# 摘要

随着智能手机发展越来越迅猛，手持智能手机的人越来越多，尤其是价格合理的android手机。人们已经习惯用手机来处理一些事情，尤其用手机的记事本或者标签来记录日常工作的事物，在这种情况下，开发一款基于android平台的简单实用的记事本很有必要，可以在很大程度上方便人们的生活。

此次课题研究的目标是在android系统上，实现一款能够流畅运行记事本应用，并支持多种记事方式。目前来说，利用android 强大的ADK工具包以及eclipse完善的开发功能，可以比较容易的实现这样一款应用。本论文主要论述了这款应用的需求场景以及系统设计和实现，其中这款应用具有两大特色：第一个特色是可以添加闹钟提醒，并且可以设置闹钟提醒次数，提醒多次；第二个特色是同时支持音频记事和图片记事，并且图片记事灵活，可以添加多张图片。

论文中最终实现了一款基于android系统的记事本应用YYnote，此应用的主要功能有添加文本，添加多张图片，录制音频以及为记事添加闹钟提醒，同时还可以分享记事，整个应用在android4.0上运行流畅。

**关键词：**安卓系统，记事本，闹钟提醒，图片记事，音频记事

# Abstract

目录

[摘要 I](#_Toc356912548)

[Abstract II](#_Toc356912549)

[1 绪论 1](#_Toc356912550)

[1.1 课题来源、目的和意义 1](#_Toc356912551)

[1.2 国内外基本研究情况 2](#_Toc356912552)

[2 需求分析 4](#_Toc356912553)

[2.1 用户需求 4](#_Toc356912554)

[2.1.1 用户需求分析 4](#_Toc356912555)

[2.1.2 预期目标 4](#_Toc356912556)

[2.2 功能需求 4](#_Toc356912557)

[2.2.1 功能概述 4](#_Toc356912558)

[2.2.2 功能模块图 5](#_Toc356912559)

[2.2.3 功能描述 5](#_Toc356912560)

[2.2.4 功能亮点和难点 6](#_Toc356912561)

[2.3 数据需求 7](#_Toc356912562)

[2.3.1 数据库选择 7](#_Toc356912563)

[2.3.2 数据信息 8](#_Toc356912564)

[2.4 性能需求 8](#_Toc356912565)

[2.5 运行需求 8](#_Toc356912566)

[2.6 可行性分析 8](#_Toc356912567)

[3 概要设计 10](#_Toc356912568)

[3.1 功能概要设计 10](#_Toc356912569)

[3.1.1 记事模块 10](#_Toc356912574)

[3.1.2 闹钟模块 15](#_Toc356912575)

[3.1.3 分享模块 18](#_Toc356912576)

[3.1.4 备份模块 19](#_Toc356912577)

[3.2 数据库概要设计 19](#_Toc356912578)

[3.2.1 E-R图 19](#_Toc356912579)

[3.2.2 数据字典 20](#_Toc356912580)

[4 详细设计 22](#_Toc356912581)

[4.1 功能设计 22](#_Toc356912582)

[4.1.1 记事模块 24](#_Toc356912583)

[4.1.2 闹钟模块 31](#_Toc356912584)

[4.1.3 分享模块 35](#_Toc356912585)

[4.1.4 备份模块 37](#_Toc356912586)

[4.2 数据库设计 38](#_Toc356912587)

[4.2.1 数据库分析 38](#_Toc356912588)

[4.2.2 逻辑结构设计 38](#_Toc356912589)

[5 系统实现 41](#_Toc356912590)

[5.1 功能实现 41](#_Toc356912591)

[5.1.1 总述 41](#_Toc356912592)

[5.1.2 记事模块 42](#_Toc356912593)

[5.1.3 闹钟模块 46](#_Toc356912598)

[5.1.4 分享模块 48](#_Toc356912600)

[5.2 数据库实现 50](#_Toc356912601)

[5.2.1 数据表创建 50](#_Toc356912602)

[5.2.2 数据库操作 50](#_Toc356912603)

[6 系统测试及演示 52](#_Toc356912604)

[6.1 概述 52](#_Toc356912605)

[6.2 添加文本 52](#_Toc356912606)

[6.3 添加图片 53](#_Toc356912607)

[6.4 添加音频 54](#_Toc356912608)

[6.5 添加闹钟 55](#_Toc356912609)

[6.6 分享 57](#_Toc356912610)

[7 总结 58](#_Toc356912611)

[致谢 59](#_Toc356912612)

[参考文献 60](#_Toc356912613)

# 绪论

## 课题来源、目的和意义

 Android是一种以Linux为基础的开放源码操作系统，主要使用于便携设备。Android操作系统最初由Andy Rubin开发，最初主要支持手机。2005年由Google收购注资，并拉拢多家制造商组成开放手机联盟开发改良，逐渐扩展到到平板电脑及其他领域上。 2010年末数据显示，仅正式推出两年的操作系统的Android已经超越称霸十年的诺基亚Symbian系统，跃居全球最受欢迎的智能手机平台。

目前移动设备发展的越来越迅速，每天的设备激活数目就达到了50万台，手机平板的使用越来越普遍，而Android软件的开发也变得越来越行。互联网行业的巨头基本都有关于Android的开发项目，同时像Motorola移动、HTC、华为更是开发基于Android操作系统的手机设备。可以看出Android的市场份额越来越大，发展前景很广阔。

在现代社会生活中，人们的工作越来越忙，生活琐事也越来越多，每天要做的事情总是很多，人们需要用一种方式来安排自己每天的生活，在移动设备发展之前，一些人选择用一个小本子或者便签贴来记录事情，或者安排自己的日常活动。但是，用小本子或者便签贴有很多不方便之处，例如，我们可能随时随地都需要记一些东西，这就需要我们随身携带一个本子，很不方便；再者，它们是死的，没有提醒功能，虽然我们可以把要做的事情写下来，但是到了一定的时间点，没有提醒，我们仍然可能会忘记。

现在，随着智能手机发展越来越迅猛，手持智能手机的人越来越多，尤其是android手机，在中国的智能手机市场占了90%的份额。人们可以利用智能手机做很多事情，发微薄，聊微信等等。在这种情况下，开发一款智能手机上的记事本，不仅实用，而且还克服了用传统方式记事的缺点，可以随时随地的记录任何信息，在生活工作中，一些重要的事情设置提醒，时间到了，手机自然会提醒我们该做什么，不用担心会忘记事情，这将大大方便人们的生活。此外，根据现在人们的生活习惯爱好，还将增加更多更实用的功能，例如分享功能，拍照功能等。

此次课题就是希望开发一款基于Android手机设备的记事本应用，方便人们的生活。另外，作为即将投身互联网行业的我们，学习最新的互联网技术是必须的。Android记事本是Android系统的基本功能，对于我们而言，开发Android记事本，可作为我们学习Android开发的入门实战，帮助我们更好的了解Android开发，掌握Android开发的基本技能，对我们将来从事相关方面的研究有很大的帮助。

## 国内外基本研究情况

1. **Android系统的发展**

市场研究机构IDC发布2012年第三季度智能手机系统数据，谷歌Android系统市场占有率依然高居榜首，Q3全球智能手机出货量达1.811亿台，Android占比为75%，苹果iOS的市场份额仅为Android的1/5，Q3为14.9%。

Android系统的发展，已经由最初的1.1更新到4.1，而在最新的4.1版本中，更多更新的功能被加进来。

其中一个就是“眨眼开锁”，即需要对着手机眨眼以证明其为“真人”，避免被照片等静止图像解锁。这项新功能无疑比以前的解锁方式更加的先进和方便，当它检测到人脸的存在后，会在屏幕上显示“Blink now”，这时对着屏幕眨眼，手机就自动解锁了。

另外一个功能是“无障碍操作”，在无障碍操作方面，果冻豆的API将会更加完整，让视力不佳用户能够更好的操作自己的手机；开发者也可根据新的API来实现更完善的辅助功能，延展Android的功能。Android 4.1还加入了强悍的Gesture Mode功能(手势模式)，这个功能让使用者可以进行对系统的导航，同时结合触摸与滑动等手势进行操作，且支持语音输出。Android 4.1还专门为残疾人设计了插件，通过这个插件消费者可以利用USB或者蓝牙外接盲人输入输出设备，令操作变得更加简单容易。

1. **国内外记事本的研究**

对于记事本的研究，目前国内外有4款比较优秀的记事本应用。分别是Evernote，小米便签，NoteEverything，麦库记事。

Evernote 应用界面清爽，交互人性化，功能强大丰富。具有登陆注册功能，在登录后可以使用；既可以作为日常生活中的记事也可以作为学习笔记使用；可以建立多个笔记本，分别记录不同方面的笔记（比如学习笔记，生活笔记，日记）；笔记方面除了编写内容外，还可以支持上传图片、音频、视频、文件等附件以及对文字内容的简单编辑圈点；支持添加标签分类，方便管理；最具特色功能是搜索笔记和发送笔记、地理位置定位和同步功能（在多个设备之间同步笔记，如PC、安卓移动终端、平板、OS设备）；缺点是内存占用比较多，而且自带的编辑功能不好用，不仅功能单一，而且对于从其他地方复制来的内容，格式编辑起来更困难。

小米便签是MIUI系统自带的应用，相当于简单记事本。界面美观大方，功能简单，可查看和新建便签；设置闹钟提醒（闹钟功能比较单一）；除了支持文字内容的便签外，还可以添加图片；交互功能比较丰富，可以选择便签的颜色、字体大小，支持拖动等；另外也支持同步功能；资源占用比较少，但是功能比较单一，没有分类标签、音频等功能，适合做一些简单的记事。

Note Everything称为万能笔记本，界面简单；新建记事功能丰富，支持文本记事、绘图记事、语音记事、条形码记事、google文档记事；可以设置多个文件夹；可以发送记事（发送到日历，发送到google文档等）；支持搜索功能；另外可以更改外观主题，字体大小颜色等；但是它界面不够美观，用户体验不是很好，没有闹钟功能。

麦库记事界面简单，功能比较多，类似于evernote，但有很多不同。支持分类、添加标签、搜索功能；支持上传附件；支持拍照和实时录音笔记；具有同步功能，需要注册。最有特色的功能是可以将语音输入转化成文字。缺点同样是界面不够精美，用户体验不好，而且CPU占用多。

# 需求分析

## 用户需求

### 用户需求分析

现代人面对的是繁重的工作和生活的琐事，每天总有做不完的事情，所以用户需要一个简单实用的记事本来做助手。

对用户的需求分析如以下几点：

1. 可以把每天的工作列一个清单，安排自己每天的工作
2. 可以把重要的事情记录下来保存
3. 可以随心随地的记录一些所见所闻，而不用额外的带一些纸笔
4. 对怕忘记的事情，用户希望可以利用记事本来提醒自己，比如一个会议
5. 用户添加闹钟选择日期时，可能也想知道这一天的农历是多少
6. 用户不想打字了，可以拍照或者录音频来做记录
7. 用户觉得重要的事情或者有趣的事情，可以把它分享给大家

### 预期目标

通过对用户需求的分析，此次记事本产品希望能够最大程度的满足用户的需求，不仅在功能上丰富实用，如简单记事，添加图片语音，闹钟提醒等，而且在交互上，能够符合用户的行为习惯。

