



浙大城市学院

ZHEJIANG UNIVERSITY CITY COLLEGE

学生实习报告

课程编号： C01098

课程名称： 移动互联网应用开发实践

学 号： 31901060

姓 名： 张亦骞

专业班级： 计算 1902

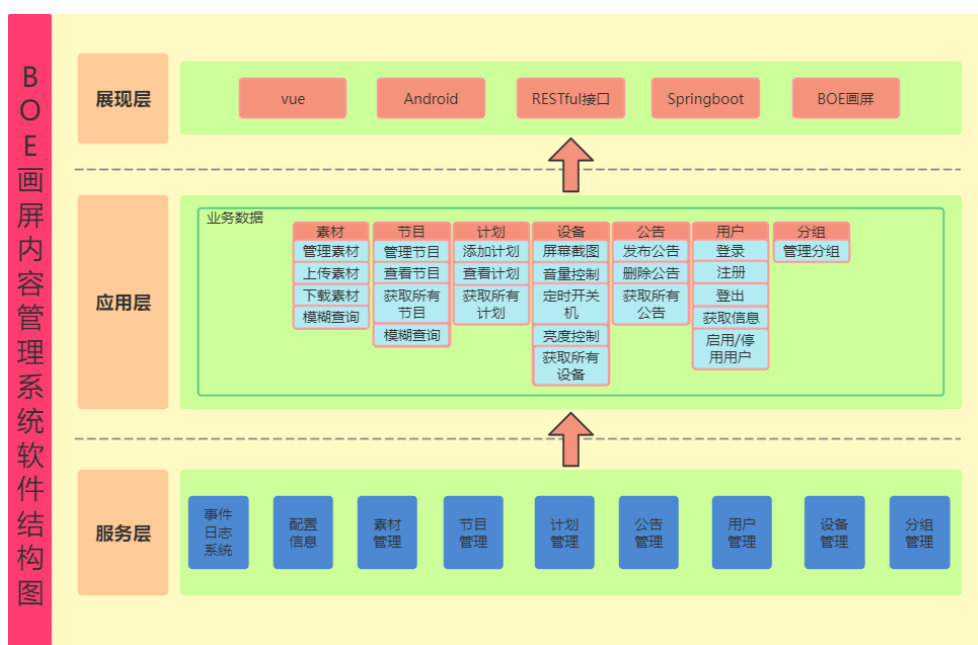
所在学院： 计算学院

报告日期： 2022 年 7 月 7 日

实习总结

这次短学期，完成了一个前后端分离的画屏内容管理系统，该系统采用 vue 端+springboot 后端（部署在服务器上）+android 端（Android 7.0）技术栈。该系统由 vue 端这一网页后台管理端执行添加素材、发布节目、添加计划、设备控制等操作，发送的指令发送给 springboot 后端，后端接收指令后，与 mysql 数据库进行数据交互、把指令发送给 android 端，android 端接收到指令后播放轮播图、视频，执行控制指令（屏幕截图、重启系统、音量控制…）等等。

