# 设计目标

# 1项目目标

设计实现基于微信服务号的在线答题系统，答题系统分电脑管理端和

手机答题端。管理端管理账号题库和创建考试，答题端主要用于候选

人在线答题。

2项目背景

随着杭州成功举办 G20，杭州俨然成为众多应届大学生就业的首

选之地。因此每年的春招和秋招，也成为了杭城各大企业 HR 们最忙

碌、最累的季节，需要往全国各高校奔波开展一系列的招聘工作。为

了减轻 HR 们、面试官们招聘面试工作，提高候选人的面试体验，我

们期望实现基于微信服务号的在线答题系统

3项目概述

基于微信服务号的在线答题系统，主要是为了实现无纸化在线考

试效果，通过建立这样一个系统平台，能够帮助企业跨越空间、时间

障碍安排候选人进行实时答题、面试官阅卷等。同时通过该系统，可

大大方便 HR 对面试进程的把控，提高效率和节省不必要的开支。

线答题系统的 HR 后台管理系统、面试官阅卷系统通过 URL 绑定在微

信服务号菜单里，HR 和面试官可通过点击菜单控制考试进程和阅卷

等功能。候选人可通过微信扫描 HR 提供的二维码，局限于在微信浏

览器内答题(如果候选人复制链接到外部浏览器，应提示该 URL 只能

在微信浏览器内打开，可通过调微信接口实现)

