目录

[广州航海学院 1](#_Toc3111018)

[【实验题目】 1](#_Toc3111019)

[【实验环境】 1](#_Toc3111020)

[【实验步骤、程序调试过程中所有遇到的问题和解决办法】 2](#_Toc3111021)

[建立项目包结构： 2](#_Toc3111022)

[创建登录活动窗口和布局文件 2](#_Toc3111023)

[创建保存密码的模板(bean) 2](#_Toc3111024)

[创建数据库连接对象（DAO） 2](#_Toc3111025)

[创建操作数据库密码表对象（DAO） 2](#_Toc3111026)

[修改AndroidManifest.xml文件并运行 2](#_Toc3111027)

[【实验结果】 2](#_Toc3111028)

[【实验小结】 3](#_Toc3111029)

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

Android案例开发 **实验报告**

**实验名称 实验一 公共类和登录模块的设计与调试**

**专业班级 软件172 班级学号 201715030208**

**姓 名 朱洪龙 实验日期 2019/03/09**

**（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. 创建个人理财通项目
2. 输入公共类代码
3. 创建登录窗口的布局文件，输入登录模块的Activity类代码，调通登录功能。

# 【实验环境】

**OS:Windows 10**

**IDE:Android Studio 2.3.2**

**AVD:Nexus 5X API 23**

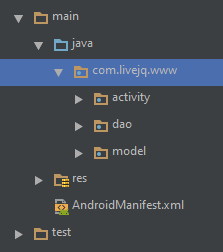
**minSdkVersion:21(android 5.0)**

**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】

***(在这里不贴代码,只是详细分析)***

## 建立项目包结构：



Activity:活动窗体文件

Dao:操作数据库文件

Model:保存数据文件

## 创建保存密码的模板(bean)

在model中创建Tb\_pwd类文件，设置密码文本，自动获得get,set方法

## 创建数据库连接对象（DAO）

主要是继承SqliteOpenHelper和在构造方法中调用父类的构造方法,然后重写onCreate()和onUpgrade方法，前者创建（若数据库中没有）或加载数据表，后者可写可不写.

## 创建操作数据库密码表对象（DAO）

在dao中创建PwdDAO类文件，其中主要是在其构造方法中调用数据库连接类并获得可操作数据库的对象（用来执行Sql语句）,其后的方法都是基于Sql语句的增删改查（获得密码并通过密码模板保存和return出来的方法和查询数据表中是否存在密码的方法等）

## 创建登录布局文件和活动窗口

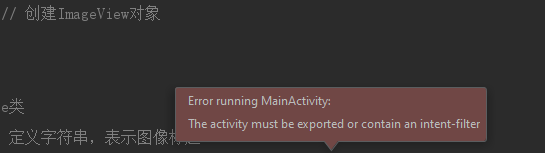
在res目录中的layout目录中，创建login.xml布局文件，其整体布局为RelativeLayout相对布局，然后是一个TextView（输入密码文本）和一个EditText(hint:输入密码框,layout\_below:在输入密码文本下面)，然后是两个登录和取消Button，定位都差不多（先让其中一个在输入框的下面，然后让另一个Button在这个Button的旁边）。

在activity中创建Login类文件，并设置其R.layout.login，之后获取输入密码框和按钮并对两个按钮添加点击事件，获取密码框中的数据对其判断，这里就要用到密码操作对象里的方法，主要判断用户是否为新用户，若为新用户，则判断没有输入密码时登录到主窗体；若已有用户，则判断输入的密码是否正确，判断为正确时登录到主窗体，点击取消时调用finish（）方法关闭窗体

