# 目 录

[目 录 1](#_Toc89190776)

[摘 要 I](#_Toc89190777)

[Abstract II](#_Toc89190778)

[第1章 前 言 3](#_Toc89190779)

[1.1 本项目的研究背景及意义 3](#_Toc89190780)

[1.2 关于本项目的国内外研究现状分析 3](#_Toc89190781)

[1.3 项目的开发范围及目标 3](#_Toc89190782)

[1.3 论文结构简介 4](#_Toc89190783)

[第2章 技术原理 4](#_Toc89190784)

[2.1 系统开发相关技术介绍 4](#_Toc89190785)

[2.2 系统开发工具介绍 5](#_Toc89190786)

[第3章 高考志愿填报系统的需求建模 6](#_Toc89190787)

[3.1 高考志愿填报系统角色定义 6](#_Toc89190788)

[3.2 高考志愿推荐填报系统功能性需求分析 7](#_Toc89190789)

[3.3 高考志愿推荐填报系统非功能性需求分析 9](#_Toc89190790)

[3.5 高考志愿推荐填报系统系统的数据流图 10](#_Toc89190791)

[第4章 概要设计 13](#_Toc89190792)

[4.1 系统架构及原理 13](#_Toc89190793)

[4.2 业务用例的实现 15](#_Toc89190794)

[4.4 数据库设计 17](#_Toc89190795)

[第5章 模块设计 21](#_Toc89190796)

[5.1 XXX模块的设计与实现（租房用户模块的设计与实现） 21](#_Toc89190797)

[图5.1 租房用户的注册流程 23](#_Toc89190798)

[5.2 XXX模块的设计与实现 26](#_Toc89190799)

[图5.1 租房用户的注册流程 26](#_Toc89190800)

[5.3 XXX模块的设计与实现 27](#_Toc89190801)

[第6章 部署和应用 28](#_Toc89190802)

[6.1 本章参考写法 28](#_Toc89190803)

[第7章 总结与展望 29](#_Toc89190804)

[7.1 本章参考写法 29](#_Toc89190805)

[参考文献 30](#_Toc89190806)

[致 谢 31](#_Toc89190807)

|  |
| --- |
| 高考志愿推荐填报平台设计与实现 |

# 摘 要

时光荏苒，转眼间网络时代到来了，我们的生活越来越离不开网络。而刚刚高考完的学子们对于高考志愿的填报，也渐渐开始从线下的问老师、问学长，转变成了自己在线上查资料。而填志愿时如果能知道各大高校的招生数据、发展重点、学校优势和地理位置等数据至关重要。

这篇文章研究的高考志愿推荐填报系统 结合了很多大学的官方公开数据，给刚刚高考完的有志青年，总结了关于志愿填报的各个学校和专业详细数据，并且还包括了少数民族加分政策、模拟志愿填报、特殊类招生政策、咨询吐槽、满意度和就业薪资等信息。学生再也不用自己到处翻阅学校资料！这篇文章主要工作内容包括：

1. 首先分析当前高考招生制度的现状，实现省份和学校、登录注册、 专业库等模块的设计，具体包括教育部招生政策规定、各校招生政策规定、分数线、少数民族优惠政策、学校信息、报送数据信息、专业库信息等等。
2. 完成对大规模数据的爬取、清理、分类、存储等等。
3. 对于系统设计进行需求分析、设计与建模。
4. 通过ER图、UML图对数据库进行数据表的设计。
5. 对系统关键功能点或重要输入/输出的测试分析。

|  |  |
| --- | --- |
| 关键词： | 高考；志愿填报； Java Web开发；爬虫；SpringBoot；SpringCloud; |

|  |
| --- |
| **Design and implementation of college entrance examination voluntary recommendation filling platform** |

# Abstract

As time goes by, the Internet era is coming, and our life is more and more inseparable from the Internet. Students who have just finished the college entrance examination have gradually changed from asking teachers and seniors offline to checking information online. When filling in volunteers, it is very important to know the enrollment data, development priorities, school advantages and geographical location of major colleges and universities.

