学号：2013040131083

密级：XXXXXXXX



**本科毕业论文（设计）**

**基于Springboot框架知识竞赛系统**

院（系）名称：计算机科学学院

专业名称 ：计算机科学与技术

学生姓名 ：何爽

指导教师 ：胡平芳

2017年 3月30日

**BACHELOR'S DEGREE THESIS OF WUHAN DONGHU UNIVERSITY**

**Knowledge competition system based on spring boot framework**

**Candidate：He Shuang**

**Supervisor：Hu Ping Fang**

Mar. 30st, 2017

**郑重声明**

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

论文作者签名：

日期： 年 月 日

基于springboot框架知识竞赛系统

摘　要

任何一个公司都有其深厚的文化内涵和文化底蕴，企业通过举行企业文化知识竞赛，可以让员工较为深刻的了解企业文化底蕴，竟然让员工热爱企业热爱工作。设计企业知识竞赛系统，对企业知识文化和对企业员工的熏陶有重要的意义。

本系统运用的是B/S 结构模型，采用了AJAX，html5等前端网页设计技术和springboot后台框架技术，本系统的主要特色是能够自主的进行活动设置，如题目的导入（excel）、难易程度配比、答题时间、快速回显答案、题目随机分配等；系统运用了随机算法，参与人员每次登录系统参与竞赛都会重新生成一套题目随机的分配给答题者，保证了参与人员不会出现重复的题目，能够相对公平的进行答题提高效率；每完成一个关卡的答题后系统会根据参与者所得分进行一次抽奖与排名，如果你全部通关则会得到通关大奖。此一系列的逻辑都由后台系统自动完成。不需要人为参与，得奖者将会被记录，以便于赛后进行奖品发放；本系统可与大部分的系统完美衔接实现单系统多用的好处。所设计基于spring boot框架知识竞赛系统，经过测试与实际的运作效果良好。

关键词：知识竞赛系统，spring boot，HTML5，B/S，AJAX

**Knowledge competition system based on springboot framework**

**ABSTRACT**

Each company has its own profound cultural connotations and cultural deposits. Through the company cultural knowledge contest, employees can get a further acquaintance of the company culture Meanwhile, the company can popularize its related work to its employees, arouse employee's enthusiasm of work, and then increase their work efficiency Design a company cultural knowledge contest system has a vital significance not only to the company culture but also to the employees.

This system uses the B / S structure model, adopt AJAX, html5 and other front-end web design technology and springboot background framework technology. The main characteristic of this system is be able to set up activities independently, such as the input of topic, comparison of difficulty level, time for answering, echo answer quickly, random assignment of questions and so on. This system adopted the random and filtration algorithm. Every time the participant participates in the contest, the system will create a different set of exercise randomly to the answer, which guaranteed the fairness and efficiency of the contest. Once the answer completes a section of the contest, the system will draw a lottery and ranking according to their score, the top prize is available for answers who get through all sections. This series of logic are automatically completed by the background system. The winner's name will be recorded, so as to the award extension. The system can be connected with most of the system and realize the advantages of single system with multi-purpose. The knowledge contest system based on the spring boot framework was tested to be well-operated.

**Key words**: Knowledge Contest；spring boot；HTML5；B/S；AJAX

**目　录**

[1　绪　论 6](#_Toc480322636)

[1.1　课题的来源、意义和目标 6](#_Toc480322637)

[1.1.1国内外研究现状与发展趋势 6](#_Toc480322638)

[1.2　主要完成的任务与设计特色 7](#_Toc480322639)

[1.3　论文组织结构 7](#_Toc480322640)

[2　相关技术 8](#_Toc480322641)

[2.1　C/S模式与B/S模式 8](#_Toc480322642)

[2.1.1 C/S模式 8](#_Toc480322643)

[2.1.2 B/S模式 8](#_Toc480322644)

[2.1.3 两种模式的特点 8](#_Toc480322645)

[2.2浏览器端的JavaScript 9](#_Toc480322646)

[2.3 前端框架JQuery 9](#_Toc480322647)

[2.4 AJAX技术 10](#_Toc480322648)

[2.5 Oracle数据库 10](#_Toc480322649)

[2.6 三层架构设计 11](#_Toc480322650)

[2.7 Springboot框架 11](#_Toc480322651)

[2.8 本章小结 12](#_Toc480322652)

[3 需求建模 12](#_Toc480322653)

[3.1业务功能目标 12](#_Toc480322654)

[3.1.1系统概述 12](#_Toc480322655)

[3.1.2此系统主要需求目标 13](#_Toc480322656)

[3.2 非功能性需求 14](#_Toc480322657)

[3.2.1可用性 14](#_Toc480322658)

[3.2.2 可靠性 14](#_Toc480322659)

[3.2.3性能需求 14](#_Toc480322660)

[3.2.4 兼容性需求 14](#_Toc480322661)

[3.2.5 可扩展性需求 15](#_Toc480322662)

[3.2.6 安全性 15](#_Toc480322663)

[3.3 业务流程描述 15](#_Toc480322664)

[3.4实体关系 16](#_Toc480322665)

[3.5 本章小结 16](#_Toc480322666)

[4 系统设计 16](#_Toc480322667)

[4.1系统目录规划 16](#_Toc480322668)

[4.1.1详细目录 16](#_Toc480322669)

[4.1.2 文件命名规范 17](#_Toc480322670)

[4.2 技术架构 17](#_Toc480322671)

[4.3 数据库设计 19](#_Toc480322672)

[4.2.1 数据库设计图 19](#_Toc480322673)

[4.2.2 数据库表设计 20](#_Toc480322674)

[4.4 本章小结 27](#_Toc480322675)

[5 详细设计 27](#_Toc480322676)

[5.1 活动创建模块 28](#_Toc480322677)

[5.1.1 类设计 28](#_Toc480322678)

[5.1.2 流程设计 28](#_Toc480322679)

[5.2导入模块 29](#_Toc480322680)

[5.2.1 类设计 29](#_Toc480322681)

[5.2.2 流程设计 31](#_Toc480322682)

[5.3答题模块 31](#_Toc480322683)

[5.3.1 类设计 31](#_Toc480322684)

[5.3.2流程设计 32](#_Toc480322685)

[5.4排名模块 33](#_Toc480322686)

[5.4.1类设计 33](#_Toc480322687)

[5.4.2流程设计 34](#_Toc480322688)

[5.5逻辑设计 34](#_Toc480322689)

[5.5.1 答题系统安全验证 34](#_Toc480322690)

[5.5.2 管理员安全验证 34](#_Toc480322691)

[5.5.3 答题倒计时 34](#_Toc480322692)

[5.5.4 抽奖设计 35](#_Toc480322693)

[5.6 本章小结 35](#_Toc480322694)

[6 系统部署与应用实例 35](#_Toc480322695)

[6.1 开发工具以及部署环境 35](#_Toc480322696)

[6.2 系统运行界面 36](#_Toc480322697)

[6.3 本章小结 43](#_Toc480322698)

[7 总结与展望 44](#_Toc480322699)

[7.1 总结 44](#_Toc480322700)

[7.2 展望 44](#_Toc480322701)

[参考文献 45](#_Toc480322702)

[致　谢 46](#_Toc480322703)

[附 录 47](#_Toc480322704)

[1.1 47](#_Toc480322705)

[1.2 47](#_Toc480322706)

[1.3 48](#_Toc480322707)

[1.4 48](#_Toc480322708)

# 1　绪　论

## 1.1　课题的来源、意义和目标

从上个世纪到现在，个人的计算机与互联网日益普及开来，用于信息创造和管理的各种自动化的系统层出不穷，在市场上日渐扩大，进入新世纪之后，国内企业的信息化，数字化工作，相继进入了新的繁荣发展阶段。越来越多的大学和公司都意识到，在这样一个历史时期机遇和挑战并存，信息数字化是企业的核心竞争力和基本战略。

随着时代的发展计算机应用竞赛也越来越普及，笔试考试比赛已经跟不上当前时代的发展与前行，笔试需要人工的填写，人工的更改；人力资源得不到合理的利用，复印过的试题难以再利用，加大了对环境的破坏，所以一个线上的答题与管理系统的问世迫在眉睫。自动化相比于人力操作完全不在一个效率上，资源相对来说得到完美的利用，而且传统的方式存在各种不定因素缺乏应有的公正，因为批改人不可能百分之一百的能够正确的给出分数总有一些疏忽，更有甚者还有可能存在人为的感情因素，种种不恰当的元素不能客观的反映参赛选手的水平，也就失去了比赛的价值。

眼观国外，各种竞赛管理系统层出不穷不乏优秀的系统，然而在国内与此类型相同的可移植性的竞赛系统少之又少。就目前情况来看我国还处在雏形阶段还没有正真的成型还需要有人不断地改造与完善。因此，在中国现阶段一款应用于知识竞赛管理系统的创建，也就成为了软件领域的重要议题。

### 1.1.1国内外研究现状与发展趋势

现在在海外在线考试系统（比赛系统试验系统的分类）的研究有很多，网上考试系统也很多，比如微软的认证考试，属于美国教育中的研究生入学考试系统，美国软件行业的CCP认定考试，认证考试思科等。在我国比如计算机的等级考试，教育部的蓝桥杯考试，还有各省市地区相关行业认证考试等。

在国内，互联网相关的技术在不断地更新迭代，知识竞赛系统作为考试系统中的一种已经越来越受到国人们的重视。如:2011年全国百家官方网站成功的举办中国社会主义法制知识竞赛，水资源部主办的中央文件知识竞赛，全国水利知识安全网络竞争等。这些成功举办的比赛，加强宣传的效果，拓宽了宣传路线，获得了非常广泛的好评。

## 1.2　主要完成的任务与设计特色

在项目开发的过程中坐着主要负责了一下内容：

首先，在项目开始的前期过程中，作者对整个项目进行了整体的需求分析与后台系统的整体构架进行分析，并且构建了一个详细整体列模型，与此同时相继完成了系统框架架构设计即springboot框架，以及前后台系统中的数据库设计与搭建，为后期系统能够成功运行奠定的稳固的开发基础；

