**学校代码：10200　　 　　　　　　学号： 2013012645**

****

本科毕业论文



**日程管理app的设计与实现**

**The design and implementation of the calendar management app**

学生姓名：温立涛

指导教师：王伟

所在学院：信息科学与技术学院

所学专业：软件工程

东北师范大学

2017 年 5 月

**独　创　性　声　明**

本人郑重声明：所提交的毕业论文是本人在导师指导下独立进行研究工作所取得的成果。据我所知，除了特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果。对本人的研究作出重要贡献的个人和集体，均已在文中做了明确的说明。本声明的法律结果由本人承担。

论文作者签名：　　温立涛　　　　日期：　　2017年5月

**摘 要**

随着生活节奏的逐步加快，人们经常会被各种各样的日程搞得手忙脚乱，比如说个人的日常工作、朋友的聚会、家庭中的各种琐事等等。如果这些繁杂的日程能在特定的时间得到及时醒目的提醒，那么就会减轻人们的生活压力。为了能尽可能的减少这些繁杂的事情给人们带来的负担，为了使人们拥有更有效率的工作和更轻松舒适的生活，就需要有专门的日程管理系统来安排这些事务，很多人开发了专门的日程管理系统。日程管理系统的主要功能是完成个人一般日程和重要日程安排并提供提醒和查看的功能。

如今伴随着信息技术的高速发展，手机已经成为现代人们生活所必不可少的物品，而且基本上所有人都会将手机随身携带，所以说将手机作为日程管理系统的开发平台可以说是一个很好的选择。此外，手机中的安卓系统基本已经成为最大众化的手机操作系统，而且安卓开发也已经成为一种热门的技术，综合考虑之后，本文的系统选择基于安卓的手机日程管理系统作为研发的软件系统。

该系统的特色是以简单的操作为用户提供最需要的功能，而减少一些杂乱次要的功能，主要的功能包括：

1. 基本日程活动的增删改查。
2. 通过日历浏览查看日程的信息。
3. 用户根据个人的需要添加各种类型的日程，便于对日程进行管理。

本系统是基于安卓开发平台，使用了Java语言以及SQLite数据库来对日程管理系统进行研究和开发的。

**关键词：**日程管理 Java 安卓

**Abstract**

With the quickening pace of life, people are often confused by a variety of schedules, such as personal daily work, a friend's party, family chores, etc.. If these complicated schedules can be timely and eye-catching reminder at a specific time, it will reduce the pressure on people's lives. In order to minimize these things to bring the burden, in order to make people have to work more efficiently and more comfortable life, you need to have the schedule management system dedicated to arrange these matters, a lot of people developed a special schedule management system. The main function of the schedule management system is to complete the general schedule and important schedule and provide the function of reminding and viewing.

Now with the rapid development of information technology, mobile phone has become essential to modern people's life items, and basically everyone will carry a mobile phone, so that the mobile phone as schedule management system development platform can be said to be a good choice. In addition, the basic system of Android mobile phone has become the most popular mobile phone operating system, and Android has become a popular technology, after comprehensive consideration, the system choice of Android mobile phone management system based on software system development as.

The system is characterized by a simple operation to provide users with the most needed functionality, and reduce the number of miscellaneous minor functions, the main features include:

(1) Basic operations for schedule activities:add,delete,change,check.

(2) through the calendar view to view the specific information of the calendar, the calendar can also highlight the date of the calendar.

(3) according to the needs of individual users, different types of schedules are designed to facilitate the management of the schedule.

This system is based on the Android development platform, using the Java language and SQLite database to study and develop the schedule management system.

**Keywords:** schedule management Java Android

**目 录**

**[摘 要](#_Toc16771)** [I](#_Toc16771)

**[Abstract](#_Toc11224)** [II](#_Toc11224)

**[1. 绪论](#_Toc6181)** [1](#_Toc6181)

**[1.1选题的目的及意义](#_Toc23694)** [1](#_Toc23694)

**[1.2研究现状](#_Toc18631)** [2](#_Toc18631)

**[1.3软件要实现的基本目标](#_Toc2159)** [2](#_Toc2159)

**[1.4论文结构介绍](#_Toc27248)** [3](#_Toc27248)

**[2. 技术与原理](#_Toc6068)** [4](#_Toc6068)

**[2.1系统开发环境](#_Toc13183)** [4](#_Toc13183)

**[2.2系统开发软件介绍](#_Toc23819)** [4](#_Toc23819)

**[2.3开发语言介绍](#_Toc29326)** [5](#_Toc29326)

**[3. 系统需求分析](#_Toc8384)** [10](#_Toc8384)

**[3.1系统目标](#_Toc6764)** [10](#_Toc6764)

**[3.2系统功能需求](#_Toc28492)** [10](#_Toc28492)

**[3.3系统性能需求](#_Toc32497)** [10](#_Toc32497)

**[3.4系统运行环境](#_Toc26983)** [11](#_Toc26983)

**[4. 系统设计](#_Toc20804)** [12](#_Toc20804)

**[4.1系统总体设计描述](#_Toc2467)** [12](#_Toc2467)

**[4.2系统总体设计](#_Toc8080)** [12](#_Toc8080)

**[4.3数据库设计](#_Toc13272)** [13](#_Toc13272)

**[5. 系统实现](#_Toc31557)** [15](#_Toc31557)

**[5.1主界面的实现](#_Toc17636)** [15](#_Toc17636)

**[5.2日历模块的实现](#_Toc16965)** [19](#_Toc16965)

**[5.3日程模块的实现](#_Toc19860)** [22](#_Toc19860)

**[5.4设置模块的实现](#_Toc10942)** [32](#_Toc10942)

**[5.5数据库的实现](#_Toc14769)** [33](#_Toc14769)

**[6. 总结与展望](#_Toc28276)** [37](#_Toc28276)

**[参考文献](#_Toc27776)** [38](#_Toc27776)

1. **绪论**

**1.1选题的目的及意义**

1.1.1时代背景

进入21世纪以来，无所不在的计算逐渐打破传统的计算机概念：使用更简单方便、携带更容易、功能更多样化的各种具有计算机功能的设备大量出现，并且在需要时，已经可以随时通过运营商和无线方式连接到移动数据通信网络，以便高效地交换信息和获得各种服务。这种设备人们称之为无线设备。移动设备最根本的特征是移动性，即能够利用无线电波、微波或红外线等传输介质允许通信中的设备在一定范围内自由移动，其位置不受束缚，从而实现通信双方至少有一方在移动中进行信息传输和交换的目的。移动设备从产生到现在的历史并不长，然而其发展速度却远远超出人们的预料。现如今，移动设备大行其道，全球有超过十亿部移动电话正在被使用，并且这个数目正在逐渐增加。在除美国之外的每个发达国家，拥有手机的人数比拥有计算机的人数更多。

1.1.2用户需求

日程管理对于我们来说已经不是一个陌生的词汇，无论你是学生，教师还是企业白领或是公务员，每天的生活都充斥着大量的学习或工作上的任务和各种各样的琐事。当大量繁琐的日程堆在你面前时，往往会让你不知所措，甚至会在慌慌忙忙中去遗漏一些重要的事情。这样不仅大大降低了人们的做事效率，也会影响人们的心情从而降低生活质量。如何能做到有条不紊的完成每天的日程据渐渐成为人们关心的话题。这时候一款能为我们工作，学习，生活提供方便，提高我们做事情效率的日程管理系统就应运而生了。虽然，现在各种应用商店中日程管理类的应用已经有了一些，但是在亲身体验了五花八门的日程管理应用之后，我感觉作为用户更需要的更希望使用的是一款界面简介操作方便的日程管理系统。因为面对繁重的工作和学习时，作为用户往往追求的是系统的简洁性和高效性，相反那些看似功能多样界面华丽的应用往往给用户体验带来了很多累赘和负担，所以一款操作简易的日程管理系统往往能给用户带来更好的体验。综上所述，考虑到现在手机系统的发展现状，Android平台的开放性以及人们对一款简洁方便的日程管理系统的需求，基于安卓平台研究和开发一款简洁友好的日程管理手机应用程序便具有了较大的意义和可行性。

日程管理是一款手机日程管理软件。由于生活节奏不断提高，人们越来越意识到时间日程管理的重要性，本软件的核心理念20/80时间管理法则，即20%的时间做时间管理，可以影响并提高80%成效。日程管理软件可以有效改善工作管理流程，让人们高效管理所有事务，获取更加充沛的精力，管理变得轻松自如，事半功倍，使人们能够更好的利用这个软件来管理的生活和事业，帮助安排日常工作，合理计划生活，可以实现人生蓝图的规划及愿景目标的规划，让生活合理高效，每天有计划的生活工作。行动系统引导每周、每天提前设定自己的具体行动计划，并及时进行总结和改进，评估自己的行动效率，时间系统会加以自动提示。

**1.2研究现状**

1.2.1安卓系统

Android是基于Linux内核的软件平台和操作系统，早期由Google开发（在华注册商标名为“安卓”），后由开放手机联盟（OpenHandsetAlliance）开发。它使用软件堆栈（软件堆栈，也称为软件堆栈）架构，分为三个部分。 下层基于Linux内核工作，仅提供基本功能; 其他应用程序由公司本身开发，Java作为程序的一部分。 另外为了推广这项技术，谷歌等数十家手机公司建立了开放手机联盟。 在传出Android之前，传言是传出Google手机或gPhone。 大部分传闻，Google正在开发自己的手机产品，而不是一套软件平台。到了2010年1月，Google开始发表自家品牌手机电话的Nexus One。同年，三星在北美的四个主要运营商发布了它的第一款安卓智能手机，Galaxy S，被认为是安卓阵营的后来者。在三星的带领下，HTC也推出了安卓手机比如 Thunderbolt。摩托罗拉推出了Atrix。从此安卓系统被各大手机生产商使用。

