

本科毕业论文

|  |  |
| --- | --- |
| **题目：** | 基于Android的电子邮件客户端系统的设计与实现 |

|  |  |
| --- | --- |
| **学 院:** | 计算机科学与技术学院 |
| **专 业:** | 计算机科学与技术 |
| **学 号:** | 201413137085 |
| **学生姓名:** | 张鑫 |
| **指导教师:** | 古权 |
| **日 期:** | 二〇一八年五月 |

摘 要

经过移动互联网对中国的洗礼，智能手机已经成为人们生活中不可缺少的工具。但是管理功能众多的智能手机已经成为人们的一种负担，严重影响了使用智能手机的体验，甚至影响了人们学习生活的效率。本文探讨一种运行在Android操作系统之上的自动化定时管理智能手机功能的应用软件的实现方法。

本文介绍的应用软件实现方法将实现定时自动化管理手机中硬件、功能开关、软件等，减轻人们管理智能手机的负担。该软件主要应用了Activity、Service、AlarmManager、Broadcast Receiver等技术提供一个人机交互良好的前台控制界面以及功能丰富的后台服务，并提供了一个根据学生课程表自动化管理手机功能的特色功能。软件通过XML文件持久化存储用户配置信息，使用多种第三方开发者服务提供商的产品提高用户的产品使用体验、帮助应用软件的运营、提高应用软件的安全性。

本文实现的《定时大师》应用软件在国内多个应用商店上架，累计用户数多达数万人，一定程度上帮助用户更便捷地管理手机。并通过应用搭载的广告服务获得了一定的经济收益。

关键词：Android； 邮件客户端； 垃圾邮件；

**Abstract**

After the baptism of mobile Internet to China, smart phones have become an indispensable tool in people's life. However, the management of a large number of smart phones has become a burden for people, seriously affecting the use of smart phones experience, and even affect the efficiency of people's learning life. This paper explores the implementation of an automated timing management smartphone for applications running on top of the Android operating system.

The functions of application software introduced in this paper include timing automation management, mobile phone hardware, function switch, software and so on, which can lighten the burden of people while managing smart phones. The system provides a well-interactive front-end UI(User Interface) and a feature-rich back-end service by applying the techniques such as Activity, Service, Alarm Manager and Broadcast Receiver in android system, and the most outstanding feature is automatic management function using student curricular table. In the software XML file is used to store user configuration information, and many third-part products is also implemented to improve user experience, to help software operation, and to improve the security of software.

The "Planned Task" software in this article has been built on many domestic application stores, which has the cumulative number of users as many as tens of thousands. It can help users to manage their mobile phones more conveniently, and make economic profit by providing adverting services.

**Key words**： Android； Alarm Manager； Timer； Automation； Service

目 录

[1绪论 1](#_Toc483516356)

[2定时大师软件需求分析 3](#_Toc483516357)

[2.1软件的可行性分析 3](#_Toc483516358)

[2.2需求场景分析 3](#_Toc483516359)

[2.3软件开发的目标、功能设计 4](#_Toc483516360)

[2.4同类软件的调查 5](#_Toc483516361)

[2.5 Android手机软件的特点 5](#_Toc483516362)

[3定时大师的开发环境及技术特点 6](#_Toc483516363)

[3.1软件的开发环境 6](#_Toc483516364)

[3.2软件的主要技术特点 6](#_Toc483516365)

[4定时大师的概要技术方案 8](#_Toc483516366)

[4.1软件的功能模块设计 8](#_Toc483516367)

[4.2用户界面模块及概要的功能 8](#_Toc483516368)

[4.2.1视图层及控制器层 8](#_Toc483516369)

[4.2.2模型层 10](#_Toc483516370)

[4.3后台服务模块概要技术设计 10](#_Toc483516371)

[4.3.1 TaskService后台计算组件 10](#_Toc483516372)

[4.3.2 TaskReceiver广播接收组件 11](#_Toc483516373)

[4.4软件的启动技术设计 11](#_Toc483516374)

[5定时大师的详细设计 12](#_Toc483516375)

[5.1用户界面模块设计 12](#_Toc483516376)

[5.1.1定时器列表TaskListFragment 12](#_Toc483516377)

[5.1.2定时器详情TaskDetailFragment 13](#_Toc483516378)

[5.1.3设置定时器任务OperationListActivity 13](#_Toc483516379)

[5.1.4课表模式界面LessonFragment 15](#_Toc483516380)

[5.1.5设置课程表界面LessonTableActivity 15](#_Toc483516381)

[5.1.6更多界面MoreFragment 16](#_Toc483516382)

[5.2用户界面模块模型层设计 17](#_Toc483516383)

[5.2.1数据结构详细设计 17](#_Toc483516384)

[5.2.2数据持久化 20](#_Toc483516385)

[5.4后台服务模块的详细设计 22](#_Toc483516386)

[5.5第三方开发者服务 24](#_Toc483516387)

[6总结 26](#_Toc483516388)

[参考文献 27](#_Toc483516389)

[致 谢 28](#_Toc483516390)

# 1绪论

在经历了多次工业革命之后，生活在现代社会之中的人们，生活节奏是人类发展历史上最快的。为了适应节奏紧张的现代社会，人们越来越需要自动化，智能化的软硬件来为人们的生活提供便利的同时，切实地提高工作和学习效率。

在信息时代的今天，特别是经过了移动互联网洗礼的中国，智能手机已经得到普及，并深刻地改变了人们的生活方式。人们用智能手机通讯、收发邮件、打车、叫外卖、听歌、安排日程等等，手机已经成为工作生活不可缺少的工具，帮助人们极大地提高了工作、生活的便捷和效率。

但是随着智能手机功能越来越多，管理智能手机以及安装在智能手机上的软件也已经成为了一种困难。研发一款可以为人们提供多样化的定时控制管理手机功能的应用，将人们从繁复的控制手机操作之中解放出来，将极大地提高其使用手机的便捷性。这就是本次毕业设计的任务。

Android操作系统是目前世界上智能手机占有率第一的手机操作系统，本次毕业设计是基于谷歌的Android操作系统，开发一款通过可以定时自动化控制管理用户手机内各种硬件功能，设置和软件开启关闭的手机软件。并在次功能的基础之上，深入学生生活中的课堂场景，开发适用于课表的上下课定时自动化控制手机工具软件。Android操作系统为主要开发者提供如下四大组件：

1. Activity。通常表现为一个单独的窗口界面，是所有Android程序与用户交互的入口。它能为用户提供可视化的人机交互方式，提供基于图形事件触发程序逻辑的功能。每个Activity拥有自己的生命周期，通常使用Activity的生命周期管理整个应用程序在手机内存中的生命周期。Activity之间使用Intent实例进行通信，可以传递参数等信息，用于控制应用程序界面逻辑等。
2. Service。通常用于应用程序在手机后台运行耗时、复杂或者为应用程序提供基础服务功能的任务，比如网络传输、图片视频处理、文件读写、复杂计算等。Service运行于单独的线程中，不需要交互，也没有图形界面，不会影响前台界面线程的运行。它有两种启动方式，一是由其他组件调用startService()函数，调用产生的Service实例将脱离调用者生命周期的控制，一直运行直到自己主动退出或者被其他组件主动调用stopService()退出；二是通过bindService()启动，产生的实例会绑定调用者的生命周期，调用者退出，自己也会跟随退出。
3. Content provider。该组件为Android系统中多个应用程序间共享数据提供了方便，为其提供统一的获取和保存数据的接口。
4. Broadcast receiver。Android系统中存在一个事件通知的机制，当设备出现一些事件变化如有电话呼入、无线网络可用、开启屏幕等的时候，系统将发送一个广播，将发生的事件推送至注册了该类事件的应用程序中。Broadcast receiver则是应用程序中用于接受该广播的组件。应用程序可以使用Broadcast receiver订阅感兴趣的事件，可以在应用程序运行时动态注册，也可在程序安装配置文件中静态地声明。其区别是动态注册只在应用程序运行在内存中时有效，而静态注册则无需应用程序运行。