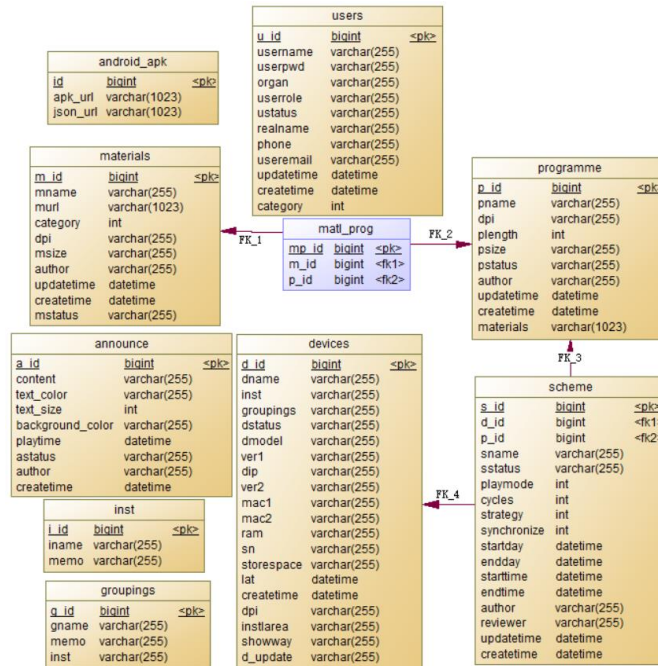
项目的架构图如下：



团队分工如下：

潘振宇	Vue 端
张泽峰	Springboot 后台：素材管理、节目管理、公告管理、首页概览【部分】
张亦骞	er 图+数据库建立；Springboot 后台：计划管理、设备管理、用户管理、首页概览【部分】
姚斯安	Android 端

项目的 ER 图如下所示：



在 springboot 中使用到的框架和工具包括：

- 1、使用 IDEA 进行 2.6.4 版本 Springboot 后端代码的编写。
- 2、使用 apifox 进行 springboot 后端接口的测试。
- 3、Sa-Token 框架。
- 4、Redis 缓存。
- 5、RabbitMQ。
- 6、使用配置实现传给前端的内容的时间格式化，配置时区
#转为 json 要到的 jackson（用于返回给前端显示）
spring.jackson.date-format=yyyy-MM-dd HH:mm:ss
spring.jackson.time-zone=GMT+8

在项目中，我遇到并解决了以下问题：

- 1、获取当前时间：Date curtime = new Date(System.currentTimeMillis());
当前时间。
- 2、一个接口传入和返回的数据采用 DTO 为宜，项目中也有采用 String 等类型的接口。
- 3、使用 Sa-Token 开源框架进行密码的 md5 加盐加密处理。
- 4、使用 redis 缓存将 android 传来的屏幕截图暂时存到缓存中，待 vue 端调用接口时，把屏幕截图传给 vue，显示在 vue 的页面上。
- 5、mysql 的分组统计：以根据机构进行设备分布的获取为例，使用 mysql 语句：
select inst,SUM(case when inst is not NULL then 1 else 0 end) 'cnt' "
+ "from devices group by inst 进行获取，其中 case when 语句表示当 inst
字段不是 NULL 的话值就是 1，否则值是 0，并遍历每条记录，把值进行累加，从而得到结果。
- 6、在 springboot 端搭建 RabbitMQ 简单模式的生产者，要做如下操作：
1) 在 MqConfig 中创建普通模式的消息队列；
2) 在 MqMessage 中写消息队列发送的消息类；

3) 在作为生产者的接口中, 写要发给 android 的消息内容。

下列是实现的功能中部分接口的关键内容【详细信息请见 Springboot 后端实现.pdf 的一~九部分】。

在 springboot 的用户管理中, 包含添加普通用户等十个接口:

1、添加普通用户

...

→查看是否已经存在该用户名和手机号

→如果存在用户, 返回错误提示信息; 若不存在, 就往数据库添加一条记录

→...

2、用户登录

...

→查找数据库判断是否存在该账号, 若存在, 判断输入的密码是否正确(把输入的密码经过 md5 加盐加密处理后和数据库的真正的密码进行比较); 否则返回错误提示信息

→若输入的密码正确, 则使用 `StpUtil.login(user.getId());` 标记当前会话用户登录

→使用 `StpUtil.getTokenInfo();` 获取当前登录用户的 token

→...

3、用户登出

...

→根据 token 间接找到用户, 执行 `StpUtil.logout();` 实现用户登出

→...

在 springboot 的计划管理中, 主要包括添加计划等接口:

1、添加计划(选择设备直接发布)

...

