文）学生作业批改系统 | | | | | | |
| （英文）Student Homework Correcting System | | | | | | |
| 目前已完成任务 | 主要内容:  截至期中，进度与计划有所出入，但相差时间在一周以内，主要是一些技术用法实践的学习和测试稍费时间。开题至今依次完成的任务有：开题报告、课题调研分析、需求分析、环境搭建配置、登录拦截、数据库连接操作封装和验证码实现。至今网页基本框架马上完成，开始逐步实现各模块的核心功能。  针对课题分析，主要是调研了课题的背景和现实意义。传统的课代表收集作业注定会被线上提交所取代，构建一个“学生作业批改系统”线上平台，不仅省去了课间收集作业的时间，也避免了遗漏丢失的风险，额外的统计模块在期末也能为教师提供许多便利。  需求分析是一个项目的先决条件，能够为后续功能开发提供必要的保障和理论上的参考，但是一个项目是可变的，需求分析只能详尽考虑模块的核心功能和函数接口，对于后续拓展功能或者优化完善都需要及时修改，比如对于管理员角色功能的考虑，之前我初步划分为年级班级、教师、学生三个管理模块，过段时间仔细考虑后发现年级这个元素是多余的，我完全可以将年级id和班级id结合也是一枚主键。基于软件工程的相关知识，我采用面向对象分析方法，通过绘制uml图对项目进行领域建模和用例建模，后续完成数据库表的构建后再完成数据流图的绘制。领域模型是对领域内的概念类或现实世界中对象的可视化表示，并建立业务领域概念之间的关系。我分析绘制了各角色的活动图和类图，前者能够清晰显示角色生命周期内的行为流程，后者着重于从项目的所有对象中提取出模型元素，侧重各元素自身的属性和元素之前的关系，包括依赖、关联、组合、聚合关系依次加深。用例模型着重对系统、子系统或类的功能行为进行建模，我绘制了各角色的用例图、系统顺序图和操作契约模板，其中系统顺序图精确到接口函数和数据库关联，完成系统模块流程模拟。完成需求分析后进行整理排版，汇成文档保存。  之后进行了环境搭建和项目实施。安装配置了Tomcat，通过XAMPP软件包安装成功了MySQL等工具，并将项目通过eclipse上传至Github指定文件夹。项目实施时针对Javaweb的一些遇到的问题，系统的学习了servlet和其过滤拦截功能、jQuery及其部分插件的使用，Ajax的写法规则，UI中的JS控件，JDBC连接项目数据库和html界面某些写法构成技巧等。基本实现了各角色的登录逻辑和验证码功能，其中验证码功能通过在特定字符集中随机选择字符构建，背景图片通过Java绘图随机划分区域和上色实现。完成了部分角色主界面的jsp文件，包括界面的跳转逻辑和界面美化。这部分也是现在工作内容，完成后开始分模块实现核心功能，优先从管理员控制修改师生课程信息开始。 | | | | | | | | |
| 是否符合任务书要求进度 符合 | | | | | | | | |
| 尚需完成的任务 | 后续任务按模块分为管理员、教师、学生的功能实现，按架构分为前端角色界面控制逻辑、后端代码控制函数、数据库搭建完备。在项目大体完成后进行模块功能接口逻辑的测试，完善修改优化。  管理员功能实现主要是与数据库访问增删改查有关，需要数据库表关系明确构建完整，所以后续E-R图的绘制也很重要，针对某些复杂功能进行结构化建模构建数据流图对逻辑控制有很大帮助。  学生和教师的功能实现设计有所交叉，主要是文件相关操作，包括文件上传下载和在服务器端的存储问题，其中文件同时访问、下载、修改时的冲突需要考虑，文件存储在服务器的文件目录也需要仔细构建，这涉及到文件类型的统一保存和在线打开，设计服务器保管所有文件，为每份作业文件设置标识包括班级、课程、布置日期、截至日期、提交日期、评分、文件类型，为学生提供自己作业保存查看，为教师提供班级、课程作业查看和统计。并考虑针对能够在线打开的文件，完成一个对比的功能，使得教师学生能够比对答案和一份作业。教师还需要额外的统计模块用于学生作业整体情况的查看，考虑打印成表单显示。  管理员、学生、教师都有个人信息模块，包含修改个人信息、注销和修改密码。其中密码修改考虑利用网页发送邮件实现。  额外功能的考虑：补交、作业备注、邮件提醒作业日期 | | | | | | | | |
| 能否按期完成设计（论文） 能够按期完成 | | | | | | | | |
| 存在问题和解决办法 | 存  在  问  题 | 1.服务器文件存储结构逻辑的设计  2.文件的批量下载和上传，  3.针对不同类型文件上传的不同处理方式  4.作业和答案的在线查看  5.对同一作业同时修改的并发控制，考虑上传下载的冲突  6.实现网页发送邮件找回密码和提醒时限 | | | | | | | |
| 拟  采  取  的  办  法 | 1.为每个作业文件进行标识，用一串字符的组合表示作业的时间班级课程学号等，访问时根据特定位上的字符进行整体查询访问，比如教师要访问某个学生的某一科成绩，只需搜索匹配两部分的字符，然后显示查询结果。  2.采用apache的开源工具common-fileupload这个文件上传组件，能够实现文件的批量上传下载。  3.设置一个缓冲区，在收到上传文件名时，判断后缀名是否合法。不合法予以提醒。  4. 使用OpenOffice将office文件转成PDF文件，使用SwfTools将PDF等类型文件转成swf文件。文件上传的同时就做好文件处理。所有的文件交由服务器处理，避免客户端安装插件，数据库设计上提供保存两个文件的存放地址的字段，一个是供给下载原件使用，另一个供给在线预览使用。  5.设定操作的优先级，当高优先级访问修改时只允许其他用户访问查看但不可修改，上传下载动态设置最大数量，确保在上传下载负荷内。  6.先上传文件附件，再使用MailMessage对象的Attachments属性发送带有附件的电子邮件 | | | | | | | |
| 指导教师签字 | |  | | | 日期 | | 年 月 日 | | |
| 检查小组意见 | | 负责人签字： 年 月 日 | | | | | | | |

