# 中山大学数据科学与计算机学院本科生实验报告

## （2017 年秋季学期）

课程名称：手机应用平台开发　　任课教师： 刘宁

| | | | |
|---|---|---|---|
| 年级 | 15 级软件工程 | 专业（方向） | 计算机应用 |
| 学号 | 15331236 | 姓名 | 马朝露 |
| 电话 | 15521122675 | Email | 1073627941@qq.com |
| 开始日期 | 2017/10/21 | 完成日期 | 2017/10/23 |

## 一、 实验题目

Broadcast 使用

## 二、 实现内容

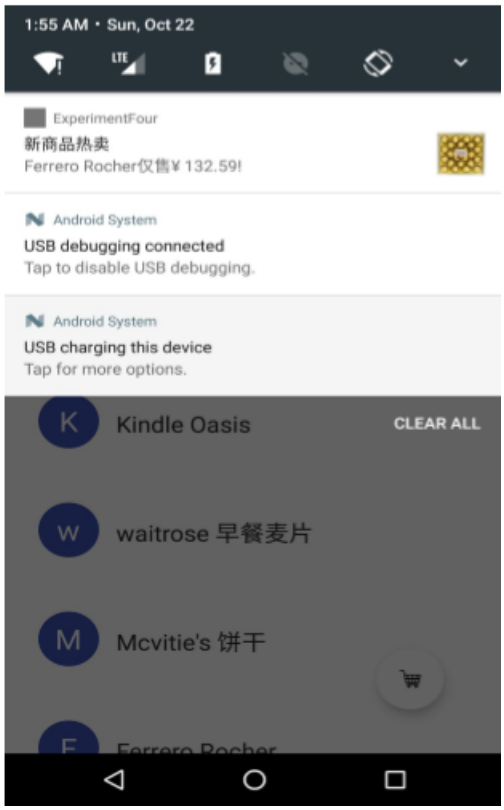
【实验目的】 1、掌握 Broadcast 编程基础 2、掌握动态注册 Broadcast 和静态注册 Broadcast 3、掌握Notification 编程基础 4、掌握 EventBus 编程基础

【实验内容】

在实验三的基础上，实现静态广播、动态广播两种改变Notification 内容的方法。

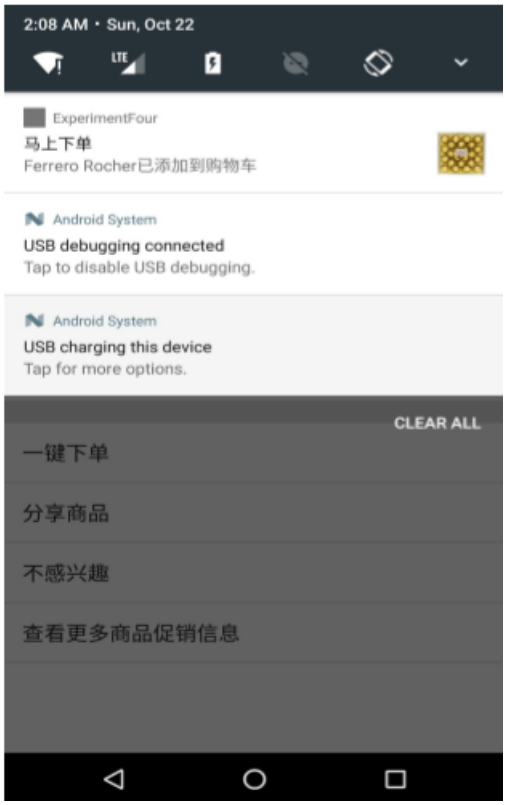
具体要求：

(1)在启动应用时，会有通知产生，随机推荐一个商品:

