

# 浙江大学

## 本科实验报告

ButianSurvey 不填问卷 调查问卷平台

## 实验报告

课程名称: BS 体系软件设计

姓 名: 黄洋逸

学 院: 竺可桢学院

系: 竺可桢学院

专 业: 混合班

计算机科学与技术

指导教师: 胡晓军

2020 年 4 月 23 日

# 浙江大学实验报告

课程名称: BS 体系软件设计 实验类型: 综合型

实验项目名称: ButianSurvey 不填问卷 调查问卷平台

学生姓名: 黄洋逸 专业: 计算机科学与技术 学号: 3170104172

同组学生姓名: 单人实验 指导老师: 胡晓军

实验地点: \_\_\_\_\_ 实验日期: 2020 年 6 月 22 日

## 目录

一、	设计报告部分 .....	4
0	项目概述 .....	4
1	需求分析与功能实现设计 .....	4
1.1	基本功能需求分析 .....	4
1.2	特色功能 .....	5
2	技术框架 .....	7
2.1	项目框架图 .....	7
2.2	前端技术 .....	7
2.3.1	前端代码结构 .....	8
2.3	后端技术 .....	8
2.4.1	后端代码结构 .....	9
3	数据库设计 .....	10
3.1	用户数据 .....	10
3.2	问卷数据 .....	11
3.4	SpringBoot MongoDB 中数据类型具体实现 .....	14
4	UI 设计 .....	15
4.1	用户问卷管理（控制台界面） .....	16
4.2	问卷编辑界面 .....	17
4.3	问卷填写界面 .....	18
4.4	问卷数据界面 .....	18

5 后端 API 设计 .....	18
二、 使用手册部分 .....	19
0 简介 .....	19
1 运行与部署 .....	19
2 登录与注册 .....	19
3 管理现有的问卷 .....	21
4 问卷编辑 .....	21
5 设置问卷发布 .....	26
6 问卷填写 .....	27
7 查看问卷结果 .....	28
三、测试报告部分 .....	29
0 登录注册功能测试 .....	29
1 主页功能测试 .....	31
2 问卷设计功能测试 .....	31
3 问卷预览功能测试 .....	37
4 问卷设置测试 .....	39
5 问卷填写功能测试 .....	40
6. 问卷结果界面测试 .....	42
7. 移动端适配测试 .....	42
四、开发感想&小结 .....	44

# 一、设计报告部分

## 0 项目概述

本部分为 2020 年《B/S 体系软件设计》课程项目的设计文档，目的是通过初步确定项目所使用的技术框架、针对需求点的功能模块设计、数据模型和重要界面原型图示等要素，为接下来课程项目的进一步开发作计划。

本项目暂命名为“**ButianSurvey**”、“不填问卷”，是一个调查问卷 web 应用。项目的目标是构建起一个对问卷发放用户和填写用户都能提供愉快的交互体验、拥有完善的问卷样式功能、贴近实际使用情景且对数据分析友好的调查问卷平台。

本部分在项目阶段性完成之后根据实际实现情况进行了部分修改。

## 1 需求分析与功能实现设计

### 1.1 基本功能需求分析

根据课程实验的相关要求，本项目将会实现以下的基本功能：

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。（已实现）

*对于该需求，我们可以用加盐 hash 的形式对用户密码进行验证；用户注册过程中的密码要求可以在前端交互逻辑中实现，用户名和 email 的唯一性交由后端 API 实现，在数据库中查重；用户在使用网站时的登录状态保持可以由 token 以及 cookie 等多种方式实现。*

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（已实现）

- 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）
- 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)

如用户选“是”，就显示一些项目，选“否”就显示另外一些项目

在前端部分，我们可以先对每一种题型预先实现好对应的组件，在前端获取问卷数据时，就可以按数据对页面进行生成；相关的一些小组件使用现在流行的一些组件库就可以轻松实现，只要对样式再进行一些调整即可；在后端，我们需要针对问卷和题型的数据格式对数据库进行设计，如果使用SQL类的关系型数据库，数据库设计会较为繁琐；所以这里我们选用文档型数据库MongoDB。

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（已实现）

- 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。

这部分功能的实现，我们只需要结合用户登录状态的维护以及问卷权限的记录信息即可；体现的主要是一个权限机制；对于无需注册的同一用户确定，我们可以通过 cookie 和 IP 等方式来判断是否为同一用户。

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。（已实现）

考虑到问卷调查的结果可能数据量会很大，如果交由前端处理可能会受限于浏览器的性能；这里可以使用先由后端对数据进行统计再将结果通过 API 传递到前端；所以前端只需要应用所需的可视化组件即可。

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

现代的组件库一般都对移动端有较好的支持（但其实 AntDesign 不提供官方的移动端支持），我们在前端中识别出当前是移动端还是桌面端，对一些组件进行调整即可（最终我采取了在移动端和桌面端进行统一调整的方式来实现）；地理位置也可以通过对应的第三方 API 实现。

## 1.2 特色功能

除了以上所述的基础功能以外，本项目还计划添加以下的拓展功能作为项目特色；但由于完成课程设计时间所限，该部分还处于尚未完成状态；此处作为项目原设计计划的一部分加以展示：

1. 更加多样有趣的问卷交互形式，如：
  - i. 多样化的评分样式，如图标，拖动条等；（已实现）  
*可以通过多样化的前端组件实现；*
  - ii. “表格”形式的问题，将选项相同，意义相近的单选题合并在一个表格中，填写体验更加良好（未实现）  
*对于此类问题，我们可以设计专门的前端组件以及数据库格式；*
  - iii. 多种问卷展示模式，如：卡片、栏目、长问卷等（未实现）  
*只要在设计问卷时对展示模式进行预先规定，将相关信息存储在数据库里；前端就可以按照对应的模板将问卷展示给用户。*
2. 对于问卷使用体验的优化，如：
  - i. 对于重复填写的问卷，自动保存上次提交的内容，或提供用户预填写默认信息的功能；（未实现）  
*考虑到用户网络的不稳定性与使用习惯的不规律性，将这些数据交给 Browser 端数据处理更为合适；我们可以利用 Cookie 机制来实现这一需求。*
  - ii. 问卷填写进度的实时显示（未实现）  
*在前端界面添加人性化的浮动进度条等组件即可。*
  - iii. 未完成问题的提示与跳转  
*同上，在前端组件上实现。*
3. 可用的导入、导出功能（未实现）
  - i. 问卷设计的导入导出
  - ii. 问卷结果的导入导出*这些需求都需要我们在数据库设计上有良好的实践；利用 MongoDB 进行问卷信息的存储，我们可以利用 BSON 在文档结构上的优势将其直接导出为 BSON、JSON 或 XML 等格式的数据，对我们进一步的数据处理非常便捷；也可以考虑增加对 CSV、Excel 等表格型数据格式的支持。*
4. 问卷编辑的高级功能（未实现）
  - i. 问卷信息栏目中支持 Markdown 格式  
*引入相应的 Markdown 渲染组件即可。*
  - ii. 允许使用类似标记语言的形式设计问卷  
*该需求的实现具有一定的难度，需要设计对应的标记语言标准，并编写 parser 进行处理；但我认为如果能够实现将会成为一个非常有意义的 feature。*

iii. 支持复杂的级联条件逻辑表达式

这里对应的是级联选择需求的拓展，如果我们能够以脚本语言等形式来表示题目之间的逻辑依赖关系，具有一定相关技能的用户将能够设计出功能更为灵活强大的问卷。

5. 问卷结果数据可视化功能

我们可以针对常见的几种题型，或者填写者的信息进行简单统计，进行对应的可视化；这方面有许多成熟的前端可视化组件可以应用。

## 2 技术框架

### 2.1 项目框架图

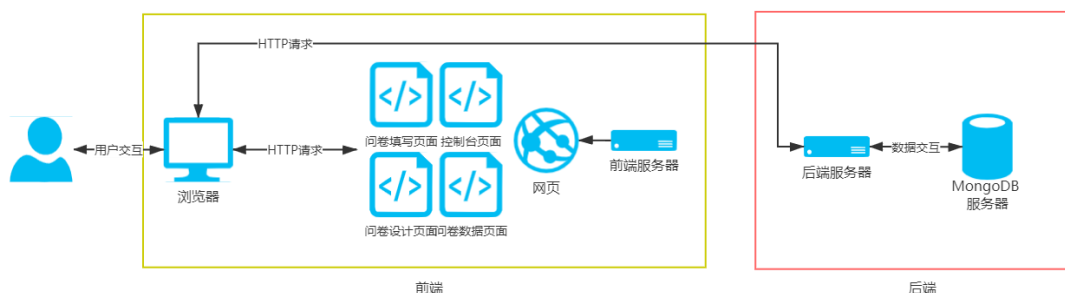


Figure 1 项目框架图

由于课程的要求，我们首先要确认的是整体上的 Browser/Server 架构；更具体而言，为了方便前端页面和后端数据处理的开发，本项目采取了更加现代的前后端分离的开发模式。前后端之间的数据交互由页面调用后端接口的方式进行，本项目的目标是尽量采用符合 RESTful 风格的 API 进行前后端的数据交互。

