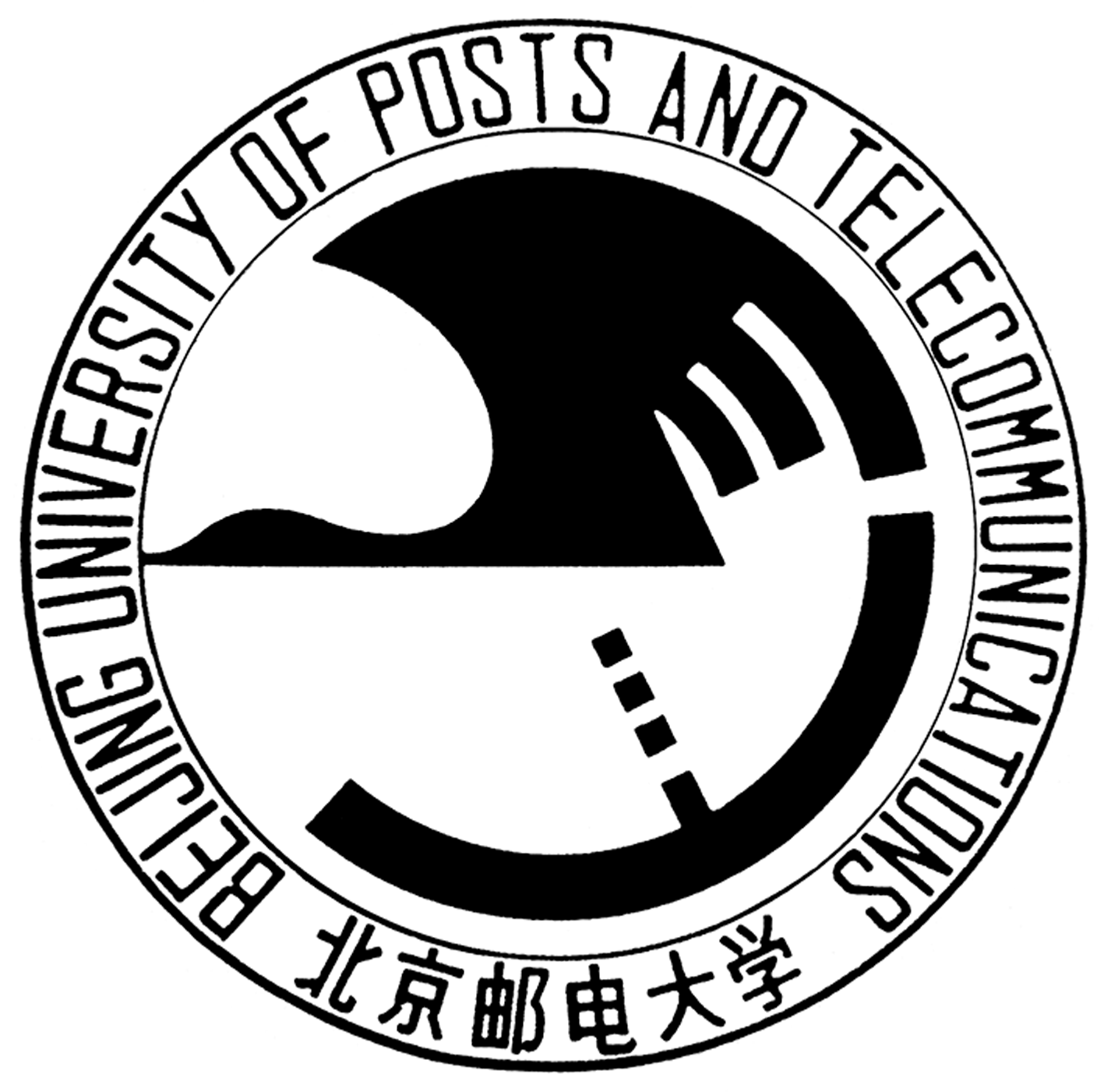
1. **本 科 毕 业 设 计（论文）**



**题目：招生咨询系统的设计与实现-Python客户端子系统**

**姓 名 李智鑫**

**学 院 2016211267**

**专 业 计算机科学与技术**

**班 级 2016211310**

**学 号 2016211267**

**班内序号 10**

**指导教师 杜晓峰**

**2020年 5月**

**招生咨询系统的设计与实现-Python客户端子系统**

摘 要

高校招生咨询工作是高招工作中的重要环节，招生咨询工作可以让学生了解专业特色、招生政策，同时高校也可以第一时间掌握分数分布等重要的相关信息。然而，由于目前缺少有效的电子支撑手段，导致老师工作量大，信息滞后、不准确，宣传口径不一致等问题，严重影响招生咨询工作的效率。本文通过实现招生咨询电子系统中PC端的子系统，为招生咨询工作提供电子化辅助工具，帮助老师提高招生咨询工作的效率。

本文设计的招生咨询系统采用B/S开发模式，Django开发框架，MySQL管理系统的各类数据，使用IIS服务进行部署，编程语言python实现系统后端逻辑，html、JavaScript、CSS实现前端。本文着重论述了系统从需求分析、总体设计、详细设计、系统实现、系统测试这一系列软件开发流程，完整的描述了本系统从设计到实现的全部过程和过程性图表。

目前系统已经能够完成在招生咨询现场辅助老师完成考生信息的收集的存储工作。通过网络传输将数据存入数据库进行汇总。汇总数据可以向各省招生组组长和招生办负责人展示。并且通过一定的算法分析后能够得出分数线和排名关系预估以及录取分数线预估，提供给招生组长作为确定宣传口径的依据。此外，系统还可以展示招生咨询材料，帮助现场的老师进行宣传；招生组组长和招办老师可以根据权限对系统内的数据进行管理。

**关键词** 招生咨询系统 web应用开发 Django Python

**Design and Implementation of Enrollment Consultation System-Python Client Subsystem**

**ABSTRACT**

Enrollment consultation work is an important part of the admissions work, enrollment consultation work can allow students to understand professional characteristics, admissions policy, while universities can also be the first time to gain the distribution of scores and other important relevant information. However, due to the current lack of effective electronic applications of support, leading to a large workload of teachers, information lagging, inaccurate, inconsistent propaganda caliber and other problems, seriously affecting the efficiency of the admissions advisory work. In this paper, through the realization of admissions counseling electronic system in the PC side of the subsystem, for admissions counseling work to provide electronic aids to help teachers improve the efficiency of admissions counseling work.

This paper designed the admissions consulting system using B/S development model, Django development framework, MySQL as management system of all kinds of data, using IIS services for deployment, the programming language are python to achieve system back-end logic and html, JavaScript, CSS to achieve the front-end. This paper focuses on a series of software development processes from requirements analysis, overall design, detailed design, system implementation, and system testing, also provides a complete description of the system from design to implementation and process diagram.

The system is now able to complete the storage of information collected from the candidates at the enrollment consultation site to assist teachers. The data is aggregated in a database through a network transmission. The aggregated data can be presented to the head of the provincial admissions team and the head of the admissions office. The algorithmic analysis of the score line and ranking relationship prediction and admission score line prediction can be used to determine the basis for the head of the admissions team to determine the message. In addition, the system can display enrollment consultation’s materials to help teachers in the field to exhibit; admissions team leaders and teachers in admissions office can manage the data in the system according to permissions.

**KEY WORDS** Enrollment Consultation System Web Application Development Django Python

目录

[第一章 引言 6](#_Toc40481007)

[1.1 课程背景和意义 6](#_Toc40481008)

[1.2 课题任务 2](#_Toc40481009)

[1.2.1 课题内容 2](#_Toc40481010)

[1.2.2 本人承担任务及方案选择 2](#_Toc40481011)

[1.3 论文结构 2](#_Toc40481012)

[第二章 系统的需求分析 4](#_Toc40481013)

[2.1 系统用户角色分析 4](#_Toc40481014)

[2.1.1 普通用户分析 4](#_Toc40481015)

[2.1.2 管理员用户分析 4](#_Toc40481016)

[2.1.3 超级管理员用户分析 4](#_Toc40481017)

[2.2 系统功能需求分析 5](#_Toc40481018)

[2.3 系统非功能需求分析 7](#_Toc40481019)

[2.4 本章小结 8](#_Toc40481020)

[第三章 系统的总体设计 9](#_Toc40481021)

[3.1 系统的结构设计 9](#_Toc40481022)

[3.2 系统功能模块设计 10](#_Toc40481023)

[3.3 系统的数据库设计 13](#_Toc40481024)

[3.3.1 系统数据库的E-R模型 13](#_Toc40481025)

[3.3.2 系统数据库表 14](#_Toc40481026)

[3.4 系统界面设计 17](#_Toc40481027)

[3.5 本章小结 19](#_Toc40481028)

[第四章 系统详细设计与实现 20](#_Toc40481029)

[4.1 用户管理模块的详细设计和实现 20](#_Toc40481030)

[4.1.1 用户注册子模块详细设计 20](#_Toc40481031)

[4.1.2 用户登陆子模块详细设计 21](#_Toc40481032)

[4.1.3 完善个人信息子模块详细设计 22](#_Toc40481033)

[4.1.4 权限限制子模块详细设计 23](#_Toc40481034)

[4.2 信息收集模块 24](#_Toc40481035)

[4.2.1 考生信息收集子模块详细设计 24](#_Toc40481036)

[4.2.1 历史排名信息收集子模块详细设计 25](#_Toc40481037)

[4.3 数据库管理模块 26](#_Toc40481038)

[4.3.1 增加数据子模块详细设计 26](#_Toc40481039)

[4.3.2 删除数据子模块详细设计 27](#_Toc40481040)

[4.3.3 修改数据子模块详细设计 28](#_Toc40481041)

[4.3.4 查询数据子模块详细设计 29](#_Toc40481042)

[4.4 分析预测模块 30](#_Toc40481043)

[4.4.1 排名与分数线关系预估子模块详细设计 30](#_Toc40481044)

[4.4.2 排名与分数线关系预估子模块详细设计 31](#_Toc40481045)

[4.5 信息展示模块 32](#_Toc40481046)

[4.5.1 导航栏模块 32](#_Toc40481047)

[4.5.2 主页模块 32](#_Toc40481048)

[4.5.3 表单展示模块 32](#_Toc40481049)

[4.5.4 表格展示模块 32](#_Toc40481050)

[4.5.5 文件展示模块 33](#_Toc40481051)

[4.6 系统的实现 33](#_Toc40481052)

[4.6.1 系统的环境配置 33](#_Toc40481053)

[4.6.2 系统开发 33](#_Toc40481054)

[4.6.3 系统运行说明 34](#_Toc40481055)

[4.7 本章小结 34](#_Toc40481056)

[第五章 系统测试 35](#_Toc40481057)

[5.1 系统测试的方法介绍 35](#_Toc40481058)

[5.2 系统的测试计划 35](#_Toc40481059)