1. 设计思路

1．HR 后台管理平台

HR 后台管理平台，可允许 HR 通过 WEB 登陆管理平台输入用户名、

密码后，登陆系统设置题库信息、面试官信息、考场信息等；也可允

许 HR 通过微信服务号点击登陆菜单，进入手机端页面进行开始考试

操作。

1.1 WEB 端题库信息

提供题库的新增、编辑、删除功能。同时，提供题目的新增、编

辑、删除功能。题目主要有单选题、多选题、判断题、主观题/翻译

题(加分题)，HR 可以设置每道题的分数；HR 可在设置题库的时候配

置一套题库的总分数，HR 可设置某个题库的题目来源是确定的或者

随机生成的，若题库的题目是随机生成的，系统会根据总分数去随机

搭配题目（该功能可更灵活点） 。

1.2 WEB 端题库自动演化

每道题有初始的难度判定，之后根据候选人的答题情况自动对题目的

难易判定进行修改。

1.3 WEB 端面试官信息

提供面试官的新增、编辑、删除功能等

1.4 WEB 端考场信息

提供考场的新增、编辑、删除功能。HR 设置考场信息的时候，

可为该考场指定确定的题库、或者选择多个题库并允许系统随机选择

题库(意味着不同的候选人扫描答题二维码得到的题库会有所不同)；

HR 也可以设置对应考场的面试官信息；另外 HR 设置考场信息的时候，

也需要设置考试开始时间、结束时间。

1.5 WEB 端统计分析

提供在不同考场每道题的准确率统计并能生成报表页面；提供候

选人个人信息的汇总查询导出功能等；提供面试官阅卷、面试的情况

汇总、查询等。

1.5 手机端考试设置

HR 可点击微信服务号里的菜单，通过输入用户名、密码后，登

陆手机端考试设置页面，可设置对应考场的开始、结束时间；可生成

对应考场的二维码供候选人扫描答题；可重置面试官阅卷状态

面试官阅卷功能需求介绍)；可通过姓名查询候选人个人信息和

答卷信息，也可通过成绩等查询一批符合条件的候选人等

2．微信端答题

2.1 候选人基本信息录入

候选人通过微信扫描考场答题二维码后，系统会提示候选人授权

获取候选人微信帐号名称，然后系统显示页面要求候选人输入个人基

本信息：姓名、性别、电话号码、学校、专业、学历等个人信息，然

后点击提交后，显示答题开始倒计时页面

2.2 开始/结束答题

当系统时间到达 HR 预设时间时，系统开始计时允许候选人开始

答题。并在即将结束的时候，系统自动帮助候选人提交答卷

3．面试官阅卷功能

3.1 个人信息修改

面试官通过点击微信服务号对应菜单，输入用户名、密码后登陆

系统，可维护个人信息。

3.2 阅卷功能

某考场面试官，可以查看当前考场所有候选人已经提交的答卷。

系统应该提供页面供面试官选择答卷，当面试官选中某一答卷后，该

答卷将被马上锁定，不能被其他面试官修改，其他面试官只能查看答

卷情况；当面试官提交阅卷结果后，该答卷是阅卷完成状态；当面试

官长时间没有提交阅卷结果，HR 可重置该阅卷状态供其他面试官阅

卷。

3.3 面试功能

面试官在面试的过程中，可登陆上述系统，通过候选人姓名查询

候选人的个人信息和答卷情况，方便开展面试

3.4历史统计功能

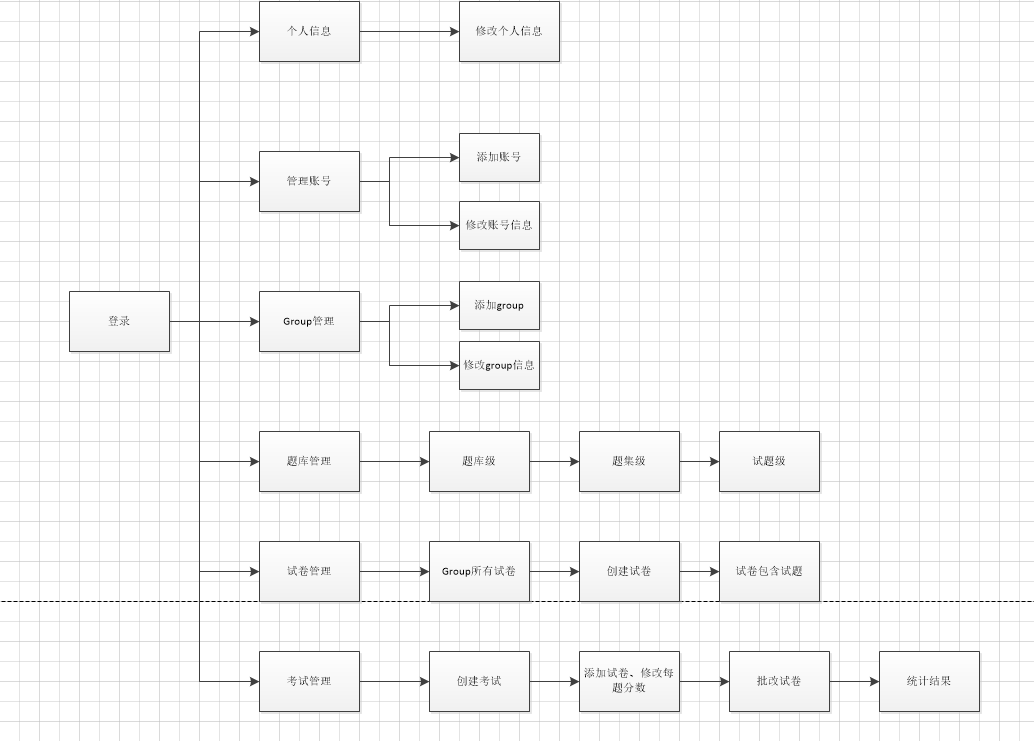
面试官在查询候选人信息时，会在展示候选人信息的同时显示候选人在

本平台的历史答题情况和面试官批改每张试卷的评价，统计历史答题的

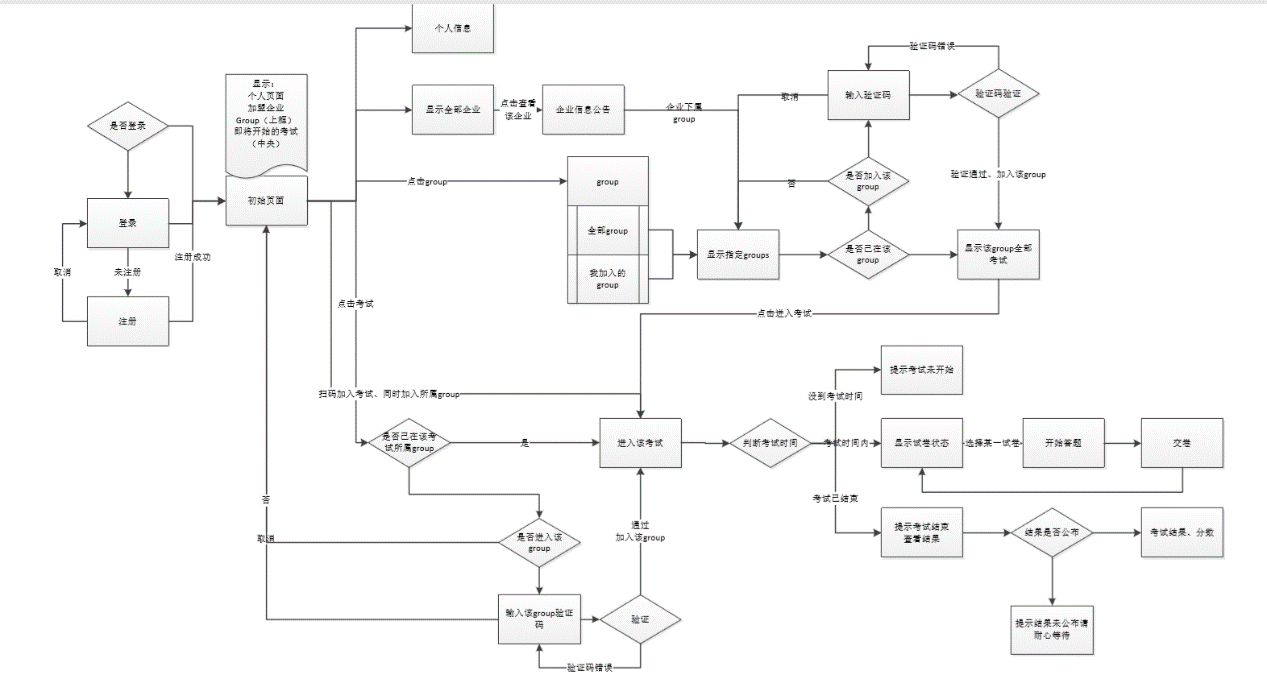
数据，分析出该候选人的一些特征。

1. 详细设计
2. 项目流程图

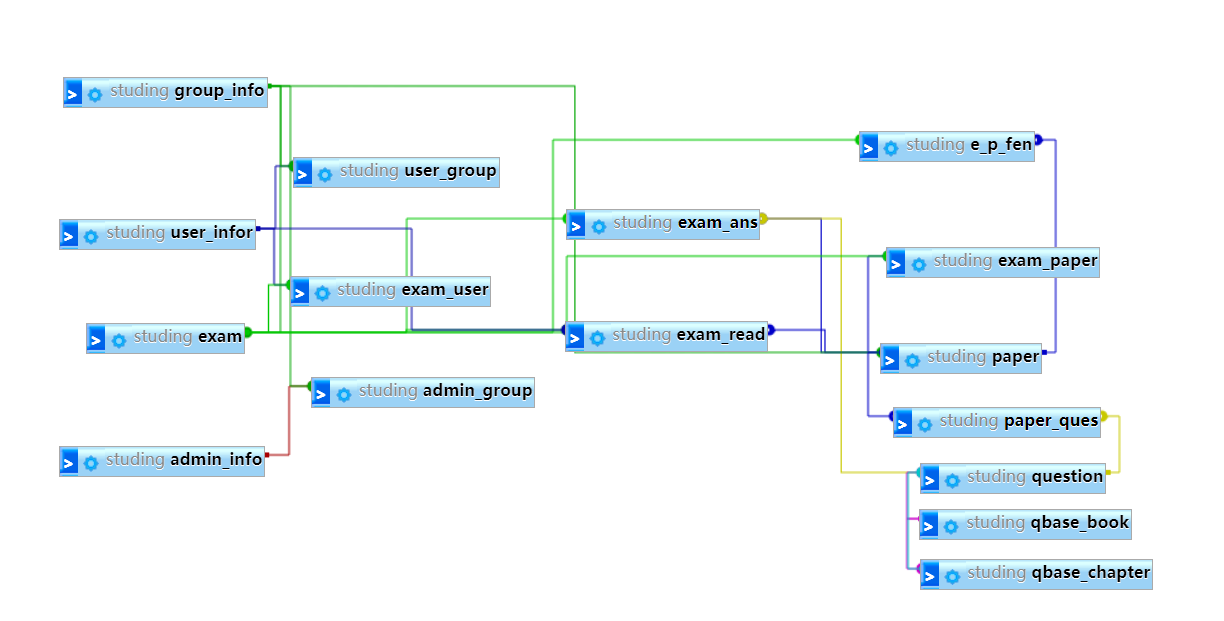
1.1管理端流程：



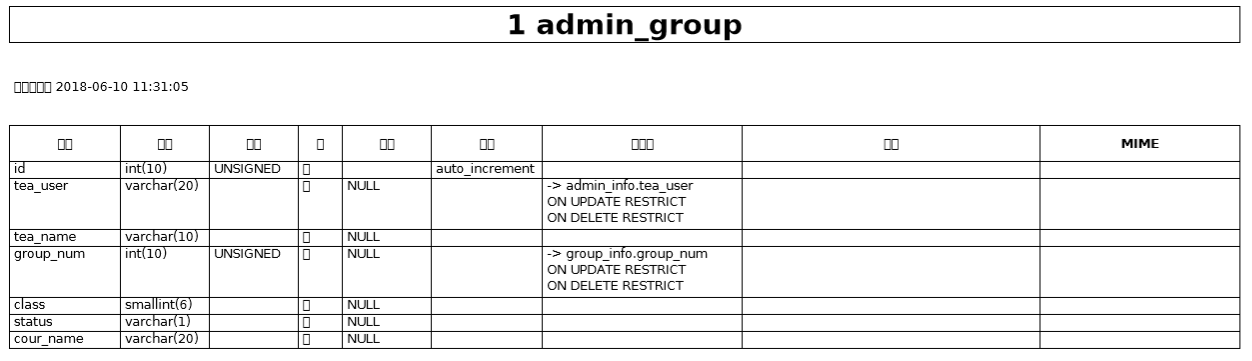
1.2答题端流程：