### 2.2 前端技术

本项目暂定使用 [Vue.js](#) 作为前端框架。Vue 是一套用于构建用户界面的渐进式 JavaScript 框架。Vue 的最大特点为其核心库主要关心视图层（view），被设计为可以自底向上逐层应用；其以基于 HTML 的模板语法作为主要使用方式的特点也使得该框架对于了解 HTML, CSS, JavaScript 的开发者而言更加易于上手；这也是本项目使用 Vue 框架的主要原因，希望能够将更多的精力放在整个 B/S 体系的软件设计与功能实现的层面。

在 UI 组件库方面，本项目将基于 Ant Design 与 Element 两个组件库进行编写：前者发展较为成熟，组件丰富且颇受开发者欢迎；后者对 Vue 具有较好的支持。如果要对两个组件库进行混用，本项目将特别注意对设计语言进行统一。

对于具体的前端开发语言，暂定以 TypeScript 作为主要开发语言。相较于 JavaScript 这一弱类型语言，TypeScript 拥有着强大的静态类型系统，在避免运行

错误，提升编写体验的方面具有显著的优势。

### 2.3.1 前端代码结构

在具体实践中，通过编写组件和页面，规定路由的方式，我实现了设计中的前端逻辑。下面简单地列出前端部分的代码结构：

```
bs-frontend/src/
├── App.vue
├── assets
│   ├── butian.png
│   ├── butiansurvey.png
│   ├── logo.png
│   └── my-assets.css
├── components
│   ├── MySurveys.vue
│   ├── ProblemDesign.vue
│   └── ProblemView.vue
├── main.ts
├── module.d.ts
├── router
│   └── index.ts
├── shims-tsx.d.ts
├── shims-vue.d.ts
├── store
│   └── index.ts
├── utils
│   └── helpers.ts
└── views
    ├── About.vue
    ├── Home.vue
    ├── Login.vue
    ├── Register.vue
    ├── SurveyEdit.vue
    └── SurveyView.vue
```

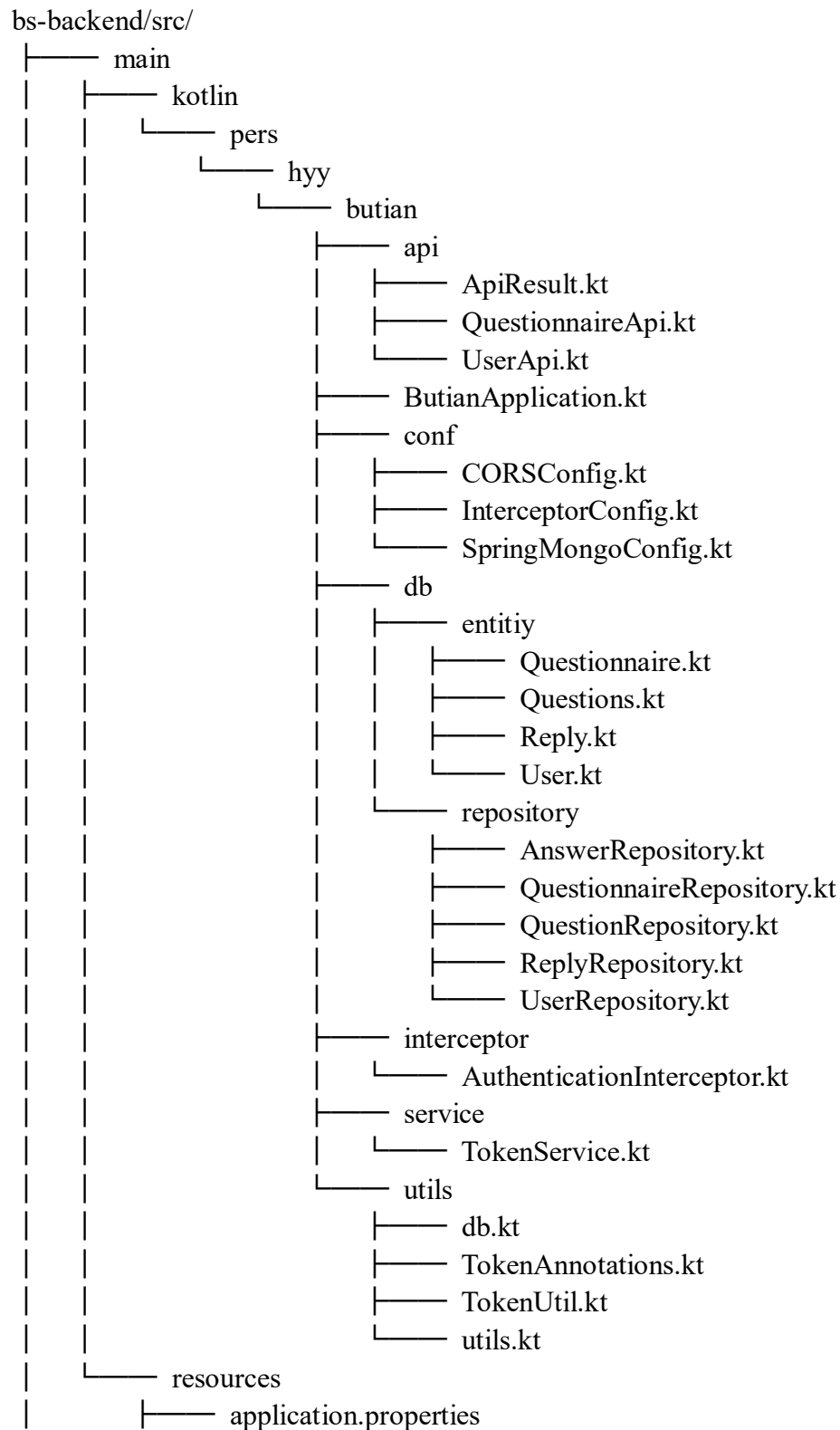
## 2.3 后端技术

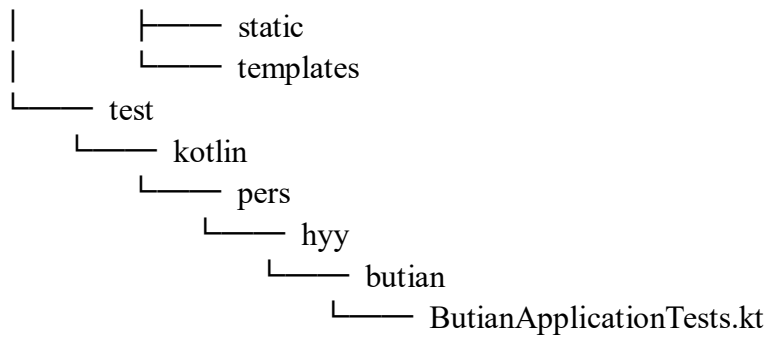
本项目暂定使用 Spring Boot 作为后端框架，通过 Kotlin 语言进行开发。Spring Boot 是一个基于 Spring 框架进行设计的，目的在于简化 Spring 应用搭建和开发过程的开源轻量级框架；Kotlin 是 JetBrains 开发的兼容 Java 的语言，较 Java 而言拥有更加灵活、简洁、丰富的语法特性；Spring Boot 框架也对使用 Kotlin 开发提供了良好的支持。



数据库方面使用 MongoDB，其灵活的文档型结构能够比 SQL 更加便于调查问卷系统的数据库设计；在后端程序中，我们使用 Spring Framework 的 Spring Data Reactive Repositories 就可以很方便地与 MongoDB 进行数据交互。

## 2.4.1 后端代码结构





## 3 数据库设计

本章中列出了应用所使用的数据库模型的初步设计，实现细节可能会有所变动。

### 3.1 用户数据

#### 3.1.1 用户信息

使用用户机制主要是为了实现用户的注册与登录功能，以及相关的问卷/填写应该能与用户建立关系。由于用户与问卷是一对多的关系，我们这里就不在用户信息中保存其所创建的问卷相关的信息。

Collection user:

```
{
  _id: <ObjectId>,
  user_id: String,
  user_email: String,
  hash_pwd: String,
  salt: String,
  profile: {
    portrait_url: String,
    description: Text
  },
  auth: {
    access_level: Integer,
    group: String
  }
}
```

这里设计的用户集合包含了用户登录所需信息，用户名与邮箱，以及相关的用户资料和授权信息，具体内容按功能的实际需求可以进行变更。

### 3.2.2 用户 Session 信息

通过 session 机制，我们就可以让用户在多平台登录，且维护用户的登录状态；与前文提到的 token 验证登录的机制密切相关。

Collection session

```
{
  _id: <ObjectId>,
  user_id: <ObjectId>,
  timestamp: DateTime,
  token: String
}
```

## 3.2 问卷数据

### 3.2.1 问卷

由于问卷数据的结构较为复杂，如果用关系型数据库的方式来储存需要对每个题型都创建一个表，然后通过 id 将题目和问卷联系起来，储存方式非常分散；而是用 MongoDB 则可以通过相对而言 schema-free 的 BSON 文档形式存储问卷的信息，如：

Collection survey:

```
{
  _id: <ObjectId>,
  title: String,
  createby: <ObjectId>,
  decription: String,
  start_time: Datetime,
  end_time: Datetime,
  display_mode: String,
  survey_mode: String,
  section: [[Integer]],
  questions: [Question]
}
```

上述的 survey 集合中存储了问卷的标题，创建信息，截止日期，展示模式，问卷填写模式以及问卷分栏，问卷中的问题信息。

### 3.2.2 问题（问卷项目）

这里的 Question 代表的是一类被嵌套的文档，而不需要成为单独的集合。由于问题可能对应着多种不同种类的问题，我们这里不对 Question 的 schema 要求完全一致，而只是要求它们都具有某些必要的共同字段。这种设计方式与 OOP

