

**数据库系统原理课程设计**

题 目： 智慧校园在线测试系统

院 （系）： 计算机与信息安全学院

组员： \*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

指导教师姓名： 刘洪波、折建章、唐麟

**摘要**

本文介绍了智慧校园在线测试系统的设计与实现，该系统采用团队合作的方式进行开发。 \*\*负责协调整个项目的进度，分配任务，负责相关运行环境（MySQL，Redis，服务器）的搭建，答题，邮箱验证码，登录限制等后端功能的实现。\*\*负责机构注册、答题测试、答题结果和排行榜的页面设计和功能的实现，\*\*同学负责首页、注册登录、配置比赛页面的设计和登录、注册和配置比赛功能的实现。

该系统采用Web开发技术，使用Django作为框架，HTML，CSS， JavaScript进行前端的编写，使用关系型数据库MySQL，非关系型数据库Redis。具备注册、出题、答题和查看功能。注册分为普通用户和机构用户；机构用户可以快速出题，通过导入外部数据按类别出题；普通用户可以选择想测试的题目类别进行答题，答题完毕可以查看成绩和排名。

本文先对需求进行分析，对系统进行了分析，详细介绍了数据库实现相关内容，介绍了智慧校园在线测试的整体流程，系统画出了逻辑结构的E-R图，对整个系统的完整性有了比较全面的了解；逻辑结构方面列出了各个表之间的关系，完善系统的实体及各属性之间的联系，并介绍了系统的设计与实现。最后对系统进行总结。

**关键字**：MySQL、Redis、Django、数据库安全、Web开发、数据库设计、前端设计

目录

[1. 系统分析 3](#_Toc108425962)

[1.1系统功能体系图 3](#_Toc108425963)

[1.2 UML活动图 3](#_Toc108425964)

[1.3结构化需求分析 4](#_Toc108425965)

[2. 所用开发工具和技术 6](#_Toc108425966)

[2.1 开发工具 6](#_Toc108425967)

[2.2使用技术 6](#_Toc108425968)

[2.3 部署环境 6](#_Toc108425969)

[2.4 团队合作方式 6](#_Toc108425970)

[2.5 数据库选择 7](#_Toc108425971)

[3. 数据库系统设计 7](#_Toc108425972)

[3.1 概念结构设计 7](#_Toc108425973)

[3.2逻辑结构图 7](#_Toc108425974)

[3.3 物理结构设计 8](#_Toc108425975)

[3.4系统安全性设计 13](#_Toc108425976)

[4. 详细设计与实现 18](#_Toc108425977)

[4.1 注册登录功能模块设计 18](#_Toc108425978)

[4.2 用户功能模块设计 23](#_Toc108425979)

[4.3 管理员功能模块设计 31](#_Toc108425980)

[5. 测试 35](#_Toc108425981)

[5.1功能测试 36](#_Toc108425982)

[5.2性能测试 36](#_Toc108425983)

[5.3用户界面测试 37](#_Toc108425984)

[5.4兼容性测试 38](#_Toc108425985)

[5.5安全测试 38](#_Toc108425986)

[5.6接口测试 38](#_Toc108425987)

[6. 存在问题及改进思路 39](#_Toc108425988)

[6.1系统的安全性 39](#_Toc108425989)

[6.2随机出题算法 39](#_Toc108425990)

[7. 总结与发展展望 39](#_Toc108425991)

[7.1 遇到的困难 40](#_Toc108425992)

[7.2 相关优化 41](#_Toc108425993)

[7.2.1 注册成为机构的用户要求 41](#_Toc108425994)

[7.2.3用户体验 41](#_Toc108425995)

[7.3感悟 42](#_Toc108425996)

[附录 44](#_Toc108425997)

[1. 后端重要代码 44](#_Toc108425998)

[2. SQL语句 50](#_Toc108425999)

[3. 小组会议 53](#_Toc108426000)

|  |  |  |  |
| --- | --- | --- | --- |
| 项目名称 | **智慧校园在线测试系统** | | |
| 院 系 | **计算机与信息安全学院** | 专业 | **信息安全** |
| 实验日期 | **2022年06月27日** | | |

团队任务分工表

|  |  |  |  |
| --- | --- | --- | --- |
| 负责人 | \*\*\*\*\* | | |
| 合作  完成  方式 | **初期**：团队根据需求一起讨论如何进行概念模型设计，逻辑模型设计，并 以较好实现完整性，安全性。讨论结束后，根据概念模型进行物理结构设计。  **中期**：1、组长配置文件并搭建整体框架，全员合作编写前后端，使用GitHub进行协作开发。2、组员以向组长线下交流请教和查阅资料的方式学习完成各自认为。3、各表数据编写。  **完结**：共同整理编写项目文档。 | | |
| 序号 | 学号 | 姓名 | 个人负责模块 |
| 1 | \*\* | \*\* | 负责运行环境（MySQL，Redis，服务器）的和相关流程的搭建，答题，邮箱验证码，登录限制等后端功能的实现。 |
| 2 | \*\* | \*\* | 机构注册、答题测试、答题结果和排行榜的页面设计和功能的实现 |
| 3 | \*\* | \*\* | 负责首页、注册登录、配置比赛页面的设计和登录、注册和配置比赛功能的实现 |

# 系统分析

**（1）业务需求**

智慧校园在线测试系统用于用户的在线测试，目的是给用户提供在线答题自检测服务，通过自测达到检测自我学习成果或提高学习效率的目的。

智慧校园在线考试系统，需要通过互联网技术实现了测试工作的网络化、无纸化和评分智能化，节省成本、减轻用户自测、评分的工作量，实现真正的无纸化学习，提高了学习效率。

**（2）用户需求**

普通用户：可以进行注册登录，选择不同的题目进行答题，答题后可查看答题结果以及排名，历史答题情况，查看不同题目的人气情况。

机构用户：进行机构用户注册，下载出题模板，按照模板出题并上传题目形成题库，设置题目类型，从题库中抽取题库出题，让答题的用户填写相关信息。

**（3）系统需求**

该系统主要分为三个模块，分别为用户基本信息模块，机构注册模块，测试模块。用户基本信息模块包含登录、注册、忘记密码等基本功能；机构注册模块用于机构邮箱注册，只有机构用户才能进行出题操作。测试模块主要分两个功能，一个是题库的配置以及出题，另一个是自测答题。

## 1.1系统功能体系图

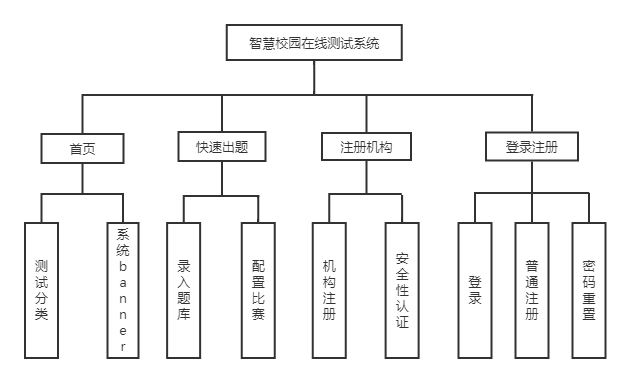


图1-1 系统功能体系图

## 1.2 UML活动图

UML活动图展示了智慧校园在线系统的活动的动态行为，简要描述了用户与系统、机构之间的关系，用户注册成功后，可经过系统安全认证成为机构，然后进行配置题库及出题让用户进行测试，用户则只需要注册成功后登录即可选择题库进行自主测试。

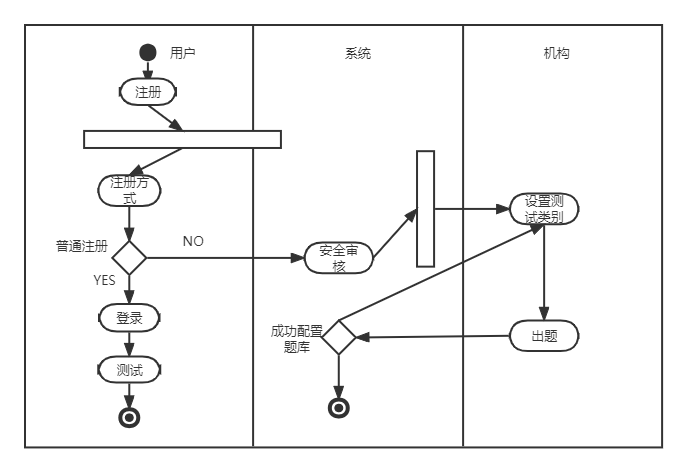


图1-2 业务流程UML活动图

## 1.3结构化需求分析

在智慧校园在线测试系统中，用户需要注册机构并认证成功成为机构，机构导入题目数据从中快速出题，并且题目都符合规则后才能成功出题配置比赛。普通用户注册成功后即可按类别来参加测试，测试完毕可以可以查看成绩、排名等数据分析

数据流图：

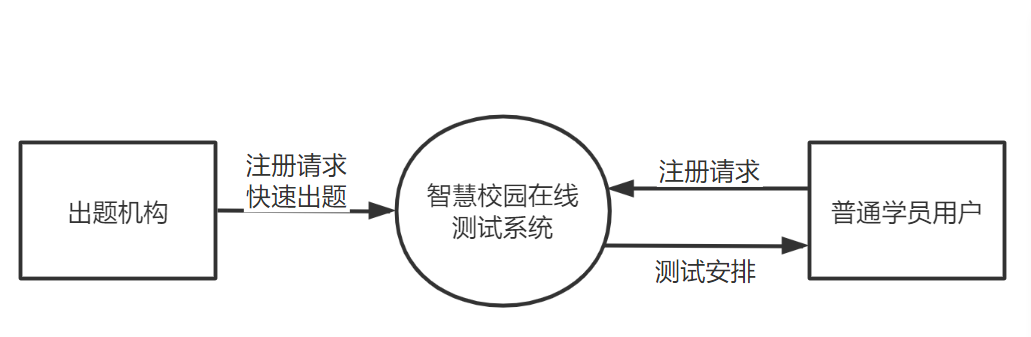


图1-3 顶层图示例

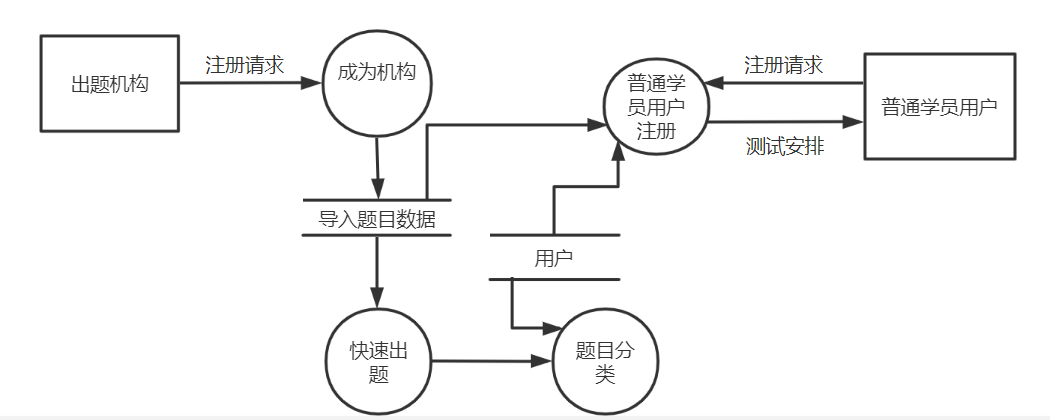


图1-4 0层图示例

状态图：

普通用户首先成功注册后可以按类别来选择题目进行测试，测试完毕后查看成绩结果。

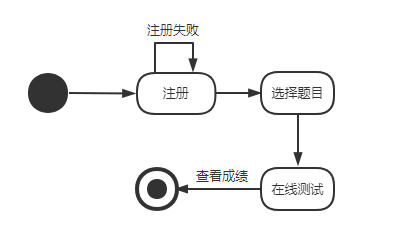


图1-5 普通用户状态图

用户注册机构并且认证成功后成为机构，机构导入题目进行配置比赛。

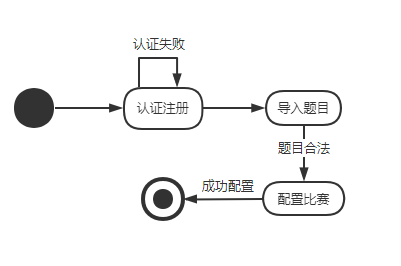


图1-6 机构用户状态图

# 所用开发工具和技术

## 2.1 开发工具

团队使用 Pycharm和VScode进行前后端的开发。

## 2.2使用技术

使用Django框架进行前后端的编写。

使用原生HTML，CSS，JavaScript等编写前端。

## 2.3 部署环境

MySQL 5.5以上版本

Redis 7.0.0

Python3.6以上版本

Django版本2.1

## 2.4 团队合作方式

本次电商系统项目为前后端分离的结构，小组成员为3人。为保证合作过程中全流程的跟进与小组成员的代码质量，我们使用git作为代码管理工具，使用GitHub作为代码管理平台。

本系统项目地址为：https://github.com/Fly-Pluche/Exam\_system\_database该项目为私有仓库，外部无法访问。

开发流程遵守标准git开发流程，每一个组员在更改项目有关代码时都要进行pr-审查-合并流程。主要由组长审查代码并反馈，所有组员共同跟进项目进度，并撰写项目手册。项目总览如下图所示:

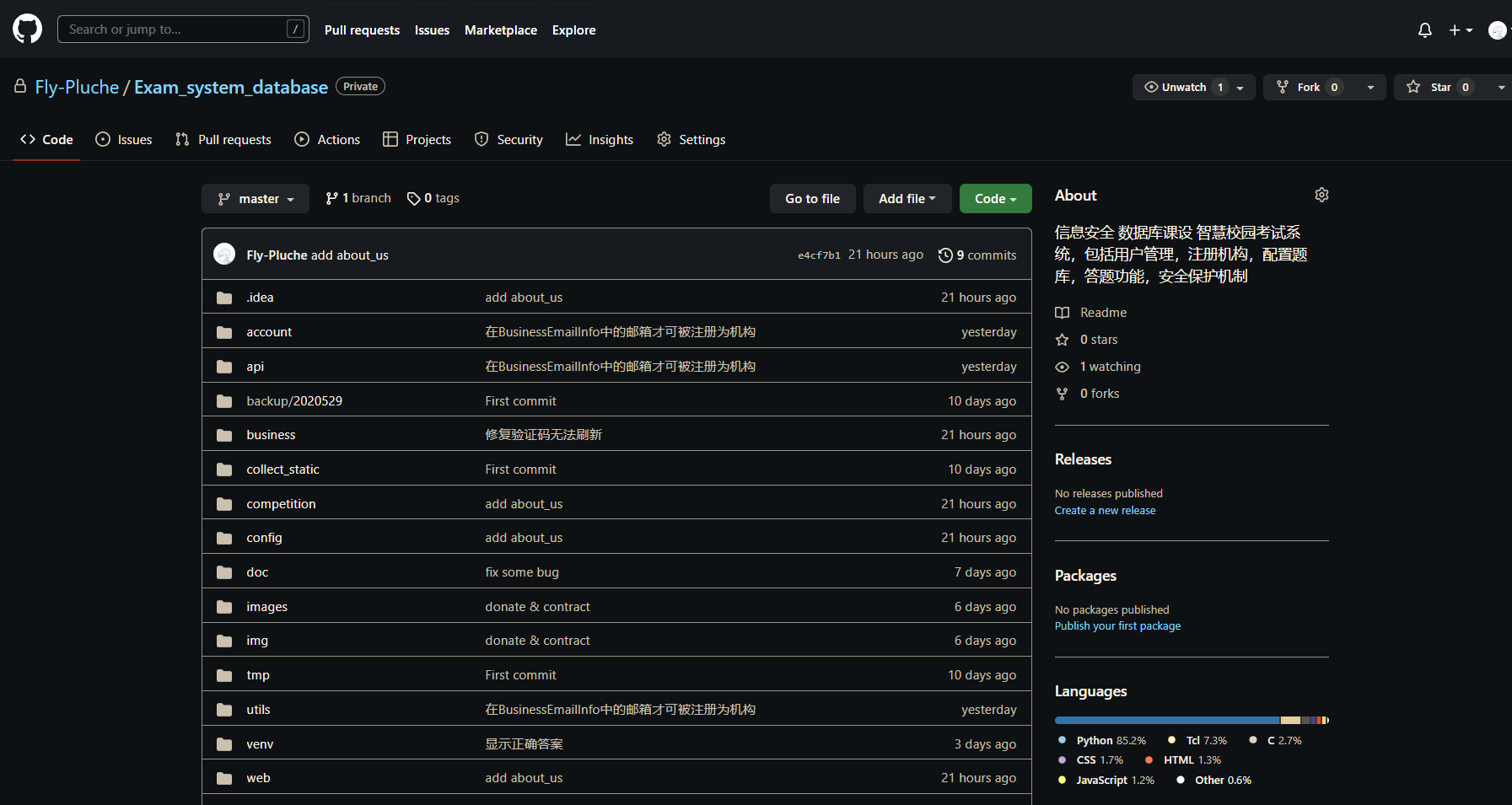


图2-4 GitHub项目总览

## 2.5 数据库选择

关系型数据库:使用 MySQL，原因如下：

* MySQL是一种关系型数据库管理系统，符合我们在课上所学，上手较快
* MySQL是最流行的DBMS之一，长期商业场景验证，功能丰富，性能稳定
* MySQL是最知名的开源数据库之一，社区庞大，学习资料丰富
* 运行速度快，成本低，可移植性强的特点。

