|  |
| --- |
| **C:\Users\hexixun0910\Desktop\logo_.png** |
| **需求与设计文档** |
|  |
| prj1\_whynotdance小组 |
| 计45班 王晨阳 何熙巽 田文龙 徐子南 |

目录

[一、项目基本情况 4](#_Toc472098930)

[1.1概述 4](#_Toc472098931)

[1.2团队成员与分工 4](#_Toc472098932)

[1.3开发环境 4](#_Toc472098933)

[1.4运行说明 4](#_Toc472098934)

[二、需求分析 6](#_Toc472098935)

[2.1概述 6](#_Toc472098936)

[2.2问卷系统设计 7](#_Toc472098937)

[2.3问卷数据的处理 7](#_Toc472098938)

[2.4用户系统管理 8](#_Toc472098939)

[三、模块及接口设计 9](#_Toc472098940)

[3.1概述 9](#_Toc472098941)

[3.2用户界面interface 9](#_Toc472098942)

[3.3前后端接口api 10](#_Toc472098943)

[3.4后端数据库database 11](#_Toc472098944)

[四、技术细节 13](#_Toc472098945)

[4.1概述 13](#_Toc472098946)

[4.2 Django部分 13](#_Toc472098947)

[4.3 HTML与CSS部分 23](#_Toc472098948)

[4.4 JavaScript部分 27](#_Toc472098949)

[五、总结 36](#_Toc472098950)

一、项目基本情况

1.1概述

本项目以清华大学为预期用户，作为校内信息化统计平台进行服务，建立网页端和移动端校内信息化统计平台，基于Project\_1的需求实现。

本项目能够满足校内学生团体的几类统计需求，如活动报名、实验室招募、多规则投票；对于收集到的数据可以进行一定的统计、分析，并且将分析结果进行可视化的展示，为提升校园工作的效率和数据的安全性供帮助。同时也为公司和校外团队提供服务。

1.2团队成员与分工

本项目的开发小组名为whyNotDance，组员共为4人，组长为王晨阳，具体分工如下：

|  |  |
| --- | --- |
| 成员名 | 分工 |
| 王晨阳 | 设计整体架构，负责前端页面的设计与实现以及前后端沟通的设计 |
| 何熙巽 | 实现整体架构，实现前端部分页面，实现前后端沟通及用户体验优化 |
| 田文龙 | 实现部分前端页面，实现部分前后端数据接口 |
| 徐子南 | 设计并实现数据库，并随着开发进程进行单元测试 |

表1: 小组成员分工表

1.3开发环境

本项目在前端及javascript部分使用了Bootstrap 3.3.5及jQuery2.1.4等框架，在后端则使用了Python 2.7.10 + Django 1.8.4的框架，并使用了Django框架中自带的SQLite数据库。

1.4运行说明

为了运行本项目，首先需要确保运行的环境已经安装好Python2.7以及Django1.8.4。

首先需要从本项目的git地址获取本项目的全部源代码，这一步需要确保运行环境已经安装了git。在命令行中使用如下命令可以获取到上述源代码:

$ git clone ssh://git@166.111.131.70:8000/2014011407/prj1\_whynotdance.git

若需要在本地运行本项目，进入源码所在目录的src文件夹，执行如下命令：

$ python manage.py runserver 0.0.0.0 8080

并在浏览器中访问<http://127.0.0.1:8080/> 便可看到登陆界面

本项目已经先期部署在了课程所给的服务器资源上，访问<http://52.187.34.62:8080>/ 可以看到已经部署好的项目的登陆界面，若需要重新部署，请查阅部署手册。

二、需求分析

2.1概述

本项目的需求分析基于项目开始前需求方提供的文档《PRJ1-校园信息化统计平台》进行。在对这份进行分析后，我们将所有的需求分为三大块:

1、问卷系统设计

该部分为本项目最主要的功能，具体表现为活动的发起方可以在线编辑问卷并发布，活动的参与方可以根据发起方发布的有平台生成的信息访问该问卷并进行填写，填写的信息需记录在平台的后台备用。问卷的设计发布功能需满足多种要求。

2、问卷数据的处理

该部分为本项目的重要功能之一，对问卷平台的活动发布者开放，根据问卷系统中所收集备用的信息处理，并向用户开放数据导出、分析与统计的端口，包括各种可视化实现。

1. 用户系统管理

设置管理员权限用户，使用内置账户登录，可以管理其它普通用户和普通用户发布的问卷，同时需要实现普通用户的个人用户界面。

值得注意的是，由于需求中包括普通要求，提高要求，终极要求等多个等级，而部分提高要求在开始是不在实现计划的范围内，在最后也并未实现，下述的详细需求分析中删去了这部分提高要求的内容，并在本组与需求方讨论后添加了用于优化用户体验的内容。

2.2问卷系统设计

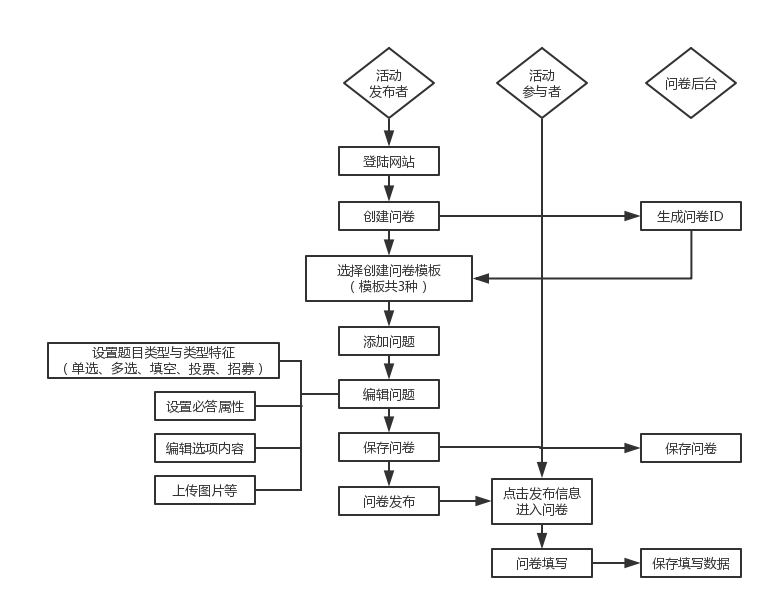


图1: 问卷系统泳道图

其中设置的三种文件模板分别对应需求中提及的报名/信息统计表、实验室对象招募设计以及多规则投票平台。题目的类型特征对应题目类型，包括多选题的选项数，填空题的内容类型（数字，电话号码，邮箱等），若使用投票平台等模板，则此处的问卷问题变为投票的项目。问卷的发布方式为导出链接与二维码，问卷的传播由发布者利用社交工具进行。

2.3问卷数据的处理

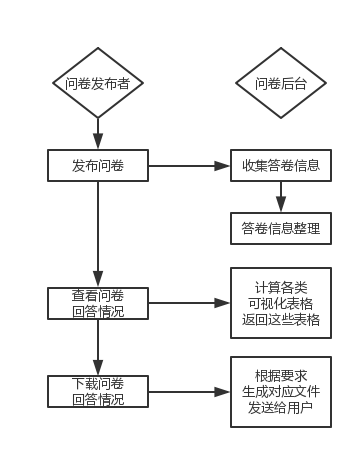


图2: 问卷数据处理泳道图

其中可视化表格包括柱状图、饼状图、折线图等，内容包括平均值、方差等简单的计算与简单的数据总量统计。需收集的答卷情况除了答卷本身外，还包括填写者的IP等信息，导出的excel/csv格式数据中包括以上内容。