[5.3 系统功能测试与结果 35](#_Toc40481060)

[5.3.1 注册登陆页面测试 35](#_Toc40481061)

[5.3.2 表单填写页面测试 38](#_Toc40481062)

[5.3.3 数据展示页面测试 46](#_Toc40481063)

[5.3.4 文件展示页面测试 50](#_Toc40481064)

[5.3.5 分析预测算法测试 52](#_Toc40481065)

[5.4 测试结果分析 56](#_Toc40481066)

[5.4.1 注册登录页面测试结果分析 56](#_Toc40481067)

[5.4.2 表单填写页面测试结果分析 56](#_Toc40481068)

[5.4.3 数据展示页面测试结果分析 56](#_Toc40481069)

[5.4.4 文件展示页面测试结果分析 56](#_Toc40481070)

[5.4.5 分析预测算法测试结果分析 56](#_Toc40481071)

[5.5 本章小结 56](#_Toc40481072)

[第六章 结束语 58](#_Toc40481073)

[6.1 论文工作总结 58](#_Toc40481074)

[6.2 问题和展望 58](#_Toc40481075)

[参考文献 60](#_Toc40481076)

[致 谢 61](#_Toc40481077)

# 第一章 引言

## 课程背景和意义

高校招生，是指国家高等教育院校按照教育部高校招生分类进行招收新生的工作。而其中的全国统招这一招生类别，更是大多高等教育院校的主要新生来源。高校的生源质量直接影响着学校的发展和办学质量，更进一步，其更是影响着国家的人才储备。正因为其重要性，高校的招生工作备受高校、考生、社会的关注。

招生咨询工作是高校高招工作中不可缺失的环节，也是一次考生和家长能够直接与高校对话的机会。对于高校而言和负责招生咨询的工作人员而言，招生咨询工作是一个宣传自己高校的机会；是了解当年该省分数分布、预估分数线的重要手段之一；是帮助学校吸引优秀人才的重要方式。对于考生家长而言，这是他们在填报志愿之前，可以直接与来自各个高校各个专业的老师直接对话的机会。考生和家长都十分渴望能借助招生咨询的机会，听取老师关于报考和专业选择等方面的建议。为作为高校老师，也很希望能够通过已有的信息，给予学生力所能及的帮助。结合学生成绩，分数线等信息，综合全面给予考生有针对性的建议和宣传。

然而，现阶段招生咨询工作缺少电子手段的支撑，导致招生老师不得不使用较为传统，效率较低的方式进行信息的收集和分析。这不仅消耗了老师大量的精力，还会导致信息滞后、不准确，宣传口径出现偏差等情况。进而导致不能在第一时间给予考生最准确的建议，使得合适的考生资源流失。很大程度上限制了招生咨询工作的准确度和效率。

本文的目标是开发一套包括数据收集，数据分析，数据展示，资料展示的招生咨询系统。通过电子手段对数据进行储存、分析和处理，可以克服很多人工处理的弊端，例如：手写的纸质信息容易丢失，不宜整理、上报数据过程繁琐、分析数据困难等，具有实用价值。利用考生在系统中填报的信息，以及系统内置算法，可以在几秒钟内迅速获取当年分数与排名关系，录取分数线等招生咨询工作中的重要信息，辅助招生咨询工作组成员进行宣传口径的设计。让老师在数据收集和处理方面更加轻松，使得老师能将更多的精力放在招生宣传工作上，进而给考生和家长们留下良好的印象，更有针对性的完成招生咨询工作。

## 课题任务

### 1.2.1 课题内容

本课题的整体目标是开发一套涵盖手机，PC设备的招生咨询系统，使招生咨询工作中所有的参与对象能够通过系统得到有效地组织，沟通和支撑。

本文是基于Python语言设计的在PC端使用的招生咨询系统的一个子系统，需要实现的主要功能有：收集考生信息、分析分数线与排名关系、预测专业录取分数线、展示汇总数据和预测结果。通过这一套流程帮助招生咨询老师承担部分考生信息存储和分数线预测的工作。同时该系统还有辅助招生咨询工作的功能，包括：展示招生材料，管理网站数据功能。

### 1.2.2 本人承担任务及方案选择

本人承担的任务是招生咨询系统中的python子系统的设计与实现。在展示方式上，本人选择了使用web网页形式进行展示，理由是web应用不需要使用的用户单独安装任何客户端应用程序，而可以直接通过浏览器进行访问，这样不会因为操作系统等差异导致该招生咨询系统在某些PC端上难以使用，就招生咨询场景而言使用浏览器访问是更为便捷的方式。

在基于Python的Web开发框架的选择中，本人选择了Django开发框架。Django开发框架遵循类似MVC模式的MVT设计模式，加上功能强大且多样的第三方插件支持，造就了Django很强的扩展性。更重要的是，相较于强调简约的Flask轻量级开发框架，使用Django开发框架的逻辑更加清晰。

了解并熟悉招生咨询工作的流程和细节，明确了传统信息收集和分析方法弊端，并着重进行改进。收集阅读有关招生咨询以及python web应用开发等方面的相关资料和文献，学习Django开发框架。完成系统的需求分析，概要设计，详细设计，编码实现，单元测试，系统测试等工作，并最终完成论文的撰写。

## 论文结构

本论文一共由六个章节组成，从系统的需求分析、设计、实现、测试进行介绍。

第一章：引言，介绍课题背景及意义，课题内容及本人的主要工作。

第二章：系统的需求分析，从系统的角色、功能需求和非功能需求这三个方面进行简要的分析。

第三章：系统总体设计，在第二章需求分析的基础上对系统进行总体结构设计、功能模块设计、数据库设计以及界面原型设计

第四章：系统的详细设计与实现，按照之前划分的功能模块，阐述系统中每个子功能模块的实现方法。最后阐述系统的总体实现方法。

第五章：系统测试，介绍测试的方法，测试计划、进行系统功能测试过程以及测试结果分析

第六章：结束语，包括对本次工作的总结，以及就系统可能存在的问题提出完善方法和展望。

# 第二章 系统的需求分析

## 2.1 系统用户角色分析

本系统需要实现一个招生咨询系统，对应现实的招生咨询工作中的三类使用者设计系统用户角色，即学校各省招生宣传组成员、各省招生宣传组组长、招办负责人。这三类角色分别对应本系统中的普通用户、管理员和超级管理员。其权限从低到高并且向下兼容。具体用例图如2-1所示。其详细功能分析如下：

### 2.1.1 普通用户分析

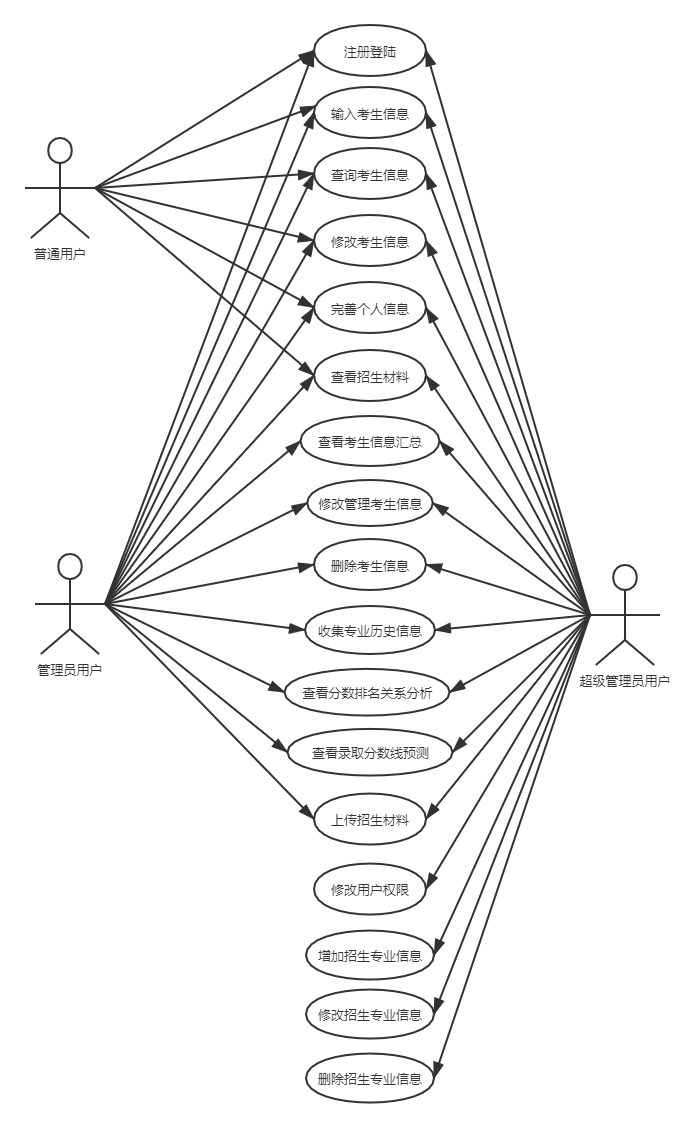
普通用户是指学校的招生宣传组组员，其主要负责在招生咨询会的现场收集考生资料，并且回答考生及考生家长的各类问题。普通用户可以通过注册和登陆获取访问招生咨询系统的权限。系统授予普通用户收集考生数据，查找考生数据，修改考生数据，完善个人信息的权限，查看招生材料。

### 2.1.2 管理员用户分析

管理员用户是指学校派至各省的招生宣传组组长，其主要负责确定招生咨询口径。管理员权限需要在注册后由超级管理员开启。管理员除了拥有普通用户所有权限外，还有查看所有考生信息，修改管理考生信息，删除考生信息，收集历史录取排名，查看分数排名关系分析信息，查看录取分数线预测信息，上传招生资料的权限。其中，修改管理考生信息与普通用户修改考生信息功能的差别是：修改管理考生信息可以修改任意考生的信息，并且可以给考生增加备注信息便于管理；而普通用户的修改考生信息功能，必须通过查找考生信息之后才能对查找到的考生信息进行修改，并且不能添加备注信息。

### 2.1.3 超级管理员用户分析

超级管理员是指学校招生办的负责老师，其主要责任是管理涉及全网站的数据。超级管理员由后台生成，同时超级管理员无法修改其他超级管理员的信息。超级管理员除了拥有管理员所有权限之外，还具有修改网站用户权限，增加招生专业信息，修改招生专业信息和删除招生专业信息的权限。



#### 图2-1 系统用例图

## 2.2 系统功能需求分析

招生咨询系统的功能主要分为五个部分：用户管理、信息收集、信息管理、分析预测、信息展示。

用户管理的功能是让用户注册账号或使用已有的账号进行登陆，以及填写或更改个人信息。以及系统中一些界面的权限限制功能。

信息收集的功能按照数据进行分类，有考生数据，专业历史排名数据两类。采集考生数据需要老师在招生咨询现场引导考生在系统中填报自己的一些基本信息；采集专业历史排名数据需要招生组组长在招生咨询工作开展之前，在系统中填报所有在本省招生专业的历史排名信息。数据采集后，系统会将考生信息数据进行核查，将数据存放到数据库中。

