目录

1.实践目的 Aim of Project

本次实践基于是互联网的调查问卷系统。传统的纸质问卷存在很多问题，比如成本高、流程复杂、统计困难等等。为了提高调查的效率、准确性和可控性，迫切需要一个方便设计、发布、统计和整理问卷内容的系统。目标采用 B/S 架构，通过 Web 界面，实现用户可以在个人电脑上使用常见的浏览器访问，完成各种增删改查操作以及前端展示的方法。具体来说，实现问卷的创建、查看和发布功能，进行数据分析，统计各题目的情况。并设计问卷，满足个性化的组卷需求，批量导入题库。

在此过程中，锻炼开发过程中的各项能力，如编写规范说明书，搭建 Spring Boot 工程环境，并利用 Junit 和 log4j 来进行 MyBatis 的测试。同时，学习使用 SonarQube 评估代码质量，使用 Jacoco 来分析测试用例的覆盖率。根据其他人的评价以及自己对其他人成果的评价总结，不断优化系统功能。完成迭代的代码编写与系统构建及改进。

2.预习内容 Preview Content

1. 需求分析：用例图、活动图、状态图等建模工具描述系统功能和行为；
2. 设计模式：类图、序列图、组件图等建模工具描述系统结构和交互；使用 MVC、单例、观察者、工厂等设计模式提高系统的可维护性和可扩展性。
3. 软件测试：白盒测试、黑盒测试、单元测试等测试方法保证软件质量。
4. 软件配置管理：版本控制工具（Git）、构建工具（Maven）。
5. Java：熟悉 Java 语言的基本语法、数据类型、控制结构、异常处理等编写程序；利用 Java 语言的面向对象特性（如封装、继承、多态）、泛型特性、集合框架等编写高效的程序。
6. Vue：学习 Vue 框架实现前端页面的渲染，将数据与视图进行绑定。包括不限于，组件化开发，将页面分解为可复用的组件；路由管理，实现页面之间的跳转。
7. MySQL：数据库存储和管理数据。
8. HTTP 协议：实现客户端和服务器之间的通信，传输请求和响应数据； HTTP 协议的方法（GET、POST）、状态码（200、404）、报文格式。
9. JavaScript：学习使用变量、函数、对象、数组，内置对象（Math、Date、String 等）、内置函数（alert、console.log、setTimeout）、内置事件（onmouseover、

onmouseout、onchange）等编写程序逻辑。实现网页中的交互和动态效果。

3.实践内容和实践过程 Project Content and Process

3.1 概述 Overview

3.1.1 项目概述：

高校满意度调查问卷系统。在高校教育阶段，为提高政府教育管理的针对性与有效性，需要从学生和教师获知其对学校、专业、教学情况、教学资源配套、学校管理水平等多方面评价的信息收集。互联网是效率最高、短时间内涉及人员广的渠道之一，所以依托互联网的调查问卷需求应运而生，通过信息的反馈，进行信息的初步筛选与分析，能够及时反应当前高校情况，为管理手段和政策方针提供依据。

设计问卷系统，问卷发起人登录后可以对项目、问卷进行管理，

查询项目，并针对查询得到的结果项目进行删除、和修改的操作，创建项目填写项目的名称等信息。还可以修改项目的信息、关联的项目问卷等，如果项目中的问卷正在进行中，不能修改。

3.1.2 工作概述：

1. 实现问卷系统的基本功能，包括创建、查看、发布问卷，以及数据的增删改查和前端展示；
2. 实现问卷系统的扩展功能，包括数据分析、个性化组卷和题库管理；
3. 使用 Vue 编写前端页面，并与后端进行数据交互和展示；
4. 学习使用 SonarQube 和 Jacoco 进行代码质量度量和测试用例覆盖率分析；
5. 使用 Junit 和 log4j 完成 mybatis 的单元测试和日志记录；
6. 学习使用 jmeter 进行性能测试和压力测试；
7. 根据互评结果，优化系统功能和界面设计；
8. 搭建 Springboot 工程环境，配置相关依赖和插件；
9. 编写开发规范说明书，规范代码风格、注释、命名等。

3.2 相关技术 Relevant Technologies

3.2.1 理论基础

1. 代码编写要遵循一致的命名规范、使用恰当的注释、避免冗余和复杂度、使用合适的设计模式和架构，做到清晰和可读。
2. 编写文档时，简洁而清晰的文档编写，与代码保持一致并及时更新。
3. 数据库设计原则：
   - 规范：将数据分解为多个表，并消除数据之间的冗余和依赖关系，以提高数据的完整性和一致性。
   - 避免冗余和异常：确保每个表只包含一个主题，并且每个属性只依赖于主键，以避免数据在插入、删除、修改时出现不一致或错误的情况。
   - 使用合适的数据类型：根据数据的特征和用途，选择合适的数据类型，如整数、浮点数、字符串、日期、布尔等，以提高数据的有效性和准确性，减少数据的存储空间和处理时间。
   - 建立索引和约束：为经常查询或排序的字段建立索引，以提高数据检索的速度和效率。为关键字段建立约束，如主键、外键、唯一性、非空性、范围等。在后续数据生成时实现一致性。
4. 前端开发的原则：
   - 响应式设计：使用灵活和自适应的布局，使用户界面能够根据不同的设备和屏幕尺寸进行调整，并保持良好的视觉效果和交互体验。
   - 可用：使用直观和友好的设计，使用户界面能够方便地完成用户任务，并提供必要的提示和反馈信息。
   - 美观：提供色彩、布局等有设计感的界面，给用户以视觉享受。
5. 后端开发的原则：
   - 模块化：将业务逻辑和数据交互分解为多个独立且可复用的模块，并通过接口进行通信和协作。
   - 封装：将模块内部的实现细节隐藏起来，并通过公开的方法提供功能和服务，以提高模块的安全性和可维护性。
   - 重用：利用已有的代码或组件来实现新的功能，重视代码复用，减少编码工作量，提高代码质量，注重模块化和标准化，与前面的模块化和封装紧密联系。
   - 测试：及时进行代码审查与测试，对模块的功能和性能进行系统地检验和验证，以发现并修复错误，验证软件是否符合需求和预期，提升系统的可靠性和稳定性。

3.2.2 方法

1. 需求驱动开发方法：以需求为核心开发，先明确和分析需求，然后根据需求来设计、实现、测试和交付。
2. 增量开发方法：将项目分解为多个小型的子项目，每个子项目都会增加软件的功能，并且可以单独地进行测试和部署。增量开发可以根据需求的优先级来安排和执行子项目，每次完成一个子项目都会向客户交付一个更完善的软件产品，并且通过客户的反馈来确定下一个子项目的需求。
3. 单一职责：每个类或方法应该只有一个明确的责任，避免功能过于复杂和耦合度高的代码。
4. 高内聚低耦合：模块之间保持高内聚，模块内部的元素相互关联紧密，而模块之间的依赖关系应尽量降低。
5. 设计模式：如工厂模式、单例模式、观察者模式等，以提高代码的可维护性、可扩展性和可复用性。
6. 模块化和组件化：将大型任务拆分为小模块或组件，每个模块或组件负责一个特定的功能，以提高代码的可管理性和可测试性。
7. 异常处理：合理处理异常，以确保程序的稳定性和可靠性。
8. 前后端分离：让前端和后端各自专注于自己的领域，提高开发效率和质量，以及方便协作和维护。前端负责创建用户界面，处理用户交互，展示数据，调用后端接口等。后端负责处理业务逻辑，存储数据，提供接口等。接口负责连接前端和后端，传递数据，定义规则。

3.2.3 相关工具与环境

工具、环境、技术	名称	版本	简介
硬件部署环境	联想拯救者	R7000	部署系统和应用
软件部署环境	Windows	10.0	操作系统
开发环境	JDK	1.8	Java 开发工具包
	Node.js	14.17.0	JS 运行环境
浏览器	谷歌	114.0.5735.198	网页浏览
开发工具	Idea	2021.3.1	编写和调试代码
编程语言	Java	8	编程语言
	JavaScript	2022	脚本语言
测试工具	Junit	4	单元测试框架
版本管理工具	Git	2.40.1	管理和追踪代码的版本
	Maven	3.3.9	项目依赖和打包
数据库	MySQL	8.0.3	开源关系型数据库管理系统
数据库可视化工具	Navicat	16	数据库管理和开发工具，提

