基于网页浏览器的《模拟电路》作业提交系统

FH

一、功能要求说明

身份：管理员、教师、助教、学生（每个人都有自己学校的ID号码，拟10位以内，可以有字母，每个人都有密码）

管理员：出题、删题、教师信息管理、上课班级组建（一个上课班级有一个主讲教师，可以没有助教，或者一个以上助教）

教师：助教管理（指定助教的权限）、班级管理、出题、布置作业、批改作业、统计作业完成情况

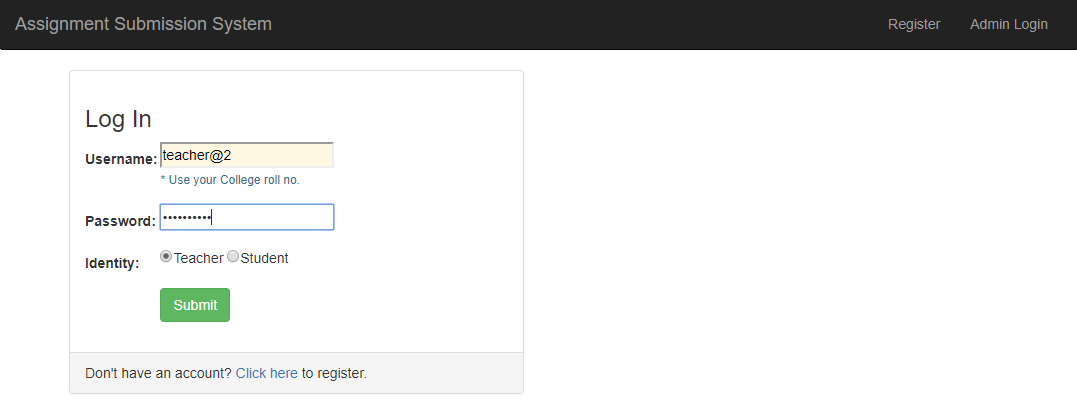
助教：按教师指定的权限完成以下全部或部分工作，班级管理，出题，布置作业，批改作业，统计作业完成情况