## 修改AndroidManifest.xml文件并调试运行

设置Login窗体为启动时首先显示的窗体，还得声明MainActivity为一个Activity。

第一次运行：

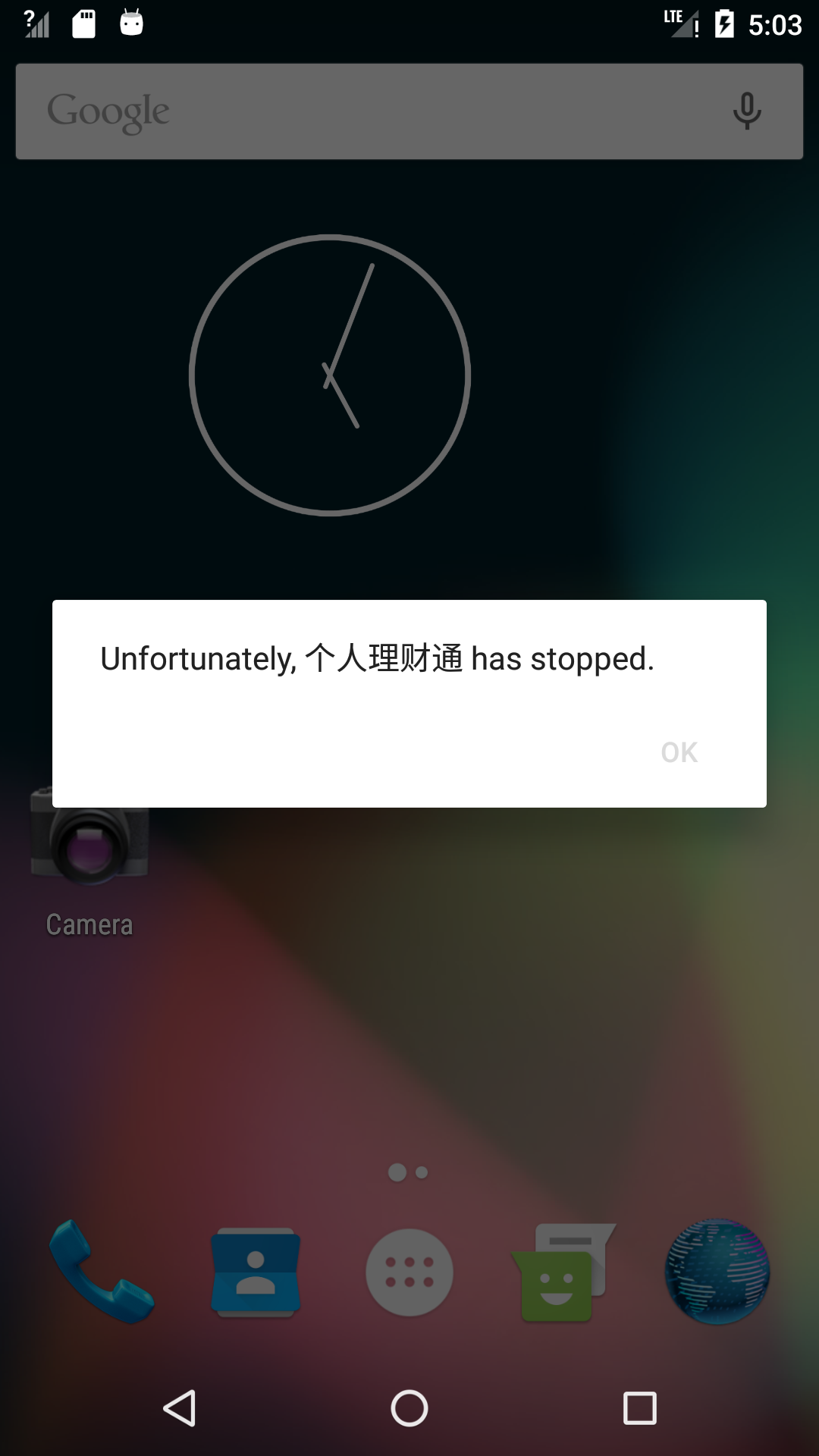


修正之后：



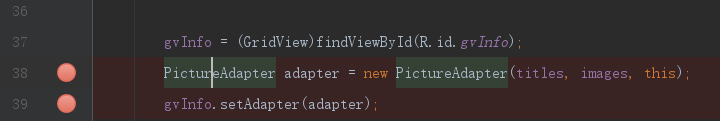
第二次运行：

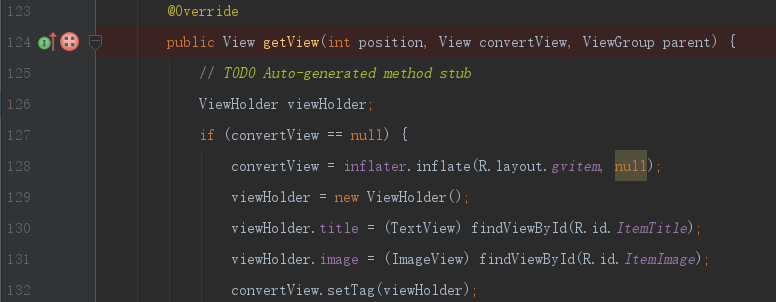


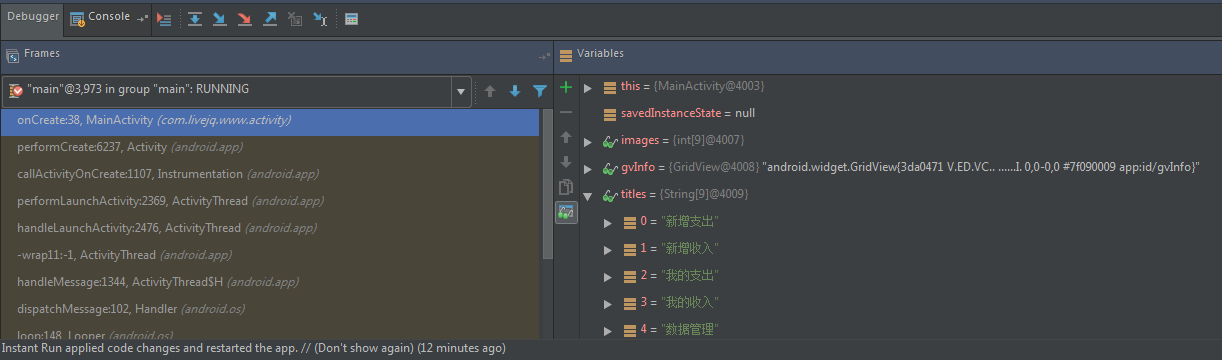


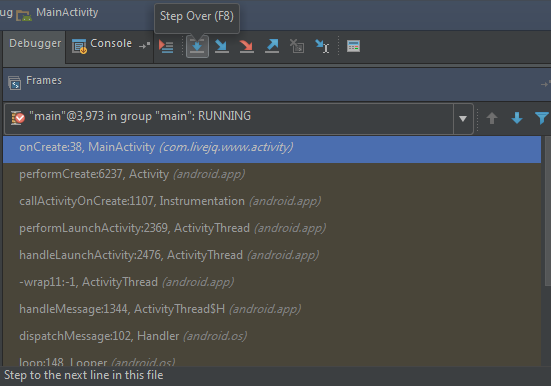
初步断定为MainActivity中的错误（因为Login中就添加了两个点击事件，还没点击就不行了）

