****

安卓综合设计报告

课题名称：基于Android的图片分享软件-看看

**课程名称： Android应用开发**

**学 院： 计算机与信息安全学院**

**组 长： 1900301236 谢浚霖**

**成 员： 1900300819 卢林军**

**1900301502 黄元炫**

**指导老师： 秦兴国**

2021年10月20日

**表1 团队任务分工表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 题目 | 基于Android的图片分享软件--“看看” | | | |
| 组长 | 谢浚霖 | | 指导教师 | 秦兴国 |
| 序号 | 学号 | 姓名 | 个人任务描述 | |
| 1 | 1900301236 | 谢浚霖 | 主要负责安卓端的整体项目架构搭建，开发个人主页模块，刷新模块，以及开发所有后端API接口，进行一定程度的页面美化。  统筹团队分工，负责总体项目管理；  参与需求分析，撰写相关内容报告；  参与后端数据库的设计，与安卓端进行数据网络请求。 | |
| 2 | 1900300819 | 卢林军 | 主要负责Android端开发，开发推荐、关注模块，频道模块，实现网络Android的网络请求模块，实现异步请求模块。  参与需求分析，撰写相关内容报告；  参与后端数据库设计，通过网络请求调用后端API接口。 | |
| 3 | 1900301502 | 黄元炫 | 负责安卓端的登录、注册模块的设计与实现，封装工具类，以及部分UI布局的设计；  负责软件测试，参与需求分析，进行报告的撰写与整理。 | |

摘 要

“看看”是一款轻便简洁、功能实用、交互友好的图片分享App软件。项目实现了一套完整的前后端系统，实现前后端分离。后端基于Flask框架，采用Python语言进行搭建服务器，用户的图片、个人信息等数据保存在云端的数据库中，Android前端采用Java语言，通过Okhttp与后端进行网络请求交互，两者通过需求分析决定好的API接口进行数据通信，整体UI设计遵循Material Design规范风格。Android端主界面包括“首页”、“频道”两大模块、支持发布分享图片、对喜欢的图片评论点赞、查看频道、搜索并下载喜欢的图片、看看聊天、抽屉布局显示个人信息等功能。

用户注册登录帐号后，可以分享自己的本地图片上传到后台服务器，用户在“发现”模块可以查看平台上其它用户分享的图片，浏览图片详情，对各个图片进行点赞、评论等社交功能，用户还可以将自己喜欢的图片保存到本地。用户在“频道”模块还可以根据个人喜好查看相关类别的图片。此外，App在抽屉布局中可以在个人信息界面修改个人信息和支持广域网内 “看看”专区聊天等功能。

在本项目的开发过程中，我们采用了Github等团队协同工具管理代码，团队内分工明确，需求分析，接口设计，前后端编码、软件测试等工作均由团队成员合作独立完成。团队编码完成后对App进行一系列测试，确保软件的高可用性，并持续维护优化App。

关键词： Android；图片分享；Flask后端；Github

目 录

[1. 概要设计 5](#_Toc26034)

[1.1 需求分析 5](#_Toc19149)

[1.1.1 功能需求 5](#_Toc14457)

[1.1.2 性能需求 10](#_Toc19623)

[1.1.3 界面需求 10](#_Toc7159)

[1.1.4 安全性需求 10](#_Toc24619)

[1.2 数据库设计 10](#_Toc15383)

[1.2.1 数据库分析 10](#_Toc12780)

[1.2.2 数据库概念结构与逻辑结构设计 11](#_Toc27148)

[1.2.2 数据库物理结构的设计与实现 12](#_Toc4478)

[2. 详细设计 15](#_Toc5539)

[2.1 登录模块 15](#_Toc7156)

[2.2 注册模块 17](#_Toc2612)

[2.3 修改信息模块 19](#_Toc20368)

[2.4 信息查看模块 21](#_Toc21525)

[2.5 说说模块 24](#_Toc31206)

[2.5.1 图片上传与发布 24](#_Toc6851)

[2.5.2 查看图片详情 26](#_Toc15343)

[2.5.3 发表评论 28](#_Toc23827)

[2.5.4 点赞和收藏图片 29](#_Toc31716)

[2.6 分类模块 31](#_Toc10280)

[2.6.1 查看分类 31](#_Toc6882)

[2.7 看看聊天 33](#_Toc26385)

[3. 软件测试 35](#_Toc13467)

[3.1 功能测试 35](#_Toc6186)

[3.1.1 注册功能测试 35](#_Toc8610)

[3.1.2 登录功能测试 35](#_Toc24252)

[3.1.3 修改信息功能测试 39](#_Toc31001)

[3.1.4 发布图片测试 40](#_Toc23154)

[3.1.5 评论图片测试 41](#_Toc29895)

[3.1.6 点赞和收藏图片测试 42](#_Toc15257)

[3.1.7 关注发布者测试 43](#_Toc29407)

[3.2 兼容性测试 44](#_Toc586)

[3.3 安全性测试 47](#_Toc27393)

[4 相关技术总结 47](#_Toc2025)

[4.1 Android 47](#_Toc6200)

[4.2 APP后端 47](#_Toc11684)

[4.3 Github代码托管 48](#_Toc26632)

[5 项目心得总结 51](#_Toc8145)

[6 参考文献 52](#_Toc32457)

# 1. 概要设计

# 1.1 需求分析

# 1.1.1 功能需求

“看看”是一款基于Android应用的图片分享软件，经过对应用商城同类App的需求调研分析，“看看”力求打造一款轻量级，支持图片分享、点赞评论常用社交互动、爬取互联网精选图片等核心功能，符合主流UI设计Material Design准则，简洁美观的图片分享软件，对于软件内每一条分享的图片内容我们称其为**说说**。其主要模块包括用户信息模块、说说推荐模块、说说分类模块、聊天模块、个人信息模块等，具体如下所示：

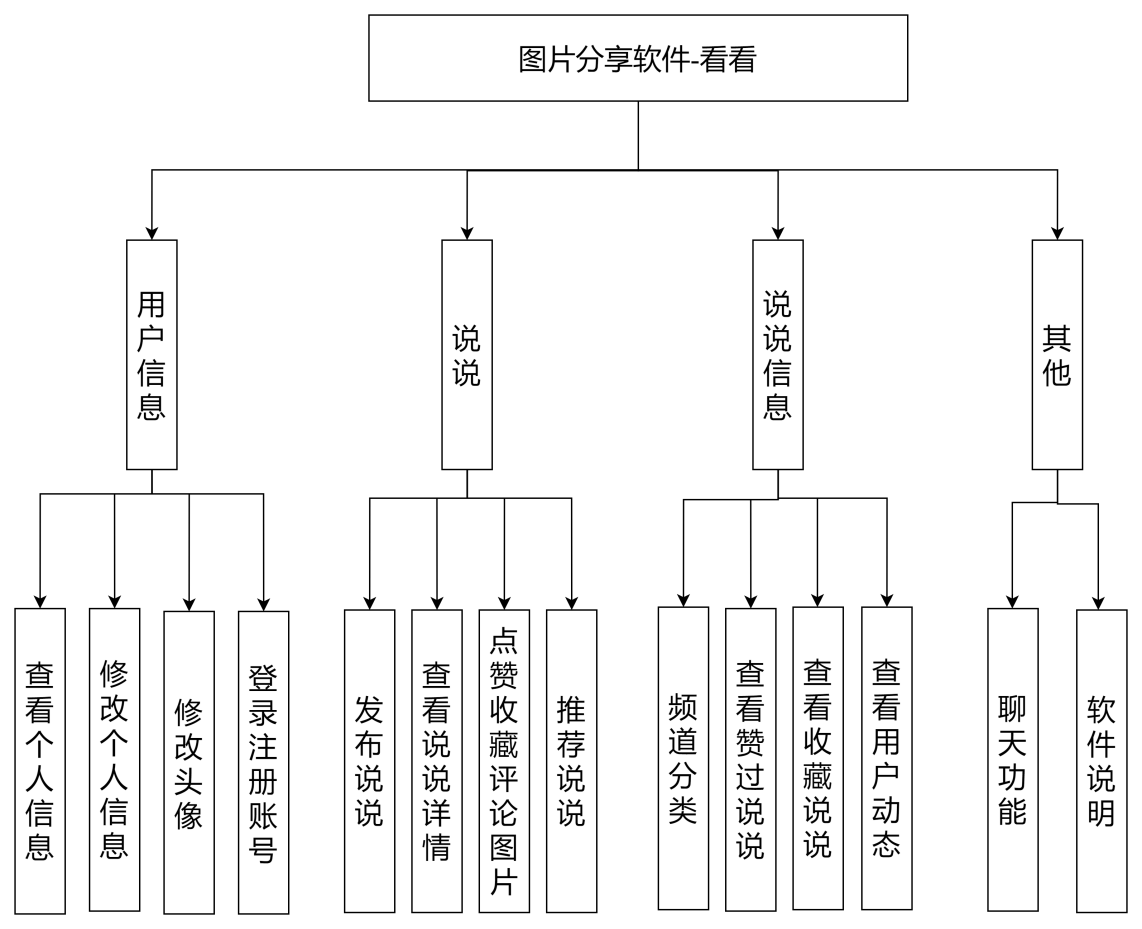
用户信息模块：查看个人信息、修改头像、修改个人信息、注册账号等功能。

说说模块：用户发布图片分享内容、查看图片详情、点赞图片、评论图片、收藏图片、推荐说说内容、显示已关注用户的发布的说说内容等功能。

说说信息模块：查看说说类别、查看用户个人主页信息和他人主页信息、查看赞过的说说、收藏的说说、用户动态等功能

其他模块：聊天室、退出登录等功能。

系统功能图如图1.1所示：



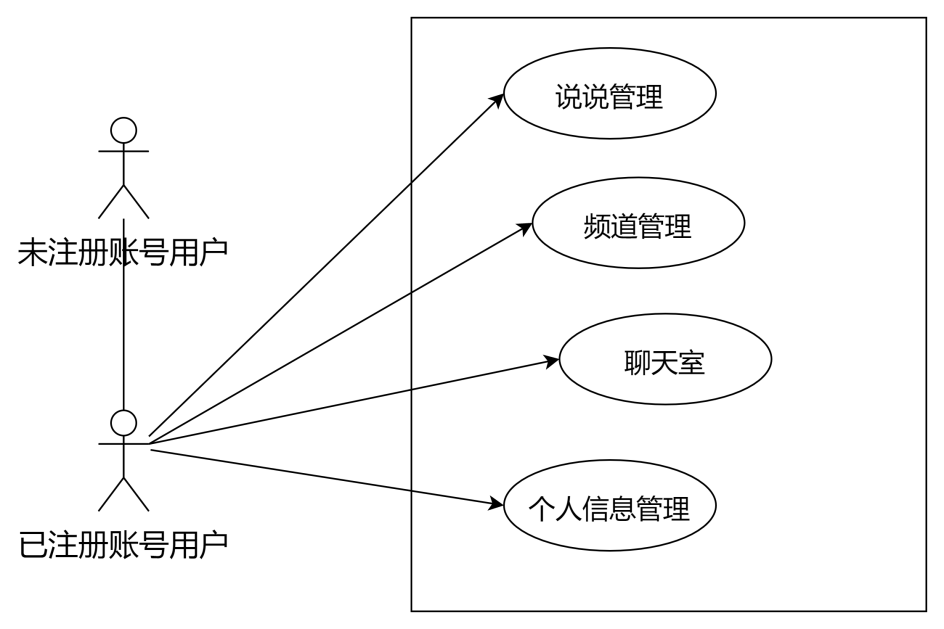
**图1.1 系统功能图**

**顶层用例**

通过分析可知，使用此软件的用户需是已注册账号用户和未注册账号用户。未注册账号用户可以进行注册账号操作、注册成功后进行登录；已注册账号用户拥有说说模块、用户信息模块和其他模块的所有功能。顶层用例分析表如下表1.1所示，顶层用例图如图1.2所示：

**表1.1 系统顶层用例表**

|  |  |
| --- | --- |
| 用例名称 | 顶层用例 |
| 事件流 | 注册账号  登录  关注说说  推荐说说  频道  聊天室  个人主页  说说动态 |
| 参与者 | 已注册账号用户  未注册账号用户 |
| 输入 | 已注册的账号和密码 |
| 输出 | 图片信息  用户个人信息 |



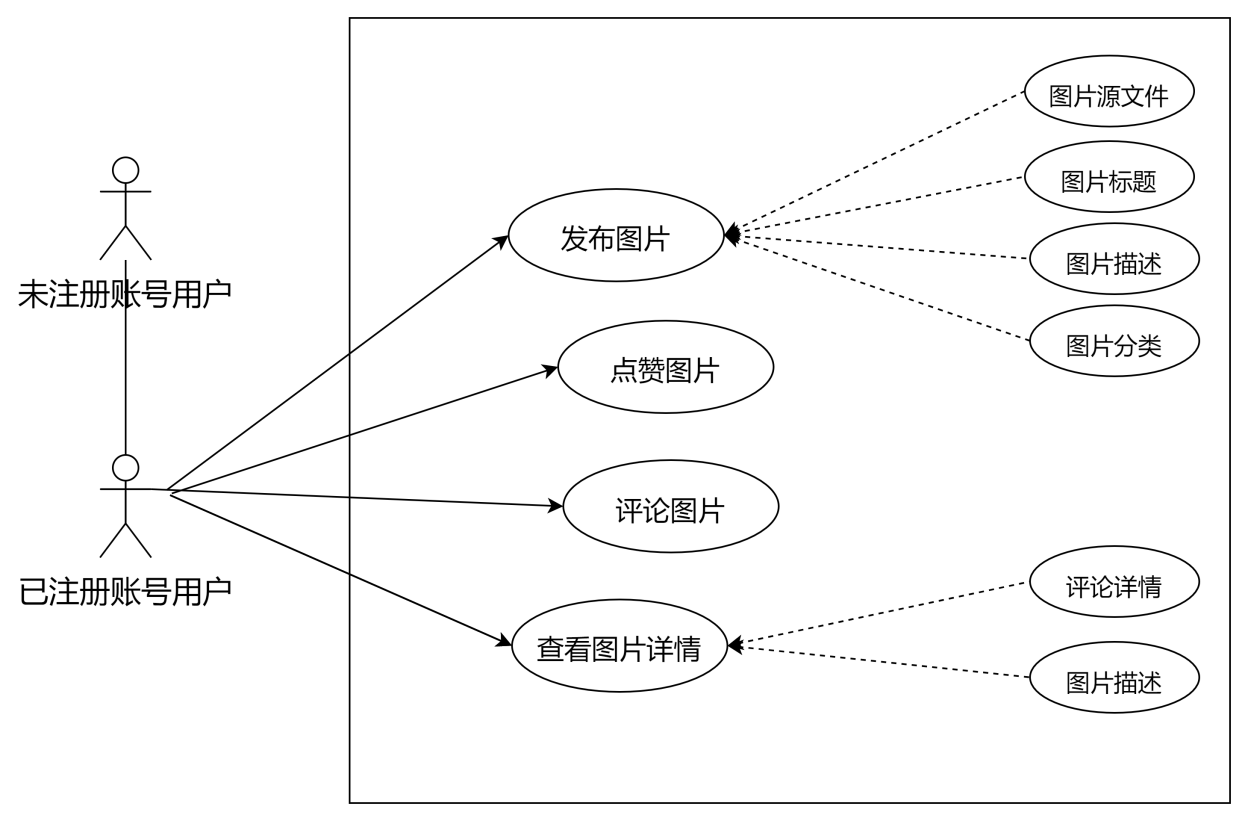
**图1.2顶层用例图**

**说说模块用例**

由顶层用例图可知，已注册账号的用户在登录成功后，会进入到应用发现主页面，发现页面为应用主页面，用户可以对已发布的图片进行点赞、评论以及查看图片的详情，在详情页面可以查看其他用户所发布的评论信息，同时用户也可以发布自己喜欢到图片到此页面。发布功能模块的用例分析表如表1.2所示，用例图如图1.3所示。