其次，设计了前后台的界面展示模型，同时把系统中涉及的活动创建模块，用户管理，时间管理，题目导入管理，题目分配管理，得分统计，抽奖模块等的代码编辑构建工作。

此外，在项目的后期工作中，作者负责了测试部分并书写规整的测试报告，对此系统进行比较详细的测试，修复了比较大的bug，并对前端框架以及页面进行了优化设计改善。

本系统的主要特色是:框架搭建运用了时下最流行的springboot框架此框架集成了springmvc+hibernate框架的优秀特性，前台页面运用了HTML5网页编辑技术。运行平台是基于Nginx转发机制实现。

## 1.3　论文组织结构

本论文总共分为 7个章节:选题背景、技术论述、系统架构分析、系统架构设计、模块分析实现、系统架构实施、论文总结展望等方面对基于springboot知识竞赛系统进行详细的论述。

第一章：首先阐述了企业文化对于企业本身的意义，然后介绍了考试系统现在的发展方向与情况，说明了在线知识竞赛在目前是一种趋势。最后，介绍了本系统相对而言比较优秀的创意部分。

第二章：介绍项目中使用的比较前沿的科学方法，以及相关技术在当下的现状分析，还有本系统开发过程中的应用状况。

第三章：选择了系统中相对于整体来说比较关键的模块进行详细的需求模型建立，并对系统的需求进行更加深入的分析；与此同时构建较为详尽的模型用例，深入理解内在的需求以及其他不是功能的需求比如性能优化。

第四章：对整个系统设计，说明了系统总体架构以及系统的架构原理，分别对架构体系、架构技术、架构的静态整体等进行微观展现，然后画图展示功能逻辑，并设计构建了后台的数据库。

第五章：以功能块为依据单位，对功能块进行设计分析，首先用形象的流程图对业务流水和算法基础进行展示，然后到具体的功能，设计出伪代码类并进行改造变成适用于系统的类。

第六章：主要介绍了本系统开发的时候使用了哪些工具和项目运行的环境，然后对已经运行的实际界面截图展示，阐述本系统在实际运用中的具体价值体现。

第七章：总结该系统的不足以及比较优秀的部分，然后对系统以后的发展进行了展望总结。

# 2　相关技术

## 2.1　C/S模式与B/S模式

### 2.1.1 C/S模式

C/S模式是一层是在客户端PC装的应用程序，一层是在服务器上装服务软件的两层体系结构。当客户端那边发送请求的时候，服务器会接收到数据然后按照指定好的业务处理发来的请求，然后将处理的结果返回到客户端。

### 2.1.2 B/S模式

在B/S中，用户这边启动浏览器，浏览器将会用超级文本向远程的服务器发起访问数据请求，服务器拿到用户的请求转变成sql语句然后发给sql系统服务，sql系统服务会先验证这个语句是否符合规则然后处理请求，随即将处理后的数据发给服务器，服务器会再次转换变成文本发送给客户，客户这边拿到结果后处理以一种比较的直观的方式变现出来。

### 2.1.3 两种模式的特点

C/S模式：

互动性良好是这种模式一种特别的优点。在此模式中客户应用端有一个完整的处理方案，在报错提示处理以及帮助方面都非常的了不起，而且在小程序中可以比较轻松的转换。由于开发的需要，所以界面要相对的美观，也要满足的用户的特别需求。

但是此种模式不能灵活的变通，如果要改变某一个需求的时候就需要全部重新开发，这样就相对的提高了成本与维护难度，而且如果业务要拓展也有相当大的难度。

B/S模式：

对于大型的系统如果要维护起来成本开销是巨大的，然而在此种框架模式下开发的项目可以很大的程度上避免了成本开销，而且这个模式对用户的电脑的配置不会有特别的要求这样就为客户节省了资源提供了便利。

这种模式都会依附浏览器开发的，所以这些界面基本上都差不多从而降低了前台界面开发的成本。而且这个模式的系统可扩展性特别的强大可以从一个十几个人的小组变成几千万的大组。

## 2.2浏览器端的JavaScript

JS目前是比较主流的前端脚本语言[1]。它的语言规则非常的简单易懂，相对于其他的语言它学习起来要快很多，而且使用性特别的强大，所以到目前为止很多的网页前端基本都用到了它，比如验证，数据接收检查，以及参数处理，cookie等功能的开发实现[2]。JS又是轻便级别的程序语言[3]，编码可以在任何的HTML网页中运用，而且可以被大部分的浏览器识别运行[4]。

因为JS比较强大所以在本系统中大量的使用，其主要负责了将静态页面转换为动态的网页，网页和用户可以实现实时交互动作；也可以实现表单的数据在送到数据库之前进行实时验证大大提高了页面的执行效率，同时还可以生成大量的cookie方便与服务器的快捷通信。

## 2.3 前端框架JQuery

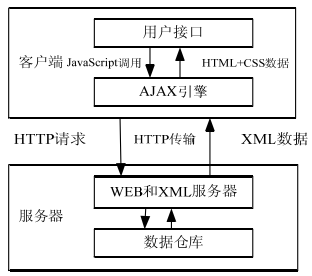
JQuery是一个轻量级的JS库，涵盖了HTML中大部分的功能比如元素的选取、元素的操作、元素的事件响应、CSS的布局操作、DOM的遍历操作和修改操作以及许多其他功能[5]，而且可以非常方便的为网页给予AJAX互动操作[6]， JQuery的理念是“less code，more work”[7]，它包含了很多前端需要的插件，与其他的JS库相比来说，它的代码更加精炼、语法通俗、操作起来更加容易、延展性很强大，而且文档解释也比其他的要更加明确 [8]。

此系统中在前端开发的过程中，很多JQuery插件被运用到了，一般用来处理事件和动画等方便，而且很有效的实现AJAX互联，在页面中添加一个<script>标签并且引用jquery.js这个库，其中基本包含了所有的即将使用到的各种函数，非常方便的对HTML元素进行操作[9]。

## 2.4 AJAX技术

AJAX不是全新的编程设计语言，只是一种比较前言的技术而已 [10]。通过利用这项技术，可以使页面更加快捷方便的交互。浏览器与服务器端相互传输的数据都是通过使用AJAX技术实现的[11]。在网页中与服务器进行数据之间的传输，不是请求变更的时候重新生成一个全新的网页，AJAX可使网页响应更迅速[12]。

传统的Web程序直接把数据信息提交到Web服务器，在Web服务器将会把发过来的数据经过特殊的处理后，然后把处理后的网页再次发送给用户[13]。只要用户有提交，服务器就会处理一下然后返回一张全新的网页，因此传统的Web程序运行起来特别的慢，而且逐渐变得不友好[14]。AJAX技术就不一样了, 它可以使Web程序不需要处理重新加载一次网页，直接交互数据[15]。通过向后台发出HttpRequest，然后使用JS技术对拿回的数据进行相应的处理然而这也仅仅是对网页很微小的一部分进行修改[16]。AJAX是一个独立的技术，它被所有浏览器所支持，是跨越浏览器和操作平台的一种高新技术[17]（如图2.1所示）。

图2.1 AJAX工作原理图

## 2.5 Oracle数据库

Oracle Database，又名Oracle RDBMS，或简称Oracle。隶属于甲骨文公司的一款数据库管理系统。劳伦斯·埃里森和他的朋友，之前的同事Bob Miner和Ed Oates在1977年创建了软件开发实验室咨询公司（SDL，Software Development Laboratories）。SDL开发了Oracle软件的最初版本。Oracle的名称来自于埃里森在Ampex工作时参加的一个由中央情报局创建的项目的代码名称。

## 2.6 三层架构设计

三层结构就是客户端与数据库之间加入了一个组件层。三层只是我们所说的业务逻辑的三个层面，只是把这个三个层面放在了同一个机器上来处理罢了。

在此体系中一般都是将主要的业务操作，数据交互，验证等都放在了组件层面进行业务处理的，为了数据的安全以及操作的方便客户端是不会与数据库发生直接的交互操作的，而是要与组件层面发生特殊的连接，在经由组件层与后台数据库进行直接交互。

三层结构具体表现为：

UI(表现层):即是用户直观的看到的图像，将数据以一种特定的布局放在界面上供用户参考使用。

BLL:(业务逻辑层):UI层和DAL连接纽带。业务功能主要包括数据验证，数据计算等。

DAL:(数据访问层):在这个层面上主要是与数据库打交道，对数据进行简单的增加删除查询更改操作。将查询的数据返回出来，待相应的层面处理完成后再存到数据库中（如图2.2所示）。

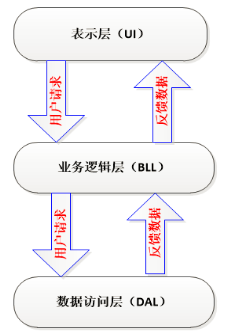


图2.2 三层结构模

## 2.7 Springboot框架

Spring Boot是简化了基于Spring的应用开发，只需要"run"就能创建一个独立的，产品级别的Spring应用，多数Spring Boot应用只需要很少的Spring配置就可以跑起来一个项目，spring boot有很多spring所不具有的优点，此框架自身带有一个tomcat，不要再去部署WAR文件，它可以使maven项目也可以是gradle项目，它能够自动的去配置spring所需要的配置文件，无需手动编写xml配置文件。

鉴于spring boot框架有很好的运行性能，此套系统的后台框架即是基于此框架来进行开发的，这也使得后期的编码工作效率较以往相比提高了很多。

## 2.8 本章小结

本章中，详细的介绍了在开发实现的过程中所用到的主要的主流技术，包括在目前的环境中技术的成熟性与施行进行了比较详尽的论述分析，以及在系统开发的过程中使用实例；系统能够成功的开发并且能够流畅的运行，较大的程度上取决于前期准备工作，只有深入的分析了系统才能保证后期的开发工作能够顺利的进行下去。

