

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： 31901056

姓 名： 姚斯安

专业班级： 计算1902

所在学院： 计算学院

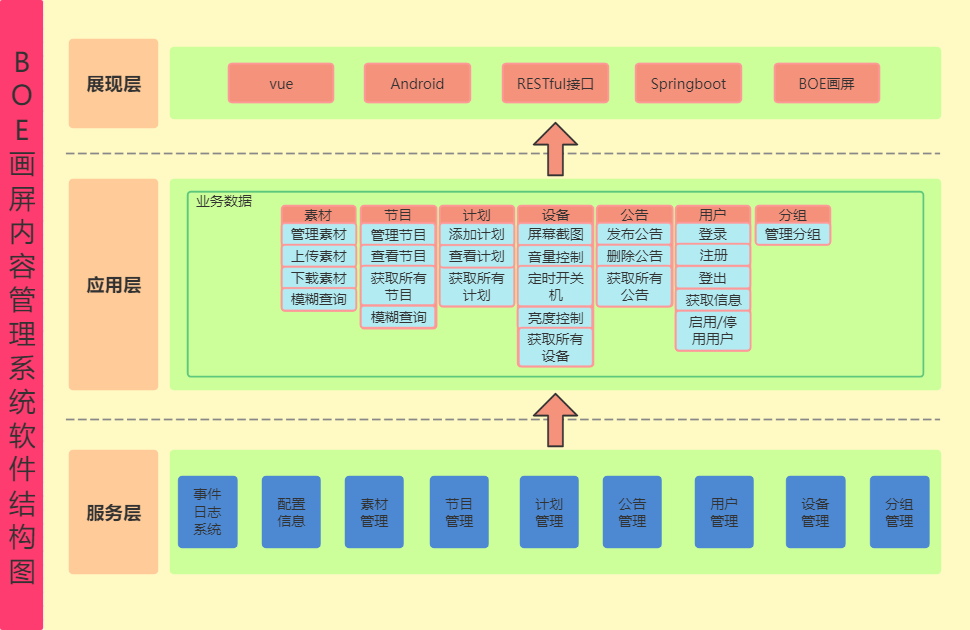
报告日期： 2022 年 7 月 7 日

|  |  |  |  |
| --- | --- | --- | --- |
| 实习时间 | 2022 －2023 学年 □ 秋季学期 □ 春季学期 暑期短学期  自 2022年 6 月 23 日 至 2022年7 月 7日 | | |
| 实习类型 | □ 认识（社会调查）实习  教学（课程、专题、金工）实习  □ 毕业（综合）实习 | 实习地点 | 校内 |
| 实习单位评语： | | | |
| 实习单位（盖章）：  年 月 日 | | | |
| 实习指导（带队）教师评语： | | | |
| 实习总评成绩：  教师（签名）：  年 月 日 | | | |

注：学生的实习总结等文档附在本封面之后

**实验报告**

本次短学期要求我们实现一个系统采用vue端+springboot后端（部署在服务器上）+android端（Android 7.0）的前后端分离的画屏内容管理系统，该系统包括但不限于素材、节目、计划、设备控制等功能。我在小组中负责Android移动端的实现测试，功能包括音量控制、屏幕亮度控制、截屏、定时开关机、播放节目等。



实现的具体功能：

1、截屏

使用bitmap将本机截屏存储到摸一个路径下，可以使用Context.getFilesDir()来获取到软件的根目录，并且需要保存到images文件下方。传入指定的存储路径，然后判断是否存在，如果存在需要创建指定目录，然后返回创建结果，创建一个FileOutputStream 来用于图片的写入，并且需要对图片进行相对应的压缩操作。截屏存入本地后使用okhttp客户端调用后端接口将该截图上传给服务器发送给web端（具体okhttp是什么在后面技术说明中会说明）。

