

## PaperPass检测报告简明打印版

### 比对结果（相似度）：

总体：10 %（总体相似度是指本地库和网络库的综合比对结果）

本地库：9 %（本地库相似度是指论文与学术期刊、学位论文数据库的比对结果）

网络库：4 %（网络库相似度是指论文与互联网资源的比对结果）

编号：VIP9AA83E1BB9768ED49

标题：基于Android平台的手机安652408571846319

作者：王浩

长度：11946 字符(不计空格)

时间：2014-5-13 19:54:54

比对库：学术期刊（1990-2013）、学位论文（硕博库1990-2013）、互联网资源

查真伪：<http://www.paperpass.com/check.aspx>

### 相似资源列表（学术期刊、学位论文）：

1. 相似度：1 % 篇名：《基于MTK平台的手机防盗技术研究》

来源：学位论文《上海交通大学》2010 作者：王伟

2. 相似度：1 % 篇名：《Symbian平台手机防盗系统的设计与实现》

来源：学位论文《北京邮电大学 北京邮电大学》2010 作者：高鹏

### 相似资源列表（互联网）：

1. 相似度：2 % 标题：《Ubuntu下搭建Android开发环境 - bingoogol -...》

<http://www.cnblogs.com/bingoogol/p/ubuntu-environment-android.html>

### 全文简明报告：

#### 摘要

随着智能机的诞生，手机终端的上网功能越来越普及，人们也逐渐离不开智能机和3G业务的发展。但是，与此同时网络安全事件也随之走向手机终端，各种网银骗局、软件骗局、中毒事件等等都开始日益影响手机用户的正常生活，与电脑客户端发生的安全问题是有过之而无不及，从而手机安全问题成为了业界人士和手机用户者最为担心的问题[1]。在这个生活节奏日益加快的年代，人们生活中的通讯和娱乐工具也开始被手机终端所占领，而智能手机和网络的不断发展更让手机用户的各种需求逐渐增大[2]。Android是一种开放源代码操作系统，它在各大手机操作系统中迅速获得广泛的市场，可以利用先进的硬件和软件使用本地或服务端的数据。本文主要介绍的是一款在Android平台的手机安全系统设计理念，并给出了具体的系统设计过程和基础界面图，详细讨论了在设计过程中遇到的难点以及解决方法。

关键词：程序锁；手机防盗；Android

## ABSTRACT

With the birth of Internet-enabled smart machines and the increasing popularity of mobile terminals , people have gradually inseparable from the development of intelligent machines and 3G services. But at the same time network security event also will move towards mobile terminals , various online banking scam , software scam , poisoning , etc. It is beginning to increasingly affect the normal life of mobile phone users , and computer security issues occurring is greater client thereby phone industry has become a security issue and most cell phone users who concern[ 1]. In this era of ever-accelerating pace of life , people living in the communication and entertainment tools are also beginning to be occupied by the mobile terminal , and the continuous development of smart phones and mobile phone networks leaving the various needs of users of increases[2]. Android is an open source operating system , which received wide market quickly in the major mobile phone operating system , you can take advantage of advanced hardware and software using a local or server- side data. This paper will describes the design of a mobile security system in the Android platform , and gives specific system design process and basic interface diagram , a detailed discussion of the difficulties encountered in the design process and solutions.

