**说明文档**

**应用名称：超级日记**

**制作动机**

**Q：我们经常觉得生活太匆忙，没有条理？**

A：记录日记是一个很好的选择，帮你梳理每天的生活

**Q：每天手写日记，太麻烦？有时想要随身记？**

A：手机记日记，方便又轻松！

**Q：想使用一款APP自动记录一天的生活状态，获得每日成就感？**

A：超级日记来帮你！

**功能介绍**

超级日记是一款能够**自动记录**您一天的生活状态，并与您自己记录的文字和图片一起，**自动生成**相应日记的一款应用。通过每日浏览超级日记，可以更方便地了解自己一天的生活状态，了解自己都做了什么，从而获得每日成就感，也能使自己的生活更有条理。

具体功能有：

1. 自动记录今日起床时间、今日睡觉时间、今日手机使用次数、今日联系最多的人、今日心情、今日最长时间的逗留地点等；

2. 可以输入并保存每一天自己想记录的文字；

3. 可以记录图片，支持拍照获取图片和从本地图库获取图片，并且图片支持裁剪和记录选中的部分；拍照获取的图片可保存到本地；

4. 私密日记的密码保护功能：可单独为每一天的日记设置不同的密码，日记加密后，日记中输入的文字和图片必须输入密码才能打开查看；设置的密码可以取消；

5. 可以查看历史日记。

**运行界面**

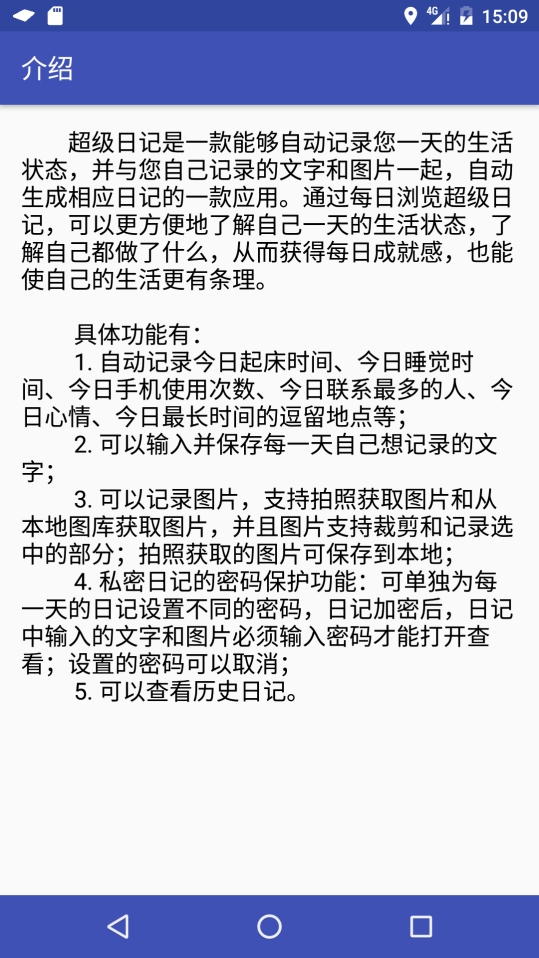
日记一览界面：

历史日记界面： 侧滑菜单：

介绍： 关于：

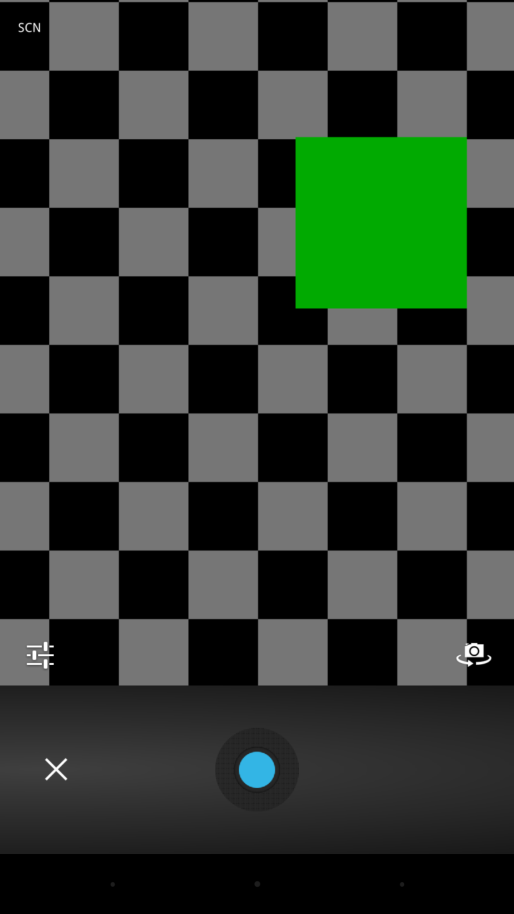
 

动画按钮菜单： 文字记录界面：

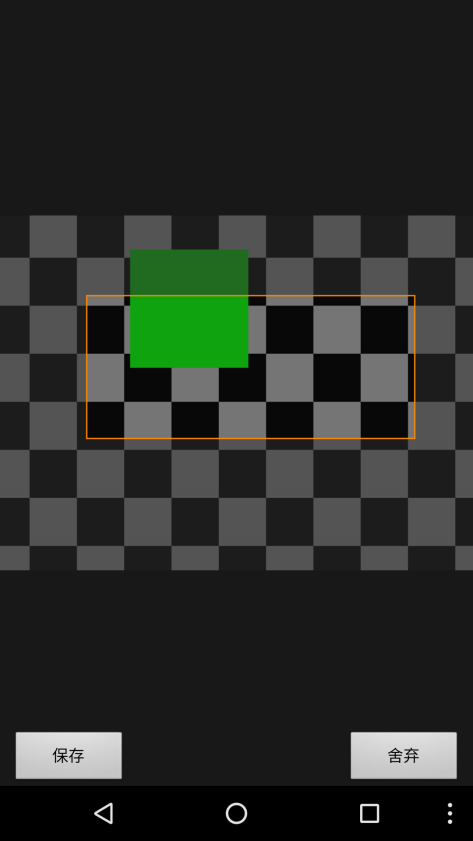
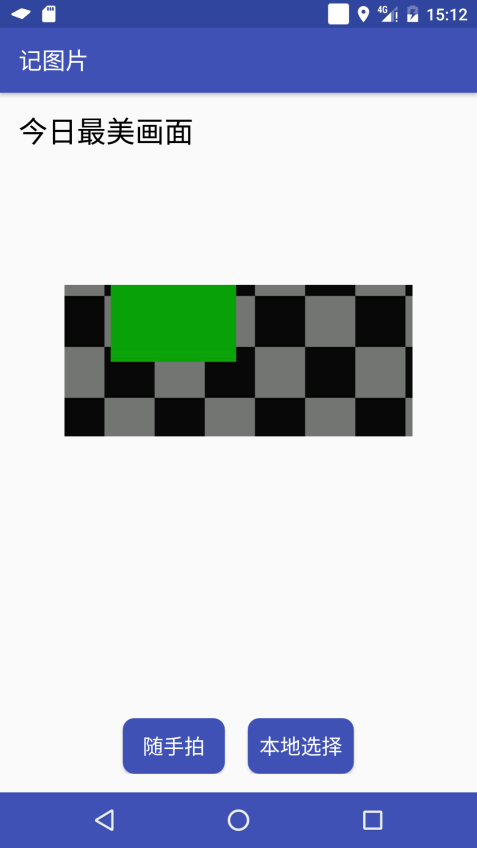
 

图片记录演示：

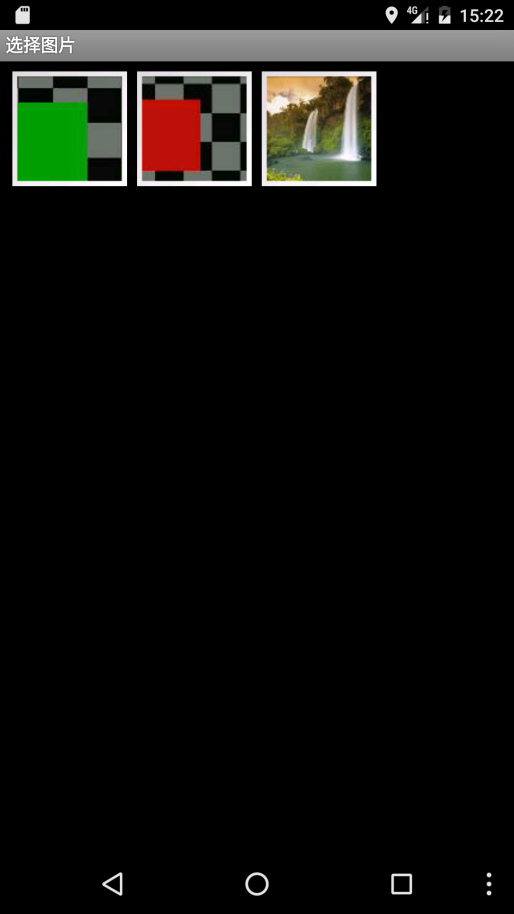
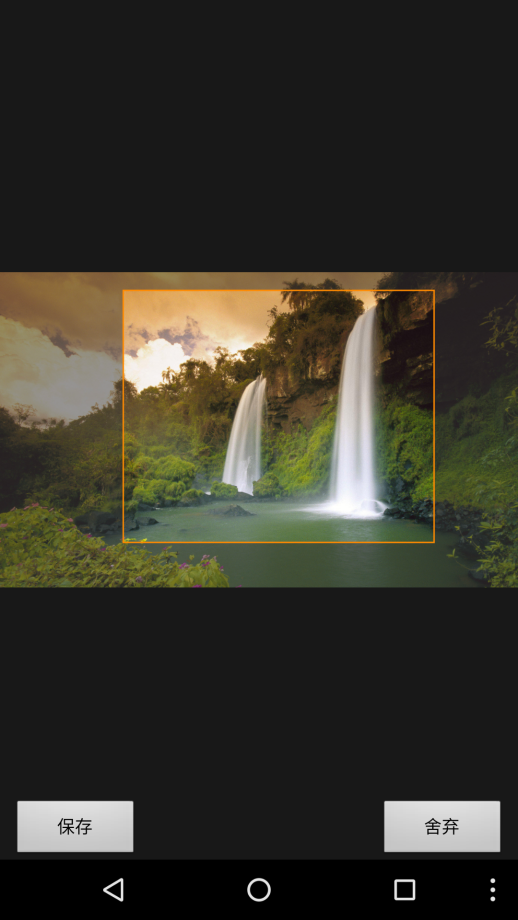
“随手拍”功能：点击随手拍，进入拍照界面（图片为模拟器自带的模拟拍照场景）：

