中山大学移动信息工程学院本科生实验报告

（2017 年秋季学期）

课程名称：移动应用开发　　任课教师：

年级	15352272	专业（方向）	软件工程（移动信息工程）
学号	15352272	姓名	彭国栋
电话	15626066058	Email	578291308@qq.com
开始日期		完成日期	

一、 实验题目

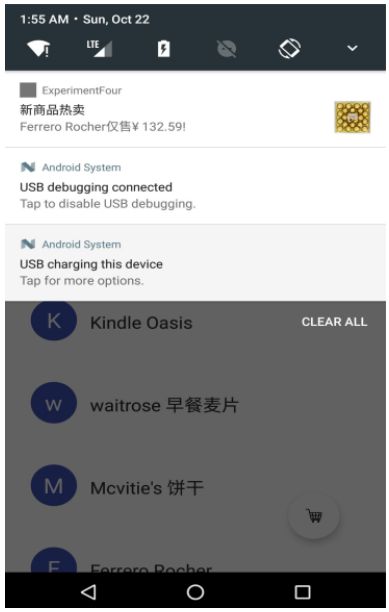
Broadcast 使用

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。

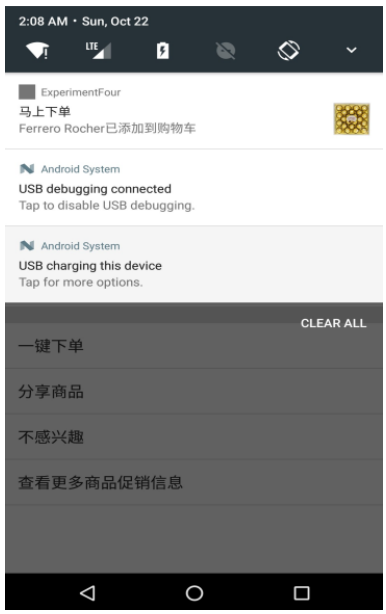
具体要求：

(1)在启动应用时，会有通知产生，随机推荐一个商品:

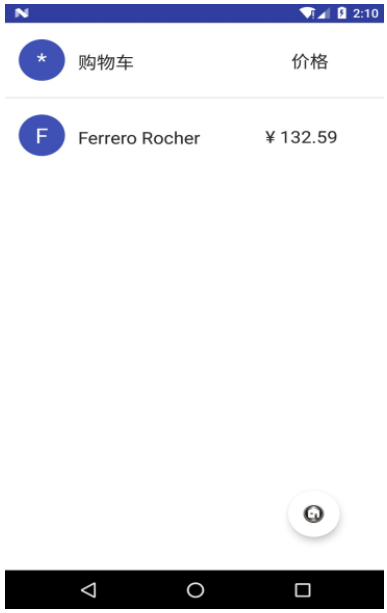


(2)点击通知跳转到该商品详情界面:

(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:



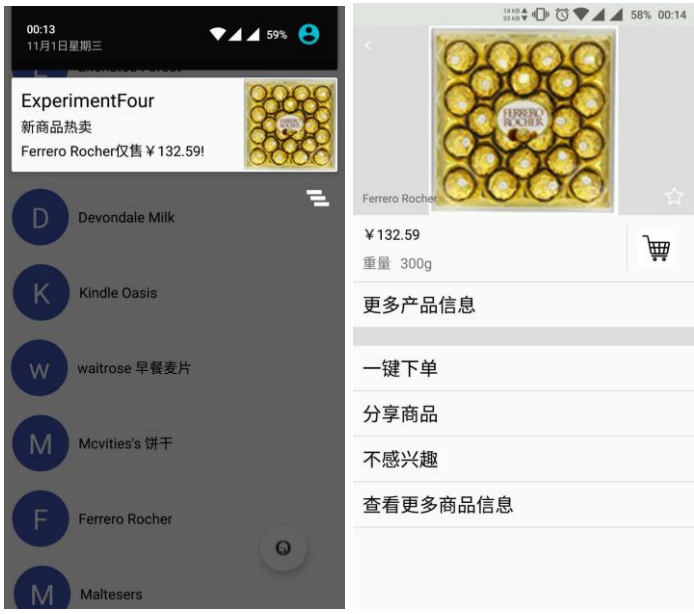
(4)点击通知返回购物车列表:

