

SERWeb 酒井体育赛事报名系统

SERWeb 酒井体育赛事报名系统

- 1. 需求分析
 - 1.1 项目背景
 - 1.2 需求分析
 - 1.3 系统功能分析
- 2. 模块及接口设计
 - 2.1 项目结构
 - 2.1.1 Back-end 项目目录
 - 2.1.2 Back-end UML 图
 - 2.1 模块设计
 - 2.1.1 admin 模块
 - 2.2.2 models 模块
 - 2.2 接口设计
- 3. 数据库设计
- 4. 开发者须知
 - 4.1 搭建开发环境
 - 4.1.1 拉取开发环境
 - 4.1.2 运行开发环境
 - 4.1.2.1 安装依赖
 - 4.1.2.2 前端开发环境
 - 4.1.2.3 后端演示环境
- 5. 许可证信息

1. 需求分析

1.1 项目背景

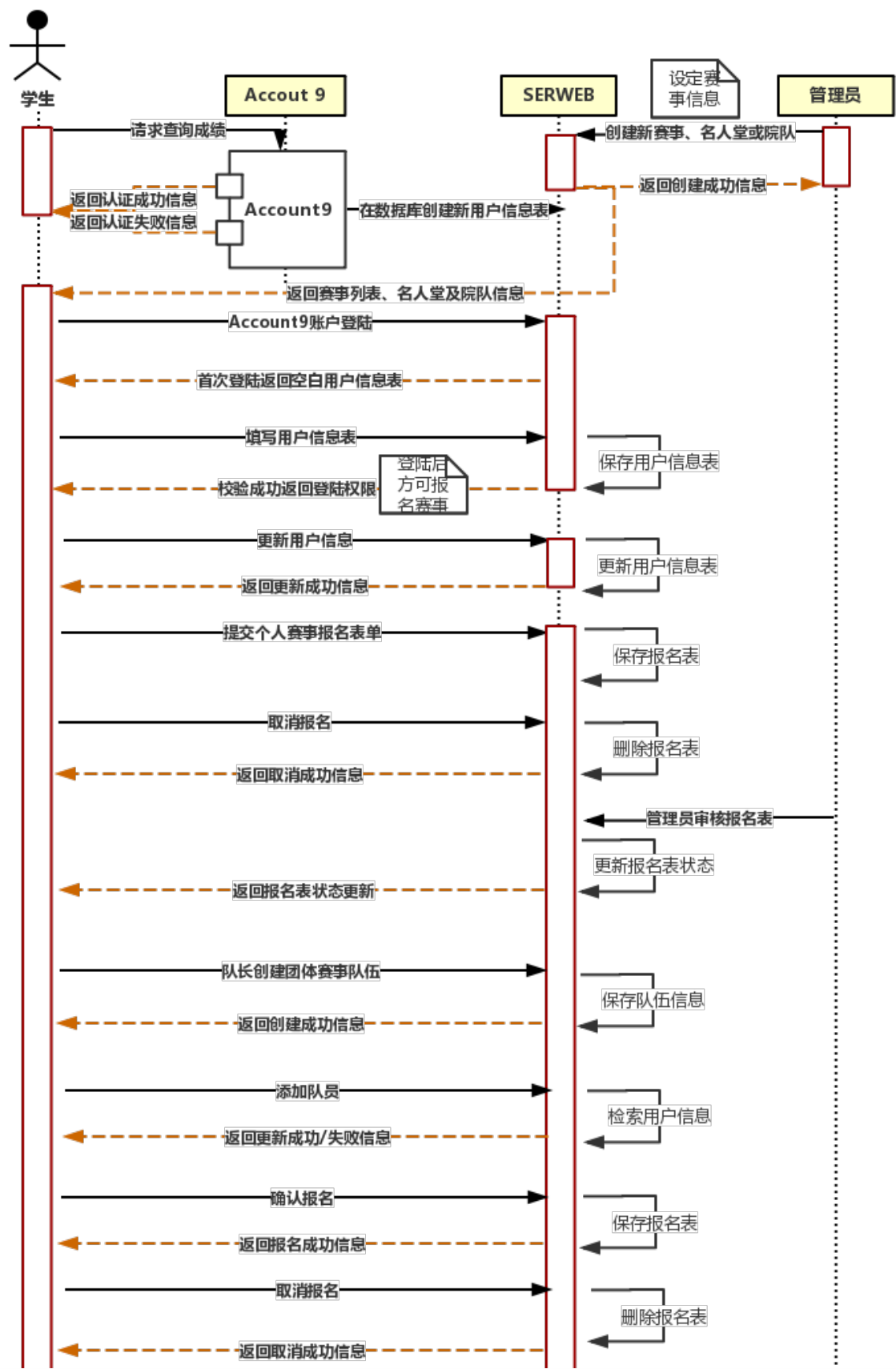
每年，清华大学举办的“马约翰杯”体育比赛以及清华大学计算机系内举办的“酒井杯”等体育赛事都需要赛前统计报名情况并给予报名人员反馈。目前，问卷星是常用的报名形式。但在很多比赛里，参赛人员需要填写的个人信息完全相同或有大部分重叠，造成不断重复相同工作的情况。同时，组织者给参赛人员回复邮件时，则需要手动添加报名人员邮箱，回复邮件，这项工作单一繁琐，耗费组织人员大量的时间和精力。有时，为了获得特定运动员的报名信息，可能需要大量的中间步骤。

