

上海第二工业大学

# 本科毕业设计（论文）



题目： 基于微服务的高校在读生及毕业生数据分析平台

英文题目： DATA ANALYSIS PLATFORM  
FOR COLLEGE STUDENTS  
AND GRADUATES BASED ON  
MICRO SERVICE

学号： 20171111135

姓名： 麦自登

班级： 18 软工 A2

专业： 软件工程

学部(院)： 计算机与信息工程学院

入学时间： 2018 级

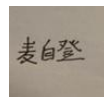
指导教师： 李丽萍

日期： 2022 年 4 月 20 日

## 毕业设计（论文）独创性声明

本人所呈交的毕业论文是在指导教师指导下进行的工作及取得的成果。除文中已经注明的内容外，本论文不包含其他个人已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意。

作者签名：



日期：2022 年 4 月 20 日

# 基于微服务的高校在读生及毕业生数据分析平台

## 摘要

随着我国教育事业的发展, 高校在校生和毕业生人数逐年增加, 而高校毕业生就业问题更是近年来的热门话题。毕业季的就业竞争激烈程度也是逐年上升, 因此高校需要根据社会的需求和反馈进行对已有培养计划的修改与调整, 为社会输送更多合适的人才, 提升在校生的就业竞争能力。在这样的背景下, 高校需要一个在线平台收集高校毕业生和在校生的就业数据, 并且科学合理地高校培养计划提出合适的修改建议。

本文介绍了一个基于 web 的基于微服务的高校在校生和毕业生就业数据分析平台, 该系统使用了 JAVA 语言进行开发, 基于 B/S 模式。而后端使用的架构有 SpringBoot, Mybatis, Eureka, Ribbon, Hystrix。为了提高性能使用了 MYSQL 作为数据库, 使用非关系型数据库 Redis 作为缓存。在前端使用了 Vue 框架进行设计。同时使用了 Git 进行了项目的版本控制。

该系统为高校教室提供了创建问卷的平台, 也为高校毕业生和在校生提供了回到问卷的平台。通过收集用户的作答结果数据, 分析出培养计划中的重点课程, 帮助教师依据社会需求改进高校的培养计划。

**关键词:** Spring Cloud; SpringBoot; Java; 微服务

# DATA ANALYSIS PLATFORM FOR COLLEGE STUDENTS AND GRADUATES BASED ON MICRO SERVICE

## ABSTRACT

With the development of China's education, the number of college students and graduates is increasing year by year, and the employment of college graduates is a hot topic in recent years. The intensity of employment competition in the graduation season is also increasing year by year. Therefore, colleges and universities need to modify and adjust the existing training plan according to the needs and feedback of the society, so as to provide more suitable talents for the society and improve the employment competitiveness of students. In this context, colleges and universities need an online platform to collect the employment data of college graduates and students, and scientifically and jointly put forward appropriate modification suggestions for the training plan of colleges and universities.

This paper introduces a web-based micro service based employment data analysis platform for college students and graduates. The system is developed using java language and based on B / S mode. The back-end architectures include springboot, mybatis, Eureka, ribbon and hystrix. In order to improve performance, MySQL is used as the database and redis, a non relational database, is used as the cache. Vue framework is used in front-end design. At the same time, Git is used for version control of the project.

The system not only provides a platform for creating questionnaires for university classrooms, but also provides a platform for college graduates and students to return to the questionnaire. By collecting the data of users' answers, this paper analyzes the key courses in the training plan, so as to help teachers improve the training plan of colleges and universities according to social needs.

**Key words:** Spring Cloud; SpringBoot; Java; Microservices

## 目录

1	绪论 .....	1
1.1	研究的背景 .....	1
1.2	国内外研究现状 .....	1
1.3	研究意义 .....	1
1.4	本论文特色 .....	1
1.5	本论文拟解决问题 .....	2
2	相关理论与技术概述 .....	3
2.1	开发工具介绍 .....	3
2.1.1	Git 简介 .....	3
2.1.2	IntelliJ IDEA 简介 .....	3
2.1.3	Maven 简介 .....	3
2.1.4	MySQL 介绍 .....	3
2.1.5	Eureka .....	4
2.1.6	Ribbon .....	4
2.1.7	Hystrix .....	5
2.1.8	Redis .....	6
2.1.9	Canal .....	7
2.1.10	RabbitMQ .....	7
3	需求分析 .....	8
3.1	系统目标 .....	8
3.2	系统可行性分析 .....	8
3.2.1	技术可行性 .....	8
3.2.2	操作可行性 .....	8
3.2.3	经济可行性 .....	8
3.3	系统的功能性需求 .....	9
3.3.1	登录用例规约 .....	10

3.3.2 注册用例规约 .....	10
3.3.3 新建问卷 .....	11
3.3.4 填写培养计划课程 .....	11
3.3.5 填写或更新课程路径 .....	11
3.3.6 填写或更新问卷内容 .....	11
3.3.7 查询问卷 .....	12
3.3.8 问卷作答 .....	12
3.3.9 问卷分析 .....	12
3.3.10 发布问卷 .....	13
4 系统设计 .....	14
4.1 功能设计 .....	14
4.1.1 系统结构 .....	14
4.2 数据库表设计 .....	15
5 系统实现 .....	18
5.1 登录功能 .....	18
5.2 用户注册 .....	19
5.3 主页面 .....	20
5.4 创建课程页面 .....	21
5.5 填写问卷相关课程 .....	22
5.6 更新或填写路径 .....	23
5.7 更新或填写问卷内容 .....	24
5.8 调查问卷 .....	25
5.9 查看该用户上传的问卷列表 .....	26
5.10 申请及时分析功能 .....	26
5.11 回答问卷 .....	30
5.12 统计答题人数 .....	31

---

5.13 培养计划中的重要课程路径 .....	32
6 系统测试 .....	34
6.1 功能测试 .....	34
6.2 Jmeter 使用介绍 .....	36
6.3 后端接口测试 .....	37
6.3.1 登录端口 .....	37
6.4 服务抢购端口的测压 .....	38
7 结论 .....	40
致谢 .....	41
参考文献 .....	42

# 1 绪论

## 1.1 研究的背景

传统的纸质问卷因地域和空间的局限性，导致相关调查的效率低下。在当今的互联网时代，线上的问卷作答平台使调查的效率有极大的提升。虽然各大平台能及时反馈调查问卷的作答情况，然而只有少数平台对作答结果进行整理并提供有价值的数据分析结果给用户。在这国内外的线上调查分析平台中，没有特别针对高校毕业生和在校生的数据进行分析 and 调查的。

## 1.2 国内外研究现状

国内：问卷星是一个国内专业的问卷平台，用户可以在上面发布问卷，答题，对部分数据进行一定程度的分析。然而，这些数据调查并没有对高校毕业生进行有针对性的分析，调查范围过大。因此在本国缺乏这一针对高校毕业生调查数据分析的线上平台的情况下，本课题具有研究的价值。

国外：国外目前拥有 **toluna** 等网站，但显然这些数据调查并没有对高校毕业生进行有针对性的分析，调查范围也是比较广泛，因此本课题具有研究的价值。

## 1.3 研究意义

大学生就业问题一直是社会比较关注的热点话题，每年的大学毕业生数量都在屡创新高！2020 年全国高校毕业生更是高达 874 万。高校的不断扩招，致使大学生毕业人数不断增加。高校的扩招不是为了扩招而扩招而是为了国家经济发展的需要，国家的发展需要人才，而人才的根本在于教育。而大学生毕业的去向和就业内容是学校教学水平改善和调整的重要线索。充分为本校（上海第二工业大学）解析大学生的毕业去向有利于学校调整课程和调动校内人力资源结构从而提升教学质量。

## 1.4 本论文特色

使用以 **SpringBoot** 技术为基础，结合 **Spring Cloud** 的组件和 **Redis** 缓存搭建一个以针对在校生的培养计划分析平台。解决服务器的单点故障问题，以及解决分布式锁及缓存与数据库一致性问题。



## 1.5 本论文拟解决问题

- (1) 网站开发的可行性
- (2) 系统开发平台的选择
- (3) 系统功能模块的划分及相互关系
- (4) 数据库的设计问题以及表与表之间的引用关系
- (5) 如何保证数据库的数据能进行高效查找
- (6) 如何处理单点故障
- (7) 缓存与数据库的一致性问题
- (8) 分布式进程之间的数据一致性

## 2 相关理论与技术概述

### 2.1 开发工具介绍

#### 2.1.1 Git 简介

Git 是一个基于分布式的轻灵小巧的版本管理软件，可在本地安装客户端。通过本地客户端可上传至 Gitee 或 Github 等线上仓库。并且支持多人合作书写软件，通过客户端上传到公用仓库的其中一个分支中，最终由审核人员审核后合并到主分支下。

#### 2.1.2 IntelliJ IDEA 简介

全球 3/4 的开发者使用的 JAVA 开发工具，用于自动代码补全助手。并且 IntelliJ IDEA 支持给种版本开发工具的插件，如 Git, SVN 等。并且能支持 Maven 仓库的使用，方便使用者快速搭建一个 Springboot 项目。

#### 2.1.3 Maven 简介

Maven 项目对象模型(POM), 可以通过一小段描述信息来管理项目的构建, 报告和文档的项目管理工具软件<sup>[1]</sup>。在 Maven 项目中, 我们可以通过 pom.xml 文件规定 JAVA 程序引用包的版本, 统一整个 Springboot 项目引用包的版本, 使得程序避免因部分包冲突而造成错误。在开发中, 倒入 JAR 包只需要简单快捷地往 pom.xml 文件上写入配置即可。

#### 2.1.4 MySQL 介绍

MySQL 是一个关系型数据库管理系统, 属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一, 在 WEB 应用方面, MySQL 是最好的应用软件之一<sup>[2]</sup>。本项目使用 MySQL8.0, 在 MySQL8.0 的版本后使用了 InnoDB 作为数据库的默认引擎, 相较于原来的 MyISAM 数据库引擎支持了事

务的操作。InnoDB 引擎使用的聚簇索引在用户 SQL 语句符合最左匹配原则的情况下, 依据其底层的 B+树数据结构能发挥出比全表扫描更加出色的查询能力。

#### 2.1.5 Eureka

Eureka 是 Netflix 开发的服务发现框架, 本身是一个基于 REST 的服务, 主要用于定位运行在 AWS 域中的中间层服务, 以达到负载均衡和中间层服务故障转移的目的<sup>[3]</sup>。简单来说, Eureka 在系统中承接了注册中心的作用。Eureka 在分布式原理的一致性, 可用性, 分区容错性中着重于可用性和分区容错性。Eureka 会向注册的服务器依据心跳机制每隔一定时间间隔 (默认 30s) 发送一个信号, 当注册的服务可能因为服务信号弱无法返回答复信号给 Eureka 时, Eureka 并不会马上从服务列表中剔除该服务。而是在一定时间内 (默认 90s) 统计已注册服务无法返回信号的次数, 达到一定比例后再自行从服务列表中剔除。这样的机制对比同为注册中心的 Zookeeper 虽然一致性有所下降不能马上剔除故障服务, 但保证了服务不会因为网络波动而错误剔除有效服务, 保证了整体分布式架构的高可用。

#### 2.1.6 Ribbon

Spring Cloud Ribbon 是一个基于 HTTP 和 TCP 的客户端负载均衡工具, 它基于 Netflix Ribbon 实现。通过 Spring Cloud 的封装, 可以让我们轻松地将面向服务的 REST 模版请求自动转换成客户端负载均衡的服务调用。Spring Cloud Ribbon 虽然只是一个工具类框架, 它不像服务注册中心、配置中心、API 网关那样需要独立部署, 但是它几乎存在于每一个 Spring Cloud 构建的微服务和基础设施中。因为微服务间的调用, API 网关的请求转发等内容, 实际上都是通过 Ribbon 来实现的, 包括现在企业中常用的 Feign, 它也是基于 Ribbon 实现的工具。所以, 对 Spring Cloud Ribbon 的理解和使用, 对于我们使用 Spring Cloud 来构建微服务非常重要<sup>[4]</sup>。

Ribbon 中拥有多种负载均衡算法，也支持用户根据自己的业务需求来补充所需要的负载均衡算法。

以下介绍 Ribbon 自带的负载均衡算法。

- (a) **RandomRule**: 随机选择一个微服务提供服务。
- (b) **RoundRobinRule**: 轮询服务列表并找到能提供服务的微服务进行调用。
- (c) **RetryRule**: 该算法使用一个重复轮询的机制，不停在服务器列表中找寻没有打开断路器的服务，直至找出能使用的服务。
- (d) **BestAvailableRule**: 轮询服务器列表，如果该服务的断路器打开了，则忽略。该算法将在没打开断路器的服务中选择一个打开的。
- (e) **AvailabilityFilteringRule**: 两次过滤，第一次过滤掉所有断路器打开的服务，也就是那些不可用的服务，第二次过滤掉那些正在处理高并发任务的服务。
- (f) **ResponseTimeWeightedRule**: 根据服务器列表中的服务进行权重分配，响应时间快的服务权重高，反之则权重低。这个算法会较为综合的考虑服务的 io 水平，网络的速度等等因素。
- (g) **ZoneAvoidanceRule**: 两次轮训，第一次先过滤掉那些不存在本区域的服务，第二次过滤掉那些不可用且断路器打开的服务。当然如果本区域没有可用的服务则调用其他区域下的可用服务。

项目中使用了 **ZoneAvoidanceRule** 作为负载均衡策略，先过滤不同地区的微服务，再过滤依据 **Hystrix** 反馈的不可用的微服务，最后调用合适的服务。若本地区中没有能使用的服务再跨地区进行调用。

