Lab2 初识 Web 开发 小组实验报告

小组成员： 潘星宇 胡彧锋 颜华

引言：*WHERE IS MY HTML, CSS, JS, PHP AND MYSQL?*

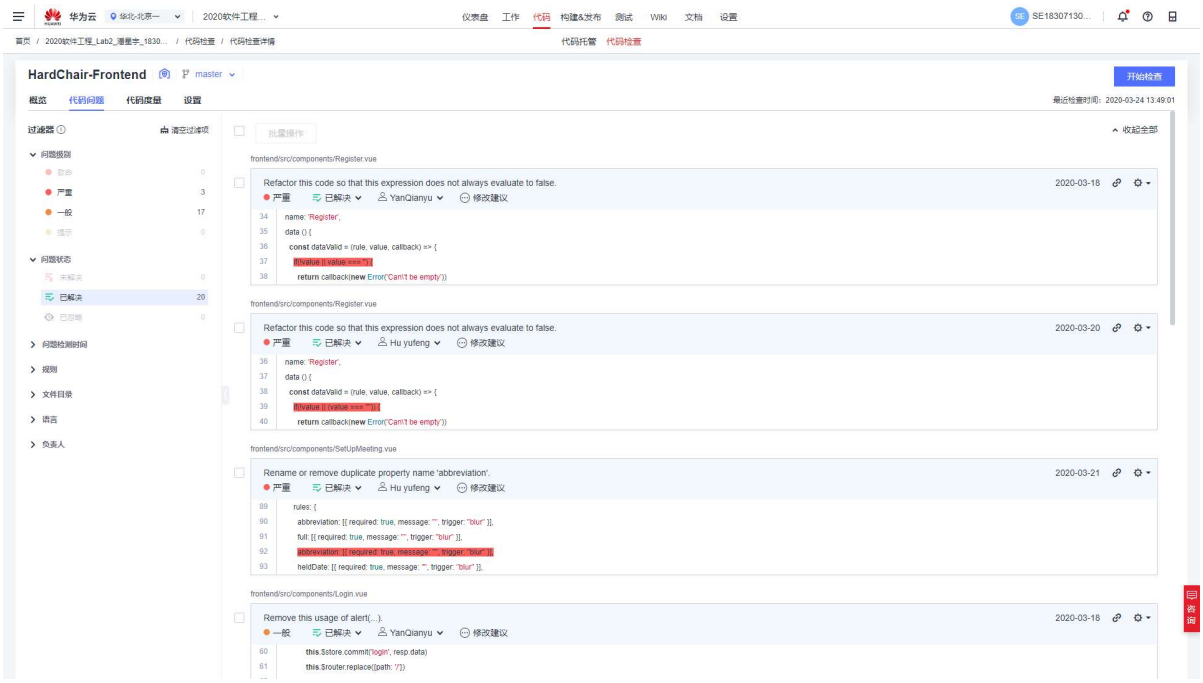
一、DevCloud 项目规划

（一）项目需求规划

image-20200324170655937

（二）代码检查

- 前端

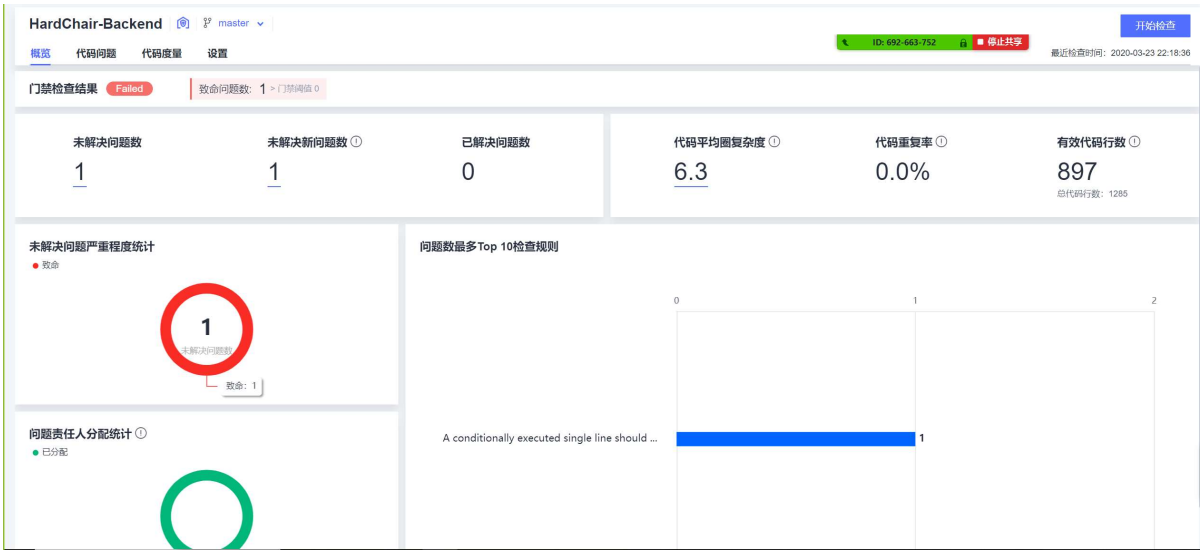


前端错误主要在 alert 部分，解决，替换成 message

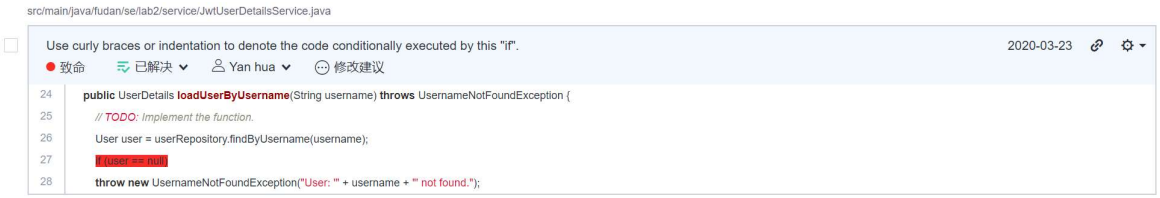
- Alert 警告不会自动消失，Message 消息提示会自动消失

```