3 需求建模

## 3.1业务功能目标

### 3.1.1系统概述

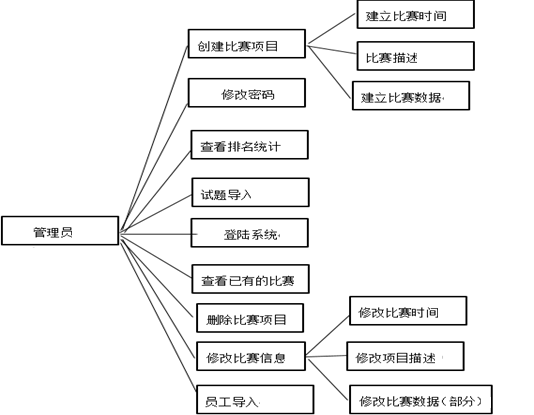
从使用的角色上来说，主要有三种角色，即竞赛的发布者，比赛参与者，以及后台管理者。比赛的发布者与后台管理者在此系统中是同一个身份，他所具有的权限有比赛发布、公告查看修改、排名查看、统计等一系列后台功能。admin可以导入试题也可以删除，也可更改用户的信息等附属功能（如图3.1所示）。

图3.1 管理员权限

### 3.1.2此系统主要需求目标

1. 根据企业定期或非定期的活动举办，员工通过知识微问答的方式进行活动参与，以游戏答题闯关的形式实现员工对知识的学习，增加工作乐趣。员工闯关成功开启下一关，并积累相应的星数。达到一定的条件，可以开启大关卡，并获得惊喜的抽奖机会。

2. 实现活动的有效期控制和每天的答题时间点控制。

每期活动均有活动的有效期限，活动有效期内，员工可以实现当日工作时间段内闯关答题（工作时间为每日8:30AM 至17:30PM）这个活动规定时间可以手动后台进行配置。

3. 实现试题库模板导入。

每个试题均带有一些不同的属性，如试题的分类（IT类，财务类，采购类，公司规章制度类等），试题知识点（如IT分类下的信息安全，知识管理系统KM），试题的难度（难，易，中）等，试题号码为试题的唯一识别码。试题与活动并无相应的逻辑关联性，试题可以通过后台管理员的配置选入不同的活动中，试题库中的试题均可以被不同活动重复利用。

4. 实现答题系统后台的可配置管理。

根据活动主题不同，实现活动的管理，灵活的实现后台相关参数的配置以对应活动的不同需求。可配置的后台内容包括：活动的有效期配置，每次活动对应的试题数量，总关数和关卡数的配置，每关的试题的难易度题量个数的配置，及对应分值的配置，不同关卡抽奖奖品内容的配置和中奖（或奖品个数）的配置。

5. 员工初始信息的后台导入。

员工信息的导入，包括所属组织，中心，部门等必要信息。 员工信息存在变动的可能性，员工的工号为员工的唯一性标识符。

6. 员工答题的互动性交流。可对个人微排行在前20名的员工点赞和对中心排名点赞。

## 3.2 非功能性需求

### 3.2.1可用性

1．系统的运行稳定状态需要保持在99%。

2．系统需要保持7\*24稳定运作 。

### 3.2.2 可靠性

可靠性是设计中需要考虑的一个主要原则，为了保证系统连续安全可靠地运行，所以在产品结构设计中选用高可靠性产品，合理设计架构，尽可能利用成熟技术，最大程度使系统正常运行。

### 3.2.3性能需求

在网络正常情况下，系统前端界面访问响应时间需要平均少于3秒，后台管理操作需要平均少于5秒。移动端访问响应时间需要平均少于5秒。同时，系统支持约1000人以上的并发登录。

### 3.2.4 兼容性需求

后台使用JAVA开发，后台系统支持 Chrome、Safari、Firefox、IE9，IE10，IE11，360主流现代浏览器，App支持 iOS 7+ 或 Android 4+。

### 3.2.5 可扩展性需求

系统的设计目前包含前端用户展示，前后台接口，后台管理3个方面的内容，考虑满足未来业务可扩展、管理有效、快速响应的目标，发挥资源的使用效率。功能设计，遵守模块化、参数化原则，功能模块相对独立，通过模块的组合实现业务需求。未来的扩展可能包括同期多个活动功能的添加和系统的优化等，但后续优化和功能扩展暂不在本期范围内实现。

### 3.2.6 安全性

系统需要有SQL 注入、XSS、CSRF 的安全防护措施。且需要定制404、500等错误页面。不允许直接暴露服务器配置信息及服务器文件结构。

## 3.3 业务流程描述

此系统的业务主要流程有竞赛流程，活动生成流程，试题管理流程和比赛统计流程。具体的流程如下：

竞赛系统流程：参赛者可通过企业首页或者其他企业通过其他的方式接入本系统，因为本系统具有可移植性所以不具备直接注册进行答题的功能；参赛者通过企业提供的登录入口进入系统，系统会随机试题给参赛者，进入答题后系统立即开始倒计时，参赛者必须在规定的时间内完成答题，否则将取消答题返回首页，每完成一个单元，系统会自动判定是否进入下一个单元，如若成功进入下一个单元系统将弹出抽奖页面进行抽奖，每一个单元只有一次抽奖机会，机会不能累计，不抽视为放弃，完成全部单元即可获取通关大奖。

试题管理流程：管理员通过登录后台管理系统，可以通过Excel导入试题，可以对导入的试题进行增加、删减、更新等操作。

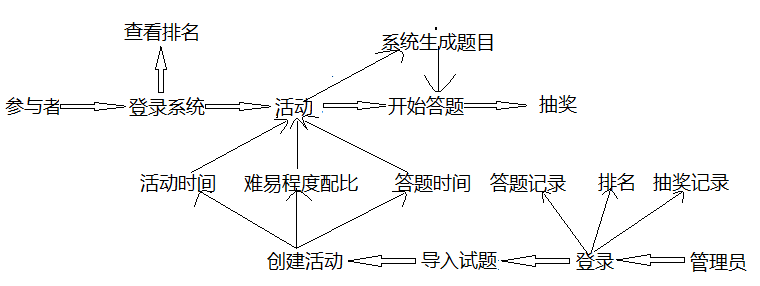
赛后统计流程：管理员通过登录后台管理系统可以看到参与人员的排名，以及积分情况。和获奖名单等条款（如图3.2所示）。

图3.2 业务流程图

## 3.4实体关系

通过分析需求，数据，流程，划分出实体数据对象[18]分别为“管理员对象”，“活动对象”，“参与者对象”，“试题对象”，“参与者成绩对象”。一个完整的竞赛系统肯定需要设置多个管理员账号，这样可以使管理后台更加方便快捷，管理账号包括账号以及登录密码。

试题对象包括编号，分类，时间，等属性。参与者实体有：名称、性别、姓名、所属组织、密码、身份证号码、电话号码、考试成绩等属性。当参与者完成比赛后可以看到排名情况，以及所属的组织或其他组织的排名情况。参与者的成绩对象包括用户，所属组织等。

## 3.5 本章小结

本章中是对需求进行分析，详细模型构建，各需求功能的明确，以及系统所需要的的性能优化、可行性、相对稳定特性、实际运用性、安全性等的非功能性的需求。多个需求的渐渐深入和明确，为其以后的整体设计打下了坚实基础，为竞赛系统顺利开发创造条件。

4 系统设计

## 4.1系统目录规划

### 4.1.1详细目录

一个知识竞赛系统，其中包括多种文件，比如程序，图片，数据等，这些都应该放在特定的模块当中，但是建立一个目录是有一个原则的，需要层次上清晰，分类要合理（如图4.1所示）。

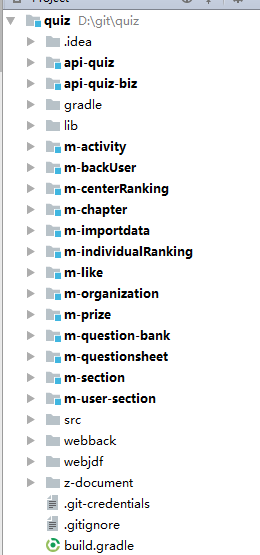


图4.1 系统模块图

### 4.1.2 文件命名规范

文件的命名规则遵循的是统一的编辑规范，一般都以英文翻译的中文作为文件的名称且都是以小驼峰的格式命名，然后前面在加上统一的命名前缀便于统一辨识（如图4.1所示）。

## 4.2 技术架构

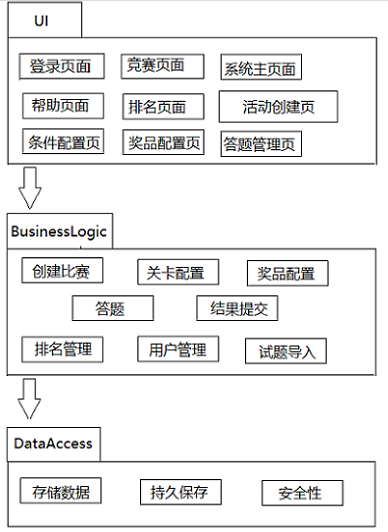
本系统采用了主流的三层模型，将不同的层、模块分别放在了不同的包下面，按照主流的规范进行命名，并且符合系统分离规范。分层逻辑架构（如图4.2所示）

图4.2 系统分层逻辑架构图

系统各层的主要组成部分及其功能如下:

顶层是交互层即界面表示层，主要有用户登陆、主界面、活动列表，以及帮助、排名几个主要模块，快捷的实现了与用户交互的操作，以及对数据的验证操作，并且将有效的实时数据传输到后台服务层。

中间是服务的逻辑，其主要任务是实现基本服务的逻辑功能，包括有：创建活动、答题、答案提交、排名，以及用户管理、帮助管理，答案回显等几大核心的模块，实现对前台传输过来的数据进行更加深入的加工操作，实现各种业务逻辑及其运算操作，构成服务器端的主要部分。

底层主要是数据库交互，实现增删查改等简单操作，对系统的活动、题目信息、难易度配比信息、用户提交信息、排名信息、帮助，以及统计数据等多项重要的数据进行存储，确保服务器能在任何时候被多人调用时能反馈出正确及时有效的数据，并且维护数据库的完整性以及正确性。