总体来说，此次产品希望能够帮助用户管理日常的生活和工作学习。

## 功能需求

### 功能概述

此次设计的记事本主要包括的功能及要求如表格2.1

表格2. 1记事本的功能需求表

|  |  |  |
| --- | --- | --- |
| **功能分类** | **主要功能点** | **功能要求** |
| 界面设计 | —— | 要求界面美观大方 |
| 交互操作 | —— | 要求交互人性化 |
| 记事 | 基本操作 | 支持记事的新建、删除、修改 |
| 文本记事 | 支持用文本来记事 |
| 图片记事 | 支持添加图片来记事，包括图库选择和拍照 |
| 语音记事 | 支持录入音频来记事 |
| 闹钟 | 基本操作 | 支持闹钟的新建、删除、修改 |
| 提醒方式 | 允许设置闹钟提醒的次数 |
| 提醒间隔 | 允许设置闹钟提醒的时间间隔 |
| 日历 | 支持农历和阳历 |
| 分享 | 分享出去 | 支持把记事的内容分享出去 |
| 接受分享 | 可以接受从其他应用分享来的内容 |
| 云端备份 | 备份恢复数据 | 可以把数据备份到云端，防止丢失 |

### 功能模块图

功能模块图如图2.1

图2. 1功能模块图

### 功能描述

1. 记事
2. 基本操作

包括新建记事、删除记事、修改记事。

1. 文本记事

主要通过文字来添加记事。

1. 图片记事

新建记事时，可以添加图片来记事。添加图片主要通过两种方式：可以直接调用图库，从图库中选择图片；也可以调用系统的照相机，拍照来添加图片。同时，添加图片时，可以对图片进行基本的编辑，如放大、缩小、旋转。

1. 语音记事

新建记事本时，能够通过录入一段语音来添加记事。

1. 闹钟
2. 基本操作

基本操作包括闹钟的添加、删除和修改

1. 提醒次数

在设置闹钟时，可以选择提醒的次数，比如提醒一次、提醒两次、提醒三次等；如果没有选择提醒次数，则默认只提醒一次。

1. 提醒间隔

在设置闹钟时，如果选择的提醒次数多于一次，则需要选择提醒的时间间隔，如间隔1分钟、2分钟、3分钟等；如果没有选择提醒间隔，则默认间隔1分钟。

1. 日历

在添加闹钟需要选择日期，日期的选择是以日历的方式呈现的，通常的日历只提供了阳历，但是按照中国用户的习惯来说，农历也是很重要的，所以本文设计的日历，不仅可以支持阳历，还可以支持农历。通过一定的转化算法将相应的阳历转换成农历。

1. 闹钟过期处理

在闹钟过期后，可以选择修改闹钟或者删除闹钟。

1. 分享
2. 分享出去

用户可以把记事的内容分享出去，可以分享到网络应用（如花瓣，evernote），还可以短信分享给朋友，可以微信分享，可以分享到微博，也可以用邮件的方式分享等等。

1. 接收分享

此应用不仅可以把记事的内容分享出去，也可以从其他应用（例如短信）接受分享进来的内容，作为一个新的记事。

1. 云端备份

能够把数据备份到云端，并从云端恢复，防止数据丢失。

### 功能亮点和难点

此次记事本主要有三个亮点和难点：

1. 记事本支持日历，而且日历同时支持阳历和农历

现有的记事本或者便签，都没有日历这个功能，在添加闹钟选择日期，日期并不是以日历的方式呈现的，更没有农历，而对于中国用户，对农历的需求是很大的，中国的一些节日甚至一些人的生日都是用农历来算的，比如同学三月初三的生日，用户担心自己会忘记，于是新建了一个记事，添加闹钟，选择日期时，如果只有阳历，用户就不知道应该选择哪一天是对的，于是他要先查一遍三月初三的阳历是多少，才能选择，如果记事本本身支持农历，那么用户就可以直接选择日期了，对用户来说很方便。

要想实现农历的支持，就需要用一定的算法来把阳历转换成农历，目前还没有成熟的函数可以实现，android也没有控件可以使用，只能自己来实现，其中还有很多细节都需要考虑，所有做起来比较复杂，这也是日历的难点所在。

1. 支持语音记事和图片记事

大部分的记事本便签只支持文本记事，支持图片记事和语音的比较少。用户有时候可能不愿意输入太多的文字来记录一件事，更希望直接通过录入一段语音来记事，所以语音的功能对用户来说是很方便；而且现在android上关于语音的应用的越来越多，语音技术越来越流行，也越来越受用户的喜爱，例如微信的语音发送，讯飞语点，google语音搜索等，所以开发支持语音记事的记事本，不仅可以增加用户的青睐度，也是记事本的一个特色所在。

另外，用户有时候会很需要用图片来记录事情，比如出行需要记录下地铁路线，就可以拍照来添加记事，同时添加一些题注，方便查看；

由于系统的资源有限，所以拍照出来的图片或者音频都需要进行压缩处理，另外还有考虑到它们的分享，所以图片和音频功能也有一定的难度。

1. 云端备份

由于记事的数据都是存放在本地的，在重装系统或者系统故障时，数据难免会丢失，所以可以把数据备份到云端，在数据丢失后，可以及时恢复数据。

难点是涉及网络操作部分，尤其是push到云端比下载更困难，加上数据信息比较多，不容易组织操作。

## 数据需求

### 数据库选择

此次的记事本应用，是基于android手机的，而且数据要存储在本地，考虑到手机资源和内存的限制，要求记事本应用占用的资源和内存要尽量少，所以在选择数据库时，要选择资源占用少，操作简单，响应速度快的数据库，综合各方面的因素，此次开发选择的数据库是SQLite。

SQLite的优点主要有以下几点：

1. 它是一款轻型的数据库，占用的资源非常低
2. 能够和多种语言结合，包括java，php，c等
3. SQLite独立，没有额外的依赖
4. 它是连接到程序中成为它的一部分，不需要独立的进程，在消耗总量、延迟时间上有积极的作用
5. 简单轻松的API，可以减少开发的难度

### 数据信息

此次的记事本应用应该包括两个表，note表和clock表。Note表主要用来存储跟记事有关的信息，clock表主要用来存储跟闹钟有关的信息。

Note表应该包括的数据如表格2.2

表格 2.2 Note表数据

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **数据类型** | Clock\_id | Create\_date | Modify\_date | Content | Picture | Audio |
| **数据描述** | 闹钟的ID | 创建时间 | 修改时间 | 文本内容 | 图片 | 音频 |

Clock表应该包括的数据如表格2.3

表格 2.3 Clock表数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **数据类型** | Id | Time | Alert\_interval | Alert\_times |
| **数据描述** | 闹钟的id | 提醒时间 | 提醒间隔 | 提醒次数 |

## 性能需求

1. CPU资源占用尽量少，各种交互操作响应速度要快，不能出现卡死或者不流畅的问题。
2. 能够兼容低版本的android系统。
3. 内存占用尽量少，不能出现内存泄露的问题。

## 运行需求

要求能够在android 2.3及以上版本运行

## 可行性分析

开发本次产品的可行性将从以下几点来分析：

1. 开发工具

现在，进行android开发已经有相当成熟的开发工具—eclipse，兼容win7 32位机；开发框架也有成熟的android SDK。

1. 数据库和服务器

本次产品的数据量总体来说不大，所以用户的数据是存放在本地，在本地建立数据库，而备份到云端，可以暂时用阿里云提供的服务，因此不需要考虑购买服务器和数据库的问题。

1. 时间

本次产品的开发周期比较长，有大概三个月的时间，所有不存在时间紧张的问题。

1. 测试机

本次产品的开发者已经有小米手机和华为手机来作为测试机。

1. 技术能力

本次产品的开发用的语言是JAVA语言，在课堂接触过，同时有官方的android API作为参考指导，所以技术能力不是问题。

# 概要设计

## 功能概要设计

此次记事本应用的功能主要分为四个模块：记事模块，闹钟模块，分享模块，云端备份模块，主要功能点包括文本记事，图片记事，语音记事，闹钟提醒，分享信息，数据备份。

此次记事本开发希望能够开发一款符合用户生活工作习惯的应用，能够满足用户的各方面需求，要求功能完善丰富，具有良好的用户界面和交互体验。

功能总体设计如图3.1

图3. 1功能总体设计



### 记事模块

#### 总述

记事模块对记事本来说，是最基本的模块，主要包括“文本记事”“图片记事”“语音记事”，基本的操作包括“新建”“删除”“修改”“保存”。

1. 记事本首页的原型图如图3.2



图3. 2记事本首页原型图

如图3.2所示，所有的记事都以列表的形式展示在首页，不同的记事方式有不同的展现方式：

文本记事，显示的是部分文本内容。

图片记事，除了显示部分文本内容外，还会用[pic]来代替图片的位置。

语音记事来说，显示部分文本内容，用语音图片来代替语音。

对于每一篇记事，最后都显示了最新的修改时间，每次编辑后，更新内容的同时，也会更新修改时间；并且当添加闹钟后，时间前面会显示一个闹钟图标，表示该记事已添加闹钟。

1. 删除

删除的原型图如图3.3



图3. 3删除原型图

如图3.3所示，长按一个记事列表，就会出现删除按钮，隐藏列表后面的时间，同时首页下面的按钮变为“Cancel”，如果想放弃删除，则可以选择“Cancel”按钮，也可以直接按手机的返回键放弃删除。

当选择了删除后，会更新数据库，删除相应的数据，并且刷新记事列表。

1. 编辑

选择一个记事列表后，就可以进入该记事的编辑页面，可以编辑内容保存，也可以不编辑直接返回。

1. 保存

在编辑页面，当用户编辑完后，可以选择导航栏中的“保存”按钮保存编辑内容，如果用户直接按了手机的返回键，则提醒用户是否放弃修改的内容，用户选择放弃，则不保存修改的内容，直接返回首页，否则，保存修改的内容，然后返回首页。

#### 文本记事

1. 要求

能够输入多行文本，进行简单的格式控制。

1. 原型图

文本记事的原型图如图3.4

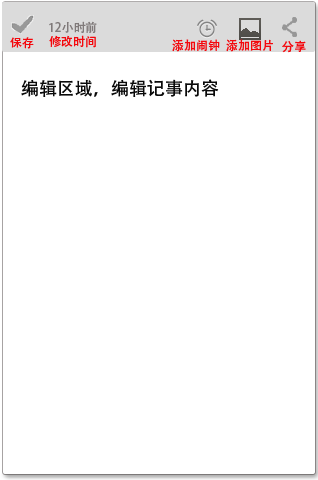


图3. 4文本记事原型图

如图3.4所示，记事的编辑页面主要由两部分组成，导航栏和编辑区域。导航栏主要包括5部分，分别是“保存”“修改时间”“添加闹钟”“添加图片”“分享”。

每部分的作用如下：

修改时间：记录了记事内容创建或修改的时间。

保存：用来保存记事的内容，只有选择了保存，修改的内容才能保存。

添加闹钟：作为“闹钟”功能的入口。

添加图片：作为“添加图片”功能的入口。

分享：分别作为“分享”功能的入口。

导航栏下面就是文本的编辑区域，可以在此编辑修改内容。对于语音记事和图片记事，编辑页面的导航基本类似，就不再赘述。

#### 图片记事

1. 要求
2. 支持调用图库选择图片
3. 支持调用相机拍照上传
4. 支持输入多行文本作为备注
5. 支持添加多张图片
6. 图片和文本的位置可任意，展示时保持相对位置不变
7. 原型图