目前国内的 Android开发还是主要以应用开发为主，主要分成 3类：对于开发应用程序，开发通用应用程序（放AndroidMarket或其他AppMarket销售）和游戏开发（放AndroidMarket或其他AppMarket销售）。 第一类开发商一般都在大型公司，这些公司主要为自有品牌或其他品牌设计手机或平板电脑的整体方案。 除了根据需要定制系统，还有更多的工作是为这些系统编写自定义应用程序。 第二类开发商一般都是在创业公司或独立开发商，他们的利润主要有两种方式：对外国公司进行外包开发，或通过Google的手机广告通过广告点击进入。 而理论上通过付费下载的形式获利，现在国内几乎没有成功。 第三种开发者类似于第二类开发者。主要差别是其主要从事游戏开发和销售。

1.2.2问题研究现状

关于日程管理系统类似的应用也不少，但往往都因为过度的追求功能的多样性和华丽的效果而使得操作复杂、界面混乱、系统的主功能弱化。实际上日程管理系统最主要的目的还是督促用户更好的完成每天的日程，让用户用最简单的操作来达到最想实现的目的。所以，在这次毕业设计初期我对比了很多类似的日程管理应用并逐步对它们的用户体验做了调查，并把这些调研结果作为自己设计日程管理系统的参考依据。最终确定了以日历的形式来展示和管理日程，这样使得用户自己添加的日程清晰明了并且方便查看和修改。综上所述，本系统虽然不是一个聚集了多重功能的应用，也没有过于绚丽的设计，但正因为这样的优点才让这个应用看起来简单友好，目标明确，简单易上手。

**1.3软件要实现的基本目标**

本软件系统能减少使用很多不必要的资源，不用像最初那样用冗余的纸张式的管理和记录，也不会像其他的日程管理系统那样过度的追求花样的功能。以简洁高效的特性使人们的日常工作和生活能有一个合理的安排，从而不会因为慌慌张张的处理每天的事情然而却错过了一些很重要的事情。

软件采用人机交互的方式，需要界面美观友好，操作灵活、方便、富有个性化。能实现让用户对日程管理系统的日程进行保存、更新、查询等操作，需要良好的数据库结构的设计，方便用户访问系统中所需要的一些数据。

软件要实现的基本功能有以下几个方面：日历的查看功能；日历上日期对应日程的查看功能；日程的总览功能；日程的增加、删除、修改功能；感想与总结；闹钟的设置。

**1.4论文结构介绍**

第一章：介绍的是日程管理系统的背景意义和要实现的基本目标。主要就Android应用的大众化、日程繁杂，分析日程管理能遇到的问题和用户体验方面提供解决方法。

第二章：介绍的是开发系统的过程中需要用的开发环境、开发工具和开发语言。该系统是基于安卓的基础知识和java语言，在windows上使用android studio来开发的。

第三章：介绍系统的需求分析。描述系统要实现的基本目标，分析基本功能和界面的需求。

第四章：介绍系统的总体设计。介绍了系统总体架构的设计和和实现、具体功能模块的设计和实现。

第五章：介绍系统的具体模块实现。介绍了系统开发环境的搭建和系统核心模块的编码和实现。

第六章：介绍对完成整个开发项目的感想和项目存在的不足，从而明确以后开发软件时需要多关注的地方。

1. **技术与原理**

**2.1系统开发环境**

Android的上层应用程序是用Java语言开发，同时还需要Dalvik虚拟机的支持，所以，Google推荐使用一直比较主流的Java集成开发环境Eclipse来进行安卓的开发工作。然而，光有Eclipse还不够，因为是使用Java语言进行开发，还应该有由SUN公司提供的JDK。另外Android应用开发和Java开发有很大的不同，所以还需要提供Google Android SDK。 同时，还要开发升级或Android开发更改，有必要安装Eclipse ADT，是下载Eclipse工具升级的工具。

简而言之，如果你想进行android应用程序的开发，需要以下必要的软件来搭建android开发的环境：

1. Eclipse
2. JDK
3. Android SDK
4. ADT(Android Development Tools)

**2.2系统开发软件介绍**

2.2.1 Eclipse

Eclipse 是一个开放[源代码](http://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81/3969" \t "_blank)的、基于[Java](http://baike.baidu.com/item/Java/85979" \t "_blank)的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，包括Java[开发工具](http://baike.baidu.com/item/%E5%BC%80%E5%8F%91%E5%B7%A5%E5%85%B7" \t "_blank)。

虽然大多数用户很乐于将 Eclipse 当作 Java [集成开发环境](http://baike.baidu.com/item/%E9%9B%86%E6%88%90%E5%BC%80%E5%8F%91%E7%8E%AF%E5%A2%83" \t "_blank)（[IDE](http://baike.baidu.com/item/IDE" \t "_blank)）来使用，但 Eclipse 的目标却不仅限于此。Eclipse 还包括插件开发环境（Plug-in Development Environment，PDE），这个组件主要针对希望扩展 Eclipse 的软件开发人员，因为它允许他们构建与 Eclipse 环境无缝集成的工具。由于 Eclipse 中的每样东西都是插件，对于给 Eclipse 提供插件，以及给用户提供一致和统一的集成开发环境而言，所有工具开发人员都具有同等的发挥场所。这种平等和一致性并不仅限于 [Java](http://baike.baidu.com/item/Java" \t "_blank) [开发工具](http://baike.baidu.com/item/%E5%BC%80%E5%8F%91%E5%B7%A5%E5%85%B7" \t "_blank)。尽管 Eclipse 是使用Java语言开发的，但它的用途并不限于 Java 语言；例如，支持诸如[C/C++](http://baike.baidu.com/item/C%2FC%2B%2B" \t "_blank)、[COBOL](http://baike.baidu.com/item/COBOL" \t "_blank)、[PHP](http://baike.baidu.com/item/PHP" \t "_blank)、Android等编程语言的插件已经可用，或预计将会推出。Eclipse 框架还可用来作为与软件开发无关的其他应用程序类型的基础，比如[内容管理系统](http://baike.baidu.com/item/%E5%86%85%E5%AE%B9%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F" \t "_blank)。

2.2.2 JDK

JDK是 [Java](http://baike.baidu.com/item/Java/85979" \t "http://baike.baidu.com/_blank) 语言的[软件开发工具包](http://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91%E5%B7%A5%E5%85%B7%E5%8C%85" \t "http://baike.baidu.com/_blank)，主要用于移动设备、嵌入式设备上的java应用程序。JDK是整个java开发的核心，它包含了JAVA的运行环境，JAVA工具和JAVA基础的类库。而专门在x86运行的平台Jrocket在服务端的运行效率也要比Sun JDK好很多。从SUN的JDK5.0开始，提供了泛型等一些非常实用的功能，其版本也不断更新，运行效率得到了很大的提升。

2.2.3 Android SDK

SDK：(software development kit)软件开发工具包。被广大软件开发工程师们用于为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件的一些常用开发工具的集合。因此，Android SDK指的是Android专属的软件开发工具包。

2.2.4 ADT

目前Android开发所用的开发工具是Eclipse，在Eclipse编译IDE环境中，安装ADT，为Android开发提供开发工具的升级或者变更。可以简单理解为在Eclipse下开发工具包。

**2.3开发语言介绍**

2.3.1 Android简介

Android是一种基于Linux的自由及开放源代码的操作系统，主要适用于移动设备，如智能手机和平板电脑。谷歌公司与开业电话联盟的领导人和开发。Android操作系统开发Rubin最初Andy，主要是支持手机。在2005年8月，根据谷歌的收购而出资。2007年的一月份，谷歌和84家硬体厂商、供应商和电信代理公司开发开放系统改进的Android手机联盟。之后在谷歌Apache开放源码许可证的执照，Android的源代码发表了。

Android一词的本义指“[机器人](http://baike.baidu.com/item/%E6%9C%BA%E5%99%A8%E4%BA%BA" \t "http://baike.baidu.com/item/android/_blank)”，同时也是[Google](http://baike.baidu.com/item/Google" \t "http://baike.baidu.com/item/android/_blank)于2007年11月5日宣布的基于[Linux](http://baike.baidu.com/item/Linux" \t "http://baike.baidu.com/item/android/_blank)平台的[开源](http://baike.baidu.com/item/%E5%BC%80%E6%BA%90" \t "http://baike.baidu.com/item/android/_blank)手机操作系统的名称，该平台由[操作系统](http://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F" \t "http://baike.baidu.com/item/android/_blank)、中间件、[用户界面](http://baike.baidu.com/item/%E7%94%A8%E6%88%B7%E7%95%8C%E9%9D%A2" \t "http://baike.baidu.com/item/android/_blank)和[应用软件](http://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E8%BD%AF%E4%BB%B6" \t "http://baike.baidu.com/item/android/_blank)组成。

2.3.2 Android特性

Android平台的特性：

（1）通信工具  
     手机最基本的功能是通信功能，所以使用运营商提供通话网络进行语音通话是Android平台最基本的功能。 除了基本的语音通话功能外，Android平台还具有短信功能，通常手机具有个人信息系统管理功能（如电话簿等）。

（2）网络工具  
     随着数字业务是的逐渐趋于普遍化，移动电话通常被当作网络工具来使用，即移动电话可以完成电脑的部分功能。而Android平台在网络方面的主要能实现的功能包括浏览器、IM（即时信息）、邮件等等。

（3）媒体播放器  
     随着多媒体技术的发展应用，在移动电话上进行音频和视频播放已经成为经常使用的功能。由此，Android平台具有支持更多的音频/视频格式，支持更高分辨率的视频，可更流畅地播放视频，以及支持网络流媒体等功能。

（4）媒体获取设备  
     随着移动电话与媒体获取设备的集成日益增强，Android平台提供了照相机、录音机、摄像机等功能。

（5）多类型的连接设备  
     Android平台提供了多种连接方式，如USB、GPS、红外、蓝牙、无线局域网等。

（6）具有友好和绚丽的用户界面  
     Android平台具有友好的用户界面，使用户容易学习和操作；同时具有绚丽的用户界面，具有良好的视觉效果。

（7）可以个性化定制的平台  
     Android平台针对于用户的个性化需求，提供了全面自定义手机的功能。

2.3.3 Android基本框架

Android的系统架构和其他操作系统一样，采用了分层的架构。从以下架构图来看，Android分为四个层，从高到低分别是应用程序层、应用程序架构层、系统运行库层和Linux内核层。