## 4.3 数据库设计

### 4.2.1 数据库设计图

系统数据库设计（如图4.3所示）。

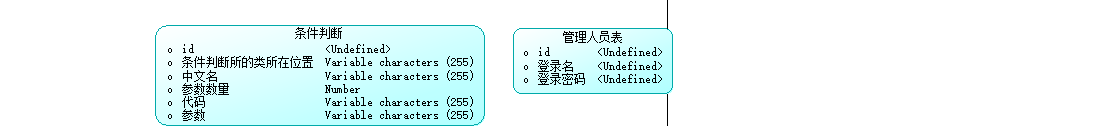
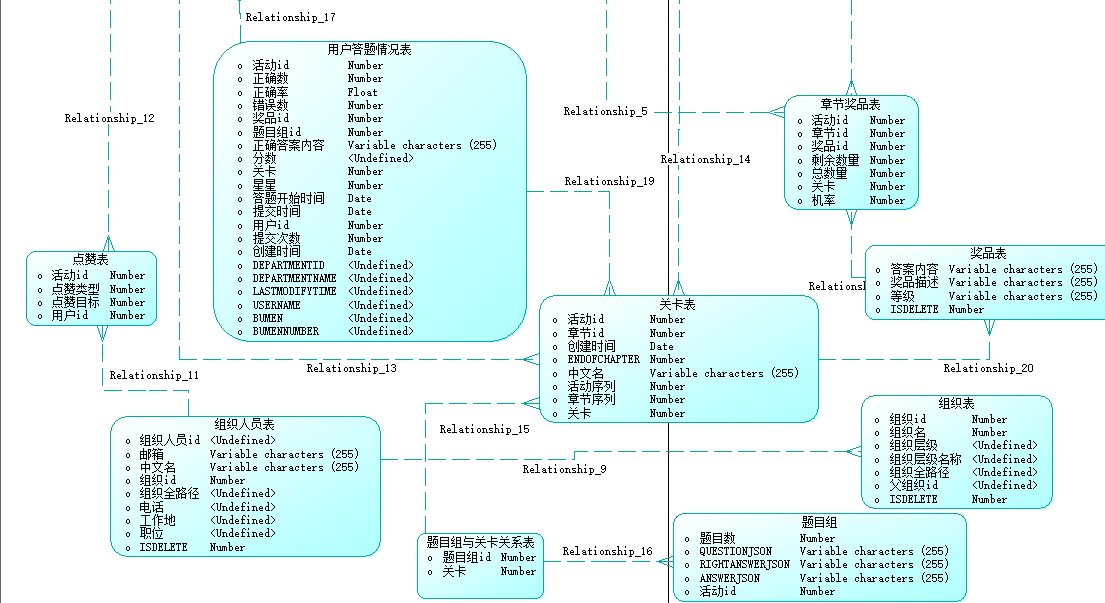
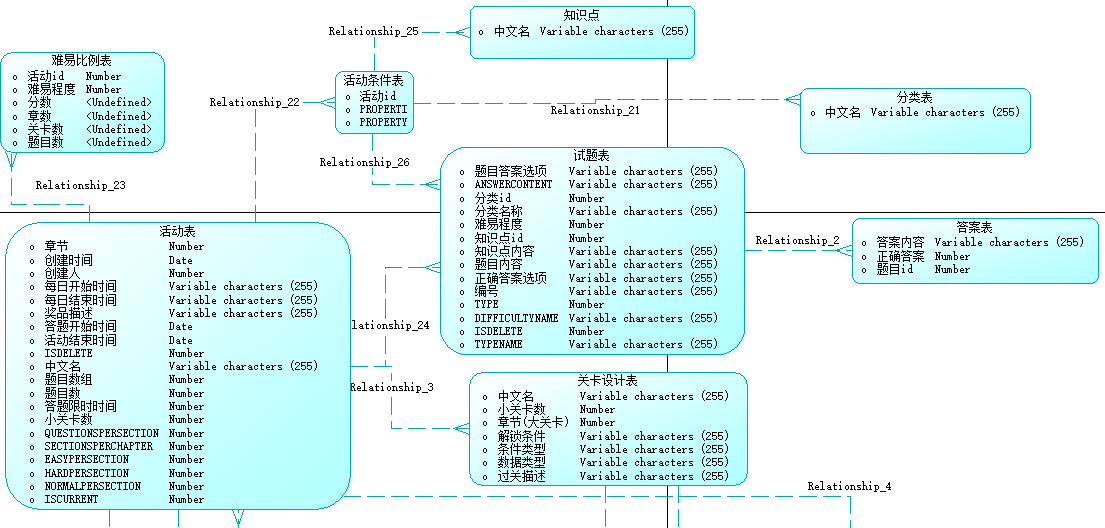


图4.3 系统数据库设计

从数据库的设计图可以看出“活动表”和“试题表”是两张核心表，其他多个表都引用了它的主键作为外键，活动生成后所对应的试题会存储在“试题表中”，每次只能开放一个活动，用户的答题提交结果都会存储在“用户答题情况表中”用户每天在规定的时间里可以重复作答任何一个关卡，答题闯关所获得积分达到一定的分数就可以进行抽奖，抽奖的概率存储在“章节奖品表中”。

### 4.2.2 数据库表设计

以下是所有的数据库表的关键字段的详细设计，顺序无前后排序。

活动表设计（如表4.1所示）。

表4.1 数据库表ACTIVITY的主要字段描述表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 活动Id | 主键，自动递增 |
| NAME | VARCHAR2(255) | 活动名称 | 非空 |
| DESCRIPTION | VARCHAR2(255) | 活动描述 | 可以为空 |
| STARTTIME | DATE | 活动开始时间 | 非空 |
| ENDTIME | DATE | 活动结束时间 | 非空 |
| DAILYSTART | VARCHAR2(255) | 当天开始时间 | 非空 |
| DAILYEND | VARCHAR2(255) | 当天结束时间 | 非空 |
| SECTIONLIMITTIME | NUMBER(10) | 答题限制时间 | 非空 |
| CHAPTERS | NUMBER(10) | 大关卡数 | 非空 |
| SECTIONS | NUMBER(10) | 总关数 | 非空 |
| QUESTIONS | NUMBER(10) | 总题目数 | 非空 |
| CREATETIME | DATE | 创建时间 | 当前时间 |
| ISDELETE | NUMBER(10) | 是否删除 | 0删除1删除 |
| SECTIONSPERCHAPTER | NUMBER(10) | 每关卡关数 | 非空 |
| QUESTIONSPERSECTION | NUMBER(10) | 没关问题数 | 非空 |
| EASYPERSECTION | NUMBER(10) | 每关简单题数 | 非空 |
| NORMALPERSECTION | NUMBER(10) | 每关中等题数 | 非空 |
| HARDPERSECTION | NUMBER(10) | 每关难题数 | 非空 |
| ISCURRENT | NUMBER(10) | 是否当前活动 | 1当前0非当前 |

存储答案信息的数据库表设计（如表4.2所示）。

表4.2 数据库表ANSWER的主要字段描述表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键Id | 自增 |
| CONTENT | VARCHAR(255) | 答案文本 | 文本 |
| CORRECT | NUMBER(10) | 是否正确答案 | 1是0不是 |
| QUESTIONID | NUMBER(10) | 问题Id | 外键 |

存储分类的数据库表设计（如表4.3所示）。

表4.3 数据库表CATEGORY的主要字段描述

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键Id | 主键，自增 |
| ISDELETE | NUMBER(10) | 是够删除 | 1删除0未删除 |
| NAME | VARCHAR(255) | 分类名称 | 非空 |

存储章节的数据库表设计（如表4.4所示）。

表4.4 数据库表CHAPTER的主要字段描述

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键id | 主键，自增 |
| ACTIVITYID | NUMBER(10) | 活动Id | 外键 |
| CONDITIONDATA | VARCHAR(255) | 过关条件数据 | 非空 |
| CONDITOONTYPE | VARCHAR(255) | 过关类型 | 非空 |
| CREATEDATE | DATE | 创建日期 | 非空 |
| PASSDECRIBE | VARCHAR(255) | 过关描述 | 非空 |
| SECTIONS | NUMBER(10) | 关卡数 | 非空 |
| SEQUENCECHAPTER | NUMBER(10) | 大关卡 | 非空 |
| UNLOCKCONDITION | VARCHAR(255) | 解锁条件 | 非空 |
| CHAPTERID | NUMBER(10) | 大关卡Id | 非空 |
| AVGSTAR | NUMBER(10) | 平均星 | 非空 |
| SUMSTAR | NUMBER(10) | 总星 | 非空 |
| SECRTION | NUMBER(10) | 关卡 | 非空 |
| LESSSTAR | NUMBER(10) | 最少星 | 非空 |

存储关卡的难易题目比的数据库表设计（如表4.5所示）。

表4.5 数据库表DIFFNOREAY的主要字段描述

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键Id | 主键，自增 |
| ACTIVITYID | NUMBER(10) | 活动Id | 外键 |
| EASYPERSECTION | NUMBER(10) | 每关简单题 | 非空 |
| HARDPERSECTION | NUMBER(10) | 每关难题 | 非空 |
| NORMALPERSECTION | NUMBER(10) | 每关一般题 | 非空 |
| SORCE | VARCHAR(255) | 分数配比 | 非空 |

存储知识信息的数据库表设计（如表4.6所示）。

表4.6 数据库表KNOWLEDGE的主要字段描述

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键Id | 主键id |
| ISDELETE | NUMBER(10) | 是否删除 | 默认为0 |
| KNOWLEDGENAME | VARCHAR(255) | 知识点名称 | 非空 |
| CATEGORYNAME | VARCHAR(255) | 分类名称 | 非空 |
| CATEGORYID | NUMBER(10) | 分类Id | 非空 |

存储组织信息的数据库表设计（如表4.7所示）。

表4.7 数据库表ORGANIZATION的主要字段描述

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键 | 自增 |
| ISDELETE | NUMBER(10) | 是否删除 | 默认0 |
| ORGANIZATIONID | NUMBER(10) | 组织ID | 非空 |
| ORGANIZATIONLEVEL | NUMBER(10) | 组织级别 | 非空 |
| ORGANIZATIONLEVELNAME | VARCHAR(255) | 级别名称 | 非空 |
| ORGANIZATIONNAME | VARCHAR(255) | 组织名称 | 非空 |
| ORGANIZATIONPATH | VARCHAR(255) | 组织全路径 | 非空 |
| PARENTID | NUMBER(10) | 父节点Id | 非空 |

