

记一记

说明文档

2016 年 12 月 28 日

# 目录

一、应用简介.....	2
二、应用详情.....	2
1. 主界面.....	2
2. 便签.....	3
(1) 便签显示页面.....	3
(2) 添加便签.....	4
(3) 查看并修改便签.....	4
(4) 查找便签.....	5
(5) 删除便签.....	5
3. 备忘.....	6
(1) 备忘显示页面.....	6
(2) 添加备忘.....	7
(3) 查看并修改备忘.....	8
(4) 备忘提醒（通知、闹铃）.....	8
4. 日记.....	9
(1) 输入日记锁.....	9
(2) 显示日记列表.....	10
(3) 添加日记.....	10
(4) 查看并修改日记.....	11
(5) 删除日记.....	11
5. 我.....	12
(1) 修改昵称/所在地.....	12
(2) 修改头像.....	12
(3) 设置/修改日记锁.....	13
(4) 摇一摇.....	14
(5) 关于.....	14
三、技术实现.....	15
1. 必选内容: .....	15
2. 自选内容: .....	15
四、参考资料.....	15
五、遇到的问题以及解决方法.....	15
1. 权限问题.....	15
2. 头像修改问题.....	16
3. Sqlite 数据库更新问题 .....	16

# 一、应用简介

此应用是一款简约易用的工具应用，为用户量身定制，包含便签、备忘、日记等常见的 app 功能，集现在、将来、过去三种时态的记录功能于一体，为减少手机上的 app 数量而开发，让你的手机减轻负担，让相关功能更加集中，便于查找、使用。此外，本应用还添加了摇一摇等功能，随机显示一条励志的言论来给予用户正能量。

# 二、应用详情

(结合运行界面、采用的技术、方法和工具对应用功能进行分析)

## 1. 主界面

“记一记”的主界面是一个 ActionBar 与一个支持左右滑动切换页面的控件 ViewPager 的结合，通过给 ViewPager 添加适配器 PagerAdapter，并对 ViewPager 进行监听，可实现左右滑动时页面的切换以及 ActionBar 中 Button 文字颜色的改变，同时对 ActionBar 中各 Button 添加点击事件用来切换页面，具体运行界面如下：



图 1-1



图 1-2



图 1-3



图 1-4

## 2. 便签

### (1) 便签显示页面



图 2-1

便签列表是用 RecyclerView 控件来维护的，它的每一项都是一个卡片控件 CardView，包含便签的第一行与最后编辑的日期时间

页面左上角的“放大镜”是搜索控件 SearchView，这是 android 自带的控件，用以输入字符串搜索全部便签

页面右下角的圆形图标是悬浮按钮控件 FloatingActionButton，用以跳转到添加便签页面。

## (2) 添加便签

点击图 2-1 中的控件 FloatingActionButton 跳转到添加便签页面,编辑成功后保存并更新到便签页面中的 RecyclerView 中。

便签的存储是外部文件存储的方式,保存时将便签的第一行与最后编辑的时间添加到 Sqlite 数据库表格 notes 中,RecyclerView 中的每一项 Sqlite 数据库 notes 表格的一行。

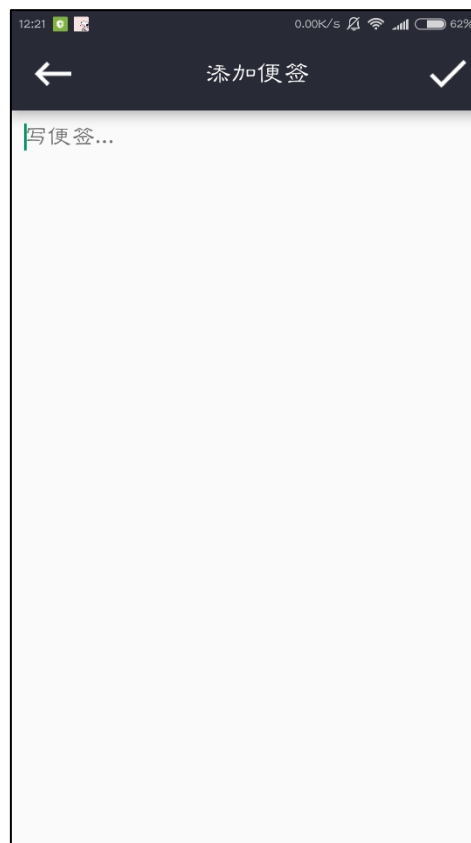


图 2-2

## (3) 查看并修改便签



图 2-3

点击图 2-1 中 RecyclerView 中的项,跳转成对应的“查看与更新便签”页面

此处可更新便签,点击右上角保存按钮后,会删除 Sqlite 数据库中表格 notes 中对应的项,再以第一行、最后编辑时间插入表格中作为新的项,以此让便签在 RecyclerView 中实现按时间顺序排列显示。