alertthis.$message({ type: 'success/warning/info/error', messa
```

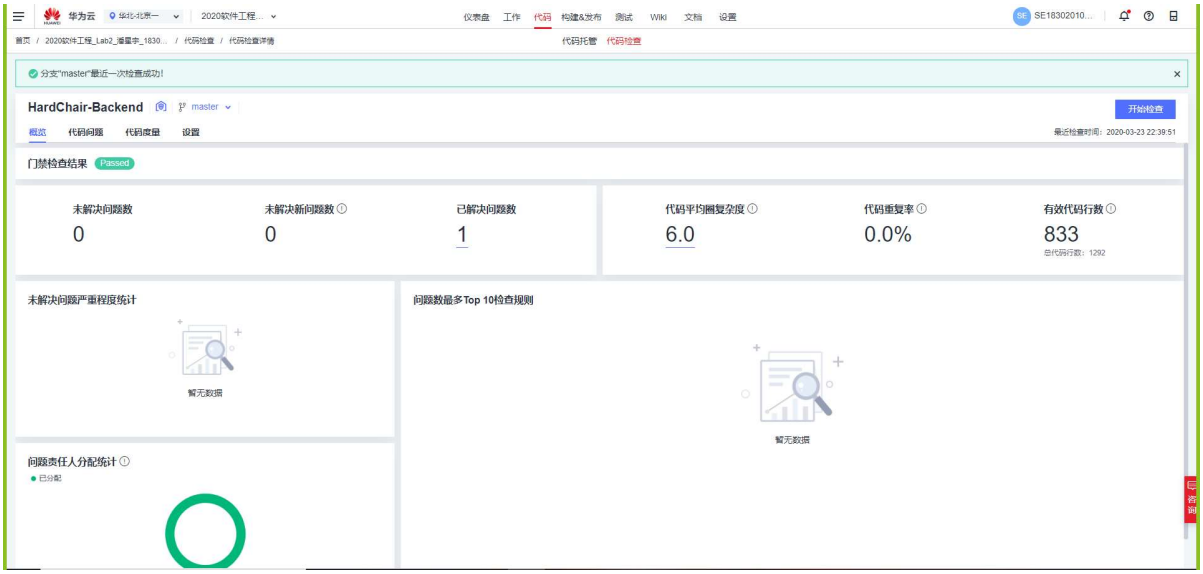
- 后端



首次对后端代码进行代码检查时发现一个致命错误，详细错误信息如下



华为云指出，对于不带大括号的 if 语句，应该通过缩进来表示被 if 包含的语句，一面引起其它开发者和其他代码阅读者（包括其他程序）在内的误解。由于该错误具有的潜在的致命威胁，华为云认为该错误是一个致命错误。后来我们将代码的缩进进行了调整之后，通过了代码检测，没有出现代码问题。下图为无代码问题时的检测结果。



代码检测还为我们提供了代码的其他信息，其中代码的圈复杂度为 6.0，代码重复率为 0.0%，有效代码行数 833/总代码行数 1292.