# 2定时大师软件需求分析

需求分析是软件开发中必不可缺的一个环节，其决定了软件开发的目标，实现的功能，使用的群体。软件开发进行前，需要细致地分析用户需要哪些功能，在哪些场景下使用，功能实现的技术难度。这样才能保证软件开发的可行性、实用性。

## 2.1软件的可行性分析

人们在工作生活中，每天变换不同的场合，切换不同的场景，如果因此而需要频繁的变更手机的设置，将极大的影响人们的工作生活效率，还可能因为遗忘造成更大的影响。因此，本软件的设计开发是十分有意义，可以让智能手机更智能。

可行性分析是为了分析开发软件是否从经济、技术、运行、法律等多方面是否能够实现并能为使用者带来切实的好处。根据需求和技术方案设计进行研究得出以下可行性分析：

1. 经济可行性：虽然本项目为个人毕业设计，无需过多追求经济效益，但本项目也将通过植入广告的方式获取收入，能够得到一定的经济收益。
2. 技术可行性：项目基于Android操作系统开发，该系统已拥有数以万计的开发者，开发途中遇到的问题在其开发者社区中应该能够搜索到大量的技术文档。利用该系统开发的软件也已在人们中普及。安卓的四大组件及系统权限机制从原理上可以实现本软件需求。
3. 运行可行性：项目运行于Android系统之上，通过Activity提供交互界面，Service提供后台任务的执行环境、Broadcast receiver注册关注事件。原理上可以将需求中的功能运行在Android操作系统之中。
4. 法律可行性：本软件不涉及非法信息传播，所使用的技术均为操作系统底层提供Api，谷歌为开发者免费提供系统Api调用，不存在侵权。

通过以上的可行性分析，我将在Android操作系统中构建《定时大师》应用程序，实现定时自动化管理手机的功能。

## 2.2需求场景分析

软件的开发需求需要紧贴用户的需求，而用户在不同的场景需要不同的功能，针对定时自动化管理手机分析得到的主要场景和功能有：

1. 人们每天的生活中使用各种手机功能，控制手机完成各种各样的任务的行为中许多都有一定的规律,比如上课和开会时关闭铃声、睡觉时关机或者开启飞行模式等。设计定时模式控制手机功能。
2. 课堂经常出现手机没有调成静音，各种铃声时常破坏工作学习氛围的情况。通过学生的课表自动设置手机静音时段，可以提高大家的学习及工作效率。设计课表模式，根据课程表执行定时任务。
3. 人们在生活中经常需要记录一些有意义的音频、照片或视频，但是录音，拍照，录像一般只能手动开启和关闭，用户需要时刻关注手机的状态，非常不便。定时观察事物，记录生活，安全监控的需求得不到满足。设计定时自动录音，拍照，录像功能。可以在手机休眠的状态下静默执行并保存
4. 如今智能手机功能众多且繁杂，用户如需开启手机的多项功能就必须进行多次繁杂而重复的操作。设计多任务执行方案，降低了用户使用手机的繁杂程度。
5. 如今智能手机电量满足不了人们高强度的使用，而手机的各种硬件功能的开启也是造成电量浪费的一大因素，因此最好有一款能自动开关手机硬件功能的软件。方便管理的同时还能节省电量。

## 2.3软件开发的目标、功能设计

本软件基于谷歌Android系统开发，使用系统提供的多种组件和多个Api，实现较好的交互界面、较便捷的操作体验、较强大的功能、较高的可用性。软件的开发目标和功能设计主要有：

1. 通过软件定时自动化控制管理用户手机内各种硬件功能，设置和软件，减轻用户每天频繁控制管理手机的负担，避免用户因忘记而导致的影响生活工作学习效率的情况。学生老师可使用课表模式实现课程前后的自动化控制管理手机设置等任务，避免手机的铃声，软件后台流量，硬件功能开启而导致的影响课堂氛围，浪费手机流量，浪费手机电量等情况，达到维持课堂良好学习氛围提高学习效率的目标。
2. 可设定定时执行以下功能：关机、重启、开关飞行模式、开关WIFI、开关移动数据连接、开关GPS、开关蓝牙、闹钟、调整情景模式（响铃、静音震动、静音）、开关自动亮度、运行特定程序、录音、拍照、录像、发送短信、拨打电话。
3. 软件特有的课表模式可通过用户设定的课表以及上下课时间，自动根据当前周数和课程所开周数自动在上课前和下课后同时执行用户所设定的上述2.中的多个任务。
4. 拍照，录像功能实现不开启屏幕，无预览，锁屏状态下后台运行及保存。
5. 实现简单易用，用户体验良好的界面。
6. 实现用户数量统计，用户反馈，自动更新，第三方分享等功能。
7. 兼容多个版本的Android系统，兼容多种不同的分辨率。

## 2.4同类软件的调查

目前，基于移动客户端应用的开发市场竞争异常激烈，各具功能的应用层出不穷，令人眼花缭乱，但是通过对目前市场上常见的同类应用的分析，尚未有可以根据用户个性化需求进行设置自动管理的应用：例如根据用户的课程表信息自动设定系统内功能的打开或关闭等设置；根据用户开会时间自动设定相关功能；根据日程信息自动设定等；

另外，市场上同类应用的用户界面不够友好，或者功能繁杂，使用不够简便，对用户的专业知识要求过高；或者应用的版本更新滞后，不能适应新的安卓官方系统，也不能适用第三方深度定制的安卓系统，造成很多应用的健壮性、可用性较差。

## 2.5 Android手机软件的特点

Android手机软件不同于运行于桌面电脑的软件，手机在硬件方面与桌面电脑有着较大的差异：

1. 手机屏幕较小，一般只有4-6英寸，需要较好的交互设计体验。
2. 手机处理器的性能较弱，不能运行复杂计算任务，且长时间处理计算任务会给机身产生较高的温度，会造成使用者的不适。
3. 手机电量有限，应用程序需尽可能地节省手机电量；
4. Android手机有多个系统版本，版本之间存在Api不兼容的问题，需要在程序逻辑中根据系统版本运行不同的代码。
5. Android手机有多种不同的屏幕分辨率，在界面上同样需要适配多种主流的屏幕分辨率，避免界面变型。

# 3定时大师的开发环境及技术特点

## 3.1软件的开发环境

Android系统版本分裂较为严重，兼容性只能倾向于市场占有率较高的几个版本做适配，部分低版本因系统陈旧等原因只能在功能、界面、交互等方面做缩减，否则将带来无法估量的技术成本和时间成本。

因此本软件选择的开发环境如下：

1. Windows 7操作系统。
2. Java 7 SDK[1]。
3. Eclipse 作为开发IDE。
4. 基于Android 5.0 SDK开发，因为现在Android操作系统版本占最大市场份额的正是Android 5.0。
5. 兼容Android 2.3及以上版本操作系统。但在Android 4.2操作系统以下应用程序界面、交互、功能会有所缩减。