Key Words: Program Lock; Phone Theft; Android

## 目 录

### 第1章 引言1

#### 1.1 研究背景1

#### 1.2 编写目的1

### { 61 % : 第2章 Android的相关技术介绍及分析2 }

#### 2.1 搭建Android开发环境2

##### 2.1.1 搭建Ubuntu系统下Java开发环境2

##### 2.1.2 搭建Ubuntu系统下Android开发环境2

#### 2.2 Android项目目录结构3

#### 2.3 Activity4

##### 2.3.1 Activity生命周期方法4

##### 2.3.2 与其他应用程序交互4

#### 2.4 BroadcastReceiver5

##### 2.4.1 广播接收者的分类5

{ 49 % : 2.4.2 注册广播接收者的两种方式5 }

## 2.5 Service6

### 2.5.1 什么是服务6

### 2.5.2 本地服务6

### 2.5.3 远程服务6

{ 53 % : 第3章 手机安全卫士分析与设计7 }

## 3.1 手机安全卫士需求分析7

{ 58 % : 3.1.1 手机防盗功能需求分析与设计7 }

### 3.1.2 通讯卫士功能需求分析与设计9

{ 48 % : 3.1.3 软件管理功能需求分析与设计9 }

{ 42 % : 3.1.4 进程管理功能需求分析与设计9 }

{ 42 % : 3.1.5 程序锁功能需求分析与设计10 }

### 3.1.6 手机杀毒功能需求分析与设计11

{ 51 % : 3.1.7 系统优化功能需求分析与设计11 }

{ 62 % : 3.1.8 自动升级功能需求分析与设计11 }

{ 48 % : 第4章 手机安全卫士详细设计与实现13 }

## 4.1 欢迎界面实现13

### 4.1.1 检查升级功能实现13

{ 44 % : 4.1.2 升级对话框显示功能实现13 }

### 4.2 应用程序主界面实现13

### 4.3 程序锁实现13

### 4.4 手机防盗实现14

#### 4.5 通讯卫士实现14

#### 4.6 系统设置实现14

### 第5章 结论与展望15

#### 5.1 结论15

#### 5.2 进一步工作的方向15

### 第1章引言

#### 1.1 研究背景

随着智能手机和网络的完美结合，使得智能机的功能越来越强大，浏览网页、网络购物、视频对话都普及到各个手机终端，然而手机平台越广泛，存在的危险就越大，越来越多的安全问题出现在手机的日常运用中。当我们以为只有通过网络才会使手机中毒的时候，没想到智能手机那方便快捷的数据传送方式：WIFI、蓝牙、内存卡等也使我们的手机暴露在病毒感染的危险之中。通讯设备本身所具备的便利的支付收费、支付方式都为不法份子提供了可乘之机[3]。

当人们已经习惯把手机当做一个移动存储设备，作为一个方便携带的移动终端时，风险并不仅来源于手机病毒，也由于3G网络的开放性[4]。当手机不慎丢失或者是在手机数据传送的过程中用户数据很容易会被窃听、复制或篡改。这不仅会面临经济损失，还会遇到甚至更为严重的隐私泄露危险。

#### 1.2 编写目的

本次毕业设计通过编写青蛙卫士，不仅可以熟悉 Android 应用开发过程，Android 框架的工作原理，而且还可以学习计算机的新知识，从而提高自我学习的能力，提升实践能力、撰写文档的能力，并且还可以培养阅读文献资料的能力和习惯， { 41 %：增加自己的创新意识，使自己拥有科学研究的基本能力。 }

{ 59 %：第2章Android的相关技术介绍及分析 }

#### 2.1 搭建Android开发环境

在Android应用程序开发过程中使用的工具软件都是可以在网上免费下载的，本手机安全卫士采用的集成开发环境为安卓官方网站提供的ADT。此外，与开发普通Java程序类似的是在Ubuntu系统下我们需要将SDK的目录添加到系统的环境变量[5]。具体的地址和安装步骤如下：

##### 2.1.1 搭建Ubuntu系统下Java开发环境

到官网下载jdk文件

跳转到下载目录

```
cd /home/bingo/下载
```

解压到安装目录

```
tar -zxvf jdk-7u45-linux-i586.tar.gz -C /home/bingo/install
```

配置环境变量

在/etc/profile.d目录下新建一个叫java.sh的文件

```
sudo vi /etc/profile.d/java.sh
```

在java.sh中加入以下内容

```
export JAVA_HOME=/home/bingo/install/jdk1.7.0_45
```

```
export CLASSPATH=.: $JAVA_HOME/lib/dt.jar: $JAVA_HOME/lib/tools.jar: $CLASSPATH
```

```
export PATH=$JAVA_HOME/bin: $JAVA_HOME/jre/bin: $PATH
```

