**目录**

[1.前言 1](#_Toc186)

[1.1项目背景及意义 1](#_Toc10235)

[1.2项目所用技术及说明 1](#_Toc4474)

[1.2.1 OKHttpClient 1](#_Toc32233)

[1.2.2 Glide 1](#_Toc10701)

[1.2.3 PhotoView 2](#_Toc20157)

[1.2.3 Fastjson 2](#_Toc12679)

[2.概要设计 3](#_Toc29190)

[2.1需求描述 3](#_Toc7396)

[2.2整体开发设计结构 3](#_Toc8814)

[2.2.1 Android端开发 3](#_Toc24017)

[3.详细设计 4](#_Toc27733)

[3.1界面设计 4](#_Toc20660)

[3.1.1 UI设计 5](#_Toc15873)

[3.1.2页面交互设计 8](#_Toc2126)

[3.2业务逻辑设计 8](#_Toc3709)

[3.2.1注册、登录业务 9](#_Toc14747)

[3.2.2列表显示业务 9](#_Toc30828)

[3.2.3列表更新业务 10](#_Toc20459)

[3.2.4异步处理中修改UI 12](#_Toc20555)

[4.软件测试 12](#_Toc12932)

[4.3拉取共享照片模块： 13](#_Toc14846)

[5.心得体会 15](#_Toc20619)

[5.1熟悉安卓开发基本流程： 15](#_Toc8099)

[5.2熟悉多人开发和Git的使用： 15](#_Toc27916)

[6. 参考文献 17](#_Toc15347)

**指定选题列表**

**以下部分为指导老师指定的选题，每个选题限4个团队进行选择：**

A.基于Android的课堂签到系统。该系统分为教师端及学生端，用户通过注册后，可通过二维码或输入口令等形式进行签到，签到时需要记录用户的位置信息；教师端可统计签到信息或查询签到信息；

B.基于Android的共享车位系统。该系统可为用户提供私家车位共享服务，用户可发布或预定共享的私家车位，对私家车位提供导航定位服务；用户可搜索指定区域周边的共享私家车位。

C.基于Android的图片分享软件。用户可将图片分享至平台以供其他用户浏览，用户可对喜欢的图片进行点赞保存分享等操作。

D.基于Android的二手物品交易系统。用户可通过本系统进行二手物品交易。用户可发布或浏览系统中的二手物品交易信息。

本小组选择的题目为题目C，即**基于Android的图片分享软件。**

**摘要**

本项目（易图）是基于Android的图片分享平台，实现了从指定API上灵活接收数据并进行图片的评论、点赞、关注等多种在线交互功能。主要采用的框架有：OKHttpClient(一个优秀的网络请求框架，用于Android开发的轻量级HTTP客户端，性能高效，API封装友好，对于本项目的网络请求处理需求具有极大的适配度)、Glide(一个快速高效的Android图片加载库，注重于平滑的滚动。它提供了易用的API，高性能、可扩展的图片解码管道，以及自动的资源池技术)、PhotoView(一个常用的图片预览控件，主要用于Android应用中大图查看，图片手势缩放，旋转，平移，合理使用能够营造良好的用户体验）。本项目的特点是界面美观大方、操作功能简单明了、运行流畅快速、数据实时刷新、功能实用易懂。

**关键词**：OKHttpClient; Glide; PhotoView; OpenAPI; Git

**1.前言**

**1.1项目背景及意义**

随着手机硬件设备的迅速发展，人们已经变得更乐于使用手机来拍摄照片；伴着互联网技术的飞速普及，人们已经习惯社交软件上的交流。本项目基于大互联网、移动化时代，旨在为用户提供一个可以展示自己所拥有图片的舞台，以及一个多姿多彩的、基于图片分享的社交空间；丰富用户的图片生活、社交生活，以图片分享引领社交新风尚。

