

B/S 体系软件设计

调查问卷网站

设计文档

指导老师： 胡晓军

姓 名： 彭子帆

学 号： 3170105860

2020 年 4 月 30 日

目录

- 一、概述与需求分析3
 - 1.1、文档说明.....3
 - 1.2、产品功能模块.....3
 - 1.3、系统需求概述.....4
 - 1.3、系统设计基本原则5
- 二、技术框架说明6
 - 2.1 前端6
 - 2.2 后端7
 - 2.3 版本控制11
- 三、系统设计11
 - 3.1 用户类及特征：11
- 四、UML 图.....12
 - 4.1 用例图.....12
 - 4.2 时序图设计12
- 五、关键数据库设计14
 - 3.1 具体 table14
 - 3.2 表 user.....15
 - 3.3 表 paper15
 - 3.4 表 question.....16
 - 3.5 表 option.....16
 - 3.6 表 result.....16
 - 3.7 表 answer.....17
 - 3.8 数据库关系图.....17
- 六、数据字典18
 - 6.1 数据流定义表.....18
 - 6.2 数据元素定义表18
 - 6.3 数据精度表19

一、概述与需求分析

1.1、文档说明

本文档适用于本项目开发人员阅读，开发人员可据此进行开发。本设计文档主要采用了OOAD的方法，从类图和的角度对系统进行了分析设计。

1.2、产品功能模块

产品功能模块	可能涉及到的用户类
用户管理模块	游客、普通登录用户
问卷创建模块	普通登录用户
问卷作答模块	游客、普通登录用户
问卷统计结果模块	普通登录用户

1.2.1 用户管理模块

- 对于游客：
可以进行注册和登录操作。注册包含邮件、用户名以及密码，发送邮箱验证码进行验证。
- 对于普通登录用户：
可以进行退出登录操作。

1.2.2 问卷创建模块

- 对于游客：
无此功能权限
- 对于普通登录用户：
能够创建多个问卷，可设置问卷起始时间、问卷的模式以及问卷的标题，描述以及问卷的回答次数，每个问卷题目不限制，题目包含

- 单选
- 多选（不限制个数）
- 文本填写（支持单行、多行）
- 数字填写（支持定义数字类型：整数、小数）
- 评分（这是单选的特殊形式，界面表示有所不同）
- 获取地理位置

1.2.3 问卷作答模块

- **对于游客：**
根据问卷设置来进行作答权限限制，部分问卷可作答特定次数。
- **对于普通登录用户：**
根据问卷设置来进行作答权限限制，部分问卷可作答特定次数。

1.2.4 问卷统计结果模块

- **对于游客：**
无此功能权限
- **对于普通登录用户：**
能够查看多个创建的问卷，并显示每份问卷中每个填写的结果，结果包含该填写人的地点、填写时间、填写结果。以及每份问卷总的统计结果，如填写人数、起止时间等。

1.3、系统需求概述

1. 实现用户注册、登录功能，用户注册时需要填写必要的信息并验证，如用户名、密码要求在 6 字节以上，email 的格式验证，并保证用户名和 email 在系统中唯一。
2. 用户登录后可以设计问卷，一个问卷由标题、问卷说明和多个问卷项目组成，提供一个界面来动态设计，问卷项目需要最少需要实现以下内容
 - 单选
 - 多选
 - 文本填写（支持单行、多行）
 - 数字填写（支持定义数字类型：整数、小数）
 - 评分（这是单选的特殊形式，界面表示有所不同）

请您对此项目进行评价

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

非常不满意 非常满意

- 级联选择（对于单选的每个选择，可以动态显示其他一个或多个问卷项目）

1 是否在校

- ☐ 是
- ☐ 否

如用户选“是”，就显示一些项目，选“否”就显示另外一些项目

3. 问卷设计完成后生成填写链接，通过分享链接由他人填写，分享时可以设定填写周期和填写方式，填写方式支持以下几种

- 仅限注册用户
- 无需注册，可填写 n 次
- 无需注册，每天可填写 n 次

自行设计如何确定同一人。

4. 保存填写结果，设计一个界面展现填写结果，要求展示界面直观清晰，有一定的统计量（如填报人数、起止时间等），对于数字类型的问卷项目支持汇总计算。

增强功能:

5. 样式适配手机端，能够在手机浏览器/微信等应用内置的浏览器中友好显示，支持额外的问卷项目

- ## ● 地理位置

1.3、系统设计基本原则

- **整体性原则:** 系统开发计划应符合应用环境本身目标和信息系统建设目标的要求,从系统开发的总体规划出发,注意与系统运行环境的其他计划相互联系、协调一致。
- **阶段性原则:** 信息系统开发是一个复杂的系统工程,是不可能一步完成的,要考虑人力、物力、财力等方面的约束,分阶段进行,使每个阶段的工作作为下一阶段的前提和基础。每个阶段应有明确可测的目标、精确的起止时间、详细具体的工作任务和可以审查的阶段成果,以便于进度控制和质量保证。
- **客观可行原则:** 制定开发计划要符合客观需要,切忌脱离实际的花架子。计划的内容应周到细致,切实可行,在人力、财力、物力等方面有切实可靠的保证。要注意避免超过实际需要的技术先进性,采用成熟的技术与方法。

- 全面性原则：计划的制定不仅仅由系统开发人员进行，应尽可能请业务、技术、行政管理等方面的人员和领导参加，这样有助于从不同角度、不同层次来全面考虑整个计划，并且可以促进计划执行过程中与有关方面的协调和配合。

二、技术框架说明

2.1 前端

2.1.1 Vue

Vue 是一个用于创建用户界面的开源框架，也是一个创建单页应用的 Web 应用框架，一套用于构建用户界面的渐进式框架。

Vue 拥有很多利于开发者，以及提高网站性能的特性。其有以下特性：

1. 响应式数据绑定

当数据发生变化时，Vue 自动更新视图。它的原理，是利用了 `Object.defineProperty` 中的 `setter/getter` 代理数据，监控对数据的操作。

2. 组合的视图组件

ui 页面映射为组件树，划分组件使得可维护、可重用、可测试。

3. 虚拟 DOM

JS 的运行速度在需要大量的 DOM 操作时会变得很慢，但是前端本身就是要通过 JS 处理 DOM 来更新视图数据的。这样在更新数据后会重新渲染页面，就造成在没有改变数据的地方也重新渲染了 DOM 节点，因此性能方面就很受影响。Vue 则利用在内存中生成与真实 DOM 对应的数据结构，虚拟 DOM，使得当数据发生变化时，能够智能地计算出重新渲染组件的最小代价，并应用到 DOM 操作上。

4. MVVM 模式

MVVM 模式中，M 即 Model，数据模型；V 即 view，视图模板；vm 即 view-Model，视图模型。ViewModel 作为层，它就像是一个中转站，负责转换 Model 中的数据对象来让数据变得更容易管理和使用。该层向上与视图层进行双向数据绑定，向下与 Model 层通过接口请求进行数据交互，起承上启下作用。View 层展现的不是 Model 层的数据，而是 ViewModel 的数据。由 ViewModel 负责与 Model 层交互，这就完全解耦了 View 层和 Model 层。这个解耦是至关重要的，它是前后端分离方案实施的最重要一环。

5. 声明式渲染

Vue 采用简洁的模板语法来声明式的将数据渲染进 DOM，初始化根实例，自动将数据绑定在 DOM 模板 上。因此，使用 Vue，只需要关心声明在哪里，做什么，而无需关心如何实现。

优势：

1. Vue 与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。
2. Vue 使用基于依赖追踪的观察并且使用异步队列更新。所有的数据都是独立触发的。因此，无需对数据做脏检查，加快响应速度。
3. Vue 是单页面应用，使页面局部刷新，不用每次跳转页面都要请求所有数据和 dom，大大加快了访问速度和提升用户体验。
4. 将注意力集中保持在核心库，有丰富的插件库。

2.1.2 Element-UI

Element UI 是一套采用 Vue 2.0 作为基础框架实现的组件库，能够快速地搭建网站，极大地减少研发的人力与时间成本。网站搭建过程中可以选用 Element UI 组件库中的组件，提高开发效率，也能够使网站给用户带来更好的体验。

2.2 后端

2.2.1 Spring Boot

Spring Boot 是由 Pivotal 团队提供的全新框架，其设计目的是用来简化新 Spring 应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，Spring Boot 致力于在蓬勃发展的快速应用开发领域(rapid application development)成为领导者。

SpringBoot 所具备的特征有：

- (1) 可以创建独立的 Spring 应用程序，并且基于其 Maven 或 Gradle 插件，可以创建可执行的 JARs 和 WARs；
- (2) 内嵌 Tomcat 或 Jetty 等 Servlet 容器；
- (3) 提供自动配置的“starter”项目对象模型（POMS）以简化 Maven 配置；
- (4) 尽可能自动配置 Spring 容器；
- (5) 提供准备好的特性，如指标、健康检查和外部化配置；
- (6) 绝对没有代码生成，不需要 XML 配置。

