** **

**Southwest university of science and technology**

本科毕业设计（论文）

程序设计能力认证系统设计与实现

|  |  |
| --- | --- |
| 学院名称 | 计算机科学与技术学院 |
| 专业名称 | 计算机科学与技术 |
| 学生姓名 | 肖杰 |
| 学号 | 20121199 |
| 指导教师 | 杨春明 讲师 |

二〇一六年六月

程序设计能力认证系统设计与实现

摘要：为加深教师了解学生的程序设计基本能力，开展程序设计能力的认证，可以有针对性的对学生开展后期的课程教学。本文通过对国内外相关平台的调研和分析，确定了系统的可行性。针对于代码评判问题，采用linux沙箱调用内核API ptrace技术来实现的，针对代码的相似度检测，采用余弦向量算法，检测学生提交代码的相似度。经测试，系统能够实现对题库的添加和试卷抽题，考试学生名单的导入和账号密码的生成，保证试卷难度统一，能够实现对提交代码的自动评判和教师的在线评卷、考试完成后试卷试题、成绩和代码的导出以便存档，实现代码的相似度检测和学生程序设计能力的在线认证功能。模块化和组件化的开发，提升了工作效率，简化了工作流程，便于后期的扩展和开发维护，整个系统对教师开展课程教学和信息系统的开发有着积极作用。

关键词：能力认证；在线评判；模块化；组件化

Programming capability Certified System Design and Implementation

**Abstract**：To help teachers understand the students' basic skills program designed to carry out the certification process design capabilities, can be targeted for students to carry out the latter part of the teaching curriculum. Based on the platform of domestic and foreign research and analysis to determine the feasibility of the system. Code for the judge to question the use of linux sandbox calls the kernel API ptrace technology to achieve, for the similarity detection code, cosine vector algorithm to detect similarities students submit code. After testing, the system can be added to the exam papers and questions pumping, generate a list of import and test student account password to ensure the unity of difficulty papers, can realize automatic submission and evaluation of teachers marking codes online, after the completion of the examination papers export questions, the results and the code for documentation, implement code similarity detection and student programming capability online authentication. Modular and component-based development, enhance the work efficiency, simplify workflows, ease of maintenance and development of post-expansion, the entire system of teachers in curriculum development and teaching of information systems has a positive effect.

**Key words:** Competencies, Online judge, Modular, Componentization

目 录

[第1章 绪 论 5](#_Toc451169862)

[1.1 课题的背景 5](#_Toc451169863)

[1.2 课题的研究意义 6](#_Toc451169864)

[1.3 课题研究现状 6](#_Toc451169865)

[1.4 课题目标 7](#_Toc451169866)

[1.5 课题实施方案 8](#_Toc451169867)

[1.6 本章小结 8](#_Toc451169868)

[第2章 系统开发关键技术介绍 9](#_Toc451169869)

[2.1 SSM框架 9](#_Toc451169870)

[2.2 前端构建工具 10](#_Toc451169871)

[2.3 系统前台页面开发中使用的框架 11](#_Toc451169872)

[2.4 本章小结 11](#_Toc451169873)

[第3章 系统需求分析 12](#_Toc451169874)

[3.1系统可行性分析 12](#_Toc451169875)

[3.2 系统非功能性需求分析 12](#_Toc451169876)

[3.3 系统功能需求分析 13](#_Toc451169877)

[3.4 系统用例分析 14](#_Toc451169878)

[3.4.1 管理员及老师模块 14](#_Toc451169879)

[3.4.2 学生模块 15](#_Toc451169880)

[3.5 本章小结 20](#_Toc451169881)

[第4章 系统概要设计 21](#_Toc451169882)

[4.1 系统运行环境 21](#_Toc451169883)

[4.2 系统设计策略 21](#_Toc451169884)

[4.3 功能结构设计 21](#_Toc451169885)

[4.3.1 老师或管理员模块 22](#_Toc451169886)

[4.3.2 学生模块 22](#_Toc451169887)

[4.4 数据库设计 23](#_Toc451169888)

[4.4.1 数据库环境 23](#_Toc451169889)

[4.4.2 数据库命名规则 23](#_Toc451169890)

[4.4.3 数据库概念设计 23](#_Toc451169891)

[4.4.4 数据库表设计 24](#_Toc451169892)

[4.4 本章小结 28](#_Toc451169893)

# 第1章 绪 论

## 1.1 课题的背景

随着计算机的普及，互联网的发展，各高校都开设了计算机相关课程，各专职技术学校等也有相关技能的培训。程序语言设计作为高等院校计算机专业学生必修课程，对于学生程序设计能力的培养具有重要的意义。程序设计类课程是高等院校绝大多数专业都会开设的一门计算机基础课程。该课程对于培养学生程序设计的基本技能、严谨的逻辑思维方式以及用计算机解决实际问题的能力等方面的素质都具有重要作用。程序设计类课程是一门注重实践和动手能力培养的学科。实际的编程能力是学科培养和教学效果考查的重点。目前，大多数程序设计类课程的课后作业、上机实验报告以及考试试卷都是由人工进行判题评阅的。这种大量的重复性劳动一方面对任课教师是种沉重负担；另一方面，对学生来说时效性差，无法对学生进行及时反馈，导致学习效果受到影响，且由于课后作业容易出现抄袭现象。

随着行业界内对程序设计人员的需求不断增多，但是并没有较好的程序设计能力评价的方法；高校内主要通过笔试或者上级考试来考核学生能力，但只针对具体的某门课程，不能准确评价学生的程序设计能力。

尽管行业内或者有的高校有online judge平台，但是并没有具体到某个具体的知识点上来考察。如果能够集中学生在某个时间进行相关课程知识点的考试，考试分为不同等级不同难度，通过的学生获得相应证书，学生可以凭证书免修相关课程或者作为求职凭证，如此可以减轻学校教师的教学负担，学生可以实时把握自己的程序设计能力，相关企业也可以以此判断学生的设计能力。

程序设计能力系统等信息系统的设计和实现中，用户对站点的期望越来越高，直接导致网站前端规模不断扩大，越来越多的前端脚本代码增加到应用系统中，前端开发出现了代码规模大、组织维护困难、代码重用性低、扩展性差等问题。功能扩展、多种浏览器的适配成了系统设计中的重点。

在此背景下，需要有一个能够认证学生程序设计能力的系统，来考核学生的程序设计综合能力，也需要一个针对于信息系统的前端开发方案，方便其他信息系统的开发和维护，甚至是后期的扩展。

## 1.2 课题的研究意义

本系统结合高校教学的特点，设计了一种自动判题系统，使学生在日常作业、上机实验和自测练习时，能够及时准确地发现错误，设计出正确、合理、高效的程序。以信息化技术更新教学方法和教学手段，辅助日常教学，达到更好的教学效果。更客观的评价学生的知识的系统性和程序设计能力。与当前存在的系统相比，本系统在功能和设计上有很多优点。

1. 试卷的随机生成，题目不同，难度统一

通过程序能力认证系统，老师和管理员可以登陆系统添加题库，组织考试并自动生成该次考试的试题，每份考卷题目不同，但是难度控制合适。这样可以保证题目的多样性和避免学生考试时的互相抄袭，使得考试认证更客观准确。

1. 代码和算法的自由空间大，可任学生选择

系统主要以特定的输入输出数据来进行测试，如ACM Online Judge系统。题目要求输入指定的数据，检测用户程序是否能够得到相应的输出数据。只对输入和输出数据格式做少量限定，可以让用户自由设计自己的算法和代码。

1. 前端工程分而治之，解决基本开发效率运行效率问题

JS/CSS模块化开发，对UI组件组件化开发，规范了开发流程，代码更清晰易懂，且代码块之间的依赖性变低，提升系统开发效率和运行效率，便于某个模块的下线或者升级。在系统后期的维护和扩展也是有帮助的。

因此，一个具有认证功能的在线程序设计评判系统是很有必要的，它可以减轻老师的教学负担，改善老师对学生作业手工批改的效率问题，帮助学生认识自己的程序设计综合能力。且组件化和模块化的开发有利于提升开发的工作效率和代码的可维护性。本系统的设计与实现，对教学的改善和信息系统的开发具有很好的现实意义

## 1.3 课题研究现状

计算机程序设计能力考试认证，旨在通过统一组织的在线考试即自动评测方法客观的评判考生的算法设计与程序设计实现能力，科学的评价计算机程序设计人才，为企业选拔人才提供参考标准。

国内PAT（Programming Ability Test, 简称PAT），每年组织多次考试，每场考试分顶级（Top Level），甲级（Advanced Level）和乙级（Basic Level）。不同等级有不同的题目难度和评判标准，各等级考试通过后的考生可以获得对应证书。还有中国计算机学会组织的CCF软件能力认证Certified Software Professional（CSP）等等。在诸多的在线程序设计考试平台中，leetcode是一个为面试技术编码做准备的练习平台，针对于面试中常遇到的算法，数据库，shell等知识点在线评判，题目难度分级，支持C, C++, Java, Python, C#, JavaScript, Ruby, Bash语言。Codeforces是一个程序设计竞赛平台，致力于规划组织编程比赛，定期举行比赛，反应参与者的技能等级。甚至可以让参与者自己组织比赛。

在开发框架方面，国内外针对于众多的Web应用，出现了很多基于MVC、MVVM和MVP模式开发框架，后端SpringMVC，Struts2等等都比较成熟，前端有BackboneJs, AngularJs, EmberJs, React等等。但是没有一套针对于信息系统开发的前端解决方案

## 1.4 课题目标

本课题需要了解和研究当前国内外存在的程序设计在线评判系统的工作方式和流程，并致力于设计和研发一套适合本校师生使用的在线程序设计能力认证系统。该系统需要实现的主要功能包括考试题库的添加，试卷的生成，老师组织考试。老师可阅卷评卷，将考生代码相似度检测，试卷、代码及成绩的存档，证书的颁发。本系统采用的B/S结构设计开发，无需强制学生老师在电脑上安装客户端，直接通过浏览器访问和使用系统，操作简单，防弊啊维护。本系统是针对程序能力认证开发的信息系统，有较强的针对性，实用性，可操作性，易交互性，安全性和可维护性。

通过使用本课题所研发的程序能力认证系统，老师可以按期组织程序设计能力认证考试，添加考试时设置试卷模板，随机抽取模板对应知识点，对应难度的试卷，并保证每套试卷的难度统一，考试过程中，老师和学生可以看到提交状态及考试排名，考试结束后老师可以阅卷评卷，将学生的代码进行相似度检测，对相似度较高的代码进行重新给分。对整场考试的学生试卷、成绩、代码等信息按照一定文件结构导出方便存档。通过认证的考生，系统自动颁发通过的证书。

本系统的研发和使用，可以有效地避免了人工阅卷的枯燥和阅卷可能出现的错误，大大提高了老师的工作效率。更客观的评价学生的知识的系统性和程序设计能力，前端通用框架的设计，代码的模块化和UI的组件化，使得代码简化，更易于扩充，对信息系统开发前端开发方案有积极作用。

## 1.5 课题实施方案

本课题将采用先对系统总体规划，然后系统功能分模块实施，需求调研与应用开发紧密结合的技术实施方案。

1. 调研相关程序设计能力认证平台，确定系统的具体需求，对系统的进行可行性分析。
2. 根据需求分析设计出概要框架，确定层次结构，确定接口，新型详细的过程设计实现题目评判。
3. 找出实现系统所要解决的技术关键点，给出解决该技术难点的可行解决方案，优化并择优选择实现本课题所要采用的解决方案；
4. 完成系统总体设计，包括提取系统功能，实现系统数据库的设计等；
5. 完成系统详细设计，包括提出系统业务流程，功能模块；
6. 根据前期的总体设计和详细设计，逐步分模块实现系统功能；
7. 完善系统功能，并加强系统在功能、安全和网络适应性等方面的性能；

## 1.6 本章小结

本章对课题所要研究的程序设计能力认证系统的项目背景进行了简要介绍，给出了本课题的项目研究意义，并且介绍了程序设计能力认证系统的研究现状和发展前景，明确了本课题所要实现的项目目标，给出系统的实施方案，为后期系统的开发明确了方向和思路。