**1.2项目所用技术及说明**

**1.2.1 OKHttpClient**

HTTP是现代应用常用的一种交换数据和媒体的网络方式，高效地使用HTTP能让资源加载更快，节省带宽。OkHttpClient是一个高效的HTTP客户端，它有以下默认特性：

①支持HTTP/2，允许所有同一个主机地址的请求共享同一个socket连接；

②连接池减少请求延时；

③透明的GZIP压缩减少响应数据的大小；

④缓存响应内容，避免一些完全重复的请求；

**1.2.2 Glide**

Glide是一个快速高效的Android图片加载库，注重于平滑的滚动。Glide提供了易用的API，高性能、可扩展的图片解码管道（decode pipeline），以及自动的资源池技术。Glide 支持拉取，解码和展示视频快照，图片，和GIF动画。Glide的Api非常灵活，非常适合本项目使用。默认情况下，Glide使用的是一个定制化的基于HttpUrlConnection的栈，但同时也提供了与Google Volley和Square OkHttp快速集成的工具库。虽然Glide 的主要目标是让任何形式的图片列表的滚动尽可能地变得更快、更平滑，但实际上，Glide几乎能满足你对远程图片的拉取/缩放/显示的一切需求。

①API：Glide 使用简明的流式语法API，这是一个非常棒的设计，因为它允许你在大部分情况下一行代码搞定需求：

Glide.with(fragment)

.load(url)

.into(imageView);

②性能：Glide 充分考虑了Android图片加载性能的两个关键方面：图片解码速度和解码图片带来的资源压力。为了让用户拥有良好的App使用体验，图片不仅要快速加载，而且还不能因为过多的主线程I/O或频繁的垃圾回收导致页面的闪烁和抖动现象。Glide使用了多个步骤来确保在Android上加载图片尽可能的快速和平滑：

自动、智能地下采样(downsampling)和缓存(caching)，以最小化存储开销和解码次数；积极的资源重用，例如字节数组和Bitmap，以最小化昂贵的垃圾回收和堆碎片影响；深度的生命周期集成，以确保仅优先处理活跃的Fragment和Activity的请求，并有利于应用在必要时释放资源以避免在后台时被杀掉。

**1.2.3 PhotoView**

PhotoView旨在帮助生成一个易于使用的缩放安卓图像视图，它简单易用，具有如下特性：

①开箱即用的缩放，使用多点触控和双击。

②滚动：具有平滑的滚动甩动。

③在滚动父级（如 ViewPager）中使用时，效果完美。

④允许在显示的矩阵发生更改时通知应用程序。当您需要根据当前缩放/滚动位置更新 UI 时非常有用。

⑤允许在用户点击照片时通知应用程序。

**1.2.3 Fastjson**

Fastjson是一个 Java 库，可用于将 Java 对象转换为其 JSON 表示形式。它还可用于将 JSON 字符串转换为等效的 Java 对象。Fastjson 可以使用任意 Java 对象，包括您没有源代码的预先存在的对象。‎Fastjson有适合本项目的以下优点：

