2018-2019 学年第二学期

移动应用开发技术 II
项目报告

题　　目：黑科技小卖部
班　　级：软件工程 16-4 班
姓　　名：　艾程
学　　号：2016024427（08）
授课教师：　王海玲

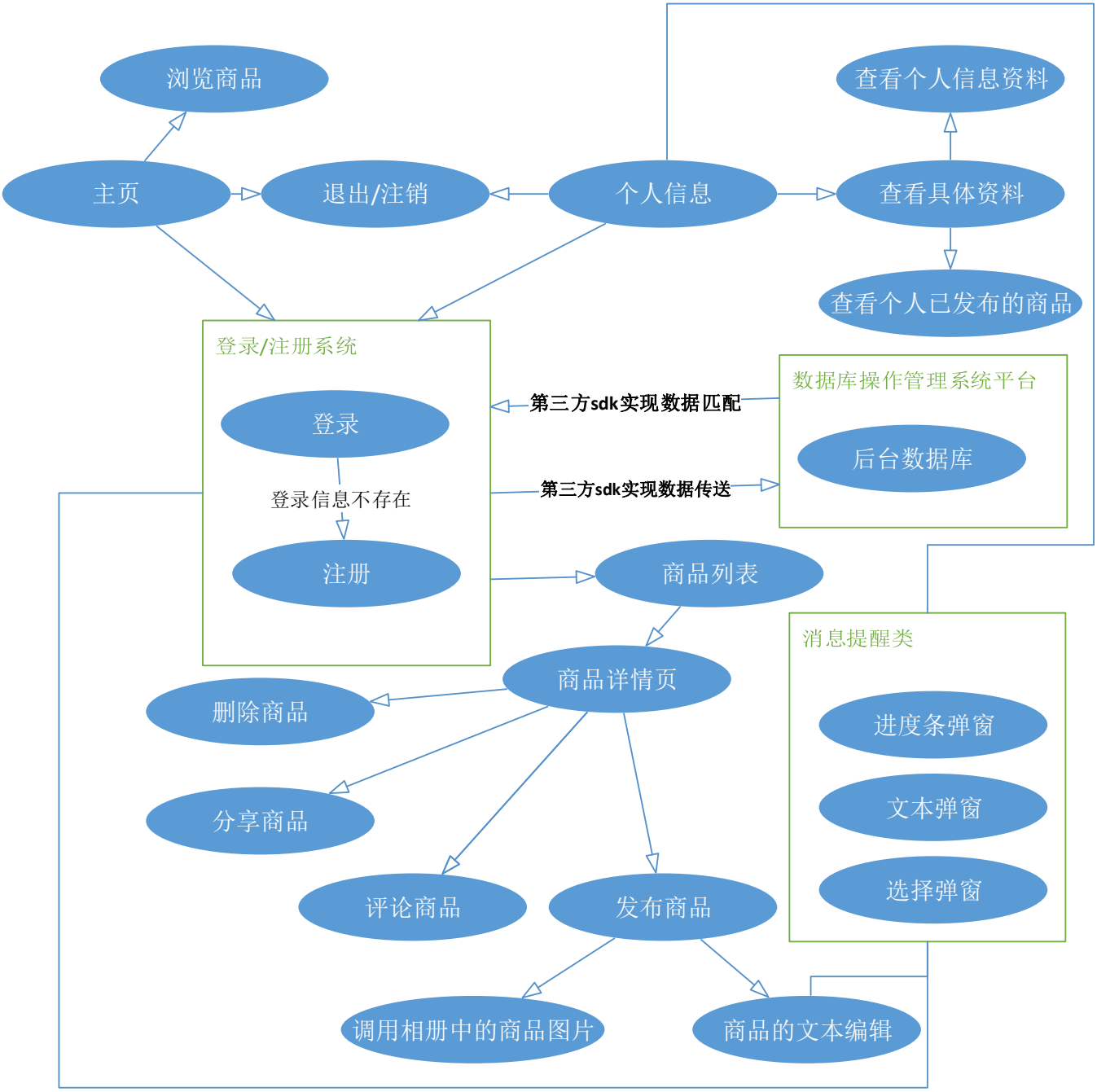
成绩：_____

一、项目简介

每年学校大四生毕业时或者平时有人想卖出闲置物品时，我们公寓群总会出现各种买卖闲置物品的同学。尽管已经有现成的线上渠道（例：闲鱼 app）可以贩卖我们的闲置物品，但是那些平台往往存在诸如：发布商品步骤繁琐、线上交易不可靠等问题。

因此我突发奇想，做一个 app 连接校内对某商品有供应、需求两个需求的小伙伴，直接实现线下交易，更方便快捷，买家少花钱网购，节省了时间。卖家多挣钱卖出闲置物品。更何况，校内交易距离也不远。

二、功能及技术描述



四、程序测试

一、注册/登录界面

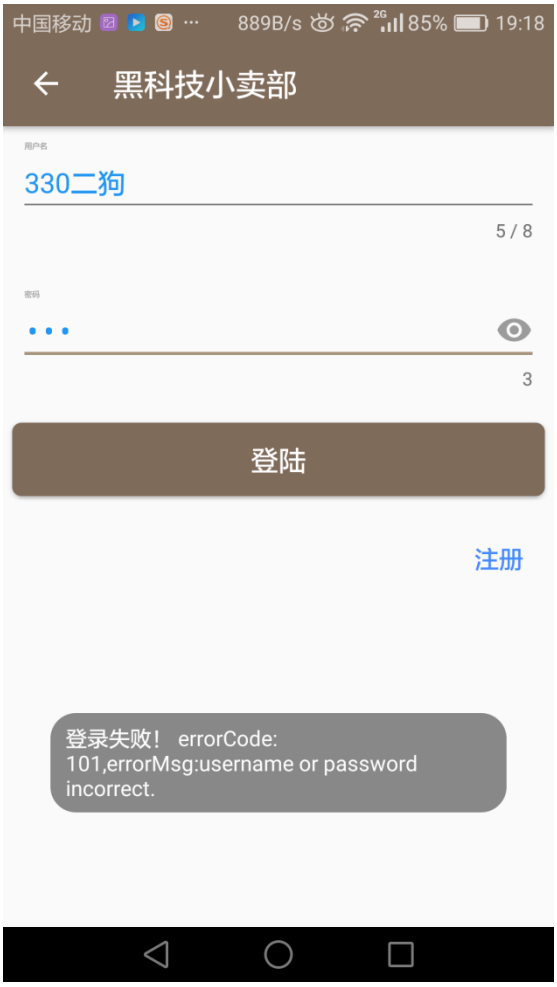
（一）注册操作

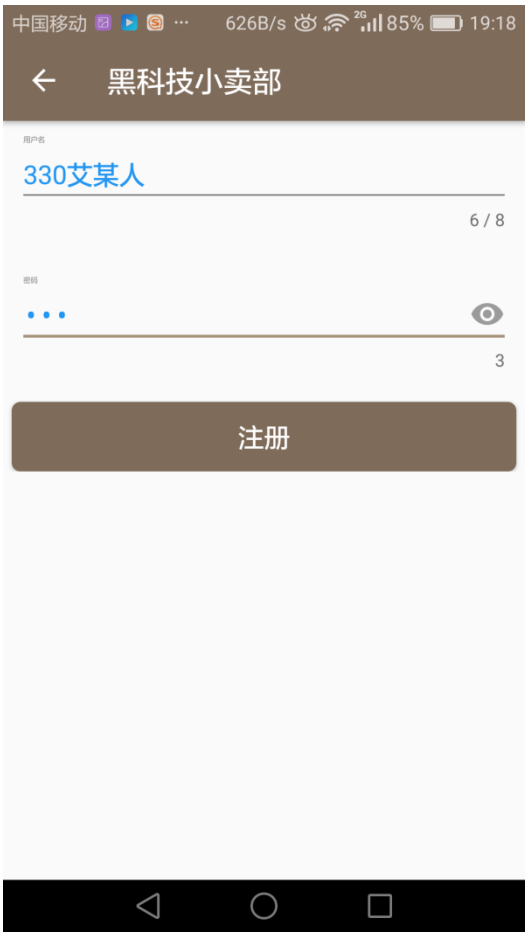
如下图所示，后台数据库可以看到——如果未注册的用户进行了登录操作，会直接弹窗提示用户。注册成功则数据库增加一行新数据。