This article studies the recommendation system of university entrance examination, which combines many official data of universities, gives the youths who have just completed the college entrance examination, summarizes the detailed data on schools and professional schools that are filled out voluntarily, and also includes information on ethnic minorities' bonus policy, simulation of voluntary reporting, special enrollment policies, make complaints about Tucao, satisfaction and employment salaries. Students no longer have to look through school materials by themselves! The main contents of this article include:

* 1. Firstly, analyze the current situation of the college entrance examination enrollment system, and realize the design of provinces and schools, login and registration, professional database and other modules, including the enrollment policies and regulations of the Ministry of education, the enrollment policies and regulations of each school, score line, preferential policies for ethnic minorities, School information, submitted data information, professional database information and so on.
  2. Complete the crawling, cleaning, classification, storage, etc. of large-scale data.
  3. Carry out requirement analysis, design and modeling for system design.
  4. The data table of the database is designed through ER diagram and UML diagram.
  5. Test and analysis of key function points or important input / output of the system.

|  |  |
| --- | --- |
| **Keywords：** | **College entrance examination; Voluntary filling; Java web development; Reptiles; SpringBoot； SpringCloud;** |

# 第1章 前 言

## 1.1 本项目的研究背景及意义

### 1.1.1 项目研究背景

自隋唐科举制以来，高考一直是人们一生重大转折点，国家对于高考的重视程度从未落下。考生不仅需要考出优异的成绩，考后的志愿填报也极其重要。据我统计，考生填报志愿可能遇到这些问题：

1. 很多考生仅仅通过招生简章 就判断出适合自己的学校和专业，过于片面，并不能得到最满意的结果。

2. 许多考生在填报志愿时，筛选范围特别小，以至于考生错过了更好的学校，是对自己十分不利的一件事。

3. 很多考生对于 大学专业类型数量、专业课程数量、学习的困难程度、就业优势劣势等等全都一知半解，很有可能导致，考生拿到录取通知时进入学校之后，发现自己根本不适合这个学校，此时后悔的代价是极大的。复读数量大多也是来自于此。

### 1.1.2 项目开发意义

很多商业机构设计了了高考志愿填报系统，但是基本都只能实现简单的筛选功能，稍微强一大点的功能要么付费，要么没有。本文通过对各大学校历年录取的信息数据挖掘，通过计算，得出最适合考生的学校和专业，并且展示出各个学校的招生信息，各个专业信息等等，帮助考生完成高质量填报。

## 1.2 关于本项目的国内外研究现状分析

随着我国经济稳定快速发展，科技越来越发达， 信息时代越来越普及，生活越来越便捷，足不出户即可通过互联网解决生活中的大多数问题。目前关于高考志愿填报推荐的文章不多，徐兰静把协同过滤方法结合到系统中，考生对应系统中的用户，通过用户填写的信息从历年招生数据中推荐出若干合适的学校和专业，并分析其有效性；许飞的数据挖掘技术结合Web服务，实现了基于web服务的高考志愿分析数据挖掘系统开发；将数据可视化到网页，是高考志愿填报系统的大进步。

本系统只针对我国考生，我国国情，未考虑外国学校。

## 1.3 项目的开发范围及目标

本项目对全国所有学校和专业进行爬取、统计、分类和总结。在模拟志愿填报时通过计算，得出最适合考生的学校和专业，并且展示出各个学校的招生信息，各个专业信息等等，帮助考生完成高质量填报。

## 1.3 论文结构简介

本文从高考志愿填报中存在的问题入手，对高考考生进行了分析和调研，了解考生填报志愿可能遇到的问题，并在系统中解决这些问题。依据该模拟志愿填报系统的需求分析，完成了模拟志愿填报系统的设计与开发，实现了登录注册、院校库、专业库、满意度、吐槽和模拟填报等功能。本论文以软件开发的大致流程来介绍本高考志愿填报系统，具体内容如下：