{ 100 % : 执行以下命令使上面的配置立即生效 }

```
source /etc/profile
```

{ 77 % : 输入以下命令，如果显示java的版本信息说明java环境配置成功 }

```
java -version。
```

## 2.1.2 搭建Ubuntu系统下Android开发环境

到官网下载sdk

跳转到下载目录

```
cd /home/bingo/下载
```

{ 58 % : 解压到安装目录，如果不存在则创建该文件夹 }

```
mkdir /home/bingo/install
```

{ 100 % : 如果没安装解压zip格式的软件，在Ubuntu软件中心中安装7zip和rar }

配置环境变量

在/etc/profile.d目录下新建一个叫android.sh的文件

```
sudo vi /etc/profile.d/android.sh
```

在android.sh中加入以下内容

```
export ANDROID_SDK_ROOT=/home/bingo/install/adt-bundle-linux-x86-20131030/sdk
```

```
export PATH=$ANDROID_SDK_ROOT/tools: $ANDROID_SDK_ROOT/platform-tools: $PATH
```

{ 100 % : 执行以下命令使上面的配置立即生效 }

```
source /etc/profile
```

{ 100 % : 输入以下命令，如果显示adb的版本信息说明android环境配置成功 }

```
adb version
```

## 2.2 Android项目目录结构

src文件夹： 存放java代码的地方，通过集成开发环境单独指定源代码目录也是可以的。

gen文件夹： 这里面最关键的是R.java文件，这个文件是集成开发环境帮我们自动生成的，不能手动去更改它，否则系统会报错[6]。 里面通过很多内部的静态类来指向res目录中的资源文件，如果开发中发现这个文件，那么很有可能是因为res目录总的文件内部错误或者是文件重复。

assets文件夹： 存放应用中需要用到的文件，例如存放初始化系统用到的数据库文件、音频资源文件等。

res文件夹： 主要用来存放系统中的资源文件，例如布局文件、国际化的字符串文件、图像资源文件等。

AndroidManifest.xml文件： { 51 % : 该文件主要包含了应用程序的基本信息。 } 如果要使用Android系统中的四大组件的话则需要在文件中进行配置，还有就是系统测试和应用也需要用到系统的一些特性，因此，这些也是需要在该文件中进行配置。

default.properties文件： 在应用正式发布时为了防止别人反编译自己的代码，要在该文件中进行配置。 { 41 % : 同时也记录了应用程序开发过程中的在集成开发环境中所需要的版本信息。 }

## 2.3 Activity

### 2.3.1 Activity生命周期方法

onCreate方法： 当 Activity第一次被创建的时候，Android系统将会会执行该生命周期方法，或者是用户在没有退出该应用程序的情况下切换到了其他应用程序，而恰好此时系统资源不够用了，这时系统就会回收该

Activity，但是并未销毁该 Activity。 用户突然又切换到当前应用程序，此时系统就会重新执行该生命周期方法 [7]。 事实上，在实际开发中我们通常在这个方法中初始化界面组件和恢复当前Activity状态。

onStart方法： 每一次 activity从用户不可见到用户可见的时候系统都会执行该生命周期方法，在某些对信息实时性要求高的应用程序中， { 49 % : 需要在该方法中更新该界面的数据， } 例如青蛙卫士中的系统设置界面，为了避免用户使用 Android系统自带的设置界面把服务停掉后回到青蛙卫士的系统界面是看到脏数据， 每次显示该界面的时候都要重新获取当前正在运行的服务。

onRestart方法： 当activity对用户不可见并且没有被系统销毁的情况下，如果该activity再一次对用户可见，那么系统就会执行该生命周期方法。

onResume方法： { 55 % : 当activity获取焦点时，系统会执行该生命周期方法。 }

onPause方法： 当 activity失去焦点时，系统会执行该生命周期方法，失去焦点并不是指该 activity对用户不可见，而是被对话框或者其他透明的 activity遮罩， 用户是可以看见该 activity的，只是不能点击而已，该方法与onResume方法相对应。

onStop方法： { 41 % : 当activity对用户不可见时，系统将会执行该生命周期方法，该方法与onStart方法是相对应的。 }

onDestory: { 51 % : 当activity被系统销毁时，系统会执行该生命周期方法。 } 在实际开发中，通常在该方法内部做一些扫尾动作。 例如解绑通过代码方式在该activity中绑定服务、取消在该activity中注册的内容观察者等。

