

摘 要

近年来，Android 系统已成为最流行的手机操作系统之一。在 Android 系统中，很多时候会需要备份用户安装的应用和短信等信息。例如，在 Android 系统的“刷机”过程中，用户安装的软件以及联系人、通话记录、短消息等用户数据都会丢失。因此，需要开发一款应用及用户数据备份软件，并且在安全性上应比较完善，以提高数据备份软件的可用性及实用性。

本文在学习和研究了 Android 系统中的应用存储机制和短信存储机制后，使用 Eclipse 作为开发工具，编写了一款完成了应用备份还原、短信备份还原等基础功能的备份软件。其中应用备份功能可以将应用安装包及数据文件打包成 zip 格式存放到 SD 卡上；短信备份功能将需要备份的短信保存为 xml 格式，并对其中的内容进行加密。短信备份文件同时可以备份到百度云网盘中。

本文介绍的基于 Android 平台的应用及用户数据备份软件虽然已实现了基本的功能，但是在流程控制、错误处理、异常检测等方面还有一些不足，这些地方都还有许多能够进一步完善的内容。相信通过对已实现功能的不断完善，能够完成更有实际应用价值的软件产品。

关键词：Android；备份；还原

ABSTRACT

In recent years, Android has become one of the most popular mobile operating systems. Also it is crucial in Android system that users owning the demands in backup applications they installed and text message and so on. For example, after the operation called “Flash” in Android, the user data such as installed applications, contacts data, call history, and text messages will lost. So it’s necessary to develop an application and user data backup application, and this application should focus on security to improve availability and applicability.

In this paper, on the basis of study and research of the application storage mechanism and text message storage mechanism in Android, develop an application using Eclipse, which implement basic functions such as application backup and restore function, and text message backup and restore function. Application backup function can archive application install files and data files to SD card as zip format. Text message backup function can store text message data into xml file, and these data could be encrypted. The backup files could be upload to Baidu Person Cloud Storage Service.

Though the application and user data backup and restore software based on Android introduced in the paper has already implemented the basic functions, it still has many errors and inadequate methods in .process control, error handling and exception detection. There are many things to do to solve these inadequacies. I believe that this application may have more practical value by constantly improving existing functions.

Key Words: Android; Backup; Restore

目 录

1 绪 论	1
1.1 选题背景	1
1.2 研究目的与意义	1
1.3 国内外研究现状	2
1.3.1 国外同类软件	2
1.3.2 国内同类软件	2
1.4 要解决的关键问题	3
1.4.1 系统数据库的连接查询	3
1.4.2 多线程中的 UI 操作	3
1.4.3 数据文件访问权限问题	4
1.5 本文主要内容	4
2 理论基础及关键技术简介	5
2.1 开发工具及设备介绍	5
2.2 技术介绍	5
2.2.1 Android 4.0 系统简介	5
2.2.2 BusyBox 简介	6
2.2.3 SQLite 简介	6
2.2.4 XML 简介	6
2.2.5 BASE64 加密算法简介	7
2.2.6 TAR 打包工具简介	7
2.2.7 百度个人云存储 PCS 简介	7
3 系统结构与模型	8
3.1 系统功能简述	8
3.2 系统模块	8
3.2.1 系统状态检测模块	8
3.2.2 应用备份及还原模块	10
3.2.3 短信备份及还原模块	11

3.2.4 百度云 PCS 模块.....	11
4 详细设计.....	13
4.1 用户界面设计.....	13
4.1.1 首页.....	13
4.1.2 应用页面.....	14
4.1.3 短信页面.....	15
4.2 模块详细设计.....	16
4.2.1 系统状态检测模块详细设计.....	17
4.2.2 应用备份还原模块详细设计.....	17
4.2.3 短信备份还原模块详细设计.....	19
4.2.4 百度云 PCS 模块详细设计.....	21
5 代码实现与测试.....	22
5.1 代码实现.....	22
5.1.1 首页模块代码实现.....	22
5.1.2 应用备份还原模块代码实现.....	23
5.1.3 短信备份还原模块代码实现.....	26
5.1.4 工具类内容介绍.....	27
5.2 测试.....	30
5.2.1 功能测试.....	30
5.2.2 性能测试.....	35
6 结束语.....	36
致 谢.....	37
参考文献.....	38

1 绪 论

1.1 选题背景

近年来，智能手机已成为移动终端市场的主流，以 Android、iOS、Windows Phone 等操作系统为主导的智能手机已广泛普及。Android 是由 Google 开发的基于 Linux 的开源手机平台，也是目前最流行的手机操作系统之一。据 IDC2013 年公布的最新数据，2013 年第一季度，Android 操作系统的市场占有率达到 75%，出货量达到 1 亿 6000 万部，同比增长 79.5%。对于开发者来说，移动终端领域的应用开发业已成为一个相当具有发展前景的方向。

Android 系统平台由操作系统、中间件和应用软件三部分组成，它是目前为止首个能够完全定制、免费、开放的手机平台。Android 是完全免费的，使用 Android 并不需要授权费，而且因为 Android 平台有丰富的应用程序，也大幅降低了应用程序的开发费用。Android 底层使用开源的 Linux 系统，同时开放了应用程序开发工具，使所有开发人员都使用统一的、开放的开发工具进行开发，保证了 Android 应用程序的可移植性。Android 应用使用 Java 语言进行开发，支持 SQLite 数据库、2D/3D 图形加速、多媒体播放和摄像头等硬件设备，并内置了丰富的应用程序。

不同于 iOS 的封闭，由于 Android 是一个开放源代码的系统，任何人都可以修改其源代码，并将修改过的系统安装在设备上。像这样为设备安装操作系统的过程，通常被称作“刷机”。在刷机过程中，已安装的应用和应用数据、以及短信等用户数据都会被清除。针对这样的问题，开发一款能够备份用户数据的应用的需求就应运而生。

1.2 研究目的与意义

能根据需要安装具有各种功能的应用，是移动智能操作系统最强大的功能之一。在 Android 系统中，Google 公司官方的应用市场 Google Play Store 中的已上架应用已超过 80 万，并有望在 2013 年 6 月突破 100 万。而除了官方应用市场外，还有多种渠道能够获取到应用安装文件，可将其直接安装到手机上。庞大的应用基数和方便的应用安装方法，使得人人都能找到很多心仪的应用来使用。但是由于上文所提到的刷机的问题。在刷机后，应用及数据都会丢失掉，此时往往便需要再找到应用的安装包来安装。然而，除了在 Google Play Store 商家的应用外，在第三方市场下载的应用是很难找回的，且应用的数据文件都会丢失。当用户安装的应用较多时，进行刷机操作后，要逐个找回曾安装过的应

用是一件很困难且繁琐的事情。因此若可以将已安装的应用及数据用某种方式做备份，当需要的时候，可以将其还原到手机中，就能很好地解决这一问题。

1992 年，世界上第一条短信在英国沃达丰的网络上通过电脑向手机发送成功，从而宣布手机短信诞生。至于中国的第一条短信诞生于何时何地已无从知晓，但据考证，中国的移动通信网络早在 1994 年就具备了短信功能，只是那时有手机的人根本不需要它罢了。随着手机的日益普及，从 1998 年开始，移动、联通先后大范围拓展短信业务：2000 年，中国手机短信息量突破 10 亿条；2001 年，达到 189 亿条；2004 年，数字飞涨到 900 亿条。短信理所应当地成为了第五种信息传播工具；据工信部最新公布统计数据显示，2012 年，全国移动短信发送量达到 8973.1 亿条。尽管随着智能手机的普及，近年来以微信为代表的即时通信软件在一些方面渐渐替代了短信，但短信仍是人们生活中所不可缺少的通讯方式。而为了保存重要的或是具有纪念性的短信，同时为了实现将短信移动到另一部 Android 手机中的功能，需要将短信备份为与设备无关的备份文件，再在需要进行还原操作。因此，在本应用中的短信备份和还原应具有设备无关性，即备份文件中的数据可通过应用被写入到任何一部满足运行条件的 Android 设备中去。

1.3 国内外研究现状

1.3.1 国外同类软件

目前，在国内外已有多款同类型的应用产生。其中国外较著名的应用有钛备份 (Titanium Backup) 和短信备份还原 (SMS Backup & Restore Pro)。

钛备份：钛备份是 Android 平台最为强大的备份工具，支持备份应用和应用数据、系统设置以及用户数据到 SD 卡，并可同步到 Dropbox、Box、Google Drive 网盘，同时支持批量备份、定时备份等计划任务。但该应用过于复杂，共有总计超过 100 项的选项，对于普通用户来说过于繁琐，不具备很好的交互性。完全版售价 6.58 美元，在国内无法通过常规手段进行购买。

短信备份还原：该应用支持将短信息备份为 XML 格式，保存在 SD 卡上，并支持上传到 Dropbox 和 Google Drive，但需要插件的支持。完全版售价 2.49 美元，在国内无法通过常规手段进行购买。