→根据 sname 查询数据库是否存在该计划

→若存在该计划, 则不能添加, 并添加日志信息到缓存中; 若不存在该计划, 表示可以添加, 根据 playmode 播放模式进行划分

→若 playmode=1 表示按时段播放, 则添加的计划记录包含 startday 开始日期、endday 结束日期、strattime 开始时间、endtime 结束时间、cycles 循环周期, 以及其他字段值(创建的计划的状态为“已发布”)

→若 playmode=2 表示持续播放, 则添加的计划记录不包含 startday 这五个字段的值, 其余字段值存在(创建的计划的状态为“已发布”)

→更新节目的状态为“使用中”

→获取刚刚创建的计划实体

→处理 materials 字段

→如果素材类型是视频, 就使用 CATEGORY_SCHEMEVIDEO 发送消息队列的消息; 如果素材类型是一张或多张图片, 先处理 url 以让 android 端使用访问 url, 让多个 url 中间用逗号隔开, 使用 CATEGORY_SCHEME 发送消息队列的消息

→添加日志信息到缓存中: 若计划添加成功, 就存“计划发布成功”的内容, 若添加不成功, 就存“计划发布失败”, 存到名字叫 scheme_log 的缓存中

→最后把事件日志的数量存到名字叫 cnt_log 的缓存中【事件日志的数量单独用 CntLogDto 表示, 在缓存中即名字叫 cnt_log 的缓存, 只要增加一条事件日志,

当前事件日志的下标就应该是缓存中已经有的数量+1】

→把添加的计划的计划信息填充返回的 dto，返回给 vue 端

2、获取所有计划

int index=1;作为返回的 dto 的下标

→查找数据库，获取所有状态不是“已失效”的计划放到 List<Scheme> list 中

→遍历 list 列表，构造返回的 dto（为了构造 playdate，形如 2022-06-28~2022-07-01，使用 SimpleDateFormat 类和 substring 方法

→此外还需把素材 url 加到 dto 中

→构造循环时间段 duration

在 springboot 的设备管理中，主要包括添加设备等接口：

1、控制——屏幕截图

首先，vue 端调用接口 screenshot1

→springboot 通过消息队列把提示 android 端开始截屏的消息发给 android 端

→构造返回给 vue 的 dto

→接着，android 端调用接口 screenshot2，传入 MultipartFile 格式的截屏文件

→把截屏上传到 MinIO 上（使用其他组员的上传 MinIO 的方法）

→把截屏的 url 和文件名称（非完整 url）存到缓存中

→最后，vue 端调用接口 screenshot3，传入获取截屏的指令（一个 String 类型的字符串）

→从缓存中获取截屏信息，构造返回给 vue 的 dto

2、控制——定时开关机

vue 端传入设定的开关机的 starttime 开机时间、endtime 关机时间和 cycles 循环周期

→通过消息队列把这些数据传给 android 端，android 端通过消息队列收到后执行

→构造返回给 vue 的 dto

3、控制——系统升级

vue 端调用接口

→通过消息队列把系统升级的消息传给 android 端，android 端通过消息队列收到后执行

→构造返回给 vue 的 dto

其中，这些控制接口除系统升级和屏幕截图外，其他接口的思路是类似的。

4、设备上下线状态

使用静态全局变量 index_log 定义事件日志在缓存中的下标：

→获取传入的设备当前状态：off（离线）、relax（空闲）、playing（播放），分别更新数据库中设备的状态

→获取当前使用的设备（因为设定一台设备在使用该系统，故其在数据库中的下标为 1）

→把设备上下线状态更新的操作存到缓存中

→更新缓存中的 cnt 【事件日志的数量单独用 CntLogDto 表示，在缓存中即名字叫 cnt_log 的缓存，只要增加一条事件日志，当前事件日志的下标就应该是缓存中已经有的数量+1】

在 springboot 的首页部分【部分】中，主要实现了获取计划数量、获取设备数量、获取设备状态、获取设备分布、获取事件日志列表等功能，这些功能体现在 vue 端的首页。在分组管理中，实现了分组的添加、编辑、删除和获取所有分组。

通过这次短学期的项目的开发，我对前后端分离系统有了更深的理解与认识，并且对 springboot 后端开发的流程更加熟悉了，也让我认识到前后端分离系统中，前后端接口匹配的重要性，一个后端接口路径中带参数，还是在请求体 body 中带参数，对于代码的编写有很大的影响与不同。这次十五天的短学期的项目实训，让我对团队项目的协调有了更深的理解，团队分工是明确且是均等合理的。