

# 深圳大学考试答题纸

(以论文、报告等形式考核专用)

二〇二一~二〇二二 学年度第 二 学期

课程编号	150037 000301	课序号	01	课程名称	数据库系统	主讲教师	彭小刚	评分
学号	2020151002	姓名	郑杨	专业年级	2020 级软件工程腾班			
学号	2020155021	姓名	陈敏涵	专业年级	2020 级软件工程腾班			

教师评语:

题目:

云汉教育管理系统

## 个人总分权重分配表

(1 人总分为 100%，2 人为 190%，3 人 280%)

排序	姓名	学号	项目个人权重
1	郑杨	2020151002	95%
2	陈敏涵	2020155021	95%

我组成员总共 2 名，权重总和为: 190 %

本组成员郑重承诺在数据库设计 Database Software Design and Implementation 项目实施的过程中不发生任何不诚信现象，一切不诚信所导致的后果均由本组成员承担。同时我组成员同意此项目个人项目得分按照教师评分乘以“个人总分权重分配表”的个人权重获得。

签名（手签，并拍照或者扫描附于此）:

陈敏涵 郑杨

# 深 圳 大 学

## 实 验 报 告

课程名称: 数据库系统

实验序号: 04

实验名称: 实验 4: 数据库设计

项目名称: 云汉教育管理系统

组 别: 畅通无组

成员名单:

学 号: 2020151002 姓名: 郑杨

学 号: 2020155021 姓名: 陈敏涵

实验日期: 2022 年 6 月 1 日 - 2022 年 6 月 13 日

目录

一、实验目的.....4

二、实验要求.....4

三、实验设备.....4

四、实验内容.....4

    1 项目简介.....4

        1.1 背景.....4

        1.2 意义.....5

        1.3 卖点.....5

        1.4 功能.....5

    2 项目架构.....6

        2.1 项目流程介绍.....7

        2.2 各大模块功能介绍.....9

    3 项目关键设计.....13

        3.1 根据需求绘制 ER 图.....13

        3.2 关键数据表设计，对标各大模块功能.....13

        3.3 一些关键功能的具体实现与代码片段.....16

    4 项目效果图.....22

        4.1 注册与登录界面效果展示.....22

        4.2 教员界面效果展示.....24

        4.3 学员界面效果展示.....29

        4.4 管理员界面效果展示.....30

    5 任务分工及所完成的状况.....35

五、问题分析.....35

六、实验心得.....36

七、参考文献.....36

## 一、实验目的

- 1、了解 DBMS 系统的功能、Web based 数据库的软件组成及工具；
- 2、掌握数据库软件的使用方法、
- 3、掌握 php+mysql 或其他界面结合数据库操作方法。
- 4、掌握数据库系统软件的制作过程

## 二、实验要求

- 1、构思一个以数据库为后台的，具备某种功能的软件系统
- 2、从模块化描述该系统的功能，对各个模块进行描述
- 3、按照每个模块的功能，设计相应的数据库以及程序
- 4、完成模块间的接口，整合成为一个完整系统并测试所有的功能

## 三、实验设备

计算机、数据库管理系统如 DB2，mysql 等软件。

## 四、实验内容

### 1 项目简介

#### 1.1 背景

项目作者在高考结束后加入了广东揭阳当地的一个非盈利家教组织“云汉教育”，并担任该机构的管理层。云汉教育成立于 2014 年，是一家由揭阳本地优秀高中毕业生创办的学生家教机构。除市区以外，我们教员在揭东，揭西，惠来等地均有分布。云汉教育团队自成立以来，始终秉承“服务教员，服务学员”的理念，吸收优秀的高考毕业生并一直致力于搭建教员与学院之间的桥梁。

##### 1、良好的教育氛围

云汉配备学员群，教员，对每一个教育环节严格认真负责，教员学员之间自由分享学习经验，击破各个弱点，在暑期弯道超车。

##### 2、亲切相伴，亦师亦友

教员均是高考毕业生，作为同龄人或大哥哥大姐姐，我们与学员的陪伴，不只是在学习上，更在生活中，用我们的阳光与热情，用我们体验过的成长历程，与陪伴指导学员。

### 3、灵活的教育方式

云汉的每一位教员会根据学员的学习吸收程度与薄弱点，自由变更教育方式，学员与教员也可自由沟通教学地点与时间，使教学更加科学与人性化

### 4、严格的教学流程

在每次教学开始，云汉教员会与学员签署合同，对这一次教学任务负责，家长可以免除后顾之忧。

## 1.2 意义

“云汉教育”在教员和学员匹配的过程中遇到比较麻烦，需要较多的匹配管理员，因为在匹配上需要浪费较多时间，具体的人工匹配流程如下。

教员和学员在问卷星填写信息之后，管理员需要将信息从问卷星中导出到 excel，并且在每次新增一个学员或者新增教员的时候进行匹配，匹配时不仅需要注意年级和科目上的匹配，还要手动复制粘贴每个学员和教员的地址到百度地图中查看距离，根据科目匹配情况和距离远近为学员挑选合适教员。在找到合适的学员之后还要对教员进行通知。

教员可能因为无法接受学员距离或者暂时不想教授相应科目等而不想要管理员给出的匹配，学员也可能因为和教员性格不合或其他原因而不想继续教学过程，这样又需要重新协调新的匹配。

本项目做的主要工作是将匹配和通知过程自动化，减少管理员重复冗杂的工作；同时给予教员和学员更改个人信息和确认匹配结果的权限，以使得匹配工作更加灵活。在教员结课后，系统自动计算教员应缴提成并展示给管理员。最后还添加了财务记账功能，收入和支出过程更加透明和易记，使得不同届管理员交接的时候更加方便。

## 1.3 卖点

界面较简洁，从具体需求出发，基本满足了整个家教流程的需求。

## 1.4 功能

大体上分为用户注册和登录界面、教员界面、学员界面和管理员界面。

功能模块	具体介绍	完成进度
用户注册登陆模块	提供用户注册，登陆管理的功能 对于管理员用户，直接登录 对于学员用户，直接注册登录 对于教员用户，需要邀请码才能注册登录 做成两个界面，登录界面和注册界面。 注册界面分为两种（教员和学员），管理员不需要注册直接登录。 并将用户创建的用户信息导入数据库。	已完成
教员首页模块（教员）	每次教员进入系统即进入此模块。顶部放置一个轮播的宣传栏，宣传内容为云汉成员的工作日常和轰趴活动等，吸引优秀教员加入	已完成

	云汉大家庭。	
教员个人信息模块（教员）	当教员首次登入系统时会跳转到该模块，提醒教员填写信息并上传。当教员想要修改信息时，可以到该模块进行信息修改并上传。	已完成
教员匹配确认模块（教员）	生成一条匹配记录会先通过邮箱通知教员，若教员确定要此学员可与学员联系，在双方均确认之后开始教学。	已完成
教员授课时长模块（教员）	教员以填写方式提交授课表，授课表的每条信息记录都需要学员确认后才能生效提交。结课之后系统计算出所需要的提成（总工资的 5%）。	已完成
学员首页模块（学员）	学员首页模块，每次学员进入系统即进入此模块。顶部放置一个轮播的宣传栏：如鼓励学员进取的话语，云汉教学模式优势等。底部放置优秀教员信息与云汉管理员联系方式。	已完成
学员个人信息模块（学员）	与教员个人信息模块类似。	已完成
学员匹配确认模块（学员）	在教员确认匹配教员和学员取得联系，学员此时可选择确认或者取消，取消后会重新匹配新教员。	已完成
学员授课时长确认模块（学员）	在教员生成授课记录后，学员需要对每条信息进行确认，但没有删除等权限。	已完成
教员管理模块（管理员）	把所有教员的信息分页展示出来，提供 <b>编辑、删除和搜索</b> 功能，同时可以生成邀请码以供新教员注册。	已完成
学员管理模块（管理员）	与教员管理模块类似	已完成
匹配信息管理模块（管理员）	匹配之后会展示未匹配的学员的课程，表明此学员无教员可匹配，此时管理员可将此学员的课程信息分享到教员 qq 群中。	已完成
结课信息（管理员）	对于结课后的教员提成，会自动生成一条结课记录，只需要管理员进行确认，管理员与对应教员联系并确认后此记录会添加到账单记录中	已完成
财务管理模块（管理员）	可添加账单记录，如为教员购买保险，印发宣传单等，可添加一条新账单。同时自动添加教员提成。	已完成