# 第2章 系统需求分析

## 2.1 系统介绍

本系统是一套分等级的程序设计能力认证系统，将程序设计能力按照大纲分为多个等级，每个等级定期开展集中的在线考试，考试达到一定分数可获得相应证书，系统题库可由老师或管理员添加，老师或管理员可以组织考试、设计试卷模板、评阅试卷等。系统自动按照试卷模板生成试卷，每份试卷题目不同，难度统一。学生根据老师或管理员处获得的账号密码登陆进行考试。

本系统主要面向校内计算机相关专业师生开发，校内计算机相关专业每年级400多人，需线上考试的课程有算法分析、C语言程序设计、数据结构等，且每门课程会分有单元考试、半期考试和期末考试，即本校计算机相关专业师生对本系统的需求量很大，每年每个学生至少10次使用本系统。通过本系统，可以减轻老师们评阅实验报告和考试试卷的负担，更客观真实了解学生程序设计能力，对学生而言，可激发他们程序设计兴趣，客观评价自己的编程能力。

## 2.2 系统非功能性需求分析

本课题所研发的程序设计能力认证系统的非功能性需求定义如下：

1. 实用性：系统满足电子支付检测报告分析与发布工作的业务要求。
2. 可扩展性：系统开发结构设计合理，方便系统功能扩展。
3. 可靠性：系统运行稳定，努力不发生故障或极少发生故障。
4. 健壮性：系统在发生故障时应有合理的处理和容错机制。
5. 安全性：系统应对数据库中的数据进行备份保护，严格控制系统用户的使用权限。
6. 用户界面：系统界面简约协调，支持PC端Google Chrome，Mozilla Firefox，Internet Explorer 9以上浏览器，暂不支持移动端适配。
7. 软硬件环境：服务器Mysql（5.0）+Tomcat（7.0）+Linux，客户端2G内存，20G硬盘，网卡100/1000Mbps自适应，支持通用操作系统，Microsoft Windows 2000及后继版本，Linux。
8. 性能需求：系统数据量大，至少支持不低于100人的在线开始，系统响应时间不大于5秒。

## 2.3 系统功能需求分析

在对程序设计能力考试平台进行了充分的调研和分析后，提取了考试认证流程，并对本课题所研发的程序能力认证系统进行了功能需求分析，可提取出系统所需实现的功能点，对于管理员和老师：

1. 考试管理： 管理员和老师可以查看所有考试列表，设置试卷模板，以及考试的添加、修改、删除等。
2. 题库管理：管理员和老师可以查看本系统所有题目，对题目进行管理，添加，修改、删除等。
3. 课程管理：管理员和老师可以管理课程及课程对应的知识点。
4. 用户管理：管理员和老师对参加考试的学生信息进行管理，添加、修改、删除等。
5. 考场管理：管理员和老师查看最近的考试，对已结束的考试进行阅卷，对未开始的考试抽题，管理员和老师可以查看进行中的考试的学生在线状态，提交状态，实时排名，考试的题目列表等，考试结束后，管理员和老师可以导出学生的试卷，成绩和代码等信息。
6. 登陆请求：管理员和老师对需要解锁的学生账号进行解锁。
7. 系统设置：管理员和老师可以设置本系统的名字，版权信息，以及一些公告信息等。

对于学生包括：

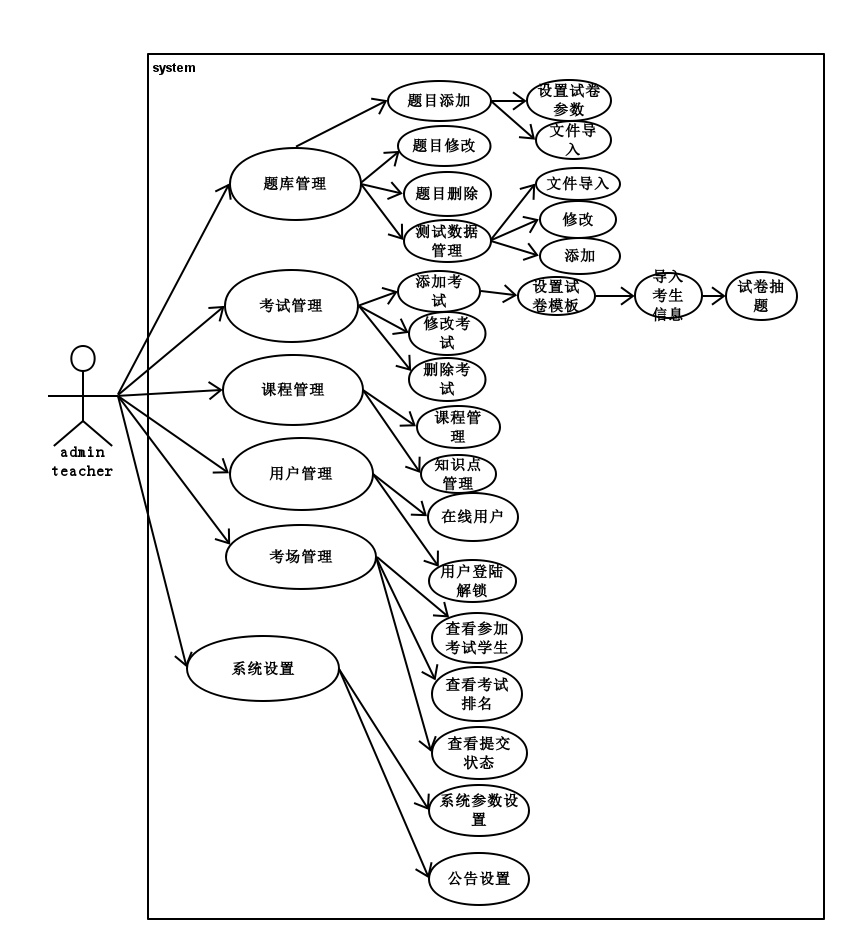
（1）查看题目：学生查看所参加考试的题目。

1. 我的提交：学生可以查看所参加考试自己的提交代码情况。
2. 排名：学生可以查看所有参加本场考试学生的排名情况。
3. FAQS：学生可查看考试注意事项，代码提交注意事项等一些简单的问答

## 2.4 系统用例分析

系统用例分析是系统需求和设计的桥梁，其专注于建立系统的逻辑模型，通过对程序设计能力系统的分析，提取出两种角色：教师或管理员、学生。

### 2.4.1 管理员及老师模块

管理员及老师需要完成对考试认证的管理，包括对题目，试卷的查看、修改、删除、和新建操作。对学生信息的管理，包括学生账号密码与学生本人学号的关联，对学生账号登陆的解锁，添加、修改、查看、删除学生信息。对考场的管理，包括查看考试的学生排名、提交情况、在线情况，还有考试结束后对试卷的评阅。对学生考试试卷的管理，包括对学生试卷、代码、成绩等信息的存档导出，对通过考试认证的学生颁发证书。该模块的用例图如图3-1所示。

**图3-1 管理员教师用例图**

### 2.4.2 学生模块

学生只需要参加考试即可。考试前，需要根据从老师或管理员出获得的账号密码登陆系统进行考试，再次登陆需要老师或者管理员手动解锁。登陆成功后可以查看与自己相关的考试列表，进入某场考试后可以查看题目列表，自己的提交状态，当场考试学生的提交状态及排名。可以查看FAQS获取系统的使用说明及注意事项，可以查看自己的证书。学生模块的用例图如图3-2所示。

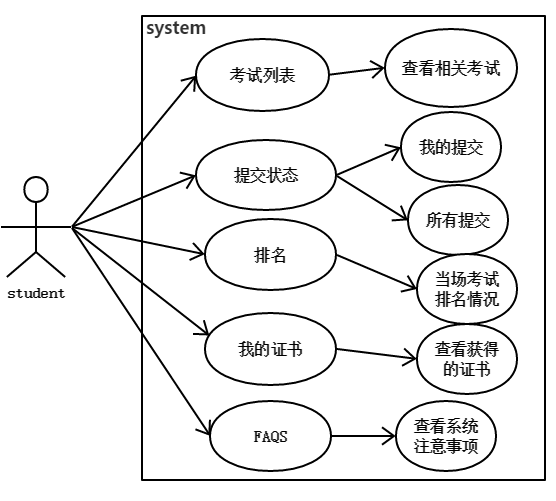


图3-2 学生用例图

在系统的需求分析阶段制定了系统的用例规约。《用例规约描述》（Use Case Specification）是描述项目小组对项目进行需求分析得到的关于用户和系统之间交互作用的文本性描述文档，通过用例规约描述，可进一步说明该系统需求，是下一阶段系统设计的基础，也是测试用例的重要依据。

限于论文篇幅限制，仅选取几个重要的功能点介绍其用例规约。

在老师模块，老师的主要职责是对考试进行管理，包括创建考试、查看考试、修改考试和删除考试，其中创建项目的用例规约描述如图表3-1所示。

**表3-1 创建项目用例规约描述**

|  |  |
| --- | --- |
| 用例名称： | 创建考试 |
| 角色： | 老师或管理员 |
| 用例说明： | 老师或管理员使用系统创建考试 |
| 前置条件： | 老师或管理员登陆系统 |
| 后置条件： | 新的考试创建成功，在考试列表中能够看到该考试信息 |
| 基本事件流： | 1. 老师或管理员进入考试管理模块 2. 选择新建考试功能 3. 填写考试名称、起止时间等基本信息 4. 填写问题模板，配置试卷难度等参数 5. 导入参考学生名单 6. 确认提交考试 |
| 异常事件流： | 6a 考试基本信息填写不完整就提交创建考试  6a.1 系统弹出提示，方案管理员重复步骤3/4完善信息 |
| 字段列表： | 考试名称、考试描述、考试起止时间、题目相关课程、题目相关知识点、题目难度、参加考试学生名单 |

在老师或管理员的另一个重要功能是评阅试卷并进行相似度检测，该功能的用例规约描述下表所示。

**表3-2评阅试卷用例规约描述**

|  |  |
| --- | --- |
| 用例名称： | 评阅试卷 |
| 角色： | 老师或管理员 |
| 用例说明： | 在考试结束后进行阅卷，查看考生代码，相似度检测，重新给分等 |
| 前置条件： | 老师或管理员登陆系统；考试已经结束 |
| 后置条件： | 老师或管理员评阅试卷修改成绩提交成功 |
| 基本事件流： | 1. 进入考场管理模块    1. 选择已经结束的考试    2. 选择阅卷逐一评阅参考学生试卷 2. 老师或管理员阅读学生代码，相似度检测，修改学生获得分数 3. 确认提交，完成学生试卷评阅 |
| 异常事件流： | 3a 修改成绩填写非法成绩，如非数字、超出最高分等。  3a.1根据提示，重新给分，然后再次提交 |
| 字段列表： | 学生ID、试卷ID、题目ID、题目分数、最高分数、题目标题、试卷代码、相似度较高学生代码 |

老师或管理员解锁学生账号也是重要功能，其对应的用例规约描述如表3-3。

**表3-3 学生账号解锁用例规约描述**

|  |  |
| --- | --- |
| 用例名称： | 学生账号解锁 |
| 角色： | 老师或管理员 |
| 用例说明： | 学生账号二次登陆需要管理员或老师解锁 |
| 前置条件： | 学生二次登陆系统；管理员或老师登陆系统；考试进行中 |
| 后置条件： | 学生可再次登陆系统进行考试 |
| 基本事件流： | 1. 老师或管理员进入登陆解锁管理模块 2. 选择需要解锁的学生账号，单个或批量解锁 3. 确认提交解锁学生名单 4. 解锁后的学生登陆系统继续考试 |
| 异常事件流： | 3a 网络不稳定导致解锁失败  3a.1 重复2\3步骤 |
| 字段列表： | 学生ID |

在题库管理模块重要的是对题库的管理，包括新增题目等，其对应的用例规约描述如表3-5所示。

**表3-5 新增题库用例规约描述**

|  |  |
| --- | --- |
| 用例名称： | 新增题库 |
| 角色： | 老师或管理员 |
| 用例说明： | 老师或管理员针对某个课程或知识点添加或导入题目 |
| 前置条件： | 老师或管理员登陆系统；题目不存在 |
| 后置条件： | 在题目列表中能够看到所添加的题目 |
| 基本事件流： | 1. 老师或管理员进入题目管理模块 2. 选择添加题目功能 3. 填写题目信息，包括标题，难度，时间内存限制，描述等等 4. 提交题目 |
| 异常事件流： | 4a 题目添加失败，系统给出相应提示  4a.1 根据系统提示，修改题目信息，重复3/4步骤 |
| 字段列表： | 题目名称、题目难度、时间限制、内存限制、课程知识点、描述、输入输入样例、参考代码 |

