

实验报告

课程名称： 智能终端软件开发技术

APP名称： 树洞

任课老师： 张 寅

目录

实验报告 1

一．应用简介 3

1.1 背景 3

1.2 与同类APP的对比 4

1.3 运行环境 7

二．总体架构及功能设计 7

2.1 总体架构 7

2.2 注册登录功能设计 8

2.3 录音功能设计 9

2.4 录音时长更新功能设计 10

2.5 录音音量显示功能设计 11

2.6 录后即焚功能设计 12

2.7 基于数据库的recycleView功能设计 13

2.8 播放功能设计 13

2.9 阅后即焚功能设计 14

三．关键数据结构/算法 15

3.1 数据库 15

3.2 MediaRecorder类实现录音和播放 16

3.3 Seekbar拖动条 17

3.4 RecycleView展示列表 19

四．开发困难及解决方案 20

4.1 获取行之有效的文件路径 20

4.2 自定义录音音量显示的View 20

4.3 RecycleView的设计实现 21

五．分工 21

六．总结 22

### 

### 一．应用简介

#### 1.1 背景

#### 心理问题频发

当今浮华的社会上，愈来愈多的心理问题或多或少出现在我们身边。抑郁症、自闭症、自杀……

随着社会节奏加快，在过去30年当中，焦虑和抑郁变大了，因此焦虑症和抑郁症也变多了。焦虑症的发病率在上世纪八十年代只有1%到2%的样子，现在是13%。我们小组私以为频发的心理问题一定程度上是由于人们没有舒缓压力、释放自己情绪的方式方法，因此想到了制作该APP。

#### 没有情感归属

当今社会由于网上各种爆料八卦，人们更倾向于保守自己的小秘密，而不是冒着被捅出去的风险向人倾诉，难以互相信任，没有情感归属。而面对父母亲人，人们又会因为怕家人担心而报喜不报忧，依旧没有情感归属。

基于种种原因，人们会将自己的情感压抑在心底，而不去给它一个发泄口，一个归属地，这不仅仅容易引发心理问题，而且会对身体不好。长此以往容易亚健康。

#### 倾诉的好处

倾诉可以通过说出自己的心事，释放很多的心理压力，从而改善情绪。研究表明经常倾诉的人有心理问题的几率会较小一些。并且通过倾诉可以给情绪一个宣泄口，让情感有归属地。进而降低焦虑和抑郁，减少产生心理问题的几率。

倾诉的过程同时也有助于理清自己的思路，能够逐渐找到自己想要的答案，减少自己一个人纠结的时间，从而改善情绪。同时倾诉中找到问题解决思路后，压力和焦虑自然也会减少。

#### 总结

基于以上这些背景，我们小组认为，开发一个供人们倾诉自己心灵的树洞app是有市场需求的。同时本着人道主义关怀，一个能为人们提供情绪宣泄口、释放人们心理压力，减少社会心理问题的app也是有必要被开发出来的。

#### 1.2 与同类APP的对比

多番寻找下，并没有找到完全一样的APP，但根据我们app的功能——记录和情感，有这些app可以比对。

#### 记事本类---华为备忘录

华为自带的备忘录具有清单功能，也就相当于todolist，插入图片功能，插入语音功能，插入手写功能（配合图片上内容讲解）等等，而且可以上指纹锁，只有自己才能打开查看，确实是可以记录心中小秘密的。如图：



#### 日记类---一本日记

一本日记作为简约化日记本，除了具备备忘录的相关功能，还具有缩进和无序列表功能，而且可以修改主题、背景以及自动插入天气和时间地点等。基本上与备忘录是一致的，也是可以记录心中小秘密的。



#### 倾诉类---倾诉

它的主打功能是陌生人分享秘密，倾诉app比较类似于一个全是陌生人的QQ空间或朋友圈，谁也不认识谁，因此可以将小秘密和倾诉的情绪作为动态传上去。可以发表动态来倾诉自己的心。

