学号：1145337056

河南大学2015届本科毕业论文

基于安卓的PDF文件浏览与格式转换应用的实现

PDF file browsing and Converter base on Android platform

论文作者姓名： 邰景磊

作 者 学 号： 1145337056

所 在 学 院： 软件学院

所 学 专 业： 软件工程

导师姓名职称： 梁胜彬（副教授）

论文完成时间： 2015年4月20日

2015年5月20日

# 摘 要

近年来，随着移动互联网在全球范围内的快速发展和普及，移动设备尤其是智能手机在人们生活中占据着越来越重要的地位，如今随着谷歌开源android系统的推出，android 手机已经迅速占据了智能手机的半壁江山。基于android系统的各种app也纷纷推出，不仅涉及有娱乐、购物、社交等方向，移动办公领域也越来越被人们关注。移动办公与IT业在[软件](http://baike.baidu.com/view/37.htm)应用上的成熟、业务内容上的丰富，完美结合到了一起，使之成为了继电脑[无纸化办公](http://baike.baidu.com/view/1139015.htm)、[互联网](http://baike.baidu.com/view/6825.htm)远程化办公之后的新一代办公模式。

PDF文档的英文全称为Portable Document Format，可以翻译为可移植文档格式。PDF文档格式是一种电子文件格式，它有众多优点，其中比较受到广大用户欢迎的是这种格式的系统独立性，它不依赖于操作平台，因此PDF文档在Lunix、Windows、Mac OS操作系统中都是通用的。也正因为它的这一优点使得PDF文档成为了在电子文档发行和传播中的佼佼者。现在大多数的电子书、电子资料、邮件以及一些公文告示都选择了使用PDF文档。有人评价PDF文档说它几乎已经成为了电子信息文档的工业标准。由此可见在android设备上实现PDF文档的阅读的意义。有时人们为了打印文档或者发送纸质不能快速转化为计算机文本的资料的时候只能选择手打内容或者使用扫描仪，现在的android设备都具有分辨率极高的摄像头，实现拍照、截屏转化成图片甚至PDF文档就相当于人们随身携带者一台高性能的扫描仪，这将为人们在办公、生活中带来极大的方便。

本论文主要研究、实现以下问题：

1. 在android设备上实现PDF文档的浏览阅读。

2. 实现拍照、截屏生成图片、生成PDF文档。

3. 设计一款操作体验良好的android app，不仅仅给用户提供便利的功能还要提供给用户友好、易用、简洁的用户体验。

关键词 Android；PDF阅读；app；拍照；截屏；生成PDF

# ABSTRACT

With the rapid development and globalization of the mobile Internet the word , the mobile devices especially the smart phones have recently occupied a more and more important position in our daily life. Now with the launch of the Google’s android system ,the android phones take more than half of the smart phones’ market. After launch of the various android apps in such as entertainment, shopping and social communication, the mobile office is more and more attracted by people. With the mature of IT industry in the software application and in the business, the Mobile office combines with the IT industry very well. Which makes the mobile office to be a new model following the computer paper-less office and remote Internet office mode.

PDF for Portable Document Format, is an electronic file format. This file format has nothing to do with the operating system platform, that is to say, the PDF files can be used in Windows, Unix or Mac OS systems. This performance make s it on the Internet for the electronic document issued and ideal document format of digital in formation transmission. More and more electronic books, product description, company proclamation, network information, and E-Mail ,are starting to use PDF files. A PDF file has become an industry standard digital information in fact. Thus it can be seen on the android device to realize a PDF of the significance of reading. In order to print documents or send paper, sometimes people fast text data into a computer only by choosing to hand play content or using scanner, now android devices have extremely high resolution camera, take photos, screenshots into images. A PDF document is equivalent to people with a high performance of scanner, this will be convenience to people in the office and life.

This thesis aims for studying and implementing the following questions:

1, To realize the PDF document browsing reading on the android devices;

2, To make the photos, screenshots into images then into a PDF document;

3, To design a good android app for the customers to get a new feeling of operate it easily and simply

Key words : Android, PDF reading, The app, Taking photos, Screenshots and Being a PDF

# 第1章 绪 论

Android这个词，原本没有什么特别的含义，google形象的把它推出的新的只能操作系统命名为Android（机器人）。它是一个开源的操作系统，主要有五部分组成，分别是操作系统、中间件、用户界面和应用软件。google生成Android是移动平台系统的一个真正开放的。它包括操作系统、用户界面和应用程序－移动电话工作所需的全部软件，而且不存在任何以往阻碍移动产业创新的专有权障碍。谷歌与开放手机联盟合作开发了 Android，这个联盟由包括中国移动、摩托罗拉、高通、宏达和 T-Mobile 在内的 30 多家技术和无线应用的领军企业组成。Google公司推出的Android系统与各大手机制造商、各大手机信号运行商以及其他有关组织及厂商合作，共同来维护Android这个标准、开放的移动软件平台，

使得Android在行业中有了不可撼动的地位，由此Android应用也在通用性和可通讯复制性上有了极大的优势。

当前移动网络技术以及互联网技术的快速发展、基于android系统的智能手机的普及以及PDF文档在电子信息业的传播，移动办公和在移动设备上操作PDF文档越来越被人们关注。目前无纸化办公和PC办公已经不能满足现在快速工作、生活的节奏。该文提出了设计实现一个基于android系统的能快速阅读PDF文档，随手抓拍图片生成图片、PDF文档的app。开发这样一款app能提高人们的办公效率、也能帮助人们快速保存身边重要的文件。

## 1.1 课题背景

近年来，随着移动互联网技术、android 系统的飞速发展，android智能手机、和便捷的3G、4G网络的覆盖，人们对手机软件的依赖越来越强烈，越来越多的人们选择使用移动设备办公，如今市面上零星有些办公助手或者文档助手的app,但是大多数功能单一，不能满足绝大多数用户的需求，用户期望一款能集阅读、生成、拍照、截图等功能于一身的app。

## 1.2 国内外在该方向的研究现状及分析

近年来，随着科学技术的迅速发展，手机的应用已经不仅仅局限于日常的通话聊天，现在的手机功能已经和日常生活、日常办公等紧紧地联系在一起。实现了以前只能在电脑上才能完成的事，比如说：购物、交话费、数据传输、信息交流等，还可以在手机上完成一些简单的办公。手机的普遍性和轻巧易携带性，决定了手机会受到越来越多人的青睐，市场也会随着手机功能的慢慢全面而需求空间不断发展，市场最终也会慢慢扩大。如果说最近几年什么系统最火爆，那么毫无疑问是安卓系统。从Google推出Android系统，到一款基于Android手机的问世并且迅速走红，全球各大移动设备厂商也对Android的前景、形式看好，纷纷加入到了Android设备研发的大军中。目前Android系统已经在手机、Pad、汽车、智能家居等行业中迅速发展，可预见以后Android所涉及的领域会越来越多，前景也是十分广阔。也正是在这样的背景下，Android APP也势必迎来自己的发展高峰，这必然是一个巨大的市场。