## 2 项目架构

本项目基于 Django 框架，使用 Python 进行 Web 开发。

## 2.1 项目流程介绍

项目大致流程图如下：

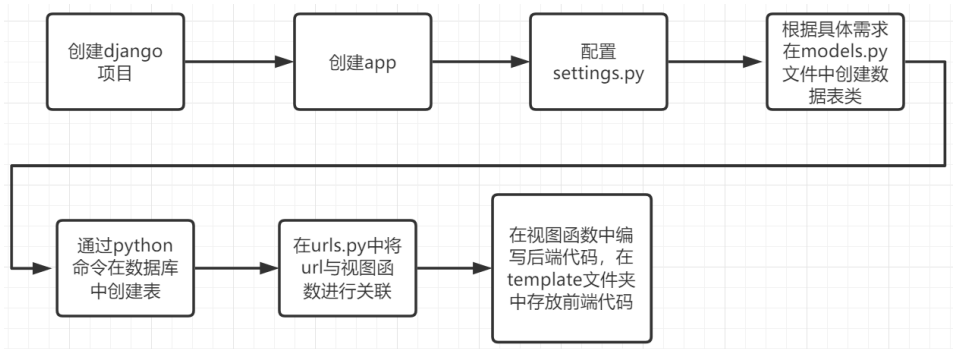


图 1：项目构建流程图

### 2.1.1 创建 django 项目

创建 django 项目可以通过命令行直接创建或者使用 pycharm 直接创建，后者较为方便。

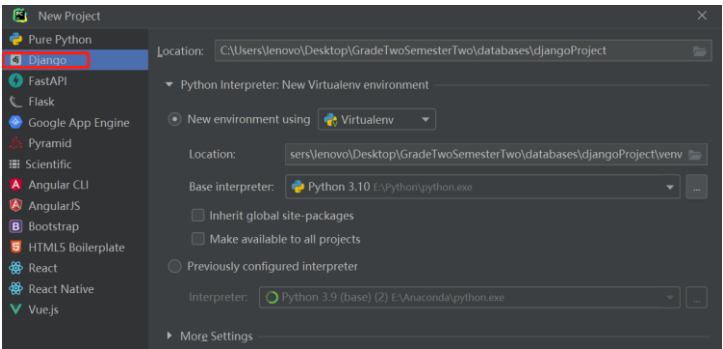


图 2：创建 django 项目

### 2.1.2 在 django 项目中创建 app

使用 manage.py 运行 startapp app01（app 名字）命令即可新建 app，之后将在 app 的文件中编写 python 代码。

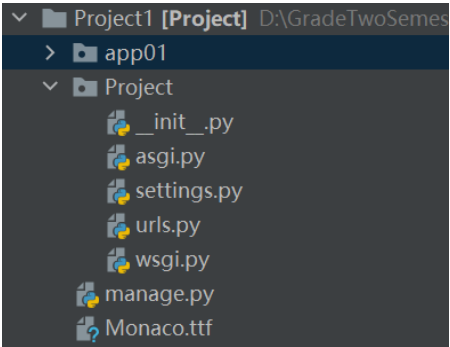


图 3：创建 app

### 2.1.3 配置 settings.py 文件

首先需要将上一步创建的 app 在 settings.py 文件中进行注册，如下：

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'app01.apps.App01Config'  
]
```

图 4：app 注册

配置所使用的数据库信息：

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'yunhan',  
        'USER': 'root',  
        'PASSWORD': '',  
        'HOST': '127.0.0.1',  
        'PORT': '3306'  
    }  
}
```

数据库名  
数据库用户名  
数据库密码

图 5：数据库配置

### 2.1.4 在 app 文件夹下的 models.py 文件中编写数据库表类

编写类似如下的数据库表类，具体数据库表信息将在 3.2 节具体阐述。

```
class Account(models.Model):  
    """ 用户账号 """  
    """ 注册的时候判断一下字段长度 """  
    username = models.CharField(verbose_name="用户名(不超过20个字符)", max_length=25)  
    password = models.CharField(verbose_name="密码", max_length=50)  
    phonenumber = models.CharField(verbose_name="手机号(用于找回密码)", max_length=11)  
    type_choices = (  
        (1, "管理员"),  
        (2, "学员"),  
        (3, "教员")  
    )  
    type = models.SmallIntegerField(verbose_name="账号类型", choices=type_choices)
```

图 6：models.py 中的数据表类

### 2.1.5 运行命令在数据库中创建对应的数据表

在 models.py 中创建数据库表类之后，通过 manage.py 运行以下两个命令即可在数据库中创建对应的数据表：

makemigration

migrate

创建的数据库表名为 app 名+类名，如下：



```
mysql> show tables;
+-----+
| Tables_in_yunhan |
+-----+
| app01_account    |
+-----+
```

图 7：数据库中对应的表

### 2.1.6 在 urls.py 文件中将 url 与视图函数关联起来

Django 框架提供了一个很好用的功能，就是将 url 与视图函数绑定起来，在浏览器访问了 url 相当于进入了某个已绑定的视图函数中，函数可以从数据库中获取数据并返回给前端。视图函数的具体实现会在 3.3 节具体阐述，urls.py 文件的部分内容如下：

```
urlpatterns = [
    # login and register
    path('login/', account.login), #over
    path('image/code/', account.image_code),
    path('register/', account.register), #over
    path('logout/', account.logout), #over
```

图 8：urls.py 中 url 与视图函数的关联

### 2.1.7 在视图函数中编写后端代码

与上图 url 相关的视图函数大致如下：

```
def login(request):...
def image_code(request):...
def register(request):...
def logout(request):...
```

图 9：视图函数

## 2.2 各大模块功能介绍

### 2.2.1 用户注册登录模块

用户分为两种：客户和管理员，其中客户包括学员和教员，需要注册才能登录；管理员为内部账号，不必注册即可登录。

#### 注册功能：

教员注册和学员注册大致一致，需要填写账户名称，账户密码，和 QQ 邮箱。其中账户名称应小于 20 个字符，同时不可与之前注册过的账户名相同，在教员和学员注册之后均会

为该账户生成此账户的类型。QQ 邮箱可用于验证本人以修改密码，进行匹配信息通知等功能。

但是在登录时教员和学员可选择对应的界面，因为二者仍有小区别：教员均需要资质满足才会被邀请成为云汉教员，为了验证教员的身份，会给每个符合要求的教员发送验证码，因此教员需要填写**邀请码**才能注册账号。

#### **登录功能：**

用户的登录功能均相同，填写账户名及密码即可登录，同时在登录界面还有跳转到注册界面，忘记密码以重新设置等功能；在登录后会根据注册时为其生成的账户类型跳转到对应的界面，即教员登录后无法访问学员界面，学员登录也无法访问教员界面，以确保信息安全性。而管理员权限较高，可以跳转到任何界面，因此必须保证管理员账户的安全性。

### **2.2.2 匹配信息模块：**

#### **匹配流程：**

我们会在早上七点的时候进行定时匹配。先枚举每个学员的科目信息为其匹配满足科目要求的教员，首先将教员到此学员的距离（先以公里为单位同时上取整）从小到大排序，排序之后可能会有相同距离的教员，则再按照教员当前已匹配的课程数量从小到大排序，即优先给匹配数量更少的教员进行匹配，若数量仍然相同就随机选取教员。

#### **信息发送：**

考虑到实际情况教员不会一直登录云汉教育系统，因此在给教员匹配到一条课程信息后，会给教员的 QQ 邮箱发送一条信息，提醒教员可到匹配确认界面中确认或删除此匹配信息。

#### **删除匹配：**

为了防止出现某些教员没有关注 QQ 邮箱等，而没有及时对匹配信息进行确认和删除的情况，我们设置每条匹配的生效时间为 3 天：第一天早上 8 点生成一条匹配信息；第二天早上 8 点若发现第一天生成的匹配仍未确认或删除，会再向教员的 QQ 邮箱发送一条提醒信息，提醒他尽快登录系统查看匹配信息；第三天早上 8 点若发现第一天生成的匹配仍未确认或删除，会将此条匹配展示在管理员界面，由管理员直接联系教员，催促其进行删除或确认操作，同时因为教员和学员还需要沟通交流和试课等时间，管理员可手动进行对匹配信息截止日期延期，使得匹配信息更加灵活。

#### **匹配信息走向：**

若学员或教员中的一方将匹配信息删除，则会将此条匹配信息加入一个失配表，之后的匹配将不会将此学员的科目推荐给教员，防止学员教员双方不停删除而不为学员科目进行新匹配的情况；若双方均确认此匹配，则会将匹配结果生成一个授课表记录此科目的教授情况。

具体匹配流程如下：

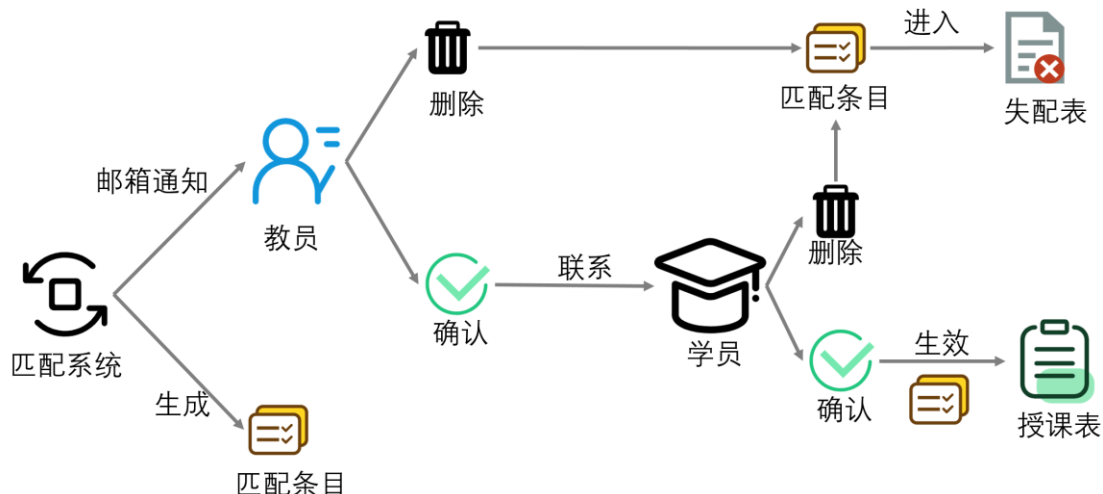


图 10: 具体匹配流程

### 2.2.3 教员界面

#### 主页模块:

除了第一次教员登录后会跳转到个人信息模块进行个人信息完善,之后每次登录都是跳转到主页,最顶部是一个轮播宣传图,是云汉成员内部的活动图,加入云汉的优势等等,下方左侧是云汉内部的编委职务和日常活动,右侧是咨询群;教员首页的主要功能是吸引优秀教员加入云汉编委,可以进行管理宣传等工作。

