



清华大学统计平台

需求与设计文档

prj1_whynotdance 小组

计 45 班 王晨阳 何熙翼 田文龙 徐子南

目录

一、项目基本情况	4
1.1 概述	4
1.2 团队成员与分工	4
1.3 开发环境	4
1.4 运行说明	4
二、需求分析	6
2.1 概述	6
2.2 问卷系统设计	7
2.3 问卷数据的处理	7
2.4 用户系统管理	8
三、模块及接口设计	9
3.1 概述	9
3.2 用户界面 interface	9
3.3 前后端接口 api	10
3.4 后端数据库 database	11
四、技术细节	13
4.1 概述	13
4.2 Django 部分	13
4.3 HTML 与 CSS 部分	23
4.4 JavaScript 部分	27
五、总结	36

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。

3、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

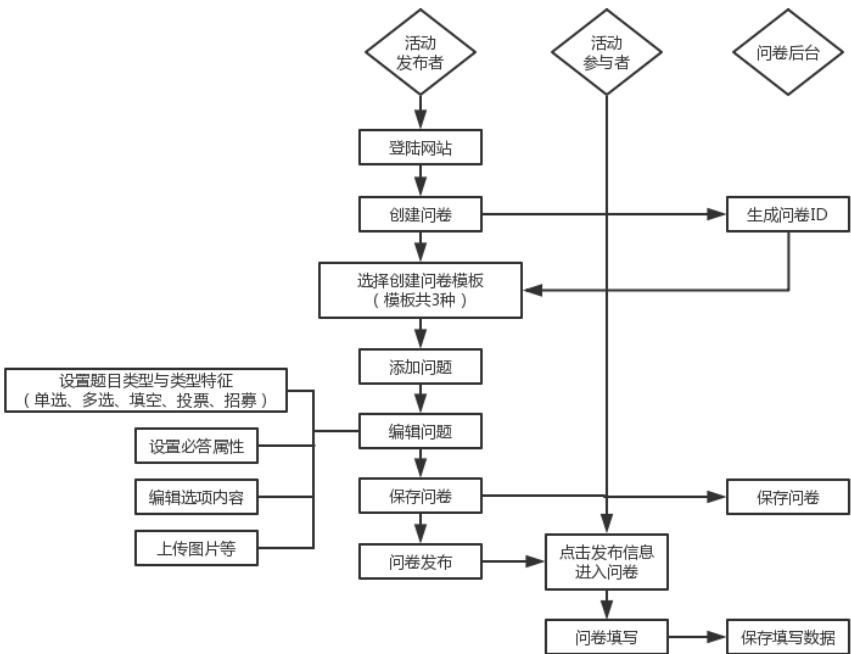


图 1: 问卷系统泳道图

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

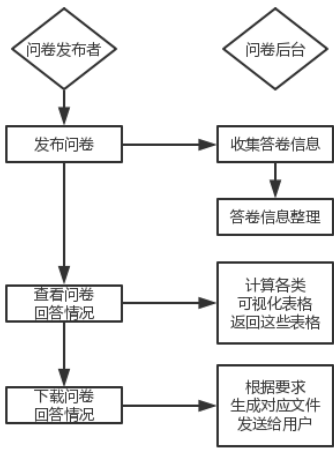


图 2: 问卷数据处理泳道图

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

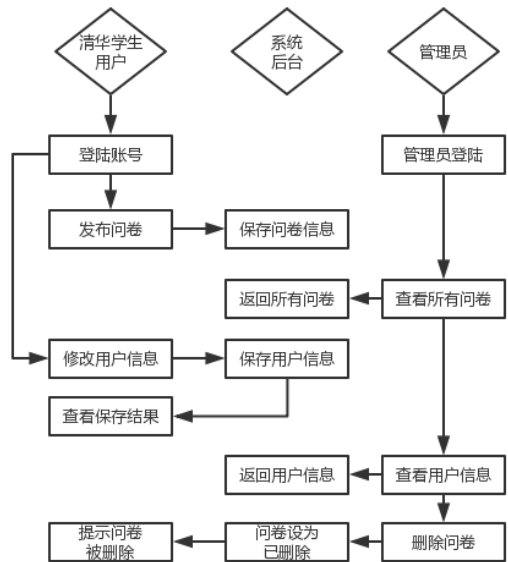


图 3: 用户系统管理泳道图

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。

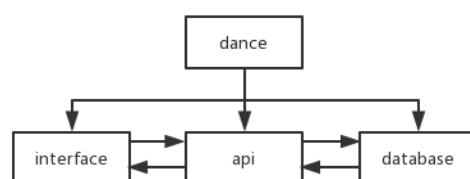


图 4: 模块拓扑关系图

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
legalUser_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。

