

## Android 移动应用开发期末项目小组报告

# Tarlesh

组号：15

小组成员：

梁颖霖 16340132 806179953@qq.com 806179953

梁育诚 16340133

梁庭 16340131

余崇斌 13331217

## 目录

1. 概述.....	3
2. 背景.....	3
3. 项目说明.....	3
4. 应用效果.....	9
5. 开发过程中遇到的问题及解决办法.....	14
7. 小组分工.....	17
8. 参考资料.....	17

## 1. 概述

Tarlesh是一个针对大学生的学习分享平台。该平台提供给在校学生学习要用到的课件、习题、试卷、学习资料等，同时还有讨论、提供等论坛功能。它支持多种不同类型的文件上传与下载，提供pdf预览等功能。使用了奖励机制，保证资料的质量，是当代大学生学习的必备神器。

以上就是我们对这个项目的概述了。

## 2. 背景

1. 实用性：处于信息爆炸的时代，面对大量的学习资料，怎样才能找到最适合自己的、最有用的学习资料成为一大难题。学生希望快速找到自己所需要的科目的相关资料。Tarlesh提供给大学生一个平台去分享学习资料，因为学生之间学习的课程是类似的，这大大减少了同学们搜索资料的难度。同时该平台提供的讨论社区也能够很好地帮助到同学们的学习。
2. 创新性：同款的产品在市场上基本没有，我们希望打造的是一个面向学生的平台。历届的优质学习资料得以长期保存与分享，这正是我们想做的。
3. 用户体验：简洁明了的用户界面，支持本地文件上传与下载，并对PDF文件实现应用内的在线预览，非常适合学生使用。
4. 高频：大学生的学习是长期存在的，每一个阶段都有该阶段所需要的学习资料，因此保证了Tarlesh的高频性。
5. 粘性：使用奖励机制，鼓励同学上传优质资源，积极回答问题，努力形成一个具有良好氛围的学习社区。
6. 本项目的开发所用到的API全都是自己编写，也保证了信息数据的安全。学习资料的分享支持多种格式，对最常用的pdf文件支持在线预览。