添加行 | 删除0行 | 添加列 | 查询 | 唯一键 | 列注释 | 更多

objectId String	username String	password Strin
a9908a7238	404艾程	***
9fb40413ec	acc	***
7f4bcd0529	ac	***

（此时数据表中没有用户“330 二狗”）





添加行 | 删除0行 | 添加列 | 查询 | 唯一键 | 列注释 | 更多

objectId String	username String	password Strin
9daf306bff	330艾某人	***
a9908a7238	404艾程	***
9fb40413ec	acc	***
7f4bcd0529	ac	***

（二）登录操作

登陆成功会自动跳转到主页面并且弹窗提示“登陆成功”。

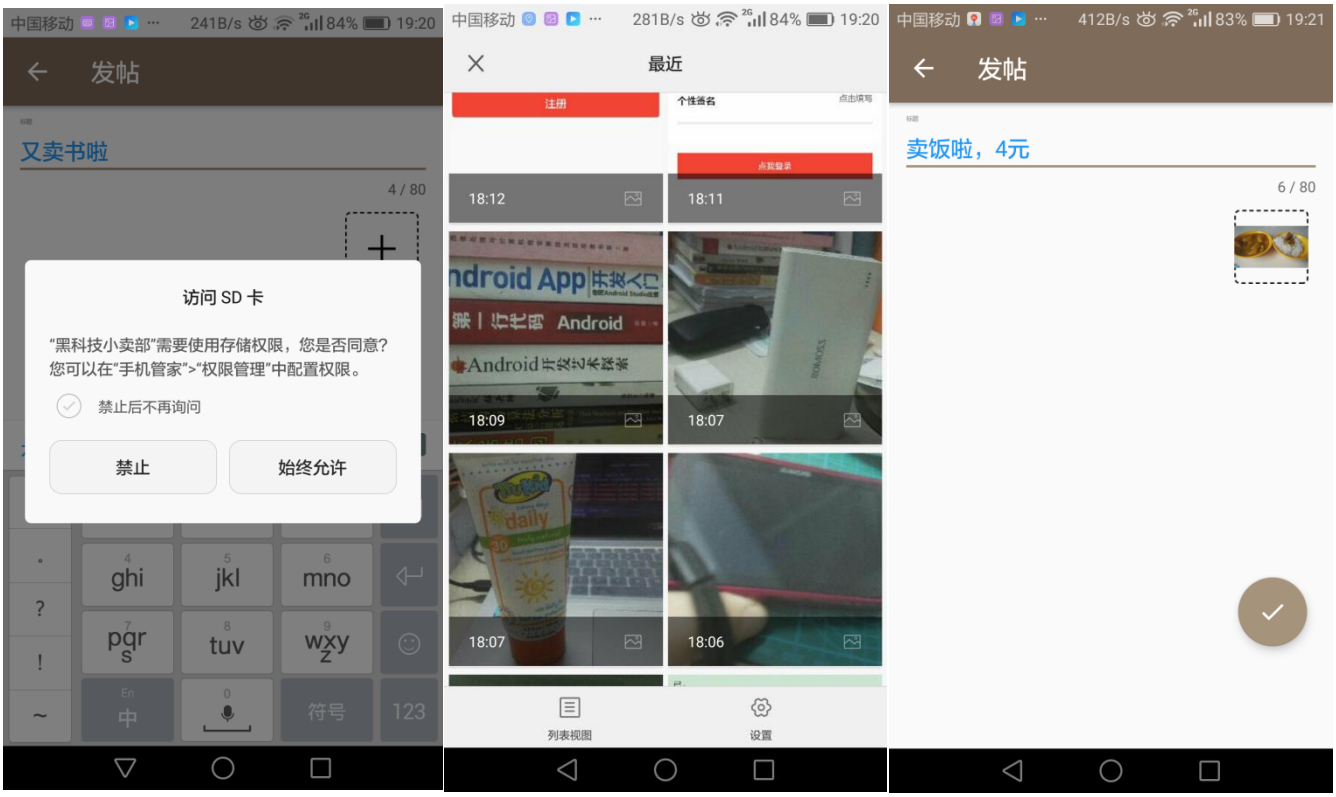
