

**毕业设计（论文）**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于BS的高校微课比赛管理系统的 | | |
|  | 分析与实现 | | |
| 学 院 | 软件学院 | | |
| 专 业 | 软件工程 | | |
| 学生姓名 | 蒋礼奎 | 学 号 | 159000302 |
| 指导教师 | 孟高原 | 职 称 | 讲师 |

2019年 3 月 24 日

学生毕业设计（论文）原创性声明

本人以信誉声明：所呈交的毕业设计（论文）是在导师的指导下进行的设计（研究）工作及取得的成果，设计（论文）中引用他（她）人的文献、数据、图件、资料均已明确标注出，论文中的结论和结果为本人独立完成，不包含他人成果及为获得重庆工程学院或其它教育机构的学位或证书而使用其材料。与我一同工作的同志对本设计（研究）所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

毕业设计（论文）作者（签字）：

年 月 日

摘 要

本文在分析目前各高校微课比赛的基础上，对高校微课比赛系统的功能和业务进行深入的调查和研究，对其主要的业务和功能进行分析、设计和实现，具体的研究内容包括：

（1）了解高校微课比赛系统在国内外发展情况，对高校微课比赛系统的主要功能和业务调查研究，应用面向对象的分析方法，进行较为详细的功能需求分析和性能需求分析。

（2）依据高校微课比赛系统的运行特点，主要从系统的设计思想、设计原则、设计目标、体系结构、系统平台等进行研究，制定出了适合当代高校微课比赛系统的系统分析方案和架构原型。

（3）分别高校微课比赛系统的各个子系统、用例，从功能、业务流程、工作原理等方面进行研究、分析和设计，应用SSM框架设计思想开发实现。

关键字：微课比赛 SPA SSM框架

## ABSTRACT

Based on the analysis of the current micro-course competition in colleges and universities, this paper makes an in-depth investigation and research on the function and business of the micro-course competition system in colleges and universities, and analyzes, designs and realizes its main business and functions, and the specific research contents include:

(1) To understand the development of the micro-course competition system in colleges and universities at home and abroad, to investigate the main functions and business of the micro-course competition system in colleges and universities, and to apply object-oriented analysis methods to carry out more detailed functional requirements analysis and performance requirements analysis.

(2) According to the operation characteristics of the micro-course competition system in colleges and universities, this paper mainly studies the system design idea, design principle, design goal, architecture, system platform and so on, and formulates the system analysis scheme and architecture prototype suitable for the micro-course competition system of contemporary colleges and universities.

(3) Each subsystem and use case of the micro-course competition system in colleges and universities are studied, analyzed and designed from the aspects of function, business process and working principle, and are developed and realized by using SSM framework design idea.

**Keyword**: Micro Course competition；SPA；SSM framework

[1 绪论 7](#_Toc780340750)

[1.1 课题背景 7](#_Toc504353021)

[1.2 国内研究现状 8](#_Toc543269238)

[1.3 国外研究现状 8](#_Toc1773099669)

[1.4 论文的组织结构 9](#_Toc2003051111)

[1.5 本章小结 9](#_Toc1326372205)

[2 系统运用技术 10](#_Toc1457393575)

[2.1面向对象的分析及设计 10](#_Toc215337343)

[2.2 开发工具 10](#_Toc664778606)

[3 需求分析 12](#_Toc1724099348)

[3.1业务背景 12](#_Toc940892865)

[3.2业务主角 14](#_Toc1664289194)

[3.3业务分析 14](#_Toc733981383)

[3.4业务实体分析 15](#_Toc879035713)

[3.5非功能需求 15](#_Toc1413220678)

[3.6本章小结 16](#_Toc830799326)

[4 系统分析 17](#_Toc305599288)

[4.1系统总体业务视图 17](#_Toc1573833439)

[4.2系统用例 17](#_Toc862529174)

[4.3 比赛管理用例描述 20](#_Toc1013210168)

[4.4 微课作品上传用例描述 21](#_Toc1625456513)

[4.5微课作品打分评论用例描述 21](#_Toc908140504)

[4.6域模型 22](#_Toc951171499)

[4.7架构设计 22](#_Toc471115425)

[4.8网络结构 25](#_Toc264741486)

[4.9 分析模型 25](#_Toc2071522265)

[4.10本章小结 28](#_Toc1069822691)

[5 系统的设计及实现 29](#_Toc1776874953)

[5.1数据库设计 29](#_Toc1029739889)

[5.2系统界面原型 33](#_Toc267603250)

[5.3 比赛管理用例 34](#_Toc777065932)

[5.4 微课作品上传用例 36](#_Toc1299061717)

[5.5 微课作品打分评论用例 39](#_Toc2011522217)

[5.6系统测试 41](#_Toc1966330045)

[5.7本章小结 43](#_Toc483222632)

[工作总结与展望 43](#_Toc1887106717)

[致谢 44](#_Toc416610076)

[参考文献 45](#_Toc1168858112)

## 1 绪论

高校微课比赛系统是当代高校学习主流的一个重要部分，是社会发展的必要趋势。打造良好的微课比赛系统是有利于高校教师喝学生的学习进步，打开了另一种薪的学习验收成果方式。在设计系统时如果出现错误或者处理方法不合适，首先会影响教师的工作积极性及学生的学习主动性，其次会影响社会网络的稳定发展。因此，一个良好的高校微课比赛系统则可以规范其过程，提高高校微课管理效率。这对于高校的管理和发展具有非常重要的意义。

### 1.1 课题背景

在Internet技术发展的社会，学生的学习不可忽视的一部分，学习是提高社会整体素质水平的基础，也是推动社会发展的前提。“微课”的灵感来源于孟加拉裔美国人萨尔曼·汗倡导的“用视频再造教育”，雏形最早见于美国北爱荷华大学Leroy A McGrew教授1993年所提出的60秒课程(60 second Course)，2008年由美国新墨西哥州圣胡安学院的戴维· 彭罗斯（ David Penrose）首次提出“微课程”， 戴维· 彭罗斯把微课程称为 “知识脉冲”（ Knowledge Burst）[3]，其核心理念是在课程中把教学内容与教学目标紧密地联系起来，以产生一种“更加聚焦的学习体验”。特别是从2012年开始，随着反转课堂、可汗学院等概念的普及，微课作为优质的新型教育资源的创新，受到众多学者、专家浓厚研究兴趣。

传统教育按同一个模式培养学生，学生被视作“一个集合体”来处理，但其实学生根据所处家庭环境的不同、心理发展和接受知识水平的不同，具有很强的差异性，同时在传统教育中，教师是教育资源的垄断者，学生是知识的承受者。随着信息化的到来，网络进入家庭、社会和学校，课堂不再是学生获取知识的唯一渠道，同时新课程标准和教学实践提出新时代的要求，教师是教学的指导者，学生是学习的主体，所以教师的工作已经不是简单的把书本上的知识内容交会给学生，而需要教师重构教学体系。此时“微课”的出现，打破了原有传统模式。“微课”既有别于传统单一资源类型的教学课例、教学课件、教学设计、教学反思等教学资源，又是在其基础上继承和发展起来的一种新型教学资源。它所讲授的内容呈点状、碎片化，这些知识点，可以是教材解读、题型精讲、考点归纳;也可以是方法传授、教学经验等技能方面的知识讲解和展示。

对于学生而言，使带有一定强制性的教学过程转变成学生高效的自学，学生的学习兴趣高涨，注意力更加集中，思维更加活跃，从而更好地掌握知识、发展技能。同时能更好的满足学生对不同学科知识点的个性化学习、按需选择学习，既可查缺补漏又能强化巩固知识，是传统课堂学习的一种重要补充和拓展资源。同时微课将学习内容细化，它符合学生一次只学一点，学太多会疲倦的学习习惯。特别是随着手持移动数码产品和无线网络的普及，基于微课的移动学习、远程学习、在线学习将会越来越普及，微课必将成为一种新型的教学模式和学习方式.

对教师而言，微课将革新传统的教学与教研方式，突破教师传统的听评课模式，教师的电子备课、课堂教学和课后反思的资源应用将更具有针对性和实效性，微课优点是课例简单，学习内容与目标单一，学习和研究时间节约，教师从“微课”中可以受到启发，有些甚至可以照搬或者迁移到自己的教育教学之中，实现教学观念、技能、风格的模仿、迁移和提升，从而迅速提升教师的课堂教学水平，促进教师的专业成长。

### 1.2 国内研究现状

进入21世纪的第二个十年，随着传媒技术的迅速发展，以“微”为标志的文化家族悄然诞生。近两年来微课作为刚刚兴起的在线教育形式，正在迅猛发展。微课内容碎片化和时间短的特点恰好满足了大众利用各种零碎时间进行学习的需求。尽管目前学校中微课应用不多，但它已经成为教育界最热门的话题之一。

微课在我国的发展趋势总体是沿着重组并创新的路线来走的，并没有彻底抛弃传统的录制视频影像资料，而是借鉴并选取其中优质资源进行重新组合，将以前的杰作升级为精简版的微课作品，微课比赛建立在微课的基础之上，大多数的微课比赛都以传统的方式建立，并无完整的管理系统的实现，综合微课在国内的现状，建立微课比赛管理系统势在必行。

### 1.3 国外研究现状

在国外，“微课”概念最早是由美国新墨西哥州圣胡安学院的高级教学设计师、学院在线服务经理戴维.彭罗斯于2008年秋首创的。戴维.彭罗斯认为微型的知识脉冲只要在相应的作业与讨论的支持下，能够在传统的长时间授课取得相同的效果。他提出了建设微课的五个步骤，罗列课堂教学中师徒传递的核心概念，这些核心概念将构成微课程的核心。