## 3. 项目说明

### 3.1 通过功能模块图，说明系统功能需求。

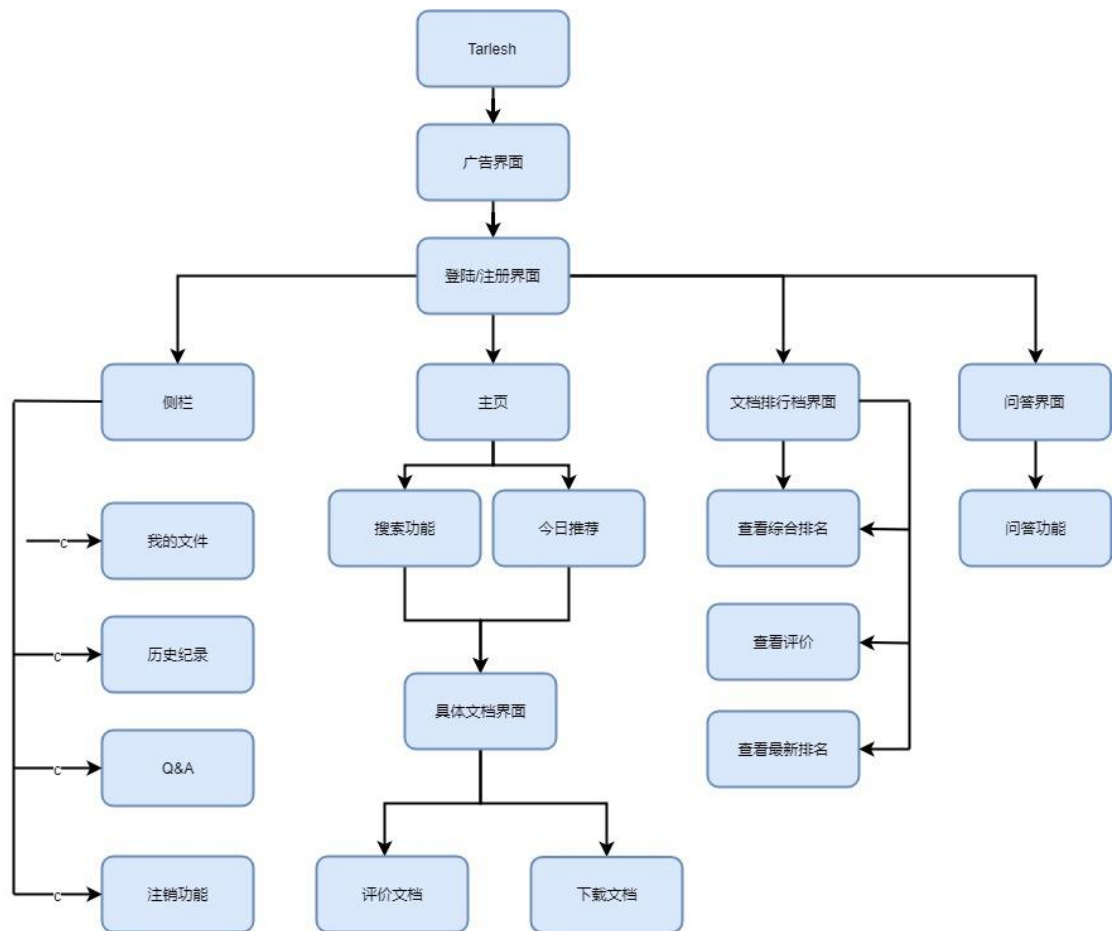


图1：功能模块图

根据功能模块图可以看出，整个应用的功能可以分为两部分：文件与问答。文件功能是我们的重点，包括了上传、下载、推荐、搜索、在线预览等众多个性化功能。问答功能则是比较传统的Q&A模式。

### 3.2逻辑流程

以下是我们应用的逻辑流程图，展示用户使用我们应用时候的操作逻辑。其中图2为总应用的逻辑流程。

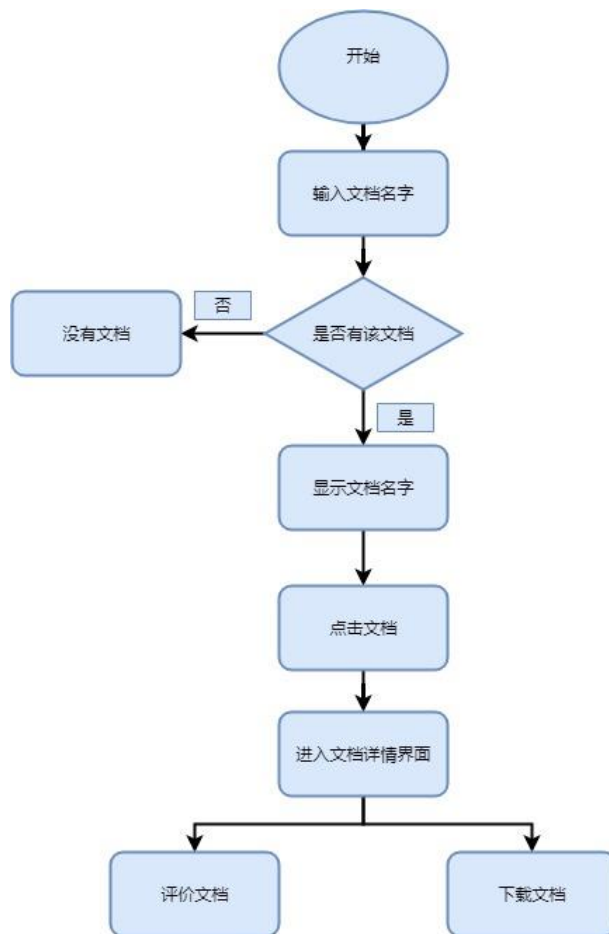


图2：文档逻辑流程图

### 3.3技术说明

#### 1. 登录注册部分

功能：提供给用户登录与注册功能。

技术说明：

1. 自定义一个类CommonSPUtil存储用户token信息，以记录登录状态和方便后续的API访问。

#### 2.主页页面部分

功能：包括文件上传功能、学习资源推荐、侧栏

技术说明：

1. **Material drawer**，开源组件，用于构建侧栏。
2. 底部导航栏：**Android Studio**自带的导航功能，我上网搜索一些适合的icon进行修改，主要添加了跳转逻辑以及将活动生命周期进行调整。
3. **Intent**的选择文件上传功能：这里选择文件需要**Intent.CATEGORY\_OPENABLE**进行访问，并开启一个新的活动。
4. 输入框使用的是**MyAlertDialog**。**Github**开源项目，更加美观