而在学生模块，最主要的功能便是参加考试，进入系统选择考试答题。该用例规约描述如表3-6所示。

**表3-6 学生考试用例规约描述**

|  |  |
| --- | --- |
| 用例名称： | 学生考试 |
| 角色： | 学生 |
| 用例说明： | 学生登陆系统参加考试整个流程 |
| 前置条件： | 考试正在进行中；学生登陆系统答题 |
| 后置条件： | 学生答题完毕，提交试卷 |
| 基本事件流： | 1. 学生根据账号密码登陆系统 2. 选择本次需要参加的考试，查看题目 3. 完成题目代码后提交代码 4. 查看提交状态和排名情况 5. 完成答题，提交试卷 |
| 异常事件流： | 3a 提交代码系统判题响应时间过长  3a.1 等待系统判断，系统会给出提示pending，再刷新页面即可 |
| 字段列表： | 学生ID、考试ID、试卷ID、题目ID、题目代码、编译器类型 |

## 2.5 本章小结

为了保证系统的成功开发， 本章首先对系统进行了可行性分析，分别指出了系统在经济、技术、操作等几方面的可行性。紧接着对系统进行了非功能性和功能需求分析，提出了本系统必须实现的功能点和肺功能要求。最后对系统进行用例分析，提出了老师管理员、学生两种角色、分别对这两种角色进行了用例分析，给出了相应的用例图。本章对系统的需求进行了详细的分析，为之后的系统设计和开发做准备。

# 第3章 系统概要设计

## 3.1 系统设计策略

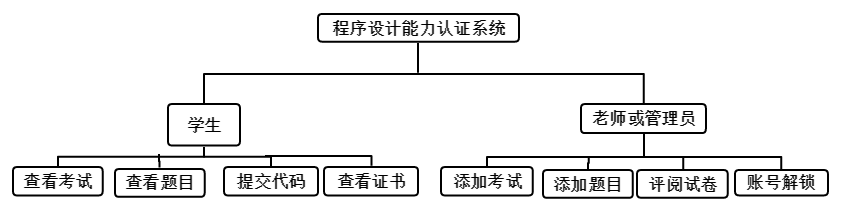
（1）扩展策略：为了方便本系统在将来的扩展功能，在开发过程中，所有接口没有一并写死，根据参数不同来交互数据。

（2）复用策略：本系统在开发过程中，提取公用工具函数，功能组件，以便于在不同模块中调用，减少了代码量，并且在后期开发过程中可以对公共函数或功能组件进行扩充，在开发其他信息系统的时候也可以抽取这些公用的组件。

（3）折衷策略：为了前后端开发进度不相互影响，本系统将前后端开发分离开来，前端数据以mock形式给出，两端制定接口和字段及数据结构的规范，开发完一个阶段后进行联调。

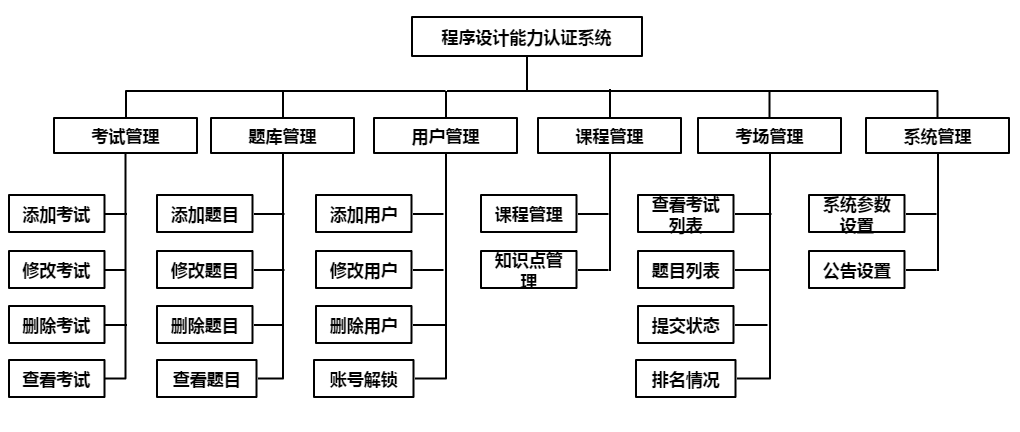
## 3.2 功能结构设计

本系统是程序设计能力认证系统实现，通过对系统的需求分析，可按角色划分系统主要功能模块，老师或管理员模块、学生模块。系统的总结逻辑结构设计如下图所示



### 3.2.1 管理员或老师模块

老师或管理员可登陆系统，对系统中的考试进行管理，对题库进行管理，对学生信息进行管理，并系统参数，当不需要使用系统时可退出系统。其功能结构图如图4-1所示。



**图4-1 老师或管理员模块功能结构图**

### 3.2.2 学生模块

学生可登陆系统，查看考试列表，参加进行中的考试，提交代码，查看提交状态及排名情况等，当用户不需要使用系统时，可退出系统。审核员模块的功能结构图如图4-2所示。

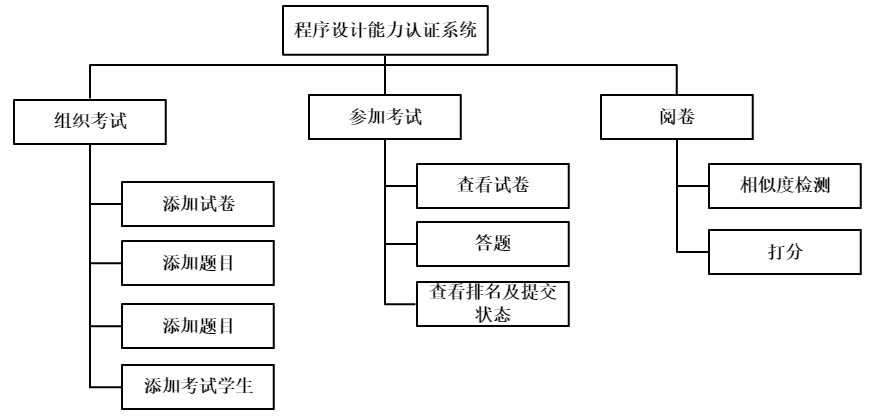
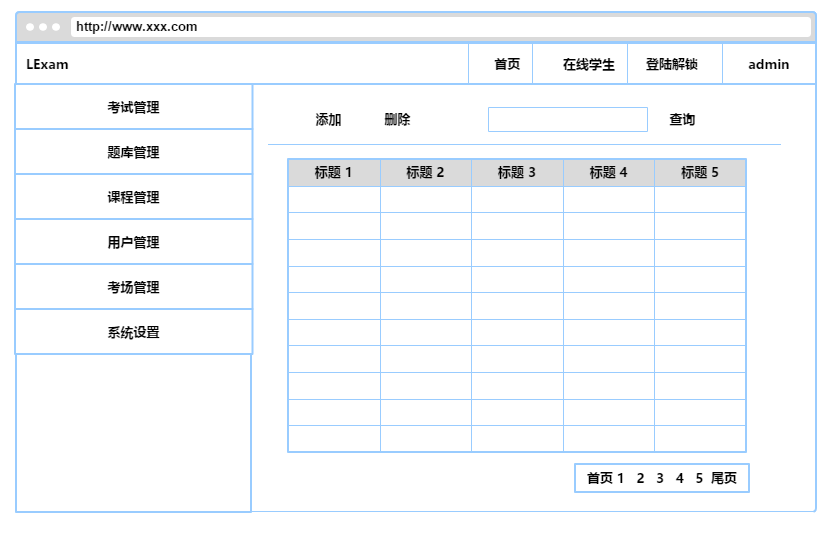


图4-2 学生模块功能结构图

## 3.3 界面与模块设计

### 3.3.1 界面设计

界面风格统一采用bootstrap样式，bootstrap支持响应式布局，兼容性较好，且整体风格简约，使用简单。管理端用传统管理系统布局进行排版，左侧为导航，右侧为内容，简单布局如下图：

点击左侧导航后页面跳转到相应模块，每个页面布局基本如上图所示，保证了风格统一，操作简单的原则。

### 3.3.1 模块设计

整个系统可分为以下几个模块：题库管理、课程管理、用户管理、考场管理和考试管理，面向使用对象为老师或管理员，每个模块按照其名词命名。题库管理主要用于管理员或老师对系统里题目的增删改操作，课程管理是管理系统题库相关的课程和知识点，以便在添加考试的时候可以按照课程知识点来选择题目设置试卷模板，用户管理是管理参加考试的学生信息，以及为学生二次登陆系统解锁。考场管理是查看考试的在线学生，以及排名和提交代码状态等情况，考试管理是组织考试，新建考试，然后导入参考学生，生成试卷。

## 3.4 数据库设计

### 3.4.1 数据库环境

数据库管理工具：MySQL 5.6

数据库设计与编程工具：Power Designer

数据库可视化工具：MySQL Navicat

### 3.4.2 数据库命名规则

（1）表名命名规则

数据库表的命名以是名词且都为小写，如exam、problem、paper等等。如果表名由几个单词组成，则单词间用下划线("\_")分割，paper\_progblen, problem\_know, submit\_result等。表名尽量用全名，表名限制在30个字符内。当表的全名超过30字符时，可用缩写来减少表名的长度，如information --> info。

（2）表字段名命名规则

字段名为小写，有意义的单词或单词的缩写。如果字段由几个单词组成，则单词间用下划线("\_")分割，如exam\_id, create\_time等。字段名限制在30个字符内。当字段名超过30字符时，可用缩写来减少字段名的长度。

（3）索引命名规则

索引须按照IDX\_table\_<column>\_<column>,其中<table>是建立索引的表名，<column>是建立索引的字段名。索引名限制在30个字符内。当索引名超过30字符时，可用缩写来减少索引名的长度。

（4）主键、外键命名规则

主键按照PK\_<table>命名，其中<table>为数据库表名。主键按照UK\_<table>\_<column>命名，其中<table>为表名，<column>为字段名。外键按照FK\_<pppp>\_<cccc>\_<nn>的规则命名。

### 3.4.3 数据库概念设计

EER（Extended Entity-Relationship）图，即扩展实体-关系模型，提供了表示数据库中实体类型、实体属性和实体间联系的方法，是数据库概念设计常用到的工具。通过对本系统进行分析，提取出本系统的主要实体，给出了系统EER图。

(………..这儿应该有张图………..)

### 3.4.4 数据库安全设计

(1) 防止用户直接操作数据库：用户只能用账号登陆到系统，通过应用本系统访问数据库，而没有其他途径操作数据库。

(2) 用户账号密码的加密方法：对用户账号的密码（数据库系统的用户和密码）进行加密处理，确保在任何地方都不会出现密码的明文。

(3) 角色与权限：确保每个角色对数据库表的操作权限，如创建、检索、更新、删除等。每个角色拥有刚好能够完成任务的权限，不多不少，在应用是在为用户分配角色，则每个用户的权限等于他所兼角色的权限之和。

### 3.4.5 数据库表设计

根据设计的系统EER图，设计了系统所需的数据库表，共有20张表。数据库表汇总情况如表4-1所示。

**表4-1 数据库表汇总表**

|  |  |
| --- | --- |
| **表名** | **功能说明** |
| exam\_info | 记录考试的相关信息，包括标题、描述、起止时间、允许IP段等 |
| exam\_judger | 记录考试判题编译器，包括考试编号、编译器编号等 |
| exam\_notice | 记录考试通知公告，包括公告编号，考试编号、添加时间、内容等 |
| exam\_paper | 记录考试试卷，包括试卷编号、考试编号、学生编号、考卷总分等 |
| exam\_param | 记录考试模板，包括模板编号、考试编号、难度、分数等 |
| judger\_info | 记录编译器信息，包括编译器编号、编译器名称、是否开启等 |
| knowledge | 记录课程知识点，包括知识点编号、知识点名称、是否是课程等 |
| paper\_problem | 记录试卷里的题目，包括试卷题目编号、试卷编号、提交次数等 |
| problem\_info | 记录题目信息，包括题目编号、题目标题、题目描述、输入提示等 |
| problem\_know | 记录题目考点，包括题目考点编号、题目编号、考点编号等 |
| school\_major | 记录学校专业信息，包括专业编号、专业名称、学院名称等 |
| site\_info | 记录系统信息，包括系统参数编号、参数类型、参数名称等 |
| submit\_info | 记录学生提交状态信息，包括提交编号、学生编号、考试编号等 |
| submit\_print | 记录考试打印信息，包括考试打印请求编号、请求时间等 |
| submit\_result | 记录提交结果，包括提交编号、测试编号、显示状态等 |
| submit\_similarity | 记录相似度检测结果，包括相似度比较编号、题目编号，相似值等 |
| user\_info | 记录学生的账号信息，包括学生姓名、考场、最近登陆时间等 |
| user\_login\_log | 记录用户登陆日志，包括登陆日志编号、用户编号、登陆时间等 |
| user\_profile | 记录用户资料，包括用户名称、真实姓名、班级、学号等 |
| user\_release\_lock | 记录学生登陆解锁信息，包括学生编号、是否被锁、登陆IP等 |

