



山东大学

数据库课程设计报告

ACM 在线评测系统

21 组

黄瑞哲 杨家齐 刘丁泚

目录

前言	3
一、系统开发平台	3
1.1 开发语言	3
1.2 开发工具	3
1.3 数据库	3
1.4 操作系统	3
二、数据库规划	4
2.1 任务陈述	4
2.2 系统层次设计	4
2.3 任务目标	6
三、用户视图	7
3.1.游客（未注册用户， GUEST 层）	7
3.2.未加入该域的登录用户	7
3.3.域创建者	7
四、需求分析	8
4.1 用户需求分析	8
4.1.1 数据需求	8
4.1.2 事务需求	9
4.2 系统需求	9
4.2.1 网络需求	10
4.2.2 性能	10
4.2.3 安全性	10
4.2.4 备份和恢复	10
五、逻辑设计	11
5.1 ER 图	11
5.2 数据字典	12
5.2.1 题目信息	12
5.2.2 赛事/训练信息	12
5.2.3 角色信息	12
5.2.4 用户信息	13
5.2.5 提交记录	13
5.2.6 站内信信息	13
5.2.7 域用户信息	14
六、物理设计	14
6.1 权限的实现	14
6.2 主要事务分析	14
6.2.1 插入数据	14
6.2.2 查询数据	15

6.2.3 更新数据	15
6.2.4 删除数据	15
6.3 GRAVATAR 的配置	15
七、界面设计.....	16
7.1 网站首页	16
7.2 登录界面	17
7.3 查看题目界面	17
7.4 提交界面	18
7.5 比赛界面	19
7.6 比赛成绩表查看	20
7.7 训练计划查看界面	21
7.8 评测队列查看界面	22
7.9 查看一条评测结果	22
7.10 用户界面	23
7.11 域界面	25
八、测试与应用.....	29
九、总结.....	30
9.1 系统优点	30
9.2 系统不足	30
9.3 经验与收获	31
十、人员分工.....	31

前言

计算机人才需求量正逐年增大，计算机领域的人才教育也显得愈发重要。而算法领域的学习主要依靠自学，社会上很多自主开发的 **Online Judge** 已经逐渐兴起，但是那些 **OJ** 大都是零散的题目分布，不够模块化。无法组队或者校内集体训练使用。为满足题目量大，便于分类，便于筛选等需求，**ACM** 实验室训练系统应运而生。

Online Judge (OJ) 在线判题系统，用户可以在线提交程序源代码，系统对源代码进行编译和执行，并通过预先设计的测试数据来检验源代码的正确性。

一、系统开发平台

1.1 开发语言

Python 3.6.6

系统全程使用 **Python** 语言进行编写。**Python** 拥有大量针对 **HTML** 及服务器的模板库。代码简单、可扩展性强、构造的服务器性能强，能够并发相应请求。

1.2 开发工具

Navicat Premium、Vim、Pycharm

1.3 数据库

MongoDB

MongoDB 是一个面向集合的基于分布式文件系统的非关系型数据库，内部数据面向集合存储，一个集合类似于关系型数据库中的表格，但由于模式自由即不用按照关系来存储，因此存储占用空间小，且容易扩展。不同于 **MySQL** 在处理大量数据时效率会变慢，**MongoDB** 的读写速度很快。

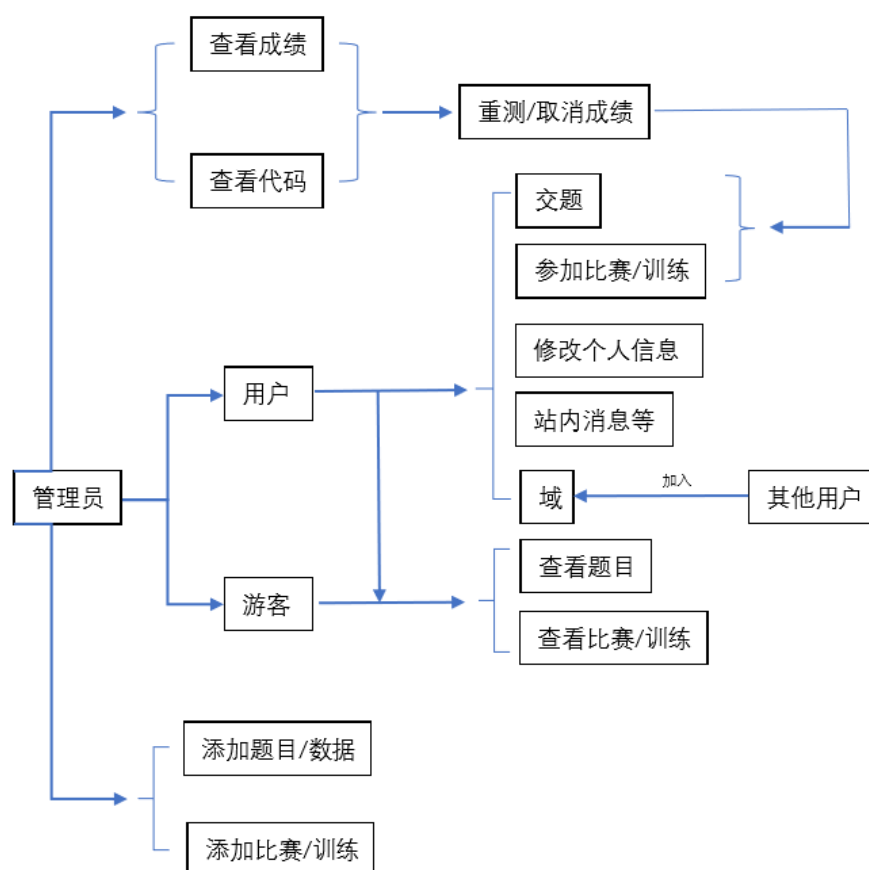
1.4 操作系统

Windows 10 Home、Linux (Ubuntu 16.04)

二、数据库规划

2.1 任务陈述

事务流程如下图：



2.2 系统层次设计

系统采用 domain 域的概念，任何人都可以创建一个域，域可以被看成一个队伍，一个学校等，每个域都与其它域相互独立，互不影响。默认进入 system 域。在自己创建的域中用户默认分为以下几个层级，且这几个层级都无法被修改：

root 层：创建域的人永远拥有不可修改的 root 层，其他用户可被 root 用户更改为 root 层，root 层拥有域内所有权限，包括：

一般	
查看此域	✓
修改角色权限	✓
展示 MOD 徽章	✓
修改此域描述	✓

题库	
创建题目	✓
修改题目	✓
修改自己的题目	✓
查看题目	✓
查看隐藏的题目	✓
递交题目	✓
读取题目数据	✓
读取自己题目的数据	✓

记录	
读取记录的代码	✓
重测题目	✓
重测记录	✓

比赛	
查看比赛	✓
查看比赛成绩表	✓
查看隐藏的比赛递交状态和成绩表	✓
创建比赛	✓
参加比赛	✓
修改任意的比赛	✓
修改自己的比赛	✓
训练	
查看训练计划	✓
创建训练计划	✓
修改训练计划	✓
修改自己的训练计划	✓

default 层：用于已经登录却未加入该域的用户，该层拥有的权限为：

- 1) 查看此域
- 2) 查看题目
- 3) 递交题目
- 4) 查看比赛
- 5) 参加比赛
- 6) 查看比赛成绩表
- 7) 查看训练计划
- 8) 参加训练

guest 层：用于未登录的用户

- 1) 查看此域
- 2) 查看题目
- 3) 查看比赛
- 4) 查看训练

上述三层除 **root** 权限不能修改外，其他所有层级权限都可以任意组合。

2.3 任务目标

- 1) 信息插入：

用户信息插入、域信息插入、赛事信息插入、训练信息插入、题目信息插入、提交记录信息插入、消息信息插入、角色信息插入

- 2) 信息查询

题目信息查询、域信息查询、赛事信息查询、训练信息查询、提交记录查询、用户信息查询、消息信息查询、角色信息查询

3) 信息修改:

用户信息修改、域信息修改、赛事信息修改、训练信息修改、题目信息修改、角色信息修改

4) 用户信息删除、域信息删除、赛事信息删除、训练信息删除、题目信息删除、提交记录信息删除、消息信息删除、角色信息删除

三、用户视图

因为每个域都是独立的所以仅针对单个域说明用户视图（未经修改的默认情况下）

3.1.游客（未注册用户，guest 层）

- 1) 查看题目
- 2) 查看训练
- 3) 查看比赛
- 4) 查看域

3.2.未加入该域的登录用户

除游客所有功能外:

- 1) 提交题目
- 2) 参加比赛
- 3) 参加训练
- 4) 查看已提交代码
- 5) 重测自己代码

3.3.域创建者

除上述所有功能外:

- 1) 管理员标志
- 2) 修改所有题目
- 3) 修改所有比赛
- 4) 修改所有训练计划
- 5) 重测所有代码
- 6) 查看所有代码

- 7) 上传题目
- 8) 制定比赛
- 9) 制定训练计划
- 10) 从题目下载数据
- 11) 修改域信息（包括加入删除用户）
- 12) 查看所有隐藏信息

四、需求分析

首先考虑本系统的现实需求：

因为现在算法学习的资源大都分布在网络上，且知识面较广，较为复杂。现在网络上各种 OJ 都已经基本成熟，但是节本都是面向个人的，因此一个针对自己校内的 OJ 非常有必要。鉴于 OJ 要求存储容量大，相应效率高，查询方式多样，因此我们采用无模式存储的 MongoDB 实现。

由于队伍以及学校的性质相似，因此系统并不单独区分来降低系统效率，而是使用了域的概念。

为了满足使用者更加个性化的需求，我们没有固定角色权限，而是向创建者分配 root 并给与其创建角色定制权限的功能。

除此之外，我们还配置了 Gravatar，方便用户定制个性化头像且不占用内部空间。

4.1 用户需求分析：

4.1.1 数据需求

- (1) 用户基本信息：
用户 ID、昵称、邮箱、密码 salt、密码 hash、
- (2) 提交记录：
提交编号、提交者 ID、题目 ID、运行时间、语言、代码、评测结果、得分、内存占用、测试点、编译器信息、评测时间、评测者编号、所在域编号、是否重测。
- (3) 题目信息：
题目编号、贡献用户 ID、标题、内容、数据、提交人数、通过人数、参加人数、题目规则、是否为隐藏题目、所属域 ID
- (4) 域信息：
域名城、域编号、域所连接的 gravatar 邮箱、创件用户 ID
- (5) 比赛记录信息（对个人、排名）
用户 ID、比赛编号、所属域 ID、是否参加、提交次数、通过题目数、每个题的当前信息（通没通过）、总罚时
- (6) 比赛提交信息：（单次提交）
用户 ID、比赛编号、提交编号

型

(7) 域用户信息:

域编号、用户 ID、做过题目数量、提交次数、通过次数、角色类

(8) 角色信息:

所属域 ID、角色类型、角色能使用的权限 (一个 perm 值)

(9) 传送站内信息的信息:

发送用户 ID、接受用户 ID、发送内容、发送时间、接收情况

(10) 比赛/训练信息:

比赛/训练 ID、开始时间、结束时间、参加人数、比赛/训练信息、创建者、所属域 ID

4.1.2 事务需求

4.1.2.1 数据插入

- (1) 用户——创建——题目
- (2) 用户——创建——比赛/训练
- (3) 用户——创建——域
- (4) 用户——发消息——用户
- (5) 用户——提交题目——提交记录
- (6) 用户——添加——角色
- (7) 用户——创建——用户 (注册)

4.1.2.2 数据更新

- (1) 用户——加入——域
- (2) 用户——修改——题目
- (3) 用户——修改——角色权限
- (4) 用户——提交题目——比赛记录
- (5) 用户——修改——用户信息

4.1.2.3 数据查询

- (1) 用户按照域/题目/比赛/训练 ID 查看域/题目/比赛/训练
- (2) 用户按照用户 ID、题目 ID 搜索提交记录
- (3) 用户按照用户 ID 搜索用户

4.1.2.4 数据删除

- (1) 管理员——删除——域/题目/比赛/训练/做题记录/消息

**只添加了主要的操作, 剩下的操作不再赘述。*

4.2 系统需求

该系统可以搭建在多个服务器上, 针对于每个服务器需求如下:

该系统需要保存大量的题目数据、题面、每次提交记录、代码、返回

结果。要求服务器容量足够大，能够保存 2-3 年的数据。

该系统需要使用大量复杂的查询方法，数据库内部可以牺牲一些空间来换取更高效的查询方法。

4.2.1 网络需求

至少支持 100 人同时访问并参加比赛，系统使用 `asyncio` 异步处理，模板搭建，能够高效并发响应。

4.2.2 性能

高峰期:比赛进行过程中(涉及到比赛数据实时录入)和中午晚上(数据查询高峰期)

要求:

- (1)高峰期比赛期间数据录入的延迟不得高 3s，平时不超过 1s
- (2)高峰期查询不超过 2s，平时不超过 1s

4.2.3 安全性

- (1) 数据库必须有口令保护，用户密码用 `pbkdf2_hmac` 加密，加盐迭代 10000 次
- (2) 每个用户分配特定的用户视图所应有的访问权限
- (3) 用户只能在适合他们完成工作需要的窗口中看到需要的数据