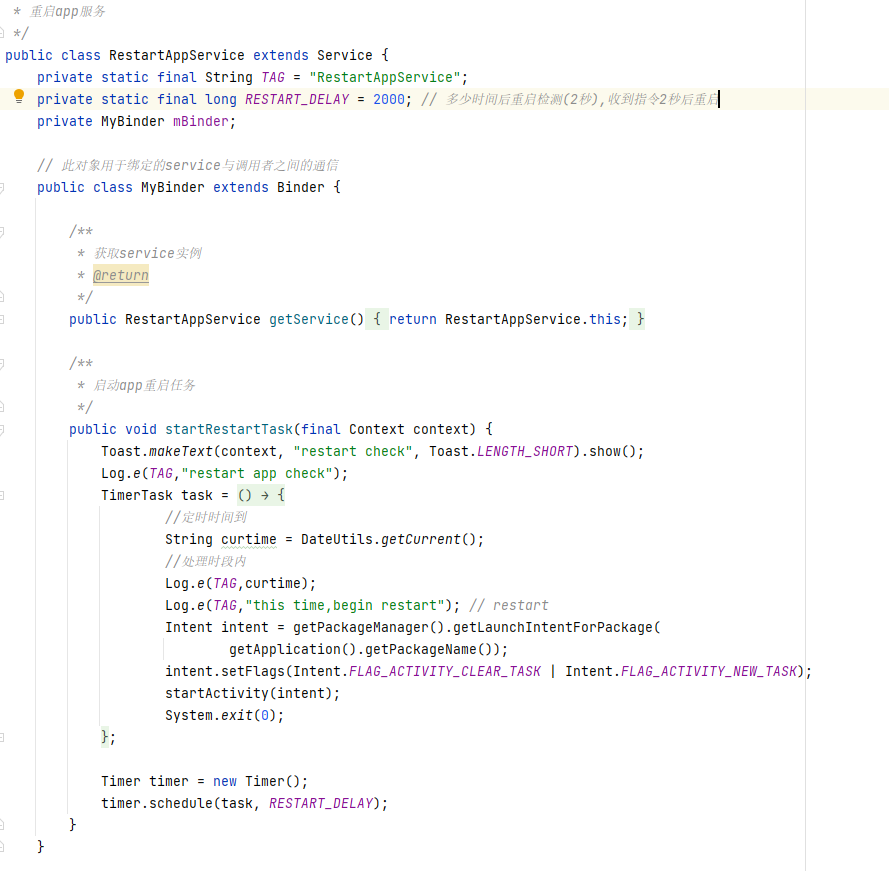
2、控制音量、屏幕亮度

音量与屏幕亮度的控制大同小异，都是消息队列接收到控制请求后调用系统函数将app的系统参数调整为发送过来的特定值，比如控制屏幕亮度使用window.setAttributes函数，控制音量使用setStreamVolume函数。

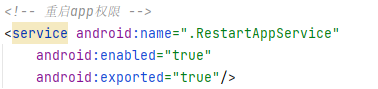
3、重启app

Android的Service实现后台定时检测并重启应用，Service(服务)是一个一种可以在后台执行长时间运行操作而没有用户界面的应用组件。服务可由其他应用组件启动（如Activity），服务一旦被启动将在后台一直运行，即使启动服务的组件（Activity）已销毁也不受影响。 此外，组件可以绑定到服务，以与之进行交互，甚至是执行进程间通信 (IPC)。 例如，服务可以处理网络事务、播放音乐，执行文件 I/O 或与内容提供程序交互，而所有这一切均可在后台进行。