但是，由于大家多数都是来释放压力的，动态中充斥着负能量，只有少数是来宣泄开心幸福的。满满负能量很可能让倾诉完的你心情更加压抑。



#### 交友类---Soul

Soul的主打功能是找寻灵魂伴侣，其实本质上就是陌生人交友软件。Soul则具有灵魂匹配、恋爱铃、语音匹配、视频匹配等一对一聊天匹配以及群聊派对这样的多人聊天项目等等。

因此可以与大数据匹配的适合你的人进行倾诉。当然，对方是否愿意接受负能量，或你是否会需要接受他人负能量也不能确定。



#### 我们的树洞APP

与这些app不同。我们采用录音方式记录你的倾诉，既方便省事，也可以避免打字排版的费时费力影响思路。

与交友app倾诉app必定有陌生人拿到秘密不同，我们倾诉的保密性很好。录音存放在app内部目录，卸载后即消失，也不会被手机识别为音频导致可以轻易找到原文件。可以录后即焚，不保存就没有暴露的可能。同时我们的app仅运行本地端，录音只有你手上有一份，很安全。而且一个本地端的应用不会接触到其他人，也就避免了交友app和倾诉app那种被他人负能量影响的情况。同时，采用绿色系风格，也更能让人敞开心扉。



#### 1.3 运行环境

1. Android SDK version 24🡪 SDK version 30
2. Android Gradle Plugin Version 3.4.1
3. Gradle Version 5.1.1

### 二．总体架构及功能设计

#### 2.1 总体架构

作为一个录音类倾诉app，首先需要录制倾诉功能。录制之后还需要播放已录制的录音，同时，需要给出自已录制录音的列表。而作为一个app，还需要一个让人敞开心扉开始向导。

因此我们的app具有四个主要Activity页面：开始界面MainActivity、录制界面（倾诉界面）recordActivity、历史录音界面（回忆界面）showActivity、播放界面PlayActivity。