### 2.3.2 与其他应用程序交互

在我们实际开发的应用程序中，很少有应用程序是单独存在的，或多或少都会使用到第三方应用程序提供的数  
据， 例如实习期间做的一个远程会议软件，需要网服务端上传照片，为了节省开发时间就是直接使用系统自带的  
相机拍照。 { 42 % : 这里就已这个例子来介绍Android系统中各应用程序之间的交互。 } 在我们自己应用程序的  
activity中，把要拍摄照片的全路径名存放在 intent中， 并且在 intent中加入照相的动作和类型，（系统会筛选出手  
机中所已安装的照相机供用户选择）， 将该 intent对象作为参数传递给 startActivityForResult方法。 程序中也调用  
系统自带的相册从相册中选择图片上传，但是相册和相机返回的数据是不同的， 所有还需要传递一个请求码给  
startActivityForResult方法来区分当前操作类型，以便在 onActivityResult方法中根据请求码的不同来出来不同的数据  
[8]。

## 2.4 BroadcastReceiver

### 2.4.1 广播接收者的分类

普通广播： 这种广播并不是同步传输的，并且可以被任意广播接收者收听。 { 46 % : 这种广播的好处是效率非常高。 } 缺点是任意收听该广播的广播接收者都改变不了该广播中所传递的数据，同时也不能阻止别的广播接收者收听该广播。

有序广播： 该广播是依照广播接收者的优先级次序，在收听了该广播的广播接收者中根据该数据依次传递。



{ 44 % : 例如有三个广播接收者X, Y, Z, 优先级是X) Y) Z。 } 那这个广播首先传递给X, 其次再传给Y, 最后传递给Z。 每个广播接收者都可接结束广播的传递, 比如Y结束了广播的传递, Z就不能收到广播消息。 另外, X收到广播消息后可以对消息内容进行更改, 当广播传递到Y那里的时候, 广播接收者Y从广播中取到的数据就是广播接收者X放进去的数据。 { 49 % : 例如, 在Android系统中接收短信广播就是有序广播。 }

{ 50 % : 2.4.2 注册广播接收者的两种方式 }

广播接收者注册方式的不同会影响其存活的时间, 用清单文件配置的方式注册的广播接收者可以在其应用程序没有运行的时候继续收听广播[9], 例如系统自带的电池电量不足的广播接收者, 不需要专程打开某个应用程序来监听它。

还有就是通过代码的方式注册广播接收者, 这种方式注册的广播接收者只有在其应用程序运行的时候才能收听广播, 在实际开发中的应用场景是时时监听广播事件来更新界面上显示的数据, 当界面关闭后该广播接收者就没有存活的意思, { 44 % : 所以就不用配置在清单文件中。 }

## 2.5 Service

### 2.5.1 什么是服务

Android中的服务和windows中的服务类似, 两者都是运行在后台用户看不见的, 其存在的级别会比activity高一些, 同样的也是需要在清单文件中进行配置, 在主进程中运行。 正因为是运行在主进程里的, 所以不能直接在服务中执行需要运行很久的操作, 应该单独开一个子线程来做这些长时间运行的操作, 例如在后台播放音乐, 在导航软件中时时获取最新的位置信息等。

### 2.5.2 本地服务

{ 44 % : 本地服务, 顾名思义就是在某一个应用程序内容运行的服务。 } 在服务外部通过执行startService方法来开启服务和通过执行stopService方法来停止服务。 在服务的内部可以通过执行stopSelf或者stopSelfResult来停止服务。 { 41 % : 值得注意的是, 同一时刻同一个服务只会有一个实例存在。 }