**北 京 邮 电 大 学**

**教师指导本科毕业设计（论文）记录表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 宋体，五号，居中 | 专业 | | 宋体，五号，居中 | | 班级 | 宋体，五号，居中 |
| 学生姓名 | 宋体，五号，居中 | 学号 | | 宋体，五号，居中 | | 班内序号 | 宋体，五号，居中 |
| 指导教师姓名 | 宋体，五号，居中 | 职称 | | 宋体，五号，居中 | | | |
| 第1－2周记录： (具体要求至少3-4句详细工作，不能只写一句话。此表应有15周记录)  宋体，五号，两端对齐，首行缩进2字符，1.2倍行距 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 年 月 日 | | |
| 第3－4周记录：  宋体，五号，两端对齐，首行缩进2字符，1.2倍行距 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 年 月 日 | | |

注：每2周指导内容记录在一个表格中，双面打印。

**北 京 邮 电 大 学**

**教师指导本科毕业设计（论文）记录表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 宋体，五号，居中 | 专业 | | 宋体，五号，居中 | | 班级 | 宋体，五号，居中 |
| 学生姓名 | 宋体，五号，居中 | 学号 | | 宋体，五号，居中 | | 班内序号 | 宋体，五号，居中 |
| 指导教师姓名 | 宋体，五号，居中 | 职称 | | 宋体，五号，居中 | | | |
| 第5－6周记录： (具体要求至少3-4句详细工作，不能只写一句话。此表应有15周记录)  宋体，五号，两端对齐，首行缩进2字符，1.2倍行距 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 年 月 日 | | |
| 第7－8周记录：  宋体，五号，两端对齐，首行缩进2字符，1.2倍行距 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 年 月 日 | | |

注：每2周指导内容记录在一个表格中，双面打印。

**北 京 邮 电 大 学**

**教师指导本科毕业设计（论文）记录表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 宋体，五号，居中 | 专业 | | 宋体，五号，居中 | | 班级 | 宋体，五号，居中 |
| 学生姓名 | 宋体，五号，居中 | 学号 | | 宋体，五号，居中 | | 班内序号 | 宋体，五号，居中 |
| 指导教师姓名 | 宋体，五号，居中 | 职称 | | 宋体，五号，居中 | | | |
| 第9－10周记录： (具体要求至少3-4句详细工作，不能只写一句话。)  宋体，五号，两端对齐，首行缩进2字符，1.2倍行距 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 年 月 日 | | |
| 第11－12周记录：  宋体，五号，两端对齐，首行缩进2字符，1.2倍行距 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 年 月 日 | | |

注：每2周指导内容记录在一个表格中，双面打印。

**北 京 邮 电 大 学**

**教师指导本科毕业设计（论文）记录表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 宋体，五号，居中 | 专业 | | 宋体，五号，居中 | | 班级 | 宋体，五号，居中 |
| 学生姓名 | 宋体，五号，居中 | 学号 | | 宋体，五号，居中 | | 班内序号 | 宋体，五号，居中 |
| 指导教师姓名 | 宋体，五号，居中 | 职称 | | 宋体，五号，居中 | | | |
| 第13－14周记录： (具体要求至少3-4句详细工作，不能只写一句话。)  宋体，五号，两端对齐，首行缩进2字符，1.2倍行距 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 年 月 日 | | |
| 第15周记录：  宋体，五号，两端对齐，首行缩进2字符，1.2倍行距 | | | | | | | |
| 指导教师签字 |  | | 日期 | | 年 月 日 | | |

注：每2周指导内容记录在一个表格中，双面打印。