**表1.2 说说功能模块用例分析表**

|  |  |
| --- | --- |
| 用例名称 | 发现功能模块 |
| 事件流 | 发布图片  查看图片详情、  点赞图片  评论图片 |
| 参与者 | 已注册账号用户 |
| 输入 | 图片源文件  图片标题  图片描述  图片分类  点赞信息  评论信息 |
| 输出 | 图片信息  图片点赞信息  图片评论信息 |



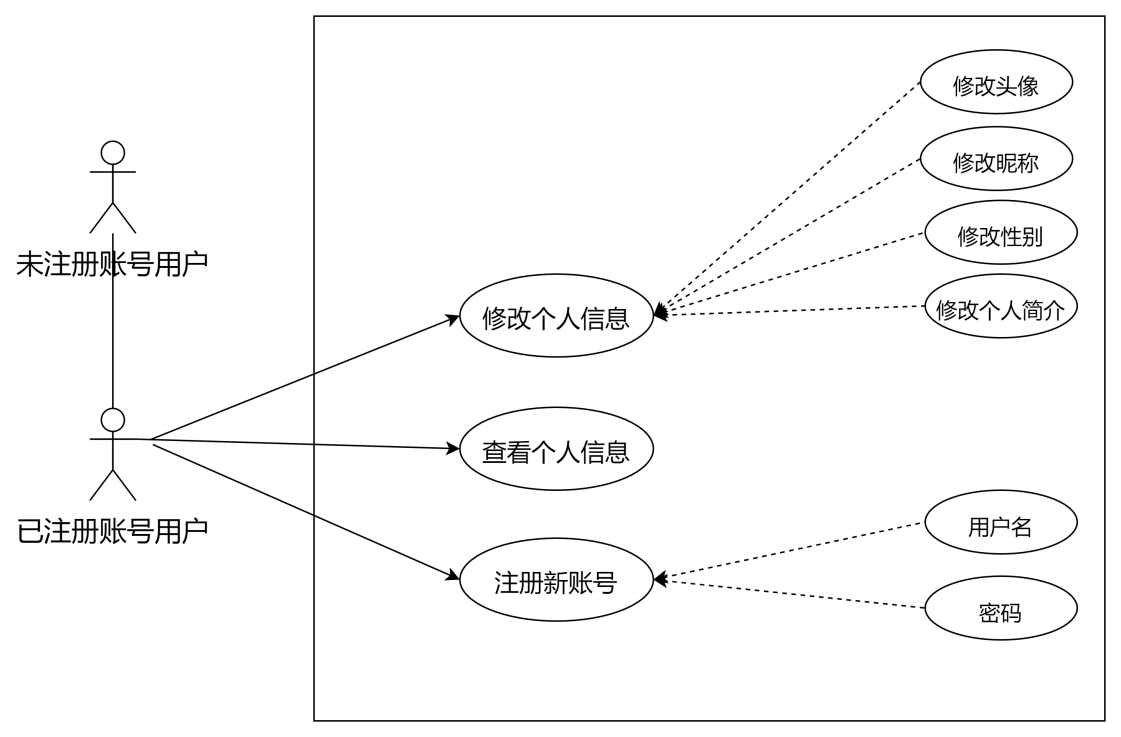
**图1.3 说说模块用例图**

**用户信息模块用例**

由顶层用例图可知，已注册账号的用户在登录成功后，可以进行用户信息的管理。包括查看个人相关信息、修改头像、修改个人相关信息和注册新账号等功能。功能模块的用例分析表如表1.3所示，用例图如图1.4所示。

**表1.3用户信息模块用例分析表**

|  |  |
| --- | --- |
| 用例名称 | 用户信息模块 |
| 事件流 | 查看个人信息  修改头像  修改昵称  修改个性签名  注册新账号 |
| 参与者 | 已注册账号用户 |
| 输入 | 个人相关信息  头像图片 |
| 输出 | 用户相关信息 |



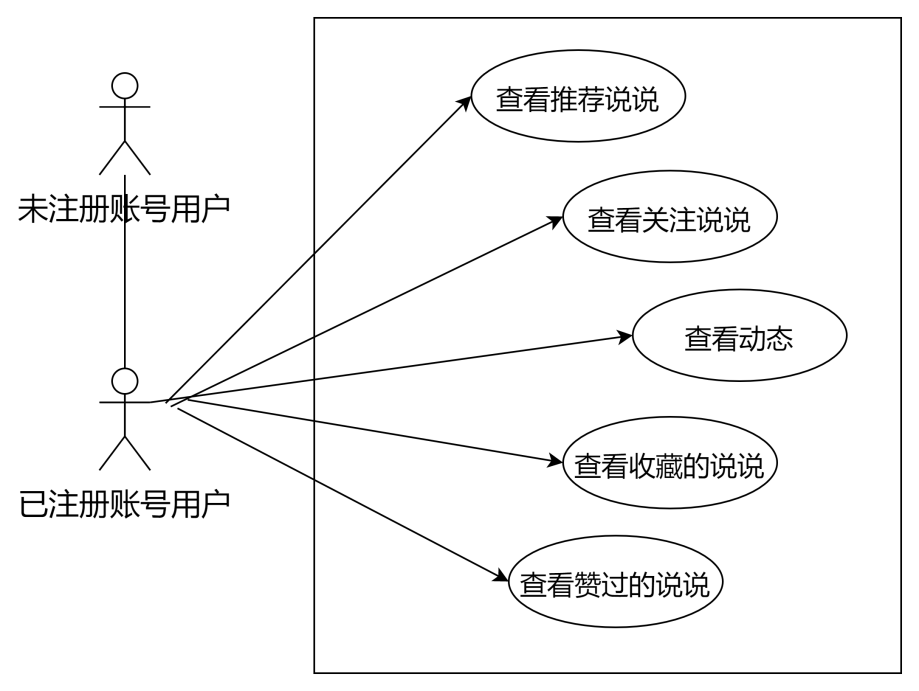
**图1.4 用户信息管理模块用例图**

**说说信息模块用例**

由顶层用例图可知，已注册账号的用户在登录成功后，可以进行用户信息的管理。包括查看个人相关信息、修改头像、修改个人相关信息和注册新账号等功能。功能模块的用例分析表如表1.4所示，用例图如图1.5所示。

**表1.4说说信息模块用例分析表**

|  |  |
| --- | --- |
| 用例名称 | 用户信息模块 |
| 事件流 | 查看推荐说说  查看关注说说  查看动态  查看赞过的说说  查看收藏的说说 |
| 参与者 | 已注册账号用户 |
| 输入 | 已注册的账号和密码 |
| 输出 | 说说信息文本 |



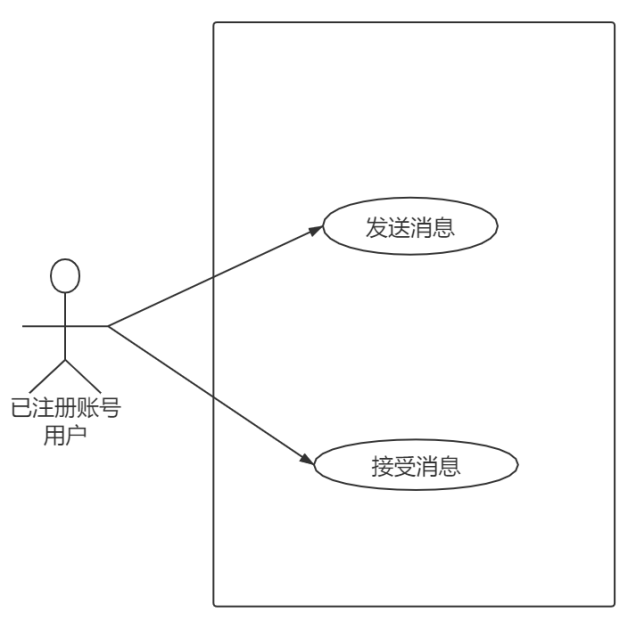
**图1.5 说说信息模块用例图**

**聊天室模块用例**

由顶层用例图可知，已注册账号的用户在登录成功后，可以进入聊天室进行聊天。包括对同一服务器下所有用户发送消息、接受来自同一服务器下所有用户的消息等功能。聊天模块的用例分析表如表1.6所示，用例图如图1.7所示。

**表1.5实时聊天模块用例分析表**

|  |  |
| --- | --- |
| 用例名称 | 实时聊天模块 |
| 事件流 | 发送消息  接受消息 |
| 参与者 | 已注册账号用户 |
| 输入 | 信息文本 |
| 输出 | 信息文本 |



**图1.6 实时聊天模块用例图**

# 1.1.2 性能需求

针对本应用的启动时间需求，因为需要连接服务器和登录需向服务器发送请求，在4秒时间内能完成开屏到登录进主页面一系列操作，已经达到较优的速度。因为应用运行所需要的数据（图片文件、账号信息等）都存储在服务器的数据库中，所以对网络要求很高。若网络延迟较高可能会出现图片加载过慢或异常的情况；若无网络连接就无法登录使用此应用。同时，在本阶段我们有专门对应用不断进行优化和维护的人员，保证软件的稳定性和更新迭代。目前，此应用可在搭载Android9.0及以上机型正常运行，最低可适配Android6.0，搭载此软件设备所需最低运行内存为2GB。后续若投入使用，会对服务器进行升级，不断优化软件。

# 1.1.3 界面需求

本应用参照Material Design的手册进行设计开发，基于已有设计稿进行开发，在此基础上对功能进行修改，同时对部分页面重新排版布局，基本遵循业界对于交互设计及UI设计的规范要求，给人清爽简洁的感觉。同时，各页面的交互方式以及图片展示形式也参考了微信和QQ等主流应用的一些方案，并尽力去还原与贴近，基本符合要求。

此外，本应用采用网上优秀的开源组件集成至应用中，致力于给用户提供更好的交互体验，例如CardView、TextInputLayout、TagFlowLayout、SmartRefreshLayout等。

# 1.1.4 安全性需求

应用后台系统需要导入用户的信息和图片信息，考虑到用户的数量以及上传的图片会比较多，所以在发送图片和信息时要保证速度和完整性。此外，大量用户有可能会同时访问系统，可能会导致系统容易崩溃，为此我们对于系统数据要定期进行备份，以防系统意外崩溃导致数据丢失。最后对于整个系统，需要完整的权限控制，防止某些人恶意攻击系统，修改原始数据。

其次用户和管理员的所有操作都必须通过登录系统界面来完成，而且还会形成操作日志。系统有严格的权限管理功能，不同的功能模块有相应的权限才能进行访问，本人以外的记录不可访问。防止数据的破坏和丢失，用户的私人信息和记录有良好的安全保障。

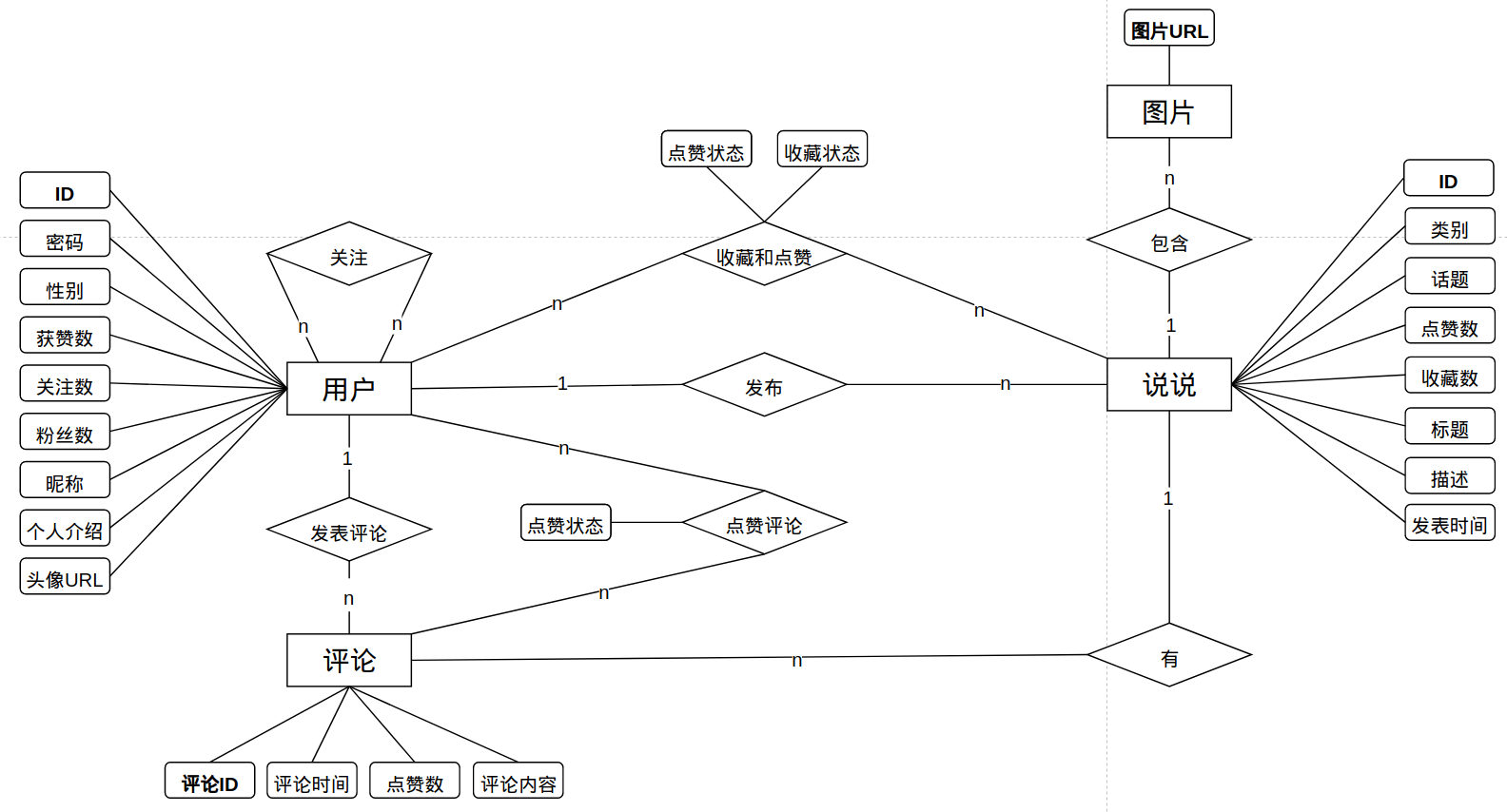
## 1.2 数据库设计

### 1.2.1 数据库分析

对数据库关系的抽象描述如下：用户有自己的ID，密码，性别，获赞数，关注数，粉丝数，昵称，个人介绍，头像。说说有对应的ID，类别，话题，点赞数，收藏数，标题，描述，发表时间。评论有对应的ID，评论时间，点赞数以及评论内容。图片有对应的URL地址。一个用户可以关注多个用户。一个用户可以发布多条说说，一条说说只能由一个用户发布。一个用户可以点赞和收藏多个说说，一个说说可以被多个用户点赞和收藏。一个用户可以发表多条评论，一个评论，只能被一个用户发表。一个用互可以点赞多条评论，一条评论可以被多个用户点赞。一条说说可以包含多张图片，一张图片只能属于一个说说。一个说说有多条评论，一个评论只能属于一个说说。