二、主页界面

app 的首界面，用于浏览商品信息。

在游客状态下可以浏览但是不能进行例如查看详情等的操作。

（一）发布商品

发布闲置商品，要先在首页点击右下角圆圈，进入左图，再点击“加号”，权限允许后从相册中选择图片。



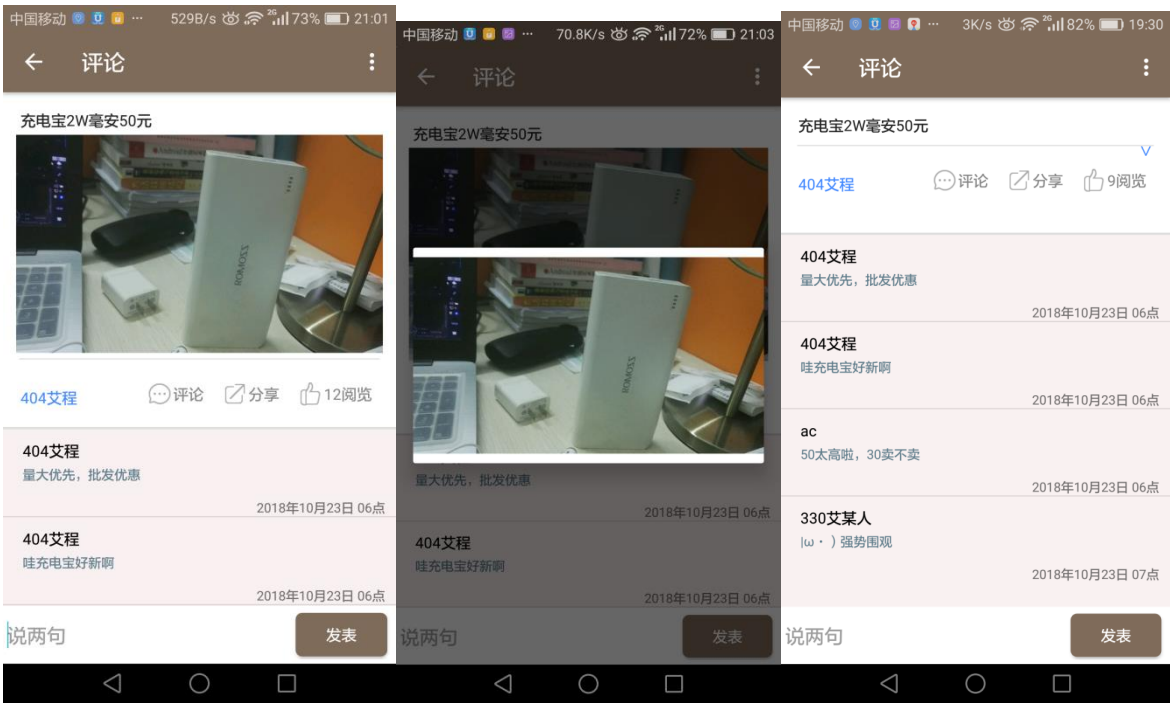
点击右下角的勾勾，可以在主页上看到，已经发布成功：

（二）上下滑动可以浏览不同的商品信息，下拉刷新商品信息。

三、商品详情页面

（一）查询商品信息

随意点击某个商品，进入“商品详细页”，点击图片可看大图，向上滑动可收起图片，节省空间。



（二）查询商户信息

点击商户昵称，可以查看该商户的信息和发布过的商品。列表中的商品也是可以点击的，会跳到相应的商品详情页

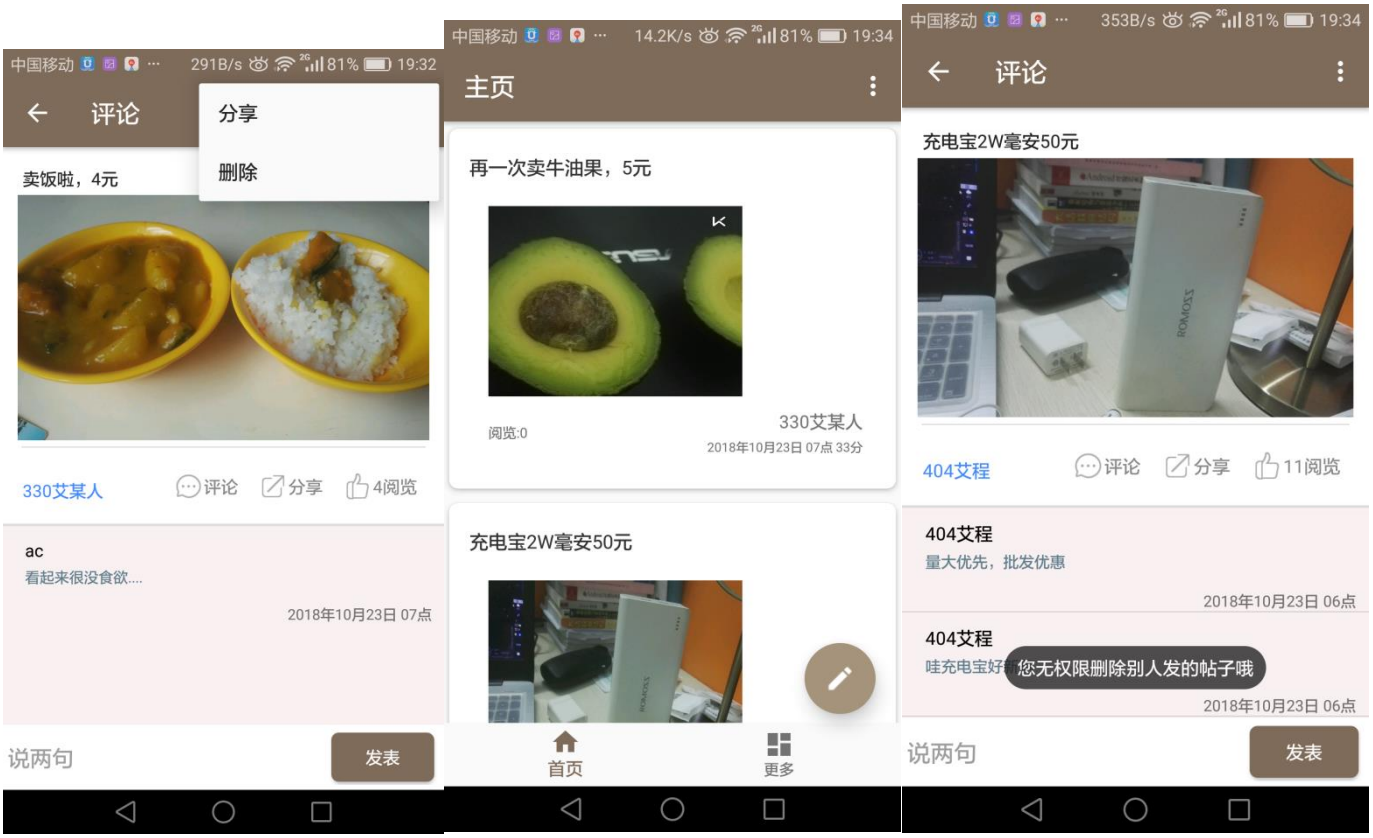
（三）商品的下架删除功能