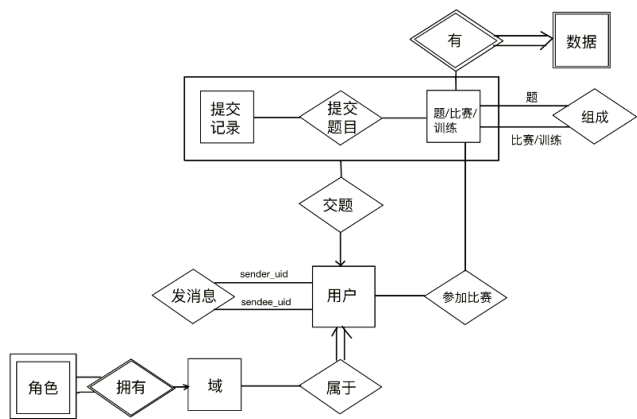
4.2.4 备份和恢复

每天晚上 24 点对数据库进行备份。

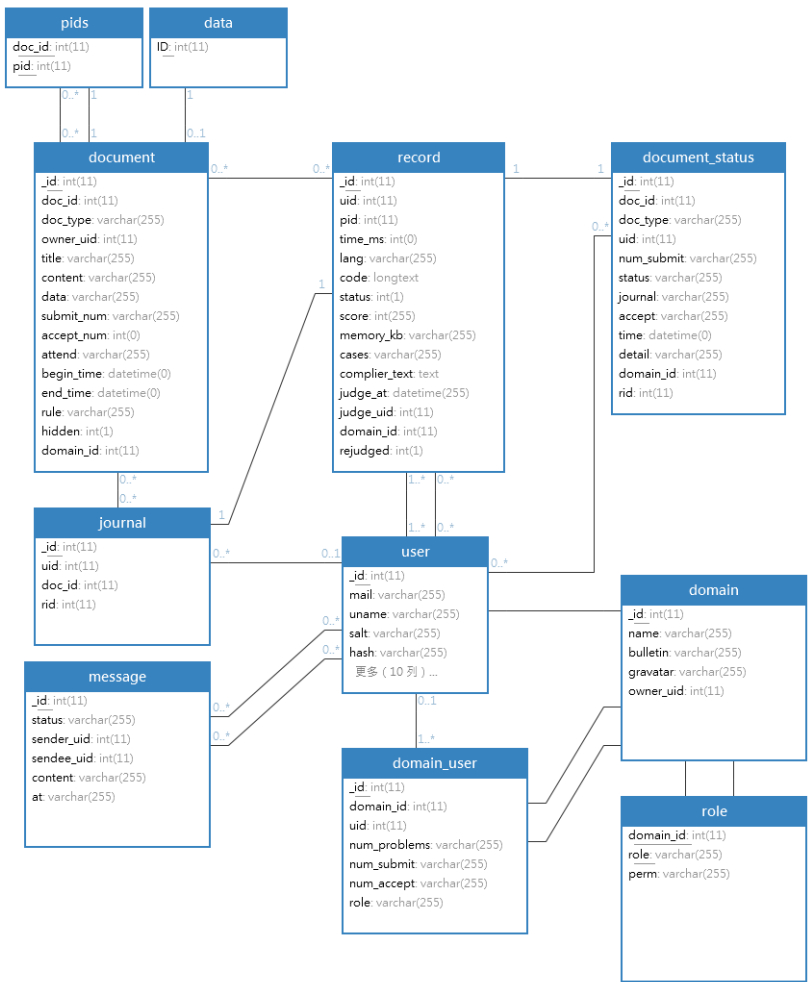
五、逻辑设计

5.1 ER 图

手绘版



可视化版详细 ER 图:



5.2 数据字典

5.2.1 题目信息

名	类型	长度	小数点	不是 null	虚拟	
▸ _id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 1
doc_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
doc_type	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
owner_uid	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
title	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
content	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
data	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
submit_num	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
accept_num	int	0	0	<input type="checkbox"/>	<input type="checkbox"/>	
attend	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
begin_time	datetime	0	0	<input type="checkbox"/>	<input type="checkbox"/>	
end_time	datetime	0	0	<input type="checkbox"/>	<input type="checkbox"/>	
rule	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
hidden	int	1	0	<input type="checkbox"/>	<input type="checkbox"/>	
domain_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

5.2.2 赛事/训练信息

名	类型	长度	小数点	不是 null	虚拟	
_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 1
doc_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
doc_type	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
X uid	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
num_submit	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
status	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
journal	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
accept	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
time	datetime	0	0	<input type="checkbox"/>	<input type="checkbox"/>	
detail	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
domain_id	int	11	0	<input type="checkbox"/>	<input type="checkbox"/>	
rid	int	11	0	<input type="checkbox"/>	<input type="checkbox"/>	

5.2.3 角色信息

名	类型	长度	小数点	不是 null	虚拟	
▸ domain_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 1
role	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 2
perm	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	

5.2.4 用户信息

名	类型	长度	小数点	不是 null	虚拟	
_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 1
mail	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
uname	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
salt	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
hash	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
gravatar	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
bio	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
gender	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
qq	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
show_bio	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
show_gender	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
show_mail	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
show_qq	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
show_wechat	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
wechat	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	

5.2.5 提交记录

名	类型	长度	小数点	不是 null	虚拟	
_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 1
uid	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
pid	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
time_ms	int	0	0	<input type="checkbox"/>	<input type="checkbox"/>	
lang	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
code	longtext	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
status	int	1	0	<input type="checkbox"/>	<input type="checkbox"/>	
score	int	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
memory_kb	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
cases	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
complier_text	text	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
judge_at	datetime	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
judge_uid	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
domain_id	int	11	0	<input type="checkbox"/>	<input type="checkbox"/>	
rejudged	int	1	0	<input type="checkbox"/>	<input type="checkbox"/>	

5.2.6 站内信信息

名	类型	长度	小数点	不是 null	虚拟	
_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 1
status	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
sender_uid	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
sendee_uid	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
content	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
at	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	

5.2.7 域用户信息

名	类型	长度	小数点	不是 null	虚拟	
_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 1
domain_id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
uid	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
num_problems	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
num_submit	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
num_accept	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
role	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	

六、物理设计

6.1 权限的实现

每个用户的在不同的所属域中都有着不同的角色，在角色关系中存在有一个 perm 值，将每个可分割的权限映射成一个二进制数，即若该角色拥有第 i 个权限，则该处的 perm 值的二进制数的第 i 位应该为 1，否则为 0。

6.2 主要事务分析

如前几个章节所分析的那样，我们采用的是 MongoDB 而非 SQL 所以下面的语句全部用 MongoDB 的操作语言来代替。主要以 document 关系为例展示四类事务的实现。

6.2.1 插入数据

插入一项

```

obj_id = objectid.ObjectId()
coll = db.coll('document')
doc = {'_id': obj_id,
       'content': content,
       'owner_uid': owner_uid,
       'domain_id': domain_id,
       'doc_type': doc_type,
       'doc_id': doc_id or obj_id,
       **kwargs}
if parent_doc_type or parent_doc_id:
    assert parent_doc_type and parent_doc_id
    doc['parent_doc_type'], doc['parent_doc_id'] = parent_doc_type, parent_doc_id
await coll.insert_one(doc)

```

6.2.2 查询数据

```

@argmethod.wrap
async def get(domain_id: str, doc_type: int, doc_id: convert_doc_id, fields=None):
    coll = db.coll('document')
    return await coll.find_one({'domain_id': domain_id,
                                'doc_type': doc_type,
                                'doc_id': doc_id}, projection=fields)

```

6.2.3 更新数据

```

async def set(domain_id: str, doc_type: int, doc_id: convert_doc_id, **kwargs):
    coll = db.coll('document')
    doc = await coll.find_one_and_update(filter={'domain_id': domain_id,
                                                  'doc_type': doc_type,
                                                  'doc_id': doc_id},
                                         update={'$set': kwargs},
                                         return_document=ReturnDocument.AFTER)
    return doc

```

6.2.4 删除数据

```

async def delete(domain_id: str, doc_type: int, doc_id: convert_doc_id):
    coll = db.coll('document')
    return await coll.delete_one({'domain_id': domain_id,
                                  'doc_type': doc_type,
                                  'doc_id': doc_id})

```

6.3 Gravatar 的配置

gravatar 的配置根据网上教程即可，过程并不复杂，只需要查询到用户信息内填写的 gravatar 邮箱的 MD5 值+读取头像的服务器地址再加上头像尺寸和获取的头像等级这几个参数就能直接拼接出获取头像的对应地址。封装的原码参考了

网上流传的代码：

```
def gravatar_url(gravatar, size=200):  
    if gravatar:  
        gravatar_hash = hashlib.md5(gravatar.lower().encode()).hexdigest()  
    else:  
        gravatar_hash = ''  
    return ('//cn.gravatar.com/avatar/' + gravatar_hash + "?" +  
            parse.urlencode({'d': 'mm', 's': str(size)}))
```

七、界面设计

界面仿照 vijos 改写，修改元素以满足需求。

7.1 网站首页



7.2 登录界面

关闭

登录

用户名

密码

☐ 记住我

登录

7.3 查看题目界面

139.196.96.35

首页 · 题库 · 训练 · 比赛 · 评测队列

sdu-public



ICPC

首页 / 题库 /

ZJM的烦恼（原版数据范围）

本题为ZJM的烦恼（原新生赛H题）在没削数据范围之前的题目，大家可以随便玩（被）一（毒）玩（瘤）

题目背景

ZJM：吾心行澄如明镜，所作皆为正义
ZJM 算起数来总是正确的，他为了证明自己，向cy保证，他可以把任意的小数精准化成分数，可他发现对他来说这是很困难的，为了证明自己，他选择向你求助。