选择添加图片原型图如图3.5



图3. 5添加图片原型图

如图3.5所示，添加图片，分两种方式：拍照和从相册中选择。

图片记事编辑页面的原型图如图3.6



图3. 6图片记事编辑原型图

如图3.6所示，图片记事可以添加图片，也可以输入文本，并且图片和文本的位置可以随意。

#### 语音记事

1. 要求
2. 支持录入语音
3. 支持添加多行文本作为备注
4. 支持录入多个语音
5. 原型图

语音记事的原型图如图3.7

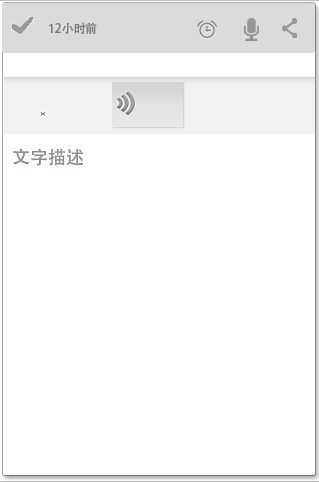


图3. 7语音原型图

如图3.7所示，编辑区域主要包括两部分，语音部分和文本部分，其中语音部分是在语音结束后自动添加的，这部分包括一个语音播放按钮，点击可以播放语音。

### 闹钟模块

#### 总述

闹钟模块的设置主要包括“日历”“时间”“弹出间隔”“弹出次数”，闹钟操作包括“新建”“编辑”“删除”。

新建闹钟的原型图如图3.8

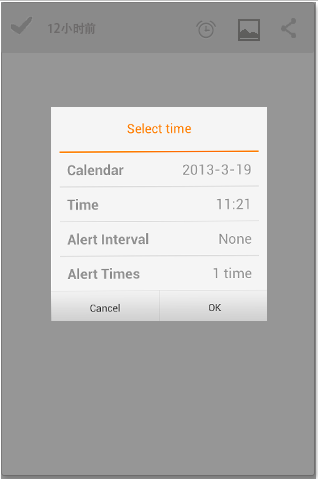


图3. 8新建闹钟原型图

如图3.8，新建闹钟时，日历和时间默认显示当前的日期时间，提醒次数默认一次，用户可以选择每一项来重新设置值。

编辑删除闹钟的原型图如图3.9

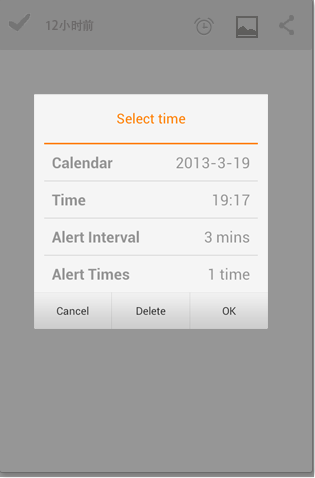


图3. 9编辑删除闹钟原型图

如图3.9，用户添加闹钟后，再次点击闹钟按钮时，会显示闹钟的全部设置，用户可以修改其中的每一项，也可以选择删除该闹钟。

#### 日历模块

1. 要求
2. 支持日期的选择
3. 支持农历
4. 支持阳历
5. 支持月份之间的切换
6. 原型图

日历的原型图如图3.10

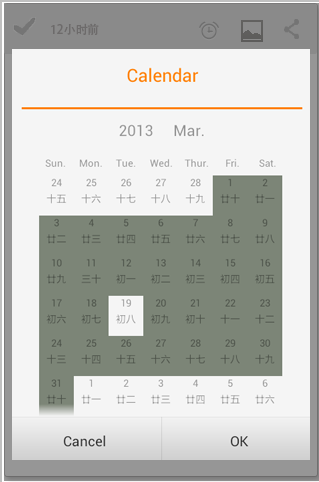


图3. 10日历原型图

如图3.10，日历包括农历和阳历，用户可以选择日期，被选择的日期以不同的背景色标识。

#### 提醒次数

1. 要求

能够选择提醒次数，包括一次、两次、三次、五次

1. 原型图

提醒次数的原型图如图3.11

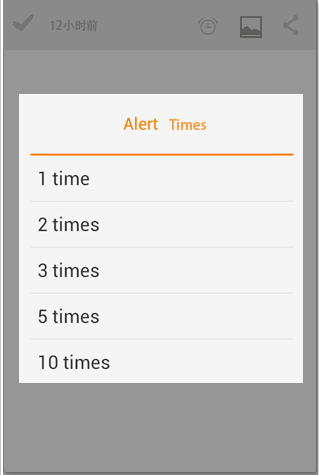


图3. 11提醒次数原型图

#### 提醒间隔

1. 要求

能够选择提醒的间隔，间隔1分钟、2分钟、3分钟、5分钟

1. 原型图

提醒间隔的原型图如图3.12

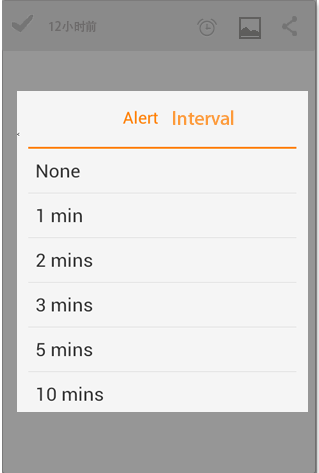


图3. 12提醒间隔原型图

### 分享模块

#### 分享出去

1. 要求
2. 支持分享到网络应用（如花瓣，evernote）
3. 支持短信分享
4. 支持微信分享
5. 支持微博分享
6. 原型图

分享原型图如图3.13

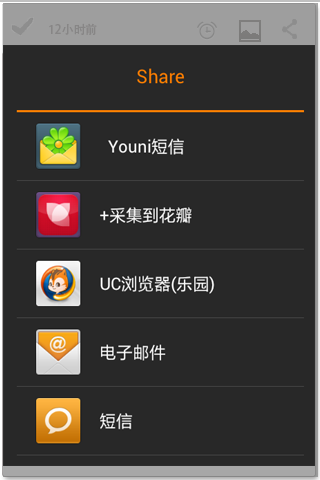


图3. 13分享原型图

#### 分享进来

1. 要求

接受从其他应用中分享来的内容，作为新的记事添加

1. 原型图

不需要额外的原型图，类似于新建一个记事。

### 备份模块

1. 要求
2. 支持把数据备份到云端
3. 支持从云端恢复数据
4. 原型图

原型图暂无

## 数据库概要设计

记事本共包括两个实体，Note和Clock，数据库建表时，也需要建立两个表，Note表和Clock表。

### E-R图

1. Note实体主要包括编号，文本内容，创建时间，修改时间，图片路径，语音路径，位置序号。E-R图如图3.14



图3. 14 Note实体E-R图

1. Clock实体主要包括编号，日期时间，提醒次数，提醒间隔。E-R图如图3.15



图3. 15 Clock实体E-R图

### 数据字典

数据字典主要包括数据项，数据含义，别名，数据类型，是对实体所包含信息的一个直观的描述，是逻辑结构设计和数据表设计的基础。

Note实体的信息表如表3.1

表格 3.1 Note的信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **数据项** | **数据含义** | **别名** | **数据类型** |
| 记事编号 | 记事的唯一标识 | NoteId | Varchar |
| 文本内容 | 记事的文本内容 | Content | Varchar |
| 图片 | 图片路径 | Picture | Varchar |
| 语音 | 语音路径 | Audio | Varchar |
| 创建时间 | 记事的创建时间 | CreateTime | Date |
| 修改时间 | 记事的修改时间 | ModifyTime | Date |

Clock实体的信息表如表3.2

表格 3.2 Clock的信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **数据项** | **数据含义** | **别名** | **数据类型** |
| 闹钟编号 | 闹钟的唯一标识 | ClockId | Varchar |
| 记事编号 | 记事的唯一标识 | NoteId | Varchar |
| 时间 | 闹钟的提醒时间 | Time | Date |
| 提醒间隔 | 闹钟提醒的间隔 | Alert\_interval | Int |
| 提醒次数 | 闹钟提醒的次数 | Alert\_times | Int |

# 详细设计

## 功能设计

为了方便代码编写与管理，本次毕业设计中对记事本的开发采用模块化的总体设计方案，可以将其大致分为这样四个模块：记事模块、闹钟模块、分享模块和备份模块，其中记事模块包括文本记事，音频记事，图片记事三个模块，这几个模块共同构成了将要设计并实现的基于android的记事本。记事本的总体操作流程图如图4.1



图 4.1 记事本总体流程图

如图4.1所示，运行记事本，进入的首页，首页是记事本的每个记事列表，选择新建记事，则调用记事模块，如果是文本记事，则编辑文本内容；如果是音频记事，则调用音频模块，录入音频，同时可以添加文本作为补充内容；如果是图片记事，则调用图片模块，拍照或者从图库中选择图片，同时可以输入文本作为补充内容。如果需要添加闹钟，则调用闹钟模块，添加闹钟，如果要分享记事内容，则调用分享模块，分享记事内容。编辑完成后，选择保存，结束记事。

### 记事模块



#### 新建记事

新建记事，是整个记事本最基本的功能，也是最核心的功能，其他所有的功能，例如闹钟、分享，都是以记事为中心展开的。这个模块分为三个主要部分：新建文本记事，新建图片记事，新建音频记事，分别对应的操作是编辑文本，插入图片，录入音频。新建记事的流程图如图4.2



图4.2 新建记事流程图

如图4.2所示，新建记事，进入的编辑页面，可以选择编辑文本，插入图片，录入音频来编辑记事，编辑完成后，可以选择保存记事或者放弃保存直接返回列表。流程图1.2是不涉及闹钟和分享模块的，只是单纯的记事，如果想为记事添加闹钟或者分享记事，对应的操作就是添加闹钟和分享，这部分在后面将详细讲述。

此次毕业设计所设计的记事本，新建记事还有另外一种方式，就是接收分享时，新建一个记事，把接收到的内容保存起来，这部分会在分享模块中详细介绍。

无论是新建记事还是编辑记事，都需要更新列表，更新列表主要包括更新记事内容，修改时间以及闹钟三部分，其流程图如图4.3



图 4.3 更新列表流程图

如图所示，更新记事列表时，分两种情况，如果是新建记事，则需要在在列表中插入新建的记事，如果是编辑记事，则只需更新编辑过的记事，然后更新相应的内容修改时间等。

1. **文本记事**

文本记事是指只有单纯的文本，没有图片和音频，相对比较简单，新建记事，直接输入文本保存即可，其中保存的内容包括“记事内容”“创建时间”“修改时间”以及闹钟信息。

1. **图片记事**

图片记事中需要调用图片模块，为记事添加图片。在本次应用中，图片记事是最复杂的，也是做的最好的，原因是图片和文本混合在一起，图片的位置随意，完全由用户控制，图片和文字紧密结合在一起，并且支持多个图片的插入。

图片记事流程图如图4.4



图 4.4 图片记事流程图

如图4.4所示，选择添加图片按钮，从图库中选择图片或者用相机拍照，获取图片的URI或者DATA，然后利用图片的URI和DATA将图片转换成位图（Bitmap），由于选择的图片一般都比较大，所以必须进行一定的压缩，压缩分为两方面，一方面是对图片尺寸的压缩，将图片的尺寸缩小，另一方面，就是将Bitmap压缩成png或者jpg等普通的图片格式，对Bitmap处理完成后，将图片保存在file输出流（Output Stream）中，最后一步就是将图片显示在记事中。

将图片显示在记事中，通常有两种方法，一种是新建一个Image View，将图片显示在Image View中，这种方法处理起来比较简单，因为Image View本来就是承载图片用的；另一种方法是直接将图片放在Edit View中，这种方法处理起来比较困难，也不是常规的承载图片的方法，Edit View是用来承载文字的，并不能添加图片，所以想要Edit View显示图片，只能通过一些特殊的方法。

