

Design and Implementation of Sound Management Software Based on Android Platform

Linhua Song, Jinxin Hai, Boxuan Dong, Jie Sun, Qiancheng Fu

Institute of Computer, North China University of Technology, Beijing
Email: hysen0101@163.com

Received: Nov. 28th, 2017; accepted: Dec. 12th, 2017; published: Dec. 19th, 2017

Abstract

Mobile phones have become an essential communication tool, but the phone's ring tone is always constantly changing between the ring and the mute, which takes time and sometimes has many troubles and inconveniences caused by forgetting to close the mute setting. This paper solved this problem by using JAVA + JDK technology, SQLite database and Android technology to develop a sound management mobile phone software.

Keywords

Mobile Sound, Timing Mute, Ring Settings, SQLite

基于Android平台音效管理软件的设计与实现

宋丽华, 海金鑫, 董博轩, 孙 捷, 傅千乘

北方工业大学计算机学院, 北京
Email: hysen0101@163.com

收稿日期: 2017年11月28日; 录用日期: 2017年12月12日; 发布日期: 2017年12月19日

摘 要

手机已经成为人们必不可少的通讯工具,但是经常对手机的来电铃声模式在响铃和静音之间不断地转换,这不仅花费时间,而且有时候因为忘记关闭静音设置带来诸多的麻烦和不便。本文通过JAVA + JDK技术和SQLite数据库及Android技术开发一款音效管理手机软件解决这一问题。

关键词

手机音效, 定时静音, 铃声设置, SQLite

Copyright © 2017 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

据工信部统计 2017 年上半年中国手机用户数量达到了 11 亿, 另外随着智能机的普及, 手机已经成为人们日常生活的必备工具。但是在一些特定应用环境下, 人们需要对手机的信息提示音进行特别的设置, 比如上课时需要设置为静音模式等, 然而这些需求也常常带来一些不便。人们往往因为繁忙忘记将手机提示的静音关闭, 错过一些重要的信息或者电话, 或者因为忘记设置静音而尴尬, 另外, 有时我们也需要为个别的联系人设置不同的来电铃声, 然而现在的智能手机缺少一款这样的统一音效管理应用。

本文基于以上背景, 又因在目前的手机操作系统中, Android 操作系统仍占有最大的比例, 所以设计开发了一款基于 Android 平台[1] [2]的手机音效管理软件, 以解决手机音效带来的不便。该软件包含了自定义静音、整时静音、教学静音、静音白名单、来电铃声设置五大功能。

2. 设计与实现

2.1. 结构设计目标

客户端(界面)使用 Android Studio [3]和 java JDK 工具开发, 主要展示用户的各种操作和操作结果的界面, 为用户提供简洁明了的界面操作, 例如添加、修改、删除具体任务以及模式选择等各种操作。

后台服务使用 Android 的广播信息机制和 Server 组件对相应的用户请求和各时间进行处理, 并使用开源数据库框架 LitePal [4]对 SQLite 数据库[5] [6]进行管理和操作。保存用户输入的数据和一些系统文件数据。前台和后台使用逻辑语言控制, 使两者进行相应的交互。

根据对结构和需求的分析[7], 软件的开发目标如下:

- 1) 用户可以按照需求选择不同的静音设置模式。
- 2) 不同模式中可以按照模式功能设置相应的静音任务。
- 3) 在当前静音任务结束时, 如果没有其他的任务, 应使手机回复原来的状态。
- 4) 对一个任务可以暂时的关闭和开启。
- 5) 静音模式下设置白名单。
- 6) 不同的联系人可以设置不同的来电铃声。

软件的功能结构图如图 1 所示。

根据系统的设计和函数分析, 音效管理软件的层次结构如图 2 所示。

2.2. 软件功能

- 1) 自定义定时静音

该模式下, 用户可以设置周一至周日中的一天或者几天的任意时间段的定时静音, 为每一个任务设置不同的任务标签, 静音模式可以为静音或振动。保存的任务以列表的形式显示出来, 每一个任务列表