2.4用户系统管理

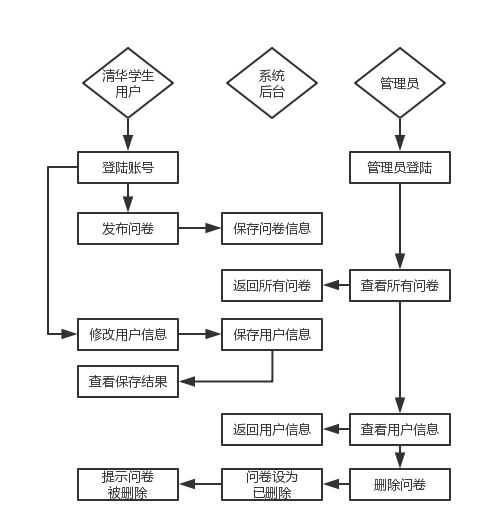


图3: 用户系统管理泳道图

该部分要求主要来自与需求方的沟通，管理员由内置账号实现，而清华学生用户由清华账号登陆。管理员主要功能为查看与删除问卷，查看用户信息与发布公告。

三、模块及接口设计

3.1概述

本项目的实现主要分为四个模块，由于本项目由Python2.7.10加上Django1.8.4实现，这四个模块体现为四个app，分别为dance、interface、api以及database。

interface、database以及api三个模块分别负责前端用户界面，数据库，以及连接前端用户界面与数据库，将在本章的其余部分做详细说明。

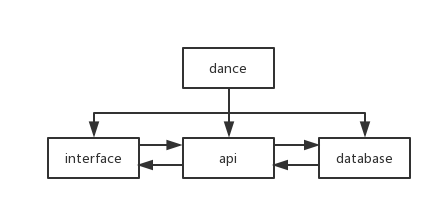


图4: 模块拓扑关系图

顶层模块dance为Django项目创建时自带的app，主要负责记录Django框架的基本设置，以及保存其余三个模块所通用的URL模板。由于本项目Django设置主要为将其余三个模块app加入顶层模块，设置语言，时区、设置SMTP服务器等，在这一章中不做更多的说明，其具体的设置见技术细节部分。

3.2用户界面interface

interface模块的接口主要在interface文件夹下的views.py中实现，该模块直接与用户进行交互，主要实现的接口功能为响应URL请求并对模板进行渲染后返回，下表以普通用户的权限可以访问的URL接口为例展示接口功能，管理员与游客身份问卷发布者的接口与之类似：

|  |  |
| --- | --- |
| 接口名 | 接口功能 |
| legalUser | 用户登录时调用，返回用户主界面 |
| legalUser\_dashboard | 用户主界面中右边展示的信息部件 |
| legalUser\_show\_app\_list | 查看所有自己发布的问卷时调用，返回一个可排序表格部件 |
| legaUser\_design | 用户进入编辑问卷界面时调用，返回一个问卷模板 |
| legalUser\_design\_question | 用户添加或编辑题目后调用，即时显示题目信息 |
| show\_model | 在多种场合调用，弹出一个阻塞式窗口，以编辑或选择 |
| user\_information | 用户查看用户信息时调用，返回用户信息页面 |
| user\_information\_change | 用户修改用户信息时调用，返回修改信息页面 |
| show\_statistics\_choose | 用户点击数据统计页面时调用，返回该页面有部的各种部件，如统计图表，用以选择需要查看统计信息的问卷与问卷中题目的下拉菜单，填写者信息表格等内容 |
| show\_statsitics |
| show\_charts |
| statistics\_question |
| statistics\_question\_list |
| questionnaire\_publish | 用户发布问卷确认后调用，展示问卷连接与二维码 |
| log\_off | 用户注销接口，返回登陆页面 |

表2：interface模块中普通用户页面的接口名称与功能表

除此之外还定义了一些问卷参与者填写问卷时调用的接口，以及模块内部互相调用的一些函数，如render\_ajax与render\_sortable，封装了返回一个部件或可排序表格的过程，以及check\_indentity和get\_username等检查session中信息以实现权限控制的内容。

3.3前后端接口api

api模块位于前端与数据库之间，负责将前端的数据传给数据库，并将数据库的处理结果返回给前端。考虑到实现效率和用户体验问题，发送邮件，请求登录等耗时操作也在此模块中进行，并在这个过程中调用database的接口，api中具体的接口名称与功能如下表，这些接口主要位于api文件夹下的views.py文件中，其中删去了一些内部调用的函数：

|  |  |
| --- | --- |
| 接口名 | 接口功能 |
| modify\_name | 修改问卷名称，并将修改后内容存入数据库 |
| create\_new\_act | 创建新问卷，并返回数据库提供的问卷id |
| operation\_qst | 修改问卷题目顺序，并返回数据库提供的修改结果 |
| modify\_qst | 修改问卷题目内容，并返回数据库提供的修改结果 |
| remove\_act | 将问卷状态设置为已删除，并返回数据库提供的修改结果 |
| save\_act | 保存问卷，并返回数据库提供的修改结果 |
| publish\_act | 发布问卷，并返回数据库提供的修改结果 |
| email\_act | 发送邮件，并返回邮件发送结果 |
| info\_change\_act | 提交用户信息修改，并返回数据库提供的修改结果 |
| login\_act | 提交用户登陆请求，调用清华账号api并返回登陆结果 |
| get\_user\_information\_act | 返回用户信息 |
| get\_all\_user | 返回所有用户与其信息的列表 |
| get\_all\_questionnaire | 返回所有问卷与其信息的列表 |
| get\_questionnaire\_byID | 根据问卷ID或者问卷状态，获取符合条件的问卷 |
| get\_questionnaire\_byST |
| get\_participants | 获取问卷参与者列表 |
| get\_result\_of\_question | 获取问卷结果与问卷结果的统计信息 |
| get\_statistics\_of\_question |
| notic\_act\get\_notice | 管理员修改系统公告\获取系统公告内容 |
| questionnaire\_submit | 问卷提交，将提交内容存入数据库，并返回处理信息 |
| stop\_act\resume\_act | 停止问卷收集与重新开始问卷收集的接口，返回处理结果 |
| get\_chart\_json | 从数据库中获取问卷统计信息并制作各种图表返回 |

表3：api模块的接口名称与接口功能表

3.4后端数据库database

database模块即我们的数据库，其接口为api所调用，主要返回数据库中所存放的信息，这些接口与api接口即功能表中提到的需要访问数据库的接口基本对应，这些接口主要位于database文件夹下的api.py文件中，具体如下。

|  |  |
| --- | --- |
| 接口名 | 接口功能 |
| createUser | 创建一个新的用户 |
| userInfoChange | 修改数据库中保存的用户信息，返回修改结果 |
| 接口名 | 接口功能 |
| getUserInfo | 在数据库中查找用户信息并返回 |
| getAllUser | 查找数据库，并返回所有用户及其信息的列表 |
| getAllQuestionnaireInfo | 查找数据库，并返回所有问卷及其信息的列表 |
| getQuestionarieByStatus | 查找数据库，根据问卷状态或者信息，返回一个问卷或多个问卷所组成的列表 |
| getQuestionarieByID |
| createNewQuestionnaire | 创建新问卷，返回问卷ID |
| saveQuestionnaireInfo | 保存问卷信息，返回处理结果 |
| createNewQuestion | 创建新问题，返回问题ID |
| operateQuestion | 调整问题的顺序，返回处理结果 |
| deleteQuestionnaire | 删除问卷，返回处理结果 |
| saveQuestionnaire | 保存问卷，返回处理结果 |
| publishQuestionnaire | 发布问卷，返回处理结果 |
| stop\wakeQuestionnaire | 停止或重新开始问卷收集，返回处理结果 |
| modifyQuestion | 调整问题内容并保存，返回处理结果 |
| fillQuestionnaire | 保存填写问卷信息，返回处理结果 |
| getFillers | 在数据库中查找问卷填写者信息并返回 |
| getQuestionFill | 在数据库中查找所有填写过的问卷并返回 |
| getStatisticsOfQuestion | 在数据库中查找问卷填写情况信息并返回 |