#### (4) 查找便签

点击图 2-1 中的 SearchView 控件，输入所要搜索的关键字符串搜索，跳转到搜索便签结果页面

此处使用的是模糊匹配，只要便签中含有所搜索的关键词，就将该便签的内容显示在 ListView 控件中。



图 2-4

#### (5) 删除便签



图 2-5

长按图 2-1 中 RecyclerView 中的卡片，此时会以 2D 动画的方式弹出一个按钮，如左图中回收箱图标所示，这里是用 PopupWindows 的技术实现的。

点击图标确认删除，则同时删除对应的内部文件、Sqlite 数据库中表格 notes 中对应的行，并更新 RecyclerView 的项。

3. 备忘

(1) 备忘显示页面



图 3-1 (a)

App 中的备忘列表:

备忘的列表使用的控件是 ListView, 从 Sqlite 数据库中的表格 Reminder 中获取每一行并显示出来, 每一项包含备忘的标题和备忘设置的日期时间

页面右下角悬浮按钮控件 FloatingActionButton 用于添加备忘。

桌面上的备忘列表:

此处使用了 AppWidget 的知识点, 将备忘的列表显示在桌面上, 由于 AppWidget 中包含了 ListView, 需使用 RemoteViewService 服务来对它进行维护: 对 ListView 中项的增加、减少做更新、对 ListView 中每一项填充点击事件, 使之点击后跳转到“查看或修改备忘”页面

此外, 点击“备忘”可快速进入写备忘页面。



图 3-1 (b)

## (2) 添加备忘

点击图 3-1(a) 中的控件 FloatingActionButton 跳转到“写备忘”页面，标题和日期、时间为必填，保存后插入到 Sqlite 数据库表格 Reminder 中，并更新图 3-1(a)、图 3-1(b) 中的 ListView 内容。

保存的同时也为该备忘添加了闹钟提醒，此处使用的是系统服务 ALARM\_SERVICE 和闹钟管理器 AlarmManager，闹钟以通知和闹铃的形式通知。

日期和时间的设置使用的是安卓自带的对话框 DatePickerDialog(如图 3-2(b) 所示)和 TimePickerDialog(如图 3-2(c) 所示)。



图 3-2(a)



图 3-2(b)

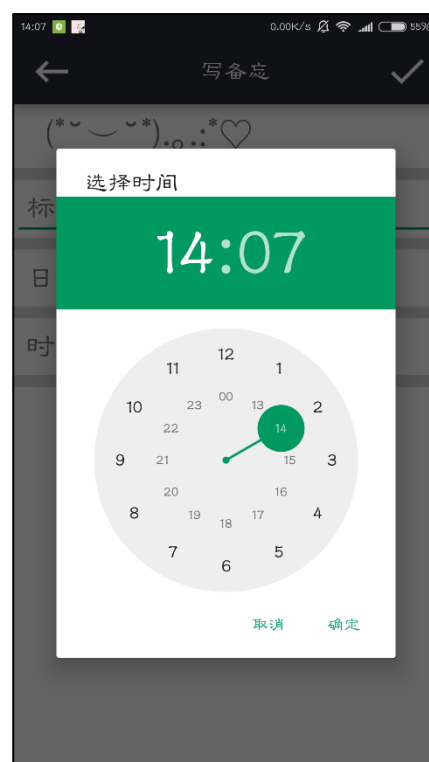


图 3-2(c)



### (3) 查看并修改备忘

点击图 3-1(a) 中 ListView 每一项，进入备忘详情，支持查看与修改，当点击右上角保存按钮时，之前对应的备忘闹钟会被取消，Sqlite 数据库中对应的行会被删除，之后设置新的闹钟提醒、插入 Sqlite 数据库 Reminder 表格中新的一项。同时更新图 3-1(a) 中的 ListView 列表。