### 2.5.3 远程服务

远程服务, 主要是应用在Android系统中各个应用之间的通信当中, 通过接口并且暴露借口给其他应用程序, 供其他应用程序调用。 其客户端连接到服务点是通过该连接来执行服务里的方法。 在执行bindService()方法连接服务时, 如果是在activity中绑定的服务, 则需要在activity的onDestory()方法中执行unbindService()方法来取消绑定的服务[10]。 要说明的是, 其中多个客户端是可以绑定同一个服务的, 但一个服务只会被开启一次。

{ 51 % : 第3章手机安全卫士分析与设计 }

## 3.1 手机安全卫士需求分析

{ 46 % : 本次设计的应用是一款基于Android平台开发的手机安全与管理软件。 } 其基本功能主要是通过释放系统中不必要的资源来提升手机的运行性能。 而且, 本应用也拥护应用程序管理和进程管理等功能, 减轻用户手机的负担。 它最核心功能是手机防盗, 防止用户的手机丢失、快速找回丢失的手机以及清理丢失手机中的私密数



据，从而保护用户的爱机。

#### { 59 % : 3.1.1 手机防盗功能需求分析与设计 }

**落地有声：** 由于我的眼镜是透明的，经常一觉睡醒找不到眼镜，每次这个时候都想能通过手机控制眼镜发出声音以便更快的找到眼镜。落地有声就是由这个想法演变来的，如果用户的爱机刚被盗不久，那么手机还在用户附近的，如果此时设定的安全号码手机可以用，只需要用安全手机发送一条警报指令给丢失的手机，那么丢失的手机就会播放报警音乐，以便用户在第一时间找回丢失的爱机。

**换卡先知：** Android手机换SIM卡都需要重启手机，每次手机重启的时候判断当前SIM卡的串号是否和被保护的SIM卡的串号是否相同，如果不相同，则手机将在后台自动发送短信给安全手机，这样就在小偷不知道的情况下第一时间得知小偷的手机号码， { 80 % : 大大提升找回手机的可能性。 }

**远程开启防盗：** 当用户爱机没有开启防盗功能的时候，使用任意的手机发送应用和密码和远程开启防盗功能命令时可以开启防盗功能， { 52 % : 并设置当前手机号码安全手机号码，从而为找回手机打下基础。 }

**通信拦截：** 获取丢失的爱机的外拨电话、来电号码、外发短信和收到的短信，并转发给安全手机。这样就可以爱机丢失的第一时间和盗窃者取得联系，也获取了盗窃者朋友的手机号码，进一步增加找回手机的可能性，也大大影响了丢失爱机的正常使用，使盗窃者觉得丢失的爱机拿来也无用，还给手机主人好些。