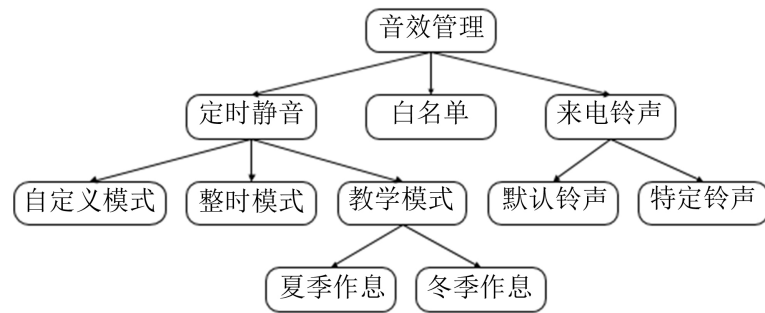


Figure 1. Functional structure
图 1. 功能结构图

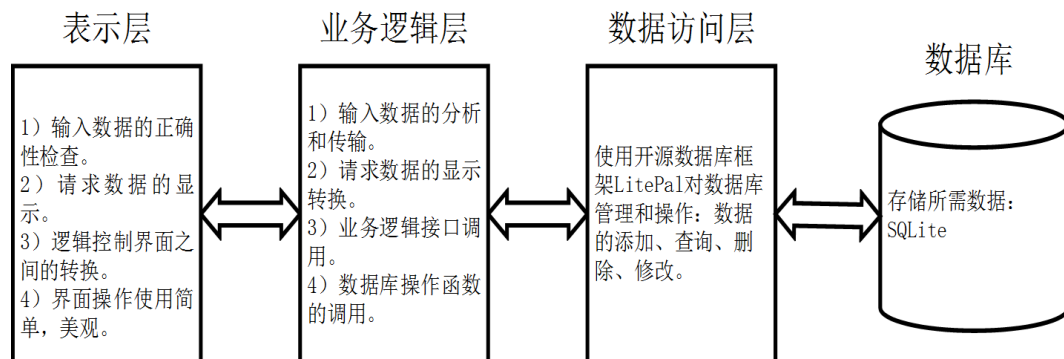


Figure 2. System hierarchy diagram
图 2. 系统层次结构图

都带有一个开关以方便用户对其随时关闭和开启，避免经常的删除和添加同一个任务。

2) 整时定时静音

整时静音模式是将一天的二十四小时按照零点至六点为一个时间段，六点至二十四点每隔一小时为一个时间段，时间段的开始时间和结束时间已经设置好，用户只需选择相应的时间段即可，虽然限制了时间的设置，但是对于添加这种任务的用户来说更加的方便。

3) 教学定时静音

教学模式顾名思义是按照教学课表时间划分一天的时间段的，每一堂课的时间就是一个定时任务的时间长度。该模式分为夏令作息和冬令作息两个模块，添加课程名即添加静音任务，同时静音列表同时也是一张课程表。

4) 静音白名单

白名单的设置是为了避免在静音状态下错过重要的来电，为此在设置静音之前可以将认为重要的联系人的号码添加到白名单中，名单中的联系人来电时将会继续响铃而不影响静音任务的正常执行。

5) 来电铃声设置

来电铃声的功能和一般手机系统的铃声设置不同的是可以为每一个联系人设置不同的来电铃声且可以不同于系统设置。这种方式也可以作为一种铃声式的来电号码显示，根据来电铃声判别联系人。

2.3. 功能实现

2.3.1. 客户端

系统客户端是用户唯一接触到的部分，所以界面的布局设计和操作性能直接影响用户对软件的使用。简洁明确的界面布局要求功能层次之间的逻辑清晰和功能之间的切换流畅，文字与图标相结合是界面更

加的丰富。操作流畅在于开发技术和数据逻辑的使用，良好的技术和清晰的逻辑影响着数据的传输和操作性能。

据以上性能可以得到客户端的结构布局如图 3 所示。

2.3.2. 后台服务

后台服务是整个系统正常运行的保障，包含了部分功能的实现和整个系统的数据管理。后台服务保障的这部分功能需要一直运行，不管用户正在操作软件还是软件处于后台运行的状态。而其他部分功能只有在用户操作软件时才会自动启动，所以并不需要后台服务的支持。在用户操作软件时和软件处于后台运行状态时都需要访问数据库，所以后台服务必须全程保障对数据的管理功能。