图 3-3

### (4) 备忘提醒（通知、闹铃）

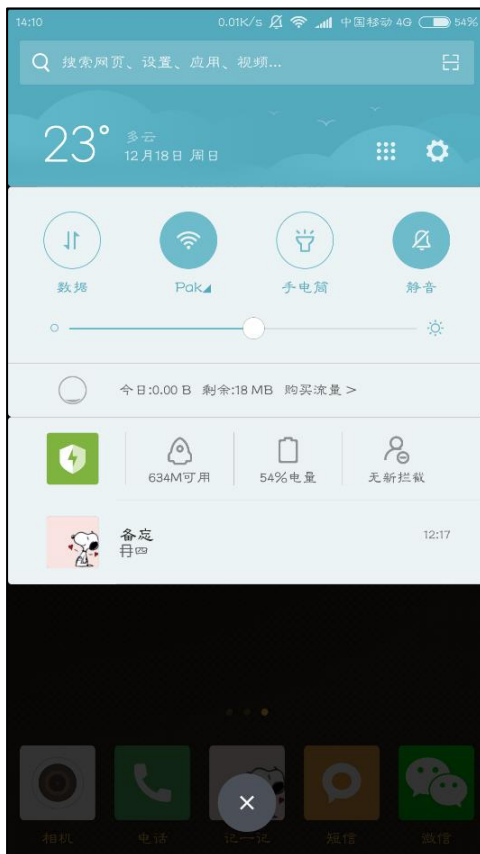


图 3-4

备忘设置的提醒时间一到，AlarmManager 设置的闹钟就会响，同时发送静态广播，AlarmReceiver 接收到静态广播后，发送相对的通知，显示在手机的通知栏中，点击通知可快速进入备忘的详情页面。

#### (5) 删除备忘



图 3-5

长按图 3-1(a)中某一项,可选择是否删除备忘,此处使用的是简单对话框,若删除,则取消该备忘所设置的闹钟服务,删除 Sqlite 数据库 Reminder 表中对应的一行,同时更新图 3-1(a)、图 3-2(b)中 ListView 的内容。

### 4. 日记

#### (1) 输入日记锁

使用 SharedPreferences 来存储日记锁,未设置日记锁时,无法进入,提示前往“我——设置日记锁”中设置日记锁以开启日记本功能,日记锁匹配才可进入日记列表。



图 4-1

## (2) 显示日记列表

日记列表使用的是 ListView 控件，与便签不同的是，它的每一项多了地点、天气状况这两项，因此 Sqlite 数据库中也多了对应的两列。

此外，日记使用的是内部文件存储，使得日记内容难以在手机 sd 卡中被找到。

页面右下角悬浮按钮 FloatingActionButton 用于添加日记。



图 4-2

## (3) 添加日记

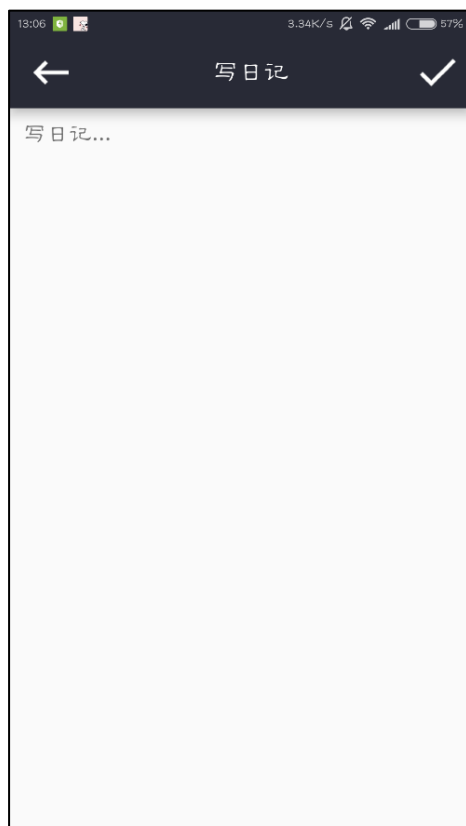


图 4-3

点击图 4-2 中悬浮按钮跳转到“写日记”页面，编辑好日记按右上角保存按钮后，以内部文件存储的方式保存文件，以当时的日期时间为文件名，自动获取所在城市、天气状况，插入到 Sqlite 数据库 diaries 表中。

获取所在城市使用的是 LocationManager，定位的方式是通过 NETWORK\_PROVIDER，获取到经纬度后，通过 Geocoder 解析出城市名。

获取当时天气是网络访问相关网站（HttpURLConnection 的知识点），使用多线程和 Handler 提交申请，用 XmlPullParserFactory 解析 xml 文档得到天气。