题目描述

给你一个小数，循环节将会使用括号括起来，让你去求出这个小数对应的分数。

提交

编辑

设置

信息

提交	✓ Accepted
提交数	2
我的提交数	2

7.4 提交界面



7.5 比赛界面

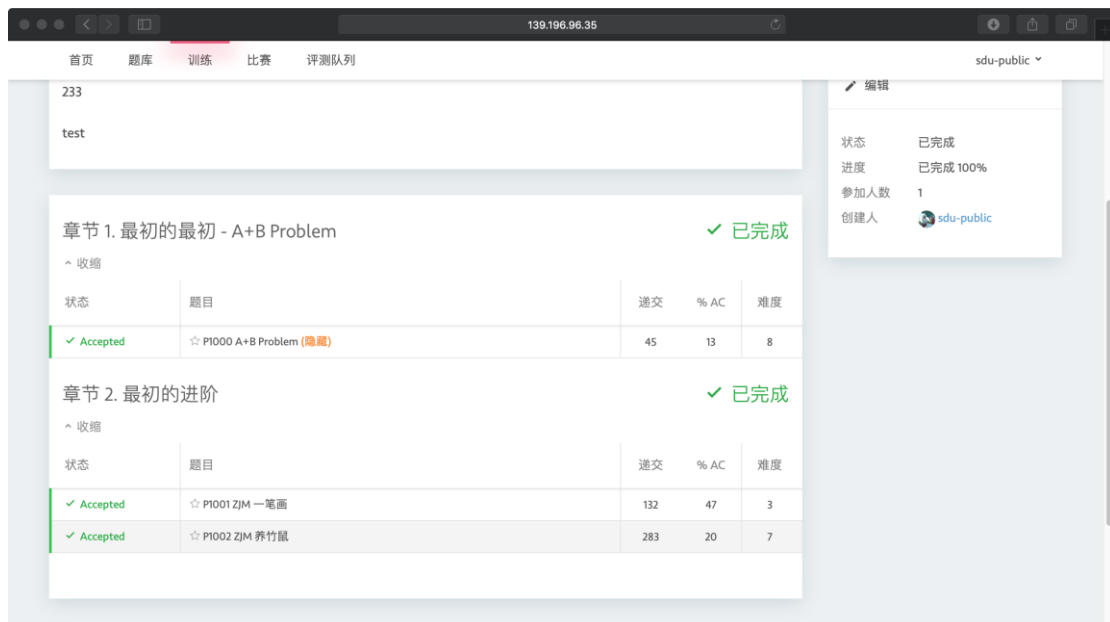
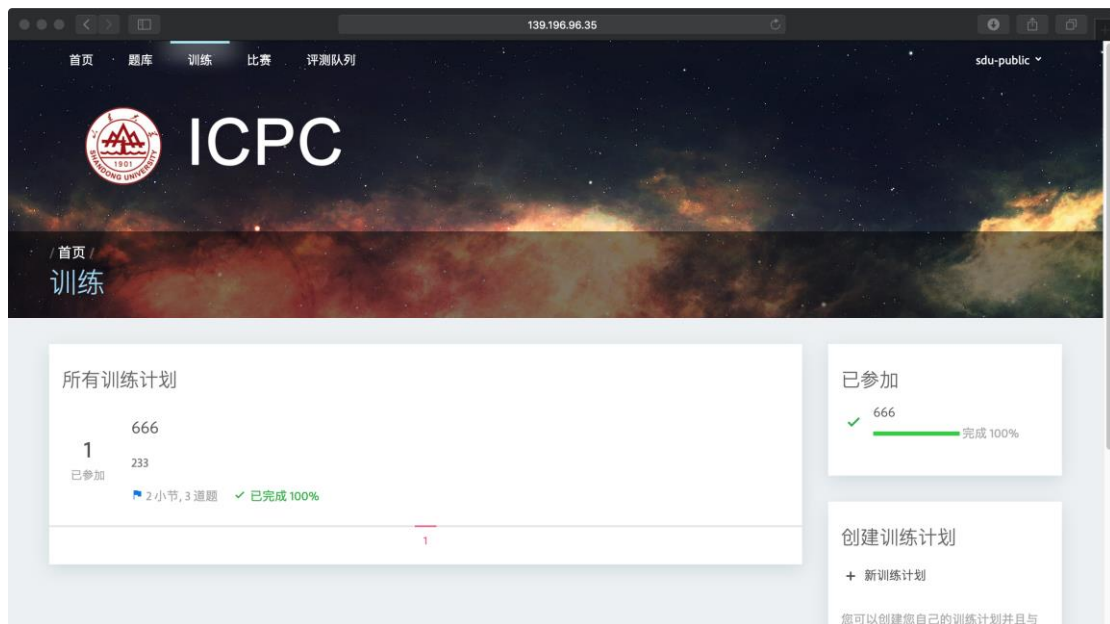


<div> 首页 题库 训练 比赛 评测队列 </div> <div>sdu-public</div>																																																																													
<div> / 首页 / 比赛 / 2018 SDU 程序设计竞赛新生赛 / </div> <div>比赛成绩表</div>																																																																													
<div> <div>导出为 HTML</div> <table> <tr> <th>排名</th><th>用户</th><th>解决题目</th><th>#1</th><th>#2</th><th>#3</th><th>#4</th><th>#5</th><th>#6</th><th>#7</th><th>#8</th><th>#9</th><th>#10</th></tr> <tr> <td>1</td><td>team42</td><td>10</td><td>Accepted 02:14:35</td><td>Accepted (-1) 02:51:03</td><td>Accepted 04:11:58</td><td>Accepted 03:06:56</td><td>Accepted 00:21:24</td><td>Accepted 00:16:24</td><td>Accepted (-2) 04:07:14</td><td>Accepted 02:43:28</td><td>Accepted (-4) 06:10:57</td><td>Accepted 02:08:37</td></tr> <tr> <td>2</td><td>team36</td><td>9</td><td>Accepted (-1) 00:23:24</td><td>Accepted 00:07:35</td><td>- -</td><td>Accepted (-13) 07:52:26</td><td>Accepted 00:42:01</td><td>Accepted 00:57:28</td><td>Accepted (-1) 01:44:07</td><td>Accepted 03:04:19</td><td>Accepted (-2) 04:57:03</td><td>Accepted 02:09:20</td></tr> <tr> <td>3</td><td>team68</td><td>9</td><td>Accepted (-1) 00:35:12</td><td>Accepted 00:18:11</td><td>- -</td><td>Accepted (-7) 04:35:44</td><td>Accepted 00:50:33</td><td>Accepted 00:57:17</td><td>Accepted 03:12:23</td><td>Accepted 04:44:21</td><td>Accepted (-2) 05:10:43</td><td>Accepted (-4) 05:54:47</td></tr> <tr> <td>4</td><td>team54</td><td>8</td><td>Accepted 00:04:09</td><td>Accepted (-3) 03:40:08</td><td>- -</td><td>Accepted (-1) 03:28:07</td><td>Accepted 00:41:56</td><td>Accepted 01:07:46</td><td>Accepted 03:58:05</td><td>Accepted 01:40:12</td><td>Accepted (-3) 03:10:19</td><td>-1 -</td></tr> </table> </div>													排名	用户	解决题目	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	1	team42	10	Accepted 02:14:35	Accepted (-1) 02:51:03	Accepted 04:11:58	Accepted 03:06:56	Accepted 00:21:24	Accepted 00:16:24	Accepted (-2) 04:07:14	Accepted 02:43:28	Accepted (-4) 06:10:57	Accepted 02:08:37	2	team36	9	Accepted (-1) 00:23:24	Accepted 00:07:35	- -	Accepted (-13) 07:52:26	Accepted 00:42:01	Accepted 00:57:28	Accepted (-1) 01:44:07	Accepted 03:04:19	Accepted (-2) 04:57:03	Accepted 02:09:20	3	team68	9	Accepted (-1) 00:35:12	Accepted 00:18:11	- -	Accepted (-7) 04:35:44	Accepted 00:50:33	Accepted 00:57:17	Accepted 03:12:23	Accepted 04:44:21	Accepted (-2) 05:10:43	Accepted (-4) 05:54:47	4	team54	8	Accepted 00:04:09	Accepted (-3) 03:40:08	- -	Accepted (-1) 03:28:07	Accepted 00:41:56	Accepted 01:07:46	Accepted 03:58:05	Accepted 01:40:12	Accepted (-3) 03:10:19	-1 -
排名	用户	解决题目	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10																																																																	
1	team42	10	Accepted 02:14:35	Accepted (-1) 02:51:03	Accepted 04:11:58	Accepted 03:06:56	Accepted 00:21:24	Accepted 00:16:24	Accepted (-2) 04:07:14	Accepted 02:43:28	Accepted (-4) 06:10:57	Accepted 02:08:37																																																																	
2	team36	9	Accepted (-1) 00:23:24	Accepted 00:07:35	- -	Accepted (-13) 07:52:26	Accepted 00:42:01	Accepted 00:57:28	Accepted (-1) 01:44:07	Accepted 03:04:19	Accepted (-2) 04:57:03	Accepted 02:09:20																																																																	
3	team68	9	Accepted (-1) 00:35:12	Accepted 00:18:11	- -	Accepted (-7) 04:35:44	Accepted 00:50:33	Accepted 00:57:17	Accepted 03:12:23	Accepted 04:44:21	Accepted (-2) 05:10:43	Accepted (-4) 05:54:47																																																																	
4	team54	8	Accepted 00:04:09	Accepted (-3) 03:40:08	- -	Accepted (-1) 03:28:07	Accepted 00:41:56	Accepted 01:07:46	Accepted 03:58:05	Accepted 01:40:12	Accepted (-3) 03:10:19	-1 -																																																																	