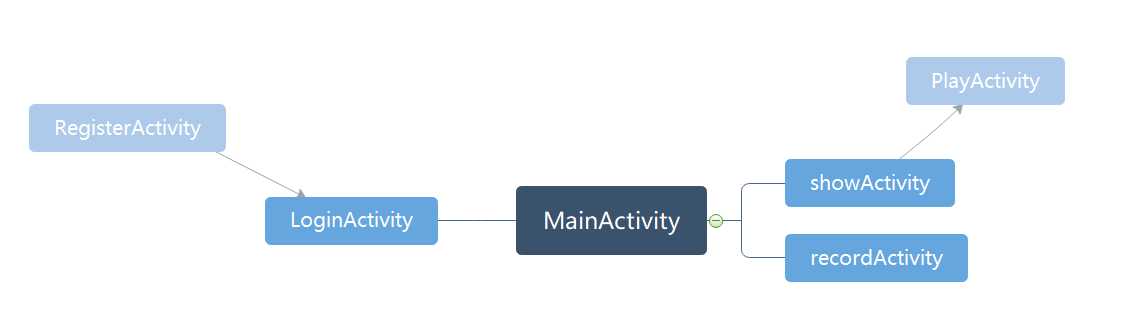
对于开始界面，要求美观、让人有诉说的欲望，并且切合app主题。



对于录音页面，同样要求让人有诉说的欲望，并且需要具有录音功能，记录录音时长功能、显示音量功能、储存功能以及可选的录后即焚功能。

对于历史录音界面，需要具备数据库实现的recycleview来显示录音列表，并且可以跳转播放录音。

对于播放界面，阅后即焚是一个可选的选项，还需要播放功能。

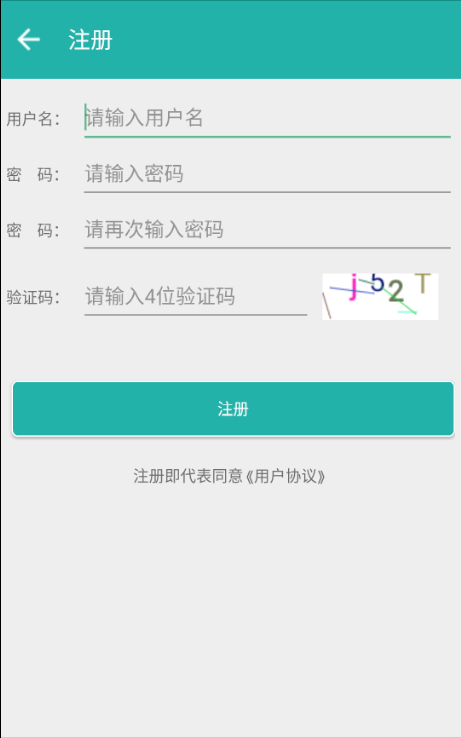


#### 2.2 注册登录功能设计

实现登录注册功能，用DBOpenHelper(extends SQLiteOpenHelper) 来创建数据表，然后再进行数据表的增、删、改、查操作和登录验证。

注册时注册页面实现了点击刷新验证码图片和一个注册按钮，其中要检查输入的验证码和图片中的验证码是否对应，两遍密码是否一致，都通过则写入数据表。其中验证码图片用的是一个随机数数组，其中去除了易混淆和经常和随机线混在一起看不清的数字和字母比如：数字0和字母o O、数字1和字母i I l L、数字6和字母b、数字9和字母q、字母c C和G、字母 t。

登录验证首先从EditText的对象上获取文本编辑框输入的数据，并把左右两边的空格去掉进行匹配验证,先判断一下用户名密码是否为空，再进而for循环判断是否与数据库中的数据相匹配一旦匹配，立即将match = true；break；否则一直匹配到结束 match = false；登录成功之后，进行页面跳转。

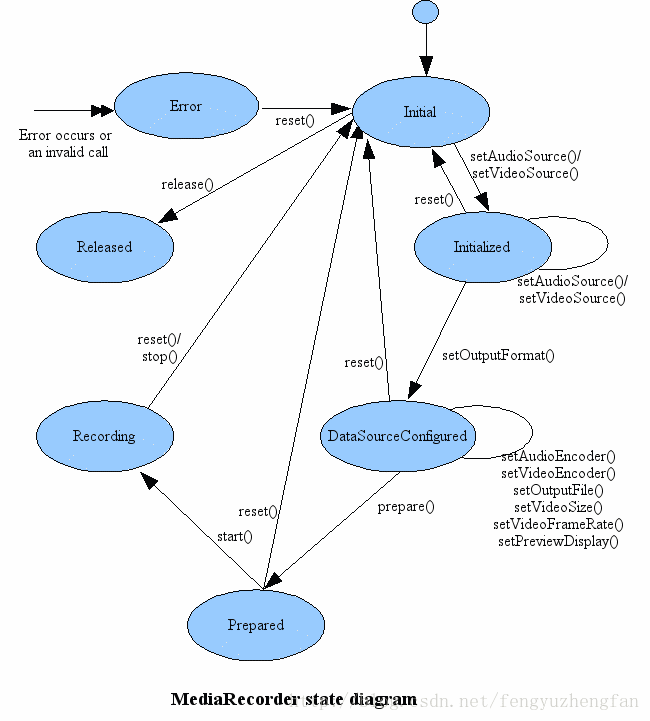
 

#### 2.3 录音功能设计

采用MediaRecorder类进行录音。

事实上，MediaRecorder和AudioRecord都可以录制音频，但相较于AudioRecord，MediaRecorder录制的音频文件是经过压缩后的，需要设置编码器。并且录制的音频文件可以用系统自带的Music播放器播放。而AudioRecord录制的是PCM格式的音频文件，需要用AudioTrack来播放，AudioTrack更接近底层。在储存播放时还需要自行编码和译码，十分麻烦，但相应的更灵活，可以有多种存储格式，能够获取的录音信息也更多。在用MediaRecorder进行录制音视频时，最终还是会创建AudioRecord用来与AudioFlinger进行交互。