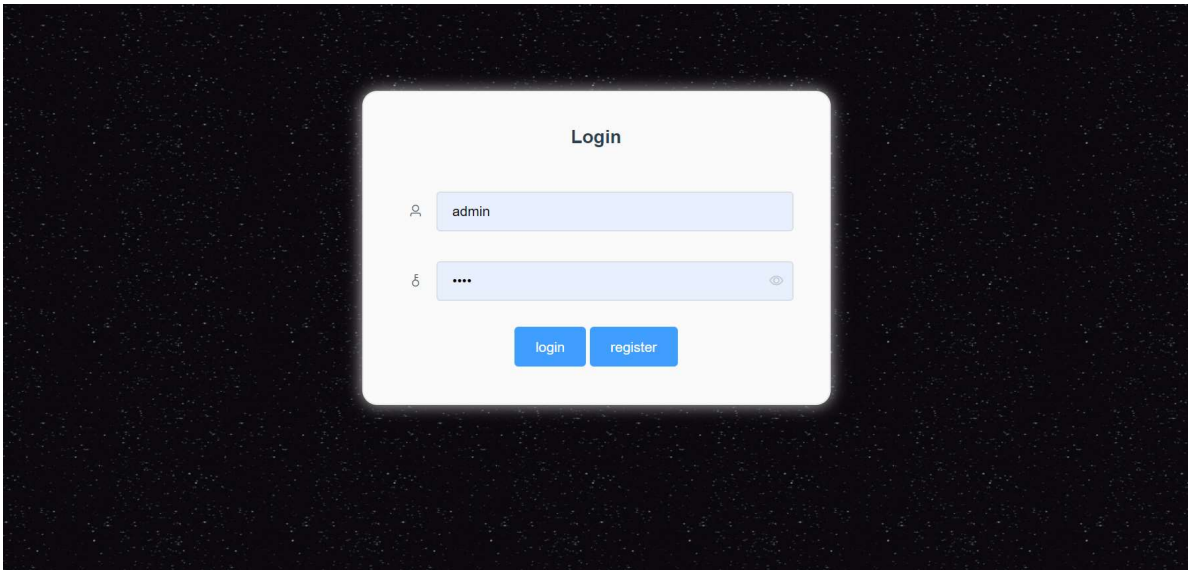
对于代码平均圈复杂度，由于采用了 SpringBoot 框架，利用 IOC 的特性，通过 @Autowire 注解大大简化了 java bean 的构造，无需为每一个 java bean 在一个 main 中手工创建，而由容器在服务启动时统一创建，统一管理，大大减少了业务无关的代码，使得代码的圈复杂度有效降低，方便可维护性和可测试性。