1.3.2 国内同类软件

国内同类产品中，具有代表性的主要有 GO 备份，QQ 同步助手，以及各厂商为自己的机型定制的专用软件，如 MIUI 备份、华为智汇云等等。

GO 备份：功能较为全面，但不能按联系人单独备份短信。部分功能需要付费使用。

QQ 同步助手：能够备份应用、短信及联系人信息，但是只能备份到云端，不能备份到本地。

MIUI 备份：功能较为全面，但不能按联系人单独备份短信。

通过对目前已有的同类应用进行分析，可以发现，在国内环境下，购买国外的应用需要较为繁琐的流程，对大多数用户来说是很难办到的。而国内应用在各方面都有些不足，举例来说，基本所有软件都只支持将短信全部备份，而无法有选择地进行备份和还原。类似的，目前还没有软件能够只备份应用的数据。对于某些本身较大的应用，如游戏等等，备份时应只备份数据，因为备份应用本身会占用较长的时间。

1.4 要解决的关键问题

1.4.1 系统数据库的连接查询

由于 Android 系统的用户权限机制较为严格，因此一个应用无法直接访问其他应用或是系统文件夹中的内容，包括短信数据库和联系人数据库。若要访问这些数据，只能通过两种方法来实现。第一种方法是通过 Shell 命令获取 root 权限后，执行 Shell 命令来对其进行操作，另一种方法是使用 Android 的跨应用共享机制，即 `ContentProvider` 和 `ContentResolver`。应用可以通过实现 `ContentProvider` 的接口，将内部的数据暴露出去，使外界可以对其进行访问，但其内部的具体实现和存储方式仍是不透明的。而 `ContentResolver` 则是访问其他应用中数据的接口，外部的应用可以通过 `ContentResolver` 接口访问 `ContentProvider` 提供的数据。用这样的方法就可以对其他应用中和系统中的数据库进行访问。第一种方法的优点在于实现起来较为简单，缺点在于需要获取 root 权限，且 Shell 层与 apk 应用层的数据交互较为繁琐。第二种方法要优于第一种方法，也是一般所推荐的方法，这种方法的优点在于无需 root 权限，只需要读取短信数据库、联系人数据库等用户权限即可，且获取到的数据可以直接使用，但缺点在于 `ContentResolver` 提供的接口功能并不完全，例如不支持多表连接查询、不支持 `GROUP BY` 语句等等，因此，若要使用第二种方法，需要解决多表连接查询的问题。

1.4.2 多线程中的 UI 操作

在应用开发过程中，不可避免地要使用多线程来完成某些功能。然而在 Android 系统中，对于 UI 的操作不是线程安全的，只有主线程能够修改 UI。因此如何在子线程中根据需要修改界面控件等 UI 元素，是一个需要考虑的问题。

1.4.3 数据文件访问权限问题

Android 系统中使用类 Unix 系统的文件权限管理机制。然而，在 Android 系统中，对于内部空间的权限管理虽然十分完善，但是对于外置存储设备比如 SD 卡，存放其中的文件，权限只能是 `r-xrwxr-x`，因此，若应用数据文件被备份到 SD 卡后，文件原本的访问权限就会丢失，这样即使再将其还原到数据文件夹，也不会被正确识别。为了解决这个问题，当备份应用数据文件时，需要考虑使用一种方法将访问权限信息也能完整保留下来。

1.5 本文主要内容

本文主要通过关键技术介绍、总体设计、详细设计、代码实现及测试等几个部分来逐步详细介绍 Android 平台下应用及用户数据备份软件的实现过程。

在关键技术介绍部分，介绍了开发平台和工具，以及中所用到的主要技术。在总体设计阶段明确了系统功能，并划分出相应的功能模块，给出模块层次及功能实现流程。在详细设计阶段，完成了界面草图设计，并细化了功能实现的具体步骤。在代码实现及测试部分从代码角度给出了核心功能的具体实现，并根据功能测试验证了程序的功能实现，根据性能测试验证了程序的执行效率优化。最后，通过结束语，总结了开发过程中遇到的种种问题及心得体会，同时也提出了在开发过程中所学习到的经验。

2 理论基础及关键技术简介

2.1 开发工具及设备介绍

本应用使用 Eclipse 4.2.1 作为开发工具，Eclipse 是一个开发源代码的、基于 Java 的可扩展开发平台，它支持多种语言和第三方插件，可适应各种开发环境的要求，也是目前开发 Android 应用的首选开发环境。Eclipse 的运行需要 JRE(Java Runtime Environment, Java 运行时环境)及 JDK(Java Development Kit)的支持，在开发本应用过程中使用的是 JDK1.7。

在本应用的开发过程中，使用的版本控制工具为 Subversion，将项目托管在 Google Code 平台上。Subversion 是一个开源的版本控制系统，诞生于 2000 年，并在之后的十年中获得了极大的成功。Subversion 最初产生的目标是取代 CVS，它在很多方面都和 CVS 很相似，但修正了 CVS 所没能解决的许多问题。

开发平台：Windows 8 64 位专业版。

测试设备：小米 1，系统版本：MIUI v5 基于 Android 4.1.2。

2.2 技术介绍

2.2.1 Android 4.0 系统简介

Android 是一个在 Linux 内核基础上开发出来的开源的操作系统，目前主要被用于手机和平板电脑等移动设备中。

开发 Android 应用需要使用 Google 提供的 Android SDK(Software Development Kit)，它是在各个平台上进行应用开发的一套软件包和开发框架。Android SDK 按照系统版本进行分级，目前最新的版本为 API 17，其对应的版本为 Android 4.2.2。新版本可以向下兼容旧版本下的应用。目前能够升级到 Android 4.2.2 版本的手机仅有极少数，绝大多数用户使用的是 Android 2.3 及 Android 4.0 版本，且 Android 4.0 以上版本的手机正在逐渐成为主流。

截至 2013 年 4 月 3 日，Android 系统不同版本的用户比例如表 2-1。从表 2-1 中可以看出，Android 4.0 以上版本的用户数量已超过用户总数的一半。由于在 Android 4.0 版本中，Google 为其新增了许多优秀的特性，最直观的表现就是统一的系统 UI 规范，被称为 Android Design 的抽象拟真设计规范。同时 Google 还提供了 Holo Theme 控件主题，使开发者能够使用与系统风格一致的空间开发应用，保持系统 UI 的整体一致性。

表 2-1 Android 系统用户分布

版本名称	API Level	用户分布
4.2.x Jelly Bean 果冻豆	17	2.00%
4.1.x Jelly Bean 果冻豆	16	23.00%
4.0.x Ice Cream Sandwich 冰激凌三明治	14-15	29.30%
3.x.x Honey Comb 蜂巢	11-13	0.20%
2.3.x Ginger Bread 姜饼	9-10	39.80%
2.2 Froyo 冻酸奶	8	4.00%
2.1 Eclair 闪电泡芙	7	1.70%
1.6 Donut 甜甜圈	4	0.10%

Android 系统有着和 Linux 系统类似的用户权限机制。在 Android 中最高的权限即 root 权限，只有获取了 root 权限，才能对系统文件进行操作。

在 Android 系统中，apk 应用层通常是无法获取到 root 权限的，因此为了实现某些需要 root 权限的功能，须使用低层 Shell 命令。但 Android 的 Shell 命令并不完整，因此需要找到一个替代的解决方案。

2.2.2 BusyBox 简介

BusyBox 是 UNIX 常用 Shell 指令的一个集合包，虽然它比起完整的 Shell 命令，缺少了一些参数，但仍不失为一个适合用在小型嵌入式系统中的完整的环境。BusyBox 可以很方便地安装在具有 root 权限的 Android 系统中，作为原生 Shell 的替代品。

2.2.3 SQLite 简介

SQLite 是一个具有完备性(self-contained)、无服务器(Serverless)、零配置(Zero-configuration)的事务性 SQL 数据库引擎的软件库，也是 Android 系统中所使用的默认数据库。它是世界上部署最广泛的 SQL 数据库引擎。SQLite 的系统资源占用非常低，因此广泛被应用于嵌入式设备中。相比起 MySQL 这样的较大型的数据库，SQLite 虽然在功能上有所不及，但响应速度及处理速度都更高。SQLite 没有分离的服务器进程，而是直接对数据库文件进行读写，因此数据库的所有内容都存放在一个单独的数据库文件中，并且可以跨平台访问。

2.2.4 XML 简介

XML(Extensible Markup Language)，即可拓展标记语言，它是一种使用类似 HTML(Hypertext Markup Language，超文本标记语言)标签的形式，对文档进行结构化标

记的标记语言。XML 与 HTML 的不同之处在于，HTML 注重对信息的显示，而 XML 注重对信息的存储和传输。在 Android 系统中，对于 XML 文件有着良好的支持，因此可以考虑使用 XML 文件来存放数据。

2.2.5 BASE64 加密算法简介

由于 XML 并没有自带加密，而 SD 卡中的文件没有读写权限的保护，因此需要使用一种加密算法，来对文件中的数据进行加密。

BASE64 算法严格来说并不算是一种加密算法，它是一种基于 64 个可打印字符来表示二进制数据的编码方式，通常用于对文本数据的存储和传输。BASE64 可以作为一种简单的加密算法来对数据进行加密，Mozilla Thunderbird 和 Evolution 等软件就使用 BASE64 编码来存放电子邮件密码等信息。