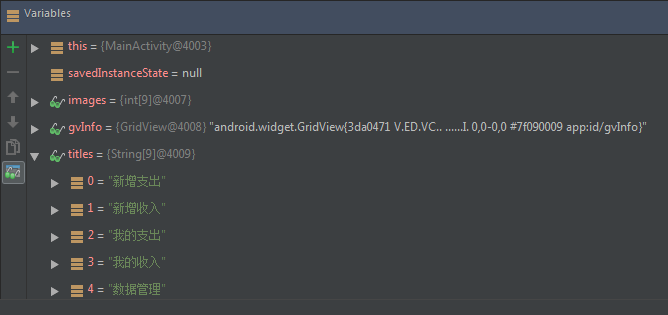
设置断点：



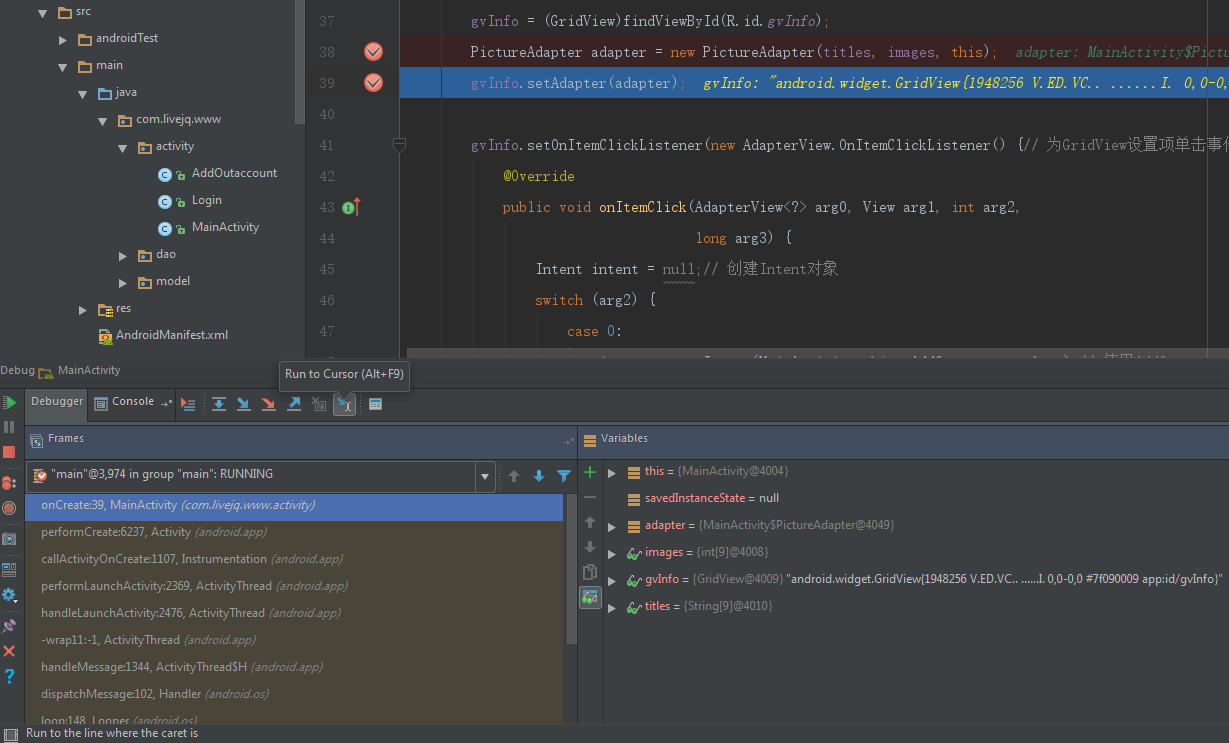




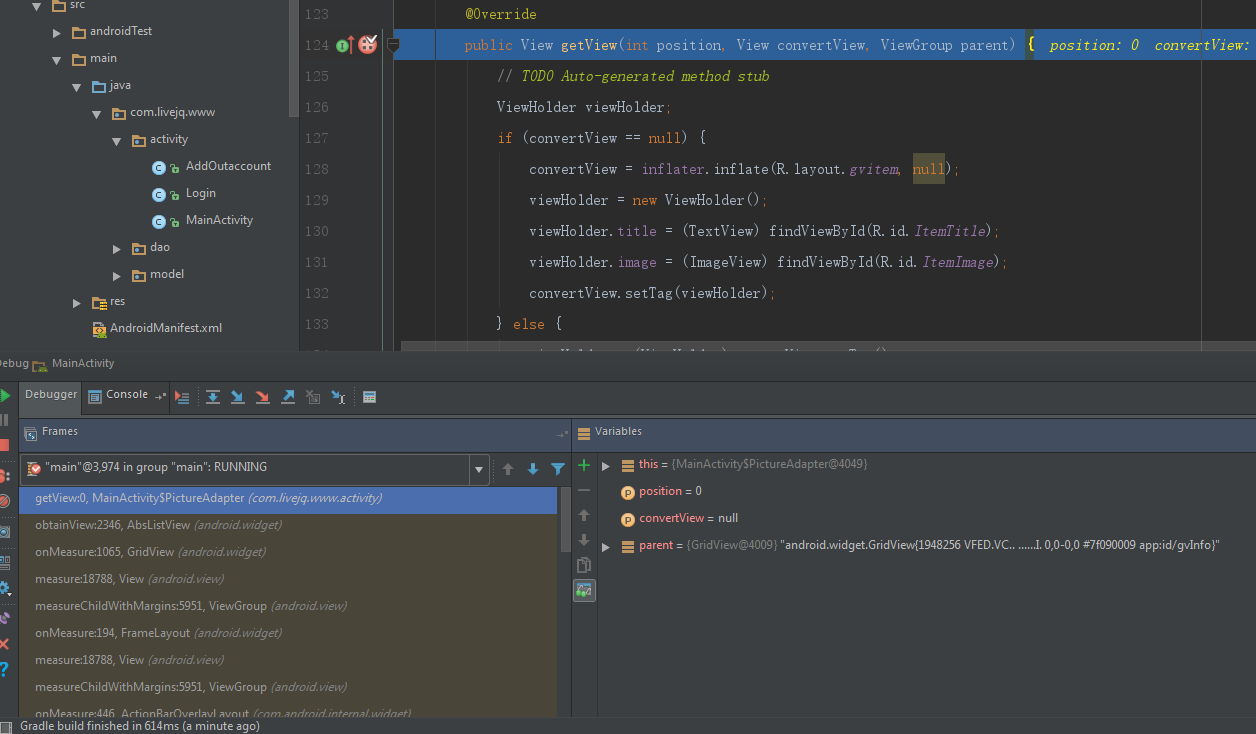




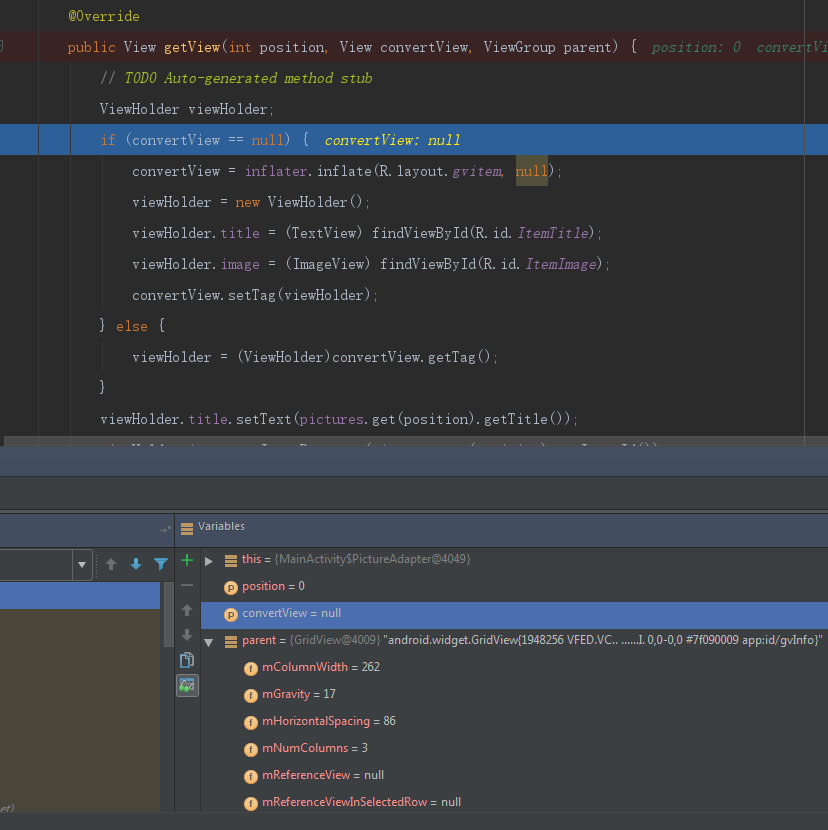
Step over后：

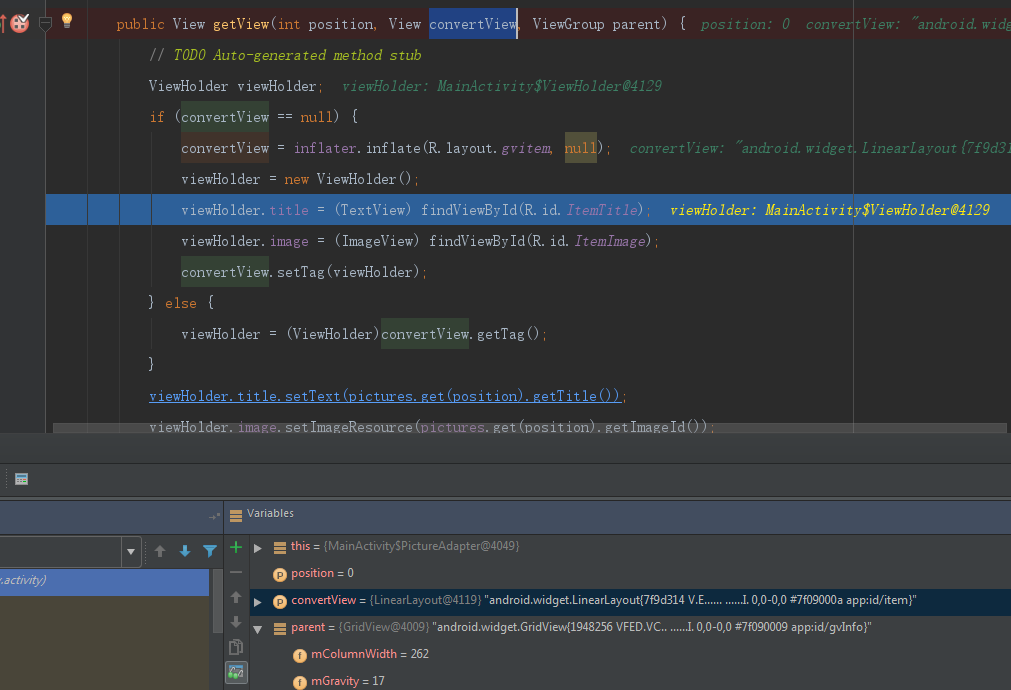