思想中类的继承相似，在 MongoDB 中是一种常见的实践。

Question

```
{
  _id: Integer,
  index: Integer,
  type: String,
  enable: String,
  description: String,
  dependencies: [Integer]
}
```

特别地，这里的 `dependencies` 一项用于实现题目之间的逻辑依赖关系（既所谓的级联）；为了方便在问卷进行时对题目进行修改，这里的记录题目的 `enable` 状态和题目在问卷中的位置 `index`，使得修改问卷后，旧的结果也能被保存下来。

单选题/多选题

```
{
  _id: Integer,
  index: Integer,
  type: String,
  description: String,
  limit: Integer,
  options: [Option]
}
```

`Limit` 表示选择的数量限制，`Option` 则又是一类嵌套的子文档：

Option

```
{
  _id: Integer,
  description: String,
  dependencies: [Integer]
}
```

这里的 `dependencies` 储存的即是选项之间的逻辑依赖关系。

文本填写题

```
{
  _id: Integer,
  index: Integer,
  type: String,
  description: String,
  multi_line: Boolean,
  word_limit: Integer
}
```

`multi_line` 区分多行与单行，`word_limit` 规定单词数限制。

### 评分题

```
{
  _id: Integer,
  index: Integer,
  type: String,
  description: String,
  half: Boolean,
  limit: Integer,
}
```

half 表示是 0.5 一档还是 1 分一档的评分；limit 表示评分范围。

### 多层次选择

这里我们只需要把 Option 改为以下形式就可以实现省市之类多层次选择题：

Option

```
{
  _id: Integer,
  description: String,
  dependencies: [Integer],
  options: [Options]
}
```

由于计划实现的题型种类较多，且剩余的题目形式在数据库设计上都大同小异，这里就不再一一赘述。

## 3.3 答卷数据

我们的答卷数据将对应问卷设计的结构进行数据库设计，依然以文档的形式储存：

Collection submission:

```
{
  _id: <ObjectId>,
  survey_id: <ObjectId>,
  submit_time: DateTime,
  userId: <ObjectId>,
  ip_address: String,
  data: [Answer]
}
```

其中 Answer 即对应着每道题的答案，以单选/多选题为例：

Answer

```
{
  problem_id: Integer,
  data: [Number]
}
```

不同的题型答案数据都可以存在 data 字段里，但可以使用不同的类型。

### 3.4 SpringBoot MongoDB 中数据类型具体实现

在 Spring data mongoDB API 中，我们可以用定义类的方法，配合@Document 装饰符，来生成文档类型的定义，在 kotlin 中，我们可以通过定义 data class 来实现；同时我们还可以通过定义 interface 的方式来实现投影（projection, view）的效果；以下的代码展示了本项目是怎样具体实现这个数据库设计的。

#### Questionnaire.kt

```
@Document
data class Questionnaire(
    @Id var id: String?,
    var title: String?,
    var requireLogin: Boolean=false,
    var submitLimit: Int?,
    var description: String?,
    @Indexed var createdBy: String?,
    var startTime: Date?,
    var endTime: Date?,
    var displayMode: String?,
    var surveyMode: String?,
    var section: List<Int>?,
    @DBRef var questions: List<Question>?
)

interface QuestionnaireBasicInfo {
    val id:String?
    var title: String?
    var createdBy: String?
    var startTime: Date?
    var endTime: Date?
}
```

#### Question.kt

```
@Document
data class Question(
    @Id var id: String?,
    var mustFill: Boolean,
    var index: Int,
    var type: String,
    var title: String,
    var description: String?,
    var dependencies: List<Int>,
    var content: Map<String, Any>
)
```

)

### **Reply.kt**

@Document

```
data class Answer(@Id var id: String?, @Indexed val questionId: String?, val value: Any?)
```

@Document

```
data class Reply(  
    @Id var id: String?,  
    @Indexed  
    val questionnaireId: String?,  
    val userName: String?,  
    @CreatedDate  
    var createdAt: Date?,  
    val userIp: String?,  
    @DBRef  
    val answer: List<Answer>  
)
```

### **User.kt**

@Document

```
data class UserAuth(val accessLevel: Int, val group: String)
```

@Document

```
data class User(@Indexed(unique=true) var userName: String,  
    @Indexed(unique=true) var userEmail: String,  
    var hashPwd: String,  
    var salt: String,  
    var auth: UserAuth= UserAuth(1, "user"))
```

## **4 UI 设计**

为了方便进一步的前端开发，在设计阶段，我们以 UI demo 的方式来对前端界面进行初步的设计；在实际开发中再进行具体的功能细化与设计美化。

本次项目的前端设计风格主要参考了 Google form 与腾讯表单的风格（其实两者某种程度上相当相似）。

下面本文档将会简单介绍控制台界面、问卷设计界面、问卷填写界面、问卷数据界面四个主要界面的设计。

## 4.1 用户问卷管理（控制台界面）

在该界面用户可以直观地获取其已创建的一些问卷的简略信息，并向用户提供了进一步管理问卷、获取数据的入口。

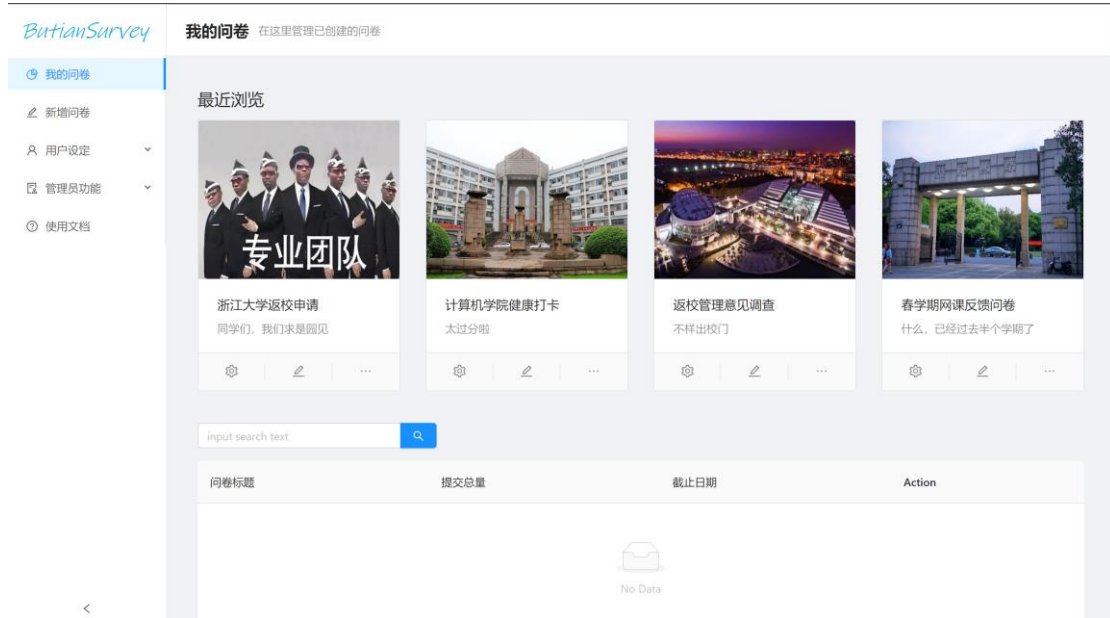
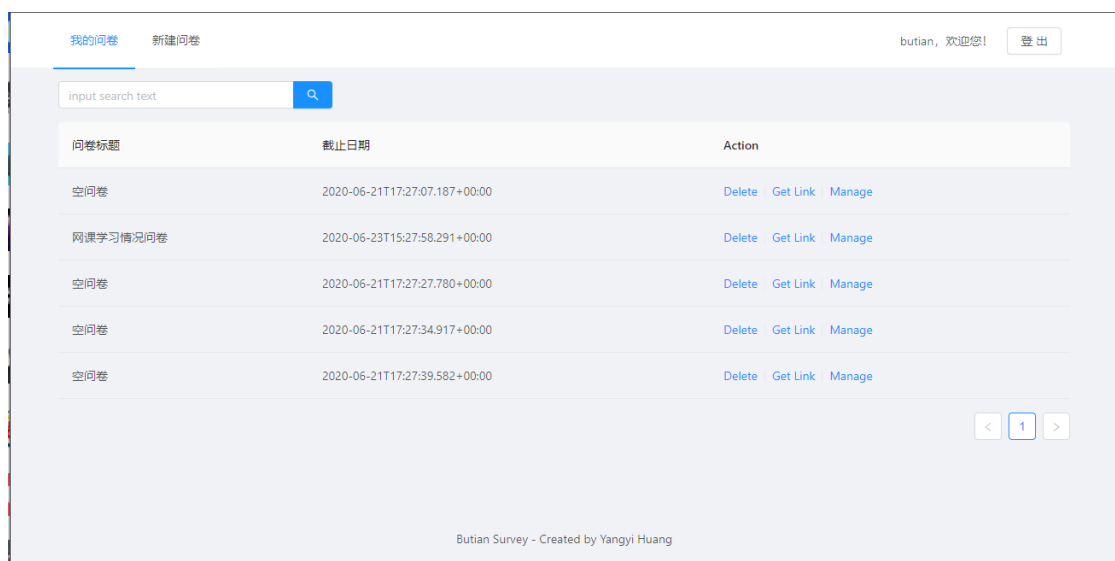