这两种方法我选择了后一种，将图片显示在Edit View中，原因是本次设计的记事本，图片记事可以添加多张图片，并且图片和文字的位置可以随意，我们可以在文本的任意地方添加图片，如果采用动态创建Image View的形式，就不能做到这么灵活，所以采用Edit View添加图片的方案。

由于Edit View本身并不能添加图片，所以让图片显示在Edit View中，需要借用Image Span和Spannable String这两个对象，将图片当成文本一样来处理。这样在保存图片时，也需要将图片和文本保存在一个字段中，显示记事内容时，重新解析记事内容，分离出图片和文本。

1. **音频记事**

音频记事，利用的是android自带的Audio类。用户新建一个记事，可以选择为记事添加音频录制，同时还可以添加一些文本作为备注。在音频记事中，添加的音频默认在记事的最前面显示，以一个播放按钮来表示，用户可以点击此按钮播放音频，同时也可以重新录制一段音频来代替原有的音频。

音频记事流程图如图4.5



图 4.5 音频记事流程图

如图4.5所示，选择“添加音频”按钮，则调用音频记事模块。弹出音频录制的dialogue，其中dialogue包括“Start”“OK”“Cancel”三个按钮，选择“Start”开始录制音频，此时“Start”变成“Stop”，选择“Stop”停止录制，选择“OK”保存音频，选择“Cancel”则取消保存。保存音频后，更新记事内容，为记事添加一个音频播放按钮，用户可以选择此按钮播放录制的音频，更新数据库，将音频的路径保存在数据库中。

#### 编辑记事

选择首页记事列表中的每一个记事，就可以进入相应的查看编辑页面，编辑页面和新建页面基本是一样的，调用的是同一个activity，不同的是新建记事页面是完全空白的，而编辑记事页面需要从数据库中加载记事内容以及闹钟信息。

编辑记事的流程图如图4.6



图 4.6 编辑记事流程图

如图4.6所示，进入编辑页面，初始化记事内容和闹钟信息。用户在此可以只查看这些信息，也可以对此进行任意的编辑，例如添加文字、添加图片、添加音频、修改闹钟等。编辑完成后，可以选择导航栏中的“保存”按钮保存，如果不想保存，则可以直接按系统的返回键，这时会提醒用户是否放弃修改的内容，如果用户选择是，则直接返回首页列表，如果选择否，则保存修改的内容到数据库，返回列表并且刷新列表，更新记事内容。

#### 删除记事

对于android系统，长按事件起着重要的作用，尤其是在删除操作上，对于一般的应用，长按事件都会出现相应的删除操作，所以在本次的记事本应用中，删除操作同样绑定在长按事件上。

删除记事的流程图如图4.7



图 4.7 删除记事流程图

如图4.7所示，用户长按某一记事，则出现“删除”按钮，用户可以选择删除相应的记事，选择删除后，会让用户再次确认是否删除，如果用户确定了删除，则把记事从列表中删除，更新数据库，把相应记事的数据以及相关联的闹钟一起从数据库中删除，同时更新记事列表，对每个记事重新排序。

另一方面，在“删除按钮”显示的时候，也会出现“取消按钮”，用户可以直接选择“取消”来取消删除操作，或者直接选择系统的返回键，也可以取消删除按钮。

### 闹钟模块

#### 新建闹钟

选择导航栏的“闹钟”按钮，即弹出新建闹钟的dialogue，在此可以设置闹钟的各项参数：日期，时间，提醒间隔，提醒次数。其流程图如图4.8



图 4.8 新建闹钟流程图

如图4.8所示，弹出新建闹钟的dialogue，依次设置“日期”“时间”“提醒次数”“提醒间隔”，对于新建操作，“日期”“时间”“提醒次数”“提醒间隔”的初始值都为默认值。设置完成后，选择OK，则会保存闹钟信息，同时返回编辑页面，在导航栏的“闹钟”图标旁边显示闹钟的提醒日期。

导航栏上显示的闹钟日期有三种格式：如果闹钟的提醒时间是当天，则日期只显示“时分（xx:xx）”；如果不是当天，则显示“月日（xx-xx）”，如果不是当年，则会连同年份一起显示（xxxx-xx-xx）。

保存闹钟信息，涉及数据库操作，操作的主要是clock表，对于新建的闹钟，需要在数据库clock表中插入一行数据，数据包括当前clock对应的noteID，闹钟提醒时间（TIME），提醒间隔（ALERT\_INTERVAL），提醒总次数（ALERT\_TIMES），剩余次数（ALERT\_COUNT），新建闹钟时，剩余提醒次数和提醒总次数值是一样的。

如果想放弃设置闹钟，则可以选择Cancel，直接返回列表。

#### 编辑查看闹钟

编辑查看闹钟和新建闹钟在界面上差不多，编辑闹钟dialogue只是比新建闹钟多一个“Delete”选项，用户可以在此查看闹钟的设置信息，修改闹钟，或者删除此闹钟。其流程图如图4.9



图 4.9 编辑查看闹钟流程图

如图4.9所示，弹出闹钟dialogue时，判断已经存在闹钟，则说明此操作是编辑或者查看闹钟操作，因此根据noteID查找clock表，读取此闹钟信息，分别把“日期”“时间”“提醒次数”“提醒间隔”初始化为用户所设置的值。此时，用户可以只查看相应的数据，也可以修改每一项内容，或者选择Delete按钮，删除此闹钟。

删除闹钟时，需要从数据库clock表删除此闹钟的整行数据，同时删除导航栏的闹钟日期。

修改闹钟保存信息时，不需要插入新数据，只需根据noteID查找clock表，找到对应的闹钟数据，UPDATE数据就可以了，同时更新导航栏的闹钟日期。

#### 闹钟提醒

当设置的闹钟时间到时，系统会启动闹钟，以弹出框dialogue的形式的提醒用户，其中弹出框的包括记事的内容摘要以及“删除闹钟”“查看记事”“知道了”三个按钮。

闹钟提醒流程图如图4.10



图 4.10 闹钟提醒流程图

如图4.10所示，当设置的闹钟时间到时，首先获取相应记事的内容，然后系统启动闹钟，弹出提醒框，同时需要更新闹钟数据表，把其中的“剩余提醒次数”字段值减1；弹出框的内容设置为记事的内容，同时为设置“查看”“知道了”按钮；选择性添加“删除”按钮，判断是否为最后一次提醒，如果不是，则添加“删除”按钮，如果是，则不添加“删除”按钮，在闹钟提醒过后自动注销该闹钟。

用户收到提醒后可以选择三个按钮的任意一个进行相应的操作，其中“删除闹钟”会直接删除该记事设置的闹钟；“查看记事”则进入相应的记事页面，查看完整的记事内容；“知道了”则弹出框直接消失，不进行其他操作。

如果用户没有删除闹钟，并且闹钟设置了多次提醒，则闹钟会继续以设置的提醒间隔来再次提醒用户，直至用户删除了闹钟或者闹钟自动注销为止。

### 分享模块

分享模块主要包括分享出去和接收分享，分享的内容分为四种类型：文本分享、单图片分享、多图片分享、音频分享。下面对这几种分享类型详细介绍。



#### 分享出去

分享出去利用的是intent，设置intent的action为ACTION\_SEND，然后设置intent的type，即设置不同的分享类型来实现图片文字分享。这里图片文字的分享跟音频的分享的不同，音频的分享是单独的，利用长按事件为音频单独添加分享。

分享出去流程图如图4.11



图 4.11 分享出去流程图

如图4.11所示，选择导航栏上的“分享”按钮，则调用分享模块。首先获取该记事中的图片列表，如果列表为空，则表示该记事没有图片，则分享的是简单文本，设置intent的type为“text/plain”；如果图片个数为1，则说明分享的内容只有一个图片，则分享的是单图文，设置intent的type为“image/\*”；如果图片个数大于1，则分享的是多图文，设置intent的type为“image/\*”，同时修改intent的action为ACTION\_SEND\_MULTIPLE。这里对图片的处理和图片记事中的处理是一样的，在此不作赘述。

分享音频时，选择相应的音频，长按音频，弹出分享对话框，用户可以选择将此音频分享出去，分享音频设置intent的type为“audio/\*”。

最后将分享的内容附加到intent中，启动分享activity，从中选择应用将内容分享出去。

#### 接收分享

接收分享也分为四部分：纯文本分享，单图文分享，多图文分享，音频分享。接收分享利用的同样是intent，通过判断intent的action和type来判断是哪一种分享。这里需要重新建立一个新的activity，作为处理接收分享的activity，另外，分享可接收的几种类型（text，image，audio）也需要在AndroidManifest.xml里面注册，否则无法接收相应类型的分享。接收分享流程图如图4.12



图 4.12 分享出去流程图

如图4.12所示，如果系统的收到其他应用分享来的数据时，启动分享接收的activity，获取其中intent，判断intent的action和type属性，如果action是ACTION\_SEND，并且type为text，则接收的是文本分享，如果type为image，则接收的分享是单图文分享，如果type为audio，则接收的分享是音频分享；如果action是ACTION\_SEND\_MULTIPLE，则接收的分享是多图文分享。确定接收的分享类型后，获取需要的数据，然后调用新建记事模块，新建一个记事，保存接收的分享数据。

### 备份模块

备份模块主要包括两种情况：备份到云端、从云端恢复。在云端是以用户的手机号为名称建立的文件，每次备份都会覆盖原有的备份文件。从云端恢复，将会下载云端的数据文件到本地，覆盖本地的数据库，将所有的记事更新成恢复的数据。这里的难点是，备份及恢复时，数据格式的设计以及购买服务器的问题，现在还没有完全免费的服务器。

备份模块的流程图4.13



图 4.13 备份流程图

## 数据库设计

### 数据库分析

本次的记事本应用，模块分为记事模块和闹钟模块，记事虽然包括三种记事，但是仍然可以设计成一个表，这样方便处理，所以数据库只需设计两个表notes表和clock表。notes表用了存储记事（包括图片记事，文本记事，音频记事）相关的信息；clock表存储闹钟相关的信息。

### 逻辑结构设计

1. **NOTES表**

NOTES(ID，CLOCK\_ID，CONTENT，CREATE\_DATE，MODIFY\_DATE，SEQUENCE，AUDIO\_URI)

主键：ID

外键：CLOCK\_ID

NOTES表设计如表格4.1

表格 4.1 NOTES表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实体** | **属性** | **描述** | **数据类型** | **长度** | **NULL值** | **默认值** |
| NOTES | ID | 记事编号 | INTEGER | 系统默认 | NOT NULL | 无 |
| CLOCK\_ID | 闹钟编号 | INTEGER | 系统默认 | NOT NULL | 0 |
| CONTENT | 记事内容 | TEXT | 变长 | NOT NULL | 空字符串 |
| CREATE\_DATE | 记事创建时间 | INTEGER | 系统默认 | NOT NULL | 0 |
| MODIFY\_DATE | 记事最新修改时间 | INTEGER | 系统默认 | NOT NULL | 0 |
| SEQUENCE | 记事的顺序 | INTEGER | 系统默认 | NOT NULL | 0 |
| AUDIO\_URI | 音频存放路径 | TEXT | 变长 | NOT NULL | 空字符串 |

如表格4.1所示，ID作为NOTES表的主键，是每一个记事（note）的唯一标识，是不可重复而且不可为空的，插入每一篇记事时，ID由数据库自动生成，，每次对NOTES数据表进行操作（查询、修改、删除）都是以识别ID来进行操作的。