## 3.2软件的主要技术特点

1. 界面大量使用Activity托管Fragment，使其实现左右滑屏切换界面的功能，拥有较好的用户体验，保持了Fragment跟Activity的独立性为以后的拓展更新奠定基础，为使安卓3.0以下系统版本正常使用Fragment，使用Support.V4支持库[2]。
2. 大量使用ListView展现设置项目，点击后的弹出的界面大多为Dialog对话框，需要托管Fragment而无法使用Dialog对话框的则使用改写成Dialog对话框的形式的Activity，使界面层次分明，易于理解，拥有较好的用户友好度。
3. 使用Canvas绘制任务列表中含有多个任务的Task对象的图标（ImageView），使Task对象中所设置的多个任务均可在列表中看到所代表其的图标，而不需要进入详情即可清楚了解用户所设置的Task对象。实现友好简单明了的用户体验。
4. 使用第三方开发者服务提供商所提供的自动更新，后台用户数量监测，用户反馈，第三方分享等SDK使用户得到更好的体验，开发者也能更接近用户。
5. 模型层定义：Task、one\_class、DaySchedule、TaskLab类，前三个分别为：任务组类、一节课类、一天的课程类，TaskLab则拥有以上3个类的ArrayList实例，控制层所有对模型层的操作均通过TaskLab类执行。
6. 因数据量不大，故程序的数据使用XML文件[3]持久保存，用户设置相关数据保存则使用安卓SharedPreferences实现。
7. 定时的实现使用安卓系统AlarmManager服务，利用该服务可在设定的时间发送特定的广播，使用BroadcastReceiver接收AlarmManager定时发送的由自定义的Action实例化的PendingIntent，并由BroadcastReceiver启动后台Service获取Task对象实例并执行。之所以不直接使用Service接收AlarmManager发送的PendingIntent是因为发现在部分深度定制的安卓系统中厂商为优化性能限制应用后台自动启动，使用Service直接接收AlarmManager发出的PendingIntent会产生较大的时间延迟。
8. 使用BroadcastReceiver接收因开机或日期改变系统自动发出的broadcast intent，更新模型层中的时间相关实例并重新设定定时任务。
9. 后台拍照需实现自动对焦自动快门，故需调用相机API实现，该API只能得到固定方向的照片。对于照片的旋转角度问题，在快门前使用OrientationEventListener获取手机的重力方向，拍照后使用Matrix根据之前获取的重力方向对得到的Bitmap进行旋转，得到用户摆放手机方式所应得到的照片。使用Handler.postDelayed(Runnabler, long delayMillis)方法实现间隔一定时间自动拍照。
10. 后台录像需实现无预览录像，安卓对其存在检测机制故一般情况很难实现在后台运行录像。因此将录像预览放在一个显示大小很小以至于用户发现不了的Activity中，录像使用单独线程运行，在相机初始化开始录像后立即结束放置预览的Activity，因安卓已完成检测预览画面的工作并且录像由单独线程运行，故此时已经实现了真正的后台录像，而不阻碍前台的其他工作。由Handler.postDelayed(Runnabler, long delayMillis)方法负责在设定时间后运行一个线程停止录像。启动时是时间极短且不可见的前台启动，初始化完成后的录像却是在后台，实现了启动时是“后台启动”的假象。
11. 研究各官方版本和厂商定制版本的安卓系统中AlarmManager服务的区别（如MIUI 5分钟才检查一次AlarmManager，导致定时时间短于5分钟时产生时间误差的几率增大）以及对安卓官方API的支持程度，可用性，健壮性。
12. 尝试通过ROOT权限获取更多安卓系统控制权以便执行更多种类更高权限的任务，研究安卓权限安全隐私安全问题。
13. 使用开发者服务提供商的云测试功能以及一定数量的各品牌各系统版本手机验证软件的可靠性，健壮性和体验的一致性。