国外重视“微课程”的在于教育方面的研究，但其核心组成资源不统一，研究取向不同，有的是教案式，有的是视频式；有的只是针对于老师，有的针对学生，有的两者都针对。还有课程结构较为松散，主要用于学习及培训等方面，应用领域有待扩充；课程资源的自我生长、扩充性不够。

国外的微课趋于成熟多样化，微课管理系统集成于其教学系统内，以学校的方式进行管理，跨校园微课比赛到较为困难。

### 1.4 论文的组织结构

第一章 绪论，本章首先简单介绍该课题的选题背景和意义，然后介绍了该系统在国内外的发展状况。

第二章 系统运用技术，本章对系统开发过程中所使用到的知识以及技术进行简单的介绍和说明。

第三章 需求分析，本章首先介绍了课题的业务背景，并根据需求捕获了系统的参与者，并且得到了参与者在高校微课比赛管理系统中的业务需求，然后再分析了业务实体和业务架构，再后捕获了系统的非功能需求。

第四章 系统分析，本章首先分析出了系统的总体用例，然后抽取了几个关键用例进行描述，然后捕捉了高校微课比赛管理系统的领域模型，完成了系统的架构设计，最后得到了分析模型，设计了实体类及服务接口。

第五章 系统的设计及实现，本章首先根据分析模型设计了数据库，然后分别从界面原型，实现原理和关键代码实现了高校微课比赛管理系统中的关键用例，最后介绍部分测试用例及测试结果。

### 1.5 本章小结

综上所述，本章首先详细的介绍了高校微课比赛管理系统的课题研究背景及意义，然后根据本课题研究了国内外的不同发展现状，最后根据该系统叙述了本次论文的组织结构。

## 2 系统运用技术

### 2.1面向对象的分析及设计

OOA的主要任务是根据需求建立一个标准、完美的面向对象分析模型。在其过程中，项目组通过详细分析其业务获得理想的模型，重点是描述需要解决的问题——称为“分析模型”[2]。

本文使用了OOA和OOD，从需求当中提取实体对象，然后根据需求提炼用例，最后实现用例，完成系统开发。

### 2.2 开发工具

2.2.1 MySQL

MySQL是一个开源的关系数据库管理系统（RDBMS）。使用结构化查询语言（SQL）进行数据库管理[14]。

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" \t "_blank)，由瑞典MySQL AB公司开发，目前属于[Oracle](https://baike.baidu.com/item/Oracle" \t "_blank)旗下产品。MySQL 是最流行的[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" \t "_blank)之一，在 WEB 应用方面，MySQL是最好的[RDBMS](https://baike.baidu.com/item/RDBMS/1048260" \t "_blank)(Relational Database Management System，关系数据库管理系统) 应用软件。

MySQL是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728" \t "_blank)的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是[开放源码](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%BA%90%E7%A0%81/7176422" \t "_blank)这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库[4]。

由于其社区版的性能卓越，搭配[PHP](https://baike.baidu.com/item/PHP" \t "_blank)和[Apache](https://baike.baidu.com/item/Apache" \t "_blank)可组成良好的开发环境。

MySQL的主要特点包括语法突出、可以修改/添加/删除域和记录、可以SQL语句脚本等。

MySQL优点：

（1）使用完全多线程，支持多处理器[15]。

（2）列类型：如float、double、char、varchar等。

（3） 通过类库以最快的速度实现SQL函数库，且初始化后无内存非配和漏洞。

（4）支持group by和order by子句，支持聚合函数。

（5）有缺省值。

2.2.2 IDEA

1. idea是一个相对较新的Java IDE。它是我所使用过的（我用过大部分Java开发环境）Java开发环境中最为有用的一个。高度优化的idea使普通任务变得相当容易，美观的主题搭配也是我喜欢的一大原因。
2. idea对各种拓展类型都有良好的支持，可以很方便的打开各种类型的文件而无需切换工具。
3. idea自带数据库可视化工具，且支持多种数据库，无需使用额外的数据库管理工具。
4. idea强大的代码错误检查，运行代码检查以报告大量的代码中潜在的错误点。
5. maven支持，使用maven可以很方便的管理你的项目依赖。

2.2.2 vs code

1. 开源免费，微软出品，有质量保证。Vs code一发布就广受关注，本身品质也非常高。介于编辑器与ide之间，快速高效。
2. 丰富的插件，通过插件，你可以让你的vs code变得更加美观与强大，通过拓展，甚至可以支持java开发环境，maven支持以及数据库可视化工具支持。

## 3 需求分析

### 3.1业务背景

#### 3.1.1高校微课比赛管理系统的定位

微课为一种新的网络学习资源形式在国内外迅速发展起来，微课比赛管理系统致力于建立一个公开，公平，公正的比赛平台，让评分有理可依，对比赛的给过展示，比赛的作品展示也更加的便利，为更多的教师同行创造了一个全新的技术交流平台，也为求知者营造了一片知识的沃土，比赛的创办有利于教师水平的提高，教师一直是走在时代前沿的岗位，在互联网迅速崛起的时代，教师团体也应该紧跟时代的步伐，为时代的未来青年灌输先进的思想。

“微课”的核心组成内容是课堂教学视频（课例片段），同时还包含与该教学主题相关的教学设计、素材课件、教学反思、练习测试及学生反馈、教师点评等辅助性教学资源，它们以一定的组织关系和呈现方式共同“营造”了一个半结构化、主题式的资源单元应用“小环境”。因此，“微课”既有别于传统单一资源类型的教学课例、教学课件、教学设计、教学反思等教学资源，又是在其基础上继承和发展起来的一种新型教学资源。

“微课”的优点明显，已经成为各个学校教学中比较重要的辅助手段，为了提高教师微课制作的热情，更广泛的推广优秀的微课，宣传微课的教学效果，因此，需要一个完善的微课比赛系统来组织微课比赛，提高比赛的效率和关注度。

高校微课比赛管理系统与高校信息管理系统有所关联，高校微课比赛管理系统可以从高校信息管理系统中获取信息。两个系统是相互协同工作，从而完成比赛的信息化管理。如图3-1：

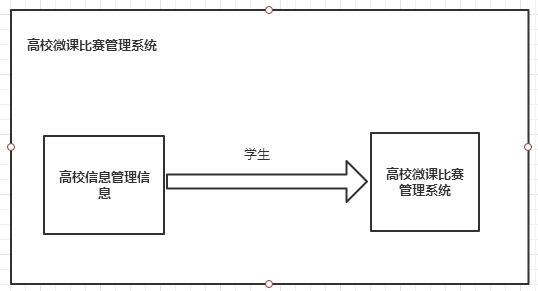
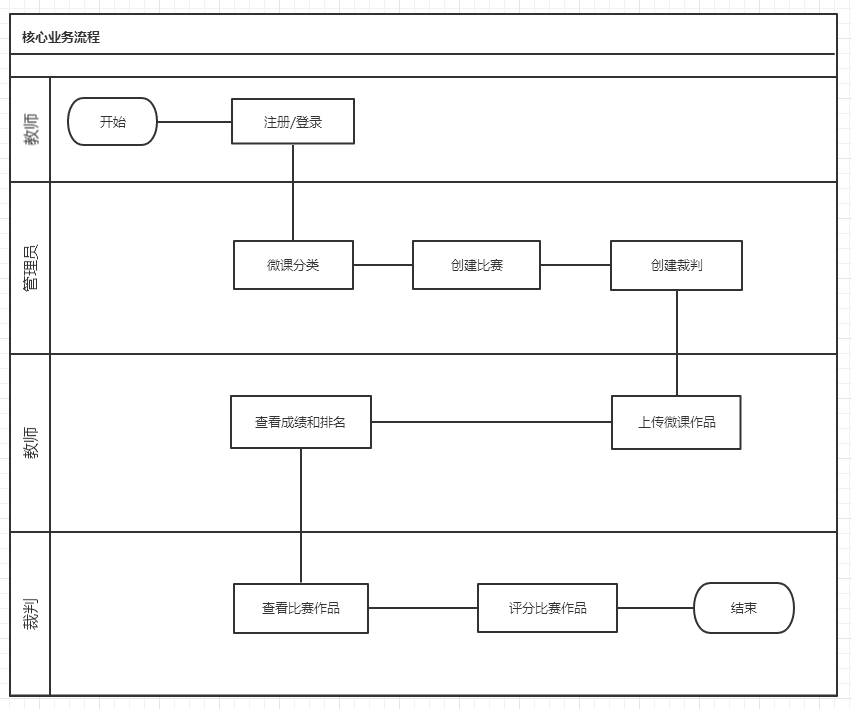


图3-1 高校微课比赛管理系统与子系统的关系图

#### 3.1.2高校微课比赛管理系统业务模型

（1）业务视图

高校微课比赛管理系统是以微课作品比赛管理为主线贯穿整个比赛的信息化管理，从参赛者到管理员对系统的分析，都始终是以微课作品管理为驱动的。高校微课比赛管理系统需求的业务流程图如图3-2：