4.1 概述

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。

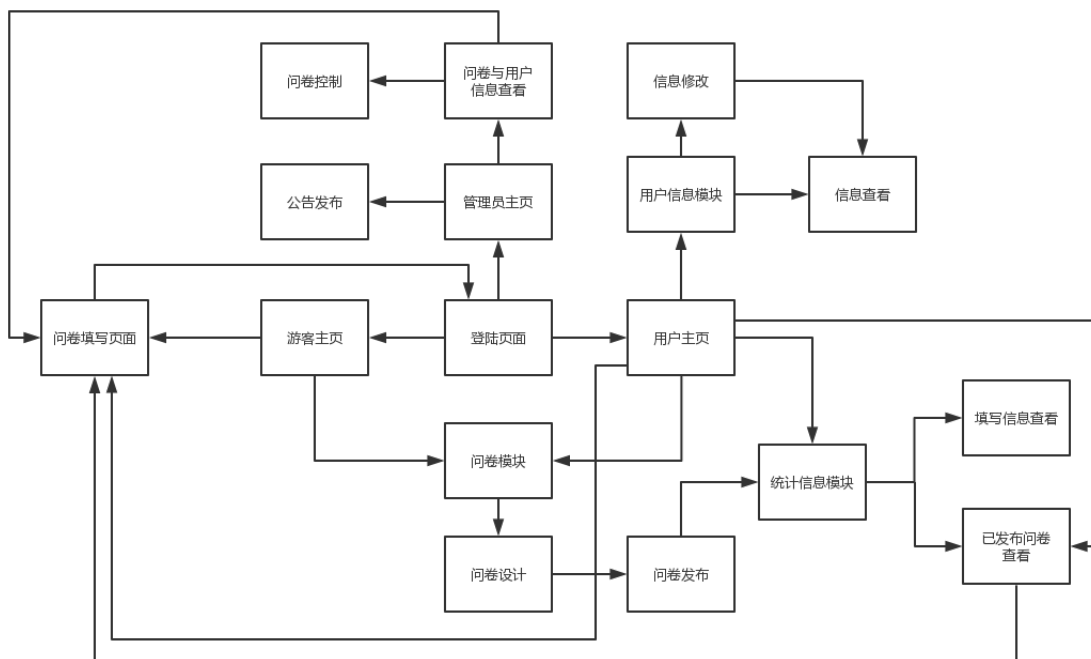


图 5：清华大学信息化统计平台网站拓扑结构图

4.2.1 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	投票题设置弹窗模板

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

Questionnaire

每条记录代表一张问卷。

域名	描述	类型与限制
questionnaire_user	问卷所属用户。	User 外键
questionnaire_title	问卷标题。	字符串，最大长度 30
questionnaire_introduction	问卷介绍。	文本
questionnaire_status	问卷状态，详情见下。	字符串
questionnaire_type	问卷类型，详情见下。	字符串
questionnaire_time	问卷创建时间。	字符串，最大长度 50
questionnaire_ip	问卷创建者 ip。	ip
questionnaire_numOfQues	问题数量。	整数
questionnaire_numOfFilled	已填写次数。	整数
questionnaire_haveMaxTime	是否拥有最大填写次数。	布尔
questionnaire_maxTime	最大填写次数。	整数
questionnaire_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

questionnaire_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

questionnaire_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
questionnaire_id	问题所属问卷，名字为历史遗留问题	Questionnaire 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_questionnaire	填写所属问卷	Questionnaire 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：

