|  |  |
| --- | --- |
| **成**  **绩** |  |

Android案例开发 **实验报告**

**实验名称 实验三 修改和删除我的支出数据并显示统计信息的设计与调试**

**专业班级 软件172 班级学号 201715030208**

**姓 名 朱洪龙 实验日期 2019/03/29**

**（报告内容包括：实验题目、实验环境、实验步骤、实验结果、遇到的问题和解决办法、实验小结等。**

# 【实验题目】

1. 可以先实现我的支出中ListView内所有item数据的点击事件。
2. 设计一个Activity，用来显示我的支出和收入数据（可以先判断传过来的数据类型再来显示），之后实现其修改和删除按钮的点击事件即可。
3. 实现数据管理，其中设置按钮点击事件来可视化支出和收入（通过统计图），余下的空间默认显示便签信息。
4. 剩下的系统管理（设置登录密码）、收支便签（添加一条文字信息）、帮助（显示一个HTML5的页面信息）和退出（一个finish()即可）都比较简单。

# 【实验环境】

**OS:Windows 10**

**IDE:Android Studio 2.3.2**

**AVD:Nexus 5X API 23**

**minSdkVersion:21(android 5.0)**

**targetSdkVersion:23(android 6.0)**

# 【实验步骤、程序调试过程中所有遇到的问题和解决办法】

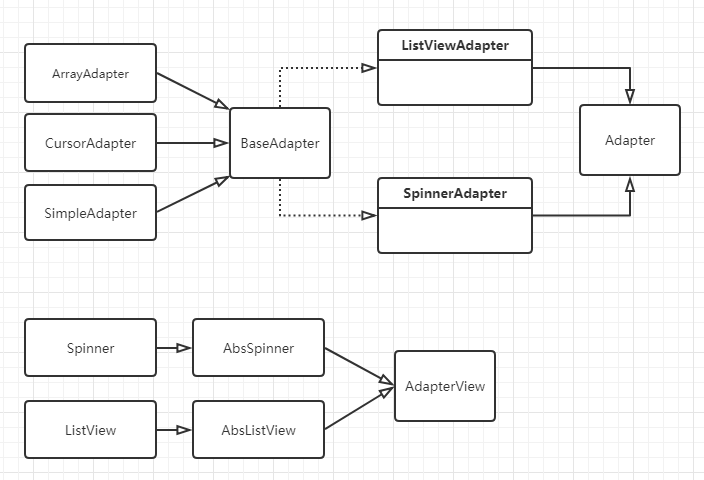
## 设置点击事件：

在收入信息中对显示的ListView设置setOnItemClickListener()点击事件，在方法onItemClick()中,先将该支出/收入信息中的编号截取出来（后面得通过它来查找到对应的数据并进行显示以供用户修改或删除）,然后通过putExtra()方法将其与此操作的数据类型（支出还是收入）等信息一并带到另一个即将启动的Activity中。这里唯一需要注意的地方就是putExtra(String name, String value[])方法中的name参数，这个参数设为静态常量(Flag)，通过文档可知，后面的Activity若要获取刚才传入的数据的话，就得通过prefix（前缀）来调用这个name。（例如：Outaccountinfo.Flag）。