MediaRecorder的状态图如下：



在具体使用中，由于我们对于储存格式等没有过多的要求，因此我们采用MediaRecorder即可。当开始录制按钮按下后，我们设置麦克风为录音声源，设置输出文件的格式为MPEG\_4，设置声音的编码为AAC。然后获取文件输出路径后设置为MediaRecorder的输出路径。接下来只需要调用MediaRecorder.start();即可开始录制，当按钮结束录制被按下后，我们调用MediaRecorder.stop();来结束录音，此时若发生异常，对应的文件若存在应当删去。

#### 2.4 录音时长更新功能设计

使用Handle类和Runnable类实现定时器功能，累加时间然后更新Textview。每次点击开始按钮时将时分秒归0，并且打开定时器。点击结束按钮时关闭定时器，也即调用mHandler\_time.removeCallbacks(runnable\_time)。



#### 2.5 录音音量显示功能设计

首先由于音量时动态显示，因此没有对应的控件可以实现，除了比较丑的seekbar。因此需要自定义一个view—AudioShow。这个控件需要实现的很简单，首先时一般必须有的几个构造函数，在原构造函数基础上加上paint的初始化定义。然后需要一个onLayout在布局变化的时候的回调函数。在该函数中我们获取控件的宽和高。接下来最重要的就是Ondraw函数。它时整个view显示的东西的关键。这里我们画一个半透明的圆角矩形，在根据音量分贝大小画一个实心的矩形覆盖在半透明矩形的底部。

同时由于音量时实时动态显示，因此也使用使用Handle类和Runnable类实现定时器功能。每0.05s调用updateMicStatus();更新一下分贝大小，该函数内部调用MediaRecorder.getMaxAmplitude()获取音量后计算分贝大小。然而分贝的更新和录音录制在recordActivity，onDraw在AudioShow内，二者需要类间信息交流，因此AudioShow类提供一个接口setDB(db);来更新其内的分贝值。因此定时器中也需要调用该函数传递分贝值。此外，为了能定时重绘该view，定时器内需要调用invalidate();标注该view的布局已失效，系统就会自动调用onLayout和onDraw函数，重绘view。



#### 2.6 录后即焚功能设计

这个部分非常简单，就是给了一个checkBox，如果check了，按结束按钮时就会自动删掉已保存的文件，调用file.delect()即可，执行完毕后输出相应的消息。



#### 2.7 基于数据库的recycleView功能设计

这个部分比较复杂，首先我们采用了数据库作为recycleView的数据来源，然后建立了holder和adapter来绑定数据。在holder中为每一个Item的控件进行初始化和响应函数设置。

在adapter中获取数据库对象并读取所有项，用这些项初始化holder。并提供clicklistener的 接口，用来传递点击Item时的响应函数。

在recordActivity中为数据库添加项，在recycleView中删除按钮响应函数中删除项。

通过上面这些步骤，我们成功建立了一个可以显示历史录音记录的recycleView，并且支持删除功能，同时点击Item可以跳转播放。

#### 2.8 播放功能设计

采用MediaPlayer类进行播放，在intent切换页面时收到播放文件路径，作为MediaPlayer的输入源。然后通过Float Action Button控制播放暂停，为了保密期间，并未使用后台服务，而是仅存在于当前页面上播放。利用线程定时更新现有播放状态，并且采用seekbar关联播放，可以通过seekbar控制播放。



#### 2.9 阅后即焚功能设计

这个部分非常简单，就是给了一个checkBox，如果check了，退出界面时，调用file.delect()即可。但是没有删数据库，因此历史录音界面仍显示条目。但是点击进去播放后会显示文件已删除。