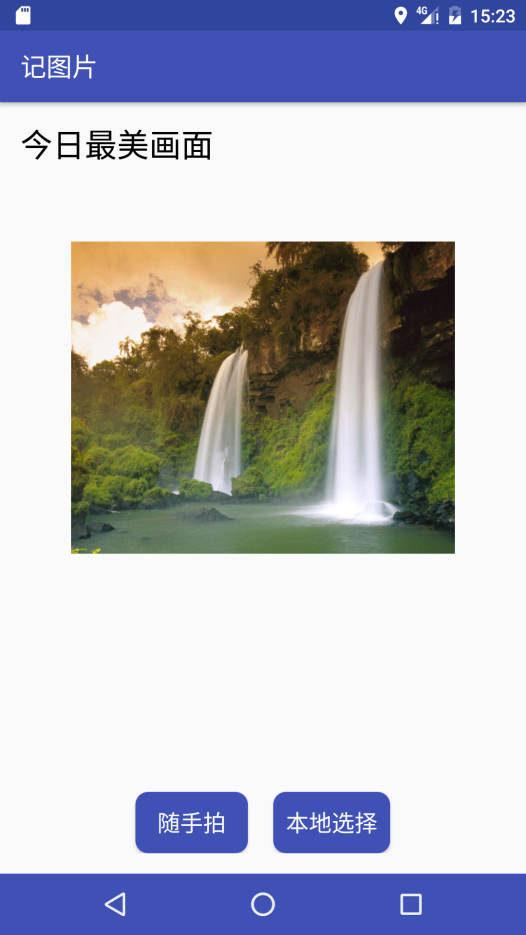
 

然后进入裁剪界面，选择裁剪区域后点击保存即可：

“本地选择”功能：点击“本地选择”，可从本地图库中选择，然后进行裁剪，最后点保存即呈现出来：



日记加密功能：

添加密码： 之后打开日记记录内容需要验证密码：

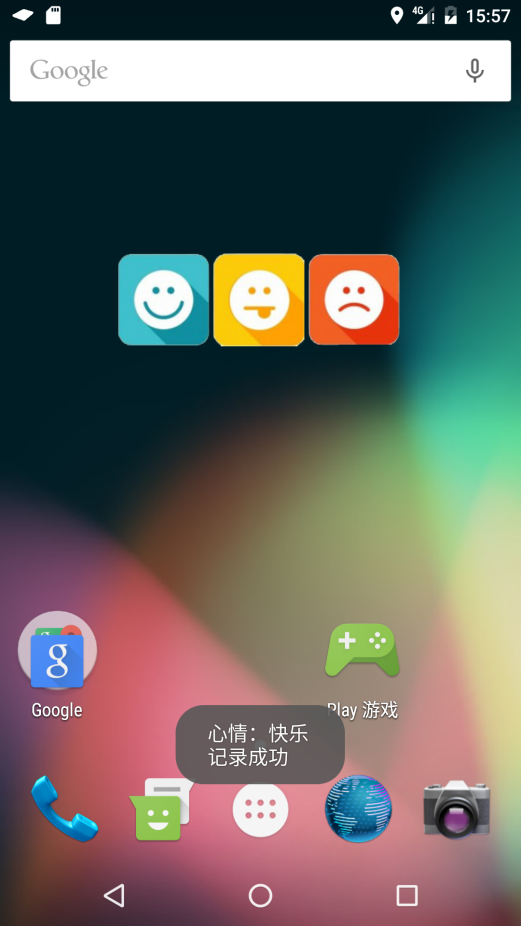
 

取消密码（需要先验证密码）：



心情记录功能：

从桌面拖出Widget，点击Widget对应图标，即可记录心情，心情指数会自动变化：



Notification提示：



**技术方法和工具**

**1. 界面设计**：使用了ViewPager，形成左右滑动切换页面的效果。另外首页还添加了一个Drawerlayout，实现侧滑菜单功能。

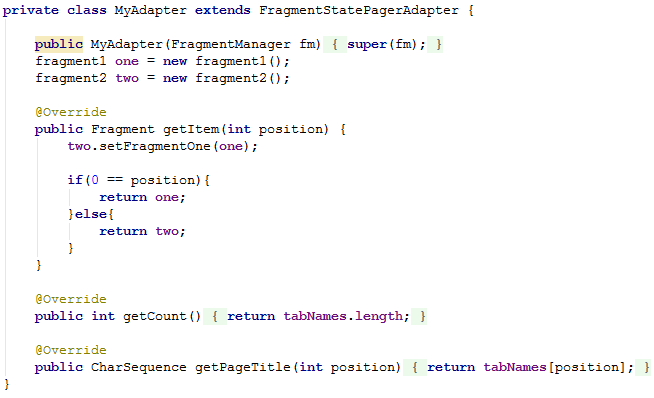
ViewPager的使用方法：

1）ViewPager类直接继承了ViewGroup类，所有它是一个容器类，可以在其中添加其他的view类。

2）ViewPager类需要一个PagerAdapter适配器类给它提供数据。

3）ViewPager经常和Fragment一起使用，并且提供了专门的FragmentPagerAdapter和FragmentStatePagerAdapter类供Fragment中的ViewPager使用。

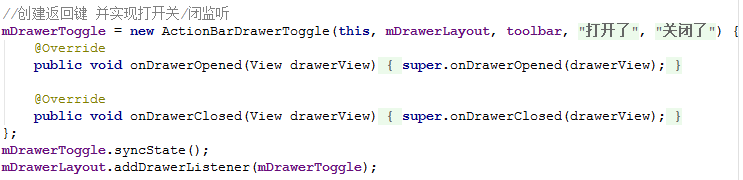
ViewPager的Adapter设置：



C:\Users\Owner\AppData\Roaming\Tencent\Users\1360275010\QQ\WinTemp\RichOle\D1[(TBZ6])}9)1HU@2W0{RS.png

Drawerlayout的使用方法：

DrawerLayout是Android support.v4 中的一个抽屉视图控件。使用这个控件，可以生成通过在屏幕上水平滑动打开或者关闭菜单，能给用户一个不错的体验效果。