信息管理功能满足用户对数据库中的数据能够进行增、删、改、查这四个基本操作的需求。所有与相关的用户需求中与操作数据库有关的需求包含在此功能需求中。

分析预测功能可以将数据库中收集到的数据进行分析。系统主要需要分析预测出两个信息：一是根据考生填报的排名信息分析出分数排名之间关系，二是综合历史信息和分数排名关系，分析预测今年各个专业的录取分数线。

信息展示的功能需要系统能够通过web界面将信息展示给用户。其中由于本系统展示数据大多为表格类型数据，为方便查看，展示数据中包括排序功能。所收集的考生信息数据提供下载功能，允许用户将数据下载到本地进行处理。

系统功能需求表见表2-1所示

#### 表2-1 系统功能需求表

|  |  |  |
| --- | --- | --- |
| 需求 | 需求的功能 | 功能描述 |
| 用户管理 | 注册登录 | 使用户在本系统内注册个人账户或使用已经注册的账户进行登陆。账户主要用于分辨用户的权限等级。 |
| 完善个人信息 | 用户登陆账户后，需要填写招生咨询省份、地点、电话等个人信息。在使用系统的任何时候，用户可以对自己已经填写过的个人信息进行修改。 |
| 页面权限限制 | 某些页面具有访问权限，拒绝没有权限的用户访问这些页面。 |
| 信息收集 | 考生信息收集数据 | 通过提交表单的方式获取所需的数据，数据将通过系统的查重，对于考号姓名均一致的数据系统将不再记录，将不重复的数据保存至数据库中。 |
| 历史排名信息收集 | 通过提交表单的方式获取该省份的历史录取排名信息，并将数据存放到数据库中。 |
| 信息管理 | 增加数据 | 允许有权限的用户向对应的数据中加入新的数据。上传招生材料、增加招生专业信息功能包含于此功能需求。 |
| 删除数据 | 允许有权限的用户向对应的数据中加入新的数据。删除考生信息、删除招生专业信息的功能包含于此功能需求。 |
| 修改数据 | 允许有对应数据库操作权限的用户对数据库中信息进行修改。修改考生信息，修改管理考生信息、修改用户权限及修改招生专业信息的功能包含于此功能需求。 |
| 查找数据 | 允许有对应数据库操作权限的用户对数据库中信息进行查找。查询考生信息功能包含于此功能需求。 |
| 分析预测 | 分数排名对应关系分析 | 通过系统采集的考生信息数据，找出同分情况下排名最高和最低的两个数据，并将其存入数据库 |
| 录取分数线预测分析 | 通过系统预估的排名与分数对应关系，以及采集到的历史录取排名数据，找出在本年分数排名最接近历史排名的分数区间，将该区间存入数据库。 |
| 信息展示 | 数据可视化 | 通过页面向用户展示表单、表格、文件等数据类型；表格类型的数据展示，提供依照数据内容施行正序或逆序排序功能；部分表格内容允许用户将数据以excel的格式下载本地，在本地进行查看和处理 |

## 2.3 系统非功能需求分析

本系统计划部署在服务器中，用户通过浏览器访问实现在PC端使用。本系统使用B/S框架，即将系统的核心实现部分，如服务器程序和数据库均在后台进行运行，用户使用浏览器同数据库进行数据交互。作为招生咨询系统，普通用户的界面会展示给所有前来咨询的考生和家长。因此在界面美观方面，本系统的所有界面均使用一套样式，保证跳转界面不会显得很突兀。同时在普通管理员的首页上，除了常用功能链接外，还有加入了登陆用户的联系方式以及学校照片，方便考生及家长联系询问以及了解学校。可扩展性方面，本系统的核心逻辑使用Python语言编写，如需要系统升级维护等工作，只需要对服务端进行修改，而不需要每个用户进行额外操作。因此在这方面，本系统的升级维护成本十分低廉。

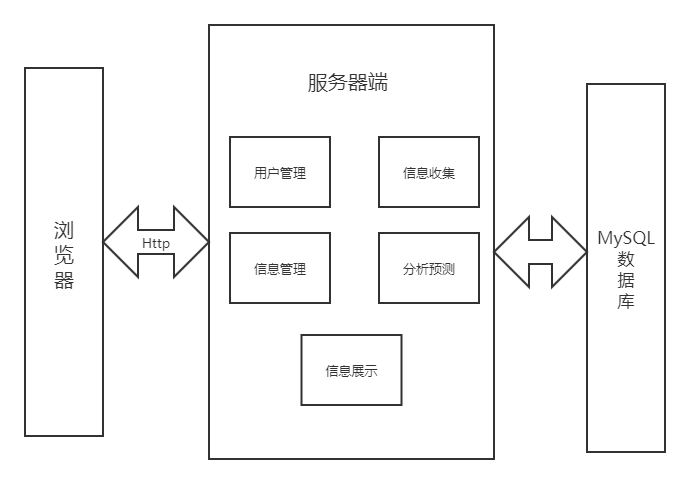
## 2.4 本章小结

本章进行了系统功能分析，分别从用户角度，系统功能需求和系统非功能需求角度进行了研究。为进行系统整体设计做好了准备。

# 第三章 系统的总体设计

## 3.1 系统的结构设计

由上一章中的非功能需求分析可知，本系统遵循B/S框架结构，系统的核心程序部署在服务器端上，而客户通过PC机的浏览器通过合同谈判请求获取服务器的信息，而后显示在浏览器上。数据库与服务器端建立连接，实现服务器与数据库的交互功能。系统结构设计图如3-1所示。



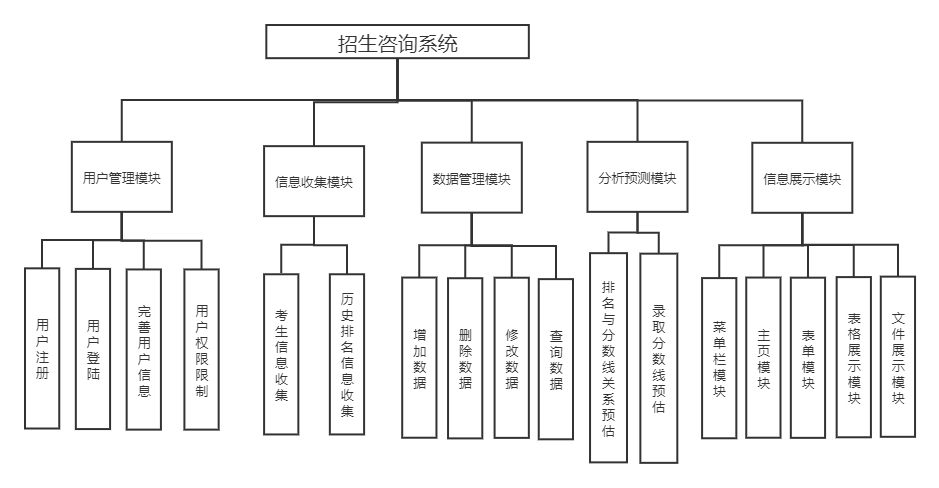
#### 图3-1 系统结构设计图

用户使用本系统时，只需要在浏览器中输入特定的地址，浏览器会通过http请求方式向服务器端发送请求数据，收到服务器端的回应后，浏览器会将完整的页面呈现给用户。

服务器端包含系统的五个功能模块，即：用户管理模块、信息收集模块、信息管理模块、分析预测模块和信息展示模块。用户管理模块在注册完成注册后将用户信息存入数据库；在用户登陆，跳转页面时请求数据库中数据进行判断，并将判断结果返回给用户。信息收集模块获取用户通过浏览器提交的表单信息，并将数据存储到数据库。信息展示模块收到请求时，会通过数据分析模块获取经过分析的数据信息，而后返回给用户。信息管理模块获取用户对数据库操作的请求，并对数据库中数据进行对应的修改并将结果返回给用户。

## 3.2 系统功能模块设计

根据之前系统功能需求分析的内容，将系统功能分为用户管理、信息收集、信息管理、分析预测和信息展示这五个功能模块。每个功能模块下又细分多个子模块，系统功能模块设计图如图3-2所示。



#### 图3-2 系统功能模块设计图

用户管理模块主要实现用户登陆注册，详细信息完善修改和跳转时权限验证功能。注册时，用户输入用户名密码姓名等信息，后台检测用户名是否被占用，检测通过后将用户信息存入数据库；登陆时后台获取前端用户所填写的用户名和密码，并与数据库中信息进行比对，比对成功后允许用户访问系统的其他功能；用户首次进入系统时，系统自动弹出完善个人信息界面，要求用户填写个人手机号，咨询省份及地点等信息，用户也可以在使用系统的任何时候通过该模块随时对这些信息进行修改；当用户进行网页跳转时，权限管理功能会询问数据库判断用户是否有目标页面的访问权，如果有则返回目标页面信息，如果没有则返回404页面。

信息收集模块主要实现考生信息收集和校验，以及历史排名信息收集。用户在浏览器的表单中填写考生的基本信息并传入服务器端，服务器收到信息后检查表单各个数据是否合法，而后检查改考生是否第一次填写。如果是则存入数据库返回成功，如果不是则返回失败并抛弃数据。历史信息收集模块是向管理员用户询问各个专业的历史录取排名，管理员用户填写后，服务器端将数据存入数据库存储。

数据管理模块主要实现对数据库的进行增删改查的操作。实现人为对数据库数据进行管理。

分析预测模块实现系统对数据进行分析的逻辑。当用户访问预测信息页面是，该模块会向对应数据库请求数据，通过服务端的算法计算出预测结果，并将预测结果返回给用户。分析预测模块主要实现两种信息的分析预测：分数与排名关系的预测以及录取分数线预测。

信息展示模块主要实现向用户战士的交互界面，包括菜单，主页表单和数据库信息展示。其中数据库信息展示分为表格展示和文件展示模块两种。其目的是实现友好的交互界面，让用户操作起来更加简便。