Figure 2 用户问卷管理界面（原设计）

页面首先会以卡片的方式对用户最近进行过浏览的问卷进行快捷方式的展示；对于更多的用户问卷，则会通过搜索栏+表格的方式予以展示；这样的展示方式能够让用户较为便捷地寻找到目标问卷的入口。

在实际实现中，发现主页并不需要实现如此复杂的功能，将设计最简化结果如下：





## 4.2 问卷编辑界面



Figure 3 问卷编辑界面

对于问卷编辑、问卷填写和问卷数据查看界面，本项目主要采用了与 Google form 相似的布局风格，将三个不同的视图整合到一个页面上，方便了问卷设计者对问卷的管理，对于轻度使用的用户有相当好的交互体验。

问卷编辑界面采用浮动卡片式的布局，用户可以通过拖动问题卡片调整问题在问卷中的信息；通过新建卡片再调整题目类型的方式，或者由快捷工具栏直接拖出的方式，用户能够很快捷地进行题目和问卷信息卡片的创建；工具栏中还能让用户新增问卷节，制作出具有多个 section 的问卷。笔者计划在问卷中对 Markdown 格式文本、图片、视频进行简单的支持，使得问卷向填写者提供的信息能够更加丰富

这里的 demo 还展示了对单选题进行编辑时的交互设计，用户可以对选项进行便捷的添加、编辑与顺序调整等操作。

在实际设计中，这一界面也进行了大幅度，首先采用了圆角矩形悬浮卡片的设计，更加友好美观，对交互逻辑也进行了重新的设计：



## 4.3 问卷填写界面

问卷填写界面整体的风格与问卷编辑相似，但是孤立浮动的卡片将会在填写界面连为一个整体，形成更加一体化的视觉体验；特别地，对于有分节的问卷，将在页头显示当前填写的节的简介与进度，以及在视图的上下两处固定翻页的交互按钮；对于采用卡片式视图（即一道一道题填写），其体验应与每道题一节的问卷视图相近。

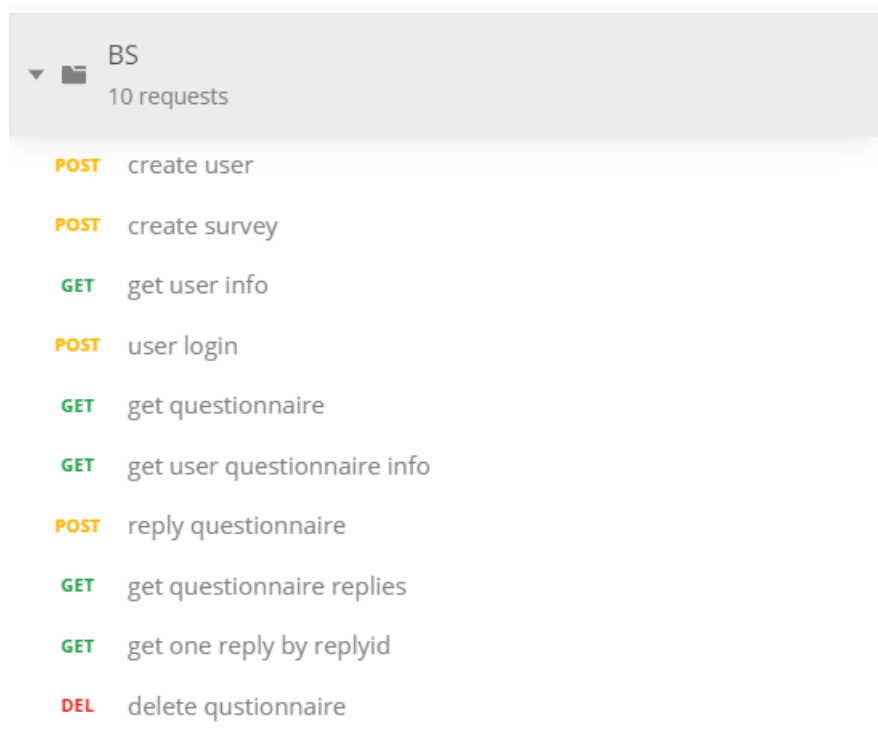
另外项目还计划在填写视图中增加回到第一个未填写的题目的快捷浮动按钮、以及浮动的填写提示进度条，这能够让填写者更加便捷的进行长问卷的填写。

## 4.4 问卷数据界面

首先该视图会以列表的形式向用户展示最近提交的若干份问卷，并对所有提交提供搜索和导出功能。列表下方还将提供简单的数据可视化功能，用户展开该界面后，应能看到卡片式的可选样式的对于每个问题的数据可视化（饼图、柱状图、地理热图等）。这能够满足用户对于直观及时地对问卷数据进行简单统计上的理解的一大需求。

## 5 后端 API 设计

在后端 API 设计的过程中，主要使用 postman 对 API 接口进行调试，以及生成 API 文档，其导出的 JSON 格式数据文件在附件中。



## 二、使用手册部分

### 0 简介

首先，欢迎您选择 ButianSurvey 作为您发布问卷的平台！本项目为实验性质的项目，可能具有各种方面的漏洞与功能的不足之处；如果您有好的建议，欢迎在 <https://github.com/huangyangyi/ButianSurvey/issues> 进行反馈！

本文是 ButianSurvey 的用户使用手册，对项目的功能使用进行了简单介绍，如果有未阐述清楚的操作上的细节问题，也欢迎在 Issues 界面内反馈！

### 1 运行与部署

如果您想在自己的服务器上部署 ButianSurvey，您需要安装以下依赖环境：

1. Node.js，以及前端项目中记录的相关依赖包；
2. JRE, Gradle，因为项目暂时还未打包发布，您可能需要运行后端 Gradle 项目的 IDE，如果您安装了 IntelliJ IDEA，那么它会是一个非常合适的选择；
3. MongoDB，请选择较新的版本以确保兼容性

由于项目尚未进入 release 阶段，部署方法可能较为繁琐复杂，敬请理解原谅！

### 2 登录与注册

要使用 ButianSurvey 的完整功能，如问卷编辑与发布，填写注册用户问卷，您需要先在平台上进行注册与登录。在登录前，您打开任意本项目中的链接都会跳转到“域名/login”界面