非关系型数据库：使用Redis，基于内存的数据库，返回速度快，常用来做缓存数据库。

实验使用的数据库在云服务器部署，共享数据库，协同开发。

# 数据库系统设计

## 3.1 概念结构设计

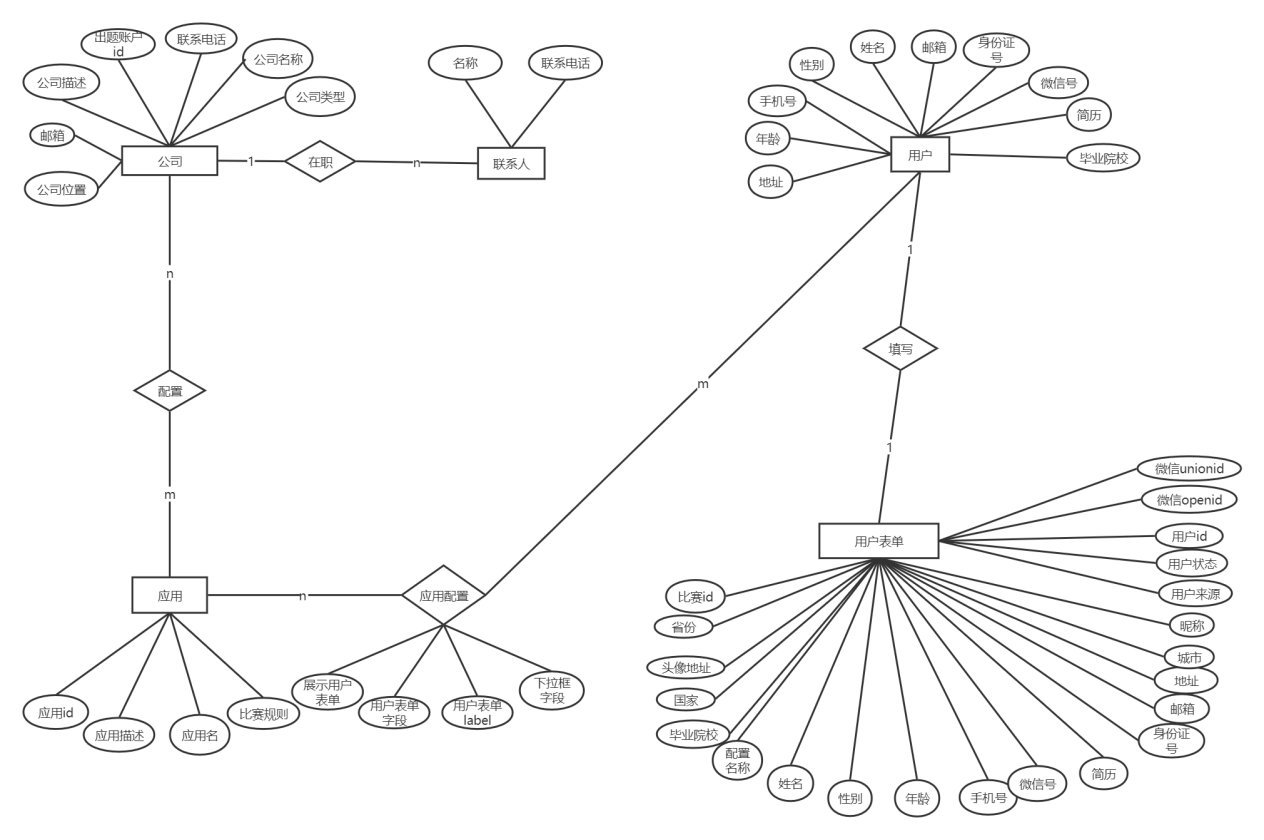


图3-1 E-R图

## 3.2逻辑结构图

概念结构是独立于任何一种数据模型的信息结构，逻辑结构设计的任务就是把概念结构设计阶段设计好的基本E-R图转换为与数据库管理系统产品所支持的数据模型相符合的逻辑结构。

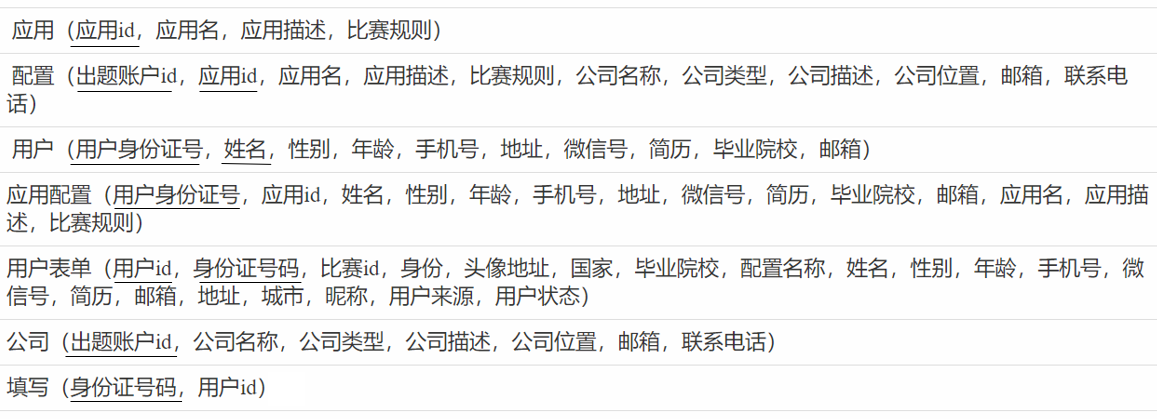


图3-2 逻辑结构图

## 3.3 物理结构设计

1. 用户表单表

用户表单表用于用来存放用户的详细信息，如表1所示。

表1 用户表单表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 描述 |
| id | int(11) | 主键、非空、自增 | 用户表单唯一标识 |
| status | tinyint(1) | 非空 | 用户身份 |
| created\_at | datetime(6) | 非空 | 创建时间 |
| updated\_at | datetime(6) | 非空 | 更新时间 |
| uid | varchar(22) | 默认空字符，列值唯一 | 用户唯一标识符 |
| user\_src | int(11) | 非空 | 用户来源 |
| user\_status | int(11) | 非空 | 用户状态 |
| unionid | varchar(32) | 默认空字符 | 微信unionid |
| opendid | varchar(32) | 默认空字符 | 微信opendid |
| wxid | varchar(32) | 默认空字符 | 微信号 |
| name | varchar(32) | 默认空字符，列值唯一 | 姓名 |
| email | varchar(40) | 默认空字符，列值唯一 | 邮箱 |
| sex | int(11) | 非空 | 性别 |
| age | int(11) | 非空 | 年龄 |
| nickname | varchar(32) | 默认空字符 | 昵称 |
| avatar | varchar(60) | 默认空字符 | 头像 |
| phone | varchar(11) | 默认空字符 | 电话号码 |
| country | varchar(32) | 默认空字符 | 国家 |
| province | varchar(32) | 默认空字符 | 省份 |
| city | varchar(32) | 默认空字符 | 城市 |
| location | varchar(60) | 默认空字符 | 地址 |
| is\_upfrade | int(11) | 非空 | 会员是否升级 |
| upgrade\_time | datetime(6) | 默认空字 | 会员升级时间 |
| expire\_time | datetime(6) | 默认空字符 | 会员到期时间 |
| upgrade\_count | int(11) | 非空 | 会员升级次数 |

1. 用户表

用户表用来存放用户的基本信息，如表2。

表2 用户表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 描述 |
| id | int(11) | 主键、非空、自增 | 用户唯一标识 |
| status | tinyint(1) | 非空 | 用户身份 |
| created\_at | datetime(6) | 非空 | 创建时间 |
| updated\_at | datetime(6) | 非空 | 更新时间 |
| kind\_id | carchar(32) | 默认空字符，列值唯一 | 比赛唯一标识 |
| uid | varchar(22) | 默认空字符，列值唯一 | 用户唯一标识 |
| name | varchar(24) | 默认空字符 | 姓名 |
| sex | varchar(1) | 默认空字符 | 性别 |
| age | int(11) | 非空 | 年龄 |
| phone | varchar(11) | 默认空字符 | 电话号码 |
| wxid | varchar(24) | 默认空字符 | 微信号 |
| pid | varchar(18) | 默认空字符 | 手机号 |
| graduated\_from | varchar(60) | 默认空字符 | 毕业院校 |
| address | varchar(60) | 默认空字符 | 地址 |

1. 权限组管理表

权限组管理表用来存放管理员的编号及姓名，如表3。

表3 权限组管理表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 描述 |
| id | int(11) | 主键、非空、自增 | 管理员唯一标识 |
| name | varchar(80) | 非空，列值为一 | 显示管理员姓名 |

1. 权限组管理许可表

权限组管理许可表用来存放管理许可信息，如表4所示。

表4 权限组管理许可表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 描述 |
| id | int(11) | 主键、非空、自增 | 唯一标识 |
| group\_id | int(11) | 非空，列值为一 | 显示管理组标识 |
| permission\_id | int(11) | 非空，列值为一 | 许可标识，表示系统允许 |

1. auth\_user表

在创建Django项目之后直接执行数据迁移命令会自动生成许多表，如auth\_user表。Django在启动之后就可以直接访问admin路由，需要输入用户名和密码。这里管理员就需要依赖auth\_user表才能完成用户的所有功能，如表6。

表5 auth\_user表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 描述 |
| id | int(11) | 主键、非空、自增 | 用户可执行操作的标识表id |
| password | varchar(128) | 非空 | 密码 |
| last\_login | datetime(6) | 默认空字符 | 上次登录 |
| is\_superuser | tinyint(1) | 非空 | 为超级用户 |
| username | varchar(150) | 非空，列值为一 | 用户姓名 |
| first\_name | varchar(30) | 非空 | 名 |
| last\_name | varchar(150) | 非空 | 姓 |
| email | varchar(150) | 非空 | 邮箱 |
| is\_staff | tinyint(1) | 非空 | 管理员用户也是常规用户，只分分配了is\_staff权限 |
| is\_active | tinyint(1) | 非空 | 用于判断用户是否处于激活状态 |
| date\_joined | datetime(6) | 非空 | Django Auth用户date\_joined字段datetime字符串 |

1. business\_appconfiginfo表

business\_appconfiginfo表用来存放测试信息。

表6 business\_appconfiginfo表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 描述 |
| id | int(11) | 主键、非空、自增 | 唯一标识 |
| status | tinyint(1) | 非空 | 状态 |
| created\_at | datetime(6) | 非空 | 创建时间 |
| updated\_at | datetime(6) | 非空 | 更新时间 |
| app\_id | varchar(32) | 非空 | 测试id |
| app\_name | varchar(40) | 默认空字符 | 测试名 |
| rule\_text | longtext |  | 测试规则 |
| is\_show\_userinfo | tinyint(1) | 非空 | 第三方授权认证 |
| userinfo\_fields | varchar(128) | 默认空字符 | 用户表单字段 |
| option\_fileds | varchar(128) | 默认空字符 | 下拉框字段 |

7、business\_busnessaccountifo表

business\_busnessaccountifo表用来存放机构的基本信息，具体如表10。

表7 business\_bunessaccountifo表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 描述 |
| id | int(11) | 主键、非空、自增 | 机构唯一标识 |
| status | tinyint(1) | 非空 | 状态 |
| created\_at | datetime(6) | 非空 | 创建时间 |
| updated\_at | datetime(6) | 非空 | 更新时间 |
| account\_id | varchar(22) | 非空 | 出题账户id |
| email | varchar(40) | 默认空字符，列值唯一 | 机构邮箱 |
| company\_name | varchar(60) | 默认空字符 | 机构名称 |
| company\_description | longtext |  | 机构描述 |
| company\_username | varchar(32) | 默认空字符 | 机构联系人姓名 |
| company\_phone | varchar(16) | 默认空字符 | 机构电话号码 |
| company\_location | longtext |  | 机构位置 |
| company\_type | int(11) | 非空 | 机构类型 |

8、business\_userinfoimage表

business\_userinfoimage表用来存放测试用户的基本信息。

表8 business\_userinfoimage表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 描述 |
| id | int(11) | 主键、非空、自增 | 测试用户唯一标识 |
| status | tinyint(1) | 非空 | 状态 |
| created\_at | datetime(6) | 非空 | 创建时间 |
| updated\_at | datetime(6) | 非空 | 更新时间 |
| name | varchar(60) | 默认空字符 | 测试用户姓名 |
| sex | varchar(60) | 默认空字符 | 测试用户性别 |
| age | varchar(60) | 默认空字符 | 测试用户年龄 |
| phone | varchar(60) | 默认空字符 | 测试用户电话号码 |
| wxid | varchar(60) | 默认空字符 | 测试用户微信号 |
| email | varchar(60) | 默认空字符 | 测试用户邮箱 |
| pid | varchar(60) | 默认空字符 | 测试用户电话id |
| graduated\_from | varchar(60) | 默认空字符 | 测试用户毕业院校 |
| address | varchar(60) | 默认空字符 | 测试用户地址 |

9、competition\_bankinfo表

competition\_bankinfo表用来存放比赛内容。

表9 competition\_bankinfo表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 描述 |
| id | int(11) | 主键、非空、自增 | 比赛唯一标识 |
| status | tinyint(1) | 非空 | 状态 |
| created\_at | datetime(6) | 非空 | 创建时间 |
| updated\_at | datetime(6) | 非空 | 更新时间 |
| bank\_id | Varchar(22) | 默认空字符 | 比赛id |
| choice\_num | int(11) | 非空 | 选择题数量 |
| fillinblan\_num | int(11) | 非空 | 填空题数量 |
| bank\_type | int(11) | 非空 | 测试类型 |
| kind\_num | int(11) | 非空 | 题型那个数量 |
| partin\_num | int(11) | 非空 | 测试分区号码 |
| bank\_name | varchar(40) | 默认空字符 | 题库名称 |
| uid | varchar(32) | 默认空字符 | 用户唯一标识符 |
| account\_id | varchar(32) | 默认空字符 | 出题用户id |