**保护隐私：** 当用户的爱机丢失时，通过短信发送特定的指令，可将丢失手机通讯录里面的所有联系人、通话记录以及短信内容全部删除，及时保护用户的相关隐私信息。

**充电防盗：** 在某些特殊的场合，用户的爱机急需充电，而用户又不在手机周围，这时开启手机充电防盗就能很好地监控手机的状况，一旦有其他人蓄意拔出正在充电的手机， { 41 % : 手机就会立即发出警报音，并同时发送短信通知安全手机。 }

{ 47 % : 手机防盗功能各个用例说明如下 : }

{ 56 % : 表 3.1 设置安全号码用例说明 }

**用例名称：** 设置安全号码用例

**参与者：** 用户

**目标：** { 53 % : 设置手机丢失后找回手机的安全手机号码 }

**前置条件：** 进入应用程序主界面

**基本事件流：** 1.进入设置手机防盗界面

2.点击安全号码设置条目

{ 56 % : 3.在弹出的安全号码对话框中输入安全号码 }

{ 51 % : 4.点击确定按钮完成安全号码的设置 }

{ 49 % : 表 3.2设置报警指令用例说明 }

用例名称： 设置报警指令用例

参与者： 用户

目标： { 54 % : 设置让手机发出报警音的指令 }

前置条件： 进入应用程序主界面

基本事件流： 1.进入设置手机防盗界面

2.点击设置报警指令条目

{ 47 % : 3.在弹出的报警指令对话框中输入报警指令 }

{ 54 % : 4.点击确定按钮保存报警指令 }

{ 43 % : 表 3.3设置远程开启防盗指令用例说明 }

用例名称： 设置远程开启防盗指令用例

参与者： 用户

目标： { 52 % : 设置远程开启手机防盗功能指令 }

前置条件： 进入应用程序主界面

基本事件流： 1.进入设置手机防盗界面

2.点击设置远程开启防盗指令条目

3.在弹出的远程开启防盗指令对话框中输入远程开启防盗指令

4.点击确定按钮保存远程开启防盗指令

表 3.4设置隐私保护指令用例说明

用例名称： 设置隐私保护指令用例

参与者： 用户

目标： 设置隐私保护指令

前置条件： 进入应用程序主界面

基本事件流： 5.进入设置手机防盗界面

{ 54 % : 6.点击设置隐私保护指令条目 }

7.在弹出的隐私保护指令对话框中输入隐私保护指令

{ 51 % : 8.点击确定按钮保存隐私保护指令 }

{ 44 % : 3.1.2 通讯卫士功能需求分析与设计 }

电话拦截： 将指定的号码加入电话拦截列表中，那么下次在该号码来电的时候不会通知用户，直接在后台挂断电话，防止用户被别人打骚扰电话。

短信拦截： 将指定的号码加入到短信拦截列表中，那么下次在该号码发送短信来的时候删除这条短信，避免用户被垃圾信息骚扰。

全部拦截： 将指定的号码加入到全部截列表中，那么下次该号码发送来的短信和拨打的电话都会被阻止调，避免用户被骚扰。

通讯卫士功能各个用例说明如下：

{ 47 % : 表 3.5添加黑名单用例说明 }

用例名称： 添加黑名单用例

参与者： 用户

目标： { 59 % : 添加黑名单号码到黑名单列表中 }

前置条件： 进入应用程序主界面

基本事件流： 1.进入通讯卫士界面

2.点击添加按钮

{ 59 % : 3.在弹出对话框中输入黑名单号码 }

{ 59 % : 4.在弹出对话框中选择拦截模式 }

{ 45 % : 5.点击确定按钮完成黑名单号码的添加 }

{ 60 % : 3.1.3 软件管理功能需求分析与设计 }

{ 41 % : 按用户程序和系统程序分类管理手机中的应用程序，可以查看、执行、卸载和分享用户程序。 } 而系统程序只能被查看和执行。

{ 46 % : 3.1.4 进程管理功能需求分析与设计 }

目前市面上的某些 Android手机性能并不是很好，多开几个应用程序就会很卡，而正在运行的程序中有些并不是用户所需要的，通过系统服务获得系统中正在运行的应用总数和手机中剩下的内存信息以及运行中进程详细信息展示给用户， { 45 % : 然后提供后台进程停止选项，根据用户的选择终止特定的应用程序。 } 并且提供一个桌面组件给用户，使其能在桌面上点一个按钮就能清理正在运行的应用程序。

{ 53 % : 表 3.6进程管理功能用例说明 }

用例名称： 进程管理功能用例

参与者： 用户

目标： 根据用户的操作结束特定的进程

前置条件： 进入应用程序主界面

基本事件流： 1.进入进程管理界面

2.选择需要停止的应用程序

{ 43 % : 3.单击停止按钮结束所选进程 }

{ 43 % : 3.1.5 程序锁功能需求分析与设计 }

在日常生活中，好友之间可能会互相借手机打电话或者是玩游戏，但是用户不想让好友看自己手机里的某些私密应用程序，如果给手机的锁屏加上密码，那么每次解锁屏幕的时候都得输入密码，自己用着也会麻烦，而且手机解锁屏幕后给玩游戏的话好友可以操作手机上的任意一个应用程序。

程序锁功能就是为了解决以上问题，用户不必给屏幕加密码锁，只需要给特定的应用加上应用锁就可以了，只有解屏后第一次进入加锁的应用程序才会要求输入程序锁密码，在应用解锁后只要没有锁屏的情况下，即使用户退出应用程序界面再次进入时不需要输入密码，此时如果好友再让你借手机给他玩，那么用户只需要把屏幕关闭后给他，除了用户的加锁程序，其他程序好友都可以玩。