#### (4) 查看并修改日记

点击图 4-1 中 ListView 某一项，可查看与更新对应日记详情，此处若更新，采取的更新措施与便签一致：先删除再创建新的日记，再更新图 4-1 中 ListView 的列表内容。

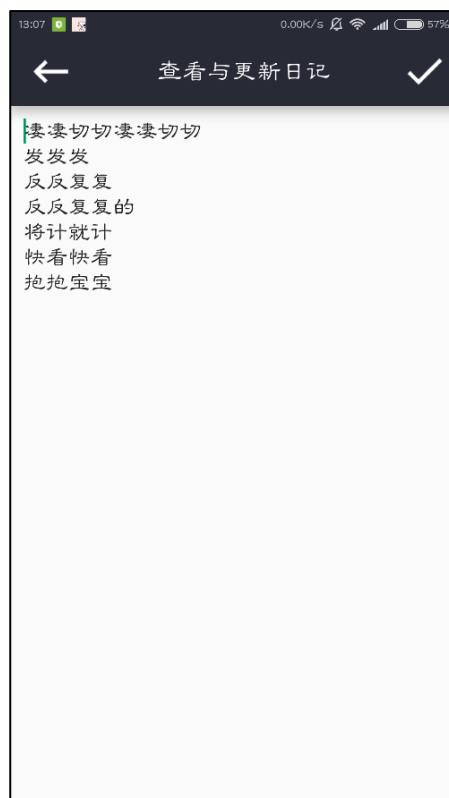


图 4-4

#### (5) 删除日记



图 4-5

长按图 4-1 中 ListView 某一项，可选择是否删除日记，这里使用了第三方的对话框：NiftyDialogBuilder，它的出现是以动画的形式从上往下滑出。

删除的动作是删除对应的内部文件、删除 Sqlite 数据库 diaries 表中对应的一行、更新图 4-1 中 ListView 的列表内容。

## 5. 我

### (1) 修改昵称/所在地

如下图 5-1(a)、5-1(b)所示，点击昵称、所在地，分别进入昵称、所在地的修改，使用 SharedPreference 保存。



图 5-1(a)



图 5-1(b)

### (2) 修改头像

点击头像可进行头像修改，这里用的是 ContentProvider 去得到系统相册中所选图的路径，再进行剪切、更换、保存。

此处需要开启权限：

READ\_EXTERNAL\_STORAGE、



图 5-2

### (3) 设置/修改日记锁

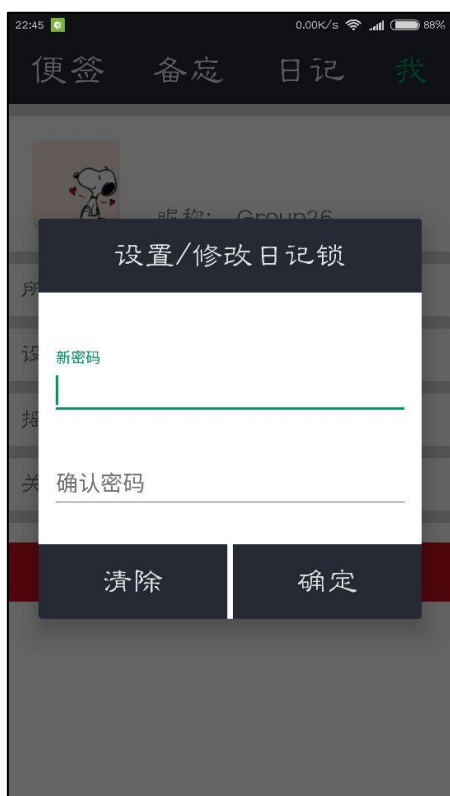


图 5-3(a)

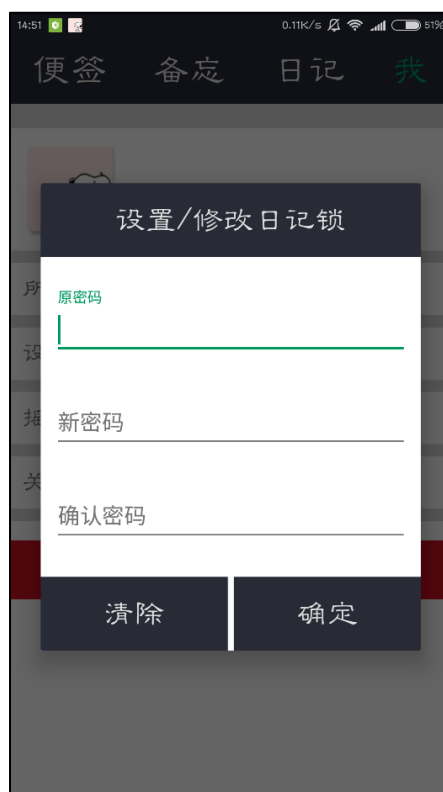


图 5-3(b)