学生：完成并提交作业。

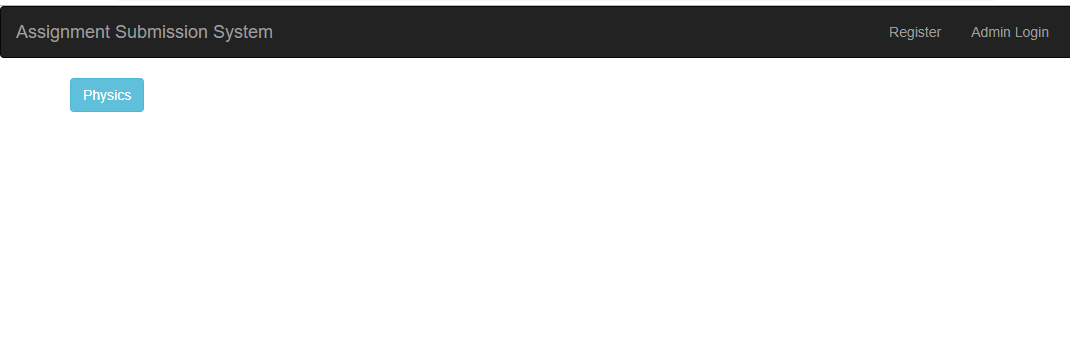
二、工程实现说明

本工程是基于Django框架实现的一个作业提交系统。下面对主要的一些页面进行展示：

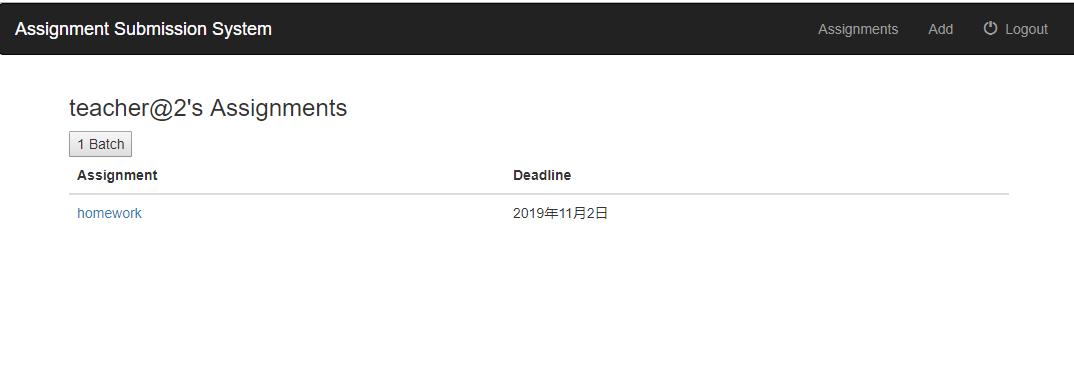
登陆界面：登陆界面分为老师登陆和学生登陆，老师和学生的注册都是通过管理员完成，管理员可以添加老师，助教，学生和出题删题，课程组建等。  
教师登陆界面：



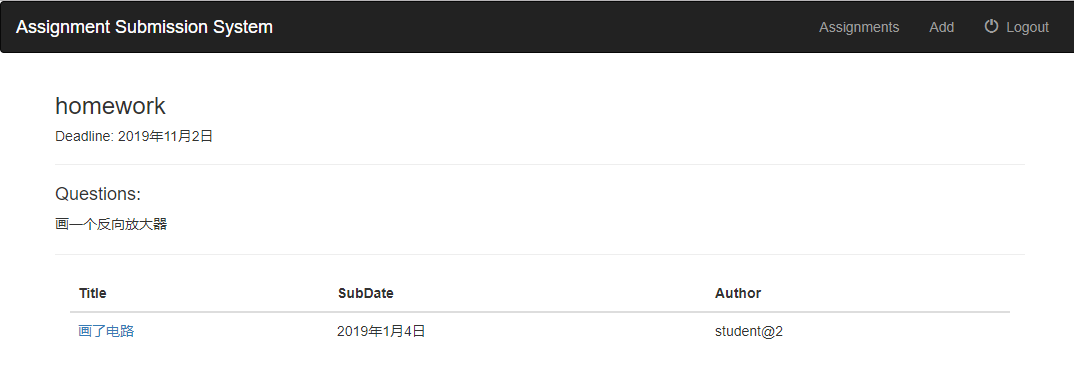
进入之后会显示当前授课老师教授的课程



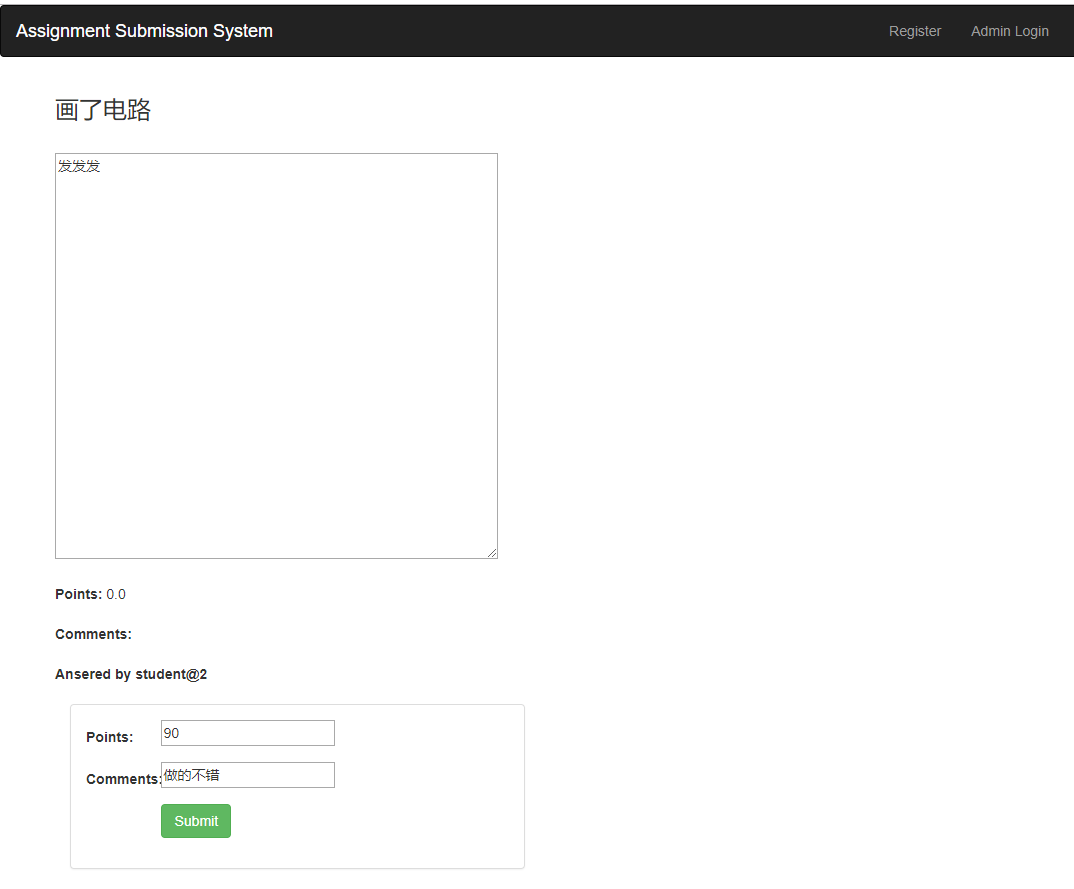
然后点击课程会看到老师布置的作业，有作业名字和作业的deadline:



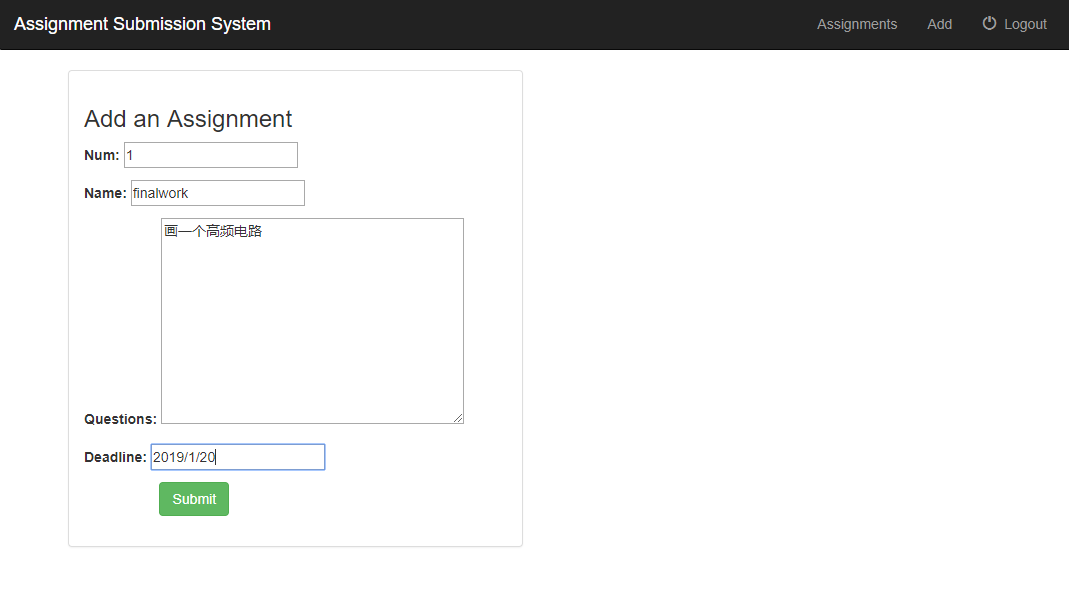
点击进入题目之后会看到学生的提交状态，提交作业的名字，提交日期和提交的学生姓名：