本系统的功能模块设计描述表如表3-1所示。

#### 表3-1 系统功能模块设计描述表

|  |  |  |
| --- | --- | --- |
| 模块 | 功能名称 | 功能描述 |
| 用户管理模块 | 用户注册 | 用户在本系统中注册一个新的普通用户账号，系统检测账户名是否被占用。 |
| 用户登录 | 用户使用已经注册过的账号进行登陆，系统检测账号密码是否匹配。 |
| 完善用户信息 | 要求初次登陆系统的用户填写电话，咨询场地，咨询省份等更细致的个人信息，并保存进数据库。 |
| 用户权限限制 | 用户进行页面跳转时，服务器端通过向数据库请求用户权限信息，判断用户是否有权限跳转至目标页面。 |
| 信息收集模块 | 考生信息收集 | 用户可以通过让考生填写表单的形式收集考生的各类基本信息。表单传到服务端后，服务端检测表单各项是否符合规范，而后向数据库请求用户信息，并将咨询地点、省份、时间等信息自动加入条目，保存至数据库中。 |
| 历史排名信息收集 | 管理员用户向系统输入各个省份历史排名信息。系统将数据存入数据库。 |
| 数据管理模块 | 增加数据 | 用户可以通过此模块向任意数据库中人为增添数据。 |
| 删除数据 | 用户可以通过此模块删除数据库中数据。 |
| 修改数据 | 用户可以通过此模块修改数据库中数据 |
| 查询数据 | 用户可以通过此模块查询数据库中数据 |
| 分析预测模块 | 排名与分数线关系预估 | 当用户请求响应页面时，系统向数据库请求所有考生填报的信息，通过算法处理后预估出排名和分数线关系，存入数据库并展示。 |
| 录取分数线预估 | 当用户请求响应页面时，系统向数据库请求排名与分数关系预估和历史排名信息，通过算法处理后预估各个专业的分数线，并存入数据库并展示。 |
| 信息展示模块 | 菜单栏展示 | 菜单栏位于网页上方，按照用户级别分为普通用户菜单栏，和管理员用户菜单栏。菜单栏包括所有向用户开放的功能连接，方便用户随时跳转到需要的功能界面。 |
| 主页展示 | 主页根据用户级别分为普通用户主页和管理员用户主页。普通用户主页包括考生信息填报功能连接、招生材料展示连接、学校风光以及咨询老师姓名和联系方式；管理员用户主页包括管理员所拥有的全部功能。 |
| 表单展示 | 所有用户向服务端提交的信息均由表单的形式提交。系统根据用户进入的页面显示相应的表单信息让用户进行填写。 |
| 表格数据展示 | 所有数据库表的展示均为表格方式展示，服务器端筛选获取相应数据后以表格形式返回给用户。同时所有表单可以按照任意列的信息，进行排序功能。 |
| 文件数据展示 | 所有招生材料以pdf形式进行展示，服务端从数据库中提取pdf名称信息，拼接pdf存储的相对地址返回给用户，让用户的浏览器进行对应的显示。 |

## 3.3 系统的数据库设计

### 3.3.1 系统数据库的E-R模型

系统数据库设计采用实体-联系模型。本系统实体包括：系统用户、考生，招生专业，招生材料四个。系统用户实体类包括以下属性：唯一的ID、用户名、姓名、密码、电话、咨询省份、咨询地点、权限等级、最后登录时间。考生实体类包括：唯一的ID、姓名、考号、分数、位次、电话、毕业中学、文理科、拟报专业、志愿次序、咨询省份、咨询地点、咨询日期、咨询时间。招生专业实体类包括：唯一的ID、专业名称、招收学生类别（文科/理科）、招生省份、历史招生排名、预测录取分。招生材料实体类包括：唯一ID、材料名称、适用省份。

系统的数据库E-R图如图3-3所示。

#### 图3-3 系统数据库E-R图

### 3.3.2 系统数据库表

本项目使用MySQL数据库进行数据存储。所创建的数据库表较多，其中两类数据库表在这里不做介绍。首先，为了尽量减少存储的冗余数据项以节省空间，在数据表设计中包含了例如：省份信息（属性包括：ID，省份名称）、文理科信息（属性包括：ID，文科/理科）等内容相似，结构简单，属性少的表。其次，是Django自动创建的大部分表，例如：数据库迁移记录表、会话记录表等。这些表信息主要用于记录使用该数据库的日志信息，展示于Django自带的管理界面。

以下简单介绍本人为实现系统功能创建的较为重要的数据库表：

user用户信息表：该表有Django框架自动创建，属性未作修改，因此存在一些不使用的属性，在此不做陈述。该表的属性包括：ID（主键）、username 用户名、password 密码、first\_name 姓名（考虑到分为姓、名存储意义不大，因此使用该属性存储姓名，last\_name为空）、last\_login 最后登录时间、is\_superuser 是否为超级管理员、is\_staff 是否为管理员、is\_active 是否准入、date\_joined 注册日期。该表的设计见表3-2所示。

#### 表3-2 user表的设计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性中文名称 | 属性英文名称 | 存储数据类型 | 长度 | 是否为空 | 备注 |
| 序号 | ID | int | 10 | 否 | 主键 |
| 用户名 | username | varchar | 150 | 否 |  |
| 密码 | password | varchar | 128 | 否 | 存储散列值 |
| 姓名 | first\_name | varchar | 30 | 否 |  |
| 最后登陆时间 | last\_login | datetime | 6 | 否 |  |
| 是否为超级管理员 | is\_superuser | tinyint | 1 | 否 |  | |
| 是否为管理员 | is\_staff | tinyint | 1 | 否 |  |
| 是否准入 | is\_active | tinyint | 1 | 否 |  |
| 注册日期 | date\_joined | datetime | 6 | 否 |  |

Userprofile用户信息扩展表：由于Django自带的用户信息表不能完全满足系统需求，本人采用一对一链接的方式扩充用户个人信息表。该表的属性包括：user\_ID（主键，与user表链接的外键）、telephone 用户电话、province 咨询省份（与省份信息表链接的外键）、place 咨询地点。该表的设计见表3-3所示

#### 表3-3 userprofile表的设计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性中文名称 | 属性英文名称 | 存储数据类型 | 长度 | 是否为空 | 备注 |
| 用户序0号 | user\_ID | int | 10 | 否 | 主键、user表外键 |
| 电话 | telephone | varchar | 20 | 否 |  |
| 咨询省份 | province | int | 10 | 否 | province表外键 |
| 咨询地点 | place | varchar | 50 | 否 |  |

Stuinfo考生信息表：本数据库表是记录考生基本信息表，包括以下属性：ID（主键）、name 姓名、testnum 考号、score 分数、rank 排名、tele 联系方式、high\_school 毕业高中、application\_rank 志愿顺序、place 咨询地点、tip 备注、date 日期、time 时间、sciorart\_id（文科或理科表外键） 文科/理科、is\_international 是否报考国际学院、major1\_id（专业信息表外键） 第一志愿专业、major2\_id（专业信息表外键，可为空） 第二志愿专业、major3\_id（专业信息表外键，可为空） 第三志愿专业、province\_id（省份信息表外键） 咨询省份、staff\_id（用户user表外键） 录入人员。该表的设计见表3-4所示：

#### 表3-4 stuinfo表的设计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性中文名称 | 属性英文名称 | 存储数据类型 | 长度 | 是否为空 | 备注 |
| 序号 | ID | int | 10 | 否 | 主键 |
| 姓名 | name | varchar | 20 | 否 |  |
| 考号 | testnum | varchar | 14 | 否 |  |
| 分数 | score | smallint | 5 | 否 |  |
| 排名 | rank | smallint | 5 | 否 |  |
| 电话 | tele | varchar | 20 | 否 |  |
| 毕业高中 | high\_school | varchar | 30 | 否 |  |
| 志愿顺序 | application\_rank | smallint | 5 | 否 |  |
| 咨询地点 | place | varchar | 30 | 否 |  |
| 备注1 | tip | varchar | 200 | 是 |  |
| 咨询日期 | date | date | 10 | 否 |  |
| 咨询时间 | time | time | 13 | 否 |  |
| 文/理科 | sciorart\_id | int | 10 | 否 | 文理科表的外键 |
| 是否报考国际学院 | is\_international | tinyint | 1 | 否 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 第一意向专业 | major\_id1 | int | 10 | 否 | 专业信息表外键 |
| 第二意向专业 | major\_id2 | int | 10 | 是 | 专业信息表外键 |
| 第三意向专业 | major\_id3 | int | 10 | 是 | 专业信息表外键 |
| 咨询省份 | province\_id | int | 10 | 否 | 省份信息表外键 |
| 咨询老师 | staff\_id | int | 10 | 否 | 用户信息表外键 |

rankpredict 排名分数关系预测表：本数据库表存储各省当年分数排名预测信息，包含以下属性：ID（主键）、score 分数、highrank 该分最高排名、lowrank 该分最低排名、province\_id（省份信息表外键） 省份、sciorart\_id（文理科信息表外键） 文理科。该表的设计见表3-5所示：

#### 表3-5 rankpredict表设计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性中文名称 | 属性英文名称 | 存储数据类型 | 长度 | 是否为空 | 备注 |
| 序号 | ID | int | 10 | 否 | 主键 |
| 分数 | score | smallint | 5 | 否 |  |
| 最高排名 | highrank | smallint | 5 | 是 |  |
| 最低排名 | lowrank | smallint | 5 | 是 |  |
| 省份 | province\_id | int | 10 | 否 | 省份信息表外键 |
| 文科/理科 | sciorart\_id | int | 10 | 否 | 文理科表外键 |

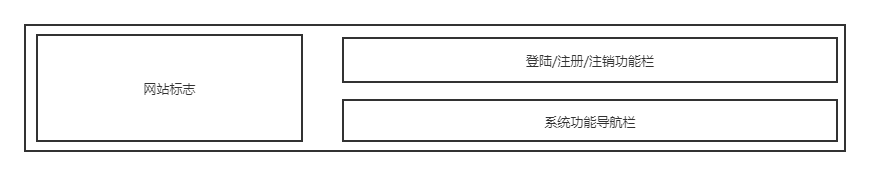
consultmaterial 招生咨询文件记录表：为了防止上传的文件命名问题导致寻找文件地址时可能出现错误，系统在上传文件后会为文件按照当前时间命名，同时通过本数据库表保存文件真实名称与存储名称之间关系，本表包括以下属性：ID（主键）、name 文件名、date 上传日期、time 上传时间、user\_id（用户信息表外键） 上传用户、file\_name 存储名、province\_name（考虑到文件可能不止作用于唯一省份，例如招生简章这种全国适用文件，这里增加了“全国”省份选项，因此该属性不是省份信息的外键） 文件作用省份。该表的设计见表3-6所示：

#### 3-6 consultmaterial表设计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性中文名称 | 属性英文名称 | 存储数据类型 | 长度 | 是否为空 | 备注 |
| 序号 | ID | int | 10 | 否 | 主键 |
| 文件名 | name | varchar | 100 | 否 |  |
| 上传日期 | date | date | 10 | 否 |  |
| 上传时间 | time | time | 13 | 否 |  |
| 上传用户 | user\_id | int | 10 | 否 | 用户信息表外键 |
| 文件存储名 | file\_name | varchar | 25 | 否 |  |
| 作用省份 | province\_name | varchar | 5 | 否 |  |