由于Android的开放性和稳定性以及易于扩展性，许多厂商已经有了相当成熟的产品，这些产品涉及通讯、交通、家居等行业，并且迅速抢占的了市场。免费开源是Android之所以有如此发展的最大原因，Android不仅仅不限制第三方软件在其平台上的运行，开发者在开发App的时候有很大的自由与，甚至可以按照自己的思路去开发扩展、相互分享。国内厂商和运营商纷纷加入了Android的阵营，包括中国移动，中国联通，华为等，近几年迅速崛起的小米也是基于Android系统进行二次定制开发的。国内还出现许多App生产“工厂”。Android平台在文档阅读器上的应用将向个性化、易用性、全面性、开辟更广阔的道路。  
随着我国移动3G技术的普及，以及Android平台的开源优势和强大功能与手机的便携特点相结合，使得移动智能计算平台（如智能手机、平板电脑等）日益成为用户日常学习、生活和办公工具。

在此领域，国内用户拥有的文件查看器资源不能说不丰富。上百度应用搜索“PDF文件查看器”，出现的搜索结果中，几乎每个应用可以下载，不过内容涉及单一，只是针对某一种格式的查看，看似无所不包的查看器，却没有广泛性和指向性，使用价值却是让人担心。大部分用户都想拥有一款能够满足自己所有要求的查看器。

## 1.3 技术介绍

### 1.3.1 Android 简介

Android这个单词本身并没有什么特表的含义，是Android开发工程师，结合自己的名字以及Android的机器人Logo涉及的。Android平台主要有四大部分组成：操作系统、中间件、用户界面和应用软件。

Android一词最早出现于法国作家利尔亚当（Auguste Villiers de l'Isle-Adam）在1886年发表的科幻小说《[未来夏娃](http://baike.baidu.com/view/4117214.htm)》（L'ève future）中。他将外表像人的机器起名为Android。

Android的Logo是由Ascender公司设计的。其中的文字使用了Ascender公司专门制作的称之为“Droid ”的字体。Android是一个全身绿色的[机器人](http://baike.baidu.com/view/2788.htm)，[绿色](http://baike.baidu.com/view/23550.htm)也是Android的标志。[颜色](http://baike.baidu.com/view/19878.htm)采用了PMS 376C和RGB中[十六进制](http://baike.baidu.com/view/230306.htm)的#A4C639来绘制，这是Android操作[系统](http://baike.baidu.com/view/25302.htm)的[品牌](http://baike.baidu.com/view/9013.htm)象徵。有时候，它们还会使用纯文字的[Logo](http://baike.baidu.com/view/457.htm)。

2012年Android操作系统获得了世人的绝对认同，它被国外一著名的科技网站评为了二十一世纪十款最重要电子产品之一，同iPhone齐名。

### 1.3.2 Android架构

Android的架构是由Linux内核、C/C++函数库、Dalvik虚拟机、应用框架以及关键应用程序构成的。通过编译基于框架的应用程序可以降低开发负荷。应用软件原则上是在DalvikVM上运行的。



图1.0 Android架构

### 1.3.3 Android特征

应用框架：用于开发可重用、可替代的组件。

Dalvik虚拟机：由于Android是由Java语言开发的，Google摒弃了JVM的缺点，保留了JVM的优点，自主开发了Dalvik虚拟机。它也相当于是对JVM的更新和改进，Dalvik虚拟机具有耗资源少、系统自动回收垃圾、运行效率高于JVM。

内嵌浏览器：基于开源的浏览器网页排版引擎WebKit。它是Google对开源的Html引擎WebKit封装开发的。

优化的图形：强大的自定义二维图形库。基于OpenGL ES 1.0标准的3D图形功能，在移动3D开发中有着至关重要的地位。

SQLite数据库：SQLite提供结构化的数据存储。与其他主流的服务端数据库的设计以及实现方式都很相似，而且它是与你的用户信息绑定的嵌入式数据库。与java web中的数据相比想好的资源和内存都远远小于服务端数据库，但是处理速度和所支持的功能却不亚于服务端数据库。

多媒体支持：支持各种通用的音频、视频和静态图像格式，如MP3、JPG、PNG等。Android的多媒体处理核心框架为Open Core，Android平台中所有的音频、视频的播放等操作都是有它提供的。目前许多Android的多媒体处理框架都是通过它实现的，开发者也可以通过其API快速的处理Android中的多媒体文件。

GSM技术：这就是全球通技术，是第二代移动通讯技术，目前几乎所有的手机或者移动通讯设备都是通过它实现的。它的实现对硬件设备有很大的依赖，Android同样也提供了GSM技术，通过GSM技术实现了电话通讯，同时也向开发者提供了一定的开发接口。

蓝牙，EDGE, 3G and WiFi：（依赖硬件）蓝牙和Wi-Fi都是一种短距离的无线连接技术，在短距离内可以实现话音和数据的无线传输。EDGE(Enhanced Data Rate for GSM Evolution),是增强型数据速率GSM演进技术。

Cameram, GPS, compass, andaccelerometer：（依赖硬件）GPS(Global Positioning System),即全球定位系统。它是由一个由覆盖全球的24颗卫星组成的卫星系统。Compass是一个开放源码的Java搜索引擎框架。

### 1.3.4 SQLite特征

SQLite，

是一款得到了许多产品认可的轻量级嵌入式数据库，同服务端大型的数据库管理系统一样它也是有传统数据ACID关联约束的。与服务端数据库不同的是它所占的内存和资源相比服务端的数据库如Mysql1、oracle等数据库来说根本不值一提，Android系统是移动电子设备的系统，由于受到移动电子设备硬件与内存大小的限制与其他嵌入式产品一样Android同样选择了SQLite数据库进行数据的管理。SQLite所具有的特性如下：

1、轻量性

SQLite与服务端的数据库软件如：Mysql、oracle等数据库软件再体积上有很大的偏差。它不需要数据库的管理端和服务端不同模块的区分。SQLite数据库有一个动态库，这个动态库非常小，同时这个动态库提供了SQLite数据库的大部分功能。

2、独立性

SQLite数据库的本质其实与系统中的文件很相似，它不像服务端数据库那样需要复杂的安装、注册的流程，在APP中使用起来十分便利，可通过代码直接创建数据库。

3、隔离性

SQLite数据库的数据库文件包括数据表、数据、触发器等数据信息都会存放到SQLite数据库的一个文件中，这样它在处理时候速度会很快，也方便程序对数据信息的操作和维护。