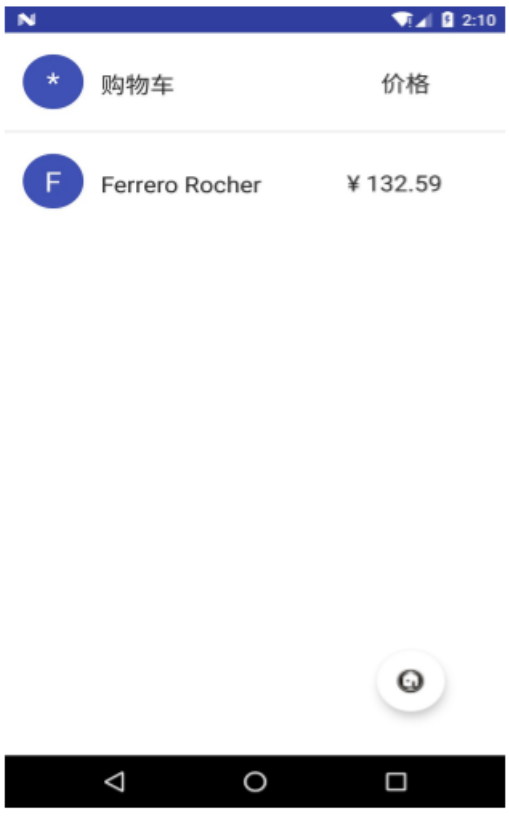



(2)点击通知跳转到该商品详情界面:


(3)点击购物车图标，会有对应通知产生，并通过Eventbus在购物车列表更新数据:




(4)点击通知返回购物车列表:

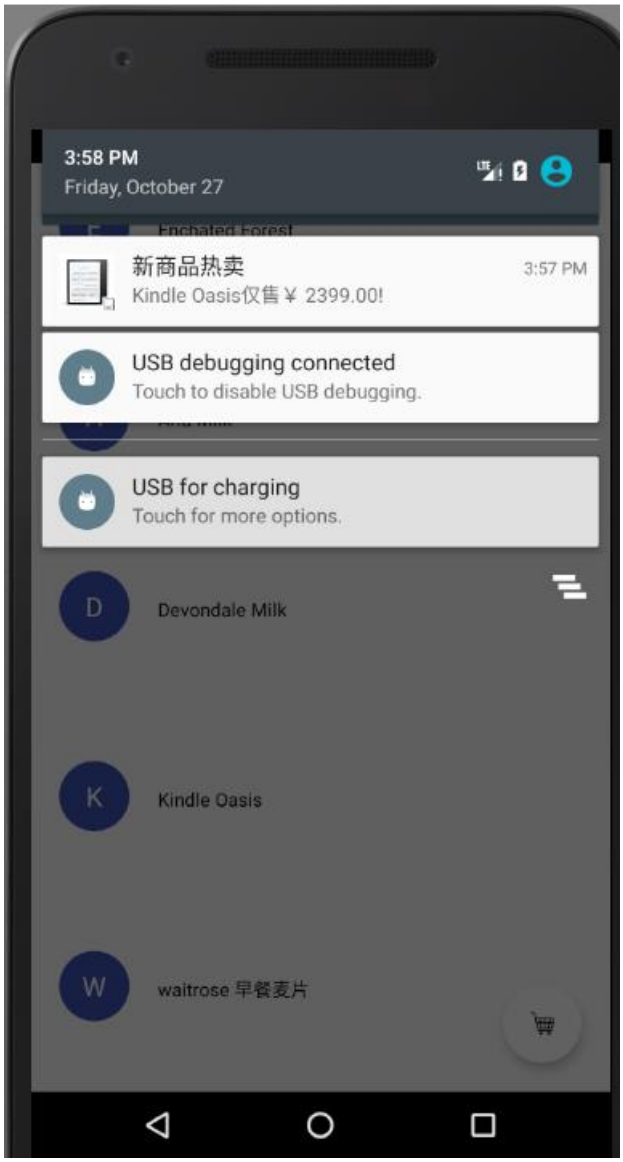



(5)实现方式要求:启动页面的通知由静态广播产生，点击购物车图标的通知由动态广播产生。

## 三、 课堂实验结果

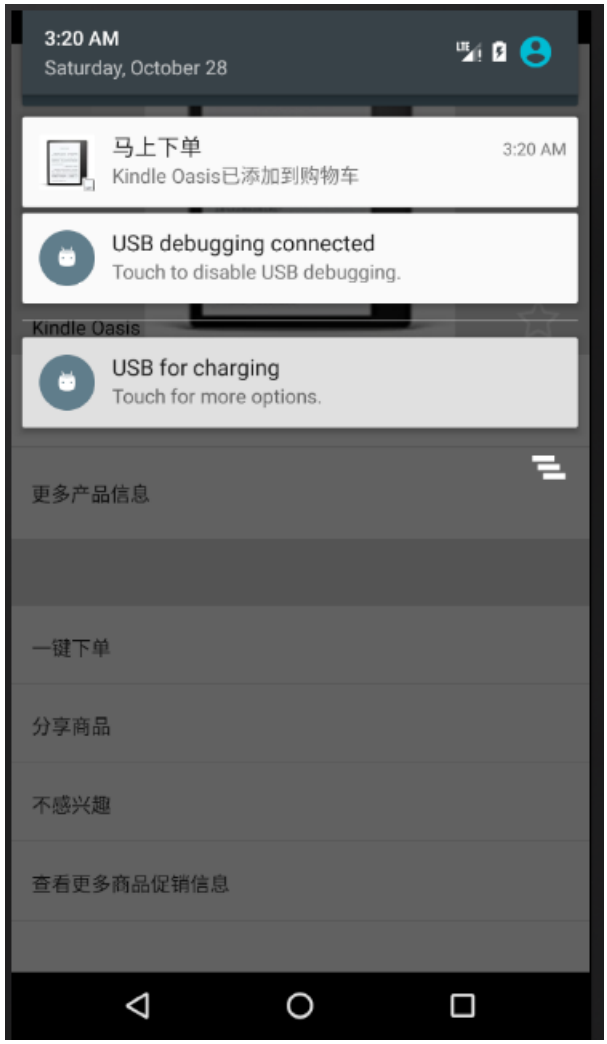
（1） 实验截图

在启动应用时，会有通知产生，随机推荐一个商品:

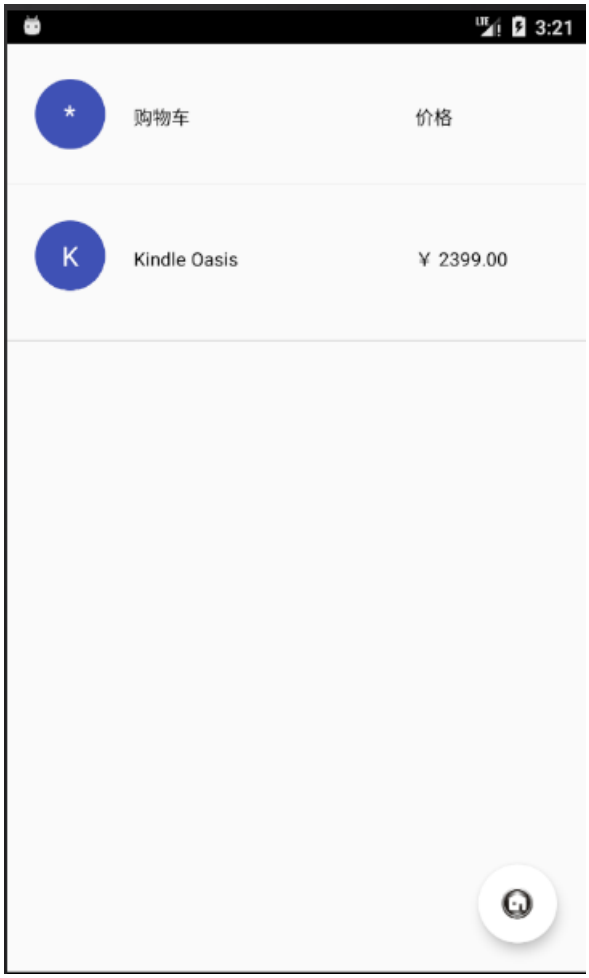



点击通知跳转到该商品详情界面:


(3)点击购物车图标，会有对应通知产生，并通过Eventbus在购物车列表更新数据:




(4)点击通知返回购物车列表:




（2） 实验步骤以及关键代码

【静态广播】

使用随机数:

```
Random random=new Random();
int random_int = random.nextInt(10);
random_name = name_[random_int];
if (random_name == "Enchated Forest") {
    random_image = R.mipmap.enchatedforest;
    random_price="￥ 5.00";
    random_birth="作者 Johanna Basford";
    random_initial="E";
} else if (random_name == "Arla Milk") {
    random_image=R.mipmap.arla;
    random_price="￥ 59.00";
    random_birth="产地 德国";
    random_initial="A";
} else if (random_name == "Devondale Milk") {
    random_image=R.mipmap.devondale;
    random_price="￥ 79.00";
    random_birth="产地 澳大利亚";
    random_initial="D";
} else if (random_name == "Kindle Oasis") {
    random_image=R.mipmap.kindle;
    random_price="￥ 2399.00";
    random_birth="版本 8GB";
    random_initial="K";
} else if (random_name == "waitrose 早餐麦片") {
    random_price="￥ 179.00";
    random_image=R.mipmap.waitrose;
    random_birth="重量 2Kg";
```

传递数据及发送广播：

```
Intent intentBroadcast = new Intent();
intentBroadcast.setAction("STATICACTION");
Bundle bundlebroadcast = new Bundle();
bundlebroadcast.putString("random_name",random_name);
bundlebroadcast.putInt("random_image",random_image);
bundlebroadcast.putString("random_price",random_price);
bundlebroadcast.putString("random_birth",random_birth);
bundlebroadcast.putString("random_initial",random_initial);
intentBroadcast.putExtras(bundlebroadcast);
sendBroadcast(intentBroadcast);
```

接收器：

```java
public void onReceive(Context context, Intent intent) {
    // TODO: This method is called when the BroadcastReceiver is receiving
    // an Intent broadcast.
    Bundle bundle=intent.getExtras();
    random_name = bundle.getString("random_name");
    random_price = bundle.getString("random_price");
   random_image = bundle.getInt("random_image");
    random_birth = bundle.getString("random_birth");
    random_initial = bundle.getString("random_initial");
    Intent clickintent = new Intent(context,detials.class);
    Bundle bundle_ = new Bundle();
    bundle_.putString("initial_1",random_initial);
    bundle_.putString("name2",random_name);
    bundle_.putString("birth",random_birth);
    bundle_.putString("price2",random_price);
    bundle_.putInt("image",random_image);
    clickintent.putExtras(bundle_);
    PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, clickintent, PendingIntent.FLAG_CANCEL_CURRENT);
    NotificationManager notifyManager = (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
    Notification.Builder builder = new Notification.Builder(context);
    builder.setContentTitle("新商品热卖").
            setLargeIcon(BitmapFactory.decodeResource(context.getResources(),random_image)).
            setSmallIcon(random_image).setContentText(random_name+"仅售"+random_price+"!").
            setContentIntent(pendingIntent).setAutoCancel(true);
    Notification notify = builder.build();
    notifyManager.notify(0,notify);
```

过滤部分：

```xml
<receiver
    android:name=".MyReceiver"
    android:enabled="true"
    android:exported="true">
    <intent-filter>
        <action android:name="STATICACTION" />
    </intent-filter>
</receiver>
```

【动态广播】

注册及注销：

```
@Override
protected void onStart() {
    super.onStart();
    // 实例化IntentFilter对象
    IntentFilter filter = new IntentFilter();
    filter.addAction("dynamic");
    dreceiver = new dynamicReceiver();
    // 注册广播接收
    registerReceiver(dreceiver,filter);
}
@Override
protected void onStop() {

    unregisterReceiver(dreceiver);
    super.onStop();
}
```