第一章节是前言，讲述高考志愿填报系统的开发的背景和意义，综述了国内当前关于在线租房网站的现状，确定了本系统的开发范围及目标。

第二章节是技术与原理，是讲述该项目需要用到的技术以及该项技术实现的原理，以及讲述开发的开发工具，还有系统开发环境的介绍。

第三章节是需求建模，首先讲述了系统可行性分析，然后对系统的需求做了个详细的分析，然后对列出系统角色的定义，最后画出系统类图。

第四章节是概要设计，首先讲述了系统技术架构，然后介绍了系统的物理架构包括服务器的选择和数据库的选择，最后是讲述了对数据库架构的设计 。

第五章节是模块设计，本章节是详细讲述了系统的各个模块设计，分别从各个模块进行了讲述，让读者能更清晰的了解本系统。

第六章节是部署与应用，分别从部署环境，以及更改环境所需要注意的事项都完整的讲述了。

第七章是本论文的尾声，是本人对此次项目的总结。

# 第2章 技术原理

## 2.1 系统开发相关技术介绍

本系统的各个架构和插件均采用稳定快速的版本，尽最大可能提高版本兼容性，和安全

采用jdk1.8（Java语言）编写，大多数公司的选择也都是jdk1.8，因为从jdk6升级到jdk8比较容易，再往上就有兼容性风险了。

Maven使用3.6.0，它是成熟稳定的版本，相比于3.6.3而言，3.6.3可能出现一些未知的错误。

SpringBoot使用2.1.4. RELEASE版本，它兼容SpringCloud的Greenwich.SR1版本。目前springCloud官方最新的长期更新的稳定版本为Greenwich.SR1

Mysql使用8.0.21，MongoDB使用4.2.17，redis使用6.2.4。都是比较稳定的版本。

前端使用Vue3组件化开发，Vue3相比Vue2来说更快、更小、加强API设计一致性，并且向下兼容Vue2。

## 2.2 系统开发工具介绍

### 2.2.1 开发工具IDEA

本系统使用IDEA 开发，它是业界公认的最好的开发工具之一。如全局查找和替换、内置Mysql和MongoDB等多种数据库客户端、批量添加Swagger注释、支持各种版本工具（Git、SVN）还有它的灵魂：重构，是开发系统的不二选择。数据库存放于阿里云的服务器上，结合GitHub版本控制，可以达到在不同的主机上代码同步和数据库同步的效果。

### 2.2.2 MySql

IDEA中同时包含了database插件，用于简单的图形化编辑MySQL等数据库。本系统同时也使用了Navicat Premium 15用于mysql的高级设置和编写。

### 2.2.3 Finalshell

FinalShell是一款免费的国产的集SSH工具、服务器管理、远程桌面加速的良心软件。它有着独特强大的功能，shell和sftp同屏显示,同步切换目录。本系统的MySQL、MongoDB、Redis均安装在阿里云服务器。

# 第3章 高考志愿填报系统的需求建模

本章基于高考志愿填报系统设计与开发，接下来对系统进行详细的需求分析。主要包括本章通过功能需求分析和性能需求分析两个方面来介绍高考志愿填报系统，其中用例分析采用UML图例来阐述。

## 3.1 高考志愿填报系统角色定义

高考志愿推荐填报系统可分为管理端和客户端两部分。管理端通过账号密码进入系统，对用户、会员、院校、专业等信息的管理以及对客户端功能的维护和更新。客户端实现用户注册，升级会员、院校库、专业库、招生数据、吐槽等功能。

用例图分别为：

1．管理员：主要包括登录注册，管理用户等用例，用例图如图3.1.1所示：



图3.1.1 高考志愿推荐填报系统管理员用例图

2．用户：主要包括登录注册、查看院校库、专业库、模拟志愿填报等用例，用例图如图3.1.2所示：



图3.1.2 高考志愿推荐填报系统用户用例图

## 3.2 高考志愿推荐填报系统功能性需求分析

### 3.2.1 管理端各项功能的需求分析

1. 用户管理功能实现用户信息的管理，可进行增加、删除、修改等，有助于管理用户。用户信息增加会员，会员可以模拟志愿填报。
2. 院校管理功能实现全国所有学校的管理，可进行增加、删除、修改学校等。
3. 具体院校管理功能实现学校信息的管理，可进行增加、删除、修改学校信息等。
4. 专业管理功能实现全国所有专业的管理，可进行增加、删除、修改专业等。
5. 具体专业管理功能实现各个学校的专业信息的管理，可进行增加、删除、修改专业信息等。
6. 满意度控制功能实现学校、专业的满意度设置，并根据满意度推荐学校和专业