①在安卓客户端上提供最佳性能‎。

‎ ②提供简单的toJSONString( )和parseObject( )方法来将Java对象转换为JSON，反之亦然。

‎ ③允许预先存在的不可修改对象与 JSON 相互转换。

④‎对 Java 泛型的广泛支持‎。

⑤‎允许对象的自定义表示。

1. ‎支持任意复杂的对象（具有深度继承层次结构和广泛使用泛型类型）‎。

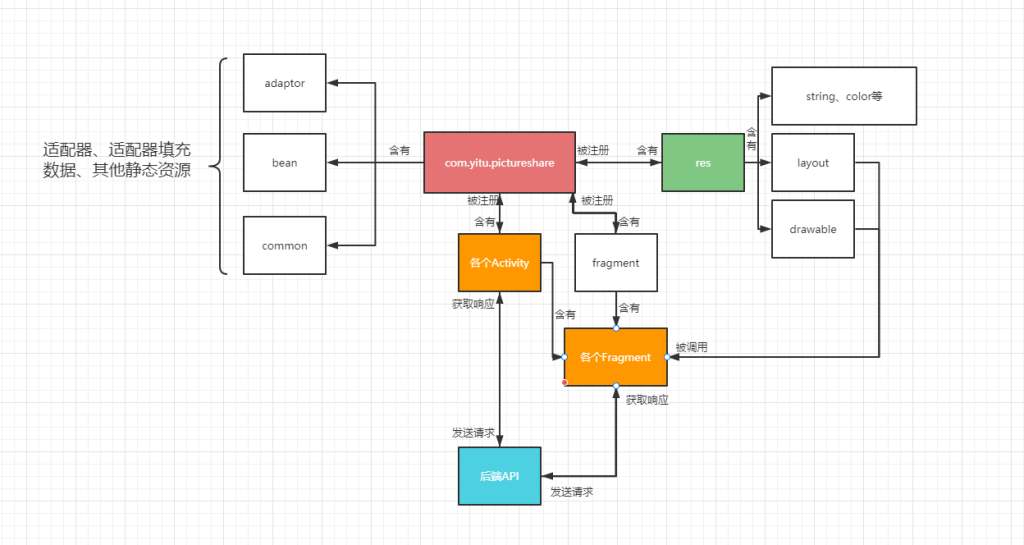
**2.概要设计**

**2.1需求描述**

本项目的需求总体是图片分享平台，具体实现为用户注册及登录、图片上传、图片动态分享、图片动态删除、图片动态点赞、图片动态评论、关注用户、获取本用户发布的动态、获取本用户关注的用户发布的动态。

**2.2整体开发设计结构**

本项目采用Android端开发＋后端API开发的模式，即在Android端编写页面UI及用户操作、向服务器发送请求、处理服务器响应，在后端API实现请求的处理及结果响应。如下图所示。



**2.2.1 Android端开发**

**①UI页面设计**：本项目的页面（res/layout中的资源）总共有11个Activity（活动）、2个Fragment（碎片），分别为：

1. activity\_login：用户登录页面；
2. activity\_register：用户注册界面；
3. activity\_index：主页界面，承载fragment\_home和fragment\_profile以及底部导航栏；
4. activity\_share：用户上传分享图片界面；
5. activity\_share\_info：用户获取分享详情界面；
6. activity\_my\_picture：用户自身分享图片列表界面；
7. activity\_focus：用户关注图片列表界面；
8. activity\_setting：用户设置后端API授权的AppId和AppSecret界面；
9. activity\_show\_picture：用户浏览大图界面；
10. activity\_version：版本信息界面；
11. activity\_about：开发信息界面；
12. fragment\_home：用户浏览公开资源图片动态界面；
13. fragment\_profile：用户进行设置、查看自身分享和关注用户的图片列表、退出登录界面；

**②用户操作及服务器响应:**

本项目中，每个Activity和每个Fragment的xml文件分别对应一个属于自身的Activity和Fragment类，在对应类内设置所需监听，发送所需服务器请求，获取所需对应服务器响应即可。