4、跨平台

SQLite数据库不仅仅支持Android系统也支持其他的操作系统，这些系统不仅仅是移动设备系统，它同样在电脑上也能管理使用。对系统的平台SQLite也不是十分的依赖，SQLite数据库在Windows、linux或者其他系统中都能良好的支持。

5、多语言接口

SQLite数据库支持很多语言编程接口，比如C\C++、Java、Python、dotNet、Ruby、Perl等，得到更多开发者的喜爱。

6、安全性

SQLite数据库与服务端数据库一样也支持事务的使用，这样一来，使用SQLite数据库一样可以解决并发或者其他的多线程问题。通过加锁的方式它能保障我们对数据的全操作。对数据库读写数据的真实唯一性给了极其方便操作数据库的方法方式。

### 1.3.5本课程中的技术要点简介

本课题的目标是设计实现一款Android app，其主要的功能有以下几点：

1、搭建友好、简洁的用户操作界面。

2、实现在android系统中阅读PDF文档。

3、实现拍照闪存功能，完成拍照保存到本地SD卡的功能。

4、实现阅读文档的同时支持截取屏幕保存到本地SD卡得功能。

5、实现查看本地拍照、截屏保存来的图片，给用户提供相应的管理功能，实现把批量选择本地图片生成对应的PDF文档。

6、实现android 实时保存的记事本，并且加密记事本内容序列化到SD卡。

7、以上功能都是集成在一个app中的看似独立的每一个功能都是不可分割的，因此要根据Android API要求设计整合以上功能。

以上功能主要涉及的技术点有：Android 自定义View、自定义控件、Android ListView GridView适配、Fragment适配、Google Zxing 二维码扫描、MuPdf 解析PDF、以及android的相关API。

# 第2章 系统总体分析和设计

本章从APP的系统UI设计、系统模块划分和主要的模块功能三个方面对该APP进行了总体分析说明。APP开发完成之后，主要通过UI主菜单栏来控制各个模块之间的切换，主要能分为各个文档的管理、图片的查看与相关操作、本地SD卡扫描添加、实用小工具、以及关于我们这几个核心模块。

## 2.1 系统可行性分析

在软件开发行业很多时候系统的需求是几乎没有能一定成功的可行解决方案的，没有好的方案的软件开发势必会走很多弯路，如果最终没有成功，我们所在菜软件开发的时间、资源、人力等等都是在做的无用功。可行性研究的任务就是在较高层次上以较抽象的方式进行的系统分析和设计的过程。下面从几个方面分析一下本应用的可行性：

### 2.1.1技术可行性

本应用基于Android系统的API，以Android 4.0版本以上的各个型号手机为基础，Android 4.0 以上的手机内存、CPU都较以前的版本有很大的提升，足以流畅的运行本app的各个功能。

App UI技术采用Android ViewGroup 来设计实现自定义的可点击按钮、菜单键抽拉的SlideMenu，采用Fragment来动态设计各个功能的界面。

PDF文档阅读功能在java开源项目Mupdf的基础上，结合Android的视图层API Activity来实现阅读功能，同时使用Android的Window以及Bitmap相关的API来实现阅读时截取屏幕生成图片的功能。

图片转换为PDF文件功能。借助Java语言的iText组件实现，iText能够灵活的对PDF进行操作。

用户数据保存方面，使用Android的SQLite数据库、SharedPreferences来实现。

加密实时保存记事本、图片的浏览等，均可利用java API结合Android API实现。

综上所述，从技术角度考虑，本应用开发可行。

### 2.1.2经济可行性

本应用在开发过程中的成本是完全可以预算的，事实上本应用的开发成本与其开发完成后带来的经济实用价值相比开发成本根本不值一提。Android 是完全开源的，在这样的背景下开发本应除了硬件和人力支出外别的再无其他开支。应用开发完成后能给用户再办公、学习上能带来极大的便利，其经济效益是不可预估的。

目前Android 应用市场上很少有本应用这样的应用，所以有较好的市场需求，在经济方面可行。

### 2.1.3操作可行性

本应用主要运行在Android系统手机、或者其他智能终端上运行。运行环境如下：CPU：1000MHZ及以上；内存512M及以上；安装有Android 4.0以上的操作系统，目前市面上的Android 手机基本都能满足本软件的硬件需求。因此，从操作可行性来看，只要系统用户的硬件设备满足以上要求，即使用本软件。用户安装本软件后，可使用本软件的扫描功能，系统自动扫描添加本地的文档，系统界面中的各个功能都有直观、人性化的按钮或者其他提示。综上所述，本系统具有良好的操作可行性。

## 2.2 应用功能需求分析

经过初步的社会市场调研分析，当今办公人群以及学生等可能会使用本应用的人群在现实办公、学习中面临以下问题：

在PDF、word等文档的时候大多依赖PC机，而且PC软件通常也是只针对其中的一两种格式来阅读的，但是一旦脱离了PC机就很难再进行文档的阅读。

在工作和学习中，当纸质文件需要多份传阅时，往往依靠打印机或者复印机进行多份打印后分发，这样对设备依赖较大，且办公耗材损耗极大，复印效果不好的话，还会影响传阅；如果纸质文件需要电子传阅时又缺失电子版文档时，必须在有扫描仪的场所使用扫描仪扫描出电子版文档；如果电子文档有远程传送需求时要使用到电脑才能进行操作。由此看来很明显，目前的工作方法效率低下，日常办公开支较大，致使日常办公繁琐枯燥,效率低下。

当用户在开会或者其他场景中遇到需要记录的场景的时候，大多数只能传统的记录在纸质文件中，或者携带笔记本电脑存储到电脑中，这样也给用户带来极大的不便。

因此开发出一款能解决上述问题的APP就能给用户带来极大的便利，一个APP能解决用户的多个问题，让用户不在依赖PC机、打印机、扫描仪、纸质文档。综上，本应用应满足以下需求：