CLOCK\_ID作为NOTES表的外键，是连接NOTES表和CLOCK表的桥梁，它代表的就是CLOCK表中的ID，所以CLOCK\_ID也是唯一不可为空的，数据库查询、删除、修改时，通过匹配NOTES表的CLOCK\_ID和CLOCK表的ID，把记事和闹钟联系在一起。

CONTENT是记事的内容，但是它实际比较特殊，它保存的不仅是文本内容，如果有图片，它还会保存图片的信息（图片路径）。之所以这样设计，是因为图片和文本的是在同一个Edit View里面，文本和图片的位置是可以任意的，文本可以环绕图片，因此必须保证每次打开记事，文本和图片的位置和新建时候的一致，若是文本和图片分开保存，就无法记录它们的位置信息，因此把图片路径当做文本的一部分，一起保存起来，当加载记事时，需要对CONTENT解析，解析出图片路径，将其替换成图片。

CREATE\_DATE和MODIFY\_DATE是记事的创建时间和最新的修改时间，时间在这里是以毫秒来保存的，因此格式是Integer，而不是Date，以毫秒存储时间也方便修改时间的显示格式。

SEQUENCE是记事的顺序，主要记录每个记事在首页列表的显示位置，当更新列表时，可以保证每个记事的相对位置不变。

AUDIO\_URI保存的是音频的路径，主要用于音频记事中。

1. **CLOCK表**

CLOCK(ID，TIME，ALERT\_INTERVAL，ALERT\_TIMES，ALERT\_COUNT)

主键：ID

外键：无

CLOCK表设计如表格4.2

表格4.2 CLOCK表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实体** | **属性** | **描述** | **数据类型** | **长度** | **NULL值** | **默认值** |
| CLOCK | ID | 闹钟编号 | INTEGER | 系统默认 | NOT NULL | 无 |
| TIME | 闹钟时间 | INTEGER | 系统默认 | NOT NULL | 0 |
| ALERT\_INTERVAL | 提醒间隔 | INTEGER | 系统默认 | NOT NULL | 0 |
| ALERT\_TIMES | 提醒总次数 | INTEGER | 系统默认 | NOT NULL | 0 |
| ALERT\_COUNT | 剩余提醒次数 | INTEGER | 系统默认 | NOT NULL | 0 |

如表格4.2所示，ID是CLOCK的主键，是每个闹钟的唯一标识，不可重复，不可为空，NOTES表中的CLOCK\_ID就是跟ID联系在一起的。新建一个闹钟时，ID由数据库自动累计生成，每次对CLOCK数据表进行（查询、修改、删除）都是以识别ID来进行操作的。

TIME是闹钟提醒的时间，包括从日历选择的日期（年月日）和时间（时分），日期和时间整合到一起，以毫秒存储，数据类型为Integer类型，这样方便管理。

ALERT\_INTERVAL和ALERT\_TIMES是闹钟提醒的间隔和提醒总次数 ，是用户在设置闹钟选项时，所设置的值。

ALERT\_COUNT是剩余的提醒次数，由于设置闹钟提醒时，如果要提醒多次，除了要知道提醒的总次数，还必须知道剩余的提醒次数是多少，以剩余的提醒次数来判断闹钟是否需要继续提醒，每次闹钟提醒后，都必须更新ALERT\_ COUNT的值。

# 系统实现

## 功能实现

### 总述

#### Android基本结构

一般android的应用程序主要有四种构成部分：Activity，Intent，Service，Content Provider。在本次的记事本应用程序中，用到了其中的Activity，Intent，Content Provider三个部分。

另外一个比较重要的文件是androidManifest.xml，这个文件用来声明应用程序组件和它们的功能需求是什么，当应用程序中需要某些组件时，必须在这个xml文件中声明才可用。

1. **Activity**

Activity是四个构建模块中最普通的一个，一个activity在应用程序中通常是一个单独的屏幕，每一个activity是实现作为一个继承于activity基类的单独的类。例如在本次的记事本应用中，首页的记事列表是一个屏幕，编辑查看记事的页面是另一个屏幕，它们都是作为单独的activity存在的，当新建或者选择某一个记事时，就会进入编辑查看记事的页面，这一过程是通过启动一个activity来实现的。并且当一个新的屏幕开启，先前的屏幕被暂停压入历史堆栈，用户可以向后导航返回到先前打开的屏幕。

1. **Intent和IntentFilter**

Android使用一个名为intent的特殊类让应用程序从一个屏幕移动到另一个屏幕，intent描述了一个应用程序想要干什么。Intent数据结构的两个重要部分是：要响应的action和data。典型的action值MAIN，VIEW，PICK，EDIT。Data表示成一个统一资源定界符。

Intent是一个有效的请求来做什么事，而IntentFilter是描述一个activity能够处理什么样的Intent。

从一个屏幕到另一个屏幕的导航通过解决Intent来完成。要向前导航，一个activity调用startActivity(myIntent)。系统接下来查找所以应用程序的IntentFilter，并选择一个IntentFilter最适合myIntent的activity。

1. **Content Provider**

应用程序可以把它们的数据存储到文件中，一个SQLite数据库中，设置中，或者其他任何有意义的结构中。Content Provider可以让应用程序之间共享数据，一个content Provider是一个类，它实现了方法的一个标准设置来让其他应用程序存储和恢复被Content Provider处理的数据类型。

#### 系统架构设计

本次的记事本应用中共分为四个包：activity，model，tools，view。

activity包里面的类继承自基类activity，主要用来处理两个activity之间的跳转和数据传递，包括的类有：ClockAlertActivity.java，ClockReceiver.java，NoteEditActivity.java，ShareReceiver.java。

model包里面的类用来从数据中读取数据或者存储数据，包括的类有：ClockModel.java，NoteItemModel.java，分别处理闹钟数据和记事数据。

view包里面的类继承自基类view，用来控制界面的显示和事件的侦听，包括的类比较多，有MainView.java，NoteItemView.java，NoteListView.java，ClockCalendar.java等。

Tools包里面只有两个类：Lunar.java，DBHelper.java。Lunar类用来处理农历的显示，DBHelper类封装了各种数据库操作。

### 记事模块



记事模块的实现主要分为NoteEditView，NoteItemView，NotesListView，NoteItemModel，NoteEditActivity，MainActivity，MainView这几个类，每个类控制着不同的部分。

NotesListView类控制的是首页记事列表的展现，包括加载每个记事item，更新列表以及每个item的点击事件处理等。

NoteItemView类控制的是记事列表中每个item的展现，包括两行记事内容，修改时间，是否有闹钟以及删除事件的处理。

NoteEditView类控制的是记事编辑页面，包括导航栏上各个按钮的点击事件处理以及编辑记事内容，添加图片、音频、闹钟等操作都是以这个页面为入口。

NoteItemModel类主要是用来获取每个记事的数据的。

NoteEditActivity类是一个activity，用来加载EditView视图以及添加图片的一些处理。

MainActivity和MainView类主要是用来控制首页的显示以及处理首页的一些事件，如“新建记事”事件，其中MainActivity是程序的入口。

#### 记事MODEL

在model包中，有NoteItemModel类，这个类的作用是创建一个记事的实例，这个实例包含了记事的所有信息，包括内容、创建时间、闹钟等。它有四个构造函数，可以创建空的实例或者根据参数从数据库中读取相应记事的所有数据，供其他类使用，它是联系数据库和视图的桥梁。

在这个类中，最重要的就是构造函数，当我们新建一个记事时，需要创建一个空的NoteItemModel实例，而当编辑记事或者需要获取记事的某一项数据如记事内容时，可以调用以记事的id为参数的构造函数，该构造函数先根据id从数据库中查询记事，再把记事的每一项数据赋值给实例属性。

在这个类中，还包括获取和设置实例属性的方法，如getId和setId分别用来获取和设置id属性，getModifyDate和setModifyDate分别用来获取和设置修改时间属性等。

#### 添加图片

在图片记事中除了添加文字之外，最重要的功能是添加图片，添加图片的功能主要在NoteEditActivity.java和NoteEditView.java中实现，其中NoteEditView类主要是为按钮注册事件，而对图片的数据处理都是在NoteEditActivity类中完成的。

在NoteEditView.java中，为导航栏上的图片按钮注册点击事件，一旦侦听到点击事件，即调用addPicture函数。addPicture函数的主要作用是启动“相机”或者“图库”程序。启用其他的应用程序或者activity需要用到android的Intent类，这里我们需要创建一个Intent实例，利用不同的参数来告诉Intent是启动“相机”还是启动“图库”，如果是启动“相机”，则参数为*android.media.action.IMAGE\_CAPTURE*，如果启动“图库”，则参数为*Intent.ACTION\_GET\_CONTENT，*最后调用startActivityForResult来启动应用程序。startActivityForResult有两个参数，第一个参数就是我们创建的Intent实例，第二个参数作为requestCode，标识是“拍照”还是“从图库中选择”。

在NoteEditActivity.java中，接收“相机”或者“图库”返回的数据的是onActivityResult函数，这个函数有两个参数requestCode和intent，intent承载着返回的图片数据，而requestCode则标识是相机返回的数据还是图库返回的数据，从而进行不同的处理。

如果是相机返回的数据，则利用intent.getData()获取数据，获取的数据是Uri实例，需要用BitmapFactory类把数据转换成bitmap（位图）；如果是图库返回的数据，则利用intent.getExtras()获取数据，获取的数据是Bundle实例，需要用get("data")获取bitmap（位图）数据。

不管是那种格式的数据，最终都将转换成bitmap，然后对bitmap进行处理。这里主要有三种处理：

首先，对bitmap进行缩小。为了保证图片不变形，这里采用的是等比缩小，并且图片的最终width不超过300px，height不超过200px，最终得到图片高度和宽度的缩放比scaleWidth和scaleHeight，再利用matrix.postScale(scaleWidth, scaleHeight)实施缩放动作，最后用Bitmap.createBitmap(originalBitmap, 0, 0, width, height,matrix, true)生成新的bitmap图片。

第二，保存图片。这里将图片保存到sdcard里面，格式为jpeg格式，并且为了保证图片命名不冲突，以当前的时间戳作为图片的名称，先利用图片路径创建File文件实例，再利用FileOutputStream类创建输出流outStream，最后将outStream作为参数传递给bitmap.compress函数，将输出流压缩成jpeg格式保存。

第三，显示图片，将图片显示在EditView里面。这种显示图片的方式比较困难，因为EditView本身是不能显示图片的，所以这里采用了变通的方式，即将图片当成文本一样来处理，这样的一个好处是我们可以添加多张图片，并且图片和文字的位置可以随意，我们可以在文本的任意地方添加图片，用户体验更好。把图片当成文本来处理，需要用到的是Image Span和SpannableString这两个类，这两个类的作用是可以对文本进行各种特别的设置，比如颜色、大小、首行缩进，或者是在一段文本中加入图片。

最后就是将图片路径连同记事内容保存在数据库中。在显示图片时，我们创建了一个SpannableString实例，它的内容是 [ox64+图片路径]字符串，我们先显示图片时，实际上只是用图片覆盖了这些字符串，但字符串还是存在的，它作为内容的一部分，所以选择导航栏上的保存按钮，即可将记事的内容保存在数据库中。

#### 添加音频

添加音频与添加图片稍微有点类似，在记事编辑页面的导航栏中，有一个audio图标，我们为这个图标绑定点击事件，当发生点击事件时，就会弹出音频的录制框，里面有三个button：“start”“ok”“cancel”，选择“start”button即可开始音频录制，并且这个button的内容会随着音频的不同状态而改变，点击后触发的操作也会随之不同。例如当音频开始录制时，这个button内容变成“stop”，点击即可停止录制，之后button变成“play”，点击即可播放刚才录制的音频。为了实现这个功能，需要为音频设置四个状态，分别是unstart，recording，recorded，playing，通过这四个状态来判断button点击后应该执行哪种操作。