7.6 比赛成绩表查看

<div> 首页 题库 训练 比赛 评测队列 </div> <div>139.198.96.35</div> <div>sdu-public</div>																																																																													
<div> / 首页 / 比赛 / 2018 SDU 程序设计竞赛新生赛 / </div> <div>比赛成绩表</div>																																																																													
<div> <div>导出为 HTML</div> <table> <tr> <th>排名</th><th>用户</th><th>解决题目</th><th>#1</th><th>#2</th><th>#3</th><th>#4</th><th>#5</th><th>#6</th><th>#7</th><th>#8</th><th>#9</th><th>#10</th></tr> <tr> <td>1</td><td>team42</td><td>10</td><td>Accepted 02:14:35</td><td>Accepted (-1) 02:51:03</td><td>Accepted 04:11:58</td><td>Accepted 03:06:56</td><td>Accepted 00:21:24</td><td>Accepted 00:16:24</td><td>Accepted (-2) 04:07:14</td><td>Accepted 02:43:28</td><td>Accepted (-4) 06:10:57</td><td>Accepted 02:08:37</td></tr> <tr> <td>2</td><td>team36</td><td>9</td><td>Accepted (-1) 00:23:24</td><td>Accepted 00:07:35</td><td>- -</td><td>Accepted (-13) 07:52:26</td><td>Accepted 00:42:01</td><td>Accepted 00:57:28</td><td>Accepted (-1) 01:44:07</td><td>Accepted 03:04:19</td><td>Accepted (-2) 04:57:03</td><td>Accepted 02:09:20</td></tr> <tr> <td>3</td><td>team68</td><td>9</td><td>Accepted (-1) 00:35:12</td><td>Accepted 00:18:11</td><td>- -</td><td>Accepted (-7) 04:35:44</td><td>Accepted 00:50:33</td><td>Accepted 00:57:17</td><td>Accepted 03:12:23</td><td>Accepted 04:44:21</td><td>Accepted (-2) 05:10:43</td><td>Accepted (-4) 05:54:47</td></tr> <tr> <td>4</td><td>team54</td><td>8</td><td>Accepted 00:04:09</td><td>Accepted (-3) 03:40:08</td><td>- -</td><td>Accepted (-1) 03:28:07</td><td>Accepted 00:41:56</td><td>Accepted 01:07:46</td><td>Accepted 03:58:05</td><td>Accepted 01:40:12</td><td>Accepted (-3) 03:10:19</td><td>-1 -</td></tr> </table> </div>													排名	用户	解决题目	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	1	team42	10	Accepted 02:14:35	Accepted (-1) 02:51:03	Accepted 04:11:58	Accepted 03:06:56	Accepted 00:21:24	Accepted 00:16:24	Accepted (-2) 04:07:14	Accepted 02:43:28	Accepted (-4) 06:10:57	Accepted 02:08:37	2	team36	9	Accepted (-1) 00:23:24	Accepted 00:07:35	- -	Accepted (-13) 07:52:26	Accepted 00:42:01	Accepted 00:57:28	Accepted (-1) 01:44:07	Accepted 03:04:19	Accepted (-2) 04:57:03	Accepted 02:09:20	3	team68	9	Accepted (-1) 00:35:12	Accepted 00:18:11	- -	Accepted (-7) 04:35:44	Accepted 00:50:33	Accepted 00:57:17	Accepted 03:12:23	Accepted 04:44:21	Accepted (-2) 05:10:43	Accepted (-4) 05:54:47	4	team54	8	Accepted 00:04:09	Accepted (-3) 03:40:08	- -	Accepted (-1) 03:28:07	Accepted 00:41:56	Accepted 01:07:46	Accepted 03:58:05	Accepted 01:40:12	Accepted (-3) 03:10:19	-1 -
排名	用户	解决题目	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10																																																																	
1	team42	10	Accepted 02:14:35	Accepted (-1) 02:51:03	Accepted 04:11:58	Accepted 03:06:56	Accepted 00:21:24	Accepted 00:16:24	Accepted (-2) 04:07:14	Accepted 02:43:28	Accepted (-4) 06:10:57	Accepted 02:08:37																																																																	
2	team36	9	Accepted (-1) 00:23:24	Accepted 00:07:35	- -	Accepted (-13) 07:52:26	Accepted 00:42:01	Accepted 00:57:28	Accepted (-1) 01:44:07	Accepted 03:04:19	Accepted (-2) 04:57:03	Accepted 02:09:20																																																																	
3	team68	9	Accepted (-1) 00:35:12	Accepted 00:18:11	- -	Accepted (-7) 04:35:44	Accepted 00:50:33	Accepted 00:57:17	Accepted 03:12:23	Accepted 04:44:21	Accepted (-2) 05:10:43	Accepted (-4) 05:54:47																																																																	
4	team54	8	Accepted 00:04:09	Accepted (-3) 03:40:08	- -	Accepted (-1) 03:28:07	Accepted 00:41:56	Accepted 01:07:46	Accepted 03:58:05	Accepted 01:40:12	Accepted (-3) 03:10:19	-1 -																																																																	

7.7 训练计划查看界面



7.8 评测队列查看界面

139.196.96.35

首页 题库 训练 比赛 评测队列 sdu-public

评测队列

过滤

由用户名或 UID

由题目

过滤 重置

状态	题目	提交者	时间	内存	语言	提交时间
Accepted	重测 P1000 A+B Problem (隐藏)	sdu-public	353ms	968.0 KiB	C++	1月前
Runtime Error	重测 P1000 A+B Problem (隐藏)	sdu-public	4057ms	8.223 MiB	C++	1月前
Wrong Answer	重测 P1000 A+B Problem (隐藏)	sdu-public	124ms	1.051 MiB	C++	1月前
Runtime Error	重测 堆.2	sdu-public	31ms	844.0 KiB	C++	1月前
Wrong Answer	重测 P1000 A+B Problem (隐藏)	sdu-public	159ms	1.199 MiB	C++	1月前
Wrong Answer	重测 P1000 A+B Problem (隐藏)	sdu-public	140ms	1.137 MiB	C++	1月前
Accepted	重测 P1004 ZJM's Dream	team48	9ms	212.0 KiB	C	1月前