点击设置日记锁，可进行设置，第一次开启为开启日记本功能，需设置日记锁密码，如图 5-3(a) 所示，之后每次开启为修改日记锁密码。如图 5-3(b) 所示。日记锁密码的保存使用的是 SharedPreference 的存储方式。

这里对输入的逻辑判断是在控件中实现的——此处不是简单的 EditText，而是 TextInputLayout 控件，可对每一个输入设置监听，在监听中进行判断是否输入正确。根据情况弹出提示，如图 5-3(c) 为输入错误提示。

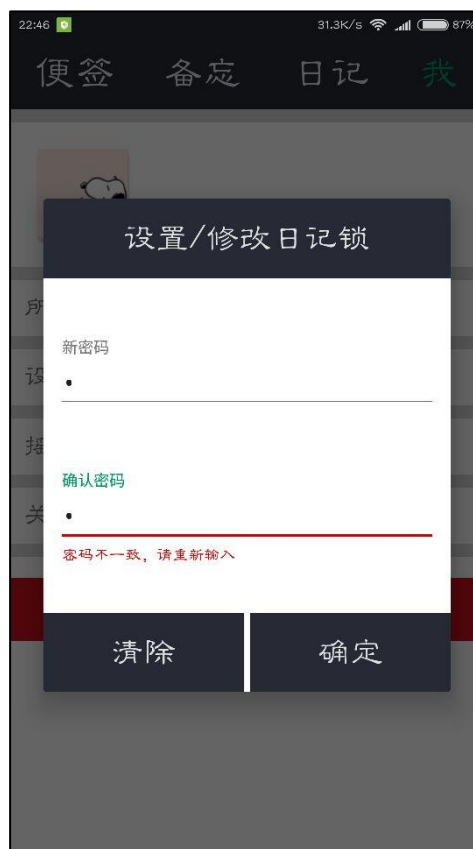


图 5-3(c)

#### (4) 摇一摇

使用加速度传感器，添加监听，实现摇一摇的功能，从 Sqlite 数据库中随机抽取一句励志的话，用以激励用户，给予用户正能量。如下图所示，另外，两次摇一摇之间需要间隔 3s。



图 5-4(a)

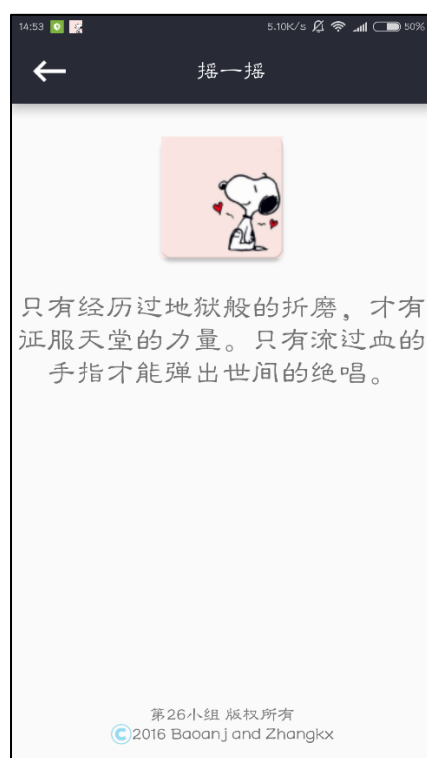


图 5-4(b)

#### (5) 关于 版本信息说明



图 5-5

### 三、技术实现

(此处仅列明此 app 用到的实验文档中的技术要求)

#### 1. 必选内容:

- 界面框架设计 (用户友好、控件布局合理、美观)
- 事件处理的实现 (点击事件、activity 跳转等)
- 数据存储方式
  - SQLite ——用于便签、备忘、日记的标题等存储
  - 内部文件存储 ——用于日记正文存储
  - 外部文件存储 ——用于便签正文存储
  - SharedPreferences ——用于日记锁、用户名、用户头像等存储