RestartAppService 服务类继承自Service：



在AndroidManifest.xml中声明：

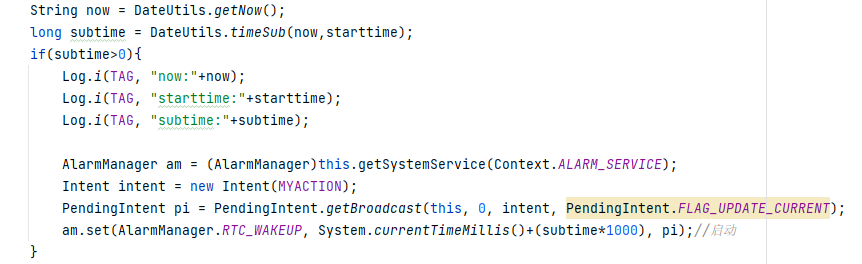


android:exported 表示是否允许除了当前程序之外的其他程序访问这个服务

android:enabled 表示是否启用这个服务

4、定时开关机

接收到定时开关机请求后将发生过来的时间格式化，使用AlarmManager搭配Receiver可以实现定时自动启动应用程序：



使用Calendar日历与Timer延时搭配可以简单实现定时关机

5、轮播图

接收到播放请求后处理发送过来的图片url、轮播开始时间、结束时间等一系列数据，创建线程在播放之前下载url中的图片，使用Timer定时器类实现在规定时间开始与关闭轮播。轮播采用ViewFlipper组件动态导入下载在本地的web端发来的图片。开始轮播时将桌面imageview隐藏，由于视频组件优先级高于轮播，因此不隐藏视频组件，以便于两个计划同时出现时视频在上面播放，播放玩后隐藏视频组件继续播放轮播。关闭轮播时查看是否在播放视频，如果不在播放视频则显示桌面。

6、公告

处理文字等信息，使用Timer定时器实现定时播放，实现公告的文字组件采用自编组件AutoScrollTextView实现文字的滚动效果，如果是安卓自带的滚动效果则是需要文字长度大于当前页面才会滚动，无法实现几个字的滚动效果。



7、播放视频

视频播放与轮播图实现逻辑基本一致，不同在于图片轮播是将图片下载到本地轮播，因为考虑到视频的体量远远大于图片，下载会消耗大量资源，所以视频是直接利用缓存播放网络视频。

8、自动更新

Web端发来自动更新请求后，首先查看权限，授予权限后检查路径中的json文件的app版本字段与当前版本比较，若当前版本小于最小版本则开始更新，把检查更新，下载apk，安装apk等操作封装成一个类AutoUpdater。

实现这些功能的过程中采用的技术：

1、rabbitmq消息队列实现前后端通信

RabbitMQ是流行的开源消息队列系统，是AMQP（Advanced Message Queuing Protocol高级消息队列协议）的标准实现，用erlang语言开发。RabbitMQ具有良好的性能和时效性，同时还能够非常好的支持集群和负载部署，非常适合在较大规模的分布式系统中使用。

使用消息队列纯粹是因为一开始没有想到websocket的使用，但是消息队列也有优势，比如设备离线时消息队列中有消息等上线后可以立即处理，有助于实现设备离线时发送计划，上线后能立即开始处理计划，也方便拥有多台设备时对设备的单独控制管理。

2、okhttp实现移动端调用后端接口

okhttp是一个处理网络请求的开源项目,是安卓端最火热的轻量级框架, 用于替代HttpUrlConnection和Apache HttpClient(android API23 6.0里已移除HttpClient,现在已经打不出来)它的主要优势在于：

允许连接到同一个主机地址的所有请求,提高请求效率

共享Socket,减少对服务器的请求次数

通过连接池,减少了请求延迟

缓存响应数据来减少重复的网络请求

减少了对数据流量的消耗

自动处理GZip压缩

此项目采用okhttp实现Android移动端调用springboot后端接口上传截图、获取数据库数据用于更新app以及向web端发送设备状态。

经过本次为期15天的短学期，我与三位同学共同合作完成了画屏内容管理系统，由于其他项目我们基本都有合作的经验，所以这次合作分工对我们来说还是较为明确合理的，明确分工之后我们也很快投入到了各自的任务之中，这个项目对我们理解团队合作又有了新的提升，我也越发体会到了团队合作的重要性，我也希望日后的学习工作中可以更多地进行有效的合作沟通。对我个人而言，这次短学期实质上是对android移动应用开发的一次复习，但是在这个过程中，减少了界面设计，增大了操作处理，我也乐在其中，与其花时间在不重要的ui设计上，我更愿意了解新的技术并加以应用，调试的过程是痛苦的，但是当一个个功能真真实实地实现的时候心里的高兴也是无与伦比的。