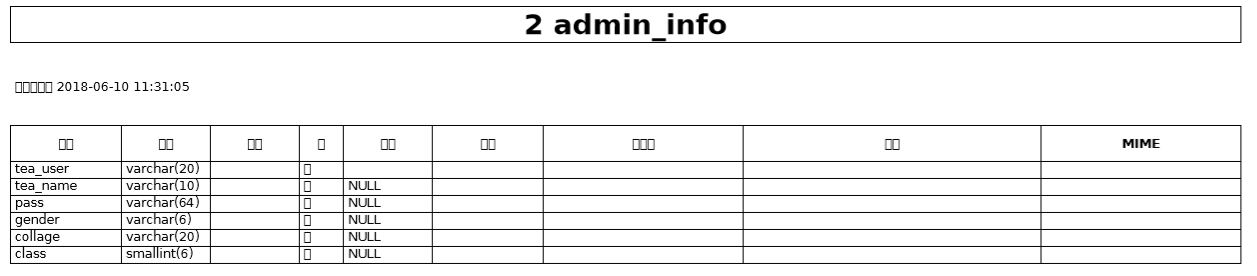
1. 数据库建立



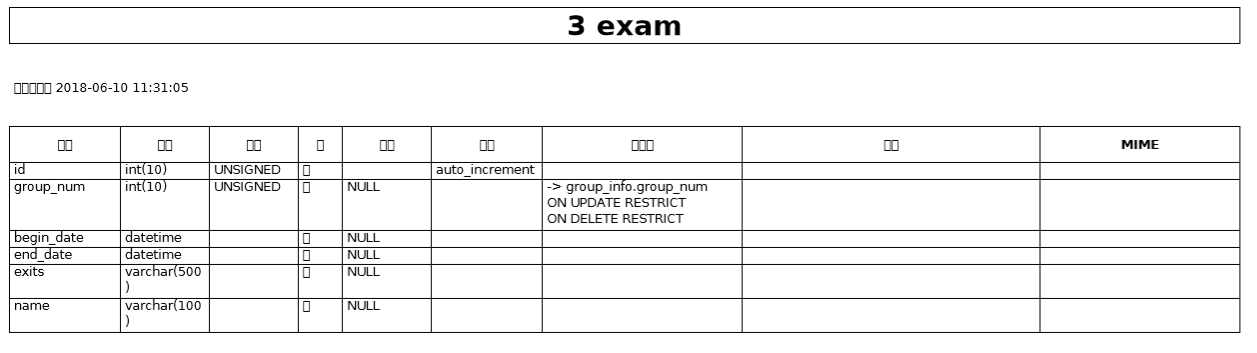
2.1.用于记录面试官与group之间的联系



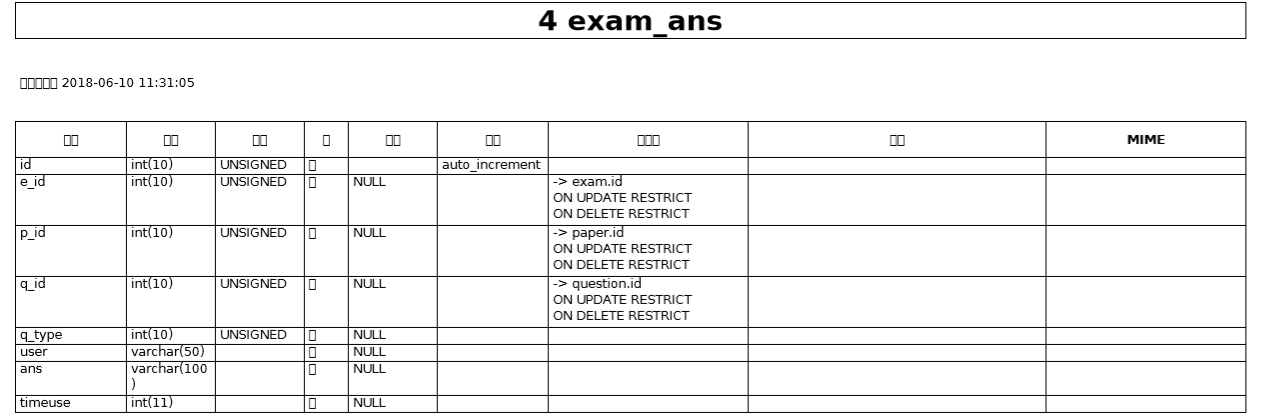
2.2.记录面试官，超级管理员的账号信息



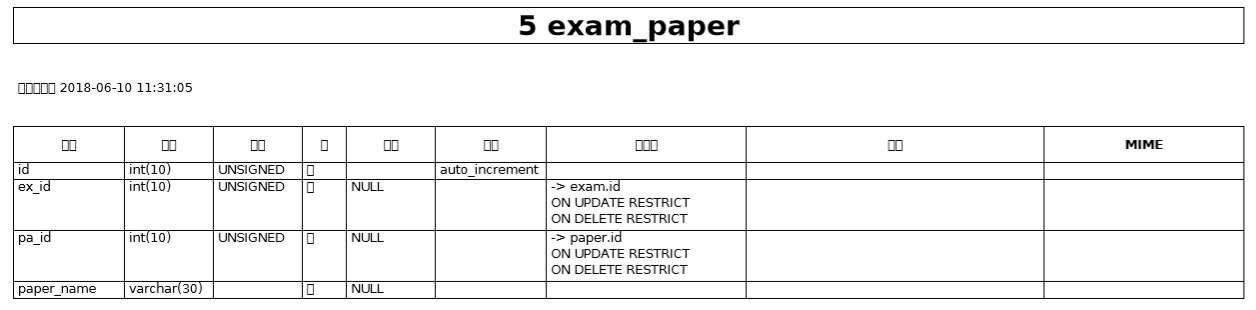
2.3.记录考试信息



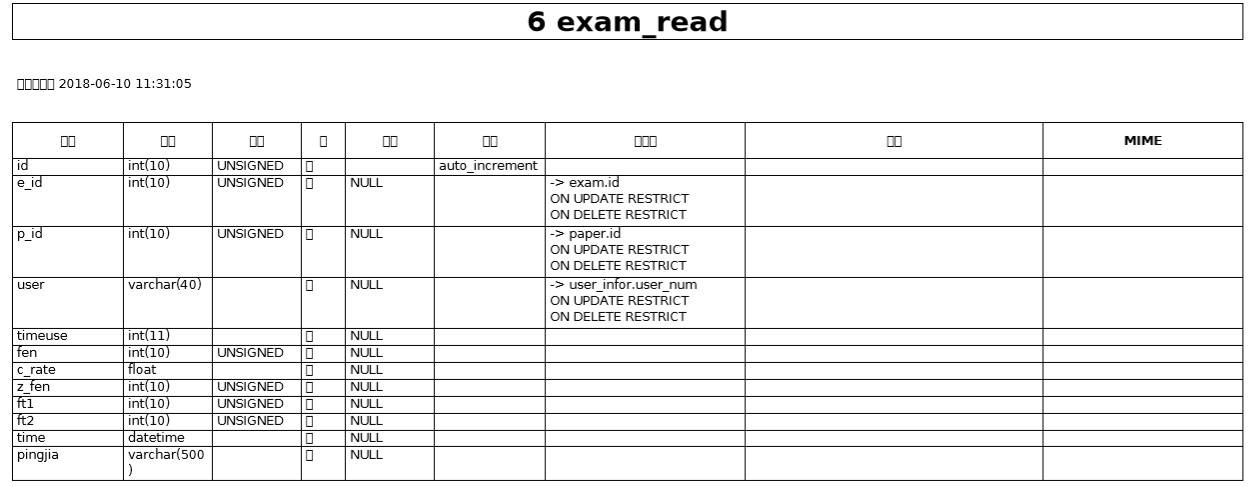
2.4.记录答题信息



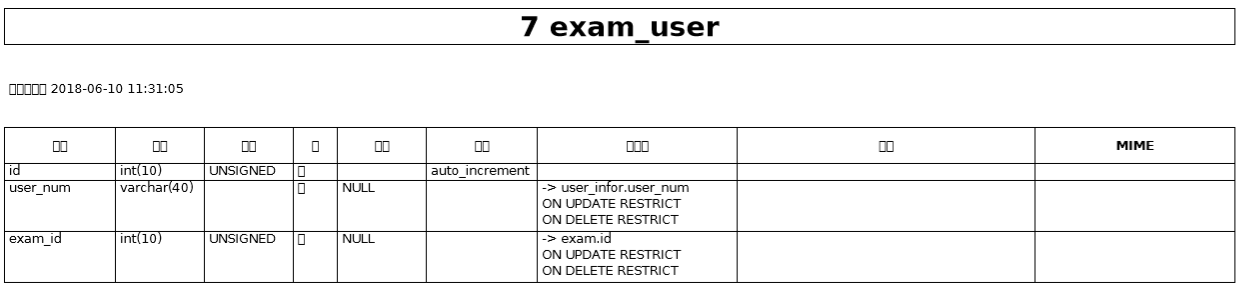
2.5.记录考试对应的试卷信息



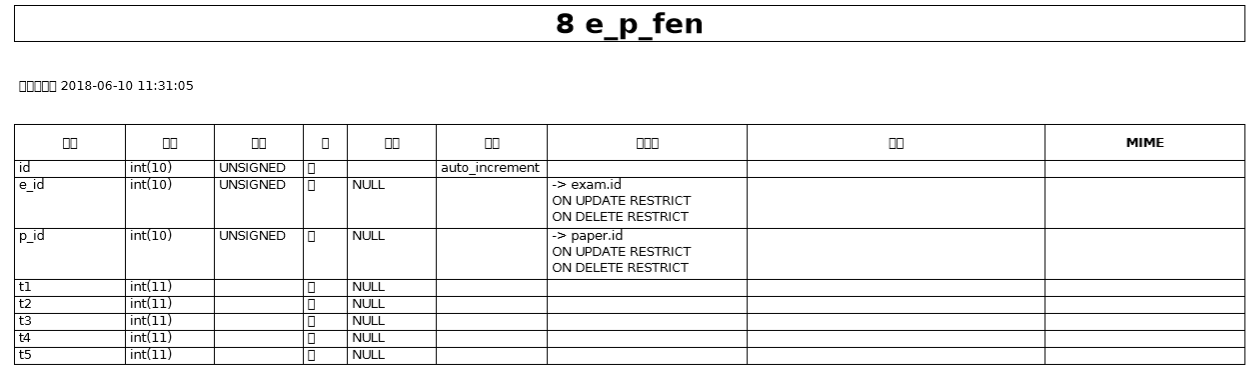
2.6.记录考试批改信息



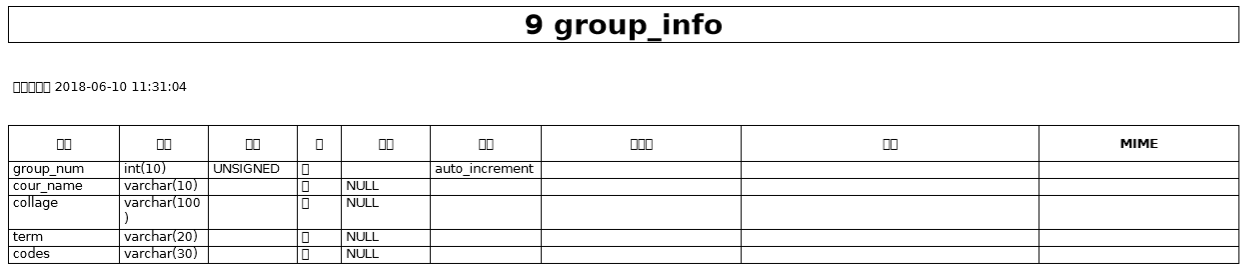
2.7.记录考试与用户之间的关系



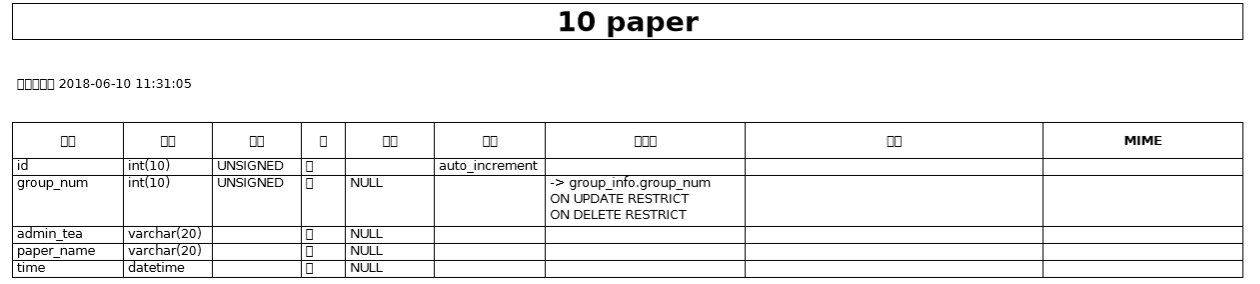
2.8.记录每张试卷在对应的考试中每种题型的分数



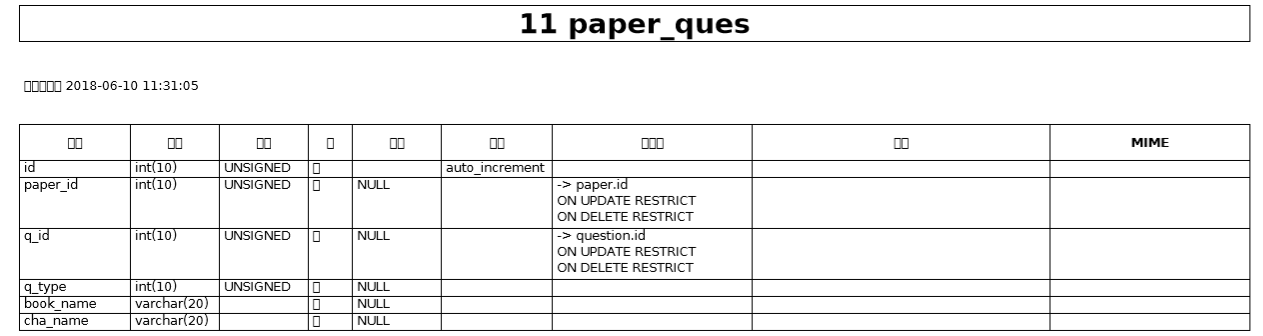
2.9.记录group信息



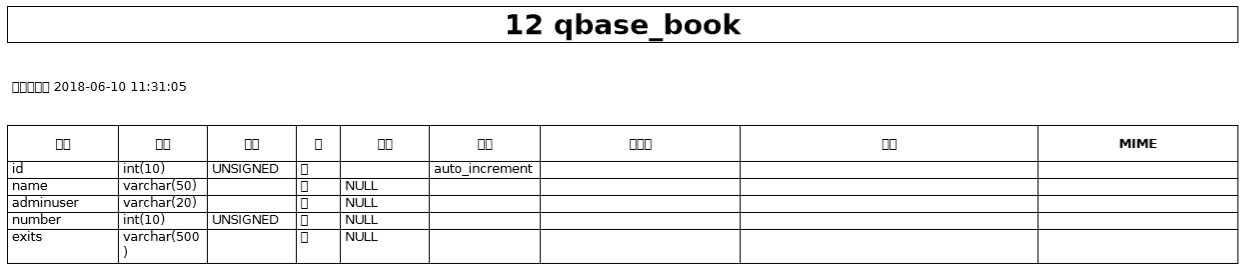
2.10.记录试卷信息



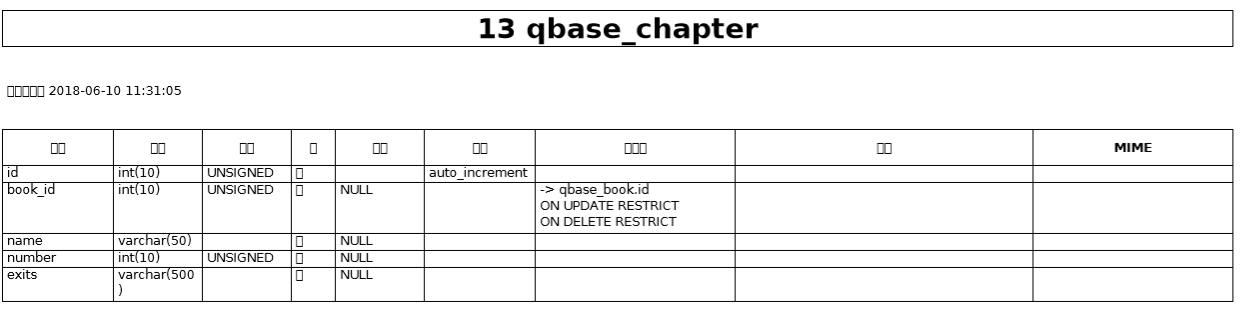
2.11.记录试卷包含题目



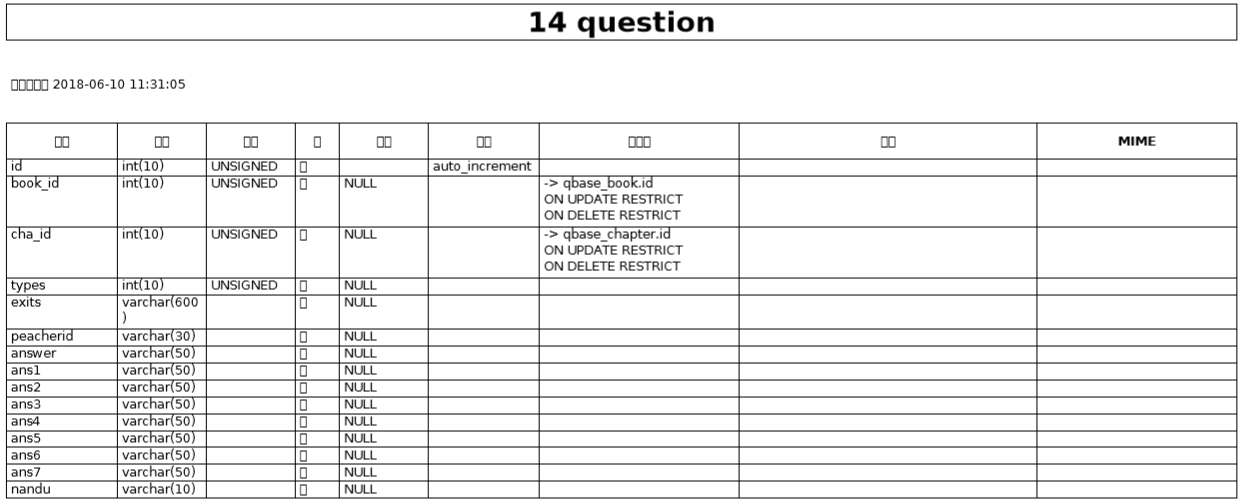
2.12.题库一级目录



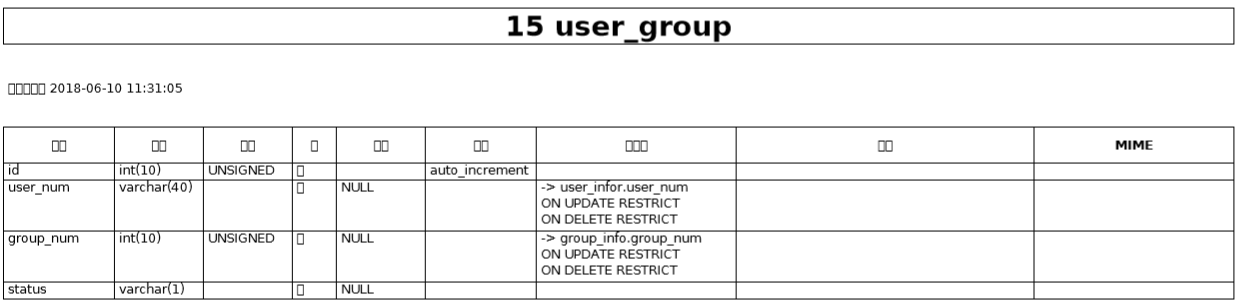
2.13.题库二级目录



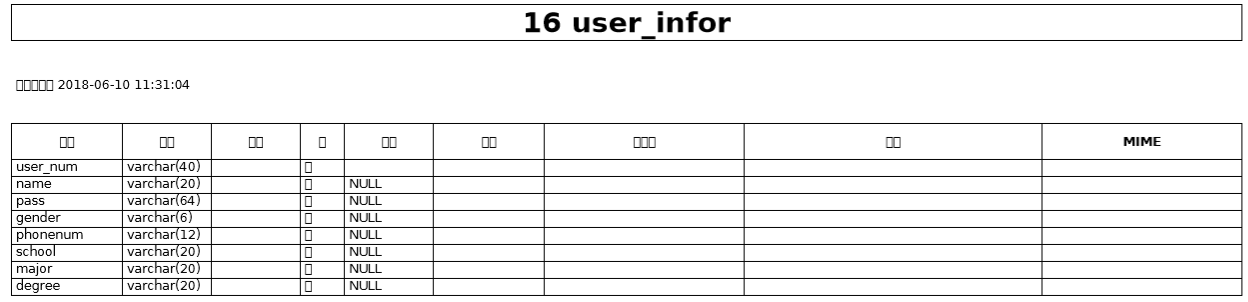
2.14.题目内容信息



2.15.记录用户与group之间的关系



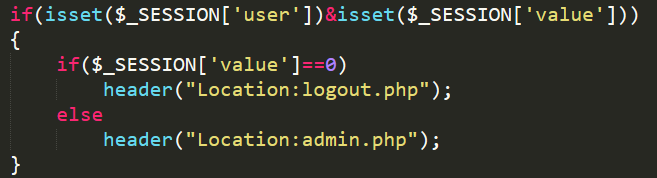
2.16.记录用户信息



1. WEB 管理端

3.1登录验证

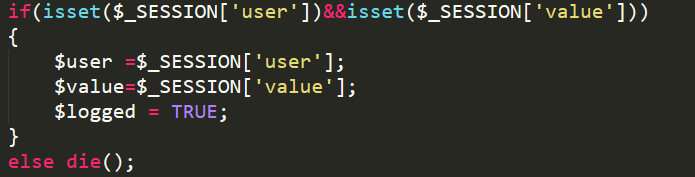
使用创建对话SESSION的方式进行登录验证，SESSION的各个变量是存在后台服务器上的，因此相较于cookie存在用户端容易被篡改而言安全性更高，但缺点在于当以SESSION方式建立大量连接时会加重服务器负担，但考虑到时面试考试因此每次同时进行考试的用户不会太多，因此可以避免SESSIOn连接的短板而发挥其安全的优势。为了增加安全性账户密码先经过哈希算法加密再存入数据库，数据库中存储的密码并非明文密码，因此即使是数据库泄露，攻击者也无法获取用户的密码。验证过程通过查询数据库中存储的账号密码，如果有匹配项则登录成功，创建SESSION，将用户名和用户权限存储在$\_SESSION数组中。



该段代码用来判断用户是否已经登录，如果已经登录则跳过登录页面



该段代码用来判断用户登录信息，如果验证通过这创建SESSION



每个页面都有的验证登录信息。

3.2账号管理

管理端账号设置为超级管理员和面试官。

超级管理员权限为1，可以管理全部事件，可以添加删除新的管理账号，可以管理题库，创建GROUP创建考试。

面试官权限为2，只可以管理自己所属的group中的事件。

账号权限储存在数据库的value项里，登陆后储存在$\_SESSION[‘value’]里。

每次进行事件操作需要进行权限判断，如果越权则直接注销会话。

添加账号，输入信息后采用SQL select查询语句先查询该账号是否存在，如果已存在则终止本次添加，如果不存在则使用insert插入语句将该账号信息添加入数据库。