## 3.4 系统界面设计

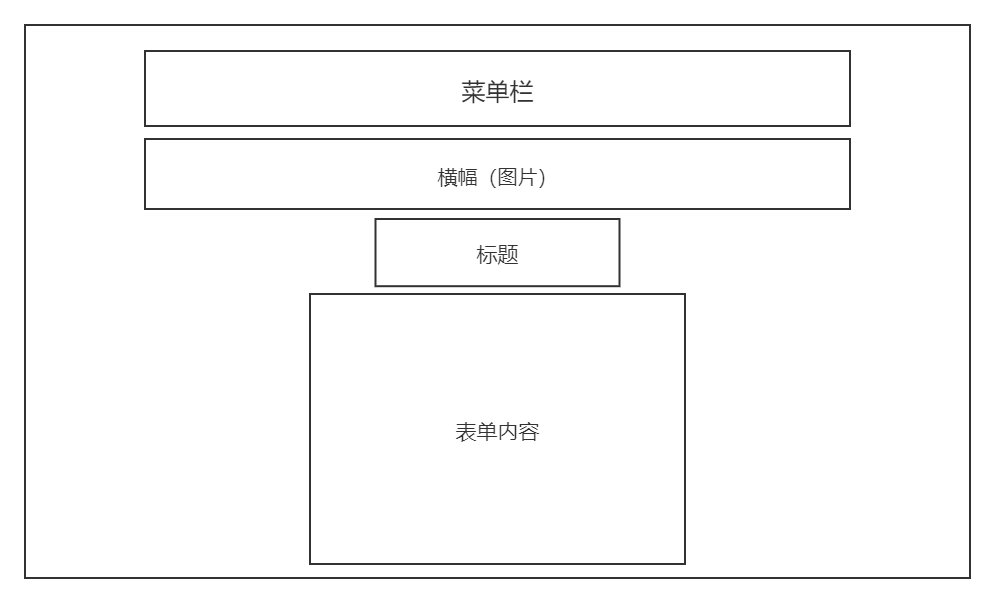
本系统主要有三个主要界面：主界面、表单展示页面、数据展示页面。

除数据展示页面外，所有页面共用一个样式的菜单栏。菜单栏的导航区域为不同权限的用户显示不同的功能。左边的标志可以方便用户随时回到首页，右上角注册/登陆/注销链接会根据用户的登录状态动态切换。菜单栏设计原型见图3-4。

#### 图3-4 菜单栏设计示意图

主界面根据登录用户权限不同显示不同的系统功能，示意图以普通用户主界面功能为例。普通用户主界面还有学校风光，老师联系方式板块，方便前来咨询的考生和家长了解学校。主界面的设计原型如图3-5所示。

#### 图3-5 主页设计示意图

表单展示页面用于所有需要数据库保存或调用数据库信息的操作，例如登陆、注册、考生信息收集等功能。表单页面包括菜单栏、标题栏和表单三部分。标题栏显示改表单的题目和作用，表单栏显示表单的具体内容，包括项目名称、填写框和填写提示。表单展示设计示意图如图3-6所示。

#### 图3-6 表单页面设计示意图

数据展示页面包括标题区域、数据表展示区域、备注区域和链接区域。标题区域展示了该数据表的标题。备注区域展示了该数据表的一些阅读注意事项，不一定所有数据表的展示都包括这块区域。链接区域包括查看省份选择、返回上一页、下载excel选项。数据展示页面设计示意图如图3-7所示。

#### 图3-7 数据表展示页面设计示意图

## 3.5 本章小结

本章在前一章需求分析的基础上，以实现需求为目的对系统的架构、功能模块、数据库和界面进行了分析。分析采用自顶向下的方法，功能模块下细分子功能模块，数据库先进行E-R图设计后确定数据表属性。为后续分析实现本系统做好了充足的准备。

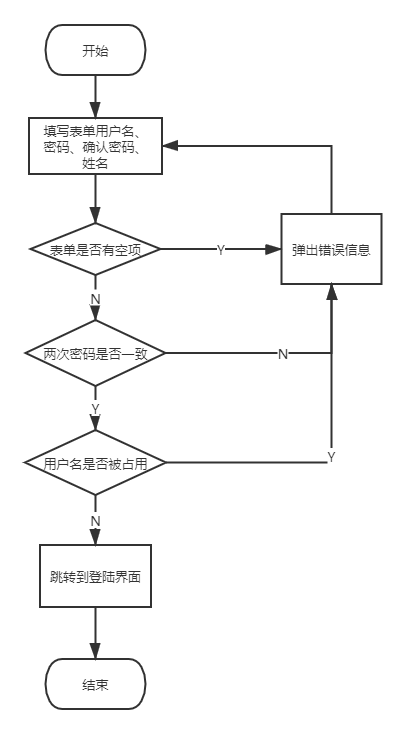
# 第四章 系统详细设计与实现

Django框架为了保持低耦合度，使用了与MVC模式大体相同MTV模式，即模型（model）、模板（Template）、视图（View）。本系统编写的业务核心逻辑代码大多在各自应用（application）文件夹下的views.py文件中，但一个view.py文件会包含多个核心业务逻辑。因此本章指出的逻辑代码位置会精确到函数名称。

## 4.1 用户管理模块的详细设计和实现

用户管理模块主要分为用户注册、用户登录、完善用户信息、修改用户信息和权限限制五个子功能模块。这五个模块前端依赖于HTML表单、服务器端使用python进行逻辑处理。其中注册登录表单和相关数据库表为Django框架自带，其余表单为自行设计。

### 4.1.1 用户注册子模块详细设计



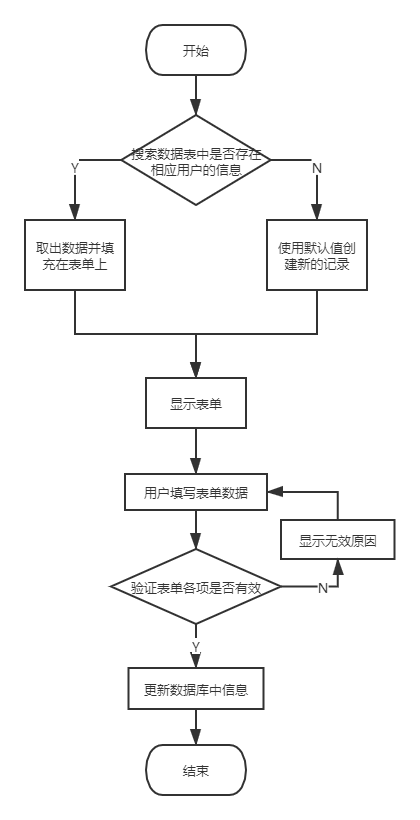
#### 图4-1 用户注册流程图

用户注册流程图如图4-1所示。用户注册相关逻辑代码位于users文件夹中views.py中的register函数；前端模板位于information/templates/users文件夹的register.html中。首先判断用户是否在使用post方式提交数据，如果不是，则服务器让前端显示注册用的表单；如果是，则用临时变量form接收表单信息。之后进行表单认证，表单认证根据对应数据库要求，确认用户所填写各类信息符合存储要求，其次运行在forms.py中自定义认证方式，即认证两次输入密码是否一致。如果不一致则会抛出无效表单的错误并弹出提示。表单认证通过后，系统尝试到数据库中寻找该用户名的数据，如果找不到则为其创建条目并返回登陆界面；反之返回用户名被占用的提示。

### 4.1.2 用户登陆子模块详细设计

本系统所使用的登陆模块逻辑是Django框架自带的登陆视图LoginView，该源代码可以在Django安装目录下contrib/auth/views.py中找到，这里简述一下大致的逻辑。首先当用户请求登陆页面是，首先用户通过get方式请求页面时，系统会首先调用dispatch()方法。之后用户通过post方式提交表单后，系统会再次调用dispatch，通过post()方法处理form数据。如果数据正确则调用form\_valid()，反之调用form\_invalid()。在form\_valid方法的最后会将用户重定向至登陆成功后的显示页面，在本系统中就是系统的主页。

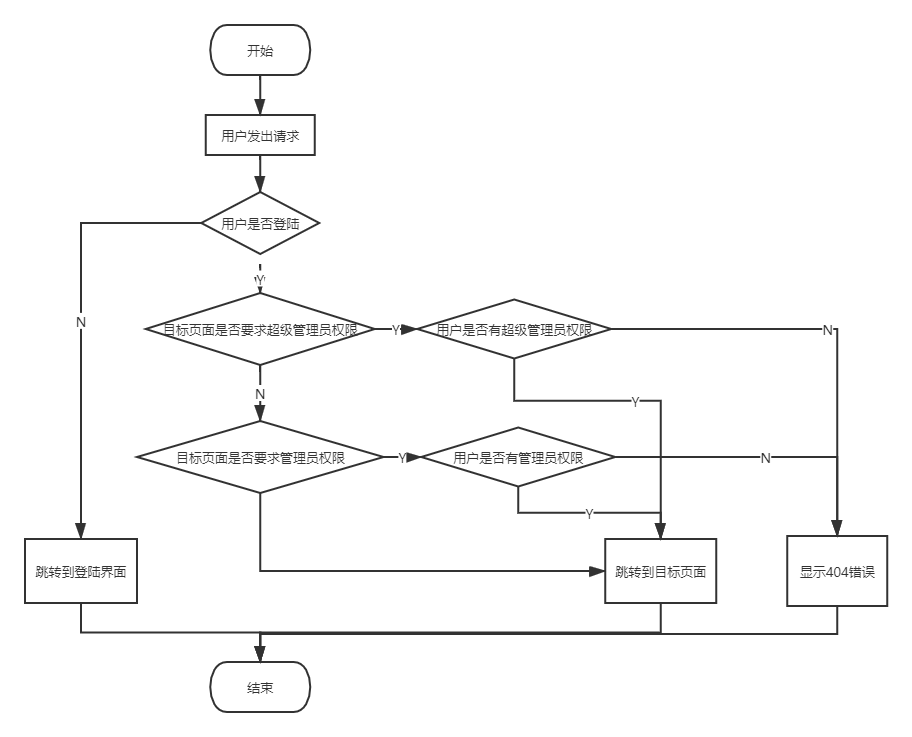
### 4.1.3 完善个人信息子模块详细设计



#### 图4-2 完善个人信息流程图

完善个人信息流程图如图4-2所示。本功能的相关逻辑代码位于users文件夹中view.py中的place函数中，前端模板代码位于information/templates/users文件夹place.html中。当用户使用该功能时，系统首先尝试从数据库中获取用户的个人信息。如果获取失败，说明用户是首次登陆系统，则系统会展示一个空表单让用户填写，并且在数据库中创建一条记录，其中非空属性预先置入默认值；否则，系统会将数据库中已有的信息取出预先填写在表单上，方便用户进行修改。用户需要在页面上填写表单的对应项目，提交之后系统会检查表单是否有空项、表单各项是否符合数据库表存储格式和规定。检查无误后通过用户修改的表单，修改数据库中的数据，以这种方式完成个人信息的完善功能。