表4：database模块的接口名与功能表

该表中删去了部分database模块内部调用的接口，这些接口主要负责将数据库中取出的信息制成dict类型，并且将多个这种dict制成一个list。还有部分接口负责进行测试，这些接口也没有在这张接口功能表中被提及。

在api模块与database模块的功能表中多次提到“返回数据库提供的处理结果”以及“返回处理结果”等表述，这里返回的处理结果为一个变量，表示处理流程是否成功完成，用以避免数据库出错或者从interface传入api模块的数据出错所导致的问题。

四、技术细节

4.1概述

本项目的实现主要包括三种类型的代码，分别是作为网站框架的Django+Python代码，前端网页的HTML文件与CSS样式表，以及前端的JavaScript脚本语言。本章将对这三种类型的代码分别展开，介绍本项目的一些重要的技术细节。

本项目所实现的网站的拓扑结构如下图所示，每个模块基本由HTML+CSS加上JavaScript实现，模块间的跳转与准入权限控制等由Django+Python实现，模块间的和模块与后台的数据沟通由JavaScript和Django+Python实现。

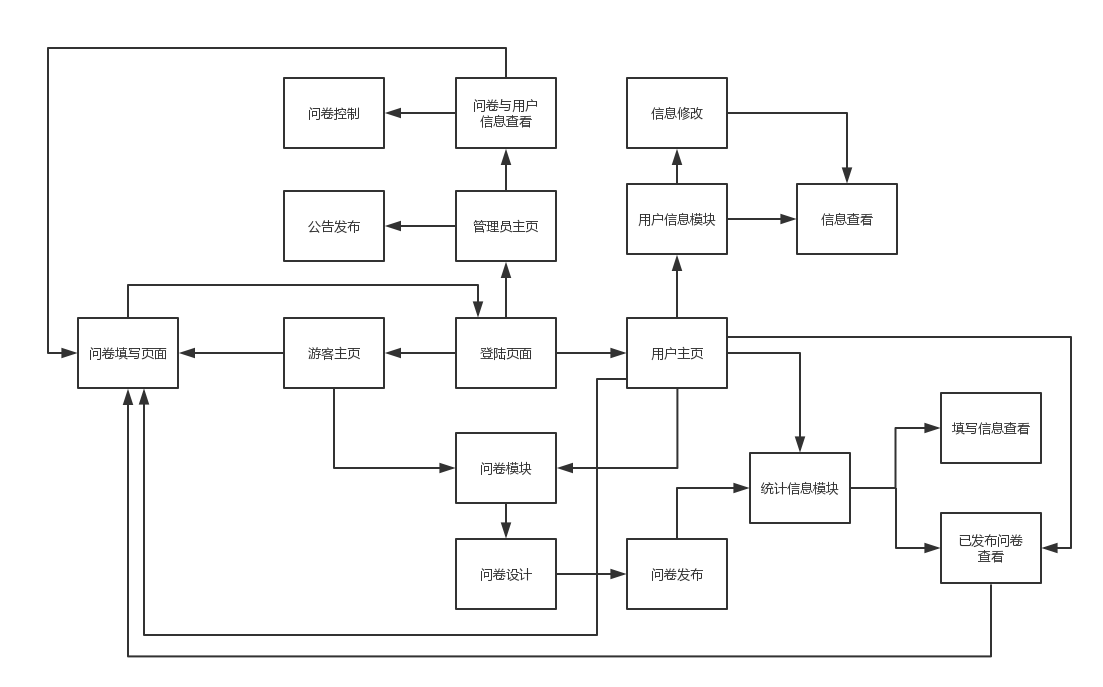


图5：清华大学信息化统计平台网站拓扑结构图

4.2 Django部分

**4.2.1 Django渲染函数**

interface文件夹中的views.py文件包含了网站所有页面对应的Django渲染函数，具体的接口名称和功能请见模块与接口设计一章的用户界面interface一节，此处不再赘述，这一节里将重点介绍在前面没有提及的封装的两个渲染函数render\_ajax与render\_sortable，分别进行Ajax请求的处理以及可排序表格的渲染。

在概述一节中提出的网页拓扑结构图中的链接跳转绝大部分是由Ajax请求完成的，通过Ajax请求载入页面时，在载入网页时不需要刷新整个页面，只需要返回一个部件即可，而当用户进行刷新等操作，或者直接通过URL访问网页时，需要刷新整个页面。render\_ajax函数可以解决此类矛盾，通过判断请求是否为Ajax请求，若是，则通过JavaScript实现仅渲染一个部件，若不是，则渲染整个网页，因此对于每个页面存在两种模板文件。

render\_ajax函数的调用方法类似render函数，其接口如下：

**render\_ajax**(request，url，params，item\_id = ‘ ’)

|  |  |
| --- | --- |
| 参数名 | 功能 |
| request | Django的Request对象 |
| url | 要渲染的HTML模板文件路径 |
| params | 模板中的各种参数 |
| item\_id | 页面对应的左边栏项目ID |

表5：render\_ajax函数的参数表

可排序表格为本项目中一个经常使用的特殊部件，例如用户查看所有自己发布的文件时会需要渲染可排序表格，关于该表格的具体技术细节见HTML与CSS部分。本项目封装了用于渲染可排序表格的函数render\_sortable，其接口如下：

**render\_sortable**(request, items, url, params)

|  |  |
| --- | --- |
| 参数名 | 功能 |
| request | Django的Request对象 |
| item | 要渲染的表格中的项目列表 |
| params | 模板中的各种参数 |
| url | 要渲染的HTML文件路径 |

表6：render\_sortable函数的参数表

可排序表格支持排序顺序的自定义，包括升序和降序，排序内容的筛选，以及关键词搜索。这些参数将会以GET请求从网页中事实获得，并对表格进行刷新。可排序表格中每一个分页的分页信息，即参数调用同一文件内的get\_pagination函数获得，其接口如下：

**get\_pagination**(item\_total, item\_per\_page, cur)

|  |  |
| --- | --- |
| 参数名 | 功能 |
| item\_total | 表格内容的总项目数 |
| item\_per\_page | 表格每页显示的项目数 |
| cur | 当前页码 |

表7: get\_pagination函数的参数表

该函数根据这些信息打包为一个list，作为params传入render\_sortable函数中

**4.2.2 Django网页模板**

本项目通过Django模板的继承避免了大量的代码重复，下面介绍主要使用到的模板，以及这些模板对应的网页的结构与作用，模板的具体实现将在HTML与CSS部分介绍。

这些模板文件均位于interface模块下的templates文件夹中。publish\_base.html与base.html为最主要的两个模板，分别对应问卷填写者，以及问卷发布者，包括清华学生用户，游客，管理员的主界面模板。前者由后者删除user-menu模块与left-column模块后得到，其余部分与base.html基本一致，因此这里仅详细介绍base.html。

base.html模板为大部分其它模板所继承，共包含8个block，如下表所示：

|  |  |
| --- | --- |
| 名称 | 描述 |
| title | 页面的标题 |
| css | 自定义的css样式表 |
| script | 自定义的script |
| subtitle | 导航栏上的副标题 |
| user-menu | 导航栏右侧用户名标签的下拉菜单 |
| main-frame | 左侧边栏+主页面的整个框架，通常不修改 |
| left-column | 左侧边栏 |
| main-page | 主页面 |