#### 2.1.7 Hystrix

**Hystrix** 是根据“断路器”模式创建的。“断路器”本身是一个开关装置，当某个服务单元发生故障之后，通过断路器的故障监控，向调用方返回一个符合预期的服务降级处理，而不是长时间地等待或抛出调用方无法处理的异常，

这样就保证了服务调用方的线程不会被长时间不必要地占用，从而避免了故障在分布式系统中蔓延乃至崩溃<sup>[5]</sup>。

**Hystrix** 三大功能服务降级，服务熔断，服务限流。

服务降级：当服务运行时因为 **java** 提示抛出异常，或者单个服务分配线程池信号量达到上限的情况下，就会触发服务降级，给出 **fallback** 方法把错误异常记录在日志中或者返回错误提示。

服务熔断：当该服务在一定时间阈值内达到一定服务降级的次数后就会触发服务熔断，熔断的服务会拒绝请求的访问，在一定时间后尝试恢复服务。

服务限流：依据舱壁模式进行对每个服务进行限流，意思就是为 **Hystrix** 管理的每个服务都分配限定数量的线程。如果请求过多超过可调用线程的上限就开始服务降级，这样的特性使得服务的错误不会蔓延到其他服务当中去。

**Hystrix** 大体执行流程如下总结：

第一二步 包装 **Hystrix** 命令。

第三步 检查缓存有没有短路器的状态（打开关闭）。

第四步 在缓存没命中时检查断路器是否打开了。

第五步 检查 **Hystrix** 分配给这个服务的线程数和信号量是否满了，满了的话向第七部反馈降级，再走到第八步降级执行。

第六步 执行方法主体，出错或者超时都向第七步反馈降级，并且走到第八步降级执行。

第七步 统计一定时间内的降级次数，达到一定次数后，把消息发送到第三步执行熔断。

第八步 **fallback** 方法的执行。

### 2.1.8 Redis

**Redis** 是一个成熟的非关系型数据库。由于其内核模型为基于事件的 **nio** 多路复用模型，因此读写的速度十分快，1 秒可以达到 10 万左右的小数据读写，因此性能十分强悍。**Redis** 对请求的单线程处理以及一些原子性命令如 **setnx**

和对 lua 脚本的支持使得它可以在高并发的场景下作为分布式锁的使用。Redis 同时拥有出色的事务机制，确保了高并发情况下分布式锁解锁操作的原子性和提升了解锁操作安全程度。在可扩展性方面，Redis 支持 cluster 模式和哨兵机制，为日后系统的规模扩大打下技术上坚实基础。在持久化方面，Redis 采用了 RDB 和 AOF 的方法来保持 Redis 数据的持久化，能在 Redis 宕机后通过相关文件恢复 Redis 的存储内容，提高缓存的安全性<sup>[6]</sup>。

### 2.1.9 Canal

Canal 是阿里巴巴开发的一个开源组组件，意在帮助开发者监 MySQL 听数据库的 binlog 日志，当 binlog 日志出现更新后，Canal 能通过异步消息队列把消息发送到微服务中，从而保证缓存与数据库读写的一致性。

### 2.1.10 RabbitMQ

RabbitMQ 是实现了高级消息队列协议 (AMQP) 的开源消息代理软件 (亦称面向消息的中间件)。RabbitMQ 支持三种模式，分别是 direct, fanout 和 topic。direct 是指一个客户端绑定一条队列，fanout 是指当交换机接收到信息后，分发到多条绑定的队列中。topic 在交换机收到信息后，根据 Binding key 匹配到适合的队列中去。

## 3 需求分析

### 3.1 系统目标

目前在国内外存在的问卷调查平台尽管都能进行问卷发布，问卷作答的功能，但是欠缺了针对高校这一特定场景对问卷结果反馈的分析。本系统的开发基于 B/S 结构，结合 WEB 开放，自主的特点建立起一个高校毕业生和在校生就业数据分析的相关平台。希望以此平台方便调查问卷的创建与发布，并将培养计划中值得关注的课程信息反馈给老师。前端保持网站 UI 风格的统一大气。

### 3.2 系统可行性分析

#### 3.2.1 技术可行性

基于 B/S 结果的 web 平台使用了 Springboot+Mybatis 这样成熟的组合架构。由于对后续可能扩展网站规模的可能性进行考虑，因此使用了分布式架构 Eureka 作为项目的注册中心，Ribbon 为每个微服务实例提供负载均衡策略，Hystrix 为每个微服务的操作保驾护航。除此之外使用了 Redis 这样的非关系型数据库作为项目的缓存。数据库使用了 MYSQL 配合 Springboot 的内置 Tomcat 组成 B/S 体系结果。

#### 3.2.2 操作可行性

第五代移动通信技术（5th Generation Mobile Communication Technology，简称 5G）是具有高速率、低时延和大连接特点的新一代宽带移动通信技术，是实现人机物互联的网络基础设施。在基于 5G 的互联网技术上搭建的本平台能被大多数人访问<sup>[7]</sup>。

#### 3.2.3 经济可行性

开发本网站只需要租借服务器的费用，能帮助师生解决就业调查的问题为减少每年就业调查外包所带来的费用。

### 3.3 系统的功能性需求

系统角色主要有学生和教师，学生用例图如图 3-1 所示。用例主要有登陆，注册，找回密码，问卷搜索，问卷作答等。教师用例图如图 3-2 所示。用例主要有创建问卷，填写课程路径，填写培养计划课程，填写问卷内容，统计作答人数，分析答卷数据生成重点课程路径。

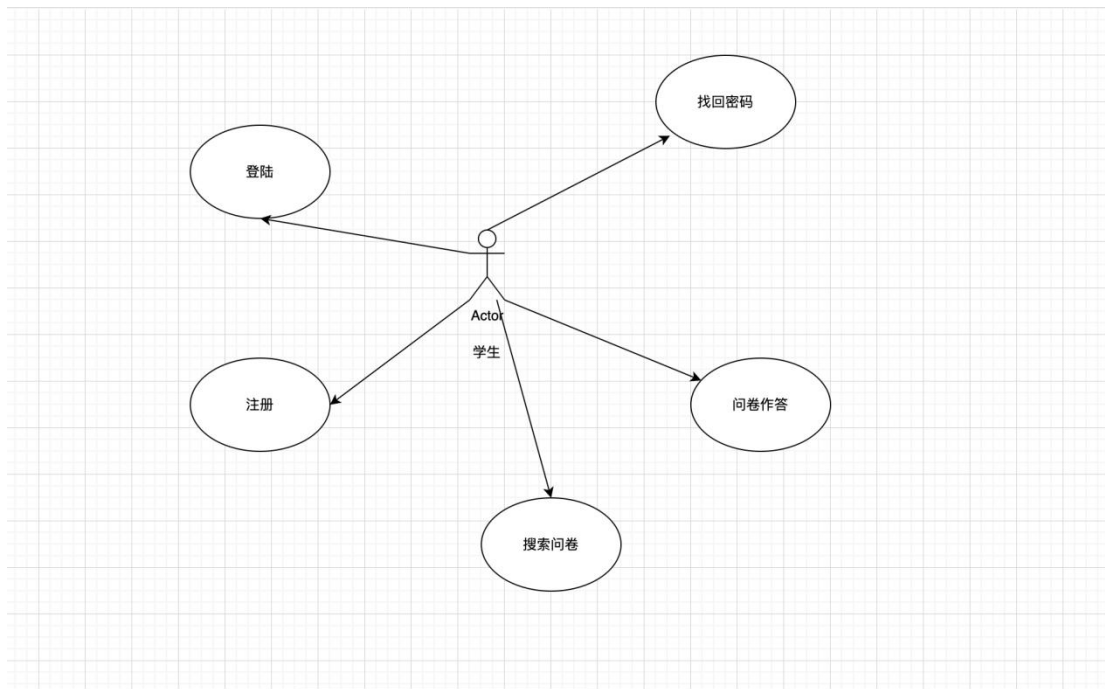


图 3-1 学生用例图



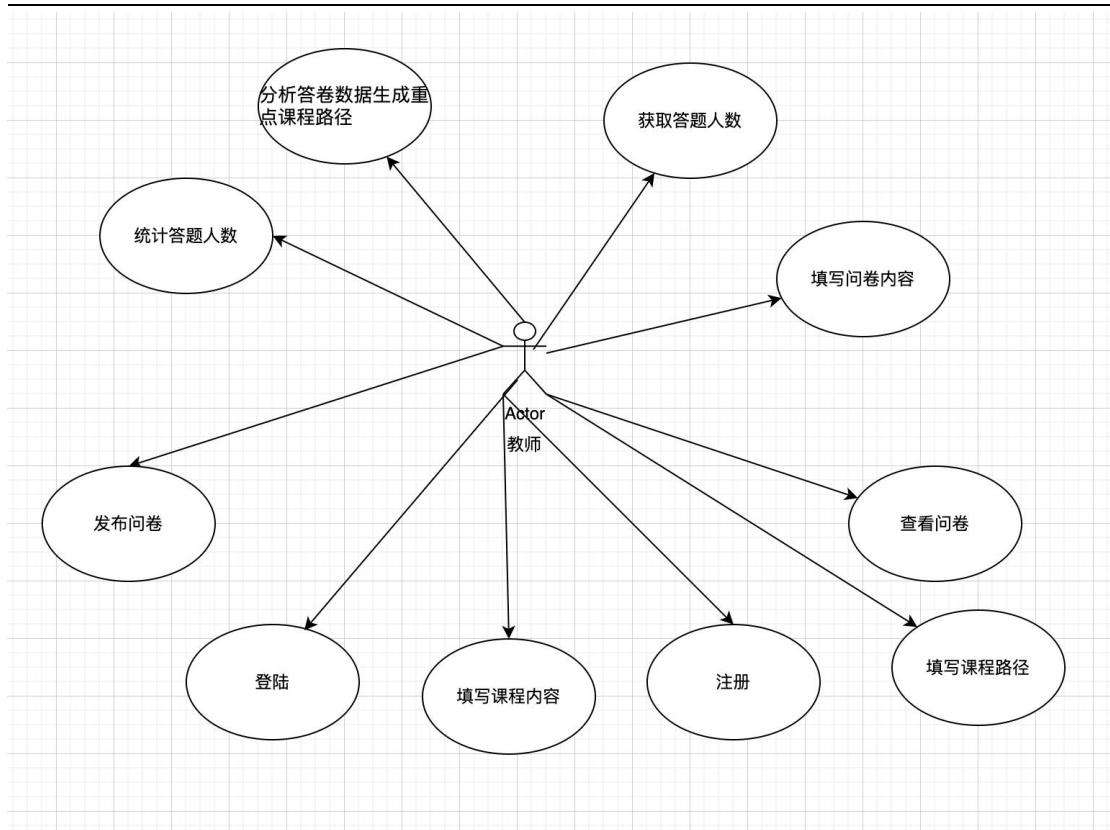


图 3-2 教师用例图

### 3.3.1 登录用例规约

用例名称：用户登陆

参与者：学生，教师

前提条件：用户已经注册

前置操作：用户注册后进入登陆页面

后置操作：进入主页面

合法流程：用户填写好账号，密码后，上传到后端，后端匹配数据库信息  
后 返回前端结果

### 3.3.2 注册用例规约

用例名称：用户注册

参与者：学生，教师

前提条件：无

前置操作：用户输入邮箱，密码和确认密码

后置操作：点击注册，等待服务器更新数据库并返回结果

合法流程：用户输入邮箱，密码和确认密码，前端和后端用正则表达式匹配邮箱和密码格式，最后返回操作结果给用户

### 3.3.3 新建问卷

用例名称：用户新建问卷

参与者：教师

前提条件：数据库中已有用户的用户 id

前置操作：用户已经注册和登录

后置操作：填写培养计划的课程

合法流程：从主页进入新建问卷页面，点击新建问卷按钮，数据库根据雪花算法分配一个唯一的问卷 id

### 3.3.4 填写培养计划课程

用例名称：填写培养计划课程

参与者：教师

前提条件：数据库中已有用户的用户 id，已有该问卷的 id

前置操作：创建问卷

后置操作：填写培养计划路径

合法流程：填写课程的名称，权重，根据问卷的 id 提交

### 3.3.5 填写或更新课程路径

用例名称：填写或更新课程路径

参与者：教师

前提条件：数据库中已有用户的用户 id，已有该问卷的 id，并且填写了培养计划的课程及权重

前置操作：填写培养计划课程

后置操作：填写问卷内容

合法流程：在网页上拖拽箭头和圆圈等可视化图形填写培养计划路径

### 3.3.6 填写或更新问卷内容

用例名称: 填写或更新问卷内容

参与者: 教师

前提条件: 数据库中已有用户的用户 id, 已有该问卷的 id, 并且填写了培养计划的课程及权重, 填写了培养计划路径

前置操作: 填写或更新课程路径

后置操作: 无

合法流程: 用户填写问卷名称, 填写相关简介, 编辑问题类型以及选项的相关课程

### 3.3.7 查询问卷

用例名称: 查询问卷

参与者: 学生

前提条件: 用户已经登陆

前置操作: 登陆

后置操作: 作答

合法流程: 用户通过问卷的 id 或者问卷名称搜寻问卷, 在搜寻结果列表中选择问卷作答

### 3.3.8 问卷作答

用例名称: 问卷作答

参与者: 学生

前提条件: 用户已经登陆, 问卷已发布

前置操作: 查询问卷

后置操作: 无

合法流程: 对选择的问卷作答, 选择用户认为正确的答案并且选择相关的课程

### 3.3.9 问卷分析

用例名称: 问卷分析

参与者: 教师

前提条件: 用户已经登陆, 问卷已发布, 且问卷是该用户创建的

前置操作: 创建问卷

后置操作: 无

合法流程: 点击分析按钮进行分析, 统计每个题目每个选项的答案数量并  
根据迪杰斯特拉算法分析出最适合的培养计划路径

### 3.3.10 发布问卷

用例名称: 发布问卷

参与者: 用户

前提条件: 用户已经登陆, 问卷已发布, 且问卷是该用户创建的

前置操作: 创建问卷

后置操作: 无

合法流程: 选择我的问卷列表, 点击发布按钮

## 4 系统设计

### 4.1 功能设计

#### 4.1.1 系统结构

使用 Eureka 作为注册中心，Ribbon 组件作为负载均衡组件，Hystrix 组件保证和反馈服务的健康程度，Gateway 作为网关层过滤不恰当的请求，以及支持跨域访问。Redis 作为缓存组件减轻数据库读写压力，RabbitMQ 传递异步信息，Canal 负责监听数据库日志及时将数据库更新的状况发送到缓存中。结构图如 4-1 所示：