因此在本应用中，对于用户短信的备份文件，就使用了 BASE64 加密技术，保证用户隐私不被盗取。

2.2.6 TAR 打包工具简介

Android 的内核基于 Linux，因此具有和类 Unix 系统相同的文件访问权限模型，而在外置 SD 卡中，则不具备完善的权限模型，因此当把应用数据文件备份到 SD 卡时，要能保存其访问权限信息，为此需要使用压缩打包工具 TAR 对其进行压缩。

TAR 是类 Unix 系统中常用的打包工具，可以将多个不同文件打包成一个，类似于常用的 RAR 格式和 ZIP 格式，但不同之处在于 TAR 文件能够保存文件被打包前的访问权限，并且在解包后能使文件的权限保持不变。因此，可以将应用数据打包为 TAR 包，这样在 SD 卡中存放时，即可保证在恢复时访问权限不变。

2.2.7 百度个人云存储 PCS 简介

百度个人云存储 PCS 是由百度公司推出的针对个人用户的云存储服务。开发者通过使用百度提供的 API，可以实现满足各种需求的个人应用，并将数据通过百度云网盘保存起来，实现“云”和“端”的无缝连接。

百度 PCS 使用 OAuth 2.0 协议进行授权管理，第三方无需获得用户的账号密码，而只需通过登录一次后生成的令牌，即可进行相应的操作。

3 系统结构与模型

3.1 系统功能简述

本应用的主要功能可分为 2 个部分：应用备份及还原，短信备份及还原。

为完成这两个功能，需要另外完成其他的功能。如检测系统相关状态，以判断应用能否在设备上正常运行；对于短信的备份文件，应对其内容进行加密，以保证用户的数据安全。登录百度云账号后，将数据备份到百度云网盘。

3.2 系统模块

根据要实现的功能，将主要模块分为四个部分：系统状态检测、应用备份及还原、短信备份及还原、百度云 API 调用。模块层次如图 3-1 所示。

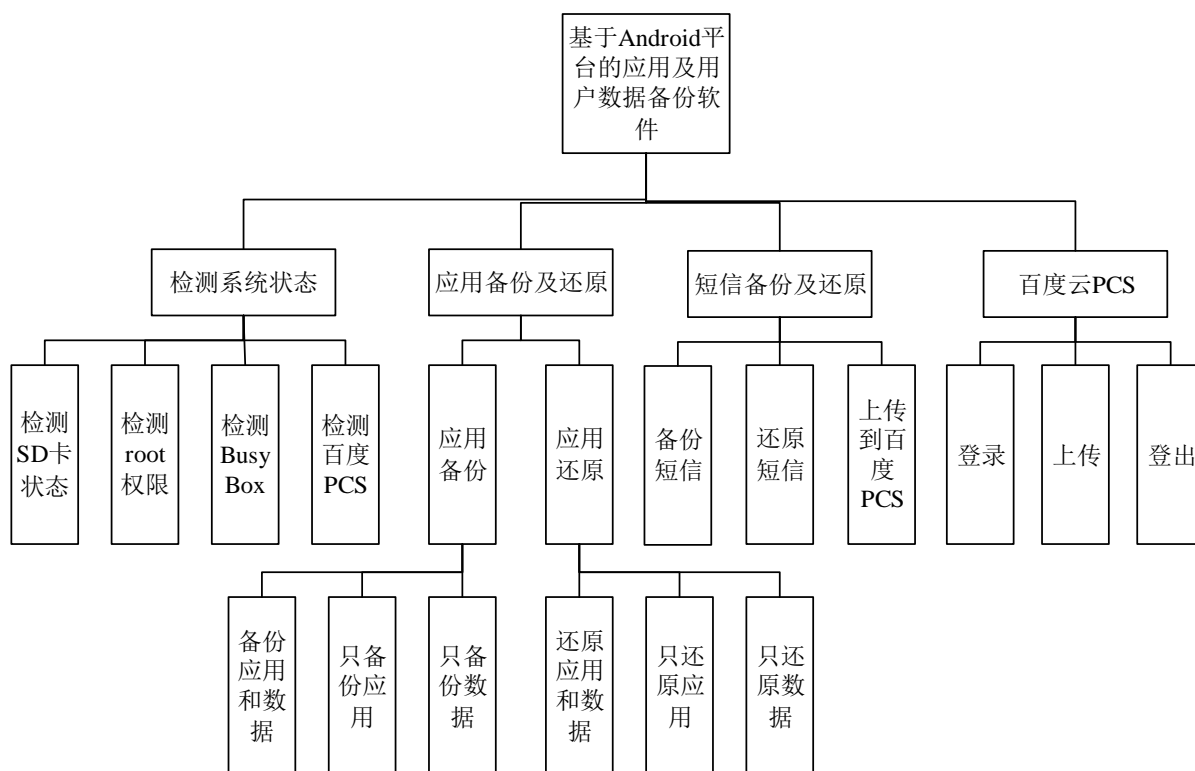


图 3-1 系统层次图

3.2.1 系统状态检测模块

对于系统状态的检测在应用被打开时运行，主要负责检测和显示系统的相关状态，并根据系统状态，禁用某些无法实现的功能。需要检测的项目有：

- 1) SD 卡挂载状态。由于所有的备份文件都需存放在 SD 卡上，因此若没有检测到 SD 卡，则所有的备份和还原操作都无法实现。此时，需要禁用所有功能。
- 2) 是否获取到 root 权限。备份及还原应用的数据文件夹需要读写/data/data 目录，而该目录需要 root 权限才能进行读写操作。因此，若系统不能获取到 root 权限，则应用数据的备份及还原无法实现。BusyBox 首次安装在系统中时，也需要 root 权限才能正常安装。
- 3) BusyBox 是否已安装。使用 BusyBox 是为了备份、还原数据文件夹，并对其进行压缩和解压。其中备份、还原的操作需要在获取 root 权限后执行。
- 4) 百度 PCS 登陆状态。当用户未登录时，将不能把短信备份文件上传到云端。当用户登录时，获取到令牌后，将其存放在 SQLite 数据库中。此时备份短信后，可通过已保存的令牌进行上传操作。

系统状态检测的程序流程图如图 3-2 所示。

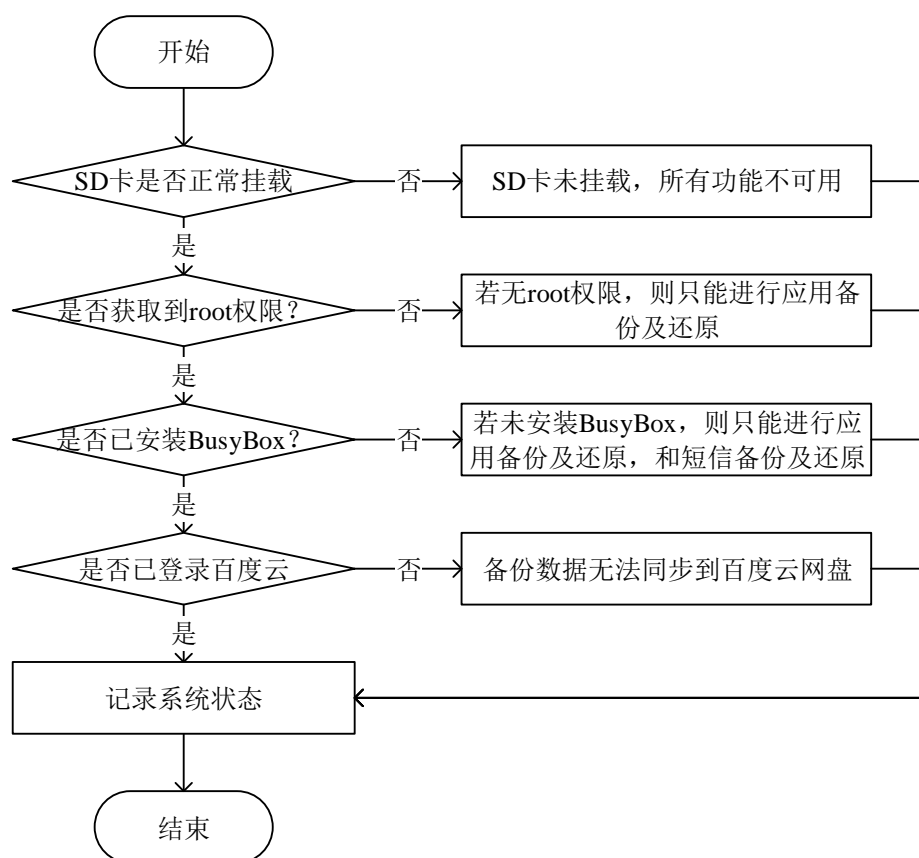


图 3-2 系统状态检测程序流程图

3.2.2 应用备份及还原模块

备份应用时，可以选择只备份应用、只备份数据及全部备份。同理，在还原应用时，可以选择只还原应用，只还原数据及全部还原。这样做的目的是为了最大限度地满足用户的需求。应用备份和还原的流程图见图 3-3 和图 3-4。