10、competition\_competitionkindinfo表

competition\_competitionkindinfo表用来存放单个测试题的具体测试要求信息。

表10 competition\_competitionkindinfo表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 描述 |
| id | int(11) | 主键、非空、自增 | 唯一标识 |
| status | tinyint(1) | 非空 | 状态 |
| created\_at | datetime(6) | 非空 | 创建时间 |
| updated\_at | datetime(6) | 非空 | 更新时间 |
| kind\_id | varchar(22) | 默认空字符 | 赛题唯一标识 |
| account\_id | varchar(32) | 默认空字符 | 出题用户id |
| app\_id | varchar(32) | 默认空字符 | 测试题id |
| bank\_id | varchar(32) | 默认空字符 | 题目id |
| kind\_type | iInt(11) | 非空 | 测试种类 |
| kind\_name | varchar(32) | 默认空字符 | 测试名称 |
| sponsor\_name | varcha(60) | 默认空字符 | sponsor姓名 |
| total\_score | int(11) | 非空 | 测试总分 |
| question\_num | int(11) | 非空 | 问题数量 |
| cop\_stratat | datetime(6) | 非空 | 测试开始时间 |
| period\_time | int(11) | 非空 | 测试限制时间 |
| cop\_finishat | datetime(6) | 默认空字符 | 测试结束时间 |
| tital\_partin\_num | int(11) | 非空 | 总分区数量 |

11、competition\_competitionqainfo表

competition\_competitionqainfo表用来存放用户测试信息。

表11 competition\_competitionqainfo表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 描述 |
| id | int(11) | 主键、非空、自增 | 测试配置信息唯一标识 |
| status | tinyint(1) | 非空 | 状态 |
| created\_at | datetime(6) | 非空 | 创建时间 |
| updated\_at | datetime(6) | 非空 | 更新时间 |
| kind\_id | varchar(32) | 默认空字符 | 比赛id |
| qa\_id | varchar22) | 默认空字符 | 答题情况 |
| uid | varchar(32) | 默认空字符 |  |
| qsrecord | longtext | 0 | 记录测试问题 |
| asrecord | longtext | 0 | 记录测试答案 |
| aslogrecord | longtext | 0 | 答题分数 |
| started\_stamp | bigint(20) | 非空 | 测试开始时间 |
| finished\_stamp | bigint(20) | 非空 | 测试结束时间 |
| expand\_time | int(11) | 非空 | 测试花费时间 |
| strated | tinyint(1) | 非空 | 测试开始 |
| finished | tinyint(1) | 非空 | 测试结束 |
| correct\_num | int(11) | 非空 | 正确的数量 |
| incorrect\_num | int(11) | 非空 | 错误的数量 |
| total\_num | int(11) | 非空 | 总体数 |
| score | double | 非空 | 得分 |
| correct\_list | varchar(10000) | 默认空字符 | 正确列表 |
| wrong\_list | varchar(10000) | 默认空字符 | 错误列表 |

12、django\_admin\_log表

django\_admin\_log表用来存放admin对用户在admin页面上所作的修改操作记录日志。

表13 django\_admin\_log表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 描述 |
| id | int(11) | 主键、非空、自增 | 唯一标识 |
| action\_time | datetime(6) | 非空 |  |
| object\_id | longtext | 非空 | 是否存在指定名称的索引 |
| object\_repr | varchar(200) | 非空 | 重新创建对象表达式 |
| action\_flag | samllint(5) | 非空 |  |
| change\_message | longtext | 非空 | 改变的信息 |
| content\_type\_id | int(11) | 默认空字符 | 内容类型id |
| user\_id | int(11) | 非空 | 用户id |

## 3.4系统安全性设计

3.4.1 数据库完整性约束

对于防止不符合规范的数据进入数据库，在用户对数据进行插入、修改、删除等操作时，DBMS自动按照一定的约束条件对数据进行监测，使不符合规范的数据不能进入数据库，以确保数据库中存储的数据正确、有效、相容。我们也进行了考虑与设计。

为了方便开发，我们将各类表中的大部分属性都默认设置为NULL或0。

而对于某些重要的属性，我们进行了合理的设定。

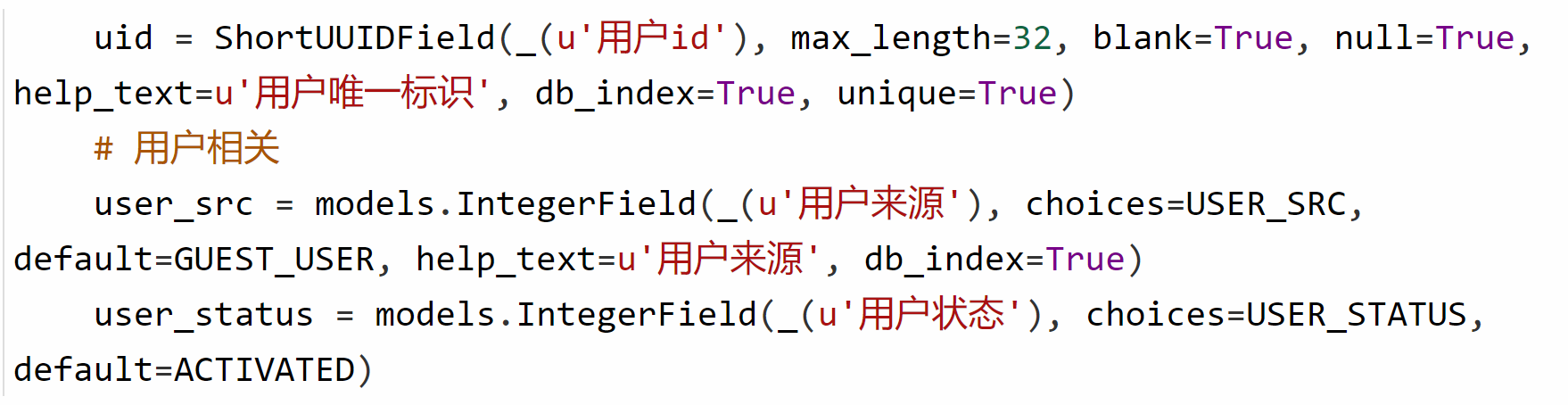


图3-4-1 数据库完整性约束代码

3.4.2 操作验证码

操作验证码，主要是为了解决三个问题：

1、账户暴力破解

2、高频次的接口访问

3、敏感操作二次确认(CSRF)

在注册以及登录界面设置相关验证码，降低机器冒充人类做暴力破解注册登录。



图3-4-2 操作验证码

3.4.3 邮箱验证

为防止暴力注册多个账号，注册后通过邮箱验证才可注册成功。



图3-4-3 邮箱验证

3.4.4 密码管理

将PBKDF2算法与SHA256哈希值一起使用，SHA256哈希是NIST推荐的密码拉伸机制。对于大多数用户来说，这足够了：它非常安全，需要大量的计算时间来破解。

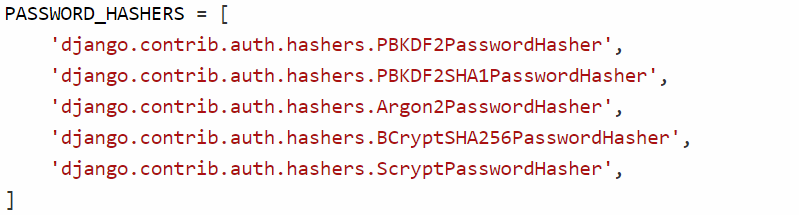


图3-4-4 密码管理

这意味着Django将使用PBKDF2来存储所有密码，但支持检查使用PBKDF2SHA1，argon2和bcrypt存储的密码。

3.4.5 CSRF防御

第一次响应来自某个客户端的请求时，会在服务器端随机生成一个 token，把这个 token 放在 Cookie 里。然后每次 POST 请求都会带上这个token。

使用csrf装饰器进行验证。

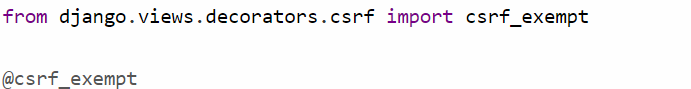


图3-4-5 CSRF防御

3.4.6 机构注册验证

为了防止普通和非法用户注册为机构用户，故设计运行注册表，只有在允许注册为邮箱才可注册为机构用户，并且同一个邮箱无法重复注册。



图3-4-6 机构注册验证代码

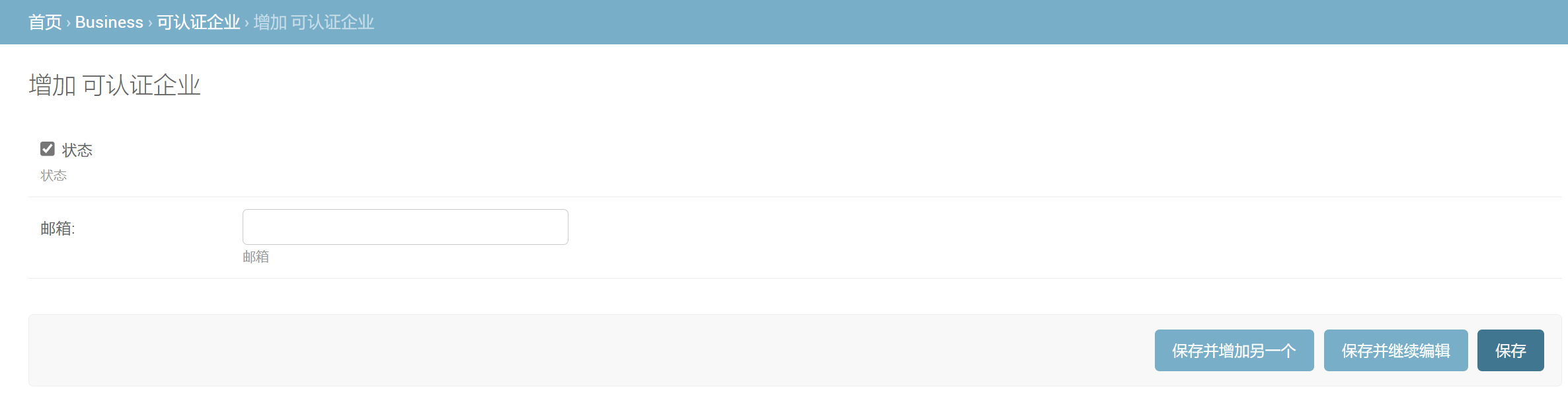


图 3-4-7 Admin账户管理

3.4.7 Superadmin 限制

对管理用户的数据操作进行限制，不允许修改唯一标识符等字段，防止出错。

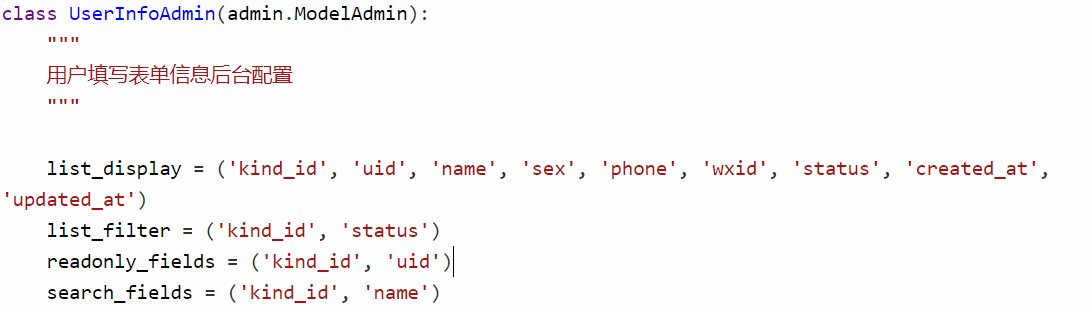


图3-4-8 Superadmin限制代码

3.4.8主备数据库

数据库一主一备指的是设置两个存储数据的数据库，一个是主数据库，一个是备用数据库设置主从数据库有以下好处：

1. 读写分离：通过MySQL复制可以实现读写分离，将读操作分布到多个不同的服务器上，减轻服务器的压力。

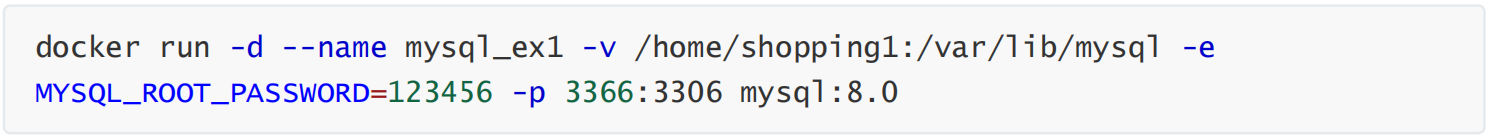
2. 备份：从库可以作为数据的异地实时备份。

3. 故障切换：当主库遇到故障，系统可以切换到从库，实现故障切换。

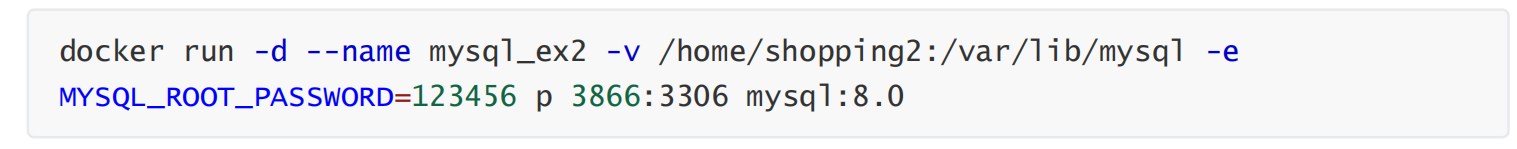
4. 升级测试：从库可作为测试服务器的数据库。

这里由于工期原因，我们所设置的备用数据库只用于主数据的数据备份，用于确保一定程度上的数据安全性。

创建主数据库：



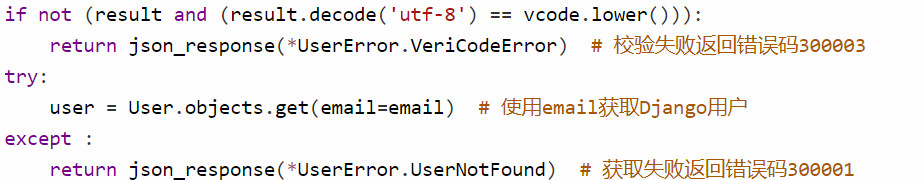
创建备用数据库：



然后在两个数据库之间通过转移日志的方式同步数据，首先在主备数据中分别创建一个用于备份的用户，然后使用 CHANGE MASTer TO 语句实现两个数据库的连接，最后再执行START SLAVE语句开始进行同步。

3.4.9 系统的稳健性

在视图函数中使用大量的try except语法，对异常进行捕获：



3.4.9 异常捕获

定义并使用大量的装饰器进行登录和异常的检测：



3.4.10 装饰器的使用

保存错误log文件，便于系统的维护：



3.4.11 错误log文件

设置大量的提示语句：



3.4.12 提示语句

# 详细设计与实现

## [4.1 注册登录功能模块设计](#_Toc5123_WPSOffice_Level2)

4.1.1 普通用户登录注册模块设计

注册页面前端部分代码如图3所示，使用Bootstrap模态框modal使登录信息覆盖在首页上，不需要再跳转到其他页面。其中用到的tabindex = - 1，表示元素可以聚焦，但是不能通过键盘导航来访问到该元素。这里用一个html表单来收集不同用户的输入，因为邮箱、密码和验证码都是同类型的结构，都可以使用<lable></ lable>的方法使点击到盒子的范围即可进行用户输入。另外对于验证码部分，使用canvas元素画背景图，用js产生随机验证码，背景颜色以及干扰图线，干扰点，然后渲染到页面上。