1)能给用户提供在App中阅读PDF、Word等文档的功能。

2)能给用户提供一种简便的扫描纸质文档生成PDF文档的功能。

3)能给用户提供一种简便安全的实时记录的记事本功能。

4)要求提供较好的用户界面、用户体验，以及一些其他功能。

## 2.3应用功能概要设计

基于Android系统的超级办公应用以Android4.0版本为基础开发，可在Android4.0以上版本各型号手机上运行。本应用主要分为：

Word文档、Excel文档管理：扫描添加、删除、查看。

PDF文档管理：扫描添加、删除、查看。

图片:查看、删除、浏览、批量转化生成PDF文档。

拍照闪存：能够使用摄像头拍取照片、保存在本地内存卡中。

记事本：查看已保存的记事、删除已经保存的记事、添加新的记事、对记事内容加密后序列化到本地内存卡中。

二维码\条形码扫描：扫描读取二维码、条形码中的内容，对有URL的内容能给用户一个便利的查看功能。

在系统功能需求分析的基础上得到系统功能结构图见下图1-1：

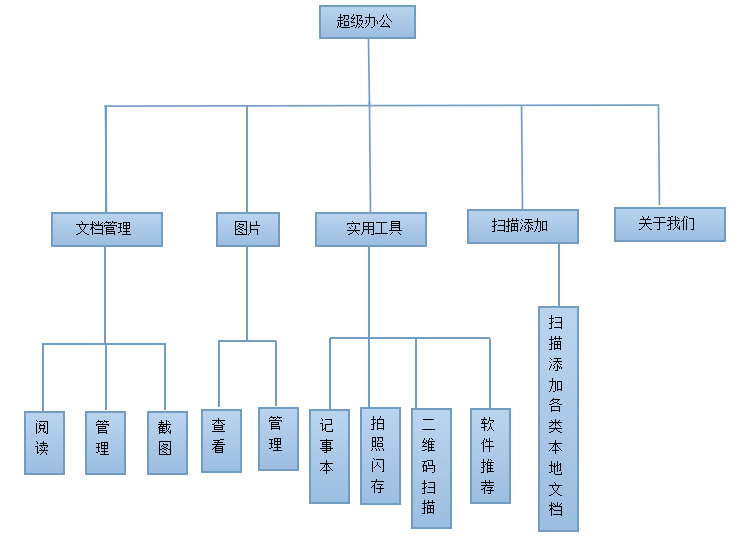


图1-1

## 2.4应用数据库设计

数据库在一个应用系统中占有非常重要的地位，数据库结构设计的好坏将直接对应用系统的效率，以及实现的效果产生影响。合理的数据库结构设计可以提高数据存储的效率，保证数据的完整和一致。

每个应用程序都要使用数据，Android应用程序也不例外，Android使用开源的、与操作系统无关的SQL数据库—SQLite。SQLite第一个Alpha版本诞生于2000年5月，它是一款轻量级数据库，它的设计目标是嵌入式的，占用资源非常的低，只需要几百K的内存就够了。SQLite已经被多种软件和产品使用，Mozilla FireFox就是使用SQLite来存储配置数据的，Android和iPhone都是使用SQLite来存储数据的。

### 2.4.1 数据库需求分析

本应用中最主要的数据就是各个文档的详细数据，无论是系统中的PDF文档还是别的文档都需要再数据库中记录它的相应信息，只有记录了文档的详细信息才能调高操作文档的效率。为了减少数据库表的冗余，根据各个文档的共性，系统设计了一张可适用于不同文档的表。一张表既可存储PDF、word、excel、记事几种类型的文档。总结出如下的表信息：

文档资源表包括的数据项：编号、资源名、资源类型、资源地址、阅读状态、权重、状态。

### 2.4.2数据库概念结构设计

文档资源实体属性图：

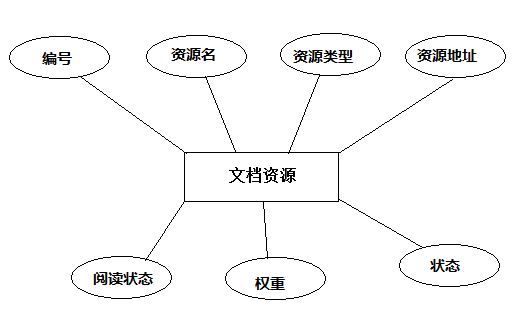


图2-1

### 2.4.3数据库逻辑结构设计

本课题对应的APP，采用的是Android常规的SQLite，以下是APP中所涉及的主要的数据表、表中字段等数据结构信息。表3.1为文档资源表，记录系统中的文档信息。

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 允许空 | 注释 |
| id | int | 否 | 编号 |
| resname | String | 否 | 资源名称 |
| type | String | 否 | 资源类型 |
| path | String | 否 | 地址 |
| hasread | int | 否 | 是否阅读 |
| read\_weight | long | 否 | 权重 |
| status | String | 否 | 状态 |

图3-2

# 第3章 应用详细设计与功能实现

## 3.1 系统模块设计

### 3.1.1 主菜单

打开应用之后用户首先看到的是欢迎页，欢迎页之后的就是阅读记录，在此界面中有打开侧滑菜单的按钮，同时用户可点击手机菜单键打开菜单：

阅读记录页（3-1）、侧滑菜单（3-2）见下图：

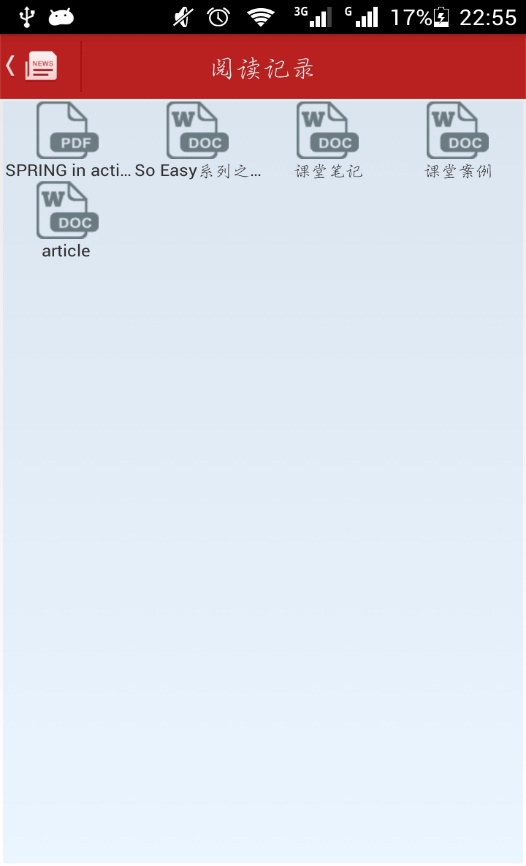
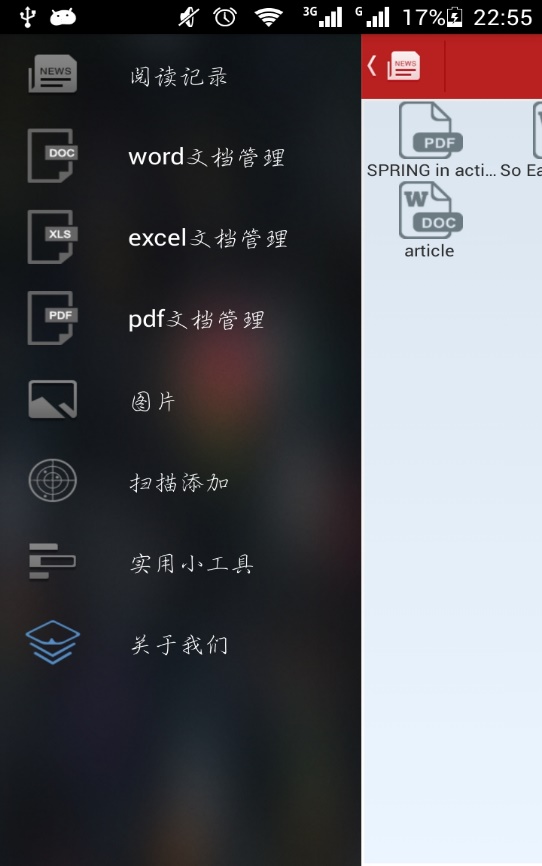
 