### 1.2.2 数据库概念结构与逻辑结构设计

根据关系模式的抽象描述，可以绘制E-R图，如图1.7所示：



**图1.7 数据库的E-R图**

根据上述E-R图的设计以及E-R图之间的转化关系模型的规范，可以将关系模型转化成如表x所示的关系，根据BC范式的定义，该关系表满足BF范式。

**表1.6 数据库的逻辑结构**

| **表名** | **属性** |
| --- | --- |
| 用户 | ID，密码，性别，获赞数，关注数，粉丝数，昵称，个人介绍，头像URL |
| 说说 | ID，类别，话题，点赞数，收藏数，标题，描述，发表时间，作者ID |
| 评论 | ID，评论时间，点赞数，评论内容，作者ID，说说ID |
| 图片 | 图片URL，说说ID |
| 说说收藏与点赞 | 用户ID，说说ID，点赞状态，收藏状态 |
| 关注 | 用户ID，用户ID |
| 点赞评论 | 用户ID，评论ID，点赞状态 |

### 1.2.2 数据库物理结构的设计与实现

依据上述的关系模式，使用数据库建立了七张关系表，他们的表明以及对应的名称如下表x所示。

**表1.7 数据库的物理结构**

| **表名** | **备注** |
| --- | --- |
| **users** | **用户表** |
| **dynamic** | **说说表** |
| **remark** | **评论表** |
| **photo** | **图片表** |
| **favor** | **说说收藏与点赞表** |
| **concern** | **关注表** |
| **thumbsup** | **点赞评论表** |

1. **用户表**

用户表用于存储用户的相关数据，在登录和个人信息查看与修改时使用。在用户表中我们记录了用户的个人id(唯一标识)，密码，性别，帐号获赞数，关注数，粉丝数，昵称，个人简介，以及头像的地址等信息。其中密码使用了MD5码进行加密，确保了用户帐号的安全。用户表的详细结构如表x所示。

**表1.8 用户表设计**

| **名称** | **字段** | **类型** | **说明** |
| --- | --- | --- | --- |
| ID | id | varchar(100) | 主码 |
| 密码 | passwd | varchar(50) |  |
| 性别 | sex | char(1) | F表示女性，M表示男性 |
| 获赞数 | great | int |  |
| 关注数 | concern | int |  |
| 粉丝数 | fan | int |  |
| 昵称 | username | varchar(50) |  |
| 个人介绍 | introduction | varchar(100) |  |
| 头像URL | url | varchar(50) |  |

1. **说说表**

说说表是存放分享图片的记录，在用户主页显示说说，或者发布说说时使用。在说说表中，我们记录了说说的id(唯一标识)，类别，话题，点赞数，收藏数，标题，描述，发表时间，发表作者ID的信息，其中发表作者的ID用于与用户表进行自然连接。说说表的详细结构如表x所示。

**表1.9 说说表设计**

| **名称** | **字段** | **类型** | **说明** |
| --- | --- | --- | --- |
| ID | id | int | 主码 |
| 类别 | class | varchar(50) |  |
| 话题 | topic | varchar(50) |  |
| 点赞数 | great | int |  |
| 收藏数 | star | int |  |
| 标题 | title | varchar(50) |  |
| 描述 | description | text |  |
| 发表时间 | date | varchar(50) |  |
| 发表作者ID | author | varchar(50) | 用户表的外码 |

1. **评论表**

评论表记录了每一条说说下的评论，在用户点开说说的详细信息和对说说发表评论时会使用。在评论表中记录了说说的ID(唯一标识)，评论时间，点赞数，评论内容，作者ID，说说ID的信息。其中作者ID与说说ID分别作为评论表与用户表和说说表的外键，说说表的详细设计结构如表x所示。

**表1.10 评论表设计**

| **名称** | **字段** | **类型** | **说明** |
| --- | --- | --- | --- |
| ID | id | int | 主码 |
| 评论时间 | date | varchar(50) |  |
| 点赞数 | great | int |  |
| 评论内容 | context | text |  |
| 作者ID | author | varchar(50) | 用户表外码 |
| 说说ID | dynamic | int | 说说表外码 |

1. **图片表**

图片表用于存储图像的URL，在客户端显示图片时调用。在说说表中存储了图片的URL和说说的ID，其中说说ID作为外码与说说表进行关联。图片表的详细设计如表x所示。

**表1.11 图片表设计**

| **名称** | **字段** | **类型** | **说明** |
| --- | --- | --- | --- |
| 图片URL | url | varchar(200) | 主码 |
| 说说ID | dynamic | int | 说说表外码 |

1. **说说收藏与点赞表**

说说收藏与点赞表用于记录用户对某个说说的状态，在用户进行点赞等操作时调用。在说说收藏与点赞表中，记录了用户的ID，说说的ID，点赞状态，收藏状态。

**表1.12 说说收藏与点赞表设计**

| **名称** | **字段** | **类型** | **说明** |
| --- | --- | --- | --- |
| 用户ID | user | varchar(50) | 联合主码 |
| 说说ID | dynamic | varchar(50) |
| 点赞状态 | great | char(1) | T表示点赞(收藏)  F表示未点赞(收藏) |
| 收藏状态 | star | char(1) |

1. **关注表**

关注表用于记录用户之间的关注关系，在关注表中记录了用户ID和被关注用户的ID。关注表的详细设计如表x所示。

**表1.13关注表设计**

| **名称** | **字段** | **类型** | **说明** |
| --- | --- | --- | --- |
| 用户ID | user | varchar(50) | 联合主码 |
| 被关注用户ID | concern | varchar(50) |

1. **点赞评论表**

点赞评论表用于记录用户对某条评论的状态，使用该状态对用户的点赞行为进行判断。点赞评论表的详细设计如表x所示。

**表1.14 点赞评论表设计**

| **名称** | **字段** | **类型** | **说明** |
| --- | --- | --- | --- |
| 用户ID | user | varchar(50) | 联合主码 |
| 评论ID | remark | varchar(50) |
| 点赞状态 | great | char(1) | T表示点赞(收藏)  F表示未点赞(收藏) |

# 2. 详细设计

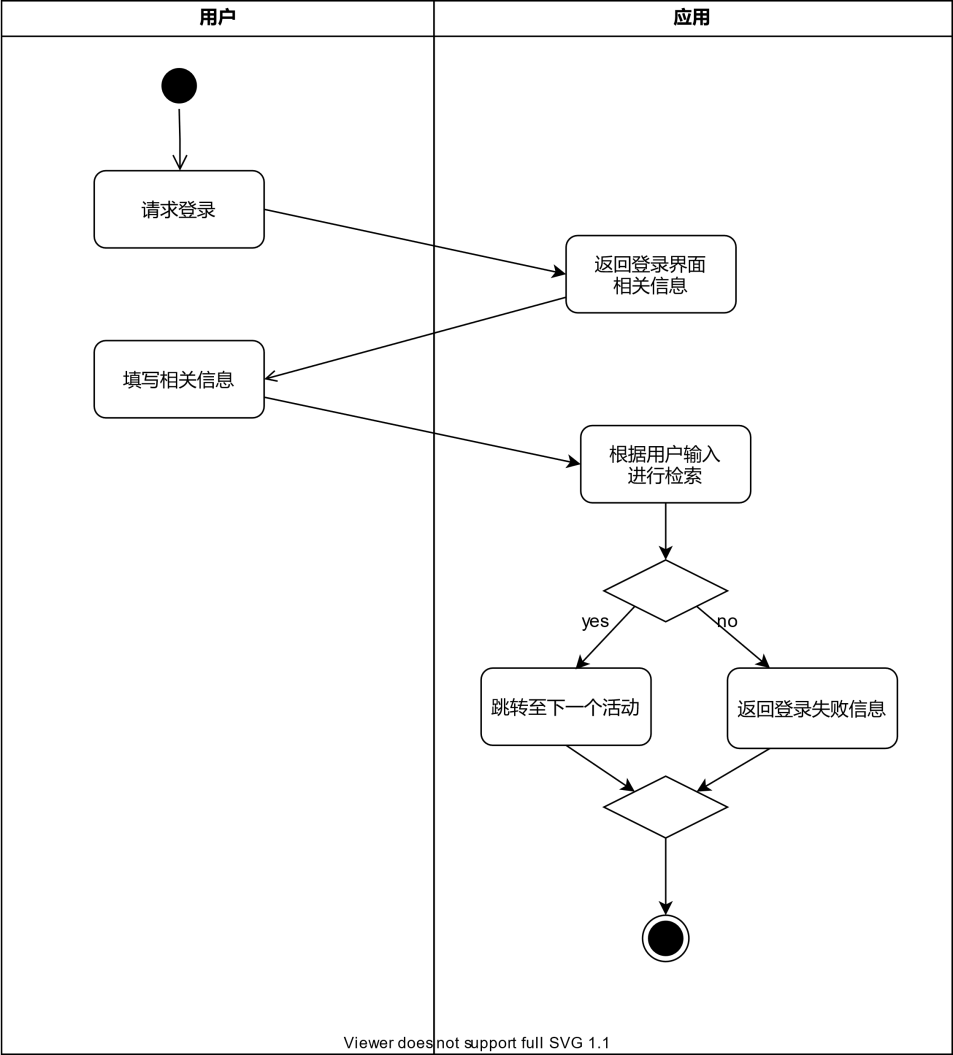
在详细设计阶段，通过需求分析的结果，设计出满足用户需求的软件系列产品。由于本应用的功能模块相对较多，所以在此部分列出相对核心的模块，进行分析。

## **2.1 登录模块**

登录模块是图片分享软件-看看的入口，若用户想要使用本应用的所有功能，都必须通过此模块。用户的登录过程为：

* 用户发起登录请求
* 应用返回登录界面的有关信息
* 用户填写相关信息（账号密码）
* 服务器根据用户输入信息进行检索
* 若在服务器的数据库中检索到此用户账号信息，则返回Toast提示，并从当前活动跳转至下一个活动，用户登录成功。
* 若未在服务器的数据库中检索到此用户账号信息或账号信息密码信息不匹配，则返回对应的Toast提示，用户登录失败。

该过程活动图如图2.1所示：



**图2.1 登录活动图**

实现效果如图2.2所示：



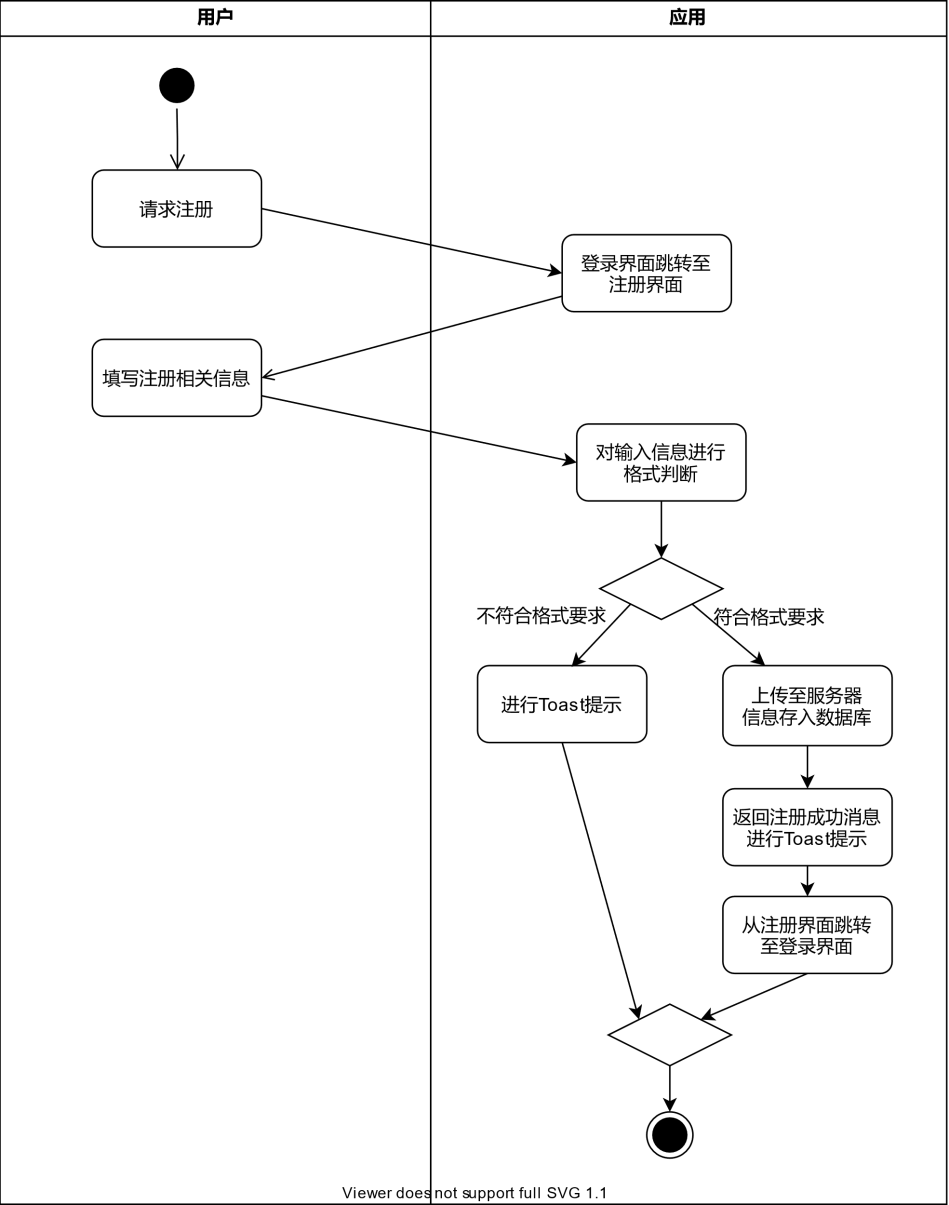
**图2.2 登录界面**

## **2.2 注册模块**

注册模块提供给未拥有账户，但将要使用此图片分享软件-看看的对象一个入口，用户注册的过程为：

* 用户发起注册请求
* 系统接受到请求后，将从登录界面跳转至注册界面
* 用户填写相关信息（用户名，密码等信息）
* 前台对填入信息进行格式判断，若不符合要求，则会进行Toast提示，符合格式要求数据会上传存储至服务器后台
* 系统成功存储用户信息，返回注册成功消息
* 应用获取到注册成功信息，进行Toast信息提示，并从注册界面跳转至登录界面。