2.2.2 MySQL

- **简介：**

MySQL 是一种开放源代码的关系型数据库管理系统 (RDBMS)，使用最常用的数据库管理语言——结构化查询语言 (SQL) 进行数据库管理。MySQL 是一种关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

- **技术用途：**

MySQL 因为其速度、可靠性和适应性而备受关注。大多数人都认为在不需要事务化处理的情况下，MySQL 是管理内容最好的选择。MySQL 使用 C 和 C++ 编写，并使用了多种编译器进行测试，保证了源代码的可移植性。其还支持 AIX、FreeBSD、HP-UX、Linux、Mac OS、NovellNetware、OpenBSD、OS/2 Wrap、Solaris、Windows 等多种操作系统。MySQL 为多种编程语言提供了 API。这些编程语言包括 C、C++、Python、Java、Perl、PHP、Eiffel、Ruby、.NET 和 Tcl 等。MySQL 在应用上的一大优点是支持多线程，充分利用了 CPU 资源。最后，MySQL 还支持大型的数据库，可以处理拥有上千万条记录的大型数据库。

- **选择理由：**

MySQL 是一款非常可用的开源数据库。我们作为计算机系的学生非常熟悉其使用的 SQL 语言，而 MySQL 数据库在使用 SQL 语言的数据库当中算得上是性能相对较好的一款。相比于较为轻量级的 SQLite 数据库，它支持基本上是全部的 SQL 指令，并且有着极高的性能，可以被用于处理大量数据上。另外，MySQL 也比较方便部署，能够简化我们的操作流程。

2.2.3 Redis

- **简介：**

Redis (Remote Dictionary Server)，中文翻译为远程字典服务，是一个开源的使用 ANSI C 语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value 数据，并提供多种语言的 API。Redis 是一个 key-value 存储系统。和 Memcached 类似，它支持存储的类型更多，包括字符串、链表、集合、有序集合和哈希类型。这些数据类型都支持 push/pop、add/remov 及取交集并集和差集及更丰富的操作，而且这些操作都是原子性的。在此基础上，Redis 支持各不同方式的排序。与 memcached 一样，为了保证效率，数据都是缓存在内存中。区别的是 Redis 会周性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了主从同步。

- **用途:**

Redis 是一个高性能的 key-value 数据库。它提供了 Java, C/C++, C#, PHP, JavaScript, Perl, Object-C, Python, Ruby, Erlang 等客户端, 使用很方便。Redis 还支持主从同步。数据可以从主服务器向任意数量的从服务器上同步, 从服务器可以是关联其他从服务器的主服务器。这使得 Redis 可执行单层树复制。存盘以有意无意的对数据进行写操作。由于完全实现了发布/订阅机制, 使得从数据库在任何地方同步树, 可订阅一个频道并接收主服务器完整的消息发布记录。同步对读取操作的可扩展性和数据冗余很有助。

- **选用理由:**

选用 Redis 的原因是我们需要一种 key-value 类型的数据库, 故采取了这种非 sql 语言的数据库, 来弥补使用的 MySQL 数据库的不足。我认为 Redis 高性能特点可以极好地帮助注册登录以及进行 token 的定时以及储存。

2.2.4 MyBatis

- **简介:**

Mybatis 是一款优秀的持久层框架, 它支持自定义 SQL、存储过程以及高级映射。Mybatis 免除了几乎所有的 JDBC 代码以及设置从参数和获取结果集的工作。Mybatis 可以通过简单的 XML 或注解来配置和映射原始类型、接口和 JAVA POJO(Plain Old Java Objects, 普通老式 Java 对象)为数据库中的记录。

- **用途:**

Mybatis 是众多 orm 框架的一种, 常见的 orm 框架还有 jpa、hibernate。当使用面向对象的编程语言来进行应用程序开发时, 从项目一开始就采用的是面向对象分析、面向对象设计、面向对象编程, 但是在访问数据库时, 又要重返关系数据库的访问方式, 即编写 sql 语句。orm 工具的作用便是将关系数据库包装成面向对象模型。只要依然采用面向对象的编程语言, 底层依然采用关系型数据库, 中间就少不了 Mybatis。采用 Mybatis 之后, 程序就不再直接访问底层数据库, 而是以面向对象的方式来操作持久化对象(例如增删改查操作), Mybatis 会将这些操作转换成底层的 sql 操作

2.2.5 Docker

- **简介：**

Docker 属于 Linux 容器的一种封装，提供简单易用的容器使用接口。它是目前最流行的 Linux 容器解决方案。Docker 将应用程序与该程序的依赖，打包在一个文件里面。运行这个文件，就会生成一个虚拟容器。程序在这个虚拟容器里运行，就好像在真实的物理机上运行一样。有了 Docker，就不用担心环境问题。总体来说，Docker 的接口相当简单，用户可以方便地创建和使用容器，把自己的应用放入容器。容器还可以进行版本管理、复制、分享、修改，就像管理普通的代码一样。

