

# 手机卫士的设计与实现

吴朝霞, 郭 强

(安阳工学院 计算机学院, 河南 安阳 455000)

**摘要:** 智能手机出现使得生活有了更多的便利,但是手机的反应速度、安全、以及如何管理手机等问题,也给非专业人士带来了许多烦恼。因此,文章给出了基于安卓系统手机卫士的实现方法,并具体讨论了手机防盗、通讯卫士、软件管理、进程管理、手机杀毒、缓存清理的实现过程。

**关键词:** 智能手机;手机卫士;应用开发

**中图分类号:** TP311.56

**文献标识码:** A

**文章编号:** 1673-1131(2017)04-0109-02

## 0 引言

安卓手机操作系统在快速发展,基于安卓系统的各类软件数量也在飞速增加。大部分软件给用户带来便利,然而这些软件中也包含小部分为病毒和流氓软件,造成了用户隐私、财产受到不法侵害。所以手机智能化背景下,短信拦截、加入黑名单、避免骚扰等简单的功能已经远远不能满足用户的需求,手机的进程管理,病毒查杀,防盗追踪,通讯录在线云备份、手机远程防盗及远程数据删除等功能也是必不可少的。一款好的手机安全软件,不仅可以维护用户隐私,还可以避免用户的财产损失,极大地提高了手机的安全性,该软件体验的好坏直接影响着用户的安全和使用者的心情。

## 1 总体设计

针对智能手机使用过程中出现的各类问题,本设计为满足用户的需求,方便用户的日常生活,将手机卫士规划为手机防盗、通讯卫士、软件管理、手机杀毒、缓存清理、设置等几个模块。利用软件实现的功能,在手机遗失, SIM 卡变更的情况下,可以进行手机的远程定位,报警和数据销毁;对于目前日益增多的垃圾短信,电话等进行骚扰拦截;同时也可以查杀恶意软件、木马蠕虫病毒等,防治个人信息的泄露。对于后台不常用进程进行清理,长期缓存的大量数据查找清除,对进程进行优化管理。

采用的开发工具为:集成 Android SDK 的 eclipse, SQLite 数据库,并利用 MVC 框架进行设计和开发。在后续文章中仅给出了几个主要模块的实现方法。

## 2 手机安全功能实现

### 2.1 手机防盗模块

操作设置基本流程为程序首先判断是否为第一次使用该软件,如果为第一次,就需要设置安全密码。然后通过重复输入确认所输密码一致。提交后密码使用 MD5 加密保存,提高密码安全性。然后将密码和手机卡绑定,安全号码设置。

一旦出现手机被盜行为,处于防盗保护中的手机可以通过发送短信来远程控制手机,进而达到手机防盗的功能。比如#location#获取手机位置,##delete\*#恢复出厂设置,发送##lockscreen\*#给丢失的手机加密并锁定屏幕,手机锁屏后,使用者无法做任何操作,在锁屏状态下,手机无法关机,拨打电话,发送信息等等,所有功能全部无效,仅有正确输入解锁密码才可正常使用。发送##alarm\*#播放报警音乐,手机报警音乐会

调至声音最大,并且会一直播放,不能中止。

手机防盗功能的实现设计,主要实现代码如下:

```
if ("##*location*#" .equals(body))
{
    Intent i=new Intent(context,GPSService.class);
    context.startService(i);
    SharedPreferences sp=context.getSharedPreferences("config", Context.MODE_PRIVATE);
    String lastlocation=sp.getString("lastlocation",null);
    if(TextUtils.isEmpty(lastlocation)){
        SmsManager.getDefault().sendTextMessage(sender, null, "
        geting loaction.....", null, null);
    }else{
        Log.i(TAG, lastlocation);
        SmsManager.getDefault().sendTextMessage(sender, null,
        lastlocation, null,null);
    }
    abortBroadcast();// 终止掉广播
} else if ("##*alarm*#" .equals(body)) {
    Log.i(TAG, "播放报警音乐");
    MediaPlayer player=MediaPlayer.create (context,
    R.raw.ylzs);
    player.setLooping(false);
    player.setVolume(1.0f, 1.0f);
    player.start();
    abortBroadcast();
} else if ("##*wipedata*#" .equals(body)) {
    Log.i(TAG, "远程删除数据");
    abortBroadcast();
}
```