# 4定时大师的概要技术方案

## 4.1软件的功能模块设计

软件分为两大模块：用户界面模块和后台服务模块。

用户界面模块分为三层：视图层、控制器层、模型层。其中视图层提供用户交互界面，控制器层负责响应用户的操作事件，模型层提供底层数据的存储模型。

后台服务模块负责旁路处理用户的定时任务。负责较复杂的计算操作等。

模块、操作系统、模块内各层及组件的关系如下图4.1所示。

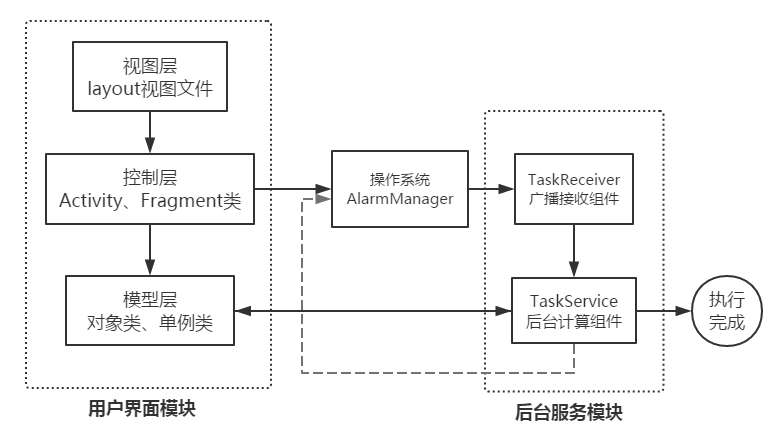


图4.1模块设计

## 4.2用户界面模块及概要的功能

### 4.2.1视图层及控制器层

视图层主要由Android操作系统生成界面类实例，并解析视图的layout文件，由此生成相应的图形界面显示在屏幕上。

控制器层主要由界面类内的事件监听器组成，监听器监听视图界面的事件并做出相应的处理。即每个界面由各自界面的控制器组件响应用户的操作。

软件提供两类定时器，一是普通定时器，位于定时器列表中；二是课表模式定时器，是两个特别的定时器，单独显示在课表模块界面，不在普通定时器的列表中显示。

如下图4.2所示是软件主要视图及视图跳转关系。

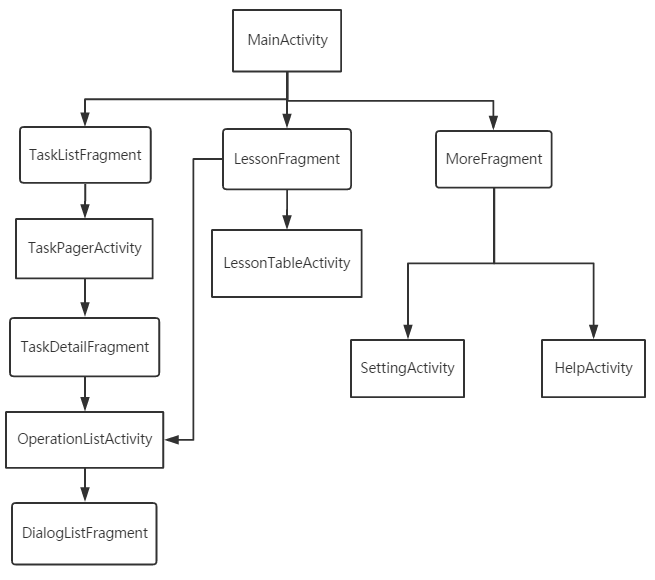


图4.2界面模块

1. MainActivity：应用程序的主界面，托管三个Fragment，提供左右滑动切换子界面的功能。这三个自界面分别为：TaskListFragment、LessonFragment、MoreFragment。
2. TaskListFragment：MainActivity的第一个子界面，用于显示普通定时器的列表。显示定时器的概要信息，点击定时器进入TaskPagerActivity可以查看定时器详情，列表支持多选、删除等操作。
3. TaskPagerActivity：没有实际界面，只作为一个Activity用于托管多个任务的详情Fragment--TaskDetailFragment。主要是提供左右滑动切换任务Fragment的功能。
4. TaskDetailFragment：真正显示定时器详情信息的子界面，托管于TaskPagerActivity中，由其提供左右滑动切换定时器Fragment的功能。
5. OperationListActivity：仅有一个界面的边框，界面的中心区域用于显示被其托管的Fragment，主要提供左右滑动切换DialogListFragment的功能，并额外提供一层透明的边框。
6. DialogListFragment：是定时器支持的操作列表，供用户设定定时器所执行的操作动作。用户可勾选所需执行的操作，然后点击“保存”以修改定时器执行的操作。
7. LessonFragment：MainActivity的第二个子界面，是“课表模式”功能的设置界面。在该界面可以通过点击“课前定时器”和“课后定时器”图标进入OperationListActivity修改定时器所执行的操作。同时能通过本界面进入到课表详情的设置界面LessonTableActivity。
8. LessonTableActivity：该界面显示一个课表的图形界面，用于给用户自定义设置课表。用户可以选择具体的星期几和具体的第几节课，点击设置课时和开课的周数，以便软件通过周数判断是否当前有课。
9. MoreFragment：MainActivity的第三个子界面，展示一些软件附属功能的入口以及软件的版本信息等。
10. SettingActivity：提供应用程序的设置选项界面。
11. HelpActivity：提供应用程序的使用说明、用户帮助等。

### 4.2.2模型层

模型层主要由数据结构抽象类组成，在内存中使用单例模式存储用户数据。本软件设计如下几个数据结构类存储数据：

* 1. 定时器类。存储定时器对象的相关配置信息。
  2. 课程类。表示一个抽象的课程，存储课程时间信息。
  3. 一天的课程类。表示一天的课程集合。
  4. 模型层单例类[4]。存储所有的模型层数据，提供访问模型层的接口。

## 4.3后台服务模块概要技术设计

### 4.3.1 TaskService后台计算组件

定时器执行时需要调用多个操作系统接口以及较为复杂的计算，并且作为一个常驻系统的后台守护进程，不提供前台界面。因此后台计算组件使用Android系统中的Service组件设计。

该组件的功能包括如下几点：

1. 通过定时器id查找模型层获取定时器对象信息。
2. 判断定时器所需执行的操作类型。
3. 执行定时器的一个或多个操作任务。
4. 根据定时器配置计算下一次执行定时器的时间。
5. 下发定时广播给操作系统。
6. 将修改后的模型对象提交至模型层保存。

### 4.3.2 TaskReceiver广播接收组件

TaskService或用户界面模块的控制器层通过Android系统的AlarmManager组件设置定时器，AlarmManager组件到达指定时间将会发送Broadcast广播[5]，本软件的TaskReceiver组件将会监听到该广播，进而调用TaskService执行定时器任务。

## 4.4软件的启动技术设计

应用程序启动需要初始化非常多的运行数据、业务数据、基础组件等，本软件的启动逻辑分为如下几部：

1. 应用程序向系统注册开机自动启动TaskService后台服务。
2. 点击应用图标启动应用程序。
3. 由操作系统调起AppStartActivity。因为应用程序在Android操作系统中配置的启动Activity为AppStartActivity。该Activity会初始化应用程序的一些基础组件和数据，如软件的运行次数、统计组件、网络组件等。
4. 软件的运行次数统计用于判断是否首次运行本软件，如果是则跳转至WelcomeActivity使用向导页面，显示软件的基本操作方法等信息。
5. 如果不是首次运行本软件，则读取配置文件，载入定时器信息和课表信息，调起MainActivity进入应用程序主界面。
6. MainActivity检查后台服务TaskService是否已启动，如未启动则将后台服务拉起。

# 5定时大师的详细设计

## 5.1用户界面模块设计

### 5.1.1定时器列表TaskListFragment

在技术实现上：

1. 该界面使用了ListView控件[6]，并实现了一个自定义的TaskAdapter类。ListView通过重载的getView方法得到Item界面，getView方法则通过重载的SetView方法生成自定义的Item界面并返回。由此实现ListView每一个Item都是一个自定义的view视图，从而显示定时器的概要信息。
2. 使用Canvas绘图技术能够将定时器的多个操作示意图标合成一个图标，显示在列表项左侧的操作示意图中。

功能上该界面显示当前用户所保存的定时器条目,每个定时器保存有所执行的功能(单一任务可设置多个功能)、执行时间、重复类型、是否振动、铃声、备注信息。列表的每个任务左侧图标显示任务设定的功能示意图，当一个任务设定了多个功能时，软件按照设定功能的数量决定多个功能图标拼合的方式。功能示意图将显示完所有已选功能图标。

长按“任务列表”中的条目会进入多选模式，选定的条目将会变更背景颜色，点击右上角的箭头状图标会立即执行所选定的条目任务，点击旁边的垃圾桶状图标会将选定的任务条目删除，点击左上角的勾型图标或按下返回键退出多选模式。