图3-2高校微课比赛管理系统的业务流程图

### 3.2业务主角

根据高校微课比赛管理系统的业务流程及其用户需求，可以分析出业务主角分为两种：

（1）教师：教师相当于高校微课比赛管理系统的的普通用户，他们使用该系统参加高校微课比赛，并查询自己参赛作品所得分数和排名。

（2）裁判：裁判不需要分任何的级别。他们使用该系统主要是查询高校微课比赛的参赛作品，并对作品进行打分、评论。

（3）管理员：管理员是高校微课比赛管理系统拥有较高权限的用户，可以对高校、裁判、作品等进行管理。

### 3.3业务分析

高校微课比赛管理系统提供用户基本信息、作品信息、高校信息。从总体上讲，高校微课比赛管理系统的功能性需求，可以从管理员和一般用户两个层面来进行分析。

#### 3.3.1管理员层面

（1）高校信息的查询

管理人员可以对各个高校的信息进行查询，并可以对该校信息进行增删改查等操作。

（2）比赛管理

管理员可以对高校微课比赛管理系统的比赛进行修改，根据实际情况可对比赛进行增加、删除等操作。

（3）裁判管理

工资核算人员可以在高校微课比赛管理系统中对部门进行管理，包括对部门进行增加、删除、修改以及查询部门的基本信息。

（4）微课分类管理

管理人员可以在高校微课比赛管理系统中管理微课分类信息，对比赛中新出现的微课类型进行增加，对已经淘汰的类型进行删除。

#### 3.3.2 裁判层面

（1）个人信息的查询

裁判可以对个人的信息进行查询，但是不能对信息进行修改。

（2）密码修改

裁判对本人登录系统的密码进行修改，修改时要进行初始密码的验证，才可以进行修改，修改密码之后，需要重新登录系统，不然访问其他功能会出现报错。

（2）微课比赛作品查询

裁判可以查询比赛中的所有参赛作品。

（3）对微课作品打分评价

裁判可以查询比赛中的所有参赛作品，对参赛的微课作品进行打分评价。

#### 3.3.3 教师层面

（1）个人信息的查询

教师可以对个人的信息进行查询，但是不能对信息进行修改。

（2）密码修改

教师对本人登录系统的密码进行修改，修改时要进行初始密码的验证，才可以进行修改，修改密码之后，要重新登录系统，不然访问其他功能会出现报错。

（3）微课比赛作品上传

教师可以查询比赛，并根据比赛上传自己的参赛的微课作品。

（4）微课作品排名查询

教师可以查询比赛中自己所参赛的所有参赛作品，对可以查询参赛的微课作品排名及裁判评价。

### 3.4业务实体分析

高校微课比赛管理系统在经过一步一步的管理后，最终生成一张微课比赛作品报表。这张微课比赛作品报表包括了才赛教师的基本信息、比赛信息、参赛作品等信息。高校微课比赛管理系统的开发目的之一就是管理系统信息化。所以，在高校微课比赛管理系统中对高校微课比赛的管理流程进行了提炼简化。微课比赛作品报表样表如表3-1：

表3-1微课比赛作品报表样表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 教师编号 | 姓名 | 作品名称 | 作品得分 | 排名 | 所属高校 | 上传时间 | 备注 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

### 3.5非功能需求

根据重庆工程学院的特殊要求，高校微课比赛管理系统具有以下技术需求：

（1）性能需求

系统实现安装简便，运行时尽量减少网络负荷，以达到所有用户同时使用系统；

本系统采用1080\*1920分辨率；

本系统要求对用户的操作响应及时，特别是查询、统计要采用优化的算法以满足网络的数据流量。

（2）安全性要求

保证网络安全以及数据和系统安全，与此同时通过安全体系的有效执行，使各种各样的信息收集、处理、控制、传送等能安全的进行。

用户通过密码登录，登录密码将不能显示在地址栏中和设为保存默认的密码。

用户权限控制：不同层次的用户拥有不同的权限，权限分为操作权限和数据权限。

（3）设计约束

界面简洁、直观，整体布局合理，不存在累赘的操作。对于错误和异常的提示信息要有导向性，避免出现开发工具自带的英文提示。

异常信息提示，必须要有清楚的中文说明。

（4）硬件约束

处理器：intel(R)Core(TM)i5-3230M CPU@2.60GHz 2.60GHz

内存：4.00GB

### 3.6本章小结

本章首先介绍了高校微课比赛管理系统的背景，并根据业务需求捕获了系统的参与者，并且得到了参与者在高校微课比赛管理系统中的业务需求，然后再分析了业务实体和业务架构，再后捕获了系统的非功能需求。

## 4 系统分析

### 4.1系统总体业务视图

根据对业务的分析，归类，得出了高校微课比赛管理系统的业务架构。高校微课比赛管理系统包括高校管理、比赛管理、裁判管理、微课作品分类管理等子系统。高校微课比赛管理系统架构视图如图4-1：

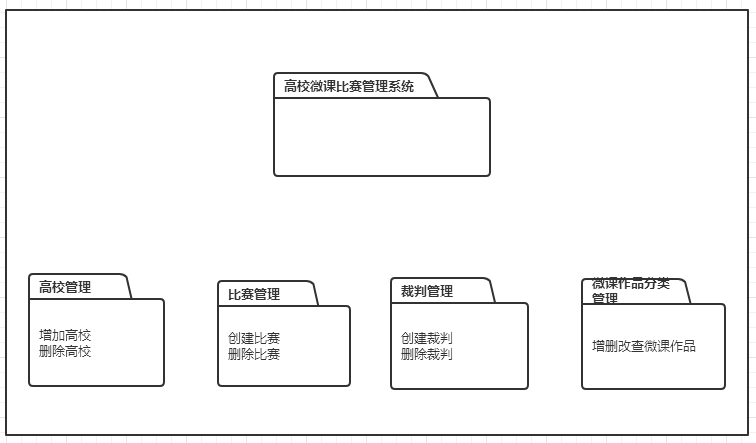


图4-1高校微课比赛管理系统架构视图

高校管理：高校信息管理是高校微课比赛系统中管理各高校信息的管理；

比赛管理：员工信息管理是高校微课比赛管理系统管理人员对各比赛的管理，可以对比赛进行增删改查等操作；

裁判管理：在高校微课比赛管理系统中，管理员可以进行创建、删除裁判等操作；

微课作品分类管理：在高校微课比赛管理系统中，管理员可以设置微课比赛中所有不同类型的微课比赛进行创建比赛，并对微课的分类进行删改查等操作。

### 4.2系统用例

我们将用户业务需求进行整理，归纳，总结，分析后，高校微课比赛管理系统的总用例图已经呈现。

用户信息管理包含了不同角色登录，不同的角色登录该系统将对个人的基本信息进行查询，可以修改初始登录系统的密码，密码修改成功后，要重新登录该系统。

比赛管理指针对于管理员开放，管理员可以在高校微课比赛管理系统当中对比赛裁判进行增删改查。

微课作品管理，是当管理员登录该系统的时候，可以设置高校微课比赛管理系统微课的基本参数，查询比赛中所有微课作品信息。

微课作品分类管理涵盖了不同角色的要求，当管理员登录该系统的时候，可以对比赛中所有微课作品设置他们的作品分类，然后可以查询系统所有分类的微课作品信息，也可以单独的进行查。

后面将依次对主要用例进行描述。

#### 4.2.1管理员角色用例图

管理员的相关用例如图4-2所示：

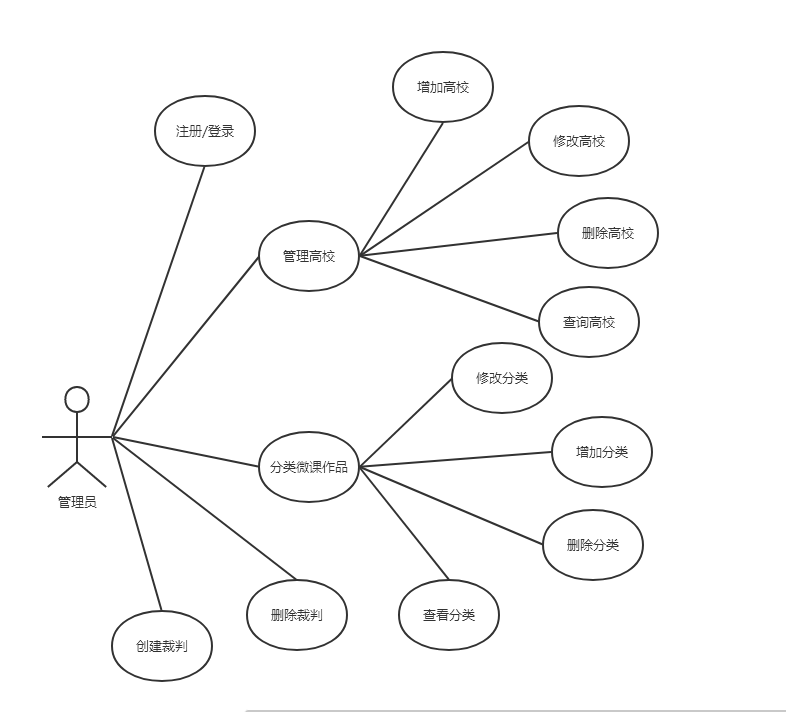


图4-2管理员用例图

管理员所实现的所有功能包含很多的用例，例如裁判管理模块中对裁判人员的增删改查等。那么管理员用例的具体描述如表4-1所示：

表4-1工资核算人员用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 子系统 | 用例名称 | 用例功能描述 | 备注 |
| 高校信息管理 | 高校信息查询 | 管理员可以查询各个高校详细信息，可进行增删改查操作 |  |
| 裁判管理 | 裁判查询 | 对比赛中的各个裁判进行增删等操作 |  |
| 微课作品分类管理 | 微课作品查询 | 对比赛中的各个参赛作品进行查询，进行分类，可对分类进行增删改查操作 |  |
| 比赛管理 | 比赛查询 | 对已创建的比赛可进行查询，可进行增删等操作 |  |