#### 2. 自选内容:

- Notification ——用于备忘的提醒通知
- Widget ——用于备忘的桌面显示
- Broadcast ——用于备忘的闹钟广播
- Service (RemoteViewService) ——用于 widget 的 ListView 维护
- 网络访问功能 ——用于日记中获取天气状况
- 传感器使用 ——用于日记中定位和实现摇一摇
- 2D 动画应用 ——用于删除时的动画显示

### 四、参考资料

#### (1) Material Design 规范

[http://wenku.baidu.com/link?url=ru87G9V35aP9UBdelfH3EVYtRfEiPEEq9KyioKDtGE3h69S0gdkqkVRuoZamM1lWINtz0rWo1jvcA\\_R3C8ylXYR4XEhW4EUI4QvwAdbsWVW](http://wenku.baidu.com/link?url=ru87G9V35aP9UBdelfH3EVYtRfEiPEEq9KyioKDtGE3h69S0gdkqkVRuoZamM1lWINtz0rWo1jvcA_R3C8ylXYR4XEhW4EUI4QvwAdbsWVW)

#### (2) NiftyDialogEffects (第三方): 动画对话框, 用于长按删除日记列表项

<https://github.com/sd6352051/NiftyDialogEffects>

#### (3) CardView : 卡片, 将列表中的每一项的多个控件集成一个控件

<http://blog.csdn.net/jdsjlzx/article/details/49511215>

#### (4) PopupWindow : 动画弹窗, 用于长按删除便签列表项

<http://blog.csdn.net/csdnzouqi/article/details/51433633>

<http://www.runoob.com/w3cnote/android-tutorial-popupwindow.html>

#### (5) AlarmManager : 闹钟, 用于备忘的提醒

<http://www.cnblogs.com/mengdd/p/3819806.html>

[http://blog.csdn.net/wangxingwu\\_314/article/details/8060312](http://blog.csdn.net/wangxingwu_314/article/details/8060312)

#### (6) RemoteViewService : 远程视图服务, 用于维护 AppWidget 中 ListView 的添加、删除, 以及点击事件

<http://www.tuicool.com/articles/u26jQb>

#### (7) ContentProvider : 修改头像

<http://blog.csdn.net/marksinoberg/article/details/51175831>

### 五、遇到的问题以及解决方法

#### 1. 权限问题

**问题描述:** 多次因为权限不足而出现闪退, 在 AndroidManifest.xml 文件



中已经给予了足够的权限,但在真机测试时,手机没有默认开启相关权限,导致部分功能不能运行,程序崩溃,比如定位权限,在写日记的页面中,日记保存时会获取当前位置,当手机权限没开启时,但会导致程序崩溃;修改头像时,存储权限如果没有开启,也会导致程序闪退。

**解决方法:** 在相关代码执行前加入权限检测,如:

```
PackageManager pm = getPackageManager();
boolean permission = (PackageManager.PERMISSION_GRANTED ==
    pm.checkPermission("android.permission.ACCESS_FINE_LOCATION",
        "com.example.kaixin.final_experiment"));
if (permission) {
    LocationUtils.getCNByLocation(AddDiary.this);
    city = LocationUtils.cityName;
}
```

## 2. 头像修改问题

**问题描述:** 头像修改此处使用的是 contentProvider 得到系统相册中所选的图的路径,然后将图片类型改为 bitmap,再 setImageBitmap(),但部分图片不能显示出来,ImageView 控件变为白色。

**解决方法:** 网上有很多人求助,相关问题很多,但试着更改后仍无效,最后将图片裁剪为正方形,再改为 Bitmap, setImageBitmap(),头像修改成功。

## 3. Sqlite 数据库更新问题

**问题描述:** 小组两人各自完成部分内容,各自使用了 Sqlite 数据库,最后在合并时,忽略了 Sqlite 数据库的更新问题,只是简单的将表汇集到同一个 Sqlite 数据库中,导致在原先的基础上没有重新生成表。

**解决方法:** 重写 SQLiteOpenHelper 中的 onUpgrade(SQLiteDatabase s, int I, int j)方法,如下:

```
@Override
public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int j) {
    sqLiteDatabase.execSQL("drop table if exists Reminder");
    sqLiteDatabase.execSQL("drop table if exists notes");
    sqLiteDatabase.execSQL("drop table if exists diaries");
    sqLiteDatabase.execSQL("drop table if exists Shark_TABLE");
    onCreate(sqLiteDatabase);
}
```