7.9 查看一条评测结果

记录详情

Accepted

#	状态	耗时	内存占用
#1	Accepted	1ms	340.0 KiB
#2	Accepted	1ms	344.0 KiB
#3	Accepted	2ms	384.0 KiB
#4	Accepted	1ms	344.0 KiB
#5	Accepted	1ms	488.0 KiB
#6	Accepted	1ms	340.0 KiB
#7	Accepted	2ms	384.0 KiB
#8	Accepted	1ms	340.0 KiB
#9	Accepted	1ms	344.0 KiB
#10	Accepted	1ms	340.0 KiB

信息

重测 取消成绩

提交者 sdu-public LV 1

类型 递交

题目 ZJM 爱刷题

题目数据 下载

语言 C++

提交时间 1月前

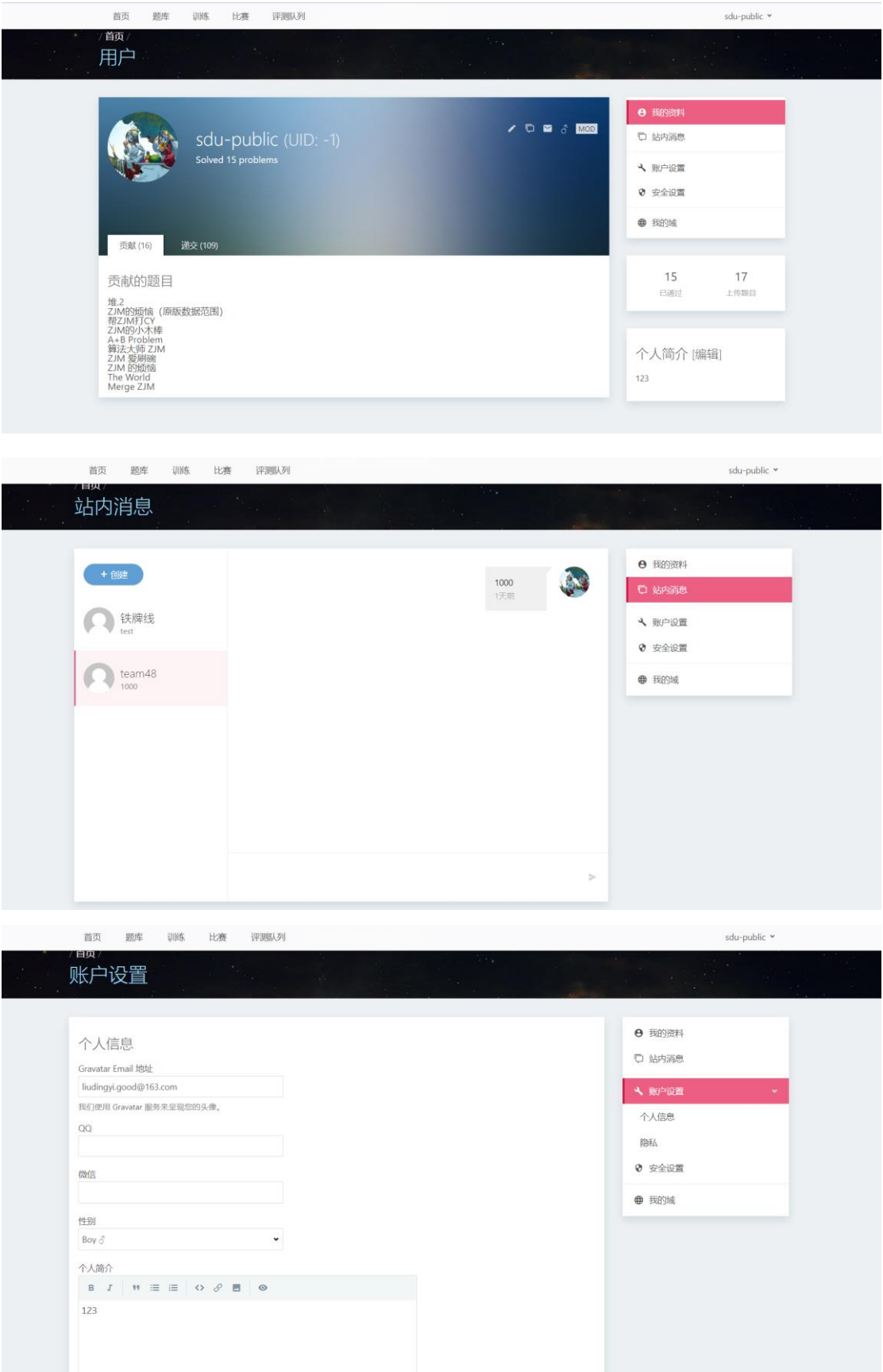
评测时间 1月前

评测机 hrz

总耗时 17ms

峰值内存 488.0 KiB

7.10 用户界面



首页题库训练比赛评测队列

sdu-public

隐私

Email 可见性

公开

QQ 可见性

公开

微信可见性

公开

性别可见性

公开

个人简介可见性

公开

保存所有修改

我的资料

站内消息

账户设置

个人信息

隐私

安全设置

我的域

首页题库训练比赛评测队列

sdu-public

1901

SHANDONG UNIVERSITY

/ 首页 /

安全设置

修改电子邮件

当前密码 (必填)

当前电子邮件

sdu-public@sdu.edu.cn

新电子邮件 (必填)

修改电子邮件

修改密码

当前密码 (必填)

新密码 (必填)

重复密码 (必填)

