图片包含 徽标

描述已自动生成

Android综合设计

题 目： 基于Android的图片分享软件

院（系）： 计算机与信息安全学院

专 业： 软件工程

组 长：

组 员：

组 员：

指导老师：

2021年12月27日

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 小组选题 | 基于Android的课堂签到系统 | | | | |
| 团队成员 | 姓名 | 性别 | 学号 | 分工 | 工作量 |
|  |  |  | 系统架构设计，使用bmob数据库相应api进行网络请求，前台页面设计与实现，系统详细设计编写 | 40% |
|  |  |  | 需求分析，数据库设计，部分页面设计开发，部分接口对接，部分报告撰写 | 30% |
|  |  |  | 参与个人界面，注册，登录，图片分享，负责报告前言，可行性分析，系统测试编写 | 30% |

# 摘要

随着互联网社交平台的发展，网络上发表状态、分享照片已经变成了人们日常生活中不可或缺的一部分，图片分享更是得到了人们的亲睐。本文主要是在Android系统上实现了照片分享应用程序的设计与实现。

本论文采用了Android 开源系统技术，利用Java语言和Andorid Sutdio开发工具对app代码进行编写。同时给出了详细的需求分析、系统设计过程、部分界面截图及主要的功能流程图，本文还对开发过程中遇到的问题和解决方法进行了详细的讨论，该系统集注册、登录、收藏等功能于一体， 性能良好，在Android系统中能独立运行。

通过测试证明，基于Android平台的软件开发简单，使用方便简洁，可以给用户带来更方便更真实的体验。本系统成功实现了项目的功能性需求，测试阶段无明显漏洞，成功为图片分享创造了一个良好的平台。

**关键词：**Android、图片分享、社交、收藏

# 前言

## 项目背景

在网络上分享自己的状态已经成为了我们日常生活中不可或缺的一部分， 伴随着我国互联网的迅速发展，图片分享更是得到了人们的亲睐。不知不觉中，手机相机已成为人们最为普遍关心的手机硬件。要高像素的相机手机的目的，就是为了能够随时随地的记录下自己当时看到的情形，而后再把要分享的照片通过QQ、微信、微博等社交平台分享给朋友。据全球社交营销代理机构WeAresocial对世界大型网络社交的调查显示，在QQ微信用户中逾80%用户都上传过照片。如今是互联网分享时代，而我们的随手一拍，便能将图片分享给朋友、陌生人。

## 项目意义

Android的手机一般都带有高分辨率的摄像头以提供用户进行拍照，相信朋友们的手机中一-定有不少的照片，下面这个应用程序可以浏览手机中的照片，还可以分享喜欢的照片给朋友。一款专业的照片分享软件是手机中必不可少的要素，几乎市场上所有的手机都拥有一个浏览图片软件,通过软件可以对照片中一些必要的元素进行人性化的设置，同时在操作的时候也能得到一定的便利，但是不能满足用户的所有要求，所以网上就出现了各种各样的照片分享软件,来满足不同用户的不同要求。所以本课题就致力于开发出一款基于Android手机平台的照片分享软件，给大家分享生活的乐趣带来更多的方便。

# 概要设计

## 2.1、可行性分析

**2.1.1 技术可行性**

此图片分享系统为分享日常生活的用户而设计，具备了基本满足用户分享需求的功能。使用Android Studio来开发本系统，通过学习，结合学生水平完成了本次课设要求。

**2.2.2 经济可行性**

Andriod Studio应用程序开发的门槛并不高，因为Google已经为Andriod Studio应用程序开发提供了免费而且跨平台的集成开发环境，而对于java、bmob数据库相关技术，网上也有健全的学习路线和设计文档，只是需要花一些时间成本，所以系统开发成本低。后期如果需要真正发布到应用市场上，只需要将应用部署到更加高效的服务器上面去即可使用了。

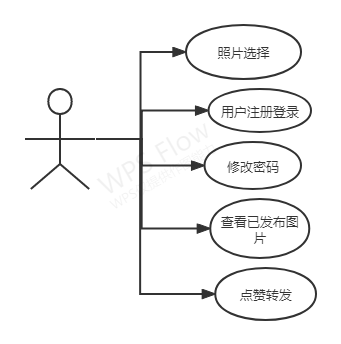
系统开发运用到的仅仅是开发者的电脑，可以通过图片分享系统吸引更多用户，增强用户流动性，在系统中扩大社交圈，增强商家入驻，提高系统经济流量。

**2.2.3 操作可行性**

各种分享系统出现在社会市场中，有通过视频分享日常生活的，通过音乐分享生活的，图片分享系统以另一种形式来分享自己。这些系统软件出现的目的都是为了让用户能够在日常生活中分享自己的想法，大家可以相互交流，给生活带来更多的趣味。

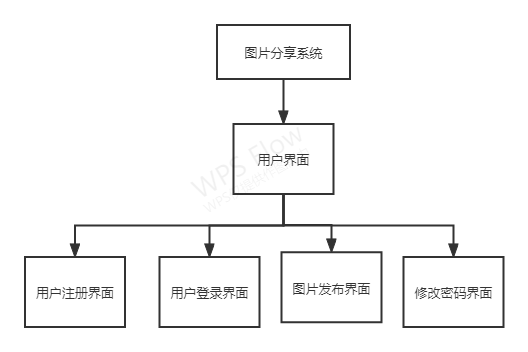
## 2.2、需求分析

**2.2.1软件应用需求**

图2- 1图片分享系统应用需求图

在图片分享软件正常运行中，用户选择发布图标进行图片发布，将在发布界面浏览图片进行选择或者是直接拍摄图片进行发布。用户可以在将要发布的图片下方添加文字设置一个主题，并用具体文字进行对图片的描述，发布成功后可以在首页看见自己刚刚发布的图片，并可以在图片下方进行点赞转发操作。

**2.2.2软件功能需求**

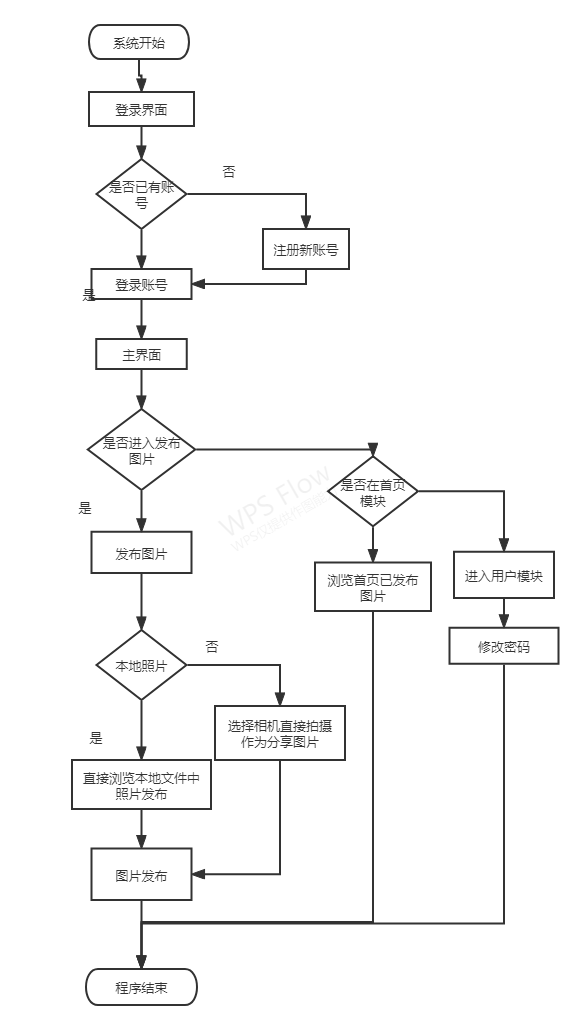
图2-2 图片分享系统性能需求图

图片分享系统是用来查找自己想要发布出去的图片，用来记录自己美好时光并通过系统来和大家一起欣赏图片的一种工具，能很好和大家分享自己日常时光。

**2.2.3 系统界面需求**

系统界面布局合理，颜色舒适，风格简约，控制按钮友好，方便用户理解以及进行操作。实现功能不重复，满足用户需要。

## 2.3、系统流程设计

 图2-3 系统流程图

点击运行程序后，进入登录界面，是否有账号，如果有账号就直接登录账号，如果没有账号就注册账号，登录后进入主界面。可以点击发布照片，可以选择本地图库照片，或者直接调用系统相机进行拍照，在点击确认发布即可发布照片。还可以进入首页浏览他人发布的照片，或者进入用户模块可以选择修改密码等操作。

## 2.4、数据库设计

数据库表设计：

表 2-1User表设计

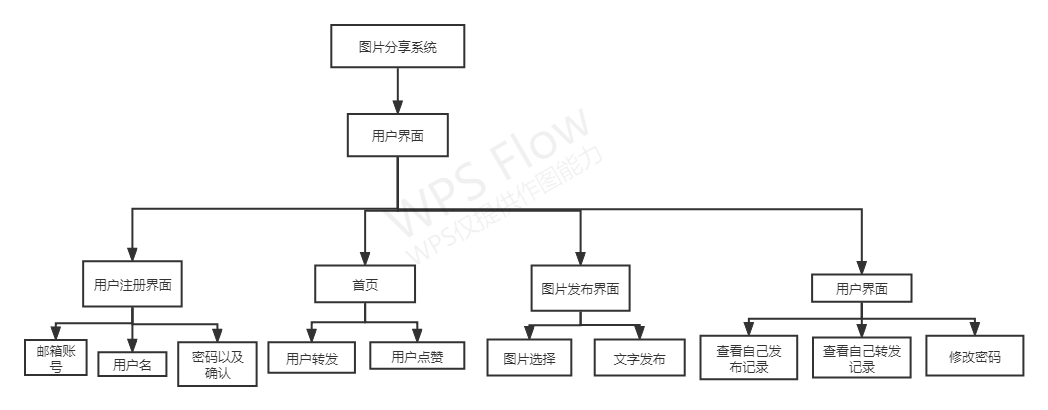
|  |  |
| --- | --- |
| 名称 | 类型 |
| nickName | String |
| password\_record | String |
| username | String |
| headPicture | BmobFile |
| imagePath | String |