系统中共存在20张表，限于论文篇幅限制，仅对最重要的几个表进行介绍。

exam\_info表用于记录考试信息，包括考试的编号、考试的名称、考试描述、考试开始时间、结束时间、允许访问IP段等信息，其中考试编号是逐渐，用于唯一标识一个考试，考试编号也是外键，exam\_info表的详细设计如表4-2所示。

**表4-2 exam\_info表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段描述 | 默认值 | 数据类型 | 主键 | 外键 | 不为空 |
| exam\_id | 考试编号 |  | INT(11) | √ | √ | √ |
| title | 考试名称 |  | VARCHAR(255) |  |  | √ |
| description | 考试描述时间 |  | LONGTEXT |  |  |  |
| start\_time | 考试开始时间 |  | DATETIME |  |  | √ |
| end\_state | 考试结束时间 |  | DATETIME |  |  | √ |
| allow\_ip | 允许登陆的IP地址段 |  | VARCHAR(255) |  |  | √ |

exam\_paper是考试试卷信息表，包括考试试卷编号、试卷属于考试的编号、学生编号、试卷总分、试卷的ac数量、完成试卷耗时、试卷是否被批阅等信息。其中试卷编号是主键，用于唯一地标识业务，项目学生编号和试卷对应考试的编号是外键。exam\_paper表的详细设计如表4-3所示。

**表4-3** exam\_paper**表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段描述 | 默认值 | 数据类型 | 主键 | 外键 | 不为空 |
| expr\_id | 试卷编号 |  | INT(11) | √ |  | √ |
| exam \_id | 试卷对应考试编号 |  | INT(11) |  | √ | √ |
| user\_id | 学生编号 |  | INT(11) |  | √ | √ |
| score | 试卷总分 |  | INT(11) |  |  |  |
| aced\_count | 试卷的通过数量 |  | INT(11) |  | √ | √ |
| used\_time | 完成试卷的耗时 |  | INT(11) |  | √ | √ |
| is\_marked | 试卷是否被批阅 |  | INT(1) |  |  |  |

program表用于记录题目的相关信息，该表字段较多，包括题目编号、题目标题、描述、入库时间、提交次数、难度限制时间等，表的主键为题目编号。其对应的表详细设计如表4-4所示。

**表4-4 program表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段描述 | 默认值 | 数据类型 | 主键 | 外键 | 不为空 |
| prob\_id | 题目编号 |  | INT(11) | √ | √ | √ |
| title | 题目标题 |  | VARCHAR(255) |  |  | √ |
| description | 描述 |  | LONGTEXT |  |  | √ |
| create\_time | 入库时间 |  | DATATIME |  |  | √ |
| total\_submit | 提交次数 |  | INT(11) |  |  | √ |
| aced\_ num | 通过次数 |  | INT(11) |  |  | √ |
| level | 难度 |  | INT (11) |  |  | √ |
| limit\_time | 限制时间 |  | INT(11) |  |  | √ |
| limit\_memory | 限制内存 |  | INT (11) |  |  | √ |
| input\_tip | 输入提示 |  | TEXT |  |  | √ |
| output\_tip | 输出提示 |  | TEXT |  |  | √ |
| input\_sample | 输入样例 |  | TEXT |  |  | √ |
| output\_sample | 输出样例 |  | TEXT |  |  | √ |
| standard\_source | 参考代码 |  | LONGTEXT |  |  | √ |
| testdata\_num | 测试数据数量 |  | INT(11) |  |  |  |
| code\_limit | 代码长度限制 |  | INT(11) |  |  |  |
| judge\_model | 判题模式 | 0:standard  1:spical judge | INT(11) |  |  |  |
| author | 作者 |  | VARCHAR(255) |  |  |  |

submit\_info表记录提交代码信息，包括提交编号、提交用户编号、提交问题编号、提交时间、编译器类型、考试编号、试卷编号、考题编号、提交IP、源码、提交编号是表主键，用于唯一标识提交。submit\_info表的详细设计如下表所示。

**表4-5 submit\_info表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段描述 | 默认值 | 数据类型 | 主键 | 外键 | 不为空 |
| subm\_id | 提交编号 |  | INT(11) | √ |  | √ |
| user\_id | 提交学生编号 |  | INT(11) |  | √ | √ |
| prob\_id | 提交问题编号 |  | INT(11) |  | √ | √ |
| subm\_time | 提交时间 |  | DATATIME |  |  | √ |
| jud\_id | 编译器类型 |  | INT(11) |  | √ | √ |
| exam\_id | 考试编号 |  | INT(11) |  |  |  |
| papr\_id | 试卷编号 |  | INT(11) |  |  |  |
| papbd\_id | 题目编号 |  | INT(11) |  |  |  |
| subm\_ip | 提交IP |  | VARCHAR(255) |  |  | √ |
| subm\_source | 源码 |  | LONGTEXT |  |  | √ |
| is\_teacher\_text | 是否老师测试 |  | INT(11) |  |  | √ |

user\_info表记录考生信息，包括学生编号、学生打印编号、登陆账号、密码、考场、是否是老师、是否是管理员、是否锁定、最后登录时间、添加时间，其中学生编号是主键，可唯一标识一个考生。user\_info表的详细设计如表4-6所示。

**表4-6** **user\_info表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段描述 | 默认值 | 数据类型 | 主键 | 外键 | 不为空 |
| user\_id | 学生编号 |  | INT(11) | √ |  | √ |
| usrpf\_id | 学生打印编号 |  | INT(11) |  |  | √ |
| username | 登陆账号 |  | VARCHAR(255) |  |  | √ |
| password | 登陆密码 |  | VARCHAR(255) |  |  | √ |
| classroom | 考场 |  | VARCHAR(255) |  |  |  |
| is\_teacher | 是否老师 |  | INT(1) |  |  | √ |
| last\_ogin | 最后登录时间 |  | DATETIME |  |  | √ |
| add\_time | 添加时间 |  | DATETIME |  |  | √ |

## 3.5 本章小结

本章对程序设计能力认证系统进行了总体设计，首先介绍本系统设计策略，然后分别从老师和学生模块，给出了系统的功能结构图。紧接着对系统的数据库设计进行了介绍，首先介绍了数据库的开发、编码环境，然后介绍了本系统在设计数据库时所遵循的命名规范，给出了系统的EER图，最后将系统中所涉及的表进行了介绍，限于论文页面的限制，只选取了系统中最主要的表进行介绍。通过本章的介绍，可对本系统的总体设计有很好的了解，为之后的系统详细设计与开发做准备。

# 第4章 系统详细设计与实现

## 4.1 技术难点解决

### 4.1.1 系统自动判题

系统自动评判模块是整个系统的核心模块，它负责编译并运行学生提交的代码，并给学生和老师反馈评判的结果。传统机器评卷的方式是将考试系统判卷程序安装在考生使用的机器上（客户端），考生交卷后，考试系统通过匹配位于客户端的out.dat文件和答案评出考生成绩，在将考试成绩上传至服务器。这种方式可以让考生一个字母都不写同事获得该题满分，这种作弊方式俗称“打表”。

本系统将判卷模块设置在服务器上，考生交卷后，服务器获得考生所提交程序源码，以重定向的方式把该代码的运行结果保存值服务器，通过这样的方式使得out.dat对考生不可见，防止了考生通过“打表”作弊。

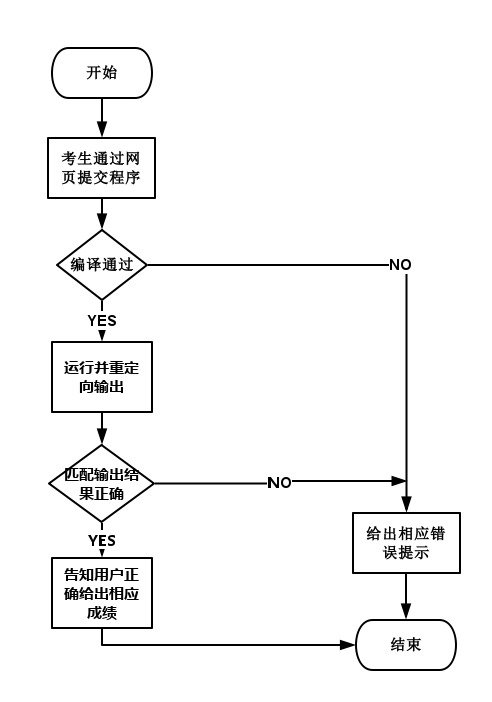
基本流程和详细流程图如下所示（以C语言为例）：

图4-1基本流程图

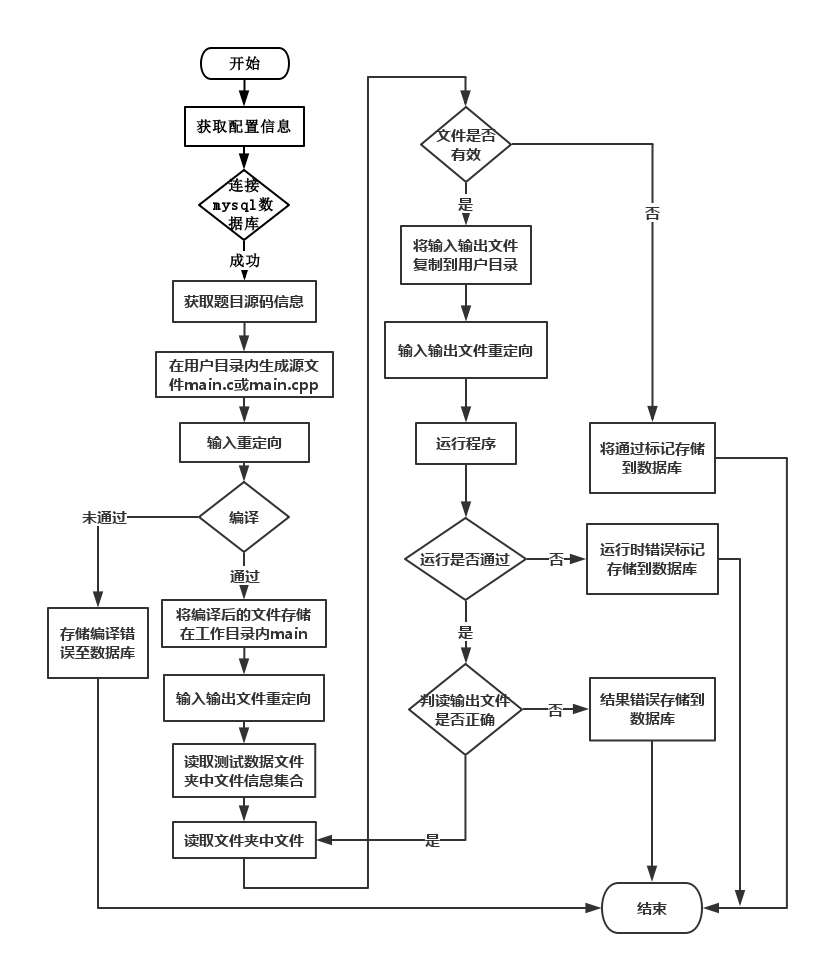


图4-2详细流程图

判题关键代码如下：

**public** **void** run() {

**try** {

String script\_path = PropertyUtil.*getPropertyByName*(

"script\_config.properties", "script\_path");

String script\_name = PropertyUtil.*getPropertyByName*(

"script\_config.properties", "script\_name");

String source\_path = PropertyUtil.*getPropertyByName*(

"script\_config.properties", "source\_path");

//获取源码文件后缀

String file\_name = PropertyUtil.*getPropertyByName*(

"script\_config.properties", language);