表8: base.html模板的block说明表

由于base.html文件为问卷发布者用户界面的模板，服务于普通清华账号用户，管理员以及游客，本项目对于者三种身份的用户在base.html模板的基础上定制了独特的用户界面。这些定制的用户界面模板均命名为index.html，存放于templates文件夹下的legalUser、manager以及guest等子文件夹中。

又由于其中管理员以及游客界面所使用的用户界面是在普通用户的界面中删去某些部分，或是稍作修改而成，这里首先以普通用户的legalUser文件夹下的模板文件为例介绍用户界面的结构。

legalUser文件夹根目录的情况如下表所示：

|  |  |
| --- | --- |
| 路径 | 描述 |
| dashboard.html | 用户首页模板，包括各种概览信息页面 |
| dashboard.ajax.html | 用户首页Ajax请求加载时的模板 |
| applications | 文件夹，用以存放可排序表格的渲染文件 |
| design | 文件夹，存放各种阻塞式窗口、问卷问题的模板  以及用户问卷页面的模板与ajax模板 |
| information | 文件夹，存放用户信息查看与修改界面的模板与ajax模板 |
| login | 文件夹，存放登陆信息界面 |
| statistic | 文件夹，存放数据统计界面的各种模板，如图表等 |

表9：legalUser文件夹内容表

* 其中可排序表格的渲染模板文件如下表所示：

|  |  |
| --- | --- |
| 文件名 | 功能 |
| applications.ajax.html | 可排序表格页面Ajax请求加载时的模板 |
| applications.html | 可排序表格页面模板 |
| applications\_content.html | 可排序表格表格体模板 |
| applications\_list.html | 可排序表格表头内容模板 |

表10：可排序表格渲染模板文件表

阻塞式窗口是为了让用户优先响应某些事件而设立的，如修改题目时的设置表，简单的确认框等，将会在HTML与CSS部分详细介绍，其模板类型如下表所示：

|  |  |
| --- | --- |
| 文件名 | 功能 |
| email\_modal.html | 填写反馈邮箱弹窗模板 |
| fillin\_modal.html | 填空题设置弹窗模板 |
| information\_modal.html | 用户信息查看弹窗模板 |
| mark\_modal.html | 实验室招募题设置弹窗模板 |
| mutl\_modal.html | 多选题设置弹窗模板 |
| publish\_modal.html | 问卷发布成功信息展示弹窗模板 |
| qst\_name\_modal.html | 问卷名称设置弹窗模板 |
| single\_modal.html | 单选题设置弹窗模板 |
| vote\_modal.html | 投票提设置弹窗模板 |

表11：阻塞式窗口渲染模板文件表

问卷问题模板用于在用户编辑时即时显示问题的样式，也用户在问卷填写者填写问卷时显示问题的样式，其模板类型如下表所示：

|  |  |
| --- | --- |
| 文件名 | 功能 |
| fillinl.html | 填空题显示模板 |
| mark\_modal.html | 实验室招募题显示模板 |
| mutl\_modal.html | 多选题显示模板 |
| single\_modal.html | 单选题显示模板 |
| vote\_modal.html | 投票题显示模板 |

表12：问卷问题模板文件表

除了已经提到的dashboard与application两个栏目以外，用户界面还设置有其它数个栏目，这些栏目的模板均由一个html文件和一个ajax.html文件组成，应对不同请求：

|  |  |
| --- | --- |
| 栏目名 | 功能 |
| dashboard | 用户主页 |
| application | 显示可排序表格 |
| user\_information | 查看用户信息 |
| user\_information\_change | 修改用户信息 |
| design | 编辑与发布问卷 |
| statistics | 数据统计查看 |
| notice | 管理员界面专有，发布公告 |

表13：用户页面界面种类表

除了用户界面外，还有继承了publish\_base.html模板的填写问卷界面模板，这些模板文件位于templates目录下的questionnaire文件夹中，该文件夹目录如下表：

|  |  |
| --- | --- |
| 路径 | 描述 |
| publish\_qst | 文件夹，存有填写问卷时题目显示模板文件 |
| err\_visit.html | 用户访问的问卷不存在时的错误页面 |
| index.html | 继承自publish\_base.html的页面 |
| questionnaire.html | 继承自index.html的问卷背景页面 |
| questionnaire\_list.html | 问卷上显示的问题列表部件 |
| questionnaire\_success.html | 问卷提交成功的提示页面 |

表14：questionnaire文件夹目录表

其中publish\_qst中的模板与上面提到的显示模板基本一致，但取消了问题操作菜单。

**4.2.3 Django 数据模型**

使用了Django ORM进行数据库的构建与管理。以下列出所使用的数据模型。

**Admin**

每条记录表示一名管理员。

|  |  |  |
| --- | --- | --- |
| 域名 | 描述 | 类型与限制 |
| username | 用户名。 | 字符串，最大长度20 |
| description | 附加说明，通常填写真实姓名。 | 字符串，最大长度100 |
| password | 密码。 | 字符串，最大长度32 |
| email | 电子邮箱。 | 字符串，最大长度254 |

**User**

每条记录代表一名用户。

|  |  |  |
| --- | --- | --- |
| 域名 | 描述 | 类型与限制 |
| student\_id | 学生号码。 | 字符串，最大长度20 |
| username | 用户名。 | 字符串，最大长度20 |
| real\_name | 真实姓名。 | 字符串，最大长度20 |
| identity | 身份，大部分用户为legaluser。 | 字符串，最大长度20 |
| password | 密码，鉴于已使用清华账号登录暂时弃之不用。 | 字符串，最大长度32 |
| age | 年龄，用户自行填写。 | 字符串，最大长度18 |
| gender | 性别。 | 字符串，最大长度10 |
| status | 状态，类似签名的记录。 | 字符串，最大长度400 |
| address | 地址，用户自行填写。 | 字符串，最大长度400 |
| tel | 电话，用户自行填写。 | 字符串，最大长度20 |
| email | 电子邮箱，用户自行填写。 | 字符串，最大长度254 |

**Questionaire**

每条记录代表一张问卷。

|  |  |  |
| --- | --- | --- |
| 域名 | 描述 | 类型与限制 |
| questionaire\_user | 问卷所属用户。 | User外键 |
| questionaire\_title | 问卷标题。 | 字符串，最大长度30 |
| questionaire\_introduction | 问卷介绍。 | 文本 |
| questionaire\_status | 问卷状态，详情见下。 | 字符串 |
| questionaire\_type | 问卷类型，详情见下。 | 字符串 |
| questionaire\_time | 问卷创建时间。 | 字符串，最大长度50 |
| questionaire\_ip | 问卷创建者ip。 | ip |
| questionaire\_numOfQues | 问题数量。 | 整数 |
| questionaire\_numOfFilled | 已填写次数。 | 整数 |
| questionaire\_haveMaxTime | 是否拥有最大填写次数。 | 布尔 |
| questionaire\_maxTime | 最大填写次数。 | 整数 |
| questionaire\_md5 | 问卷唯一标识，用创建时间+创建者id hash得到。 | 字符串，最大长度32 |

questionaire\_status域可能的取值与含义如下：

|  |  |
| --- | --- |
| 值 | 描述 |
| IN | 新建问卷 |
| SA | 已保存问卷 |
| LA | 已发布问卷 |
| PA | 已暂停问卷 |