**③后端API开发:**

本项目使用http://47.107.52.7:88/member/photo提供的各种API，该API功能较为完善，在此不逐一赘述。

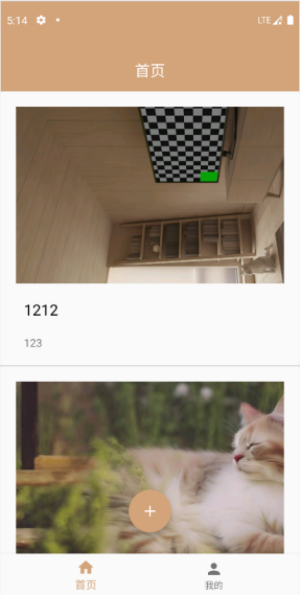
**3.详细设计**

**3.1界面设计**

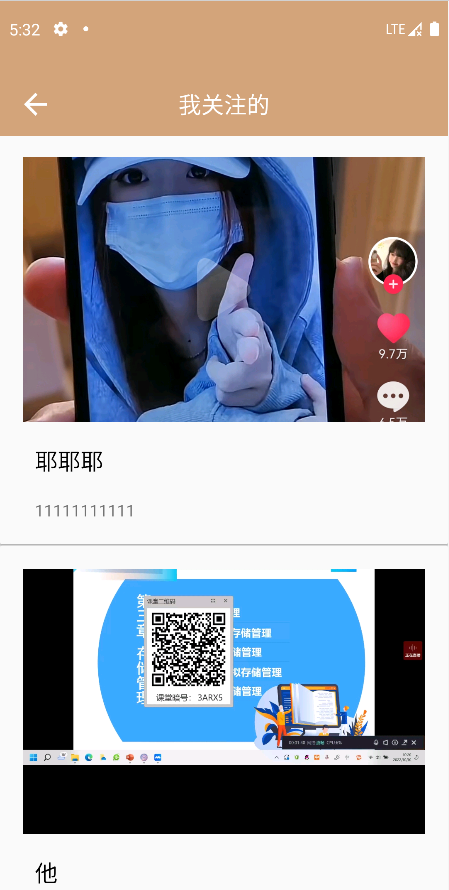
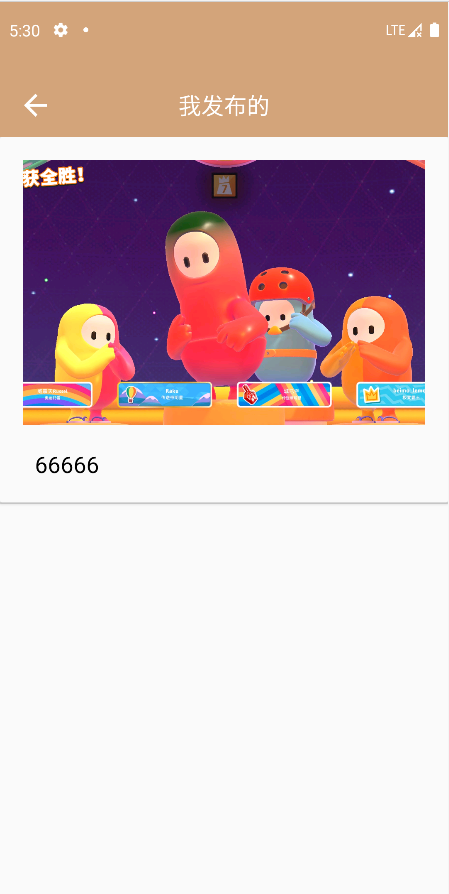
本项目的界面设计中采用了浅咖啡色→橘黄色的沉浸式渐变，采用“功能之上，简洁为主”的设计理念，将最核心、最实用的功能直接向用户呈现出来。不夹杂杂乱无章的页面元素，不将无用功能过分地喧宾夺主，不将过时的设计重现在20年代的课程设计中。

**3.1.1 UI设计**

**①首页 ②“我的”界面**



**⑤我的发布界面 ⑥我的关注动态界面**



1. **图片分享详情界面 ⑧图片分享界面**

**⑨版本信息界面 ⑩应用设置界面**

**⑪“关于我们的界面”**



**3.1.2页面交互设计**

本项目在页面交互的设计中，使用了“先登录后使用”的开启方式。用户登录和注册负责拦截未登录的行为；“主页”负责浏览所有图片分享概览和提供发布图片的入口；“图片分享详情”负责浏览动态图片的细节，并支持点赞、评论；“我的”页面负责浏览用户发布的动态和用户关注的动态，以及进行应用设置、浏览应用开发信息和版本信息；“分享”界面负责发布用户的图片动态。界面的流通交互如图1所示。