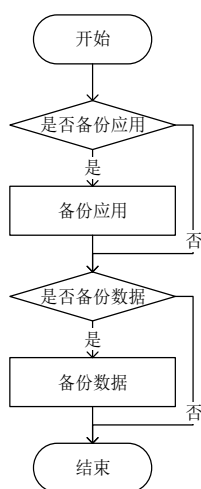


图 3-3 应用备份流程图

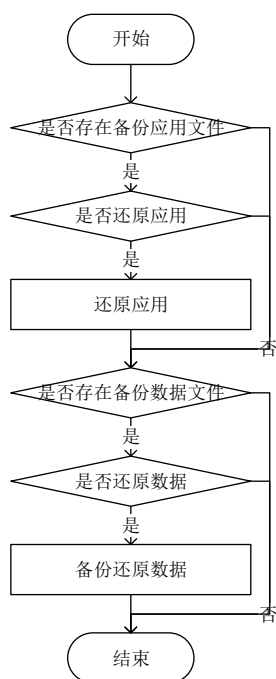


图 3-4 应用还原流程图

3.2.3 短信备份及还原模块

主要负责从数据库中读取短信的相关信息，并完成短信的备份还原操作。若已登陆百度账号，还要将备份文件上传到云端。

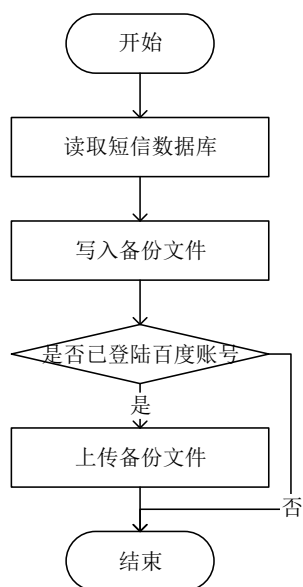


图 3-5 短信备份流程图

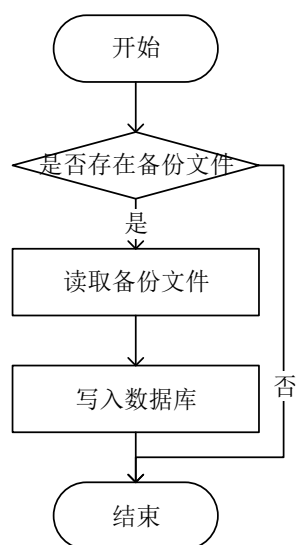


图 3-6 短信还原流程图

3.2.4 百度云 PCS 模块

该模块主要负责登录和登出百度账号、保存应用令牌，并为其他模块上传文件提供

接口。

当用户要使用百度云 PCS 时,首先需登录百度账号,此时会弹出百度账号登陆界面,用户可输入账号及密码进行登录。由于使用 OAuth 2.0 协议,因此该登录过程是直接与百度公司的服务器进行通信的,第三方无法获得用户的账号及密码,而只能获得登陆成功后所返回的一个令牌,通过令牌进行相应的操作。因此,需将该令牌保存到数据库中,以备以后使用。

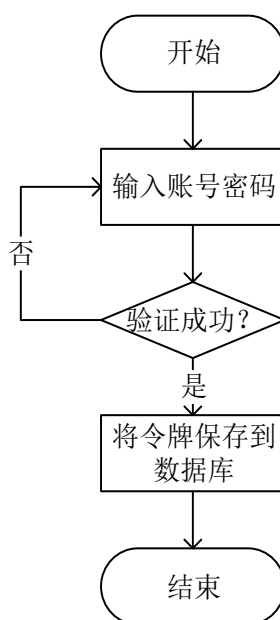


图 3-7 百度云登录流程图

由于该模块的所有工作都可在后台进行,因此可将其与其他模块进行整合。

4 详细设计

4.1 用户界面设计

人机界面设计是接口设计的一个重要组成部分。对于交互式系统来说，人机界面设计和数据设计、体系结构设计及过程设计一样重要。近年来，人机界面在系统中所占的比例越来越大，在个别系统中人机界面的设计工作量甚至占总设计量的一半以上。由于人机界面是用户对软件最直观的印象，因此人机界面的设计质量，直接影响用户对软件产品的评价，从而影响软件产品的竞争力和寿命，因此，必须对人机界面设计给予足够重视。

在 Android 3.0 后的版本中，引入了 Fragment 的概念。Fragment 正如其名，代表了 Activity 中用户接口的一个部分，所以可以在一个 Activity 中，使用多个 Fragment 来构建多面板的应用，并且这些 Fragment 能够被复用。Fragment 具有独立的生命周期和消息响应，并且在 Activity 运行期间可以任意地添加和移除。因此可以认为 Fragment 是 Activity 的一个子集。同时，由于不同的 Fragment 在同一个 Activity，互相之间的消息和数据传递就可在内部进行，而无需使用较复杂的 AIDL(Android Interface Definition Language，Android 接口定义语言)或是 Broadcast 来实现进程间的通信。

Fragment 与 Activity 有着很多相似之处，但它们的生命周期不同。Fragment 与 Activity 的生命周期对比见图 4-1。

在已分析出的四个功能模块中，百度 PCS 模块除登录与登出功能外，主要在后台运行，没有具体的前端交互，因此无需为其设置单独的界面。综合考虑后，可将系统界面分为 3 个部分，每个部分使用一个 Fragment 来实现，多个 Fragment 间使用 ViewPager 来进行切换。系统的三个页面，可分为首页、应用页面及短信页面。

4.1.1 首页

首页用来显示系统状态及提供百度账号的登录与登出。其前端界面主要为文字信息，及登录/登出百度账号的按钮。当未登录时，点击按钮，弹出百度登录界面。当登录后，点击按钮，退出登录，清除登录信息。首页的人机界面设计见图 4-1。

首页	应用
SD卡状态:	已挂载/未挂载
	详细信息
root权限:	已获取/未获取
	详细信息
BusyBox状态:	已安装/未安装
	详细信息
百度云:	已登录/未登录
	详细信息
<div>登录/登出</div>	

图 4-1 首页页面人机界面设计

4.1.2 应用页面

应用界面考虑使用列表形式来显示已安装应用信息。每一个列表项显示一个应用的信息。需显示的信息有：

- 1) 应用图标，使用应用的图标可提高其辨识度，方便查找。
- 2) 应用名称
- 3) 应用版本号
- 4) 备份文件信息，需显示备份文件的备份时间和备份类型（仅应用、仅数据或应用及数据）

当点击某一项后，弹出对话框，在对话框中进行功能的选择。应提供的功能有：

- 1) 备份该应用的安装文件和数据
- 2) 仅备份安装文件
- 3) 仅备份数据
- 4) 还原安装文件和数据
- 5) 仅还原安装文件
- 6) 仅还原数据

- 7) 删除该应用的备份文件
- 8) 卸载该应用

应用页面的人机界面设计见图 4-2。

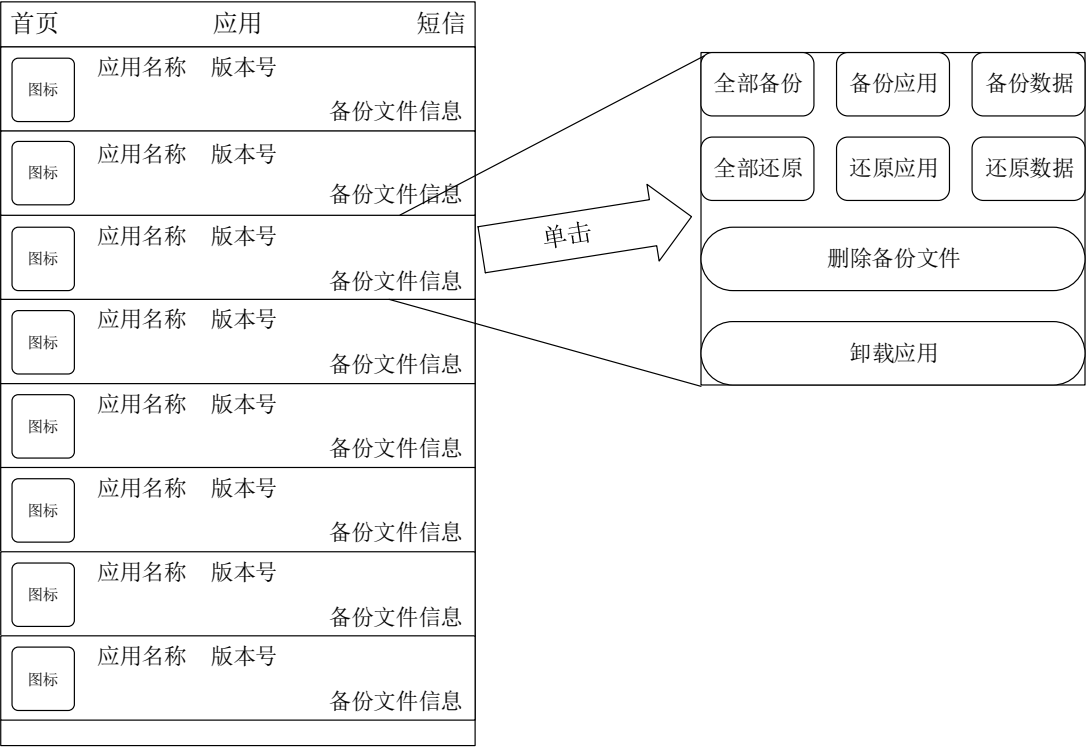


图 4-2 应用页面人机界面设计

4.1.3 短信页面

短信页面与应用页面类似，也使用列表来显示短信信息。列表中的每一个项表示一个对话。在 Android 中，一个对话指的是与某一个联系人的所有短信记录，包括发送给该联系人和该联系人发来的短信。需要显示的列表项有：