表 2-2Card表设计

|  |  |
| --- | --- |
| 名称 | 类型 |
| username | String |
| nickName | String |
| headPicture | BmobFile |
| Picture | BmobFile |
| Description | String |
| LikeNumber | Integer |
| LikeState | Boolean |
| ShareState | Boolean |

其中Username和Password表示用户和登录密码，Nickname表示用户的昵称。在imagePath可以看见图片的存储路径，方便管理员进行查找，在Description中存储的是用户对自己分享的图片的描述，headPicture表示用户的头像存储路径。

## 2.5、系统架构设计

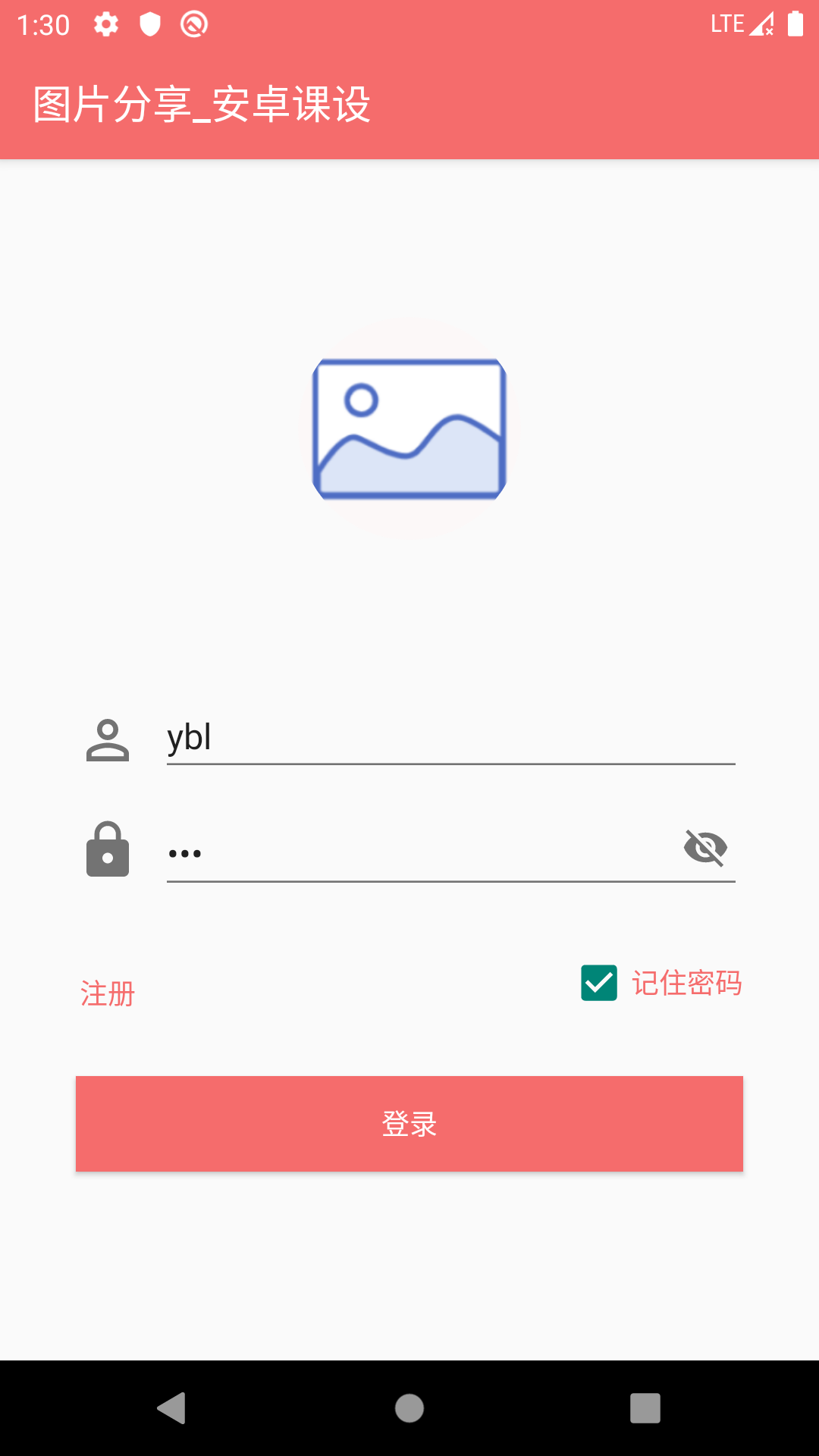
图2-4 系统架构图

用户通过进入系统的登录界面选择注册账号或者是利用已有账号直接登录，登录后进入系统可以直接在首页上浏览以及发布了的图片，并进行转发或者点赞操作；或者选择自己发布想要享的图片并加上文字来描述图片或者是表达自己的心情。在用户模块中，用户可以查看自己发布过的图片记录，转发和点赞记录，或者进行自己账户密码的修改。

# 详细设计与实现

## 3.1、启动页：

启动页设计为登录页面。程序的开端从此处开始。在登录页面通过textView来获取输入的账号密码。通过bmob数据库查询验证用户密码是否正确。同时设置记住密码功能。通过复选框实现。设计注册按钮用以跳转注册页面。代码详见附录。效果如下：



## 3.2、注册页面：

注册页面设计以数据库的存储和图像选取为主要功能。通过TextView来获取要输入的账号密码用户名。通过设计弹窗绑定不同的事件来决定不同的图片添加方式。分别调用不同的相册权限来实现拍照和剪裁。调用系统api来实现拍照功能。代码详见附录，效果如下：