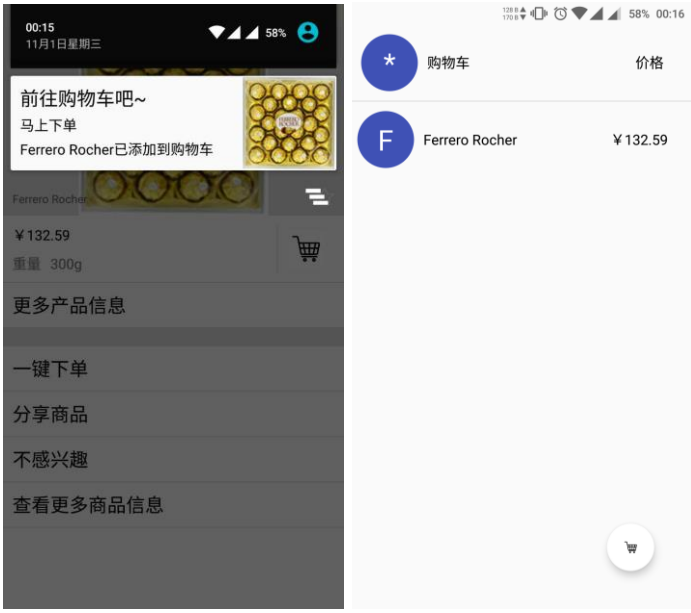


(5)实现方式要求:启动页面的通知由静态广播产生，点击购物车图标的通知由动态广播产生。

三、 课堂实验结果

（1） 实验截图





（2） 实验步骤以及关键代码

这次实验在上次的基础上进行添加功能，主要分为静态广播和动态广播部分：

1、首先是通知的样式设计：这里采用 RelativeLayout 来实现

```
<ImageView
    android:id="@+id/notification_icon"
    android:layout_width="300px"
    android:layout_height="300px"
    android:src="@mipmap/arla"
    android:layout_alignParentRight="true"
    />
<TextView
    android:id="@+id/notification_title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="aaa"
    android:layout_alignParentLeft="true"
    android:textColor="@color/colorBlack"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="10dp"
    android:textSize="20dp"
    />
```

```
<TextView
    android:id="@+id/content1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@id/notification_title"
    android:layout_marginTop="5dp"
    android:layout_marginLeft="10dp"
    android:text="nnn"
    android:textSize="15dp"
    android:textColor="@color/colorBlack"
    />
<TextView
    android:id="@+id/content2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@id/content1"
    android:layout_marginTop="5dp"
    android:text="aaa"
    android:textSize="15dp"
    android:layout_marginLeft="10dp"
    android:layout_alignParentBottom="true"
    android:textColor="@color/colorBlack"
    />
```

样式如下所示：

2、静态广播部分：

新建一个 Receiver 类，并在 AndroidMainifest.xml 中设置：

```
<receiver android:name=".Receiver">
    <intent-filter>
        <action android:name="MystaticFliters" />
    </intent-filter>
</receiver>
```

首先在 MainActivity 中发送一个静态的广播。

```
//创建receiver
Random random=new Random();//随机函数
Bundle sendBundle=new Bundle();//发送的视图
sendBundle.putSerializable("random",goodsDetail.get(random.nextInt(goodsName.length)));
//随机生成一个n并在goodsDetail列表中随机找到一个数据发送过去
Intent intentBroadcast=new Intent(Receiver.STATICACTION);//一个跳转通知
intentBroadcast.putExtras(sendBundle);
sendBroadcast(intentBroadcast);//发送广播
```