			供图形化界面和操作功能
应用服务器	Tomcat	9.0.38	Java Web 应用服务器
框架	SpringBoot	2.3.1	基于 Spring 框架的快速开发框架
日志记录工具	log4j	2.14.1	记录应用程序运行时的日志信息
数据访问层框架	mybatis	3.5.13	简化数据库访问和操作
代码质量度量工具	SonarQube	8.9.10.61524	用于静态代码分析和质量度量
测试覆盖率工具	Jacoco	0.8.9	评估测试用例的覆盖率
前端框架	vue	5.0.8	前端 JavaScript 框架，构建交互式的 Web 界面
性能测试工具	jmeter	5.5	模拟负载和评估系统的性能

表格 1

3.3 系统设计 System Design

本人在系统设计过程中与小组成员一起进行功能分析，负责架构设计，接口设计模块设计，数据库设计，交互设计。成果如下。交互设计成果在报告后续测试部分给出。

3.3.1 功能分析

1. 数据分析-题目统计功能：
   - 在问卷列表页面，点击统计链接，可以统计当前问卷的答题情况。
   - 单题统计：统计单题的答题情况，显示每个选项的选择数量以及占总答题人数的比例。
   - 答题总人数统计。
2. 查看问卷功能：
   - 通过项目列表的查看按钮，进入问卷查看列表页面。
   - 在问卷查看列表页面，点击统计按钮，进入答卷查看页面。
   - 输入答卷人相关信息，可以查到答卷人当前项目中所有问卷的答卷明细。
   - 答卷明细页面显示当前答卷人、当前问卷内容以及答卷人的答题结果。当前问卷不可编辑。
3. 发布问卷功能：
   - 通过项目列表的查看按钮，进入问卷列表查看界面。
   - 问卷列表页面包含问卷名称、问卷发布时间、预览、发布、删除、统计等操作链接。
   - 可预览问卷，模拟答题过程。

- 可发布问卷并生成问卷访问链接。

- 删除问卷功能，正在进行中的问卷不能修改，删除的问卷为逻辑删除，系统保留数据。

4. 创建问卷功能：

- 通过项目列表中的创建问卷按钮或空白模板创建按钮，打开创建问卷页面。

- 创建问卷页面包含所属项目下拉框和调查类型下拉框，默认值根据项目名称和学生类型。

- 可选择空白模板创建，填写问卷名称和调查说明。点击立即创建按钮，提示必填项不可为空；否则进入创建题目页面。

- 创建成功后直接进入编辑试题页面，可选择不同类型的题目，支持修改和删除题目。

5. 问卷设计-个性化组卷功能：

- 在创建问卷页面中，增加设置答卷的开始时间、截止时间和答卷次数等功能。

- 限时问卷设置：设定答卷时间范围，只有在限定时间内答卷链接才有效，超出范围不能答卷或有相应提示。

- 定次问卷设置：设定答卷次数，每个账户或主机可答卷的次数。

- 发放范围设置：面向公众和指定群组。选择指定群组时，显示系统内置的群组，调查问卷发送给指定的群组。

- 问卷风格设置：内置不同问卷风格，根据选择设定问卷的风格。

6. 题库管理-批量导入功能：

- 在创建问卷页面中，可以批量导入题目。

- 规划建立题库功能，可通过数据库直接配置题目。

- 可通过题库筛选调查问卷题目，并批量导入题目信息。

- 导入的题目可作为答卷的题目，可以进行再编辑，不影响原题库中的题目内容。

3.3.2 架构设计

1. 用户界面层（User Interface Layer）：

   用户界面层负责与用户进行交互，展示问卷列表、统计结果、答卷明细等信息。

   包括问卷列表页面、问卷统计页面、答卷查看页面、创建问卷页面等。

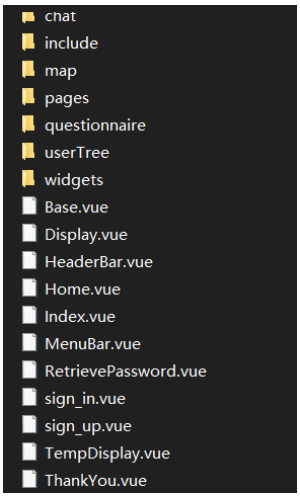


图 1

2. 应用逻辑层（Application Logic Layer）：

应用逻辑层处理用户请求，调用适当的服务和算法来完成相应的功能。包括问卷管理、统计分析、答卷管理、题库管理等模块。

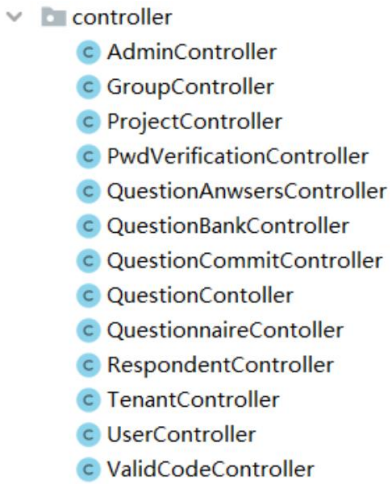


图 2

3. 服务层（Service Layer）：

服务层提供各种服务接口，供应用逻辑层调用，完成具体的业务逻辑处理。包括问卷服务、统计服务、答卷服务、题库服务等。

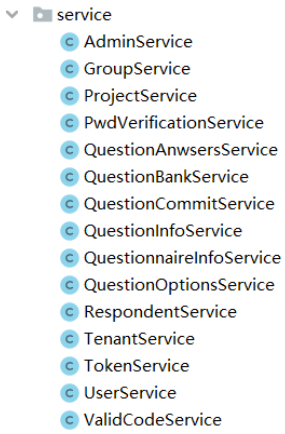
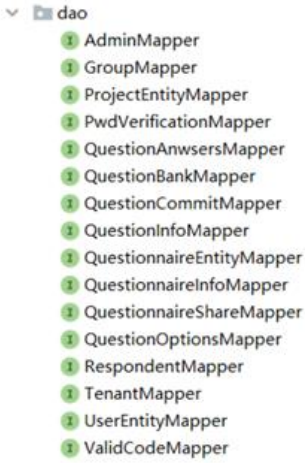


图 3

4. 数据访问层（Data Access Layer）：

数据访问层负责与数据库进行交互，提供数据的读取、写入和查询等功能。

包括问卷数据访问、统计数据访问、答卷数据访问、题库数据访问等。



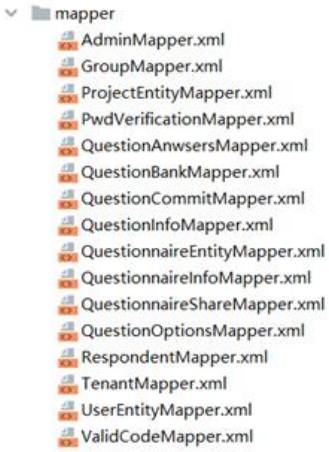


图 4

5. 数据库层（Database Layer）：

数据库层存储问卷、统计、答卷和题库等数据。

使用适当的数据库管理系统（MySQL）来管理数据。

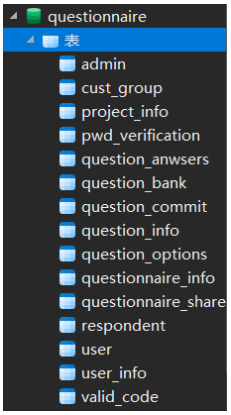


图 5

3.3.3 模块设计

1. 问卷管理模块：
   - 创建问卷：包括创建空白问卷和从题库导入题目。
   - 编辑问卷：对已创建的问卷进行编辑和修改。
   - 预览问卷：在问卷列表中查看问卷内容，进行模拟答题。
   - 发布问卷：将问卷发布，并生成问卷访问链接。
   - 删除问卷：逻辑删除已发布或未发布的问卷。
2. 统计分析模块：
   - 单题统计：统计单个题目的答题情况。
   - 问卷统计：显示当前问卷所有题目的统计状况，包括选项选择数量和占总答题人数的比例。
   - 单题分类统计：对相同内容的问题进行分类统计，并以表格、柱状图等形式展示结果。
3. 答卷管理模块：
   - 答卷查看：根据答卷人的相关信息，查看其在当前项目中所有问卷的答卷明细。
   - 明细查看：查看当前问卷的答卷情况，显示答卷人和答题结果。
   - 答卷限制：根据设置的限时和定次规则，限制答卷的时间范围和次数。
4. 题库管理模块：
   - 题目导入：从题库中选择题目并批量导入到问卷中。
   - 题目编辑：对导入的题目进行编辑和修改。
   - 题目分类：对题库中的题目进行分类管理和统计。
5. 模块内交互模型：