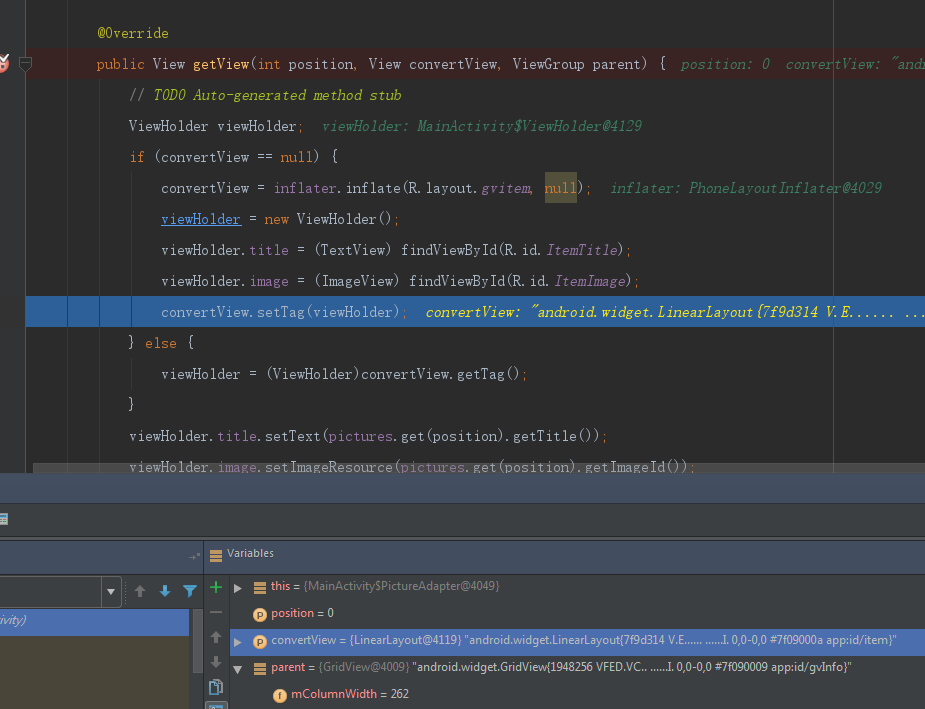
Run to cursor后：

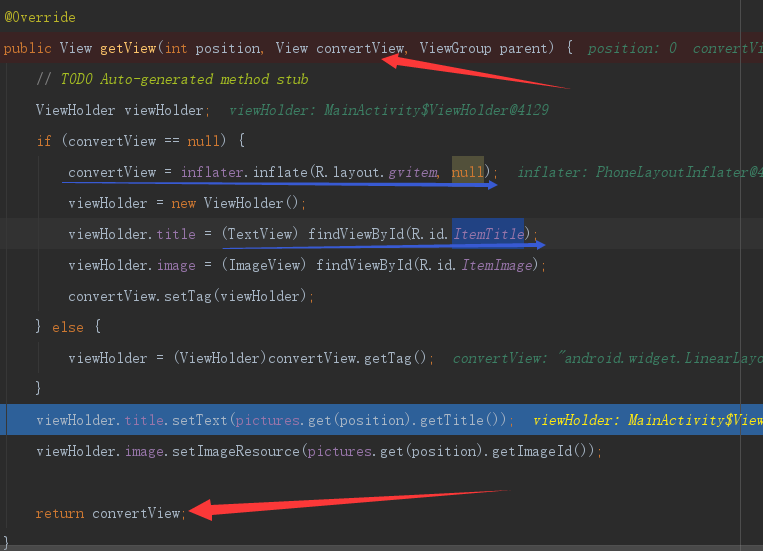


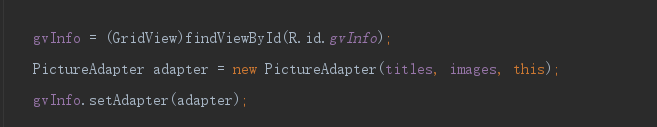
Step over后：



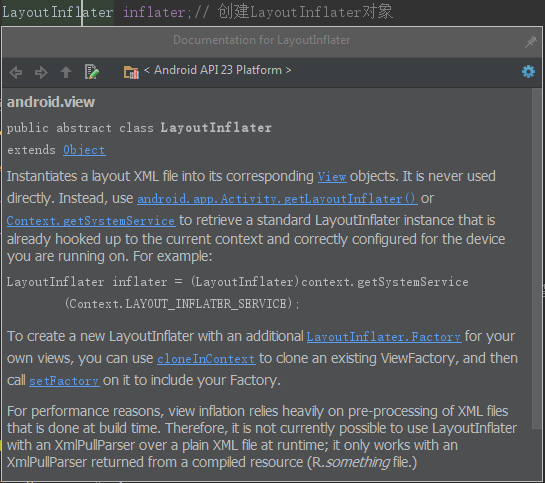