questionaire\_type域可能的取值与含义如下：

|  |  |
| --- | --- |
| 值 | 描述 |
| VO | 投票 |
| LW | 实验室招募 |
| SU | 报名 |

**Question**

每条记录代表一个问题。

|  |  |  |
| --- | --- | --- |
| 域名 | 描述 | 类型与限制 |
| questionaire\_id | 问题所属问卷，名字为历史遗留问题 | Questionaire外键 |
| question\_text | 问题文本内容 | 文本 |
| question\_type | 问题类型，详情见下 | 字符串 |
| question\_order | 问题在问卷中的序号 | 整数 |
| question\_choices | 选项数，对选择题有意义 | 整数 |
| question\_time | 问题创建时间 | 字符串，最大长度20 |
| question\_fillinrow | 可以填写的行数，对填空题有意义 | 整数 |
| question\_fillinhint | 提示文字，对填空题有意义 | 字符串，最大长度200 |
| question\_fillincheck | 填写内容的检查，对填空题有意义 | 字符串，最大长度200 |
| question\_mustfill | 是否必须填写 | 布尔 |
| question\_minfill | 最少选择的项数，对选择题有意义 | 整数 |
| question\_maxfill | 最少选择的项数，对选择题有意义 | 整数 |
| question\_displayVotes | 是否显示投票结果，对投票题有意义 | 布尔 |
| question\_ipTimes | 每ip的填写上限，null则为无上限 | 整数 |
| question\_dayTimes | 每天的填写上限，null则为无上限 | 整数 |

question\_type域可能的取值与含义如下：

|  |  |
| --- | --- |
| 值 | 描述 |
| SI | 单选题 |
| MU | 多选题 |
| FI | 填空题 |
| VO | 投票题 |
| MA | 打分题 |
| SO | 排序题 |

**Choice**

每条记录代表一个选项。

|  |  |  |
| --- | --- | --- |
| 域名 | 描述 | 类型与限制 |
| question | 选项所属问题 | Question外键 |
| choice\_text | 选项文本内容 | 字符串，最大长度200 |
| choice\_order | 选项在问题中的序号 | 整数 |
| choice\_limit | 最大选择次数，null则为无上限 | 整数 |
| choice\_hasPicture | 是否包含图片 | 布尔 |
| votes | 已选择次数 | 整数 |

**Filler**

每条记录代表一次填写。

|  |  |  |
| --- | --- | --- |
| 域名 | 描述 | 类型与限制 |
| filler\_ip | 填写者ip | 字符串，最大长度30 |
| filler\_time | 填写时间，与ip共同确定一个填写 | 字符串，最大长度30 |
| filler\_address | 填写者地址 | 字符串，最大长度50 |
| filler\_questionaire | 填写所属问卷 | Questionaire外键 |

**Answer**

每条记录代表一个回答。

|  |  |  |
| --- | --- | --- |
| 域名 | 描述 | 类型与限制 |
| answer\_filler | 回答所属填写 | Filler外键 |
| answer\_question | 回答所属问题 | Question外键 |
| answer\_content | 回答内容 | 文本 |
| answer\_choice | 回答的选项，仅选择题有意义 | Choice外键 |

4.3 HTML与CSS部分

**4.3.1概况**

在前端设计部分，本项目使用了<https://github.com/imSoNervousAh/anervouswebsite>这一开源框架，经过发布者允许后利用了其中的部分html文件模板，经过优化和细节方面的修改，应用于本项目。如上节中提到的base.html文件，这些html文件模板主要用于用户界面。并复用了该项目中使用的多种运行库。同时本项目借鉴了上述开源项目的部分JS代码，并根据自身需求加以修改和优化，这一部分将在下一节JavaScript代码部分重点强调。

本项目用户界面以外的部分由小组成员设计并实现，与使用了该框架的用户界面部分风格有所不用，但是配色等整体印象大体一致，例如登陆界面以及问卷提交成功界面，这些界面基本上由静态HTML文件实现，并没有使用Bootstrap等框架。

**4.3.2主页面情况**

本项目的主页面如下图：

****

图6：清华大学信息化统计平台主页面

页面最顶端为导航栏，左侧为侧边栏，右侧为主页面，对应上一节提到的base.html的block，在legalUser/index.html中具体实现。主页面中每一个方框在html标签中为一个div容器，通过Bootstrap中的col-md标签来控制宽度并自适应高度。

单击左边栏中的按钮会发送Ajax请求，动态的加载对应的主页面部件，如下图所示：



图7：清华大学信息化统计平台问卷设计界面

其它用户界面的风格与实现与之类似，而上文提到的用户登陆界面等其它一些界面的风格与之有所不同，下面以该界面为例，其html与css代码位于login\_page.html文件：

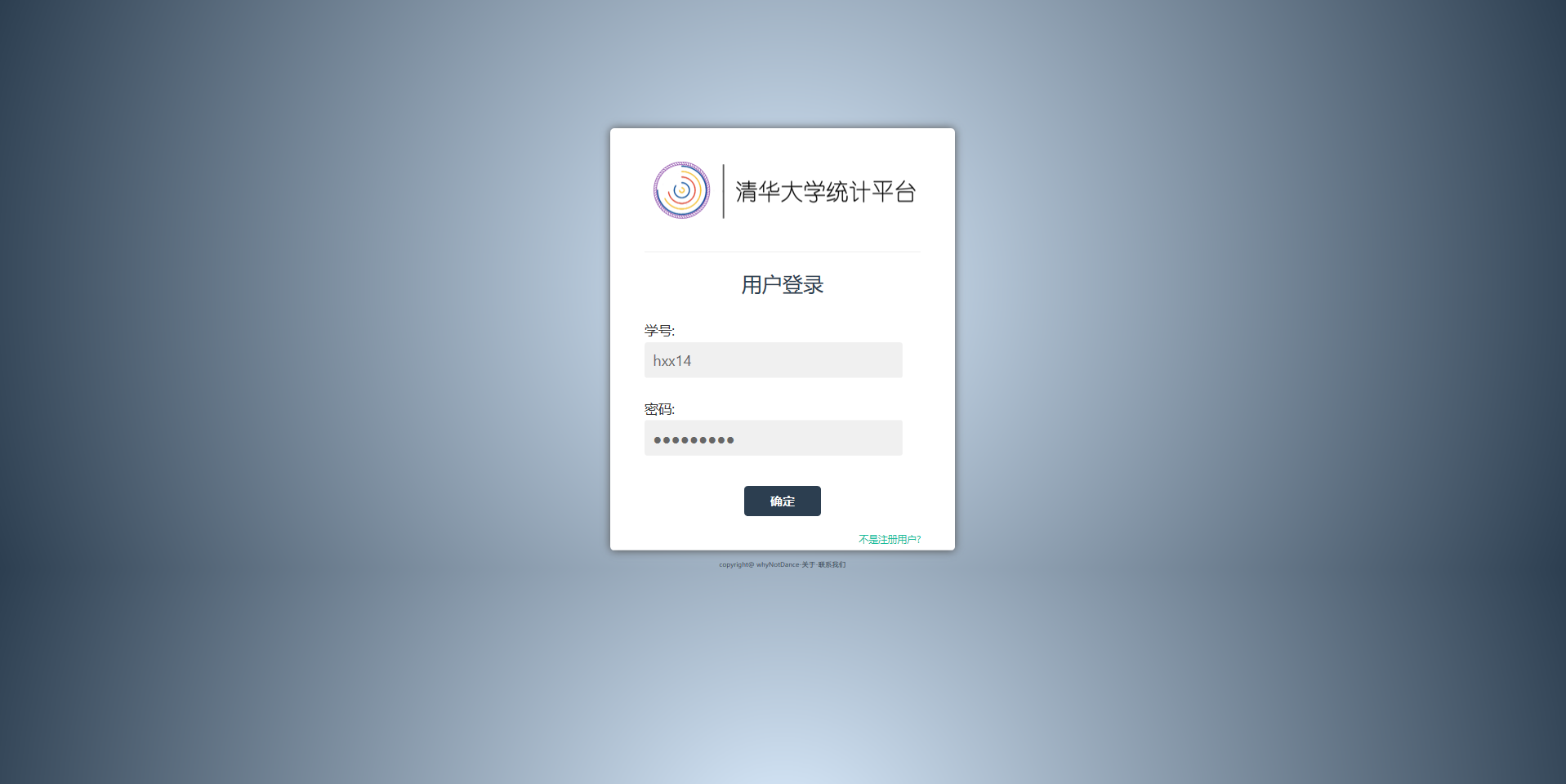


图8：清华大学信息化统计平台用户登陆界面

该页面中使用了表单form，图像img和容器div，表单项目input等html标签，css部分主要设置了部件颜色，间距，位置，大小等信息，并根据不同宽度屏幕设置了自适应宽度。