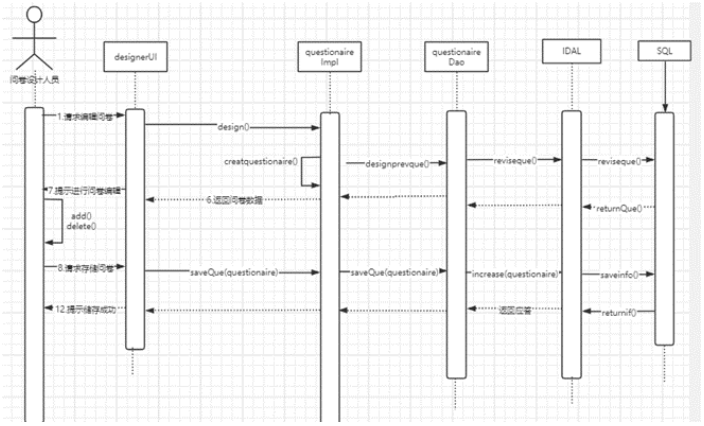


图 6

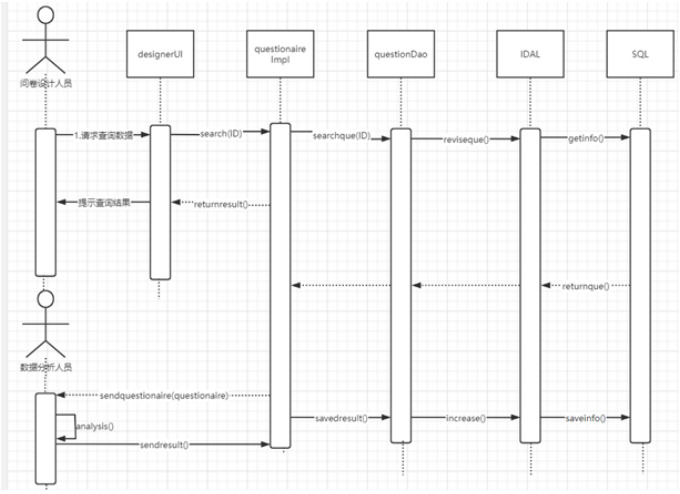


图 7

3.3.4 数据库设计

数据库设计如下：
cust_group：记录不同的人群分组。
project_info：项目信息。
question_anwsers：用户提交的问卷的答案。
question_bank：可以导入问卷的问卷题库。
question_commit：用户提交的问卷信息存储。
question_info：问卷中的题目信息。
question_options：问卷题目的选项信息。
questionnaire_info：问卷的信息。
questionnaire_share：问卷的分享链接。
Respondent：答卷人信息。
user_info：用户信息。

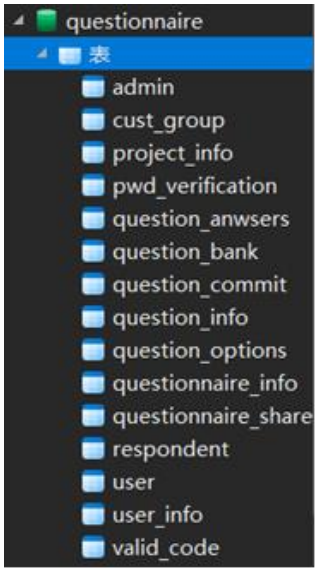


图 8

user_info：用户信息。包含主键 id，用户名和密码（用以登陆），补充信息，

诸如性别、创建时间、创建人、姓名等等，通过 role_id 划分角色，可以对应答卷人，问卷系统使用者，问卷系统管理者等权限。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	varchar	50		☑	☐	🔑1	用户表主键
username	varchar	32		☐	☐		用户名
password	varchar	160		☐	☐		密码
business_no	varchar	255		☐	☐		学号/工号
reaal_name	varchar	32		☐	☐		姓名
start_time	datetime			☐	☐		开始时间
school	varchar	64		☐	☐		学校
major	varchar	32		☐	☐		专业
class	varchar	32		☐	☐		班级
gender	varchar	2		☐	☐		性别
wx_id	varchar	32		☐	☐		微信号
qq_id	varchar	32		☐	☐		qq号
telephone	varchar	32		☐	☐		手机号
email	varchar	32		☐	☐		邮箱
stop_time	datetime			☐	☐		截止时间（时间戳）
status	varchar	2		☐	☐		是否启用（1启用，0不启用）
created_by	char	32		☐	☐		创建人
creation_date	datetime			☐	☐		创建时间

图 9

project_info：项目信息。用以实现关于项目功能的增删改查。Id 是项目 id，作为主键。Userid 外键对应 user_info 的主键，说明这个问卷由谁创建。Projectname 是项目名称，project_content 是项目描述。补充信息包含创建人、创建时间等等。

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	varchar	50		☑	☐	🔑1	项目表主键
user_id	varchar	50		☐	☐		用户id（没有用）
project_name	varchar	100		☐	☐		项目名称
project_content	text			☐	☐		项目说明
created_by	char	32		☐	☐		创建人
creation_date	datetime			☐	☐		创建时间
last_updated_by	char	32		☐	☐		最后修改人
last_update_date	datetime			☐	☐		最后修改时间

图 10

名	字段	被引用的模式	被引用的表（父）	被引用的字段	删除时	更新时
uuu	user_id	questionnaire	user_info	id	RESTRICT	RESTRICT

图 11

questionnaire_info：问卷的信息。Questionnaire_id 作为主键，唯一的标识问卷。Title 为问卷题目。User_id 也是对应 user_info 中 id 的外键，标识问卷由哪个用户创建。Status 标识问卷的发布状态。Desc 记载问卷描述。Project_id 与项目 id 进行外键绑定，每个问卷存在于项目之下。Is_template 标识是否为模板问卷。其他属性作为补充信息，比如发起时间，截止时间，所属人群等等。

名	类型	长度	小数点	不是 null	虚拟	键	注释
questionnaire_id	varchar	50		☑	☐	🔑1	
user_id	varchar	50		☐	☐		
status	int			☐	☐		
desc	varchar	255		☐	☐		
start_time	datetime			☐	☐		
end_time	datetime			☐	☐		
project_id	varchar	50		☐	☐		
title	varchar	255		☐	☐		
is_template	int			☐	☐		
survey_crowd	varchar	16		☐	☐		
template_id	varchar	50		☐	☐		

图 12

名	字段	被引用的模式	被引用的表（父）	被引用的字段	删除时	更新时
pi1	project_id	questionnaire	project_info	id	SET NULL	CASCADE
ui	user_id	questionnaire	user_info	id	SET NULL	CASCADE

图 13

question_info：问卷中的题目信息。Question_id 作为主键唯一的标识问题。每个问题都有 questionnaire_id 与问卷进行外键绑定。Type 标识是单选还是多选等。Is_must 标识是否为必选题。

名	类型	长度	小数点	不是 null	虚拟	键	注释
question_id	varchar	50		☑	☐	1	
question_title	varchar	255		☐	☐		
questionnaire_id	varchar	50		☑	☐		
type	varchar	16		☐	☐		
status	int			☐	☐		
is_must	int			☐	☐		
sort_no	int			☐	☐		
image	text			☐	☐		

图 14

名	字段	被引用的模式	被引用的表（父）	被引用的字段	删除时	更新时
question_info_	questionnaire_id	questionnaire	questionnaire_info	questionnaire_id	CASCADE	CASCADE

图 15

question_options：问卷题目的选项信息。Option_id 唯一的标识每个选项，与问题 id 进行外键绑定。Title 存储选项描述。

名	类型	长度	小数点	不是 null	虚拟	键	注释
option_id	varchar	50		☑	☐	1	
question_id	varchar	50		☐	☐		
title	varchar	255		☐	☐		