## 创建InfoManage(继承自Activity)并设计其布局文件：

布局文件跟新增支出/收入类似（在支出和收入布局TextView中不同的地方有标题和地点/付款方,这些地方可以不设置，因为在InfoManage中会根据传过来的类型进行设置），不在赘述。在InfoManage中，获取9个组件（相同的有7个，还有两个是地点/付款方），然后获取传过来的数据，用getStringArray()获取。接着判断支出/收入，对应其类型设置其特殊的TextView，然后根据传过来的id值调用DAO数据库访问对象中的find()方法返回一个收入/支出模板对象，将所有的数据设置到EditText中，难点在于设置其中的一个下拉列表中的数据。因为收入和支出的类型不同，所以得先判断后再设置下拉列表中的array数据。根据判断结果，通过ArrayAdapter调用createFromResource()返回一个字符队列适配器，其所需参数为Context context(this即可)、int textArrayResId(在value目录中定义的字符资源中设置的id(其通过转换为R类中的int型来进行索引))、int textViewId(设置显示数据的视图类型)，最后再通过Spinner对象调用setAdapter()来设置。接着通过setPrompt()设置其显示的类型（setPrompt()中的参数为CharSequence，是一个接口，而模板返回的是一个String类型，这是典型的接口回调技术，由于String类实现了这个CharSequence接口中的方法），最后就是实现修改和删除按钮的点击事件了。两个按钮都得判断收入/支出类型，修改就是建立个数据模板，将编辑框中的数据设置到模板中（别忘了模板中还有个id），然后调用DAO中的update方法并弹出提示信息即可（Spinner中的数据通过getSelectedItem()方法获得）；删除就更简洁了，有了id值就可直接判断收支类型，调用DAO中的delete方法即可。

拓展：



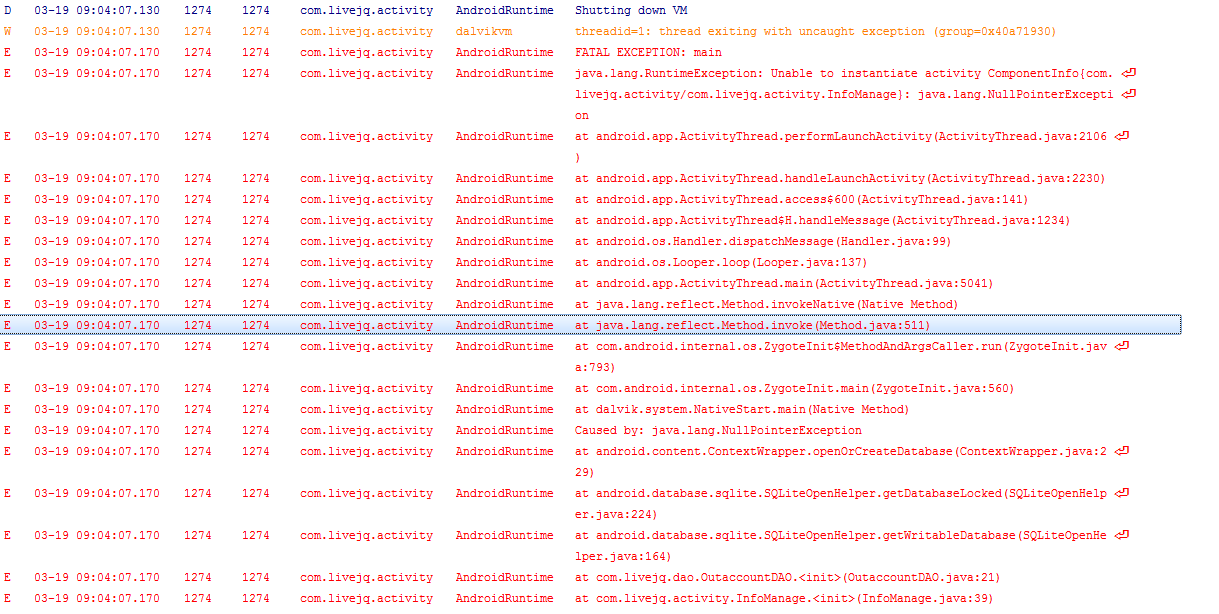
## 实现数据管理

新建一个继承自Activity的Showinfo类，其中在顶部水平设置三个按钮，通过点击分别可以显示收入、支出和便签信息，其中收入和支出通过一个intent进入另一个Activity进行直方图的显示；而便签则为默认的信息显示在下方空白处（通过一个ListView组件）。三个按钮代表三个不同的信息类型，所以可以将显示信息作为一个方法封装起来作为ShowInfo(int intType)方法使用，只需传入一个资源参数。在ShowInfo方法中，通过Switch来对传入的类型分别进行处理。收入/支出类型时，建立Intent对象并传入此类型（收入/支出）；若为便签，则通过DAO对象查找所有的数据返回一个List集合，然后遍历这个集合，格式化每条数据，然后新建适配器并进行设置。对便签中的数据项设置点击事件，将此便签的id通个intent传递到FlagManage中，FlagManage与InfoManage类似。