## 3.3、主页面

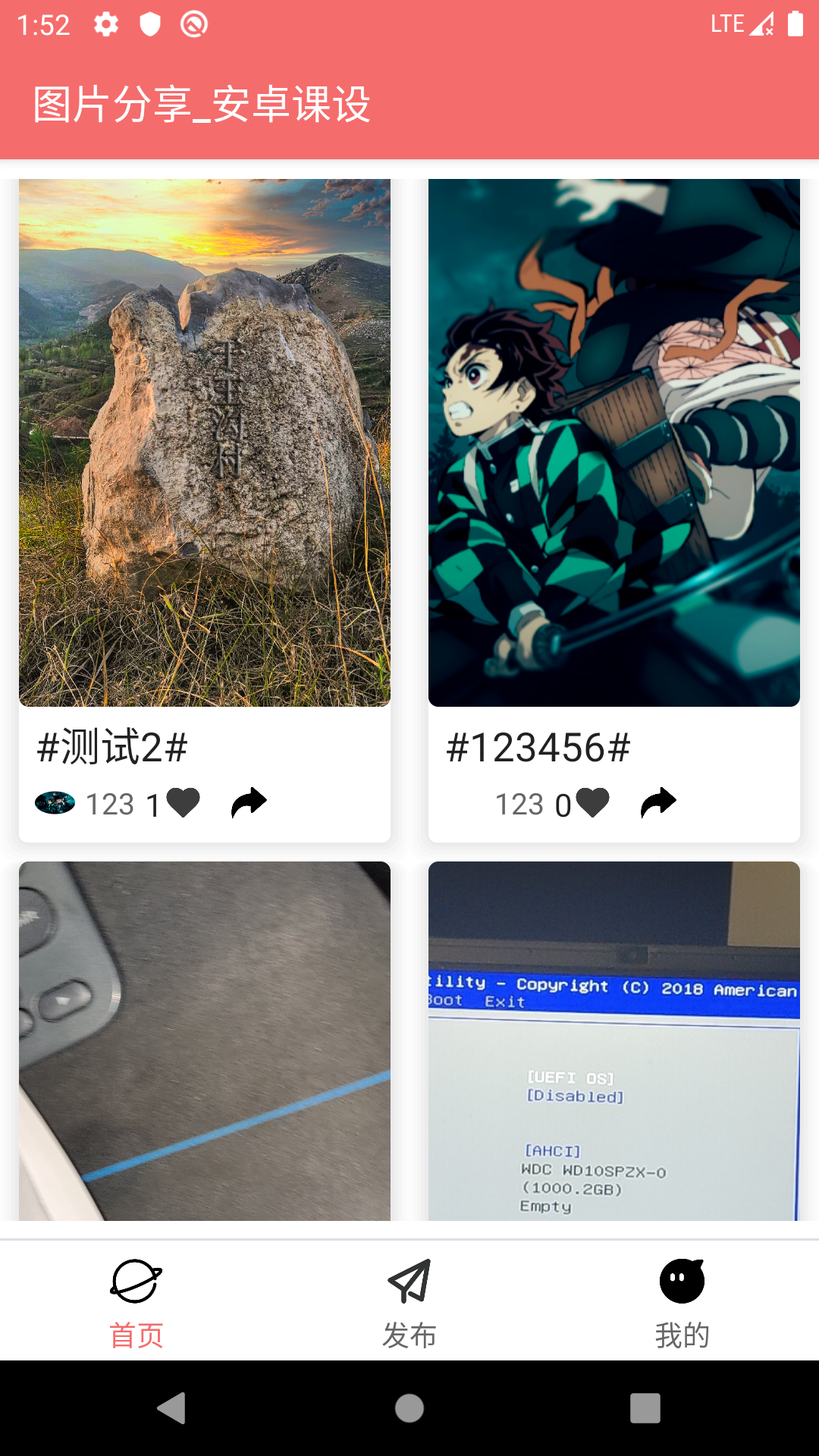
### 3.3.1基础页面

由于项目使用Fragment。因此主页面仅有按钮控件用以控制事件显示不停的页面。详细代码见附件。效果图如下：



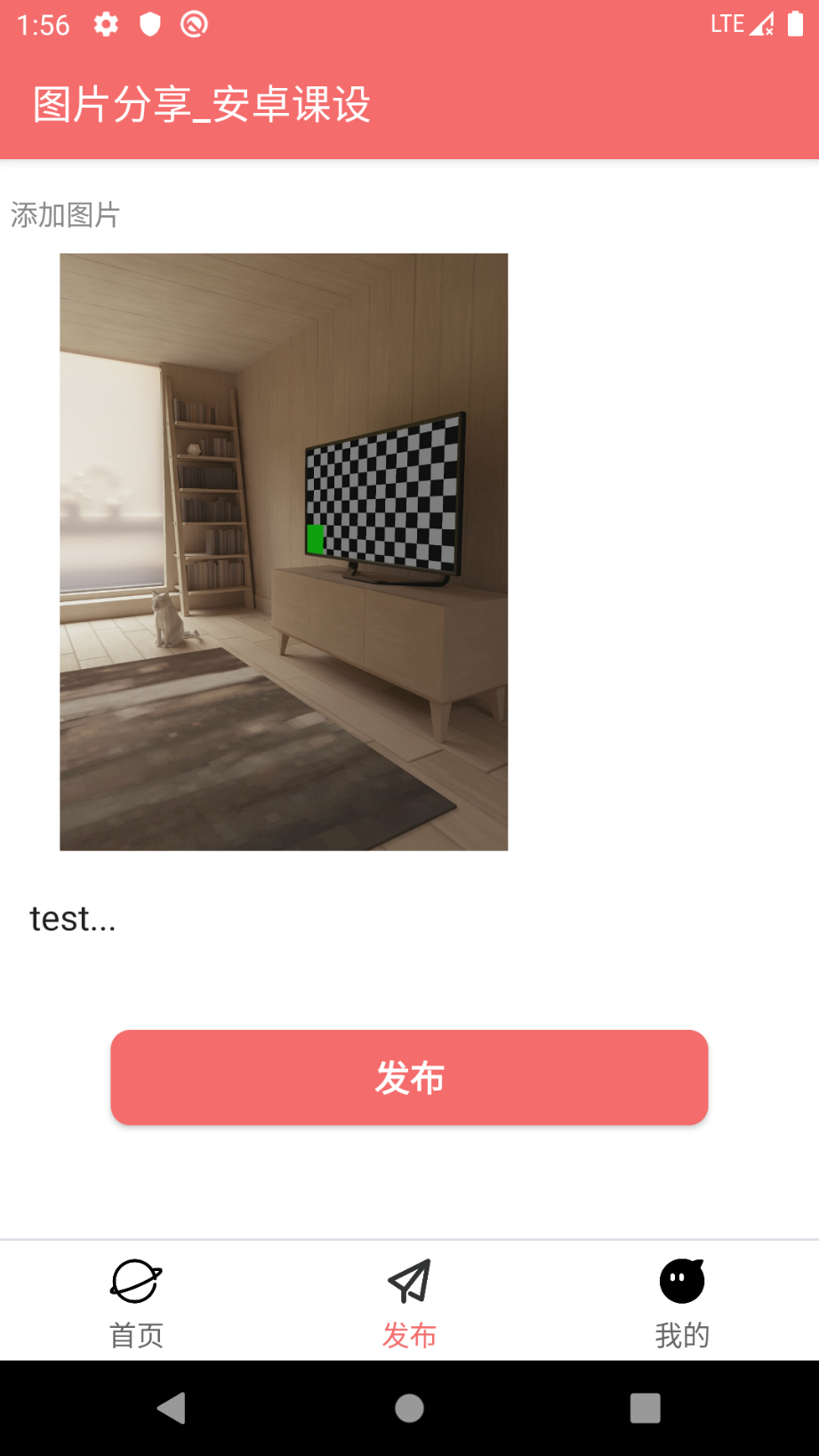
### 3.3.2首页

首页通过RecyclerView控件来实现子项效果。通过子项来定义每项的页面布局。显示由图片标题、发布者用户名、头像、收藏按钮、分享按钮组成。逻辑由继承RecyclerView.Adapter来实现适配器。通过Card类来对接数据详见数据表。详细代码见附录。效果如下：



### 3.3.3、发布功能：

发布功能与注册页面的头像选择和文字选择实现方式相似，选取图片采用相同的逻辑。具体代码见附件。效果如下：



4.

## 3.4我的页面：

我的页面主要由与用户相关的内容组成。设计的时候考虑到收藏作品的保存。喜欢作品的保存。已经更改密码等问题。分别涉及不同的控件用以细化功能。在相应的页面来实现相应的布局。主要功能有，更换头像,通过定义头像点击事件来实现。修改密码。通过跳转新的页面，以简单版注册来替换布局。相应的文件列表，仿照首页布局，尽在数据获取方面做逻辑处理。切换登录和退出。通过跳转登录页面实现。详细代码见附录：具体效果如下：



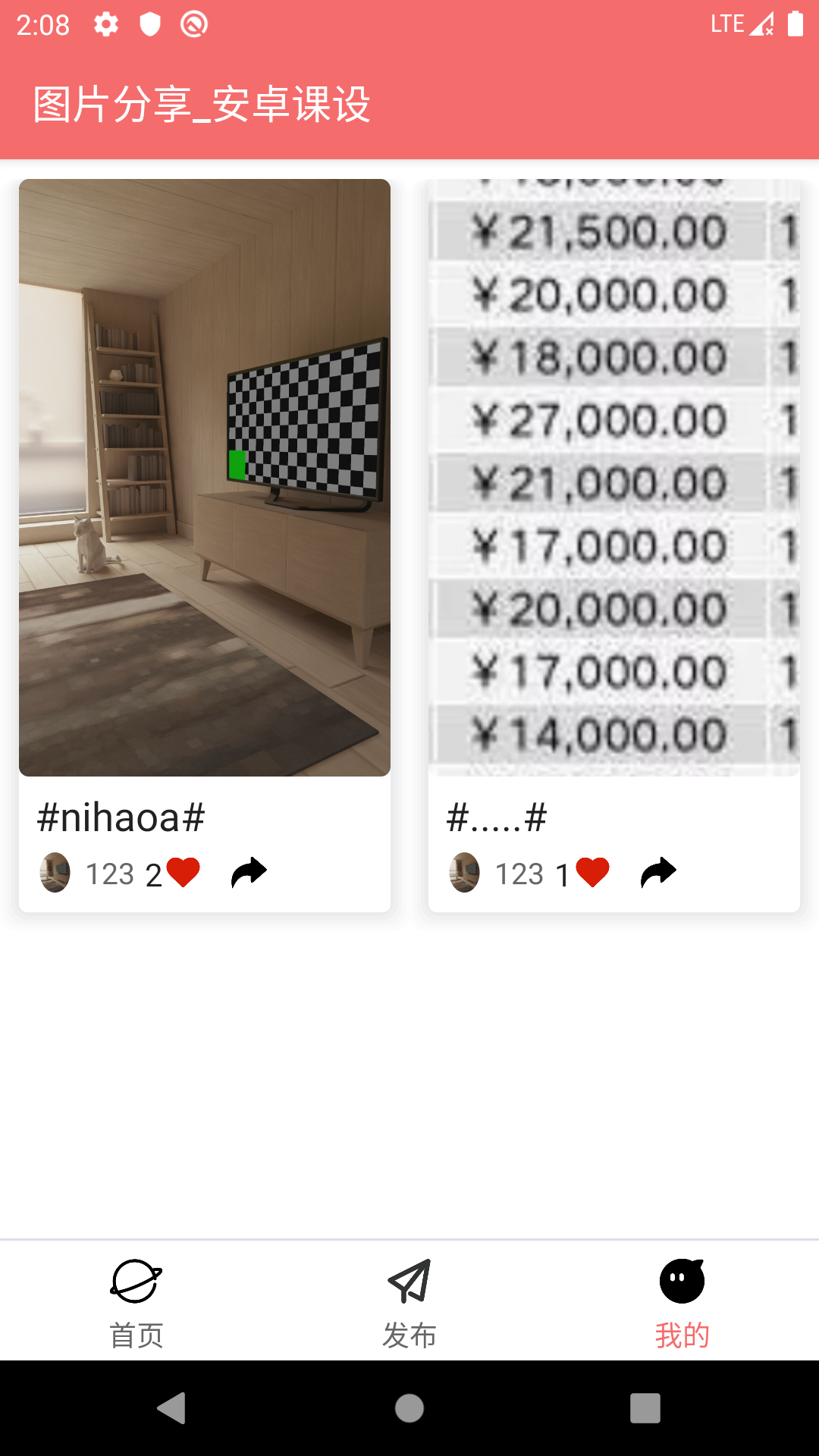
主页面



修改密码界面



开发信息界面



点过喜欢作品界面（转发和收藏同）



帮助与反馈界面

# 第4章 系统测试

本章节对做出的图片分享软件进行功能测试，编写的目的在于总结测试阶段的测试和分析测试结果，得到对软件质量的评价，描述软件是否符合预期需求。同时分析软件存在的缺陷，为修复 bug和预防后续开发可能会引发更多缺陷提供建议。

## 4.1、测试环境和配置

测试环境：社相机必须处在可以连接外网的网络环境之下，网络环境良好。对于某些安全相关的网络，会对联网时的加密协议有所要求，比如SMTP协议使用SSL加密的情况下无法联通服务器，这回使测试结果不够全面，所以这样的网络也是不行的。建议本项目测试环境在WiFi情况下进行。

测试设备配置：应该安装在不低于Android10版本的系统中;手机需要接入网络, 必须有网络硬件支持;为了能够让软件能够流畅的运行，手机需要有至少512M内存，CPU主频最少为1GHz。

## 4.2、测试方法

本次测试将要使用到黑盒测试方法，主要针对个功能某块的实现情况，通过比较需求设计阶段功能块的描述与测试中系统的实际功能实现，来了解系统是否达到了预期目标，以及了解系统存在的问题，做出缺陷分析。设计开发阶段测试，采用白盒测试方法，排查代码设计实现逻辑错误、以及容易忽视的漏洞。功能测试阶段，采用黑盒测试方法，与项目需求阶段进行比较，观察功能实现缺陷与不足。

## 4.3、测试过程

表4-1 系统测试表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用户登录页 | 正确登录 | 输入正确账号密码 | 成功登录 | 同预期 | 通过 |
| 错误登录 | 输入错误账号密码 | 登录失败 | 同预期 | 通过 |
| 新用户注册 | 点击注册，填写全部信息，并且信息正确 | 成功注册，无报错 | 同预期 | 通过 |
| 点击注册，填写全部信息，两次密码不一致 | 注册失败 | 同预期 | 通过 |
| 首页 | 查看已分享图片 | 点击已分享图片 | 显示图片信息 | 同预期 | 通过 |
| 上传图片 | 编辑文字，并上传图片分享 | 他人能够看到所分享的图片，并且能查看图片信息 | 同预期 | 通过 |
| 查看个人信息 | 点击“我” | 能查询到正确个人信息 | 同预期 | 通过 |