该过程活动图表示如图2.3所：



**图2.3 注册活动图**

实现效果如图2.4所示：



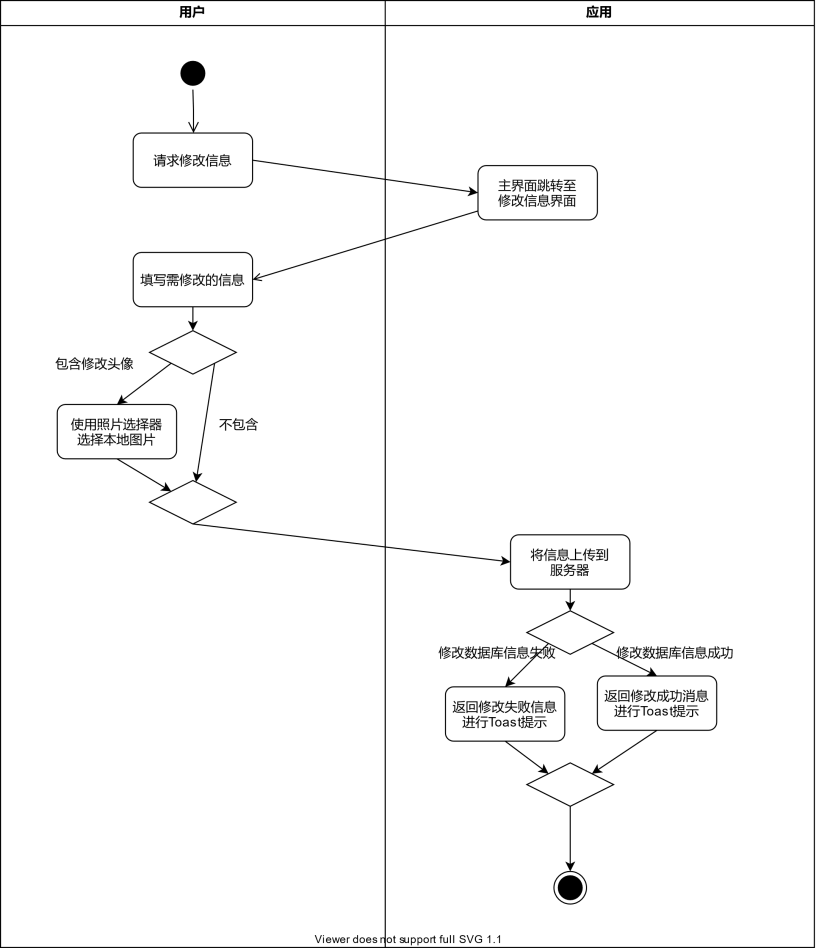
**图2.4 注册界面**

## **2.3 修改信息模块**

用户在注册时并未提供选取图片作为用户头像的功能，且用户有不断修改个人信息的需求，所以在应用中提供修改用户信息的功能。用户点击我的-编辑资料即可修改信息，修改用户信息的过程为：

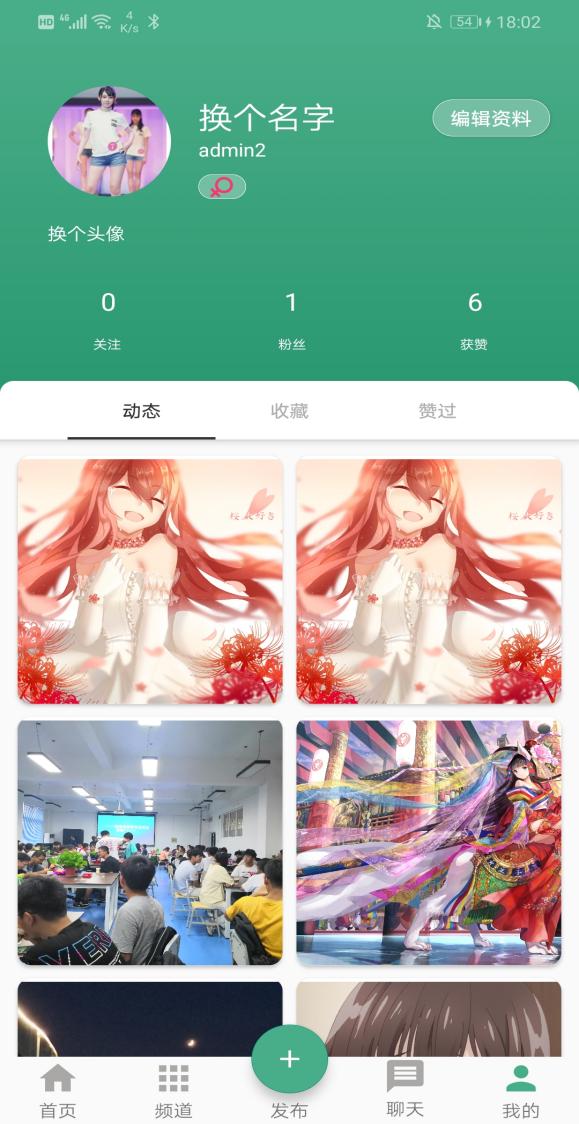
* 用户发起修改信息请求
* 系统接受到请求后，将从首页跳转至修改信息界面
* 用户填写需修改的信息并提交（头像、昵称、性别和个性介绍等信息）若用户需修改的信息为头像，前台使用已开源的照片选择器展示用户设备内部的本地图片，用户选取一张图片作为头像上传。
* 系统根据用户提交的信息，分别修改服务器数据库中此用户的相关信息。若提交的信息包含修改头像，则上传的图片是由应用前台通过图片存储路径获取本地文件，再上传至服务器中进行修改。
* 应用获取到信息修改的返回信息，若信息修改成功，刷新后用户的信息会发现改变，并出现修改成功的Toast信息提示；若信息修改失败，出现修改失败的Toast信息提示，用户信息未发现变化。

该过程活动图表示如图2.5所示：



**图2.5 修改信息活动图**

实现效果如图2.6、2.7所示：

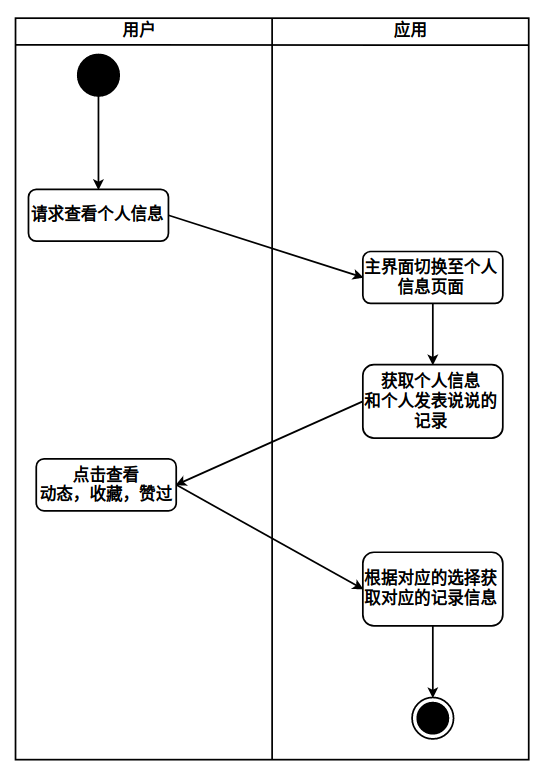
**图2.6 修改信息界面 图2.7 修改头像等信息成功界面**

# 2.4 个人信息查看模块

信息查看模块，我们的个人界面显示了个人的基本信息，外还提供了个人说说查看，收藏说说和点赞说说查看，分别对应自己发的动态，自己的收藏，自己点赞的说说。其基本过程为：

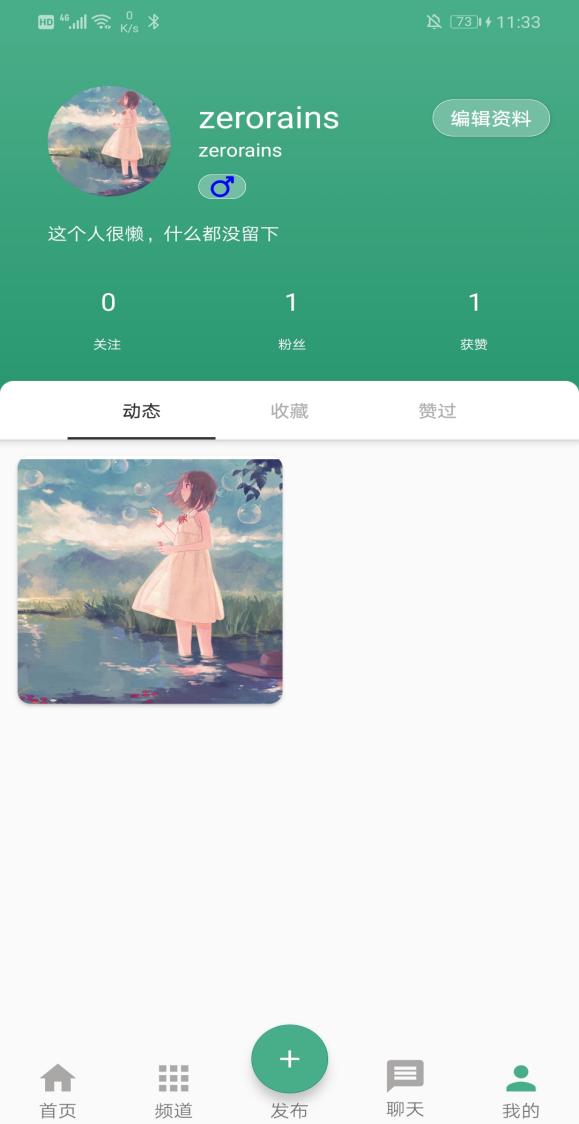
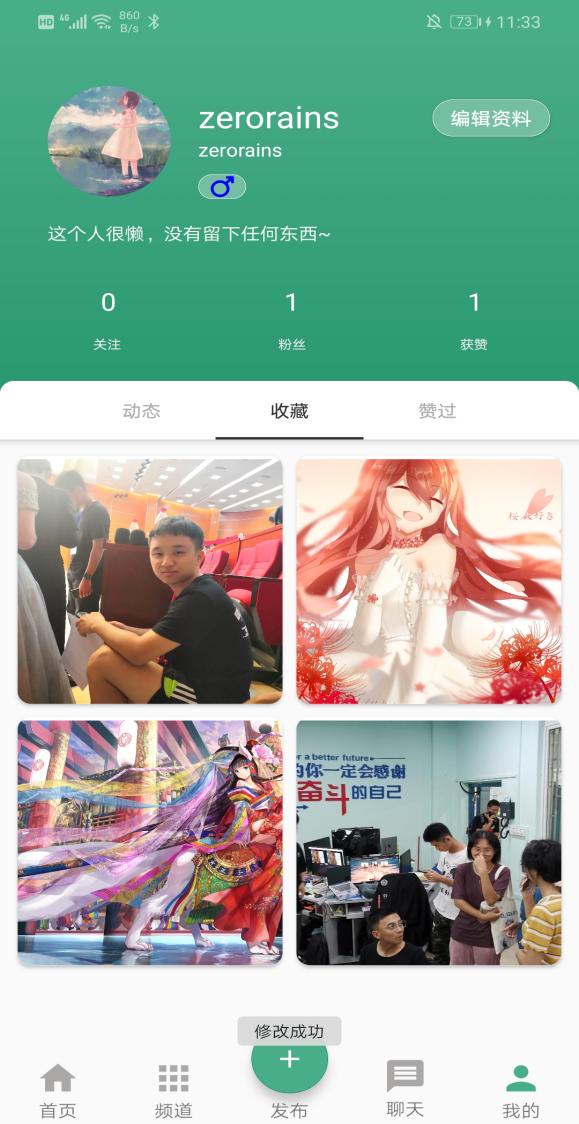
* 用户点击导航栏”我的”按钮
* 系统接受到请求后，将从首页跳转至个人信息界面
* 用户端获取用户的个人信息，并获取用户发布过，点赞，收藏过的图片说说。