图 16

question_bank：可以导入问卷的问卷题库。存储作为可选导入题库的问题。

名	类型	长度	小数点	不是 null	虚拟	键
question_id	varchar	50		☑	☐	1
question_title	varchar	255		☐	☐	
type	varchar	16		☐	☐	
status	int			☐	☐	
is_must	int			☐	☐	
image	text			☐	☐	

图 17

question_commit：用户提交的问卷信息存储。Commit_user_id 与提交人 id 进行外键绑定。每次提交都是唯一的。并且与问卷 id 也进行外键绑定。并存储提交时间，作答时长，作答 id 等补充信息。

名	类型	长度	小数点	不是 null	虚拟	键
commit_id	varchar	50		☑	☐	1
commit_user_id	varchar	50		☐	☐	
questionnaire_id	varchar	50		☐	☐	
commmit_time	datetime			☐	☐	
anwser_time	int			☐	☐	
commit_ip	varchar	16		☐	☐	

图 18

名	字段	被引用的模式	被引用的表（父）	被引用的字段	删除时	更新时
question_comr	questionnaire_id	questionnaire	questionnaire_info	questionnaire_id	RESTRICT	RESTRICT
question_comr	commit_user_id	questionnaire	respondent	id	RESTRICT	RESTRICT

图 19

question_anwsers：用户提交的问卷的答案。通过 commit_id 与提交信息进行绑定，question_id 与问题绑定，记录选择的内容。

名	类型	长度	小数点	不是 null	虚拟	键	注
anwser_id	varchar	50		☑	☐	1	
commit_id	varchar	50		☐	☐		
question_id	varchar	50		☐	☐		
anwser	text			☐	☐		

图 20

名	字段	被引用的模式	被引用的表（父）	被引用的字段	删除时	更新时
question_anws	commit_id	questionnaire	question_commit	commit_id	RESTRICT	RESTRICT
question_anws	question_id	questionnaire	question_info	question_id	RESTRICT	RESTRICT

图 21

questionnaire_share：问卷的分享链接。包含问卷 id，问卷链接，链接分享人，链接发布时间等。

名	类型	长度	小数点	不是 null	虚拟
questionnaire_id	varchar	50		☑	☐
share_link	varchar	255		☐	☐
share_by	varchar	255		☐	☐
share_tim	datetime			☐	☐
user_id	varchar	50		☐	☐

图 22

3.3.5 接口设计

后端			
ProjectController	@RequestMapping("/project")	queryProjectList	查全部项目
		deleteProjectById	删除
		addProjectInfo	添加
		modifyProjectInfo	修改
		queryProjectName	查询特定
后略			
前端			
问卷设计者操作		designOpera	/api/design
		loginPost	/api/admin/userLogin
		queryProjectList	/api/project/queryProjectList

问卷操作	addQuestionnaire	/api/questionnaire/add
	updateQuestionnaire	/api/questionnaire/update
	queryQuestions	/api/questionnaire/queryQuestions
	publishQuestionnaire	/api/questionnaire/publish
	analysis	/api/questionnaire/analysis
	answerView	/api/questionnaire/answerView
题目操作	sageQuestions	/api/question/save
问卷回答者操作	answerOpera	/api/questionnaire/answerView
	answerQuestionnaire	/api/questionnaire/anwser
用户操作	queryUserList	/api/admin/queryUserList
	modifyUserStatus	/api/admin/modifyUserStatus
	deleteUser	/api/admin/deleteUserInfoById
	resetPwd	/api/admin/resetPwd
	modifyUserInfo	/api/admin/modifyUserInfo
	addUserInfo	/api/admin/addUserInfo

表格 2

如：

```
@RestController
@RequestMapping("/project")
public class ProjectController {
```

图 23

```
@RequestMapping(value = "/deleteProjectById", method = RequestMethod.POST, headers = "Accept=application/json")
public HttpResponseEntity deleteProjectById(@RequestBody ProjectEntity projectId) {
    HttpResponseEntity httpResponseEntity = new HttpResponseEntity();
    if (!projectService.hasPublishedQuestionnaire(projectId.getId())) {
        httpResponseEntity.setCode(Constans.SUCCESS_CODE);
        projectService.deleteProjectById(projectId.getId());
    } else {
        httpResponseEntity.setCode(Constans.EXIST_CODE);
        httpResponseEntity.setMessage("有正在进行中的问卷，无法删除项目");
    }
    return httpResponseEntity;
}
```

图 24

```
//问卷设计者操作
export const designOpera = data => {
  return axios.post("/api/design", data).then(res => res.data);
};

export const loginPost = data => {
  return axios.post("/api/admin/userLogin", data).then(res => res.data);
};

export const queryProjectList = data => {
  return axios.post("/api/project/queryProjectList", data).then(res => res.data);
};
```

图 25

3.3.6 交互设计

1. 用户需求和目标：
   - 快速设计和发布问卷，并能够准确地收集和统计调查结果。
   - 易用性，能够满足不同类型的问卷设计需求。
2. 功能和结构：
   - 提供友好的问卷设计界面，包括创建、编辑和组织问题、选项和页面的功能。
   - 支持多种题型，如单选题、多选题、文本题等，以满足不同类型的调查需求。
   - 具备问卷模板库，供用户选择现有的模板或自定义设计问卷。
   - 提供问卷预览功能，让用户在发布之前查看问卷的外观和逻辑。
   - 支持问卷发布和分享，包括生成唯一的问卷链接、嵌入到网站或通过邮件发送给受访者。
   - 提供实时统计和分析功能，以便能够及时了解和分析收集到的数据。
   - 支持数据分析。
3. 界面和视觉：
   - 设计简洁、直观的用户界面，使用户能够轻松导航和操作系统的各项功能。
   - 使用清晰的图标、按钮和颜色，以增强用户对功能和操作的识别和理解。
   - 保持界面的一致性，使用户在整个问卷设计和统计过程中能够保持一致的视觉感知。
4. 行为和反馈：
   - 设计用户友好的交互行为，如点击、拖拽、滑动等，以便用户能够自然地与问卷系统进行交互。
   - 提供实时反馈，如状态指示器、进度条和提示信息，让用户了解其操作的结果和进展。

- 在问卷设计过程中，及时验证用户的输入和逻辑设置，提供错误提示和修正机制，以确保问卷的正确性和完整性。

5. 原型图如：

图 26

3.3.7 详细设计说明书等文档编写

按照任务资料中的文档资料，编写详细设计说明书。

3.4 编码 Coding

编码部分本人负责实现后端创建问卷、查看问卷、发布问卷、数据分析、个性化问卷、题库导入功能并采用前后端分离思想，利用 node.js、js、vue 编写前端界面并实现展示，确保系统成功运行。

3.4.1 后端：

问卷向下划分为问卷信息，问卷题目，题目选项，创建问卷、查看问卷、发布问卷、数据分析、个性化问卷、题库导入功能通过 controller 调用 service，在数据库中通过 xml 中编写的 sql 语句调用即可。

具体代码放在源代码中，报告部分只做部分展示。

如问卷 questionnaire 信息的获取：

Mapper.xml：

```
1. <select id="selectByPrimaryKey" parameterType="java.lang.St
   ring" resultMap="BaseResultMap">
2.         <!--@mbg.generated-->
3.         select
```

```
        <include refid="Base_Column_List"/>
        from questionnaire_info
        where questionnaire_id = #{questionnaireId,jdbcType=VARCHAR}
    </select>
```

Service：

```
public QuestionnaireInfo answerView(String questionnaireId, String commitId) {
        QuestionnaireInfo questionnaireInfo = questionnaireInfoMapper.selectByPrimaryKey(questionnaireId);
        if (questionnaireInfo != null) {
            if(StringUtils.isNotBlank(commitId)){
                questionnaireInfo.setQuestions(queryQuestionsWithAnwser(questionnaireId,commitId));
            }else{
                questionnaireInfo.setQuestions(queryQuestions(questionnaireId));
            }
        }
        return questionnaireInfo;
    }
```

Controller：

```
@RequestMapping(value = "/get", method = RequestMethod.POST, headers = "Accept=application/json")
    public HttpResponseEntity get(@RequestBody QuestionnaireInfo questionnaireInfo) {
        HttpResponseEntity httpResponseEntity = new HttpResponseEntity();
        httpResponseEntity.setCode("666");
        QuestionnaireInfo questionnaireInfo1 = questionnaireInfoService.getQuestionnaireInfoById(questionnaireInfo.getQuestionnaireId());
        httpResponseEntity.setData(questionnaireInfo1);
        return httpResponseEntity;
    }
```