对于代码重复率，由于整个项目的体积不是很庞大，并且每个模块的逻辑不是很复杂，因此重复率为 0 也是情理之中。

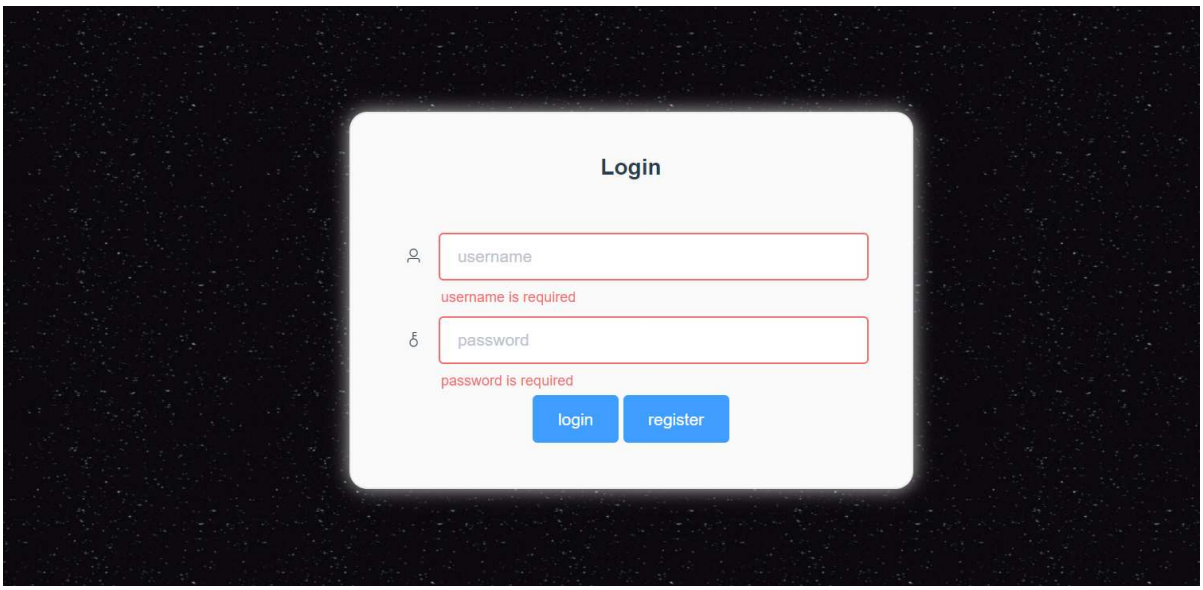
对于有效代码行数，由于考虑到将来的可扩展性，我们对于 Conference 类做了比较完整和详细的设计，其中包含了大量在 Lab2 中不会使用到的属性和方法，同时我们正在建立一些其他的模块，而这些模块由于没有在 Lab2 中使用到，因此被识别为了无效代码。在将来其他功能逐渐加入之后，这些代码将转变为有效代码。