1. Applications

应用程序Android将发布一个核心应用程序包，包括电子邮件客户端，短信短信程序，日历，地图，浏览器，联系人管理程序。 所有的应用程序都是用Java编写的。

1. Application Framework

Framework层为我们开发应用程序提供了非常多的API，我们通过调用特殊的API构造我们的APP，满足我们业务上的需求。写APP的人都知道，学习Android开发的第一步就是去学习各种各样的API，什么Activity,Service,Notification等。这些都是framework提供给我们的，那么我就详细的讲讲Framework到底在整个Android架构中扮演着什么角色。

 Framework其实可以简单的理解为一些API的库房，android开发人员将一些基本功能实现，通过接口提供给上层调用，可以重复的调用。

我们可以称Framework层才真正是Java语言实现的层，在这层里定义的API都是用Java语言编写。但是又因为它包含了JNI的方法，JNI用C/C++编写接口，根据函数表查询调用核心库层里的底层方法，最终访问到Linux内核。那么Framework层的作用就有2个。

所有的应用程序都由一系列的服务和系统组成，主要包括：内容管理器、资源管理器、通知管理器、活动类管理器以及可扩展的视图等等。

1. Libraries库

Android包括一个被Android系统中各种不同组件所使用的的C/C++库集。Android 的各种库一般是以系统中间件的形式提供的，它们均有的一个显著特点就是与移动设备的平台的应用密切相关。该库通过Android应用程序框架为开发者提供服务。

1. Android Runtime

Android包括了一个核心库，该核心库为开发人员提供了java编程语言核心库的大多数功能。

每一个android应用本身的过程中运行，也拥有独立Dalvik虚拟机实例。Dalvik同时复数VMs高效能实行。Dalvik虚拟机执行。dex的Dalvik执行文件格式文件，对最小内存使用了最适化。这个虚拟机寄存器，所有的类由java汇编器编译，经由SDK中的型工具转化为dex格式是虚拟机运行。。

1. Linux内核

Android的核心系统服务依赖于linux2.6内核，如安全性，内存管理，进程管理，网络协议栈和驱动模型。硬体和软件的作为硬件的抽象层。未来的发展的老字号厂商Symbian智能手机软件平台Android的发言人谷歌只是另外的Linux，symbian其他软件和形成的竞争不担心。除了北美之外，Symbian在其它地区智能手机市场都占有很大一部分的市场份额。与iPhone很类似，Android采用webkit浏览器发动机，具备触摸屏，高级图形显示和互联网的功能，用户用手机确认邮件，搜索网站和录像节目等，iphone等其他更强调检索功能的收集，界面强大，可以说是一种所有的单一平台Web应用程序。

2.3.4 Android系统的四大组件

[Android](http://lib.csdn.net/base/15" \o "Android知识库" \t "http://www.cnblogs.com/to-creat/p/_blank)四大组件分别为activity、service、content provider、broadcast receiver。

Activity：

活动是Android程序和用户交互窗口，是Android架构最为基础的一个，它需要维护界面状态，做很多持久的事情，妥善管理生命周期和一些跳转逻辑。

Activity之间通过Intent进行通信。在Intent的描述结构中，有两个最重要的部分：动作action和动作对应的数据data,还有一个类型category。

通过将Intent对象传递给Context.startActivity（）或Activity.startActivityForResult（）来加载（或指定新作业）来激活活动。 可以通过调用getIntent（）方法来查看相应的活动，以查看其激活它的意图。 如果它期望开始返回结果的活动，它将用调用startActivityForResult（）替换startActivity（）。 例如，如果启动另一个活动以使用户可以选择照片，则可能会怀疑选择了哪张照片。 结果将被包装在Intent对象中，并传递给调用活动的onActivityResult（）方法。

Service：

后台服务于Activity，封装有一个完整的功能逻辑实现，接受上层指令，完成相关的事物，定义好需要接受的Intent提供同步和异步的接口。

服务不能自己运行,需要通过Context.startService()或Context.bindService()启动服务。  
 StartService（）方法启动的服务与调用者无关，即使调用者关闭，服务仍在运行。 要停止服务，请调用Context.stopService（），并且系统将调用onDestory（）。 当使用此方法时，系统首次在onCerate（） - > onStartCommond（）上启动服务时，如果服务已经重新启动，它将仅触发onStart（）的Commod（）方法，该方法调用onDestroy（ ）方法被破坏。 只要调用者关闭服务并终止，则使用bindService（）启动的服务与调用方一起调用。 当此方法启动时，服务启动系统首次调用服务onCreate（） - > onBind（），如果服务已经重新启动Call将不会触发这两种方法，调用者将调用服务onUnbind（） - > onDestory（），想主动提取绑定可以使用Contex.unbindService（），系统调用onUnbind（） - > OnDestory（）。

BroadCast Receiver：

接受一种或者多种Intent作触发事件，接受相关消息，做一些简单处理，转换成一条Notification，统一了Android的事件广播模型。例如：电量改变，开机，收发短信，屏幕解锁。

两种广播类型：

无序广播，通过Context.sendBroadcast(Intent myIntent)发送的。(新闻联播)

有序广播，通过Context.sendOrderedBroadcast(intent, receiverPermission)发送的，该方法第2个参数决定该广播的级别，级别数值是在-1000 到1000 之间, 值越大 , 发送的优先级越高；广播接收者接收广播时的级别（可通过intentfilter中的priority进行设置设为1000时优先级最高），同级别接收的先后是随机的， 再到级别低的收到广播，高级别的或同级别先接收到广播的可以通过abortBroadcast()方法截断广播使其他的接收者无法收到该广播。

Content Provider：

Content Provider是由Android访问程序提供的第三方应用数据，您可以派生内容提供者类，外部数据，可以排序与数据库排序一样，屏蔽内部数据存储细节，提供统一的界面模型，大大简化 上层应用，数据集成提供了更方便的方式。

2.3.5 SQLite简介

SQLite，是一款轻型的数据库，是遵守[ACID](http://baike.baidu.com/item/ACID/10738" \t "http://baike.baidu.com/item/_blank)的关系型[数据库管理系统](http://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F" \t "http://baike.baidu.com/item/_blank)，它包含在一个相对小的C库中。它是D.RichardHipp建立的公有领域项目。他的设计不是嵌入式目标，现在使用了很多嵌入式产品，占用的资源很少，只有在嵌入式设备中，可能是必要的。 支持这个Windows / Linux / Unix和一些主流的操作系统，还有很多编程语言，如Tcl，C＃，PHP，Java，有接口，同样的Mysql相比PostgreSQL这两个开源的世界 着名的数据库管理系统，这是他们的处理速度更快。 SQLite第一个alpha版本在2000年5月。 至今已有17个年头，SQLite也迎来了一个版本 SQLite 3已经发布。Android作为目前主流的移动操作系统，完全符合SQLite占用自软少的优势，故在android平台上，集成了一个嵌入式关系型数据库—SQLite。如果你想要开发android应用程序，一定需要在android上存储数据，使用SQLite数据库是一种非常好的选择。

SQLite3支持 NULL、INTEGER、REAL（浮点数字）、TEXT(字符串文本)和BLOB(二进制对象)等五种数据类型； 虽然它支持的类型只有五种，但实际上sqlite3也接受varchar(n)、char(n)、decimal(p,s) 等数据类型，只不过在运算或保存时会转成对应的五种数据类型。

SQLite数据库的特点：

Android通过 SQLite 数据库引擎来实现结构化数据的存储。在一个数据库应用程序中，任何类都可以通过名字对已经创建的数据库进行访问，但是在应用程序之外就不可以。

SQLite 数据库是一种用C语言编写的嵌入式数据库，它是一个轻量级的数据库，最初为嵌入式设计的。它是在一些基础简单的语句处理上要比oracle / mysql快很多，而且其对内存的要求很低，在内存中只需要几百KB的存储空间。这是Android中采用 SQLite 数据库的主要原因。

SQLite 支持事务处理功能。

SQLite 处理速度比MySQL等著名的开源数据库系统更快。它没有服务器进程。

SQLite 通过文件保存数据库，该文件是跨平台的，可以自由复制。一个文件就是一个数据库。数据库名即文件名。

JDBC会消耗太多系统资源，所以JDBC对于手机并不合适，因此Android提供了新的API来使用 SQLite 数据库。

SQLite最大的特点是你可以把各种类型的数据保存到任何字段中，而不用关心字段声明的数据类型是什么。 例如：可以在Integer类型的字段中存放字符串，或者在布尔型字段中存放浮点数，或者在字符型字段中存放日期型值。

1. **系统需求分析**

**3.1系统目标**

本系统是日程管理系统，可以方便用户平时对自己日程的管理和统计。本系统应达到以下目标：

1. 软件采用人机交互的方式，界面简洁大方并且美观友好，操作灵活、方便、富有个性化。
2. 能够对日程管理系统中用户定义的数据进行基本操作和统计。
3. 系统应最大限度地实现易维护性和易操作性。

**3.2系统功能需求**

该系统是一个基于安卓平台的手机日程管理软件。它是为本软件的持有者提供日程管理方面服务的，它包括日历的查看，日历上日期对应日程的查看、总日程的查看、日程的增加、日程的删除、日程的修改、对本软件的设置以及软件相关信息的查看等。

日程管理APP主要分为两大块功能：

1. 日程记录功能：日程添加、查看、修改等

日程添加功能顾名思义就是添加新的日程，内容包括日程类型的选择、日程内容的填写、日程时间和日程耗时的填写、以及日程的评分。被添加的日程的所有信息将添加到数据库中，以后如果需要相关信息可以在数据库中获取并供用户查看。

日程查看功能分为两种查看方式，分别是在日历中查看和以列表形式查看。日历查看的话就是选择日历上的一个日期就会显示当前日期的日程信息。列表形式查看是把所有日程以列表的形式简单显示，选择其中的某一个就可以看到该日程的详细信息。

日程修改功能是在查看日程时可以对日程的一些信息进行修改，适应用户可能临时变化的日程安排。

2.系统设置：设置日程类型 日程提醒 感想记录等

设置日程类型就是用户可以自定义日程的类型，比如把日程分为必要的、重要的和次要的等等的类型，这些设置好的日程类型会存入数据库里边，在进行日程添加或者修改的时候查询出来显示到选择日程类型的地方。类型选择的话就是在日程添加或者修改的时候查询出自定义的日程类型供用户选择。