（1） 当前用户删除自己商品情况（左、中图）

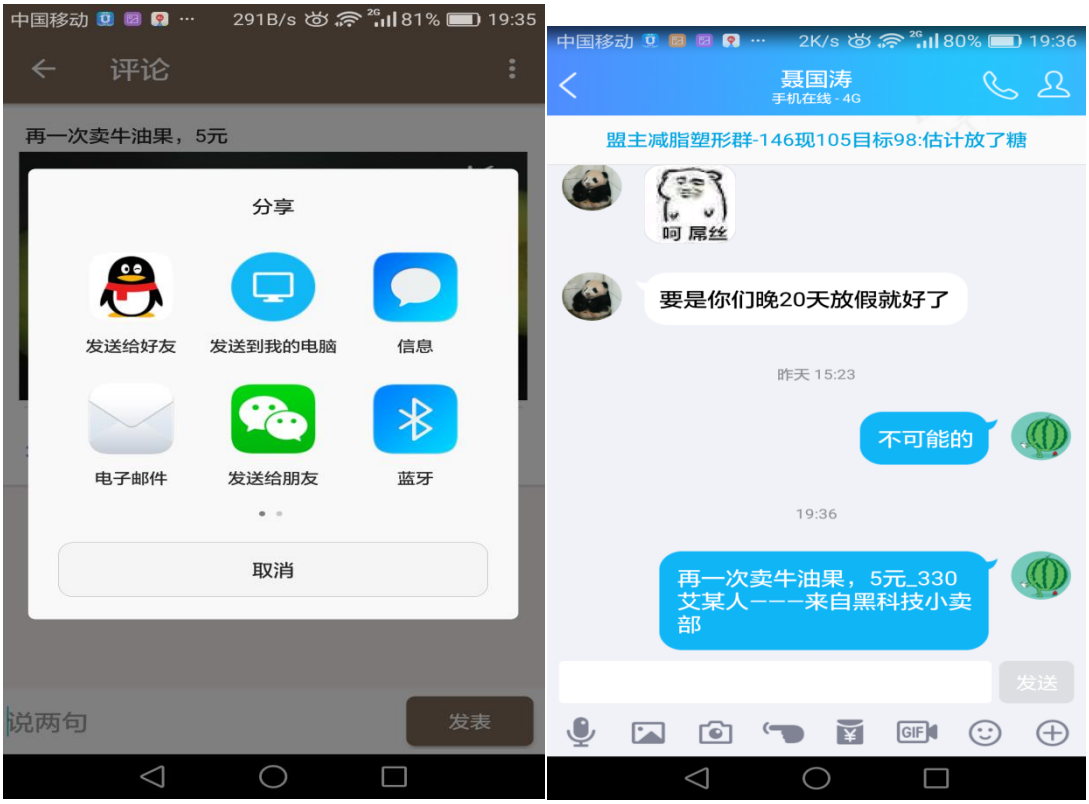
如中图所示，主页上已经没有已删除信息。删除成功。

（2）当前用户删除其他用户商品情况（右图）

如右图所示，会弹窗提醒用户不可删除其他用户商品。删除失败。



（四）商品的推广分享功能（以 QQ 为例）

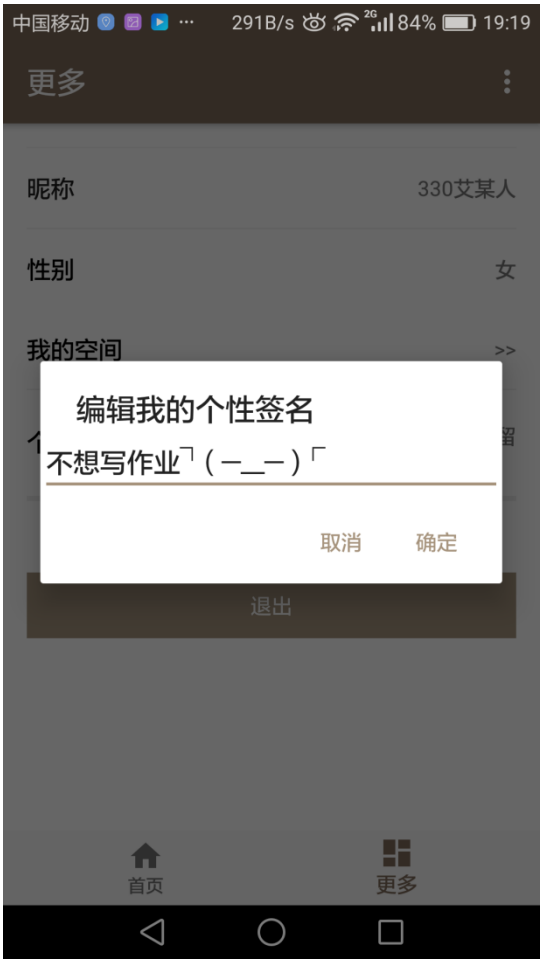


（五）商品的评论功能

四、个人信息界面

浏览个人信息的界面，和主页一样，需要登录才可以操作。

（一）修改个人信息

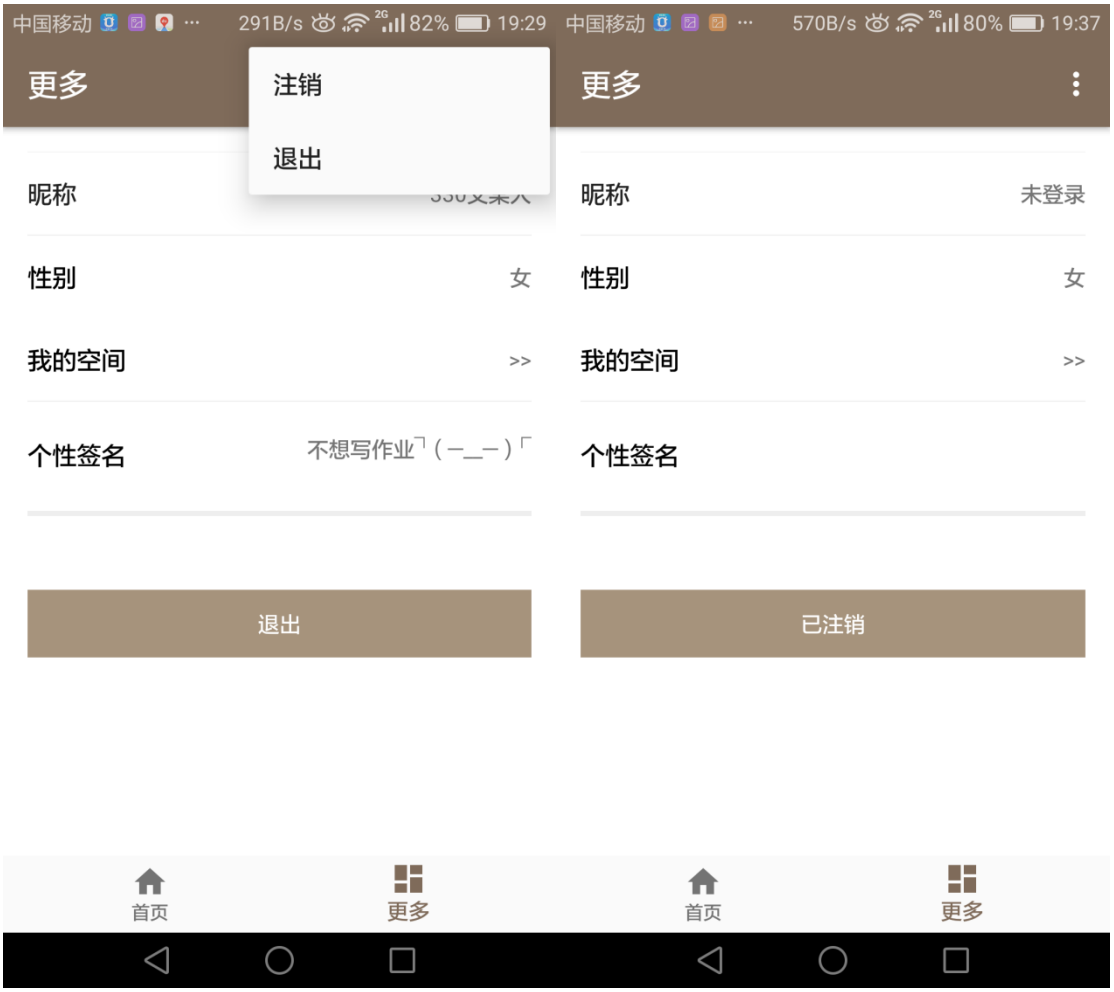

中国移动 291B/s 84% 19:19
更多
昵称 330艾某人
性别 女
我的空间 >>
编辑我的个性签名
不想写作业┐(－_－)┌
取消 确定
退出
首页
更多