#### 4.2.2裁判角色用例图

裁判用例图的相关的主要用例如图4-3所示：

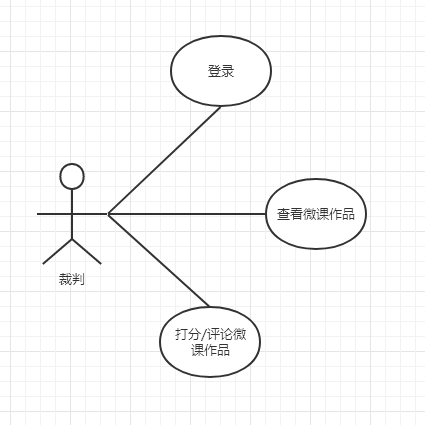


图4-3裁判用例图

裁判主要是对教师微课作品进行打分操作，下面主要介绍裁判用例的具体描述如表4-2所示：

表4-2裁判用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 子系统 | 用例名称 | 用例功能描述 | 备注 备注 |
| 个人用户信息管理 | 个人信息查询 | 员工可以查询个人的用户信息，但不能修改 |  |
| 修改密码 | 员工可以修改初始密码，密码修改后，要重新登录系统 |  |
| 微课作品管理 | 微课作品查询 | 裁判可以查询教师当场微课比赛的作品 |  |
| 微课作品打分 | 对教师进行微课比赛的微课作品给予打分、评论 |  |

#### 4.2.3教师角色用例图

教师用例图的相关的主要用例如图4-4所示：

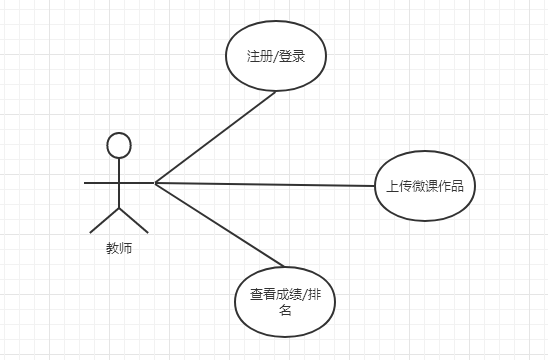


图4-4教师用例图

教师主要是对微课比赛作品进行上传操作，下面主要介绍教师用例的具体描述如表4-3所示：

表4-3教师用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 子系统 | 用例名称 | 用例功能描述 | 备注 备注 |
| 个人用户信息管理 | 个人信息查询 | 员工可以查询个人的用户信息，但不能修改 |  |
| 修改密码 | 员工可以修改初始密码，密码修改后，要重新登录系统 |  |
| 微课作品管理 | 微课作品上传 | 教师可以根据参加的微课比赛进行上传作品 |  |
| 微课作品排名 | 教师通过高校微课比赛管理系统的查看自己的作品分数排名 |  |

### 4.3 比赛管理用例描述

比赛管理是高校微课比赛管理系统中管理员对比赛基本信息的管理，其主要的功能就是对比赛进行增删改查。

参与者：管理员

基本流：

（1）新增比赛，填写比赛的基本信息，并赋予相应的权限（基本信息包括：比赛名称、参赛时间、参赛人数、参赛人姓名、性别、出生日期、所属高校、联系电话、住址）；

（2）修改比赛的信息；

（3）删除比赛的信息；

（4）根据比赛名称查询参赛作品信息。

替代流：当用户登录失败时、数据库操作失败时、用户操作不符合条件的操作时，以及程序运行出现异常时将提示响应失败信息。

前置条件：用户成功登录系统，符合权限要求。

后置条件：保存后的微课比赛的信息，将会自动录入到数据库当中。

特殊需求：无

### 4.4 微课作品上传用例描述

微课作品上传是高校微课比赛管理系统中教师对于微课比赛参赛的信息进行的录入过程。

参与者：教师

基本流：

（1）选择所微课比赛；

（2）上传自己的微课比赛作品。

替代流：当用户登录失败、数据库操作失败、操作不符合条件的操作时，程序运行出现异常时将提示响应失败信息。

前置条件：用户成功登录系统，符合权限要求。

后置条件：保存上传微课作品的各项信息，将自动的录入数据库当中。

特殊需求：无

### 4.5微课作品打分评论用例描述

微课作品打分评论是高校微课比赛管理系统中裁判对参赛教师的微课作品进行打分的操作，在查询参赛教师的微课作品信息的时候，可以当场比赛的所有微课作品信息进行查询，并且对微课作品进行打分及评论。

参与者：裁判

基本流：

（1）查询比赛的基本信息；

（2）根据比赛的信息查询参赛作品的信息；

（2）裁判根据上传的微课水平进行评论打分。

替代流：当用户登录失败时、当数据库操作失败时、当用户操作不符合条件的操作时，程序运行出现异常时将提示响应失败信息。

前置条件：裁判成功登录系统，符合权限要求。

后置条件：保存微课作品打分及评论的各项信息，将会自动的录入到数据库当中。

特殊需求：无

### 4.6域模型

根据用例说明，可以得出高校微课比赛管理系统的域模型，如图4-4：

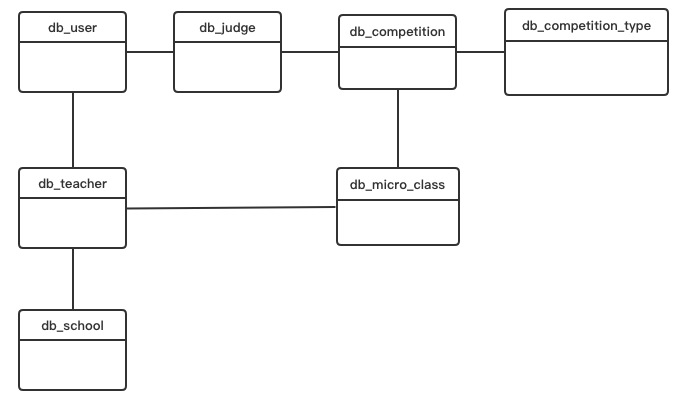


图4-4高校微课比赛管理系统域模型

db\_user:员工信息域，记录了员工的基本信息；

db\_judge：裁判信息域，记录了裁判的基本信息；

db\_competition：比赛信息域，记录了比赛的基本信息；

db\_competition\_type：比赛类型域，记录了比赛的类型信息；

db\_teacher：教师信息域，记录了教师的基本信息。

db\_school：高校信息域，记录了高校的基本信息。

### 4.7架构设计

#### 4.7.1架构设计思想

SSM框架技术应用于信息技术上。由SPRING MVC、SPRING和MYBATIS三个开源框架的整合而成的SSM框架，经常作为简单项目的开源框架，它是一个标准的MVC模式。该模式将系统划分为view层、controller层、service层、DAO四层。用途：

（1）SPRING MVC用于请求的转发和视图管理；

（2）SPRING用于业务的对象管理；

（3）MYBATIS作为数据对象的持久化引擎。

springmvc原理是用户端发出要求到前端控制器；由控制器查询处理器，找寻处理请求的控制器；然后使用控制器调用业务逻辑处理，返回数据显示视图；再用前端控制器找到视图解析器，用来找到指定的视图；最后显示结果到客户端[17]。

spring：使用最多的是控制反转，用来装载bean之后，对类不用进行初始化。

mybatis：对JDBC的封装。它通过配置文件关联到实体类间，在配置文件中配置所有的SQL语句的映射。交互时，通过SQLSESSIONFACTORY得到一个SQLSESSION来执行SQL命令。

#### 4.7.2架构设计原则

系统设计原则如下：

（1）实用性原则

高校微课比赛管理系统必须最大限度地满足实际工作的需要，要充分的考虑用户的使用该系统的实用性，是用户能够方便、快捷地进行业务办理。

（2）先进性原则

在技术上采用先进、成熟的开发技术。

（3）安全、可靠性原则

系统的数据库设计要安全可靠，不能让网页记住密码或使密码显示在地址栏中。

（4）可扩展性和可维护性

为以后的发展，设计系统应具备好的可扩展性和维护性。设计软件要使软件模块化，并提供配置模块的工具，使系统能够很好的配置，适应不同的情况。

#### 4.7.3系统结构模型

SSM框架包括四层，如图4-5:

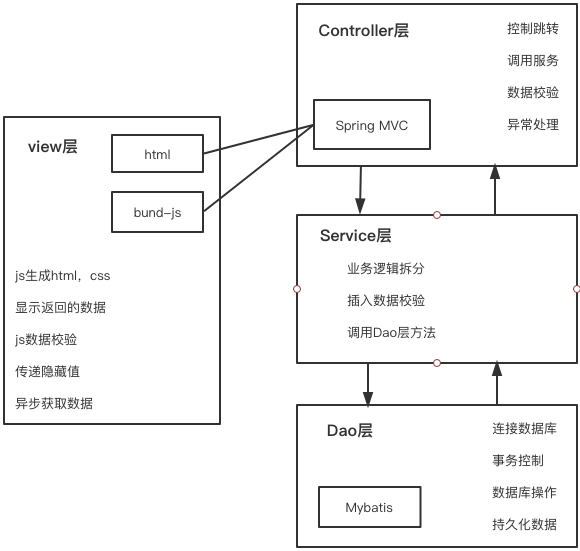


图4-5高校微课比赛管理系统结构模型

（1）controller控制层

Controller控制层负责业务模块流程控制，利用Service服务层接口来控制业务流程，Controller层使用spring-boot的@Controller注解实现，针对不同的业务逻辑，用不同的控制器。

（2）service服务层

Service服务层负责应用逻辑设计。首先设计接口，再实现他的类，然后使用@Servers注解。与此同时可以在应用中调用Service接口来进行业务处理。其业务处理，具体要使用已定义的DAO层的接口。