### 三．关键数据结构/算法

#### 3.1 数据库

数据库包含三个部分：Dao、Database class、Entity。分别。是数据库操作、数据库类、数据库的项。

在Database class中我们定义了inst函数来获取一个符合Entity定义的数据库，如果没有就创建一个数据库并返回。

在Dao中，我们定义了以下四种操作：

@Query(**"SELECT \* FROM todo"**)  
List<ShowListEntity> loadAll();  
  
@Query(**"DELETE FROM todo"**)  
**void** deleteAll();  
  
@Insert  
**long** addTodo(ShowListEntity entity);  
  
@Delete()  
**void** deleteTodo(ShowListEntity entity);

它们分别是：获取所有数据库内的项，以便于后续Adapter使用获取数据。删除数据库内所有的数据项，暂时还没有用到过。向数据库中加入一个项。在数据库中删除一个项。

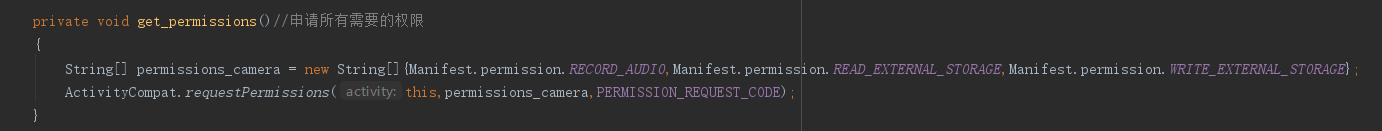
在Entity中，我们定义了主键mID、filename、filepath三个Colum如下：

@PrimaryKey(autoGenerate = **true**)  
@ColumnInfo(name = **"id"**)  
**private** Long **mId**;  
  
@ColumnInfo(name = **"filepath"**)  
**private** String **filepath**;  
  
@ColumnInfo(name = **"filename"**)  
**private** String **filename**;