## 4.4、测试结论

通过测试可以得知，此软件能够完成设计时所提出的功能需求，各个模块测试得以通过，符合要求。然而，在测试过程中，也会发现一系列的待解决的问题，针对这些问题，应该通过更多更详细的测试及用户使用反馈加以发现和改正，以免带来不必要的麻烦。

# 第5章 心得体会

基于安卓的手机图片分享客户端的研究、设计与实现，从最开始的技术储备到需求分析,再到最后的设计与实现，全文详尽描述了一套安卓软件从开始准备到最后成型应用的整个过程，最终实现的手机图片分享客户端和服务器端的实现。在主界面设计上，设计了三个主要模块界面，通过设计在下面的按钮切换三个主要功能模块“首页”、“发布”、“我的”。用户在操作时更方便，而且省略掉一些不是很实用的功能，化繁为简，也让用户节省了许多加载中浪费的流量。

当然由于时间仓促还有很多不足之处，值的改进的地方，比如（1）没有设置图片剪辑剪裁功能。（2）没有评论互动功能，不能让用户点评他人分享的照片。（3）没有加入位置信息，让用户更轻松简单的找到自己周围的人。

通过本学期的安卓课程设计，学到了很多知识的细节，比如：Android Studio环境搭建，初级控件的使用，数据在云端服务器上存储，了解了事件，明白了如何检测点击等等多方面的细节。

通过本学期的安卓课程设计，发现这其中需要的很多知识我们没有接触过，去网上查资料的时候发现我们前边所学到的仅仅是皮毛，还有很多需要我们掌握的东西我们根本不知道。同时也发现有很多已经学过的东西我们没有理解到位，不能灵活运用于实际，不能很好的用来解决问题，这就需要我们不断的大量的实践，通过不断的自学，不断地发现问题，思考问题，进而解决问题。在这个过程中我们将深刻理解所学知识，同时也可以学到不少很实用的东西。

纸上得来终觉浅，绝知此事要躬行。学习不在于知，而在于行，学习的最终目标还在于实践，还需要知行合一，在学中知，在行中明。

# 参考文献

[1]Android开发应用实战详解[M].中国铁道出版社.张元亮, 2011

[2]Android下基于用户识别的文件共享设计与实现 [J].吕少伟,李永忠,王艳敏.电子设计工程. 2014(04)

[3]Android中Camera类库分析及其典型应用 [J].胡江楠,刘高平.浙江万里学院学报. 2014(01)

[4]Android平台移动社交网络的设计与实现[D].王戴钰.大连海事大学. 2014

[5]Google Android程序设计指南[M].电子工业出版社.杨文志.2009

[6]疯狂Android讲义[M].电子工业出版社.李刚.2015

# 附录：各核心业务模块程序清单

#### 1、1登录页：

<RelativeLayout

android:id="@+id/keyboard\_"

android:clickable="true"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="48dp"

android:layout\_weight="1">

<ImageView

android:layout\_width="112dp"

android:layout\_height="112dp"

android:layout\_centerInParent="true"

android:src="@mipmap/ic\_launcher" />

</RelativeLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="43dp"

android:layout\_marginTop="48dp">

<ImageView

android:layout\_width="32dp"

android:layout\_height="43dp"

android:layout\_centerVertical="true"

android:layout\_marginStart="38dp"

android:layout\_marginBottom="8dp"

android:src="@drawable/ic\_baseline\_person\_outline\_24"

android:layout\_marginLeft="38dp" />

<EditText

android:id="@+id/et\_username"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_gravity="center"

android:layout\_marginStart="80dp"

android:layout\_marginEnd="38dp"

android:layout\_marginBottom="1dp"

android:hint="邮箱/账号"

android:inputType="text"

android:maxLines="1"

android:layout\_marginLeft="80dp"

android:layout\_marginRight="38dp" />

</RelativeLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="43dp"

android:layout\_marginTop="16dp">

<ImageView

android:layout\_width="32dp"

android:layout\_height="40dp"

android:layout\_alignParentBottom="true"

android:layout\_marginStart="38dp"

android:layout\_marginLeft="38dp"

android:layout\_marginBottom="8dp"

android:src="@drawable/ic\_baseline\_lock\_24" />

<EditText

android:id="@+id/et\_pwd"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_marginStart="80dp"

android:layout\_marginEnd="38dp"

android:layout\_marginBottom="1dp"

android:hint="密码"

android:inputType="textPassword"

android:maxLines="1"

android:layout\_marginLeft="80dp"

android:layout\_marginRight="38dp" />

<ImageView

android:id="@+id/iv\_pwd\_switch"

android:layout\_width="24dp"

android:layout\_height="24dp"

android:layout\_alignParentEnd="true"

android:layout\_alignParentBottom="true"

android:layout\_marginEnd="45dp"

android:layout\_marginBottom="15dp"

android:src="@drawable/ic\_baseline\_visibility\_off\_24"

android:layout\_alignParentRight="true"

android:layout\_marginRight="45dp" />

</RelativeLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="40dp"

android:layout\_marginTop="16dp">

<TextView

android:id="@+id/tv\_sign\_up"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentStart="true"

android:layout\_alignParentBottom="true"

android:layout\_alignParentLeft="true"

android:text="注册"

android:textColor="?attr/colorPrimary"

android:clickable="true"

android:layout\_marginLeft="40dp"

android:layout\_alignParentRight="true"

/>

<CheckBox

android:id="@+id/cb\_remember\_pwd"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_alignParentBottom="true"

android:layout\_alignBaseline="@id/tv\_sign\_up"

android:layout\_marginRight="38dp"

android:layout\_marginBottom="4dp"

android:text="记住密码"

android:layout\_marginEnd="38dp"

android:textColor="?attr/colorPrimary"

/>

</RelativeLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="32dp"

android:layout\_weight="1">

<Button

android:id="@+id/bt\_login"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_marginStart="38dp"

android:layout\_marginTop="32dp"

android:layout\_marginEnd="38dp"

android:background="?attr/colorPrimary"

android:text="登录"

android:textColor="@android:color/white" />

</RelativeLayout>

#### 1.2登录逻辑：

btLogin.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View view) {

LoadingDialog laoding = new LoadingDialog(mContext,0);

laoding.show();

String spFileName = getResources()

.getString(R.string.shared\_preferences\_file\_name);

String accountKey = getResources()

.getString(R.string.login\_account\_name);

String passwordKey = getResources()

.getString(R.string.login\_password);

String rememberPasswordKey = getResources()

.getString(R.string.login\_remember\_passoword);

//本地存储读取保存的用户名密码

SharedPreferences spFile = getSharedPreferences(

spFileName,

Context.MODE\_PRIVATE);

SharedPreferences.Editor editor = spFile.edit();

if(cbRememberPwd.isChecked()) {

String password = etPwd.getText().toString();

String account = etUsername.getText().toString();

editor.putString(accountKey, account);

editor.putString(passwordKey, password);

editor.putBoolean(rememberPasswordKey, true);

editor.apply();

} else {

editor.remove(accountKey);

editor.remove(passwordKey);

editor.remove(rememberPasswordKey);

editor.apply();

}

final \_User user= new \_User();

user.setUsername(etUsername.getText().toString());

user.setPassword(etPwd.getText().toString());

//登录部分

user.login(new SaveListener<\_User>() {

@Override

public void done(\_User bmobUser, BmobException e) {

if (e==null) {

\_User user = BmobUser.getCurrentUser(\_User.class);

if (user.isLogin()) {

user = BmobUser.getCurrentUser(\_User.class);

} else {

Snackbar.make(view, "尚未登录，请先登录", Snackbar.LENGTH\_LONG).show();

}

Intent intent=new Intent(LoginActivity.this, MainActivity.class);

startActivity(intent);

} else {

laoding.dismiss();

Snackbar.make(view, "登录失败：" + e.getMessage(), Snackbar.LENGTH\_LONG).show();

}

}

});

}

});

//密码是否可见部分

ivPwdSwitch.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

bPwdSwitch = !bPwdSwitch;

if (bPwdSwitch) {

ivPwdSwitch.setImageResource(

R.drawable.ic\_baseline\_visibility\_24);

etPwd.setInputType(

InputType.TYPE\_TEXT\_VARIATION\_VISIBLE\_PASSWORD);

} else {

ivPwdSwitch.setImageResource(

R.drawable.ic\_baseline\_visibility\_off\_24);

etPwd.setInputType(

InputType.TYPE\_TEXT\_VARIATION\_PASSWORD |

InputType.TYPE\_CLASS\_TEXT);

etPwd.setTypeface(Typeface.DEFAULT);

}

}

});

//加载已保存账号密码

String spFileName = getResources()

.getString(R.string.shared\_preferences\_file\_name);

String accountKey = getResources()

.getString(R.string.login\_account\_name);

String passwordKey = getResources()

.getString(R.string.login\_password);

String rememberPasswordKey = getResources()

.getString(R.string.login\_remember\_passoword);

SharedPreferences spFile = getSharedPreferences(

spFileName,

MODE\_PRIVATE);

String account = spFile.getString(accountKey, null);

String password = spFile.getString(passwordKey, null);

Boolean rememberPassword = spFile.getBoolean(

rememberPasswordKey,

false);

if (account != null && !TextUtils.isEmpty(account)) {

etUsername.setText(account);

}

if (password != null && !TextUtils.isEmpty(password)) {

etPwd.setText(password);

}

cbRememberPwd.setChecked(rememberPassword);

}

private void toast(String s) {

Toast.makeText(LoginActivity.this, s, Toast.LENGTH\_SHORT).show();