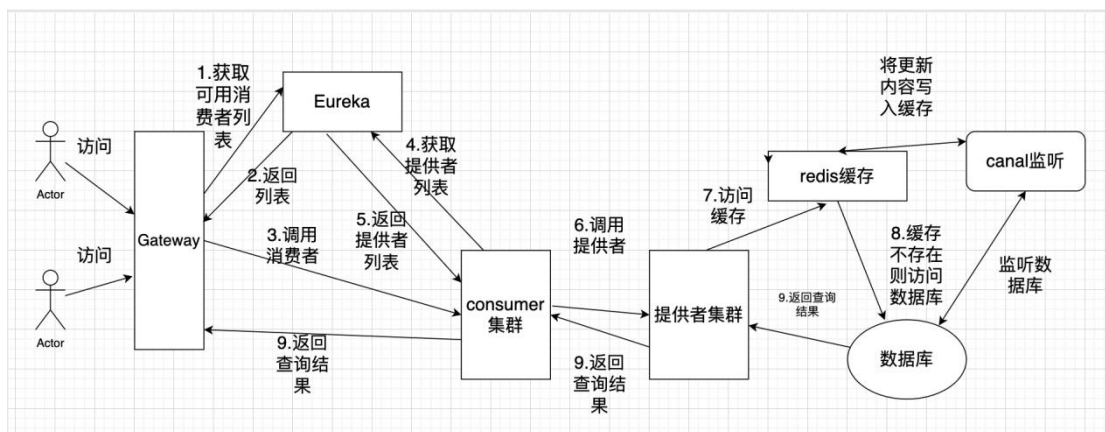


图 4-1 系统结构图

#### 4.1.2 系统功能模块

本系统的功能模块图如图 4-2 所示。

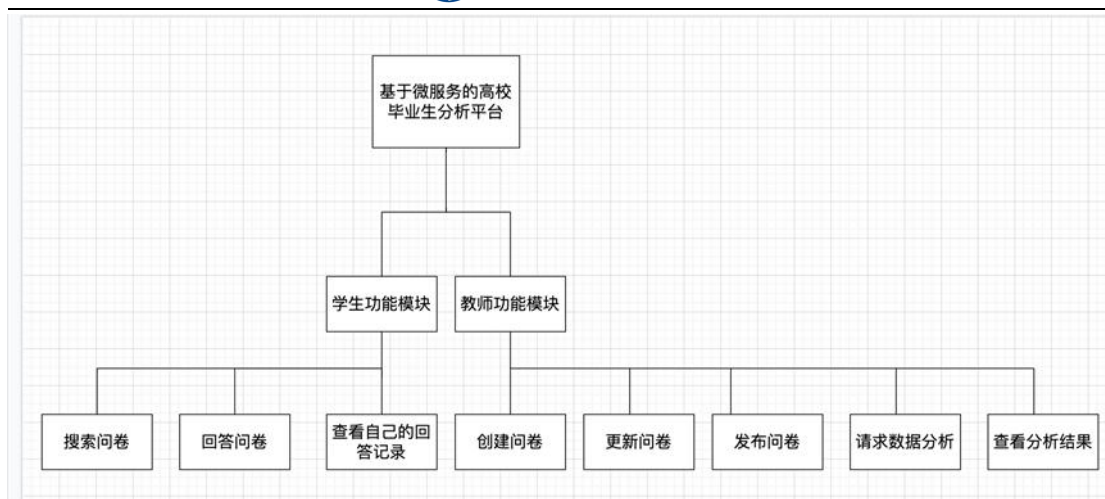


图 4-2 模块功能图

## 4.2 数据库表设计

### 4.2.1 系统 E-R 图

根据他们的关系生成的 E-R 图如图 4-3 所示。

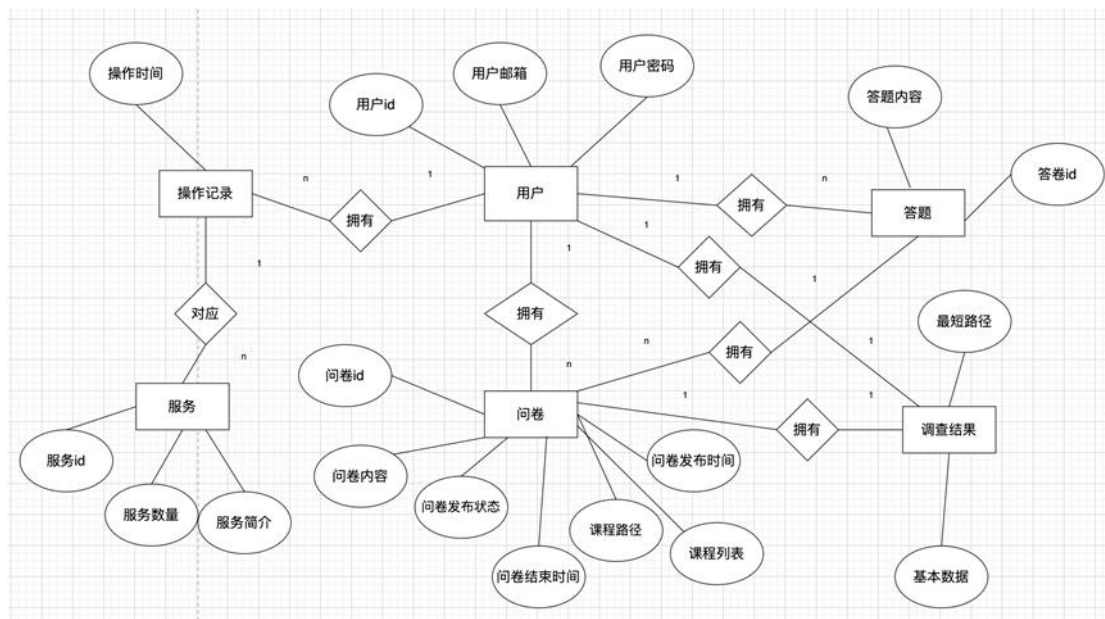


图 4-3 E-R 图

### 4.2.2 数据库表的设计

在确定了实体类后，需要把将实体类更具体化的体现在数据库的表中，一个问卷有一个创建者，多张问卷可以有同一个创建者，一个用户可以回答多个问卷。一张问卷可以被多个用户回答。因此依据这样的关系创立了下列几张表：

### (1) 用户表

用于描述用户实体，结构如表 4-1 所示。

表 4-1 USER 表结构

名称	说明	数据类型	长度	主键	外来键
user_id	用户主键	BIGINT	19 位	true	false
user_name	用户注册邮箱	VARCHAR	45 位	false	false
user_password	用户登陆密码	VARCHAR	45 位	false	false
user_expired	验证码有效时间	DATE	无	false	false
user_test	用户验证码	VARCHAR	45 位	false	false

### (2) 问卷表

用于描述问卷的信息，结构如表 4-2 所示：

表 4-2 QUESTION 表结构

名称	说明	数据类型	长度	主键	外来键
q_id	问卷编号	BIGINT	19 位	true	false
user_id	问卷创建者的编号	BIGINT	19 位	true	true
q_content	问卷内容	JSON	无	false	false
courses	问卷相关课程列表	JSON	无	false	false
c_relations	问卷相关培养计划的有向图	JSON	无	false	false
q_name	问卷名称	VARCAHR	100	false	false
q_script	问卷简介	VARCAHR	300	false	false
q_start_time	问卷创建时间	DATE	无	false	false
q_end_time	问卷过期时间	DATE	无	false	false
route_start	分析课程路径的起点	INT	11	false	false
route_ends	分析课程路径的终点	JSON	无	false	false

### (3) 答卷表

用于描述用户作答情况的表，结构如表 4-3 所示。

表 4-3 ANSWERS 表结构

名称	说明	数据类型	长度	主键	外来键
user_id	答题用户的编码	BIGINT	19 位	true	true
q_id	答案所属问卷的编码	BIGINT	19 位	true	true
ans_content	用户答案	JSON	无	false	false

### (4) 调查结果表

用于描述调查结果的表，结构如表 4-4 所示。

表 4-4 BASIC\_RESULT 表结构

名称	说明	数据类型	长度	主键	外来键
user_id	问卷所有者的主键	BIGINT	19 位	true	true
q_id	该结果所属问卷的主键	BIGINT	19 位	true	true
res_basic	每条题目的选项数量统计	JSON	无	false	false
basic_relations	培养计划重点课程路径	JSON	无	false	false

### (5) 加急服务表

用于描述用户申请加急服务的表，结构如表 4-5 所示。

表 4-5 SERVICE 表结构

名称	说明	数据类型	长度	主键	外来键
service_id	服务号，主键	BIGINT	19 位	true	false



nums	剩余次数	INT	11 位	false	false
res_basic	每条题目的选项数量统计	JSON	无	false	false

## (6) 用户操作记录

用于描述用户操作记录的表，结构如表 4-6 所示。

表 4-6 HISTORY 表结构

名称	说明	数据类型	长度	主键	外来键
his_id	操作记录号	BIGINT	19 位	true	false
user_id	进行该操作用户的 id	BIGINT	19 位	false	false
type	操作对应的服务号	BIGINT	19 位	false	true
time	操作时间	DATE	无	false	false

# 5 系统实现

本系统可基于高校在校生或毕业生的作答情况，帮助发布问卷的教师分析出当前培养计划中哪些课程对学生的学习与就业有宝贵意义，并把包含这些课程培养计划路径反馈给教师以便教师在下一年的教学中做出适当调整。

## 5.1 登录功能

进入登录页面用户的名称，密码为必填选项，在页面上是用了 Vue 技术，保证整个前端拥有统一的 UI 风格，如下图 5-1 所示。

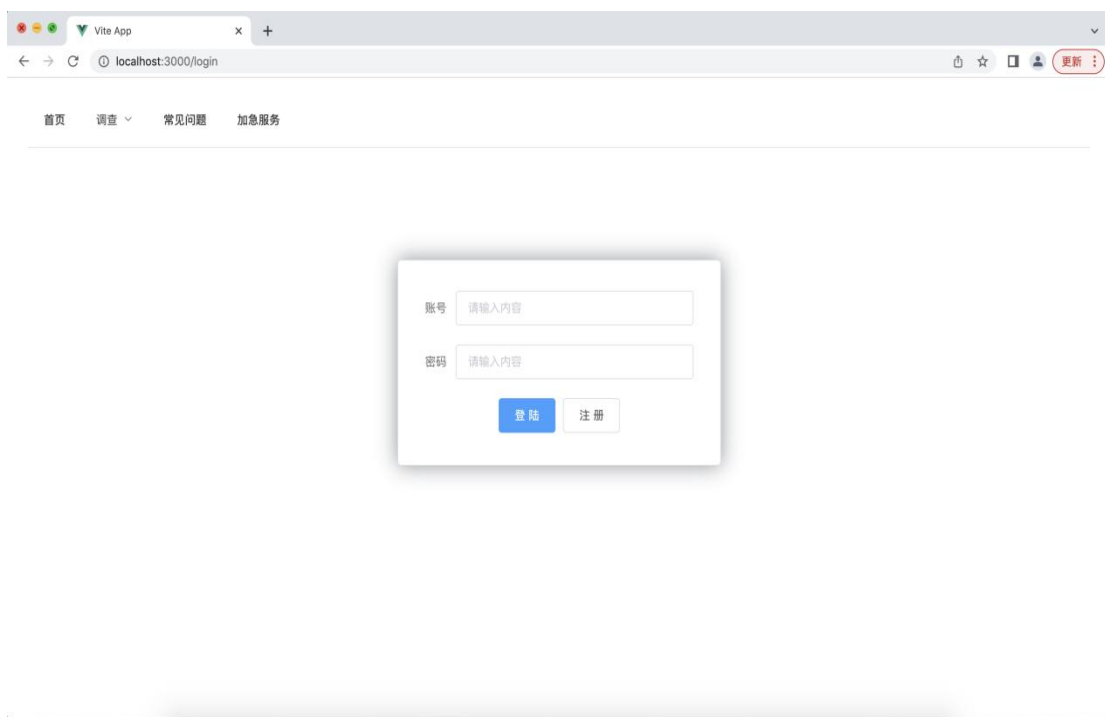


图 5-1 登陆界面

## 5.2 用户注册

注册页面，用户填写自己的邮箱，密码以及确认密码，利用 Vue 的正则表达式匹配用户邮箱格式以及用户密码和确认密码是否相同。符合所有要求后上传至数据库，返回注册成功的提示。如图 5-2 所示。用户 id 由数据库进行分配，使用雪花算法进行分配用户 id。雪花算法是一种适合分布式系统使用的基于全局生成唯一 id 的算法，基本原理为生成 64 位的二进制长整型数字，其中最高位是 0，不可用。41 位是作为时间序列精确到毫秒级，长度可以使用 69 年。10 位机器编码号，可以支持 1024 个微服务结点。12 位计数序列号，这是一些列自增 id，一秒内支持生成 4096 个 id。相较于传统系统生成的 uuid，雪花算法更能发挥出数据库索引的功能，原因是雪花算法由时间和有序 id 号生成，这样的特性契合使用索引搜索的高效性。如图 5-3 所示。由于索引的使用是在 B+ 树上进行，依据有序 id 的搜索可以避免数据库的全表扫描，搜寻复杂度为  $\log(n)$  比全表扫描的  $O(n)$  复杂度更具有优势<sup>[8]</sup>。