图3-1 图3-2

**3.1.2 PDF文档管理**

用户点击侧滑菜单中的“PDF管理”就能进入，PDF管理页面，进入后用户可对页面中展示的文档进行管理：

（1）单击阅读，进入阅读页面（见图3-3），阅读页面有截取屏幕、搜索关键字、查看PDF文档目录的功能，用户可根据自己需求选择相应的功能。当用户截取屏幕之后会生成对应的图片保存在SD卡中，用户可在图片管理中查看、删除图片以及用新截取的图片生成新的PDF文档。

（2）长按弹出删除对话框等待用户操作，取消或者确定删除。

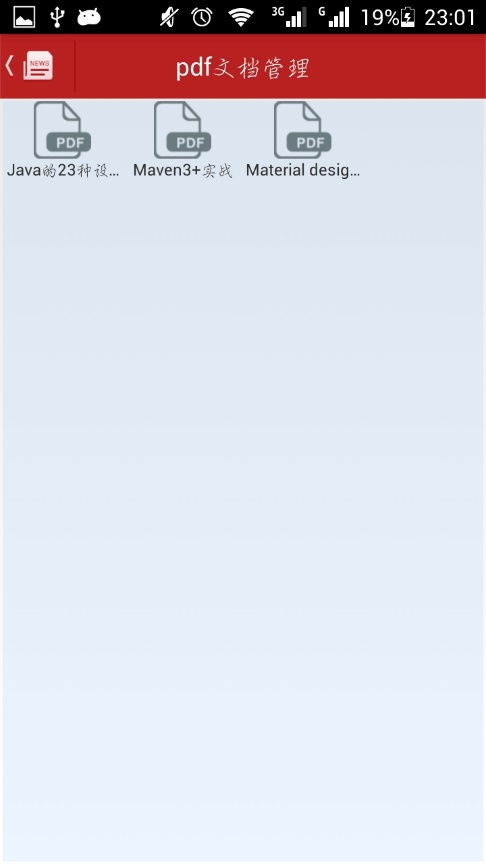
 

图3-3 图3-4

### 3.1.2 图片管理

用户点击侧滑菜单中的“图片”就能进入，图片管理页面，进入后用户可对页面中展示的图片进行管理：

1. 单击可查看当前图片、图片查看界面支持左右滑动查看别的图片。
2. 长按图片项可弹出对话框等待用户操作，取消或者确定删除。
3. 单击图片管理界面下方“选择生成PDF”按钮，可单选或者批量选择图片，单击确定按钮生成PDF文档。

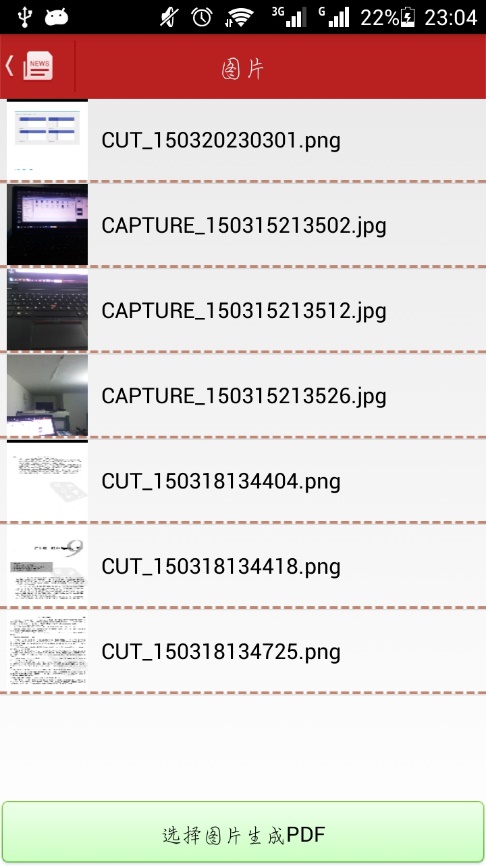
 

图3-5 图3-6

### 3.1.3 实用小工具

用户点击侧滑菜单中的“实用小工具”就能进入，使用小工具页面，进入后用户可对页面中展示的工具进行操作：

1. 单击“拍照闪存”可进行照片的拍取，拍取的照片会存储在SD卡中，用户可在图片菜单中查看、删除所拍照片，也可以选择生成新的PDF文档。
2. 单击“二维码扫描”可打开摄像头扫描二维码或者条形码，当扫描结果仅仅是文本的时候会展示其内容，当扫描结果含有网址的时候会弹出是否打开对应地址的对话框等待用户选择，用户也可以单击扫描结果去跳转到对应的网址。
3. 单击“记事本”可打开记事本页面，用户单击记事条目可查看所点击的记事，长按弹出操作对话框等待用户操作。用户也可以点击页面下方的“新建记事”的按钮去新建记事。
4. 单击“软件推荐”可打开软件推荐页面。

## 3.2 应用模块编码与实现

### 3.2.1 侧滑菜单编码设计

侧滑菜单通过自定的 ViewGroup 来实现。ViewGroup 的原理是把不同的视图组合到一起，通过系统定义的onMeasure （测量子视图大小的方法）和onLayout（设置控件位置的方法）来控制界面显示的效果、各个视图显示的位置。通过检测滑动事件中滑动的距离、方向、大小和当前视图来显现不同的视图。具体编码如下：

public class SlideMenu extends ViewGroup {

private int touchSlop;

private int mMostRecentX;// 最后一次的X轴偏移量

private int MenuScreen = 0;// 菜单页面

private int MainScreen = 1;// 主界面

private int mCurrentScreen = MainScreen;// 当前屏幕的主界面

// 定义数据模拟器 来动态模拟数据的增量

private Scroller mScroller;

public SlideMenu(Context context, AttributeSet attrs) {

super(context, attrs);

mScroller = new Scroller(context);

// startX, startY, dx, dy,duration

// 滑动间距的距离 默认为8

touchSlop = ViewConfiguration.get(context).getScaledTouchSlop();