}

#### 2.1注册页面：

<RelativeLayout

android:id="@+id/keyboard\_"

android:clickable="true"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="48dp"

android:layout\_weight="1">

<ImageView

android:layout\_width="112dp"

android:layout\_height="112dp"

android:layout\_centerInParent="true"

android:src="@mipmap/ic\_launcher" />

</RelativeLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="43dp"

android:layout\_marginTop="48dp">

<ImageView

android:layout\_width="32dp"

android:layout\_height="43dp"

android:layout\_centerVertical="true"

android:layout\_marginStart="38dp"

android:layout\_marginBottom="8dp"

android:src="@drawable/ic\_baseline\_person\_outline\_24"

android:layout\_marginLeft="38dp" />

<EditText

android:id="@+id/et\_username"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_gravity="center"

android:layout\_marginStart="80dp"

android:layout\_marginEnd="38dp"

android:layout\_marginBottom="1dp"

android:hint="邮箱/账号"

android:inputType="text"

android:maxLines="1"

android:layout\_marginLeft="80dp"

android:layout\_marginRight="38dp" />

</RelativeLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="43dp"

android:layout\_marginTop="16dp">

<ImageView

android:layout\_width="32dp"

android:layout\_height="40dp"

android:layout\_alignParentBottom="true"

android:layout\_marginStart="38dp"

android:layout\_marginLeft="38dp"

android:layout\_marginBottom="8dp"

android:src="@drawable/ic\_baseline\_lock\_24" />

<EditText

android:id="@+id/et\_pwd"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_marginStart="80dp"

android:layout\_marginEnd="38dp"

android:layout\_marginBottom="1dp"

android:hint="密码"

android:inputType="textPassword"

android:maxLines="1"

android:layout\_marginLeft="80dp"

android:layout\_marginRight="38dp" />

<ImageView

android:id="@+id/iv\_pwd\_switch"

android:layout\_width="24dp"

android:layout\_height="24dp"

android:layout\_alignParentEnd="true"

android:layout\_alignParentBottom="true"

android:layout\_marginEnd="45dp"

android:layout\_marginBottom="15dp"

android:src="@drawable/ic\_baseline\_visibility\_off\_24"

android:layout\_alignParentRight="true"

android:layout\_marginRight="45dp" />

</RelativeLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="40dp"

android:layout\_marginTop="16dp">

<TextView

android:id="@+id/tv\_sign\_up"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentStart="true"

android:layout\_alignParentBottom="true"

android:layout\_alignParentLeft="true"

android:text="注册"

android:textColor="?attr/colorPrimary"

android:clickable="true"

android:layout\_marginLeft="40dp"

android:layout\_alignParentRight="true"

/>

<CheckBox

android:id="@+id/cb\_remember\_pwd"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_alignParentBottom="true"

android:layout\_alignBaseline="@id/tv\_sign\_up"

android:layout\_marginRight="38dp"

android:layout\_marginBottom="4dp"

android:text="记住密码"

android:layout\_marginEnd="38dp"

android:textColor="?attr/colorPrimary"

/>

</RelativeLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="32dp"

android:layout\_weight="1">

<Button

android:id="@+id/bt\_login"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_marginStart="38dp"

android:layout\_marginTop="32dp"

android:layout\_marginEnd="38dp"

android:background="?attr/colorPrimary"

android:text="登录"

android:textColor="@android:color/white" />

</RelativeLayout>

#### 2.2注册主要逻辑：

##### 数据库插入：

signup.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

\_User user = new \_User();

if(imgUri == null){

Toast.makeText(mContext, "请补全以上信息!",

Toast.LENGTH\_SHORT).show();

return;

}

Log.d(TAG,"目标图片uri为: "+imgUri);

if(takePhotoUrl != null){

imgUrl = takePhotoUrl;

}

else{

imgUrl = getRealPathFromURI(imgUri,mContext);

}

final BmobFile headPicture = new BmobFile( new File(imgUrl) );

String password = Password.getText().toString();

String username = Username.getText().toString();

String nickname = Nickname.getText().toString();

String confirm = Confirm.getText().toString();

if (username.length() != 0 && password.length() != 0 && confirm.length() != 0 &&nickname.length()!=0) {

if (password.equals(confirm)) {

LoadingDialog laoding = new LoadingDialog(mContext,0);

laoding.show();

//

user.setNickName(nickname);

user.setUsername(username);

user.setPassword(password);

user.setHeadPicture(headPicture);

user.setImagePath(imgUrl);

user.setPassword\_record(password);

headPicture.uploadblock(new UploadFileListener() {

@Override

public void done(BmobException e) {

if(e==null){

user.signUp(new SaveListener<\_User>() {

@Override

public void done(\_User user, BmobException e) {

if (e == null) {

laoding.dismiss();

toast("注册成功");

passwordList.put(username,password);

Constants.PasswordList=passwordList;

Intent intent=new Intent();

//防止返回键返回上级页面

intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK|Intent.FLAG\_ACTIVITY\_NEW\_TASK);

intent.setClass(RegisterActivity.this, LoginActivity.class);

startActivity(intent);

} else

{

toast(e.getMessage());

laoding.dismiss();

}

}

});

}else{

laoding.dismiss();

toast("上传文件失败：" + e.getMessage());

}

}

});

}

else {

toast("两次输入密码不一致");

}

} else {

toast("请补全以上信息");

}

}

});

}

##### 图片选择：

//得到图片地址

public static String getRealPathFromURI(Uri contentUri, Context mContext) {

String[] proj = {MediaStore.Images.Media.DATA};

CursorLoader loader = new CursorLoader(mContext, contentUri, proj, null, null, null);

Cursor cursor = loader.loadInBackground();

int column\_index = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);

cursor.moveToFirst();

return cursor.getString(column\_index);

}

//得到图片文件名称

private String getPhotoFileName() {

Date date = new Date(System.currentTimeMillis());

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyyMMdd\_HHmmss");

return "IMG\_" + dateFormat.format(date);

}

//拍照

protected void getImageFromCamera() {

takePhotoUrl = null;

String state = Environment.getExternalStorageState();

//取消严格模式 FileProvider

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.N) {

StrictMode.VmPolicy.Builder builder = new StrictMode.VmPolicy.Builder();

StrictMode.setVmPolicy(builder.build());

}

//保存到相册

if (state.equals(Environment.MEDIA\_MOUNTED)) {

File file = new File(path);

if (!file.exists()) {

file.mkdir();

}

String fileName = getPhotoFileName() + ".jpg";

Intent intent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);

//photoUri = Uri.fromFile(new File(file, fileName));

takePhotoUrl = path + fileName;

Log.d(TAG, "图片保存路径为: " + takePhotoUrl);

photoUri = Uri.fromFile(new File(takePhotoUrl));

Log.d(TAG, "拍照1: ");

intent.putExtra(MediaStore.EXTRA\_OUTPUT, photoUri);

Log.d(TAG, "拍照2");

startActivityForResult(intent, REQUEST\_IMAGE\_CAPTURE);

Log.d(TAG, "拍照3");

}

}

//相册和拍照选择会话框

private void showSelectDialog(){

//创建存放头像的文件夹

//PictureUtil.mkdirMyPetRootDirectory();

mPhotoPopupWindow = new PhotoPopupWindow( this, new View.OnClickListener() {

@Override

public void onClick(View v) {

// 文件权限申请

if (ContextCompat.checkSelfPermission(mContext,

Manifest.permission.WRITE\_EXTERNAL\_STORAGE)

!= PackageManager.PERMISSION\_GRANTED) {

// 权限还没有授予，进行申请

ActivityCompat.requestPermissions((Activity)mContext,

new String[]{Manifest.permission.WRITE\_EXTERNAL\_STORAGE}, 200); // 申请的 requestCode 为 200

} else {

// 如果权限已经申请过，直接进行图片选择

mPhotoPopupWindow.dismiss();

getImageFromAlbum();

}

}

}, new View.OnClickListener()

{

@Override

public void onClick (View v){

// 拍照及文件权限申请

if (ContextCompat.checkSelfPermission(mContext,

Manifest.permission.CAMERA) != PackageManager.PERMISSION\_GRANTED

|| ContextCompat.checkSelfPermission(mContext,

Manifest.permission.WRITE\_EXTERNAL\_STORAGE)

!= PackageManager.PERMISSION\_GRANTED) {

// 权限还没有授予，进行申请

ActivityCompat.requestPermissions((Activity)mContext,

new String[]{Manifest.permission.CAMERA, Manifest.permission.WRITE\_EXTERNAL\_STORAGE}, 300); // 申请的 requestCode 为 300

} else {

// 权限已经申请，直接拍照

mPhotoPopupWindow.dismiss();

getImageFromCamera();

}

}

});

View rootView = LayoutInflater.from(mContext).inflate(R.layout.fragment\_add, null);

mPhotoPopupWindow.showAtLocation(rootView,

Gravity.BOTTOM | Gravity.CENTER\_HORIZONTAL, 0, 0);

}

//相册

protected void getImageFromAlbum() {

Intent intent = new Intent(Intent.ACTION\_PICK);

intent.setType("image/\*");//相片类型

startActivityForResult(intent, REQUEST\_IMAGE\_GET);

}

//调用系统照片裁剪

private void startPhotoZoom(Uri uri) {

Log.d(TAG, "进入修改");

Uri cropUri;

cropUri = uri;

Intent intent = new Intent("com.android.camera.action.CROP");

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.N) {

//添加这一句表示对目标应用临时授权该Uri所代表的文件

intent.addFlags(Intent.FLAG\_GRANT\_READ\_URI\_PERMISSION);

}

intent.setDataAndType(uri, "image/\*");

intent.putExtra("crop", "true");

intent.putExtra("scale", true);

intent.putExtra("return-data", false);

//Log.e(TAG,"cropUri = "+cropUri.toString());

intent.putExtra(MediaStore.EXTRA\_OUTPUT, cropUri);

intent.putExtra("outputFormat", Bitmap.CompressFormat.JPEG.toString());

intent.putExtra("noFaceDetection", true); // no face detection

startActivityForResult(intent, REQUEST\_SMALL\_IMAGE\_CUTTING);

}

//根据上次选择结果选择是否进入裁剪

@Override

public void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

Uri uri = null;

switch (requestCode) {

case REQUEST\_IMAGE\_GET:

uri = data.getData();

Log.d(TAG, "图片选择的Uri为: " + uri);

imgUri = uri;

if (uri != null) {

Log.d(TAG, "图片Uri为: " + uri);

Bitmap img = null;

try {

img = MediaStore.Images.Media.getBitmap(mContext.getContentResolver(), uri);

} catch (IOException e) {

e.printStackTrace();

}

Log.d(TAG, "图片大小为: " + img.getWidth() + img.getHeight());

imageView.setImageBitmap(img);

} else {

Log.d(TAG, "找不到Uti");

}

//startPhotoZoom(uri);

//mImage.setImageURI(uri);

break;

case REQUEST\_IMAGE\_CAPTURE:

if (data != null && data.getData() != null) {

uri = data.getData();