**if** (file\_name == **null**) {**return**;}

// 生成源码文件

FileUtil.*generateFile*(source, script\_path + source\_path,file\_name);

//执行脚本

Runtime rt = Runtime.*getRuntime*();

Process proc = rt.exec(cmd.toString());

String errMsg = StreamUtil.*output*(proc.getErrorStream());

String consoleMsg = StreamUtil.*output*(proc.getInputStream());

//判断结果

**if** (errMsg != **null** && "".equals(errMsg) == **false**) {

*logger*.debug("RuntimeError:" + errMsg);

} **else** {

**if** (consoleMsg.indexOf("ERROR:TESTDATA IS INCORRECT") != -1) {

*logger*.debug("SystemError:" + errMsg);

} **else** {

Pattern pattern = Pattern.*compile*("\\{'memoryused': (.+?)L, "+ "'timeused': (.+?)L, "+ "'result': '(.+?)', "+ "'errMsg': '(.\*?)', "+ "'testdata\_id': (.+?)\\}");

Matcher matcher = pattern.matcher(consoleMsg);

**while** (matcher.find()) {

*logger*.debug(matcher.group(1));

*logger*.debug(matcher.group(2));

*logger*.debug(matcher.group(3));

*logger*.debug(matcher.group(4));

*logger*.debug(matcher.group(5));

}

}

}

}

}

### 4.1.2 代码相似度检测

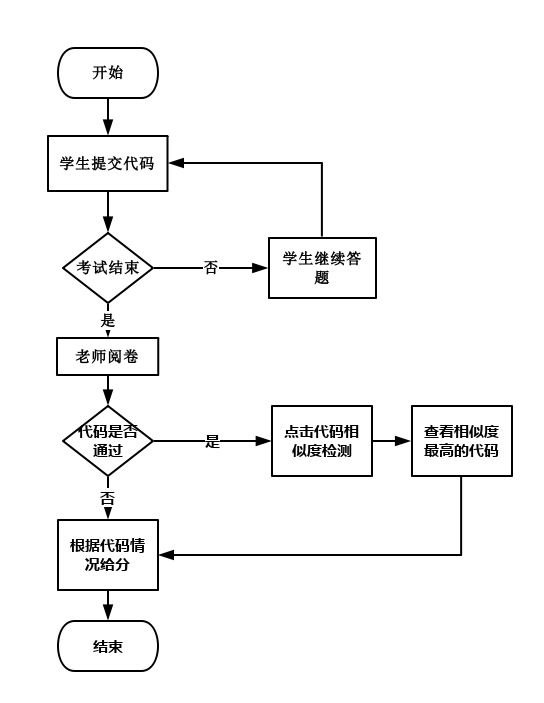
代码相似度检测也是系统的核心功能，由于系统接收提交的代码量很大，参加考试的学生人数也多，难免出现抄袭等现象。所以需要对学生提交代码进行相似度检测。在老师进行阅卷的时候，老师可以手动操作，系统自动列举出与当前批阅代码相似度最高的前五份代码，老师对比代码，若是发现抄袭，则重新给分。本系统采用余弦向量算法实现相似度检测，先用es-ik进行代码分词，得到两份代码的词频向量，然后计算两个向量的预先相似度，值越大相似度越高。检测流程图如下图所示：

图4-3 相似度检测流程图

相似度检测关键代码如下：

**private** **static** Double calculateCos(LinkedHashMap<String, Integer> first,LinkedHashMap<String, Integer> second){

List<Map.Entry<String, Integer>> firstList = **new** ArrayList<Map.Entry<String, Integer>>(first.entrySet());

List<Map.Entry<String, Integer>> secondList = **new** ArrayList<Map.Entry<String, Integer>>(second.entrySet());

//计算相似度

**double** vectorFirstModulo = 0.00;//向量1的模

**double** vectorSecondModulo = 0.00;//向量2的模

**double** vectorProduct = 0.00; //向量积

**int** secondSize=second.size();

**for**(**int** i=0;i<firstList.size();i++){

**if**(i<secondSize){

vectorSecondModulo+=secondList.get(i).getValue().doubleValue()\*secondList.get(i).getValue().doubleValue();

vectorProduct+=firstList.get(i).getValue().doubleValue()\*secondList.get(i).getValue().doubleValue();

}

vectorFirstModulo+=firstList.get(i).getValue().doubleValue()\*firstList.get(i).getValue().doubleValue();

}

**return** vectorProduct/(Math.sqrt(vectorFirstModulo)\*Math.sqrt(vectorSecondModulo));

}

## 4.2 系统模块设计与实现

### 4.2.1 公共登陆模块

根据对系统功能需求的分析，设计了系统的公共登陆模块，为了严格限制用户权限，需要对登陆系统的用户进行身份验证。用户访问系统首页，输入账号密码，登陆系统，系统会根据用户输入的账号和密码进行验证，如果验证成功，则跳转到用户角色对应的模块页面，如果验证失败，则系统会给出提示。

登陆模块的时序图如图4-4所示:

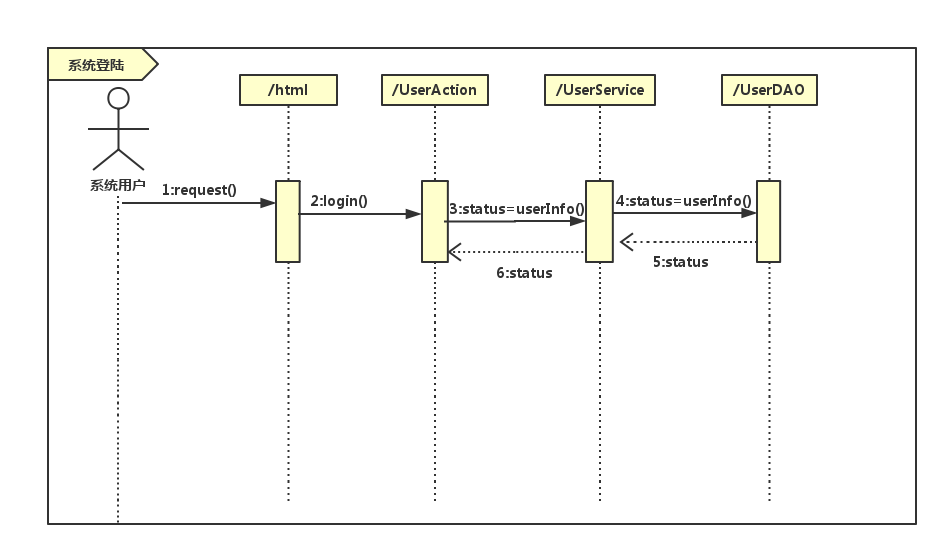


图4-4 登陆功能时序图

便于用户使用，给用户良好的交互体验，系统设计了简约大方且符合用户操作习惯的登陆界面，如图4-5所示：



图4-5 系统登陆界面效果图

用户登陆模块的关键代码如下：

**public** String login() **throws** UnsupportedEncodingException, IOException {

Map<String, Object> res = **this**.userService.login(username, password,**this**.getIp());

Map<String, Object> map = **new** HashMap<String, Object>();

map.put("status", res.get("status"));

map.put("isTeacher",res.get("data") == **null** ? **null** : ((UserInfo) res.get("data")).getIsTeacher());

map.put("isAdmin",res.get("data") == **null** ? **null** : ((UserInfo) res.get("data")).getIsAdmin());

map.put("userId",res.get("data") == **null** ? **null** : ((UserInfo) res.get("data")).getUserId());

map.put("username",res.get("data") == **null** ? **null** : ((UserInfo) res.get("data")).getUsername());

**if** ((Integer) res.get("status") == 1) {

**this**.saveSession(map);

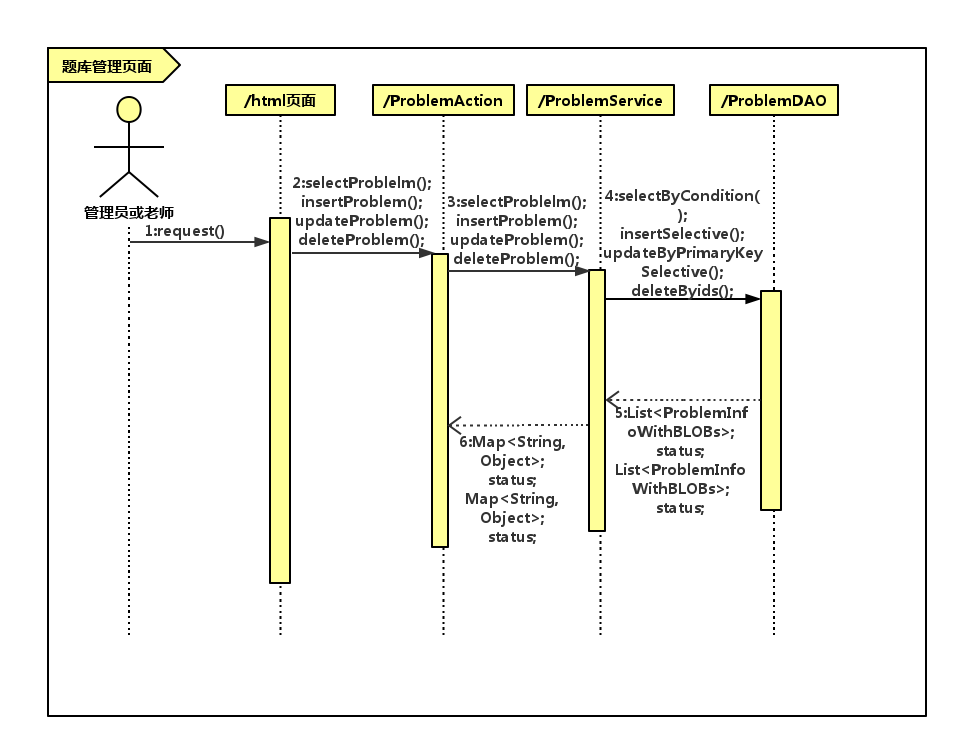
}

JsonUtil.*sendJson*(ServletActionContext.*getResponse*(), map);

**return** ***SUCCESS***;

}

### 4.2.2 题库管理模块

老师或管理员登陆系统并通过身份验证之后，可进入后台管理界面，页面左侧显示管理员或老师可使用的功能导航，右侧显示操作按钮和系统中已经存在的题目列表，点击操作按钮可以添加、修改和删除题目。题库管理功能的时序图如下图所示：

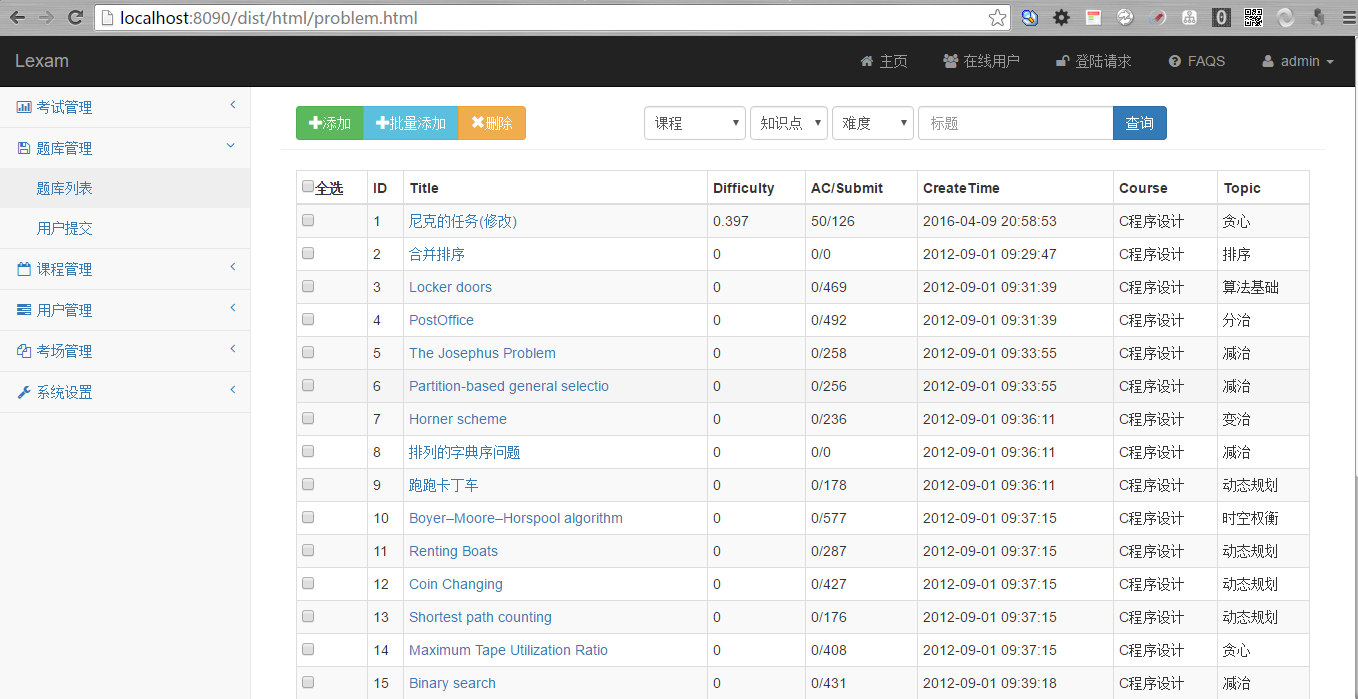
进入题库管理页面后，默认显示的题目列表，如图4-6所示：

图4-6题目列表页面

题库管理中，主要是对题目的增加、修改、删除操作，这三个操作中增加和修改用的是同一个页面，根据是否传入题目编号区分，删除操作直接在列表页面完成。编辑题目页面如图4-7所示：