正在保存屏幕截图...
更多
昵称 330艾某人
性别 女
我的空间 >>
个性签名 不想写作业┐(－_－)┌
退出
更新签名成功
首页
更多

（二）查询自己发布过的商品信息（列表内容可以点击进入相应的商品详情页面）

（三）注销/退出

注销用户有两种方式，可以通过右上角的菜单，也可以直接在个人信息页面点击退出。



五、总结

学到的知识

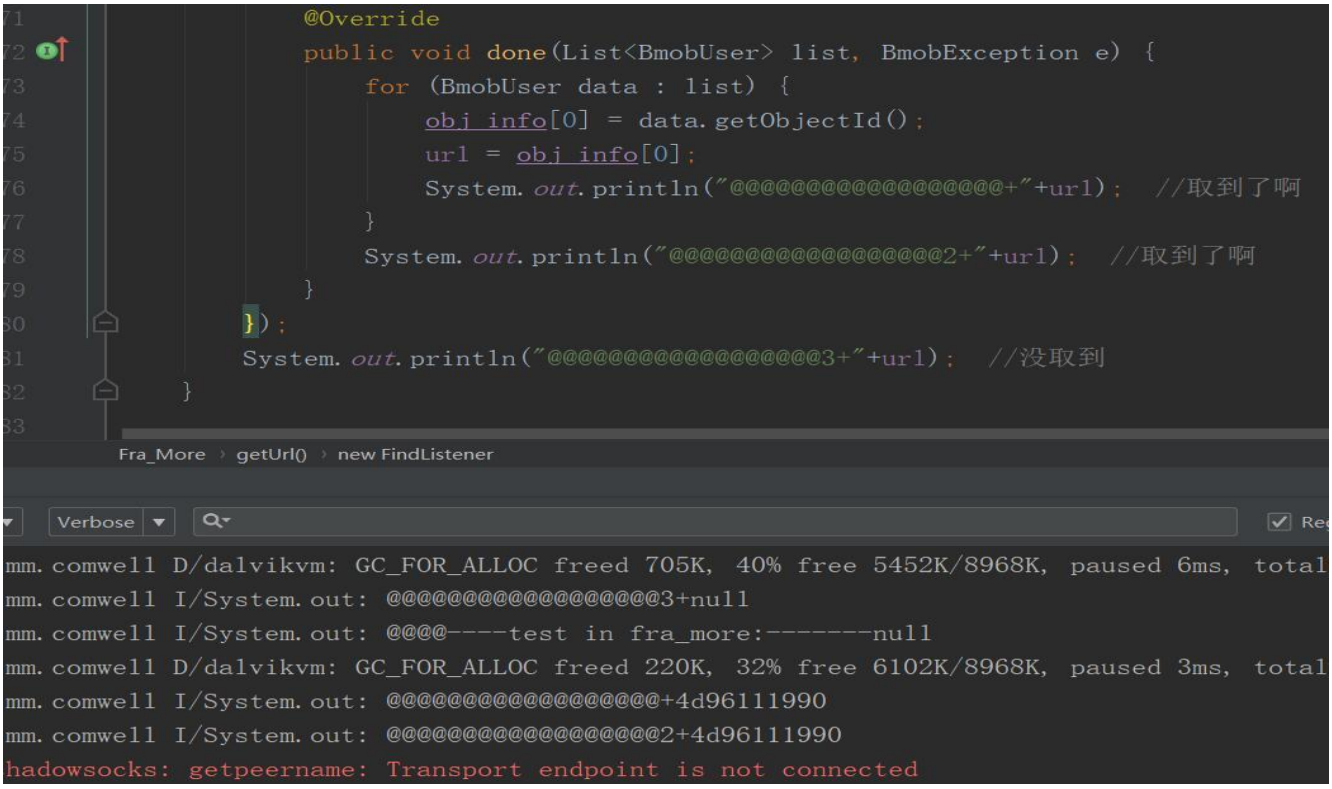
除了第三方 sdk，其余代码都参考了《第一行代码》。

1、新的高级控件 CardView 和 RecyleView 的使用
2、下拉刷新 SwipeRefreshLayout 的使用
3、悬浮按钮 FloatingActionButton 的使用
4、碎片 Fragment 的使用
5、弹窗类（进度条窗口、选择窗口、编辑文本窗口）的使用
6、知道怎么通过网络，用 app 连接自己后台的数据库数据表
7、更清晰了解使用第三方 sdk 开发 app 的流程
8、提升了解决 Bug 的能力
9、会利用 value/styles.xml 文件自定义 app 的样式
10、意图的使用与复习（显示：常规的界面跳转。隐式：例如调用相册、传递数值等）

遇到的问题

1、匿名监听类的数据传送问题

现象：



测试的时候发现，CommentActivity 和 Fra_More 中同样的代码，前者能得到 url 数据，后者不行。

原因：

我使用的监听都是匿名内部类，会出现内部类中的局部变量无法赋值给外部类的全局变量的情况。

因此我猜想这个和 android 内部结构的执行顺序有关，可能存在一个树状的层次结构，代码的执行顺序和编写代码的顺序不同。同样的代码，如果我放到 Activity 中的”onCreate()”方法中，则顺序是“1-2-3”。如果放在普通类中的某方法执行，则是“3-1-2”，导致 3 得不到 1 的值。

解决方法：

将取值代码放到 Fra_More 的 onActivityCreate()中强制其一开始就执行，而不是等到需要的时候再调用执行。

2、资源文件出现问题：Resources$NotFoundException: Resource is not a Drawable

原因：

由于我的 ui 图都是直接从网上找的，并不是自己设计的原生图片。所以尽管复制粘贴使用 ui 图的时候在 ide 上显示是正常的，程序在编译的时候会出现部分图片的格式错误，导致资源文件解析错误，出现这个报错。

常规办法：

网上的教程是将图片导入 Ps 软件中，转换图片格式为 png 进行修复。

我的办法：

然而我觉得有点麻烦。平常一般将资源文件放在 drawable 中，但是也有时候是放在 mipmap 中的。二者的区别是，后者常用于存放图标文件，可以提高系统渲染图片的速度，提高图片质量，减少 GPU 压力。mipmap 支持多尺度缩放，显然性能要更好一点，我就将出错的图片放置在 mipmap 中，发现问题解决了。

3、第三方库文件配置出错

解决：

按照 Bmob 官方文档进行错误排查即可。细节问题，例如漏掉了某必要的关键代码。

4、高级控件运行时没有预想的正确预览效果。

解决：

百度，发现一般是自己的代码有细节问题，例如在编写 RecyclerView 适配器配置 ViewHolder 的时

候会习惯地写“R.id.资源名字”，然而正确的应该是“R.layout.资源名字”。

app 的改进意见

1、 自动实时更新。
当有一个商品发布时，需要用户手动下拉刷新才可以收到商品消息，可以考虑改进下，隔一段时间自动更新或者使用其他实时技术实现自动更新。
2、 功能扩展。
例如评论区的互评功能以及商品的分类，有利于用户直接搜索到自己想要的闲置商品。然而由于时间有限，目前暂未实现该功能。
3、 更全面的分享。
目前该 app 的分享功能只是一段文字，可以考虑将产品图片以及产品连接也一同分享给他人。
4、 代码的解耦。
尽管前期代码是分开写的，但是随着需要实现的功能越来越多，对于部分功能的代码，我会图省事直接复制粘贴，导致多次复用同样代码，光是适配器代码，居然有 7 个实现类，代码大概都差不多的...为了日后更好维护，以及代码的简洁直观，后期必须考虑整理代码。
5、 细节的改善。
可以考虑做个图片适配器，防止浏览图片时，看到的尺寸和上传时不一样。
6、加入即时通讯模块，让买卖双方能够及时交流意见。

三、程序设计实现

注：由于篇幅有限，xml 代码省略不写，java 中的初始化控件代码也省略。仅写重要的功能实现代码。

工具类

一、第三方库的配置代码

```
public class BmobApplication extends Application {
    private List<Activity> mList = new LinkedList<Activity>();
    private static BmobApplication instance;
    //我在官网申请的 key，必须要有，否则算非法使用 sdk 服务，功能无法正常运行
    public static String APPID ="c93e110e041738076e17e3cb355915e4";
    public synchronized static BmobApplication getInstance() {        //固定写法调用第三方库连接数据库的服务
        if (null == instance) {
            instance = new BmobApplication();                //生成新的服务
        }
        return instance;
    }
    //必须重写，正常情况下低内存运行。
    @Override
    public void onLowMemory() {
        super.onLowMemory();
        System.gc();
```

```
    }
    @Override
    public void onCreate() {
        super.onCreate();
        Bmob.initialize(this,APPID,"bmob");      // 使用第三方 sdk 操作数据库服务时的初始化操作
        BmobPush.setDebugMode(true);
        BmobInstallation.getCurrentInstallation().save();     // 启动数据库 crud 操作的服务
        BmobPush.startWork(this,APPID);             //设置第三方库的 BmobConfig 配置文件
        BmobConfig config =new BmobConfig.Builder(BmobApplication.this)
                .setConnectTimeout(15).build();             //请求超时时间（单位为秒）：默认 15s
    }
}
```