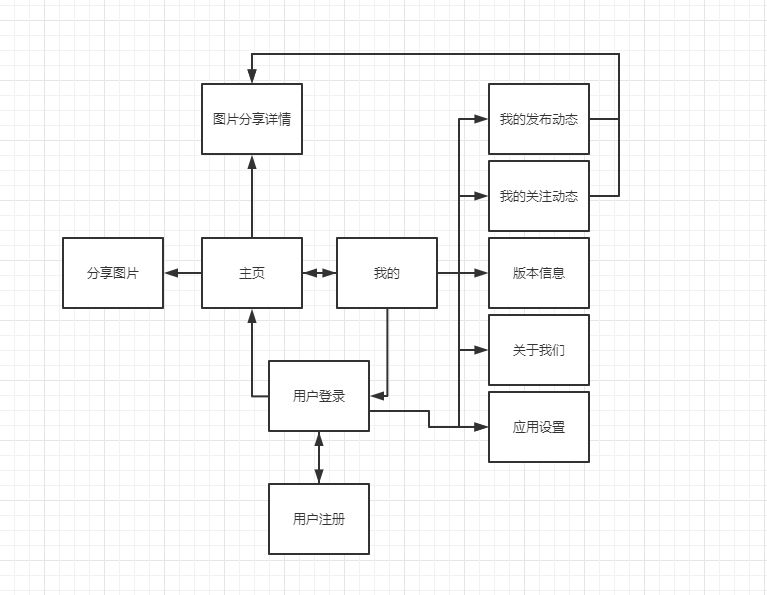


图1：界面的流通交互图

**3.2业务逻辑设计**

本项目的业务逻辑使用Android端+后端API的开发模式，即Android端负责处理用户的操作，向服务器发送请求并接受响应。于是乎，本项目的后端API实现了OpenAPI的图片分享应用的诸多接口，接口地址为http://47.107.52.7:88/member/photo，将在OpenAPI注册应用后返回的appId和appSecret加入本项目每个请求的请求头并通过授权验证，方可使用该API。

**3.2.1注册、登录业务**

本项目的注册、登录业务中，使用了OKHttpClient作为网络请求工具。在页面中添加两个EditText，分别是用户名和密码（确认密码）；添加一个Button，负责监听用户注册/登录事件。当用户点击“注册”或“登录”按钮时，触发注册或登录请求，将EditText中的用户名和密码信息向API的user/register或/user/login这两个接口发送请求，处理服务器响应即可。值得一提的是，用户注册的请求体参数要求是JSON字符串，而用户登录的要求是FormData。如果不仔细甄别，可能造成服务器无法正确处理本项目的请求体数据。

**3.2.2列表显示业务**

本项目的主页动态、我的关注动态、我的动态列表以及评论列表均采用了ListView控件，其中主页动态、我的关注动态列表的每个Item的设计如图2所示，我的动态列表的每个Item的设计如图3所示。

①对于封装主页动态、我的关注动态列表的适配器，需要封装shareBean对象，其中包括title、username、imageUrl这三个变量以及对应get、set方法。再建立SharesAdapter适配器（继承BaseAdapter），并重写其getView方法将变量填充入每个Item对应控件中，URL图片填充则交给Glide完成。

②对于封装我的动态列表的MySharesAdapter，相比sharesAdapter，只需去除username这一变量即可。

③对于封装评论列表的适配器CommentAdapter，需要用到username和content（内容）两个参数封装适配器。因为需要将所有评论不分页地一次展示出来，所以会进行两次get请求（本项目中称之为嗅探）。第一次get请求会获取该图片分享的评论数量，第二次get请求则将第一次请求中获取的评论数量作为请求参数size（表示每页显示评论的数量），一次性抓取所有属于该分享的评论。



