



中期交付 文档

iKNOW大学生竞赛平台项目小组

马福崑	2015013186
侯建国	2015013212
陈 超	2015013222
胡 哲	2015013187



一. 产品名称

iKNOW大学生竞赛平台

二. 网站链接&源代码链接地址

网站链接: <http://118.89.240.107:8000>

源代码链接地址: <https://gitlab.thsecourse.com/iKNOW-CPCS/iKNOW-Project>

后台管理员账号: iKNOW 密码: iKNOWHCMH

三. 产品目标

1. 系统功能

iKNOW大学生竞赛平台, 致力于实现当前市场现有竞赛平台网站所具有的基本功能, 并在此基础上充分考虑用户需求, 开发出新的功能, 以提高竞争力。当前已实现的功能主要包括:

- 完整的注册、登录以及邮箱验证功能, 包括参赛者和组织者两种类型
- 用户可以进行个人信息设置
- 用户可以查看当前各种比赛信息, 并可收藏比赛
- 用户可以查看当前发布的各种教程, 包括官方教程(由组织者发布)和(学友分享)
- 用户之间可以进行私信往来

2. 性能方面

暂未完成

四. 开发组织管理

1. 开发过程

第一次迭代(10.27-11.9): 准备开发环境, 完善所需配置; 学习Vue.js. 实现基本的注册、登录功能, 包括前端对应页面及后端对应API, 对应数据模型。

第二次迭代(11.10-11.23): 完善注册、登录功能。实现首页、个人/组织主页、官方教程、学友分享等页面及其所需要的API、数据模型; 实现教程、私信系统的部分功能。

2. 成员分工

大体上, 前端由侯建国和陈超负责, 后端由马福崑和胡哲负责。前后端交互则由双方共同完成, 每个人在小组内贡献度均等。

3. 开发环境

Ubuntu16.04 LTS

4. 配置管理

前端部分: node.js 9.0.0
npm 5.5.1
vue.js 2.9.1
后端部分: python 3.5.2
django 1.11.7
mysql 5.7.20
django-rest-framework 3.7.3

五. 系统设计

1. 前端交互

前端模块的相关内容列表如下:

Views: 每一个vue文件是一个页面, 完成了平台主要页面的设计

Components: 页面布局中的一些组件设计, 包括导航栏, 个人信息设置等

Router: 提供了页面跳转的路由设置

2. 后端模块

后端模块的相关内容列表如下:

backend: 对各子模块进行统筹管理, 包含数据模型的定义;

userpage: 提供与用户相关的API, 其中实现了需要的序列化器、权限控制及URL配置;

tutorial: 提供与教程相关的API, 实现了需要的序列化器、权限控制及URL配置;

competition: 提供与比赛相关的API, 实现了需要的序列化器、权限控制及URL配置(按计划暂未完成);

forum: 提供与论坛相关的API, 实现了需要的序列化器、权限控制及URL配置(按计划暂未完成)。

3. 接口规范

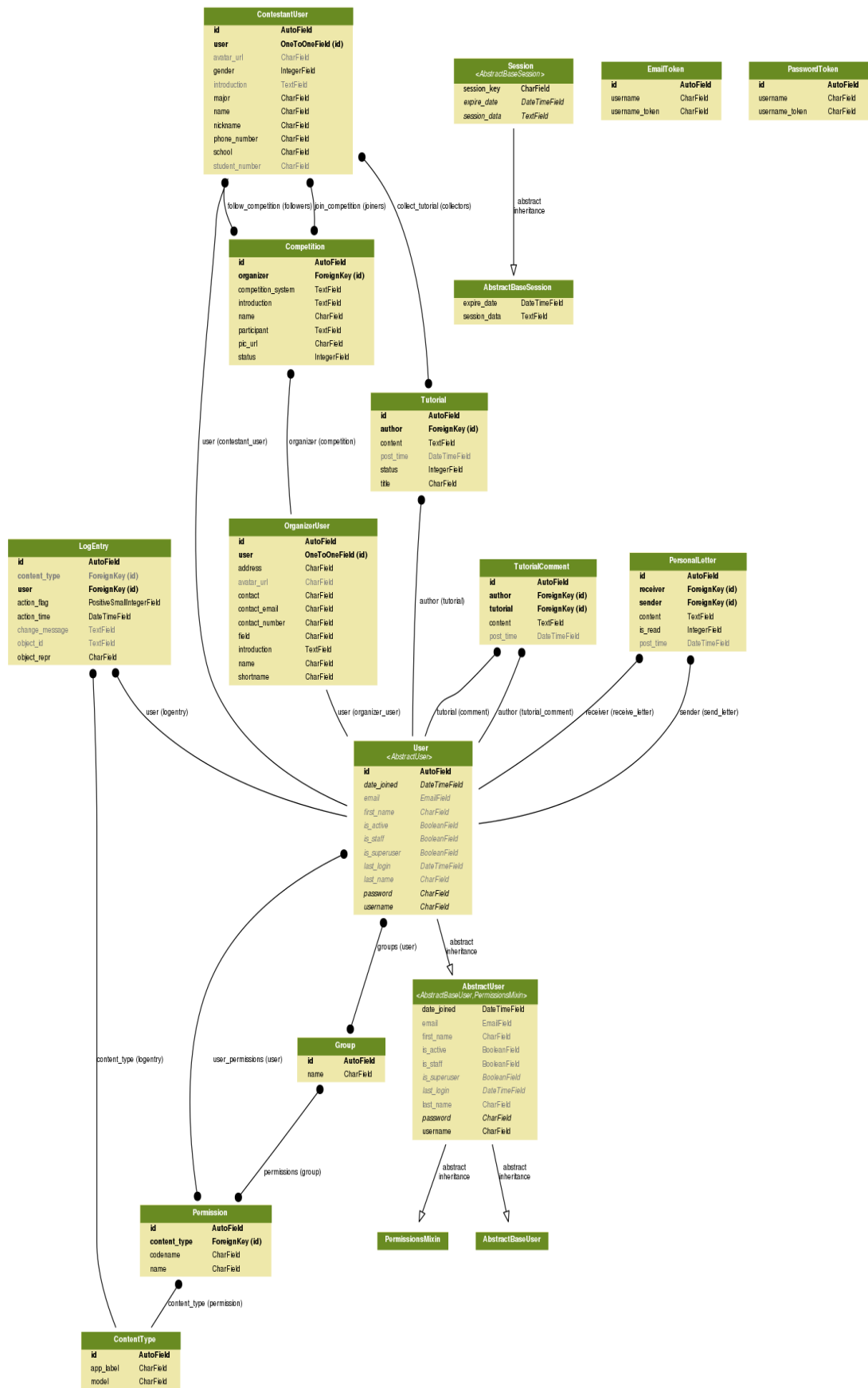
前后端之间的接口均为RESTful API, 详见附件一(第二次迭代接口设定v2.2)。