如图5.1所示是定时器列表界面。简而言之，普通定时器列表界面提供如下功能：

1. 任务列表概要信息查看
2. 定时器的开启关闭
3. 定时器的立即执行功能入口
4. 定时器的删除功能入口
5. 新增定时器的入口
6. 定时器的开启和关闭

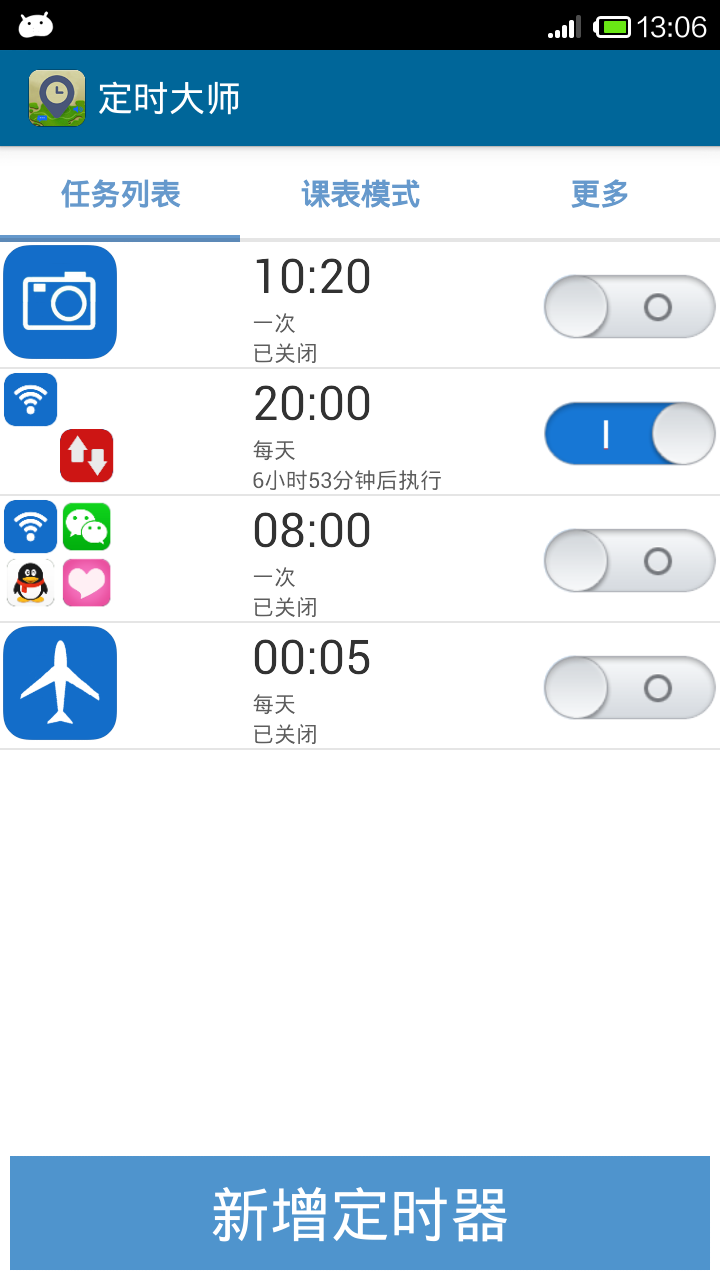
**** 

图5.1定时器列表界面 图5.2定时器详细设置界面

### 5.1.2定时器详情TaskDetailFragment

在定时器列表界面界面，点击某一个定时器或“新增定时器”按钮，显示如下左图界面，界面中有“任务”、“重复”、“铃声”、“震动”、“日期”、“备注”以及左下方的时刻选择器设置。点击相应设置栏按提示进行设置。

左下方时刻设置可通过上下滑动选择定时器设定的的时分。

设置好相应设置后点击右下方圆形按钮即可保存或新建任务，并在“任务列表”界面显示相应的状态。

定时器详细设置界面如上图5.2所示。定时器详细设置界面提供如下的多个功能：

1. 显示定时器的设置
2. 提供修改定时器设置的功能入口

### 5.1.3设置定时器任务OperationListActivity

在技术上该界面实现了一个Dialog对话框样式的Activity，原因是该界面从交互上看使用对话框体验更佳，但是技术上该界面需要托管Fragment，因此必须是一个Activity[7]。实现方式是自定义一个Dialog主题，并在应用的AndroidManifest.xml文件中配置该Activity的主题为该自定义Dialog主题[8]。

主要代码如图5.3所示：

|  |
| --- |
| <style name="dialogcustom" parent="AppTheme">  <item name="android:layout\_width">wrap\_content</item>  <item name="android:layout\_height">wrap\_content</item>  <item name="android:windowNoTitle">true</item>  <item name="android:windowIsFloating">true</item>  <item name="android:windowAnimationStyle">@+android:style/Animation.Translucent</item>  </style> |

图5.3主题文件styles.xml

在功能上，设置定时器任务界面如图5.4所示。进入定时器详情界面后，点击“任务列表”中的定时器或者点击“新增定时器”进入“设置定时器”的详情界面，在该界面，用户可以查看或设置任务的具体条目（可左右滑动切换不同任务的详情页面）。点击“任务”条目弹出一个类似对话框的功能选择界面，该界面有“硬件”、“情景”、“多媒体”三个分页，左右滑动页面即可切换分页，在此界面可选择任务所执行的功能，支持多选。



图5.4设置定时器任务

### 5.1.4课表模式界面LessonFragment

课表模式界面如图5.5所示。在主界面中向左滑动，或者点击上侧导航栏的“课表模式”，即可切换至课表模式界面。“课表模式”中的“课前执行”任务在上课前3分钟会自动执行，“课后任务”会在下课后3分钟自动执行，点击任务条目可弹出“5.1.3设置定时器任务OperationListActivity”界面，在此界面可自定义任务执行的功能。点击开启课表模式需设置好当前周数，以及设置好详细的的课程表。



图5.5课表模式界面

### 5.1.5设置课程表界面LessonTableActivity

点击“课表模式界面”中的“设置课程表”按钮进入“设置课程表”界面，设置课程表界面如图5.6所示。该页面显示了当前设置的当前课表信息。点击左侧边缘“课程节数”列表弹出课程时间设置对话框，在此页面可设置课程的时间，时间使用24小时制。

点击“设置课程表”页面中 对应星期数的对应节数的区域，弹出“设置周数”页面，该页面需至少输入一组上课周数，若课程存在间隔周，可输入最多三组上课周数。如果课程时间不规律（比如今天第一节课10点下课，明天第一节课却是11点下课），请设置每个课程独立的课程时间。