部分功能运行的保障。如定时静音功能，在系统不是人为结束时，必须保证某些功能的长期运行，需要 Android 组件[8]中的服务 Service 和进程守护的结合；其中的来电铃声功能需要接受到手机的广播信息才可以使用，需要 Android 组件中的广播接收器 BroadcastReceiver 和 Service 相结合。

管理系统数据。客户端与数据库之间数据传输需要借助后台服务完成交互，同时对客户端传输的数据进行逻辑分析处理，使用 Intent 完成界面之间数据的传输；SQLite 数据库数据使用开源数据框架 LitePal 管理，将数据库的操作转换为函数接口的调用，数据的操作不用任何的 SQL 语句，只需要调用相关功能的函数接口就行，保证数据的安全性和正确性。对一些过期的数据做合理化的处理，避免无用数据长期占用内存空间影响系统运行速度。

由以上叙述得后台服务的结构布局如图 4 所示。

2.4. 关键技术与算法

2.4.1. 客户端

客户端的界面布局因功能不同在结构上也不相同，所以整体的逻辑结构也不相同，客户端的数据接收和数据显示以及数据的传输和正确性的保证是关键技术。

1) 布局结构的重用

不同界面中的一些相似或者相同的布局可以提取出来形成一个布局文件，需要时只需嵌入布局文件即可，减少代码的重复率并提高编码质量。

2) 数据的正确性

用户在客户端输入数据时不会考虑数据的正确性，所以对输入数据的查错和规格有所处理，并对用

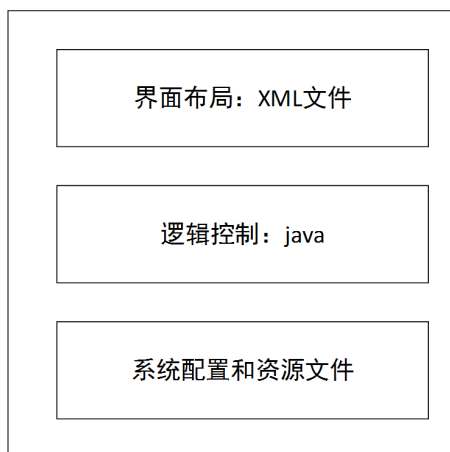


Figure 3. Structure layout of client

图 3. 客户端结构布局图

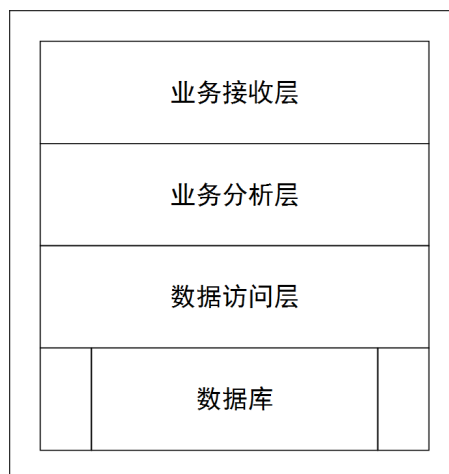


Figure 4. System hierarchy diagram

图 4. 系统层次结构图

户做出相应的提示，减少用户输入错误数据的次数，提高使用效率。

为解决界面中显示的中文字符在程序编码中不易识别且容易出差的情况，将所有使用到的中文字符统一写入 `string.xml` 文件中，使用不同的变量代替不同的中文含义，在程序中只需调用相应的字符变量即可，同时多次使用的字符不需要重复定义，只需简单的调用变量名即可，这将保证了部分字符无法识别而造成系统错误。

3) 不同界面之间的数据传输

不同界面的数据传输使用 `Intent` 完成，传输界面调用 `Intent` 的 `putExtra()` 函数接口发送要传输的数据，显示界面调用 `Intent` 的 `getTypeExtra(String,...)` 函数接口接收要显示的数据(其中 `Type` 表示接受的数据类型，可为 `String`、`Int` 等数据类型的关键字)。例如查看或修改定时任务时需要传输全部或部分数据至不同的界面以便显示，联系人来电铃声信息的查看。