存储用户信息的数据库表设计（如表4.8所示）。

表4.8 数据库表ORGANIZATIONUSER的主要字段描述

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键 | 自增 |
| ISDELETE | NUMBER(10) | 是否删除 | 默认0 |
| JOBNUMBER | VARCHAR(255) | 工号 | 非空 |
| NAME | VARCHAR(255) | 名字 | 非空 |
| ORGANIZATIONID | NUMBER(10) | 组织ID | 外键 |
| ORGANIZATIONPATH | VARCHAR(255) | 组织全路径 | 非空 |
| PHONENUMBER | VARCHAR(255) | 电话号码 | 非空 |

存储奖品信息的数据库表设计（如表4.9所示）。

表4.9 数据库表PRIZE\_CHAPTER的主要字段描述

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键ID | 自增 |
| ACTIVITYID | NUMBER(10) | 活动ID | 外键 |
| CHAPTERID | NUMBER(10) | 大关卡ID | 非空 |
| CONTENT | VARCHAR(255) | 奖品名 | 非空 |
| REMAININGAMOUNT | NUMBER(10) | 剩余奖品数 | 非空 |
| RATE | FLOAT(126) | 中奖率 | 非空 |
| TITLE | VARCHAR(255) | 奖品级别 | 非空 |
| TOTALAMOUNT | NUMBER(10) | 总数奖品 | 非空 |

存储用户得奖信息的数据库表设计（如表4.10所示）。

表4.10 数据库表PRIZE\_USER的主要字段描述

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键 | 自增 |
| ACTIVITYID | NUMBER(10) | 活动ID | 外键 |
| CHAPTERID | NUMBER(10) | 大关ID | 外键 |
| CHAPTERPRIZEID | NUMBER(10) | 大关奖品ID | 外键 |
| CONTENT | VARCHAR(255) | 奖品名称 | 非空 |
| SECTIONID | NUMBER(10) | 关卡ID | 非空 |
| TITLE | VARCHAR(255) | 中奖级别 | 非空 |
| USERID | NUMBER(10) | 用户ID | 非空 |
| CREATETIME | DATE | 创建时间 | 当前时间 |
| USERNAME | VARCHAR(255) | 用户名 | 非空 |

存储试题信息的数据库表设计（如表4.11所示）。

表4.11 数据库表QUESTIONBANK的主要字段描述

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键ID | 自增 |
| CATEGORYID | NUMBER(10) | 分类ID | 非空 |
| CATEGORYNAME | VARCHAR(255) | 分类名称 | 非空 |
| DIFFICULTY | NUMBER(10) | 困难程度 | 非空 |
| DIFFICULTYNAME | VARCHAR(255) | 难度 | 困难名称 |
| ISDELETE | NUMBER(10) | 是否删除 | 默认0 |
| KNOWLEDGENAME | VARCHAR(255) | 知识点名称 | 知识点名称 |
| QUESTIONCONTENT | VARCHAR(255) | 问题文本 | 问题 |
| RIGHTANSWER | VARCHAR(255) | 正确答案 | 非空 |
| SERIALNUMBER | VARCHAR(255) | 编号 | 非空 |
| TYPE | NUMBER(10) | 类型 | 非空 |
| TYPENAME | VARCHAR(255) | 类型名称 | 非空 |

存储抽题信息的数据库表设计（如表4.12所示）。

表4.12 数据库表QUESTIONSHEET的主要字段描述

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键ID | 自增 |
| ACTIVITYID | NUMBER(10) | 活动ID | 外键 |
| CREATEDATE | DATE | 创建时间 | 当前时间 |
| QUESTIONJSON | CLOB | 问题json | 大字段 |
| QUESTIONS | NUMBER(10) | 问题数 | 非空 |
| RIGHTANSWEJSON | CLOB | 答案json | 大字段 |

存储排名的数据库表设计（如表4.13所示）。

表4.13 数据库表RANKLIKE的主要字段描述

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键 | 非自增 |
| ACTIVITYID | NUMBER(10) | 活动ID | 外键 |
| CERATETIME | DATE | 创建时间 | 当前时间 |
| LIKETYPE | NUMBER(10) | 排名类型 | 人或者组织 |
| TARGEID | NUMBER(10) | 被点赞ID | 非空 |
| USERID | NUMBER(10) | 用户ID | 用户非空 |

存储用户答题概况的数据库表设计（如表4.14所示）。

表4.14 数据库表USERSECTION的主要字段描述

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键 | 自增 |
| ACTIVITYID | NUMBER(10) | 活动ID | 外键 |
| CORRECTCOUNT | NUMBER(10) | 正确个数 | 非空 |
| CORRECTRATE | FLOAT | 正确率 | 非空 |
| CREATETIME | DATE | 创建时间 | 创建时间 |
| DEPARTMENTID | NUMBER(10) | 部门ID | 非空 |
| DEPARTMENTNAME | VARCHAR(255) | 部门名称 | 非空 |
| ERRORCOUNT | NUMBER(10) | 错误个数 | 非空，默认0 |
| LASTMODIFYTIME | DATE | 最后修改时间 | 当前时间 |
| QUESTIONSHEETID | NUMBER(10) | 问题列ID | 非空 |
| SCORE | NUMBER(10) | 分数 | 非空 |
| SECTIONID | NUMBER(10) | 关卡ID | 非空 |
| STARS | NUMBER(10) | 星数 | 非空 |
| SUBMITCOUNT | NUMBER(10) | 提交次数 | 实时 |
| USERID | NUMBER(10) | 用ID | 非空 |
| USERNAME | VARCHAR(255) | 用户名 | 非空 |
| SUBMITSECONDS | NUMBER(10) | 提交时间 | 当前时间 |

存储用户所有答题记录的数据库表设计（如表4.15所示）。

表4.15 数据库表USERSECTIONHISTORY的主要字段描述

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键 | 自增 |
| ACTIVITYID | NUMBER(10) | 活动ID | 外键 |
| CORRECTCOUNT | NUMBER(10) | 正确个数 | 非空 |
| CORRECTRATE | FLOAT | 正确率 | 非空 |
| CREATETIME | DATE | 创建时间 | 创建时间 |
| ERRORCOUNT | NUMBER(10) | 错误个数 | 非空，默认0 |
| QUESTIONSHEETID | NUMBER(10) | 问题列ID | 非空 |
| SCORE | NUMBER(10) | 分数 | 非空 |
| SECTIONID | NUMBER(10) | 关卡ID | 非空 |
| STARS | NUMBER(10) | 星数 | 非空 |
| SUBMITCOUNT | NUMBER(10) | 提交次数 | 实时 |
| STARTIME | DATE | 开始时间 | 非空 |
| SUBMINTTIME | DATE | 提交时间 | 非空 |
| USERID | NUMBER(10) | 用ID | 非空 |
| USERNAME | VARCHAR(255) | 用户名 | 非空 |
| SUBMITSECONDS | NUMBER(10) | 提交时间 | 当前时间 |

存储后台管理员的数据库表设计（如表4.16所示）。

表4.16 数据库表BACKGROUNDUSER的主要字段描述

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 含义 | 备注 |
| ID | NUMBER(10) | 主键 | 自增 |
| LOGINNAME | VARCHAR(255) | 登录名 | 非空 |
| PASSWORD | VARCHAR(255) | 登录密码 | 加密 |

## 4.4 本章小结

本章展示了系统整体模块的静态结构图，并绘制了一张技术架构图和数据库E-R设计图，展现表中的主要信息字段。整体设计是系统开发的重要一部分，为后续开发铺垫。

5 详细设计

## 5.1 活动创建模块

以下对于活动模块的创建的详细设计。首先对功能介绍分析以及类的设计，然后对比较重要的逻辑用流程图进行了展现。

### 5.1.1 类设计

该模块主要有两大类：ActivityController与ActivityBiz，下面将对这几个类进行简单介绍（如表5.1所示），活动创建后试题相应生成。

表5.1 ActivityController类主要方法描述表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 方法名 | 返回类型 | 作用 | 调用方式 |
| 1 | Create() | String | 调用活动创建方法 | /activity |
| 2 | Delete（） | String | 调用删除活动方法 | /activity/d/{id} |
| 3 | Update（） | String | 调用更新活动方法 | /activity/update |
| 4 | show（） | String | 调用展示活动方法 | /activity/showOne |
| 5 | checkTime（） | String | 调用检查时间方法 | /activity/checkTime |

表5.2 ActivityBiz类主要方法描述表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 方法名称 | 返回类型 | 作用 |
| 1 | Create() | String | 创建活动 |
| 2 | Delete() | String | 删除活动 |
| 3 | Update() | String | 更新活动 |
| 4 | Show() | Activity实体 | 展示活动 |
| 5 | checkTime() | Integer | 检查时间 |

### 5.1.2 流程设计

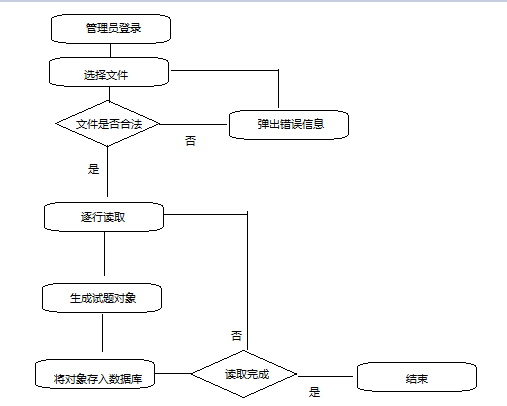
以下是对这一模块的关键步----试题导入进行流程分析。

图5.1 批量试题导入流程图

批量导入试题的主要流程如下：

1. 管理员登录后台的系统。
2. 选择文件，判断文件是否符合要求如果不符合要求返回错误提示信息。
3. 文件上传到服务器开始导入。
4. 每读取一行，则生成一个试题对象，同时存入数据库。
5. 判断文件读取是否完成，若没有完成继续读取。
6. 文件读取结束，则关闭软件，结束所有流程。

## 5.2导入模块

以下对于导入模块的创建的详细设计。对类设计进行详细阐述说明，然后对核心业务用流程图进行了展现。

### 5.2.1 类设计

该模块是整个系统的导入模块，负责用户的导入，组织导入，试题导入等，主要的类有FileController、QuestionImport、OrganizationService