Log.d(TAG, "照相的Uri为: " + uri);

//mImage.setImageURI(photoUri);

}

if (uri == null) {

if (photoUri != null) {

uri = photoUri;

Log.d(TAG, "uri为null,照相图片的Uri为: " + uri);

startPhotoZoom(uri);

}

}

break;

case REQUEST\_SMALL\_IMAGE\_CUTTING:

uri = data.getData();

imgUri = uri;

if (uri != null) {

Log.d(TAG, "图片Uri为: " + uri);

Bitmap img = null;

try {

img = MediaStore.Images.Media.getBitmap(mContext.getContentResolver(), uri);

} catch (IOException e) {

e.printStackTrace();

}

Log.d(TAG, "图片大小为: " + img.getWidth() + img.getHeight());

imageView.setImageBitmap(img);

} else {

Log.d(TAG, "找不到Uti");

}

break;

default:

break;

}

}

##### 图片上传

private void uploadCard(){

//找不到图片Uri

if(imgUri == null){

Toast.makeText(mContext, "图片未上传，请重新上传图片!",

Toast.LENGTH\_SHORT).show();

return;

}

Log.d(TAG,"目标图片uri为: "+imgUri);

Card card = new Card();

if(takePhotoUrl != null){

imgUrl = takePhotoUrl;

}

else{

imgUrl = getRealPathFromURI(imgUri,mContext);

}

BmobFile Picture = new BmobFile( new File(imgUrl) );

String username = null;

String nickName = null;

BmobFile headPicture = null;

//打印信息

Log.d(TAG,"图片Uri为: " + imgUri);

Log.d(TAG,"图片地址为: " + imgUrl);

Log.d(TAG,"图片文件为: " + Picture);

//设置加载样式

LoadingDialog laoding = new LoadingDialog(mContext,0);

laoding.setMessage("上传头像中").show();

//将图片上传到服务器

//用uploadblock将图片上传至服务器

Picture.uploadblock(new UploadFileListener() {

@Override

public void done(BmobException e) {

if(e==null){

//上传至Bmob

card.save(new SaveListener<String>() {

@Override

public void done(String s, BmobException e) {

if(e == null){

laoding.dismiss();

Toast.makeText(mContext, "图片上传成功!!!",

Toast.LENGTH\_SHORT).show();

}else{

Log.d(TAG,"失败信息为: " + e.getMessage());

Toast.makeText(mContext, "图片上传失败!!!",

Toast.LENGTH\_SHORT).show();

}

}

});

Log.d(TAG,"上传文件成功:" + Picture.getFileUrl());

}else{

Log.d(TAG,"上传文件失败：" + e.getMessage());

laoding.dismiss();

Toast.makeText(mContext, "图片上传失败!!!",

Toast.LENGTH\_SHORT).show();

}

}

@Override

public void onProgress(Integer value) {

// 返回的上传进度（百分比）

}

});

}

#### 3.1主页逻辑

//重写底部导航栏的点击事件

@Override

public void onClick(View v) {

switch (v.getId()){

case R.id.bottom\_bar\_1\_btn:

//切换Fragment

getSupportFragmentManager().beginTransaction().replace(R.id.main\_body,mHomeFragment).commit();

setSelectStatus(0);

break;

case R.id.bottom\_bar\_2\_btn:

//切换Fragment

getSupportFragmentManager().beginTransaction().replace(R.id.main\_body,mAddFragment).commit();

setSelectStatus(1);

break;

case R.id.bottom\_bar\_3\_btn:

//切换Fragment

getSupportFragmentManager().beginTransaction().replace(R.id.main\_body,mMyFragment).commit();

setSelectStatus(2);

break;

}

}

//用于打开初始页面

private void setMain() {

//默认进入首页

setSelectStatus(0);

//getSupportFragmentManager() -> beginTransaction() -> add -> (R.id.main\_boy,显示课程 new CourseFragment()

this.getSupportFragmentManager().beginTransaction().add(R.id.main\_body,mHomeFragment).commit();

}

#### 4.1首页界面

##### 界面

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/lv\_card\_list"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:scrollbars="vertical"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

android:dividerHeight="8dp"

/>

##### 子项

<RelativeLayout

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@drawable/card\_shape"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

>

<!--图片-->

<com.makeramen.roundedimageview.RoundedImageView

android:id="@+id/iv\_image"

android:layout\_width="200dp"

android:layout\_height="300dp"

android:layout\_marginBottom="3dp"

android:scaleType="centerCrop"

android:layout\_gravity="center"

android:src="@drawable/loadingfail"

android:layout\_centerHorizontal="true"

app:riv\_corner\_radius="5dp"

/>

<!--内容-->

<TextView

android:id="@+id/tv\_content"

style="@style/TextAppearance.AppCompat.Subhead"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="8dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="3dp"

android:textSize="20sp"

android:layout\_below="@id/iv\_image"

android:layout\_alignLeft="@id/iv\_image"

android:text="#测试内容#"

/>

<!--头像-->

<com.makeramen.roundedimageview.RoundedImageView

android:id="@+id/iv\_head"

android:src="@drawable/head"

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_below="@id/tv\_content"

android:layout\_alignLeft="@id/iv\_image"

android:layout\_marginLeft="8dp"

android:layout\_marginTop="5dp"

android:layout\_marginBottom="8dp"

app:riv\_oval="true"

/>

<!--昵称-->

<TextView

android:id="@+id/tv\_name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="默认昵称"

android:textSize="15dp"

android:textColor="@color/textColor"

style="@style/TextAppearance.AppCompat.Subhead"

android:layout\_alignBottom="@id/iv\_head"

android:layout\_toRightOf="@id/iv\_head"

android:layout\_marginLeft="5dp"

android:layout\_marginTop="5dp"

/>

<!--收藏数-->

<TextView

android:id="@+id/tv\_likeNumber"

android:text="100"

style="@style/TextAppearance.AppCompat.Subhead"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="16sp"

android:layout\_marginTop="35dp"

android:layout\_toRightOf="@id/tv\_name"

android:layout\_marginLeft="5dp"

android:layout\_marginBottom="8dp"

android:layout\_below="@id/iv\_image"

/>

<!--收藏图标-->

<ImageView

android:id="@+id/iv\_love"

android:src="@drawable/love"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:clickable="true"

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_marginTop="35dp"

android:layout\_toRightOf="@id/tv\_likeNumber"

android:layout\_marginLeft="5dp"

android:layout\_below="@id/iv\_image" />

<!--分享图标-->

<ImageView

android:id="@+id/iv\_share"

android:src="@drawable/share"

android:scaleType="centerCrop"

android:clickable="true"

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_below="@id/iv\_image"

android:layout\_toRightOf="@id/iv\_love"

android:layout\_marginLeft="5dp"

android:layout\_marginTop="35dp"

android:layout\_marginRight="10dp"

android:layout\_marginBottom="10dp"

/>

</RelativeLayout>

#### 4.2首页逻辑：

##### 适配器：

public class CardAdapter extends RecyclerView.Adapter<CardAdapter.ViewHolder>{

Bitmap img;

private List<Card> mCardList;

private Context mContext;

private String imgName;

//大图会话

private Dialog dialog;

//显示大图

private ImageView mImageView;

//当前用户的点赞列表

private Map<String, Boolean> likeList = new HashMap<>();

private \_User currentUser;

private int likeNumber = 0;

private Map<String, Boolean> shareList = new HashMap<>();

private ViewHolder targetHolder;

//分享信息

public static String TITLE = "图片分享";

//handler 用于线程间的通信

private Handler myHandler = new Handler() {

@Override

public void handleMessage(Message msg) {

switch (msg.what) {

//当点击图片时，通过url加载图片，并显示大图

case 0:

//测试

Log.d("Adapter","img的内容为: "+imgName);

Log.d("Adapter","img的图片为: "+img);

mImageView = getImageView(img);

//初始化会话

initDialog();

//显示会话

dialog.show();

break;

//分享图片

case 1:

Uri uri = Uri.parse(MediaStore.Images.Media.insertImage(mContext.getContentResolver(), img, null,null));;

shareImage(uri);

//激活分享图标

targetHolder.ShareIcon.setImageResource(R.drawable.share\_a);

break;

default:

break;

}

}

};

static class ViewHolder extends RecyclerView.ViewHolder{

RoundedImageView CardImage;

TextView CardContent;

RoundedImageView HeadPic;

TextView NickName;

TextView LikeNumber;

ImageView LikeIcon;

ImageView ShareIcon;

public ViewHolder(View view){

super(view);

CardImage = view.findViewById(R.id.iv\_image);

CardContent = view.findViewById(R.id.tv\_content);

HeadPic = view.findViewById(R.id.iv\_head);

NickName = view.findViewById(R.id.tv\_name);

LikeNumber = view.findViewById(R.id.tv\_likeNumber);

LikeIcon = view.findViewById(R.id.iv\_love);

ShareIcon = view.findViewById(R.id.iv\_share);

}

}

public CardAdapter(Context context,List<Card> CardList){

Log.d("Adapter","适配器创建成功");

//super();

mContext =context;

mCardList = CardList;

}

@NonNull

@Override

public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

Log.d("Adapter","适配器视图创建成功");

View view = LayoutInflater.from(parent.getContext()).

inflate(R.layout.card\_item,parent,false);

ViewHolder holder = new ViewHolder(view);

//得到当前用户

currentUser = BmobUser.getCurrentUser(\_User.class);

//图片监听事件

holder.CardImage.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

int position = holder.getAdapterPosition();

Card card = mCardList.get(position);

String imgUrl = null;

if(card.getPicture()!=null){

imgUrl = card.getPicture().getUrl();

}

//获取img名称

imgName = card.getDescription();

//通过url,加载图片

if(imgUrl!=null){

initNetWorkImage(imgUrl,mContext,0);

}

//图片加载失败

else{

BitmapDrawable bd = (BitmapDrawable)mContext.getResources().getDrawable(R.drawable.loadingfail);

img = bd.getBitmap();

mImageView = getImageView(img);

//初始化会话

initDialog();

//显示会话

dialog.show();

}

Log.d("Adapter","图片地址为: "+ imgUrl);

}

});

//点赞点击事件

holder.LikeIcon.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

int position = holder.getAdapterPosition();

Card card = mCardList.get(position);