在静态广播中实现的内容为：

```
manager = (NotificationManager)
        context.getSystemService(Context.NOTIFICATION_SERVICE);

Intent intentFromReceiver=new Intent();
if(intent.getAction().equals(STATICACTION))
{
    temp=(Goods) intent.getExtras().get("random");//获取ramdom物品
    intentFromReceiver.setClass(context,GoodDetail.class);//设置跳转
    intentFromReceiver.putExtra("goods",temp);//设置传输的key和对应的商品
    mPenddingIntent= PendingIntent.getActivity(context,0,intentFromReceiver,PendingIntent.FLAG_UPDATE_CURRENT);
    /*FLAG_UPDATE_CURRENT会更新之前PendingIntent的消息，
    比如，你推送了消息1，并在其中的Intent中putExtra了一个值“ABC”，
    在未点击该消息前，继续推送第二条消息，并在其中的Intent中putExtra了一个值“CBA”，
    好了，这时候，如果你单击消息1或者消息2，你会发现，他俩个的Intent中读取过来的信息都是“CBA”，
    就是说，第二个替换了第一个的内容*/
    remoteViews=new RemoteViews(context.getPackageName(),R.layout.layout_notification);
    /*绑定RemoteViews的布局*/
    remoteViews.setImageViewResource(R.id.notification_icon,temp.getImageid());
    /*设置布局中的图片*/
    remoteViews.setTextViewText(R.id.notification_title,"ExperimentFour");//设置标题
    remoteViews.setTextViewText(R.id.content1,"新商品热卖");//设置内容1
    remoteViews.setTextViewText(R.id.content2,temp.getName()+"仅售"+temp.getPrice()+"!");//设置内容2

    Notification.Builder builder=new Notification.Builder(context);
    builder.setContentIntent(mPenddingIntent)
            .setSmallIcon(temp.getImageid())
            .setContentText("·")
            .setAutoCancel(true);
    Notification notification=builder.build();
    notification.bigContentView=remoteViews; //可以显示更多内容的通知
    //通过这种方式来绑定布局
    manager.notify(0,notification);
}
```

2、动态广播

新建一个 DynamicReceiver 类，不需要在 AndroidMainifest.xml 中设置 receiver。

借助 eventBus 实现，在 MainActivity 中，进行注册广播：

```
//动态广播:
EventBus.getDefault().register(this);//注册
DynamicReceiver receiver=new DynamicReceiver();//
IntentFilter intentFilter=new IntentFilter();//过滤器
intentFilter.addAction(DynamicReceiver.DYNAMICACTION);//设置过滤条件
registerReceiver(receiver,intentFilter);//注册接收器
```

并重载 onDestroy 注销订阅者：

```
@Override
protected void onDestroy()
{
    super.onDestroy();
    EventBus.getDefault().unregister(this);//注销订阅者
}
```

在 GoodDetail 中，设置购物车点击时添加发送到 EventBus 的操作：

```
detail_shoppingcar.setOnClickListener((v) → {
        data_back.putSerializable("shoppingcar",goods);
        goods.setMarkShopcar();
        Toast.makeText(getApplicationContext(),"商品已添加到购物车",Toast.LENGTH_SHORT).show();
        EventBus.getDefault().post(goods);//发送该商品
});
```

在 MainActivity 中，准备订阅者：声明并注释订阅方法，在这里实现了添加该物品到购物车，并更新购物车，再发送动态广播。

```
@Subscribe(threadMode = ThreadMode.MAIN)
public void onEventMainThread(Goods event)
{
    /*添加该物品到购物车*/
    Map<String,Object> temp=new HashMap<~>();
    temp.put("goods_Firstletter",event.getGoods_firstletter());
    temp.put("goodsname",event.getName());
    temp.put("price",event.getPrice());
    temp.put("index",event.getIndex());
    shoppingcarList.add(temp);
    simpleAdapter.notifyDataSetChanged();
    /*发送广播，设置跳转*/
    Bundle sendBundleShoppingcar=new Bundle();
    sendBundleShoppingcar.putSerializable("dynamic",event);
    Intent dynamicBroadcast=new Intent(DynamicReceiver.DYNAMICACTION);
    dynamicBroadcast.putExtras(sendBundleShoppingcar);
    sendBroadcast(dynamicBroadcast);
}
```