### 2.2 通讯卫士模块

通讯卫士模块主要实现了黑名单拦截功能,通过设置电话拦截和短信拦截,来阻止被拦截电话和短信访问本机。该模块主要通过设置电话和短信的黑名单来实现。首先,将黑名单号码,存储到数据库 BackNumberDao 中,显示在页面中的 ListView 上,设置完数据库之后,然后实现黑名单拦截功能和黑名单显示界面的的实现,设置完成后,通过输入黑名单,设置电话或短信拦截,便可以保存至数据库,同时显示在指定的 ListView 界面中,除此之外,通过调用数据库中添加和删除的方法,也实现了对黑名单数的添加和删除功能。这些都设

置完成后,最后实现黑名单拦截功能,主要是通过使用 Service 服务来完成。

### 2.3 手机杀毒模块

手机杀毒模块主要是通过扫描应用程序的包名和数字签名来和数据库里存储的病毒的数字签名作比较,比较结果相同则认为是病毒。一旦手机丢失,通讯录再次一个个添加是很难完成且非常繁琐的事务,远程备份刚好可以为我们做好这样的后勤工作,一键获取之前遗失的通讯录信息。具体操作是通过给丢失手机发送短信,通过后台服务自动获取通讯录,然后联系方式会通过短信形式发送到接受手机。

### 3 手机管理模块

进程管理模块主要通过调用系统方法扫描手机中的系统程序进程和用户程序进程并显示在进程管理界面的 ListView 中,用户点击后会有给出卸载、运行、分享供选择。点击分享可以将其通过短信分享给其他人。其次,读取手机后台运行进程占用内存大小和总内存。以及清理后台内存的功能。主要代码如下:

```
public void killAll(View view) {
    ActivityManager am = (ActivityManager) getSystemService(ACTIVITY_SERVICE);
    int count = 0;
    long savedMem = 0;
    List<TaskInfo> killedTaskinfos = new ArrayList<TaskInfo>();
    for (TaskInfo info : allTaskInfos) {
        if (info.isChecked()) {
            am.killBackgroundProcesses(info.getPackageName());
            if (info.isUserTask()) {
                userTaskInfos.remove(info);
            } else {
                systemTaskInfos.remove(info);
            }
            killedTaskinfos.add(info);
            count++;
            savedMem += info.getMemsize();
        }
    }
    allTaskInfos.removeAll(killedTaskinfos);
    adapter.notifyDataSetChanged();
    processCount=count;
    availMem+=savedMem;
    tv_process_count.setText("运行中的进程:"+processCount+"个");
    tv_mem_info.setText("剩余/总内存:"+Formatter.formatFileSize(this, availMem)+"/"
        +Formatter.formatFileSize(this, totalMem));
}
```

缓存清理模块主要通过调用系统方法扫描手机内各个应用程序的缓存,然后将应用程序加载到缓存清理界面的 ListView 界面上,点击可以跳转系统的程序清理缓存界面,进而清

理缓存,也可以通过系统的垃圾回收机制来实现缓存一键清理功能。

最后,一键清理功能的实现主要是利用 android 系统垃圾回收机制来实现的,通过设置假的缓存,去主动调用起垃圾回收器实现垃圾回收。

```
public void cleanAll(View v) {
    try {
        Method[] methods = PackageManager.class.getDeclaredMethods();
        for (Method method : methods) {
            if ("freeStorageAndNotify".equals(method.getName())) {
                method.invoke(pm, new Object[] { Long.MAX_VALUE,
                    new MyDataObersver() });
            }
        }
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    private class MyDataObersver extends IPackageDataObserver.Stub {
        @Override
        public void onRemoveCompleted(String packageName, boolean succeeded)
            throws RemoteException {
            System.out.println(succeeded);
        }
    }
}
```

### 4 结语

本文给出了一种新的安全管理软件在手机安卓系统上的开发与实现,该设计充分考虑了用户的体验,对程序进行了优化,使其更加流畅。该设计程序占用内存小,没有多余的代码,对手机性能的要求不高。增加了与服务器交互的部分,使用了 Http 协议,使其从一个离线程序跨入了联网型程序。使用 SQLite 数据库、以及 MD5 加密处理,使程序的数据更加安全可靠。

#### 参考文献:

- [1] 杨丰盛.Android 应用开发揭秘.机械工业出版社.
- [2] 王家林,王家俊,王家虎.Android 项目实战-手机安全卫士开发案例解析.电子工业出版社.
- [3] 李刚.疯狂 Android 讲义(第2版)[M].电子工业出版社,2013.
- [4] 雍俊海.Java 程序设计教程(第二版)[M].清华大学出版社,2008.
- [5] 传智播客.http://www.itcast.cn.

作者简介:吴朝霞(1986-),女,河南安阳人,硕士,助教,研究方向为计算智能。