本项目中多次，多处使用了html的form表单标签，后端对表单的处理方式为通过URL请求调用interface模块中的接口，这些接口的具体定义见第三章内容，网页端同时使用JS对表单的提交进行处理，详见本章的JavaScript一节。

**4.3.3阻塞式窗口**

阻塞式窗口的设置可以让用户优先响应某些事件，本项目中的阻塞式窗口有两种类型，一种是包含复杂的HTML窗口，具体种类如表11所示，如图9，另一种是简单的确认框。

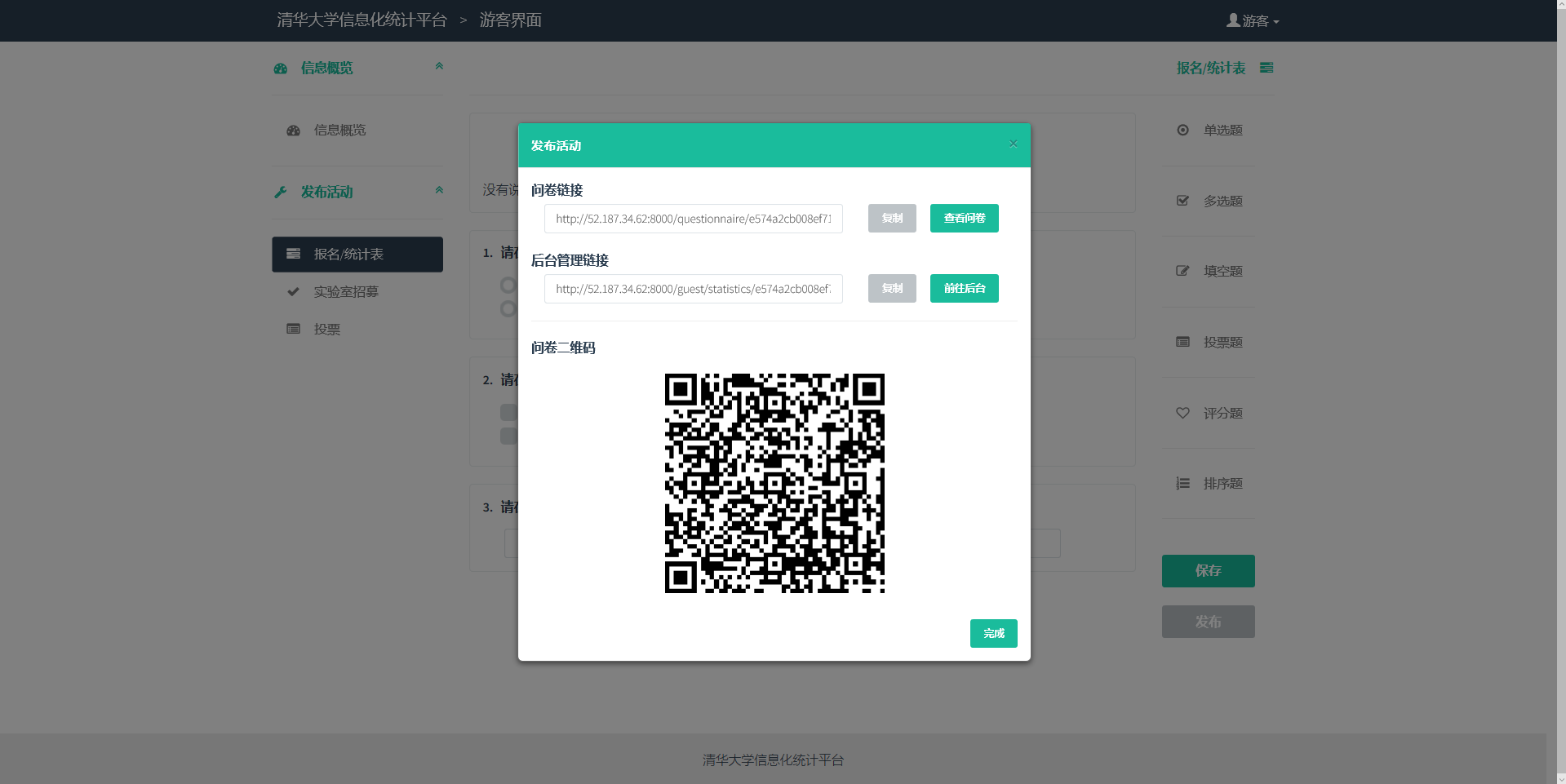


图9：复杂HTML阻塞式窗口



图10：简单的确认框

**4.3.4移动端优化**

本项目对移动端进行了部分优化，这些优化继承于概述中提到的开源项目，并且针对用户体验进行了大幅度的优化，使得手机端的操作更加人性化，加载更加迅速。

|  |  |
| --- | --- |
|  |  |

图11：移动端访问本网站的效果

移动端优化如隐藏侧边栏为左侧的滑动窗口，一些长度过长的表格进行分页等。移动端的布局基于Bootstrap响应式布局，当页面宽度低于一个固定值768px时会自动切换。

**4.3.5可排序表格**

可排序表格为本项目的一个特点，继承于概述中提到的开源项目，并针对本项目的特点进行了优化和重新设计，并且修复了一些运行中的错误，具体效果如下图所示：



图12：可排序表格

整个表格外部被一个div所容纳，搜索框与表示表内项目数的气泡均基于Bootstrap框架实现。表格内部被分为表头和表体两部分，分别由两个HTML文件作为模板，详情见上一节可排序表格的内容，表格的显示与HTML标签原生的表格与Bootstrap框架类似，但通过表格体item的筛选实现可排序功能。表格体内容的筛选通过调用后端interface模块的接口进行，表格体内容的实时加载在后面JavaScript部分会提及。

4.4 JavaScript部分

**4.4.1概况**

JavaScript部分解决了绝大多数页面逻辑以及交互效果的实现，是本项目中代码量最多的部分。本项目使用了两个核心JS框架，分别为jQuery.js和History.js。

前者是当下最流行的JS库之一，目前能看到的大多数网站都是基于jQuery开发的。它对ECMAScript原生操作进行了很好地包装，提供了操作DOM元素的便捷方法，使前端开发更为方便和快捷。

后者则可以借助HTML5的函数修改浏览器的地址栏与前进/后退历史。由于本项目使用了Ajax技术实现页面切换，在进行跳转的时候只载入了部分页面，避免了实际的链接跳转。因此浏览器不会修改地址栏与前进/后退历史，如此地址栏的URL会一直保持第一个页面的URL，如果用户刷新页面或者点击前进/后退，就会跳转到错误的页面，与预期相应不符。History.js则解决了这个问题，使利用Ajax请求完成页面切换的同时，也能正确响应刷新、前进、后退的操作。

除了两个核心JS框架之外，在实现一些细节功能时本项目还用到了其他JavaScript开源库，比如FusionCharts图表套件、Validator表单验证、TableExport导出文件、icheck美化选框等。再次向各位库作者表示感谢。

另外，关于下文将要提到的JS文件，有一点需要说明的是，在各目录中可能存在一些同名，但是以.min.js结尾的文件。这些文件是对应JS文件压缩之后得到的小体积JS文件，内容上没有任何区别。在实际部署使用时，应当使用这些JS文件。