修改账号，使用update修改语句修改已插入的账号信息。

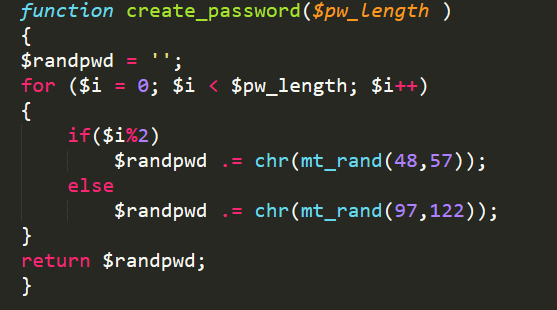
删除账号，使用delete删除语句删除对应的账号信息。

3.3Group管理

为了方便考场和面试官的分配，比如一家公司在同一时间可能进行着不同地点的招聘考试，同一组候选人可能会进行多次笔试考核，我们设置了group来方便管理。Group的信息储存在数据库group\_info里。每个group有若干候选人和若干面试官组成，考试的单位为group。不同的group之间进行的考试互不影响。一个面试官也可以属于多个group，一个group也可以有多个面试官。候选人加入group需要填写该group的邀请码，为了减少麻烦邀请码永久有效。邀请码在管理员创建group时随机生成或者由创建者自行设定，group邀请码储存在数据库group\_info的code项中。面试官在后台登录时只能看待自己所属的group。

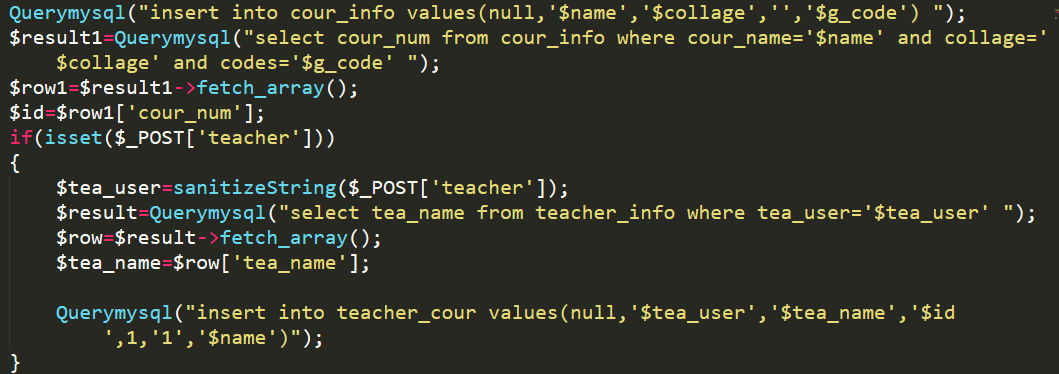
Group添加：

Group\_add.php文件处理添加group的请求



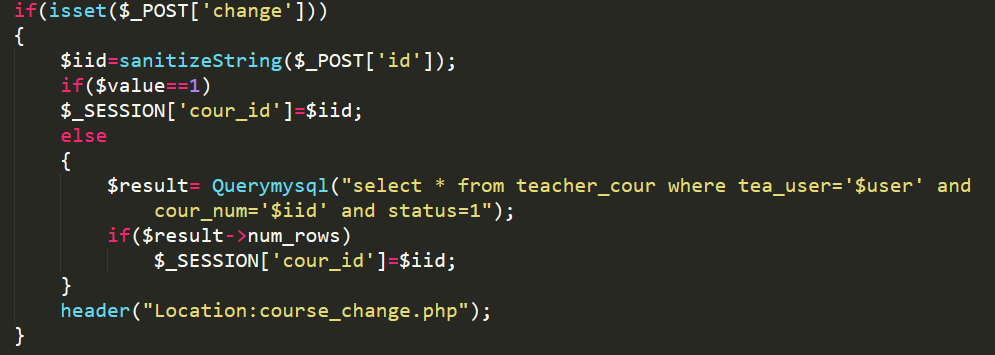


通过create\_password()函数来生成随机的邀请码



该段代码用来向数据库中添加group信息和添加group与面试官对应的信息。

安全方面，为了安全性、防止越权操作，每次在修改或者添加group信息时会先进行权限判定，具体方式为：先判定操作者的权限，如果value为1，则为超级管理员拥有最高权限，可以操作全部group。如果value为2则为面试官，在通过查询数据库中的admin\_group表单，查找该面试官是否属于该group如果属于则允许操作，如果非法越权操作则拒接本次请求并注销当前会话。