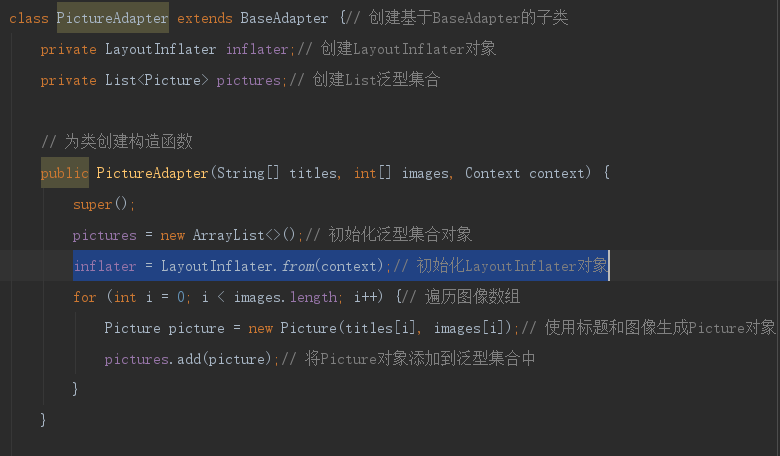


在重写的方法getView()中，ConvertView是这个这个适配器最终返回的一个视图，而之后将这个适配器设置到gvInfo中。然而在getView()中获取视图中保存的数据类型（标题和图片）时，是以this获取的（未进行特殊处理），而gvInfo（可以认为是用来装载另一个视图中的数据）也是this获取的；为此这两个视图之间无法取得联系（数据通信），先来看看LayoutInflater的作用。