点击作业之后可以对作业进行打分并写上评语，然后提交：

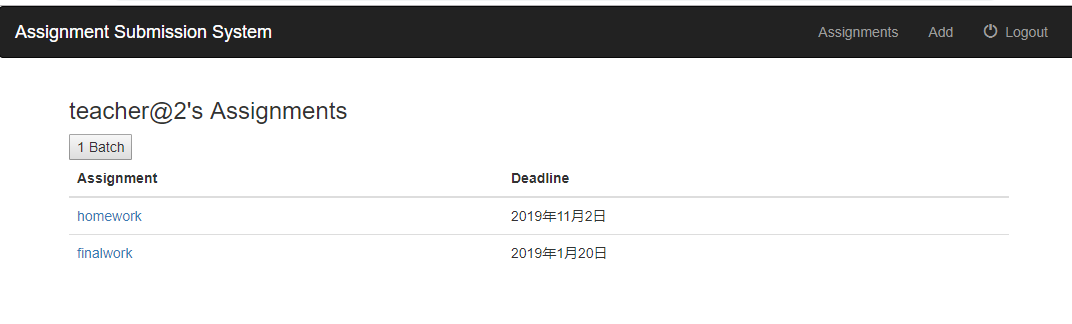


同时老师可以在线出题，点击右上add键：

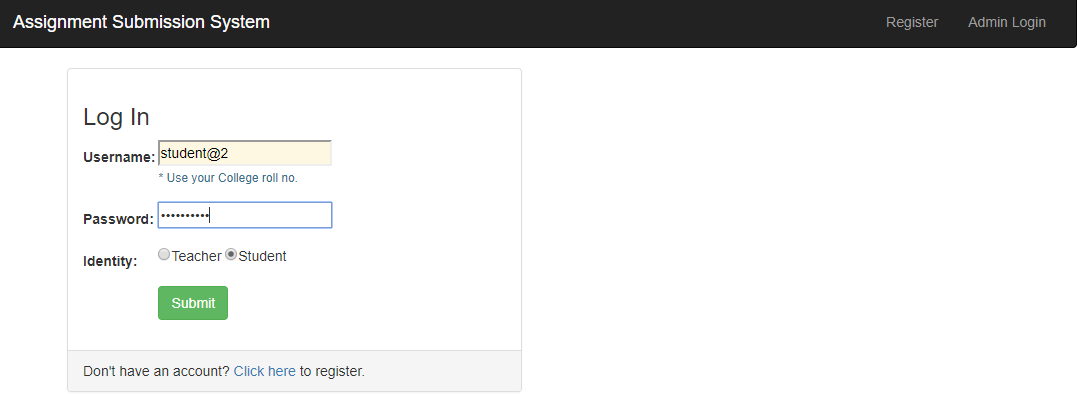


出题，填写题目相关信息（题目信息需要修改一下），然后submit

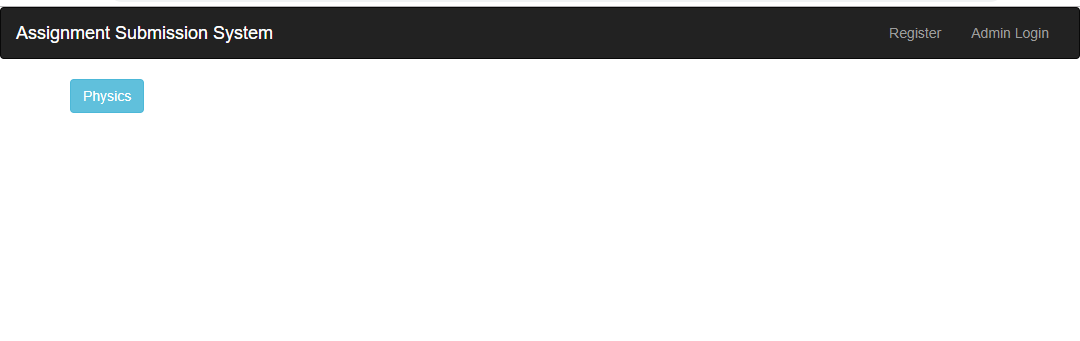
然后在题目界面就可以看到出的题目：



学生登陆界面：



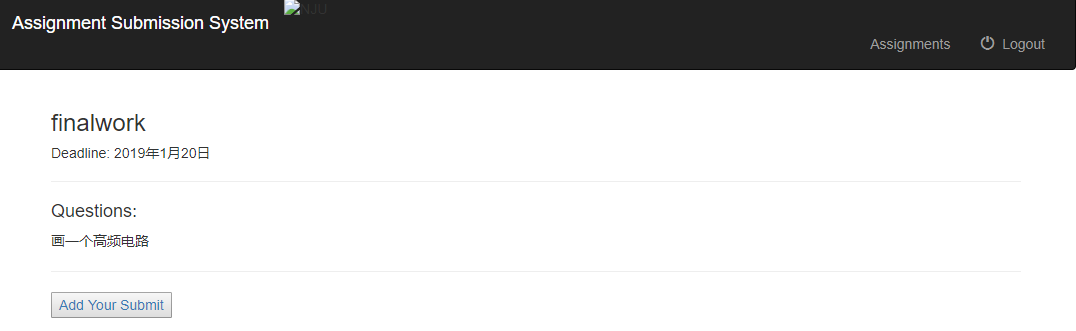
登陆之后可以看到学生选的课：



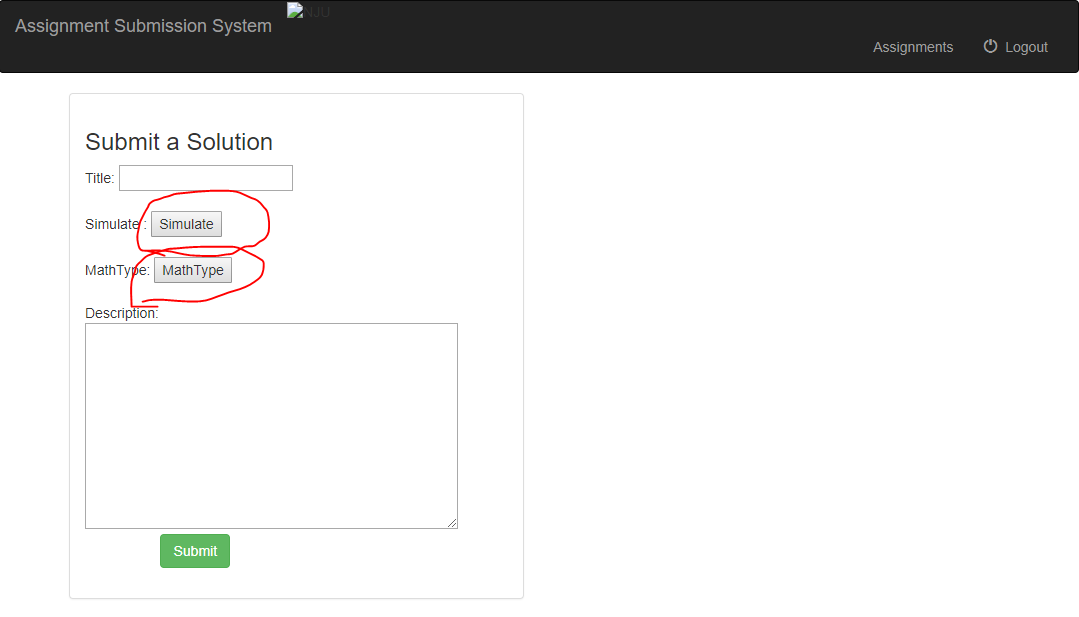
点击选的课程可以看到课程作业以及老师给上个作业的打分和评语：



点击我们刚刚发布的题目：



进入我们的提交页面，然后留了两个接口直接跳转我们的公式编辑和画电路图的页面



然后可以看到自己提交的作业：

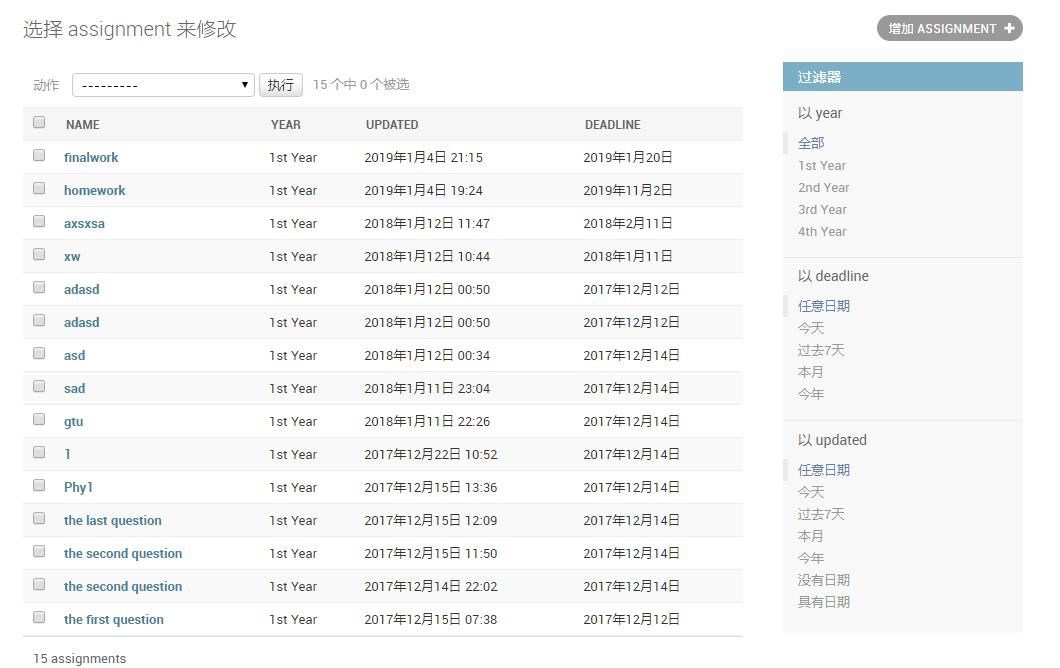


管理员账户：

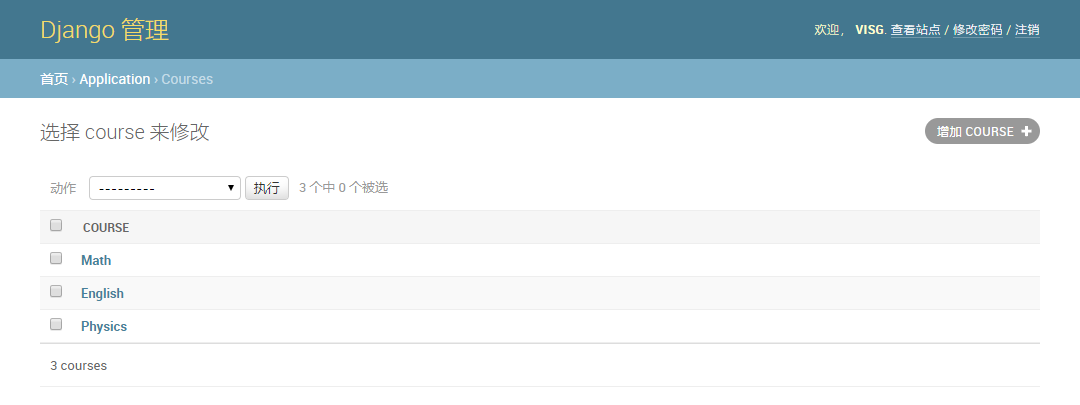
管理员负责Django的后台管理：



可以管理作业：