- **用途：**

Docker 的主要用途，目前有三大类。

1. **提供一次性的环境。**比如，本地测试他人的软件、持续集成的时候提供单元测试和构建的环境。
2. **提供弹性的云服务。**因为 Docker 容器可以随开随关，很适合动态扩容和缩容。
3. **组建微服务架构。**通过多个容器，一台机器可以跑多个服务，因此在本机就可以模拟出微服务架构。

2.2.6 Nginx

- **简介：**

Nginx 同 Apache 一样都是一种 Web 服务器。基于 REST 架构风格，以统一资源描述符（Uniform Resources Identifier）URI 或者统一资源定位符（Uniform Resources Locator）URL 作为沟通依据，通过 HTTP 协议提供各种网络服务。Apache 的特点使其不可能成为高性能 Web 服务器，轻量级高并发服务器 Nginx 就应运而生了。

- **用途：**

Nginx 是一款自由的、开源的、高性能的 HTTP 服务器和反向代理服务器；同时也是一个 IMAP、POP3、SMTP 代理服务器。Nginx 可以作为一个 HTTP 服务器进行网站的发布处理，另外 Nginx 可以作为反向代理进行负载均衡的实现。

2.3 版本控制

GitHub 是当前最受欢迎的版本控制工具。GitHub 是一个面向开源及私有软件项目的托管平台，因为只支持 git 作为唯一的版本库格式进行托管，故名 GitHub。GitHub 于 2008 年 4 月 10 日正式上线，除了 Git 代码仓库托管及基本的 Web 管理界面以外，还提供了订阅、讨论组、文本渲染、在线文件编辑器、协作图谱（报表）、代码片段分享（Gist）等功能。目前，其注册用户已经超过 350 万，托管版本数量也是非常之多，其中不乏知名开源项目 Ruby on Rails、jQuery、python 等。

版本控制能够让我们开发更加具有效率，出现错误操作可以回滚等等。

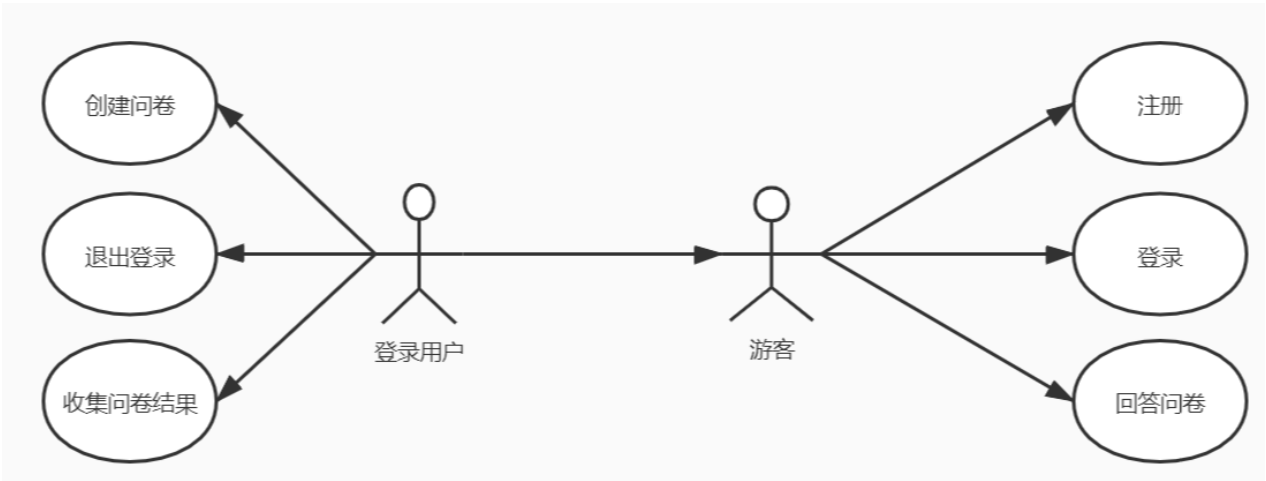
三、系统设计

3.1 用户类及特征：

用户类	特征
普通登录用户	1.主要用户。 2.需要能够方便创建问卷、收集问卷。 3.能够随时跟踪问卷的情况。
游客	1.次要用户。 2.功能基本是仅有填写问卷。