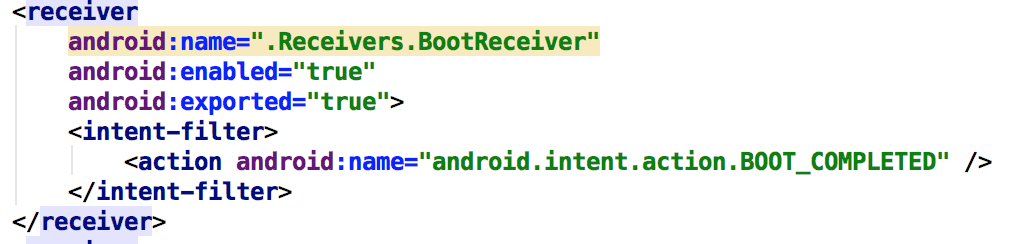


**2. 自动记录功能实现**

记录每天起床时间：

通过**开机广播接收器**记录开机时间，通过之前实现的MyDBHelper写到**数据库**中

前台**服务**通过查数据库获得今日起床时间，显示到**CardView**





记录每天睡觉时间：

通过**关机广播接收器**记录关机时间，通过之前实现的MyDBHelper写到**数据库**中

前台查数据库获得昨晚睡觉时间，显示到CardView

　　（注：起床和睡觉时间采用的是手机开关机时间，可能与实际的起床和睡觉时间略有不符，这里是为了自动获取，故采用此方法）

记录手机使用次数：

通过**解锁广播接收器**记录解锁时间，每次解锁将解锁时间通过之前实现的MyDBHelper写到数据库中

前台查数据库获得今日手机使用次数，显示到CardView

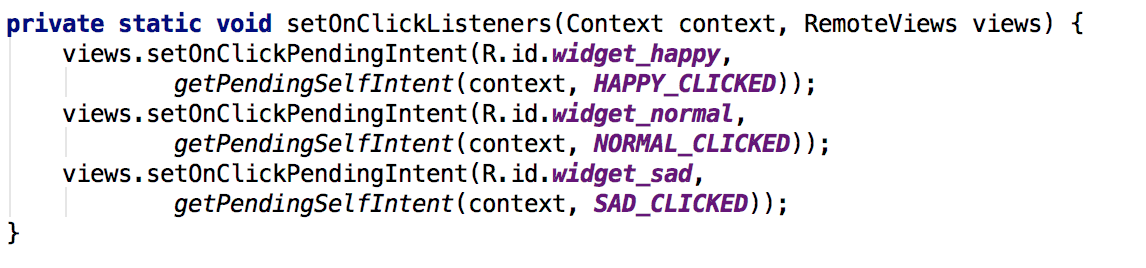
记录联系最多的人：

通过**接收来电广播和去电广播**，将来电去电号码记录到数据库中

前台通过查数据库获得今日联系最多的人，显示到CardView

记录心情：

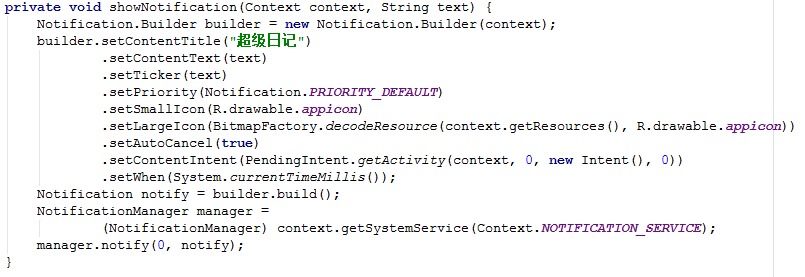
布局好记录心情的**Widget**，点击Widget就调用**后台服务**的相关接口记录当前时间和心情（量化为数），同时弹出**Notification**提示。下图1是设定OnClickPendingIntent的方法，相当于在Activity中设置OnClickListener。下图2是响应相应的Intent的方法。





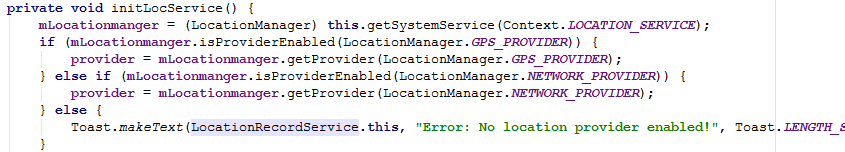
前台查数据库获得今日心情值，即今日所有心情记录的加权均值，显示到CardView

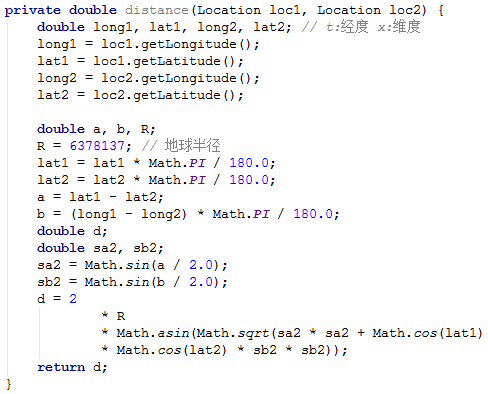
Notification设置：



记录逗留地点：

**后台服务**维护一个今日曾逗留地点的数据表，表头为（经纬度，逗留分钟数）。位置变化在100m范围内算同一个位置。每1min从**GPS传感器**获取一次位置，若当前位置是新位置，则插入新纪录。则若位置未改变，则当前位置的逗留时间增加1min。然后将逗留地点显示在地图中。使用了百度地图时用到了**网络访问**相关知识。