管理课程：



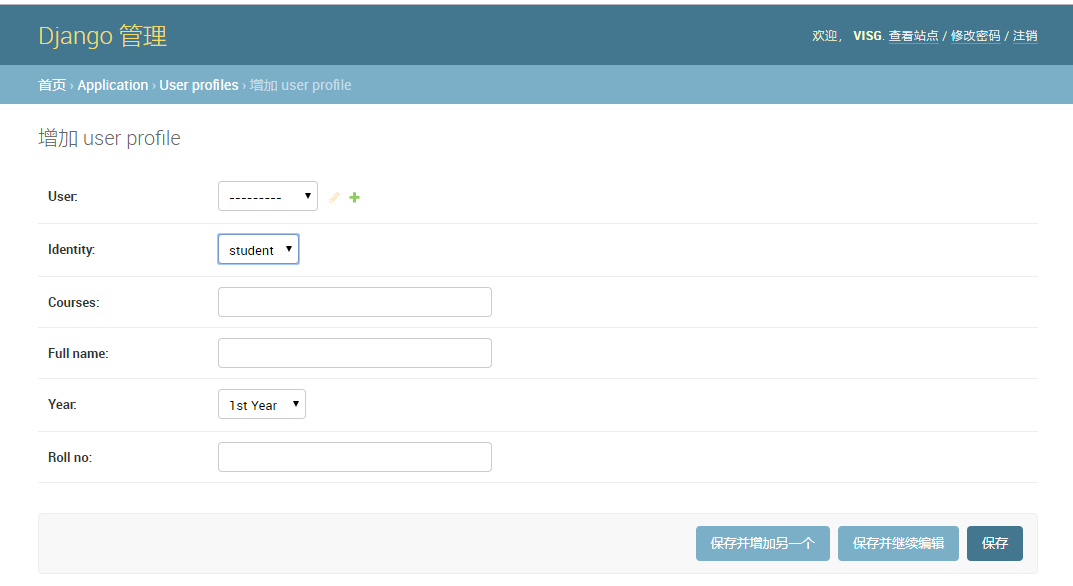
管理作业：



管理人员：



增加人员：

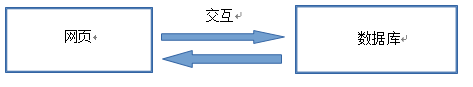


修改用户：



三、程序实现思路

本作业提交系统主要包括网页页面，后台数据库，网页和后台数据库的交互三大部分。整个工程的框架用django框架实现，如下图所示：



Django框架的核心包括：一个物件导向的映射器，用作数据模型和关联性数据库间的媒介，即工程中用Python定义后台数据库的部分；一个基于正则表达式的URL分发器，即工程中urls.py文件所执行的操作；一个视图系统，用于处理请求，即工程中views.py文件的作用，提供了网页和数据库交互的接口；以及一个模板系统，工程里在模板系统中用HTML实现了网页的界面显示等。

1.后台数据库：

在application的models.py文件了定义了数据库中所需要的模型。其中，

**class** Course(models.Model): 定义了课程类。在该类中，只提供了课程名这个属性，目前提供了3门可以选择的课程。

**class** UserProfile(models.Model): 定义了用户类。