图2：我的关注动态列表item



图3：我的动态列表item

**3.2.3列表更新业务**

分享图片、删除图片、取消关注等行为会对相应列表中的Item产生影响，此时需要本业务对指定列表进行实时刷新以提升用户体验。于是乎，Fragment和Activity的生命周期、适配器的notifyDataSetChanged( )方法就被运用到本项目中。

①合理利用Fragment和Activity的生命周期。如图和所示为Fragment和Activity的生命周期。本项目的刷新机制包括4个：在主页发布图片后主页列表的刷新、在主页删除我的图文共享后的刷新、在我的发布列表删除图文共享后的刷新、在我的关注动态列表中取消关注后的刷新。本项目则利用Intent的requestCode和responseCode传参以及重写Fragment和Activity的生命周期中的onResume( )方法实现此需求。onResume( )指将本ACtivity或Fragment从背景切换回显示状态的时间点，此时需要利用Intent传递的responseCode来判断是否需要令适配器通知列表刷新。例如本项目中HomeFragment类的requestCode以及重写的onResume( )的内容以及ShareInfo类的responseCode如下所示。

**1）HomeFragment中的onResume接收删除图片分享详情后返回的resultCode并刷新。**

@Override  
public void onResume() {  
 super.onResume();  
 if(resultCode == 2 || resultCode == 3 || sharesAdapter == null) {  
 *//如果是来自share成功的返回，或者是删除分享成功，或者是新加载主页，就将current重置* current = 1; *//初始化current* getPicture();  
 }  
}

**2）HomeFragment中进入图片分享详情携带的requestCode以及来源信息**

intent.putExtra("from",2);*//2代表从HomeFragment中传入shareInfo的信息*System.*out*.println("以userId="+sp.getString("id",null)+"进入title为"+shareBean.getTitle()+"的图文分享详情");  
startActivityForResult(intent, 222);*//222指从HomeFragment发出的Intent*

**3）ShareInfo中删除图片分享后携带指定requestCode返回HomeFragment，以达到通知HomeFragment刷新的目的。**

if(getIntent().getIntExtra("from",0) == 2 || getIntent().getIntExtra("from",0) == 5) {  
 Intent data = new Intent();  
 setResult(2, data);*//resultCode=2是指删除分享成功后，返回myShares/Shares列表时，提交给mySharesList/shareList列表刷新的指令*}

finish();

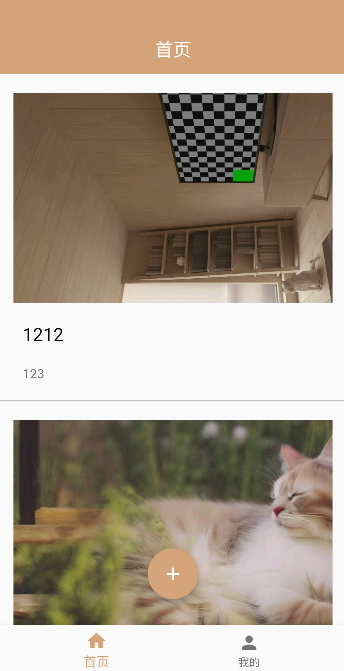
**3.2.4异步处理中修改UI**

若在异步请求中得到了服务器的响应，我们可能需要修改文字、图片等UI内容。此时容易有一个坑，就是在onResponse( )中直接修改UI的话，会收到来自系统的报错：“**Only the original thread that created a view hierarchy can touch its views**”，即我们不能在异步处理（非UI线程中）直接修改UI。这是本项目就引入了Handler，它的作用是将我们需要对UI进行的操作POST到UI线程中执行，从而避免该报错的发生。

语法很简单，即：“Handler handler = new Handler(本括号中重写一个Runnable对象的run方法，在run方法中写入对UI的操作即可).post;”。