日程提醒功能是根据用户自己的设定，选择哪些日程需要开启提醒功能，一旦开启了提醒功能，在日程时间将至时会用铃声或者推送通知等的方式来提醒用户。

感想记录功能是用户根据日程的完成情况做的一些总结和感想，这些信息也会存在数据库中，用户以后可以查看自己之前做的一些感想记录。

**3.3系统性能需求**

数据处理的准确性和实效性。在系统中，数据的输入也许是大量的，因此系统要有一定的处理能力，以保证能够迅速的处理输入的数据。同时对于输入的数据，如果不能符合能够存储的范围，需要给出提示并拒绝该数据。

在时间特性方面，一般操作的响应不能超过1.5秒，使用户感受到瞬间有反馈的感觉，提升用户使用的体验。

手机屏幕下的日程管理系统，系统界面要求布局合理，颜色舒适，控制按钮友好，要求整体有一个主要的元素标准，统一的主题思想。

**3.4系统运行环境**

本系统是基于移动平台，如今的移动领域有Apple的iOS，Google的Android，以及Microsoft的Windows phone三大平台，其中Android的市场份额在三大平台中是最大的，另外Android的跨结构中处于顶层的是应用开发层，而且是基于Java语言的开发。所以本系统采用Android平台为用户提供一个日程管理的解决方案。为了设计更好的用户体验，适应智能移动终端的快速更新换代，采用稳定且比较新的Android SDK进行开发。

1. **系统设计**

**4.1系统总体设计描述**

系统的总体设计阶段所需要解决的问题是构建整个项目的总体框架，因为如果自己没有确定好一个明确的框架，那么代码的编写一旦出现了一点差错，就会使得后边的编程工作越来越乱，甚至导致整个系统的编写需要再次从头开始。所以说系统总体设计阶段很重要。软件工程同时也是一门工程性的学科，所以需要对本系统提前搭建一个总体的框架。

经过需求分析阶段的工作，本“日程管理系统”所要完成的功能已经成型，现在所要做的工作是设计整个系统的工作原理。

总体设计过程可分为两个主要阶段：功能设计，根据各种资料和调查决定本系统的相应功能；结构设计，确定整个软件的结构，由哪些子类组成。其中功能设计是在需求分析的基础上进行的，这里的“功能”是泛指的概念，例如软件所要实现的功能，还包括系统设计师的数据处理，实现功能的算法。而结构设计，是将整个系统按照不同的功能和层次划分为一个一个相对简单而独立的模块，每个模块实现系统的某一项具体的功能，自顶向下，逐步求精，这样增加了软件的可复用性，而且对于代码的结构则更加清晰明了，阅读起来也更加有条理，维护起来的也更加方便省事。

**4.2系统总体设计**

模块层次图是描述软件层次结构的工具，它描述了模块如何管理和控制上下模块之间的哪些模块和一些数据传输关系，或者统计模块可以清楚地指出系统的结构，并且可以使用 粗略估计整个系统的大小。

根据需求分析所得到的一些数据和信息，得出该系统只要实现数据保存、数据查询、数据统计以及数据更新删除等。

经过分析研究，日程管理系统的总的模块结构图如图4-1。

日程管理系统

主界面

日历模块

日程模块

设置模块

日历显示

总日程显示

类型设置

单一日程显示

图4-1

系统的功能模块说明：

1. 日历模块：系统拥有一个完整的日历界面，包括头部的年月日。在日历界面上每个日期都会跳转到相应日期的日程显示界面。日历界面上还会对有日程的日期进行标记，方便用户快速查看某天的日程信息。
2. 日程模块：日程模块的主要功能是日程的添加和日程的查看。日程的查看以列表的形式为用户显示了当日所有的日程。在本模块中包括总的日程显示页面和日历页面所跳转到的日程显示界面。总日程显示界面中每一个日程都会有它所包含的耗时、日程类型、日程内容、日程打分情况。日程的添加，用户可以添加新的日程并有选择日程的类型、写入日程内容、日程时间、日程耗时和日程评分这五项功能。
3. 设置模块：在设置模块中主要包括了对日程类型的自定义，然而还有提醒铃声的设置、是否打开提醒功能等等一些小的功能。另外用户还可以写入对当天日程完成情况的感想和总结。

**4.3数据库设计**

根据该应用的功能分析和系统流程的分析，该应用主要涉及的实体有日程信息、日程类型等。期间可能还会有一些存储临时变量的数据表。日程信息和日程类型之间有如下关系：一条日程信息只能有一个类型，一个日程类型可以对应多个日程信息。主要的实体对应的er图如图：

总体的er图：

时间

日程内容

耗时

编号

类型

n n

日程信息

1 1

日程类型

日程提醒

开启提醒

日程编号

类型

类型编号

图4-2

日程信息：

编号

时间

耗时

类型

日程内容

日程信息

图4-3

日程类型：

编号

类型

日程类型

图4-4

1. **系统实现**

**5.1主界面的实现**

5.1.1系统主界面的头部菜单栏

制作这个头部菜单栏是为了提升本软件的用户友体验，使软件操作起来更加方便，我用的是TabHost+RadioGroup相结合的方法来实现这个部分的。菜单栏的底部位置是通过TabHost这个布局来实现的，然后是LinearLayout的线性布局来布置内容。内部的选项卡是通过TabWiget来实现。首先新建一个android的工程（工程名为：TimeManager），在工程的res/layout下建立一个名为main的空xml文件，然后在这个文件里按照思路来添加布局代码。

主界面的菜单选项是使用了RadioGroup来建立，在这里的布局中RadioGroup里容纳4个RadioButton，其中的RadioButton是单选按钮。在这种情况下有且只有一个RadioButton能被选择，并且开始的时候会默认选中第一个按钮。其中RadioButton的实现方式是在设置内部的文字之后，然后再按钮的顶部上添加上图片，实现的代码如下：

<RadioButton

android:text="@string/main\_date"

android:drawableTop="@drawable/tab\_date"

android:id="@+id/radio\_button0"

android:checked="true"

style="@style/main\_tab\_bottom"/>

从上述代码可以看到，在菜单上边的每个选项卡样式都是调用了style上的main\_tab\_bottom来实现自己的样式的，而main\_tab\_bottom的背景的值是@drawable/btn\_bg ，btn\_bg不是用一个简单的背景的图片实现，而是用一个xml文件，通过这个文件可以实现不同菜单选项卡切换时的背景切换的效果，使得这个菜单栏看起来更加的美观，代码是使用了安卓里边的selector来实现，在这里是通过进行不停的条件来判断使用哪一个item的，里面包括了几个常用的属性值，如是否获得焦点、是否响应事件、是否点击这三个属性条件值。具体的实现代码如下：

<selector xmlns:android="http://schemas.android.com/apk/res/android">

<item android:state\_focused="true"

android:state\_enabled="true"

android:state\_pressed="false"

android:drawable="@drawable/btn\_bg\_s" />

<item android:state\_enabled="true"

android:state\_pressed="true"

android:drawable="@drawable/btn\_bg\_s" />

<item android:state\_enabled="true"

android:state\_checked="true"

android:drawable="@drawable/btn\_bg\_d" />

</selector>

设计完主页面布局后，在MainPage文件中OnCreate中调用布局，到现在为止顶部菜单栏的布局工作已经基本完成，但现在布局里面所对应的内容还是空的。所以接下来就需要在这里给TabHost中添加tab得内容，以第一格的tab为例，先是通过Intent的方式新建个mDate，这个mDate用来实现页面的跳转的方法，然后动态调用tabHost的addTab方法添加对应TabSpec的值和各个属性。最后在添加RadioGroup的选择操作函数,这里是通过switch来完成对应按钮的功能的实现，在每个case项中通过tabHost.setCurrentTabByTag分别给予特定的值实现页面上的跳转。

5.1.2底部菜单项的实现

底部菜单项是通过安卓手机底部的meun键来跳出菜单实现各种操作。

5.1.2.1 日历界面使用的底部菜单项

首先是创建菜单，这些菜单项作用分别是跳转到今天、实现自己设置日期的跳转、查看所有日程、直接退出本软件系统。

public boolean onCreateOptionsMenu(Menu menu) {

menu.add(0, menu.FIRST, menu.FIRST, "今天");

menu.add(0, menu.FIRST+1, menu.FIRST+1, "跳转");

menu.add(0, menu.FIRST+2, menu.FIRST+2, "日程");

menu.add(0, menu.FIRST+3, menu.FIRST+3, "退出");

//设定菜单的值与文本内容

return super.onCreateOptionsMenu(menu);

}

然后是通过onMenuItemSelected（）中switch来实现菜单功能

public boolean onMenuItemSelected(int featureId, MenuItem item) {

switch (item.getItemId()){

case Menu.FIRST:

//中间写对应的的方法

break;//后面省略

在这些功能中，分别写入对应的方法：

（1）今天的选项是让页面不在今天所在月份时的情况跳回来的

//跳转到今天

int xMonth = jumpMonth;

int xYear = jumpYear;

int gvFlag =0;//日期标记的初值

jumpMonth = 0;//月份的初值

jumpYear = 0; //年份的初值

addGridView();//添加一个gridView，即添加日期界面

year\_c = Integer.parseInt(currentDate.split("-")[0]);

month\_c = Integer.parseInt(currentDate.split("-")[1]);

day\_c = Integer.parseInt(currentDate.split("-")[2]);

//分别取得年、月、日的值，并且设定一定的格式

calV = new CalendarView(this, getResources(),jumpMonth,jumpYear,year\_c,month\_c,day\_c);

gridView.setAdapter(calV);//将calV值加入到日期的总布局中

addTextToTopTextView(topText);//创建头部文本

gvFlag++;//日期标识的递增

flipper.addView(gridView,gvFlag);

if(xMonth == 0 && xYear == 0){

}else if((xYear == 0 && xMonth >0) || xYear >0){

this.flipper.showNext();

}else{

this.flipper.showPrevious();