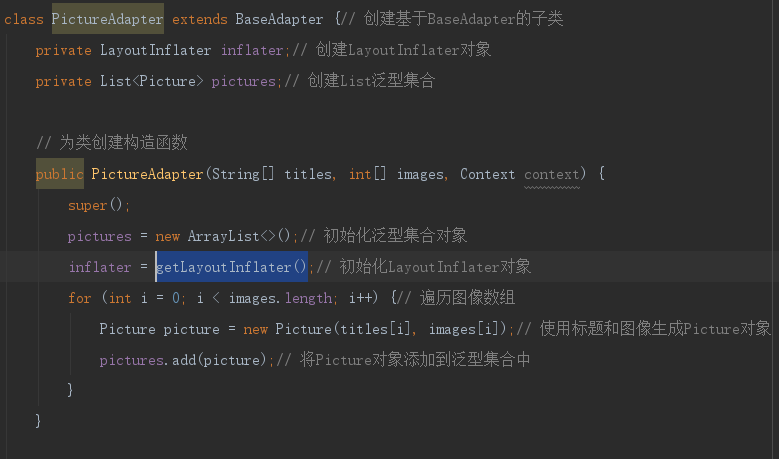


大概意思就是说可以初始化一个xml的布局文件来响应或与之通信的一个视图对象（不能直接使用），但可以通过已挂钩的当前上下文来获取。