#### 个人信息模块:

主要用于填写基本信息和授课信息。基本信息包括姓名,性别,毕业学校,地址,联系方式等。授课信息给予教员 11 个科目进行选择,每个科目再给予对应的年级段可供选择,考虑到实际情况小学难度相近,初中难度差距不大,而高中难度差距较大,因此年龄段分为“小学,初中,高一,高二,高三”,对难度进行区分的同时,又不用让教员重复填写相同科目的信息,更加便捷高效。

#### 匹配确认模块:

此处有多条每天早上七点系统匹配的科目信息,每条科目信息会展示学员的基本个人信息,联系方式,课程的当前水平和目标水平,我们不会给出学员的详细地址,以保护学员隐私,但是我们会给出教员与学员的住址距离。对于每条科目信息,若教员对学员的情况和距离等均满意,则可进行确认并根据联系方式与学员联系,在教员确认后会在学员的界面也生成一条匹配信息,在与学员联系后若学员也确认这段辅导关系,则此条匹配生效;若教员和学员中的一方删除此匹配信息,则此匹配信息失效,系统继续给学员该科目继续匹配新的教员。

此外教员可能教授科目已经达到饱和,不想系统继续匹配新的学员,此时可点击按钮“不再推荐新学员”,这样在匹配时就不会将此教员考虑进来;类似的,若教员教授科目课后还想继续匹配科目,则再点击按钮“继续为我推荐教员”,则可继续匹配新的科目信息。

#### 授课表登记模块:

在教员和学员双方均确认某条匹配信息后,该匹配信息自动生成一个授课表存放在教员的授课表登记界面,每次授课结束后,教员对授课情况进行填写,包括教授科目,具体专题,授课日期和时长,形成一个授课记录后会在对应学员端的授课时长确认模块展示,待学员确

认之后此授课记录则生效。

对于教员的一条匹配信息会生成多条授课记录，当结束此匹配科目的授课，并且所有的授课记录都被学员确认之后，教员可点击“确认结课”按钮，就无法对该匹配信息进行添加和删除，此时系统自动生成教员应提交的提成，待教员找管理员缴费之后，此匹配信息会变成“已缴费”状态。

## 2.2.4 学员界面

### 主页模块：

除了第一次学员登录后会跳转到个人信息模块进行个人信息完善，之后每次登录都是跳转到主页，最顶部是一个轮播宣传图，是鼓励学员积极进取的海报，云汉教育的优势等等，下方左侧是部分优秀教员的成绩展示，让学员对云汉教育有更多的信赖，中间是管理员的联系方式，右侧是咨询群；学员首页的主要功能是对云汉进行宣传，让更多的学员在假期时获得我们的帮助，实现弯道超车。

### 个人信息模块：

主要用于填写基本信息和课程信息。基本信息包括姓名，性别，地址，联系方式等。考虑到实际教学时，有较多学员是升年级的，比如高一升高二，可能需要补习高一的数学，预习高二的物理这类情况，因此每条课程信息的年级均可不同，给予学员每个年级段进行选择，而不直接限制学员的年级。

### 匹配确认模块：

此处会针对学员在个人信息模块填写的课程信息，生成多条匹配信息，每条匹配信息会展示教员的基本个人信息，包括姓名，性别，毕业院校，联系方式，而不必展示教员的地址信息，以保护教员隐私。因为是在教员确认后学员才会有匹配信息产生，因此只要学员对匹配信息确认后即生效。若教员和学员中的一方删除此匹配信息，则此匹配信息失效，系统继续给学员该科目继续匹配新的教员。

### 授课时长确认模块：

每次授课结束教员对授课情况进行填写后形成一个授课记录后会在对应学员端的授课时长确认模块展示，待学员确认之后此授课记录则生效。但是学员只有确认授课记录的权限而没有编辑和删除的权限，避免信息被误改误删，确保信息安全。在教员确认此条匹配结课之后，学员此处也会展示此条匹配为已结课状态。

## 2.2.5 管理员界面

### 学员/教员管理模块：

展示学员的简略信息：学员/教员 ID，姓名，联系方式，管理员可对学员/教员的详细信息进行查看和编辑，也可对整条学员/教员信息进行删除。管理员也可通过对学员/教员的 ID 和姓名进行检索以查看，编辑，删除该学员/教员信息。

### 匹配信息模块：

因此在每天早上 7 点系统进行匹配之后，会将所有未成功匹配的教员信息展示在此模块中。考虑到学员的课程信息较为固定，而教员的可教授科目可变动性较大，因此可以由管理员将暂未匹配到的学员信息分享在教员群中，让可接受学员科目的教员提供教员自身对应的 ID，由管理员直接输入教员 ID 生成一条匹配信息。这样可以较高效率地进行匹配，防止出现学员长期无法获得匹配的情况。

### 结课信息模块：

在教员的某条匹配结课之后，会在管理员的结课信息模块展示该匹配的科目，教员 ID，学员 ID，授课时长，同时展示教员应当上交的提成。在管理员确认该教员提成上交后，即可将此条结课信息状态改变为“已缴费”。此“已缴费”状态会同时在教员的匹配信息处显示。

**财务信息模块：**

此处给管理员添加日常的收支记录，如印发宣传单，为教员购买保险等等。同时在每条结课信息缴费之后，会自动添加到财务信息模块进行展示。这样记录可以使得管理员换届交接时更加方便，使整个机构的收支清晰透明。

### 3 项目关键设计

#### 3.1 根据需求绘制 ER 图

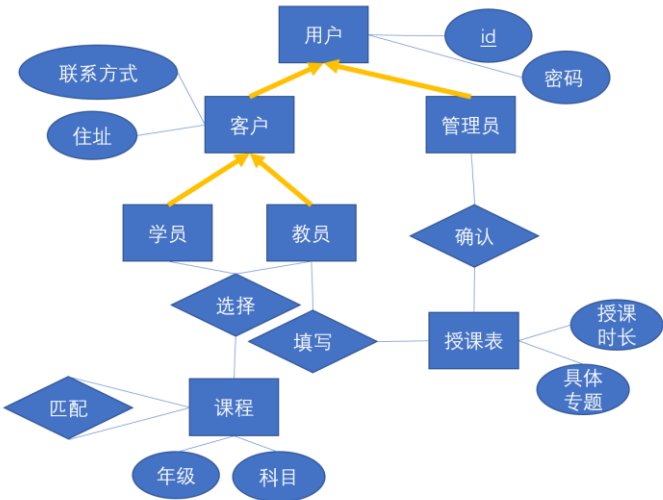


图 11：产品需求 ER 图

#### 3.2 关键数据表设计，对标各大模块功能

在 2.2 节中，具体介绍了各大模块的功能，本节主要结合 3.1 节的 ER 图与实际功能需求设计数据表，并在下一节中详细介绍数据表在一些关键功能实现上的应用。

由于用户注册与登录模块涉及的数据表比较简单，故不再进行赘述，本节具体介绍学员教员与匹配模块数据表的设计（管理员模块使用的表都在这三个模块中）。

##### 3.2.1 学员模块

首先每一个学员都有自己的个人信息，故需要设计一张**学员个人信息表**，然后每个学员都可以选择多门补习课程，故再抽象出一张**学员课程表**。

个人信息表的设计大致如下：

学员表			
字段名	字段含义	数据类型	约束
学员id	区分学员	整数	主键、自增
姓名	学员真实姓名	字符串	无
性别	学员性别	字符串	无
就读学校	学员就读学校	字符串	无
年级段	学员当前年级段（小学初中等）	字符串	无
就读年级	实际就读年级（如高二升高三等）	字符串	无
手机号	顾名思义	字符串	11位数字
qq号		字符串	无
微信号		字符串	无
家长手机号		字符串	11位数字
住址地区	住址大概地区（如榕城区等）	字符串	无
详细住址	学员详细住址，详细到小区或村	字符串	无

图 12：学院表

对标学员填写个人信息模块，学员填写完个人信息之后，点击提交按钮提交数据到后台，后台进行正确性判断（例如：手机号为 11 位数字等）。若数据为空或不符合要求，返回错误信息到前端并提示学员进行修改，若数据符合要求，则保存进数据库中。

由于每一个学员可以选择多门课程，而课程又不需要单独存在，可以把学员课程表看成一个弱实体，依赖于学员存在。学员课程表的设计大致如下：

学员课程表			
字段名	字段含义	数据类型	约束
课程id	用于区分不同课程	整数	主键、自增
年级	学员所在年级段	字符串	无
科目	学员所选择科目	字符串	无
成绩水平	学员所选课程的目前水平	整数	0-100
目标分数	学员期望的目标分数	整数	0-100
是否已被匹配	判断该课程是否被匹配	Bool值	无
学员id	选择课程的学员	整数	外键

图 13：学员课程表

对标学员个人信息模块，学员个人信息中包含学员选择的希望被教授的课程，在之后可以根据课程信息与教员学员信息进行匹配。在学员填写完课程信息选择添加课程之后，将数据传输到后台进行正确性判断（成绩水平与目标分数为范围内的整数等）。若正确则保存进数据库中，不正确则提示学员进行修改。

### 3.2.2 教员模块

针对教员模块设计的表与学员模块十分类似，只有部分地方不同，也是包括**教员个人信息表**与**教员课程表**。出于之后功能查找的方便（不用多表查询），没有把教员学员相同的表项合并起来，教员个人信息表如下所示：