4) 界面和数据库之间数据传输

a) 界面向程序传输数据

用户在界面数据数据时不知道数据的正确性，所以对数据的数据要验证和纠错并以是用户输入符合规则的数据。程序内部的数据都有一定的数据结构，因此对输入的数据必须要进行处理，转换为程序可以正确识别的数据格式。

b) 程序将数据显示在界面

程序中的数据都是按照相应的规则存储的，因此不需要做纠错处理，只需将数据内容绑定相应的适配器，调用 `setAdapter()` 把显示的数据加载到界面。

2.4.2. 后台服务

后台服务包含了业务接受、业务分析、数据访问不同层次，调用下层，向上次提供服务是后台服务的逻辑结构每一层相互独立有相互联系，保证相互之间的协调合作，提高层内部的内聚，降低层之间的耦合，确保系统功能的正常运行。

各层功能运行的主要技术支持和算法介绍如下：

1、业务接收层

业务接收层主要接收本地和手机系统发送的信息并发送给业务分析层处理。

本地发送的信息主要来自 `Service` 的自启动，为了保证定时静音功能能够根据数据的变化改变任务的

设定，需要不断的扫描数据库的数据，为此在系统启动的时候启动一个服务 Service，并且给改服务设定一个自启动的定时器。除非用户手动结束进程或者手机系统达到一定的运行限制时自动结束。

通过 Android 的广播信息机制，并在程序中注册想要接受的信息类型就可以接受到需要的信息。本软件系统主要接受手机的开机广播信息和手机电话的状态。广播的注册分为动态注册和静态注册，一般使用静态注册，一些特殊的广播只能使用动态注册。本系统的开机和电话广播使用静态注册，在配置文件 AndroidManifest.xml 中的<intent-filter>标签中添加需要的 action。

2、业务分析层

接收上一层发送的数据并分析处理。通过接收的信息不同，触发不同的服务功能。接收的本地信息就做定时静音任务的处理，接收的电话广播信息就做来电铃声设置和白名单的处理。信息的处理及传递主要涉及以下技术：

(1) 进程守护

为了避免系统服务的中断导致某些功能无法正常使用，使用 AIDL 通信机制将两个 Service 相互连接绑定，达到进程守护的效果。本系统中分别为本地服务 LocalService 和远程服务 RemoteService，两者通过 AIDL 相互通信，在软件系统启动时将两个服务同时启动，运行中一旦有一方中断，另一方将会受到信息并将其重行启动，保证永远有一个服务运行。进程守护的通信关系如图 5 所示。

(2) Android 多线程[9]

系统中的代码都是默认在程序的主线程中执行的，而这种机制就会出现因为执行的任务逻辑复杂导致出现主线程被阻塞的情况，通常会出现 ANR 问题，所以定时静音任务的设定和更新使用多线程完成，为了简单方便，本系统使用匿名类的方法：

```

New Thread(new Runnable){
    Public void run(){
        静音任务处理;
    }
}).start();

```

(3) 进程等级提升

手机系统中的进程等级决定了在系统资源紧张的情况下，释放进程顺序，想要系统一直运行就要尽可能的避免被销毁释放，就要提高进程等级。服务是一直运行在后台的，并且它的系统优先级是比较低的，利用 Notification 类创建一个显示通知就可以将服务的等级提升为前台服务。

3、数据访问层

系统所有功能的实现最终都转化为数据信息的交互，数据信息成功交互是功能完成的提前和标志。本系统的数据都存放在本地数据库中，为了提高对数据的操作效率，使用开源的 Android 数据库框架

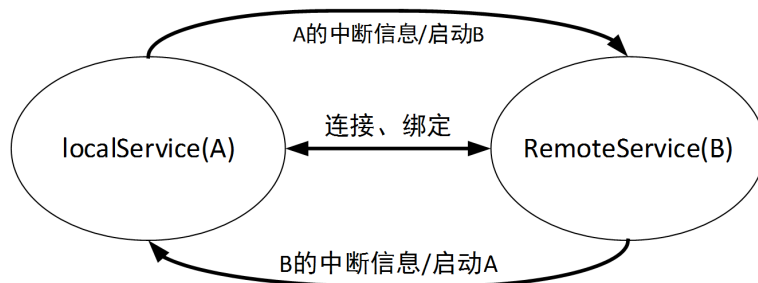


Figure 5. Daemon relation
图 5. 进程守护关系图

LitePal。采用关系映射的模式将数据库的操作进行封装，创建一个继承 DataSupport 类的类即可创建数据表，类的数据成员代表数据表的字段，每一个数据字段的访问都是使用类的对象对成员函数的调用。创建 litepal.xml 文件并添加数据库的名称和版本等信息，添加类名即添加数据表。