在 DynamicBroadcast 文件中定义接收器动作：

```
public void onReceive(Context context, Intent intent) {

    manager = (NotificationManager)
            context.getSystemService(Context.NOTIFICATION_SERVICE);

    Intent intentFromReceiver=new Intent();
    if(intent.getAction().equals(DYNAMICACTION))
    {
        temp=(Goods)  intent.getExtras().get("dynamic");//获取动态广播的物品

        intentFromReceiver.setClass(context,MainActivity.class);//设置跳转界面
        //intentFromReceiver.putExtra("toShoppingCar",true);
        mPenddingIntent= PendingIntent.getActivity(context,0,intentFromReceiver,0);
        remoteViews=new RemoteViews(context.getPackageName(),R.layout.layout_notification);
        //绑定remoteViews
        remoteViews.setImageViewResource(R.id.notification_icon,temp.getImageid());
        remoteViews.setTextViewText(R.id.notification_title,"前往购物车吧~");
        remoteViews.setTextViewText(R.id.content1,"马上下单");
        remoteViews.setTextViewText(R.id.content2,temp.getName()+"已添加到购物车");

        Notification.Builder builder=new Notification.Builder(context);
        builder.setContentIntent(mPenddingIntent)
                .setSmallIcon(temp.getImageid())
                .setContentText("·")
                .setAutoCancel(true);
        Notification notification=builder.build();
        notification.bigContentView=remoteViews;
        manager.notify(num++,notification);
    }
```

然后再利用 onNewIntent 来对动态广播发送的 intent 进行显示购物车操作：

```
@Override
protected void onNewIntent(Intent intent)//设置显示购物车界面
{
    super.onNewIntent(intent);
    setIntent(intent);
    mListView.setVisibility(View.VISIBLE);
    mRecyclerView.setVisibility(View.INVISIBLE);
    floatingActionButton.setImageResource(R.mipmap.shoplist);
    exchange=true;
}
```

（3） 实验遇到困难以及解决思路

一开始的时候并没有使用 RemoteView，直接使用了自带的方式使用 set 的方式来设置，这样导致了两个结果，一个是不太好看，另一个就是一开始在不知情的情况下没有设置 SmallIcon，然后导致程序直接崩溃，后面在查看了

Monitor 里的信息之后才发现是一定要设置 SmallIcon 的。

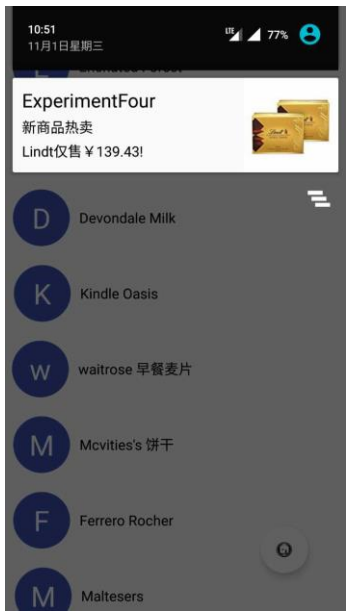
刚启动的时候，弹出了一条通知之后，点进去发现无论显示的是哪个物品的通知，点进去的详情都是同一个，后面发现是需要在 mPenddingIntent= PendingIntent.getActivity(context,0,intentFromReceiver,PendingIntent.FLAG_UPDATE_CURRENT);将第四个参数改成 FLAG_UPDATE_CURRENT。

而还有就是在点击购物车进入购物车界面的时候，并没有跳转至购物车，而是跳转到了商品列表界面，后面发现原因是因为在接收到回来的动态广播发送的 Intent 并不是从 onCreate 开始执行，所以这时候借助了 onNewIntent 来实现对于动态广播的传过来的 intent 显示购物车列表。

后面又出现了在购物车列表并没有显示刚加入的物品，后面通过在 onEventMainThread 中设置购物车的更新即可。

四、 课后实验结果

使用了 RemoetView 实现通知栏，而不是系统自带的功能实现。



一开始使用 RemoteView 设置 XML 布局的时候，我使用了比较新的 Constrain 布局，然后后面程序老是崩溃，后面看了 Monitor 发现好像不能用这个，改成了 Relative 布局才得以实现。

然后就是在使用的时候，主要采用两个函数 public void setTextViewText(int viewId, CharSequence text)和 setImageViewResource(int viewId, int srcId)实现即

可。

五、 实验思考及感想

这次实验我是在上次实验 3 的基础上继续进行实验，在保证实验 3 的前提下，完成这次实验。主要是通过添加两个广播来实现两个功能。总体上难度并没有上次的大，但是十分容易出现 BUG，还有就是使用 EventBus 来帮助实现传递消息，“EventBus 是一款针对 Android 优化的发布/订阅事件总线。主要功能是替代 Intent,Handler,BroadCast 在 Fragment，Activity，Service，线程之间传递消息.优点是开销小，代码更优雅。以及将发送者和接收者解耦。”。