改段代码用来判断权限。

3.4题库管理

超级管理员拥有管理题库的权限，题库设有创建题库、导入试题、添加试题、修改试题的接口。

根据我校（中国人民大学）在教学平台上的实践和经验教训，我们将题库设计为三级索引。这样可以极大地方便题目分类和试卷组装。

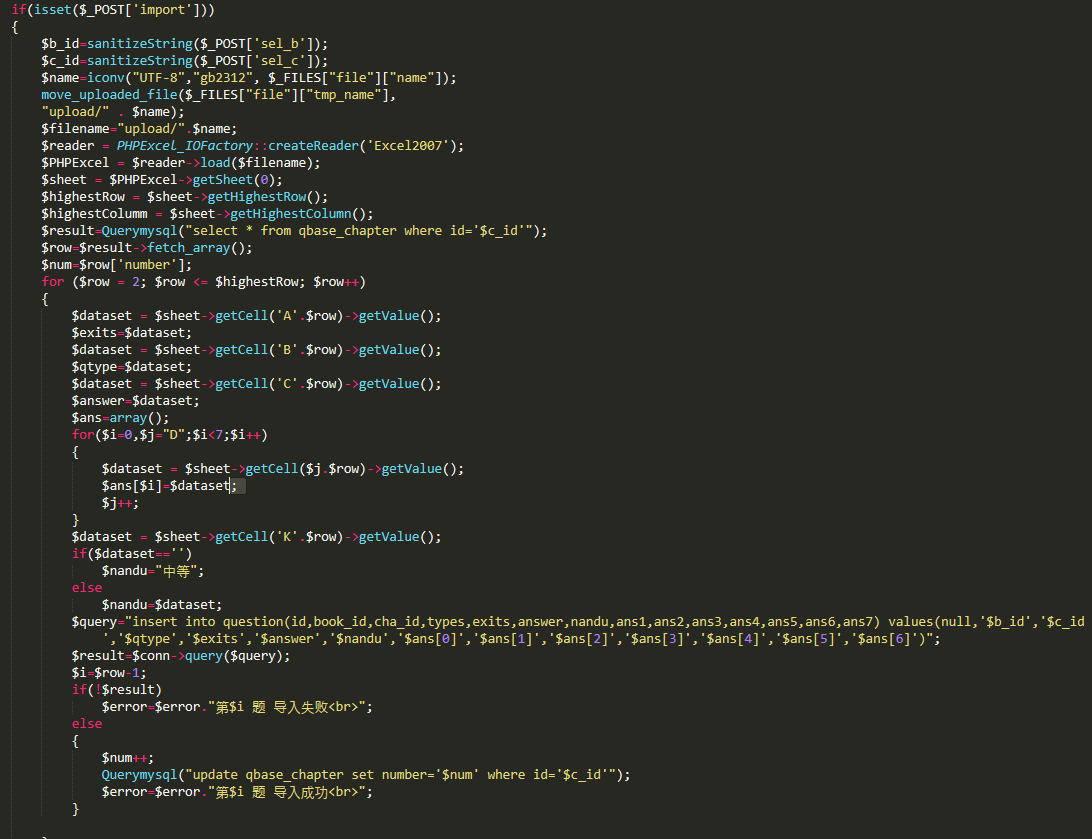
题库的第一级为题库级（相当于一本书），

第二级为题集级（相当于书的目录），

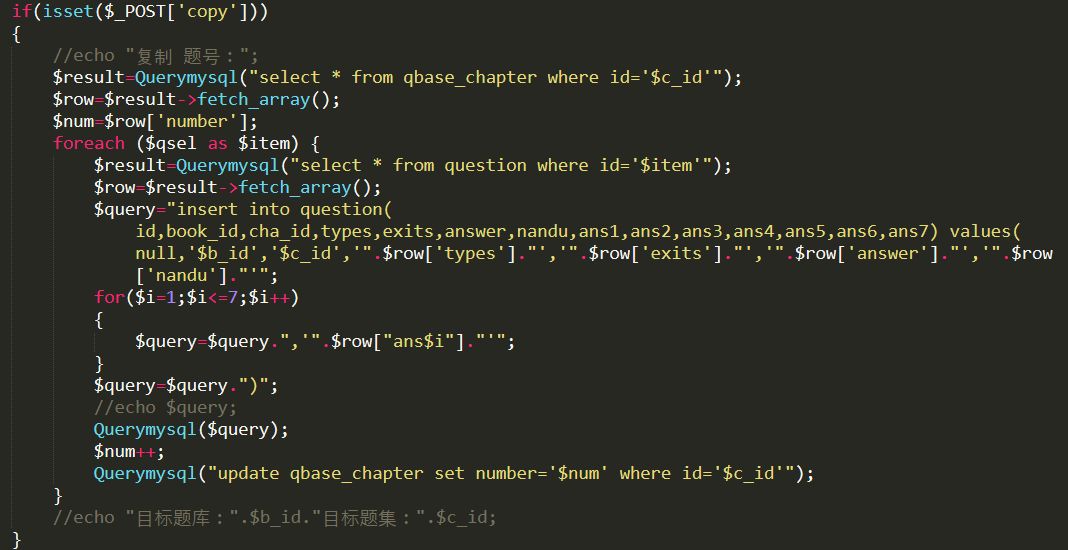
第三级根据题目的类型划分为六种题型，

试题储存在每个对应题型下。

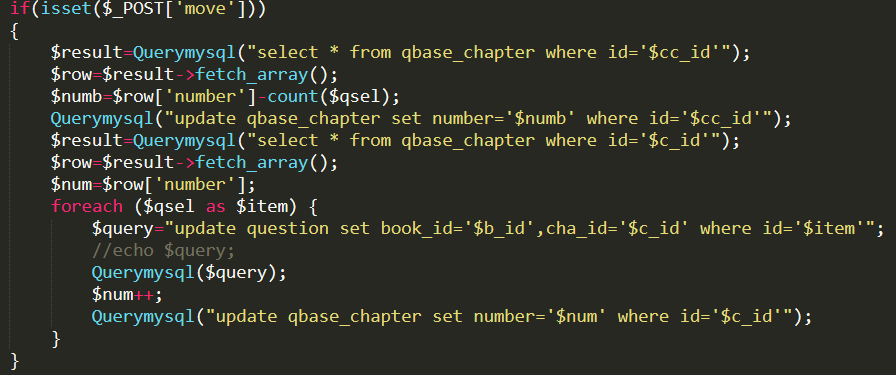
为了方便操作，题库里的题可以在不同题库之间移动，也可以对不需要的试题进行销毁。



用Excel表导入试题



复制试题到另一个题集



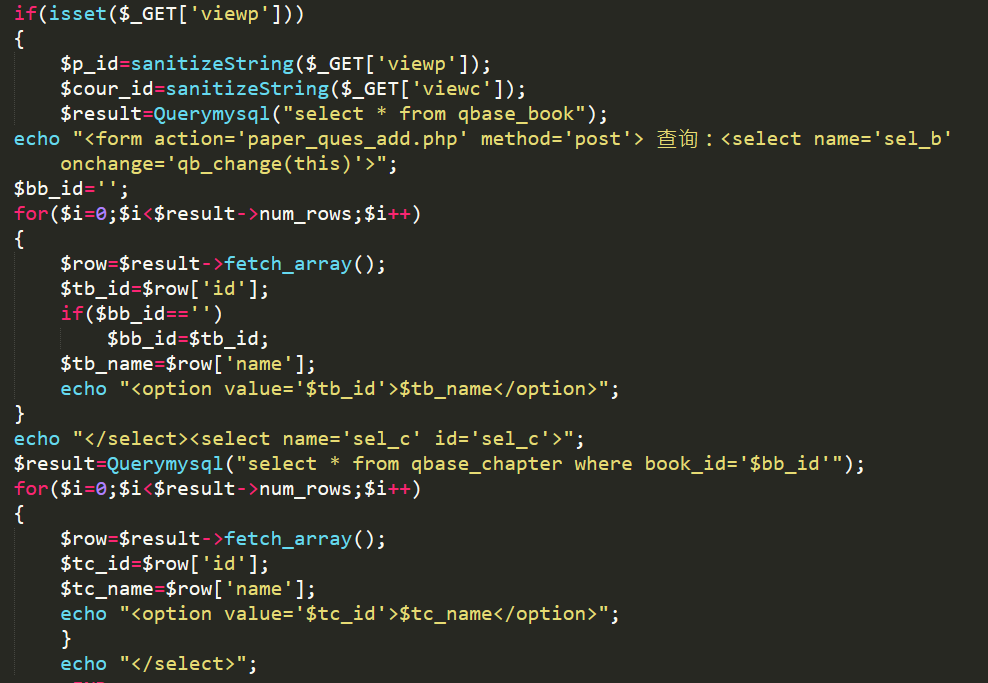
移动试题

3.5创建试卷

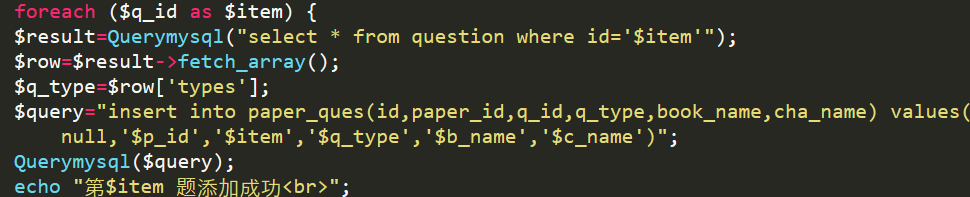
试卷可以反复利用，一套好的试卷可以供多次笔试考试使用。



添加试卷信息



检索题库题目信息

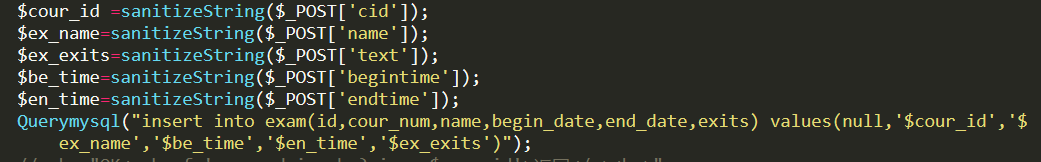


向试卷中添加试题

3.6创建考试

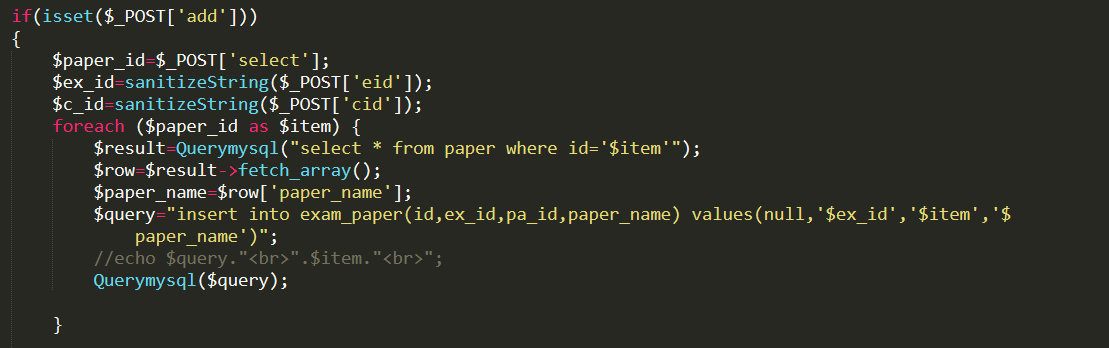
考试的所属单位为group，超级管理员和该group的面试官可以创建修改考试。创建考试时要输入考试的名称，信息，考试开始时间，考试结束时间。





添加考试

每场考试的信息储存在数据库的exam表单里面。考试创建完成后可以添加试卷，试卷可以添加多张。



添加试卷

设定该场考试每套试卷每种题型的分数。因为考虑到要划分等级，一场考试可以添加多套试卷，可以是难度系数不同的试卷，每套试卷对应的每种题型的分数可以不相同。每套试卷题型的分数储存在数据库里，修改时采用ajax异步方式及时的显示修改后每种题型的总分和整张卷子的总分。



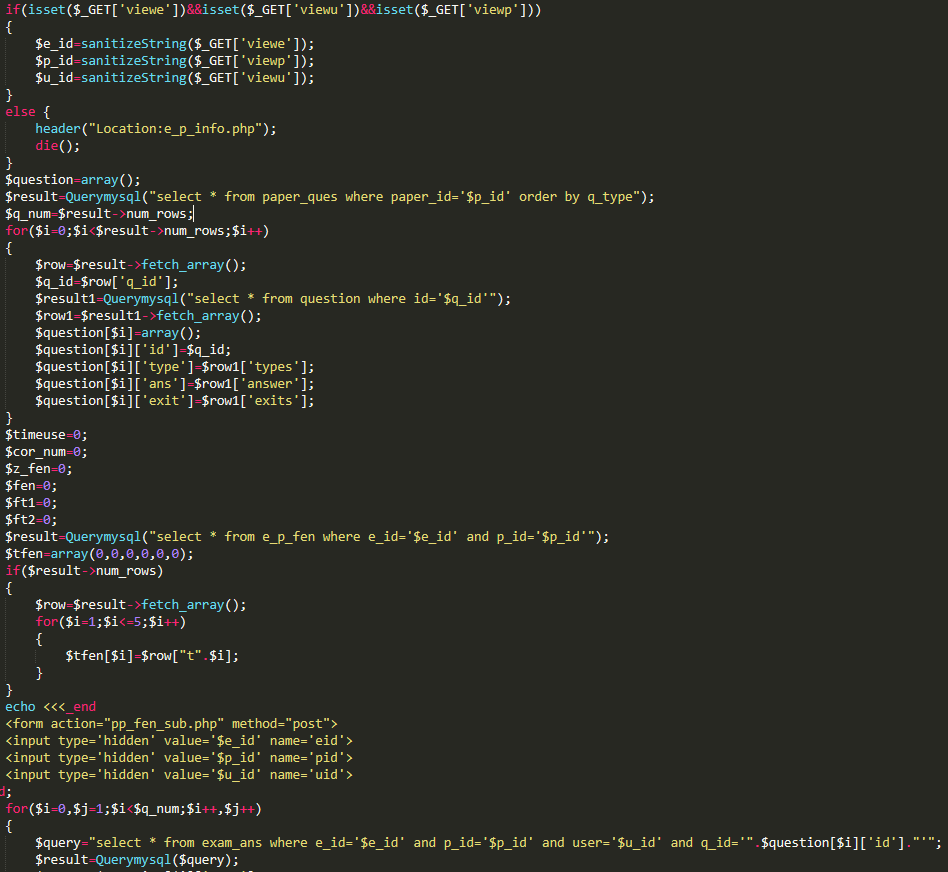
Javascript实现异步修改每题分数。

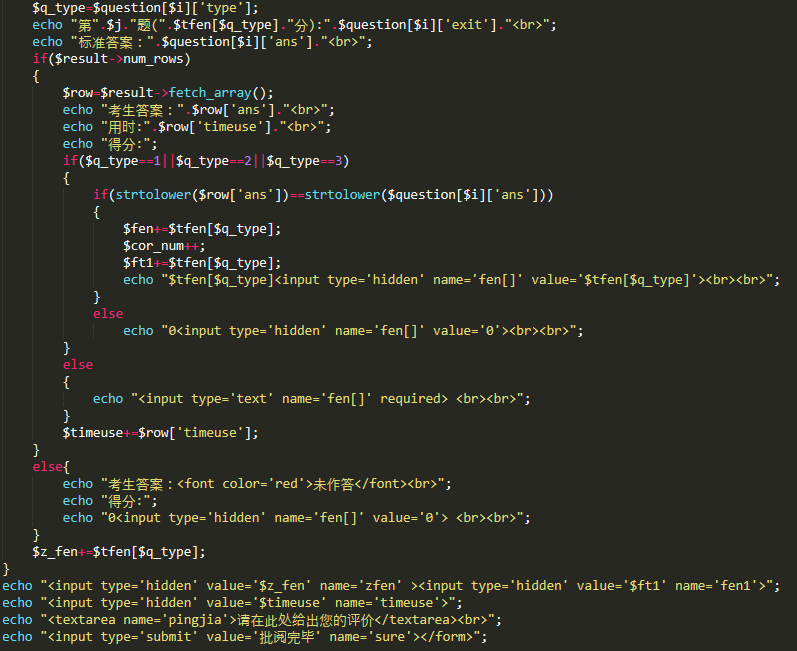
3.7批改试卷

实现客观题自动判分，主观题半自动反分，

如果整张试卷全为客观题，则系统可以自动进行判分，批改人只需填写批改意见即可。

试卷批改的结果储存在数据库exam\_read表单中。





1. 手机答题端

4.1登录验证

登录使用微信账号登录，微信账号对应每一个公众号有唯一的openID，

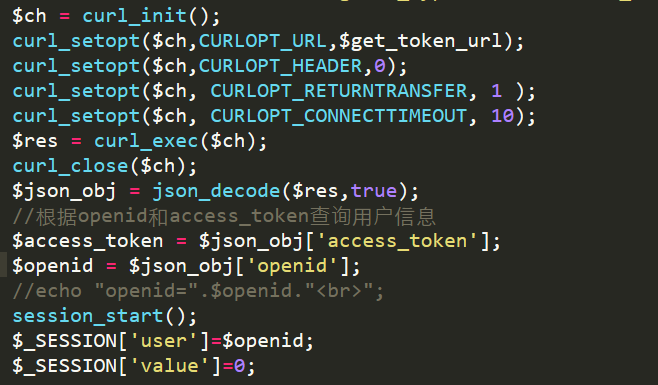
openID可以用来当做用户的账号来使用。

获取openID的方式：

公众号设置连接获得code：

https://open.weixin.qq.com/connect/oauth2/authorize?appid=这里是你的公众号的APPID&redirect\_uri=http://www.xx.com/getcode&response\_type=code&scope=snsapi\_userinfo&state=123#wechat\_redirect

用code获得openID



获得openID并建立SESSION。

如果用户是第一次登录该公众号，则把该openID存入数据库，并要求其完善个人信息。

4.2信息填写

如果个人信息没有完善将无法参加考试，因为是直接使用微信账号作为该平台的账号所以账号的真实性有一定的保证。

4.3加入group

用户与group的关系储存在数据库user\_group表单中。

若用户申请加入某group用javascript验证申请：



Ajax异步验证。

4.4参加考试

参加考试首先要在该考试所属的group中。

一场考试可能有一张或者多张试卷，按照考试要求进行答题。

试题显示：

用ajax异步随时上传每次答题的结果，避免考生忘记提交，

用来处理异步响应。

Show\_q.php位于后台用来处理每次上传的答题情况和输出下一道题。

1. 候选人信息统计分析

为了方便面试官更加了解候选人，除了在阅卷之后可以统计本次考试每个候选人的得分和耗时，还可以拆分来看，分析本次考试候选人在每个试题范围的正确率，用以辅助分析候选人在对应范围的能力情况。在查看考生信息的同时还可以看到该考生历史考试情况和其他面试官对他的评价。

1. 题库自动演化

题库里的题有初始的难度设定值，每次考试会对对应的试题进行修正，

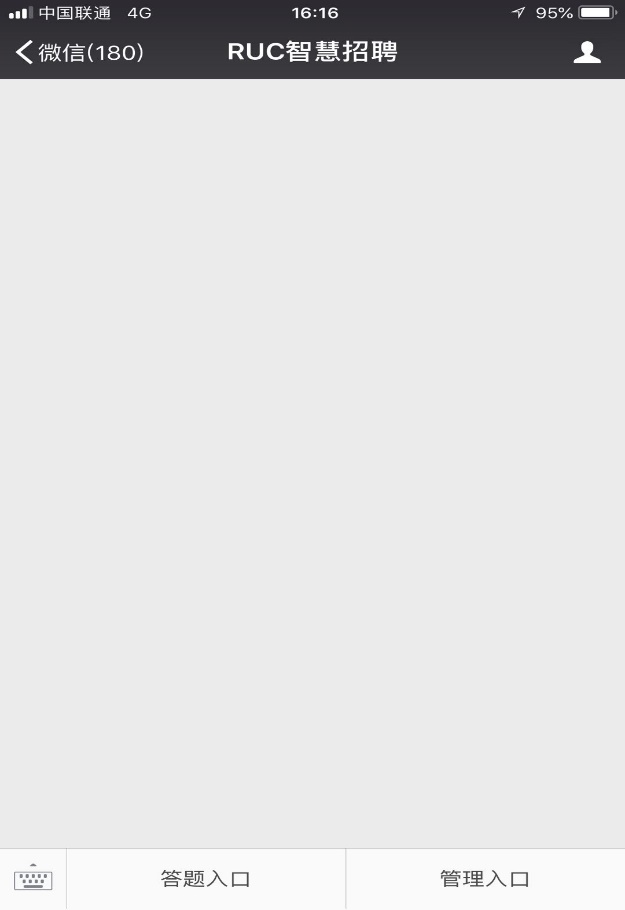
将一场考试全部考生对某道题的答题情况进行统计，统计本题的答题正确率和平均耗时。每道题的难度分数为：10\*（1-（正确率\*）最小为0