### 4.1.4 权限限制子模块详细设计



#### 图4-3 权限管理限制流程图

权限管理限制流程图如图4-3所示。本功能模块在用户每次进行跳转操作时触发，防止用户已匿名身份访问系统或是访问自己权限外的功能。本功能判定用户是否登陆使用的是Django框架提供的装饰器@login\_required，限制管理员逻辑代码位于adminsite文件夹下的views.py中admin\_only中；限制和超级管理员权限的代码位于managesite文件夹中views.py的superuser\_only中。之所以这两个功能在不同的应用文件夹下，是因为对应的应用文件夹中所实现的页面大多要求管理员或超级管理员权限。本功能模块没有前端展示部分。当用户进行跳转页面请求时，系统首先判定用户是否是匿名用户，如果是则自动跳转到登陆界面引导用户进行跳转。之后依次判断目标页面是否需要超级管理员和管理员权限，因为超级管理员包含所有管理员权限，因此优先判定超级管理员。如果需要则依照用户权限做出相应的响应。

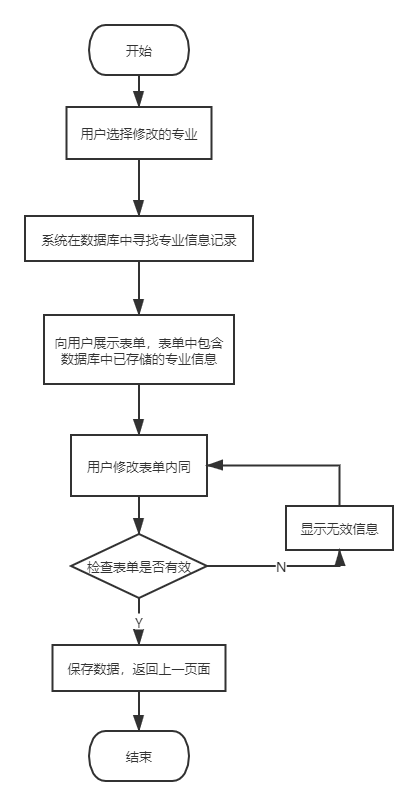
## 4.2 信息收集模块

### 4.2.1 考生信息收集子模块详细设计

#### 图4-4 考生信息收集流程图

考生信息收集子模块流程图见图4-4所示。本功能模块的逻辑代码位于information文件夹中views.py的new\_stu函数中。在收集到信息，进行查重时，其会调用同文件下的dupcheck方法进行查重。首先只要用户没有使用post方法提交数据，系统就会展示信息收集表；如果用户提交数据，系统首先查看表单填写是否有效，如果有效则进行查重。查重逻辑是使用变量curnum，curname保存用户的考号和姓名，使用objects.get的方法在数据库中搜索满足这两条件的数据。如果存在则返回存储失败页面，如果不存在则为该条数据增加录入人、录入时间、录入省份、录入地点等信息，一起存入数据库中。最后返回成功页面。

### 4.2.1 历史排名信息收集子模块详细设计



#### 图4-5 历史排名信息收集流程图

历史排名信息收集子模块流程图见图4-5所示。本功能的代码位于adminsite文件夹中views.py中的history\_edit函数中。首先用户通过展示界面选择其要进行填写的专业，系统通过get的方法得到用户修改的专业id，并使用该id通过objects.get的方式到数据库中进行搜索，并将相关数据预先填写到表单中，方便用户查看。用户填写完表单后，通过post方式提交，系统会判断用户提交的表单是否有效，有效则将其保存并退回至上一界面；无效则显示错误信息。

## 4.3 数据库管理模块

### 4.3.1 增加数据子模块详细设计

#### 图4-6 增加数据流程图

增加数据流程图如图4-6所示。本子功能模块主要用于管理员管理数据库时进行增加数据的操作。此处以增加招生材料为例对流程图进行解释，上传招生文件代码位于material文件夹的view.py文件中upload。首先用户从招生材料管理界面进入上传招生材料的界面。系统收到请求后从表单中调出上传招生材料的表单，并返回给用户呈现在浏览器上。用户上传文件，通过Post方法进行提交。系统收到表单后首先检查表单是否有效，如果表单有效，则在ConsultMaterial表中创建一个新的记录，将用户起的名字进行保存。之后os.path方法对资料进行重命名，命名为当前的计算机时间，即1970.1.1至今的秒数。这么做的目的是为了防止用户起的名字出现乱码等意想不到的错误。之后将文件存储到指定位置，数据库的纪录中加上上传人的信息存入数据库。

### 4.3.2 删除数据子模块详细设计

#### 图4-6 删除数据流程图

删除数据流程图见图4-6所示。本子功能模块主要用于管理员对指定数据进行删除时使用。此处以管理员删除考生信息为例进行解释，该部分的逻辑代码位于information文件夹中views.py的delete函数。首先用户需要选择制定的记录进行删除操作，用户点击删除链接后，浏览器会弹出提示，向用户进行二次确认。弹出提示的代码是在前端使用JS实现的，位于information/template/information的admin\_display.html文件中。用户进行确认后，系统会跳转到一个URL中包含该数据id的页面，不过该页面没有前端代码，因此不会有任何的显示。系统通过get的方式获取URL中的id，并在数据库中定位该数据。之后将其删除并将用户重定向到考生信息汇总界面。

### 4.3.3 修改数据子模块详细设计

#### 图4-7 修改数据流程图

修改数据流程图见图4-7所示。本子功能模块主要用于管理员在数据展示界面对指定数据进行修改时使用。此处以管理员修改考生信息数据为例进行解释。该部分代码位于information文件夹中views.py的edit\_info中。当用户在展示界面选择需要修改的数据时，系统会将用户重定向到带有数据id和考生考号信息的URL的界面。传参中的info\_id和info\_testnum接收来自同一文件夹下urls.py中<int:info\_id>/<int:info\_testnum>捕获的数据。通过这两条信息系统可以唯一确定一条数据，并将数据内容预置到表单上对应位置。用户修改完毕提交后，系统将通过valid()方法判断用户填写的表单是否有效，有效的话就存入数据库并返回展示界面；无效则显示错误信息。

### 4.3.4 查询数据子模块详细设计

#### 图4-8查询数据流程图

查询数据子模块流程图如图4-8所示。该模块主要用于学生修改信息时查询调取学生已经填写的数据。该部分代码位于information文件夹中views.py的search\_info中。首先系统展示SearchForms表单，让用户填写需要查找的考生姓名和考号。经过表单审核后，通过try的方式尝试获取用户所填写信息对应的记录。如果不存在则返回失败页面并重定向到表单让用户重新填写；如果存在则使用表单形式展示数据，并调用修改数据子功能模块允许用户对数据进行修改。

## 4.4 分析预测模块

### 4.4.1 排名与分数线关系预估子模块详细设计

#### 图4-9 排名与分数线关系预估流程图

分数与排名关系预测流程图如图4-9所示。该模块的代码位于information文件夹中views.py中的update\_predict和rank\_display。每当有新数据插入考生信息时，系统会调用update\_predict对预测进行更新；而当用户请求展示页面时，系统会调用rank\_display，其中会使用update\_predict进行更新，并进行空缺分数的填补。首先当系统收到表单数据时，Django会创建一个新的条目对象保存到变量new\_info中，但暂时不保存到数据库。通过new\_info的score、province、sciorart属性确认数据库是否曾经记录过与该考生同分的信息。如果有则通过该考生信息更新排名上下界；如果没有则新建条目，将该考生排名定义其分数的上下界并存入数据库。为了防止相邻两个条目的分数信息相差不为1，在新建了条目后，需要检查其与之前本表中的分数上下限的差距，并创建二者之间相差得分数记录以将差距补齐。新创建的分数条目中排名信息定义为空，展示时展示为0。新的考生信息录入时使用该算法得时间复杂度为O(1)，管理员请求时该算法的时间复杂度为O(n)，n为该省考生信息手机个数。

### 4.4.2 排名与分数线关系预估子模块详细设计

#### 图4-10 各专业录取分数线预估流程图

专业录取分数线预估流程图如图4-10所示。该子功能模块在用户请求对应的展示页面时触发。涉及的代码位于adminsite文件夹views.py中borderline\_predict函数中。首先系统根据请求的用户确定向其展示的省份，并通过省份信息向数据库请求该省份的分数排名预测和专业历史信息。之后遍历所有专业，对每个专业先确定预测录取分数上界，再确定预测录取分数下界。上界确定方法：取考生信息中最高分为上界，每次查看上界分数对应的最低排名是否小于历史排名，如果小于则将上界分数减一，反之结束确定上界。下界确定方法：取考生信息中最低分为下界，每次查看下界分数对应的最高排名是否大于历史排名，如果大于则将下界分数加一，反之结束确定下界。所有专业都确定上下界后，向用户展示计算结果。该算法时间复杂度为O(m\*n)，m为专业个数，n为考生分数跨度。

## 4.5 信息展示模块

### 4.5.1 导航栏模块

导航栏模块位于所有网页的最上方，分为普通用户导航栏和管理用户导航栏，为使用者提供各种系统功能的链接。导航栏代码位于所有包含导航栏的html文件中。

### 4.5.2 主页模块

主页模块分为两种主页，一种是管理员主页，模板代码位于information文件夹的templates/information/admin\_index.html中；另一种普通用户主页，模板代码位于同一路径下的index.html中。在information/views.py中的index函数会根据请求用户的权限调用这两个模板中的一个。管理员主页包含管理员所能使用的各种功能，普通用户主页包括普通用户所能访问的各种功能、老师联系方式展示和校园风光展示。

### 4.5.3 表单展示模块

表单展示模块位于所有需要显示表单的页面，以考生信息收集页面为例，其模板位于information文件夹的templates/information/new\_stu.html。本系统创建表单时使用的是modelform，前端可以通过后端传过来的表单变量form，使用{{form.as\_p}}自动创建表单信息用于展示。

### 4.5.4 表格展示模块

表格展示模块用于所有需要多行数据库信息的页面，以考生信息收集汇总为例，其模板位于information文件夹的templates/information/admin\_display.html。其中展示列需要根据数据库属性的不同调整，使用系统传来的数据进行循环遍历输出至单元格。表单排序功能本人是通过JS实现冒泡排序对数据表进行重新排序实现的。

### 4.5.5 文件展示模块

文件展示方面本人采用的是通过端口号+路径的方式让浏览器访问制定文件。具体的做法是在material/urls.py中配置访问路径，而后就可以通过端口号—+路径的方式对文件进行访问，而后在需要展示文件的前端页面上增加指向各个文件的链接。

## 4.6 系统的实现

### 4.6.1 系统的环境配置

本招生咨询系统项目主要使用了python语言，web开发相关技术以及MySQL数据库技术进行开发。浏览器使用Chrome浏览器；web开发技术主要使用Django开发框架，前端包括html，JavaScript，CSS相关技术；MySQL数据库使用MySQL WorkBench提供数据库可视化工具；通过Windows系统自带的IIS(Internet Information Services 互联网信息服务)充当web服务器，实现项目的部署。