}//判断条件实现翻页的效果

flipper.removeViewAt(0);

跳转项点击时，会弹出一个时间选择框，来修改要跳转到的时间，最后点击设置可以实现日期页面上时间的跳转，跳转的范围用条件函数设定在一定的范围之间，如果超出这个范围则跳出错误提示。

跳转所需的判断函数，当设置的时间不在范围内时，跳出当前会话界面。

if(year < 1901 || year > 2049){

//不在查询范围内

new AlertDialog.Builder(CalendarActivity.this).setTitle("错误日期").setMessage("跳转日期范围(1901/1/1-2049/12/31)").setPositiveButton("确认", null).show();

}

而在范围内时则会跳转到输入的日期，这就需要重新加载入界面，同时也需要重新载入标记，以完善页面的显示。

（3）日程选项是跳到所有日程的界面，实施跳转的代码较简单，如下所示：

Intent intent = new Intent();

intent.setClass(CalendarActivity.this, ScheduleActivity.class);

//设置跳转的class文件参数

startActivity(intent);//运行

退出选项是退出本软件：

finish();//退出的方法

5.1.2.2 其他界面使用的底部菜单项设计与实现

（1）总日程页面的底部菜单项只有一个返回选项，用于回到主页。

menu.add(1, menu.FIRST, menu.FIRST, "返回");

return super.onCreateOptionsMenu(menu);

//菜单的选择项返回

switch(item.getItemId()){

case Menu.FIRST:

Intent intent = new Intent();

intent.setClass (ScheduleActivity.this, MainPage.class);

//设置跳转的class文件参数

startActivity(intent);//运行

break;

}//返回项的实现效果

return super.onOptionsItemSelected(item);

（2）日程添加页面、关于页面、设置页面，没有编写底部菜单栏。因为在日程添加界面上的取消按钮已经能实现其效果，关于界面与设置界面都和日历的这个默认界面都是在一个头部菜单栏下，所以可以很方便的实现所需要的效果。

（3）单一的日程页面的底部菜单项则是包括所有日程、添加日程、返回3个选项，分别用于跳转到总日程界面、添加日程界面和返回主界面。

其中跳转到添加日程界面，由于需要带着当前页日程的时间到添加页面，所以要完成数据传送，在它所属的case中写入：

intent = new Intent();

Intent intentOther = getIntent();

schduleDate = intentOther.getStringArrayListExtra("scheduleDate");

//取得日期的数组值

intent.putStringArrayListExtra("scheduleDate", scheduleDate);

//传入日期的数组值

intent.setClass(ScheduleInfoView.this, ScheduleAddActivity.class);

//设置跳转的class文件参数

startActivity(intent);//运行

这样所有的底部菜单及所需功能基本都实现完成了。。

**5.2日历模块的实现**

日历模块主要包括日历界面的显示和单个日程的显示。

5.2.1日历的显示

系统主界面默认选择日历界面在头部菜单栏下方。

（1）首先新建一个布局文件，上面一部分是显示年月日、生肖和农历的干支纪年。使用的是一个TextView类，对应的在日历的CalendarActivity文件中加入函数来进行添加头部的年份等信息的工作。

首先需要写了一个CalendarView类，这是为了进行日历显示所写的一个类，在其中使用了函数重载的方法定义了几个不同参数的CalendarView，用于不同的情况。

在CalendarActivity中实例化不同参数的CalendarView并且使用CalendarView中的一个方法CalendarView(this, getResources(),jumpMonth,jumpYear,year\_c,month\_c,day\_c);然后编写一个方法来添加头部的年份、 闰哪月等信息。

首先是对这个页面的头部进行一下布局，这里是通过动态化的方法来完成的。

然后建立一个可以改变值的StringBuffer类型的参数textDate。对这个参数使用append方法进行操作，添加例如年月日的参数。其中还有一个闰年判断的条件函数，是通过calV.getLeapMonth()的返回值进行判断的。以上这些都完成后就可以对界面使用setText方法来赋值到界面上。

最后是对界面的一些美化，同样使使用的动态布局模式进行美化的。

（2）日历界面中主体日期部分显示的完成

在CalendarActivity文件初始化实例对象时使用动态添加布局文件的方法。

将该方法命名为addGridView。在这个方法中同样是先进行动态化布局，由于这里用的是GridView，这个GridView应用于将一系列的空间组织成二维网格的形式显示出来，所以很适合于我在这里进行的日期界面的建立。

同样的首先是动态化布局的建立，这可以实例化GridView为gridView参数进行的。在这里需要添加GridView控件的列为7，列宽为46。并且还加上判断屏幕大小的条件，这需要加入取得屏幕大小的对应函数，这里如果是800\*480屏幕的话，就将列宽设置为69。然后进行设置水平和竖直的间距、取值边框、设置居中等操作。

在这个方法里面还需加入个OnTouch方法，用来将gridView中的触摸事件回传给gestureDetector，进行下面的触摸操作，即：

//将gridview中的触摸事件回传给gestureDetector

@Override

public boolean onTouch(View v, MotionEvent event) {

return CalendarActivity.this.gestureDetector

.onTouchEvent(event);

}

});

创建完日历的布局后，接下来就是在里面添加上具体的日期。就当前显示的页面来说，是建立几个对应的参数，用来显示或传接数据。例如在函数开始时就使用String[] dayNumber = new String[49]; 这是作为对GridView控件内来进行标记序列的，用于对下面条件语句的判断等，还有就是在界面内的一些控件的实例化。 然后就是动态化的对页面的属性值进行设置。

其中在日期中的第一行，即周一等值的这行，设置字体的颜色以及背景图。得出这行的位置就需要通过对参数的对应编号进行判断，判断其是否在7以内。

下一步是当前月信息的显示，这也需要条件函数来判断是否为当前月，也是来判断参数position是否在范围内。在这个月中还需要对当天来一个特殊化的设置，即设置当天日期的背景为图片current\_day\_bgc，字体颜色为白。

在这些操作外，还需要得到一些数值来用于日期数据的插入，在这里建立了个getCalendar方法，如判断是否为闰年的函数返回值isLeapyear，某月的总天数，某月的第一天是星期几，上个月总天数。在这方法中再执行方法getweek，用来得出以后需要标记的日期，这个在后面来介绍。

通过上述的步骤基本完成了日历界面中的初步实现。

（3）由于日历菜单需要能实现通过手势触摸来进行翻页，所以要在总体布局下建立一个ViewFlipper类，这个类是用来实现屏幕的不同切换效果的。首先是在布局中建立这个类，然后实例化手势gestureDetector并取得布局中ViewFlipper来用于下面的调用。

然后是编写书写手势的各个方法，最主要的是onFling的方法，在这里实现了页面的向左及向右滑动的操作，这个操作是由条件语句来实现的，由滑动两点的x坐标的值进行差值计算，如果是起点减去重点大于120时着向左滑动，反之则向右滑动。在滑动操作开始，并且已切换一面后就需要载入新的一张日历页面，这里又会用到参数calV，当向左滑动时则将参数calV所代表的函数CalendarView里的jumpMonth加1，代表下一个月；当向右滑动时则将参数calV所代表的函数CalendarView里的jumpMonth减1，代表上一个月。为了完整性剩余的手势函数只是写了方法未使用。

本系统日历的特色显示方式：

在日历的显示中除了一些基本的日期显示，在日期页面头部会有每年相应的生肖和闰年显示，在日期上有每日的阴历以及节假日的显示。

这里有两个方法，分别是LunarCalendar 和SpecialCalendar。

在LunarCalendar里面包括干支纪年，阴历，生肖，节假日等信息。

例如生肖则是建立一个字节数组来记录十二生肖，并且用函数来传回参数对应的生肖。

final public String animalsYear(int year) {

final String[] Animals = new String[] { "鼠", "牛", "虎", "兔", "龙", "蛇","马", "羊", "猴", "鸡", "狗", "猪" };

return Animals[(year - 4) % 12];

}

同样的其他信息，如节假日的名称，也是通过字节数组来储存的。

在SpecialCalendar是闰年的判断函数，每月天数，以及对每月的第一天星期的判断。

闰年通过传回年值通过函数进行判断，并传回布尔值。

每月的天数用switch进行判断赋值给月份天数的参数。

对每月的第一天为星期几的判断并指定。

//指定某年中的某月的第一天是星期几

public int getWeekdayOfMonth(int year, int month){

Calendar cal = Calendar.getInstance();

cal.set(year, month-1, 1);

dayOfWeek = cal.get(Calendar.DAY\_OF\_WEEK)-1;

return dayOfWeek;

}

5.2.2对有日程的日期的标记

这个日程管理系统为了给予用户更多的便利，在日历的显示上都会有对于已有日程的日期的进行标记，这样就能让用户能够直观的看到自己哪些天会有日程，使得管理更直观，更具有针对性和实用性。

日期上标记的实现

使用判断条件来判断是否需要标记，这里是使用数据库中的scheduletagdate表里数据来判断的，以这个日期与表中的数据相比较，如果是需要标记的则设置日程标记背景。

if(schDateTagFlag != null && schDateTagFlag.length >0){

for(int i = 0; i < schDateTagFlag.length; i++){

if(schDateTagFlag[i] == position){

//设置日程标记背景

textView.setBackgroundResource(R.drawable.mark);

}

}

}

保存任务时标记的存储

在保存任务时将本次任务的相关信息存入scheduletagdate表中，用于日期上的标记这样就能实现在有日程的日期上显示出明显的标识。

int schduleID = dao.save(schedulevo);

//设置日程标记日期(将所有日程标记日期封装到list中)

setScheduleDateTag(remindID, scheduleYear, tempMonth, tempDay, scheduleID);