### 4.2.3 考试管理模块

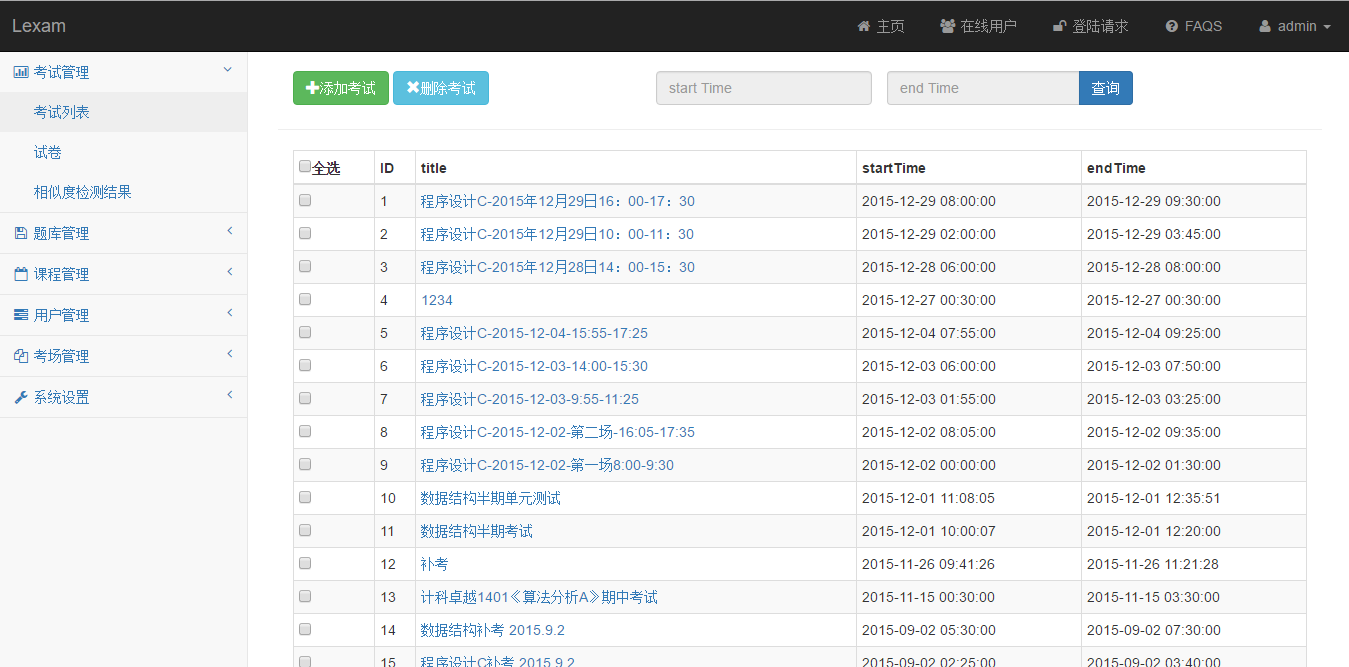
老师或管理员登陆系统并通过身份验证之后，可进入系统，页面左侧同样是功能导航，包括本系统所有的管理功能，右侧是默认正在进行中的考试。页面效果图如图4-8所示：

图4-7 考试管理主页面

点击列表中某场考试，可以编辑修改考试试卷参数，添加考试可以通过设置试卷模板新建考试。考试编辑页面如图4-8所示：

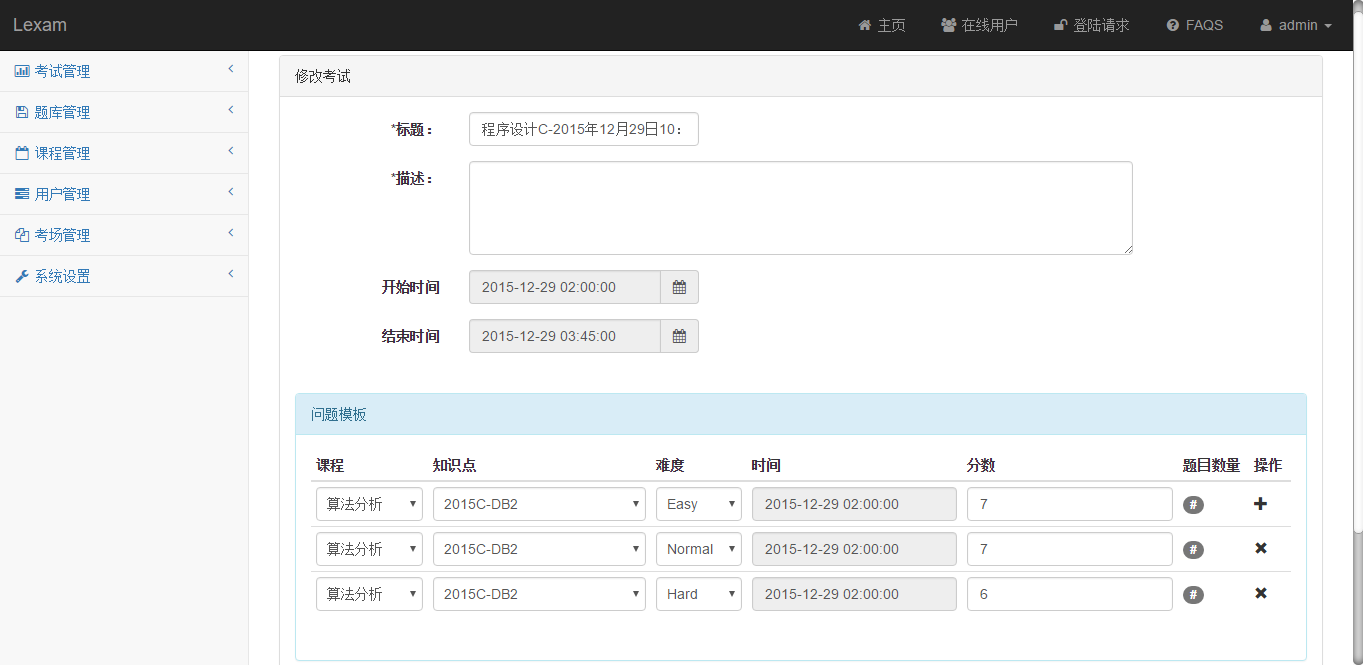


图4-8 考试编辑页面

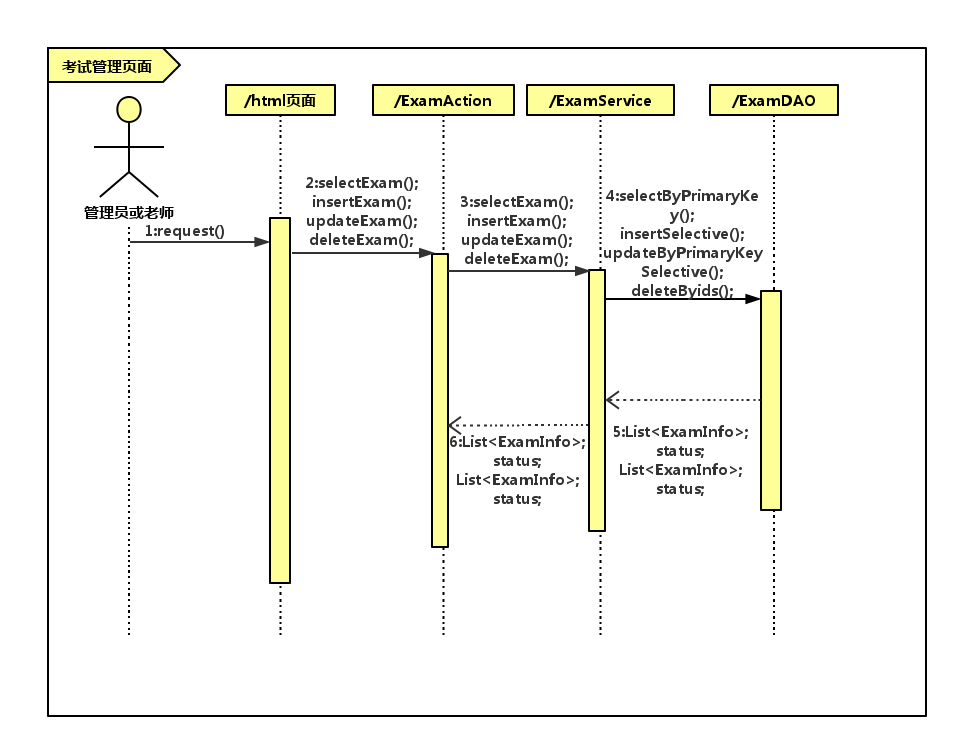
考试管理是本系统重点功能，程序设计能力认证系统里最基本就是考试。老师或管理员按期组织认证考试，新建一场考试需要提前组建好试卷模板，设置考试参数，设置试卷题目相关课程、知识点、题目难度、以及每个题的分数。设置完成后导入考试学生名单，到此考试就新建成功，然后在考试列表出开启抽题，就为每个学生生成试卷，题目不同，但难度统一。考试完成后老师可以阅卷，查看每个学生试卷，进行相似度检测，重新打分。考试管理的功能的时序图如图4-9所示：

图4-9 考试管理功能时序图

考试管理的关键代码如下：

**public** String insertExam() **throws** UnsupportedEncodingException, IOException {

Map<String, Object> map = **new** HashMap<String, Object>();

ExamInfo examInfo = **this**.getExamInfo(title, startTime, endTime,allowIp, description);

List<ExamParam> examParamList = **this**.getExamParamList(

ArrayUtil.*StringToArray*(knowIds, ","),

ArrayUtil.*StringToArray*(levels, ","),

ArrayUtil.*StringToArray*(scores, ","));

**try** {

**int** pos = **this**.examInfoService.insertOne(examInfo, examParamList);

map.put("status", pos > 0 ? 1 : 0);

} **catch** (DataAccessException e) {

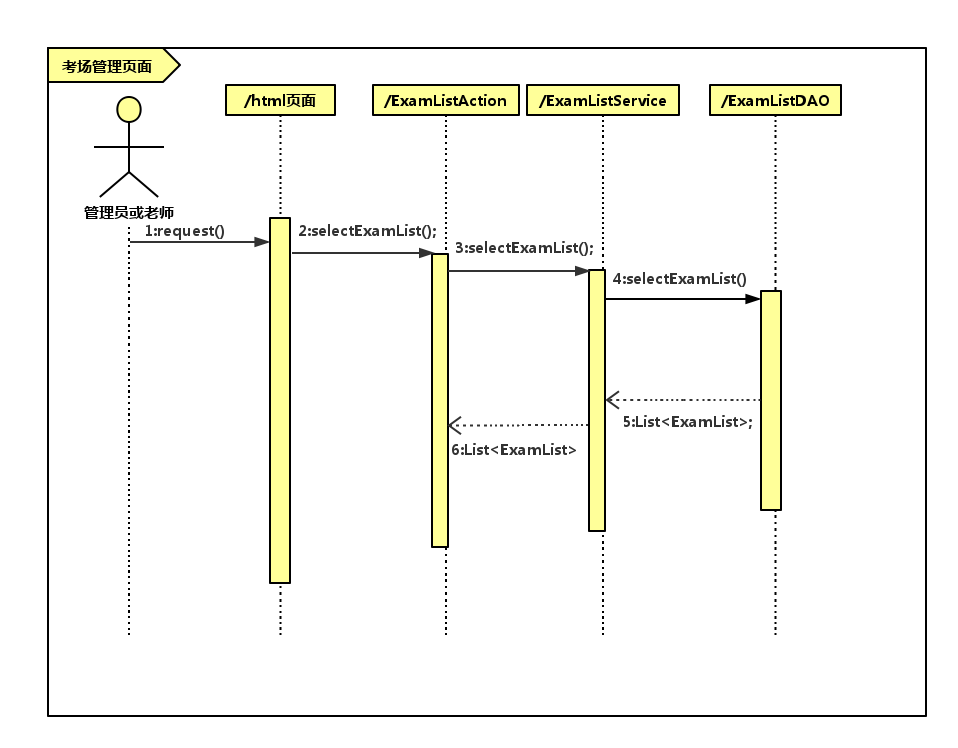
map.put("status", 0);