图4-1-1 普通注册页面代码

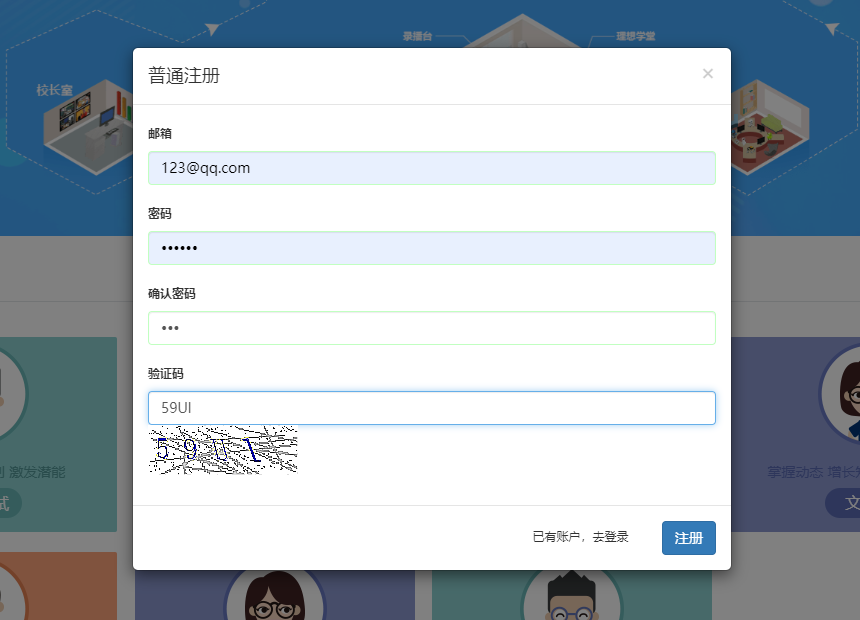


图4-1-2 普通注册图

登录页面前端部分代码与注册页面相类似，都是就用一个html表单来控制不同用户的输入。在用户输入过程中，利用JavaScript检查输入的用户名密码等是否有误图5，错误则利用js弹出警示框，如图6。这里采用json的连接方式，当服务器请求状态为200时，表示服务器已经准备就绪，可以接收数据。

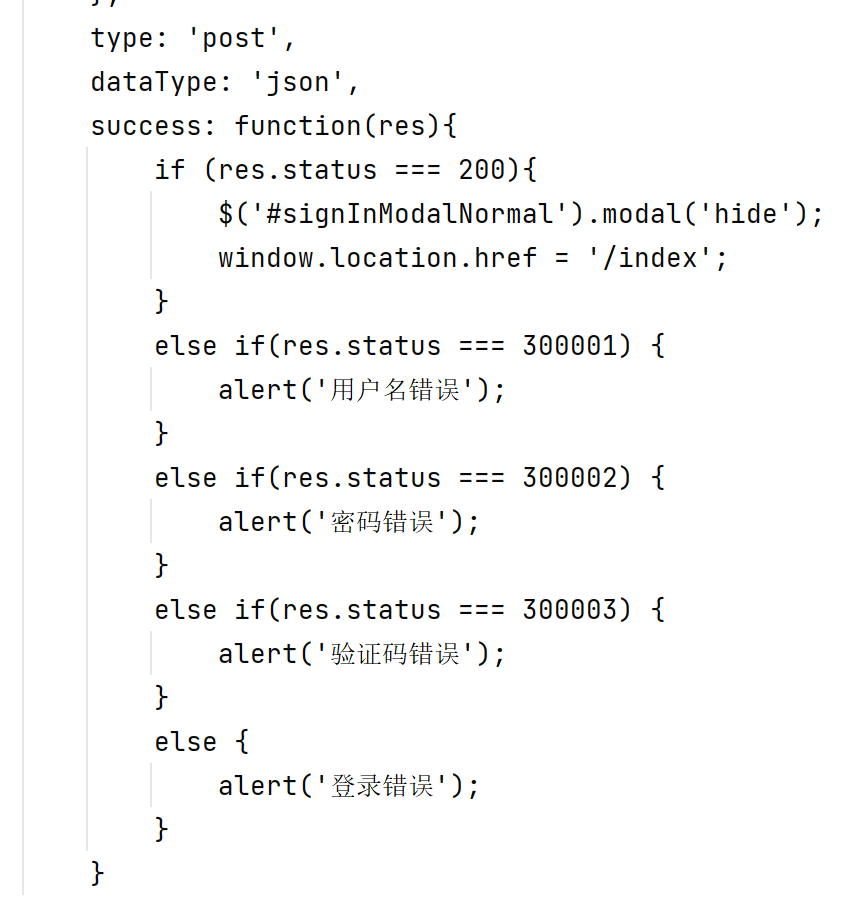


图4-1-3 js检查用户输入

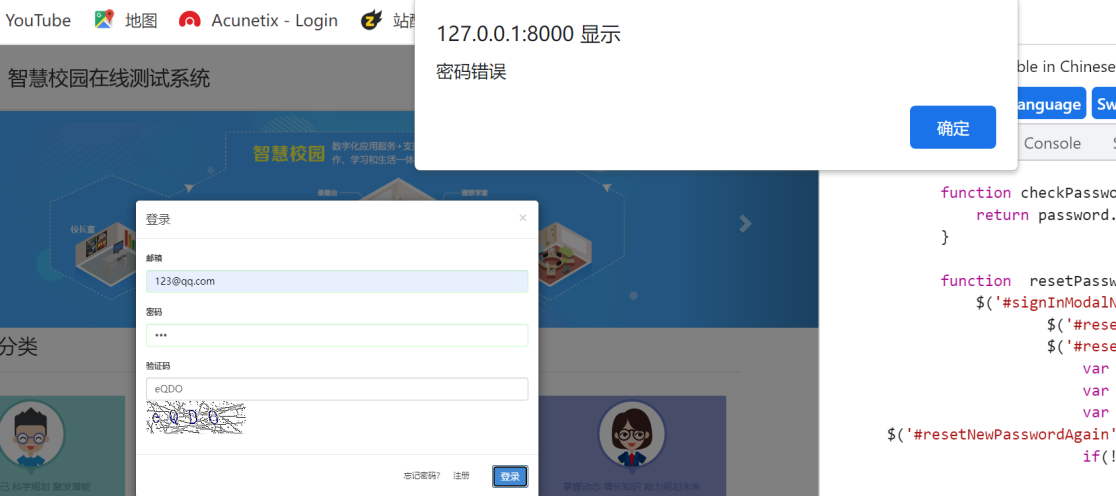


图4-1-4 密码错误效果图

重置密码部分，用户可以点击忘记密码进行密码重置，在输入新密码后，通过安全性考虑，需经过邮箱验证才可成功重置密码，只有返回数据库的请求状态为200时，才会显示邮件发送成功，这里也是采用modal模态框使提示信息悬浮在父页面上。

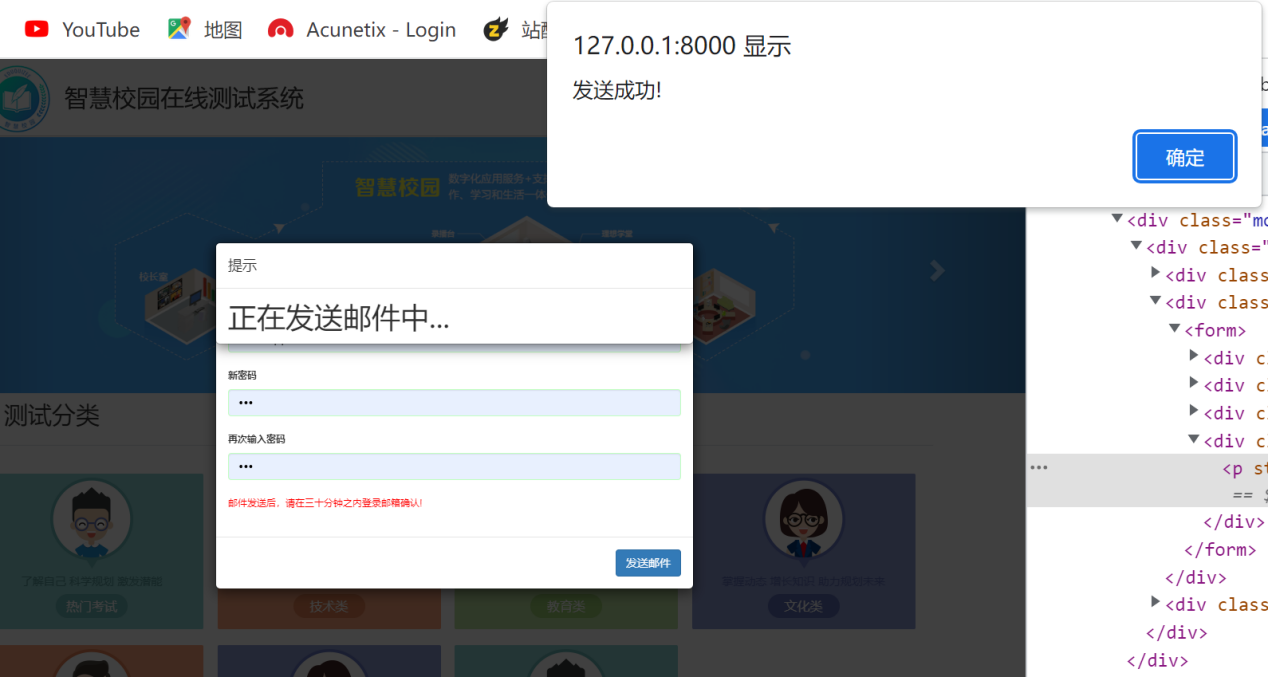


图4-1-5 重置密码

4.1.2 机构用户注册模块设计

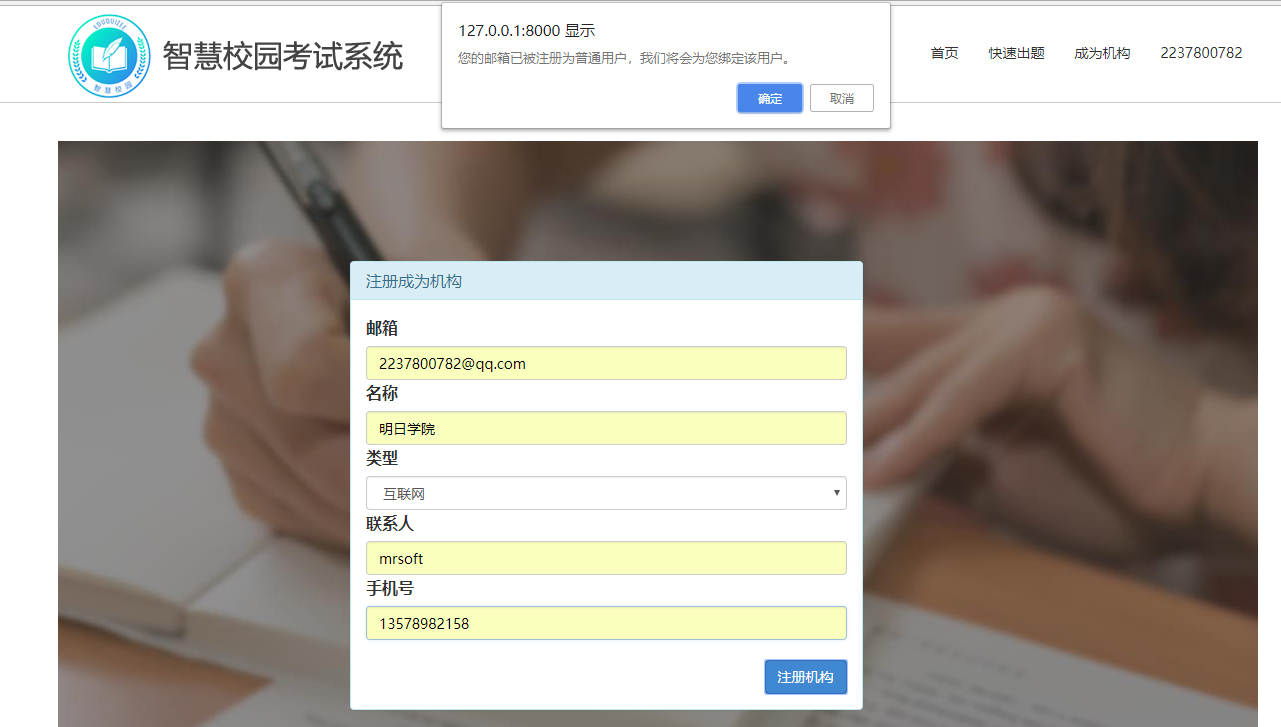


图4-1-6 注册机构前端界面

页面使用<label>标签实现表单信息的填写，label标签是一种常见的表单控件，使用户在填写表单的项目时有更好的体验。其中类型是一个选择列表，使用<option>实现类型的选择。



图4-1-7 主要界面前端代码

主要功能：

1、判断输入的邮箱是否已经注册成为机构用户。

2、判断输入的邮箱、名称、联系人和手机号是否符合规则。

实现代码：

主要通过正则表达式来判断输入是否合法。正则表达式又称规则表达式，在代码中常简写为regex、regexp或RE，正则表达式通常被用来检索、替换那些符合某个模式的文本。

正则表达式'^\\w+([-+.]\\w+)\*@\\w+([-.]\\w+)\*\\.\\w+([-.]\\w+)\*$'判断邮箱是否符规，'^[a-zA-Z0-9\_\\u4e00-\\u9fa5]{4,20}$'要求机构名称为4-29中文字母数字'^[\u4E00-\u9FA5A-Za-z]+$'要求用户名匹配姓名应该为汉字或大小写字母，手机号规则要求'^1[3|4|5|8][0-9]\\d{4,8}$'。最后通过post请求传递数据。



图4-1-8 判断输入是否符合规则

## 4.2 用户功能模块设计

4.2.1普通用户答题功能模块



图4-2-1 比赛列表界面

比赛列表界面第一个块状标签里点击比赛列表onclick=”history.back(-1)”回退到上一页，第二个<div>显示热门比赛，通过一个request.session.uid来访问超链接的数据，开始时间，结束时间，赞助商，热度。热度：kind.total\_partin\_num点击的次数



图4-2-2 比赛列表前端主要代码



图4-2-3比赛首页界面

比赛首页比赛首页，通过kind\_info.kind\_name获取测试题目标题并<span>加粗加大显示，通过请求request.session.username获取测试用户的用户名。使用<li>标签定义列表项目：出题机构、题目总分、题目数量、题库大小、开始时间和结束时间，同样通过kind\_info.属性获取对应的数据。点击开始跳转通过超链接跳转进入另一个页面。

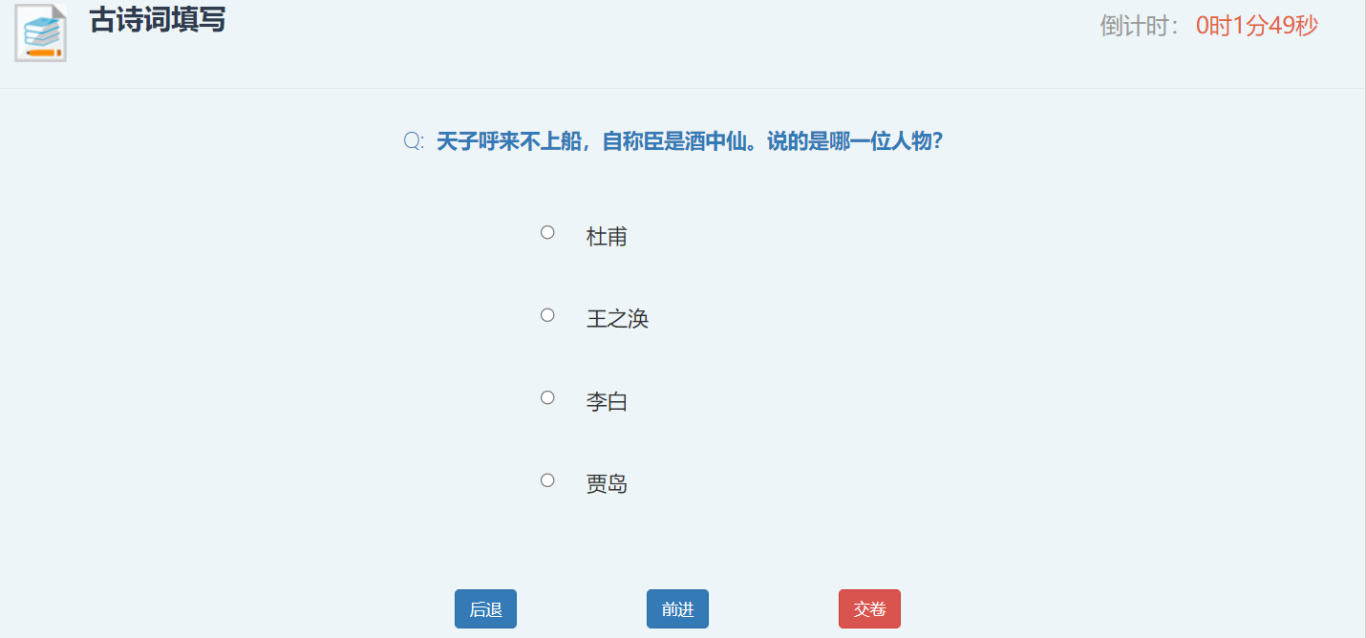


图4-2-4 答题界面

主要功能：

1. 题目的前进和后退功能
2. 记住之前的选项
3. 倒计时

首先定义变量当前页数currentPage=1、问题数量questionNum=0、答案answerDict=0、回答response、haeNext=falese、hasPrevious=false。开始计时，获取答题用户信息和哪套题目信息，通过get方式获取两个变量。根据该套题获取题目数量赋给questionNum,根据题目数量创建答案数组长度，并获取题目的答案。通过for遍历题目，当是选择题时候将答案数组索引设为”c\_”开头，填空题则设为”f\_”开头。