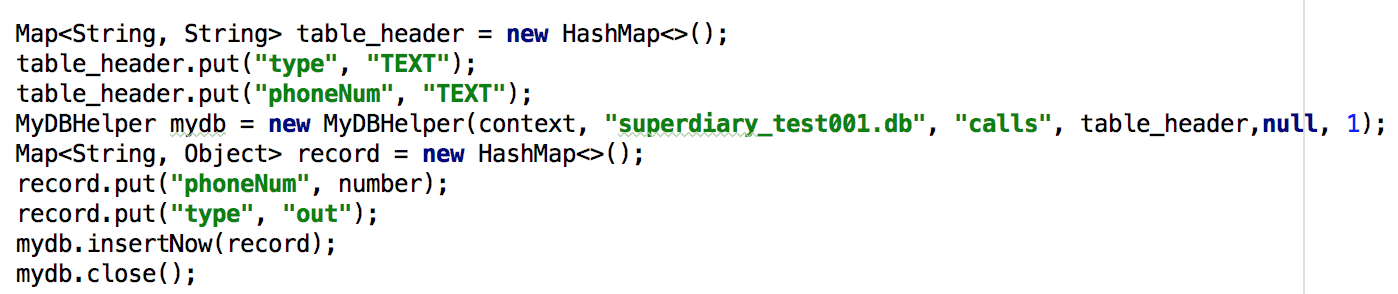


**3. 数据库实现**

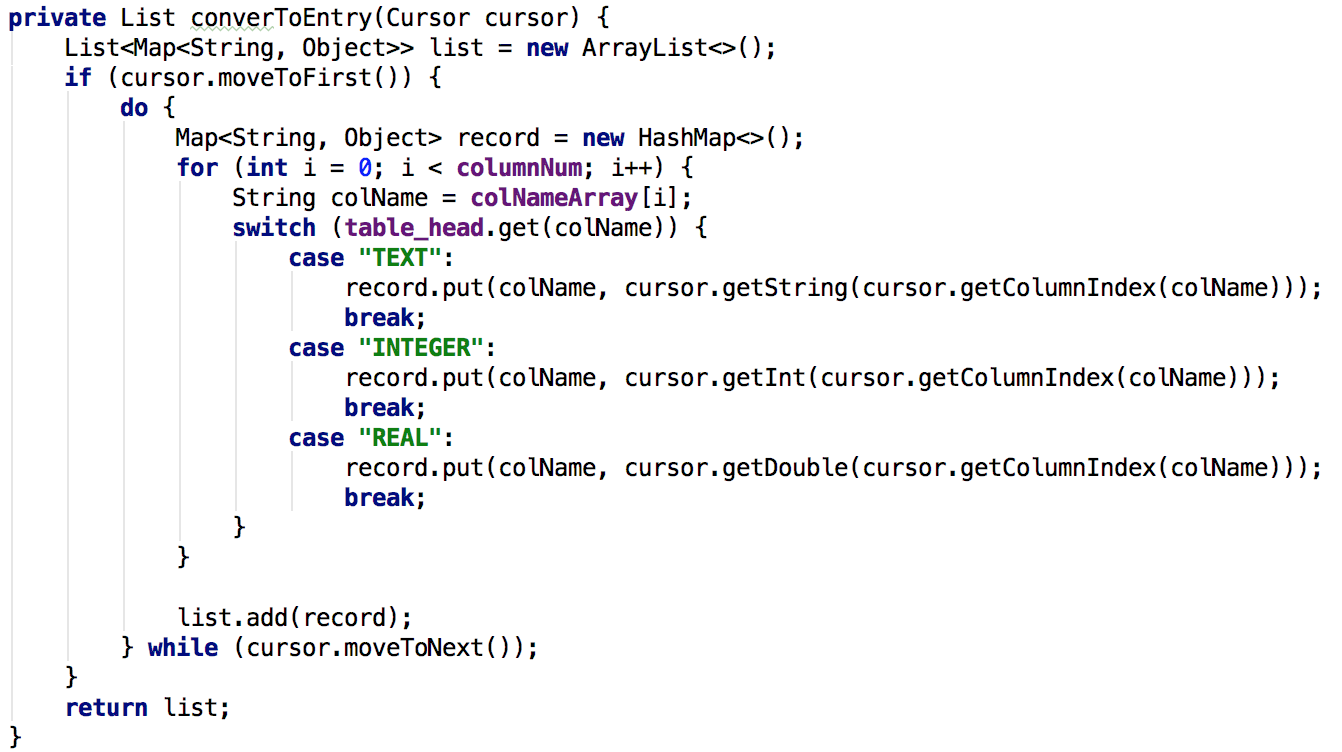
MyDBHelper类

数据库使用**SQLite**，并使用了**ContentProvider**的知识。该类与数据库的某个表绑定。与之前实验中的实现不同的是，这里的实现支持自定义表头。这是通过在初始化实例的时候传入一个存储键值对(列名，列类型)的Map容器来实现的。