（3）DAO持久层

DAO数据持久层，进行数据库的连接与操作，设计接口，通过@Mapper注解sql实现数据库操作，最后调用接口来进行数据业务处理。

（4)view层

View视图层与控制层联系紧密，视图向控制器发起HTTP通信。主要负责数据的传递。

### 4.8网络结构

采用B/S架构，使用网络结构来构建如图4-6：

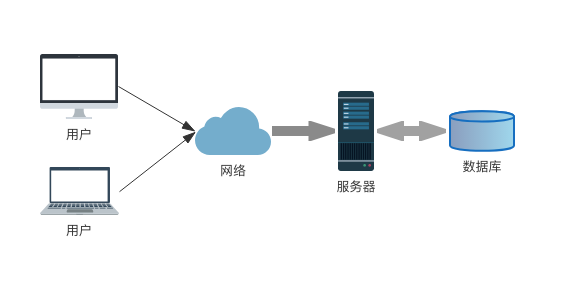


图4-6高校微课比赛管理系统网络结构图

1．成本低

用户只需要一台装有浏览器且连入网络（本地局域网或互联网）的计算机即可访问系统。

2．易于维护

该架构没有客户端程序，降低了系统的维护难度，降低了成本。

### 4.9 分析模型

#### 4.9.1 比赛管理用例

比赛管理需求由各个不同的层次协调工作并完成相应的业务功能，比赛管理的时序图如图4-7：

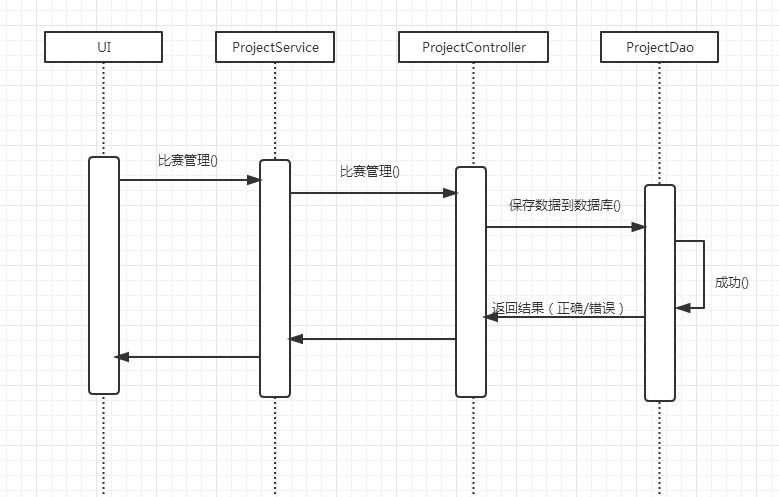


图4-7比赛管理时序图

比赛管理，当管理员对比赛信息进行管理时，所做的增删改查操作，都要进入数据库进行判断操作，数据库再进行返回数据结果是否成功，如果不成功，再次进行重新的输入。比赛信息所用到的具体属性如图4-8：



图4-8比赛管理的对象属性

#### 4.9.2 微课作品上传用例

微课作品上传是教师参赛的关键步骤，微课作品上传的时序图如图4-9：

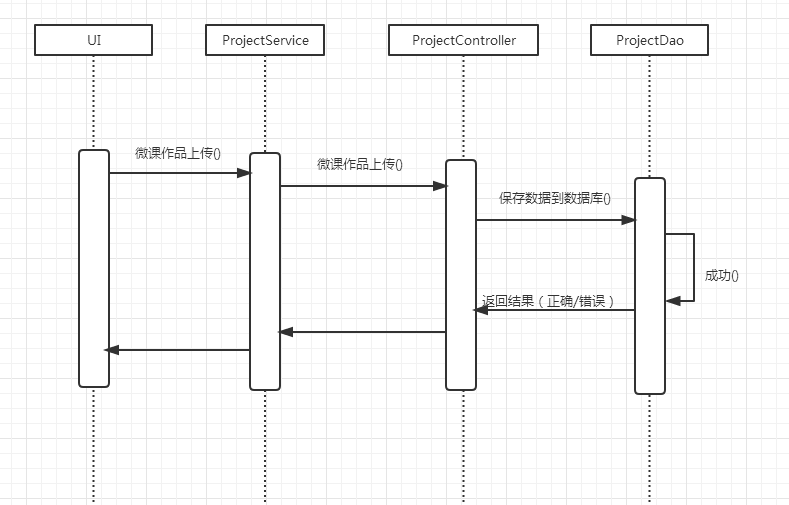


图4-9微课作品上传时序图

微课作品上传，微课作品上传需要对身份进行校验，数据格式进行认证，然后会返回一定的提示，显示数据是否正确。如图4-10微课作品所用到的具体属性：



图4-10微课作品上传的对象属性

#### 4.9.3 微课作品打分评论用例

微课作品打分，裁判可以对微课作品进行打分，需要对数据格式，有效性个进行验证。微课作品打分评论的时序图如图4-11：

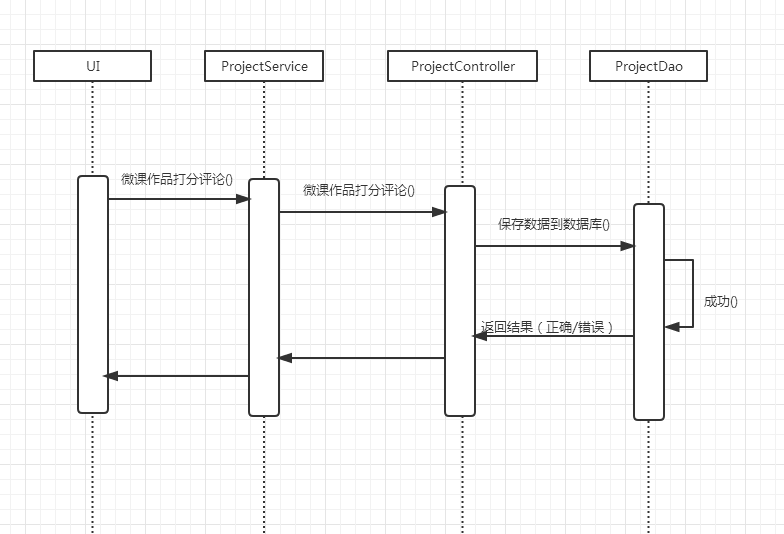


图4-11微课作品打分评论时序图

查询分数，首先教师可以查询自己所参赛的分数，如果裁判已经打分，则显示分数，管理员也可以根据教师信息，比赛信息来查看分数以及微课作品。查询分数所用到的具体属性如图4-12：

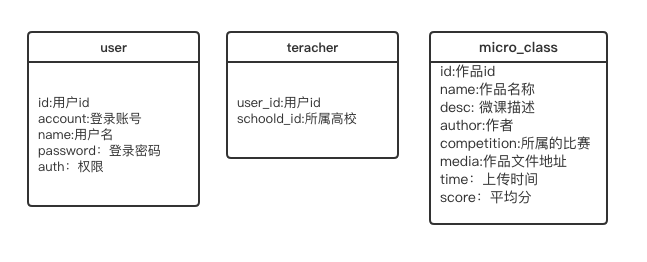


图4-12微课作品打分评论的对象属性

### 4.10本章小结

本章首先分析出了系统的总体用例，然后抽取了几个关键用例进行描述，然后捕捉了微课比赛系统的领域模型，完成了系统的架构设计，最后得到了分析模型，设计了实体类及服务接口

## 5 系统的设计及实现

### 5.1数据库设计

#### 5.1.1数据库设计原则

数据库的设计是系统开发的基础，不同的设计方案有不同的数据库设计，数据库设计有以下约束[11]：

一致性约束：对数据进行分析和设计，调整好数据之间的关系，保证数据库设计的一致性。

完整性约束：指数据的正确性和相容性。防止用户在使用数据库时向数据库中添加不正确的数据。

安全性约束：保护好数据，防止非法用户攻击数据，造成数据泄露、更改、破坏，用户在进行登录时要严格的进行严格的登录控制。

可伸缩性约束：数据库结构设计应充分考虑系统扩展的需要、向异构数据库存移植的需要，具有良好的可扩展性、伸缩性和较小的冗余度。

规范化约束：数据库设计应遵循规范化原则，规范化分为六级。在规范化级别当中并不是规范化程度越高就越好。一般情况下，使用第三范式。

#### 5.1.2 数据库设计模型

分析系统需求的主要目的是找出它们之间的关系。它的缺点是对数据库的存储以及数据之间关系的表达。在数据库设计中使用的E-R图来进行弥补。[11]