- 1) 电话号码
- 2) 联系人姓名，若该号码是用户的联系人，则应该显示联系人姓名，以方便识别。
- 3) 对话条数
- 4) 短信预览，该对话最新一条短信的预览，当同一联系人有多个手机号时，通过此项能将不同对话区分开来。
- 5) 备份信息，包括备份时间及备份条数。

短信页面的人机界面设计见图 4-3。

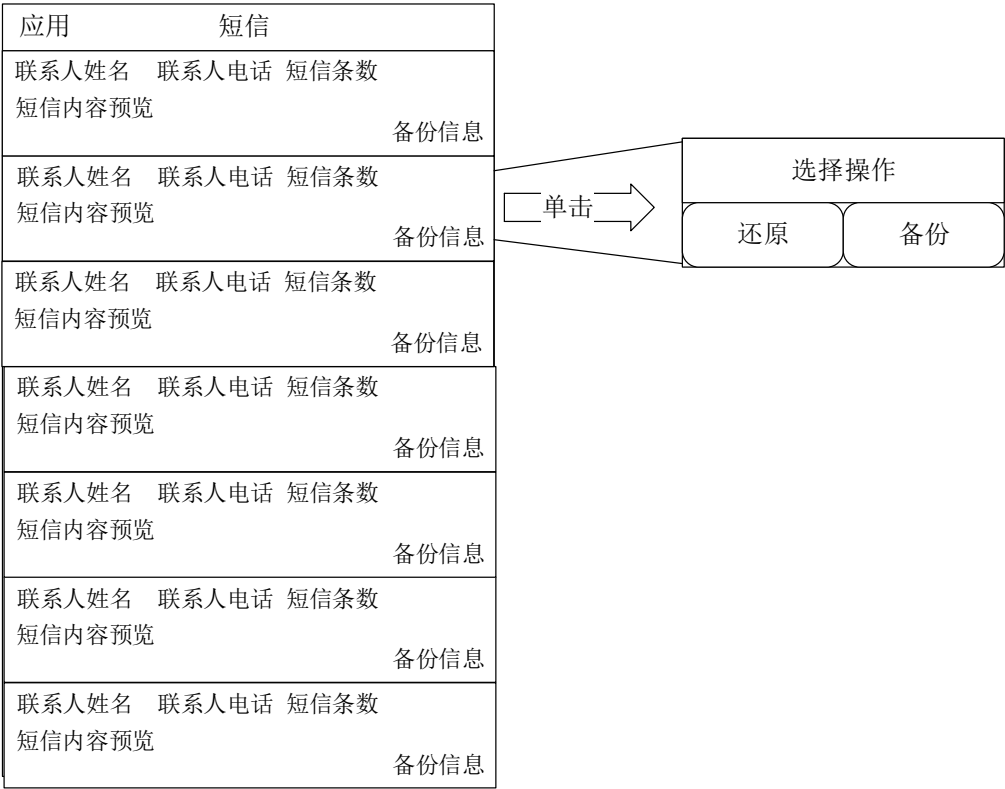


图 4-3 短信页面人机界面设计

4.2 模块详细设计

在 Android 系统中，由于对 UI 的操作不是线程安全的，即如果有多个线程同时修改 UI，则修改结果是无法预计的。因此对于 Android 系统中的应用，只有其主线程才能修改 UI。而由于实际操作的需要，往往会使用子线程来完成一些较耗时的工作，当工作完成后子线程往往会发出修改 UI 的需求。此时就要使用 Android 系统中的广播(Broadcast)或句柄(Handler)机制来进行线程间通信,使 UI 更改请求能发送给主线程。由于这个过程中，主线程与子线程是并发执行的，因此使用 UML 中的活动图来描述工作流程。

活动图是 UML 用于对系统的动态行为建模的一种常用图形，它描述了满足用例要求所要进行的活动以及活动间的约束关系，有利于识别并行活动。概括地说，活动图是由活动节点和转换流程所构成的图，它显示了控制权通过一次计算行为的各步骤流程。

4.2.1 系统状态检测模块详细设计

首页模块中检测系统状态的操作需要消耗一定的时间，且这段时间里不应该进行其他操作。因此，当进入应用时，首先首页被初始化，然后主线程开启一个新的线程在后台进行检测操作，而主线程本身弹出进度框等待操作完成。当子线程完成操作后，发送消息给主线程，此时主线程停止等待，并根据返回的结果修改首页的 UI。

该模块的操作的活动图如图 4-4 所示。

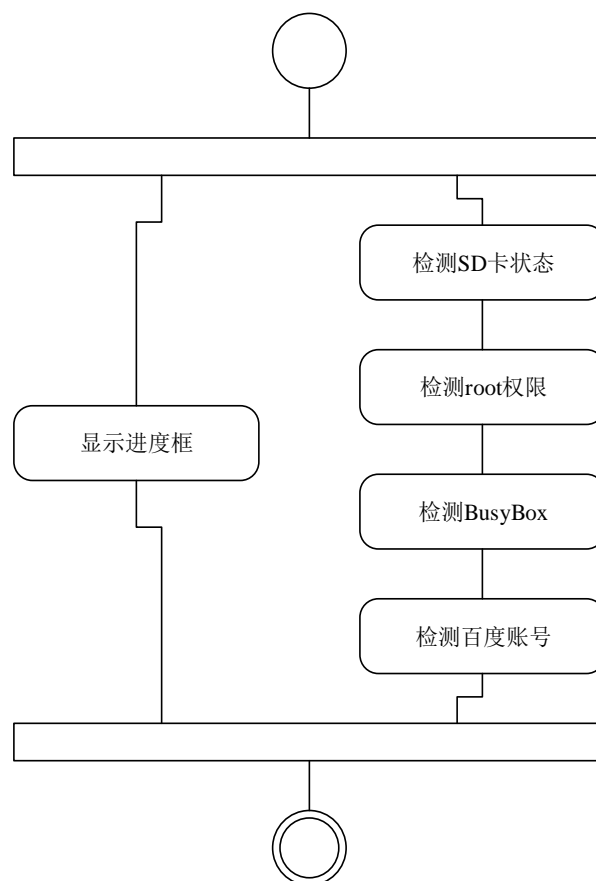


图 4-4 系统检测模块活动图

4.2.2 应用备份还原模块详细设计

在应用备份及还原模块的初始化阶段，首先要生成已安装应用列表，再获取 SD 卡中的备份文件，判断每一个备份文件是否为已安装的应用的备份。若是，则将备份文件的信息作为该应用的备份信息；若不是，则说明该备份文件所属的应用并未在本机上安装，

此时将该文件信息添加到列表末尾。

已安装应用的信息可通过其 `PackageInfo` 对象获得，该对象里存放着在应用的 `AndroidManifest.xml` 中收集到的所有信息。其中需要使用到的内容见表 4-1。

表 4-1 需要获取的应用信息字段

字段名	描述
label	应用名称
packageName	包名
versionName	版本
versionCode	版本号
icon	图标
sourceDir	应用安装文件所在目录
dataDir	应用数据文件所在目录

当点击列表中任意一项时，即弹出主要功能对话框。该对话框由 4 列 8 个按钮控件组成，第一列三个按钮控件负责对备份功能的调用，分别为全部备份、仅备份应用、仅备份数据。

类似的，第二列的三个按钮控件负责对还原功能的调用，分别为全部还原、仅还原应用、仅还原数据。

第三列为删除备份按钮，即将 SD 卡上的备份文件删除。

第四列为卸载应用按钮，即若该应用已安装，则将其卸载。卸载后的应用数据会被清除。

对于应用安装文件的备份，由于应用安装文件放置在 `/data/app` 目录下，用户具有对该目录的访问权限，因此可以使用 Java 中的 I/O 流函数直接将其备份到 SD 卡中。

对于数据文件的备份，则需以 root 权限执行 Shell 命令，将其打包为 `.tar.gz` 格式，因为 Android 系统对于 SD 卡并没有完整的权限管理，若直接将数据文件夹拷贝到 SD 卡上，则文件夹和文件的访问权限都会丢失，即使还原回 `/data/data` 目录也无法恢复，而 `.tar.gz` 格式能够完整地保存其访问权限信息。

为了管理上的方便，将备份后的应用文件和数据文件再打包为 `zip` 文件。同时创建 `xml` 格式的配置文件，保存备份时的相关信息。

当还原应用安装文件时，使用 `Intent` 机制，可以调用系统接口进行安装操作。

当还原数据时，由于在备份时已保存了完整信息，因此仅需把.tar.gz 文件直接解包即可。

4.2.3 短信备份还原模块详细设计

在短信备份还原模块的初始化阶段，需要读取系统短信数据库中的 threads 表和 sms 表，以及联系人数据库中的 contacts 表。需要从数据库中读取的数据条目见表 4-2。

表 4-2 Thread 表变量描述

表名	字段名	描述
sms	thread_id	threads 表中_id 字段的外键，用以表示短信所属的会话
	address	电话号
contacts	display_name	联系人姓名
	data1	联系人电话号
threads	_id	threads 表的主键，唯一表示一个会话
	message_count	会话中短信条数
	snippet	最新一条短信内容