同时我们还为这些数据项分别定义了get和set函数来修改它们的值。

#### 3.2 MediaRecorder类实现录音和播放

录制音频需要访问麦克风设备，访问设备需要添加权限，存储音频需要存储权限：

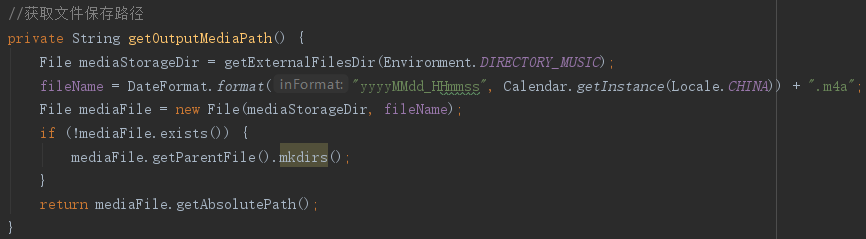


MediaRecorder录制音频的步骤如下：

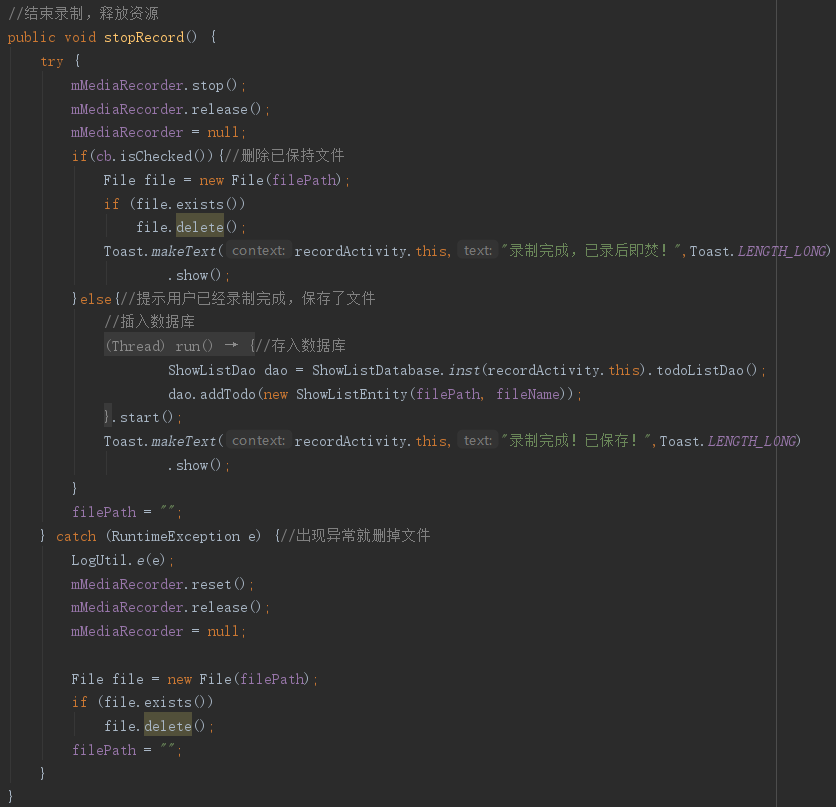
1. 调用new MediaRecorder()构造函数得到MediaRecorder的实例。
2. 调用setOutputFormat()设定媒体文件的输出格式。
3. 调用setAudioSource()设定音频的录入源以及调用setAudioEncoder()设定音频的编码方式。



1. 调用setOutputFile()设定记录的媒体文件保存的路径。



1. 先调用prepare()准备录制，准备完成之后调用start()开始录制。
2. 记录完成后，调用stop()停止录制。



#### 3.3 Seekbar拖动条

SeekBar常用的属性如下：

***android:max****="100" //滑动条的最大值*

***android:progress****="60" //滑动条的当前值*

***android:secondaryProgress****="70" //二级滑动条的进度*

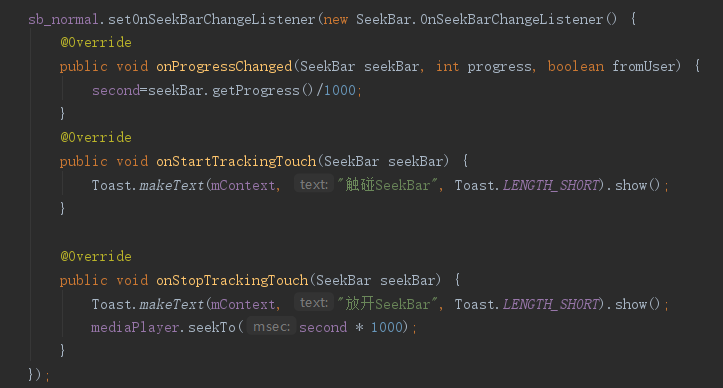
***android:thumb****= "@mipmap/sb\_icon" //滑块的drawable*

SeekBar的事件SeekBar.OnSeekBarChangeListener 我们只需重写三个对应的方法：

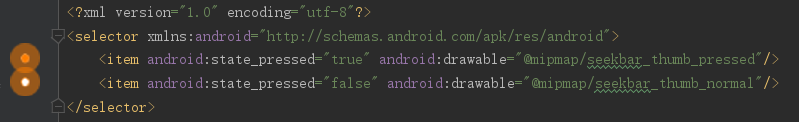
***onProgressChanged****：进度发生改变时会触发*

***onStartTrackingTouch****：按住SeekBar时会触发*

***onStopTrackingTouch****：放开SeekBar时触发*



拖动滑块时检测滑块状态，在Drawable的sb\_thumb.xml中定义如下：



要实现滑块的左右已播放和未播放的进度条颜色不同，已播放的是绿色，未播放的是黄色，用到一个layer-list的drawable资源，其实就是层叠图片，依次是:背景，二级进度条，当前进度。