3.4.2 前端：

配置路由：

```
Vue.use(Router)

export default new Router({
```

```
  mode: 'history',
  routes: [
    {
      path: '/',
      name: 'Index',
      meta: { hidden: true },
      component: Login
    },
    {
      path: '/login',
      name: 'Login',
      meta: { hidden: true },
      component: Login
    }
…
```

调用 api：

```
export const post = function(url,data,files){
  return new Promise((resolve,reject) => {
    service({
      url:url,
      method:'post',
      data:data
    }).then(res => resolve(res)).catch(err =>reject(err))

  })
}
```

设计每个页面：

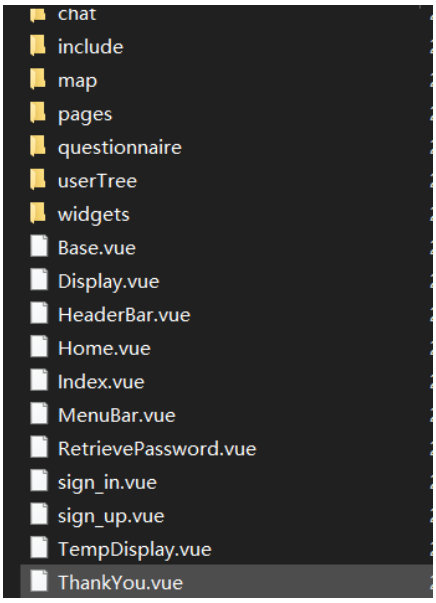


图 27

如项目管理界面进行问卷管理（创建、分析、预览）：

图 28

利用 elementui 进行组件导入与美化：

```
<el-row :gutter="30" >
      <el-col :span="8" style="margin-left: 18px">
        <!--<el-card class="sidebar-card" v-for="item in detail" :key="item.id">-->
        <el-card>
          <el-row style="margin-top: 5px;margin-bottom: 5px;" :gutter="20">
            <el-col :span="30">
              <el-button size="small" type="primary" style="margin-left: 10px;height: 40px;"
                @click="projectDialog = true, editProject = {}">添加项目</el-button>
            </el-col>
            <el-col :span="30">
              <el-input placeholder="输入项目名称进行过滤" v-model="filterText"></el-input>
            </el-col>
          </el-row>
```

脚本编写通过 js 调用 api 实现获取数据显示到界面上：（如问卷添加）

```
addProject() {
      if (!this.editProject.id) {
        post('/api/project/addProjectInfo', this.editProject)
          .then((res) => {
            this.$message({
              type: 'success',
              message: '新建项目成功',
              showClose: true,
            })
```

```
            this.projectDialog = false
            this.getWjList()
          })
          .catch((err) => {
            this.$message({
              type: 'error',
              message: err,
            })
          })
      } else {
        post('/api/project/modifyProjectInfo', this.editProject)
          .then((res) => {
            this.$message({
              type: 'success',
              message: '修改项目成功',
              showClose: true,
            })
            this.projectDialog = false
          })
          .catch((err) => {
            this.$message({
              type: 'error',
              message: err,
            })
          })
      }
    },
```

又如，当查看问卷时，会调用导入的自定义组件并传入参数，丰富界面并实现数据传输。

```
  //查看问卷详情
    lookDetail() {
      //console.log(this.curQuestionnaire.questionnaireId,"--------",this.curQuestionnaire.title,"-----------", this.curQuestionnaire.desc);
          this.$refs.design.init(this.curQuestionnaire.questionnaireId,this.curQuestionnaire.title, this.curQuestionnaire.desc)
          this.$refs.dataShow.dataAnalysis(this.curQuestionnaire.questionnaireId)
    },
```

通过 style 部分实现布局：

```
<style scoped>
#projectHome {
```

```
3.    font-size: 16px;
4.  }
5.
6.  .home {
7.    width: 100%;
8.    height: calc(100vh - 100px);
9.    text-align: center;
10. }
11. ...
```

通过 vue 基本指令如 v-if、V-on，结合全局组件、私有组件，调用 data 和 methods，实现组件切换与父子组件传值，ref 获取元素和组件等方法实现前端设计。

除了要求给出的必做和选做题目之外，我还增添了通过题目导入 xls、xlsx 文件添加题目到题库的功能，使个性化更进一步。

3.5 测试 Testing

测试部分我主要负责用户管理和项目管理，以及问卷创建和查看和发布部分的 junit 测试以及系统成功运行的功能整体测试。

3.5.1 Junit 测试

编写 junit 代码进行测试：

如：

```
@RunWith(SpringRunner.class)
@SpringBootTest
public class QuestionnaireApplicationTests {

    @Resource
    private UserController userController;
    //Logger log = Logger.getLogger(QuestionnaireApplicationTests.class);


    @Test
    public void testSelect() {
        UserEntity userEntity = new UserEntity();
        //HttpResponseEntity httpResponseEntity=new HttpResponseEntity();
        //报错
        userEntity.setUsername("testCatch");
        HttpResponseEntity httpResponseEntity = userController.queryUserList( map: null);
        Assert.assertNull(httpResponseEntity.getCode());
        //log.info("---报错---");


        //存在
        userEntity.setUsername("admin");
        httpResponseEntity = userController.queryUserList( map: null);
        Assert.assertEquals( expected: "666", httpResponseEntity.getCode());
        //log.info("---存在---");
        //log.info(httpResponseEntity.getData().toString());

        //不存在
        userEntity.setUsername("no");
        httpResponseEntity=userController.queryUserList( map: null);
        Assert.assertEquals( expected: "0", httpResponseEntity.getCode());
```