4、算法设计

算法在时间上的耗费要降到最低，并且在逻辑上清晰，对每一种可能出现的错误都要有相对应的处理，不能出现程序进入死循环的流程，每前进一步检测是否有继续执行的必要，不需要则即可结束当前的执行，减少无用功对资源的耗费。定时静音服务和来电监听服务的算法流程图如图 6 和图 7 所示。

从图 6 的定时静音监听算法流程图中可知，软件开始运行时访问数据库读取开启的任务信息，对每一条任务的各项信息逐步验证，只要有任一项与系统信息不匹配就中止后续的信息检查，然后继续下一条任务的各项信息检查；对于周期为一次的任务，首次检查出信息不匹配时，将其任务状态修改为关闭，避免系统再次访问读取无用任务信息。这两种方式都为了减少对数据库的访问时间，提高系统运行效率，增加用户的体验。

从图 7 的来电监听算法流程图中可知，检测到手机来电状态改变的信息后，只对来电信息做处理。来电状态时，利用手机来电后与铃声响起之间的时间间隔，通过手机号码访问数据库中的白名单信息表和来电铃声信息表，检查到信息后启动白名单服务或铃声设置服务，服务中记录当天系统信息并设置新的信息；接听和挂断状态时，因为来电状态肯定在这两种状态之前发生，所以不用再次访问数据库，只需要根据来电状态记录的系统信息恢复设置信息。