、OrganizationUserService、MSExcel等，其详细介绍如下表。

表5.3 FileController类主要方法描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 方法名 | 返回类型 | 作用 | 调用方式 |
| 1 | uploadFile（） | String | 前台导入的入口 | File/upload |

表5.4 QuestionImport类主要方法描述

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 方法名 | 返回类型 | 作用 |
| 1 | importFromExcel（） | Boolean | 导入问题 |

表5.5 OrganizationService类主要方法描述

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 方法名 | 返回类型 | 作用 |
| 1 | inputOrganizationFromExcel（） | String | 导入组织 |

表5.6 OrganizationUserService类主要方法描述

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 方法名 | 返回类型 | 作用 |
| 1 | importFromExcel（） | String | 导入用户 |

表5.7 MSExcel类主要方法描述

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 方法名 | 返回类型 | 作用 |
| 1 | parseExcel（） | List | 导入类的基类 |

首先FileController接入导入流程，在此类中进行一次盘判断，如果是问题导入则调用QuestionImport的importFromExcel（）方法；如果是组织的导入则调用

OrganizationService的inputOrganizationFromExcel（）方法；如果是人员导入的话则调用OrganizationUserService的importFromExcel（）方法；所有的导入方法都是基于MSExcel的parseExcel（）方法。

### 5.2.2 流程设计

图5.2 导入流程图

1. 进入导入模块，
2. 选择导入类型：人员导入，试题导入或者组织导入等。
3. 导入过程中是否报错，如果报错返回报错信息。
4. 具体导入流程见图5.1。
5. 导入完成。

## 5.3答题模块

本模块主要是的答题界面的题目显示，答案的提交以及提交后是否过关的判断三大模块。本模块是整个系统的主要服务对象。

### 5.3.1 类设计

表5.8 SectionController类主要方法描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 方法名 | 返回类型 | 作用 | 调用方式 |
| 1 | showQuestion（） | String | 展示题目 | /section/question |
| 2 | returnShow（） | String | 提交答案 | /section/commitAnswer |
| 3 | Sections（） | String | 个人关卡 | /section |

前台接入答题流程，调用showQuestion（）方法进行基础判断后展示题目内容，用户答题后点击提交，调用returnShow（）方法并展示答题详情。返回至关卡页面调用Section（）方法展示关卡。

### 5.3.2流程设计

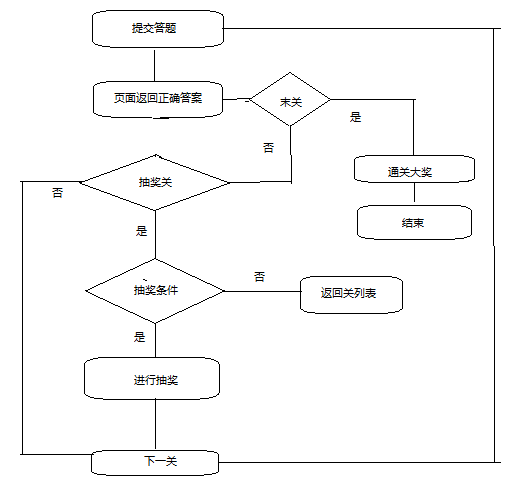
此部分只对相对复杂的提交页面进行流程设计展示（如图5.3所示）。

图5.3 提交答案流程

下面针对提交流程进行阐述说明：

1. 选择答案然后提交，答案在页面回显。
2. 判断是否是最后一关，如果是最后一关，直接获得通关大奖，答题结束。
3. 不是最后一关则要判断是否是抽奖关。若不是抽奖关则直接下一关。
4. 是抽奖关，则判断是否达到抽奖条件，如果每达到就直接返回关列表。
5. 达到条件则进行抽奖。
6. 进入下一关再次循环，直至结束。

## 5.4排名模块

本模块是对于系统中的排名模块进行分析设计。作者先是做了类模块设计，之后把核心的业务用流程变现出来，画出流程图来直观展现。

### 5.4.1类设计

表5.9 RankController类主要方法描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 方法名 | 返回类型 | 作用 | 调用方式 |
| 1 | personRank（） | String | 个人排名 | /rank/person |
| 2 | personRankSituation（） | String | 个人排名情况 | /rank/ personSituation |
| 3 | departmentRank（） | String | 部门排名 | /rank/department |
| 4 | departmentRankSituation（） | String | 部门排名情况 | /rank/departmentSituation |

表5.10 UserSectionBiz类主要方法描述

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 方法名 | 返回类型 | 作用 |
| 1 | personRank() | map | 个人排名 |
| 2 | personRankSituation | map | 个人排名情况 |
| 3 | departmentRank | map | 部门排名 |

当提交答案后会显示你当前排名以及打败人数，此时前台接入排名流程，调用RankController中personRank（）方法显示出这些参数。当查看中心以及个人排名的时候，系统会调用RankController类中的personRankSituation（）方法和departmentRankSituation（）方法，后台经过一系列的运算得出最后数据返回给前台。

### 5.4.2流程设计

图5.3 中心和个人排名流程

## 5.5逻辑设计

本模块作者选取安全验证、管理员安全验证、答题倒计时、抽奖功能进行代码阐述。

### 5.5.1 答题系统安全验证

知识竞赛系统的安全性在很大程度上就是检测系统非法登录，避免人为的去破坏系统以及更改系统配置，为避免此类事件发生设计答题安全验证功能，部分代码实现（附录1.1）。

作者的设计思路是：答题时候前台会传入一个用ID值，如果答题人员并非本系统认证人员将无法进入本系统，系统会捕捉到异常并返回给前台错误信息，致使无法正常进入答题系统。

### 5.5.2 管理员安全验证

后台管理是整个系统的核心部分，如果后台被他人盗取或者篡改将会引起很严重的错误，致使系统宕机，给企业带来经济亏损，所以管理员登录验证这一块是必不可少的。对于后台管理员的验证，作者列出了部分实现代码（附录1.2）。

作者的设计思路是：通过前台登录后生成cookie 系统检测这个cookie的有效性，判断cookie是否过期或者是否没有，如果过期或者没有则页面不允许登录吗，保证了系统的安全。

### 5.5.3 答题倒计时

网络知识竞赛系统中一个倒计时是不可或缺的功能，它给参与者提供了时间参考，能让比赛者在有限的时间发挥最佳的水平。本系统作者利用AJAX技术实现了无动态刷新实现倒计时功能，在参与者登陆成功，主界面就会显示倒计时，部分的代码实现（附录1.3）。

作者的设计思路是：先从数据库中拿到一个现在时间加上当前时间等于一个未来时间，当达到未来时间或者大于未来时间时倒计时结束，进行后续的操作，停止答题返回首页，具体的UI实现详见下一章节。

### 5.5.4 抽奖设计

为了提高系统的可玩性，在系统中添加了抽奖环节，当用户每关达到一定的条件后都会弹出抽奖界面，给与抽奖，系统会根据给定的概率以及相对应的奖品随机得出中奖与否，抽奖模块的运作部分流程代码（附录1.4）。

作者设计思路：作者首先用map集合以键值对形式存放奖品以及对应的概率然后将对应集合存放在list集合中，然后以这些键值对的Value概率进行随机得到一个角标，然后通过这个角标值拿到一个list集合中一个值，即抽奖获得奖品。

## 5.6 本章小结

本章中，作者以模块为单位做了较为设计与构建，对几个关键的模块进行类设计之后对其流程做了阐述，运用流程图进行了比较形象的展示。该运作模式在企业级应用开发中运用较多[19]，系统设计是整个系统的核心部分，它是整个系统开发与编码的指向，对后面的开发起了关键的作用。

6 系统部署与应用实例

## 6.1 开发工具以及部署环境

本系统的开发工具（如图表 6.1所示）。

表6.1 系统开发工具

|  |  |
| --- | --- |
| 名称 | 描述 |
| 操作系统 | Windows10/Linux（Ubuntu 16.10） |
| 开发语言 | Java+JavaScript |
| 开发工具 | IntelliJ IDEA the Java IDE |
| 数据库设计 | Powerdesigner 16.5 |
| 数据库 | Oracle-xe-11g |