### 3.排行页面部分

功能：按照文件评分、下载量、时间分别排序

技术说明：

1. 在后端数据库中先对文件进行排序，再通过API返回，这样减少了在前端的计算量。
2. 使用RecyclerView与Adpater适配展示数据列表。

### 4. 问答页面部分

功能：提供用户提问功能，以及回答该问题功能（暂未实现）

技术说明：

1. 主要使用了API网络请求，用户的提问及时更新渲染。
2. 页面渲染。使用CardView来布局，使页面更美观。

### 5. 文件详情页面部分

功能：显示文件的详细信息，包括文件名、文件介绍、评分、创建人、下载量。支持用户对该文件进行评分。支持用户下载该文件到本地，对pdf文件提供在线预览服务。

技术说明：

1. 评分用到了RatingBar，使用星星的评分条给用户进行评分。评分时在服务器确认用户是否重复评分，评分后评分条禁止修改，并及时更新该文件评分。
2. 下载功能。下载时服务器返回的是编码的Buffer类型数据。转化为byte[]后直接解码，根据文件名使用OutputStream写成文件存放到SD卡中。

3. 预览功能。这个是最有技术含量的功能。在下载文件后，我们支持pdf预览。Android并没有提供原生的pdf预览功能。我们可以使用google或mozilla的文档服务，但是他们都有明显的确定。谷歌的文档服务在国内不能使用，而且网络服务对于本地应用来说也不稳定。Mozilla的服务需要下载一个大约为9M的js包，这显然增大了整个应用的体积。因此这里我对Mozilla的方案做了一个简化，自己编写了html，使用pdf.js进行渲染，这样整体增加的大小不超过3M。思路是，使用webview做页内的显示，然后再webview中打开已经写好的html，html中调用pdf.js进行渲染。最终效果是非常不错的。

## 6. 服务器部分

功能：提供远程服务器数据存储功能

技术说明：

1. SQL数据库，使用SQL语句进行增删查改、事务等操作。

## 7. 广告页面

功能：显示广告，有三秒跳转功能。

技术说明：Handler修改UI页面与多线程：这里是模拟应用市场大多数应用的广告跳转页面，点击右上角的跳过可以直接进入应用，否则倒计时三秒后进入应用。

每隔一秒向Handler发送消息，如果计数已经到达三秒则直接进入。

### 3.4代码架构

adapter是作为相应的容器管理类:

MyFileAdapter. java,  
MyQuizAdapter. java,  
MyViewHolder. java,  
RecommendAdapter. java

View里面的类是相关页面的显示管理, 和逻辑类:

FileDetail. java,  
HomeActivity. java  
MainActivity. java,  
QuizActivity. java,  
RankingActivity. java,  
WelcomeActivity. java,  
LoginActivity. java,  
RegisterActivity. java

实体类:

Comment. java  
File. java  
Quiz. java

接受API返回数据类:

CommentResponseBody. java  
FileDownloadResponse. java  
QuizPostResponseBody. java  
QuizResponseBody. java