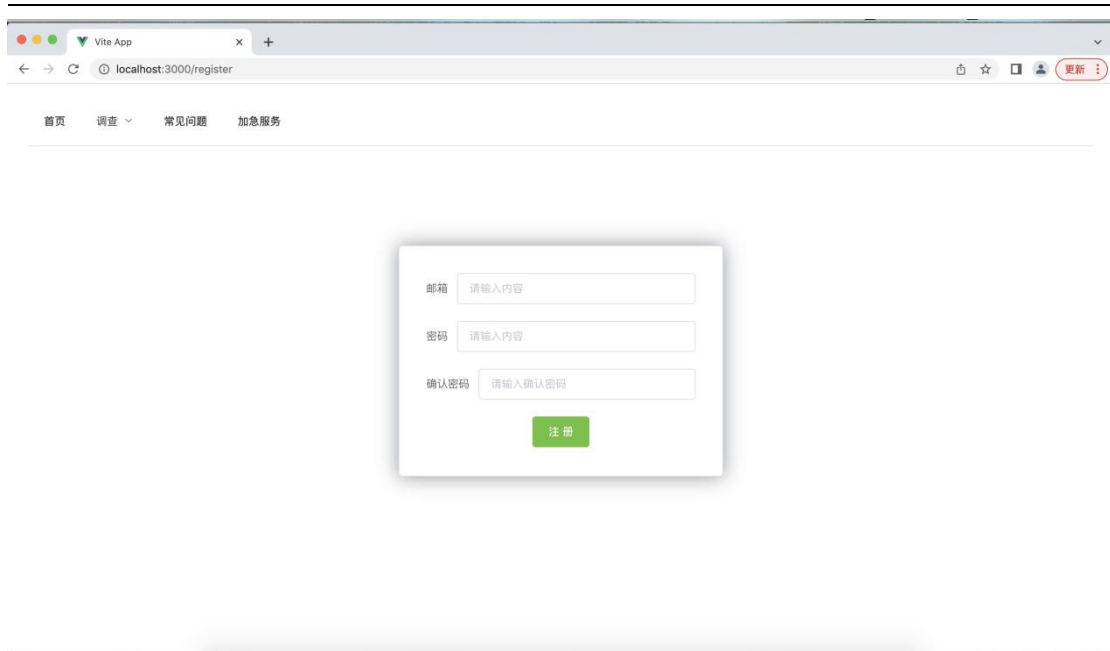


图 5-2 注册界面

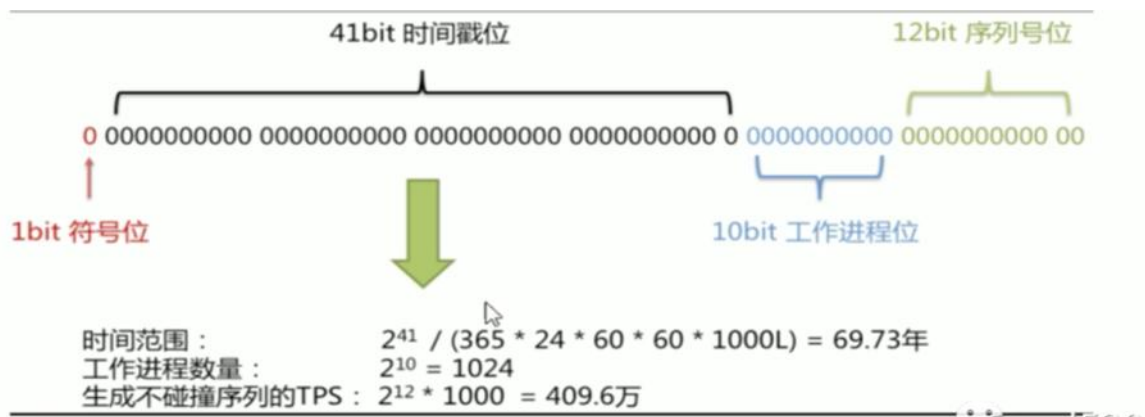


图 5-3 雪花算法原理

### 5.3 主页面

此页面为登陆后所进入的页面，拥有转跳各个页面的标签。如图 5-4 所示。



图 5-4 主页

## 5.4 创建课程页面

点击创建按钮后就会依据雪花算法生成一个问卷的 id 录入数据库，页面上提示了接下来需要进行操作的步骤。如图 5-5 所示。

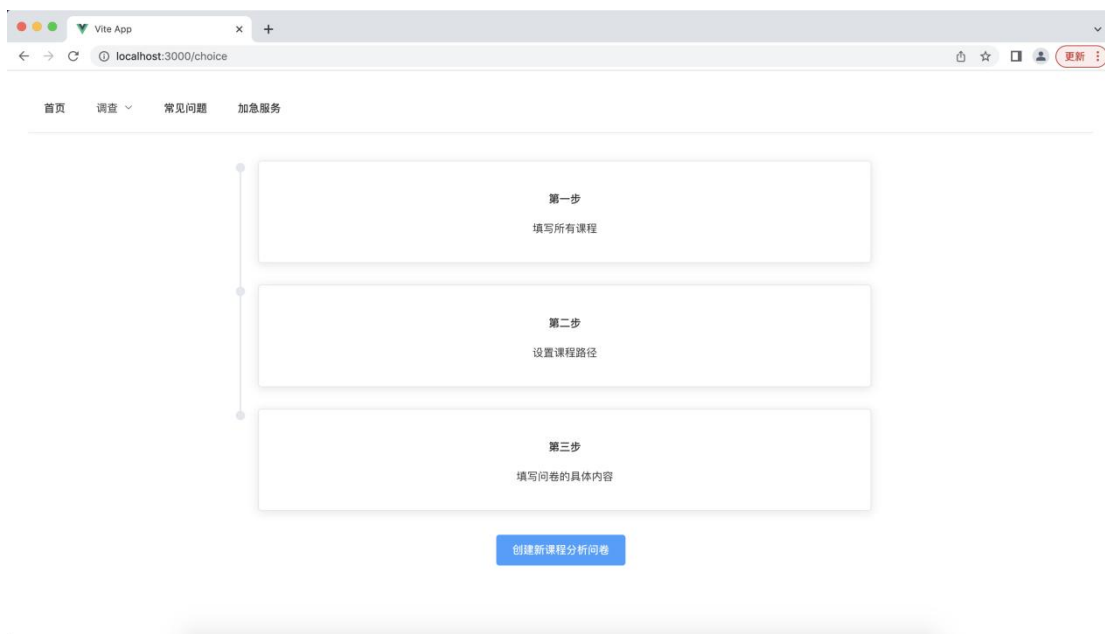


图 5-5 创建课程页面

## 5.5 填写问卷相关课程

在此页面用户可以填写问卷相关的培养计划的课程以及课程路径, 用户可以自由添加, 删除列表上的课程。用户点击提交按钮后课程列表以及课程内容, 以 JSON 的格式录入数据库, 其中后端会使用 DFA 自动机算法对用户提交的内容进行过滤。DFA 全称为: **Deterministic Finite Automaton**, 即确定有穷自动机。其特征为: 有一个有限状态集合和一些从一个状态通向另一个状态的边, 每条边上标记有一个符号, 其中一个状态是初态, 某些状态是终态<sup>[9]</sup>。但不同于不确定的有限自动机, DFA 中不会有从同一状态出发的两条边标志有相同的符号。首先列出关于暴力, 色情等内容的敏感词列表, 然后对其构造 DFA 自动机, 例如敏感词“贝吉塔”, 我们根据“贝”字在 HashMap 中生成一个结点, “吉”字作为一个结点的 key, value 则是一个新的 HashMap, HashMap 中的一个结点是“主”, 一次类推, 当我们扫描一次文本时, 文本在识别到“塔”字后就会在 DFA 中不停匹配, 如果在下两个字中 DFA 的扫描到达了自动机的终态, 则把非法词记录并反馈给用户。而对比传统的 KMP 字符串匹配算法去进行识别, DFA 扫描只需要对所有关键词在系统初始化时扫描一次, 对上传文字扫描一次即可。界面如下图 5-6, 图 5-7 所示。

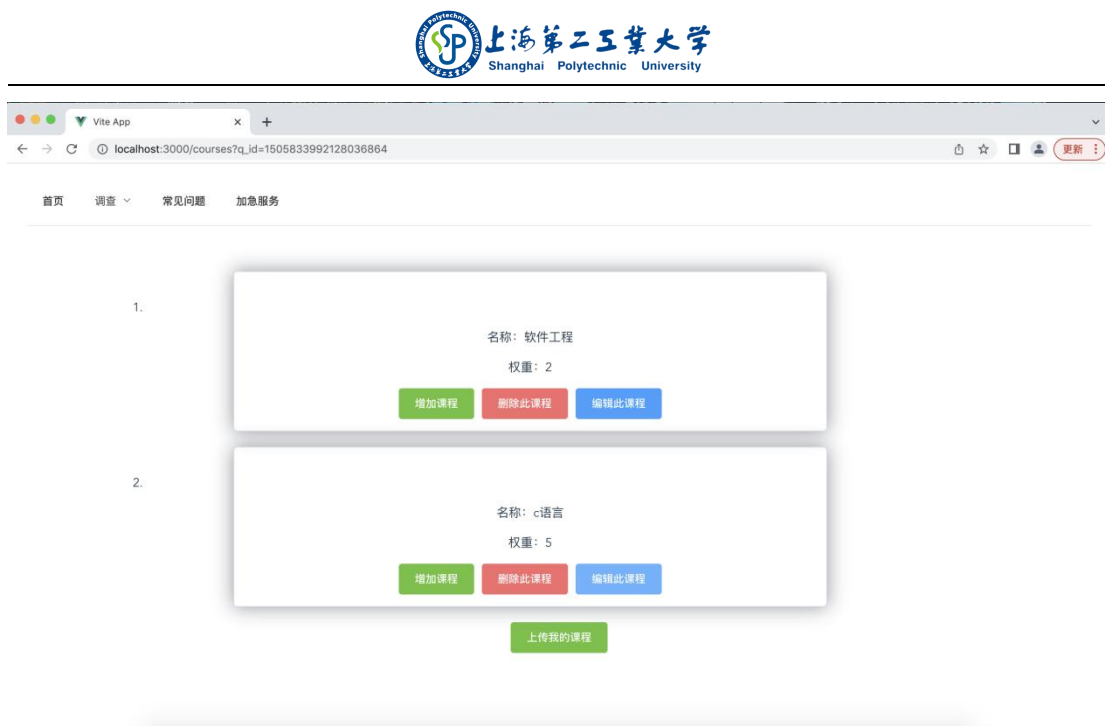


图 5-6 填写课程页面

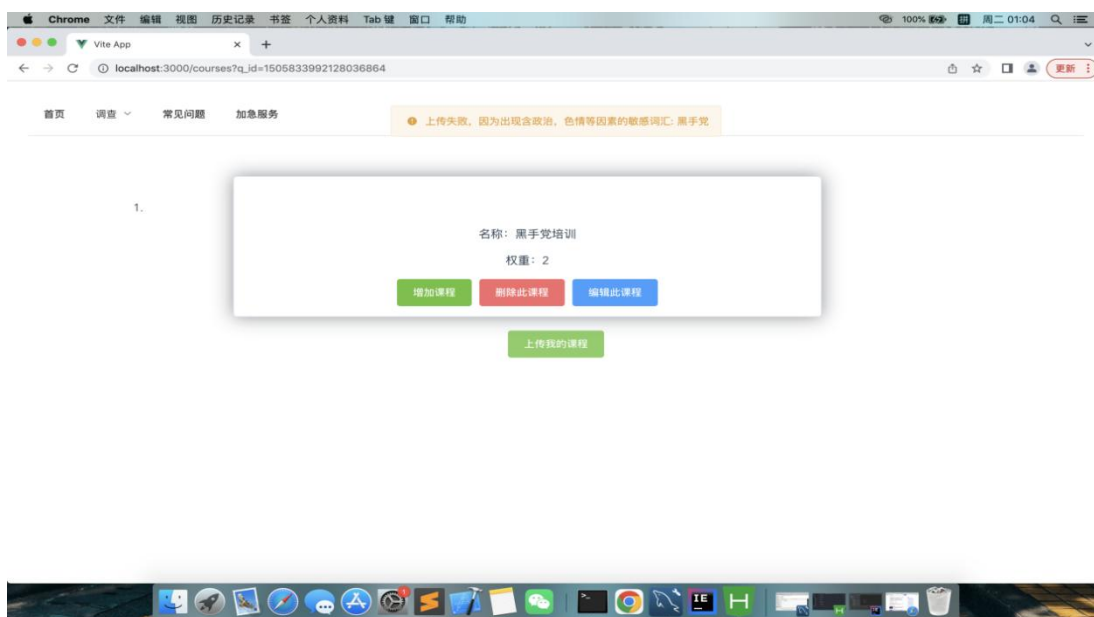


图 5-7 敏感词过滤提醒

## 5.6 更新或填写路径

在此页面你可以输入新建结点，根据上一步培养计划所录入的课程，使用拖拽，旋转，缩放等方式构建你的培养计划路径，点击上传课程关系既可以JSON 格式以二维数组的方式录入数据库，如图 5-8，5-9 所示。