}

// 测量出所有子布局的宽和高

@Override

protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {

super.onMeasure(widthMeasureSpec, heightMeasureSpec);

// 这个super必须要重写的

measureView(widthMeasureSpec, heightMeasureSpec);

}

/\*\*

\* int widthMeasureSpec, int heightMeasureSpec

\*

\* @param widthMeasureSpec

\* 父布局的测量

\* @param heightMeasureSpec

\*/

private void measureView(int widthMeasureSpec, int heightMeasureSpec) {

// 测量菜单布局

View menuView = getChildAt(0);

menuView.measure(menuView.getLayoutParams().width, heightMeasureSpec);

// 测量主界面的布局

View mainView = getChildAt(1);

mainView.measure(widthMeasureSpec, heightMeasureSpec);

}

// 设置菜单和主界面的布局

@Override

protected void onLayout(boolean chenged, int l, int t, int r, int b) {

// 布置菜单的位置

View menuView = getChildAt(0);

// 得到测量之后的边长

menuView.layout(-menuView.getMeasuredWidth(), 0, 0, b);

// 布置主界面的位置

View mainView = getChildAt(1);

mainView.layout(0, 0, r, b);

}

private void switchScreen() {

int scrollX = getScrollX();

int dx = 0;

if (mCurrentScreen == MainScreen) {// 切换到主界面

// scrollTo(0, 0);

dx = 0 - scrollX;

} else if (mCurrentScreen == MenuScreen) {// 切换到菜单界面

// scrollTo(-getChildAt(0).getWidth(),

// 0);

dx = -getChildAt(0).getWidth() - scrollX;

}

mScroller.startScroll(scrollX, 0, dx, 0, Math.abs(dx) \* 1);

invalidate();// invalidate --> drawChild -->chlid.draw -->computerScroll

}

@Override

public void computeScroll() {

// invaliadate调用此方法，更新X轴的偏移量

if (mScroller.computeScrollOffset()) // 判断是否正在模拟数据中，true 正在进行

{

scrollTo(mScroller.getCurrX(), 0);

invalidate();// 引起computeScroll的调用

}

}

// 时候显示菜单

public boolean isShow() {

return mCurrentScreen == MainScreen;

}

public void hideMenu() {

mCurrentScreen = MainScreen;

switchScreen();

}

public void showMenu() {

mCurrentScreen = MenuScreen;

switchScreen();

}

}

以上ViewGroup定义好之后就可以在布局文件中配置使用了，使用的时候只需要按照上述代码中的配置其中的子视图就能把不同的子视图组合到一起从而达到侧边栏的效果。布局文件如下：

<com.tai.dandelion.view.SlideMenu

android:id="@+id/slidmenu"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent">

<include layout="@layout/slidemenu\_menu"/>

<include layout="@layout/mian"/>

</com.tai.dandelion.view.SlideMenu>

### 3.2.2 文件扫描添加

文件扫描添加的原理是利用java API，迭代遍历SD卡中的目录，匹配遍历所得的File类型，如果是文件夹，继续遍历，当不是文件夹的时候匹配文件类型，当文件类型是当前要扫描的类型时，提取文件信息，存储到数据库中。

由于Android 手机系统属于轻量级系统，系统限制主线程中不能进行长耗时的操作，如此必须再子线程中进行扫描操作。这里使用子线程结合Android Handler来实现，以达到避免主线程阻塞导致应用闪退或者出现异常。

主要的编码实现：

public class ScannerUtil

{

private static List<Document\_resources> list = new ArrayList<Document\_resources>();

/\*\*

\* 扫描外存中的所有的文件夹把type类型的数据添加到数据库

\* @param con

\* @param type

\* @param scannerListener

\*/

public static void ScannerAdd(final Context con,final int type,final ScannerListener scannerListener){

final Handler handler = new Handler(){

@Override

public void handleMessage(Message msg)

{

scannerListener.ScanDone((Boolean) msg.obj,type);

}

};

Runnable runnable = new Runnable()

{

@Override

public void run()

{

boolean sdCardExist = Environment.getExternalStorageState()

.equals(Environment.MEDIA\_MOUNTED); //判断sd卡是否存在

if(sdCardExist)

{

System.out.println("SD卡是存在的");

File sdDir = Environment.getExternalStorageDirectory();//获取根目录

boolean result = ScannerAdd(con,sdDir,type);

Message msg = handler.obtainMessage();

msg.obj = result;

handler.sendMessage(msg);

list.removeAll(list);

} else{

System.out.println("SD卡不存在");

Message msg = handler.obtainMessage();

msg.obj =false;

list.removeAll(list);

} }

};

ThreadPoolManager.getInstance().addTask(runnable);

}

private static boolean ScannerAdd(Context con,File file,int type)

{

Document\_ResourcesDao document\_ResourcesDao = new Document\_ResourcesDao(con);

List<Document\_resources> list = scaner(file,type);

switch (type)//扫描之前先删除同类文档，以免出现多个文档的重复问题

{

case Document\_resources.WORD:

document\_ResourcesDao.DeleteByType("word");

break;

case Document\_resources.EXCEL:

document\_ResourcesDao.DeleteByType("excel");

break;

case Document\_resources.PDF:

document\_ResourcesDao.DeleteByType("pdf");

break;

default:

break;

}

return document\_ResourcesDao.add(list);

}

/\*\*

\* 按照父文件夹以及type返回当前目录下所有的type类型的数据

\* @param file

\* @param type

\* @return

\*/

private static List<Document\_resources> scaner(File file,int type){

File[] files = file.listFiles();

if (null!=files)

{

for (File f: files)

{

if (f.isDirectory())

{

scaner(f,type);

}

else

{

if (f.getName().contains(".")){

String suffix = f.getName().substring(f.getName().lastIndexOf("."));

switch (type)

{

case Document\_resources.WORD:

if(suffix.equalsIgnoreCase(".doc")||suffix.equalsIgnoreCase(".docx"))

{

System.out.println(f.getName());

Document\_resources document = new Document\_resources(6, f.getName(), "word", f.getAbsolutePath(), 0, System.currentTimeMillis(), Document\_resources.ADD);

list.add(document);

}

break;

case Document\_resources.EXCEL:

if(suffix.equalsIgnoreCase(".xls")||suffix.equalsIgnoreCase(".xlsx"))

{

System.out.println(f.getName());

Document\_resources document = new Document\_resources(1, f.getName(), "excel", f.getAbsolutePath(), 0, System.currentTimeMillis(), Document\_resources.ADD);

list.add(document);

}

break;

case Document\_resources.PDF:

if(suffix.equalsIgnoreCase(".pdf"))

{

System.out.println(f.getName());