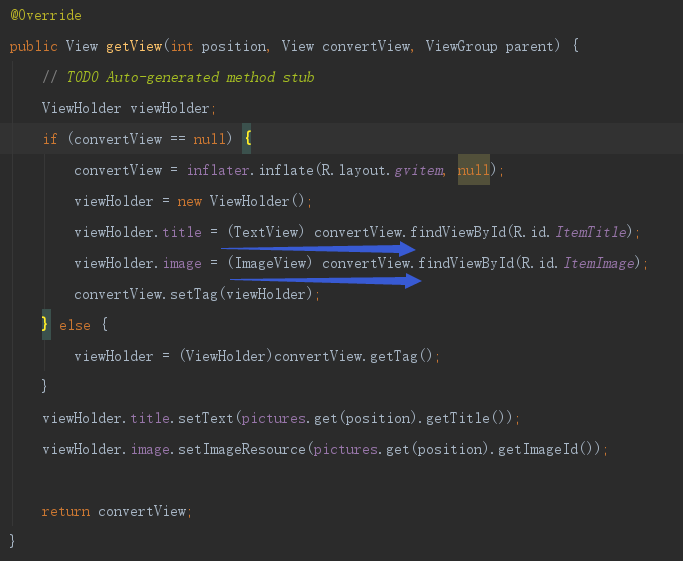
So,



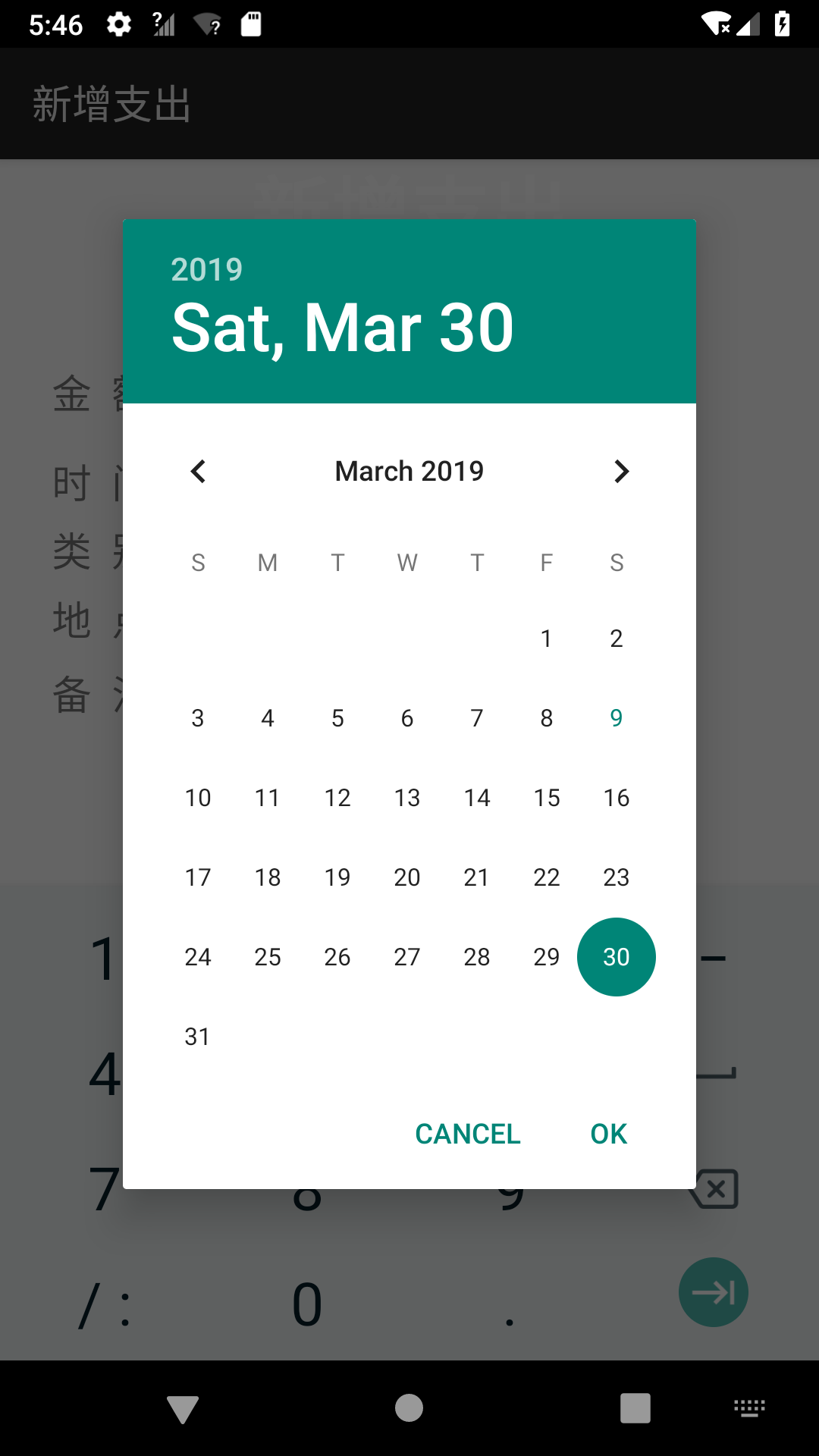
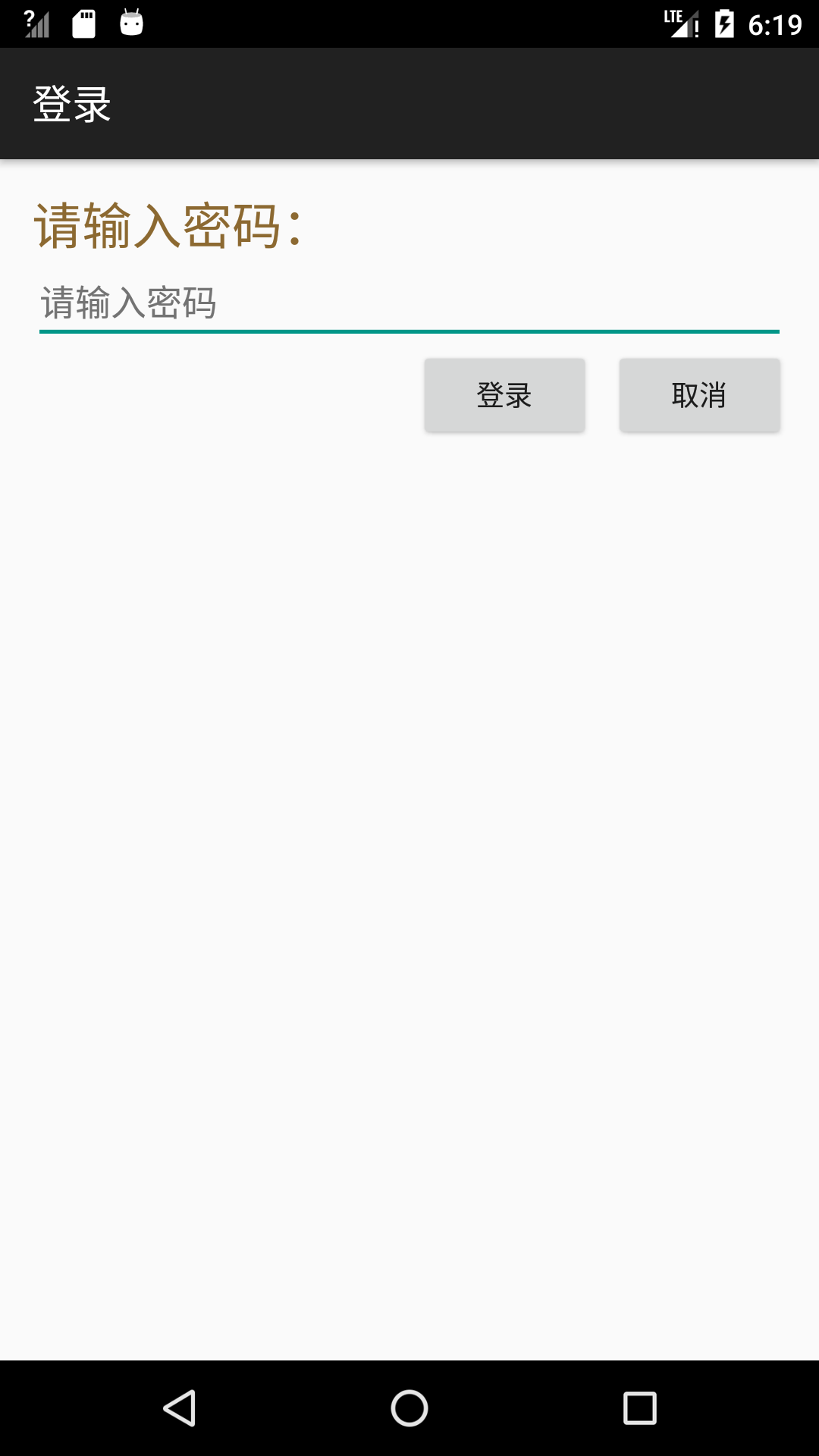
也可以：