图 5-8 填写路径页面

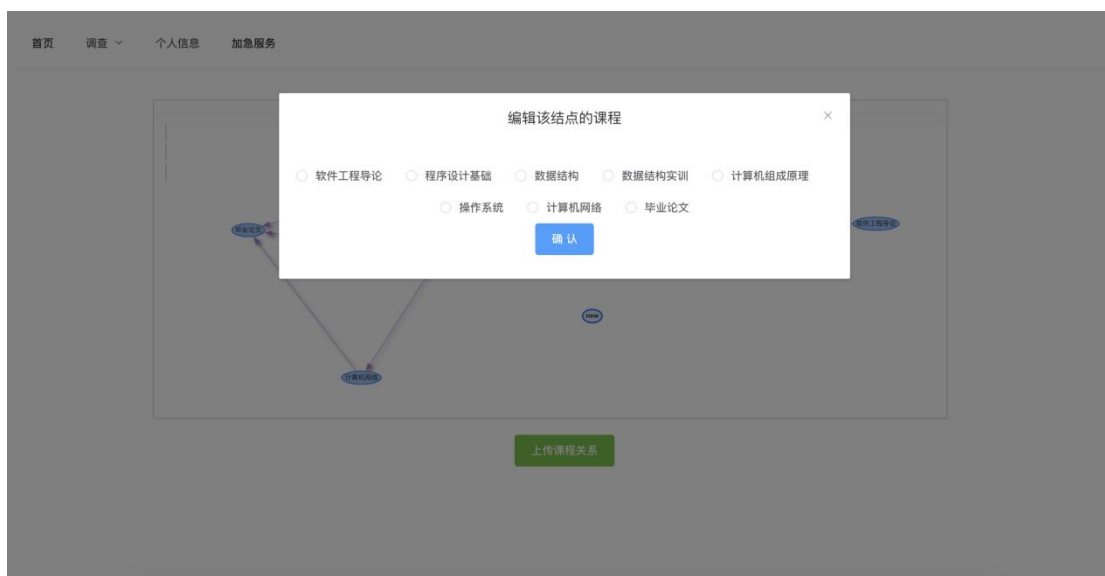


图 5-9 结点赋值选项

## 5.7 更新或填写问卷内容

用户可有在此页面你可以更新或填写你的问卷名称，问卷问题类别，选项，选项相关课程。填写后以 JSON 格式把问卷内容上传至数据库。技术上由 Vue 的动态渲染支持。如下图 5-10，5-11 所示。

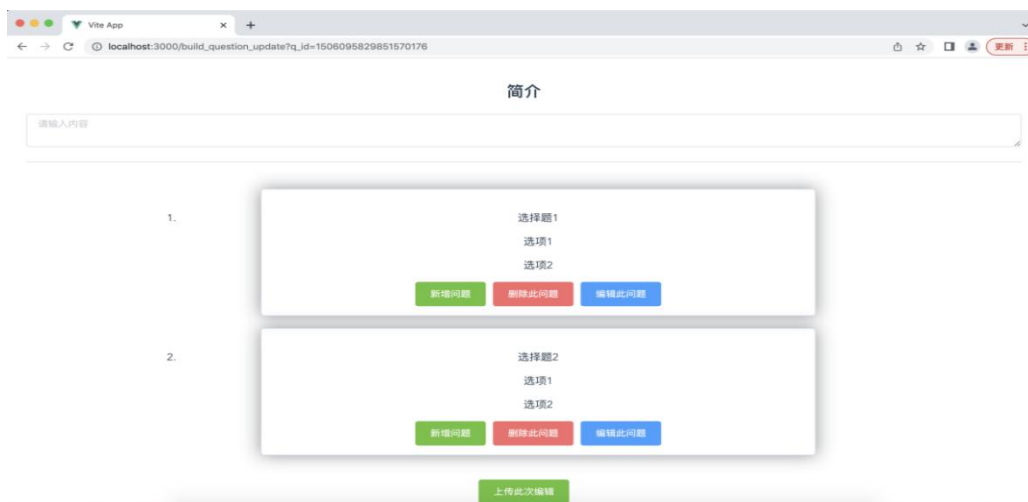


图 5-10 填写答卷页面

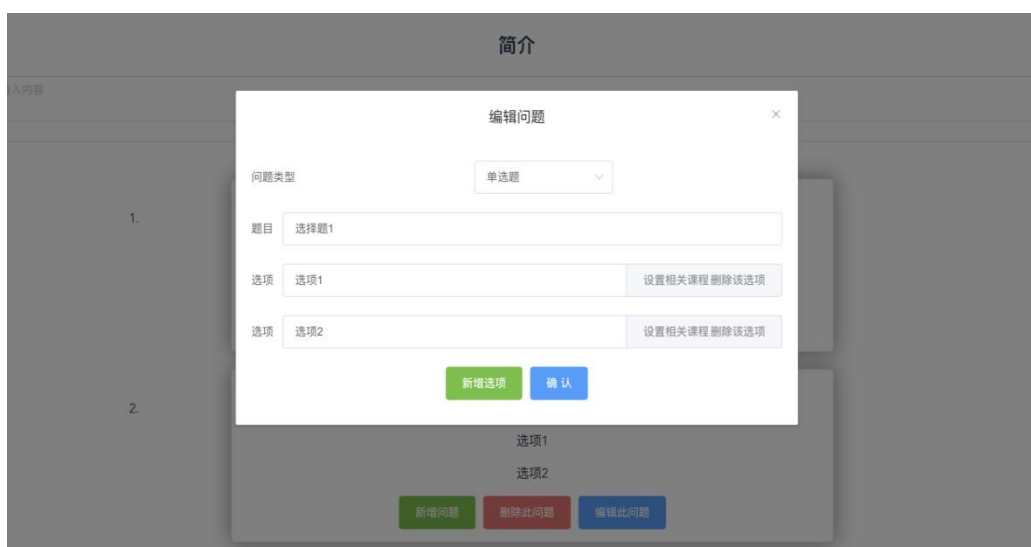


图 5-11 编辑单个问题界面

## 5.8 调查问卷

用户在此页面可以通过问卷 id 或者问卷名称查询问卷，并在查询后可选在问卷进行作答进入到问卷作答的页面。为方便用户使用，使用 Redis 记录该用户近五次的查询结果。效果如图 5-12。



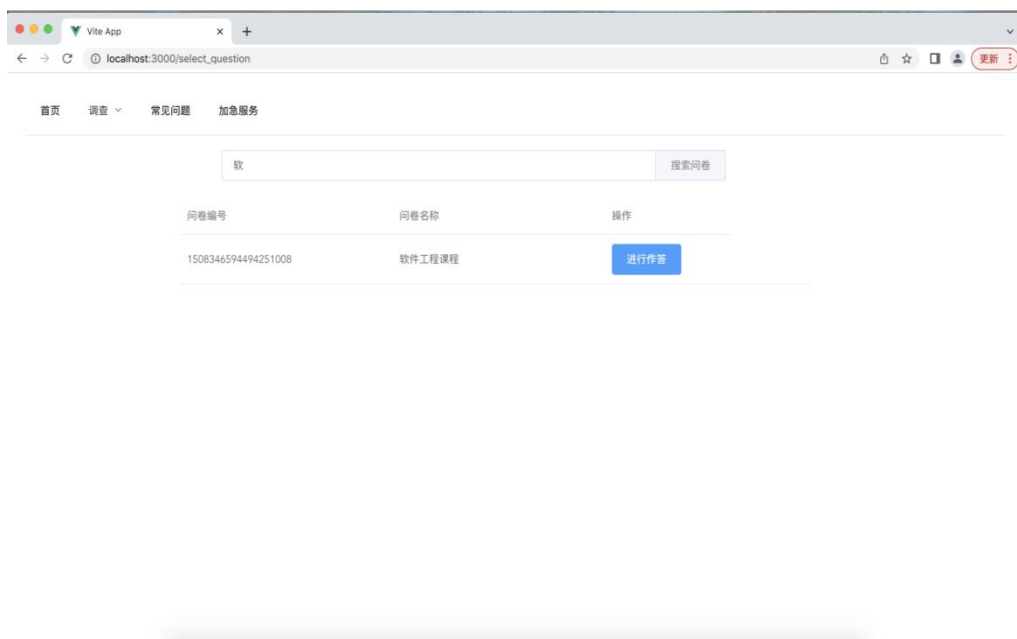


图 5-12 查找问卷界面

## 5.9 查看该用户上传的问卷列表

在这个页面中用户可以看到自己发起的调查问卷，在此页面中可以选择任一问卷进行更新，发布到等操作，正式发布问卷后就可以被其余用户搜索到进行作答，但正式发布后部分内容不能更改。界面如图 5-13 所示。

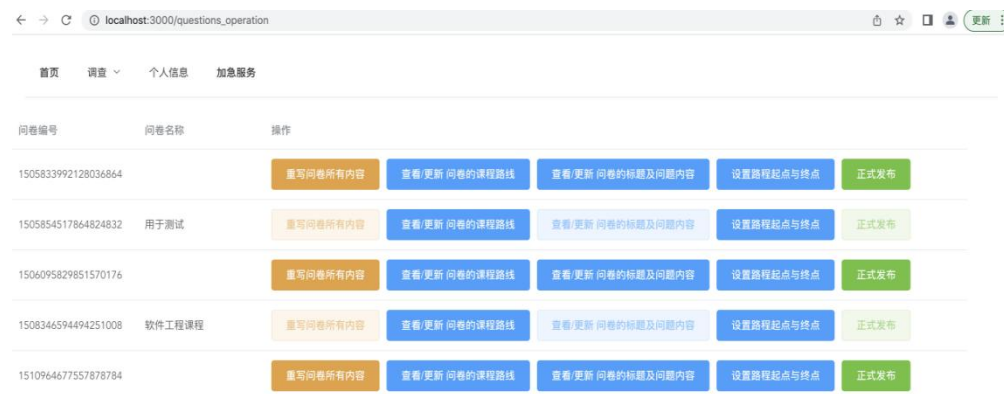


图 5-13 查看我的问卷界面

## 5.10 申请及时分析功能

### (1) 功能概览

其下有三个选项供用户选择，每个选项每日名额有限

全面分析：统计每个题目每个选项的答题数据，并且提供重点课程路径给教师

分析答题人数：统计每个题目每个选项的答题数据

分析路径：提供重点课程路径给教师

## (2) 路径分析算法及原理

其中分析课程路径用到的算法为迪杰斯特拉最短路径算法，在前面的多次操作中，根据用户的作答填写的相关课程作为权重，教师给培养计划的课程分配的学分作为权重，把这些权重，以减法的形式减到每个课程结点中，最后根据该算法求出最短路径。

迪杰斯特拉算法每次寻找到离当前集合  $v$  最近的结点  $n$ ，尝试用此结点优化当前集合到其他点的距离（注意这些点必须和当前最近结点  $n$  相连），然后把最近的结点加入集合  $v$  中，循环往复。

## (3) 分布式锁原理

此处由于涉及高并发的抢购案例，因此使用了高并发的分布式锁技术。在高并发的场景中，业务有可能由于程序员的不恰当操作导致脏读现象的发生。在高并发环境下查询与更新的两条数据库操作的非原子性会导致用户读取了过期数据从而导致业务读写错误。但若在高并发场景下使用 MySQL 数据库自带的事物功能或者 InnoDB 引擎的行级锁，同样会造成数据库读写压力的倍增。

采用 Redis 这样的非关系型数据库作为分布式锁的相关组件更多是基于 Redis 线程模型与事务的特点。灵活运用了 Redis 线程模型中基于事件的单线程的 io 多路复用模型来进行操作。具体过程为用户在抢锁的过程中以 `service_id` 为 key 在 Redis 中以 `string` 的数据结构设立锁，以用户当前线程的哈希码作为 `value`。上锁设置 4s 中的过期时间。由于该指令使用的是 `setnx` 指

令，因此把查询锁和创建锁的步骤作为一个单线程且原子性的操作在 Redis 中进行<sup>[10]</sup>。在用户完成更新数据库后解锁，解锁的判断是当前用户的哈希码是否和锁中的 value 值的哈希码一值，避免其他线程操作不当对当前锁进行解锁。解锁过程中同时也使用了 Redis 的事务的 watch 命令来监视锁。一旦锁的值被更改了，解锁事务将不发送并且向日志写入错误。运用 Redis 事务提高系统的安全性和健壮性。

