

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	专业 (方向)	计应
学号	15331421	姓名	郑佳锋
电话	13719325472	Email	1773643139@qq.com
开始日期	10.28	完成日期	10.29

一、 实验题目

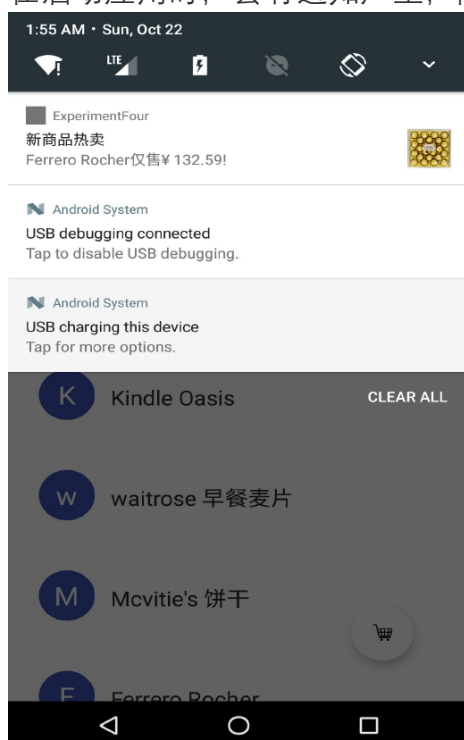
Lab6

二、 实现内容

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：

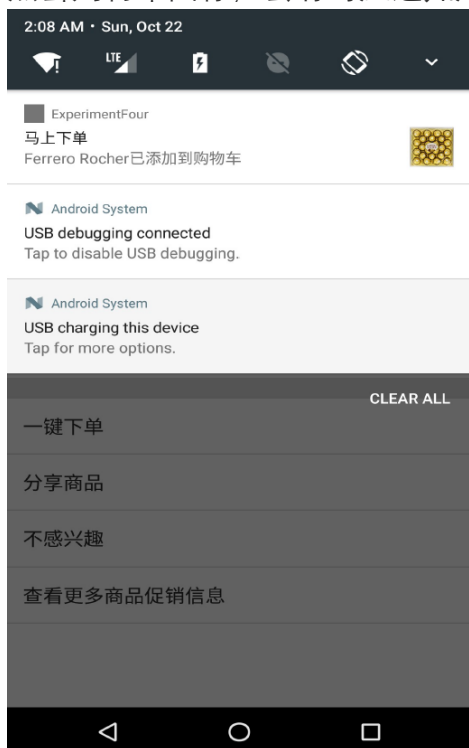
(1) 在启动应用时，会有通知产生，随机推荐一个商品：



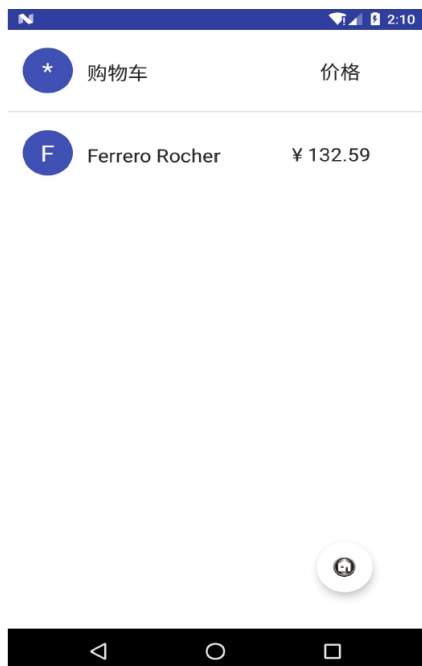
(2) 点击通知跳转到该商品详情界面：



(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。

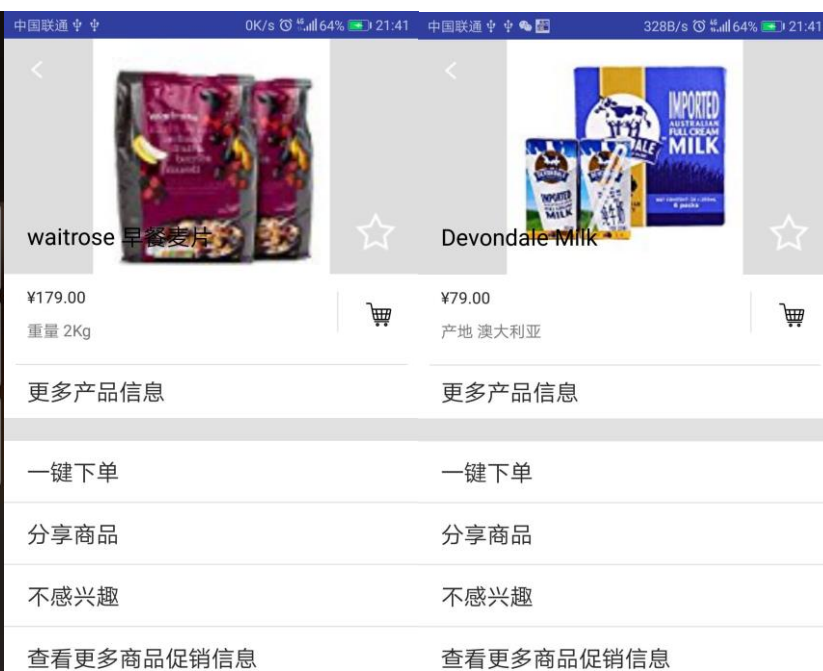
三、 课堂实验结果

(1) 实验截图

启动应用：

点击通知栏：

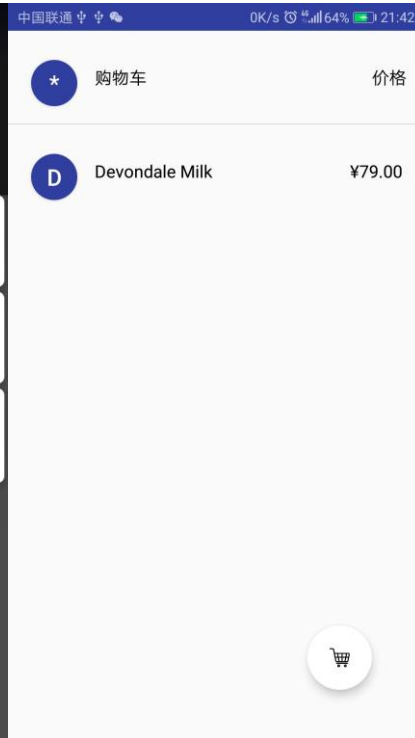
点击购物车图标：



通知栏：



点击通知栏：



(2) 实验步骤以及关键代码

实现静态广播：

在主界面 MainActivity 中，产生一个随机数，代表商品编号。通过编号提取去商品名称、价格、图片，作为 intent 的附带信息，intent 用 setAction 设置唯一标示，最后用 sendBroadcast 发送。代码如下：

```
1. Random random = new Random();
2. int ranNum = random.nextInt(mDatas.size());
3. Intent intent = new Intent();
4. intent.setAction("MLY");
5. intent.putExtra("name", mDatas.get(ranNum));
6. intent.putExtra("price", name_price.get(mDatas.get(ranNum)));
7. intent.putExtra("picture", name_pic.get(mDatas.get(ranNum)));
8. sendBroadcast(intent);
```

接着，就要定义接送广播的类 MyReceiver 了，该类继承自 BroadcastReceiver。需要先在 AndroidManifest.xml 中注册，注册时 intent 过滤器的名称与上面 intent 的唯一标示相同。

```
1. <receiver
2.     android:name=".MyReceiver"
3.     android:enabled="true"
4.     android:exported="true"
5. >
6.     <intent-filter>
7.         <action android:name="MLY" />
8.     </intent-filter>
```

9. </receiver>

MyReceiver 类定义如下：

```
1. public class MyReceiver extends BroadcastReceiver {