教员表			
字段名	字段含义	数据类型	约束
教员id	区分教员	整数	主键、自增
姓名	教员真实姓名	字符串	无
性别	教员性别	字符串	无
毕业学校	教员毕业学校	字符串	无
目前就读学校	教员目前就读学校	字符串	无
可接受的学生阶段	高中、初中、小学	字符串	无
手机号	顾名思义	字符串	11位数字
qq号		字符串	无
微信号		字符串	无
家长手机号		字符串	11位数字
住址地区	住址大概地区（如榕城区等）	字符串	无
详细住址	学员详细住址，详细到小区或村	字符串	无
教学科目	具体教学科目，可选择多项	字符串	无
是否能接受基础补习	是否能接受未选科目的基础补习	bool	无
备注	如对教员的要求等等	字符串	无

图 14：教员表

教员课程表除了没有成绩水平之外，其他表项都与学员课程表一致，篇幅问题不再展示。

### 3.2.3 匹配模块

匹配过程将在下一节进行详细介绍。匹配成功之后，需要使用一个**匹配表**记录当前存在的匹配信息，匹配表的设计如下所示：

匹配表			
字段名	字段含义	数据类型	约束
教员id	此匹配对应的教员	整数	无
学员id	此匹配对应的学员	整数	无
学员课程id	学员所选课程	整数	无
距离	学员与教员住址之间的距离	浮点数	无
状态	该匹配的状态（待确认、已确认等）	整数	无
提成金额	该课程结课后教员需要提交的提成	浮点数	无

图 15：匹配表

通过**教员个人信息表**、**教员课程表**、**学员个人信息表**和**学员课程表**的信息进行匹配后，得到的匹配信息保存于数据库中，每一条匹配信息有以下五个状态：

```
(1, "待确认"),
(2, "待学员确认"),
(3, "已确认"),
(4, "待提交提成"),
(5, "已提交提成"),
```

图 16：匹配信息状态

刚匹配完的状态为 1（待教员确认），若教员取消该匹配或者教员确认之后学员取消该匹配，则该匹配对应的学员课程与对应的教员以后将不能匹配，故需要一个**失配表**保存这个信息，包括学员课程 id 与教员 id 两个表项。

当教员与学员都确认该匹配时，匹配信息进入授课状态（即状态 3），此时需要一个**教员授课表**记录教员授课信息以便最后结课时提成的计算。**教员授课表**的设计如下：

授课表			
字段名	字段含义	数据类型	约束
教员id	此匹配对应的教员	整数	无
学员id	此匹配对应的学员	整数	无
学员课程id	学员所选课程	整数	无
科目	授课的具体科目	字符串	无
具体专题	具体授课专题	字符串	无
辅导时长	授课时长	浮点数	无
辅导日期	具体授课日期	字符串	无
状态	待学员确认和已确认两个状态	整数	无

图 17：教员授课表

授课表由教员根据授课情况填写，再经由学员确认，最后用于计算该教员的应缴提成。

### 3.2.4 管理员模块

管理员模块对数据表的依赖如下：

学员列表的展示依赖于**学员个人信息表**与**学员课程表**，教员列表的展示依赖于**教员个人信息表**与**教员课程表**，匹配信息展示与结课信息展示依赖于**匹配表**，还存在一个财务账单模块，该模块需要在数据库中建一张账单表，比较简单不再赘述。

## 3.3 一些关键功能的具体实现与代码片段

Django 提供的视图函数功能使得代码框架与实现思路十分明确，总体思路就是，前端编写 HTML 代码，后端编写视图函数功能，通过字典或列表的方式返回前端所需的数据。前端接收到后端返回的数据之后，使用 django 的**模板语言**进行数据的展示。关于 django 的模板语言限于篇幅不再详细介绍。

### 3.3.1 用户注册与登录界面

用户注册与登录界面的实现相对比较简单，故这里不再赘述。主要讲一下登录界面的**图片验证码**实现与**cookie 保存机制**的实现。

**图片验证码**的实现思路是，在前端页面的图片验证码处，把<img>标签的 src 属性写为一个 url，然后把 url 连接到视图函数 image\_code 中，之后在函数中随机生成字符串与图片之后返回到前端页面上即可。

```

```

图 18：图片验证码<img>标签



```
def image_code(request):
    img, code_string = check_code()

    request.session['code'] = code_string
    request.session.set_expiry(60)

    stream = BytesIO()
    img.save(stream, 'png')

    return HttpResponse(stream.getvalue())
```

图 19：生成图片验证码对应的视图函数

Image\_code 函数中的 check\_code 函数是自己参考 django 教程[1]编写的一个函数，功能是随机生成字符串 code\_string 与使用 PIL 库中的函数生成字符串对应的图片 img。

在用户登录成功之后，浏览器可以把此时登录用户的用户名与 id 作为该用户的 cookie 保存起来，用于验证用户身份以分配页面访问权限，如下图所示：

```
# 登录成功
request.session['info'] = {'id': account_obj.id, 'username': account_obj.username, 'type': account_obj.type}
# session 保存七天 七天免登录
request.session.set_expiry(3600 * 24 * 7)
```

图 20：保存用户的 cookie

分配页面访问权限的功能可以使用 django 提供的中间件技术实现，中间件技术讲的就是，在用户对网页发起请求之前，需要先通过中间件的检测，此时可以通过 cookie 判断该用户的类型，比如说学员用户不能访问教员用户的网页，那么如果此时用户的 cookie 中的 'type' 字段（保存用户类型）为学员，其访问的页面为教员的个人信息页面，中间件把这个请求重定向后学员的页面。中间件的实现大致如下：

```
from django.utils.deprecation import MiddlewareMixin
from django.shortcuts import render, redirect

## settings.py 配置中间件

## 中间件可以实现登录的校验，可以避免重复代码

class AuthMiddleware(MiddlewareMixin):
    def process_request(self, request):
        #0. 排除不需要登录就可以访问的页面
        if (request.path_info in ['/login/', '/image/code/', '/register/', '/logout/']):
            return None

        #1. 读取当前用户的session信息，如果能读到，继续向后走
        info_dict = request.session.get('info')
        if info_dict:
            path_info = request.path_info
            type = info_dict['type']
            if type == 1:
                return None
            elif type == 2:
                if path_info[0:8] == '/learner':
                    return None
                else:
                    return redirect('/learner/index/')
            else:
                if path_info[0:6] == '/tutor':
                    return None
                else:
                    return redirect('/tutor/index/')

        # 如果方法中没有返回值，继续往后走
        # 如果有返回值，直接返回
        return redirect('/login/')
```

图 21：中间件实现用户校验

### 3.3.2 学员页面

个人信息页面采用 django 提供的 modelform 进行编写，就是编写学员个人信息相关的 form 类，该类的对象会自动生成 HTML 的<input>框，可以将该类的对象传到前端中，前端使用模板语言把对应的<input>框展示出来，省去了编写<input>框的工作。

```
class LearnerInfoModelForm(BootstrapModelForm):
    class Meta:
        model = models.LearnerInfo
        fields = '__all__'
        widgets = {
            'detail_address': forms.TextInput(attrs={'placeholder': 'xx市xx区xx街道xx小区'}),
            'time': forms.TextInput(attrs={'placeholder': '周末/暑假/可协商'})
        }

class LearnerCourseModelForm(BootstrapModelForm):
    class Meta:
        model = models.LearnerCourseInfo
        fields = '__all__'
```

图 22：学员对应的 modelform 类

```
def userinfo(request):
    account_id = request.session['info']['id']
    queryset = models.LearnerInfo.objects.filter(user=str(account_id))
    if not queryset:
        form = LearnerInfoModelForm()
        Courseform = LearnerCourseModelForm()
        course_queryset = models.LearnerCourseInfo.objects.filter(user=str(account_id))
        context = {
            'form': form,
            'Courseform': Courseform,
            'course_queryset': course_queryset,
        }
    return render(request, 'learner/learnerinfo.html', context)
```

图 23：后端把 modelform 对象传到前端进行展示

```
<label for="exampleFormControlTextarea1">姓名</label>
{{ form.name }}
<span class="error-msg" style="..."></span>

</div>
<div class="form-group col-md-2">

    <label for="exampleFormControlTextarea1">性别</label>
    {{ form.gender }}
    <span class="error-msg" style="..."></span>

</div>
<div class="form-group col-md-3">

    <label for="exampleFormControlTextarea1">就读学校</label>
    {{ form.school }}
    <span class="error-msg" style="..."></span>

</div>
```

图 24：前端使用模板语言展示自动生成的 input 框

在个人信息填写完毕时，我们使用 jquery 中提供的 ajax 数据发送方法，通过 post 方式把数据发送到后台进行校验。具体的 js 代码如下：

```

function bindBtnEvent(){
    $('#btn1').click(function(){
        $(".error-msg").empty();
        $.ajax({
            url: '/Learner/userinfo/update/',
            type: 'post',
            data: $('#learnform').serialize(),
            dataType: 'JSON',
            success: function(res)
            {
                if (res.status){
                    alert('提交信息成功! ');
                    location.reload();
                }
                else
                {
                    $.each(res.error, function(name, data){
                        if (name !== 'more')
                        {
                            $("#id_" + name).next().text(data[0]);
                        }
                    })
                }
            }
        })
    })
}

```

图 25: 使用 js 代码发送数据到后台

后台接到数据之后，对数据进行校验，若数据校验通过，则可以通过 modelform 对象的 save 函数直接将数据保存到数据库中，并返回正确信息给前端。若数据校验不通过，则返回错误信息给前端。前端根据后台返回的信息判断数据是否异常，若异常则添加错误信息提示到具体的 input 框下方。

学员个人信息中，提供了选择课程与删除课程功能，我们把它抽象为两个视图函数去实现，独立于学员其他信息之外。具体实现起来与个人信息的填写差别不大，故不再赘述。

```

@csrf_exempt
def updateCourse(request):...

@csrf_exempt
def deleteCourse(request):...