} JsonUtil.*sendJson*(ServletActionContext.*getResponse*(), map);

**return** ***SUCCESS***;

}

### 4.2.4 考场管理模块

老师或管理员登陆系统并通过身份验证之后，可进入系统进入考场管理模块，由于本系统后台管理页面布局一致，所以左侧同样是功能导航，而右侧为最近的考试列表。主页面如图4-10所示：

点击某一场进行中的考试后，可以查看该场考试的题目列表、提交状态、排名、公告等信息，具体页面效果如图4-11所示

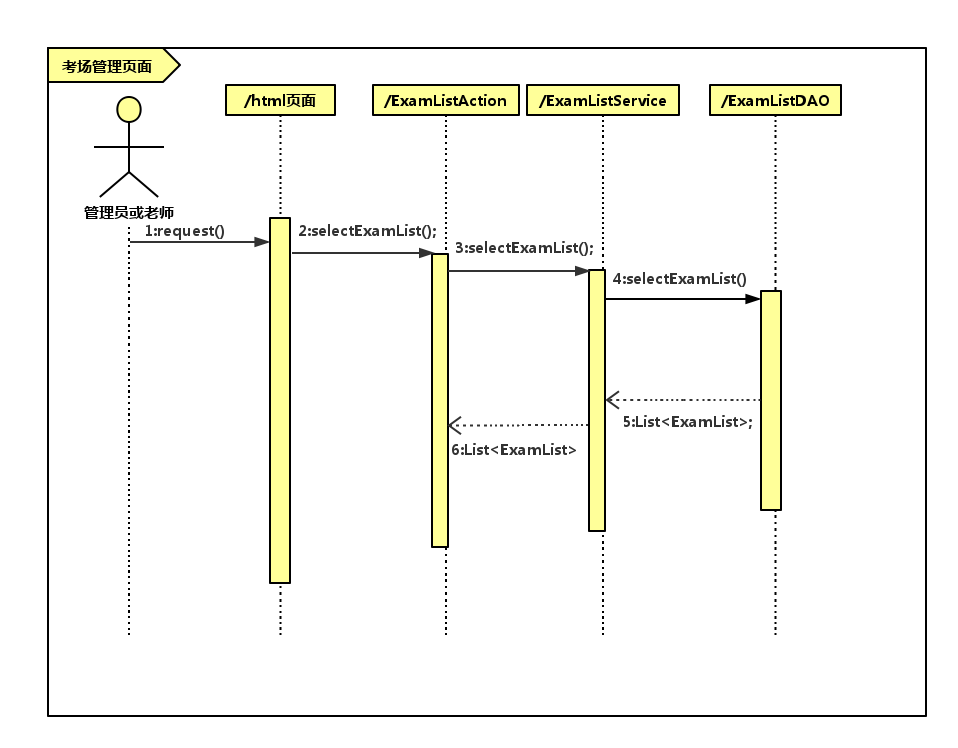
考场管理模块的主要功能是老师或管理员实时了解当前考试的状态，包括对本场考试的题目的查看、本场考试学生的代码提交状态、考生排名、和考试的公告信息，主要是一些显示的功能，涉及到表单操作的功能很少。考场管理模块的时序图如图4-12所示：

图4-12 考场管理模块时序图

考场管理的关键代码如下：

**public** String updateExam() **throws** UnsupportedEncodingException, IOException {

Map<String, Object> map = **new** HashMap<String, Object>();

ExamInfo examInfo = **this**.getExamInfo(title, startTime, endTime,

allowIp, description);

examInfo.setExamId(examId);

List<ExamParam> examParamList = **this**.getExamParamList(

ArrayUtil.*StringToArray*(knowIds, ","),

ArrayUtil.*StringToArray*(levels, ","),

ArrayUtil.*StringToArray*(scores, ","));

List<Integer> expmidsList = ArrayUtil.*StringToArray*(expmIds, ",");

Integer count = expmidsList.size();

**for** (**int** i = 0; i < count; i++) {

examParamList.get(i).setExpmId(expmidsList.get(i));

examParamList.get(i).setExamId(examId);

System.***out***.println(examParamList.get(i));

}

**try** {

Integer flag = **this**.examInfoService.updateOne(examInfo,

examParamList);

map.put("status", flag > 0 ? 1 : 0);

} **catch** (DataAccessException e) {

map.put("status", 0);

}

JsonUtil.*sendJson*(ServletActionContext.*getResponse*(), map);

**return** ***SUCCESS***;

}

## 4.3 本章小结

本章对程序设计能力认证系统的详细设计与实现进行了介绍，首先介绍了本系统在实现过程中需要解决的关键技术，并给出了系统采用的解决方案。接着分模块对系统的详细设计与实现进行了介绍，由于系统功能点较多，限于文章篇幅，只选取了几个关键模块的关键功能进行介绍，并展示了功能点的实现效果和关键代码。通过本章对本系统详细设计与实现的介绍，可进一步了解本系统。

# 第5章 系统测试

## 5.1 测试原则

当开发人员已经全部模块开发完成，各单元模块已经过开发人员代码检查和程序互测（单元测试），单元测试后，各单元模块集成在一起，独立测试正式开始。

系统测试准则为实现了本项目的软件需求规格说明书中所有功能，在操作使用中没有出现错误，系统运行及业务处理正常。确保测试的功能正常，其中包括导航，数据输入，处理和检索等功能。测试中所发现的错误已经全部得到修复。非功能需求（包括性能、安全性等）满足需求规格说明书中规定要求。

当所测内容均都达到软件需求规格说明书要求，所发现的问题已全部解决。遗留问题中无致命性和严重性的错误，一般性错误不能超过总数的2%，轻微性错误不能超过8%，遗留问题总数不能超过问题总数的10%，测试可结束，当系统在部署期间或使用过程中发现严重错误或客户需求已发生重大变化时，测试可以再启动。

## 5.2 测试环境

硬件环境：Core i5-3320M 双核处理器 2.6GHz 8GB DDR3内存

软件环境：Windows 10 专业版

Eclipse Mars Release (4.5.0)

Sublime Text 3103

Chrome devtools

## 5.2 测试结果

### 5.2.1 功能测试

1、“系统登陆”测试

**表5-1 “系统登陆”功能测试**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | UC-001 | | | |
| **功能描述** | 老师或管理员、学生登陆系统使用系统功能 | | | |
| **用例目的** | 对不同角色身份验证进行功能测试 | | | |
| **用例前提** | 用户进入系统登陆页面 | | | |
| **子用例编号** | **输入/动作** | **期望的输出/响应** | **实际输出** | **状态** |
| UC-001-01 | 1. 学生在登陆表单中输入登录账号 2. 学生在登陆表单中输入错误登录密码 3. 点击登陆按钮 | 1. 系统提示账号或密码错误，登陆失败 2. 焦点自动定位到输入框要求学生重新输入账号密码 | 1. 系统提示账号或密码错误，登陆失败  2．焦点自动定位到输入框要求学生重新输入账号密码 | 正确 |
| UC-001-02 | 1. 学生在登陆表单中输入登录账号 2. 学生在登陆表单中输入正确的登陆密码 3. 点击登陆按钮 | 1.登陆成功  2.自动跳转到学生登陆首页 | 1.登陆成功  2.自动跳转到学生登陆首页 | 正确 |
| UC-001-03 | 1. 老师或管理员在登陆表单中输入登录账号 2. 在登陆表单中输入正确的登陆密码 3. 点击登陆按钮 | 1.登陆成功  2.自动跳转到系统后台管理界面 | 1.登陆成功  2.自动跳转到系统后台管理界面 | 正确 |

2．“考试管理”测试

**表5-2 “考试管理”功能测试**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | UC-002 | | | |
| **功能描述** | 老师或管理员登陆系统后进行考试的添加、修改、删除 | | | |
| **用例目的** | 对老师或管理员对考试管理进行功能测试。 | | | |
| **用例前提** | 1.老师或管理员通过系统身份验证登录系统  2. 进入考试管理页面 | | | |
| **子用例编号** | **输入/动作** | **期望的输出/响应** | **实际输出** | **状态** |
| UC-002-01 | 1.进入考试管理页面  2.点击添加考试  3.填写考试相关参数  4.设置考试试卷模板  5.点击保存 | 1.系统提示考试添加成功。  2.跳转到考试列表页面，添加的考试已存在于列表中 | 1.系统提示考试添加成功。  2.跳转到考试列表页面，添加的考试已存在于列表中 | 正确 |
| UC-002-02 | 1.进入考试管理页面  2.点击添加考试  3.填写考试相关参数  4.设置考试试卷模板  5.点击保存并添加另一个 | 1.系统提示考试添加成功  2.系统自动清空表单数据以便继续添加考试。 | 1.系统提示考试添加成功  2.系统自动清空表单数据以便继续添加考试。 | 正确 |
| UC-002-03 | 1.进入考试管理页面  2.点击添加考试  3.填写考试相关参数  4.设置考试试卷模板  5.点击保存并继续编辑 | 1.系统提示保存成功  2.系统停留在当前页面，并且表单数据为清空  3.系统向服务端提交表单数据 | 1.系统提示保存成功  2.系统停留在当前页面，并且表单数据为清空  3.系统向服务端提交表单数据 | 正确 |
| UC-002-04 | 1.进入考试管理页面  2.点击添加考试  3.填写错误或未填关键参数  4.点击保存并继续编辑或者保存 | 1.系统提示必填字段不能为空  2.光标焦点自动定位到未填写或填写错误所在位置 | 1.系统提示必填字段不能为空  2.光标焦点自动定位到未填写或填写错误所在位置 | 正确 |
| UC-002-05 | 1.进入考试管理页面  2.在考试列表中点击某一场考试的名称  3.进入考试修改页面后修改某些参数  4.点击保存 | 1.跳转到考试修改页面  2.系统提示修改成功  3.系统自动跳转到考试列表页面 | 1.跳转到考试修改页面  2.系统提示修改成功  3.系统自动跳转到考试列表页面 | 正确 |
| UC-002-06 | 1.进入考试管理页面  2.在考试列表中点击某一场考试的名称  3.进入考试修改页面后错误修改某些参数  4.点击保存 | 1.跳转到考试修改页面  2.系统提示必填字段填写错误或不能为空  3.光标焦点定位到错误位置 | 1.跳转到考试修改页面  2.系统提示必填字段填写错误或不能为空  3.光标焦点定位到错误位置 | 正确 |
| UC-002-07 | 1.进入考试管理页面  2.点击删除 | 系统提示请选择需要删除的项 | 系统提示请选择需要删除的项 | 正确 |
| UC-002-08 | 1.进入考试管理页面  2.点击考试列表中几个考试前的checkbox框  3.点击删除 | 1.系统提示删除成功  2.刷新当前页面，删除的项不存在列表中 | 1.系统提示删除成功  2.刷新当前页面，删除的项不存在列表中 | 正确 |