修改密码

我的资料

站内消息

账户设置

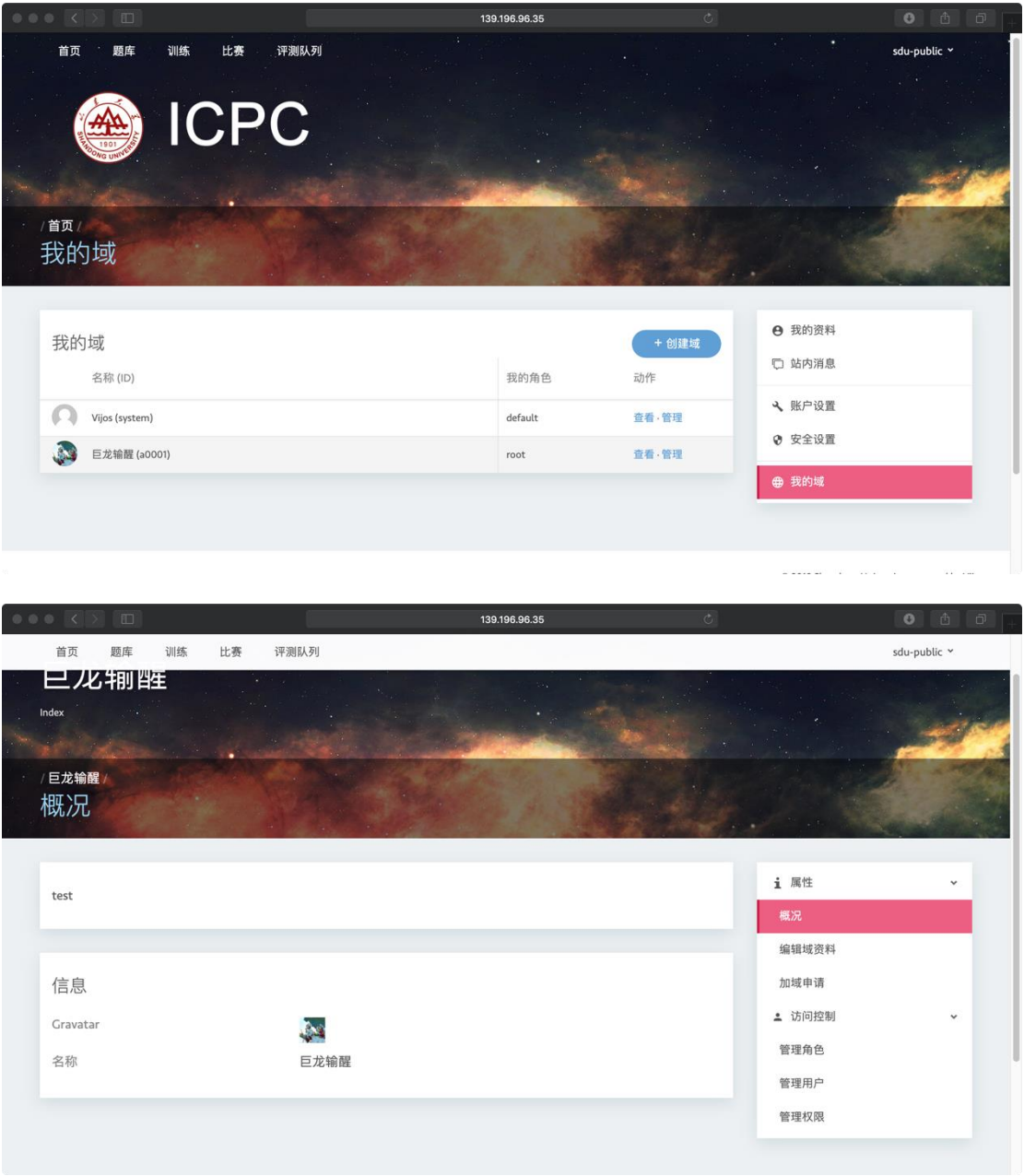
安全设置

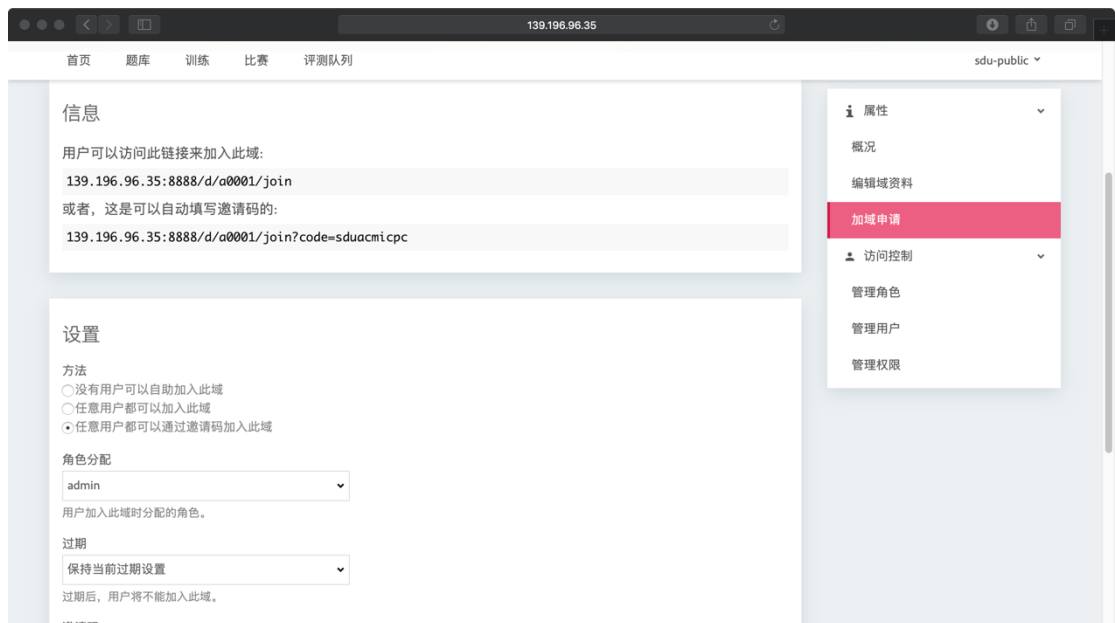
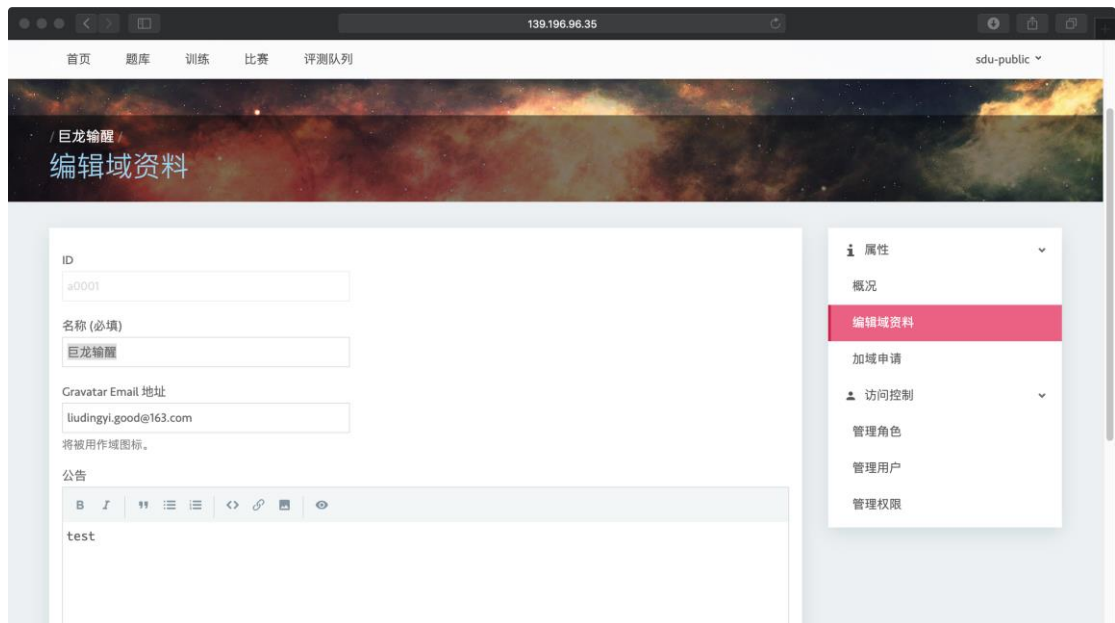
修改电子邮件