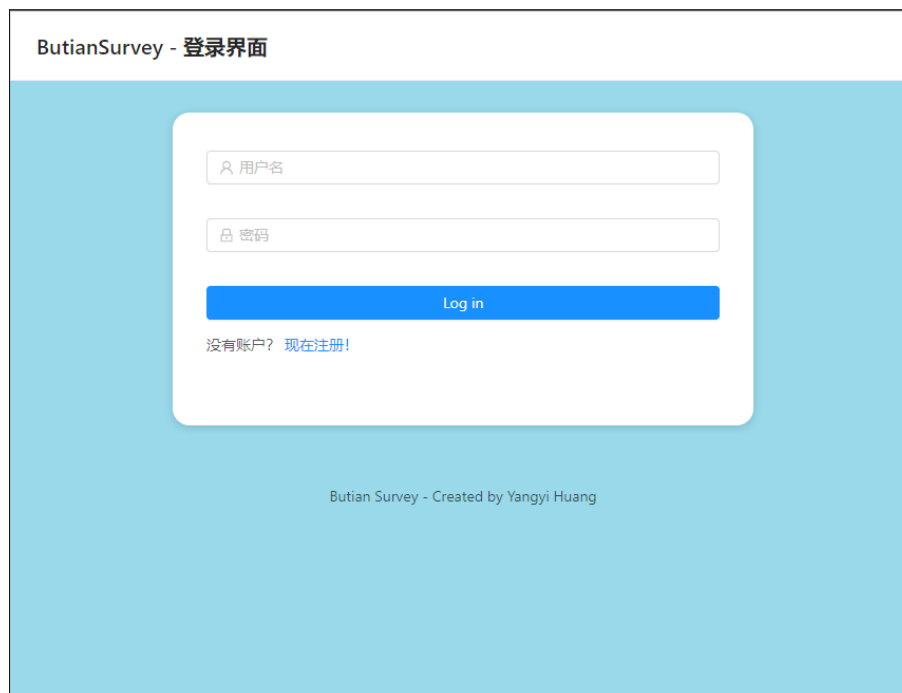


Figure 4 Login 界面

在登陆界面中，没有注册过的用户可以点击链接进入注册界面，注册信息有以下要求：

- 用户名应由英文字母大小写，数字和下划线组成，长度应不少于 6 个字符
- 用户邮箱应当为合法的邮箱格式
- 密码不应少于 6 个字符



Figure 5 注册界面

### 3 管理现有的问卷

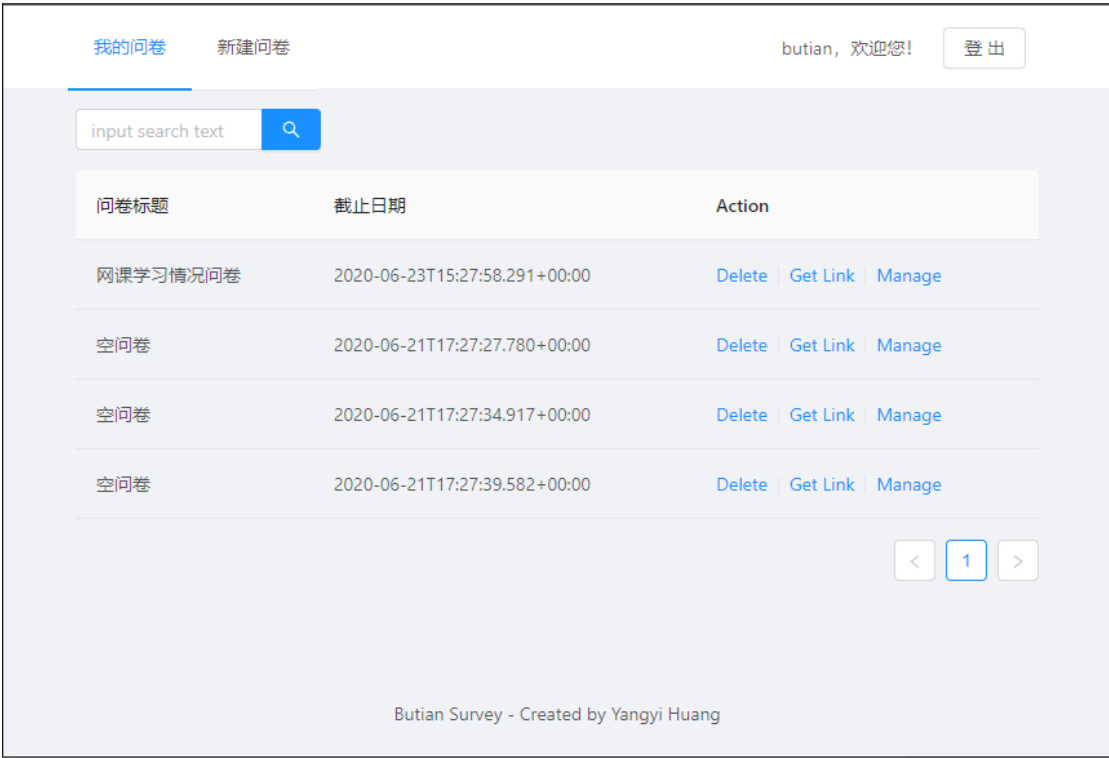


Figure 6 主页问卷管理

登录后，访问域名您即可跳转到网站的主页，在此界面，您可以直观地获取已经创建的问卷大致信息，以及对他们进行删除、查看链接、管理等操作。

单击表项中的 **Delete** 会将问卷删除；单击 **Get Link** 会跳转到问卷的填写界面，您可以将该链接复制并分享给您的问卷收集对象，他们便可以在相应的页面进行问卷的填写；**Manage** 将会链接至该问卷集编辑、预览、查看结果、设置于一体的管理界面。

点击新建问卷将跳转到新建问卷的管理界面，在您选择保存之前，新建问卷的数据将不会被提交。

### 4 问卷编辑

通过点击 **Manage** 链接或者新建问卷，我们能进入问卷管理界面；在上方的选项卡中选择“编辑”分页即可进入。



Figure 7 编辑界面

将光标移到标题位置以及问卷描述位置，您一定能注意到您可以很便捷直观地随时对这两项属性进行修改。



Figure 8 标题及描述编辑

在页面下方，您能看见一个蓝色圆形加号按钮，单击它便可创建新问卷：



Figure 9 添加题目

对于每道题的编辑操作，我们可以通过题目标题旁的编辑按钮、右上方的必填/非必填开关、题目类型选择等位置进行操作。以单选题为例，能够进行的编辑操作如下示意图所示：

01

你的专业类别？

题目类型选择

单项选择题

必

编辑标题

1. 工科类专业

2. 理科类专业

3. 社科类专业

4. 其他

新增选项

点击以新增选项

选项可拖动调整顺序

其他选项 ...

删除选项

删除问卷

添加级联依赖

添加问题说明

02

当前所处年级

单项选择题

必

Figure 10 问卷编辑界面

目前本系统支持以下几种题型：

01

你的专业类别？

单项选择题

必

1. 工科类专业

2. 理科类专业

3. 社科类专业

4. 其他

新增选项

...

Figure 11 单选题

03

网课学习对于你的毕业过程有什么影响？

填空题

必

如毕业设计的进行、毕业所需课程的参与等等。

字符限制: 200 (< 10000)

填空类型:

单行文本

多行文本

输入框预览

依赖题目 第2题

选项 4.

取消依赖

...

Figure 12 填空题

Figure 13 多选题

Figure 14 评分题

Figure 15 数值题

特别地，我们还可以设置一种级联选择机制，即通过一些题目的选项，可以决定另一些题目是否被显示出来；对于需要依赖某些题目答案的题目，我们选取“更多”选单中的添加级联依赖即可进行设置。

Figure 16 设置级联依赖



点击取消依赖，我们就可以移除这项级联依赖。

有时候题目的标题因为长度有限，不能很好的向用户传达题意；通过选单中的“添加题目描述”，你可以在标题下方的描述区域添加描述；题目描述也是可选的。



Figure 17 题目描述的编辑

编辑到这里，想必你已经注意到了，当鼠标光标放到题目卡片上时，会显示“移动”状态的光标；没错！您可以通过拖动题目卡片便捷地对题目顺序进行调整，免却了同类项目中一下下点击移动按钮甚至是只能重新新建题目的痛苦。



Figure 18 题目顺序调整

在进行编辑的过程中，您随时可以切换到预览模式对您问卷的显示效果进行查看，只需切换到“预览”分页即可；预览问卷的数据会随着编辑的过程实时更新。

ButianSurvey - 网课学习情况问卷 问卷管理界面

保存 删除

编辑 预览 回复 设定

## 网课学习情况问卷

大家使用网课学习了一个学期，把你的学习感受反馈一下吧？

01 | 你的专业类别？ (必填)

☐ 工科类专业

☐ 理科类专业

☐ 社科类专业

☐ 其他

02 | 当前所处年级 (必填)

当前所处年级

☐ 大一

Figure 19 问卷预览界面

记得保存！完成了问卷的编辑，您可以点击右上方的保存按钮保存更改；别担心，您仍可以随时修改这一问卷；新增问卷将在第一次保存后跳转到该问卷专属的编辑页面。如果保存成功，您将能看到相应的提示消息。

ButianSurvey - 网课学习情况问卷 问卷管理界面

保存成功!

保存 删除

编辑 预览 回复 设定

Figure 20 保存成功

## 5 设置问卷发布

别着急，您还没对问卷的发布信息进行设置呢！ButianSurvey 采用的发布模式是按时间和填写者进行发布的，还可以对每位填写者的提交次数做出限制。



值得注意的是，对于注册用户，本系统是通过用户名进行用户身份识别的；对于匿名用户（即未注册用户），本系统则通过其 IP 地址分辨其身份。

## 6 问卷填写

使用首页中获得的分享链接，您就可以在问卷填写界面对问卷进行填写了：



Figure 21 问卷填写界面

提交问卷后，页面将会弹出提交成功的消息，页面的输入组件将会被禁用，因而您不用担心提交是否成功的事情；如果问卷需要登录才能填写，页面中也会

弹出相应的提示。

## 7 查看问卷结果

在问卷管理界面的“回复”分页，您可以查看目前填写问卷的总人数，不同用户的填写时间和他们填写的答卷具体结果：

编辑预览回复设置

总计提交：7

用户名	提交时间	Action
匿名	2020-06-21T15:17:05.976+00:00	查看答卷
匿名	2020-06-21T15:17:26.846+00:00	查看答卷
匿名	2020-06-21T15:26:31.322+00:00	查看答卷
匿名	2020-06-21T15:35:30.679+00:00	查看答卷
匿名	2020-06-21T15:42:01.406+00:00	查看答卷
butian	2020-06-21T15:45:52.552+00:00	查看答卷
butian	2020-06-21T15:46:09.544+00:00	查看答卷

< 1 >

点击“查看答卷”，答卷将以填写时的形式向您直观清晰地展现出来：

网课学习情况问卷

大家使用网课学习了一个学期，把你的学习感受反馈一下吧？

01 | 你的专业类别？（必填）

☐ 工科类专业

☐ 理科类专业

☐ 社科类专业

☒ 其他

02 | 当前所处年级（必填）

当前所处年级

☐ 大一

☒ 大二

Figure 22 结果答卷详情

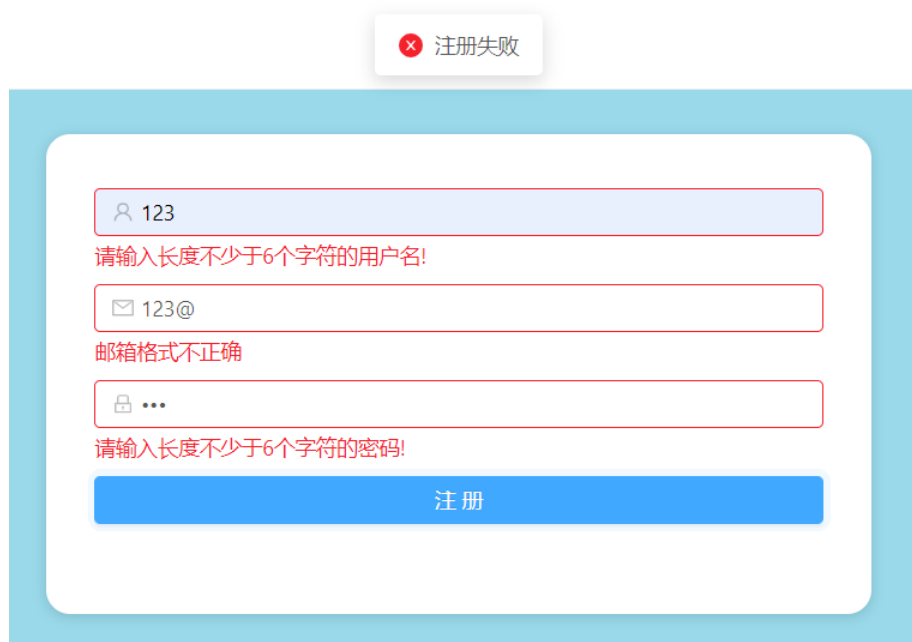
## 三、测试报告部分

### 0 登录注册功能测试

首先测试用户注册界面，实验要求中对用户注册有以下的要求：


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

先尝试输入错误格式的用户名、邮箱和密码，可以发现页面上出现了错误提示信息，且注册信息不能提交：



The image shows a registration form with three input fields and a '注册' (Register) button. Above the form is a red error message: '注册失败' (Registration failed). The first field (username) contains '123' and has a red error message below it: '请输入长度不少于6个字符的用户名!' (Please enter a username of at least 6 characters!). The second field (email) contains '123@' and has a red error message below it: '邮箱格式不正确' (Email format is incorrect). The third field (password) contains '...' and has a red error message below it: '请输入长度不少于6个字符的密码!' (Please enter a password of at least 6 characters!).