ResponseBody.java

UploadResponseBody.java

#### 4. 应用效果



打开应用，广告页面



登录界面



注册界面



主页面



侧栏显示



选择文件上传



添加文件描述



成功上传文件



文件排行页面（综合排序）



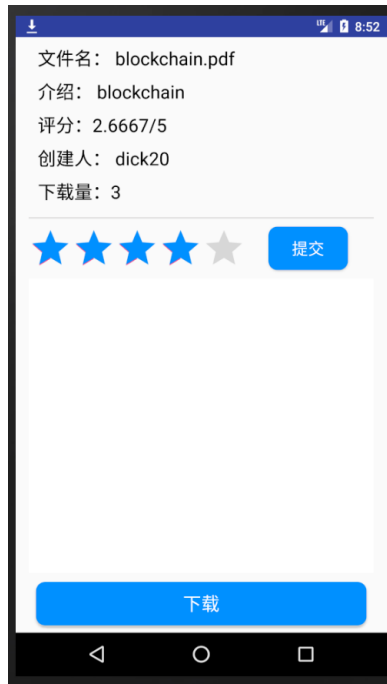
文件排行（下载量排序）



推荐文件改变



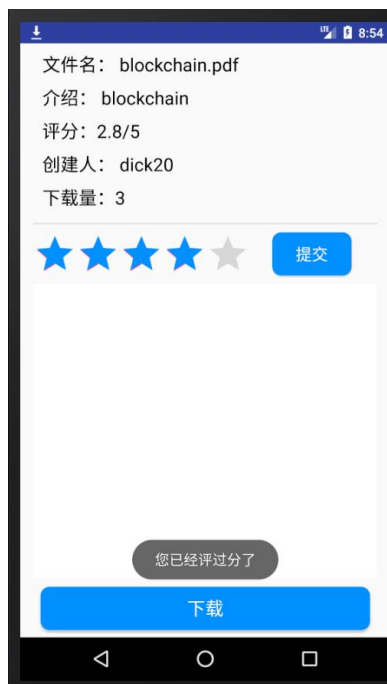
进入文件详情页面



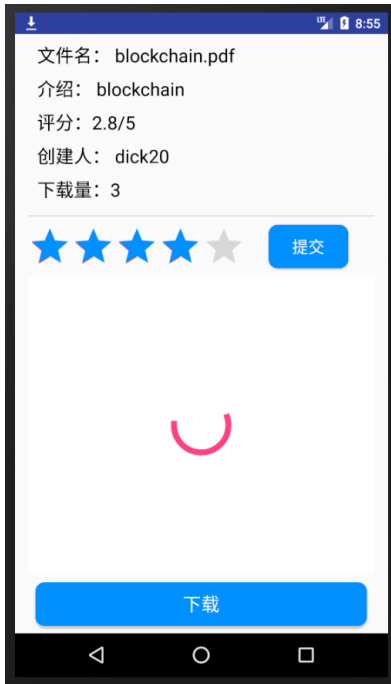
点击评分条



点击提交评分



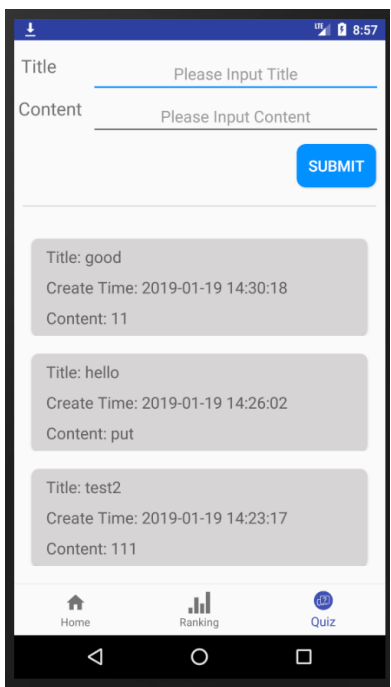
提示重复评分



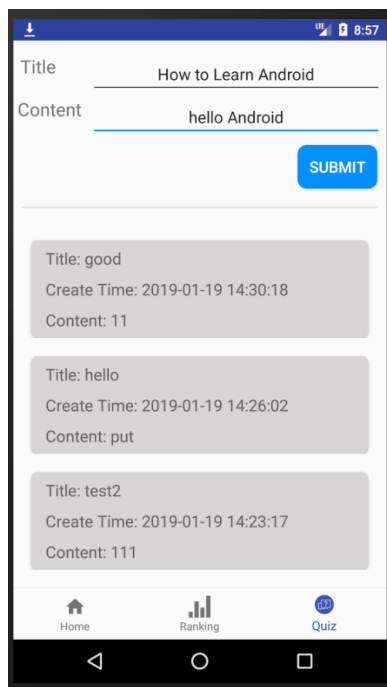
点击下载按钮，出现进度条



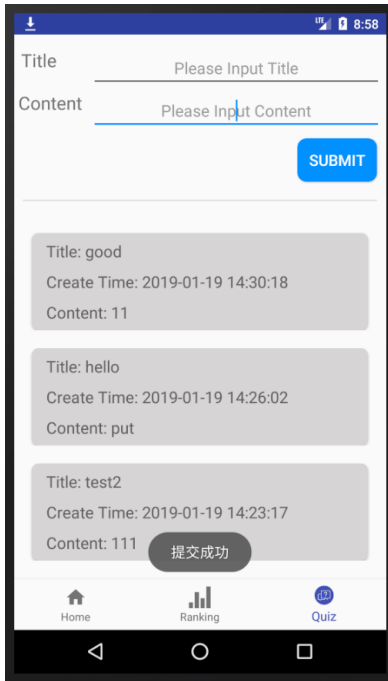
下载完成，页面内自动预览



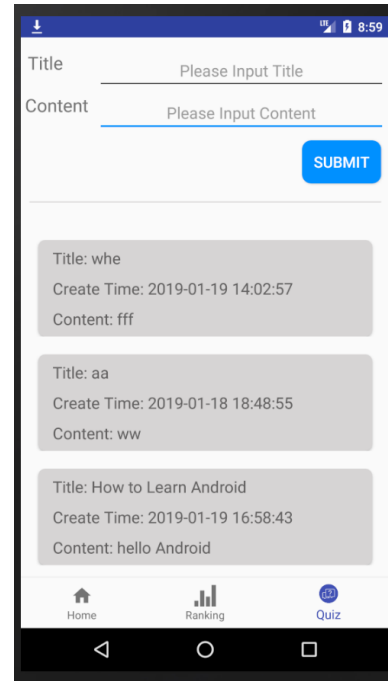
进入问答页面



填写问题信息



提交问题



拖到最下方看到新提交的问题

## 5. 开发过程中遇到的问题及解决办法

### 问题一： 用户对同一文件重复评分的问题

**解决方法：**首先是对RatingBar修改，在用户提交评分后，设置RatingBar的IsInditcator属性，禁止用户再次修改。当用户再次打开该文件页面并评分，在服务器端进行判断，如果重复评分则返回一个提示，在前端提示用户已经评过分了。

### 问题二： pdf预览解决方案

**解决方法：**首先是确定使用哪种方式。