String content = holder.CardContent.getText().toString();

int likes = Integer.parseInt(holder.LikeNumber.getText().toString());

boolean flag = likeList.get(content);

if(flag){

//已点赞状态 --> 未点赞状态

flag = !flag;

likeList.put(content,flag);

//设置图标

holder.LikeIcon.setImageResource(R.drawable.love);

//设置点赞数量

likes -= 1;

String like\_Number = String.valueOf(likes);

holder.LikeNumber.setText(like\_Number);

//移除用户信息

removeUserToLikes(card);

}else{

//未点赞状态 --> 已点赞状态

flag = !flag;

likeList.put(content,flag);

//设置图标

holder.LikeIcon.setImageResource(R.drawable.love\_a);

//设置点赞数量

likes += 1;

String like\_Number = String.valueOf(likes);

holder.LikeNumber.setText(like\_Number);

//添加用户信息

addUserToLikes(card);

}

}

});

//分享点击事件

holder.ShareIcon.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

int position = holder.getAdapterPosition();

Card card = mCardList.get(position);

String content = holder.CardContent.getText().toString();

shareList.put(content,true);

String url = card.getPicture().getUrl();

addUserToShares(card);

//分享图片

initNetWorkImage(url,mContext,1);

//设置图标和标志

targetHolder = holder;

//holder.ShareIcon.setImageResource(R.drawable.share\_a);

//isShare.put(content,true);

}

});

return holder;

}

@Override

public void onBindViewHolder(@NonNull ViewHolder holder, int position) {

Card card = mCardList.get(position);

String imgUrl = null;

String headUrl = null;

String content = "#"+card.getDescription()+"#";

if( card.getPicture() != null){

imgUrl = card.getPicture().getUrl();

}

if( card.getHeadPicture() != null){

headUrl = card.getHeadPicture().getUrl();

}

// //设置点赞列表

// Constants.LIKELIST = likeList;

//查询该卡片的喜欢用户列表

queryLikesUser(holder,card);

queryShareUser(holder,card);

Log.d("Adapter","图片地址为:"+imgUrl);

Log.d("Adapter","头像地址为:"+headUrl);

Log.d("Adapter","用户昵称为:"+card.getNickName());

//图片

if(imgUrl!=null){

Glide.with(mContext).load(imgUrl)

.into(holder.CardImage);

}

//内容

if(card.getDescription()!=null){

String description = "#"+card.getDescription()+"#";

holder.CardContent.setText(description);

}

//头像

if(headUrl!=null){

Glide.with(mContext).load(headUrl)

.into(holder.HeadPic);

}

//昵称

if(card.getNickName()!= ""){

holder.NickName.setText(card.getNickName());

}

}

@Override

public int getItemCount() {

return mCardList.size();

}

public void addData(Card card){

mCardList.add(card);

//notifyDataSetChanged();

}

public void setData(List<Card> cardData){

mCardList.addAll(cardData);

//notifyDataSetChanged();

}

//查询点赞用户中是否包含已登录用户

private void queryLikesUser(ViewHolder holder,Card card){

// 查询喜欢这个帖子的所有用户，因此查询的是用户表

BmobQuery<\_User> query = new BmobQuery<\_User>();

//likes是Card表中的字段，用来存储所有喜欢该帖子的用户

query.addWhereRelatedTo("likes", new BmobPointer(card));

query.findObjects(new FindListener<\_User>() {

@Override

public void done(List<\_User> object, BmobException e) {

if(e==null){

likeNumber = object.size();

String content = holder.CardContent.getText().toString();

Log.i("bmob","查询对象为："+content);

Log.i("bmob","查询个数："+object.size());

//设置点赞数

String like\_Number = String.valueOf(likeNumber);;

holder.LikeNumber.setText(like\_Number);

//查询是否包含当前用户

boolean isContain = false;

for(\_User user : object){

String userName = user.getUsername();

String currentUserName = currentUser.getUsername();

Log.d("Adapter","用户名列表为: "+userName);

Log.d("Adapter","当前用户名列表为: "+currentUserName);

if(userName.equals(currentUserName)){

isContain = true;

}

}

Log.d("Adapter","点赞列表为: "+object);

Log.d("Adapter","当前用户为: "+currentUser);

Log.i("bmob", "查询结果为："+isContain);

likeList.put(content,isContain);

Constants.LIKELIST = likeList;

//激活为点赞状态

if(isContain){

holder.LikeIcon.setImageResource(R.drawable.love\_a);

}

//myHandler.sendEmptyMessage(1);

}else{

Log.i("bmob","失败："+e.getMessage());

}

}

});

}

private void queryShareUser(ViewHolder holder,Card card){

// 查询喜欢这个帖子的所有用户，因此查询的是用户表

BmobQuery<\_User> query = new BmobQuery<\_User>();

//likes是Card表中的字段，用来存储所有喜欢该帖子的用户

query.addWhereRelatedTo("shares", new BmobPointer(card));

query.findObjects(new FindListener<\_User>() {

@Override

public void done(List<\_User> object, BmobException e) {

if(e==null){

String content = holder.CardContent.getText().toString();

//查询是否包含当前用户

boolean isContain = false;

for(\_User user : object){

String userName = user.getUsername();

String currentUserName = currentUser.getUsername();

Log.d("Adapter","用户名列表为: "+userName);

Log.d("Adapter","当前用户名列表为: "+currentUserName);

if(userName.equals(currentUserName)){

isContain = true;

}

}

shareList.put(content,isContain);

Constants.SHARELIST = shareList;

}else{

Log.i("bmob","失败："+e.getMessage());

}

}

});

}

//添加当前用户信息到 收藏 表中

private void addUserToLikes(Card card){

BmobRelation relation = new BmobRelation();

//将当前用户添加到多对多关联中

relation.add(currentUser);

//多对多关联指向`Card`的`likes`字段

card.setLikes(relation);

card.update(new UpdateListener() {

@Override

public void done(BmobException e) {

if(e==null){

Log.i("bmob","用户B和该帖子关联成功");

}else{

Log.i("bmob","失败："+e.getMessage());

}

}

});

}

//移除当前用户信息到 收藏 表中

private void removeUserToLikes(Card card){

BmobRelation relation = new BmobRelation();

relation.remove(currentUser);

card.setLikes(relation);

card.update(new UpdateListener() {

@Override

public void done(BmobException e) {

if(e==null){

Log.i("bmob","关联关系删除成功");

}else{

Log.i("bmob","失败："+e.getMessage());

}

}

});

}

private void addUserToShares(Card card){

BmobRelation relation = new BmobRelation();

//将当前用户添加到多对多关联中

relation.add(currentUser);

//多对多关联指向`Card`的`likes`字段

card.setShares(relation);

card.update(new UpdateListener() {

@Override

public void done(BmobException e) {

if(e==null){

Log.i("bmob","用户B和该帖子关联成功");

}else{

Log.i("bmob","失败："+e.getMessage());

}

}

});

}

//分享图片

private void shareImage(Uri uri){

Intent intent = new Intent(Intent.ACTION\_SEND);

intent.setType("image/png");

intent.putExtra(Intent.EXTRA\_STREAM, uri);

//intent.putExtra(Intent.EXTRA\_STREAM, bitmap);

mContext.startActivity(Intent.createChooser(intent,TITLE));

}

private void initDialog() {

Log.d("Adapter","点赞列表为: "+likeList);

//大图所依附的dialog

dialog = new Dialog(mContext, R.style.Theme\_AppCompat);

dialog.setContentView(mImageView);

//大图的点击事件（点击让他消失）

mImageView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

dialog.dismiss();

}

});

//大图的长按监听

mImageView.setOnLongClickListener(new View.OnLongClickListener() {

@Override

public boolean onLongClick(View v) {

//弹出的“保存图片”的Dialog

AlertDialog.Builder builder = new AlertDialog.Builder(mContext);

builder.setItems(new String[]{mContext.getResources().getString(R.string.save\_picture)}, new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

//saveCroppedImage(((BitmapDrawable) mImageView.getDrawable()).getBitmap());

Bitmap img = ((BitmapDrawable) mImageView.getDrawable()).getBitmap();

//保存文件

GalleryFileSaver.saveBitmapToGallery(mContext,imgName,img);

}

});

builder.show();

return true;

}

});

}

//动态的ImageView

private ImageView getImageView(Bitmap img){

ImageView iv = new ImageView(mContext);

//宽高

iv.setLayoutParams(new RelativeLayout.LayoutParams(ViewGroup.LayoutParams.MATCH\_PARENT, ViewGroup.LayoutParams.WRAP\_CONTENT));

//设置Padding

iv.setPadding(20,20,20,20);

//imageView设置图片

iv.setImageBitmap(img);

return iv;

}

/\*\*

\* 自己写的加载网络图片的方法

\* img\_url 图片的网址

\*/

public void initNetWorkImage(final String imgUrl, final Context context,int flag) {

new AsyncTask<Void, Void, Bitmap>() {

@Override

protected Bitmap doInBackground(Void... params) {

Bitmap bitmap = null;

try {

bitmap = Glide.with(context)

.asBitmap()

.load(imgUrl)

.submit(360, 480).get();

} catch (Exception e) {

e.printStackTrace();

}

return bitmap;

}

@Override

protected void onPostExecute(Bitmap bitmap) {

img = bitmap;

myHandler.sendEmptyMessage(flag);

}

}.execute();

}

##### 主逻辑

//初始化对象并保存数据

private void initData() {

CardData = new ArrayList<>();

// StaggeredGridLayoutManager layoutManager =

// new StaggeredGridLayoutManager(2,StaggeredGridLayoutManager.VERTICAL);

GridLayoutManager layoutManager = new GridLayoutManager(this.context,2);

lvCardList.setLayoutManager(layoutManager);

adapter = new CardAdapter(context, CardData);

lvCardList.setAdapter(adapter);

//获取首页卡片数据列表

getCardData();

}

//从Bmob后台数据库，获取首页卡片数据

private void getCardData() {

BmobQuery<Card> query = new BmobQuery<>();

//按照时间降序

query.order("-createdAt");

query.findObjects(new FindListener<Card>() {

@Override

public void done(List<Card> cardData, BmobException e) {

if (e == null) {

adapter.setData(cardData);

adapter.notifyDataSetChanged();

//Snackbar.make(rootView, "查询成功：" + cardData.size(), Snackbar.LENGTH\_LONG).show();