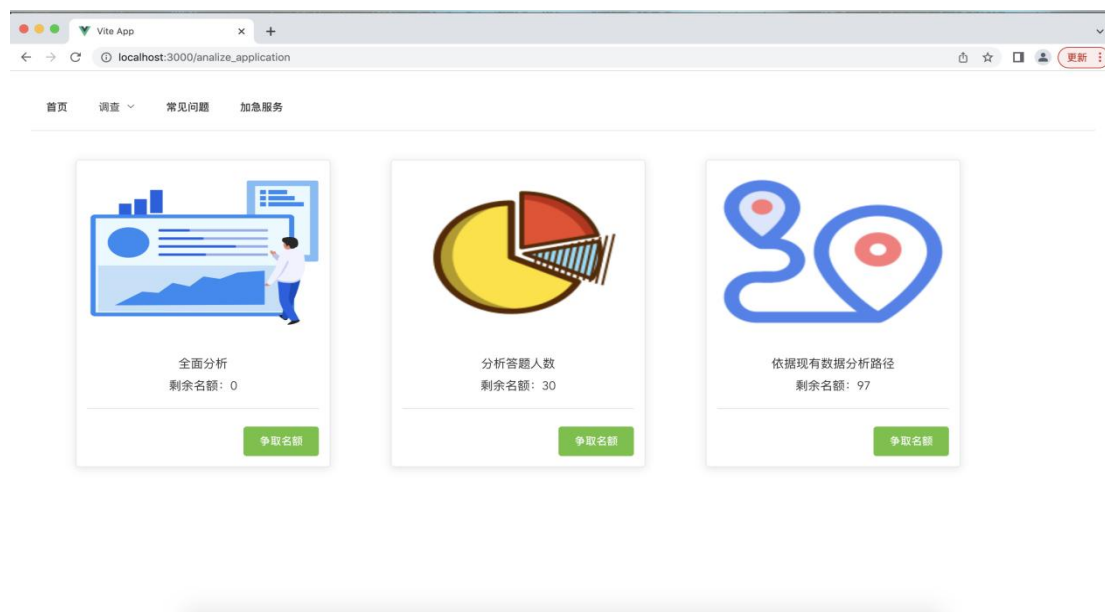


图 5-14 加急服务申请界面

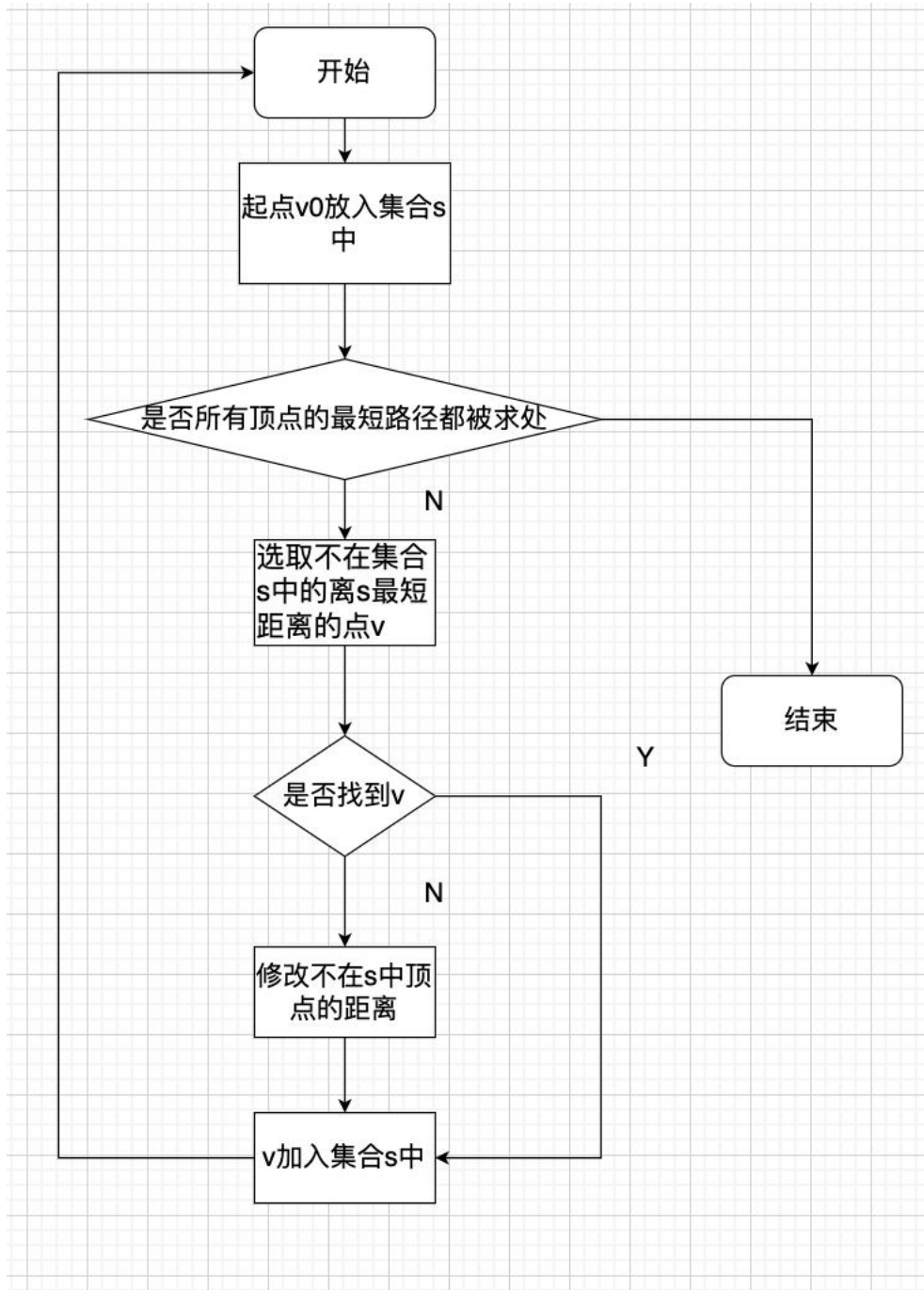


图 5-15 迪杰斯特拉算法流程图

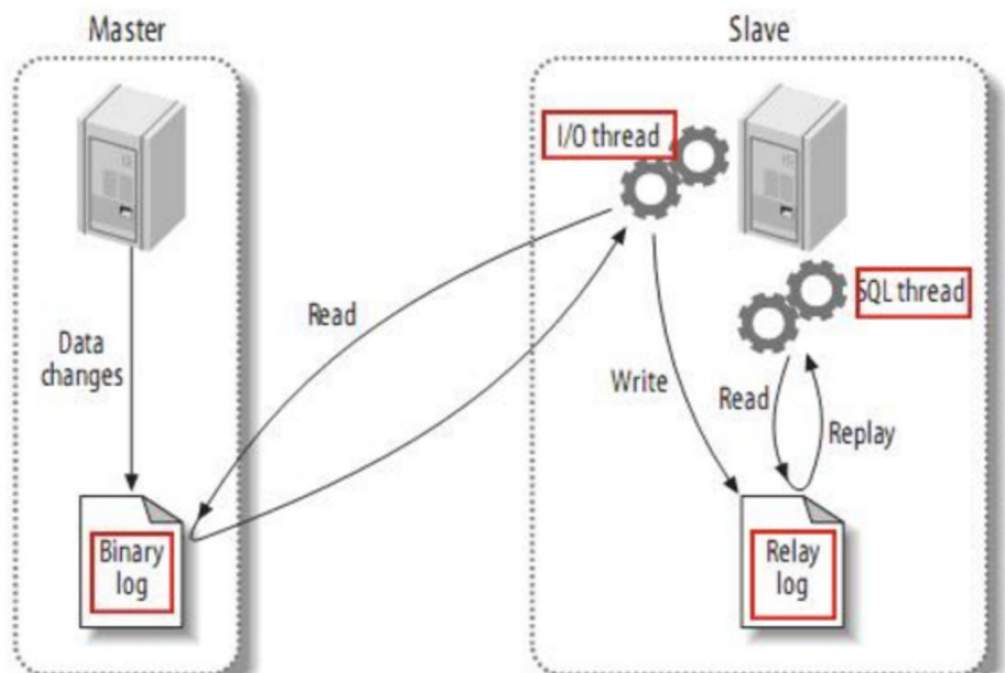


图 5-16 canal 组件原理图

## 5.11 回答问卷

用户在此页面可以回答问卷，并且选择与答案相关的课程，点击上传按钮即可以 JSON 格式上传到数据库当中。界面如下图 5-17 和 5-18 所示。

[首页](#)   [调查](#)   [个人信息](#)   [加急服务](#)