图3.2.1 管理员模型图

### 3.2.2 客户端主要的功能模块包括:

1. 登录功能可分为两个部分，会员登录和非会员登录。会员登录后可以直接跳转到模拟填报页面，与非会员相比，非会员不能使用这个功能。
2. 院校库功能可以根据名称、所在地、主管部门基本、院校类型、专业查询所有学校。每个院校资料包括基本资料、专业满意度排好序的专业、吐槽区、专业介绍、招生简章、录取规则、奖学金设置、食宿条件、往年录取信息查询。
3. 专业库功能根据本科（普通教育），本科（职业教育），高职（专科）分类显示所有门类，门类再划分出专业，专业再划分出专业名称，比如工学门类分为力学类、计算机类等，计算机类分为软件工程、网络工程等。查看软件工程这门专业的具体信息（专业介绍、毕业生规模、男女比例、相似专业、考研仿效、就业方向、薪酬水平等等）、开设院校、开设课程、专业解读
4. 招生简章功能根据院校所在地、主管部门类别、院校类型、学历层次、院校特性、院校名可查询到所有院校的招生简章信息。还包括招生政策、名单公示等。
5. 咨询功能包括咨询周和咨询室（吐槽评论）



图3.2.2 管理员模型图

## 3.3 高考志愿推荐填报系统非功能性需求分析

高考志愿推荐填报系统是为考生用户各种信息查看和模拟填报的系统。一个系统如果想拥有良好的用户体验，就要拥有良好的性能特性。下面就针对数据精确度、可使用性及兼容性、时间特性、安全保密性、可维护性几个方面的性能需求来详细介绍本高考志愿推荐填报系统。

### 3.3.1 数据精确度

高考志愿推荐填报系统在各种资料等信息录入时，需要保证填写的信息经过数据格式和现实逻辑的校验后才录入系统。同时需要保证数据显示的格式符合现实认知，必要数据的精确程度不会过繁过简，而是符合现实常识和逻辑需要。

### 3.3.2 可使用性及兼容性

本系统具备友好的用户界面，使用方便，易于维护，操作简单，易于被用户接受。用户只需要可以熟练的操作计算机，即可方便使用，而且使用此系统可以以大减少管理人员的负担。并且兼容各个常用的浏览器。

### 3.3.3 时间特性

在数据库存放时会同步存放于redis缓存中，再取出时优先在redis中查询，增加查询效率。同时需要模糊搜索数据量比较大的表格时使用ElectricSearch实现快速查询，同时同步数据库的数据。

### 3.3.4 安全保密性

采用token实现单点登录，在密码数据存入数据库时用SpringSecurity中的BCryptPasswordEncoder加密算法对密码加密后再存入数据库。保证数据在采集、传输、处理过程中不被愉窥、窃取、篡改。

### 3.3.5 可维护性

在各个操作时写入日志文件和打印日志，并且编写响应的注解，使系统更加易于维护。

## 3.5 高考志愿推荐填报系统系统的数据流图

数据流图也称为数据流程图（DFD），它可以让系统分析者更加直观清楚地明白这个系统是做什么的可解决什么问题的一种图形工具。

### 3.5.1 顶层层数据流图

第0层数据流图展示系统的参与角色及对应角色间对系统进行的操作。从大体上展现了本系统的基本功能。



图3.5.1 顶层数据流图

### 3.5.2 第一层数据流图

第一层的数据流图中，实体类包括考生用户和管理员。考生用户的数据流程只概括成了模拟填报、查看学校和专业，填报可以查看学校和专业的相关信息，或者模拟志愿填报生成一个填报后的合适的学校和专业数据。管理员管理学校和专业等信息。



图3.5.2 一层数据流图

### 3.5.3 第二层数据流图

第二层数据流图如图3.5.3所示。第二层的数据流图中，更详细的描述了本系统的业务逻辑。用户登录进入系统注册会员，然后进行模拟志愿填报，得到可能录取的大学学校和专业。进入系统同时还可以进行学校各种资料信息和专业的各种专业资料信息的查询，并查看各个满意度信息以便于更直观的了解更适合自己的学校和专业。除了查看各个信息意外，还可以咨询和吐槽，但是需要经过管理员的审核防止出现不良吐槽，审核完成才能存入吐槽表信息中。管理员还可以管理吐槽信息，同时将修改后的信息存入吐槽表中。在管理用户、学校和专业等时也是同理。