我们重新对用户注册信息进行更正，尝试提交：



The image shows the same registration form, but with corrected information. The first field (username) contains 'butiansurvey'. The second field (email) contains 'butiansurvey@butiansurvey.com'. The third field (password) contains '.....'. The '注册' (Register) button is still present.

注册面板上的错误提示信息消失了，提交后浏览器自动跳转到了登录界面。我们再来尝试注册重名、重复电子邮箱的用户：

✗ 注册失败Duplicated User!



The image shows a registration form with three input fields: a username field containing 'butiansurvey', an email field containing 'butiansurvey@butiansurvey.com', and a password field with masked characters. Below the fields is a blue '注册' (Register) button. A red error message '✗ 注册失败Duplicated User!' is displayed above the form.

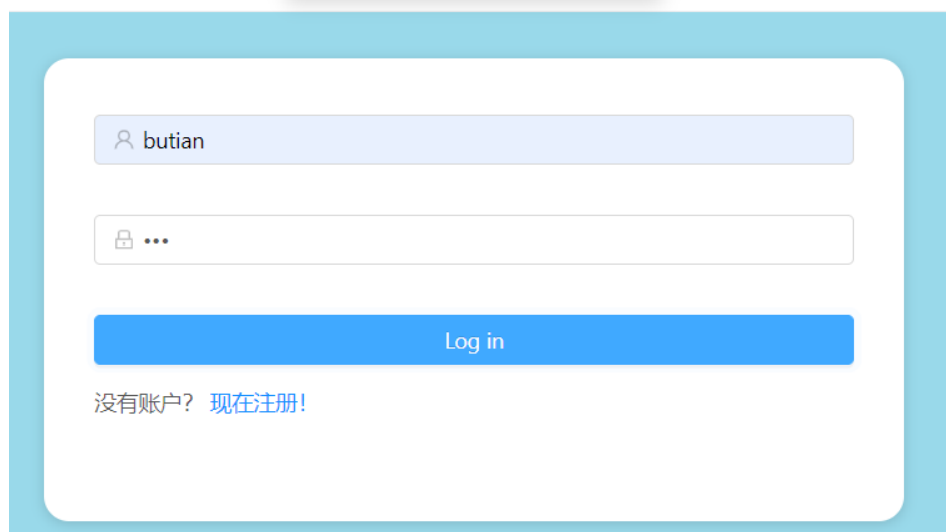
注册失败，网页出现了注册重复用户的错误提示

我们对登录功能的功能设计是，在登录前只能访问少数不需要授权的界面（如匿名问卷、登录界面、注册界面）。我们尝试在不登陆的情况下访问这些界面，结果都是跳转到登录界面。（这一部分不能很好地通过截图反映出来）。

下面我们对登录功能做一个简单的测试：

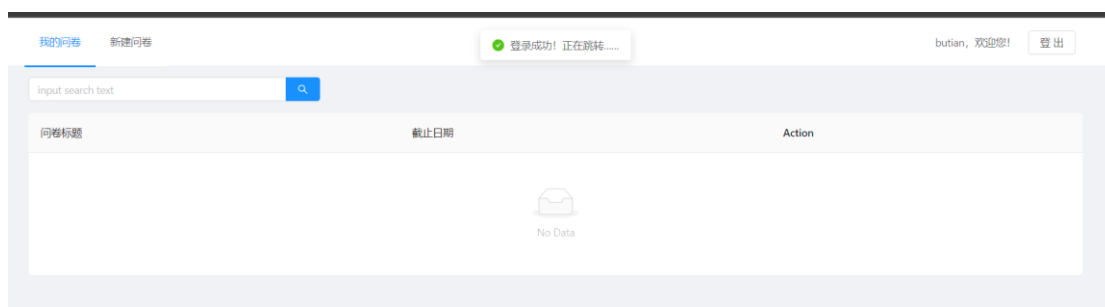
登陆失败：

✗ 登陆失败Authentication failed



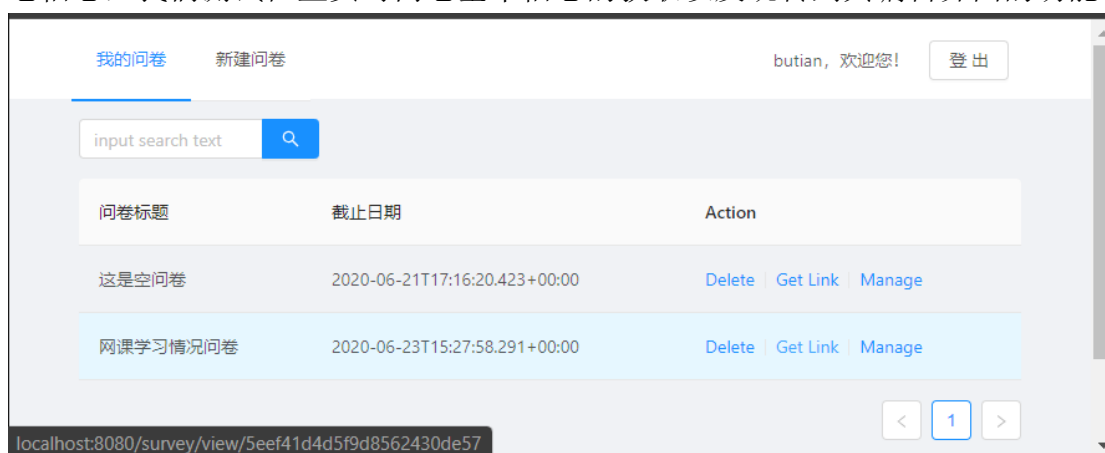
The image shows a login form with two input fields: a username field containing 'butian' and a password field with masked characters. Below the fields is a blue 'Log in' button. A red error message '✗ 登陆失败Authentication failed' is displayed above the form. At the bottom of the form, there is a link '没有账户? 现在注册!' (No account? Register now!).

登录成功：（页面跳转到了用户主页）



## 1 主页功能测试

登录过后，我们会自动跳转到主页，主页中会在一个表格内显示用户所简历的问卷信息，我们测试在主页对问卷基本信息的获取以及跳转到其编辑界面的功能。



可以看到我们能够在这里获取问卷的编辑链接和管理链接。  
接下来我们测试管理界面的删除问卷功能：



可以看到我们成功删除了测试用的空问卷，测试通过。

## 2 问卷设计功能测试

点击主页的新建问卷按钮，我们可以跳转到编辑问卷界面。一个新问卷应当拥有

一个默认的单选题以及空标题、简介等基本数据：



在带有背景图的卡片中，我们可以对问卷标题和问卷描述进行修改；将鼠标放到对应的位置即可触发编辑下划线的显示，单击即可进入编辑：编辑后网页标题栏的问卷标题信息也应当随之更新：



可见基本界面情况正常，我们现在来测试新增题目，单击题目卡片正下方的蓝色加号按钮，即可添加问题。



测试对题目基本信息的修改：点击编辑按钮后触发，题目修改；修改框失去焦点后恢复到一般的显示状态。





测试对题目添加额外的说明，点击编辑卡片右下角的“更多”按钮，即可唤出包含“添加问题说明”的选项；选择后问题会多出一栏可供编辑的问题说明区域。



当鼠标不悬浮且失去焦点的情况下，问题说明编辑框应当不显示编辑框，只是显示更为简洁的一条表示可编辑属性的横线；右上角表示必填的问题基本属性也可以进行修改。且变更后，改选项会在选单中消失。



右上方的下拉选择框可以对题目类型进行变更，下面我们尝试把它变为填空题：



可以看到题目类型顺利变更了，题目编辑的内容也改变了。在“更多”选单中我们还可以为题目添加依赖关系，测试如下：我们添加级联依赖后，将第二题设置为单选题，第三题设置为填空题；



可以看到依赖题目的可选项中只出现了第二题，测试选取依赖题目成功；下面我们来测试选取依赖选项；现在往第二题中添加至四个选项：



可以发现可选的依赖选项能够对应起来，测试通过。

下面我们来测试设置依赖选项的一个难点：改变题目顺序和选项顺序而保持选项不变。首先我们来测试改变题目顺序，我们对题目进行拖拽即可改变题目顺序，这里我们将第二题和第三题对调：



修改问题顺序的功能通过，我们再来修改一下刚才问题的选项顺序。同样我们也用拖动的交互逻辑实现了这一功能，测试效果十分流畅：

依赖题目

第3题

选项

2

三

1.