**5.3日程模块的实现**

日程模块包括添加日程，修改删除日程，还包括日程的列表查看功能以及单一日程的查看。

5.3.1添加日程

添加日程的界面是通过在主界面点击添加按钮进行跳转的。

（1）首先为了界面的美观我编写了几个带边框的类，这些类是在原有的textview等基本类的基础上进行一些美化后形成的。这里一共做了下面的三个实现了画笔效果的类。

首先是带下边框的TextView，这里TextView类的基础上用画笔实现了下边框的效果。

Paint paint = new Paint();// 实例化一支画笔

paint.setColor(android.graphics.Color.BLACK); // 设置所绘制的边框颜色为黑色

canvas.drawLine(0, this.getHeight() - 1, this.getWidth() - 1, this

.getHeight() - 1, paint); // 绘制下边框

带边框的EditText，这原本是一个基本的EditText，文本编辑框的类，在这里加上了特殊的样式、颜色及边框的圆弧效果。

Paint paint = new Paint();// 实例化一支画笔

paint.setStrokeWidth(1);//设置空心现款

paint.setStyle(Style.STROKE);//设置样式

paint.setColor(android.graphics.Color.GRAY);//设置颜色

paint.setAntiAlias(true);//抗锯齿方法

RectF rectF = new RectF(2,0,this.getWidth()-2,this.getHeight()-2);

//现在设置相关的参数用于下面的操作。

canvas.drawRoundRect(rectF, 8, 8, paint);//画上圆弧

最后一个是带边框的TextView，这个与上述的EditText相类似，完成这个列。

完成这些边框类之后就是使用它们，这个与以前的基本类相似，只是将布局中原来的基本类名类，如TextView，改为对应路径的名称为com.manage.borderText.BorderTextView。（BorderTextView是新建的TextView类的文件名。）

（2）进行日程界面的布局的添加，新建一个名为schedule的空布局文件。这里的使用的布局是相对布局RelativeLayout。这样可以更清晰将这些个框框组合在一起，使得更加美观。

在这个界面中，总体的以RelativeLayout这个相对布局名为总体的布局标准，最上面是标题栏，这就是一个上述建立的一个带边框的TextView类。

<com.pwp.borderText.BorderTextView

android:id="@+id/scheduleTop"//id号

android:gravity="center"//位置居中

android:layout\_width="fill\_parent"

android:layout\_height="40dp"//这两个分别为宽和高

android:background="@drawable/top\_day"//背景图片的位置

android:textStyle="bold"//文本的样式加粗

android:clickable="true"//可点击的选项

android:textColor="#000000"//文本颜色设为黑色

android:text="编辑日程"//文本框上的文字

/>

下面是日程类型的选择点击框。由于使用的相对布局，那么就需要自己设置这两个点击框的位置。这里先建立了日程类型的文本框，以layout\_below来将这个文本框放在标题框（用id来定位标题框）下面，下面几个需要特殊设定的参数：

android:layout\_below="@id/scheduleTop"//文本框的位置设置

android:layout\_width="160dp"//文本框的宽

android:layout\_height="40dp"//文本框的高

android:drawableRight="@drawable/schedule\_type"//在这个文本框右边添加一个图标然后是选择日程类型的文本框，这需要放在日程类型文本框的右边，这是利用

layout\_ toRightOf来实现，相应的也要设置位于标题框下方，其他的与上面的相类似。

android:layout\_toRightOf="@id/scheduleType"//文本框的位置设置位于右方

android:drawableRight="@drawable/schedule\_type"//文本框的位置设置位于下方

（3）写一个ScheduleAddActivity 类来调用布局和书写方法

新增界面的上方的TextView是用来选择类型的，另有一个时间显示框可以用来显示传过来的时间或者修改时间。

首先是取得这TextView的位置，把它实例化，以便于后面的使用，

scheduleType = (BorderTextView) findViewById(R.id.scheduleType);

点击时跳到选择界面并传值，跳转的时候需要获得原来界面上的日程类型，这里需要在点击触发效果的方法上将数据传到类型选择页面上去，用于显示它的默认值。

scheduleType.setOnClickListener(new OnClickListener() {

//日程类型的点击事件

@Override

public void onClick(View v) {

//点击时方法实现的内容

schText = scheduleText.getText().toString();

Intent intent = new Intent();

intent.setClass(ScheduleAddActivity.this, ScheduleTypeView.class);//实现页面从日程添加页面到日程类型选择页面的跳转

intent.putExtra("sch\_remind", new int[]{sch\_typeID,remindID});

//跳转时需要传过去的数据，两个数据组成一个数组传过去

startActivity(intent);//运行

}

});

这样就需要一个新建ScheduleTypeView文件来实现跳转后的界面。

这里还包括一个CalendarConstant类，这个类是用于记录日程类型的字符数组，以便于类型选择的调用。

public class CalendarConstant {

public final static String[] sch\_type = { “重要但不紧急”,”不重要但紧急”,”重要而且紧急”,”不重要也不紧急”}; // 日程类型的值

}

这里是存储ScheduleTypeView和ScheduleRemindView里面选项的value值。

这个页面是通过实例化来实现布局的：

LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(LayoutParams.FILL\_PARENT, LayoutParams.WRAP\_CONTENT); //设置宽与高

cc = new CalendarConstant();

layout = new LinearLayout(this); // 实例化布局对象

layout.setOrientation(LinearLayout.VERTICAL);

layout.setBackgroundResource(R.drawable.schedule\_bk);//设置背景图片

layout.setLayoutParams(params);

实例化布局里的组件需要为后面使用时做准备，用于给这些组件添加各项参数。

group = new RadioGroup(this);//选择按钮组

btSave = new BorderTextView(this,null);//保存按钮实例化

btCancel = new BorderTextView(this, null); //取消按钮实例化

textTop = new BorderTextView(this, null); //顶部文本实例化

这里有个从ScheduleAddActivity传来的值，是日程新增界面里的日程类型的已选择的ID值，用来选择RadioGroup的默认选项

Intent intent = getIntent();

int sch\_remind[] = intent.getIntArrayExtra("sch\_remind");

if(sch\_remind != null){ //判断不为空时进行下列的赋值

sch\_typeID = sch\_remind[0]; //日程类型的赋值

remindID = sch\_remind[1]; }//重复类型的赋值

for(int i = 0 ; i < cc.sch\_type.length ; i++){

radio = new RadioButton(this);

if(i == sch\_typeID){ //判断是否是当前选择的项

radio.setChecked(true); //设置为已选择

}

radio.setText(cc.sch\_type[i]); //设置文本内容

radio.setId(i); //设置编号

radio.setTextColor(Color.BLACK); //设置文本颜色

group.addView(radio); //设这系列的内容添加到数组中

}

将RadioGroup加入到布局中去：

layout.addView(group);

RadioGroup的触发效果，再点击这里的每一个选项都会显示已选中并且会将选择项的ID值传到schType\_temp中用于后面的使用。

group.setOnCheckedChangeListener(new OnCheckedChangeListener() {

@Override

//重写onCheckedChanged方法

public void onCheckedChanged(RadioGroup group, int checkedId) {

schType\_temp = checkedId; //给在上面建立的参数赋值

}

});

添加确定和取消按钮，此处只以确定按钮为例，按钮的话在这里也是要通过动态方法来实现的，先新建一个布局参数

layButton = new LinearLayout(this);

然后给参数设置各项属性，这些都是按钮所需要兼备的基本属性，有利于按钮的美观与实用性，

layButton.setOrientation(LinearLayout.HORIZONTAL);

layButton.setLayoutParams(params);

btSave.setTextColor(Color.BLACK); //设置文本颜色

btSave.setBackgroundResource(R.drawable.top\_day); //设置背景

btSave.setText("确定"); //设置按钮上的文本信息

btSave.setHeight(47); //设置高

btSave.setWidth(160); //设置宽

btSave.setGravity(Gravity.CENTER); //设置按钮位置

btSave.setClickable(true); //设置按钮的可点击性

layButton.addView(btSave); //将保存按钮添加到按钮参数中

最后将整个layButton的参数加入到整体布局中去，这里面包括添加的保存和取消按钮这两个功能按钮。

layout.addView(layButton);

按钮的界面建立之后就需要为这两个按钮添加它上面的触发效果，首先是确定按钮，这个按钮的设计是当点击按钮后则会将已得到的schType\_temp值赋给sch\_typeID，然后继续通过传值的方法将其返还到日程添加界面去。这里传值的方法与从日程添加界面传值过来的方法相类似。

//触发确定按钮

btSave.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

sch\_typeID = schType\_temp;

//将上面得到的schType\_temp默认值或改变后的值传给传sch\_typeID。

Intent intent = new Intent();

intent.setClass(ScheduleTypeView.this, ScheduleAddActivity.class);

//设置跳转的class文件参数

intent.putExtra("schType", sch\_typeID);

//将上面得到的sch\_typeID传到schType中以便于调用

startActivity(intent);

}

});

然后是取消按钮的触发实现，这个就比较简单了，只是需要将页面转回到添加页面，但最好也还是把值传回去，以使得日程添加页面的两个类型选择项的值的显示不出现错误，方法类同于保存按钮的触发。

下一步是时间显示框的获得时间和修改时间后的获得，时间框时间的获得需要从日期页面传入时间数据到本页面上来，这里的一系列数据都放在名为scheduleDate的数组中。

intent.putStringArrayListExtra("scheduleDate", scheduleDate);//需要传递的数据

intent.setClass(CalendarActivity.this, ScheduleInfoView.class);//用于实现从单一日程界面跳转到日程界面上去

intent.putExtra("scheduleID", scheduleIDs);

startActivity(intent); //运行

其中scheduleIDs的数据的取得是通过数据库来取的，方法用的就是ScheduleDAO文件中的数据库查询方法getScheduleByTagDate方法来实现的，通过所点击日期上的标记来得到对应日期的日程信息，可以是一个或者几个，所以需要使用的是字符串数组的形式来得到日程的ID值。

String[] scheduleIDs = dao.getScheduleByTagDate(Integer.parseInt(scheduleYear), Integer.parseInt(scheduleMonth), Integer.parseInt(scheduleDay));

然后是获得时间信息，这里的时间信息包括年月日和星期，都是用于显示在日程选择框上的。而这些日期信息都是从scheduleDate这个数组中得来，存在于数组的各个位置，首先取得年月日和星期的信息，用于在界面显示。

scheduleYear = scheduleDate.get(0);//日程年份

scheduleMonth = scheduleDate.get(1); //日程月份

tempMonth = scheduleMonth; //日程年份

if (Integer.parseInt(scheduleMonth) < 10) {

scheduleMonth = "0" + scheduleMonth;