点击如图5.6的设置课程表界面的左侧时间，弹出如图5.7所示的设置课程表时间界面，该界面可以设置每节课的默认上下课时间。

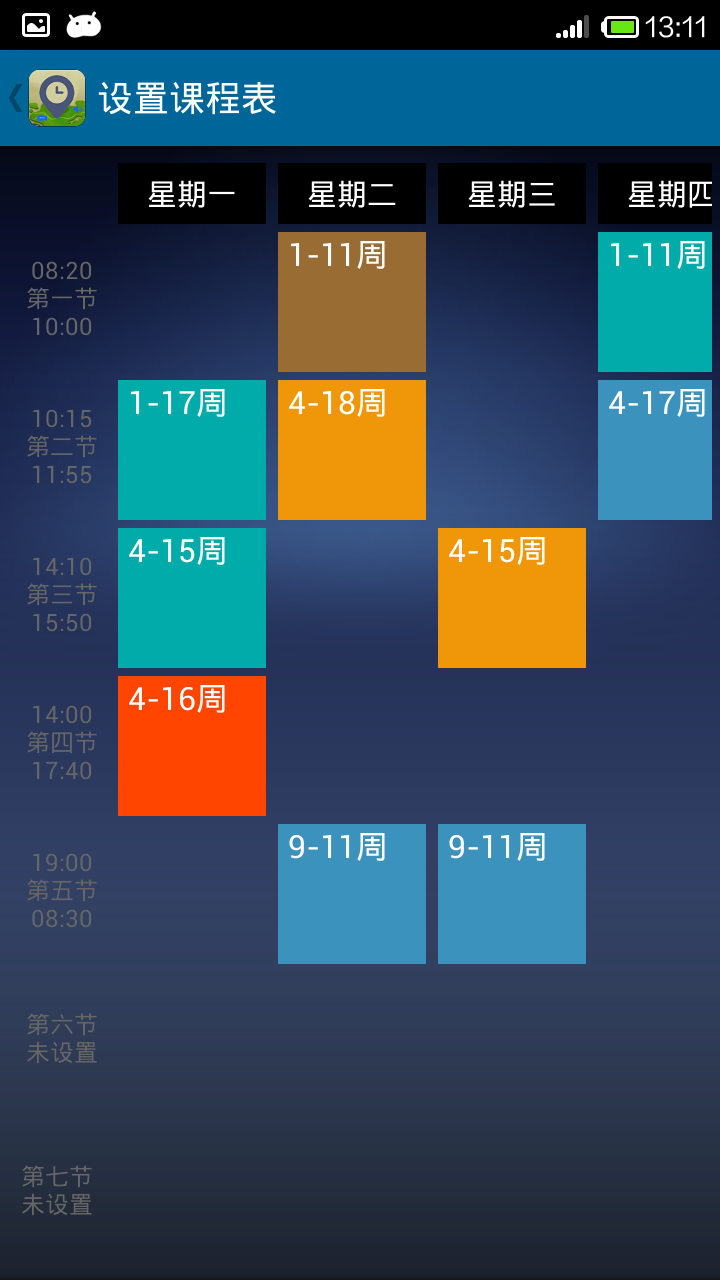
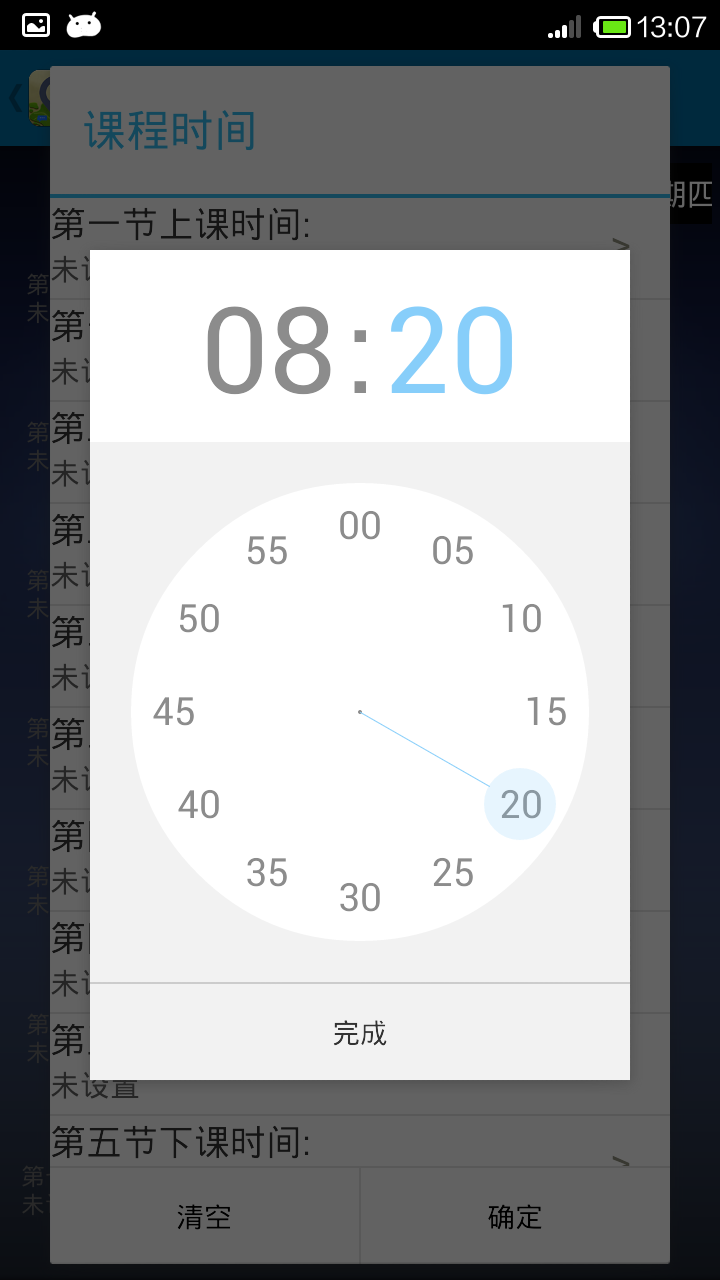
** **

图5.6设置课程表 图5.7设置课程表时间

### 5.1.6更多界面MoreFragment

“更多”界面如图5.8所示。主界面第三个页面“更多”界面中有“设置”、“偷拍模式”、“用户反馈”、“检查更新”、“帮助”、“分享给好友”、“更多好玩的应用”（特定条件下出现）。“设置”中的“拍照服务运行中，点击关闭”和“录像服务运行中，点击关闭”按钮只有在相应服务正在运行时（拍照，录像任务没有执行完毕，或是出现问题，无法继续）才可点击，否则呈现无法点击的灰色状态。进入“用户反馈”界面，可发送反馈消息给开发者，开发者回复后打开应用会有相应提示。“检查更新”将检查服务器是否有可用更新版本。“帮助”包含了软件的一些说明性内容。“分享给好友”提供微信朋友圈、QQ空间、新浪微博、腾讯微博的免登陆分享功能。“更多好玩的应用”显示推荐应用，用户可以选择自己感兴趣的应用下载安装使用。



图5.8更多界面

## 5.2用户界面模块模型层设计

### 5.2.1数据结构详细设计

1. Task类：定时器的数据结构对象。该类存储了定时器的配置信息。类结构如图5.9所示。其变量功能及含义如下：
2. mId是定时器的唯一标识，通过id可以查找到具体一个定时器。
3. mOperation存储了一个定时器所需要执行的所有操作的字符串资源id。
4. isOn存储该定时器是否为启用状态。mDate存储该定时器下一次执行的时间。 mRepetition存储定时器的重复设置，如每周一、每天、一次等。
5. mBellurl存储定时器执行时的铃声资源id。
6. isShake存储定时器执行时是否震动。
7. mTitle存储定时器的备注信息。
8. mRunable变量当定时器设置了定时启动其他应用时存储需要执行的软件包名。
9. recording\_time变量当定时器设置了录音时，存储录音时间。
10. Camera变量为定时拍照、录像时使用，变量按照一定的格式存储配置信息，具体格式为kind(整型)/time(浮点型)/num持续时间(长,分)/kind(录像整型)/time(浮点型)/quality(整型)。
11. SMSnum，SMScontent变量分别是设置定时短信时使用，分别存储手机号码和短信内容。
12. PHONEnum变量是设置定时拨打电话时用于存储手机号码。
13. Special变量用于标识该定时器是普通定时器还是课程模式定时器。
14. Website变量用于设置定时打开网页时存储网址。
15. mRunabletemp，mOperationtemp用于用户修改时临时存储使用，不会持久化存储。

|  |
| --- |
| public class Task  {  private int mId; //id  private String mOperation; //任务类型:/  private boolean isOn; //是否开启  private Calendar mDate; //设定时间/  private String mRepetition; //重复类型:/  private String mBellurl;//铃声url地址/  private boolean isShake;//是否震动/  private String mTitle; //标题备注/  private String mRunable; //定时运行的程序 包名,信息  private float recording\_time; //录音时间  private String Camera;  private long SMSnum;//发送短信号码  private String SMScontent;//发送短信内容  private long PHONEnum;//拨打电话号码  private int Special;//是否为特殊Task,0为普通  private String Website;  private String mRunabletemp; //临时变量,不存入XML文件  private String mOperationtemp; //临时变量,不存入XML文件  } |

图5.9 Task类

1. one\_class：课程类，class\_on和class\_off分别保存课程的上下课时间，a\_class链表保存了可曾开课的周数。

类结构如图5.10所示。

|  |
| --- |
| public class one\_class  {  private ArrayList<Integer> a\_class;  private Calendar class\_on,class\_off;  } |

图5.10 one\_class类

1. DaySchedule类：一天的课程类。保存一天当中所有的课程对象。

类结构如图5.11所示。

|  |
| --- |
| public class DaySchedule  {  private ArrayList<one\_class> day;  } |