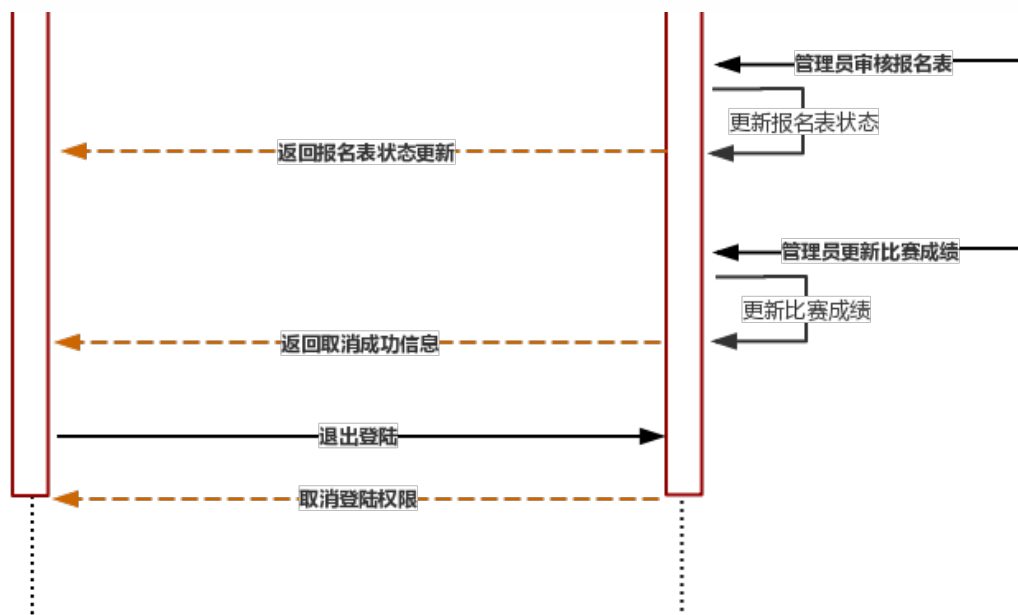
为了改善这一情况，简化报名步骤，提高赛事组织者的工作效率，清华大学计算机系学生会希望建立一个体育赛事报名管理系统，赛事组织者可以在此发布赛事信息及报名通道。同学们可以通过个人info 账号信息登录，在赛事报名界面点击报名，自动完成赛事报名。系统将自动收集报名人员的相关信息，并统一进行回复与管理。