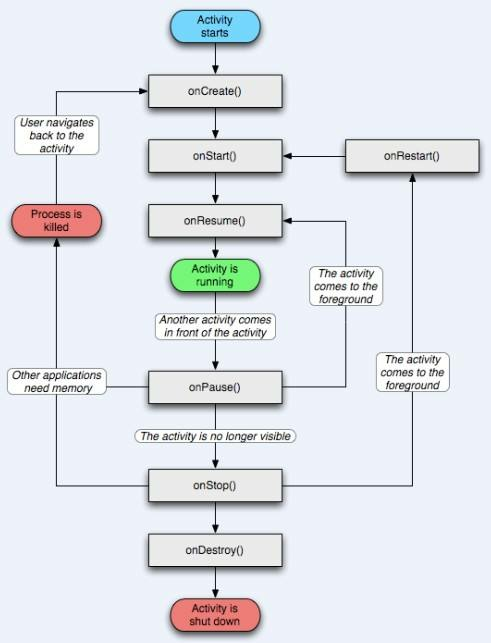
这里的难点主要是设计直方图并将数据显示出来，在eclipse中设计好的直方图搬到AS这里来显示时，效果差强人意（同比缩小了好几倍，文字略显模糊，难道是API版本问题？还是IDE问题？），因此需要修改数据，但是这些数据基本都是有联系的，一个一个修改太麻烦了，所以得找到这些数据最终的依赖项，没错，就是横竖的坐标轴的数据。在model包中创建一个Point类来描述一个点的位置，然后在图表中将表示横竖轴的起点和终点都换成Point来表述，由此，其它可依靠此数据进行适量增减常量即可，一些像字体大小、画笔大小等设为常量，修改因此更加简便。

直方图显示支出或收入信息，因此这里有两个判断的地方，一个是取到需要显示在横轴下的一组收入或支出的类型，另一个是通过DAO取出对应的的数据。整个过程就是在Activity主视图上显示一个自定义的View，因此可以创建一个内部View类，然后在onCreate()中通过addView（View child）方法加入进来即可。建立一个MyView，调用View默认的构造方法，重写onDrag()方法。在onDrag()中调用默认的已写好的onDrag(Canvas canvas)，其已为我们初始化好了画板，我们只需通过画笔渲染即可。创建画笔，设置笔触宽度，抗锯齿，笔触颜色，然后画出横竖坐标轴；难点在柱形图这里，首先设置画笔为填充样式，首先获得所有金额数据（一个浮点型数组对象），然后获得其中的最大值。因为柱形图中代表收支类型的数据的高度都得有个参照物，也就是设置个比例。至于为何不是取最小值，显然，这个柱形图得约束不得超过横竖坐标轴的表示范围，所以得用其中的最大值，将其约束在范围内。至于这个约束方法，得先获得x、y轴所能表示的范围的数值（x只需获得第一个柱形图的起点x位置，y只需设置其终点，即不能超过的位置即可，然后设置一个height，用来表示所能表示的柱形图的最适高度（这个高度为其它柱形的参照（实体），而最大金额是其它金额的参照（抽象））），因此，其中的难点主要在于如何将实体与抽象联系起来。画柱形图用到drawRect()方法其中的参数为起点x，起点y，对角x，对角y（根据对角线确立一个矩形）。上面已经有了起点x，只需按照每个柱形图的位置增加一定的间隔和i个柱形图宽度即可，主要是这个起点y的确立。实体与抽象的联系，y-height/max\*money[i]，这个数组中的最大值就是max，即整个最大值为y-height，这个就是最高的柱形图的起点y值了。