每次对试题难度的修正公式为：原难度\*0.95+本次难度分数\*0.05；

试题难度的设定可以更加方便面试官出一套更加合适的试卷。

1. 设计成果

1.微信公众号界面见图1



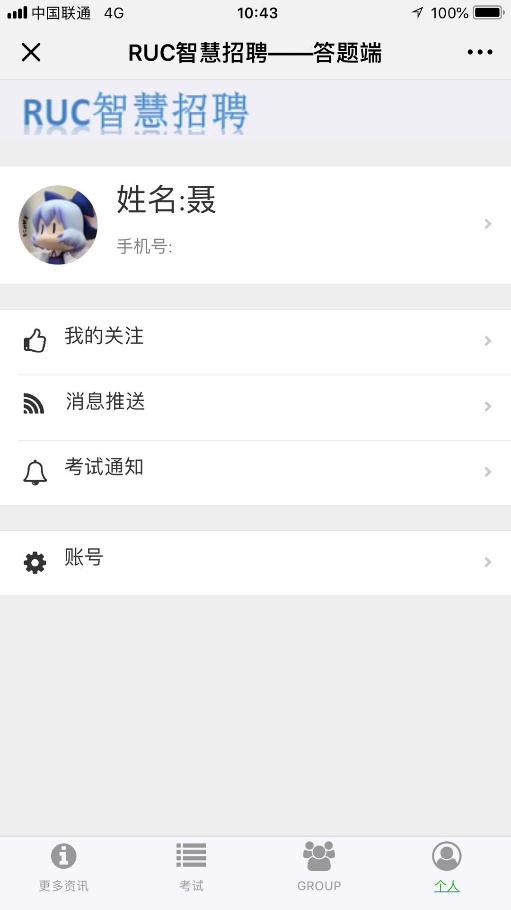
**图1 微信公众号界面**

2.点击答题入口，进入RUC智慧招聘——答题端，图2为答题端考试信息界面



**图2 答题端考试信息界面**

3.图3为答题端个人信息页面



**图3 答题端个人信息页面**

4.图4为答题端GROUP信息界面



**图4 答题端GROUP信息界面**

5.在图2的考试列表中点击你想参加的考试，即可参加该场考试，每场考试都属于不同GROUP，只有当你属于某一GROUP时，才能参加该GROUP下的考试，可通过输入GROUP邀请码加入GROUP，GROUP邀请码在创建GROUP时生成，图5为参加考试时提示输入GROUP邀请码加入GROUP的界面