该过程活动图表示如图2.8所示：

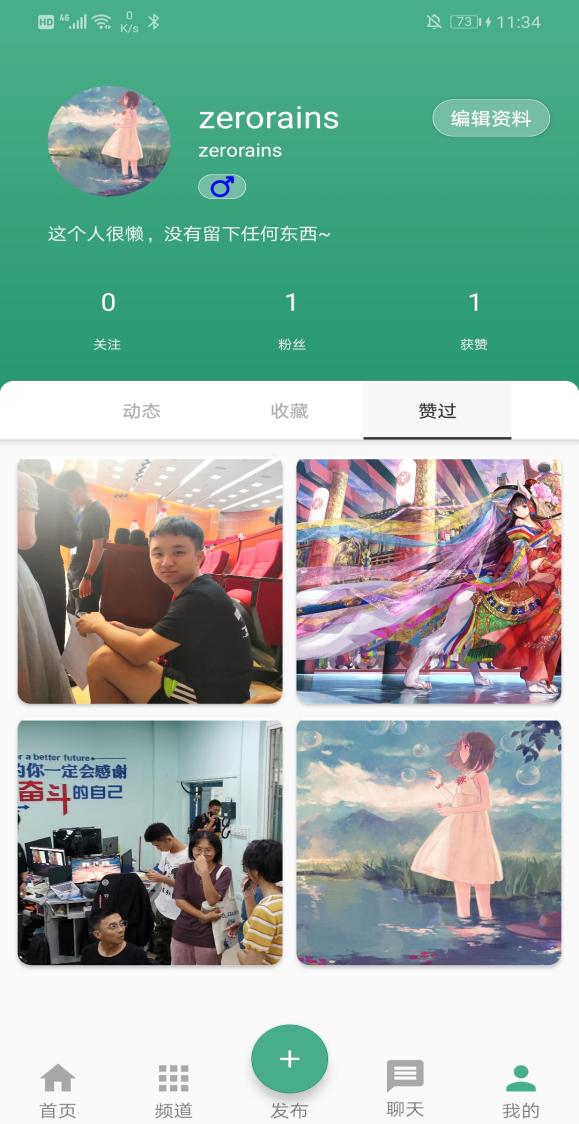


**图2.8 个人信息查看活动图**

实现效果如图2.9、图2.10所示：

**图2.9 个人动态 图2.10个人收藏**



**图2.10 个人点赞**

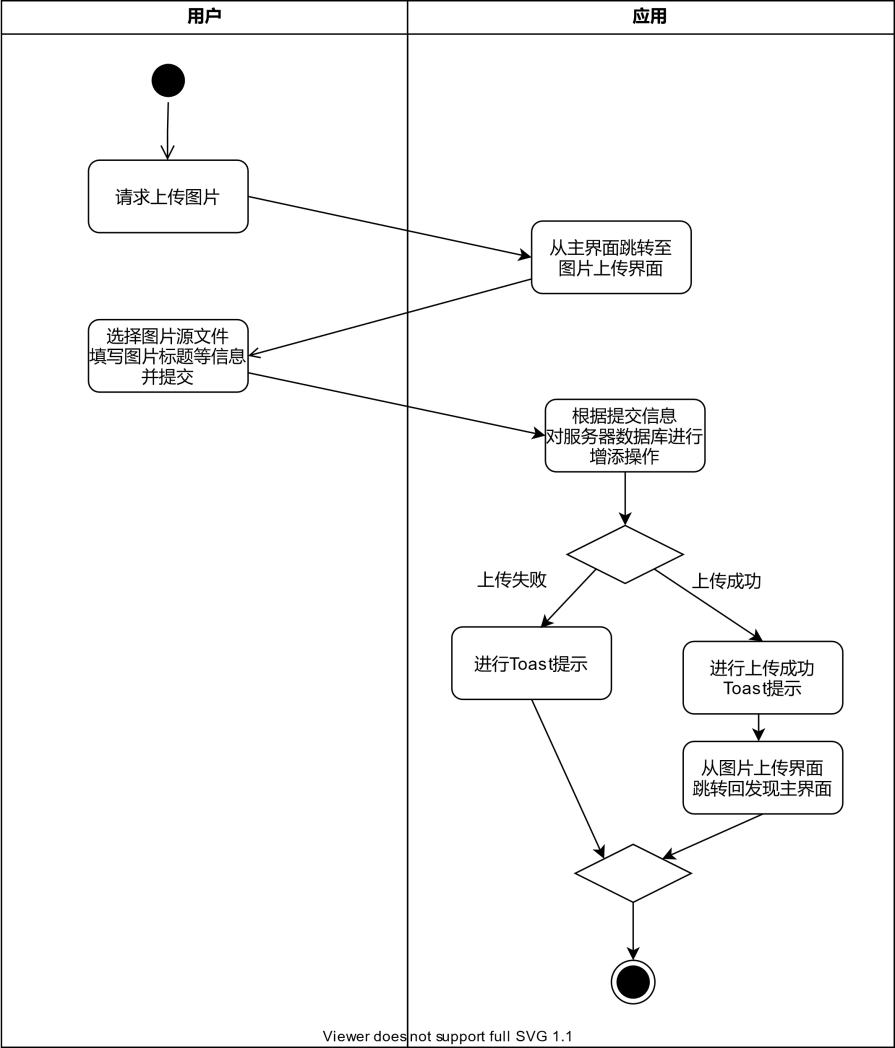
# 2.5 说说模块

# 2.5.1 图片上传与发布

用户成功登录后，可以进入首页，点击＋”号按钮，可以进行本地图片的上传。图片上传与发布的过程为：

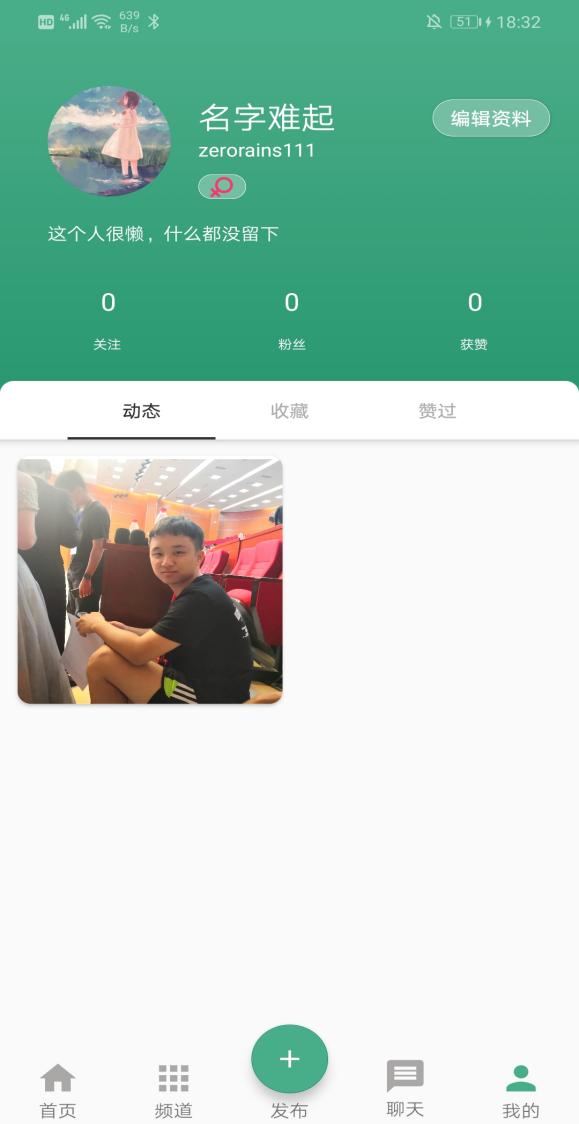
* 用户点击上传图片按钮，发出上传图片请求
* 系统接受到请求后，将从首页跳转至图片上传界面
* 用户填写相关信息并提交（图片标题、图片描述、图片话题、图片类别）。前台使用已开源的照片选择器展示用户设备内部的本地图片，用户选取一张图片上传。
* 系统根据用户提交的信息，对服务器数据库进行增添信息操作。其中上传的图片文件是由应用前台通过图片存储路径获取本地文件，再传至服务器中，进行增添。
* 应用获取到系统返回的提示信息，若上传发布成功，则会跳转回首页。若发布成功就会出现Toast的提示，并点击我的可以查看到自己分享的图片信息；若信息修改失败，出现发布失败的Toast信息提示。

该过程活动图表示如图2.8所示：



**图2.8 图片上传与发布活动图**

实现效果如图2.9、2.10所示：

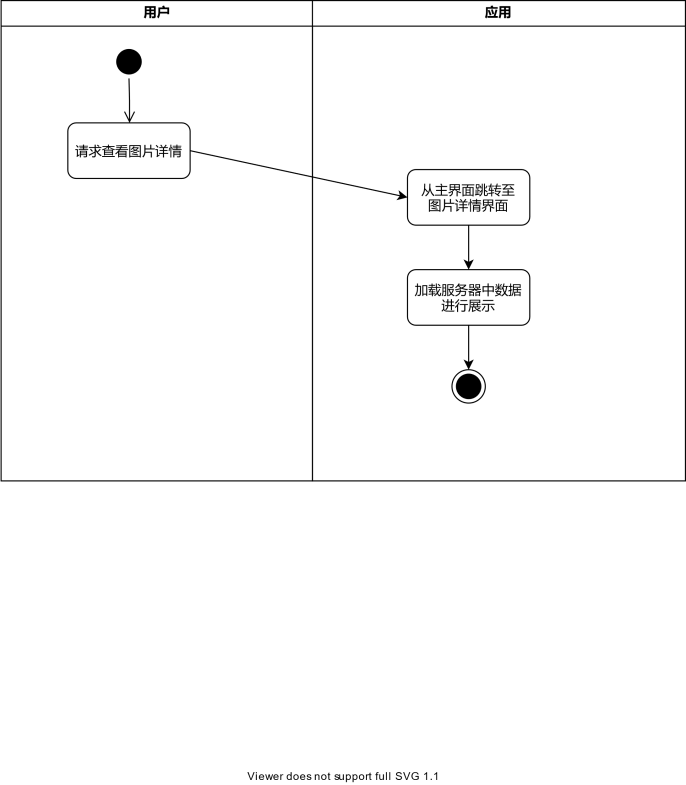
**图2.9 上传图片界面 图2.10 上传成功界面**

# 2.5.2 查看图片详情

用户在首页，点击已经发布图片可以进入图片详情页面，查看图片标题、描述及评论信息。查看图片详情的过程为：

* 用户点击首页图片，发出查看图片详情请求
* 应用接受到请求后，将从首页跳转至图片详情页面。同时服务器将此图片的数据（标题、描述和评论信息等）发送给应用。
* 应用接受到图片相关数据，进行处理后展示给用户。

该过程活动图表示如图2.11所示：



**图2.11 查看图片详情活动图**

实现效果如图2.12所示：

****

**图2.12 查看图片详情**

# 2.5.3 发表评论

用户在首页，点击图片下方评论按钮，即对图片进行评论。发表评论的过程为：

* 用户点击首页图片评论按钮，发出评论请求
* 应用接受到请求后，将会弹出评论文本输入框
* 用户输入评论文本信息，点击发布后，应用会将此内容上传至服务器。
* 服务器将评论信息保存至数据库中，并返回相关信息
* 应用接受到反馈信息展示给用户，若评论发布成功，会出现评论成功的Toast信息提示；若发布失败，则会出现评论失败的Toast信息提示

该过程活动图表示如图2.13所示：



**图2.13 发表评论活动图**

实现效果如图2.14、2.15所示：

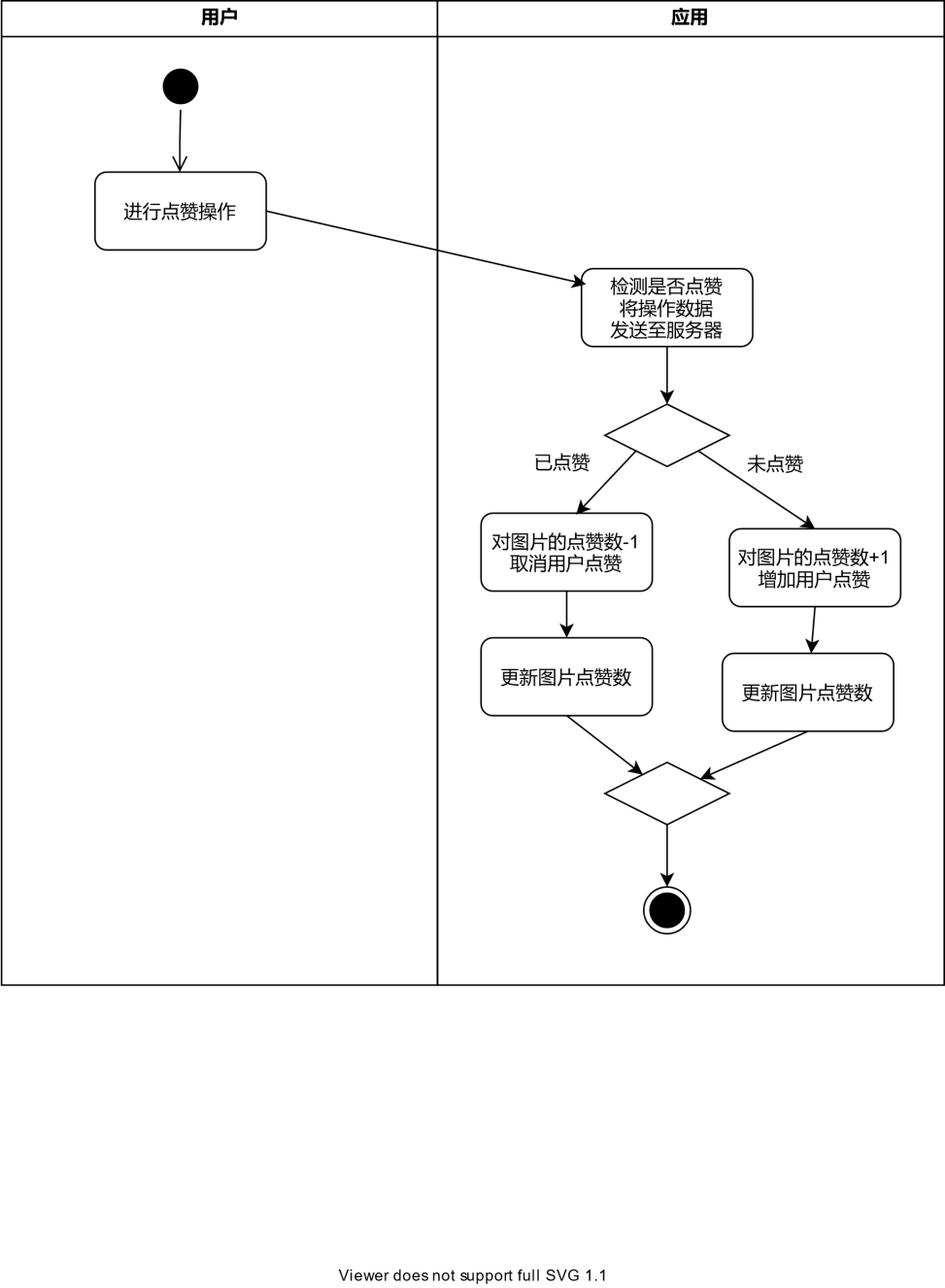
**图2.14 发表评论界面 图2.15 发表评论成功界面**

# 2.5.4 点赞和收藏图片

用户在首页，点击图片下方点赞按钮，即可对图片进行点赞/取消点赞，收藏/取消收藏。点赞的流程为：

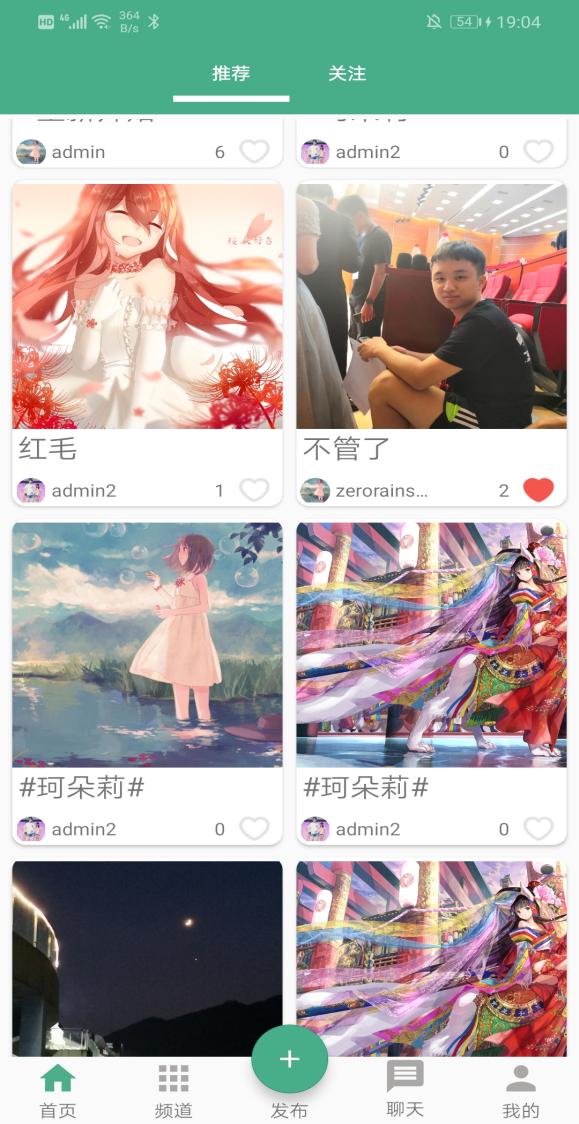
* 用户点击首页图片点赞按钮
* 应用检测是否点赞，将点赞/取消点赞等操作数据发送至服务器
* 服务器根据操作将图片中的点赞数+1/-1，并返回数据信息给应用
* 应用获取到反馈信息，更新图片点赞数并展示给用户