} else {

Log.e("BMOB", e.toString());

//Snackbar.make(rootView, e.getMessage(), Snackbar.LENGTH\_LONG).show();

}

}

});

}

#### 5发布逻辑

添加类同注册逻辑。不再赘述。

#### 6.我的界面

##### 6.1页面布局

<RelativeLayout

android:id="@+id/re\_myinfo"

android:layout\_width="match\_parent"

android:layout\_height="80dp"

android:layout\_marginTop="10dp"

android:layout\_marginBottom="10dp"

android:background="#ffff"

android:clickable="true"

android:paddingTop="8dp"

android:paddingBottom="8dp">

<ImageView

android:id="@+id/picture"

android:layout\_width="70dp"

android:layout\_height="60dp"

android:layout\_marginLeft="15dp"

android:layout\_marginTop="8dp"

android:src="@drawable/touxiang" />

<LinearLayout

android:id="@+id/ll\_name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="16dp"

android:layout\_marginTop="12dp"

android:layout\_toRightOf="@id/picture"

android:orientation="horizontal">

<TextView

android:id="@+id/nickname\_"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="昵称："

android:textColor="#353535"

android:textSize="14sp" />

<TextView

android:id="@+id/nick\_name"

android:layout\_width="100dp"

android:layout\_height="wrap\_content"

android:textColor="#353535"

android:textSize="14sp" />

</LinearLayout>

<TextView

android:id="@+id/username\_"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_marginLeft="16dp"

android:layout\_marginBottom="15dp"

android:layout\_toRightOf="@id/picture"

android:text="账号:"

android:textColor="#AAAAAA"

android:textSize="14sp" />

<TextView

android:id="@+id/user\_name"

android:layout\_width="100dp"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_marginLeft="50dp"

android:layout\_marginBottom="15dp"

android:layout\_toRightOf="@id/picture"

android:textColor="#AAAAAA"

android:textSize="14sp"/>

</RelativeLayout>

<View

android:layout\_width="match\_parent"

android:layout\_height="0.7dp"

android:layout\_alignParentBottom="true"

android:layout\_marginRight="12dp"

android:background="#dadada" />

<RelativeLayout

android:id="@+id/hobby"

android:layout\_width="match\_parent"

android:layout\_height="140dp"

android:layout\_marginTop="20dp"

android:layout\_marginBottom="18dp"

android:paddingTop="8dp"

android:paddingBottom="8dp">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="14dp"

android:text="我的"

android:textColor="#353535"

android:textSize="16sp" />

<ImageView

android:id="@+id/like"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_alignParentLeft="true"

android:layout\_centerVertical="true"

android:layout\_marginLeft="40dp"

android:clickable="true"

android:src="@drawable/like"

android:visibility="visible" />

<ImageView

android:id="@+id/publish"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_alignLeft="@id/like"

android:layout\_alignRight="@id/share"

android:layout\_centerVertical="true"

android:layout\_marginLeft="130dp"

android:layout\_marginRight="130dp"

android:src="@drawable/fabu" />

<ImageView

android:id="@+id/share"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_alignParentRight="true"

android:layout\_centerVertical="true"

android:layout\_marginRight="40dp"

android:src="@drawable/zhuanfa" />

<TextView

android:id="@+id/text1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below = "@id/like"

android:layout\_alignLeft = "@id/like"

android:layout\_marginTop="10dp"

android:layout\_marginLeft="1dp"

android:text="我喜欢的"

android:textColor="@color/colorInfo"

android:textSize="12sp" />

<TextView

android:id="@+id/text2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below = "@id/publish"

android:layout\_marginTop="10dp"

android:layout\_centerVertical="true"

android:layout\_alignLeft="@id/text1"

android:layout\_marginLeft="130dp"

android:layout\_alignRight="@id/text3"

android:layout\_marginRight="100dp"

android:text="已发布"

android:textColor="@color/colorInfo"

android:textSize="12sp" />

<TextView

android:id="@+id/text3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below = "@id/share"

android:layout\_alignLeft = "@id/share"

android:layout\_marginTop="10dp"

android:layout\_marginLeft="2dp"

android:text="我转发的"

android:textColor="@color/colorInfo"

android:textSize="12sp" />

</RelativeLayout>

<View

android:layout\_width="match\_parent"

android:layout\_height="0.7dp"

android:layout\_alignParentBottom="true"

android:layout\_marginRight="12dp"

android:background="#dadada" />

<RelativeLayout

android:id="@+id/safe"

android:layout\_width="match\_parent"

android:layout\_marginTop="40dp"

android:layout\_height="48dp"

android:clickable="true"

android:paddingLeft="12dp"

>

<ImageView

android:id="@+id/safe1"

android:layout\_width="26dp"

android:layout\_height="26dp"

android:layout\_alignParentLeft="true"

android:layout\_centerVertical="true"

android:layout\_marginLeft="12dp"

android:src="@drawable/ic\_baseline\_lock\_24" />

<TextView

android:id="@+id/iv\_safe"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_marginLeft="20dp"

android:layout\_toRightOf="@id/safe1"

android:clickable="true"

android:text="修改密码"

android:textColor="#353535"

android:textSize="16sp" />

</RelativeLayout>

<RelativeLayout

android:id="@+id/about\_us"

android:layout\_width="match\_parent"

android:layout\_height="48dp"

android:clickable="true"

android:paddingLeft="12dp">

<ImageView

android:id="@+id/iv\_shoucang"

android:layout\_width="24dp"

android:layout\_height="24dp"

android:layout\_alignParentLeft="true"

android:layout\_centerVertical="true"

android:layout\_marginLeft="12dp"

android:src="@drawable/ic\_baseline\_person\_outline\_24" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_marginLeft="20dp"

android:layout\_toRightOf="@id/iv\_shoucang"

android:text="开发信息"

android:textColor="#353535"

android:textSize="16sp" />

</RelativeLayout>

<RelativeLayout

android:id="@+id/help"

android:layout\_width="match\_parent"

android:layout\_height="48dp"

android:layout\_marginTop="0dp"

android:clickable="true"

android:paddingLeft="12dp">

<ImageView

android:id="@+id/iv\_money\_bag"

android:layout\_width="24dp"

android:layout\_height="24dp"

android:layout\_alignParentLeft="true"

android:layout\_centerVertical="true"

android:layout\_marginLeft="12dp"

android:src="@drawable/help" />

<TextView

android:id="@+id/help1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_marginLeft="20dp"

android:layout\_toRightOf="@id/iv\_money\_bag"

android:text="帮助与反馈"

android:textColor="#353535"

android:textSize="16sp" />

<View

android:layout\_width="match\_parent"

android:layout\_height="0.5dp"

android:layout\_alignParentBottom="true"

android:layout\_marginRight="12dp"

android:background="#dadada" />

</RelativeLayout>

<RelativeLayout

android:id="@+id/settings"

android:layout\_width="match\_parent"

android:layout\_height="48dp"

android:layout\_marginTop="0dp"

android:clickable="true"

android:paddingLeft="12dp">

<ImageView

android:id="@+id/iv\_setting"

android:layout\_width="24dp"

android:layout\_height="24dp"

android:layout\_alignParentLeft="true"

android:layout\_centerVertical="true"

android:layout\_marginLeft="12dp"

android:src="@drawable/shezhi" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_marginLeft="20dp"

android:layout\_toRightOf="@id/iv\_setting"

android:text="账号设置"

android:textColor="#353535"

android:textSize="16sp" />

</RelativeLayout>

##### 6.2页面逻辑：

更换头像与注册页面的头像上传逻辑相似。不再赘述。我喜欢，我转发，我发布瀑布流类似首页逻辑类似不再赘述。更换密码类似注册页面，不再赘述。退出仅为跳转登录页面不再赘述。其余按钮功能简单文本框不再赘述。

#### 7.1选择框界面：

<LinearLayout

android:id="@+id/ll\_pop"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="15dp"

android:layout\_marginRight="15dp"

android:orientation="vertical"

android:layout\_alignParentBottom="true">

<Button

android:id="@+id/icon\_btn\_select"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@drawable/white\_btn\_top"

android:textColor="@color/colorPrimary"

android:text="从本地选择图片"/>

<View

android:layout\_width="match\_parent"

android:layout\_height="1dp"

/>

<Button

android:id="@+id/icon\_btn\_camera"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@drawable/white\_btn\_bottom"

android:textColor="@color/colorPrimary"

android:text="拍照"/>

<Button

android:id="@+id/icon\_btn\_cancel"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:layout\_marginBottom="15dp"

android:background="@drawable/white\_btn"

android:textColor="@color/colorPrimary"

android:text="取消"/>

</LinearLayout>

#### 7.2选择框逻辑：

\*\*

\* 设置布局以及点击事件

\*/

public void Init() {

LayoutInflater inflater = (LayoutInflater) mContext

.getSystemService(Context.LAYOUT\_INFLATER\_SERVICE);

assert inflater != null;

mView = inflater.inflate(R.layout.pop\_item, null);

Button btn\_camera = (Button) mView.findViewById(R.id.icon\_btn\_camera);

Button btn\_select = (Button) mView.findViewById(R.id.icon\_btn\_select);

Button btn\_cancel = (Button) mView.findViewById(R.id.icon\_btn\_cancel);

btn\_select.setOnClickListener(mSelectListener);

btn\_camera.setOnClickListener(mCaptureListener);

btn\_cancel.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

dismiss();

}

});

// 导入布局

this.setContentView(mView);

// 设置动画效果

this.setAnimationStyle(R.style.Animation\_Design\_BottomSheetDialog);

this.setWidth(WindowManager.LayoutParams.MATCH\_PARENT);

this.setHeight(WindowManager.LayoutParams.WRAP\_CONTENT);

// 设置可触

this.setFocusable(true);

ColorDrawable dw = new ColorDrawable(0x0000000);

this.setBackgroundDrawable(dw);

// 单击弹出窗以外处 关闭弹出窗

mView.setOnTouchListener(new View.OnTouchListener() {

@Override

public boolean onTouch(View v, MotionEvent event) {

int height = mView.findViewById(R.id.ll\_pop).getTop();

int y = (int) event.getY();

if (event.getAction() == MotionEvent.ACTION\_UP) {

if (y < height) {

dismiss();

}

}

return true;

}

});

}