```

图 26: 选择课程与删除课程对应的视图函数

学员页面还提供匹配信息展示与确认功能，分别对应了两个视图函数，如下所示：

```

def matchinfo(request):
    account_id = request.session['info']['id']
    Learner = models.LearnerInfo.objects.filter(user=str(account_id)).first()

    MatchQueryset = models.MatchInfo.objects.filter(Learner=Learner).filter(Q(State=2)|Q(State=3)|Q(State=4)|Q(State=5))

    content = {
        'Match': MatchQueryset,
        'MatchNum': len(MatchQueryset),
    }

    return render(request, 'learner/matchinfo.html', content)

@csrf_exempt
def confirmMatch(request):
    data_dict = {'status': True}

    matchid = request.POST['id']

    models.MatchInfo.objects.filter(id=matchid).update(State=3)

    return JsonResponse(data_dict)

```

图 27: 匹配信息展示与确认视图函数

匹配信息展示的视图函数中，首先利用此时的用户 id 从数据库中获取学员信息，再通

过学员 id 获取该学员对应的所有匹配信息，并返回至前端进行展示。匹配信息确认的视图函数中，就是利用前端传过来的匹配信息 id，对该匹配信息的状态进行更新，从待学员确认状态变为已确认状态。

对于匹配信息删除的视图函数，也比较简单，故不再说明。

学员页面的最后一个功能是授课信息展示与确认删除授课信息，分别对应三个视图函数，授课信息展示对应的视图函数如下：

```
def coursetable(request):
    account_id = request.session['info']['id']
    Learner = models.LearnerInfo.objects.filter(user=str(account_id)).first()

    MatchQueryset = models.MatchInfo.objects.filter(Learner=Learner).filter(Q(State=3) | Q(State=4) | Q(State=5))
    MatchList = []
    for match in MatchQueryset:
        Matchobj = {}
        Matchobj['match'] = match
        Matchobj['Unconfirmed'] = models.TeachTableInfo.objects.filter(Learner=match.Learner, Course=match.Course, state=1)
        Matchobj['Confirmed'] = models.TeachTableInfo.objects.filter(Learner=match.Learner, Course=match.Course, state=2)
        MatchList.append(Matchobj)

    content = {
        'Match': MatchList,
        'MatchNum': len(MatchList),
    }

    return render(request, 'learner/coursetable.html', content)
```

图 28：授课信息展示

由于学员可能有许多匹配信息，而对于每一条匹配信息，又存在许多对应的授课信息，故我们需要对其进行整合，我使用列表套字典，字典再套列表的方式进行整合，把匹配信息和对应的授课信息存放在一个字典里，对应的授课信息存放在一个列表里。由于确认授课信息和删除授课信息与确认匹配信息类似，故不再赘述。至此，学员页面的所有功能已叙述完毕。

### 3.3.3 教员页面

教员的许多页面都与学员类似，故不再赘述。值得一提的是教员授课表展示页面的结课按钮。点击结课按钮之后，系统会根据教员所教授科目和总时长计算应提交的提成，如下图所示：

```
@csrf_exempt
def finishcourse(request):
    data_dict = {'status': True}

    matchobj = models.MatchInfo.objects.filter(id=request.POST['id']).first()
    UnconfirmedTeachQueryset = models.TeachTableInfo.objects.filter(Tutor=matchobj.Tutor, Learner=matchobj.Learner, state=1)

    if (len(UnconfirmedTeachQueryset)):
        data_dict = {'status': False}
    else:
        models.MatchInfo.objects.filter(id=matchobj.id).update(State=4)
        TeachQueryset = models.TeachTableInfo.objects.filter(Tutor=matchobj.Tutor, Learner=matchobj.Learner, state=2)
        Amount = 0
        primaryBase = 50
        juniorBase = 50
        highBase = 55
        for teach in TeachQueryset:
            if teach.Course.education == '高中':
                Amount = Amount + highBase * teach.tutortime
            elif teach.Course.education == '初中':
                Amount = Amount + juniorBase * teach.tutortime
            else:
                Amount = Amount + primaryBase * teach.tutortime
        models.MatchInfo.objects.filter(id=matchobj.id).update(Amount=Amount * 0.05)

    return JsonResponse(data_dict)
```

图 29：结课视图函数

### 3.3.4 管理员页面

管理员页面首先是学员信息与教员信息的展示，直接从数据库中提取信息后包装好发送到前端展示即可，这里值得一提的是包装了一个分页组件，可以用于前端分页展示。并且拥有查询功能。

```
def userinfo_learner(request):  
    data_dict = {}  
    value = request.GET.get('q', '')  
    if value:  
        if value.isdigit():  
            data_dict["id__contains"] = value  
        else:  
            data_dict["name__contains"] = value  
  
    queryset = models.LearnerInfo.objects.filter(**data_dict)  
  
    page_obj = Pagination(request, queryset, page_size=5)  
  
    context = {  
        "value": value,  
        "queryset": page_obj.page_queryset,  
        "page_string": page_obj.html()  
    }  
  
    return render(request, 'admin/userinfo_learner.html', context)
```

图 30：管理员学员信息展示

之后是展示当前未匹配信息，也是比较简单，从数据库查询数据之后传到前端展示。其中，提供了为当前未匹配信息根据教员 id 添加一个匹配信息。

```
def match(request):  
    CourseQueryset = models.LearnerCourseInfo.objects.filter(ismatch=False)  
    Course = []  
  
    for course in CourseQueryset:  
        courseobj = {}  
        Learner = models.LearnerInfo.objects.filter(user=course.user).first()  
        courseobj['Learner'] = Learner  
        courseobj['course'] = course  
        Course.append(courseobj)  
  
    content = {  
        'Course': Course,  
    }  
  
    return render(request, 'admin/match.html', content)
```

图 31：未匹配信息展示

添加匹配信息，就是根据教员 id，学员 id，课程 id，在后台为数据库表添加一条数据即可：

```

@csrf_exempt
def matchcourse(request):
    data_dict = {'status': True}

    TutorQueryset = models.TutorInfo.objects.filter(id=request.POST['tutorid'])

    if not TutorQueryset:
        data_dict['status'] = False
    else:
        Tutor = TutorQueryset.first()
        Course = models.LearnerCourseInfo.objects.filter(id=request.POST['courseid']).first()
        Learner = models.LearnerInfo.objects.filter(user=Course.user).first()
        dist = Dist(Tutor.detail_address, Learner.detail_address)
        startDate = datetime.datetime.now()
        delta = datetime.timedelta(days=3)
        endDate = (startDate + delta).date()
        startDate = startDate.date()
        models.MatchInfo.objects.create(Tutor=Tutor, Course=Course, Learner=Learner, dist=dist, StartDate=startDate, EndDate=endDate, State=1)
        models.LearnerCourseInfo.objects.filter(id=Course.id).update(ismatch=True)
        models.TutorInfo.objects.filter(id=Tutor.id).update(match_number=Tutor.match_number+1)

    return JsonResponse(data_dict)

```

图 32：添加匹配信息

财务报表的功能实现起来比较简单，这里就不赘述了。

### 3.3.5 匹配函数

匹配函数是本次项目的核心功能，后台进行定时匹配，也就是利用代码定时调用匹配函数进行匹配，具体流程如下：

匹配过程：

1. 获取教员信息（优先考虑当前匹配信息少的教员）与学员信息
2. 教员信息有教员id、教员所选科目（包括该科目能够接受的年级段）、教员详细位置
3. 学员信息有学员id、学员所选科目（包括该科目的具体年级）、学员详细位置
4. 按照以下方式进行匹配
  1. 按照id枚举学员的所有所选科目（包括该科目的具体年级段）
  2. 根据当前学员的当前科目，找到所有满足科目要求的且仍要求推荐学员的教员
  3. 对所有满足科目要求的教员，计算他们与该学员详细位置的距离
  4. 对所有满足科目要求的教员，按照已匹配信息数量（包括已成功和待确认的匹配信息）作为第一关键字（升序），距离作为第二关键字排序（升序），选择排序后的第一个教员

图 33：匹配流程

其中，计算两个地址间的距离调用了百度地图进行计算。在为某一条课程记录匹配到一个教员之后，给该教员发送一条邮箱信息提醒该教员及时与已匹配上的学员联系。

```

# 发邮件
qqEmail = str(FinalTutor.qq_number + "@qq.com")
content = "您有一条成功匹配的信息！！请您登录云汉教育网站，查看匹配界面。\\n若您认为此学员的年级，科目，成绩水平，住址等信息不符合您的预期，请尽快删除此条匹配记录，以便系统继续为您匹配更加合适的学员。若您双方中一人删除此次匹配信息，系统会自动删除匹配信息并为您匹配新的教员。\\n"
sendEmail(qqEmail, content)


```

图 34：匹配成功后为教员发送邮箱

## 4 项目效果图

### 4.1 注册与登录界面效果展示

用户可根据自己的身份选择注册方式，可进行学员注册或教员注册。



The image shows a registration form for students. At the top, there is a user icon and two tabs: '学员注册' (Student Registration) and '教员注册' (Teacher Registration). The form contains five input fields: '用户名(不超过20个字符)' (Username, max 20 characters), '密码' (Password), '确认密码' (Confirm Password), 'QQ邮箱' (QQ Email), and a '提交' (Submit) button. Below the button is a link: '已有账号? 登录' (Already have an account? Login).