2.     public MyReceiver(){}
3.     @Override
4.     public void onReceive(Context context, Intent intent) {
5.         Notification.Builder builder = new Notification.Builder(context);
6.         Bitmap bm = BitmapFactory.decodeResource(context.getResources(), Integer.parseInt(
7.             intent.getStringExtra("picture")));
8.         Intent appIntent=null;
9.         appIntent = new Intent(context,NewActivity.class);
10.        appIntent.putExtra("passname",intent.getStringExtra("name"));
11.        appIntent.setAction(Intent.ACTION_MAIN);
12.        appIntent.addCategory(Intent.CATEGORY_LAUNCHER);
13.        appIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK| Intent.FLAG_ACTIVITY_RESET_TASK_IF_NEEDED);
14.        PendingIntent contentIntent =PendingIntent.getActivity(context, 0,appIntent,PendingIntent.FLAG_UPDATE_CURRENT);
15.        builder.setContentTitle("新商品热卖")
16.            .setContentText(intent.getStringExtra("name") +"仅售"
17.                + intent.getStringExtra("price") + "!")
18.            .setSmallIcon(Integer.parseInt(intent.getStringExtra("picture")))
19.            .setLargeIcon(bm)
20.            .setContentIntent(contentIntent)
21.            .setAutoCancel(true);
22.        NotificationManager manager = (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
23.        Notification notify = builder.build();
24.        manager.notify(0,notify);
25.    }
26. }
```

当接收到 intent 消息时，调用 Notification 类产生一条通知。通知要设置标题、信息、大图标、小图标，同时为了使点击后跳到商品详情页面，可以使用 PendingIntent，将该类收到的信息发送给新页面。