传递数据及发送：

```
        @Override
        public void onClick(View v) {
            /*Map<String, Object> map = new HashMap<String, Object>();
            map.put("initial_1",s3);
            map.put("name_1",s1);
            map.put("price",s2);
            list.add(map);
            shopList.mDatas_ = list;
            s. this, "商品已加入购物车",Toast.LENGTH_SHORT).show();*/
            ntent();
            amic");
            Bundle bundle_ = new Bundle();
            bundle_.putString("shop_name",s1);
            bundle_.putInt("shop_image",shop_image);
            intent.putExtras(bundle_);
            sendBroadcast(intent);
            EventBus.getDefault().postSticky(new MessageEvent(s3,s1,s2));
        }
    });
    Bundle bundle=this.getIntent().getExtras();
```

droid API 25 Platform >] android.os
c final class Bundle extends BaseBundle
ements Cloneable, Parcelable

接收器：

```
public class dynamicReceiver extends BroadcastReceiver {
    private String shop_name;
    private int shop_image;
    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO: This method is called when the BroadcastReceiver is receiving
        // an Intent broadcast.
        Bundle bundle=intent.getExtras();
        shop_name = bundle.getString("shop_name");
        shop_image = bundle.getInt("shop_image");
        Intent clickintent = new Intent(context,shopList.class);
        PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, clickintent, 0);
        NotificationManager notifyManager = (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
        Notification.Builder builder = new Notification.Builder(context);
        builder.setContentTitle("马上下单").
                setLargeIcon(BitmapFactory.decodeResource(context.getResources(),shop_image)).
                setSmallIcon(shop_image).setContentText(shop_name+"已添加到购物车").
                setContentIntent(pendingIntent).setAutoCancel(true);
        Notification notify = builder.build();
        notifyManager.notify(0,notify);
    }
}
```

【Eventbus的使用】

声明一个事件类：

```
public class MessageEvent {
    public final String message1;
    public final String message2;
    public final String message3;

    public MessageEvent(String message1, String message2, String message3) {
        this.message1 = message1;
        this.message2 = message2;
        this.message3 = message3;
    }
}
```

传递数据：

```
@Override
public void onClick(View v) {
    /*Map<String, Object> map = new HashMap<String, Object>();
    map.put("initial_1",s3);
    map.put("name_1",s1);
    map.put("price",s2);
    list.add(map);
    shopList.mDatas_ = list;
     Toast.makeText(detials.this,"商品已加入购物车",Toast.LENGTH_SHORT
    Intent intent = new Intent();
    intent.setAction("dynamic");
    Bundle bundle_ = new Bundle();
    bundle_.putString("shop_name",s1);
    bundle_.putInt("shop_image",shop_image);
    intent.putExtras(bundle_);
    sendBroadcast(intent);
    EventBus.getDefault().postSticky(new MessageEvent(s3,s1,s2));
}
```

准备注册以及注销订阅者：

```
@Override
public void onStart() {
    super.onStart();
    EventBus.getDefault().register(this);
}


@Subscribe(threadMode = ThreadMode.MAIN,sticky = true)
public void handleSomethingElse(MessageEvent event) {
   //TextView t = (TextView)findViewById(R.id.initial_1);
   // t.setText(event.message1);
    Map<String, Object> map = new HashMap<~>();
    map.put("initial_1",event.message1);
    map.put("name_1",event.message2);
    map.put("price",event.message3);
    mDatas_.add(map);
    adapter.notifyDataSetChanged();
}

@Override
public void onStop() {
    EventBus.getDefault().unregister(this);
    super.onStop();
}
```

（3） 实验遇到困难以及解决思路

1. 点击新商品热卖后跳到详情界面，但是没有产生对应的详情

解决方法：

创建 PendingIntent 的时候需要注意参数 PendingIntent.FLAG_CANCEL_CURRENT 这个标志位用来指示：如果当前的 Activity 和 PendingIntent 中设置的 intent 一样，那么久先取消当前的 Activity，用 PendingIntent 中指定的 Activity 取代之。并且需要在 Manifest 中对指定的 Activity 设置属性：android.lauchMode=”singleInstance”