# 【实验结果】

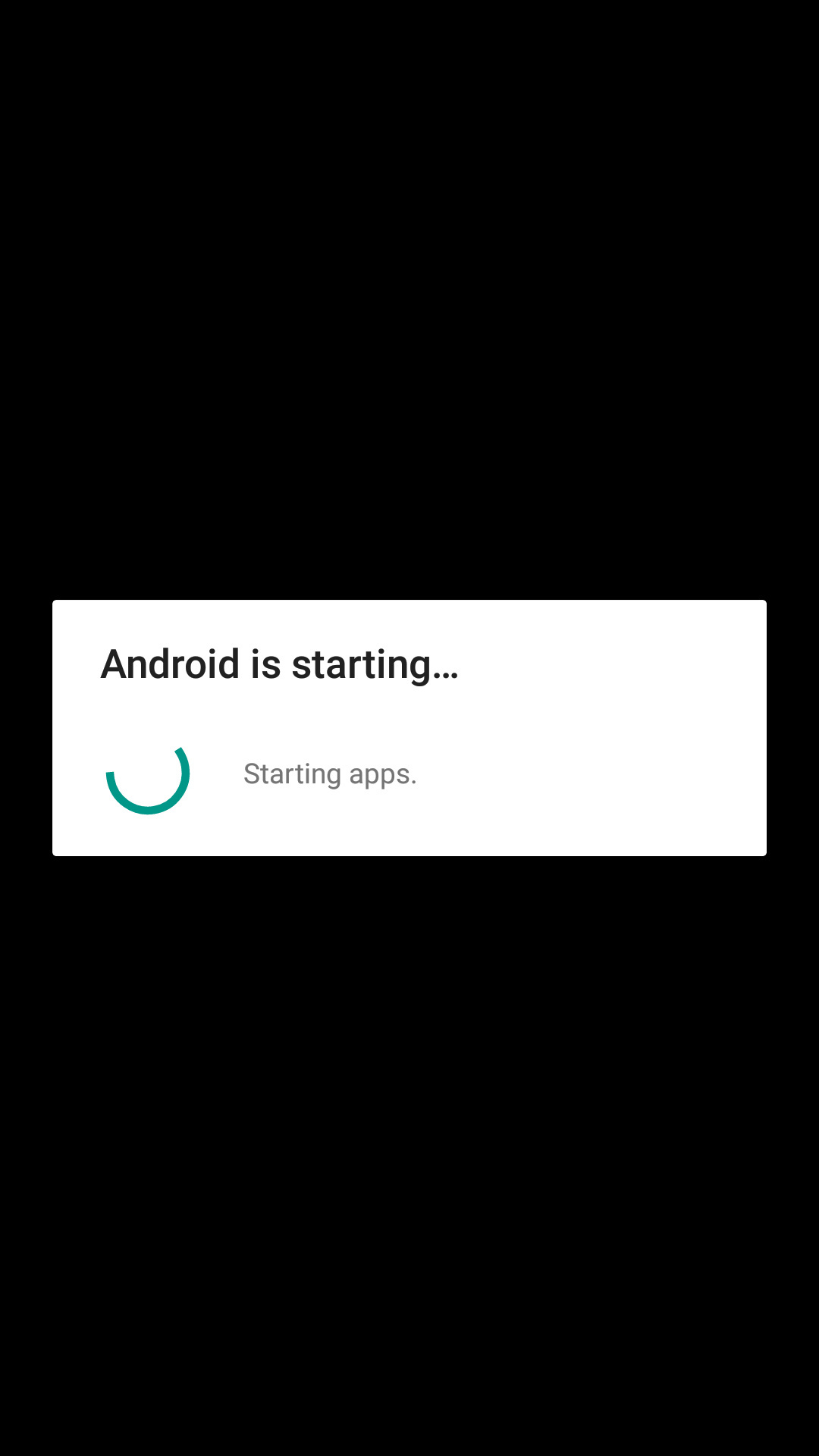


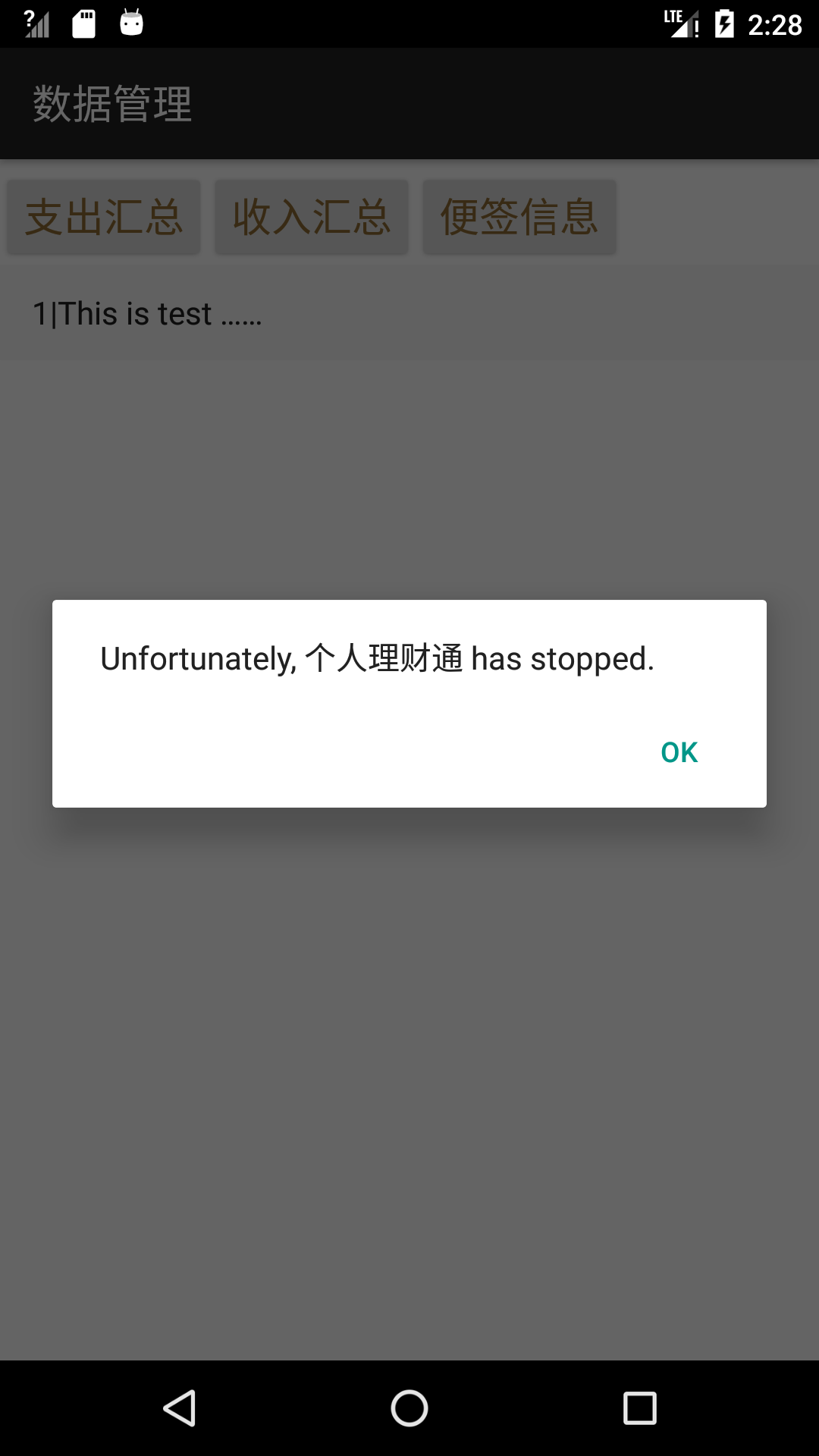
在我的支出中点击其中的数据项时跳转到InfoManage中显示详细信息，在InfoManage中发生了如上的空指针错误，经老师提醒，仔细查看后发现：InfoManage调用DAO时，是在声明属性的地方直接实例化一个对象，然后返回作为其中一个属性来使用。但是Activity的机制并非想象的如此执行，它是先调用onCreate()方法来初始化,而属性是后面才进行初始化和空间的分配的，然而在onCreate()中却调用了还没实例化的空对象（对于InfoManage是属性）所引起的空指针异常的情况：