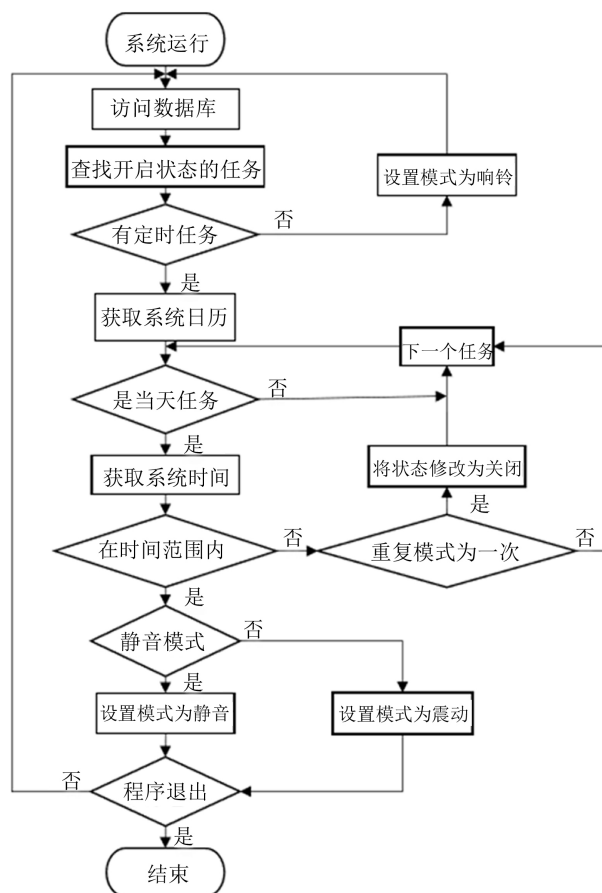


Figure 6. Timing muting algorithm flow chart
图 6. 定时静音算法流程图

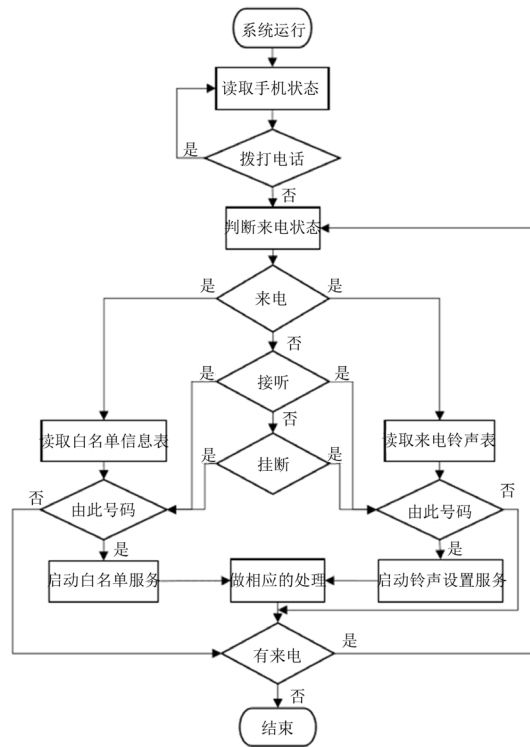


Figure 7. Calls to monitor algorithm flow chart
图 7. 来电监听算法流程图

3. 测试

3.1. 测试环境介绍

本系统的功能使用 Android 模拟器和安卓真机两种方式测试验证。其中 Android 模拟器只测试定时静音功能，真机测试全部功能。

1、Android 模拟器

(1) 测试环境搭建：Android Studio2.2，Android SDK，java jdk7。

(2) 模拟器环境：

测试使用的安卓模拟器的基本参数配置如表 1 所示。

2、安卓真机

按照本系统支持的最低版本，将软件安装在不同版本 Android 操作系统的手机上，针对不同版本的手机做相同功能测试。测试真机的基本参数如表 2 所示。

3.2. 测试策略

测试时，将生成的软件安装包安装在测试机上，正常打开软件。测试中需要验证：

- 1) 不同模式的定时静音是否可以正常设置。
- 2) 静音时白名单是否作用。
- 3) 联系人来电铃声设置是否成功。
- 4) 需要权限的功能在相应权限禁止时是否仍然能够运行。
- 5) 任务信息在设备屏幕上显示是否正常。

- 6) 软件运行中是否具有前台服务(下拉通知栏是否正常显示)。
- 7) 后台服务的双进程守护的通信是否良好。
- 8) 客户端与数据库数据交互是否正常。
- 9) 软件是否接受已注册的系统广播。
- 10) 系统接受广播后是否可以启动后台服务的相关功能。

3.3. 测试结果

3.3.1. 功能验证

在测试机正常使用，并且可以正常通信下，测试结果如表 3 所示。

说明：“√”表示功能验证通过，“—”表示未验证。因模拟器无法接受手机来电，所以白名单和来电铃声功能不能再模拟器验证，只能在真机上验证。

分析表 3 的测试结果表明，按照测试策略的要求，在模拟器和真机上对各功能的验证都达到系统的要求，所有功能全部实现。

部分可视功能的验证结果如下：

Table 1. Simulator Parameters Configuration

表 1. 模拟器参数配置表

模拟器	API	CPU/ABI	Android 版本	内存大小
1	21	x86	Android 5.0	1 GB
2	23	x86_64	Android 6.0	2 GB
3	24	X86	Android 7.0	2 GB

Table 2. Testing machine parameters

表 2. 测试机参数表

	测试机 1	测试机 1	测试机 3
Android 版本	Android 5.0	Android 6.0	Android 7.0
处理器	四核 1.2 GHz	八核最高 2.0 GHz	八核最高 2.3 GHz
运行内存	2Gigabytes	3Gigabytes	4Gigabytes
内核版本	3.10.28-svn3245	3.10.94 android@localhost #1 Fri Jan 20 03:20:10 CST 2017	4.1.18-g09ffadf android@localhost #1 Wed Sep 13 02:13:29 CST 2017

Table 3. Test results of system function

表 3. 系统功能测试结果

功能	自定义模式	整时模式	教学模式	白名单设置	来电铃声设置
模拟器 1	√	√	√	—	—
模拟器 2	√	√	√	—	—
模拟器 3	√	√	√	—	—
测试机 1	√	√	√	√	√
测试机 2	√	√	√	√	√
测试机 3	√	√	√	√	√

1) 自定义定时静音模式, 设定时间段 06:09~06:15, 重复周期为“定时一次”, 静音模式为静音, 任务标签为 test。当前系统时间在静音时间段外和时段内的结果如图 8(a)和图 8(b)所示, 状态栏的“☆”图标为静音标志。