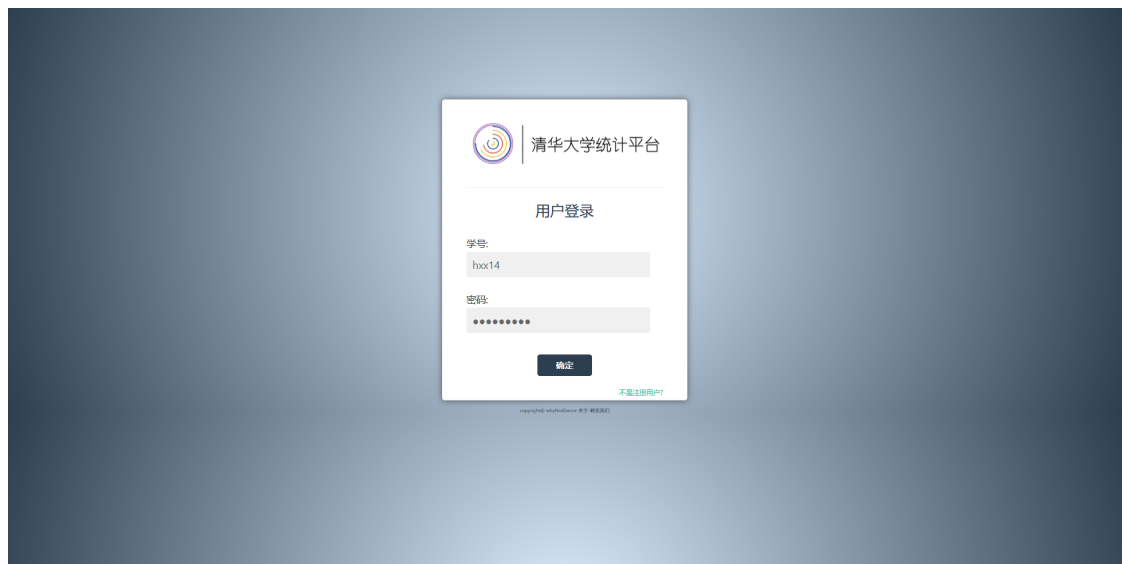


图 8：清华大学信息化统计平台用户登陆界面

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。

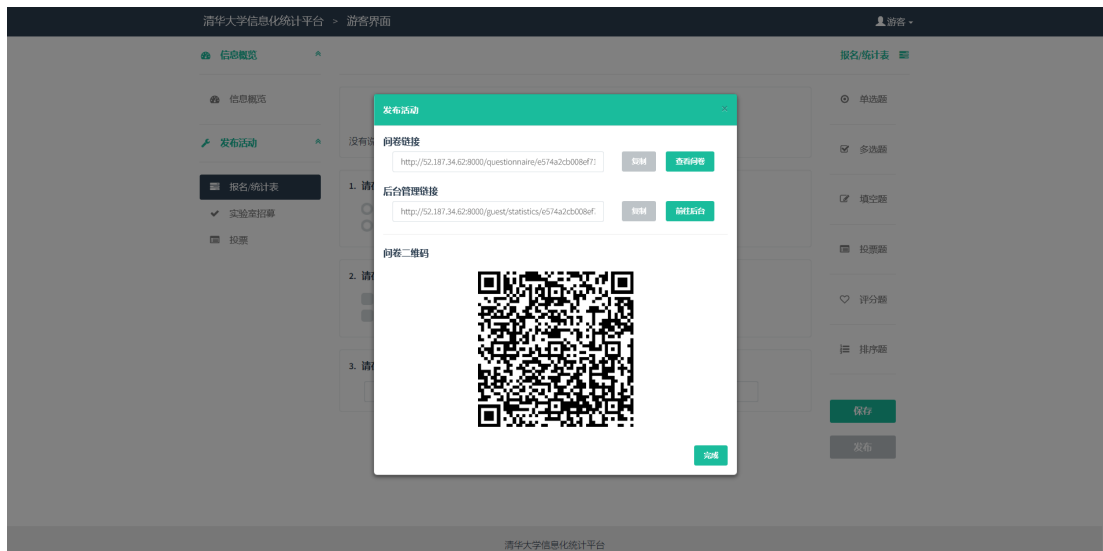


图 9：复杂 HTML 阻塞式窗口



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：

4.4 JavaScript 部分

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。

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 FusionCharts 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此大作业要复杂的多,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间后产生了松懈情绪。

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