### 4.6.2 系统开发

本项目使用Python语言进行开发，使用了Django开发框架，项目文件夹下共有8个文件夹和2个py文件。8个文件夹除了admission、static、media外其余的5个文件夹为Django创建的应用。由于每个界面都需要在views.py文件中编写相应的视图逻辑，因此分出多个应用是为了便于查找对应的逻辑和数据库。

8个文件夹及其各自用途如下：

admission：项目目录，其中的settings和urls需要包括所有其他应用名称，以便项目运行过程中进行调用。

information：所有与考生信息相关的界面，例如考生信息填写、汇总等。

users：所有与用户相关的操作，例如用户注册、登陆等。

adminsite：所有需要管理员权限才能查看的相关信息汇总和预测信息。

managesite：所有需要超级管理员用户权限才能查看的相关信息展示、修改、增加等。

material：与招生资料有关的界面，如上传、删除、查看等

media：存储招生资料的文件夹

static：静态文件存储位置，存储展示的图片，css样式等

2个py文件分别为：manage.py，用于接收命令并交给Django相关部分去运行，生成Django项目时自动创建的；wfastcgi.py，将python与IIS（Internet Information Service 互联网信息服务）链接的工具，在项目部署时需要使用。

### 4.6.3 系统运行说明

通过命令行，到项目所在目录，输入python manage.py runserver启动服务器。在本机可以通过访问http://127.0.0.1:8000/进入系统登陆界面，其他PC端可以通过浏览器，输入项目主机的ip地址和指定端口的方式对系统进行访问。

## 4.7 本章小结

本章对五个业务模块下各个子模块进行了详细设计，通过流程图的方式对子模块中的主要程序执行过程进行了详细的阐述。之后描述了项目的结构，并分别解释了每个应用所包含的代码意义。

# 第五章 系统测试

## 5.1 系统测试的方法介绍

通常系统测试分为两种方法：白盒测试和黑盒测试。本系统使用的时B/S结构，即用户使用时并不了解服务器的运作机制，因此从用户的角度出发，使用黑盒测试是更为合理的。

黑盒测试不需要了解代码逻辑，只测试软件的结构和功能。本测试通过展示给用户的浏览器界面对系统进行测试输入，检测系统能否正确执行设计功能。如果系统出现异常即需要修改。

## 5.2 系统的测试计划

本次系统测试采用等价类划分法进行测试。等价类划分是将程序输入域划分称若干个等价类，并在每个等价类中选取一个具有代表性的测试用例。等价类分为有效等价类和无效等价类。有效等价类代表对于程序而言，数据输入是合理的有意义的；二无效等价类是指输入的数据是无意义的，应当被程序抛弃。

## 5.3 系统功能测试与结果

本系统通过web页面与用户交互，本次测试将一一测试页面的功能。本次测试使用chrome浏览器进行。测试结果中所有截图均位系统执行测试操作后所显示的结果。

### 5.3.1 注册登陆页面测试

#### 表5-1 注册登陆页面等价类划分表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 用户名 | 非空用户名 | 1 | 空用户名 | 5 |
| 未注册用户名 | 2 | 已注册用户名 | 6 |
| 密码 | 登陆与注册密码相同 | 3 | 登陆与注册密码不同 | 7 |
| 注册时两次密码相同 | 4 | 注册时两次密码不同 | 8 |

#### 表5-2 注册登陆界面测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 用例序号 | 测试用例（用户名，密码，确认密码） | 覆盖等价类 | 用例结果 |
| 1 | test，1234，1234（注册） | 1、2、4 | 进入登陆界面，如图5-1所示 |
| 2 | test，1234（登陆） | 1、3 | 进入个人信息完善界面，如图5-2所示 |
| 3 | （空），2222，2222（注册） | 5 | 提示填写用户名字段，如图5-3所示 |
| 4 | test（已注册），2345，2345（注册） | 6 | 提示用户名已占用，如图5-4所示 |
| 5 | test，2345（登陆） | 7 | 提示密码不正确，如图5-5所示 |
| 6 | test1，1234，2345（注册） | 8 | 提示两次密码输入不一致，如图5-6所示 |

#### 

#### 图5-1 注册成功测试图

#### 图5-2 登陆成功测试图

#### 图5-3 空用户名注册测试图

#### 图5-4 用户名重复注册测试图

#### 图5-5 密码错误测试图

#### 图5-6 注册确认密码不匹配测试图

### 5.3.2 表单填写页面测试

系统中所有需要数据存储和变更的地方都涉及到表单的填写。由于表单众多，而且测试样例设计思路以及测试方法一致，本文中将不一一展示所有表单页面的测试过程和结果，以考生信息填写页面作为这一类测试的代表在此进行说明。表单填写页面等价类划分表如表5-3所示，表单填写页面测试用例如表5-4所示。

#### 表5-3 表单填写页面等价类划分表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| 数字信息 | 在有效范围内的数值 | 1 | 文字信息 | 5 |
| 在有效范围外的数字 | 6 |
| 数值信息为空 | 7 |
| 文字/信息 | 在限定长度内的文字 | 2 | 超出限定长度的文字 | 8 |
| 文字信息为空 | 9 |
| 选项信息 | 选择一项 | 3 | 不做选择 | 10 |
| 综合信息 | 考号，姓名没有重复记录 | 4 | 存在考号姓名均一样的记录 | 11 |

由于表单填写项目众多，测试用例中不一一列举每项所填的内容，只指出本测试用例中与覆盖全部有效等价类用例（第一条用例）之间的区别。

#### 表5-4 注册登陆界面测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 用例序号 | 测试用例 | 覆盖用例 | 用例结果 |
| 1 | 所有项目按规定填写 | 1，2，3，4 | 系统保存该条记录，显示成功页面，填写信息如图5-7所示，成功页面如图5-8所示 |
| 2 | 分数栏填写六百13 | 5 | 仅能输入13，自动略过文字信息，如图5-9所示 |
| 3 | 排名位置填写超过65535的数字 | 6 | 提示数字超过范围，如图5-10所示 |
| 4 | 不填写分数信息 | 7 | 提示该项为必填项，如图5-11所示 |
| 5 | 填写考号（用字符串存储）超出14位，例如12345678901234567890 | 8 | 只记录前14位数，之后的自动舍弃，如图5-12所示 |
| 6 | 不填写考号信息 | 9 | 提示考号信息是必填项，如图5-13 |
| 7 | 在文理科选择中不做选择 | 10 | 提示文理科信息为必填项，如题5-14所示 |
| 8 | 使用与1号测试样例相同的信息进行重新填写 | 11 | 告知用户该信息以被记录，指引用户前往修改信息页面进行修改。告知用户信息如图5-15所示，用户修改页面如图5-16所示。 |



#### 图5-7 表单信息正确填写测试图

#### 图5-8 表单填写成功测试图



#### 图5-9 数字栏填写文字信息结构测试图



#### 图5-10 数字栏填写数字超出范围测试图



#### 图5-11 分数栏未填测试图

#### 图5-12 考号栏填写信息超过长度测试图



#### 图5-13 考号栏未填写测试图



#### 图5-14 文理科未选择测试图



#### 5-15 重复填写提示页面测试图



#### 图5-16 考生信息修改页面

### 5.3.3 数据展示页面测试

同表单展示页面一样，数据展示页面也有很多，这里以考生信息汇总页面为例进行测试展示。数据展示页面等价类划分如表5-5所示，测试用例表如表5-6所示。

#### 表5-5 数据展示页面等价类划分表

|  |  |
| --- | --- |
| 编号 | 有效等价类 |
| 1 | 点击任意一个表头一次 |
| 2 | 点击任意表头两次 |
| 3 | 点击修改摁扭 |
| 4 | 点击删除摁扭 |
| 5 | 点击返回首页摁扭 |
| 6 | 点击导出excel摁扭 |

#### 表5-6 数据展示页面用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 测试用例 | 覆盖等价类 | 用例结果 |
| 1 | 点击考号表头一次 | 1 | 所有考生数据按照考号从小到大顺序排列，如图5-17所示 |
| 2 | 点击排名表头两次 | 2 | 所有考生数据按照排名从大到小的顺序排列，如图5-18所示 |
| 3 | 点击第一行数据的修改摁扭 | 3 | 进入第一个数据的修改界面，如图5-19所示 |
| 4 | 点击排名为1的数据的删除摁扭 | 4 | 弹出确认删除提示框，如图5-20所示，点击确认后数据删除，如图5-21所示 |
| 5 | 点击返回首页摁扭 | 5 | 系统跳转到首页，如图5-22所示（这里使用的是管理员账号，主页较之前有所不同） |
| 6 | 点击导出excel摁扭 | 6 | 自动下载一个excel表格，如图5-23所示，打开后便是展示页面上的信息，如图5-24所示。 |



#### 图5-17 考生信息按照考号正序排序测试图

#### 图5-18 考生信息按排名逆序排序测试图

#### 图5-19 修改考生信息页面测试图

#### 图5-20 删除提示框测试图

#### 图5-21 删除结果测试图

#### 图5-22 返回首页摁扭结果测试图

#### 图5-24 下载excel功能测试图

#### 图5-25 excel导出结果测试图

### 5.3.4 文件展示页面测试

#### 表5-7 文件展示页面等价类划分表

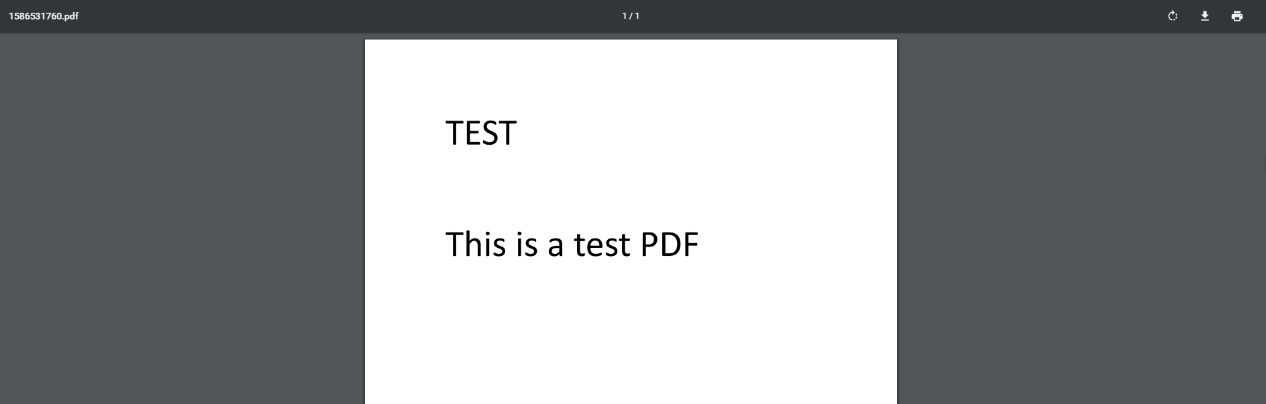
|  |  |
| --- | --- |
| 编号 | 有效等价类 |
| 1 | 点击文件名 |