**4.4.2文件列表**

|  |  |
| --- | --- |
| 路径 | 描述 |
| base/js/charts.js | 渲染FusionCharts图表。详见“可视化图表”部分。 |
| base/js/scripts.js | 核心部分。详见下文“核心部分介绍”。 |
| bootstrap/js/\* | Bootstrap相关的文件。 |
| fusioncharts/\* | FusionCharts图表相关文件。 |
| ie/\* | 兼容IE的相关文件。包括旧版本的jQuery等。 |
| jquery/jquery.js | jQuery的核心文件 |
| jquery/jquery.fixer.js | 基于jQuery的简易布局插件，用于确定左边栏位置。 |
| jquery/jquery.history.js | History.js框架核心文件。 |
| jquery/jquery.scrollto.js | 基于jQuery的滚动插件。用于将页面滚动到指定位置。 |
| md5/md5.js | MD5加密。用于密码和问序号等加密。 |
| validator/validator.js | 表单验证插件。 |
| TableExport/\* | 表格导出成文件插件。用于导出Excel表格。 |
| icheck-1.x/icheck.js | 基于jQuery的选框美化插件。 |
| FileSaver/\* | 供TableExport使用的文件保存插件。 |

表15：JavaScript文件列表

**4.4.3核心部分介绍**

interface模块static/base/js文件夹中的scripts.js是本项目的核心JS文件，里面包括了一些常用的函数以及各页面通用的一些操作，比如Ajax载入、表单提交、绘制图表等。其中也包含了一些自定义的工具函数，方便各个地方调用。下面具体说明几个核心函数的功能和属性：

**Ajax载入响应函数**

**loadContent**(url, params, item\_selector, load\_params, callback)

通过Ajax请求加载指定url的页面内容到#main-page上。

|  |  |  |
| --- | --- | --- |
| 参数列表 | | |
| 参数 | 类型 | 描述 |
| \* url | string | 需要载入的目标URL。 |
| params | JSON | Ajax请求附加的信息，根据具体页面而变化。 |
| item\_selector | string | 对应左边栏项目的id。 |
| load\_params | JSON | 载入时的选项。 |
| callback | function() | 函数执行完后调用的回调函数。 |
| load\_params参数列表 | | |
| replace | boolean | 是否是对页面的更新，而非载入新的页面。 |
| anim | boolean | 是否使用淡入淡出的动画效果。 |

表16：loadContent函数参数列表

**loadContentOn**(container, url, params, load\_params, callback)

通过Ajax请求加载指定url的页面内容到指定DOM对象container上。

|  |  |  |
| --- | --- | --- |
| 参数列表 | | |
| 参数 | 类型 | 描述 |
| \* container | string | 待载入内容的DOM对象选择器。 |
| \* url | string | 需要载入的目标URL。 |
| params | JSON | Ajax请求附加的信息，根据具体页面而变化。 |
| load\_params | JSON | 载入时的选项。 |
| callback | function() | 函数执行完后调用的回调函数。 |
| load\_params参数列表 | | |
| anim | boolean | 是否使用淡入淡出的动画效果。 |

表17：loadContentOn函数参数列表

**loadContentOfItem**(item, load\_params, callback)

通过Ajax请求加载左边栏项目对应的主页面。

|  |  |  |
| --- | --- | --- |
| 参数列表 | | |
| 参数 | 类型 | 描述 |
| \* item | string | 待载入的左边栏选择器。 |
| load\_params | JSON | 载入时的选项。 |
| callback | function() | 函数执行完后调用的回调函数。 |
| load\_params参数列表 | | |
| replace | boolean | 是否是对页面的更新，而非载入新的页面。 |
| anim | boolean | 是否使用淡入淡出的动画效果。 |

表18：loadContentOfItem函数参数列表

**displayContent**(data, params, container, callback)

将Ajax请求得到的页面内容data载入到指定DOM元素container中。

|  |  |  |
| --- | --- | --- |
| 参数列表 | | |
| 参数 | 类型 | 描述 |
| \* data | string/object | 载入的页面内容。可以使HTML字符串或DOM。 |
| params | JSON | 载入时的选项。 |
| container | string | 待载入内容的DOM对象选择器，默认为主界面。 |
| callback | function() | 函数执行完后调用的回调函数。 |
| load\_params参数列表 | | |
| scroll | boolean | 是否将页面滚动到带载入对象的顶部。 |
| anim | boolean | 是否使用淡入淡出的动画效果。 |

表19：displayContent函数参数列表

**表单提交处理函数**

**handleFormPost**(form\_selector, post\_url, params)

通过Ajax请求加载指定url的页面内容到#main-page上。

|  |  |  |
| --- | --- | --- |
| 参数列表 | | |
| 参数 | 类型 | 描述 |
| \* form\_selector | string | 需要处理的表单选择器。 |
| \* post\_url | string | 表单提交发送POST请求的目标URL。 |
| params | JSON | 表单处理的选项。 |
| params参数列表 | | |
| success\_callback | function(data) | 表单提交成功时调用的回调函数。 |
| error\_callback | function(data) | 表单提交失败时调用的回调函数。 |
| success\_message | function(data) | 表单提交成功时显示的提示信息。 |
| before\_submit | function(data) | 提交表单之前调用的回调函数。 |

表20：handleFormPost函数参数列表

**弹出窗口函数**

**showConfirmModal**(title, message, one\_button, callback)

弹出Bootstrap风格的对话框。可以弹出只含有一个按钮的提示对话框，也可以弹出包含两个按钮的的确认对话框。后者在点击“确认”按钮时会调用指定的回调函数。

|  |  |  |
| --- | --- | --- |
| 参数列表 | | |
| 参数 | 类型 | 描述 |
| \* title | string | 对话框标题。 |
| \* message | string | 对话框内容。 |
| one\_button | boolean | 是否弹出只包含一个按钮的对话框，默认false。 |
| callback | function(event) | 点击“确认”按钮时调用的回调函数。 |

表21：showConfirmModal函数参数列表

**showModal**(url, id)

弹出指定的URL中的对话框。

|  |  |  |
| --- | --- | --- |
| 参数列表 | | |
| 参数 | 类型 | 描述 |
| \* url | string | 对话框的URL。 |
| \* id | string | 该对话框DOM对象的ID。 |

表22：showModal函数参数列表

**页面更新及处理函数**

**initAjaxPage**(container)

为指定DOM对象container中需要使用Ajax请求完成跳转的链接（.ajax-link）进行点击事件的绑定，通常在刷新页面之后对刷新的DOM对象调用。

**loadComplete**()

刷新页面内所有的数字气泡（.badge），通常在刷新页面之后调用。

**resizeComponents**()

页面宽度发生变化时的处理函数，会调整页面的最小高度，并触发左边栏的滚动事件，以更新其坐标。另外，函数还会调整对话框（阻塞式窗口）的暗色背景高度，因此在弹出对话框的函数中也会用到此函数。

**图表绘制函数**

**drawCharts**(selected\_charts)

对于指定的包含图表的DOM对象，载入其中的FusionCharts图表。详见“可视化图表”部分。

**初始化函数**

**initLeftColumn**()

处理左边栏的相关事件，具体包括：

* 绑定移动端界面的弹出左侧边栏按钮的点击事件；
* 处理顶部标题中箭头的方向切换；
* 绑定左侧边栏内部的滚动事件；
* 为侧边栏中非选中项目绑定点击事件；
* 使用jQuery.fixer完成左侧边栏的页面内位置处理。