二、项目成果概况

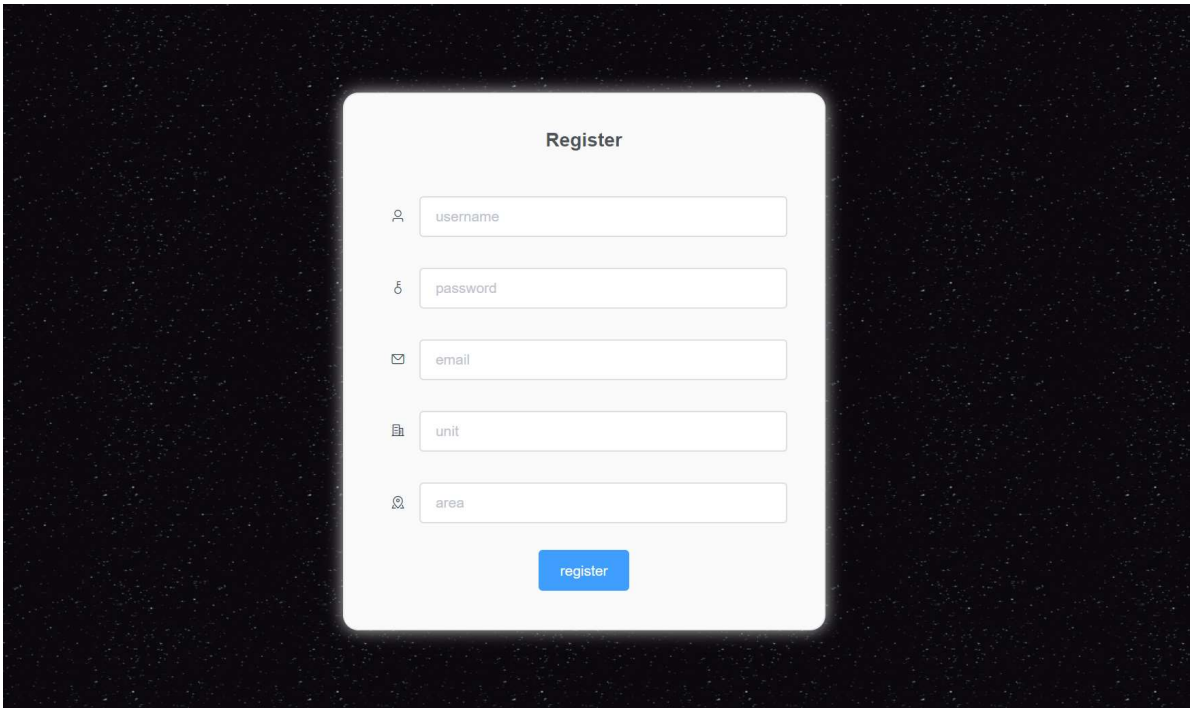
- **登录界面：**



用户可以在此界面进行通过输入用户名和密码进行登录，也可以单击 register 跳转到注册页面。同时前端会对信息是否输入完整进行检测。检查效果如下图：

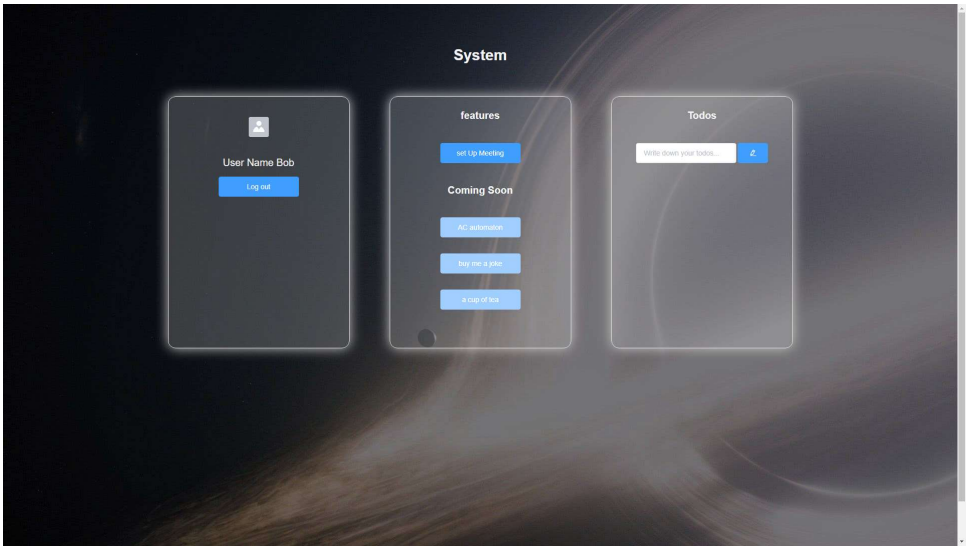


- **注册界面**



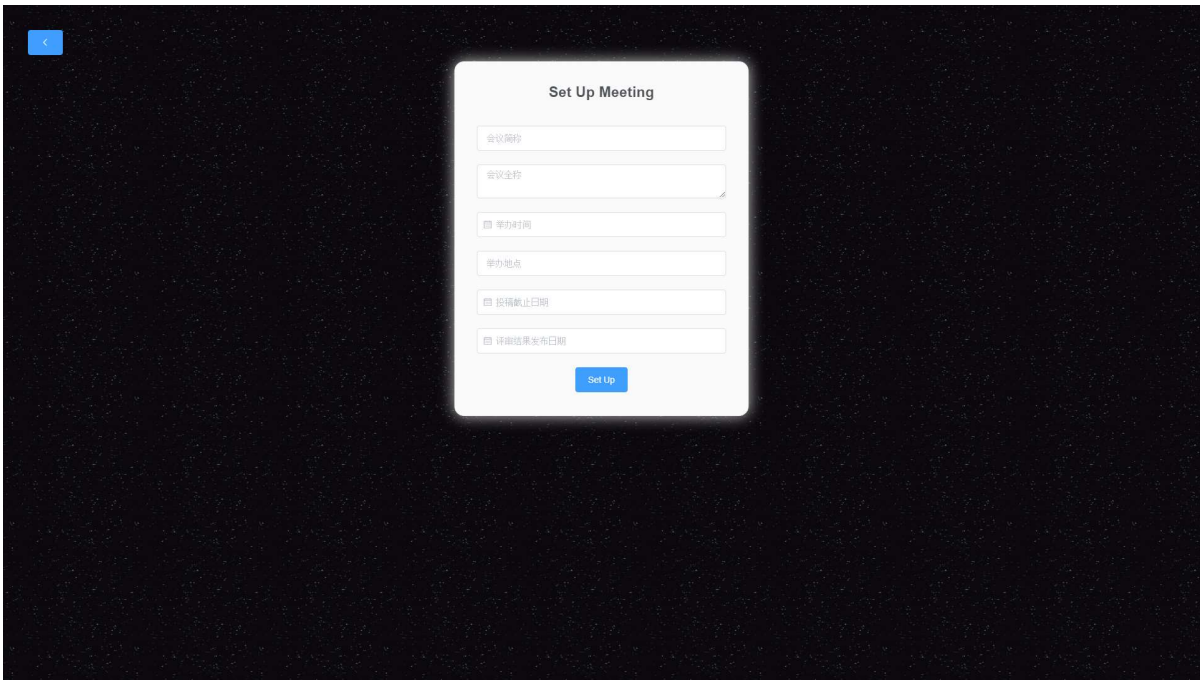
用户可以在此界面进行通过输入用户名（***username***），密码（***password***），邮箱（***email***），单位（***unit***）以及地区（***area***）来进行注册。与登录界面相同，当用户输入的信息不符合格式时，前端会进行实时的提醒。