1. 选择题1

☒ 选项1
 ☐ 选项2

选择与答案相关的课程

2. 选择题2

☒ 选项1
 ☐ 选项2

选择与答案相关的课程

上传此次编辑

图 5-17 回答问题界面

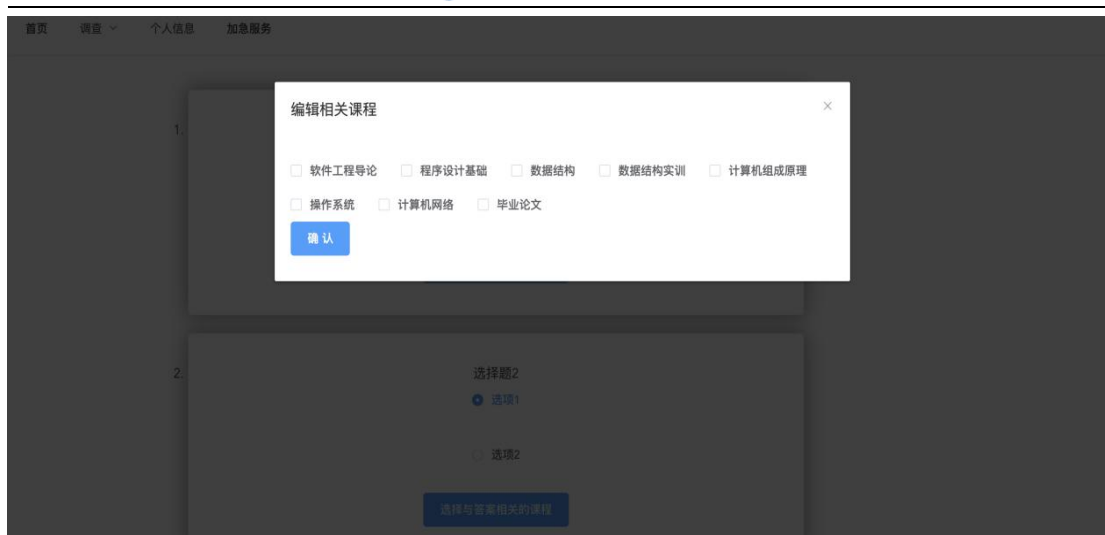


图 5-18 编辑回答相关课程界面

## 5.12 统计答题人数

用户可使用三种视图来观看每个题目每个选项的答题人数，如图 5-19 ~ 图 5-21 所示。

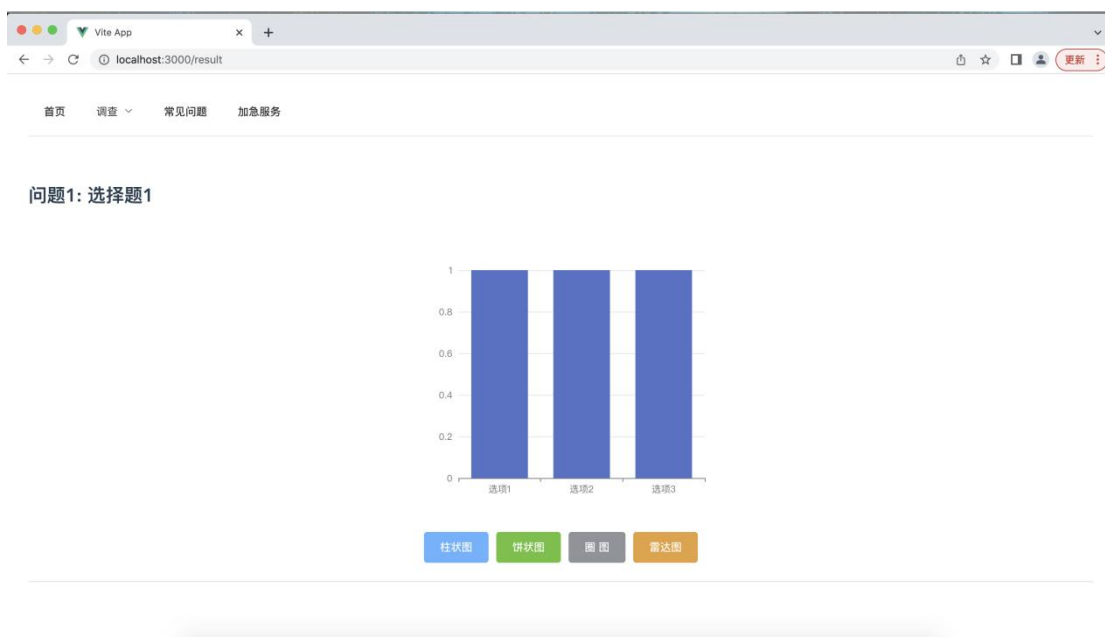


图 5-19 柱状图显示答题人数

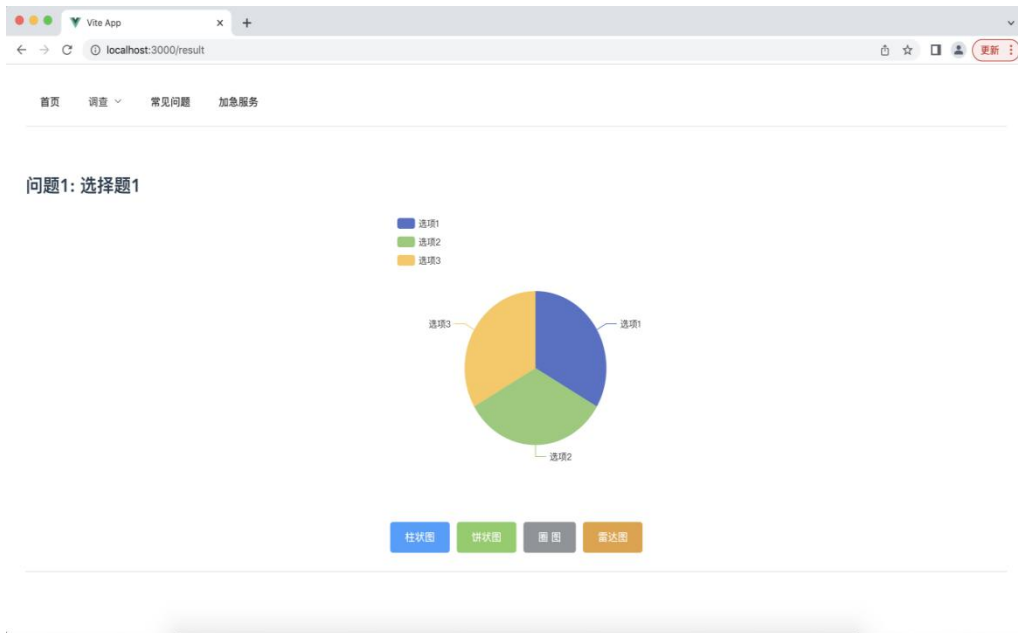


图 5-20 饼状图显示答题人数

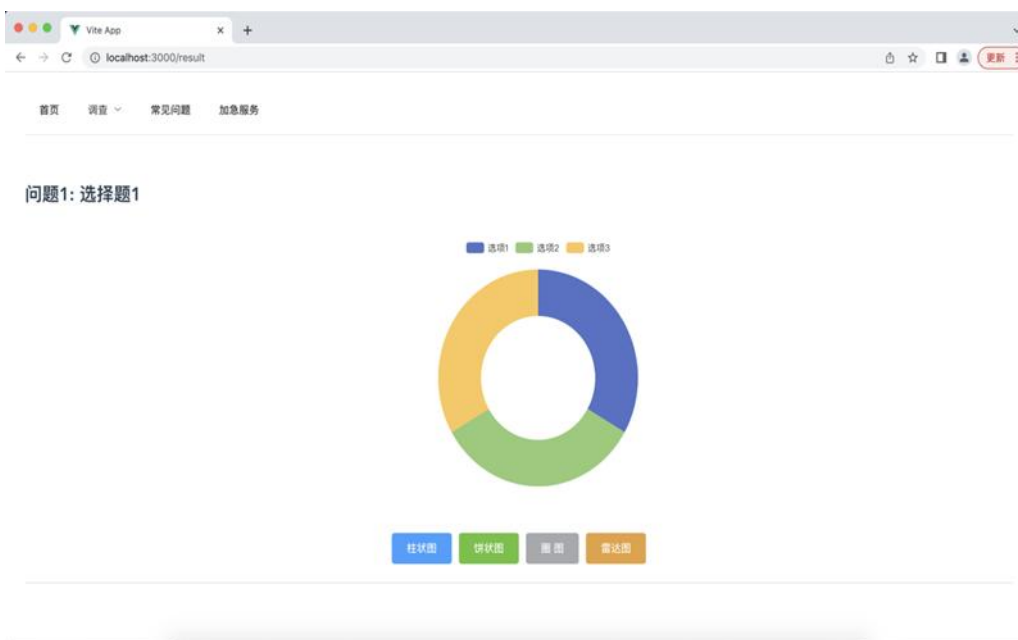


图 5-21 圆圈图显示答题人数

### 5.13 培养计划中的重要课程路径

如图 5-22 所示, 图中红色边框的课程为经过对答卷作答情况分析后得出的培养计划中的重点课程, 他们所经过的路径则用红色的箭头表示。

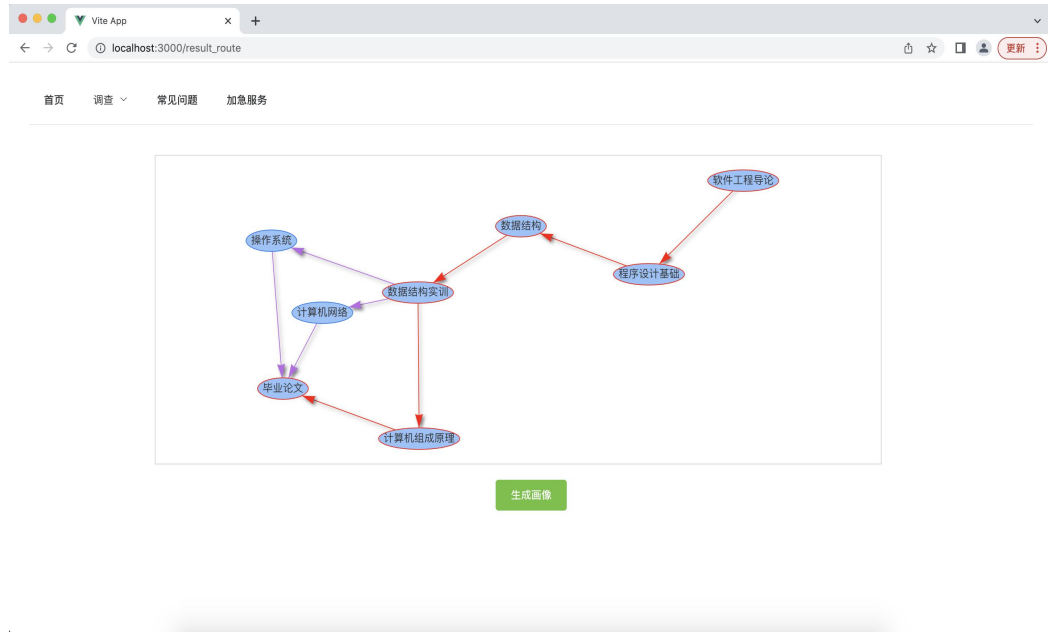


图 5-22 重要课程路径界面



## 6 系统测试

### 6.1 功能测试

功能测试主要测试系统的每个功能是否满足用户的需求而进行测试，这里例举了几个测试。

(1) 注册测试，用户输入格式正确的邮箱与密码进行测试即可，如表 6-1 所示。

表 6-1 用户注册测试

测试编号	前提	输入条件	预期结果	实际结果	是否通过
注册用例 _001	用户未进行注册	邮箱: 2623171556@qq.com 密码: 0096unicorn 确认密码: 0096unicorn	提示: 注册成功	提示: 注册成功	Y
注册用例 _002	用户已注册	邮箱: 2623171556@qq.com 密码: 0096unicorn 确认密码: 0096unicorn	提示: 该邮箱已被注册	提示: 该邮箱已被注册	Y
注册用例 _003	用户未注册	邮箱: 2623171556 密码: 0096unicorn 确认密码: 0096unicorn	提示: 邮箱格式错误	提示: 邮箱格式错误	Y
注册用例 _004	用户未注册	邮箱: 2623171556@qq.com 密码: 0096unicorn 确认密码: 0096	提示: 确认密码和密码不一致	提示: 确认密码和密码不一致	Y

(2) 填写培养计划课程，填写课程权重和课程名称即可，如表 6-2 所示。

表 6-2 课程填写测试

测试编号	前提	输入条件	预期结果	实际结果	是否通过
用例_001	用户已经创建问卷	课程名称: 软件工程 课程权重: 2	转跳到课程路径页面	转跳到课程路径页面	Y
用例_002	用户已经创建问卷	课程名称: 毛主席人生导读 课程权重: 2	提示: 上传不成功, 因为含有敏感词: 毛主席	提示: 上传不成功, 因为含有敏感词: 毛主席	Y
用例_003	用户已经创建问卷	课程名称: 软件工程 课程权重: -1	提示: 权重不能为 0	提示: 权重不能为 0	Y

(3) 找回密码，填写验证码、邮箱、密码、确认密码即可，如表 6-3 所示。

表 6-3 用户找回密码测试

测试编号	前提	输入条件	预期结果	实际结果	是否通过
用例_001	用户已经注册	邮箱: 2623171556@qq.com 验证码: 123456 发送验证码: 123456 新密码: 0096unicorn	提示: 密码更新成功	提示: 密码更新成功	Y
用例_002	用户未注册	邮箱: 2623171556@qq.com 验证码: 123456 发送验证码: 123456 新密码: 0096unicorn	提示: 该用户未注册	提示: 该用户未注册	Y

续表 6-3

测试编号	前提	输入条件	预期结果	实际结果	是否通过
用例_003	用户已经注册	邮箱： 2623171556@qq.com  验证码：1234  发送验证码：123456  新密码：0096unicorn	提示：验证码错误	提示：验证码错误	Y

## 6.2 Jmeter 使用介绍

Jmeter 是一款 Apache 基金会发布的开源的测压软件，可以测试多种网络请求。建立线程组后，可以手动设置运行线程的数量以及线程运转间隔。在结果合集中可以看到 http 请求的请求头及各种返回参数。Jmeter 是一款非常成熟的常用于高并发测压的软件。界面如图 6-1 所示。