}//判断如果当前月份小于10时则在月号前加0以使得其美观。

scheduleDay = scheduleDate.get(2);

tempDay = scheduleDay;

if (Integer.parseInt(scheduleDay) < 10) {

scheduleDay = "0" + scheduleDay;

}//判断如果当前天小于10时则在天前加0

week = scheduleDate.get(3);

String hour\_c = String.valueOf(hour); //将小时数赋值给所设参数

String minute\_c = String.valueOf(minute); //将分钟数赋值给所设参数

if(hour < 10){

hour\_c = "0"+hour\_c;

}//判断如果当前小时数小于10时则在小时数前加0

if(minute < 10){

minute\_c = "0"+minute\_c;

}//判断如果当前分钟数小于10时则在分钟前加0

后面是得到相关的阴历信息用于显示在时间框中，这样在一些特殊的日子也会有一些特殊的标志显示。以特定的格式来显示

String scheduleLunarDay = getLunarDay(Integer.parseInt(scheduleYear),

Integer.parseInt(scheduleMonth), Integer.parseInt(scheduleDay));

String scheduleLunarMonth = lc.getLunarMonth(); // 得到阴历的月份

最终在时间文本框上上的显示以append的形式添加上去，显示为yyyy-mm-dd hh：mm，换行显示阴历日期及星期数。

StringBuffer scheduleDateStr = new StringBuffer();//建立日期储存的可变字符串

scheduleDateStr.append(scheduleYear).append("-").append(scheduleMonth)

.append("-").append(scheduleDay).append(" ").append(hour\_c).append(":").append(minute\_c).append("\n").append(

scheduleLunarMonth).append(scheduleLunarDay)

.append(" ").append(week); //添加相应的日期数据

return scheduleDateStr.toString(); //返回到字符串

然后是修改时间的触发事件，当点击时间框时就会弹出时间设置的小界面，在这里可以修改自己需要的指定的时间。而弹出这个界面的时候也会将时间文本框的时间传入到其中来显示，使用的是TimePickerDialog的会话框来显示与修改。

（4）ScheduleAddActivity 类页面的保存设计，保存就是点击保存按钮之后进行的相关的操作，保存时首先要判断输入框是否为空，判断之后就可以保存日程信息了，然后实现页面的调转，使其跳转回到单一日程详情的显示页面。

//保存日程信息

if(TextUtils.isEmpty(scheduleText.getText().toString())){

//判断输入框是否为空

new AlertDialog.Builder(ScheduleAddActivity.this).setTitle("输入日程").setMessage("日程信息不能为空").setPositiveButton("确认", null).show();

}else{

//将日程信息保存

String showDate = handleInfo(Integer.parseInt(scheduleYear), Integer.parseInt(tempMonth), Integer.parseInt(tempDay), hour, minute, week, remindID);

ScheduleVO schedulevo = new ScheduleVO();

schedulevo.setScheduleTypeID(sch\_typeID);//给日程类型赋值

schedulevo.setRemindID(remindID); //给重复类型赋值

schedulevo.setScheduleDate(showDate); //给日期赋值

schedulevo.setScheduleContent(scheduleText.getText().toString());

dao.save(schedulevo);

保存完之后是直接调转到ScheduleInfoView即日程（单一）显示页面，以intent的方法来实现。

Intent intent = new Intent();

intent.setClass(ScheduleAddActivity.this, ScheduleInfoView.class);

//跳转到日程显示页面去

intent.putExtra("scheduleID", scheduleIDs);//传入所需的日程ID数组

startActivity(intent);

5.3.2日程列表

（1）列表界面的布局

列表界面使用listview来进行布局。首先需要定义一个实体类，用于需要在列表中显示的信息类。Listview的每个item存着日程的简要信息，包括时间、日程类型、日程的时间、简短的日程内容等信息。新建一个用于显示每个item的布局文件item.xml。Item的布局主要是通过几个textview来进行排布从而实现了显示一条日程的简要信息。布局文件部分代码如下：

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

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

<TextView

android:id="@+id/fruit\_name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginLeft="10dip"/>

<TextView

android:id="@+id/fruit\_name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginLeft="10dip"/>

...

<LinearLayout>

自定义adapter，新建自定义的adapter继承arrayadapter，泛型指定为之前的实体类。构造函数包括上下文、listview的子项布局，还有数据。然后重写getView方法，先通过getitem获取当前实例，然后用LayoutInflater加载布局，调用view的findviewbyid方法与控件联系起来，然后用settext等方法设置内容，返回view。

public class ScheAdapter extends ArrayAdapter<Fruit> {

private int resourceId;

public ScheAdapter (Context context, int textViewResourceId,

List<Fruit> objects) {

super(context, textViewResourceId, objects);

resourceId = textViewResourceId;

}

.....

}

（2）取得所有日程，调用数据库查找方法

得到所有的日程信息，判断日程内容是否为空，为空这显示没有日程的文字，有则列出所有的日程列表，并将需要的信息显示到listview中。

public void getScheduleAll(){

schList = dao.getAllSchedule();

if(schList != null){

for (ScheduleVO vo : schList) {

String content = vo.getScheduleContent();//取得正文内容框

int startLine = content.indexOf("\n");

if(startLine > 0){//判断正文的长度

content = content.substring(0, startLine)+"...";

}else if(content.length() > 30){

content = content.substring(0, 30)+"...";

}

scheduleInfo = CalendarConstant.sch\_type[vo.getScheduleTypeID()]+"\n"+vo.getScheduleDate()+"\n"+content; //正文的书写格式

scheduleID = vo.getScheduleID();

createInfotext(scheduleInfo, scheduleID);

}

}else{

scheduleInfo = "没有日程";

createInfotext(scheduleInfo,-1); //日程空时的显示

}

}

（3）设置监听方法来跳转到ScheduleInfoView来显示单个日程的具体信息，当点击日程列表中某一个时会跳转到这条日程所对应的单一日程的显示界面。即是在日程的显示的每一个框中添加触发事件，点击后者会跳转到相应的页面，并且传入所需的信息。

//点击每一个textview就跳转到shceduleInfoView中显示详细信息

info.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

String schID = String.valueOf(v.getTag());

String scheduleIDs[] = new String[]{schID};

Intent intent = new Intent();

intent.setClass(ScheduleActivity.this, ScheduleInfoView.class);

//设置跳转的class文件参数

intent.putExtra("scheduleID", scheduleIDs); //传日程的ID数组值

startActivity(intent);

}

});

}

5.3.3单一日程详情

(1)日程界面ScheduleInfoView的设计

在单一日程界面，直接使用动态布局显示日程的详细信息，

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

dao = new ScheduleDAO(this);

sv = new ScrollView(this);

params.setMargins(0, 5, 0, 0);//设置四面的边距

layout = new LinearLayout(this); // 实例化布局对象

layout.setOrientation(LinearLayout.VERTICAL);

layout.setBackgroundResource(R.drawable.schedule\_bk);

//设置任务界面的背景图

layout.setLayoutParams(params);

sv.addView(layout);//添加到sv这个布局空间中去

setContentView(sv);

}

创建放日程信息的textview，这个文本框使用于存放日程的所有信息，包括时间、日程类型、日程重复类型、日程内容等信息。

public void createInfotext(String scheduleInfo, int scheduleID){

final BorderTextView info = new BorderTextView(this, null);

info.setText(scheduleInfo);//设置文本值

info.setTextColor(Color.BLACK); //设置文本的颜色

info.setBackgroundColor(Color.WHITE); //设置背景颜色

info.setLayoutParams(params);

info.setGravity(Gravity.CENTER\_VERTICAL);

info.setPadding(10, 5, 10, 5); //设置四面的距离

info.setTag(scheduleID);

layout.addView(info);

日程的信息需要在数据出中调用查询方法来获取。

（2）在ScheduleInfoView界面上有对日程的删除操作，只要长按日程类型就会出现删除选项，删除后后会跳转回主界面。删除日程的界面如图4-7。

长按时时间点击的触发效果：

public boolean onLongClick(View v) {

final String scheduleID = String.valueOf(v.getTag());

new AlertDialog.Builder(ScheduleInfoView.this).setTitle("删除日程").setMessage("确认删除").setPositiveButton("确认", new OnClickListener() {

//删除界面确定的的实现

public void onClick(DialogInterface dialog, int which) {

dao.delete(Integer.parseInt(scheduleID));

Intent intent1 = new Intent();

intent1.setClass(ScheduleInfoView.this, MainPage.class);

startActivity(intent1);

}

//点击后的效果实现删除后跳转页面

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

//删除界面取消的实现

**5.4设置模块的实现**

设置模块包括一些系统的设置和日程类型的设置，同时还有设置日程是否需要开启提醒功能。

系统设置：安卓系统内的设置界面由Android Preference相关类提供，使用preference这个键值对的方式来自动保存这些数据，并即时生效，同时Android提供一种类似layout的方式来进行Preference的布局。

其中，使用PreferenceScreen和PreferenceCategory对布局进行控制，

<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">

<PreferenceCategory android:title="@string/msgPreCategoryTask">

</PreferenceCategory>

</PreferenceScreen>

这种布局设置子项包括ListPreference，CheckBoxPreference，EditTextPreference 3个子项。

任务设置的布局内容，包括缺省日程类型和缺省重复类型，使用的是ListPreference 带下拉的类型的是：设置项，其中缺省日程是：

<ListPreference

android:entries="@array/categoryList" //控件的内容

android:title="@string/msgTitleSelectDefaultCategory" //控件的标题

android:key="@string/msgKeySelectDefaultCategory" //控件的key值

android:summary="@string/msgSumSelectDefaultCategory"

android:defaultValue="@string/valueDefaultCategory" //控件上的缺省选项

android:entryValues="@array/categoryList" /> //控件内容的value值

Key值是用于导引到这个设置模块，以便于从找到这个控件的值。

日程类型的设置：点击日程类型设置，跳转到类型设置的编辑列表，已有的或者系统默认的类型会直接显示在listview中，在这些内容底下放置一个添加按钮，并且每个item也就是每个已有的日程类型都是可以编辑的。如果是编辑已有的日程类型就需要对数据库进行更新操作，如果是添加新的日程类型，就需要在数据库中插入一条新的数据。