• 用户系统界面



用户登陆成功后将会进入主界面，在后续功能逐渐完善之后，用户将可以在此界面查看自己的个人信息，查看自己创建的会议，作为投稿人加入的会议，作为审稿人加入的会议等等。此时浏览器已经接收到了登录时从后端返回的 ***token*** ，并存放在 ***COOKIE*** 中。目前，用户系统主界面可以通过点击界面中央的 ***Set up Meeting*** 按钮跳转到会议申请界面。

- **会议申请界面**



与登录注册界面类似，在填写完毕申请信息后，单击 Set Up 按钮就可以提交会议申请。前端会检查每一项信息是否填写完整（但对会议申请的逻辑验证还未完成，包括举办地点的格式验证，举办时间，投稿截止时间，评审结果时间等，留待会议类的实现敲定后再统一实现），并向后端提交申请。此时，前端会在 **HTTP Request Header** 中加入 ***token*** 字段，并将存储在 ***local storage*** 中的 ***token*** 值赋给该字段。

三、组员任务分配情况

胡彧锋（前端）：

注册界面：

- 正则表达式

系统界面

- 跳转路由
- 登出

会议界面

登陆界面

潘星宇（后端）：

会议申请模块：

- Conference 类
- Conference 类数据库接口

Security 模块：

- 访问配置（包括跨预放行与 URI 匹配）
- token 验证（编写 Filter 对 token 进行验证）

颜华（后端）：

登录注册模块：

- 登录用户名密码验证及数据库匹配
- 注册信息查验及数据库更新

会议申请模块：

- 会议申请处理

四、实验过程记录及问题解决

阶段一：学习阶段 （3.15 - 3.17）

该阶段小组内成员根据自身兴趣对 Vue 前端框架以及 Spring boot 后端框架进行了学习。

对于 Vue

- 学习资料
    - Vue.js 官方文档
    - ~~三天~~ **4 个小时速成 Vue.js** 😶

- 头文件引入：`<script src="https://cdn.staticfile.org/vue/2.4.2/vue.min.js"></script>` 或 `<link rel="stylesheet" href="https://unpkg.com/element-ui/lib/theme-chalk/index.css">`

对于 *Spring boot*

我们了解到这是一个 ***Spring*** 应用的开发框架，对 ***Spring*** 的各类技术，包括**Spring Security , Spring MVC** 等，进行了整合。对 ***IOC ， AOP*** 的基本概念有了了解，通过注解配置的方式省去了配置 ***XML*** 的繁琐过程，并对一些常见注解的含义进行了学习。

问题 1：框架学习上手困难

解决：Vue 框架通过学习 Vue 官网 （https://cn.vuejs.org/） 的系列教程可以对 Vue 建立一个初步的认识，同时对照官网教程中的样例代码可以很快入门。对于 Spring Boot 框架，通过自学网上的教学视频（http://www.gulixueyuan.com/my/course/193） 可以对 JAVA EE 有一个基本了解，包括 Servlet， JSP，同时对 MVC 等设计模式的思想进行学习。

阶段二：任务分配 （3.17）

在对整个项目架构有了基本的了解之后，我们对小组内成员的任务进行了分配，初步确定开发流程和时间节点。

阶段三：前后端分离开发 （3.18 - 3.21）

在前端开发出了一个基本的骨架之后，向后端提供了基本的 AJAX 请求实现。开发转入前后端分离开发。前端继续对各个界面进行优化，实现前端逻辑检测和验证。后端通过前端提供的 AJAX 请求接口进行开发。

阶段四：前后端接口统一 （3.21）

经过一段时间的开发，前后端均实现了基本功能。为了实现前后端数据有效交互，前后端对一些接口进行了再次统一，保证前后端能通过 AJAX 请求顺利交互。

阶段五：本地测试以及 DEBUG （3.22 - 3.23）

前后端已经完成基本功能的实现，同时能进行数据交互。因此在本地针对边界情况和系统逻辑进行测试和 DEBUG。

问题 1：关闭浏览器或点击登出后，再次打开浏览器后直接进入到用户主页面。

原因：关闭浏览器或者点击登出后并未删除本地 localStorage 中的 token，再次打开浏览器时前端检测到浏览器中依旧存在 token，则自动重定向到了用户主界面