图3.5.3 二层数据流图

# 第4章 概要设计

本章基于高考志愿推荐填报系统的需求分析，给出高考志愿推荐填报系统的总体设计架构、功能模块的具体细节设计以及数据库的设计等。

## 4.1 系统架构及原理

高考志愿推荐填报系统采用前后端分离架构实现，目的是开发和使用时方便，提高使用效率，降低使用者运行成本。如下图



图2.1.1 前后端分离结构图

后端框架采用SpringCloud框架模块化开发，降低模块间耦合，提高开发效率。



图2.1.2 本系统的SpringCloud架构图

代码采用MVC结构（分层开发设计模式）进行设计，层次之间相互独立，如下图



图2.1.3 MVC结构

## 4.2 业务用例的实现

用户在首页点击登录，进入登录界面，输入账号密码登录系统，然后点击开通VIP按钮，选择套餐后付款，跳转到模拟志愿填报页面。在模拟志愿填报界面填写各项信息之后点击智能填报按钮，展示出所有相符合的学校与专业。如下图4.2.1描述了用户登录系统后的主要操作顺序：



图4.2.1 模拟高考志愿填报系统顺序图

## 4.4 数据库设计

### 4.4.1 数据库概念设计

从模拟志愿填报系统功能的角度出发，本文设计了每个实体与主要功能的E-R图，如图4.4.1所示。本次论文整体设计的E-R图模型主要涉及了学校信息、专业信息、地址信息、分数信息四大类别。

学校信息包括：学校的院校名称、院校所在地、教育行政主管部门、院校类型、学历层次、是否一流大学建设高校、是否一流学科建设高校、是否设有研究生院、满意度；学校信息更新时间、学校领导、学校简介、周边环境、通讯地址、联系电话、重点实验室、重点学科、可授予的学位、师资力量、学生人数、录取批次、对艺术类招生的录取办法、加分政策、对少数民族考生的特殊政策、对往届生的录取政策、转专业的实施办法、其它、奖学金设置、困难生资助办法、宿舍、食堂、其它、往年录取信息。

专业信息包括: 专业名、专业代码、专业满意度、从属专业类、专业介绍、毕业生规模低点、毕业生规模高点、男生比例占比、综合满意度、综合满意度人数、办学条件满意度、办学条件满意度人数、教学质量满意度、教学质量满意度人数、就业满意度、就业满意度人数。

地址信息包括：省份名。

这三大类构成了整个系统的核心，为用户查询提供方便。其中学校和专业是之间是多对多的关系，从而产生了两个个新的实体，第一个实体是某学校的某专业的具体信息表，包括学校id、专业id、学制、培养目标、推荐指数、推荐人数、综合满意度、综合满意度人数、办学条件满意度、办学条件满意度人数、教学质量满意度、教学质量满意度人数、就业满意度、就业满意度人数；第二个实体是某学校的某专业的以往若干年分数线情况。省份和学校是一对多关系，但省份和专业是多对多关系，产生了一个实体：每个专业在各个省的的薪资水平情况。专业和课程之间是多对多的关系。



图4.4.1 模拟志愿填报系统的部分E-R图

### 4.4.2 数据库主要表设计

模拟志愿填报系统主要使用SQL数据库来存储信息，总共设计了三十多张属性表，主要表如下所示：

1. 城市表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 为空 | 自增 | 字段说明 |
| id | int(11) | NO | 自增 | ID |
| name | varchar(50) | YES | / | 市名 |
| province\_id | int(11) | YES | / | 省份id |

1. 课程表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 为空 | 自增 | 字段说明 |
| id | int(11) | NO | 自增 | ID |
| name | varchar(50) | YES | / | 课程 |

1. 课程专业对照表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 为空 | 自增 | 字段说明 |
| id | int(11) | NO | 自增 | ID |
| tb\_course\_id | int(11) | NO | / | 课程ID |
| tb\_major\_id | int(11) | NO | / | 专业ID |
| course\_difficulty | decimal(2,1) | YES | / | 课程难易度 |
| course\_practical | decimal(2,1) | YES | / | 课程实用性（于工作、个人成长 |

1. 主管部门表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 为空 | 自增 | 字段说明 |
| id | int(11) | NO | 自增 | ID |
| name | varchar(30) | YES | / | 部门名 |
| tb\_department\_type\_id | int(11) | YES | / | 主管部门类别id |

1. 主管部门类别表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 为空 | 自增 | 字段说明 |
| id | int(11) | NO | 自增 | ID |
| name | varchar(30) | YES | / | 主管部门类别 |

1. 历年录取分数线表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 为空 | 自增 | 字段说明 |
| id | int(11) | NO | 自增 | ID |
| tb\_school\_major\_id | int(11) | NO | / | 学校专业对应表id |
| tb\_year\_id | int(11) | NO | / | 年份表id |
| grade | int(11) | NO | / | 分数 |