图 29

测试效果：

图 30

3.5.2 系统功能测试及整体界面展示

1. 登录

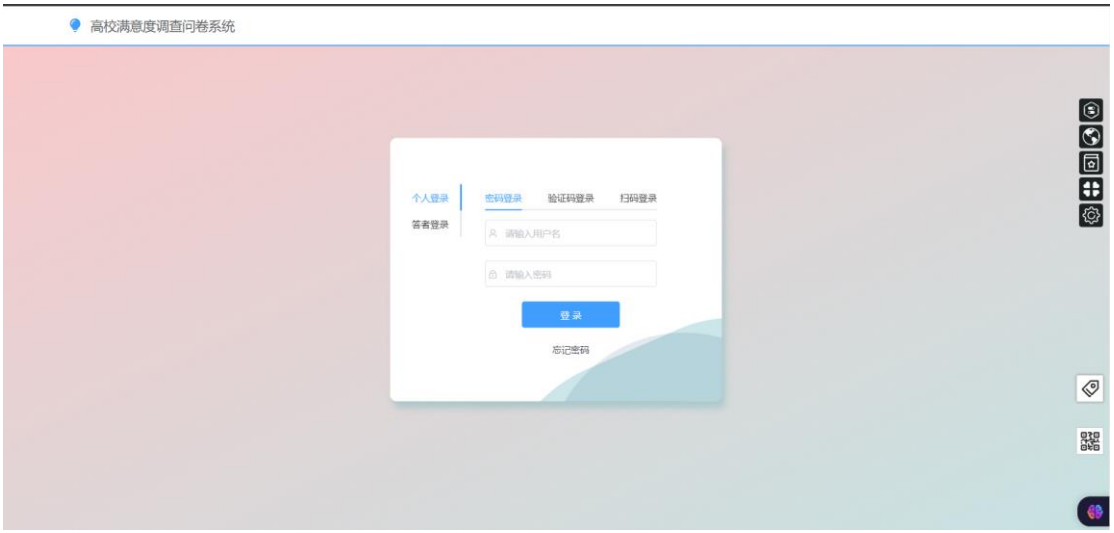


图 31

主界面：（可以通过左上角项目管理切换至项目主视角）

图 32

2. 项目管理

项目管理主界面（增删改查点击按钮）：

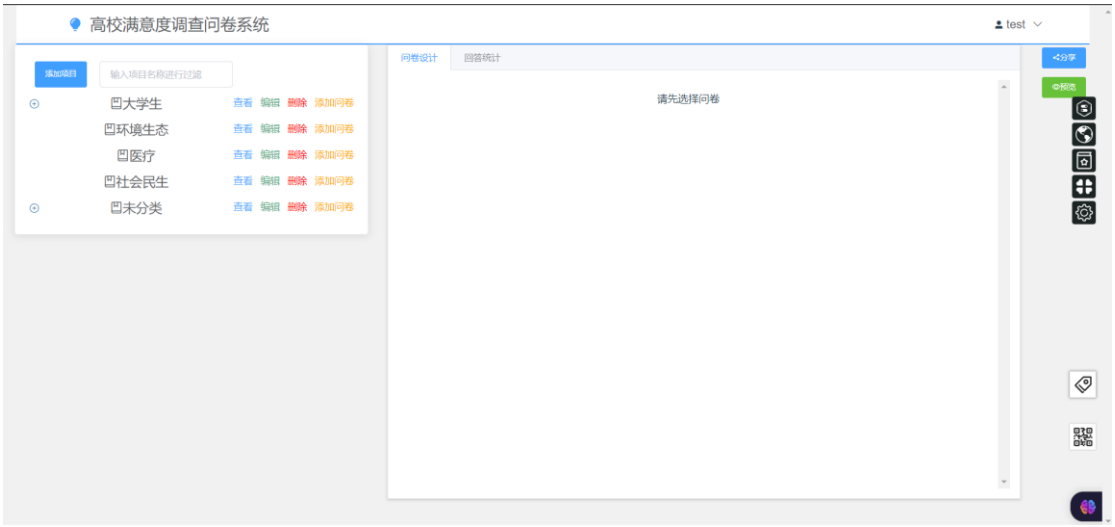


图 33

在项目界面可以看见下属问卷的情况：

图 34

添加项目：

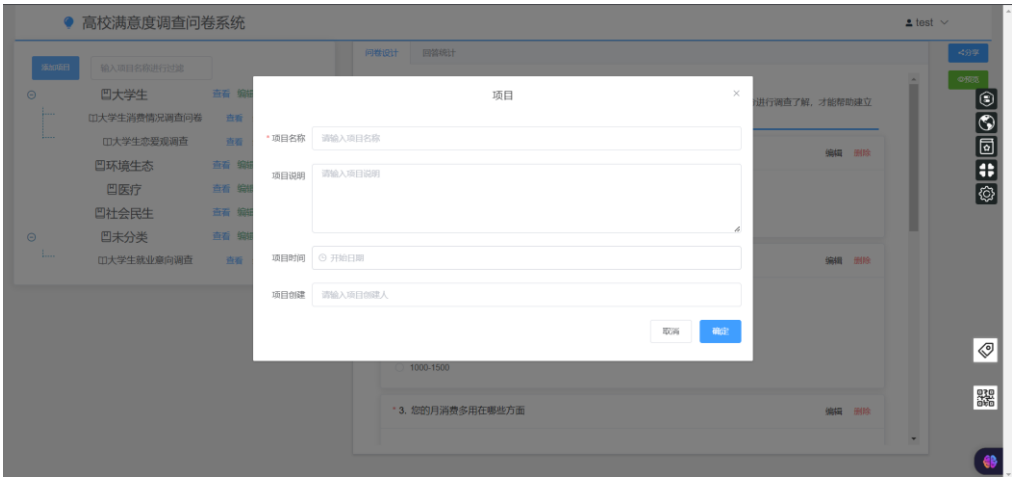


图 35

查看/修改项目：

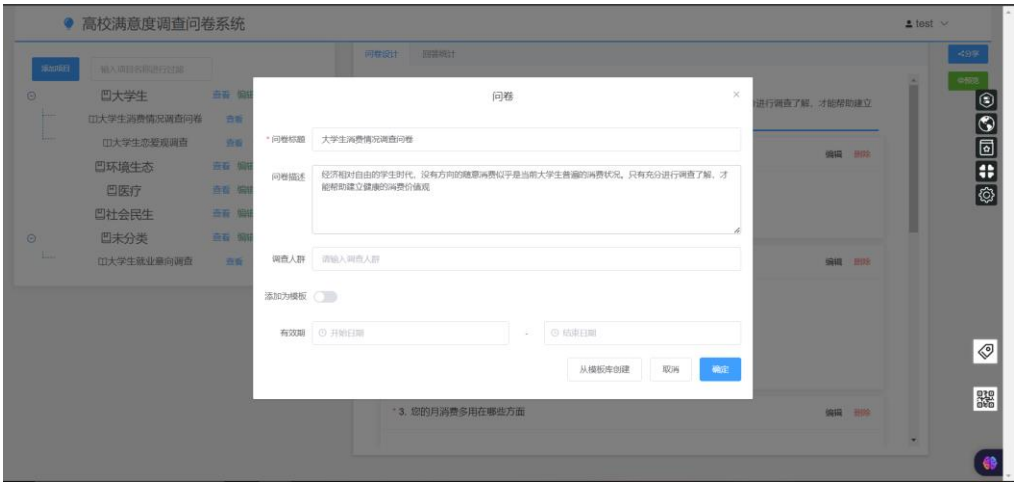


图 36

3. 创建问卷

➢ 项目管理入口的创建问卷：

图 37

添加题目：

图 38

➢ 问卷列表主界面创建（编辑）问卷：（点击删除即删除）

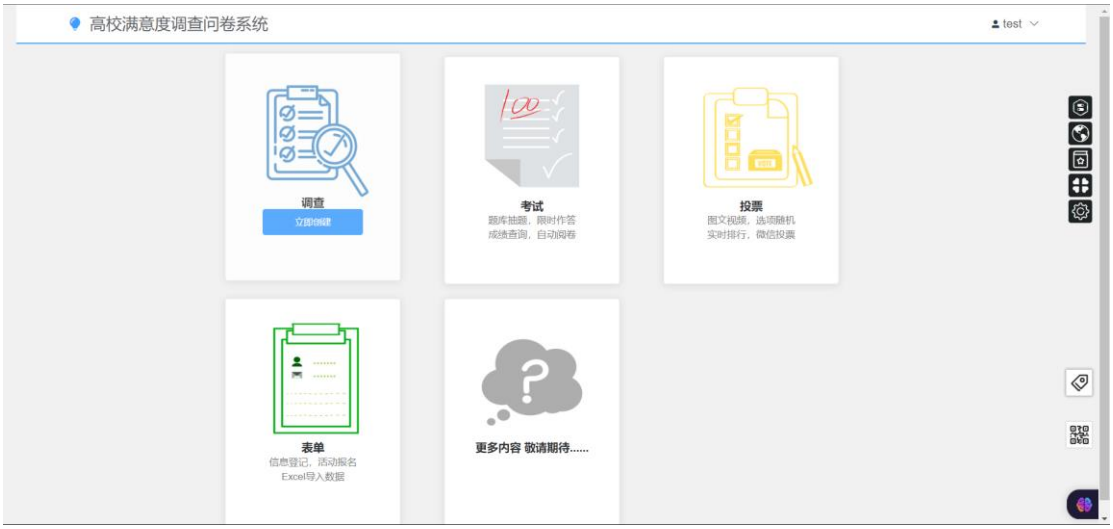


图 39

图 40

4. 查看问卷

预览：

图 41

用户填写情况：

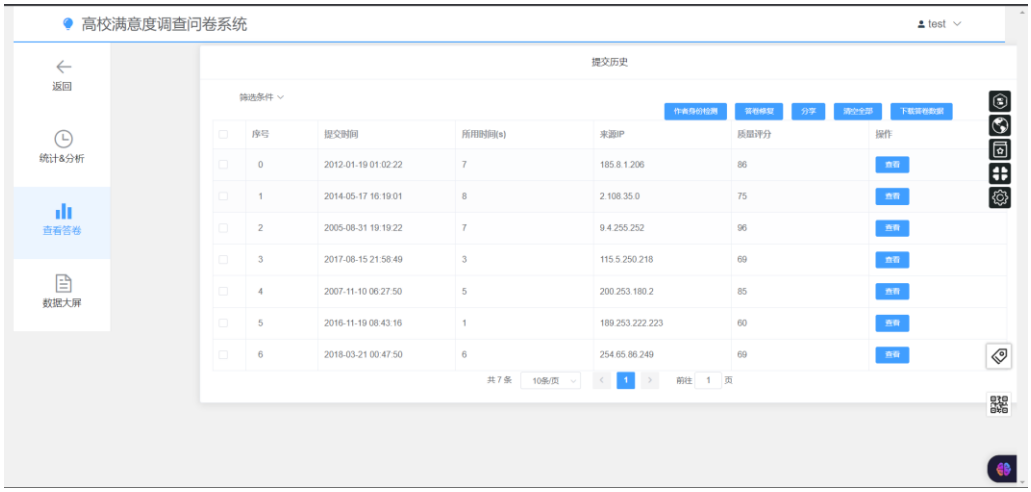