解决：已解决，解决方法：删除本地 localStorage 中的 token

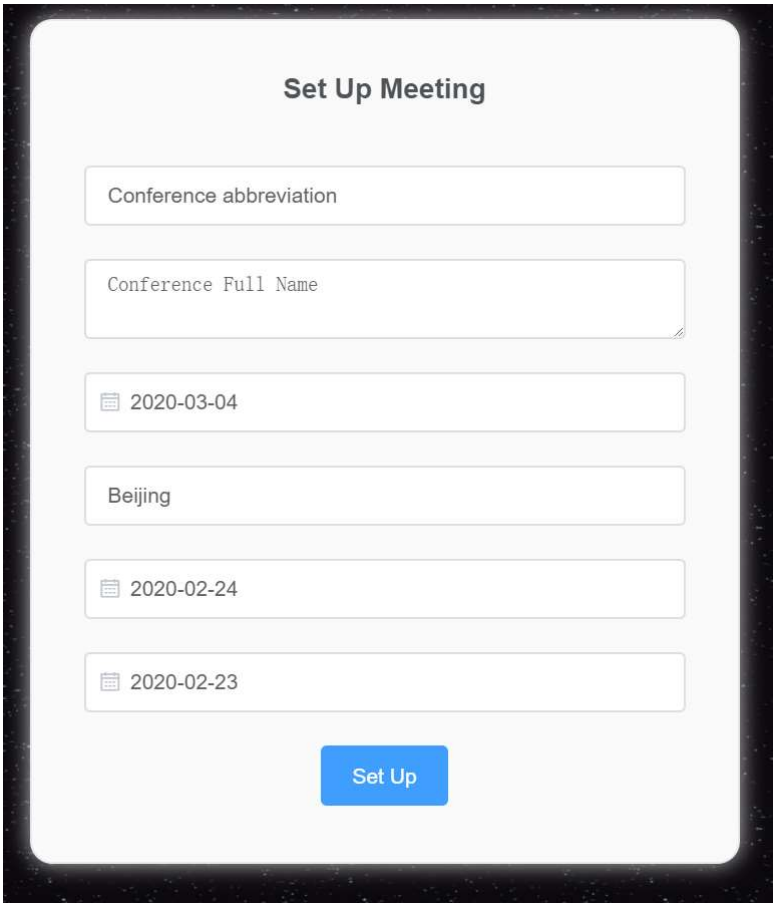
问题 2：用户主界面并未展示用户具体信息

原因：用户主界面现为静态页面，没有通过 AJAX 向后端请求用户具体信息，同时后端也没有提供该请求的实现

解决：留待下一阶段解决

问题 3：会议注册时，前后端均未对会议开始时间，投稿截止时间，评审结果发布时间进行合理性检查

解决：留待下一阶段解决



阶段六：云端部署 （3.23 - 3.24）

本地 DEBUG 之后，将前后端分别部署到云服务器上。前后端均部署在 /usr/local/lab2 目录下，部署顺利。

前后端编译构建

前后端部署

细节

- 编译构建改相对路径

1. 命令：

```
npm config set registry https://mirrors.huaweicloud.com/repository/npm/
npm config set prefix '~/.npm-global'
#如需安装node-sass
#npm config set sass_binary_site https://repo.huaweicloud.com/node-sass/
#npm install node-sass
#进入frontend
cd frontend
#加载依赖
npm install
#默认构建
npm run build
```

2. 构建包路径：

```
./frontend/dist/*
```

阶段七：云端测试 （3.24）

通过 IP（http://114.115.246.37:80/#/login ）访问登录界面，测试后与本地行为一致，登录，注册，会议申请等基本功能实现。

五、小组成员实验总结

胡彧锋

> **本 lab 我负责前端部分**

- 合作感想：小组有分工后，每个人都有明确的任务，这样每个人都可以负责好自己的部分，不会有重复完成相同工作的出现，但在任务交叉处沟通会耗费很多时间，要让别人理解自己也不精通的知识的确不容易，看来全栈工程师着实很厉害 👍

学到知识：

- 什么是前后端分离
    - 传统做法：后台运行 Java 代码，生成全部的 html 代码，接着通过 http 协议把 html 代码传输到浏览器，问题：
        1. 用户等待时间长
        2. 前后端开发效率低