工科类专业

三

2.

社科类专业

三

3.

其他

三

4.

理科类专业

三

1.

工科类专业

三

2.

社科类专业

三

3.

其他

三

4.

理科类专业

可以发现选项依旧指向该题的“社科类专业”，测试通过。  
 点击“取消依赖按钮”，我们可以将当前题目的依赖关系清除，再次加入后，依赖关系会还原为默认值：

三

4.

大四

依赖题目

无

选项

无

下面我们来对各种问题的设置进行测试。为了节约篇幅，下面列出一份具有所有题型题目的问卷设计完成后的结果。

编辑

预览

回复

设置

网课学习情况问卷

大家使用网课学习了一个学期，把你的学习感受反馈一下吧？

01 你的专业类别？

单项选择题

必

三

1.

工科类专业

X

三

2.

社科类专业

X

三

3.

理科类专业

X

三

4.

其他

X

新增选项

...

删除

02 | 当前所处年级 [🔗](#)

单项选择题 

▼

必

🔴

当前所处年级

≡

1.

大一

X

≡

2.

大二

X

≡

3.

大三

X

≡

4.

大四

X

⊕

新增选项

...

🗑

03 | 网课学习对于你的毕业过程有什么影响? [🔗](#)

填空题 

▼

必

🔴

如毕业设计的进行、毕业所需课程的参与等等。

字符限制: 

200

 (< 10000)

填空类型: 

单行文本

多行文本

输入框预览

依赖题目 

第2题

▼

 选项 

4.

▼

⊖

取消依赖

...

🗑

04 | 上网课的设备 [🔗](#)

多项选择 

▼

必

🔴

你通过什么设备参与网课?

选择数量限制: 

0

 至 

4

 个

≡

1.

手机

🗑

≡

2.

平板电脑

🗑

≡

3.

个人电脑

🗑

≡

4.

其他智能终端

🗑

⊕

新增选项

...

🗑

05 | 你对这段时间的网课教学评价如何? [🔗](#)

评分 

▼

必

🔴

评分单位: 

1

0.5

评分图标: 

☆

♥

👍

👎

...

🗑



之后对问卷进行保存，如果测试成功应该弹出保存成功的提示，且路径应当从新建页面跳转到该问卷的专属编辑路径：



由此可见保存成功，我们再次打开该链接查看能否进入该问卷的编辑界面：



由此可见，问卷编辑功能能够通过基本使用功能的测试。

### 3 问卷预览功能测试

在编辑界面选择预览模式即可查看问卷的预览，因为要展示所有的题目设计情况，所以此处不会考虑题目的级联逻辑。这里我们主要测试两点：问卷预览的正确显示，以及修改问卷后问卷预览能否及时刷新。我们先暂时对第一题的选项顺序进行变换，然后查看预览界面。

编辑

预览

回复

设定

# 网课学习情况问卷

大家使用网课学习了一个学期，把你的学习感受反馈一下吧？

## 01 | 你的专业类别？（必填）

- ☐ 社科类专业
- ☐ 工科类专业
- ☐ 理科类专业
- ☐ 其他

## 02 | 当前所处年级（必填）

当前所处年级

- ☐ 大一
- ☐ 大二
- ☐ 大三
- ☐ 大四

## 03 | 网课学习对于你的毕业过程有什么影响？（必填）

如毕业设计的进行、毕业所需课程的参与等等。

## 04 | 上网课的设备（必填）

你通过什么设备参与网课？

请选择0至4个选项

☐ 手机

☐ 平板电脑

☐ 个人电脑

☐ 其他智能终端

05 | 你对这段时间的网课教学评价如何？（必填）

👍 👍 👍 👍 👍

06 | 每周在网课学习上大概花费多少个小时？（必填）

请输入 0 至 100 范围内的小数

提交

可以看到显示情况正常，提交按钮也被设置为禁用状态，测试通过。

## 4 问卷设置测试

问卷设置界面因为比较简单，具体影响到的逻辑其实在问卷填写功能中，我们这里就对其基本的属性设置等功能进行测试：

若设置为填写次数无限制，则输入填写次数的输入框消失

开放日期： 2020-06-09 ~ 2020-06-18

填写权限： ☐ 所有人 ☒ 注册用户

填写次数限制： 无限制： ☒

若设置为填写次数有限制，则输入填写次数的输入框显示

开放日期： 2020-06-09 ~ 2020-06-18

填写权限： ☐ 所有人 ☒ 注册用户

填写次数限制： 3 无限制： ☐

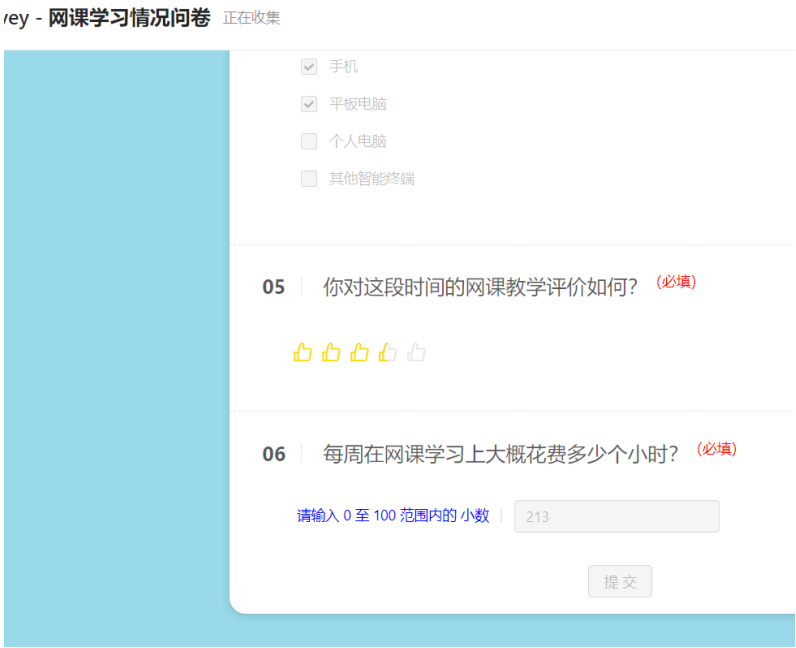
## 5 问卷填写功能测试

问卷保存后，我们就可以使用其链接在收集界面进行填写了，首先我们来测试问卷的访问权限功能是否正常填写功能是否正常

首先我们看到之前所关心的级联选择问题，可以发现填空题的显示与否与第二道选择题的答案有关：



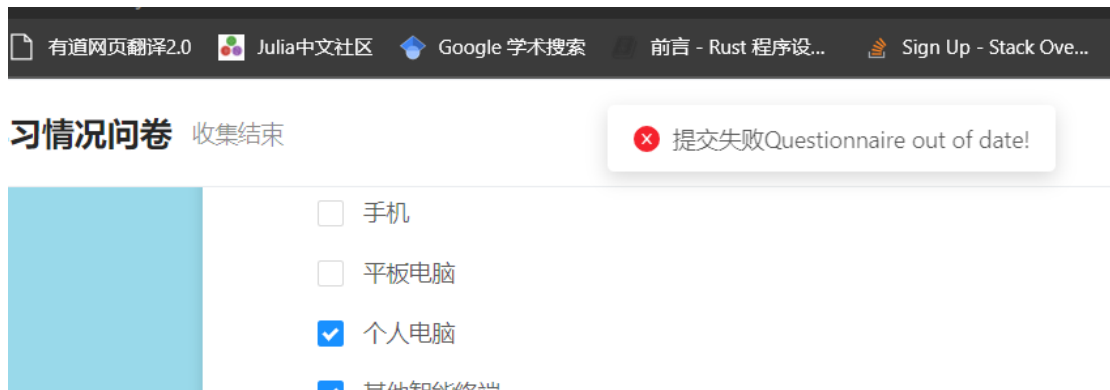
证明我们的问题级联选择机制正确地实现了，测试通过。  
下面我们来测试正常地提交一份问卷：



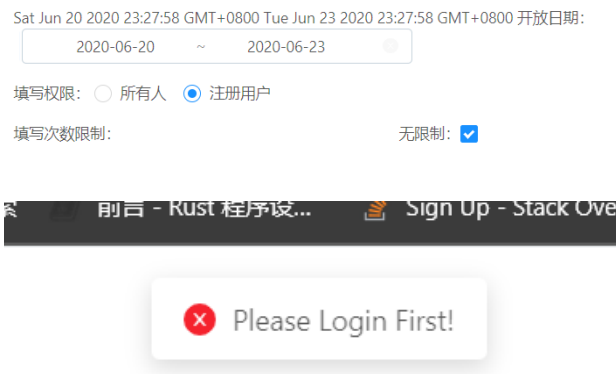
提交成功，且提交按钮被禁用，表示问卷已经提交，测试通过。



下面来测试填写日期期间的限制，我们将问卷的截止日期设在当前日期之前，尝试提交一份问卷：



可以发现问卷状态显示为收集结束，不能提交问卷，测试通过。  
我们将问卷设置为只允许注册用户提交，登出用户后，尝试提交问卷：



可以看见登出后的用户无法提交问卷，被提示先行登录，证明该功能被正确实现。  
我们再将问卷填写次数限制设为 1，这样再次提交问卷的时候用户应当收到报错，下面对此进行测试：