图 35: 学员注册



The image shows a registration form for teachers. It has the same layout as the student registration form, with tabs for '学员注册' (Student Registration) and '教员注册' (Teacher Registration). The input fields are: '用户名(不超过20个字符)' (Username, max 20 characters), '密码' (Password), '确认密码' (Confirm Password), 'QQ邮箱' (QQ Email), and a '提交' (Submit) button. A link '已有账号? 登录' (Already have an account? Login) is at the bottom.

图 36: 教员注册

学员, 教员, 管理员均用以下登录界面登录。



The image shows a login form. At the top, there is a lock icon and the title '登录' (Login). The form has three input fields: 'admin' (username), '\*\*\*\*\*' (password), and 'ENRRQ|' (a field with a QR code icon). Below the fields is a '确定' (Confirm) button. At the bottom, there is a link: '没有账号? 注册' (No account? Register).

图 37: 登录界面

在注册和登录会有一些非空判断与格式判断：如账号不应和已有账号重复，密码应多于6位，密码和二次确认的密码必须一致，所有条目不可为空等等，部分测试如下：

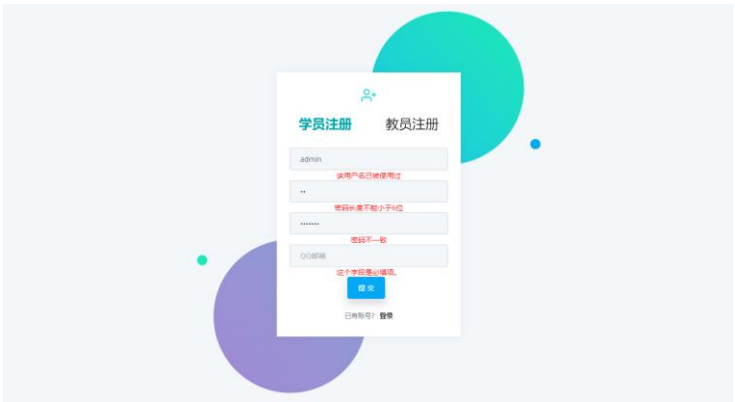


图 38：学员注册异常测试

4.2 教员界面效果展示

教员主页

上方是轮播宣传图，下方是云汉编委职务和交流群。



图 39：教员主页 1



图 40：教员主页 2



个人信息编辑

个人信息

ID: 10

基本信息

我们不会泄露任何关于您的个人基本信息!

姓名

陈敬涵

性别

女

毕业学校

揭阳第一中学

目前学校

深圳大学

大致住址

榕城

详细地址

揭阳市榕城区仙女镇

QQ号

1911688012

微信号

dsfasf

手机号

15813833265

图 41：教员个人信息填写

课程信息

添加教学科目

科目

语文

可接受教学年级:

☐ 小学

☐ 初中

☐ 高一

☐ 高二

☐ 高三

添加

已添加教学科目

科目	年级范围	操作
语文	小学	删除
化学	初中 高二	删除
科学	小学	删除
地理	高一 高二 高三	删除

图 42：教员课程信息填写

匹配信息确认

根据教员地理可教授高一高二高三的课程信息为其匹配了一个学员的地理科目，同时展示其部分个人信息。

匹配信息

陈敏涵 高二地理

基本信息

姓名

性别

就读学校

大致住址

相距约为

陈敏涵

女

揭阳第一中学

榕城

7.82km

QQ号

微信号

2785355983

dsfasf

手机号

家长手机号

15813533265

13924444291

需要辅导科目

年级	科目	水平	目标
高二	地理	34	90

图 43：匹配信息展示

教员可选择确认或删除，删除后不会再为教员匹配此条信息，确认则可联系学员进一步讨论是否继续进行此科目教学工作，需要学员确认后才可开展教学。

需要辅导科目

年级	科目	水平	目标
高二	地理	34	90

备注

none

确认此匹配

删除此匹配

图 44：匹配信息确认或删除

需要辅导科目

年级	科目	水平	目标
高二	地理	34	90

备注

none

待学员确认

图 45：匹配信息确认后状态改变

若教员不想系统继续推荐，可点击“不再继续推荐学员”，若教员想继续获得新的学员，可点击“继续为我推荐学员”。

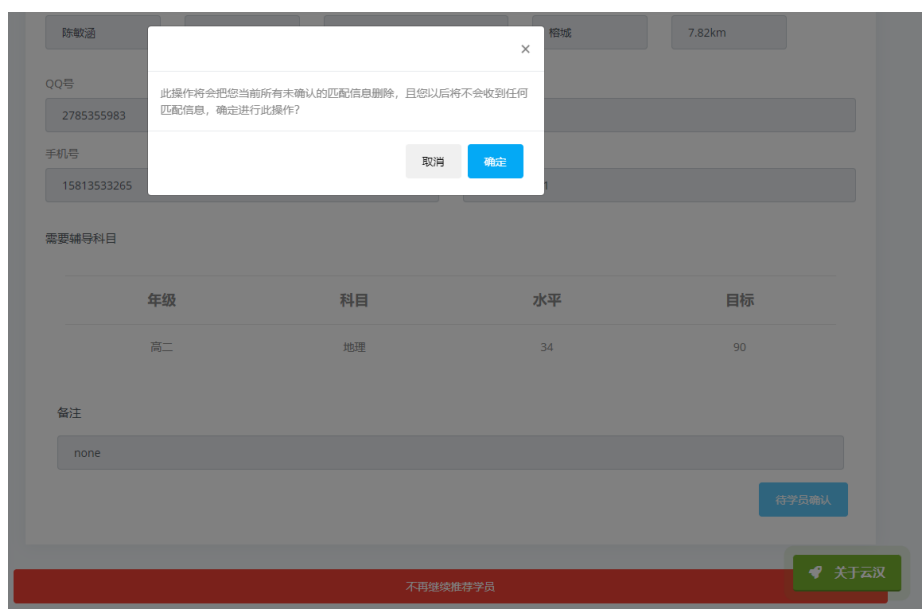


图 46：教员点击不再推荐学员

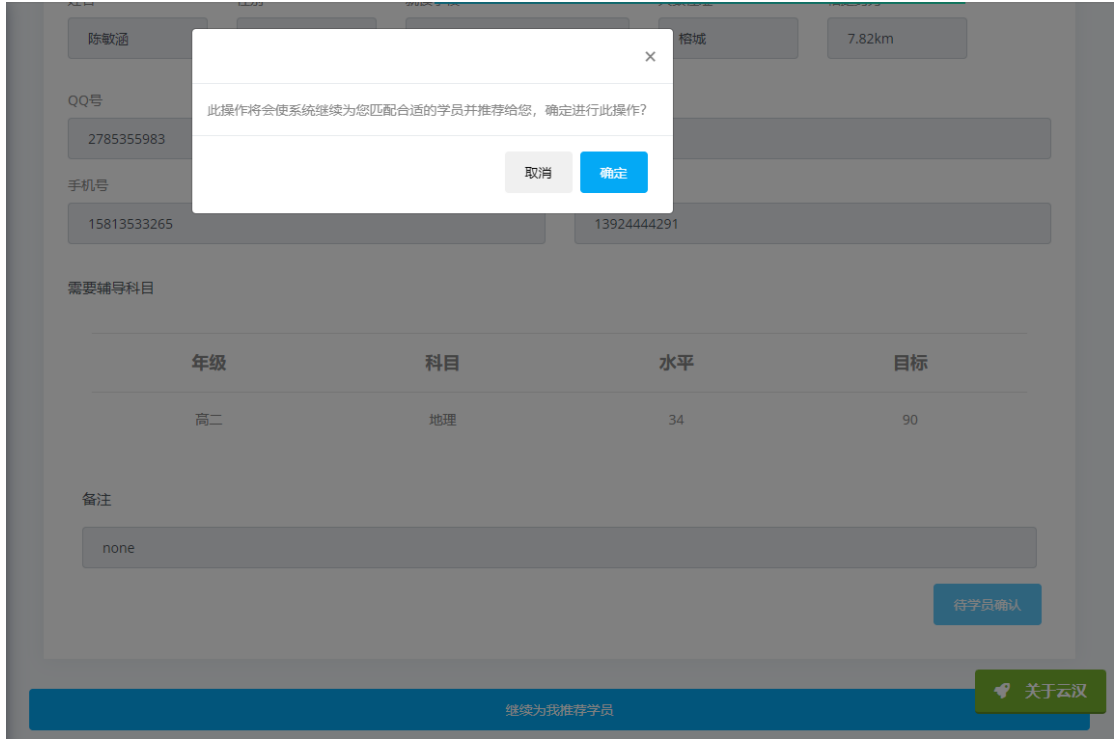


图 47：教员点击继续推荐学员

## 授课表登记

在学员也进行匹配信息确认之后开始授课，授课结束后可生成一条授课信息。

添加教学记录

科目

具体专题

辅导时长/小时

辅导日期

添加

待确认教学记录

科目	具体专题	辅导时长	辅导日期	操作
地理	东北平原	1		

已添加教学记录

图 48：教员添加授课信息

添加的授课信息需要等待学员确认。

添加教学记录

科目

具体专题

辅导时长/小时

辅导日期

添加

待确认教学记录

科目	具体专题	辅导时长	辅导日期	操作
地理	东北平原	1.0	2022-06-07	删除

已添加教学记录

确认结课

图 49：授课信息待确认

待确认教学记录

科目	具体专题	辅导时长	辅导日期	操作
----	------	------	------	----

已添加教学记录

科目	具体专题	辅导时长	辅导日期	操作
地理	东北平原	1.0	2022-06-07	删除

确认结课

图 50：授课信息已确认

### 4.3 学员界面效果展示

学员的个人信息编辑，首页界面等均和教员个人信息相似，在此不再重复展示。

#### 确认匹配界面

教员陈敏涵 辅导您的高二地理

基本信息

姓名

陈敏涵

性别

女

毕业学校

揭阳第一中学

目前学校

深圳大学

QQ号

1911688012

微信号

dsfasf

手机号

15813533265

确认此匹配

删除此匹配

关于云汉

图 51：学员确认或删除匹配

#### 授课时长确认界面

教员陈敏涵 辅导您的高二地理

待确认教学记录

科目	具体专题	辅导时长	辅导日期	操作
地理	东北平原	1.0	2022-06-07	<div><div>确认</div><div>删除</div></div>