可以看出，查找出的数据中有很多是冗余的，比如 sms 表中的 threads_id 和 threads 表中的_id 字段，以及 sms 表中的 address 字段和 contacts 表中的 data1 字段。这是为了解决使用 ContentResolver 接口访问数据库时无法多表连接查询的问题，方法是将数据读取出来后，自己写算法实现数据项的拼接。虽然 contacts 表和其他两个表并不在一个数据库中，但这三个表的关系可以使用图 4-5 中的 E-R 图来表示。

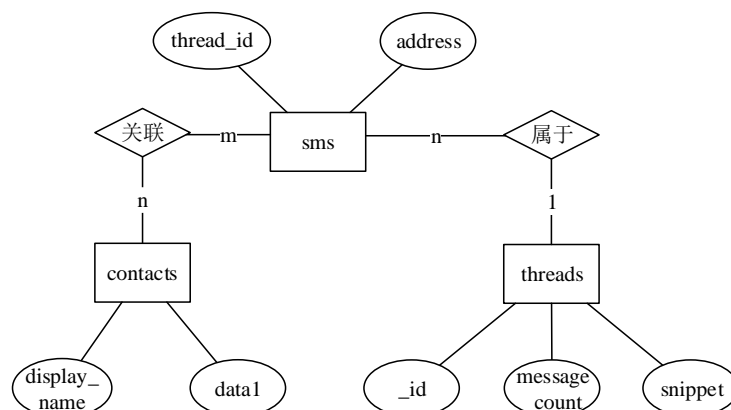


图 4-5 数据库 E-R 图

根据图 4-6 可以看出，只要首先查出 sms 表中的信息，就可根据已查出的信息再分别查询 contacts 表和 threads 表，实现连接查询的效果。具体流程为：首先查询 sms 表，

得到 thread_id 和 address 的列表, 然后对于列表中的每一项, 查询 contacts 表, 根据 address 与 data1 的匹配得到 display_name; 查询 threads 表, 根据 thread_id 和 _id 的匹配得到 message_count 和 snippet。

短信备份的流程为: 根据用户所选择要备份的项, 获取该项的 thread_id, 然后查找短信数据库的 sms 表, 将返回的结果写入到 xml 文件中。其中需要备份的字段见表 4-3。

表 4-3 需备份的短信字段

变量名	描述
address	电话号码
date	发送时间
protocol	协议类型, 默认均为 0
read	已读标记
status	短信状态
type	1 表示接收, 2 表示发送
reply_path_present	回复路径, 默认为 0
body	正文
locked	锁定标志
error_code	错误代码
seen	已读标记

xml 文件的结构如下:

```
<thread person="联系人姓名" address="电话号码" number="备份数量" backupTime="备份时间">
```

```
<sms>
```

```
<address>电话号码</address>
```

```
<date>发送时间</date>
```

```
<protocol>协议</protocol>
```

```
<read>已读标记</read>
```

```
<status>短信状态</status>
```

```
<type>短信类型</type>
```

```
<reply_path_present>回复路径</reply_path_present>
```

```
<body>正文</body>
```

```
<locked>锁定标志</locked>
```



```
<error_code>错误码</error_code>
<seen>已读标志</seen>
</sms>
</thread>
```

其中，联系人姓名、电话号码和短信正文需要使用算法加密后再写入文件。

当短信还原时，从 xml 文件中依次读取短信信息，每读完一条，首先将信息解密，然后先查询数据库中是否已存在该条短信，若不存在，则将其插入到数据库中。这样可以防止同一条短信出现多次。

4.2.4 百度云 PCS 模块详细设计

百度 PCS 模块需实现的功能有登录、上传和登出。由于该模块的功能都是用来被其他模块调用，因此可将其写为工具类。

登录和登出的功能在主页中调用。当用户未登录时，显示登录按钮，点击登录按钮后，跳转到百度账号登录页面，用户输入账号密码并登录成功后，将获得的令牌保存在数据库中。当用户已登录时，显示登出按钮，点击登出按钮后，向百度服务器发出登出消息，同时删除本地数据库中的令牌信息。由于应用判断百度账号登录与否的唯一标志即是数据库中是否存在令牌，因此删除令牌后，即可认为已经登出。

上传操作在短信备份还原模块中在后台被调用。当确认存在令牌时，则在短信备份完成后，将 xml 格式的备份文件上传到网盘中。

5 代码实现与测试

5.1 代码实现

5.1.1 首页模块代码实现

首页模块使用 HomeFragment 类来实现，该类重载自 Fragment 类。在 onCreate() 方法中，开启新线程进行系统状态检测，该线程类即 CheckStatusTask 类。

CheckStatusTask 类中，获取到的系统状态存放在 Status 类的一个对象中。Status 类是专门用来存放系统状态变量的类，获取到系统状态后，可以通过该类的对象将系统状态变量封装起来。

检测 SD 卡是否成功挂载的代码为：

```
if ( Environment.getExternalStorageState().equalsIgnoreCase(
    Environment.MEDIA_MOUNTED) )
    status.setSdcard(true);
```

调用 Android 系统库函数中外置存储设备状态的方法，若返回的结果为 MOUNTED 状态，即证明 SD 卡已成功挂载。

检测 root 权限的代码为：

```
if ( ShellUtil.RootCmd("") )
    status.setRoot(true);
```

ShellUtil 是为了方便执行 Shell 命令而封装的一个工具类，RootCmd() 方法的作用是以 root 权限执行参数中的 Shell 命令。在此处传入的参数是空字符串，意味着仅获取 root 权限，不执行任何命令。若 root 权限获取成功，则会返回 true，失败则返回 false。

检测 BusyBox 是否安装的代码如下：

```
if ( ShellUtil.Cmd("busybox") )
    status.setBusybox(true);
```

ShellUtil.Cmd() 方法的作用是以普通权限执行 Shell 命令，适用于不需要 root 权限也能访问的地方。此处传入的参数为 busybox，若已安装 BusyBox，则返回 true，否则返回 false。

检测百度账号是否登录的代码如下：

```
if ( DBUtil.query(activity).length() != 0 )
```

```
status.setBae(true);
```

DBUtil 类是用来访问应用内部数据库的工具类。query()方法的作用是查找数据库中的令牌信息，若令牌不存在，则返回一个空字符串。因此若返回信息的长度不为 0，即证明已登陆百度账号。

5.1.2 应用备份还原模块代码实现

在应用备份和还原模块中，使用 AppInfo 类来存储一个应用的信息，完成对该应用的相关操作。AppInfo 类中的变量和主要方法见表 5-1。

表 5-1 AppInfo 类主要变量和方法

名称	描述
name	应用名称
packageName	应用包名
version	版本名称
versionCode	版本号
icon	图标
appPath	应用安装路径
dataPath	应用数据路径
backupTime	备份时间
type	备份类型 TYPE_LOCAL 表示该应用无备份 TYPE_SDCARD 表示该应用有备份，但没有被安装 TYPE_LOCAL_SDCARD 表示该应用已安装，且有备份
mode	备份模式 MODE_APP_DATA 表示全部备份 MODE_APP_ONLY 表示仅备份应用 MODE_DATA_ONLY 表示仅备份数据
backup()	备份该应用的方法
restore()	还原该应用的方法
delete()	删除备份的方法
uninstall()	卸载该应用的方法

其中 backup()方法和 restore()方法均需要打开新的线程，在子线程中执行相关操作。应用备份功能的主要代码如下：

```
String originalAppPath = appInfo.getAppPath();
```

```
String originalDataPath = appInfo.getDataPath();
```

```

String backupPath = Environment.getExternalStorageDirectory()
    + "/SimpleBackup/" + appInfo.getPackageName();
if ( mode != MODE_BACKUP_DATA )
    CopyUtil.copyFile(originalAppPath, backupPath + ".apk");
if ( mode != MODE_BACKUP_APP )
{
    String cmd = "busybox tar zcvf " + backupPath + ".tar.gz "
        + originalDataPath;
    ShellUtil.RootCmd(cmd);
}

```

在这一段代码中，CopyUtil.copyFile()方法即是使用 I/O 流函数拷贝文件。而数据的备份方法则是使用 RootCmd()方法以 root 权限将数据文件夹打包为.tar.gz 格式。

当应用及数据被拷贝到 SD 卡后，需要再将其打包为 zip 文件，代码如下：

```

ArrayList<File> resFileList = new ArrayList<File>();
File apk;
if ( mode != MODE_BACKUP_DATA )
{
    apk = new File(backupPath + ".apk");
    if ( apk.exists() )
        resFileList.add(apk);
}
File data;
if ( mode != MODE_BACKUP_APP )
{
    data = new File(backupPath + ".tar.gz");
    if ( data.exists() )
        resFileList.add(data);
}
File zipFile = new File(backupPath + ".zip");
ZipUtil.zipFiles(resFileList, zipFile);