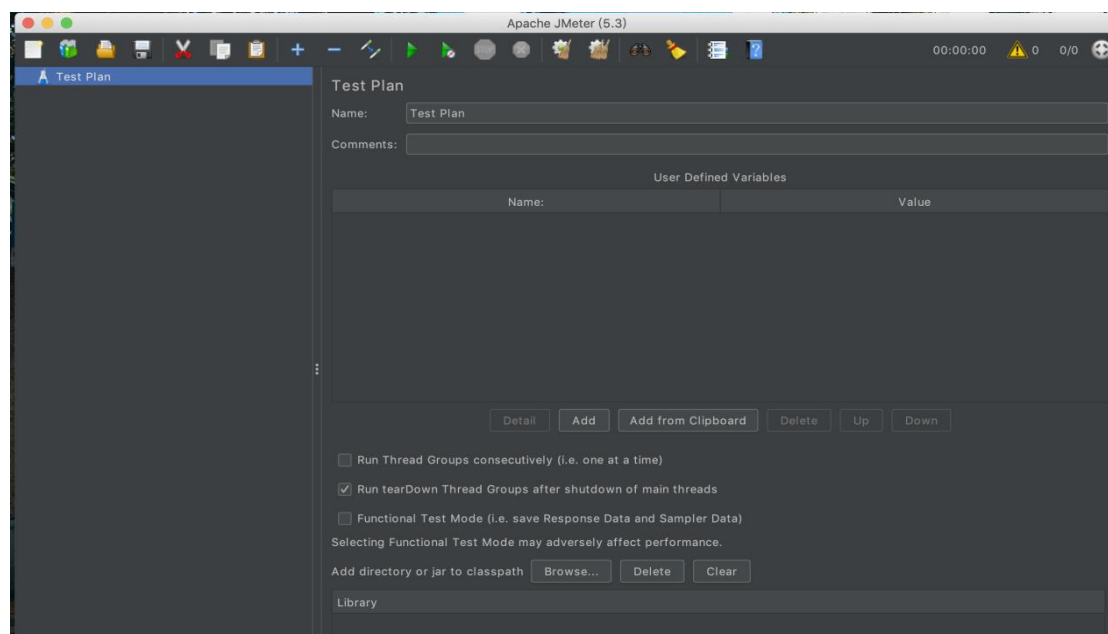


图 6-1 jmeter 测压软件界面图

## 6.3 后端接口测试

在系统中写了较多接口，因此选取几个比较有代表性的接口使用 Jmeter 进行测试，在此 Cookie 的方式去进行测试。

### 6.3.1 登录端口

测试设定在图 6-2 中，测试结果返回正常在图 6-3 中。

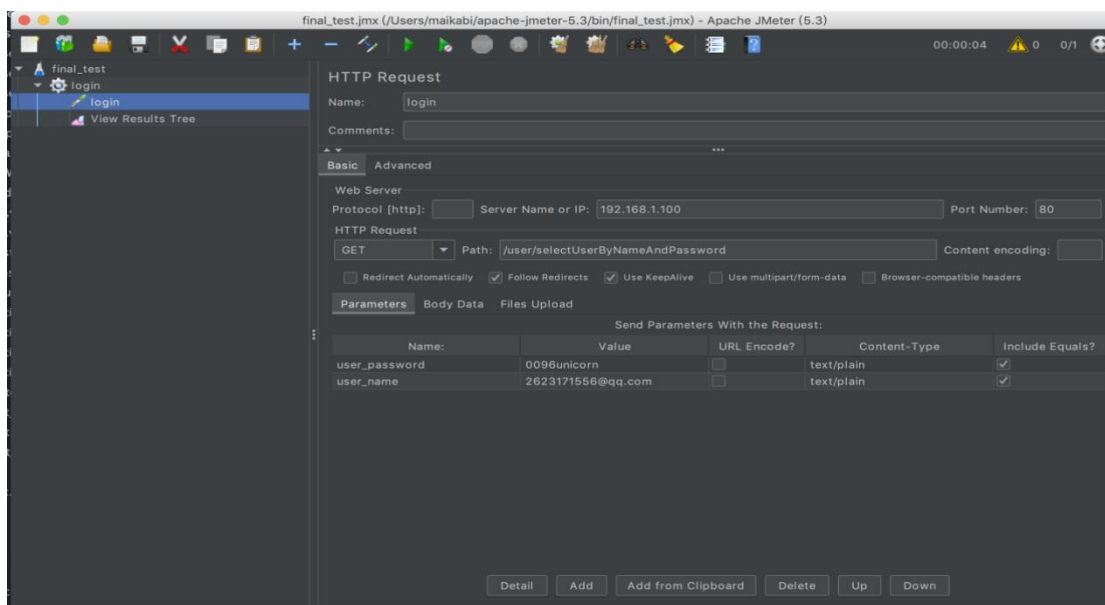


图 6-2 登录功能测试配置

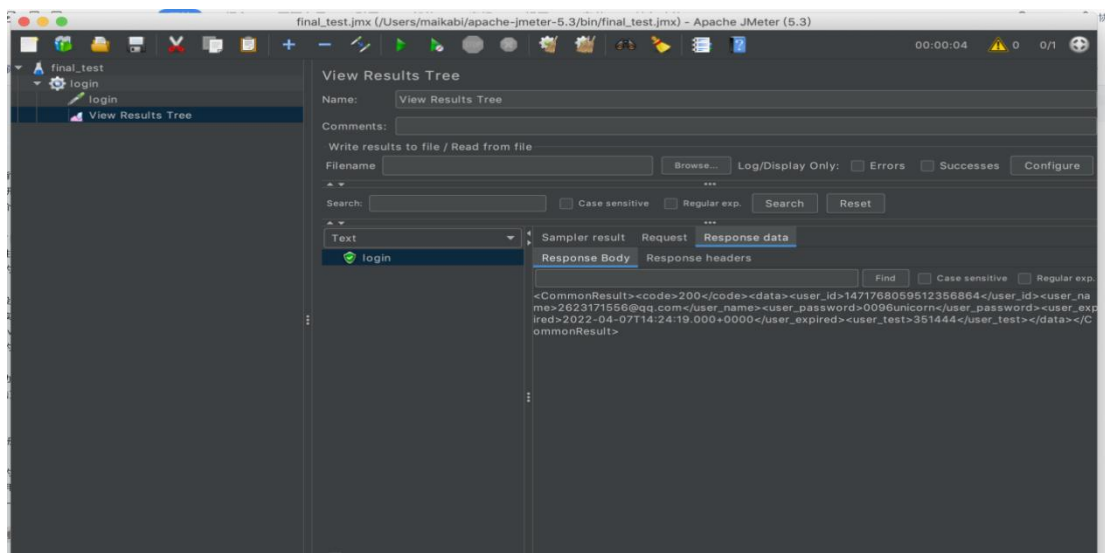


图 6-3 测试结果

## 6.4 服务抢购端口的测压

100 条线程测试准备如图 6-4 所示，结果如图 6-5，6-6 所示，没有出现任何超卖现象。

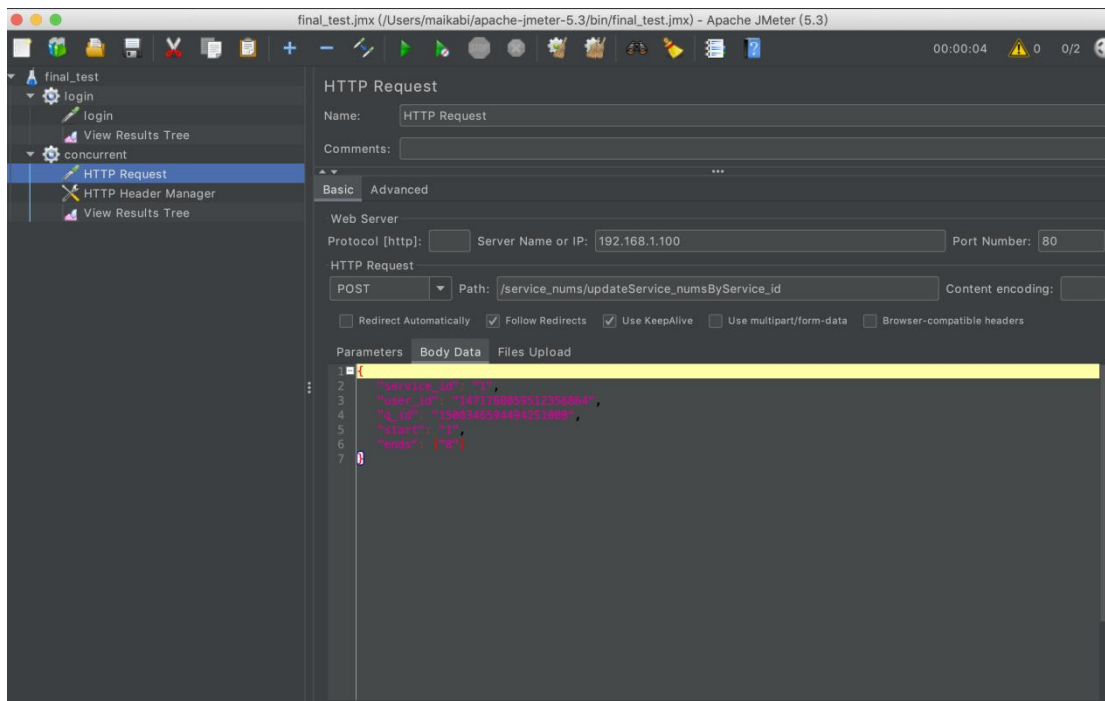


图 6-4 测压配置

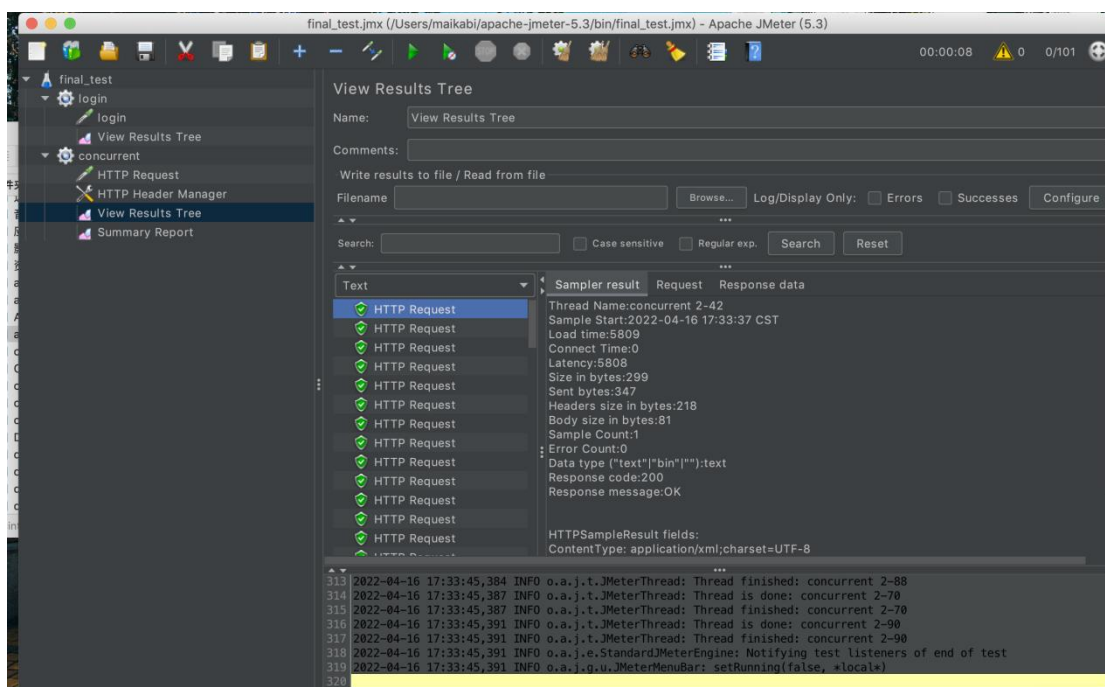


图 6-5 测压结果集

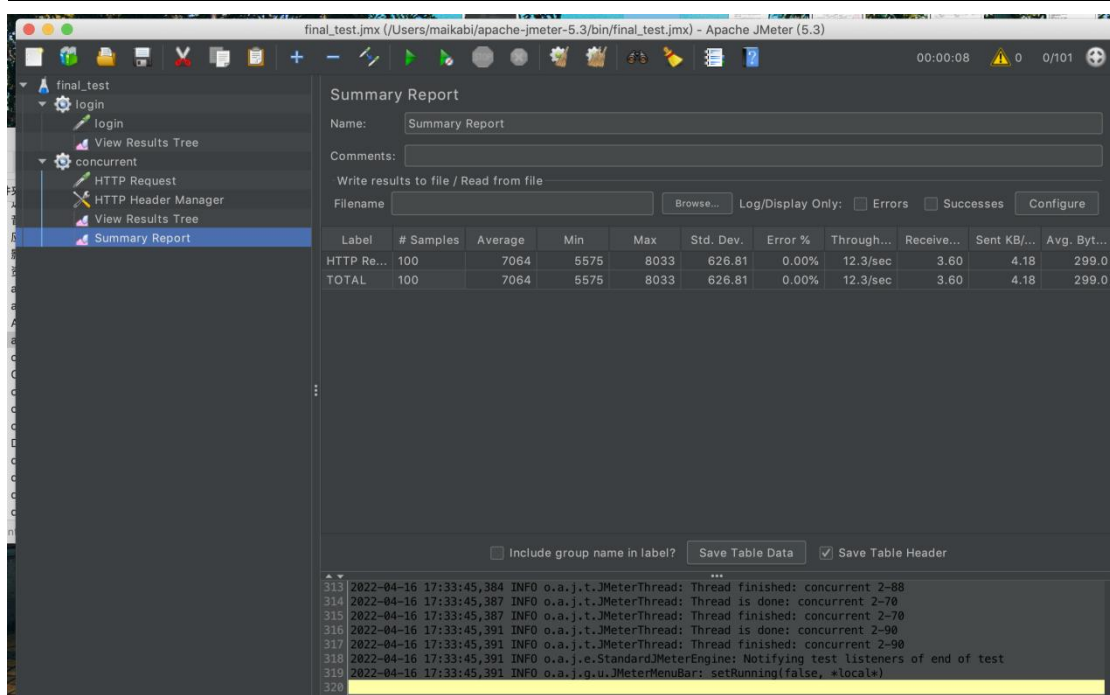


图 6-6 测压汇总报告

## 7 结论

经过几个月的努力，在上海疫情如此严重的情况下，一直两耳不闻窗外事赶进度，终于完成了基于微服务的高校在读生及毕业生数据分析平台的开发以及我的毕业论文。在这次毕业设计中，我充分运用了我大学四年所学的分布式理论，java 语言基础，以及对关系型数据库底层和非关系型数据库底层的知识，尽力开发了一个基于分布式的 web 系统。从需求分析到代码设计，每一步都十分重要，缺少了任何一步，整个项目都无法完整地完成。大学四年中尽管我努力学习，但是通过这次毕业设计仍然让我自己感受到自己对 java 语言底层理解的部分缺失。所以在将来的进修当中，需要学习和吸收的知识还有很多很多。

此次毕业设计主要运用了 java 语言和 SSM 架构，以及 Springcloud 的组件（Eureka，Ribbon，Gateway，Hystrix 等）和关系型数据库 MySQL 和非关系型 Redis，以及异步消息队列 RabbitMQ，开源监听数据库的组件 canal。在完成的过程中遇到许多问题，但依然坚持不懈查阅各种资料。在开发的过程中也能把 web 开发理论和计算机学科的基础课程有一个良好的交互，把迪杰斯特拉算法和 DFA 有穷自动机等数据结构经典算法融入到自己的毕业设计的实践当中去。当然仍有许多细节问题需要改进不算完美，但最后还成功靠自己的努力完成了毕业设计。

在当下毕业生就业难的社会环境下，希望我的系统能为本校的老师们提供宝贵意见，帮助他们在合适的时机调整培养计划，为社会输送大量合适的人才，提高我校就业率，解决学生最大的民生问题——就业。也希望在此系统的帮助下能为同学指明学习的道路，他们能努力学习知识，成为一个对社会有贡献的人才。

## 致谢

首先感谢我的指导老师李丽萍老师，在毕业设计的过程中，从选题到系统的实现，她都用心且亲力亲为指导我。我的毕设指导老师在我隔离期间依然在我提供宝贵的论文修改意见。还要感谢校内的其他同学，比如一些在知名企业工作的同学为我的后端运行效率提出过宝贵建议，也为我的前端风格的 UI 统一性与用户友好度提出了许多宝贵的建议。由于我对前端和用户友好性不够熟悉，因此在编写前端代码的过程中遇到了重重困难，有了老师和同学的指导，我才能一一把这些问题解决，使这个系统成为一个普通人轻易看得懂，用得着的系统。

最后我也要对大学四年来我所有的授课老师老师表示感谢。老师们丰富的计算机知识让我学会到了专业的知识技能。我也要感谢上海第二工业大学，我的母校，在大学中收到母校校训的影响，秉持着“厚生、厚德、厚技”的校训原则，在将来在国外进修的过程中我也会努力拼搏，认真学习计算机相关知识，争取做一个能为社会作出贡献的创新型人才。

在此，我要再次向他们表示衷心的祝福和深深的谢意。



## 参考文献

- [1] Narayanan K , Li J . MAVEN: a tool for visualization and functional analysis of genome-wide association results.[C]// Bioinformatics, 2009. OCCBIO '09. Ohio Collaborative Conference on. Bioinformatics, 2010.
- [2] 周艳聪, 罗雅, 孟巍, 等. 基于 MySQL 的教学案例分享平台设计与实现——以工科课程思政教学案例为例[J]. 现代信息科技, 2021, 5 (19) : 5.
- [3] 陆腾跃, 聂广俊. Spring Cloud 微服务架构综述[J]. 科学与信息化, 2021.
- [4] 王方旭. 基于 Spring Cloud 实现业务系统微服务化的设计与实现[J]. 电子技术与软件工程, 2018 (8) : 2.
- [5] 孙宇, 周纲. 基于微服务架构的资源发现系统平台构建研究[J]. 中国图书馆学报, 2020, 46 (1) : 11.
- [6] 黄健宏. Redis 设计与实现 : The design and implementation of Redis[M]. 机械工业出版社, 2014.
- [7] 师伟伦. 5G 无线通信技术概念及其应用[J]. 科学大众, 2020 (2) : 53-53.
- [8] Schwartz B , Zaitsev P , Tkachenko V , et al. High Performance MySQL[C]// O'Reilly. O'Reilly, 2008.
- [9] Cormen T H , Leiserson C E , Rivest R L , et al. Introduction to Algorithms (Second Edition)[M]. DBLP, 2001.
- [10] 刘世超, 杨斌, 刘卫国. 高性能高可用 Redis 客户端的设计与实现[J]. 电子技术应用, 2022, 48 (1) : 8.