在本次的开发中，对音频的操作涉及到“录制”、“播放”，“暂停”，“保存”，“删除”，这些操作都封装在NoteEditAudioView.java这个类中。

音频的录制主要利用的是android的MediaRecorder类，这个类是专门用来录制音频的，它包含了录制音频基本的方法，如prepare()，start()，stop()，release()，利用这些方法可以实现音频的录制和停止功能。

音频的播放利用的是android的MediaPlayer类，这个类包含的方法跟MediaRecorder类似，它可以用来实现音频的播放和暂停功能。

音频的保存与图片的保存方式类似，以时间戳为名称保存成3gp格式。与图片不同是，在数据库中，音频的路径保存在单独的字段audio中，而图片的路径保存在与文本相同的字段。

#### 解析内容

由于记事中可能会含有图片，甚至是多张图片，而图片路径和文字都保存在数据库同一个字段中，所以在显示内容时，需要对内容进行解析，将其中的图片路径替换成图片显示。

对于含有图片的记事内容，我们保存时都是将图片的路径连同记事的文本内容一起保存的，其中图片的路径前面含有”[ox64”字符，后面为”]”字符，解析时就说以此为标识解析的。解析内容的方法为setContent(String content)。

首先，需要先将content显示在记事中，this.mContent.setText(content)。

然后，解析出图片的路径，这里需要一个位置的标识变量index，初值为0。获取”[ox64”字符串在content中从index开始第一次出现的位置index，index+5即为图片路径的开始位置start，然后获取”]”在content中从start开始第一次出现的位置end，为图片路径的结束位置，content.substring(start, end)即可获取图片的路径。

最后，将图片显示在内容中，方法与添加图片的时相同，先把图片编码成bitmap，再利用ImageSpan和SpannableString将图片显示在记事内容中。因为会有多张图片，所以需要一个while循环遍历出所有的图片，每次循环后，index都要加1进入下次循环。代码如下：

#### 其他

1. **新建记事**

新建记事在MainView和NoteEditActivity类实现，在MainView中有个新建记事按钮，选择新建记事时，利用intent启动NoteEditActivity这个activity，同时用Bundle传递数据，传递的数据为ID=-1，NoteEditActivity通过ID=-1来识别该操作是否为创建新记事的操作，当NoteEditActivity获取ID为-1时，就会创建一个空的NoteItemModel实例。新建记事时，需要将状态设置成STATUS\_CREATE。

1. **删除记事**

在首页上，长按某一个记事item时，就会出现“delete”按钮，这一事件是在NotesListView类中注册的，当侦听到长按事件时，就会显示delete按钮和cacel按钮，选择“delete”时，将状态设置成STATUS\_DELETE，弹出窗口让用户确认是否删除，当确认删除时，就会从数据库将该记事以及与该记事相关的闹钟删除。

1. **编辑记事**

编辑记事的实现主要在NoteEditView类中。当点击记事列表中的某一个item记事时，就会启动NoteEditActivity，将该记事的id传递给NoteEditActivity，并将状态设置成STATUS\_EDIT。NoteEditActivity获取id，判断大于0，则说明是编辑记事，就会调用NoteEditView类中的setNoteEditModel方法，这个方法首先根据id从数据库中查询得到cursor；然后以cursor为参数，创建一个记事model实例mNoteItemModel，以便获取记事的所有数据；最后调用setContent方法将记事内容显示，判断该记事是否含有audio，如果有则显示一个audio播放图标，继续判断该记事是否设置了闹钟，如果设置了闹钟，则将闹钟的时间显示在导航栏中。

### 闹钟模块



闹钟的实现主要利用的是android系统的AlarmManager以及Broadcast机制。首先我们需要设置闹钟的参数，包括弹出日期，弹出时间，弹出间隔，弹出次数，设置好这些参数后，一件很重要的事情就是为该闹钟启动一个AlarmManager进程，只有为闹钟启动了AlarmManager进程，闹钟才能正常工作，并且我们还需要用intent去调用ClockReceiver.class（广播接收器），将此记事的id传递给ClockReceiver，这样在闹钟时间到时就可以接收闹钟广播（这里利用的就是Broadcast机制），启动闹钟提醒。

#### 闹钟MODEL

对闹钟参数的设置和获取以及数据库的操作都封装在ClockModel.java类中，这个类在model包中，它的主要作用的从数据库中读取设置闹钟的各种参数，供其他类使用，同时也负责新建闹钟，把用户设置的各项参数保存到数据库。它有两个构造函数：ClockModel()和ClockModel(Cursor cursor)，前者用来新建一个空的clock实例，后者则从数据库中读取闹钟的信息，建一个带有各项数据的clock，供其他类使用。

另外一个比较重要的方法是formatContentValues，它的作用是更新闹钟的数据，利用的是ContentValues类实现的。

ClockModel.java中还封装了设置和获取闹钟各项数据信息的方法，例如：

getAlertInterval ()：用来获取闹钟的弹出间隔；

setAlertInterval (int alertInterval)：用来设置闹钟的弹出间隔；

addAlertCount()：用来更新闹钟的已经弹出的次数，闹钟每弹出一次都会调用这个方法来更新数据库，ALERT\_COUNT字段加1；

isClockFinished()：用来判断闹钟是否结束，根据闹钟的弹出次数ALERT\_TIMES以及已经弹出的次数ALERT\_COUNT来判断；

另外还有设置和获取闹钟时间、闹钟弹出次数等信息的方法，由于比较多，在这不一一列举了。

#### 日历实现

日历部分的实现主要分为两部分，一部分是农历的实现，一部分是日历视图的实现。日历视图的实现利用的是GridView网格布局，每一个网格是一个日期item，包括阳历和公历。这里重点介绍农历是怎样实现的。

农历的实现封装在lunar.java类中，主要是利用一份1900--2049的阴历月份的数据实现的，数据如下：

**final** **static** **long**[] *lunarInfo* = **new** **long**[] { 0x04bd8, 0x04ae0, 0x0a570,

0x054d5, 0x0d260, 0x0d950, 0x16554, 0x056a0, 0x09ad0, 0x055d2,

0x04ae0, 0x0a5b6, 0x0a4d0, 0x0d250, 0x1d255, 0x0b540, 0x0d6a0,

...... };

每一项以二进制的格式表示，共有20位，代表某一年的农历信息。1-4位表示当年有无闰年，有的话，为闰月的月份，没有的话，为0。5-16位表示除了闰月外的正常月份是大月还是小月，1为30天，0为29天。17-20位表示表示闰月是大月还是小月，仅当存在闰月的情况下有意义。

例如：1982年的数据是0x0a974(0000 1010 1010 0111 0100)

表示1982年的4月为闰月，即有第二个4月，且是闰小月。从1月到13月的天数依次为：30、29、30、29、29(闰月)、30、29、30、29、29、30、30、30。

利用这份数据，我们可以获取某一年的总天数，某个月的天数以及闰月的天数。

Lunar类有一个构造函数lunar(Date cal)，它的作用是求出cal是当月的第几天。所以我们只需要用某一天的date来创建一个lunnar实例，就可以获得这一天的农历日期。

#### 闹钟提醒

闹钟弹出提醒的实现封装在ClockAlertActivity.java类中，这个类继承activity，是一个activity子类，所以需要在AndroidManifest.xml里面声明：

<activity

android:name=*".activity.ClockAlertActivity"*

android:launchMode=*"singleInstance"*

android:theme=*"@style/Theme.clock"*>

</activity>

另外，闹钟的启动需要用到android的Broadcast机制，它是一种广泛运用的在应用程序之间传输信息的机制，其中BroadcastReceiver是对发送出来的 Broadcast进行过滤接受并响应的一类组件。当闹钟的时间到时，系统就会向应用程序发出广播，所以需要建一个继承自BroadcastReceiver的广播接收器ClockReceiver，来接收系统发出的广播。并且该广播接收器需要在AndroidManifest.xml里面注册声明：

<receiver android:name=*".activity.ClockReceiver"*

android:process=*":remote"*/>

当广播接收器接收到广播时，就会调用ClockAlertActivity类，该类负责获取记事的内容以及闹钟信息，然后弹出提醒框，同时响铃提醒用户。

弹出的提醒框中，包括三个button以及记事的内容，这三个button分别是“go it”“delete”和“I known”，其中“go it”和“delete”绑定了点击事件，选择“go it”会调用NoteEditActivity，进入相应的记事页面，显示完整的记事内容。

选择“delete”，则调用deleteClock方法，删除该闹钟。删除时首先获取系统的AlarmManager进程，将该进程取消，然后再从数据库中将闹钟删除。

最后，当提醒结束后，弹框消失，响铃停止，程序就会判断闹钟是否已经完全结束，如果是，则调用deleteClock删除闹钟，反之，则调用mClockModel.addAlertCount()，将闹钟已经弹出的次数加1。

### 分享模块

分享模块分为分享出去和接收分享，它们的实现利用的都是Intent类，通过设置或者获取intent的action和type，可以区分是何种分享。其中分享出去实现了图文分享和音频分享，而接收分享，现在暂不支持接收音频类的分享。

#### 分享出去

1. **图文分享**

图文分享包括纯文本分享、单图文分享、多图文分享，通过设置intent的type和action来区分。

当用户选择了分享时，首先创建一个intent实例，action为ACTION\_SEND；然后判断分享的内容是否含有图片，如果图片为空，则说明是纯文字分享，如果含有一张图片，则说明此分享是单图文分享，如果含有多张图片，则为多图文分享。

对于几种不同分享类型的action和type值的设置如下：

纯文字分享：action为ACTION\_SEND，type为text/plain

单图文分享：action为ACTION\_SEND，type为image/\*

多图文分享：action为ACTION\_SEND\_MULTIPLE，type为images/\*

设置了intent的action和type后，需要为intent添加数据，数据主要分为两种：文本数据和图片数据，而添加数据用的是intent的putExtra方法。其中Intent.EXTRA\_TEXT承载的是文本数据， Intent.EXTRA\_STREAM承载的是图片数据。最后利用startActivity启动分享activity。

1. **音频分享**

音频分享跟图文分享的原理是一样的，都是利用intent，音频分享的intent action为ACTION\_SEND，type为audio/\*。当一个记事中添加了音频后，就会有一个音频的图片，如果用户长按这个音频时，就可以分享此音频和记事的文本内容。文本数据赋值给Intent.EXTRA\_TEXT,音频数据赋值给Intent.EXTRA\_STREAM。

#### 接收分享

接收分享部分的实现单独作为一个类ShareReceiver，这个类继承自activity，只有在AndroidManifest.xml里面声明后才可以用，除了声明activity外，还需要声明intent-filter，用来对分享的类型进行过滤。其中intent-filter声明需要包含action，data，category。其声明如下：

<activity android:name=*".activity.ShareReceiver"* >

<intent-filter>

<action android:name=*"android.intent.action.SEND"* />

<action android:name=*"android.intent.action.SEND\_MULTIPLE"*/>

<category android:name=*"android.intent.category.DEFAULT"* />

<data android:mimeType=*"text/plain"* />

<data android:mimeType=*"image/\*"* />

</intent-filter>

</activity>

Action和data都有两种类型，分别用来接收文字和图片的分享，其他类型的分享都会被过滤掉。当有分享进来时，就会启动ShareReceiver这个activity，调用onCreate函数，onCreate函数获取传递进来的Intent以及Intent的action和type属性，然后判断Intent的action和type类型，确定是文字分享还是图片分享，再分别调用相应的函数进行处理。