4. 数据库设计

数据库设计利用django提供的ORM进行，不直接对底层数据库进行操作。

决定数据库的django Model详见附件二（model.py）。

关系图如下：



五. 重点和难点问题及其解决方法

1. 新工具学习

为了更好地完成本项目，小组成员采用了一些新工具。学习新工具，更好地使用新工具便是一个重点，也是一个难点。各位成员在此过程中认真阅读官方文档、教程，从互联网上查阅大量相关资料，结合自己的摸索和实验，逐渐掌握了新工具，包括Vue.js与django-rest-framework。

2. 前后端结合

前后端结合的问题包括两个方面。

首先是工程结合的问题。本项目由Vue.js与django结合来实现，因此由Vue工程与django工程两个独立的部分。如何把这两个部分组合起来，正常地共同运行，是我们面临的一个大问题。通过参考网上多位前辈的技术博客，我们按照步骤依次操作，逐个解决过程中出现的问题，最终解决了这个大问题。

另一个方面是功能结合的问题。本项目利用RESTful API实现了前后端分离，如果一切顺利前后端功能结合应该并不困难。但是在实际操作中，由于我们对新工具的不熟练，对新架构的不熟悉，以及前后端的工程中难以完全避免的一些bug，前后端功能结合中出现了各式各样的奇怪问题，让人疲于应付。对此，我们也没有什么特别有效的方法，只能不断地从互联网上搜索相关内容作为参考，同时自己也不断摸索尝试。解决问题的过程中最令人沮丧的就是，互联网上很多的解决方案有着时间版本等差异，直接参考不能解决问题反而会有新的问题，进行调整又缺乏合适的思路。最终，我们通过堆时间不断尝试这种最土的办法解决了遇到的所有问题。


六. 测试总结

目前我们的测试采用了手动测试的方式，主要针对以下几方面测试：

- 注册、登录的逻辑检测，包括用户名查重，邮箱查重以及是否格式正确；密码位数是否低于6位，各选项是否位空等
- 登录状态测试，包括登陆前跳转页面以及登陆后跳转页面是否显示正确的登录状态
- 前端页面跳转逻辑测试
- 前后端数据传输测试

对已经实现的功能进行测试，我们发现了一些小bug，比如，刷新页面数据丢失等。目前均已经得到解决

七. 系统部署



在Ubuntu16.04 LTS中，我们当前情况下准备使用NGINX部署, 如下：

```
sudo apt install nginx
```

```
service nginx restart
```

接着，我们要支持php文件解析：

```
sudo apt install php php-fpm
```

配置虚拟环境并运行：

```
sudo apt install python3-dev
```

```
sudo apt install python3-venv
```

```
python3 -m venv ./venv
```

```
Source ./venv/bin/activate
```

安装uwsgi：

```
pip install uwsgi
```

安装django：

```
pip install Django
```

接着，我们连接django到NGINX，运行uwsgi：

```
uwsgi -ini uwsgi.ini
```

启动nginx：

```
service nginx restart
```

但由于时间问题，在运行过程中遇到了尚未解决的问题，因此我们暂时采用直接runserver的方式

八. 个人文档博客地址

侯建国：<https://thssmonkey.github.io/2017/11/25/iKNOW项目中期个人总结/>

胡 哲：[https://thsshz.github.io/2017/11/26/iKNOW-middle/ - more](https://thsshz.github.io/2017/11/26/iKNOW-middle/- more)