1. 学历层次表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 为空 | 自增 | 字段说明 |
| id | int(11) | NO | 自增 | ID |
| name | varchar(30) | YES | / | 学历层次名 |

1. 专业表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 为空 | 自增 | 字段说明 |
| id | int(11) | NO | 自增 | ID |
| name | varchar(30) | YES | / | 专业名 |
| major\_code | varchar(30) | YES | / | 专业代码 |
| satisfaction | decimal(2,1) | YES | / | 专业满意度 |
| tb\_major\_professional\_category\_id | int(11) | YES | / | 从属专业类 |

1. 门类表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 为空 | 自增 | 字段说明 |
| id | int(11) | NO | 自增 | ID |
| name | varchar(30) | YES | / | 门类名 |
| tb\_major\_levels\_type\_id | int(11) | YES | / | 从属专业层次类型 |

1. 学校表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 为空 | 自增 | 字段说明 |
| id | int(11) | NO | / | ID |
| name | varchar(30) | YES | / | 院校名称 |
| tb\_province\_id | int(11) | YES | / | 院校所在地 |
| tb\_department\_id | int(11) | YES | / | 教育行政主管部门 |
| tb\_type\_id | int(11) | YES | / | 院校类型 |
| tb\_levels\_id | int(11) | YES | / | 学历层次 |
| top\_university | tinyint(1) | YES | / | 一流大学建设高校 |
| top\_discipline | tinyint(1) | YES | / | 一流学科建设高校 |
| graduate\_school | tinyint(1) | YES | / | 研究生院 |
| satisfaction | decimal(2,1) | YES | / | 满意度 |

# 第5章 模块设计

## 5.1 用户模块的设计与实现

### 5.1.1 用户模块的功能描述

用户子模块的业务逻辑设计包含考生用户注册、登录、注销、个人信息管理、购买会员、模拟志愿填报、吐槽咨询查看所有高考相关信息功能。用户注册需提供手机号，点击发送验证码，手机收到验证输入框中核实即可注册成功。之后用户不仅可以用密码登录，也可以使用短信验证码登录。而密码在数据库中也是经过了SpringSecurity的加密处理后以密文的形式存储。在使用完网页关闭了网页后，浏览器中的Cookie 保存了登录信息，只要用户不手动注销并且Cookie不过期，下次打开浏览器永远处于登录状态不需要验证登录。用户登录之后考可以更改绑定的手机号、看到个人信息和修改信息、看到各种高考相关的信息，模拟志愿填报（需要先购买VIP），吐槽等等。

### 5.1.2 考生用户操作流程

用户模块的界面设计实现用户登录、注册、个人信息的查看与修改、购买会员、模拟志愿填报、吐槽咨询查看所有高考相关信息等界面，用户登录注册时，前台会给出相应的填写信息提示，用时在输入信息错误时给出相应的反馈，在个人信息查看界面，用户可以查看自己的手机号，姓名，昵称，并且在修改之后自动更新数据。用户可以模拟志愿填报，或者在主页进行各种高考信息的查看，具体流程如下图所示：



图3.4 用户流程图



图3.5 管理员流程图

## 5.1 用户的注册流程

### 5.1.3 考生用户功能界面设计

模拟志愿填报推荐平台网站分为前台和后台，前台有注册和登录页面，商家后台只有登录页面，但前台和后台均使用同一个登录界面。管理员登录后再平台进行各种信息的管理，其中就包括用户管理。对于用户通过手机号和验证码的形式注册，也同时激活手机号验证码登录的功能，同时设置验证码的时间限制，保证注册和登录的安全进行，用户登录后方可参与各种操作。



图5.1.2 前台登录界面设计图

### 登录注册功能类图结构

为了更好地描述功能与数据表之间的关系，本文设计了模拟志愿填报系统的UML图，图5.1.4所示。UML图主要描述了系统中类与类之间的关系，其中包含有聚合关系、关联关系、组合关系、泛化和依赖关系等。

(1) 泛化

泛化表示类与类之间的继承关系，或者接口与接口之间的继承关系。它表示一个类得到了另一个类的所有属性和功能，并且还能在其基础上增加一些自己独有的功能，用空心三角形箭头+实线表示，箭头指向父类。