程序锁功能各个用例说明如下：

{ 46 % : 表 3.7设置程序锁状态用例说明 }

用例名称： 设置程序锁状态用例

参与者： 用户

目标： { 43 %：按照用户指定的方式设置程序锁状态 }

前置条件： 进入应用程序主界面

基本事件流： 1. 进入设置中心界面

2. 点击程序锁条目，选择程序锁条目设置程序锁的状态

{ 45 %：表 3.8添加需要保护的应用用例说明 }

用例名称： { 53 %：添加需要保护的应用用例 }

参与者： 用户

目标： { 53 %：添加应用程序到已加锁应用列表中 }

前置条件： 进入应用程序主界面

基本事件流： 1. 进入程序锁界面

2. 点击未加锁列表中的应用条目

表 3.9移除已加锁应用用例说明

用例名称： { 44 %：移除已经加锁的应用用例 }

参与者： 用户

目标： { 40 %：从已加锁应用列表中移除应用程序 }

前置条件： 进入应用程序主界面

基本事件流： 1. 进入程序锁界面

2. 点击已加锁标签

{ 43 %：3. 点击已加锁列表中的应用条目 }

{ 58 %：3.1.6 手机杀毒功能需求分析与设计 }

手机安全厂商和各大应用市场收集了市面上常见的病毒软件，本功能需要收集他们所提供的病毒软件的包名，将其存储到本地数据库中，在执行病毒扫描的时候将手机中所有已安装的用户程序包名与病毒数据库中的包名进行对比， { 59 % : 如果相同则询问用户是否卸载该应用程序。 }

{ 56 % : 3.1.7 系统优化功能需求分析与设计 }

清理应用程序缓存在手机中的缓存数据，但是这需要具备root权限，所以本应用的实现方式打开系统自带的查看应用程序详细信息界面，在该界面中清理缓存数据。

{ 61 % : 3.1.8 自动升级功能需求分析与设计 }

青蛙卫士自动升级功能各个用例说明如下：

表 3.10 青蛙卫士升级方式用例说明

用例名称： 应用升级方式用例

参与者： 用户

目标： { 42 % : 按照用户指定的方式设置应用升级方式 }

前置条件： 进入应用程序主界面

基本事件流： 1. 进入设置中心界面

{ 41 % : 2. 点击自动更新条目，选择升级方式 }

表 3.11 青蛙卫士自动升级用例说明

用例名称： 青蛙卫士自动升级用例

参与者： 用户

目标： 在wifi打开的情况下，按照用户指定的升级方式升级应用程序

前置条件： 进入安全卫士欢迎界面

基本事件流： { 44 % : 1. 用户允许手机安全卫士应用程序 }

2. 程序自动判断应用程序当前版本和服务器最新版本，并显示

{ 66 % : 3. 在弹出的对话框中选择立即更新 }



{ 55 % : 第4章手机安全卫士详细设计与实现 }

#### 4.1 欢迎界面实现

该界面的主要功能是显示软件升级信息和初始化应用程序，如果服务器的最新版本和当前版本相同则显示应用的当前版本信息， 否则显示提示升级信息对话框，用户根据需要自行决定是否升级。 运行界面如图：

##### 4.1.1 检查升级功能实现

判断当前用户是否设置了自动更新应用程序，手机有没有链接 wifi，如果有设置自动更新并且已链接 wifi， { 52 % : 则从服务器获取最新版本信息和本地版本对比，如果本地版本号低于服务器版本号， } { 48 % : 则弹出升级对话框供用户选择是否立即升级。 }

{ 51 % : 4.1.2 升级对话框显示功能实现 }

升级对话框采用自定义的对话框，创建一个自定义对话框类继承于 Dialog类，其关键是编写自定义布局文件和重写构造方法， 实现该对话框能在应用的其他功能模块中能重用，关键代码如下：

首先通过抽取样式文件，达到样式代码的重用，然后通过重写构造方法，根据不同的应用场景传递不同的参数，达到自定义对话框的重用。

#### 4.2 应用程序主界面实现

应用程序主页是展示程序的各个功能模块，从而使用户能够快速进入到自己想使用的功能。 该界面的另一个很重要的功能就是修改程序锁密码，在该界面连续三次点击菜单按钮弹出修改密码对话框。 运行界面如图：