- 前后端分离：先准备一个不包含数据的 html， 把它传给浏览器，然后再通过 Ajax 技术，仅仅从服务器获取 “纯数据”，然后把纯数据显示在 html 上，好处：
    1. 即便是后台数据库比较花时间，但是用户体验也比前面的方式好
    2. 后端只提供数据，所以前后端开发耦合度降低了很多，整体开发效率可以得到较大提高

- 什么是单页面应用（）
    - 单页面应用指只有一个主页面的应用，浏览器一开始就要加载所有必须的 html，js，ss，单页面的页面跳转仅刷新局部资源，多应用于 pc 端
    - 多页面就是指一个页面中有多个页面，页面跳转时是整页刷新
    - 单页面优点：
        1. 用户体验好，快，内容改变不需要重新加载整个页面
        2. 没有页面之间的切换，就不会出现“白屏现象”
    - 单页面缺点：
        1. 首次加载耗时比较多；
        2. 不利于 SEO
        3. 不可以用导航实现前进后退效果；
        4. 页面复杂度高

潘星宇

本次lab我负责后端部分。这次是我第一次接触JAVA WEB，也是一个大型项目构建的开始。

- **前期学习：**

  在LAB布置之后，我开始了对于java web方面知识的恶补。包括
    - servlet
    - JSP
    - JDBC
    - MVC设计模式

- Spring框架
- Spring MVC
- Spring Boot

从这些过程中，我看到了 java web 发展的历程，逐渐领悟到了 IOC 依赖注入的好处，大大简化了 WEB 的开发。也领悟到了 AOP 的重要性，可以在 WEB 开发中减少大量的重复代码，提取出诸如安全验证，日志记录等通用的横切模块，使得业务逻辑的代码更加纯粹。因此选择Spring boot框架可以非常快速地建立起一个功能齐全的 WEB 项目，并且可以轻松通过导入外部包的方式引入新的功能，甚至是整合别的框架。

- **开发过程**

在本次lab的开发中，我初步认识到了前后端分离做法的好处。首先，前端开发人员可以专注于开发前端的界面，并且通过AJAX的方式请求数据。前端开发人员可以通过假设AJAX返回数据，来进行前端的开发，在实际开发的过程中可以为返回值设置缺省值来便于前端开发测试，而不必要求后端相应的请求处理已经实现完毕。其次对于后端人员来说，后端的开发可以纯粹关注在业务逻辑之上，并且和前端的平台完全分离开来。这样项目在不同平台上移植的时候，只要保证正确实现了前后端的接口，就可以一样地从后端请求数据。从而拼装出相应的 view 来进行显示。这样前后端只需要在一些重要的节点上保持一致的进度，而不需要随时保持一致的进度。

随后我认识到了前端框架相比于原生HTML, CSS, JS 的优势。首先是对大量常用的功能进行了封装，利用面向对象大大简化了代码，尤其简化了 AJAX 请求。其次可以轻易重用他人写过的代码，实现很多炫酷的功能和UI。 同样相比于PHP的前后端容易混杂，利用 java 实现后端能够很好地保证大型项目的稳定，可维护性以及可扩展性。利用框架可以轻易地针对一个对象建立一张表，只要为该对象加上@Entity 注解，并新建一个类加上@Repository 注解并继承Spring boot内已有的Repositroy，就可以轻松实现表和对象的映射。并且表与表之间的关系可以很容易通过@ManyToMany，@ManyToOne 等注解标识，非常的简单省事。

- **合作感想**

自从合作项目开始，zoom会议成为了开发日常。协同开发时，及时交流，共享屏幕，共享测试结果，可以发现问题立刻解决，不让问题堆积起来。这样在合并的时候就不会一下子出现大量的错误。同时前后端要注意接口的统一，尤其是对于AJAX请求的键值对。本次LAB还有不足的地方在于没有合理地使用远程仓库进行版本控制，之后应该养成开发过后就上传远程仓库进行版本控制的好习惯。

颜华

通过这次Lab，我们初步了解了web开发的团队协作基本模式， Vue和Spring Boot 的Java Web前后端开发框架和工具。我参与的是以Spring Boot为框架的后端开发。因为是初次用Java写网站，所以前期的准备学习时间比较长。对Spring Boot有了初步了解之后就开始一步步的进行Lab的完成。也是在整个写代码的过程中不断学习的，前期的开发过程比较艰难，尤其是与前端交互的这一方面，但是对整个框架基本熟知之后就轻松多了。总之这次Lab也算是一个快速上手Java Web和 Spring Boot不错的实践。