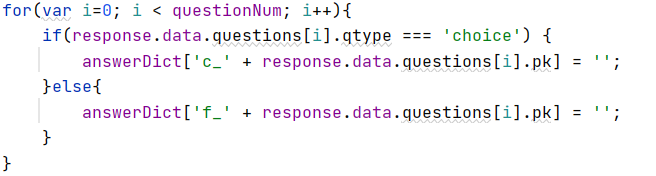


图4-2-5 答题界面代码

当该题目是选择题的情况：隐藏填空题，选择题获取四个选项的结果和一个自己的选项结果，如果当前页数currentPage小于题目总数questionNum时，hasNext设为True。填空题的情况是：获取问题和自己的答案变量，填空题的时候隐藏选择题，如果当前页数currentPage小于题目总数questionNum时，hasNext设为True。答完题目隐藏答题界面。

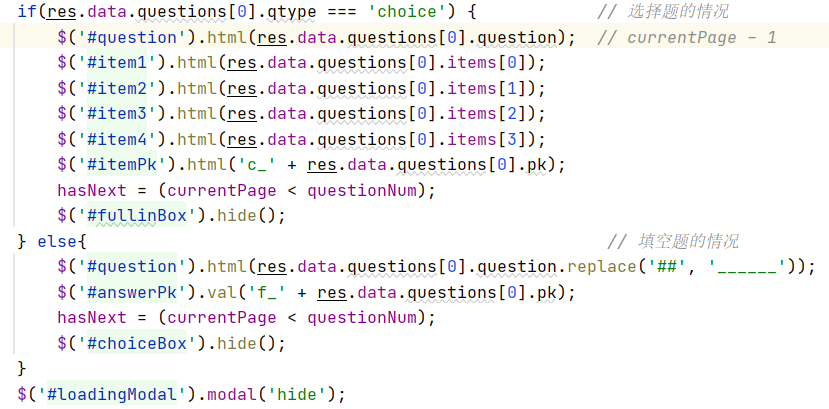


图4-2-6 题目展示代码

实现题目的点击后退情况：点击后退，如果hasPrevious=True,移除上一题的选择点removeCheck和清空填空框clearInputData。当前页面currentPage减一，如果是后退后的题目时选择题则展示选择框，定义正确答案变量，获取问题，四个选项和自己选项的答案。回退后如果原来有选择要记住之前的选项，如果答案是A，则第一个radion设为true，即显示第一个选项选中。以此类推......填空题的时候切记隐藏填空文本框。如果当前currentPage>1,设hasPrevious为true。如果当前页面小于问题总数，设hasNext=Ture。如果是填空题情况：获取后退题目的正确填空框展示，获取问题，获取正确答案。如果填空题的答案数组里还有’f\_’元素（即该题是填空题）就把赋给变量answerF(填空题)。填空题时隐藏选择框，如果当前currentPage>1,设hasPrevious为true。如果当前页面小于问题总数，设hasNext=Ture。当currentPage后退时不满足currentPage>1时设hasPrevious为False，显示“没有题目了”。

点击前进情况类似，主要区别是把currentPage加一。



图4-2-7 题目后退情况

选择题情况下，当点击第一个radio的时候，获取自己的选择变量，把A加入答案数组中，当点击第二个radio时，获取自己的选择变量，把B加入自己的答案数组中，点击第三第四的radio时为C、D。填空题情况下，获取自己的填空答案变量，将answerF的值加入answerDict数组中。

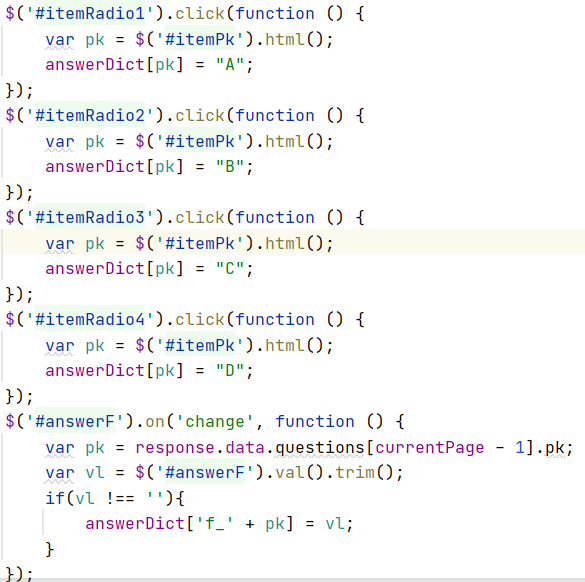


图4-2-8 题目选择代码

提交答案：点击提交按钮，首先弹出框确认提交，如果限制时间内提交是安全的就结束计时。定义一个空答案变量，遍历先前的答案数组，把答案一个个用“,”赋给answer值里。通过post请求传递数据，res.status === 200表示请求成功，请求成功则跳转页面。

一开始清空文本框，一开始没有选择。定义时分秒和后台配置时间，倒计时通过后台配置时间减去已经过的时间。

4.2.2普通用户查看测试结果和排行榜功能模块



图4-2-9 排行榜界面

实现主要功能：

1. 显示用户测试成绩排名

2. 显示历史最佳成绩

界面主要通过<tr> 标签定义 HTML 表格中的行来实现排行榜中用户的排名、昵称、得分和用时，for r in rank遍历每一行显示数据。历史最佳成绩：通过request.session.uid获取请求，res.status === 200表示请求成功，请求成功则显示我的排名和我的用时数据。requests.session()可以自动处理Cookies，做状态保持。如果获取到排名数据则显示排名和用时，否则提示“你还没有参加比赛”。

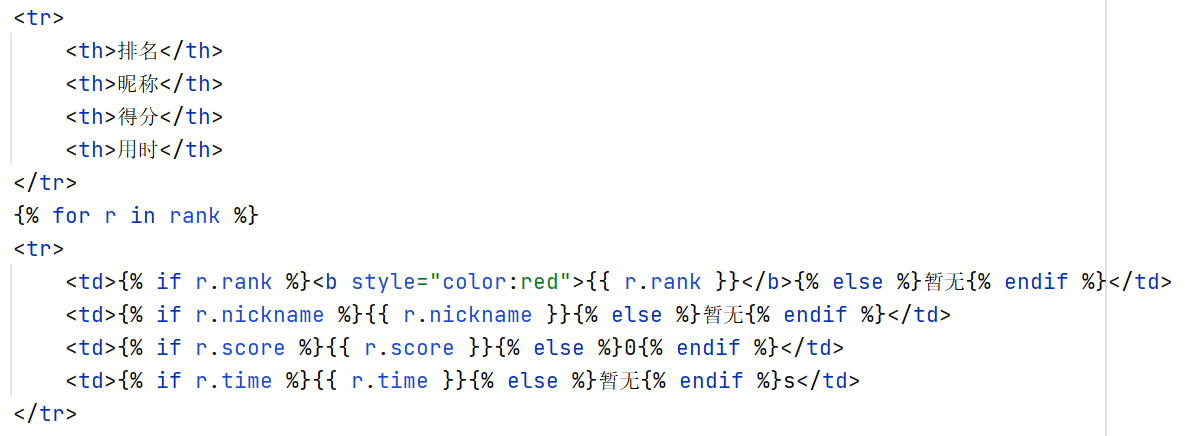


图4-2-10 排名表单代码



图4-2-11 历史最佳成绩

答题结果：

实现主要功能：显示同学的成绩等信息数据



图4-2-12答题结果界面

答题结果页面的成绩单首先request.session.username获取用户名字，使用<ul> 标签定义无序列表，获取得得分、耗时、答对题数和答错题数。

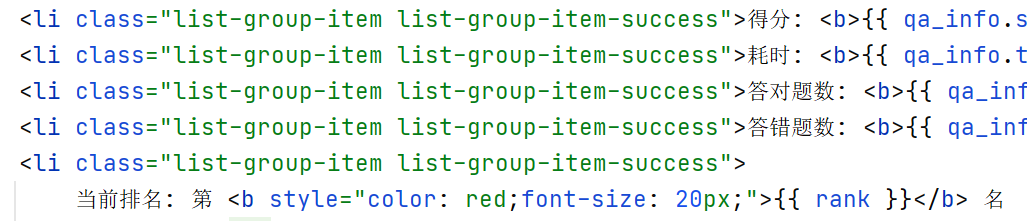


图4-5-14 排列列表代码



图4-2-13通过超链接跳转

4.2.3 机构用户快速出题功能模块

在快速出题前端页面，用<h1></ h1>标签来将标题单独成行放大，利用一个按钮链接到录入题库页面，同理配置比赛页面也是一样的方法。

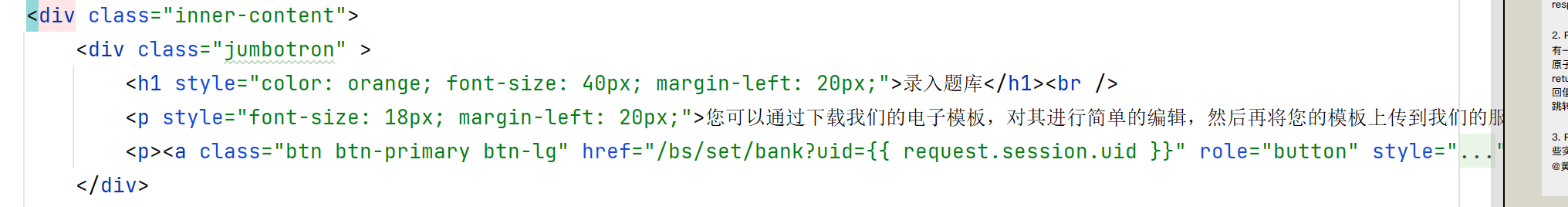


图4-2-14 配置比赛

在这里“col-md-3”表示一行最多能分成四列，以第一个模块“下载题库”为例，用一个<label>标签使点击范围扩大，增加用户体验。这里通过外部链接导入下载的题库。

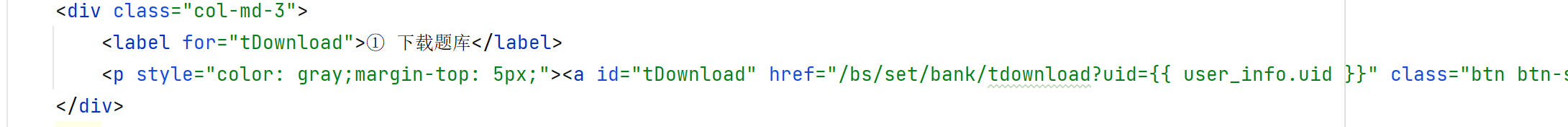


图4-2-15 下载题库

当没有按要求录入好题库时，当点击“开始录入”时会出现错误的页面，<strong>标签加粗错误提示。

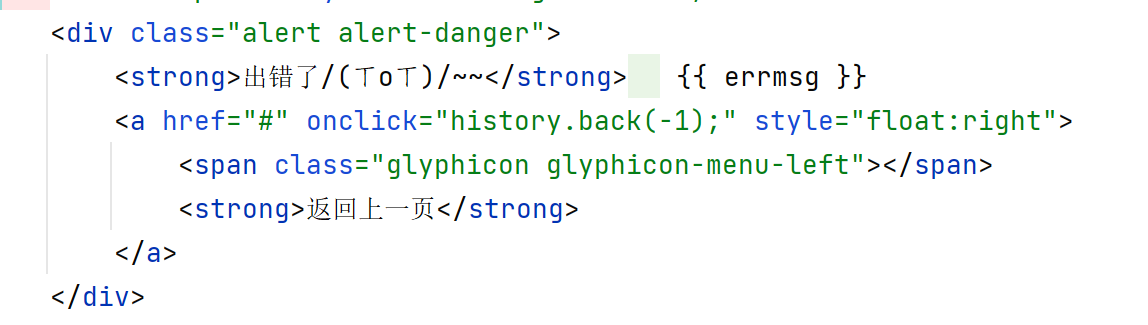


图4-2-16 错误页面提示

配置题库页面，选择题库时，利用for循环半段，如果选择的存放类型题库cbt在选定范围内且等价于应答的类型时，把应答的类型的id赋值给cbt。

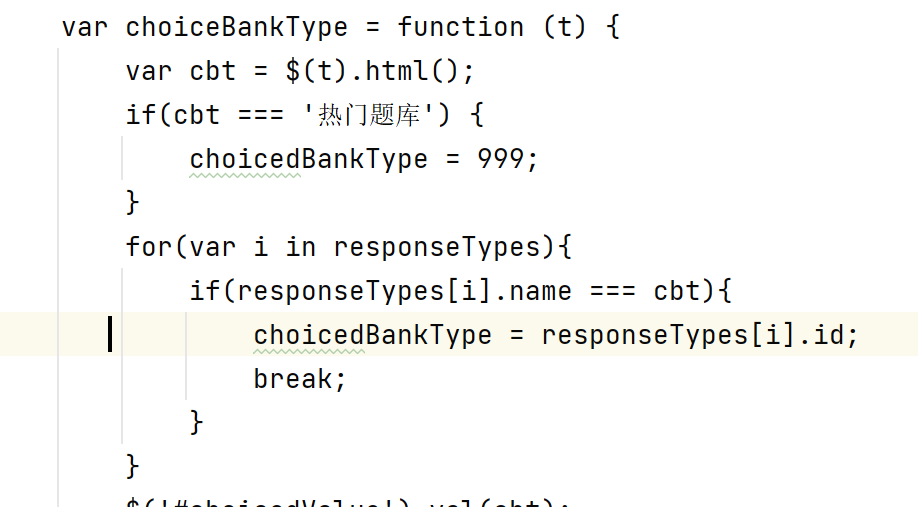


图4-2-17 选择题库类型

这里采用JQ的ajax（）方法通过http请求加载远程数据，通过指定的URL来跳转到指定的题库，这里的目的地址仍未当前配置比赛的页面。连接数据库类型采用默认的‘get’方法远程请求载入信息，当请求回应的状态为200时，可成功调用回调函数。这里返回的数据类型用json格式来处理，所以传入参数data{}为空。