该系统所实现的功能将不仅限于系内学生组织，同样适用于校内其他院系以及其他类似体育赛事的报名和管理等场景。

1.2 需求分析

1.3 系统功能分析





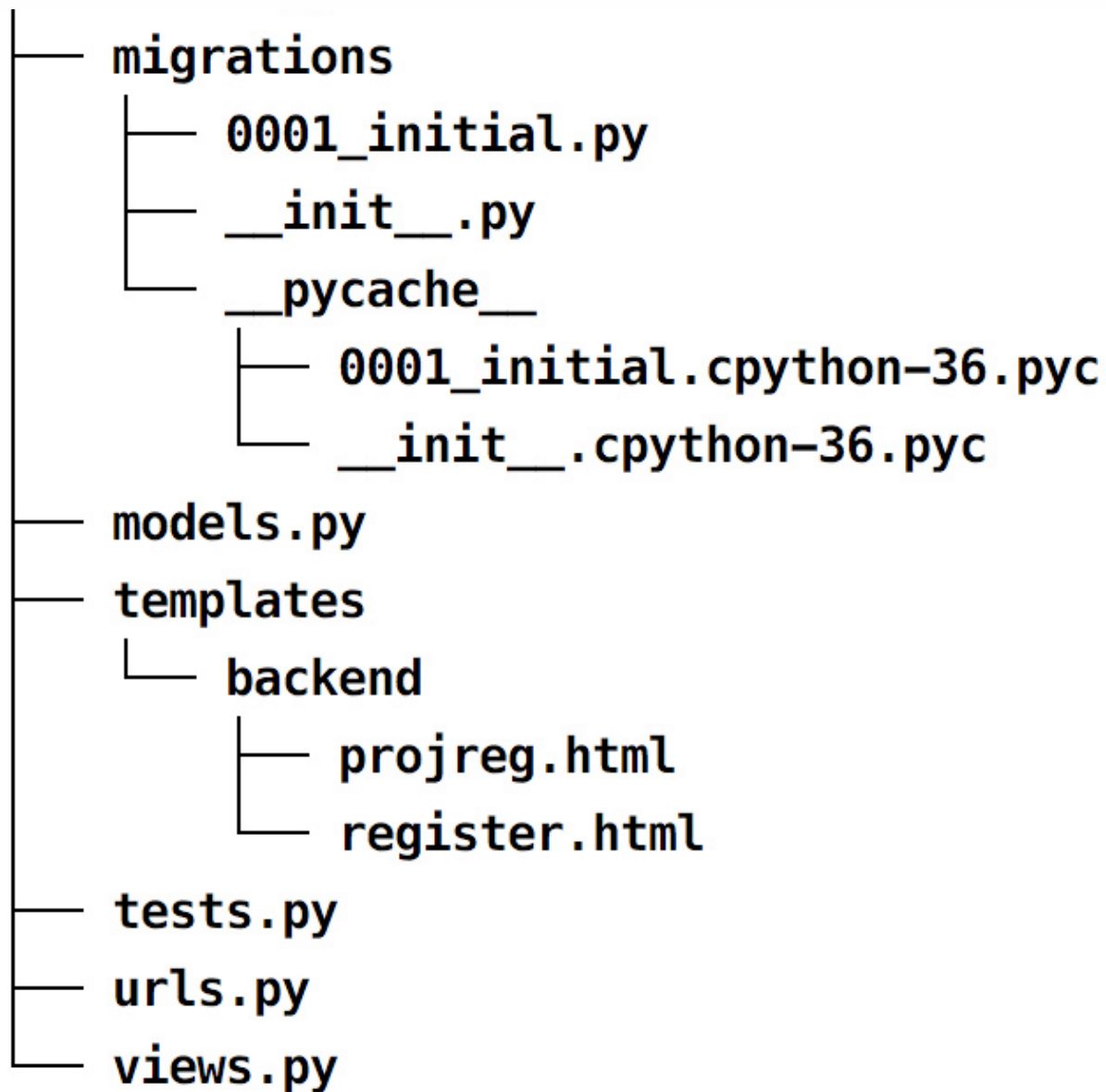
2. 模块及接口设计

2.1 项目结构

2.1.1 Back-end 项目目录

```

.
├── __init__.py
├── __pycache__
│   ├── __init__.cpython-36.pyc
│   ├── admin.cpython-36.pyc
│   ├── apps.cpython-36.pyc
│   ├── forms.cpython-36.pyc
│   ├── models.cpython-36.pyc
│   ├── urls.cpython-36.pyc
│   └── views.cpython-36.pyc
├── admin.py
├── apps.py
└── forms.py
  
```



5 directories, 21 files

2.1.2 Back-end UML 图



2.1 模块设计

2.1.1 admin 模块

如下图，admin 模块包含

- UserInfoAdmin
- ProjectRegisterRelationshipAdmin
- GroupAdmin
- MembershipAdmin