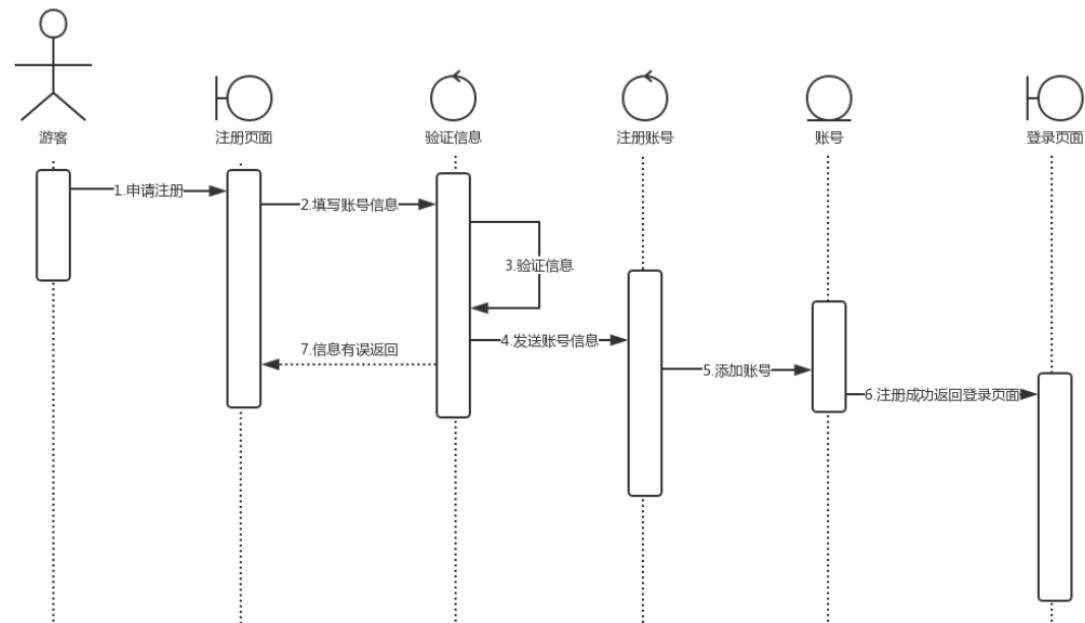
四、UML 图

4.1 用例图

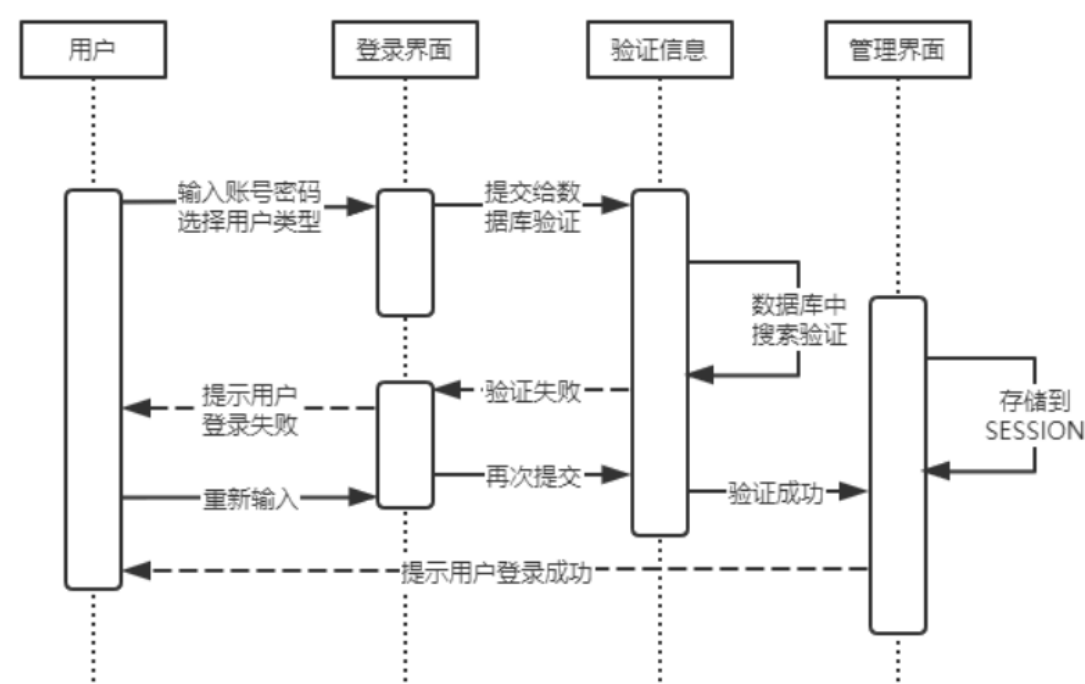


4.2 时序图设计

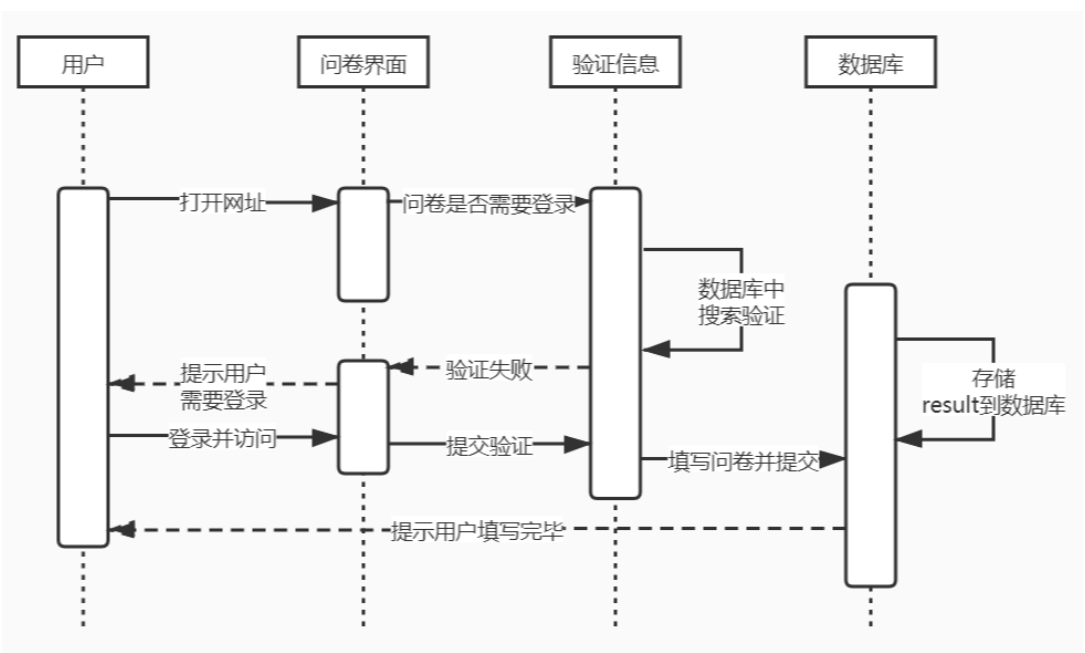
4.2.1 游客注册账号



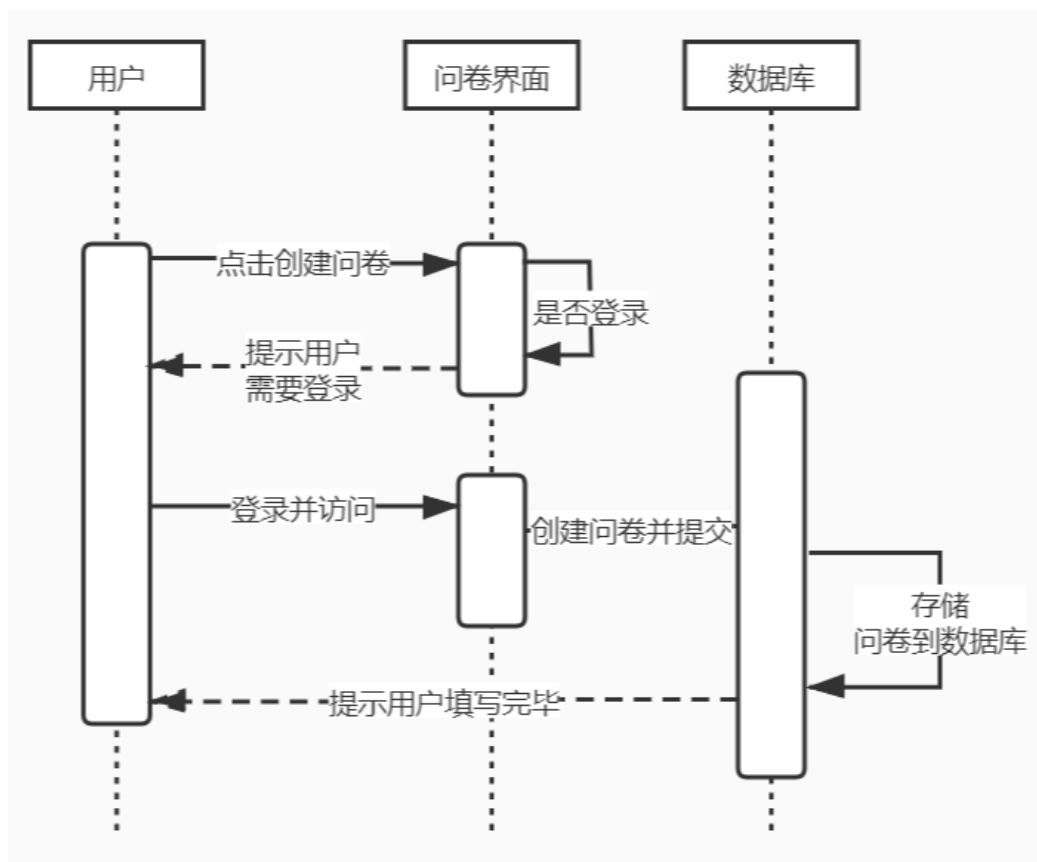
4.2.2 登录账号



4.2.3 填写问卷时序图



4.2.4 创建问卷时序图



五、关键数据库设计

3.1 具体 table

表名	功能
user	用户
paper	问卷信息
question	问题信息（仅包含问题 id、内容、类型以及父问题 id）
option	每个选项的信息
result	问卷回答信息
answer	题目回答信息

3.2 表 user

列名	数据类型	Not Null (Y/N)	约束	Other