使用谷歌文档服务在国内并不适用，使用Mozilla的在线服务或者下载到本地都会有一定的问题。后来想到，既然核心功能是pdf.js，那么其他的部分自己写好了。简单的写了一个html来调用pdf.js，用的是国内的cdn，保证了预览的速度。然后是遇到了跨域的问题，查看了许多资料后，要对websetting进行设置，最终实现了页面内的在线预览。

### 问题三： API约束问题

**解决方法：**前后端负责的组员多沟通，尽量将每个API写成规范的文档来使用，嫩能够避免误会。在写进应用前可以用Postman多测试，确定成功后在写到应用中。

#### 问题四： Material Drawer布局遮挡问题

**解决方法：** Action Bar卡住视线，挡在侧栏的前面。直接去掉这个action bar，然后自己加上一个toolbar就可以有一样的效果了。

#### 问题五： 文件URI转文件真实路径

**解决方法：** 根据之前作业的做法，选择文件后，我只能获得文件的uri。而我传递到api需要是一个java的File类，所以这里的难题在于如何通过uri转化成文件的真实路径。这里使用到了actualimagecursor来获取，该方法参考了网上的做法。解决这个问题尝试了很多种方法，也了解到了更多关于android访问文件的知识，关于文件路径之间转化的关系。

#### 问题六： 如何使用Retrofit2上传文件

**解决方法：** 首先，我是先使用了postman测试了该api，传递的参数包括一个file类型一个text类型的描述。但是到了android里面并没有找到相应的方法。首先我尝试的还是使用requestbody简单的将file与描述封装成json传送到服务器，结果服务器无解析。然后，又试着使用@Field来解决这个问题，结果还是一样，无法解释传送的数据。