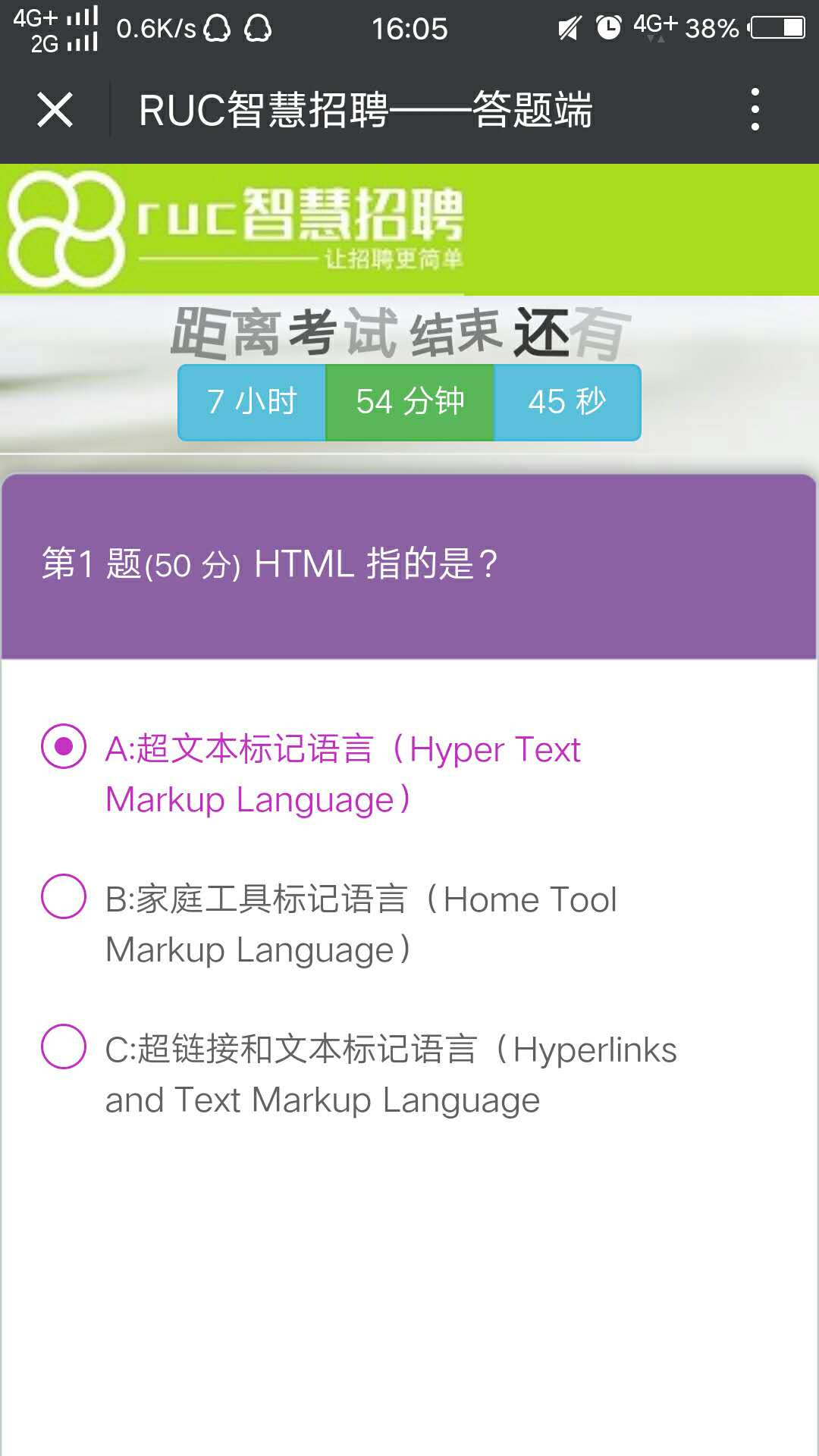
**图5 答题端加入GROUP界面**

6.图6为答题端成功参加考试后界面



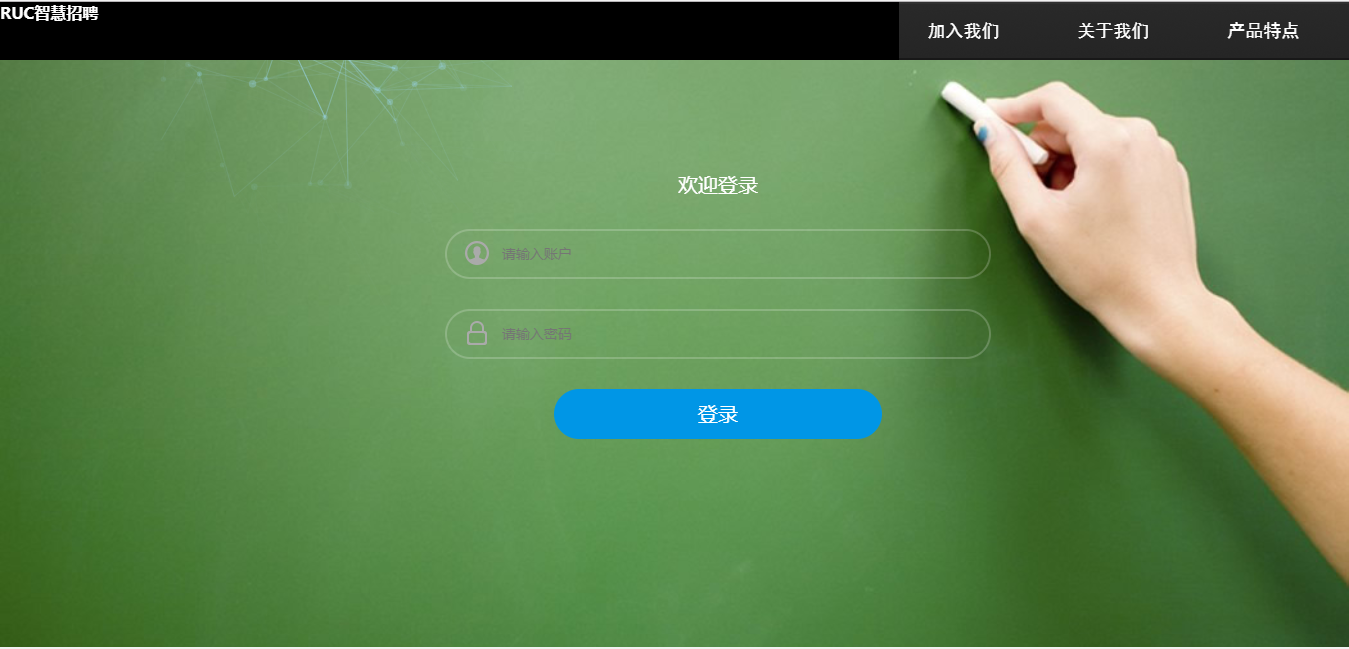
**图6 答题端参加考试界面**

7.点击图6中的开始答题，即可开始作答，图7为答题端考试中界面

****

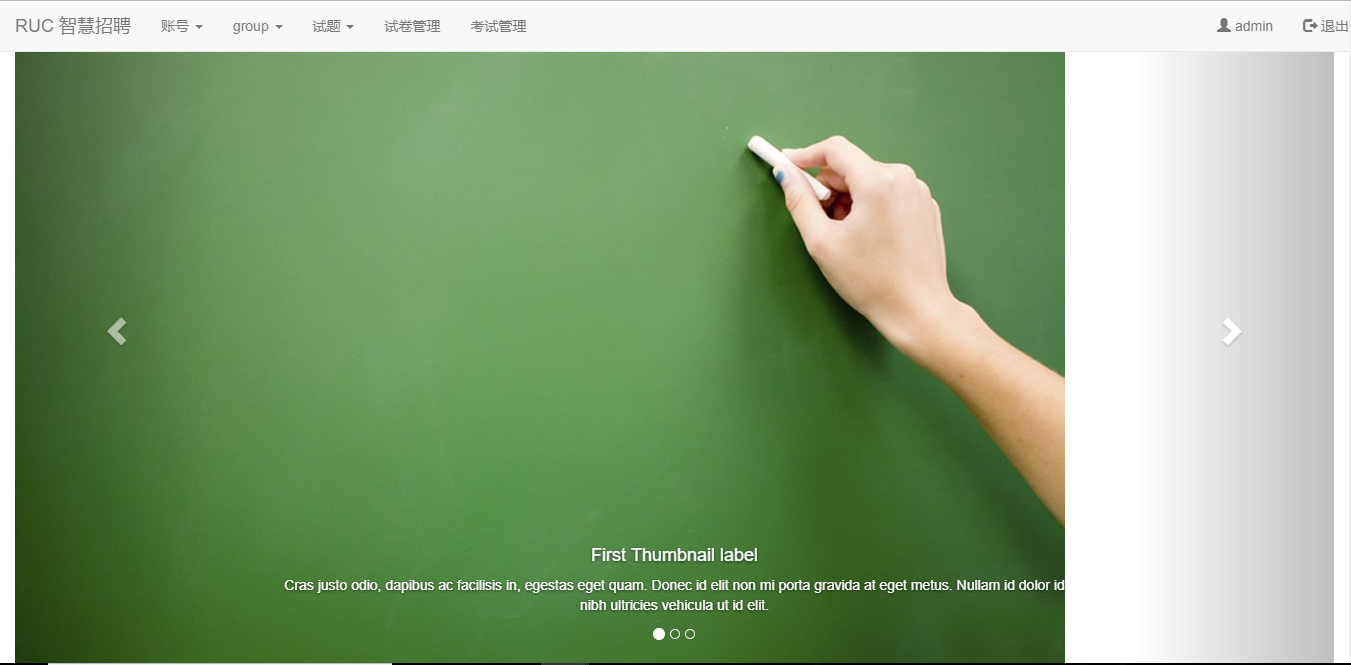
**图7 答题端考试中界面**

8.图8为RUC智慧招聘系统管理端登陆界面



**图8 管理端登陆界面**

9.在图8登陆界面中输入账号和密码，即可进入RUC智慧招聘系统管理端，图9为登陆后的主页面



**图9 管理端主页面**

10.点击图9中右上角admin，即可进入图10所示的个人信息界面



**图10 个人信息界面**

11.点击图9账号下的账号管理，进入图11所示的账号管理页面



**图11 管理端账号管理界面**

12.点击图11中的修改按钮，即可对该账号的信息进行修改，图12为修改账号信息界面



**图12 管理端修改账号信息界面**

13. 点击图9账号下的添加管理账号，进入图13所示的添加管理账号界面



**图13 管理端添加管理账号界面**

14.点击图9中group下的添加group，进入图14所示的添加group界面



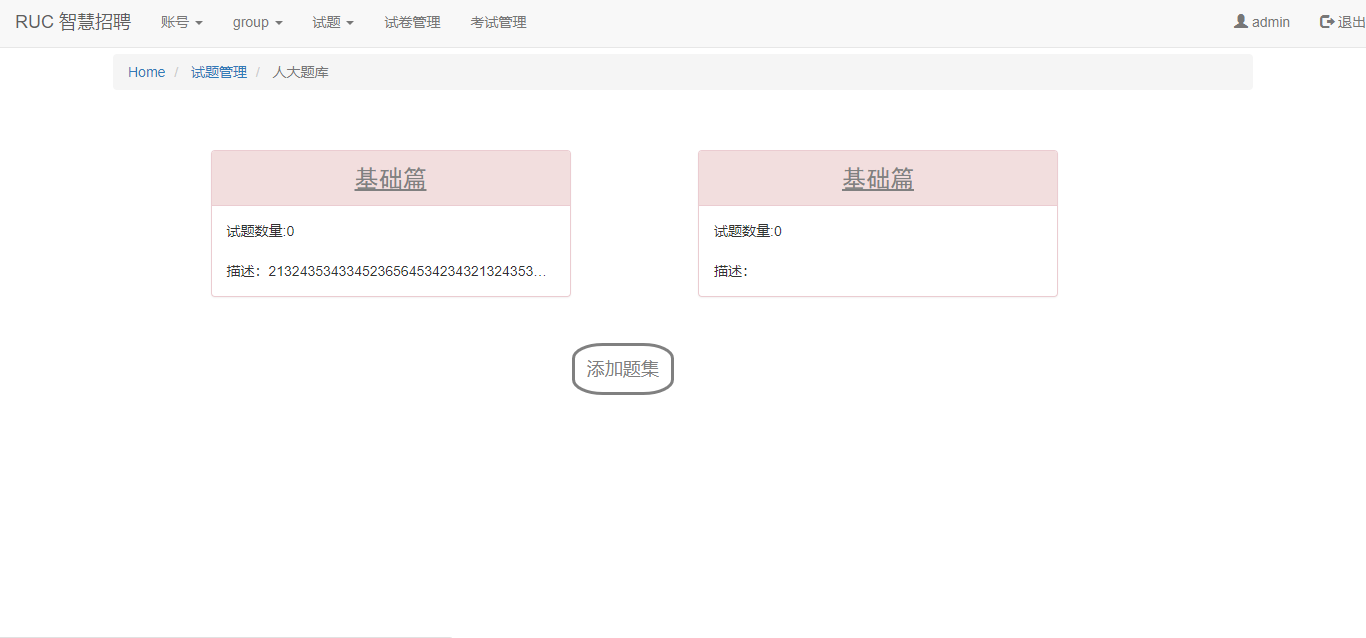
**图14 管理端添加group界面**

15.点击图9中试题下的试题管理，进入图15所示的试题管理界面



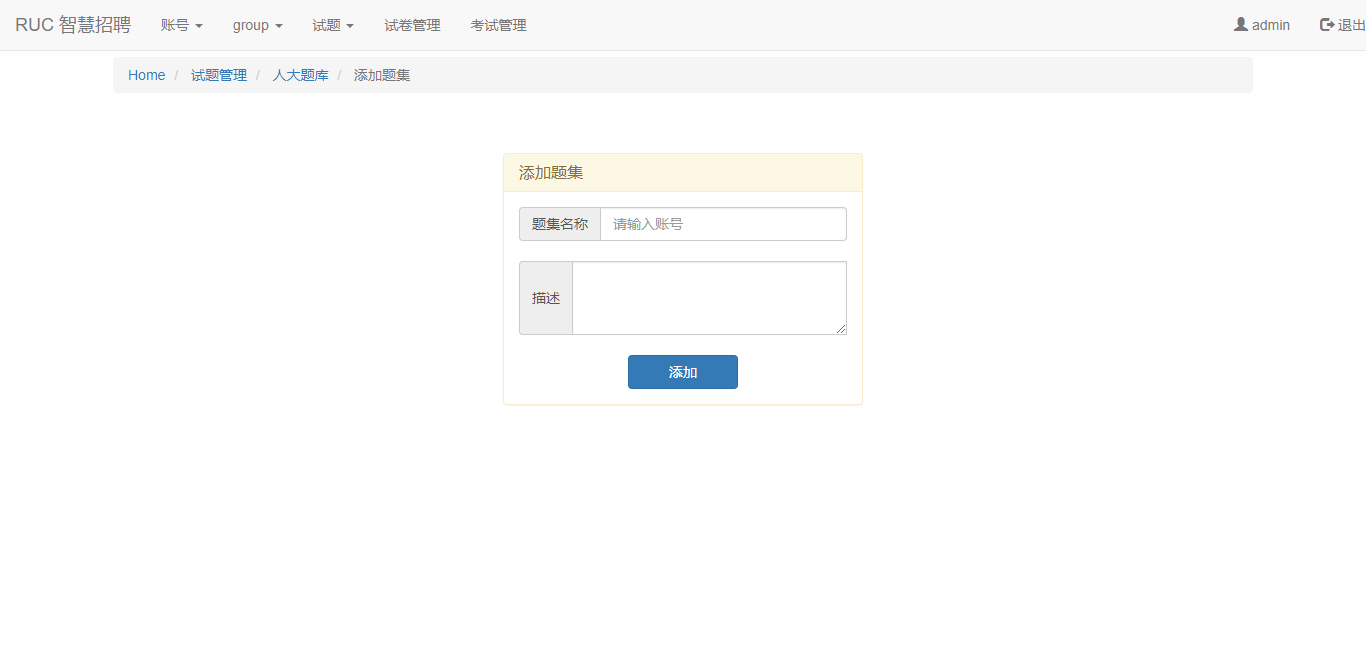
**图15 管理端试题管理界面**

16.点击图15中的某一题库，即可查看题库信息，图16为管理端的题库信息界面



**图16 管理端题库信息界面**

17.点击图16中的添加题集，进入图17所示的添加题集界面



**图17 管理端添加题集界面**

18.点击图9中试题下的添加题库，进入图18所示的添加题库界面



**图18 管理端添加题库界面**

19．点击图9中试题下的导入试题，进入图19所示的导入试题界面



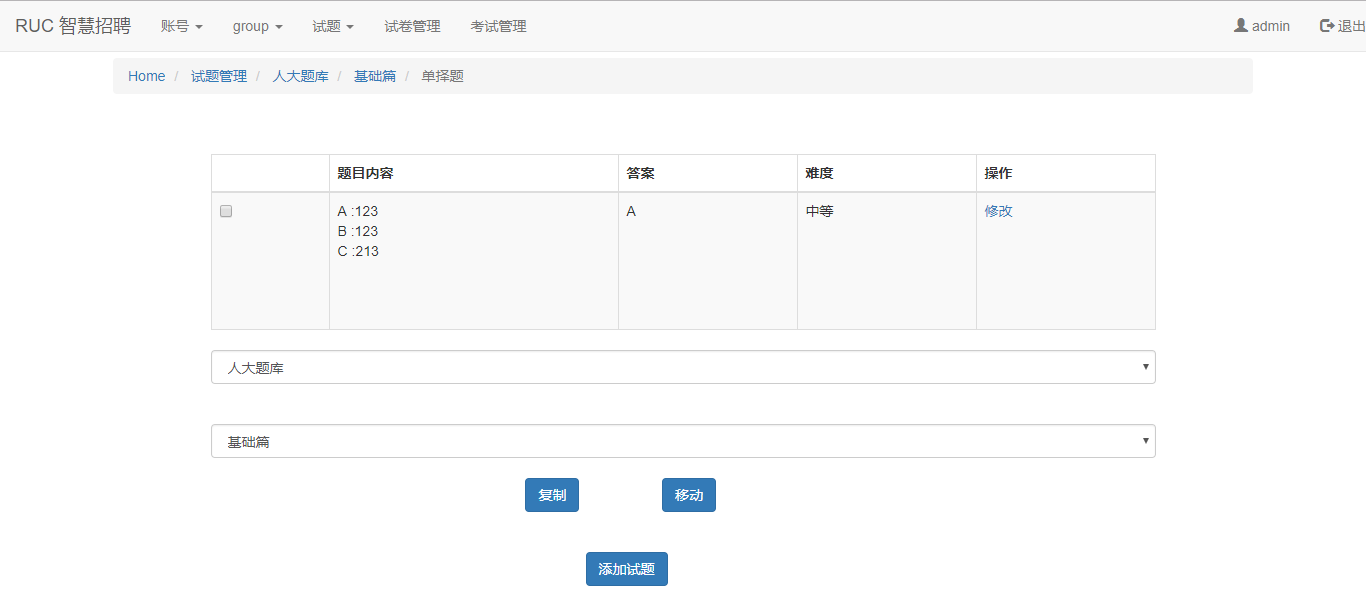
**图19 管理端导入试题界面**

20.点击图16中的题库名称，进入图20所示的题库信息界面



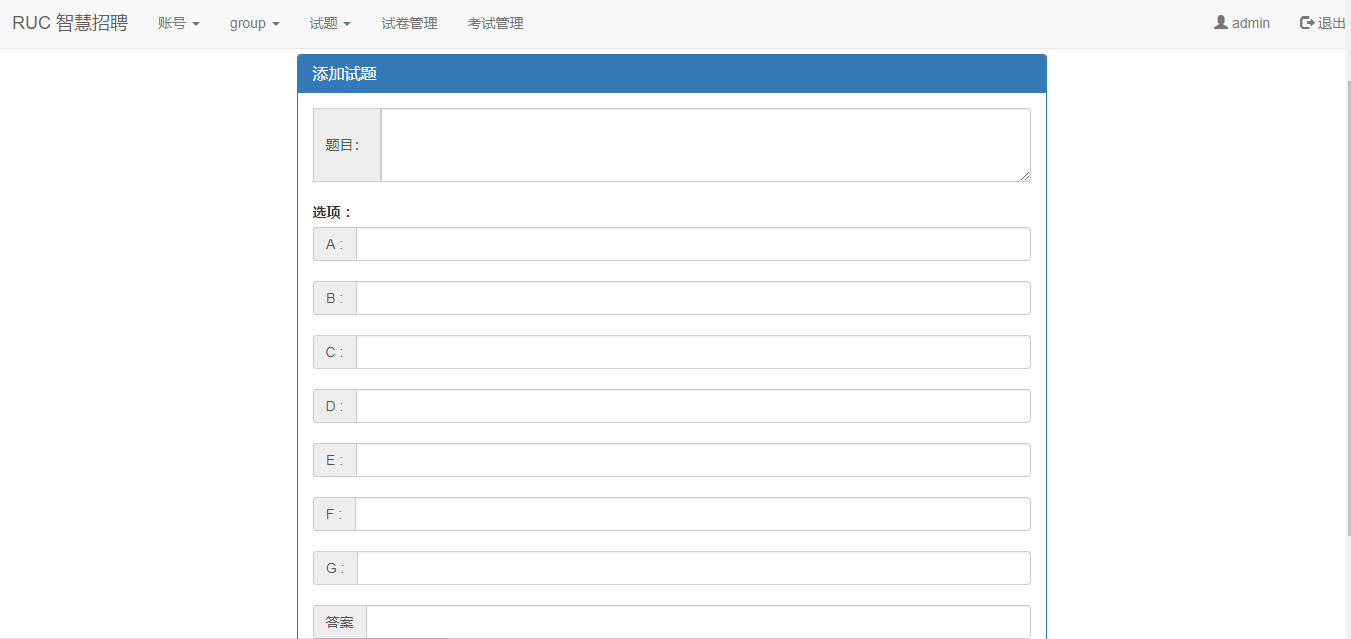
**图20 管理端题库信息界面**

21.点击图20中的单择题，可查看图21所示的该题库下单择题信息界面



**图21 管理端题库信息-单择题界面**

22.点击图21中的添加试题，进入图22所示的添加选择题界面





**图22 管理端添加选择题界面**

23.点击图9中的试卷管理，进入图23所示的试卷管理界面



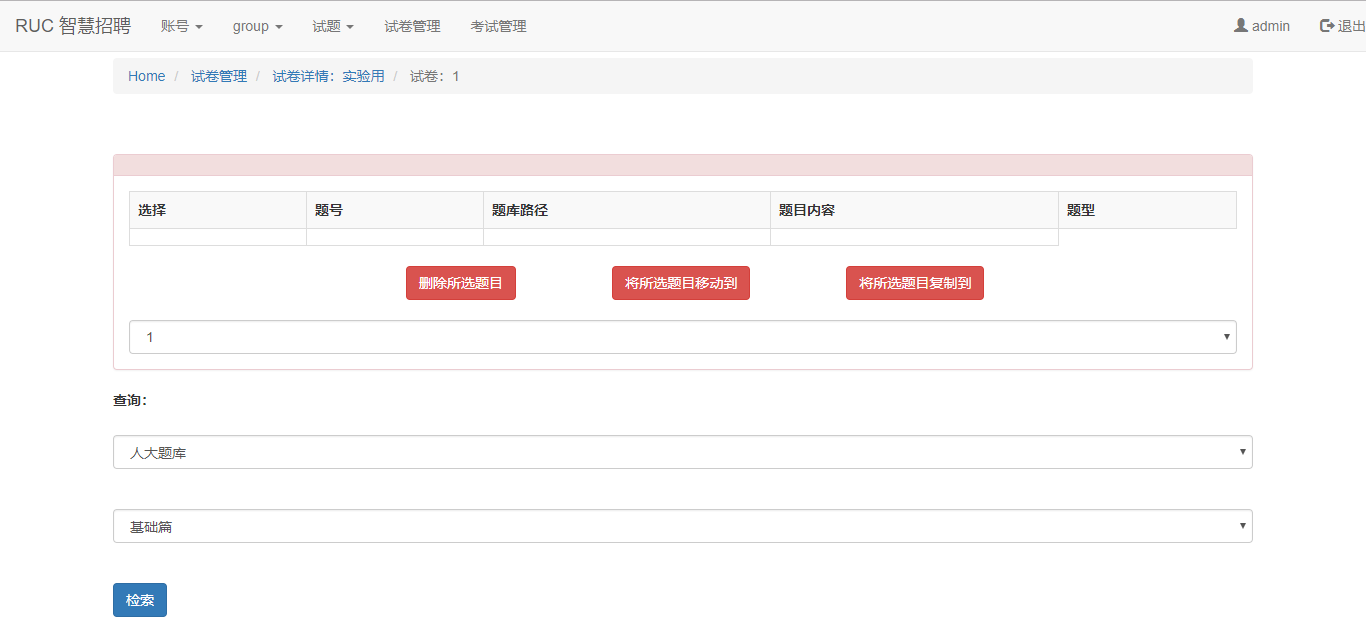
**图23 管理端试卷管理界面**

24.点击图23中的进入试卷详情，进入图24所示的试卷详情界面



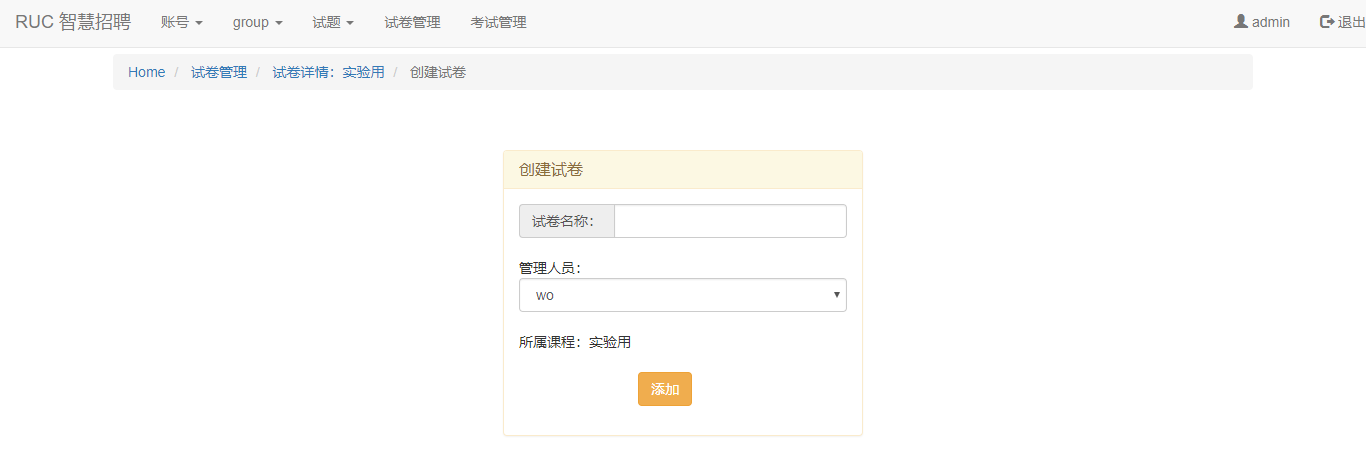
**图24 管理端试卷详情界面**

25.点击图24中的进入试卷，进入图25所示的试卷信息界面



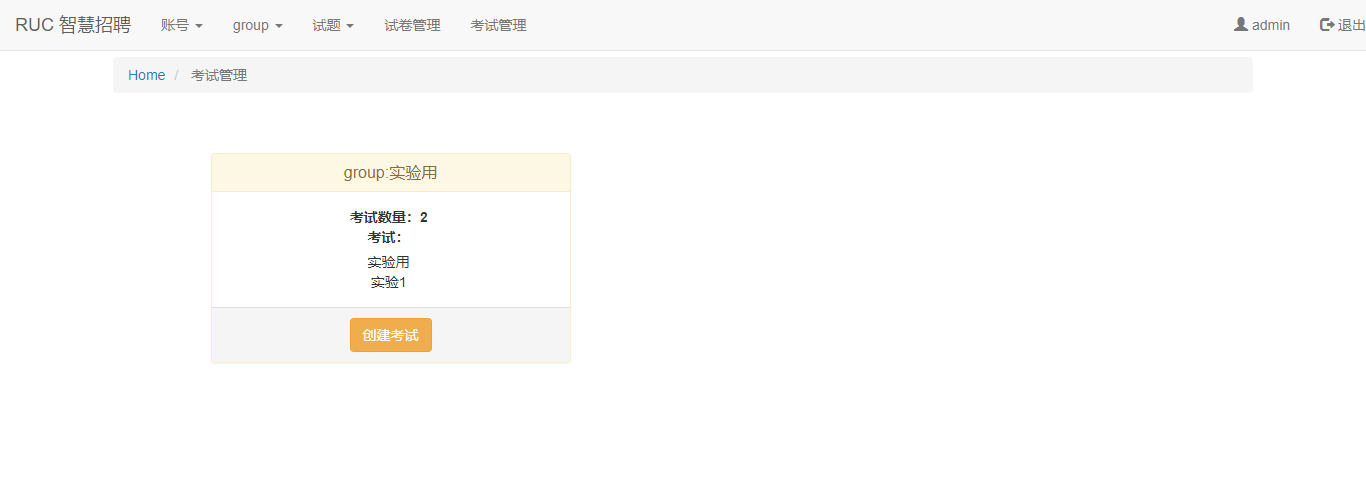
**图25 管理端试卷信息界面**

26.点击图24中的创建试卷，进入图26所示的创建试卷界面



**图26 管理端创建试卷界面**

27.点击图9中的考试管理，进入图27所示的考试管理界面



**图27 管理端考试管理界面**

1. 安装说明
2. 服务器配置

1.1服务器使用：Ubuntu

1.2安装Apache2

**使用开发包的打包服务——apt-get**

**安装apache，在命令行终端中输入一下命令：**

**$ sudo apt-get install apache2**

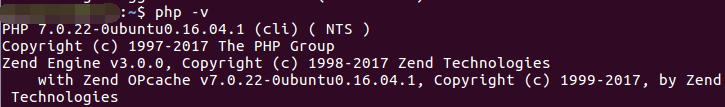
**安装完成后，重启apache服务，在命令行终端中输入一下命令：**

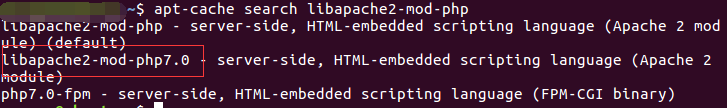
**$ sudo /etc/init.d/apache2 restart**

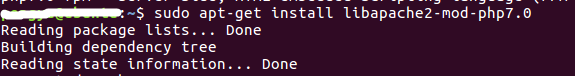
1.3安装php7.0

**这里我准备安装PHP7.0版本的，在命令行输入:sudo apt-get install php7.0**

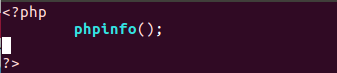
**安装完成后输入：php -v 查看PHP是否安装成功**

****

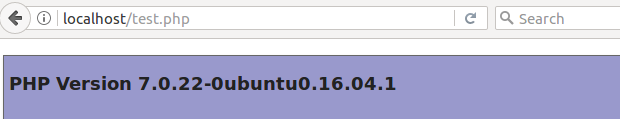