图5.11 DaySchedule类

1. TaskLab类：定时器对象和课程对象的存储池。单例模式对象。控制层所有对模型层的操作均通过TaskLab类执行。类结构如图5.12所示。
2. mTasks保存了所有的普通定时器。
3. sTaskLab静态变量，用于实现单例模式。
4. class\_beforeTask用于存储课前定时器信息。
5. class\_afterTask用于存储课后定时器信息。
6. Schedule用于存储一周的所有课程信息。
7. classTimeArrayList用于存储课程表上下课时间。
8. is\_has\_read用于存储是否已加载完持久化数据。

|  |
| --- |
| public class TaskLab  {  private ArrayList<Task> mTasks;  private static TaskLab sTaskLab;  private static Task class\_beforeTask=null;  private static Task class\_afterTask=null;  private static ArrayList<DaySchedule> Schedule;  private static ArrayList<Calendar> classTimeArrayList;  private static boolean is\_has\_read=false;  } |

图5.12 TaskLab类

### 5.2.2数据持久化

用户使用过程中设置的定时器、课表等信息运行时会暂时存储在内存中，但是应用程序退出内存空间会被回收，内存断电也会使内存中的数据丢失。因此需要将数据持久化到非易失性存储体中，使应用关闭或设备断电后仍能获取之前存储的信息。

将内存中的数据模型转换为存储模型,以及将存储模型转换为内存中的数据模型就叫做数据持久化。

本软件使用XML文件的持久化方式保存用户的配置信息、定时器配置、课表信息等。具体数据的持久化方式如下：

1. 应用运行配置：使用Android操作系统的SharedPreferences组件Api保存用户配置。该方式底层实现依然为XML文件持久化。

示例如图5.13：

|  |
| --- |
| <?xml version='1.0' encoding='utf-8' standalone='yes' ?>  <map>  <int name="start\_num" value="31" />  <boolean name="is\_toupai\_on" value="true" />  <long name="this\_week\_calendar" value="1495382400865" />  <boolean name="isLessonmode\_on" value="false" />  <long name="JIFEN\_STRING" value="166" />  <long name="qiandao\_calendar" value="1495474097011" />  <boolean name="2014\_11\_08" value="false" />  <int name="this\_week" value="1" />  <boolean name="notify" value="true" />  <boolean name="first" value="false" />  </map> |

图5.13 用户配置持久化

1. 用户的普通定时器配置：逻辑代码将任务实例序列化为XML格式保存在文件中。

定时器持久化的示例如图5.14：

|  |
| --- |
| <task>  <Id>3990</Id>  <Operation>R.string.wifi\_onR.string.Data\_connection\_off</Operation>  <isOn>false</isOn>  <Date>1495540800152</Date>  <Repetition>R.string.once</Repetition>  <Bellurl>R.string.noon</Bellurl>  <isShake>true</isShake>  <Title></Title>  <Runable></Runable>  <recording\_time>-1.0</recording\_time>  <Camera>0/1/1/0/1/3</Camera>  <SMSnum>-1</SMSnum>  <SMScontent></SMScontent>  <PHONEnum>-1</PHONEnum>  <Special>0</Special>  <Website>http://www.baidu.com</Website>  </task> |

图5.14 定时器持久化

1. 课表模式定时器配置：与普通任务的存储结构一致，但是保存在单独的XML的节点中。
2. 用户设置的课表模式设置信息：使用自定义结构的文本格式保存在XML文件中。程序逻辑代码中按约定格式读取写入，具体如下。

myclass节点存储了课程的信息，每节课程能够设置三组开始和结束的周数，并能设置单独的上下课时间。具体的格式为：开始周数/结束周数/开始周数/结束周数/开始周数/结束周数/上课时间/下课时间，并以此重复得到一周的课程信息。例如如图5.15的第一段1/5/0/0/0/0/1495468800658/1495468800658表示星期一的第一节课的信息，1/5表示从第一周到第五周有课，/0/0/0/0表示另外两组周数未配置，1495468800658/1495468800658表示配置的上下课时间戳。

myclass\_time节点保存一天中每节课的默认的上下课时间戳。使用“/”分隔每个数值。例如图5.15所示myclass\_time第一段1495498801660/1495505401660表示第一节课的默认上课时间戳和下课时间戳。

课表持久化示例如图5.15：

|  |
| --- |
| <myclass>1/5/0/0/0/0/1495468800658/1495468800658/0/0/0/0/0/0/1495468800658/1495468800658/0/0/0/0/0/0/1495468800658/1495468800658/0/0/0/0/0/0/1495468800658/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800659/1495468800659/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660/0/0/0/0/0/0/1495468800660/1495468800660</myclass>  <myclass\_time>1495498801660/1495505401660/1495468831660/1495468831660/1495468831661/1495468831661/1495468831661/1495468831661/1495468831661/1495468831661/1495468831661/1495468831661/1495468831661/1495468831661</myclass\_time> |

图5.15 课表信息持久化

## 5.4后台服务模块的详细设计

用户通过前台安卓交互界面设置定时器，控制层将界面传进来的修改提交到模型层中并给Android操作系统设置定时广播，Android操作系统会在设定的时间点向程序的Receiver模块发送广播[9]，Receiver模块接收到广播会通知后台服务Service，Service从模型层中查找该定时器，获取其执行的操作，并实施操作。

后台服务模块结构图如下图5.17所示。

TaskService处理流程图如图5.18所示。

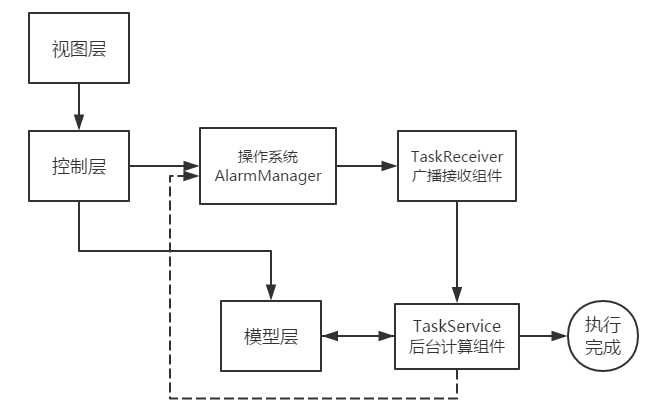


图5.17软件的后台服务设计

1. 控制层主要是界面的事件监听器逻辑，对界面的操作会调用事件监听器的代码。控制层将对象模型的修改提交至模型层保存，同时将定时器id传递给操作系统的AlarmManager服务并设定定时广播[10]。
2. AlarmManager服务是Android系统中自带的一个定时服务。当操作系统时间到达设定的时间点时，AlarmManager服务会发送一个自定义的广播，并携带一定的参数[11]。
3. TaskReceiver是软件中负责接收AlarmManager服务发送的自定义广播的模块。TaskReceiver接收到广播后会将参数放进Intent中，发送至TaskService后台服务[12]。
4. TaskService后台服务将Intent中的定时器Id取出，到模型层中搜索定时器对象。得到定时器对象后，根据对象的数据结构获取定时器所需执行的操作，并将其一一执行。
5. TaskService后台服务将定时器执行完毕后，还将根据定时器的重复设置计算其下一次执行的时间，将其发送给AlarmManager服务，并修改模型层对象保存。