（1）概念数据库设计

系统概念数据库设计包括两方面：第一数据库的概念模式，反映了所有用户对于现实世界的抽象，形成整体逻辑结构。第二事务设计，检查数据库操作任务，并形成事务管理。

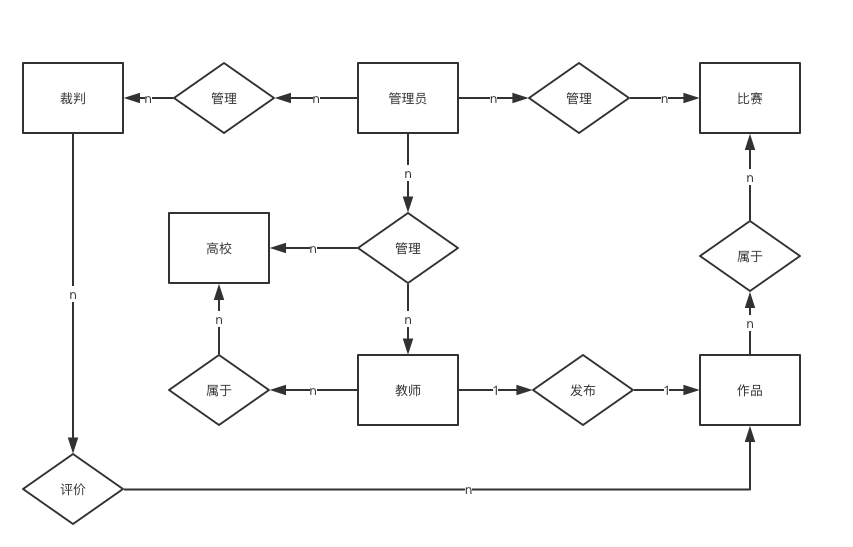
（2）逻辑数据库设计

主要任务是把概念数据库转换为逻辑数据库。逻辑数据库设计依赖于具体的数据库系统。该系统使用的关系数据库为MySQL。而关系模式的集合就是数据库。

（3）物理数据库设计

目的是为关系模式选取存储方法，提高运行速度。根据分析系统的数据处理结果，分析出了市场需求及其相关系统数据库表的关系。

#### 5.1.3 实体-联系图



#### 5.1.4 数据库逻辑模型

其主要的数据库逻辑模型如图5-1所示：

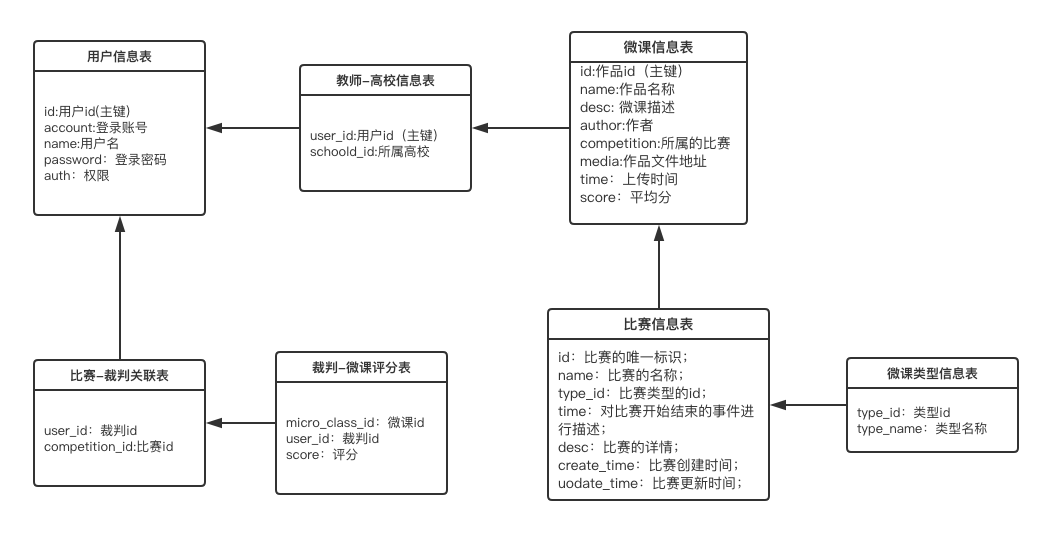


图5-1微课比赛系统数据库逻辑模型

具体描述如下所示：

表总汇

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 表名 | 含义 | 功能描述 | 主要信息 |
| 1 | user | 用户信息表 | 存放用户详细信息 | 用户名 |
| 2 | teacher | 教师信息表 | 教师学校关联信息 | 用户id，学校id |
| 3 | school | 学校信息表 | 学校的基本系信息 | 学校id，名称 |
| 4 | competition | 比赛信息表 | 比赛的基本信息 | 名称，标题，分类 |
| 5 | type | 比赛分类表 | 比赛分类信息 | 分类名称 |
| 6 | judge | 裁判信息表 | 裁判指定信息 | 裁判id，比赛id |
| 7 | micro\_class | 微课信息表 | 存放作品基本信息 | 作者，文件，分数 |

User用户表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **小数位** | **标识** | **主键** | **允许空** | **默认值** | **说明** |
| 1 | id | int | 4 | 0 | 是 | 是 | 否 |  |  |
| 2 | account | nvarchar | 32 | 0 | 是 |  | 否 |  |  |
| 3 | password | nvarchar | 32 | 0 |  |  | 否 |  |  |
| 4 | sex | int | 1 |  |  |  | 是 |  |  |
| 5 | auth | int | 1 |  |  |  | 否 | 1 |  |

teacher教师表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **小数位** | **标识** | **主键** | **允许空** | **默认值** | **说明** |
| 1 | User\_id | int | 4 | 0 | 是 | 是 | 否 |  |  |
| 2 | University | int | 4 | 0 | 是 |  | 否 |  |  |

university高校表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **小数位** | **标识** | **主键** | **允许空** | **默认值** | **说明** |
| 1 | id | int | 4 | 0 | 是 |  | 否 |  |  |
| 2 | name | Varchar | 32 | 0 | 是 |  | 否 |  |  |

micro-class微课表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **小数位** | **标识** | **主键** | **允许空** | **默认值** | **说明** |
| 1 | ID | int | 4 | 0 | 是 | 是 | 否 |  |  |
| 2 | name | nvarchar | 32 | 0 |  |  | 否 |  |  |

competition比赛表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **小数位** | **标识** | **主键** | **允许空** | **默认值** | **说明** |
| 1 | ID | int | 4 | 0 | 是 | 是 | 否 |  |  |
| 2 | name | int | 4 | 0 |  |  | 否 |  |  |
| 3 | title | varchar | 32 | 0 |  |  | 否 |  |  |
| 4 | desc | varchar | 255 | 0 |  |  | 否 |  |  |
|  | time | varchar | 64 | 0 |  |  | 否 |  |  |

judge裁判表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **小数位** | **标识** | **主键** | **允许空** | **默认值** | **说明** |
| 1 | ID | int | 4 | 0 | 是 | 是 | 否 |  |  |
| 2 | competition | nvarchar | 64 | 0 | 是 |  | 否 |  |  |
| 3 | User\_id | int | 4 | 0 | 是 |  | 否 |  |  |

media作品表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **小数位** | **标识** | **主键** | **允许空** | **默认值** | **说明** |
| 1 | ID | int | 4 | 0 | 是 | 是 | 否 |  |  |
| 2 | User\_id | int | 4 | 0 |  |  | 否 |  |  |
| 3 | competition | int | 4 | 0 |  |  | 否 |  |  |
| 5 | url | nvarchar | 255 | 0 |  |  | 否 |  | 链接 |

judge-micro-class评分表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **小数位** | **标识** | **主键** | **允许空** | **默认值** | **说明** |
| 1 | ID | int | 4 | 0 | 是 | 是 | 否 |  |  |
| 2 | Mc\_id | int | 4 | 0 |  |  | 否 |  |  |
| 3 | judge\_id | int | 4 | 0 |  |  | 否 |  |  |
| 5 | score | double | 4 | 0 |  |  | 否 |  |  |
| 6 | commend | nvarchar | 255 | 0 |  |  | 否 |  |  |