2. 点击新商品热卖后跳到详情页面，点击购物车加入购物车后，再点击主页面。然后任意点击一个 item，跳到的详情页面却是刚才点击新商品热卖时的详情页面

解决方法：不要设置主页面的 android.lauchMode=”singleInstance”。

3. 使用 eventbus 传递数据时在接受数据的 java 文件中不知道何时接受注册何时注销。解决方法：看了官方文档后知道了应该在 onStart 中注册，在与其对应的 onStop 中注销。

4.不知道动态广播应该何时注册何时注销。解决方法，看了以前得 ppt，查了安卓 activity 的生命周期几种状态转化。决定在 onstart 中注册，onstop 中注销。

## 四、 实验思考及感想

这次的代码量又很少，可能老师想让我们轻松一下。深刻的了解了静态广播与动态广播的使用，也学会了简单的使用 eventbus 在两个 java 文件中传递数据。

但是我得到的最大的收获是了解了安卓 activity 的生命周期。如下图：

Running状态：一个新的Activity启动入栈后，它在屏幕最前端，处于栈的最顶端，此时它处于可见并可和用户交互的激活状态。

Paused状态：当Activity被另一个透明或者Dialog样式的Activity覆盖时的状态。此时它依然与窗口管理器保持连接，系统继续维护其内部状态，它仍然可见，但它已经失去了焦点，故不可与用户交互。

Stopped状态：当Activity不可见时，Activity处于Stopped状态。当Activity处于此状态时，一定要保存当前数据和当前的UI状态，否则一旦Activity退出或关闭时，当前的数据和UI状态就丢失了。

Killed状态：Activity被杀掉以后或者被启动以前，处于Killed状态。这是Activity已从Activity堆栈中移除，需要重新启动才可以显示和使用。

4种状态中，Running状态和Paused状态是可见的，Stopped状态和Killed状态时不可见的。

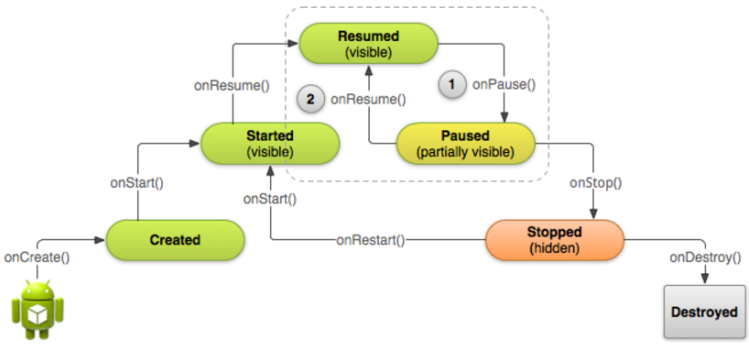
onStart()和onResume()的区别

onStart()是activity界面被显示出来的时候执行的，用户可见，包括有一个activity在他上面，但没有将它完全覆盖，用户可以看到部分activity但不能与它交互

onResume()是当该activity与用户能进行交互时被执行，用户可以获得activity的焦点，能够与用户交互。

onStart()通常就是onStop()（也就是用户按下了home键，activity变为后台后），之后用户再切换回这个activity就会调用onRestart()而后调用onStart()

onResume()是onPause()（通常是当前的acitivty被暂停了，比如被另一个透明或者Dialog样式的Activity覆盖了），之后dialog取消，activity回到可交互状态，调用onResume()。



作业要求：

1. 命名要求: 学号_姓名_实验编号，例如 15330000_林 XX_lab1。
2. 实验报告提交格式为 pdf。
3. 实验内容不允许抄袭，我们要进行代码相似度对比。如发现抄袭，按 0 分处理。