email	varchar(50)	Y	Primary key	Unique
username	varchar(20)	Y		Unique
password	varchar(20)	Y		

3.3 表 paper

列名	数据类型	Not Null (Y/N)	约束	Other
paperID	int	Y	Primary key	Auto increment Unsigned
author	varchar(20)	Y	Foreign key	
title	varchar(50)	Y		
description	varchar(500)	N		
mode	tinyint	Y	default 0	Unsigned
fillNumber	int	N		Unsigned
open	tinyint	Y	default 1	

3.4 表 question

列名	数据类型	Not Null (Y/N)	约束	Other
questionID	int	Y	Primary key	Auto increment, Unsigned
paperID	int	Y	Foreign key	Unsigned
type	tinyint	Y		

3.5 表 option

列名	数据类型	Not Null (Y/N)	约束	Other
optionID	int	Y	Primary key	Auto increment, Unsigned
questionID	int	Y	Foreign key	Unsigned
content	varchar(100)	Y		

3.6 表 result

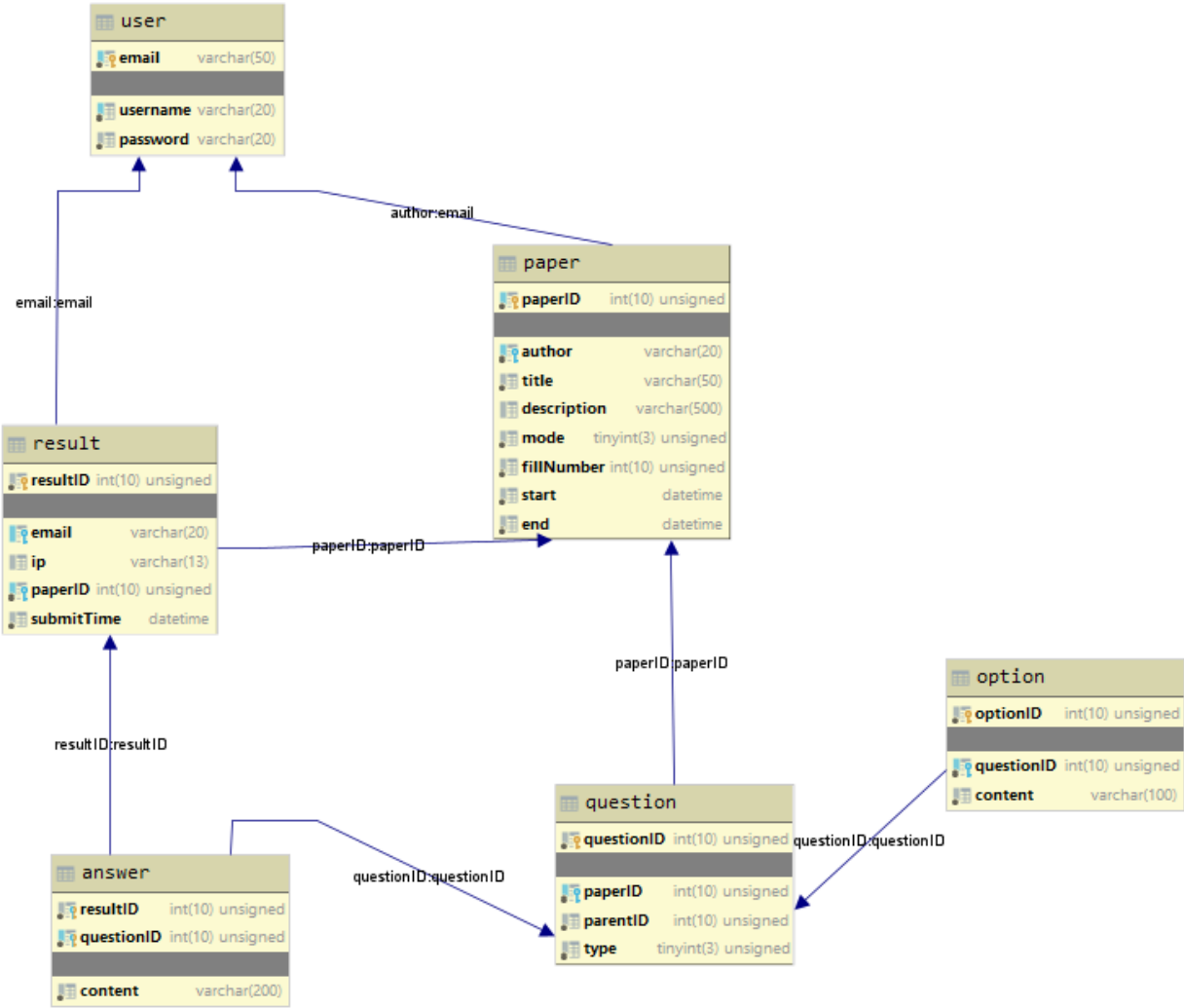
列名	数据类型	Not Null (Y/N)	约束	Other
resultID	int	Y	Primary key	Auto increment, Unsigned
email	varchar(20)	N	Foreign key	
ip	varchar(13)	N		