**4.软件测试**

**4.1基本界面测试**

分别测试主页面，分享页面，个人主页的界面：

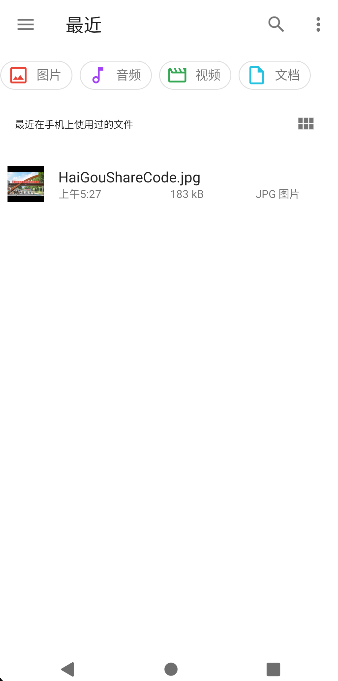
**4.2登录模块**：

登录模块的activity类通过控制台的反馈能够监视到是否成功登录，登录成功会输出部分语句，同时返回API对应的APPID和APPsecret，如下为测试结果：

当登入界面后。控制台有如下的输出：

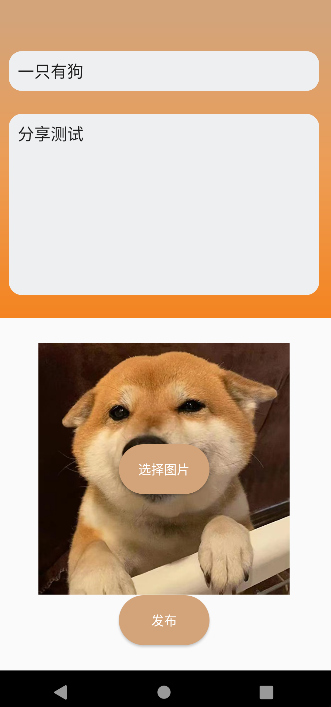
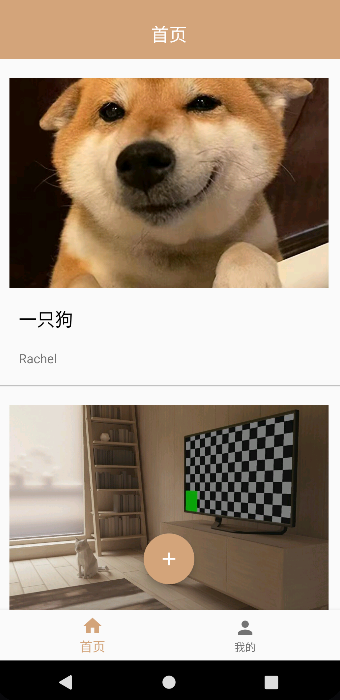
**4.3拉取共享照片模块**：

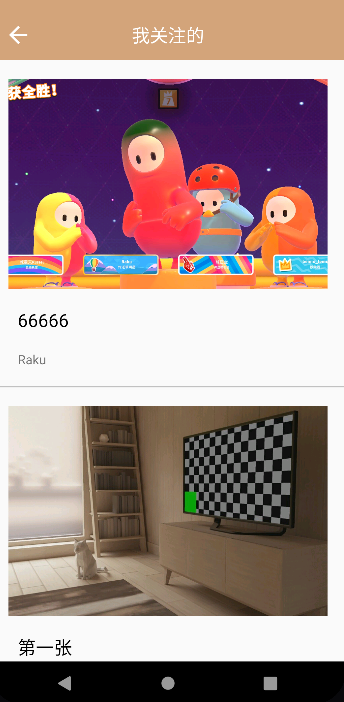
本项目采用的是OpenAPI提供的接口来实现了基本的拉取图片的功能，共同使用这一接口的用户们，在实现上传图片时，会被同步拉取到我们的主页里面，同理我们的分享也可被公开，实现了真正的图片共享，同时也能实现照片的保存，通过获取到图片的基本信息，转换成对应的格式即可保存在本地。如下是拉取的主页和保存结果：