#### 4.3 程序锁实现

通过帧布局实现Android界面的复杂布局，在页面的底部放置两个按钮控制FrameLayout里的两个ListView的显示与否，模拟出TabHost的效果。 由于已加锁应用和未加锁应用列表极其相似，所以这里只需在编写一个 ListView适配器， { 43 % : 在适配器中通过一个 flag标记区分当前适配器是未加锁适配器还是已加锁适配器。 } 为了提升程序锁的运行效率，不每次都从数据库中获取最新已加锁应用程序列表，而是用两个集合来分表保存已加锁应用程序和临时不需要保护的应用程序， 在每一次收到锁屏事件的时候清空临时不需要保护的集合，在解锁某一个应用程序的时候将该应用程序加入到临时停止保护的集合中。 需要注意的是，这里需要通过代码的方式注册监听锁屏事件的广播和解屏事件的广播。 { 44 % : 程序锁界面的运行截图如图： }

#### 4.4 手机防盗实现

手机防盗界面主要是编写 ListView及其适配器，根据用户点击不同的条目弹出不同类型的自定义对话框， 有些对话框功能不同，但是内容的格式完全相同，这就需要重写自定义度画框的构造方法， { 43 % : 根据传入参数的不同显示不同内容和类型的自定义对话框。 } 监听手机启动完毕广播事件的广播必须在清单文件中注册。 { 47 % : 手机防盗界面的运行截图如图： }

#### 4.5 通讯卫士实现

通讯卫士界面主要也是便表列表和适配器，在注册监听电话到来和短信到来事件的广播接收者时，需要将其优先级设置到最大值，以确保本应用最先收到这两个广播事件，在第一时间挂断电话以及删除最近一条通话记录和短信记录，避免用户看到脏数据。 { 40 % : 通讯卫士界面的运行截图如图 : }

#### 4.6 系统设置实现

系统设置界面的难点是通过自定义组合控件的方式实现自定义view，从而使每一个设置条目只需要引用一个组件，实现代码的复用。为了防止用户打开android系统自带的设置界面关掉安全卫士中的某些服务出现脏数据，每次显示该界面的时候必须在onStart方法内重新获取最新数据。 { 57 % : 系统设置界面的运行截图如图 : }

### 第5章 结论与展望

#### 5.1 结论

{ 40 % : 半年前，对毕业后有着强烈意愿从事 Android开发工作的我，选择了基于 Android平台的手机安全卫士的设计与实现这个课题作为我的毕业课题， } 在长达三个月的 Android实习工作中学习到了从事 Android开发工作必备的一些技能，例如屏幕适配、兼容多版本系统等，了解了移动应用程序开发的基本流程，同时在林老师的细心指导下完成了我的毕业设计作品《青蛙卫士》。

由于 Android是基于 Linux系统的，所以我尝试着去学习 Linux操作系统，在这半年里我学会了 Ubuntu系统中开发人员必备一些技能，例如掌握 Ubuntu系统下开发 Android应用程序的能力，学会使用 Ubuntu系统中的常用办公软件等。 { 43 % : 在编写青蛙卫士的过程中巩固了 Android的基础知识， } 通过看别人的技术博客掌握了 Android中常见动画效果、编写自定义组件、编写自定义组合控件以及加深了对 Framework层 API的印象，从以前看教学视频学习编程过度到看技术博客学习编程，为以后的职业生涯打下了坚实的基础。

#### 5.2 进一步工作的方向

目前位置，青蛙卫士的功能基本上都实现了，但是在一些功能中仍存在不完善之处，例如手机防盗模块没有实现每次手机重启完毕后调用系统前置摄像头拍照以便手机丢失后获取小偷的长相、没有加入第三方平台用户登录、没有编写后台管理系统上传和恢复用户数据等。作为一个手机安全卫士，功能还需在日后的工作中进一步地考查调研并完善。