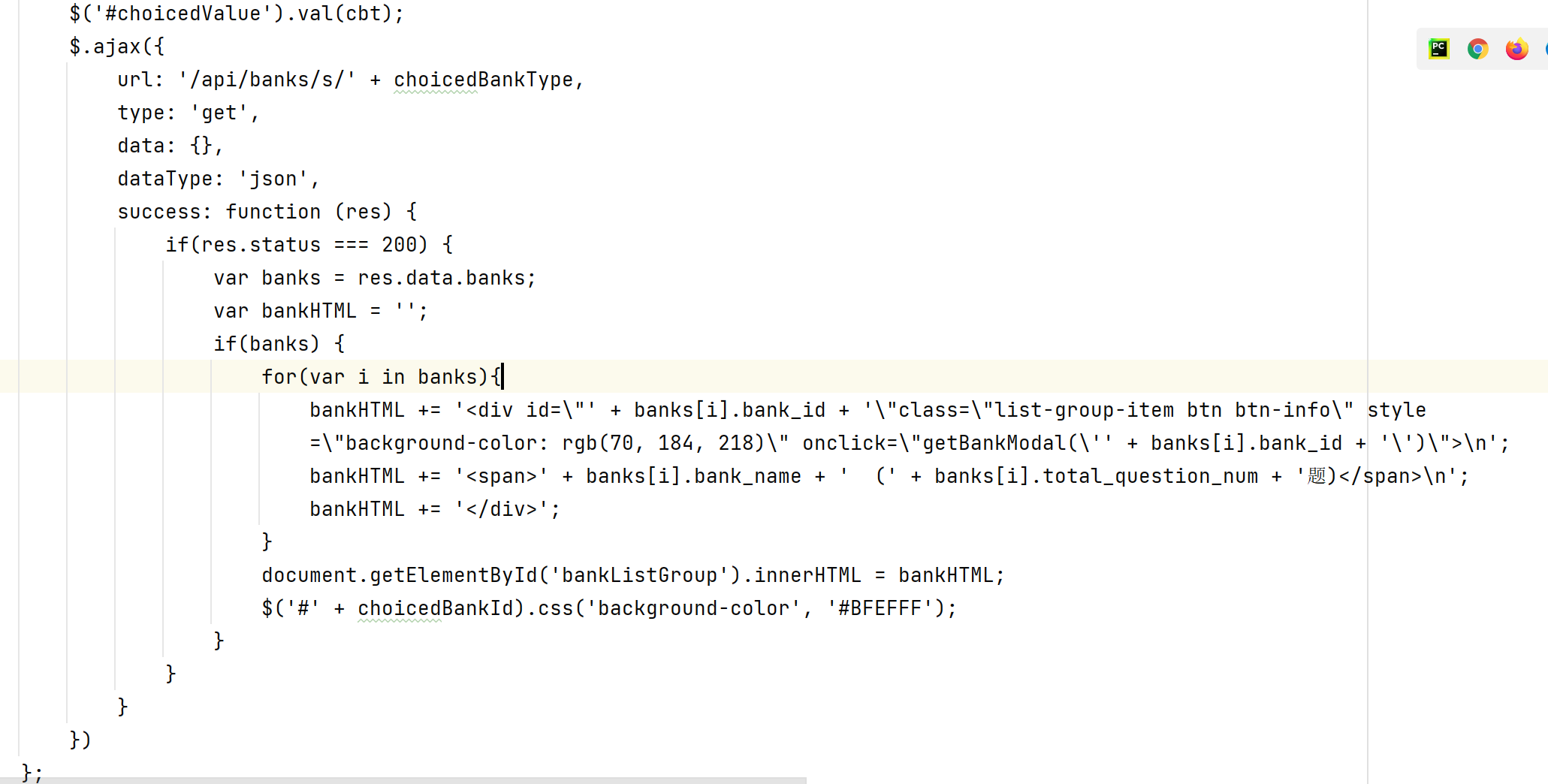


图4-2-18 JQ ajax 选择题库类型

选定好题库后并成功来连接后，会弹出该题库的详细页面，如图13所示，用一个无序列表<ul></ul>来列出题库的详细信息。每一个详细信息都是一个行内块元素第一列悬浮，第二列将左外边距设置为200，即与第一列元素间隔200px。最后设置点击按钮即可跳回父页面。



图4-2-19 题库信息效果图

在输入字段时，在已选定至少一个字段的前提下，才可以点击下拉框自行添加所需要的字段，这里采用的是if-else条件语句来判断用户输入，若没有勾选想要输入的字段，也不能在下拉框添加字段，在页面上会显现出红色的边框警告，返回错误，否则正确，可添加下拉框。自行添加字段不知一个时，根据提示信息用‘#’分隔开。

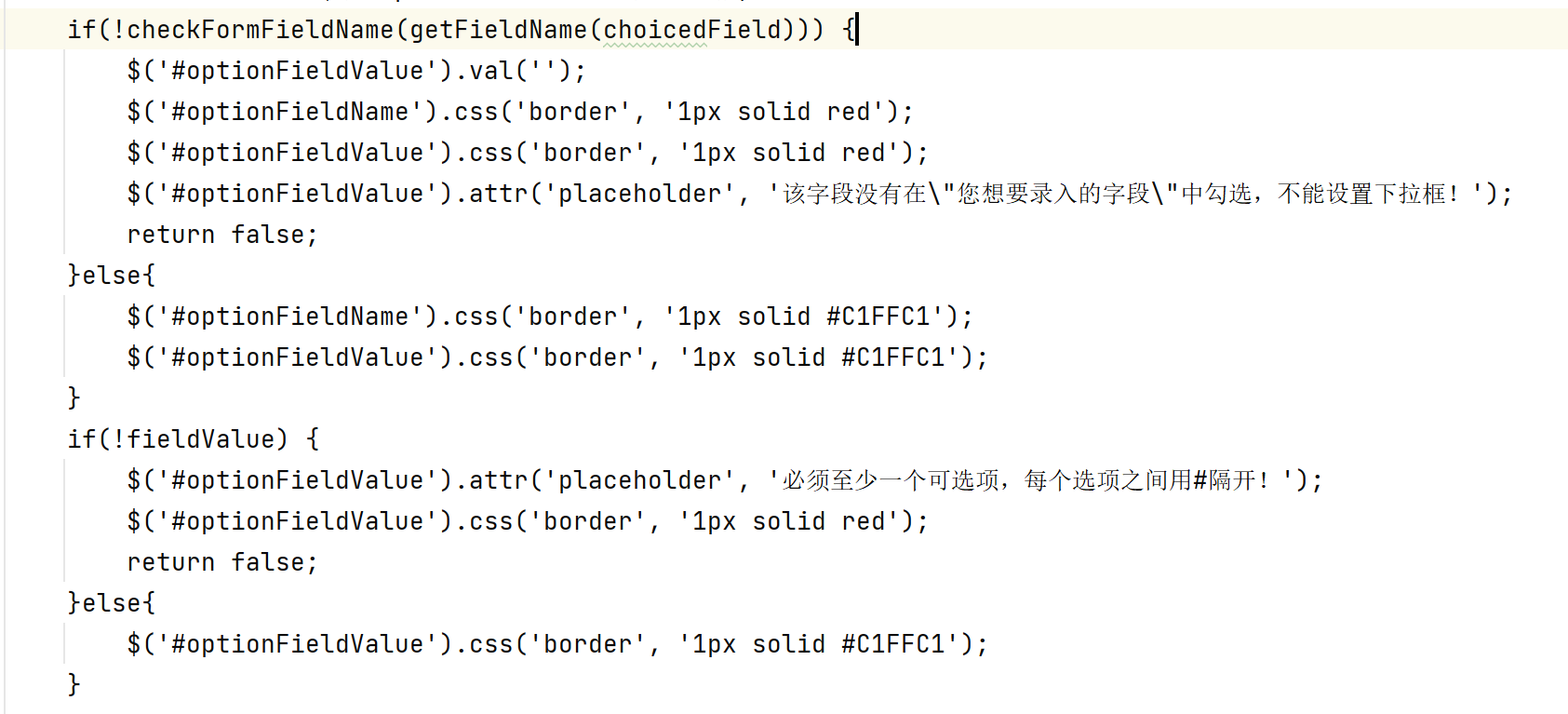


图4-2-20 下拉框字段

根据js的提示信息填写完整配置比赛的信息后，通过ajax的post()方法通过过http请求加载远程数据。这里的传入的参数不在为空，包括问题数量，参与测试的人数等等，当成功建立请求是返回回调函数。

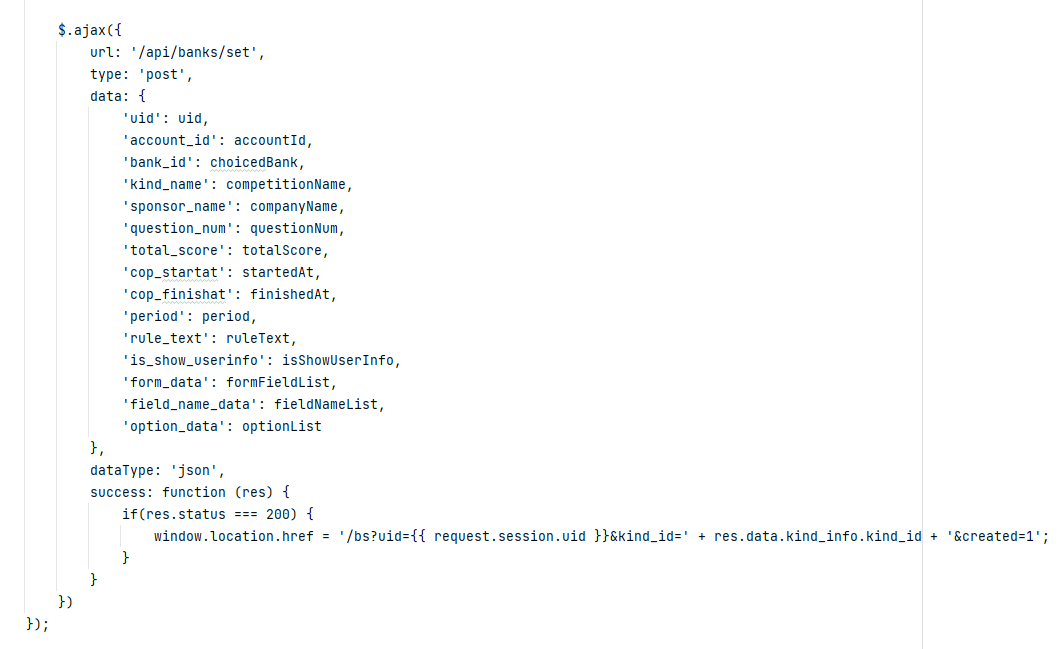


图4-2-21 建立配置比赛连接请求

## [4.3 管理员功能模块设计](#_Toc29730_WPSOffice_Level2)

4.3.1 管理员查看用户信息

管理员可以查看用户的某些信息，比如用户的唯一标识UID，用户来源(普通用户或机构用户)和性别，对于管理员也有权限限制，管理员不能查看用户的地址和微信号等隐私信息，这样也增加了系统的安全性和可靠性。



图4-3-1 管理员界面

4.3.2 管理员的筛选功能

管理员可以通过某些信息对用户进行筛选



图4-3-2 过滤器界面

4.3.3 管理员权限

管理员(Administrator)一般是指负责一定系统或者软件的维护或管理更新的实际个人或帐号,也有的是专门管理违反原则的。但是我们也要对管理员也设置了一定的权限，即管理员不能修改用户的信息，也不能查看用户的隐私信息，这对于用户的信息起到了保密作业，也符合系统的安全性。

后台可以查看uid，用户来源、姓名、性别、国家、身份、城市、是否升级会员、会员升级时间、会员过期时间、会员升级次数、创建时间、更新时间、状态。可以根据是否升级会员、性别、状态来筛选。管理员可以通过用户的uid和姓名来从查找。后台不能修改用户的唯一标识uid。



图4-3-3 用户信息后台配置

4.3.4角色设计

本系统所设计的权限设计三种角色，分别是普通用户、机构用户、管理员。

其中普通用户和机构用户之间的关系组成系统功能的基本逻辑。在用户注册时就会被分配到某个具体角色中。并且只能做出角色权限范围之内的操作。

它们之间的操作权限也需要进行细分，对于普通用户来说，能够对用户表、应用配置表、用户表单表都有或多或少的权限。对于机构用户来说，类似的对公司表、应用表、配置表有一定的操作权限。而管理员具有最高权限，能够对所有表的任意数据进行增删查改。但不能够操作服务器数据库中除该系统以外的表。设计该用户的目的是为了能够在非root权限下对数据表比较方便的进行操作。

具体的角色设计如下表所示

|  |  |  |
| --- | --- | --- |
| 角色名 | 权限 | 备注 |
| 普通用户 | 查改（普通用户，{密码、邮箱、身份证号码、姓名、性别、年龄、手机号、简历}）  增删查改（用户表单，{用户id、比赛id、配置名称}）  查（应用配置，{应用id、应用名、应用描述、比赛规则}） | 能够对用户表、用户表单有主动更改权限、对有被动更改权限 |
| 机构用户 | 查改（机构用户，{密码、邮箱、手机号、身份证号码、姓名、性别、年龄、}）  增删查（应用，{应用id、应用名、应用描述、比赛规则}，配置，{出题账户id，应用id、应用名、应用秒速、比赛规则、公司名称、公司类型、公司描述、公司位置、邮箱、联系电话}） | 能够对公司表、应用表、配置表有主动主动更改权限 |
| 管理员 | 全表增删查改 | 能够对比赛配置、应用配置有主动更改权限 |

[4.4 UI界面设计](#_Toc30207_WPSOffice_Level2)



图4-4-1 首页界面

底部导航栏部分，在系统中我们用fixed方法把他固定在页面的底部，然后用一个block模块把首页、联系我们等放在同一行内，通过点击可以连接到相应的页面上。



图4-4-2 底部导航栏

在除首页之外的页面，我们采用Django模板继承的方法，在页面头部添加{% extends 'base.html' %}，载入‘base’的模板中的内容到当前模板，仅需在当前模板中编写‘body’部分代码即可，这样大大减少了模板间的代码冗余。



图4-4-3 关于我们界面



图4-4-4 无法连接时界面

# 测试

在Web工程过程中，基于Web系统的测试、确认和验收是一项重要而富有挑战性的工作。基于Web的系统测试与传统的软件测试不同，它不但需要检查和验证是否按照设计的要求运行，而且还要测试系统在不同用户的浏览器端的显示是否合适。重要的是，还要从最终用户的角度进行安全性和可用性测试。然而，Internet和Web媒体的不可预见性使测试基于Web的系统变得困难。因此，我们必须为测试和评估复杂的基于Web的系统研究新的方法和技术。

本文将Web测试分为6个部分：功能测试、性能测试、用户界面测试、兼容性测试、安全测试、接口测试。

## 5.1功能测试

5.1.1链接测试

链接是Web应用系统的一个主要特征，它是在页面之间切换和指导用户去一些不知道地址的页面的主要手段。链接测试可分为三个方面。首先，测试所有链接是否按指示的那样确实链接到了该链接的页面；其次，测试所链接的页面是否存在；最后，保证Web应用系统上没有孤立的页面，所谓孤立页面是指没有链接指向该页面，只有知道正确的URL地址才能访问。

5.1.2表单测试

当用户通过表单提交信息的时候，都希望表单能正常工作。如果使用表单来进行在线注册，要确保提交按钮能正常工作，当注册完成后应返回注册成功的消息。如果使用表单收集配送信息，应确保程序能够正确处理这些数据，最后能让顾客收到包裹。要测试这些程序，需要验证服务器能正确保存这些数据，而且后台运行的程序能正确解释和使用这些信息。  
 当用户使用表单进行用户注册、登陆、信息提交等操作时，我们必须测试提交操作的完整性，以校验提交给服务器的信息的正确性。例如：用户填写的出生日期与职业是否恰当，填写的所属省份与所在城市是否匹配等。如果使用了默认值，还要检验默认值的正确性。如果表单只能接受指定的某些值，则也要进行测试。例如：只能接受某些字符，测试时可以跳过这些字符，看系统是否会报错。

5.1.3 Cookies测试  
 Cookies通常用来存储用户信息和用户在某应用系统的操作，当一个用户使用Cookies访问了某一个应用系统时，Web服务器将发送关于用户的信息，把该信息以Cookies的形式存储在客户端计算机上，这可用来创建动态和自定义页面或者存储登陆等信息。  
 如果Web应用系统使用了Cookies，就必须检查Cookies是否能正常工作。测试的内容可包括Cookies是否起作用，是否按预定的时间进行保存，刷新对Cookies有什么影响等。如果在Cookies中保存了注册信息，请确认该 Cookie能够正常工作而且已对这些信息已经加密。如果使用Cookie来统计次数，需要验证次数累计正确。

5.1.4数据库测试

在Web应用技术中，数据库起着重要的作用，数据库为Web应用系统的管理、运行、查询和实现用户对数据存储的请求等提供空间。在Web应用中，最常用的数据库类型是关系型数据库，可以使用SQL对信息进行处理。

数据库的Web应用系统中，一般情况下，可能发生两种错误，分别是数据一致性错误和输出错误。数据一致性错误主要是由于用户提交的表单信息不正确而造成的，而输出错误主要是由于网络速度或程序设计问题等引起的，针对这两种情况，可分别进行测试。