图 52：学员授课时长确认

## 4.4 管理员界面效果展示

### 学员/教员管理界面

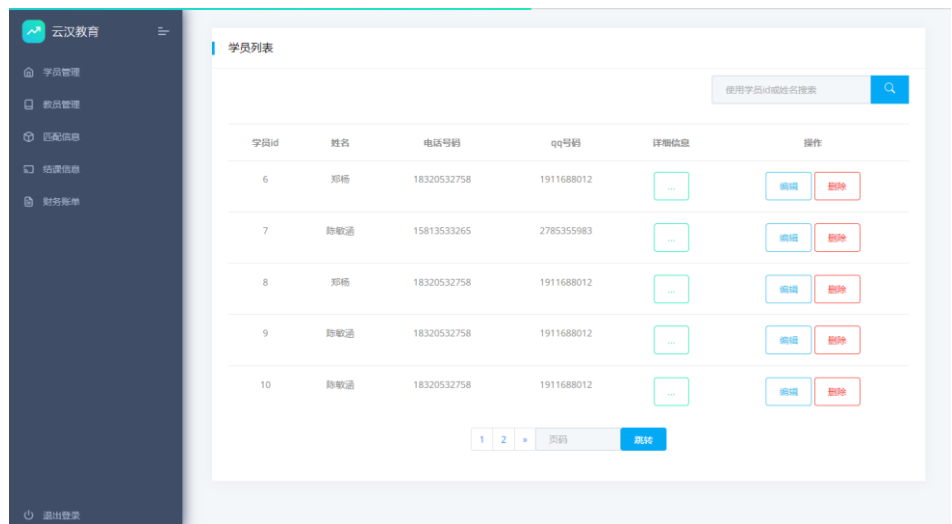


图 53: 学员/教员管理界面

点击学员的详细信息可获得下图，所有信息不可编辑，仅供查看。

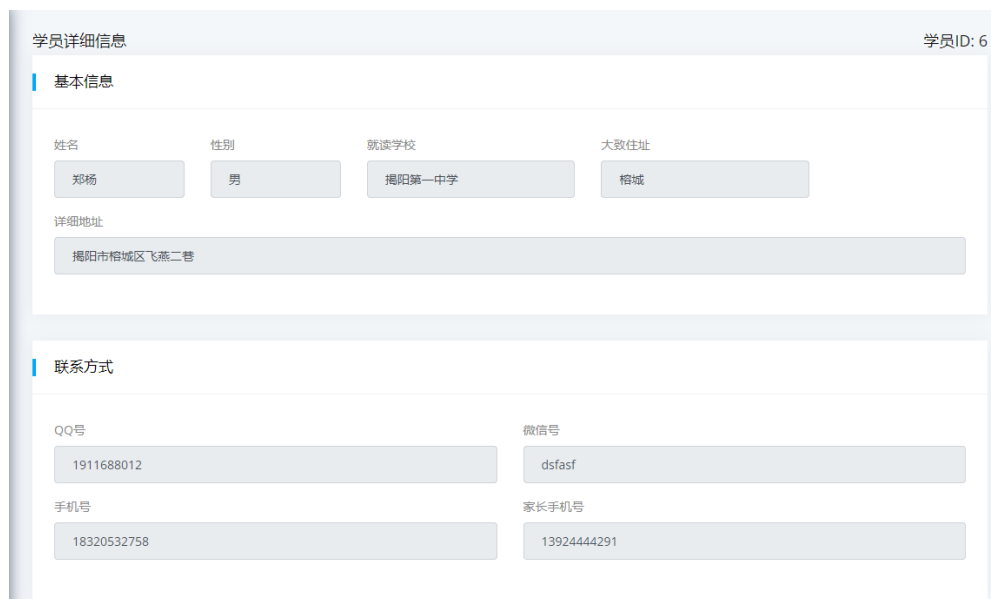


图 54: 学员详细信息

若点击“编辑”键即可编辑学员信息。

个人信息——郑杨

学员ID: 6

基本信息

姓名

性别

就读学校

大致住址

郑杨

男

揭阳第一中学

榕城

详细地址

揭阳市榕城区飞燕二巷

联系方式

QQ号

微信号

1911688012

dsfasf

手机号

家长手机号

18320532758

13924444291

图 55：学员信息编辑

点击“删除”键会跳出一个提示框，确认是否删除，本系统其他所有删除键均会弹出如图的提示框，以后遇到则不再赘述。

学员列表

确认删除此学员的所有信息?

取消 确定

使用学员id或姓名搜索

Q

学员id	姓名	电话号码	qq号码	详细信息	操作
6	郑杨	18320532758	1911688012	...	编辑 删除
7	陈敏涵	15813533265	2785355983	...	编辑 删除
8	郑杨	18320532758	1911688012	...	编辑 删除
9	陈敏涵	18320532758	1911688012	...	编辑 删除
10	陈敏涵	18320532758	1911688012	...	编辑 删除

1 2 »