Drawable：sb\_bar.xml：

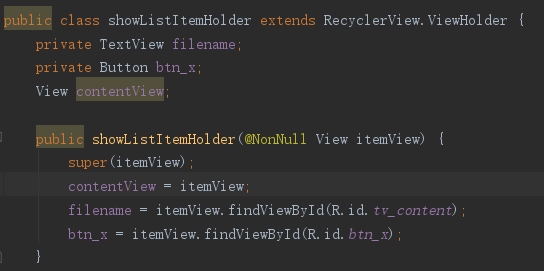


然后布局引入SeekBar后，设置下progressDrawable与thumb即可。



#### 3.4 RecycleView展示列表

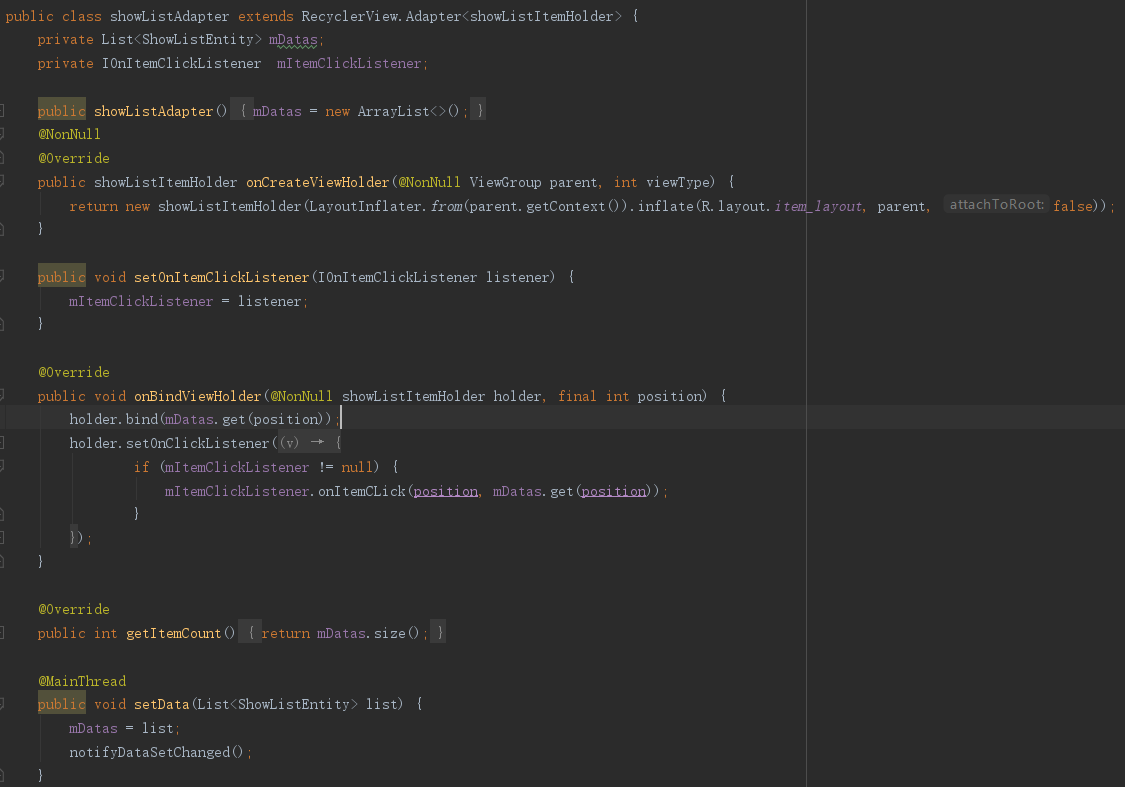
新建定义item（列表子项）的布局文件，把showListItemHolder类写在activity类里面。



设计一个适配器Adapter继承自RecyclerView.Adapter<showListItemHolder>，ShowListAdapter中持有它自己的数据列表，需要实现3个方法：

1. onCreateViewHolder方法，要求返回VH对象，这里就是创建VH对象并返回。而VH构造器要求传入一个View，我们利用LayoutInflater创建一个view给它。当然，创建的根据就是前面设计好的item。
2. onBindViewHolder是把数据交给对应的VH来显示。
3. getItemCount方法要求返回数据的数量。