Document\_resources document = new Document\_resources(1, f.getName(), "pdf", f.getAbsolutePath(), 0, System.currentTimeMillis(), Document\_resources.ADD);

list.add(document);

}

break;

default:

break;

}

}

}

}

}

return list;

}

}

### 3.2.3 数据库设计

应用中需要保存用户使用的数据信息，各种文档的信息，由于数据可能很多，直接使用I/O流保存到文件中或者序列化到XML中这样无论是保存或者是读取的时候都将会大量占用系统资源，由此这里使用了数据库来存储这些信息。本课题中的APP使用的Android SQLite数据库，以下项目中使用的数据库操作的相关主要编码：

数据库建表：

@Override

public void onCreate(SQLiteDatabase db)

{

String initSql =

"create table document\_resources (" +

"id integer primary key autoincrement," +

"resname varchar(500), " +

"type varchar(20)," +

"path varchar(500)," +

"hasread integer," +

"read\_weight long," +

"status varchar(40))";

db.execSQL(initSql);

}

应用中对文档信息的操作主要有：单条数据添加、批量添加、单条数据删除、分类删除、单条数据查询、分类查询、查询所有、删除所有等操作，针对应用中可能会有的数据操作设计了以下的数据访问接口：

public interface ResourceDao {

/\*\*

\* 单个添文档信息

\* @param dres

\* @return

\*/

public boolean addSingle(Document\_resources dres);

/\*\*

\* 添加

\* @param res

\* @return

\*/

public boolean add(List<Document\_resources> res);

/\*\*

\* 单个删除

\* @param id

\* @return

\*/

public boolean singleDelete(int ... id);

/\*\*

\* 根据类型删除

\* @param type

\* @return

\*/

public boolean DeleteByType(String type);

/\*\*

\* 查询所有

\* @return

\*/

public List<Document\_resources> findAll();

/\*\*

\* 很据ID单个查询

\* @param id

\* @return

\*/

public Document\_resources findById(int id);

/\*\*

\* 根据类型查询

\* @param type

\* @return

\*/

public List<Document\_resources> findByType(String type);

/\*\*

\* 根据操作查询

\* @param read

\* @return

\*/

public List<Document\_resources> findByOperation(int read);

/\*\*

\* 更新数据

\* @param document\_resources

\* @return

\*/

public boolean update(Document\_resources document\_resources);

}

根据数据访问接口应用中的方法定义了数据访问接口帮助类，类中具体实现了以上操作数据的方法，主要就是利用Android API中定义的规范来实现的，以下是以上接口中定义的方法中其中一个的实现，其他具体实现十分相似这里就不再一一列举：

/\*\*

\* 根据类型查询

\* @param type

\* @return

\*/

public List<Document\_resources> findByType(String type)

{

List<Document\_resources> list = new ArrayList<Document\_resources>();

SQLiteDatabase Database = DbHelper.getWritableDatabase();

Database.beginTransaction();

Cursor cursor = Database.query(TABLE\_NAME, null, "type=?", new String[]{type}, null, null, null);

while (cursor.moveToNext())

{

int resid = cursor.getInt(cursor.getColumnIndex("id"));

String resname = cursor.getString(cursor.getColumnIndex("resname"));

String respath = cursor.getString(cursor.getColumnIndex("path"));

String restype = cursor.getString(cursor.getColumnIndex("type"));

String resstatus = cursor.getString(cursor.getColumnIndex("status"));

int hasread = cursor.getInt(cursor.getColumnIndex("hasread"));

long read\_weight = cursor.getLong(cursor.getColumnIndex("read\_weight"));

Document\_resources document\_resources = new Document\_resources(resid,resname,restype,respath,hasread,read\_weight,resstatus);

list.add(document\_resources);

}

cursor.close();

Database.endTransaction();

Database.close();

return list;

}

### 3.2.4 应用中文档管理界面

应用中文档管理界面中主要使用的GridView来实现，应用中的管理界面都十分相似，采用了共用机制，让不同的界面共用同一布局、同一实现，只在数据源上做了区分，以下是管理界面中的编码实现：

布局文件：

<GridView

android:id="@+id/gridview"

android:layout\_width="fill\_parent"

android:layout\_height="match\_parent"

android:layout\_margin="2dp"

android:layout\_marginTop="3dp"

android:background="@color/press"

android:numColumns="4" >

</GridView >

GridView的使用主要是设计他的数据源设配器 Adapter，Android为我们提供了相关的接口规范，以下是应用中的实现：

public class FileListAdapter extends BaseAdapter {

private List<Document\_resources> doclist;

private Context con;

public FileListAdapter( List<Document\_resources> list, Context context) {

this.doclist = list;

this.con = context;

}

public FileListAdapter() {

}

@Override

public int getCount() {

return doclist.size();

}

@Override

public Object getItem(int position) {

return doclist.get(position);

}

@Override

public long getItemId(int position) {

return position;

}

@Override

public View getView(int position, View convertView, ViewGroup parent) {

View view;

ViewHolder holder;

Document\_resources resources = doclist.get(position);

if (null!=convertView) {

view = convertView;

holder = (ViewHolder) view.getTag();

}else {

view = View.inflate(con, R.layout.file\_item, null);

holder = new ViewHolder();

holder.iv\_image = (ImageView) view.findViewById(R.id.im\_v);

holder.tv\_text = (TextView) view.findViewById(R.id.tv);

view.setTag(holder);

}

if ("word".equalsIgnoreCase(resources.getType())) {

holder.iv\_image.setImageResource(R.drawable.item\_doc);

}else if ("excel".equalsIgnoreCase(resources.getType())) {

holder.iv\_image.setImageResource(R.drawable.item\_xls);

}else if ("pdf".equalsIgnoreCase(resources.getType())) {

holder.iv\_image.setImageResource(R.drawable.item\_pdf);

}

String name = resources.getResname();

if (name.contains(".")) {

name = name.substring(0, name.lastIndexOf("."));

}

holder.tv\_text.setText(name);

return view;

}

static class ViewHolder {

ImageView iv\_image;

TextView tv\_text;

}

}

界面中的数据均来自于数据库，界面中视图的点击事件以及长按事件设置给了GridView，Android中的事件处理机制使用了监听者模式，Android API中做了一定的规范，以下是应用中的文档列表项的监听器：

public class DocumentClickListener implements OnItemClickListener,

OnItemLongClickListener {

public DocumentClickListener() {

}

public DocumentClickListener(Context con, FileListAdapter adapter,List<Document\_resources> list) {

this.context = con;

this.adapter = adapter;

this.doclist = list;

dao = new Document\_ResourcesDao(context);

}

private Context context;

private Document\_ResourcesDao dao;

private ProgressDialog pd;

private FileListAdapter adapter;

private List<Document\_resources> doclist;