二、用户个人信息的封装（User 类，属性以及属性的 get、set 方法）
三、商品信息的封装（Comment 类，属性和 get、set 方法）
四、用于传递数据库信息对象的封装（Post 类，属性和 get、set 方法）
五、RecylerView 适配器 MyAdapter

```
public class MyAdapter extends RecyclerView.Adapter<MyAdapter.ItemViewHolder> implements View.OnClickListener {
    private LayoutInflater myInflater;              //相当于配置布局所需的管理器
    private OnRecyclerViewItemClickListener mOnItemClickListener = null;   //自定义监听
    private List<Post> list;                        //商品列表的信息
    private String user_obj;                        //商品所属用户信息
    private BmobUser user;                          //第三方 sdk 的用户实例
    private Context context;                        //全局上下文

    public MyAdapter(Context context, List<Post> list) {
        this.context=context;           //传入当前上下文
        this.list = list;               //传入数据
        myInflater = LayoutInflater.from(context);      //布局信息的上下文
    }

    //设置绑定子布局
    @Override
    public ItemViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = myInflater.inflate(R.layout.card_item, parent, false);
        ItemViewHolder vh = new ItemViewHolder(view);
        view.setOnClickListener(this);
        return vh;
    }

    //子布局内控件的具体操作与显示
    @Override
    public void onBindViewHolder(final ItemViewHolder holder, final int position) {
        final Post post = list.get(position);           //通过列表索引得到当前列表对象
        if (list.size()<0){
            return;
        }                                               //根据得到的列表对象信息
        holder.content.setText(post.getContent());     //改变子布局中空间的文本信息
```

```
        holder.time.setText(post.getTime());
        holder.tv_user.setText(post.getName());
        holder.tvPraise.setText("阅览:"+post.getPraise());
        if (post.getImg_url()==null){                    //如果商品没有图片，就将此控件设置为不可见
            holder.post_image.setVisibility(View.GONE);
        }
        else {                                           //否则显示
            holder.tv_url.setText(post.getImg_url());

Glide.with(context).load(post.getImg_url()).placeholder(R.mipmap.loading).thumbnail(0.3f).error(R.mipmap.loading).dontAnimate().into(holder.post_image);
        }
        //将数据保存在 itemView 的 Tag 中，以便点击时进行获取（定位、锁定操作的控件）
        holder.itemView.setTag(this);
        //获取 POST USER OBJID
        if (mOnItemClickListener != null) {
            holder.rl_post.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {     //一整个卡片的监听
                    user=BmobUser.getCurrentUser();        //利用第三方 sdk 得到当前用户对象
                    if(user==null){                  //如果还没有用户（即没登陆），提示
                        Toast.makeText(context,"登陆后查看详情",Toast.LENGTH_SHORT).show();
                        return;
                    }
                    mOnItemClickListener.onItemClick(holder.itemView,                post.getContent()+","+post.getName()+","+post.getObjectId()+","+post.getPraise()
                            +","+post.getImg_url());//自定义监听，传递正在操作的列表对象信息
                }
            });
        }
    }

    @Override
    public int getItemCount() {
        return list.size();
    }

    //触发点击事件，获取被点击 view 的信息，再传递出去
    @Override
    public void onClick(View v) {
        if (mOnItemClickListener != null) {
            //注意这里使用 getTag 方法获取数据
            mOnItemClickListener.onItemClick(v, (String) v.getTag());
        }
    }

    //找到 recyclerView 单个子布局中的所有空间
    class ItemViewHolder extends RecyclerView.ViewHolder {
        CardView rl_post;          //与 xml 控件的关联代码，略。
```

```
            ......
        ItemViewHolder(View itemView) {
            super(itemView);
            rl_post = (CardView) itemView.findViewById(R.id.card_view); //卡片式布局
              .....
        }
    }
    //自定义监听接口，当用户点击时传递被点击子布局的信息，然后再进行跳转
    public static interface OnRecyclerViewItemClickListener {
        void onItemClick(View view, String data);
    }
    //设置自定义接口
    public void setOnItemClickListener(OnRecyclerViewItemClickListener listener) {
        this.mOnItemClickListener = listener;
    }
}
```

六、弹窗的消息提示类

```
  private void showDialog() {
        builder = new AlertDialog.Builder(getActivity());    //builder 为 AlertDialog.Builder
        builder.setTitle("Material Design Dialog");          //弹窗消息
        builder.setNegativeButton("取消", null);             //取消按钮
        builder.setPositiveButton("确定", null);             //确认信息按钮
        builder.show();
    }
     public void show_dialogimg(){
            //  首先得到整个 View
            View view = LayoutInflater.from(this).inflate(       //自定义布局生成 view 对象
                    R.layout.dialog_comm_img, null);
            ImageView diaimageView=(ImageView)view.findViewById(R.id.dialog_img);

Glide.with(CommentActivity.this).load(dialog_url).placeholder(R.mipmap.loading).into(diaimageView);
            al= new AlertDialog.Builder(this)           //设置自定义布局的图片资源信息
                    .setView(R.layout.dialog_comm_img)
                    .setView(view)
                    .show();                            //显示自定义布局
        }
```

主要实现类

二、注册/登录界面

主要结合第三方库实现和数据库的信息匹配或者信息传送。

逻辑代码类：RegisterAvticity.java、LoginActivity.java

页面布局：register_layout.xml、login_layout.xml

（一）注册操作的方法代码

```
private void register() {
    name=et_name.getText().toString();   //获得用户名以及密码
    pasd=et_pasd.getText().toString();
    if(name.isEmpty()){
        toast("用户名为空");          //名字为空时，浮窗提醒用户
```

```
            return;
        }
        if(pasd.isEmpty()){
            toast("密码为空");        //密码为空时，提醒用户
            return;
        }
        User user=new User();       //创建用户实例，用于将注册的用户信息导入数据库
        user.setUsername(name);
        user.setPassword(pasd);
        user.setSex("男");                          //默认性别
        user.signUp(new SaveListener<User>() {        //导入数据库
            @Override
            public void done(User s, BmobException e) {
                if(e==null){          //如果插入数据不报错
                    toast("注册成功！");
                    Intent intent = new Intent();//成功，则显示意图跳转到主页面
                    intent.setClass(RegisterActivity.this,MainActivity.class);
                    startActivity(intent);
                    finish();
                }else{
                    toast("注册失败！  "+e.toString()); //提示注册失败
                }
            }
        });

}
```

（二）登录操作的方法代码

登陆成功会自动跳转到主页面并且浮窗提示“登陆成功”。

```
private void login() {
        name=log_user.getText().toString();   //获取用户名和密码
        password=log_pasd.getText().toString();
        if (name.isEmpty()) {
            toast("用户名为空");    //用户名为空时，消息提醒
            return;
        }
        if (password.isEmpty()) {
            toast("密码为空");    //密码为空时，消息提醒
            return;
        }
        User user = new User();         //创建 user 实例，用于得到数据库信息并匹配
        user.setUsername(name);
        user.setPassword(password);
        user.login(new SaveListener<BmobUser>() {
            @Override            //调用第三方 sdk 的函数进行数据库和用户信息的数据匹配
            public void done(BmobUser bmobUser, BmobException e) {
                if(e==null){                                    //如果数据库返回没有报错
                    toast("登录成功！");                              //则提示登录成功
```

```
                    Intent intent = new Intent();                    //显示意图跳转到主界面
                    intent.setClass(LoginActivity.this, MainActivity.class);
                    startActivity(intent);
                    finish();
                }else{
                    toast("登录失败！  " + e.toString());                    //提示登录失败
                }
            }
        });
}
```