插入的记录也相应地根据自定义的表头来确定格式，同样通过Map容器以(列名，值)的形式保存记录，传入该类的insert方法。

下图是为记录来电和去电的数据表绑定MyDBHelper.

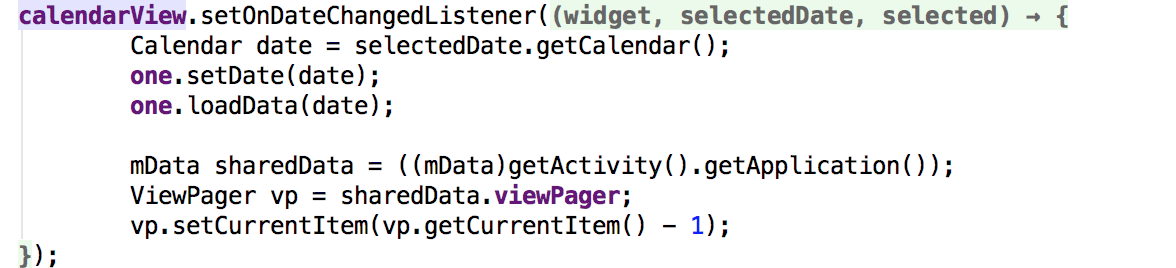


查询数据库时，该类会将所得Cursor变量转化为元素为以Map容器的形式保存的record的List容器再返回，而之前lab中的实现则是用固定的类的对象装。这里的实现更加灵活。下图是具体实现。