**PHP和Apache都安装好后就需要让Apache能够识别解析PHP文件，我们先搜一下有没有适合PHP7的插件，输入命令：apt-cache search libapache2-mod-php**

**可以看到搜出来的结果里面有一个是PHP7.0版本的，我们就安装这个：sudo apt-get install libapache2-mod-php7.0**

**下面我们就可以随便写一个php文件看是否可以解析访问。输入命令: cd /var/www/html 切换到apache项目目录下，新建文件：sudo vim test.php**

****

**保存后浏览器访问：localhost/test.php**

****

**如果一切正常的话，就会看到php的一些信息。**

1.4配置php session

1. 公众号配置

1．首先申请公众号

2.在自定义菜单上设置接口

1. 数据库搭建

### 步骤1 – 安装MySQL

**在 Ubuntu 16.04 中，默认情况下，只有最新版本的 MySQL 包含在 APT 软件包存储库中。**

**sudo apt-get update**

**sudo apt-get install mysql-server**

**系统将提示您在安装过程中创建 root 密码。选择一个安全的密码，并确保你记住它，因为你以后需要它。接下来，我们将完成 MySQL 的配置。**

### 步骤2 – 配置MySQL

**因为是全新安装，您需要运行附带的安全脚本。这会更改一些不太安全的默认选项，例如远程 root 登录和示例用户。在旧版本的 MySQL 上，您需要手动初始化数据目录，但 Mysql 5.7 已经自动完成了。**

**运行安全脚本。**

**sudo mysql\_secure\_installation**

**这将提示您输入您在步骤1中创建的 root 密码。您可以按 Y，然后 ENTER 接受所有后续问题的默认值，但是要询问您是否要更改 root 密码。您只需在步骤 1 中进行设置即可，因此无需现在更改。**

**最后，我们来测试MySQL安装。**

### 步骤3 – 测试MySQL

**按上边方式安装完成后，MySQL应该已经开始自动运行了。要测试它，请检查其状态。**

**systemctl status mysql.service**

**您将看到类似于以下内容的输出：**

**mysql.service - MySQL Community Server**

**Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: en Active: active (running) since Wed 2016-11-23 21:21:25 UTC; 30min ago Main PID: 3754 (mysqld) Tasks: 28 Memory: 142.3M CPU: 1.994s CGroup: /system.slice/mysql.service └─3754 /usr/sbin/mysqld**

**如果MySQL没有运行，您可以启动它：**

**sudo systemctl mysql start**

**如果额外的检查，您可以尝试使用该 mysqladmin 工具连接到数据库，该工具是允许您运行管理命令的客户端。例如，该命令表示以 root（-u root）方式连接到 MySQL ，提示输入密码（-p）并返回版本。**

**mysqladmin -p -u root version**

**你应该看到类似的输出：**

**mysqladmin Ver 8.42 Distrib 5.7.16, for Linux on x86\_64**

**Copyright (c) 2000, 2016,** [**Oracle**](https://www.linuxidc.com/topicnews.aspx?tid=12) **and/or its affiliates. All rights reserved.**

**Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.**

**Server version 5.7.16-0ubuntu0.16.04.1 Protocol version 10 Connection Localhost via UNIX socket UNIX socket /var/run/mysqld/mysqld.sock Uptime: 30 min 54 sec**

**Threads: 1 Questions: 12 Slow queries: 0 Opens: 115 Flush tables: 1 Open tables: 34 Queries per second avg: 0.006**

**这意味着MySQL正在运行**

**Mysql -u root -p**

**登录mysql**

**Create database studing；**

**alter database studing character set utf8mb4 collate utf8mb4\_general\_ci;设定数据库编码模式；**

**运行setup.php**