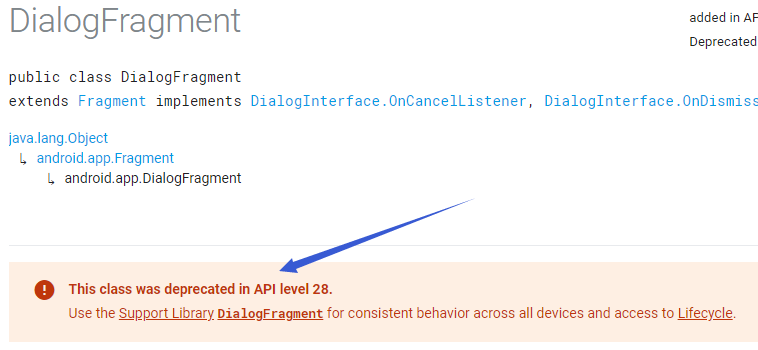


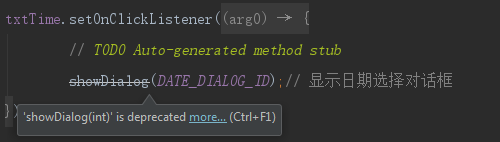
修正后（用LayoutInflater处理过的视图对象来加载需要作为适配器的视图（图片和标题））：



# 【实验结果】









# 【实验小结】

对于调试还不太熟，日后还得多加操作。在处理新增支出时，发现在showDialog()方法和Activity中的onCreateDialog()方法已经弃用了，查了官方文档说明是改用了继承自Fragment的DialogFragment（而在这篇文档顶端却说了此API在android API 28已弃用）,哎~。

将在下篇报告中详谈。