三、主页界面

app 的首界面，用于浏览商品信息，每个大功能都在某个类中存放着，主页通过 intent 进行跳转以及信息传递。

逻辑代码：Fra_main.java，继承于 Fragment 类

页面布局：fragment_main.xml、card_item.xml（CardView 的子布局）

（一）发布商品

逻辑代码：PhotoActivity.java、PerActivity.java

页面布局：editor_layout.xml、user_photo_layout.xml

发布闲置商品，要先在首页点击右下角圆圈，进入左图，再点击“加号”按钮，权限允许后从相册中选择图片，并生成进度条弹窗提示进度，成功后跳转界面至商品详情。

```
/*
调用打开相册，选取相片内容
*/
private void openAlbum() {
    Intent intent = new Intent("android.intent.action.GET_CONTENT");        //隐式意图调用相册
    intent.setType("image/*");
    startActivityForResult(intent, CHOOSE_PHOTO);       //  打开相册
}

/*
选择图片
 */
public void choose_img(View view) {
    //动态权限申请，申请打开访问相册
    if(ContextCompat.checkSelfPermission(EditActivity.this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(EditActivity.this,    new    String[]{    Manifest.permission.
WRITE_EXTERNAL_STORAGE }, 1);
    } else {
        openAlbum();                    //打开系统相册
    }
}
/*
获取用户“打开相册”权限，否则不操作
*/
```

```
@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
    switch (requestCode) {
        case 1:
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                openAlbum();
            } else {
                Toast.makeText(this, "你拒绝了相册使用权限", Toast.LENGTH_SHORT).show();
            }
            break;
        default:
    }
}

/*
该代码用于传递 PhotoActivity 和 PerActivity 的数据，即发布商品时的图片资源信息传递。
同时上传商品信息给数据库。
*/
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    //如果请求码和响应码正常，说明两个 activity 可以正常传输数据。
    if (requestCode == RESULT_LOAD_IMAGE && resultCode == RESULT_OK && null != data) {
        Uri selectedImage = data.getData();
        String[] filePathColumn = { MediaStore.Images.Media.DATA };
        Cursor cursor = getContentResolver().query(selectedImage,      //生成可指向数据表的游标
                filePathColumn, null, null, null);
        cursor.moveToFirst();                                          //利用游标遍历主页数据
        int columnIndex = cursor.getColumnIndex(filePathColumn[0]);
        path = cursor.getString(columnIndex);
        cursor.close();                                  //遍历完成需要关闭游标
        Handler h = new Handler();                       //网络耗时操作，开启异步线程处理
        h.postDelayed(new Runnable() {                   //上传数据到网络
            @Override
            public void run() {
                dialog = new ProgressDialog(PhotoActivity.this);      //创建进程可视化的进程条
                dialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL); //设置进程条样式
                dialog.setTitle("上传中...");                          //进程条弹窗提示
                dialog.setIndeterminate(false);                        //设置进程条“具体”显示进度
                dialog.setCancelable(true);                            //防止用户误触导致进城取消
                dialog.setCanceledOnTouchOutside(false);        //如果触发点击事件，不影响进程条
                dialog.show();                                  //进程条显示
                final BmobFile file = new BmobFile(new File(path));  //创建需要上传的文件实例对
象
                file.uploadblock(new UploadFileListener() {      //调用第三方 sdk 上传文件到数据库
                    @Override
                    public void done(BmobException p1) {
                        final Photo photo = new Photo();         //需要上传的图片
                        photo.setPhoto(file);                    //图片路径
                        photo.setUser(user);                     //图片所属商户
```

```
                                photo.save(new SaveListener<String>() {    //将图片上传
                                    @Override
                                    public void done(String p1, BmobException p2) {
                                        // TODO: Implement this method
                                        if (p2 == null) {       //网络报错响应码为空，则成功
                                            toast("成功上传图片"+user.getUsername()+file.getUrl());
                                            String img_url=file.getUrl();       //获取图片路径
                                            photo.setImg_url(img_url);
                                            photo.save(new SaveListener<String>() {
                                                @Override
                                                public void done(String s, BmobException e) {
                                                    if (e==null){
                                                        toast("上传 url 成功");//上传成功后跳转界面
                                                        Intent                                  intent=new
Intent(PhotoActivity.this,PerActivity.class);
                                                        startActivity(intent);    //跳转到商品详情页
                                                    }else{
                                                        toast("上传 url 失败");      //提示失败
                                                    }
                                                }
                                            });
                                            dialog.dismiss();           //进度条弹窗消失
                                        } else {
                                            toast(p2.toString() + "失败") ;     //提示失败
                                            dialog.dismiss();
                                        }
                                    }
                                });
                            }
                            @Override
                            public void onProgress (Integer value){
                                    dialog.setProgress(value);       //  返回的上传进度（百分比）
                            }
                        });
                    }
            },1000*2);

        }

}
```

（二）按照时间顺序，浏览所有的商品列表，利用 RecylerView 实现上下滑动浏览不同的商品信息，利用 SwipeRefreshLayout 实现下拉刷新商品信息。
逻辑代码：Fra_main.java，继承于 Fragment 类
页面布局：fragment_main.xml、card_item.xml（CardView 的子布局）

```
  private void getData() {
        // TODO Auto-generated method stub
```

```
        this.refresh.setRefreshing(true);        //refresh 为 SwipeRefreshLayout，下拉刷新
        final Post post = new Post();            //生成一个用户对象，准备传递给数据库
        BmobQuery<Post> bq = new BmobQuery<Post>();          //第三方 sdk 对象，实现数据传递
        bq.setLimit(6);
        bq.order("-updatedAt");                              //设置需要传递的信息
        bq.include("author");
        bq.include("username");
        bq.include("time");
        bq.include("title");
        bq.include("img_url");
        bq.include("user.objectId");
        mlist.clear();
        bq.findObjects(new FindListener<Post>() {            //从数据库中找用户的所有信息
            @Override
            public void done(List<Post> list, BmobException e) {
                if (e==null) {
                    for (Post p : list) {                    //遍历得到所有信息
                        p.getContent();
                        p.getName();
                        p.getTitle();
                        p.getTime();
                        mlist.add(p);            //加到列表中，mlist 列表为 RecylerView 的数据

                        refresh.setRefreshing(false);        //刷新完成，ui 显示关闭
                        linearLayoutManager.setScrollEnabled(true);//设 RecyclerView 的列表可滑动
                        adapter.notifyDataSetChanged();      //监听 RecyclerView 列表数据变化

                    }
                }else{
                    toast("加载失败" );                      //浮窗提示失败
                    refresh.setRefreshing(false);            //设 RecyclerView 的列表可滑动
                    adapter.notifyDataSetChanged();          //监听 RecyclerView 列表数据变化
                }
            }
        });
    }
```

四、商品详情页面
逻辑代码：CommentActivity.java
页面布局：activity_comment.xml

（一）查询商品信息
随意点击某个商品，进入“商品详细页”，点击图片可看大图，向上滑动可收起图片，节省空间。

```
/*
使用第三方库提供的方法接口，完成查询数据库商品表中的具体信息。
*/
 public void getUrl() {
        showDialog();            //弹窗提示信息，显示“数据装载中”
```

```
        final String[] obj_info = {""};
           //固定写法，创建第三方库的数据库数据对象，取得数据。
        final BmobQuery<BmobUser> query = new BmobQuery<BmobUser>();
query.addWhereEqualTo("username", username.getText().toString());
        query.findObjects(new FindListener<BmobUser>() {   //匿名调用监听接口
            @Override
            public void done(List<BmobUser> list, BmobException e) {
                al.dismiss();                       //关闭弹窗 al
                for (BmobUser data : list) {       //遍历某一行的所有数据
                    obj_info[0] = data.getObjectId();
                    url = obj_info[0];              //遍历到的数据记录到 url 中
                }
            }
        });
    }
/*
弹窗提示信息
*/
    private void showDialog() {
        //  首先得到整个 View
        LayoutInflater inflater = getLayoutInflater();
        al= new AlertDialog.Builder(this)           //设置弹窗内的具体内容
                .setCancelable(false)
                .setTitle("数据装载中...")
                .setView(R.layout.dialog_com)
                .setCancelable(true)
                .show();

    }
/*
显示图片
*/
     public void showImg(View view){
        if (img_url==null){       //img_url 为全局变量，如果有资源文件则放入控件中
            comm_img.setVisibility(View.GONE);

        }
    }
```