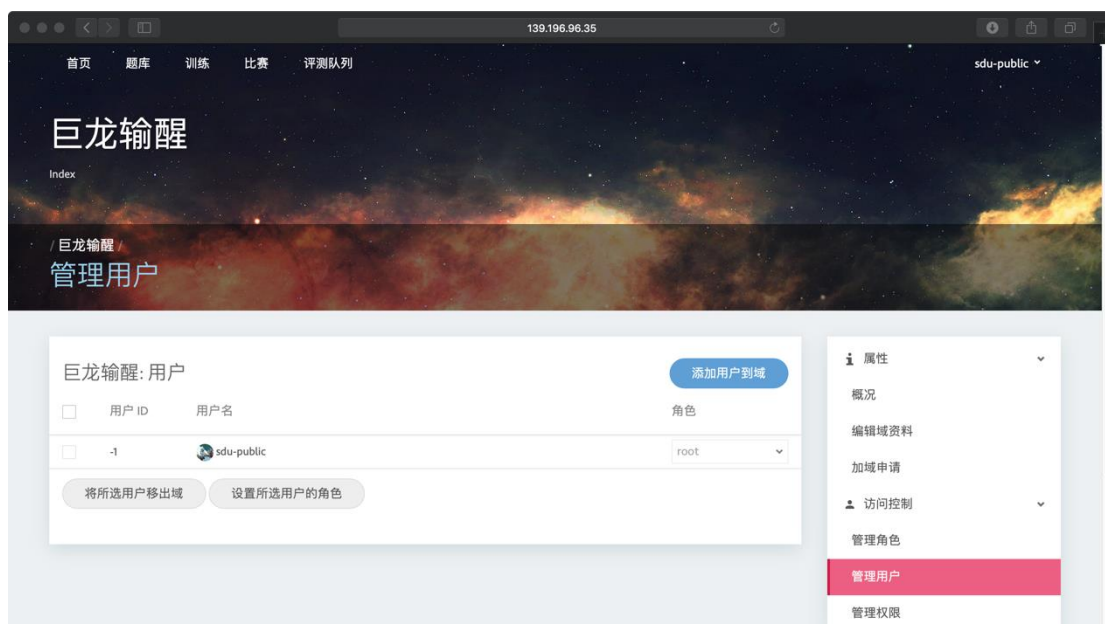
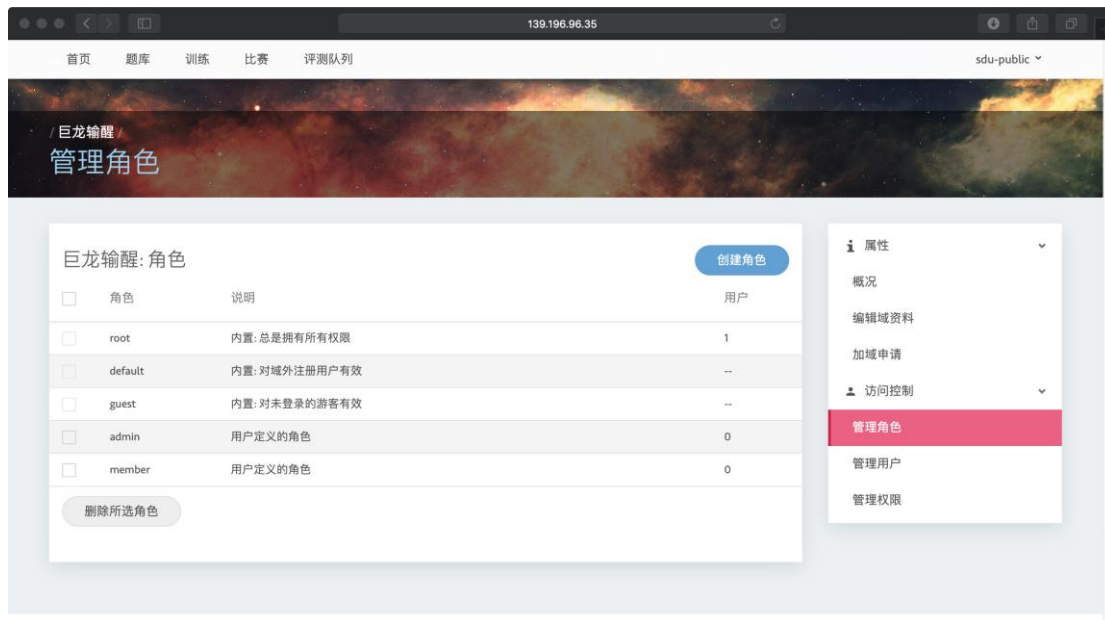
修改密码

我的域

7.11 域界面







139.196.96.35

首页 题库 训练 比赛 评测队列

sdu-public

巨龙输醒 / 管理权限

巨龙输醒: 权限

权限	admin	default	guest	member
一般				
查看此域	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
修改角色权限	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
展示 MOD 徽章	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
修改此域描述	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
题库				
创建题目	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
修改题目	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
修改自己的题目	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
查看题目	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

属性

概况

编辑域资料

加域申请

访问控制

管理角色

管理用户

管理权限

八、测试与应用

2018 SDU 程序设计新生赛 12.9 - 12.10



九、总结

9.1 系统优点

- 1) 系统功能的基本功能实现完全。能够满足所有基本的 OJ 操作需求。很好的解决了需求分析中的问题。
- 2) 总体上借鉴了 Vijos 的架构使用了 MongoDB 来替换 MySQL 极大的压缩了数据库中信息所占用的存储空间，并且简化了代码的工作量。
- 3) 没有对用户权限作确定分类，而是使用了 hash 的方法，将多维权限状态压缩成一个二进制数来存储，以使用户自己定义自己所需要的角色和角色所能拥有的权限内容。
- 4) 因为队伍和学校架构的相似之处统一采用域的概念进行划分，这样简化了内核代码量以及界面设计的工作量。同时也兼顾了不同用户拥有各自独立空间的需求。
- 5) 在用户信息中配置了 Gravatar (Gravatar 是 Globally Recognized Avatar 的缩写,是 gravatar 推出的一项服务，意为“全球通用头像”。如果在 Gravatar 的服务器上放置了你自己的头像，那么在任何支持 Gravatar 的 blog 或者留言本上留言时，只要提供你与这个头像关联的 email 地址，就能够显示出你的 Gravatar 头像来。) 这既让用户信息拥有了头像，又节省出了存储用户头像的空间。而且方便了已经注册过的用户，真正与互联网接轨。
- 6) 大量使用标准化的 async 异步函数来处理大量的并发访问。
- 7) 界面简洁明了，操作和普通 OJ 相似，更容易上手使用。
- 8) 录入题面、创建比赛和训练时使用了 commonMark 语法，方便了内部代码的实现，简化了数据库内部的结构（原本要区别题目描述、数据范围、样例输入、样例输出等等），且给予用户更大的自由度来构建题目和比赛。
- 9) 在 2018ACM 新生赛中就已经投入使用，实践证明了该系统拥有较好的稳定性。

9.2 系统不足

- 1) 没有相应的网站使用说明书以及很明显的操作错误提示。
- 2) 因为主要针对校内训练，因此没有开放注册功能，只能在后台创建注册信息。
- 3) 还有很多有用的功能比如：题解、讨论、滚榜等功能没有实现。
- 4) 由于获取不到题目测试数据因此题库中目前只有 2018ACM 新生赛使用过的原创题目，后期应该会使用爬虫获取各大 OJ 的题干，在本地提交后会统一使用公共账号向对应 OJ 发送 submit 请求，再将评测结果返回至本地，用这种方法来解决大量题目没有数据的问题。
- 5) 时间原因数据库内部应该还存在很多 bug，后面会统一内部开放二次测试，来检测内部实现的 bug 和 UI 设计的不足之处。

9.3 经验与收获

1) 第一次合作开发一个完整的程序，从需求分析、逻辑设计、物理设计，从 UI 到内核到算法，大半个学期的时间，我们感受到了艰难。开发一个程序实属不易。

2) 加深了对 OJ 的认识，从一个使用者的身份向开发者身份转换，我们了解到了一些简单操作内部的实现原理。比如一个头像的获取，我们需要根据协议拼接出获取头像的链接，再用服务器从链接获取数据并加载。

3) 更加认识到了 python 语言的强大，python 拥有大量时候服务器处理请求的模板库，能够很好的链接 html 文本，是链接前端后端的有力工具。

4) 最重要的是我们真正理解了数据库设计的重要。每一次请求返回结果的效率，信息插入的效率，信息更新的效率，数据存储占空间的大小都与我们的设计方法息息相关。一个好的数据库架构，可以更好地为程序本身服务，不管对运行效率还是存储效率都有很大的提升。

十、人员分工

成员姓名	工作量
黄瑞哲	1.代码的调试与编写 2.程序维护运行工作 3.UI 设计与代码的 debug 工作 4.数据库逻辑设计
刘丁泚	1.需求分析&初始方案策划 2.实验报告编写 3.UI 与代码的 debug 工作 4.数据库逻辑设计
杨家齐	1.展示报告编写 2.第一次大型测试工作 3.逻辑设计&逻辑设计报告编写