该过程活动图表示如图2.16所示：



**图2.16 点赞图片活动图**

实现效果如图2.17、图2.18所示：

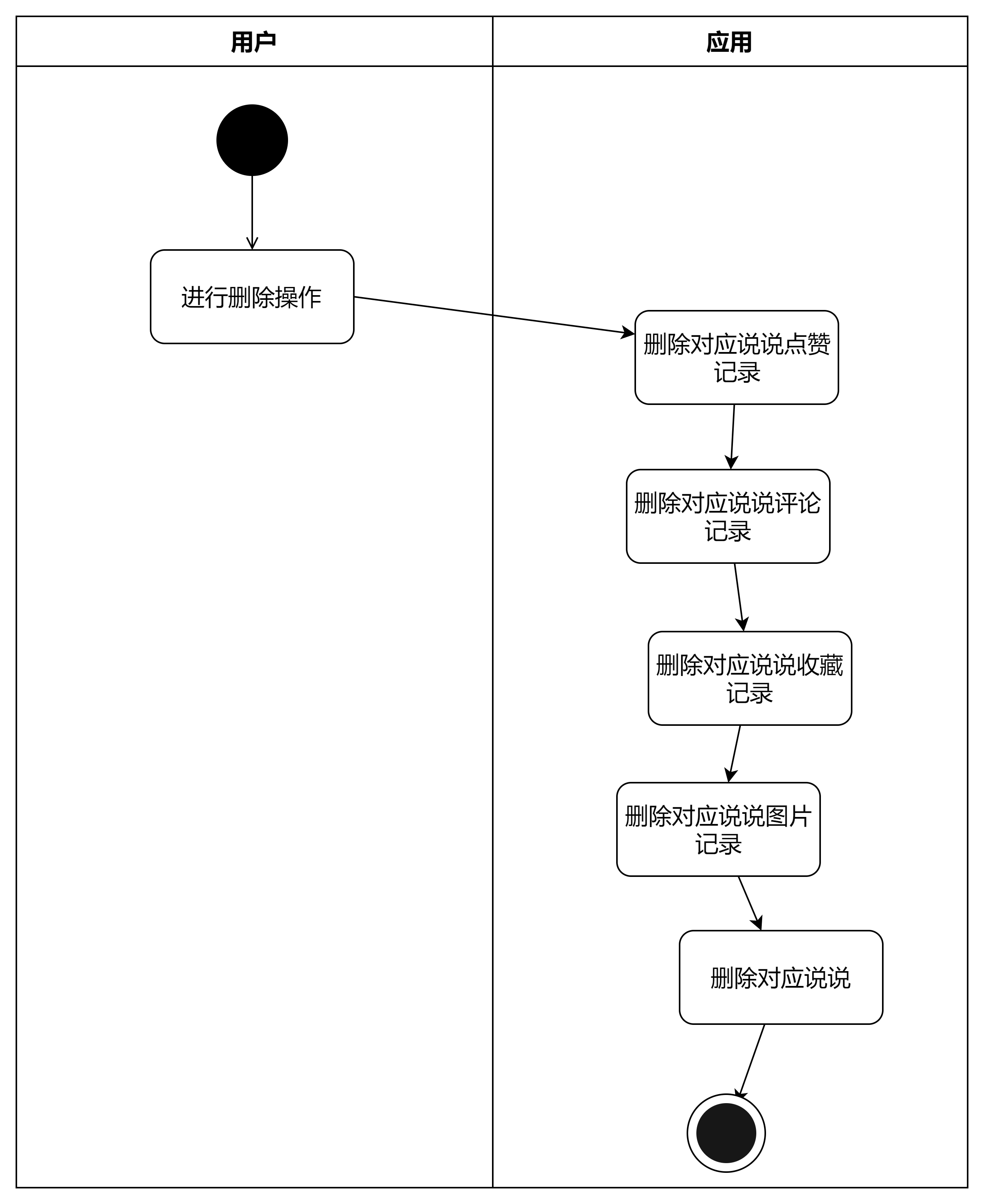
**图2.17 详细界面点赞收藏界面 图2.18 首页收藏界面**

# 2.5.5 删除图片

用户在自己发布说说的详情界面，点击右上角删除按钮，即可对图片说说金星和删除操作。点赞的流程为：

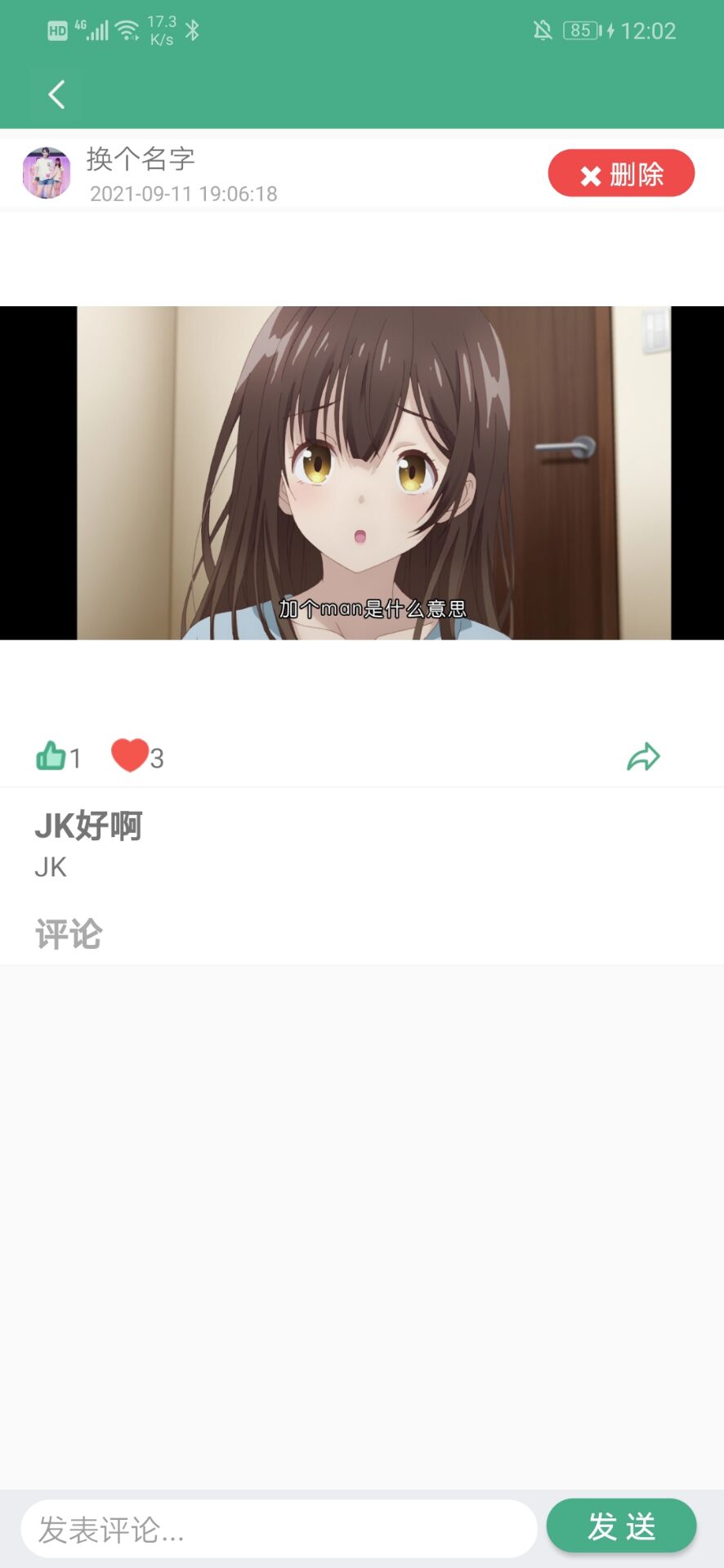
* 用户点击删除按钮
* 调用后端服务器删除说说接口
* 根据外键引用方式，依次删除对应说说相关的数据，依次为thumbsup\_comments表、comments表、favor表、photo表、shuoshuo表
* 将操作结果返回给用户并给予提示

该过程活动图表示如图2.19所示：



**图2.19 删除说说活动图**

实现效果如图2.20所示：

**图2.20 删除界面效果图**

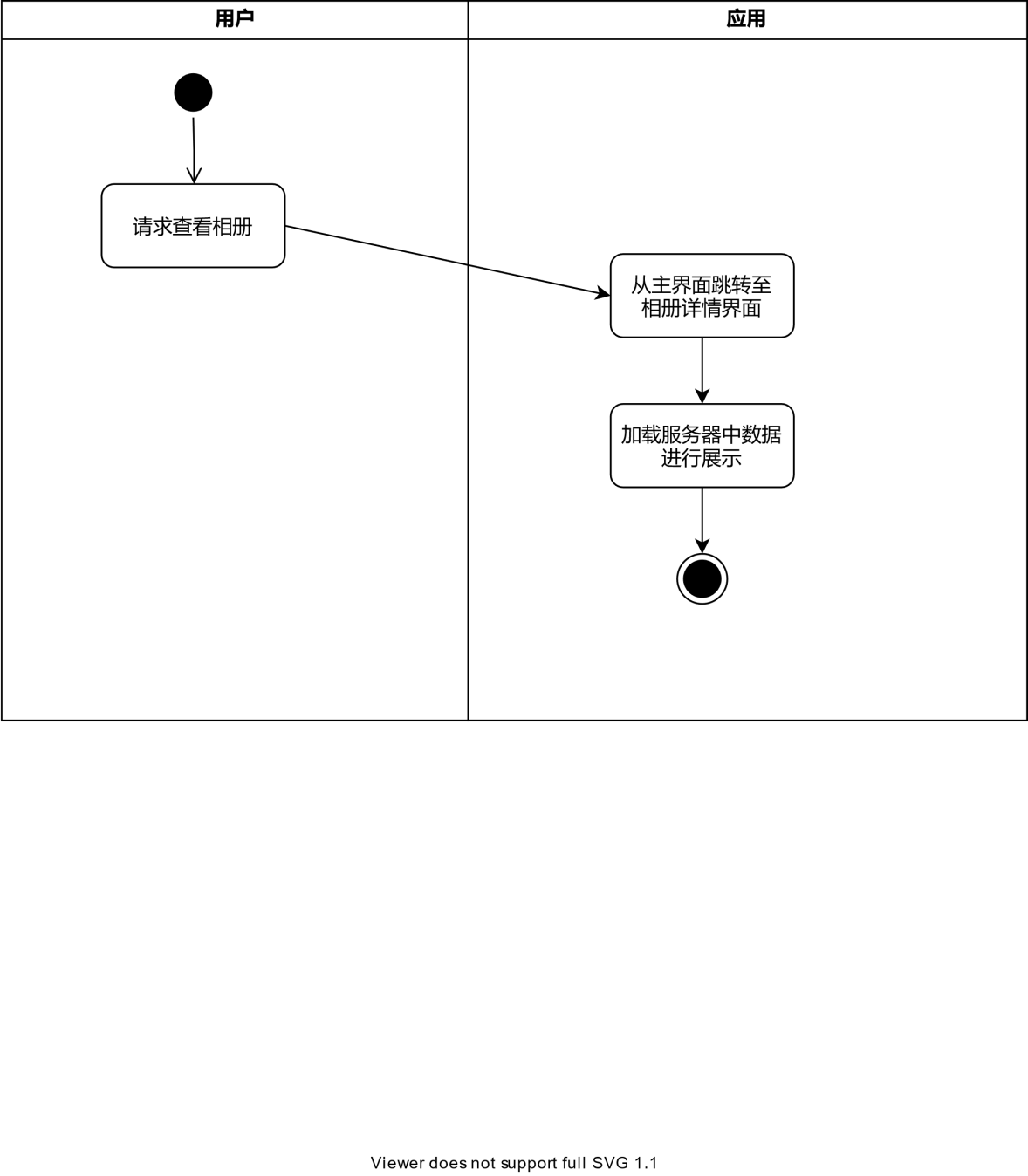
# 2.6 分类模块

# 2.6.1 查看分类

用户在频道界面，可以查看已发布图片的分类频道，查看频道的过程为：

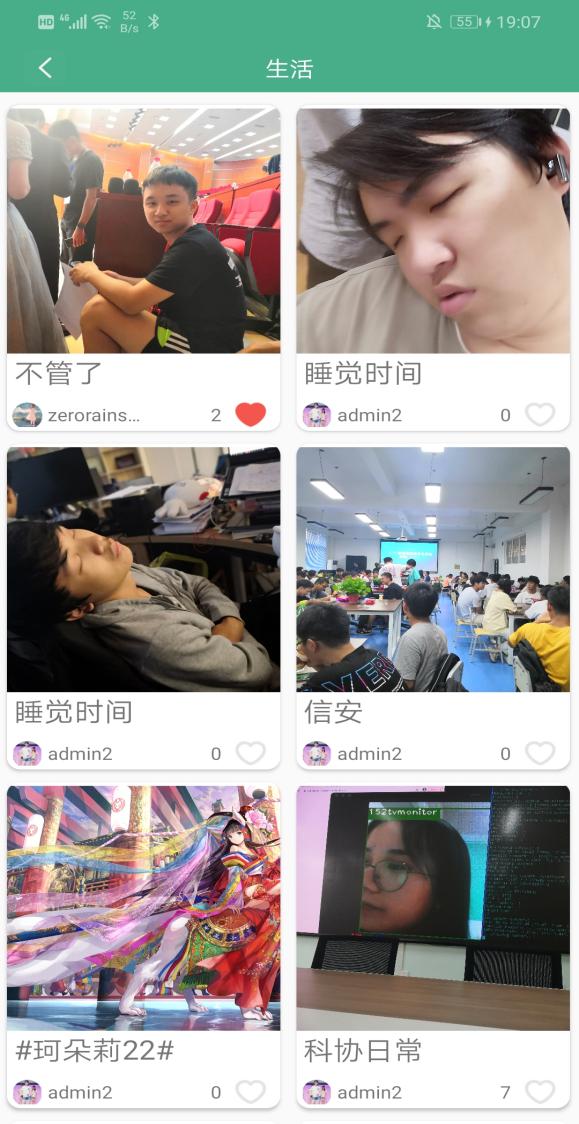
* 用户进入分类模块
* 应用向服务器发送信息请求分类数据
* 服务器接受到请求后，查找数据返回相关信息
* 应用接受到返回信息后进行检验，数据替换和排版，最终展示给用户。