```

ZipUtil 类是对 zip 文件进行操作的工具类。zipFiles()方法的功能是将多个文件打包成一个 zip 文件，存放在指定路径下。当 zip 文件生成后，原有的 apk 文件和 tar.gz 文件便不再需要了，因此需要删除掉这两个文件。代码为：

```
String cmd = "rm -r ";
if ( mode != MODE_BACKUP_DATA )
    cmd += backupPath + ".apk ";
if ( mode != MODE_BACKUP_APP )
    cmd += backupPath + ".tar.gz";
```

```
ShellUtil.Cmd(cmd);
```

应用备份工作的最后一步是创建 xml 格式的配置文件，主要代码如下：

```
File xmlFile = new File(backupPath + ".xml");
FileOutputStream outputStream = new FileOutputStream(xmlFile);
OutputStreamWriter outputStreamWriter = new OutputStreamWriter(
    outputStream, "UTF-8");
BufferedWriter writer = new BufferedWriter(outputStreamWriter);
XmlUtil.writeAppCfg(writer, appInfo);
writer.flush();
writer.close();
```

xml 文件是一种文本文件，因此直接使用 OutPutStreamWriter 即可创建。而对于 Writer 内容的写入，则是通过 XmlUtil 工具类中的 writeAppCfg()方法来实现的。XmlUtil 类是完成对 xml 文件操作的工具类，writeAppCfg()方法的功能是创建应用备份配置文件。

应用的还原功能与备份操作是刚好相反的。首先解压 zip 格式的备份文件，若备份文件中包括应用安装文件，安装该应用；若包括数据文件，则解压 tar.gz 格式的数据文件即可。解压缩 zip 文件是调用 ZipUtil 类中的 unzipFile()方法，当 zip 文件解压后，若解压出的文件中包括应用安装文件，则首先安装应用。安装应用的代码如下：

```
Intent intent = new Intent(Intent.ACTION_VIEW);
intent.setDataAndType(
    Uri.fromFile(new File(restorePath
        + appInfo.getPackageName() + ".apk")),
    "application/vnd.android.package-archive");
```

```
activity.startActivity(intent);
```

可以看到这里使用了 Intent 机制与系统进行通信，调用安装应用的功能。

若解压出的文件中包括数据文件，由于数据文件打包为 tar.gz 格式，原始路径和权限都完好保留，因此直接调用 Shell 命令将数据文件解压即可。数据还原的代码如下：

```
String cmd = "busybox tar zxvf " + restorePath  
            + appInfo.getPackageName() + ".tar.gz";  
ShellUtil.RootCmd(cmd);
```

当还原完成后，还原过程中产生的临时文件也需要清除，此处的操作与备份过程中清除临时文件的操作是一致的。

在备份和还原过程中，需要对程序的 UI 进行修改，具体的工作为：当备份开始时，在通知栏放置一条不可移除的通知，同时应用主界面弹出进度对话框。对话框的作用有两个，其一是提示用户的当前工作状态，其二是阻止用户进行其他操作，因为此时进行其他操作可能会对后台工作造成影响。当备份完成后，通知栏显示完成通知，同时关闭对话框。还原操作中的 UI 操作与备份操作是类似的。

由于修改 UI 只能在线程中进行，因此为了实现这样的功能，需要使用 Handler 来进行异步消息的处理。在线程中，声明一个 Handler，并令其根据不同的消息类型执行不同的操作，如修改 UI 等。当创建子线程时，将 handler 传进子线程，若需要主线程进行某些操作，则只需发送相应的消息给 Handler 即可。首先在主 Activity 中声明一个 Handler，在 Handler 的 handleMessage()方法中，可以根据捕获到的消息进行相应的处理。

5.1.3 短信备份还原模块代码实现

在短信备份还原模块中，使用 ThreadInfo 类来存储一个对话的信息，使用 SmsInfo 类存储一条短信的内容。

短信的备份、还原操作均需在新的线程中执行。在该模块中，使用 AsyncTask 类来实现多线程。由于 AsyncTask 类在 onPreExecute()方法、onProgressUpdate()方法和 onPostExecute()方法中均能直接修改 UI，因此无需使用 Handler 也可达到相同的目的。

短信的备份工作在 SmsBackupTask 类中实现。首先需要根据用户选择的 ThreadInfo 的信息，在短信数据库中查询该对话所对应的全部短信。具体代码为：

```
String[] selectionArgs = { threadInfo.getThreadID() };  
Cursor cursor = contentResolver.query(SMS_URI, null, "thread_id=?",  
                                     selectionArgs, "date DESC");
```

```

if ( cursor != null )
{
    ArrayList<SmsInfo> smsInfos = new ArrayList<SmsInfo>();
    SmsInfo smsInfo;
    int num = 1;
    while (cursor.moveToNext())
    {
        onProgressUpdate(num);
        smsInfo = new SmsInfo();
        // 将当前条目写入到 smsInfo 对象中，此处代码略去
        smsInfos.add(smsInfo);
        num++;
    }
}

```

当完成此步操作后，在 `smsInfos` 列表中就存放着已选择对话的所有信息。为了安全性考虑，联系人姓名、电话号码及短信正文均加密后进行存放。这里使用的是较容易实现的 BASE64 算法，加密操作可通过调用 `EncryptionUtil.encrypt()` 方法来实现。接下来即可将这些信息写入到 xml 文件中，该功能使用 `XmlUtil` 工具类中的 `writeSMSstoXML()` 方法实现。当备份文件写入完成后，检测是否登录百度账号，若已登录，则将该备份文件上传到云端。上传操作通过调用 `BAEUtil.upload()` 方法实现。

短信的还原工作在 `SmsRestoreTask` 类中实现。首先读取 xml 文件，将文件中的数据储存为列表形式，然后再将这些数据依次插入短信数据库中。从 xml 文件中读取数据在 `XmlUtil.readSMSfromXML()` 方法中实现。向短信数据库中插入数据使用 `ContentResolver` 中的 `insert()` 方法。

5.1.4 工具类内容介绍

1) 百度 API 工具类 `BAEUtil`

该工具类主要实现百度账号的登录、登出及文件的上传操作。主要方法介绍见表 5-2。

表 5-2 BAEUtil 类主要方法

方法名	描述
login()	登录百度账号
logout()	登出
mkdir()	登录百度账号成功后，创建云端备份文件夹
upload()	上传文件

2) 文件拷贝工具类 CopyUtil

该类封装了拷贝文件相关的操作，主要方法见表 5-3。

表 5-3 CopyUtil 类主要方法

方法名	描述
copyFile(String from,String to)	以普通权限复制文件，将 from 路径指定的文件复制到 to
copyWithRoot(String from,String to)	以 root 权限复制文件，将 from 路径指定的文件复制到 to

3) 内部数据库操作工具类 DBUtil

应用内部数据库是为了存放百度云 PCS 所需的令牌信息。当应用刚打开时，需检测数据库是否创建，若未创建则创建。登录百度账号成功后，将获取到的令牌存入数据库。短信备份工作完成后，将备份文件上传到云端。登出时，删除本地令牌信息。该类的方法简介见表 5-4。

表 5-4 DBUtil 类主要方法

方法名	描述
create()	创建数据库
insert()	将令牌信息存入数据库
query()	从数据库中获取令牌
delete()	删除令牌

4) 加密解密工具类 EncryptionUtil

该类的作用是在向 xml 文件写入数据时对其加密，并在读取 xml 文件中的数据时进行解密。该类的主要方法简介见表 5-5。

表 5-5 EncryptionUtil 类主要方法

方法名	描述
-----	----

encrypt(String text,int mode)	将 text 字符串，根据 mode 指定的加密算法加密。返回值是加密后的结果
decrypt(String text,int mode)	将 text 字符串，根据 mode 指定的加密算法进行解密，返回值为解密后的结果

5) Shell 命令操作类 ShellUtil

该类封装了执行 Shell 命令所需的相关操作，主要方法简介见表 5-6。

表 5-6 ShellUtil 类主要方法

方法名	描述
Cmd(String cmd)	以普通权限执行 Shell 命令
RootCmd(String cmd)	以 root 权限执行 Shell 命令

6) xml 文件操作类 XmlUtil

该类封装了对 xml 文件的相关操作。主要方法简介见表 5-7。

表 5-7 XmlUtil 类主要方法

方法名	描述
readAppCfg()	读取应用配置文件
readSMSfromXML()	从 xml 文件中读取短信信息
readThreadInfo()	从 xml 文件中读取会话信息
writeAppCfg()	写入应用配置文件
writeSMStoXML()	将短信数据写入到 xml 文件

7) zip 文件操作类 ZipUtil

该类封装了对 zip 文件的相关操作。主要方法简介见表 5-8。

表 5-8 ZipUtil 类主要方法

方法名	描述
zipFile(File resFile,ZipOutputStream zipout,String rootpath)	备份文件。其中 resFile 为需要打包的文件，zipout 为需要压缩的目的文件夹，rootpath 为备份文件路径
zipFiles(Collection<File> resFileList,File zipFile)	批量备份文件。将 resFileList 中保存的文件统一打包到 zipFile 中
upZipFile(File zipFile,String folderPath)	解压缩文件，将 zipFile 文件解压缩到 folderPath 文件夹中