接着查询了许多资料，如何使用retrofit上传，功夫不负有心人。这里使用的是其中的一个方法，利用@MultiPart来进行传输。将文件与描述分成不同的part，使用不同的码转换传输。

#### 问题七： 关于调用api返回参数的问题

**解决方法：** 在使用一些api的时候，由于与后台沟通失误，有时返回的数据为数组，而我在接受的时候仅以变量接受，这样导致上传错误等一系列问题。经过查询很多网上博客都没有解决办法，最终是通过仔细查看postman返回结果才找到问题所在。

这里我通过retrofit接受的返回值都是其api返回参数所一一对应的，这样才能保证传输的正确性。

#### 问题八： 存储用户token到本地

**解决方法：**使用CommonSPUtil在应用中保存token，以便该用户在后续的操作中能够使用token去调用API。

## 6. 思考及感想

从应用的角度来说，这次我们选择了开发Tarlesh这个项目也是因为自己在平时的学习过程中遇到了找资料的问题。目前的方法一般都是问以前的师兄师姐拿资料，而这些资料往往比较散乱，而且不全面。于是我们就想做一个学习资料的分享平台。而网络访问是这个应用很主要的一个部分，相比起使用别人的API，我们决定用自己的服务器，自己写API。这样的难度和工作量都是大大增加的，但是这样对于资料的管理就更加有我们自己的特色。在前期的时候遇到了各种困难，有一些是技术上的，有一些则是方法上的。因为有很多部分都是课程中没有涉及到的，所以我们一开始也不知道怎么去实现。经过大量的讨论和查找资料，逐步尝试之后最终才做成了这个应用。整体的效果还是令人满意的，在较短时间内我们预期设想的几个核心功能都很好地实现了。当然还存在很大的进步空间，比如UI的设计、动画的添加、还有一些小功能的扩展等。

从小组合作的角度看，经过了一个学期的学习，以及期中项目的团队开发，我们对于协作开发的模式有了更加深入的了解。已经能够熟练地运用gitee对项目进行开发的代码管理。在前后端对接上，使用了写API文档的方式来表明接口，减少了很多不必要的麻烦。大家的工作效率也很高，基本都能按时按质按量地完成任务。

总而言之，这次期末项目的效果我们是满意的，从中我们不但学会到了安卓应用的知识，更加重要是学会到了团队合作开发。



## 7. 小组分工

组员	学号	负责内容	贡献百分比 (%)
梁颖霖	16340132	广告页面、主页面、文件排行、文件上传	25
梁育诚	16340133	文件详情页面、文件下载、文件预览、问答页面	25
梁庭	16340131	后台数据库、服务器API、主页侧栏	25
余崇斌	13331217	负责登录注册界面	25

## 仓库代码统计:



## 8. 参考资料

- bytes数组转pdf文件: <https://stackoverflow.com/questions/8644459/how-to-convert-byte-array-to-pdf-file-in-android>
- webview跨域访问文件解决:  
<https://blog.csdn.net/yclfdn2004/article/details/51364660>
- webview的基本使用:

[https://blog.csdn.net/lowprofile\\_coding/article/details/77928614](https://blog.csdn.net/lowprofile_coding/article/details/77928614)

4. 修改RatingBar样式:

<https://blog.csdn.net/yaochangliang159/article/details/67637185>

5. PDF预览: <https://blog.csdn.net/DeMonliuhui/article/details/81185611>

6. github API: <https://developer.github.com/v3/>

7. material: <https://github.com/mikepenz/MaterialDrawer>

8. postman: <https://www.getpostman.com/>

9. token使用: <https://www.ctolib.com/zwStar-token-login.html>

10. 上传文件: <https://www.jianshu.com/p/b6ffab850d35>

11. Android Uri, Path与File、Bitmap的相互转换:

[https://blog.csdn.net/weixin\\_37577039/article/details/79242455](https://blog.csdn.net/weixin_37577039/article/details/79242455)

12. Android Uri获取真实路径以及文件名的方法:

<https://www.cnblogs.com/jooy/p/9134020.html>

13. Retrofit 2.0 超能实践（三），轻松实现多文件/图片上传/Json字符串/表单:

<https://www.jianshu.com/p/acfefb0a204f>