该过程活动图表示如图2.21所示：



**图2.21 查看频道活动图**

实现效果如图2.22、2.23所示：

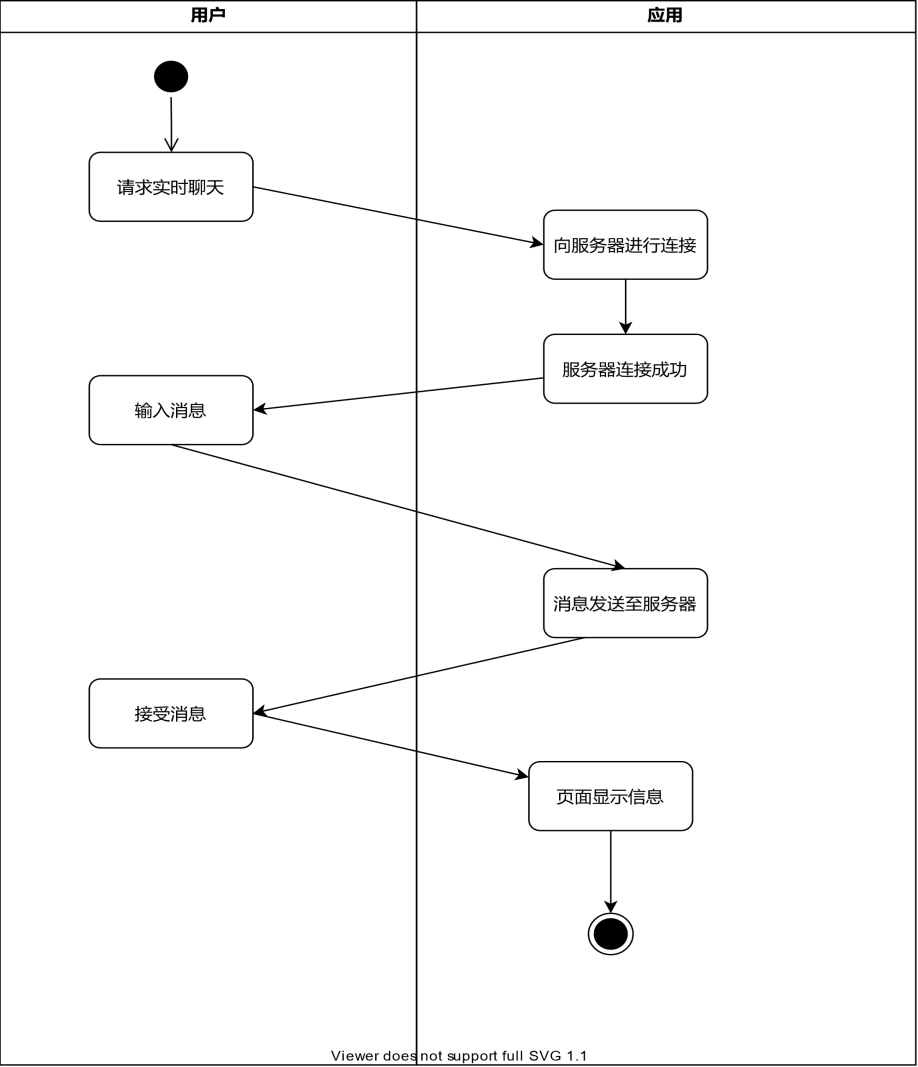
**图2.22 频道界面 图2.23 查看频道图片界面**

# 2.7 看看聊天

用户可以在看看聊天界面发送消息，同时收到其余用户消息，其过程为：

* 用户进入聊天模块，实时发送消息。
* 服务端响应请求，接受用户发送消息，同时socket转发给其余用户。

该过程活动图如图2.24所示：



**图2.24 聊天活动图**

实现效果如图2.25所示：



**图2.25 聊天演示**

# 3. 软件测试

# 3.1 功能测试

# 3.1.1 注册功能测试

测试数据：

|  |
| --- |
| 用户名：zerorainssakura  密码：android |

测试结果：注册成功，有Toast提示注册成功。如图3.1所示：

****

**图3.1 注册界面测试**

# 3.1.2 登录功能测试

* 未注册账号，进行登录

测试结果：登录失败，有Toast提示登录失败，请检查用户名和密码。如图3.2所示：



**图3.2 登录测试1**

* 未输入账号或密码，进行登录

测试结果：登录失败，有Toast提示用户名或密码为空。如图3.3所示：



**图3.3 登录测试2**

* 网络连接断开

测试结果：登录失败，有Toast提示网络请求失败失败。如图3.4所示：



**图3.4 登录测试3**

* 用户名正确，密码正确

测试结果：登录成功，跳转至首页，并弹出Toast提示登录成功。如图3.5、3.6所示

**图3.5 登录测试4 图3.6登录测试成功**

# 3.1.3 修改信息功能测试

修改用户头像、个性签名等，保存修改。测试结果如图3.7、 3.8所示：

**图3.7 修改信息前 图3.8修改信息成功**

# 3.1.4 发布图片测试

选取想要上传的图片，输入图片标题，描述信息和话题以及图像分类，点击发布按钮。测试结果如图3.9、3.10所示：

**图3.9 发布图片测试 图3.10 发布图片成功**

# 3.1.5 评论图片测试

在发现页面，点击图片评论按钮，会弹出输入框，输入“这是一条评论测试”，评论成功后弹出Toast提示“评论成功”，在图片详情页会出现测试评论。测试结果如图3.11、3.12所示：

**图3.11 发布评论测试 图3.12 发布评论成功**

# 3.1.6 点赞和收藏图片测试

点击图片下方的拇指，对说说进行点赞，点赞成功后会点赞数会变化且图标会发生改变。收藏也是一样的效果，测试结果如3.13，3.14所示：

**图3.13 点赞收藏测试前 图3.14点赞收藏测试后**

# 3.1.7 关注发布者测试

在图片的详细页面中，还显示了说说发布者的简单资料，并提供了关注的按钮，当前用户可以通过点击该按钮进行对发布者进行关注或取消关注，测试结果如3.15，3.16所示：

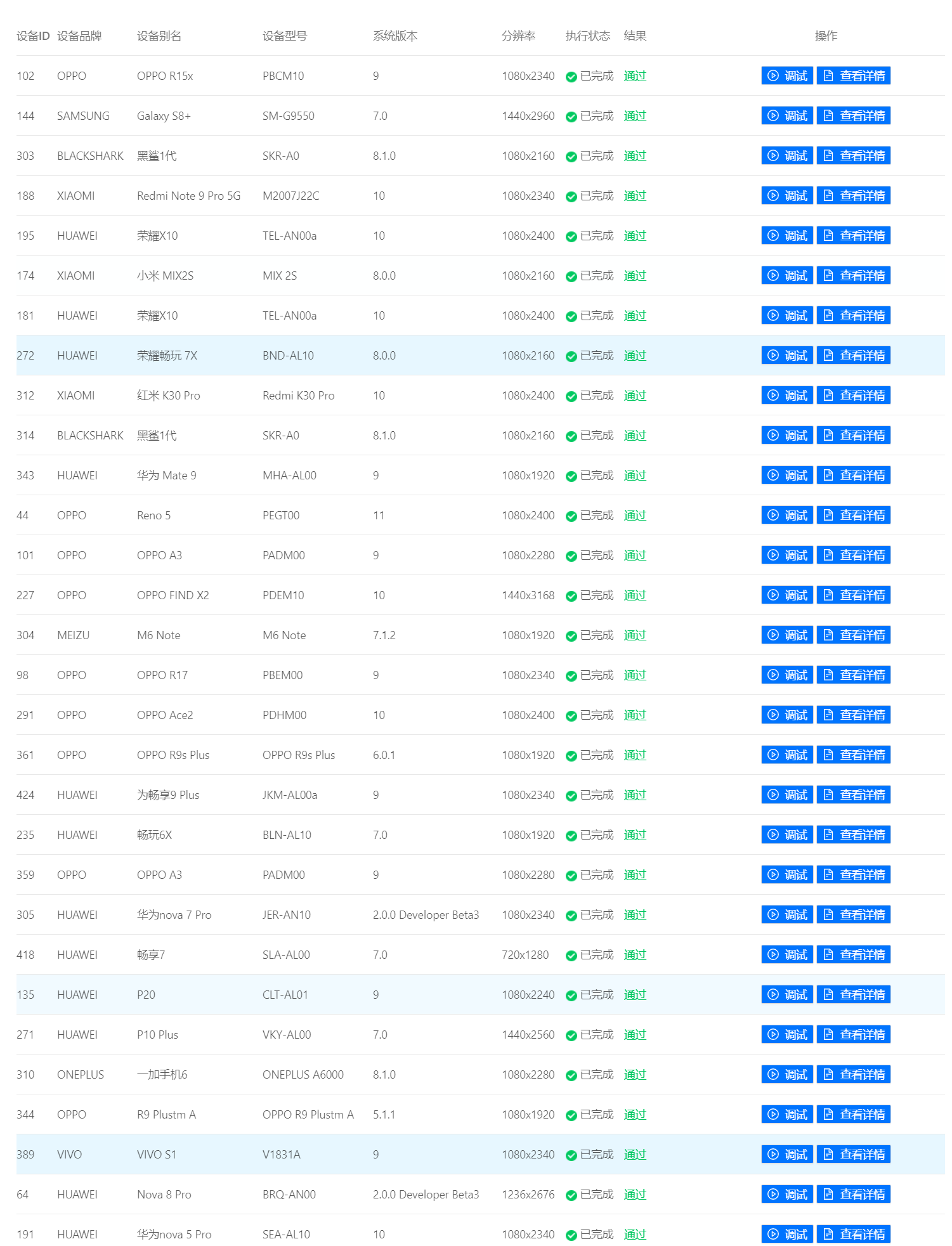
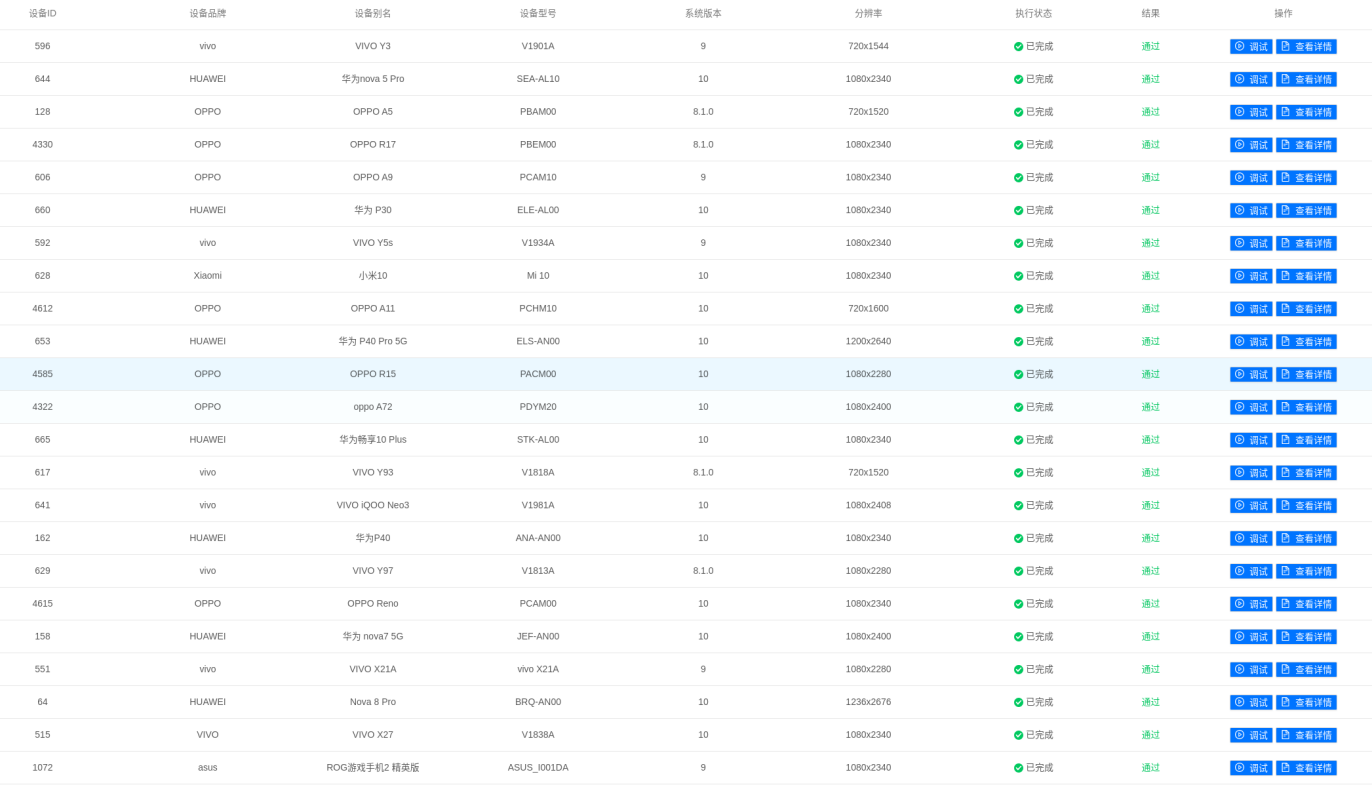
 

**图3.15 用户关注前 图3.16 用户关注后**

# 3.2 兼容性测试

使用Wetest平台进行兼容性自动化测试，选取了50台不同型号和系统的手机进行测试，测试结果如图3.18-3.21所示：

**图3.18 兼容性测试结果1**



**图3.20 兼容性测试结果3**

由以上测试结构可知，我们软件的测试通过率达到了100%，说明此软件的适配率非常不错！

# 3.3 安全性测试



**图3.22 安全性测试报告**

从测试结果可以看出，此APP存在一个低危漏洞，可以通过修改AndroidManifest.xml文件消除此漏洞。总体来看，本应用安全系数相对较高，符合实际使用要求。

# 4 相关技术总结

# 4.1 Android

“看看”是一款基于原生Android Studio开发的安卓图片分享App软件，采用Java语言编写。在整体架构上采用MVC架构模式，项目目录分为Adapter、Common、Entity、Fragment、Utils、View等子目录。

安卓端在布局上采用Fragment+ViewPager实现顶部导航栏切换碎片，提高屏幕空间的利用率，NavigationView实现抽屉侧边个人信息布局，recyclerview和SmartRefreshLayout实现下拉刷新和上拉加载数据效果，自定义不同的recyclerview对应的适配器adapter实现线性布局、九宫格网格布局等，融合CardView、TabLayout 、SmartRefreshLayout等开源组件实现提UI的简洁美观性，整体UI设计规范符合最新的Material Design准则。

安卓端在具体网络请求逻辑中，采用OkHttp、Gson来与后端讨论好的接口进行json数据解序列化。为了由于网络请求耗时的问题导致UI线程阻塞，我通过Handler、Runnable来实现定时通知UI线程更新的方法，进一步提高App的实时效率。在聊天功能模块中，我融合了Socket与Handler子线程的方式，绑定腾讯云后台服务器，实现基本聊天功能。

# 4.2 APP后端

APP后端是基于Flask框架和MySQL数据库进行搭建。

Flask 是一个微型的 Python 开发的 Web 的框架，基于Werkzeug WSGI工具箱和Jinja2 模板引擎。 Flask使用BSD授权。 Flask也被称为“microframework”，因为它使用简单的核心，用extension增加其他功能。Flask没有默认使用的数据库、窗体验证工具。但Flask基于Python快捷方便的特性，能够实现在短时间内实现中小型项目后端的需求。经调查，在同一个项目的前提下，Flask的性能约为SpringBoot的80%，兼顾了性能和功能。

MySQL是一个关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加对数据操作的速度并提高了灵活性。所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