提醒功能的设置：点击提醒功能设置，跳转到显示所有日程的列表，选择需要进行提醒的日程，就会弹出一个确认对话框来确定是否需要对该日程进行提醒。提醒的功能是使用闹钟来进行提醒的。

**5.5数据库的实现**

（1）数据表

分别建立了schedule表（日程的表）和scheduletype（日程类型所需的表）

Android 为开发人员提供了标准的创建数据库的方式。建立DBOpenHelper类继承到SQLiteOpenHelper ,实现onCreate 和 onUpgrade 两个方法。

OnCreate方法是用于初次使用软件时生成数据库，onUpgrade方法是用于升级软件时更新数据库表结构。

首先是schedule表的建立，在这个表中包括日程ID、日程类型、日程的耗时、日程正文内容、日程时间这几个参数。建立时对应需要写上参数名、参数类型、主键等。

db.execSQL("CREATE TABLE IF NOT EXISTS schedule(scheduleID integer primary key autoincrement,scheduleTypeID integer,remindID integer,scheduleContent text,scheduleDate text)");

下面是scheduletagdate表的建立，在这个表中包括标记的ID、年信息、月信息、日信息、日程ID这几个参数。

db.execSQL("CREATE TABLE IF NOT EXISTS scheduletagdate(tagID integer primary key autoincrement,year integer,month integer,day integer,scheduleID integer)");

}

建立完表后是继续写入onUpgrade方法，这个方法会在数据库升级时被触发，以便于软件升级时更新表结构。

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL("DROP TABLE IF EXISTS schedule");

db.execSQL("DROP TABLE IF EXISTS scheduletagdate");

onCreate(db);

}

（2）再分别建立ScheduleVO 和ScheduleDateTag ，里面分别写了表里面各个数据的get（表示获取某个变量值）和set（表示设置某个变量值）方法，以便于调用。

（3）建立ScheduleDAO文件，这个文件是用来写数据库的查找、删除等方法。

首先是需要实例化DBOpenHelper用于数据库数据的get与set。

private DBOpenHelper dbOpenHelper = null;

public ScheduleDAO(Context context){

dbOpenHelper = new DBOpenHelper(context, "schedules.db");

}

下面就是对表中数据的操作了，包括了保存日程信息、查询某一条日程信息、查询所有的信息、删除日程、将日程标记的日期保存到数据库中等等一些数据库的操作。

（1）保存日程信息，先是从日程添加页面里获取日程ID、日程类型、日程耗时、日程正文内容、日程时间这几个参数，然后是使用ContentValues类（这个类是一个键值对，是用来存储名值对，它的名一般是String类型，而值都是基本类型）来new一个参数，用这个参数的put方法将这些个参数放入到数据库中。

values.put("scheduleTypeID", scheduleVO.getScheduleTypeID());

//放入日程类型ID

values.put("remindID", scheduleVO.getRemindID()); //放入重复类型ID

values.put("scheduleContent", scheduleVO.getScheduleContent());

//放入日程正文内容

values.put("scheduleDate", scheduleVO.getScheduleDate());

//放入日程日期信息

最后调用添加方法insert。

db.insert("schedule", null, values);//插入信息

另外需要moveToFirst方法来判断是否为空，进行相关的操作。

（2）查询某一条日程信息，由于我这里的设计是通过对日程的点击来查看当前点击日期的详细信息，即是根据日程的ID来查看的。android sqlite的查询方法是query方法，它的查询语句是：

db.query("schedule", new String[]{"scheduleID","scheduleTypeID","remindID","scheduleContent","scheduleDate"}, "scheduleID=?", new String[]{String.valueOf(scheduleID)}, null, null, null);

另外还需要的是判断语句，先用cursor.moveToFirst()判断curor是否为空值，不为空则就给cursor赋值，空着直接cursor.close()来关闭。

（3）查询所有的日程信息，由于所有日期可能不止一个，所以返回值设为数组形式。查询依旧是用query方法，这里的的where语句相关的就直接设为null值。其他的依旧与查询一条语句相仿。

（4）删除日程，这里的删除日程需要将日程及对应日期上的标记一起删除，这样才可以便于管理，删除使用的语句是delete。

db.delete("schedule", "scheduleID=?", new String[]{String.valueOf(scheduleID)});//删除日程信息

db.delete("scheduletagdate", "scheduleID=?", new String[]{String.valueOf(scheduleID)});//删除日程对应的日期标识信息

（5）将日程标志日期保存到数据库中，这里是为了将在日期上标识的信息存储到数据库中，方法同第一步的保存相仿，只是数据的不一样，而且使用的是数组形式。

for(int i = 0; i < dateTagList.size(); i++){//以for循环的方式进行

dateTag = dateTagList.get(i);

ContentValues values = new ContentValues();

values.put("year", dateTag.getYear());//放入年信息

values.put("month", dateTag.getMonth());//放入月信息

values.put("day", dateTag.getDay());//放入日信息

values.put("scheduleID", dateTag.getScheduleID());//翻入日程的ID

db.insert("scheduletagdate", null, values);//插入到表scheduletagdate中

}

（6）只查询出当前日程的日期，这个查询语句是通过当前月的年和月信息来得到日程标记上的所有信息。这里也是使用query语句进行查询的，只不过改变了里面的查询使用的参数。所以方法类同于上面所有日程的查询。

（7）当点击每一个gridview中item时,查询出此日期上所有的日程标记(scheduleID),

通过点击后得到对应日期的年月日信息，然后用query方法得到次日程标记的ID，用于后面得到想到相关的日程信息。

query("scheduletagdate",

new String[]{"scheduleID"}, "year=? and month=? and day=?", new

String[]{String.valueOf(year),String.valueOf(month),String.valueOf(day)}, null, null, null);

（8）关闭数据库。判断后直接关闭当前数据库。

if(dbOpenHelper != null){

dbOpenHelper.close();

1. **总结与展望**

经过几个月的奋斗，我的毕业设计——基于安卓的日程管理app终于完成了，系统基本达到了预期的效果，通过在手机和模拟器上的一些简单测试，基本的操作都能够正常使用。在毕业设计刚开始时，我还有些茫然，但是经过刚开始一系列的准备工作，包括查看一些参考书籍和文献、编写开题报告等等，使得我对自己的毕业设计内容有了初步的认识，慢慢的进入了状态，之后就是对安卓基本知识的学习和钻研。在这一切准备工作就绪后就正式开始了毕业设计的编写。在刚开始设计的时候，还是遇到了各种各样的问题，通过自己的钻研和指导老师的指导，每当解决一个问题，都会觉得豁然开朗，逐渐就有了信心。最终，伴随着毕业设计项目中的难点逐个被击破，终于完成了这个日程管理系统。在这整个过程中，我经历了很多，也收获了很多。而且在过程中我了解了很多以前在书本中无法学习到的知识，我发现只有自己实践才能更好地提升自己的能力，我发现光有知识是不够的，还需要与实践相结合，这样才能提高自己的专业知识和操作能力。

此项目虽然经过我的努力终于完成了，但是其中还存在不少的问题。当然，作为第一次做一个完整的安卓app，还有许多未熟悉的知识，所在开发出来的项目可能存在着问题。只有在我自己不断地熟悉与学习，并且去尝试，才能更好的完成自己以后的工作任务。虽然存在一些问题，但是这次的毕业设计，给了我们一次很好的锻炼机会。指导老师耐心的带领我们，深刻的认识了开发一个项目的完成过程，从中我学到了很多很多。

这次的毕业设计的制作我体会最深的是我们做任何事只要坚持就会成功的。刚开始做这个系统的时候我自己无从下手，当中存在着很多很多的疑问，我灰心丧气，感觉自己学了那么多的东西一样都没有用到。但是在老师的耐心指导下，我重拾信心，通过去图书馆找相关的资料，通过网上搜索相关的内容，慢慢就有了头绪。从一开始的手足无措，毫无进展，到把系统的一个个模块一点一点实现出来，我感到很高兴很兴奋。

总之，这次的毕业设计是一个不断学习、探索、发现问题和解决问题的过程。在以后的工作中还需要不段的积累，不段的努力。

# 参考文献

[1] 王向辉，张国印，沈洁.Android应用程序开发[M].北京：清华大学出版社，2010.

[2] 李金霖，赖超，龙曦 等.基于Android平台的手机日程管理系统[J].计算机与数字工程，2003.

[3] 郭霖.第一行代码[M].人民邮电出版社，2014.

[4] [David Allen](https://book.douban.com/search/David%20Allen).Getting Things Done[M].Penguin Books，2002.12.31.

[5] 邹鑫.小强升职记[M].电子工业出版社，2014.4.1

[6] 李刚.疯狂Android讲义[M].电子工业出版社，2011.6

[7] 陈璟，陈平华，李文亮.Android内核分析[J].现代计算机（专业版），2009（11）：112-115

[8] J.F.DiMarzio.Android a programmer’s Guide[M]. McGraw.Hill,2009

[9] 赵亮，张维.基于Android技术的界面设计与研究[J].电脑知识与技术，2009（29）：8183-8185

[10] Y.Daniel Liang. Java语言程序设计[M].机械工业出版社，2011.6

[11] 郝玉龙. Android程序设计基础[M].清华大学出版社，2011.10

[12] 毕广吉. Java程序设计实例教程[M].冶金工业出版社，2007

[13] 余志龙.Google Android SDK开发范例大全[M].人民邮电出版社，2009.7

[14] 靳岩 姚尚朗.Google Android开发入门与实战[M].清华大学出版社，2009.7

[15 希尔伯沙茨. 数据库系统概念[M].机械工业出版社，2008

[16] 孙金玉，陈萍.数据库原理与应用[M].清华大学出版社，2011.6

[17] Android开发文档.Android Developers[EB/OL]. developer.android.com，2016

[18] 朱少民.软件工程导论[M]. 清华大学出版社，2009.1

[19] 成先海.数据库基础与应用[M].北京:机械工业出版社，2008

[20] 孙更新.Android从入门到精通[M]. 电子工业出版社，2011.10