设置RecyclerView，在Activity的onCreate方法里进行初始化。RecyclerView需要两项设置，适配器与LayoutInflater。



### 四．开发困难及解决方案

#### 4.1 获取行之有效的文件路径

获取一个行之有效的文件路径并非很容易的事情，一开始使用getFileDir()试图直接使用内部目录，后来发现该函数返回的路径在手机中已经不存在了

解决：

1. 尝试新建文件目录，但是或许时因为涉及根目录失败了。
2. 之后还是使用了Environment.DIRECTORY\_MUSIC，这个宏对应的目录路径基本上都是已存在的，但是仍然没有存入。
3. 后来发现可能还需要调用mkdirs()新建一下子目录，之后就终于解决啦！

#### 4.2 自定义录音音量显示的View

刚开想直接用NEW🡪UI Component🡪Custom View新建一个模板然后直接改onDraw。结果app都打不开了，新建的类也十分复杂难懂。

解决：自己新建了一个空类继承自View，然后回顾课件+百度研究了view的生命周期和一些常见自定义view的代码，研究出我这种音量显示控件究竟需要哪些生命周期函数，哪些函数是一个view必不可少的，然后自己写了一个view，就可以正常打开app看见这个控件啦！

之后在交换分贝值的时候又被困住了，因为这个view的构造函数和里面的生命周期函数都是系统自动调用，没机会让我传值。

于是很蠢的把这个view做成record Activity的内部类。传值时不用传了。但是布局文件XML也调用不到这个view了。

这个时候才突然发现自己忽视了可以在类内加一个公共函数让其他类调用更新值，几经周折发现解决方法时真的简单。

#### 4.3 RecycleView的设计实现

RecycleView确实是比较复杂，实现一个recycleView至少需要数据源、adapter、holder、item的xml文件这四个部分。而此处采用的数据库作为数据源有加大了困难。因为数据库至少需要一个Dao文件、一个database文件、一个Entity文件。林林总总需要七大部件，而任何一个部件出问题一般都会在页面一打开就退出应用。

因此刚开始设计的时候光是搭建好几个部件就花费了大量的时间精力。而后运行时更是一直闪退。后来就学会了看debug信息。发现若干错误，比如数据库操作必须要在子线程里面执行、没有添加依赖库等等。然后一个一个的修改更正后非常艰难的完成了recycleView。

但是为item设计点击响应函数的时候又卡住了，不知道到底给哪个控件设计响应函数。后来查到直接给holder构造函数传进来的view设置响应函数就好。

总的来说，recycleView的设计是蛮艰难的。

### 五．分工

### 六．总结

在这次的APP制作中，我们发现其实Android平台上app的实现都不算太难，只要用心找资料学习，一般都已经有对应接口提供，但是要实现的精致细化，具有特色就很难了。例如音量显示的自定义view，这只是很简单的一个view，但是也让我们花费了很多精力，如果要更具特色，就需要更多自定义view，显然会很困难。同时对于app的icon、app界面美观的调整也很费时间。

同时我们发现创意更是一个艰难的点，我们多次讨论后提出的创意也就是树洞app而已，在概念上确实比较有创意了，市面上基本没有同类型产品。但是具体到功能上，就没有多大的创新了，和市面上很大产品都有交叉。其实如果真的有好的功能创新提出来，在安卓这个非常成熟的平台上实现一般来说都不会太困难，但是想要有这么一个创新就很难了。

基于本次的创意app实践，我们一方面实践了我们学习到的安卓开发知识，另一方面感受到了创新创业的历程和艰难。同时也发现编程万变不离其宗，尽管安卓有众多接口、控件、库函数，可以方便我们的编程，但是又什么精细化的操作、自定义的操作，依旧还是要靠自己一行一行制作出一个个类，一个个函数才行。