图 42

图 43

5. 发布问卷

（未发布不能分享）

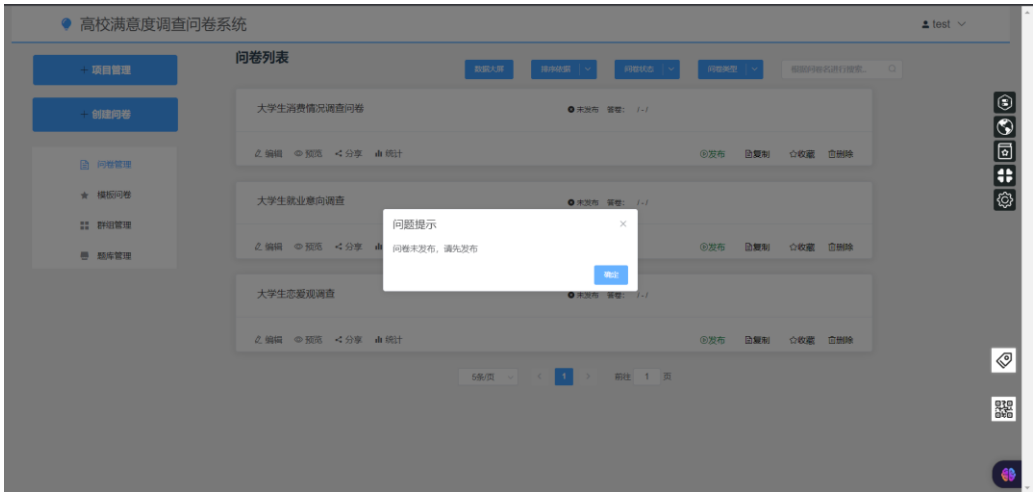


图 44

图 45

分享：

图 46

打开分享网址：

图 47

停止发布：

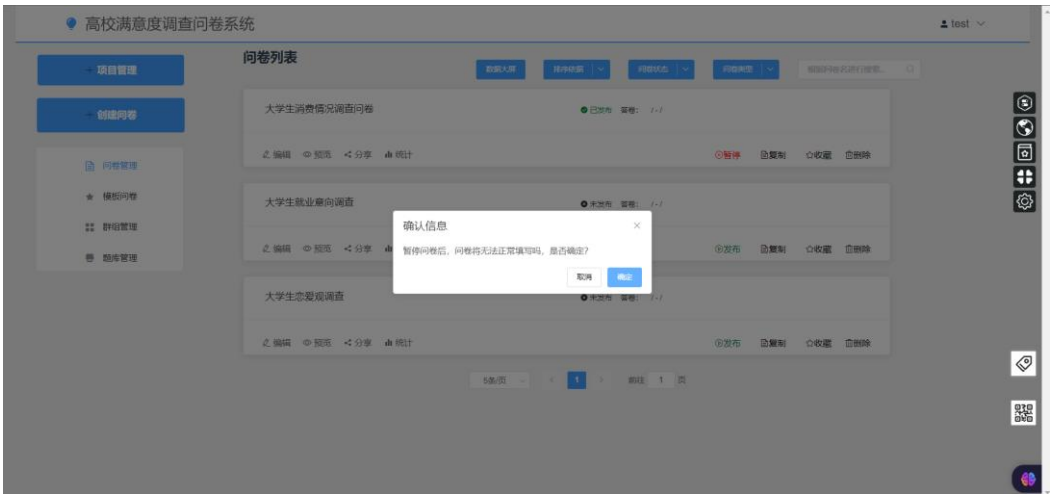


图 48

图 49

6. 数据统计

整体：

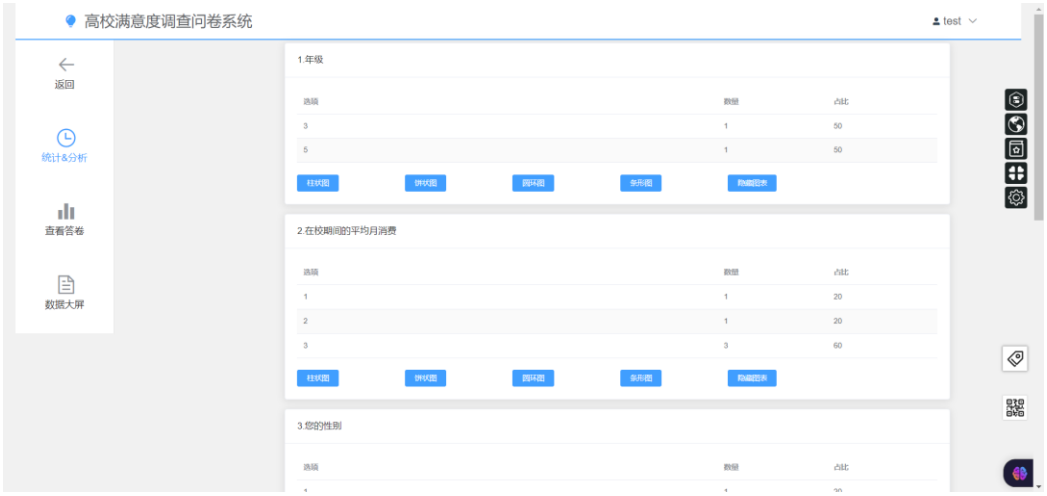


图 50

柱状图：

图 51

圆环图：

图 52

饼图：

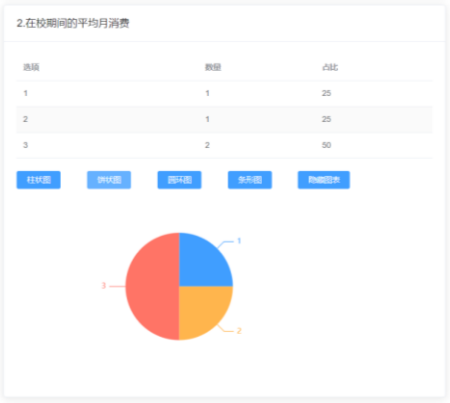


图 53

条形图：

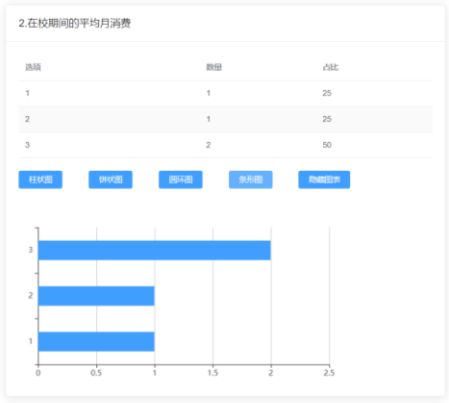


图 54

文字类回答详情：

图 55

词云：

图 56

7. 个性化组卷

创建问卷时可以实现人群，模板，有效期的添加：

图 57

问题设计时还可以实现次数限制：

图 58

8. 题库导入

题库管理：

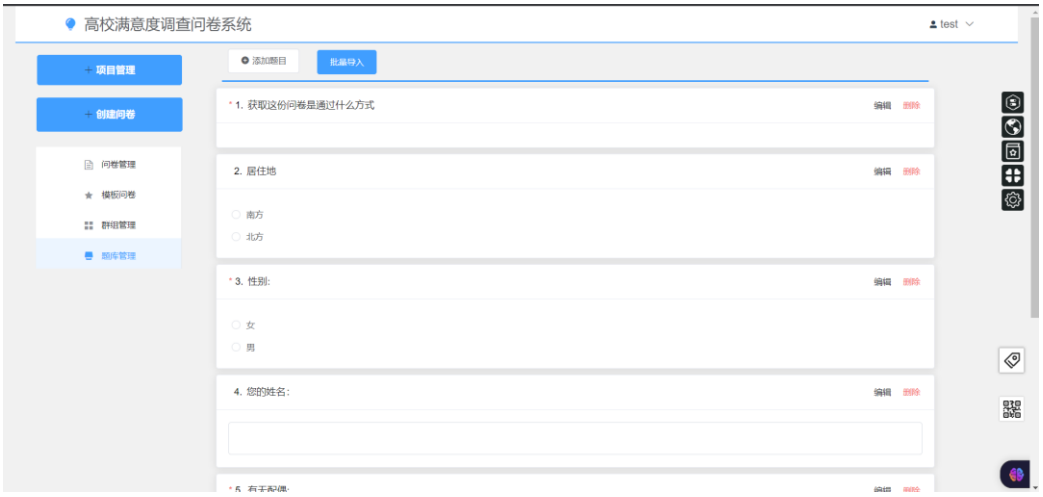