后台服务器接口采用Postman等工具进行测试，图4.2是我们的根据需求分析得出的API接口设计，并写成API接口文档的形式来辅助开发。



**图4.2 团队编写的API接口文档**

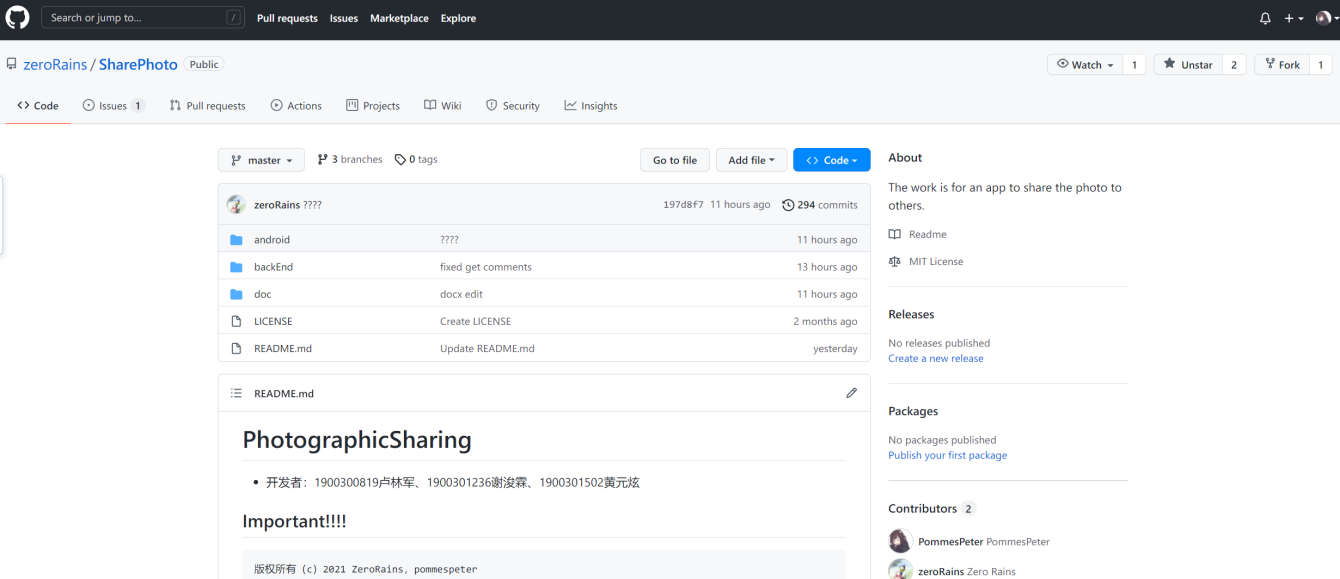
# 4.3 Github代码托管

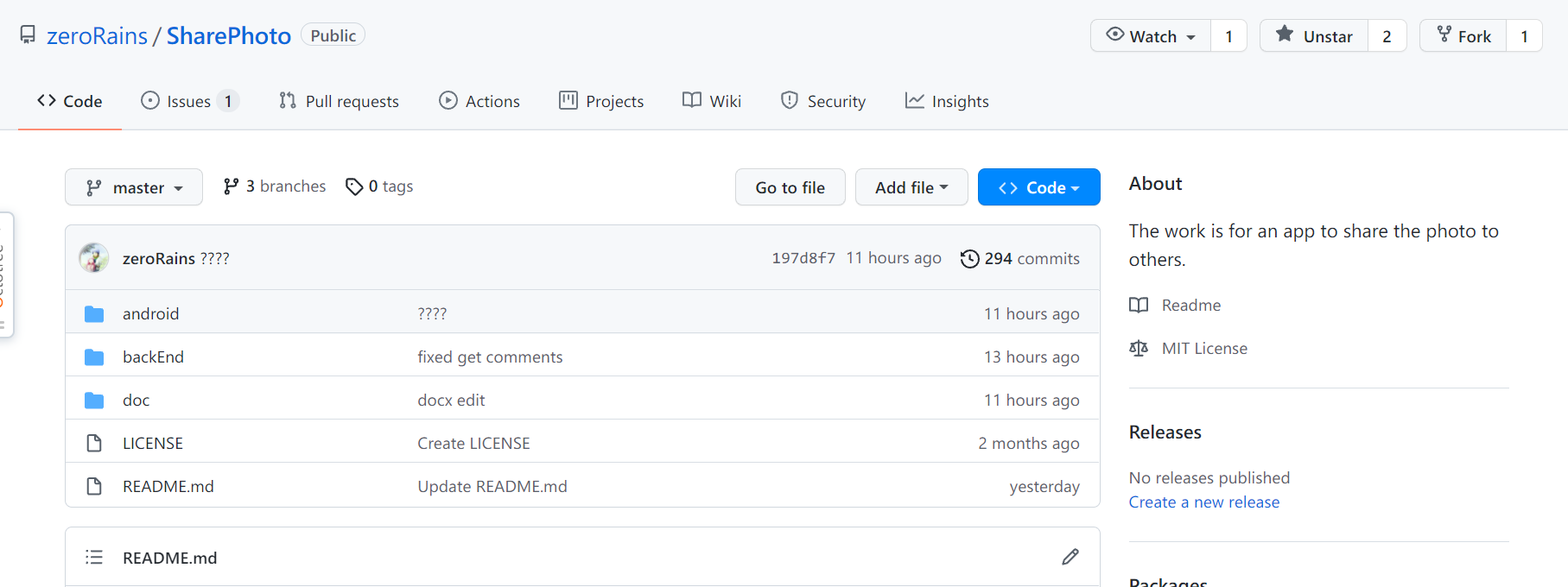
在本次开发中，我们采用Git版本控制系统，基于Github代码托管仓库管理项目的代码，团队成员通过使用Github来进行高效的开发，结合使用fork、pull request、branch等功能极大地提高我们的开发效率，我们的项目Github仓库及部分提交记录如图4.3和图4.4所示。

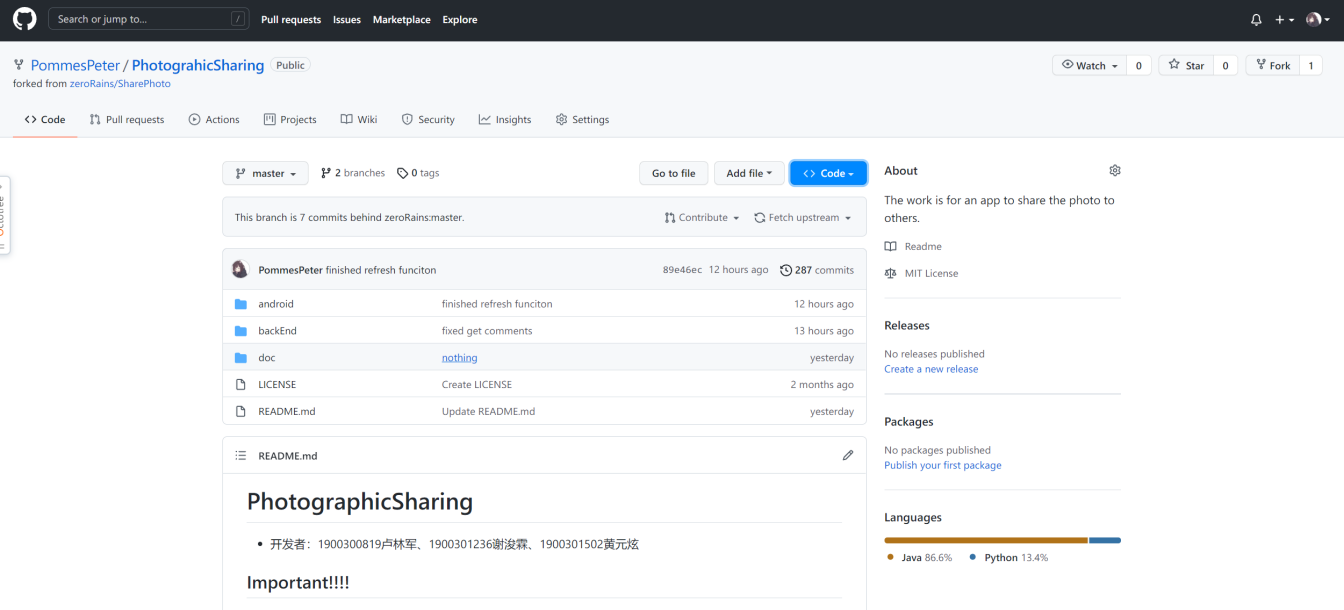
Github仓库地址如下：

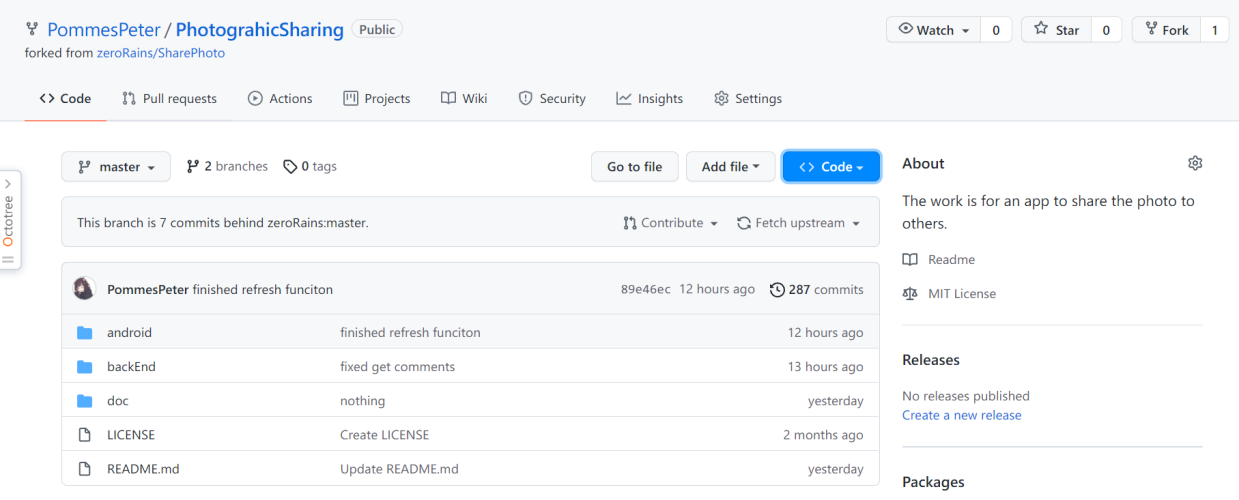
<https://github.com/zeroRains/SharePhoto> (1900301236卢林军)

<https://github.com/PommesPeter/PhotograhicSharing> (1900301236谢浚霖)

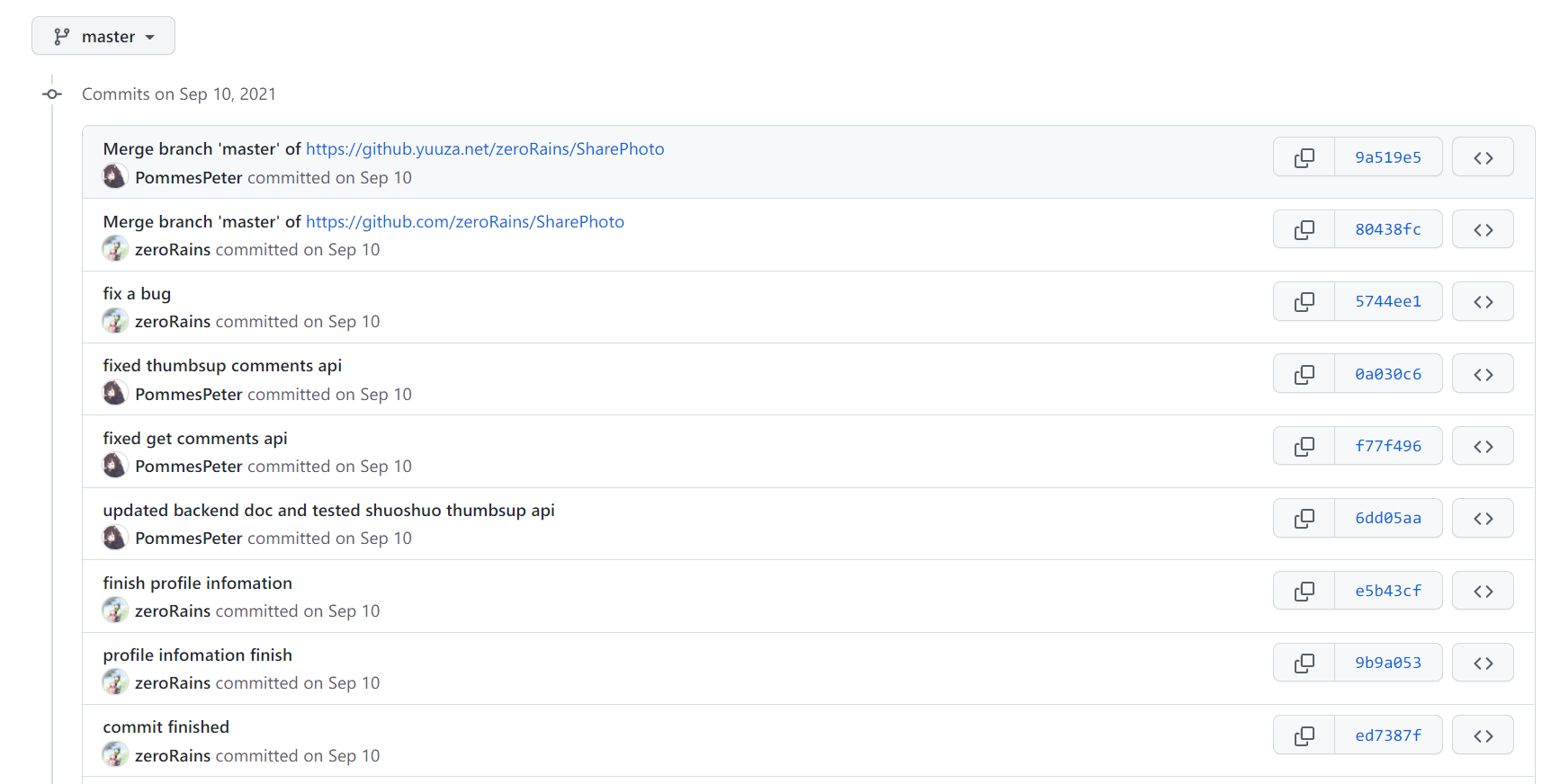








**图4.3 Github仓库**



**图4.4 部分commit记录**

# 5 项目心得总结

本次Android应用程序综合软件开发项目耗时接近一个月，结合了Android、服务器后端、数据库、程序设计与问题求解等学科知识的综合运用。团队三人商量各自分工，对市面上同类型的App进行需求分析，同时，结合现有的UI进行本项目开发，我们希望能够开发一款简易美观、功能实用的轻量级图片分享App ——“看看”，最终确定了安卓端“推荐”、“频道”、“我的”三大核心模块。推荐模块功能主要有：浏览他人分享的图片，集成点赞评论、发布图片等核心社交功能。频道模块可以根据自己需求，按照图片的类别进行分类，提供给用户能够方便查找自己想看图片的功能。为了打造一个“看看”社交专区，我们还搭建了一个简易公共聊天室，聊天室内的内容不会被记录，不会存储到数据库，保障了用户的隐私安全，实现用户在APP端进行聊天的功能。

本次课设过程中，我们的后端选择使用Flask进行APP后端的开发，搭建了基于Flask的后端服务器，Android端根据需求调用相对应的后端接口，如登录注册、发布图片、评论点赞等20个自主编写的API接口，Android端采用图片加载库Glide、网络编程Okhttp3、对Json解序列化Gson成实体类等第三方开源库，结合Handler机制来处理请求后端数据更新UI数据展示等操作，并使用AsyncTask对所有请求进行异步处理，保证了应用程序执行的效率；采用了总多遵循Material Design准则的开源组件，提升UI的简洁美观性。

在近一个月的安卓课设开发周期内，我们分工明确，谢浚霖负责搭建后端Flask后端以及Android端的“个人主页”页面和简易公共聊天室的编码实现，并负责把握总体开发进度；卢林军负责Android端的大部分功能设计，包括“发现”、“频道”、发布等页面的编码实现；黄元炫负责UI界面的选定、报告撰写以及部分Layout页面的实现；后端数据库设计则室由我们团队三个人根据需求分析共同商讨数据的设计。我们在开发的过程中其实遇到了很多问题，例如UI线程和后台线程的使用不明确、OKHttp和Gson如何解析列表实体类、ToolBar如何实现可折叠的工具栏、Fragment如何和Activity进行数据交互、AsyncTask的使用技巧等。最终我们通过查阅官方文档、网络博客等手段进行解决，学习了Handler在耗时任务的用法，学会使用AsyncTask异步进行网络请求，自定义RecycleView的适配器Apater来实现网格等自己喜欢的item布局设计、实现折叠的工具栏等技术难关，收获甚多，加深了我们对安卓系统以及安卓开发的理解。

最后，我们设计的“看看”APP基本达到我们预期的效果，实现了用户所需要的功能，该APP能够投入生产生活使用，实现了前后端分离的开发方式，完成一款轻量级、简洁美观、交互友好的图片分享APP。同时，我们在后期打算进一步开发好友关注、多人聊天、拍照识图等功能，完善“看看”APP的各个功能模块细节。

# 6 参考文献

[1] 任玉刚. Android 开发艺术探索[M]. 北京：电子工业出版社，2015.

[2] 郭霖. 第一行代码——Android (第 2 版) [M]. 北京：人民邮电出版社，2016.

[3] Cay S Horstmann. Java 核心技术·卷 I[M]. 北京：机械工业出版社，2016.

[4] Google.Android开发指南[EB/OL]. https://developer.android.google.cn/guide. 2020-04-06.

[5] Muyangmin. Glide v4 (高效的Android图片加载库) [EB/OL]. https://muyangmin.github.i

o/glide-docs-cn/. 2020-07.

[6] Android API Reference. <https://developer.android.com/reference>

[7] Android多线程：手把手教你使用AsyncTask. https://www.jianshu.com/p/ee1342fcf5e7