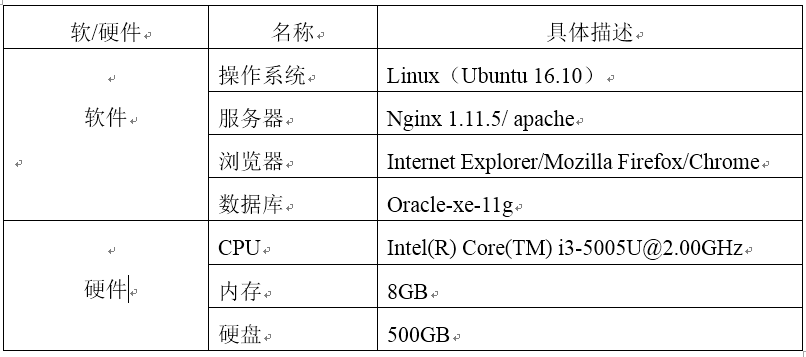
本系统的部署环境（如图6.1所示）。

图6.1 部署环境

## 6.2 系统运行界面

下面将展示系统的实际运行效果，这里列举了从后台登录到活动的发布，以及前台答题提交等几个核心模块的部分运行界面。

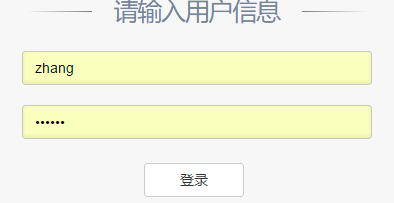
 首先我们看到的是后台登录（如图6.2所示）。

图6.2 后台登录

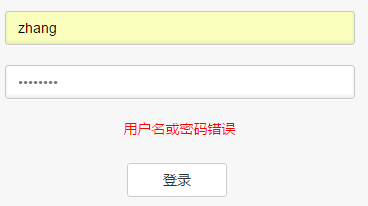
如果两项其中一项错误时，界面将会提示（如图6.3所示）。

图6.3 密码错误提示

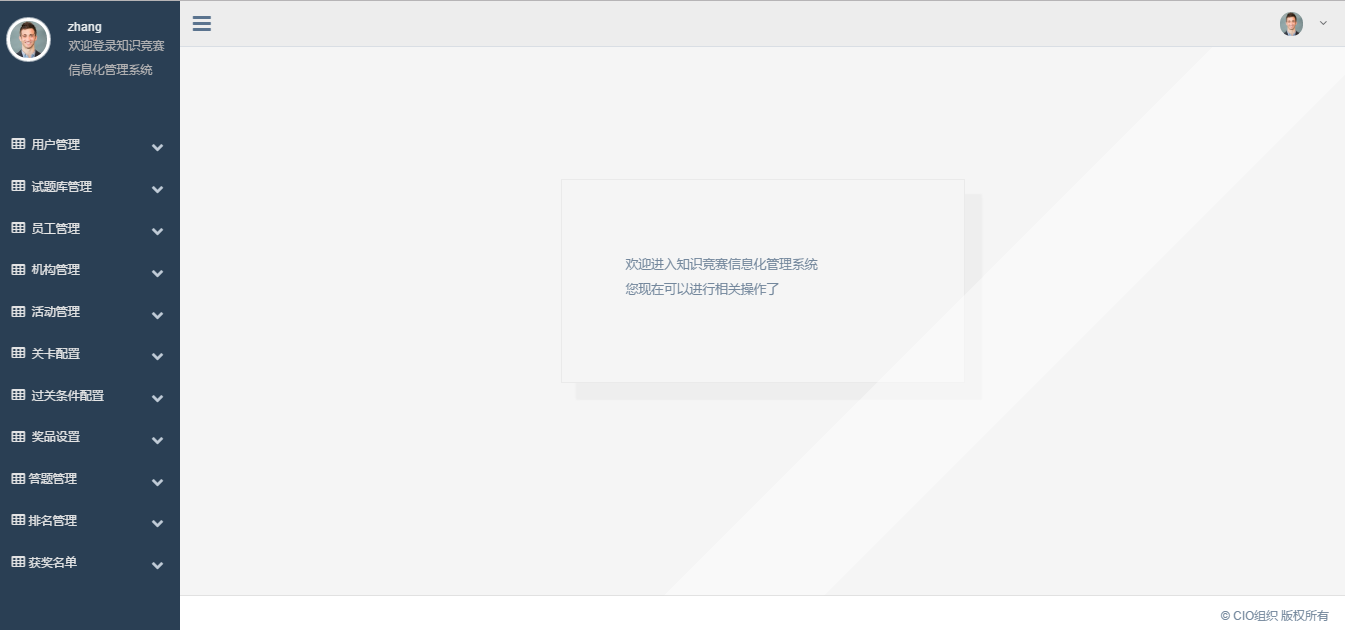
当登录进入系统之后将会看到所有必要模块（如图6.4所示）。

图6.4后台主界面

创建比赛需要导入题目、活动创建、关卡分数配比、过关条件配置以及奖品配置几个步骤，首先我们要先导入试题（如图6.5所示）。

图6.5试题导入

导入试题之后就可以创建活动，活动可以设置有效时间，活动描述，活动关卡数，活动题目难易程度比等（如图6.6所示）。

图6.6 活动创建

活动创建完成后可进行关卡配置，关卡分数配比难易度配比（如图6.7所示）。

图6.7 关卡分数配比

试题难易程度配比完成后可进行关卡过关条件配置（如图6.8所示）。

图6.8 过关条件配置

最后进行奖品设置（如图6.9所示）。

图6.9 奖品设置

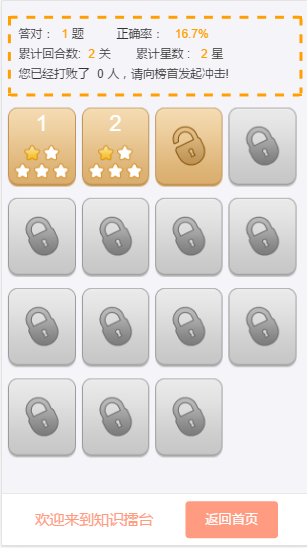
 最后就可以进行知识竞赛答题，首页（如图6.10所示）。

图6.10 答题首页

 点击帮助可以查看帮助事宜（如图6.11所示），微排名包括中心排名，个人排名情况等，可以对排名的条目进行点赞（如图6.12所示），点击知识微答，即可展示关卡项目（如图6.13所示）。

图6.11 帮助选项

图6.12 排名以及点赞

图6.13 关卡展示列表

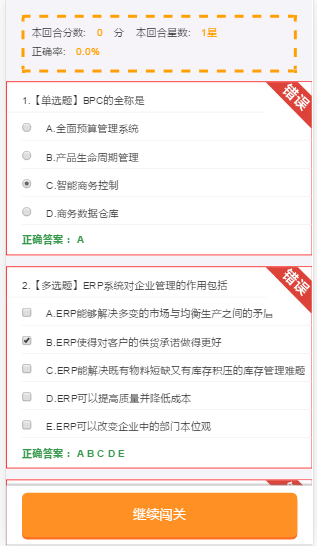
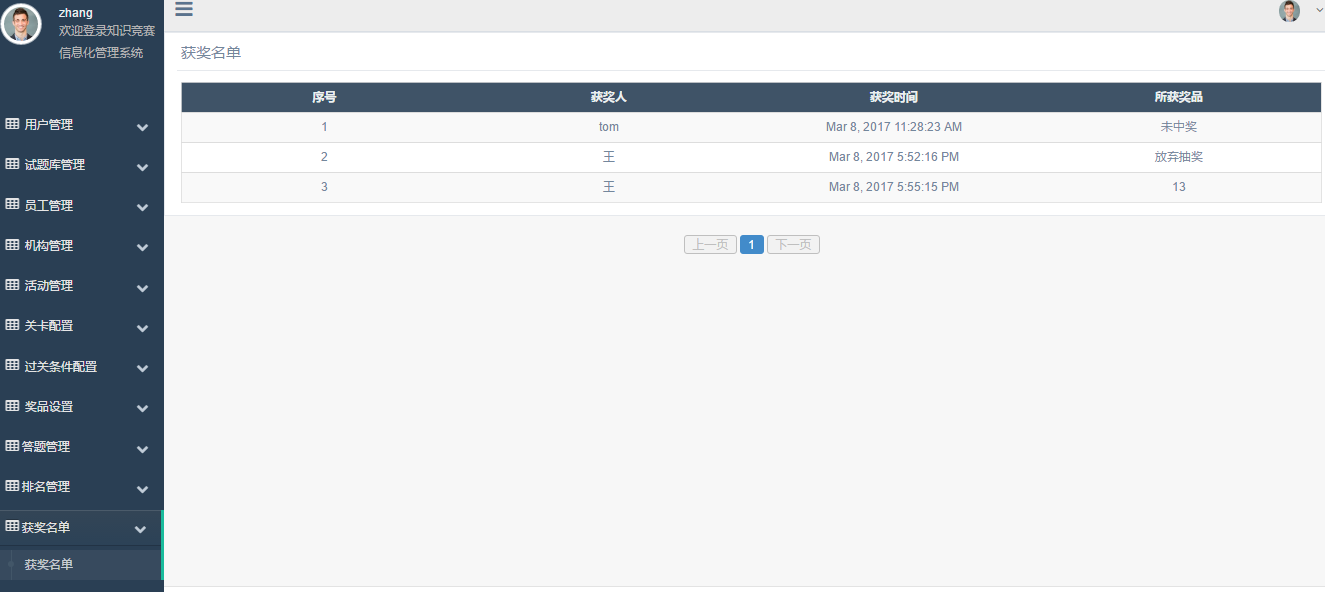
点击黄色的图表即可答题，提交答案后即可显示正确答案（如图6.14所示）。

图6.14 答题界面展示

当获得一定的积分或者星数时就可以进行抽奖（如图6.15所示）与（图6.16所示）。

图6.15 抽奖提醒 图6.16 抽奖界面

至此一套完整的流程基本完成，后台可查询到抽奖详细情况，排名情况，以及答题情况，供以后奖品的发放做参考（如图6.17所示），（如图6.18所示），（如图6.19所示），（如图6.20所示）。