（三）查询商户的信息（个人信息+商品信息）

代码逻辑：PersonalActicity.java

界面布局：user_info_layout.xml

点击商户昵称，可以查看该商户的信息和发布过的商品。列表中的商品也是可以点击的，会跳到相应的商品详情页

```
/*
查询商户的所有信息
*/
 void person_post(){
        User user =new User();
```

```
        user.setObjectId(objId);
        BmobQuery<Post> query = new BmobQuery<Post>();        //使用第三方 sdk 的对象方法功能
        query.addWhereEqualTo("author", user);       //  查询当前用户的所有帖子
        query.include("author");//  希望在查询帖子信息的同时也把发布人的信息查询出来
        query.findObjects(new FindListener<Post>() {  //查询当前用户（商户）的所有信息
            @Override
            public void done(List<Post> list, BmobException e) {
                if (e==null){                          //如果第三方 sdk 传送数据没有出错的话
                    for(Post p: list) {                //遍历该用户发布过的所有商品信息
                        p.getContent();
                        p.getTitle();
                        p.getTime();
                        mlist.add(p);
                    }
                }else{
                    toast(e.toString());           //出错则浮窗显示报错
                }
            }
        });
    }

/*
点击图片后进行查看图片操作
*/
  private void displayImage(String imagePath) {       //传入图片路径
        if (imagePath != null) {
            Bitmap bitmap = BitmapFactory.decodeFile(imagePath);         //根据路径得到图片对象
            select_img.setImageBitmap(bitmap);                           //在控件中展示图片
        } else {
            Toast.makeText(this, "获取图像失败", Toast.LENGTH_SHORT).show();   //提示失败
        }
    }
```

（三）商品的下架删除功能

```
public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()){
            case    R.id.del:
                user=BmobUser.getCurrentUser(User.class);
                Post p = new Post();           //数据库数据对象
                p.setObjectId(obj);            //需要操作对象的 id
                if (this.user.getObjectId().equals(url)){
                    p.delete(new UpdateListener() {     //调用第三方 sdk 实现删除功能
                        @Override
                        public void done(BmobException e) {
                            if(e==null){
```

```
                                toast("删除成功");
                            }else{
                                toast("失败："+e.getMessage()+","+e.getErrorCode());
                            }
                        }
                    });
                }else {
                    toast("您无权限删除别人发的帖子哦");
                }
}
```

（四）商品的推广分享功能

```
 public void share(View view){
        Intent intent=new Intent(Intent.ACTION_SEND);  //直接使用隐式意图，实现。
        intent.setType("text/*");
        intent.putExtra(Intent.EXTRA_SUBJECT, "分享");
        intent.putExtra(Intent.EXTRA_TEXT,content.getText().toString()
+"_"+username.getText()+"———来自黑科技小卖部");                    //设置分享给他人的文本
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(Intent.createChooser(intent, "分享"));
    }
```

（五）商品的评论功能

```
 private void publishComment(String content){
        getTime();              //得到当前时间
        user = new User();      //创建当前用户对象，以便第三方 sdk 操作数据库记录评论信息
        user=BmobUser.getCurrentUser(User.class);
        if(user == null){                           //如果当前没有登录
            toast("发表评论前请先登陆");       //浮窗提示
            return;
        }else if(TextUtils.isEmpty(content)){       //如果当前评论为空，浮窗提示
            toast("发表评论不能为空");
            return;
        }
        final Comment comment = new Comment();        //被评论的商品的所有信息
        comment.setContent(content);
        comment.setPost(weibo);
        comment.setUser(user);
        comment.setTime(time);
        comment.save(new SaveListener<String>() {       //调用第三方 sdk 保存该商品的信息
            @Override
            public void done(String s, BmobException e) {
                if (e==null){               //如果第三方 sdk 操作数据无错
                    findComments();       //查询该商品的所有评论（看到自己的评论）
                    toast("评论成功");
                    adapter.notifyDataSetInvalidated();      //recyclerView 适配器中的方法，当列表
数据变化时，view 也发生变化
                    ed_comm.setText("");
                }else{
                    toast(e.toString());
```

```
                }
            }
        });
    }
/*
查找当前商品的所有评论信息
*/
 private void findComments(){
        //利用第三方 sdk，创建连接数据库的对象
         BmobQuery<Comment> query = new BmobQuery<Comment>();
         list.clear();
         Post post = new Post();              //创建提供数据给数据库信息的对象
         post.setObjectId(obj);               //插入数据到表时必要的主键 id
         query.addWhereEqualTo("post",new BmobPointer(post));  //通过第三方 sdk，以 post 的方式传递信息给数据库。
         query.include("user,,author,post.author,comment.time,comment.user");
         query.findObjects(new FindListener<Comment>() {
             @Override
             public void done(List<Comment> arg0, BmobException e) {
                 if (e == null) {                    //上传数据如果第三方 sdk 没有出错
                     list.addAll(arg0);
                     com_num = list;                 //提取所有的评论信息
                     adapter.notifyDataSetInvalidated();       //如果数据列表发生改变则更新
                 } else {

                     toast("查询评论失败");                 //否则失败
                     adapter.notifyDataSetChanged();
                 }
             }
         });}
```

四、个人信息界面

逻辑代码类：Fra_More.java，继承于 Fragment

页面布局：

（一）修改个人信息

逻辑代码：PersonalActivity.java

页面布局：R.layout.user_info_layout

```
protected void onCreate(@Nullable Bundle savedInstanceState) {
         super.onCreate(savedInstanceState);
         setContentView(R.layout.user_info_layout);
     BmobQuery<User> query = new BmobQuery<User>();          //借助第三方 sdk 取得个人信息
         query.getObject(objId, new QueryListener<User>() {
             @Override
             public void done(User user, BmobException e) {
                 if (e==null){                           //如果第三方 sdk 功能没有出错
                     if (TextUtils.isEmpty(user.getWords())){
                         user_info_word.setText("这个小孩很懒，什么都没留下");
                     }else {
                         user_info_word.setText(user.getWords());
```

```
                    }
                    if (user.getSex().equals("男")){          //设置修改性别信息
                        user_info_sex.setText("(帅哥)");
                    }else if (user.getSex().equals("女")){
                        user_info_sex.setText("(美女)");
                    }
                    user_info_name.setText(user.getUsername());
                }else{
                    toast(e.toString());                      //提示错误信息
                }
            }
        });
        person_post();                                        //修改完成的信息通过第三方 sdk 提供给数
据库
}
```

（二）查询自己发布过的商品信息（列表内容可以点击进入相应的商品详情页面）

```
void person_post(){
    User user =new User();                          //创建一个用户实例
    user.setObjectId(objId);
    BmobQuery<Post> query = new BmobQuery<Post>();
    query.addWhereEqualTo("author", user);          // 查询当前用户的所有帖子
    query.order("-updatedAt");
    query.include("author");// 希望在查询帖子信息的同时也把发布人的信息查询出来
    query.findObjects(new FindListener<Post>() {
        @Override
        public void done(List<Post> list, BmobException e) {
            if (e==null){                           //如果第三方 sdk 不报错
                for(Post p: list) {                 //则取得该用户帖子的所有信息
                    p.getContent();
                    p.getTitle();
                    p.getTime();
                    mlist.add(p);
                }
            }else{
                toast(e.toString());                //报错则打印信息
            }
        }
    });

}
```

（三）注销/退出

```
tv_log.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (user==null){                    //用户已注销，重新跳转到登录界面
            Intent intent = new Intent();
            intent.setClass(getActivity(), LoginActivity.class);
            startActivity(intent);
```

```
            }else {
                BmobUser.logOut();              //用户
                 toast("已注销，重启程序生效");    //浮窗提示用户注销
                 tv_name.setText("未登录");       //个人信息界面回复默认数值
                 tv_word.setText("");
                 tv_log.setText("已注销");          //将该按钮设置成注销状态，如果重新点击则能再登录
            }
        }
});
```