(2) 实现

实现表示类对接口的实现关系。它表示一个类实现一个或者多个接口的功能，用空心三角形箭头+虚线表示，箭头指向接口。

(3) 依赖

两个独立的对象，但是其中一个对象使用了另一个对象。用带普通箭头的虚线表示，指向被使用者。

(4) 关联

两个独立的对象，但是存在固定的关系，比如说一个类的成员变量是另一个类。它可以是双向的，也可以是单向的。单向的用带普通箭头的实线表示，指向被关联者，双向的没有箭头。

(5) 聚合

聚合是关联关系的一种。是整体和部分的联系，比如说一个类的属性是另一个类。在语法上无法区分，必须要考察具体的逻辑关系。用带空心菱形的实心线，菱形指向整体。

(6) 组合

聚合也是关联关系的一种，是整体和部分的联系，比如说一个类的属性是另一个类。但它表示没有整体就没有部分，比集合还要强的联系。带实心菱形的实线，菱形指向整体

这些类与类之间的关系不仅显示了模拟志愿填报系统内的信息结构，也描述了系统内的信息行为。其中，用户类的属性主要包括用户名和密码。

图5.1.4 展示了用户子模块的类图结构，其中主要由 UserDao类、UserSerice类、UserController类实现登录和注册。UserDao类继承了Springdata jpa的JpaRepository类用户于数据库的增删查改等操作。在用户注册时需要先发送验证码，进入后台首先通过JwtUtil的parseJWT方法解析头信息，没有头信息（注册或登录时）则为空，然后调用UserController层的sendsms方法，sendsms方法再调用UserDao类的sendSms方法，此方法中首先随机生成一个6位数短信验证码，然后通过RedisTemplate类的opsForValue方法把验证码放入redis中；再通过RabbitTemplate类的convertAndSend方法将验证码和手机号发送到rabbitmq中让另外一个微服务去接收。



图 5.1.4 用户子模块类图

### 5.1.4 注册功能核心代码

/\*\*

\* service层中的发送短信验证码

\*

\* @param mobile 手机号

\*/

public void sendSms(String mobile) {

//1.生成6位短信验证码

Random random = new Random();

int max = 999999;//最大数

int min = 100000;//最小数

int code = random.nextInt(max);//随机生成

if (code < min) {

code = code + min;

}

System.out.println(mobile + "验证码是：" + code);

//2.将验证码放入redis

redisTemplate.opsForValue().set("smscode\_" + mobile, code + "", 5, TimeUnit.MINUTES);//五分钟过期

//3.将验证码和手机号发动到rabbitMQ中

Map<String, String> map = new HashMap();

map.put("mobile", mobile);

map.put("code", code + "");

rabbitTemplate.convertAndSend("sms", map);

}

/\*\*

\* service层中的注册用户

\* @param user 用户

\* @param code 用户填写的验证码

\*/

public void add(User user, String code) {

//判断验证码是否正确

String sysCode = (String) redisTemplate.opsForValue().get("smscode\_" + user.getMobile());

//提取系统正确的验证码

if (sysCode == null) {

throw new RuntimeException("请点击获取短信验证码");

}

if (!sysCode.equals(code)) {

throw new RuntimeException("验证码输入不正确");

}

user.setRegisterDate(new Date());//注册日期

user.setUpdateDate(new Date());//更新日期

user.setLastDate(new Date());//最后登陆日期

user.setIsVip(0);

userDao.save(user);

}

/\*\*

\* Controller层中的用户登陆

\*/

@RequestMapping(value="/login",method=RequestMethod.POST)

public Result login(@RequestBody Map<String,String> loginMap){

User user = userService.findByMobileAndPassword(loginMap.get("mobile"),loginMap.get("password"));

if(user!=null){

String token = jwtUtil.createJWT(String.valueOf(user.getId()),user.getNickname(), "user");

Map map=new HashMap();

map.put("token",token);

map.put("name",user.getNickname());//昵称

map.put("avatar",user.getAvatar());//头像

return new Result(true,StatusCode.OK,"登陆成功",map);

}else{

return new Result(false,StatusCode.LOGINERROR,"用户名或密码错误");