2) 整时定时静音模式, 按照图 9(a)所示的信息设置静音任务, 图 9(b), 图 9(c)为设置结果和系统时间。

从结果图中可以看出当前的定时静音任务的各项设置和系统时间系统, 所以当天手机模式改变, 图 (b)中的振动图标表明任务设置成功。

3) 教学定时静音模式, 选择夏令作息测试。按照图 10(a)所示的信息设置静音任务, 图 10(b)和图 10(c)

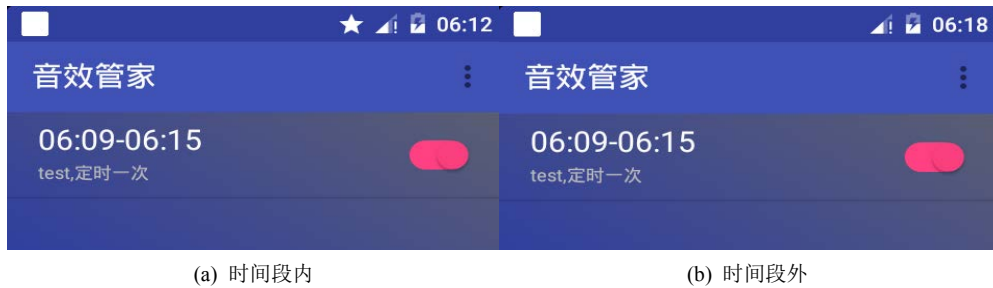


Figure 8. Custom timed mute test result
图 8. 自定义定时静音测试结果



Figure 9. Whole time Mute test result
图 9. 整时定时静音测试结果



Figure 10. Teaching time mute test result
图 10. 教学定时静音测试结果

分别为不同时间段的手机状态。

从结果图(b)中可以看出在任务时间段内,手机状态更改为任务设置的状态(状态栏的“☆”图标),功能测试正常。

3.3.2. 性能分析

测试过程中,在不同版本的 Android 系统测试机上验证了所有的功能。数据输入可控,界面数据显示正常;操作中界面切换和功能使用流畅;后台服务常驻正常,除人为的释放资源外,可以保证日常的使用;前台服务运行正常;软件运行之后占用的内存空间最多几十兆。

总体而言,软件系统的性能良好,可作为一般性的使用。

4. 结束语

本文系统实现了基于 Android 操作系统的音效管家,包括手机定时静音、静音白名单、来电铃声等功能。系统数据的存储使用轻量级的嵌入式关系模式数据库 SQLite,运算速度快且占用系统资源少,减少数据库访问延迟;数据的操作使用开源的 Android 数据库框架 LitePal,使用类的对象映射模式对传输数据进行封装;在不同厂商不同版本的手机上测试,软件运行良好。相对同类相似软件,本系统将手机几种不同的功能结合使用,相当于其他几款软件的结合,使用方便,适用于大多数人群;系统维护简单。

基金项目

北方工业大学学生科技活动项目,北方工业大学优秀青年教师培养计划项目。

参考文献 (References)

- [1] 张磊. 基于 android 平台的应用开发研究[J]. 计算机光盘软件与应用, 2011(17): 211.
- [2] 黄永丽, 王晓, 孔美云. Android 应用开发完全学习手册[M]. 北京: 清华大学出版社, 2015.
- [3] 李宁宁. 基于 Android Studio 的应用程序开发教程[M]. 北京: 电子工业出版社, 2016.
- [4] GitHub.LitePalFramework/LitePal [EB/OL]. <https://github.com/LitePalFramework/LitePal>, 2017.
- [5] 韩善峰, 曹凤海, 易昌华. SQLite 数据库在嵌入式程序开发中的应用[J]. 物探装备, 2011, 21(3): 170-178.
- [6] 张恒喜, 史争军. 基于 SQLite 的 Android 数据库编程[J]. 电脑编程技巧与维护, 2011(21).
- [7] Shari Lawrence Pfleeger. Software Engineering Theory and Practice. Beijing: Higher Education Press, 2001.
- [8] 姚昱昱, 刘卫国. Android 的架构与应用开发研究[J]. 计算机系统应用, 2008(11): 110-112.
- [9] 张海燕. JAVA 多线程技术在手机互联网中的应用[J]. 农业网络信息, 2008(3): 97-99.

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2325-2286, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: sea@hanspub.org