#### 表5-8 文件展示页面用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 测试用例 | 覆盖等价类 | 用例结果 |
| 1 | 点击文件名 | 1 | 打开pdf查看页面。文件选择页面如图5-26所示，pdf展示页面如图5-27所示。 |



#### 图5-26 招生文件选择页面测试图



#### 图5-27 招生文件展示页面测试图

### 5.3.5 分析预测算法测试

#### 表5-9 预测算法测试等价类划分表

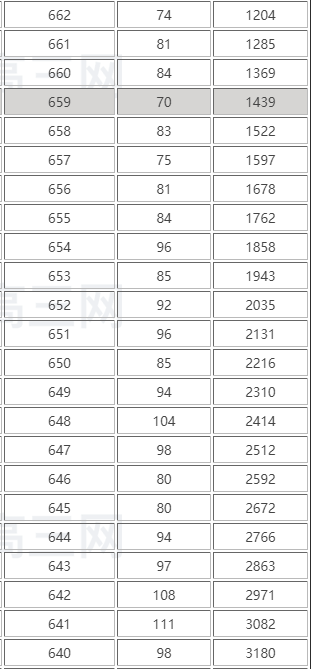
|  |  |
| --- | --- |
| 序号 | 有效等价类 |
| 1 | 收集的考生信息 |
| 2 | 管理员填写的历史信息 |

#### 表5-10 预测算法测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试用例 | 覆盖等价类 | 用例结果 |
| 1 | 随机置入80个考生信息，如图5-28所示，考生分数和排名关系在符合2019年北京高考的分数段表的前提下进行随机生成，排名范围在1200-3200左右 | 1 | 依次，系统可以预测出较为准确分数排名预测关系，如图5-29所示。对比真实情况，如图5-30所示误差较少。 |
| 2 | 管理员输入历史排名信息，历史排名数据来自于2018年北邮各专业在北京录取的排名信息，如图5-31所示。 | 2 | 依照系统预测的分数排名关系和历史信息，系统可以预测出今年的录取分数线，如图5-32所示，对照真是的录取情况，如图5-33所示，大部分专业录取分数线预测误差只有1分。 |

#### 图5-28 随机生成的部分考生信息

#### 图5-29 分数与排名预测结果



#### 图5-30 2019年北京高考分数一分一段表

#### （三列从左到右分别为：分数，本段人数，累计人数）

#### 图5-31 历史信息录入测试图

#### 图5-32 录取分数线预测结果测试图

#### 图5-33 北邮2019年各专业录取分数线

## 5.4 测试结果分析

### 5.4.1 注册登录页面测试结果分析

注册登录页面能够完成注册登录功能模块需求中的所有功能，并且对于错误的填写能够给出对应的提示，成功注册和的登陆后可以进行正确的跳转。界面符合之前的界面设计。

### 5.4.2 表单填写页面测试结果分析

表单填写界面能够完成完善用户信息模块，信息收集模块，增加、修改、查询数据子模块以及表单展示模块需求中的功能，可以结合数据库对数据存储类型的限制，给予用户相应的提示。成功填写表单后能够跳转至成功界面，重复填写表单后能够帮助用户搜寻并展示之前填写的数据。有效表单内容能够准确无误的存储到数据库中。展示界面符合界面设计

### 5.4.3 数据展示页面测试结果分析

数据展示界面能够完成数据展示模块和删除数据模块的功能需求，所有展示数据可以根据用户需求正确地按照正序或逆序进行排序，删除数据之前会弹出提示让用户进行二次确认。页面符合页面设计

### 5.4.4 文件展示页面测试结果分析

文件展示页面能够完成文件展示模块的功能需求，其能够展示所有文件的文件名，并且在用户点击之后能够正确的显示用户请求的文件内容。

### 5.4.5 分析预测算法测试结果分析

分析预测算法能够完成分析预测功能模块的功能性需求，测试模拟了2019年的北京市招生咨询场景，利用2018年的历史信息，对分数排名关系和分数线进行了预测，比较2019年的真实数据后，预测结果较为准确。

## 5.5 本章小结

本章主要介绍了系统的测试过程，从选择黑盒测试的等价类划分法，到划分等价类、设计测试用例、进行测试、比较测试结果。进行了较为完整的测试步骤。本章依照用户页面类别进行分类，分出了注册登陆页面、表单填写页面、数据库展示页面以及文件展示页面。同时还对系统的核心预测算法进行了测试。测试结果表明，本系统能够完成了第二章中分析得出的各类需求。

# 第六章 结束语

## 6.1 论文工作总结

本文主要介绍了招生咨询资讯系统的设计与实现方案。从课题的背景意义，依次介绍了课题任务、系统需求分析、系统总体设计、系统功能模块设计和系统测试方法及结果。

本系统最终实现的功能包括：用户的注册与登陆、用户信息的完善与修改、收集考生信息、上传展示招生材料、展示考生信息汇总并生成报表、通过考生信息和历史信息分析预测出分数排名关系和录取分数线，展示给管理员用户。

在拿到这个课题时，本人首先考虑的是使用C/S架构还是B/S架构。在之前的课程实践中，这两种架构的项目本人都参与制作过。结合自身经验和网上评论，本人认为B/S框架开发的系统通用性更强，安装使用更加便捷，这就是我最终决定使用B/S架构的原因。决定了架构后本人继续研究了两种基于Python语言的web开发框架，Django开发框架和Flask开发框架。最终选用Django开发框架的原因有两点：一是其使用的MTV框架模式，逻辑较为清晰；二是大量的第三方插件的加入，使得其功能十分完备。之后在这几个月的时间里，本人从学习Python、Django、MySQL、ISS配置方法，到需求分析，软件设计，编码实现这一系列工作完成后，我算是独立完成了一个较为完整的项目。在这过程中，我不仅在编码方面有了很大的提高，在解决问题的能力上也有所突破。同时还为我养成了较为良好的软件开发习惯，为软件设计积累了一些宝贵的经验，对我今后的学习和工作有很大的帮助。

## 6.2 问题和展望

本文所设计的系统目前能够完成本课题在初期所提出的各项需求，系统中还存在一些可以优化的问题：

1、算法触发时机问题：本系统目前实现的预测分数和排名关系预测算法是在用户新增数据以及用户请求展示的时候。如果删去前者，则可能导致在用户请求展示时响应时间过长，如果删去后者，则更改数据的信息不会被记入统计信息中。因此两者缺一不可，但是后者触发时机可以改为在用户修改数据时触发，这样当管理员请求页面时就可以直接请求数据库数据而不需要对学生信息进行遍历。然而我认为这样的更改会导致修改数据后响应时间过长，且只要管理员两次请求页面操作之间有多于两人修改信息，则总运行时间应该会超过现有的系统。但是修改自己排名和分数数据概率有多大，以及修改触发时机后会对修改数据这一功能造成多大影响，这两点需要进行实际测试后，才能得出结论。

2、页面美观问题：本系统在向考生展示的各类页面做了统一的页面，然而管理员页面中的数据展示页面没有使用任何样式。原因是表格中项目众多，而给网站设计的模板仅使用了屏幕正中的一小块区域，不足以完整展示表格信息，因此没能套用其他页面的样式。

3、同时相应问题：本系统有些操作不能实现两人在同一秒钟做同一个请求。例如上传招生材料，由于在系统中招生材料使用秒数命名，如果两人在同一秒同时上传两个不同的文件，系统肯定只能保存一个。但由于个人能力以及设备原因，没有找到合适的方法测试这种情况，因此目前本人不了解系统如何做出取舍以及是否会给出错误提示。

4、提示信息语言问题：由于本系统大多表单检查的方法是使用Django框架自带的函数，自带的提示信息全部为英文，该提示信息可以使用中文进行更改，不过需要修改Django框架的源文件。考虑到在服务器部署时候不会使用本机的Django相关文件，因此本项目中未作这个方面的优化。

# 参考文献

[1] Magnus Lie Hetland 著，袁国忠 译 Python基础教程（第3版）人民邮电出版社 2018-02-01

[2] Eric Matthes 著，袁国忠 译 Python编程从入门到实践 人民邮电出版社 2016-07-01

[3] [美] Wesley Chun 著，孙波翔，李斌，李晗 译 Python核心编程（第3版）人民邮电出版社 2016-05-01

[4] [美] Julia Elman,[美] Mark Lavin 著， 侯荣涛 吴磊 译 轻量级Django 中国电力出版社 2016-11-01

[5] 李健 著 Django 2.0 入门与实践 清华大学出版社 2018-11

[6] 胡晓霞 著 HTML+CSS+JavaScript网页设计从入门到精通 清华大学出版社 2017-10-01

[7] [美] Zaitsev,P.，[美] Tkachenko,V. 高性能MySQL（第3版） 电子工业出版社，2013.5

[8] 唐汉明，翟振兴，关宝军，王洪权，黄潇 深入浅出MySQL：数据库开发、优化与管理维护(第2版) 人民邮电出版社，2014.1

[9] 余后荣. 浅析高校招生宣传工作的理性思考[J]. 新教育时代电子杂志(教师版), 2014, 000(027):65.

[10] 王咏梅. 谈高校招生宣传工作的挑战与思考[J]. 汉江师范学院学报, 2017(05):127-130.

[11] 段学强. 对高考志愿填报录取线以及录取概率的预测[J]. 赢未来, 2018, 000(007):P.289-289.

# 致 谢

随着本次毕业设计接近尾声，我大学四年的本科生活也即将结束。在次感谢所有在四年学习生活中帮助过我的每一个人。

首先，感谢我本次毕业设计的指导老师，杜晓峰老师。杜晓峰老师在大四上学期选题后就及时的与我沟通，了解我的情况。杜老师在寒假之前就组织组内学生，讲解毕设的题目、流程和计划，让我可以利用寒假的时间为毕设做好充足的准备。毕设正式开始后，杜老师要求我每周撰写一周工作总结和下周计划，发现我有进度问题或安排问题都会及时与我沟通。感谢杜晓峰老师的督促，帮助我按时完成这次毕业设计。

其次，感谢我的班主任杨亚老师。在我加入计算机学院实验班这三年半的时间以来，杨亚老师多次组织我们参观互联网企业，安排优秀学长传授经验，邀请知名教授普及前沿的技术知识。即便是杨老师在美国访学期间，他也积极为我们联系学长，传授他们的经历。感谢杨亚老师这几年辛苦准备这些课余活动，给予我们提前接触业界的机会，了解到学长们的成功经历，使我对未来有了更加清晰地规划。

最后感谢北京邮电大学对我的培养。是北邮为我创造了这个平台，让我能够认识这些知识渊博的老师，能够接触到这些热情开朗的学长学姐，能够与这些乐观活泼的同学一同学习，一同进步。