纯文字分享的处理函数是handleSendText，只含有单张图片的分享处理函数时handleSendImage，含有多个图片的分享处理函数是handleSendMultipleImages，它们的参数都为Intent。每一种分享，都会新建一个记事，调用NoteEditActivity，数据绑定在Bundle实例中传递。由于新建记事有很多不同的情况，所以用ID来区别是哪一种情况，比如ID=-1是用户自己新建的记事，ID=-2是分享进来的文字和单图片新建的记事，ID=-3是分享进来的多图片新建的记事。

新建记事时，对于分享中的图片处理与图片记事中对图片的处理方式基本相同，都是利用bitmap，调用showBitmapImg函数。

## 数据库实现

在此次的开发中，数据存储利用的是SQLite数据库，为了方便管理以及代码的简洁规范，所有的数据库操作都封装在DBHelper.java类中，包括数据表的创建以及对数据的增删查改等主要操作。另外由于对数据库的操作需要用到SQLite的一些方法，所以在DBHelper类的开头需要引入SQLiteDatabase，SQLiteOpenHelper两个包。

### 数据表创建

此次的记事本应用共有两个数据表：

notes（ID,CLOCK\_ID,CREATE\_DATE,MODIFY\_DATE,CONTENT,AUDIO,SEQUENCE）

alerts（ID,TIME,ALERT\_INTERVAL,ALERT\_TIMES,ALERT\_COUNT）

它们分别用来存储单个记事的信息和闹钟的信息，其创建字符串如下：

notes表的创建字符串：

**private** **static** **final** String *CREATE\_NOTES\_TABLE\_SQL* = "CREATE TABLE "

+ TABLE.*NOTES*

+ " ("

+ NoteItemModel.*ID*

+ " INTEGER PRIMARY KEY,"

+ NoteItemModel.*CLOCK\_ID* + " INTEGER NOT NULL DEFAULT 0,"

+ NoteItemModel.*CREATE\_DATE* + " INTEGER NOT NULL DEFAULT 0,"

+ NoteItemModel.*MODIFY\_DATE* + " INTEGER NOT NULL DEFAULT 0,"

+ NoteItemModel.*CONTENT* + " TEXT　NOT NULL DEFAULT '',"

+ NoteItemModel.*AUDIO* + " TEXT　NOT NULL DEFAULT '',"

+ NoteItemModel.*SEQUENCE* + " INTEGER NOT NULL DEFAULT 0" + ")";

Alerts表的创建字符串：

**private** **static** **final** String *CREATE\_ALERTS\_TABLE\_SQL* = "CREATE TABLE "

+ TABLE.*ALERTS* + " (" + ClockModel.*ID* + " INTEGER PRIMARY KEY,"

+ ClockModel.*TIME* + " INTEGER NOT NULL DEFAULT 0,"

+ ClockModel.*ALERT\_INTERVAL* + " INTEGER　NOT NULL DEFAULT 0,"

+ ClockModel.*ALERT\_TIMES* + " INTEGER　NOT NULL DEFAULT 0,"

+ ClockModel.*ALERT\_COUNT* + " INTEGER　NOT NULL DEFAULT 0" + ")";

### 数据库操作

DAHelper类中封装的基本操作主要有“插入”“删除”“查找”“更新”，这里用到的一个重要的类是ContentValues类，它是数据库中用到的一种存储机制，主要用来存储基本类型的数据，如string之类的。

这些函数的封装基本类似，首先都必须确定数据库是否打开以及是否准备好，然后调用SQLite本身的数据库API来执行插入删除等操作。例如插入函数如下：

**public** **long** insert(String tableName, ContentValues values) {

**if** (**this**.db.isOpen() && **this**.db.isReadOnly()) {

db.close();

db = **this**.getWritableDatabase();

}

**long** id = db.insert(tableName, **null**, values);

**return** id;

}

# 系统测试及演示

## 概述

本文设计的记事本名为YYnote，记事本的logo如图6.1



图 6. 1 记事本logo

记事本的主页面如图6.2

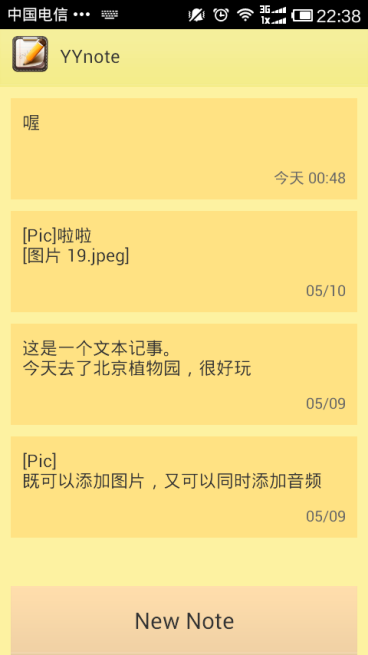


图 6. 2记事本主页面

## 添加文本

选择“New Note”，进入编辑记事页面，输入文本，如图6.3。

输入完文本后，如果直接选择物理返回键，则会弹出提示窗口，如图6.4，每次编辑记事时，如果选择物理返回键，就会检测记事的内容是否发生变化，发生变化则会弹出提示框。

保存记事后，返回主页面，主页面列表就会出现新建的记事，如图6.5。

经过测试，添加文本保存等功能基本没有问题。

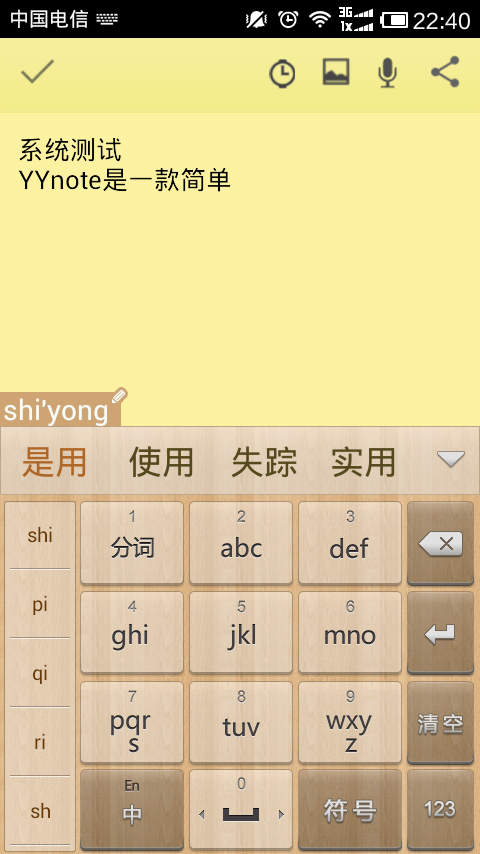
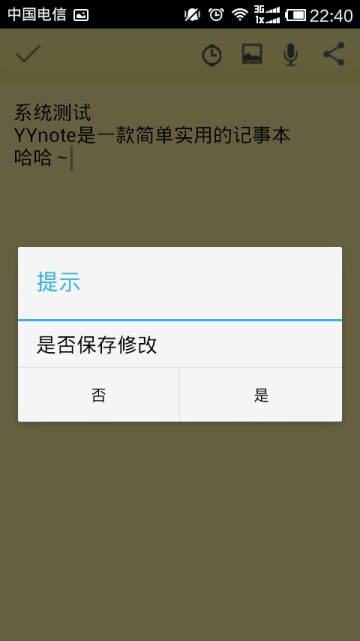
 

图 6. 3 添加文本 图 6. 4 提示是否保存

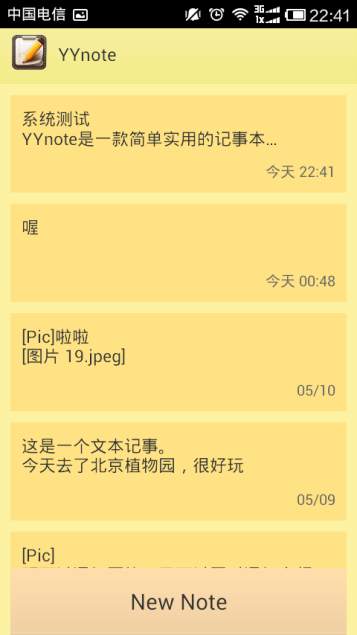


图 6. 5 新建记事后主页面变化

## 添加图片

选择导航栏上的图片icon，弹出选择框，如图6.6；从相册中选择添加一张图片，结果如图6.7，同样，如果拍照也会得到同样的效果；可以继续添加多张图片，效果如图6.8，说明添加图片功能基本没有问题。

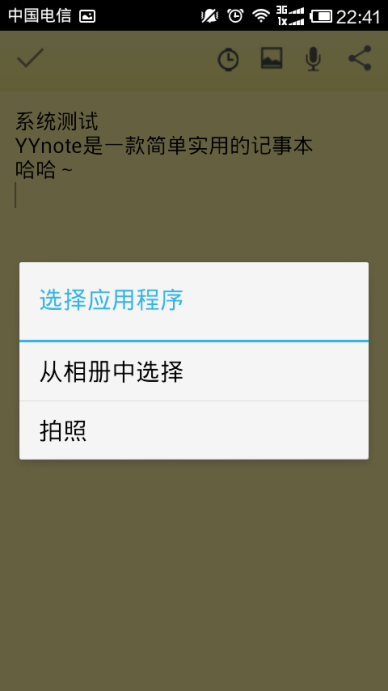
 

图 6. 6 添加图片选择框 图 6. 7 从相册中选择

图 6. 8 添加多张图片 图 6. 9 添加多张图片

## 添加音频

选择导航栏上的音频icon，弹出录制音频界面，如图6.10；选择start开始录制，录制完成后，选择ok，就会在记事的头部创建一个音频播放icon，如图6.11；点击播放icon，则播放刚才录制的音频，播放中的界面如图6.12和6.13，添加音频功能基本没有问题。

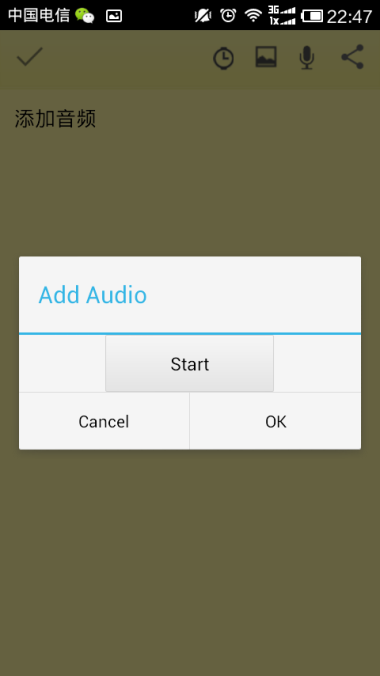
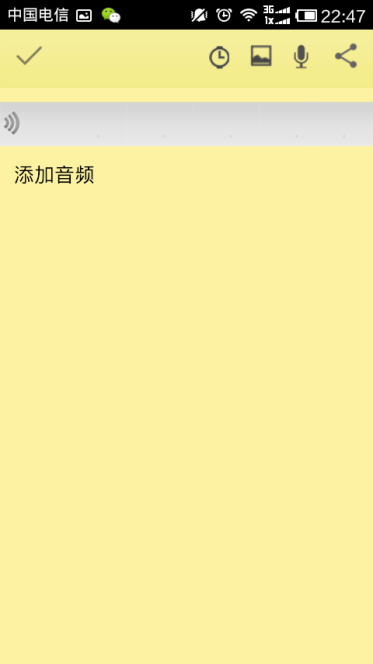
 