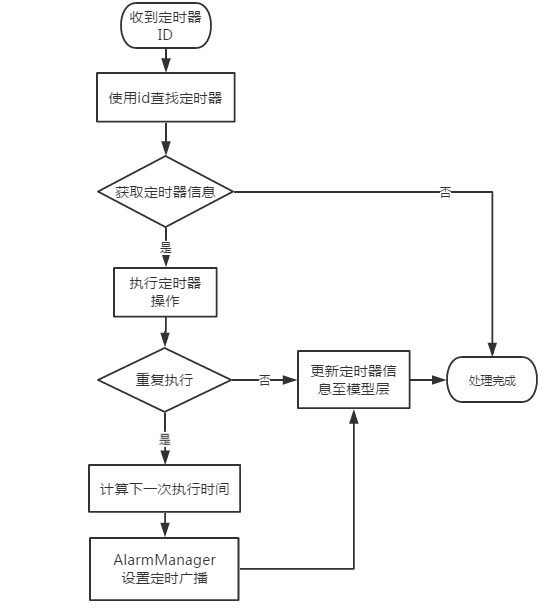


图5.18 TaskService处理流程图

## 5.5第三方开发者服务

除了提供手机应用程序的关键功能外，还需要提供一些额外的功能帮助提升用户的使用体验[13]。在本应用中主要使用第三方服务提供了如下的功能：

1. 版本更新：手机应用程序的日常运营当中，因为新功能的推出，缺陷的修复等原因时常需要发布新版本给用户使用，因此需提供应用版本自动检查更新功能。本软件使用了国内开发者服务提供商友盟的自动更新模块。用户打开软件会自动检查软件版本是否有更新，如有更新则弹窗询问用户是否更新软件。用户也可手动去检查软件版本是否有更新。
2. 用户反馈：用户使用过程中可能会存在各种各样的问题、发现软件的缺陷、有相关的建议意见等，需要有一个渠道与软件的开发商进行交流。本软件使用了友盟提供的用户反馈模块。用户可以在用户反馈界面中给开发商发送消息，开发商通过友盟管理后台能看到用户发送的反馈，并可进行相应的回复，用户打开软件会看到开发商的回复。
3. 统计数据，应用开发商需要了解用户在自己应用上的使用统计数据以便帮助应用程序的运营和改进[14]。比如用户的增长数，活跃用户数、用户打开最多的界面、使用最多的功能等等。这些数据能够帮助应用开发商进一步了解市场，了解用户，分析需求，优化产品。本软件使用的也是由友盟提供的应用统计模块，能在友盟后台查看应用的运营等数据。
4. 分享：智能手机软件的运营中，推广能力是决定软件生死的关键因素之一。因为软件提供了分享功能，能将软件推荐分享给用户的QQ、微信、微博等等好友。该功能也同样使用了友盟提供的服务模块。
5. 错误分析：应用程序在用户手机上可能发生意想不到的崩溃或异常，崩溃和错误的堆栈需要上传到服务器中供开发人员分析错误原因并修复[15]。本软件使用了友盟提供的错误分析服务。
6. 应用加密：Android应用程序使用Java语言开发，其容易被反编译，Android应用的打包也相对容易被破解，导致相关资源文件被非法替换或修改。因此需要对应用程序安装包进行代码加密混淆和资源文件的保护。本软件使用了爱加密提供的应用安装包加密服务。
7. 广告服务：本软件的使用完全免费，因此智能通过植入广告的方式盈利，故植入了多家移动广告运营商的广告服务。本软件在打开和退出界面时会有短暂的图片广告。

# 6总结

论文首先介绍了《定时大师》Android应用程序的需求分析，从需求中得到软件开发的目标和功能设计，再通过经济、技术、运行、法律四个方面的可行性分析得出软件的可行结论。同时分析了市场的同类软件的现状。之后介绍了本软件的开发环境以及主要的技术特点。接着从整体介绍了软件的概要技术设计方案以及主要的界面模块。然后介绍了软件的实现细节，展示了各个界面的具体功能以及后台服务的实现方式。最后从软件的运营角度介绍了第三方服务的使用。

本软件在Android操作系统上使用了Activity、Service、Broadcast receiver、AlarmManager等组件，使用文件存储对用户数据进行持久化保存。实现了定时控制手机的功能、硬件开关、软件等，并支持根据学生课表执行定时任务，帮助学生课堂上专心学习。软件的开发过程中，借鉴了多本Android软件开发书籍，查阅了网上许多相关资料文档。软件的发布运营中，将软件在国内十多个应用商店上架。在软件的后期运营上，根据用户的反馈以及后台收集到的错误堆栈，修复了多个软件缺陷，主要由于Android系统版本多、机型多、机型配置多样等原因造成的。

整个软件的技术上，还有许多可以优化的地方。比如软件的持久化保存，使用文件的方式每次都需要读取整个文件，修改也需要覆盖原文件的所有内容，软件的启动关闭性能因此受到一定的影响，可以优化使用sqlite数据库存储。再比如软件的Service是常驻后台的，可能给手机造成了电量上的损失，可以优化成软件界面退出的同时Service也退出，Service需要的时候由Broadcast receiver启动。

随着智能手机的发展，相信会有越来越多的智能化软件推出，帮助用户更方便智能地使用智能手机。相信在将来，手机能够自动识别场景并根据场景变更自己的设置。

# 参考文献

1. Cay S．Horstmann．Core Java Volume I--Fundamentals[M]．(10th Ed)．Upper Saddle River:prentice hall．2016: 03~100
2. Bill Phillips,Chris Stewart．Android Programming:the Big Nerd Ranch Guide[M]．Atlanta:Big Nerd Ranch Guides．2013: 30~150
3. Brett McLaughlin．Java and XML[M]．(3rd Ed)．California:O'Reilly Media．2006: 30~60
4. 茂生，王明文．软件设计模式及其使用[J]．计算机应用，2002,22(8): 32~33
5. 柯元旦．Android 内核剖析[M]．北京：电子工业出版社，2011: 67~98
6. 毕小朋．精通Android Studio[M]．清华大学出版社．2017: 90~110
7. 李刚．疯狂 Android 讲义[M]．北京：电子工业出版社，2013: 25~42
8. 杨丰盛．Android 技术内幕[M]．北京： 机械工业出版社，2011: 77-89
9. 杨云君．Android 的设计与实现[M]．北京：机械工业出版社，2013: 45~49
10. 丰生强．Android 软件安全与逆向分析[M]．北京：人民邮电出版社，2013: 78~90
11. 余成锋，李代平，毛永华．Android3.0 内存管理机制分析[M]．计算机应用与软件，2013: 55~80
12. 佐冰冰．Android 平台下 Launcher 启动器的设计与实现[D]．哈尔滨工业大学，2012: 2~4
13. 杜吉志，徐明昆．Android 系统内存管理研究及优化[J]．软件，2012,24(5): 69~80
14. 马越．Android 的架构与应用[D]．北京：中国地质大学，2008: 3~4.
15. 姚昱旻，刘卫国．Android 的架构与应用开发研究[J]．计算机系统应用，2008,77(11): 99~111

# 致 谢

在导师杨湘的耐心指导之下，毕业设计和论文编写才得以完成。杨湘老师的帮助和指导意见，贯穿项目的选题到同类软件的调研、需求分析到技术选型再到软件的开发、论文编写到论文的成型。

在此，谨向杨湘老师致以诚挚的敬意和衷心的感谢。

感谢大学所有的授课老师，你们传授与我专业知识，教会我如何自我学习自我进步，为我打开了一扇扇计算机科学的大门。

感谢我的同学，我的朋友。感谢大家在我毕业设计和论文编写中提供了许多帮助和支持。你们也让我大学四年收货了知识的同时也收货了友谊。人生路上我们一起成长。