**4. 历史日记功能实现**

采用了**CalendarView**组件。从CalendarView中点击某一天会跳转到当天的日记界面，并且直接从数据库中相关的表中调出当天的信息，显示在CardView中。日历点击事件监听：



**5. 动画按钮实现**

有两处使用了按钮的动画效果：滑动菜单左上角的按钮和右下角的按钮。右下角的按钮使用FloatingActionButton组件实现。动画效果的实现原理是使用**Animation**。

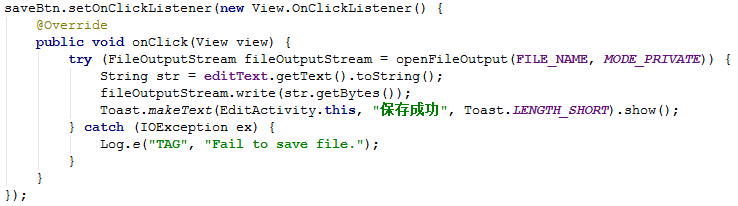
Android系统提供了很多丰富的API去实现UI的2D与3D动画，最主要的划分可以分为如下几类：

* **View Animation：** 视图动画在古老的Android版本系统中就已经提供了，只能被用来设置View的动画。
* **Drawable Animation：** 这种动画（也叫Frame动画、帧动画）其实可以划分到视图动画的类别，专门用来一个一个的显示Drawable的resources，就像放幻灯片一样。
* **Property Animation：** 属性动画只对Android 3.0（API 11）以上版本的Android系统才有效，这种动画可以设置给任何Object，包括那些还没有渲染到屏幕上的对象。这种动画是可扩展的，可以让你自定义任何类型和属性的动画。

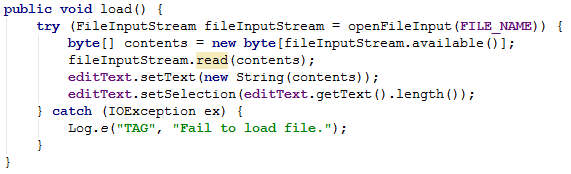
**6. 文字记录功能实现**

使用到了**文件读写**的知识。将每天记录的文字使用**Internal Storage**保存在文件中。保存在Internal Storage上的文件，只有这个应用自己可以看到和使用，其他应用和用户本身无法访问这些文件。

写文件：



读文件：

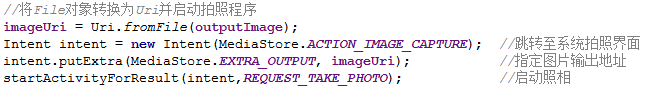


**7. 图片记录功能实现**

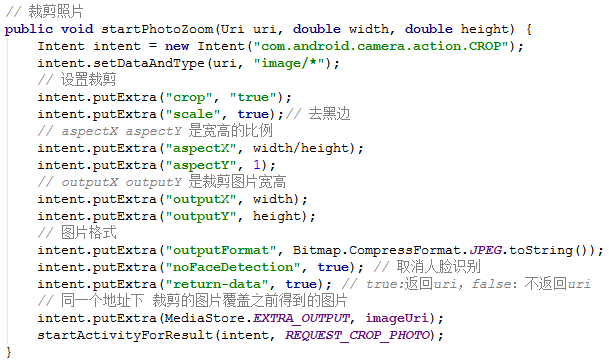
拍照记录功能：

首先要开启拍照功能权限，然后调用系统摄像头进行**拍照（使用MediaStore.ACTION\_IMAGE\_CAPTURE）**，并编写图片裁剪函数对图片进行**裁剪**，最后使用setImageBitmap把图片显示到ImgeView中。另外，使用**External Storage**方式将拍照并裁剪后的图片**保存到本地**。

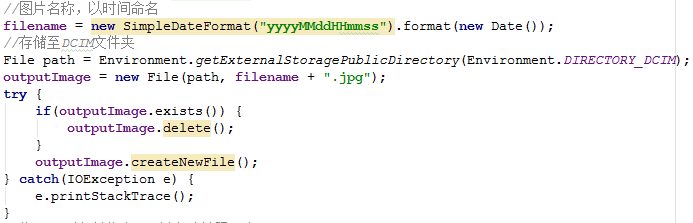
拍照：

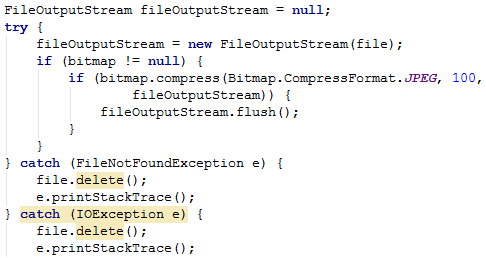


裁剪图片：



保存图片：





本地选择图片上传：

使用Intent.ACTION\_PICK，并设置类型为"image/\*"。然后同样调用图片裁剪函数对图片进行裁剪并显示出来，然后使用External Storage方式将裁剪后的图片保存到本地。