实现动态广播：当点击购物车图标时，商品添加到购物车列表（已经在 Lab5 中使用 EventBus 实现），并在通知栏显示信息。因此，在购物车的点击事件中添加 Intent 并发送广播。

```
1. shopbus.setClickListener(new View.OnClickListener() {
2.     @Override
3.     public void onClick(View v) {
4.         Toast.makeText(NewActivity.this, "商品已加入购物车", Toast.LENGTH_SHORT).show();
5.         EventBus.getDefault().post(
6.             new FirstEvent(passname, name_price.get(passname)));
7.         Intent intent = new Intent();
8.         intent.setAction("shop");
9.         intent.putExtra("name",passname);
10.        intent.putExtra("picture", name_pic.get(passname));
11.        sendBroadcast(intent);
12.    }
13. });
```

在跳转到的页面 MainActivity 中注册广播：注意 addAction 与上面的 setAction 对应

```
1. IntentFilter filter = new IntentFilter();
2. filter.addAction("shop");
3. registerReceiver(dynamicReceiver,filter);
```

同时在 MainActivity 的 onDestroy 函数中解除注册：

```
1. protected void onDestroy(){
2.     super.onDestroy();
3.     EventBus.getDefault().unregister(this);//反注册 EventBus
4.     unregisterReceiver(dynamicReceiver);
5. }
```

接着，定义动态广播接收类 DynamicReceiver，并在里面设置通知。（与静态广播类似）

```
1. class DynamicReceiver extends BroadcastReceiver{
2.     @Override
3.     public void onReceive(Context context, Intent intent) {
4.         Notification.Builder builder = new Notification.Builder(context);
5.         Bitmap bm = BitmapFactory.decodeResource(context.getResources(), Integer.parseInt(intent.getStringExtra("picture")));
6.         Intent appIntent=null;
7.         appIntent = new Intent(context,MainActivity.class);
8.         appIntent.putExtra("open","open");
9.         appIntent.setAction(Intent.ACTION_MAIN);
10.        appIntent.addCategory(Intent.CATEGORY_LAUNCHER);
11.        appIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK| Intent.FLAG_ACTIVITY_RESET_TASK_IF_NEEDED);
12.        PendingIntent contentIntent =PendingIntent.getActivity(context, 0,appIntent,PendingIntent.FLAG_UPDATE_CURRENT);
13.        builder.setContentTitle("马上下单")
14.            .setContentText(intent.getStringExtra("name") +"已添加到购物车")
15.            .setSmallIcon(Integer.parseInt(intent.getStringExtra("picture")))
16.            .setLargeIcon(bm)
17.            .setContentIntent(contentIntent)
18.            .setAutoCancel(true);
19.        NotificationManager manager = (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
20.        Notification notify = builder.build();
21.        manager.notify(0,notify);
22.    }
23. }
```

同样，用 PendingIntent 导航到另一个 MainActivity 页面。在 MainActivity 的 onNewIntent 函数中，设置购物车列表为可见。

```
1. protected void onNewIntent(Intent intent) {
2.     super.onNewIntent(intent);
3.     setIntent(intent);
4.     Bundle bundle = intent.getExtras();
5.     if(bundle != null){
6.         myRecyclerView.setVisibility(View.INVISIBLE);
7.         shopList.setVisibility(View.VISIBLE);
8.     }
9. }
```

（3）实验遇到困难以及解决思路

本次实验比较简单，所以遇到的困难不多。

在实现点击通知栏跳转到 MainActivity 的时候，最开始把处理 Bundle 的操作放到了 onCreate 函数里面，发现一直不起作用。后来请教同学后才知道，采用 Intent 跳转到 MainActivity 后，由于 MainActivity 已经存在，所以不需要再次调用 onCreate 方法，所以我放在 onCreate 的操作不会生效。所以，只能采用另一个函数 onNewIntent，这个函数在每次收到 Intent 时都会被执行，如下：

```
1. protected void onNewIntent(Intent intent) {  
2.     super.onNewIntent(intent);  
3.     setIntent(intent);  
4.     Bundle bundle = intent.getExtras();  
5.     if(bundle != null){  
6.         myRecyclerView.setVisibility(View.INVISIBLE);  
7.         shopList.setVisibility(View.VISIBLE);  
8.     }  
9. }
```

另一个困难是在通知栏设置大图标时，一开始不知道 setLargeIcon 的参数是啥，以为是和 setSmallIcon 一样传入图片的 id，所以一直报错，后来查看函数才知道 setLargeIcon 的参数是一个 Bitmap，所以使用以下语句获得图片：

```
1. Bitmap bm = BitmapFactory.decodeResource(context.getResources(), Integer.parseInt(intent  
    .getStringExtra("picture")));
```

四、实验思考及感想

本次实验比较简单，内容比较少，主要是静态广播和动态广播的使用，两个小时内可以完成。通过这次实验，我对静态广播和动态广播的区别有了更加深刻的认识。两者的实现其实非常相似，需要定义一个广播接收类，然后再里面重写 onReceive 方法。不同的是，静态广播需要在 AndroidManifest.xml 文件中注册，而动态广播只需要在使用的地方进行注册、解除注册，除了实现不同，两者在实际中用途也大不相同，如优先级、生命周期等等。