**自定义功能函数**

**stop\_act**(act\_id, url, item)

停止收集问卷，向后台发送消息并刷新当前页面。

|  |  |  |
| --- | --- | --- |
| 参数列表 | | |
| 参数 | 类型 | 描述 |
| \* act\_id | string | 要停止的问卷ID。 |
| \* url | string | 通知后台时要GET的目标URL。 |
| \* item | string | 当前所在页面对应左边栏的项目，用于刷新。 |

表22：stop\_act函数参数列表

**resume\_act**(act\_id, url, item)

继续收集问卷，向后台发送消息并刷新当前页面，参数同上。

**remove\_act**(act\_id, url, item)

删除问卷，向后台发送消息并刷新当前页面，参数同上。

**publish\_act**(act\_id, url, modal\_url)

发布问卷，弹出提示框，不要求用户输入邮箱，modal\_url为对话框URL。

**email\_act**(act\_id, modal\_url)

弹出发布问卷的对话框，需要用户输入联系邮箱并将问卷和后台链接发送至邮箱。

**callRepeated**(callback, cycles, time)

每隔一段时间反复调用指定的函数若干次。

**animate**(item, animation)

对指定DOM对象施加单次的Animate.css动画（默认情况下动画会循环播放）。

**removePx**(str)

从字符串中删除末尾的“px“，并转换成数字返回。用于处理CSS的height一类属性。

**4.4.4可视化图表**

图表绘制主要使用了FusionCharts开源库（<http://www.fusioncharts.com/dev/chart-attributes.html>），项目中与其相关的JavaScript文件共有9个：

|  |  |
| --- | --- |
| 路径 | 描述 |
| base/js/charts.js | 渲染FusionCharts图表。 |
| fusioncharts/startrender.js | 指示开始渲染的文件。详见下文。 |
| fusioncharts/fusioncharts\*（共3个）  fusioncharts/themes/\*（共4个） | Fusionchart图表主题等相关文件。 |

表23：FusionCharts相关文件列表

其中charts.js中只有一个函数：

**renderChart**(container, type, data, params, callback)

用于在指定的DOM对象中绘制FusiongCharts图表。具体参数含义基本类似其官方示例，可以参照官网的说明文档。

由于FusionCharts的JS文件体积庞大，加载需要相当长的时间，因此需要使用非阻塞的载入方式。在绘制时先载入所有JS文件，最后载入startrender.js，其中包含开始绘制的指令。

fusioncharts文件夹下的其他文件则是图表主题等配置文件，为开源库自带。

项目中用到图表的页面只有一个，为对单个题目进行分析的页面，其中图表页面被单独分离出来作为一个单独的HTML（legalUser/statistics/charts/charts.html）。这个页面中包含一些调用图表绘制的JavaScript代码，主要作用是初始化页面内的图表组件，并载入其他几个JS文件。图表的信息则是在Django模板渲染时就得到的，以JSON格式存入页面之中。具体来说，该信息转换为字符串存入待绘制的DOM对象的data-json属性中。图表的类型则存在data-type属性中。drawCharts函数会读取DOM对象的这两个属性，并调用renderChart函数完成绘制。

**4.4.4导出表格和美化选框**

导出表格使用的是github上TableExport插件，使用非常简单，只需要对需要导出的表格调用tableExport函数即可，具体参数如下：

|  |  |  |
| --- | --- | --- |
| 参数列表 | | |
| 参数 | 类型 | 描述 |
| headings | boolean | 是否显示table headings。 |
| footers | boolean | 是否显示table footers。 |
| formats | string[] | 导出的文件类型集合。 |
| fileName | string | 导出文件的文件名。 |
| bootstrap | boolean | 是否采用Bootstrap风格的按钮。 |
| position | string | 取值为top或bottom，决定按钮位置。 |

表24：tableExport函数参数列表

本项目中需要导出的是填写问卷的情况。导出的这个表格包含信息较多，页面上没有哪个显示的表格同时包含这么多的信息，所以单独设置了一个隐藏的表格专门用来导出，在页面上不显示（legalUser/statistics/print\_table.html）。其中包含一段JavaScript调用tableExport函数来支持表格导出。使用时只要在需要导出表格的地方使用”{% include ... %}”语法包含该HTML页面即可实现导出。

选框上由于Bootstrap自带的单选和复选框不甚美观，所以使用了iCheck插件来美化选框。在所有使用到选框的页面中加入iCheck的配置语句，即可把选框的样式重新设置为指定的风格。同时该插件还可以自定义增加选框的点击范围，使点击在较小的屏幕上也变得更加容易。插件也支持各种回调函数，用于处理选中和取消选中的事件。

五、总结

这次参与本项目是本小组所有成员第一次接触这种系统的，可以被称作软件工程的软件开发。在本项目开始之前的几年中，我们在大作业中也完成了各种各样的“软件”，例如大一时候完成的中英文字典，FlowFree小游戏，在线五子棋，大二小学期完成的安卓APP等。在完成了本项目之后我们深刻体会到了即使是简化后的软件开发流程也比原先所接触的这些大作业要复杂的多，对软件品质的要求也要高得多。

从十月初理想到十一月末正式完成，这两个月时间的经历中，我们觉得以下几点是给我们感触最大的部分，也是我们在这门课中除了完成课程以外最大的收获。

**作为计算机行业的从业人员，学习能力非常重要**

本项目开发的过程中，主要使用了Django+Python，HTML与CSS，javaScript三种语言，除了Django+Python的架构一年多之前曾经短暂接触过，并且留下的印象并不深外，其余的语言可以说从来没有接触过。在本次开发的最初阶段，小组成员的主要工作便是学习数门新的语言，并且在没有完全掌握的情况下便投入到了开发中。

为此一度出现了多人工作进度处于停滞状态的现象，以及开始开发时想要获得一个能够被接受的版本非常难。小组某位成员刚开始接触HTML加上CSS，便接手任务绘制用户登陆界面，在最终完成之前被否决了三个版本。在后期开发熟练后情况有了很大好转。

在以后的工作中，作为计算机行业的从业人员，一定也会遇到许多这种情况，会去接触一些新的工具并尝试去使用它，不能有吃老本的思想，一定要贯彻终身学习的方针。

**团队工作中，一定要进行充分的沟通**

本小组的成员均来自一个班，其中三人更是来自同一个宿舍，按照常理来讲沟通渠道畅通无比，然而在实际开发的过程中还是或多或少出现了沟通方面的问题。这方面的问题主要出现在开发工作的初期，由于所有人对于本项目开发的构想均还没有一个明确的目标，再加上适逢国庆假期，沟通略有不便，因此造成了本项目启动阶段非常缓慢的问题。

在这里要感谢小组长王晨阳个人的努力，他依靠自身的能力率先确立了一个大致的方案，并完成了初步的实现，其余小组成员落实这个方案的实现，最终使我们的开发走上了正轨。但是凡事不能总依靠个人的力量，在实际的开发过程中一定要注意充分的沟通。

**软件开发过程中必须谨遵流程**

在本学期的过程中我们逐渐学习了一些软件工程的原则，而且在开发的开始阶段就被教授了关于迭代开发计划等方面的流程，然而实际实践中，由于人的大意，并没有能够完全的遵守这些原则与流程，特别是在开发进行了一段时候后产生了松懈情绪。

例如某位同学在开发过程中没有即使的利用git工具保存完成的工作并进行版本控制，因而在误删了工程文件夹后丢失了一周的工作量。例如某另一位同学没有遵守迭代开发计划，在一个迭代周期中开发了其它迭代周期的内容，因而影响了计划的进行。这些教训让我们体会到，若不谨遵流程，会在某些条件下为此付出代价。

在最后，请允许我们对我们所导入的开源框架的开发者表示感谢，他们的帮主使我们的开发工作更加顺利，也允许我对PRJ1的助教与白老师表示感谢，你们在例会中对工作进度的指导给予了我们很大的帮助。

whyNotDance全体组员敬上