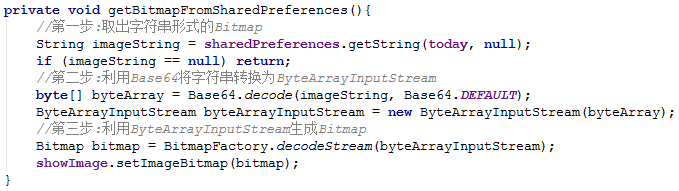
C:\Users\Owner\AppData\Roaming\Tencent\Users\1360275010\QQ\WinTemp\RichOle\(`S1XS0)`U{51M@LFTGH40D.png

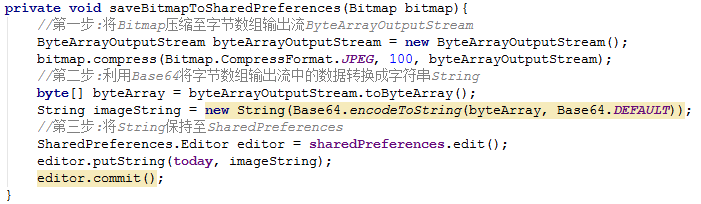
各个功能的调用流程处理：使用startActivityForResult启动功能，并在onActivityResult中根据requestCode进行对应的处理。



图片持久化的实现：

这里使用SharedPreferences来保存。使用SharedPreferences保存图片时，需要用Base64对Bitmap和String进行互相转换。





**8. 密码保护功能实现**

设置密码时，密码经过MD5哈希加密后，将哈希值存入SharedPreferences中。之后打开日记时，需要先输入密码，将输入的密码的MD5哈希值与SharedPreferences中保存的哈希值进行比较，若相同则视为验证成功，否则验证失败。

设置密码代码实现：



**使用到的技术点总结如下：**

1. 界面框架设计（用户友好，控件布局合理，美观）

2. 事件处理的实现（点击事件、activity 跳转等）

3. SQLite、ContentProvider、SharedPreferences等数据存储方式

4. Notification、Widget 等显示方式

5. Broadcast

6. Service

7. 网络访问

8. 传感器使用

9. 动画应用

10. 文件访问和存储（包括内部存储和外部存储）

11. Viewpager、Drawerlayout、CardView、CalendarView等组件的使用

12. 拍照功能应用

13. 图片裁剪功能

**参考资料**

1. ViewPager的使用

<https://my.oschina.net/summerpxy/blog/210026>

<http://www.jianshu.com/p/e5abbda4a71c>

2. Drawerlayout的使用

<http://www.jcodecraeer.com/a/anzhuokaifa/androidkaifa/2014/0925/1713.html>

3. 拍照功能的两种实现方式

<http://blog.csdn.net/napolun007/article/details/6103307/>

**遇到的问题以及解决方法**

　　1. 从文件中添加文字到EditText中后，会出现底部红线

　　这是拼写检查功能引起的，需要设置inputType属性为textNoSuggestions。但这样做又会遇到另一个问题：无法自动换行。因此需要进行如下设置：

　　android:inputType="textMultiLine|textNoSuggestions"

2. EditText导入文字后，光标出现在最前面

在代码中设置editText.setSelection(editText.getText().length())。

3. 拍照时出现异常

权限问题所致。在API23中运行，需要动态开启权限；而在API24及以上运行时，异常是由URI问题导致的，官方解释如下：

对于面向 Android 7.0 的应用，Android 框架执行的 [StrictMode](https://developer.android.google.cn/reference/android/os/StrictMode.html) API 政策禁止在您的应用外部公开 file:// URI。如果一项包含文件 URI 的 intent 离开您的应用，则应用出现故障，并出现 FileUriExposedException 异常。

即Android升级到7.0后，对权限又做了一个更新，即不允许出现以file://的形式调用隐式APP，需要用共享文件的形式：content:// URI。具体实现时，要用ContentValues方法来实现。而本程序目前不是用该方法实现的，因此不支持API24，可使用API22或23运行。

4. 图片无法存入数据库中

数据库中一般存的是字符串，可以考虑存储图片保存的路径，也可以考虑将Bitmap转换成String再存入数据库（使用Base64转换）。