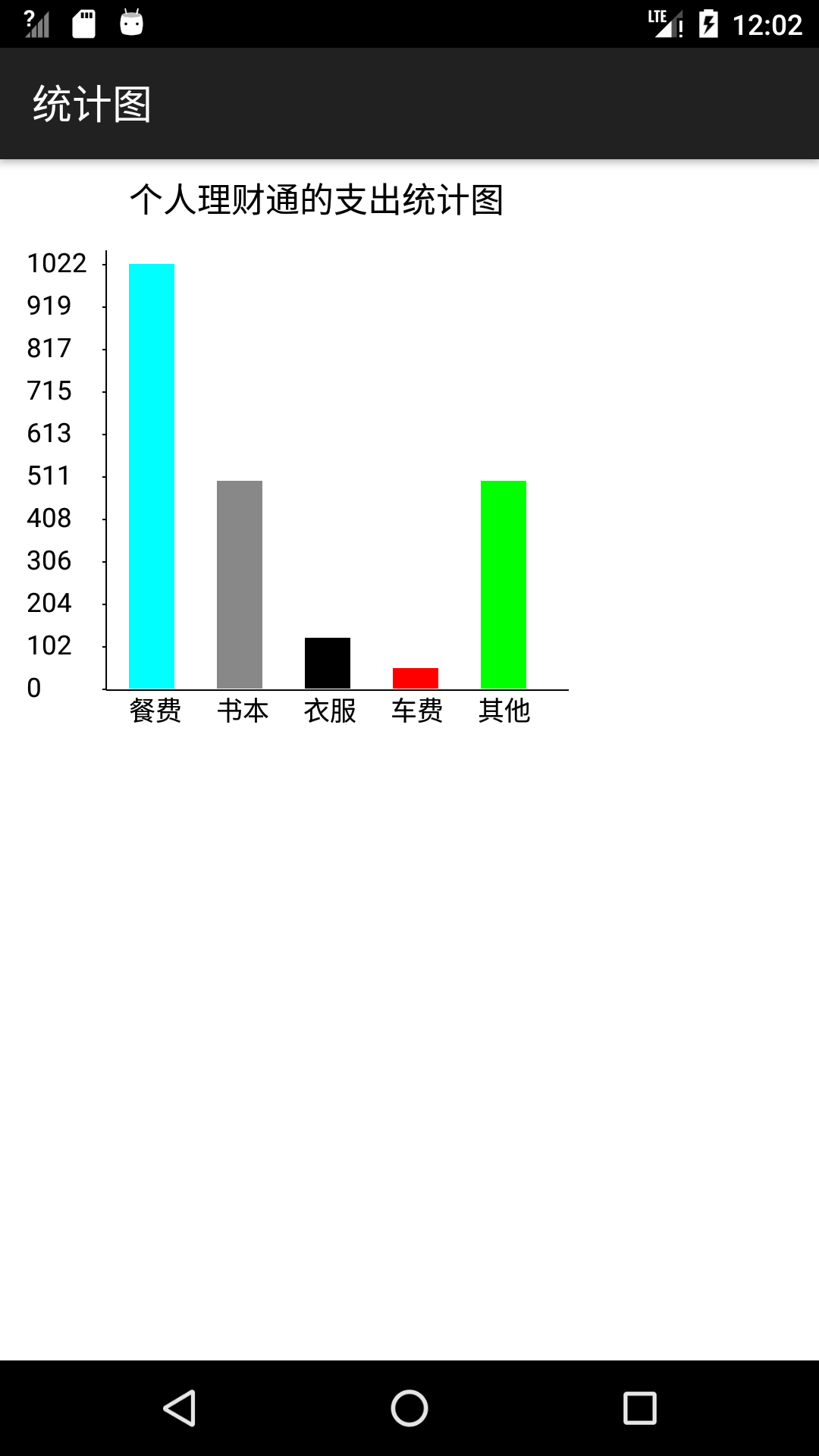
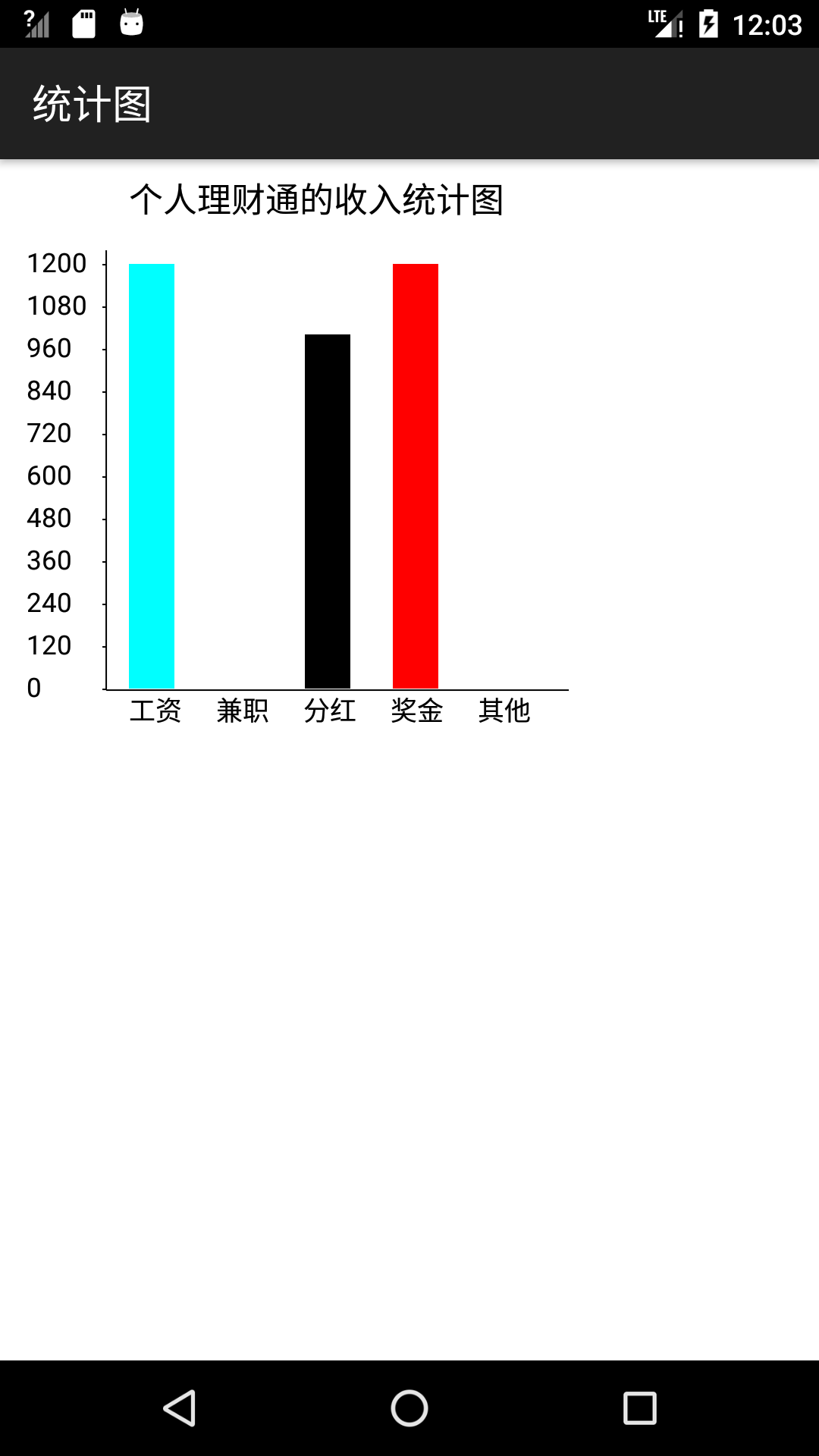
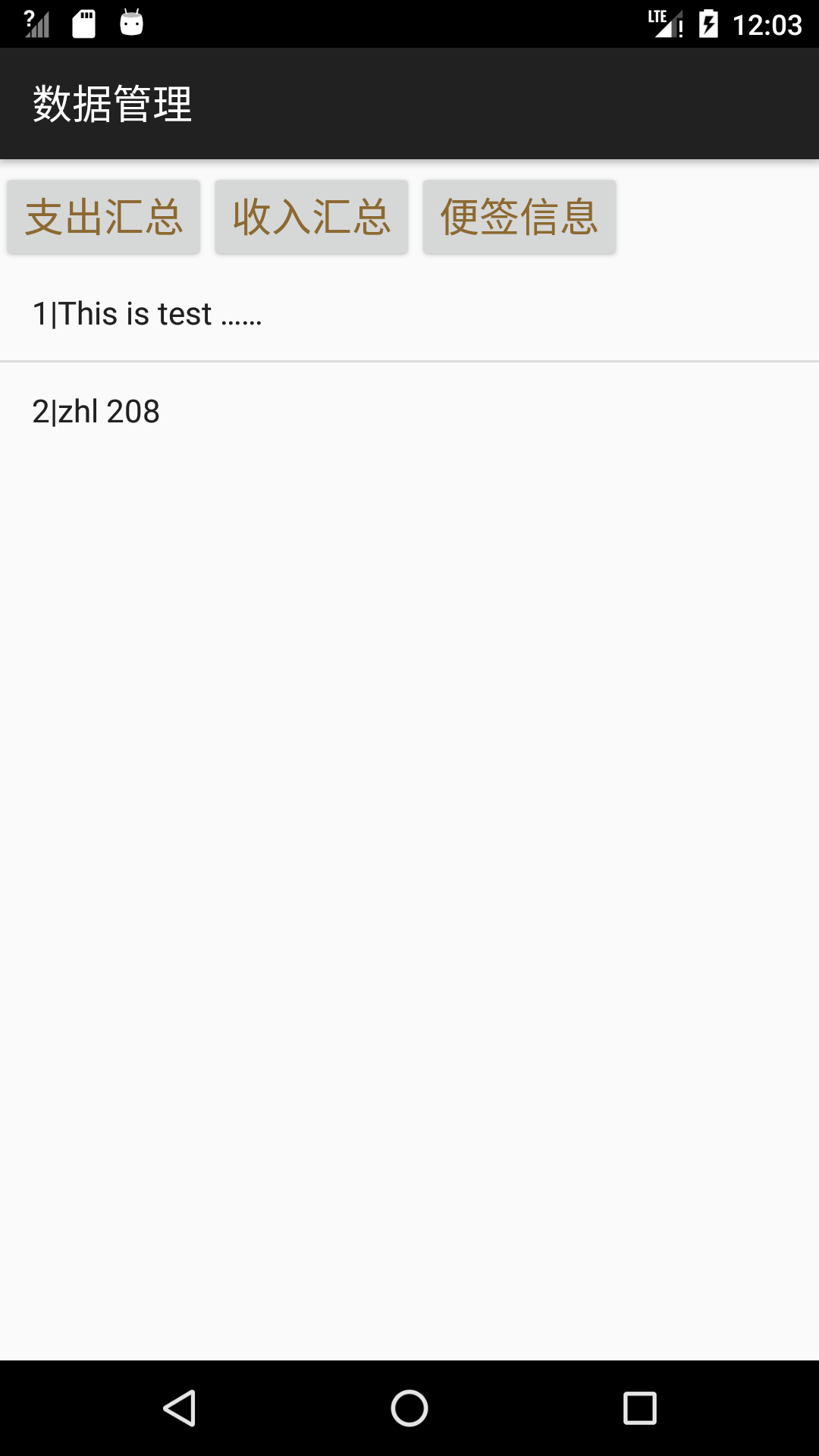


解决办法很简单，只需直接声明两个支出/收入DAO对象为null，然后在onCreate()中进行实例化即可。

注意：将所有Activity在AndroidManifest.xml中声明（常漏），不然出现如下：







# 【实验小结】

现在开发的app（android 6.0）安装到我们现在的手机出现不适配问题（全面屏android 8.1.0），下面出现一部分“黑屏”现象，如上,可能跟API版本有关，有待探究….