图 6. 10 音频录制页面 图 6. 11录制完成页面

图 6. 12 音频播放界面 图 6. 13 音频播放界面

## 添加闹钟

选择导航栏上的闹钟icon，弹出闹钟设置页面，如图6.14；设置闹钟的日期、时间、间隔时间、总次数等参数，如图6.15、6.16、6.17和6.18；设置了闹钟后，参数如图6.19，选择ok保存闹钟，这时导航栏上出现闹钟的日期，日期有两种形式，如果闹钟不是当天的则只显示日期，如果闹钟是当天的，则显示“今天+具体时间”。添加闹钟后主页面相应的记事item会出现一个闹钟的图标，表示此记事有闹钟，如图6.20和6.21。经测试，闹钟基本功能没有问题。

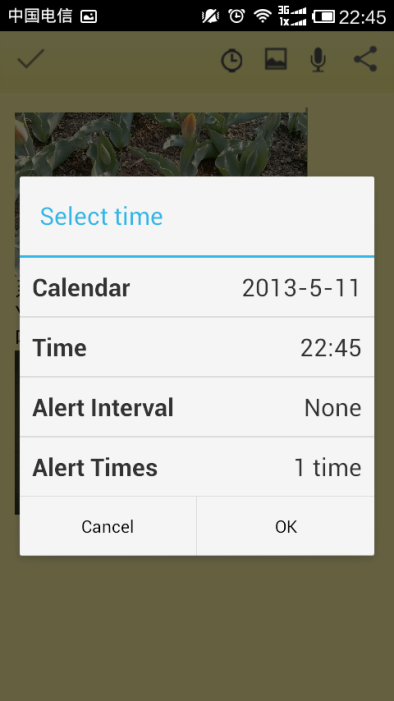
 

图 6. 14 设置闹钟 图 6. 15 选择日历

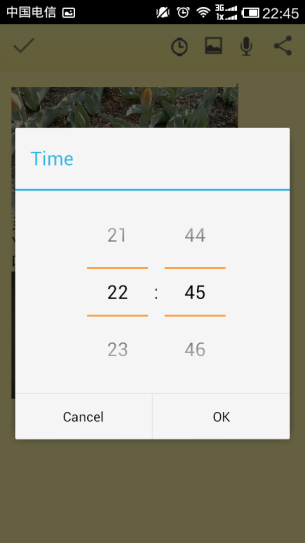
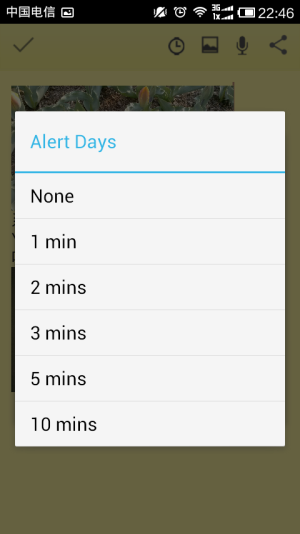
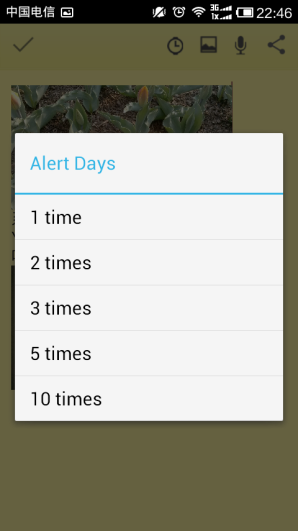
  

图 6. 16设置日期 图 6. 17 设置间隔时间 图 6. 18 设置总次数

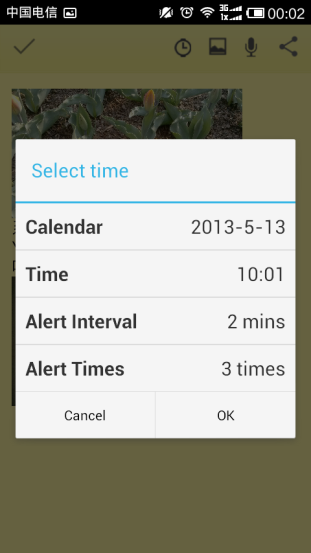
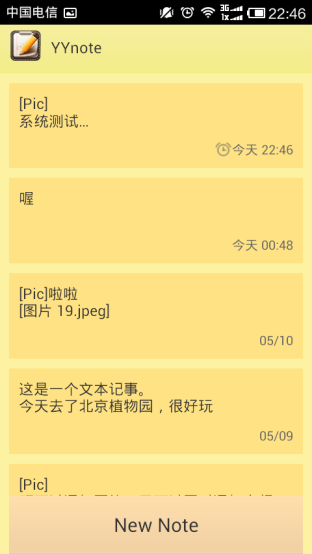
  

图 6. 19参数设置完成 图 6. 20 保存闹钟后页面 图6. 21 保存闹钟后主页面

## 分享

选择导航栏的分享icon，出现分享应用列表，如图6.22；选择分享到印象笔记，分享的结果如图6.23；接收从小米便签分享进来的记事如图6.24和6.25，说明分享功能基本没有问题。

图 6. 22分享应用列表 图 6. 23 分享到印象笔记

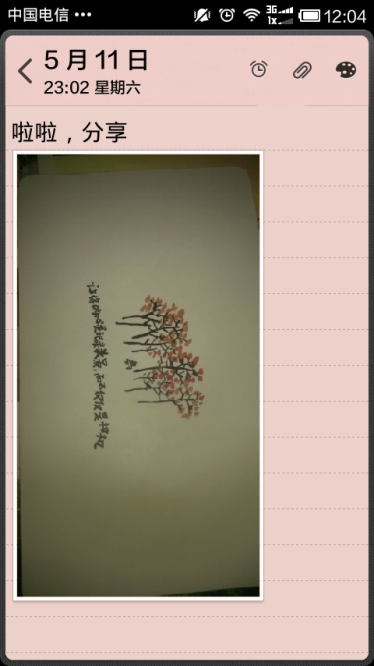
 

图 6. 24 小米标签分享 图 6. 25 接收分享

经过对每个功能的测试，主要功能基本没有问题，不过在细节上，有一点问题，比如在闹钟提醒弹出后，直接删除闹钟，在主页面相应的记事item上，仍会有闹钟图标，刷新过后才会消失。

# 总结

经过这几个月的努力学习与研究，我最终实现了本次毕业设计的任务—基于android的记事本应用，这款记事本应用取名为YYnote，是一款简单实用的记事本，不仅可以添加文本记事，还有支持添加多张图片，录制音频，同时还可以设置闹钟提醒，并且可以提醒多次，在界面上也不失美观。总体上来说，对于本次开发的记事本很满意。

但是仍有不足之处，接收分享没有只能接收图片文字的分享，没有实现接收音频分享的功能；另外，在用户体验也需要提升，没有像小米便签或者evernote那样好的用户体验，界面的设计也需要进一步提升；还有在音频方面，不能像添加图片那样添加多个音频，一次只能添加一个音频，新的音频会覆盖原来的音频，这部分需要改进。

通过这次的android开发，我对android有了一定的了解，android开发相对来说是比较简单的，它有一套完善的开发工具，成熟的API，以及网上很多的实例教程，对于初学者来说，比较容易入门。在开发过程中，我也遇到了很多难题以及很多令人头疼的bug，比如闹钟无法多次弹出的问题或者闹钟无法删除死循环的问题等，每次遇到问题时，一方面我会从网上寻找答案，另一方面会加上自己的思考来解决问题，有时候技术的问题往往能够通过搜索找到答案，但是涉及逻辑方面的问题时，就需要自己多思考，才能设计出好的解决方案。在解决问题的过程中，我不仅学到了很多新的知识，而且对我独立思考和解决问题能力的提升有很大的帮助。

总之，通过这几个月的努力，我不仅对android开发有了深入的了解，而且对于java的使用的也逐渐熟练，对我以后工作学习有很大的帮助，

最后，由于本人学识有限，加之时间仓促，文中不免有错误和待改进之处，真诚欢迎各位老师提出宝贵意见。

# 致谢

在论文完成之际，我首先要向赵贻竹老师表示最真挚的谢意，赵老师的指导给了我很大的帮助。在论文写作期间，经过自己不断的努力学习以及赵老师的耐心指导，本次毕业设计已经基本完成。赵老师还给我提出了很多好建议，她的指导使我受益匪浅。在本次毕业设计中，赵老师对我严格要求，并及时督促我完成毕业设计的任务，使得我能够顺利的完成此次毕业设计，非常感谢赵老师在毕业设计过程中对我的帮助。

同时，我要感谢自己大学四年的室友，她们不仅在此次的毕业设计中给了我很大的帮助和关心，在大学这四年中，也对给予了我很多的关心和照顾，有了她们的支持，我才能有今天的成就，在此特别感谢我的室友们。

当然，还要感谢学院提供了这次机会以及各方面的支持，有了好的环境和支持我才能顺利完成本次毕业设计的任务。

最后，我要感谢我的家人，尤其是我的父母，有了你们这么多年来对我的亲切呵护，我才能够让自己一点一点地改善，一点一点地进步，并最终取得今天的成绩，我从内心深处感谢你们的养育之恩！

毕业在即，感慨万千。回首这段时间，所有的一切都历历在目，那些曾经一起相处过、帮助过我的老师们、同学们，你们的帮助让我学会了很多，感谢你们的无私帮助，有了你们的关心、支持和帮助，我的毕业设计才得以顺利完成。再次衷心感谢你们！

# 参考文献

1. E2ECloud工作室，深入浅出GoogleAndroid，人民邮电出版社，2009
2. [扶松柏](http://search.dangdang.com/book/search_pub.php?category=01&key2=%B7%F6%CB%C9%B0%D8&order=sort_xtime_desc)， Android开发从入门到精通，北京[希望电子出版社](http://search.dangdang.com/book/search_pub.php?category=01&key3=%B1%B1%BE%A9%CF%A3%CD%FB%B5%E7%D7%D3%B3%F6%B0%E6%C9%E7&order=sort_xtime_desc)，2012
3. 李刚，疯狂Android讲义，电子工业出版社，2011
4. [伯内特Ed Burnette](http://www.360buy.com/writer/ï¼ç¾ï¼ä¼¯åç¹Ed%20Burnette_1.html) ，Android基础教程、第3版，人民邮电出版社，2011
5. [杨丰盛](http://book.douban.com/search/æ¨ä¸°ç)，Android应用开发揭秘，机械工业出版社 2010
6. 邓凡平，深入理解Android，机械工业出版社，2012
7. 杨丰盛，Android技术内幕:系统卷，机械工业出版社，2011
8. 余志龙等，Google Android SDK开发范例大全，人民邮电出版社，2009
9. [Bruce Eckel](http://book.douban.com/search/%5bç¾%5d%20Bruce%20Eckel) ，java编程思想，机械工业出版社，2002
10. Mark L. Murphy, Beginning Android, 人民邮电出版社，2010
11. Frank Ableson. CharlieCollins, Unlocking Android, Manning Publications, 2009
12. Ed Burnette, Hello.Android, Pragmatic Bookshelf ,2010
13. Dave MacLean. Dave MacLean , Pro Android 2.Apress ,2010
14. Reto Meier, Professional Android 2 Application Development, Wrox, 2010
15. MR Mark L. Murphy, The Busy Coder's Guide to Advanced Android Development, Commonsware, LLC,2011.7
16. Jonathan Simon, Head First Android Development, O'Reilly Media,2012.10