图 59

问卷中题库导入：

图 60

9. 用户填写

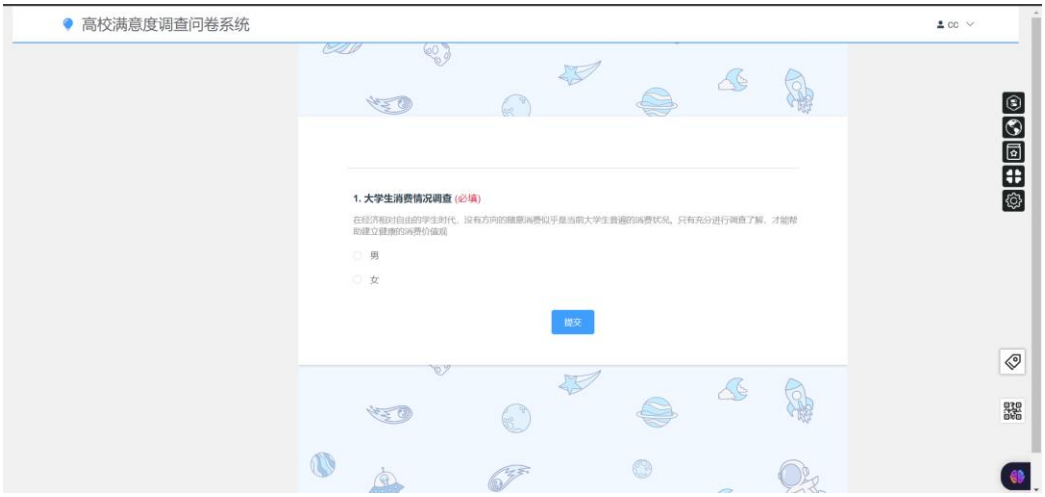


图 61

10. 用户管理

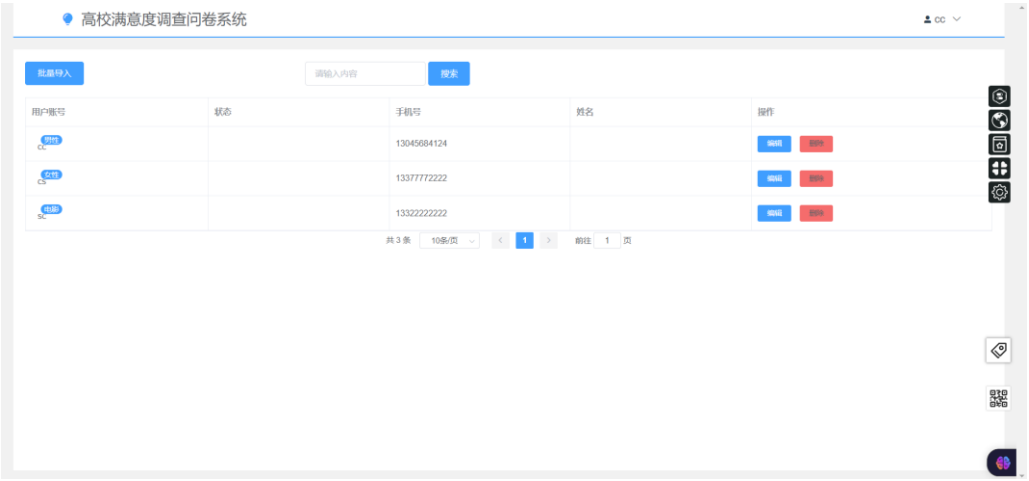


图 62

11. 模板管理

模板库：

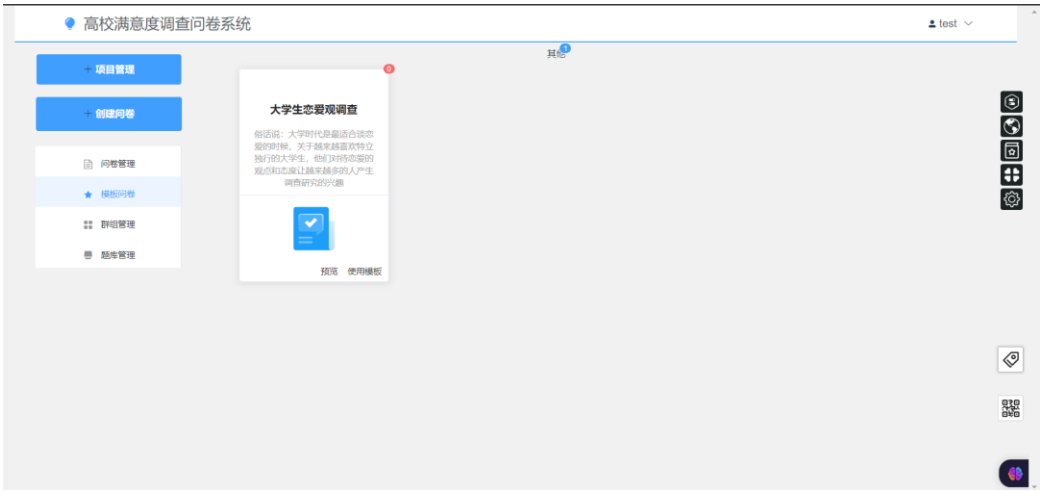


图 63

创建问卷时选择模板导入：

图 64

12. 群组管理

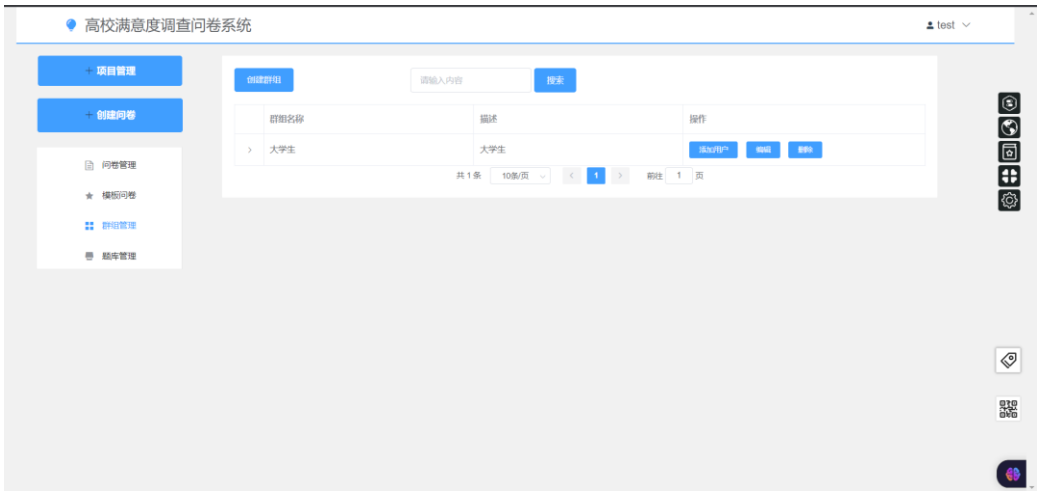


图 65

4.参考资料 References

[1] 《Java 编程思想》- Bruce Eckel 著 [M]. 机械工业出版社. 2004-01-01.
[2] 《Effective Java 中文版》- Joshua Bloch 著 [M]. 机械工业出版社.2009-06-01.
[3] 《Vue.js 权威指南》- 尤雨溪等著 [M]. 人民邮电出版社. 2018-06-01.
[4] 《Vue.js 实战》- 梁灏著 [M]. 人民邮电出版社. 2018-08-01.