图6.17 获奖名单

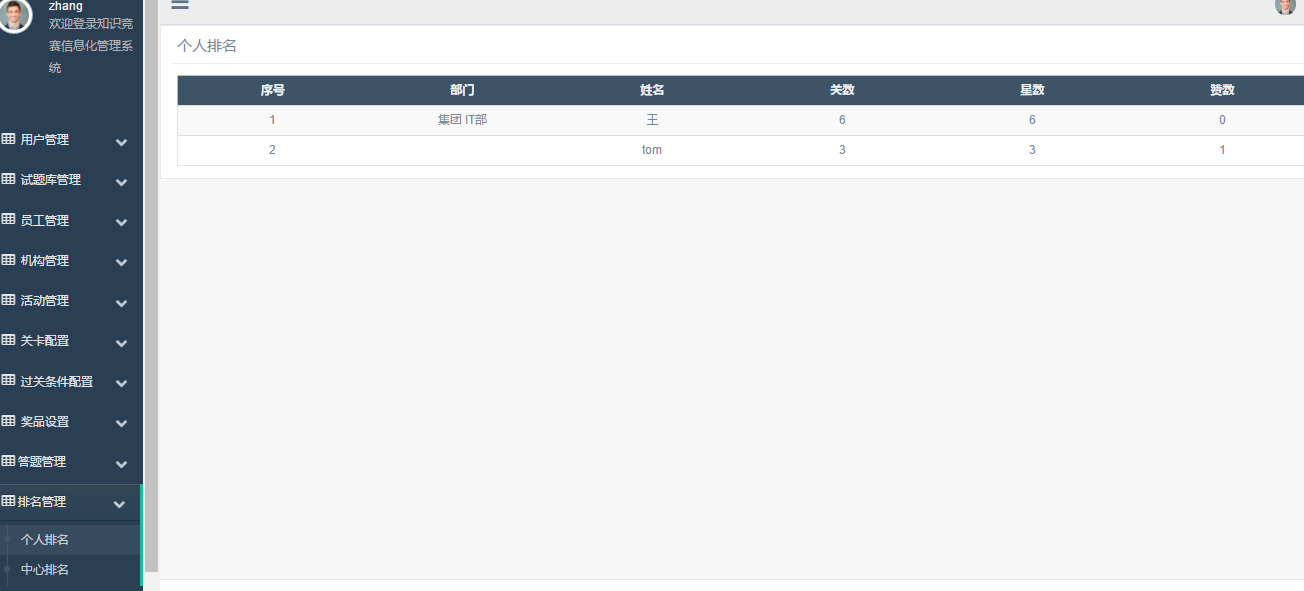
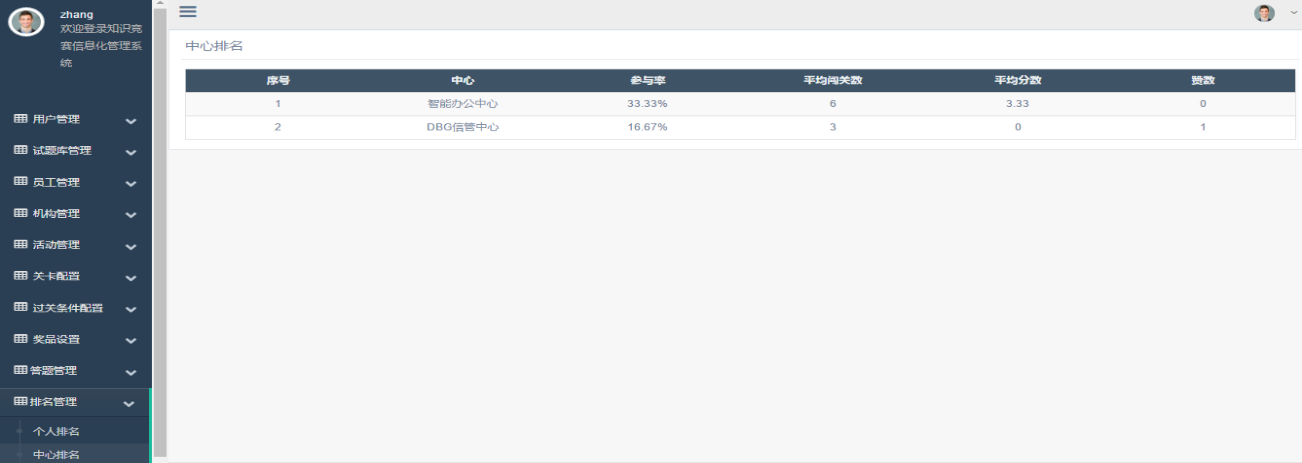
图6.18 个人排名

图 6.19 中心排名

图 6.20 答题情况

## 6.3 本章小结

本中章，论文从总体上上介绍了开发工具以及部署的环境，之后对系统整个流程就行了实际运作界面的展示。至此本竞赛系统从需求建模、设计、实现、再到编码以及整个前后台的部署基本全完成，系统可以投入实际运用中。

7 总结与展望

## 7.1 总结

本系统的设计进行了非常的详细的需求研究，根据实际以及使用之需，精心开发了一个用于知识竞赛信息化管理的平台，系统分析了具体的应用建立了详细的使用模型，实现了页面设计和数据库的建设，并对系统中的几个核心模块活动创建、用户管理、导入数据、比赛管理、以及抽奖管理等进行了详细的分析，系统可以投入实际使用。

利用了html5的以及JQuery前端框架等较为先进的技术，对系统的界面进行了精心的排版布局，并对传入后台的数据进行了充分的验证以及处理工作，同时利用AJAX技术对后台实现了无动态刷新，有效的解决了页面加载慢的问题，提高了系统流畅性和可用性。

与此同时本系统最大的亮点的是加上了抽奖环节，这样非常的有效了的提高了用户参与进来的兴趣，非常愿意参与到本答题竞赛中，这也正好符合企业创建此次活动的意图，提高了员工的积极性并且把企业的文化得到了很好的宣传，可以说的两全其美，进言之也大大提高了本系统的用户体验度与实际使用价值。

系统使用了springboot做基础框架大大提高了开发效率，少了许多配置类文件，减少了工作难度以及复杂度[20]，其中镶嵌Spring Data JPA等主流的技术，增加了系统的灵活性以及开发便捷性。

## 7.2 展望

基于springboot知识竞赛系统的设计与实现涉及的方面非常多，本系统还有些许问题需要改进，需要在正常的使用中寻找不足然后不断地改进不断优化，总结了以下几个方面需要做进一步的研究：

1. 本系统只考虑了试题信息以及分类等几个基本组成，并没有在试题的统计面有更加深入的挖掘。
2. 在用户答题的统计方面只是对用于的几个基本信息进行了统计，并没有统计答对或者答错什么题目，然而这时未来将要做的一个重要环节，这个可以方便用户知道在什么方面比较薄弱，进而也可以提高系统的实用范围，增加应用面。
3. 在试题库的方面，略显不够灵活须在固定格式之后才能正常导入，这一点也需要进一步的完善，试题的接口还需要增加一些比如修改之类的。

# 参考文献

1. 高春燕、张金辉，JavaScript入门经典，北京：机械工业出版社，2013
2. 邱俊涛，Java Script 核心概念及实践，北京：人民邮电出版社，2013
3. Michael Morrison，Head first Java Script，Nanjing: Southeast University Press，2008
4. Paul Hatcher with John W. Gosney，Java Script professional projects，Boston，MA: Premier Press，2004
5. 孙向阳、李军，JavaScript和j Query实战手册，北京:机械工业出版社，2013
6. 周志雄，基于j Query框架技术开发的Web应用，科协论坛，1:91-92, 2013
7. Ryan Benedetti, Ronan Cranley, Head first jQuery, Nanjing: Southeast University Press，2012
8. jQuery Community Expert, jQuery cookbook, Nanjing: Southeast University Press， 2010
9. 姚军，CSS与HTML Web设计实践指南，北京:人民邮电出版社，2009
10. 梁文新、宋强、刘凌霞，Ajax + PHP程序设计实战详解，北京:清华大学出版社，2010
11. 刘红伟，Ajax, JavaScript和PHP编程导学，北京:机械工业出版社，2009
12. Anthony T. Holdener III, Ajax: the definitive guide, Nanjing: Southeast University Press，2008
13. Brett McLaughlin, Head rush Ajax，Nanjing: Southeast University Press，2006
14. Michael Mahemoff , Ajax design patterns，Nanjing: Southeast University Press，2006
15. 王德民、王新颖、刘听，Ajax与PHP Web开发，北京:人民邮电出版社，2007
16. 褚法正、周峰、孙更新，Ajax开发技术原理与实践教程，北京:电子工业出版社，2007
17. 陈俊荣，网页程序设计:HTML, JavaScript, CSS, XHTML, Ajax，北京:清华大学出版社，2008
18. 李丽新，程序设计竞赛系统的设计及实现 Dl.吉林:吉林大学，2008:25-30
19. 陈雄华，Spring 企业级应用开发详解[M] 北京:电子工业出版社 2009
20. PaulDeck， SpringMVC学习指南 ，北京:人民邮电出版社 2005

致　谢

时光荏苒，转眼间大学四年学习生活就要结束了，在论文完成之际，向我的导师胡平芳老师致以最真挚的感谢和最诚挚的敬意。从理论学习、论文选题、工程实践到论文的撰写，胡均给予了悉心、细致的指导，使我不断的进步，受益匪浅。

在学习期间，结识了武汉东湖学院的各位老师，您们渊博的学识，严谨的治学态度，科学的工作方法，使我在学习和工作中受到极大的鼓舞和启发。感谢您们在学习中给予我的支持和帮助。也感谢本论文中给予我热心帮助的张瑞同学。

感谢学习期间同窗好友给我的帮助，结识你们是我的另一份收获!

附 录

## 1.1

if (null == userId || "".equals( userId ) || "undefined".equals( userId ) ) {

return ValueUtil.toError("notLogin","未找到员工工号");

}

if ( null == activityId || 1==activityId ) {

try {

activityId = activityBiz.currentActivityId();

} catch (Exception e) {

return ValueUtil.toError("noCurrentActivity",e.getMessage());

}

}

## 1.2

|  |
| --- |
| public static String verify(HttpServletRequest request,HttpServletResponse response) throws HzbuviException {  response.setHeader("Access-Control-Allow-Origin", ConstantData.backendManageUrl);  Cookie[] cookies = request.getCookies();  String userCode = "";  if ( null != cookies ) {  for (int i = 0; i < cookies.length ; i++) {  switch (cookies[i].getName()) {  case "uid" :  userCode = cookies[i].getValue();  System.out.println(System.currentTimeMillis()+":uid:"+userCode);  break;  }}  return userCode;  } else {  System.out.println(System.currentTimeMillis()+":notLogin:");  HzbuviException exception = new HzbuviException("notLogin");  exception.setMsg("请重新登录");  throw exception;  }} |
| try {  SessionData.verify(request,response);  } catch (HzbuviException e) {  return ValueUtil.toError(e.getCode(),e.getMsg());  } |

## 1.3

var timeStamp = '';

function calcTime(e){

if( document[prefix + 'Hidden'] || document.hidden ){

timeStamp = +new Date();

} else {

timeStamp = Math.floor((+new Date() - timeStamp)/1000);

time = time - timeStamp;

if(time<0){ time = 0; }

}

}

## 1.4

|  |
| --- |
| Map<Integer, Double> map = new HashMap<>();  List<Double> list = new ArrayList<>();  chapterPrizes.forEach(chapterPrize -> {  Integer prizeId = chapterPrize.getId();  Double rate =chapterPrize.getRate();  map.put(prizeId, rate);  list.add(rate);  });  Integer pid =sortByRate(map, list); |
| private Integer sortByRate(Map<Integer, Double> map, List<Double> list) {  Collections.sort(list);  List<Integer> keys = getKeyList(map, list);  Integer n = changeToInt(list);  if (ValueUtil.isEmpity(n)) {  return null;  }  return keys.get(n);  } |