## 5.2性能测试

5.2.1连接速度测试  
 用户连接到Web应用系统的速度根据上网方式的变化而变化，他们或许是电话拨号，或是宽带上网。当下载一个程序时，用户可以等较长的时间，但如果仅仅访问一个页面就不会这样。如果Web系统响应时间太长（例如超过5秒钟），用户就会因没有耐心等待而离开。  
 另外，有些页面有超时的限制，如果响应速度太慢，用户可能还没来得及浏览内容，就需要重新登陆了。而且，连接速度太慢，还可能引起数据丢失，使用户得不到真实的页面。

5.2.2负载压力测试  
 在这里的负载\压力和功能测试中的不同,他是系统测试的内容,是基本功能已经通过后进行的.可以在集成测试阶段,亦可以在系统测试阶段进行。  
 使用负载测试工具进行,虚拟一定数量的用户看一看系统的表现,是否满足定义中的指标。负载测试一般使用工具完成，loadrunner，Webload，was，ewl，e-test等，主要的内容都是编写出测试脚本，脚本中一般包括用户一般常用的功能，然后运行，得出报告。负载测试技术在各种极限情况下对产品进行测试 (如很多人同时使用该软件，或者反复运行该软件)，以检查产品的长期稳定性。例如，使用压力测试工具对Web服务器进行压力测试. 本项测试可以帮助找到一些大型的问题，如死机、崩损、内存泄漏等，因为有些存在内存泄漏问题的程序，在运行一两次时可能不会出现问题，但是如果运行了成千上万次，内存泄漏得越来越多，就会导致系统崩滑。

## **5.3用户界面测试**

界面是软件与用户交互的最直接的层，界面的好坏决定用户对软件的第一印象。而且设计良好的界面能够引导用户自己完成相应的操作，起到向导的作用。同时界面如同人的面孔，具有吸引用户的直接优势。设计合理的界面能给用户带来轻松愉悦的感受和成功的感觉，相反由于界面设计的失败，让用户有挫败感，再实用强大的功能都可能在用户的畏惧与放弃中付诸东流。目前界面的设计引起软件设计人员的重视的程度还远远不够，直到最近网页制作的兴起，才受到专家的青睐。而且设计良好的界面由于需要具有艺术美的天赋而遭拒绝。

5.3.1窗口

1、窗口是否基于相关的输入和菜单命令适当地打开？  
2、窗口能否改变大小、移动和滚动？  
3、窗口中的数据内容能否用鼠标、功能键、方向键和键盘访问？  
4、当被覆盖并重新调用后，窗口能否正确地再生？  
5、需要时能否使用所有窗口相关的功能？  
6、所有窗口相关的功能是可操作的吗？  
7、是否有相关的下拉式菜单、工具条、滚动条、对话框、按钮、图标和其他控8、制可为窗口使用，并适当地显示？  
9、显示多个窗口时，窗口的名称是否被适当地表示？  
10、活动窗口是否被适当地加亮？  
11、如果使用多任务，是否所有的窗口被实时更新？  
12、多次或不正确按鼠标是否会导致无法预料的副作用？  
13、窗口的声音和颜色提示和窗口的操作顺序是否符合需求？  
14、窗口是否正确地被关闭？

## 5.4兼容性测试

5.4.1平台测试

市场上有很多不同的操作系统类型，最常见的有Windows、Unix、Macintosh、Linux等。Web应用系统的最终用户究竟使用哪一种操作系统，取决于用户系统的配置。这样，就可能会发生兼容性问题，同一个应用可能在某些操作系统下能正常运行，但在另外的操作系统下可能会运行失败。  
　　因此，在Web系统发布之前，需要在各种操作系统下对Web系统进行兼容性测试。

5.4.2浏览器测试  
 浏览器是Web客户端最核心的构件，来自不同厂商的浏览器对Java、JavaScript、ActiveX、plug-ins或不同的HTML规格有不同的支持。例如，ActiveX是Microsoft的产品，是为Internet Explorer而设计的，JavaScript是Netscape的产品，Java是Sun的产品等等。另外，框架和层次结构风格在不同的浏览器中也有不同的显示，甚至根本不显示。不同的浏览器对安全性和Java的设置也不一样。

测试浏览器兼容性的一个方法是创建一个兼容性矩阵。在这个矩阵中，测试不同厂商、不同版本的浏览器对某些构件和设置的适应性。

## 5.5安全测试

主要是测试系统在没有授权的情况下，内部或者外部用户对系统进行攻击或者恶意破坏时如何进行处理，是否仍能保证数据的安全。测试人员可以学习一些黑客技术，来对系统进行攻击。

5.5.1登录  
 有些站点需要用户进行登录，以验证他们的身份。这样对用户是方便的，他们不需要每次都输入个人资料。你需要验证系统阻止非法的用户名/口令登录，而能够通过有效登录。用户登录是否有次数限制? 是否限制从某些 IP 地址登录? 如果允许登录失败的次数为3，你在第三次登录的时候输入正确的用户名和口令，能通过验证吗? 口令选择有规则限制吗?是否可以不登陆而直接浏览某个页面？  
 Web应用系统是否有超时的限制，也就是说，用户登陆后在一定时间内（例如15分钟）没有点击任何页面，是否需要重新登陆才能正常使用。

## 5.6接口测试

数据一般通过接口输入和输出，所以接口测试是白盒测试的第一步。每个接口可能有多个输入参数，每个参数有“典型值”、“边界值”、“异常值”之分，所以输入的组合数可能并不少。根据接口的定义，可以推断某种输入应当产生什么样的输出。输出包括函数的返回值和输出参数。如果实际输出与期望的输出不一致，那么说明程序有错误。

5.6.1服务器接口  
 第一个需要测试的接口是浏览器与服务器的接口。测试人员提交事务，然后查看服务器记录，并验证在浏览器上看到的正好是服务器上发生的。测试人员还可以查询数据库，确认事务数据已正确保存。

5.6.2外部接口  
 有些Web系统有外部接口。例如，网上商店可能要实时验证信用卡数据以减少欺诈行为的发生。测试的时候，要使用Web接口发送一些事务数据，分别对有效信用卡、无效信用卡和被盗信用卡进行验证。

5.6.3错误处理  
 最容易被测试人员忽略的地方是接口错误处理。通常我们试图确认系统能够处理所有错误，但却无法预期系统所有可能的错误。尝试在处理过程中中断事务，看看会发生什么情况？订单是否完成？尝试中断用户到服务器的网络连接。尝试中断Web服务器到信用卡验证服务器的连接。在这些情况下，系统能否正确处理这些错误？是否已对信用卡进行收费？如果用户自己中断事务处理，在订单已保存而用户没有返回网站确认的时候，需要由客户代表致电用户进行订单确认。

# 存在问题及改进思路

## 6.1系统的安全性

存在SQL注入，危及服务安全。故此我们计划使用[orm](https://so.csdn.net/so/search?q=orm&spm=1001.2101.3001.7020)操作数据库增删改查:因为orm采用的是参数化形式执行SQL语句。在执行原生[SQL语句](https://so.csdn.net/so/search?q=sql%E8%AF%AD%E5%8F%A5&spm=1001.2101.3001.7020)时，不进行url拼接操作，而是使用参数化的形式。

## 6.2随机出题算法

在本系统中我们采用的是random.sample函数在题库中随机选取题目，这在一定程度上保证了用户所作题目的随机多样性。但运用random.sample函数也存在一个问题，对所选出的题目难度没有限定，不能让用户很好的根据自身需求选取当前所想要的难度等级，所以我们的改进思路是运用算法，对题目的难度等级分类，再从分类的题目中抽取一定数量的题目再整合成一份难度合适的测试题。

# 总结与发展展望

整个系统的设计过程对于我来说算是个学习、探索的过程，通过实践和对比别人开发程序的过程。在整个设计过程中，出现过很多的问题，很多繁琐的东西都需要反复的修改，主要是前期工作不彻底，对系统的需求分析的要求认识不够清楚，使得在后边的工作中不得不经常反复去修改。使我体会到设置中每一步的重要性。所以在分析一个问题时，我们需要站在一个有远见的高度。

## 7.1 遇到的困难

7.1.1 底部栏

在Web前端底部导航栏处，一开始出现紧贴页面底部问题，为了提高用户体验，利用类方法将footer固定在页面底部，但再不同电脑上布局存在差异，比如在进行响应式放大缩小时出现布局中的问题，后续通过CSS修饰改进解决了上述问题，提高了用户体验。

7.1.2 网页请求被拒绝

注册机构：点击注册按钮之后，有时候出现注册失败的情况。

在pycharm上配置相关的Django server进行debug，进行多重排查。如：认为自己不小心添加的功能导致的问题，但是通过git版本回退，发行以前就存在这个问题，只是没有发现。经过一番修改之后认为不是后端代码问题。

后面发现点击提交之后网页都会进行刷新，但是我们的代码并没有请求编写出错后网页重定向的操作，故此怀疑是网页的刷新导致前后端通信的失败（请问还未发出，网页就重定向，导致请求发出失败），并且通过网页的检查工具发现确实是后端return的信息被refuse。故此认为把解决思路转向ajax的请求，jQuery版本，crsf校验等等方向，依旧没有解决。

最后发现是HTML编写的时候，注册的相关信息设置为Form进行提交，并且注册的按钮被设为submit，故此会导致页面的刷新，将网页进行修改。将其进行修改，最终解决问题。

7.1.3 数据类型改变

在实现查看答题历史这一功能的时候，考虑到答题题目是从数据库中随机选取的，故为了保存答题历史，需要将相应的题目，答案，用户答题记录，正误情况等保存到对应的数据库中。

由于存在诸多信息，故计划采用key-value的方式进行存取。但是前端在读取存储的信息时，发现无法索引到相关的key，但是后端存储的数据并没有毛病。

通过一番探索发现，是原本的key-value在存储的时候被转化成了字符串，故在读取的时候需要进行额外的处理，将str转化为key-value才行。

7.1.4 路由请求

自己的views已经写了POST规则，但是在前段请求的时候得到的却是GET请求，或者得到的403页面！

发现问题：

1. POST请求url不是以/结尾。
2. urls.py路由设置中的路径没有以/结尾。

总的来说Django很多地方都有这个/陷阱！注意windows和linux的路径分割符是不一样的，如果开发和部署环境不一样，注意修改路径分隔符。

7.1.5 网页缓存

网页存在缓存，如果没有清除缓存，会保存原来的信息，导致前后端修改后，运行的依然是旧版本，需要强制刷新Ctrl+F5。

7.1.6 Redis连接失败，版本问题

使用Redis包进行Redis的连接，由于查找相关的教程的时候没有注意版本问题，使用旧版本的连接语法，导致Redis的连接失败。当时一度以为是Redis的配置问题，导致连接失败。

最后通过使用新的连接语法成功连接。

7.1.7 没有找到表

在model.py中建表后没有进行makemigrations 以及migrate的操作导致无法重新找到表。

7.1.8 前端页面出错

HTML中多一个或者少了一个标签导致的出错。

7.1.9 MySQL连接失败

第一次在Docker上配置MySQL，在配置信息无误的情况下依然连接不上。通过查找相关信息，发现在Docker上第一次配置MySQL的时候，即使有默认密码，还是需要重新修改。通过重新设置密码后，连接成功。

## 7.2 相关优化

7.2.1 注册成为机构的用户要求

一开始我们的系统对注册成为机构的用户没有要求，即任何人都可以成为用户，后来考虑到合理性，为了改进这个问题，我们的改进思路是：对注册成为机构的用户限制要求，只有在管理员允许机构的名单内才能注册为机构出题。这样才更加合理和安全。

7.2.2 系统的热度和可用度

如果没有分析该系统有没有流量，我们不知道该系统的受欢迎程度以及使用的人数，为了分析该系统有没有流量，我们改进方法是查看测试的使用人数，通过热度高低来判断该系统是否受欢迎。

**7.2.3用户体验**

在考虑到系统兼容性的问题，不同用户所用到的浏览器可能不同，我们的系统该如何保证能适用于大多数的浏览器，为此做了系统测试。在用户界面设计方面，我们采用直观的按钮、图像，便于用户快速找到对应的功能，增加用户体验。对于成绩排名方面，该如何对不同成绩等级划分不同的鼓励语句，我们也采用了比较客观的结果对用户成绩做出评价。另外系统导航栏我们固定在页面底部，用户可快速的跳转到首页、联系我们等页面。

7.2.4 前端界面的问题

系统名称和一些页面标题名称与功能内容不符合，发现了这个问题后，我们把“成为用户”改为“注册用户”，为了方便，在各个页面增加了导航栏，符合Web开发。根据成绩结果的提示我们一开始没有考虑全面，为了更加合理化，我们根据成绩排名改为不一样的提示，同时也是对测试用户的鼓励。为了前端美观，我们参考其他网址设置了美观合理的布局设计。

7.2.5 存储空间

对于配置比赛类型的选项，一个中文占两个字节，为了减少存储空间，我们把互联网选项设为0，金融设为1，能源设为2，基础建设设为3，交通设为4，通信设为5。这样方便了存储和减少了使用空间。

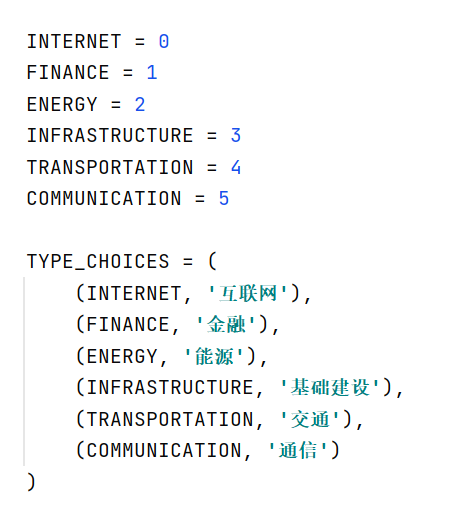


图7-2 降低存储大小

## 7.3感悟

1. 在遇到bug的时候都需要定位到报错的位置更容易解决报错。

2. 需要了解清楚整个代码的逻辑，而不是凭着自己的经验进行揣测，否则一点的理解错误就容易在后期因为功能增添导致代码错误。

3. 使用Django编写前后端，只能适合于短期，并且开发人员少的小项目，因为Django将前后端合并，但是这样任务分配不太清晰，前后端代码混杂在一起，容易导致前后端职责不清。相比于前后端分离的方式来说，主要做的是产品和面向用户群多，且开发人员多，职责清晰。而前后端分离，维护性更强。

4. 掌握好debug工具可以让debug事半功倍，例如使用网页的断点调试以及pycharm的Django引擎。

5. 做一个系统需要考虑很多东西，尤其是系统的稳健性和安全性，非常重要。

6. 使用git进行项目管理，可以大大降低工作的失误率，提高工作效率：版本回退，版本对比，团队合作等。

虽然时间紧迫，但是为了做好这个课设，我们学会了很多,也感到自身知识的贫乏,希望在日后的努力中能做出更完善的系统。

附录

## 1. 后端重要代码

相关代码量过大，故贴只取部分重点代码，具体代码请看提交的项目代码。

普通登录：



图8-1 普通登录

登录验证码获取：



图8-2登录验证码获取

发送邮件：



图8-3 发送邮件

注册校验：

图8-4注册校验

获取题目信息：



图8-5获取题目信息

提交答案：



图8-6提交答案

查看历史：



图8-7 查看历史

## 2. SQL语句

相关代码量过大，故贴只取部分重点代码，具体代码请看提交的项目代码。

**1. 帐户配置文件**

DROP TABLE IF EXISTS `account\_profile`;

CREATE TABLE `account\_profile` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`status` tinyint(1) NOT NULL,