}

}

# 第6章 部署和应用

## 6.1 系统架构

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 应用面向用户层 | 登录 | | | | | | 模拟志愿填报系统 |
| 面向用户基本功能 | | | | | |
| 登录注册 | 查询院校库 | 查询专业库 | 查询满意度 | 吐槽咨询 | 名单公示与特殊招生 |
| 面向管理员基本功能 | | | | | |
| 管理用户 | 管理院校库 | 管理专业库 | 管理满意度 | 吐槽管理 | 名单公示与特殊招生管理 |
| 数据库层 | 账户信息 | 院校库 | 专业库 | 满意度 | 吐槽信息 | 招生名单与特殊招生信息管理 |
| 基础系统层 | 局域网安全平台 | | | | | |
| 硬件基础设施平台 | | | | | |

图6.1 模拟志愿填报系统架构图

## 6.2 实现系统功能所采用技术

|  |  |  |  |
| --- | --- | --- | --- |
| id | 功能点 | 前端技术 | 后端技术 |
| 1 | 注册 | Vue  Nuxt渲染开发  axios技术  json数据交换格式  js-cookie处理cookies  element-ui框架  vue-infinite-scroll瀑布流插件  vue-quill-editor富文本编辑器 | springBoot框架  spring data jpa  spring data redis  spring security  springcloud框架(Eureka,Zuul,Ribbon)  rabbitmq消息队列  elastic search查询  redis、mongodb、redis |
| 2 | 登录 |
| 3 | 浏览主页政策 |
| 4 | 院校库 |
| 5 | 院校满意度 |
| 6 | 特殊类型招生 |
| 7 | 专业库 |
| 8 | 专业满意度 |
| 9 | 专业推荐 |
| 10 | 专业解读 |
| 11 | 模拟志愿填报 |
| 12 | 咨询周 |
|  |  |  |

## 6.3 系统测试与部署

### 6.1 测试方法

#### 6.1.1 黑盒测试

黑盒测试又称功能性测试。它是用来测试系统的各个功能能否正常运行的一种测试方法。一般的测试都在程序的接口处进行，查看系统是否能正常输出信息。黑盒测试着眼于外部结构，从用户的角度出发，根据输入数据与输出数据的对应关系进行相应的测试。

#### 6.1.2 白盒测试

白盒测试又称结构测试，是一种测试用例设计的方法，白盒测试可以让人看清楚是如何进行测试的。它可以对软件的过程性细节做细致的检查。通过在不同点检查程序状态，确定实际状态是否与预期的状态一致。

## 6.2 测试类型

### 6.2.1 界面测试

对每个界面都进行测试。

### 6.2.2 功能测试

对于增删查找等各种功能进行整体测试，保证每一个功能能够正常的运行。

### 6.2.3 综合测试

把界面个功能联合一起测试，检查系统功能是否满足实际需求。

## 6.3 测试过程和结果分析

### 6.3.1 用户登录界面测试

登录模块是进入整个模拟志愿登录填报系统的钥匙，在进行系统测试时，作为首先需要被测试的的环节。测试的过程中主要是通过黑盒测试和白盒测试确定点餐系统在登录时系统运行是否顺畅。进入饭店后，首先会出现的是登录的界面，在账号栏输入已经注册好的电话号，在密码那栏输入密码， 点击登录即可进入手机自助点餐系统，反之，如果密码输入不正确，则无法直接进入会员系统，只能点击注册，注册完以普通用户身份进入选菜。测试登录模块的功能，测试用例如下表6.1所示。

|  |  |
| --- | --- |
| 字段名 | 测试 |
| 测试项 | 登录界面测试 |
| 测试步骤 | 进入饭店到达终端系统  输入错误的用户名密码  注册  输入正确的用户名密码 |
| 测试结果 | 连接成果  登陆失败时弹出失败提示  用户名密码输入框内容清空  登录成果直接进入订餐系统 |
| 测试是否通过 | pass |

表6.1 测试图

（\*

1）描述系统运行环境，可以使用UML的部署图；

2）选择若干典型界面说明系统的重要输入/输出；

3）与已有系统进行多方面的对比说明先进性或特色，或与本系统投入使用前状况对比说明应用效果。

4）对系统关键功能点或重要输入/输出的测试分析

\*）

|  |
| --- |
| 第7章 总结与展望 |

## 7.1 本章参考写法

(\*总结、比较与展望。这一部分要总结自己工作的优点在哪里、不足之处在哪里、进一步的改进方案如何等。如果有类似应用背景的系统，还须比较你的工作与他人工作的优、劣。 \*)

# 参考文献

# 致 谢