### 5.2系统界面原型

本系统采用GUI设计原则：一致性、简易性 、规范性、美观与协调性等规范原则；界面非常直观、对用户透明：用户能够自己使用该系统，不需要任何的指导[12]。

界面设计必须经过最后的详细确认才能完成，登录界面原型如图5-2，系统主界面原型图5-3：

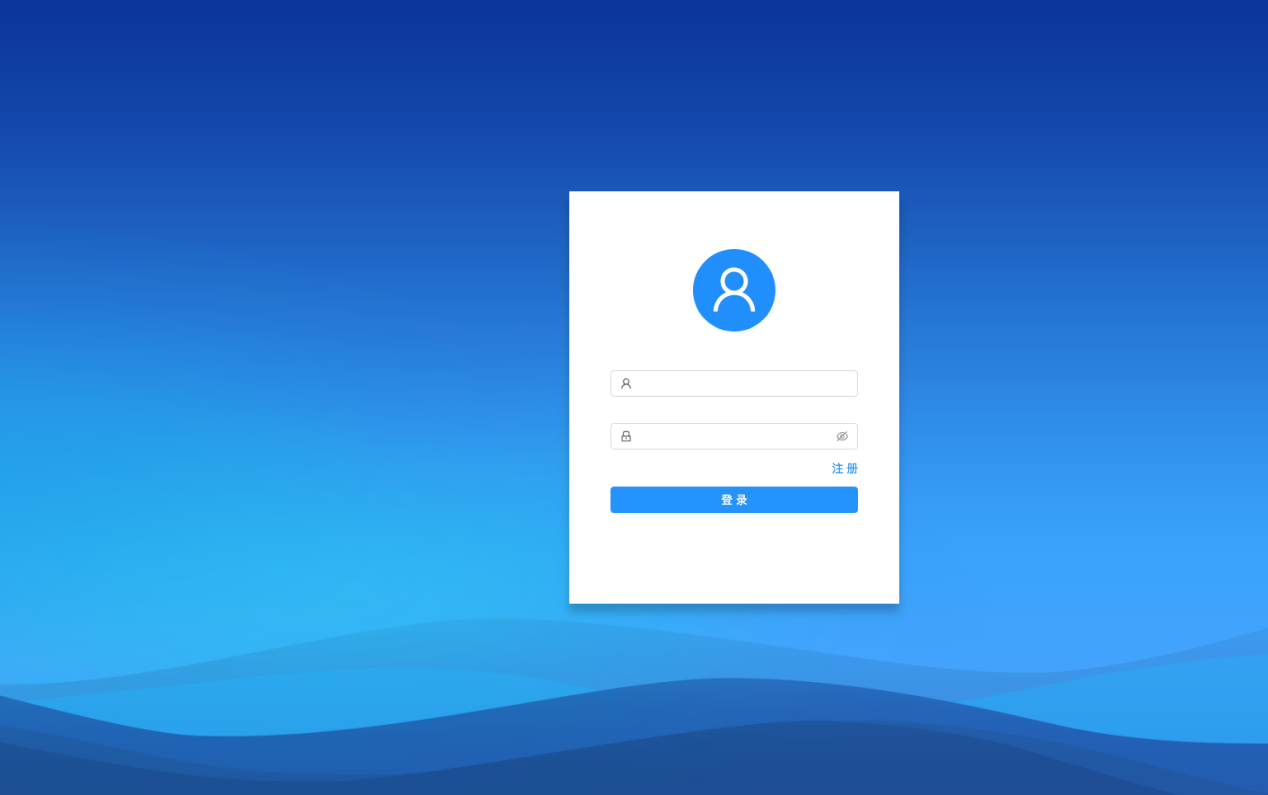


图5-2登录界面原型

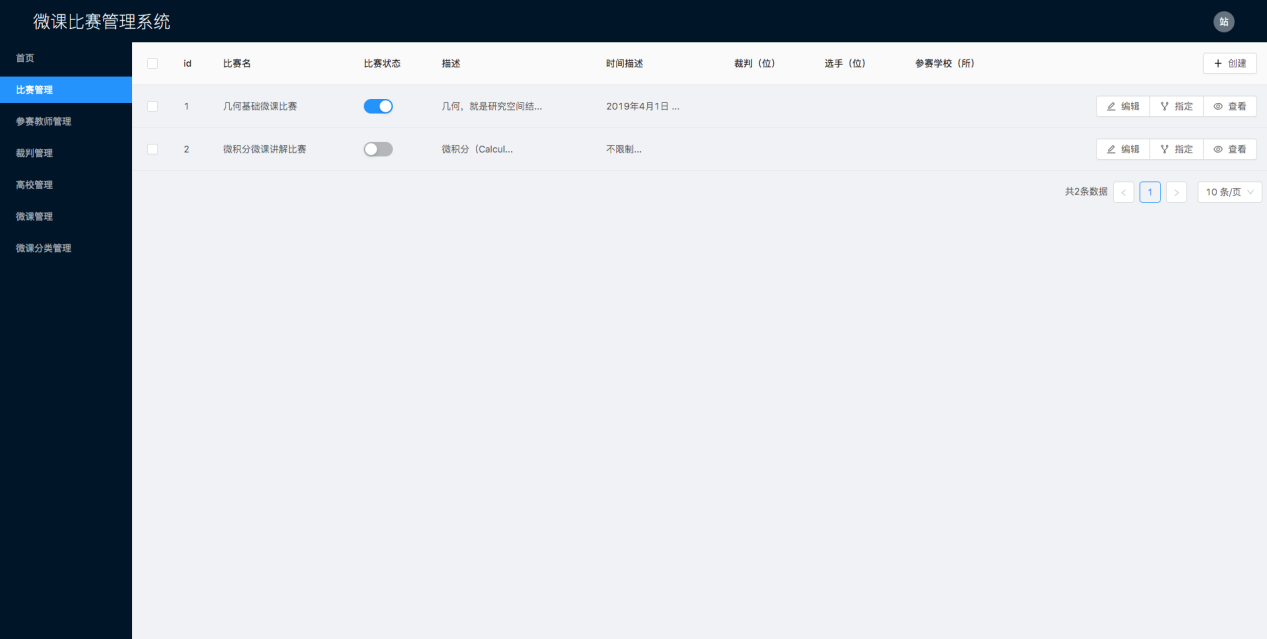


图5-3系统主界面原型

### 5.3 比赛管理用例

#### 5.3.1界面原型

管理员登录系统，对比赛以及用户信息进行相应的管理。例如比赛信息的增删改查。微课比赛系统信息维护需求时的界面原型如图5-4：



图5-4员工信息维护需求时的界面原型

界面中的信息维护，具体包括了用户账号、角色、真实姓名、用户性别、联系电话。可以根据用户姓名查询具体的用户，新增加用户，对用户信息编辑和删除。

#### 5.3.2实现原理

当管理员进入教师信息维护功能中，分页显示教师的信息，将为管理员提供删除操作，提示是否确认删除，确认后返回教师列表页面；提供修改操作，进入修改用户信息页面，账号密码不能修改，因为已经在修改密码处提供，可以重新上传教师信息资料，点击保存后提示保存成功，然后返回员工列表页面；提供增加员工按钮，进入添加员工页面，包括账号密码等个人信息。

#### 5.3.3关键代码

/\*\*

\* 查教师列表

\*/

@GetMapping("/user")  
public HashMap user(@RequestParam HashMap<String, Object> hashMap, String offset, String limit){  
 if(offset!=null){  
 hashMap.put("offset",Integer.parseInt(offset));  
 }  
 if(limit!=null){  
 hashMap.put("limit",Integer.parseInt(limit));  
 }  
 return new HashMap<String,Object>(){{  
 put("user",userMapper.find(hashMap));  
 put("count",userMapper.count(hashMap));  
 }};  
}

@Select("<script>select user\_id,type,account,user\_name,sex,tel,email" +  
 "<if test='type==1'>,university\_name</if>" +  
 "<if test='competition\_id!=null'>,judge\_competition\_id</if>" +  
 " from user " +  
 "<if test='type==1'>left join teacher on teacher\_user\_id=user\_id left join university on university\_id=teacher\_university\_id</if>"+  
 "<if test='competition\_id!=null'>left join judge on judge\_user\_id=user\_id and judge\_competition\_id=#{competition\_id}</if>"+  
 "where true " +  
 "<if test='type!=null'>and type=#{type}</if>" +  
 "<if test='account!=null'>and account=#{account}</if>" +  
 "<if test='password!=null'>and password=#{password}</if>" +  
 "<if test='limit!=null'>limit #{limit} <if test='offset!=null'>offset #{offset}</if> </if>" +  
 "</script>")  
List<HashMap> find(HashMap map);  
@Select("<script>select count(1) from user where true " +  
 "<if test='account!=null'>and account=#{account}</if>" +  
 "<if test='password!=null'>and password=#{password}</if>" +  
 "<if test='type!=null'>and type=#{type}</if>" +  
 "</script>")  
int count(HashMap map);

### 5.4 微课作品上传用例

#### 5.4.1 界面原型

界面原型如图5-5：

图5-5作品上传界面原型

#### 5.4.2实现原理

教师登录系统，进入系统主界面，banner上有微课比赛的详细说明，滑动比赛列表选择比赛，点击报名参赛参加比赛，然后比赛会出现在我的比赛列表内，点击上传按钮上传微课视屏，实时展示进度百分比，上传成功提供预览按钮，点击预览播放微课比赛视屏。

#### 5.4.3关键代码

//后台上传接口

@PostMapping("/competition/{id}/upload")

public void upload(MultipartFile file,HttpServletRequest req,@PathVariable int id) throws IOException {

int user\_id=(int)req.getAttribute("this\_user\_id");

File path = new File(ResourceUtils.getURL("classpath:").getPath());

if(!path.exists()) path = new File("");

String media="/upload/"+new Date().getTime()+"&&"+user\_id+"&&"+id+"&&"+file.getOriginalFilename();

mcMapper.update(new HashMap<String,Object>(){{

put("media",media);

put("user\_id",user\_id);

put("competition\_id",id);

}});

file.transferTo(new File(path.getAbsolutePath()+"/static"+media));

}

//前端上传组建

import React from "react";

import {Layout,Divider,Empty, Button, message, Upload, Modal} from "antd";

import UserAvatar from "@/components/avatar";

import banner from '@/images/banner.jpg';

import {get,ip} from '@/utils/request'

import bg from '@/images/true.jpeg'

class MyUpload extends React.Component{

state={}

uploadProps = {

name: 'file',

action: `${ip}/teacher/competition/${this.props.competition\_id}/upload`,

headers: {

authorization: localStorage.getItem('token'),

},

beforeUpload:()=>{

this.setState({

loading:true

})

},

onChange:info=> {

if (info.file.status === 'uploading') {

if(info.event){

this.setState({progress:parseFloat(info.event.percent)})

}

}

if (info.file.status === 'done') {

this.props.onSuccess()

this.setState({

loading:false

})

message.success(`${info.file.name} file uploaded successfully`);

} else if (info.file.status === 'error') {

message.error(`${info.file.name} 上传失败`);

this.setState({

loading:false

})

}

},

};

render(){

return<Upload {...this.uploadProps} ><Button loading={this.state.loading} ghost size="small" icon="upload" >

{this.state.loading?parseInt(this.state.progress)+'%':this.props.title}

</Button></Upload>

}

}

class Competition extends React.Component{

state={competition:[]};

componentDidMount(){

get("/teacher/competition").then(json=>{

console.log(json)

this.setState({competition:json.data.competition})

})

}

handleJoin=id=>{

get(`/teacher/competition/${id}/join`).then(json=>{

if(json.code===0){

message.success("报名成功");

this.props.onSuccess();

}else{

message.error("报名失败");

}

})

}

render(){

return<div>

{this.state.competition.length===0&&<Empty description={"很遗憾，目前还没有比赛，耐心等待比赛发布"} />}

<div style={{width:'100%',overflowX:'auto',whiteSpace:'nowrap'}}>

{this.state.competition.map(com=>{

return <div key={com.competition\_id} style={{width:500,padding:20,whiteSpace:'normal',color:'#fff',margin:20,background:'#0d143c',display:'inline-block'}}>

<div style={{fontSize:26,textAlign:'center',fontWeight:'lighter'}}>【{com.type\_name}】{com.competition\_name}</div>

<div style={{textAlign:'center',color:'rgba(255,255,255,.6)',paddingBottom:20}}>时间：{com.competition\_time}</div>

<pre style={{whiteSpace:'normal',lineHeight:2,height:140,overflow:'hidden'}}>{com.competition\_desc}</pre>

<div style={{textAlign:'right'}}><Button onClick={()=>{this.handleJoin(com.competition\_id)}} ghost>报名参赛</Button></div>

</div>

})}

</div>

</div>

}

}

### 5.5 微课作品打分评论用例

#### 5.5.1 界面原型

界面原型如图5-6：



图5-6裁判界面原型

#### 5.5.2实现原理

裁判登入系统，系统展示该裁判所有的比赛，当比赛处于完成状态时，裁判可以进入比赛进行作品预览、打分、评论。

#### 5.5.3关键代码

/\*\*

\* 根据裁判信息，获取比列表

\* @param request 教师信息

\* @return 比赛列表

\*/

@GetMapping("/competition")

public List<HashMap> competition(HttpServletRequest request){

int judge=(int)request.getAttribute("this\_user\_id");

return judgeMapper.findCompetition(new HashMap<String,Object>(){{

put("user\_id",judge);