用户类可以选择老师或者学生，并且可以选择学生的年级，即从一年级到四年级。用户类的属性包括选择是老师还是学生，所选或者所教的课程，老师或学生的名字，学生年级，学号，以及用户创建时间。还提供了保存用户信息的成员函数。

**class** Assignment(models.Model): 定义了作业类。作业类包括课程名字，老师，学生年级，作业的标题，作业的具体内容，创建时间，更新时间，截止日期等多个属性，一个课程可以对应多个作业。

**class** Solution(models.Model): 定义了提交答案类，即学生交作业的类。该类有对应的作业，学生信息，提交日期，提交答案的标题，提交的答案的具体内容，提交答案后的得分，老师对该学生提交答案的评语等属性，一个作业可以对应多次提交。

2.模板系统：

这一部分包含了静态网页内容的目录，用HTML语言完成。主要包括注册界面的实现，即选择注册为老师还是学生，以及姓名id密码等信息的录入。之后的界面即为老师和学生对应不同的界面，包括老师发布作业，批改作业等的界面，以及学生查看作业，提交作业，查看评分等信息的界面。

批量作业的batch的展开和折叠操作是通过JavaScript语言实现的，主要用到了jQuery实现动态效果。

整个html实现的所有网页的静态样式是通过bootstrap框架实现的。在学生提交作业界面里的Body文本框禁止复制粘贴是通过调整HTML的TextArea的onpaste属性实现的。

实现文件在工程template下的application里。

3.settings.py

settings.py文件是对整个项目进行配置。其中用到的sqlite3是数据库的一个功能包，主要提供了在django框架下用Python创建数据库的接口。

4.URL分发器

在urls.py里通过一系列正则表达式定义了访问url的地址，通过这些地址，连接到数据库中的视图函数，从而达到操作网页界面后与数据库交互的目的。形式如：

url(**r'register/$'**, views.register, name=**'register'**),即连接到了register注册的视图函数，其他的操作与上述相似。

5.视图系统

views.py文件即为处理请求的视图函数。当网页上执行操作时，URL分发器连接到相应的视图函数，然后视图函数做出反应，相应的处理数据库中的数据，从而实现了网页和数据库的交互。

其中表单函数即forms.py保存了每个用户的id和password等内容，视图函数会讲表单内容从网页传入到数据库，从而对数据执行一系列的操作。

目前的工程还没完全完成，后续一些功能需要进一步实现，暂时先这些。