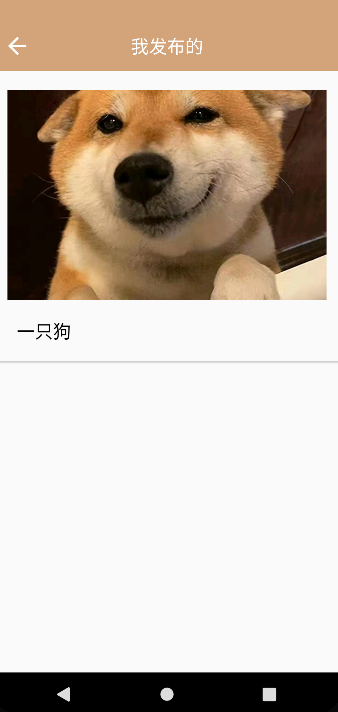
**4.4分享图片模块**：

使用OpenAPI的上传接口，将(jpg,png……)类型的图片转换成File类型的数据后整入FileLIst集合中，即可通过调用API来上传图片。如下为测试结果：

测试结果成功，按照预期的要求将所选的照片共享到首页中。



**4.5关注、点赞、评论模块**：

通过对另一个开发者所上传的照片进行评论操作和点赞分享操作，可以展示了评论功能和点赞叠加功能。进行了图片的关注后，可以在自己主页中的关注列表中看到自己添加关注的照片，预期功能基本实现。同时也能通过主页中我发布的跳转模块找寻到自己已经发布的照片和其中的内容信息：

通过对本项目中的模块及其对应的基本功能进行了测试，发现该项目基本实现了预期的图片分享功能，没有出现错误，具有较高的稳定性。

**5.心得体会**

**5.1熟悉安卓开发基本流程**：

通过本次安卓课程设计，我们小组三人共同合作，基本了解了安卓开发的基本流程，从设计UI界面到编写Activity再到Adapter的编写，熟悉整套交互的流程，同时复习了Java语言的一些特性以及课上不够熟悉的安卓知识。重新在本次课设过程中认识和了解了Activity及其生命周期的一些基本概念。UI部分的设计也重新学习了布局方面的知识和一些基本的Layout布局方式。学会了如何根据Activity的要求结合UI布局来完成Adapter的编写，完善Activity部分，实现完善的功能。

另一方面，学会了前后端分离的流程，负责API的开发者清楚了如何根据要求来编写工具类和对应的API接口和实现相应的功能。本次采用的都是OpenAPI提供的接口，只需完善部分的工具类。在此基础上，将API接口所请求返回的信息对数据进行json解析，转化成熟悉的数据类型，在这部分学会了API接口的使用、一些请求和响应、数据转换的知识。本次采用前后端分离的开发模式，对我们个人的一些数据库数据进行了有效的保护。

**5.2熟悉多人开发和Git的使用**：

Git工具是众多开发者使用的源代码管理工具（版本控制器），方便管理开发者的代码。GitHub是世界上最大的代码托管平台，使用Git分布式版本控制系统，是利于多人协同开发的代码管理平台。本次课设我们采用的是GitHub进行多人协作，在使用过程中，学会了一些基本的Git指令和对代码进行提交推送和更新、克隆等等的操作，同时简化了本次课设的开发复杂度，提升了作为开发者的一些基本素养。

1. **参考文献**

[1]陈强.精通Java开发技术.清华大学出版社.2014

[2]郭志宏.Android应用开发详解[M].电子工业出版社.2010.

[3]靳岩,姚尚朗.Google Android开发入门与实践[M].人民邮电出版社.2009.

[4]胡伟.Android系统架构及其驱动研究[J].广州广播电视大学学报.2010,10(4).

[5]余志龙,陈昱勋,郑名杰等Google Android SDK开发范例大全[M].人民邮电出版社.2009.