页码

跳转

图 56：删除学员信息

可根据 ID 和姓名对学员进行信息筛选。

学员列表

7

Q

学员id	姓名	电话号码	qq号码	详细信息	操作
7	陈敏涵	15813533265	2785355983	...	编辑 删除

1

页码

跳转

图 57：根据学员 id 搜索学员信息

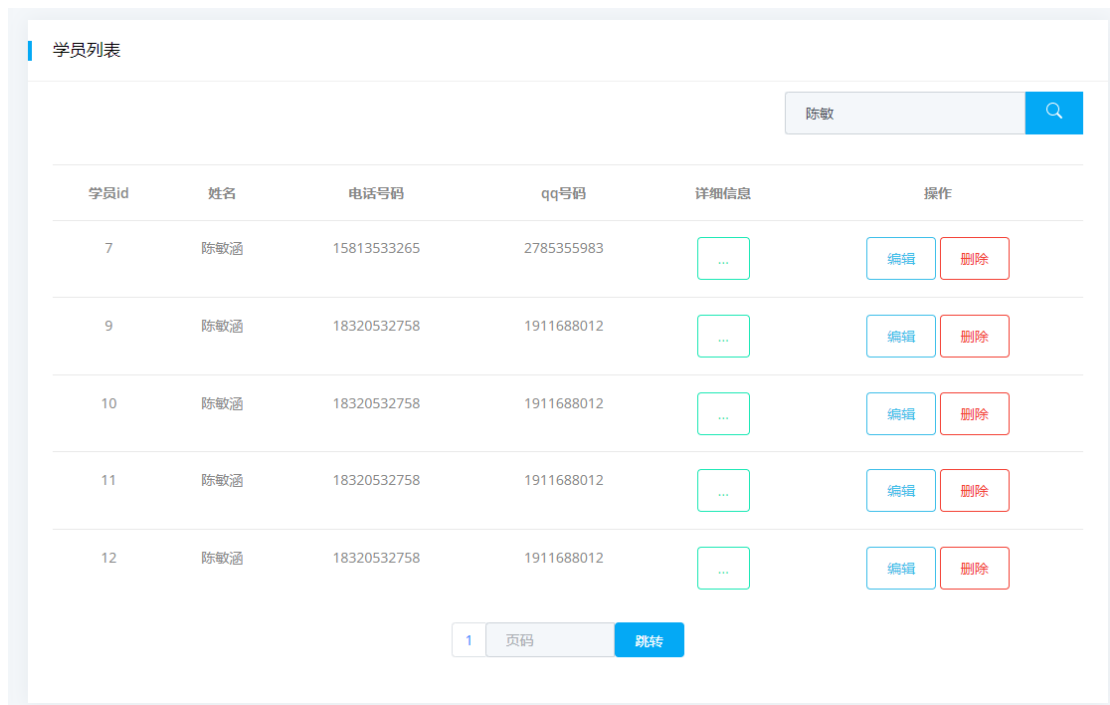


图 58：根据学员姓名搜索学员信息  
根据页码进行页码跳转。

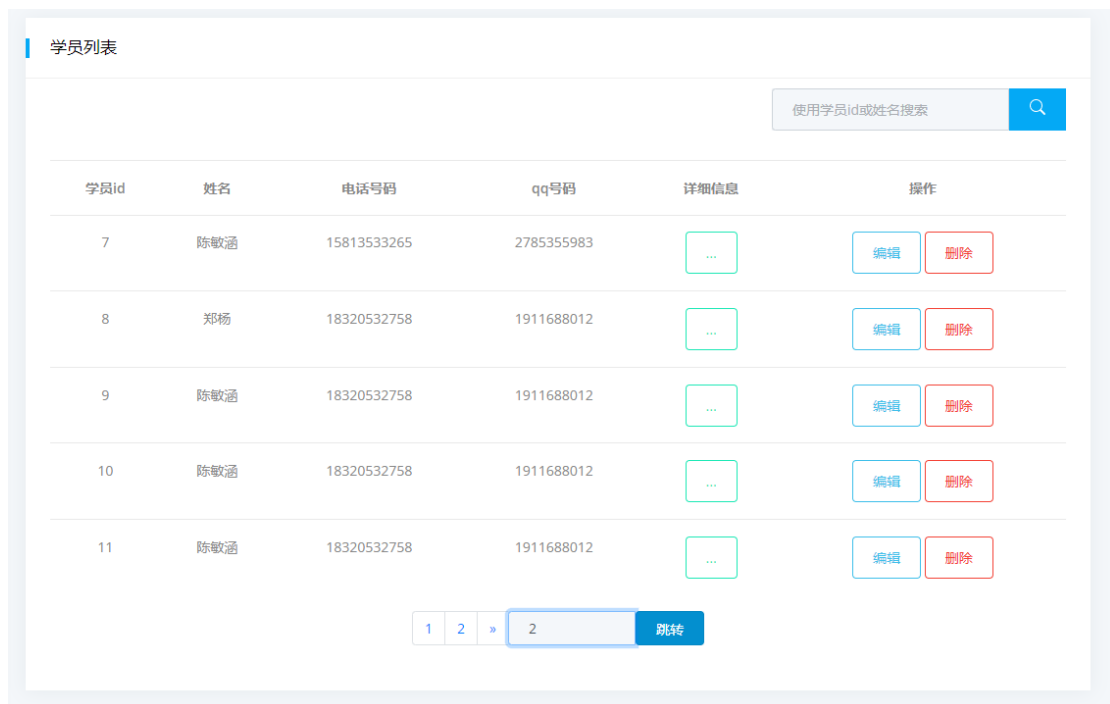


图 59：页码跳转  
教员管理界面和学员管理界面类似，此处不再赘述。  
教员管理与学员管理大致相似，仅增加了一个“新增邀请码”的功能，可自动生成邀请码。



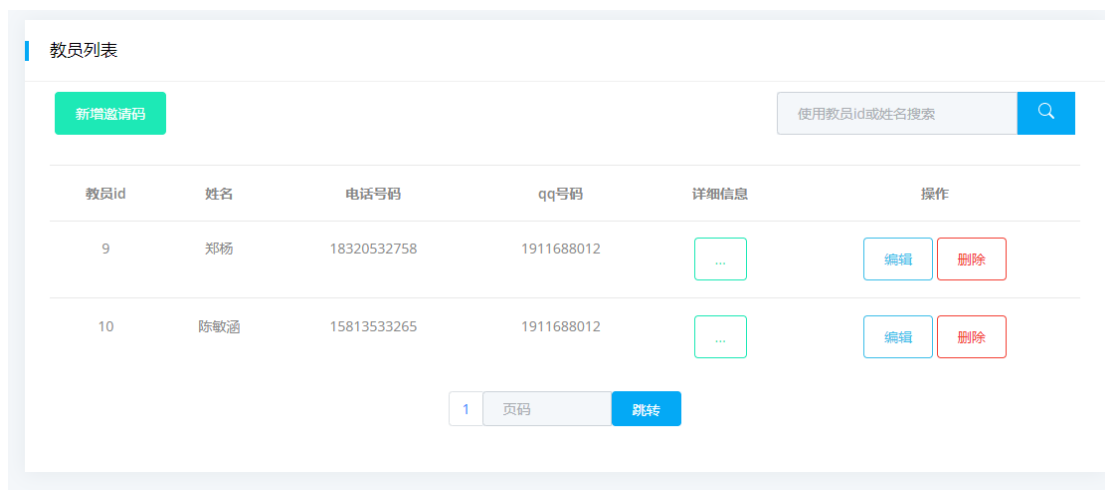


图 60: 教员管理界面



图 61: 生成邀请码

## 匹配信息界面

系统匹配之后会将未匹配学员添加到此处。



图 62: 未匹配课程列表

管理员可根据教员意愿直接指定教员 ID，新生成一个匹配信息。



图 63: 对未匹配学员匹配

若输入教员 ID 不存在，系统会进行提示。

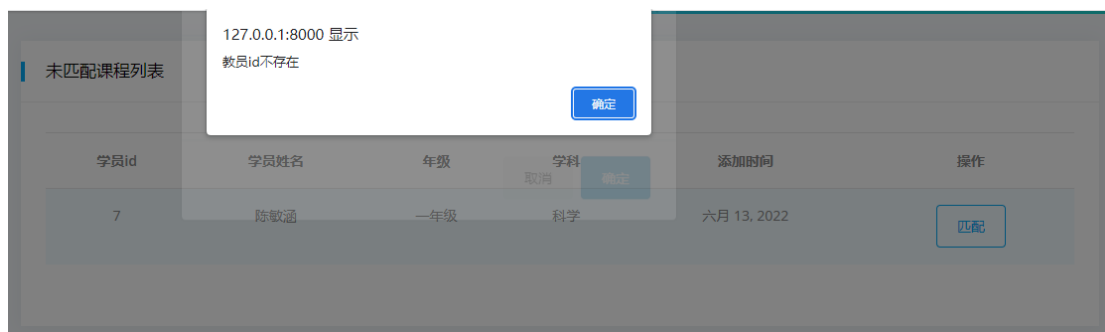


图 64：错误提示

## 结课信息界面

对每条教员确定结课的信息进行确认，管理员确认之后变成已缴费状态。



图 65：结课信息

## 账单信息界面

结课信息生成会直接添加到账单列表。



图 66：账单信息

管理员可新增账单，如宣传单印发等支出。

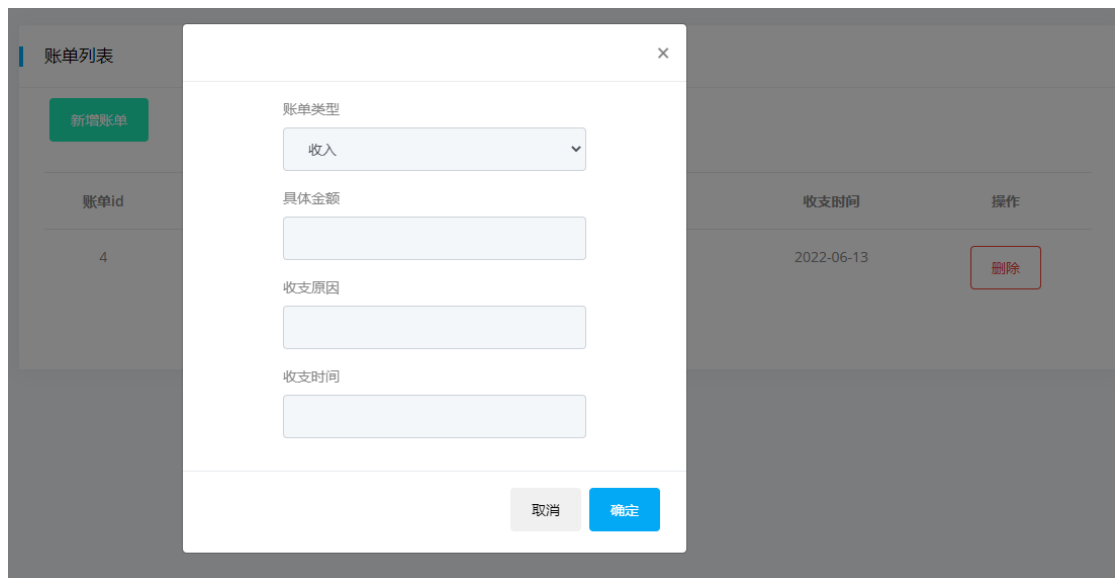


图 67：新增账单

## 5 任务分工及所完成的状况

郑杨主要负责后端视图函数的编写，陈敏涵主要负责前端界面的协调分布，两人共同探索匹配函数的实现与各个数据表的创建以及前后端的连接。

对于 1.4 节提出的所有功能均已完成，个人觉得完成情况挺好的。但是大量的测试还没来得及做，后续再进行测试。

整个系统的功能还没有很完善，后续再进行探究与编码之后考虑将系统上线使用。

## 五、问题分析

1.自动发送信息功能：主要就是三步，登录邮件、写邮件内容、发送，用到的库是 `smtplib` 和 `email`，使用的是 QQ 邮箱，需要设置 QQ 邮箱 POP3/SMTP 服务即可。

2.定时功能：使用 `django-apscheduler` 可支持调度任务：固定时间间隔，固定时间点（日期），`Crontab` 命令。实现定时匹配和发送邮件。

3.计算距离功能：利用百度地图开放平台获取百度地图的调用权限，通过地址名称获得其经纬度，再通过轻量级路线规划服务得到之间的路线规划距离。

## 六、实验心得

通过本次实验，我们从 0 开始学会了一个简单页面的搭建，从 0 开始学习 djangoweb 框架与前端语言，以前 python 语言也没怎么使用过，经过这几个月的努力，最终也顺利完成了整个大作业。个人还是有一点成就感的。

## 七、参考文献

[1] 最新 Python 的 web 开发全家桶（django+前端+数据库），  
[https://www.bilibili.com/video/BV1rT4y1v7uQ?p=27&vd\\_source=04e437cbfcd185dfb751e7b7c1eef7f](https://www.bilibili.com/video/BV1rT4y1v7uQ?p=27&vd_source=04e437cbfcd185dfb751e7b7c1eef7f)

[2] 前端模板使用了 django-datta-able-master 开源代码框架