// 定义Handler对象

private Handler handler = new Handler() {

@Override

// 当有消息发送出来的时候就执行Handler的这个方法

public void handleMessage(Message msg) {

super.handleMessage(msg);

Document\_resources res = (Document\_resources) msg.obj;

pd.dismiss();

String htmlDir = FileUtil.HtmlDirectory

+ File.separator

+ res.getResname().substring(0,

res.getResname().lastIndexOf(".")) + ".html";

System.out.println(htmlDir);

Intent i = new Intent();

i.setClass(context, Web\_Activity.class);

i.putExtra("htmlDir", htmlDir);

context.startActivity(i);

}

};

@Override

public void onItemClick(AdapterView<?> parent, View view, int position,

long id) {

Document\_resources dresource = (Document\_resources) adapter

.getItem(position);

if (dresource.getType().equals("word")

|| dresource.getType().equals("excel")) {

processThread(dresource);

} else if (dresource.getType().equals("pdf")) {

// Intent i = new Intent();

// i.setClass(context, PdfReaderActivity.class);

// i.putExtra("dpfDir", dresource.getPath());

Uri uri = Uri.parse(dresource.getPath());

Intent intent = new Intent(context,MuPDFActivity.class);

intent.setAction(Intent.ACTION\_VIEW);

intent.setData(uri);

dresource.setRead\_weight(System.currentTimeMillis());

dresource.setHasread(1);

dresource.setStatus("read");

dao.update(dresource);

context.startActivity(intent);

}

}

private void processThread(final Document\_resources document) {

// 构建一个下载进度条

pd = ProgressDialog.show(context, "正在读取内容...", "请稍后...");

new Thread() {

public void run() {

if (document.getHasread() == 0) {// 没有读过

try {

try {

if (document.getType().equals("word")) {

Word2Html.convert2Html(document.getPath(),

FileUtil.HtmlDirectory,

FileUtil.ImageDirectory,

document.getResname());

} else if (document.getType().equals("excel")) {

// Excel2Html toHtml = Excel2Html.create(new

// FileInputStream(document.getPath()),new

// PrintWriter(new

// FileWriter(FileUtil.HtmlDirectory+File.separator+document.getResname().substring(0,document.getResname().lastIndexOf("."))+".html")));

// toHtml.setCompleteHTML(true);

// toHtml.printPage();

JxlExcel2Html.excel2html(document.getPath(),FileUtil.HtmlDirectory+ File.separator+ document.getResname().substring(0,document.getResname().lastIndexOf("."))+ ".html");

}

} catch (TransformerException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} catch (ParserConfigurationException e) {

e.printStackTrace();

}

document.setRead\_weight(System.currentTimeMillis());

document.setHasread(1);

document.setStatus("read");

dao.update(document);

} catch (Exception e) {

e.printStackTrace();

}

} else {

document.setRead\_weight(System.currentTimeMillis());

dao.update(document);

}

// 执行完毕后给handler发送一个消息

Message msg = handler.obtainMessage();

msg.obj = document;

handler.sendMessage(msg);

}

}.start();

}

@Override

public boolean onItemLongClick(AdapterView<?> parent, View view,

int position, long id) {

final Document\_resources resource = (Document\_resources) adapter

.getItem(position);

String[] items = null;

if (resource.getType().equals("word")){

items = new String[]{"转化为PDF","删除该文档"};

}else if (resource.getType().equals("pdf")){

items = new String[]{"删除该文档"};

}else if(resource.getType().equals("excel")){

items = new String[]{"转化为PDF","删除该文档"};

};

AlertDialog.Builder builder = new Builder(context);

builder.setTitle("请选择操作");

builder.setSingleChoiceItems(items, 0, new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

switch (which) {

case 0:

if (resource.getType().equals("pdf")) {

deleteDoc(resource);

}else{

//这里要生成PDF

}

dialog.dismiss();

break;

case 1:

deleteDoc(resource);

dialog.dismiss();

break;

default:

dialog.dismiss();

break;

}

}

}).setNegativeButton("取消", null).show();

return false;

}

private void deleteDoc(Document\_resources resource){

String path = resource.getPath();

Document\_ResourcesDao dao = new Document\_ResourcesDao(context);

boolean flag = dao.singleDelete(resource.getId());

File file = new File(path);

boolean result = file.delete();

if (result) {

doclist.remove(resource);

adapter.notifyDataSetChanged();

Toast.makeText(context, "删除成功！", Toast.LENGTH\_SHORT).show();

}

}

}

# 结 论

本文论述了一个基于Android操作系统手机的办公帮助软件，基本上实现了应用设计的各个功能，对办公学习阅读、实时记事存储、实时拍照转化成文档等功能，能极大的帮助用户、提高用户的办公学习效率。课题所设计的办公助手App，主要是熟悉和掌握Android的开发技术以及对移动设备平台的初步探讨和设计。在几个月的设计过程中，初步了解了Android的优点和先进之处，也看到了Android技术在移动通信设备上的光明前途。

设计和开发办公助手软件的整个过程对我来说本身就是一个很好的学习过程，随着项目的深入进展，各种问题也会浮出水面，在软件开发的过程中解决问题的能力才是最重要的能力。本课题中的软件助手软件所涉及的模块涵盖了很多Android系统的API，完成项目之后我对Android开发有了自己更深入明了的认识。整个项目下来也锻炼了我的编程和设计能力。

当然，其中也遗留下了一些待解决的问题，但出于自己水平有限，作为一个办公、学习软件，该项目尚有一些不完善和待改进之处。

通过这次的毕业设计，了解了目前流行的主流移动通信设备软件的运作原理，掌握了用Android相关知识和技术原理，锻炼了自己的动手实践能力。更重要的是，通过这次的毕业设计，培养出了刻苦钻研的学习精神和严肃认真的学习态度，对以后的学习和工作有很大的益处。

由于项目创作时间有限，自己也是新自学Android，知识水平有限，App中可能有许多不足之处，有待日后完善。

# 参考文献

[1] 李兴华，android开发实战经典（第1版），清华大学出版社，2012年3月.

[2] 杨丰盛，Android 应用开发揭秘，机械工业出版社，2012年4月.

[3] 杨丰盛，Android技术内幕:系统卷，机械工业出版社 2011年6月．

[4] 梅尔(Reto Meier)、 王超 ，Android 2高级编程(第2版) 清华大学出版社

[5] 刘平. Android手机访问服务器的一种数据交互方法[J] . 西安财经学院 网络 与现在教育技术中心，2010年9月：96-102.

[6] 刘卫国，姚昱禹. Android与J2ME平台间即时通信的研究与实现[J]. 中南大学信息科学与工程学院. 2008年12期：118-120.