paperID	int	Y	Foreign key	
submitTime	datetime	Y		

3.7 表 answer

列名	数据类型	Not Null (Y/N)	约束	Other
resultID	int	Y	Primary key	Unsigned
questionID	int	Y		Unsigned
content	varchar(200)	Y		

3.8 数据库关系图



六、数据字典

6.1 数据流定义表

编号	数据流名	来源	去向	组成	说明
L1	创建问卷	用户	系统	E1+E2+E3+	创建一个问卷以及相应问题
L2	统计结果	系统	用户	E4+E6+E8+E9+E10	返回统计结果
L3	登录	用户	系统	E1+E3	登录
L4	注册	用户	系统	E1+E2+E3	注册
L5	退出登录	用户	系统	E1	退出登录
L6	填写问卷	用户	系统	E11+E12+E13+E14+E15+E16+E17+E18	填写问卷并储存在数据库中

6.2 数据元素定义表

编号	数据元素名	类型	值域	说明
E1	用户 Email	字符	"U"+"0000000".. "9999999"	自增
E2	用户名	字符		2-20 位字符
E3	用户密码	字符		6-20 位字符
E4	问卷编号	数字	0...	自增
E5	问卷名	文件		1-50 位字符
E6	问卷描述	字符		0-500 位字符
E7	问卷模式	数字	0...2	
E8	填写人数	数字	0...	
E9	开始时间	时间	YYYY-MM-DD hh:mm:ss	
E10	结束时间	时间	YYYY-MM-DD hh:mm:ss	
E11	题目编号	数字	0...	自增

E12	题目类型	数字	0...6	0-6 分别对应单选、多选、文本、数字整数、数字小数、评分、地理位置
E13	选项编号	数字	0...	自增
E14	选项内容	字符		1-100 位字符
E15	结果编号	数字	0...	自增
E16	ip 地址	字符	"0.0.0.0"... "255.255.255.255"	
E17	提交时间	时间	YYYY-MM-DD hh:mm:ss	
E18	回答内容	字符		0-200 位字符

6.3 数据精度表

数据	类型	精度要求	说明	示例
用户 Email	字符	"U"	自增	3170105860@z.ju.edu.cn
用户名	字符	2-20 位字符	2-20 位字符	abc
用户密码	字符	6-20 位字符	6-20 位字符	root123
问卷编号	数字	0...	自增	3
问卷名	文件	1-50 位字符	1-50 位字符	
问卷描述	字符	0-500 位字符	0-500 位字符	
问卷模式	数字			2
填写人数	数字			100
开始时间	时间			2020-04-29 21:50:50
结束时间	时间			2020-04-29 21:50:50
题目编号	数字		自增	2
题目类型	数字		0-6	
选项编号	数字		自增	1

选项内容	字符	1-100 位字符	1-100 位字符	
结果编号	数字		自增	5
ip 地址	字符			192.168.1.1
提交时间	时间			2020-04-29 21:50:50
回答内容	字符	0-200 位字符	0-200 位字符	“123”