`created\_at` datetime(6) NOT NULL,

`updated\_at` datetime(6) NOT NULL,

`uid` varchar(22) DEFAULT NULL,

`user\_src` int(11) NOT NULL,

`user\_status` int(11) NOT NULL,

`unionid` varchar(32) DEFAULT NULL,

`openid` varchar(32) DEFAULT NULL,

`wxid` varchar(32) DEFAULT NULL,

`name` varchar(32) DEFAULT NULL,

`email` varchar(40) DEFAULT NULL,

`sex` int(11) NOT NULL,

`age` int(11) NOT NULL,

`nickname` varchar(32) DEFAULT NULL,

`avatar` varchar(60) DEFAULT NULL,

`phone` varchar(11) DEFAULT NULL,

`country` varchar(32) DEFAULT NULL,

`province` varchar(32) DEFAULT NULL,

`city` varchar(32) DEFAULT NULL,

`location` varchar(60) DEFAULT NULL,

`is\_upgrade` int(11) NOT NULL,

`upgrade\_time` datetime(6) DEFAULT NULL,

`expire\_time` datetime(6) DEFAULT NULL,

`upgrade\_count` int(11) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `uid` (`uid`),

UNIQUE KEY `account\_profile\_name\_email\_52e79ff2\_uniq` (`name`,`email`),

KEY `account\_profile\_status\_cef42cf5` (`status`),

KEY `account\_profile\_user\_src\_28c2847b` (`user\_src`),

KEY `account\_profile\_unionid\_1021e9fa` (`unionid`),

KEY `account\_profile\_openid\_06478752` (`openid`),

KEY `account\_profile\_name\_13db32ec` (`name`),

KEY `account\_profile\_email\_df723ccb` (`email`),

KEY `account\_profile\_phone\_923990b3` (`phone`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8;

**2.机构配置文件:**

DROP TABLE IF EXISTS `business\_appconfiginfo`;

CREATE TABLE `business\_appconfiginfo` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`status` tinyint(1) NOT NULL,

`created\_at` datetime(6) NOT NULL,

`updated\_at` datetime(6) NOT NULL,

`app\_id` varchar(32) NOT NULL,

`app\_name` varchar(40) DEFAULT NULL,

`rule\_text` longtext,

`is\_show\_userinfo` tinyint(1) NOT NULL,

`userinfo\_fields` varchar(128) DEFAULT NULL,

`option\_fields` varchar(128) DEFAULT NULL,

`userinfo\_field\_names` varchar(128) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `business\_appconfiginfo\_status\_23f940ab` (`status`),

KEY `business\_appconfiginfo\_app\_id\_2366db42` (`app\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8;

**3. 选择题题库**

DROP TABLE IF EXISTS `competition\_choiceinfo`;

CREATE TABLE `competition\_choiceinfo` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`status` tinyint(1) NOT NULL,

`created\_at` datetime(6) NOT NULL,

`updated\_at` datetime(6) NOT NULL,

`image\_url` varchar(255) DEFAULT NULL,

`audio\_url` varchar(255) DEFAULT NULL,

`audio\_time` int(11) NOT NULL,

`bank\_id` varchar(32) DEFAULT NULL,

`ctype` int(11) NOT NULL,

`question` varchar(255) DEFAULT NULL,

`answer` varchar(255) DEFAULT NULL,

`item1` varchar(255) DEFAULT NULL,

`item2` varchar(255) DEFAULT NULL,

`item3` varchar(255) DEFAULT NULL,

`item4` varchar(255) DEFAULT NULL,

`source` varchar(255) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `competition\_choiceinfo\_status\_dd6aa9e9` (`status`),

KEY `competition\_choiceinfo\_bank\_id\_019a8691` (`bank\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=268 DEFAULT CHARSET=utf8;

**4. 用户答题历史**

DROP TABLE IF EXISTS `competition\_competitionqainfo`;

CREATE TABLE `competition\_competitionqainfo` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`status` int(11) NOT NULL,

`created\_at` datetime(6) NOT NULL,

`updated\_at` datetime(6) NOT NULL,

`kind\_id` varchar(32) DEFAULT NULL,

`qa\_id` varchar(22) DEFAULT NULL,

`uid` varchar(32) DEFAULT NULL,

`qsrecord` longtext,

`asrecord` longtext,

`aslogrecord` longtext,

`started\_stamp` bigint(20) NOT NULL,

`finished\_stamp` bigint(20) NOT NULL,

`expend\_time` int(11) NOT NULL,

`started` tinyint(1) NOT NULL,

`finished` tinyint(1) NOT NULL,

`correct\_num` int(11) NOT NULL,

`incorrect\_num` int(11) NOT NULL,

`total\_num` int(11) NOT NULL,

`score` double NOT NULL,

`correct\_list` varchar(10000) DEFAULT NULL,

`wrong\_list` varchar(10000) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `competition\_competitionqainfo\_kind\_id\_10b82c49` (`kind\_id`),

KEY `competition\_competitionqainfo\_qa\_id\_3e2673ba` (`qa\_id`),

KEY `competition\_competitionqainfo\_uid\_d8905edc` (`uid`),

KEY `competition\_competitionqainfo\_started\_f1ffb449` (`started`),

KEY `competition\_competitionqainfo\_finished\_b77ac7ad` (`finished`)

) ENGINE=InnoDB AUTO\_INCREMENT=77 DEFAULT CHARSET=utf8;

**5. 填空题题库**

DROP TABLE IF EXISTS `competition\_fillinblankinfo`;

CREATE TABLE `competition\_fillinblankinfo` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`status` tinyint(1) NOT NULL,

`created\_at` datetime(6) NOT NULL,

`updated\_at` datetime(6) NOT NULL,

`image\_url` varchar(255) DEFAULT NULL,

`audio\_url` varchar(255) DEFAULT NULL,

`audio\_time` int(11) NOT NULL,

`bank\_id` varchar(32) DEFAULT NULL,

`ctype` int(11) NOT NULL,

`question` varchar(255) DEFAULT NULL,

`answer` varchar(255) DEFAULT NULL,

`source` varchar(255) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `competition\_fillinblankinfo\_status\_903a1b01` (`status`),

KEY `competition\_fillinblankinfo\_bank\_id\_c7404b1d` (`bank\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=37 DEFAULT CHARSET=utf8;

**6. 管理日志表**

DROP TABLE IF EXISTS `django\_admin\_log`;

CREATE TABLE `django\_admin\_log` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`action\_time` datetime(6) NOT NULL,

`object\_id` longtext,

`object\_repr` varchar(200) NOT NULL,

`action\_flag` smallint(5) unsigned NOT NULL,

`change\_message` longtext NOT NULL,

`content\_type\_id` int(11) DEFAULT NULL,

`user\_id` int(11) NOT NULL,

PRIMARY KEY (`id`),

KEY `django\_admin\_log\_content\_type\_id\_c4bce8eb\_fk\_django\_co` (`content\_type\_id`),

KEY `django\_admin\_log\_user\_id\_c564eba6\_fk\_auth\_user\_id` (`user\_id`),

CONSTRAINT `django\_admin\_log\_content\_type\_id\_c4bce8eb\_fk\_django\_co` FOREIGN KEY (`content\_type\_id`) REFerENCES `django\_content\_type` (`id`),

CONSTRAINT `django\_admin\_log\_user\_id\_c564eba6\_fk\_auth\_user\_id` FOREIGN KEY (`user\_id`) REFerENCES `auth\_user` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=12 DEFAULT CHARSET=utf8;

**7. Django session表**

DROP TABLE IF EXISTS `django\_session`;

CREATE TABLE `django\_session` (

`session\_key` varchar(40) NOT NULL,

`session\_data` longtext NOT NULL,

`expire\_date` datetime(6) NOT NULL,

PRIMARY KEY (`session\_key`),

KEY `django\_session\_expire\_date\_a5c62663` (`expire\_date`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## 3. 小组会议

一共进行6次小组会议，小组会议会议记录如下。

**数据库系统课程设计小组会议记录**

**课题名称：**智慧校园在线考试系统 **第\_1\_次会议记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **主 持 人：** | 姓名 \*\* | **记录人员：** | 姓名 \*\* |
| **会议主题：** | 选定主题与任务布置 | **会议地点：** | 5404 |
| **会议时间：** | 日期：2022.06.27  时间：10：30-13：00 | **整理时间：** | 日期：2022.06.27  时间：14:30-16：00 |
| **参与人员：** | 姓名 \*\*、\*\*、\*\* | | |
| **会议内容** | | | |
| 1．确定小组组长为\*\*  2．确定选题为智慧校园在线考试系统以及做这个系统的目的意义。  3．小组讨论使用Django作为编写框架。使用Django框架进行前后端的编写。  使用原生HTML，CSS，JavaScript等编写前端。  4．确定小组各成员的分工，\*\*负责协调整个项目的进度，分配任务，负责相关运行环境（MySQL，Redis，服务器）的搭建，答题，邮箱验证码，登录限制等后端功能的实现。\*\*负责机构注册、答题测试、答题结果和排行榜的页面设计和功能的实现，\*\*同学负责首页、注册登录、配置比赛页面的设计和登录、注册和配置比赛功能的实现。  5．大致规划时间安排表 | | | |
| **已解决问题：** | | | |
| 1．确定课设的选题  2．小组成员以及每个成员的任务分配  3．规划时间安排表以及小组集体讨论的时间 | | | |
| **未解决问题：** | | | |
| 1．课设各个技术的具体实现方法  2．如果把数据各个表合理的联立以及它所用到的技术  3．如何把数据可视化，还没有确定使用前端还是其它方法把数据呈现出来 | | | |
| **备注：** | | | |
| 智慧校园在线考试系统，Django，MySQL、Web、Redis  **组长：200031708\*\*** | | | |

**数据库课程设计小组会议记录**

**课题名称：**智慧校园在线考试系统 **第\_2\_次会议记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **主 持 人：** | 姓名 \*\* | **记录人员：** | 姓名 \*\* |
| **会议主题：** | 课设任务需求分析 | **会议地点：** | 5教咖啡厅 |
| **会议时间：** | 日期：2022.06.28  时间：10：00-12：00 | **整理时间：** | 日期：2022.06.29  时间：11：00-12：30 |
| **参与人员：** | 姓名 \*\*、\*\*、\*\* | | |
| **会议内容** | | | |
| 1．确定智慧校园在线考试系统的主要功能。  2．大致描绘出整个系统的主要框架，分析各个实体的属性已经他们之间的联系。  3．按照前两点的讨论，画出E-R图，以及系统的在工作流程图，最终敲定确定系统的功能。  4．分析数据来源，以及如何把各个数据表进行连接，将数据可视化。  5．分析各个部分所用的实验环境。  6. 安全性分析，如何从信息安全专业的角度去剖析该系统的安全性问题。 | | | |
| **已解决问题：** | | | |
| 1．确定智慧校园在线考试系统的主要功能以及做这个系统的目的意义  2．敲定工作流程图以及该系统的主要功能  3. 系统可能受到的攻击。 | | | |
| **未解决问题：** | | | |
| 1．数据的可视化，如何把表中数据在界面中呈现出来  2．E-R图暂时还不能完整的画出来  3．如何正地编写需求分析报告  4．系统如何防御外来的可能攻击 | | | |
| **备注：** | | | |
| 需求分析，系统流程及功能，数据可视化  **组长：200031708\*\*** | | | |

**数据库课程设计小组会议记录**

**课题名称：**智慧校园在线考试系统 **第\_3\_次会议记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **主 持 人：** | 姓名 \*\* | **记录人员：** | 姓名 \*\* |
| **会议主题：** | 系统设计 | **会议地点：** | 5教咖啡厅 |
| **会议时间：** | 日期：2022.07.02  时间：14：30-16：30 | **整理时间：** | 日期：2022.07.02  时间：16：00-17：00 |
| **参与人员：** | 姓名 \*\*、\*\*、\*\* | | |
| **会议内容** | | | |
| 1．修改小组分工表以及重新确定需求分析报告的编写  2．再次明确考试系统是倾向于在线测试而不是有教师端参与考试系统  3．再次进行系统防御分析，使用密码加密，邮箱验证，数据库备份，验证码等方式保证系统的安全  4．相关运行环境以及编译器，解释器的配置  5. 数据库的搭建 | | | |
| **已解决问题：** | | | |
| 1．小组分工表以及需求分析报告的修改  2．E-R图的绘制与再次确定该系统为个人使用的在线测试系统  3．确定系统的防御方案。  4. 选用MySQL以及Reids作为数据库，进行环境的搭建以及数据库的创建 | | | |
| **未解决问题：** | | | |
| 1．代码的编写与数据的导入  2．\*\*同学的环境搭建出问题，仍在解决 | | | |
| **备注：** | | | |
| E-R图，系统防御，环境配置  **组长：200031708\*\*** | | | |

**数据库课程设计小组会议记录**

**课题名称：**智慧校园在线考试系统 **第\_4\_次会议记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **主 持 人：** | 姓名 \*\* | **记录人员：** | 姓名 \*\* |
| 系统设计 | 详细设计和编码 | **会议地点：** | 5教咖啡厅 |
| **会议时间：** | 日期：2022.07.03  时间：14：30-16：30 | **整理时间：** | 日期：2022.07.03  时间：15：30-16：30 |
| **参与人员：** | 姓名 \*\*、\*\*、\*\* | | |
| **会议内容** | | | |
| 1．区分注册分机构和普通用户  2．如何实现认证过的机构才能出题  3．讨论如何随机出题  4．考试结果的不合理提示删掉  5．如何实现机构统计有多少人参与测试，以确定有没有流量  6．需修改该测试系统的名称  7．一些功能名称需要修改为简单理解的名称  8．测试时间不需要填入年月日 | | | |
| **已解决问题：** | | | |
| 1．测试系统名称已修改  2．修改了不易理解的功能名称  3．删除了测试结果中不合理的提示  4．系统的安全性分析 | | | |
| **未解决问题：** | | | |
| 1．如何实现机构统计有多少人参与测试  2．\*\*的环境出现了问题 | | | |
| **备注：** | | | |
| **组长：200031708\*\*** | | | |

**数据库课程设计小组会议记录**

**课题名称：**智慧校园在线考试系统 **第\_5\_次会议记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **主 持 人：** | 姓名 刘洪波 | **记录人员：** | 姓名 \*\* |
| **会议主题：** | 课程设计报告写作详解 | **会议地点：** | 5404 |
| **会议时间：** | 日期：2022.07.06  时间：10：30-12：00 | **整理时间：** | 日期：2022.07.06  时间：12：00-13：00 |
| **参与人员：** | 姓名 \*\*、\*\*、\*\* | | |
| **会议内容** | | | |
| 1．讲解课程设计报告写作  2．对报告的流程以及图表的讲解  3．报告的写作规范  4．报告的注意事项  5．讲解验收的基本流程要求 | | | |
| **已解决问题：** | | | |
| 1．课程设计报告写作的基本要求以及一些注意事项  2．课程设计报告的写作规范  3．报告的基本编写 | | | |
| **未解决问题：** | | | |
| 1．报告需要的图表  2．报告整合 | | | |
| **备注：** | | | |
| 课程设计报告写作  **组长：200031708\*\*** | | | |

**数据库课程设计小组会议记录**

**课题名称：**智慧校园在线考试系统 **第\_6\_次会议记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **主 持 人：** | 姓名 \*\* | **记录人员：** | 姓名 \*\* |
| **会议主题：** | 存在问题及改进思路 | **会议地点：** | 5教咖啡厅 |
| **会议时间：** | 日期：2022.07.10  时间：9:00-10:00 | **整理时间：** | 日期：2022.07.10  时间：11:00-12:00 |
| **参与人员：** | 姓名 \*\*、\*\*、\*\* | | |
| **会议内容** | | | |
| 1．注册机构成为机构用户的要求  2．分析系统的热度和可用度  3．随机出题算法  4．用户体验  5．微信、QQ授权登录更加方便  6．用户注册为会员  7．网站的盈利模式 | | | |
| **已解决问题：** | | | |
| 1．只有在后台管理员认证的邮箱才能注册为机构用户  2．通过统计测试的次数来评估系统有没有流量  3．通过美观的用户界面和测试系统的兼容性增加用户体验  4．通过收取用户会员服务费以及用户自愿捐赠盈利 | | | |
| **未解决问题：** | | | |
| 1．随机出题算法  2．微信、QQ授权登录  3．用户注册为会员 | | | |
| **备注：** | | | |
| 现存问题及改进  **组长：200031708\*\*** | | | |