5.2 测试

5.2.1 功能测试

为了进行表 5-9 为功能测试测试用例清单。

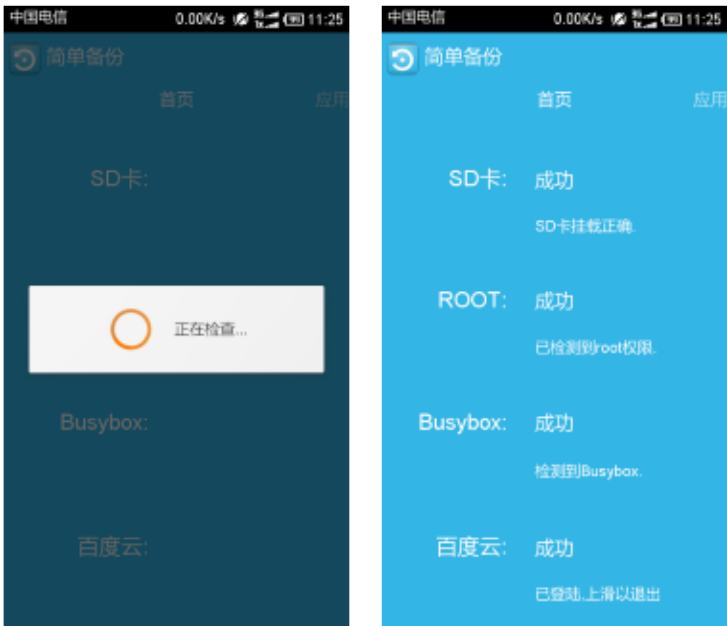
表 5-9 功能测试用例表

序号	功能点	关注点
1	系统状态检测	能否正确获取相关信息并显示
2	已安装应用信息读取	能否获取已安装应用信息并显示
3	应用备份	能否正确备份应用及数据
4	应用还原	能否正确还原应用及数据
5	短信信息读取	能否正确读取系统短信数据库并显示相关信息
6	短信备份	能否正确备份短信
7	短信还原	能否正确还原短信
8	百度云网盘上传	能否正确将文件上传到百度云网盘

1) 系统状态检测

测试方法：通过应用在设备上执行，根据显示的结果检测功能是否正常。

测试结果：测试通过，系统状态检测功能正常，运行结果见图 5-1。



(a)打开应用时检查系统状态 (b)检测完成，显示首页

图 5-1 系统状态检测功能

2) 已安装应用信息读取

测试方法：通过应用在设备上运行，根据显示的结果检测功能是否正常。

测试结果：测试通过，已安装应用信息读取正常。运行结果见图 5-2。



图 5-2 已安装应用信息读取

3) 应用备份

测试方法：运行应用中的备份功能，查看备份文件是否正常创建。

测试结果：测试通过，应用备份功能正常，运行结果见图 5-3。



图 5-3 应用备份功能测试

4) 应用还原

测试方法：运行应用中的还原功能，查看是否被正确还原。

测试结果：测试通过，应用安装文件及数据文件均被成功还原。运行结果见图 5-4。



图 5-4 应用还原功能测试

5) 短信信息读取

测试方法：通过应用在设备上运行，根据显示的结果检测功能是否正常。

测试结果：测试通过，短信信息读取正常。运行结果见图 5-5。



图 5-5 短信信息读取

6) 短信备份

测试方法：运行应用中的备份功能，查看短信备份文件是否被正常创建。
测试结果：测试通过，短信被成功备份到 xml 文件中。运行结果见图 5-6。



图 5-6 短信备份功能测试

7) 短信还原

测试方法：运行应用中的还原功能，查看备份的短信是否被正确恢复。
测试结果：测试通过，短信被成功还原。运行结果见图 5-7。



图 5-7 短信还原功能测试

8) 百度云网盘上传

测试方法：运行百度账号登陆功能，并检测登录后是否能成功上传备份文件。

测试结果：测试通过，百度云网盘连接成功，备份文件上传成功。运行结果见图 5-8。



图 5-8 百度云网盘上传测试

5.2.2 性能测试

在本应用中，着重对短信备份及还原部分的执行效率进行了优化。与现有的同类应用 SMS Backup&Restore 进行对比，对比结果如表 5-10。

表 5-10 性能测试结果

短信条数	SMS Backup&Restore 备份时间	本应用备份时间
47	1 秒	1 秒
300	2.5 秒	1 秒
511	3.5 秒	1 秒
2903	16 秒	3 秒
5744	33 秒	6 秒

可以看出本应用的短信备份性能要远远优于同类产品。

6 结束语

本应用使用 Eclipse 作为 IDE 开发完成，完成了一个移动平台中备份还原软件的基本功能，同时在执行效率和安全性等方面做出了自己的一些努力。

本着学习和研究的态度，在分析过同类应用之后，我尝试着按照自己的想法去实现一个 Android 平台的备份还原软件。虽然通过分析、设计、实现进行了简单的实现，但是还是可以以当前的项目为基础，进行进一步的完善，增加新的功能，从而实现一个功能完整的程序。

在学习和研究了 Android 系统中的应用存储机制和短信存储机制后，我使用 Eclipse 作为开发工具，编写了一款完成了应用备份还原、短信备份还原等基础功能的备份软件。其中应用备份功能可以将应用安装包及数据文件打包成 zip 格式存放到 SD 卡上；短信备份功能将需要备份的短信保存为 xml 格式，并对其中的内容进行加密。短信备份文件同时可以备份到百度云网盘中。

通过这段时间的研究和学习，基本实现了备份还原程序最核心的功能。但是在流程控制、错误处理、异常检测等方面还有一些不足。只有做到了这些，应用才能更适合在实际中使用。接下来，就对应用能够提高和优化的地方做一个展望和说明。

首先，本应用只能对用户安装的应用进行备份还原，不支持备份和还原系统应用，在以后可以考虑增加系统应用的备份和还原功能。其次，在短信还原时，还原速度和同类应用几乎一致，并没有性能上的优势，可以考虑对这部分代码进行优化，提高速度。最后，短信备份时，可以考虑改为增量备份，即若存在备份文件，则只增加备份文件中没有的部分，这样可再次提高备份时的速度。

在本应用编写过程中，我认识到了编码风格和注释的重要性。良好的编码风格和完整的注释不光能使他人阅读代码变得方便，自己修改时也能很容易地找到需要修改的功能点。同时，善用版本控制软件，能够为代码修改提供很多方便，后期总结时，也能根据每次版本的更新内容整理出开发流程。总的来说，养成优良的编码习惯，学会使用多种辅助工具，才能写出健壮的代码和程序。

致 谢

转眼间，大学四年的生活就要告一段落。回首这四年的大学生活，有很多的留恋和不舍。在这四年中，遇到了很多困难和不解，是热心、友好的老师和同学，在我生活和学习上遇到困难时，一次次伸出援助之手，一次次耐心地帮我解决问题，答疑解惑。在你们的帮助和鼓励之下，我学到的不只是知识和专业技能，更有做人的道理，让我学会了怎样去做一个做事认真、态度积极的人。在这里，向曾经帮助过我的老师和同学们表示感谢。

历时一个多月的毕业设计和论文写作工作终于完成了。在这段时间里，感谢各位老师同学对我的指导。我特别要向知识渊博、治学严谨、孜孜不倦的付燕老师表示衷心的感谢。付老师本着认真、负责、细心、的治学态度，耐心的指导、及时监督，为我们毕业设计的完成和论文的写作进行了很好的指导，并提出了建设性的建议和宝贵的意见，为我们毕业设计保质保量地完成提供了坚实的后盾。在毕业设计这一段时间和同学的交流过程中，我认识到合作和沟通交流的重要性。同时也认识到了在软件开发过程中甚至是学习的过程中需要秉承的严谨的态度和积极向上的态度。

在论文中难免会有许多纰漏和错误，希望各位老师和同学不吝赐教。

参考文献

- [1] 王向辉,张国印,赖明珠.Android 应用程序开发(第二版)[M].清华大学出版社,2012.1~2
- [2] 肖梓航.再谈 Android 软件的安全开发[J].程序员,2013.3
- [3] (美)迈耶.余健伟,赵凯.Android 4 高级编程[M].清华大学出版社,2013
- [4] 范怀宇.Android 开发精要[M].机械工业出版社,2012
- [5] 曾健平,邵艳洁.Android 系统架构及应用程序开发研究.微计算机信息,2011.9
- [6] 郑萌.Android 系统下 Java 编程详解.电子工业出版社,2012
- [7] 余志龙,陈昱勋等.Google Android SDK 开发范例大全[M].人民邮电出版社,2009
- [8] Grant Allen, Mike Owens.杨谦,刘义宣,谢志强.SQLite 权威指南(第二版)[M].电子工业出版社,2012
- [9] Elliotte Rusty Harold. Java 语言与 XML 处理教程[M].电子工业出版社,2004
- [10] 李刚.疯狂 XML 讲义[M].电子工业出版社,2009
- [11] Sandra E.Eddy,B.K.Delong.XML 精要:语法详解与编程指南.清华大学出版社,2002
- [12] 梁栋.Java 加密与解密的艺术[M].机械工业出版社,2010
- [13] 张海藩.软件工程导论(第五版)[M].清华大学出版社,2011
- [14] 朱少民.软件测试方法和技术(第二版)[M].清华大学出版社,2010
- [15] Richard C. Lee. UML 与 Java 面向对象开发实践[M].清华大学出版社,2003