}});

}

/\*\*

\* 查询裁判的比赛列表

\* @param map 裁判信息

\* @return 比赛列表

\*/

@Select("<script>"+

"select type\_name,judge\_id,competition\_id,competition\_name,competition\_time,competition\_state,judge\_id,(select count(1) from micro\_class where mc\_competition\_id=competition\_id) count ,(select count(1) from judge\_mc where judge.judge\_id=judge\_id) review from judge left join competition on competition\_id=judge\_competition\_id left join type on competition\_type=type\_id where judge\_user\_id=#{user\_id}"+

"</script>")

List<HashMap> findCompetition(HashMap map);

//api获取数据

getMyCompetition=()=>{

get('/judge/competition').then(json=>{

if(json.code!==0){

return

}

this.setState({myCompetition:json.data})

})

};

//数据展示

{this.state.myCompetition.map(com=>{

return <div style={{margin:20,display:'inline-block'}}>

<div style={{padding:20,background:'#fff',borderRadius:6,

boxShadow:'1px 1px 3px #aaa',

maxWidth:430,width:'90vw',color:'#555',lineHeight:2}} key={com.competition\_id+com.judge\_user\_id}>

<div style={{textAlign:'center',fontSize:20,marginBottom:20}}>{com.competition\_name}</div>

<div>类型：{com.type\_name}</div>

<div style={{whiteSpace:"nowrap",overflowX:'hidden'}}>时间：{com.competition\_time}</div>

<div>作品{com.count}个，{com.review}个已打分 <span style={{float:'right'}}>进度{com.review/com.count\*100}%</span></div>

<div style={{textAlign:'right',marginTop:20}}><Link to={'/judge/'+com.judge\_id}><Button type="primary" >进入比赛</Button></Link></div>

</div>

</div>

})}

### 5.6系统测试

测试阶段非常的重要，在系统的开发里，都很的明确的进行了确认 [10]。

测试阶段常分为系统测试、集成测试和单元测试，单元测试由开发人员在开发时完成。

在微课比赛系统项目开发中，我们完成系统测试工作。依据开发计划、需求以及分析文档编制测试计划，编写测试用例，执行测试，最后完成测试分析。开发时配合着测试，依据执行测试的缺陷记录进行修改，关闭缺陷。从而最大可能地模拟用户场景测试软件功能，提高软件质量[7]。

#### 5.6.1微课比赛系统测试环境

在微课比赛管理系统中，采用的软硬件环境，如表5-2所示。

表5-2测试环境

|  |  |
| --- | --- |
| 硬件 | |
| 服务器：双核Intel3 CPU , 内存4G | 客户端：1台intel(R)Core (TM) i5-3230M CPU@2.40GHz 2.40GHz ； |
| 软件测试环境 | |
| 数据库系统：MySQL  客户机操作系统： Windows 10 | |

#### 5.6.2微课比赛系统主要功能测试

测试是软件开发的一个必不可少的环节，开发软件的过程中，我们一直非常注重软件测试。就算是开发很大的系统时，对于复杂的问题，人类的思想理解满足不了实际的需求，就算是我们本着刨根问底的精神，对于与开发人员的沟通我们也不能保证做到更好。因为软件测试期间的每个阶段都会出现不同的错误。所以项目开发结束后，我们会对开发的软件系统进行严格的测试，尽早的发现错误以及修改错误。但也无法排除所有的错误，难免会有意外bug。当软件正式使用的时候，会出现这些当时进行软件测试没有发现的错误，这样再进行修改会极大的增加成本。所以软件测试具备下列原则：

第一，开发与测试并行；

第二，测试不能完全消除bug；

第三，测试的流程我们无法保证正确性；

第四，我们未必能消除所有一直bug；

第五，逆向思维对测试来说非常重要；

第六，测试不能过于笼统，从最小单元开始测试；

第七，交叉测试或交给专业的测试人员效果要好得多；

第八，按照需求进行测试，非需求性bug修复会增加成本。

测试是测试人员使用人工或者专业软件进行软件测试的一个过程，本质上就是弄清楚生产的软件与需求的软件的差别。当下，软件测试是软件质量保证的关键。也是铲平迈向客户的最后一关，目的就是避免意外的bug给客户带来不良影响，造成验证的后果。

一般而言，软件测试工程师的数量仅次于开发人员。因此，开发系统是要重视软件测试工作。不能以为开发结束后，项目就结束了。实际上，他们之间工作的时间是同等的。就测试而言，目标是发现软件中的错误，但是大型错误也不算是最终的目的 [16]。主要目的还是开发出满足用户需求的软件 [12]。

在业务整合和功能优化实现后的微课比赛管理系统，部分子系统、用例功能测试结果描述如表5-3：

表5-3功能测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 子系统名称 | 用例 | 功能说明 | 测试结果 | 备注 |
| 用户信息管理系统 | 个人信息查询 | 所有用户可查看，不可修改 | 符合要求 |  |
| 密码修改 | 自己、管理员可修改 | 符合要求 |  |
| 注册、新增 | 用户注册，管理员新增用户 | 符合要求 |  |
| 封号 | 管理员封号 | 符合要求 |  |
| 比赛管理系统 | 管理员创建比赛 | 创建比赛，指定裁判 | 符合要求 |  |
| 教师参赛 | 上传作品，查看结果 | 符合要求 |  |
| 裁判打分 | 打分 | 符合要求 |  |
| 管理员创建分类 | 创建、修改分类 | 符合要求 |  |

### 5.7本章小结

本章首先根据分析模型设计了数据库，然后分别从界面原型，实现原理和关键代码实现了项目管理系统中的关键用例，最后介绍部分测试用例及测试结果。

## 工作总结与展望

至此，本论文大体已经完成，通过本次编写论文，我强化了自己的专业能力，掌握了更多的技能，也对莅临知识有了非常深刻的体会，本次项目主要使用了以ssm为核心的spring-boot框架进行设计与开发。首先，对系统的开发和工具是精心挑选的。然后，根据系统目标和具体实现的功能，详细分析了系统的发展现状、趋势和需求。最后以这些分析为基础，对系统所需要的功能进行了设计，并结合所设计的系统给出了具体实现方法。

系统实现的基本功能是：管理员登录系统，创建比赛，指定裁判。教师注册，登录系统，上传作品，查看比赛结果。裁判登录，评分。

在系统开发过程中，本人学识浅薄，经验不足，走了不少弯路，前期代码缺乏组织规范，随着项目的进行，系统愈发臃肿，代码变得难以维护，不得不对代码进行重构，前端使用react框架将代码分块，使用规范化的开发讲代码组织起来，虽然开发速度上并没有多大提升，但对于后期维护，工作量就小的多了，我们可以很轻松的定位到模块，对模块的代码进行修改而不影响整体的运行，这也是低耦合模块化开发的好处，在后台开发上，我对数据库操作愈发的熟练，对mybatis有了更深层次的理解，使用注解的方式解放mapper.xml文件，代码变得轻盈起来，xml与注解各有千秋，个人更偏好注解的方式。在api设计上，由以前的毫无组织的api到restful api，通过设计规范，减少不必要的文档。

虽然此次项目有遇到不少问题，但也觉得受益匪浅，看到了许多我自己的不足之处，我对未来的方向也更加清晰，一个好的软件，从外到内，都应该具有设计感，我们的手指，就像刻刀一样，在键盘上跳动，雕刻出无暇的软件艺术品。

## 致谢

在我的辛勤努力下，本文包含的研究成果已经完成，但仍然有很多细节需要优化，就我而言，本系统不仅仅是一个简单的系统，他也是我大学期间的所学的结晶，亦将成为我成长路上的一道分水岭，于我而言，它承载着我在校期间的回忆，他的价值远超于它本身，或是多年后，系统所用的技术被淘汰，但它们，会永存在我的回忆里。

在系统研发期间，我非常感谢指导老师对我无私的指导，也感谢公司给我充足的时间来完善系统，感谢我的朋友提的宝贵建议。在项目前期，指导老师通过严格控制关键时间节点，关键任务完成度来督促我们完成项目，无论是论文的写作规范，还是项目的具体实现，老师都给予了我耐心的指导，并提出了许多宝贵的建议，让我从中不仅学到了很多专业知识，更是学会了对工作认真负责的态度。在未来的道路上，这将非常有用。

在此，向所有关心和帮助过我的老师、同学和朋友们表示由衷的感谢！

再次衷心地感谢各位老师在百忙之中来参加本论文的审阅和答辩！

## 参考文献

[1] Bruce Eckel，《think in java》[M]，Prentice Hall PTR 2003

[2] 阮一峰，《ES6标准入门 （第三版）》[M]，电子工业出版社 2017

[3] 徐超，《React进阶之路》[M]，清华大学出版社 2018

[4] 肖睿，《SSM企业级框架实战》[M]，中国水利水电出版社 2017

[5] 朱少民，《软件测试方法和技术》[M]，清华大学出版社

[6] (英)戴特，《数据库系统导论》，机械工业出版社[M] 2007

[7] 范展源； 罗福强，《JWT认证技术及其在WEB中的应用》[C]， 四川大学锦城学院 2016

[8] 孟祥增； 刘瑞梅； 王广新，《微课设计与制作的理论与实践》[C]，山东师范大学川传媒学院 2014

[9]王金柱； 李元诚，《MD5算法在J2EE平台下用户管理系统中的应用》[C]，华北电力大学计算机科学与技术学院 2018

[10] 娄亚丽，《基于B/S模式教学视频资源库的研究与设计》[C]，华东师范大学 2011