可以发现我们以同一用户进行登录时会收到提交超出限制的提示。

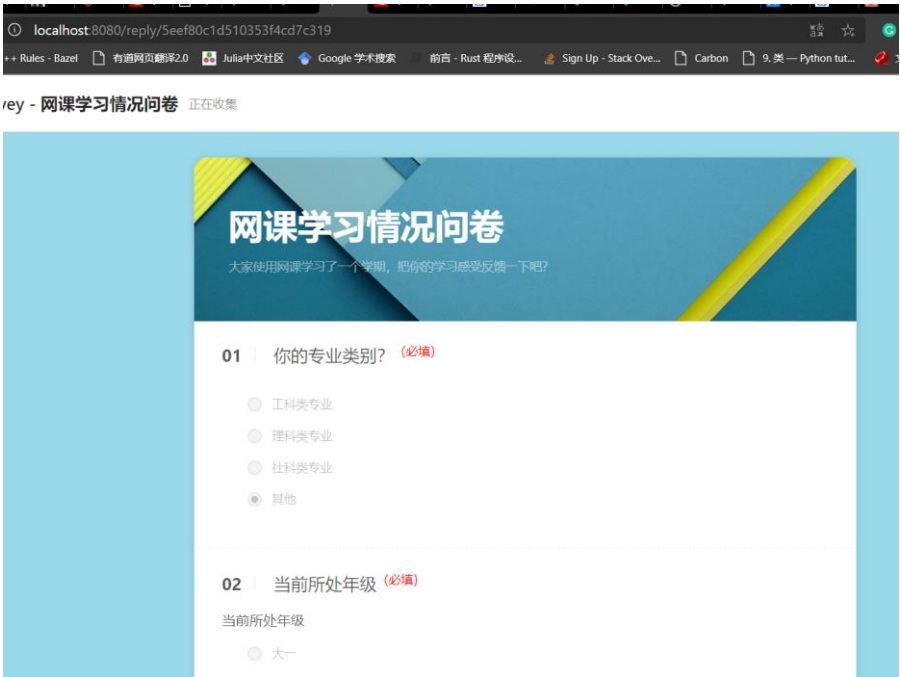
## 6. 问卷结果界面测试

问卷结果界面在问卷管理界面的“回复”分页中，首先在该分页我们能够看到目前已提交的所有问卷以及总数统计，在这里我们对之前的问卷进行测试：



用户名	提交时间	Action
匿名	2020-06-21T15:17:05.976+00:00	<a href="#">查看答卷</a>
匿名	2020-06-21T15:17:26.846+00:00	<a href="#">查看答卷</a>
匿名	2020-06-21T15:26:31.322+00:00	<a href="#">查看答卷</a>
匿名	2020-06-21T15:35:30.679+00:00	<a href="#">查看答卷</a>
匿名	2020-06-21T15:42:01.406+00:00	<a href="#">查看答卷</a>
butian	2020-06-21T15:45:52.552+00:00	<a href="#">查看答卷</a>
butian	2020-06-21T15:46:09.544+00:00	<a href="#">查看答卷</a>

接下来测试提交记录中查看答卷的链接，随意选取最后一个提交的问卷，查看结果：



网课学习情况问卷

大家使用网课学习了一个学期，把你的学习感受反馈一下吧？

01 | 你的专业类别? (必填)

☐ 工科类专业

☐ 理科类专业

☐ 社科类专业

☒ 其他

02 | 当前所处年级 (必填)

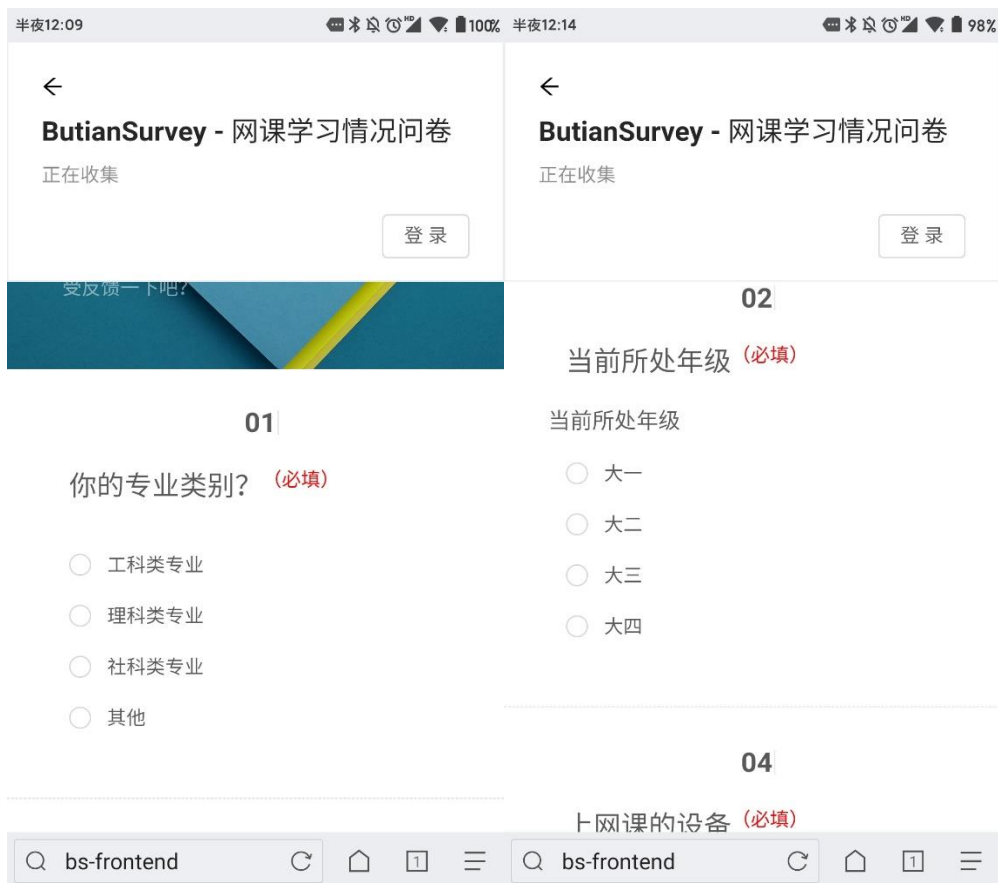
当前所处年级

☐ 大一

链接被跳转到该回复对应的网页，且答卷数据显示情况正常，相关问题选项及提交按钮也显示禁用状态，测试通过。

## 7. 移动端适配测试

这里以移动端最常使用的问卷提交界面作为示例，以展示为主要形式作为测试结果。



在 Mi 6 手机使用 Via 浏览器浏览的结果  
测试得知项目在移动端也具有较好的呈现与交互效果。

## 四、开发感想&小结

总的来说，这次 B/S 课程项目开发我还是遇到了不少的问题的：

首先是题目本身具有一定的难度，问卷系统相对于一般的 Web 开发起手项目来说，交互逻辑、功能等等方面更加复杂，在设计阶段我没有能想到所有的细节，有很多部分都是在实际开发中一边进行开发一遍进行调整；这证明一个良好的项目设计流程也是需要有一定相关经验的。

其次，由于页面逻辑较为复杂，这就要求我们对前端组件的数据流有一个良好的理解和设计；比如我这里所实现的“问题卡片”这一组件，因为其相互之间会互相影响，修改后的数据会影响到页面全局的问卷数据，全局加载数据时又需要传值到组件内；这样父子组件之间的数据流就较为复杂。由于我在本次项目中使用的框架是 Vue，是一种“响应式风格”的框架，一般通过 watcher，双向绑定等机制能够非常无脑地完成根据数据渲染页面的操作；但是在数据流较为复杂的时候，就需要对其存在感并不高的数据流进行进一步的了解才能够写出 bug-free 的代码。这个问题让我对前端设计的难点有了一个新的认识，值得一提的是，另一个框架 react 使用的是函数式的风格，强调数据的 immutable，自然能够“强迫”我这种菜鸟开发者把数据流在设计阶段提前清楚；也许未来了解一下这种框架的使用能令我有进一步的收获。

后端方面开发还是非常顺利的，Springboot 功能之强大实在是令我拍案叫绝，SpringBoot、Spring 利用大量 Java 设计模式以及面向对象的特性对许多机制的封装大大降低了编程的复杂程度，配合 Kotlin 使用更是体验甚佳；Spring 框架相关的一系列高性能功能强大的工具链也让我切身感受到了其在服务端开发中的重要地位治所在；另一方面，MongoDB 的灵活便携也同样让我印象深刻，对于文档型数据的存储，相较于 SQL 型数据库而言，确实是一个非常有说服力的选择。

总得来说，这次项目乃至整个 B/S 课程的学习中，我切实地进一步了解到了 B/S 开发中的细节，也对整个 B/S 体系软件开发的流程、工具、开发模式有了更加系统性的了解；通过实践，我认为我还是在实践中具备了开发小型 Web 项目

的基础能力；从这两点来说，我也算是达成了选课时的初衷。只不过在这次项目开发上，我在熟练度上还有所欠缺，由于开发周期较短，很多很好的设计设想都没来得及实现，这也是一个不小的遗憾吧。