3．“题库管理”测试

**表5-3 “题库管理”功能测试**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | UC-003 | | | |
| **功能描述** | 老师或管理员登陆系统后进行题目的添加、修改、删除 | | | |
| **用例目的** | 对老师或管理员对题目管理进行功能测试。 | | | |
| **用例前提** | 1.老师或管理员通过系统身份验证登录系统  2. 进入题目管理页面 | | | |
| **子用例编号** | **输入/动作** | **期望的输出/响应** | **实际输出** | **状态** |
| UC-003-01 | 1.进入题目管理页面  2.点击添加题目  3.填写题目相关参数  4.导入题目测试数据  5.点击保存 | 1.系统提示题目添加成功。  2.跳转到题目列表页面，添加的题目已存在于列表中 | 1.系统提示题目添加成功。  2.跳转到题目列表页面，添加的题目已存在于列表中 | 正确 |
| UC-003-02 | 1.进入题目管理页面  2.点击添加题目  3.填写题目相关参数  4.导入题目测试数据  5.点击保存并添加另一个 | 1.系统提示题目添加成功  2.系统自动清空表单数据以便继续添加题目。 | 1.系统提示题目添加成功  2.系统自动清空表单数据以便继续添加题目。 | 正确 |
| UC-003-03 | 1.进入题目管理页面  2.点击添加题目  3.填写题目相关参数  4.导入题目测试数据  5.点击保存并继续编辑 | 1.系统提示保存成功  2.系统停留在当前页面，并且表单数据为清空  3.系统向服务端提交表单数据 | 1.系统提示保存成功  2.系统停留在当前页面，并且表单数据为清空  3.系统向服务端提交表单数据 | 正确 |
| UC-003-04 | 1.进入题目管理页面  2.点击添加题目  3.填写错误或未填关键参数  4.点击保存并继续编辑或者保存 | 1.系统提示必填字段不能为空  2.光标焦点自动定位到未填写或填写错误所在位置 | 1.系统提示必填字段不能为空  2.光标焦点自动定位到未填写或填写错误所在位置 | 正确 |
| UC-003-05 | 1.进入题目管理页面  2.在题目列表中点击某一场题目的名称  3.进入题目修改页面后修改某些参数  4.点击保存 | 1.跳转到题目修改页面  2.系统提示修改成功  3.系统自动跳转到题目列表页面 | 1.跳转到题目修改页面  2.系统提示修改成功  3.系统自动跳转到题目列表页面 | 正确 |
| UC-003-06 | 1.进入题目管理页面  2.在题目列表中点击某一场题目的名称  3.进入题目修改页面后错误修改某些参数  4.点击保存 | 1.跳转到题目修改页面  2.系统提示必填字段填写错误或不能为空  3.光标焦点定位到错误位置 | 1.跳转到题目修改页面  2.系统提示必填字段填写错误或不能为空  3.光标焦点定位到错误位置 | 正确 |
| UC-003-07 | 1.进入题目管理页面  2.点击删除 | 系统提示请选择需要删除的项 | 系统提示请选择需要删除的项 | 正确 |
| UC-003-08 | 1.进入题目管理页面  2.点击题目列表中几个题目前的checkbox框  3.点击删除 | 1.系统提示删除成功  2.刷新当前页面，删除的项不存在列表中 | 1.系统提示删除成功  2.刷新当前页面，删除的项不存在列表中 | 正确 |

4．“考场管理”测试

**表5-4 “考场管理”功能测试**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | UC-004 | | | |
| **功能描述** | 老师或管理员登陆系统后进行考场的情况实时了解 | | | |
| **用例目的** | 对老师或管理员对考场管理进行功能测试。 | | | |
| **用例前提** | 1.老师或管理员通过系统身份验证登录系统  2. 进入考场管理页面  3. 考试正在进行中 | | | |
| **子用例编号** | **输入/动作** | **期望的输出/响应** | **实际输出** | **状态** |
| UC-004-01 | 1.进入考场管理页面  2.点击某场正在进行中的考试  3.默认进入本场考试题目列表页面  4.点击某个题目查看 | 1.系统跳转到本场考试题目列表页面  2.题目列表显示正确  3.题目详情显示正确 | 1.系统跳转到本场考试题目列表页面  2.题目列表显示正确  3.题目详情显示正确 | 正确 |
| UC-004-02 | 1.进入考场管理页面  2.点击某场正在进行中的考试  3.默认进入本场考试题目列表页面  4.点击我的提交 | 1.页面跳转成功  2.正确显示自己所提交过的代码 | 1.页面跳转成功  2.正确显示自己所提交过的代码 | 正确 |
| UC-004-03 | 1.进入考场管理页面  2.点击某场正在进行中的考试  3.默认进入本场考试题目列表页面  4.点击所有提交  5.点击某个学生提交代码查看 | 1.页面跳转成功  2.显示本场考试所有学生提交的代码提交状态  3.显示选中学生的代码详情 | 1.页面跳转成功  2.显示本场考试所有学生提交的代码提交状态  3.显示选中学生的代码详情 | 正确 |
| UC-004-04 | 1.进入考场管理页面  2.点击某场正在进行中的考试  3.默认进入本场考试题目列表页面  4.点击排名 | 1.页面跳转成功  2.显示本场考试的学生排名 | 1.页面跳转成功  2.显示本场考试的学生排名 | 正确 |

5．“用户管理”测试

**表5-5 “用户管理”功能测试**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | UC-005 | | | |
| **功能描述** | 老师或管理员登陆系统后进行用户的信息的添加、修改、删除、解锁等 | | | |
| **用例目的** | 对老师或管理员对用户管理进行功能测试。 | | | |
| **用例前提** | 1.老师或管理员通过系统身份验证登录系统 | | | |
| **子用例编号** | **输入/动作** | **期望的输出/响应** | **实际输出** | **状态** |
| UC-005-01 | 1.进入用户管理页面  2.点击添加  3.在弹出的模态框中填写学生信息  4.点击保存 | 1.弹出模态框  2.提示用户添加成功  3.模态框消失  4.学生信息出现在列表中 | 1.弹出模态框  2.提示用户添加成功  3.模态框消失  4.学生信息出现在列表中 | 正确 |
| UC-005-02 | 1.进入用户管理页面  2.点击批量添加  3.弹出文件上传模态框  4.上传学生名单文件  5.点击保存 | 1.弹出模态框  2.文件上传成功  3.模态框消失，提示添加用户成功  4.学生信息出现在列表中 | 1.弹出模态框  2.文件上传成功  3.模态框消失，提示添加用户成功  4.学生信息出现在列表中 | 正确 |
| UC-005-03 | 1.进入用户管理页面  2.点击删除 | 系统提示选择需要删除的项 | 系统提示选择需要删除的项 | 正确 |
| UC-005-04 | 1.进入用户管理页面  2.点击列表中某几个学生信息前的多选框  3.点击删除 | 系统提示删除成功 | 系统提示删除成功 | 正确 |

6．“系统管理”测试

**表5-6 “系统管理”功能测试**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | UC-006 | | | |
| **功能描述** | 老师或管理员登陆系统后设置系统参数、系统公告等 | | | |
| **用例目的** | 对老师或管理员对系统管理进行功能测试。 | | | |
| **用例前提** | 1.老师或管理员通过系统身份验证登录系统 | | | |
| **子用例编号** | **输入/动作** | **期望的输出/响应** | **实际输出** | **状态** |
| UC-006-01 | 1.进入系统管理页面  2.修改系统名称、页脚信息、FAQS  3.点击保存修改数据 | 1.系统提示修改成功  2.停留在当前页面 | 1.系统提示修改成功  2.停留在当前页面 | 正确 |
| UC-006-02 | 1.进入系统管理页面  2.清空表单或错误填写表单  3.点击保存修改数据 | 系统提示关键字段必须填写，并将焦点定位到有误的位置 | 系统提示关键字段必须填写，并将焦点定位到有误的位置 | 正确 |

7．“学生考试”测试

**表5-7 “学生考试”功能测试**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用例编号** | UC-007 | | | |
| **功能描述** | 学生根据账号密码登陆系统参加考试，答题并提交代码 | | | |
| **用例目的** | 对学生参加考试的整个流程进行功能测试 | | | |
| **用例前提** | 1.学生登录系统并通过系统身份验证  2.考试正在进行中 | | | |
| **子用例编号** | **输入/动作** | **期望的输出/响应** | **实际输出** | **状态** |
| UC-007-01 | 1.学生登陆后进入到考试列表页面  2.选择进行中的考试  3.查看题目，并编写代码  4.选择编译器，提交代码 | 1.代码提交成功  2.页面跳转到提交状态显示页面  3.提交的代码状态显示在表格中，并能够看到题目是否通过 | 1.代码提交成功  2.页面跳转到提交状态显示页面  3.提交的代码状态显示在表格中，并能够看到题目是否通过 | 正确 |
| UC-007-02 | 1.上一个用例之后在提交代码状态显示页面  2.找到自己提交的代码状态，点击编译器名称 | 页面跳转到代码详情页面 | 页面跳转到代码详情页面 | 正确 |

### 5.2.2 非功能测试

本项测试主要包括对系统的界面测试，压力测试，可靠性能测试等。界面测试通常是为浏览器兼容性测试、色调、布局测试等；压力测试主要看系统能够是否满足了相应时间，高并发下的是否提供稳定的服务等，本次测试响应时间主要通过各浏览器的开发工具测试可靠性测试主要看系统能长时间运行过程中发生概率的概率。详细数据汇总如下表：

**表5-8 非功能性测试报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 兼容性 | 静态资源平均加载时间（ms） | 最快相应时间（ms） | 最长响应时间(ms) | 可靠性能 |
| 分析 | Google Chrome，Mozilla Firefox，Internet Explorer 9 | 1324 | 620 | 3042 | 3% |

## 5.3 本章小结

本章首先介绍了系统测试原则和本次测试的系统环境，从功能测试和非功能两个方面对系统进行测试，在系统功能测试部分，选取系统主要功能点进行测试，并给出功能的预期结果和实际结果。在非功能测试部分对界面测试、压力测试、可靠性测试总结汇总分析。经测试，本系统满足功能需求，能够稳定运行。

# 结论

本文首先介绍了课题的研究背景，指出了程序设计能力认证系统开发的必要性，说明了课题的研究意义及最总将达到的目标。接下来对系统的需求分析，明确了系统的功能和非功能目标，为后期的系统设计与实现做了准备。在对系统设计时，首先对系统设计策略进行描述，然后对功能结构和界面设计进行介绍，接着是对数据库设计的介绍。在系统详细设计与实现章节中，主要介绍了本系统技术难点及解决方案和分模块详细介绍了系统功能。最后对系统进行了测试，从测试结果来看，本系统基本满足系统需求，其能够稳定的运行。

在本系统实现过程中，也发现了系统存在一些不足。主要有：

（1）浏览器兼容问题。在开发过程中，系统页面主要针对chorme浏览器来开发的，由于较低版本的IE浏览器不能很好支持一些新的css属性，因此页面中圆角、阴影、倒影等在不同浏览器中显示的效果有些差别。所以进入系统时会提示使用chrome等高级浏览器，体验效果会更佳。

（2）系统角色问题。本系统为了角色管理方便，只设置了两种角色方案，即老师和学生，虽然本文提到了管理员，其实管理员权限和老师一样。但是可以将系统管理员角色提取出来，用于专门设置系统的参数，不涉及考试方面的功能。这样角色权限更加明确。

# 致谢

大学时光转眼就要结束了，思念的时光既充实又美好。在完成毕业论文的过程中，许多人给予了我热心的帮助和指导，在此，我由衷的向他们表示最诚挚的感谢。

首先，感谢我的指导老师杨春明老师。在大学期间，无论是实验室项目的完成还是学位论文的撰写中，杨老师都给予我许多的帮助。从选题到课题的研发和实现，杨老师悉心指导，论文的文章结构、知识体系、术语应用等都细心检查和修改。杨老师渊博的学识、严谨的学风和平易近人的高尚品德是我受益匪浅。在此，特向杨老师表示我最崇高的谢意与敬意。

同时，感谢实验室以及班上的全体同学，感谢你们陪我一起学习，一起奋斗，使我的大学生活充实而快乐，和你们在一起奋斗的时光是我宝贵的人生财富。

感谢舍友，四年的朝夕相处让我们更加了解彼此，让我大学的生活丰富多彩，我不会忘记我们一起去过的远方，一起奋斗的夜晚，运动场上留下的汗水，这将是我难以遗忘的一段经历，愿我们毕业后工作都顺利。

衷心感谢我的父母和家人，他们对我无微不至的关怀和鼓励是我永远的精神支持，他们对我的爱和期盼是我在求学路上最强大的精神动力。最后，感谢所有参加论文评审和答辩的老师在百忙之中抽出宝贵的时间审阅本论文。