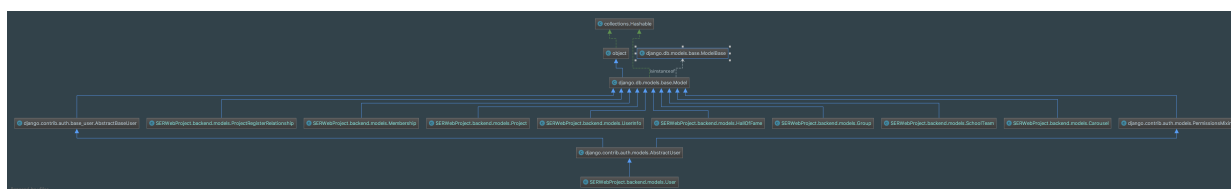
- CarouseAdmin
- HallOfFameAdmin
- SchoolTeamAdmin



2.2.2 models 模块

如下图，models 模块包含

- User
- UserInfo
- Project
- Project_Register_Relationship
- Group
- Membership
- Carousel
- HallOfFame
- SchoolTeam



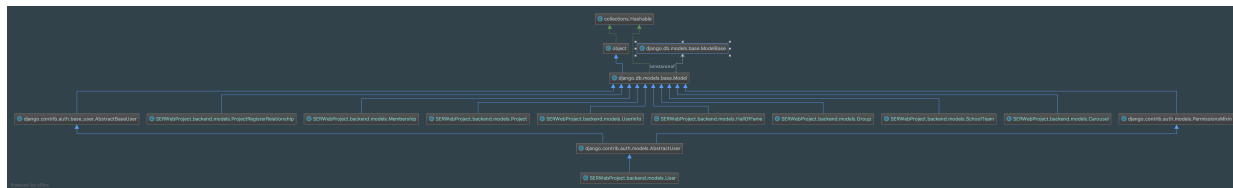
2.2 接口设计

接口列表如下，各接口文档请点击链接访问项目软工平台 Gitlab wiki

- register()
- login()
- [user_info_submit\(\)](#)
- [user_info_request\(\)](#)
- project_list_display()
- project_card_display()
- [project_info_request\(\)](#)
- [project_register\(\)](#)
- [project_register_relationship_request\(\)](#)
- project_grade_request()
- [add_group\(\)](#)
- [add_teammate\(\)](#)
- [set_teammate_confirm\(\)](#)
- [carousel_request\(\)](#)
- [hall_of_fame_request\(\)](#)
- [school_team_request\(\)](#)
- [group_project_register API 使用流程](#)

3. 数据库设计

考虑到项目规模和目标使用人群数目，项目采用 Django 内置的 SQLite3 作为数据库。数据库表和 Models 保持一致。



4. 开发者须知

4.1 搭建开发环境

4.1.1 拉取开发环境

- 在 gitlab 拉取 SERWeb 项目 master 分支文件即可，git 相关操作如果不熟悉可以参考 [廖雪峰 Git教程](#) 或者其他教程。
- 如果使用 JetBrains 系的 IDE，可以使用内置的 VCS 系统拉取代码，步骤参考 [JetBrains 文档](#)。

4.1.2 运行开发环境

4.1.2.1 安装依赖

运行开发环境依赖如下包：

- node.js
 - 安装方法[参考这里](#)
- python 3.5.2 or later
 - django 1.11.5 or later
 - django-cors-headers
 - 以上两个包均用 pip install

```
$ pip3 install django
$ pip3 install django-cors-headers
```

- Vue.js Cli

```
# 全局安装 vue-cli
$ npm install --global vue-cli
# 进入frontend文件夹，安装依赖
$ cd ../SERWeb/SERWebProject/frontend
$ npm install
$ npm run dev
```

4.1.2.2 前端开发环境

- 在 IDE 内置 terminal（或其他 terminal）内进入 SERWebProject/frontend 文件夹

- 输入命令 `npm install`
- 输入命令 `npm run dev`，如果成功即可自动弹出前端开发环境（默认端口8080）。
- 前端开发环境支持所见即所得，可立即反映页面改动。

```
$ cd SERwebProject/frontend
$ npm install
$ npm run dev
```

4.1.2.3 后端演示环境

- 在 IDE 内置 terminal（或其他 terminal）内进入 SERWebProject 文件夹
- 输入命令 `python manage.py runserver`
- 打开浏览器输入 `localhost:8000` 即可看到后端演示环境（默认端口8000）。
- 后端演示环境不支持所见即所得，如果要反映页面改动需要重新 build，因此日常开发建议使用前端开发环境。

```
# 启动后端演示环境
$ cd SERwebProject
$ python manage.py runserver

# 重新 build 后启动后端演示环境
$ cd /SERwebProject/frontend
$ npm run build
$ cd ..
$ python manage.py runserver
```

5. 许可证信息



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).